



ByteXpress - Team 9

ByteXecom E-commerce System

Aphiwe Shoji

u19363967
u19363967@tuks.co.za
0827135234

Kyle van Eeden

u18035176
u18035176@tuks.co.za
0614700577

Nomusa Vumisa (GL)

u17254061
u17254061@tuks.co.za
0662254267

Ofhani Mungani

u18022571
u18022571@tuks.co.za
0766495693

Thenjiwe Ntsonda (PM)

u18139958
u18139958@tuks.co.za
0817473388

ITERATION - 9

System Documentation - This iteration has the collated system documentation of the final versions of all system design documentation of the ByteXpress E-commerce system.

The meticulously compiled documents consist of the updated requirements lists, complexity matrix, logical and technical narratives, technical primitive, UML activity, sequence and state diagrams as well as the test cases and screen designs for all the use cases.



Client Information - Jannes Janse van Rensburg is the co-owner of Natuurlik. He is currently working as a Business Area Manager for DSV - Global Transport and Logistics.



Client

Jannes Janse van Rensburg

Email

jan2rens@gmail.com

Cellphone number

072 881 0004

Table of Contents

1. Iteration Introduction	5
2. Technical Use Case Narratives	6
Introduction	6
2.1. User Management Subsystem.....	6
2.2. Administrative Subsystem.....	44
2.3. Reseller Subsystem	83
2.4. Product Subsystem.....	89
2.5. Order Subsystem	183
2.6. Query Reason Subsystem.....	287
2.7. Review Subsystem.....	313
2.8. Inventory Management Subsystem	332
2.9. Location Subsystem.....	437
2.10. Reporting Subsystem	504
2.11. Supplier Subsystem	568
2.12. Driver Subsystem	603
2.13. VAT Subsystem.....	613
2.14. Reseller Credit Settlement Time Subsystem	633
2.15. Video Subsystem	637
2.16. Help Subsystem	658
Conclusion.....	661
3. Iteration Conclusion	662

Table of Tables

Table 1 - 1.1 Login User.....	6
Table 2 - 1.2 Logout User	10
Table 3 - 1.3 View Profile Details	12
Table 4 - 1.4 Reset Password	16
Table 5 - 1.5 Forgot Password.....	22
Table 6 - 1.6 Register User	27
Table 7 - 1.7 Update Profile Details	36
Table 8 - 1.8 Update Email Address	40
Table 9 - 2.1 Register User	44
Table 10 - 2.2 Search User	62
Table 11 - 2.3 Update User	66
Table 12 - 2.4 Delete User.....	76

Table 13 - 2.5 View Audit Trail	82
Table 14 - 2.6 Search Audit Trail	82
Table 15 - 3.1 Send Payment Reminder	83
Table 16 - 3.2 Capture Proof of Payment	86
Table 17 - 4.1 Add Product	89
Table 18 - 4.2 Search Product	100
Table 19 - 4.3 Update Product	104
Table 20 - 4.4 Delete Product	113
Table 21 - 4.5 Add Product Category	116
Table 22 - 4.6 Search Product Category	122
Table 23 - 4.7 Update Product Category	126
Table 24 - 4.8 Delete Product Category	131
Table 25 - 4.9 Add Product Brand	135
Table 26 - 4.10 Search Product Brand	140
Table 27 - 4.11 Update Product Brand	144
Table 28 - 4.12 Delete Product Brand	149
Table 29 - 4.13 Create Return Reason	153
Table 30 - 4.14 Search Return Reason	159
Table 31 - 4.15 Update Return Reason	163
Table 32 - 4.16 Delete Return Reason	168
Table 33 - 4.17 Browse Product Catalogue	172
Table 34 - 4.18 Configure Product	175
Table 35 - 5.1 Add Item(s) to Cart	183
Table 36 - 5.2 Update Cart	189
Table 37 - 5.3 Remove Item(s) from Cart	192
Table 38 - 5.4 View Cart	195
Table 39 - 5.5 Checkout Cart	198
Table 40 - 5.6 Make Payment	209
Table 41 - 5.7 View Orders Overview	215
Table 42 - 5.8 Search Order	221
Table 43 - 5.9 Query Order	225
Table 44 - 5.10 Review Order	229
Table 45 - 5.11 Create Courier	233
Table 46 - 5.12 Search Courier	238
Table 47 - 5.13 Update Courier	241
Table 48 - 5.14 Delete Courier	245
Table 49 - 5.15 Search Placed Order	248
Table 50 - 5.16 View Placed Order	252
Table 51 - 5.17 Cancel Placed Order	258
Table 52 - 5.18 Cancel Reseller Order	261
Table 53 - 5.19 Approve/Reject Reseller Order	264
Table 54 - 5.20 Dispatch Order	267
Table 55 - 5.21 Log Back Order	270
Table 56 - 5.22 Confirm Order	275
Table 57 - 5.23 Reject Order	279
Table 58 - 5.24 Send Order Confirmation Reminder	282
Table 59 - 5.25 Update Order Confirmation Time	284
Table 60 - 6.1 Create Query Reason	287

Table 61 - 6.2 Search Query Reason	292
Table 62 - 6.3 Update Query Reason	295
Table 63 - 6.4 Delete Query Reason	299
Table 64 - 6.5 Review Order Query.....	302
Table 65 - 6.6 View Order Query Feedback	308
Table 66 - 7.1 Create Review Reason.....	313
Table 67 - 7.2 Search Review Reason	318
Table 68 - 7.3 Update Review Reason	321
Table 69 - 7.4 Delete Review Reason.....	325
Table 70 - 7.5 View Order Review	328
Table 71 - 8.1 Create Inventory Item	332
Table 72 - 8.2 Search Inventory Item.....	338
Table 73 - 8.3 Update Inventory Item.....	341
Table 74 - 8.4 Delete Inventory Item	347
Table 75 - 8.5 Create Inventory Type.....	350
Table 76 - 8.6 Search Inventory Type.....	355
Table 77 - 8.7 Update Inventory Type	358
Table 78 - 8.8 Delete Inventory Type.....	362
Table 79 - 8.9 Create Write-Off Reason.....	365
Table 80 - 8.10 Search Write-Off Reason.....	369
Table 81 - 8.11 Update Write-Off Reason.....	372
Table 82 - 8.12 Delete Write-Off Reason	376
Table 83 - 8.13 Write-Off Inventory.....	379
Table 84 - 8.14 Write-Off Product.....	387
Table 85 - 8.15 Convert Inventory to Product	395
Table 86 - 8.16 Search Procured Inventory.....	403
Table 87 - 8.17 View Procured Inventory Details	407
Table 88 - 8.18 Capture Procured Inventory.....	410
Table 89 - 8.19 Capture Returned Product	416
Table 90 - 8.20 Package Order	424
Table 91 - 8.21 Send Low Inventory Alert.....	433
Table 92 - 8.22 Send Inventory Out of Stock Alert	435
Table 93 - 9.1 Create Country	437
Table 94 - 9.2 Search Country	443
Table 95 - 9.3 Update Country	447
Table 96 - 9.4 Delete Country	452
Table 97 - 9.5 Add Province	456
Table 98 - 9.6 Search Province.....	461
Table 99 - 9.7 Update Province.....	464
Table 100 - 9.8 Delete Province	468
Table 101 - 9.9 Create City	472
Table 102 - 9.10 Search City.....	477
Table 103 - 9.11 Update City.....	480
Table 104 - 9.12 Delete City	484
Table 105 - 9.13 Add Suburb.....	488
Table 106 - 9.14 Search Suburb	493
Table 107 - 9.15 Update Suburb	496
Table 108 - 9.16 Delete Suburb.....	500

Table 109 - 10.1 Generate Inventory List Report.....	504
Table 110 - 10.2 Generate Sales (by Top 5 Clients) Report	508
Table 111 - 10.3 Generate Product List Report	518
Table 112 - 10.4 Generate Customer List Report	522
Table 113 - 10.5 Generate Reseller List Report	526
Table 114 - 10.6 Generate Produced Products Report	530
Table 115 - 10.7 Generate Orders (by Status) Report	538
Table 116 - 10.8 Generate Orders (by Region) Report	542
Table 117 - 10.9 Generate Sales (by Product) Report.....	546
Table 118 - 10.10 Generate Sales (by Month) Report	553
Table 119 - 10.11 View Dashboard Overview	560
Table 120 - 11.1 Create Supplier	568
Table 121 - 11.2 Search Supplier.....	578
Table 122 - 11.3 Update Supplier	582
Table 123 - 11.4 Delete Supplier	589
Table 124 - 11.5 Send Supplier Order	594
Table 125 - 12.1 View Deliveries.....	603
Table 126 - 12.2 View Previous Deliveries	607
Table 127 - 12.3 Confirm Delivery	610
Table 128 - 13.1 Create VAT	613
Table 129 - 13.2 Search VAT	620
Table 130 - 13.3 Update VAT Details	623
Table 131 - 13.4 Delete VAT.....	629
Table 132 - 14.1 Update Reseller Credit Settlement Time	633
Table 133 - 15.1 Add Video.....	637
Table 134 - 15.2 Search Video	643
Table 135 - 15.3 Update Video Details	647
Table 136 - 15.4 Delete Video.....	652
Table 137 - 15.5 View Video	655
Table 138 - 16.1 View Help Document	658

1. Iteration Introduction

For this iteration the ByteXpress team have compiled a document that illustrates the collated system documentation of the final versions of all system design documentation for the ByteXpress E-commerce system.

2. Technical Use Case Narratives

Introduction

The following technical use case narratives will be describing the various interactions to take place between end-users or other systems and the system to be developed. The actors, their interactions with the ByteXecom system, automated system responses and technologies to be used for each use case is described along with identified business rules, design specifications and constraints.

2.1. User Management Subsystem

Table 1 - 1.1 Login User

USER MANAGEMENT SUBSYSTEM		
AUTHOR(s): Kyle van Eeden		DATE: 17/04/2022
VERSION: 1.3		LAST REVIEW DATE: 27/04/2022
USE CASE NAME:	Login	USE CASE TYPE
USE CASE ID:	1.1	Business Requirements: ☐
PRIORITY:	High	System Analysis: ☐
SOURCE:	ByteXpress Requirements Document	System Design: ☒
PRIMARY BUSINESS ACTOR:	User	
PRIMARY THE SYSTEM ACTOR:	None	
OTHER PARTICIPATING ACTORS:	None	
OTHER INTERESTED STAKEHOLDERS:	None	
DESCRIPTION:	This use case describes the event of a system user logging into the system using their account credentials. The use case begins with the user navigating to the ByteXecom website's home page, and the user selecting the "Login" button. The system will display the login screen where the user is to provide their login credentials and the system will then validate the login details provided. The use case will conclude once the user is successfully logged onto the system.	
PRE-CONDITION:	The system user must have registered on the system and their account details must still be stored in the database in order to be successfully authenticated. The user must not already be logged into the system on their current or any other device.	
TRIGGER:	The user wishes to log into the system.	

TYPICAL COURSE OF EVENTS:	ACTOR ACTION:	SYSTEM RESPONSE:																												
		Manual Action	Automated Action																											
	Step 1: The user clicks on the “ Login ” nav link on the navigation bar		Step 2: The system displays the Login screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Login</td><td>Label</td><td>Main heading</td></tr><tr><td>Email</td><td>Label</td><td>This control will be used as a label to indicate the users email address attribute</td></tr><tr><td>Email</td><td>Textbox</td><td>This control will be used by the user to enter their email address</td></tr><tr><td>Password</td><td>Label</td><td>This control will be used as a label to indicate the user's password attribute</td></tr><tr><td>Password</td><td>Textbox</td><td>This control will be used by the user to enter their password</td></tr><tr><td>Login</td><td>Button</td><td>This control will be used to submit the login form</td></tr><tr><td>Forgot your password?</td><td>Hyperlink</td><td>Navigates to the Forgot Password screen</td></tr><tr><td>Don't have an account? Register</td><td>Hyperlink</td><td>Navigates to the Register screen</td></tr></table> Note: All controls are enabled by default, unless otherwise stated. All text input fields are empty by default, unless stated otherwise.	Control Name	Control Type	Notes	Login	Label	Main heading	Email	Label	This control will be used as a label to indicate the users email address attribute	Email	Textbox	This control will be used by the user to enter their email address	Password	Label	This control will be used as a label to indicate the user's password attribute	Password	Textbox	This control will be used by the user to enter their password	Login	Button	This control will be used to submit the login form	Forgot your password?	Hyperlink	Navigates to the Forgot Password screen	Don't have an account? Register	Hyperlink	Navigates to the Register screen
Control Name	Control Type	Notes																												
Login	Label	Main heading																												
Email	Label	This control will be used as a label to indicate the users email address attribute																												
Email	Textbox	This control will be used by the user to enter their email address																												
Password	Label	This control will be used as a label to indicate the user's password attribute																												
Password	Textbox	This control will be used by the user to enter their password																												
Login	Button	This control will be used to submit the login form																												
Forgot your password?	Hyperlink	Navigates to the Forgot Password screen																												
Don't have an account? Register	Hyperlink	Navigates to the Register screen																												
	Step 3: The user enters their details and clicks on the Login button. [ALT]		Step 4: The system validates that the entered login credentials are valid against the user account's details retrieved from the User table using an SQL READ query and the following attributes: <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Email Address</td><td>EmailAddress</td></tr><tr><td>Password</td><td>Password</td></tr></table> [ALT]	Details to be read	Attributes in the table	Email Address	EmailAddress	Password	Password																					
Details to be read	Attributes in the table																													
Email Address	EmailAddress																													
Password	Password																													
			Step 5: The system uses an SQL READ query to retrieve the role assigned to the user in the UserRole table, to determine what																											

			access is to be granted to the user using the following attribute: <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Role Id</td><td>Role_Id</td></tr></table>	Details to be read	Attributes in the table	Role Id	Role_Id											
Details to be read	Attributes in the table																	
Role Id	Role_Id																	
			Step 6: The system generates an authentication cookie which is stored in local storage.															
			Step 7: The system logs the user into the system and replaces the ' Register ' and ' Login ' buttons in the navigation bar to indicate the user is authenticated with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>My Profile</td><td>Hyperlink</td><td>Invokes use case 1.3 View Profile Details</td></tr><tr><td>Logout</td><td>Button</td><td>Logs the user out of the system</td></tr></table>	Control Name	Control Type	Notes	My Profile	Hyperlink	Invokes use case 1.3 View Profile Details	Logout	Button	Logs the user out of the system						
Control Name	Control Type	Notes																
My Profile	Hyperlink	Invokes use case 1.3 View Profile Details																
Logout	Button	Logs the user out of the system																
ALTERNATE COURSES	ALT Step 3A: The user does not remember their password and selects the 'Forgot Password' hyperlink: - The system invokes Use Case 1.5 Forgot Password - The use case is terminated																	
	ALT Step 3B: The user does not have an existing account registered on the system and selects the Register hyperlink: - The system invokes Use Case 1.6 Register Customer - The use case is terminated																	
	ALT Step 4: The details entered are not in the correct format or the required fields are empty The system displays a validation error with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Email Address</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>The Email Address field is required or "Invalid Email Address provided"</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Email Address textbox</td></tr><tr><td>Password</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>The Password field is required or "Invalid Password provided"</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Password textbox</td></tr></table>			Control Name	Control Name	Notes	Email Address	Textbox	This control will be outline in red to show a validation error	The Email Address field is required or "Invalid Email Address provided"	Label	This control will be used as a label to indicate a validation error in the Email Address textbox	Password	Textbox	This control will be outline in red to show a validation error	The Password field is required or "Invalid Password provided"	Label	This control will be used as a label to indicate a validation error in the Password textbox
	Control Name	Control Name	Notes															
	Email Address	Textbox	This control will be outline in red to show a validation error															
The Email Address field is required or "Invalid Email Address provided"	Label	This control will be used as a label to indicate a validation error in the Email Address textbox																
Password	Textbox	This control will be outline in red to show a validation error																
The Password field is required or "Invalid Password provided"	Label	This control will be used as a label to indicate a validation error in the Password textbox																
Return to Step 3																		
CONCLUSION:	The use case concludes when the user is logged into the system																	
POST-CONDITION:	The user is logged into the system and granted access based on their role assigned.																	
BUSINESS RULES:	All account passwords stored in the database should be encrypted.																	

	• The system is to implement access control based on the pre-defined roles assigned to user accounts.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	• The Login button should not be visible to users who are currently logged into the system. • The email address of the account holder is to be used as the username. • The system should have a vague login failure message so that it does not give potential hackers the information they need to potentially breach an account they do not own.
ASSUMPTIONS:	None
OPEN ISSUES:	None

Table 2 - 1.2 Logout User

USER MANAGEMENT SUBSYSTEM											
AUTHORS (s): Kyle van Eeden			DATE: 17/04/2022								
VERSION: 1.3			LAST REVIEW DATE: 20/08/2022								
USE CASE NAME:	Logout		USE CASE TYPE								
USE CASE ID:	1.2		Business Requirements:	☐							
PRIORITY:	High		System Analysis:	☐							
SOURCE:	ByteXpress Requirements Document		System Design:	☒							
PRIMARY BUSINESS ACTOR:	User										
PRIMARY THE SYSTEM ACTOR:	None										
OTHER PARTICIPATING ACTORS:	None										
OTHER INTERESTED STAKEHOLDERS:	None										
DESCRIPTION:	This use case will delineate the process of a system user logging out of the system. The use case begins with the user selecting the “Logout” button in the navigation bar and concludes once the user is successfully logged out of the system and the navigation controls are set back to the default options for an unauthenticated user.										
PRE-CONDITION:	The system user must currently be logged into the system.										
TRIGGER:	The user wishes to log out of the system.										
TYPICAL COURSE OF EVENTS:	ACTOR ACTION:	SYSTEM RESPONSE:									
		Manual Action	Automated Action								
	Step 1: The user clicks on the “Logout” nav item in the navigation bar.		Step 2: The system removes the user’s authentication cookie details from local storage.								
			Step 3: The system logs the user out of the system and navigates the user to the Home screen and replaces the controls displayed in the navigation bar to the following:								
			<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Register</td><td>Nav link</td><td>Invokes use case 1.6 Register Customer</td></tr><tr><td>Login</td><td>Nav link</td><td>Invokes use case 1.1 Login</td></tr></table>	Control Name	Control Type	Notes	Register	Nav link	Invokes use case 1.6 Register Customer	Login	Nav link
Control Name	Control Type	Notes									
Register	Nav link	Invokes use case 1.6 Register Customer									
Login	Nav link	Invokes use case 1.1 Login									
ALTERNATE COURSES	None										
CONCLUSION:	The use case concludes when the user is successfully logged out of the system.										

POST-CONDITION:	The controls displayed in the navigation bar for a logged in user are changed to 'Register' and 'Login' nav links.
BUSINESS RULES:	Authenticated users should be logged out of the system automatically after 15 minutes of inactivity.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	• The Logout button should not be visible to users who are not logged into the system. • Unauthenticated users should be able to see the "Register" and "Login" option in the navigation bar at all times.
ASSUMPTIONS:	None
OPEN ISSUES:	None

Table 3 - 1.3 View Profile Details

USER MANAGEMENT SUBSYSTEM																	
AUTHORS (s): Ofhani Mungani		DATE: 18/04/2022															
VERSION: 1.0		LAST REVIEW DATE: 18/04/2022															
USE CASE NAME:	View Profile Details	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒															
USE CASE ID:	1.3																
PRIORITY:	High																
SOURCE:	ByteXpress Requirements List																
PRIMARY BUSINESS ACTOR	• User																
PRIMARY SYSTEM ACTOR	• None																
OTHER PARTICIPATING ACTORS:	• None																
OTHER INTERESTED STAKEHOLDERS:	• None																
DESCRIPTION:	This use case describes an event where the user wants to view their profile on the system. The use case begins when the user requests to view their details and the system responds by retrieving the details. The use case concludes when the details are displayed on the screen																
PRE-CONDITION:	The user must be logged on the system																
TRIGGER:	The use case is called when the user wants to view their details																
TYPICAL COURSE OF EVENTS:	Actor Action	System Response															
		Manual Action	Automated Action														
	Step 1: The user clicks on the MY PROFILE nav link in the navigation bar		Step 2: The system retrieves the following attributes from the following tables, using a method in a .NET controller and an SQL READ query: From the User table. <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>First Name</td><td>FirstName</td></tr><tr><td>Surname</td><td>Surname</td></tr><tr><td>Phone Number</td><td>PhoneNumber</td></tr><tr><td>Street Address</td><td>StreetAddress</td></tr></table> From the Country Table <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Country</td><td>CountryName</td></tr></table>	Details to be read	Attributes in the table	First Name	FirstName	Surname	Surname	Phone Number	PhoneNumber	Street Address	StreetAddress	Details to be read	Attributes in the table	Country	CountryName
Details to be read			Attributes in the table														
First Name			FirstName														
Surname			Surname														
Phone Number			PhoneNumber														
Street Address			StreetAddress														
Details to be read			Attributes in the table														
Country			CountryName														

			From the Province Table <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Province</td><td>ProvinceName</td></tr></table> From the City Table <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>City</td><td>CityName</td></tr></table> From the Suburb Table <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Suburb</td><td>SuburbName</td></tr></table>	Details to be read	Attributes in the table	Province	ProvinceName	Details to be read	Attributes in the table	City	CityName	Details to be read	Attributes in the table	Suburb	SuburbName												
	Details to be read	Attributes in the table																									
	Province	ProvinceName																									
	Details to be read	Attributes in the table																									
	City	CityName																									
	Details to be read	Attributes in the table																									
	Suburb	SuburbName																									
			Step 3: The system displays the user's details on the Profile Details Screen which contains an MVC form with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>My Account</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Profile Details</td><td>Subheading</td><td>Page Subheading</td></tr><tr><td>First Name</td><td>Label</td><td>This control will be used as a label to indicate the users first name</td></tr><tr><td>First Name</td><td>Textbox</td><td>This control will be used by the users to insert their first name</td></tr><tr><td>Surname</td><td>Label</td><td>This control will be used as a label to indicate the user's surname</td></tr><tr><td>Surname</td><td>Textbox</td><td>This control will be used by the users to insert their surname</td></tr><tr><td>Email</td><td>Label</td><td>This control will be used as a label to indicate the</td></tr></table>	Control Name	Control Type	Notes	My Account	Heading	Page main heading	Profile Details	Subheading	Page Subheading	First Name	Label	This control will be used as a label to indicate the users first name	First Name	Textbox	This control will be used by the users to insert their first name	Surname	Label	This control will be used as a label to indicate the user's surname	Surname	Textbox	This control will be used by the users to insert their surname	Email	Label	This control will be used as a label to indicate the
	Control Name	Control Type	Notes																								
	My Account	Heading	Page main heading																								
Profile Details	Subheading	Page Subheading																									
First Name	Label	This control will be used as a label to indicate the users first name																									
First Name	Textbox	This control will be used by the users to insert their first name																									
Surname	Label	This control will be used as a label to indicate the user's surname																									
Surname	Textbox	This control will be used by the users to insert their surname																									
Email	Label	This control will be used as a label to indicate the																									

					user's email address
			Email	Textbox (disabled)	This control will be used to display their email address
			Phone Number	Label	This control will be used as a label to indicate the users phone number attribute
			Phone Number	Textbox	This control will be used by the user to enter their phone number
			Street Name	Label	This control will be used as a heading to display the user's street address
			Street Name	Textbox	This control will be used by the users to insert their street address
			Country	Label	This control will be used as a heading to display the user's country name
			Country	Dropdown list	This control will be used by the users to choose their country name
			Province	Label	This control will be used as a heading to display the user's province name
			Province	Dropdown list	This control will be used by the users to choose their province name
			City	Label	This control will be used as a heading to display the user's city name
			City	Dropdown list	This control will be used by the users to choose their city name

			Suburb	Label	This control will be used as a heading to display the user's suburb name
			Suburb	Dropdown list	This control will be used by the users to choose their suburb name
			Save	Button	This control is used to save user's profile details.
			Note: All controls except the email textbox are enabled by default.		
ALTERNATE COURSES:	None				
CONCLUSION:	This use case concludes when the user details are displayed				
POST-CONDITION:	Attributes were retrieved from the User table				
BUSINESS RULES	None				
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	- The user cannot view their details without an internet connection - All uses have access to this function				
ASSUMPTIONS:	None				
OPEN ISSUES:	None				

Table 4 - 1.4 Reset Password

USER MANAGEMENT SUBSYSTEM			
AUTHOR(s): Kyle van Eeden		DATE: 17/04/2022	
VERSION: 1.0		LAST REVIEW DATE: 21/04/2022	
USE CASE NAME:	Reset Password	USE CASE TYPE	
USE CASE ID:	1.4	Business Requirements:	☐
PRIORITY:	High	System Analysis:	☐
SOURCE:	ByteXpress Requirements Document	System Design:	☒
PRIMARY BUSINESS ACTOR:	User		
PRIMARY THE SYSTEM ACTOR:	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This use case describes the process of a user resetting the current password of their account. The use case will begin with the user requesting to update their current password and the system responds by prompting the user to specify their old password, a new password, and confirmation of the new password. Once the user has entered the required password details and clicked the Update Password button, the system will validate the captured password details and if validation passes, the system will hash the New Password provided and save the updated password in the User entity. The use case concludes once the system has displayed a confirmation message box control which informs the user that their password has been successfully reset.		
PRE-CONDITION:	The user must be logged into the system.		
TRIGGER:	The registered system user wishes to update the password of their account.		
TYPICAL COURSE OF EVENTS:	ACTOR ACTION:	SYSTEM RESPONSE:	
		Manual Action	Automated Action
	Step 1: The user selects the "My Account" hyperlink displayed in the navigation bar.		Step 2: The system retrieves the user's details from the User table using a .NET controller method and a SQL Read Query, and populates the View Profile Details screen, which is displayed with the following controls:

			Control Name	Control Type	Notes
			Profile Details	Heading	This control is used to display the heading of the View profile screen
			Reset Password	Button	Displayed in the side-navigation bar. Navigates the user to the 'Reset Password' page.
			First Name	Label	This control is used for to describe the content in the First Name textbox
			First Name	Textbox	Populated with the [FirstName] value retrieved for the user
			Surname	Label	This control is used for to describe the content in the Surname textbox
			Surname	Textbox	Populated with the [Surname] value retrieved for the user
			Email	Label	This control is used for to describe the content in the Email textbox
			Email	Textbox	Read-Only; Populated with the [Email] value retrieved for the user
			Phone Number	Label	This control is used for to describe the content in the Email textbox
			Phone Number	Textbox	Populated with the [PhoneNumber] value retrieved for the user
			Street Address	Label	This control is used for to describe the content in the Email textbox
			Street Address	Textbox	Populated with the [StreetAddress]

					value retrieved for the user
			Country	Label	Used for the Country drop-down list
			Country	Textbox	Populated with the [CountryName] value retrieved for the user
			Province	Label	Used for the Province drop-down list
			Province	Textbox	Populated with the [ProvinceName] value retrieved for the user
			City	Label	Used for the City drop-down list
			City	Textbox	Populated with the [City] value retrieved for the user
			Suburb	Label	Used for the Suburb drop-down list
			Suburb	Textbox	Populated with the [Suburb] value retrieved for the user
			Back	Button	Navigates administrator back to the Users page if clicked
			Update	Button	Allows for the submission of updated user details to the User table in the database.
	Step 3: The user selects the Reset Password navigation hyperlink.		Step 4: The system displays the Reset Password screen with the following controls:		
			Control Name	Control Type	Notes
			Update Password	Heading	This control is used to display the heading of the Reset Password screen.
			Current Password	Label	Label used to display the name

					for the Current Password field.
			Current Password	Textbox	This control is used to capture the user's current password.
			New Password	Label	Label used to display the name for the New Password field.
			New Password	Textbox	This control is used to capture the user's New password.
			Confirm New Password	Label	Label used to display the name for the Confirm New Password field.
			Password Confirmation	Textbox	This control is used by the user to reconfirm their new password and must match the value provided in the New Password textbox.
			Update Password	Button	The system first validates the input provided and then attempts to reset the user's password once clicked.
			Additional Note: All controls are enabled by default, unless specified otherwise in the 'Notes' section.		
	Step 5: The user enters their Current Password, the New Password, and the Password Confirmation then clicks the Update Password button.		Step 6: The system validates that all required fields were populated, the New Password meets the minimum criteria and the Current Password value provided matches that of the decrypted password retrieved. The system uses a .NET controller method with a SQL Read query to retrieve the [PasswordHash] attribute value from the User table and decrypts the password with a hashing algorithm to check for a match with the Current Password field value provided. [ALT]		

			Step 7: The system hashes the New Password provided using a PDKDF2 algorithm with HMAC-SHA256.												
			Step 8: The system updates the password using a .NET Controller with a SQL Update query to save the newly hashed master key for the [PasswordHash] field in the User table.												
			Step 9: The system displays a message box to inform the user that their password has been successfully updated with the following controls:												
			<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Password Updated Successfully</td><td>Label</td><td>Message displayed to inform user that their password was successfully updated.</td></tr><tr><td>X</td><td>Button</td><td>Closes the confirmation notification once clicked.</td></tr></table>	Control Name	Control Type	Notes	Password Updated Successfully	Label	Message displayed to inform user that their password was successfully updated.	X	Button	Closes the confirmation notification once clicked.			
			Control Name	Control Type	Notes										
			Password Updated Successfully	Label	Message displayed to inform user that their password was successfully updated.										
X	Button	Closes the confirmation notification once clicked.													
ALTERNATE COURSES	ALT Step 6 a : The system raises a validation error when it detects that not all required fields have been provided with a value and displays the following validation errors if applicable:														
	<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Current Password field is required.</td><td>Label</td><td>Informs the user that they must provide their current password.</td></tr><tr><td>New Password Field is required.</td><td>Label</td><td>Informs the user that they must provide a new password.</td></tr><tr><td>Password Confirmation Field is required.</td><td>Label</td><td>Informs the user that they must provide reconfirm their new password to be set.</td></tr></table>			Control Name	Control Type	Notes	Current Password field is required.	Label	Informs the user that they must provide their current password.	New Password Field is required.	Label	Informs the user that they must provide a new password.	Password Confirmation Field is required.	Label	Informs the user that they must provide reconfirm their new password to be set.
	Control Name	Control Type	Notes												
	Current Password field is required.	Label	Informs the user that they must provide their current password.												
	New Password Field is required.	Label	Informs the user that they must provide a new password.												
	Password Confirmation Field is required.	Label	Informs the user that they must provide reconfirm their new password to be set.												
Return to Step 5															
ALT Step 6 b: The system detects that the current password provided does not match that which is stored in the User table for the specific user:															
- The system displays the following validation error message: Label: 'Incorrect password.' - Return to Step 5															
ALT Step 6 c: Invalid input has been provided and the system displays relevant validation error messages using the following controls:															
<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Passwords must have at least one uppercase character, one lowercase</td><td>Label</td><td>Informs the user that they must provide a password that contains at least 1 uppercase letter, 1</td></tr></table>			Control Name	Control Type	Notes	Passwords must have at least one uppercase character, one lowercase	Label	Informs the user that they must provide a password that contains at least 1 uppercase letter, 1							
Control Name	Control Type	Notes													
Passwords must have at least one uppercase character, one lowercase	Label	Informs the user that they must provide a password that contains at least 1 uppercase letter, 1													

	character, a digit and one alpha-numeric character.		lowercase character, a digit and a non-alphanumeric character.
	The New password must be at least six and at maximum 100 characters long.	Label	Informs the user that a password must be between 6 and 100 characters long.
	'The new password and confirmation password do not match'	Label	Informs the user that the new password and password confirmation fields did not match exactly.
	Return to Step 5		
CONCLUSION:	The use case concludes when the system informs the user that their password has been reset successfully.		
POST-CONDITION:	- The user's password is updated in the User table. - The master key stored in the [PasswordHash] field is hashed.		
BUSINESS RULES:	- The user has to confirm the new password to be set for their account before the password is updated in the database. - The user is required to confirm their current password. - All passwords stored in the database need to be hashed for security purposes.		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	The new password provided should contain at least six characters, must include one uppercase and lowercase character, at least one digit and at least one non-alphanumeric character.		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

Table 5 - 1.5 Forgot Password

USER MANAGEMENT SUBSYSTEM					
AUTHOR(s): Nomusa Vumisa			DATE: 23/04/2022		
VERSION: 1.2			LAST REVIEW DATE:20/08/2022		
USE CASE NAME:	Forgot Password		USE CASE TYPE		
USE CASE ID:	1.5		Business Requirements: ☐		
PRIORITY:	High		System Analysis: ☐		
SOURCE:	ByteXpress Requirements List		System Design: ☒		
PRIMARY BUSINESS ACTOR:	User				
PRIMARY THE SYSTEM ACTOR:	None				
OTHER PARTICIPATING ACTORS:	None				
OTHER INTERESTED STAKEHOLDERS:	None				
DESCRIPTION:	This use case describes an event where the user forgets their password. The use case begins when the user requests to set a new password on the system, as they forgot their current one. The system responds by requesting the user to enter the email address to which the reset password link will be sent. Once the link has been received and accessed, the user will be able to change their password. The use case concludes when the user logs in with their new password and accesses the system.				
PRE-CONDITION:	The user must be registered on the system				
TRIGGER:	The use case is called when the user forgets their password				
TYPICAL COURSE OF EVENTS:	ACTOR ACTION:	SYSTEM RESPONSE:			
		Manual Action	Automated Action		
	Step 1: The user clicks on the “Forgot your password?” link		Step 2: The system displays the Forgot Password screen with the following controls:		
			Control Name	Control Type	Notes
			Forgot Password	Heading	Page heading
			Email	Label	This control will be used as a label to indicate the email address attribute of the user record
	Email	Textbox	This control will be used by the user to insert their email address		

			<table><tr><td>Reset Password</td><td>Button</td><td>This control will be used to send the reset password link to the user</td></tr></table>	Reset Password	Button	This control will be used to send the reset password link to the user																
	Reset Password	Button	This control will be used to send the reset password link to the user																			
			Note: All controls are enabled by default, unless otherwise stated. All text input fields are empty by default, unless stated otherwise.																			
	Step 3: The user enters their email address		Step 4: The system validates the entered email address against the User table: [ALT] <table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Email</td><td>EmailAddress</td></tr></table>	Details to be read	Attributes in the table	Email	EmailAddress															
	Details to be read	Attributes in the table																				
	Email	EmailAddress																				
			Step 5: The system uses an SQL READ query to check if the email address exists in the User table using the following attribute: [ALT] <table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Email</td><td>EmailAddress</td></tr></table>	Details to be read	Attributes in the table	Email	EmailAddress															
	Details to be read	Attributes in the table																				
	Email	EmailAddress																				
			Step 6: The system generates a new call back URI and email message with the following details: <table><tr><td>Details</td><td>Notes</td></tr><tr><td>URL</td><td>The reset password page that the user will be directed to after accessing the reset password link</td></tr><tr><td>Token</td><td>The reset password token generated in a .NET controller</td></tr><tr><td>To</td><td>The entered user email address</td></tr><tr><td>From</td><td>The company email address, saved as a variable on the system</td></tr><tr><td>Subject</td><td>"Reset Password"</td></tr><tr><td>Email Template</td><td>Retrieved from the wwwroot folder</td></tr><tr><td>Body – Main Text</td><td>We've received a request to reset your password. Click the link below to reset your Natuurlik account password.</td></tr><tr><td>Button</td><td>"Reset Password" directs the user to the Reset Password screen</td></tr><tr><td>Body – Main Text</td><td>If you did not request to change your password, you can ignore this email.</td></tr></table>	Details	Notes	URL	The reset password page that the user will be directed to after accessing the reset password link	Token	The reset password token generated in a .NET controller	To	The entered user email address	From	The company email address, saved as a variable on the system	Subject	"Reset Password"	Email Template	Retrieved from the wwwroot folder	Body – Main Text	We've received a request to reset your password. Click the link below to reset your Natuurlik account password.	Button	"Reset Password" directs the user to the Reset Password screen	Body – Main Text
Details	Notes																					
URL	The reset password page that the user will be directed to after accessing the reset password link																					
Token	The reset password token generated in a .NET controller																					
To	The entered user email address																					
From	The company email address, saved as a variable on the system																					
Subject	"Reset Password"																					
Email Template	Retrieved from the wwwroot folder																					
Body – Main Text	We've received a request to reset your password. Click the link below to reset your Natuurlik account password.																					
Button	"Reset Password" directs the user to the Reset Password screen																					
Body – Main Text	If you did not request to change your password, you can ignore this email.																					
		Step 7: The system sends the reset password email to the user using the Sendgrid API.																				
		Step 8: The user is directed to the Forgot Password Confirmation page which has the following controls: <table><tr><td>Control Name</td><td>Control Type</td><td>Notes</td></tr><tr><td>Forgot Password Confirmation</td><td>Heading</td><td>Page heading</td></tr><tr><td>"Please check your email to reset your password"</td><td>Label</td><td>This control will be used to inform the user that the reset password link has been sent to</td></tr></table>	Control Name	Control Type	Notes	Forgot Password Confirmation	Heading	Page heading	"Please check your email to reset your password"	Label	This control will be used to inform the user that the reset password link has been sent to											
Control Name	Control Type	Notes																				
Forgot Password Confirmation	Heading	Page heading																				
"Please check your email to reset your password"	Label	This control will be used to inform the user that the reset password link has been sent to																				

					their email address	
	Step 9: The user clicks on the link sent to their email address		Step 10: The system directs the user to the “Reset Password” screen which has the following attributes:	Control Name	Control Type	Notes
				Reset Password	Heading	Page heading
				Password	Label	This control will be used as a label to indicate the password attribute of the user record
				Password	Textbox	This control will be used by the user to insert their new password
				Confirm Password	Label	This control will be used as a label to indicate the password attribute of the user record
				Confirm Password	Textbox	This control will be used by the user to confirm their new password
				Reset	Button	This control will be used to submit the entered details
	Note: All controls are enabled by default, unless otherwise stated. All text input fields are empty by default, unless stated otherwise.					
	Step 11: The user enters their new password and confirms it by entering their password again		Step 12: The system validates the details provided [ALT]	Details to be read	Attributes in the table	Notes
Password				Password	Required Maximum:100 Minimum: 6 Must have a combination of different characters Must contain at least one uppercase letter Cannot be the same as the old password	
Confirm Password				Not stored in the table	Used to confirm the new password	

			Step 12: The system saves the entered details into the User table using an SQL UPDATE query:								
			<table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Password</td><td>Password</td><td>Required Maximum:100 Minimum: 6 Must have a combination of different characters Must contain at least one uppercase letter Cannot be the same as the old password</td></tr></table>	Details to be read	Attributes in the table	Notes	Password	Password	Required Maximum:100 Minimum: 6 Must have a combination of different characters Must contain at least one uppercase letter Cannot be the same as the old password		
	Details to be read	Attributes in the table	Notes								
Password	Password	Required Maximum:100 Minimum: 6 Must have a combination of different characters Must contain at least one uppercase letter Cannot be the same as the old password									
		Step 13: The system directs the user to the Reset Password Confirmation screen which has the following controls:									
		<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Forgot Password Confirmation</td><td>Heading</td><td>Page heading</td></tr><tr><td>"Your password has been reset. Please click here to log in"</td><td>Label and link</td><td>This control will be used to inform the user that their password has been reset</td></tr></table>	Control Name	Control Type	Notes	Forgot Password Confirmation	Heading	Page heading	"Your password has been reset. Please click here to log in"	Label and link	This control will be used to inform the user that their password has been reset
Control Name	Control Type	Notes									
Forgot Password Confirmation	Heading	Page heading									
"Your password has been reset. Please click here to log in"	Label and link	This control will be used to inform the user that their password has been reset									
ALTERNATE COURSES	ALT Step 4: The email entered is not in the correct format or the field is empty										
	The system displays a validation error with the following controls										
	<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>The Email field is not a valid e-mail address</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Email textbox</td></tr><tr><td>The Email field is required</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Email textbox</td></tr></table>	Control Name	Control Name	Notes	The Email field is not a valid e-mail address	Label	This control will be used as a label to indicate a validation error in the Email textbox	The Email field is required	Label	This control will be used as a label to indicate a validation error in the Email textbox	
Control Name	Control Name	Notes									
The Email field is not a valid e-mail address	Label	This control will be used as a label to indicate a validation error in the Email textbox									
The Email field is required	Label	This control will be used as a label to indicate a validation error in the Email textbox									
	Return to step 3										
	ALT Step 5: The email address does not exist in the system										
	The system displays a validation error with the following controls										
	<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>The Email address does not exist</td><td>Label</td><td>This control will be used to indicate a validation error in the Email textbox</td></tr></table>	Control Name	Control Name	Notes	The Email address does not exist	Label	This control will be used to indicate a validation error in the Email textbox				
Control Name	Control Name	Notes									
The Email address does not exist	Label	This control will be used to indicate a validation error in the Email textbox									
	Return to step 3										
	ALT Step 12: The details entered is not in the correct format or the field is empty										
	The system displays a validation error with the following controls										
	<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr></table>	Control Name	Control Name	Notes							
Control Name	Control Name	Notes									

	The Password field is required	Label	This control will be used as a label to indicate a validation error in the Password textbox
	The Password must be at least 6 and at max 100 characters long.	Label	This control will be used as a label to indicate a validation error in the Password textbox
	Passwords must have at least one non alphanumeric character.	Label	This control will be used as a label to indicate a validation error in the Password textbox
	Passwords must have at least one digit ('0'-'9').	Label	This control will be used as a label to indicate a validation error in the Password textbox
	Passwords must have at least one uppercase ('A'-'Z').	Label	This control will be used as a label to indicate a validation error in the Password textbox
	The password and confirmation password do not match.	Label	This control will be used as a label to indicate a validation error in the Confirm Password textbox
Return to step 11			
CONCLUSION:	This use case concludes when the user's password has been updated successfully		
POST-CONDITION:	The user can login to the system using their new password		
BUSINESS RULES:	- The password must be at least 6 characters and max of 100 - The password must have at least one digit - The password should contain at least one uppercase letter - Passwords must have at least one non alphanumeric character.		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	- The user cannot change their password without an internet connection - The password must be hashed - All uses have access to this function		
ASSUMPTIONS:	The user has access to their email account		
OPEN ISSUES:	None		

Table 6 - 1.6 Register User

USER MANAGEMENT SUBSYSTEM								
AUTHOR(s): Kyle van Eeden		DATE: 11/05/2022						
VERSION: 1.4		LAST REVIEW DATE: 22/08/2022						
USE CASE NAME:	Register Customer	USE CASE TYPE						
USE CASE ID:	1.6	Business Requirements:	☐					
PRIORITY:	High	System Analysis:	☐					
SOURCE:	ByteXpress Requirements Document	System Design:	☒					
PRIMARY BUSINESS ACTOR:	Customer							
PRIMARY THE SYSTEM ACTOR:	None							
OTHER PARTICIPATING ACTORS:	None							
OTHER INTERESTED STAKEHOLDERS:	None							
DESCRIPTION:	This use case will describe the process of a customer registering an account on the ByteXecom system. The use case will begin once the customer selects the option to register a new account. The user will be prompted to provide all the required details to register a new account and after having entered the details and successfully creating the account, an email will be sent to the now registered customer which contains a URI link for confirming the account. The use case concludes once the new customer account is successfully added to the User table and has been confirmed by the user via email.							
PRE-CONDITION:	The customer must not yet have registered on the system using the Email Address they intend to use. The potential customer must not be logged into any account.							
TRIGGER:	The potential customer wishes to register an account on the system.							
TYPICAL COURSE OF EVENTS:	ACTOR ACTION:	SYSTEM RESPONSE:						
		Manual Action	Automated Action					
	Step 1: The customer clicks on the 'Register' nav link in the navigation bar.		Step 2: The system retrieves the following attributes from the following tables, using a method in a .NET controller and an SQL READ query: From the Country Table					
			<table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Country</td><td>CountryName</td></tr></table>		Details to be read	Attributes in the table	Country	CountryName
			Details to be read	Attributes in the table				
Country	CountryName							
From the Province Table								

			<table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Province</td><td>ProvinceName</td></tr></table> From the City Table <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>City</td><td>CityName</td></tr></table> From the Suburb Table <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Suburb</td><td>SuburbName</td></tr></table> to populate the drop down lists and displays the Register Customer Screen which contains an MVC form with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Register</td><td>Heading</td><td>Main Heading</td></tr><tr><td>First Name</td><td>Label</td><td>Label for First Name textbox</td></tr><tr><td>First Name</td><td>Textbox</td><td>Used to capture the First Name attribute value of the customer</td></tr><tr><td>Surname</td><td>Label</td><td>Label for Surname textbox</td></tr><tr><td>Surname</td><td>Textbox</td><td>Used to capture the Surname attribute value of the customer</td></tr><tr><td>Email</td><td>Label</td><td>Label for Email textbox</td></tr><tr><td>Email</td><td>Textbox</td><td>Used to capture the Email Address attribute value of the customer</td></tr><tr><td>Phone Number</td><td>Label</td><td>Label for Phone Number textbox</td></tr><tr><td>Phone Number</td><td>Textbox</td><td>Used to capture the Phone Number attribute value of the customer</td></tr><tr><td>Street Address</td><td>Label</td><td>Label for Street Address textbox</td></tr><tr><td>Street Address</td><td>Textbox</td><td>Used to capture the Street Address</td></tr></table>	Details to be read	Attributes in the table	Province	ProvinceName	Details to be read	Attributes in the table	City	CityName	Details to be read	Attributes in the table	Suburb	SuburbName	Control Name	Control Type	Notes	Register	Heading	Main Heading	First Name	Label	Label for First Name textbox	First Name	Textbox	Used to capture the First Name attribute value of the customer	Surname	Label	Label for Surname textbox	Surname	Textbox	Used to capture the Surname attribute value of the customer	Email	Label	Label for Email textbox	Email	Textbox	Used to capture the Email Address attribute value of the customer	Phone Number	Label	Label for Phone Number textbox	Phone Number	Textbox	Used to capture the Phone Number attribute value of the customer	Street Address	Label	Label for Street Address textbox	Street Address	Textbox	Used to capture the Street Address
Details to be read	Attributes in the table																																																		
Province	ProvinceName																																																		
Details to be read	Attributes in the table																																																		
City	CityName																																																		
Details to be read	Attributes in the table																																																		
Suburb	SuburbName																																																		
Control Name	Control Type	Notes																																																	
Register	Heading	Main Heading																																																	
First Name	Label	Label for First Name textbox																																																	
First Name	Textbox	Used to capture the First Name attribute value of the customer																																																	
Surname	Label	Label for Surname textbox																																																	
Surname	Textbox	Used to capture the Surname attribute value of the customer																																																	
Email	Label	Label for Email textbox																																																	
Email	Textbox	Used to capture the Email Address attribute value of the customer																																																	
Phone Number	Label	Label for Phone Number textbox																																																	
Phone Number	Textbox	Used to capture the Phone Number attribute value of the customer																																																	
Street Address	Label	Label for Street Address textbox																																																	
Street Address	Textbox	Used to capture the Street Address																																																	

					attribute value of the customer
			Country	Label	Label for Country drop-down list
			Country	Drop-down list	Displays the name attribute value of all country instances retrieved from Country entity. Used to select the Country attribute value of the customer.
			Province	Label	Label for Province drop-down list
			Province	Drop-down list	Displays the name attribute value of all province instances retrieved from Province entity that are associated with the selected Country . Used to select the Province attribute value of the customer.
			City	Label	Label for Province drop-down list
			City	Drop-down list	Displays the name attribute value of all city instances retrieved from City entity that are associated with the selected Province . Used to select the City attribute value of the customer.
			Suburb	Label	Label for Suburb drop-down list
			Suburb	Drop-down list	Displays the name attribute value of all city instances retrieved from Suburb entity that are associated with the selected City via

					configured relationships. Used to select the Suburb attribute value of the customer.
			Password	Label	Label for the Password field.
			Password	Textbox	Used to enter customer password and apply necessary validations.
			Confirm Password	Label	Label for the Confirm Password field
			Confirm Password	Textbox	Must match the value specified in the Password textbox exactly.
			Create Account	Button	Triggers validation checks for the input captured on the form.
			Already have an account? Login	Hyperlink	Navigates to the Login screen
			Note: All controls are enabled by default, unless otherwise stated. All text input fields are empty by default, unless stated otherwise.		
	Step 3: The customer enters their details and clicks on the Create Account button [ALT]		Step 4: The system validates that all the required details have been provided in the correct format as per the validation requirements of the User table.		
			Details to be read	Attributes in the table	Notes
			First Name	FirstName	Alphabetical characters; All cases. Minimum: 1 Maximum: 50; Required
			Surname	Surname	Alphabetical characters; All cases. Minimum: 1 Maximum: 50; Required
			Email Address	EmailAddress	Required Unique Valid email domain
			Phone Number	PhoneNumber	Required Digits only Minimum: 10 Maximum: 10
			Street Address	StreetAddress	Required

					Alphanumeric characters only Minimum: 1 Maximum: 50	
			Suburb	Suburb_Id	Required FK from Suburb table	
			Password	Password	At least one lowercase and uppercase alphabetical character; one digit and one non-alphanumeric character. Minimum: 6 Maximum: 100; Required	
				[ALT]		
				Step 5: The system uses an SQL READ query to check if the user already exists in the User table using the following attribute:		
				Details to be read	Attributes in the table	
				Email Address	EmailAddress	
				[ALT]		
				Step 6: The system saves the details in the User table using a method in a .NET controller and an SQL INSERT query:		
				Details to be saved	Attributes in the table	Notes
Id				[User_ID]	A unique key generated by the system to uniquely identify the user by.	
First Name				[FirstName]	Required; Alphabetical characters only; Minimum: 1; Maximum: 50	
Surname				[Surname]	Required; Alphabetical characters only; Minimum: 1; Maximum: 50	
Email Address				[EmailAddress]	Required; Valid Email domain format. Maximum: 255; Minimum: 1	
Phone Number				[PhoneNumber]	Required; Digits only; Minimum: 10; Maximum: 10	

			Street Address	[StreetAddress]	Required; Alphanumeric characters only; Minimum: 1; Maximum: 50																
			Email Confirmed	[EmailConfirmed]	Boolean; set to 0 by default to represent email has not yet been confirmed.																
			Password	Password	The password provided is hashed before being stored.																
			Suburb Id	[SuburbId]	Required; valid suburb name selected from options provided. SuburbID attribute value is stored.																
			Role Id	[RoleId]	A unique key retrieved where [Role_Name] is 'Customer' in the UserRole table																
			Step 7: The system generates an email confirmation with the following details:																		
			<table><tr><th>Details</th><th>Notes</th></tr><tr><td>To</td><td>The users email address – retrieved from the User Table using an SQL READ query</td></tr><tr><td>From</td><td>The company email address, saved as a variable on the system</td></tr><tr><td>Subject</td><td>"Confirm your Natuurlik Account"</td></tr><tr><td>Email Template</td><td>Retrieved from the wwwroot folder</td></tr><tr><td>Body – Main Text</td><td>"Welcome to Natuurlik!" Click the button below to verify your account.</td></tr><tr><td>Button</td><td>The "Verify Account" button confirms the user's email address and directs them to the Account Confirmation Success screen</td></tr><tr><td>Body – Main Text</td><td>Your temporary password is: ***** Please reset your password once your account has been confirmed.</td></tr></table>			Details	Notes	To	The users email address – retrieved from the User Table using an SQL READ query	From	The company email address, saved as a variable on the system	Subject	"Confirm your Natuurlik Account"	Email Template	Retrieved from the wwwroot folder	Body – Main Text	"Welcome to Natuurlik!" Click the button below to verify your account.	Button	The "Verify Account" button confirms the user's email address and directs them to the Account Confirmation Success screen	Body – Main Text	Your temporary password is: ***** Please reset your password once your account has been confirmed.
	Details	Notes																			
	To	The users email address – retrieved from the User Table using an SQL READ query																			
	From	The company email address, saved as a variable on the system																			
Subject	"Confirm your Natuurlik Account"																				
Email Template	Retrieved from the wwwroot folder																				
Body – Main Text	"Welcome to Natuurlik!" Click the button below to verify your account.																				
Button	The "Verify Account" button confirms the user's email address and directs them to the Account Confirmation Success screen																				
Body – Main Text	Your temporary password is: ***** Please reset your password once your account has been confirmed.																				
		Step 8: The system sends the confirmation email to the user using the Sendgrid API																			
		Step 9: The system navigates the customer to the ' Register Confirmation ' screen which has the following controls:																			
		<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Register Confirmation</td><td>Heading</td><td>Page heading</td></tr><tr><td>Please check your email inbox to confirm your account.</td><td>Label</td><td>This control will be used to inform the user that the email confirmation link has been sent to their email address</td></tr></table>			Control Name	Control Type	Notes	Register Confirmation	Heading	Page heading	Please check your email inbox to confirm your account.	Label	This control will be used to inform the user that the email confirmation link has been sent to their email address								
Control Name	Control Type	Notes																			
Register Confirmation	Heading	Page heading																			
Please check your email inbox to confirm your account.	Label	This control will be used to inform the user that the email confirmation link has been sent to their email address																			

	Step 10: The user clicks on the link sent to their email address		Step 11: The system uses an SQL UPDATE and a Linq Query to update the customer's account to a confirmed state in the User table: <table><tr><th>Attribute in the table</th><th>Notes</th></tr><tr><td>[EmailConfirmed]</td><td>The value is updated to '1' to indicate that the user's account has been confirmed and can now be logged into.</td></tr></table>	Attribute in the table	Notes	[EmailConfirmed]	The value is updated to '1' to indicate that the user's account has been confirmed and can now be logged into.														
	Attribute in the table	Notes																			
[EmailConfirmed]	The value is updated to '1' to indicate that the user's account has been confirmed and can now be logged into.																				
			Step 12: The user is redirected to the ByteXecom website and the system displays the ' Account Confirmation Success ' page with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Account Confirmation Success</td><td>Label</td><td>Used as the main heading of the page.</td></tr><tr><td>Your email address has been confirmed.</td><td>Label</td><td>Informs the user that their email has been confirmed successfully.</td></tr><tr><td>Click here to log in</td><td>Hyperlink</td><td>Navigates the user to the "Login" screen if clicked.</td></tr></table>	Control Name	Control Type	Notes	Account Confirmation Success	Label	Used as the main heading of the page.	Your email address has been confirmed.	Label	Informs the user that their email has been confirmed successfully.	Click here to log in	Hyperlink	Navigates the user to the " Login " screen if clicked.						
Control Name	Control Type	Notes																			
Account Confirmation Success	Label	Used as the main heading of the page.																			
Your email address has been confirmed.	Label	Informs the user that their email has been confirmed successfully.																			
Click here to log in	Hyperlink	Navigates the user to the " Login " screen if clicked.																			
ALTERNATE COURSES	ALT Step 3: The user has an account on the system and selects the Login hyperlink: - The system invokes Use Case 1.1 Login - The use case is terminated																				
	ALT Step 4A: The system fails to validate the registration information captured since required fields were not provided with values: Blank Fields Validation Error(s): <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>The First Name field is required.</td><td>Label</td><td>This label is displayed when the First Name field is left blank upon requesting to register the user.</td></tr><tr><td>The Surname field is required.</td><td>Label</td><td>This label is displayed when the Surname field is left blank upon requesting to register the user.</td></tr><tr><td>The Email field is required.</td><td>Label</td><td>This label is displayed when the Email field is left blank upon requesting to register the user.</td></tr><tr><td>The Phone Number field is required.</td><td>Label</td><td>This label is displayed when the Phone Number field is left blank upon requesting to register the user.</td></tr><tr><td>The Street Address field is required.</td><td>Label</td><td>This label is displayed when the Street Address field is left blank upon requesting to register the user.</td></tr></table>			Control Name	Control Name	Notes	The First Name field is required.	Label	This label is displayed when the First Name field is left blank upon requesting to register the user.	The Surname field is required.	Label	This label is displayed when the Surname field is left blank upon requesting to register the user.	The Email field is required.	Label	This label is displayed when the Email field is left blank upon requesting to register the user.	The Phone Number field is required.	Label	This label is displayed when the Phone Number field is left blank upon requesting to register the user.	The Street Address field is required.	Label	This label is displayed when the Street Address field is left blank upon requesting to register the user.
	Control Name	Control Name	Notes																		
The First Name field is required.	Label	This label is displayed when the First Name field is left blank upon requesting to register the user.																			
The Surname field is required.	Label	This label is displayed when the Surname field is left blank upon requesting to register the user.																			
The Email field is required.	Label	This label is displayed when the Email field is left blank upon requesting to register the user.																			
The Phone Number field is required.	Label	This label is displayed when the Phone Number field is left blank upon requesting to register the user.																			
The Street Address field is required.	Label	This label is displayed when the Street Address field is left blank upon requesting to register the user.																			

	The Country field is required.	Label	This label is displayed when the Country field is left blank upon requesting to register the user.
	The Province field is required.	Label	This label is displayed when the Province field is left blank upon requesting to register the user.
	The City field is required.	Label	This label is displayed when the City field is left blank upon requesting to register the user.
	The Suburb field is required.	Label	This label is displayed when the Suburb field is left blank upon requesting to register the user.
	The Password field is required.	Label	This label is displayed when the Password field is left blank upon requesting to register the user.
Return to Step 3			
ALT Step 4B: The system fails to validate the registration information captured since incorrect details were entered or which was not in the required format:			
Invalid input validation per attribute:			
	Control Name	Control Name	Notes
	Invalid First Name provided. No digits or special characters are allowed.	Label	Validation for [FirstName] attribute. Indicates valid format for First Name field. Constrained to typing 50 characters only.
	Invalid Surname provided. No digits or special characters are allowed.	Label	Validation for [Surname] attribute. Indicates valid format for Surname field. Constrained to typing 50 characters only.
	The Email field does not contain a valid e-mail address.	Label	Validation for [Email] attribute. Requires a valid email format. Constrained to 255 characters.
	Please enter a valid Phone Number .	Label	Validation for [PhoneNumber] attribute. Requires a valid phone number format. Constrained to typing 10 digits only.
	Please enter a valid Street Number .	Label	Validation for [StreetAddress] attribute. Requires a valid street address format. Constrained to typing 50 characters. Alphanumerical characters supported.

	Confirm Password must match Password	Label	Displayed for the Confirm Password field if it does not exactly match the value in the Password field.						
	Passwords must have at least one non-alphanumeric character	Label	Displayed when a Password contains only digits and alphabetical characters.						
	Passwords must have at least one uppercase ('A'-'Z') character	Label	Displayed when a Password contains no uppercase characters.						
	Passwords must have at least one lowercase ('a'-'z') character	Label	Displayed when a Password contains no lowercase characters.						
	The Password must be at least 6 characters long.	Label	Displayed when a Password contains less than 6 characters.						
	Return to Step 3								
	ALT Step 5: The email address provided already exists in the User table.								
	The system displays an error message with the following controls:								
	<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Username '[Email]' already exists.</td><td>Label</td><td>This control is used to inform the administrator that the email address is already used by another user account.</td></tr></table>			Control Name	Control Name	Notes	Username '[Email]' already exists.	Label	This control is used to inform the administrator that the email address is already used by another user account.
	Control Name	Control Name	Notes						
Username '[Email]' already exists.	Label	This control is used to inform the administrator that the email address is already used by another user account.							
Terminate Use Case									
CONCLUSION:	The system displays a confirmation message to inform the customer that their account has been successfully created on the system.								
POST-CONDITION:	- A new account is created in the User table. - The customer is logged into their newly created account upon successfully registering on the system.								
BUSINESS RULES:	- The customer account's password has to be encrypted for security purposes. - The email address of the customer is to be used as the username of their account for unique identification of the registered user. - The customer role is automatically to be assigned where the role is not explicitly defined.								
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	- The customer should be prompted to re-enter the password they wish to set for their account. - All passwords should have at least one lowercase and uppercase alphabetical character; one digit and a non-alphanumeric character.								
ASSUMPTIONS:	None								
OPEN ISSUES:	None								

Table 7 - 1.7 Update Profile Details

USER MANAGEMENT SUBSYSTEM				
AUTHORS (s): Ofhani Mungani			DATE: 18/04/2022	
VERSION: 1.2			LAST REVIEW DATE: 20/08/2022	
USE CASE NAME:	Update Profile Details		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒	
USE CASE ID:	1.7			
PRIORITY:	High			
SOURCE:	ByteXpress Requirements List			
PRIMARY BUSINESS ACTOR	• Customer			
PRIMARY SYSTEM ACTOR	• None			
OTHER PARTICIPATING ACTORS:	• None			
OTHER INTERESTED STAKEHOLDERS:	• None			
DESCRIPTION:	This use case describes the event where a customer wishes to edit their profile details. The system will retrieve the customer information stored on the system which can be edited. The customer then edits their details accordingly. The use case ends when the customer's information has been successfully updated on the system.			
PRE-CONDITION:	• The customer must be logged in			
TRIGGER:	This use case is called when the customer wishes to edit their details			
TYPICAL COURSE OF EVENTS:	Actor Action	System Response		
		Manual Action	Automated Action	
	Step 1: The customer wishes to edit their details		Step 2: The system invokes use case 1.3 View Profile Details	
	Step 3: The customer edits their details and clicks on the Save button		Step 4: The system captures and validates the profile details against the following tables:	
		Details to be saved	Attributes in the table	Notes
		First Name	FirstName	Required Alphabetical characters only Minimum: 1 Maximum: 50

			Surname	Surname	Required Alphabetical characters only Minimum: 1 Maximum: 50
			Phone Number	PhoneNumber	Required Digits only Minimum: 10 Maximum: 10
			Street Address	StreetAddress	Required Alphanumeric characters only Minimum: 1 Maximum: 50
			Suburb ID	Suburb_ID	Required FK from Suburb table
			[ALT]		
			Step 5: The system saves the updated details into the User table using a method in a .NET controller and an SQL INSERT query:		
			Details to be saved	Attributes in the table	Notes
			First Name	FirstName	Required Alphabetical characters only Minimum: 1 Maximum: 50
			Surname	Surname	Required Alphabetical characters only Minimum: 1 Maximum: 50
			Phone Number	PhoneNumber	Required Digits only Minimum: 10 Maximum: 10
			Street Address	StreetAddress	Required Alphanumeric characters only Minimum: 1 Maximum: 50
			Suburb ID	Suburb_ID	Required FK from Suburb table
			Step 6: The system displays a bootstrap alert notification with the following controls:		
			Control Name	Control Name	Notes
			"Your profile has been updated successfully"	Label	This control is used to indicate a successful update

ALTERNATE COURSES:

ALT Step 4: The details entered are not in the correct format or the required fields are empty

- The system displays a validation error with the following controls

Control Name	Control Name	Notes
First Name	Textbox	This control will be outline in red to show a validation error
The First Name field is required or "Invalid First Name provided"	Label	This control will be used as a label to indicate a validation error in the First Name textbox
Surname	Textbox	This control will be outline in red to show a validation error
The Surname field is required or "Invalid Surname provided"	Label	This control will be used as a label to indicate a validation error in the Surname textbox
Phone Number	Textbox	This control will be outline in red to show a validation error
The Phone Number field is required or "Invalid Phone Number provided"	Label	This control will be used as a label to indicate a validation error in the Phone Number textbox
Street Address	Textbox	This control will be outline in red to show a validation error
The Street Address field is required or "Invalid Street Address provided"	Label	This control will be used as a label to indicate a validation error in the Street Address textbox
Country	Dropdown list	This control will be outline in red to show a validation error
The Country field is required	Label	This control will be used as a label to indicate a validation error in the Country dropdown
Province	Dropdown list	This control will be outline in red to show a validation error
The Province field is required	Label	This control will be used as a label to indicate a validation error in the Province dropdown
City	Dropdown list	This control will be outline in red to show a validation error
The City field is required	Label	This control will be used as a label to indicate a validation error in the City dropdown
Suburb	Dropdown list	This control will be outline in red to show a validation error
The Suburb field is required	Label	This control will be used as a label to indicate a validation error in the Suburb dropdown

- Return to step 3

CONCLUSION:	This use case concludes when the user details have successfully been updated on the system
POST-CONDITION:	The user details are updated in the User table and displayed on the screen
BUSINESS RULES	None
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• The customer cannot update their details without an internet connection • All users have access to this function
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 8 - 1.8 Update Email Address

USER MANAGEMENT SUBSYSTEM										
AUTHORS (s): Nomusa Vumisa			DATE: 16/05/2022							
VERSION: 1.0			LAST REVIEW DATE: 01/09/2022							
USE CASE NAME:	Update Email Address		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒							
USE CASE ID:	1.8									
PRIORITY:	High									
SOURCE:	ByteXcom Requirements List									
PRIMARY BUSINESS ACTOR	• User									
PRIMARY SYSTEM ACTOR	• None									
OTHER PARTICIPATING ACTORS:	• None									
OTHER INTERESTED STAKEHOLDERS:	• None									
DESCRIPTION:	This use case describes the event where a user wishes to edit their email address. The system will retrieve the user information stored on the system. The user then edits their email address. The system will then send an email address verification email to the user. The use case ends when the user's email address has been successfully updated on the system and the user has been directed to the successful verification page.									
PRE-CONDITION:	• The user must be logged in									
TRIGGER:	This use case is called when the user wishes to update their email address.									
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE								
		MANUAL ACTION	AUTOMATED ACTION							
	Step 1: The user clicks on the Email tab on the user profile side navigation bar		Step 2: The system retrieves the user's email address from the User table, using a method in a .NET controller and an SQL READ query:							
			<table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Email</td><td>Email</td></tr></table>		Details to be read	Attributes in the table	Email	Email		
			Details to be read	Attributes in the table						
			Email	Email						
and displays it on the Manage Email Screen which contains an MVC form with the following controls:										
<table><tr><td>Control Name</td><td>Control Type</td><td>Notes</td></tr><tr><td>Manage Email</td><td>Heading</td><td>Page heading</td></tr><tr><td>Email</td><td>Label</td><td>This control will be used as a label to</td></tr></table>		Control Name	Control Type	Notes	Manage Email	Heading	Page heading	Email	Label	This control will be used as a label to
Control Name	Control Type	Notes								
Manage Email	Heading	Page heading								
Email	Label	This control will be used as a label to								

					indicate the users email address attribute		
			Email	Textbox	This control will be used to display the users email address which has been retrieved from the user table		
			New email	Label	This control will be used as a label to indicate the users new email address attribute		
			New email	Textbox	This control will be used by the user to enter their new email address		
			Change email	Button	This control will be used to confirm the updating of the users email address		
			Note: The “Email” textbox is disabled while the “New email” textbox is enabled.				
	Step 3: The user enters their new email address and clicks on the change email button		Step 4: The system captures and validates the entered details against the validation requirements of the User table: [ALT]				
			<table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Email</td><td>Email</td><td>Required Unique Valid email domain</td></tr></table>	Details to be read	Attributes in the table	Notes	Email
	Details to be read	Attributes in the table	Notes				
	Email	Email	Required Unique Valid email domain				
		Step 5: The system uses an SQL READ query to check if the email already exists in the User table using the following attribute. [ALT]					
		<table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Email Address</td><td>EmailAddress</td></tr></table>	Details to be read	Attributes in the table	Email Address	EmailAddress	
Details to be read	Attributes in the table						
Email Address	EmailAddress						
		Step 6: The system generates a new call back URI and email with the following details:					

			<table><tr><th>Details</th><th>Attributes</th></tr><tr><td>URL</td><td>The Confirm Email Change page that the user will be directed to after accessing the link</td></tr><tr><td>Token</td><td>The change email token generated in a .NET controller</td></tr><tr><td>To</td><td>The users email address – retrieved from the User Table using an SQL read query</td></tr><tr><td>From</td><td>The company email address</td></tr><tr><td>Subject</td><td>"Confirm your email"</td></tr><tr><td>Message</td><td>"Please confirm your account by clicking here"</td></tr><tr><td>URI</td><td>Embedded in the "clicking here" hyperlink</td></tr></table>	Details	Attributes	URL	The Confirm Email Change page that the user will be directed to after accessing the link	Token	The change email token generated in a .NET controller	To	The users email address – retrieved from the User Table using an SQL read query	From	The company email address	Subject	"Confirm your email"	Message	"Please confirm your account by clicking here"	URI	Embedded in the "clicking here" hyperlink
	Details	Attributes																	
	URL	The Confirm Email Change page that the user will be directed to after accessing the link																	
	Token	The change email token generated in a .NET controller																	
	To	The users email address – retrieved from the User Table using an SQL read query																	
	From	The company email address																	
	Subject	"Confirm your email"																	
	Message	"Please confirm your account by clicking here"																	
	URI	Embedded in the "clicking here" hyperlink																	
			Step 7: The system sends the confirm email to the user using the Sendgrid API.																
		Step 8: The system displays a bootstrap alert notification with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"Confirmation link to change email sent. Please check your email."</td><td>Label</td><td>This control is used to indicate a successful creation</td></tr></table>	Control Name	Control Name	Notes	"Confirmation link to change email sent. Please check your email."	Label	This control is used to indicate a successful creation											
Control Name	Control Name	Notes																	
"Confirmation link to change email sent. Please check your email."	Label	This control is used to indicate a successful creation																	
Step 9: The user clicks on the link sent to their email address		Step 10: The system updates the entered details into the User table using a method in a .NET controller and an SQL UPDATE query: <table><tr><th>Details to be updated</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Email</td><td>Email</td><td>Required Unique Valid email domain</td></tr></table>	Details to be updated	Attributes in the table	Notes	Email	Email	Required Unique Valid email domain											
Details to be updated	Attributes in the table	Notes																	
Email	Email	Required Unique Valid email domain																	
		Step 11: The system directs the user to the Confirm Email Change screen which has the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Confirm Email Change</td><td>Heading</td><td>Page heading</td></tr><tr><td>"Thank you for confirming your email change."</td><td>Label</td><td>This control will be used to inform the user that their email address has been verified</td></tr></table>	Control Name	Control Name	Notes	Confirm Email Change	Heading	Page heading	"Thank you for confirming your email change."	Label	This control will be used to inform the user that their email address has been verified								
Control Name	Control Name	Notes																	
Confirm Email Change	Heading	Page heading																	
"Thank you for confirming your email change."	Label	This control will be used to inform the user that their email address has been verified																	

	ALT Step 4A: The email entered is not in the correct format or the field is empty The system displays a validation error with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>The Email field is not a valid e-mail address</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the New Email textbox</td></tr><tr><td>The Email field is required</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the New Email textbox</td></tr></table>	Control Name	Control Name	Notes	The Email field is not a valid e-mail address	Label	This control will be used as a label to indicate a validation error in the New Email textbox	The Email field is required	Label	This control will be used as a label to indicate a validation error in the New Email textbox
	Control Name	Control Name	Notes							
	The Email field is not a valid e-mail address	Label	This control will be used as a label to indicate a validation error in the New Email textbox							
	The Email field is required	Label	This control will be used as a label to indicate a validation error in the New Email textbox							
	Return to step 3									
ALT Step 4B: The user clicks on the change email button without updating their email The system displays a bootstrap alert notification with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Your email is unchanged</td><td>Label</td><td>This control is used to inform the user that no changes have been made</td></tr></table>	Control Name	Control Name	Notes	Your email is unchanged	Label	This control is used to inform the user that no changes have been made				
Control Name	Control Name	Notes								
Your email is unchanged	Label	This control is used to inform the user that no changes have been made								
Return to step 3										
ALT Step 5: The email already exists The system displays a bootstrap alert notification with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>The email already exists</td><td>Label</td><td>This control is used to inform the user that the email already</td></tr></table>	Control Name	Control Name	Notes	The email already exists	Label	This control is used to inform the user that the email already				
Control Name	Control Name	Notes								
The email already exists	Label	This control is used to inform the user that the email already								
The use case is terminated										
CONCLUSION:	This use case concludes when the user’s email address has been successfully updated on the system									
POST-CONDITION:	The user’s email address is updated in the user table and displayed on the screen									
BUSINESS RULES	None									
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	- The user cannot update their email address without an internet connection - All users have access to this function									
ASSUMPTIONS:	None									
OPEN ISSUES:	None									

2.2. Administrative Subsystem

Table 9 - 2.1 Register User

2. ADMINISTRATIVE SUBSYSTEM					
AUTHORS (s): Kyle van Eeden			DATE: 03/05/2022		
VERSION: 1.2			LAST REVIEW DATE: 30/08/2022		
USE CASE NAME:	Register User		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒		
USE CASE ID:	2.1				
PRIORITY:	High				
SOURCE:	ByteXecom Requirements List				
PRIMARY BUSINESS ACTOR	Administrator				
PRIMARY SYSTEM ACTOR	None				
OTHER PARTICIPATING ACTORS:	User (ESA)				
OTHER INTERESTED STAKEHOLDERS:	Staff and Resellers: All of Natuurlik's staff and resellers should be informed of when their account has been registered and provided with instructions of how they can log into their newly registered account.				
DESCRIPTION:	This use case describes the event in which the administrator wishes to add a new user to the system with a predefined role assigned which controls the access permissions the user account will have on the system. The use case begins with the administrator selecting the option to register a new user on the system, after which the system will prompt the administrator to enter all the required details needed to register a new user account. The system will inform the administrator when the account has been successfully created. An account registration confirmation email will also be sent to the user's email address, and after confirming the account by clicking on the link in the email, the user will be redirected to the Bytexecom website and the user's account will be updated to a confirmed state. The use case concludes when the new user is created in the User entity and the user's account has been verified so they can log into the account.				
PRE-CONDITION:	- The administrator must be logged onto the system. - The user to be registered must not already exist on the system.				
TRIGGER:	The use case is triggered when the administrator wishes to add a new user onto the system.				
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE			
		MANUAL ACTION	AUTOMATED ACTION		
			Step 2: The system displays the 'Manage' screen with the following controls: <table><tr><td>Control Name</td><td>Control Type</td><td>Notes</td></tr></table>		
Control Name	Control Type	Notes			
	Step 1: The administrator wishes to register a new user on the system and clicks on the 'Manage Store'				

	hyperlink in the navigation bar.		Manage Menu	Side Navigation Menu	Displays all of the functions the administrator has access to.
			Store	Hyperlink	Navigates the administrator to the Home screen if clicked.
			Products	Hyperlink	Displays hyperlinks for all product-related functionality if collapsed.
			Orders	Hyperlink	Displays hyperlinks for all order-related functionality if collapsed.
			Suppliers	Hyperlink	Displays hyperlinks for all supplier-related functionality if collapsed.
			Inventory Management	Hyperlink	Displays hyperlinks for all inventory-related functionality if collapsed.
			Locations	Hyperlinks	Displays hyperlinks for all location-related functionality if collapsed.
			Reports	Hyperlink	Displays hyperlinks for all reporting-related functionality if collapsed.
			Users	Hyperlink	Displays hyperlinks for all user management-related functionality if collapsed.
			Manage VAT	Hyperlink	Displays hyperlinks for all VAT-related functionality if collapsed.
			Help	Hyperlink	Opens a Help document in PDF format.
	Step 3: The administrator clicks on		Step 4: The system uses a .NET Controller with a SQL Read query to retrieve the		

the the **Users** hyperlink button.

following details for all existing user accounts retrieved from the **User** entity:

Details to be read	Attributes in the table
First Name	[FirstName]
Surname	[Surname]
Email Address	[EmailAddress]
Phone Number	[PhoneNumber]

And displays the **Users** screen, populated with the retrieved details for all existing users, and with the following controls:

Control Name	Control Type	Notes
Users	Heading	Page main heading
Add User	Button	When clicked, this control will direct the administrator to the ' Register User ' screen
Search bar	Textbox	This control will be used when the administrator inserts the search parameter to base a search on.
Users table	Table	This control will be used to display specific details of all the user accounts that exist on the system OR for user accounts that match the search parameter specified if applicable.
First Name	Label	This control will be used as a heading to display the [FirstName] attribute value for all users retrieved
Surname	Label	This control will be used as a heading to display the

					[Surname] attribute value for users retrieved
			Email	Label	This control will be used as a heading to display the [EmailAddresses] attribute value for all users retrieved
			Phone Number	Label	This control will be used as a heading to display the [PhoneNumber] attribute value for all users retrieved
			Update Icon	Button	When clicked, this control will direct the administrator to the 'Update User' page for a specific user account selected.
			Delete Icon	Button	When clicked, this control will direct the administrator to the 'Delete User' screen for the specific user account selected.
			Page count	Label	This control will be used as a label to display the number of pages that the results make up and to indicate on which page the user is currently on.
			Previous	Button	This control is used to navigate to the next page of user results.
			Next	Button	This control is used to navigate to the previous page of user results.

Step 5: The system administrator clicks on the **Add User** button.

Step 6: The system retrieves the following attribute values from their respective entities using an Entity Framework SQL Read Query:

All countries retrieved from the **Country** entity:

Details to be read	Attributes in the table
Country	CountryName

All provinces associated with selected country from the **Province** entity:

Details to be read	Attributes in the table
Province	ProvinceName

All cities associated with the selected province From the **City** entity:

Details to be read	Attributes in the table
City	CityName

All suburbs associated with the selected city From the **Suburb** entity:

Details to be read	Attributes in the table
Suburb	SuburbName

All roles From the **Role** entity:

Details to be read	Attributes in the table
Role	RoleName

The system displays the **Register User screen** which contains an MVC form with the following controls:

Control Name	Control Type	Notes
Register User	Heading	Main page heading
User Information	Heading	Page subheading
Role	Label	Displayed as a label for the Role drop-down list control.
Role	Drop-down list	Selectable options populated with all [RoleName] attribute values retrieved from the Role entity.

			First Name	Label	This control will be used as a label for the First Name textbox
			First Name	Textbox	This control will be used by the administrator to insert the [FirstName] attribute value of the user
			Surname	Label	This control will be used as a label for the Surname textbox
			Surname	Textbox	This control will be used by the administrator to insert the [Surname] attribute value of the user
			Email	Label	This control will be used as a label for the Email textbox
			Email	Textbox	This control will be used by the administrator to insert the [EmailAddresses] attribute value of the user
			Phone Number	Label	This control will be used as a label for the Phone Number textbox
			Phone Number	Textbox	This control will be used by the administrator to insert the [PhoneNumber] attribute value of the user
			Street Address	Label	This control will be used as a label for the Street Address textbox
			Street Address	Textbox	This control will be used by the administrator to insert the

					[StreetAddress] attribute value of the user
			Country	Label	This control will be used as a label for the Country drop-down list
			Country	Drop-down list	All countries are retrieved from the Country entity and the [CountryName] attribute value is displayed as a selectable option for each country retrieved. This control will be used by the administrator to select the [CountryID] attribute value to be assigned to the user.
			Province	Label	This control will be used as a label for the Province drop-down list.
			Province	Drop list	Control's selectable options are populated with the [ProvinceName] attribute value of all provinces retrieved from the Province entity and which has a foreign key relationship with the selected [CountryID] in the Country control. This control will be used by the administrator to select the [ProvinceID]

					attribute value to be assigned to the user.
			City	Label	This control will be used as a label for the City drop-down list.
			City	Drop list	Control's options are populated with the [CityName] attribute values of all cities retrieved from the City entity and which has a foreign key relationship with the selected [ProvinceID] in the Province control. This control will be used by the administrator to select the [CityID] attribute value to be assigned to the user.
			Suburb	Label	This control will be used as a label for the Suburb drop-down list.
			Suburb	Drop list	Control's options are populated with the [SuburbName] attribute values of all provinces retrieved from the Suburb entity and which has a foreign key relationship with the selected [CityID] in the City control.

			This control will be used by the administrator to select the [SuburbID] attribute value to be assigned to the user.						
	Password	Label	Displayed as a label for the Password textbox control.						
	Password	Textbox	User account's password to be used for authentication purposes.						
	Confirm Password	Label	Displayed as a label for the Confirm Password textbox control.						
	Confirm Password	Textbox	A confirmation of the Password value which has been entered.						
	Create Account	Button	This control will be used to confirm the registration of the new user account						
	Cancel	Button	This control will be used to cancel the user registration operation						
	Note: All controls are enabled by default, unless otherwise stated. All text input fields are empty by default, unless stated otherwise.								
Step 7: The system administrator enters the requested user details and clicks on the Create Account button. [ALT]		Step 8: The system captures and validates the entered details against the validation requirements of the following tables: From the User entity: <table><tr><th>Details to be read</th><th>Attribute in the table</th><th>Validation Rules</th></tr><tr><td>First Name</td><td>[FirstName]</td><td>Required; Alphabetical characters only; Minimum: 1; Maximum: 50</td></tr></table>		Details to be read	Attribute in the table	Validation Rules	First Name	[FirstName]	Required; Alphabetical characters only; Minimum: 1; Maximum: 50
Details to be read	Attribute in the table	Validation Rules							
First Name	[FirstName]	Required; Alphabetical characters only; Minimum: 1; Maximum: 50							

			Surname	[Surname]	Required; Alphabetical characters only; Minimum: 1; Maximum: 50
			Email	[Email]	Required; Unique; Valid email domain; Unique
			Phone Number	[PhoneNum ber]	Required; Digits only; Minimum: 10; Maximum: 10
			Street Address	[StreetAddre ss]	Required; Alphanumeric characters only; Minimum: 1; Maximum: 50
			Country	[CountryNa me]	Required; Valid [CountryID] attribute value referenced from the Country entity via a foreign key association
			Province	[ProvinceNa me]	Required; Valid [ProvinceID] attribute value referenced from the Province entity via a foreign key association
			Suburb	[SuburbNam e]	Required; Valid [SuburbID] attribute value referenced from the Suburb entity via a foreign key association.
			User Role	[RoleID]	Required; Valid [RoleID] attribute value referenced from the Role entity via a foreign key association.
			Password	N/A	Must contain at least one uppercase and lowercase character, one digit and a

					non-alphanumeric character; Minimum: 6; Maximum: 25
			Confirm Password	N/A	Must exactly match the password provided.
			From the Country Table:		
			Details to be read	Attributes in the table	Notes
			Country	Country_ID	Required
			From the Province Table:		
			Details to be read	Attributes in the table	Notes
			Province	Province_ID	Required
			From the City Table:		
			Details to be read	Attributes in the table	Notes
			City	City_ID	Required
			From the Suburb Table:		
			Details to be read	Attributes in the table	Notes
			Suburb	Suburb_ID	Required
			[ALT]		
			Step 9: The system uses a .NET controller with a SQL INSERT query to add the following information into the User entity:		
			Details to be saved	Attributes in the table	Notes
			First Name	[FirstName]	Stores the [FirstName] attribute value captured in the First Name control.
			Surname	[Surname]	Stores the [Surname] attribute value captured in the Surname control.
			Email	[Email]	Stores the [Email] attribute value captured in the Email control.
			Phone Number	[PhoneNumber]	Stores the [PhoneNum

			er] attribute value captured in the Phone Number control.
	Street Address	[StreetAddress]	Stores the [StreetAddress] attribute value captured in the Street Address control.
	Country	[CountryID]	Foreign Key referenced from the Country entity.
	Province	[ProvinceID]	Foreign Key referenced from the Province entity.
	City	[CityID]	Foreign Key referenced from the City entity.
	Suburb	[SuburbID]	Foreign Key referenced from the Suburb entity.
	Email Confirmed	[EmailConfirmed]	Boolean; set to 0 by default upon account registration to represent email has not yet been confirmed.
	Password	[PasswordHash]	The password provided is encrypted via a hashing algorithm before being stored.

The system uses a SQL Insert Query to insert the following information into the **UserRole** table:

Details to be saved	Attributes in the table	Notes
User Identifier	[User_ID]	A unique key referenced from the User entity for the specific user being registered.
Role Identifier	[Role_ID]	A unique key referenced from the Role

					entity based on role assigned to user.												
			Step 10: The system uses the SendGrid email service to send a registration confirmation email to the email address of the created user using the [EmailAddress] attribute value specified for the registered user and which uses the following controls to populate the email: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Natuurlik</td><td>Label</td><td>Indicates where the Email was sent from.</td></tr><tr><td>Confirm your email</td><td>Label</td><td>Control is used for the subject of the email sent.</td></tr><tr><td>"Welcome to Natuurlik! Your account has been successfully registered. <u>Please confirm your account by clicking here.</u>"</td><td>Hyperlink</td><td>Navigates the user to the 'Account Confirmation Success' screen.</td></tr></table>			Control Name	Control Type	Notes	Natuurlik	Label	Indicates where the Email was sent from.	Confirm your email	Label	Control is used for the subject of the email sent.	"Welcome to Natuurlik! Your account has been successfully registered. <u>Please confirm your account by clicking here.</u> "	Hyperlink	Navigates the user to the 'Account Confirmation Success' screen.
	Control Name	Control Type	Notes														
Natuurlik	Label	Indicates where the Email was sent from.															
Confirm your email	Label	Control is used for the subject of the email sent.															
"Welcome to Natuurlik! Your account has been successfully registered. <u>Please confirm your account by clicking here.</u> "	Hyperlink	Navigates the user to the 'Account Confirmation Success' screen.															
		Step 11: The system navigates the administrator to the Users screen and confirms that the user's account details has been created in the User table by displaying the following message box notification: <table><tr><th>Control Name</th><th>Control Type</th></tr><tr><td>User Account Created Successfully</td><td>Label</td></tr></table>			Control Name	Control Type	User Account Created Successfully	Label									
Control Name	Control Type																
User Account Created Successfully	Label																
	Step 12: The registered user selects the option to verify their account via email by clicking on the hyperlink provided.		Step 13: The system uses a .NET controller with an Entity Framework SQL UPDATE query to update the user's account to a confirmed state in User table:														

			<table><tr><th>Attribute in the table</th><th>Notes</th></tr><tr><td>[EmailConfirmed]</td><td>The value is updated to the value '1' in order to indicate that the user's account has been confirmed and can now be logged into.</td></tr></table>	Attribute in the table	Notes	[EmailConfirmed]	The value is updated to the value '1' in order to indicate that the user's account has been confirmed and can now be logged into.											
	Attribute in the table	Notes																
[EmailConfirmed]	The value is updated to the value '1' in order to indicate that the user's account has been confirmed and can now be logged into.																	
			Step 14: The user is redirected to the ByteXecom website and the system displays the 'Account Confirmation Success' page with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Successful Email Confirmation</td><td>Label</td><td>Used as the main heading of the page.</td></tr><tr><td>Your email address has been confirmed.</td><td>Label</td><td>Informs the user that their account and email has been confirmed successfully.</td></tr><tr><td>Click here to log in</td><td>Hyperlink</td><td>Navigates the user to the Log In screen if clicked.</td></tr></table>	Control Name	Control Type	Notes	Successful Email Confirmation	Label	Used as the main heading of the page.	Your email address has been confirmed.	Label	Informs the user that their account and email has been confirmed successfully.	Click here to log in	Hyperlink	Navigates the user to the Log In screen if clicked.			
Control Name	Control Type	Notes																
Successful Email Confirmation	Label	Used as the main heading of the page.																
Your email address has been confirmed.	Label	Informs the user that their account and email has been confirmed successfully.																
Click here to log in	Hyperlink	Navigates the user to the Log In screen if clicked.																
ALTERNATE COURSES:	ALT Step 7: The administrator does not wish to register a new user account anymore: - The administrator clicks on the 'Cancel' button - The use case is terminated																	
	ALT Step 8 a: The system fails to validate the registration information captured since required fields were not provided with values: Required field validation: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>First Name</td><td>Textbox</td><td>Border outlined with red colour if no input has been provided.</td></tr><tr><td>The First Name field is required.</td><td>Label</td><td>This label is displayed when the First Name field is left blank upon requesting to register the user.</td></tr><tr><td>Surname</td><td>Textbox</td><td>Border outlined with red colour if no input has been provided.</td></tr><tr><td>The Surname field is required.</td><td>Label</td><td>This label is displayed when the Surname field is left blank upon requesting to register the user.</td></tr></table>			Control Name	Control Type	Notes	First Name	Textbox	Border outlined with red colour if no input has been provided.	The First Name field is required.	Label	This label is displayed when the First Name field is left blank upon requesting to register the user.	Surname	Textbox	Border outlined with red colour if no input has been provided.	The Surname field is required.	Label	This label is displayed when the Surname field is left blank upon requesting to register the user.
	Control Name	Control Type	Notes															
	First Name	Textbox	Border outlined with red colour if no input has been provided.															
	The First Name field is required.	Label	This label is displayed when the First Name field is left blank upon requesting to register the user.															
Surname	Textbox	Border outlined with red colour if no input has been provided.																
The Surname field is required.	Label	This label is displayed when the Surname field is left blank upon requesting to register the user.																

	Email	Textbox	Border outlined with red colour if no input has been provided.
	The Email field is required.	Label	This label is displayed when the Email field is left blank upon requesting to register the user.
	Phone Number	Textbox	Border outlined with red colour if no input has been provided.
	The Phone Number field is required.	Label	This label is displayed when the Phone Number field is left blank upon requesting to register the user.
	Street Address	Textbox	Border outlined with red colour if no input has been provided.
	The Street Address field is required.	Label	This label is displayed when the Street Address field is left blank upon requesting to register the user.
	Country	Drop-down List	Border outlined with red colour if no input has been provided.
	The Country field is required.	Label	This label is displayed when the Country field is left blank upon requesting to register the user.
	Province	Drop-down List	Border outlined with red colour if no input has been provided.
	The Province field is required.	Label	This label is displayed when the Province field is left blank upon requesting to register the user.
	City	Drop-down List	Border outlined with red colour if no input has been provided.
	The City field is required.	Label	This label is displayed when the City field is left blank upon requesting to register the user.
	Suburb	Drop-down List	Border outlined with red colour if no input has been provided.
	The Suburb field is required.	Label	This label is displayed when the Suburb field is left blank upon requesting to register the user.
	Role	Drop-down List	Border outlined with red colour if no input has been provided.
	The Role field is required.	Label	This label is displayed when the Role field is left blank upon requesting to register the user.

	Password	Textbox	Border outlined with red colour if no input has been provided.
	The Password field is required.	Label	This label is displayed when the Password field is left blank upon requesting to register the user.
	Return to Step 7		
	ALT Step 8 b: The system fails to validate the registration information captured since incorrect details were entered or which was not in the required format:		
	Invalid input validation per attribute:		
	Control Name	Control Name	Notes
	First Name	Textbox	Field's border outline is the colour red if it contains invalid input.
	Invalid First Name provided. No digits or special characters are allowed.	Label	Validation for [FirstName] attribute. Indicates valid format for First Name field. Constrained to typing 50 characters only.
	Surname	Textbox	Field's border outline is the colour red if it contains invalid input.
	Invalid Surname provided. No digits or special characters are allowed.	Label	Validation for [Surname] attribute. Indicates valid format for Surname field. Constrained to typing 50 characters only.
	Email	Textbox	Field's border outline is the colour red if it contains invalid input.
	The Email field does not contain a valid e-mail address.	Label	Validation for [Email] attribute. Requires a valid email format. Constrained to 255 characters.
	Phone Number	Textbox	Field's border outline is the colour red if it contains invalid input.
	Please enter a valid Phone Number .	Label	Validation for [PhoneNumber] attribute. Requires a valid phone number format. Constrained to typing 10 digits only.
	Street Address.	Textbox	Field's border outline is the colour red if it contains invalid input.

	Please enter a valid Street Address .	Label	Validation for [StreetAddress] attribute. Requires a valid street address format. Constrained to typing 50 characters. Alphanumeric characters supported.					
	Confirm Password must match Password	Label	Displayed for the Confirm Password field if it does not exactly match the value in the Password field.					
	Passwords must have at least one non-alphanumeric character	Label	Displayed when a Password contains only digits and alphabetical characters.					
	Passwords must have at least one uppercase ('A'- 'Z') character	Label	Displayed when a Password contains no uppercase characters.					
	Passwords must have at least one lowercase ('a'- 'z') character	Label	Displayed when a Password contains no lowercase characters.					
	The Password must be at least 6 characters long.	Label	Displayed when a Password contains less than 6 characters.					
	Return to Step 7							
	ALT Step 8 c: The email address provided already exists in the User table.							
	The system displays an error message with the following controls:							
	<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>'[Email]' has already been taken.</td><td>Label</td><td>This control is used to inform the administrator that the email address is already used by another user account.</td></tr></table>			Control Name	Control Name	Notes	'[Email]' has already been taken.	Label
Control Name	Control Name	Notes						
'[Email]' has already been taken.	Label	This control is used to inform the administrator that the email address is already used by another user account.						
Terminate Use Case								
CONCLUSION:	This use case concludes when the user verifies their account via email.							
POST-CONDITION:	A new user account has been added to the User entity in the database. The user has confirmed their account and can log into the system.							
BUSINESS RULES	- Only the administrator has the necessary permissions to add a new user to the system with a specific pre-defined role assigned. - A user should be required to confirm their registered account via email. - All emails stored on the system should be unique per account registered. - A user should be required to confirm their account before being able to log into their registered account. - The administrator should be able to control what access a user will have on the system by assigning a relevant user role.							
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	- The system is offline, and the system administrator will not be able to add a new user to the system. - User roles to be assigned to users are pre-defined and access permissions cannot be amended.							

ASSUMPTIONS:	• Only valid South African numbers are to be registered on the system.
OPEN ISSUES:	• None

Table 10 - 2.2 Search User

2. ADMINISTRATIVE SUBSYSTEM					
AUTHORS (s): Kyle van Eeden		DATE: 03/05/2022			
VERSION: 1.1		LAST REVIEW DATE: 30/08/2022			
USE CASE NAME:	Search User	USE CASE TYPE			
USE CASE ID:	2.2	Business Requirements: ☐			
PRIORITY:	High	System Analysis: ☐			
SOURCE:	ByteXecom Requirements List	System Design: ☒			
PRIMARY BUSINESS ACTOR	Administrator				
PRIMARY SYSTEM ACTOR	None				
OTHER PARTICIPATING ACTORS:	None				
OTHER INTERESTED STAKEHOLDERS:	None				
DESCRIPTION:	This use case describes the event of an administrator searching an existing user defined on the system. The use case begins when the administrator navigates to the Users screen and enters the search criteria in the Search textbox control provided. The system will capture and process the search criteria specified and return all user records that matched the search criteria. The use case concludes once the Users table control is populated with all user accounts that matched the search criteria specified.				
PRE-CONDITION:	The administrator must be logged into the system.				
TRIGGER:	This use case is called when the administrator wishes to search for a user account on the system.				
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE			
		MANUAL ACTION	AUTOMATED ACTION		
	Step 1: The administrator selects the Users button in the side navigation menu.		Step 2: The system retrieves the following details for each user from the User entity using an .NET controller and Entity Framework SQL Read query and displays the Users screen with the following controls:		
			Control Name	Control Type	Notes
			All Users	Heading	Page main heading
Add	Button	When clicked, this control will direct the administrator to the Add User page			

			Search	Textbox	This control will be used when the administrator inserts the search parameters. Search is automatically performed as text is entered
			Show Entries Per Page	Label	Indicates that the administrator can use the control to specify amount of users to view per page
			Number of Entries	Drop-down list	This control will be used to select the amount of entries to display per page
			User	Table	Populated with the [FirstName], [Surname], [PhoneNumber] and [EmailAddress] attribute values for all user accounts retrieved from User entity.
			Update Icon	Button	When clicked, this control will redirect the administrator to the ' Update User ' screen.
			Delete Icon	Button	When clicked, this control will redirect the administrator to the ' Delete User ' screen.
			Page count	Button	This control will be used to navigate between different pages.
			Previous	Button	This control is used to navigate the administrator from the current page

					to the previous page																
			Next	Button	This control is used to navigate the administrator from the current page to the next page of results if any.																
			Show Entries	Label	Shows the number of entries on the current page out of total number of entries retrieved from the User entity.																
	Step 3: The administrator enters their search query into the ' Search ' textbox control.		Step 4: The system populates the User table control displayed with the search results that have matched the search criteria entered. The following details is displayed in the table control:	<table><tr><th>Control Name</th><th>Attribute retrieved from table</th><th>Notes</th></tr><tr><td>First Name</td><td>[FirstName]</td><td>Table heading displayed along with all [FirstName] attribute values that matched the search query.</td></tr><tr><td>Surname</td><td>[Surname]</td><td>Table heading displayed along with all [Surname] attribute values that matched the search query.</td></tr><tr><td>Phone Number</td><td>[PhoneNumber]</td><td>Table heading displayed along with all [PhoneNumber] attribute values that matched the search query.</td></tr><tr><td>Email</td><td>[Email]</td><td>Table heading displayed along with all [Email] attribute values that matched the search query.</td></tr></table>			Control Name	Attribute retrieved from table	Notes	First Name	[FirstName]	Table heading displayed along with all [FirstName] attribute values that matched the search query.	Surname	[Surname]	Table heading displayed along with all [Surname] attribute values that matched the search query.	Phone Number	[PhoneNumber]	Table heading displayed along with all [PhoneNumber] attribute values that matched the search query.	Email	[Email]	Table heading displayed along with all [Email] attribute values that matched the search query.
				Control Name	Attribute retrieved from table	Notes															
				First Name	[FirstName]	Table heading displayed along with all [FirstName] attribute values that matched the search query.															
				Surname	[Surname]	Table heading displayed along with all [Surname] attribute values that matched the search query.															
				Phone Number	[PhoneNumber]	Table heading displayed along with all [PhoneNumber] attribute values that matched the search query.															
				Email	[Email]	Table heading displayed along with all [Email] attribute values that matched the search query.															
				[ALT]																	
ALTERNATE COURSES:	ALT Step 4: The search query entered does not match any records existing in the User entity:																				

	- The system displays a “No Matching Records Found” label in User table. - The use case concludes
CONCLUSION:	This use case concludes when the user accounts which match the search query specified is displayed in the Users table control.
POST-CONDITION:	All matched users are displayed on the screen by populating the Users table control.
BUSINESS RULES	- Only the administrator should be authorized to use the search user functionality. - The administrator should be able to search for a registered user on the system by name, email address or phone number.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	- The search operation is executed immediately as the administrator specifies the search criteria in the Search textbox control. - The administrator can only use the First Name, Surname, Phone Number and Email attributes to search for a specific user on the system.
ASSUMPTIONS:	None
OPEN ISSUES:	None

Table 11 - 2.3 Update User

ADMINISTRATIVE SUBSYSTEM			
AUTHORS (s): Kyle van Eeden		DATE: 03/05/2022	
VERSION: 1.3		LAST REVIEW DATE: 30/05/2022	
USE CASE NAME:	Update User		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒
USE CASE ID:	2.3		
PRIORITY:	Medium		
SOURCE:	ByteXecom Requirements List		
PRIMARY BUSINESS ACTOR	• Administrator		
PRIMARY SYSTEM ACTOR	• None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	This use case describes the event in which the administrator wishes to update a user account's details on the system. The use case starts with the administrator selecting the 'Update' button for the specific user account they wish to update, and the system will display the current details retrieved for the specific user account requested from the User entity. After the entered user account details has been validated, it will be up updated in User entity. The use case concludes with the system displaying a message confirming that the user details has been successfully updated.		
PRE-CONDITION:	• The administrator must be logged into the system. • The user account must exist on the system before it can be updated.		
TRIGGER:	The use case is triggered when the administrator wishes to update the details stored on a specific user account.		
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE	
		MANUAL ACTION	AUTOMATED ACTION
	Step 1: The administrator logs onto the system and wishes to update an existing user's details on the system.		Step 2: The system invokes Use Case 2.2 "Search User" .
	Step 3: The administrator selects the Update button for the specific user account they wish to update.		Step 4: The system retrieves the following attributes for the requested user from the User entity using a .NET controller with an Entity Framework SQL Read query:
		Details to be read	Attributes in the table
		Name	[FirstName]
		Surname	[Surname]
		Email Address	[Email]

			<table><tr><td>Phone Number</td><td>[PhoneNumber]</td></tr><tr><td>Street Name</td><td>[StreetName]</td></tr><tr><td>Account Status</td><td>[EmailConfirmed]</td></tr></table> The name of the current country value is retrieved from the Country entity: <table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Country</td><td>CountryName</td></tr></table> The name of the current province value is retrieved from the Province entity: <table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Province</td><td>ProvinceName</td></tr></table> The name of the current city value is retrieved from the City entity: <table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>City</td><td>CityName</td></tr></table> The name of the current suburb value is retrieved from the Suburb entity: <table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Suburb</td><td>SuburbName</td></tr></table> The system displays the populated MVC form on Update User screen upon loading with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Update User Details</td><td>Heading</td><td>This control is used to display the heading of the Update User screen</td></tr><tr><td>First Name</td><td>Label</td><td>Displays the label text for the First Name textbox control.</td></tr><tr><td>First Name</td><td>Textbox</td><td>Populated with the [FirstName] attribute value</td></tr></table>	Phone Number	[PhoneNumber]	Street Name	[StreetName]	Account Status	[EmailConfirmed]	Details to be read	Attributes in the table	Country	CountryName	Details to be read	Attributes in the table	Province	ProvinceName	Details to be read	Attributes in the table	City	CityName	Details to be read	Attributes in the table	Suburb	SuburbName	Control Name	Control Type	Notes	Update User Details	Heading	This control is used to display the heading of the Update User screen	First Name	Label	Displays the label text for the First Name textbox control.	First Name	Textbox	Populated with the [FirstName] attribute value
Phone Number	[PhoneNumber]																																				
Street Name	[StreetName]																																				
Account Status	[EmailConfirmed]																																				
Details to be read	Attributes in the table																																				
Country	CountryName																																				
Details to be read	Attributes in the table																																				
Province	ProvinceName																																				
Details to be read	Attributes in the table																																				
City	CityName																																				
Details to be read	Attributes in the table																																				
Suburb	SuburbName																																				
Control Name	Control Type	Notes																																			
Update User Details	Heading	This control is used to display the heading of the Update User screen																																			
First Name	Label	Displays the label text for the First Name textbox control.																																			
First Name	Textbox	Populated with the [FirstName] attribute value																																			

					retrieved for the requested user.
	Surname	Label			Displays the label text for the Surname textbox control.
	Surname	Textbox			Populated with the [Surname] attribute value retrieved for the requested user.
	Email	Label			Displays the label text for the Email textbox control.
	Email	Textbox			Read-Only ; Populated with the [Email] attribute value retrieved for the requested user.
	Phone Number	Label			Displays the label text for the Phone Number textbox control.
	Phone Number	Textbox			Populated with the [PhoneNumber] attribute value retrieved for the requested user.
	Street Address	Label			Displays the label text for the Street Address textbox control.
	Street Address	Textbox			Populated with the [StreetAddress] attribute value retrieved for the requested user.
	Country	Label			Used to display a label for the Country drop-down list
	Country	Drop-down list			Populated with the current [CountryName] value

					retrieved for the user based on the currently assigned [CountryID] value referenced from the Country entity. All countries are also retrieved from the Country entity to populate selectable options in control.
	Province	Label			Used to display a label for the Province drop-down list.
	Province	Drop-down list			Populated with the [ProvinceName] attribute value retrieved for the user using the currently stored foreign key referenced from the Province entity. Control options is populated with all city names retrieved from the Province entity where they have a foreign key relationship established with the selected [CountryID] value.
	City	Label			Used to display a label for the City drop-down list.
	City	Drop-down list			Populated with the [CityName] attribute value retrieved for the user using

					the currently stored foreign key referenced from the City entity. Control options is populated with all city names retrieved from the City entity where they have a foreign key relationship established with the selected [ProvinceID] value.
	Suburb	Label			Used to display a label for the Suburb drop-down list.
	Suburb	Drop-down list			Populated with the [SuburbName] attribute value retrieved for the user using the currently stored foreign key referenced from the Suburb entity. Control options is populated with all suburb names retrieved from the Suburb entity where they have a foreign key relationship established with the selected [CityID] value.
	Account Status	Label			Used as a label for the Account Status drop-down list control.
	Account Status	Drop-Down List			Control is populated with either 'Active' or 'Inactive' value to represent the

					current account status assigned. Selectable options which are loaded include 'Active' or 'Inactive' with the values 1 and 0, respectively.																		
			Back	Button	Navigates administrator back to the Users screen if clicked.																		
			Update	Button	Allows for the submission of updated user details, if any, to the User entity in the database.																		
	Step 5: The administrator enters the updated user details and clicks the Update button. [ALT]		Step 6: The system captures and validates the entered details against the validation requirements for the following entities: From the User entity:																				
			<table><tr><th>Attribute</th><th>Validation Requirement</th></tr><tr><td>FirstName</td><td>Alphabetical characters only; Minimum: 1; Maximum: 50</td></tr><tr><td>Surname</td><td>Alphabetical characters only; Minimum: 1; Maximum: 50</td></tr><tr><td>PhoneNumber</td><td>Digits only; Minimum: 10; Maximum: 10</td></tr><tr><td>StreetAddress</td><td>Alphanumeric characters only; Minimum: 50; Maximum: 50</td></tr><tr><td>Country</td><td>Alphabetical characters only; Minimum: 1; Maximum: 50</td></tr><tr><td>Province</td><td>Alphabetical characters only; Minimum: 1; Maximum: 50</td></tr><tr><td>City</td><td>Alphabetical characters only; Minimum: 1; Maximum: 50</td></tr><tr><td>Suburb</td><td>Alphabetical characters only; Minimum: 1; Maximum: 50</td></tr></table>			Attribute	Validation Requirement	FirstName	Alphabetical characters only; Minimum: 1; Maximum: 50	Surname	Alphabetical characters only; Minimum: 1; Maximum: 50	PhoneNumber	Digits only; Minimum: 10; Maximum: 10	StreetAddress	Alphanumeric characters only; Minimum: 50; Maximum: 50	Country	Alphabetical characters only; Minimum: 1; Maximum: 50	Province	Alphabetical characters only; Minimum: 1; Maximum: 50	City	Alphabetical characters only; Minimum: 1; Maximum: 50	Suburb	Alphabetical characters only; Minimum: 1; Maximum: 50
			Attribute	Validation Requirement																			
			FirstName	Alphabetical characters only; Minimum: 1; Maximum: 50																			
			Surname	Alphabetical characters only; Minimum: 1; Maximum: 50																			
			PhoneNumber	Digits only; Minimum: 10; Maximum: 10																			
			StreetAddress	Alphanumeric characters only; Minimum: 50; Maximum: 50																			
			Country	Alphabetical characters only; Minimum: 1; Maximum: 50																			
			Province	Alphabetical characters only; Minimum: 1; Maximum: 50																			
			City	Alphabetical characters only; Minimum: 1; Maximum: 50																			
			Suburb	Alphabetical characters only; Minimum: 1; Maximum: 50																			

			Account Status	Required; Boolean.
			[ALT]	
			Step 7: The system prompts the administrator to confirm whether they wish to proceed with the update using a modal dialogue box with the following controls:	
			Control Name	Control Type
			Notes	
			Confirmation required	Label
			Confirm updated user details?	Label
			Confirm	Button
			Cancel	Button
			Step 8: The administrator clicks on the Confirm button as they wish to proceed with the update. [ALT]	
			Step 9: The system updates the user account's information in the User table using an Entity Framework SQL UPDATE query:	
			Field Details to be saved	Attributes in the table
			First Name	FirstName
			Surname	Surname
			Phone Number	PhoneNumber
			Street Address	StreetAddress
			Country	CountryID
			Province	ProvinceID
			City	CityID
			Suburb	SuburbID
			Account Status	EmailConfirmed
			Step 10: The system redirects the administrator to the Users screen and displays a confirmation message box with the following controls:	
			Control Name	Control Name
			Notes	

			User details updated successfully	Label	This control is used to indicate a successful update																																	
ALTERNATE COURSES:	ALT Step 5: The administrator clicks on the 'Cancel' button The use case is terminated																																					
	ALT Step 6 a: The system fails to validate the user details entered and notifies the administrator of what caused the validation errors.																																					
	Required validation error messages for empty fields:																																					
	<table><tr><th>Validation Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>First Name</td><td>Textbox</td><td>Control's border outline has the colour red if the field contained invalid input.</td></tr><tr><td>'First Name' field is required</td><td>Label</td><td>Red label used to inform that the [FirstName] field cannot be null when attempting an update in User table</td></tr><tr><td>Surname</td><td>Textbox</td><td>Control's border outline has the colour red if the field contained invalid input.</td></tr><tr><td>'Surname' field is required</td><td>Label</td><td>Red label used to inform that the [Surname] field cannot be null when attempting an update in User table</td></tr><tr><td>Phone Number</td><td>Textbox</td><td>Control's border outline has the colour red if the field contained invalid input.</td></tr><tr><td>'Phone Number' field is required</td><td>Label</td><td>Red label used to inform that the [PhoneNumber] field cannot be null when attempting an update in User table</td></tr><tr><td>Street Address</td><td>Textbox</td><td>Control's border outline has the colour red if the field contained invalid input.</td></tr><tr><td>'Street Address' field is required</td><td>Label</td><td>Red label used to inform that the [StreetAddress] field cannot be null when attempting an update in User table</td></tr><tr><td>Country</td><td>Drop-Down List</td><td>Control's border outline has the colour red if the field contained invalid input.</td></tr><tr><td>'Country' field is required</td><td>Label</td><td>Red label used to inform that the [Country] field cannot be null when attempting an update in User table</td></tr></table>					Validation Name	Control Type	Notes	First Name	Textbox	Control's border outline has the colour red if the field contained invalid input.	'First Name' field is required	Label	Red label used to inform that the [FirstName] field cannot be null when attempting an update in User table	Surname	Textbox	Control's border outline has the colour red if the field contained invalid input.	'Surname' field is required	Label	Red label used to inform that the [Surname] field cannot be null when attempting an update in User table	Phone Number	Textbox	Control's border outline has the colour red if the field contained invalid input.	'Phone Number' field is required	Label	Red label used to inform that the [PhoneNumber] field cannot be null when attempting an update in User table	Street Address	Textbox	Control's border outline has the colour red if the field contained invalid input.	'Street Address' field is required	Label	Red label used to inform that the [StreetAddress] field cannot be null when attempting an update in User table	Country	Drop-Down List	Control's border outline has the colour red if the field contained invalid input.	'Country' field is required	Label	Red label used to inform that the [Country] field cannot be null when attempting an update in User table
	Validation Name	Control Type	Notes																																			
	First Name	Textbox	Control's border outline has the colour red if the field contained invalid input.																																			
	'First Name' field is required	Label	Red label used to inform that the [FirstName] field cannot be null when attempting an update in User table																																			
	Surname	Textbox	Control's border outline has the colour red if the field contained invalid input.																																			
	'Surname' field is required	Label	Red label used to inform that the [Surname] field cannot be null when attempting an update in User table																																			
	Phone Number	Textbox	Control's border outline has the colour red if the field contained invalid input.																																			
'Phone Number' field is required	Label	Red label used to inform that the [PhoneNumber] field cannot be null when attempting an update in User table																																				
Street Address	Textbox	Control's border outline has the colour red if the field contained invalid input.																																				
'Street Address' field is required	Label	Red label used to inform that the [StreetAddress] field cannot be null when attempting an update in User table																																				
Country	Drop-Down List	Control's border outline has the colour red if the field contained invalid input.																																				
'Country' field is required	Label	Red label used to inform that the [Country] field cannot be null when attempting an update in User table																																				

	Province	Drop-Down List	Control's border outline has the colour red if the field contained invalid input.
	'Province' field is required	Label	Red label used to inform that the [Country] field cannot be null when attempting an update in User table
	City	Drop-Down List	Control's border outline has the colour red if the field contained invalid input.
	'City' field is required	Label	Red label used to inform that the [City] field cannot be null when attempting an update in User table.
	Suburb	Drop-Down List	Control's border outline has the colour red if the field contained invalid input.
	'Suburb' field is required	Label	Red label used to inform that the [Suburb] field cannot be null when attempting an update in User table.
	ALT Step 6 b: Invalid input has been provided when updating the user's account details and the following validation error message(s) are displayed for each field that raised an input validation error respectively:		
	Control Name	Control Name	Notes
	First Name	Textbox	Field's border outline is the colour red if it contains invalid input.
	Invalid First Name provided. No digits or special characters are allowed.	Label	Validation for [FirstName] attribute. Indicates valid format for First Name field. Constrained to typing 50 characters only.
	Surname	Textbox	Field's border outline is the colour red if it contains invalid input.
	Invalid Surname provided. No digits or special characters are allowed.	Label	Validation for [Surname] attribute. Indicates valid format for Surname field. Constrained to typing 50 characters only.
	Phone Number	Textbox	Field's border outline is the colour red if it contains invalid input.
	Please enter a valid Phone Number .	Label	Validation for [PhoneNumber] attribute. Requires a valid phone number format.

			Constrained to typing 10 digits only.
	Street Address	Textbox	Field's border outline is the colour red if it contains invalid input.
	Please enter a valid Street Address .	Label	Validation for [StreetAddress] attribute. Requires a valid street address format. Constrained to typing 50 characters. Alphanumerical characters supported.
	ALT Step 8: The administrator clicks on the ' Cancel ' button as they do not wish to proceed with the update: Return to Step 8		
CONCLUSION:	This use case concludes when the system displays an update successful confirmation message box on the Users screen.		
POST-CONDITION:	The user account's details is updated in the User entity .		
BUSINESS RULES	- Only the administrator has the necessary permissions to use this functionality on the system. - The administrator should not be allowed to view or update the passwords of the users registered on the system. - The administrator should not be permitted to update the email address of an existing user account using this use case. - A user's role cannot be changed after the account has been created.		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	- Only digits will be allowed to be typed for the Phone Number attribute. - Only alphabetical characters can be captured in the First Name and Surname textboxes - The address that can be set for the user location is limited to what selectable options has been added to the Country, Province, City and Suburb entities respectively. - The email address of a user account cannot be updated as the field will be read-only. - Multiple roles cannot be assigned to a single user.		
ASSUMPTIONS:	A search will always be performed to find a specific user account to update.		
OPEN ISSUES:	None		

Table 12 - 2.4 Delete User

ADMINISTRATIVE SUBSYSTEM			
AUTHORS (s): Kyle van Eeden		DATE: 03/05/2022	
VERSION: 1.1		LAST REVIEW DATE: 30/05/2022	
USE CASE NAME:	Delete User	USE CASE TYPE	
USE CASE ID:	2.4	Business Requirements:	☐
PRIORITY:	High	System Analysis:	☐
SOURCE:	ByteXecom Requirements List	System Design:	☒
PRIMARY BUSINESS ACTOR	Administrator		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This use case describes the event in which the administrator wishes to remove a user's account from the system. The use case begins with the administrator requesting for a specific user account to be removed, and then the system retrieves a read-only view of all the current data stored on the user from the User table. The administrator clicks the 'Delete' button and after confirming the modal dialogue prompt to indicate whether to proceed with the user account removal, the system will remove the user account from the database if the user account is not associated with an existing order in the Order entity. The use case concludes with the system displaying a user delete success message box notification on the Orders screen.		
PRE-CONDITION:	- The administrator must be logged onto the system. - The user to be deleted must exist on the system.		
TRIGGER:	The use case is triggered when the administrator wishes to delete a user from the system.		
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE	
		MANUAL ACTION	AUTOMATED ACTION
	Step 1: The administrator wishes to remove a user account from the system.		Step 2: The system invokes use case 2.2 "Search User"
	Step 3: The administrator clicks on Delete button for the specific user account they wish to remove.		Step 4: The system retrieves the following details for the user requested from the User entity using an Entity Framework SQL Read query:

Details to be read	Attributes in the table
Name	[FirstName]
Surname	[Surname]
Email Address	[Email]
Phone Number	[PhoneNumber]
Street Name	[StreetName]
Country	[Country_ID]
Province	[Province_ID]
City	[City_ID]
Suburb	[Suburb_ID]
Account Status	[EmailConfirmed]

The system displays the **Delete User** screen populated with the specific user's account details using the following controls:

Control Name	Control Type	Notes
User Details	Heading	This control is used to display the heading of the Delete User screen.
First Name	Label	This control is used as a label for the First Name textbox control.
First Name	Textbox	Read-Only ; Populated with the [FirstName] attribute value retrieved for the user.
Surname	Label	This control is used for to describe the content in the Surname textbox control.
Surname	Textbox	Read-Only ; Populated with the [Surname] attribute value retrieved for the specific user.

			Email	Label	This control is used for to describe the content in the Email textbox control.
			Email	Textbox	Read-Only ; Populated with the [Email] attribute value retrieved for the requested user.
			Phone Number	Label	This control is as a label for the the Phone Number textbox control.
			Phone Number	Textbox	Read-Only ; Populated with the [PhoneNumber] attribute value retrieved for the requested user.
			Street Address	Label	This control is used for to describe the content in the Street Address textbox
			Street Address	Textbox	Read-Only ; Populated with the [StreetAddress] attribute value retrieved for the requested user.
			Country	Label	Used as a label for the Country drop-down list control.
			Country	Textbox	Read-Only ; Populated with the current [CountryName] attribute value retrieved for the user and which is referenced from the City entity via the [CountryID {FK}] foreign key attribute

					value currently assigned to and stored for the user.
	Province	Label			Used as a label for the Province drop-down list control.
	Province	Textbox			Read-Only ; Populated with the current [ProvinceName] attribute value retrieved for the user and which is referenced from the City entity via the [ProvinceID {FK}] foreign key attribute value currently assigned to and stored for the user.
	City	Label			Used as a label for the City drop-down list.
	City	Textbox			Read-Only ; Populated with the current [CityName] attribute value retrieved for the user and which is referenced from the City entity via the [CityID {FK}] foreign key attribute value currently assigned to and stored for the user.
	Suburb	Label			Used as a label for the Suburb drop-down list.
	Suburb	Drop-Down List			Read-Only ; Populated with the current [SuburbName] attribute value retrieved for the user and

					which is referenced from the City entity via the [SuburbName {FK}] foreign key attribute value currently assigned to and stored for the user.
			Back	Button	Navigates administrator back to the 'Users' screen if clicked.
			Delete	Button	Allows the administrator to remove the user's account from the database ,if no orders are associated, once clicked.
	Step 5: The administrator clicks on the 'Delete' button. [ALT]		Step 6: The system prompts the administrator to confirm whether they wish to proceed with the delete operation by displaying a modal dialogue box on the current screen with the following controls:		
			Control Name	Control Type	Notes
			Confirmation required	Label	This control is used to indicate the type of message box displayed.
			Confirm user removal?	Label	This control is used to request confirmation for the deletion of the user account requested.
			Confirm	Button	This control is used to confirm proceeding with the deletion of the requested user' account.
			Cancel	Button	This control is used to cancel the delete

					operation and close the modal dialogue message box.					
	Step 7: The administrator clicks on the 'Confirm' button. [ALT]		Step 8: The system uses a .NET controller with an Entity Framework SQL Read query to retrieve all orders from the Order entity and performs a check to see whether the user account requested for removal has an association with any order which has been placed on the system The validation to be performed will check whether the [User_ID] attribute value of the user account requested for removal is associated with any of the orders which has been retrieved via a foreign key relationship. [ALT]							
			Step 9: The system uses an Entity Framework SQL Delete query to remove the specific user account requested from the User entity .							
			Step 10: The system redirects the administrator to the Users screen and displays a message box notification with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>User Deleted Successfully</td><td>Label</td><td>This control is used to indicate a successful deletion of the user account.</td></tr></table>			Control Name	Control Name	Notes	User Deleted Successfully	Label
Control Name	Control Name	Notes								
User Deleted Successfully	Label	This control is used to indicate a successful deletion of the user account.								
	ALT Step 5: The administrator no longer wishes to remove the user's account and clicks on the 'Back' button. - The administrator is navigated back to the 'Users' screen. - The use case is terminated									
	ALT Step 7: The administrator does not wish to delete the user account and clicks the 'Cancel' button in the modal dialogue box. Return to Step 6									
	ALT Step 8: The user requested for removal has placed an order on the system and cannot be deleted based on identified business rules. The system will display a message box notification on the user interface to inform the administrator that the user cannot be deleted. The message box will have the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>User cannot be deleted</td><td>Label</td><td>This control is used to indicate that the user account cannot be</td></tr></table>					Control Name	Control Name	Notes	User cannot be deleted	Label
Control Name	Control Name	Notes								
User cannot be deleted	Label	This control is used to indicate that the user account cannot be								

	since their account is associated with an order placed on the system.		removed since the user is associated with an order from the Order entity.	
CONCLUSION:	This use case concludes when a user account has been successfully deleted from the system and the administrator is notified of the successful removal via a confirmation message box.			
POST-CONDITION:	The user account has been removed from the User entity.			
BUSINESS RULES	- Only the administrator is permitted access to this functionality. - A user account cannot be deleted from the system if it is associated with an order which has been placed on the system. - No details displayed for requested user account should be amendable.			
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	- Inability to delete a user from the system if their account is associated with an order in the Order table. - All details for fields displayed on the Delete User screen will be in read-only format.			
ASSUMPTIONS:	Business stakeholders only intend on storing information on the system if it can be used for business reporting purposes. Users which has not placed an order on the system can be removed since no useful insights can be retrieved from the information stored on the user account.			
OPEN ISSUES:	None			

Table 13 - 2.5 View Audit Trail

Table 14 - 2.6 Search Audit Trail

2.3. Reseller Subsystem

Table 15 - 3.1 Send Payment Reminder

RESELLER SUBSYSTEM				
AUTHORS (s): Nomusa Vumisa		DATE: 19/07/2022		
VERSION: 1.0		LAST REVIEW DATE: 19/07/2022		
USE CASE NAME:	Send Payment Reminder		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒	
USE CASE ID:	3.1			
PRIORITY:	High			
SOURCE:	ByteXcom Requirements List			
PRIMARY BUSINESS ACTOR	• Time			
PRIMARY SYSTEM ACTOR	• None			
OTHER PARTICIPATING ACTORS:	• Reseller			
OTHER INTERESTED STAKEHOLDERS:	• None			
DESCRIPTION:	The use case begins when a reseller order payment is still outstanding. and the order has reached the notification threshold. The system then sends an email reminder to the reseller at half the due date and on the due date.			
PRE-CONDITION:	The payment status must be “Payment Outstanding”			
TRIGGER:	This use case is called when the payment status is outstanding and the payment threshold has been reached.			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE		
		MANUAL ACTION	AUTOMATED ACTION	
	Step 1: The payment threshold has been reached		Step 2: The system retrieves the following attributes	
			From the Order Table:	
			Details to be read	Attributes in the table
			Payment Status	PaymentStatus
			Order Date	OrderDate
			From the PaymentReminder Table:	
			Details to be read	Attributes in the table
			Value	Value

			and verifies if the payment status is “Payment Outstanding” and if half the due date threshold has been reached, by adding the payment reminder value to the order date and then dividing that value by 2																										
			[ALT]																										
			Step 3: The system generates a payment reminder email with the following details:																										
			<table><tr><th>Details</th><th>Notes</th></tr><tr><td>To</td><td>The resellers email address – retrieved from the User Table using an SQL READ query</td></tr><tr><td>From</td><td>The company email address, stored as a “do not reply” address on the system</td></tr><tr><td>Subject (If half the due date is reached)</td><td>“Payment Reminder”</td></tr><tr><td>Subject (If the due date is reached)</td><td>“Payment Due”</td></tr><tr><td>Body – Order Number</td><td>The order number retrieved from the Order table using an SQL READ query</td></tr><tr><td>Body – Order Date</td><td>The order date retrieved from the Order table using an SQL READ query</td></tr><tr><td>Body – Payment Status</td><td>The payment status retrieved from the Order table using an SQL READ query</td></tr><tr><td>Body – Order Total</td><td>The order total retrieved from the Order table using an SQL READ query</td></tr><tr><td>Body – Payment Due Date</td><td>The payment due date retrieved from the PaymentReminder table using an SQL READ query</td></tr><tr><td>Body – First Name</td><td>The users name retrieved from the User table using an SQL READ query</td></tr><tr><td>Body – Main Text (If half time is reached)</td><td>“Hi (first name), A kind reminder that the invoice for order #(order number), with a total of (order total) is due on (payment due date). To query your order, click the button below or contact us on 075 589 4569”</td></tr><tr><td>Body – Main Text (If due date is reached)</td><td>“Hi (first name), The invoice for order #(order number), with a total of (order total) is due today (payment due date). To query your order, click the button below or</td></tr></table>	Details	Notes	To	The resellers email address – retrieved from the User Table using an SQL READ query	From	The company email address, stored as a “do not reply” address on the system	Subject (If half the due date is reached)	“Payment Reminder”	Subject (If the due date is reached)	“Payment Due”	Body – Order Number	The order number retrieved from the Order table using an SQL READ query	Body – Order Date	The order date retrieved from the Order table using an SQL READ query	Body – Payment Status	The payment status retrieved from the Order table using an SQL READ query	Body – Order Total	The order total retrieved from the Order table using an SQL READ query	Body – Payment Due Date	The payment due date retrieved from the PaymentReminder table using an SQL READ query	Body – First Name	The users name retrieved from the User table using an SQL READ query	Body – Main Text (If half time is reached)	“Hi (first name), A kind reminder that the invoice for order #(order number), with a total of (order total) is due on (payment due date). To query your order, click the button below or contact us on 075 589 4569”	Body – Main Text (If due date is reached)	“Hi (first name), The invoice for order #(order number), with a total of (order total) is due today (payment due date). To query your order, click the button below or
	Details	Notes																											
	To	The resellers email address – retrieved from the User Table using an SQL READ query																											
	From	The company email address, stored as a “do not reply” address on the system																											
	Subject (If half the due date is reached)	“Payment Reminder”																											
	Subject (If the due date is reached)	“Payment Due”																											
	Body – Order Number	The order number retrieved from the Order table using an SQL READ query																											
	Body – Order Date	The order date retrieved from the Order table using an SQL READ query																											
	Body – Payment Status	The payment status retrieved from the Order table using an SQL READ query																											
	Body – Order Total	The order total retrieved from the Order table using an SQL READ query																											
	Body – Payment Due Date	The payment due date retrieved from the PaymentReminder table using an SQL READ query																											
	Body – First Name	The users name retrieved from the User table using an SQL READ query																											
Body – Main Text (If half time is reached)	“Hi (first name), A kind reminder that the invoice for order #(order number), with a total of (order total) is due on (payment due date). To query your order, click the button below or contact us on 075 589 4569”																												
Body – Main Text (If due date is reached)	“Hi (first name), The invoice for order #(order number), with a total of (order total) is due today (payment due date). To query your order, click the button below or																												

				contact us on 075 589 4569”
			Button	The “Manage Order” button directs the user to their order on the system
			Body – Banking Details	“Bankind Details Bank Name: Sasfin Bank Account Name: Stromacraft Pty Ltd Branch Code: 683 000 Account Number: 30000228020”
			Step 4: The system sends the reminder to the reseller using the Sendgrid API.	
ALTERNATIVE STEPS:	ALT Step 2 – The full due date threshold has been reached • Proceed to step 3			
CONCLUSION:	This use case concludes when the reminder has been sent to the reseller			
POST-CONDITION:	The reseller is informed of their outstanding payment using information retrieved from the User and Order tables			
BUSINESS RULES	• None			
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• Only the reseller should be notified. • Reminders must be sent via email.			
ASSUMPTIONS:	• None			
OPEN ISSUES:	• None			

Table 16 - 3.2 Capture Proof of Payment

RESELLER SUBSYSTEM				
AUTHORS (s): Aphiwe Shoji			DATE: 21/07/2022	
VERSION: 1.0			LAST REVIEW DATE: 21/07/2022	
USE CASE NAME:	Capture Proof of Payment		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒	
USE CASE ID:	3.2			
PRIORITY:	High			
SOURCE:	ByteXcom Requirements List			
PRIMARY BUSINESS ACTOR	• Administrator			
PRIMARY SYSTEM ACTOR	• None			
OTHER PARTICIPATING ACTORS:	• None			
OTHER INTERESTED STAKEHOLDERS:	• None			
DESCRIPTION:	This use case describes the process in which the administrator wishes to capture a resellers proof of payment on the system. For resellers with orders that have an order status of “processing” and payment status “payment outstanding” , the reseller can capture the resellers proof of payment on the system once the reseller has paid thus updating the order status of the order to “paid”			
PRE-CONDITION:	• The administrator must be logged in. • The payment status of the order must be “payment outstanding” • The administrator is on the View Order Details Screen			
TRIGGER:	This use case is triggered when the administrator wishes to capture a reseller’s proof of payment on the system.			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE		
		MANUAL ACTION	AUTOMATED ACTION	
	Step 1: The administrator clicks the “capture payment” button		Step 2: the system prompts the administrator to confirm the capture through loading a confirmation modal dialogue with the following controls:	
			Control Name	Control Type
		“Confirm Dispatching of Order”	Label	This control is used to request confirmation on the dispatching of the order

			Confirm	Button	This control is used to confirm the dispatching of the order
			Cancel	Button	This control is used to close the modal dialogue
	Step 3: The system administrator clicks on the confirm button[ALT]		Step 4: The system will then save the capture to the Proof of payment to the Proof of payment table using a method in a .NET controller with the following attributes:		
			Details to be read	Attributes in the table	Notes
			Capture Payment	CapturePayment	Required
			Reseller	Reseller	Required
			Step 5: The system will then update the OrderStatus in the Order entity to “Paid” using a method in a .NET controller		
			Details to be updated	Attributes in the table	Notes
			OrderStatus	OrderStatus	
			Step 6: the system will displays a bootstrap alert notification with the following controls:		
		Control Name	Control Name	Notes	
		“Reseller order payment captured successfully”	Label		
		Step 3: The system generates a payment reminder email with the following details:			
		Details	Notes		
		To	The resellers email address – retrieved from the User Table using an SQL READ query		
		From	The company email address, stored as a “do not reply” address on the system		
		Subject	“Payment Received”		

			Body – Order Number	The order number retrieved from the Order table using an SQL READ query
			Body – Order Date	The order date retrieved from the Order table using an SQL READ query
			Body – Order Total	The order total retrieved from the Order table using an SQL READ query
			Body – Payment Received	The payment received information retrieved from the Capture Payment table using an SQL READ query
			Body – First Name	The users name retrieved from the User table using an SQL READ query
			Body – Main Text	“Hi (first name), Payment for order # (order number) with a total of (order total) has been received. Thank you! To query your order, click the button below or contact us on 075 589 4569”
			Button	The “Manage Order” button directs the user to their order on the system
			Step 4: The system will finally send and email to the reseller using the SendGrid API.	
ALTERNATE COURSES:	ALT Step 5: The administrator does not confirm the capture • Terminate use case			
CONCLUSION:	The system displays a notification confirming the proof of payment was captured successfully			
POST-CONDITION:	The reseller’s order status is updated to paid and an email is sent to the reseller.			
BUSINESS RULES	• Only resellers can make payments after placing an order • Only the administrator can capture a proof of payment			
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• The administrator has a stable internet connection.			
ASSUMPTIONS:	• None			
OPEN ISSUES:	• None			

2.4. Product Subsystem

Table 17 - 4.1 Add Product

PRODUCT SUBSYSTEM			
AUTHORS (s): Thenjiwe Ntsonda		DATE: 18/04/2022	
VERSION: 1.0		LAST REVIEW DATE: 21/08/2022	
USE CASE NAME:	Add Product	USE CASE TYPE	
USE CASE ID:	4.1	Business Requirements:	☐
PRIORITY:	High	System Analysis:	☐
SOURCE:	ByteXcom Requirements List	System Design:	☒
PRIMARY BUSINESS ACTOR	Administrator		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	Customers and Resellers: The newly added product is displayed on the product catalogue with different prices depending on the user.		
DESCRIPTION:	This use case describes the process in which the administrator wishes to add a new product to the system to be viewed and ordered by customers and resellers. The use case begins when the administrator adds the new product as an option for customers and resellers to choose from. The administrator will add all the pertinent information regarding the product and save it to the business database.		
PRE-CONDITION:	- The administrator must be logged onto the system. - A product category or brand relevant to the product added needs to exist before the product is created.		
TRIGGER:	The use case is triggered when the administrator wants to add a new product which customers and resellers can order.		
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE	
		MANUAL ACTION	AUTOMATED ACTION

Step 1: The administrator navigates to the **Product Screen** under the Product tab on the side bar

Step 2: The system retrieves the product information from the following attributes and tables using a method in a .NET Controller and an SQL READ Query:

From the **Product** table:

Details to be read	Attributes in the table
Name	Name
Description	Description
PictureUrl	PictureUrl
Customer Price	CustomerPrice
Reseller Price	ResellerPrice
Quantity On Hand	QuantityOnHand
Threshold Value	ThresholdValue

From the **ProductCategory** table:

Details to be read	Attributes in the table
Name	Name

From the **ProductBrand** table:

Details to be read	Attributes in the table
Name	Name

And displays the **Product Screen** with the following controls:

Control Name	Control Type	Notes
Products List	Heading	Page main heading
Question Mark Icon	Button	When clicked, this control will open a pdf which contains the help document
Create New Product	Button	When clicked, this control will direct the user to the Add product page
Search bar	Textbox	This control will be used

					when the user inserts the search parameters
	Search icon	Button			This control will be used to activate the search
	Results table	Table			This control will be used to display all the products that exist on the system
	Name	Label			This control will be used as a heading to display the Name for all products
	Description	Label			This control will be used as a heading to display the Description for all products
	Customer Price	Label			This control will be used as a heading to display the customer price of the product
	Reseller Price	Label			This control will be used as a heading to display the reseller price of the product
	Product Category	Label			This control will be used as a heading to display the product category of the product.
	Product Brand	Label			This control will be used as a heading to display the product brand of the product.
	Action	Table Heading			This control will be used as a heading to display the actions the administrator can perform with each entry – Edit and Delete
	Action - Update	Button			This control will be placed in a row, with the

					delete button, under the "Action" heading for each entry in the Supplier table. The button will direct the user to the Update Product screen
			Action - Delete	Button	This control will be placed in a row, with the delete button, under the "Action" heading for each entry in the Supplier table. The button will direct the user to the Delete Product Details screen
			Showing "" to "" of "" entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database
			"Previous" and "Next"	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on

Step 3: The system administrator clicks on the **Create New Product** button

Step 4: The system retrieves following information to populate the Category and Brand dropdowns using an SQL READ QUERY

From the **ProductCategory** table:

Details to be read	Attributes in the table
Name	Name

From the **ProductBrand** table:

Details to be read	Attributes in the table
Name	Name

And displays the **Create Product** screen with the following controls:

Control Name	Control Type	Notes
Add Product	Heading	Page main heading
Product information	Heading	Page subheading
Name	Label	This control will be used as a label to indicate the Name attribute of the product record
Name	Textbox	This control will be used by the administrator to insert a name of the product
Description	Label	This control will be used as a label to indicate the Description attribute of the product record
Description	Textbox	This control will be used by the administrator to insert a description for the product
Image	Label	This control will be used as a label to indicate the Image attribute of the product record
Image	Image Upload Field	This control will be used

					by the administrator to insert an image for the product
	Customer Price	Label			This control will be used as a label to indicate the customer price attribute of the product record
	Customer Price	Textbox			This control will be used by the administrator to enter a customer price for the product
	Reseller Price	Label			This control will be used as a label to indicate the reseller price attribute of the product record
	Reseller Price	Textbox			This control will be used by the administrator to enter a reseller price for the product
	Threshold Value	Label			This control will be used as a label to indicate the threshold value attribute of the product record
	Threshold Value	Textbox			This control will be used by the administrator to enter the threshold value of the product
	Product Brand	Label			This control will be used as a label to indicate the product brand attribute of the product record
	Product Brand	Dropdown			This control will be used by the administrator to select a brand for the product

			Product Category	Label	This control will be used as a label to indicate the product category attribute of the product record
			Product Category	Dropdown	This control will be used by the administrator to select a category for the product
			Display Product	Label	This control will be used as a label to indicate the display product attribute of the product record
			Display Product	Dropdown	This control will be used by the administrator to select the display options for the product
			Create	Button	This control will be used to confirm the addition of the product
			Cancel	Button	This control will be used to cancel the addition of the product
Note: All controls are enabled by default unless stated otherwise.					

Step 5: The system administrator enters the details and clicks on the **Create** button [ALT]

Step 6: The system captures and validates the entered details against the **Product** table [ALT]

Details to be validated	Attributes in the product table	Notes
Name	Name	Required, Unique,
Description	Description	Required
Image	Image	Required
Customer Price	CustomerPrice	Required, No letters or special characters allowed.
Reseller Price	ResellerPrice	Required, No letters or special characters allowed.
Threshold Value	ThresholdValue	Required, No letters or special characters allowed, Only numerical characters allowed
Product Brand	Brand	Required
Product Category	Category	Required
Display Product	DisplayProduct	Required

Step 7: The system uses an SQL READ query to check if the product already exists in the **Product** table using the following attributes [ALT]

Details to be read	Attributes in the table
Name	Name
Description	Description
PictureUrl	PictureUrl
Customer Price	CustomerPrice
Reseller Price	ResellerPrice
Quantity On Hand	QuantityOnHand
Threshold Value	ThresholdValue

			Step 8: The system saves the information into the Product table using an INSERT SQL query:																																	
			<table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>ProductId</td><td>ProductId</td><td>A unique value generated by the system and saved to the database.</td></tr><tr><td>Name</td><td>Name</td><td>Required</td></tr><tr><td>Description</td><td>Description</td><td>Required</td></tr><tr><td>Image</td><td>Image</td><td>Required</td></tr><tr><td>Customer Price</td><td>CustomerPrice</td><td>Required</td></tr><tr><td>Reseller Price</td><td>ResellerPrice</td><td>Required</td></tr><tr><td>Threshold Value</td><td>ThresholdValue</td><td>Required</td></tr><tr><td>Product Brand</td><td>Brand</td><td>Required</td></tr><tr><td>Product Category</td><td>Category</td><td>Required</td></tr><tr><td>Display Product</td><td>DisplayProduct</td><td>Required</td></tr></table>	Details to be saved	Attributes in the table	Notes	ProductId	ProductId	A unique value generated by the system and saved to the database.	Name	Name	Required	Description	Description	Required	Image	Image	Required	Customer Price	CustomerPrice	Required	Reseller Price	ResellerPrice	Required	Threshold Value	ThresholdValue	Required	Product Brand	Brand	Required	Product Category	Category	Required	Display Product	DisplayProduct	Required
	Details to be saved	Attributes in the table	Notes																																	
	ProductId	ProductId	A unique value generated by the system and saved to the database.																																	
	Name	Name	Required																																	
	Description	Description	Required																																	
	Image	Image	Required																																	
	Customer Price	CustomerPrice	Required																																	
	Reseller Price	ResellerPrice	Required																																	
	Threshold Value	ThresholdValue	Required																																	
	Product Brand	Brand	Required																																	
	Product Category	Category	Required																																	
Display Product	DisplayProduct	Required																																		
		Step 9: The system redirects the system administrator to the Product Screen and displays a message box with the following controls:																																		
		<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Checkbox</td><td>Icon</td><td>This control is used to indicate a successful creation</td></tr><tr><td>“Product Created Successfully”</td><td>Label</td><td>This control is used to indicate a successful creation</td></tr></table>	Control Name	Control Name	Notes	Checkbox	Icon	This control is used to indicate a successful creation	“Product Created Successfully”	Label	This control is used to indicate a successful creation																									
Control Name	Control Name	Notes																																		
Checkbox	Icon	This control is used to indicate a successful creation																																		
“Product Created Successfully”	Label	This control is used to indicate a successful creation																																		
ALTERNATE COURSES:	ALT Step 6: The administrator does not want to add a product anymore - The administrator clicks on the cancel button - The use case is terminated																																			
	ALT Step 7: The Product already exists on the system - The system displays an error modal dialogue with the following controls																																			
	<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Product already exists!</td><td>Label</td><td>This control is used to inform the administrator that the supplier already exists</td></tr><tr><td>Okay</td><td>Button</td><td>This control is used to cancel the creation of the supplier</td></tr></table>	Control Name	Control Name	Notes	Product already exists!	Label	This control is used to inform the administrator that the supplier already exists	Okay	Button	This control is used to cancel the creation of the supplier																										
	Control Name	Control Name	Notes																																	
Product already exists!	Label	This control is used to inform the administrator that the supplier already exists																																		
Okay	Button	This control is used to cancel the creation of the supplier																																		

- The administrator clicks the “Okay” button
- The use case is terminated

ALT Step 8A: The details entered are not in the correct format or the required fields are empty

- The system displays a validation error message box with the following controls

Control Name	Control Name	Notes
Name	Textbox	This control will be outline in red to show a validation error
Name is required	Label	This control will be used as a label to indicate a validation error in the Name textbox
Description	Textbox	This control will be outline in red to show a validation error in the Description textbox
Description is required	Label	This control will be used as a label to indicate a validation error in the Description textbox
Customer Price	Textbox	This control will be outline in red to show a validation error in the Customer Price textbox
Customer Price is required	Label	This control will be used as a label to indicate a validation error in the Customer Price textbox
Reseller Price	Textbox	This control will be outline in red to show a validation error in the Reseller Price textbox
Reseller Price is required	Label	This control will be used as a label to indicate a validation error in the Reseller Price textbox
Threshold Value	Label	This control will be outline in red to show a validation error in the Threshold Value textbox
Threshold Value is required	Textbox	This control will be used as a label to indicate a validation error in the Threshold Value textbox
Image	Image Upload Field	This control will be outline in red to show a validation error in the Image upload field
Image is required	Label	This control will be used as a label to indicate a validation error in the Image upload field
Product Brand	Dropdown	This control will be outline in red to show a validation error in the Product Brand dropdown
Product Brand is required	Label	This control will be used as a label to indicate a validation error in the Product Brand dropdown
Product Category	Dropdown	This control will be outline in red to show a validation error in the Product Category dropdown
Product Category is required	Label	This control will be used as a label to indicate a validation

			error in the Product Category dropdown
	Display Product	Dropdown	This control will be outline in red to show a validation error in the Display Product dropdown
	Display Product is required	Label	This control will be used as a label to indicate a validation error in the Display Product dropdown
	Return to step 3		
CONCLUSION:	This use case concludes when a product has been added on the system		
POST-CONDITION:	A new user has been added to the Product table in the database		
BUSINESS RULES	Only the administrator has access to this function.		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	The system is offline, and the system administrator will not be able to add a new product.		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

Table 18 - 4.2 Search Product

PRODUCT SUBSYSTEM											
AUTHORS (s): Thenjiwe Ntsonda		DATE: 18/04/2022									
VERSION: 1.0		LAST REVIEW DATE: 21/08/2022									
USE CASE NAME:	Search Product		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒								
USE CASE ID:	4.2										
PRIORITY:	Medium										
SOURCE:	ByteXcom Requirements List										
PRIMARY BUSINESS ACTOR	• Administrator										
PRIMARY SYSTEM ACTOR	• None										
OTHER PARTICIPATING ACTORS:	• None										
OTHER INTERESTED STAKEHOLDERS:	• None										
DESCRIPTION:	This use case describes the process in which the administrator searches for products on the system. The system responds by displaying the information relating to the search query.										
PRE-CONDITION:	• The administrator must be logged onto the system.										
TRIGGER:	This use case is called when the administrator wants to search for a product on the system.										
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE									
		MANUAL ACTION	AUTOMATED ACTION								
	Step 1: The administrator wishes to search for a product on the system.										
	Step 2: The administrator navigates to the Product Screen under the Product tab on the side bar		Step 3: The system retrieves following information using an SQL READ QUERY From the ProductCategory table: <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Name</td><td>Name</td></tr></table> From the ProductBrand table: <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Name</td><td>Name</td></tr></table> And displays the Product Screen with the following controls:	Details to be read	Attributes in the table	Name	Name	Details to be read	Attributes in the table	Name	Name
	Details to be read	Attributes in the table									
Name	Name										
Details to be read	Attributes in the table										
Name	Name										

			Control Name	Control Type	Notes
			Products List	Heading	Page main heading
			Question Mark Icon	Button	When clicked, this control will open a pdf which contains the help document
			Create New Product	Button	When clicked, this control will direct the user to the Add product page
			Search bar	Textbox	This control will be used when the user inserts the search parameters
			Search icon	Button	This control will be used to activate the search
			Results table	Table	This control will be used to display all the products that exist on the system
			Name	Label	This control will be used as a heading to display the Name for all products
			Description	Label	This control will be used as a heading to display the Description for all products
			Customer Price	Label	This control will be used as a heading to display the customer price of the product
			Reseller Price	Label	This control will be used as a heading to display the reseller price of the product
			Product Category	Label	This control will be used as a heading to display the product category of the product.

			Product Brand	Label	This control will be used as a heading to display the product brand of the product.
			Action	Table Heading	This control will be used as a heading to display the actions the administrator can perform with each entry – Edit and Delete
			Action - Update	Button	This control will be placed in a row, with the delete button, under the “Action” heading for each entry in the Supplier table. The button will direct the user to the Update Product screen
			Action - Delete	Button	This control will be placed in a row, with the delete button, under the “Action” heading for each entry in the Supplier table. The button will direct the user to the Delete Product Details screen
			Showing “” to “” of “” entries	Label	This control will be used to display the number of entries displayed per page as well as the

					total number of entries in the database	
				"Previous" and "Next"	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on
	Step 4: The administrator enters their search query into the search textbox.		Step 5: The system searches the Product table and displays the product(s) matching the search query in the Product Screen [ALT]			
	ALT Step 5: The search query entered does not match any records existing in the Product entity. The system displays a "No Records Found" message in the Product Screen and on the results table.					
CONCLUSION:	This use case concludes when the required product is displayed on the screen.					
POST-CONDITION:	The administrator selects the required product from the displayed results.					
BUSINESS RULES	Only the administrator has access to this function.					
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None					
ASSUMPTIONS:	None					
OPEN ISSUES:	None					

Table 19 - 4.3 Update Product

PRODUCT SUBSYSTEM				
AUTHORS (s): Thenjiwe Ntsonda		DATE: 18/04/2022		
VERSION: 1.0		LAST REVIEW DATE: 18/04/2022		
USE CASE NAME:	Update Product		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒	
USE CASE ID:	4.3			
PRIORITY:	Medium			
SOURCE:	ByteXcom Requirements List			
PRIMARY BUSINESS ACTOR	• Administrator			
PRIMARY SYSTEM ACTOR	• None			
OTHER PARTICIPATING ACTORS:	• None			
OTHER INTERESTED STAKEHOLDERS:	• Customers and Resellers: The newly updated product is displayed on the product catalogue with different prices depending on the user.			
DESCRIPTION:	This use case describes the process in which the administrator wishes to update a product to the system to be viewed and ordered by customers and resellers. The use case begins when the administrator updates the product as an option for customers and resellers to choose from. The administrator will update all the pertinent information regarding the product and save it to the business database.			
PRE-CONDITION:	• The administrator must be logged onto the system. • A product category or brand relevant to the product added needs to exist before the product is updated.			
TRIGGER:	The use case is triggered when the administrator wants to update a product which customers and resellers can order.			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE		
		MANUAL ACTION	AUTOMATED ACTION	
	Step 1: The administrator wishes to update a product in the product catalogue.		Step 2: The system invokes use case 4.2 Search Product	
	Step 2: The administrator clicks on the Edit button		Step 3: The system retrieves the selected product's information from the following attributes and tables using a method in a .NET Controller and an SQL READ Query: From the Product table: <table><tr><td>Details to be read</td><td>Attributes in the table Notes</td></tr></table>	Details to be read
Details to be read	Attributes in the table Notes			

			<table><tr><td>Name</td><td>Name</td></tr><tr><td>Description</td><td>Description</td></tr><tr><td>PictureUrl</td><td>PictureUrl</td></tr><tr><td>Customer Price</td><td>CustomerPrice</td></tr><tr><td>Reseller Price</td><td>ResellerPrice</td></tr><tr><td>Quantity On Hand</td><td>QuantityOnHand</td></tr><tr><td>Threshold Value</td><td>ThresholdValue</td></tr></table>	Name	Name	Description	Description	PictureUrl	PictureUrl	Customer Price	CustomerPrice	Reseller Price	ResellerPrice	Quantity On Hand	QuantityOnHand	Threshold Value	ThresholdValue							
Name	Name																							
Description	Description																							
PictureUrl	PictureUrl																							
Customer Price	CustomerPrice																							
Reseller Price	ResellerPrice																							
Quantity On Hand	QuantityOnHand																							
Threshold Value	ThresholdValue																							
			From the ProductCategory table:																					
			<table><tr><td>Details to be read</td><td>Attributes in the table Notes</td></tr><tr><td>Name</td><td>Name</td></tr></table>	Details to be read	Attributes in the table Notes	Name	Name																	
Details to be read	Attributes in the table Notes																							
Name	Name																							
			From the ProductBrand table:																					
			<table><tr><td>Details to be read</td><td>Attributes in the table Notes</td></tr><tr><td>Name</td><td>Name</td></tr></table>	Details to be read	Attributes in the table Notes	Name	Name																	
Details to be read	Attributes in the table Notes																							
Name	Name																							
			And displays the Update Product Screen with the following controls:																					
			<table><tr><td>Control Name</td><td>Control Type</td><td>Notes</td></tr><tr><td>Update Product</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Product information</td><td>Heading</td><td>Page subheading</td></tr><tr><td>Name</td><td>Labe</td><td>This control will be used as a label to indicate the Name attribute of the product record</td></tr><tr><td>Name</td><td>Textbox</td><td>This control will be used by the administrator to update a name of the product</td></tr><tr><td>Description</td><td>Label</td><td>This control will be used as a label to indicate the Description attribute of the product record</td></tr><tr><td>Description</td><td>Textbox</td><td>This control will be used by the administrator to update a description for the product</td></tr></table>	Control Name	Control Type	Notes	Update Product	Heading	Page main heading	Product information	Heading	Page subheading	Name	Labe	This control will be used as a label to indicate the Name attribute of the product record	Name	Textbox	This control will be used by the administrator to update a name of the product	Description	Label	This control will be used as a label to indicate the Description attribute of the product record	Description	Textbox	This control will be used by the administrator to update a description for the product
Control Name	Control Type	Notes																						
Update Product	Heading	Page main heading																						
Product information	Heading	Page subheading																						
Name	Labe	This control will be used as a label to indicate the Name attribute of the product record																						
Name	Textbox	This control will be used by the administrator to update a name of the product																						
Description	Label	This control will be used as a label to indicate the Description attribute of the product record																						
Description	Textbox	This control will be used by the administrator to update a description for the product																						

			Image	Label	This control will be used as a label to indicate the Image attribute of the product record
			Image	Image Upload Field	This control will be used by the administrator to update an image for the product
			Customer Price	Label	This control will be used as a label to indicate the customer price attribute of the product record
			Customer Price	Textbox	This control will be used by the administrator to update a customer price for the product
			Reseller Price	Label	This control will be used as a label to indicate the reseller price attribute of the product record
			Threshold Value	Label	This control will be used as a label to indicate the threshold value attribute of the product record
			Threshold Value	Textbox	This control will be used by the administrator to update the threshold value for the product
			Reseller Price	Textbox	This control will be used by the administrator to update a reseller price for the product
			Quantity On Hand	Label	This control will be used as a label to indicate the quantity on hand attribute

					of the product record
			Quantity On Hand	Disabled Textbox	This control will be used to store the quantity of a product, but its value will be calculated by the system.
			Product Brand	Label	This control will be used as a label to indicate the product brand attribute of the product record
			Product Brand	Dropdown	This control will be used by the administrator to select a brand for the product
			Product Category	Label	This control will be used as a label to indicate the product category attribute of the product record
			Product Category	Dropdown	This control will be used by the administrator to select a category for the product
			Display Product	Label	This control will be used as a label to indicate the display product attribute of the product record
			Display Product	Dropdown	This control will be used by the administrator to select the display options for the product
			Update	Button	This control will be used to confirm the addition of the product
			Cancel	Button	This control will be used to cancel the edit of the product

			Note: All controls are enabled by default unless stated otherwise.																														
	Step 4: The system administrator updates the product information and clicks on the Update button [ALT]		Step 5: The system captures and validates the entered details against the Product table [ALT] <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Name</td><td>Name</td><td>Required, Unique,</td></tr><tr><td>Description</td><td>Description</td><td>Required</td></tr><tr><td>Image</td><td>Image</td><td>Required</td></tr><tr><td>Customer Price</td><td>CustomerPrice</td><td>Required, No letters or special characters are allowed.</td></tr><tr><td>Reseller Price</td><td>ResellerPrice</td><td>Required, No letters or special characters are allowed.</td></tr><tr><td>Threshold Value</td><td>ThresholdValue</td><td>Required, No letters or special characters are allowed, Only numerical characters allowed</td></tr><tr><td>Product Brand</td><td>Brand</td><td>Required</td></tr><tr><td>Product Category</td><td>Category</td><td>Required</td></tr><tr><td>Display Product</td><td>DisplayProduct</td><td>Required</td></tr></table>	Details to be read	Attributes in the table	Notes	Name	Name	Required, Unique,	Description	Description	Required	Image	Image	Required	Customer Price	CustomerPrice	Required, No letters or special characters are allowed.	Reseller Price	ResellerPrice	Required, No letters or special characters are allowed.	Threshold Value	ThresholdValue	Required, No letters or special characters are allowed, Only numerical characters allowed	Product Brand	Brand	Required	Product Category	Category	Required	Display Product	DisplayProduct	Required
	Details to be read	Attributes in the table	Notes																														
	Name	Name	Required, Unique,																														
	Description	Description	Required																														
	Image	Image	Required																														
	Customer Price	CustomerPrice	Required, No letters or special characters are allowed.																														
	Reseller Price	ResellerPrice	Required, No letters or special characters are allowed.																														
	Threshold Value	ThresholdValue	Required, No letters or special characters are allowed, Only numerical characters allowed																														
	Product Brand	Brand	Required																														
Product Category	Category	Required																															
Display Product	DisplayProduct	Required																															
		Step 6: The system uses an SQL READ query to check if the product already exists in the Product table using the following attributes [ALT] <table><tr><th>Details to be updated</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Name</td><td>Name</td><td></td></tr><tr><td>Description</td><td>Description</td><td></td></tr><tr><td>ImageUrl</td><td>ImageUrl</td><td></td></tr><tr><td>Customer Price</td><td>CustomerPrice</td><td></td></tr><tr><td>Reseller Price</td><td>ResellerPrice</td><td></td></tr><tr><td>Quantity On Hand</td><td>QuantityOnHand</td><td></td></tr><tr><td>Threshold Value</td><td>ThresholdValue</td><td></td></tr></table>	Details to be updated	Attributes in the table	Notes	Name	Name		Description	Description		ImageUrl	ImageUrl		Customer Price	CustomerPrice		Reseller Price	ResellerPrice		Quantity On Hand	QuantityOnHand		Threshold Value	ThresholdValue								
Details to be updated	Attributes in the table	Notes																															
Name	Name																																
Description	Description																																
ImageUrl	ImageUrl																																
Customer Price	CustomerPrice																																
Reseller Price	ResellerPrice																																
Quantity On Hand	QuantityOnHand																																
Threshold Value	ThresholdValue																																

		Step 7: The system displays a message box with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Confirmation required</td><td>Label</td><td>This control is used to display the type of message box displayed</td></tr><tr><td>Confirm updated product details?</td><td>Label</td><td>This control is used to request confirmation on the updated product details</td></tr><tr><td>Confirm</td><td>Button</td><td>This control is used to confirm the updating of the product details</td></tr><tr><td>Cancel</td><td>Button</td><td>This control is used to close the message box for the user to edit the product details</td></tr></table>	Control Name	Control Type	Notes	Confirmation required	Label	This control is used to display the type of message box displayed	Confirm updated product details?	Label	This control is used to request confirmation on the updated product details	Confirm	Button	This control is used to confirm the updating of the product details	Cancel	Button	This control is used to close the message box for the user to edit the product details															
	Control Name	Control Type	Notes																													
	Confirmation required	Label	This control is used to display the type of message box displayed																													
	Confirm updated product details?	Label	This control is used to request confirmation on the updated product details																													
	Confirm	Button	This control is used to confirm the updating of the product details																													
	Cancel	Button	This control is used to close the message box for the user to edit the product details																													
	Step 8: The system administrator clicks on the Confirm button [ALT]	Step 9: The system saves the information into the Product table using an UPDATE SQL query: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Name</td><td>Name</td><td>Required, Unique,</td></tr><tr><td>Description</td><td>Description</td><td>Required</td></tr><tr><td>Image</td><td>Image</td><td>Required</td></tr><tr><td>Customer Price</td><td>CustomerPrice</td><td>Required</td></tr><tr><td>Reseller Price</td><td>ResellerPrice</td><td>Required</td></tr><tr><td>Threshold Value</td><td>ThresholdValue</td><td>Required</td></tr><tr><td>Product Brand</td><td>Brand</td><td>Required</td></tr><tr><td>Product Category</td><td>Category</td><td>Required</td></tr><tr><td>Display Product</td><td>DisplayProduct</td><td>Required</td></tr></table>	Details to be saved	Attributes in the table	Notes	Name	Name	Required, Unique,	Description	Description	Required	Image	Image	Required	Customer Price	CustomerPrice	Required	Reseller Price	ResellerPrice	Required	Threshold Value	ThresholdValue	Required	Product Brand	Brand	Required	Product Category	Category	Required	Display Product	DisplayProduct	Required
	Details to be saved	Attributes in the table	Notes																													
	Name	Name	Required, Unique,																													
	Description	Description	Required																													
Image	Image	Required																														
Customer Price	CustomerPrice	Required																														
Reseller Price	ResellerPrice	Required																														
Threshold Value	ThresholdValue	Required																														
Product Brand	Brand	Required																														
Product Category	Category	Required																														
Display Product	DisplayProduct	Required																														

			Step 10: The system redirects the system administrator to the Product Screen and displays a message box with the following controls:
ALTERNATE COURSES:	ALT Step 4: The administrator does not want to edit the product anymore - The administrator clicks on the cancel button - The use case is terminated		
	ALT Step 5: The details entered are not in the correct format or the required fields are empty The system displays a validation error message box with the following controls		

	Image	Image Upload Field	This control will be outline in red to show a validation error in the Image upload field								
	Image is required	Label	This control will be used as a label to indicate a validation error in the Image upload field								
	Product Brand	Dropdown	This control will be outline in red to show a validation error in the Product Brand dropdown								
	Product Brand is required	Label	This control will be used as a label to indicate a validation error in the Product Brand dropdown								
	Product Category	Dropdown	This control will be outline in red to show a validation error in the Product Category dropdown								
	Product Category is required	Label	This control will be used as a label to indicate a validation error in the Product Category dropdown								
	Display Product	Dropdown	This control will be outline in red to show a validation error in the Display Product dropdown								
	Display Product is required	Label	This control will be used as a label to indicate a validation error in the Display Product dropdown								
	ALT Step 6: The Supplier already exists on the system - The system displays an error modal dialogue with the following controls										
<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Product already exists!</td><td>Label</td><td>This control is used to inform the administrator that the supplier already exists</td></tr><tr><td>Okay</td><td>Button</td><td>This control is used to cancel the creation of the supplier</td></tr></table>			Control Name	Control Name	Notes	Product already exists!	Label	This control is used to inform the administrator that the supplier already exists	Okay	Button	This control is used to cancel the creation of the supplier
Control Name	Control Name	Notes									
Product already exists!	Label	This control is used to inform the administrator that the supplier already exists									
Okay	Button	This control is used to cancel the creation of the supplier									
- The administrator clicks the “Okay” button - The use case is terminated											
ALT Step 8: The administrator clicks on the cancel button The use case is terminated											
CONCLUSION:	This use case concludes when a product has been added on the system										
POST-CONDITION:	A product has been updated in the Product table in the database										
BUSINESS RULES	Only the administrator has access to this function.										
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None										

ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 20 - 4.4 Delete Product

PRODUCT SUBSYSTEM					
AUTHORS (s): Thenjiwe Ntsonda		DATE: 18/04/2022			
VERSION: 1.0		LAST REVIEW DATE: 18/04/2022			
USE CASE NAME:	Delete Product	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒			
USE CASE ID:	4.4				
PRIORITY:	Medium				
SOURCE:	ByteXcom Requirements List				
PRIMARY BUSINESS ACTOR	• Administrator				
PRIMARY SYSTEM ACTOR	• None				
OTHER PARTICIPATING ACTORS:	• None				
OTHER INTERESTED STAKEHOLDERS:	• None				
DESCRIPTION:	This use case describes the process in which the administrator wishes to delete a product on the system. The use case begins when the administrator selects the desired product and selects the option to delete it. The system then deletes the selected product from the database				
PRE-CONDITION:	• The administrator must be logged onto the system.				
TRIGGER:	The use case is triggered when the administrator wants to delete a product existing in the product catalogue.				
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE			
	MANUAL ACTION	AUTOMATED ACTION			
	Step 1: The administrator wishes to delete a product		Step 2: The system invokes use case 4.2 Search Product		
	Step 3: The administrator clicks on the Delete button		Step 4: The system displays a message box with the following controls:		
			Control Name	Control Type	Notes
			Confirmation required	Label	This control is used to display the type of message box displayed
		Are you sure you want to delete this product?	Label	This control is used to request confirmation	

					of the deletion of the product
			Confirm	Button	This control is used to confirm the deletion of the product details
			Cancel	Button	This control is used to close the message box deletion of the product
	Step 5: The system administrator clicks on the Confirm button [ALT]		Step 6: The system uses an SQL READ query to check if the Product has an association with Order Table. [ALT]		
			Details to be read	Attributes in the table	Notes
			Product ID	ProductId	
			Step 7: The system deletes the selected product from the Product entity in the database using a method in a .NET controller and an SQL DELETE query.		
			Details to be deleted	Attributes in the table	Notes
			ProductId	ProductId	
			Name	Name	
			Description	Description	
			PictureUrl	PictureUrl	
			Customer Price	CustomerPrice	
			Reseller Price	ResellerPrice	
			Quantity On Hand	QuantityOnHand	
			Threshold Value	ThresholdValue	
			Step 7: The system redirects the system administrator to the Product screen and displays a message box with the following controls:		
			Control Name	Control Name	Notes
			Checkbox	Icon	This control is used to indicate a successful deletion
			"Product Deleted Successfully"	Label	This control is used to indicate a successful deletion

ALTERNATE COURSES:	ALT Step 5: The administrator does not want to delete the product anymore - The administrator clicks on the cancel button - The use case is terminated		
	ALT Step 6: The product cannot be deleted as it has an association with the Order Table The system redirects the administrator and displays a bootstrap alert notification with the following controls:		
	Control Name	Control Name	Notes
	"Product cannot be deleted"	Label	This control is used to indicate an unsuccessful deletion
	The use case is terminated		
CONCLUSION:	This use case concludes when a product has been deleted on the system		
POST-CONDITION:	The product has been removed from the system database.		
BUSINESS RULES	Only the administrator has access to this function.		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

Table 21 - 4.5 Add Product Category

PRODUCT CATEGORY SUBSYSTEM														
AUTHORS (s): Thenjiwe Ntsonda		DATE: 18/04/2022												
VERSION: 1.0		LAST REVIEW DATE: 21/08/2022												
USE CASE NAME:	Add Product category	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒												
USE CASE ID:	4.5													
PRIORITY:	High													
SOURCE:	ByteXcom Requirements List													
PRIMARY BUSINESS ACTOR	• Administrator													
PRIMARY SYSTEM ACTOR	• None													
OTHER PARTICIPATING ACTORS:	• None													
OTHER INTERESTED STAKEHOLDERS:	• None													
DESCRIPTION:	This use case describes the process in which the administrator wishes to add a new product category to the system. The use case begins when the administrator adds the new product category. The administrator will add all the pertinent information regarding the product category and save it to the business database.													
PRE-CONDITION:	• The administrator must be logged onto the system.													
TRIGGER:	The use case is triggered when the administrator wants to add a new product category.													
TYPICAL COURSE OF EVENTS:	SYSTEM RESPONSE													
	ACTOR ACTION	MANUAL ACTION	AUTOMATED ACTION											
				Step 2: The system retrieves the product category information from the following attributes and tables using a method in a .NET Controller and an SQL READ Query: From the ProductCategory table: <table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Name</td><td>Name</td></tr></table> And displays the Product Category Screen with the following controls: <table><tr><td>Control Name</td><td>Control Type</td><td>Notes</td></tr><tr><td>Product Category List</td><td>Heading</td><td>Page main heading</td></tr></table>	Details to be read	Attributes in the table	Name	Name	Control Name	Control Type	Notes	Product Category List	Heading	Page main heading
				Details to be read	Attributes in the table									
				Name	Name									
Control Name	Control Type			Notes										
Product Category List	Heading	Page main heading												
Step 1: The administrator navigates to the Product Category Screen under the Product category tab on the side bar														

			Question Mark Icon	Button	When clicked, this control will open a pdf which contains the help document
			Create New Product Category	Button	When clicked, this control will direct the user to the Create product category page
			Search bar	Textbox	This control will be used when the user inserts the search parameters
			Search icon	Button	This control will be used to activate the search
			Results table	Table	This control will be used to display all the product categories that exist on the system
			Name	Label	This control will be used as a heading to display the Name for all product categories
			Action - Update	Button	This control will be placed in a row, with the delete button, under the "Action" heading for each entry in the Supplier table. The button will direct the user to the Update Product category screen
			Action - Delete	Button	This control will be placed in a row, with the delete

					button, under the “Action” heading for each entry in the Supplier table. The button will direct the user to the Delete Product category Details screen
			Showing “” to “” of “” entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database
			“Previous” and “Next”	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on

			<table><tr><td>Name</td><td>Textbox</td><td>This control will be used by the administrator to insert a name of the product category</td></tr><tr><td>Create</td><td>Button</td><td>This control will be used to confirm the addition of the product category</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the addition of the product category</td></tr></table>	Name	Textbox	This control will be used by the administrator to insert a name of the product category	Create	Button	This control will be used to confirm the addition of the product category	Cancel	Button	This control will be used to cancel the addition of the product category
	Name	Textbox	This control will be used by the administrator to insert a name of the product category									
	Create	Button	This control will be used to confirm the addition of the product category									
	Cancel	Button	This control will be used to cancel the addition of the product category									
			Note: All controls are enabled by default unless stated otherwise.									
	Step 5: The system administrator enters the details and clicks on the Create button [ALT]		Step 6: The system captures and validates the entered details against the Product category table [ALT] <table><tr><th>Details to be validated</th><th>Attributes in the product category table</th><th>Notes</th></tr><tr><td>Name</td><td>Name</td><td>Required, Unique</td></tr></table>	Details to be validated	Attributes in the product category table	Notes	Name	Name	Required, Unique			
	Details to be validated	Attributes in the product category table	Notes									
	Name	Name	Required, Unique									
			Step 7: The system uses an SQL READ query to check if the product category already exists in the ProductCategory table using the following attributes [ALT] <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Name</td><td>Name</td></tr></table>	Details to be read	Attributes in the table	Name	Name					
	Details to be read	Attributes in the table										
Name	Name											
		Step 8: The system saves the information into the Product category table using an INSERT SQL query: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Product CategoryId</td><td>Product CategoryId</td><td>A unique value generated by the system and saved to the database.</td></tr><tr><td>Name</td><td>Name</td><td>Required</td></tr></table>	Details to be saved	Attributes in the table	Notes	Product CategoryId	Product CategoryId	A unique value generated by the system and saved to the database.	Name	Name	Required	
Details to be saved	Attributes in the table	Notes										
Product CategoryId	Product CategoryId	A unique value generated by the system and saved to the database.										
Name	Name	Required										

			Step 9: The system redirects the system administrator to the Product Category Screen and displays a message box with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Checkbox</td><td>Icon</td><td>This control is used to indicate a successful creation</td></tr><tr><td>"Product Category Created Successfully"</td><td>Label</td><td>This control is used to indicate a successful creation</td></tr></table>	Control Name	Control Name	Notes	Checkbox	Icon	This control is used to indicate a successful creation	"Product Category Created Successfully"	Label	This control is used to indicate a successful creation
Control Name	Control Name	Notes										
Checkbox	Icon	This control is used to indicate a successful creation										
"Product Category Created Successfully"	Label	This control is used to indicate a successful creation										
ALTERNATE COURSES:	ALT Step 6: The administrator does not want to add a product category anymore - The administrator clicks on the cancel button - The use case is terminated											
	ALT Step 7: The Supplier already exists on the system - The system displays an error modal dialogue with the following controls											
	<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Product category already exists!</td><td>Label</td><td>This control is used to inform the administrator that the supplier already exists</td></tr><tr><td>Okay</td><td>Button</td><td>This control is used to cancel the creation of the supplier</td></tr></table>	Control Name	Control Name	Notes	Product category already exists!	Label	This control is used to inform the administrator that the supplier already exists	Okay	Button	This control is used to cancel the creation of the supplier		
	Control Name	Control Name	Notes									
	Product category already exists!	Label	This control is used to inform the administrator that the supplier already exists									
Okay	Button	This control is used to cancel the creation of the supplier										
- The administrator clicks the "Okay" button - The use case is terminated												
ALT Step 8A: The details entered are not in the correct format or the required fields are empty The system displays a validation error message box with the following controls												
	<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Name</td><td>Textbox</td><td>This control will be outlined in red to show a validation error</td></tr><tr><td>Name is required</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Name textbox</td></tr></table>	Control Name	Control Name	Notes	Name	Textbox	This control will be outlined in red to show a validation error	Name is required	Label	This control will be used as a label to indicate a validation error in the Name textbox		
Control Name	Control Name	Notes										
Name	Textbox	This control will be outlined in red to show a validation error										
Name is required	Label	This control will be used as a label to indicate a validation error in the Name textbox										
	Return to step 3											
CONCLUSION:	This use case concludes when a product category has been added on the system											
POST-CONDITION:	A new user has been added to the Product category table in the database											
BUSINESS RULES	Only the administrator has access to this function.											
IMPLEMENTATION CONSTRAINTS	The system is offline, and the system administrator will not be able to add a new product category.											

AND SPECIFICATIONS	
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 22 - 4.6 Search Product Category

PRODUCT SUBSYSTEM											
AUTHORS (s): Thenjiwe Ntsonda			DATE: 18/04/2022								
VERSION: 1.0			LAST REVIEW DATE: 21/08/2022								
USE CASE NAME:	Search Product category		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒								
USE CASE ID:	4.6										
PRIORITY:	Medium										
SOURCE:	ByteXcom Requirements List										
PRIMARY BUSINESS ACTOR	• Administrator										
PRIMARY SYSTEM ACTOR	• None										
OTHER PARTICIPATING ACTORS:	• None										
OTHER INTERESTED STAKEHOLDERS:	• None										
DESCRIPTION:	This use case describes the process in which the administrator searches for product categories on the system. The system responds by displaying the information relating to the search query.										
PRE-CONDITION:	• The administrator must be logged onto the system.										
TRIGGER:	This use case is called when the administrator wants to search for a product category on the system.										
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE									
		MANUAL ACTION	AUTOMATED ACTION								
	Step 1: The administrator wishes to search for a product category on the system.										
	Step 2: The administrator navigates to the Product Category Screen under the Product category tab on the side bar		Step 3: The system retrieves following information using an SQL READ QUERY From the ProductCategory table: <table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Name</td><td>Name</td></tr></table> And displays the Product Category Screen with the following controls: <table><tr><td>Control Name</td><td>Control Type</td><td>Notes</td></tr></table>		Details to be read	Attributes in the table	Name	Name	Control Name	Control Type	Notes
	Details to be read	Attributes in the table									
	Name	Name									
Control Name	Control Type	Notes									

			Product categories List	Heading	Page main heading
			Question Mark Icon	Button	When clicked, this control will open a pdf which contains the help document
			Create New Product Category	Button	When clicked, this control will direct the user to the Add product category page
			Search bar	Textbox	This control will be used when the user inserts the search parameters
			Search icon	Button	This control will be used to activate the search
			Results table	Table	This control will be used to display all the product categories that exist on the system
			Name	Label	This control will be used as a heading to display the Name for all product categories
			Action	Table Heading	This control will be used as a heading to display the actions the administrator can perform with each entry – Edit and Delete
			Action - Update	Button	This control will be placed in a row, with the delete button, under the “Action” heading for each entry in the Supplier table. The button will direct the

				user to the Update Product category screen	
			Action - Delete	Button	This control will be placed in a row, with the delete button, under the "Action" heading for each entry in the Supplier table. The button will direct the user to the Delete Product category Details screen
			Showing "" to "" of "" entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database
			"Previous" and "Next"	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on
	Step 4: The administrator enters their search query into the search textbox.		Step 5: The system searches the ProductCategory table and displays the product category(s) matching the search query in the Product Category Screen [ALT]		
ALT Step 5: The search query entered does not match any records existing in the Product category entity. The system displays a "No Records Found" message in the Product category Screen and on the results table.					

CONCLUSION:	This use case concludes when the required product category is displayed on the screen.
POST-CONDITION:	The administrator selects the required product category from the displayed results.
BUSINESS RULES	• Only the administrator has access to this function.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 23 - 4.7 Update Product Category

PRODUCT SUBSYSTEM							
AUTHORS (s): Thenjiwe Ntsonda		DATE: 18/04/2022					
VERSION: 1.0		LAST REVIEW DATE: 18/04/2022					
USE CASE NAME:	Update Product category		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒				
USE CASE ID:	4.7						
PRIORITY:	Medium						
SOURCE:	ByteXcom Requirements List						
PRIMARY BUSINESS ACTOR	• Administrator						
PRIMARY SYSTEM ACTOR	• None						
OTHER PARTICIPATING ACTORS:	• None						
OTHER INTERESTED STAKEHOLDERS:	• None						
DESCRIPTION:	This use case describes the process in which the administrator wishes to update. The use case begins when the administrator updates the product category. The administrator will update all the pertinent information regarding the product category and save it to the business database.						
PRE-CONDITION:	• The administrator must be logged onto the system.						
TRIGGER:	The use case is triggered when the administrator wants to update a product category.						
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE					
		MANUAL ACTION	AUTOMATED ACTION				
	Step 1: The administrator wishes to update a product category in the product category catalogue.		Step 2: The system invokes use case 4.6 Search Product category				
	Step 2: The administrator clicks on the Edit button		Step 3: The system retrieves the selected product category's information from the following attributes and tables using a method in a .NET Controller and an SQL READ Query: From the ProductCategory table: <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Name</td><td>Name</td></tr></table>		Details to be read	Attributes in the table	Name
Details to be read	Attributes in the table						
Name	Name						

		And displays the Update Product Category Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Update Product category</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Product category information</td><td>Heading</td><td>Page subheading</td></tr><tr><td>Name</td><td>Labe</td><td>This control will be used as a label to indicate the Name attribute of the product category record</td></tr><tr><td>Name</td><td>Textbox</td><td>This control will be used by the administrator to update a name of the product category</td></tr><tr><td>Update</td><td>Button</td><td>This control will be used to confirm the addition of the product category</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the edit of the product category</td></tr></table> Note: All controls are enabled by default unless stated otherwise.	Control Name	Control Type	Notes	Update Product category	Heading	Page main heading	Product category information	Heading	Page subheading	Name	Labe	This control will be used as a label to indicate the Name attribute of the product category record	Name	Textbox	This control will be used by the administrator to update a name of the product category	Update	Button	This control will be used to confirm the addition of the product category	Cancel	Button	This control will be used to cancel the edit of the product category
	Control Name	Control Type	Notes																				
	Update Product category	Heading	Page main heading																				
	Product category information	Heading	Page subheading																				
	Name	Labe	This control will be used as a label to indicate the Name attribute of the product category record																				
	Name	Textbox	This control will be used by the administrator to update a name of the product category																				
	Update	Button	This control will be used to confirm the addition of the product category																				
Cancel	Button	This control will be used to cancel the edit of the product category																					
Step 4: The system administrator updates the product category information and clicks on the Update button [ALT]		Step 5: The system captures and validates the entered details against the ProductCategory table [ALT] <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Name</td><td>Name</td><td>Required, Unique,</td></tr></table>	Details to be read	Attributes in the table	Notes	Name	Name	Required, Unique,															
Details to be read	Attributes in the table	Notes																					
Name	Name	Required, Unique,																					
		Step 6: The system uses an SQL READ query to check if the product category already exists in the ProductCategory table using the following attributes [ALT] <table><tr><th>Details to be updated</th><th>Attributes in the table</th><th>Notes</th></tr></table>	Details to be updated	Attributes in the table	Notes																		
Details to be updated	Attributes in the table	Notes																					

			<table><tr><td>Name</td><td colspan="2">Name</td></tr></table>	Name	Name													
	Name	Name																
			Step 7: The system displays a message box with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Confirmation required</td><td>Label</td><td>This control is used to display the type of message box displayed</td></tr><tr><td>Confirm updated product category details?</td><td>Label</td><td>This control is used to request confirmation on the updated product category details</td></tr><tr><td>Confirm</td><td>Button</td><td>This control is used to confirm the updating of the product category details</td></tr><tr><td>Cancel</td><td>Button</td><td>This control is used to close the message box for the user to edit the product category details</td></tr></table>	Control Name	Control Type	Notes	Confirmation required	Label	This control is used to display the type of message box displayed	Confirm updated product category details?	Label	This control is used to request confirmation on the updated product category details	Confirm	Button	This control is used to confirm the updating of the product category details	Cancel	Button	This control is used to close the message box for the user to edit the product category details
	Control Name	Control Type	Notes															
	Confirmation required	Label	This control is used to display the type of message box displayed															
	Confirm updated product category details?	Label	This control is used to request confirmation on the updated product category details															
	Confirm	Button	This control is used to confirm the updating of the product category details															
	Cancel	Button	This control is used to close the message box for the user to edit the product category details															
	Step 8: The system administrator clicks on the Confirm button [ALT]		Step 9: The system saves the information into the Product category table using an UPDATE SQL query: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Name</td><td>Name</td><td>Required, Unique,</td></tr></table>	Details to be saved	Attributes in the table	Notes	Name	Name	Required, Unique,									
	Details to be saved	Attributes in the table	Notes															
Name	Name	Required, Unique,																

			Step 10: The system redirects the system administrator to the Product Category Screen and displays a message box with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Checkbox</td><td>Icon</td><td>This control is used to indicate a successful update</td></tr><tr><td>"Product Category Created Successfully"</td><td>Label</td><td>This control is used to indicate a successful update</td></tr></table>	Control Name	Control Name	Notes	Checkbox	Icon	This control is used to indicate a successful update	"Product Category Created Successfully"	Label	This control is used to indicate a successful update
Control Name	Control Name	Notes										
Checkbox	Icon	This control is used to indicate a successful update										
"Product Category Created Successfully"	Label	This control is used to indicate a successful update										
ALTERNATE COURSES:	ALT Step 4: The administrator does not want to edit the product category anymore - The administrator clicks on the cancel button - The use case is terminated											
	ALT Step 5: The details entered are not in the correct format or the required fields are empty The system displays a validation error message box with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Name</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>Name is required</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Name textbox</td></tr></table>			Control Name	Control Name	Notes	Name	Textbox	This control will be outline in red to show a validation error	Name is required	Label	This control will be used as a label to indicate a validation error in the Name textbox
	Control Name	Control Name	Notes									
	Name	Textbox	This control will be outline in red to show a validation error									
	Name is required	Label	This control will be used as a label to indicate a validation error in the Name textbox									
ALT Step 6: The Product Category already exists on the system - The system displays an error modal dialogue with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Product category already exists!</td><td>Label</td><td>This control is used to inform the administrator that the supplier already exists</td></tr><tr><td>Okay</td><td>Button</td><td>This control is used to cancel the creation of the supplier</td></tr></table> - The administrator clicks the "Okay" button - The use case is terminated			Control Name	Control Name	Notes	Product category already exists!	Label	This control is used to inform the administrator that the supplier already exists	Okay	Button	This control is used to cancel the creation of the supplier	
Control Name	Control Name	Notes										
Product category already exists!	Label	This control is used to inform the administrator that the supplier already exists										
Okay	Button	This control is used to cancel the creation of the supplier										
ALT Step 8: The administrator clicks on the cancel button The use case is terminated												
CONCLUSION:	This use case concludes when a product category has been added on the system											
POST-CONDITION:	A product category has been updated in the Product category table in the database											

BUSINESS RULES	• Only the administrator has access to this function.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 24 - 4.8 Delete Product Category

PRODUCT SUBSYSTEM								
AUTHORS (s): Thenjiwe Ntsonda		DATE: 18/04/2022						
VERSION: 1.0		LAST REVIEW DATE: 18/04/2022						
USE CASE NAME:	Delete Product Category	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒						
USE CASE ID:	4.8							
PRIORITY:	Medium							
SOURCE:	ByteXcom Requirements List							
PRIMARY BUSINESS ACTOR	• Administrator							
PRIMARY SYSTEM ACTOR	• None							
OTHER PARTICIPATING ACTORS:	• None							
OTHER INTERESTED STAKEHOLDERS:	• None							
DESCRIPTION:	This use case describes the process in which the administrator wishes to delete a product category to the system. The use case begins when the administrator selects the product category they wish to delete. This product will then be removed from the system.							
PRE-CONDITION:	• The administrator must be logged onto the system.							
TRIGGER:	The use case is triggered when the administrator wants to delete a product category.							
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE						
		MANUAL ACTION	AUTOMATED ACTION					
	Step 1: The administrator wishes to delete a product category		Step 2: The system invokes use case 4.6 Search Product category					
			Step 3: The system retrieves the product category information from the following attributes and tables using a method in a .NET Controller and an SQL READ Query: From the ProductCategory table: <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Name</td><td>Name</td></tr></table>		Details to be read	Attributes in the table	Name	Name
	Details to be read	Attributes in the table						
Name	Name							

		And displays the Delete Product Category Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Delete Product Category</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Name</td><td>Label</td><td>This control will be used as a label to indicate the Name attribute of the product category record</td></tr><tr><td>Name</td><td>Label</td><td>This control will be used by the administrator to display the name of the product category</td></tr><tr><td>Delete</td><td>Button</td><td>This control will be used to confirm the addition of the product category</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the addition of the product category</td></tr></table>	Control Name	Control Type	Notes	Delete Product Category	Heading	Page main heading	Name	Label	This control will be used as a label to indicate the Name attribute of the product category record	Name	Label	This control will be used by the administrator to display the name of the product category	Delete	Button	This control will be used to confirm the addition of the product category	Cancel	Button	This control will be used to cancel the addition of the product category
	Control Name	Control Type	Notes																	
	Delete Product Category	Heading	Page main heading																	
	Name	Label	This control will be used as a label to indicate the Name attribute of the product category record																	
	Name	Label	This control will be used by the administrator to display the name of the product category																	
	Delete	Button	This control will be used to confirm the addition of the product category																	
	Cancel	Button	This control will be used to cancel the addition of the product category																	
	Step 4: The administrator clicks on the Delete button	Step 5: The system displays a message box with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Confirmation required</td><td>Label</td><td>This control is used to display the type of message box displayed</td></tr><tr><td>Are you sure you want to delete this product category?</td><td>Label</td><td>This control is used to request confirmation of the deletion of the product category</td></tr></table>	Control Name	Control Type	Notes	Confirmation required	Label	This control is used to display the type of message box displayed	Are you sure you want to delete this product category?	Label	This control is used to request confirmation of the deletion of the product category									
	Control Name	Control Type	Notes																	
	Confirmation required	Label	This control is used to display the type of message box displayed																	
Are you sure you want to delete this product category?	Label	This control is used to request confirmation of the deletion of the product category																		

			Confirm	Button	This control is used to confirm the deletion of the product category details
			Cancel	Button	This control is used to close the message box deletion of the product category
	Step 6: The system administrator clicks on the Confirm button [ALT]		Step 7: The system uses an SQL READ query to check if the Product Category has an association with Product Table. [ALT]		
			Details to be read	Attributes in the table	
			ProductCategoryId	ProductCategoryId	
			Step 8: The system deletes the selected product category from the ProductCategory entity in the database using a method in a .NET controller and an SQL DELETE query.		
			Details to be deleted	Attributes in the table	
			ProductCategoryId	ProductCategoryId	
			Name	Name	
			Step 9: The system redirects the system administrator to the Product category screen and displays a message box with the following controls:		
Control Name			Control Name	Notes	
Checkbox			Icon	This control is used to indicate a successful deletion	
"Product category Deleted Successfully"			Label	This control is used to indicate a successful deletion	
ALTERNATE COURSES:	ALT Step 5: The administrator does not want to delete the product category anymore - The administrator clicks on the cancel button - The use case is terminated				

	ALT Step 6: The product category cannot be deleted as it has an association with the Product Table The system redirects the administrator and displays a bootstrap alert notification with the following controls:		
	Control Name	Control Name	Notes
	"Product category cannot be deleted"	Label	This control is used to indicate an unsuccessful deletion
	The use case is terminated		
CONCLUSION:	This use case concludes when a product category has been deleted on the system		
POST-CONDITION:	The product category has been removed from the system database.		
BUSINESS RULES	Only the administrator has access to this function.		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

Table 25 - 4.9 Add Product Brand

PRODUCT SUBSYSTEM														
AUTHORS (s): Thenjiwe Ntsonda		DATE: 18/04/2022												
VERSION: 1.0		LAST REVIEW DATE: 21/08/2022												
USE CASE NAME:	Add Product brand	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒												
USE CASE ID:	4.9													
PRIORITY:	High													
SOURCE:	ByteXcom Requirements List													
PRIMARY BUSINESS ACTOR	• Administrator													
PRIMARY SYSTEM ACTOR	• None													
OTHER PARTICIPATING ACTORS:	• None													
OTHER INTERESTED STAKEHOLDERS:	• None													
DESCRIPTION:	This use case describes the process in which the administrator wishes to add a new product brand to the system. The use case begins when the administrator adds the new product brand. The administrator will add all the pertinent information regarding the product brand and save it to the business database.													
PRE-CONDITION:	• The administrator must be logged onto the system.													
TRIGGER:	The use case is triggered when the administrator wants to add a new product brand.													
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE												
		MANUAL ACTION	AUTOMATED ACTION											
	Step 1: The administrator navigates to the Product brand Screen under the Product brand tab on the side bar		Step 2: The system retrieves the product brand information from the following attributes and tables using a method in a .NET Controller and an SQL READ Query: From the ProductBrand table: <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Name</td><td>Name</td></tr></table> And displays the Product brand Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Product brand List</td><td>Heading</td><td>Page main heading</td></tr></table>		Details to be read	Attributes in the table	Name	Name	Control Name	Control Type	Notes	Product brand List	Heading	Page main heading
			Details to be read	Attributes in the table										
			Name	Name										
			Control Name	Control Type	Notes									
			Product brand List	Heading	Page main heading									

			Question Mark Icon	Button	When clicked, this control will open a pdf which contains the help document
			Create New Product brand	Button	When clicked, this control will direct the user to the Create product brand page
			Search bar	Textbox	This control will be used when the user inserts the search parameters
			Search icon	Button	This control will be used to activate the search
			Results table	Table	This control will be used to display all the product categories that exist on the system
			Name	Label	This control will be used as a heading to display the Name for all product categories
			Action - Update	Button	This control will be placed in a row, with the delete button, under the "Action" heading for each entry in the Supplier table. The button will direct the user to the Update Product brand screen
			Action - Delete	Button	This control will be placed in a row, with the delete button,

					under the “Action” heading for each entry in the Supplier table. The button will direct the user to the Delete Product brand Details screen
			Showing “” to “” of “” entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database
			“Previous” and “Next”	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on
	Step 3: The system administrator clicks on the Create New Product brand button		Step 4: The system displays the Create Product brand screen with the following controls:		
			Control Name	Control Type	Notes
			Create Product brand	Heading	Page main heading
			Product brand information	Heading	Page subheading
			Name	Label	This control will be used as a label to indicate the Name attribute of the product brand record
			Name	Textbox	This control will be used

			<table><tr><td></td><td></td><td>by the administrator to insert a name of the product brand</td></tr><tr><td>Create</td><td>Button</td><td>This control will be used to confirm the addition of the product brand</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the addition of the product brand</td></tr></table>			by the administrator to insert a name of the product brand	Create	Button	This control will be used to confirm the addition of the product brand	Cancel	Button	This control will be used to cancel the addition of the product brand
			by the administrator to insert a name of the product brand									
	Create	Button	This control will be used to confirm the addition of the product brand									
	Cancel	Button	This control will be used to cancel the addition of the product brand									
			Note: All controls are enabled by default unless stated otherwise.									
Step 5: The system administrator enters the details and clicks on the Create button [ALT]		Step 6: The system captures and validates the entered details against the Product brand table [ALT] <table><tr><th>Details to be validated</th><th>Attributes in the product brand table</th><th>Notes</th></tr><tr><td>Name</td><td>Name</td><td>Required, Unique</td></tr></table>	Details to be validated	Attributes in the product brand table	Notes	Name	Name	Required, Unique				
Details to be validated	Attributes in the product brand table	Notes										
Name	Name	Required, Unique										
		Step 7: The system uses an SQL READ query to check if the product brand already exists in the ProductBrand table using the following attributes [ALT] <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Name</td><td>Name</td><td></td></tr></table>	Details to be read	Attributes in the table	Notes	Name	Name					
Details to be read	Attributes in the table	Notes										
Name	Name											
		Step 8: The system saves the information into the Product brand table using an INSERT SQL query: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Product brandId</td><td>Product brandId</td><td>A unique value generated by the system and saved to the database.</td></tr><tr><td>Name</td><td>Name</td><td>Required</td></tr></table>	Details to be saved	Attributes in the table	Notes	Product brandId	Product brandId	A unique value generated by the system and saved to the database.	Name	Name	Required	
Details to be saved	Attributes in the table	Notes										
Product brandId	Product brandId	A unique value generated by the system and saved to the database.										
Name	Name	Required										
		Step 9: The system redirects the system administrator to the Product brand Screen and displays a message box with the following controls:										

			Control Name	Control Name	Notes
			Checkbox	Icon	This control is used to indicate a successful creation
			"Product brand Created Successfully"	Label	This control is used to indicate a successful creation
ALTERNATE COURSES:	ALT Step 6: The administrator does not want to add a product brand anymore - The administrator clicks on the cancel button - The use case is terminated				
	ALT Step 7: The Supplier already exists on the system - The system displays an error modal dialogue with the following controls				
	Control Name		Control Name	Notes	
	Product brand already exists!		Label	This control is used to inform the administrator that the supplier already exists	
	Okay		Button	This control is used to cancel the creation of the supplier	
	- The administrator clicks the "Okay" button - The use case is terminated				
	ALT Step 8A: The details entered are not in the correct format or the required fields are empty The system displays a validation error message box with the following controls				
Control Name		Control Name	Notes		
Name		Textbox	This control will be outlined in red to show a validation error		
Name is required		Label	This control will be used as a label to indicate a validation error in the Name textbox		
Return to step 3					
CONCLUSION:	This use case concludes when a product brand has been added on the system				
POST-CONDITION:	A new user has been added to the Product brand table in the database				
BUSINESS RULES	Only the administrator has access to this function.				
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	The system is offline, and the system administrator will not be able to add a new product brand.				
ASSUMPTIONS:	None				
OPEN ISSUES:	None				

Table 26 - 4.10 Search Product Brand

PRODUCT SUBSYSTEM													
AUTHORS (s): Thenjiwe Ntsonda		DATE: 18/04/2022											
VERSION: 1.0		LAST REVIEW DATE: 21/08/2022											
USE CASE NAME:	Search Product brand	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒											
USE CASE ID:	4.10												
PRIORITY:	Medium												
SOURCE:	ByteXcom Requirements List												
PRIMARY BUSINESS ACTOR	• Administrator												
PRIMARY SYSTEM ACTOR	• None												
OTHER PARTICIPATING ACTORS:	• None												
OTHER INTERESTED STAKEHOLDERS:	• None												
DESCRIPTION:	This use case describes the process in which the administrator searches for product categories on the system. The system responds by displaying the information relating to the search query.												
PRE-CONDITION:	• The administrator must be logged onto the system.												
TRIGGER:	This use case is called when the administrator wants to search for a product brand on the system.												
TYPICAL COURSE OF EVENTS:	SYSTEM RESPONSE												
	ACTOR ACTION	MANUAL ACTION	AUTOMATED ACTION										
	Step 1: The administrator wishes to search for a product brand on the system.		Step 3: The system retrieves following information using an SQL READ QUERY From the ProductBrand table: <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Name</td><td>Name</td></tr></table> And displays the Product brand Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Product categories List</td><td>Heading</td><td>Page main heading</td></tr></table>	Details to be read	Attributes in the table	Name	Name	Control Name	Control Type	Notes	Product categories List	Heading	Page main heading
Details to be read	Attributes in the table												
Name	Name												
Control Name	Control Type	Notes											
Product categories List	Heading	Page main heading											

			Question Mark Icon	Button	When clicked, this control will open a pdf which contains the help document
			Create New Product brand	Button	When clicked, this control will direct the user to the Add product brand page
			Search bar	Textbox	This control will be used when the user inserts the search parameters
			Search icon	Button	This control will be used to activate the search
			Results table	Table	This control will be used to display all the product categories that exist on the system
			Name	Label	This control will be used as a heading to display the Name for all product categories
			Action	Table Heading	This control will be used as a heading to display the actions the administrator can perform with each entry – Edit and Delete
			Action - Update	Button	This control will be placed in a row, with the delete button, under the “Action” heading for each entry in the Supplier table. The button will direct the user to the Update Product

					brand screen
			Action - Delete	Button	This control will be placed in a row, with the delete button, under the "Action" heading for each entry in the Supplier table. The button will direct the user to the Delete Product brand Details screen
			Showing "" to "" of "" entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database
			"Previous" and "Next"	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on
	Step 4: The administrator enters their search query into the search textbox.		Step 5: The system searches the ProductBrand table and displays the product brand(s) matching the search query in the Product brand Screen [ALT]		
	ALT Step 5: The search query entered does not match any records existing in the Product brand entity. The system displays a "No Records Found" message in the Product Brand Screen and on the results table.				
	CONCLUSION:	This use case concludes when the required product brand is displayed on the screen.			

POST-CONDITION:	The administrator selects the required product brand from the displayed results.
BUSINESS RULES	• Only the administrator has access to this function.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 27 - 4.11 Update Product Brand

PRODUCT SUBSYSTEM							
AUTHORS (s): Thenjiwe Ntsonda		DATE: 18/04/2022					
VERSION: 1.0		LAST REVIEW DATE: 18/04/2022					
USE CASE NAME:	Update Product Brand		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒				
USE CASE ID:	4.11						
PRIORITY:	Medium						
SOURCE:	ByteXcom Requirements List						
PRIMARY BUSINESS ACTOR	• Administrator						
PRIMARY SYSTEM ACTOR	• None						
OTHER PARTICIPATING ACTORS:	• None						
OTHER INTERESTED STAKEHOLDERS:	• None						
DESCRIPTION:	This use case describes the process in which the administrator wishes to update. The use case begins when the administrator updates the product brand. The administrator will update all the pertinent information regarding the product brand and save it to the business database.						
PRE-CONDITION:	• The administrator must be logged onto the system.						
TRIGGER:	The use case is triggered when the administrator wants to update a product brand.						
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE					
		MANUAL ACTION	AUTOMATED ACTION				
	Step 1: The administrator wishes to update a product brand in the product brand catalogue.		Step 2: The system invokes use case 4.6 Search Product brand				
	Step 2: The administrator clicks on the Edit button		Step 3: The system retrieves the selected product brand’s information from the following attributes and tables using a method in a .NET Controller and an SQL READ Query: From the ProductBrand table: <table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Name</td><td>Name</td></tr></table>	Details to be read	Attributes in the table	Name	Name
	Details to be read	Attributes in the table					
Name	Name						

		And displays the Update Product brand Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Update Product brand</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Product brand information</td><td>Heading</td><td>Page subheading</td></tr><tr><td>Name</td><td>Labe</td><td>This control will be used as a label to indicate the Name attribute of the product brand record</td></tr><tr><td>Name</td><td>Textbox</td><td>This control will be used by the administrator to update a name of the product brand</td></tr><tr><td>Update</td><td>Button</td><td>This control will be used to confirm the addition of the product brand</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the edit of the product brand</td></tr></table> Note: All controls are enabled by default unless stated otherwise.	Control Name	Control Type	Notes	Update Product brand	Heading	Page main heading	Product brand information	Heading	Page subheading	Name	Labe	This control will be used as a label to indicate the Name attribute of the product brand record	Name	Textbox	This control will be used by the administrator to update a name of the product brand	Update	Button	This control will be used to confirm the addition of the product brand	Cancel	Button	This control will be used to cancel the edit of the product brand
Control Name	Control Type	Notes																					
Update Product brand	Heading	Page main heading																					
Product brand information	Heading	Page subheading																					
Name	Labe	This control will be used as a label to indicate the Name attribute of the product brand record																					
Name	Textbox	This control will be used by the administrator to update a name of the product brand																					
Update	Button	This control will be used to confirm the addition of the product brand																					
Cancel	Button	This control will be used to cancel the edit of the product brand																					
Step 4: The system administrator updates the product brand information and clicks on the Update button [ALT]		Step 5: The system captures and validates the entered details against the ProductBrand table [ALT] <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Name</td><td>Name</td><td>Required, Unique,</td></tr></table>	Details to be read	Attributes in the table	Notes	Name	Name	Required, Unique,															
Details to be read	Attributes in the table	Notes																					
Name	Name	Required, Unique,																					
		Step 6: The system uses an SQL READ query to check if the product brand already exists in the ProductBrand table using the following attributes [ALT] <table><tr><th>Details to be updated</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Name</td><td>Name</td><td></td></tr></table>	Details to be updated	Attributes in the table	Notes	Name	Name																
Details to be updated	Attributes in the table	Notes																					
Name	Name																						

		Step 7: The system displays a message box with the following controls:															
		<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Confirmation required</td><td>Label</td><td>This control is used to display the type of message box displayed</td></tr><tr><td>Confirm updated product brand details?</td><td>Label</td><td>This control is used to request confirmation on the updated product brand details</td></tr><tr><td>Confirm</td><td>Button</td><td>This control is used to confirm the updating of the product brand details</td></tr><tr><td>Cancel</td><td>Button</td><td>This control is used to close the message box for the user to edit the product brand details</td></tr></table>	Control Name	Control Type	Notes	Confirmation required	Label	This control is used to display the type of message box displayed	Confirm updated product brand details?	Label	This control is used to request confirmation on the updated product brand details	Confirm	Button	This control is used to confirm the updating of the product brand details	Cancel	Button	This control is used to close the message box for the user to edit the product brand details
	Control Name	Control Type	Notes														
	Confirmation required	Label	This control is used to display the type of message box displayed														
	Confirm updated product brand details?	Label	This control is used to request confirmation on the updated product brand details														
Confirm	Button	This control is used to confirm the updating of the product brand details															
Cancel	Button	This control is used to close the message box for the user to edit the product brand details															
Step 8: The system administrator clicks on the Confirm button [ALT]		Step 9: The system saves the information into the Product brand table using an UPDATE SQL query:															
		<table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Name</td><td>Name</td><td>Required, Unique,</td></tr></table>	Details to be saved	Attributes in the table	Notes	Name	Name	Required, Unique,									
Details to be saved	Attributes in the table	Notes															
Name	Name	Required, Unique,															

			Step 10: The system redirects the system administrator to the Product brand Screen and displays a message box with the following controls:									
			<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Checkbox</td><td>Icon</td><td>This control is used to indicate a successful update</td></tr><tr><td>"Product brand Created Successfully"</td><td>Label</td><td>This control is used to indicate a successful update</td></tr></table>	Control Name	Control Name	Notes	Checkbox	Icon	This control is used to indicate a successful update	"Product brand Created Successfully"	Label	This control is used to indicate a successful update
	Control Name	Control Name	Notes									
	Checkbox	Icon	This control is used to indicate a successful update									
"Product brand Created Successfully"	Label	This control is used to indicate a successful update										
ALTERNATE COURSES:												
ALT Step 4: The administrator does not want to edit the product brand anymore - The administrator clicks on the cancel button - The use case is terminated												
ALT Step 5: The details entered are not in the correct format or the required fields are empty The system displays a validation error message box with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Name</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>Name is required</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Name textbox</td></tr></table>			Control Name	Control Name	Notes	Name	Textbox	This control will be outline in red to show a validation error	Name is required	Label	This control will be used as a label to indicate a validation error in the Name textbox	
Control Name	Control Name	Notes										
Name	Textbox	This control will be outline in red to show a validation error										
Name is required	Label	This control will be used as a label to indicate a validation error in the Name textbox										
ALT Step 6: The Product brand already exists on the system - The system displays an error modal dialogue with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Product brand already exists!</td><td>Label</td><td>This control is used to inform the administrator that the supplier already exists</td></tr><tr><td>Okay</td><td>Button</td><td>This control is used to cancel the creation of the supplier</td></tr></table> - The administrator clicks the "Okay" button - The use case is terminated			Control Name	Control Name	Notes	Product brand already exists!	Label	This control is used to inform the administrator that the supplier already exists	Okay	Button	This control is used to cancel the creation of the supplier	
Control Name	Control Name	Notes										
Product brand already exists!	Label	This control is used to inform the administrator that the supplier already exists										
Okay	Button	This control is used to cancel the creation of the supplier										
ALT Step 8: The administrator clicks on the cancel button The use case is terminated												
CONCLUSION:	This use case concludes when a product brand has been added on the system											
POST-CONDITION:	A product brand has been updated in the ProductBrand table in the database											

BUSINESS RULES	• Only the administrator has access to this function.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 28 - 4.12 Delete Product Brand

PRODUCT SUBSYSTEM							
AUTHORS (s): Thenjiwe Ntsonda		DATE: 18/04/2022					
VERSION: 1.0		LAST REVIEW DATE: 18/04/2022					
USE CASE NAME:	Delete Product Brand	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒					
USE CASE ID:	4.12						
PRIORITY:	Medium						
SOURCE:	ByteXcom Requirements List						
PRIMARY BUSINESS ACTOR	• Administrator						
PRIMARY SYSTEM ACTOR	• None						
OTHER PARTICIPATING ACTORS:	• None						
OTHER INTERESTED STAKEHOLDERS:	• None						
DESCRIPTION:	This use case describes the process in which the administrator wishes to delete a product brand to the system. The use case begins when the administrator selects the product brand they wish to delete. This product will then be removed from the system.						
PRE-CONDITION:	• The administrator must be logged onto the system.						
TRIGGER:	The use case is triggered when the administrator wants to delete a product brand.						
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE					
		MANUAL ACTION	AUTOMATED ACTION				
	Step 1: The administrator wishes to delete a product brand		Step 2: The system invokes use case 4.6 Search Product brand				
			Step 3: The system retrieves the product brand information from the following attributes and tables using a method in a .NET Controller and an SQL READ Query: From the ProductBrand table: <table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Name</td><td>Name</td></tr></table>	Details to be read	Attributes in the table	Name	Name
	Details to be read	Attributes in the table					
Name	Name						

		And displays the Delete Product brand Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Delete Product brand</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Name</td><td>Label</td><td>This control will be used as a label to indicate the Name attribute of the product brand record</td></tr><tr><td>Name</td><td>Label</td><td>This control will be used by the administrator to display the name of the product brand</td></tr><tr><td>Delete</td><td>Button</td><td>This control will be used to confirm the addition of the product brand</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the addition of the product brand</td></tr></table>	Control Name	Control Type	Notes	Delete Product brand	Heading	Page main heading	Name	Label	This control will be used as a label to indicate the Name attribute of the product brand record	Name	Label	This control will be used by the administrator to display the name of the product brand	Delete	Button	This control will be used to confirm the addition of the product brand	Cancel	Button	This control will be used to cancel the addition of the product brand
	Control Name	Control Type	Notes																	
	Delete Product brand	Heading	Page main heading																	
	Name	Label	This control will be used as a label to indicate the Name attribute of the product brand record																	
	Name	Label	This control will be used by the administrator to display the name of the product brand																	
	Delete	Button	This control will be used to confirm the addition of the product brand																	
	Cancel	Button	This control will be used to cancel the addition of the product brand																	
	Step 4: The administrator clicks on the Delete button	Step 5: The system displays a message box with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Confirmation required</td><td>Label</td><td>This control is used to display the type of message box displayed</td></tr><tr><td>Are you sure you want to delete this product brand?</td><td>Label</td><td>This control is used to request confirmation of the deletion of the product brand</td></tr><tr><td>Confirm</td><td>Button</td><td>This control is used to confirm the deletion of the</td></tr></table>	Control Name	Control Type	Notes	Confirmation required	Label	This control is used to display the type of message box displayed	Are you sure you want to delete this product brand?	Label	This control is used to request confirmation of the deletion of the product brand	Confirm	Button	This control is used to confirm the deletion of the						
	Control Name	Control Type	Notes																	
	Confirmation required	Label	This control is used to display the type of message box displayed																	
Are you sure you want to delete this product brand?	Label	This control is used to request confirmation of the deletion of the product brand																		
Confirm	Button	This control is used to confirm the deletion of the																		

					product brand details
			Cancel	Button	This control is used to close the message box deletion of the product brand
	Step 6: The system administrator clicks on the Confirm button [ALT]		Step 7: The system uses an SQL READ query to check if the Product brand has an association with Product Table. [ALT]		
			Details to be read	Attributes in the table	
			ProductBrandId	ProductBrandId	
		Step 8: The system deletes the selected product brand from the ProductBrand entity in the database using a method in a .NET controller and an SQL DELETE query.			
		Details to be deleted	Attributes in the table		
		ProductBrandId	ProductBrandId		
		Name	Name		
		Step 9: The system redirects the system administrator to the Product brand screen and displays a message box with the following controls:			
		Control Name	Control Name	Notes	
		Checkbox	Icon	This control is used to indicate a successful deletion	
		"Product brand Deleted Successfully"	Label	This control is used to indicate a successful deletion	
ALTERNATE COURSES:	ALT Step 5: The administrator does not want to delete the product brand anymore - The administrator clicks on the cancel button - The use case is terminated				
	ALT Step 6: The product brand cannot be deleted as it has an association with the Product Table The system redirects the administrator and displays a bootstrap alert notification with the following controls:				

	Control Name	Control Name	Notes
	"Product brand cannot be deleted"	Label	This control is used to indicate an unsuccessful deletion
	The use case is terminated		
CONCLUSION:	This use case concludes when a product brand has been deleted on the system		
POST-CONDITION:	The product brand has been removed from the system database.		
BUSINESS RULES	Only the administrator has access to this function.		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

Table 29 - 4.13 Create Return Reason

PRODUCT SUBSYSTEM									
AUTHORS (s): Ofhani Mungani			DATE: 18/04/2022						
VERSION: 1.0			LAST REVIEW DATE: 21/08/2022						
USE CASE NAME:	Create Return Reason		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒						
USE CASE ID:	4.13								
PRIORITY:	High								
SOURCE:	ByteXcom Requirements List								
PRIMARY BUSINESS ACTOR	• Administrator								
PRIMARY SYSTEM ACTOR	• None								
OTHER PARTICIPATING ACTORS:	• None								
OTHER INTERESTED STAKEHOLDERS:	• None								
DESCRIPTION:	This use case describes the process where the administrator wishes to add a new Return Reason onto the system. The use case begins with the administrator wishing to add a country. The administrator will capture the country’s details and save the information on the ReturnReason entity. The use case concludes when the system alerts the administrator with “Return Reason successfully added” message.								
PRE-CONDITION:	• The administrator must be logged onto the system.								
TRIGGER:	The use case is triggered when the administrator wants to add a new Country.								
TYPICAL COURSE OF EVENTS:	ACTOR ACTION		SYSTEM RESPONSE						
			MANUAL ACTION	AUTOMATED ACTION					
	Step 1: The administrator navigates to the Return Reason List Screen under the return reasons tab on the side navi bar		Step 2: The system retrieves the return reason information from the following attributes and tables using a method in a .NET Controller and an SQL READ Query: From the ReturnReason table:						
			<table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Name</td><td>CountryName</td></tr></table>		Details to be read	Attributes in the table	Name	CountryName	
			Details to be read	Attributes in the table					
Name	CountryName								
And displays the Return Reason Screen with the following controls:									
		<table><tr><td>Control Name</td><td>Control Type</td><td>Notes</td></tr><tr><td></td><td></td><td></td></tr></table>		Control Name	Control Type	Notes			
Control Name	Control Type	Notes							

			Return Reason List	Heading	Page main heading
			Question Mark Icon	Button	When clicked, this control will open a pdf which contains the help document
			Create New Country	Button	When clicked, this control will direct the user to the Create Return Reason page
			Search bar	Textbox	This control will be used when the user inserts the search parameters
			Search icon	Button	This control will be used to activate the search
			Results table	Table	This control will be used to display all the product categories that exist on the system
			Return Reason	Label	This control will be used as a heading to display the Name for all return reasons
			Action - Update	Button	This control will be placed in a row, with the delete button, under the "Action" heading for each entry in the Supplier table. The button will direct the user to the Update Return Reason screen
			Action - Delete	Button	This control will be placed in a row, with the delete

					button, under the “Action” heading for each entry in the Supplier table. The button will direct the user to the Delete Return Reason Details screen
			Showing “” of “” entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database
			“Previous” and “Next”	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on
	Step 3: The system administrator clicks on the Create New Return Reason button		Step 4: The system displays the Create Return Reason screen with the following controls:		
			Control Name	Control Type	Notes
			Create Country	Heading	Page main heading
			Return Reason information	Heading	Page subheading
			Return Reason	Label	This control will be used as a label to indicate the Name attribute of the Return Reason record

			<table><tr><td>Name</td><td>Textbox</td><td>This control will be used by the administrator to insert a name of theReturn Reason</td></tr><tr><td>Create</td><td>Button</td><td>This control will be used to confirm the addition of the Return Reason</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the addition of the Return Reason</td></tr></table>	Name	Textbox	This control will be used by the administrator to insert a name of theReturn Reason	Create	Button	This control will be used to confirm the addition of the Return Reason	Cancel	Button	This control will be used to cancel the addition of the Return Reason
	Name	Textbox	This control will be used by the administrator to insert a name of theReturn Reason									
	Create	Button	This control will be used to confirm the addition of the Return Reason									
	Cancel	Button	This control will be used to cancel the addition of the Return Reason									
			Note: All controls are enabled by default unless stated otherwise.									
Step 5: The system administrator enters the details and clicks on the Create button [ALT]		Step 6: The system captures and validates the entered details against the Return Reason table [ALT] <table><tr><th>Details to be validated</th><th>Attributes in the Return Reason table</th><th>Notes</th></tr><tr><td>Name</td><td>ReturnReasonName</td><td>Required, Unique</td></tr></table>	Details to be validated	Attributes in the Return Reason table	Notes	Name	ReturnReasonName	Required, Unique				
Details to be validated	Attributes in the Return Reason table	Notes										
Name	ReturnReasonName	Required, Unique										
		Step 7: The system uses an SQL READ query to check if the Return Reason already exists in the Return Reason table using the following attributes [ALT] <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Name</td><td>ReturnReasonName</td><td></td></tr></table>	Details to be read	Attributes in the table	Notes	Name	ReturnReasonName					
Details to be read	Attributes in the table	Notes										
Name	ReturnReasonName											
		Step 8: The system saves the information into the Return Reason table using an INSERT SQL query: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Id</td><td>Id</td><td>A unique value generated by the system and saved to the database.</td></tr><tr><td>Name</td><td>ReturnReasonName</td><td>Required</td></tr></table>	Details to be saved	Attributes in the table	Notes	Id	Id	A unique value generated by the system and saved to the database.	Name	ReturnReasonName	Required	
Details to be saved	Attributes in the table	Notes										
Id	Id	A unique value generated by the system and saved to the database.										
Name	ReturnReasonName	Required										

			Step 9: The system redirects the system administrator to the Return Reason List Screen and displays a message box with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Checkbox</td><td>Icon</td><td>This control is used to indicate a successful creation</td></tr><tr><td>"Return Reason Created Successfully"</td><td>Label</td><td>This control is used to indicate a successful creation</td></tr></table>	Control Name	Control Name	Notes	Checkbox	Icon	This control is used to indicate a successful creation	"Return Reason Created Successfully"	Label	This control is used to indicate a successful creation
Control Name	Control Name	Notes										
Checkbox	Icon	This control is used to indicate a successful creation										
"Return Reason Created Successfully"	Label	This control is used to indicate a successful creation										
ALTERNATE COURSES:	ALT Step 5: The administrator does not want to add a Return Reason anymore - The administrator clicks on the cancel button - The use case it terminated											
	ALT Step 6: The details entered are not in the correct format or the required fields are empty The system displays a validation error message box with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Name</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>Name is required</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Name textbox</td></tr></table>			Control Name	Control Name	Notes	Name	Textbox	This control will be outline in red to show a validation error	Name is required	Label	This control will be used as a label to indicate a validation error in the Name textbox
	Control Name	Control Name	Notes									
	Name	Textbox	This control will be outline in red to show a validation error									
Name is required	Label	This control will be used as a label to indicate a validation error in the Name textbox										
Return to step 5												
	ALT Step 7: The Return Reason already exists on the system - The system displays an error modal dialogue with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Return Reason already exists!</td><td>Label</td><td>This control is used to inform the administrator that the supplier already exists</td></tr><tr><td>Okay</td><td>Button</td><td>This control is used to cancel the creation of the country</td></tr></table> - The administrator clicks the "Okay" button - The use case is terminated			Control Name	Control Name	Notes	Return Reason already exists!	Label	This control is used to inform the administrator that the supplier already exists	Okay	Button	This control is used to cancel the creation of the country
Control Name	Control Name	Notes										
Return Reason already exists!	Label	This control is used to inform the administrator that the supplier already exists										
Okay	Button	This control is used to cancel the creation of the country										
CONCLUSION:	This use case concludes when a Return Reason has been added on the system											
POST-CONDITION:	A new return reason has been added to the Return Reason table in the database											
BUSINESS RULES	Only the administrator has access to this function.											

IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• The system is offline, and the system administrator will not be able to add a new country.
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 30 - 4.14 Search Return Reason

PRODUCT SUBSYSTEM							
AUTHORS (s): Ofhani Mungani		DATE: 18/04/2022					
VERSION: 1.0		LAST REVIEW DATE: 21/08/2022					
USE CASE NAME:	Search Return Reason		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒				
USE CASE ID:	4.14						
PRIORITY:	Medium						
SOURCE:	ByteXcom Requirements List						
PRIMARY BUSINESS ACTOR	• Administrator						
PRIMARY SYSTEM ACTOR	• None						
OTHER PARTICIPATING ACTORS:	• None						
OTHER INTERESTED STAKEHOLDERS:	• None						
DESCRIPTION:	This use case describes the process where the administrator would like to search for a return reason onto the system. The use case begins with the administrator wishing to search for a return reason. The system will prompt the administrator to enter a return reason the administrator wishes to search for. The system will search through the ReturnReason entity to find the instance value in the ReturnReasonName attribute which match the return reason name provided by the administrator. The use case concludes when the system displays the results of the search query						
PRE-CONDITION:	• The administrator must be logged onto the system.						
TRIGGER:	This use case is called when the administrator wants to search for a Return Reason on the system.						
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE					
		MANUAL ACTION	AUTOMATED ACTION				
	Step 1: The administrator wishes to search for a Return Reason on the system.						
	Step 2: The administrator navigates to the Return Reason List Screen under the Return Reason tab on the side bar		Step 3: The system retrieves following information using an SQL READ QUERY From the Return Reason table: <table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Name</td><td>ReturnReasonName</td></tr></table>	Details to be read	Attributes in the table	Name	ReturnReasonName
	Details to be read	Attributes in the table					
Name	ReturnReasonName						

And displays the **Return Reason Screen** with the following controls:

Control Name	Control Type	Notes
Return Reason List	Heading	Page main heading
Question Mark Icon	Button	When clicked, this control will open a pdf which contains the help document
Create New Country	Button	When clicked, this control will direct the user to the Add Return Reason page
Search bar	Textbox	This control will be used when the user inserts the search parameters
Search icon	Button	This control will be used to activate the search
Results table	Table	This control will be used to display all the product categories that exist on the system
Return Reason Name	Label	This control will be used as a heading to display the Name for all countries
Action	Table Heading	This control will be used as a heading to display the actions the administrator can perform with each entry – Edit and Delete
Action - Update	Button	This control will be placed in a row, with the delete button, under the “Action” heading for each entry in

					the Supplier table. The button will direct the user to the Update Return Reason screen
			Action - Delete	Button	This control will be placed in a row, with the delete button, under the "Action" heading for each entry in the Supplier table. The button will direct the user to the Delete Return Reason Details screen
			Showing "" to "" of "" entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database
			"Previous" and "Next"	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on
	Step 4: The administrator enters their search query into the search textbox.		Step 5: The system searches the Return Reason table and displays the Country(s) matching the search query in the Return Reason Screen [ALT]		

	ALT Step 5: The search query entered does not match any records existing in the Return Reason entity. The system displays a “No Records Found” message in the Return Reason Screen and on the results table.
CONCLUSION:	This use case concludes when the required Return Reason is displayed on the screen.
POST-CONDITION:	The administrator selects the required Return Reason from the displayed results.
BUSINESS RULES	Only the administrator has access to this function.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

Table 31 - 4.15 Update Return Reason

PRODUCT SUBSYSTEM			
AUTHORS (s): Ofhani Mungani		DATE: 18/04/2022	
VERSION: 1.0		LAST REVIEW DATE: 21/08/2022	
USE CASE NAME:	Update Country		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒
USE CASE ID:	4.15		
PRIORITY:	Medium		
SOURCE:	ByteXcom Requirements List		
PRIMARY BUSINESS ACTOR	• Administrator		
PRIMARY SYSTEM ACTOR	• None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	This use case describes the process where the administrator would like to update the return reason onto the system. The use case begins with the administrator requesting to update a return reason. The administrator will select an option to search for a return reason and the system will invoke use case case 4.14 Search ReturnReason to retrieve the return reason. The administrator will select and edit the return reason and save it onto the system database on the ReturnReason entity . The use case concludes when the system displays a return reason successfully updated message to the administrator. concludes when the system displays the Return Reason successfully updated message to the administrator.		
PRE-CONDITION:	• The administrator must be logged onto the system.		
TRIGGER:	The use case is triggered when the administrator wants to update a Country.		
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE	
		MANUAL ACTION	AUTOMATED ACTION
	Step 1: The administrator wishes to update a country's name.		Step 2: The system invokes use case 4.14 Search Return Reason
	Step 2: The administrator clicks on the Update button		Step 3: The system retrieves the selected Country's information from the following attributes and tables using a method in a .NET Controller and an SQL READ Query: From the Return Reason table:

		<table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Name</td><td>ReturnReasonName</td></tr></table> And displays the Update Return Reason Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Update Country</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Return Reason information</td><td>Heading</td><td>Page subheading</td></tr><tr><td>Return Reason</td><td>Label</td><td>This control will be used as a label to indicate the Name attribute of the Return Reason record</td></tr><tr><td>Name</td><td>Textbox</td><td>This control will be used by the administrator to update a name of the Country</td></tr><tr><td>Update</td><td>Button</td><td>This control will be used to confirm the addition of the Country</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the edit of the Country</td></tr></table> Note: All controls are enabled by default unless stated otherwise.	Details to be read	Attributes in the table	Name	ReturnReasonName	Control Name	Control Type	Notes	Update Country	Heading	Page main heading	Return Reason information	Heading	Page subheading	Return Reason	Label	This control will be used as a label to indicate the Name attribute of the Return Reason record	Name	Textbox	This control will be used by the administrator to update a name of the Country	Update	Button	This control will be used to confirm the addition of the Country	Cancel	Button	This control will be used to cancel the edit of the Country
	Details to be read	Attributes in the table																									
	Name	ReturnReasonName																									
	Control Name	Control Type	Notes																								
	Update Country	Heading	Page main heading																								
	Return Reason information	Heading	Page subheading																								
	Return Reason	Label	This control will be used as a label to indicate the Name attribute of the Return Reason record																								
	Name	Textbox	This control will be used by the administrator to update a name of the Country																								
	Update	Button	This control will be used to confirm the addition of the Country																								
	Cancel	Button	This control will be used to cancel the edit of the Country																								
Step 4: The system administrator updates the Return Reason information and clicks on the Update button [ALT]	Step 5: The system captures and validates the entered details against the Return Reason table [ALT] <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Name</td><td>ReturnReasonName</td><td>Required, Unique,</td></tr></table>	Details to be read	Attributes in the table	Notes	Name	ReturnReasonName	Required, Unique,																				
Details to be read	Attributes in the table	Notes																									
Name	ReturnReasonName	Required, Unique,																									
	Step 6: The system uses an SQL READ query to check if the Return Reason already exists in the Return Reason table using the following attributes [ALT] <table><tr><th>Details to be updated</th><th>Attributes in the table</th></tr></table>	Details to be updated	Attributes in the table																								
Details to be updated	Attributes in the table																										

			<table><tr><td>Name</td><td>ReturnReasonName</td></tr></table>	Name	ReturnReasonName													
	Name	ReturnReasonName																
			Step 7: The system displays a message box with the following controls: <table><tr><td>Control Name</td><td>Control Type</td><td>Notes</td></tr><tr><td>Confirmation required</td><td>Label</td><td>This control is used to display the type of message box displayed</td></tr><tr><td>Confirm updated Return Reason details?</td><td>Label</td><td>This control is used to request confirmation on the updated Return Reason details</td></tr><tr><td>Confirm</td><td>Button</td><td>This control is used to confirm the updating of the Return Reason details</td></tr><tr><td>Cancel</td><td>Button</td><td>This control is used to close the message box for the user to edit the Return Reason details</td></tr></table>	Control Name	Control Type	Notes	Confirmation required	Label	This control is used to display the type of message box displayed	Confirm updated Return Reason details?	Label	This control is used to request confirmation on the updated Return Reason details	Confirm	Button	This control is used to confirm the updating of the Return Reason details	Cancel	Button	This control is used to close the message box for the user to edit the Return Reason details
	Control Name	Control Type	Notes															
	Confirmation required	Label	This control is used to display the type of message box displayed															
	Confirm updated Return Reason details?	Label	This control is used to request confirmation on the updated Return Reason details															
	Confirm	Button	This control is used to confirm the updating of the Return Reason details															
	Cancel	Button	This control is used to close the message box for the user to edit the Return Reason details															
			Step 8: The system administrator clicks on the Confirm button [ALT]															
			Step 9: The system saves the information into the Return Reason table using an UPDATE SQL query: <table><tr><td>Details to be saved</td><td>Attributes in the table</td><td>Notes</td></tr><tr><td>Name</td><td>ReturnReasonName</td><td>Required, Unique,</td></tr></table>	Details to be saved	Attributes in the table	Notes	Name	ReturnReasonName	Required, Unique,									
Details to be saved	Attributes in the table	Notes																
Name	ReturnReasonName	Required, Unique,																

			Step 10: The system redirects the system administrator to the Return Reason Screen and displays a message box with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Checkbox</td><td>Icon</td><td>This control is used to indicate a successful update</td></tr><tr><td>"Return Reason Created Successfully"</td><td>Label</td><td>This control is used to indicate a successful update</td></tr></table>	Control Name	Control Name	Notes	Checkbox	Icon	This control is used to indicate a successful update	"Return Reason Created Successfully"	Label	This control is used to indicate a successful update
Control Name	Control Name	Notes										
Checkbox	Icon	This control is used to indicate a successful update										
"Return Reason Created Successfully"	Label	This control is used to indicate a successful update										
ALTERNATE COURSES:	ALT Step 4: The administrator does not want to update the Return Reason anymore - The administrator clicks on the cancel button - The use case is terminated											
	ALT Step 5: The details entered are not in the correct format or the required fields are empty The system displays a validation error message box with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Name</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>Name is required</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Name textbox</td></tr></table>			Control Name	Control Name	Notes	Name	Textbox	This control will be outline in red to show a validation error	Name is required	Label	This control will be used as a label to indicate a validation error in the Name textbox
	Control Name	Control Name	Notes									
	Name	Textbox	This control will be outline in red to show a validation error									
	Name is required	Label	This control will be used as a label to indicate a validation error in the Name textbox									
ALT Step 6: The Return Reason already exists on the system - The system displays an error modal dialogue with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Return Reason already exists!</td><td>Label</td><td>This control is used to inform the administrator that the supplier already exists</td></tr><tr><td>Okay</td><td>Button</td><td>This control is used to cancel the creation of the return reason</td></tr></table> - The administrator clicks the "Okay" button - The use case is terminated			Control Name	Control Name	Notes	Return Reason already exists!	Label	This control is used to inform the administrator that the supplier already exists	Okay	Button	This control is used to cancel the creation of the return reason	
Control Name	Control Name	Notes										
Return Reason already exists!	Label	This control is used to inform the administrator that the supplier already exists										
Okay	Button	This control is used to cancel the creation of the return reason										
ALT Step 8: The administrator clicks on the cancel button The use case is terminated												
CONCLUSION:	This use case concludes when a Return Reason has been added on the system											
POST-CONDITION:	A Return Reason has been updated in the Return Reason table in the database											

BUSINESS RULES	• Only the administrator has access to this function.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 32 - 4.16 Delete Return Reason

PRODUCT SUBSYSTEM							
AUTHORS (s): Ofhani Mungani		DATE: 18/04/2022					
VERSION: 1.0		LAST REVIEW DATE: 21/08/2022					
USE CASE NAME:	Delete Return Reason		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒				
USE CASE ID:	4.16						
PRIORITY:	Medium						
SOURCE:	ByteXcom Requirements List						
PRIMARY BUSINESS ACTOR	• Administrator						
PRIMARY SYSTEM ACTOR	• None						
OTHER PARTICIPATING ACTORS:	• None						
OTHER INTERESTED STAKEHOLDERS:	• None						
DESCRIPTION:	This use case narrative describes the process in which the administrator wishes to delete a return reason from the system. The administrator will search for the required return reason selected for deletion. The administrator will then confirm the deletion of the return reason and the use case concludes when the system notifies the administrator of a successful deletion.						
PRE-CONDITION:	• The administrator must be logged onto the system.						
TRIGGER:	The use case is triggered when the administrator wants to delete a Country.						
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE					
		MANUAL ACTION	AUTOMATED ACTION				
	Step 1: The administrator wishes to delete a Return Reason		Step 2: The system invokes use case 4.14 Search Return Reason				
	Step 3: The administrator clicks on the Delete Icon		Step 4: The system retrieves the Return Reason information from the following attributes and tables using a method in a .NET Controller and an SQL READ Query: From the ReturnReason table: <table><tr><td>Details to be read</td><td>Attributes in the table Notes</td></tr><tr><td>Name</td><td>ReturnReasonName</td></tr></table>	Details to be read	Attributes in the table Notes	Name	ReturnReasonName
	Details to be read	Attributes in the table Notes					
	Name	ReturnReasonName					

		And displays the Delete Return Reason screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Delete Country</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Return Reason</td><td>Label</td><td>This control will be used as a label to indicate the Name attribute of the Return Reason record</td></tr><tr><td>Name</td><td>Label</td><td>This control will be used by the administrator to display the name of the Return Reason</td></tr><tr><td>Delete</td><td>Button</td><td>This control will be used to confirm the delete of the Return Reason</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the addition of the Return Reason</td></tr></table>	Control Name	Control Type	Notes	Delete Country	Heading	Page main heading	Return Reason	Label	This control will be used as a label to indicate the Name attribute of the Return Reason record	Name	Label	This control will be used by the administrator to display the name of the Return Reason	Delete	Button	This control will be used to confirm the delete of the Return Reason	Cancel	Button	This control will be used to cancel the addition of the Return Reason
	Control Name	Control Type	Notes																	
	Delete Country	Heading	Page main heading																	
	Return Reason	Label	This control will be used as a label to indicate the Name attribute of the Return Reason record																	
	Name	Label	This control will be used by the administrator to display the name of the Return Reason																	
	Delete	Button	This control will be used to confirm the delete of the Return Reason																	
	Cancel	Button	This control will be used to cancel the addition of the Return Reason																	
	Step 5: The administrator clicks on the Delete button [ALT]	Step 6: The system displays a message box with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Confirmation required</td><td>Label</td><td>This control is used to display the type of message box displayed</td></tr><tr><td>Are you sure you want to delete this Country?</td><td>Label</td><td>This control is used to request confirmation of the deletion of the Return Reason</td></tr><tr><td>Confirm</td><td>Button</td><td>This control is used to</td></tr></table>	Control Name	Control Type	Notes	Confirmation required	Label	This control is used to display the type of message box displayed	Are you sure you want to delete this Country?	Label	This control is used to request confirmation of the deletion of the Return Reason	Confirm	Button	This control is used to						
	Control Name	Control Type	Notes																	
	Confirmation required	Label	This control is used to display the type of message box displayed																	
Are you sure you want to delete this Country?	Label	This control is used to request confirmation of the deletion of the Return Reason																		
Confirm	Button	This control is used to																		

					confirm the deletion of the Return Reason details								
			Cancel	Button	This control is used to close the message box deletion of the Country								
	Step 7: The system administrator clicks on the Confirm button [ALT]		Step 8: The system deletes the selected Return Reason from the Return Reason entity in the database using a method in a .NET controller and an SQL DELETE query. <table><tr><th>Details to be deleted</th><th>Attributes in the table</th></tr><tr><td>Id</td><td>Id</td></tr><tr><td>Name</td><td>ReturnReasonName</td></tr></table> [ALT]			Details to be deleted	Attributes in the table	Id	Id	Name	ReturnReasonName		
						Details to be deleted	Attributes in the table						
Id	Id												
Name	ReturnReasonName												
		Step 9: The system redirects the system administrator to the Return Reason screen and displays a message box with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Checkbox</td><td>Icon</td><td>This control is used to indicate a successful deletion</td></tr><tr><td>"Return Reason Deleted Successfully"</td><td>Label</td><td>This control is used to indicate a successful deletion</td></tr></table>			Control Name	Control Name	Notes	Checkbox	Icon	This control is used to indicate a successful deletion	"Return Reason Deleted Successfully"	Label	This control is used to indicate a successful deletion
Control Name	Control Name	Notes											
Checkbox	Icon	This control is used to indicate a successful deletion											
"Return Reason Deleted Successfully"	Label	This control is used to indicate a successful deletion											
ALTERNATE COURSES:	ALT Step 5: The administrator does not want to delete the Return Reason anymore - The administrator clicks on the cancel button - The use case is terminated												
	ALT Step 7: The administrator does not want to confirm to delete the Return Reason anymore - The administrator clicks on the cancel button - Return to Step 5												
	ALT Step 8: The Return Reason cannot be deleted as it has an association with the ReturnedProduct Table The system redirects the administrator and displays a bootstrap alert notification with the following controls:												

	Control Name	Control Name	Notes
	"Return Reason cannot be deleted"	Label	This control is used to indicate an unsuccessful deletion
	The use case is terminated		
CONCLUSION:	This use case concludes when a Return Reason has been deleted on the system		
POST-CONDITION:	The Return Reason has been removed from the system database.		
BUSINESS RULES	Only the administrator has access to this function.		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

Table 33 - 4.17 Browse Product Catalogue

PRODUCT SUBSYSTEM																				
AUTHOR(s): Thenjiwe Ntsonda		DATE: 02/09/2022																		
VERSION: 1.0		LAST REVIEW DATE: 02/09/2022																		
USE CASE NAME:	Browse Product Catalogue	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒																		
USE CASE ID:	4.17																			
PRIORITY:	Medium																			
SOURCE:	ByteXpress Requirements List																			
PRIMARY BUSINESS ACTOR:	• User																			
PRIMARY THE SYSTEM ACTOR:	• None																			
OTHER PARTICIPATING ACTORS:	• None																			
OTHER INTERESTED STAKEHOLDERS:	• None																			
DESCRIPTION:	This use case describes the process in which the users of the system, namely customers and resellers, want to browse the product catalogue existing on the system. The users will navigate to the product catalogue on the system and the products available will be displayed.																			
PRE-CONDITION:	• Resellers must be logged into the system for their prices to reflect.																			
TRIGGER:																				
TYPICAL COURSE OF EVENTS:	ACTOR ACTION:	SYSTEM RESPONSE:																		
		Manual Action	Automated Action																	
	Step 1: The user clicks on the Product link in the navigation bar.		Step 2: The system retrieves the product information from the following entities using a method in a .NET Controller and an SQL READ Query: From the Product Table:																	
			<table><tr><th>Details to be read</th><th>Attributes in the table Notes</th></tr><tr><td>Name</td><td>Name</td></tr><tr><td>Description</td><td>Description</td></tr><tr><td>PictureUrl</td><td>PictureUrl</td></tr><tr><td>Customer Price</td><td>CustomerPrice</td></tr><tr><td>Reseller Price</td><td>ResellerPrice</td></tr><tr><td>Quantity On Hand</td><td>QuantityOnHand</td></tr><tr><td>Threshold Value</td><td>ThresholdValue</td></tr></table>		Details to be read	Attributes in the table Notes	Name	Name	Description	Description	PictureUrl	PictureUrl	Customer Price	CustomerPrice	Reseller Price	ResellerPrice	Quantity On Hand	QuantityOnHand	Threshold Value	ThresholdValue
			Details to be read	Attributes in the table Notes																
Name			Name																	
Description	Description																			
PictureUrl	PictureUrl																			
Customer Price	CustomerPrice																			
Reseller Price	ResellerPrice																			
Quantity On Hand	QuantityOnHand																			
Threshold Value	ThresholdValue																			

From the **ProductCategory** Table:

Details to be read	Attributes in the table Notes
Name	Name

From the ProductBrand Table:

Details to be read	Attributes in the table Notes
Name	Name

And displays the **Product Catalogue** Screen with the following controls:

Control Name	Control Type	Notes
Products	Heading	Page main heading
Search Bar	Textbox	This control will be used when the user inserts the search parameters
Search Icon	Button	This control will be used to activate the search
Showing “” to “” of “” entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database
Categories	Label	This control will be used as a heading to display all available categories on the system.
Categories	Hyper-Link Button	This control will be used to filter the products in the product catalogue.
Product Image	Image	This control will be used to display the Image of the product and is retrieved from the Image attribute
Product Name	Label	This control will be used to display the

					name of the product and is retrieved from the Name attribute
			Product Price	Label	This control will be used to display the price of the product and is retrieved from the CustomerPrice attribute
			View Product	Button	This control will be used to redirect the user to view more details about the product
			[ALT]		
	Step 3: The enters a search query in the search bar. [ALT]		Step 4: The system searches the Product table and displays the product(s) matching the search query in the Product Catalogue Screen		
ALTERNATE COURSES	ALT Step 2: The user is logged in as a reseller. - The system displays the reseller prices from the ResellerPrice attribute - Go to Step 3				
	ALT Step 3: The user filters the product catalogue using the selected product category in the system. The system filters the product catalogue using the selected product category				
	ALT Step 4A: The entered search query does not match any records in the table. Display no records found message				
CONCLUSION:	This use case concludes when the product catalogue with the relevant prices is displayed according to which type of user is logged in.				
POST-CONDITION:	All product details are retrieved from the Product, ProductCategory and ProductBrand entities entity is read.				
BUSINESS RULES:	The reseller prices are only displayed to users logged in as resellers.				
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None				
ASSUMPTIONS:	None				
OPEN ISSUES:	None				

Table 34 - 4.18 Configure Product

PRODUCT SUBSYSTEM						
AUTHORS (s): Kyle van Eeden			DATE: 09/06/2022			
VERSION: 1.1			LAST REVIEW DATE: 10/07/2022			
USE CASE NAME:	Configure Product		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒			
USE CASE ID:	4.18					
PRIORITY:	High					
SOURCE:	ByteXcom Requirements List					
PRIMARY BUSINESS ACTOR	Inventory Manager or Administrator					
PRIMARY SYSTEM ACTOR	None					
OTHER PARTICIPATING ACTORS:	None					
OTHER INTERESTED STAKEHOLDERS:	None					
DESCRIPTION:	This use case describes the event in which the administrator or inventory manager wishes to configure a product for assembly by assigning all inventory items that make up the final product to be sold as well as the quantity needed of each inventory item to produce a single product. The inventory manager or administrator will request the system to view the current production configuration of a particular product and the system will retrieve the Product Name, Quantity on Hand, Inventory Item Name of all associated Inventory Items and the quantities needed for each. The system administrator or inventory manager user can then update the list of currently assigned inventory items and update the quantities of all assigned inventories that make up the list. Upon submitting the updated details the system will save the changes to the ProductInventoryItem entity in the database and the use case concludes as the system displays a product configuration successful message notification box.					
PRE-CONDITION:	- The Inventory Manager or Administrator must be logged into the system. - The product to be configured for production should be defined on the system.					
TRIGGER:	This use case is triggered when the Inventory Manager or Administrator requests the system to view a specific product's production configuration.					
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE				
		MANUAL ACTION	AUTOMATED ACTION			
	Step 1: The Inventory Manager or Administrator wishes to configure a product's assembly requirements and clicks on the Product		Step 2: The system retrieves the following details for all products in the Product entity using a .NET controller and SQL Read Query: <table><tr><td>Details to be read</td><td>Attributes in the table</td><td>Notes</td></tr></table>		Details to be read	Attributes in the table
Details to be read	Attributes in the table	Notes				

	Configuration navigation link in the side navigation bar.		Product Name	Product Name	Name of each product retrieved
			Quantity On Hand	Quantity On Hand	Quantity available of each product retrieved.
			The system then displays the Products List page which contains a ASP.NET form with the following controls:		
			Control Name	Control Type	Control Notes
			Products List	Header	Main heading of the page.
			Search Product	Search Box	Used to filter the results displayed in the Products table.
			Clear	Button	Used to clear all search filters on the Products table.
			Products	Table	Table which contains the details retrieved for all products.
			Product	Table Heading	Heading of the table field that holds the name of all products retrieved.
			Product	Table Row	The ProductName value is displayed for each product retrieved.
			Quantity On Hand	Table Heading	Heading of the table's field that holds the quantity on hand for each product retrieved.
			Quantity On Hand	Table Row	The QuantityOnHand attribute value is displayed for each product retrieved.
			Configure Inventories	Button	Allows the user to navigate to the Configure Product

						Inventories page populated with the details of the specific product selected.																	
				Back	Button	Navigates user back to the previous page if clicked.																	
	Note: All controls on the form are enabled by default, unless specified otherwise in the Control Notes section.																						
	Step 3: The Inventory Manager or Administrator enters the search criteria to filter the products displayed.		Step 4: The system displays all products where the ProductName attribute value(s) matched the search parameter entered by populating results in the Products table.																				
	Step 5: The Inventory Manager or Administrator clicks on the Configure Inventories button for the specific product they wish to configure for assembly. [ALT]		Step 6: The system retrieves the following details for the specific product selected for configuration using a .NET controller with SQL Read queries: From the Product entity: <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Product</td><td>ProductName</td><td>Name of the product requested for assembly configuration.</td></tr></table> From the InventoryItem entity: <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Inventory Item</td><td>InventoryItemName</td><td>Names of all inventory items associated with the selected product's assembly configuration.</td></tr></table> From the ProductInventoryItem entity:: <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Product ID</td><td>ProductID</td><td>Product requested for production configuration</td></tr></table>				Details to be read	Attributes in the table	Notes	Product	ProductName	Name of the product requested for assembly configuration.	Details to be read	Attributes in the table	Notes	Inventory Item	InventoryItemName	Names of all inventory items associated with the selected product's assembly configuration.	Details to be read	Attributes in the table	Notes	Product ID	ProductID
Details to be read	Attributes in the table	Notes																					
Product	ProductName	Name of the product requested for assembly configuration.																					
Details to be read	Attributes in the table	Notes																					
Inventory Item	InventoryItemName	Names of all inventory items associated with the selected product's assembly configuration.																					
Details to be read	Attributes in the table	Notes																					
Product ID	ProductID	Product requested for production configuration																					

			Inventory Item ID	InventoryItemID	Retrieve currently associated Inventory Items.
			Inventory Item Quantity	InventoryItemQuantity	Quantities assigned for each associated Inventory Item
			The system then displays the Configure Product Inventories page containing a .NET form with the following controls:		
			Control Name	Control Type	Notes
			Product Inventories Configuration	Heading	The main heading of the page displayed.
			Product	Label	Label for the Product textbox control.
			Product	Textbox	Read-Only field that is populated with the ProductName attribute value for the requested product to configure.
			Search Inventories	Label	Label for the Search Inventories Search List Box Control
			Search Inventories	Search List Box	Used to search for Inventory Items and selecting inventories to add them to the Inventories List.
			Inventories List	Label	Label for the Inventories List table.
			Inventories List	Table	Table control containing all the Inventory Items assigned to the product for its production configuration.
			Inventory Item Name	Table Heading	Heading of the Inventory Item Name of

					the Inventories List table.
			Inventory Item Name	Table Row	Populated with the InventoryItemName attribute value for each inventory item added.
			Inventory Item Quantity	Table Heading	Heading of the Inventory Item Quantity field in the Inventories List table.
			Inventory Item Quantity	Table Row	Used to capture the InventoryItemQuantity attribute value for each inventory item added.
			Remove	Button	Allows the user to remove a currently assigned inventory item.
			Cancel	Button	Redirects the user to the Product List screen if clicked.
			Save	Button	Used to save the configuration details of the particular product.
	Step 7: The Inventory Manager or Administrator enters the search parameter value for the specific inventory item they wish to add. [ALT]		Note: All controls on the form are enabled by default, unless specified otherwise in the Control Notes section.		
			Step 8: The system retrieves the InventoryName attribute value and populates the Inventories list with all the inventory items that matched the search criteria from the InventoryItem entity using a .NET controller method and a SQL Read Query:		
			Details to be Read	Attributes in the table	Notes
			Inventory Item Name	InventoryItemName	Name values of all inventory items that match the search

					parameter entered.									
	Step 9: The Inventory Manager or Administrator selects the Inventory Item option displayed in the Search Inventories Search List Box control.		Step 10: The system updates the details of the Inventories List table displayed. The following details are displayed for each inventory item assigned to the Inventories List table : <table><tr><th>Details to be Displayed</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Inventory Item Name</td><td>InventoryItemName</td><td>Name of each inventory item in the current inventories list.</td></tr><tr><td>Inventory Item Quantity</td><td>InventoryItemQuantity</td><td>Quantity needed of each inventory item. Set to 0 by default.</td></tr></table>	Details to be Displayed	Attributes in the table	Notes	Inventory Item Name	InventoryItemName	Name of each inventory item in the current inventories list.	Inventory Item Quantity	InventoryItemQuantity	Quantity needed of each inventory item. Set to 0 by default.		
	Details to be Displayed	Attributes in the table	Notes											
Inventory Item Name	InventoryItemName	Name of each inventory item in the current inventories list.												
Inventory Item Quantity	InventoryItemQuantity	Quantity needed of each inventory item. Set to 0 by default.												
Step 11: The Inventory Manager or Administrator specifies the quantity needed for each inventory item in the Inventories List.														
Step 12: The Inventory Manager or Administrator clicks on the Save button. [ALT]		Step 13: The system captures the updated Inventories List table information and uses a .NET controller with a SQL Update Query to update the quantities of all inventory items associated with the product in the ProductInventoryItem entity: <table><tr><th>Details to be Updated</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Inventory Item Quantity</td><td>InventoryItemQuantity</td><td>Required.</td></tr></table> The system will use a .NET Controller and SQL Insert Query to add a new Inventory item to the ProductInventoryItem entity if a new inventory item has been added to the Inventories List:	Details to be Updated	Attributes in the table	Notes	Inventory Item Quantity	InventoryItemQuantity	Required.						
Details to be Updated	Attributes in the table	Notes												
Inventory Item Quantity	InventoryItemQuantity	Required.												

			<table><tr><th>Details to be Inserted</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Product ID</td><td>ProductID</td><td>Required</td></tr><tr><td>Inventory Item ID</td><td>InventoryItemID</td><td>Required</td></tr><tr><td>Inventory Item Quantity</td><td>InventoryItemQuantity</td><td>Required</td></tr></table>	Details to be Inserted	Attributes in the table	Notes	Product ID	ProductID	Required	Inventory Item ID	InventoryItemID	Required	Inventory Item Quantity	InventoryItemQuantity	Required
	Details to be Inserted	Attributes in the table	Notes												
	Product ID	ProductID	Required												
	Inventory Item ID	InventoryItemID	Required												
	Inventory Item Quantity	InventoryItemQuantity	Required												
		The system will use a .NET Controller and SQL Delete Query to remove the following details of an inventory item from the ProductInventoryItem entity if the inventory item has been removed from the Inventories List: <table><tr><th>Details to be Deleted</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Product ID</td><td>ProductID</td><td>To be removed</td></tr><tr><td>Inventory Item ID</td><td>InventoryItemID</td><td>To be removed</td></tr><tr><td>Inventory Item Quantity</td><td>InventoryItemQuantity</td><td>To be removed</td></tr></table>	Details to be Deleted	Attributes in the table	Notes	Product ID	ProductID	To be removed	Inventory Item ID	InventoryItemID	To be removed	Inventory Item Quantity	InventoryItemQuantity	To be removed	
Details to be Deleted	Attributes in the table	Notes													
Product ID	ProductID	To be removed													
Inventory Item ID	InventoryItemID	To be removed													
Inventory Item Quantity	InventoryItemQuantity	To be removed													
		Step 14: The system redirects the Inventory Manager or Administrator to the Product List page and displays a configuration details saved successfully confirmation message box.													
ALTERNATE COURSES:	ALT Step 5: The Inventory Manager or Administrator no longer wishes to configure the production details for a product defined on the system: • The Inventory Manager or Administrator clicks on the Back button • The Inventory Manager or Administrator is redirected to the Manage Store screen. • The use case is terminated														
	ALT Step 7: The Inventory Manager or Administrator wishes to remove an Inventory Item currently assigned to the inventories list: • The Inventory Manager or Administrator clicks on the Remove button of the specific Inventory Item they wish to remove. • Proceed to Step 10														
	ALT Step 9 : The Inventory Manager or Administrator wishes to remove an inventory item from the inventories list: • The Inventory Manager or Administrator clicks on the Remove button for the specific inventory item they wish to remove. • The system removes the Inventory Item’s details from the Inventory List table. • Proceed to Step 11														
	ALT Step 12: The Inventory Manager or Administrator does not wish to save the current configuration details for the product: • The Inventory Manager or Administrator clicks on the Cancel button • Return to Step 2														

CONCLUSION:	This use case concludes when the system redirects the user to the Product List page and displays a production configuration success alert.
POST-CONDITION:	The product's configuration details are either created or updated in the ProductInventoryItem entity.
BUSINESS RULES	- Only the Inventory Manager and Administrator roles should have access to this functionality. - The system should allow for the Administrator or Inventory Manager to configure the production details of a finished product on the system.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	The system is offline, and the Inventory Manager or Administrator will not be able to configure a finished product's production details.
ASSUMPTIONS:	None
OPEN ISSUES:	None

2.5. Order Subsystem

Table 35 - 5.1 Add Item(s) to Cart

ORDER SUBSYSTEM												
AUTHORS (s): Ofhani Mungani		DATE: 06/24/2022										
VERSION: 1.1		LAST REVIEW DATE: 07/29/2022										
USE CASE NAME:	Add Item(s) to Cart	USE CASE TYPE Business Requirements: ☐ System Analysis: ☒ System Design: ☐										
USE CASE ID:	5.1											
PRIORITY:	High											
SOURCE:	ByteXpress Requirements List											
PRIMARY BUSINESS ACTOR	• Customer or Reseller											
PRIMARY SYSTEM ACTOR	• None											
OTHER PARTICIPATING ACTORS:	• None											
OTHER INTERESTED STAKEHOLDERS:	• None											
DESCRIPTION:	This use case begins when a customer or reseller selects a product to be added to their cart and enters the quantity. They will then verify that the user making the request is logged into their registered account. Once the system has verified that the user is logged in, the requested quantity of the product will be added to the user's cart. The use case concludes with the system displaying an alert that confirms the product has been successfully added to the customer/reseller's cart.											
PRE-CONDITION:	• At least one product must be defined in order to add it to a user's cart. • The customer or reseller must have internet access in order to perform this use case. • A user should have a registered account to be able to add items to their cart.											
TRIGGER:	This use case is called when a customer or reseller wishes to add a specific product to their cart.											
TYPICAL COURSE	Actor Action	System Response										
	Step 1: The user navigates to the Products Screen on the navigation list top navi-bar	Step 2: The system retrieves the following details for all products from the Product entity:										
		<table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Product Name</td><td>ProductName</td></tr><tr><td>Product Picture Url</td><td>ProductPictureUrl</td></tr><tr><td>Customer Price</td><td>CustomerPrice</td></tr><tr><td></td><td></td></tr></table>	Details to be read	Attributes in the table	Product Name	ProductName	Product Picture Url	ProductPictureUrl	Customer Price	CustomerPrice		
		Details to be read	Attributes in the table									
		Product Name	ProductName									
		Product Picture Url	ProductPictureUrl									
Customer Price	CustomerPrice											
The system then displays the Products Screen with the following controls:												
<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Products</td><td>Heading</td><td>Page main heading</td></tr></table>	Control Name	Control Type	Notes	Products	Heading	Page main heading						
Control Name	Control Type	Notes										
Products	Heading	Page main heading										

		View Product	Button	When clicked, this control will direct the user to the product details page								
		Product image	image	This control used to display Product images								
		Search bar	Textbox	This control will be used when the user inserts the search parameters								
		Categories	list	This control will be used to display the list of categories								
		Showing “” to “” of “” entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database								
		Product list	table	This control will be used to display all the products that exist on the system								
		[ALT]										
	Step 3: The reseller or customer click View Product button.	Step 4: The system retrieves the following details for the specific product requested from the respective entities: From the Product entity: <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Product Name</td><td>ProductName</td></tr><tr><td>Product Picture Url</td><td>ProductPictureUrl</td></tr><tr><td>Customer Price or Reseller Price</td><td>CustomerPrice/ ResellerPrice</td></tr></table> From the ProductCategory entity:			Details to be read	Attributes in the table	Product Name	ProductName	Product Picture Url	ProductPictureUrl	Customer Price or Reseller Price	CustomerPrice/ ResellerPrice
	Details to be read	Attributes in the table										
Product Name	ProductName											
Product Picture Url	ProductPictureUrl											
Customer Price or Reseller Price	CustomerPrice/ ResellerPrice											

		<table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Product Category Name</td><td>ProductCategoryName</td></tr></table>	Details to be read	Attributes in the table	Product Category Name	ProductCategoryName																													
	Details to be read	Attributes in the table																																	
	Product Category Name	ProductCategoryName																																	
		From the ProductBrand entity:																																	
		<table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Product Brand Name</td><td>ProductBrandName</td></tr></table>	Details to be read	Attributes in the table	Product Brand Name	ProductBrandName																													
	Details to be read	Attributes in the table																																	
	Product Brand Name	ProductBrandName																																	
		<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Product image</td><td>image</td><td>This control used to display Product images</td></tr><tr><td>Count</td><td>Numeric updown</td><td>This control will be used to enter the count of products</td></tr><tr><td>Back</td><td>button</td><td>This control will be used to navigate back to the Product Screen</td></tr><tr><td>Add to Cart</td><td>button</td><td>This control will be used to add products into a cart</td></tr><tr><td>Name</td><td>Label</td><td>This control will be used to display the product's name</td></tr><tr><td>Description</td><td>Label</td><td>This control will be used to display the product's description</td></tr><tr><td>Brand</td><td>Label</td><td>This control will be used to display the product's brand</td></tr><tr><td>Category</td><td>Label</td><td>This control will be used to display the product's category</td></tr><tr><td>Price</td><td>Label</td><td>This control will be used to display the product's price</td></tr><tr><td>Related Products</td><td>Heading</td><td>Sub section heading</td></tr></table>	Control Name	Control Type	Notes	Product image	image	This control used to display Product images	Count	Numeric updown	This control will be used to enter the count of products	Back	button	This control will be used to navigate back to the Product Screen	Add to Cart	button	This control will be used to add products into a cart	Name	Label	This control will be used to display the product's name	Description	Label	This control will be used to display the product's description	Brand	Label	This control will be used to display the product's brand	Category	Label	This control will be used to display the product's category	Price	Label	This control will be used to display the product's price	Related Products	Heading	Sub section heading
	Control Name	Control Type	Notes																																
	Product image	image	This control used to display Product images																																
Count	Numeric updown	This control will be used to enter the count of products																																	
Back	button	This control will be used to navigate back to the Product Screen																																	
Add to Cart	button	This control will be used to add products into a cart																																	
Name	Label	This control will be used to display the product's name																																	
Description	Label	This control will be used to display the product's description																																	
Brand	Label	This control will be used to display the product's brand																																	
Category	Label	This control will be used to display the product's category																																	
Price	Label	This control will be used to display the product's price																																	
Related Products	Heading	Sub section heading																																	

		<table><tr><td>Product list</td><td>table</td><td>This control will be used to display all the products that exist on the system</td></tr></table> [ALT]	Product list	table	This control will be used to display all the products that exist on the system						
	Product list	table	This control will be used to display all the products that exist on the system								
	Step 5: The user enters the quantity they wish to add and selects the option to add the product to their cart. [ALT]	Step 6: The system verifies that the customer or reseller is logged into an account. [ALT]									
		Step 7: The system verifies that the quantity to be added to cart does not exceed the quantity of the requested product. From the Product entity: <table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Quantity on Hand</td><td>QuantityonHand</td></tr></table> [ALT]	Details to be read	Attributes in the table	Quantity on Hand	QuantityonHand					
	Details to be read	Attributes in the table									
Quantity on Hand	QuantityonHand										
	Step 8: The system read the following account details of the customer or reseller from the User entity: <table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>UserId</td><td>UserId</td></tr></table>	Details to be read	Attributes in the table	UserId	UserId						
Details to be read	Attributes in the table										
UserId	UserId										
	Step 9: The system adds the product and the requested quantity to the customer or reseller user's cart by creating the following details in the Cart entity using SQL INSERT query: <table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Cart Id</td><td>CartId</td></tr><tr><td>User Id</td><td>UserId</td></tr><tr><td>Product Name</td><td>Product Name</td></tr><tr><td>Product Quantity</td><td>ProductQuantity</td></tr></table> [ALT]	Details to be read	Attributes in the table	Cart Id	CartId	User Id	UserId	Product Name	Product Name	Product Quantity	ProductQuantity
Details to be read	Attributes in the table										
Cart Id	CartId										
User Id	UserId										
Product Name	Product Name										
Product Quantity	ProductQuantity										
	Step 10: The system displays an alert notification that informs the customer or reseller that their cart has been updated successfully by displaying the count of products on the cart icon.										

ALTERNATE COURSES:	ALT Step 2: The user making the request is logged in has a Reseller role assigned: - The system retrieves the following details for all products from the Product entity: <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Product Name</td><td>ProductName</td></tr><tr><td>Product Picture Url</td><td>ProductPictureUrl</td></tr><tr><td>Reseller Price</td><td>Reseller Price</td></tr></table> - Continue to Step 3.		Details to be read	Attributes in the table	Product Name	ProductName	Product Picture Url	ProductPictureUrl	Reseller Price	Reseller Price
	Details to be read	Attributes in the table								
	Product Name	ProductName								
	Product Picture Url	ProductPictureUrl								
	Reseller Price	Reseller Price								
	ALT Step 4: The product is out of stock - The system display product out of stock message under a product and disable the Add to Cart button <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Out of stock</td><td>Label</td><td>This control is used to display the product out of stock message</td></tr></table> - The use case concludes.		Control Name	Control Name	Notes	Out of stock	Label	This control is used to display the product out of stock message		
Control Name	Control Name	Notes								
Out of stock	Label	This control is used to display the product out of stock message								
ALT Step 5: The customer or reseller no longer wishes to add a product to their cart: - The customer or reseller selects the option to navigate to the product catalogue interface. - The use case concludes.										
ALT Step 6: The user requesting to add products to their cart is not logged into a registered account: - The system prompts the unauthenticated user to log into an existing account or register a new account. - Use case concludes.										
ALT Step 9: The user requesting to add the product to their cart already has the specific product added to their cart: - The system updates the quantity of the already existing product in the cart by increasing the current value with the quantity requested. - The system saves the updated quantity of the product in the UserCart entity. - Proceed to Step 10.										
CONCLUSION:	The system displays an alert notification that informs the customer or reseller that their cart has been updated successfully.									
POST-CONDITION:	The customer or reseller’s cart is updated successfully as the system either adds new cart details to the UserCart entity or the quantity of the product is incremented by 1 if the user already had the item added to their cart.									
BUSINESS RULES	Customers and resellers should be allowed to add a product along with the desired quantity to their cart.									

	- The prices displayed in Natuurlik's catalogue should display the customer price for all users that are not resellers while reseller prices are displayed to resellers. - A product added to a user's cart should be persisted for at least 7 days.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	- The cart total should be updated the moment that a customer or reseller adds a new product to their cart. - A user's cart items should be persisted in the database for at least 7 days from the time it is added.
ASSUMPTIONS:	The customer or reseller is unaware of any products currently in their cart when performing this use case.
OPEN ISSUES:	None

Table 36 - 5.2 Update Cart

ORDER SUBSYSTEM										
AUTHORS (s): Nomusa Vumisa			DATE: 06/22/2022							
VERSION: 1.0			LAST REVIEW DATE: 06/22/2022							
USE CASE NAME:	Update Cart		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒							
USE CASE ID:	5.2									
PRIORITY:	High									
SOURCE:	ByteXcom Requirements List									
PRIMARY BUSINESS ACTOR	• Customer or Reseller									
PRIMARY SYSTEM ACTOR	• None									
OTHER PARTICIPATING ACTORS:	• None									
OTHER INTERESTED STAKEHOLDERS:	• None									
DESCRIPTION:	This use case begins when a customer or reseller wishes to update the products and their respective quantities in their cart. The customer or reseller will either increment or decrement the current quantity of any product in their cart. The use case concludes when the system displays the updated quantities for each product in the user’s cart.									
PRE-CONDITION:	• The customer or reseller must be logged in The cart must not be empty									
TRIGGER:	The customer or reseller wishes to update their cart.									
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE								
		MANUAL ACTION	AUTOMATED ACTION							
	Step 1: The customer or reseller wishes to update their cart.		Step 2: Extends from Use Case 5.4 View Cart.							
	Step 3: The customer or reseller clicks on the “+” button [ALT]		Step 4: The system verifies that there is enough stock on hand to fulfil the order using the following attribute in the Product table and a method in a .NET controller and an SQL READ query: <table><tr><td>Details to be saved</td><td>Attributes in the table</td><td>Notes</td></tr><tr><td>Quantity on Hand</td><td>QuantityOn Hand</td><td>Minimum: 1</td></tr></table> [ALT]		Details to be saved	Attributes in the table	Notes	Quantity on Hand	QuantityOn Hand	Minimum: 1
	Details to be saved	Attributes in the table	Notes							
	Quantity on Hand	QuantityOn Hand	Minimum: 1							
		Step 5: The system saves the updated quantity of the product into the Cart								

ALTERNATIVE STEPS			entity using a method in a .NET controller and an SQL UPDATE query:							
			<table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Quantity</td><td>Quantity</td><td>Minimum: 1</td></tr></table>	Details to be saved	Attributes in the table	Notes	Quantity	Quantity	Minimum: 1	
	Details to be saved	Attributes in the table	Notes							
	Quantity	Quantity	Minimum: 1							
			[ALT]							
			Step 6: The system retrieves the updated product quantity from the Cart entity using a method in a .NET controller and an SQL READ query:							
			<table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Quantity</td><td>Quantity</td><td>Minimum: 1</td></tr></table>	Details to be saved	Attributes in the table	Notes	Quantity	Quantity	Minimum: 1	
	Details to be saved	Attributes in the table	Notes							
	Quantity	Quantity	Minimum: 1							
			and recalculates the cart total based on the updated quantity of each product in the user’s cart using a method in a .NET controller							
		Step 7: The system displays the updated product quantity’s and cart total								
ALT Step 3: The customer or reseller click on the “-” button Proceed to Step 4										
ALT Step 4: The customer or reseller’s increments to a value more than what is currently in stock The system restricts the incrementor and displays a bootstrap alert using the following controls										
<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>“Unable to increase quantity. We are receiving new stock soon!”</td><td>Label</td><td>This control is used to inform the customer/reseller that their specified quantity exceeds the current stock levels</td></tr></table>			Control Name	Control Name	Notes	“Unable to increase quantity. We are receiving new stock soon!”	Label	This control is used to inform the customer/reseller that their specified quantity exceeds the current stock levels		
Control Name	Control Name	Notes								
“Unable to increase quantity. We are receiving new stock soon!”	Label	This control is used to inform the customer/reseller that their specified quantity exceeds the current stock levels								
Terminate the use case										
ALT Step 5: The customer or reseller’s decrements the quantity of a specific product to the value of 0 The system removes the product from the Cart entity using a method in a .NET controller and an SQL DELETE query										
<table><tr><th>Details to be deleted</th><th>Attributes in the table</th></tr><tr><td>Cart Id</td><td>Id</td></tr><tr><td>Product Id</td><td>ProductId</td></tr><tr><td>Count</td><td>Count</td></tr></table>			Details to be deleted	Attributes in the table	Cart Id	Id	Product Id	ProductId	Count	Count
Details to be deleted	Attributes in the table									
Cart Id	Id									
Product Id	ProductId									
Count	Count									
Terminate the use case										
CONCLUSION:	The system displays the updated product quantity’s and cart total.									

POST-CONDITION:	The cart item's quantity is updated in the Cart entity.
BUSINESS RULES	• Different product prices are displayed for users with the reseller role.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• The cart total should be updated the moment that a customer or reseller changes the quantity of a product in their cart.
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 37 - 5.3 Remove Item(s) from Cart

ORDER SUBSYSTEM			
AUTHORS (s): Thenjiwe Ntsonda		DATE: 23/06/2022	
VERSION: 1.0		LAST REVIEW DATE: 23/06/2022	
USE CASE NAME:	Remove Item(s) from Cart	USE CASE TYPE	
USE CASE ID:	5.3	Business Requirements:	☐
PRIORITY:	High	System Analysis:	☐
SOURCE:	ByteXcom Requirements List	System Design:	☒
PRIMARY BUSINESS ACTOR	Customer or Reseller		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This use case begins when a customer or reseller wishes to remove an item which is currently displayed in their shopping cart. The customer or reseller will be able to select the option to remove a specific product from their cart. The use case concludes with an alert message that confirms the product/cart item has been removed successfully.		
PRE-CONDITION:	- The customer or reseller should have a registered account to be able to remove a product from their cart. - At least one product should already be added to the customer or reseller's cart in order for it to be requested for removal.		
TRIGGER:	This use case is called when a customer or reseller wishes to remove specific product(s) from their cart.		
TYPICAL COURSE OF EVENTS:	SYSTEM RESPONSE		
	ACTOR ACTION	MANUAL ACTION	AUTOMATED ACTION
	Step 1: The customer or reseller wishes to remove an item from their cart.		Step 2: The system extends use case 5.4 View Cart.

	Step 3: The customer or reseller clicks on the Remove icon next to the product they wish to remove from their cart.		Step 4: The system removes the selected product from the user's shopping cart and also removes it from the Cart entity using a method in a .NET controller.																				
			Step 5: The system reads the updated Cart entity details with the following attributes using an SQL READ query: <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Cart Item Quantity</td><td>CartItemQuantity</td><td>Quantity of all products added to their cart.</td></tr><tr><td>Cart Item Price</td><td>CartItemPrice</td><td>The price of each product added to the user's cart.</td></tr></table> From the Product entity: <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Product Name</td><td>Name</td><td>Name of each product in the user's cart.</td></tr><tr><td>Product Description</td><td>Description</td><td>Product Description of each product in the user's cart.</td></tr><tr><td>Product Image</td><td>ProductImageUrl</td><td>Path to image of each product added to the user's cart.</td></tr></table> The system then calculates the cart total based on the changes made using	Details to be read	Attributes in the table	Notes	Cart Item Quantity	CartItemQuantity	Quantity of all products added to their cart.	Cart Item Price	CartItemPrice	The price of each product added to the user's cart.	Details to be read	Attributes in the table	Notes	Product Name	Name	Name of each product in the user's cart.	Product Description	Description	Product Description of each product in the user's cart.	Product Image	ProductImageUrl
Details to be read	Attributes in the table	Notes																					
Cart Item Quantity	CartItemQuantity	Quantity of all products added to their cart.																					
Cart Item Price	CartItemPrice	The price of each product added to the user's cart.																					
Details to be read	Attributes in the table	Notes																					
Product Name	Name	Name of each product in the user's cart.																					
Product Description	Description	Product Description of each product in the user's cart.																					
Product Image	ProductImageUrl	Path to image of each product added to the user's cart.																					

			a method in a .NET controller and displays it to the user.
			Step 6: The system displays the updated cart item count on the screen
ALTERNATE COURSES:	None		
CONCLUSION:	This use case concludes when the selected product is removed from the user's shopping cart.		
POST-CONDITION:	The cart entity in the database is updated according to the removed product(s).		
BUSINESS RULES	- Customers and resellers should be allowed to remove a product from their cart at any time that it exists within the cart. - Different product prices are to be displayed for users with the reseller role only. - A product which has been added to a user's cart should be persisted for at least 7 days from creation.		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None		
ASSUMPTIONS:	The customer and reseller have a stable internet connection.		
OPEN ISSUES:	None		

Table 38 - 5.4 View Cart

ORDER SUBSYSTEM						
AUTHOR(s): Aphiwe Shoji				DATE: 04/07/2022		
VERSION: 1.0				LAST REVIEW DATE: 04/09/2022		
USE CASE NAME:		View Cart		USE CASE TYPE		
USE CASE ID:		5.4		Business Requirements: ☐		
PRIORITY:		medium		System Analysis: ☐		
SOURCE:		ByteXecom Requirements List		System Design: ☒		
PRIMARY BUSINESS ACTOR:		Customer/Reseller				
PRIMARY THE SYSTEM ACTOR:		None				
OTHER PARTICIPATING ACTORS:		None				
OTHER INTERESTED STAKEHOLDERS:		None				
DESCRIPTION:		The use case describes the event in which a customer or reseller wishes to view the contents of their cart. The customer/reseller requests to view their cart, the system in turn responds by reading the cart contents from the appropriate entities and finally displaying the cart contents as well as the corresponding prices and the cart total.				
PRE-CONDITION:		- The customer/reseller must have an account registered on the system - The customer/reseller needs to be logged in				
TRIGGER:		When a customer/reseller wishes content of their cart.				
TYPICAL COURSE OF EVENTS:		ACTOR ACTION:	SYSTEM RESPONSE:			
			Manual Action	Automated Action		
		Step 1:The customer or reseller wishes to view their cart.		Step 2: The system retrieves all of the cart information in the respective user's Cart entity using a method in a .NET controller and an SQL READ query		
				Details to be read	Attributes in the table	Notes

			Product Quantity	CartItemCount	The quantity of each product associated with the user's retrieved cart.
			Product Price	CartItemPrice	The price associated with each product in the user's retrieved cart.
			From the Product entity:		
			Details to be read	Attributes in the table	Notes
			Product Name	ProductName	The name of each product associated with the user's retrieved cart.
			The system then displays the Cart screen with the following controls:		
			Control Name	Control Type	Notes
			Cart	Heading	Page main heading
			Products	Subheading	Page subheading
			Name	Label	This control will be used to display the product's name
			Product image	image	This control used to display Product images
			Price	Label	This control will be used to display the product's price
			Quantity	Label	This control will be used to display the

					product's quantity
			Cart total	Label	This control will be used to display the cart total
			Count	Numeric updown	This control will be used to enter the count of products
			Continue Shopping	Button	When clicked, this control will direct the user to the product catalogue page
ALTERNATE COURSES	None				
CONCLUSION:	The contents of the cart are displayed to the customer/reseller.				
POST-CONDITION:	- All the user's cart details are retrieved from the Cart entity. - The user's cart details are displayed on the user interface.				
BUSINESS RULES:	- Users should only be able to see their cart details if they are logged into their account. - The user's cart details are to be persisted.				
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	The system is offline or the user does not have an internet connection, and they are not able to view their cart details.				
ASSUMPTIONS:	None				
OPEN ISSUES:	None				

Table 39 - 5.5 Checkout Cart

ORDER SUBSYSTEM			
AUTHORS (s): Kyle van Eeden		DATE: 05/07/2022	
VERSION: 1.2		LAST REVIEW DATE: 31/07/2022	
USE CASE NAME:	Checkout Cart		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒
USE CASE ID:	5.5		
PRIORITY:	High		
SOURCE:	ByteXcom Requirements List		
PRIMARY BUSINESS ACTOR	• Customer or Reseller		
PRIMARY SYSTEM ACTOR	• None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	This use case describes the event in which a customer or reseller wishes to check out their current cart on the system. Upon selecting the button to proceed to checkout, the system will retrieve the authenticated user's cart details from the Cart entity and the user's personal information from the User entity. The system displays the Order Summary screen which is populated with the user's current details stored on the system. All products are displayed along with their respective quantities assigned for the specific cart. A calculated cart total is displayed which sums all of the cart item amounts. The amount for each cart item is calculated by multiplying the price of the item with the requested quantity. The VAT percentage applicable to the order is retrieved from the VAT entity, this value is then calculated using the Cart Total in order to calculate the order's VAT inclusive amount. The reseller or customer enters the Delivery Details for the order, selects a courier service to be used and clicks on the option provided to confirm cart checkout. The system then validates the captured order information and invokes Use Case 5.6 "Make Order Payment" if validation passed for a customer checking out. Alternatively, when a reseller checks out, the use case will conclude with the order details being created in the Order entity. The system will clear the reseller's cart details stored in the Cart entity, an order confirmation email is sent to the reseller's email address and the use case concludes when the reseller is redirected to the Reseller Order Confirmation screen.		
PRE-CONDITION:	• The customer or reseller must be logged into their registered account. • The Cart Summary screen is displayed.		
TRIGGER:	The use case is triggered when a Customer or Reseller requests the system to check out their current cart.		
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE	
		MANUAL ACTION	AUTOMATED ACTION

Step 1: The Customer or Reseller clicks on the “**Proceed To Checkout**” button.

Step 2: The system retrieves the following details from the **Cart** entity using an ASP.NET Controller method with a SQL Read Query to identify all products along with their respective quantities and prices as per the user’s current cart:

Details to be read	Attributes in the table	Notes
Product	ProductID	Unique identifier of each product in the user’s cart.
Product Quantity	CartItemQuantity	Requested quantity.
Cart Item Price	CartItemPrice	Price per added product.

The names of all products in the user’s current cart is retrieved from the **Product** entity using a SQL Read Query:

Details to be read	Attributes in the table	Notes
Product Name	ProductName	Unique name of each product in the user’s cart

The system also retrieves the following details from the **User** entity using an ASP.NET Controller with a SQL Read Query for the specific user requesting to check out their cart:

Details to be read	Attributes in the table	Notes
First Name	FirstName	First Name attribute value of the user requesting to check out.
Surname	Surname	Surname attribute value of the user requesting to check out.
Email Address	Email	Email address value of the user.
Phone Number	PhoneNumber	Phone Number attribute.
Street Address	StreetAddress	Currently assigned Street Address attribute value.

The following location names are retrieved from the listed entities using an ASP.NET Controller method with an Entity Framework SQL Read query:

From the **Country** entity:

Details to be read	Attributes in the table	Notes
Country Name	CountryName	Name of the retrieved Country.

From the **Province** entity:

Details to be read	Attributes in the table	Notes
Province Name	ProvinceName	Name of the retrieved Province.

From the **City** entity:

Details to be read	Attributes in the table	Notes
City Name	CityName	Name of the retrieved City.

From the **Suburb** entity:

Details to be read	Attributes in the table	Notes
Suburb Name	SuburbName	Name of the retrieved Suburb.

A list of all couriers is retrieved from the **Courier** entity using an ASP.NET Controller method with an Entity Framework SQL Read Query:

Details to be read	Attributes in the table	Notes
Courier Name	CourierName	Name of each courier retrieved
Courier Fee	CourierFee	Delivery fee of each courier.

The current VAT percentage that is set as 'Active' is retrieved from the **VAT** entity using an ASP.NET Controller

			method with a LINQ and SQL Read Query:																					
			<table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>VATValue</td><td>VATPercentage</td><td>Returns decimal value stored for the current VAT Percentage that has an 'Active' status assigned.</td></tr></table>	Details to be read	Attributes in the table	Notes	VATValue	VATPercentage	Returns decimal value stored for the current VAT Percentage that has an 'Active' status assigned.															
	Details to be read	Attributes in the table	Notes																					
	VATValue	VATPercentage	Returns decimal value stored for the current VAT Percentage that has an 'Active' status assigned.																					
			Step 3: The system calculates the amount of all cart items by multiplying the cart item price value retrieved with the cart item quantity requested for each product within the user's cart. The CartTotal value is then calculated by summing the total of all cart items. The VAT inclusive amount for the order is calculated using an ASP.NET Controller method performing the following calculation: VATInclusive = (CartTotal * (VATAmount/100)).																					
			Step 4: The system displays the Delivery Details screen containing a ASP.NET form with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Control Notes</th></tr><tr><td>Order Overview</td><td>Header</td><td>Main heading of the page.</td></tr><tr><td>Delivery Details</td><td>Sub-Heading</td><td>Used as a heading for the delivery details section.</td></tr><tr><td>First Name</td><td>Label</td><td>Name label for the First Name textbox control.</td></tr><tr><td>First Name</td><td>Textbox</td><td>Control is populated with the user's FirstName attribute value retrieved.</td></tr><tr><td>Surname</td><td>Label</td><td>Name label for the Surname textbox control.</td></tr><tr><td>Surname</td><td>Textbox</td><td>Control is populated with the user's Surname</td></tr></table>	Control Name	Control Type	Control Notes	Order Overview	Header	Main heading of the page.	Delivery Details	Sub-Heading	Used as a heading for the delivery details section.	First Name	Label	Name label for the First Name textbox control.	First Name	Textbox	Control is populated with the user's FirstName attribute value retrieved.	Surname	Label	Name label for the Surname textbox control.	Surname	Textbox	Control is populated with the user's Surname
	Control Name	Control Type	Control Notes																					
	Order Overview	Header	Main heading of the page.																					
	Delivery Details	Sub-Heading	Used as a heading for the delivery details section.																					
First Name	Label	Name label for the First Name textbox control.																						
First Name	Textbox	Control is populated with the user's FirstName attribute value retrieved.																						
Surname	Label	Name label for the Surname textbox control.																						
Surname	Textbox	Control is populated with the user's Surname																						

					attribute value retrieved.
			Phone Number	Label	Name label for the Phone Number textbox control.
			Phone Number	Textbox	Control is populated with the user's PhoneNumber attribute value retrieved.
			Street Address	Label	Name label for the Street Number textbox control.
			Street Address	Textbox	Control is populated with the user's StreetNumber attribute value retrieved.
			Country	Label	Name label for the Country drop-down list control.
			Country	Drop-Down List	Populated with the CountryName attribute value retrieved.
			Province	Label	Name label for the Province drop-down list control.
			Province	Drop-Down List	Populated with the ProvinceName attribute value retrieved.
			City	Label	Name label for the City drop-down list control.
			City	Drop-Down List	Populated with the CityName attribute value retrieved.
			Suburb	Label	Name label for the Suburb drop-down list control.
			Suburb	Drop-Down List	Populated with the SuburbName attribute value retrieved.
			Select a Courier	Label	Name label of the Courier

			using a .NET controller with a SQL Read query and Ajax method to retrieve the following details:						
			<table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Courier Fee</td><td>CourierFee</td><td>The delivery fee for the selected courier.</td></tr></table>	Details to be read	Attributes in the table	Notes	Courier Fee	CourierFee	The delivery fee for the selected courier.
	Details to be read	Attributes in the table	Notes						
	Courier Fee	CourierFee	The delivery fee for the selected courier.						
			The system then recalculates the CartTotal value by adding the DeliveryFee value based on the selected courier using the following calculation: CartTotal += DeliveryFee						
Step 7: The Customer or Reseller updates the delivery details on the form.									
Step 8: The Customer or Reseller clicks the Place Order button. [ALT]		Step 9: The system validates the captured details to check that all required form details have been provided as per the Order entity's validation requirements. [ALT]							
		Step 10: The system uses a .NET controller with a SQL read query to retrieve the following details from the Role entity to determine if the user requesting to check out has a customer role assigned:							
		<table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Role ID</td><td>RoleID</td><td>The unique value which identifies the role assigned to the user's account.</td></tr></table> [ALT]	Details to be read	Attributes in the table	Notes	Role ID	RoleID	The unique value which identifies the role assigned to the user's account.	
Details to be read	Attributes in the table	Notes							
Role ID	RoleID	The unique value which identifies the role assigned to the user's account.							
		Step 11: The system identifies that the user making the request to check out is a customer, the system will retrieve the following details from the Product entity to determine whether there are still sufficient levels of the products available which the customer wishes to order:							

			Details to be read	Attributes in the table	Notes
			Product Quantity On Hand	QuantityOn Hand	The current quantity on hand of each product in the customer's cart.
			If there are sufficient levels of all products required, the system will invoke Use Case 5.6 "Make Order Payment" to allow the customer to make the payment for their order.		
ALTERNATE COURSES:	ALT Step 8: The Customer or Reseller no longer wishes to check out their current cart: • The Customer or Reseller clicks on the Back button. • The user is redirected to the View Cart Summary screen. • The use case concludes.				
	ALT Step 9 a : The details entered are not in the correct format or the required fields are empty • The system displays a validation error message box with the following controls				
	Control Name		Control Type		Notes
	Delivery Method		Drop-Down List		Outline border with red and display a validation error message stating that the Courier value is required.
	First Name		Textbox		Outline border with red and display a validation error message stating that the First Name value is required.
	Surname		Textbox		Outline border with red and display a validation error message stating that the Surname value is required.
	Phone Number		Textbox		Outline border with red and display a validation error message stating that the Phone Number value is required.
	Street Address		Textbox		Outline border with red and display a validation error message stating that the Street Address value is required.
	Country		Drop-Down List		Outline border with red and display a validation error message stating that the Country value is required.
	Province		Drop-Down List		Outline border with red and display a validation error message stating that the Province value is required.
	City		Drop-Down List		Outline border with red and display a validation error message stating that the City value is required.
	Suburb		Drop-Down List		Outline border with red and display a validation error

		message stating that the Suburb value is required.
Return to Step 7.		
ALT Step 9 b: The details entered is not valid as per the following validation requirements :		
Control Name	Control Type	Notes
First Name	Textbox	Red label indicating that first name cannot be longer than 50 characters.
Surname	Textbox	Red label indicating that surname value cannot be longer than 50 characters.
Phone Number	Textbox	Must contain 10 digits only.
Street Address	Textbox	Red label indicating that street address value cannot be longer than 50 characters.
Return to Step 7.		
ALT Step 10 a: There are not enough stock available of all products the customer wishes to order.		
The system will redirect the customer to the View Cart screen and display a bootstrap alert with the following controls:		
Control Name	Control Type	Notes
Natuurlik unfortunately no longer has the quantities available for each product requested.	Label	A message is displayed to the customer which informs them that the requested quantities of all products are no longer available.
The use case concludes.		
ALT Step 10 b: The user requesting to place an order is identified as a Reseller user:		
The system uses a .NET Controller method with an Entity Framework SQL Insert query to add the following reseller order details to the Order entity:		
Details to be Saved	Attribute in Table	Notes
First Name	FirstName	First Name value of the recipient specified for the order.
Surname	Surname	Surname value of the recipient specified for the order.
Phone Number	PhoneNumber	The contact number entered for the recipient of the order.
Street Address	StreetAddress	The street address of the delivery location captured.
Country	Country	Saves the CountryName value for the selected country which is referenced from the Country entity at the time of creation.
Province	Province	Saves the ProvinceName value for the selected country which is referenced

		from the Country entity at the time of creation.
City	City	Saves the CityName value for the selected country which is referenced from the Country entity at the time of creation.
Suburb	Suburb	Saves the SuburbName value for the selected country which is referenced from the Country entity at the time of creation.
Order Status	OrderStatus	The status value of the reseller's order is set to 'Pending' on creation to indicate that the order has not yet been confirmed.
Order Payment Status	OrderPaymentStatus	The order payment status value of the reseller's order is set to 'Delayed Reseller Payment' on creation.
Order Total	OrderTotal	The calculated order total value upon placing the order is saved.

- The system creates the following order details in the **OrderLine** entity using a .NET controller with a SQL Insert query:

Details to be Saved	Attribute in Table	Notes
Order Line ID	OrderLineID	The highest ID value currently stored in the OrderLine entity is incremented to assign a new, unique ID value. A new ID value is created for each product that has been ordered.
Order ID	OrderID	The order to which the order line instance is associated is referenced from the Order entity.
Product ID	ProductID	Each product that has been ordered is referenced from the Product entity. A new order line instance is created for each order-product association.
Price	Price	The price for each order line instance created is calculated using the following formula: Product Price x Quantity Ordered.
Product Count	Count	The quantity of each product which has been ordered is saved.

- The reseller is redirected to the **Reseller Order Confirmation** screen which contains the following controls:

Control Name	Control Type	Notes
'Thank you!'	Label	Main heading of the screen.

	'Your order has been placed successfully'	Label	Sub-heading of the screen displayed.
	'An email will be sent once your order has been approved.'	Label	Informs the reseller that they will be notified once their order has been approved or rejected.
	Order Number: [OrderID]	Label	Informs the user of what the order reference number is for the order they have placed, and which is referenced from the Order entity.
	View My Orders	Button	Navigates the reseller to the Placed Orders screen if clicked.
	The use case concludes.		
CONCLUSION:	This use case concludes when the system successfully has invoked Use Case 5.6 "Make Order Payment" or alternatively, for resellers successfully checking out, and the reseller is redirected to the Reseller Order Confirmation screen.		
POST-CONDITION:	The system has invoked Use Case 5.6 if the customer has requested to proceed to make the order payment. Alternatively, the reseller's order details is saved to the Order entity with a 'Pending' order status assigned and a 'Delayed Reseller Payment' order payment status.		
BUSINESS RULES	- Customers and Resellers should select the delivery method they wish to use out of the list of couriers made available. - All prices should be displayed in ZAR currency. - All product prices displayed are already inclusive of VAT and the VAT amount should be calculated accordingly. - Customers and Resellers should be allowed to change the delivery address details from what is currently stored for their user account. - Customers are required to make the payment in order for their order to be placed. - Resellers are not required to make a payment for their order for it to be placed on the system.		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	- The system is offline, and the Customer or Reseller will not be able to check out their cart. - On displaying the Order Summary screen, the form should be pre-populated with the applicable VAT percentage, the owner of the cart's registered details and the Cart totals should be calculated. - Customers or Resellers can only select a delivery location that is within the locations allowed as per the system's configuration. - Customers are to be restricted from ordering products at the specified quantities if Natuurlik no longer has sufficient stock available for all requested products.		
ASSUMPTIONS:	Customers or Resellers might need to change the delivery details based on who will be collecting the order as well as where the order should be delivered to.		
OPEN ISSUES:	None		

Table 40 - 5.6 Make Payment

ORDER SUBSYSTEM					
AUTHORS (s): Thenjiwe Ntsonda		DATE: 24/07/2022			
VERSION: 1.1		LAST REVIEW DATE: 29/07/2022			
USE CASE NAME:	Make Payment		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒		
USE CASE ID:	5.6				
PRIORITY:	High				
SOURCE:	ByteXcom Requirements List				
PRIMARY BUSINESS ACTOR	• Customer				
PRIMARY SYSTEM ACTOR	• None				
OTHER PARTICIPATING ACTORS:	• Payment Gateway Provider (ESA)				
OTHER INTERESTED STAKEHOLDERS:	• None				
DESCRIPTION:	This use case begins when the customer wishes to proceed with the payment for their current cart which includes products that have been added with their respective prices, the delivery details, selected courier service and inclusive VAT amount. The system will redirect the user to the Stripe payment gateway, where the payment for the user's cart will be processed. Once the payment has been confirmed for the user's cart, the system will create the order details and save it to the Order and OrderLine entities. The user's cart will be cleared, and an order confirmation email will be sent to the customer. The use case concludes when the user is redirected to the Customer Order Confirmation screen.				
PRE-CONDITION:	• The customer must be logged into their account. • The Checkout Cart screen has already been loaded.				
TRIGGER:	This use case is called when the customer wishes to proceed with the payment of an order which includes products that have been added to the user's cart.				
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE			
		MANUAL ACTION	AUTOMATED ACTION		
			Step 2: The system uses a .NET controller with a SQL Read Query to retrieve the following details from the Product entity to determine whether there are sufficient inventories on hand:		
			Details to be Read	Attributes in the table	Notes
			Quantity On Hand	QuantityOn Hand	The current quantity on hand of each

			<table><tr><td></td><td></td><td>product in the user's cart.</td></tr></table>			product in the user's cart.																	
			product in the user's cart.																				
		If there are sufficient inventory on hand the system redirects the customer to the Stripe payment gateway where the payment for the order will be made. [ALT]																					
		Step 3: The system receives a payment successful response from the Stripe payment gateway and the system saves the following order details to the Order table using an SQL INSERT query: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Order Number</td><td>OrderId</td><td>The highest order number stored on the system is incremented to assign a new unique order number.</td></tr><tr><td>User ID</td><td>ApplicationUserId</td><td>The unique identifier value used to identify the registered user who placed the order.</td></tr><tr><td>First Name</td><td>FirstName</td><td>The First Name value of the individual who will be receiving the order.</td></tr><tr><td>Surname</td><td>Surname</td><td>The Surname value of the individual who will be receiving the order.</td></tr><tr><td>Phone Number</td><td>PhoneNumber</td><td>The Phone Number value of the individual who will be receiving the order.</td></tr><tr><td>Street Address</td><td>StreetAddress</td><td>Populated with the Street Address value captured upon checking out.</td></tr></table>	Details to be saved	Attributes in the table	Notes	Order Number	OrderId	The highest order number stored on the system is incremented to assign a new unique order number.	User ID	ApplicationUserId	The unique identifier value used to identify the registered user who placed the order.	First Name	FirstName	The First Name value of the individual who will be receiving the order.	Surname	Surname	The Surname value of the individual who will be receiving the order.	Phone Number	PhoneNumber	The Phone Number value of the individual who will be receiving the order.	Street Address	StreetAddress	Populated with the Street Address value captured upon checking out.
Details to be saved	Attributes in the table	Notes																					
Order Number	OrderId	The highest order number stored on the system is incremented to assign a new unique order number.																					
User ID	ApplicationUserId	The unique identifier value used to identify the registered user who placed the order.																					
First Name	FirstName	The First Name value of the individual who will be receiving the order.																					
Surname	Surname	The Surname value of the individual who will be receiving the order.																					
Phone Number	PhoneNumber	The Phone Number value of the individual who will be receiving the order.																					
Street Address	StreetAddress	Populated with the Street Address value captured upon checking out.																					

			Order Date	OrderDate	Populated with the current system date and time value when the order details is created.
			Order Status	OrderStatus	Populated with the value “Processing” upon creating order details.
			Order Payment Status	OrderPaymentStatus	Populated with the value “Paid” upon creating order details.
			Order Total	OrderTotal	Total order calculated including the Delivery Fee for the selected courier service.
			Inclusive VAT Amount	InclusiveVAT	The total VAT amount applied to the order.
			Delivery Fee	DeliveryFee	Delivery fee of the selected courier selected. Referenced from the Courier entity.
The following details will be added to the OrderLine table:					
			Details to be saved	Attributes in the table	Notes
			Order Line ID	OrderLineId	The last created OrderLineId stored on the system will be incremented to create a unique identifier for each order-product association.
			Order ID	OrderId	The order with which each order line

					record is associated.														
			Product ID	ProductId	The unique identifier value for each product which has been ordered. Referenced from the Product entity.														
			Product Price	ProductPrice	The price paid for each product when the order was created.														
			Product Count	ProductCount	The quantity value of each product which was ordered.														
			Step 4: The system retrieves the following details for all products which have been ordered: From the Product entity:																
			<table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Quantity On Hand</td><td>QuantityOnHand</td><td>The current quantity on hand of each product ordered.</td></tr></table>			Details to be saved	Attributes in the table	Notes	Quantity On Hand	QuantityOnHand	The current quantity on hand of each product ordered.								
			Details to be saved	Attributes in the table	Notes														
	Quantity On Hand	QuantityOnHand	The current quantity on hand of each product ordered.																
	The system deducts the ProductCount value from the current QuantityOnHand value retrieved for each product which has been ordered.																		
			Step 5: The system uses a .NET controller method with a SQL Delete query to clear the user's cart details which is stored in the Cart entity:																
<table><tr><th>Details to be Deleted</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Cart ID</td><td>CartID</td><td>Removed.</td></tr><tr><td>Product ID</td><td>ProductID</td><td>Removed</td></tr><tr><td>Product Quantity</td><td>Count</td><td>Removed</td></tr><tr><td>User ID</td><td>UserID</td><td>Removed</td></tr></table>			Details to be Deleted	Attributes in the table	Notes	Cart ID	CartID	Removed.	Product ID	ProductID	Removed	Product Quantity	Count	Removed	User ID	UserID	Removed		
Details to be Deleted			Attributes in the table	Notes															
Cart ID			CartID	Removed.															
Product ID			ProductID	Removed															
Product Quantity			Count	Removed															
User ID			UserID	Removed															

		Step 6: The system retrieves the following details from the listed entities using a method in a .NET controller and an SQL READ query: From the Order entity: <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>User ID</td><td>UserID</td><td>Unique value that identifies the user who placed the order.</td></tr></table> From the User entity: <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Email Address</td><td>EmailAddresses</td><td>Email address stored for the user who placed the order.</td></tr></table> The system then generates an order confirmation email which is sent to the customer with the following details : <table><tr><th>Details</th><th>Notes</th></tr><tr><td>To</td><td>Populated with the EmailAddress value which was retrieved.</td></tr><tr><td>From</td><td>Populated with Natuurlik's email address.</td></tr><tr><td>Subject</td><td>Populated with the OrderID value retrieved and followed by static text that indicates the order was received.</td></tr><tr><td>Body</td><td></td></tr></table>	Details to be read	Attributes in the table	Notes	User ID	UserID	Unique value that identifies the user who placed the order.	Details to be read	Attributes in the table	Notes	Email Address	EmailAddresses	Email address stored for the user who placed the order.	Details	Notes	To	Populated with the EmailAddress value which was retrieved.	From	Populated with Natuurlik's email address.	Subject	Populated with the OrderID value retrieved and followed by static text that indicates the order was received.	Body	
Details to be read	Attributes in the table	Notes																						
User ID	UserID	Unique value that identifies the user who placed the order.																						
Details to be read	Attributes in the table	Notes																						
Email Address	EmailAddresses	Email address stored for the user who placed the order.																						
Details	Notes																							
To	Populated with the EmailAddress value which was retrieved.																							
From	Populated with Natuurlik's email address.																							
Subject	Populated with the OrderID value retrieved and followed by static text that indicates the order was received.																							
Body																								
		Step 7: The system sends the order confirmation to the customer using the Sendgrid API which has been configured.																						
		Step 8: The system displays the Customer Order Placed screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr></table>	Control Name	Control Type	Notes																			
Control Name	Control Type	Notes																						

			Green Checkbox	Icon	This control will be used to display a successful checkbox
			“Your order has been placed”	Label	This control will be used to inform the customer that their order has been placed successfully
			‘Order Number #[OrderId]”	Label	This control will be used to display the order number to the customer
			Browse Products	Button	This control will be used to redirect the customer to the Product Catalogue Screen.
ALTERNATE COURSES:	Step 2: There are no longer sufficient inventories available to fulfill the order based on the products in the customer’s cart: • The system redirects the customer to the View Cart screen. • The system displays a bootstrap alert with the following details: •				
	Control Name		Control Type		Notes
	Insufficient inventories available to place this order. New stock coming soon!		Message Box		A text message populates the message box control to inform the customer that they cannot place the order since there are not sufficient stock available to place the order.
CONCLUSION:	The system displays the Successful Order Placed Screen.				
POST- CONDITION:	The customer order details are saved to the Order and OrderLine entities. An order confirmation email is sent to the customer’s email address.				
BUSINESS RULES	• Only customers are required to pay for their order upfront. • The quantity on hand of each product ordered should be amended upon having successfully created the customer order details. • The order details are only to be created and saved after the payment gateway has informed that the payment for the order has been made successfully.				
IMPLEMENTATIO N CONSTRAINTS AND SPECIFICATIONS	• The customer will not be aware that their order cannot be fulfilled due to insufficient product levels until they request to make a payment for their order.				
ASSUMPTIONS:	• The customer has a stable internet connection.				
OPEN ISSUES:	• None				

Table 41 - 5.7 View Orders Overview

ORDER SUBSYSTEM										
AUTHORS (s): Nomusa Vumisa		DATE: 14/05/2022								
VERSION: 1.1		LAST REVIEW DATE: 24/05/2022								
USE CASE NAME:	View Orders Overview	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒								
USE CASE ID:	5.7									
PRIORITY:	High									
SOURCE:	ByteXpress Requirements List									
PRIMARY BUSINESS ACTOR	• Administrator/Sales Assistant									
PRIMARY SYSTEM ACTOR	• None									
OTHER PARTICIPATING ACTORS:	• None									
OTHER INTERESTED STAKEHOLDERS:	• None									
DESCRIPTION:	This use case describes an event where the administrator or sales assistant wants to view an order on the system. The use case begins when the administrator requests to view an order and the system responds by retrieving the details. The use case concludes when the details are displayed on the screen									
PRE-CONDITION:	The administrator or sales assistant must be logged on the system									
TRIGGER:	The use case is called when the administrator or sales assistant wants to view an order									
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE								
		MANUAL ACTION	AUTOMATED ACTION							
	Step 1: The administrator or sales assistant wishes to view an order on the system.		Step 2: The system invokes use case 5.8 Search Order							
	Step 3: The administrator or sales assistant clicks on the View Details button		Step 3: The system retrieves the following attributes from the following tables, using a method in a .NET controller and an SQL READ query: From the User table							
			<table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>First Name</td><td>FirstName</td></tr><tr><td>Surname</td><td>Surname</td></tr><tr><td>Phone Number</td><td>PhoneNumber</td></tr></table>	Details to be read	Attributes in the table	First Name	FirstName	Surname	Surname	Phone Number
Details to be read	Attributes in the table									
First Name	FirstName									
Surname	Surname									
Phone Number	PhoneNumber									

			Email Address	EmailAddress
			From the Order table	
			Details to be read	Attributes in the table
			Order Number	OrderId
			First Name	FirstName
			Surname	Surname
			Phone Number	PhoneNumber
			Street Address	StreetAddress
			Order Date	OrderDate
			Order Status	OrderStatus
			Order Total	OrderTotal
			Parcel Tracking Number	ParcelTrackingNumber
			Dispatched Date	DispatchedDate
			From the Country Table	
			Details to be read	Attributes in the table
			Country	CountryName
			From the Province Table	
			Details to be read	Attributes in the table
			Province	ProvinceName
			From the City Table	
			Details to be read	Attributes in the table
			City	CityName
			From the Suburb Table	
			Details to be read	Attributes in the table
			Suburb	SuburbName
			From the Product table	
			Details to be read	Attributes in the table
			Name	Name
			From the OrderLine table	
			Details to be read	Attributes in the table
			Price	OrderItemPrice

Count	OrderItemCount

and displays them on the **Order Details Screen** which contains an MVC form with the following controls:

Control Name	Control Type	Notes
Order Details	Heading	Main Heading
Order Number	Label	This control will be used to display the order number
Order Date	Label	This control will be used to display the order date
Order Status	Label	This control will be used to display the order status
Account Holder	Heading	Subheading
Full Name	Label	This control will be used to display the name of the owner of the account
Phone Number	Label	This control will be used to display the contact number of the owner of the account
Payment Status	Label	This control will be used to display the payment status
Approve	Button	This control will be used to approve a reseller order (Button is only displayed if the order status is "Pending")
Reject	Button	This control will be used to reject a reseller order (Button is only displayed if the order status is "Pending")
Backlog Order	Button	This control will be used to

					backlog an order (Button is only displayed if the order status is "Pending")
	Dispatch Order	Button			This control will be used to dispatch an order (Button is only displayed if the order status is "Processing")
	Capture Payment	Button			This control will be used to capture payment for a reseller order (Button is only displayed if the order status is "Processing" and the payment status is "Payment Outstanding")
	Cancel Order	Button			This control will be used to cancel an order (Button is only displayed if the order status is "Processing")
	Log Returned Product	Button			This control will be used to log a return (Button is only displayed if the order status is "Dispatched")
	Contact Person	Heading			Sub heading
	Full name	Label			This control will be used to display the recipient's Name
	Phone Number	Label			This control will be used to

					display the recipient's phone number
	Email Address				This control will be used to display the account owners email address
	Delivery Details	Heading			Sub heading
	Courier	Label			This control will be used to display the courier used
	Dispatched Date	Label			This control will be used to display the dispatch date
	Tracking Number	Label			This control will be used as a label to indicate the tracking number attribute
	Tracking Number	Textbox			This control will be used to display the tracking number (textbox is only displayed if the order status is "Processing")
	Order Summary	Heading			Sub heading
	Product Name	Label			This control will be used to display the Product Name
	Product Total	Label			This control will be used to display the Product Total
	Price	Label			This control will be used to display the individual products price
	Quantity	Label			This control will be used to display the order product quantity
	Total	Label			This control will be used as a label to indicate the order total
	Delivery Address	Heading			Sub heading

			Street Name	Label	This control will be used to display the recipient's street name
			Suburb	Label	This control will be used to display the recipient's suburb
			City	Label	This control will be used to display the recipient's city
			Province	Label	This control will be used to display the recipient's province
			Country	Label	This control will be used to display the recipient's country
			Postal Code	Label	This control will be used to display the recipient's Postal Code
			Back	Button	This control will be used to return to the Order List Screen
ALTERNATE COURSES:	None				
CONCLUSION:	This use case concludes when the selected order's details are displayed				
POST-CONDITION:	The User, Order, Suburb, City, Province, Country, Courier, Product and Order line Tables were read				
BUSINESS RULES	None				
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	The administrator or sales assistant cannot view their order details without an internet connection				
ASSUMPTIONS:	None				
OPEN ISSUES:	None				

Table 42 - 5.8 Search Order

ORDER SUBSYSTEM														
AUTHORS (s): Nomusa Vumisa		DATE: 11/07/2022												
VERSION: 1.1		LAST REVIEW DATE: 11/07/2022												
USE CASE NAME:	Search Order		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒											
USE CASE ID:	5.8													
PRIORITY:	Medium													
SOURCE:	ByteXpress Requirements List													
PRIMARY BUSINESS ACTOR	• Administrator/Sale Assistant													
PRIMARY SYSTEM ACTOR	• None													
OTHER PARTICIPATING ACTORS:	• None													
OTHER INTERESTED STAKEHOLDERS:	• None													
DESCRIPTION:	This use case describes the process where the administrator or sales assistant searches for an order on the system. The system responds by displaying the results matching the search query.													
PRE-CONDITION:	The administrator or sales assistant must be logged onto the system.													
TRIGGER:	This use case is called when the administrator or sales assistant wants to search for an order on the system.													
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE												
		MANUAL ACTION	AUTOMATED ACTION											
	Step 1: The administrator or sales assistant clicks on the Order navigation link on the admin side navigation bar		Step 2: The system retrieves the following attributes for each entry using a method in a .NET controller and an SQL READ query: From the User table											
			<table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Email Address</td><td>EmailAddress</td></tr></table>		Details to be read	Attributes in the table	Email Address	EmailAddress						
			Details to be read	Attributes in the table										
			Email Address	EmailAddress										
			From the Order table											
			<table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Order Number</td><td>OrderId</td></tr><tr><td>First Name</td><td>FirstName</td></tr><tr><td>Surname</td><td>Surname</td></tr><tr><td>Phone Number</td><td>PhoneNumber</td></tr></table>		Details to be read	Attributes in the table	Order Number	OrderId	First Name	FirstName	Surname	Surname	Phone Number	PhoneNumber
			Details to be read	Attributes in the table										
			Order Number	OrderId										
First Name	FirstName													
Surname	Surname													
Phone Number	PhoneNumber													

Order Date	OrderDate
Order Status	OrderStatus
Order Total	OrderTotal

and displays them on the **Order List Screen** with the following controls:

Control Name	Control Type	Notes
Orders List	Heading	Page main heading
Pending	Button	This control will be used to filter and display orders with a “Pending” status
Processing	Button	This control will be used to filter and display orders with a “Processing” status
Dispatched	Button	This control will be used to filter and display orders with a “Dispatched” status
Cancelled	Button	This control will be used to filter and display orders with a “Cancelled” status
Delayed	Button	This control will be used to filter and display orders with a “Delayed” status
Clear Filter	Button	This control will be used to clear the order filter
Show “” entries	Label and drop down	This control will be used to set the number of entries the administrator wants to display on a page
Search bar	Textbox	This control will be used to enter the search query

			Results table	Table	This control will be used to display all the orders that exist on the system
			Order Number	Table Heading	This control will be used as a heading to display the order number
			Order Date	Table Heading	This control will be used as a heading to display the order date
			First Name	Table Heading	This control will be used as a heading to display the recipient's Name
			Surname	Table Heading	This control will be used as a heading to display the recipient's Surname
			Email Address	Table Heading	This control will be used as a heading to display the recipient's Email Address
			Phone Number	Table Heading	This control will be used as a heading to display the recipient's Phone Number
			Status	Table Heading	This control will be used as a heading to display the order status
			Total	Table Heading	This control will be used as a heading to display the order total
			Action	Table Heading	This control will be used as a heading to display the actions the administrator can perform with each entry – View order details
			Action – View Details	Button	This control will be placed in a row, under the “Action”

					heading for each entry in the Order table. The button will direct the user to the Order Details page
			Showing “” of “” entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database
			“Previous” and “Next”	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on
	Step 4: The administrator or sales assistant enters their search query into the search textbox		Step 5: The system searches the table rows in the data table and displays the Order(s) matching the search query [ALT]		
	ALT Step 5: The search query entered does not match any existing records. • The system displays a “No Records Found” message in the results table. • Return to step 4 or terminate the use case				
CONCLUSION:	This use case concludes when the desired order details are displayed on the system.				
POST-CONDITION:	The Order and User entity was searched for and an entry that corresponds to the search query or parameter entered.				
BUSINESS RULES	• None				
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• None				
ASSUMPTIONS:	• None				
OPEN ISSUES:	• None				

Table 43 - 5.9 Query Order

ORDER SUBSYSTEM											
AUTHORS (s): Ofhani Mungani		DATE: 20/07/2022									
VERSION: 1.0		LAST REVIEW DATE: 29/07/2022									
USE CASE NAME:	Query Order		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒								
USE CASE ID:	5.9										
PRIORITY:	High										
SOURCE:	ByteXcom Requirements List										
PRIMARY BUSINESS ACTOR	• Customer • Reseller										
PRIMARY SYSTEM ACTOR	• None										
OTHER PARTICIPATING ACTORS:	• None										
OTHER INTERESTED STAKEHOLDERS:	• None										
DESCRIPTION:	This use case describes the process in which the customer/reseller wishes to query their order. The use case begins when the customer/reseller request to submit an order query, the system then retrieve the query reasons from the Query Reason entity. The use case concludes when the customer/reseller submits a query for an order and the system saves it to the business database.										
PRE-CONDITION:	• The customer/reseller must be logged onto the system.										
TRIGGER:	The use case is triggered when the customer/reseller wants to query an order on the system.										
TYPICAL COURSE OF EVENTS:		SYSTEM RESPONSE									
	ACTOR ACTION	MANUAL ACTION	AUTOMATED ACTION								
	Step 1: The customer/reseller wants to query their placed order.		Step 2: The system invokes use case 5.16 View Placed Order								
	Step 3: The customer/reseller clicks on the Query button		Step 4: The system retrieves the values with following attributes From the QueryReason entity: <table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Query Reason</td><td>Name</td></tr></table> From the OrderQuery entity: <table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Description</td><td>Description</td></tr></table>	Details to be read	Attributes in the table	Query Reason	Name	Details to be read	Attributes in the table	Description	Description
	Details to be read	Attributes in the table									
Query Reason	Name										
Details to be read	Attributes in the table										
Description	Description										

		The system then displays the Query Order Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Query Order</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Query Reason</td><td>Label</td><td>This control will be used as a label to indicate the Name attribute of the QueryReson</td></tr><tr><td>Query Reason</td><td>Dropdown</td><td>This control will be used by the customer/reseller to select a query reason for their order</td></tr><tr><td>Description</td><td>Label</td><td>This control will be used by the customer/reseller to indicate a query reason for the order</td></tr><tr><td>Description</td><td>Textbox</td><td>This control will be used by the customer/reseller to write a description for the order</td></tr><tr><td>Submit</td><td>Button</td><td>This control will be used to confirm the submission for an order query</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the query for an order</td></tr></table>	Control Name	Control Type	Notes	Query Order	Heading	Page main heading	Query Reason	Label	This control will be used as a label to indicate the Name attribute of the QueryReson	Query Reason	Dropdown	This control will be used by the customer/reseller to select a query reason for their order	Description	Label	This control will be used by the customer/reseller to indicate a query reason for the order	Description	Textbox	This control will be used by the customer/reseller to write a description for the order	Submit	Button	This control will be used to confirm the submission for an order query	Cancel	Button	This control will be used to cancel the query for an order
	Control Name	Control Type	Notes																							
	Query Order	Heading	Page main heading																							
	Query Reason	Label	This control will be used as a label to indicate the Name attribute of the QueryReson																							
	Query Reason	Dropdown	This control will be used by the customer/reseller to select a query reason for their order																							
	Description	Label	This control will be used by the customer/reseller to indicate a query reason for the order																							
	Description	Textbox	This control will be used by the customer/reseller to write a description for the order																							
	Submit	Button	This control will be used to confirm the submission for an order query																							
	Cancel	Button	This control will be used to cancel the query for an order																							
Step 5: The customer/reseller enters the details and clicks on the submit button [ALT]	Step 6: The system captures and validates the entered details against the OrderQuery table [ALT] <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Description</td><td>Description</td><td>Required</td></tr><tr><td>Query Reason</td><td>QueryReason</td><td>Required</td></tr></table>	Details to be read	Attributes in the table	Notes	Description	Description	Required	Query Reason	QueryReason	Required																
Details to be read	Attributes in the table	Notes																								
Description	Description	Required																								
Query Reason	QueryReason	Required																								
	Step 7: The system saves the information into the OrderQuery table using an INSERT SQL query:																									

			Details to be saved	Attributes in the table	Notes						
			ID	Id	A unique value generated by the system by checking from previous order query id and increment the new entry by one and saves to the database.						
			Description	Description	Required						
			Query Reason	Name	Required, linked by association form the Query Reason table						
			Order Number	OrderId	Required, linked by association form the Order table						
			Date	Date	The date is auto-generated using Date.Now function						
			Status	Status	The status is set to pending review						
			Rating	Rating	Required						
						Step 8: The system directs the customer/reseller to the View Order Details screen [View Placed Order use case 5.16] and displays a bootstrap alert notification with the following controls: Bootstrap notification:					
<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"Order Query Successfully Submitted"</td><td>Label</td><td>This control is used to indicate a successful creation</td></tr></table>						Control Name	Control Name	Notes	"Order Query Successfully Submitted"	Label	This control is used to indicate a successful creation
Control Name	Control Name	Notes									
"Order Query Successfully Submitted"	Label	This control is used to indicate a successful creation									

ALTERNATE COURSES:	ALT Step 5: The customer/reseller doesn't want to query their order anymore:

	• The customer/reseller clicks on the cancel button • The use case is terminated															
	ALT Step 6: The details entered are not in the correct format or the required fields are empty • The system displays a validation error message box with the following controls															
	<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Description</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>Description is required</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Description textbox</td></tr><tr><td>Query Reason</td><td>Drop-down</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>Query Reason</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Query Reason drop down</td></tr></table>	Control Name	Control Name	Notes	Description	Textbox	This control will be outline in red to show a validation error	Description is required	Label	This control will be used as a label to indicate a validation error in the Description textbox	Query Reason	Drop-down	This control will be outline in red to show a validation error	Query Reason	Label	This control will be used as a label to indicate a validation error in the Query Reason drop down
	Control Name	Control Name	Notes													
	Description	Textbox	This control will be outline in red to show a validation error													
	Description is required	Label	This control will be used as a label to indicate a validation error in the Description textbox													
Query Reason	Drop-down	This control will be outline in red to show a validation error														
Query Reason	Label	This control will be used as a label to indicate a validation error in the Query Reason drop down														
• Return to step 3																
CONCLUSION:	This use case concludes when an order query is submitted and a confirmation alert is displayed which confirms the order query has been created successfully.															
POST-CONDITION:	• The customer/reseller’s order query is added to the OrderQuery entity in the database.															
BUSINESS RULES	• Only the customer/reseller can query an order.															
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• The system is offline or the reseller/customer does not have an internet connection and they are unable to query their order.															
ASSUMPTIONS:	• None															
OPEN ISSUES:	• None															

Table 44 - 5.10 Review Order

ORDER SUBSYSTEM							
AUTHORS (s): Ofhani Mungani		DATE: 20/07/2022					
VERSION: 1.0		LAST REVIEW DATE: 20/07/2022					
USE CASE NAME:	Review Order		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒				
USE CASE ID:	5.10						
PRIORITY:	High						
SOURCE:	ByteXcom Requirements List						
PRIMARY BUSINESS ACTOR	• Customer • Reseller						
PRIMARY SYSTEM ACTOR	• None						
OTHER PARTICIPATING ACTORS:	• None						
OTHER INTERESTED STAKEHOLDERS:	• None						
DESCRIPTION:	This use case describes the process in which the customer/reseller wishes to review their order. The use case begins when the customer/reseller request to submit an order query, the system then retrieve the review reasons from the ReviewReason entity. The use case concludes when the customer/reseller submits a review for an order and the system saves it to the business database.						
PRE-CONDITION:	• The customer/reseller must be logged onto the system.						
TRIGGER:	The use case is triggered when the customer/reseller wants to query an order on the system.						
TYPICAL COURSE OF EVENTS:	ACTOR ACTION		SYSTEM RESPONSE				
			MANUAL ACTION				
			AUTOMATED ACTION				
	Step 1: The customer/reseller wants to query their placed order.		Step 2: The system invokes use case 5.16 View Placed Order				
Step 3: The customer/reseller clicks on the review button		Step 4: The system retrieves the values with following attributes From the ReviewReason entity: <table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Review Reason</td><td>Name</td></tr></table> The system then displays the Query Order Screen with the following controls:		Details to be read	Attributes in the table	Review Reason	Name
Details to be read	Attributes in the table						
Review Reason	Name						

			<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Query Order</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Query Reason</td><td>Label</td><td>This control will be used as a label to indicate the Name attribute of the ReviewReson</td></tr><tr><td>Review Reason</td><td>Dropdown</td><td>This control will be used by the customer/reseller to select a review reason for their order</td></tr><tr><td>Description</td><td>Label</td><td>This control will be used by the customer/reseller to indicate a review reason for the order</td></tr><tr><td>Description</td><td>Textbox</td><td>This control will be used by the customer/reseller to write a description for the order</td></tr><tr><td>Submit</td><td>Button</td><td>This control will be used to confirm the submission for an order review</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the review for an order</td></tr></table>	Control Name	Control Type	Notes	Query Order	Heading	Page main heading	Query Reason	Label	This control will be used as a label to indicate the Name attribute of the ReviewReson	Review Reason	Dropdown	This control will be used by the customer/reseller to select a review reason for their order	Description	Label	This control will be used by the customer/reseller to indicate a review reason for the order	Description	Textbox	This control will be used by the customer/reseller to write a description for the order	Submit	Button	This control will be used to confirm the submission for an order review	Cancel	Button	This control will be used to cancel the review for an order
	Control Name	Control Type	Notes																								
	Query Order	Heading	Page main heading																								
	Query Reason	Label	This control will be used as a label to indicate the Name attribute of the ReviewReson																								
	Review Reason	Dropdown	This control will be used by the customer/reseller to select a review reason for their order																								
	Description	Label	This control will be used by the customer/reseller to indicate a review reason for the order																								
	Description	Textbox	This control will be used by the customer/reseller to write a description for the order																								
	Submit	Button	This control will be used to confirm the submission for an order review																								
	Cancel	Button	This control will be used to cancel the review for an order																								
	Step 5: The customer/reseller enters the details and clicks on the submit button [ALT]		Step 6: The system captures and validates the entered details against the ReviewOrder table [ALT] <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Description</td><td>Description</td><td>Required</td></tr><tr><td>Rating</td><td>Rating</td><td>Required</td></tr><tr><td>Review Reason</td><td>ReviewReason</td><td>Required</td></tr></table>	Details to be read	Attributes in the table	Notes	Description	Description	Required	Rating	Rating	Required	Review Reason	ReviewReason	Required												
Details to be read	Attributes in the table	Notes																									
Description	Description	Required																									
Rating	Rating	Required																									
Review Reason	ReviewReason	Required																									
		Step 7: The system saves the information into the ReviewOrder table using an INSERT SQL query: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr></table>	Details to be saved	Attributes in the table	Notes																						
Details to be saved	Attributes in the table	Notes																									

			ID	Id	A unique value generated by the system by checking from previous order review id and increment the new entry by one and saves to the database.	
			Description	Description	Required	
			Review Reason	Name	Required, linked by association form the Review Reason table	
			Order Number	Id	Required, linked by association form the Order Reason table	
			Date	Date	Date is auto generated using Date.Now function	
			Step 8: The system directs the customer/reseller to the View Order Details screen [View Placed Order use case 5.16] and displays a bootstrap alert notification with the following controls: Bootstrap notification:			
			<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"Order Review Successfully Submitted"</td><td>Label</td><td>This control is used to indicate a successful creation</td></tr></table>	Control Name	Control Name	Notes
Control Name	Control Name	Notes				
"Order Review Successfully Submitted"	Label	This control is used to indicate a successful creation				
ALTERNATE COURSES:	ALT Step 5: The customer/reseller doesn't want to review their order anymore: - The customer/reseller clicks on the cancel button - The use case is terminated					
	ALT Step 6: The details entered are not in the correct format or the required fields are empty The system displays a validation error message box with the following controls					
	Control Name	Control Name	Notes			

	Description	Textbox	This control will be outline in red to show a validation error
	Description is required	Label	This control will be used as a label to indicate a validation error in the Description textbox
	Rating	Radio buttons	This control will be outline in red to show a validation error
	Rating is required	Label	This control will be used as a label to indicate a validation error in the Rating radio buttons
	Review Reason	Drop down list	This control will be outline in red to show a validation error
	Review Reason is required	Label	This control will be used as a label to indicate a validation error in the Review Reason drop-down list
	Return to step 3		
CONCLUSION:	This use case concludes when an order review is submitted and the system displays a confirmation message to inform the customer or reseller that their order review has been submitted successfully.		
POST-CONDITION:	The customer/reseller's order review is added to the ReviewOrder entity in the database.		
BUSINESS RULES	Only the customer/reseller can review an order which they have placed on the system.		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

Table 45 - 5.11 Create Courier

COURIER SUBSYSTEM																
AUTHORS (s): Thenjiwe Ntsonda			DATE: 14/05/2022													
VERSION: 1.0			LAST REVIEW DATE: 14/05/2022													
USE CASE NAME:	Create Courier		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒													
USE CASE ID:	5.11															
PRIORITY:	High															
SOURCE:	ByteXcom Requirements List															
PRIMARY BUSINESS ACTOR	• Administrator															
PRIMARY SYSTEM ACTOR	• None															
OTHER PARTICIPATING ACTORS:	• None															
OTHER INTERESTED STAKEHOLDERS:	• None															
DESCRIPTION:	This use case describes the process in which the administrator wishes to add a new courier to the system. The use case begins when the administrator adds all the pertinent information regarding the courier and save it to the business database.															
PRE-CONDITION:	• The administrator must be logged onto the system.															
TRIGGER:	The use case is triggered when the administrator wants to add a new courier which the administrator															
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE														
		MANUAL ACTION	AUTOMATED ACTION													
	Step 1: The administrator navigates to the Courier Screen under the Courier tab on the side bar		Step 2: The system retrieves courier details from the Courier entity with the following attributes:													
			<table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Courier Name</td><td>CourierName</td><td></td></tr><tr><td>Courier Fee</td><td>CourierFee</td><td></td></tr><tr><td>Estimated Delivery Time</td><td>EstimatedDeliveryTime</td><td></td></tr></table>		Details to be read	Attributes in the table	Notes	Courier Name	CourierName		Courier Fee	CourierFee		Estimated Delivery Time	EstimatedDeliveryTime	
			Details to be read	Attributes in the table	Notes											
			Courier Name	CourierName												
			Courier Fee	CourierFee												
			Estimated Delivery Time	EstimatedDeliveryTime												
			The system then displays the Courier Screen with the following controls:													
			<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Courier List</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Create New Courier</td><td>Button</td><td>When clicked, this control</td></tr></table>		Control Name	Control Type	Notes	Courier List	Heading	Page main heading	Create New Courier	Button	When clicked, this control			
Control Name	Control Type	Notes														
Courier List	Heading	Page main heading														
Create New Courier	Button	When clicked, this control														

					will direct the user to the Add Courier page
	Show "" entries	Label and drop down			This control will be used to set the number of entries the administrator wants to display on a page
	Search bar	Textbox			This control will be used when the user inserts the search parameters
	Results table	Table			This control will be used to display all the couriers that exist on the system
	Courier Name	Table heading			This control will be used as a heading to display the Courier Name for all couriers
	Courier Fee	Table heading			This control will be used as a heading to display the Courier Fee for all couriers
	Estimated Delivery Time	Table heading			This control will be used as a heading to display the Estimated Delivery Time for all couriers
	Action	Table heading			This control will be used as a heading to display the actions the administrator can perform with each entry – Edit and Delete
	Showing "" to "" of "" entries	Label			This control will be used to display the number of entries displayed per page as well as the total number of entries in the database

			"Previous" and "Next"	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on
	Step 3: The system administrator clicks on the Create New Courier button		Step 4: The system displays the Create Courier Screen with the following controls:		
			Control Name	Control Type	Notes
			Create Courier	Heading	Page main heading
			Courier Name	Label	This control will be used as a label to indicate the Courier Name attribute of the courier record
			Courier Name	Textbox	This control will be used by the administrator to insert a name of the courier
			Courier Fee	Label	This control will be used as a label to indicate the Courier Fee attribute of the courier record
			Courier Fee	Textbox	This control will be used by the administrator to insert a fee of the courier
			Estimated Delivery Time	Label	This control will be used as a label to indicate the Name attribute of the courier record
			Estimated Delivery Time	Textbox	This control will be used by the administrator to insert an estimated delivery time of the courier
Create	Button	This control will be used to confirm the			

			<table><tr><td></td><td></td><td>addition of the courier</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the addition of the courier</td></tr></table>			addition of the courier	Cancel	Button	This control will be used to cancel the addition of the courier								
			addition of the courier														
	Cancel	Button	This control will be used to cancel the addition of the courier														
	Step 5: The system administrator enters the details and clicks on the create button [ALT]		Step 6: The system captures and validates the entered details against the Courier table [ALT] <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Courier Name</td><td>CourierName</td><td>Required, Unique, No numerical characters</td></tr><tr><td>Courier Fee</td><td>CourierFee</td><td>Required, Numerical Characters Only</td></tr><tr><td>Estimated Delivery Time</td><td>EstimatedDeliveryTime</td><td>Required,</td></tr></table>	Details to be read	Attributes in the table	Notes	Courier Name	CourierName	Required, Unique, No numerical characters	Courier Fee	CourierFee	Required, Numerical Characters Only	Estimated Delivery Time	EstimatedDeliveryTime	Required,		
	Details to be read	Attributes in the table	Notes														
Courier Name	CourierName	Required, Unique, No numerical characters															
Courier Fee	CourierFee	Required, Numerical Characters Only															
Estimated Delivery Time	EstimatedDeliveryTime	Required,															
		Step 7: The system saves the information into the Courier table using an INSERT SQL query: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>ID</td><td>ID</td><td>A unique value generated by the system and saved to the database.</td></tr><tr><td>Courier Name</td><td>CourierName</td><td>Required, Unique,</td></tr><tr><td>Courier Fee</td><td>CourierFee</td><td>Required, Numerical Characters Only</td></tr><tr><td>Estimated Delivery Time</td><td>EstimatedDeliveryTime</td><td>Required,</td></tr></table>	Details to be saved	Attributes in the table	Notes	ID	ID	A unique value generated by the system and saved to the database.	Courier Name	CourierName	Required, Unique,	Courier Fee	CourierFee	Required, Numerical Characters Only	Estimated Delivery Time	EstimatedDeliveryTime	Required,
Details to be saved	Attributes in the table	Notes															
ID	ID	A unique value generated by the system and saved to the database.															
Courier Name	CourierName	Required, Unique,															
Courier Fee	CourierFee	Required, Numerical Characters Only															
Estimated Delivery Time	EstimatedDeliveryTime	Required,															
		Step 8: The system redirects the system administrator to the Courier Screen and displays a bootstrap alert notification with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"Courier created successfully"</td><td>Label</td><td>This control is used to indicate a successful creation</td></tr></table>	Control Name	Control Name	Notes	"Courier created successfully"	Label	This control is used to indicate a successful creation									
Control Name	Control Name	Notes															
"Courier created successfully"	Label	This control is used to indicate a successful creation															
ALTERNATE COURSES:	ALT Step 5: The administrator does not want to add a courier anymore The administrator clicks on the cancel button																

	The use case is terminated									
	ALT Step 6A: The details entered are not in the correct format or the required fields are empty									
	The system displays a validation error with the following controls									
	<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Name</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>Name is required</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Name textbox</td></tr></table>	Control Name	Control Name	Notes	Name	Textbox	This control will be outline in red to show a validation error	Name is required	Label	This control will be used as a label to indicate a validation error in the Name textbox
	Control Name	Control Name	Notes							
	Name	Textbox	This control will be outline in red to show a validation error							
	Name is required	Label	This control will be used as a label to indicate a validation error in the Name textbox							
	Return to step 3									
	ALT Step 6B: The courier already exists on the system									
	The system displays a modal dialog with the following controls									
<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Courier already exists!</td><td>Label</td><td>This control is used to inform the user that the user already exists</td></tr><tr><td>Okay</td><td>Button</td><td>This control is used to cancel the creation of the user role</td></tr></table>	Control Name	Control Name	Notes	Courier already exists!	Label	This control is used to inform the user that the user already exists	Okay	Button	This control is used to cancel the creation of the user role	
Control Name	Control Name	Notes								
Courier already exists!	Label	This control is used to inform the user that the user already exists								
Okay	Button	This control is used to cancel the creation of the user role								
- The administrator clicks the OK button - The use case is terminated										
CONCLUSION:	This use case concludes when a courier has been added on the system									
POST-CONDITION:	A new user has been added to the Courier table in the database									
BUSINESS RULES	Only the administrator has access to this function.									
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	The system is offline, and the system administrator will not be able to add a new courier.									
ASSUMPTIONS:	None									
OPEN ISSUES:	None									

Table 46 - 5.12 Search Courier

COURIER SUBSYSTEM															
AUTHORS (s): Thenjiwe Ntsonda		DATE: 14/05/2022													
VERSION: 1.0		LAST REVIEW DATE: 14/05/2022													
USE CASE NAME:	Search Courier		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒												
USE CASE ID:	5.12														
PRIORITY:	Medium														
SOURCE:	ByteXcom Requirements List														
PRIMARY BUSINESS ACTOR	• Administrator														
PRIMARY SYSTEM ACTOR	• None														
OTHER PARTICIPATING ACTORS:	• None														
OTHER INTERESTED STAKEHOLDERS:	• None														
DESCRIPTION:	This use case describes the process in which the administrator searches for product categories on the system. The system responds by displaying the information relating to the search query.														
PRE-CONDITION:	• The administrator must be logged onto the system.														
TRIGGER:	This use case is called when the administrator wants to search for a courier on the system.														
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE													
		MANUAL ACTION	AUTOMATED ACTION												
	Step 1: The administrator wishes to search for a courier on the system.														
	Step 2: The administrator navigates to the Courier Screen under the Order tab on the side bar		Step 3: The system retrieves courier details from the Courier entity with the following attributes: <table><tr><td>Details to be read</td><td>Attributes in the table</td><td>Notes</td></tr><tr><td>Courier Name</td><td>CourierName</td><td></td></tr><tr><td>Courier Fee</td><td>CourierFee</td><td></td></tr><tr><td>Estimated Delivery Time</td><td>EstimatedDeliveryTime</td><td></td></tr></table> The system then displays the Courier Screen with the following controls:	Details to be read	Attributes in the table	Notes	Courier Name	CourierName		Courier Fee	CourierFee		Estimated Delivery Time	EstimatedDeliveryTime	
	Details to be read	Attributes in the table	Notes												
Courier Name	CourierName														
Courier Fee	CourierFee														
Estimated Delivery Time	EstimatedDeliveryTime														

			Control Name	Control Type	Notes
			Courier List	Heading	Page main heading
			Create New Courier	Button	When clicked, this control will direct the user to the Create Courier Screen
			Show “” entries	Label and drop down	This control will be used to set the number of entries the administrator wants to display on a page
			Search bar	Textbox	This control will be used when the user inserts the search parameters
			Results table	Table	This control will be used to display all the couriers that exist on the system
			Courier Name	Table heading	This control will be used as a heading to display the Courier Name for all couriers
			Courier Fee	Table heading	This control will be used as a heading to display the Courier Fee for all couriers
			Estimated Delivery Time	Table heading	This control will be used as a heading to display the Estimated Delivery Time for all couriers
			Action	Table heading	This control will be used as a heading to display the actions the administrator can perform with each entry – Edit and Delete
			Showing “” to “” of “” entries	Label	This control will be used to display the number of entries

					displayed per page as well as the total number of entries in the database	
				"Previous" and "Next"	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on
	Step 4: The administrator enters their search query into the search textbox		Step 5: The system searches the Courier table and displays the courier(s) matching the search query in the Courier screen [ALT]			
	ALT Step 5: The search query entered does not match any records existing in the Courier entity. The system displays a "No Records Found" message in the Courier Screen and on the results table.					
CONCLUSION:	This use case concludes when the required courier is displayed on the screen.					
POST-CONDITION:	The administrator selects the required courier from the displayed results.					
BUSINESS RULES	Only the administrator has access to this function.					
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None					
ASSUMPTIONS:	None					
OPEN ISSUES:	None					

Table 47 - 5.13 Update Courier

COURIER SUBSYSTEM															
AUTHORS (s): Thenjiwe Ntsonda			DATE: 14/05/2022												
VERSION: 1.0			LAST REVIEW DATE: 14/05/2022												
USE CASE NAME:	Update Courier		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒												
USE CASE ID:	5.13														
PRIORITY:	Medium														
SOURCE:	ByteXcom Requirements List														
PRIMARY BUSINESS ACTOR	• Administrator														
PRIMARY SYSTEM ACTOR	• None														
OTHER PARTICIPATING ACTORS:	• None														
OTHER INTERESTED STAKEHOLDERS:	• None														
DESCRIPTION:	This use case describes the process in which the administrator wishes to update a courier on the system. The use case begins when the administrator updates all the pertinent information regarding the courier and save it to the business database.														
PRE-CONDITION:	• The administrator must be logged onto the system.														
TRIGGER:	The use case is triggered when the administrator wants to update a courier.														
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE													
		MANUAL ACTION	AUTOMATED ACTION												
	Step 1: The administrator wishes to update a courier in the system.		Step 2: The system invokes use case 5.12 Search Courier												
	Step 2: The administrator clicks on the update button of their desired courier.		Step 3: The system retrieves courier details from the Courier entity with the following attributes:												
			<table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Courier Name</td><td>CourierName</td><td></td></tr><tr><td>Courier Fee</td><td>CourierFee</td><td></td></tr><tr><td>Estimated Delivery Time</td><td>EstimatedDeliveryTime</td><td></td></tr></table> The system then displays the Update Courier Screen with the following controls:		Details to be read	Attributes in the table	Notes	Courier Name	CourierName		Courier Fee	CourierFee		Estimated Delivery Time	EstimatedDeliveryTime
Details to be read	Attributes in the table	Notes													
Courier Name	CourierName														
Courier Fee	CourierFee														
Estimated Delivery Time	EstimatedDeliveryTime														

			Control Name	Control Type	Notes
			Update Courier	Heading	Page main heading
			Courier Name	Label	This control will be used as a label to indicate the Courier Name attribute of the courier record
			Courier Name	Textbox	This control will be used by the administrator to update a name of the courier
			Courier Fee	Label	This control will be used as a label to indicate the Courier Fee attribute of the courier record
			Courier Fee	Textbox	This control will be used by the administrator to update a fee of the courier
			Estimated Delivery Time	Label	This control will be used as a label to indicate the Name attribute of the courier record
			Estimated Delivery Time	Textbox	This control will be used by the administrator to update an estimated delivery time of the courier
			Update	Button	This control will be used to confirm the addition of the courier
			Cancel	Button	This control will be used to cancel the addition of the courier
	Step 4: The system administrator enters the courier information and		Step 5: The system captures and validates the entered details against the Courier table [ALT]		
			Details to be saved	Attributes in the table	Notes

	clicks on the update button [ALT]		ID	ID	A unique value generated by the system and saved to the database.
			Courier Name	CourierName	Required, Unique,
			Courier Fee	CiurierFee	Required, Numerical Characters Only
			Estimated Delivery Time	EstimatedDeliveryTime	Required,
	Step 6: The system administrator clicks on the confirm button [ALT]		Step 7: The system displays a modal dialogue with the following controls:		
			Control Name	Control Type	Notes
			Confirm updated courier details?	Label	This control is used to request confirmation on the updated courier details
			Confirm	Button	This control is used to confirm the updating of the courier details
			Cancel	Button	This control is used to close the message box for the user to edit the courier details
			Step 8: The system saves the information into the Courier table using an UPDATE SQL query:		
			Details to be saved	Attributes in the table	Notes
			ID	ID	A unique value generated by the system and saved to the database.
			Courier Name	CourierName	Required, Unique,
			Courier Fee	CourierFee	Required, Numerical Characters Only
			Estimated Delivery Time	EstimatedDeliveryTime	Required,

			Step 9: The system redirects the system administrator to the Courier Screen and displays a bootstrap alert notification with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"Courier successfully updated"</td><td>Label</td><td>This control is used to indicate a successful update</td></tr></table>	Control Name	Control Name	Notes	"Courier successfully updated"	Label	This control is used to indicate a successful update			
Control Name	Control Name	Notes										
"Courier successfully updated"	Label	This control is used to indicate a successful update										
ALTERNATE COURSES:	ALT Step 4: The administrator does not wish to update the courier anymore - The administrator clicks on the cancel button - The use case is terminated											
	ALT Step 6: The administrator clicks on the cancel button The use case is terminated											
	ALT Step 7: The details entered are not in the correct format or the required fields are empty The system displays a validation bootstrap alert notification with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Name</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>Name is required</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Name textbox</td></tr></table>			Control Name	Control Name	Notes	Name	Textbox	This control will be outline in red to show a validation error	Name is required	Label	This control will be used as a label to indicate a validation error in the Name textbox
	Control Name	Control Name	Notes									
	Name	Textbox	This control will be outline in red to show a validation error									
Name is required	Label	This control will be used as a label to indicate a validation error in the Name textbox										
Return to step 3												
CONCLUSION:	This use case concludes when a courier has been added on the system											
POST-CONDITION:	A category has been updated in the Courier table in the database											
BUSINESS RULES	Only the administrator has access to this function.											
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None											
ASSUMPTIONS:	None											
OPEN ISSUES:	None											

Table 48 – 5.14 Delete Courier

COURIER SUBSYSTEM				
AUTHORS (s): Thenjiwe Ntsonda			DATE: 14/05/2022	
VERSION: 1.0			LAST REVIEW DATE: 14/05/2022	
USE CASE NAME:	Delete Courier		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒	
USE CASE ID:	5.14			
PRIORITY:	Medium			
SOURCE:	ByteXcom Requirements List			
PRIMARY BUSINESS ACTOR	• Administrator			
PRIMARY SYSTEM ACTOR	• None			
OTHER PARTICIPATING ACTORS:	• None			
OTHER INTERESTED STAKEHOLDERS:	• None			
DESCRIPTION:	This use case narrative describes the process in which the administrator wishes to delete a courier from the system. The use case begins with the administrator searching for the required courier selected for removal. The courier will be removed from the system database.			
PRE-CONDITION:	• The administrator must be logged onto the system.			
TRIGGER:	The use case is triggered when the administrator wants to delete a courier existing in the courier catalogue.			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE		
		MANUAL ACTION	AUTOMATED ACTION	
	Step 1: The administrator wishes to delete a courier in the system.		Step 2: The system invokes use case 5.12 Search Courier	
	Step 3: The administrator clicks on the Delete button of their desired courier		Step 4: The system retrieves courier details from the Courier entity with the following attributes:	

			The system then displays the Delete Courier Details Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Courier</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Courier Name</td><td>Label</td><td>This control will be used as a heading to display the Courier Name for the selected courier.</td></tr><tr><td>Courier Fee</td><td>Label</td><td>This control will be used as a heading to display the Courier Fee for the selected courier.</td></tr><tr><td>Estimated Delivery Time</td><td>Label</td><td>This control will be used as a heading to display the Estimated Delivery Time for the selected courier.</td></tr></table>	Control Name	Control Type	Notes	Courier	Heading	Page main heading	Courier Name	Label	This control will be used as a heading to display the Courier Name for the selected courier.	Courier Fee	Label	This control will be used as a heading to display the Courier Fee for the selected courier.	Estimated Delivery Time	Label	This control will be used as a heading to display the Estimated Delivery Time for the selected courier.
	Control Name	Control Type	Notes															
	Courier	Heading	Page main heading															
	Courier Name	Label	This control will be used as a heading to display the Courier Name for the selected courier.															
	Courier Fee	Label	This control will be used as a heading to display the Courier Fee for the selected courier.															
	Estimated Delivery Time	Label	This control will be used as a heading to display the Estimated Delivery Time for the selected courier.															
			Step 5: The system displays a modal dialogue with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Are you sure you want to delete this Courier?</td><td>Label</td><td>This control is used to request confirmation of the deletion of the courier</td></tr><tr><td>Confirm</td><td>Button</td><td>This control is used to confirm the deletion of the courier details</td></tr><tr><td>Cancel</td><td>Button</td><td>This control is used to close the message box deletion of the courier</td></tr></table>	Control Name	Control Type	Notes	Are you sure you want to delete this Courier?	Label	This control is used to request confirmation of the deletion of the courier	Confirm	Button	This control is used to confirm the deletion of the courier details	Cancel	Button	This control is used to close the message box deletion of the courier			
	Control Name	Control Type	Notes															
	Are you sure you want to delete this Courier?	Label	This control is used to request confirmation of the deletion of the courier															
	Confirm	Button	This control is used to confirm the deletion of the courier details															
Cancel	Button	This control is used to close the message box deletion of the courier																
Step 5: The system administrator clicks on the confirm button [ALT]		Step 6: The system deletes the selected courier from the Courier entity in the system database using an SQL DELETE query from the following attributes:																

			<table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Courier Name</td><td>CourierName</td><td></td></tr><tr><td>Courier Fee</td><td>CourierFee</td><td></td></tr><tr><td>Estimated Delivery Time</td><td>EstimatedDeliveryTime</td><td></td></tr></table>	Details to be read	Attributes in the table	Notes	Courier Name	CourierName		Courier Fee	CourierFee		Estimated Delivery Time	EstimatedDeliveryTime	
	Details to be read	Attributes in the table	Notes												
	Courier Name	CourierName													
	Courier Fee	CourierFee													
	Estimated Delivery Time	EstimatedDeliveryTime													
		[ALT] Step 7: The system redirects the system administrator to the Courier Screen and displays a bootstrap alert notification with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"Courier deleted successfully"</td><td>Label</td><td>This control is used to indicate a successful deletion</td></tr></table>	Control Name	Control Name	Notes	"Courier deleted successfully"	Label	This control is used to indicate a successful deletion							
Control Name	Control Name	Notes													
"Courier deleted successfully"	Label	This control is used to indicate a successful deletion													
ALTERNATE COURSES:	ALT Step 5: The administrator does not want to delete the courier anymore - The administrator clicks on the cancel button - The use case is terminated														
	ALT Step 6: The courier has attached to an existing record of an order placed on the system. The system displays a bootstrap alert notification with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Unsuccessful deletion</td><td>Label</td><td>This control is used to indicate an unsuccessful deletion</td></tr></table>			Control Name	Control Name	Notes	Unsuccessful deletion	Label	This control is used to indicate an unsuccessful deletion						
Control Name	Control Name	Notes													
Unsuccessful deletion	Label	This control is used to indicate an unsuccessful deletion													
CONCLUSION:	This use case concludes when a courier has been deleted on the system														
POST-CONDITION:	The courier has been removed from the system database.														
BUSINESS RULES	Only the administrator has access to this function.														
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None														
ASSUMPTIONS:	None														
OPEN ISSUES:	None														

Table 49 - 5.15 Search Placed Order

ORDER SUBSYSTEM				
AUTHORS (s): Nomusa Vumisa		DATE: 11/07/2022		
VERSION: 1.1		LAST REVIEW DATE: 11/07/2022		
USE CASE NAME:	Search Placed Order		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒	
USE CASE ID:	5.15			
PRIORITY:	Medium			
SOURCE:	ByteXpress Requirements List			
PRIMARY BUSINESS ACTOR	• Customer/Reseller			
PRIMARY SYSTEM ACTOR	• None			
OTHER PARTICIPATING ACTORS:	• None			
OTHER INTERESTED STAKEHOLDERS:	• None			
DESCRIPTION:	This use case describes the process where the customer/reseller searches for an order on the system. The system responds by displaying the results matching the search query.			
PRE-CONDITION:	The customer/reseller must be logged onto the system.			
TRIGGER:	This use case is called when the customer/reseller wants to search for an order on the system.			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE		
		MANUAL ACTION	AUTOMATED ACTION	
	Step 1: The customer/reseller clicks on the Order navigation link on the client side-navigation bar		Step 2: The system retrieves the following attributes for each entry using a method in a .NET controller and an SQL READ query: From the User table	
			Details to be read	Attributes in the table
			Email Address	EmailAddress
			From the Order table	
			Details to be read	Attributes in the table
			Order Number	OrderId
			First Name	FirstName
			Surname	Surname

Phone Number	PhoneNumber
Order Date	OrderDate
Order Status	OrderStatus
Order Total	OrderTotal

and displays them on the **My Orders Screen** with the following controls:

Control Name	Control Type	Notes
My Orders	Heading	Page main heading
Show "" entries	Label and drop down	This control will be used to set the number of entries the administrator wants to display on a page
Filters	Dropdown List	This control will be used to filter the orders by their status
Search bar	Textbox	This control will be used to enter the search query
Results table	Table	This control will be used to display all the orders that exist on the system
Order #	Table Heading	This control will be used as a heading to display the order number
Order Date	Table Heading	This control will be used as a heading to display the order date
First Name	Table Heading	This control will be used as a heading to display the recipient's Name
Surname	Table Heading	This control will be used as a heading to display the recipient's Surname
Phone Number	Table Heading	This control will be used as a heading to display the recipient's

					Phone Number
			Email Address	Table Heading	This control will be used as a heading to display the recipient's Email Address
			Status	Table Heading	This control will be used as a heading to display the order status
			Total	Table Heading	This control will be used as a heading to display the order total
			Action	Table Heading	This control will be used as a heading to display the actions the administrator can perform with each entry – View order details
			Action - Details	Button	This control will be placed in a row, under the "Action" heading for each entry in the Order table. The button will direct the user to the My Order Details page
			Showing "" to "" of "" entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database
			"Previous" and "Next"	Label and button	This control is used to navigate through the entry pages and to indicate on which page the user is currently on

	Step 4: The customer/reseller enters their search query into the search textbox		Step 5: The system searches the table rows in the data table and displays the Order(s) matching the search query [ALT]
	ALT Step 5: The search query entered does not match any existing records. - The system displays a “No Records Found” message in the results table. - Return to step 4 or terminate the use case		
CONCLUSION:	This use case concludes when the desired order details are displayed on the system.		
POST-CONDITION:	The Order and User entity was searched for and an entry that corresponds to the search query or parameter entered.		
BUSINESS RULES	None		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

Table 50 - 5.16 View Placed Order

ORDER SUBSYSTEM							
AUTHORS (s): Nomusa Vumisa		DATE: 14/05/2022					
VERSION: 1.1		LAST REVIEW DATE: 24/05/2022					
USE CASE NAME:	View Placed Order	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒					
USE CASE ID:	5.16						
PRIORITY:	High						
SOURCE:	ByteXpress Requirements List						
PRIMARY BUSINESS ACTOR	• Customer/Reseller						
PRIMARY SYSTEM ACTOR	• None						
OTHER PARTICIPATING ACTORS:	• None						
OTHER INTERESTED STAKEHOLDERS:	• None						
DESCRIPTION:	This use case describes an event where the customer/reseller wants to view their orders on the system. The use case begins when the user requests to view their orders and the system responds by retrieving the details. The use case concludes when the details are displayed on the screen						
PRE-CONDITION:	The customer/reseller must be logged on the system						
TRIGGER:	The use case is called when the customer/reseller wants to view their order details						
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE					
		MANUAL ACTION	AUTOMATED ACTION				
	Step 1: The customer/reseller wishes to view an order on the system.		Step 2: The system invokes use case 5.15 Search Order				
	Step 3: The customer/reseller clicks on the View Details button		Step 3: The system retrieves the following attributes from the following tables, using a method in a .NET controller and an SQL READ query: From the User table <table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Email Address</td><td>EmailAddress</td></tr></table> From the Order table	Details to be read	Attributes in the table	Email Address	EmailAddress
	Details to be read	Attributes in the table					
Email Address	EmailAddress						

			Details to be read	Attributes in the table
			Order Number	OrderId
			First Name	FirstName
			Surname	Surname
			Phone Number	PhoneNumber
			Street Address	StreetAddress
			Order Date	OrderDate
			Order Status	OrderStatus
			Order Total	OrderTotal
			Parcel Tracking Number	ParcelTrackingNumber
			Dispatched Date	DispatchedDate
			From the Country Table	
			Details to be read	Attributes in the table
			Country	CountryName
			From the Province Table	
			Details to be read	Attributes in the table
			Province	ProvinceName
			From the City Table	
			Details to be read	Attributes in the table
			City	CityName
			From the Suburb Table	
			Details to be read	Attributes in the table
			Suburb	SuburbName
			From the Product table	
			Details to be read	Attributes in the table
			Name	Name
			From the OrderLine table	
			Details to be read	Attributes in the table
			Price	OrderItemPrice
			Count	OrderItemCount

and displays them on the **My Order Details Screen** which contains an MVC form with the following controls:

Control Name	Control Type	Notes
My Order Details	Main Heading	This control will be used to display the main heading
Order Number	Heading	This control will be used to display the order number
Order Date	Label	This control will be used display the order date
Order Status	Label	This control will be used as a label to display the order status
Payment Status	Label	This control will be used display the payment status
Query Order	Button	This control will be used to send an order query
Review Order	Button	This control will be used to send an order review
Approve	Button	This control will be used by a reseller to approve their delayed order (Button is only displayed if the user role is "Reseller" and the order status is "Delayed")
Reject	Button	This control will be used by a reseller to reject their delayed order (Button is only displayed if the user role is "Reseller" and the order status is "Delayed")
Cancel Order	Button	This control will be used

					by a reseller to cancel their order (Button is only displayed if the user role is "Reseller" and the order status is "Pending")
	Contact Person	Heading			Sub heading
	First Name	Label			This control will be used to display the recipient's Name
	Surname	Label			This control will be used to display the recipient's Surname
	Phone Number	Label			This control will be used to display the recipient's phone number
	Email Address	Label			This control will be used to display the recipient's email address
	Delivery Address	Heading			Sub heading
	Street Name	Label			This control will be used by the administrator to display the recipient's street name
	Country	Label			This control will be used to display the recipient's country
	Province	Label			This control will be used to display the recipient's province
	City	Label			This control will be used to display the recipient's city
	Suburb	Label			This control will be used to display the recipient's suburb
	Postal Code	Label			This control will be used to display the recipient's Postal Code

			Delivery Details	Heading	Sub heading
			Courier	Label	This control will be used display the courier used to deliver the goods
			Tracking Number	Label	This control will be used display the tracking number
			Dispatched Date	Label	This control will be used display the dispatch date
			Order Summary	Heading	Sub heading
			Product Name	Label	This control will be used as a label to display the Product Name
			Product Total	Label	This control will be used as a label to display the Product Total
			Price	Label	This control will be used as a label to display the individual products price
			Quantity	Label	This control will be used as a label to display the order product quantity
			Total	Label	This control will be used as a label to indicate the order total
			Total	Label	This control will be used as a label to display the order total
			Back	Button	This control will be used to return the My Orders Screen
			Note: All controls are enabled by default unless stated otherwise		
			ALTERNATE COURSES:	• None	
CONCLUSION:	This use case concludes when the selected order's details are displayed				

POST-CONDITION:	The Order , Suburb , City , Province , Country , Courier , Product and Order line Tables were read
BUSINESS RULES	Customers and resellers should only be allowed to view the details of the orders they have placed.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	The user cannot view their order details without an internet connection
ASSUMPTIONS:	None
OPEN ISSUES:	None

Table 51 - 5.17 Cancel Placed Order

ORDER SUBSYSTEM			
AUTHORS (s): Nomusa Vumisa		DATE: 12/07/2022	
VERSION: 1.1		LAST REVIEW DATE: 12/07/2022	
USE CASE NAME:	Cancel Placed Order	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒	
USE CASE ID:	5.17		
PRIORITY:	Medium		
SOURCE:	ByteXcom Requirements List		
PRIMARY BUSINESS ACTOR	• Reseller		
PRIMARY SYSTEM ACTOR	• None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	This use case describes the process where the reseller cancels their order on the system. The reseller will search for the required order and then cancel their order. The use case concludes when the system notifies the reseller that the order has been cancelled.		
PRE-CONDITION:	• The reseller must be logged onto the system • The order status must be “Pending”		
TRIGGER:	This use case is called when the reseller wants to cancel an order on the system.		
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE	
		MANUAL ACTION	AUTOMATED ACTION
	Step 1: The reseller wishes to cancel an order on the system.		Step 2: The system invokes use case 5.16 View Placed Order

	Step 3: The reseller clicks on the Cancel Order button [ALT]		Step 4: The system displays a modal dialogue with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>"Confirm Order Cancellation"</td><td>Label</td><td>This control is used to request confirmation on the cancellation of the order</td></tr><tr><td>Confirm</td><td>Button</td><td>This control is used to confirm the cancelling of the order</td></tr><tr><td>Cancel</td><td>Button</td><td>This control is used to close the modal dialogue</td></tr></table>	Control Name	Control Type	Notes	"Confirm Order Cancellation"	Label	This control is used to request confirmation on the cancellation of the order	Confirm	Button	This control is used to confirm the cancelling of the order	Cancel	Button	This control is used to close the modal dialogue					
	Control Name	Control Type	Notes																	
	"Confirm Order Cancellation"	Label	This control is used to request confirmation on the cancellation of the order																	
Confirm	Button	This control is used to confirm the cancelling of the order																		
Cancel	Button	This control is used to close the modal dialogue																		
Step 5: The reseller clicks on the confirm button [ALT]		Step 6: The system updates the order status in the Order table using a method in a .NET controller and an SQL UPDATE query: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Order Status</td><td>OrderStatus</td><td>The order status is updated to "Cancelled"</td></tr></table>	Details to be saved	Attributes in the table	Notes	Order Status	OrderStatus	The order status is updated to "Cancelled"												
Details to be saved	Attributes in the table	Notes																		
Order Status	OrderStatus	The order status is updated to "Cancelled"																		
		Step 7: The system displays a bootstrap alert notification with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"Order status updated successfully"</td><td>Label</td><td>This control is used to indicate a successful cancellation</td></tr></table> And generates a cancellation email which is sent to the reseller with the following details <table><tr><th>Details</th><th>Notes</th></tr><tr><td>To</td><td>The resellers email address – retrieved from the User Table using an SQL READ query</td></tr><tr><td>From</td><td>The company email address, saved as a variable on the system</td></tr><tr><td>Subject</td><td>"Order Cancelled"</td></tr><tr><td>Body – Order Number</td><td>The order number retrieved from the Order table using an SQL READ query</td></tr><tr><td>Body – Order Date</td><td>The order date retrieved from the Order table using an SQL READ query</td></tr></table>	Control Name	Control Name	Notes	"Order status updated successfully"	Label	This control is used to indicate a successful cancellation	Details	Notes	To	The resellers email address – retrieved from the User Table using an SQL READ query	From	The company email address, saved as a variable on the system	Subject	"Order Cancelled"	Body – Order Number	The order number retrieved from the Order table using an SQL READ query	Body – Order Date	The order date retrieved from the Order table using an SQL READ query
Control Name	Control Name	Notes																		
"Order status updated successfully"	Label	This control is used to indicate a successful cancellation																		
Details	Notes																			
To	The resellers email address – retrieved from the User Table using an SQL READ query																			
From	The company email address, saved as a variable on the system																			
Subject	"Order Cancelled"																			
Body – Order Number	The order number retrieved from the Order table using an SQL READ query																			
Body – Order Date	The order date retrieved from the Order table using an SQL READ query																			

			Body – Payment Status	The payment status retrieved from the Order table using an SQL READ query
			Body – First Name	The users name retrieved from the User table using an SQL READ query
			Body – Main Text	“Hi (first name), Unfortunately your order has been cancelled. To query your order, click the button below”
			Button	The “Manage Order” button directs the user to their order on the system
			Step 8: The system sends the cancellation email to the reseller using the Sendgrid API	
ALTERNATE COURSES:	ALT Step 3: The reseller does not want to cancel the order - The reseller clicks on the Back button - The use case is terminated			
	ALT Step 5: The reseller does not confirm the cancellation of their order - The reseller clicks on the Cancel button - The use case is terminated			
CONCLUSION:	This use case concludes when the reseller is notified of the successful cancellation and an email is sent to them confirming the order cancellation			
POST-CONDITION:	The system updates the order status to “Cancelled”			
BUSINESS RULES	None			
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	Only orders with a Pending status can be cancelled			
ASSUMPTIONS:	None			
OPEN ISSUES:	None			

Table 52 - 5.18 Cancel Reseller Order

ORDER SUBSYSTEM				
AUTHORS (s): Nomusa Vumisa			DATE: 12/07/2022	
VERSION: 1.1			LAST REVIEW DATE: 12/07/2022	
USE CASE NAME:	Cancel Reseller Order		USE CASE TYPE	
USE CASE ID:	5.18		Business Requirements: ☐	
PRIORITY:	Medium		System Analysis: ☐	
SOURCE:	ByteXcom Requirements List		System Design: ☒	
PRIMARY BUSINESS ACTOR	Administrator/Sales Assistant			
PRIMARY SYSTEM ACTOR	None			
OTHER PARTICIPATING ACTORS:	Reseller			
OTHER INTERESTED STAKEHOLDERS:	None			
DESCRIPTION:	This use case describes the process where the administrator or sales assistant cancels a reseller order on the system. The administrator or sales assistant will search for the required order and then update the order status to cancelled. The use case concludes when the system notifies the administrator or sales assistant that the order has been cancelled and sends an email to the reseller informing them of the cancellation.			
PRE-CONDITION:	- The administrator or sales assistant must be logged onto the system. - The order status must be “Processing”			
TRIGGER:	This use case is called when the administrator or sales assistant wants to cancel an order on the system.			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE		
		MANUAL ACTION	AUTOMATED ACTION	
	Step 1: The administrator or sales assistant wishes to cancel an order on the system.		Step 2: The system invokes use case 5.7 View Order Overview	
	Step 3: The administrator or sales assistant clicks on the Cancel Order button [ALT]		Step 4: The system displays a modal dialogue with the following controls:	

					cancellation of the order
			Confirm	Button	This control is used to confirm the cancelling of the order
			Cancel	Button	This control is used to close the modal dialogue
	Step 5: The system administrator or sales assistant clicks on the confirm button [ALT]		Step 6: The system updates the order status in the Order table using a method in a .NET controller and an SQL UPDATE query:		
			Details to be saved	Attributes in the table	Notes
			Order Status	OrderStatus	The order status is updated to "Cancelled"
			Step 7: The system displays a bootstrap alert notification with the following controls:		
			Control Name	Control Name	Notes
			"Order status updated successfully"	Label	This control is used to indicate a successful cancellation
			And generates a cancellation email which is sent to the reseller with the following details		
			Details	Notes	
			To	The resellers email address – retrieved from the User Table using an SQL READ query	
			From	The company email address, saved as a variable on the system	
			Subject	"Order Cancelled"	
			Body – Order Number	The order number retrieved from the Order table using an SQL READ query	
			Body – Order Date	The order date retrieved from the Order table using an SQL READ query	
			Body – Payment Status	The payment status retrieved from the Order table using an SQL READ query	
			Body – First Name	The users name retrieved from the User table using an SQL READ query	
			Body – Main Text	"Hi (first name), Unfortunately your order has been cancelled. To query your order, click the button below"	

			Button	The “Manage Order” button directs the user to their order on the system
			Step 8: The system sends the cancellation email to the reseller using the Sendgrid API	
ALTERNATE COURSES:	ALT Step 3: The administrator or sales assistant does not want to cancel the order - The administrator or sales assistant clicks on the Back button - The use case is terminated			
	ALT Step 5: The administrator or sales assistant does not confirm the cancellation of the reseller order - The administrator or sales assistant clicks on the Cancel button - The use case is terminated			
CONCLUSION:	This use case concludes when the administrator or sales assistant is notified of the successful cancellation and an email is sent to the reseller informing them of the order cancellation			
POST-CONDITION:	The system updates the order status to “Cancelled”			
BUSINESS RULES	None			
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	Only the administrator and sales assistant can cancel an order.			
ASSUMPTIONS:	None			
OPEN ISSUES:	None			

Table 53 - 5.19 Approve/Reject Reseller Order

ORDER SUBSYSTEM				
AUTHORS (s): Nomusa Vumisa			DATE: 12/07/2022	
VERSION: 1.1			LAST REVIEW DATE: 12/07/2022	
USE CASE NAME:	Approve/Reject Reseller Order		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒	
USE CASE ID:	5.19			
PRIORITY:	Medium			
SOURCE:	ByteXcom Requirements List			
PRIMARY BUSINESS ACTOR	• Administrator			
PRIMARY SYSTEM ACTOR	• None			
OTHER PARTICIPATING ACTORS:	• Reseller			
OTHER INTERESTED STAKEHOLDERS:	• None			
DESCRIPTION:	This use case describes the process where the administrator approves or rejects a reseller order on the system. The administrator will search for the required order and confirm the approval/rejection of the order. The system then updated the order status. The use case concludes when the system notifies the administrator that the order has been approved/rejected.			
PRE-CONDITION:	• The administrator must be logged onto the system. • The order status must be “Pending”			
TRIGGER:	This use case is called when the administrator wants to approve or reject an order on the system.			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE		
		MANUAL ACTION	AUTOMATED ACTION	
	Step 1: The administrator wishes to approve/reject an order on the system.		Step 2: The system invokes use case 5.7 View Order Overview	
	Step 3: The administrator clicks on the Approve button [ALT]		Step 4: The system displays a modal dialogue with the following controls:	
			Control Name	Control Type
		“Confirm Order Approval/Rejection”	Label	This control is used to request confirmation on the approval/rejection of the order

			Confirm	Button	This control is used to confirm the approval/rejection of the order
			Cancel	Button	This control is used to close the modal dialogue
	Step 5: The system administrator clicks on the confirm button [ALT]		Step 6: The system updates the order status in the Order table using a method in a .NET controller and an SQL UPDATE query:		
			Details to be saved	Attributes in the table	Notes
			Order Status	OrderStatus	The order status is updated to “Processing” if approved or “Rejected“
		Step 7: The system displays a bootstrap alert notification with the following controls:			
		Control Name	Control Name	Notes	
		“Order status updated successfully”	Label	This control is used to indicate a successful approval/rejection	
		And generates an approval/rejection email which is sent to the reseller with the following details			
		Details	Notes		
		To	The resellers email address – retrieved from the User Table using an SQL READ query		
		From	The company email address, saved as a variable on the system		
		Subject	“Order Approved/Rejected”		
		Body – Order Number	The order number retrieved from the Order table using an SQL READ query		
		Body – Order Date	The order date retrieved from the Order table using an SQL READ query		
Body – Payment Status	The payment status retrieved from the Order table using an SQL READ query				
Body – First Name	The users name retrieved from the User table using an SQL READ query				
Body – Main Text (If order is approved)	“Hi (first name), Your order has been approved! An SMS will be sent once your order has been dispatched. To				

				query your order, click the button below”
			Body – Main Text (If order is rejected)	“Hi (first name), unfortunately, your order has been rejected. To query your order, click the button below”
			Button	The “Manage Order” button directs the user to their order on the system
			Step 8: The system sends the approval/rejection email to the reseller using the Sendgrid API	
ALTERNATE COURSES:	ALT Step 3A: The administrator clicks on the Reject button • Proceed to Step 6			
	ALT Step 3B: The administrator does not want to approve or reject the order • The administrator clicks on the Back button • The use case is terminated			
	ALT Step 5: The administrator does not confirm the approval/rejection of the reseller order • The administrator clicks on the cancel button • The use case is terminated			
CONCLUSION:	This use case concludes when the administrator is notified of the successful approval/rejection and an email is sent to the reseller informing them of their order status			
POST-CONDITION:	The system updates the order status to “Processing” if approved or to “Rejected” if rejected			
BUSINESS RULES	• None			
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• Only the administrator can approval/rejection an order. • Only orders with a Pending status can be approved or rejected			
ASSUMPTIONS:	• None			
OPEN ISSUES:	• None			

Table 54 - 5.20 Dispatch Order

ORDER SUBSYSTEM				
AUTHORS (s): Nomusa Vumisa		DATE: 12/07/2022		
VERSION: 1.1		LAST REVIEW DATE: 12/07/2022		
USE CASE NAME:	Dispatch Order		USE CASE TYPE	
USE CASE ID:	5.20		Business Requirements: ☐	
PRIORITY:	Medium		System Analysis: ☐	
SOURCE:	ByteXcom Requirements List		System Design: ☒	
PRIMARY BUSINESS ACTOR	Administrator/Sales Assistant			
PRIMARY SYSTEM ACTOR	None			
OTHER PARTICIPATING ACTORS:	Customer/Reseller			
OTHER INTERESTED STAKEHOLDERS:	None			
DESCRIPTION:	This use case describes the process where an order has been handed over to the couriers for delivery. The administrator or sales assistant will search for the required order, add the tracking number and then update the order status to dispatched. The use case concludes when the system notifies the administrator or sales assistant that the order has been dispatched and an email is sent to the recipient informing them of the dispatching of their order.			
PRE-CONDITION:	- The administrator or sales assistant must be logged onto the system. - The order status must be “Processing”			
TRIGGER:	This use case is called when the administrator or sales assistant wants to dispatch an order.			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE		
		MANUAL ACTION	AUTOMATED ACTION	
	Step 1: The administrator or sales assistant wishes to dispatch an order on the system.		Step 2: The system invokes use case 5.7 View Order Overview	
	Step 3: The administrator or sales assistant enter the parcel tracking number and click on the Dispatch Order button [ALT]		Step 4: The system captures and validates the entered details against the following attribute: [ALT] From the Order table	

		Step 5: The system displays a modal dialogue with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>“Confirm Dispatching of Order”</td><td>Label</td><td>This control is used to request confirmation on the dispatching of the order</td></tr><tr><td>Confirm</td><td>Button</td><td>This control is used to confirm the dispatching of the order</td></tr><tr><td>Cancel</td><td>Button</td><td>This control is used to close the modal dialogue</td></tr></table>	Control Name	Control Type	Notes	“Confirm Dispatching of Order”	Label	This control is used to request confirmation on the dispatching of the order	Confirm	Button	This control is used to confirm the dispatching of the order	Cancel	Button	This control is used to close the modal dialogue		
	Control Name	Control Type	Notes													
	“Confirm Dispatching of Order”	Label	This control is used to request confirmation on the dispatching of the order													
Confirm	Button	This control is used to confirm the dispatching of the order														
Cancel	Button	This control is used to close the modal dialogue														
Step 6: The system administrator or sales assistant clicks on the confirm button [ALT]		Step 7: The system updates the order status and saves the tracking number in the Order table using a method in a .NET controller and an SQL UPDATE query: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Order Status</td><td>OrderStatus</td><td>The order status is updated to “Dispatched”</td></tr><tr><td>Parcel Tracking Number</td><td>ParcelTrackingNumber</td><td>Provided by the courier company</td></tr></table>	Details to be saved	Attributes in the table	Notes	Order Status	OrderStatus	The order status is updated to “Dispatched”	Parcel Tracking Number	ParcelTrackingNumber	Provided by the courier company					
Details to be saved	Attributes in the table	Notes														
Order Status	OrderStatus	The order status is updated to “Dispatched”														
Parcel Tracking Number	ParcelTrackingNumber	Provided by the courier company														
		Step 8: The system displays a bootstrap alert notification with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>“Order status updated successfully”</td><td>Label</td><td>This control is used to indicate a successful status update</td></tr></table> And generates an order dispatched SMS which is sent to the recipient with the following details <table><tr><th>Details</th><th>Notes</th></tr><tr><td>To</td><td>The recipients phone number – retrieved from the Order Table using an SQL READ query</td></tr><tr><td>From</td><td>The company phone number, saved as a variable on the system</td></tr><tr><td>Body – First Name</td><td>The recipients first name retrieved from the Order table using an SQL READ query</td></tr></table>	Control Name	Control Name	Notes	“Order status updated successfully”	Label	This control is used to indicate a successful status update	Details	Notes	To	The recipients phone number – retrieved from the Order Table using an SQL READ query	From	The company phone number, saved as a variable on the system	Body – First Name	The recipients first name retrieved from the Order table using an SQL READ query
Control Name	Control Name	Notes														
“Order status updated successfully”	Label	This control is used to indicate a successful status update														
Details	Notes															
To	The recipients phone number – retrieved from the Order Table using an SQL READ query															
From	The company phone number, saved as a variable on the system															
Body – First Name	The recipients first name retrieved from the Order table using an SQL READ query															

			Body – Order Number	The order number retrieved from the Order table using an SQL READ query
			Body – Tracking Number	The parcel tracking number retrieved from the Order table using an SQL READ query
			Body – Main Text	Hi, (first name), your order #(order number) has been dispatched, your tracking number is (tracking number)
			Step 9: The system sends the order dispatched SMS to the recipient using the Twilio API	
ALTERNATE COURSES:	ALT Step 3: The administrator or sales assistant does not want to dispatch the order - Click on the Back button - The use case is terminated			
	ALT Step 4: The required field is empty The system displays a validation error using the following controls			
	Control Name		Control Name	Notes
	Tracking Number		Textbox	This control will be outline in red to show a validation error
	The Tracking Number field is required		Label	This control will be used as a label to indicate a validation error in the Tracking Number textbox
	Return to step 3			
	ALT Step 6: The administrator or sales assistant does not confirm the dispatching of the order - Click on the Cancel button - The use case is terminated			
CONCLUSION:	This use case concludes when the administrator or sales assistant is notified of the successful status update and a sms is sent to the recipient informing them of their order status			
POST-CONDITION:	The system updates the order status to “Dispatched”			
BUSINESS RULES	None			
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	Only the administrator and sales assistant can dispatch an order			
ASSUMPTIONS:	None			
OPEN ISSUES:	None			

Table 55 - 5.21 Log Back Order

ORDER SUBSYSTEM			
AUTHOR(s): Kyle van Eeden		DATE: 20/08/2022	
VERSION: 1.1		LAST REVIEW DATE: 29/08/2022	
USE CASE NAME:	Log Back Order	USE CASE TYPE	
USE CASE ID:	5.21	Business Requirements:	☐
PRIORITY:	High	System Analysis:	☐
SOURCE:	ByteXecom Requirements List	System Design:	☒
PRIMARY BUSINESS ACTOR:	Administrator		
PRIMARY THE SYSTEM ACTOR:	None		
OTHER PARTICIPATING ACTORS:	Reseller (ERA)		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This use case outlines the process in which an administrator backlogs a reseller order that has been placed on the system. The use case is initiated with the administrator clicking the Log Back Order button provided, and the system will prompt the administrator to confirm the back order operation by displaying a modal pop-up dialogue box. Once the administrator confirms that they wish to proceed to delay the order, the system will update the status of the order to "Delayed" and the change is saved in the Order entity. The system will retrieve the required details from the Order, User and ConfirmationReminder entities, respectively, which is used to generate a back order confirmation email that is sent to the reseller's email address to confirm that their order will be delayed. The use case concludes with the system displaying a bootstrap confirmation box which informs the administrator that the order has been backlogged successfully.		
PRE-CONDITION:	- The administrator must be logged into their account. - Use case 5.7: The View Order Details screen must be displayed on screen. - The order must have a status of "Pending" in order for it to be backlogged on the system.		
TRIGGER:	The administrator wishes to delay a reseller order placed on the system.		
TYPICAL COURSE OF EVENTS:	ACTOR ACTION:	SYSTEM RESPONSE:	
		Manual Action	Automated Action
	Step 1: The administrator clicks the Log		Step 2: The system uses a .NET Controller method to display a bootstrap modal dialogue box which

	Back Order button. [ALT]		prompts the administrator to confirm the back order operation: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"Do you wish to delay this order?"</td><td>Label</td><td>This control is used to prompt the administrator to confirm whether they wish to proceed.</td></tr><tr><td>Confirm</td><td>Button</td><td>Used to confirm adding order to orders backlog.</td></tr><tr><td>Cancel</td><td>Button</td><td>Used to cancel the operation.</td></tr></table> Additional Note(s): All controls are enabled by default, unless specified otherwise in the 'Notes' section.	Control Name	Control Name	Notes	"Do you wish to delay this order?"	Label	This control is used to prompt the administrator to confirm whether they wish to proceed.	Confirm	Button	Used to confirm adding order to orders backlog.	Cancel	Button	Used to cancel the operation.
	Control Name	Control Name	Notes												
"Do you wish to delay this order?"	Label	This control is used to prompt the administrator to confirm whether they wish to proceed.													
Confirm	Button	Used to confirm adding order to orders backlog.													
Cancel	Button	Used to cancel the operation.													
	Step 3: The administrator clicks on the Confirm button. [ALT]		Step 4: The system uses a .NET Controller method with a SQL UPDATE query to save the following details for the specific order to the Order entity: <table><tr><th>Details to be Updated</th><th>Attributes in the Table</th><th>Notes</th></tr><tr><td>Order Status</td><td>OrderStatus</td><td>The status of the order is changed to "Pending Confirmation".</td></tr><tr><td>Back Order Date</td><td>BackOrderDate</td><td>Populated with the system's date at the time of the request. Format: dd/mm/yyyy</td></tr></table>	Details to be Updated	Attributes in the Table	Notes	Order Status	OrderStatus	The status of the order is changed to "Pending Confirmation".	Back Order Date	BackOrderDate	Populated with the system's date at the time of the request. Format: dd/mm/yyyy			
Details to be Updated	Attributes in the Table	Notes													
Order Status	OrderStatus	The status of the order is changed to "Pending Confirmation".													
Back Order Date	BackOrderDate	Populated with the system's date at the time of the request. Format: dd/mm/yyyy													

		Step 5: The system will retrieve the following details from the list entities using a .NET controller with a SQL Read query: From the User entity: <table><tr><th>Details to be Read</th><th>Attributes in the Table</th><th>Notes</th></tr><tr><td>Email Address</td><td>EmailAddress</td><td>The email address of the reseller, who the order was placed by, is retrieved.</td></tr><tr><td>First Name</td><td>FirstName</td><td>The name of the reseller is retrieved.</td></tr></table> The following details are also retrieved from the Order entity: <table><tr><th>Details to be Read</th><th>Attributes in the Table</th><th>Notes</th></tr><tr><td>Order Number</td><td>OrderID</td><td>The unique reference number assigned to the reseller's order.</td></tr><tr><td>Order Status</td><td>OrderStatus</td><td>The current status of "Pending Confirmation" assigned to the order.</td></tr><tr><td>Order Payment Status</td><td>OrderPayment Status</td><td>The current payment status assigned to the order.</td></tr><tr><td>Order Date</td><td>OrderDate</td><td>The date on which the order was placed on the system.</td></tr></table>	Details to be Read	Attributes in the Table	Notes	Email Address	EmailAddress	The email address of the reseller, who the order was placed by, is retrieved.	First Name	FirstName	The name of the reseller is retrieved.	Details to be Read	Attributes in the Table	Notes	Order Number	OrderID	The unique reference number assigned to the reseller's order.	Order Status	OrderStatus	The current status of "Pending Confirmation" assigned to the order.	Order Payment Status	OrderPayment Status	The current payment status assigned to the order.	Order Date	OrderDate	The date on which the order was placed on the system.
Details to be Read	Attributes in the Table	Notes																								
Email Address	EmailAddress	The email address of the reseller, who the order was placed by, is retrieved.																								
First Name	FirstName	The name of the reseller is retrieved.																								
Details to be Read	Attributes in the Table	Notes																								
Order Number	OrderID	The unique reference number assigned to the reseller's order.																								
Order Status	OrderStatus	The current status of "Pending Confirmation" assigned to the order.																								
Order Payment Status	OrderPayment Status	The current payment status assigned to the order.																								
Order Date	OrderDate	The date on which the order was placed on the system.																								

			The currently active confirmation reminder time value is retrieved from the ConfirmationReminder entity: <table><tr><th>Details to be Read</th><th>Attributes in the Table</th><th>Notes</th></tr><tr><td>Confirmation Reminder Time</td><td>Value</td><td>The amount of days the reseller has to confirm the delayed order.</td></tr></table> The system then generates an order backlog confirmation email with the following controls: <table><tr><th>Details</th><th>Notes</th></tr><tr><td>To</td><td>The reseller's email address – retrieved from the User Table using an SQL READ query</td></tr><tr><td>From</td><td>The email address of Natuurlik</td></tr><tr><td>Subject</td><td>Order Delayed– Pending Confirmation</td></tr><tr><td>Body</td><td>Populated with the [OrderID], [OrderStatus], [OrderPaymentStatus], [OrderDate] and [FirstName] values retrieved to populate all relevant order details.</td></tr></table>	Details to be Read	Attributes in the Table	Notes	Confirmation Reminder Time	Value	The amount of days the reseller has to confirm the delayed order.	Details	Notes	To	The reseller's email address – retrieved from the User Table using an SQL READ query	From	The email address of Natuurlik	Subject	Order Delayed– Pending Confirmation	Body	Populated with the [OrderID] , [OrderStatus] , [OrderPaymentStatus] , [OrderDate] and [FirstName] values retrieved to populate all relevant order details.
Details to be Read	Attributes in the Table	Notes																	
Confirmation Reminder Time	Value	The amount of days the reseller has to confirm the delayed order.																	
Details	Notes																		
To	The reseller's email address – retrieved from the User Table using an SQL READ query																		
From	The email address of Natuurlik																		
Subject	Order Delayed– Pending Confirmation																		
Body	Populated with the [OrderID] , [OrderStatus] , [OrderPaymentStatus] , [OrderDate] and [FirstName] values retrieved to populate all relevant order details.																		
			Step 6: The system sends the generated email to the reseller's email address using the configured SendGrid API.																
			Step 7: The system navigates the administrator to the View Order Details screen and displays a bootstrap alert notification with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"Order added to backlog successfully."</td><td>Label</td><td>This control is used to confirm that the order has been delayed on the system.</td></tr></table>	Control Name	Control Name	Notes	"Order added to backlog successfully."	Label	This control is used to confirm that the order has been delayed on the system.										
Control Name	Control Name	Notes																	
"Order added to backlog successfully."	Label	This control is used to confirm that the order has been delayed on the system.																	

ALTERNATE COURSES	ALT Step 3: The administrator does not wish to proceed with logging the back order request: • The administrator clicks on the Cancel button. • The system displays the View Order Details screen. • The use case concludes.		
CONCLUSION:	The system displays a bootstrap alert which informs the administrator that the order has been added to the orders backlog successfully.		
POST-CONDITION:	• The order's status is updated to "Delayed" and saved in the Order entity. • A back order confirmation email is sent to the reseller's email address.		
BUSINESS RULES:	• Only the administrator of the system should be allowed to modify a reseller order's details. • Only reseller orders which is in a "Pending" state can be backlogged by an administrator.		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	• An email should be sent to the reseller which informs them that their placed order has been assigned to the orders backlog by Natuurlik and that it will therefore be delayed. • Only reseller orders with a status of "Pending" assigned can be placed on the orders backlog by an administrator user.		
ASSUMPTIONS:	• The administrator has confirmed the current quantity on hand of all ordered products are insufficient to process the order at the time and wishes to delay the fulfilment of the order.		
OPEN ISSUES:	• None		

Table 56 - 5.22 Confirm Order

ORDER SUBSYSTEM				
AUTHOR(s): Kyle van Eeden			DATE: 20/08/2022	
VERSION: 1.1			LAST REVIEW DATE: 29/08/2022	
USE CASE NAME:	Confirm Order		USE CASE TYPE	
USE CASE ID:	5.22		Business Requirements: ☐	
PRIORITY:	High		System Analysis: ☐	
SOURCE:	ByteXecom Requirements List		System Design: ☒	
PRIMARY BUSINESS ACTOR:	Reseller			
PRIMARY THE SYSTEM ACTOR:	None			
OTHER PARTICIPATING ACTORS:	None			
OTHER INTERESTED STAKEHOLDERS:	None			
DESCRIPTION:	This use case outlines the process in which a reseller confirms an order which they have placed on the system and which has been assigned to the orders backlog. The use case is initiated once the reseller clicks the Confirm Order button to indicate they wish to accept the delayed order. The system will then prompt the reseller to confirm whether they wish to proceed with accepting the order. Once the reseller confirms that they wish to proceed, the system updates the order's status to "Processing" and generates an email which is sent to the reseller's email address to inform them that the backlogged has been accepted successfully. The use case concludes when the system displays a bootstrap confirmation alert which informs the reseller that the delayed order has been successfully accepted.			
PRE-CONDITION:	- The reseller must be logged into their account. - Use case 5.7: The View Order Details screen must be displayed on screen. - The reseller's order status assigned must be "Delayed" in order to initiate this use case.			
TRIGGER:	The reseller wishes to confirm their delayed order on the system.			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION:	SYSTEM RESPONSE:		
		Manual Action	Automated Action	
	Step 1: The reseller clicks on the Accept Order button. [ALT]		Step 2: The system displays a bootstrap modal dialogue box with the following controls:	
		Control Name	Control Type	Notes

			<table><tr><td>"Proceed to accept this delayed order?"</td><td>Label</td><td>Prompts reseller to confirm whether they wish to accept the delayed order.</td></tr><tr><td>Cancel</td><td>Button</td><td>Cancels the order confirmation operation if clicked.</td></tr><tr><td>Confirm</td><td>Button</td><td>Allows the reseller to confirm the delayed order if clicked.</td></tr></table>	"Proceed to accept this delayed order?"	Label	Prompts reseller to confirm whether they wish to accept the delayed order.	Cancel	Button	Cancels the order confirmation operation if clicked.	Confirm	Button	Allows the reseller to confirm the delayed order if clicked.
			"Proceed to accept this delayed order?"	Label	Prompts reseller to confirm whether they wish to accept the delayed order.							
			Cancel	Button	Cancels the order confirmation operation if clicked.							
			Confirm	Button	Allows the reseller to confirm the delayed order if clicked.							
Additional Note(s): All controls are enabled by default, unless specified otherwise in the 'Notes' section.												
Step 3: The reseller clicks on the Confirm button. [ALT]		Step 4: The system uses a .NET Controller method with a SQL UPDATE query to save the following details to the Order entity: <table><tr><th>Details to be Updated</th><th>Attributes in the Table</th><th>Notes</th></tr><tr><td>Order Status</td><td>OrderStatus</td><td>The status of the order is changed to "Processing".</td></tr></table>	Details to be Updated	Attributes in the Table	Notes	Order Status	OrderStatus	The status of the order is changed to "Processing".				
Details to be Updated	Attributes in the Table	Notes										
Order Status	OrderStatus	The status of the order is changed to "Processing".										
		Step 5: The system uses a .NET controller method with a SQL Read query to retrieve the following details from the listed entities: From the User entity: <table><tr><th>Details to be Read</th><th>Attributes in the Table</th><th>Notes</th></tr><tr><td>First Name</td><td>FirstName</td><td>The FirstName attribute value stored for the reseller's account.</td></tr></table>	Details to be Read	Attributes in the Table	Notes	First Name	FirstName	The FirstName attribute value stored for the reseller's account.				
Details to be Read	Attributes in the Table	Notes										
First Name	FirstName	The FirstName attribute value stored for the reseller's account.										

			Email Address	EmailAddress	The EmailAddress attribute value stored for the reseller's account.
			From the Order entity:		
			Details to be Read	Attributes in the Table	Notes
			Order Number	OrderID	The unique reference number assigned to the reseller order.
			Order Status	OrderStatus	The current status of "Delayed" assigned to the order.
			Order Payment Status	OrderPayment Status	The current payment status assigned to the reseller order.
			Order Date	OrderDate	The date on which the order was initially placed.
			The system then generates an email which is populated using the above details retrieved and with the following controls:		
			Details	Notes	
			To	Populated with the retrieved [EmailAddress] value.	
From	Populated with Natuurlik's hard-coded business email address.				
Subject	"Delayed Order Confirmed Successfully"				
Body	Populated with the retrieved [OrderID], [OrderStatus], [OrderPaymentStatus], [OrderDate] and [FirstName] values retrieved.				

			Step 6: The system sends the generated email with to the reseller’s email address using the configured SendGrid API .						
			Step 7: The system displays the View Order Details screen with a bootstrap alert notification that contains the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>“Order confirmed successfully .”</td><td>Label</td><td>This control is used to inform the reseller that the backlogged order has been accepted successfully.</td></tr></table>	Control Name	Control Name	Notes	“Order confirmed successfully .”	Label	This control is used to inform the reseller that the backlogged order has been accepted successfully.
Control Name	Control Name	Notes							
“Order confirmed successfully .”	Label	This control is used to inform the reseller that the backlogged order has been accepted successfully.							
ALTERNATE COURSES	ALT Step 3 : The reseller no longer wishes to accept the delayed order: - The reseller clicks on the Cancel button. - The View Order Details screen is displayed. - The use case is terminated.								
CONCLUSION:	The system displays a bootstrap alert which informs the reseller that the backlogged order has been confirmed successfully.								
POST-CONDITION:	- The order’s status is updated to “Processing” and saved in the Order entity. - An order confirmation email is sent to the reseller’s email address.								
BUSINESS RULES:	- A reseller should only be allowed to confirm an order placed by them, and which has been assigned to the orders backlog by Natuurlik. - The reseller only has five days to confirm their order after it has been added to the orders backlog.								
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	- Only orders with a status of “Delayed” can be confirmed or rejected by the reseller. - An email is to be generated by the system and sent to the reseller’s email address to inform them that the order has been accepted/confirmed successfully. - The system is to cancel the delayed order 7 days after it has been added to the orders backlog.								
ASSUMPTIONS :	- The reseller has access to the internet and the system is not offline. - The reseller has reviewed the email sent for the specific order which has been delayed and still wishes to accept the order.								
OPEN ISSUES:	None								

Table 57 - 5.23 Reject Order

ORDER SUBSYSTEM					
AUTHOR(s): Kyle van Eeden			DATE: 20/08/2022		
VERSION: 1.1			LAST REVIEW DATE: 29/08/2022		
USE CASE NAME:	Reject Order		USE CASE TYPE		
USE CASE ID:	5.23		Business Requirements: ☐		
PRIORITY:	High		System Analysis: ☐		
SOURCE:	ByteXecom Requirements List		System Design: ☒		
PRIMARY BUSINESS ACTOR:	Reseller				
PRIMARY THE SYSTEM ACTOR:	None				
OTHER PARTICIPATING ACTORS:	None				
OTHER INTERESTED STAKEHOLDERS:	None				
DESCRIPTION:	This use case outlines the process in which a reseller rejects an order which they have placed on the system and which has been added to Natuurlik's orders backlog from the time of having placed the order. The use case is initiated once the reseller clicks on the Reject Order button. The system will prompt the reseller to confirm that they wish to reject the delayed order and the system will then update the order's status to "Rejected" and save it to the Order entity. The use case concludes when the system displays a bootstrap confirmation alert which informs the reseller that the order has been successfully rejected.				
PRE-CONDITION:	- The reseller must be logged into their account. - Use case 5.7: The View Order Details screen must be displayed on screen. - The reseller's order status assigned must be "Delayed" in order to initiate this use case.				
TRIGGER:	The reseller wishes to reject the backlogged order and clicks the Reject Order button.				
TYPICAL COURSE OF EVENTS:	ACTOR ACTION:	SYSTEM RESPONSE:			
		Manual Action	Automated Action		
	Step 1: The reseller clicks on the Reject Order button. [ALT]		Step 2: The system displays a modal dialogue box with the following controls:		
			Control Name	Control Type	Notes
		"Do you wish to reject this order?"	Label	Prompts reseller to confirm the rejection of the order.	

			<table><tr><td>Cancel</td><td>Button</td><td>Terminates the order rejection operation if clicked.</td></tr><tr><td>Confirm</td><td>Button</td><td>Rejects the order if clicked.</td></tr></table>	Cancel	Button	Terminates the order rejection operation if clicked.	Confirm	Button	Rejects the order if clicked.
	Cancel	Button	Terminates the order rejection operation if clicked.						
	Confirm	Button	Rejects the order if clicked.						
Additional Note(s): All controls are enabled by default, unless specified otherwise in the 'Notes' section.									
	Step 3: The reseller clicks on the Confirm button. [ALT]		Step 4: The system uses a .NET Controller method with a SQL UPDATE query to save the following details to the Order entity: <table><tr><th>Details to be Updated</th><th>Attributes in the Table</th><th>Notes</th></tr><tr><td>Order Status</td><td>OrderStatus</td><td>The status of the order is changed to "Rejected".</td></tr></table>	Details to be Updated	Attributes in the Table	Notes	Order Status	OrderStatus	The status of the order is changed to "Rejected".
	Details to be Updated	Attributes in the Table	Notes						
Order Status	OrderStatus	The status of the order is changed to "Rejected".							
			Step 5: The system displays a bootstrap alert notification on the View Order Details screen with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"Order rejected successfully."</td><td>Label</td><td>This control is used to inform the reseller that the order has been rejected successfully.</td></tr></table>	Control Name	Control Name	Notes	"Order rejected successfully."	Label	This control is used to inform the reseller that the order has been rejected successfully.
Control Name	Control Name	Notes							
"Order rejected successfully."	Label	This control is used to inform the reseller that the order has been rejected successfully.							
ALTERNATE COURSES	ALT Step 3 : The reseller no longer wishes to reject the backlogged order: - The reseller clicks on the Cancel button. - The use case is terminated.								
CONCLUSION:	The system displays a bootstrap alert which informs the reseller that the order has been rejected successfully.								
POST-CONDITION:	The order's status is updated to "Rejected" and saved in the Order entity.								

BUSINESS RULES:	• A reseller should only be allowed to reject an order placed by them, and which has been added to the orders backlog list by Natuurlik. • Only orders with a status of “Delayed” can be confirmed or rejected by a reseller.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	• The status of the reseller’s order is not “Delayed” and the reseller is not able to reject the order. • The system will automatically reject a reseller order that has been assigned to the orders back log five days after having logged the back order.
ASSUMPTIONS:	• The reseller has access to the internet and the system is not offline.
OPEN ISSUES:	• None

Table 58 - 5.24 Send Order Confirmation Reminder

ORDER SUBSYSTEM				
AUTHORS (s): Nomusa Vumisa		DATE: 27/08/2022		
VERSION: 1.0		LAST REVIEW DATE: 27/08/2022		
USE CASE NAME:	Send Order Confirmation Reminder	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒		
USE CASE ID:	5.24			
PRIORITY:	High			
SOURCE:	ByteXcom Requirements List			
PRIMARY BUSINESS ACTOR	• Time			
PRIMARY SYSTEM ACTOR	• None			
OTHER PARTICIPATING ACTORS:	• Reseller			
OTHER INTERESTED STAKEHOLDERS:	• None			
DESCRIPTION:	The use case begins when a reseller has a back order and it has reached the notification threshold. The system then sends an email reminder to the reseller on the confirmation due date.			
PRE-CONDITION:	The order status must be “Delayed”			
TRIGGER:	This use case is called when the order status is delayed and the confirmation threshold has been reached.			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE		
		MANUAL ACTION	AUTOMATED ACTION	
	Step 1: The confirmation threshold has been reached		Step 2: The system retrieves the following attributes	
			From the Order Table:	
			Details to be read	Attributes in the table
			Order Status	OrderStatus
			Back Order Date	BackOrderDate
			From the ConfirmationReminder Table:	
			Details to be read	Attributes in the table
			Value	Value

			and verifies if the order status is “Delayed” and the due date threshold has been reached																				
			Step 3: The system generates a confirmation reminder email with the following details: <table><tr><th>Details</th><th>Notes</th></tr><tr><td>To</td><td>The resellers email address – retrieved from the User Table using an SQL READ query</td></tr><tr><td>From</td><td>The company email address, stored as a “do not reply” address on the system</td></tr><tr><td>Subject</td><td>“Order Confirmation Reminder”</td></tr><tr><td>Body – Order Number</td><td>The order number retrieved from the Order table using an SQL READ query</td></tr><tr><td>Body – Order Date</td><td>The order date generated in the system using a datetime variable</td></tr><tr><td>Body – Confirmation Due Date</td><td>The confirmation due date retrieved from the ConfirmationReminder table using an SQL READ query</td></tr><tr><td>Body – First Name</td><td>The users name retrieved from the User table using an SQL READ query</td></tr><tr><td>Body – Main Text</td><td>“Hi (first name), A kind reminder that you have until tomorrow (confirmation due date) to confirm the invoice for order #(order number). To query your order, click the button below or contact us on 075 589 4569”</td></tr><tr><td>Button</td><td>The “Manage Order” button directs the user to their order on the system</td></tr></table>	Details	Notes	To	The resellers email address – retrieved from the User Table using an SQL READ query	From	The company email address, stored as a “do not reply” address on the system	Subject	“Order Confirmation Reminder”	Body – Order Number	The order number retrieved from the Order table using an SQL READ query	Body – Order Date	The order date generated in the system using a datetime variable	Body – Confirmation Due Date	The confirmation due date retrieved from the ConfirmationReminder table using an SQL READ query	Body – First Name	The users name retrieved from the User table using an SQL READ query	Body – Main Text	“Hi (first name), A kind reminder that you have until tomorrow (confirmation due date) to confirm the invoice for order #(order number). To query your order, click the button below or contact us on 075 589 4569”	Button	The “Manage Order” button directs the user to their order on the system
	Details	Notes																					
	To	The resellers email address – retrieved from the User Table using an SQL READ query																					
	From	The company email address, stored as a “do not reply” address on the system																					
	Subject	“Order Confirmation Reminder”																					
	Body – Order Number	The order number retrieved from the Order table using an SQL READ query																					
	Body – Order Date	The order date generated in the system using a datetime variable																					
	Body – Confirmation Due Date	The confirmation due date retrieved from the ConfirmationReminder table using an SQL READ query																					
	Body – First Name	The users name retrieved from the User table using an SQL READ query																					
Body – Main Text	“Hi (first name), A kind reminder that you have until tomorrow (confirmation due date) to confirm the invoice for order #(order number). To query your order, click the button below or contact us on 075 589 4569”																						
Button	The “Manage Order” button directs the user to their order on the system																						
		Step 4: The system sends the reminder to the reseller using the Sendgrid API.																					
ALTERNATIVE STEPS:																							
CONCLUSION: This use case concludes when the reminder has been sent to the reseller																							
POST-CONDITION: The reseller is reminded to confirm their order using attributes retrieved from the User and Order tables																							
BUSINESS RULES None																							
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS - Only the reseller should be notified. - Reminders must be sent via email.																							
ASSUMPTIONS: None																							
OPEN ISSUES: None																							

Table 59 - 5.25 Update Order Confirmation Time

ORDER SUBSYSTEM								
AUTHORS (s): Nomusa Vumisa			DATE: 28/08/2022					
VERSION: 1.1			LAST REVIEW DATE: 28/08/2022					
USE CASE NAME:	Update Order Confirmation Time		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒					
USE CASE ID:	5.25							
PRIORITY:	High							
SOURCE:	ByteXpress Requirements List							
PRIMARY BUSINESS ACTOR	• Administrator							
PRIMARY SYSTEM ACTOR	• None							
OTHER PARTICIPATING ACTORS:	• None							
OTHER INTERESTED STAKEHOLDERS:	• None							
DESCRIPTION:	This use case describes the process in which the administrator wishes to update the order confirmation time on the system. The administrator will then update the confirmation time and save it to the database. The use case ends when the administrator receives a successful update notification.							
PRE-CONDITION:	The administrator must be logged onto the system							
TRIGGER:	The use case is triggered when the administrator wants to update the order confirmation time							
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE						
		MANUAL ACTION	AUTOMATED ACTION					
	Step 1: The administrator clicks on the Order Confirmation Reminders link in the Orders dropdown on the admin side navigation bar		Step 2: The system retrieves the following attribute from the ConfirmationReminder table, using a method in a .NET controller and an SQL READ query:					
			<table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Days</td><td>Days</td></tr></table>		Details to be read	Attributes in the table	Days	Days
			Details to be read	Attributes in the table				
Days	Days							
and displays it on the Update Order Confirmation Time Screen which contains an MVC form with the following controls:								
		<table><tr><td>Control Name</td><td>Control Type</td><td>Notes</td></tr><tr><td>Update Order</td><td>Heading</td><td>Page main heading</td></tr></table>	Control Name	Control Type	Notes	Update Order	Heading	Page main heading
Control Name	Control Type	Notes						
Update Order	Heading	Page main heading						

			Confirmation Time		
			Current Time	Label	This control will be used as a label to display the current confirmation time
			Order Confirmation Time Intervals	Label	This control will be used as a label to indicate the confirmation time intervals
			Confirmation Time Intervals	Dropdown list	This control will be used by the administrator to select the confirmation time intervals
			Update	Button	This control will be used to confirm the updating of the confirmation time
			Note: All controls are enabled by default, unless stated otherwise.		
	Step 3: The system administrator selects their desired confirmation time interval and clicks on the Update button		Step 4: The system displays a modal dialogue with the following controls:		
			Control Name	Control Type	Notes
			"Are you sure you want to update the order confirmation time"	Label	This control is used to request confirmation on the updated order confirmation time
			Confirm	Button	This control is used to confirm the updating of the order confirmation time
			Cancel	Button	This control is used to close the modal dialogue
			Step 6: The system uses an SQL READ query to check if the confirmation time is not already active in the ConfirmationReminder table using the following attribute. [ALT]		
	Step 5: The system administrator clicks on the confirm button [ALT]				

			<table><tr><th>Details to be read</th><th colspan="2">Attributes in the table</th></tr><tr><td>Is Active</td><td colspan="2">IsActive</td></tr></table>	Details to be read	Attributes in the table		Is Active	IsActive	
	Details to be read	Attributes in the table							
	Is Active	IsActive							
		Step 7: The system activates the selected confirmation time in the ConfirmationReminder table using a method in a .NET controller and an SQL UPDATE query: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Is Active</td><td>Is Active</td><td>Updates the selected confirmation time record to "True" and the old confirmation time record to "False"</td></tr></table>	Details to be saved	Attributes in the table	Notes	Is Active	Is Active	Updates the selected confirmation time record to "True" and the old confirmation time record to "False"	
Details to be saved	Attributes in the table	Notes							
Is Active	Is Active	Updates the selected confirmation time record to "True" and the old confirmation time record to "False"							
		Step 8: The system displays a bootstrap alert notification with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"Order Confirmation Time updated successfully"</td><td>Label</td><td>This control is used to indicate a successful update</td></tr></table>	Control Name	Control Name	Notes	"Order Confirmation Time updated successfully"	Label	This control is used to indicate a successful update	
Control Name	Control Name	Notes							
"Order Confirmation Time updated successfully"	Label	This control is used to indicate a successful update							
ALTERNATE COURSES:	ALT Step 5: The administrator clicks on the cancel button Return to Step 2 or terminate the use case								
	ALT Step 6: The confirmation time is already active on the system - The system displays a bootstrap alert notification with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"The confirmation time is already active"</td><td>Label</td><td>This control is used to inform the administrator that the confirmation time is already active</td></tr></table> - The use case is terminated			Control Name	Control Name	Notes	"The confirmation time is already active"	Label	This control is used to inform the administrator that the confirmation time is already active
Control Name	Control Name	Notes							
"The confirmation time is already active"	Label	This control is used to inform the administrator that the confirmation time is already active							
CONCLUSION:	This use case concludes when the confirmation time is updated in the Confirmation Reminder table and the system displays an alert notification that informs the administrator that the confirmation time has been successfully updated.								
POST-CONDITION:	The confirmation time was updated in the ConfirmationReminder entity.								
BUSINESS RULES	- Only one confirmation time can be active at a time - Only the administrator users of the system can update the order confirmation time details on the system.								
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	- The confirmation time cannot be updated without an internet connection - Only the administrator can update the confirmation time								
ASSUMPTIONS:	None								
OPEN ISSUES:	None								

2.6. Query Reason Subsystem

Table 60 - 6.1 Create Query Reason

QUERY REASON SUBSYSTEM				
AUTHORS (s): Aphiwe Shoji		DATE: 14/05/2022		
VERSION: 1.0		LAST QUERY DATE: 14/05/2022		
USE CASE NAME:	Create Query Reason	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒		
USE CASE ID:	6.1			
PRIORITY:	High			
SOURCE:	ByteXcom Requirements List			
PRIMARY BUSINESS ACTOR	• Administrator			
PRIMARY SYSTEM ACTOR	• None			
OTHER PARTICIPATING ACTORS:	• None			
OTHER INTERESTED STAKEHOLDERS:	• None			
DESCRIPTION:	This use case describes the process in which the administrator wishes to add a new Query Reason onto the system. The use case begins when the administrator wishes to adds a Query Reason. The administrator will navigate the Query reason screen and proceed to capture the Query Reason details and save it to the system database.			
PRE-CONDITION:	• The administrator must be logged onto the system.			
TRIGGER:	The use case is triggered when the administrator wants to add a new Query Reason.			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE		
		MANUAL ACTION	AUTOMATED ACTION	
	Step 1: The administrator clicks on the Query Reason tab on the admin side navigation bar		Step 2: The system retrieves the following attributes from each entry in the Query Reason Table:	
		Details to be read	Attributes in the table	
		ID	QueryReasonID	
		Name	Name	

and displays them on the **Query Reason Screen** with the following controls:

Control Name	Control Type	Notes
Query Reason 's List	Heading	Page main heading
Create new Query Reason	Button	When clicked, this control will direct the administrator to the Create Query Reason screen
Show "" entries	Label and drop down	This control will be used to set the number of entries the administrator wants to display on a page
Search bar	Textbox	This control will be used to enter the search Query
Results table	Table	This control will be used to display all the Query Reason s that exist on the system
Name	Table Heading	This control will be used as a heading to display the Query Reason 's Name
Action	Table Heading	This control will be used as a heading to display the actions the administrator can perform with each entry – Edit and Delete
Showing "" to "" of "" entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database
"Previous" and "Next"	Label and button	This control is used to

					navigate through the entry pages and to indicate on which page the administrator is currently on																					
	Step 3: The administrator clicks on the Create New Query Reason button		Step 4: The system retrieves the displays the Create Query Reason Screen with the following controls:																							
			<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Create Query Reason</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Name</td><td>Label</td><td>This control will be used as a label to indicate the Query Reason 's name attribute</td></tr><tr><td>Name</td><td>Textbox</td><td>This control will be used by the administrator to enter the Query Reason 's name</td></tr><tr><td>Create</td><td>Button</td><td>This control will be used to confirm the creation of the Query Reason</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the creation of the Query Reason and return the administrator to the Query Reason Screen</td></tr><tr><td></td><td></td><td></td></tr></table>	Control Name	Control Type	Notes	Create Query Reason	Heading	Page main heading	Name	Label	This control will be used as a label to indicate the Query Reason 's name attribute	Name	Textbox	This control will be used by the administrator to enter the Query Reason 's name	Create	Button	This control will be used to confirm the creation of the Query Reason	Cancel	Button	This control will be used to cancel the creation of the Query Reason and return the administrator to the Query Reason Screen					
	Control Name	Control Type	Notes																							
	Create Query Reason	Heading	Page main heading																							
	Name	Label	This control will be used as a label to indicate the Query Reason 's name attribute																							
	Name	Textbox	This control will be used by the administrator to enter the Query Reason 's name																							
	Create	Button	This control will be used to confirm the creation of the Query Reason																							
Cancel	Button	This control will be used to cancel the creation of the Query Reason and return the administrator to the Query Reason Screen																								
Step 5: The system administrator enters the details and clicks on the create button [ALT]		Step 6: The system captures and validates the entered details against the following tables: [ALT] From the Query Reason Table																								
		<table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Name</td><td>Name</td><td>Required Alphabetical characters only Minimum: 1 Maximum: 50</td></tr></table>	Details to be read	Attributes in the table	Notes	Name	Name	Required Alphabetical characters only Minimum: 1 Maximum: 50																		
Details to be read	Attributes in the table	Notes																								
Name	Name	Required Alphabetical characters only Minimum: 1 Maximum: 50																								

			Step 7: The system saves the details in the Query Reason table using an SQL INSERT Query: <table><tr><th>Details to be created</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>ID</td><td>ID</td><td>Generated by the system</td></tr><tr><td>Name</td><td>Name</td><td>Required Alphabetical characters only Minimum: 1 Maximum: 50</td></tr></table>	Details to be created	Attributes in the table	Notes	ID	ID	Generated by the system	Name	Name	Required Alphabetical characters only Minimum: 1 Maximum: 50
	Details to be created	Attributes in the table	Notes									
	ID	ID	Generated by the system									
Name	Name	Required Alphabetical characters only Minimum: 1 Maximum: 50										
		Step 8: The system directs the administrator to the Query Reason Screen and displays a bootstrap alert notification with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"Query Reason created successfully"</td><td>Label</td><td>This control is used to indicate a successful creation</td></tr></table>	Control Name	Control Name	Notes	"Query Reason created successfully"	Label	This control is used to indicate a successful creation				
Control Name	Control Name	Notes										
"Query Reason created successfully"	Label	This control is used to indicate a successful creation										
ALTERNATE COURSES:	ALT Step 5: The administrator does not want to add a Query Reason anymore - The administrator clicks on the cancel button - The use case is terminated											
	ALT Step 6A: The details entered are not in the correct format or the required fields are empty The system displays a validation error with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Name</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>"The Name field is required" or "Invalid Name provided"</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Name textbox</td></tr></table>			Control Name	Control Name	Notes	Name	Textbox	This control will be outline in red to show a validation error	"The Name field is required" or "Invalid Name provided"	Label	This control will be used as a label to indicate a validation error in the Name textbox
	Control Name	Control Name	Notes									
	Name	Textbox	This control will be outline in red to show a validation error									
	"The Name field is required" or "Invalid Name provided"	Label	This control will be used as a label to indicate a validation error in the Name textbox									
Return to step 5												
ALT Step 6B: The Query Reason already exists on the system The system displays an error modal dialogue with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Query Reason already exists!</td><td>Label</td><td>This control is used to inform the administrator that the Query Reason already exists</td></tr><tr><td>Okay</td><td>Button</td><td>This control is used to cancel the creation of the Query Reason</td></tr></table>			Control Name	Control Name	Notes	Query Reason already exists!	Label	This control is used to inform the administrator that the Query Reason already exists	Okay	Button	This control is used to cancel the creation of the Query Reason	
Control Name	Control Name	Notes										
Query Reason already exists!	Label	This control is used to inform the administrator that the Query Reason already exists										
Okay	Button	This control is used to cancel the creation of the Query Reason										
- The administrator clicks the "Okay" button - The use case is terminated												
CONCLUSION:	This use case concludes when a new Query Reason is created and the system displays a bootstrap notification to confirm the creation.											

POST-CONDITION:	The Query Reason details are added to the QueryReason entity in the database.
BUSINESS RULES	Only the administrator can add a new Query Reason.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	The system is offline or the administrator does not have an internet connection, resulting in the administrator not being able to add a new query reason.
ASSUMPTIONS:	None
OPEN ISSUES:	None

Table 61 - 6.2 Search Query Reason

QUERY REASON SUBSYSTEM										
AUTHORS (s): Aphiwe Shoji		DATE: 14/05/2022								
VERSION: 1.0		LAST QUERY DATE: 14/05/2022								
USE CASE NAME:	Search Query Reason	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒								
USE CASE ID:	6.2									
PRIORITY:	Medium									
SOURCE:	ByteXcom Requirements List									
PRIMARY BUSINESS ACTOR	• Administrator									
PRIMARY SYSTEM ACTOR	• None									
OTHER PARTICIPATING ACTORS:	• None									
OTHER INTERESTED STAKEHOLDERS:	• None									
DESCRIPTION:	This use case describes the process in the administrator wishes to search through the existing Query Reasons on the system in the Query Reason Entity Table. The system will prompt the administrator to use the Query Reason details to perform a search Query and the system will retrieve the Query Reason record that matches the inputs of the search Query									
PRE-CONDITION:	The administrator must be logged onto the system.									
TRIGGER:	This use case is called when the administrator wants to search for a Query Reason on the system.									
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE								
		MANUAL ACTION	AUTOMATED ACTION							
	Step 1: The administrator clicks on the Query Reason tab on the admin side navigation bar		Step 2: The system retrieves the following attributes from each entry in the Query Reason Table:							
			<table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>ID</td><td>QueryReasonID</td></tr><tr><td>Name</td><td>Name</td></tr></table>		Details to be read	Attributes in the table	ID	QueryReasonID	Name	Name
			Details to be read	Attributes in the table						
			ID	QueryReasonID						
Name	Name									
and displays them on the Query Reason Screen with the following controls:										

			<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Query Reason 's List</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Create new Query Reason</td><td>Button</td><td>When clicked, this control will direct the administrator to the Create Query Reason screen</td></tr><tr><td>Show "" entries</td><td>Label and drop down</td><td>This control will be used to set the number of entries the administrator wants to display on a page</td></tr><tr><td>Search bar</td><td>Textbox</td><td>This control will be used to enter the search Query</td></tr><tr><td>Results table</td><td>Table</td><td>This control will be used to display all the Query Reason s that exist on the system</td></tr><tr><td>Name</td><td>Table Heading</td><td>This control will be used as a heading to display the Query Reason 's Name</td></tr><tr><td>Showing "" to "" of "" entries</td><td>Label</td><td>This control will be used to display the number of entries displayed per page as well as the total number of entries in the database</td></tr><tr><td>"Previous" and "Next"</td><td>Label and button</td><td>This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on</td></tr></table>	Control Name	Control Type	Notes	Query Reason 's List	Heading	Page main heading	Create new Query Reason	Button	When clicked, this control will direct the administrator to the Create Query Reason screen	Show "" entries	Label and drop down	This control will be used to set the number of entries the administrator wants to display on a page	Search bar	Textbox	This control will be used to enter the search Query	Results table	Table	This control will be used to display all the Query Reason s that exist on the system	Name	Table Heading	This control will be used as a heading to display the Query Reason 's Name	Showing "" to "" of "" entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database	"Previous" and "Next"	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on
	Control Name	Control Type	Notes																											
	Query Reason 's List	Heading	Page main heading																											
	Create new Query Reason	Button	When clicked, this control will direct the administrator to the Create Query Reason screen																											
	Show "" entries	Label and drop down	This control will be used to set the number of entries the administrator wants to display on a page																											
	Search bar	Textbox	This control will be used to enter the search Query																											
	Results table	Table	This control will be used to display all the Query Reason s that exist on the system																											
	Name	Table Heading	This control will be used as a heading to display the Query Reason 's Name																											
	Showing "" to "" of "" entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database																											
"Previous" and "Next"	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on																												
Step 4: The administrator enters their search Query		Step 5: The system searches the table rows in the data table and displays the Query Reason (s) matching the search Query [ALT]																												

	into the search textbox		
	ALT Step 5: The search Query entered does not match any records existing in the Query Reason entity. The system displays a “No Records Found” message in the results table.		
CONCLUSION:	This use case concludes when the desired Query Reason’s details are displayed on the system.		
POST-CONDITION:	The QueryReason entity was searched for and an entry that corresponds to the search Query or parameter entered.		
BUSINESS RULES	Only the administrators of the system should be allowed to search for a query reason existing on the system.		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	The system is offline or the administrator does not have an internet connection, resulting in the administrator not being able to search for query reasons defined on the system.		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

Table 62 - 6.3 Update Query Reason

QUERY REASON SUBSYSTEM				
AUTHORS (s): Aphiwe Shoji		DATE: 14/05/2022		
VERSION: 1.0		LAST QUERY DATE: 14/05/2022		
USE CASE NAME:	Update Query Reason		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒	
USE CASE ID:	6.3			
PRIORITY:	Medium			
SOURCE:	ByteXcom Requirements List			
PRIMARY BUSINESS ACTOR	• Administrator			
PRIMARY SYSTEM ACTOR	• None			
OTHER PARTICIPATING ACTORS:	• None			
OTHER INTERESTED STAKEHOLDERS:	• None			
DESCRIPTION:	This use case describes the process in which the administrator wishes to update the details of Query Reason record in the Query Reason Entity table. The use case begins when the administrator updates an existing Query Reason in the system. The administrator will update all of the pertinent information regarding the Query Reason and the system will update the record in the Query Reason entity table			
PRE-CONDITION:	The administrator must be logged in to the system.			
TRIGGER:	The use is triggered when the administrator wishes to update a Query Reason			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE		
		MANUAL ACTION	AUTOMATED ACTION	
	Step 1: The administrator wishes to update a Query Reason in the system.		Step 2: The system invokes use case 6.12 Search Query Reason	
	Step 3: The administrator clicks on the update button		Step 3: The system retrieves the following attributes from the following tables From the Query Reason Table: <table><tr><td>Details to be read</td><td>Attributes in the table</td></tr></table>	Details to be read
Details to be read	Attributes in the table			

		<table><tr><td>ID</td><td>QueryReasonID</td></tr><tr><td>Name</td><td>Name</td></tr></table>	ID	QueryReasonID	Name	Name											
	ID	QueryReasonID															
	Name	Name															
		and displays them on the Update Query Reason Screen with the following controls:															
		<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Create Query Reason</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Name</td><td>Label</td><td>This control will be used as a label to indicate the Query Reason's name attribute</td></tr><tr><td>Name</td><td>Textbox</td><td>This control will be used by the administrator to enter the Query Reason's name</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the updating of the Query Reason and return the administrator to the Query Reason Screen</td></tr></table>	Control Name	Control Type	Notes	Create Query Reason	Heading	Page main heading	Name	Label	This control will be used as a label to indicate the Query Reason's name attribute	Name	Textbox	This control will be used by the administrator to enter the Query Reason's name	Cancel	Button	This control will be used to cancel the updating of the Query Reason and return the administrator to the Query Reason Screen
	Control Name	Control Type	Notes														
	Create Query Reason	Heading	Page main heading														
	Name	Label	This control will be used as a label to indicate the Query Reason's name attribute														
	Name	Textbox	This control will be used by the administrator to enter the Query Reason's name														
	Cancel	Button	This control will be used to cancel the updating of the Query Reason and return the administrator to the Query Reason Screen														
	Step 4: The system administrator enters the updated Query Reason details and clicks on the update button [ALT]																
	Step 5: The system captures and validates the entered details against the following tables: [ALT] From the Query Reason Table <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Name</td><td>Name</td><td>Required Alphabetical characters only Minimum: 1 Maximum: 50</td></tr></table>	Details to be saved	Attributes in the table	Notes	Name	Name	Required Alphabetical characters only Minimum: 1 Maximum: 50										
Details to be saved	Attributes in the table	Notes															
Name	Name	Required Alphabetical characters only Minimum: 1 Maximum: 50															
	Step 6: The system displays a modal dialogue with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr></table>	Control Name	Control Type	Notes													
Control Name	Control Type	Notes															

			“Confirm Query Reason Details”	Label	This control is used to request confirmation on the updated Query Reason details						
			Confirm	Button	This control is used to confirm the updating of the Query Reason details						
			Cancel	Button	This control is used to close the modal dialogue and return the administrator to the Update Query Reason Screen						
	Step 7: The system administrator clicks on the confirm button [ALT]		Step 8: The system saves the updated details into the Query Reason table using an SQL UPDATE Query:								
			<table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>ID</td><td>ID</td><td>Generated by the system</td></tr><tr><td>Name</td><td>Name</td><td>Required Alphabetical characters only Minimum: 1 Maximum: 50</td></tr></table>	Details to be saved	Attributes in the table	Notes	ID	ID	Generated by the system	Name	Name
Details to be saved	Attributes in the table	Notes									
ID	ID	Generated by the system									
Name	Name	Required Alphabetical characters only Minimum: 1 Maximum: 50									
			Step 9: The system directs the administrator to the Query Reason Screen and displays a bootstrap alert notification with the following controls:								
			<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>“Query Reason updated successfully”</td><td>Label</td><td>This control is used to indicate a successful creation</td></tr></table>	Control Name	Control Name	Notes	“Query Reason updated successfully”	Label	This control is used to indicate a successful creation		
Control Name	Control Name	Notes									
“Query Reason updated successfully”	Label	This control is used to indicate a successful creation									
ALTERNATE COURSES:	ALT Step 5: The administrator does not want to update the Query Reason anymore - The administrator clicks on the cancel button - The use case is terminated										
	ALT Step 5A: The details entered are not in the correct format or the required fields are empty The system displays a validation error modal dialogue with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr></table>					Control Name	Control Name	Notes			
Control Name	Control Name	Notes									

	Name	Textbox	This control will be outline in red to show a validation error
	"The Name field is required" or "Invalid Name provided"	Label	This control will be used as a label to indicate a validation error in the Name textbox
	Return to step 3		
	ALT Step 5B: The Query Reason already exists on the system		
	The system displays an error modal dialogue with the following controls		
	Control Name	Control Name	Notes
	Query Reason already exists!	Label	This control is used to inform the administrator that the Query Reason already exists
	Okay	Button	This control is used to cancel the updating of the Query Reason
	- The administrator clicks the "Okay" button - The use case is terminated		
	ALT Step 7: The administrator clicks on the cancel button		
	Return to Step 4 or terminate the use case		
CONCLUSION:	This use case concludes when the details of the respective Query Reason have been updated successfully on the system		
POST-CONDITION:	The Query Reason details are updated on the database.		
BUSINESS RULES	Only the administrator should be permitted to update a Query Reason on the system.		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	The system is offline or the administrator does not have an internet connection, resulting in the administrator not being able to update an existing query reason.		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

Table 63 - 6.4 Delete Query Reason

QUERY REASON SUBSYSTEM			
AUTHORS (s): Aphiwe Shoji		DATE: 14/05/2022	
VERSION: 1.0		LAST QUERY DATE: 14/05/2022	
USE CASE NAME:	Delete Query Reason		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒
USE CASE ID:	6.4		
PRIORITY:	Medium		
SOURCE:	ByteXcom Requirements List		
PRIMARY BUSINESS ACTOR	• Administrator		
PRIMARY SYSTEM ACTOR	• None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	This use case narrative describes the process in which the administrator wishes to delete a Query Reason from the system. The administrator will search for the required Query Reason selected for deletion. The administrator will then confirm the deletion of the Query Reason and the use case concludes when the system notifies the administrator of a successful deletion.		
PRE-CONDITION:	The administrator must be logged in to the system.		
TRIGGER:	This use case is called when the administrator wishes to delete a Query Reason .		
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE	
		MANUAL ACTION	AUTOMATED ACTION
	Step 1: The administrator wishes to delete a Query Reason from the system.		Step 2: The system invokes use case 6.12 Search Query Reason
	Step 3: The administrator clicks on the Delete button		Step 4: The system retrieves the following attributes in the following tables From the Query Reason Table:
		Details to be read	Attributes in the table
	ID	QueryReasonID	

		<table><tr><td>Name</td><td>Name</td></tr></table> and displays them on the Delete Query Reason Details Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Create Query Reason</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Name</td><td>Label</td><td>This control will be used to display the Query Reason's name</td></tr><tr><td>Delete</td><td>Button</td><td>This control will be used to confirm the deletion of the Query Reason</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the deletion of the Query Reason and return the administrator to the Query Reason Screen</td></tr></table>	Name	Name	Control Name	Control Type	Notes	Create Query Reason	Heading	Page main heading	Name	Label	This control will be used to display the Query Reason's name	Delete	Button	This control will be used to confirm the deletion of the Query Reason	Cancel	Button	This control will be used to cancel the deletion of the Query Reason and return the administrator to the Query Reason Screen
	Name	Name																	
	Control Name	Control Type	Notes																
	Create Query Reason	Heading	Page main heading																
	Name	Label	This control will be used to display the Query Reason's name																
	Delete	Button	This control will be used to confirm the deletion of the Query Reason																
	Cancel	Button	This control will be used to cancel the deletion of the Query Reason and return the administrator to the Query Reason Screen																
		Step 5: The system displays a modal dialogue with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>"Confirm Query Reason Deletion"</td><td>Label</td><td>This control is used to request confirmation on the deletion of the Query Reason details</td></tr><tr><td>Confirm</td><td>Button</td><td>This control is used to confirm the deletion of the Query Reason details</td></tr><tr><td>Cancel</td><td>Button</td><td>This control is used to close the modal dialogue</td></tr></table>	Control Name	Control Type	Notes	"Confirm Query Reason Deletion"	Label	This control is used to request confirmation on the deletion of the Query Reason details	Confirm	Button	This control is used to confirm the deletion of the Query Reason details	Cancel	Button	This control is used to close the modal dialogue					
	Control Name	Control Type	Notes																
	"Confirm Query Reason Deletion"	Label	This control is used to request confirmation on the deletion of the Query Reason details																
Confirm	Button	This control is used to confirm the deletion of the Query Reason details																	
Cancel	Button	This control is used to close the modal dialogue																	
Step 5: The system administrator clicks on the confirm button [ALT]	Step 6: The system deletes the selected Query Reason from the QueryReason entity in the database using an SQL DELETE Query. [ALT] <table><tr><th>Details to be deleted</th><th>Attributes in the table</th><th>Notes</th></tr></table>	Details to be deleted	Attributes in the table	Notes															
Details to be deleted	Attributes in the table	Notes																	

			ID	ID	Generated by the system	
			Name	Name	Required Alphabetical characters only Minimum: 1 Maximum: 50	
				Step 7: The system redirects the administrator to the Query Reason screen and displays a modal dialogue with the following controls:		
				Control Name	Control Name	Notes
ALTERNATE COURSES:			"Query Reason Deleted Successfully"	Label	This control is used to indicate a successful deletion	
	ALT Step 5: The administrator does not want to delete the Query Reason anymore - The administrator clicks on the cancel button - The use case is terminated ALT Step 6: The Query Reason cannot be deleted as it has an association with the Order Query Table - The system redirects the administrator and displays a bootstrap alert notification with the following controls:					
	Control Name	Control Name	Notes			
	"Supplier cannot be deleted"	Label	This control is used to indicate an unsuccessful deletion			
The use case is terminated						
CONCLUSION:	This use case concludes when a Query Reason record has been removed from the system and a bootstrap notification is displayed to confirm that the selected query reason has been removed successfully.					
POST-CONDITION:	The Query Reason record is removed from the QueryReason entity table.					
BUSINESS RULES	Only the administrators should be allowed to delete a Query Reason from the system.					
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	The system is offline or the administrator does not have access to the internet, resulting in the system user not being able to remove a query reason from the system.					
ASSUMPTIONS:	None					
OPEN ISSUES:	None					

Table 64 - 6.5 Review Order Query

QUERY REASON SUBSYSTEM									
AUTHORS (s): Ofhani Mungani		DATE: 07/28/2022							
VERSION: 1.0		LAST REVIEW DATE: 20/07/2022							
USE CASE NAME:	Review Order Query		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒						
USE CASE ID:	6.5								
PRIORITY:	High								
SOURCE:	ByteXcom Requirements List								
PRIMARY BUSINESS ACTOR	• Administrator/Sales Assistant								
PRIMARY SYSTEM ACTOR	• None								
OTHER PARTICIPATING ACTORS:	• None								
OTHER INTERESTED STAKEHOLDERS:	• None								
DESCRIPTION:	This use case describes an event where the administrator or sales assistant wants to view and provides feedback to an order query on the system. The use case begins when the administrator/ sales assistant requests to view an order query and the system responds by retrieving the order query details. The use case concludes when the details are displayed on the screen.								
PRE-CONDITION:	• The administrator or sales assistant must be logged on the system								
TRIGGER:	The use case is called when the administrator or sales assistant wants to view an order query								
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE							
		MANUAL ACTION	AUTOMATED ACTION						
	Step 1: The administrator or sales assistant clicks on the “View Order Queries” on the side navi link		Step 2: The system retrieves the values with following attributes From the QueryReason entity: <table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Query Reason</td><td>Name</td></tr></table> From the OrderQuery entity: <table><tr><td>Details to be read</td><td>Attributes in the table</td></tr></table>	Details to be read	Attributes in the table	Query Reason	Name	Details to be read	Attributes in the table
			Details to be read	Attributes in the table					
			Query Reason	Name					
Details to be read	Attributes in the table								

Date	Date
Description	Description

From the **QueryStatus** entity:

Details to be read	Attributes in the table	Notes
Status	Status	Initialized to pending review

From the **Order** entity:

Details to be read	Attributes in the table	Notes
Order Number	OrderId	

The system then displays the **Order Queries List Screen** with the following controls:

Control Name	Control Type	Notes
Queries List	Heading	Page main heading
Search bar	Textbox	This control will be used when the user inserts the search parameters
Results table	Table	This control will be used to display all the order queries that exist on the system
Show “” entries	Label and drop down	This control will be used to set the number of entries the administrator wants to display on a page
Order Number	Table Heading	This control will be used as a heading to display the OrderId for all Orders
Query Reason	Table Heading	This control will be used as a heading to display the query reason for all order queries

			Description	Table Heading	This control will be used as a heading to display the description for all order queries
			Status	Table Heading	This control will be used as a heading to display the Status for all order queries
			Date	Table Heading	This control will be used as a heading to display the date for all order queries
			Review	button	This control will be used to navigate to the Order Query Reply screen
			View Order	button	This control will be used to navigate to the Order Details Screen
			Action	Heading	This control will be used to house the table's buttons
			Showing “” to “” of “” entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database
			“Previous” and “Next”	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on
Step 3: The administrator or			Step 4:		

sales assistant clicks on the **Review** button.

The system retrieves the values with following attributes

From the **QueryReason** entity:

Details to be read	Attributes in the table
Query Reason	Name

From the **OrderQuery** entity:

Details to be read	Attributes in the table
Description	Description
Feedback	Feedback

From the **Order** entity:

Details to be read	Attributes in the table
Order Number	OrderId

The system then displays the **Order Query Reply Screen** with the following controls:

Control Name	Control Type	Notes
Order Query Reply	Heading	Page main heading
Order Number	Label	This control will be used as a label to indicate the OrderId attribute of an Order
Order Number	Textbox	This control will be used by the administrator to view a OrderId for an order
Query Reason	Label	This control will be used as a label to indicate the Name attribute of the QueryReason
Query Reason	Textbox	This control will be used by the administrator to view a query reason for an order query
Description	Label	This control will be used

					by the administrator to indicate a query reason for the order query								
	Description	Textbox			This control will be used by the administrator to view a description for an order query								
	Provide Feedback	Label			This control will be used by the administrator to indicate a query reason for the order								
	Feedback	Textbox			This control will be used by the administrator to write feedback for an order query								
	Submit	Button			This control will be used to confirm the submission for order query feedback								
	Cancel	Button			This control will be used to cancel the feedback for an order query								
	Step 5: The administrator or sales assistant enters the details and clicks on the Submit button [ALT]		Step 6: The system captures and validates the entered details against the OrderQuery table [ALT] <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Feedback</td><td>Feedback</td><td>Required</td></tr></table>			Details to be read	Attributes in the table	Notes	Feedback	Feedback	Required		
	Details to be read	Attributes in the table	Notes										
Feedback	Feedback	Required											
		Step 7: The system Update and saves the information using an INSERT SQL query: Into the OrderQuery table <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Feedback</td><td>Feedback</td><td>Required</td></tr><tr><td>Status</td><td>Status</td><td>Status is changed from pending</td></tr></table>			Details to be saved	Attributes in the table	Notes	Feedback	Feedback	Required	Status	Status	Status is changed from pending
Details to be saved	Attributes in the table	Notes											
Feedback	Feedback	Required											
Status	Status	Status is changed from pending											

					review to reviewed									
			Step 8: The system directs the administrator or sales assistant to the Order Queries List Screen and displays a bootstrap alert notification with the following controls: Bootstrap notification: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>“Order Query Feedback Successfully Submitted”</td><td>Label</td><td>This control is used to indicate a successful creation</td></tr></table>			Control Name	Control Name	Notes	“Order Query Feedback Successfully Submitted”	Label	This control is used to indicate a successful creation			
	Control Name	Control Name	Notes											
“Order Query Feedback Successfully Submitted”	Label	This control is used to indicate a successful creation												
ALTERNATE COURSES:	ALT Step 5: The administrator/sales assistant doesn’t want to provide feedback to an order query • The administrator or sales assistant clicks on the Cancel button • Terminate use case													
	ALT Step 6: The details entered are not in the correct format or the required fields are empty • The system displays a validation error message box with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Feedback</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>Feedback is required</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Feedback textbox</td></tr></table>					Control Name	Control Name	Notes	Feedback	Textbox	This control will be outline in red to show a validation error	Feedback is required	Label	This control will be used as a label to indicate a validation error in the Feedback textbox
	Control Name	Control Name	Notes											
Feedback	Textbox	This control will be outline in red to show a validation error												
Feedback is required	Label	This control will be used as a label to indicate a validation error in the Feedback textbox												
• Return to Step 5														
CONCLUSION:	The use case concludes when the feedback for an order query is added to the system and a bootstrap notification is displayed which confirms order query feedback has been captured successfully.													
POST-CONDITION:	The customer or reseller receives feedback on the system about their order query													
BUSINESS RULES	• Only the administrator or sales assistant can provide feedback for an order query.													
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• The administrator or sales assistant cannot view an order query without an internet connection.													
ASSUMPTIONS:	• None													
OPEN ISSUES:	• None													

Table 65 - 6.6 View Order Query Feedback

QUERY REASON SUBSYSTEM			
AUTHORS (s): Thenjiwe Ntsonda		DATE: 23/06/2022	
VERSION: 1.0		LAST REVIEW DATE: 23/06/2022	
USE CASE NAME:	View Order Query Feedback	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒	
USE CASE ID:	6.6		
PRIORITY:	High		
SOURCE:	ByteXcom Requirements List		
PRIMARY BUSINESS ACTOR	• Customer or Reseller		
PRIMARY SYSTEM ACTOR	• None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	This use case begins when a customer or reseller wishes to view the feedback of a query they submitted on the system. The system retrieves the selected order query details from the relevant tables and displays the details to the customer or reseller.		
PRE-CONDITION:	• Customer or reseller must have submitted an order query prior • Customer or reseller must be logged in		
TRIGGER:	This use case is called when a customer or reseller wishes to view the feedback on a query they submitted on the system.		
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE	
		MANUAL ACTION	AUTOMATED ACTION
	Step 1: The customer or reseller wishes to view the feedback on a query they submitted on the system		Step 2: The system invokes use case 1.3 View Profile Details

Step 3: The customer or reseller clicks on the **Queries** nav link on the side bar.

Step 4: The system retrieves the following using a method in a .NET Controller and an SQL READ query

From the **OrderQuery** table

Details to be read	Attributes in the table
Description	Description
Query Feedback	QueryFeedback

From the **Order** table:

Details to be read	Attributes in the table
Order Number	OrderId

From the **QueryReason** table:

Details to be read	Attributes in the table
Query Reason	QueryReasonName

And displays them on the **My Order Queries Screen** with the following controls:

Control Name	Control Type	Notes
My Order Queries	Heading	Page main heading
Show "" entries	Label and drop down	This control will be used to set the number of entries the administrator wants to display on a page
Search bar	Textbox	This control will be used to enter the search query
Results table	Table	This control will be used to display all the order queries that exists on the system

			Order Number	Table Heading	This control will be used as a heading to display the Order Number
			Query Reason	Table Heading	This control will be used as a heading to display the name of the query reason
			Query Description	Table Heading	This control will be used as a heading to display the description of the query
			Status	Table Heading	This control will be used as a heading to display the status of the query
			Action	Label	This control will be used as a heading to display the View action method
			Showing “” to “” of “” entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database
			“Previous” and “Next”	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on

	Step 5: The customer or reseller clicks on the View button.		Step 6: The system retrieves the following attributes from the following tables, using a method in a .NET controller and an SQL READ query: From the OrderQuery table <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Description</td><td>Description</td></tr><tr><td>Query Feedback</td><td>QueryFeedback</td></tr></table> From the Order table: <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Order Number</td><td>OrderId</td></tr></table> From the QueryReason table: <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Query Reason</td><td>QueryReasonName</td></tr></table> And displays the Order Query Overview Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Order Query Overview</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Order Number</td><td>Label</td><td>This control will be used as a heading to display the Order Number</td></tr><tr><td>Order Number</td><td>Textbox</td><td>This control will be used to display the order number</td></tr><tr><td>Description</td><td>Label</td><td>This control will be used as a heading</td></tr></table>	Details to be read	Attributes in the table	Description	Description	Query Feedback	QueryFeedback	Details to be read	Attributes in the table	Order Number	OrderId	Details to be read	Attributes in the table	Query Reason	QueryReasonName	Control Name	Control Type	Notes	Order Query Overview	Heading	Page main heading	Order Number	Label	This control will be used as a heading to display the Order Number	Order Number	Textbox	This control will be used to display the order number	Description	Label	This control will be used as a heading
	Details to be read	Attributes in the table																														
	Description	Description																														
	Query Feedback	QueryFeedback																														
	Details to be read	Attributes in the table																														
	Order Number	OrderId																														
	Details to be read	Attributes in the table																														
	Query Reason	QueryReasonName																														
	Control Name	Control Type	Notes																													
Order Query Overview	Heading	Page main heading																														
Order Number	Label	This control will be used as a heading to display the Order Number																														
Order Number	Textbox	This control will be used to display the order number																														
Description	Label	This control will be used as a heading																														

					to display the description of the query
			Description	Textbox	This control will be used to display the description of the query
			Feedback from Natuurlik	Label	This control will be used as a heading to display the feedback received from Natuurlik
			Feedback from Natuurlik	Textbox	This control will be used to display the feedback received from Natuurlik
			Back	Button	This control will be used to redirect the user back to the My Order Queries Screen
			Note: All controls are disabled by default, unless otherwise stated. All text input fields are empty by default, unless stated otherwise.		
ALTERNATE COURSES:	None				
CONCLUSION:	The system displays the selected order query details to the customer or reseller				
POST-CONDITION:	The OrderQuery, Order and QueryReason tables are searched, and the corresponding values are retrieved.				
BUSINESS RULES	Only the logged in customer or reseller should be allowed to view their submitted order queries.				
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None				
ASSUMPTIONS:	The customer or reseller has a stable internet connection.				
OPEN ISSUES:	None				

2.7. Review Subsystem

Table 66 - 7.1 Create Review Reason

REVIEW SUBSYSTEM										
AUTHORS (s): Aphiwe Shozi		DATE: 14/05/2022								
VERSION: 1.0		LAST REVIEW DATE: 14/05/2022								
USE CASE NAME:	Create Review Reason	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒								
USE CASE ID:	7.1									
PRIORITY:	High									
SOURCE:	ByteXcom Requirements List									
PRIMARY BUSINESS ACTOR	• Administrator									
PRIMARY SYSTEM ACTOR	• None									
OTHER PARTICIPATING ACTORS:	• None									
OTHER INTERESTED STAKEHOLDERS:	• None									
DESCRIPTION:	This use case describes the process in which the administrator wishes to add a new Review Reason onto the system. The use case begins when the administrator wishes to adds a Review Reason. The administrator will navigate the Review reason screen and proceed to capture the Review Reason details and save it to the system database.									
PRE-CONDITION:	• The administrator must be logged onto the system.									
TRIGGER:	The use case is triggered when the administrator wants to add a new Review Reason.									
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE								
		MANUAL ACTION	AUTOMATED ACTION							
	Step 1: The administrator clicks on the Review Reason tab on the admin side navigation bar		Step 2: The system retrieves the following attributes from each entry in the Review Reason Table:							
			<table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>ID</td><td>ReviewReasonID</td></tr><tr><td>Name</td><td>Name</td></tr></table>		Details to be read	Attributes in the table	ID	ReviewReasonID	Name	Name
			Details to be read	Attributes in the table						
			ID	ReviewReasonID						
Name	Name									

and displays them on the **Review Reason Screen** with the following controls:

Control Name	Control Type	Notes
Review Reason 's List	Heading	Page main heading
Create new Review Reason	Button	When clicked, this control will direct the administrator to the Create Review Reason screen
Show "" entries	Label and drop down	This control will be used to set the number of entries the administrator wants to display on a page
Search bar	Textbox	This control will be used to enter the search Review
Results table	Table	This control will be used to display all the Review Reason s that exist on the system
Name	Table Heading	This control will be used as a heading to display the Review Reason 's Name
Action	Table Heading	This control will be used as a heading to display the actions the administrator can perform with each entry – Edit and Delete
Showing "" to "" of "" entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of

					entries in the database
			"Previous" and "Next"	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on
	Step 3: The administrator clicks on the Create New Review Reason button		Step 4: The system retrieves the displays the Create Review Reason Screen with the following controls:		
			Control Name	Control Type	Notes
			Create Review Reason	Heading	Page main heading
Name			Label	This control will be used as a label to indicate the Review Reason 's name attribute	
Name	Textbox	This control will be used by the administrator to enter the Review Reason 's name			
Create	Button	This control will be used to confirm the creation of the Review Reason			
Cancel	Button	This control will be used to cancel the creation of the Review Reason and return the administrator to the Review Reason Screen			
Step 5: The system administrator enters the details and clicks on the create button [ALT]		Step 6: The system captures and validates the entered details against the following tables: [ALT]			
		From the Review Reason Table			

			<table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Name</td><td>Name</td><td>Required Alphabetical characters only Minimum: 1 Maximum: 50</td></tr></table>	Details to be read	Attributes in the table	Notes	Name	Name	Required Alphabetical characters only Minimum: 1 Maximum: 50			
	Details to be read	Attributes in the table	Notes									
	Name	Name	Required Alphabetical characters only Minimum: 1 Maximum: 50									
		Step 7: The system saves the details in the ReviewReason entity using an SQL INSERT Review: <table><tr><th>Details to be created</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Name</td><td>Name</td><td>Required Alphabetical characters only Minimum: 1 Maximum: 50</td></tr></table>	Details to be created	Attributes in the table	Notes	Name	Name	Required Alphabetical characters only Minimum: 1 Maximum: 50				
Details to be created	Attributes in the table	Notes										
Name	Name	Required Alphabetical characters only Minimum: 1 Maximum: 50										
		Step 8: The system directs the administrator to the Review Reason Screen and displays a bootstrap alert notification with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"Review Reason created successfully"</td><td>Label</td><td>This control is used to indicate a successful creation</td></tr></table>	Control Name	Control Name	Notes	"Review Reason created successfully"	Label	This control is used to indicate a successful creation				
Control Name	Control Name	Notes										
"Review Reason created successfully"	Label	This control is used to indicate a successful creation										
ALTERNATE COURSES:	ALT Step 5: The administrator does not want to add a Review Reason anymore - The administrator clicks on the cancel button - The use case is terminated											
	ALT Step 6A: The details entered are not in the correct format or the required fields are empty The system displays a validation error with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Name</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>"The Name field is required" or "Invalid Name provided"</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Name textbox</td></tr></table>			Control Name	Control Name	Notes	Name	Textbox	This control will be outline in red to show a validation error	"The Name field is required" or "Invalid Name provided"	Label	This control will be used as a label to indicate a validation error in the Name textbox
	Control Name	Control Name	Notes									
	Name	Textbox	This control will be outline in red to show a validation error									
"The Name field is required" or "Invalid Name provided"	Label	This control will be used as a label to indicate a validation error in the Name textbox										
Return to step 5												
	ALT Step 6B: The Review Reason already exists on the system The system displays an error modal dialogue with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Review Reason already exists!</td><td>Label</td><td>This control is used to inform the administrator that the Review Reason already exists</td></tr><tr><td>Okay</td><td>Button</td><td>This control is used to cancel the creation of the Review Reason</td></tr></table>			Control Name	Control Name	Notes	Review Reason already exists!	Label	This control is used to inform the administrator that the Review Reason already exists	Okay	Button	This control is used to cancel the creation of the Review Reason
Control Name	Control Name	Notes										
Review Reason already exists!	Label	This control is used to inform the administrator that the Review Reason already exists										
Okay	Button	This control is used to cancel the creation of the Review Reason										

	• The administrator clicks the “Okay” button • The use case is terminated
CONCLUSION:	This use case concludes when a new Review Reason is created and an alert is displayed which confirms that the new review reason has been added to the system.
POST-CONDITION:	The order review reason details are added to the ReviewReason entity in the database.
BUSINESS RULES	• Only the administrator can add a new Review Reason to the system.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• The system is offline or the administrator does not have access to the internet and the administrator is not able to add a new review reason to the system.
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 67 - 7.2 Search Review Reason

REVIEW SUBSYSTEM										
AUTHORS (s): Aphiwe Shoji		DATE: 14/05/2022								
VERSION: 1.0		LAST REVIEW DATE: 14/05/2022								
USE CASE NAME:	Search Review Reason	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒								
USE CASE ID:	7.2									
PRIORITY:	Medium									
SOURCE:	ByteXcom Requirements List									
PRIMARY BUSINESS ACTOR	• Administrator									
PRIMARY SYSTEM ACTOR	• None									
OTHER PARTICIPATING ACTORS:	• None									
OTHER INTERESTED STAKEHOLDERS:	• None									
DESCRIPTION:	This use case describes the process in the administrator wishes to search through the existing Review Reasons on the system in the Review Reason Entity Table. The system will prompt the administrator to use the Review Reason details to perform a search Review and the system will retrieve the Review Reason record that matches the inputs of the search Review									
PRE-CONDITION:	The administrator must be logged onto the system.									
TRIGGER:	This use case is called when the administrator wants to search for a Review Reason on the system.									
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE								
		MANUAL ACTION	AUTOMATED ACTION							
	Step 1: The administrator clicks on the Review Reason tab on the admin side navigation bar		Step 2: The system retrieves the following attributes from each entry in the Review Reason Table:							
			<table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>ID</td><td>ReviewReasonID</td></tr><tr><td>Name</td><td>Name</td></tr></table>		Details to be read	Attributes in the table	ID	ReviewReasonID	Name	Name
			Details to be read	Attributes in the table						
			ID	ReviewReasonID						
Name	Name									
and displays them on the Review Reason Screen with the following controls:										

			Control Name	Control Type	Notes
			Review Reason 's List	Heading	Page main heading
			Create new Review Reason	Button	When clicked, this control will direct the administrator to the Create Review Reason screen
			Show "" entries	Label and drop down	This control will be used to set the number of entries the administrator wants to display on a page
			Search bar	Textbox	This control will be used to enter the search Review
			Results table	Table	This control will be used to display all the Review Reason s that exist on the system
			Name	Table Heading	This control will be used as a heading to display the Review Reason 's Name
			Showing "" to "" of "" entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database
			"Previous" and "Next"	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on

	Step 4: The administrator enters their search Review into the search textbox		Step 5: The system searches the table rows in the data table and displays the Review Reason (s) matching the search Review [ALT]
	ALT Step 5: The search Review entered does not match any records existing in the ReviewReason entity. The system displays a “No Records Found” message in the results table.		
CONCLUSION:	This use case concludes when the desired Review Reason's details are displayed on the user interface.		
POST-CONDITION:	The ReviewReason entity was searched for and an entry that corresponds to the search Review or parameter entered.		
BUSINESS RULES	Only the administrators of the system are permitted to search for an existing review reason the system.		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	The system should indicate clearly that no results were found in an instance where the search query details could not retrieve a match.		
ASSUMPTIONS:	The system is online and the administrator has a stable internet connection.		
OPEN ISSUES:	None		

Table 68 - 7.3 Update Review Reason

REVIEW SUBSYSTEM			
AUTHORS (s): Aphiwe Shoji		DATE: 14/05/2022	
VERSION: 1.0		LAST REVIEW DATE: 14/05/2022	
USE CASE NAME:	Update Review Reason		USE CASE TYPE
USE CASE ID:	7.3		Business Requirements: ☐
PRIORITY:	Medium		System Analysis: ☐
SOURCE:	ByteXcom Requirements List		System Design: ☒
PRIMARY BUSINESS ACTOR	Administrator		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This use case describes the process in which the administrator wishes to update the details of Review Reason record in the Review Reason Entity table. The use case begins when the administrator updates an existing Review Reason in the system. The administrator will update all of the pertinent information regarding the Review Reason and the system will update the record in the Review Reason entity table		
PRE-CONDITION:	The administrator must be logged in to the system.		
TRIGGER:	The use is triggered when the administrator wishes to update a Review Reason		
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE	
		MANUAL ACTION	AUTOMATED ACTION
	Step 1: The administrator wishes to update a Review Reason in the system.		Step 2: The system invokes use case 7.12 Search Review Reason
	Step 3: The administrator clicks on the update button		Step 3: The system retrieves the following attributes from the following tables From the Review Reason Table:

		<table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>ID</td><td>ReviewReasonID</td></tr><tr><td>Name</td><td>Name</td></tr></table> and displays them on the Update Review Reason Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Create Review Reason</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Name</td><td>Label</td><td>This control will be used as a label to indicate the Review Reason 's name attribute</td></tr><tr><td>Name</td><td>Textbox</td><td>This control will be used by the administrator to enter the Review Reason 's name</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the updating of the Review Reason and return the administrator to the Review Reason Screen</td></tr></table>	Details to be read	Attributes in the table	ID	ReviewReasonID	Name	Name	Control Name	Control Type	Notes	Create Review Reason	Heading	Page main heading	Name	Label	This control will be used as a label to indicate the Review Reason 's name attribute	Name	Textbox	This control will be used by the administrator to enter the Review Reason 's name	Cancel	Button	This control will be used to cancel the updating of the Review Reason and return the administrator to the Review Reason Screen
Details to be read	Attributes in the table																						
ID	ReviewReasonID																						
Name	Name																						
Control Name	Control Type	Notes																					
Create Review Reason	Heading	Page main heading																					
Name	Label	This control will be used as a label to indicate the Review Reason 's name attribute																					
Name	Textbox	This control will be used by the administrator to enter the Review Reason 's name																					
Cancel	Button	This control will be used to cancel the updating of the Review Reason and return the administrator to the Review Reason Screen																					
Step 4: The system administrator enters the updated Review Reason details and clicks on the update button [ALT]		Step 5: The system captures and validates the entered details against the following tables: [ALT] From the Review Reason Table <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Name</td><td>Name</td><td>Required Alphabetical characters only Minimum: 1 Maximum: 50</td></tr></table>	Details to be saved	Attributes in the table	Notes	Name	Name	Required Alphabetical characters only Minimum: 1 Maximum: 50															
Details to be saved	Attributes in the table	Notes																					
Name	Name	Required Alphabetical characters only Minimum: 1 Maximum: 50																					

			Step 6: The system displays a modal dialogue with the following controls:												
			<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>“Confirm Review Reason Details”</td><td>Label</td><td>This control is used to request confirmation on the updated Review Reason details</td></tr><tr><td>Confirm</td><td>Button</td><td>This control is used to confirm the updating of the Review Reason details</td></tr><tr><td>Cancel</td><td>Button</td><td>This control is used to close the modal dialogue and return the administrator to the Update Review Reason Screen</td></tr></table>	Control Name	Control Type	Notes	“Confirm Review Reason Details”	Label	This control is used to request confirmation on the updated Review Reason details	Confirm	Button	This control is used to confirm the updating of the Review Reason details	Cancel	Button	This control is used to close the modal dialogue and return the administrator to the Update Review Reason Screen
	Control Name	Control Type	Notes												
	“Confirm Review Reason Details”	Label	This control is used to request confirmation on the updated Review Reason details												
Confirm	Button	This control is used to confirm the updating of the Review Reason details													
Cancel	Button	This control is used to close the modal dialogue and return the administrator to the Update Review Reason Screen													
Step 7: The system administrator clicks on the confirm button [ALT]		Step 8: The system saves the updated details into the Review Reason table using an SQL UPDATE Review:													
		<table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>ID</td><td>ID</td><td>Generated by the system</td></tr><tr><td>Name</td><td>Name</td><td>Required Alphabetical characters only Minimum: 1 Maximum: 50</td></tr></table>	Details to be saved	Attributes in the table	Notes	ID	ID	Generated by the system	Name	Name	Required Alphabetical characters only Minimum: 1 Maximum: 50				
Details to be saved	Attributes in the table	Notes													
ID	ID	Generated by the system													
Name	Name	Required Alphabetical characters only Minimum: 1 Maximum: 50													
		Step 9: The system directs the administrator to the Review Reason Screen and displays a bootstrap alert notification with the following controls:													
		<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>“Review Reason updated successfully”</td><td>Label</td><td>This control is used to indicate a successful creation</td></tr></table>	Control Name	Control Name	Notes	“Review Reason updated successfully”	Label	This control is used to indicate a successful creation							
Control Name	Control Name	Notes													
“Review Reason updated successfully”	Label	This control is used to indicate a successful creation													
ALTERNATE COURSES:	ALT Step 5: The administrator does not want to update the Review Reason anymore • The administrator clicks on the cancel button • The use case is terminated														

	ALT Step 5A: The details entered are not in the correct format or the required fields are empty									
	The system displays a validation error modal dialogue with the following controls									
	<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Name</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>“The Name field is required” or “Invalid Name provided”</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Name textbox</td></tr></table>	Control Name	Control Name	Notes	Name	Textbox	This control will be outline in red to show a validation error	“The Name field is required” or “Invalid Name provided”	Label	This control will be used as a label to indicate a validation error in the Name textbox
	Control Name	Control Name	Notes							
	Name	Textbox	This control will be outline in red to show a validation error							
	“The Name field is required” or “Invalid Name provided”	Label	This control will be used as a label to indicate a validation error in the Name textbox							
Return to step 3										
ALT Step 5B: The Review Reason already exists on the system										
The system displays an error modal dialogue with the following controls										
	<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Review Reason already exists!</td><td>Label</td><td>This control is used to inform the administrator that the Review Reason already exists</td></tr><tr><td>Okay</td><td>Button</td><td>This control is used to cancel the updating of the Review Reason</td></tr></table>	Control Name	Control Name	Notes	Review Reason already exists!	Label	This control is used to inform the administrator that the Review Reason already exists	Okay	Button	This control is used to cancel the updating of the Review Reason
	Control Name	Control Name	Notes							
	Review Reason already exists!	Label	This control is used to inform the administrator that the Review Reason already exists							
	Okay	Button	This control is used to cancel the updating of the Review Reason							
- The administrator clicks the “Okay” button - The use case is terminated										
	ALT Step 7: The administrator clicks on the cancel button									
	Return to Step 4 or terminate the use case									
CONCLUSION:	This use case concludes when the details of the respective Review Reason have been updated successfully on the system and a message is displayed to confirm that the update was successful.									
POST-CONDITION:	The Review Reason details are updated and saved to the ReviewReason entity in the database.									
BUSINESS RULES	Only the administrator can update the details of an existing Review Reason.									
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	- No duplicate review reasons may exist at any time. - The system is offline or the administrator does not have access to the internet and cannot update the details of an existing review reason.									
ASSUMPTIONS:	None									
OPEN ISSUES:	None									

Table 69 - 7.4 Delete Review Reason

REVIEW SUBSYSTEM			
AUTHORS (s): Aphiwe Shoji		DATE: 14/05/2022	
VERSION: 1.0		LAST REVIEW DATE: 14/05/2022	
USE CASE NAME:	Delete Review Reason	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒	
USE CASE ID:	7.4		
PRIORITY:	Medium		
SOURCE:	ByteXcom Requirements List		
PRIMARY BUSINESS ACTOR	• Administrator		
PRIMARY SYSTEM ACTOR	• None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	This use case narrative describes the process in which the administrator wishes to delete a Review Reason from the system. The administrator will search for the required Review Reason selected for deletion. The administrator will then confirm the deletion of the Review Reason and the use case concludes when the system notifies the administrator of a successful deletion.		
PRE-CONDITION:	The administrator must be logged in to the system.		
TRIGGER:	This use case is called when the administrator wishes to delete a Review Reason .		
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE	
		MANUAL ACTION	AUTOMATED ACTION
	Step 1: The administrator wishes to delete a Review Reason in the system.		Step 2: The system invokes use case 7.12 Search Review Reason
	Step 3: The administrator clicks on the Delete button		Step 4: The system retrieves the following attributes in the following tables From the Review Reason Table:

			<table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>ID</td><td>ReviewReasonID</td></tr><tr><td>Name</td><td>Name</td></tr></table>	Details to be read	Attributes in the table	ID	ReviewReasonID	Name	Name									
			Details to be read	Attributes in the table														
			ID	ReviewReasonID														
			Name	Name														
			and displays them on the Delete Review Reason Details Screen with the following controls:															
			<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Create Review Reason</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Name</td><td>Label</td><td>This control will be used to display the Review Reason 's name</td></tr><tr><td>Delete</td><td>Button</td><td>This control will be used to confirm the deletion of the Review Reason</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the deletion of the Review Reason and return the administrator to the Review Reason Screen</td></tr></table>	Control Name	Control Type	Notes	Create Review Reason	Heading	Page main heading	Name	Label	This control will be used to display the Review Reason 's name	Delete	Button	This control will be used to confirm the deletion of the Review Reason	Cancel	Button	This control will be used to cancel the deletion of the Review Reason and return the administrator to the Review Reason Screen
			Control Name	Control Type	Notes													
			Create Review Reason	Heading	Page main heading													
			Name	Label	This control will be used to display the Review Reason 's name													
			Delete	Button	This control will be used to confirm the deletion of the Review Reason													
Cancel	Button	This control will be used to cancel the deletion of the Review Reason and return the administrator to the Review Reason Screen																
Step 5: The system displays a modal dialogue with the following controls:																		
<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>"Confirm Review Reason Deletion"</td><td>Label</td><td>This control is used to request confirmation on the deletion of the Review Reason details</td></tr><tr><td>Confirm</td><td>Button</td><td>This control is used to confirm the deletion of the Review Reason details</td></tr><tr><td>Cancel</td><td>Button</td><td>This control is used to close the modal dialogue</td></tr></table>	Control Name	Control Type	Notes	"Confirm Review Reason Deletion"	Label	This control is used to request confirmation on the deletion of the Review Reason details	Confirm	Button	This control is used to confirm the deletion of the Review Reason details	Cancel	Button	This control is used to close the modal dialogue						
Control Name	Control Type	Notes																
"Confirm Review Reason Deletion"	Label	This control is used to request confirmation on the deletion of the Review Reason details																
Confirm	Button	This control is used to confirm the deletion of the Review Reason details																
Cancel	Button	This control is used to close the modal dialogue																

	Step 5: The system administrator clicks on the confirm button [ALT]		Step 6: The system deletes the selected Review Reason from the Review Reason entity in the database using an SQL DELETE Review. [ALT]								
			<table><tr><th>Details to be deleted</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>ID</td><td>ID</td><td>Generated by the system</td></tr><tr><td>Name</td><td>Name</td><td>Required Alphabetical characters only Minimum: 1 Maximum: 50</td></tr></table>	Details to be deleted	Attributes in the table	Notes	ID	ID	Generated by the system	Name	Name
Details to be deleted	Attributes in the table	Notes									
ID	ID	Generated by the system									
Name	Name	Required Alphabetical characters only Minimum: 1 Maximum: 50									
			Step 7: The system redirects the administrator to the Review Reason Screen and displays a modal dialogue with the following controls:								
			<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"Review Reason Deleted Successfully"</td><td>Label</td><td>This control is used to indicate a successful deletion</td></tr></table>	Control Name	Control Name	Notes	"Review Reason Deleted Successfully"	Label	This control is used to indicate a successful deletion		
Control Name	Control Name	Notes									
"Review Reason Deleted Successfully"	Label	This control is used to indicate a successful deletion									
ALTERNATE COURSES:	ALT Step 5: The administrator does not want to delete the Review Reason anymore - The administrator clicks on the cancel button - The use case is terminated ALT Step 6 : The Review Reason cannot be deleted as it has an association with the Order Review Table - The system redirects the administrator and displays a bootstrap alert notification with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"Review Reason cannot be deleted"</td><td>Label</td><td>This control is used to indicate an unsuccessful deletion</td></tr></table> The use case is terminated			Control Name	Control Name	Notes	"Review Reason cannot be deleted"	Label	This control is used to indicate an unsuccessful deletion		
Control Name	Control Name	Notes									
"Review Reason cannot be deleted"	Label	This control is used to indicate an unsuccessful deletion									
CONCLUSION:	This use case concludes when a Review Reason record has been deleted from the system and the system confirms that the review reason has been removed successfully by displaying an alert message.										
POST-CONDITION:	The selected review reason's details are removed from the ReviewReason entity.										
BUSINESS RULES	Only the administrator can delete a Review Reason from the system.										
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	A review reason cannot be deleted if it is associated with an order.										

ASSUMPTIONS:	The system is online and the administrator has a stable internet connection.
OPEN ISSUES:	None

Table 70 - 7.5 View Order Review

REVIEW SUBSYSTEM				
AUTHORS (s): Ofhani Mungani		DATE: 20/07/2022		
VERSION: 1.0		LAST REVIEW DATE: 20/07/2022		
USE CASE NAME:	View Order Review		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒	
USE CASE ID:	7.5			
PRIORITY:	High			
SOURCE:	ByteXcom Requirements List			
PRIMARY BUSINESS ACTOR	• Administrator/Sales Assistant			
PRIMARY SYSTEM ACTOR	• None			
OTHER PARTICIPATING ACTORS:	• None			
OTHER INTERESTED STAKEHOLDERS:	• None			
DESCRIPTION:	This use case describes an event where the administrator or sales assistant wants to view an order review on the system. The use case begins when the administrator/ sales assistant requests to view an order review and the system responds by retrieving the order review details. The use case concludes when the details are displayed on the screen.			
PRE-CONDITION:	• The administrator or sales assistant must be logged on the system.			
TRIGGER:	The use case is called when the administrator or sales assistant wants to view an order review			
TYPICAL COURSE OF EVENTS:		SYSTEM RESPONSE		
	ACTOR ACTION	MANUAL ACTION	AUTOMATED ACTION	
	Step 1: The administrator or sales assistant clicks on the “View Order Reviews” on the side navi link.		Step 2: The system retrieves the values with following attributes From the ReviewReason entity: <table><tr><td>Details to be read</td><td>Attributes in the table</td></tr></table>	Details to be read
Details to be read	Attributes in the table			

			<table><tr><td>ReviewReason</td><td>Name</td></tr></table>	ReviewReason	Name																			
ReviewReason	Name																							
			From the OrderReview entity:																					
			<table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Date</td><td>Date</td></tr><tr><td>Description</td><td>Description</td></tr><tr><td>Rating</td><td>Rating</td></tr></table>	Details to be read	Attributes in the table	Date	Date	Description	Description	Rating	Rating													
Details to be read	Attributes in the table																							
Date	Date																							
Description	Description																							
Rating	Rating																							
			From the Order entity:																					
			<table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Order Number</td><td>OrderId</td></tr><tr><td>FirstName</td><td>FirstName</td></tr><tr><td>Surname</td><td>Surname</td></tr></table>	Details to be read	Attributes in the table	Order Number	OrderId	FirstName	FirstName	Surname	Surname													
Details to be read	Attributes in the table																							
Order Number	OrderId																							
FirstName	FirstName																							
Surname	Surname																							
			The system then displays the Order Reviews Screen with the following controls:																					
			<table><tr><td>Control Name</td><td>Control Type</td><td>Notes</td></tr><tr><td>Order Reviews</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Search bar</td><td>Textbox</td><td>This control will be used when the user inserts the search parameters</td></tr><tr><td>Results table</td><td>Table</td><td>This control will be used to display all the order reviews that exist on the system</td></tr><tr><td>Show “” entries</td><td>Label and drop down</td><td>This control will be used to set the number of entries the administrator wants to display on a page</td></tr><tr><td>Order Reviews List</td><td>Table Heading</td><td>This control will be used as a heading to display the order reviews for all Orders</td></tr><tr><td>FirstName</td><td>label</td><td>This control will be used as a label to display the first name for all orders</td></tr></table>	Control Name	Control Type	Notes	Order Reviews	Heading	Page main heading	Search bar	Textbox	This control will be used when the user inserts the search parameters	Results table	Table	This control will be used to display all the order reviews that exist on the system	Show “” entries	Label and drop down	This control will be used to set the number of entries the administrator wants to display on a page	Order Reviews List	Table Heading	This control will be used as a heading to display the order reviews for all Orders	FirstName	label	This control will be used as a label to display the first name for all orders
Control Name	Control Type	Notes																						
Order Reviews	Heading	Page main heading																						
Search bar	Textbox	This control will be used when the user inserts the search parameters																						
Results table	Table	This control will be used to display all the order reviews that exist on the system																						
Show “” entries	Label and drop down	This control will be used to set the number of entries the administrator wants to display on a page																						
Order Reviews List	Table Heading	This control will be used as a heading to display the order reviews for all Orders																						
FirstName	label	This control will be used as a label to display the first name for all orders																						

			Surname	label	This control will be used as a label to display the surname for all orders
			Rating	label	This control will be used as a label to display the rating for all orders
			Date	label	This control will be used as a label to display the date for all order reviews
			Order Number	label	This control will be used as a label to display the orderId for all orders
			Description	label	This control will be used as a label to display the description for all order reviews
			View Order	button	This control will be used to navigate to the Order Details Screen
			Showing “” to “” of “” entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database
			“Previous” and “Next”	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on
	Step 3: The administrator or sales assistant clicks		Step 4: The system Invoke use case 5.7. View Orders Overview		

	on the View Order button. [ALT]		
ALTERNATE COURSES:	ALT Step 3: The administrator or sales assistant doesn't wish to view the order details. The use case concludes.		
CONCLUSION:	The use case concludes when more details about the order is displayed by invoking use case 5.7. View Orders Overview		
POST-CONDITION:	Order details is read from the database. .		
BUSINESS RULES	None		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	The administrator or sales assistant cannot view their order details without an internet connection. None		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

2.8. Inventory Management Subsystem

Table 71 - 8.1 Create Inventory Item

INVENTORY MANAGEMENT SUBSYSTEM			
AUTHORS (s): Ofhani Mungani		DATE: 16/05/2020	
VERSION: 1.0		LAST REVIEW DATE: 16/05/2020	
USE CASE NAME:	Create Inventory Item		USE CASE TYPE
USE CASE ID:	8.1		Business Requirements: ☐
PRIORITY:	High		System Analysis: ☐
SOURCE:	ByteXcom Requirements List		System Design: ☒
PRIMARY BUSINESS ACTOR	Administrator		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This use case describes the process in which the administrator wishes to add a new inventory item to the system. The use case begins when the administrator adds the new inventory item as an option to choose from when placing an order to a supplier. The administrator will add all the pertinent information regarding the inventory item and save it to the business database.		
PRE-CONDITION:	The administrator must be logged onto the system.		
TRIGGER:	The use case is triggered when the administrator wants to add a new Inventory Item.		
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE	
		MANUAL ACTION	AUTOMATED ACTION
	Step 1: The administrator navigates to the Inventory Item Screen under the		Step 3: The system retrieves Inventory Item details: From the InventoryItem entity with the following attributes

Inventory Item tab
on the side bar

Details to be read	Attributes in the table	Notes
Inventory Item	inventory ItemName	
Quantity on Hand	QuantityOn Hand	

From the **InventoryType** entity with the following attributes:

Details to be read	Attributes in the table	Notes
Inventory Type	inventory TypeId	

The system then displays the **Inventory Item Screen** with the following controls:

Control Name	Control Type	Notes
Inventory Items List	Heading	Page main heading
Create Inventory Item	Button	When clicked, this control will direct the user to the Add Inventory Item page
Search bar	Textbox	This control will be used when the user inserts the search parameters
Results table	Table	This control will be used to display all the Inventory Items that exist on the system
Inventory Item	Table Heading	This control will be used as a heading to display the name for all Inventory Items
Inventory Type	Table Heading	This control will be used as a heading to display the name for all Inventory types
Quantity on Hand	Table Heading	This control will be used as a heading to display the value of all Inventory

					Item's quantity on hand
			Show "" entries	Label and drop down	This control will be used to set the number of entries the administrator wants to display on a page
			Showing "" to "" of "" entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database
			"Previous" and "Next"	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on
	Step 3: The system administrator clicks on the Create Inventory Item button		Step 4: The system retrieves the values from the InventoryType entity with the following attributes:		
			Details to be read	Attributes in the table	Notes
			Inventory Type	inventory TypeId	
			The system then displays the Inventory Item Screen with the following controls:		
			Control Name	Control Type	Notes
			Create Inventory Item	Heading	Page main heading
			Inventory Item	Label	This control will be used as a label to indicate the InventoryItem Name attribute of the Inventory Item record
			Inventory Item	Textbox	This control will be used by the administrator to insert a

					name of the Inventory Item
			Inventory Type	Label	This control will be used by the administrator to indicate an inventory type for the inventory item
			Inventory Type	Dropdown	This control will be used by the administrator to select an inventory type for the inventory item
			Quantity on Hand	Label	This control will be used as a label to indicate the quantity on hand attribute of the inventory item record
			Quantity on Hand	Textbox	This control will be used to store the quantity of an inventory item.
			ThresholdValue	Label	This control will be used as a label to indicate the threshold Value attribute of the inventory item record
			ThersholdValue	Textbox	This control will be used to store the thershold of an inventory item.
			Create	Button	This control will be used to confirm the addition of the Inventory Item
			Cancel	Button	This control will be used to cancel the addition of the Inventory Item
Step 6: The system captures and validates the entered details against the InventoryItem table [ALT]					
			Details to be read	Attributes in the table	Notes

			Inventory Item	Inventory ItemName	Required, Unique,															
			Inventory Type	InventoryTypepeld	Required															
			Quantity on Hand	QuantityonHnad	Required															
			ThersholdValue	ThersholdValue	Required															
			Step 7: The system uses an SQL READ query to check if the inventory item already exists in the InventoryItem table using the following attributes [ALT] <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Name</td><td>InventoryItemName</td><td></td></tr></table>			Details to be read	Attributes in the table	Notes	Name	InventoryItemName										
Details to be read	Attributes in the table	Notes																		
Name	InventoryItemName																			
			Step 8: The system saves the information into the InventoryItem table using an INSERT SQL query: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>ID</td><td>Id</td><td>A unique value generated by the system and saved to the database.</td></tr><tr><td>Inventory Item</td><td>InventoryTypepename</td><td>Required, Unique,</td></tr><tr><td>Inventory Type</td><td>InventoryTypepeld</td><td>Required</td></tr><tr><td>Quantity on Hand</td><td>QuantityonHnad</td><td>Required</td></tr></table>			Details to be saved	Attributes in the table	Notes	ID	Id	A unique value generated by the system and saved to the database.	Inventory Item	InventoryTypepename	Required, Unique,	Inventory Type	InventoryTypepeld	Required	Quantity on Hand	QuantityonHnad	Required
Details to be saved	Attributes in the table	Notes																		
ID	Id	A unique value generated by the system and saved to the database.																		
Inventory Item	InventoryTypepename	Required, Unique,																		
Inventory Type	InventoryTypepeld	Required																		
Quantity on Hand	QuantityonHnad	Required																		
			Step 9: The system redirects the system administrator to the Inventory Item screen and displays a bootstrap alert notification with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>“Inventory Item Successful created”</td><td>Label</td><td>This control is used to indicate a successful creation</td></tr></table>			Control Name	Control Name	Notes	“Inventory Item Successful created”	Label	This control is used to indicate a successful creation									
Control Name	Control Name	Notes																		
“Inventory Item Successful created”	Label	This control is used to indicate a successful creation																		
ALTERNATE COURSES:	ALT Step 5: The administrator does not want to add an Inventory Item anymore - The administrator clicks on the cancel button - The use case is terminated																			

	ALT Step 6: The details entered are not in the correct format or the required fields are empty The system displays a validation error message box with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Inventory Item</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>Inventory Item Name is required</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Inventory Item textbox</td></tr><tr><td>Inventory Type</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>Inventory Type Name is required</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Inventory Type textbox</td></tr><tr><td>Quantity on Hand</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>Quantity on Hand is required</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Quantity on Hand textbox</td></tr></table>	Control Name	Control Name	Notes	Inventory Item	Textbox	This control will be outline in red to show a validation error	Inventory Item Name is required	Label	This control will be used as a label to indicate a validation error in the Inventory Item textbox	Inventory Type	Textbox	This control will be outline in red to show a validation error	Inventory Type Name is required	Label	This control will be used as a label to indicate a validation error in the Inventory Type textbox	Quantity on Hand	Textbox	This control will be outline in red to show a validation error	Quantity on Hand is required	Label	This control will be used as a label to indicate a validation error in the Quantity on Hand textbox
	Control Name	Control Name	Notes																			
	Inventory Item	Textbox	This control will be outline in red to show a validation error																			
	Inventory Item Name is required	Label	This control will be used as a label to indicate a validation error in the Inventory Item textbox																			
	Inventory Type	Textbox	This control will be outline in red to show a validation error																			
	Inventory Type Name is required	Label	This control will be used as a label to indicate a validation error in the Inventory Type textbox																			
	Quantity on Hand	Textbox	This control will be outline in red to show a validation error																			
	Quantity on Hand is required	Label	This control will be used as a label to indicate a validation error in the Quantity on Hand textbox																			
	Return to step 5																					
	ALT Step 7: The Inventory Item already exists on the system The system displays an error message box with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Validation Error</td><td>Label</td><td>This control is used to display the type of message box displayed</td></tr><tr><td>Inventory Item details already exist</td><td>Label</td><td>This control is used to inform the administrator that the Inventory Item already exists</td></tr><tr><td>Okay</td><td>Button</td><td>This control is used to cancel the creation of the user role</td></tr></table>	Control Name	Control Name	Notes	Validation Error	Label	This control is used to display the type of message box displayed	Inventory Item details already exist	Label	This control is used to inform the administrator that the Inventory Item already exists	Okay	Button	This control is used to cancel the creation of the user role									
Control Name	Control Name	Notes																				
Validation Error	Label	This control is used to display the type of message box displayed																				
Inventory Item details already exist	Label	This control is used to inform the administrator that the Inventory Item already exists																				
Okay	Button	This control is used to cancel the creation of the user role																				
- The administrator clicks the OK button - The use case is terminated																						
CONCLUSION:	This use case concludes when an Inventory Item has been added on the system and the system displays a confirmation alert to confirm its details were created successfully.																					
POST-CONDITION:	A new inventory item has been added to the InventoryItem table in the database.																					
BUSINESS RULES	Only the administrator has access to this function.																					
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	- The system is offline, and the system administrator will not be able to add a new Inventory Item. - An inventory item should have an inventory type instance assigned at all times. - No duplicate inventory items should be added to the system.																					
ASSUMPTIONS:	None																					
OPEN ISSUES:	None																					

Table 72 - 8.2 Search Inventory Item

INVENTORY MANAGEMENT SUBSYSTEM		
AUTHORS (s): Ofhani Mungani		DATE: 16/05/2020
VERSION: 1.0		LAST REVIEW DATE: 16/05/2020
USE CASE NAME:	Search Inventory Item	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒
USE CASE ID:	8.2	
PRIORITY:	Medium	
SOURCE:	ByteXcom Requirements List	
PRIMARY BUSINESS ACTOR	• Administrator	
PRIMARY SYSTEM ACTOR	• None	
OTHER PARTICIPATING ACTORS:	• None	
OTHER INTERESTED STAKEHOLDERS:	• None	
DESCRIPTION:	This use case describes the process in which the administrator searches for an inventory item. The system responds by displaying the information relating to the search query.	
PRE-CONDITION:	• The administrator must be logged onto the system.	
TRIGGER:	This use case is called when the administrator wants to search for an Inventory Item on the system.	
		SYSTEM RESPONSE

TYPICAL COURSE OF EVENTS:	ACTOR ACTION	MANUAL ACTION	AUTOMATED ACTION																																	
	Step 1: The administrator wishes to search for an inventory item on the system.																																			
	Step 2: The administrator navigates to the Inventory Item page under the Inventory Item tab on the side bar		Step 3: The system retrieves the values from the InventoryType entity with the following attributes: <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Inventory Type</td><td>inventory TypeId</td><td></td></tr></table> From the InventoryItem entity with the following attributes: <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Inventory Item</td><td>inventory ItemName</td><td></td></tr><tr><td>Quantity on Hand</td><td>QuantityOn Hand</td><td></td></tr></table> The system then displays the Inventory Item Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Create Inventory Item</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Inventory Item</td><td>Label</td><td>This control will be used as a label to indicate the InventoryItem Name attribute of the Inventory Item record</td></tr><tr><td>Inventory Item</td><td>Textbox</td><td>This control will be used by the administrator to insert a name of the Inventory Item</td></tr><tr><td>Inventory Type</td><td>Label</td><td>This control will be used by the administrator to indicate an inventory type for the inventory item</td></tr><tr><td>Inventory Type</td><td>Dropdown</td><td>This control will be used by the administrator to select an</td></tr></table>	Details to be read	Attributes in the table	Notes	Inventory Type	inventory TypeId		Details to be read	Attributes in the table	Notes	Inventory Item	inventory ItemName		Quantity on Hand	QuantityOn Hand		Control Name	Control Type	Notes	Create Inventory Item	Heading	Page main heading	Inventory Item	Label	This control will be used as a label to indicate the InventoryItem Name attribute of the Inventory Item record	Inventory Item	Textbox	This control will be used by the administrator to insert a name of the Inventory Item	Inventory Type	Label	This control will be used by the administrator to indicate an inventory type for the inventory item	Inventory Type	Dropdown	This control will be used by the administrator to select an
	Details to be read	Attributes in the table	Notes																																	
	Inventory Type	inventory TypeId																																		
	Details to be read	Attributes in the table	Notes																																	
	Inventory Item	inventory ItemName																																		
	Quantity on Hand	QuantityOn Hand																																		
	Control Name	Control Type	Notes																																	
	Create Inventory Item	Heading	Page main heading																																	
	Inventory Item	Label	This control will be used as a label to indicate the InventoryItem Name attribute of the Inventory Item record																																	
Inventory Item	Textbox	This control will be used by the administrator to insert a name of the Inventory Item																																		
Inventory Type	Label	This control will be used by the administrator to indicate an inventory type for the inventory item																																		
Inventory Type	Dropdown	This control will be used by the administrator to select an																																		

					inventory type for the inventory item
			Quantity on Hand	Labe	This control will be used as a label to indicate the quantity on hand attribute of the inventory item record
			Quantity on Hand	Textbox	This control will be used to store the quantity of an inventory item.
			Create	Button	This control will be used to confirm the addition of the Inventory Item
			Cancel	Button	This control will be used to cancel the addition of the Inventory Item
		Step 4: The administrator enters their search query into the search textbox		Step 5: The system searches the InventoryType table and displays the Inventory Item(s) matching the search query in the Inventory Item screen [ALT]	
	ALT Step 5: The search query entered does not match any records existing in the InventoryType entity. The system displays a “No Records Found” message in the Inventory Item screen and on the results table.				
CONCLUSION:	This use case concludes when the required inventory item is displayed on the screen.				
POST-CONDITION:	The administrator selects the required Inventory Item from the displayed results.				
BUSINESS RULES	Only the administrator has access to this function.				
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None				
ASSUMPTIONS:	None				
OPEN ISSUES:	None				

Table 73 - 8.3 Update Inventory Item

INVENTORY MANAGEMENT SUBSYSTEM		
AUTHORS (s): Ofhani Mungani		DATE: 16/05/2020
VERSION: 1.0		LAST REVIEW DATE: 16/05/2020
USE CASE NAME:	Update Inventory Item	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒
USE CASE ID:	8.3	
PRIORITY:	Medium	
SOURCE:	ByteXcom Requirements List	
PRIMARY BUSINESS ACTOR	• Administrator	
PRIMARY SYSTEM ACTOR	• None	
OTHER PARTICIPATING ACTORS:	• None	
OTHER INTERESTED STAKEHOLDERS:	• None	
DESCRIPTION:	This use case describes the process in which the administrator wishes to update an inventory item on the system. The use case begins when the administrator updates the inventory item as an option to choose from when placing an order to a supplier. The administrator will update all the pertinent information regarding the inventory item and save it to the business database.	
PRE-CONDITION:	• The administrator must be logged onto the system.	
TRIGGER:	The use case is triggered when the administrator wants to update an Inventory Item.	

TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE																																			
		MANUAL ACTION	AUTOMATED ACTION																																		
	Step 1: The administrator wishes to update an Inventory Item in the system.		Step 2: The system invokes use case 8.6 Search Inventory Item																																		
		Step 2: The administrator clicks on the update icon		Step 3: The system retrieves the selected inventory item details: From the InventoryItem entity with the following attributes: <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Inventory Type</td><td>inventory Typeld</td><td></td></tr></table> From the InventoryItem entity with the following attributes: <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Inventory Item</td><td>inventory ItemName</td><td></td></tr><tr><td>Quantity on Hand</td><td>QuantityOn Hand</td><td></td></tr><tr><td>Low Inventory Alert Threshold</td><td>ThresholdVa lue</td><td></td></tr></table> The system then displays the Update Inventory Item Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Update Inventory Item</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Inventory Item</td><td>Label</td><td>This control will be used as a label to indicate the InventoryIteN ame attribute of the Inventory Item record</td></tr><tr><td>Inventory Item</td><td>Textbox</td><td>This control will be used by the administrator to update a name of the Inventory Item</td></tr><tr><td>Inventory Type</td><td>Label</td><td>This control will be used by the</td></tr></table>		Details to be read	Attributes in the table	Notes	Inventory Type	inventory Typeld		Details to be read	Attributes in the table	Notes	Inventory Item	inventory ItemName		Quantity on Hand	QuantityOn Hand		Low Inventory Alert Threshold	ThresholdVa lue		Control Name	Control Type	Notes	Update Inventory Item	Heading	Page main heading	Inventory Item	Label	This control will be used as a label to indicate the InventoryIteN ame attribute of the Inventory Item record	Inventory Item	Textbox	This control will be used by the administrator to update a name of the Inventory Item	Inventory Type	Label
	Details to be read		Attributes in the table	Notes																																	
	Inventory Type		inventory Typeld																																		
	Details to be read		Attributes in the table	Notes																																	
	Inventory Item		inventory ItemName																																		
	Quantity on Hand		QuantityOn Hand																																		
	Low Inventory Alert Threshold		ThresholdVa lue																																		
Control Name	Control Type		Notes																																		
Update Inventory Item	Heading	Page main heading																																			
Inventory Item	Label	This control will be used as a label to indicate the InventoryIteN ame attribute of the Inventory Item record																																			
Inventory Item	Textbox	This control will be used by the administrator to update a name of the Inventory Item																																			
Inventory Type	Label	This control will be used by the																																			

					administrator to indicate an inventory type for the inventory item
		Inventory Type	Dropdown		This control will be used by the administrator to select an inventory type for the inventory item
		Quantity on Hand	Labe		This control will be used as a label to indicate the quantity on hand attribute of the inventory item record
		Quantity on Hand	Textbox		This control will be used to store the quantity of an inventory item.
		Save	Button		This control will be used to confirm the update of the Inventory Item
		Cancel	Button		This control will be used to cancel the edit of the Inventory Item
	Step 4: The system administrator enters the Inventory Item information and clicks on the save button [ALT]		Step 5: The system captures and validates the entered details against the Inventory Item table [ALT]		
			Step 6: The system uses an SQL READ query to check if the inventory item already exists in the		

			InventoryItem table using the following attributes [ALT] <table><tr><th>Details to be updated</th><th>Attributes in the table</th></tr><tr><td>Name</td><td>InventoryItemName</td></tr></table>	Details to be updated	Attributes in the table	Name	InventoryItemName										
	Details to be updated	Attributes in the table															
	Name	InventoryItemName															
		Step 7: The system displays a message box with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Confirmation required</td><td>Label</td><td>This control is used to display the type of message box displayed</td></tr><tr><td>Confirm updated inventory item details?</td><td>Label</td><td>This control is used to request confirmation on the updated inventory item details</td></tr><tr><td>Confirm</td><td>Button</td><td>This control is used to confirm the updating of the inventory item details</td></tr><tr><td>Cancel</td><td>Button</td><td>This control is used to close the message box for the user to edit the Inventory Item details</td></tr></table>	Control Name	Control Type	Notes	Confirmation required	Label	This control is used to display the type of message box displayed	Confirm updated inventory item details?	Label	This control is used to request confirmation on the updated inventory item details	Confirm	Button	This control is used to confirm the updating of the inventory item details	Cancel	Button	This control is used to close the message box for the user to edit the Inventory Item details
Control Name	Control Type	Notes															
Confirmation required	Label	This control is used to display the type of message box displayed															
Confirm updated inventory item details?	Label	This control is used to request confirmation on the updated inventory item details															
Confirm	Button	This control is used to confirm the updating of the inventory item details															
Cancel	Button	This control is used to close the message box for the user to edit the Inventory Item details															
	Step 8: The system administrator clicks on the confirm button [ALT]	Step 9: The system saves the information into the Inventory Item table using an ALTER TO SQL query: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Inventory Item</td><td>InventoryItemName</td><td>Required, Unique,</td></tr><tr><td>Inventory Type</td><td>InventoryTypeName</td><td>Required</td></tr><tr><td>Quantity on Hand</td><td>QuantityOnHand</td><td>Required</td></tr><tr><td>Low Inventory Alert Threshold</td><td>ThresholdValue</td><td>Required</td></tr></table>	Details to be saved	Attributes in the table	Notes	Inventory Item	InventoryItemName	Required, Unique,	Inventory Type	InventoryTypeName	Required	Quantity on Hand	QuantityOnHand	Required	Low Inventory Alert Threshold	ThresholdValue	Required
Details to be saved	Attributes in the table	Notes															
Inventory Item	InventoryItemName	Required, Unique,															
Inventory Type	InventoryTypeName	Required															
Quantity on Hand	QuantityOnHand	Required															
Low Inventory Alert Threshold	ThresholdValue	Required															

			Step 10: The system redirects the system administrator to the Inventory Item screen and displays a bootstrap alert notification with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Inventory item Successful update"</td><td>Label</td><td>This control is used to indicate a successful update</td></tr></table>	Control Name	Control Name	Notes	Inventory item Successful update"	Label	This control is used to indicate a successful update																					
Control Name	Control Name	Notes																												
Inventory item Successful update"	Label	This control is used to indicate a successful update																												
ALTERNATE COURSES:	ALT Step 4: The administrator does not want to edit the Inventory Item anymore • The administrator clicks on the cancel button • The use case is terminated																													
	ALT Step 5: The details entered are not in the correct format or the required fields are empty • The system displays a validation error message box with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Inventory Item</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>Inventory Item Name is required</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Inventory Item textbox</td></tr><tr><td>Inventory Type</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>Inventory Type Name is required</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Inventory Type textbox</td></tr><tr><td>Quantity on Hand</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>Quantity on Hand is required</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Quantity on Hand textbox</td></tr><tr><td>Low Inventory Alert Threshold</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Low Inventory Alert Threshold textbox</td></tr><tr><td>Low Inventory Alert Threshold</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr></table> • Return to step 4			Control Name	Control Name	Notes	Inventory Item	Textbox	This control will be outline in red to show a validation error	Inventory Item Name is required	Label	This control will be used as a label to indicate a validation error in the Inventory Item textbox	Inventory Type	Textbox	This control will be outline in red to show a validation error	Inventory Type Name is required	Label	This control will be used as a label to indicate a validation error in the Inventory Type textbox	Quantity on Hand	Textbox	This control will be outline in red to show a validation error	Quantity on Hand is required	Label	This control will be used as a label to indicate a validation error in the Quantity on Hand textbox	Low Inventory Alert Threshold	Label	This control will be used as a label to indicate a validation error in the Low Inventory Alert Threshold textbox	Low Inventory Alert Threshold	Textbox	This control will be outline in red to show a validation error
	Control Name	Control Name	Notes																											
	Inventory Item	Textbox	This control will be outline in red to show a validation error																											
	Inventory Item Name is required	Label	This control will be used as a label to indicate a validation error in the Inventory Item textbox																											
	Inventory Type	Textbox	This control will be outline in red to show a validation error																											
	Inventory Type Name is required	Label	This control will be used as a label to indicate a validation error in the Inventory Type textbox																											
	Quantity on Hand	Textbox	This control will be outline in red to show a validation error																											
	Quantity on Hand is required	Label	This control will be used as a label to indicate a validation error in the Quantity on Hand textbox																											
	Low Inventory Alert Threshold	Label	This control will be used as a label to indicate a validation error in the Low Inventory Alert Threshold textbox																											
Low Inventory Alert Threshold	Textbox	This control will be outline in red to show a validation error																												
ALT Step 6: The Inventory Item already exists on the system • The system displays an error message box with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Validation Error</td><td>Label</td><td>This control is used to display the type of message box displayed</td></tr><tr><td>Inventory Item details already exist</td><td>Label</td><td>This control is used to inform the administrator that the Inventory Item already exists</td></tr><tr><td>Okay</td><td>Button</td><td>This control is used to cancel the creation of the user role</td></tr></table> • The administrator clicks the OK button			Control Name	Control Name	Notes	Validation Error	Label	This control is used to display the type of message box displayed	Inventory Item details already exist	Label	This control is used to inform the administrator that the Inventory Item already exists	Okay	Button	This control is used to cancel the creation of the user role																
Control Name	Control Name	Notes																												
Validation Error	Label	This control is used to display the type of message box displayed																												
Inventory Item details already exist	Label	This control is used to inform the administrator that the Inventory Item already exists																												
Okay	Button	This control is used to cancel the creation of the user role																												

	The use case is terminated
	ALT Step 8: The administrator clicks on the cancel button The use case is terminated
CONCLUSION:	This use case concludes when an inventory item has been added on the system
POST-CONDITION:	An inventory item has been updated in the InventoryType table in the database
BUSINESS RULES	Only the administrator has access to this function.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None
ASSUMPTIONS:	None
OPEN ISSUES:	None

Table 74 - 8.4 Delete Inventory Item

INVENTORY MANAGEMENT SUBSYSTEM			
AUTHORS (s): Ofhani Mungani		DATE: 16/05/2020	
VERSION: 1.0		LAST REVIEW DATE: 16/05/2020	
USE CASE NAME:	Delete Inventory Item	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒	
USE CASE ID:	8.4		
PRIORITY:	Medium		
SOURCE:	ByteXcom Requirements List		
PRIMARY BUSINESS ACTOR	• Administrator		
PRIMARY SYSTEM ACTOR	• None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	This use case narrative describes the process in which the administrator wishes to delete an inventory item from the system. The usecase begins with the administrator searching for the required inventory item selected for removal. The Inventory Item will be removed from the system database.		
PRE-CONDITION:	• The administrator must be logged onto the system.		
TRIGGER:	The use case is triggered when the administrator wants to delete an Inventory Item existing in the InventoryType table.		
TYPICAL COURSE OF EVENTS:		SYSTEM RESPONSE	
	ACTOR ACTION	MANUAL ACTION	AUTOMATED ACTION
	Step 1: The administrator wishes to delete an		Step 2: The system invokes use case 8.6 Search Inventory Item

	Inventory Item in the system.																																
	Step 3: The administrator clicks on the Delete icon		Step 4: The system retrieves the selected Inventory Item details: From the InventoryItem entity with the following attributes: <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Inventory Type</td><td>inventory TypeId</td><td></td></tr></table> From the InventoryItem entity with the following attributes: <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Inventory Item</td><td>inventory ItemName</td><td></td></tr><tr><td>Quantity on Hand</td><td>QuantityOn Hand</td><td></td></tr></table> The system then displays the Inventory Item Details Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Inventory Item</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Inventory Item</td><td>Label</td><td>This control will be used as a heading to display the Name for all Inventory Items</td></tr><tr><td>Inventory Type</td><td>Label</td><td>This control will be used as a heading to display the Name for all Inventory Type</td></tr><tr><td>Quantity on Hand</td><td>Label</td><td>This control will be used as a heading to display the value for Quantity on Hand for inventory item</td></tr></table>	Details to be read	Attributes in the table	Notes	Inventory Type	inventory TypeId		Details to be read	Attributes in the table	Notes	Inventory Item	inventory ItemName		Quantity on Hand	QuantityOn Hand		Control Name	Control Type	Notes	Inventory Item	Heading	Page main heading	Inventory Item	Label	This control will be used as a heading to display the Name for all Inventory Items	Inventory Type	Label	This control will be used as a heading to display the Name for all Inventory Type	Quantity on Hand	Label	This control will be used as a heading to display the value for Quantity on Hand for inventory item
	Details to be read	Attributes in the table	Notes																														
Inventory Type	inventory TypeId																																
Details to be read	Attributes in the table	Notes																															
Inventory Item	inventory ItemName																																
Quantity on Hand	QuantityOn Hand																																
Control Name	Control Type	Notes																															
Inventory Item	Heading	Page main heading																															
Inventory Item	Label	This control will be used as a heading to display the Name for all Inventory Items																															
Inventory Type	Label	This control will be used as a heading to display the Name for all Inventory Type																															
Quantity on Hand	Label	This control will be used as a heading to display the value for Quantity on Hand for inventory item																															
Step 5: The administrator clicks on the Delete button [ALT]		Step 6: The system displays a modal dialogue with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr></table>	Control Name	Control Type	Notes																												
Control Name	Control Type	Notes																															

			Confirmation required	Label	This control is used to display the type of message box displayed
			Confirm updated Inventory Item details?	Label	This control is used to request confirmation of the deletion of the Inventory Item
			Confirm	Button	This control is used to confirm the deletion of the Inventory Item details
			Cancel	Button	This control is used to close the message box deletion of the Inventory Item
	Step 7: The system administrator clicks on the Confirm button [ALT]		Step 8 The system deletes the selected inventory item from the InventoryItem entity in the system database. [ALT]		
ALTERNATE COURSES:			Step 9: The system redirects the system administrator to the Inventory Item screen and displays a bootstrap alert notification with the following controls:		
			Control Name	Control Name	Notes
			Successful deletion	Label	This control is used to indicate a successful deletion
	ALT Step 5: The administrator does not want to delete the Inventory Item anymore - The administrator clicks on the cancel button - The use case is terminated				
	ALT Step 7: The administrator does not want to confirm to delete the Inventory Item anymore Return to step 5				
	ALT Step 8: The Inventory Item has attached to an existing InventoryProcured , ProductInventoryItem or InventoryWrittenOff entity on the system. The system displays a bootstrap alert notification with the following controls:				

		Control Name Unsuccessful deletion	Control Name Label	Notes This control is used to indicate an unsuccessful deletion	
CONCLUSION:	This use case concludes when an inventory item has been deleted on the system				
POST-CONDITION:	The inventory item has been removed from the system database.				
BUSINESS RULES	Only the administrator has access to this function.				
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None				
ASSUMPTIONS:	None				
OPEN ISSUES:	None				

Table 75 - 8.5 Create Inventory Type

INVENTORY MANAGEMENT SUBSYSTEM			
AUTHORS (s): Ofhani Mungani		DATE: 16/05/2022	
VERSION: 1.0		LAST REVIEW DATE: 30/08/2022	
USE CASE NAME:	Create Inventory Type	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒	
USE CASE ID:	8.5		
PRIORITY:	High		
SOURCE:	ByteXcom Requirements List		
PRIMARY BUSINESS ACTOR	• Administrator		
PRIMARY SYSTEM ACTOR	• None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	This use case describes the process in which the administrator wishes to add a new inventory type to the system. The use case begins when the administrator adds the new inventory type as an option to choose from when creating an inventory item. The administrator will add all the pertinent information regarding the inventory type and save it to the business database.		
PRE-CONDITION:	• The administrator must be logged onto the system.		
TRIGGER:	The use case is triggered when the administrator wants to add a new Inventory Type.		
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE	
		MANUAL ACTION	AUTOMATED ACTION

Step 1: The administrator navigates to the **Inventory Type Screen** under the Inventory Type tab on the side bar

Step 2: The system retrieves Inventory Type details from the **InventoryType** entity with following attributes:

Details to be read	Attributes in the table	Notes
Inventory Type	inventory TypeName	

The system then displays the **Inventory Type Screen** with the following controls:

Control Name	Control Type	Notes
Inventory Types List	Heading	Page main heading
Create Inventory Type	Button	When clicked, this control will direct the user to the Add Inventory Type page
Search bar	Textbox	This control will be used when the user inserts the search parameters
Results table	Table	This control will be used to display all the Inventory Types that exist on the system
Show "" entries	Label and drop down	This control will be used to set the number of entries the administrator wants to display on a page
Inventory Type	Table Heading	This control will be used as a heading to display the name for all Inventory Types
Showing "" to "" of "" entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database

			"Previous" and "Next"	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on																
	Step 3: The system administrator clicks on the Create Inventory Type button		Step 4: The system displays the Create Inventory Type Screen with the following controls:																		
			<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Create Inventory Type</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Inventory Type</td><td>Label</td><td>This control will be used as a label to indicate the Name attribute of the Inventory Type record</td></tr><tr><td>Inventory Type</td><td>Textbox</td><td>This control will be used by the administrator to insert an Inventory Type</td></tr><tr><td>Create</td><td>Button</td><td>This control will be used to confirm the addition of the Inventory Type</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the addition of the Inventory Type</td></tr></table>	Control Name	Control Type	Notes	Create Inventory Type	Heading	Page main heading	Inventory Type	Label	This control will be used as a label to indicate the Name attribute of the Inventory Type record	Inventory Type	Textbox	This control will be used by the administrator to insert an Inventory Type	Create	Button	This control will be used to confirm the addition of the Inventory Type	Cancel	Button	This control will be used to cancel the addition of the Inventory Type
	Control Name	Control Type	Notes																		
Create Inventory Type	Heading	Page main heading																			
Inventory Type	Label	This control will be used as a label to indicate the Name attribute of the Inventory Type record																			
Inventory Type	Textbox	This control will be used by the administrator to insert an Inventory Type																			
Create	Button	This control will be used to confirm the addition of the Inventory Type																			
Cancel	Button	This control will be used to cancel the addition of the Inventory Type																			
Step 5: The system administrator enters the details and clicks on the Create button [ALT]		Step 6: The system captures and validates the entered details against the InventoryType table [ALT]																			
		<table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Inventory Type</td><td>InventoryTypeName</td><td>Required, Unique,</td></tr></table>	Details to be read	Attributes in the table	Notes	Inventory Type	InventoryTypeName	Required, Unique,													
Details to be read	Attributes in the table	Notes																			
Inventory Type	InventoryTypeName	Required, Unique,																			
		Step 7: The system uses an SQL READ query to check if the inventory type already exists in the InventoryType table using the following attributes																			

			[ALT] <table><tr><th>Details to be read</th><th colspan="2">Attributes in the table Notes</th></tr><tr><td>Name</td><td colspan="2">InventoryTypeName</td></tr></table>	Details to be read	Attributes in the table Notes		Name	InventoryTypeName			
	Details to be read	Attributes in the table Notes									
	Name	InventoryTypeName									
		Step 8: The system saves the information into the InventoryItem table using an INSERT SQL query: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>ID</td><td>Id</td><td>A unique value generated by the system, incremented by 1 every time a new inventory type record is inserted in the database</td></tr><tr><td>Inventory Type</td><td>InventoryTypeName</td><td>Required, Unique,</td></tr></table>	Details to be saved	Attributes in the table	Notes	ID	Id	A unique value generated by the system, incremented by 1 every time a new inventory type record is inserted in the database	Inventory Type	InventoryTypeName	Required, Unique,
Details to be saved	Attributes in the table	Notes									
ID	Id	A unique value generated by the system, incremented by 1 every time a new inventory type record is inserted in the database									
Inventory Type	InventoryTypeName	Required, Unique,									
		Step 9: The system redirects the system administrator to the Inventory Type screen and displays bootstrap alert notifications with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"Inventory Type Successful created"</td><td>Label</td><td>This control is used to indicate a successful Creation</td></tr><tr><td>"Click here to add a new inventory item"</td><td>Label</td><td>Suggestion link to create a new inventory item</td></tr></table>	Control Name	Control Name	Notes	"Inventory Type Successful created"	Label	This control is used to indicate a successful Creation	"Click here to add a new inventory item"	Label	Suggestion link to create a new inventory item
Control Name	Control Name	Notes									
"Inventory Type Successful created"	Label	This control is used to indicate a successful Creation									
"Click here to add a new inventory item"	Label	Suggestion link to create a new inventory item									
ALTERNATE COURSES:	ALT Step 5: The administrator does not want to add an Inventory Type anymore - The administrator clicks on the cancel button - The use case is terminated										
	ALT Step 6: The details entered are not in the correct format or the required fields are empty The system displays a validation error message box with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr></table>			Control Name	Control Name	Notes					
Control Name	Control Name	Notes									

	Inventory Type	Textbox	This control will be outline in red to show a validation error
	Inventory Type Name is required	Label	This control will be used as a label to indicate a validation error in the Name textbox
	Return to step 5		
	ALT Step 7: The Inventory Type already exists on the system		
	The system displays an error message box with the following controls		
	Control Name	Control Name	Notes
	Validation Error	Label	This control is used to display the type of message box displayed
	Inventory Type details already exist	Label	This control is used to inform the user that the Inventory Type already exists
	Okay	Button	This control is used to cancel the creation of the user role
	- The administrator clicks the OK button - The use case is terminated		
CONCLUSION:	This use case concludes when an Inventory Type has been added on the system		
POST-CONDITION:	An inventory type has been added to the InventoryType table in the database		
BUSINESS RULES	Only the administrator has access to this function.		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	The system is offline, and the system administrator will not be able to add a new Inventory Type.		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

Table 76 - 8.6 Search Inventory Type

INVENTORY MANAGEMENT SUBSYSTEM				
AUTHORS (s): Ofhani Mungani			DATE: 16/05/2022	
VERSION: 1.0			LAST REVIEW DATE: 30/08/2022	
USE CASE NAME:	Search Inventory Type		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒	
USE CASE ID:	8.6			
PRIORITY:	Medium			
SOURCE:	ByteXcom Requirements List			
PRIMARY BUSINESS ACTOR	• Administrator			
PRIMARY SYSTEM ACTOR	• None			
OTHER PARTICIPATING ACTORS:	• None			
OTHER INTERESTED STAKEHOLDERS:	• None			
DESCRIPTION:	This use case describes the process in which the administrator searches for an inventory type. The system responds by displaying the information relating to the search query.			
PRE-CONDITION:	• The administrator must be logged onto the system.			
TRIGGER:	This use case is called when the administrator wants to search for an Inventory Type on the system.			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE		
		MANUAL ACTION	AUTOMATED ACTION	
	Step 1: The administrator wishes to search for an inventory type on the system.			
	Step 2: The administrator navigates to the Inventory Type page under the Inventory Type tab on the side bar		Step 3: The system retrieves the existing items in the InventoryType table with the following attributes:	
		Details to be read	Attributes in the table	Notes
		Inventory Type	inventory TypeName	

The system then displays the **Inventory Type Screen** with the following controls:

Control Name	Control Type	Notes
Inventory Types List	Heading	Page main heading
Create Inventory Type	Button	When clicked, this control will direct the user to the Add Inventory Type page
Search bar	Textbox	This control will be used when the user inserts the search parameters
Results table	Table	This control will be used to display all the Inventory Types that exist on the system
Show "" entries	Label and drop down	This control will be used to set the number of entries the administrator wants to display on a page
Inventory Type	Table Heading	This control will be used as a heading to display the name for all Inventory Types
Showing "" to "" of "" entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database
"Previous" and "Next"	Label and button	This control is used to navigate through the entry pages and to indicate on which page the

					administrator is currently on
	Step 4: The administrator enters their search query into the search textbox		Step 5: The system searches the InventoryType table and displays the Inventory Type(s) matching the search query in the Inventory Type screen [ALT]		
	ALT Step 5: The search query entered does not match any records existing in the InventoryType entity. The system displays a “No Records Found” message in the Inventory Type screen and on the results table.				
CONCLUSION:	This use case concludes when the required inventory type is displayed on the screen.				
POST-CONDITION:	The administrator selects the required Inventory Type from the displayed results.				
BUSINESS RULES	Only the administrator has access to this function.				
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None				
ASSUMPTIONS:	None				
OPEN ISSUES:	None				

Table 77 - 8.7 Update Inventory Type

INVENTORY MANAGEMENT SUBSYSTEM				
AUTHORS (s): Ofhani Mungani			DATE: 16/05/2022	
VERSION: 1.0			LAST REVIEW DATE: 30/08/2022	
USE CASE NAME:	Update Inventory Type		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒	
USE CASE ID:	8.7			
PRIORITY:	Medium			
SOURCE:	ByteXcom Requirements List			
PRIMARY BUSINESS ACTOR	• Administrator			
PRIMARY SYSTEM ACTOR	• None			
OTHER PARTICIPATING ACTORS:	• None			
OTHER INTERESTED STAKEHOLDERS:	• None			
DESCRIPTION:	This use case describes the process in which the administrator wishes to update an inventory type on the system. The use case begins when the administrator updates the inventory type as an option to choose from when creating an inventory item. The administrator will update all the pertinent information regarding the inventory type and save it to the business database.			
PRE-CONDITION:	• The administrator must be logged onto the system.			
TRIGGER:	The use case is triggered when the administrator wants to update an Inventory Type.			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE		
		MANUAL ACTION	AUTOMATED ACTION	
	Step 1: The administrator wishes to update an Inventory Type in the system.		Step 2: The system invokes use case 8.2 Search Inventory Type.	
	Step 2: The administrator clicks on the update icon		Step 3: The system retrieves the selected Inventory Type details from the InventoryType entity with following attributes:	
		Details to be read	Attributes in the table	Notes

			<table><tr><td>Inventory Type</td><td>inventory TypeName</td><td></td></tr></table>	Inventory Type	inventory TypeName																
	Inventory Type	inventory TypeName																			
			The system then displays the Inventory Type Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Update Inventory Type</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Inventory Type</td><td>Label</td><td>This control will be used as a label to indicate the Name attribute of the Inventory Type record</td></tr><tr><td>Inventory Type</td><td>Textbox</td><td>This control will be used by the administrator to update a name of the Inventory Type</td></tr><tr><td>Save</td><td>Button</td><td>This control will be used to confirm the update of the Inventory Type</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the edit of the Inventory Type</td></tr></table>	Control Name	Control Type	Notes	Update Inventory Type	Heading	Page main heading	Inventory Type	Label	This control will be used as a label to indicate the Name attribute of the Inventory Type record	Inventory Type	Textbox	This control will be used by the administrator to update a name of the Inventory Type	Save	Button	This control will be used to confirm the update of the Inventory Type	Cancel	Button	This control will be used to cancel the edit of the Inventory Type
	Control Name	Control Type	Notes																		
	Update Inventory Type	Heading	Page main heading																		
Inventory Type	Label	This control will be used as a label to indicate the Name attribute of the Inventory Type record																			
Inventory Type	Textbox	This control will be used by the administrator to update a name of the Inventory Type																			
Save	Button	This control will be used to confirm the update of the Inventory Type																			
Cancel	Button	This control will be used to cancel the edit of the Inventory Type																			
	Step 4: The system administrator enters the Inventory Type information and clicks on the save button [ALT]		Step 5: The system captures and validates the entered details against the Inventory Type table [ALT] <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Inventory Type</td><td>Inventory TypeName</td><td>Required, Unique,</td></tr></table>	Details to be read	Attributes in the table	Notes	Inventory Type	Inventory TypeName	Required, Unique,												
Details to be read	Attributes in the table	Notes																			
Inventory Type	Inventory TypeName	Required, Unique,																			
			Step 6: The system uses an SQL READ query to check if the inventory type already exists in the InventoryType table using the following attributes [ALT]																		

			Details to be updated		Attributes in the table	
			Name		InventoryTypeName	
			Step 7: The system displays a message box with the following controls:			
			Control Name	Control Type	Notes	
			Confirmation required	Label	This control is used to display the type of message box displayed	
			Confirm updated inventory type details?	Label	This control is used to request confirmation on the updated inventory type details	
			Confirm	Button	This control is used to confirm the updating of the inventory type details	
			Cancel	Button	This control is used to close the message box for the user to edit the Inventory Type details	
	Step 8: The system administrator clicks on the confirm button [ALT]		Step 9: The system saves the information into the Inventory Type table using an UPDATE TO SQL query:			
			Details to be saved	Attributes in the table	Notes	
			Inventory Type	Inventory TypeName	Required, Unique,	
			Step 10: The system redirects the system administrator to the Inventory Type screen and displays a bootstrap alert notification with the following controls:			
		Control Name	Control Name	Notes		
		"Inventory Type Successful update"	Label	This control is used to indicate a successful update		

	ALT Step 4: The administrator does not want to update the inventory type anymore - The administrator clicks on the cancel button - The use case is terminated											
	ALT Step 5: The details entered are not in the correct format or the required fields are empty - The system displays a validation error message box with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Inventory Type</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>Inventory Type Name is required</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Inventory Type textbox</td></tr></table> - Return to step 4	Control Name	Control Name	Notes	Inventory Type	Textbox	This control will be outline in red to show a validation error	Inventory Type Name is required	Label	This control will be used as a label to indicate a validation error in the Inventory Type textbox		
	Control Name	Control Name	Notes									
	Inventory Type	Textbox	This control will be outline in red to show a validation error									
	Inventory Type Name is required	Label	This control will be used as a label to indicate a validation error in the Inventory Type textbox									
ALT Step 6: The Inventory Type already exists on the system - The system displays an error message box with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Validation Error</td><td>Label</td><td>This control is used to display the type of message box displayed</td></tr><tr><td>Inventory Type details already exist</td><td>Label</td><td>This control is used to inform the user that the Inventory Type already exists</td></tr><tr><td>Okay</td><td>Button</td><td>This control is used to cancel the creation of the user role</td></tr></table> - The administrator clicks the OK button - The use case is terminated	Control Name	Control Name	Notes	Validation Error	Label	This control is used to display the type of message box displayed	Inventory Type details already exist	Label	This control is used to inform the user that the Inventory Type already exists	Okay	Button	This control is used to cancel the creation of the user role
Control Name	Control Name	Notes										
Validation Error	Label	This control is used to display the type of message box displayed										
Inventory Type details already exist	Label	This control is used to inform the user that the Inventory Type already exists										
Okay	Button	This control is used to cancel the creation of the user role										
ALT Step 8: The administrator clicks on the cancel button The use case is terminated												
CONCLUSION:	This use case concludes when an inventory type has been added on the system											
POST-CONDITION:	An inventory type has been updated in the InventoryType table in the database											
BUSINESS RULES	Only the administrator has access to this function.											
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None											
ASSUMPTIONS:	None											
OPEN ISSUES:	None											

Table 78 - 8.8 Delete Inventory Type

INVENTORY MANAGEMENT SUBSYSTEM				
AUTHORS (s): Ofhani Mungani			DATE: 16/05/2022	
VERSION: 1.0			LAST REVIEW DATE: 30/08/2022	
USE CASE NAME:	Delete Inventory Type		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒	
USE CASE ID:	8.8			
PRIORITY:	Medium			
SOURCE:	ByteXcom Requirements List			
PRIMARY BUSINESS ACTOR	• Administrator			
PRIMARY SYSTEM ACTOR	• None			
OTHER PARTICIPATING ACTORS:	• None			
OTHER INTERESTED STAKEHOLDERS:	• None			
DESCRIPTION:	This use case narrative describes the process in which the administrator wishes to delete an inventory type from the system. The usecase begins with the administrator searching for the required inventory type selected for removal. The Inventory Type will be removed from the system database.			
PRE-CONDITION:	• The administrator must be logged onto the system.			
TRIGGER:	The use case is triggered when the administrator wants to delete an Inventory Type existing in the InventoryType table.			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE		
		MANUAL ACTION	AUTOMATED ACTION	
	Step 1: The administrator wishes to delete an Inventory Type in the system.		Step 2: The system invokes use case 8.2 Search Inventory Type .	
	Step 3: The administrator clicks on the Delete icon		Step 4: The system retrieves the selected Inventory Type details from the Inventory Type entity with following attributes:	
			Details to be read	Attributes in the table
		Inventory Type	inventory TypeName	

		The system then displays the Inventory Type Details Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Delete Inventory Type Details</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Name</td><td>Label</td><td>This control will be used as a heading to display the Name for all Inventory Types</td></tr></table>	Control Name	Control Type	Notes	Delete Inventory Type Details	Heading	Page main heading	Name	Label	This control will be used as a heading to display the Name for all Inventory Types						
Control Name	Control Type	Notes															
Delete Inventory Type Details	Heading	Page main heading															
Name	Label	This control will be used as a heading to display the Name for all Inventory Types															
Step 5: The administrator clicks on the Delete button [ALT]		Step 6: The system displays a message box with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Confirmation required</td><td>Label</td><td>This control is used to display the type of message box displayed</td></tr><tr><td>Confirm updated user details?</td><td>Label</td><td>This control is used to request confirmation of the deletion of the Inventory Type</td></tr><tr><td>Confirm</td><td>Button</td><td>This control is used to confirm the deletion of the Inventory Type details</td></tr><tr><td>Cancel</td><td>Button</td><td>This control is used to close the message box deletion of the Inventory Type</td></tr></table>	Control Name	Control Type	Notes	Confirmation required	Label	This control is used to display the type of message box displayed	Confirm updated user details?	Label	This control is used to request confirmation of the deletion of the Inventory Type	Confirm	Button	This control is used to confirm the deletion of the Inventory Type details	Cancel	Button	This control is used to close the message box deletion of the Inventory Type
Control Name	Control Type	Notes															
Confirmation required	Label	This control is used to display the type of message box displayed															
Confirm updated user details?	Label	This control is used to request confirmation of the deletion of the Inventory Type															
Confirm	Button	This control is used to confirm the deletion of the Inventory Type details															
Cancel	Button	This control is used to close the message box deletion of the Inventory Type															
Step 7: The system administrator clicks on the confirm button [ALT]		Step 8: The system deletes the selected inventory type from the InventoryType entity in the system database. using a method in a .NET controller and an SQL DELETE query.															

			<table><tr><th>Details to be deleted</th><th>Attributes in the table</th></tr><tr><td>Id</td><td>Id</td></tr><tr><td>Inventory Type</td><td>inventory TypeName</td></tr></table>	Details to be deleted	Attributes in the table	Id	Id	Inventory Type	inventory TypeName
	Details to be deleted	Attributes in the table							
	Id	Id							
	Inventory Type	inventory TypeName							
			[ALT]						
			Step 9: The system redirects the system administrator to the Inventory Type screen and displays a bootstrap alert notification with the following controls:						
		<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"Inventory Type Successful deleted"</td><td>Label</td><td>This control is used to indicate a successful deletion</td></tr></table>	Control Name	Control Name	Notes	"Inventory Type Successful deleted"	Label	This control is used to indicate a successful deletion	
Control Name	Control Name	Notes							
"Inventory Type Successful deleted"	Label	This control is used to indicate a successful deletion							
	ALT Step 5: The administrator does not want to delete the Inventory Type anymore - The administrator clicks on the cancel button - The use case is terminated								
	ALT Step 7: The administrator does not want to confirm to delete the Inventory Type anymore Return to step 5								
	ALT Step 8: The Inventory Type has attached to an existing inventory item on the system. The system displays a bootstrap alert notification with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Unsuccessful deletion</td><td>Label</td><td>This control is used to indicate an unsuccessful deletion</td></tr></table>		Control Name	Control Name	Notes	Unsuccessful deletion	Label	This control is used to indicate an unsuccessful deletion	
Control Name	Control Name	Notes							
Unsuccessful deletion	Label	This control is used to indicate an unsuccessful deletion							
CONCLUSION:	This use case concludes when an inventory type has been deleted on the system								
POST-CONDITION:	The inventory type has been removed from the system database.								
BUSINESS RULES	Only the administrator has access to this function.								
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None								
ASSUMPTIONS:	None								
OPEN ISSUES:	None								

Table 79 - 8.9 Create Write-Off Reason

INVENTORY MANAGEMENT SUBSYSTEM										
AUTHORS (s): Thenjiwe Ntsonda			DATE: 18/05/2022							
VERSION: 1.0			LAST REVIEW DATE: 18/05/2022							
USE CASE NAME:	Create Write-Off Reason		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒							
USE CASE ID:	8.9									
PRIORITY:	High									
SOURCE:	ByteXcom Requirements List									
PRIMARY BUSINESS ACTOR	• Administrator									
PRIMARY SYSTEM ACTOR	• None									
OTHER PARTICIPATING ACTORS:	• None									
OTHER INTERESTED STAKEHOLDERS:	• None									
DESCRIPTION:	This use case describes the process in which the administrator wishes to add a new write-off reason to the system. The use case begins when the administrator adds all the pertinent information regarding the write-off reason and save it to the business database.									
PRE-CONDITION:	• The administrator must be logged onto the system.									
TRIGGER:	The use case is triggered when the administrator wants to add a new write-off reason which the administrator									
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE								
		MANUAL ACTION	AUTOMATED ACTION							
	Step 1: The administrator navigates to the Write-Off Reason Screen under the Write-off reason tab on the side bar		Step 2: The system retrieves write-off reason details from the Write-Off Reason entity with the following attributes:							
			<table><tr><td>Details to be read</td><td>Attributes in the table</td><td>Notes</td></tr><tr><td>Name</td><td>Name</td><td></td></tr></table>		Details to be read	Attributes in the table	Notes	Name	Name	
			Details to be read	Attributes in the table	Notes					
			Name	Name						
The system then displays the Write-Off Reason Screen with the following controls:										
<table><tr><td>Control Name</td><td>Control Type</td><td>Notes</td></tr><tr><td>Write-Off Reason List</td><td>Heading</td><td>Page main heading</td></tr></table>		Control Name	Control Type	Notes	Write-Off Reason List	Heading	Page main heading			
Control Name	Control Type	Notes								
Write-Off Reason List	Heading	Page main heading								

			Create Write-Off Reason	Button	When clicked, this control will direct the user to the Add Write-Off Reason page
			Show “” entries	Label and drop down	This control will be used to set the number of entries the administrator wants to display on a page
			Search bar	Textbox	This control will be used when the user inserts the search parameters
			Results table	Table	This control will be used to display all the write-off reasons that exist on the system
			Name	Table Heading	This control will be used as a heading to display the Name for all write-off reasons
			Showing “” to “” of “” entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database
			“Previous” and “Next”	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on
			Step 3: The system administrator clicks on the Create Write-Off Reason button		
			Step 4: The system displays the Create Write-Off Reason Screen with the following controls:		
			Control Name	Control Type	Notes

			Create Write-off reason	Heading	Page main heading
			Name	Label	This control will be used as a label to indicate the Name attribute of the write-off reason record
			Name	Textbox	This control will be used by the administrator to insert a name of the write-off reason
			Create	Button	This control will be used to confirm the addition of the write-off reason
			Cancel	Button	This control will be used to cancel the addition of the write-off reason
	Step 5: The system administrator clicks on the Create Write-Off Reason button		Step 6: The system captures and validates the entered details against the WriteOffReason table [ALT]		
			Details to be read	Attributes in the table	Notes
			Name	Name	Required, Unique,
	Step 7: The system administrator enters the details and clicks on the create button [ALT]		Step 8: The system saves the information into the WriteOffReason table using an INSERT SQL query:		
			Details to be saved	Attributes in the table	Notes
			ID	Write-off reasonID	A unique value generated by the system and saved to the database.
			Name	Name	Required, Unique,
			Step 9: The system redirects the system administrator to the Write-Off Reason screen and displays a bootstrap alert notification with the following controls:		
			Control Name	Control Name	Notes

			"Write-Off Reason successfully created"	Label	This control is used to indicate a successful creation												
ALTERNATE COURSES:	ALT Step 7: The administrator does not wish to add a write-off reason anymore - The administrator clicks on the cancel button - The use case is terminated																
	ALT Step 6A: The details entered are not in the correct format or the required fields are empty The system displays a validation error message box with the following controls																
	<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Name</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>Name is required</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Name textbox</td></tr></table>					Control Name	Control Name	Notes	Name	Textbox	This control will be outline in red to show a validation error	Name is required	Label	This control will be used as a label to indicate a validation error in the Name textbox			
	Control Name	Control Name	Notes														
	Name	Textbox	This control will be outline in red to show a validation error														
	Name is required	Label	This control will be used as a label to indicate a validation error in the Name textbox														
	Return to step 3																
	ALT Step 6B: The write-off reason already exists on the system The system displays an error message box with the following controls																
	<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Validation Error</td><td>Label</td><td>This control is used to display the type of message box displayed</td></tr><tr><td>Write-off reason already exists!</td><td>Label</td><td>This control is used to inform the user that the user already exists</td></tr><tr><td>Okay</td><td>Button</td><td>This control is used to cancel the creation of the user role</td></tr></table>					Control Name	Control Name	Notes	Validation Error	Label	This control is used to display the type of message box displayed	Write-off reason already exists!	Label	This control is used to inform the user that the user already exists	Okay	Button	This control is used to cancel the creation of the user role
	Control Name	Control Name	Notes														
Validation Error	Label	This control is used to display the type of message box displayed															
Write-off reason already exists!	Label	This control is used to inform the user that the user already exists															
Okay	Button	This control is used to cancel the creation of the user role															
- The administrator clicks the OK button - The use case is terminated																	
CONCLUSION:	This use case concludes when a write-off reason has been added on the system																
POST-CONDITION:	A new user has been added to the Write-off reason table in the database																
BUSINESS RULES	Only the administrator has access to this function.																
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	The system is offline, and the system administrator will not be able to add a new write-off reason.																
ASSUMPTIONS:	None																
OPEN ISSUES:	None																

Table 80 - 8.10 Search Write-Off Reason

INVENTORY MANAGEMENT SUBSYSTEM										
AUTHORS (s): Thenjiwe Ntsonda			DATE: 18/05/2022							
VERSION: 1.0			LAST REVIEW DATE: 18/05/2022							
USE CASE NAME:	Search Write-Off Reason		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒							
USE CASE ID:	8.10									
PRIORITY:	Medium									
SOURCE:	ByteXcom Requirements List									
PRIMARY BUSINESS ACTOR	• Administrator									
PRIMARY SYSTEM ACTOR	• None									
OTHER PARTICIPATING ACTORS:	• None									
OTHER INTERESTED STAKEHOLDERS:	• None									
DESCRIPTION:	This use case describes the process in which the administrator searches for product categories on the system. The system responds by displaying the information relating to the search query.									
PRE-CONDITION:	• The administrator must be logged onto the system.									
TRIGGER:	This use case is called when the administrator wants to search for a write-off reason on the system.									
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE								
		MANUAL ACTION	AUTOMATED ACTION							
	Step 1: The administrator wishes to search for a write-off reason on the system.									
	Step 2: The administrator navigates to the Write-Off Reason Screen under the Inventory tab on the navigation bar		Step 3: The system retrieves the existing items in the Write-Off Reason table with the following attributes: <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Name</td><td>Name</td><td></td></tr></table> The system then displays the Write-Off Reason Screen with the following controls:		Details to be read	Attributes in the table	Notes	Name	Name	
	Details to be read	Attributes in the table	Notes							
Name	Name									

			<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Write-Off Reason List</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Create Write-Off Reason</td><td>Button</td><td>When clicked, this control will direct the user to the Add Write-Off Reason page</td></tr><tr><td>Show "" entries</td><td>Label and drop down</td><td>This control will be used to set the number of entries the administrator wants to display on a page</td></tr><tr><td>Search bar</td><td>Textbox</td><td>This control will be used when the user inserts the search parameters</td></tr><tr><td>Results table</td><td>Table</td><td>This control will be used to display all the write-off reasons that exist on the system</td></tr><tr><td>Name</td><td>Table Heading</td><td>This control will be used as a heading to display the Name for all write-off reasons</td></tr><tr><td>Showing "" to "" of "" entries</td><td>Label</td><td>This control will be used to display the number of entries displayed per page as well as the total number of entries in the database</td></tr><tr><td>"Previous" and "Next"</td><td>Label and button</td><td>This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on</td></tr></table>	Control Name	Control Type	Notes	Write-Off Reason List	Heading	Page main heading	Create Write-Off Reason	Button	When clicked, this control will direct the user to the Add Write-Off Reason page	Show "" entries	Label and drop down	This control will be used to set the number of entries the administrator wants to display on a page	Search bar	Textbox	This control will be used when the user inserts the search parameters	Results table	Table	This control will be used to display all the write-off reasons that exist on the system	Name	Table Heading	This control will be used as a heading to display the Name for all write-off reasons	Showing "" to "" of "" entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database	"Previous" and "Next"	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on
			Control Name	Control Type	Notes																									
			Write-Off Reason List	Heading	Page main heading																									
			Create Write-Off Reason	Button	When clicked, this control will direct the user to the Add Write-Off Reason page																									
			Show "" entries	Label and drop down	This control will be used to set the number of entries the administrator wants to display on a page																									
			Search bar	Textbox	This control will be used when the user inserts the search parameters																									
			Results table	Table	This control will be used to display all the write-off reasons that exist on the system																									
			Name	Table Heading	This control will be used as a heading to display the Name for all write-off reasons																									
			Showing "" to "" of "" entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database																									
"Previous" and "Next"	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on																												
Step 4: The administrator enters	Step 5: The system searches the Write-Off Reason table and displays																													

	their search query into the search textbox		the write-off reason(s) matching the search query in the Write-Off Reason screen [ALT]
	ALT Step 5: The search query entered does not match any records existing in the Write-off reason entity. The system displays a “No Records Found” message in the Write-Off Reason Screen and on the results table.		
CONCLUSION:	This use case concludes when the required write-off reason is displayed on the screen.		
POST-CONDITION:	The administrator selects the required write-off reason from the displayed results.		
BUSINESS RULES	Only the administrator has access to this function.		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

Table 81 - 8.11 Update Write-Off Reason

INVENTORY MANAGEMENT SUBSYSTEM				
AUTHORS (s): Thenjiwe Ntsonda			DATE: 18/05/2022	
VERSION: 1.0			LAST REVIEW DATE: 18/05/2022	
USE CASE NAME:	Update Write-Off Reason		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒	
USE CASE ID:	8.11			
PRIORITY:	Medium			
SOURCE:	ByteXcom Requirements List			
PRIMARY BUSINESS ACTOR	• Administrator			
PRIMARY SYSTEM ACTOR	• None			
OTHER PARTICIPATING ACTORS:	• None			
OTHER INTERESTED STAKEHOLDERS:	• None			
DESCRIPTION:	This use case describes the process in which the administrator wishes to update a write-off reason on the system. The use case begins when the administrator updates all the pertinent information regarding the write-off reason and save it to the business database.			
PRE-CONDITION:	• The administrator must be logged onto the system.			
TRIGGER:	The use case is triggered when the administrator wants to update a write-off reason.			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE		
		MANUAL ACTION	AUTOMATED ACTION	
	Step 1: The administrator wishes to update a write-off reason in the system.		Step 2: The system invokes use case 8.10 Search Write-off Reason	
	Step 2: The administrator clicks on the update button		Step 3: The system retrieves the selected write-off reason details from the Write-Off Reason entity with the following attributes:	

		The system then displays the Update Write-Off Reason Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Update Write-off reason</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Name</td><td>Label</td><td>This control will be used as a label to indicate the Name attribute of the write-off reason record</td></tr><tr><td>Name</td><td>Textbox</td><td>This control will be used by the administrator to update a name of the write-off reason</td></tr><tr><td>Update</td><td>Button</td><td>This control will be used to confirm the update of the write-off reason</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the edit of the write-off reason</td></tr></table>	Control Name	Control Type	Notes	Update Write-off reason	Heading	Page main heading	Name	Label	This control will be used as a label to indicate the Name attribute of the write-off reason record	Name	Textbox	This control will be used by the administrator to update a name of the write-off reason	Update	Button	This control will be used to confirm the update of the write-off reason	Cancel	Button	This control will be used to cancel the edit of the write-off reason
Control Name	Control Type	Notes																		
Update Write-off reason	Heading	Page main heading																		
Name	Label	This control will be used as a label to indicate the Name attribute of the write-off reason record																		
Name	Textbox	This control will be used by the administrator to update a name of the write-off reason																		
Update	Button	This control will be used to confirm the update of the write-off reason																		
Cancel	Button	This control will be used to cancel the edit of the write-off reason																		
	Step 4: The system administrator enters the write-off reason information and clicks on the update button [ALT]	Step 5: The system displays a modal dialog with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Confirmation required</td><td>Label</td><td>This control is used to display the type of message box displayed</td></tr><tr><td>Confirm updated write-off reason details?</td><td>Label</td><td>This control is used to request confirmation on the updated write-off reason details</td></tr><tr><td>Confirm</td><td>Button</td><td>This control is used to confirm the updating of</td></tr></table>	Control Name	Control Type	Notes	Confirmation required	Label	This control is used to display the type of message box displayed	Confirm updated write-off reason details?	Label	This control is used to request confirmation on the updated write-off reason details	Confirm	Button	This control is used to confirm the updating of						
Control Name	Control Type	Notes																		
Confirmation required	Label	This control is used to display the type of message box displayed																		
Confirm updated write-off reason details?	Label	This control is used to request confirmation on the updated write-off reason details																		
Confirm	Button	This control is used to confirm the updating of																		

					the write-off reason details
			Cancel	Button	This control is used to close the message box for the user to edit the write-off reason details
	Step 6: The system administrator clicks on the confirm button [ALT]		Step 7: The system captures and validates the entered details against the Write-Off Reason table [ALT]		

CONCLUSION:	This use case concludes when a write-off reason has been added on the system
POST-CONDITION:	A category has been updated in the Write-off reason table in the database
BUSINESS RULES	• Only the administrator has access to this function.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 82 - 8.12 Delete Write-Off Reason

INVENTORY MANAGEMENT SUBSYSTEM				
AUTHORS (s): Thenjiwe Ntsonda			DATE: 18/05/2022	
VERSION: 1.0			LAST REVIEW DATE: 18/05/2022	
USE CASE NAME:	Delete Write-off reason		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒	
USE CASE ID:	8.12			
PRIORITY:	Medium			
SOURCE:	ByteXcom Requirements List			
PRIMARY BUSINESS ACTOR	• Administrator			
PRIMARY SYSTEM ACTOR	• None			
OTHER PARTICIPATING ACTORS:	• None			
OTHER INTERESTED STAKEHOLDERS:	• None			
DESCRIPTION:	This use case narrative describes the process in which the administrator wishes to delete a write-off reason from the system. The use case begins with the administrator searching for the required write-off reason selected for removal. The write-off reason will be removed from the system database.			
PRE-CONDITION:	• The administrator must be logged onto the system.			
TRIGGER:	The use case is triggered when the administrator wants to delete a write-off reason existing in the write-off reason catalogue.			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE		
		MANUAL ACTION	AUTOMATED ACTION	
	Step 1: The administrator wishes to delete a write-off reason in the system.		Step 2: The system invokes use case 8.10 Search Write-Off Reason	
	Step 3: The administrator clicks on the Delete button		Step 4: The system retrieves the selected Write-Off Reason details from the Write-Off Reason entity with the following attributes:	
			Details to be read	Attributes in the table

		<table><tr><td>Name</td><td>Name</td><td></td></tr></table> The system then displays the Write-Off Reason Details Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Delete Write-Off Reason</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Name</td><td>Label</td><td>This control will be used as a heading to display the Name for the selected write-off reason</td></tr><tr><td>Name</td><td>Label</td><td>This control is used to display the name of the selected write-off reason</td></tr></table>	Name	Name		Control Name	Control Type	Notes	Delete Write-Off Reason	Heading	Page main heading	Name	Label	This control will be used as a heading to display the Name for the selected write-off reason	Name	Label	This control is used to display the name of the selected write-off reason
	Name	Name															
	Control Name	Control Type	Notes														
	Delete Write-Off Reason	Heading	Page main heading														
	Name	Label	This control will be used as a heading to display the Name for the selected write-off reason														
	Name	Label	This control is used to display the name of the selected write-off reason														
		Step 5: The system displays a modal dialog with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Confirmation required</td><td>Label</td><td>This control is used to display the type of message box displayed</td></tr><tr><td>Are you sure you want to delete this Write-Off Reason?</td><td>Label</td><td>This control is used to request confirmation of the deletion of the write-off reason</td></tr><tr><td>Confirm</td><td>Button</td><td>This control is used to confirm the deletion of the write-off reason details</td></tr><tr><td>Cancel</td><td>Button</td><td>This control is used to close the message box deletion of the write-off reason</td></tr></table>	Control Name	Control Type	Notes	Confirmation required	Label	This control is used to display the type of message box displayed	Are you sure you want to delete this Write-Off Reason?	Label	This control is used to request confirmation of the deletion of the write-off reason	Confirm	Button	This control is used to confirm the deletion of the write-off reason details	Cancel	Button	This control is used to close the message box deletion of the write-off reason
	Control Name	Control Type	Notes														
	Confirmation required	Label	This control is used to display the type of message box displayed														
	Are you sure you want to delete this Write-Off Reason?	Label	This control is used to request confirmation of the deletion of the write-off reason														
Confirm	Button	This control is used to confirm the deletion of the write-off reason details															
Cancel	Button	This control is used to close the message box deletion of the write-off reason															
Step 5: The system administrator clicks on the confirm button [ALT]	Step 6: The system deletes the selected write-off reason from the Write-off Reason entity in the system database. [ALT]																

			Step 7: The system redirects the system administrator to the Write-Off Reason screen and displays a bootstrap alert notification with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"Write-Off Reason deleted successfully"</td><td>Label</td><td>This control is used to indicate a successful deletion</td></tr></table>	Control Name	Control Name	Notes	"Write-Off Reason deleted successfully"	Label	This control is used to indicate a successful deletion
Control Name	Control Name	Notes							
"Write-Off Reason deleted successfully"	Label	This control is used to indicate a successful deletion							
ALTERNATE COURSES:	ALT Step 5: The administrator does not wish to delete the write-off reason anymore - The administrator clicks on the cancel button - The use case is terminated								
	ALT Step 5: The write-off reason has attached to an existing record of a written off product/inventory item. The system displays an error message box with the following controls:<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Unsuccessful deletion</td><td>Label</td><td>This control is used to indicate an unsuccessful deletion</td></tr></table>			Control Name	Control Name	Notes	Unsuccessful deletion	Label	This control is used to indicate an unsuccessful deletion
Control Name	Control Name	Notes							
Unsuccessful deletion	Label	This control is used to indicate an unsuccessful deletion							
CONCLUSION:	This use case concludes when a write-off reason has been deleted on the system								
POST-CONDITION:	The write-off reason has been removed from the system database.								
BUSINESS RULES	Only the administrator has access to this function.								
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None								
ASSUMPTIONS:	None								
OPEN ISSUES:	None								

Table 83 - 8.13 Write-Off Inventory

INVENTORY MANAGEMENT SUBSYSTEM													
AUTHORS (s): Ofhani Mungani			DATE: 06/06/2022										
VERSION: 1.0			LAST REVIEW DATE: 06/06/2022										
USE CASE NAME:	Write-Off Inventory		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒										
USE CASE ID:	8.13												
PRIORITY:	High												
SOURCE:	ByteXcom Requirements List												
PRIMARY BUSINESS ACTOR	• Administrator												
PRIMARY SYSTEM ACTOR	• None												
OTHER PARTICIPATING ACTORS:	• None												
OTHER INTERESTED STAKEHOLDERS:	• None												
DESCRIPTION:	This use case describes the process in which the administrator wishes to write off an Inventory Item out of the system. The administrator will add all of the pertinent information regarding the Inventory Item write-off and save it to the business database.												
PRE-CONDITION:	• The administrator must be logged onto the system.												
TRIGGER:	The use case is triggered when the administrator wants to write-off an Inventory Item on the system.												
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE											
		MANUAL ACTION	AUTOMATED ACTION										
	Step 1: The administrator navigates to the Inventory Item Screen under the Inventory Item tab on the side bar		Step 2: The system retrieves Inventory Item details: From the InventoryItem entity with the following attributes										
			<table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Inventory Item</td><td>inventory ItemName</td><td></td></tr><tr><td>Quantity on Hand</td><td>QuantityOn Hand</td><td></td></tr></table>		Details to be read	Attributes in the table	Notes	Inventory Item	inventory ItemName		Quantity on Hand	QuantityOn Hand	
			Details to be read	Attributes in the table	Notes								
			Inventory Item	inventory ItemName									
Quantity on Hand	QuantityOn Hand												
From the InventoryType entity with the following attributes:													

Details to be read	Attributes in the table	Notes
Inventory Type	inventory TypeName	
The system then displays the Inventory Item Screen with the following controls:		
Control Name	Control Type	Notes
Inventory Items List	Heading	Page main heading
Create Inventory Item	Button	When clicked, this control will direct the user to the Add Inventory Item page
Write-Off Inventory Item	Button	When clicked, this control will direct the user to the Write-Off Inventory Item page
Search bar	Textbox	This control will be used when the user inserts the search parameters
Results table	Table	This control will be used to display all the Inventory Items that exist on the system
Inventory Item	Table Heading	This control will be used as a heading to display the name for all Inventory Items
Inventory Type	Table Heading	This control will be used as a heading to display the name for all Inventory types
Quantity on Hand	Table Heading	This control will be used as a heading to display the value of all Inventory Item's quantity on hand

			Show “” entries	Label and drop down	This control will be used to set the number of entries the administrator wants to display on a page																					
			Showing “” to “” of “” entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database																					
			“Previous” and “Next”	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on																					
	Step 3: The system administrator clicks on the Write-Off Inventory Item button	Step 4: The system retrieves the values with following attributes From the InventoryItem entity: <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Inventory Item</td><td>inventory ItemName</td><td></td></tr></table> From the WriteOffReason entity: <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>WriteOffReason</td><td>Name</td><td></td></tr></table> The system then displays the Write-off Inventory Item Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Write-off Inventory Item</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Inventory Item</td><td>Label</td><td>This control will be used as a label to indicate the InventoryItem Name</td></tr></table>				Details to be read	Attributes in the table	Notes	Inventory Item	inventory ItemName		Details to be read	Attributes in the table	Notes	WriteOffReason	Name		Control Name	Control Type	Notes	Write-off Inventory Item	Heading	Page main heading	Inventory Item	Label	This control will be used as a label to indicate the InventoryItem Name
Details to be read	Attributes in the table	Notes																								
Inventory Item	inventory ItemName																									
Details to be read	Attributes in the table	Notes																								
WriteOffReason	Name																									
Control Name	Control Type	Notes																								
Write-off Inventory Item	Heading	Page main heading																								
Inventory Item	Label	This control will be used as a label to indicate the InventoryItem Name																								

					attribute of the Inventory Item record
			Inventory Item	Dropdown	This control will be used by the administrator to select an inventory type for the inventory item
			Write-off Reason	Label	This control will be used by the administrator to indicate an write off reason for the inventory item
			Write-off Reason	Dropdown	This control will be used by the administrator to select an write off reason for the inventory item
			Write-off Quantity	Label	This control will be used as a label to indicate the quantity for the write off inventory item record
			Write -off Quantity	Textbox	This control will be used to store the quantity for the write off inventory item.
			Create	Button	This control will be used to confirm the addition of the Inventory Item
			Cancel	Button	This control will be used to cancel the addition of the Inventory Item

	Step 5: The system administrator enters the details and clicks on the write-off button [ALT]		Step 6: The system captures and validates the entered details against the WriteoffInventoryItem table [ALT] <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Write-off Quantity</td><td>WriteOffQuantity</td><td>Required</td></tr><tr><td>Inventory item</td><td>InventoryItemName</td><td>Required</td></tr><tr><td>Write of Reason</td><td>Name</td><td>Required</td></tr></table>	Details to be read	Attributes in the table	Notes	Write-off Quantity	WriteOffQuantity	Required	Inventory item	InventoryItemName	Required	Write of Reason	Name	Required												
	Details to be read	Attributes in the table	Notes																								
	Write-off Quantity	WriteOffQuantity	Required																								
	Inventory item	InventoryItemName	Required																								
Write of Reason	Name	Required																									
			Step 7: The system prompts the administrator to confirm the write-off details.																								
	Step 8: The administrator confirms the information provided. [ALT]		Step 9: The system saves the information into the WriteInventoryItem table using an INSERT SQL query: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>ID</td><td>Id</td><td>A unique value generated by the system and saved to the database.</td></tr><tr><td>Inventory Item</td><td>InventoryType</td><td>Required</td></tr><tr><td>WriteOffReason</td><td>WriteoffReasonId</td><td>Required</td></tr><tr><td>WriteoffQuantity</td><td>WriteoffQuantity</td><td>Required</td></tr><tr><td>WriteOffDate</td><td>WriteOffDate</td><td>The is auto generated by the system and saved to the database</td></tr></table> The system decreases the QuantityOnHand attribute on the InventoryItem table using UPDATE SQL query: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Quantity on Hand</td><td>QuantityOnHand</td><td>Required</td></tr></table> [ALT]	Details to be saved	Attributes in the table	Notes	ID	Id	A unique value generated by the system and saved to the database.	Inventory Item	InventoryType	Required	WriteOffReason	WriteoffReasonId	Required	WriteoffQuantity	WriteoffQuantity	Required	WriteOffDate	WriteOffDate	The is auto generated by the system and saved to the database	Details to be saved	Attributes in the table	Notes	Quantity on Hand	QuantityOnHand	Required
			Details to be saved	Attributes in the table	Notes																						
			ID	Id	A unique value generated by the system and saved to the database.																						
			Inventory Item	InventoryType	Required																						
WriteOffReason	WriteoffReasonId	Required																									
WriteoffQuantity	WriteoffQuantity	Required																									
WriteOffDate	WriteOffDate	The is auto generated by the system and saved to the database																									
Details to be saved	Attributes in the table	Notes																									
Quantity on Hand	QuantityOnHand	Required																									
		Step 10: The system directs the system administrator to the Inventory Item Write-off Records screen and																									

displays a bootstrap alert notification with the following controls:

Bootstrap notification:

Control Name	Control Name	Notes
"Inventory Item written off Successful"	Label	This control is used to indicate a successful creation

Inventory Item Write-off Records screen:

Control Name	Control Type	Notes
Inventory Item Write-off I Records	Heading	Page main heading
Back to Inventory Item	Button	When clicked, this control will direct the user to the Inventory Item List page
Search bar	Textbox	This control will be used when the user inserts the search parameters
Results table	Table	This control will be used to display all the Inventory Items that exist on the system
Inventory Item	Table Heading	This control will be used as a heading to display the name for all Inventory Items
Inventory Type	Table Heading	This control will be used as a heading to display the name for all Inventory types
Write off Quantity	Table Heading	This control will be used as a heading to display the value of all Inventory Item's written off quantity

			Write off date	Table Heading	This control will be used as a heading to display the value of all written off Inventory Items date
			Show “” entries	Label and drop down	This control will be used to set the number of entries the administrator wants to display on a page
			Showing “” to “” of “” entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database
			“Previous” and “Next”	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on
ALTERNATE COURSES:	ALT Step 5: The administrator does not want to write-off an Inventory Item anymore - The administrator clicks on the cancel button - The use case is terminated				
	ALT Step 6: The details entered are not in the correct format or the required fields are empty The system displays a validation error message box with the following controls				
	Control Name	Control Name	Notes		
	Write-off Quantity	Textbox	This control will be outline in red to show a validation error		
	Write-off Quantity is required	Label	This control will be used as a label to indicate a validation error in the Write off Quantity textbox		
Inventory Item	Dropdown list	This control will be outline in red to show a validation error			

	Inventory Item is required	Label	This control will be used as a label to indicate a validation error in the Inventory Item dropdown list
	Write-Off Reason	Dropdown list	This control will be outline in red to show a validation error
	Write-off Reason is required	Label	This control will be used as a label to indicate a validation error in the Write-off Reason dropdown list
	Return to step 3		
	ALT Step 8: The administrator does not confirm the addition of the new product. - The administrator clicks the cancel button - Return to Step 3		
	ALT Step 9: QuantityOnHand is less than the writeOffQuantity The system displays an error message box with the following controls		
	Control Name	Control Name	Notes
	Validation Error	Label	This control is used to display the type of message box displayed
	Write-off quantity is more than inventory quantity on hand	Label	This control is used to inform the administrator that the Write-off quantity is more than inventory quantity on hand
	Okay	Button	This control is used to cancel the creation of the user role
- The administrator clicks the OK button - Return to step 3			
CONCLUSION:	This use case concludes when an Inventory Item is written off.		
POST-CONDITION:	The write-off Inventory Item details are added to the database.		
BUSINESS RULES	Only the administrator has access to this function.		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	The system is offline, and the system administrator will not be able to write off an Inventory Item.		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

Table 84 - 8.14 Write-Off Product

INVENTORY MANAGEMENT SUBSYSTEM																
AUTHORS (s): Ofhani Mungani			DATE: 06/06/2022													
VERSION: 1.0			LAST REVIEW DATE: 06/06/2022													
USE CASE NAME:	Write-Off Product		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒													
USE CASE ID:	8.14															
PRIORITY:	High															
SOURCE:	ByteXcom Requirements List															
PRIMARY BUSINESS ACTOR	Administrator															
PRIMARY SYSTEM ACTOR	None															
OTHER PARTICIPATING ACTORS:	None															
OTHER INTERESTED STAKEHOLDERS:	None															
DESCRIPTION:	This use case describes the process in which the administrator wishes to write off a product out of the system. The administrator will add all of the pertinent information regarding the product write-off and save it to the business database.															
PRE-CONDITION:	The administrator must be logged onto the system.															
TRIGGER:	The use case is triggered when the administrator wants to write-off a product on the system.															
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE														
		MANUAL ACTION	AUTOMATED ACTION													
	Step 1: The administrator navigates to the Product Screen under the Products tab on the side bar		Step 2: The system retrieves products from the Product entity with their associated Product Categories and Product Brands and displays the Product Screen with the following controls:													
			<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Products</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Add Product</td><td>Button</td><td>When clicked, this control will direct the user to the Add product page</td></tr><tr><td>Search bar</td><td>Textbox</td><td>This control will be used when the user inserts the</td></tr></table>		Control Name	Control Type	Notes	Products	Heading	Page main heading	Add Product	Button	When clicked, this control will direct the user to the Add product page	Search bar	Textbox	This control will be used when the user inserts the
			Control Name	Control Type	Notes											
Products	Heading	Page main heading														
Add Product	Button	When clicked, this control will direct the user to the Add product page														
Search bar	Textbox	This control will be used when the user inserts the														

					search parameters
	Search icon	Button			This control will be used to activate the search
	Results table	Table			This control will be used to display all the products that exist on the system
	Name	Label			This control will be used as a heading to display the Name for all products
	Description	Label			This control will be used as a heading to display the Description for all products
	Customer Price	Label			This control will be used as a heading to display the customer price of the product
	Reseller Price	Label			This control will be used as a heading to display the reseller price of the product
	Product Category	Label			This control will be used as a heading to display the product category of the product.
	Product Brand	Label			This control will be used as a heading to display the product brand of the product.
	Page count	Label			This control will be used as a label to display the number of pages that the results make up and to indicate on which page the user is currently on
	Back and forward	Button			This control is used to navigate through the

				pages used to display the results from the table	
	Step 3: The system administrator clicks on the Write-Off Product button		Step 4: The system retrieves the values with following attributes		
			From the Product entity:		
			Details to be read	Attributes in the table	Notes
			Product	Name	
			From the WriteOffReason entity:		
			Details to be read	Attributes in the table	Notes
			WriteOffReason	Name	
			The system then displays the Write-off Product Screen with the following controls:		
			Control Name	Control Type	Notes
			Write-off Product	Heading	Page main heading
Product	Label	This control will be used as a label to indicate the product name attribute			
Product	Dropdown	This control will be used by the administrator to select a name of the product			
Write-off Reason	Label	This control will be used by the administrator to indicate an write off reason for the inventory item			
Write-off Reason	Dropdown	This control will be used by the administrator to select an write off reason for the inventory item			
Write-off Quantity	Label	This control will be used as a label to indicate the quantity for the write off			

					inventory item record																		
			Write -off Quantity	Textbox	This control will be used to store the quantity for the write off inventory item.																		
			Create	Button	This control will be used to confirm the addition of the Inventory Item																		
			Cancel	Button	This control will be used to cancel the addition of the Inventory Item																		
	Step 5: The system administrator enters the details and clicks on the write-off button [ALT]		Step 6: The system captures and validates the entered details against the WriteoffProduct table [ALT]																				
			<table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Write-off Quantity</td><td>WriteOffQuantity</td><td>Required</td></tr><tr><td>Product</td><td>Name</td><td>Required</td></tr><tr><td>Write of Reason</td><td>Name</td><td>Required</td></tr></table>	Details to be read	Attributes in the table	Notes	Write-off Quantity	WriteOffQuantity	Required	Product	Name	Required	Write of Reason	Name	Required								
Details to be read	Attributes in the table	Notes																					
Write-off Quantity	WriteOffQuantity	Required																					
Product	Name	Required																					
Write of Reason	Name	Required																					
			Step 7: The system prompts the administrator to confirm the write-off details.																				
	Step 8: The administrator confirms the information provided. [ALT]		Step 9: The system saves the information into the WriteProduct table using an INSERT SQL query:																				
			<table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>ID</td><td>Id</td><td>A unique value generated by the system and saved to the database.</td></tr><tr><td>Product</td><td>ProductId</td><td>Required</td></tr><tr><td>WriteOffReason</td><td>WriteoffReasonId</td><td>Required</td></tr><tr><td>WriteoffQuantity</td><td>WiteoffQuantity</td><td>Required</td></tr><tr><td>WriteOffDate</td><td>WriteOffDate</td><td>The is auto generated by the system and saved to the database</td></tr></table>	Details to be saved	Attributes in the table	Notes	ID	Id	A unique value generated by the system and saved to the database.	Product	ProductId	Required	WriteOffReason	WriteoffReasonId	Required	WriteoffQuantity	WiteoffQuantity	Required	WriteOffDate	WriteOffDate	The is auto generated by the system and saved to the database		
Details to be saved	Attributes in the table	Notes																					
ID	Id	A unique value generated by the system and saved to the database.																					
Product	ProductId	Required																					
WriteOffReason	WriteoffReasonId	Required																					
WriteoffQuantity	WiteoffQuantity	Required																					
WriteOffDate	WriteOffDate	The is auto generated by the system and saved to the database																					

		The system decreases the QuantityOnHand attribute on the Product table using UPDATE SQL query: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Quantity on Hand</td><td>QuantityonH nad</td><td>Required</td></tr></table> [ALT]	Details to be saved	Attributes in the table	Notes	Quantity on Hand	QuantityonH nad	Required																		
Details to be saved	Attributes in the table	Notes																								
Quantity on Hand	QuantityonH nad	Required																								
		Step 10: The system directs the system administrator to the Product Write-off Records screen and displays a bootstrap alert notification with the following controls: Bootstrap notification: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>“Product written off Successfully ”</td><td>Label</td><td>This control is used to indicate a successful creation</td></tr></table> Product Write-off Records screen: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Product Write-off Records</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Back to Products</td><td>Button</td><td>When clicked, this control will direct the user to the Product List page</td></tr><tr><td>Search bar</td><td>Textbox</td><td>This control will be used when the user inserts the search parameters</td></tr><tr><td>Results table</td><td>Table</td><td>This control will be used to display all the Inventory Items that exist on the system</td></tr><tr><td>Product</td><td>Table Heading</td><td>This control will be used as a heading to display the name for all products</td></tr></table>	Control Name	Control Name	Notes	“Product written off Successfully ”	Label	This control is used to indicate a successful creation	Control Name	Control Type	Notes	Product Write-off Records	Heading	Page main heading	Back to Products	Button	When clicked, this control will direct the user to the Product List page	Search bar	Textbox	This control will be used when the user inserts the search parameters	Results table	Table	This control will be used to display all the Inventory Items that exist on the system	Product	Table Heading	This control will be used as a heading to display the name for all products
Control Name	Control Name	Notes																								
“Product written off Successfully ”	Label	This control is used to indicate a successful creation																								
Control Name	Control Type	Notes																								
Product Write-off Records	Heading	Page main heading																								
Back to Products	Button	When clicked, this control will direct the user to the Product List page																								
Search bar	Textbox	This control will be used when the user inserts the search parameters																								
Results table	Table	This control will be used to display all the Inventory Items that exist on the system																								
Product	Table Heading	This control will be used as a heading to display the name for all products																								

			Description	Table Heading	This control will be used as a heading to display the Description for all products
			Write off Quantity	Table Heading	This control will be used as a heading to display the value of all written off quantity
			Write off date	Table Heading	This control will be used as a heading to display the value of all written off products date
			Show "" entries	Label and drop down	This control will be used to set the number of entries the administrator wants to display on a page
			Showing "" to "" of "" entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database
			"Previous" and "Next"	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on
ALTERNATE COURSES:	ALT Step 5: The administrator does not want to write-off a product anymore - The administrator clicks on the cancel button - The use case is terminated				
	ALT Step 6: The details entered are not in the correct format or the required fields are empty The system displays a validation error message box with the following controls				

	<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Write-off Quantity</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>Write-off Quantity is required</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Write off Quantity textbox</td></tr><tr><td>Product</td><td>Dropdown list</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>Product Item is required</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the product dropdown list</td></tr><tr><td>Write-Off Reason</td><td>Dropdown list</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>Write-off Reason is required</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Write-off Reason dropdown list</td></tr></table>	Control Name	Control Name	Notes	Write-off Quantity	Textbox	This control will be outline in red to show a validation error	Write-off Quantity is required	Label	This control will be used as a label to indicate a validation error in the Write off Quantity textbox	Product	Dropdown list	This control will be outline in red to show a validation error	Product Item is required	Label	This control will be used as a label to indicate a validation error in the product dropdown list	Write-Off Reason	Dropdown list	This control will be outline in red to show a validation error	Write-off Reason is required	Label	This control will be used as a label to indicate a validation error in the Write-off Reason dropdown list
	Control Name	Control Name	Notes																			
	Write-off Quantity	Textbox	This control will be outline in red to show a validation error																			
	Write-off Quantity is required	Label	This control will be used as a label to indicate a validation error in the Write off Quantity textbox																			
	Product	Dropdown list	This control will be outline in red to show a validation error																			
	Product Item is required	Label	This control will be used as a label to indicate a validation error in the product dropdown list																			
	Write-Off Reason	Dropdown list	This control will be outline in red to show a validation error																			
	Write-off Reason is required	Label	This control will be used as a label to indicate a validation error in the Write-off Reason dropdown list																			
	Return to step 3																					
	ALT Step 8: The administrator does not confirm the addition of the new product. - The administrator clicks the cancel button - Return to Step 3																					
ALT Step 9: QuantityOnHand is less than the writeOffQuantity - The system displays an error message box with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Validation Error</td><td>Label</td><td>This control is used to display the type of message box displayed</td></tr><tr><td>Write-off quantity is more than inventory quantity on hand</td><td>Label</td><td>This control is used to inform the administrator that the Write-off quantity is more than inventory quantity on hand</td></tr><tr><td>Okay</td><td>Button</td><td>This control is used to cancel the creation of the user role</td></tr></table> - The administrator clicks the OK button - Return to step 3	Control Name	Control Name	Notes	Validation Error	Label	This control is used to display the type of message box displayed	Write-off quantity is more than inventory quantity on hand	Label	This control is used to inform the administrator that the Write-off quantity is more than inventory quantity on hand	Okay	Button	This control is used to cancel the creation of the user role										
Control Name	Control Name	Notes																				
Validation Error	Label	This control is used to display the type of message box displayed																				
Write-off quantity is more than inventory quantity on hand	Label	This control is used to inform the administrator that the Write-off quantity is more than inventory quantity on hand																				
Okay	Button	This control is used to cancel the creation of the user role																				
CONCLUSION:	This use case concludes when a product is written off.																					
POST-CONDITION:	The write-off Product details are added to the database.																					
BUSINESS RULES	Only the administrator has access to this function.																					
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	The system is offline, and the system administrator will not be able to write off a product.																					

ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 85 - 8.15 Convert Inventory to Product

INVENTORY MANAGEMENT SUBSYSTEM						
AUTHORS (s): Kyle van Eeden			DATE: 09/06/2022			
VERSION: 1.1			LAST REVIEW DATE: 04/09/2022			
USE CASE NAME:	Convert Inventory to Product		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒			
USE CASE ID:	8.15					
PRIORITY:	High					
SOURCE:	ByteXcom Requirements List					
PRIMARY BUSINESS ACTOR	• Inventory Manager or Administrator					
PRIMARY SYSTEM ACTOR	• None					
OTHER PARTICIPATING ACTORS:	• None					
OTHER INTERESTED STAKEHOLDERS:	• None					
DESCRIPTION:	This use case describes the event in which the administrator or inventory manager requests the system to convert inventory items into a finished products. The inventory manager or administrator specifies which final product they wish to produce and the quantity of the finished product to be converted. The system captures the entered details and validates whether there are sufficient quantities of all the required inventory items to produce the desired amount of the specific final product by retrieving the selected product's assembly configuration information from the ProductInventoryItem entity. After validation passes, the system will update the quantity on hand of the product which was produced in the Product entity and the quantities of all inventory items used during the conversion process in the InventoryItem entity. The use case concludes with the system displaying a message to confirm that the requested product has been successfully produced.					
PRE-CONDITION:	• The administrator or inventory manager must be logged into the system. • The product to be produced must have its assembly configuration details setup.					
TRIGGER:	The use case is triggered when the Inventory Manager or Administrator wishes to produce a finished product on the system.					
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE				
		MANUAL ACTION	AUTOMATED ACTION			
	Step 1: The Inventory Manager or Administrator clicks on the Produce Finished Products navigation link button displayed		Step 2: The system retrieves the following details for all products in the Product entity using a .NET controller and SQL Read Query: <table><tr><td>Details to be read</td><td>Attributes in the table</td><td>Notes</td></tr></table>		Details to be read	Attributes in the table
Details to be read	Attributes in the table	Notes				

on the **Manage Store** screen.

Product Name	ProductName	Name of each product retrieved
Quantity On Hand	QuantityOnHand	Available quantity of each product.

The system displays the **Produce Finished Product** screen containing a ASP.NET form with the following controls:

Control Name	Control Type	Control Notes
Product Name	Label	Label for the Product search list box control
Search Product	Search List Box	Used to search for the ProductName attribute value of the product to be produced.
Product to produce:	Label	Label displayed for the Product To Produce textbox control.
Product To Produce	Textbox	Read-Only textbox that displays the currently selected product, if selected.
Quantity To Produce	Label	Label control to provide context for the Quantity textbox control.
Quantity To Produce	Numeric Up-Down	Used to capture the quantity of finished products to be produced.
Produce	Button	Captures and processes the captured details for production process.
Cancel	Button	Navigates user back to the previous page if clicked.

			Note: All controls on the form are enabled by default, unless specified otherwise in the Control Notes section.											
	Step 3: The Inventory Manager or Administrator user enters the name search parameter value of the finished product to be produced. [ALT]		Step 4: The system displays the name of all product(s) that have matched the product search query entered by populating the selectable options of the Search Product search list box control.											
	Step 5: The Inventory Manager or Administrator user selects the product for which to capture the production activity.		Step 6: The system captures the selection and displays the following details for the selected product : <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Product Name</td><td>ProductName</td><td>Populates the text box that captures the currently selected product.</td></tr></table>	Details to be read	Attributes in the table	Notes	Product Name	ProductName	Populates the text box that captures the currently selected product.					
	Details to be read	Attributes in the table	Notes											
	Product Name	ProductName	Populates the text box that captures the currently selected product.											
Step 7: The Inventory Manager or Administrator enters the quantity of the finished product produced.														
Step 8: The Inventory Manager or Administrator clicks on the Produce button. [ALT]		Step 9: The system validates that all the required details have been captured and/or is valid based on the following validation requirements of the ProducedProduct entity: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Product ID</td><td>ProductID</td><td>Required</td></tr><tr><td>Transaction Type</td><td>ActivityType</td><td>Required; Set to enumerable value 'ProduceProduct' by default.</td></tr><tr><td>Production Quantity</td><td>ProducedQuantity</td><td>Required. Numerical values only. The quantity to be produced must be</td></tr></table>	Details to be saved	Attributes in the table	Notes	Product ID	ProductID	Required	Transaction Type	ActivityType	Required; Set to enumerable value 'ProduceProduct' by default.	Production Quantity	ProducedQuantity	Required. Numerical values only. The quantity to be produced must be
Details to be saved	Attributes in the table	Notes												
Product ID	ProductID	Required												
Transaction Type	ActivityType	Required; Set to enumerable value 'ProduceProduct' by default.												
Production Quantity	ProducedQuantity	Required. Numerical values only. The quantity to be produced must be												

					greater than 1.												
			[ALT] Step 10: The system uses a SQL Read Query to retrieve the following details from the ProductInventoryItem entity to retrieve the quantities needed of each inventory item associated with the product to be produced: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Product ID</td><td>ProductID</td><td>The unique identifier of the selected product.</td></tr><tr><td>Inventory Item ID</td><td>InventoryItemID</td><td>Unique identifier for all inventory items needed to produce the finished product requested.</td></tr><tr><td>Inventory Item Quantity</td><td>InventoryItemQuantity</td><td>The quantity of each inventory item needed to produce the product.</td></tr></table>			Details to be saved	Attributes in the table	Notes	Product ID	ProductID	The unique identifier of the selected product.	Inventory Item ID	InventoryItemID	Unique identifier for all inventory items needed to produce the finished product requested.	Inventory Item Quantity	InventoryItemQuantity	The quantity of each inventory item needed to produce the product.
Details to be saved	Attributes in the table	Notes															
Product ID	ProductID	The unique identifier of the selected product.															
Inventory Item ID	InventoryItemID	Unique identifier for all inventory items needed to produce the finished product requested.															
Inventory Item Quantity	InventoryItemQuantity	The quantity of each inventory item needed to produce the product.															
			Step 11: The system validates that there are sufficient quantities available of each inventory item needed to produce the requested quantity of the finished product by using a SQL Read Query to retrieve the following details from the InventoryItem entity: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Quantity On Hand</td><td>QuantityOnHand</td><td>Quantity available of each inventory item needed for production of the requested product.</td></tr></table> The following calculation is used to determine if the quantity on hand of each inventory item required are sufficient: Total of Inventory Item = (Produced Quantity) x (InventoryItemQuantity)			Details to be saved	Attributes in the table	Notes	Quantity On Hand	QuantityOnHand	Quantity available of each inventory item needed for production of the requested product.						
Details to be saved	Attributes in the table	Notes															
Quantity On Hand	QuantityOnHand	Quantity available of each inventory item needed for production of the requested product.															

			The calculated Total of Inventory Item should be less than the QuantityOnHand attribute value retrieved for each required inventory item. [ALT]																							
			Step 12: The system calculates the updated finished product quantity after the production activity and saves the following changes to the respective entities in the database. A .NET controller method is used to create and/or update the following: Save updated product quantity on hand to the Product entity by using a SQL Update query: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Quantity On Hand</td><td>QuantityOnHand</td><td>Added quantity of final product produced to current quantity on hand.</td></tr></table> Add conversion transaction details to the ProducedProduct entity by using a SQL Insert query to save the following information: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Product ID</td><td>ProductID</td><td>Identifier of the product selected for production.</td></tr><tr><td>Activity Type</td><td>ProductTransactionType</td><td>Captured as 'ProduceProduct' from the ProductTransactionType entity</td></tr><tr><td>Quantity Before</td><td>QuantityBefore</td><td>Quantity On Hand of the selected product.</td></tr><tr><td>Quantity After</td><td>Quantity After</td><td>Quantity after production quantity is added.</td></tr><tr><td>Captured Date</td><td>Transaction Date</td><td>Current date and time stamp of the production date.</td></tr></table>	Details to be saved	Attributes in the table	Notes	Quantity On Hand	QuantityOnHand	Added quantity of final product produced to current quantity on hand.	Details to be saved	Attributes in the table	Notes	Product ID	ProductID	Identifier of the product selected for production.	Activity Type	ProductTransactionType	Captured as 'ProduceProduct' from the ProductTransactionType entity	Quantity Before	QuantityBefore	Quantity On Hand of the selected product.	Quantity After	Quantity After	Quantity after production quantity is added.	Captured Date	Transaction Date
Details to be saved	Attributes in the table	Notes																								
Quantity On Hand	QuantityOnHand	Added quantity of final product produced to current quantity on hand.																								
Details to be saved	Attributes in the table	Notes																								
Product ID	ProductID	Identifier of the product selected for production.																								
Activity Type	ProductTransactionType	Captured as 'ProduceProduct' from the ProductTransactionType entity																								
Quantity Before	QuantityBefore	Quantity On Hand of the selected product.																								
Quantity After	Quantity After	Quantity after production quantity is added.																								
Captured Date	Transaction Date	Current date and time stamp of the production date.																								

		The system simultaneously updates the quantities of all the inventory items used for the production of the final product by saving the following inventory item transaction details: In the InventoryItem entity by using a SQL Update Query: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Quantity on Hand</td><td>QuantityonHand</td><td>Subtract the quantity of each inventory item used for producing the requested quantity of the final product.</td></tr></table>	Details to be saved	Attributes in the table	Notes	Quantity on Hand	QuantityonHand	Subtract the quantity of each inventory item used for producing the requested quantity of the final product.												
	Details to be saved	Attributes in the table	Notes																	
	Quantity on Hand	QuantityonHand	Subtract the quantity of each inventory item used for producing the requested quantity of the final product.																	
		Add the inventory item transaction details to the InventoryItemTransaction entity by using a SQL Insert query: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Inventory Item Transaction ID</td><td>InventoryTransactionItem ID</td><td>Unique GUID generated by the system.</td></tr><tr><td>Activity Type</td><td>Transaction Type</td><td>Captured as 'ProduceProduct' for conversion activities by default.</td></tr><tr><td>Quantity Before</td><td>QuantityBefore</td><td>Quantity On Hand of the needed Inventory Item.</td></tr><tr><td>Quantity After</td><td>Quantity After</td><td>Quantity after quantity used for production is subtracted.</td></tr><tr><td>Captured Date</td><td>Transaction Date</td><td>Current date and time stamp of the production date.</td></tr></table>	Details to be saved	Attributes in the table	Notes	Inventory Item Transaction ID	InventoryTransactionItem ID	Unique GUID generated by the system.	Activity Type	Transaction Type	Captured as ' ProduceProduct ' for conversion activities by default.	Quantity Before	QuantityBefore	Quantity On Hand of the needed Inventory Item.	Quantity After	Quantity After	Quantity after quantity used for production is subtracted.	Captured Date	Transaction Date	Current date and time stamp of the production date.
	Details to be saved	Attributes in the table	Notes																	
Inventory Item Transaction ID	InventoryTransactionItem ID	Unique GUID generated by the system.																		
Activity Type	Transaction Type	Captured as ' ProduceProduct ' for conversion activities by default.																		
Quantity Before	QuantityBefore	Quantity On Hand of the needed Inventory Item.																		
Quantity After	Quantity After	Quantity after quantity used for production is subtracted.																		
Captured Date	Transaction Date	Current date and time stamp of the production date.																		
		Step 13: The system displays a production successful confirmation message box on the re-initialized Produce Finished Products screen.																		
ALTERNATE COURSES:	ALT Step 3: The Inventory Manager or Administrator no longer wishes to capture finished products produced on the system:																			

	• The Inventory Manager or Administrator clicks on the Cancel button • The Inventory Manager or Administrator is redirected to the Manage Store screen • The use case is terminated									
	ALT Step 8: The Inventory Manager or Administrator no longer wishes to capture finished products produced on the system: • The Inventory Manager or Administrator clicks on the Cancel button • The Inventory Manager or Administrator is redirected to the Manage Store screen • The use case is terminated									
	ALT Step 9 a: The details entered are not in the correct format or the required fields are empty • The system displays a validation error message box with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Product Name</td><td>Search List Box</td><td>This control will be outline in red to show a validation error has occurred in this control.</td></tr><tr><td>Selection of Product to be produced is Required.</td><td>Label</td><td>This control will be used as a label to indicate a validation error that was raised for the Product Name Search List Box control.</td></tr></table>	Control Name	Control Name	Notes	Product Name	Search List Box	This control will be outline in red to show a validation error has occurred in this control.	Selection of Product to be produced is Required.	Label	This control will be used as a label to indicate a validation error that was raised for the Product Name Search List Box control.
	Control Name	Control Name	Notes							
	Product Name	Search List Box	This control will be outline in red to show a validation error has occurred in this control.							
	Selection of Product to be produced is Required.	Label	This control will be used as a label to indicate a validation error that was raised for the Product Name Search List Box control.							
	• Return to Step 3.									
	ALT Step 9 b: The details entered are not in the correct format or the required fields are empty • The system displays a validation error message box with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Production Quantity</td><td>Textbox</td><td>This control will be outline in red to show a validation error has occurred in this control.</td></tr><tr><td>The Quantity of the Product to be produced must be greater than 1.</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Production Quantity Textbox control.</td></tr></table>	Control Name	Control Name	Notes	Production Quantity	Textbox	This control will be outline in red to show a validation error has occurred in this control.	The Quantity of the Product to be produced must be greater than 1.	Label	This control will be used as a label to indicate a validation error in the Production Quantity Textbox control.
	Control Name	Control Name	Notes							
	Production Quantity	Textbox	This control will be outline in red to show a validation error has occurred in this control.							
The Quantity of the Product to be produced must be greater than 1.	Label	This control will be used as a label to indicate a validation error in the Production Quantity Textbox control.								
• Return to Step 5.										
ALT Step 11: There is insufficient quantities available of all the required inventory items to produce the requested amount of a finished product. • The system displays a validation error message box with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Production Failed. Insufficient inventory to produce [Quantity] of [ProductName]</td><td>Label</td><td>Informs the Administrator or Inventory Manager that the the quantities of all required inventory items are insufficient to produce the requested quantity of the product selected.</td></tr></table>	Control Name	Control Name	Notes	Production Failed. Insufficient inventory to produce [Quantity] of [ProductName]	Label	Informs the Administrator or Inventory Manager that the the quantities of all required inventory items are insufficient to produce the requested quantity of the product selected.				
Control Name	Control Name	Notes								
Production Failed. Insufficient inventory to produce [Quantity] of [ProductName]	Label	Informs the Administrator or Inventory Manager that the the quantities of all required inventory items are insufficient to produce the requested quantity of the product selected.								
• Return to Step 3.										
CONCLUSION:	This use case concludes when the system displays a production success alert message on the re-initialized Produce Finished Product screen.									

POST-CONDITION:	- The [QuantityOnHand] attribute value in the Product entity is updated. - The [QuantityOnHand] attribute value in the InventoryItem entity is updated for each inventory item used in the conversion to finished products. - The conversion's transactional details are added to the ProducedProduct entity.
BUSINESS RULES	- Only the Inventory Manager and Administrator roles should have access to this functionality. - The system should not allow for more finished products to be produced than what is possible given the available quantities of required inventory items as per the conversion configuration.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	- The system is offline, and the Inventory Manager or Administrator will not be able to capture finished products produced on the system. - The quantity of the finished product to be produced must be at least one in order to use this functionality.
ASSUMPTIONS:	This functionality is to be used after the inventories have been assembled and packaged into the final product to be sold.
OPEN ISSUES:	None

Table 86 - 8.16 Search Procured Inventory

INVENTORY MANAGMENT SUBSYSTEM												
AUTHORS (s): Thenjiwe Ntsonda		DATE: 08/06/2022										
VERSION: 1.1		LAST REVIEW DATE: 08/06/2022										
USE CASE NAME:	Search Procured Inventory	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒										
USE CASE ID:	8.15											
PRIORITY:	Medium											
SOURCE:	ByteXcom Requirements List											
PRIMARY BUSINESS ACTOR	• Administrator											
PRIMARY SYSTEM ACTOR	• None											
OTHER PARTICIPATING ACTORS:	• None											
OTHER INTERESTED STAKEHOLDERS:	• None											
DESCRIPTION:	This use case describes the process where the administrator searches for a supplier on the system. The system responds by displaying the information relating to the search query.											
PRE-CONDITION:	The administrator must be logged onto the system.											
TRIGGER:	This use case is called when the administrator wants to search for a supplier on the system.											
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE										
		MANUAL ACTION	AUTOMATED ACTION									
	Step 1: The administrator clicks on the Inventory Procured navigation link on the Inventory navigation bar		Step 2: The system retrieves the following attributes from each entry in the InventoryProcured table, using a method in a .NET controller and an SQL READ query:									
			<table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Invoice Number</td><td>InvoiceNo</td></tr><tr><td>Quantity Received</td><td>QuantityReceived</td></tr><tr><td>Date Logged</td><td>DateLogged</td></tr></table>		Details to be read	Attributes in the table	Invoice Number	InvoiceNo	Quantity Received	QuantityReceived	Date Logged	DateLogged
			Details to be read	Attributes in the table								
			Invoice Number	InvoiceNo								
Quantity Received	QuantityReceived											
Date Logged	DateLogged											
The system retrieves the company name associated with each record from												

the **Supplier** table, using a method in a .NET controller with an SQL READ query:

Details to be read	Attributes in the table
Company Name	CompanyName

The system retrieves the inventory item name associated with each record from the **InventoryItem** table, using a method in a .NET controller with an SQL READ query:

Details to be read	Attributes in the table
Inventory Item Name	InventoryItemName

and displays them on the **Inventory Procured Screen** with the following controls:

Control Name	Control Type	Notes
Procured Inventory Item List	Heading	Page main heading
Capture Procured Inventory Item	Button	When clicked, this control will direct the administrator to the Capture Procured Inventory Item screen
Show “” entries	Label and drop down	This control will be used to set the number of entries the administrator wants to display on a page
Search bar	Textbox	This control will be used to enter the search query
Results table	Table	This control will be used to display all

					the procured inventory that exists on the system
	Supplier	Table Heading			This control will be used as a heading to display the Supplier's Company Name
	Inventory Item Name	Table Heading			This control will be used as a heading to display the name of the Inventory Item
	Quantity Received	Table Heading			This control will be used as a heading to display the quantity of the inventory item received
	Date Logged	Table Heading			This control will be used as a heading to display the date of which the Inventory Item was procured.
	Showing “” to “” of “” entries	Label			This control will be used to display the number of entries displayed per page as well as the total number of entries in the database
	“Previous” and “Next”	Label and button			This control is used to navigate through the entry pages and to indicate on which page the administrator

					is currently on
	Step 4: The administrator enters their search query into the search textbox		Step 5: The system searches the table rows in the data table and displays the Procured Inventory Item(s) matching the search query [ALT]		
	ALT Step 5: The search query entered does not match any records existing in the InventoryProcured entity. • The system displays a “No Records Found” message in the results table. • Return to step 4 or terminate the use case				
CONCLUSION:	This use case concludes when the desired procured inventory item’s details are displayed on the system.				
POST-CONDITION:	The InventoryProcured entity was searched for and an entry that corresponds to the search query or parameter entered.				
BUSINESS RULES	• Only the administrator can search for procured inventory items on the system.				
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• None				
ASSUMPTIONS:	• None				
OPEN ISSUES:	• None				

Table 87 - 8.17 View Procured Inventory Details

INVENTORY MANAGEMENT SUBSYSTEM								
AUTHORS (s): Thenjiwe Ntsonda		DATE: 08/06/2022						
VERSION: 1.0		LAST REVIEW DATE: 08/06/2022						
USE CASE NAME:	View Procured Inventory Details	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒						
USE CASE ID:	8.16							
PRIORITY:	High							
SOURCE:	ByteXcom Requirements List							
PRIMARY BUSINESS ACTOR	• Administrator							
PRIMARY SYSTEM ACTOR	• None							
OTHER PARTICIPATING ACTORS:	• None							
OTHER INTERESTED STAKEHOLDERS:	• None							
DESCRIPTION:	This use case describes the process in which the administrator wishes to view the details of the inventory received from a supplier. The use case begins when the administrator wishes to view the details of the inventory received from a supplier on the system. The administrator will select the relevant inventory item and view its details.							
PRE-CONDITION:	• The administrator must be logged onto the system.							
TRIGGER:	The use case is triggered when the administrator wishes to view the details of the inventory received from a supplier.							
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE						
		MANUAL ACTION	AUTOMATED ACTION					
	Step 1: The administrator clicks on the View Button button		Step 2: The system retrieves the following attributes from the following tables, using a method in a .NET controller and an SQL READ query: From the Supplier Table					
			<table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Company Name</td><td>CompanyName</td></tr></table>		Details to be read	Attributes in the table	Company Name	CompanyName
			Details to be read	Attributes in the table				
Company Name	CompanyName							
From the Inventory Item Table								

			Details to be read	Attributes in the table	
			Inventory Item Name	InventoryItemName	
			and displays the Procured Inventory Item Details Screen which contains an MVC form with the following controls:		
			Control Name	Control Type	Notes
			Procured Inventory Item Details	Heading	Page main heading
			Company Name	Label	This control will be used as a label to indicate the supplier's company name attribute
			Company Name	Label	This control will be used by the administrator to display the supplier's company name
			Invoice Number	Label	This control will be used as a label to indicate the supplier's invoice number attribute
			Invoice Number	Label	This control will be used by the administrator to display the supplier's invoice number
			Inventory Item Name	Label	This control will be used as a label to indicate the inventory item's name attribute

			Inventory Item Name	Label	This control will be used by the administrator to display the inventory item's name
			Quantity Received	Label	This control will be used as a label to indicate the quantity received attribute
			Quantity Received	Label	This control will be used by the administrator to display the quantity received
			Return	Button	This control will be used to return the administrator to the Procured Inventory Screen
			Note: All controls are enabled by default, unless otherwise stated. All text input fields are empty by default, unless stated otherwise. [ALT]		
ALTERNATE COURSES:	ALT Step 2: The administrator wishes to return to the previous page The administrator clicks the Return button The system redirects the administrator to the Procured Inventory Items Screen .				
CONCLUSION:	This use case concludes when the inventory procured details are displayed on the system				
POST-CONDITION:	The selected procured inventory details are displayed on the system.				
BUSINESS RULES	Only the administrator can view the details of the procured inventory.				
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None				
ASSUMPTIONS:	None				
OPEN ISSUES:	None				

Table 88 - 8.18 Capture Procured Inventory

INVENTORY MANAGEMENT SUBSYSTEM				
AUTHORS (s): Thenjiwe Ntsonda		DATE: 08/06/2022		
VERSION: 1.0		LAST REVIEW DATE: 08/06/2022		
USE CASE NAME:	Capture Procured Inventory	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒		
USE CASE ID:	8.17			
PRIORITY:	High			
SOURCE:	ByteXcom Requirements List			
PRIMARY BUSINESS ACTOR	• Administrator			
PRIMARY SYSTEM ACTOR	• None			
OTHER PARTICIPATING ACTORS:	• None			
OTHER INTERESTED STAKEHOLDERS:	• None			
DESCRIPTION:	This use case describes the process in which the administrator wishes to capture the inventory received from a supplier. The use case begins when the administrator wishes to increase the inventory count on the system. The administrator will add all of the pertinent information regarding the procured inventory item(s) and saves it to the database.			
PRE-CONDITION:	• The administrator must be logged onto the system.			
TRIGGER:	The use case is triggered when the administrator wishes to capture the inventory received from a supplier.			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE		
		MANUAL ACTION	AUTOMATED ACTION	
	Step 1: The administrator clicks on the Capture Procured Inventory button		Step 2: The system retrieves the following attributes from the following tables, using a method in a .NET controller and an SQL READ query: From the Supplier Table	
			<table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Company Name</td><td>CompanyName</td></tr></table>	Details to be read
Details to be read	Attributes in the table			
Company Name	CompanyName			

From the **Inventory Item** Table

Details to be read	Attributes in the table
Inventory Item Name	Inventory/ItemName

to populate the drop down lists and displays the **Capture Procured Inventory Item Screen** which contains an MVC form with the following controls:

Control Name	Control Type	Notes
Capture Procured Inventory Item	Heading	Page main heading
Company Name	Label	This control will be used as a label to indicate the supplier's company name attribute
Company Name	Dropdown	This control will be used by the administrator to enter the supplier's company name
Invoice Number	Label	This control will be used as a label to indicate the supplier's invoice number attribute
Invoice Number	Textbox	This control will be used by the administrator to enter the supplier's invoice number
Inventory Item Name	Label	This control will be used as a label to indicate the inventory item's name attribute
Inventory Item Name	Dropdown	This control will be used

					by the administrator to enter the inventory item's name
			Quantity Received	Label	This control will be used as a label to indicate the quantity received attribute
			Quantity Received	Numeric UpDown	This control will be used by the administrator to enter the quantity received
			Capture	Button	This control will be used to confirm the capture of the procured inventory item
			Cancel	Button	This control will be used to cancel the capture of the Procured Inventory and return the administrator to the Capture Procured Inventory Screen
	Step 3: The system administrator enters the details and clicks on the Capture button [ALT]		Note: All controls are enabled by default, unless otherwise stated. All text input fields are empty by default, unless stated otherwise.		
			Step 4: The system displays a modal dialogue with the following controls:		
			Control Name	Control Type	Notes
			"Confirm Procured Inventory Details"	Label	This control is used to request confirmation on the capturing of the procured inventory
			Confirm	Button	This control is used to confirm the

					capturing of the procured inventory						
			Cancel	Button	This control is used to close the modal dialogue and return the administrator to the Capture Procured Inventory Item Screen						
	Step 5: The administrator clicks on the Confirm button [ALT]		Step 6: The system captures and validates the entered details against the following tables:								
			From the InventoryProcured Table								
			<table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Invoice Number</td><td>InvoiceNumber</td><td>Required Unique</td></tr><tr><td>Quantity Received</td><td>QuantityReceived</td><td>Required Digits only No negative</td></tr></table>	Details to be read	Attributes in the table	Notes	Invoice Number	InvoiceNumber	Required Unique	Quantity Received	QuantityReceived
Details to be read	Attributes in the table	Notes									
Invoice Number	InvoiceNumber	Required Unique									
Quantity Received	QuantityReceived	Required Digits only No negative									
		From the Supplier Table									
		<table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Supplier</td><td>Id</td><td>Required</td></tr></table>	Details to be read	Attributes in the table	Notes	Supplier	Id	Required			
Details to be read	Attributes in the table	Notes									
Supplier	Id	Required									
		From the Inventory Item Table									
		<table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Inventory Item</td><td>Id</td><td>Required</td></tr></table>	Details to be read	Attributes in the table	Notes	Inventory Item	Id	Required			
Details to be read	Attributes in the table	Notes									
Inventory Item	Id	Required									
		[ALT]									
		Step 7: The system increments the QuantityOnHand attribute value from the InventoryItem Table:									
		<table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>QuantityOnHand</td><td>QuantityOnHand</td></tr></table>	Details to be read	Attributes in the table	QuantityOnHand	QuantityOnHand					
Details to be read	Attributes in the table										
QuantityOnHand	QuantityOnHand										
		With the value added from the QuantityReceived attribute in the InventoryProcured Table.									

			The system then saves the details in the Inventory Procured table using a method in a .NET controller and an SQL INSERT query: <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Id</td><td>Id</td><td>Unique attribute automatically generated by the system</td></tr><tr><td>Company Name</td><td>SupplierId</td><td>Required FK from Supplier Table</td></tr><tr><td>Invoice Number</td><td>InvoiceNumber</td><td>Required Unique</td></tr><tr><td>Quantity Received</td><td>QuantityReceived</td><td>Required Digits only No negative</td></tr><tr><td>Inventory Item Name</td><td>ItemID</td><td>Required FK from Inventory Item Table</td></tr><tr><td>Date Logged</td><td>DateLogged</td><td>Automatically Generated</td></tr></table>	Details to be read	Attributes in the table	Notes	Id	Id	Unique attribute automatically generated by the system	Company Name	SupplierId	Required FK from Supplier Table	Invoice Number	InvoiceNumber	Required Unique	Quantity Received	QuantityReceived	Required Digits only No negative	Inventory Item Name	ItemID	Required FK from Inventory Item Table	Date Logged	DateLogged	Automatically Generated
	Details to be read	Attributes in the table	Notes																					
Id	Id	Unique attribute automatically generated by the system																						
Company Name	SupplierId	Required FK from Supplier Table																						
Invoice Number	InvoiceNumber	Required Unique																						
Quantity Received	QuantityReceived	Required Digits only No negative																						
Inventory Item Name	ItemID	Required FK from Inventory Item Table																						
Date Logged	DateLogged	Automatically Generated																						
			Step 8: The system directs the administrator to the Captured Inventory Item List and displays a bootstrap alert notification with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"Procured Inventory Captured Successfully"</td><td>Label</td><td>This control is used to indicate a successful capture</td></tr></table>	Control Name	Control Name	Notes	"Procured Inventory Captured Successfully"	Label	This control is used to indicate a successful capture															
Control Name	Control Name	Notes																						
"Procured Inventory Captured Successfully"	Label	This control is used to indicate a successful capture																						
ALTERNATE COURSES:	ALT Step 3: The administrator does not want to capture the procured inventory anymore - The administrator clicks on the cancel button - The use case is terminated																							
	ALT Step 5: The administrator clicks on the Cancel button Return to Step 2																							
	ALT Step 6: The details entered are not in the correct format or the required fields are empty The system displays a validation error with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Company Name</td><td>Dropdown</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>The Company Name field is required</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Company Name dropdown</td></tr></table>			Control Name	Control Name	Notes	Company Name	Dropdown	This control will be outline in red to show a validation error	The Company Name field is required	Label	This control will be used as a label to indicate a validation error in the Company Name dropdown												
	Control Name	Control Name	Notes																					
Company Name	Dropdown	This control will be outline in red to show a validation error																						
The Company Name field is required	Label	This control will be used as a label to indicate a validation error in the Company Name dropdown																						

	Invoice Number	Textbox	This control will be outline in red to show a validation error
	The Invoice Number field is required	Label	This control will be used as a label to indicate a validation error in the Invoice Number textbox
	Inventory Item Name	Dropdown	This control will be outline in red to show a validation error
	The Inventory Item Name field is required	Label	This control will be used as a label to indicate a validation error in the Inventory Item Name dropdown
	Quantity Received	Numerical UpDown	This control will be outline in red to show a validation error
	The Quantity Received field is required	Label	This control will be used as a label to indicate a validation error in the Quantity Received textbox
Return to step 2			
CONCLUSION:	This use case concludes when the inventory procured is captured		
POST-CONDITION:	The procured inventory details are saved to the database and the quantity on hand is increased.		
BUSINESS RULES	Only the administrator can capture procured inventory.		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

Table 89 - 8.19 Capture Returned Product

INVENTORY MANAGEMENT SUBSYSTEM			
AUTHORS (s): Thenjiwe Ntsonda		DATE: 14/05/2022	
VERSION: 1.0		LAST REVIEW DATE: 14/05/2022	
USE CASE NAME:	Capture Returned Product	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒	
USE CASE ID:	8.19		
PRIORITY:	High		
SOURCE:	ByteXcom Requirements List		
PRIMARY BUSINESS ACTOR	• Administrator or Sales Assistant		
PRIMARY SYSTEM ACTOR	• None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	This use case begins when the administrator or sales assistant wishes to capture products which have been returned by customers and resellers on the system. The administrator or sales assistant begins by selecting the order they wish to log the return for and adding the relevant information regarding its return. The system will then increment the product quantity with the quantity of products returned and change the order status to “Returned”.		
PRE-CONDITION:	• The administrator or sales assistant must be logged in. • The order status of the order must be “Dispatched” or “Delivered”		
TRIGGER:	This use case is called when the administrator or sales assistant wishes to capture products which have been returned by customers and resellers on the system.		
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE	
		MANUAL ACTION	AUTOMATED ACTION
	Step 1: The administrator or wishes to capture the return of product(s) in an order.		Step 2: The system invokes use case 5.7 View Order Overview
	Step 3: The system administrator or sales assistant clicks on the “Log Return” button		Step 4: The system retrieves the name(s) of the product(s) from the Product entity using an SQL query with the following attributes:
		Details to be read	Attributes in the table
	Name	ProductId	

The system retrieves the return reasons from the **ReturnReason** entity with the following attributes:

Details to be read	Attributes in the table
Return Reason	ReturnReasonId

And displays the **Log Returned Product Screen** with the following controls:

Control Name	Control Type	Notes
Log Returned Product	Heading	Page main heading
Order Number	Label	This control will be used as a label to indicate the order number of the selected order
Order Number	Disabled Dropdown	This control will be used as a label to populate the order number of the selected order
Product Returned	Label	This control will be used as a label to indicate the Product Name attribute of the returned product record
Product Returned	Dropdown	This control will be used by the administrator or sales assistant to select the name of the product returned
Quantity Received	Label	This control will be used as a label to indicate the Quantity received attribute of the product returned
Quantity Received	Numerical Updown	This control will be used by the

					administrator to insert the quantity received of the returned product.
			Return Reason	Label	This control will be used as a label to indicate the Return Reason attribute of the product returned.
			Return Reason	Dropdown	This control will be used by the administrator to select a return reason for the product returned.
			Log Return	Button	This control will be used to confirm the logging of the return of a product
			Cancel	Button	This control will be used to cancel the return of a product
			Note: All textboxes are enabled by default unless stated otherwise		
	Step 5: The system administrator or system administrator enters the details and clicks on the Log Return button [ALT]		Step 6: The system displays a modal dialogue with the following controls:		
			Control Name	Control Type	Notes
			"Confirm Returned Product Details"	Label	This control is used to request confirmation on the capturing of the returned product
			Confirm	Button	This control is used to confirm the capture of the returned product
			Cancel	Button	This control is used to close the modal dialogue and return the

					administrator to the Log Returned Product Screen												
			Step 7: The system reads the following from the OrderLine table using an SQL READ QUERY:														
			<table><tr><th>Details to be read</th><th colspan="2">Attributes in the table</th></tr><tr><td>ProductName</td><td colspan="2">ProductId</td></tr><tr><td>Quantity</td><td colspan="2">Quantity</td></tr></table>			Details to be read	Attributes in the table		ProductName	ProductId		Quantity	Quantity				
	Details to be read	Attributes in the table															
	ProductName	ProductId															
	Quantity	Quantity															
			And then captures and validates the entered details against the ReturnedProduct table [ALT]														
			<table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Product Returned</td><td>ProductId</td><td>Required, must exist in selected order</td></tr><tr><td>Quantity Received</td><td>QuantityReceived</td><td>Required, Numerical Characters Only, Cannot be zero, Cannot include negative numbers, must not exceed quantity ordered</td></tr><tr><td>Return Reason</td><td>ReturnReasonId</td><td>Required</td></tr></table>			Details to be read	Attributes in the table	Notes	Product Returned	ProductId	Required, must exist in selected order	Quantity Received	QuantityReceived	Required, Numerical Characters Only, Cannot be zero, Cannot include negative numbers, must not exceed quantity ordered	Return Reason	ReturnReasonId	Required
	Details to be read	Attributes in the table	Notes														
	Product Returned	ProductId	Required, must exist in selected order														
Quantity Received	QuantityReceived	Required, Numerical Characters Only, Cannot be zero, Cannot include negative numbers, must not exceed quantity ordered															
Return Reason	ReturnReasonId	Required															

		Step 8: The system increases the QuantityOnHand attribute in the Product entity with the value inputted in the QuantityReceived attribute in the ReturnedProduct entity using a method in a .NET controller <table><tr><th>Details to be incremented</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>QuantityOnHand</td><td>QuantityOnHand</td><td></td></tr></table> The system then updates the OrderStatus attribute value in the Order entity to “Returned” using a method in a .NET controller <table><tr><th>Details to be updated</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>OrderStatus</td><td>OrderStatus</td><td></td></tr></table>	Details to be incremented	Attributes in the table	Notes	QuantityOnHand	QuantityOnHand		Details to be updated	Attributes in the table	Notes	OrderStatus	OrderStatus							
	Details to be incremented	Attributes in the table	Notes																	
	QuantityOnHand	QuantityOnHand																		
	Details to be updated	Attributes in the table	Notes																	
	OrderStatus	OrderStatus																		
		Step 9: The system saves the details to the ReturnedProduct entity using an SQL INSERT query with the following attributes: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>ReturnedProductId</td><td>ReturnedProductId</td><td>Automatically generated by the system.</td></tr><tr><td>Product Returned</td><td>ProductId</td><td>Required</td></tr><tr><td>Quantity Received</td><td>QuantityReceived</td><td>Required, Numerical Characters Only, Cannot be zero, Cannot include negative numbers</td></tr><tr><td>Return Reason</td><td>ReturnReasonId</td><td>Required</td></tr><tr><td>DateLogged</td><td>DateLogged</td><td>Automatically generated by the system.</td></tr></table>	Details to be saved	Attributes in the table	Notes	ReturnedProductId	ReturnedProductId	Automatically generated by the system.	Product Returned	ProductId	Required	Quantity Received	QuantityReceived	Required, Numerical Characters Only, Cannot be zero, Cannot include negative numbers	Return Reason	ReturnReasonId	Required	DateLogged	DateLogged	Automatically generated by the system.
	Details to be saved	Attributes in the table	Notes																	
	ReturnedProductId	ReturnedProductId	Automatically generated by the system.																	
	Product Returned	ProductId	Required																	
	Quantity Received	QuantityReceived	Required, Numerical Characters Only, Cannot be zero, Cannot include negative numbers																	
Return Reason	ReturnReasonId	Required																		
DateLogged	DateLogged	Automatically generated by the system.																		

Step 10: The system retrieves the returned product details from the **ReturnedProduct** entity with the following attributes:

Details to be read	Attributes in the table
Order Number	Id
Product Returned	ProductId
Return Reason	ReturnReasonId
Quantity Received	QuantityReceived
Date Logged	DateLogged

redirects the system administrator to the **Returned Products Screen** with the following controls:

Control Name	Control Type	Notes
Returned Products List	Heading	Page main heading
Show "" entries	Label and drop down	This control will be used to set the number of entries the administrator wants to display on a page
Search bar	Textbox	This control will be used to enter the search query
Results table	Table	This control will be used to display all the orders that exist on the system
Order Number	Table Heading	This control will be used as a heading to display the order number
Products Returned	Table Heading	This control will be used as a heading to display the product which are returned.
Return Reason	Table Heading	This control will be used as a heading

					to display the product's reason for return
			Date Logged	Table Heading	This control will be used as a heading to display the date on which the return was logged.
			Showing “” to “” of “” entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database
			“Previous” and “Next”	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on
			and displays a bootstrap alert notification with the following controls:		
			Control Name	Control Name	Notes
			“Return Logged successfully”	Label	This control is used to indicate a successful log of a return

ALTERNATE COURSES:	ALT Step 5: The administrator or sales assistant does not wish to log the return of a product anymore - The administrator or sales assistant clicks on the cancel button - Return to Step 2		
	ALT Step 7A: The details entered are not in the correct format or the required fields are empty The system displays a validation error with the following controls		
	Control Name	Control Name	Notes
	Product Returned	Dropdown	This control will be outline in red to show a validation error
	Product is required	Label	This control will be used as a label to indicate a validation error in the Product textbox
	Quantity Received	Numerical Updown	This control will be outline in red to show a validation error
	Quantity is Required	Label	This control will be used as a label to indicate a validation

			error in the QuantityReceived textbox						
	Return Reason	Dropdown	This control will be outline in red to show a validation error						
	Return Reason is Required	Label	This control will be used as a label to indicate a validation error in the ReturnReason textbox						
	Return to step 3								
	ALT Step 7B: The selected product does not exist in the selected order's OrderLine entity:								
	The system displays an error message with the following controls:								
	<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"Oops! You attempted to return a product that was not ordered!"</td><td>Label</td><td>This control is used to indicate the unsuccessful log of a return if the product was not ordered in the selected order.</td></tr></table>	Control Name	Control Name	Notes	"Oops! You attempted to return a product that was not ordered!"	Label	This control is used to indicate the unsuccessful log of a return if the product was not ordered in the selected order.		
	Control Name	Control Name	Notes						
	"Oops! You attempted to return a product that was not ordered!"	Label	This control is used to indicate the unsuccessful log of a return if the product was not ordered in the selected order.						
	ALT Step 7C: The entered product quantity exceeds what was ordered:								
The system displays an error message with the following controls:									
<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"Oops! You attempted to return more than what was ordered!"</td><td>Label</td><td>This control is used to indicate the unsuccessful log of a return if the quantity entered exceeds the quantity ordered.</td></tr></table>	Control Name	Control Name	Notes	"Oops! You attempted to return more than what was ordered!"	Label	This control is used to indicate the unsuccessful log of a return if the quantity entered exceeds the quantity ordered.			
Control Name	Control Name	Notes							
"Oops! You attempted to return more than what was ordered!"	Label	This control is used to indicate the unsuccessful log of a return if the quantity entered exceeds the quantity ordered.							
CONCLUSION:	The system displays an alert which confirms that the return has been logged onto the system successfully.								
POST-CONDITION:	The order status is changed to "Includes Returned Product(s)", the product [QuantityOnHand] attribute value is increased, and the returned product's details are saved to the ReturnedProduct entity in the database.								
BUSINESS RULES	Only products which have been received by the business can be logged for a return.								
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	The administrator and sales assistant have a stable internet connection.								
ASSUMPTIONS:	None								
OPEN ISSUES:	None								

Table 90 - 8.20 Package Order

INVENTORY MANAGEMENT SUBSYSTEM		
AUTHORS (s): Kyle van Eeden		DATE: 18/07/2022
VERSION: 1.1		LAST REVIEW DATE: 25/07/2022
USE CASE NAME:	Package Order	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒
USE CASE ID:	8.20	
PRIORITY:	High	
SOURCE:	ByteXpress Requirements List	
PRIMARY BUSINESS ACTOR	• Administrator/ Sales Assistant	
PRIMARY SYSTEM ACTOR	• None	
OTHER PARTICIPATING ACTORS:	• None	
OTHER INTERESTED STAKEHOLDERS:	• None	
DESCRIPTION:	This use case begins when either a system user with an Admin or Sales Assistant role assigned to their account wishes to package an order on the system before dispatching the order’s parcel to the courier services. After requesting to package an order, the system will display all existing package order transactions along with an option to capture packaged product for an order on the system. Once the system user selects the option to capture packaged products for an order, the system will display a form which prompts the the user to select a product to package, the order for which it is being packaged and the quantity of the product to be packaged. Once the user has filled out the required details and submitted the form details, the system will validate the captured form input and determine whether there is sufficient stock of the requested product on hand. Once validation passes, the system will read the order’s details from the Order entity to determine whether it is a reseller order or not. For customer orders, the system will create a new package order transactional record in the OrderProduct entity without subtracting the quantity of packaged products from the inventory levels. For reseller orders the system will read the QuantityOnHand value for the selected product from the Product entity to determine whether there are sufficient stock on hand. The system will then deduct the requested packaging quantity from the QuantityOnHand value and save the updated quantity. The use case concludes once a bootstrap	

	alert is displayed which informs the user that the products have been successfully packaged.																				
PRE-CONDITION:	• The user requesting to package an order must be logged into an account which has either the Admin or Sales Assistant role assigned. • At least one product has to exist on the system to perform this use case. • At least one order must have been placed in order to perform this use case. • The user is on the 'View Order Details' screen.																				
TRIGGER:	This use case is called when a system administrator or sales assistant wishes to capture products that have been packaged for a particular order on the system.																				
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE																			
		MANUAL ACTION	AUTOMATED ACTION																		
	Step 1: The system administrator or sales assistant clicks on the Package Order button.		Step 2: The system will retrieve all package order transactions that exist in the OrderProduct table using a .NET Controller with a SQL Read query to retrieve the following details:																		
			<table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Order Number</td><td>OrderId</td><td>The order reference number retrieved for each order.</td></tr><tr><td>Product ID</td><td>ProductId</td><td>The name of each product which has been packaged for a specific order.</td></tr><tr><td>Product Quantity</td><td>ProductQuantity</td><td>The quantity of each product which has been packaged.</td></tr><tr><td>Transaction Date</td><td>DateLogged</td><td>The date which each transaction was captured.</td></tr><tr><td>Created By</td><td>ActorName</td><td>The name of the user which has effected the package order transaction.</td></tr></table>	Details to be read	Attributes in the table	Notes	Order Number	OrderId	The order reference number retrieved for each order.	Product ID	ProductId	The name of each product which has been packaged for a specific order.	Product Quantity	ProductQuantity	The quantity of each product which has been packaged.	Transaction Date	DateLogged	The date which each transaction was captured.	Created By	ActorName	The name of the user which has effected the package order transaction.
		Details to be read	Attributes in the table	Notes																	
		Order Number	OrderId	The order reference number retrieved for each order.																	
		Product ID	ProductId	The name of each product which has been packaged for a specific order.																	
Product Quantity	ProductQuantity	The quantity of each product which has been packaged.																			
Transaction Date	DateLogged	The date which each transaction was captured.																			
Created By	ActorName	The name of the user which has effected the package order transaction.																			

The system will also read the following details from the **Product** entity:

Details to be read	Attributes in the table	Notes
Prodduct Name	ProductName	Name each product packaged for an order.

The system will then display the **Packaged Orders** screen with a MVC form which contains the following controls:

Control Name	Control Type	Notes
Packaged Order Details	Heading	Main Heading of the screen.
Prepare Order	Button	Navigates the user to the Prepare Order screen.
Order Number	Table Heading	Populated with the OrderId reference number retrieved for each packaged order.
Product Name	Table Heading	Populated with the ProductName attribute value of each product which has been retrieved for each packaged order.
Product Quantity	Table Heading	The quantity of each product which has been packaged for each order retrieved.
Transaction Date	Table Heading	The date which each

					package order transaction was captured.						
	Created By	Table Heading	The name of the user which has effected the package order transaction. Populated with ActorName attribute value.								
	Back	Button	Navigates the user to the previous page if clicked.								
	Step 3: The user wishes to search for a specific order number to view all the associated products which have been packaged for the order and to confirm whether all the necessary products of the order have been prepared.		Step 4: The system will filter all transactions from the OrderProduct table and display only those package order transactions which have matched search criteria entered. [ALT]								
	Step 5: The system user selects Prepare Order button. [ALT]		Step 6: The system retrieves the following details from the listed entities below using a .NET Controller with a SQL Read query: From the Product table: <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Prodduct Name</td><td>ProductName</td><td>Name of each product defined on the system.</td></tr></table>			Details to be read	Attributes in the table	Notes	Prodduct Name	ProductName	Name of each product defined on the system.
Details to be read	Attributes in the table	Notes									
Prodduct Name	ProductName	Name of each product defined on the system.									

		From the Order table: <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Order Number</td><td>OrderId</td><td>Unique order reference number for each order placed on the system.</td></tr></table> The system then displays the Capture Prepared Order screen containing the following form details which prompts the user to capture the required details pertaining to the order to be prepared: <table><tr><th>Details to be Displayed</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Capture Prepared Order</td><td>Heading</td><td>The main heading of the screen.</td></tr><tr><td>Product Name</td><td>ProductName</td><td>The user is prompted to select a product from the options provided.</td></tr><tr><td>Order Number</td><td>OrderId</td><td>User is prompted to select a specific order.</td></tr><tr><td>Quantity</td><td>ProductQuantity</td><td>User is prompted to specify the quantity of the product to be packaged.</td></tr><tr><td>Cancel</td><td>Button</td><td>Navigates the user to the Packaged Orders screen if clicked.</td></tr><tr><td>Capture</td><td>Button</td><td>Submits the captured form details if clicked.</td></tr></table>	Details to be read	Attributes in the table	Notes	Order Number	OrderId	Unique order reference number for each order placed on the system.	Details to be Displayed	Attributes in the table	Notes	Capture Prepared Order	Heading	The main heading of the screen.	Product Name	ProductName	The user is prompted to select a product from the options provided.	Order Number	OrderId	User is prompted to select a specific order.	Quantity	ProductQuantity	User is prompted to specify the quantity of the product to be packaged.	Cancel	Button	Navigates the user to the Packaged Orders screen if clicked.	Capture	Button	Submits the captured form details if clicked.
	Details to be read	Attributes in the table	Notes																										
	Order Number	OrderId	Unique order reference number for each order placed on the system.																										
	Details to be Displayed	Attributes in the table	Notes																										
	Capture Prepared Order	Heading	The main heading of the screen.																										
	Product Name	ProductName	The user is prompted to select a product from the options provided.																										
	Order Number	OrderId	User is prompted to select a specific order.																										
	Quantity	ProductQuantity	User is prompted to specify the quantity of the product to be packaged.																										
	Cancel	Button	Navigates the user to the Packaged Orders screen if clicked.																										
	Capture	Button	Submits the captured form details if clicked.																										
Step 7: The user selects and enters all the required details	Step 8: The system validates the captured details based on the following validation requirements of OrderProduct table:																												

	and selects the Capture button. [ALT]	<table><tr><th>Details to be Validated</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Order Number</td><td>OrderId</td><td>Required.</td></tr><tr><td>Product</td><td>ProductId</td><td>Required.</td></tr><tr><td>Product Quantity</td><td>ProductQuantity</td><td>Required.</td></tr></table> [ALT]	Details to be Validated	Attributes in the table	Notes	Order Number	OrderId	Required.	Product	ProductId	Required.	Product Quantity	ProductQuantity	Required.
	Details to be Validated	Attributes in the table	Notes											
Order Number	OrderId	Required.												
Product	ProductId	Required.												
Product Quantity	ProductQuantity	Required.												
		Step 9: The system checks whether the order was placed by a reseller to determine if there are sufficient stock on hand to package the specified quantity of the requested product selected for packaging. The following details are retrieved from the Product table using a .NET Controller with a SQL Read query: <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Quantity On Hand</td><td>QuantityOnHand</td><td>Current count of the selected product in stock.</td></tr></table> The system decreases the quantity on hand value of the selected product for packaging by subtracting the packaged product quantity from the quantity on hand value currently stored for the product. The system will save the updated quantity on hand value in the Product entity for the selected product using a .NET controller with a SQL Update method: <table><tr><th>Details to be Saved</th><th>Attributes in the table</th><th>Notes</th></tr></table>	Details to be read	Attributes in the table	Notes	Quantity On Hand	QuantityOnHand	Current count of the selected product in stock.	Details to be Saved	Attributes in the table	Notes			
Details to be read	Attributes in the table	Notes												
Quantity On Hand	QuantityOnHand	Current count of the selected product in stock.												
Details to be Saved	Attributes in the table	Notes												

			Quantity On Hand	QuantityOn Hand	Updated count of the product selected for packaging.
			[ALT] Step 10: The system also creates a new transaction record for the packaged products in the OrderProduct entity by saving the following details using a SQL Insert query:		
			Details to be created	Attributes in the table	Notes
			Package Order	Package Order Id	A unique identifier generated and assigned to each package order instance created. The system automatically increments the highest OrderId integer value within the OrderProduct entity.
			Order Number	OrderId	Unique number for each order for which products have been packaged.
			Product ID	ProductId	The name of each product which has been packaged for a specific order.
			Product Quantity	ProductQuantity	The quantity of each product which has been packaged.
			Transaction Date	DateLogged	The date which each transaction was captured.
			Created By	ActorName	The name of the user which has effected the package

					order transaction.					
			Step 11: The system displays a Bootstrap confirmation alert with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>"Product packaged successfully!"</td><td>Label</td><td>Informs the user that the package order details have been captured successfully.</td></tr></table>			Control Name	Control Type	Notes	"Product packaged successfully!"	Label
Control Name	Control Type	Notes								
"Product packaged successfully!"	Label	Informs the user that the package order details have been captured successfully.								
ALTERNATE COURSES:	ALT Step 4: The system has retrieved no package order transactions that match the search parameter captured: - The system displays a message informing the user that no results have been found. - Return to Step 3									
	ALT Step 7: The user does not wish to capture products packaged for an order anymore: The use case is terminated.									
	ALT Step 8: The captured form details is incomplete or incorrect and the system displays a validation error message informing the user of what details were incorrect or incomplete: Return to Step 7									
	ALT Step 9 a: The order belongs to a customer user and the system adds the package order details to the OrderProduct entity.									
	ALT Step 9 b: The system detects that there are insufficient stock of the selected product available and displays an alert message informing the user that the order could not be packaged: The use case concludes.									
CONCLUSION:	This use case concludes when the system displays a confirmation notification which informs the user that the selected product has been packaged successfully.									
POST-CONDITION:	A new transactional record is added to the OrderProduct entity.									
BUSINESS RULES	- Only the system administrator or sales assistant users should be permitted to capture products packaged for a specific order. - The quantity on hand for a packaged product should only be deducted for orders placed by resellers.									

IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• The user has no internet connection and are unable to capture packaged products for an order using the system. • Users should be restricted to capture packaged products if the quantity on hand is less than the packaged quantity specified. • Only one product can be packaged at a time for a particular order.
ASSUMPTIONS:	• The key stakeholders identified will first look up what products and their respective quantities have already been captured for a specific order to determine what products are still not packaged according to the system.
OPEN ISSUES:	• None

Table 91 - 8.21 Send Low Inventory Alert

INVENTORY MANAGEMENT SUBSYSTEM										
AUTHORS (s): Nomusa Vumisa		DATE: 06/06/2022								
VERSION: 1.0		LAST REVIEW DATE: 16/06/2022								
USE CASE NAME:	Send Low Inventory Alert		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒							
USE CASE ID:	8.21									
PRIORITY:	High									
SOURCE:	ByteXcom Requirements List									
PRIMARY BUSINESS ACTOR	• Time									
PRIMARY SYSTEM ACTOR	• None									
OTHER PARTICIPATING ACTORS:	• Administrator and Inventory Manager									
OTHER INTERESTED STAKEHOLDERS:	• None									
DESCRIPTION:	The use case begins when the quantity on hand has reached the specified threshold. The system then sends an email to the system administrator and inventory manager.									
PRE-CONDITION:	The threshold value has been reached									
TRIGGER:	This use case is called when the quantity on hand has reached the specified threshold.									
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE								
		MANUAL ACTION	AUTOMATED ACTION							
	Step 1: The threshold value has been reached		Step 2: The system retrieves the following attributes from the Inventory Item Table:							
			<table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Quantity On Hand</td><td>QuantityOnHand</td></tr><tr><td>Threshold Value</td><td>ThresholdValue</td></tr></table>		Details to be read	Attributes in the table	Quantity On Hand	QuantityOnHand	Threshold Value	ThresholdValue
			Details to be read	Attributes in the table						
			Quantity On Hand	QuantityOnHand						
Threshold Value	ThresholdValue									
And verifies if the quantity on hand has reached the specified threshold										
Step 3: The system generates a low inventory email with the following details:										

			<table><tr><th>Details</th><th>Notes</th></tr><tr><td>To</td><td>The administrator and inventory managers email addresses – retrieved from the User Table using an SQL READ query</td></tr><tr><td>From</td><td>The company email address</td></tr><tr><td>Subject</td><td>“Low Inventory Alert”</td></tr><tr><td>Body</td><td>“The following item has reached the threshold for notification.”</td></tr><tr><td>Body - Inventory Item Name:</td><td>The name of the inventory item retrieved from the Inventory Item table using an SQL READ query</td></tr><tr><td>Body – Quantity on Hand:</td><td>The inventory item quantity on hand is retrieved from the Inventory Item table using an SQL READ query</td></tr></table>		Details	Notes	To	The administrator and inventory managers email addresses – retrieved from the User Table using an SQL READ query	From	The company email address	Subject	“Low Inventory Alert”	Body	“The following item has reached the threshold for notification.”	Body - Inventory Item Name:	The name of the inventory item retrieved from the Inventory Item table using an SQL READ query	Body – Quantity on Hand:	The inventory item quantity on hand is retrieved from the Inventory Item table using an SQL READ query
			Details	Notes														
			To	The administrator and inventory managers email addresses – retrieved from the User Table using an SQL READ query														
			From	The company email address														
			Subject	“Low Inventory Alert”														
			Body	“The following item has reached the threshold for notification.”														
			Body - Inventory Item Name:	The name of the inventory item retrieved from the Inventory Item table using an SQL READ query														
	Body – Quantity on Hand:	The inventory item quantity on hand is retrieved from the Inventory Item table using an SQL READ query																
Step 4: The system sends the alert email to the administrator and Inventory manager using the Sendgrid API.																		
ALTERNATIVE STEPS:																		
CONCLUSION:	This use case concludes when alert has been sent																	
POST-CONDITION:	The administrator and inventory manager are informed of the stock levels																	
BUSINESS RULES	• None																	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• Only the system administrator and inventory manager roles should be notified of the inventory levels. • Alerts must be sent via email.																	
ASSUMPTIONS:	• None																	
OPEN ISSUES:	• None																	

Table 92 - 8.22 Send Inventory Out of Stock Alert

INVENTORY MANAGEMENT SUBSYSTEM				
AUTHORS (s): Nomusa Vumisa		DATE: 06/06/2022		
VERSION: 1.0		LAST REVIEW DATE: 06/06/2022		
USE CASE NAME:	Send Inventory Out of Stock Alert	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒		
USE CASE ID:	8.22			
PRIORITY:	High			
SOURCE:	ByteXcom Requirements List			
PRIMARY BUSINESS ACTOR	• Time			
PRIMARY SYSTEM ACTOR	• None			
OTHER PARTICIPATING ACTORS:	• Administrator and Inventory Manager			
OTHER INTERESTED STAKEHOLDERS:	• None			
DESCRIPTION:	The use case begins when the quantity on hand is out of stock. The system then sends an email to the system administrator and inventory manager.			
PRE-CONDITION:	The quantity on hand has run out			
TRIGGER:	This use case is called when the quantity on hand is out of stock.			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE		
		MANUAL ACTION	AUTOMATED ACTION	
	Step 1: The quantity on hand has run out		Step 2: The system retrieves the following attributes from the Inventory Item Table:	
			Details to be read	Attributes in the table
			Quantity On Hand	QuantityOnHand
			And verifies if the quantity on hand has run out	
			Step 3: The system generates an out-of-stock email with the following details:	
			Details	Notes
			To	The administrator and inventory managers email addresses – retrieved from the User Table using an SQL READ query
			From	The company email address
Subject	“Out of Stock Alert”			

			Body	"The following item has reached the threshold for notification."	
			Body - Inventory Item Name:	The name of the inventory item retrieved from the Inventory Item table using an SQL READ query	
			Body – Quantity on Hand:	The inventory item quantity on hand is retrieved from the Inventory Item table using an SQL READ query	
				Step 4: The system sends the alert email to the administrator and Inventory manager using the Sendgrid API.	
ALTERNATIVE STEPS:					
CONCLUSION:	This use case concludes when alert has been sent				
POST-CONDITION:	The administrator and inventory manager are informed of the stock levels				
BUSINESS RULES	• None				
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• Only the system administrator and inventory manager roles should be notified of the inventory levels. • Alerts must be sent via email.				
ASSUMPTIONS:	• None				
OPEN ISSUES:	• None				

2.9. Location Subsystem

Table 93 - 9.1 Create Country

LOCATION SUBSYSTEM			
AUTHORS (s): Ofhani Mungani		DATE: 18/04/2022	
VERSION: 1.0		LAST REVIEW DATE: 21/08/2022	
USE CASE NAME:	Create Country	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒	
USE CASE ID:	9.1		
PRIORITY:	High		
SOURCE:	ByteXcom Requirements List		
PRIMARY BUSINESS ACTOR	• Administrator		
PRIMARY SYSTEM ACTOR	• None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	This use case describes the process where the administrator wishes to add a new country onto the system. The use case begins with the administrator wishing to add a country. The administrator will capture the country’s details and save the information on the Country entity. The use case concludes when the system alerts the administrator with “Country successfully added” message.		
PRE-CONDITION:	• The administrator must be logged onto the system.		
TRIGGER:	The use case is triggered when the administrator wants to add a new Country.		
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE	
		MANUAL ACTION	AUTOMATED ACTION

Step 1: The administrator navigates to the **Country List Screen** under the Countries tab on the side bar

Step 2: The system retrieves the country information from the following attributes and tables using a method in a .NET Controller and an SQL READ Query:

From the **Country** table:

Details to be read	Attributes in the table
Name	CountryName

And displays the **Country Screen** with the following controls:

Control Name	Control Type	Notes
Country List	Heading	Page main heading
Question Mark Icon	Button	When clicked, this control will open a pdf which contains the help document
Create New Country	Button	When clicked, this control will direct the user to the Create Country page
Search bar	Textbox	This control will be used when the user inserts the search parameters
Search icon	Button	This control will be used to activate the search
Results table	Table	This control will be used to display all the product categories that exist on the system
Country Name	Label	This control will be used as a heading to display the Name for all countries
Action - Update	Button	This control will be placed in a row, with the delete button,

					under the “Action” heading for each entry in the Supplier table. The button will direct the user to the Update Country screen
			Action - Delete	Button	This control will be placed in a row, with the delete button, under the “Action” heading for each entry in the Supplier table. The button will direct the user to the Delete Country Details screen
			Showing “” to “” of “” entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database
			“Previous” and “Next”	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on
	Step 3: The system administrator clicks		Step 4: The system displays the Create Country screen with the following controls:		

	on the Create New Country button		<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Create Country</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Country information</td><td>Heading</td><td>Page subheading</td></tr><tr><td>Name</td><td>Label</td><td>This control will be used as a label to indicate the Name attribute of the Country record</td></tr><tr><td>Name</td><td>Textbox</td><td>This control will be used by the administrator to insert a name of the Country</td></tr><tr><td>Create</td><td>Button</td><td>This control will be used to confirm the addition of the Country</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the addition of the Country</td></tr></table> Note: All controls are enabled by default unless stated otherwise.	Control Name	Control Type	Notes	Create Country	Heading	Page main heading	Country information	Heading	Page subheading	Name	Label	This control will be used as a label to indicate the Name attribute of the Country record	Name	Textbox	This control will be used by the administrator to insert a name of the Country	Create	Button	This control will be used to confirm the addition of the Country	Cancel	Button	This control will be used to cancel the addition of the Country
	Control Name	Control Type	Notes																					
	Create Country	Heading	Page main heading																					
Country information	Heading	Page subheading																						
Name	Label	This control will be used as a label to indicate the Name attribute of the Country record																						
Name	Textbox	This control will be used by the administrator to insert a name of the Country																						
Create	Button	This control will be used to confirm the addition of the Country																						
Cancel	Button	This control will be used to cancel the addition of the Country																						
	Step 5: The system administrator enters the details and clicks on the Create button [ALT]		Step 6: The system captures and validates the entered details against the Country table [ALT] <table><tr><th>Details to be validated</th><th>Attributes in the Country table</th><th>Notes</th></tr><tr><td>Name</td><td>CountryName</td><td>Required, Unique</td></tr></table>	Details to be validated	Attributes in the Country table	Notes	Name	CountryName	Required, Unique															
Details to be validated	Attributes in the Country table	Notes																						
Name	CountryName	Required, Unique																						
			Step 7: The system uses an SQL READ query to check if the country already exists in the Country table using the following attributes [ALT] <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Name</td><td>CountryName</td></tr></table>	Details to be read	Attributes in the table	Name	CountryName																	
Details to be read	Attributes in the table																							
Name	CountryName																							

			Step 8: The system saves the information into the Country table using an INSERT SQL query: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Id</td><td>Id</td><td>A unique value generated by the system and saved to the database.</td></tr><tr><td>Name</td><td>CountryName</td><td>Required</td></tr></table>	Details to be saved	Attributes in the table	Notes	Id	Id	A unique value generated by the system and saved to the database.	Name	CountryName	Required
	Details to be saved	Attributes in the table	Notes									
Id	Id	A unique value generated by the system and saved to the database.										
Name	CountryName	Required										
			Step 9: The system redirects the system administrator to the Country List Screen and displays a message box with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>“Country Created Successfully”</td><td>Label</td><td>This control is used to indicate a successful creation</td></tr><tr><td>“Click here to add a new province”</td><td>Label</td><td>Suggestion link to create a new province</td></tr></table>	Control Name	Control Name	Notes	“Country Created Successfully”	Label	This control is used to indicate a successful creation	“Click here to add a new province”	Label	Suggestion link to create a new province
Control Name	Control Name	Notes										
“Country Created Successfully”	Label	This control is used to indicate a successful creation										
“Click here to add a new province”	Label	Suggestion link to create a new province										
ALTERNATE COURSES:	ALT Step 5: The administrator does not want to add a country anymore • The administrator clicks on the cancel button • The use case is terminated											
	ALT Step 6: The details entered are not in the correct format or the required fields are empty • The system displays a validation error message box with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Name</td><td>Textbox</td><td>This control will be outlined in red to show a validation error</td></tr><tr><td>Name is required</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Name textbox</td></tr></table>			Control Name	Control Name	Notes	Name	Textbox	This control will be outlined in red to show a validation error	Name is required	Label	This control will be used as a label to indicate a validation error in the Name textbox
	Control Name	Control Name	Notes									
Name	Textbox	This control will be outlined in red to show a validation error										
Name is required	Label	This control will be used as a label to indicate a validation error in the Name textbox										
Return to step 5												

	ALT Step 7: The country already exists on the system - The system displays an error modal dialogue with the following controls		
	Control Name	Control Name	Notes
	Country already exists!	Label	This control is used to inform the administrator that the supplier already exists
	Okay	Button	This control is used to cancel the creation of the country
	- The administrator clicks the “Okay” button - The use case is terminated		
CONCLUSION:	This use case concludes when a country has been added on the system and the system displays an alert message to confirm the addition of the country’s details.		
POST-CONDITION:	The new country’s details has been added to the Country table in the database.		
BUSINESS RULES	Only the administrator has access to this function.		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	- The system is offline, and the system administrator will not be able to add a new country. - No duplicate [CountryName] attribute values may exist at any time.		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

Table 94 - 9.2 Search Country

LOCATION SUBSYSTEM							
AUTHORS (s): Ofhani Mungani		DATE: 18/04/2022					
VERSION: 1.0		LAST REVIEW DATE: 21/08/2022					
USE CASE NAME:	Search Country		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒				
USE CASE ID:	9.2						
PRIORITY:	Medium						
SOURCE:	ByteXcom Requirements List						
PRIMARY BUSINESS ACTOR	• Administrator						
PRIMARY SYSTEM ACTOR	• None						
OTHER PARTICIPATING ACTORS:	• None						
OTHER INTERESTED STAKEHOLDERS:	• None						
DESCRIPTION:	This use case describes the process where the administrator would like to search for a country onto the system. The use case begins with the administrator wishing to search for a country. The system will prompt the administrator to enter a country’s name the administrator wishes to search for. The system will search through the Country entity to find the instance value in the CountryName attribute which match the name provided by the administrator. The use case concludes when the system displays the results of the search query.						
PRE-CONDITION:	• The administrator must be logged onto the system.						
TRIGGER:	This use case is called when the administrator wants to search for a Country on the system.						
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE					
		MANUAL ACTION	AUTOMATED ACTION				
	Step 1: The administrator wishes to search for a Country on the system.						
	Step 2: The administrator navigates to the Country List Screen under the Country tab on the side bar		Step 3: The system retrieves following information using an SQL READ QUERY From the Country table: <table><tr><td>Details to be read</td><td>Attributes in the table Notes</td></tr><tr><td>Name</td><td>CountryName</td></tr></table>	Details to be read	Attributes in the table Notes	Name	CountryName
	Details to be read	Attributes in the table Notes					
Name	CountryName						

			And displays the Country Screen with the following controls:		
			Control Name	Control Type	Notes
			Country List	Heading	Page main heading
			Question Mark Icon	Button	When clicked, this control will open a pdf which contains the help document
			Create New Country	Button	When clicked, this control will direct the user to the Add Country page
			Search bar	Textbox	This control will be used when the user inserts the search parameters
			Search icon	Button	This control will be used to activate the search
			Results table	Table	This control will be used to display all the product categories that exist on the system
			Country Name	Label	This control will be used as a heading to display the Name for all countries
			Action	Table Heading	This control will be used as a heading to display the actions the administrator can perform with each entry – Edit and Delete
Action - Update	Button	This control will be placed in a row, with the delete button, under the “Action” heading for each entry in the Supplier			

					table. The button will direct the user to the Update Country screen
			Action - Delete	Button	This control will be placed in a row, with the delete button, under the "Action" heading for each entry in the Supplier table. The button will direct the user to the Delete Country Details screen
			Showing "" to "" of "" entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database
			"Previous" and "Next"	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on
	Step 4: The administrator enters their search query into the search textbox.		Step 5: The system searches the Country table and displays the Country(s) matching the search query in the Country Screen [ALT]		
ALT Step 5: The search query entered does not match any records existing in the Country entity.					

	The system displays a “No Records Found” message in the Country Screen and on the results table.
CONCLUSION:	This use case concludes when all matched country details are displayed on the screen.
POST-CONDITION:	All country records that matched the search query are retrieved from the Country entity.
BUSINESS RULES	Only the administrator has access to this function.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	The system should indicate if no results were found for the search query specified.
ASSUMPTIONS:	None
OPEN ISSUES:	None

Table 95 - 9.3 Update Country

LOCATION SUBSYSTEM					
AUTHORS (s): Ofhani Mungani		DATE: 18/04/2022			
VERSION: 1.0		LAST REVIEW DATE: 21/08/2022			
USE CASE NAME:	Update Country	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒			
USE CASE ID:	9.3				
PRIORITY:	Medium				
SOURCE:	ByteXcom Requirements List				
PRIMARY BUSINESS ACTOR	• Administrator				
PRIMARY SYSTEM ACTOR	• None				
OTHER PARTICIPATING ACTORS:	• None				
OTHER INTERESTED STAKEHOLDERS:	• None				
DESCRIPTION:	This use case describes the process where the administrator would like to update a country's details onto the system. The use case begins with the administrator requesting to update a country name. The administrator will select an option to search a country and the system will invoke use case 9.2 Search Country to retrieve the county's name. The administrator will select and edit the name of the country and save it onto the system database on the Country entity. The use case concludes when the system displays the country successfully updated message to the administrator.				
PRE-CONDITION:	• The administrator must be logged onto the system.				
TRIGGER:	The use case is triggered when the administrator wants to update a Country.				
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE			
		MANUAL ACTION	AUTOMATED ACTION		
	Step 1: The administrator wishes to update a country's name.		Step 2: The system invokes use case 9.2 Search Country		
	Step 2: The administrator clicks on the Update button		Step 3: The system retrieves the selected Country's information from the following attributes and tables using a method in a .NET Controller and an SQL READ Query: From the Country table:		
	<table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Name</td><td>CountryName</td></tr></table>		Details to be read	Attributes in the table	Name
Details to be read	Attributes in the table				
Name	CountryName				

		And displays the Update Country Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Update Country</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Country information</td><td>Heading</td><td>Page subheading</td></tr><tr><td>Name</td><td>Labe</td><td>This control will be used as a label to indicate the Name attribute of the Country record</td></tr><tr><td>Name</td><td>Textbox</td><td>This control will be used by the administrator to update a name of the Country</td></tr><tr><td>Update</td><td>Button</td><td>This control will be used to confirm the addition of the Country</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the edit of the Country</td></tr></table> Note: All controls are enabled by default unless stated otherwise.	Control Name	Control Type	Notes	Update Country	Heading	Page main heading	Country information	Heading	Page subheading	Name	Labe	This control will be used as a label to indicate the Name attribute of the Country record	Name	Textbox	This control will be used by the administrator to update a name of the Country	Update	Button	This control will be used to confirm the addition of the Country	Cancel	Button	This control will be used to cancel the edit of the Country
	Control Name	Control Type	Notes																				
	Update Country	Heading	Page main heading																				
	Country information	Heading	Page subheading																				
	Name	Labe	This control will be used as a label to indicate the Name attribute of the Country record																				
	Name	Textbox	This control will be used by the administrator to update a name of the Country																				
	Update	Button	This control will be used to confirm the addition of the Country																				
	Cancel	Button	This control will be used to cancel the edit of the Country																				
	Step 4: The system administrator updates the Country information and clicks on the Update button [ALT]	Step 5: The system captures and validates the entered details against the Country table [ALT] <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Name</td><td>CountryName</td><td>Required, Unique,</td></tr></table>	Details to be read	Attributes in the table	Notes	Name	CountryName	Required, Unique,															
	Details to be read	Attributes in the table	Notes																				
Name	CountryName	Required, Unique,																					
	Step 6: The system uses an SQL READ query to check if the country already exists in the Country table using the following attributes [ALT] <table><tr><th>Details to be updated</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Name</td><td>ReturnReasonName</td><td></td></tr></table>	Details to be updated	Attributes in the table	Notes	Name	ReturnReasonName																	
Details to be updated	Attributes in the table	Notes																					
Name	ReturnReasonName																						

		Step 7: The system displays a message box with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Confirmation required</td><td>Label</td><td>This control is used to display the type of message box displayed</td></tr><tr><td>Confirm updated Country details?</td><td>Label</td><td>This control is used to request confirmation on the updated Country details</td></tr><tr><td>Confirm</td><td>Button</td><td>This control is used to confirm the updating of the Country details</td></tr><tr><td>Cancel</td><td>Button</td><td>This control is used to close the message box for the user to edit the Country details</td></tr></table>	Control Name	Control Type	Notes	Confirmation required	Label	This control is used to display the type of message box displayed	Confirm updated Country details?	Label	This control is used to request confirmation on the updated Country details	Confirm	Button	This control is used to confirm the updating of the Country details	Cancel	Button	This control is used to close the message box for the user to edit the Country details
	Control Name	Control Type	Notes														
	Confirmation required	Label	This control is used to display the type of message box displayed														
	Confirm updated Country details?	Label	This control is used to request confirmation on the updated Country details														
	Confirm	Button	This control is used to confirm the updating of the Country details														
Cancel	Button	This control is used to close the message box for the user to edit the Country details															
Step 8: The system administrator clicks on the Confirm button [ALT]	Step 9: The system saves the information into the Country table using an UPDATE SQL query: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Name</td><td>CountryName</td><td>Required, Unique,</td></tr></table>	Details to be saved	Attributes in the table	Notes	Name	CountryName	Required, Unique,										
Details to be saved	Attributes in the table	Notes															
Name	CountryName	Required, Unique,															

			Step 10: The system redirects the system administrator to the Country Screen and displays a message box with the following controls:									
			<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Checkbox</td><td>Icon</td><td>This control is used to indicate a successful update</td></tr><tr><td>"Country Created Successfully"</td><td>Label</td><td>This control is used to indicate a successful update</td></tr></table>	Control Name	Control Name	Notes	Checkbox	Icon	This control is used to indicate a successful update	"Country Created Successfully"	Label	This control is used to indicate a successful update
	Control Name	Control Name	Notes									
	Checkbox	Icon	This control is used to indicate a successful update									
"Country Created Successfully"	Label	This control is used to indicate a successful update										
ALTERNATE COURSES:												
ALT Step 4: The administrator does not want to update the Country anymore - The administrator clicks on the cancel button - The use case is terminated												
ALT Step 5: The details entered are not in the correct format or the required fields are empty The system displays a validation error message box with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Name</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>Name is required</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Name textbox</td></tr></table>			Control Name	Control Name	Notes	Name	Textbox	This control will be outline in red to show a validation error	Name is required	Label	This control will be used as a label to indicate a validation error in the Name textbox	
Control Name	Control Name	Notes										
Name	Textbox	This control will be outline in red to show a validation error										
Name is required	Label	This control will be used as a label to indicate a validation error in the Name textbox										
ALT Step 6: The Country already exists on the system - The system displays an error modal dialogue with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Country already exists!</td><td>Label</td><td>This control is used to inform the administrator that the supplier already exists</td></tr><tr><td>Okay</td><td>Button</td><td>This control is used to cancel the creation of the supplier</td></tr></table> - The administrator clicks the "Okay" button - The use case is terminated			Control Name	Control Name	Notes	Country already exists!	Label	This control is used to inform the administrator that the supplier already exists	Okay	Button	This control is used to cancel the creation of the supplier	
Control Name	Control Name	Notes										
Country already exists!	Label	This control is used to inform the administrator that the supplier already exists										
Okay	Button	This control is used to cancel the creation of the supplier										
ALT Step 8: The administrator clicks on the cancel button The use case is terminated												
CONCLUSION:	This use case concludes when a Country has been added on the system											
POST-CONDITION:	The selected country's details has been updated in the Country table and saved to the database.											

BUSINESS RULES	• Only the administrator has access to this function.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• No duplicate country names may exist for different country records at any time.
ASSUMPTIONS:	• The system is online and the administrator has a stable internet connection.
OPEN ISSUES:	• None

Table 96 - 9.4 Delete Country

LOCATION SUBSYSTEM							
AUTHORS (s): Ofhani Mungani		DATE: 18/04/2022					
VERSION: 1.0		LAST REVIEW DATE: 21/08/2022					
USE CASE NAME:	Delete Country	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒					
USE CASE ID:	9.4						
PRIORITY:	Medium						
SOURCE:	ByteXcom Requirements List						
PRIMARY BUSINESS ACTOR	• Administrator						
PRIMARY SYSTEM ACTOR	• None						
OTHER PARTICIPATING ACTORS:	• None						
OTHER INTERESTED STAKEHOLDERS:	• None						
DESCRIPTION:	This use case narrative describes the process in which the administrator wishes to delete a country from the system. The administrator will search for the required country selected for deletion. The administrator will then confirm the deletion of the country and the use case concludes when the system notifies the administrator of a successful deletion.						
PRE-CONDITION:	• The administrator must be logged onto the system.						
TRIGGER:	The use case is triggered when the administrator wants to delete a Country.						
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE					
		MANUAL ACTION	AUTOMATED ACTION				
	Step 1: The administrator wishes to delete a Country		Step 2: The system invokes use case 4.6 Search Country				
	Step 3: The administrator clicks on the Delete icon		Step 4: The system retrieves the Country information from the following attributes and tables using a method in a .NET Controller and an SQL READ Query: From the Country table: <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Name</td><td>CountryName</td></tr></table> And displays the Delete Country Screen with the following controls:	Details to be read	Attributes in the table	Name	CountryName
	Details to be read	Attributes in the table					
Name	CountryName						

			<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Delete Country</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Name</td><td>Label</td><td>This control will be used as a label to indicate the Name attribute of the Country record</td></tr><tr><td>Name</td><td>Label</td><td>This control will be used by the administrator to display the name of the Country</td></tr><tr><td>Delete</td><td>Button</td><td>This control will be used to confirm the addition of the Country</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the addition of the Country</td></tr></table>	Control Name	Control Type	Notes	Delete Country	Heading	Page main heading	Name	Label	This control will be used as a label to indicate the Name attribute of the Country record	Name	Label	This control will be used by the administrator to display the name of the Country	Delete	Button	This control will be used to confirm the addition of the Country	Cancel	Button	This control will be used to cancel the addition of the Country
	Control Name	Control Type	Notes																		
	Delete Country	Heading	Page main heading																		
	Name	Label	This control will be used as a label to indicate the Name attribute of the Country record																		
	Name	Label	This control will be used by the administrator to display the name of the Country																		
	Delete	Button	This control will be used to confirm the addition of the Country																		
	Cancel	Button	This control will be used to cancel the addition of the Country																		
	Step 5: The administrator clicks on the Delete button [ALT]		Step 6: The system displays a message box with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Confirmation required</td><td>Label</td><td>This control is used to display the type of message box displayed</td></tr><tr><td>Are you sure you want to delete this Country?</td><td>Label</td><td>This control is used to request confirmation of the deletion of the Country</td></tr><tr><td>Confirm</td><td>Button</td><td>This control is used to confirm the deletion of the Country details</td></tr><tr><td>Cancel</td><td>Button</td><td>This control is used to close the message</td></tr></table>	Control Name	Control Type	Notes	Confirmation required	Label	This control is used to display the type of message box displayed	Are you sure you want to delete this Country?	Label	This control is used to request confirmation of the deletion of the Country	Confirm	Button	This control is used to confirm the deletion of the Country details	Cancel	Button	This control is used to close the message			
	Control Name	Control Type	Notes																		
	Confirmation required	Label	This control is used to display the type of message box displayed																		
	Are you sure you want to delete this Country?	Label	This control is used to request confirmation of the deletion of the Country																		
	Confirm	Button	This control is used to confirm the deletion of the Country details																		
Cancel	Button	This control is used to close the message																			

					box deletion of the Country								
	Step 7: The system administrator clicks on the Confirm button [ALT]		Step 8: The system deletes the selected Country from the Country entity in the database using a method in a .NET controller and an SQL DELETE query. [ALT]										
			<table><tr><th>Details to be deleted</th><th>Attributes in the table</th></tr><tr><td>Id</td><td>Id</td></tr><tr><td>Name</td><td>CountryName</td></tr></table>			Details to be deleted	Attributes in the table	Id	Id	Name	CountryName		
	Details to be deleted	Attributes in the table											
Id	Id												
Name	CountryName												
		Step 9: The system redirects the system administrator to the Country screen and displays a message box with the following controls:											
			<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Checkbox</td><td>Icon</td><td>This control is used to indicate a successful deletion</td></tr><tr><td>"Country Deleted Successfully"</td><td>Label</td><td>This control is used to indicate a successful deletion</td></tr></table>	Control Name	Control Name	Notes	Checkbox	Icon	This control is used to indicate a successful deletion	"Country Deleted Successfully"	Label	This control is used to indicate a successful deletion	
Control Name	Control Name	Notes											
Checkbox	Icon	This control is used to indicate a successful deletion											
"Country Deleted Successfully"	Label	This control is used to indicate a successful deletion											
ALTERNATE COURSES:	ALT Step 5: The administrator does not want to delete the Country anymore - The administrator clicks on the cancel button - The use case is terminated												
	ALT Step 7: The administrator does not want to confirm to delete the Country anymore Return to step 5												
	ALT Step 8: The Country cannot be deleted as it has an association with either the Province, User or Order entities: The system redirects the administrator and displays a bootstrap alert notification with the following controls:												
	<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"Country cannot be deleted"</td><td>Label</td><td>This control is used to indicate an unsuccessful deletion</td></tr></table> The use case is terminated					Control Name	Control Name	Notes	"Country cannot be deleted"	Label	This control is used to indicate an unsuccessful deletion		
Control Name	Control Name	Notes											
"Country cannot be deleted"	Label	This control is used to indicate an unsuccessful deletion											
CONCLUSION:	This use case concludes when the selected country's details has been deleted from the system and a confirmation alert is displayed to confirm the country's successful removal.												
POST-CONDITION:	The selected country's details has been removed from the Country entity and the changes are saved to the system's database.												

BUSINESS RULES	• Only the administrator has access to this function.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• A country cannot be deleted if a province is associated to it. • A country cannot be deleted if it is associated to a user's profile details. • A country cannot be deleted if it is associated to an order.
ASSUMPTIONS:	• The system is online and the administrator has a stable internet connection.
OPEN ISSUES:	• None

Table 97 - 9.5 Add Province

LOCATION SUBSYSTEM										
AUTHORS (s): APIWE SHOZI			DATE: 22/08/2022							
VERSION: 1.0			LAST REVIEW DATE: 22/08/2022							
USE CASE NAME:	Create Province		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒							
USE CASE ID:	9.5									
PRIORITY:	High									
SOURCE:	ByteXcom Requirements List									
PRIMARY BUSINESS ACTOR	• Administrator									
PRIMARY SYSTEM ACTOR	• None									
OTHER PARTICIPATING ACTORS:	• None									
OTHER INTERESTED STAKEHOLDERS:	• None									
DESCRIPTION:	This use case describes the process in which the administrator wishes to add a new Province to the system as one of the locations that Natuurlik offers its Provinces and services to. The use case begins when the administrator adds the new Province under its respective Suburb which as the two entities are linked via the SuburbID_FK. The administrator will add all of the pertinent information regarding the new Province and save it to the Province entity in the database.									
PRE-CONDITION:	• The administrator must be logged onto the system. • A country must be defined on the system to which the province can be assigned.									
TRIGGER:	The use case is triggered when the administrator wants to add a new province.									
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE								
		MANUAL ACTION	AUTOMATED ACTION							
	Step 1: The administrator clicks on the Province nav link		Step 2: The system retrieves the following attributes from each entry in the Province Table, using a method in a .NET controller and an SQL READ query:							
			<table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Name</td><td>ProvinceName</td></tr><tr><td>Country Name</td><td>CountryName</td></tr></table>		Details to be read	Attributes in the table	Name	ProvinceName	Country Name	CountryName
			Details to be read	Attributes in the table						
			Name	ProvinceName						
Country Name	CountryName									
and displays them on the Province Screen with the following controls:										
<table><tr><td>Control Name</td><td>Control</td><td>Notes</td></tr></table>		Control Name	Control	Notes						
Control Name	Control	Notes								

				Type	
			Province	Heading	Page main heading
			Add	Button	When clicked, this control will direct the user to the Add province record page
			Search bar	Textbox	This control will be used when the user inserts the search parameters
			Search icon	Button	This control will be used to activate the search
			Results table	Table	This control will be used to display all the Provinces that exist on the system
			Name	Label	This control will be used as a heading to display the Name for all Provinces
			Page count	Label	This control will be used as a label to display the number of pages that the results make up and to indicate on which page the user is currently on
			Back and forward	Button	This control is used to navigate through the pages used to display the results from the table

	Step 3: The administrator clicks on the Create New Province button	Step 4: The system displays the Add Province screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Add Province</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Province information</td><td>Heading</td><td>Page subheading</td></tr><tr><td>Name</td><td>Label</td><td>This control will be used as a label to indicate the Name attribute of the Province record</td></tr><tr><td>Name</td><td>Textbox</td><td>This control will be used by the administrator to insert a name of the Province</td></tr><tr><td>Save</td><td>Button</td><td>This control will be used to confirm the addition of the Province</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the addition of the Province</td></tr></table> Note: All controls are enabled by default, unless otherwise stated. All text input fields are empty by default, unless stated otherwise.	Control Name	Control Type	Notes	Add Province	Heading	Page main heading	Province information	Heading	Page subheading	Name	Label	This control will be used as a label to indicate the Name attribute of the Province record	Name	Textbox	This control will be used by the administrator to insert a name of the Province	Save	Button	This control will be used to confirm the addition of the Province	Cancel	Button	This control will be used to cancel the addition of the Province
	Control Name	Control Type	Notes																				
	Add Province	Heading	Page main heading																				
Province information	Heading	Page subheading																					
Name	Label	This control will be used as a label to indicate the Name attribute of the Province record																					
Name	Textbox	This control will be used by the administrator to insert a name of the Province																					
Save	Button	This control will be used to confirm the addition of the Province																					
Cancel	Button	This control will be used to cancel the addition of the Province																					
	Step 5: The system administrator enters the details and clicks on the create button [ALT]	Step 6: The system captures the information provided by the administrator and validates the information using jQuery validation ensuring the format is adhered to and no required text fields are left empty. [ALT]																					
		Step 7: The system uses an SQL READ query to check if the province already exists in the Province table using the following attribute. [ALT] <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Name</td><td>ProvinceName</td></tr></table>	Details to be read	Attributes in the table	Name	ProvinceName																	
Details to be read	Attributes in the table																						
Name	ProvinceName																						

			Step 8: The system saves the details in the Province table using a method in a .NET controller and an SQL INSERT query: <table><tr><th>Details to be created</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>ID</td><td>ID</td><td>Generated by the system</td></tr><tr><td>Name</td><td>ProvinceName</td><td>Required Alphabetical characters only Minimum: 1 Maximum: 50</td></tr><tr><td>Country Name</td><td>CountryName</td><td>Required Alphabetical characters only Minimum: 1 Maximum: 50</td></tr><tr><td>Country ID</td><td>Country_Id</td><td>Required FK from Country table</td></tr></table>	Details to be created	Attributes in the table	Notes	ID	ID	Generated by the system	Name	ProvinceName	Required Alphabetical characters only Minimum: 1 Maximum: 50	Country Name	CountryName	Required Alphabetical characters only Minimum: 1 Maximum: 50	Country ID	Country_Id	Required FK from Country table
	Details to be created	Attributes in the table	Notes															
ID	ID	Generated by the system																
Name	ProvinceName	Required Alphabetical characters only Minimum: 1 Maximum: 50																
Country Name	CountryName	Required Alphabetical characters only Minimum: 1 Maximum: 50																
Country ID	Country_Id	Required FK from Country table																
			Step 9: The system directs the administrator to the Province Screen and displays a bootstrap alert notification with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Successful creation</td><td>Label</td><td>This control is used to indicate a successful creation</td></tr><tr><td>"X"</td><td>Button</td><td>This control is used to close the successful creation message box</td></tr></table>	Control Name	Control Name	Notes	Successful creation	Label	This control is used to indicate a successful creation	"X"	Button	This control is used to close the successful creation message box						
Control Name	Control Name	Notes																
Successful creation	Label	This control is used to indicate a successful creation																
"X"	Button	This control is used to close the successful creation message box																
ALTERNATE COURSES:	ALT Step 5: The administrator does not want to add a Province anymore - The administrator clicks on the cancel button - The use case is terminated																	
	ALT Step 6: The details entered are not in the correct format or the required fields are empty The system displays a validation error with the following controls Return to step 5																	
	ALT Step 7: The Province already exists on the system The system displays an error modal dialogue with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Province already exists!</td><td>Label</td><td>This control is used to inform the administrator that the province already exists</td></tr><tr><td>Okay</td><td>Button</td><td>This control is used to cancel the creation of the province</td></tr></table>			Control Name	Control Name	Notes	Province already exists!	Label	This control is used to inform the administrator that the province already exists	Okay	Button	This control is used to cancel the creation of the province						
	Control Name	Control Name	Notes															
Province already exists!	Label	This control is used to inform the administrator that the province already exists																
Okay	Button	This control is used to cancel the creation of the province																

	- The administrator clicks the “Okay” button - The use case is terminated
CONCLUSION:	This use case concludes when a new province’s details are created and the system displays an alert which confirms the creation.
POST-CONDITION:	The province’s details are added to the Province entity and saved in the database.
BUSINESS RULES	- Only the administrator can add a new province. - A province must be associated with an existing country.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	No duplicate province names may exist at any time.
ASSUMPTIONS:	The administrator has a stable internet connection and the system is not offline.
OPEN ISSUES:	None

Table 98 - 9.6 Search Province

LOCATION SUBSYSTEM										
AUTHORS (s): APIWE SHOZI			DATE: 22/08/2022							
VERSION: 1.1			LAST REVIEW DATE: 24/05/2022							
USE CASE NAME:	Search Province		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒							
USE CASE ID:	9.6									
PRIORITY:	Medium									
SOURCE:	ByteXcom Requirements List									
PRIMARY BUSINESS ACTOR	• Administrator									
PRIMARY SYSTEM ACTOR	• None									
OTHER PARTICIPATING ACTORS:	• None									
OTHER INTERESTED STAKEHOLDERS:	• None									
DESCRIPTION:	This use case describes the process in the administrator wishes to search through the existing Provinces on the system in the Province Entity Table. The system will prompt the administrator to use the Province details in order to perform a search query and the system will retrieve the Province record that matches the inputs of the search query									
PRE-CONDITION:	The administrator must be logged onto the system.									
TRIGGER:	This use case is called when the administrator wants to search for a province on the system.									
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE								
		MANUAL ACTION	AUTOMATED ACTION							
	Step 1: The administrator clicks on the Province navigation link on the admin side navigation bar		Step 2: The system retrieves the following attributes for each entry from the Province Table, using a method in a .NET controller and an SQL READ query:							
			<table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Name</td><td>ProvinceName</td></tr><tr><td>Country Name</td><td>CountryName</td></tr></table>		Details to be read	Attributes in the table	Name	ProvinceName	Country Name	CountryName
			Details to be read	Attributes in the table						
			Name	ProvinceName						
Country Name	CountryName									
and displays then on the Province Screen with the following controls:										
<table><tr><td>Control Name</td><td>Control Type</td><td>Notes</td></tr><tr><td>Province</td><td>Heading</td><td>Page main heading</td></tr></table>		Control Name	Control Type	Notes	Province	Heading	Page main heading			
Control Name	Control Type	Notes								
Province	Heading	Page main heading								

			Add	Button	When clicked, this control will direct the user to the Add province record page
			Search bar	Textbox	This control will be used when the user inserts the search parameters
			Search icon	Button	This control will be used to activate the search
			Results table	Table	This control will be used to display all the Provinces that exist on the system
			Name	Label	This control will be used as a heading to display the Name for all Provinces
			Page count	Label	This control will be used as a label to display the number of pages that the results make up and to indicate on which page the user is currently on
			Back and forward	Button	This control is used to navigate through the pages used to display the results from the table
	Step 4: The administrator enters their search query into the search textbox		Step 5: The system searches the table rows in the data table and displays the Province(s) matching the search query [ALT]		
	ALT Step 5: The search query entered does not match any records existing in the Province entity. - The system displays a “No Records Found” message in the results table. - Return to step 4 or terminate the use case				
CONCLUSION:	This use case concludes when the desired province’s details are displayed on the system.				
POST-CONDITION:	The Province entity was searched for and an entry that corresponds to the search query or parameter entered.				

BUSINESS RULES	• Only the administrators can search for existing provinces on the system.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• The system is online and the administrator has a stable internet connection.
OPEN ISSUES:	• None

Table 99 - 9.7 Update Province

LOCATION SUBSYSTEM									
AUTHORS (s): APIWE SHOZI		DATE: 22/08/2022							
VERSION: 1.1		LAST REVIEW DATE: 24/05/2022							
USE CASE NAME:	Update Province		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒						
USE CASE ID:	9.7								
PRIORITY:	Medium								
SOURCE:	ByteXcom Requirements List								
PRIMARY BUSINESS ACTOR	• Administrator								
PRIMARY SYSTEM ACTOR	• None								
OTHER PARTICIPATING ACTORS:	• None								
OTHER INTERESTED STAKEHOLDERS:	• None								
DESCRIPTION:	This use case describes the process in which the administrator wishes to update the details of Province record in the Province entity. The use case begins when the administrator updates a Province that Natuurlik offers its products and services to. The administrator will update all of the pertinent information regarding the Province and the system will update the record in the Province entity table								
PRE-CONDITION:	The administrator must be logged in to the system.								
TRIGGER:	The use is triggered when the administrator wishes to update a province								
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE							
		MANUAL ACTION	AUTOMATED ACTION						
	Step 1: The administrator wishes to update a Province in the system.		Step 2: The system invokes use case 9.7 Search Province						
	Step 3: The administrator clicks on the update button		Step 4: The system retrieves the following attributes from the following tables, using a method in a .NET controller and an SQL READ query: From the Province Table: <table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Name</td><td>ProvinceName</td></tr><tr><td>Country Name</td><td>CountryName</td></tr></table>	Details to be read	Attributes in the table	Name	ProvinceName	Country Name	CountryName
	Details to be read	Attributes in the table							
Name	ProvinceName								
Country Name	CountryName								

		and displays them on the Update Province Screen which contains an MVC form with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Edit Province</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Province information</td><td>Heading</td><td>Page subheading</td></tr><tr><td>Name</td><td>Label</td><td>This control will be used as a label to indicate the Name attribute of the Province record</td></tr><tr><td>Name</td><td>Textbox</td><td>This control will be used by the administrator to update a name of the Province</td></tr><tr><td>Save</td><td>Button</td><td>This control will be used to confirm the update of the Province</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the edit of the Province</td></tr></table> Note: All controls are enabled by default, unless stated otherwise.	Control Name	Control Type	Notes	Edit Province	Heading	Page main heading	Province information	Heading	Page subheading	Name	Label	This control will be used as a label to indicate the Name attribute of the Province record	Name	Textbox	This control will be used by the administrator to update a name of the Province	Save	Button	This control will be used to confirm the update of the Province	Cancel	Button	This control will be used to cancel the edit of the Province
Control Name	Control Type	Notes																					
Edit Province	Heading	Page main heading																					
Province information	Heading	Page subheading																					
Name	Label	This control will be used as a label to indicate the Name attribute of the Province record																					
Name	Textbox	This control will be used by the administrator to update a name of the Province																					
Save	Button	This control will be used to confirm the update of the Province																					
Cancel	Button	This control will be used to cancel the edit of the Province																					
Step 5: The system administrator enters the updated province details and clicks on the update button [ALT]		Step 6: The system validates the provided Province detail using jQuery validations by checking if the Textbox is not empty [ALT]																					
		Step 7: The system displays a modal dialogue with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>"Confirm Province Details"</td><td>Label</td><td>This control is used to request confirmation on the updated province details</td></tr><tr><td>Confirm</td><td>Button</td><td>This control is used to confirm the updating of</td></tr></table>	Control Name	Control Type	Notes	"Confirm Province Details"	Label	This control is used to request confirmation on the updated province details	Confirm	Button	This control is used to confirm the updating of												
Control Name	Control Type	Notes																					
"Confirm Province Details"	Label	This control is used to request confirmation on the updated province details																					
Confirm	Button	This control is used to confirm the updating of																					

					the province details
			Cancel	Button	This control is used to close the modal dialogue and return the administrator to the Update Province Screen
	Step 8: The system administrator clicks on the confirm button [ALT]		Step 9: The system uses an SQL READ query to check if the province already exists in the Province table using the following attribute. [ALT]		
			Details to be read	Attributes in the table	
			Name	ProvinceName	
			Country Name	CountryName	
			Step 10: The system saves the updated details into the Province table using a method in a .NET controller and an SQL UPDATE query:		
			Details to be saved	Attributes in the table	Notes
			ID	ProvinceID	Generated by the system
			Name	ProvinceName	Required Alphabetical characters only Minimum: 1 Maximum: 50
		Country Name	CountryName	Required Alphabetical characters only Minimum: 1 Maximum: 50	
		Country ID	Country_ID	Required FK from Country table	
		Step 11: The system directs the administrator to the Province Screen and displays a bootstrap alert notification with the following controls:			
		Control Name	Control Name	Notes	
		"Province updated successfully"	Label	This control is used to indicate a successful creation	

ALTERNATE COURSES:	ALT Step 5: The administrator does not want to update the Province anymore - The administrator clicks on the cancel button - The use case is terminated		
	ALT Step 6 and Step 9: The details provided are not in the correct format or the required fields are empty The system displays a validation error message box with the following controls		
	Control Name	Control Name	Notes
	Name	Textbox	This control will be outline in red to show a validation error
	Invalid input	Label	This control will be used as a label to indicate a validation error in the Name textbox
	Return to step 3		
	ALT Step 8: The administrator does not confirm the update of a Province information and selects cancel - Terminate this use case.		
CONCLUSION:	This use case concludes when the province details are updated in the Province entity and the system displays an alert notification that informs the administrator that the province's details has been successfully updated.		
POST-CONDITION:	The province's details are updated in the Province table.		
BUSINESS RULES	Only the administrator can update an existing province's details.		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	- The system is offline and the administrator is not able to access the system to update a province's details. - A province must be associated with a country instance at all times. - No duplicate province instances may exist on the system at any time.		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

Table 100 - 9.8 Delete Province

LOCATION SUBSYSTEM				
AUTHORS (s): APIWE SHOZI		DATE: 22/08/2022		
VERSION: 1.0		LAST REVIEW DATE: 22/08/2022		
USE CASE NAME:	Delete Province	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒		
USE CASE ID:	9.8			
PRIORITY:	Medium			
SOURCE:	ByteXcom Requirements List			
PRIMARY BUSINESS ACTOR	• Administrator			
PRIMARY SYSTEM ACTOR	• None			
OTHER PARTICIPATING ACTORS:	• None			
OTHER INTERESTED STAKEHOLDERS:	• None			
DESCRIPTION:	This use case narrative describes the process in which the administrator wishes to delete a Province from the system. The administrator will search for the required Province selected for deletion. The administrator will then confirm the deletion of the Province and the use case concludes when the system notifies the administrator of a successful deletion.			
PRE-CONDITION:	The administrator must be logged in to the system.			
TRIGGER:	This use case is called when the administrator wishes to delete a province.			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE		
		MANUAL ACTION	AUTOMATED ACTION	
	Step 1: The administrator wishes to delete a province defined on the system.		Step 2: The system invokes use case 9.6 Search Province	
	Step 3: The administrator clicks on the Delete button		Step 4: The system retrieves the following attributes from the following tables, using a method in a .NET controller and an SQL READ query: From the Province Table:	
			Details to be read	Attributes in the table

Name	ProvinceName
Country Name	CountryName

and displays them on the **Delete Province Details Screen** with the following controls:

Control Name	Control Type	Notes
Province	Heading	Page main heading
Add	Button	When clicked, this control will direct the user to the Add province record page
Search bar	Textbox	This control will be used when the user inserts the search parameters
Search icon	Button	This control will be used to activate the search
Results table	Table	This control will be used to display all the Provinces that exist on the system
Name	Label	This control will be used as a heading to display the Name for all Provinces
Page count	Label	This control will be used as a label to display the number of pages that the results make up and to indicate on which page the user is currently on
Back and forward	Button	This control is used to navigate through the pages used to display the results from the table

	Step 5: The administrator clicks on the delete button [ALT]	Step 6: The system displays a modal dialogue with the following controls:															
		<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>"Confirm Province Deletion"</td><td>Label</td><td>This control is used to request confirmation on the deletion of the province details</td></tr><tr><td>Confirm</td><td>Button</td><td>This control is used to confirm the deletion of the province details</td></tr><tr><td>Cancel</td><td>Button</td><td>This control is used to close the modal dialogue</td></tr></table>	Control Name	Control Type	Notes	"Confirm Province Deletion"	Label	This control is used to request confirmation on the deletion of the province details	Confirm	Button	This control is used to confirm the deletion of the province details	Cancel	Button	This control is used to close the modal dialogue			
Control Name	Control Type	Notes															
"Confirm Province Deletion"	Label	This control is used to request confirmation on the deletion of the province details															
Confirm	Button	This control is used to confirm the deletion of the province details															
Cancel	Button	This control is used to close the modal dialogue															
	Step 7: The system administrator clicks on the confirm button [ALT]	Step 8: The system deletes the selected Province from the Province entity in the database using a method in a .NET controller and an SQL DELETE query.															
		<table><tr><th>Details to be created</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>ID</td><td>ID</td><td>Generated by the system</td></tr><tr><td>Name</td><td>ProvinceName</td><td>Required Alphabetical characters only Minimum: 1 Maximum: 50</td></tr><tr><td>Country Name</td><td>CountryName</td><td>Required Alphabetical characters only Minimum: 1 Maximum: 50</td></tr><tr><td>Country ID</td><td>Country_Id</td><td>Required FK from Country table</td></tr></table>	Details to be created	Attributes in the table	Notes	ID	ID	Generated by the system	Name	ProvinceName	Required Alphabetical characters only Minimum: 1 Maximum: 50	Country Name	CountryName	Required Alphabetical characters only Minimum: 1 Maximum: 50	Country ID	Country_Id	Required FK from Country table
Details to be created	Attributes in the table	Notes															
ID	ID	Generated by the system															
Name	ProvinceName	Required Alphabetical characters only Minimum: 1 Maximum: 50															
Country Name	CountryName	Required Alphabetical characters only Minimum: 1 Maximum: 50															
Country ID	Country_Id	Required FK from Country table															
		Step 9: The system redirects the administrator to the Province Screen and displays a modal dialogue with the following controls:															

			Control Name	Control Name	Notes
			"Province Deleted Successfully"	Label	This control is used to indicate a successful deletion
ALTERNATE COURSES:	ALT Step 5: The administrator does not want to delete the Province anymore - The administrator clicks on the cancel button - The use case is terminated				
	ALT Step 7: The administrator does not confirm the deletion of the Province - The administrator clicks on the cancel button - Return to step 4				
CONCLUSION:	This use case concludes when the administrator is notified of the successful deletion of the province.				
POST-CONDITION:	The system deletes all the details related to the selected province from the Province table.				
BUSINESS RULES	Only the administrator can delete a province using this functionality.				
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	- A province should not be deleted if it is associated with a City on the system. - Restrict the removal of a province if it is associated with either an order or a user's profile details.				
ASSUMPTIONS:	The system is online and the administrator has a stable internet connection.				
OPEN ISSUES:	None				

Table 101 - 9.9 Create City

LOCATION SUBSYSTEM				
AUTHORS (s): APIWE SHOZI		DATE: 22/08/2022		
VERSION: 1.0		LAST REVIEW DATE: 22/08/2022		
USE CASE NAME:	Create City	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒		
USE CASE ID:	9.9			
PRIORITY:	High			
SOURCE:	ByteXcom Requirements List			
PRIMARY BUSINESS ACTOR	• Administrator			
PRIMARY SYSTEM ACTOR	• None			
OTHER PARTICIPATING ACTORS:	• None			
OTHER INTERESTED STAKEHOLDERS:	• None			
DESCRIPTION:	This use case describes the process in which the administrator wishes to add a new City to the system as one of the locations that Natuurlik offers its Cities and services to. The use case begins when the administrator adds the new City under its respective Province which as the two entities are linked via the ProvinceID_FK. The administrator will add all of the pertinent information regarding the new City and save it to the City entity table.			
PRE-CONDITION:	• The administrator must be logged onto the system. • Only city that fall under countries that are already in the system can be added to the system			
TRIGGER:	The use case is triggered when the administrator wants to add a new city.			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE		
		MANUAL ACTION	AUTOMATED ACTION	
	Step 1: The administrator clicks on the City nav link		Step 2: The system retrieves the following attributes from each entry in the City Table, using a method in a .NET controller and an SQL READ query:	
		and displays them on the City Screen with the following controls:		

			Control Name	Control Type	Notes
			City	Heading	Page main heading
			Add	Button	When clicked, this control will direct the user to the Add city record page
			Search bar	Textbox	This control will be used when the user inserts the search parameters
			Search icon	Button	This control will be used to activate the search
			Results table	Table	This control will be used to display all the Cities that exist on the system
			Name	Label	This control will be used as a heading to display the Name for all Cities
			Page count	Label	This control will be used as a label to display the number of pages that the results make up and to indicate on which page the user is currently on
			Back and forward	Button	This control is used to navigate through the pages used to display the results from the table

	Step 3: The administrator clicks on the Create New City button		Step 4: The system displays the Add City screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Add City</td><td>Heading</td><td>Page main heading</td></tr><tr><td>City information</td><td>Heading</td><td>Page subheading</td></tr><tr><td>Name</td><td>Label</td><td>This control will be used as a label to indicate the Name attribute of the City record</td></tr><tr><td>Name</td><td>Textbox</td><td>This control will be used by the administrator to insert a name of the City</td></tr><tr><td>Save</td><td>Button</td><td>This control will be used to confirm the addition of the City</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the addition of the City</td></tr></table> Note: All controls are enabled by default, unless otherwise stated. All text input fields are empty by default, unless stated otherwise.	Control Name	Control Type	Notes	Add City	Heading	Page main heading	City information	Heading	Page subheading	Name	Label	This control will be used as a label to indicate the Name attribute of the City record	Name	Textbox	This control will be used by the administrator to insert a name of the City	Save	Button	This control will be used to confirm the addition of the City	Cancel	Button	This control will be used to cancel the addition of the City
	Control Name	Control Type	Notes																					
	Add City	Heading	Page main heading																					
City information	Heading	Page subheading																						
Name	Label	This control will be used as a label to indicate the Name attribute of the City record																						
Name	Textbox	This control will be used by the administrator to insert a name of the City																						
Save	Button	This control will be used to confirm the addition of the City																						
Cancel	Button	This control will be used to cancel the addition of the City																						
	Step 5: The system administrator enters the details and clicks on the create button [ALT]		Step 6: The system captures the information provided by the administrator and validates the information using jQuery validation ensuring the format is adhered to and no required text fields are left empty. [ALT]																					
			Step 7: The system uses an SQL READ query to check if the city already exists in the City table using the following attribute. [ALT] <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Name</td><td>CityName</td></tr></table>	Details to be read	Attributes in the table	Name	CityName																	
Details to be read	Attributes in the table																							
Name	CityName																							

			Step 8: The system saves the details in the City table using a method in a .NET controller and an SQL INSERT query: <table><tr><th>Details to be created</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>ID</td><td>ID</td><td>Generated by the system</td></tr><tr><td>Name</td><td>CityName</td><td>Required Alphabetical characters only Minimum: 1 Maximum: 50</td></tr><tr><td>Province Name</td><td>ProvinceName</td><td>Required Alphabetical characters only Minimum: 1 Maximum: 50</td></tr><tr><td>Province ID</td><td>Province_Id</td><td>Required FK from Province table</td></tr></table>	Details to be created	Attributes in the table	Notes	ID	ID	Generated by the system	Name	CityName	Required Alphabetical characters only Minimum: 1 Maximum: 50	Province Name	ProvinceName	Required Alphabetical characters only Minimum: 1 Maximum: 50	Province ID	Province_Id	Required FK from Province table
	Details to be created	Attributes in the table	Notes															
ID	ID	Generated by the system																
Name	CityName	Required Alphabetical characters only Minimum: 1 Maximum: 50																
Province Name	ProvinceName	Required Alphabetical characters only Minimum: 1 Maximum: 50																
Province ID	Province_Id	Required FK from Province table																
			Step 9: The system directs the administrator to the City Screen and displays a bootstrap alert notification with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Successful creation</td><td>Label</td><td>This control is used to indicate a successful creation</td></tr><tr><td>“X”</td><td>Button</td><td>This control is used to close the successful creation message box</td></tr></table>	Control Name	Control Name	Notes	Successful creation	Label	This control is used to indicate a successful creation	“X”	Button	This control is used to close the successful creation message box						
Control Name	Control Name	Notes																
Successful creation	Label	This control is used to indicate a successful creation																
“X”	Button	This control is used to close the successful creation message box																
ALTERNATE COURSES:	ALT Step 5: The administrator does not want to add a City anymore - The administrator clicks on the cancel button - The use case is terminated																	
	ALT Step 6: The details entered are not in the correct format or the required fields are empty The system displays a validation error with the following controls																	

	Return to step 5									
	ALT Step 7: The City already exists on the system The system displays an error modal dialogue with the following controls									
	<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>City already exists!</td><td>Label</td><td>This control is used to inform the administrator that the city already exists</td></tr><tr><td>Okay</td><td>Button</td><td>This control is used to cancel the creation of the city</td></tr></table>	Control Name	Control Name	Notes	City already exists!	Label	This control is used to inform the administrator that the city already exists	Okay	Button	This control is used to cancel the creation of the city
	Control Name	Control Name	Notes							
	City already exists!	Label	This control is used to inform the administrator that the city already exists							
Okay	Button	This control is used to cancel the creation of the city								
- The administrator clicks the “Okay” button - The use case is terminated										
CONCLUSION:	This use case concludes when a new city’s details are created and the system displays an alert notification which confirms the city details were created successfully.									
POST-CONDITION:	The city details are added to the City entity and saved to the database.									
BUSINESS RULES	Only the administrator can add a new city using this functionality.									
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	- A city should always be associated with a Province defined on the system. - No duplicate city details may exist on the system at any time.									
ASSUMPTIONS:	None									
OPEN ISSUES:	None									

Table 102 - 9.10 Search City

LOCATION SUBSYSTEM											
AUTHORS (s): APIWE SHOZI		DATE: 22/08/2022									
VERSION: 1.1		LAST REVIEW DATE: 24/05/2022									
USE CASE NAME:	Search City		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒								
USE CASE ID:	9.10										
PRIORITY:	Medium										
SOURCE:	ByteXcom Requirements List										
PRIMARY BUSINESS ACTOR	• Administrator										
PRIMARY SYSTEM ACTOR	• None										
OTHER PARTICIPATING ACTORS:	• None										
OTHER INTERESTED STAKEHOLDERS:	• None										
DESCRIPTION:	This use case describes the process in the administrator wishes to search through the existing Cities on the system in the City Entity Table. The system will prompt the administrator to use the City details in order to perform a search query and the system will retrieve the City record that matches the inputs of the search query										
PRE-CONDITION:	The administrator must be logged onto the system.										
TRIGGER:	This use case is called when the administrator wants to search for a city on the system.										
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE									
		MANUAL ACTION	AUTOMATED ACTION								
	Step 1: The administrator clicks on the City navigation link on the admin side navigation bar		Step 2: The system retrieves the following attributes for each entry from the City Table, using a method in a .NET controller and an SQL READ query:								
			<table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Name</td><td>CityName</td></tr><tr><td>Province Name</td><td>ProvinceName</td></tr></table>		Details to be read	Attributes in the table	Name	CityName	Province Name	ProvinceName	
			Details to be read	Attributes in the table							
Name	CityName										
Province Name	ProvinceName										
and displays then on the City Screen with the following controls:											
		<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>City</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Add</td><td>Button</td><td>When clicked, this control</td></tr></table>	Control Name	Control Type	Notes	City	Heading	Page main heading	Add	Button	When clicked, this control
Control Name	Control Type	Notes									
City	Heading	Page main heading									
Add	Button	When clicked, this control									

					will direct the user to the Add city record page
			Search bar	Textbox	This control will be used when the user inserts the search parameters
			Search icon	Button	This control will be used to activate the search
			Results table	Table	This control will be used to display all the Cities that exist on the system
			Name	Label	This control will be used as a heading to display the Name for all Cities
			Page count	Label	This control will be used as a label to display the number of pages that the results make up and to indicate on which page the user is currently on
			Back and forward	Button	This control is used to navigate through the pages used to display the results from the table
	Step 4: The administrator enters their search query into the search textbox		Step 5: The system searches the table rows in the data table and displays the City(s) matching the search query [ALT]		
	ALT Step 5: The search query entered does not match any records existing in the City entity. - The system displays a “No Records Found” message in the results table. - Return to step 4 or terminate the use case				
CONCLUSION:	This use case concludes when the desired city’s details are displayed on the system.				
POST-CONDITION:	The City entity was searched for and an entry that corresponds to the search query or parameter entered.				
BUSINESS RULES	Only the administrators should be allowed to search for existing city details using this functionality.				

IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• The system is online and the administrator has a stable internet connection.
OPEN ISSUES:	• None

Table 103 - 9.11 Update City

LOCATION SUBSYSTEM									
AUTHORS (s): APIWE SHOZI		DATE: 22/08/2022							
VERSION: 1.1		LAST REVIEW DATE: 24/05/2022							
USE CASE NAME:	Update City		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒						
USE CASE ID:	9.11								
PRIORITY:	Medium								
SOURCE:	ByteXcom Requirements List								
PRIMARY BUSINESS ACTOR	• Administrator								
PRIMARY SYSTEM ACTOR	• None								
OTHER PARTICIPATING ACTORS:	• None								
OTHER INTERESTED STAKEHOLDERS:	• None								
DESCRIPTION:	This use case describes the process in which the administrator wishes to update the details of City record in the City Entity table. The use case begins when the administrator updates a City that Natuurlik offers its products and services to. The administrator will update all of the pertinent information regarding the City and the system will update the record in the City entity table								
PRE-CONDITION:	The administrator must be logged in to the system.								
TRIGGER:	The use is triggered when the administrator wishes to update a city								
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE							
		MANUAL ACTION	AUTOMATED ACTION						
	Step 1: The administrator wishes to update a City in the system.		Step 2: The system invokes use case 9.10 Search City						
	Step 3: The administrator clicks on the update button		Step 4: The system retrieves the following attributes from the following tables, using a method in a .NET controller and an SQL READ query: From the City Table: <table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Name</td><td>CityName</td></tr><tr><td>Province Name</td><td>ProvinceName</td></tr></table>	Details to be read	Attributes in the table	Name	CityName	Province Name	ProvinceName
	Details to be read	Attributes in the table							
Name	CityName								
Province Name	ProvinceName								

			and displays them on the Update City Screen which contains an MVC form with the following controls:																					
			<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Edit City</td><td>Heading</td><td>Page main heading</td></tr><tr><td>City information</td><td>Heading</td><td>Page subheading</td></tr><tr><td>Name</td><td>Label</td><td>This control will be used as a label to indicate the Name attribute of the City record</td></tr><tr><td>Name</td><td>Textbox</td><td>This control will be used by the administrator to update a name of the City</td></tr><tr><td>Save</td><td>Button</td><td>This control will be used to confirm the update of the City</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the edit of the City</td></tr></table>	Control Name	Control Type	Notes	Edit City	Heading	Page main heading	City information	Heading	Page subheading	Name	Label	This control will be used as a label to indicate the Name attribute of the City record	Name	Textbox	This control will be used by the administrator to update a name of the City	Save	Button	This control will be used to confirm the update of the City	Cancel	Button	This control will be used to cancel the edit of the City
	Control Name	Control Type	Notes																					
	Edit City	Heading	Page main heading																					
	City information	Heading	Page subheading																					
	Name	Label	This control will be used as a label to indicate the Name attribute of the City record																					
	Name	Textbox	This control will be used by the administrator to update a name of the City																					
	Save	Button	This control will be used to confirm the update of the City																					
	Cancel	Button	This control will be used to cancel the edit of the City																					
			Note: All controls are enabled by default, unless stated otherwise.																					
Step 5: The system administrator enters the updated city details and clicks on the update button [ALT]		Step 6: The system validates the provided City detail using jQuery validations by checking if the Textbox is not empty [ALT]																						
		Step 7: The system displays a modal dialogue with the following controls:																						
		<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>“Confirm City Details”</td><td>Label</td><td>This control is used to request confirmation on the updated city details</td></tr><tr><td>Confirm</td><td>Button</td><td>This control is used to confirm the updating of the city details</td></tr><tr><td>Cancel</td><td>Button</td><td>This control is used to close</td></tr></table>	Control Name	Control Type	Notes	“Confirm City Details”	Label	This control is used to request confirmation on the updated city details	Confirm	Button	This control is used to confirm the updating of the city details	Cancel	Button	This control is used to close										
Control Name	Control Type	Notes																						
“Confirm City Details”	Label	This control is used to request confirmation on the updated city details																						
Confirm	Button	This control is used to confirm the updating of the city details																						
Cancel	Button	This control is used to close																						

					the modal dialogue and return the administrator to the Update City Screen														
	Step 8: The system administrator clicks on the confirm button [ALT]		Step 9: The system uses an SQL READ query to check if the city already exists in the City table using the following attribute. [ALT] <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Name</td><td>CityName</td></tr><tr><td>Province Name</td><td>ProvinceName</td></tr></table>			Details to be read	Attributes in the table	Name	CityName	Province Name	ProvinceName								
	Details to be read	Attributes in the table																	
	Name	CityName																	
Province Name	ProvinceName																		
		Step 10: The system saves the updated details into the City table using a method in a .NET controller and an SQL UPDATE query: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>ID</td><td>CityID</td><td>Generated by the system</td></tr><tr><td>Name</td><td>CityName</td><td>Required Alphabetical characters only Minimum: 1 Maximum: 50</td></tr><tr><td>Province Name</td><td>ProvinceName</td><td>Required Alphabetical characters only Minimum: 1 Maximum: 50</td></tr><tr><td>Province ID</td><td>Province_ID</td><td>Required FK from Province table</td></tr></table>			Details to be saved	Attributes in the table	Notes	ID	CityID	Generated by the system	Name	CityName	Required Alphabetical characters only Minimum: 1 Maximum: 50	Province Name	ProvinceName	Required Alphabetical characters only Minimum: 1 Maximum: 50	Province ID	Province_ID	Required FK from Province table
Details to be saved	Attributes in the table	Notes																	
ID	CityID	Generated by the system																	
Name	CityName	Required Alphabetical characters only Minimum: 1 Maximum: 50																	
Province Name	ProvinceName	Required Alphabetical characters only Minimum: 1 Maximum: 50																	
Province ID	Province_ID	Required FK from Province table																	
		Step 11: The system directs the administrator to the City Screen and displays a bootstrap alert notification with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"City updated successfully"</td><td>Label</td><td>This control is used to indicate a successful creation</td></tr></table>			Control Name	Control Name	Notes	"City updated successfully"	Label	This control is used to indicate a successful creation									
Control Name	Control Name	Notes																	
"City updated successfully"	Label	This control is used to indicate a successful creation																	
ALTERNATE COURSES:	ALT Step 5: The administrator does not want to update the City anymore - The administrator clicks on the cancel button - The use case is terminated																		

	ALT Step 6 and Step 9: The details provided are not in the correct format or the required fields are empty The system displays a validation error message box with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Name</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>Invalid input</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Name textbox</td></tr></table>	Control Name	Control Name	Notes	Name	Textbox	This control will be outline in red to show a validation error	Invalid input	Label	This control will be used as a label to indicate a validation error in the Name textbox
	Control Name	Control Name	Notes							
	Name	Textbox	This control will be outline in red to show a validation error							
	Invalid input	Label	This control will be used as a label to indicate a validation error in the Name textbox							
	Return to step 3									
ALT Step 8: The administrator does not wish to proceed with updating the city’s details for the update operation: - The administrator clicks the Cancel button - The use case is terminated										
CONCLUSION:	This use case concludes when the city details are updated and the system displays an alert notification that informs the administrator that the city’s details has been successfully updated.									
POST-CONDITION:	The city details are updated in the City table.									
BUSINESS RULES	Only the administrator can update a city’s details.									
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	The system is offline and the administrator is not able to access the system to update a city’s details.									
ASSUMPTIONS:	None									
OPEN ISSUES:	None									

Table 104 - 9.12 Delete City

LOCATION SUBSYSTEM					
AUTHORS (s): APIWE SHOZI		DATE: 22/08/2022			
VERSION: 1.0		LAST REVIEW DATE: 22/08/2022			
USE CASE NAME:	Delete City		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒		
USE CASE ID:	9.12				
PRIORITY:	Medium				
SOURCE:	ByteXcom Requirements List				
PRIMARY BUSINESS ACTOR	• Administrator				
PRIMARY SYSTEM ACTOR	• None				
OTHER PARTICIPATING ACTORS:	• None				
OTHER INTERESTED STAKEHOLDERS:	• None				
DESCRIPTION:	This use case narrative describes the process in which the administrator wishes to delete a City from the system. The administrator will search for the required City selected for deletion. The administrator will then confirm the deletion of the City and the use case concludes when the system notifies the administrator of a successful deletion.				
PRE-CONDITION:	The administrator must be logged in to the system.				
TRIGGER:	This use case is called when the administrator wishes to delete a city.				
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE			
		MANUAL ACTION	AUTOMATED ACTION		
	Step 1: The administrator wishes to delete a city defined on the system.		Step 2: The system invokes use case 9.6 Search City		
	Step 3: The administrator clicks on the Delete button		Step 4: The system retrieves the following attributes from the following tables, using a method in a .NET controller and an SQL READ query: From the City Table:		
	<table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Name</td><td>CityName</td></tr></table>		Details to be read	Attributes in the table	Name
Details to be read	Attributes in the table				
Name	CityName				

		<table><tr><td>Province Name</td><td>ProvinceName</td></tr></table> and displays them on the Delete City Details Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>City</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Add</td><td>Button</td><td>When clicked, this control will direct the user to the Add city record page</td></tr><tr><td>Search bar</td><td>Textbox</td><td>This control will be used when the user inserts the search parameters</td></tr><tr><td>Search icon</td><td>Button</td><td>This control will be used to activate the search</td></tr><tr><td>Results table</td><td>Table</td><td>This control will be used to display all the Cities that exist on the system</td></tr><tr><td>Name</td><td>Label</td><td>This control will be used as a heading to display the Name for all Cities</td></tr><tr><td>Page count</td><td>Label</td><td>This control will be used as a label to display the number of pages that the results make up and to indicate on which page the user is currently on</td></tr><tr><td>Back and forward</td><td>Button</td><td>This control is used to navigate through the pages used to display the results from the table</td></tr></table>	Province Name	ProvinceName	Control Name	Control Type	Notes	City	Heading	Page main heading	Add	Button	When clicked, this control will direct the user to the Add city record page	Search bar	Textbox	This control will be used when the user inserts the search parameters	Search icon	Button	This control will be used to activate the search	Results table	Table	This control will be used to display all the Cities that exist on the system	Name	Label	This control will be used as a heading to display the Name for all Cities	Page count	Label	This control will be used as a label to display the number of pages that the results make up and to indicate on which page the user is currently on	Back and forward	Button	This control is used to navigate through the pages used to display the results from the table
Province Name	ProvinceName																														
Control Name	Control Type	Notes																													
City	Heading	Page main heading																													
Add	Button	When clicked, this control will direct the user to the Add city record page																													
Search bar	Textbox	This control will be used when the user inserts the search parameters																													
Search icon	Button	This control will be used to activate the search																													
Results table	Table	This control will be used to display all the Cities that exist on the system																													
Name	Label	This control will be used as a heading to display the Name for all Cities																													
Page count	Label	This control will be used as a label to display the number of pages that the results make up and to indicate on which page the user is currently on																													
Back and forward	Button	This control is used to navigate through the pages used to display the results from the table																													
Step 5: The administrator clicks on the delete button		Step 6: The system displays a modal dialogue with the following controls:																													

	[ALT]		<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>"Confirm City Deletion"</td><td>Label</td><td>This control is used to request confirmation on the deletion of the city details</td></tr><tr><td>Confirm</td><td>Button</td><td>This control is used to confirm the deletion of the city details</td></tr><tr><td>Cancel</td><td>Button</td><td>This control is used to close the modal dialogue</td></tr></table>	Control Name	Control Type	Notes	"Confirm City Deletion"	Label	This control is used to request confirmation on the deletion of the city details	Confirm	Button	This control is used to confirm the deletion of the city details	Cancel	Button	This control is used to close the modal dialogue
	Control Name	Control Type	Notes												
	"Confirm City Deletion"	Label	This control is used to request confirmation on the deletion of the city details												
	Confirm	Button	This control is used to confirm the deletion of the city details												
Cancel	Button	This control is used to close the modal dialogue													
	Step 7: The system administrator clicks on the confirm button [ALT]		Step 8: The system deletes the selected City from the City entity in the database using a method in a .NET controller and an SQL DELETE query. <table><tr><th>Details to be created</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>City ID</td><td>CityID</td><td>To be removed</td></tr><tr><td>Name</td><td>CityName</td><td>To be removed</td></tr><tr><td>Province ID</td><td>ProvinceID</td><td>To be removed</td></tr></table> [ALT]	Details to be created	Attributes in the table	Notes	City ID	CityID	To be removed	Name	CityName	To be removed	Province ID	ProvinceID	To be removed
	Details to be created	Attributes in the table	Notes												
City ID	CityID	To be removed													
Name	CityName	To be removed													
Province ID	ProvinceID	To be removed													
			Step 9: The system redirects the administrator to the City Screen and displays a modal dialogue with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"City Deleted Successfully"</td><td>Label</td><td>This control is used to indicate a successful deletion</td></tr></table>	Control Name	Control Name	Notes	"City Deleted Successfully"	Label	This control is used to indicate a successful deletion						
Control Name	Control Name	Notes													
"City Deleted Successfully"	Label	This control is used to indicate a successful deletion													
ALTERNATE COURSES:	ALT Step 5: The administrator does not want to delete the City anymore - The administrator clicks on the cancel button - The use case is terminated														
	ALT Step 7: The administrator does not confirm the deletion of the City - The administrator clicks on the Cancel button - Return to step 4														
	ALT Step 8: The city selected for removal cannot be deleted since it is associated with an instance in either the Suburb, User or Order entities, and the system displays a validation error message containing the following controls:														

	Control Name	Control Name	Notes
	"City cannot be deleted"	Label	This control is used to indicate an unsuccessful deletion due to the city being associated to the above-mentioned entities.
CONCLUSION:	This use case concludes when the administrator is notified of the successful removal of the selected city.		
POST-CONDITION:	The system deletes all the details related to the selected city from the City table .		
BUSINESS RULES	Only the administrator can delete a city from the system.		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	- A city cannot be removed if it has a suburb associated. - A city cannot be removed if it is associated with an order placed on the system or assigned to a user's profile details.		
ASSUMPTIONS:	The system is online and the administrator has a stable internet connection.		
OPEN ISSUES:	None		

Table 105 - 9.13 Add Suburb

LOCATION SUBSYSTEM										
AUTHORS (s): APIWE SHOZI		DATE: 22/08/2022								
VERSION: 1.0		LAST REVIEW DATE: 22/08/2022								
USE CASE NAME:	Create Suburb	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒								
USE CASE ID:	9.13									
PRIORITY:	High									
SOURCE:	ByteXcom Requirements List									
PRIMARY BUSINESS ACTOR	• Administrator									
PRIMARY SYSTEM ACTOR	• None									
OTHER PARTICIPATING ACTORS:	• None									
OTHER INTERESTED STAKEHOLDERS:	• None									
DESCRIPTION:	This use case describes the process in which the administrator wishes to add a new Suburb to the system as one of the locations that Natuurlik offers its products and services to. The use case begins when the administrator adds the new Suburb under its respective City. The administrator will add all of the pertinent information regarding the new Suburb and save it to the Suburb entity table.									
PRE-CONDITION:	• The administrator must be logged onto the system. • Only suburb that fall under countries that are already in the system can be added to the system									
TRIGGER:	The use case is triggered when the administrator wants to add a new suburb.									
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE								
		MANUAL ACTION	AUTOMATED ACTION							
	Step 1: The administrator clicks on the Suburb nav link		Step 2: The system retrieves the following attributes from each entry in the Suburb Table, using a method in a .NET controller and an SQL READ query:							
			<table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Name</td><td>SuburbName</td></tr><tr><td>City Name</td><td>CityName</td></tr></table>		Details to be read	Attributes in the table	Name	SuburbName	City Name	CityName
			Details to be read	Attributes in the table						
			Name	SuburbName						
			City Name	CityName						
and displays them on the Suburb Screen with the following controls:										
<table><tr><td>Control Name</td><td>Control Type</td><td>Notes</td></tr><tr><td>Suburb</td><td>Heading</td><td>Page main heading</td></tr></table>		Control Name	Control Type	Notes	Suburb	Heading	Page main heading			
Control Name	Control Type	Notes								
Suburb	Heading	Page main heading								

			Add	Button	When clicked, this control will direct the user to the Add suburb record page
			Search bar	Textbox	This control will be used when the user inserts the search parameters
			Search icon	Button	This control will be used to activate the search
			Results table	Table	This control will be used to display all the Cities that exist on the system
			Name	Label	This control will be used as a heading to display the Name for all Cities
			Page count	Label	This control will be used as a label to display the number of pages that the results make up and to indicate on which page the user is currently on
			Back and forward	Button	This control is used to navigate through the pages used to display the results from the table

	Step 3: The administrator clicks on the Create New Suburb button	Step 4: The system displays the Add Suburb screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Add Suburb</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Suburb information</td><td>Heading</td><td>Page subheading</td></tr><tr><td>Name</td><td>Label</td><td>This control will be used as a label to indicate the Name attribute of the Suburb record</td></tr><tr><td>Name</td><td>Textbox</td><td>This control will be used by the administrator to insert a name of the Suburb</td></tr><tr><td>Save</td><td>Button</td><td>This control will be used to confirm the addition of the Suburb</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the addition of the Suburb</td></tr></table> Note: All controls are enabled by default, unless otherwise stated. All text input fields are empty by default, unless stated otherwise.	Control Name	Control Type	Notes	Add Suburb	Heading	Page main heading	Suburb information	Heading	Page subheading	Name	Label	This control will be used as a label to indicate the Name attribute of the Suburb record	Name	Textbox	This control will be used by the administrator to insert a name of the Suburb	Save	Button	This control will be used to confirm the addition of the Suburb	Cancel	Button	This control will be used to cancel the addition of the Suburb
	Control Name	Control Type	Notes																				
	Add Suburb	Heading	Page main heading																				
	Suburb information	Heading	Page subheading																				
Name	Label	This control will be used as a label to indicate the Name attribute of the Suburb record																					
Name	Textbox	This control will be used by the administrator to insert a name of the Suburb																					
Save	Button	This control will be used to confirm the addition of the Suburb																					
Cancel	Button	This control will be used to cancel the addition of the Suburb																					
Step 5: The system administrator enters the details and clicks on the create button [ALT]	Step 6: The system captures the information provided by the administrator and validates the information using jQuery validation ensuring the format is adhered to and no required text fields are left empty. [ALT]																						
	Step 7: The system uses an SQL READ query to check if the suburb already exists in the Suburb table using the following attribute. [ALT] <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Name</td><td>SuburbName</td></tr></table>	Details to be read	Attributes in the table	Name	SuburbName																		
Details to be read	Attributes in the table																						
Name	SuburbName																						

			Step 8: The system saves the details in the Suburb table using a method in a .NET controller and an SQL INSERT query: <table><tr><th>Details to be created</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>ID</td><td>ID</td><td>Generated by the system</td></tr><tr><td>Name</td><td>SuburbName</td><td>Required Alphabetical characters only Minimum: 1 Maximum: 50</td></tr><tr><td>City Name</td><td>CityName</td><td>Required Alphabetical characters only Minimum: 1 Maximum: 50</td></tr><tr><td>City ID</td><td>City_Id</td><td>Required FK from City table</td></tr></table>	Details to be created	Attributes in the table	Notes	ID	ID	Generated by the system	Name	SuburbName	Required Alphabetical characters only Minimum: 1 Maximum: 50	City Name	CityName	Required Alphabetical characters only Minimum: 1 Maximum: 50	City ID	City_Id	Required FK from City table
	Details to be created	Attributes in the table	Notes															
ID	ID	Generated by the system																
Name	SuburbName	Required Alphabetical characters only Minimum: 1 Maximum: 50																
City Name	CityName	Required Alphabetical characters only Minimum: 1 Maximum: 50																
City ID	City_Id	Required FK from City table																
			Step 9: The system directs the administrator to the Suburb Screen and displays a bootstrap alert notification with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Successful creation</td><td>Label</td><td>This control is used to indicate a successful creation</td></tr><tr><td>“X”</td><td>Button</td><td>This control is used to close the successful creation message box</td></tr></table>	Control Name	Control Name	Notes	Successful creation	Label	This control is used to indicate a successful creation	“X”	Button	This control is used to close the successful creation message box						
Control Name	Control Name	Notes																
Successful creation	Label	This control is used to indicate a successful creation																
“X”	Button	This control is used to close the successful creation message box																
ALTERNATE COURSES:	ALT Step 5: The administrator does not want to add a Suburb anymore - The administrator clicks on the cancel button - The use case is terminated																	
	ALT Step 6: The details entered are not in the correct format or the required fields are empty The system displays a validation error with the following controls																	

	Return to step 5									
	ALT Step 7: The Suburb already exists on the system The system displays an error modal dialogue with the following controls									
	<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Suburb already exists!</td><td>Label</td><td>This control is used to inform the administrator that the suburb already exists</td></tr><tr><td>Okay</td><td>Button</td><td>This control is used to cancel the creation of the suburb</td></tr></table>	Control Name	Control Name	Notes	Suburb already exists!	Label	This control is used to inform the administrator that the suburb already exists	Okay	Button	This control is used to cancel the creation of the suburb
	Control Name	Control Name	Notes							
	Suburb already exists!	Label	This control is used to inform the administrator that the suburb already exists							
Okay	Button	This control is used to cancel the creation of the suburb								
- The administrator clicks the “Okay” button - The use case is terminated										
CONCLUSION:	This use case concludes when a new suburb details are created and the system displays a confirmation alert which informs the administrator that the suburb has been successfully added.									
POST-CONDITION:	The suburb details are added to the Suburb entity and the changes are saved in the database.									
BUSINESS RULES	Only the administrator can add a new suburb.									
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	- The suburb to be added must be associated to an existing province which is defined on the system. - All suburbs defined on the system must be unique at all times.									
ASSUMPTIONS:	The administrator has access to the internet and the system is online.									
OPEN ISSUES:	None									

Table 106 - 9.14 Search Suburb

LOCATION SUBSYSTEM													
AUTHORS (s): APIWE SHOZI			DATE: 22/08/2022										
VERSION: 1.1			LAST REVIEW DATE: 24/05/2022										
USE CASE NAME:	Search Suburb		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒										
USE CASE ID:	9.14												
PRIORITY:	Medium												
SOURCE:	ByteXcom Requirements List												
PRIMARY BUSINESS ACTOR	• Administrator												
PRIMARY SYSTEM ACTOR	• None												
OTHER PARTICIPATING ACTORS:	• None												
OTHER INTERESTED STAKEHOLDERS:	• None												
DESCRIPTION:	This use case describes the process in the administrator wishes to search through the existing Cities on the system in the Suburb Entity Table. The system will prompt the administrator to use the Suburb details in order to perform a search query and the system will retrieve the Suburb record that matches the inputs of the search query												
PRE-CONDITION:	The administrator must be logged onto the system.												
TRIGGER:	This use case is called when the administrator wants to search for a suburb on the system.												
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE											
		MANUAL ACTION	AUTOMATED ACTION										
	Step 1: The administrator clicks on the Suburb navigation link on the admin side navigation bar		Step 2: The system retrieves the following attributes for each entry from the Suburb Table, using a method in a .NET controller and an SQL READ query:										
			<table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Name</td><td>SuburbName</td></tr><tr><td>City Name</td><td>CityName</td></tr></table>		Details to be read	Attributes in the table	Name	SuburbName	City Name	CityName			
			Details to be read	Attributes in the table									
			Name	SuburbName									
			City Name	CityName									
			and displays then on the Suburb Screen with the following controls:										
			<table><tr><td>Control Name</td><td>Control Type</td><td>Notes</td></tr><tr><td>Suburb</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Add</td><td>Button</td><td>When clicked, this control will direct the</td></tr></table>		Control Name	Control Type	Notes	Suburb	Heading	Page main heading	Add	Button	When clicked, this control will direct the
			Control Name	Control Type	Notes								
Suburb	Heading	Page main heading											
Add	Button	When clicked, this control will direct the											

				user to the Add suburb record page
		Search bar	Textbox	This control will be used when the user inserts the search parameters
		Search icon	Button	This control will be used to activate the search
		Results table	Table	This control will be used to display all the Cities that exist on the system
		Name	Label	This control will be used as a heading to display the Name for all Cities
		Page count	Label	This control will be used as a label to display the number of pages that the results make up and to indicate on which page the user is currently on
		Back and forward	Button	This control is used to navigate through the pages used to display the results from the table
	Step 4: The administrator enters their search query into the search textbox		Step 5: The system searches the table rows in the data table and displays the Suburb(s) matching the search query [ALT]	
	ALT Step 5: The search query entered does not match any records existing in the Suburb entity. • The system displays a “No Records Found” message in the results table. • Return to step 4 or terminate the use case			
CONCLUSION:	This use case concludes when the desired suburb’s details are displayed on the system.			
POST-CONDITION:	The Suburb entity was searched for and an entry that corresponds to the search query or parameter entered.			
BUSINESS RULES	• Only the administrators of the system should have access to this functionality.			

IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• A suburb can be searched for by name or by the name of the associated city.
ASSUMPTIONS:	• The system is online and the administrator has access to the internet.
OPEN ISSUES:	• None

Table 107 - 9.15 Update Suburb

LOCATION SUBSYSTEM								
AUTHORS (s): APIWE SHOZI		DATE: 22/08/2022						
VERSION: 1.1		LAST REVIEW DATE: 24/05/2022						
USE CASE NAME:	Update Suburb	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒						
USE CASE ID:	9.15							
PRIORITY:	Medium							
SOURCE:	ByteXcom Requirements List							
PRIMARY BUSINESS ACTOR	• Administrator							
PRIMARY SYSTEM ACTOR	• None							
OTHER PARTICIPATING ACTORS:	• None							
OTHER INTERESTED STAKEHOLDERS:	• None							
DESCRIPTION:	This use case describes the process in which the administrator wishes to update the details of Suburb record in the Suburb Entity table. The use case begins when the administrator updates a Suburb that Natuurlik offers its products and services to. The administrator will update all of the pertinent information regarding the Suburb and the system will update the record in the Suburb entity table							
PRE-CONDITION:	The administrator must be logged in to the system.							
TRIGGER:	The use is triggered when the administrator wishes to update a suburb							
TYPICAL COURSE OF EVENTS:	SYSTEM RESPONSE							
	ACTOR ACTION	MANUAL ACTION	AUTOMATED ACTION					
	Step 1: The administrator wishes to update a Suburb in the system.		Step 2: The system invokes use case 9.14 Search Suburb					
	Step 3: The administrator clicks on the update button		Step 4: The system retrieves the following attributes from the following tables, using a method in a .NET controller and an SQL READ query: From the Suburb Table: <table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Name</td><td>SuburbName</td></tr><tr><td>City Name</td><td>CityName</td></tr></table>	Details to be read	Attributes in the table	Name	SuburbName	City Name
Details to be read	Attributes in the table							
Name	SuburbName							
City Name	CityName							

		and displays them on the Update Suburb Screen which contains an MVC form with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Edit Suburb</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Suburb information</td><td>Heading</td><td>Page subheading</td></tr><tr><td>Name</td><td>Label</td><td>This control will be used as a label to indicate the Name attribute of the Suburb record</td></tr><tr><td>Name</td><td>Textbox</td><td>This control will be used by the administrator to update a name of the Suburb</td></tr><tr><td>Save</td><td>Button</td><td>This control will be used to confirm the update of the Suburb</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the edit of the Suburb</td></tr></table> Note: All controls are enabled by default, unless stated otherwise.	Control Name	Control Type	Notes	Edit Suburb	Heading	Page main heading	Suburb information	Heading	Page subheading	Name	Label	This control will be used as a label to indicate the Name attribute of the Suburb record	Name	Textbox	This control will be used by the administrator to update a name of the Suburb	Save	Button	This control will be used to confirm the update of the Suburb	Cancel	Button	This control will be used to cancel the edit of the Suburb
Control Name	Control Type	Notes																					
Edit Suburb	Heading	Page main heading																					
Suburb information	Heading	Page subheading																					
Name	Label	This control will be used as a label to indicate the Name attribute of the Suburb record																					
Name	Textbox	This control will be used by the administrator to update a name of the Suburb																					
Save	Button	This control will be used to confirm the update of the Suburb																					
Cancel	Button	This control will be used to cancel the edit of the Suburb																					
Step 5: The system administrator enters the updated suburb details and clicks on the update button [ALT]		Step 6: The system validates the provided Suburb detail using jQuery validations by checking if the Textbox is not empty [ALT]																					
		Step 7: The system displays a modal dialogue with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>“Confirm Suburb Details”</td><td>Label</td><td>This control is used to request confirmation on the updated suburb details</td></tr><tr><td>Confirm</td><td>Button</td><td>This control is used to confirm the updating of the suburb details</td></tr></table>	Control Name	Control Type	Notes	“Confirm Suburb Details”	Label	This control is used to request confirmation on the updated suburb details	Confirm	Button	This control is used to confirm the updating of the suburb details												
Control Name	Control Type	Notes																					
“Confirm Suburb Details”	Label	This control is used to request confirmation on the updated suburb details																					
Confirm	Button	This control is used to confirm the updating of the suburb details																					

			Cancel	Button	This control is used to close the modal dialogue and return the administrator to the Update Suburb Screen														
	Step 8: The system administrator clicks on the confirm button [ALT]		Step 9: The system uses an SQL READ query to check if the suburb already exists in the Suburb table using the following attribute. [ALT] <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Name</td><td>SuburbName</td></tr><tr><td>City Name</td><td>CityName</td></tr></table>			Details to be read	Attributes in the table	Name	SuburbName	City Name	CityName								
	Details to be read	Attributes in the table																	
	Name	SuburbName																	
	City Name	CityName																	
		Step 10: The system saves the updated details into the Suburb table using a method in a .NET controller and an SQL UPDATE query: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>ID</td><td>SuburbID</td><td>Generated by the system</td></tr><tr><td>Name</td><td>SuburbName</td><td>Required Alphabetical characters only Minimum: 1 Maximum: 50</td></tr><tr><td>City Name</td><td>CityName</td><td>Required Alphabetical characters only Minimum: 1 Maximum: 50</td></tr><tr><td>City ID</td><td>City_ID</td><td>Required FK from City table</td></tr></table>			Details to be saved	Attributes in the table	Notes	ID	SuburbID	Generated by the system	Name	SuburbName	Required Alphabetical characters only Minimum: 1 Maximum: 50	City Name	CityName	Required Alphabetical characters only Minimum: 1 Maximum: 50	City ID	City_ID	Required FK from City table
Details to be saved	Attributes in the table	Notes																	
ID	SuburbID	Generated by the system																	
Name	SuburbName	Required Alphabetical characters only Minimum: 1 Maximum: 50																	
City Name	CityName	Required Alphabetical characters only Minimum: 1 Maximum: 50																	
City ID	City_ID	Required FK from City table																	
		Step 11: The system directs the administrator to the Suburb Screen and displays a bootstrap alert notification with the following controls: <table><tr><th>Control Name</th><th>Control</th><th>Notes</th></tr><tr><td>"Suburb updated successfully"</td><td>Label</td><td>This control is used to indicate a successful creation</td></tr></table>			Control Name	Control	Notes	"Suburb updated successfully"	Label	This control is used to indicate a successful creation									
Control Name	Control	Notes																	
"Suburb updated successfully"	Label	This control is used to indicate a successful creation																	
	ALT Step 5: The administrator does not want to update the Suburb anymore The administrator clicks on the cancel button																		

ALTERNATE COURSES:	The use case is terminated									
	ALT Step 6 and Step 9: The details provided are not in the correct format or the required fields are empty									
	The system displays a validation error message box with the following controls									
	<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Name</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>Invalid input</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Name textbox</td></tr></table>	Control Name	Control Name	Notes	Name	Textbox	This control will be outline in red to show a validation error	Invalid input	Label	This control will be used as a label to indicate a validation error in the Name textbox
	Control Name	Control Name	Notes							
	Name	Textbox	This control will be outline in red to show a validation error							
Invalid input	Label	This control will be used as a label to indicate a validation error in the Name textbox								
Return to step 3										
ALT Step 8: The administrator does not confirm the update of a Suburb information and selects cancel										
	- Terminate this use case.									
CONCLUSION:	This use case concludes once the system displays an alert notification that informs the administrator that the suburb's details has been successfully updated.									
POST-CONDITION:	The suburb details are updated in the Suburb table and the changes are saved to the database.									
BUSINESS RULES	Only the administrator can update a suburb.									
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	- The system is offline and the administrator is not able to access the system to update a suburb's details. - All suburb's names should be unique at all times. - The suburb must be assigned to a city at all times.									
ASSUMPTIONS:	None									
OPEN ISSUES:	None									

Table 108 - 9.16 Delete Suburb

LOCATION SUBSYSTEM				
AUTHORS (s): APIWE SHOZI		DATE: 22/08/2022		
VERSION: 1.0		LAST REVIEW DATE: 22/08/2022		
USE CASE NAME:	Delete Suburb	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒		
USE CASE ID:	9.16			
PRIORITY:	Medium			
SOURCE:	ByteXcom Requirements List			
PRIMARY BUSINESS ACTOR	• Administrator			
PRIMARY SYSTEM ACTOR	• None			
OTHER PARTICIPATING ACTORS:	• None			
OTHER INTERESTED STAKEHOLDERS:	• None			
DESCRIPTION:	This use case narrative describes the process in which the administrator wishes to delete a Suburb from the system. The administrator will search for the required Suburb selected for deletion. The administrator will then confirm the deletion of the Suburb and the use case concludes when the system notifies the administrator of a successful deletion.			
PRE-CONDITION:	The administrator must be logged in to the system.			
TRIGGER:	This use case is called when the administrator wishes to delete a suburb.			
TYPICAL COURSE OF EVENTS:		SYSTEM RESPONSE		
	ACTOR ACTION	MANUAL ACTION	AUTOMATED ACTION	
	Step 1: The administrator wishes to delete a suburb defined on the system.		Step 2: The system invokes use case 9.6 Search Suburb	
	Step 3: The administrator clicks on the Delete button		Step 4: The system retrieves the following attributes from the following tables, using a method in a .NET controller and an SQL READ query: From the Suburb Table: <table><tr><td>Details to be read</td><td>Attributes in the table</td></tr></table>	Details to be read
Details to be read	Attributes in the table			

Name	SuburbName
City Name	CityName

and displays them on the **Delete Suburb Details Screen** with the following controls:

Control Name	Control Type	Notes
Suburb	Heading	Page main heading
Add	Button	When clicked, this control will direct the user to the Add suburb record page
Search bar	Textbox	This control will be used when the user inserts the search parameters
Search icon	Button	This control will be used to activate the search
Results table	Table	This control will be used to display all the Cities that exist on the system
Name	Label	This control will be used as a heading to display the Name for all Cities
Page count	Label	This control will be used as a label to display the number of pages that the results make up and to indicate on which page the user is currently on
Back and forward	Button	This control is used to navigate through the pages used to display the results from the table

	Step 5: The administrator clicks on the delete button [ALT]		Step 6: The system displays a modal dialogue with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>"Confirm Suburb Deletion"</td><td>Label</td><td>This control is used to request confirmation on the deletion of the suburb details</td></tr><tr><td>Confirm</td><td>Button</td><td>This control is used to confirm the deletion of the suburb details</td></tr><tr><td>Cancel</td><td>Button</td><td>This control is used to close the modal dialogue</td></tr></table>	Control Name	Control Type	Notes	"Confirm Suburb Deletion"	Label	This control is used to request confirmation on the deletion of the suburb details	Confirm	Button	This control is used to confirm the deletion of the suburb details	Cancel	Button	This control is used to close the modal dialogue
	Control Name	Control Type	Notes												
	"Confirm Suburb Deletion"	Label	This control is used to request confirmation on the deletion of the suburb details												
Confirm	Button	This control is used to confirm the deletion of the suburb details													
Cancel	Button	This control is used to close the modal dialogue													
Step 7: The system administrator clicks on the confirm button [ALT]		Step 8: The system deletes the selected Suburb from the Suburb entity in the database using a method in a .NET controller and an SQL DELETE query. <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>ID</td><td>SuburbID</td><td>To be removed.</td></tr><tr><td>Name</td><td>SuburbName</td><td>To be removed.</td></tr><tr><td>City ID</td><td>CityID</td><td>Association to be removed.</td></tr></table> [ALT]	Details to be saved	Attributes in the table	Notes	ID	SuburbID	To be removed.	Name	SuburbName	To be removed.	City ID	CityID	Association to be removed.	
Details to be saved	Attributes in the table	Notes													
ID	SuburbID	To be removed.													
Name	SuburbName	To be removed.													
City ID	CityID	Association to be removed.													
		Step 9: The system redirects the administrator to the Suburb Screen and displays a modal dialogue with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"Suburb Deleted Successfully"</td><td>Label</td><td>This control is used to indicate a successful deletion</td></tr></table>	Control Name	Control Name	Notes	"Suburb Deleted Successfully"	Label	This control is used to indicate a successful deletion							
Control Name	Control Name	Notes													
"Suburb Deleted Successfully"	Label	This control is used to indicate a successful deletion													
ALTERNATE COURSES:	ALT Step 5: The administrator does not want to delete the Suburb anymore - The administrator clicks on the cancel button - The use case is terminated														
	ALT Step 7: The administrator does not confirm the deletion of the Suburb - The administrator clicks on the cancel button - Return to step 4														
	ALT Step 8: The suburb selected for removal cannot be deleted since it is associated with a record in either or both the User, Supplier and Order entities,														

	and the system displays a validation error message containing the following controls:		
	Control Name	Control Name	Notes
	"Suburb cannot be deleted"	Label	This control is used to indicate an unsuccessful deletion due to the suburb being associated to the above-mentioned entities.
CONCLUSION:	This use case concludes once the system displays a confirmation message informing the administrator that the suburb's details have been removed successfully.		
POST-CONDITION:	The system removes all the details related to the selected suburb from the Suburb table and the changes are saved to the database.		
BUSINESS RULES	- Only the administrator can delete a suburb. - A suburb should not be deleted if it is associated to a user's profile details or a delivery address specified for an order placed on the system.		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	Restrict the deletion of a suburb if it is assigned to a user's profile or has been used to specify a delivery address for an order.		
ASSUMPTIONS:	The administrator has a stable internet connection and the system is online.		
OPEN ISSUES:	None		

2.10. Reporting Subsystem

Table 109 - 10.1 Generate Inventory List Report

REPORTING SUBSYSTEM			
AUTHOR(s): APIWE SHOZI		DATE: 07/08/2022	
VERSION: 1.0		LAST REVIEW DATE: 07/08/2022	
USE CASE NAME:	Generate Inventory List Report	USE CASE TYPE	
USE CASE ID:	10.1	Business Requirements:	☐
PRIORITY:	High	System Analysis:	☐
SOURCE:	ByteXpress Requirements List	System Design:	☒
PRIMARY BUSINESS ACTOR:	- Administrator - Inventory Manager		
PRIMARY THE SYSTEM ACTOR:	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS :	None		
DESCRIPTION:	This use case describes an event where the administrator or the inventory manager wishes to generate an inventory list report on the system. The administrator or the inventory manager wishes to generate an inventory list report and the system will retrieve the details regarding the company's inventory and generate a list report to display on the screen.		
PRE-CONDITION:	The administrator or inventory manager must be logged on the system.		
TRIGGER:	The use case is triggered when the administrator wants to generate inventory list report		
TYPICAL COURSE OF EVENTS:	ACTOR ACTION:	SYSTEM RESPONSE:	
		Manual Action	Automated Action

	Step 1: The administrator selects the “Inventory List Report” under the “Reports” tab on the side navigation link.		Step 2: The system responds by retrieving the following details regarding inventory from the respective entities using a method in a .NET controller and an SQL READ query: From the InventoryItem entity: <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Name</td><td>InventoryItemName</td></tr><tr><td>Quantity</td><td>QuantityonHand</td></tr></table> And from the InventoryType entity <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Name</td><td>InventoryTypeName</td></tr></table> And the following user claims details from the User table for the administrator or inventory manager who made the request: <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>First Name</td><td>FirstName</td></tr><tr><td>Surname</td><td>Surname</td></tr></table>	Details to be read	Attributes in the table	Name	InventoryItemName	Quantity	QuantityonHand	Details to be read	Attributes in the table	Name	InventoryTypeName	Details to be read	Attributes in the table	First Name	FirstName	Surname	Surname
Details to be read	Attributes in the table																		
Name	InventoryItemName																		
Quantity	QuantityonHand																		
Details to be read	Attributes in the table																		
Name	InventoryTypeName																		
Details to be read	Attributes in the table																		
First Name	FirstName																		
Surname	Surname																		
			Step 3: The system calculates the total number of inventory retrieved using a method in a .NET Controller.																
			Step 4: The system will then generate and display an inventory list report in a table format on the Inventory List Report Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Inventory List Report</td><td>Main heading</td><td>This control will be used to show the main heading</td></tr><tr><td>Export</td><td>Button</td><td>This control will be used to download the report</td></tr></table>	Control Name	Control Type	Notes	Inventory List Report	Main heading	This control will be used to show the main heading	Export	Button	This control will be used to download the report							
Control Name	Control Type	Notes																	
Inventory List Report	Main heading	This control will be used to show the main heading																	
Export	Button	This control will be used to download the report																	

			Results table	Table	This control will be used to display all the customers that exist on the system								
			Name	Column heading	This control will be used as the label for the respective attribute in the table that illustrate the inventory name								
			Inventory Type	Column heading	This control will be used as the label for the respective attribute in the table that illustrate the inventory type								
			Quantity	Column heading	This control will be used as the label for the respective attribute in the table that illustrate the inventory quantity								
			Total	Column heading	This control will be used as the label for the respective attribute in the table that illustrate the inventory total								
			Note: All controls are enabled by default, unless stated otherwise.										
	Step 5: The administrator/inventory manager clicks on the Export button		Step 6: The system exports the report using a JsPDF script in the relevant Razor page using the following layout and controls:										
			<table><tr><th>Details</th><th>Notes</th></tr><tr><td>Natuurlijk Header</td><td>The company logo and address in png format from the www root folder</td></tr><tr><td>Report Title</td><td>“Inventory Report”</td></tr><tr><td>Printed by</td><td>The authenticated user’s name and surname from</td></tr></table>			Details	Notes	Natuurlijk Header	The company logo and address in png format from the www root folder	Report Title	“Inventory Report”	Printed by	The authenticated user’s name and surname from
			Details	Notes									
			Natuurlijk Header	The company logo and address in png format from the www root folder									
			Report Title	“Inventory Report”									
Printed by	The authenticated user’s name and surname from												

				the claim details in the User table.
			Printed Date	The date in which the report is generated and populated by the system's current date in (DD/MM/YYYY) format.
			Body – Table	Populated with the generated inventory details.
			Footer Section	"End of Report" and page number(s)
ALTERNATE COURSES	None			
CONCLUSION:	The use case concludes when the inventory list report is downloaded from the system			
POST-CONDITION:	The InventoryItem and InventoryType table were read			
BUSINESS RULES:	Only the administrator and the inventory manager can generate an inventory list report			
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	None			
ASSUMPTIONS:	The administrator and the inventory manager have a stable internet connection			
OPEN ISSUES:	None			

Table 110 - 10.2 Generate Sales (by Top 5 Clients) Report

REPORTING SUBSYSTEM				
AUTHORS (s): Ofhani Mungani		DATE: 08/08/2022		
VERSION: 1.0		LAST REVIEW DATE: 25/08/2022		
USE CASE NAME:	Generate Sales (by Top 5 Clients) Report	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒		
USE CASE ID:	10.2			
PRIORITY:	High			
SOURCE:	ByteXcom Requirements List			
PRIMARY BUSINESS ACTOR	• Administrator/Sales Assistant			
PRIMARY SYSTEM ACTOR	• None			
OTHER PARTICIPATING ACTORS:	• None			
OTHER INTERESTED STAKEHOLDERS:	• None			
DESCRIPTION:	This use case describes an event where the administrator or sales assistant wants to generate sales (by top 5 clients) report on the system. The use case begins when the administrator/ sales assistant requests to generate sales (by top 5 clients) report and the system responds by retrieving the order details. The use case concludes when the report is displayed on the screen.			
PRE-CONDITION:	• The administrator or sales assistant must be logged on the system.			
TRIGGER:	The use case is called when the administrator or sales assistant wants to generate sales (by top 5 clients) report			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE		
		MANUAL ACTION	AUTOMATED ACTION	
	Step 1: The administrator or sales assistant clicks on the “Sales Report” in the Reports dropdown on the admin side navigation bar.		Step 2: The system retrieves the following attributes using a method in a .NET controller and an SQL READ query: From the Product entity:	
			Details to be read	
			Attributes in the table	
		Name	Name	

			From the OrderLine entity: <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Price</td><td>Price</td></tr><tr><td>Count</td><td>Count</td></tr></table> From the Order entity: <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>OrderTotal</td><td>OrderTotal</td></tr><tr><td>FirstName</td><td>FirstName</td></tr><tr><td>Surname</td><td>Surname</td></tr><tr><td>DeliveryFee</td><td>DeliveryFee</td></tr><tr><td>CreatedDate</td><td>CreatedDate</td></tr></table>	Details to be read	Attributes in the table	Price	Price	Count	Count	Details to be read	Attributes in the table	OrderTotal	OrderTotal	FirstName	FirstName	Surname	Surname	DeliveryFee	DeliveryFee	CreatedDate	CreatedDate
Details to be read	Attributes in the table																				
Price	Price																				
Count	Count																				
Details to be read	Attributes in the table																				
OrderTotal	OrderTotal																				
FirstName	FirstName																				
Surname	Surname																				
DeliveryFee	DeliveryFee																				
CreatedDate	CreatedDate																				
			Step 3: The system calculates the sales values using the following retrieved attributes in a method using .NET Controller. From the OrderLine entity: Sales =Price × Count TotalSales = Sum of Sales TotalProductQuantity = Sum of Count From the Order entity: Sales =OrderTotal – DeliveryFee TotalSales = Sum of Sales																		
			Step 4: The system displays the Sales Dashboard Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Sale Report</td><td>heading</td><td>This control will be used to display the heading of the screen</td></tr><tr><td>Sales by Top 5 Clients</td><td>button</td><td>This control will be used to navigate to Sales by Top 5 Clients Screen</td></tr><tr><td>Sales by Product</td><td>button</td><td>This control will be used to navigate to Sales by Product Screen</td></tr></table>	Control Name	Control Type	Notes	Sale Report	heading	This control will be used to display the heading of the screen	Sales by Top 5 Clients	button	This control will be used to navigate to Sales by Top 5 Clients Screen	Sales by Product	button	This control will be used to navigate to Sales by Product Screen						
Control Name	Control Type	Notes																			
Sale Report	heading	This control will be used to display the heading of the screen																			
Sales by Top 5 Clients	button	This control will be used to navigate to Sales by Top 5 Clients Screen																			
Sales by Product	button	This control will be used to navigate to Sales by Product Screen																			

			Monthly Sales	button	This control will be used to navigate to Sales by Month Screen
			Bar Graph:		
			Control Name	Control Type	Notes
			Top 5 Sales by Client Report	heading	This control will be used to show the main heading for the bar graph section
			Rands Value of Sales	label	This control will be used as the chart label of the bar graph
			x-axis	label	This control will be used as a label to display the name and surname for the top five clients
			y-axis	label	This control will be used as a label to display the amounts of sales in rands
			Sales	dataset	This control will be used to display the value of sales made by each customer and reseller in the top five calculated by OrderTotal - DeliveryFee from the Order table
			Pie Chart:		
			Control Name	Control Type	Notes
			Sales By Product Report	heading	This control will be used to show the main heading for the pie chart section

			Product Name	label	This control will be used to indicate the products name occupying the pie chart
			Sales	dataset	This control will be used to display the value of sales made by each product calculated by $\text{Price} \times \text{Count}$ from the OrderLine table
			Line Graph layout:		
			Control Name	Control Type	Notes
			Sales by Month Report	heading	This control will be used to show the main heading for the line graph section
			Rands Value of Sales	label	This control will be used as the chart label of the line graph
			x-axis	label	This control will be used as a label to display the month and year
			y-axis	label	This control will be used as a label to display the amounts of sales in rands
			Sales	dataset	This control will be used to display the value of sales made by each customer and reseller calculated by $\text{OrderTotal} - \text{DeliveryFee}$ from the Order table

	Step 5: The administrator or sales assistant clicks on the Sales by Top 5 Clients button		Step 6: The system retrieves the following attributes using a method in a .NET controller and an SQL READ query: From the Order entity: <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>OrderTotal</td><td>OrderTotal</td></tr><tr><td>FirstName</td><td>FirstName</td></tr><tr><td>Surname</td><td>Surname</td></tr><tr><td>DeliveryFee</td><td>DeliveryFee</td></tr></table> And the following claims details from the User entity for the administrator or sales assistant who made the request <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>FirstName</td><td>FirstName</td></tr><tr><td>Surname</td><td>Surname</td></tr></table>	Details to be read	Attributes in the table	OrderTotal	OrderTotal	FirstName	FirstName	Surname	Surname	DeliveryFee	DeliveryFee	Details to be read	Attributes in the table	FirstName	FirstName	Surname	Surname
Details to be read	Attributes in the table																		
OrderTotal	OrderTotal																		
FirstName	FirstName																		
Surname	Surname																		
DeliveryFee	DeliveryFee																		
Details to be read	Attributes in the table																		
FirstName	FirstName																		
Surname	Surname																		
			Step 7: The system calculates the sales values using the following retrieved attributes in a method using .NET Controller. From the Order entity: Sales =OrderTotal – DeliveryFee TotalSales = Sum of Sales																
			Step 8: The system generates the sales report and displays the Sales by Top 5 Clients Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Sale (by Top 5 Clients) Report</td><td>heading</td><td>This control will be used to display the heading of the screen</td></tr><tr><td>Back</td><td>button</td><td>This control will be used to navigate back to the sales dashboard</td></tr><tr><td>Export</td><td>button</td><td>This control will be used to download the report in Portable</td></tr></table>	Control Name	Control Type	Notes	Sale (by Top 5 Clients) Report	heading	This control will be used to display the heading of the screen	Back	button	This control will be used to navigate back to the sales dashboard	Export	button	This control will be used to download the report in Portable				
Control Name	Control Type	Notes																	
Sale (by Top 5 Clients) Report	heading	This control will be used to display the heading of the screen																	
Back	button	This control will be used to navigate back to the sales dashboard																	
Export	button	This control will be used to download the report in Portable																	

					Document Format (pdf)
			Bar Graph layout:		
			Control Name	Control Type	Notes
			Top Five(5) Sales byClient Report	heading	This control will be used to show the main heading for the bar graph section
			Rands Value of Sales	label	This control will be used as the chart label of the bar graph
			x-axis	label	This control will be used as a label to display the name and surname for the top five clients
			y-axis	label	This control will be used as a label to display the amounts of sales in rands
			Sales	dataset	This control will be used to display the value of sales made by each customer and reseller in the top five calculated by OrderTotal - DeliveryFee from the Order table
			Control Name	Control Type	Notes
			Results table	Table	This control will be used to give detail of the sale bar graph
			First Name	Table Heading	This control will be used as a heading to display the users Name
			Surname	Table Heading	This control will be used as a heading to display the

		users Surname
Sales Value (ZAR)	Table Headin g	This control will be used as a heading to display the users' total sales
Total	Table Row	This control will be used as the last row of the table to display total value of sales

Pie Chart layout:

Control Name	Control Type	Notes
Sales ByProduct Report	heading	This control will be used to show the main heading for the pie chart section
Product Name	label	This control will be used to indicate the products name occupying the pie chart
Sales	dataset	This control will be used to display the value of sales made by each product calculated by Price × Count from the OrderLine table

Control Name	Control Type	Notes
Results table	Table	This control will be used to give detail of the sale Pie Chart
Product Name	Table Headin g	This control will be used as a heading to display the product name
Product Quantity	Table Headin g	This control will be used as a heading to display the product quantity

			Sales Value (ZAR)	Table Heading	This control will be used as a heading to display the product's total sales
			Total	Table Row	This control will be used as the last row of the table to display total value of sales
			Line Graph Layout:		
			Control Name	Control Type	Notes
			Monthly Sales Report	heading	This control will be used to show the main heading for the line graph section
			Rands Value of Sales	label	This control will be used as the chart label of the line graph
			x-axis	label	This control will be used as a label to display the month and year
			y-axis	label	This control will be used as a label to display the amounts of sales in rands
			Sales	dataset	This control will be used to display the value of sales made by each customer and reseller calculated by $\text{OrderTotal} - \text{DeliveryFee}$ from the Order table
			Control Name	Control Type	Notes
			Results table	Table	This control will be used to give detail of the sale line graph

			Month	Table Heading	This control will be used as a heading to display the month name														
			Year	Table Heading	This control will be used as a heading to display the year														
			Sales Value (ZAR)	Table Heading	This control will be used as a heading to display the month total sales														
			Total	Table Row	This control will be used as the last row of the table to display total value of sales														
	Step 9: The administrator or sales assistant clicks on the Export button.		Step 10: The system exports the report using a JsPDF script in the relevant Razor page using the following layout and controls:																
			<table><tr><th>Details</th><th>Notes</th></tr><tr><td>Natuurlik Header</td><td>The company logo and address in png format from the www root folder</td></tr><tr><td>Report Title</td><td>Title for each sales report</td></tr><tr><td>Printed by</td><td>The authenticated user's name and surname from the claim details in the User table.</td></tr><tr><td>Printed Date</td><td>The date in which the report is generated and populated by the system's current date in (DD/MM/YYYY) format.</td></tr><tr><td>Body – Charts with Tables</td><td>Populated with the sales details.</td></tr><tr><td>Footer Section</td><td>"End of Report" and page number(s)</td></tr></table>			Details	Notes	Natuurlik Header	The company logo and address in png format from the www root folder	Report Title	Title for each sales report	Printed by	The authenticated user's name and surname from the claim details in the User table.	Printed Date	The date in which the report is generated and populated by the system's current date in (DD/MM/YYYY) format.	Body – Charts with Tables	Populated with the sales details.	Footer Section	"End of Report" and page number(s)
			Details	Notes															
			Natuurlik Header	The company logo and address in png format from the www root folder															
			Report Title	Title for each sales report															
			Printed by	The authenticated user's name and surname from the claim details in the User table.															
			Printed Date	The date in which the report is generated and populated by the system's current date in (DD/MM/YYYY) format.															
			Body – Charts with Tables	Populated with the sales details.															
Footer Section	"End of Report" and page number(s)																		
ALTERNATE COURSES:	None.																		
CONCLUSION:	The use case concludes when the sales report is downloaded																		
POST-CONDITION:	Sales details is read from the database using the Order , OrderLine , Product tables.																		
BUSINESS RULES	None.																		
IMPLEMENTATION CONSTRAINTS	The administrator or sales assistant cannot generate sales report without an internet connection.																		

AND SPECIFICATIONS	
ASSUMPTIONS:	• The administrator will always export the report
OPEN ISSUES:	• None

Table 111 - 10.3 Generate Product List Report

REPORTING SUBSYSTEM				
AUTHORS (s): Thenjiwe Ntsonda		DATE: 13/08/2022		
VERSION: 1.0		LAST REVIEW DATE: 13/08/2022		
USE CASE NAME:	Generate Product List Report	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒		
USE CASE ID:	10.3			
PRIORITY:	High			
SOURCE:	ByteXpress Requirements List			
PRIMARY BUSINESS ACTOR:	• Administrator or Inventory Manager			
PRIMARY THE SYSTEM ACTOR:	• None			
OTHER PARTICIPATING ACTORS:	• None			
OTHER INTERESTED STAKEHOLDERS:	• None			
DESCRIPTION:	This use case describes the process in which the administrator or inventory manager wishes to generate a product list report on the system. The system will display the report results and provide the options for it to be downloaded. The use case ends when the administrator or inventory manager downloads the report.			
PRE-CONDITION:	The administrator or inventory manager must be logged onto the system			
TRIGGER:	The use case is triggered when the administrator or inventory manager wants to generate a product list report			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION:	SYSTEM RESPONSE:		
		Manual Action	Automated Action	
	Step 1: The administrator clicks on the Product Report link in the Reports dropdown on the admin side navigation bar		Step 2: The system retrieves the following attributes from the Product table, using a method in a .NET controller and an SQL READ query:	
		Details to be read	Attributes in the table	
		Product Name	Name	
		Quantity on Hand	QuantityOnHand	
		Reseller Price	ResellerPrice	
		Customer Price	CustomerPrice	

			And the ProductBrand table with the following attributes: <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Product Brand</td><td>Name</td></tr></table>	Details to be read	Attributes in the table	Product Brand	Name		
	Details to be read	Attributes in the table							
	Product Brand	Name							
			And the ProductCategory table with the following attributes: <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Product Category</td><td>Name</td></tr></table>	Details to be read	Attributes in the table	Product Category	Name		
	Details to be read	Attributes in the table							
	Product Category	Name							
			And the User table with the following attributes: <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>First Name</td><td>FirstName</td></tr><tr><td>Surname</td><td>Surname</td></tr></table>	Details to be read	Attributes in the table	First Name	FirstName	Surname	Surname
	Details to be read	Attributes in the table							
	First Name	FirstName							
	Surname	Surname							
		Step 3: The system calculates the total quantity of products on hand using the QuantityOnHand attribute from the Product table and a method in a .NET Controller.							

		Step 4: The system generates the product list report using the retrieved attributes and displays the Product Report Screen which contains the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Product Report</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Export</td><td>Button</td><td>This control will be used to download the report</td></tr><tr><td>Results table</td><td>Table</td><td>This control will be used to display all the products that exist on the system</td></tr><tr><td>Product Name</td><td>Table Heading</td><td>This control will be used as a heading to display the product name</td></tr><tr><td>Quantity On Hand</td><td>Table Heading</td><td>This control will be used as a heading to display the product quantity on hand</td></tr><tr><td>Reseller Price</td><td>Table Heading</td><td>This control will be used as a heading to display the reseller price</td></tr><tr><td>Customer Price</td><td>Table Heading</td><td>This control will be used as a heading to display the customer price</td></tr></table> Note: All controls are enabled by default, unless stated otherwise.	Control Name	Control Type	Notes	Product Report	Heading	Page main heading	Export	Button	This control will be used to download the report	Results table	Table	This control will be used to display all the products that exist on the system	Product Name	Table Heading	This control will be used as a heading to display the product name	Quantity On Hand	Table Heading	This control will be used as a heading to display the product quantity on hand	Reseller Price	Table Heading	This control will be used as a heading to display the reseller price	Customer Price	Table Heading	This control will be used as a heading to display the customer price
	Control Name	Control Type	Notes																							
	Product Report	Heading	Page main heading																							
	Export	Button	This control will be used to download the report																							
	Results table	Table	This control will be used to display all the products that exist on the system																							
	Product Name	Table Heading	This control will be used as a heading to display the product name																							
	Quantity On Hand	Table Heading	This control will be used as a heading to display the product quantity on hand																							
	Reseller Price	Table Heading	This control will be used as a heading to display the reseller price																							
	Customer Price	Table Heading	This control will be used as a heading to display the customer price																							
	Step 5: The administrator clicks on the Export button	Step 6: The system exports the report using a JsPDF script in the relevant Razor page using the following layout and controls: <table><tr><th>Details</th><th>Notes</th></tr><tr><td>Natuurlik Header</td><td>The company logo and address in png format from the www root folder</td></tr></table>	Details	Notes	Natuurlik Header	The company logo and address in png format from the www root folder																				
Details	Notes																									
Natuurlik Header	The company logo and address in png format from the www root folder																									

			Report Title	“Product Report”
			Printed by	The authenticated user’s name and surname from the claim details in the User table.
			Printed Date	The date in which the report is generated and populated by the system’s current date in (DD/MM/YYYY) format.
			Body – Table	Populated with the generated product details.
			Footer Section	“End of Report” and page number(s)
ALTERNATE COURSES	• None			
CONCLUSION:	This use case concludes when the report is downloaded			
POST-CONDITION:	The Product , ProductBrand and ProductCategory tables were read			
BUSINESS RULES:	None			
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	Only the administrator or inventory manager have access to this function			
ASSUMPTIONS:	The administrator or inventory manager will always export the report			
OPEN ISSUES:	None			

Table 112 - 10.4 Generate Customer List Report

REPORTING SUBSYSTEM				
AUTHORS (s): Nomusa Vumisa		DATE: 08/08/2022		
VERSION: 1.0		LAST REVIEW DATE: 01/09/2022		
USE CASE NAME:	Generate Customer List Report	USE CASE TYPE		
USE CASE ID:	10.4	Business Requirements:	☐	
PRIORITY:	High	System Analysis:	☐	
SOURCE:	ByteXpress Requirements List	System Design:	☒	
PRIMARY BUSINESS ACTOR:	Administrator			
PRIMARY THE SYSTEM ACTOR:	None			
OTHER PARTICIPATING ACTORS:	None			
OTHER INTERESTED STAKEHOLDERS:	None			
DESCRIPTION:	This use case describes the process in which the administrator wishes to generate a customer list report on the system. The system will display the report results and provide the options for it to be downloaded. The use case ends when the administrator downloads the report.			
PRE-CONDITION:	The administrator must be logged onto the system			
TRIGGER:	The use case is triggered when the administrator wants to generate a customer list report			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION:	SYSTEM RESPONSE:		
		Manual Action	Automated Action	
	Step 1: The administrator clicks on the Customer Report link in the Reports dropdown on the admin side navigation bar		Step 2: The system retrieves the following attributes from the User table, using a method in a .NET controller and an SQL READ query:	
			Details to be read	Attributes in the table
			First Name	FirstName
			Surname	Surname
			Phone Number	PhoneNumber
Email Address	Email			
		From the UserRole table:		

			<table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Name</td><td>Name</td></tr></table>	Details to be read	Attributes in the table	Name	Name																
	Details to be read	Attributes in the table																					
	Name	Name																					
			And the following user claims details from the User table for the administrator who made the request:																				
			<table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>First Name</td><td>FirstName</td></tr><tr><td>Surname</td><td>Surname</td></tr></table>	Details to be read	Attributes in the table	First Name	FirstName	Surname	Surname														
	Details to be read	Attributes in the table																					
	First Name	FirstName																					
	Surname	Surname																					
			Step 3: The system calculates the total number of customers retrieved using a method in a .NET Controller.																				
			Step 4: The system generates the customers report using the retrieved attributes and displays the Customer List Report Screen which contains the following controls:																				
		<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Customers List Report</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Help Icon</td><td>Button</td><td>When clicked, the admin help document will open</td></tr><tr><td>Export</td><td>Button</td><td>This control will be used to download the report</td></tr><tr><td>Total Customers</td><td>Label</td><td>This control will be used to display the total number of customers on the system</td></tr><tr><td>Results table</td><td>Table</td><td>This control will be used to display all the customers that exist on the system</td></tr><tr><td>First Name</td><td>Table Heading</td><td>This control will be used as a heading to display the customer's Name</td></tr></table>	Control Name	Control Type	Notes	Customers List Report	Heading	Page main heading	Help Icon	Button	When clicked, the admin help document will open	Export	Button	This control will be used to download the report	Total Customers	Label	This control will be used to display the total number of customers on the system	Results table	Table	This control will be used to display all the customers that exist on the system	First Name	Table Heading	This control will be used as a heading to display the customer's Name
Control Name	Control Type	Notes																					
Customers List Report	Heading	Page main heading																					
Help Icon	Button	When clicked, the admin help document will open																					
Export	Button	This control will be used to download the report																					
Total Customers	Label	This control will be used to display the total number of customers on the system																					
Results table	Table	This control will be used to display all the customers that exist on the system																					
First Name	Table Heading	This control will be used as a heading to display the customer's Name																					

			Surname	Table Heading	This control will be used as a heading to display the customer's Surname																	
			Phone Number	Table Heading	This control will be used as a heading to display the customer's Phone Number																	
			Email Address	Table Heading	This control will be used as a heading to display the customer's Email Address																	
			Note: All controls are enabled by default, unless stated otherwise.																			
	Step 5: The administrator clicks on the Export button		Step 6: The system exports the report using a JsPDF script in the relevant Razor page using the following layout and controls:																			
			<table><tr><th>Details</th><th>Notes</th></tr><tr><td>Natuurlik Header</td><td>The company logo and address in png format from the www root folder</td></tr><tr><td>Report Title</td><td>"Customer List Report"</td></tr><tr><td>Printed by</td><td>The authenticated user's name and surname from the claim details in the User table.</td></tr><tr><td>Printed Date</td><td>The date in which the report is generated and populated by the system's current date in (DD/MM/YYYY) format.</td></tr><tr><td>Total Customers</td><td>The total amount of customers retrieved from the generated details</td></tr><tr><td>Body – Table</td><td>Populated with the generated customer details.</td></tr><tr><td>Footer Section</td><td>"End of Report" and page number(s)</td></tr></table>				Details	Notes	Natuurlik Header	The company logo and address in png format from the www root folder	Report Title	"Customer List Report"	Printed by	The authenticated user's name and surname from the claim details in the User table.	Printed Date	The date in which the report is generated and populated by the system's current date in (DD/MM/YYYY) format.	Total Customers	The total amount of customers retrieved from the generated details	Body – Table	Populated with the generated customer details.	Footer Section	"End of Report" and page number(s)
			Details	Notes																		
			Natuurlik Header	The company logo and address in png format from the www root folder																		
			Report Title	"Customer List Report"																		
			Printed by	The authenticated user's name and surname from the claim details in the User table.																		
Printed Date			The date in which the report is generated and populated by the system's current date in (DD/MM/YYYY) format.																			
Total Customers			The total amount of customers retrieved from the generated details																			
Body – Table	Populated with the generated customer details.																					
Footer Section	"End of Report" and page number(s)																					
None																						
This use case concludes when the report is downloaded																						
The User and UserRole table were read																						
None																						

IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	Only the administrator has access to this function
ASSUMPTIONS:	The administrator will always export the report
OPEN ISSUES:	None

Table 113 - 10.5 Generate Reseller List Report

REPORTING SUBSYSTEM				
AUTHORS (s): Nomusa Vumisa		DATE: 08/08/2022		
VERSION: 1.0		LAST REVIEW DATE: 08/08/2022		
USE CASE NAME:	Generate Reseller List Report	USE CASE TYPE		
USE CASE ID:	10.5	Business Requirements:	☐	
PRIORITY:	High	System Analysis:	☐	
SOURCE:	ByteXpress Requirements List	System Design:	☒	
PRIMARY BUSINESS ACTOR:	Administrator			
PRIMARY THE SYSTEM ACTOR:	None			
OTHER PARTICIPATING ACTORS:	None			
OTHER INTERESTED STAKEHOLDERS:	None			
DESCRIPTION:	This use case describes the process in which the administrator wishes to generate a reseller list report on the system. The system will display the report results and provide the options for it to be downloaded. The use case ends when the administrator downloads the report.			
PRE-CONDITION:	The administrator must be logged onto the system			
TRIGGER:	The use case is triggered when the administrator wants to generate a reseller list report			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION:	SYSTEM RESPONSE:		
		Manual Action	Automated Action	
	Step 1: The administrator clicks on the Reseller Report link in the Reports dropdown on the admin side navigation bar		Step 2: The system retrieves the following attributes from the User table, using a method in a .NET controller and an SQL READ query:	
			Details to be read	Attributes in the table
			First Name	FirstName
			Surname	Surname
			Phone Number	PhoneNumber
Email Address	Email			
		From the UserRole table:		

			Details to be read	Attributes in the table
			Name	Name
			And the following user claims details from the User table for the administrator who made the request:	
			Details to be read	Attributes in the table
			First Name	FirstName
			Surname	Surname
			Step 3: The system calculates the total number of resellers retrieved using a method in a .NET Controller.	

Step 4: The system generates the resellers report using the retrieved attributes and displays the **Reseller List Report Screen** which contains the following controls:

Control Name	Control Type	Notes
Reseller List Report	Heading	Page main heading
Help Icon	Button	Displays help tooltip.
Export	Button	This control will be used to download the report
Total Resellers	Label	Displays total count of resellers.
Results table	Table	This control will be used to display all the resellers that exist on the system
First Name	Table Heading	This control will be used as a heading to display the resellers Name
Surname	Table Heading	This control will be used as a heading to display the resellers Surname
Phone Number	Table Heading	This control will be used as a heading to display the resellers Phone Number
Email Address	Table Heading	This control will be used as a heading to display the resellers Email Address

Note: All controls are enabled by default, unless stated otherwise.

	Step 5: The administrator clicks on the Export button		Step 6: The system exports the report using a JsPDF script in the relevant Razor page using the following layout and controls:

Table 114 - 10.6 Generate Produced Products Report

REPORTING SUBSYSTEM						
AUTHOR(s): Kyle van Eeden			DATE: 08/08/2022			
VERSION: 1.1			LAST REVIEW DATE: 14/08/2022			
USE CASE NAME:	Generate Produced Products Report	USE CASE TYPE				
USE CASE ID:	10.6	Business Requirements: ☐ System Analysis: ☐ System Design: ☒				
PRIORITY:	Low					
SOURCE:	ByteXecom Requirements List					
PRIMARY BUSINESS ACTOR:	• Administrator or Inventory Manager					
PRIMARY THE SYSTEM ACTOR:	• None					
OTHER PARTICIPATING ACTORS:	• None					
OTHER INTERESTED STAKEHOLDERS:	• None					
DESCRIPTION:	This use case describes the process where the administrator or inventory manager wishes to generate the Produced Products Report on the system. The administrator or inventory manager will allowed to optionally enter a Start Date and End Date or apply a product name filter on the report to be generated. Once the user selects the generate button, the system will retrieve all production transaction records from the ProductionTransaction entity which have met the specified filter criteria, if any, and the use case concludes once the generated Produced Products report details is displayed to the administrator or inventory manager.					
PRE-CONDITION:	• The administrator or inventory manager must be logged into their registered Natuurlik account.					
TRIGGER:	The administrator or inventory manager wishes to generate a production report using the filter criteria they wish to specify.					
TYPICAL COURSE OF EVENTS:	ACTOR ACTION:	SYSTEM RESPONSE:				
		Manual Action	Automated Action			
	Step 1: The administrator or inventory manager selects the 'Produced Products Report' navigation link under the		Step 2: The system displays the Produced Products Report screen with the following controls which also prompts the user to specify an optional filter criteria for the report:			
			<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Produced Products Report</td><td>Heading</td><td>The main heading of the page displayed.</td></tr></table>	Control Name	Control Type	Notes
Control Name	Control Type	Notes				
Produced Products Report	Heading	The main heading of the page displayed.				

	Reporting section in the side navigation menu.		Natuurlik Logo	Image	The Natuurlik logo image to be displayed on the screen.
			Date From	Label	Label for the Date From date- picker control.
			Date From Value	Date- Picker	Allows the user to select a DateFrom filter value for the report. Default format is set to : 'dd/mm/yyyy'
			Date To	Label	Label for the Date To date- picker control.
			Date To Value	Date- Picker	Allows the user to select a DateTo filter value for the report. Default format is set to : 'dd/mm/yyyy'
			Search Product Name	Drop- Down List	Label for the Search By Product Name textbox control.
			Product To Filter By	Textbox	Displays the selected product to filter report by. This control is always disabled.
			Generate Report	Button	Clicked if the user wishes to proceed with generating the report.
			Note: All controls are enabled by default unless specified otherwise in the 'Notes' section.		
	Step 3: The administrator or inventor manager clicks on the 'Generate Report' button. [ALT]		Step 4: The system uses a .NET controller method with a SQL Read query to retrieve the following details from the ProductionTransaction entity for all production records retrieved:		
			Details to be read	Attributes in the table	Notes
			Product ID	ProductId	The unique identifier value of the product(s) which has been produced for each production

					transaction retrieved.
			Quantity Before	QuantityBefore	The QuantityOnHand value of each product before the production activity took place.
			Quantity After	QuantityAfter	The QuantityOnHand value of each product after the production activity took place.
			Transaction Date	TransactionDate	The date and time which each production activity was captured on the system.
			The following details are also retrieved from the Product entity:		
			Details to be read	Attributes in the table	Notes
			Product Name	ProductName	The name of each product which was retrieved.
			The system will retrieve the requesting user's claim details from the User entity using a .NET Controller method. The following details are retrieved:		
			Details to be read	Attributes in the table	Notes
			Name	Name	The first name of the administrator or inventory manager making the request to generate the report.
			Surname	Surname	The surname of the administrator or inventory manager requesting to generate the report.
			[ALT]		

			Step 5: The system uses a .NET Controller method to subtract the QuantityAfter attribute value from the QuantityBefore attribute value for each production record retrieved in order to calculate the number of finished products which was produced. This calculated value is assigned to a QtyProduced variable of type integer.																														
			Step 6: The system displays the generated produced finished products report details on the Produced Products Report screen with the following controls and layout: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Produced Products Report</td><td>Heading</td><td>Main Heading of the report.</td></tr><tr><td>Date From</td><td>Label</td><td>Label control for the Date From date- picker control.</td></tr><tr><td>Date From</td><td>DateFrom</td><td>Populated with the from date filter value if specified. Format displayed in: 'dd-mm-yyyy'</td></tr><tr><td>Date To</td><td>Label</td><td>Label control for the Date To date- picker control.</td></tr><tr><td>Date To</td><td>DateTo</td><td>Populated with the to date filter value if specified. Format displayed in: 'dd-mm-yyyy'</td></tr><tr><td>Search By Product Name</td><td>Label</td><td>Label control for the Search By Product Name textbox control.</td></tr><tr><td>Search By Product Name</td><td>Textbox</td><td>Allows the user to filter the displayed production transactions by ProductName attribute value.</td></tr><tr><td>Production Transactions</td><td>Table</td><td>Table containing all the report details retrieved.</td></tr><tr><td>Product Produced</td><td>Table Heading</td><td>Houses the ProductName attribute value for each production</td></tr></table>	Control Name	Control Type	Notes	Produced Products Report	Heading	Main Heading of the report.	Date From	Label	Label control for the Date From date- picker control.	Date From	DateFrom	Populated with the from date filter value if specified. Format displayed in: 'dd-mm-yyyy'	Date To	Label	Label control for the Date To date- picker control.	Date To	DateTo	Populated with the to date filter value if specified. Format displayed in: 'dd-mm-yyyy'	Search By Product Name	Label	Label control for the Search By Product Name textbox control.	Search By Product Name	Textbox	Allows the user to filter the displayed production transactions by ProductName attribute value.	Production Transactions	Table	Table containing all the report details retrieved.	Product Produced	Table Heading	Houses the ProductName attribute value for each production
Control Name	Control Type	Notes																															
Produced Products Report	Heading	Main Heading of the report.																															
Date From	Label	Label control for the Date From date- picker control.																															
Date From	DateFrom	Populated with the from date filter value if specified. Format displayed in: 'dd-mm-yyyy'																															
Date To	Label	Label control for the Date To date- picker control.																															
Date To	DateTo	Populated with the to date filter value if specified. Format displayed in: 'dd-mm-yyyy'																															
Search By Product Name	Label	Label control for the Search By Product Name textbox control.																															
Search By Product Name	Textbox	Allows the user to filter the displayed production transactions by ProductName attribute value.																															
Production Transactions	Table	Table containing all the report details retrieved.																															
Product Produced	Table Heading	Houses the ProductName attribute value for each production																															

					transaction record retrieved.										
			Quantity Before	Table Heading	Houses the QuantityBefore attribute value for each production transaction record retrieved.										
			Quantity Produced	Table Heading	Displays the calculated quantity produced value for each production transaction record retrieved.										
			Quantity After	Table Heading	Houses the QuantityAfter attribute value for each production transaction record retrieved.										
			Transaction Date	Table Heading	Displays the TransactionDate attribute value for each production transaction record retrieved.										
			Generate Report	Button	Allows the user to refresh the report details using new filter criteria when clicked.										
			Export	Button	Allows the user to export the Production Report in a PDF format.										
	Step 7: The administrator or inventory manager selects the Export button.		Step 8: The system exports the Produced Products Report report using a JsPDF script in the relevant Razor screen using the following layout and controls:												
			<table><tr><th>Details</th><th>Notes</th></tr><tr><td>Natuurlik Header</td><td>The company logo and address in png format from the www root folder</td></tr><tr><td>Report Title</td><td>“Produced Products Report”</td></tr><tr><td>Printed By</td><td>The authenticated user’s name and surname using the claim details retrieved from the User table.</td></tr><tr><td>Printed Date</td><td>The date in which the report is generated and populated by the system’s current date in (DD/MM/YYYY) format.</td></tr></table>			Details	Notes	Natuurlik Header	The company logo and address in png format from the www root folder	Report Title	“Produced Products Report”	Printed By	The authenticated user’s name and surname using the claim details retrieved from the User table.	Printed Date	The date in which the report is generated and populated by the system’s current date in (DD/MM/YYYY) format.
			Details	Notes											
			Natuurlik Header	The company logo and address in png format from the www root folder											
			Report Title	“Produced Products Report”											
			Printed By	The authenticated user’s name and surname using the claim details retrieved from the User table.											
Printed Date	The date in which the report is generated and populated by the system’s current date in (DD/MM/YYYY) format.														

			Body – Table	Populated with a table of the generated production details.													
			Footer Section	“End of Report” and page number(s)													
ALTERNATE COURSES	ALT Step 3: The administrator or inventory manager wishes to apply a filter criteria to the report to be generated: - The administrator or inventory manager enters the the filter criteria in any or all of the following fields: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Start Date</td><td>Date Picker</td><td>Specify the Start Date value to filter all production transaction by using the CreatedDate attribute value.</td></tr><tr><td>End Date</td><td>Date Picker</td><td>Specify the End Date value to filter all production transaction by using the CreatedDate attribute value.</td></tr><tr><td>Product Name</td><td>Drop-Down List</td><td>Filter production transactions by selecting a single product's name value. List Populated by retrieving the ProductName attribute values of all products in the Product entity.</td></tr></table> - The administrator or inventory manager clicks on the Generate Report button. - Proceed to Step 4.					Control Name	Control Type	Notes	Start Date	Date Picker	Specify the Start Date value to filter all production transaction by using the CreatedDate attribute value.	End Date	Date Picker	Specify the End Date value to filter all production transaction by using the CreatedDate attribute value.	Product Name	Drop-Down List	Filter production transactions by selecting a single product's name value. List Populated by retrieving the ProductName attribute values of all products in the Product entity.
Control Name	Control Type	Notes															
Start Date	Date Picker	Specify the Start Date value to filter all production transaction by using the CreatedDate attribute value.															
End Date	Date Picker	Specify the End Date value to filter all production transaction by using the CreatedDate attribute value.															
Product Name	Drop-Down List	Filter production transactions by selecting a single product's name value. List Populated by retrieving the ProductName attribute values of all products in the Product entity.															
	ALT Step 5: The administrator or inventory manager has specified a filter criteria for the report to be generated: The system uses a .NET Controller with a SQL Read query to retrieve the following details for all production transaction records in the ProductionTransaction entity that has met the report filter criteria captured: <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Product ID</td><td>ProductId</td><td>The unique identifier value of the product(s) which has been produced for all</td></tr></table>					Details to be read	Attributes in the table	Notes	Product ID	ProductId	The unique identifier value of the product(s) which has been produced for all						
Details to be read	Attributes in the table	Notes															
Product ID	ProductId	The unique identifier value of the product(s) which has been produced for all															

		production transaction records retrieved.
Quantity Before	QuantityBefore	The QuantityOnHand value of each product before the production activity took place.
Quantity After	QuantityAfter	The QuantityOnHand value of each product after the production activity took place.
Transaction Date	TransactionDate	The date and time which each production activity was captured on the system.

- The system retrieves the following details for each finished product from the **Product** entity which is referenced in a production transaction record retrieved:

Details to be read	Attributes in the table	Notes
Product Name	ProductName	The name of each product which was retrieved.

- The system retrieves the user claims details from the **User** entity:

Details to be read	Attributes in the table	Notes
Name	Name	The first name of the administrator or inventory manager making the request to generate the report.
Surname	Surname	The surname of the administrator or inventory manager requesting to generate the report.

	• Proceed to Step 5.
CONCLUSION:	The system displays the generated production finished products report details on the Produced Products Report screen.
POST-CONDITION:	• The system has retrieved production transaction records from the ProductionTransaction entity based on the filter criteria specified. • A table is populated with all the details retrieved for each production transaction record that has met the specified criteria.
BUSINESS RULES:	• Only the system administrator or inventory manager must be allowed to generate and view the production report.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS :	• The system should allow the administrator or inventory manager to optionally specify a filter criteria for the report to be generated. • The report should be able to be exported. • The production report cannot be generated if the system is offline. • The exported report should include details on who generated the report and the date when it was generated.
ASSUMPTIONS:	• The administrator or inventory manager has access to the internet. • The administrator or inventory manager will want to export the report after viewing it on the system.
OPEN ISSUES:	• None

Table 115 - 10.7 Generate Orders (by Status) Report

REPORTING SUBSYSTEM			
AUTHOR(s): Kyle van Eeden		DATE: 08/08/2022	
VERSION: 1.1		LAST REVIEW DATE: 26/08/2022	
USE CASE NAME:	View Orders (By Status) Report	USE CASE TYPE	
USE CASE ID:	10.7	Business Requirements: ☐	
PRIORITY:	Low	System Analysis: ☐	
SOURCE:	ByteXecom Requirements List	System Design: ☒	
PRIMARY BUSINESS ACTOR:	Administrator		
PRIMARY THE SYSTEM ACTOR:	None		
OTHER PARTICIPATING ACTORS:	Power BI Service (ESA)		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This use case describes the process in which the administrator wishes to view the total number of orders within a particular order status and a list which shows the number of orders and the accumulated totals grouped by order status and order payment status. The administrator clicks the Orders By Status Report hyperlink and the system will retrieve all orders information from the Order entity, then count the total number of orders assigned to each distinct status. All orders are then grouped by order status and order payment status along with the accumulated order totals per grouping. The use case concludes once the generated report is retrieved from the Power BI Service workspace configured and the report details is displayed on the Orders By Status Report screen.		
PRE-CONDITION:	- The administrator must be logged into their account. - The administrator must have a Power BI Service account which has access to the Natuurlik Reporting workspace.		
TRIGGER:	The administrator wishes to view the latest Orders By Status report on the system.		
TYPICAL COURSE OF EVENTS:	SYSTEM RESPONSE:		
	ACTOR ACTION:	Manual Action	Automated Action
	Step 1: The administrator selects the Orders By Status Report		Step 2: The following details are retrieved from the Order entity using a SQL Read query:

navigation link under the 'Reports' section in the side-navigation menu.			Details to be read	Attribute in Table	Notes
			Order Number	OrderId	The unique reference number assigned to each order retrieved.
			Order Status	OrderStatus	The current status assigned to each order retrieved.
			Order Payment Status	OrderPayment Status	The current payment status assigned to each order retrieved.
			Order Total	OrderTotal	The order total value for each order retrieved.
			Step 3: The count of all orders per status are calculated and populated within the following table that is generated:		
			Control Name	Attribute in Table	Notes
			Order Status	Table Heading	This fields holds the names all distinct order statuses retrieved.
			Order Status	Table Row	Populated with distinct OrderStatus attribute values retrieved.
			Total Orders	Table Heading	This field will be used to store the total number of orders that have a specific order status assigned.
			Total Orders	Table Row	The sum of all order totals per OrderStatus is calculated.

			Step 4: The system displays the Orders By Status Report screen which contains the embedded Orders By Status Power BI Report retrieved from the Natuurlik Reporting Power BI workspace, and contains the following controls:		
			Control Name	Control Type	Notes
			'Orders By Status Report'	Heading	Main heading of the screen.
			Order Status Summary	Table	A table is displayed to show the count of orders assigned to each status.
			Status	Table Heading	Populated with the distinct [OrderStatus] attribute values retrieved.
			Total	Table Heading	Populated with the values used to calculate the count of orders assigned to each status.
			Order Details	Table	A table is displayed which groups all orders by their assigned [OrderStatus] value and [OrderPaymentStatus] attribute value.
			Status	Table Heading	Populated with the order statuses retrieved. Used as a parent group to group orders by.
			Payment Status	Table Heading	Populated with the order payment statuses retrieved. Used as a child group to further group orders by.
			Order Number	Table Heading	Populated with the [Orderid] attribute values of all orders retrieved and placed within the correct groups.

			Order Total	Table Heading	Populated with the [OrderTotal] attribute values of all orders retrieved. Also displays the summed totals of all orders within a particular grouping.
			Export	Button	Allows the administrator export the report to a PDF formatted file.
			All orders retrieved are grouped by using the [OrderStatus] attribute value for the parent grouping operation and are further grouped by using the [OrderPaymentStatus] attribute value as the child group. The [OrderTotal] values are aggregated to calculate the totals for each group of orders using a sum function in Power BI Report builder.		
ALTERNATE COURSES	ALT Step 5: The administrator does not wish to export the report: The use case concludes.				
CONCLUSION:	The system displays the Orders By Status Report screen which is loaded with the embedded Orders By Status report retrieved from the Power BI Service.				
POST-CONDITION:	The Orders By Status Report is generated using data retrieved from the Order entity through using the Power BI Service and it is displayed on the system.				
BUSINESS RULES:	- Only the system administrator should be permitted to view the Orders By Status Report on the system. - The system administrator requires the report to display the total number of orders within each status. - The report should make use of logical groupings to display the aggregated order totals for all orders within a specific status. - The system administrator should be allowed to export the report.				
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	- All orders should to be categorized by order status and order payment status. - The running totals of all groupings of orders should be calculated correctly and displayed with the generated report. - The report can only be exported as a PDF document.				
ASSUMPTIONS:	- The administrator has access to the internet. - This report will help the administrator prioritize orders by clearly identifying what orders need to be attended to first.				
OPEN ISSUES:	None				

Table 116 - 10.8 Generate Orders (by Region) Report

REPORTING SUBSYSTEM			
AUTHOR(s): Kyle van Eeden		DATE: 08/08/2022	
VERSION: 1.0		LAST REVIEW DATE: 26/08/2022	
USE CASE NAME:	Generate Orders (By Region) Report	USE CASE TYPE	
USE CASE ID:	10.8	Business Requirements:	☐
PRIORITY:	Low	System Analysis:	☐
SOURCE:	ByteXecom Requirements List	System Design:	☒
PRIMARY BUSINESS ACTOR:	Administrator		
PRIMARY THE SYSTEM ACTOR:	None		
OTHER PARTICIPATING ACTORS:	Power BI Service (ESA)		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This use case describes the process in which the administrator wishes to view a report containing details about how many orders were or are to be delivered to a particular province or city, the total number of orders per province, and the total revenue that was generated by all orders for a particular province and city. The report will be generated using orders details retrieved from the Order entity, and all province and city names associated to each particular order from the Province and City entities, respectively. Power BI Report Builder is used to generate the report using the latest retrieved data upon making the request to view the Orders By Regions Report. The use case concludes once the generated report is embedded and displayed on the Orders By Regions Report screen.		
PRE-CONDITION:	- The administrator must be logged into their account. - The administrator must have a Power BI Service account which has access to the Natuurlik Reporting workspace.		
TRIGGER:	The administrator wishes to view the latest Orders By Regions report on the system.		
TYPICAL COURSE OF EVENTS:	SYSTEM RESPONSE:		
	ACTOR ACTION:	Manual Action	Automated Action
	Step 1: The administrator selects the Orders By		Step 2: The following details are retrieved from the Order entity using a SQL Read query:

</

			Step 3: The system displays the Orders By Region Report screen which contains the embedded iframe for the Orders By Region Power BI report retrieved from the Natuurlik Reporting Power BI workspace, and contains the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>'Orders By Region Report'</td><td>Heading</td><td>Main heading of the screen.</td></tr><tr><td>Orders By Province Summary</td><td>Table</td><td>A table is displayed to show the count of orders retrieved for each province.</td></tr><tr><td>Province</td><td>Table Heading</td><td>Populated with the distinct [ProvinceName] attribute values retrieved.</td></tr><tr><td>Total</td><td>Table Heading</td><td>Populated with the values used to calculate the count of orders assigned to each distinct province.</td></tr><tr><td>Order Details</td><td>Table</td><td>A table is displayed which groups all orders by their assigned [ProvinceName] and [CityName] attribute values.</td></tr><tr><td>Order Number</td><td>Table Heading</td><td>Populated with the [OrderId] attribute values of all orders retrieved and placed within the correct groups based on its [OrderStatus] attribute value assigned.</td></tr><tr><td>Order Total</td><td>Table Heading</td><td>Populated with the [OrderTotal] attribute values of all orders retrieved. Used to calculate the aggregated total of all orders within a particular parent-child grouping.</td></tr><tr><td>Export</td><td>Button</td><td>Allows the administrator export the report to all desired file</td></tr></table>	Control Name	Control Type	Notes	'Orders By Region Report'	Heading	Main heading of the screen.	Orders By Province Summary	Table	A table is displayed to show the count of orders retrieved for each province.	Province	Table Heading	Populated with the distinct [ProvinceName] attribute values retrieved.	Total	Table Heading	Populated with the values used to calculate the count of orders assigned to each distinct province.	Order Details	Table	A table is displayed which groups all orders by their assigned [ProvinceName] and [CityName] attribute values.	Order Number	Table Heading	Populated with the [OrderId] attribute values of all orders retrieved and placed within the correct groups based on its [OrderStatus] attribute value assigned.	Order Total	Table Heading	Populated with the [OrderTotal] attribute values of all orders retrieved. Used to calculate the aggregated total of all orders within a particular parent-child grouping.	Export	Button	Allows the administrator export the report to all desired file
Control Name	Control Type	Notes																												
'Orders By Region Report'	Heading	Main heading of the screen.																												
Orders By Province Summary	Table	A table is displayed to show the count of orders retrieved for each province.																												
Province	Table Heading	Populated with the distinct [ProvinceName] attribute values retrieved.																												
Total	Table Heading	Populated with the values used to calculate the count of orders assigned to each distinct province.																												
Order Details	Table	A table is displayed which groups all orders by their assigned [ProvinceName] and [CityName] attribute values.																												
Order Number	Table Heading	Populated with the [OrderId] attribute values of all orders retrieved and placed within the correct groups based on its [OrderStatus] attribute value assigned.																												
Order Total	Table Heading	Populated with the [OrderTotal] attribute values of all orders retrieved. Used to calculate the aggregated total of all orders within a particular parent-child grouping.																												
Export	Button	Allows the administrator export the report to all desired file																												

					formats supported by Power BI.
ALTERNATE COURSES	None				
CONCLUSION:	The system displays the Orders By Region Report screen which is loaded with the embedded Orders By Status report retrieved from the Natuurlik Reporting Power BI Service workspace.				
POST-CONDITION:	- All required order details are retrieved from the Order entity where the [OrderStatus] attribute value of the order is equal to either "Processing" or "Dispatched". - The province and city that is associated with the retrieved order's delivery address is retrieved from the Province and City entities, respectively. - Power BI Report Builder refreshes the data source at the time of the request to view the report and a new report is generated using the latest relevant orders data. - The generated report is embedded and displayed on the Orders By Region Report screen.				
BUSINESS RULES:	- Only the system administrator should be permitted to view the Orders By Region Report on the system. - The system administrator requires the report to display a summary of the total orders associated with each distinct province. - The report should make use of logical groupings to display the aggregated order totals for all orders within a specific region. - The system administrator should be allowed to export the report in order to save it to their local machine.				
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	- Only orders with a status of "Processing", "Delivered" and "Dispatched" are to be retrieved from the Order entity. - All orders should be categorized by its associated province and city. - The running totals of all groupings of orders should be calculated correctly and displayed with the generated report. - The administrator no longer has access to a Power BI Service account that has access to the Natuurlik Reporting workspace. - The report will only be updated every 2 hours until the ByteXecom system has been deployed.				
ASSUMPTIONS:	- The administrator has access to the internet. - The administrator has access to the Natuurlik Reporting workspace within Power BI Service.				
OPEN ISSUES:	None				

Table 117 - 10.9 Generate Sales (by Product) Report

REPORTING SUBSYSTEM								
AUTHORS (s): Ofhani Mungani		DATE: 08/08/2022						
VERSION: 1.0		LAST REVIEW DATE: 25/08/2022						
USE CASE NAME:	Generate Sales (by Product) Report	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒						
USE CASE ID:	10.9							
PRIORITY:	High							
SOURCE:	ByteXcom Requirements List							
PRIMARY BUSINESS ACTOR	• Administrator/Sales Assistant							
PRIMARY SYSTEM ACTOR	• None							
OTHER PARTICIPATING ACTORS:	• None							
OTHER INTERESTED STAKEHOLDERS:	• None							
DESCRIPTION:	This use case describes an event where the administrator or sales assistant wants to generate sales (by product) report on the system. The use case begins when the administrator/ sales assistant requests to generate sales (by product) report and the system responds by retrieving the order details. The use case concludes when the report is displayed on the screen.							
PRE-CONDITION:	• The administrator or sales assistant must be logged on the system.							
TRIGGER:	The use case is called when the administrator or sales assistant wants to generate sales (by product) report							
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE						
		MANUAL ACTION	AUTOMATED ACTION					
	Step 1: The administrator or sales assistant clicks on the “Sales Report” in the Reports dropdown on the admin side navigation bar.		Step 2: The system retrieves the following attributes using a method in a .NET controller and an SQL READ query: From the Product entity:					
			<table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Name</td><td>Name</td></tr></table>		Details to be read	Attributes in the table	Name	Name
			Details to be read	Attributes in the table				
			Name	Name				
From the OrderLine entity:								
<table><tr><td>Details to be read</td><td>Attributes in the table</td></tr></table>		Details to be read	Attributes in the table					
Details to be read	Attributes in the table							

			<table><tr><td>Price</td><td>Price</td></tr><tr><td>Count</td><td>Count</td></tr></table>	Price	Price	Count	Count								
			Price	Price											
			Count	Count											
From the Order entity:															
<table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>OrderTotal</td><td>OrderTotal</td></tr><tr><td>FirstName</td><td>FirstName</td></tr><tr><td>Surname</td><td>Surname</td></tr><tr><td>DeliveryFee</td><td>DeliveryFee</td></tr><tr><td>CreatedDate</td><td>CreatedDate</td></tr></table>	Details to be read	Attributes in the table	OrderTotal	OrderTotal	FirstName	FirstName	Surname	Surname	DeliveryFee	DeliveryFee	CreatedDate	CreatedDate			
Details to be read	Attributes in the table														
OrderTotal	OrderTotal														
FirstName	FirstName														
Surname	Surname														
DeliveryFee	DeliveryFee														
CreatedDate	CreatedDate														
			Step 3: The system calculates the sales values using the following retrieved attributes in a method using .NET Controller. From the OrderLine entity: Sales =Price × Count TotalSales = Sum of Sales TotalProductQuantity = Sum of Count From the Order entity: Sales =OrderTotal – DeliveryFee TotalSales = Sum of Sales												
			Step 4: The system displays the Sales Dashboard Screen with the following controls:												
			<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Sale Report</td><td>heading</td><td>This control will be used to display the heading of the screen</td></tr><tr><td>Sales by Top 5 Clients</td><td>button</td><td>This control will be used to navigate to Sales by Top 5 Clients Screen</td></tr><tr><td>Sales by Product</td><td>button</td><td>This control will be used to navigate to Sales by Product Screen</td></tr><tr><td>Monthly Sales</td><td>button</td><td>This control will be used to navigate to Sales by Month Screen</td></tr></table>	Control Name	Control Type	Notes	Sale Report	heading	This control will be used to display the heading of the screen	Sales by Top 5 Clients	button	This control will be used to navigate to Sales by Top 5 Clients Screen	Sales by Product	button	This control will be used to navigate to Sales by Product Screen
Control Name	Control Type	Notes													
Sale Report	heading	This control will be used to display the heading of the screen													
Sales by Top 5 Clients	button	This control will be used to navigate to Sales by Top 5 Clients Screen													
Sales by Product	button	This control will be used to navigate to Sales by Product Screen													
Monthly Sales	button	This control will be used to navigate to Sales by Month Screen													
			Bar Graph:												

			Sales	dataset	This control will be used to display the value of sales made by each product calculated by $\text{Price} \times \text{Count}$ from the OrderLine table
			Line Graph:		
			Control Name	Control Type	Notes
			Sales by Month Report	heading	This control will be used to show the main heading for the line graph section
			Rands Value of Sales	label	This control will be used as the chart label of the line graph
			x-axis	label	This control will be used as a label to display the month and year
			y-axis	label	This control will be used as a label to display the amounts of sales in rands
			Sales	dataset	This control will be used to display the value of sales made by each customer and reseller calculated by $\text{OrderTotal} - \text{DeliveryFee}$ from the Order table

	Step 5: The administrator or sales assistant clicks on the Sales by Product button		Step 6: The system retrieves the following attributes using a method in a .NET controller and an SQL READ query: From the Product entity: <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Name</td><td>Name</td></tr></table> From the OrderLine entity: <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Price</td><td>Price</td></tr><tr><td>Count</td><td>Count</td></tr></table> And the following claims details from the User entity for the administrator or sales assistant who made the request <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>FirstName</td><td>FirstName</td></tr><tr><td>Surname</td><td>Surname</td></tr></table>	Details to be read	Attributes in the table	Name	Name	Details to be read	Attributes in the table	Price	Price	Count	Count	Details to be read	Attributes in the table	FirstName	FirstName	Surname	Surname
Details to be read	Attributes in the table																		
Name	Name																		
Details to be read	Attributes in the table																		
Price	Price																		
Count	Count																		
Details to be read	Attributes in the table																		
FirstName	FirstName																		
Surname	Surname																		
			Step 7: The system calculates the sales values using the following retrieved attributes in a method using .NET Controller. From the OrderLine entity: Sales =Price × Count TotalSales = Sum of Sales TotalProductQuantity = Sum of Count																
			Step 8: The system generates the sales report and displays the Sales by Product Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Sale (by Product) Report</td><td>heading</td><td>This control will be used to display the heading of the screen</td></tr><tr><td>Back</td><td>button</td><td>This control will be used to navigate back to the sales dashboard</td></tr></table>	Control Name	Control Type	Notes	Sale (by Product) Report	heading	This control will be used to display the heading of the screen	Back	button	This control will be used to navigate back to the sales dashboard							
Control Name	Control Type	Notes																	
Sale (by Product) Report	heading	This control will be used to display the heading of the screen																	
Back	button	This control will be used to navigate back to the sales dashboard																	

			Export	button	This control will be used to download the report in Portable Document Format (pdf)
			Pie Chart:		
			Control Name	Control Type	Notes
			Sales by Product Report	heading	This control will be used to show the main heading for the pie chart section
			Product Name	label	This control will be used to indicate the products name occupying the pie chart
			Sales	dataset	This control will be used to display the value of sales made by each product calculated by Price × Count from the OrderLine table
			Control Name	Control Type	Notes
			Results table	Table	This control will be used to give detail of the sale Pie Chart
			Product Name	Table Heading	This control will be used as a heading to display the product name
			Product Quantity	Table Heading	This control will be used as a heading to display the product quantity
			Sales Value (ZAR)	Table Heading	This control will be used as a heading to display the product's total sales

			<table><tr><td>Total</td><td>Table Row</td><td>This control will be used as the last row of the table to display total value of sales</td></tr></table>	Total	Table Row	This control will be used as the last row of the table to display total value of sales										
	Total	Table Row	This control will be used as the last row of the table to display total value of sales													
Step 9: The administrator or sales assistant clicks on the Export button.		Step 10: The system exports the report using a JsPDF script in the relevant Razor page using the following layout and controls: <table><tr><th>Details</th><th>Notes</th></tr><tr><td>Natuurlik Header</td><td>The company logo and address in png format from the www root folder</td></tr><tr><td>Report Title</td><td>Sales (by Product) Report</td></tr><tr><td>Printed by</td><td>The authenticated user's name and surname from the claim details in the User table.</td></tr><tr><td>Printed Date</td><td>The date in which the report is generated and populated by the system's current date in (DD/MM/YYYY) format.</td></tr><tr><td>Body – Charts with Tables</td><td>Populated with the sales details.</td></tr><tr><td>Footer Section</td><td>"End of Report" and page number(s)</td></tr></table>	Details	Notes	Natuurlik Header	The company logo and address in png format from the www root folder	Report Title	Sales (by Product) Report	Printed by	The authenticated user's name and surname from the claim details in the User table.	Printed Date	The date in which the report is generated and populated by the system's current date in (DD/MM/YYYY) format.	Body – Charts with Tables	Populated with the sales details.	Footer Section	"End of Report" and page number(s)
Details	Notes															
Natuurlik Header	The company logo and address in png format from the www root folder															
Report Title	Sales (by Product) Report															
Printed by	The authenticated user's name and surname from the claim details in the User table.															
Printed Date	The date in which the report is generated and populated by the system's current date in (DD/MM/YYYY) format.															
Body – Charts with Tables	Populated with the sales details.															
Footer Section	"End of Report" and page number(s)															
ALTERNATE COURSES:	None.															
CONCLUSION:	The use case concludes when the sales report is downloaded															
POST-CONDITION:	Sales details is read from the database using the Order, OrderLine, Product tables.															
BUSINESS RULES	None.															
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	The administrator or sales assistant cannot generate sales report without an internet connection.															
ASSUMPTIONS:	The administrator will always export the report															
OPEN ISSUES:	None															

Table 118 - 10.10 Generate Sales (by Month) Report

REPORTING SUBSYSTEM													
AUTHORS (s): Ofhani Mungani		DATE: 08/08/2022											
VERSION: 1.0		LAST REVIEW DATE: 25/08/2022											
USE CASE NAME:	Generate Sales (by Month) Report	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒											
USE CASE ID:	10.10												
PRIORITY:	High												
SOURCE:	ByteXcom Requirements List												
PRIMARY BUSINESS ACTOR	Administrator/Sales Assistant												
PRIMARY SYSTEM ACTOR	None												
OTHER PARTICIPATING ACTORS:	None												
OTHER INTERESTED STAKEHOLDERS:	None												
DESCRIPTION:	This use case describes an event where the administrator or sales assistant wants to generate sales (by month) report on the system. The use case begins when the administrator/ sales assistant requests to generate sales (by month) report and the system responds by retrieving the order details. The use case concludes when the report is displayed on the screen.												
PRE-CONDITION:	The administrator or sales assistant must be logged on the system.												
TRIGGER:	The use case is called when the administrator or sales assistant wants to generate sales (by month) report												
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE											
		MANUAL ACTION	AUTOMATED ACTION										
	Step 1: The administrator or sales assistant clicks on the “Sales Report” in the Reports dropdown on the admin side navigation bar.		Step 2: The system retrieves the following attributes using a method in a .NET controller and an SQL READ query: From the Product entity: <table border="1"> <thead> <tr> <th>Details to be read</th> <th>Attributes in the table</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>Name</td> </tr> </tbody> </table> From the OrderLine entity: <table border="1"> <thead> <tr> <th>Details to be read</th> <th>Attributes in the table</th> </tr> </thead> <tbody> <tr> <td>Price</td> <td>Price</td> </tr> <tr> <td>Count</td> <td>Count</td> </tr> </tbody> </table> From the Order entity:	Details to be read	Attributes in the table	Name	Name	Details to be read	Attributes in the table	Price	Price	Count	Count
Details to be read	Attributes in the table												
Name	Name												
Details to be read	Attributes in the table												
Price	Price												
Count	Count												

			<table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>OrderTotal</td><td>OrderTotal</td></tr><tr><td>FirstName</td><td>FirstName</td></tr><tr><td>Surname</td><td>Surname</td></tr><tr><td>DeliveryFee</td><td>DeliveryFee</td></tr><tr><td>CreatedDate</td><td>CreatedDate</td></tr></table>	Details to be read	Attributes in the table	OrderTotal	OrderTotal	FirstName	FirstName	Surname	Surname	DeliveryFee	DeliveryFee	CreatedDate	CreatedDate						
Details to be read	Attributes in the table																				
OrderTotal	OrderTotal																				
FirstName	FirstName																				
Surname	Surname																				
DeliveryFee	DeliveryFee																				
CreatedDate	CreatedDate																				
			Step 3: The system calculates the sales values using the following retrieved attributes in a method using .NET Controller. From the OrderLine entity: Sales =Price × Count TotalSales = Sum of Sales TotalProductQuantity = Sum of Count From the Order entity: Sales =OrderTotal – DeliveryFee TotalSales = Sum of Sales																		
			Step 4: The system displays the Sales Dashboard Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Sale Report</td><td>heading</td><td>This control will be used to display the heading of the screen</td></tr><tr><td>Sales by Top 5 Clients</td><td>button</td><td>This control will be used to navigate to Sales by Top 5 Clients Screen</td></tr><tr><td>Sales by Product</td><td>button</td><td>This control will be used to navigate to Sales by Product Screen</td></tr><tr><td>Monthly Sales</td><td>button</td><td>This control will be used to navigate to Sales by Month Screen</td></tr></table> Bar Graph: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr></table>	Control Name	Control Type	Notes	Sale Report	heading	This control will be used to display the heading of the screen	Sales by Top 5 Clients	button	This control will be used to navigate to Sales by Top 5 Clients Screen	Sales by Product	button	This control will be used to navigate to Sales by Product Screen	Monthly Sales	button	This control will be used to navigate to Sales by Month Screen	Control Name	Control Type	Notes
Control Name	Control Type	Notes																			
Sale Report	heading	This control will be used to display the heading of the screen																			
Sales by Top 5 Clients	button	This control will be used to navigate to Sales by Top 5 Clients Screen																			
Sales by Product	button	This control will be used to navigate to Sales by Product Screen																			
Monthly Sales	button	This control will be used to navigate to Sales by Month Screen																			
Control Name	Control Type	Notes																			

			Top 5 Sales by Client Report	heading	This control will be used to show the main heading for the bar graph section
			Rands Value of Sales	label	This control will be used as the chart label of the bar graph
			x-axis	label	This control will be used as a label to display the name and surname for the top five clients
			y-axis	label	This control will be used as a label to display the amounts of sales in rands
			Sales	dataset	This control will be used to display the value of sales made by each customer and reseller in the top five calculated by OrderTotal - DeliveryFee from the Order table
			Pie Chart:		
			Control Name	Control Type	Notes
			Sales By Product Report	heading	This control will be used to show the main heading for the pie chart section
			Product Name	label	This control will be used to indicate the products name occupying the pie chart
			Sales	dataset	This control will be used to display the value of sales made by each product calculated by Price × Count

					from the OrderLine table
			Line Graph:		
			Control Name	Control Type	Notes
			Sales by Month Report	heading	This control will be used to show the main heading for the line graph section
			Rands Value of Sales	label	This control will be used as the chart label of the line graph
			x-axis	label	This control will be used as a label to display the month and year
			y-axis	label	This control will be used as a label to display the amounts of sales in rands
			Sales	dataset	This control will be used to display the value of sales made by each customer and reseller calculated by OrderTotal - DeliveryFee from the Order table

	Step 5: The administrator or sales assistant clicks on the Monthly Sales button		Step 6: The system retrieves the following attributes using a method in a .NET controller and an SQL READ query: From the Order entity: <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>OrderTotal</td><td>OrderTotal</td></tr><tr><td>DeliveryFee</td><td>DeliveryFee</td></tr><tr><td>CreatedDate</td><td>CreatedDate</td></tr></table> And the following claims details from the User entity for the administrator or sales assistant who made the request <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>FirstName</td><td>FirstName</td></tr><tr><td>Surname</td><td>Surname</td></tr></table>	Details to be read	Attributes in the table	OrderTotal	OrderTotal	DeliveryFee	DeliveryFee	CreatedDate	CreatedDate	Details to be read	Attributes in the table	FirstName	FirstName	Surname	Surname
Details to be read	Attributes in the table																
OrderTotal	OrderTotal																
DeliveryFee	DeliveryFee																
CreatedDate	CreatedDate																
Details to be read	Attributes in the table																
FirstName	FirstName																
Surname	Surname																
			Step 7: The system calculates the sales values using the following retrieved attributes in a method using .NET Controller. From the OrderLine entity: Sales =Price × Count TotalSales = Sum of Sales TotalProductQuantity = Sum of Count														
			Step 8: The system generates the sales report and displays the Sales by Month Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Sale (by Month) Report</td><td>heading</td><td>This control will be used to display the heading of the screen</td></tr><tr><td>Back</td><td>button</td><td>This control will be used to navigate back to the sales dashboard</td></tr><tr><td>Export</td><td>button</td><td>This control will be used to download the report in Portable</td></tr></table>	Control Name	Control Type	Notes	Sale (by Month) Report	heading	This control will be used to display the heading of the screen	Back	button	This control will be used to navigate back to the sales dashboard	Export	button	This control will be used to download the report in Portable		
Control Name	Control Type	Notes															
Sale (by Month) Report	heading	This control will be used to display the heading of the screen															
Back	button	This control will be used to navigate back to the sales dashboard															
Export	button	This control will be used to download the report in Portable															

					Document Format (pdf)
			Line Graph:		
			Control Name	Control Type	Notes
			Monthly Sales Report	heading	This control will be used to show the main heading for the line graph section
			Rands Value of Sales	label	This control will be used as the chart label of the line graph
			x-axis	label	This control will be used as a label to display the month and year
			y-axis	label	This control will be used as a label to display the amounts of sales in rands
			Sales	dataset	This control will be used to display the value of sales made by each customer and reseller calculated by OrderTotal - DeliveryFee from the Order table
			Control Name	Control Type	Notes
			Results table	Table	This control will be used to give detail of the sale line graph
			Month	Table Heading	This control will be used as a heading to display the month name
			Year	Table Heading	This control will be used as a heading to display the year

			Sales Value (ZAR)	Table Heading	This control will be used as a heading to display the month total sales													
			Total	Table Row	This control will be used as the last row of the table to display total value of sales													
	Step 9: The administrator or sales assistant clicks on the Export button.		Step 10: The system exports the report using a JsPDF script in the relevant Razor page using the following layout and controls: <table><tr><th>Details</th><th>Notes</th></tr><tr><td>Natuurlik Header</td><td>The company logo and address in png format from the www root folder</td></tr><tr><td>Report Title</td><td>Sales (by Month) Report</td></tr><tr><td>Printed by</td><td>The authenticated user's name and surname from the claim details in the User table.</td></tr><tr><td>Printed Date</td><td>The date in which the report is generated and populated by the system's current date in (DD/MM/YYYY) format.</td></tr><tr><td>Body – Charts with Tables</td><td>Populated with the sales details.</td></tr><tr><td>Footer Section</td><td>"End of Report" and page number(s)</td></tr></table>			Details	Notes	Natuurlik Header	The company logo and address in png format from the www root folder	Report Title	Sales (by Month) Report	Printed by	The authenticated user's name and surname from the claim details in the User table.	Printed Date	The date in which the report is generated and populated by the system's current date in (DD/MM/YYYY) format.	Body – Charts with Tables	Populated with the sales details.	Footer Section
Details	Notes																	
Natuurlik Header	The company logo and address in png format from the www root folder																	
Report Title	Sales (by Month) Report																	
Printed by	The authenticated user's name and surname from the claim details in the User table.																	
Printed Date	The date in which the report is generated and populated by the system's current date in (DD/MM/YYYY) format.																	
Body – Charts with Tables	Populated with the sales details.																	
Footer Section	"End of Report" and page number(s)																	
ALTERNATE COURSES:	None.																	
CONCLUSION:	The use case concludes when the sales report is downloaded																	
POST-CONDITION:	Sales details is read from the database using the Order , OrderLine , Product tables.																	
BUSINESS RULES	None.																	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	The administrator or sales assistant cannot generate sales report without an internet connection.																	
ASSUMPTIONS:	The administrator will always export the report																	
OPEN ISSUES:	None																	

Table 119 - 10.11 View Dashboard Overview

REPORTING SUBSYSTEM			
AUTHOR(s): Kyle van Eeden		DATE: 08/08/2022	
VERSION: 1.1		LAST REVIEW DATE: 26/08/2022	
USE CASE NAME:	View Dashboard Overview	USE CASE TYPE	
USE CASE ID:	10.11	Business Requirements:	☐
PRIORITY:	Low	System Analysis:	☐
SOURCE:	ByteXecom Requirements List	System Design:	☒
PRIMARY BUSINESS ACTOR:	Administrator		
PRIMARY THE SYSTEM ACTOR:	None		
OTHER PARTICIPATING ACTORS:	Power BI Service (ESA)		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This use case describes the process in which the administrator wishes to view a dashboard overview which contains import measures and metrics using data retrieved from the database. The reporting workspace in Power BI Service has a dashboard configured which uses details retrieved from the Order, OrderLine, User, Product, Category, Courier, Country, Province, City, Suburb, OrderProduct, OrderQuery, QueryReason, ReturnedProduct, ReturnReason and Brand entities. The latest details is retrieved from these entities and Power BI Service embeds the dashboard on the View Dashboard Overview screen. The use case concludes once the system displays the embedded dashboard.		
PRE-CONDITION:	- The administrator must be logged into their account. - The administrator must have a Power BI Service account which has access to the Natuurlik Reporting workspace. - The administrator must be logged into their Power BI Service account to view the report.		
TRIGGER:	The administrator wishes to view the latest dashboard overview on the system.		
TYPICAL COURSE OF EVENTS:	ACTOR ACTION:	SYSTEM RESPONSE:	
		Manual Action	Automated Action
	Step 1: The administrator selects the Manage Store option		Step 2: The following details are retrieved from the Order entity using a SQL Read query:

	to view the dashboard.		Details to be read	Attribute in Table	Notes
			Order Number	OrderId	The unique reference number assigned to each order retrieved.
			Order Status	OrderStatus	The current status that is assigned to each existing order that will be used to filter orders by.
			Order Payment Status	OrderPaymentStatus	The payment status associated with each order retrieved.
			Order Total	OrderTotal	The order total value retrieved for each order.
			The following details are retrieved from the OrderProduct entity:		
			Details to be read	Attribute in Table	Notes
			Order ID	OrderID	Uniquely identifies an order placed on the system.
			Product ID	ProductID	Uniquely identifies a product which is associated to each order.

			Product Quantity	ProductQuantity	The quantity of each product which was ordered for a particular order.
			Transaction Date	TransactionDate	The date on which the record was captured on the system.
			The following details are retrieved from the OrderQuery entity:		
			Details to be read	Attribute in Table	Notes
			Order ID	OrderID	Uniquely identifies an order placed on the system and which has a query associated.
			Query Description	QueryFeedback	Feedback provided by user who has placed the order.
			Query Status	QueryStatus	The status assigned to the order query.
			Date Logged	DateLogged	The date on which the order query was logged.
			The following details are retrieved from the QueryReason entity:		
			Details to be read	Attribute in Table	Notes

			Query Reason Name	QueryReasonName	The pre-defined reason for the query.
			The following details are retrieved from the ReturnedProduct entity:		
			Details to be read	Attribute in Table	Notes
			Order ID	OrderID	Uniquely identifies an order placed on the system and which has a product return transaction associated.
			Product ID	ProductID	Uniquely identifies the product that has been returned for each product return record.
			Quantity Received	QuantityReceived	The quantity of the product logged as returned.
			Return Reason ID	ReturnReasonID	The unique identifier to the reason for the product return.
			The following details are retrieved from the ReturnReason entity:		
			Details to be read	Attribute in Table	Notes
			Return Reason Name	ReturnReasonName	The name of each return reason.

The following details are retrieved from the **Country** entity:

Details to be read	Attribute in Table	Notes
Country Name	CountryName	The name value of each country which is associated with each order retrieved.

The following details are retrieved from the **Province** entity:

Details to be read	Attribute in Table	Notes
Province Name	ProvinceName	The name value of each province which is associated with each order retrieved.

The following details are retrieved from the **City** entity:

Details to be read	Attribute in Table	Notes
City Name	CityName	The name value of each city which is associated with each order retrieved.

The following details are retrieved from the **Suburb** entity:

Details to be read	Attribute in Table	Notes

			<table><tr><td>Suburb Name</td><td>SuburbName</td><td>The name value of each suburb which is associated with each order retrieved.</td></tr></table>	Suburb Name	SuburbName	The name value of each suburb which is associated with each order retrieved.						
Suburb Name	SuburbName	The name value of each suburb which is associated with each order retrieved.										
			The following details are retrieved from the User entity: <table><tr><th>Details to be read</th><th>Attribute in Table</th><th>Notes</th></tr><tr><td>First Name</td><td>FirstName</td><td>The name value of each user associated with an order.</td></tr><tr><td>Surname</td><td>Surname</td><td>The surname value of each user associated with an order.</td></tr></table>	Details to be read	Attribute in Table	Notes	First Name	FirstName	The name value of each user associated with an order.	Surname	Surname	The surname value of each user associated with an order.
Details to be read	Attribute in Table	Notes										
First Name	FirstName	The name value of each user associated with an order.										
Surname	Surname	The surname value of each user associated with an order.										
			The following details are retrieved from the OrderLine entity: <table><tr><th>Details to be read</th><th>Attribute in Table</th><th>Notes</th></tr><tr><td>Produce Name</td><td>CartItemQuantity</td><td>The quantity ordered of each product for a specific order.</td></tr><tr><td>Product Price</td><td>CartItemPrice</td><td>The price of each product ordered.</td></tr></table>	Details to be read	Attribute in Table	Notes	Produce Name	CartItemQuantity	The quantity ordered of each product for a specific order.	Product Price	CartItemPrice	The price of each product ordered.
Details to be read	Attribute in Table	Notes										
Produce Name	CartItemQuantity	The quantity ordered of each product for a specific order.										
Product Price	CartItemPrice	The price of each product ordered.										
			The following details are retrieved from the Product entity:									

			<table><tr><th>Details to be read</th><th>Attribute in Table</th><th>Notes</th></tr><tr><td>Product Name</td><td>ProductName</td><td>The name value of each product which is associated with each order retrieved.</td></tr></table>	Details to be read	Attribute in Table	Notes	Product Name	ProductName	The name value of each product which is associated with each order retrieved.
	Details to be read	Attribute in Table	Notes						
	Product Name	ProductName	The name value of each product which is associated with each order retrieved.						
			The following details are retrieved from the Category entity:						
			<table><tr><th>Details to be read</th><th>Attribute in Table</th><th>Notes</th></tr><tr><td>Category Name</td><td>CategoryName</td><td>The name value of each product category which is associated with product ordered.</td></tr></table>	Details to be read	Attribute in Table	Notes	Category Name	CategoryName	The name value of each product category which is associated with product ordered.
Details to be read	Attribute in Table	Notes							
Category Name	CategoryName	The name value of each product category which is associated with product ordered.							
		The following details are retrieved from the Brand entity:							
		<table><tr><th>Details to be read</th><th>Attribute in Table</th><th>Notes</th></tr><tr><td>Brand Name</td><td>BrandName</td><td>The name value of each product brand retrieved.</td></tr></table>	Details to be read	Attribute in Table	Notes	Brand Name	BrandName	The name value of each product brand retrieved.	
Details to be read	Attribute in Table	Notes							
Brand Name	BrandName	The name value of each product brand retrieved.							
		The following details are retrieved from the Courier entity:							
		<table><tr><th>Details to be read</th><th>Attribute in Table</th><th>Notes</th></tr><tr><td>Courier Name</td><td>CourierName</td><td>The name value of each courier.</td></tr></table>	Details to be read	Attribute in Table	Notes	Courier Name	CourierName	The name value of each courier.	
Details to be read	Attribute in Table	Notes							
Courier Name	CourierName	The name value of each courier.							

			Courier Fee	CourierFee	The courier service fee for each courier retrieved.
			Step 3: The system displays the View Dashboard Overview screen which contains the embedded iframe for the dashboard retrieved from the Natuurlik Reporting Power BI workspace.		
ALTERNATE COURSES	None				
CONCLUSION:	The system displays the View Dashboard Overview screen which is loaded with the embedded dashboard retrieved from the Natuurlik Reporting Power BI Service workspace.				
POST-CONDITION:	- The latest orders data is retrieved from the Order entity. - All details are retrieved from the following entities: OrderProduct, ReturnedProduct, ReturnReason, OrderQuery, QueryReason and Brand. - All user-related details is retrieved from the User entity. - All location-related details is referenced from the Country, Province, City and Suburb entities. - A dashboard is generated using all information retrieved from the database.				
BUSINESS RULES:	- Only the system administrator should be permitted to view the dashboard overview on the system. - Only administrators with a Power BI Service account with access to the Natuurlik Reporting workspace should be allowed to view the dashboard.				
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS:	- The administrator no longer has access to a Power BI Service account that has access to the Natuurlik Reporting workspace. - The report will only be updated every 2 hours until the ByteXecom system has been deployed.				
ASSUMPTIONS:	- The administrator has access to the internet. - The administrator has access to the Natuurlik Reporting workspace within Power BI Service.				
OPEN ISSUES:	None				

2.11. Supplier Subsystem

Table 120 - 11.1 Create Supplier

SUPPLIER SUBSYSTEM				
AUTHORS (s): Nomusa Vumisa		DATE: 14/05/2022		
VERSION: 1.0		LAST REVIEW DATE: 21/08/2022		
USE CASE NAME:	Create Supplier	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒		
USE CASE ID:	11.1			
PRIORITY:	High			
SOURCE:	ByteXcom Requirements List			
PRIMARY BUSINESS ACTOR	• Administrator			
PRIMARY SYSTEM ACTOR	• None			
OTHER PARTICIPATING ACTORS:	• None			
OTHER INTERESTED STAKEHOLDERS:	• None			
DESCRIPTION:	This use case describes the process in which the administrator wishes to add a new supplier to the system. The use case begins when the administrator wishes to add a new supplier on the system. The administrator will enter all the required information regarding the supplier and save it to the database. The use case ends when the administrator receives a successful creation notification.			
PRE-CONDITION:	• The administrator must be logged onto the system. • The supplier must not already exist on the system			
TRIGGER:	The use case is triggered when the administrator wants to create a new supplier.			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE		
		MANUAL ACTION	AUTOMATED ACTION	
	Step 1: The administrator clicks on the Supplier nav link on the admin side navigation bar		Step 2: The system retrieves the following attributes from each entry in the Supplier Table, using a method in a .NET controller and an SQL READ query:	
			Details to be read	Attributes in the table
			Company Name	CompanyName
			Email Address	EmailAddress
			Phone Number	PhoneNumber

			and displays them on the Supplier Screen with the following controls:		
			Control Name	Control Type	Notes
			Supplier's List	Heading	Page main heading
			Help icon	Button	When clicked, the admin help document will open
			Create new Supplier	Button	When clicked, this control will direct the administrator to the Create Supplier screen
			Show "" entries	Label and drop down	This control will be used to set the number of entries the administrator wants to display on a page
			Search bar	Textbox	This control will be used to enter the search query
			Results table	Table	This control will be used to display all the suppliers that exist on the system
			Company Name	Table Heading	This control will be used as a heading to display the Supplier's Company Name
			Email Address	Table Heading	This control will be used as a heading to display the Supplier's Email Address
			Phone Number	Table Heading	This control will be used as a heading

					to display the Supplier's Phone Number
	Action	Table Heading			This control will be used as a heading to display the actions the administrator can perform with each entry – Edit and Delete
	Action - Update	Button			This control will be placed in a row, with the delete button, under the "Action" heading for each entry in the Supplier table. The button will direct the user to the Update Supplier page
	Action - Delete	Button			This control will be placed in a row, with the delete button, under the "Action" heading for each entry in the Supplier table. The button will direct the user to the Delete Supplier Details page
	Showing " of entries	Label			This control will be used to display the number of entries displayed per page as well as the

					total number of entries in the database																
				"Previous" and "Next"	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on															
	Step 3: The administrator clicks on the Create New Supplier button		Step 4: The system retrieves the following attributes from the following tables, using a method in a .NET controller and an SQL READ query: From the Country Table	<table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Country</td><td>CountryName</td></tr></table>			Details to be read	Attributes in the table	Country	CountryName											
				Details to be read	Attributes in the table																
				Country	CountryName																
				to populate the drop down lists and displays the Create Supplier Screen which contains an MVC form with the following controls:																	
				<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Create Supplier</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Company Name</td><td>Label</td><td>This control will be used as a label to indicate the supplier's company name attribute</td></tr><tr><td>Company Name</td><td>Textbox</td><td>This control will be used by the administrator to enter the supplier's company name</td></tr><tr><td>Email Address</td><td>Label</td><td>This control will be used as a label to indicate the supplier's email</td></tr></table>			Control Name	Control Type	Notes	Create Supplier	Heading	Page main heading	Company Name	Label	This control will be used as a label to indicate the supplier's company name attribute	Company Name	Textbox	This control will be used by the administrator to enter the supplier's company name	Email Address	Label	This control will be used as a label to indicate the supplier's email
				Control Name	Control Type	Notes															
				Create Supplier	Heading	Page main heading															
				Company Name	Label	This control will be used as a label to indicate the supplier's company name attribute															
Company Name	Textbox	This control will be used by the administrator to enter the supplier's company name																			
Email Address	Label	This control will be used as a label to indicate the supplier's email																			

					address attribute
			Email Address	Textbox	This control will be used by the administrator to enter the supplier's email address
			Phone Number	Label	This control will be used as a label to indicate the supplier's phone number attribute
			Phone Number	Textbox	This control will be used by the administrator to enter the supplier's phone number
			Street Address	Label	This control will be used as a label to indicate the supplier's Street Address attribute
			Street Address	Textbox	This control will be used by the administrator to enter the supplier's Street Address
			Country	Label	This control will be used as a label to indicate the supplier's country attribute
			Country	Dropdown list	This control will be used by the administrator to select the supplier's country
			Province	Label	This control will be used as a label to

					indicate the supplier's province attribute
	Province	Dropdown list			This control will be used by the administrator to select the supplier's province
	City	Label			This control will be used as a label to indicate the supplier's city attribute
	City	Dropdown list			This control will be used by the administrator to select the supplier's city
	Suburb	Label			This control will be used as a label to indicate the supplier's suburb attribute
	Suburb	Dropdown list			This control will be used by the administrator to select the supplier's suburb
	Create	Button			This control will be used to confirm the creation of the supplier
	Cancel	Button			This control will be used to cancel the creation of the Supplier and return the administrator to the Supplier Screen
	Note: All controls are enabled by default, unless otherwise stated. All text input fields				

			are empty by default, unless stated otherwise.																		
	Step 5: The system administrator enters the details and clicks on the create button [ALT]		Step 6: The system captures and validates the entered details against the following tables: From the Supplier Table <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Company Name</td><td>CompanyName</td><td>Required Alphabetical characters only Minimum: 1 Maximum: 50</td></tr><tr><td>Email Address</td><td>EmailAddress</td><td>Required Unique Valid email domain</td></tr><tr><td>Phone Number</td><td>PhoneNumber</td><td>Required Digits only Minimum: 10 Maximum: 10</td></tr><tr><td>Street Address</td><td>StreetAddress</td><td>Required Alphanumeric characters only Minimum: 1 Maximum: 50</td></tr><tr><td>Suburb</td><td>Suburb_Id</td><td>Required FK from Suburb table</td></tr></table> [ALT]	Details to be read	Attributes in the table	Notes	Company Name	CompanyName	Required Alphabetical characters only Minimum: 1 Maximum: 50	Email Address	EmailAddress	Required Unique Valid email domain	Phone Number	PhoneNumber	Required Digits only Minimum: 10 Maximum: 10	Street Address	StreetAddress	Required Alphanumeric characters only Minimum: 1 Maximum: 50	Suburb	Suburb_Id	Required FK from Suburb table
	Details to be read	Attributes in the table	Notes																		
Company Name	CompanyName	Required Alphabetical characters only Minimum: 1 Maximum: 50																			
Email Address	EmailAddress	Required Unique Valid email domain																			
Phone Number	PhoneNumber	Required Digits only Minimum: 10 Maximum: 10																			
Street Address	StreetAddress	Required Alphanumeric characters only Minimum: 1 Maximum: 50																			
Suburb	Suburb_Id	Required FK from Suburb table																			

			Step 7: The system uses an SQL READ query to check if the supplier already exists in the Supplier table using the following attribute. [ALT] <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Email Address</td><td>EmailAddress</td></tr><tr><td>Company Name</td><td>CompanyName</td></tr></table>	Details to be read	Attributes in the table	Email Address	EmailAddress	Company Name	CompanyName															
	Details to be read	Attributes in the table																						
	Email Address	EmailAddress																						
	Company Name	CompanyName																						
			Step 8: The system saves the details in the Supplier table using a method in a .NET controller and an SQL INSERT query: <table><tr><th>Details to be created</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>ID</td><td>ID</td><td>A unique value generated by the system and saved to the database.</td></tr><tr><td>Company Name</td><td>CompanyN ame</td><td>Required Alphabetical characters only Minimum: 1 Maximum: 50</td></tr><tr><td>Email Address</td><td>EmailAddr ess</td><td>Required Unique Valid email domain</td></tr><tr><td>Phone Number</td><td>PhoneNum ber</td><td>Required Digits only Minimum: 10 Maximum: 10</td></tr><tr><td>Street Address</td><td>StreetAddr ess</td><td>Required Alphanumeri c characters only Minimum: 1 Maximum: 50</td></tr><tr><td>Suburb ID</td><td>Suburb_Id</td><td>Required FK from Suburb table</td></tr></table>	Details to be created	Attributes in the table	Notes	ID	ID	A unique value generated by the system and saved to the database.	Company Name	CompanyN ame	Required Alphabetical characters only Minimum: 1 Maximum: 50	Email Address	EmailAddr ess	Required Unique Valid email domain	Phone Number	PhoneNum ber	Required Digits only Minimum: 10 Maximum: 10	Street Address	StreetAddr ess	Required Alphanumeri c characters only Minimum: 1 Maximum: 50	Suburb ID	Suburb_Id	Required FK from Suburb table
	Details to be created	Attributes in the table	Notes																					
	ID	ID	A unique value generated by the system and saved to the database.																					
	Company Name	CompanyN ame	Required Alphabetical characters only Minimum: 1 Maximum: 50																					
	Email Address	EmailAddr ess	Required Unique Valid email domain																					
	Phone Number	PhoneNum ber	Required Digits only Minimum: 10 Maximum: 10																					
Street Address	StreetAddr ess	Required Alphanumeri c characters only Minimum: 1 Maximum: 50																						
Suburb ID	Suburb_Id	Required FK from Suburb table																						
		Step 9: The system directs the administrator to the Supplier Screen and displays a bootstrap alert notification with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr></table>	Control Name	Control Name	Notes																			
Control Name	Control Name	Notes																						

			"Supplier created successfully"	Label	This control is used to indicate a successful creation
ALTERNATE COURSES:	ALT Step 5: The administrator does not want to add a Supplier anymore - The administrator clicks on the cancel button - The use case is terminated				
	ALT Step 6: The details entered are not in the correct format or the required fields are empty The system displays a validation error with the following controls				
	Control Name	Control Name	Notes		
	Company Name	Textbox	This control will be outline in red to show a validation error		
	The Company Name field is required	Label	This control will be used as a label to indicate a validation error in the Company Name textbox		
	Email Address	Textbox	This control will be outline in red to show a validation error		
	The Email Address field is required or "Invalid Email Address provided"	Label	This control will be used as a label to indicate a validation error in the Email Address textbox		
	Phone Number	Textbox	This control will be outline in red to show a validation error		
	The Phone Number field is required or "Invalid Phone Number provided"	Label	This control will be used as a label to indicate a validation error in the Phone Number textbox		
	Street Address	Textbox	This control will be outline in red to show a validation error		
	The Street Address field is required or "Invalid Street Address provided"	Label	This control will be used as a label to indicate a validation error in the Street Address textbox		
	Country	Textbox	This control will be outline in red to show a validation error		
	The Country field is required	Label	This control will be used as a label to indicate a validation error in the Country dropdown		
	Province	Textbox	This control will be outline in red to show a validation error		
	The Province field is required	Label	This control will be used as a label to indicate a validation error in the Province dropdown		
	City	Textbox	This control will be outline in red to show a validation error		

	The City field is required	Label	This control will be used as a label to indicate a validation error in the City dropdown
	Suburb	Textbox	This control will be outline in red to show a validation error
	The Suburb field is required	Label	This control will be used as a label to indicate a validation error in the Suburb dropdown
	Return to step 5		
	ALT Step 7: The Supplier already exists on the system		
	The system displays an error modal dialogue with the following controls		
	Control Name	Control Name	Notes
	Supplier already exists!	Label	This control is used to inform the administrator that the supplier already exists
	Okay	Button	This control is used to cancel the creation of the supplier
	- The administrator clicks the “Okay” button - The use case is terminated		
CONCLUSION:	This use case concludes when a new supplier is created.		
POST-CONDITION:	The supplier details are added to the database.		
BUSINESS RULES	Only the administrator can add a new supplier.		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

Table 121 - 11.2 Search Supplier

SUPPLIER SUBSYSTEM				
AUTHORS (s): Nomusa Vumisa			DATE: 14/05/2022	
VERSION: 1.1			LAST REVIEW DATE: 24/05/2022	
USE CASE NAME:	Search Supplier		USE CASE TYPE	
USE CASE ID:	11.2		Business Requirements: ☐	
PRIORITY:	Medium		System Analysis: ☐	
SOURCE:	ByteXcom Requirements List		System Design: ☒	
PRIMARY BUSINESS ACTOR	Administrator			
PRIMARY SYSTEM ACTOR	None			
OTHER PARTICIPATING ACTORS:	None			
OTHER INTERESTED STAKEHOLDERS:	None			
DESCRIPTION:	This use case describes the process where the administrator searches for a supplier on the system. The system responds by displaying the information relating to the search query.			
PRE-CONDITION:	The administrator must be logged onto the system.			
TRIGGER:	This use case is called when the administrator wants to search for a supplier on the system.			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE		
		MANUAL ACTION	AUTOMATED ACTION	
	Step 1: The administrator clicks on the Supplier navigation link on the admin side navigation bar		Step 2: The system retrieves the following attributes for each entry from the Supplier Table, using a method in a .NET controller and an SQL READ query:	
			Details to be read	Attributes in the table
			Company Name	CompanyName
			Email Address	EmailAddress
			Phone Number	PhoneNumber
			Street Address	StreetAddress
			and displays them on the Supplier Screen with the following controls:	
			Control Name	Control Type
Supplier's List	Heading	Page main heading		

			Create new Supplier	Button	When clicked, this control will direct the administrator to the Create Supplier screen
			Show "" entries	Label and drop down	This control will be used to set the number of entries the administrator wants to display on a page
			Search bar	Textbox	This control will be used to enter the search query
			Results table	Table	This control will be used to display all the suppliers that exist on the system
			Company Name	Table Heading	This control will be used as a heading to display the Supplier's Company Name
			Email Address	Table Heading	This control will be used as a heading to display the Supplier's Email Address
			Phone Number	Table Heading	This control will be used as a heading to display the Supplier's Phone Number
			Action	Table Heading	This control will be used as a heading to display the actions the administrator can perform with each entry – Edit and Delete
			Action - Update	Button	This control will be placed in a row, with the delete button, under the "Action" heading for each entry in the Supplier table. The

					button will direct the user to the Update Supplier page
			Action - Delete	Button	This control will be placed in a row, with the delete button, under the “Action” heading for each entry in the Supplier table. The button will direct the user to the Delete Supplier Details page
			Showing “” to “” of “” entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database
			“Previous” and “Next”	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on
	Step 4: The administrator enters their search query into the search textbox		Step 5: The system searches the table rows in the data table and displays the Supplier(s) matching the search query [ALT]		
	ALT Step 5: The search query entered does not match any records existing in the Supplier entity. • The system displays a “No Records Found” message in the results table. • Return to step 4 or terminate the use case				
CONCLUSION:	This use case concludes when the desired supplier’s details are displayed on the system.				
POST-CONDITION:	The Supplier entity was searched for and an entry that corresponds to the search query or parameter entered.				
BUSINESS RULES	• None				
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• None				
ASSUMPTIONS:	• None				

OPEN ISSUES:

- None

Table 122 - 11.3 Update Supplier

SUPPLIER SUBSYSTEM													
AUTHORS (s): Nomusa Vumisa		DATE: 14/05/2022											
VERSION: 1.1		LAST REVIEW DATE: 24/05/2022											
USE CASE NAME:	Update Supplier		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒										
USE CASE ID:	11.3												
PRIORITY:	Medium												
SOURCE:	ByteXcom Requirements List												
PRIMARY BUSINESS ACTOR	• Administrator												
PRIMARY SYSTEM ACTOR	• None												
OTHER PARTICIPATING ACTORS:	• None												
OTHER INTERESTED STAKEHOLDERS:	• None												
DESCRIPTION:	This use case describes the process in which the administrator wishes to update a supplier on the system. The use case begins when the administrator wishes to update the details of a supplier. The administrator will then update the supplier details and save it to the database. The use case ends when the administrator receives a successful update notification.												
PRE-CONDITION:	The administrator must be logged in to the system.												
TRIGGER:	The use is triggered when the administrator wishes to update a supplier												
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE											
		MANUAL ACTION	AUTOMATED ACTION										
	Step 1: The administrator wishes to update a Supplier in the system.		Step 2: The system invokes use case 11.12 Search Supplier										
	Step 3: The administrator clicks on the update button		Step 3: The system retrieves the following attributes from the following tables, using a method in a .NET controller and an SQL READ query: From the Supplier Table: <table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Company Name</td><td>CompanyName</td></tr><tr><td>Email Address</td><td>EmailAddress</td></tr><tr><td>Phone Number</td><td>PhoneNumber</td></tr><tr><td>Street Address</td><td>StreetAddress</td></tr></table>	Details to be read	Attributes in the table	Company Name	CompanyName	Email Address	EmailAddress	Phone Number	PhoneNumber	Street Address	StreetAddress
	Details to be read	Attributes in the table											
	Company Name	CompanyName											
Email Address	EmailAddress												
Phone Number	PhoneNumber												
Street Address	StreetAddress												

			From the Country Table															
			<table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Country</td><td>CountryName</td></tr></table>	Details to be read	Attributes in the table	Country	CountryName											
			Details to be read	Attributes in the table														
			Country	CountryName														
			From the Province Table															
			<table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Province</td><td>ProvinceName</td></tr></table>	Details to be read	Attributes in the table	Province	ProvinceName											
			Details to be read	Attributes in the table														
			Province	ProvinceName														
			From the City Table															
			<table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>City</td><td>CityName</td></tr></table>	Details to be read	Attributes in the table	City	CityName											
Details to be read	Attributes in the table																	
City	CityName																	
From the Suburb Table																		
<table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Suburb</td><td>SuburbName</td></tr></table>	Details to be read	Attributes in the table	Suburb	SuburbName														
Details to be read	Attributes in the table																	
Suburb	SuburbName																	
and displays them on the Update Supplier Screen which contains an MVC form with the following controls:																		
<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Create Supplier</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Company Name</td><td>Label</td><td>This control will be used as a label to indicate the supplier's company name attribute</td></tr><tr><td>Company Name</td><td>Textbox</td><td>This control will be used by the administrator to enter the supplier's company name</td></tr><tr><td>Email Address</td><td>Label</td><td>This control will be used as a label to indicate the supplier's email address attribute</td></tr><tr><td>Email Address</td><td>Textbox</td><td>This control will be used by the administrator to enter the</td></tr></table>	Control Name	Control Type	Notes	Create Supplier	Heading	Page main heading	Company Name	Label	This control will be used as a label to indicate the supplier's company name attribute	Company Name	Textbox	This control will be used by the administrator to enter the supplier's company name	Email Address	Label	This control will be used as a label to indicate the supplier's email address attribute	Email Address	Textbox	This control will be used by the administrator to enter the
Control Name	Control Type	Notes																
Create Supplier	Heading	Page main heading																
Company Name	Label	This control will be used as a label to indicate the supplier's company name attribute																
Company Name	Textbox	This control will be used by the administrator to enter the supplier's company name																
Email Address	Label	This control will be used as a label to indicate the supplier's email address attribute																
Email Address	Textbox	This control will be used by the administrator to enter the																

					supplier's email address
			Phone Number	Label	This control will be used as a label to indicate the supplier's phone number attribute
			Phone Number	Textbox	This control will be used by the administrator to enter the supplier's phone number
			Street Address	Label	This control will be used as a label to indicate the supplier's Street Address attribute
			Street Address	Textbox	This control will be used by the administrator to enter the supplier's Street Address
			Country	Label	This control will be used as a label to indicate the supplier's country attribute
			Country	Dropdown list	This control will be used by the administrator to select the supplier's country
			Province	Label	This control will be used as a label to indicate the supplier's province attribute
			Province	Dropdown list	This control will be used by the administrator to select the supplier's province
			City	Label	This control will be used as a label to indicate the supplier's city attribute

			City	Dropdown list	This control will be used by the administrator to select the supplier's city
			Suburb	Label	This control will be used as a label to indicate the supplier's suburb attribute
			Suburb	Dropdown list	This control will be used by the administrator to select the supplier's suburb
			Update	Button	This control will be used to confirm the updating of the supplier
			Cancel	Button	This control will be used to cancel the updating of the supplier and return the administrator to the Supplier Screen
			Note: All controls are enabled by default, unless stated otherwise.		
	Step 4: The system administrator enters the updated supplier details and clicks on the update button [ALT]		Step 5: The system captures and validates the entered details against the following tables: [ALT]		
			From the Supplier Table		
			Details to be saved	Attributes in the table	Notes
			Company Name	CompanyName	Required Alphabetical characters only Minimum: 1 Maximum: 50
			Email Address	EmailAddresses	Required Unique Valid email domain
			Phone Number	PhoneNumber	Required Digits only Minimum: 10 Maximum: 10
				Street Address	StreetAddress

					Minimum: 1 Maximum: 50																	
			Suburb	Suburb_Id	Required FK from Suburb table																	
			Step 6: The system displays a modal dialogue with the following controls:																			
			<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>"Confirm Supplier Details"</td><td>Label</td><td>This control is used to request confirmation on the updated supplier details</td></tr><tr><td>Confirm</td><td>Button</td><td>This control is used to confirm the updating of the supplier details</td></tr><tr><td>Cancel</td><td>Button</td><td>This control is used to close the modal dialogue and return the administrator to the Update Supplier Screen</td></tr></table>	Control Name	Control Type	Notes	"Confirm Supplier Details"	Label	This control is used to request confirmation on the updated supplier details	Confirm	Button	This control is used to confirm the updating of the supplier details	Cancel	Button	This control is used to close the modal dialogue and return the administrator to the Update Supplier Screen							
	Control Name	Control Type	Notes																			
	"Confirm Supplier Details"	Label	This control is used to request confirmation on the updated supplier details																			
	Confirm	Button	This control is used to confirm the updating of the supplier details																			
	Cancel	Button	This control is used to close the modal dialogue and return the administrator to the Update Supplier Screen																			
			Step 7: The system administrator clicks on the confirm button [ALT]																			
			Step 8: The system uses an SQL READ query to check if the supplier already exists in the Supplier table using the following attribute. [ALT]																			
		<table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Company Name</td><td>CompanyName</td></tr><tr><td>Email Address</td><td>EmailAddress</td></tr><tr><td>Phone Number</td><td>PhoneNumber</td></tr><tr><td>Street Address</td><td>StreetAddress</td></tr><tr><td>Country</td><td>CountryID</td></tr><tr><td>Province</td><td>ProvinceID</td></tr><tr><td>City</td><td>CityID</td></tr><tr><td>Suburb</td><td>SuburbID</td></tr></table>	Details to be read	Attributes in the table	Company Name	CompanyName	Email Address	EmailAddress	Phone Number	PhoneNumber	Street Address	StreetAddress	Country	CountryID	Province	ProvinceID	City	CityID	Suburb	SuburbID		
Details to be read	Attributes in the table																					
Company Name	CompanyName																					
Email Address	EmailAddress																					
Phone Number	PhoneNumber																					
Street Address	StreetAddress																					
Country	CountryID																					
Province	ProvinceID																					
City	CityID																					
Suburb	SuburbID																					
		Step 9: The system saves the updated details into the Supplier table using a method in a .NET controller and an SQL UPDATE query:																				
		<table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr></table>	Details to be saved	Attributes in the table	Notes																	
Details to be saved	Attributes in the table	Notes																				

			Company Name	CompanyNa me	Required Alphabetical characters only Minimum: 1 Maximum: 50
			Email Address	EmailAddres s	Required Unique Valid email domain
			Phone Number	PhoneNumb er	Required Digits only Minimum: 10 Maximum: 10
			Street Address	StreetAddre ss	Required Alphanumeric characters only Minimum: 1 Maximum: 50
			Suburb ID	Suburb_ID	Required FK from Suburb table
			Step 10: The system directs the administrator to the Supplier Screen and displays a bootstrap alert notification with the following controls:		
			Control Name	Control Name	Notes
			"Supplier updated successfully"	Label	This control is used to indicate a successful creation
	ALTERNATE COURSES:	ALT Step 4: The administrator does not want to update the Supplier anymore - The administrator clicks on the cancel button - The use case is terminated			
		ALT Step 5: The details entered are not in the correct format or the required fields are empty The system displays a validation error with the following controls			
		Control Name	Control Name	Notes	
		Company Name	Textbox	This control will be outline in red to show a validation error	
		The Company Name field is required or "Invalid Company Name provided"	Label	This control will be used as a label to indicate a validation error in the Company Name textbox	
		Email Address	Textbox	This control will be outline in red to show a validation error	
		The Email Address field is required or "Invalid Email Address provided"	Label	This control will be used as a label to indicate a validation error in the Email Address textbox	
		Phone Number	Textbox	This control will be outline in red to show a validation error	
		The Phone Number field is required or "Invalid Phone Number provided"	Label	This control will be used as a label to indicate a validation error in the Phone Number textbox	
		Street Address	Textbox	This control will be outline in red to show a validation error	

	The Street Address field is required or "Invalid Street Address provided"	Label	This control will be used as a label to indicate a validation error in the Street Address textbox
	Country	Dropdown list	This control will be outline in red to show a validation error
	The Country field is required	Label	This control will be used as a label to indicate a validation error in the Country dropdown
	Province	Dropdown list	This control will be outline in red to show a validation error
	The Province field is required	Label	This control will be used as a label to indicate a validation error in the Province dropdown
	City	Dropdown list	This control will be outline in red to show a validation error
	The City field is required	Label	This control will be used as a label to indicate a validation error in the City dropdown
	Suburb	Dropdown list	This control will be outline in red to show a validation error
	The Suburb field is required	Label	This control will be used as a label to indicate a validation error in the Suburb dropdown
	Return to step 3		
ALT Step 7: The administrator clicks on the cancel button Return to Step 4 or terminate the use case			
ALT Step 8: The Supplier already exists on the system The system displays an error modal dialogue with the following controls			
Control Name		Control Name	Notes
Supplier already exists!		Label	This control is used to inform the administrator that the supplier already exists
Okay		Button	This control is used to cancel the updating of the supplier
- The administrator clicks the "Okay" button - The use case is terminated			
CONCLUSION:	This use case concludes when the supplier details are updated in the Supplier entity and the system displays an alert notification that informs the administrator that the supplier's details has been successfully updated.		
POST-CONDITION:	The supplier details are updated in the Supplier table.		
BUSINESS RULES	Only the administrator can update a supplier.		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	The system is offline and the administrator is not able to access the system to update a supplier's details.		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

Table 123 - 11.4 Delete Supplier

SUPPLIER SUBSYSTEM			
AUTHORS (s): Nomusa Vumisa		DATE: 14/05/2022	
VERSION: 1.0		LAST REVIEW DATE: 14/05/2022	
USE CASE NAME:	Delete Supplier		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒
USE CASE ID:	11.4		
PRIORITY:	Medium		
SOURCE:	ByteXcom Requirements List		
PRIMARY BUSINESS ACTOR	• Administrator		
PRIMARY SYSTEM ACTOR	• None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	This use case narrative describes the process in which the administrator wishes to delete a supplier from the system. The administrator will search for the required supplier and then confirm the deletion of the supplier and the use case concludes when the system notifies the administrator of a successful deletion.		
PRE-CONDITION:	The administrator must be logged in to the system.		
TRIGGER:	This use case is called when the administrator wishes to delete a supplier.		
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE	
		MANUAL ACTION	AUTOMATED ACTION
	Step 1: The administrator wishes to delete a supplier defined on the system.		Step 2: The system invokes use case 11.12 Search Supplier
	Step 3: The administrator clicks on the Delete button		Step 4: The system retrieves the following attributes from the following tables, using a method in a .NET controller and an SQL READ query: From the Supplier Table:
		Details to be read	Attributes in the table
	Company Name	CompanyName	

			<table><tr><td>Email Address</td><td>EmailAddress</td></tr><tr><td>Phone Number</td><td>PhoneNumber</td></tr><tr><td>Street Address</td><td>StreetAddress</td></tr></table>	Email Address	EmailAddress	Phone Number	PhoneNumber	Street Address	StreetAddress												
Email Address	EmailAddress																				
Phone Number	PhoneNumber																				
Street Address	StreetAddress																				
			From the Country Table																		
			<table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Country</td><td>CountryName</td></tr></table>	Details to be read	Attributes in the table	Country	CountryName														
Details to be read	Attributes in the table																				
Country	CountryName																				
			From the Province Table																		
			<table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Province</td><td>ProvinceName</td></tr></table>	Details to be read	Attributes in the table	Province	ProvinceName														
Details to be read	Attributes in the table																				
Province	ProvinceName																				
			From the City Table																		
			<table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>City</td><td>CityName</td></tr></table>	Details to be read	Attributes in the table	City	CityName														
Details to be read	Attributes in the table																				
City	CityName																				
			From the Suburb Table																		
			<table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Suburb</td><td>SuburbName</td></tr></table>	Details to be read	Attributes in the table	Suburb	SuburbName														
Details to be read	Attributes in the table																				
Suburb	SuburbName																				
			and displays them on the Delete Supplier Details Screen with the following controls:																		
			<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Delete Supplier Details</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Company Name</td><td>Label</td><td>This control will be used to display the supplier's company name</td></tr><tr><td>Email Address</td><td>Label</td><td>This control will be used to display the supplier's email address</td></tr><tr><td>Phone Number</td><td>Label</td><td>This control will be used to display the supplier's phone number</td></tr><tr><td>Street Address</td><td>Label</td><td>This control will be used to</td></tr></table>	Control Name	Control Type	Notes	Delete Supplier Details	Heading	Page main heading	Company Name	Label	This control will be used to display the supplier's company name	Email Address	Label	This control will be used to display the supplier's email address	Phone Number	Label	This control will be used to display the supplier's phone number	Street Address	Label	This control will be used to
Control Name	Control Type	Notes																			
Delete Supplier Details	Heading	Page main heading																			
Company Name	Label	This control will be used to display the supplier's company name																			
Email Address	Label	This control will be used to display the supplier's email address																			
Phone Number	Label	This control will be used to display the supplier's phone number																			
Street Address	Label	This control will be used to																			

					display the supplier's Street Address
	Country	Label			This control will be used to display the supplier's country
	Province	Label			This control will be used to display the supplier's province
	City	Label			This control will be used to display the supplier's city
	Suburb	Label			This control will be used to display the supplier's suburb
	Delete	Button			This control will be used to confirm the deletion of the supplier
	Cancel	Button			This control will be used to cancel the deletion of the supplier and return the administrator to the Supplier Screen
	Step 5: The administrator clicks on the delete button [ALT]				
	Step 6: The system displays a modal dialogue with the following controls:				
				</	

	on the confirm button [ALT]		has an association with InventoryProcured Table. [ALT]																				
			<table><tr><th>Details to be read</th><th colspan="2">Attributes in the table</th></tr><tr><td>Supplier ID</td><td>SupplierID</td><td>Notes</td></tr></table>	Details to be read	Attributes in the table		Supplier ID	SupplierID	Notes														
	Details to be read	Attributes in the table																					
	Supplier ID	SupplierID	Notes																				
			Step 9: The system deletes the selected Supplier from the Supplier entity in the database using a method in a .NET controller and an SQL DELETE query.																				
		<table><tr><th>Details to be deleted</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>ID</td><td>ID</td><td>A unique value generated by the system and saved to the database.</td></tr><tr><td>Company Name</td><td>CompanyNa me</td><td>Required Alphabetical characters only Minimum: 1 Maximum: 50</td></tr><tr><td>Email Address</td><td>EmailAddres s</td><td>Required Unique Valid email domain</td></tr><tr><td>Phone Number</td><td>PhoneNumb er</td><td>Required Digits only Minimum: 10 Maximum: 10</td></tr><tr><td>Street Address</td><td>StreetAddre ss</td><td>Required Alphanumeric characters only Minimum: 1 Maximum: 50</td></tr><tr><td>Suburb ID</td><td>Suburb_ID</td><td>Required FK from Suburb table</td></tr></table>	Details to be deleted	Attributes in the table	Notes	ID	ID	A unique value generated by the system and saved to the database.	Company Name	CompanyNa me	Required Alphabetical characters only Minimum: 1 Maximum: 50	Email Address	EmailAddres s	Required Unique Valid email domain	Phone Number	PhoneNumb er	Required Digits only Minimum: 10 Maximum: 10	Street Address	StreetAddre ss	Required Alphanumeric characters only Minimum: 1 Maximum: 50	Suburb ID	Suburb_ID	Required FK from Suburb table
Details to be deleted	Attributes in the table	Notes																					
ID	ID	A unique value generated by the system and saved to the database.																					
Company Name	CompanyNa me	Required Alphabetical characters only Minimum: 1 Maximum: 50																					
Email Address	EmailAddres s	Required Unique Valid email domain																					
Phone Number	PhoneNumb er	Required Digits only Minimum: 10 Maximum: 10																					
Street Address	StreetAddre ss	Required Alphanumeric characters only Minimum: 1 Maximum: 50																					
Suburb ID	Suburb_ID	Required FK from Suburb table																					
		Step 10: The system redirects the administrator to the Supplier Screen and displays a modal dialogue with the following controls:																					
		<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"Supplier Deleted Successfully"</td><td>Label</td><td>This control is used to indicate a successful deletion</td></tr></table>	Control Name	Control Name	Notes	"Supplier Deleted Successfully"	Label	This control is used to indicate a successful deletion															
Control Name	Control Name	Notes																					
"Supplier Deleted Successfully"	Label	This control is used to indicate a successful deletion																					
ALTERNATE COURSES:	ALT Step 5: The administrator does not want to delete the Supplier anymore • The administrator clicks on the cancel button • The use case is terminated																						

	ALT Step 7: The administrator does not confirm the deletion of the Supplier - The administrator clicks on the cancel button - Return to step 4						
	ALT Step 8: The supplier cannot be deleted as it has an association with the InventoryProcured Table The system redirects the administrator and displays a bootstrap alert notification with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"Supplier cannot be deleted"</td><td>Label</td><td>This control is used to indicate an unsuccessful deletion</td></tr></table>	Control Name	Control Name	Notes	"Supplier cannot be deleted"	Label	This control is used to indicate an unsuccessful deletion
	Control Name	Control Name	Notes				
	"Supplier cannot be deleted"	Label	This control is used to indicate an unsuccessful deletion				
The use case is terminated							
CONCLUSION:	This use case concludes when the administrator is notified of the successful deletion of the supplier.						
POST-CONDITION:	The system deletes all the details related to the selected supplier from the Supplier table .						
BUSINESS RULES	- Only the administrator can delete a supplier. - A supplier from which inventory has been received or ordered cannot be deleted on the system.						
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None						
ASSUMPTIONS:	None						
OPEN ISSUES:	None						

Table 124 - 11.5 Send Supplier Order

SUPPLIER SUBSYSTEM			
AUTHORS (s): Kyle van Eeden		DATE: 27/06/2022	
VERSION: 1.0		LAST REVIEW DATE: 27/06/2022	
USE CASE NAME:	Send Supplier Order		USE CASE TYPE
USE CASE ID:	11.5		Business Requirements: ☐
PRIORITY:	High		System Analysis: ☐
SOURCE:	ByteXcom Requirements List		System Design: ☒
PRIMARY BUSINESS ACTOR	Administrator		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This use case describes the event in which an administrator requests the system to send a request for inventory to a specific supplier via email. The administrator will be prompted to select a specific supplier defined on the system which to send the email to. Once a supplier is selected, the system displays all the retrieved details for the supplier and retrieves the names of all inventory items from the SupplierInventoryItem entity where the selected supplier has been associated. The administrator is able to search for a specific inventory item to be added to the supplier order. The search query provided will retrieve all matched records from the InventoryItem entity. After the inventory items have been added to the list of inventories, the quantity of each inventory item needed is specified by the system administrator. Upon requesting to send the supplier order email, the system will validate all the details captured and display exception messages for all controls containing invalid input. Once validations passed, the system will generate an email using SendGrid and send the supplier order email to the selected supplier. This use case concludes once the system displays a notification confirming that the email has been sent successfully.		
PRE-CONDITION:	- The administrator must be logged into the system. - At least one Inventory Item has to be defined on the system which can be included in the supplier order email. - The supplier to be requested for inventories must already be defined on the system.		
TRIGGER:	The use case is triggered when a system administrator requests to send a supplier order email using the system.		
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE	
		MANUAL ACTION	AUTOMATED ACTION
	Step 1: The Administrator		Step 2: The system retrieves the following details for all suppliers in the

requests to send a supplier order by selecting the '**Send Supplier Order**' navigation link provided in the side-navigation menu.

Supplier entity using a .NET controller and SQL Read Query:

Details to be read	Attributes in the table	Notes
Supplier Name	CompanyNa me	Name of each supplier retrieved
Supplier Email	SupplierEm ail	Email address of each supplier.

The system displays the **Suppliers List** screen containing a ASP.NET form with the following controls:

Control Name	Control Type	Control Notes
Supplier List	Header	Main heading of the page.
Search Supplier	Search Box	Used to filter the results displayed in the Suppliers table. Allows user to search by a supplier's name.
View All	Button	Used to retrieve all suppliers to populate the Suppliers table.
Suppliers	Table	Table which contains the details retrieved for all suppliers.
Supplier	Table Heading	Heading of the table field that holds the names of all suppliers retrieved.
Supplier	Table Row	The CompanyNa me value is displayed for each supplier retrieved.
Email	Table Heading	Heading of the table's field that displays the email address for each supplier retrieved.
Email	Table Row	The SupplierEmai l attribute value is

					displayed for each supplier retrieved.											
			Request Inventory	Button	Allows the user to navigate to the Send Supplier Order page, populated with details relevant to the supplier selected.											
			Back	Button	Navigates the user back to the previous page if clicked.											
	Note: All controls on the form are enabled by default, unless specified otherwise in the Control Notes section.															
	Step 3: The administrator enters a search criteria value in the Search Supplier control to search for the specific supplier they wish to send an order request to.		Step 4: The system updates the Suppliers table displayed by only populating it with and displaying suppliers' details which have matched the search criteria entered. The performed search retrieves suppliers where the Company Name attribute value exactly matches the search query.													
	Step 5: The administrator clicks on the Request Inventory button for the specific supplier they wish to contact. [ALT]		Step 6: The system uses a .NET controller with a SQL Read query to retrieve the following details from the Supplier entity for the specific supplier selected: <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Company Name</td><td>CompanyNa me</td><td>Populates the CompanyNam e attribute value in the relevant control.</td></tr><tr><td>Supplier Email Address</td><td>SupplierEm ail</td><td>Populates the SupplierEmail attribute value in the relevant control.</td></tr><tr><td>Supplier Contact Number</td><td>SupplierPho neNumber</td><td>Populates the SupplierPhon eNumber attribute value in the relevant control.</td></tr></table>			Details to be read	Attributes in the table	Notes	Company Name	CompanyNa me	Populates the CompanyNam e attribute value in the relevant control.	Supplier Email Address	SupplierEm ail	Populates the SupplierEmail attribute value in the relevant control.	Supplier Contact Number	SupplierPho neNumber
Details to be read	Attributes in the table	Notes														
Company Name	CompanyNa me	Populates the CompanyNam e attribute value in the relevant control.														
Supplier Email Address	SupplierEm ail	Populates the SupplierEmail attribute value in the relevant control.														
Supplier Contact Number	SupplierPho neNumber	Populates the SupplierPhon eNumber attribute value in the relevant control.														

The system retrieves the following details from the **SupplierInventoryItem** entity using a .NET controller with an EF Core SQL Read query:

Details to be read	Attributes in the table	Notes
Inventory Item	InventoryItemId	Retrieves the Foreign Keys of all Inventory Items that are associated with the selected supplier's SupplierId.

The system retrieves the following details from the **InventoryItem** entity using .NET Controller with an EF Core SQL Read query:

Details to be read	Attributes in the table	Notes
Inventory Item Name	InventoryItemName	Populate the names of all inventory items associated with selected supplier.

The system displays the **Send Supplier Order** screen which contains a ASP.NET form with the following controls:

Control Name	Control Type	Control Notes
Supplier Order Email	Header	Main heading of the page.
Supplier Name	Label	Descriptive label for the Supplier Name Textbox control.
Supplier Name	Textbox	Used to populate the retrieved CompanyName value; Disabled .
Supplier Email	Label	Descriptive label for the Supplier Email Textbox control.

			Supplier Email	Textbox	Used to populate the retrieved SupplierEmail value; Disabled .
			Contact Number	Label	Descriptive label for the Contact Number Textbox control.
			Contact Number	Textbox	Used to populate the retrieved SupplierPhoneNumber value; Disabled .
			Inventories Needed	Label	Descriptive label for the Inventories table.
			Search Inventories	Search List Box	Used to display search results.
			Inventories	Table	Displays all inventory items associated with supplier in the SupplierInventory entity.
			Inventory Item Name	Table Heading	Heading for the field that is used to display the InventoryItemName value for each inventory item added to the Inventories table.
			Order Quantity	Table Heading	Heading for the Order Quantity field displayed.
			Inventories List	Table Row	Used to display the name of each inventory item and capture the quantity of each inventory item to be requested. The quantity value will be initialized to 0 upon form load.
			Send Request	Button	Validates provided form

					input if clicked.
			Back	Button	Navigates the user back to the previous page if clicked.
	Step 7: The administrator enters a search criteria value in the Search Inventories control. [ALT]		Step 8: The system uses a .NET repository method with a EF Core SQL Read query to display the following attribute values of all inventory items retrieved from the InventoryItem entity where the name has matched the search criteria entered:		
			</		

					greater than 1.																					
			[ALT]																							
			Step 13: The system generates an email populated with all the supplier order details captured on the ASP.NET form using SendGrid:																							
			<table><tr><th>Details to be Populated</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>From</td><td></td><td>Set to Natuurlik's bussiness email address.</td></tr><tr><td>To</td><td>SupplierEmail</td><td>Populated with the email address of the selected supplier.</td></tr><tr><td>Subject</td><td></td><td>Populated with the Email's Subject value entered on the form.</td></tr><tr><td>Message</td><td></td><td>Populated with the value entered for the Message field on the form.</td></tr><tr><td>Inventory Item</td><td>InventoryItemName</td><td>Populated with all Inventory Items added to the supplier order request.</td></tr><tr><td>Order Quantity</td><td></td><td>Populated with the quantities entered for each inventory item respectively.</td></tr></table>			Details to be Populated	Attributes in the table	Notes	From		Set to Natuurlik's bussiness email address.	To	SupplierEmail	Populated with the email address of the selected supplier.	Subject		Populated with the Email's Subject value entered on the form.	Message		Populated with the value entered for the Message field on the form.	Inventory Item	InventoryItemName	Populated with all Inventory Items added to the supplier order request.	Order Quantity		Populated with the quantities entered for each inventory item respectively.
Details to be Populated	Attributes in the table	Notes																								
From		Set to Natuurlik's bussiness email address.																								
To	SupplierEmail	Populated with the email address of the selected supplier.																								
Subject		Populated with the Email's Subject value entered on the form.																								
Message		Populated with the value entered for the Message field on the form.																								
Inventory Item	InventoryItemName	Populated with all Inventory Items added to the supplier order request.																								
Order Quantity		Populated with the quantities entered for each inventory item respectively.																								
			Step 14: The system uses SendGrid to send the generated supplier order email to the supplier's email address.																							

			Step 15: The system displays a confirmation message box that contains the following details:									
			<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Supplier Order Request Sent Successfully</td><td>Label</td><td>Contains the success message populated.</td></tr><tr><td>Ok</td><td>Button</td><td>The administrator acknowledges the alert by clicking this button.</td></tr></table>	Control Name	Control Type	Notes	Supplier Order Request Sent Successfully	Label	Contains the success message populated.	Ok	Button	The administrator acknowledges the alert by clicking this button.
	Control Name	Control Type	Notes									
	Supplier Order Request Sent Successfully	Label	Contains the success message populated.									
Ok	Button	The administrator acknowledges the alert by clicking this button.										
ALTERNATE COURSES:	ALT Step 5: The Administrator no longer wishes to send a supplier order email from the system or the supplier's details is not yet created on the system: - The Administrator clicks on the Cancel button - The use case is terminated											
	ALT Step 7 a: The administrator does not wish to search for a specific inventory item as the inventories list already contains all the inventory items needed for the supplier order email: Proceed to Step 11.											
	ALT Step 7 b: The administrator wishes to remove an inventory item existing in the supplier's Inventories table: - The administrator clicks the Remove button for the specific inventory item they wish to remove from the supplier order email. - The system removes the selected inventory item from the Inventories List which is displayed on screen and the system retrieves the updated inventories list to populate the Inventories table control. - Proceed to Step 11.											
	ALT Step 9: The inventory item the administrator wishes to include in the supplier order does not exist on the system: - The administrator clicks the Cancel button. - Return to Step 2.											
	ALT Step 11: The administrator no longer wishes to send the supplier order email and requests to cancel the operation: - The administrator clicks the Cancel button. - Return to Step 2.											
	ALT Step 12 a: The details entered are not in the correct format or the required fields are empty The system displays a validation error message box with the following controls <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Subject</td><td>Textbox</td><td>Outline border with red and display a validation error message stating that the Subject value is required.</td></tr><tr><td>Message</td><td>Textbox</td><td>Outline border with red and display a validation error</td></tr></table>			Control Name	Control Type	Notes	Subject	Textbox	Outline border with red and display a validation error message stating that the Subject value is required.	Message	Textbox	Outline border with red and display a validation error
Control Name	Control Type	Notes										
Subject	Textbox	Outline border with red and display a validation error message stating that the Subject value is required.										
Message	Textbox	Outline border with red and display a validation error										

			message stating that the Message value is required.						
	Return to Step 11.								
	ALT Step 12 b: The details entered is not valid as per the following validation requirements :								
	<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Value of greater than 1 is required.</td><td>Message Box</td><td>This control will be used as a label to indicate a validation error in the Order Quantity numeric-up-down control.</td></tr></table>	Control Name	Control Name	Notes	Value of greater than 1 is required.	Message Box	This control will be used as a label to indicate a validation error in the Order Quantity numeric-up-down control.		
	Control Name	Control Name	Notes						
Value of greater than 1 is required.	Message Box	This control will be used as a label to indicate a validation error in the Order Quantity numeric-up-down control.							
Return to Step 11.									
CONCLUSION:	This use case concludes when the system displays a supplier order sent successfully message on the Send Supplier Order screen.								
POST-CONDITION:	- The list of inventories that was used in the supplier order email updates the supplier’s inventory item list in the SupplierInventory entity. - The generated email is sent to the selected supplier’s email address.								
BUSINESS RULES	- Only administrator users should be allowed to send order requests to suppliers. - Administrators should be allowed to indicate which inventories they wish to request from a specific supplier along with the required quantities. - The administrator should be allowed to add additional information regarding the supplier order email by entering a custom message.								
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	- The system is offline, and the Administrator will not be able to send an email to a supplier from the system. - The list of inventories last used in an order email sent to a specific supplier should be remembered on the system.								
ASSUMPTIONS:	Suppliers will be familiar with the names of inventory items requested from Natuurlik’s system.								
OPEN ISSUES:	None								

2.12. Driver Subsystem

Table 125 - 12.1 View Deliveries

DRIVER SUBSYSTEM			
AUTHORS (s): Ofhani Mungani		DATE: 30/08/2022	
VERSION: 1.0		LAST REVIEW DATE: 30/08/2022	
USE CASE NAME:	View Deliveries		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒
USE CASE ID:	12.1		
PRIORITY:	High		
SOURCE:	ByteXcom Requirements List		
PRIMARY BUSINESS ACTOR	Delivery Driver		
PRIMARY SYSTEM ACTOR	None		
OTHER PARTICIPATING ACTORS:	None		
OTHER INTERESTED STAKEHOLDERS:	None		
DESCRIPTION:	This use case describes an event where the delivery driver wants to view orders to deliver on a mobile device. The use case begins when the delivery driver requests to view orders to deliver and the system responds by retrieving the details. The use case concludes when the details are displayed on the mobile application screen.		
PRE-CONDITION:	- The main system server should be online. - The order status should be dispatched.		
TRIGGER:	The use case is called when the delivery driver wants to view orders to deliver.		
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE	
		MANUAL ACTION	AUTOMATED ACTION

Step 1: The delivery driver opens the delivery mobile application.

Step 2: The mobile application sends an Application Programming Interface (API) request to the system's database and retrieves the following attributes from the following tables, using a method in a .NET controller and an SQL READ query:

From the **Order** table

Details to be read	Attributes in the table
Order Number	OrderId
First Name	FirstName
Surname	Surname
Order Date	CreatedDate
Phone Number	PhoneNumber
Street Address	StreetAddress

From the **Country** Table

Details to be read	Attributes in the table
Country	CountryName

From the **Province** Table

Details to be read	Attributes in the table
Province	ProvinceName

From the **City** Table

Details to be read	Attributes in the table
City	CityName

From the **Suburb** Table

Details to be read	Attributes in the table
Suburb	SuburbName

and displays them on the **Order(s) to Deliver Screen** which contains an Ionic tab view with the following controls:

Control Name	Control Type	Notes
Order(s) to Deliver	Heading	This control will be used to display the orders to deliver
Order Number	Subheading	This control is used to display the orderId
First Name	Label	This control will be used to display the recipient's Name
Surname	Label	This control will be used to display the recipient's Surname
Phone Number	Label	This control will be used to display the recipient's phone number
Order Date	Label	This control will be used to display the order created date
Street Name	Label	This control will be used to display the recipient's street name
Country	Label	This control will be used to display the recipient's country
Province	Label	This control will be used to display the recipient's province
City	Label	This control will be used to display the recipient's city
Suburb	Label	This control will be used to

					display the recipient's suburb
			Complete Task	Label	This control is used to give description to the Checkmark-outline icon
			Checkmark-outline icon	Button	This control will be used to mark completion of order deliveries
			Deliveries	Button	This control will be used to navigate to the Order(s) to Deliver Screen
			Delivered Orders	Button	This control will be used to navigate to the Previous Orders Screen
ALTERNATE COURSES:	None				
CONCLUSION:	This use case concludes when the order's details are displayed.				
POST-CONDITION:	The Order , Suburb , City , Province and Country Tables were read				
BUSINESS RULES	The mobile application only handles deliveries linked to Garsfontein suburb.				
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	- The delivery driver cannot view orders without an internet connection. - The delivery driver should own an android device.				
ASSUMPTIONS:	The delivery driver owns an android device.				
OPEN ISSUES:	None				

Table 126 - 12.2 View Previous Deliveries

DRIVER SUBSYSTEM				
AUTHORS (s): Ofhani Mungani		DATE: 30/08/2022		
VERSION: 1.0		LAST REVIEW DATE: 30/08/2022		
USE CASE NAME:	View Previous Deliveries	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒		
USE CASE ID:	12.2			
PRIORITY:	High			
SOURCE:	ByteXcom Requirements List			
PRIMARY BUSINESS ACTOR	• Delivery Driver			
PRIMARY SYSTEM ACTOR	• None			
OTHER PARTICIPATING ACTORS:	• None			
OTHER INTERESTED STAKEHOLDERS:	• None			
DESCRIPTION:	This use case describes an event where the delivery driver wants to view previous deliveries on a mobile device. The use case begins when the delivery driver requests to view previous deliveries and the system responds by retrieving the details. The use case concludes when the details are displayed on the mobile application screen.			
PRE-CONDITION:	• The main system server should be online.			
TRIGGER:	The use case is called when the delivery driver wants to view previous deliveries.			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE		
		MANUAL ACTION	AUTOMATED ACTION	
	Step 1: The delivery driver wants to view previous deliveries on the mobile application.		Step 2: The system invokes Use Case 12.1 View Delivery	
	Step 3: The delivery driver clicks on the Delivered Orders tab button		Step 4: The mobile application sends an Application Programming Interface (API) request to the system's database and retrieves the following attributes from the following tables, using a method in a .NET controller and an SQL READ query: From the Order table	
		Details to be read	Attributes in the table	

			<table><tr><td>Order Number</td><td>OrderId</td></tr></table>	Order Number	OrderId																
			Order Number	OrderId																	
			From the Delivery Table																		
			<table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Date</td><td>Date</td></tr></table>	Details to be read	Attributes in the table	Date	Date														
			Details to be read	Attributes in the table																	
			Date	Date																	
			and displays them on the Previous Deliveries Screen which contains an Ionic tab view with the following controls:																		
			<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Previous Deliveries</td><td>Heading</td><td>This control will be used to display the delivered orders</td></tr><tr><td>Order Number</td><td>Label</td><td>This control will be used to display the orderId</td></tr><tr><td>Date</td><td>Label</td><td>This control will be used to display the delivery date</td></tr><tr><td>Deliveries</td><td>Button</td><td>This control will be used to navigate to the Order(s) to Deliver Screen</td></tr><tr><td>Delivered Orders</td><td>Button</td><td>This control will be used to navigate to the Previous Orders Screen</td></tr></table>	Control Name	Control Type	Notes	Previous Deliveries	Heading	This control will be used to display the delivered orders	Order Number	Label	This control will be used to display the orderId	Date	Label	This control will be used to display the delivery date	Deliveries	Button	This control will be used to navigate to the Order(s) to Deliver Screen	Delivered Orders	Button	This control will be used to navigate to the Previous Orders Screen
			Control Name	Control Type	Notes																
			Previous Deliveries	Heading	This control will be used to display the delivered orders																
Order Number	Label	This control will be used to display the orderId																			
Date	Label	This control will be used to display the delivery date																			
Deliveries	Button	This control will be used to navigate to the Order(s) to Deliver Screen																			
Delivered Orders	Button	This control will be used to navigate to the Previous Orders Screen																			
ALTERNATE COURSES:	• None																				
CONCLUSION:	This use case concludes when the delivery's details are displayed.																				
POST-CONDITION:	The Delivery Tables were read																				
BUSINESS RULES	• The mobile application only handles deliveries linked to Garsfontein suburb. • This functionality is only to be used by Natuurlik's drivers.																				

IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• The delivery driver cannot view orders without an internet connection. • The delivery driver cannot view orders without an android device
ASSUMPTIONS:	• The delivery driver owns an android device.
OPEN ISSUES:	• None

Table 127 - 12.3 Confirm Delivery

DRIVER SUBSYSTEM					
AUTHORS (s): Ofhani Mungani			DATE: 30/08/2022		
VERSION: 1.0			LAST REVIEW DATE: 30/08/2022		
USE CASE NAME:	Confirm Deliveries		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒		
USE CASE ID:	12.3				
PRIORITY:	High				
SOURCE:	ByteXcom Requirements List				
PRIMARY BUSINESS ACTOR	• Delivery Driver				
PRIMARY SYSTEM ACTOR	• None				
OTHER PARTICIPATING ACTORS:	• None				
OTHER INTERESTED STAKEHOLDERS:	• None				
DESCRIPTION:	This use case describes an event where the delivery driver is done delivering an order and wants to confirm the completion of the task. The use case begins when the delivery driver wants to confirm the completion of a delivery and the system responds by creating a new delivery on the system’s database. The use case concludes when a message indicating that order is delivered is displayed on the mobile application screen.				
PRE-CONDITION:	• The main system server should online.				
TRIGGER:	The use case is called when the delivery driver wants to confirm the completion of an order delivery.				
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE			
		MANUAL ACTION	AUTOMATED ACTION		
	Step 1: The delivery driver wants to confirm the completion of an order delivery.		Step 2: The system invokes Use Case 12.1 View Delivery		
	Step 3: The delivery driver clicks on the checkmark-outline Icon button		Step 4: The mobile application sends an Application Programming Interface (API) request to the system’s database and update the order status in the Order table to “Delivered” using a method in a .NET controller and an SQL UPDATE query:		
				Details to be Updated	Attributes in the Table
			Order Status	OrderStatus	The status of the order is

					changed to “Delivered ”.									
			Step 5: The mobile application sends an Application Programming Interface (API) request to the system’s database and create a new completed delivery using a method in a .NET controller and an SQL INSERT query:											
			Into the Delivery table											
			<table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Id</td><td>Id</td></tr><tr><td>Date</td><td>Date</td></tr><tr><td>Order Number</td><td>OrderId</td></tr></table>			Details to be read	Attributes in the table	Id	Id	Date	Date	Order Number	OrderId	
Details to be read	Attributes in the table													
Id	Id													
Date	Date													
Order Number	OrderId													
			and displays a toaster alert with the following controls:											
			<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>“Delivery was completed successfully ”</td><td>Label</td><td>This control is used to indicate a successful creation</td></tr><tr><td>Close</td><td>Button</td><td>This control will be used to dismiss the toaster alert</td></tr></table>			Control Name	Control Type	Notes	“Delivery was completed successfully ”	Label	This control is used to indicate a successful creation	Close	Button	This control will be used to dismiss the toaster alert
Control Name	Control Type	Notes												
“Delivery was completed successfully ”	Label	This control is used to indicate a successful creation												
Close	Button	This control will be used to dismiss the toaster alert												
	Step 6: The delivery driver clicks on the Close button													
ALTERNATE COURSES:	• None													
CONCLUSION:	This use case concludes when the delivery driver dismisses the toaster alert.													
POST-CONDITION:	A new completed delivery is created in the Delivery table													
BUSINESS RULES	• The mobile application only handles deliveries linked to Garsfontein suburb.													
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• The delivery driver cannot view orders without an internet connection. • The delivery driver should own an android device.													
ASSUMPTIONS:	• The delivery driver owns an android device.													

OPEN ISSUES:

- None

2.13. VAT Subsystem

Table 128 - 13.1 Create VAT

VAT SUBSYSTEM						
AUTHORS (s): Aphiwe Shozi			DATE: 06/06/2022			
VERSION: 1.1			LAST QUERY DATE: 18/06/2022			
USE CASE NAME:	Create VAT Percentage		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒			
USE CASE ID:	13.1					
PRIORITY:	High					
SOURCE:	ByteXcom Requirements List					
PRIMARY BUSINESS ACTOR	• Administrator					
PRIMARY SYSTEM ACTOR	• None					
OTHER PARTICIPATING ACTORS:	• None					
OTHER INTERESTED STAKEHOLDERS:	• None					
DESCRIPTION:	This use case describes the process in which the administrator wishes to add new VAT details to be applied on the system. The use case begins when the administrator user requests the system to add a VAT Value that is not yet defined on the system. The system will then prompt the user to enter the VAT value which represents the percentage to be applied and select the status to be applied upon adding the new VAT instance’s details. The system will display a confirmation alert once the new VAT details have been created successfully.					
PRE-CONDITION:	• The administrator must be logged onto the system.					
TRIGGER:	The use case is triggered when the administrator wants to add a new VAT details to the system.					
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE				
		MANUAL ACTION	AUTOMATED ACTION			
	Step 1: The administrator clicks on the VAT navigation link on the admin side navigation bar.		Step 2: The system retrieves the following attribute values for each existing entry in the VAT Table using a .NET controller method with a SQL Read query: <table><tr><td>Details to be Read</td><td>Attribute in Table</td><td>Notes</td></tr></table>		Details to be Read	Attribute in Table
Details to be Read	Attribute in Table	Notes				

			VAT Value	VATValue	The numeric value assigned to each VAT instance retrieved and appended with a hard-coded % symbol.
			Status	VATStatus	The current status assigned to each VAT instance retrieved.
			Created Date	CreatedDate	The created date value for each existing instance.
			and displays the retrieved details on the VAT Index Screen , which contains the following controls:		
			Control Name	Control Type	Notes
			VAT	Heading	Page main heading.
			Add VAT	Button	When clicked, this control will direct the administrator to the Create VAT Details screen.
			Show "" entries	Label and drop down	This control will be used to set the number of entries the administrator wants to display on a page
			Search bar	Textbox	This control will be used to enter the search query
			Results table	Table	This control will be used to display all the VAT Amount s that exist on the system
			VAT Value	Table Heading	This control will be used as a heading to display the VAT Value attribute values of each instance.

			Status	Table Heading	Column which holds the status value retrieved for each instance.
			Action	Table Heading	This control will be used as a heading to display the actions the administrator can perform with each entry – Edit and Delete
			Showing “” to “” of “” entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database
			“Previous” and “Next”	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on
			Step 3: The administrator clicks on the Add VAT button.		
			Step 4: The system displays the Create VAT Details Screen which prompts the administrator to capture the new details to be added. The screen contains a MVC form with the following controls:		
			Control Name	Control Type	Notes
			Create VAT Details	Heading	Page main heading
			VAT Value (excl. %)	Label	This label will provide context for the VAT Value textbox control.
			VAT Value	Textbox	This control will be used to capture VAT numeric value to be used for calculating VAT when placing orders

					on the system.				
	Status	Label			Label control used for the Status Drop-Down List control.				
	Status	Drop- Down List			Enables the selection of either 'Active' or 'Inactive'				
	Create	Button			This control will be used to confirm the creation of the VAT Details				
	Cancel	Button			This control will be used to cancel the creation of the VAT Details and return the administrator to the VAT Index Screen .				
	Step 5: The system administrator enters the requested details and clicks on the Create button. [ALT]		Step 6: The system captures and validates the entered details against the validation requirements which have been configured for the VAT Table :						
			<table><tr><th>Details to be Validated</th><th>Attribute in the table</th><th>Notes</th></tr><tr><td>VAT Value</td><td>VATValue</td><td>Required, decimal number that cannot be less than 0.</td></tr><tr><td>VAT Status</td><td>VATStatus</td><td>Required. Either 'Active' or 'Inactive' values can be assigned.</td></tr></table> [ALT]	Details to be Validated	Attribute in the table	Notes	VAT Value	VATValue	Required, decimal number that cannot be less than 0.
Details to be Validated	Attribute in the table	Notes							
VAT Value	VATValue	Required, decimal number that cannot be less than 0.							
VAT Status	VATStatus	Required. Either 'Active' or 'Inactive' values can be assigned.							
			Step 7: The system uses a .NET controller method for performing a SQL READ query to retrieve the following attribute values for all existing VAT instances in the VAT entity:						
			<table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>VAT Value</td><td>VATValue</td></tr></table>	Details to be read	Attributes in the table	VAT Value	VATValue		
Details to be read	Attributes in the table								
VAT Value	VATValue								

			The system then compares the captured VATValue value to all the existing records' assigned VATValue attribute value to confirm whether it is unique across all existing instances. [ALT]														
			Step 8: The system uses a .NET controller method for performing a SQL READ query to retrieve the following attribute values for all existing VAT instances in the VAT entity: <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Status</td><td>VATStatus</td></tr></table> The system then performs a check to determine whether there are any other VAT instances that already have a status of 'Active' assigned only if the status captured for the VAT details to be created is 'Active'. [ALT]	Details to be read	Attributes in the table	Status	VATStatus										
	Details to be read	Attributes in the table															
Status	VATStatus																
		Step 9: The system saves the following details in the VAT table using a post method within a .NET controller with a SQL INSERT Query: <table><tr><th>Details to be Inserted</th><th>Attribute in the table</th><th>Notes</th></tr><tr><td>VAT ID</td><td>VATID</td><td>A unique system-generated numerical value which increments the previous highest value stored.</td></tr><tr><td>VAT Value</td><td>VATValue</td><td>The decimal number value captured on the form.</td></tr><tr><td>Status</td><td>VATStatus</td><td>Set to either 'Active' or 'Inactive' based on selection.</td></tr><tr><td>Created Date</td><td>CreatedDate</td><td>The date on which the VAT instance was created.</td></tr></table>	Details to be Inserted	Attribute in the table	Notes	VAT ID	VATID	A unique system-generated numerical value which increments the previous highest value stored.	VAT Value	VATValue	The decimal number value captured on the form.	Status	VATStatus	Set to either 'Active' or 'Inactive' based on selection.	Created Date	CreatedDate	The date on which the VAT instance was created.
Details to be Inserted	Attribute in the table	Notes															
VAT ID	VATID	A unique system-generated numerical value which increments the previous highest value stored.															
VAT Value	VATValue	The decimal number value captured on the form.															
Status	VATStatus	Set to either 'Active' or 'Inactive' based on selection.															
Created Date	CreatedDate	The date on which the VAT instance was created.															

ALTERNATE COURSES:			Step 10: The system directs the administrator to the VAT Index Screen and displays a bootstrap alert notification with the following controls:															
			<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"VAT details created successfully"</td><td>Label</td><td>This control is used to indicate a successful creation</td></tr></table>	Control Name	Control Name	Notes	"VAT details created successfully"	Label	This control is used to indicate a successful creation									
	Control Name	Control Name	Notes															
	"VAT details created successfully"	Label	This control is used to indicate a successful creation															
	ALT Step 5: The administrator does not wish to add VAT details anymore - The administrator clicks on the cancel button - The use case is terminated																	
	ALT Step 6: The details entered are not in the correct format or the required fields are empty The system displays a validation error with the following controls																	
	<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>VAT Percentage</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>"The VAT Value field is required" OR "Invalid VAT Value provided"</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the VAT Value textbox</td></tr><tr><td>Status</td><td>Drop-Down List</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>"Status field is required"</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Status drop-down control.</td></tr></table>	Control Name	Control Name	Notes	VAT Percentage	Textbox	This control will be outline in red to show a validation error	"The VAT Value field is required" OR "Invalid VAT Value provided"	Label	This control will be used as a label to indicate a validation error in the VAT Value textbox	Status	Drop-Down List	This control will be outline in red to show a validation error	"Status field is required"	Label	This control will be used as a label to indicate a validation error in the Status drop-down control.		
	Control Name	Control Name	Notes															
	VAT Percentage	Textbox	This control will be outline in red to show a validation error															
	"The VAT Value field is required" OR "Invalid VAT Value provided"	Label	This control will be used as a label to indicate a validation error in the VAT Value textbox															
Status	Drop-Down List	This control will be outline in red to show a validation error																
"Status field is required"	Label	This control will be used as a label to indicate a validation error in the Status drop-down control.																
Return to Step 5																		
ALT Step 7: The VAT Value provided already exists on the system : The system displays an error modal dialogue with the following controls																		
<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>VAT Value already exists!</td><td>Label</td><td>This control is used to inform the administrator that the VAT Value attribute value is already assigned to another instance.</td></tr><tr><td>Okay</td><td>Button</td><td>This control is used to acknowledge the validation error displayed.</td></tr></table>	Control Name	Control Name	Notes	VAT Value already exists!	Label	This control is used to inform the administrator that the VAT Value attribute value is already assigned to another instance.	Okay	Button	This control is used to acknowledge the validation error displayed.									
Control Name	Control Name	Notes																
VAT Value already exists!	Label	This control is used to inform the administrator that the VAT Value attribute value is already assigned to another instance.																
Okay	Button	This control is used to acknowledge the validation error displayed.																
- The administrator clicks the "Okay" button - The use case is terminated																		
ALT Step 8: Another VAT instance already has an 'Active' status value assigned on the system : The system displays an error modal dialogue with the following controls																		
<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Only one VAT instance may be Active at a time.</td><td>Label</td><td>This control is used to inform the administrator that another VAT instance is already 'Active'.</td></tr><tr><td>Okay</td><td>Button</td><td>This control is used to acknowledge the validation error displayed.</td></tr></table>	Control Name	Control Name	Notes	Only one VAT instance may be Active at a time.	Label	This control is used to inform the administrator that another VAT instance is already 'Active'.	Okay	Button	This control is used to acknowledge the validation error displayed.									
Control Name	Control Name	Notes																
Only one VAT instance may be Active at a time.	Label	This control is used to inform the administrator that another VAT instance is already 'Active'.																
Okay	Button	This control is used to acknowledge the validation error displayed.																

	The administrator clicks the “Okay” button The use case is terminated
CONCLUSION:	This use case concludes when the system displays a bootstrap notification informing the administrator that the VAT details have been created successfully.
POST-CONDITION:	The captured VAT details is added to the VAT entity in the database.
BUSINESS RULES	- Only administrator users should be allowed to create new VAT details on the system. - The details of all VAT instances are to be unique. - Only one VAT instance should be active at a time.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	- There may never be two VAT instances with a status value of ‘Active’ assigned. - Duplicate VATValue attribute values may not exist at any time.
ASSUMPTIONS:	None
OPEN ISSUES:	None

Table 129 - 13.2 Search VAT

VAT SUBSYSTEM				
AUTHORS (s): Aphiwe Shozi			DATE: 06/06/2022	
VERSION: 1.1			LAST QUERY DATE: 23/07/2022	
USE CASE NAME:	Search VAT		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒	
USE CASE ID:	13.2			
PRIORITY:	Medium			
SOURCE:	ByteXcom Requirements List			
PRIMARY BUSINESS ACTOR	• Administrator			
PRIMARY SYSTEM ACTOR	• None			
OTHER PARTICIPATING ACTORS:	• None			
OTHER INTERESTED STAKEHOLDERS:	• None			
DESCRIPTION:	This use case describes the process in the administrator wishes to search for existing VAT Details defined in the VAT entity. The administrator user enters the details for which they wish to base the search on and the system will filter all details retrieved from the VAT entity which have matched the search query specified. This use case concludes once the system has filtered the list of VAT details displayed in the table containing all VAT details.			
PRE-CONDITION:	The administrator must be logged onto the system.			
TRIGGER:	This use case is called when the administrator requests to search for existing VAT details defined on the system.			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE		
		MANUAL ACTION	AUTOMATED ACTION	
	Step 1: The administrator clicks on the VAT navigation link on the admin side-navigation bar.		Step 2: The system retrieves the following attribute values for each existing entry in the VAT Table using a .NET controller method with a SQL Read query:	
			Details to be Read	Attribute in Table
		VAT Value	VATValue	The numeric value assigned to each VAT

					instance retrieved and appended with % symbol.
			Status	VATStatus	The current status assigned to each VAT instance retrieved.
			and displays the retrieved details on the VAT Index Screen , which contains the following controls:		
			Control Name	Control Type	Notes
			VAT	Heading	Page main heading.
			Add VAT	Button	When clicked, this control will direct the administrator to the Create VAT Details screen.
			Show "" entries	Label and drop down	This control will be used to set the number of entries the administrator wants to display on a page
			Search bar	Textbox	This control will be used to enter the search query
			Results table	Table	This control will be used to display all the VAT Amount s that exist on the system
			VAT Value	Table Heading	This control will be used as a heading to display the VAT Value attribute value assigned for each instance.
			Status	Table Heading	Column which holds the status value retrieved for each instance.
			Action	Table Heading	This control will be used as a heading to display the actions the

					administrator can perform with each entry – Edit and Delete
			Showing “” to “” of “” entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database
			“Previous” and “Next”	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on
	Step 3: The administrator enters their search Query into the search textbox.		Step 4: The system filters the table rows in the data table and displays the details of all VAT instances which have matched the search query provided. [ALT]		
	ALT Step 4: The search query entered does not match any records existing in the VAT Table . • The system displays a “No Records Found” message in the results table. • The use case concludes.				
CONCLUSION:	This use case concludes when all the matched VAT instance details are displayed in the data table.				
POST-CONDITION:	All VAT instances which have matched the search query specified are displayed from the VAT entity.				
BUSINESS RULES	• Only users with the Administrator role assigned should be allowed to search for specific VAT details defined on the system.				
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• The user performing the search should be informed if there are no details which have matched the search query specified. • Authorized users can search for VAT instances using either the VATValue or VATStatus attribute values.				
ASSUMPTIONS:	• None				
OPEN ISSUES:	• None				

Table 130 - 13.3 Update VAT Details

VAT SUBSYSTEM			
AUTHORS (s): Aphiwe Shozi		DATE: 06/06/2022	
VERSION: 1.1		LAST QUERY DATE: 23/07/2022	
USE CASE NAME:	Update VAT		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒
USE CASE ID:	13.3		
PRIORITY:	High		
SOURCE:	ByteXcom Requirements List		
PRIMARY BUSINESS ACTOR	• Administrator		
PRIMARY SYSTEM ACTOR	• None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	This use case describes the process in which the administrator requests the system to update the details of an existing VAT record defined in the VAT Table. The use case begins when the administrator requests the system to update an existing VAT instance’s details. The system will then retrieve the current details stored for the requested VAT instance. The administrator will update all of the pertinent information regarding the requested VAT record, and once the updated details have been submitted, the system will validate the captured details. Once validation has passed, the system will save the updated VAT details in the VAT entity. The use case concludes with the system displaying a confirmation alert that informs the administrator that the details have been updated successfully.		
PRE-CONDITION:	The administrator must be logged in to the system.		
TRIGGER:	The use is triggered when an administrator user requests the system to update an existing VAT instance’s details.		
TYPICAL COURSE OF EVENTS:		SYSTEM RESPONSE	
	ACTOR ACTION	MANUAL ACTION	AUTOMATED ACTION
	Step 1: The administrator wishes to update a VAT instance’s details.		Step 2: The system invokes use case 13.2 Search VAT
	Step 3: The administrator clicks on the Update		Step 4: The system retrieves the following attributes from the listed entities:

button for the VAT instance they wish to update.

From the **VAT Table**:

Details to be validated	Attribute in the table	Notes
VAT Value	VATValue	The number value which represents the VAT to be applied.
VAT Status	VATStatus	Current status assigned to the VAT instance.

and displays them on the **Update VAT Details Screen** which contains an MVC form with the following controls:

Control Name	Control Type	Notes
Update VAT Details	Heading	Page's main heading.
VAT Value	Label	Label for the VAT Value textbox control.
VAT Value	Textbox	Populated with the current VATValue attribute value stored for the requested VAT instance.
VAT Status	Label	Used as a label for the VAT Status drop-down list control.
VAT Status	Drop-Down List	Populated with the current VATStatus attribute value of either 'Active' or 'Inactive'
Update	Button	Submits the updated form details if clicked.
Cancel	Button	This control will be used to cancel the update operation and redirects the administrator to the VAT Index Screen .

	Step 5: The system administrator enters the updated VAT details and clicks on the Update button. [ALT]		Step 6: The system captures and validates the entered details against the following table's validation requirements: From the VAT Table: <table><tr><th>Details to be validated</th><th>Attribute in the table</th><th>Notes</th></tr><tr><td>VAT ID</td><td>VATID</td><td>Required. Unique primary key for the stored VAT instance.</td></tr><tr><td>VAT Value</td><td>VATValue</td><td>Required. Value cannot be negative or of a non-numeric type.</td></tr><tr><td>VAT Status</td><td>VATStatus</td><td>Required. Only the values 'Active' and 'Inactive' are allowed.</td></tr></table> [ALT]	Details to be validated	Attribute in the table	Notes	VAT ID	VATID	Required. Unique primary key for the stored VAT instance.	VAT Value	VATValue	Required. Value cannot be negative or of a non-numeric type.	VAT Status	VATStatus	Required. Only the values 'Active' and 'Inactive' are allowed.
	Details to be validated	Attribute in the table	Notes												
	VAT ID	VATID	Required. Unique primary key for the stored VAT instance.												
VAT Value	VATValue	Required. Value cannot be negative or of a non-numeric type.													
VAT Status	VATStatus	Required. Only the values 'Active' and 'Inactive' are allowed.													
		Step 7: The system displays a modal dialogue with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>"Confirm VAT Details"</td><td>Label</td><td>This control is used to request confirmation on the updated VAT details</td></tr><tr><td>Confirm</td><td>Button</td><td>This control is used to confirm the updating of the VAT details captured.</td></tr><tr><td>Cancel</td><td>Button</td><td>This control is used to close the modal dialogue and return the administrator to the Update VAT Details Screen.</td></tr></table>	Control Name	Control Type	Notes	"Confirm VAT Details"	Label	This control is used to request confirmation on the updated VAT details	Confirm	Button	This control is used to confirm the updating of the VAT details captured.	Cancel	Button	This control is used to close the modal dialogue and return the administrator to the Update VAT Details Screen .	
Control Name	Control Type	Notes													
"Confirm VAT Details"	Label	This control is used to request confirmation on the updated VAT details													
Confirm	Button	This control is used to confirm the updating of the VAT details captured.													
Cancel	Button	This control is used to close the modal dialogue and return the administrator to the Update VAT Details Screen .													
Step 8: The system administrator clicks on the Confirm button.		Step 9: The system uses a SQL READ query to retrieve the following details for all VAT instances defined on the system:													

	[ALT]		From the VAT entity: <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>VAT Value</td><td>VATValue</td></tr><tr><td>VATStatus</td><td>VatStatus</td></tr></table> The retrieved details is used to verify if the specified VATValue value already is assigned to another VAT instance and to check if another VAT instance is already 'Active' if the updated VAT instance's status has been changed from 'Inactive' to 'Active'. [ALT]	Details to be read	Attributes in the table	VAT Value	VATValue	VATStatus	VatStatus		
	Details to be read	Attributes in the table									
	VAT Value	VATValue									
VATStatus	VatStatus										
		Step 10: The system saves the updated details to the VAT table using an SQL UPDATE Query: <table><tr><th>Details to be Updated</th><th>Attribute in the table</th><th>Notes</th></tr><tr><td>VAT Value</td><td>VATValue</td><td>The updated VAT percentage value is saved if unique.</td></tr><tr><td>VAT Status</td><td>VATStatus</td><td>The updated status value is saved.</td></tr></table>	Details to be Updated	Attribute in the table	Notes	VAT Value	VATValue	The updated VAT percentage value is saved if unique.	VAT Status	VATStatus	The updated status value is saved.
Details to be Updated	Attribute in the table	Notes									
VAT Value	VATValue	The updated VAT percentage value is saved if unique.									
VAT Status	VATStatus	The updated status value is saved.									
		Step 11: The system redirects the administrator to the VAT Index Screen and displays a bootstrap alert notification with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"VAT details updated successfully"</td><td>Label</td><td>This control is used to indicate that the VAT instance's details have been updated successfully.</td></tr></table>	Control Name	Control Name	Notes	"VAT details updated successfully"	Label	This control is used to indicate that the VAT instance's details have been updated successfully.			
Control Name	Control Name	Notes									
"VAT details updated successfully"	Label	This control is used to indicate that the VAT instance's details have been updated successfully.									
ALTERNATE COURSES:	ALT Step 4: The administrator does not wish to proceed with the update operation: • The administrator clicks on the Cancel button. • The administrator is redirected to the VAT Index screen. • The use case is terminated.										
	ALT Step 5: The details entered are not in the correct format or the required fields are empty										

	The system displays a validation error controls for the following where applicable:															
	<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>VAT Value</td><td>Textbox</td><td>This control will be outline in red to show a validation error has occurred.</td></tr><tr><td>"VAT Value field is required"</td><td>Label</td><td>This control will be used as a label to indicate that the VAT Value field requires valid input.</td></tr><tr><td>VAT Status</td><td>Drop-Down List</td><td>This control will be outline in red to show a validation error to show that a validation error has occurred.</td></tr><tr><td>"VAT Status field is required"</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the VAT Status Drop-Down List control.</td></tr></table>	Control Name	Control Name	Notes	VAT Value	Textbox	This control will be outline in red to show a validation error has occurred.	"VAT Value field is required"	Label	This control will be used as a label to indicate that the VAT Value field requires valid input.	VAT Status	Drop-Down List	This control will be outline in red to show a validation error to show that a validation error has occurred.	"VAT Status field is required"	Label	This control will be used as a label to indicate a validation error in the VAT Status Drop-Down List control.
	Control Name	Control Name	Notes													
	VAT Value	Textbox	This control will be outline in red to show a validation error has occurred.													
	"VAT Value field is required"	Label	This control will be used as a label to indicate that the VAT Value field requires valid input.													
	VAT Status	Drop-Down List	This control will be outline in red to show a validation error to show that a validation error has occurred.													
	"VAT Status field is required"	Label	This control will be used as a label to indicate a validation error in the VAT Status Drop-Down List control.													
	Return to Step 4.															
	ALT Step 7: The administrator clicks on the Cancel button: Return to Step 4															
	ALT Step 8: The VAT Value specified is already assigned to another VAT instance or another VAT instance is already active: The system displays an error modal dialogue with the following controls															
<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>VAT Value already exists!</td><td>Label</td><td>This control is used to inform the administrator that the VAT Value already exists.</td></tr><tr><td>Another VAT instance is already Active</td><td>Label</td><td>This control is used to inform the administrator that another VAT instance already has a status of 'Active' assigned.</td></tr><tr><td>Okay</td><td>Button</td><td>This control is used to acknowledge the validation error modal dialogue control.</td></tr></table>	Control Name	Control Name	Notes	VAT Value already exists!	Label	This control is used to inform the administrator that the VAT Value already exists.	Another VAT instance is already Active	Label	This control is used to inform the administrator that another VAT instance already has a status of 'Active' assigned.	Okay	Button	This control is used to acknowledge the validation error modal dialogue control.				
Control Name	Control Name	Notes														
VAT Value already exists!	Label	This control is used to inform the administrator that the VAT Value already exists.														
Another VAT instance is already Active	Label	This control is used to inform the administrator that another VAT instance already has a status of 'Active' assigned.														
Okay	Button	This control is used to acknowledge the validation error modal dialogue control.														
- The administrator clicks the Okay button. - Return to Step 4.																
CONCLUSION:	This use case concludes when the system displays a Bootstrap alert which informs the administrator that the VAT instance's details have been updated successfully.															
POST-CONDITION:	The VAT record's details are updated in the VAT entity.															
BUSINESS RULES	- Only the administrator user should be allowed to update a VAT instance's details using this functionality. - Only one VAT instance may have a status of 'Active' assigned at all times.															
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	- The VATValue attribute value is to be captured as a decimal value which is used for VAT calculations. - Prevent users from creating duplicate VATValue attribute values on the system. - Prevent users to update VAT details if it will result in more than one VAT instance having a status of 'Active' assigned at the same time.															

	The system is offline or the administrator has no internet connection, resulting in the administrator being unable to update the details of a particular VAT instance on the system.
ASSUMPTIONS:	None
OPEN ISSUES:	None

Table 131 - 13.4 Delete VAT

VAT SUBSYSTEM			
AUTHORS (s): Aphiwe Shoji		DATE: 06/06/2022	
VERSION: 1.1		LAST QUERY DATE: 23/07/2022	
USE CASE NAME:	Delete VAT		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒
USE CASE ID:	13.4		
PRIORITY:	High		
SOURCE:	ByteXcom Requirements List		
PRIMARY BUSINESS ACTOR	• Administrator		
PRIMARY SYSTEM ACTOR	• None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	This use case narrative describes the process in which the administrator wishes to delete a VAT instance’s details from the system. The administrator will search for the required VAT Amount to be deleted. The system then prompts the administrator to confirm the removal of the requested VAT instance with a modal dialogue box. After the administrator confirms the removal, the system will perform a check to see if the VAT instance has been applied to any orders in the Order entity. If no orders are associated, the system will remove the VAT instance details from the system and the use case concludes with the system displaying a confirmation alert which informs the administrator that the requested details have been removed successfully.		
PRE-CONDITION:	The administrator must be logged in to the system.		
TRIGGER:	This use case is called when the administrator wishes to delete a VAT instance’s details from the system.		
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE	
		MANUAL ACTION	AUTOMATED ACTION
	Step 1: The administrator wishes to remove a particular VAT instance’s details from the system.		Step 2: The system invokes use case 13.2 Search VAT
	Step 3: The administrator clicks on the Delete button		Step 4: The system retrieves the following attributes details for the requested VAT instance using a .NET

	for the specific VAT record they wish to remove.		controller method with a SQL Read query: From the VAT Table: <table><tr><th>Details to be Read</th><th>Attributes in the table</th></tr><tr><td>VAT Value</td><td>VATValue</td></tr><tr><td>VAT Status</td><td>VATStatus</td></tr><tr><td>Created Date</td><td>CreatedDate</td></tr></table> and displays the retrieved details on the Delete VAT Details Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Delete VAT Details</td><td>Heading</td><td>Page's main heading.</td></tr><tr><td>VAT Value</td><td>Label</td><td>Populated with the VATValue value retrieved followed by a hard-coded percentage sign. E.g. 15 %.</td></tr><tr><td>VAT Status</td><td>Label</td><td>Populated with the VATStatus attribute value retrieved.</td></tr><tr><td>Created Date</td><td>Label</td><td>Populated with the created date attribute value retrieved.</td></tr><tr><td>Delete</td><td>Button</td><td>Triggers the deletion confirmation modal dialogue box to be displayed.</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the deletion operation and navigates the user to the VAT Index Screen</td></tr></table>	Details to be Read	Attributes in the table	VAT Value	VATValue	VAT Status	VATStatus	Created Date	CreatedDate	Control Name	Control Type	Notes	Delete VAT Details	Heading	Page's main heading.	VAT Value	Label	Populated with the VATValue value retrieved followed by a hard-coded percentage sign. E.g. 15 %.	VAT Status	Label	Populated with the VATStatus attribute value retrieved.	Created Date	Label	Populated with the created date attribute value retrieved.	Delete	Button	Triggers the deletion confirmation modal dialogue box to be displayed.	Cancel	Button	This control will be used to cancel the deletion operation and navigates the user to the VAT Index Screen
	Details to be Read	Attributes in the table																														
	VAT Value	VATValue																														
	VAT Status	VATStatus																														
	Created Date	CreatedDate																														
	Control Name	Control Type	Notes																													
	Delete VAT Details	Heading	Page's main heading.																													
	VAT Value	Label	Populated with the VATValue value retrieved followed by a hard-coded percentage sign. E.g. 15 %.																													
	VAT Status	Label	Populated with the VATStatus attribute value retrieved.																													
	Created Date	Label	Populated with the created date attribute value retrieved.																													
Delete	Button	Triggers the deletion confirmation modal dialogue box to be displayed.																														
Cancel	Button	This control will be used to cancel the deletion operation and navigates the user to the VAT Index Screen																														
Step 5: The administrator clicks on the Delete button.		Step 6: The system displays a modal dialogue with the following controls:																														

	[ALT]		<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>"Confirm VAT Amount Deletion"</td><td>Label</td><td>This control is used to request confirmation on the deletion of the VAT Amount details</td></tr><tr><td>Confirm</td><td>Button</td><td>This control is used to confirm the deletion of the VAT Amount details</td></tr><tr><td>Cancel</td><td>Button</td><td>This control is used to close the modal dialogue</td></tr></table>	Control Name	Control Type	Notes	"Confirm VAT Amount Deletion"	Label	This control is used to request confirmation on the deletion of the VAT Amount details	Confirm	Button	This control is used to confirm the deletion of the VAT Amount details	Cancel	Button	This control is used to close the modal dialogue
	Control Name	Control Type	Notes												
	"Confirm VAT Amount Deletion"	Label	This control is used to request confirmation on the deletion of the VAT Amount details												
	Confirm	Button	This control is used to confirm the deletion of the VAT Amount details												
	Cancel	Button	This control is used to close the modal dialogue												
Step 7: The system administrator clicks on the confirm button [ALT]		Step 8: The system uses a .NET controller method with a SQL READ query to retrieve the following details from the Order entity: <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>VAT ID</td><td>VATID</td></tr></table> The system uses the retrieved information to determine whether the VAT instance requested for removal is associated with any existing orders on the system. [ALT]	Details to be read	Attributes in the table	VAT ID	VATID									
Details to be read	Attributes in the table														
VAT ID	VATID														
		Step 9: The system deletes the selected VAT instance's details from the VAT entity in the database using an SQL DELETE Query: Details to be deleted from the VAT entity: <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>VAT ID</td><td>VATID</td></tr><tr><td>VAT Value</td><td>VATValue</td></tr><tr><td>VAT Status</td><td>VATStatus</td></tr><tr><td>Created Date</td><td>CreatedDate</td></tr></table>	Details to be read	Attributes in the table	VAT ID	VATID	VAT Value	VATValue	VAT Status	VATStatus	Created Date	CreatedDate			
Details to be read	Attributes in the table														
VAT ID	VATID														
VAT Value	VATValue														
VAT Status	VATStatus														
Created Date	CreatedDate														
		Step 10: The system redirects the administrator to the VAT Screen and displays a modal dialogue with the following controls:													

			<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"VAT Details Removed Successfully"</td><td>Label</td><td>This control is used to indicate the successful deletion of the requested VAT instance.</td></tr></table>	Control Name	Control Name	Notes	"VAT Details Removed Successfully"	Label	This control is used to indicate the successful deletion of the requested VAT instance.			
Control Name	Control Name	Notes										
"VAT Details Removed Successfully"	Label	This control is used to indicate the successful deletion of the requested VAT instance.										
ALTERNATE COURSES:	ALT Step 5: The administrator does not want to proceed with the removal the VAT instance's details anymore: - The administrator clicks on the Cancel button. - The administrator is redirected to the VAT Index screen. - The use case is terminated.											
	ALT Step 7: The administrator does not wish to proceed with the removal of the requested VAT record: - The administrator clicks on the Cancel button - The administrator is redirected to the VAT Index screen. - The use case concludes.											
	ALT Step 8: The VAT instance's details cannot be deleted since it is associated to an order on the system: The system displays a bootstrap alert notification with the following controls:											
	<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>"VAT cannot be deleted since it is associated to an order"</td><td>Label</td><td>This control is used to indicate that the VAT details cannot be deleted if it is associated to an order which has been placed on the system.</td></tr><tr><td>Okay</td><td>Button</td><td>Used to acknowledge the exception message displayed.</td></tr></table>	Control Name	Control Name	Notes	"VAT cannot be deleted since it is associated to an order"	Label	This control is used to indicate that the VAT details cannot be deleted if it is associated to an order which has been placed on the system.	Okay	Button	Used to acknowledge the exception message displayed.		
	Control Name	Control Name	Notes									
	"VAT cannot be deleted since it is associated to an order"	Label	This control is used to indicate that the VAT details cannot be deleted if it is associated to an order which has been placed on the system.									
Okay	Button	Used to acknowledge the exception message displayed.										
- The administrator clicks on the Okay button. - The administrator is redirected to the VAT Index screen. - The use case concludes.												
CONCLUSION:	This use case concludes when the system displays an alert which informs the administrator that the VAT record's details has been removed successfully.											
POST-CONDITION:	The requested VAT instance's details are removed from the VAT entity.											
BUSINESS RULES	Only administrator users should be allowed to remove VAT details from the system.											
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	- The system should restrict the removal of a VAT instance's details if it is associated with an order which has been placed on the system. - The system is offline or the administrator does not have access to the internet, resulting in the administrator user not being able to remove VAT details from the system at that time.											
ASSUMPTIONS:	None											
OPEN ISSUES:	None											

2.14. Reseller Credit Settlement Time Subsystem

Table 132 - 14.1 Update Reseller Credit Settlement Time

SUPPLIER SUBSYSTEM								
AUTHORS (s): Nomusa Vumisa		DATE: 22/07/2022						
VERSION: 1.1		LAST REVIEW DATE: 22/07/2022						
USE CASE NAME:	Update Reseller Credit Settlement Time	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒						
USE CASE ID:	14.1							
PRIORITY:	High							
SOURCE:	ByteXpress Requirements List							
PRIMARY BUSINESS ACTOR	• Administrator							
PRIMARY SYSTEM ACTOR	• None							
OTHER PARTICIPATING ACTORS:	• None							
OTHER INTERESTED STAKEHOLDER S:	• None							
DESCRIPTION:	This use case describes the process in which the administrator wishes to update the reseller credit settlement time on the system. The administrator will then update the settlement time and save it to the database. The use case ends when the administrator receives a successful update notification.							
PRE-CONDITION:	The administrator must be logged onto the system							
TRIGGER:	The use case is triggered when the administrator wants to update the reseller credit settlement time							
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE						
		MANUAL ACTION	AUTOMATED ACTION					
	Step 1: The administrator clicks on the Payment Reminders link in the Orders dropdown on the admin side navigation bar		Step 2: The system retrieves the following attribute from the PaymentReminder table, using a method in a .NET controller and an SQL READ query:					
			<table><tr><td>Details to be read</td><td>Attributes in the table</td></tr><tr><td>Days</td><td>Days</td></tr></table>		Details to be read	Attributes in the table	Days	Days
			Details to be read	Attributes in the table				
Days	Days							
and displays it on the Update Reseller Credit Settlement Time Screen which contains an MVC form with the following controls:								

			<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Update Reseller Credit Settlement Time</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Help icon</td><td>Button</td><td>When clicked, the admin help document will open</td></tr><tr><td>Current Time</td><td>Label</td><td>This control will be used as a label to display the current settlement time</td></tr><tr><td>Settlement Time Intervals</td><td>Label</td><td>This control will be used as a label to indicate the settlement time intervals</td></tr><tr><td>Settlement Time Intervals</td><td>Dropdown list</td><td>This control will be used by the administrator to select the settlement time intervals</td></tr><tr><td>Update</td><td>Button</td><td>This control will be used to confirm the updating of the settlement time</td></tr></table>	Control Name	Control Type	Notes	Update Reseller Credit Settlement Time	Heading	Page main heading	Help icon	Button	When clicked, the admin help document will open	Current Time	Label	This control will be used as a label to display the current settlement time	Settlement Time Intervals	Label	This control will be used as a label to indicate the settlement time intervals	Settlement Time Intervals	Dropdown list	This control will be used by the administrator to select the settlement time intervals	Update	Button	This control will be used to confirm the updating of the settlement time
	Control Name	Control Type	Notes																					
	Update Reseller Credit Settlement Time	Heading	Page main heading																					
	Help icon	Button	When clicked, the admin help document will open																					
	Current Time	Label	This control will be used as a label to display the current settlement time																					
	Settlement Time Intervals	Label	This control will be used as a label to indicate the settlement time intervals																					
	Settlement Time Intervals	Dropdown list	This control will be used by the administrator to select the settlement time intervals																					
	Update	Button	This control will be used to confirm the updating of the settlement time																					
			Note: All controls are enabled by default, unless stated otherwise.																					
	Step 3: The system administrator selects their desired settlement time interval and clicks on the update button		Step 4: The system displays a modal dialogue with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>"Are you sure you want to update the settlement time"</td><td>Label</td><td>This control is used to request confirmation on the updated settlement time</td></tr><tr><td>Confirm</td><td>Button</td><td>This control is used to confirm the updating of the settlement time</td></tr><tr><td>Cancel</td><td>Button</td><td>This control is used to close the modal dialogue</td></tr></table>	Control Name	Control Type	Notes	"Are you sure you want to update the settlement time"	Label	This control is used to request confirmation on the updated settlement time	Confirm	Button	This control is used to confirm the updating of the settlement time	Cancel	Button	This control is used to close the modal dialogue									
	Control Name	Control Type	Notes																					
	"Are you sure you want to update the settlement time"	Label	This control is used to request confirmation on the updated settlement time																					
	Confirm	Button	This control is used to confirm the updating of the settlement time																					
Cancel	Button	This control is used to close the modal dialogue																						

	Step 5: The system administrator clicks on the confirm button [ALT]		Step 6: The system uses an SQL READ query to check if the settle time is not already active in the PaymentReminder table using the following attribute. [ALT] <table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Is Active</td><td>IsActive</td></tr></table>	Details to be read	Attributes in the table	Is Active	IsActive	
	Details to be read	Attributes in the table						
	Is Active	IsActive						
		Step 7: The system activates the selected settlement time in the PaymentReminder table using a method in a .NET controller and an SQL UPDATE query: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Is Active</td><td>Is Active</td><td>Updates the selected settlement time record to “True” and the old settlement time record to “False”</td></tr></table>	Details to be saved	Attributes in the table	Notes	Is Active	Is Active	Updates the selected settlement time record to “True” and the old settlement time record to “False”
Details to be saved	Attributes in the table	Notes						
Is Active	Is Active	Updates the selected settlement time record to “True” and the old settlement time record to “False”						
		Step 8: The system displays a bootstrap alert notification with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>“Reseller Credit Settlement Time updated successfully”</td><td>Label</td><td>This control is used to indicate a successful update</td></tr></table>	Control Name	Control Name	Notes	“Reseller Credit Settlement Time updated successfully”	Label	This control is used to indicate a successful update
Control Name	Control Name	Notes						
“Reseller Credit Settlement Time updated successfully”	Label	This control is used to indicate a successful update						
ALTERNATE COURSES:	ALT Step 5: The administrator clicks on the cancel button Return to Step 2 or terminate the use case							
	ALT Step 6: The settlement time is already active on the system - The system displays a bootstrap alert notification with the following controls<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>“The settlement time is already active”</td><td>Label</td><td>This control is used to inform the administrator that the settlement time is already active</td></tr></table> - The use case is terminated			Control Name	Control Name	Notes	“The settlement time is already active”	Label
Control Name	Control Name	Notes						
“The settlement time is already active”	Label	This control is used to inform the administrator that the settlement time is already active						
CONCLUSION:	This use case concludes when the settlement time is updated.in the Payment Reminder table and the system displays an alert notification that informs the administrator that the settlement time has been successfully updated.							
POST-CONDITION:	The settlement time was updated in the PaymentReminder Table							
BUSINESS RULES	Only one settlement time can be active at a time							
IMPLEMENTATI ON CONSTRAINTS AND	- The settlement time cannot be updated without an internet connection - Only the administrator can update the settlement time							

SPECIFICATIONS	
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

2.15. Video Subsystem

Table 133 - 15.1 Add Video

VIDEO SUBSYSTEM			
AUTHORS (s): Thenjiwe Ntsonda		DATE: 18/04/2022	
VERSION: 1.0		LAST REVIEW DATE: 23/08/2022	
USE CASE NAME:	Add Video	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒	
USE CASE ID:	15.1		
PRIORITY:	High		
SOURCE:	ByteXcom Requirements List		
PRIMARY BUSINESS ACTOR	• Administrator		
PRIMARY SYSTEM ACTOR	• None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	This use case describes the process in which the administrator wishes to add a new video to the system. The use case begins when the administrator adds the new video. The administrator will add all the pertinent information regarding the video and save it to the business database.		
PRE-CONDITION:	• The administrator must be logged onto the system.		
TRIGGER:	The use case is triggered when the administrator wants to add a new video.		
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE	
		MANUAL ACTION	AUTOMATED ACTION

Step 1: The administrator navigates to the **Video Screen** under the Video tab on the side bar

Step 2: The system retrieves the video information from the following attributes and tables using a method in a .NET Controller and an SQL READ Query:

From the **Video** table:

Details to be read	Attributes in the table
Name	Name
Video Link	VideoUrl

And displays the **Video Screen** with the following controls:

Control Name	Control Type	Notes
Video List	Heading	Page main heading
Question Mark Icon	Button	When clicked, this control will open a pdf which contains the help document
Create New Video	Button	When clicked, this control will direct the user to the Create video page
Search bar	Textbox	This control will be used when the user inserts the search parameters
Search icon	Button	This control will be used to activate the search
Results table	Table	This control will be used to display all the videos that exist on the system
Name	Label	This control will be used as a heading to display the Name for all videos
Video Link	Label	This control will be used as a heading to display a video link for all videos

			Action - Update	Button	This control will be placed in a row, with the delete button, under the "Action" heading for each entry in the Video table. The button will direct the user to the Update Video screen
			Action - Delete	Button	This control will be placed in a row, with the delete button, under the "Action" heading for each entry in the Video table. The button will direct the user to the Delete Video Details screen
			Showing "" of "" entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database
			"Previous" and "Next"	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on

	Step 3: The system administrator clicks on the Create New Video button		Step 4: The system displays the Create Video screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Create Video</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Video information</td><td>Heading</td><td>Page subheading</td></tr><tr><td>Name</td><td>Label</td><td>This control will be used as a label to indicate the Name attribute of the video record</td></tr><tr><td>Name</td><td>Textbox</td><td>This control will be used by the administrator to insert a name of the video</td></tr><tr><td>Video Link</td><td>Label</td><td>This control will be used as a label to indicate the Video Link attribute of the video record</td></tr><tr><td>Video Link</td><td>Textbox</td><td>This control will be used by the administrator to insert the link of the video</td></tr><tr><td>Create</td><td>Button</td><td>This control will be used to confirm the addition of the video</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the addition of the video</td></tr></table>	Control Name	Control Type	Notes	Create Video	Heading	Page main heading	Video information	Heading	Page subheading	Name	Label	This control will be used as a label to indicate the Name attribute of the video record	Name	Textbox	This control will be used by the administrator to insert a name of the video	Video Link	Label	This control will be used as a label to indicate the Video Link attribute of the video record	Video Link	Textbox	This control will be used by the administrator to insert the link of the video	Create	Button	This control will be used to confirm the addition of the video	Cancel	Button	This control will be used to cancel the addition of the video
	Control Name	Control Type	Notes																											
	Create Video	Heading	Page main heading																											
	Video information	Heading	Page subheading																											
	Name	Label	This control will be used as a label to indicate the Name attribute of the video record																											
	Name	Textbox	This control will be used by the administrator to insert a name of the video																											
	Video Link	Label	This control will be used as a label to indicate the Video Link attribute of the video record																											
	Video Link	Textbox	This control will be used by the administrator to insert the link of the video																											
	Create	Button	This control will be used to confirm the addition of the video																											
	Cancel	Button	This control will be used to cancel the addition of the video																											
			Note: All controls are enabled by default unless stated otherwise.																											

	Step 5: The system administrator enters the details and clicks on the Create button [ALT]		Step 6: The system captures and validates the entered details against the Video table [ALT] <table><tr><th>Details to be validated</th><th>Attributes in the video table</th><th>Notes</th></tr><tr><td>Name</td><td>Name</td><td>Required, Unique</td></tr><tr><td>Video Link</td><td>VideoUrl</td><td>Required, Unique</td></tr></table>	Details to be validated	Attributes in the video table	Notes	Name	Name	Required, Unique	Video Link	VideoUrl	Required, Unique		
	Details to be validated	Attributes in the video table	Notes											
	Name	Name	Required, Unique											
	Video Link	VideoUrl	Required, Unique											
		Step 7: The system uses an SQL READ query to check if the video already exists in the Video table using the following attributes [ALT] <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Name</td><td>Name</td><td></td></tr><tr><td>Video Link</td><td>VideoUrl</td><td></td></tr></table>	Details to be read	Attributes in the table	Notes	Name	Name		Video Link	VideoUrl				
Details to be read	Attributes in the table	Notes												
Name	Name													
Video Link	VideoUrl													
		Step 8: The system saves the information into the Video table using an INSERT SQL query: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Videoid</td><td>Videoid</td><td>A unique value generated by the system and saved to the database.</td></tr><tr><td>Name</td><td>Name</td><td>Required</td></tr><tr><td>Video Link</td><td>VideoUrl</td><td>Required</td></tr></table>	Details to be saved	Attributes in the table	Notes	Videoid	Videoid	A unique value generated by the system and saved to the database.	Name	Name	Required	Video Link	VideoUrl	Required
Details to be saved	Attributes in the table	Notes												
Videoid	Videoid	A unique value generated by the system and saved to the database.												
Name	Name	Required												
Video Link	VideoUrl	Required												
		Step 9: The system redirects the system administrator to the Video Screen and displays a message box with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Checkbox</td><td>Icon</td><td>This control is used to indicate a successful creation</td></tr><tr><td>“Video Created Successfully”</td><td>Label</td><td>This control is used to indicate a successful creation</td></tr></table>	Control Name	Control Name	Notes	Checkbox	Icon	This control is used to indicate a successful creation	“Video Created Successfully”	Label	This control is used to indicate a successful creation			
Control Name	Control Name	Notes												
Checkbox	Icon	This control is used to indicate a successful creation												
“Video Created Successfully”	Label	This control is used to indicate a successful creation												
ALTERNATE COURSES:	ALT Step 6: The administrator does not want to add a video anymore - The administrator clicks on the cancel button - The use case it terminated													
	ALT Step 7: The Video already exists on the system The system displays an error modal dialogue with the following controls													

	•															
	<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Video already exists!</td><td>Label</td><td>This control is used to inform the administrator that the video already exists</td></tr><tr><td>Okay</td><td>Button</td><td>This control is used to cancel the creation of the video</td></tr></table>	Control Name	Control Name	Notes	Video already exists!	Label	This control is used to inform the administrator that the video already exists	Okay	Button	This control is used to cancel the creation of the video						
	Control Name	Control Name	Notes													
	Video already exists!	Label	This control is used to inform the administrator that the video already exists													
	Okay	Button	This control is used to cancel the creation of the video													
	• The administrator clicks the “Okay” button • The use case is terminated															
	ALT Step 8A: The details entered are not in the correct format or the required fields are empty • The system displays a validation error message box with the following controls															
	<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Name</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>Name is required</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Name textbox</td></tr><tr><td>Video Link</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>Video Link is required</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Video Link textbox</td></tr></table>	Control Name	Control Name	Notes	Name	Textbox	This control will be outline in red to show a validation error	Name is required	Label	This control will be used as a label to indicate a validation error in the Name textbox	Video Link	Textbox	This control will be outline in red to show a validation error	Video Link is required	Label	This control will be used as a label to indicate a validation error in the Video Link textbox
	Control Name	Control Name	Notes													
	Name	Textbox	This control will be outline in red to show a validation error													
Name is required	Label	This control will be used as a label to indicate a validation error in the Name textbox														
Video Link	Textbox	This control will be outline in red to show a validation error														
Video Link is required	Label	This control will be used as a label to indicate a validation error in the Video Link textbox														
• Return to step 3																
CONCLUSION:	This use case concludes when a video has been added on the system															
POST-CONDITION:	A new user has been added to the Video table in the database															
BUSINESS RULES	• Only the administrator has access to this function.															
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• The system is offline, and the system administrator will not be able to add a new video.															
ASSUMPTIONS:	• None															
OPEN ISSUES:	• None															

Table 134 - 15.2 Search Video

VIDEO SUBSYSTEM			
AUTHORS (s): Thenjiwe Ntsonda		DATE: 18/04/2022	
VERSION: 1.0		LAST REVIEW DATE: 23/08/2022	
USE CASE NAME:	Search Video	USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒	
USE CASE ID:	15.2		
PRIORITY:	Medium		
SOURCE:	ByteXcom Requirements List		
PRIMARY BUSINESS ACTOR	• Administrator		
PRIMARY SYSTEM ACTOR	• None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	This use case describes the process in which the administrator searches for videos on the system. The system responds by displaying the information relating to the search query.		
PRE-CONDITION:	• The administrator must be logged onto the system.		
TRIGGER:	This use case is called when the administrator wants to search for a video on the system.		
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE	
		MANUAL ACTION	AUTOMATED ACTION
	Step 1: The administrator wishes to search for a video on the system.		
	Step 2: The administrator navigates to the Video Screen under the Video tab on the side bar		Step 3: The system retrieves following information using an SQL READ QUERY From the Video table:

			Details to be read	Attributes in the table	
				Notes	
			Name	Name	
			Video Link	VideoUrl	
And displays the Video Screen with the following controls:					
			Control Name	Control Type	Notes
			Videos List	Heading	Page main heading
			Question Mark Icon	Button	When clicked, this control will open a pdf which contains the help document
			Create New Video	Button	When clicked, this control will direct the user to the Add video page
			Search bar	Textbox	This control will be used when the user inserts the search parameters
			Search icon	Button	This control will be used to activate the search
			Results table	Table	This control will be used to display all the videos that exist on the system
			Name	Label	This control will be used as a heading to display the Name for all videos
			Video Link	Label	This control will be used as a heading to display the Video Link for all videos
			Action	Table Heading	This control will be used as a heading to display the actions the administrator can perform with each entry – Edit and Delete

			Action - Update	Button	This control will be placed in a row, with the delete button, under the "Action" heading for each entry in the Video table. The button will direct the user to the Update Video screen
			Action - Delete	Button	This control will be placed in a row, with the delete button, under the "Action" heading for each entry in the Video table. The button will direct the user to the Delete Video Details screen
			Showing "" of "" entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of entries in the database
			"Previous" and "Next"	Label and button	This control is used to navigate through the entry pages and to indicate on which page the administrator is currently on

	Step 4: The administrator enters their search query into the search textbox.		Step 5: The system searches the Video table and displays the video(s) matching the search query in the Video Screen [ALT]
	ALT Step 5: The search query entered does not match any records existing in the Video entity. The system displays a “No Records Found” message in the Video Screen and on the results table.		
CONCLUSION:	This use case concludes when the required video is displayed on the screen.		
POST-CONDITION:	The administrator selects the required video from the displayed results.		
BUSINESS RULES	Only the administrator has access to this function.		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

Table 135 - 15.3 Update Video Details

VIDEO SUBSYSTEM				
AUTHORS (s): Thenjiwe Ntsonda		DATE: 18/04/2022		
VERSION: 1.0		LAST REVIEW DATE: 23/08/2022		
USE CASE NAME:	Update Video		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒	
USE CASE ID:	15.3			
PRIORITY:	Medium			
SOURCE:	ByteXcom Requirements List			
PRIMARY BUSINESS ACTOR	• Administrator			
PRIMARY SYSTEM ACTOR	• None			
OTHER PARTICIPATING ACTORS:	• None			
OTHER INTERESTED STAKEHOLDERS:	• None			
DESCRIPTION:	This use case describes the process in which the administrator wishes to update. The use case begins when the administrator updates the video. The administrator will update all the pertinent information regarding the video and save it to the business database.			
PRE-CONDITION:	• The administrator must be logged onto the system.			
TRIGGER:	The use case is triggered when the administrator wants to update a video.			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE		
		MANUAL ACTION	AUTOMATED ACTION	
	Step 1: The administrator wishes to update a video in the video catalogue.		Step 2: The system invokes use case 15.2 Search Video	
	Step 2: The administrator clicks on the Edit button		Step 3: The system retrieves the selected video’s information from the following attributes and tables using a	

method in a .NET Controller and an SQL READ Query:

From the **Video** table:

Details to be read	Attributes in the table
Name	Name
Video Link	VideoUrl

And displays the **Update Video Screen** with the following controls:

Control Name	Control Type	Notes
Update Video	Heading	Page main heading
Video information	Heading	Page subheading
Name	Label	This control will be used as a label to indicate the Name attribute of the video record
Name	Textbox	This control will be used by the administrator to update a name of the video
Video Link	Label	This control will be used as a label to indicate the Video Link attribute of the video record
Video Link	Textbox	This control will be used by the administrator to update a video link of the video
Update	Button	This control will be used to confirm the addition of the video
Cancel	Button	This control will be used to cancel the edit of the video

Note: All controls are enabled by default unless stated otherwise.

	Step 4: The system administrator updates the video information and clicks on the Update button [ALT]		Step 5: The system captures and validates the entered details against the Video table [ALT] <table><tr><th>Details to be read</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Name</td><td>Name</td><td>Required, Unique,</td></tr><tr><td>Video Link</td><td>VideoUrl</td><td>Required, Unique</td></tr></table>	Details to be read	Attributes in the table	Notes	Name	Name	Required, Unique,	Video Link	VideoUrl	Required, Unique					
	Details to be read	Attributes in the table	Notes														
	Name	Name	Required, Unique,														
	Video Link	VideoUrl	Required, Unique														
		Step 6: The system uses an SQL READ query to check if the video already exists in the Video table using the following attributes [ALT] <table><tr><th>Details to be updated</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Name</td><td>Name</td><td></td></tr><tr><td>Video Link</td><td>VideoUrl</td><td></td></tr></table>	Details to be updated	Attributes in the table	Notes	Name	Name		Video Link	VideoUrl							
Details to be updated	Attributes in the table	Notes															
Name	Name																
Video Link	VideoUrl																
		Step 7: The system displays a message box with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Confirmation required</td><td>Label</td><td>This control is used to display the type of message box displayed</td></tr><tr><td>Confirm updated video details?</td><td>Label</td><td>This control is used to request confirmation on the updated video details</td></tr><tr><td>Confirm</td><td>Button</td><td>This control is used to confirm the updating of the video details</td></tr><tr><td>Cancel</td><td>Button</td><td>This control is used to close the message box for the user to edit the video details</td></tr></table>	Control Name	Control Type	Notes	Confirmation required	Label	This control is used to display the type of message box displayed	Confirm updated video details?	Label	This control is used to request confirmation on the updated video details	Confirm	Button	This control is used to confirm the updating of the video details	Cancel	Button	This control is used to close the message box for the user to edit the video details
Control Name	Control Type	Notes															
Confirmation required	Label	This control is used to display the type of message box displayed															
Confirm updated video details?	Label	This control is used to request confirmation on the updated video details															
Confirm	Button	This control is used to confirm the updating of the video details															
Cancel	Button	This control is used to close the message box for the user to edit the video details															

	Step 8: The system administrator clicks on the Confirm button [ALT]		Step 9: The system saves the information into the Video table using an UPDATE SQL query: <table><tr><th>Details to be saved</th><th>Attributes in the table</th><th>Notes</th></tr><tr><td>Name</td><td>Name</td><td>Required, Unique,</td></tr><tr><td>Video Link</td><td>VideoUrl</td><td>Required, Unique</td></tr></table>	Details to be saved	Attributes in the table	Notes	Name	Name	Required, Unique,	Video Link	VideoUrl	Required, Unique						
	Details to be saved	Attributes in the table	Notes															
Name	Name	Required, Unique,																
Video Link	VideoUrl	Required, Unique																
			Step 10: The system redirects the system administrator to the Video Screen and displays a message box with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Checkbox</td><td>Icon</td><td>This control is used to indicate a successful update</td></tr><tr><td>"Video Created Successfully"</td><td>Label</td><td>This control is used to indicate a successful update</td></tr></table>	Control Name	Control Name	Notes	Checkbox	Icon	This control is used to indicate a successful update	"Video Created Successfully"	Label	This control is used to indicate a successful update						
Control Name	Control Name	Notes																
Checkbox	Icon	This control is used to indicate a successful update																
"Video Created Successfully"	Label	This control is used to indicate a successful update																
ALTERNATE COURSES:	ALT Step 4: The administrator does not want to edit the video anymore - The administrator clicks on the cancel button - The use case is terminated																	
	ALT Step 5: The details entered are not in the correct format or the required fields are empty The system displays a validation error message box with the following controls <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Name</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>Name is required</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Name textbox</td></tr><tr><td>Video Link</td><td>Textbox</td><td>This control will be outline in red to show a validation error</td></tr><tr><td>Video Link is required</td><td>Label</td><td>This control will be used as a label to indicate a validation error in the Video Link textbox</td></tr></table>			Control Name	Control Name	Notes	Name	Textbox	This control will be outline in red to show a validation error	Name is required	Label	This control will be used as a label to indicate a validation error in the Name textbox	Video Link	Textbox	This control will be outline in red to show a validation error	Video Link is required	Label	This control will be used as a label to indicate a validation error in the Video Link textbox
	Control Name	Control Name	Notes															
	Name	Textbox	This control will be outline in red to show a validation error															
Name is required	Label	This control will be used as a label to indicate a validation error in the Name textbox																
Video Link	Textbox	This control will be outline in red to show a validation error																
Video Link is required	Label	This control will be used as a label to indicate a validation error in the Video Link textbox																
ALT Step 6: The Video already exists on the system - The system displays an error modal dialogue with the following controls																		
	<table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Video already exists!</td><td>Label</td><td>This control is used to inform the administrator that the video already exists</td></tr></table>	Control Name	Control Name	Notes	Video already exists!	Label	This control is used to inform the administrator that the video already exists											
Control Name	Control Name	Notes																
Video already exists!	Label	This control is used to inform the administrator that the video already exists																

	Okay	Button	This control is used to cancel the creation of the video
	- The administrator clicks the “Okay” button - The use case is terminated		
	ALT Step 8: The administrator clicks on the cancel button The use case is terminated		
CONCLUSION:	This use case concludes when a video has been added on the system		
POST-CONDITION:	A video has been updated in the Video table in the database		
BUSINESS RULES	Only the administrator has access to this function.		
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	None		
ASSUMPTIONS:	None		
OPEN ISSUES:	None		

Table 136 - 15.4 Delete Video

VIDEO SUBSYSTEM				
AUTHORS (s): Thenjiwe Ntsonda		DATE: 18/04/2022		
VERSION: 1.0		LAST REVIEW DATE: 23/08/2022		
USE CASE NAME:	Delete Video		USE CASE TYPE Business Requirements: ☐ System Analysis: ☐ System Design: ☒	
USE CASE ID:	15.4			
PRIORITY:	Medium			
SOURCE:	ByteXcom Requirements List			
PRIMARY BUSINESS ACTOR	• Administrator			
PRIMARY SYSTEM ACTOR	• None			
OTHER PARTICIPATING ACTORS:	• None			
OTHER INTERESTED STAKEHOLDERS:	• None			
DESCRIPTION:	This use case describes the process in which the administrator wishes to delete a video on the system. The use case begins when the administrator selects the video they wish to delete. This product will then be removed from the system.			
PRE-CONDITION:	• The administrator must be logged onto the system.			
TRIGGER:	The use case is triggered when the administrator wants to delete a video.			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE		
		MANUAL ACTION	AUTOMATED ACTION	
		Step 1: The administrator wishes to delete a video		Step 2: The system invokes use case 15.2 Search Video
			Step 3: The system retrieves the video information from the following attributes and tables using a method in a .NET Controller and an SQL READ Query: From the Video table:	
			<table><tr><td>Details to be read</td><td>Attributes in the table Notes</td></tr><tr><td>Name</td><td>Name</td></tr></table>	Details to be read
Details to be read	Attributes in the table Notes			
Name	Name			

		<table><tr><td>Video Link</td><td>VideoUrl</td></tr></table> And displays the Delete Video Screen with the following controls: <table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Delete Video</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Name</td><td>Label</td><td>This control will be used as a label to indicate the Name attribute of the video record</td></tr><tr><td>Name</td><td>Label</td><td>This control will be used by the administrator to display the name of the video</td></tr><tr><td>Video Link</td><td></td><td>This control will be used as a label to indicate the Video Link attribute of the video record</td></tr><tr><td>Video Link</td><td></td><td>This control will be used by the administrator to display the Video Link of the video</td></tr><tr><td>Delete</td><td>Button</td><td>This control will be used to confirm the deletion of the video</td></tr><tr><td>Cancel</td><td>Button</td><td>This control will be used to cancel the deletion of the video</td></tr></table>	Video Link	VideoUrl	Control Name	Control Type	Notes	Delete Video	Heading	Page main heading	Name	Label	This control will be used as a label to indicate the Name attribute of the video record	Name	Label	This control will be used by the administrator to display the name of the video	Video Link		This control will be used as a label to indicate the Video Link attribute of the video record	Video Link		This control will be used by the administrator to display the Video Link of the video	Delete	Button	This control will be used to confirm the deletion of the video	Cancel	Button	This control will be used to cancel the deletion of the video
	Video Link	VideoUrl																										
	Control Name	Control Type	Notes																									
	Delete Video	Heading	Page main heading																									
	Name	Label	This control will be used as a label to indicate the Name attribute of the video record																									
	Name	Label	This control will be used by the administrator to display the name of the video																									
	Video Link		This control will be used as a label to indicate the Video Link attribute of the video record																									
	Video Link		This control will be used by the administrator to display the Video Link of the video																									
	Delete	Button	This control will be used to confirm the deletion of the video																									
	Cancel	Button	This control will be used to cancel the deletion of the video																									
Step 4: The administrator clicks on the Delete button	Step 5: The system displays a message box with the following controls:	<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>Confirmation required</td><td>Label</td><td>This control is used to display the</td></tr></table>	Control Name	Control Type	Notes	Confirmation required	Label	This control is used to display the																				
Control Name	Control Type	Notes																										
Confirmation required	Label	This control is used to display the																										

					type of message box displayed									
			Are you sure you want to delete this video?	Label	This control is used to request confirmation of the deletion of the video									
			Confirm	Button	This control is used to confirm the deletion of the video details									
			Cancel	Button	This control is used to close the message box deletion of the video									
	Step 6: The system administrator clicks on the Confirm button [ALT]		Step 7: The system deletes the selected video from the Video entity in the database using a method in a .NET controller and an SQL DELETE query. <table><tr><th>Details to be deleted</th><th>Attributes in the table</th></tr><tr><td>Videoid</td><td>Videoid</td></tr><tr><td>Name</td><td>Name</td></tr></table>			Details to be deleted	Attributes in the table	Videoid	Videoid	Name	Name			
Details to be deleted	Attributes in the table													
Videoid	Videoid													
Name	Name													
			Step 8: The system redirects the system administrator to the Video screen and displays a message box with the following controls: <table><tr><th>Control Name</th><th>Control Name</th><th>Notes</th></tr><tr><td>Checkbox</td><td>Icon</td><td>This control is used to indicate a successful deletion</td></tr><tr><td>"Video Deleted Successfully"</td><td>Label</td><td>This control is used to indicate a successful deletion</td></tr></table>			Control Name	Control Name	Notes	Checkbox	Icon	This control is used to indicate a successful deletion	"Video Deleted Successfully"	Label	This control is used to indicate a successful deletion
Control Name	Control Name	Notes												
Checkbox	Icon	This control is used to indicate a successful deletion												
"Video Deleted Successfully"	Label	This control is used to indicate a successful deletion												
ALTERNATE COURSES:	ALT Step 5: The administrator does not want to delete the video anymore - The administrator clicks on the cancel button - The use case is terminated													
CONCLUSION:	This use case concludes when a video has been deleted on the system													
POST-CONDITION:	The video has been removed from the system database.													
BUSINESS RULES	Only the administrator has access to this function.													
IMPLEMENTATION CONSTRAINTS	None													

AND SPECIFICATIONS	
ASSUMPTIONS:	None
OPEN ISSUES:	None

Table 137 - 15.5 View Video

VIDEO SUBSYSTEM			
AUTHORS (s): Thenjiwe Ntsonda		DATE: 08/06/2022	
VERSION: 1.0		LAST REVIEW DATE: 23/08/2022	
USE CASE NAME:	View Video Details		USE CASE TYPE Business Requirements: ☐ System Analysis: System Design:
USE CASE ID:	15.5		
PRIORITY:	High		
SOURCE:	ByteXcom Requirements List		
PRIMARY BUSINESS ACTOR	• User		
PRIMARY SYSTEM ACTOR	• None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	This use case describes the process in which the user wishes to view the videos on the system. The use case begins when the user wishes to view videos on the system. The user will select the relevant inventory item and view its details.		
PRE-CONDITION:	• None		
TRIGGER:	The use case is triggered when the user wishes to view the video(s) on the system		
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE	
		MANUAL ACTION	AUTOMATED ACTION
	Step 1: The user clicks on the About Us nav link		Step 2: The system retrieves the following attributes from the following tables, using a method in a .NET controller and an SQL READ query: From the Video Table
		Details to be read	Attributes in the table

			<table><tr><td>Name</td><td>Name</td></tr><tr><td>Video Link</td><td>VideoUrl</td></tr></table>	Name	Name	Video Link	VideoUrl																										
			Name	Name																													
			Video Link	VideoUrl																													
			and displays the About Us Screen which with the following controls:																														
			<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr><tr><td>About Us</td><td>Heading</td><td>Page main heading</td></tr><tr><td>Company Logo</td><td>Label</td><td>This control will be used as a label to indicate the company logo which is retrieved from the www/root folder</td></tr><tr><td>Company Description</td><td>Label</td><td>This control will be used to display the company's description name</td></tr><tr><td>Our Team</td><td>Label</td><td>This control will be used as a label to display the second page heading</td></tr><tr><td>The team's images</td><td>Images</td><td>This control will be used to display the team's images which are retrieved from the www/root folder</td></tr><tr><td>Team's Names</td><td>Label</td><td>This control will be used as a label to indicate the names of the team</td></tr><tr><td>Team's Description</td><td>Label</td><td>This control will be used as a label to indicate the descriptions of the team</td></tr><tr><td>Why Natuurlik</td><td>Label</td><td>This control will be used as a label to display the third page heading</td></tr><tr><td>Video</td><td>Video</td><td>This control will be used to display the videos added to the system which are</td></tr></table>	Control Name	Control Type	Notes	About Us	Heading	Page main heading	Company Logo	Label	This control will be used as a label to indicate the company logo which is retrieved from the www/root folder	Company Description	Label	This control will be used to display the company's description name	Our Team	Label	This control will be used as a label to display the second page heading	The team's images	Images	This control will be used to display the team's images which are retrieved from the www/root folder	Team's Names	Label	This control will be used as a label to indicate the names of the team	Team's Description	Label	This control will be used as a label to indicate the descriptions of the team	Why Natuurlik	Label	This control will be used as a label to display the third page heading	Video	Video	This control will be used to display the videos added to the system which are
			Control Name	Control Type	Notes																												
			About Us	Heading	Page main heading																												
			Company Logo	Label	This control will be used as a label to indicate the company logo which is retrieved from the www/root folder																												
			Company Description	Label	This control will be used to display the company's description name																												
			Our Team	Label	This control will be used as a label to display the second page heading																												
The team's images	Images	This control will be used to display the team's images which are retrieved from the www/root folder																															
Team's Names	Label	This control will be used as a label to indicate the names of the team																															
Team's Description	Label	This control will be used as a label to indicate the descriptions of the team																															
Why Natuurlik	Label	This control will be used as a label to display the third page heading																															
Video	Video	This control will be used to display the videos added to the system which are																															

			<table><tr><td></td><td></td><td>retrieved from the Video Link attribute</td></tr><tr><td>Video Title</td><td>Label</td><td>This control will be used to display the title of all videos which are on the system and are retrieved from the Title attribute</td></tr></table>			retrieved from the Video Link attribute	Video Title	Label	This control will be used to display the title of all videos which are on the system and are retrieved from the Title attribute	
			retrieved from the Video Link attribute							
	Video Title	Label	This control will be used to display the title of all videos which are on the system and are retrieved from the Title attribute							
Step 3: The user clicks on the video they wish to play		Step 4: The system plays the video embedded from YouTube using the following controls: <table><tr><th>Control Name</th><th>Notes</th></tr><tr><td>Frameborder</td><td>This control is used to determine the width of the video border</td></tr><tr><td>Autoplay</td><td>This control is used to allow the video to auto play when the page loads</td></tr><tr><td>Allowfullscreen</td><td>This control is used to allow view their selected video on full screen</td></tr></table> Note: These controls are configured on YouTube.	Control Name	Notes	Frameborder	This control is used to determine the width of the video border	Autoplay	This control is used to allow the video to auto play when the page loads	Allowfullscreen	This control is used to allow view their selected video on full screen
Control Name	Notes									
Frameborder	This control is used to determine the width of the video border									
Autoplay	This control is used to allow the video to auto play when the page loads									
Allowfullscreen	This control is used to allow view their selected video on full screen									
ALTERNATE COURSES:	• None									
CONCLUSION:	This use case concludes when the inventory procured details are displayed on the system									
POST-CONDITION:	The selected video is played on the system.									
BUSINESS RULES	• None									
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• None									
ASSUMPTIONS:	• The user will always choose to play the selected video									
OPEN ISSUES:	• None									

2.16. Help Subsystem

Table 138 - 16.1 View Help Document

HELP SUBSYSTEM				
AUTHORS (s): Nomusa Vumisa		DATE: 03/09/2022		
VERSION: 1.1		LAST REVIEW DATE: 03/09/2022		
USE CASE NAME:	View Help Document		USE CASE TYPE	
USE CASE ID:	16.1		Business Requirements: ☐	
PRIORITY:	Medium		System Analysis: ☐	
SOURCE:	ByteXpress Requirements List		System Design: ☒	
PRIMARY BUSINESS ACTOR	Administrator			
PRIMARY SYSTEM ACTOR	None			
OTHER PARTICIPATING ACTORS:	None			
OTHER INTERESTED STAKEHOLDERS:	None			
DESCRIPTION:	This use case describes the process where the administrator wants to view a help document on the system. The system responds by displaying the results matching the search query and the administrator selects their desired document.			
PRE-CONDITION:	The administrator must be logged onto the system.			
TRIGGER:	This use case is called when the administrator wants to view a help document on the system.			
TYPICAL COURSE OF EVENTS:	ACTOR ACTION	SYSTEM RESPONSE		
		MANUAL ACTION	AUTOMATED ACTION	
	Step 1: The administrator clicks on the Help navigation link on the admin side navigation bar		Step 2: The system displays the Help Documents Screen with the following controls:	
			Control Name	Control Type
		Help Documents	Heading	Page main heading

			Show “” entries	Label and drop down	This control will be used to set the number of entries the administrator wants to display on a page
			Search bar	Textbox	This control will be used to enter the search query
			Results table	Table	This control will be used to display all the help documents that exist on the system
			Title	Table Heading	This control will be used as a heading to display the title of the help document
			Action	Table Heading	This control will be used as a heading to display the actions the administrator can perform with each entry – View
			Action – View	Button	This control will be placed in a row, under the “Action” heading for each entry in table. The button will open the help document
			Showing “” to “” of “” entries	Label	This control will be used to display the number of entries displayed per page as well as the total number of documents
			“Previous” and “Next”	Label and button	This control is used to navigate through the entry pages and to indicate on which page the

					administrator is currently on				
	Step 3: The administrator enters the search query into the search textbox		Step 4: The system searches the table rows in the data table and displays the document(s) matching the search query [ALT]						
	Step 5: The administrator clicks on the View button		Step 6: The system retrieves the help document from the following file using links on the razor page:						
			<table><tr><th>Details to be read</th><th>Folder</th></tr><tr><td>Help document</td><td>wwwroot/helpDocs</td></tr></table>			Details to be read	Folder	Help document	wwwroot/helpDocs
	Details to be read	Folder							
	Help document	wwwroot/helpDocs							
		Step 7: The system displays the help document							
ALT Step 5: The search query entered does not match any existing records. • The system displays a “No Records Found” message in the results table. • Return to step 4 or terminate the use case									
CONCLUSION:	This use case concludes when the desired order details are displayed on the system.								
POST-CONDITION:	The Order and User entity was searched for and an entry that corresponds to the search query or parameter entered.								
BUSINESS RULES	• None								
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• None								
ASSUMPTIONS:	• None								
OPEN ISSUES:	• None								

Conclusion

This section concludes the Technical Use Case Narratives for the ByteXecom system. These narratives outlined the actors who will be interacting with the system for each of the use cases, the steps to be carried out by the information system in order to satisfy business requirements and the identified technologies to be used in facilitating the various tasks the system is required to perform. The design specifications and constraints identified were also provided where applicable as well as required pre-conditions and expected post-conditions.

3. Iteration Conclusion

In conclusion the compiled iteration detailed the final documentation for the design and development of the ByteXecom E-commerce system. The documents contained the updated requirements lists, complexity matrix, logical and technical narratives, technical primitive, UML activity, sequence and state diagrams as well as the test cases and screen designs for our use cases.