ByteXpress - Team 9
ByteXecom E-commerce System

Aphiwe Shozi
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 Logistsics.

Client
Jannes Janse van Rensburg

Email
jan2rens@gmail.com

Cellphone number
072 881 0004

Table of Contents

Table of Tables

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. Logical Use Case Narratives

Introduction

In this section, the ByteXpress team compiled a comprehensive list of the logical flow of their use cases through the logical narratives. These narratives include all of the business processes carried out by the system.

2.1. User Management Subsystem

Table 1- 1.1 Login

<table>
<tr><th colspan="4">USER MANAGEMENT SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Nomusa Vumisa</td><td colspan="2">DATE: 20/08/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 20/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>Login</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>1.1</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• User</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes an event where the user wants to login to the system. The user enters their login details which the system validates. The use case concludes once validation is successful, and the user has been granted access to the system.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The user must be registered on the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use case is called when the user wants to login</td></tr>
<tr><td rowspan="3">TYPICAL COURSE</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The user wants to login</td><td colspan="2">Step 2: The system requests the user to enter their login details which include the following attributes from the User table:
• EmailAddress
• Password</td></tr>
<tr><td>Step 3: The user enters their details</td><td colspan="2">Step 4: The system captures and validates the details</td></tr>
</table>

	[ALT]	[ALT]
		Step 5: The system retrieves the user role details using the following attributes in the **UserRole** table • Role_Id and generates an authentication cookie
		Step 6: The system grants the user access
ALTERNATE COURSES:	ALT **Step 3A**: The user forgot their password • Invoke use case **1.5 Forgot Password**	
	ALT **Step 3B**: The user does not have an account on the system • Invoke use case **1.6 Register Customer**	
	ALT **Step 4**: The details entered are not in the correct format or the required fields are empty • Return to Step 3	
CONCLUSION:	This use case concludes when the user is logged in	
POST-CONDITION:	The user will have access to specific sections of the system based on their user role	
BUSINESS RULES	• None	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• The user cannot login without an internet connection • All users access the system using the same login page • All uses have access to this function	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 2- 1.2 Logout

USER MANAGEMENT SUBSYSTEM			
AUTHORS (s): Nomusa Vumisa		**DATE: 20/08/2022**	
VERSION: 1.0		**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 List		System Design: ☐
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 logout of the system. The user requests to logout and the system responds by logging them out and returns them to the login page.		
PRE-CONDITION:	The user must be logged on the system		
TRIGGER:	The use case is called when the user wants to logout		
TYPICAL COURSE	Actor Action	System Response	
	Step 1: The user wants to logout	**Step 2**: The system removes the authentication cookie and logs the user out of the system	
		Step 3: The system returns the user to the home page	
ALTERNATE COURSES:	• None		
CONCLUSION:	This use case concludes when the user is logged out of the system		
POST-CONDITION:	The user can only access areas of the system that do not require them to be logged in		
BUSINESS RULES	• None		

IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• The user cannot logout without an internet connection • The user should be able to logout from any page • All uses have access to this function
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 3- 1.3 View Profile Details

<table>
<tr><th colspan="4">USER MANAGEMENT SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Nomusa Vumisa</td><td colspan="2">DATE: 20/08/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 20/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>View Profile Details</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>1.3</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• User</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes an event where the user wants to view their profile on the system. The user requests to view their details and the system responds by retrieving the details and displaying them on the screen</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The user must be logged on the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use case is called when the user wants to view their details</td></tr>
<tr><td rowspan="2">TYPICAL COURSE</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The user wants to view their details</td><td colspan="2">Step 2: The system retrieves the following details:

From the User table.
• FirstName
• Surname
• EmailAddress
• PhoneNumber
• StreetAddress

From the Suburb Table
• SuburbName</td></tr>
</table>

		From the **City** Table • CityName From the **Province** Table • ProvinceName From the **Country** Table • CountryName
		Step 3: The system displays the user details
ALTERNATE COURSES:	• None	
CONCLUSION:	This use case concludes when the user details are displayed	
POST-CONDITION:	The user details are displayed	
BUSINESS RULES	• None	
IMPLEMENTATION CONTRAINTS 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

<table>
<tr><th colspan="4">USER MANAGEMENT SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Kyle van Eeden</td><td colspan="2">DATE: 07/04/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 07/04/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Reset Password</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">1.4</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• User</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process where a registered system user wishes to update the password of their account. The use case is initiated once the user selects the option to reset their password. The system will request the user to provide their current password, the new password to be used along with the reconfirmation of the new password. The system will validate the captured details to ensure that it adheres to the minimum password complexity criteria and that the confirmation password matches the new password specified. The use case concludes once the updated password is hashed and saved to the system for the specific user account and the user is notified that the password has been successfully reset.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The user must already have an account registered on the system.
• The user must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use case is triggered once the user requests to update their current password which is associated with their account.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The user requests to reset their password.</td><td colspan="2">Step 2: The system prompts the user to enter their current password, the new password as well as reconfirmation of the new password to be used.</td></tr>
<tr><td>Step 3: The user provides the requested details and selects</td><td colspan="2">Step 4: The system captures and validates the information provided</td></tr>
</table>

	the option to update their password. **[ALT]**	against the current password stored in the **User** entity. **[ALT]**
		Step 5: The system updates the user's password by hashing the new password provided and storing the hashed password details in the **User** entity.
		Step 6: The system displays a message on the interface to confirm that the user's password has been updated successfully.
ALTERNATE COURSES:	**ALT** Step 3: The user cancels the 'Reset Password' operation: • The use case is terminated	
	ALT Step 4 a: The system fails to validate the information provided against the details in the **User** table because of an incorrect current password provided: • Return to Step 3.	
	ALT Step 4 b: Validation fails since the new password does not meet the minimum requirements: • The system displays a validation error message to inform the user of what the minimum password requirements are. • Return to Step 3.	
	ALT Step 4 c: Validation fails since the new password provided does not match the reconfirm new password entered: • The system displays a validation error message to inform the user that the new password must match the confirm password exactly • Return to Step 3.	
CONCLUSION:	This use case concludes when the system informs the user that their password has been successfully reset/updated.	
POST-CONDITION:	• The user account's password is updated in the **User** entity.	
BUSINESS RULES	• A user can only update the password of the account which they are the owner of. • The updated password provided must be at least six characters long, and must contain at least one uppercase and lower case character, a digit, and a non-alphanumeric character for the password to be successfully updated. • The user must reconfirm the updated password which they have provided. • All passwords stored in the database are to be encrypted by using hashing algorithms for security reasons.	

IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• The system is offline or the user does not have an internet connection, and the user is unable to reset the password of their Natuurlik account.
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 5- 1.5 Forgot Password

<table>
<tr><th colspan="4">USER MANAGEMENT SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Nomusa Vumisa</td><td colspan="2">DATE: 20/08/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 20/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>Forgot Password</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>1.5</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• User</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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 logins in with their new password and accesses the system.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The user must be registered on the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use case is called when the user forgets their password</td></tr>
<tr><td rowspan="3">TYPICAL COURSE</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The user forgot their password</td><td colspan="2">Step 2: The system requests the user to enter the email address that they used upon registration</td></tr>
<tr><td>Step 3: The user enters their email address</td><td colspan="2">Step 4: The system validates the email address from the User table using the following attribute:

• EmailAddress</td></tr>
</table>

		[ALT]
		Step 5: The system sends a reset password token to the user by email
	Step 6: The user receives the email and accesses the link provided	**Step 7:** The system requests the new details from the user
	Step 8: The user enters their new password and confirms it by entering the password again	**Step 9:** The system captures and validates the entered details [ALT]
		Step 10: The system saves the details provided in the **User** table with the following attribute: • Password
		Step 11: The system informs the user that their password has been reset successfully
ALTERNATE COURSES:	ALT **Step 4A**: The details entered are not in the correct format or the required field is empty • The system displays an error message • Return to Step 3	
	ALT **Step 4B**: The email address does not exist in the system • The system displays an error message • Return to Step 3	
	ALT **Step 9A**: The details entered are not in the correct format or the required fields are empty • The system displays an error message • Return to Step 8	
	ALT **Step 9B**: The new password and confirmed password do not match • The system displays an error message • Return to Step 8	
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 CONTRAINTS 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 Customer

<table>
<tr><th colspan="4">USER MANAGEMENT SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Nomusa Vumisa</td><td colspan="2">DATE: 20/08/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 20/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Register Customer</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">1.6</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Customer</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the event where the customer wishes to register on the system. The customer enters their details, and the system responds by validating the details. Once successful the details are saved on the system and an email confirmation is sent to the user. The use case ends once the user confirms their email and is directed to the login page.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The customer must not exist on the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use case is called when the customer wants to register</td></tr>
<tr><td rowspan="2">TYPICAL COURSE</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The customer wishes to register on the system</td><td colspan="2">Step 2: The system requests the customer to enter/select details for the following attributes:

From the User table.
• FirstName
• Surname
• EmailAddress
• PhoneNumber
• StreetAddress</td></tr>
</table>

		From the **Suburb** Table • SuburbName From the **City** Table • CityName From the **Province** Table • ProvinceName From the **Country** Table • CountryName
	Step 3: The customer enters their details **[ALT]**	**Step 4**: The system captures and validates the details **[ALT]**
		Step 5: The system saves the customer details into the following tables with the following attributes From the **User** table. • ID • FirstName • Surname • EmailAddress • PhoneNumber • StreetAddress • SuburbId
		Step 6: The system sends an email confirmation link to the user
	Step 7: The user receives the email and accesses the link provided	**Step 8:** The system captures the link and updates the following attribute in the **User** table: • EmailConfirmed
		Step 10: The system informs the user that their email address has been confirmed successfully
ALTERNATE COURSES:	ALT **Step 3**: The user has an account on the system Invoke use case **1.1 Login**	

	ALT Step 4: The details entered are not in the correct format or the required fields are empty • The system displays a validation error message • Return to step 3
CONCLUSION:	This use case concludes when the customer has successfully registered on the system
POST-CONDITION:	The customer is added to the user table and now has access to the system
BUSINESS RULES	• None
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• The customer cannot register without an internet connection • Only the customers should use this function
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 7- 1.7 Update Profile Details

<table>
<tr><th colspan="4">USER MANAGEMENT SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Nomusa Vumisa</td><td colspan="2">DATE: 20/08/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 20/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>Update Profile Details</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>1.7</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• User</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the event where a user wishes to edit their profile details. The system will retrieve the user information stored on the system which can be edited. The user then edits their details accordingly. The use case ends when the users' details has been successfully updated on the system.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The user must be logged in</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the user wishes to edit their details</td></tr>
<tr><td rowspan="4">TYPICAL COURSE</td><td colspan="2">Actor Action</td><td>System Response</td></tr>
<tr><td colspan="2">Step 1: The user wishes to edit their details</td><td>Step 2: The system invokes use case 1.3 View Profile Details</td></tr>
<tr><td colspan="2">Step 3: The user edits their details</td><td>Step 4: The system captures and validates the details
[ALT]</td></tr>
<tr><td colspan="2"></td><td>Step 5: The system saves the changes made to the User table with the following attributes:

From the User table.</td></tr>
</table>

		• FirstName • Surname • PhoneNumber • StreetAddress • Suburb ID
		Step 6: The system displays a successful update message
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 message • 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 CONTRAINTS 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

<table>
<tr><td colspan="4">USER MANAGEMENT SUBSYSTEM</td></tr>
<tr><td colspan="2">AUTHORS (s): Nomusa Vumisa</td><td colspan="2">DATE: 13/05/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 13/05/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Update Email Address</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">1.8</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• User</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The user must be logged in</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the user wishes to update their email address.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The user wishes to update their email address.</td><td colspan="2">Step 2: The system retrieves the user’s email address from the User table with the following attribute
• Email</td></tr>
<tr><td>Step 3: The user edits their email address</td><td colspan="2">Step 4: The system captures and validates the email address
[ALT]</td></tr>
</table>

		Step 5: The system reads from the **User** table to ensure that the record being updated does not match any existing records using the following attribute • Email **[ALT]**
		Step 6: The system sends an account verification email to the user
	Step 7: The user receives the email and accesses the link provided	**Step 8:** The system saves the changes made to the **User** table with the following attributes: • Email
		Step 9: The system displays a successful update message
ALTERNATE COURSES:	**ALT Step 4A:** The email address entered is not in the correct format or the required field is empty • The system displays a validation error message • Return to step 3	
	ALT Step 4B: The user attempts to save their details but has not made any changes to their email • The system displays a validation error message • Return to step 3	
	ALT Step 5: The email already exists • 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 CONTRAINTS 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

<table>
<tr><th colspan="4">ADMINISTRATIVE SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Kyle van Eeden</td><td colspan="2">DATE: 10/05/2022</td></tr>
<tr><td colspan="2">VERSION: 1.1</td><td colspan="2">LAST REVIEW DATE: 30/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Register User</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">2.1</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator (PSA)</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• User (PBA)</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process in which the administrator wishes to register a new user account on the system with a particular pre-defined user role assigned which controls what actions the user account is allowed to perform on the system. The use case begins when the administrator requests to add a new user to the system, captures the required information on the user to be registered including the User Role, First Name, Surname, Phone Number, Email Address, Street Address, Country, Province, City and Suburb. Once the new user account details have been successfully added to the system, the system will send an account confirmation email to the registered user’s email address. The use case concludes once the system displays a confirmation message which confirms that the user account has been registered successfully.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.
• The user account must not already exist on the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The administrator wishes to register a new user account on the system,</td></tr>
<tr><td>TYPICAL COURSE</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
</table>

OF EVENTS:	**Step 1**: The administrator wishes to add a new user account to the system and selects the option to view all current users on the system.	**Step 2**: The system displays the following details of all existing users that are defined on the system: • First Name • Surname • Phone Number • Email Address
	Step 3: The administrator selects the option to register a new user account on the system.	**Step 4**: The system prompts the administrator to capture the following required details to register a new user account: • User Role • First Name • Surname • Phone Number • Email • Street Address • Country • Province • City • Suburb A list of selectable countries, provinces, cities, suburbs and user roles are retrieved from the database to capture the required form details. The following details are retrieved from the below listed entities: From the **UserRole** entity: • RoleName From the **Country** entity: • CounryName From the **Province** entity: • ProvinceName

		From the **City** entity: • CityName From the **Suburb** entity: • SuburbName
	Step 5: The administrator captures the required details and selects the option to proceed with creating the user account. **[ALT]**	**Step 6**: The system validates the correctness and completeness of the provided details using the requirements configured for the **User** table. **[ALT]**
		Step 7: The system validates that there is not already an existing user account in the **User** table with the **Email** provided in **Step 5**. **[ALT]**
		Step 8: The system creates the following new user account details in the **User** entity: • User ID • First Name • Surname • Email • Phone Number • Street Address • Country • Province • City • Suburb • Email Confirmed (Set to “False” upon creating the account) • Password (Hashed) The system creates the following details in the **UserRole** table to associate the necessary permissions based on the selected role for the user account: • **User ID**

		• **Role ID**
		Step 9: The system retrieves the following details from the **User** entity: • EmailAddress The system then generates and sends an account verification email to registered user's email address.
		Step 10: The system displays an alert notification which informs the administrator that the user account has been registered on the system successfully.
	Step 11: The newly registered user accesses their email address used for account registration and selects the option to confirm their account.	**Step 12:** The system updates the user's account to a confirmed state in the **User** table using the following attribute: • **Email Confirmed** (Updated to "True"
		Step 13: The system displays a
ALTERNATE COURSES:	**ALT** Step 5: The administrator no longer wishes to add a new user account and clicks the option provided for cancelling the operation: • The use case is terminated.	
	ALT Step 6 a: The system fails to validate the required fields which have not been provided with any information: • Display detailed messages pointing out the required fields that raised the validation errors. • Return to **Step 5**.	
	ALT Step 6 b: The system fails to validate the correctness and/or format of input provided in any of the fields on the screen: • Display a relevant validation error message for all the fields with **incorrect input**. • Return to **Step 5**.	
	ALT Step 6 c: The password provided does not meet the minimum criteria: • The system displays a validation error message to inform the administrator that the password provided did not meet the minimum requirements. • Return to **Step 5**.	

	ALT Step 6 d: The confirm password field's value does not match the password provided: • The system displays a validation error message which informs the administrator that the password and confirm password fields must have an exact match. • Return to **Step 5**.
	ALT Step 7: The system matches the email captured with an already existing user account in the **User** table. • The system will inform the administrator that a user with that Email Address has already been registered on the system. • The use case is terminated.
CONCLUSION:	This use case concludes when the newly registered user account is verified via email and the system informs the user that their account has been successfully confirmed.
POST-CONDITION:	• A new account's details is created and stored in the **User** entity. • The details of the new account along its specified role is stored in the **UserRole** entity. • The user is able to log into their account after verifying their email address.
BUSINESS RULES	• Only the administrator(s) should be able to see an option to specify a user role when registering a new user on the system. • Email addresses are to be used as a user's username and to uniquely identify users on the system. • The user to be registered must not already have an existing account on the system. • All passwords stored in the database should be hashed for security reasons.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• The email address need be unique for each registered account on the system. • A default password is to be used when registering a new user account on the system.
ASSUMPTIONS:	• The registered user will make use of the 'Update Password' functionality to change their password once their account has been confirmed/verified.
OPEN ISSUES:	• None

Table 10 -2.2 Search User

<table>
<tr><th colspan="4">ADMINISTRATIVE SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Kyle van Eeden</td><td colspan="2">DATE: 10/05/2022</td></tr>
<tr><td colspan="2">VERSION: 1.1</td><td colspan="2">LAST REVIEW DATE: 30/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>Search User</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>2.2</td></tr>
<tr><td>PRIORITY:</td><td>Medium</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process in which the administrator searches for an existing user account on the system. The use case begins with the administrator entering the search criteria details on which they wish to perform the search. The system will retrieve and display user accounts which have matched the search criteria specified and the use case concludes with the system displaying the search results in a table.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the administrator wishes to search for a specific user account on the system.</td></tr>
<tr><td rowspan="2">TYPICAL COURSE
OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wishes to search for an existing user defined on the system and clicks the</td><td colspan="2">Step 2: The system displays the following details for all existing user accounts defined on the system and which is retrieved from the User entity:
• First Name
• Surname
• Phone Number
• Email Address</td></tr>
</table>

	Step 3: The administrator enters the search parameter in the provided search bar field.	**Step 4:** The system searches the following attributes in the **User** entity for which the associated values match the search parameter provided: • First Name • Surname • Email Address • Phone Number
		Step 5: The system displays the search result(s) on the user interface. **[ALT]**
ALTERNATE COURSES:	**ALT Step 5**: The search query entered does not match any existing details on user accounts from the **User** table: • The system indicates that no user accounts' details have matched the search query provided. • The use case is terminated.	
CONCLUSION:	This use case concludes when the search results are displayed on the user interface.	
POST-CONDITION:	All the results that matched the search query specified is retrieved from the **User** entity and displayed on the interface.	
BUSINESS RULES	• The system should clearly indicate on the user interface when no search results were found. • The system administrator should be allowed to search for a user by their first name, surname, email address and phone number.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• Sorting, paging and search functionality is required on the '**Users'** page. • The administrator will not be able to base the search on a specific single attribute.	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 11 - 2.3 Update User Details

<table>
<tr><th colspan="4">ADMINISTRATIVE SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Kyle van Eeden</td><td colspan="2">DATE: 10/05/2022</td></tr>
<tr><td colspan="2">VERSION: 1.1</td><td colspan="2">LAST REVIEW DATE: 30/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>Update User Details</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>2.3</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the event in which the administrator wishes to update an existing user account’s details on the system. The use case begins when the administrator views the current details of a particular user account and updates the details of the user account. The use case concludes when the system confirms that the update has been made successfully and the user’s details in the User table has been updated successfully.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.
• The user account which is to be updated must exist on the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use case is triggered when the administrator updates the existing user account’s details.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wishes to update an existing user account’s details on the system.</td><td colspan="2">Step 2: The system invokes Use Case 2.2 “Search User”.

[ALT]</td></tr>
<tr><td>Step 3: The administrator selects the update option for the relevant user account displayed on the user interface.</td><td colspan="2">Step 4: The system displays the following details retrieved for the user account selected and which is retrieved from the User entity:</td></tr>
</table>

		• **First Name** • **Surname** • **Email Address** • **Phone Number** • **Street Address** • **Country** • **Province** • **City** • **Suburb** • **Is Active**
	Step 5: The administrator provides the updated details and selects the option to update the details of the user account. **[ALT]**	**Step 6**: The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. **[ALT]**
		Step 7: The system prompts the administrator to confirm the update of the user account's details.
	Step 8: The administrator confirms that they wish to proceed with the update. **[ALT]**	**Step 9:** The system saves the updated details in the **User** table.
		Step 10: The system displays an update successful message on the screen to confirm that the user account's details is updated and saved successfully.
ALTERNATE COURSES:	**ALT** Step 2: The user account to be updated by the administrator is displayed on the first page of the '**Users**' screen and no search is needed to be performed: • Proceed to **Step 3**	
	ALT Step 5: The administrator no longer wishes to update the details of the user and clicks the option to cancel the update operation: • Redirect administrator to the 'Users' screen • The use case is terminated	
	ALT Step 6 a: The system fails to validate the information provided due to required fields that have not been provided with a value. • The system displays validation error messages for the fields that were left blank. • Return to **Step 5**.	

	ALT Step 6 b: The system detects input validation errors for the details which has been captured: • The system displays validation error messages for the fields that contained incorrect information. • Return to **Step 5**.
	ALT Step 8: The administrator does not wish to proceed with the update: • The administrator selects the option to cancel the update operation. • Return to **Step 5**.
CONCLUSION:	This use case concludes when the system displays a message that confirms the user's details has been updated successfully.
POST-CONDITION:	• The user's account details are updated and saved in the **User** entity.
BUSINESS RULES	• Only the administrator should be allowed to update the details of another existing user's account. • The administrator should not be allowed to update the password of another user's account defined on the system. • A registered user's role cannot be changed from the moment it has been defined on the system. • The administrator should be allowed to change the email address associated with a user's account.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• The administrator will not be allowed to change the role assigned to a particular user. • The administrator will not be allowed to view or change the password of a particular user's account.
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 12 - 2.4 Delete User

ADMINISTRATIVE SUBSYSTEM		
AUTHORS (s): Kyle van Eeden		**DATE: 10/05/2022**
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:	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 narrative describes the event in which the administrator wishes to delete a user account from the system. Once the administrator has searched for the specific user account to be deleted, the administrator then will then view all the details of the user account in read-only format. The administrator then requests to remove the user account, and the system prompts confirmation of the user account removal, after which the administrator will confirm the removal and the user is removed from the **User** table if the user account has not been associated with an order on the system. This use case concludes once the system confirms that the user has been successfully removed by displaying a confirmation message on the user interface.	
PRE-CONDITION:	• The administrator must be logged into the system. • The user account to be deleted must currently exist on the system. • The user to be removed must not have placed an order on the system.	
TRIGGER:	This use case is called when the administrator wishes to delete an existing user account from the system.	
TYPICAL COURSE OF EVENTS:	Actor Action	System Response
	Step 1: The administrator wishes to delete a specific user from the system.	**Step 2:** The system invokes Use Case 2.2 “Search User”. **[ALT]**

	Step 3: The administrator selects the option to delete a specific user from the system.	**Step 4:** The system displays the '**Delete User**' screen with the following details retrieved for the specific user from the **User** entity: • **First Name** • **Surname** • **Email Address** • **Phone Number** • **Street Address** • **Country** • **Province** • **City** • **Suburb** • **User Role**
	Step 5: The administrator selects the option to remove the user account from the system.	**Step 6:** The system validates whether the account requested for removal is associated with an order by retrieving all orders from the **Order** entity. **[ALT]**
		Step 7: The system prompts the administrator to confirm the removal of the user account.
	Step 8: The administrator confirms that they wish to proceed with deletion of the user account from the system. **[ALT]**	**Step 9:** The system removes all of the details related to the selected user account from the **User** table.
		Step 10: The system informs the administrator of the successful deletion of the user account by displaying a notification on the user interface.
ALTERNATE COURSES:	**ALT** Step 2: The user account to be updated by the administrator is displayed on the first page of the '**Users**' screen and no search is needed to be performed: • Proceed to **Step 3**	
	ALT Step 6: The system detects that the user account requested for deletion is associated with an order in the **Order** entity. • The system displays an error message to the administrator to inform them that the user account cannot be deleted.	

	• Use case is terminated.
	ALT Step 8: The administrator selects the option not to proceed with deletion. • The use case is terminated.
CONCLUSION:	This use case concludes when the administrator is notified of the successful removal of the user account.
POST-CONDITION:	• The system deletes all of the details related to the selected user account from the **User entity**.
BUSINESS RULES	• Only the administrator can delete a user from the system. • A user that has placed an order on the system cannot be deleted.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• The administrator will not be allowed to change any of the user account's details displayed when the particular account has been requested for removal.
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

2.3. Reseller Subsystem

Table 13 - 3.1 Send Payment Reminder

RESELLER SUBSYSTEM			
AUTHORS (s): Nomusa Vumisa		**DATE: 19/07/2022**	
VERSION: 1.1		**LAST REVIEW DATE: 19/07/2022**	
USE CASE NAME:	Send Payment Reminder	**USE CASE TYPE**	
USE CASE ID:	3.1	Business Requirements: ☐	
PRIORITY:	High	System Analysis: ☒	
SOURCE:	ByteXpress Requirements List	System Design: ☐	
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	Actor Action	System Response	
	Step 1: The payment threshold has been reached	**Step 2:** The system reads the following attributes to verify that the payment status is “Payment Outstanding” and if half the due date threshold has been reached From the **Order** table • PaymentStatus • OrderDate From the **PaymentReminder** table • Value **[ALT]**	
		Step 3: The system generates a payment reminder email using the following attributes: From the **User** table	

		• Email • FirstName From the **Order** table • OrderId • OrderTotal • OrderDate • PaymentStatus From the **PaymentReminder** table • Value
		Step 4: The system sends a payment reminder to the reseller
ALTERNATE COURSES:	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 attributes retrieved from the User and Order tables	
BUSINESS RULES	None	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• Only the reseller should be notified. • Reminders must be sent via email.	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 14 - 3.2 Capture Proof of Payment

<table>
<tr><th colspan="4">RESELLER SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Aphiwe Shozi</td><td colspan="2">DATE: 07/20/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 07/20/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>Capture Proof of payment</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>3.2</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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”</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged in.
• The payment status of the order must be “payment outstanding”</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is triggered when the administrator wishes to capture a reseller’s proof of payment on the system.</td></tr>
<tr><td rowspan="5">TYPICAL COURSE</td><td colspan="2">Actor Action</td><td>System Response</td></tr>
<tr><td colspan="2">Step 1: The administrator wishes to capture a payment on the system.</td><td>Step 2: The system requires the administrator to confirm whether the reseller’s payment was recieved and upload proof of payment</td></tr>
<tr><td colspan="2">Step 3: The administrator confirms that the payment was received and uploads proof of payment [ALT]</td><td>Step 4: The system prompts the administrator to confirm the Payment Capture.</td></tr>
<tr><td colspan="2">Step 5: The administrator confirms the Payment capture [ALT]</td><td>Step 6: The system then updates the Order Status attribute in the Order entity to “paid ”.</td></tr>
<tr><td colspan="2"></td><td>Step 7: The system displays a success message to the administrator</td></tr>
<tr><td>ALTERNATE COURSES:</td><td colspan="3">ALT Step 3: The administrator or sales assistant does not wish to capture proof of payment anymore
• Terminate use case.</td></tr>
<tr><td></td><td colspan="3">ALT Step 5: The administrator does not confirm the Payment Capture</td></tr>
</table>

	• Return to Step 3.
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
BUSINESS RULES	• Only resellers can make payments after placing an order • Only the administrator can capture a proof of payment
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• The administrator has a stable internet connection.
OPEN ISSUES:	• None

2.4. Product Subsystem

Table 15- 4.1 Add Product

<table>
<tr><th colspan="4">PRODUCT SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Thenjiwe Ntsonda</td><td colspan="2">DATE: 06/04/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 06/04/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Add Product</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">4.1</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• Customers and Resellers – The newly added product is displayed on the customer and reseller catalogue with different prices as required.</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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 of the pertinent information regarding the product and save it to the business database.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• 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.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use case is triggered when the administrator wants to add a new product that customers and resellers can order.</td></tr>
<tr><td>TYPICAL COURSE</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
</table>

OF EVENTS:	**Step 1**: The administrator wants to add a new product	**Step 2**: The system requests the new product details which have the following attributes: From the **Product** table: • Name • Description • Image • CustomerPrice • ResellerPrice • QuantityOnHand • ThresholdValue • DisplayProduct From the **ProductCategory** table: • Name From the **ProductBrand** table: • Name
	Step 3: The administrator inputs the details. **[ALT]**	**Step 4**: The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. **[ALT]**
		Step 5: The system reads from the Product table to ensure that the record being added does not match any existing records using the following attributes: • Name • Description • Image • CustomerPrice • ResellerPrice • QuantityOnHand • ThresholdValue • DisplayProduct • ProductCategoryName • ProductBrandName **[ALT]**

		Step 6: The system saves the new product details which have the following attributes: • ProductId • Name • Description • Image • CustomerPrice • ResellerPrice • QuantityOnHand • ThresholdValue • DisplayProduct • ProductCategoryName • ProductBrandName
		Step 7: The system displays a success message on the screen to confirm the addition of the new product.
	ALT **Step 3:** The administrator does not want to add a Product anymore • The use case is terminated	
ALTERNATE COURSES:	ALT **Step 4:** The system fails to validate the information provided because of missing fields or incorrectly entered information. • Return to Step 3.	
	ALT **Step 6:** The product already exists on the system • The system displays a validation error message • The use case is terminated	
CONCLUSION:	This use case concludes when a new product is added.	
POST-CONDITION:	• The product details are added to the database. • The product details are displayed on the system for selection by customers and resellers.	
BUSINESS RULES	• Only the administrator can add a new product. • Resellers and customers will see different prices of the added product.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 16- 4.2 Search Product

<table>
<tr><th colspan="4">PRODUCT SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Thenjiwe Ntsonda</td><td colspan="2">DATE: 06/04/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 06/04/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Search Product</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">4.2</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">Medium</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the administrator wants to search for a product on the system.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wants to search for a product.</td><td colspan="2"></td></tr>
<tr><td>Step 2: The administrator enters the search parameter.</td><td colspan="2">Step 3: The system searches the following attributes in the Product entity for a match:

• Name
• Description
• Product Category
• Product Brand</td></tr>
<tr><td></td><td colspan="2">Step 4: The system displays the search results. [ALT]</td></tr>
</table>

ALTERNATE COURSES:	[ALT] **Step 4**: The search query entered does not match any existing instances of products in the system database. • Return to Step 3 or terminate use case
CONCLUSION:	This use case concludes when the information is displayed on the system.
POST-CONDITION:	The **Product** entity was searched for and an entry that corresponds to the search query or parameter entered.
BUSINESS RULES	• None
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 17- 4.3 Update Product

PRODUCT SUBSYSTEM		
AUTHORS (s): Thenjiwe Ntsonda	**DATE: 06/04/2022**	
VERSION: 1.0	**LAST REVIEW DATE: 06/04/2022**	
USE CASE NAME:	Update Product	**USE CASE TYPE**
USE CASE ID:	4.3	Business Requirements: ☐
PRIORITY:	High	System Analysis: ☒
SOURCE:	ByteXpress Requirements List	System Design: ☐
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 customer and reseller catalogue with different prices as required.	
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 of 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 needs to exist before the product is updated.	
TRIGGER:	The use is triggered when the administrator updates the product as an option for customers and resellers to choose from.	
TYPICAL COURSE	Actor Action	System Response
OF EVENTS:	**Step 1**: The administrator wishes to update an existing product's details.	**Step 2:** The system invokes Use Case **4.2 Search Product**.

		Step 3: The system retrieves the desired product details from the following tables: From the **Product** table: • Name • Description • Image • CustomerPrice • ResellerPrice • QuantityOnHand • ThresholdValue • DisplayProduct From the **ProductCategory** table: • Name From the **ProductBrand** table: • Name
	Step 4: The administrator inputs the details required. **[ALT]**	**Step 5**: The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. **[ALT]**
		Step 6: The system prompts the administrator to confirm the update of a product with the details provided.
	Step 7: The administrator confirms the information provided. **[ALT]**	**Step 8:** The system reads from the **Product** table to ensure that the record being updated does not match any existing records using the following attributes: • Name • Description • Image • CustomerPrice • ResellerPrice • QuantityOnHand • ThresholdValue

		• DisplayProduct • ProductCategoryName • ProductBrandName **[ALT]**
		Step 9: The system saves the updated product details which have the following attributes: • Name • Description • Image • CustomerPrice • ResellerPrice • QuantityOnHand • ThresholdValue • DisplayProduct • ProductCategoryName • ProductBrandName
		Step 10: The system displays a success message on the screen to confirm the update of a product.
ALTERNATE COURSES:	ALT **Step 4:** The administrator does not want to add a Product anymore • The use case is terminated	
	ALT **Step 5:** The system fails to validate the information provided because of missing fields or incorrectly entered information. • Return to Step 4	
	ALT **Step 7:** The administrator does not confirm the updating of the product details: • Return to Step 4 or terminate the use case	
	ALT **Step 8:** The product already exists on the system • The system displays a validation error message • The use case is terminated	
CONCLUSION:	This use case concludes when a product is updated.	
POST-CONDITION:	• The product details are updated to the database. • The new product details are displayed on the system for selection by customers and resellers.	
BUSINESS RULES	• Only the administrator can update a product.	

IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 18- 4.4 Delete Product

<table>
<tr><th colspan="4">PRODUCT SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Thenjiwe Ntsonda</td><td colspan="2">DATE: 06/04/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 06/04/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Delete Product</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">4.4</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">Medium</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case narrative describes the process in which the administrator wishes to delete a product from the system. The administrator will search for the required product selected for deletion. The administrator will then confirm the deletion of the product and the use case concludes when the system notifies the administrator of a successful deletion.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the administrator wishes to delete a product.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wishes to delete a product.</td><td colspan="2">Step 2: The system invokes Use Case 4.2 Search Product.</td></tr>
<tr><td>Step 3: The administrator selects the delete option [ALT]</td><td colspan="2">Step 4: The system prompts the administrator to confirm the deletion of the product.</td></tr>
<tr><td>Step 5: The administrator confirms the deletion of the product from the system [ALT]</td><td colspan="2">Step 6: The system reads from the Order table to ensure that the record being deleted does not have an association with orders using the following attribute:

• ProductId</td></tr>
</table>

		[ALT]
		Step 7: The system removed all the details related to the product from the database from the following table: • Name • Description • Image • CustomerPrice • ResellerPrice • QuantityOnHand • ThresholdValue • DisplayProduct • ProductCategoryName • ProductBrandName
		Step 8: The system informs the administrator of the successful deletion of the product
ALTERNATE COURSES:	**ALT Step 3:** The administrator does not want to delete the product anymore. • Terminate use case.	
	ALT Step 5: The administrator does not confirm the deletion of the product. • Return to Step 4	
	ALT Step 6: The product cannot be deleted as it has an association with the Order table • The system displays an error message • The use case is terminated	
CONCLUSION:	This use case concludes when the product is deleted from the system	
POST-CONDITION:	• The system deletes all of the details related to the selected product from the **product entity**.	
BUSINESS RULES	• Only the administrator can delete a product.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 19- 4.5 Add Product category

PRODUCT CATEGORY SUBSYSTEM			
AUTHORS (s): Thenjiwe Ntsonda		**DATE: 06/04/2022**	
VERSION: 1.0		**LAST REVIEW DATE: 06/04/2022**	
USE CASE NAME:	Add Product category	**USE CASE TYPE**	
USE CASE ID:	4.5	Business Requirements:	☐
PRIORITY:	High	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 in which the administrator wishes to add a new product category to the system to be linked to products. The use case begins when the administrator adds the new product category to the system. The administrator will add all of the pertinent information regarding the product category and save it to the 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:	Actor Action	System Response	
	Step 1: The administrator wants to add a new product category	**Step 2**: The system requests the new product category details which have the following attributes: From the **ProductCategory** table: • Name	
	Step 3: The administrator inputs the details. **[ALT]**	**Step 4**: The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. **[ALT]**	

		Step 5: The system reads from the **ProductCategory** table to ensure that the record being added does not match any existing records using the following attribute: • Name **[ALT]**
		Step 6: The system saves the new product category details which have the following attributes: • ProductCategoryId • Name
		Step 7: The system displays a success message on the screen to confirm the addition of the new product category.
	ALT **Step 3:** The administrator does not want to add a product category anymore • The use case is terminated	
ALTERNATE COURSES:	ALT **Step 4:** The system fails to validate the information provided because of missing fields or incorrectly entered information. • Return to Step 3.	
	ALT **Step 5:** The product category already exists on the system • The system displays a validation error message • The use case is terminated	
CONCLUSION:	This use case concludes when a new product category is added.	
POST-CONDITION:	• The product category details are added to the database.	
BUSINESS RULES	• Only the administrator can add a new product category.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 20- 4.6 Search Product Category

<table>
<tr><th colspan="4">PRODUCT CATEGORY SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Thenjiwe Ntsonda</td><td colspan="2">DATE: 06/04/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 06/04/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Search Product Category</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">4.6</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">Medium</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the administrator wants to search for a product category on the system.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wants to search for a product category.</td><td colspan="2"></td></tr>
<tr><td>Step 2: The administrator enters the search parameter.</td><td colspan="2">Step 3: The system searches the following attributes in the ProductCategory entity for a match:
• Name</td></tr>
<tr><td></td><td colspan="2">Step 4: The system displays the search results. [ALT]</td></tr>
</table>

ALTERNATE COURSES:	**[ALT] Step 4**: The search query entered does not match any existing instances of product categories in the system database. • Return to Step 3 or terminate use case
CONCLUSION:	This use case concludes when the information is displayed on the system.
POST-CONDITION:	The **ProductCategory** entity was searched for and an entry that corresponds to the search query or parameter entered.
BUSINESS RULES	• None
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 21- 4.7 Update Product category

<table>
<tr><th colspan="4">PRODUCT CATEGORY SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Thenjiwe Ntsonda</td><td colspan="2">DATE: 06/04/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 06/04/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Update Product category</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">4.7</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process in which the administrator wishes to update a product category to the system. The use case begins when the administrator updates the product category details. The administrator will update all of the pertinent information regarding the product category and save it.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.
• A product category or brand relevant to the product category needs to exist before the product category is updated.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use is triggered when the administrator updates the product category as an option for customers and resellers to choose from.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wishes to update an existing product category’s details.</td><td colspan="2">Step 2: The system invokes Use Case 4.6 Search Product Category.</td></tr>
<tr><td></td><td colspan="2">Step 3: The system retrieves the desired product category details from the following tables:

From the ProductCategory table:
• Name</td></tr>
</table>

	Step 4: The administrator inputs the details required. **[ALT]**	**Step 5**: The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. **[ALT]**
		Step 6: The system prompts the administrator to confirm the update of a product category with the details provided.
	Step 7: The administrator confirms the information provided. **[ALT]**	**Step 8:** The system reads from the **ProductCategory** table to ensure that the record being updated does not match any existing records using the following attributes: • Name **[ALT]**
		Step 9: The system saves the updated product category details which have the following attributes: • Name
		Step 10: The system displays a success message on the screen to confirm the update of a product category.
ALTERNATE COURSES:	ALT **Step 4:** The administrator does not want to add a Product category anymore • The use case is terminated	
	ALT **Step 5:** The system fails to validate the information provided because of missing fields or incorrectly entered information. • Return to Step 4	

	ALT Step 7: The administrator does not confirm the updating of the product category details: • Return to Step 4 or terminate the use case
	ALT Step 8: The product category already exists on the system • The system displays a validation error message • The use case is terminated
CONCLUSION:	This use case concludes when a product category is updated.
POST-CONDITION:	• The product category details are updated to the database.
BUSINESS RULES	• Only the administrator can update a product category.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 22- 4.8 Delete Product Category

<table>
<tr><th colspan="4">PRODUCT CATEGORY SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Thenjiwe Ntsonda</td><td colspan="2">DATE: 06/04/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 06/04/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>Delete Product Category</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>4.8</td></tr>
<tr><td>PRIORITY:</td><td>Medium</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case narrative describes the process in which the administrator wishes to delete a product category from the system. The administrator will search for the required product category selected for deletion. The administrator will then confirm the deletion of the product category and the use case concludes when the system notifies the administrator of a successful deletion.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the administrator wishes to delete a product category.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wishes to delete a product category.</td><td colspan="2">Step 2: The system invokes Use Case 4.6 Search Product category.</td></tr>
<tr><td></td><td colspan="2">Step 3: The system retrieves the desired product category details from the following tables:

From the ProductCategory table:

• Name</td></tr>
</table>

	Step 4: The administrator selects the delete option **[ALT]**	**Step 5:** The system prompts the administrator to confirm the deletion of the product category.
	Step 6: The administrator confirms the deletion of the product category from the system **[ALT]**	**Step 7:** The system reads from the Product table to ensure that the record being deleted does not have an association with products using the following attribute: • ProductCategoryId **[ALT]**
		Step 8: The system removed all the details related to the product category from the database from the following table: • Name
		Step 9: The system informs the administrator of the successful deletion of the product category
ALTERNATE COURSES:	**ALT Step 4:** The administrator does not want to delete the product category anymore. • Terminate use case.	
	ALT Step 6: The administrator does not confirm the deletion of the product category. • Return to Step 4	
	ALT Step 7: The product category cannot be deleted as it has an association with the Product table • The system displays an error message • The use case is terminated	
CONCLUSION:	This use case concludes when the product category is deleted from the system	
POST-CONDITION:	• The system deletes all of the details related to the selected product category from the **ProductCategory table**.	
BUSINESS RULES	• Only the administrator can delete a product category.	

IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 23- 4.9 Add Product Brand

<table>
<tr><th colspan="4">PRODUCT BRAND SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Thenjiwe Ntsonda</td><td colspan="2">DATE: 06/04/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 06/04/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Add Product Brand</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">4.9</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process in which the administrator wishes to add a new product brand to the system to be linked to products. The use case begins when the administrator adds the new product brand to the system. The administrator will add all of the pertinent information regarding the product brand and save it to the database.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use case is triggered when the administrator wants to add a new product brand.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wants to add a new product brand</td><td colspan="2">Step 2: The system requests the new product brand details which have the following attributes:

From the ProductBrand table:
• Name</td></tr>
<tr><td>Step 3: The administrator inputs the details. [ALT]</td><td colspan="2">Step 4: The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. [ALT]</td></tr>
</table>

		Step 5: The system reads from the **ProductBrand** table to ensure that the record being added does not match any existing records using the following attribute: • Name **[ALT]**
		Step 6: The system saves the new product brand details which have the following attributes: • ProductBrandId • Name
		Step 7: The system displays a success message on the screen to confirm the addition of the new product brand.
	ALT Step 3: The administrator does not want to add a product brand anymore • The use case is terminated	
ALTERNATE COURSES:	**ALT Step 4:** The system fails to validate the information provided because of missing fields or incorrectly entered information. • Return to Step 3.	
	ALT Step 5: The product brand already exists on the system • The system displays a validation error message • The use case is terminated	
CONCLUSION:	This use case concludes when a new product brand is added.	
POST-CONDITION:	• The product brand details are added to the database.	
BUSINESS RULES	• Only the administrator can add a new product brand.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 24- 4.10 Search Product brand

<table>
<tr><th colspan="4">PRODUCT BRAND SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Thenjiwe Ntsonda</td><td colspan="2">DATE: 06/04/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 06/04/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>Search Product brand</td><td colspan="2">USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td>4.10</td><td colspan="2">Business Requirements: ☐</td></tr>
<tr><td>PRIORITY:</td><td>Medium</td><td colspan="2">System Analysis: ☒</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td><td colspan="2">System Design: ☐</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process in which the administrator searches for product brands on the system. The system responds by displaying the information relating to the search query.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the administrator wants to search for a product brand on the system.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wants to search for a product brand.</td><td colspan="2"></td></tr>
<tr><td>Step 2: The administrator enters the search parameter.</td><td colspan="2">Step 3: The system searches the following attributes in the ProductBrand entity for a match:

• Name</td></tr>
<tr><td></td><td colspan="2">Step 4: The system displays the search results. [ALT]</td></tr>
</table>

ALTERNATE COURSES:	**[ALT] Step 4**: The search query entered does not match any existing instances of product brands in the system database. • Return to Step 3 or terminate use case
CONCLUSION:	This use case concludes when the information is displayed on the system.
POST-CONDITION:	The **ProductBrand** entity was searched for and an entry that corresponds to the search query or parameter entered.
BUSINESS RULES	• None
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 25- 4.11 Update Product brand

<table>
<tr><th colspan="4">PRODUCT BRAND SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Thenjiwe Ntsonda</td><td colspan="2">DATE: 06/04/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 06/04/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Update Product brand</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">4.11</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process in which the administrator wishes to update a product brand to the system. The use case begins when the administrator updates the product brand details. The administrator will update all of the pertinent information regarding the product brand and save it.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.
• A product brand or brand relevant to the product brand needs to exist before the product brand is updated.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use is triggered when the administrator updates the product brand as an option for customers and resellers to choose from.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wishes to update an existing product brand details.</td><td colspan="2">Step 2: The system invokes Use Case 4.10 Search Product brand.</td></tr>
<tr><td></td><td colspan="2">Step 3: The system retrieves the desired product brand details from the following tables:

From the ProductBrand table:
• Name</td></tr>
</table>

	Step 4: The administrator inputs the details required. **[ALT]**	**Step 5**: The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. **[ALT]**
		Step 6: The system prompts the administrator to confirm the update of a product brand with the details provided.
	Step 7: The administrator confirms the information provided. **[ALT]**	**Step 8:** The system reads from the **ProductBrand** table to ensure that the record being updated does not match any existing records using the following attributes: • Name **[ALT]**
		Step 9: The system saves the updated product brand details which have the following attributes: • Name
		Step 10: The system displays a success message on the screen to confirm the update of a product brand.
ALTERNATE COURSES:	**ALT Step 4:** The administrator does not want to add a Product brand anymore • The use case is terminated	
	ALT Step 5: The system fails to validate the information provided because of missing fields or incorrectly entered information. • Return to Step 4	

	ALT Step 7: The administrator does not confirm the updating of the product brand details: • Return to Step 4 or terminate the use case
	ALT Step 8: The product brand already exists on the system • The system displays a validation error message • The use case is terminated
CONCLUSION:	This use case concludes when a product brand is updated.
POST-CONDITION:	• The product brand details are updated to the database.
BUSINESS RULES	• Only the administrator can update a product brand.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 26- 4.12 Delete Product brand

<table>
<tr><th colspan="4">PRODUCT BRAND SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Thenjiwe Ntsonda</td><td colspan="2">DATE: 06/04/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 06/04/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Delete Product brand</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">4.12</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">Medium</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case narrative describes the process in which the administrator wishes to delete a product brand from the system. The administrator will search for the required product brand selected for deletion. The administrator will then confirm the deletion of the product brand and the use case concludes when the system notifies the administrator of a successful deletion.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the administrator wishes to delete a product brand.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wishes to delete a product brand.</td><td colspan="2">Step 2: The system invokes Use Case 4.10 Search Product brand.</td></tr>
<tr><td></td><td colspan="2">Step 3: The system retrieves the desired product brand details from the following tables:

From the ProductBrand table:

• Name</td></tr>
</table>

	Step 4: The administrator selects the delete option **[ALT]**	**Step 5:** The system prompts the administrator to confirm the deletion of the product brand.
	Step 6: The administrator confirms the deletion of the product brand from the system **[ALT]**	**Step 7:** The system reads from the Product table to ensure that the record being deleted does not have an association with products using the following attribute: • ProductBrandId **[ALT]**
		Step 8: The system removed all the details related to the product brand from the database from the following table: • Name
		Step 9: The system informs the administrator of the successful deletion of the product brand
ALTERNATE COURSES:	**ALT Step 4:** The administrator does not want to delete the product brand anymore. • Terminate use case.	
	ALT Step 6: The administrator does not confirm the deletion of the product brand. • Return to Step 4	
	ALT Step 7: The product brand cannot be deleted as it has an association with the Product table • The system displays an error message • The use case is terminated	
CONCLUSION:	This use case concludes when the product brand is deleted from the system	
POST-CONDITION:	• The system deletes all of the details related to the selected product brand from the **ProductBrand table**.	
BUSINESS RULES	• Only the administrator can delete a product brand.	

IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 27- 4.13 Add Return Reason

PRODUCT SUBSYSTEM			
AUTHORS (s): Ofhani Mungani		**DATE: 06/04/2022**	
VERSION: 3.0		**LAST REVIEW DATE: 21/08/2022**	
USE CASE NAME:	Add Return Reason	**USE CASE TYPE**	
USE CASE ID:	4.13	Business Requirements:	☐
PRIORITY:	High	System Analysis:	☒
SOURCE:	ByteXpress 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 return reason is displayed to the customer and reseller when they log to return a product.		
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 return reason. The administrator will capture a return reason and save it 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 wishes to add a new Return Reason onto the system.		
TYPICAL COURSE	Actor Action	System Response	
OF EVENTS:	**Step 1:** The administrator wishes to add a new return reason.	**Step 2:** The system requests the new return reason details which have the following attribute: From the **ReturnReason** table: • ReturnReasonName	

	Step 3: The administrator inputs the details. **[ALT]**	**Step 4:** The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. **[ALT]**
	Step 5: The administrator selects the option to add a new return reason onto the system.	**Step 6**: T The system reads from the **ReturnReason** table to ensure that the record being added does not match any existing records using the following attribute: • ReturnReasonName . **[ALT]**
		Step 7: The system saves the new return reason details which have the following attributes: • Id • ReturnReasonName
		Step 8: The system displays a success message on the screen to confirm the addition of a new return reason.
ALTERNATE COURSES:	**ALT Step 3**: The administrator wishes not to procced to add a new return reason. • Terminate Use Case	
	ALT Step 4: The system fails to validate the information provided because of missing fields or incorrectly entered information • Return to Step 3	
	ALT Step 6: The return reason's name already exists on the system. • Terminate Use Case	
CONCLUSION:	A new instance of return reason has been successfully added to the system database.	
POST-CONDITION:	The system now has a new return reason for customers and resellers to choose from when logging a product return.	
BUSINESS RULES	Only the administrator can add a new Return Reason to the system.	

IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 28- 4.14 Search Return Reason

<table>
<tr><th colspan="4">PRODUCT SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Ofhani Mungani</td><td colspan="2">DATE: 06/04/2022</td></tr>
<tr><td colspan="2">VERSION: 3.0</td><td colspan="2">LAST REVIEW DATE: 21/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>Search Return Reason</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>4.14</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged into the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The wishes to search for a Return Reason onto the system.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator requests to search for a return reason onto the system.</td><td colspan="2"></td></tr>
<tr><td>Step 2: The administrator inputs the search parameter.</td><td colspan="2">Step 3: The system searches the following attributes in the ReturnReason entity for a match:

• ReturnReasonName</td></tr>
</table>

		Step 4: The system displays the results to the administrator. **[ALT]**
ALTERNATE COURSES:	**[ALT] Step 4**: The search query entered does not match any existing instances of Return Reason in the system database. • Terminate use case	
CONCLUSION:	This use case concludes when a list of matching results from the search query is displayed on the screen.	
POST-CONDITION:	The Administrator can see the return reason matched from the search query.	
BUSINESS RULES	Only the administrator can search for a return reason to the system	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	
	•	

Table 29- 4.15 Update Return Reason

PRODUCT SUBSYSTEM			
AUTHORS (s): Ofhani Mungani		**DATE: 06/04/2022**	
VERSION: 3.0		**LAST REVIEW DATE: 21/08/2022**	
USE CASE NAME:	Update Return Reason	**USE CASE TYPE**	
USE CASE ID:	4.15	Business Requirements: ☐	
PRIORITY:	High	System Analysis: ☒	
SOURCE:	ByteXpress Requirements List	System Design: ☐	
PRIMARY BUSINESS ACTOR	• Administrator		
PRIMARY SYSTEM ACTOR	• None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• Customers and Resellers – The newly updated return reason is displayed to customers and resellers when they want to log a product return.		
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 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.		
PRE-CONDITION:	• The administrator must be logged onto the system.		
TRIGGER:	The use is triggered when the administrator wishes to update a Return Reason.		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: The administrator requests to update a return reason on the system	**Step 2:** The system **invokes use case 4.2 "Search Return Reason".**	
	Step 3: The administrator selects the desired return reason.	**Step 4**: The system retrieves the desired return reason details from the following tables:	

		From the **ReturnReason** table: • ReturnReasonName .
	Step 5: The administrator inputs the details required. **[ALT]**	**Step 6**: The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. **[ALT]**
		Step 7: The system prompts the administrator to confirm the update of a return reason with the details provided
	Step 8: The administrator confirms the information provided. **[ALT]**	**Step 9:** The system reads from the **ReturnReason** table to ensure that the record being updated does not match any existing records using the following attribute: • ReturnReasonName **[ALT]**
		Step 10: The system saves the updated return reason details which have the following attribute: • ReturnReasonName .
		Step 11: The system displays a success message on the screen to confirm the update of a return reason.
ALTERNATE COURSES:	**ALT Step 5:** The administrator wishes not to procced to update a name of a return reason. • Terminate Use Case	
	ALT Step 6: The system fails to validate the information provided because of missing fields or incorrectly entered information. • Return to Step 5	

	ALT Step 8: The administrator does not confirm the updating of the return reason's details: • Return to Step 5 or terminate the use case
	ALT Step 9: The entered return reason's name already exists on the system. • Terminate Use Case
CONCLUSION:	An instance of return reason has been successfully updated to the system database.
POST-CONDITION:	• The system now has an updated return reason for customers and resellers to choose from when they log a product return on the system.
BUSINESS RULES	• Only the administrator can update a return reason to the system
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 30- 4.16 Delete Return Reason

PRODUCT SUBSYSTEM			
AUTHORS (s): Ofhani Mungani		**DATE: 06/04/2022**	
VERSION: 3.0		**LAST REVIEW DATE: 21/08/2022**	
USE CASE NAME:	Delete Return Reason	**USE CASE TYPE**	
USE CASE ID:	4.16	Business Requirements:	☐
PRIORITY:	High	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 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:	This use case is called when the administrator wishes to delete a Return Reason.		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: The administrator wishes to delete a Return Reason.	**Step 2:** The system invokes Use Case 4.2 “Search Return Reason”.	
	Step 3: The administrator selects the desired return reason	**Step 4:** The system retrieves the return reason information from the database with the following attributes: From the **Country entity** • **CountryName**	

		.
	Step 5: The administrator selects the delete option [ALT	**Step 6:** The system prompts the administrator to confirm the deletion of the selected return reason. [ALT]
	Step 7: The administrator confirms the deletion of the return reason from the system [ALT]	**Step 8:** The system removed all the details related to the Return Reason from the database [ALT]
		Step 9: The system informs the administrator of the successful deletion of the return reason
ALTERNATE COURSES:	[ALT] **Step 5**: The administrator wishes not to procced to delete a return reason. • Terminate use case	
	[ALT] **Step 7**: The administrator does not confirm the deletion of the return reason. • Return to step 5	
	ALT **Step 8:** The return reason cannot be deleted as it has an association with the ProductReturn table • The system displays an error message • Terminate usecase	
CONCLUSION:	This use case concludes when the administrator is notified of the successful deletion of a return reason.	
POST-CONDITION:	• The system deletes all of the details related to the selected return reason from the **ReturnReason entity**.	
BUSINESS RULES	• Only the administrator can delete a return reason.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 31- 4.17 Browse Product Catalogue

<table>
<tr><th colspan="4">PRODUCT SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Thenjiwe Ntsonda</td><td colspan="2">DATE: 06/04/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 06/04/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Browse Product Catalogue</td><td>USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">4.17</td><td>Business Requirements: ☐</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td><td>System Analysis: ☒</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td><td>System Design: ☐</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• User</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• Resellers must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use case is triggered when the users of the system want to browse the product catalogue.</td></tr>
<tr><td rowspan="2">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The user wants to browse the product catalogue.</td><td colspan="2">Step 2: The system retrieves the product information available from the following tables:

From the Product table:

• Name
• Description
• Image
• CustomerPrice</td></tr>
</table>

		From the **ProductCategory** table • Name From the **ProductBrand** table: • Name **[ALT]**
	Step 3: The user enters their required search parameter into the search box. **[ALT]**	**Step 4:** The system retrieves the products matching the entered search query and displays it to the user.
ALTERNATE COURSES:	ALT **Step 2:** The user is logged in as a reseller. • The system displays the reseller prices from the **Product** entity • Go to Step 3	
	ALT **Step 3:** The user filters the product catalogue using the product categories 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:	• The product catalogue is displayed on the system.	
BUSINESS RULES	• The reseller prices are only displayed to users logged in as resellers.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 32 - 4.18 Configure Product Inventory Item(s)

<table>
<tr><th colspan="4">PRODUCT SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Kyle van Eeden</td><td colspan="2">DATE: 06/07/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 06/07/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>Configure Product Inventory Item(s)</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>4.18</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Inventory Manager or Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">The use case begins when the inventory manager or administrator wishes to specify the components a specific product is composed of and their respective quantities of each inventory item. The system will prompt the administrator or inventory manager to search for and then select the inventory items used for the production of the finished product and specify the quantities required for each inventory item. The use case concludes when the inventory items list has been updated and saved for the specific product.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The product to be configured for production must exist on the system.

The inventory items used to produce the final product must already be defined on the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the system administrator or inventory manager wishes to configure an already existing product for production.</td></tr>
<tr><td rowspan="2">TYPICAL COURSE</td><td colspan="2">Actor Action</td><td>System Response</td></tr>
<tr><td colspan="2">Step 1: The system administrator or inventory manager wishes to configure an existing product on the system for production activities.</td><td>Step 2: The system retrieves the following attributes from the Product entity for each existing product and displays the following details:

• ProductName</td></tr>
</table>

		• QuantityOnHand
	Step 3: The system administrator or inventory manager selects the option to configure a specific product.	**Step 3:** The system retrieves the following details for the selected product: From the **Product** entity: • ProductName • QuantityOnHand Details retrieved from the **ProductInventory** entity: • InventoryItemName • InventoryItemQuantity
	Step 4: The system administrator or inventory manager enters the search parameter for the inventory item they wish to select. **[ALT]**	**Step 5:** The system invokes Use Case 8.2 ‘Search Inventory Item’
	Step 6: The administrator or inventory manager selects the option to add the inventory item to the list of inventory items for the specific product. **[ALT]**	**Step 7:** The system adds the selected inventory item to the product’s list of inventory items.
	Step 8: The administrator or inventory manager specifies the quantity for each inventory item within the list of selected inventory items. **[ALT]**	**Step 9:** The system saves the updated list of inventory items and their respective quantities to the **ProductInventory** entity using the following attributes: • InventoryId • InventoryQuantity
		Step 10: The system displays a confirmation message to indicate the update was successful.

ALTERNATE COURSES:	**ALT Step 4:** The administrator or inventory manager selects the option to cancel the product configuration operation: • Return to step 2
	ALT Step 6: The administrator or inventory manager selects the option to cancel the product configuration operation: • Return to step 2
	ALT Step 8: The administrator or inventory manager selects the option to cancel the product configuration operation: • Return to step 2
CONCLUSION:	This use case concludes when the system displays an alert confirming the product's inventory items has been successfully updated.
POST-CONDITION:	The product's assigned inventory items and respective quantities is saved in the **ProductInventory** entity.
BUSINESS RULES	• Only the system administrator and inventory manager roles should be allowed to configure a product's required inventory items for production.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• The quantity required of each inventory item in the product's list must be at least greater than 0 at all times.
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

2.5. Order Subsystem

Table 33 - 5.1 Add Item(s) to Cart

<table>
<tr><th colspan="4">ORDER SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Ofhani Mungani</td><td colspan="2">DATE: 06/24/2022</td></tr>
<tr><td colspan="2">VERSION: 1.1</td><td colspan="2">LAST REVIEW DATE: 07/29/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>Add Item(s) to Cart</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>5.1</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Customer or Reseller</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• 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.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when a customer or reseller wishes to add a specific product to their cart.</td></tr>
<tr><td rowspan="2">TYPICAL COURSE</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The user navigates to the Products Screen on the navigation list top navi-bar</td><td colspan="2">Step 2: The system retrieves the following details for all products from the Product entity:</td></tr>
</table>

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:

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:

Details to be read	Attributes in the table
Product Name	ProductName
Product Picture Url	ProductPictureUrl
Customer Price or Reseller Price	CustomerPrice/ ResellerPrice

From the **ProductCategory** entity:

Details to be read	Attributes in the table
Product Category Name	ProductCategoryName

From the **ProductBrand** entity:

Details to be read	Attributes in the table
Product Brand Name	ProductBrandName

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
Product list	table	This control will be used to display all the products that exist on the system

[ALT]

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:

Details to be read	Attributes in the table
Quantity on Hand	QuantityonHand

<table>
<tr><td></td><td></td><td>[ALT]</td></tr>
<tr><td></td><td></td><td>Step 8: The system read the following account details of the customer or reseller from the User entity:

<table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>UserId</td><td>UserId</td></tr></table></td></tr>
<tr><td></td><td></td><td>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 UserCart entity using SQL INSERT query:
<table><tr><th>Details to be read</th><th>Attributes in the table</th></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]</td></tr>
<tr><td></td><td></td><td>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.</td></tr>
<tr><td>ALTERNATE COURSES:</td><td colspan="2">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.</td></tr>
</table>

<table>
<tr><td></td><td>
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.
</td></tr>
<tr><td></td><td>
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.
</td></tr>
<tr><td></td><td>
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.
</td></tr>
<tr><td></td><td>
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.
</td></tr>
<tr><td>CONCLUSION:</td><td>The system displays an alert notification that informs the customer or reseller that their cart has been updated successfully.</td></tr>
<tr><td>POST-CONDITION:</td><td>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.</td></tr>
<tr><td>BUSINESS RULES</td><td>
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.
</td></tr>
</table>

IMPLEMENTATION CONTRAINTS 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 34 - 5.2 Update Cart

ORDER SUBSYSTEM			
AUTHORS (s): Nomusa Vumisa		**DATE: 06/21/2022**	
VERSION: 1.1		**LAST REVIEW DATE: 06/22/2022**	
USE CASE NAME:	Update Cart	**USE CASE TYPE**	
USE CASE ID:	5.2	Business Requirements:	☐
PRIORITY:	High	System Analysis:	☒
SOURCE:	ByteXpress 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 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	Actor Action	System Response	
	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 increments the quantity of a specific product in their cart. **[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 • QuantityOnHand	

		[ALT]
		Step 5: The system saves the updated quantity of the product in the **Cart** entity using the following attribute: • Quantity **[ALT]**
		Step 6: The system reads the updated product quantity and recalculates the cart total based on the updated quantity of each product in the user's cart.
		Step 7: The system displays the updated product quantity's and cart total
ALTERNATE COURSES:	**ALT Step 3:** The customer or reseller decrements the quantity of a specific product in their cart • 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 displays an error message • Terminate 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 user's cart • Terminate 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 CONTRAINTS AND SPECIFICATIONS	• The cart total should be updated the moment that a customer or reseller changes the quantity of a product in their cart. • A user's cart items should be persisted in the database for at least 14 days from the time it is added.	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 35 - 5.3 Remove Item(s) from Cart

<table>
<tr><th colspan="4">ORDER SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Thenjiwe Ntsonda</td><td colspan="2">DATE: 06/23/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 06/23/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>Remove Item(s) from Cart</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis:
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>5.3</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Customer or Reseller</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• 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.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when a customer or reseller wishes to remove specific product from their cart.</td></tr>
<tr><td rowspan="5">TYPICAL COURSE</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The customer or reseller wishes to remove a specific cart item within their current cart.</td><td colspan="2">Step 2: Extends Use Case 5.4 View Cart.</td></tr>
<tr><td>Step 3: The customer or reseller selects the option to remove a specific product from their cart.</td><td colspan="2">Step 4: The system removes the requested product from the user's cart and saves the updated cart details which are stored in the Cart entity with the following attributes:
• Id
• ProductID
• Count
• CartItemPrice
• UserID</td></tr>
<tr><td></td><td colspan="2">Step 6: The system reads the new details in the Cart entity and recalculates the cart total and displays the updated CartItemPrice and count on the user interface.</td></tr>
<tr><td></td><td colspan="2">Step 7: The system decreases and displays the cart count and confirms</td></tr>
</table>

		that the product has been removed successfully.
ALTERNATE COURSES:	• None	
CONCLUSION:	The system displays an alert which confirms that the product has been removed from the cart successfully.	
POST-CONDITION:	The cart item requested for removal is deleted from the **Cart** entity.	
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 CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• The customer and reseller have a stable internet connection.	
OPEN ISSUES:	• None	

Table 36 - 5.4 View Cart

<table>
<tr><th colspan="4">ORDER SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHOR(s): APHIWE SHOZI</td><td colspan="2">DATE: 07/02/2022</td></tr>
<tr><td colspan="2">VERSION: 1.1</td><td colspan="2">LAST REVIEW DATE: 29/02/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>View Cart</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>5.4</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Customer or Reseller</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The customer/reseller must have an account registered on the system
• The customer/reseller needs to be logged in</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">When a customer/reseller wishes content of their cart.</td></tr>
<tr><td rowspan="2">TYPICAL COURSE</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The customer/reseller wishes to view the contents of their cart.</td><td colspan="2">Step 2: The systems retrieves the corresponding cart using the Cart_ID in the CART table and retrieve the following information:
• Products
• Quantity
• Cart total

The system will retrieve the items in the cart using the Product_ID (foreign key) referencing the PRODUCT table and retrieve the following information:
• Name
• Description
• Image
• Price
The system will then retrieve the VAT percentage from the VAT table and calculate the VAT of the cart</td></tr>
</table>

		[ALT]
		Step 3: The system will then perform the following calculations: • VAT exclusive cart total: The sum total of all the products in the cart • VAT amount: The sum total of all VAT applicable items ((Product Price *quantity) * Vat Percentage) Sum of the product items in the cart • The overall Cart Total: VAT exclusive Cart Total + VAT The system will then display this information .
ALTERNATE COURSES:	**Alt Step 2** The customer/reseller's cart is empty. The system will notify the user. • Terminate use case.	
CONCLUSION:	The contents of the cart are displayed to the customer/reseller	
POST-CONDITION:	The items within the cart are displayed correctly	
BUSINESS RULES	• 15% VAT is charged on applicable items	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 37- 5.5 Checkout Cart

<table>
<tr><th colspan="4">ORDER SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Kyle van Eeden</td><td colspan="2">DATE: 07/02/2022</td></tr>
<tr><td colspan="2">VERSION: 1.1</td><td colspan="2">LAST REVIEW DATE: 29/02/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>Checkout Cart</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>5.5</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Customer or Reseller</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case begins when a customer or reseller wishes to check out their current cart. The system displays a form that is populated with the following details retrieved for the user requesting to proceed to checkout: First Name, Surname, Phone Number, Street Address, Country, Province, City and Suburb. The system also displays the cart summary details which includes all products within the cart along with their respective quantities and prices. A cart total value is displayed which is calculated using the prices of all added products and adding the calculated VAT inclusive amount applicable to the cart. The customer or reseller will fill out all the required delivery details and select a courier service to be used. For customers checking out, the use case concludes once the system invokes use case 5.6 ‘Make Payment’. For resellers checking out, the system will create and save the reseller order details in the Order and OrderLine entities, and the use case concludes once the reseller is informed that their order has been successfully placed.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The customer or reseller should have a registered account on the system.
• The customer or reseller must be logged into their account.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when a customer or reseller wishes to check out their current cart captured on the system.</td></tr>
<tr><td>TYPICAL COURSE</td><td colspan="2">Actor Action</td><td>System Response</td></tr>
</table>

	Step 1: The customer or reseller wishes to check out their current cart. **[ALT]**	**Step 2:** The system retrieves the following user account details from the **User** entity: • FirstName • Surname • PhoneNumber • StreetAddress • CountryID (FK) • ProvinceID (FK) • CityID (FK) • SuburbID (FK) The below user address details are read from the following entities: From the **Country** entity: • CountryName From the **Province** entity: • ProvinceName From the **City** entity: • CityName From the **Suburb** entity: • SuburbName The system also retrieves the following details for the identified user who made the request to check out from the **Cart** entity: • ProductID • Count The names of all products within the user's cart is retrieved from the **Product** entity: • ProductName

		The following details are retrieved from the **Courier** entity for all couriers: • CourierName • CourierFee The VAT percentage to be applied to the cart subtotal is retrieved from the **VAT** entity: • VatValue
		Step 3: The system calculates the cart subtotal by adding up the prices for all products within the cart multiplied by their respective quantities specified. The applicable VAT amount is calculated by multiplying the retrieved VATValue value with the current cart subtotal. The cart's grand total is calculated by adding the calculated VAT inclusive amount to the cart's subtotal.
		Step 4: The system displays the details retrieved for the user as well as the calculated cart totals. The customer or reseller is prompted to select a delivery method to be used.
	Step 5: The customer or reseller selects the delivery method they wish to use for their potential order.	**Step 6:** The system captures the selected delivery method and adds the price associated with the selected method to the cart's calculated sub-total amount. The system will also recalculate the order's total by adding the updated cart sub-total amount to the VAT amount.

	Step 7: The customer or reseller enters their desired delivery details and selects a courier service to be used.	
	Step 8: The customer or reseller selects the option to proceed to make a payment for their potential order. **[ALT]**	**Step 9:** The system validates that all the captured details for the potential order to be placed is correct according to the validation requirements in the **Order** entity. **[ALT]**
		Step 10: The user requesting to check out their cart is identified by the system as a customer user and the system checks whether there are still sufficient products on hand to fulfil the order by reading the following details from the **Product** entity for each product added to the customer's cart: • QuantityOnHand The system then invokes Use Case 5.6 "Make Payment" to allow the customer to make the required payment. **[ALT]**
ALTERNATE COURSES:	**Alt Step 1**: The customer or reseller no longer wishes to check out their current cart and selects the option provided to cancel the operation. • The use case concludes.	
	Alt Step 8: The customer or reseller no longer wishes to perform a checkout on their current cart and selects the option provided to cancel the operation. • The use case concludes.	
	Alt Step 9: The captured order details is incomplete and the system displays the required validation details: • Return to **Step 7**	
	Alt Step 10 a: There are no longer sufficient products available based on the quantity of each product added to the customer's cart: • The system displays a message to inform the customer that the order cannot be placed due to insufficient quantities of products being available. • The use case concludes.	

	Alt Step 10 b: The user requesting to check out their cart is registered as a Reseller: • The system will create the reseller's captured order details in the **Order** and **OrderLine** entities. The order to be created will have an order status of 'Pending' and a payment status of "Payment Outstanding". • The use case concludes.
CONCLUSION:	The system invokes Use Case 5.6 "Make Order Payment".
POST-CONDITION:	The user's cart details are retrieved from the **Cart** entity and the cart summary totals are calculated and displayed on the user interface.
BUSINESS RULES	• Customers and resellers should be allowed to specify a different address to be used for delivery from what is currently stored according to the logged in user's profile details. • VAT is to be applied and calculated for each user's cart based on the current VAT rate stored on the system. • Customers are required to make a payment before the order can be placed on the system. • Resellers are not required to make a payment in order to place an order on the system.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• The customer or reseller has no internet access and cannot check out their current cart as a result. • The system is offline and the customer or reseller is not able to check out their current cart. • Customers should be restricted from checking out their cart if the quantities of all products are not available as per what is stored on the system.
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 38 - 5.6 Make Payment

ORDER SUBSYSTEM			
AUTHORS (s): Thenjiwe Ntsonda		DATE: 24/07/2022	
VERSION: 1.0		LAST REVIEW DATE: 24/07/2022	
USE CASE NAME:	Make Payment	USE CASE TYPE	
USE CASE ID:	5.6	Business Requirements: ☐	
PRIORITY:	High	System Analysis: ☒	
SOURCE:	ByteXpress Requirements List	System Design: ☐	
PRIMARY BUSINESS ACTOR	• Customer		
PRIMARY SYSTEM ACTOR	• None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	This use case begins when the customer wishes to proceed with the payment of an order which includes products that have been added to the user's cart. The system will redirect the user to the Stripe payment gateway, in which the payment will be processed. Once the payment is successful, the order details will be created, the user cart will be cleared, and an order confirmation email will be sent to the customer.		
PRE-CONDITION:	• The customer must be logged in. • 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	**Actor Action**	**System Response**	
	Step 1: The customer selects the option to proceed with the payment of the items added to their cart.	**Step 2:** The system redirects the customer to the Stripe payment gateway in which the payment will be made.	
		Step 3: The system receives a successful payment message from Stripe and saves the following attributes to the **Order** table: • **OrderId** • **FirstName** • **Surname** • **PhoneNumber** • **StreetAddress** • **OrderDate** • **OrderStatus** • **OrderTotal** • **ParcelTrackingNumber** • **Disptatched Date**	

		And to the **OrderLine** table with the following attributes: • **OrderLineId** • **OrderId** • **ProductId** • **Price** • **Count**
		Step 4: The system clears the user's cart details which are stored in the **UserCart** table with the following attributes: • CartId • ProductId • Count • CartItemPrice • UserId
		Step 5: The system retrieves the **OrderId** from the **Order** table and displays a success message to the customer
		Step 6: The system retrieves the customer's email and generates an order confirmation email
		Step 7: The system sends the generated email to the customer.
ALTERNATE COURSES:	• None	
CONCLUSION:	The system displays an alert which confirms that the order has been placed successfully.	
POST-CONDITION:	The order details are saved to the Order table and the OrderLine table. A confirmation order email is sent to the customer.	
BUSINESS RULES	• Only customers have access to this usecase • The order details are only saved after payment has been received successfully.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• The customer has a stable internet connection.	
OPEN ISSUES:	• None	

Table 39 - 5.7 View Orders Overview

<table>
<tr><th colspan="5">ORDER SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Nomusa Vumisa</td><td colspan="3">DATE: 11/07/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="3">LAST REVIEW DATE: 11/07/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">View Orders Overview</td><td colspan="2">USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">5.7</td><td>Business Requirements:</td><td>☐</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td><td>System Analysis:</td><td>☒</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td><td>System Design:</td><td>☐</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="4">• Administrator/Sales Assistant</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="4">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="4">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="4">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="4">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</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="4">The administrator or sales assistant must be logged on the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="4">The use case is called when the administrator or sales assistant wants to view an order</td></tr>
<tr><td rowspan="3">TYPICAL COURSE</td><td>Actor Action</td><td colspan="3">System Response</td></tr>
<tr><td>Step 1: The administrator or sales assistant wants to view an order on the system</td><td colspan="3">Step 2: The system invokes Use Case 5.8 Search Orders</td></tr>
<tr><td>Step 3: The administrator or sales assistant selects the order they wish to view</td><td colspan="3">Step 4: The system displays the selected order details using the following attributes:

From the User table.
• FirstName
• Surname
• PhoneNumber
• EmailAddress

From the Order table
• OrderId
• FirstName
• Surname
• PhoneNumber
• StreetAddress
• OrderDate
• OrderStatus
• OrderTotal</td></tr>
</table>

		• ParcelTrackingNumber • DispatchedDate From the **Suburb** Table • SuburbName • PostalCode From the **City** Table • CityName From the **Province** Table • ProvinceName From the **Country** Table • CountryName From the **Courier** Table • CourierName From the **Product** table • Name From the **OrderLine** table • OrderItemPrice • OrderItemCount
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 CONTRAINTS AND SPECIFICATIONS	• The administrator or sales assistant cannot view an order without an internet connection	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 40 - 5.8 Search Order

<table>
<tr><th colspan="4">ORDER SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Nomusa Vumisa</td><td colspan="2">DATE: 07/11/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 07/11/2022</td></tr>
<tr><td>USE CASE INVENTORYITEMNAME:</td><td colspan="2">Search Order</td><td rowspan="4">USE CASE TYPE

Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">5.8</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">Medium</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator/Sales Assistant</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator or sales assistant must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the administrator or sales assistant wants to search for an order on the system.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator or sales assistant want to search for an order</td><td colspan="2"></td></tr>
<tr><td>Step 2: The administrator or sales assistant enter the search parameter</td><td colspan="2">Step 3: The system searches for a match using the following attributes:

From the User table.
• EmailAddress

From the Order table
• OrderId
• FirstName
• Surname
• PhoneNumber
• OrderDate
• OrderStatus
• OrderTotal</td></tr>
<tr><td></td><td colspan="2">Step 4: The system displays the search results [ALT]</td></tr>
<tr><td>ALTERNATE COURSES:</td><td colspan="3">ALT Step 4: The search query entered does not match any existing instances of an order.</td></tr>
</table>

	Return to Step 2 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 CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 41 -5.9 Query Order

INVENTORY MANAGEMENT SUBSYSTEM		
AUTHORS (s): Ofhani Mungani	DATE: 06/06/2022	
VERSION: 1.0	LAST REVIEW DATE: 20/07/2022	
USE CASE INVENTORYITEMNAME:	Query Order	USE CASE TYPE Business Requirements: ☐ System Analysis: ☒ System Design: ☐
USE CASE ID:	5.9	
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 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:	**Actor Action**	**System Response**
	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 selects the order they wish to query.	**Step 4**: The system displays the appropriate query reason details: From the **QueryReason** entity: • Name
	Step 5: The customer/reseller inputs the required details. **ALT]**	**Step 6**: The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. **[ALT]**
		Step 7: The system saves the query reason associated with the order and customer/reseller on the database with the following attributes

		From the **OrderQuery** entity • Description From the **Query Reason** entity: • Name From the **Order** entity: • Id
		Step 8: The system displays a success message on the screen to confirm the submission of the order query.
ALTERNATE COURSES:	**ALT Step 5**: The customer/reseller doesn't want to query their order: • Return to step 2 • Terminate use case	
	ALT Step 6: The system fails to validate the information provided because of missing fields or incorrectly entered information. • Return to Step 5.	
CONCLUSION:	This use case concludes when an order query is submitted.	
POST-CONDITION:	• The customer/reseller's order query is added to the database. .	
BUSINESS RULES	• Only the customer/reseller can query an order.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 42 - 5.10 Review Order

INVENTORY MANAGEMENT SUBSYSTEM		
AUTHORS (s): Ofhani Mungani	**DATE: 06/06/2022**	
VERSION: 1.0	**LAST REVIEW DATE: 20/07/2022**	
USE CASE INVENTORYITEMNAME:	Review Order	**USE CASE TYPE**
USE CASE ID:	5.10	Business Requirements: ☐
PRIORITY:	High	System Analysis: ☒
SOURCE:	ByteXpress Requirements List	System Design: ☐
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 review, the system then retrieve the reviews from the **Review Reason** entity for customers/resellers to choose from. 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 review an order on the system.	
TYPICAL COURSE OF EVENTS:	**Actor Action**	**System Response**
	Step 1: The customer/reseller wants to review their placed order.	**Step 2**: The system invokes use case **5.16 View Placed Order**
	Step 3: The customer/reseller selects the order they wish to review.	**Step 4**: The system displays the appropriate review reason details: From the **Review Reason** entity: • Name
	Step 5: The customer/reseller inputs the required details **[ALT]**	**Step 6**: The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. **[ALT]**
		Step 7: The system saves the query reason associated with the order and customer/reseller on the database with the following attributes

		From the **ReviewOrder** entity • Description From the **Review Reason** entity: • Name From the **Order** entity: • Id • Product
		Step 8: The system displays a success message on the screen to confirm the submission of the order review.
ALTERNATE COURSES:	ALT **Step 5**: The customer/reseller doesn't want to review their order: • Return to step 2 • Terminate use case	
	ALT **Step 6**: The system fails to validate the information provided because of missing fields or incorrectly entered information. • Return to Step 5.	
CONCLUSION:	• This use case concludes when an order review is submitted.	
POST-CONDITION:	• The customer/reseller's order review is added to the database.	
BUSINESS RULES	• Only the customer/reseller can review an order.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 43 - 5.11 Create Courier

COURIER SUBSYSTEM			
AUTHORS (s): Thenjiwe Ntsonda		**DATE: 12/05/2022**	
VERSION: 1.0		**LAST REVIEW DATE: 12/05/2022**	
USE CASE NAME:	Create Courier	**USE CASE TYPE**	
USE CASE ID:	5.11	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 in which the administrator wishes to add a new courier to the system. The use case begins when the administrator adds the new courier on the system. The administrator will add all of 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 to the system.		
TYPICAL COURSE OF EVENTS:	**Actor Action**	**System Response**	
	Step 1: The administrator wants to add a new courier to the system.	**Step 2**: The system requires the admin to add information about the courier in the **Courier entity** such as: • **CourierName** • **CourierFee** • **EstimatedDeliveryTime**	
	Step 3: The administrator inputs the required details.	**Step 4**: The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. **[ALT]**	
		Step 5: The system reads from the **Courier entity** to ensure that the record being added does not match any existing records. **[ALT]**	
		Step 6: The system saves its information to the **Courier entity** in the database with the following attributes • **ID** • **CourierName** • **CourierFee** • **EstimatedDeliveryTime**	

		Step 7: The system displays a successful creation message.
	ALT Step 4: The system fails to validate the information provided because of missing fields or incorrectly entered information. • Return to Step 3.	
	ALT Step 5: The courier being added matches an existing record on the system • The system restricts the addition of the courier to prevent duplicate values and displays a modal dialog. • The use case is terminated.	
CONCLUSION:	This use case concludes when a new courier is added.	
POST-CONDITION:	• The courier details are added to the database.	
BUSINESS RULES	• Only the administrator can add a new courier.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 44 - 5.12 Search courier

COURIER SUBSYSTEM			
AUTHORS (s): Thenjiwe Ntsonda		**DATE: 12/05/2022**	
VERSION: 1.0		**LAST REVIEW DATE: 12/05/2022**	
USE CASE NAME:	Search Courier	**USE CASE TYPE**	
USE CASE ID:	5.12	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 in which the administrator searches for couriers 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**	
	Step 1: The administrator wants to search for a courier.		
	Step 2: The administrator inputs the search parameter.	**Step 3:** The system searches and retrieves the following attributes in the **Courier entity** for a match: • **CourierName** • **CourierFee** • **EstimatedDeliveryTime**	
		Step 4: The system displays the search results information. **[ALT]**	
ALTERNATE COURSES:	**[ALT] Step 4**: The search query entered does not match any existing instances of couriers in the system database. • Return to Step 3		
CONCLUSION:	This use case concludes when the information is displayed on the system.		
POST-CONDITION:	The **Courier** entity was searched for and an entry that corresponds to the search query or parameter entered.		
BUSINESS RULES	• None		
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None		
ASSUMPTIONS:	• None		
OPEN ISSUES:	• None		

Table 45 - 5.13 Update Courier

COURIER SUBSYSTEM			
AUTHORS (s): Thenjiwe Ntsonda		**DATE: 12/05/2022**	
VERSION: 1.0		**LAST REVIEW DATE: 12/05/2022**	
USE CASE NAME:	Update Courier	**USE CASE TYPE**	
USE CASE ID:	5.13	Business Requirements: ☐	
PRIORITY:	High	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 in which the administrator wishes to update a courier on the system. The use case begins when the administrator updates all of the pertinent information regarding the courier and save it to the system.		
PRE-CONDITION:	• The administrator must be logged onto the system.		
TRIGGER:	The use is triggered when the administrator wishes to update the courier.		
TYPICAL COURSE OF EVENTS:	**Actor Action**	**System Response**	
	Step 1: The administrator wants to update a courier existing on the system.	**Step 2:** The system invokes Use Case 5.12 "Search Courier"	
	Step 3: The administrator selects the desired Courier.	**Step 4**: The system requires the admin to update more information about the courier in the **Courier entity** such as: • **CourierName** • **CourierFee** • **EstimatedDeliveryTime**	
	Step 5: The administrator inputs the details required.	**Step 6**: The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. **[ALT]**	
		Step 7: The system reads from the **Courier entity** to ensure that the record being added does not match any existing records. **[ALT]**	
		Step 8: The system updates and saves its information to the **Courier entity** with the following attributes • **CourierName** • **CourierFee** • **EstimatedDeliveryTime**	

		Step 9: The system displays a successfully updated message.
ALTERNATE COURSES:	**ALT Step 6:** The system fails to validate the information provided because of missing fields or incorrectly entered information. • Return to Step 4.	
	ALT Step 7: The courier being updated matches an existing record on the system • The system restricts the update of the courier to prevent duplicate values and displays a modal dialog. • The use case is terminated.	
CONCLUSION:	This use case concludes when a courier is updated.	
POST-CONDITION:	• The courier details are updated to the database.	
BUSINESS RULES	• Only the administrator can update a courier.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 46 - 5.14 Delete Courier

<table>
<tr><th colspan="4">COURIER SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Thenjiwe Ntsonda</td><td colspan="2">DATE: 12/05/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 12/05/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>Delete Courier</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>5.14</td></tr>
<tr><td>PRIORITY:</td><td>Medium</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case narrative describes the process in which the administrator wishes to delete a courier from the system. The administrator will search for the required courier selected for deletion. The administrator will then confirm the deletion of the courier and the use case concludes when the system notifies the administrator of a successful deletion.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the administrator wishes to delete a courier.</td></tr>
<tr><td rowspan="5">TYPICAL COURSE OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wishes to delete a courier.</td><td colspan="2">Step 2: The system invokes Use Case 5.12 “Search Courier”</td></tr>
<tr><td>Step 3: The administrator selects the desired Courier.</td><td colspan="2">Step 4: The system retrieves the selected courier’s details from the Courier entity with the following attributes
• CourierName
• CourierFee
• EstimatedDeliveryTime

The system prompts the administrator to confirm the deletion of the courier selected.</td></tr>
<tr><td>Step 5: The administrator confirms the deletion of the courier from the system. [ALT]</td><td colspan="2">Step 6: The system removes all of the details related of the selected Courier and the following attributes:

• ID
• CourierName
• CourierFee
• EstimatedDeliveryTime
[ALT]</td></tr>
<tr><td></td><td colspan="2">Step 7: The system informs the administrator of the successful deletion of the courier. [ALT]</td></tr>
</table>

ALTERNATE COURSES:	**[ALT] Step 5:** The administrator does not confirm the deletion of the courier. • Terminate this use case.
	[ALT] Step 6: An existing record of an active order has the courier assigned to it. • The system restricts the deletion of the courier to prevent null values. • Go to ALT step 6.
	[ALT] Step 7: The system informs the administrator of the unsuccessful deletion of the courier.
CONCLUSION:	This use case concludes when the administrator is notified of the successful deletion of the courier.
POST-CONDITION:	• The system deletes all of the details related to the selected courier from the **Courier entity**. • The system removes all of the details related to the selected courier.
BUSINESS RULES	• Only the administrator can delete a courier.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 47 - 5.15 Search Placed Order

ORDER SUBSYSTEM		
AUTHORS (s): Nomusa Vumisa	**DATE: 11/07/2022**	
VERSION: 1.0	**LAST REVIEW DATE: 11/07/2022**	
USE CASE NAME:	Search Placed Order	**USE CASE TYPE**
USE CASE ID:	5.15	Business Requirements: ☐
PRIORITY:	Medium	System Analysis: ☒
SOURCE:	ByteXpress Requirements List	System Design: ☐
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	Actor Action	System Response
	Step 1: The customer/reseller wants to search for an order	
	Step 2: The customer/reseller enters the search parameter	**Step 3:** The system searches for a match using the following attributes: From the **User** table. • EmailAddress From the **Order** table • OrderId • FirstName • Surname • PhoneNumber • OrderDate • OrderStatus • OrderTotal
		Step 4: The system displays the search results **[ALT]**
ALTERNATE COURSES:	**ALT Step 4**: The search query entered does not match any existing instances of an order. • Return to Step 2 or terminate the use case	

CONCLUSION:	This use case concludes when the desired order details are displayed on the system.
POST-CONDITION:	
BUSINESS RULES	• None
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 48 - 5.16 View Placed Order

<table>
<tr><th colspan="4">ORDER SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Nomusa Vumisa</td><td colspan="2">DATE: 11/07/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 11/07/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">View Placed Order</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">5.16</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Customer/Reseller</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The customer/reseller must be logged on the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use case is called when the customer/reseller wants to view their order details</td></tr>
<tr><td rowspan="3">TYPICAL COURSE</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The customer/reseller wants to view their order details</td><td colspan="2">Step 2: The system invokes Use Case 5.15 Search Placed Orders</td></tr>
<tr><td>Step 3: The customer/reseller selects the order they wish to view</td><td colspan="2">Step 4: The system displays the selected order details using the following attributes:

From the User table
• EmailAddress

From the Order table
• OrderId
• FirstName
• Surname
• PhoneNumber
• StreetAddress
• OrderDate
• OrderStatus
• OrderTotal
• ParcelTrackingNumber
• DispatchedDate</td></tr>
</table>

		From the **Suburb** Table • SuburbName • PostalCode From the **City** Table • CityName From the **Province** Table • ProvinceName From the **Country** Table • CountryName From the **Courier** Table • CourierName From the **Product** table • Name From the **OrderLine** table • OrderItemPrice • OrderItemCount
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	• None	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• The customer/reseller cannot view their order details without an internet connection	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 49 - 5.17 Cancel Placed Order

ORDER SUBSYSTEM			
AUTHORS (s): Nomusa Vumisa		**DATE: 07/11/2022**	
VERSION: 1.0		**LAST REVIEW DATE: 07/11/2022**	
USE CASE NAME:	Cancel Placed Order	**USE CASE TYPE**	
USE CASE ID:	5.17	Business Requirements: ☐	
PRIORITY:	Medium	System Analysis: ☒	
SOURCE:	ByteXpress Requirements List	System Design: ☐	
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 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	Actor Action	System Response	
	Step 1: The reseller wants to cancel an order.	**Step 2:** The system invokes Use Case **5.16 View Placed Order**	
	Step 3: The reseller selects the cancel option **[ALT]**	**Step 4:** The system prompts the reseller to confirm the cancellation of the reseller order	
	Step 5: The reseller confirms the cancellation of the reseller order. **[ALT]**	**Step 6:** The system updates the order status to "Cancelled" using the following attribute: From the **Order** table OrderStatus	
		Step 7: The system informs the reseller of the successful cancellation and sends them an email confirming the order cancellation using the following attributes: From the **User** table • FirstName • EmailAddress	

		From the **Order** table • OrderId • OrderDate PaymentStatus
ALTERNATE COURSES:	**ALT Step 3**: The reseller does not want to cancel their order • Select the back option • Terminate this use case.	
	ALT Step 5: The reseller does not confirm the cancellation of their order. • Select the cancel option • 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 CONTRAINTS AND SPECIFICATIONS	• Only orders with a Pending status can be cancelled •	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 50- 5.18 Cancel Reseller Order

ORDER SUBSYSTEM			
AUTHORS (s): Nomusa Vumisa		**DATE: 07/11/2022**	
VERSION: 1.0		**LAST REVIEW DATE: 07/11/2022**	
USE CASE NAME:	Cancel Reseller Order	**USE CASE TYPE**	
USE CASE ID:	5.18	Business Requirements: ☐	
PRIORITY:	Medium	System Analysis: ☒	
SOURCE:	ByteXpress 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	Actor Action	System Response	
	Step 1: The administrator or sales assistant wants to cancel an order.	**Step 2:** The system invokes Use Case **5.7 View Order Overview**.	
	Step 3: The administrator or sales assistant selects the cancel option **[ALT]**	**Step 4:** The system prompts the administrator to confirm the cancellation of the reseller order	
	Step 5: The administrator or sales assistant confirms the cancellation of the reseller order. **[ALT]**	**Step 6:** The system updates the order status to "Cancelled" using the following attribute: From the **Order** table • OrderStatus	

		Step 7: The system informs the administrator of the successful cancellation and sends an email to the reseller informing them of the order cancellation using the following attribute: From the **User** table • FirstName • EmailAddress From the **Order** table • OrderId • OrderDate • PaymentStatus
ALTERNATE COURSES:	**ALT Step 3**: The administrator or sales assistant does not want to cancel the reseller order • Select the back option • Terminate this use case.	
	ALT Step 5: The administrator or sales assistant does not confirm the cancellation of the reseller order. • Select the cancel option • 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 CONTRAINTS AND SPECIFICATIONS	• Only the administrator and sales assistant can cancel an order. •	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 51- 5.19 Approve/Reject Reseller Order

<table>
<tr><th colspan="4">ORDER SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Nomusa Vumisa</td><td colspan="2">DATE: 07/11/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 07/11/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Approve/Reject Reseller Order</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">5.19</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">Medium</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• Reseller</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.
• The order status must be “Pending”</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the administrator wants to approve or reject an order on the system.</td></tr>
<tr><td rowspan="5">TYPICAL COURSE</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wants to approve/reject an order</td><td colspan="2">Step 2: The system invokes Use Case 5.7 View Orders Overview.</td></tr>
<tr><td>Step 3: The administrator selects the approve option [ALT]</td><td colspan="2">Step 4: The system prompts the administrator to confirm the approval/rejection of the reseller order</td></tr>
<tr><td>Step 5: The administrator confirms the approval/rejection of the reseller order. [ALT]</td><td colspan="2">Step 6: The system updates the order status to “Processing” if approved or “Rejected” using the following attribute:

From the Order table
• OrderStatus</td></tr>
<tr><td></td><td colspan="2">Step 7: The system informs the administrator of the successful approval/rejection and sends an email to the reseller informing them of their</td></tr>
</table>

		order status using the following attribute: From the **User** table • FirstName • EmailAddress From the **Order** table • OrderId • OrderDate • PaymentStatus
ALTERNATE COURSES:	**ALT Step 3A**: The administrator selects the reject option • Proceed to Step 4	
	ALT Step 3B: The administrator does not want to approve or reject the order • Select the back option • The use case is terminated	
	ALT Step 5: The administrator does not confirm the approval/rejection of the reseller order. • Select the cancel option • 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 CONTRAINTS 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 52- 5.20 Dispatch Order

ORDER SUBSYSTEM		
AUTHORS (s): Nomusa Vumisa		**DATE: 07/11/2022**
VERSION: 1.0		**LAST REVIEW DATE: 07/11/2022**
USE CASE NAME:	Dispatch Order	**USE CASE TYPE** Business Requirements: ☐ System Analysis: ☒ System Design: ☐
USE CASE ID:	5.20	
PRIORITY:	Medium	
SOURCE:	ByteXpress Requirements List	
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	Actor Action	System Response
	Step 1: The administrator or sales assistant wants to dispatch an order	**Step 2:** The system invokes Use Case **5.7 View Orders Overview**
	Step 3: The administrator or sales assistant enters the parcel tracking number **[ALT]**	**Step 4:** The system captures and validates the entered details • **[ALT]**
		Step 5: The system prompts the administrator to confirm the dispatching of the order
	Step 6: The administrator or sales assistant confirms the dispatching of the order. **[ALT]**	**Step 7:** The system updates the order status to "Dispatched" and saves the tracking number using the following attributes:

		From the **Order** table • OrderStatus • PatcelTrackingNumber
		Step 8: The system informs the administrator of the successful status update and sends a SMS to the recipient, informing them that their order has been dispatched using the following attributes: From the **Order** table • PhoneNumber • FirstName • OrderId • ParcelTrackingNumber
ALTERNATE COURSES:	ALT **Step 3**: The administrator or sales assistant does not want to dispatch the order • Select the back option • The use case is terminated	
	ALT **Step 4:** The required field is empty • The system displays a validation error message • Return to step 3	
	ALT **Step 6**: The administrator or sales assistant does not confirm the dispatching of the order • Select the cancel option. • 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 CONTRAINTS AND SPECIFICATIONS	• Only the administrator and sales assistant can dispatch an order	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 53- 5.21 Log Back Order

Order Subsystem			
AUTHORS (s): Kyle van Eeden		**DATE: 21/08/2022**	
VERSION: 1.0		**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:	ByteXpress Requirements List	System Design:	☐
PRIMARY BUSINESS ACTOR	• Administrator		
PRIMARY SYSTEM ACTOR	• None		
OTHER PARTICIPATING ACTORS:	• Reseller (ERA)		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	This use case describes the process in which the administrator wishes to back-log a reseller order placed on the system. Once the administrator requests the system to log the back-log, the system will prompt the administrator to confirm that whether they wish to proceed with logging the back-order request. After confirming to proceed, the system will update the status of the order to reflect it has been back-logged. An email is also generated and sent to the reseller which informs the reseller who placed the order that their order has been backlogged. The use case concludes with the system displaying a confirmation alert which informs the administrator that the order has been backlogged successfully.		
	The administrator must be logged into the system. Use case 5.7 The reseller order must have a status of "Pending" assigned.		
TRIGGER:	The administrator wishes to delay a reseller order on the system.		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: The administrator requests the system to backlog the reseller order due to insufficient stock availability.	**Step 2**: The system prompts the administrator to confirm whether they wish to proceed with adding the specific order to the orders backlog.	

	Step 3: The administrator selects the option provided to confirm to backlog the order. **[ALT]**	**Step 4:** The system updates the order's status to "Delayed" by updating the following attributes in the **Order** entity: • OrderStatus • BackOrderDate
		Step 5: The system generates an email which is populated with the following details retrieved for the order from the below-mentioned entities: From the **User** entity: • FirstName • EmailAddress From the **Order** entity: • OrderId • OrderStatus • OrderPaymentStatus • OrderDate From the **ConfirmationReminder** entity: • Value
		Step 6: The system sends the email to the reseller's email address who has placed the order.

		Step 7: The system displays a confirmation alert which informs the administrator that the order backlog has been performed successfully.
ALTERNATE COURSES:	**ALT** Step 3: The administrator does not wish to proceed with adding the order to the orders backlog: • The use case concludes.	
CONCLUSION:	The system displays a confirmation alert to confirm that the order has been backlogged successfully.	
POST-CONDITION:	• The backlogged order's details is sent to the reseller's email address. • The order has a status of "Delayed" assigned.	
BUSINESS RULES	• Only the administrator of the system should be allowed to add orders to the orders backlog list. • Only reseller orders which is in a "Pending" state can be backlogged. • The reseller should be informed that their order has been delayed.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• An email should be sent to the reseller which informs them that their placed order has been backlogged by Natuurlik. • Only "Pending" reseller orders can be added to the orders backlog by the administrator.	
ASSUMPTIONS:	• The administrator has access to the internet and the system is not offline.	
OPEN ISSUES:	• None	

Table 54- 5.22 Confirm Order

Order Subsystem			
AUTHORS (s): Kyle van Eeden		**DATE: 21/08/2022**	
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:	ByteXpress Requirements List	System Design:	☐
PRIMARY BUSINESS ACTOR	• Reseller		
PRIMARY SYSTEM ACTOR	• None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	This use case describes the process in which a reseller confirms an order that has been backlogged on the system. The use case begins with reseller requesting the system to confirm the order by selecting the relevant option provided. The system then prompts the reseller to confirm whether they wish to accept the delayed order and once the reseller confirms, the system will update the order's status and the system sends an order confirmation email to the reseller's email address. The use case concludes once the system displays a confirmation alert which informs the reseller that their order has been successfully confirmed.		
PRE-CONDITION:	The reseller must be logged into the system. The order to be confirmed must have an order status of "Delayed" assigned. Use Case 5.7		
TRIGGER:	The reseller wishes to confirm an order which has been added to the orders backlog.		
TYPICAL COURSE	Actor Action	System Response	

OF EVENTS:	**Step 1**: The reseller requests the system to confirm their backlogged order.	**Step 2**: The system prompts the reseller to confirm whether they wish to proceed with accepting the delayed order.
	Step 3: The reseller selects the option to accept the delayed order. **[ALT]**	**Step 4:** The system updates the following attribute for the specific reseller order in the **Order** entity: • OrderStatus Upon having confirmed the backlogged order, the system will assign the "Processing" status value to the order.
		Step 5: The system generates an email using the following details retrieved from the respective entities: From the **User** entity: • FirstName • Email Address From the **Order** entity: • OrderID • OrderStatus • OrderPaymentStatus • OrderDate
		Step 6: The system sends the generated email to the reseller's email address.
		Step 7: The system displays a confirmation alert which informs the reseller that the backlogged order has been confirmed successfully.
ALTERNATE COURSES:	**ALT** Step 3: The reseller no longer wishes to confirm the backlogged order:	

	• The reseller selects the option provided to cancel the order confirmation operation. • The use case concludes.
CONCLUSION:	The system displays a confirmation alert to inform the reseller that their order has been confirmed successfully.
POST-CONDITION:	• The reseller order's status is "Processing" after having confirmed a modified order. • An email is sent to the reseller's email address which informs them that the delayed order has been accepted successfully.
BUSINESS RULES	• A reseller should only be allowed to confirm an order placed by them, and which has been placed in the orders backlog list by Natuurlik. • The reseller only has 7 days to accept or reject the delayed order before it being rejected automatically by the system.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• Only "Delayed" reseller orders can be confirmed/accepted by the reseller. • An email is to be sent to the reseller's email address, which confirms that the order has been accepted successfully and is being processed by Natuurlik.
ASSUMPTIONS:	• The reseller has access to the internet and the system is not offline. • The reseller has reviewed the details of the backlogged order and wishes to accept it.
OPEN ISSUES:	• None

Table 55- 5.23 Reject Order

<table>
<tr><th colspan="4">Order Subsystem</th></tr>
<tr><td colspan="2">AUTHORS (s): Kyle van Eeden</td><td colspan="2">DATE: 25/08/2022</td></tr>
<tr><td colspan="2">VERSION: 1.1</td><td colspan="2">LAST REVIEW DATE: 29/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Reject Order</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">5.23</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Reseller</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process in which a reseller rejects an order that has been placed on the orders backlog after being placed. The use case begins with the reseller requesting the system to reject the order by selecting the relevant option provided. The system then prompts the reseller to confirm whether they wish proceed with rejecting the delayed order. Once the reseller confirms that they wish to reject the order, the system will update the order’s status to “Rejected” and the details is saved in the Order entity. The use case concludes once the system displays a confirmation alert which informs the reseller that their order has been rejected successfully.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The reseller must be logged into the system.
The order to be rejected must have an order status of “Delayed” assigned.
Use Case 5.7</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The reseller wishes to reject their backlogged order on the system.</td></tr>
<tr><td>TYPICAL COURSE</td><td colspan="2">Actor Action</td><td>System Response</td></tr>
</table>

OF EVENTS:	**Step 1**: The reseller requests the system to reject the backlogged order.	**Step 2**: The system prompts the reseller to confirm whether they wish to proceed with the order rejection operation.
	Step 3: The reseller selects the option to proceed to reject the backlogged order. **[ALT]**	**Step 4:** The system updates the following attribute for the specific reseller order in the **Order** entity: • OrderStatus Upon having rejected the modified order, the system will assign the "Rejected" status to the order.
		Step 5: The system displays a confirmation alert which informs the reseller that the backlogged order has been rejected successfully.
ALTERNATE COURSES:	**ALT** Step 3: The reseller no longer wishes to reject the delayed order: • The reseller selects the option provided to cancel the order rejection operation. • The use case concludes.	
CONCLUSION:	The system displays a confirmation alert to inform the reseller that their order has been rejected successfully on the system.	
POST-CONDITION:	• The reseller order's status is changed to "Rejected" after having rejected a modified order and is saved to the **Order** entity.	
BUSINESS RULES	• A reseller should only be allowed to reject an order placed by them, and which has been placed on the orders backlog by Natuurlik. • The reseller should be given the option to either accept or reject an order which has been assigned to the orders backlog.	
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• Only orders with a "Delayed" status can be rejected by a reseller using this functionality.	
ASSUMPTIONS:	• The reseller has access to the internet and the system is not offline. • The reseller has reviewed the order backlog email sent by Natuurlik and wishes to reject the order on the system.	
OPEN ISSUES:	• None	

Table 56- 5.24 Send Order Confirmation Reminder

ORDER SUBSYSTEM		
AUTHORS (s): Nomusa Vumisa	**DATE: 27/08/2022**	
VERSION: 1.1	**LAST REVIEW DATE: 27/08/2022**	
USE CASE NAME:	Send Order Confirmation Reminder	**USE CASE TYPE**
USE CASE ID:	5.24	Business Requirements: ☐
PRIORITY:	High	System Analysis: ☒
SOURCE:	ByteXpress Requirements List	System Design: ☐
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	Actor Action	System Response
	Step 1: The confirmation threshold has been reached	**Step 2:** The system reads the following attributes to verify that the order status is "Delayed" and if the due date threshold has been reached From the **Order** table • OrderStatus • BackOrderDate From the **ConfirmationReminder** table • Value

<table>
<tr><td rowspan="4"></td><td></td><td></td></tr>
<tr><td></td><td>Step 3: The system generates a confirmation reminder email using the following attributes:

From the User table
• Email
• FirstName

From the Order table
• OrderId
• OrderStatus

From the ConfirmationReminder table
• Value</td></tr>
<tr><td></td><td>Step 4: The system sends a confirmation reminder to the reseller</td></tr>
<tr><td colspan="2"></td></tr>
<tr><td>ALTERNATE COURSES:</td><td colspan="2"></td></tr>
<tr><td>CONCLUSION:</td><td colspan="2">This use case concludes when the reminder has been sent to the reseller</td></tr>
<tr><td>POST-CONDITION:</td><td colspan="2">The reseller is reminded to confirm their order using attributes retrieved from the User and Order tables</td></tr>
<tr><td>BUSINESS RULES</td><td colspan="2">None</td></tr>
<tr><td>IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS</td><td colspan="2">• Only the reseller should be notified.
• Reminders must be sent via email.</td></tr>
<tr><td>ASSUMPTIONS:</td><td colspan="2">• None</td></tr>
<tr><td>OPEN ISSUES:</td><td colspan="2">• None</td></tr>
</table>

Table 57- 5.25 Update Order Confirmation Time

<table>
<tr><th colspan="4">ORDER SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Nomusa Vumisa</td><td colspan="2">DATE: 28/08/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 28/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Update Order Confirmation Time</td><td>USE CASE TYPE</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">5.25</td><td>Business Requirements: ☐</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td><td>System Analysis: ☒</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td><td>System Design: ☐</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The administrator must be logged onto the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use case is triggered when the administrator wants to update the order confirmation time</td></tr>
<tr><td rowspan="4">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wishes to update the reseller order confirmation time</td><td colspan="2">Step 2: The system retrieves the following attribute from the ConfirmationReminder table:
• Days</td></tr>
<tr><td>Step 3: The administrator selects their desired order confirmation time interval</td><td colspan="2">Step 4: The system requests the administrator to confirm the updating of the order confirmation time</td></tr>
<tr><td>Step 5: The administrator confirms the updating of the order confirmation time

[ALT]</td><td colspan="2">Step 6: The system reads from the ConfirmationReminder table to ensure that the record being updated is not already active using the following attribute:</td></tr>
</table>

		• IsActive **[ALT]**
		Step 7: The system activates the selected order confirmation time using the following attributes: From the **ConfirmationReminder** Table • IsActive
		Step 8: The system displays a successful creation message
ALTERNATE COURSES:	**ALT Step 5**: The administrator does not confirm the updating of the order confirmation time • The use case is terminated	
	ALT Step 6: The order confirmation time is already active on the system • The system displays an error message • The use case is terminated	
CONCLUSION:	This use case concludes when the order confirmation time is updated.	
POST-CONDITION:	The order confirmation time was updated in the ConfirmationReminder Table	
BUSINESS RULES	• Only the administrator can update the order confirmation time	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• Only one order confirmation time can be active at a time • The order confirmation time cannot be updated without an internet connection	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

2.6. Query Reason Subsystem

Table 58 - 6.1 Create Query Reason

<table>
<tr><th colspan="4">QUERY REASON SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Aphiwe Shozi</td><td colspan="2">DATE: 14/05/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 14/05/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Create Query Reason</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">6.1</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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 proceed to capture the Query Reason details and save it to the system database.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.
• The Query Reason must not already exist on the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use case is triggered when the administrator wants to create a new Query Reason.</td></tr>
<tr><td>TYPICAL COURSE</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
</table>

OF EVENTS:	**Step 1**: The administrator wants to create a new Query Reason.	**Step 2**: The system requires the admin to add information about the Query Reason in the **Query Reason Table** such as : • Name
	Step 3: The administrator provides the required the details. **[ALT]**	**Step 4**: The system captures and validates the entered details **[ALT]**
		Step 5: The system reads from the **Query Reason** table to ensure that the record being added doesn't match any existing record using the following attribute: • Id **[ALT]**
		Step 6: The system saves the new Query Reason details which have the following attributes: From the **Query Reason Table** • Id • Name
		Step 7: The system save the new Query Reason record to the **Query Reason table**
ALTERNATE COURSES:	**ALT Step 3**: The administrator no longer wants to add a Query Reason • The use case is terminated **ALT Step 4:** The entered details are not in the correct format or the required fields are empty • The system displays a validation error message • Return to step 3 **ALT Step 5:** The Query Reason already exists on the system	

	• The system displays a validation error message • The use case is terminated
CONCLUSION:	This use case concludes when a new Query Reason is created.
POST-CONDITION:	• The Query Reason details are added to the database.
BUSINESS RULES	• Only the administrator can add a new Query Reason.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 59 - 6.2 Search Query Reason

QUERY REASON SUBSYSTEM			
AUTHORS (s): Aphiwe Shozi		**DATE: 14/05/2022**	
VERSION: 1.0		**LAST REVIEW DATE: 14/05/2022**	
USE CASE NAME:	Search Query Reason	**USE CASE TYPE**	
USE CASE ID:	6.2	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 in the administrator wishes to search through the existing Query Reasons on the system. The administrator can use the Query Reason details for a search query and the system will retrieve the Query Reason that matches		
PRE-CONDITION:	• The administrator must be logged onto the system. • The administrator can only search through Query Reasons that are saved on 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	
	Step 1: The administrator wants to search for a Query Reason.		
	Step 2: The administrator enters the search parameter.	**Step 3:** The system searches using the following attributes in the **Query Reason** table:	

		• Name
		Step 4: The system displays the search results **[ALT]**
ALTERNATE COURSES:	**ALT Step 4**: The search query entered does not match any existing instances of Query Reasons in the system database. • Return to Step 2 or terminate the use case	
CONCLUSION:	This use case concludes when the desired Query Reason's details are displayed on the system.	
POST-CONDITION:	The **Query Reason** Table was searched for and an entry that corresponds to the search query or parameter entered.	
BUSINESS RULES	• None	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 60 - 6.3 Update Query Reason

QUERY REASON SUBSYSTEM			
AUTHORS (s): Aphiwe Shozi		**DATE: 14/05/2022**	
VERSION: 1.0		**LAST REVIEW DATE: 14/05/2022**	
USE CASE NAME:	Update Query Reason	**USE CASE TYPE**	
USE CASE ID:	6.3	Business Requirements: ☐	
PRIORITY:	High	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 in which the administrator wishes to update the details of Query Reason to the system. The use case begins when the administrator wishes to update the details of an existing Query Reason record .		
PRE-CONDITION:	• The administrator must be logged onto the system. • Only Query Reasons that exists on the system can be updated		
TRIGGER:	The use is triggered when the administrator updates the details of a Query Reason that already exists on the system.		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: The administrator wishes to update a Query Reason's details	**Step 2:** The system invokes Use Case **11.2 Search Query Reason**.	
	Step 3: The administrator selects the desired Query Reason	**Step 4**: The system retrieves the desired Query Reasons' details from the following tables	

		From the **Query Reason** Table • Name
	Step 5: The administrator enters the updated details. **[ALT]**	**Step 6**: The system captures and validates the entered details **[ALT]**
		Step 7: The system requests the administrator to confirm the updating of the Query Reason details
	Step 8: The administrator confirms the updating of the Query Reason details **[ALT]**	**Step 9:** The system reads from the **Query Reason** table to ensure that the record being updated does not match any existing records using the following attribute: • Id **[ALT]**
		Step 10: The system saves the updated Query Reason details which have the following attributes: From the **Query Reason** Table • Name
		Step 11: The system displays a successful update message
ALTERNATE COURSES:	**ALT Step 5**: The administrator no longer wants to update a Query Reason • The use case is terminated **ALT Step 6:** The entered details are not in the correct format or the required fields are empty • The system displays a validation error message • Return to step 5	

	ALT Step 8: The administrator does not confirm updating the Query Reason • Return to Step 5 or terminate the use case **ALT Step 9:** The Query Reason already exists on the system • The system displays a validation error message • The use case is terminated
CONCLUSION:	This use case concludes when the details of the respective Query Reason record has been updated successfully on the system
POST-CONDITION:	• The Query Reason details are updated on the database.
BUSINESS RULES	• Only the administrator can update a Query Reason.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 61 - 6.4 Delete Query Reason

<table>
<tr><th colspan="4">QUERY REASON SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Aphiwe Shozi</td><td colspan="2">DATE: 14/05/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 14/05/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>Delete Query Reason</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>6.4</td></tr>
<tr><td>PRIORITY:</td><td>Medium</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The administrator must be logged in to the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the administrator wishes to delete a Query Reason.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wishes to delete a Query Reason.</td><td colspan="2">Step 2: The system invokes Use Case 11.2 Search Query Reason.</td></tr>
<tr><td>Step 3: The administrator selects the desired Query Reason</td><td colspan="2">Step 4: The system retrieves the desired Query Reasons’ details from the following tables

From the Query Reason Table
• Name</td></tr>
</table>

	Step 5: The administrator selects the delete option **[ALT]**	**Step 6:** The system requests the administrator to confirm the deletion of the Query Reason.
	Step 7: The administrator confirms the deletion of the Query Reason from the system. **[ALT]**	**Step 8:** The system removes all the details related to the Query Reason from the database from the following tables: From the **Query Reason** Table • Name
		Step 9: The system informs the administrator of the successful deletion of the Query Reason
ALTERNATE COURSES:	**ALT Step 5**: The administrator does not want to delete the Query Reason and selects the cancel option. • Terminate this use case. **ALT Step 7**: The administrator does not confirm the deletion of the Query Reason. • Return to step 4	
CONCLUSION:	This use case concludes when the administrator is notified of the successful deletion of the Query Reason.	
POST-CONDITION:	• The system deletes all the details related to the selected Query Reason from the **Query Reason table**.	
BUSINESS RULES	• Only the administrator can delete a Query Reason.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 62 6.5 Review Order Query

ORDER SUBSYSTEM			
AUTHORS (s): Ofhani Mungani		**DATE: 07/28/2022**	
VERSION: 1.1		**LAST REVIEW DATE: 07/28/2022**	
USE CASE NAME:	View Order Query	**USE CASE TYPE**	
USE CASE ID:	6.5	Business Requirements:	☐
PRIORITY:	High	System Analysis:	☒
SOURCE:	ByteXpress Requirements List	System Design:	☐
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 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	Actor Action	System Response	
	Step 1: The administrator or sales assistant wishes to view an order query on the system	**Step 2:** The system retrieves the following details for all order query: from the **OrderQuery** entity: • Date • Description	

		• Status from the **Order** entity: • OrderId from the **QueryReason** entity: • Name
	Step 3: The administrator or sales assistant selects an option to review an order query.	**Step 4:** The system requires the administrator to provide feedback to an order query in the **OrderQuery** entity on the following attribute: • Feedback
	Step 5: The administrator or assistant inputs the required details **[ALT]**	**Step 6**: The system captures and validates the information provided by the administrator ensuring the format is adhered to and no required text fields are left empty. **[ALT]**
		Step 7: The system saves the information to the **OrderQuery** entity on the database and update the order query status attribute from pending review to null.
		Step 8: The system displays a success message on the screen to confirm the submission of the feedback on the order query
ALTERNATE COURSES:	ALT **Step 5**: The administrator/sales assistant doesn't want to provide feed back to an order query • Terminate use case	
	ALT **Step 6**: The system fails to validate the information provided because of a missing field. • Return to Step 5.	
CONCLUSION:	The use case concludes when the feedback for an order query is added on the system.	
POST-CONDITION:	The customer or reseller receives feedback on the system about their order query	

BUSINESS RULES	• None.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• The administrator or sales assistant cannot view an order query without an internet connection. • Only the administrator or sales assistant can query an order.
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 63- 6.6 View Order Query Feedback

<table>
<tr><th colspan="4">ORDER QUERY SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Thenjiwe Ntsonda</td><td colspan="2">DATE: 07/29/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 07/2392022</td></tr>
<tr><td>USE CASE NAME:</td><td>View Order Query Feedback</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>6.6</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Customer or Reseller</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• Customer or reseller must have submitted an order query prior
• Customer or reseller must be logged in</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when a customer or reseller wishes to view the feedback on a query they submitted on the system.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The customer or reseller wishes to view the feedback on a query they submitted on the system</td><td colspan="2">Step 2: Invoke use case 1.3 View Profile Details</td></tr>
<tr><td>Step 3: The customer or reseller selects the desired query they submitted</td><td colspan="2">Step 4: The system retrieves the following from the OrderQuery table and displays the order query details:
• Description
• QueryFeedback</td></tr>
</table>

		From the **Order** table: • **Order Number** From the **QueryReason** table: • **Query Reason**
ALTERNATE COURSES:	• None	
CONCLUSION:	The system displays the selected order query details to the customer or reseller	
POST-CONDITION:	The OrderQuery, Order and Query Reason 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 CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• The customer and reseller have a stable internet connection.	
OPEN ISSUES:	• None	

2.7. Review Subsystem

Table 64 - 7.1 Create Review Reason

<table>
<tr><th colspan="4">REVIEW REASON SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Aphiwe Shozi</td><td colspan="2">DATE: 14/05/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 14/05/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>Create Review Reason</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>7.1</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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 proceed to capture the Review Reason details and save it to the system database.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.
• The Review Reason must not already exist on the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use case is triggered when the administrator wants to create a new Review Reason.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wants to create a new Review Reason.</td><td colspan="2">Step 2: The system requires the admin to add information about the Review Reason in the <u>Review Reason Table</u> such as :
• Name</td></tr>
<tr><td>Step 3: The administrator provides the required the details.</td><td colspan="2">Step 4: The system captures and validates the entered details
[ALT]</td></tr>
</table>

		Step 5: The system reads from the Review Reason table to ensure that the record being added doesn't match any existing record using the following attribute: • Id **[ALT]**
		Step 6: The system saves the new Review Reason details which have the following attributes: From the Review Reason Table • Id • Name
		Step 7: The system save the new Review Reason record to the Review Reason table
ALTERNATE COURSES:	**ALT Step 3**: The administrator no longer wants to add a Review Reason • The use case is terminated **ALT Step 4:** The entered details are not in the correct format or the required fields are empty • The system displays a validation error message • Return to step 3 **ALT Step 5:** The Review Reason already exists on the system • The system displays a validation error message • The use case is terminated	
CONCLUSION:	This use case concludes when a new Review Reason is created.	
POST-CONDITION:	• The Review Reason details are added to the database.	
BUSINESS RULES	• Only the administrator can add a new Review Reason.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 65 -7.2 Search Review Reason

<table>
<tr><th colspan="4">REVIEW REASON SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Aphiwe Shozi</td><td colspan="2">DATE: 14/05/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 14/05/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Search Review Reason</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">7.2</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">Medium</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process in the administrator wishes to search through the existing Review Reasons on the system. The administrator can use the Review Reason details for a search Review and the system will retrieve the Review Reason that matches</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.
• The administrator can only search through Review Reasons that are saved on the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the administrator wants to search for a Review Reason on the system.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wants to search for a Review Reason.</td><td colspan="2"></td></tr>
<tr><td>Step 2: The administrator enters the search parameter.</td><td colspan="2">Step 3: The system searches using the following attributes in the Review Reason table:

• Name</td></tr>
</table>

		Step 4: The system displays the search results **[ALT]**
ALTERNATE COURSES:	**ALT Step 4**: The search Review entered does not match any existing instances of Review Reasons in the system database. • Return to Step 2	
CONCLUSION:	This use case concludes when the desired Review Reason's details are displayed on the system.	
POST-CONDITION:	The **Review Reason** Table was searched for and an entry that corresponds to the search Review or parameter entered.	
BUSINESS RULES	• None	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 66 - 7.3 Update Review Reason

<table>
<tr><th colspan="4">REVIEW REASON SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Aphiwe Shozi</td><td colspan="2">DATE: 14/05/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 14/05/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>Update Review Reason</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>7.3</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process in which the administrator wishes to update the details of Review Reason to the system. The use case begins when the administrator wishes to update the details of an existing Review Reason record .</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.
• Only Review Reasons that exists on the system can be updated</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use is triggered when the administrator updates the details of a Review Reason that already exists on the system.</td></tr>
<tr><td rowspan="2">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wishes to update a Review Reason’s details</td><td colspan="2">Step 2: The system invokes Use Case 11.2 Search Review Reason.</td></tr>
</table>

	Step 3: The administrator selects the desired Review Reason	**Step 4**: The system retrieves the desired Review Reasons' details from the following tables From the **Review Reason** Table • Name
	Step 5: The administrator enters the updated details. **[ALT]**	**Step 6**: The system captures and validates the entered details **[ALT]**
		Step 7: The system requests the administrator to confirm the updating of the Review Reason details
	Step 8: The administrator confirms the updating of the Review Reason details **[ALT]**	**Step 9:** The system reads from the **Review Reason** table to ensure that the record being updated does not match any existing records using the following attribute: • Id **[ALT]**
		Step 10: The system saves the updated Review Reason details which have the following attributes: From the **Review Reason** Table • Name
		Step 11: The system displays a successful update message
ALTERNATE COURSES:	**ALT Step 5**: The administrator no longer wants to update a Review Reason • The use case is terminated **ALT Step 6:** The entered details are not in the correct format or the required fields are empty • The system displays a validation error message • Return to step 5	

	ALT Step 8: The administrator does not confirm updating the Review Reason • Return to Step 5 or terminate the use case **ALT Step 9:** The Review Reason already exists on the system • The system displays a validation error message • The use case is terminated
CONCLUSION:	This use case concludes when the details of the respective Review Reason record has been updated successfully on the system
POST-CONDITION:	• The Review Reason details are updated on the database.
BUSINESS RULES	• Only the administrator can update a Review Reason.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 67 - 7.4 Delete Review Reason

<table>
<tr><th colspan="4">REVIEW REASON SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Aphiwe Shozi</td><td colspan="2">DATE: 14/05/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 14/05/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>Delete Review Reason</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>7.4</td></tr>
<tr><td>PRIORITY:</td><td>Medium</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The administrator must be logged in to the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the administrator wishes to delete a Review Reason.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wishes to delete a Review Reason.</td><td colspan="2">Step 2: The system invokes Use Case 11.2 Search Review Reason.</td></tr>
<tr><td>Step 3: The administrator selects the desired Review Reason</td><td colspan="2">Step 4: The system retrieves the desired Review Reasons’ details from the following tables</td></tr>
</table>

		From the **Review Reason** Table • Name
	Step 5: The administrator selects the delete option **[ALT]**	**Step 6:** The system requests the administrator to confirm the deletion of the Review Reason.
	Step 7: The administrator confirms the deletion of the Review Reason from the system. **[ALT]**	**Step 8:** The system removes all the details related to the Review Reason from the database from the following tables: From the **Review Reason** Table • Name
		Step 9: The system informs the administrator of the successful deletion of the Review Reason
ALTERNATE COURSES:	**ALT Step 5**: The administrator does not want to delete the Review Reason and selects the cancel option. • Terminate this use case. **ALT Step 7**: The administrator does not confirm the deletion of the Review Reason. • Return to step 4	
CONCLUSION:	This use case concludes when the administrator is notified of the successful deletion of the Review Reason.	
POST-CONDITION:	• The system deletes all the details related to the selected Review Reason from the **Review Reason table**.	
BUSINESS RULES	• Only the administrator can delete a Review Reason.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	

OPEN ISSUES:	• None

Table 68 -7.5 View Order Review

<table>
<tr><th colspan="4">REVIEW SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Ofhani Mungani</td><td colspan="2">DATE: 07/28/2022</td></tr>
<tr><td colspan="2">VERSION: 1.1</td><td colspan="2">LAST REVIEW DATE: 07/28/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>View Order Review</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>7.5</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator/Sales Assistant</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator or sales assistant must be logged on the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use case is called when the administrator or sales assistant wants to view an order review</td></tr>
<tr><td rowspan="2">TYPICAL COURSE</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator or sales assistant wishes to view an order review on the system</td><td colspan="2">Step 2: The system retrieves the following details for all order reviews:

from the OrderReview entity:
• Date
• Rating
• Description

from the Order entity:
• FirstName
• Surname
• OrderId

from the ReviewReason entity:
• Name</td></tr>
</table>

	Step 3: The administrator or sales assistant selects an option to view more order information. **[ALT]**	**Step 4:** The system Invoke use case **5.7. View Orders Overview**
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 CONTRAINTS AND SPECIFICATIONS	• The administrator or sales assistant cannot view their order details without an internet connection.	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

2.8. Inventory Management Subsystem

Table 69 - 8.1 Create Inventory Item

INVENTORY MANAGEMENT SUBSYSTEM			
AUTHORS (s): Ofhani Mungani		**DATE: 14/05/2022**	
VERSION: 1.0		**LAST REVIEW DATE: 14/05/2022**	
USE CASE INVENTORYITEMNAME:	Create Inventory Item	**USE CASE TYPE** Business Requirements: ☐ System Analysis: ☒ System Design: ☐	
USE CASE ID:	8.1		
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 add a new Inventory Item to the system. The use case begins when the administrator adds the new Inventory Item as an option to be choose from when placing an order to the supplier. The administrator will add all of 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 to the system.		
TYPICAL COURSE OF EVENTS:	**Actor Action**	**System Response**	
	Step 1: The administrator wants to add a new Inventory Item to the system.	**Step 2**: The system requires the admin to add information about the Inventory Item with the following attributes: From the **InventoryItem entity** • **InventoryItemName** • **QuntityonHand** From **Inventory Type entity** • **InventoryTypeName**	
	Step 3: The administrator inputs the required details.	**Step 4**: The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. **[ALT]**	
		Step 5: The system reads from the **InventoryItem entity** to ensure that the record being added does not match any existing records. **[ALT]**	

<table>
<tr><td></td><td></td><td>Step 6: The system saves the new inventory item information to the database with the following attributes:

From the InventoryItem entity
InventoryItemName
QuntityonHand
From Inventory Type entity
InventoryTypeName</td></tr>
<tr><td></td><td></td><td>Step 7: The system displays a success message on the screen to confirm the addition of the new Inventory Item.</td></tr>
<tr><td>ALTERNATE COURSES:</td><td colspan="2">ALT Step 4: The system fails to validate the information provided because of missing fields or incorrectly entered information.
Return to Step 3.</td></tr>
<tr><td></td><td colspan="2">ALT Step 5: An inventory item already exists on the database.
Terminate this use case.</td></tr>
<tr><td>CONCLUSION:</td><td colspan="2">This use case concludes when a new Inventory Item is added.</td></tr>
<tr><td>POST-CONDITION:</td><td colspan="2">The Inventory Item details are added to the database.
The Inventory Item details are displayed on the system for selection by the administrator when adding an Inventory item.</td></tr>
<tr><td>BUSINESS RULES</td><td colspan="2">Only the administrator can add a new Inventory Item.</td></tr>
<tr><td>IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS</td><td colspan="2">None</td></tr>
<tr><td>ASSUMPTIONS:</td><td colspan="2">None</td></tr>
<tr><td>OPEN ISSUES:</td><td colspan="2">None</td></tr>
</table>

Table 70 - 8.2 Search Inventory Item

<table>
<tr><th colspan="4">INVENTORY MANAGEMENT SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Ofhani Mungani</td><td colspan="2">DATE: 14/05/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 14/05/2022</td></tr>
<tr><td>USE CASE INVENTORYITEMNAME:</td><td colspan="2">Search Inventory Item</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">8.2</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">Medium</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process in which the administrator searches for Inventory Items on the system. The system responds by displaying the information relating to the search query.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the administrator wants to search for an Inventory Item on the system.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wants to search for an Inventory Item.</td><td colspan="2"></td></tr>
<tr><td>Step 2: The administrator inputs the search parameter.</td><td colspan="2">Step 3: The system searches the following attributes in the Inventory Item entity for a match:
• InventoryItemName
• QuntityonHand
• InventoryTypeName- linked by association from Inventory Type entity</td></tr>
<tr><td></td><td colspan="2">Step 4: The system displays the search results information. [ALT]</td></tr>
<tr><td>ALTERNATE COURSES:</td><td colspan="3">[ALT] Step 4: The search query entered does not match any existing instances of Inventory Items in the system database.
• Return to Step 3
Terminate use case</td></tr>
<tr><td>CONCLUSION:</td><td colspan="3">This use case concludes when the information is displayed on the system.</td></tr>
<tr><td>POST-CONDITION:</td><td colspan="3">The Inventory Item entity was searched for and an entry that corresponds to the search query or parameter entered.</td></tr>
<tr><td>BUSINESS RULES</td><td colspan="3">• None</td></tr>
</table>

IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 71 - 8.3 Update Inventory Item

<table>
<tr><th colspan="4">INVENTORY MANAGEMENT SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Ofhani Mungani</td><td colspan="2">DATE: 14/05/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 14/05/2022</td></tr>
<tr><td>USE CASE INVENTORYITEMNAME:</td><td>Update Inventory Item</td><td colspan="2" rowspan="4">USE CASE TYPE

Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>8.3</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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 for the administrator to choose from when placing an order to a supplier. The administrator will update all of the pertinent information regarding the Inventory Item and save it to the business database.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use is triggered when the administrator updates the Inventory Item as an option for administrator to choose from.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wants to update an Inventory Item existing on the system.</td><td colspan="2">Step 2: The system invokes Use Case 4.11 “Search Inventory Item”.</td></tr>
<tr><td>Step 3: The administrator selects the desired inventory item.</td><td colspan="2">Step 4: The system requires the admin to update more information about the Inventory Item in the Inventory Item entity such as:
• InventoryItemName
• QuntityonHand
• InventoryTypeName- linked by association from Inventory Type entity</td></tr>
<tr><td>Step 5: The administrator inputs the details required.</td><td colspan="2">Step 6: The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. [ALT]</td></tr>
</table>

		Step 7: The system prompts the administrator to confirm the update of an Inventory Item with the details provided.
		Step 8: The system reads from the **InventoryItem entity** to ensure that the record being added does not match any existing records. **[ALT]**
	Step 9: The administrator confirms the information provided. **[ALT]**	**Step 10:** The system saves the new inventory item information to the database with the following attributes: From the **InventoryItem entity** • **InventoryItemName** • **QuntityonHand** From **Inventory Type entity** • **InventoryTypeName**
		Step 11: The system displays a success message on the screen to confirm the update of an Inventory Item.
ALTERNATE COURSES:	**ALT Step 6**: The system fails to validate the information provided because of missing fields or incorrectly entered information. • Return to Step 5.	
	ALT Step 8: An inventory item already exists on the database • Terminate this use case.	
	ALT Step 9: The administrator does not confirm the update of a new Inventory Item information. • Return to Step 5. • Terminate this use case.	
CONCLUSION:	This use case concludes when an Inventory Item is updated.	
POST-CONDITION:	• The Inventory Item details are updated to the database.	
BUSINESS RULES	• Only the administrator can update an Inventory Item.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 72 - 8.4 Delete Inventory Item

INVENTORY MANAGEMENT SUBSYSTEM			
AUTHORS (s): Ofhani Mungani		**DATE: 14/05/2022**	
VERSION: 1.0		**LAST REVIEW DATE: 14/05/2022**	
USE CASE INVENTORYITEMNAME:	Delete Inventory Item	**USE CASE TYPE**	
USE CASE ID:	8.4	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 narrative describes the process in which the administrator wishes to delete an Inventory Item from the system. The administrator will search for the required Inventory Item selected for deletion. The administrator will then confirm the deletion of the Inventory Item 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:	This use case is called when the administrator wishes to delete an Inventory Item.		
TYPICAL COURSE OF EVENTS:	**Actor Action**	**System Response**	
	Step 1: The administrator wishes to delete an Inventory Item.	**Step 2:** The system invokes Use Case 4.11 “Search Inventory Item”.	
	Step 3: The administrator selects the desired inventory item.	**Step 4:** The system retrieves the inventory item information from the database with the following attributes: From the **InventoryItem entity** • **InventoryItemName** • **QuntityonHand** From **Inventory Type entity** • **InventoryTypeName**	
	Step 5: The administrator selects the delete option **[ALT]**	**Step 6:** The system prompts the administrator to confirm the deletion of the Inventory Item selected.	

	Step 7: The administrator confirms the deletion of the Inventory Item from the system. **[ALT]**	**Step 8:** The system removes all of the details related of the selected Inventory Item from database. **[ALT]**
		Step 9: The system informs the administrator of the successful deletion of the Inventory Item. **[ALT]**
ALTERNATE COURSES:	**ALT Step 2:** The system does not invoke the Search Inventory Type use case. • Go to Step 3	
	ALT Step 5: The administrator no longer wants to delete an Inventory Item. • Selects cancel button. • Terminate this use case.	
	ALT Step 7: The administrator does not confirm the deletion of Inventory Item. • Return to ALT Step 5	
	ALT Step 8: An existing Inventory item has a supplier or product assigned to it. • The system restricts the deletion of the Inventory Item. • Go to ALT step 8.	
	ALT Step 9: The system informs the administrator of the unsuccessful deletion of the Inventory Item. • Terminate this use case.	
CONCLUSION:	This use case concludes when the administrator is notified of the successful deletion of the Inventory Item.	
POST-CONDITION:	• The system deletes all of the details related to the selected Inventory Item from the **Inventory Item entity**. • The system removes all of the details related to the selected Inventory Item.	
BUSINESS RULES	• Only the administrator can delete an Inventory Item.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 73 - 8.5 Create Inventory Type

<table>
<tr><th colspan="4">INVENTORY MANAGEMENT SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Ofhani Mungani</td><td colspan="2">DATE: 14/05/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 30/08/2022</td></tr>
<tr><td>USE CASE INVENTORYTYPENAME:</td><td colspan="2">Create Inventory Type</td><td rowspan="4">USE CASE TYPE

Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">8.5</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process in which the administrator wishes to add a new Inventory Type to the system to be assigned to an Inventory item. The use case begins when the administrator adds the new Inventory Type as an option to choose from when an Inventory item is added on the system. The administrator will add all of the pertinent information regarding the Inventory Type and save it to the business database.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use case is triggered when the administrator wants to add a new Inventory Type to the system.</td></tr>
<tr><td rowspan="5">TYPICAL COURSE OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wants to add a new Inventory Type to the system.</td><td colspan="2">Step 2: The system requires the admin to add information about the Inventory Type in the InventoryType entity such as:
• InventoryTypeName</td></tr>
<tr><td>Step 3: The administrator inputs the required details. [ALT]</td><td colspan="2">Step 4: The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. [ALT]</td></tr>
<tr><td></td><td colspan="2">Step 5: The system reads from the InventoryType entity to ensure that the record being added does not match any existing records. [ALT]</td></tr>
<tr><td></td><td colspan="2">Step 6: The system saves its information to the InventoryType entity with the following attributes.
• Id
• InventoryTypeName</td></tr>
</table>

		Step 7: The system displays a success message on the screen to confirm the addition of the new inventory type.
ALTERNATE COURSES:	**ALT Step 3**: The administrator wishes not to procced to add a new inventory type. Terminate Use Case	
	ALT Step 4: The system fails to validate the information provided because of missing fields or incorrectly entered information. • Return to Step 3. •	
	ALT Step 5: The inventory type already exists on the database. • The system restricts the addition of the inventory type to prevent duplicate values • Terminate this use case.	
CONCLUSION:	This use case concludes when a new inventory type is added.	
POST-CONDITION:	• The inventory type details are added to the database. • The inventory type details are displayed on the system for selection by the administrator when adding an Inventory item.	
BUSINESS RULES	• Only the administrator can add a new inventory type.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 74 - 8.6 Search Inventory Type

<table>
<tr><th colspan="4">INVENTORY MANAGEMENT SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Ofhani Mungani</td><td colspan="2">DATE: 14/05/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 30/08/2022</td></tr>
<tr><td>USE CASE INVENTORYTYPENAME:</td><td colspan="2">Search Inventory Type</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">8.6</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">Medium</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process in which the administrator searches for inventory types on the system. The system responds by displaying the information relating to the search query.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the administrator wants to search for an inventory type on the system.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wants to search for an inventory type.</td><td colspan="2"></td></tr>
<tr><td>Step 2: The administrator inputs the search parameter.</td><td colspan="2">Step 3: The system searches and retrieves the following attributes in the InventoryType entity for a match:
• InventoryTypeName</td></tr>
<tr><td></td><td colspan="2">Step 4: The system displays the search results information. [ALT]</td></tr>
<tr><td>ALTERNATE COURSES:</td><td colspan="3">[ALT] Step 4: The search query entered does not match any existing instances of Inventory Types in the system database.
• Return to Step 3
Terminate use case</td></tr>
<tr><td>CONCLUSION:</td><td colspan="3">This use case concludes when the information is displayed on the system.</td></tr>
<tr><td>POST-CONDITION:</td><td colspan="3">The InventoryType entity was searched for and an entry that corresponds to the search query or parameter entered.</td></tr>
<tr><td>BUSINESS RULES</td><td colspan="3">• None</td></tr>
<tr><td>IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS</td><td colspan="3">• None</td></tr>
<tr><td>ASSUMPTIONS:</td><td colspan="3">• None</td></tr>
<tr><td>OPEN ISSUES:</td><td colspan="3">• None</td></tr>
</table>

Table 75 - 8.7 Update Inventory Type

INVENTORY MANAGEMENT SUBSYSTEM			
AUTHORS (s): Ofhani Mungani		**DATE: 14/05/2022**	
VERSION: 1.0		**LAST REVIEW DATE: 30/08/2022**	
USE CASE INVENTORYTYPENAME:	Update Inventory Type		**USE CASE TYPE**
USE CASE ID:	8.7		Business Requirements: ☐
PRIORITY:	High		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 in which the administrator wishes to update an inventory type to the system. The use case begins when the administrator updates inventory type as an option for the administrator to choose from when creating an inventory item. The administrator will update all of the pertinent information regarding the inventory type and save it to the database.		
PRE-CONDITION:	• The administrator must be logged onto the system.		
TRIGGER:	The use is triggered when the administrator updates the Inventory Type as an option for administrator to choose from.		
TYPICAL COURSE OF EVENTS:	**Actor Action**	**System Response**	
	Step 1: The administrator wants to update an inventory type existing on the system.	**Step 2:** The system invokes Use Case 4.11 “Search Inventory Type”.	
	Step 3: The administrator selects the desired inventory type.	**Step 4**: The system requires the admin to update more information about the inventory type in the **InventoryType entity** such as: • **InventoryTypeName**	
	Step 5: The administrator inputs the details required. **[ALT]**	**Step 6**: The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. **[ALT]**	
		Step 7: The system prompts the administrator to confirm the update of an inventory type with the details provided.	

	Step 8: The administrator confirms the information provided. **[ALT]**	**Step 9:** The system reads from the **InventoryType entity** to ensure that the record being added does not match any existing records. **[ALT]**
		Step 10: The system saves its information to the **InventoryType** entity in the database.
		Step 11: The system displays a success message on the screen to confirm the update of an inventory type.
ALTERNATE COURSES:	ALT **Step 5**: The administrator wishes not to procced to update an inventory type. Terminate Use Case	
	ALT **Step 6**: The system fails to validate the information provided because of missing fields or incorrectly entered information. • Return to Step 5.	
	ALT **Step 8:** The administrator does not confirm the update of a new Inventory Type information. • Return to Step 4. • Terminate this use case	
	ALT **Step 9:** The inventory type already exists on the database. • Terminate this use case.	
CONCLUSION:	This use case concludes when an inventory type is updated.	
POST-CONDITION:	• The inventory type details are updated to the database.	
BUSINESS RULES	• Only the administrator can update an Inventory Type.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 76 - 8.8 Delete Inventory Type

<table>
<tr><th colspan="4">INVENTORY MANAGEMENT SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Ofhani Mungani</td><td colspan="2">DATE: 14/05/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 30/08/2022</td></tr>
<tr><td>USE CASE INVENTORYTYPENAME:</td><td>Delete Inventory Type</td><td colspan="2" rowspan="4">USE CASE TYPE

Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>8.8</td></tr>
<tr><td>PRIORITY:</td><td>Medium</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case narrative describes the process in which the administrator wishes to delete an inventory type from the system. The administrator will search for the required inventory type selected for deletion. The administrator will then confirm the deletion of the Inventory Type and the use case concludes when the system notifies the administrator of a successful deletion.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the administrator wishes to delete an Inventory Type.</td></tr>
<tr><td rowspan="5">TYPICAL COURSE OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wishes to delete an inventory type.</td><td colspan="2">Step 2: The system invokes Use Case 4.11 “Search Inventory Type”.</td></tr>
<tr><td>Step 3: The administrator selects the desired inventory type</td><td colspan="2">Step 4: The system retrieves the inventory type information from the database with the following attributes:

From the InventoryType entity
• InventoryTypeName</td></tr>
<tr><td>Step 5: The administrator selects the delete option

[ALT]</td><td colspan="2">Step 6: The system prompts the administrator to confirm the deletion of the inventory type selected.</td></tr>
<tr><td>Step 7: The administrator confirms the deletion of the Inventory Type from the system. [ALT]</td><td colspan="2">Step 8: The system removes all of the details related of the selected inventory type from database. [ALT]</td></tr>
</table>

		Step 9: The system informs the administrator of the successful deletion of the inventory type. **[ALT]**
ALTERNATE COURSES:	**ALT Step 5**: The administrator no longer wants to delete an Inventory type. • Selects cancel button. • Terminate this use case	
	ALT Step 7: The administrator does not confirm the deletion of the Inventory Type. • Return to ALT Step 5	
	ALT Step 8: An existing Inventory item has the Inventory Type assigned to it. • The system restricts the deletion of the Inventory Type. • Go to ALT step 9.	
	ALT Step 9: The system informs the administrator of the unsuccessful deletion of the inventory type. • Terminate use case	
CONCLUSION:	This use case concludes when the administrator is notified of the successful deletion of the inventory type.	
POST-CONDITION:	• The system deletes all of the details related to the selected inventory type from the **InventoryType entity**. • The system removes all of the details related to the selected inventory type	
BUSINESS RULES	• Only the administrator can delete an inventory type.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 77 - 8.9 Add Write-Off Reason

WRITE-OFF REASON SUBSYSTEM			
AUTHORS (s): Thenjiwe Ntsonda		**DATE: 12/05/2022**	
VERSION: 1.0		**LAST REVIEW DATE: 12/05/2022**	
USE CASE NAME:	Add Write-Off Reason	**USE CASE TYPE**	
USE CASE ID:	8.9	Business Requirements:	☐
PRIORITY:	High	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 in which the administrator wishes to add a new write-off reason to the system. The use case begins when the administrator adds the new write-off reason on the system. The administrator will add all of 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 to the system.		
TYPICAL COURSE OF EVENTS:	**Actor Action**	**System Response**	
	Step 1: The administrator wants to add a new write-off reason to the system.	**Step 2**: The system requires the admin to add information about the write-off reason in the **Write-Off Reason table** such as: • **Name**	
	Step 3: The administrator inputs the required details.	**Step 4**: The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. **[ALT]**	
		Step 5: The system requests the administrator to confirm the details of a new write-off reason.	
	Step 6: The administrator confirms the information provided. **[ALT]**	**Step 7:** The system saves its information to the Write-off reason table in the database. **[ALT]**	
		Step 8: The system displays a success message on the screen to confirm the addition of the new write-off reason.	
ALTERNATE COURSES:	ALT **Step 4:** The system fails to validate the information provided because of missing fields or incorrectly entered information. • Return to Step 3.		

	ALT Step 6: The administrator does not confirm the addition of the new write-off reason. • Return to Step 3 • Terminate this use case.
	ALT Step 7: The added write-off reason matches an existing record on the system • The system restricts the addition of the write-off reason to prevent duplicate values.
CONCLUSION:	This use case concludes when a new write-off reason is added.
POST-CONDITION:	• The write-off reason details are added to the database.
BUSINESS RULES	• Only the administrator can add a new write-off reason.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 78 - 8.10 Search Write-Off Reason

<table>
<tr><th colspan="4">WRITE-OFF REASON SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Thenjiwe Ntsonda</td><td colspan="2">DATE: 12/05/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 12/05/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>Search Write-Off Reason</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>8.10</td></tr>
<tr><td>PRIORITY:</td><td>Medium</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process in which the administrator searches for write-off reasons on the system. The system responds by displaying the information relating to the search query.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the administrator wants to search for a write-off reason on the system.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wants to search for a write-off reason.</td><td colspan="2">Step 2: The system requests the administrator to enter a search query or parameter.</td></tr>
<tr><td>Step 3: The administrator inputs the search parameter.</td><td colspan="2">Step 4: The system searches and retrieves the following attributes in the write-off reason table for a match:
• Name</td></tr>
<tr><td></td><td colspan="2">Step 5: The system displays the search results information. [ALT]</td></tr>
<tr><td>ALTERNATE COURSES:</td><td colspan="3">[ALT] Step 5: The search query entered does not match any existing instances of write-off reasons in the system database.
• Return to Step 3
Terminate use case</td></tr>
<tr><td>CONCLUSION:</td><td colspan="3">This use case concludes when the information is displayed on the system.</td></tr>
<tr><td>POST-CONDITION:</td><td colspan="3">The Write-off reason table was searched for and an entry that corresponds to the search query or parameter entered.</td></tr>
<tr><td>BUSINESS RULES</td><td colspan="3">• None</td></tr>
<tr><td>IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS</td><td colspan="3">• None</td></tr>
<tr><td>ASSUMPTIONS:</td><td colspan="3">• None</td></tr>
<tr><td>OPEN ISSUES:</td><td colspan="3">• None</td></tr>
</table>

Table 79 - 8.11 Update Write-Off Reason

WRITE-OFF REASON SUBSYSTEM		
AUTHORS (s): Thenjiwe Ntsonda		**DATE: 12/05/2022**
VERSION: 1.0		**LAST REVIEW DATE: 12/05/2022**
USE CASE NAME:	Update Write-Off Reason	**USE CASE TYPE**
USE CASE ID:	8.11	Business Requirements: ☐
PRIORITY:	High	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 in which the administrator wishes to update a write-off reason on the system. The use case begins when the administrator updates all of the pertinent information regarding the write-off reason and save it to the system.	
PRE-CONDITION:	• The administrator must be logged onto the system.	
TRIGGER:	The use is triggered when the administrator wishes to update the write-off reason.	
TYPICAL COURSE OF EVENTS:	**Actor Action**	**System Response**
	Step 1: The administrator wants to update a write-off reason existing on the system.	**Step 2:** The system invokes Use Case 4.11 “Search Write-off reason”.
		Step 3: The system requires the admin to update more information about the write-off reason in the **Write-Off Reason table** such as: • **Name**
	Step 4: The administrator inputs the details required.	**Step 5**: The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. **[ALT]**
		Step 6: The system requests the administrator to confirm the update of a write-off reason with the details provided.
	Step 7: The administrator confirms the information provided. **[ALT]**	**Step 8:** The system saves its information to the write-off reason table in the database.
		Step 9: The system displays a success message on the screen to confirm the update of a write-off reason.

ALTERNATE COURSES:	**ALT** Step 5: The system fails to validate the information provided because of missing fields or incorrectly entered information. • Return to Step 4. **ALT** Step 7: The administrator does not confirm the update of a new write-off reason information. • Return to Step 4. • Terminate this use case.
CONCLUSION:	This use case concludes when a write-off reason is updated.
POST-CONDITION:	• The write-off reason details are updated to the database.
BUSINESS RULES	• Only the administrator can update a write-off reason.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 80 - 8.12 Delete Write-Off Reason

WRITE-OFF REASON SUBSYSTEM		
AUTHORS (s): Thenjiwe Ntsonda	**DATE: 12/05/2022**	
VERSION: 1.0	**LAST REVIEW DATE: 12/05/2022**	
USE CASE NAME:	Delete Write-Off Reason	**USE CASE TYPE**
USE CASE ID:	8.12	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 narrative describes the process in which the administrator wishes to delete a write-off reason from the system. The administrator will search for the required write-off reason selected for deletion. The administrator will then confirm the deletion of the write-off 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:	This use case is called when the administrator wishes to delete a write-off reason.	
TYPICAL COURSE OF EVENTS:	**Actor Action**	**System Response**
	Step 1: The administrator wishes to delete a write-off reason.	**Step 2:** The system invokes Use Case 4.11 “Search Write-off reason”.
		Step 3: The system requests the administrator to confirm the deletion of the write-off reason selected.
	Step 4: The administrator confirms the deletion of the write-off reason from the system. **[ALT]**	**Step 5:** The system removes all of the details related of the selected write-off reason from database. **[ALT]**
		Step 6: The system informs the administrator of the successful deletion of the write-off reason. **[ALT]**
ALTERNATE COURSES:	**[ALT]** Step 4: The administrator does not confirm the deletion of the write-off reason. • Terminate this use case. **[ALT]** Step 5: An existing record of an inventory item being written off has the write-off reason assigned to it. • The system restricts the deletion of the write-off reason. • Go to ALT step 6. **[ALT]** Step 6: The system informs the administrator of the unsuccessful deletion of the write-off reason.	

CONCLUSION:	This use case concludes when the administrator is notified of the successful deletion of the write-off reason.
POST-CONDITION:	• The system deletes all of the details related to the selected write-off reason from the **Write-Off Reason table**. • The system removes all of the details related to the selected write-off reason.
BUSINESS RULES	• Only the administrator can delete a write-off reason.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 81 -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 INVENTORYITEMNAME:	Write-Off Inventory	**USE CASE TYPE**	
USE CASE ID:	8.13	Business Requirements: ☐	
PRIORITY:	High	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 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**	
	Step 1: The administrator wants to write-off an Inventory Item out of the system.	**Step 2**: The system requires the admin to add the write-off information for the Inventory Item with the following attributes: From the **InventoryWriteOff entity** • **writeOffQuantity** From the **InventoryItem entity** • **InventoryItemName** From **WriteOffReason entity** • **Name**	
	Step 3: The administrator inputs the required details.	**Step 4**: The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. **[ALT]**	
		Step 5: The system prompts the administrator to confirm the write-off details.	

	Step 6: The administrator confirms the information provided. **[ALT]**	**Step 7:** The system reads from the **InventoryItem entity** to check if **QuantityOnHand** is less than the **writeOffQuantity** from the **InventoryWriteOff entity**. **[ALT]**
		Step 8: The system increases the **QuantityOnHand** attribute in the **InventoryItem entity** and saves the new inventory item write-off information to the database with the following attributes: From the **InventoryWriteOff** entity • **writeOffQuantity** • **writeOffDate** From the **InventoryItem entity** • **InventoryItemName** • **QuantityOnHand** From **WriteOffReason entity** • **Name**
		Step 9: The system displays a success message on the screen to confirm the Inventory Item write-off.
ALTERNATE COURSES:	**ALT Step 4**: The system fails to validate the information provided because of missing fields or incorrectly entered information. • Return to Step 3.	
	ALT STEP 6: The administrator does not confirm the addition of the new product. • Return to Step 3	
	ALT Step 7: **QuantityOnHand** is less than the **writeOffQuantity** • Display write-off error message • 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 can write off an Inventory Item.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 82 - 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 INVENTORYITEMNAME:	Write-Off Product	**USE CASE TYPE** Business Requirements: ☐ System Analysis: ☒ System Design: ☐	
USE CASE ID:	8.14		
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 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**	
	Step 1: The administrator wants to write-off a product out of the system.	**Step 2**: The system requires the admin to add the write-off information for the product with the following attributes: From the **ProductWriteOff entity** • **writeOffQuantity** From the **Product entity** • **Name** From **WriteOffReason entity** • **Name**	
	Step 3: The administrator inputs the required details.	**Step 4**: The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. **[ALT]**	
		Step 5: The system prompts the administrator to confirm the write-off details	

	Step 6: The administrator confirms the information provided. **[ALT]**	**Step 7:** The system reads from the **Product entity** to check if **QuantityOnHand** is less than the **writeOffQuantity** from the **ProductWriteOff entity**. **[ALT]**
		Step 8: The system increases the **QuantityOnHand** attribute in the **Product entity** and saves the new product write-off information to the database with the following attributes: From the **ProductWriteOff** entity • **WriteOffQuantity** • **WriteOffDate** From the **Product entity** • **Name** • **QuantityOnHand** From **WriteOffReason entity** • **Name**
		Step 9: The system displays a success message on the screen to confirm the product write-off.
ALTERNATE COURSES:	**ALT Step 4**: The system fails to validate the information provided because of missing fields or incorrectly entered information. • Return to Step 3.	
	ALT STEP 6: The administrator does not confirm the addition of the new product. • Return to Step 3	
	ALT Step 7: **QuantityOnHand** is less than the **WriteOffQuantity** • Display write-off error message • Return to Step 3.	
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 can write off a product.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 83 -8.15 Convert Inventory to Product

<table>
<tr><th colspan="4">INVENTORY MANAGEMENT SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Kyle van Eeden</td><td colspan="2">DATE: 06/06/2022</td></tr>
<tr><td colspan="2">VERSION: 1.1</td><td colspan="2">LAST REVIEW DATE: 09/06/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>Convert Inventory to Product</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>8.15</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Inventory Manager or Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">The use case begins when the inventory manager or administrator wishes to produce finished products using the available inventory items on hand. The system will prompt the administrator or inventory manager to select a specific product and the quantity to be produced. The system will update the quantity of various inventory items used to produce the desired amount of products specified and save the updated quantities to the database. This use case concludes when the system displays a production success confirmation message.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The product to be produced must be defined on the system.
The product to be produced must have been configured for production.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the system administrator or inventory manager wishes to produce new finished products on the system.</td></tr>
<tr><td rowspan="2">TYPICAL COURSE</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The system administrator or inventory manager wishes to produce finished products on the system.</td><td colspan="2">Step 2: The system retrieves all the products defined on the system from the Product entity and prompts the system administrator or inventory manager to capture the following details for production:</td></tr>
</table>

		• ProductName • ProductionQuantity
	Step 3: The system administrator or inventory manager searches for the name of the desired product they wish to produce. **[ALT]**	**Step 4:** The system displays the names of all products that have matched the search query specified.
	Step 5: The system administrator or inventory manager selects the product they wish to produce on the system.	**Step 6:** The system captures the following details for the selected product: • ProductID • ProductName
	Step 7: The system administrator or inventory manager enters the quantity of the product they wish to produce and selects the option to produce the finished product.	**Step 8:** The system validates the following details that have been captured: • ProductName • ProductionQuantity **[ALT]**
		Step 9: The system validates that there are sufficient quantities of all required inventory items as per the product's production configuration details stored in the **ProductInventoryItem** entity: • ProductID • InventoryItemID • InventoryItemQuantity **[ALT]**
		Step 10: The system saves the quantity of the specific product and all inventory items used for the production process in the following entities: In the **Product** entity: • QuantityOnHand

		In the **InventoryItem** entity: • QuantityOnHand
		Step 11: The system displays an alert confirming that the production operation was successful.
ALTERNATE COURSES:	**ALT Step 3:** The administrator or inventory manager selects the option to cancel the production operation: • The use case is terminated.	
	ALT Step 8: The system detects that all required details have not been captured by the administrator or inventory manager: • The system displays a validation error message to point out which fields have raised exceptions. • Return to **Step 3**.	
	ALT Step 9: The system detects that the desired quantity of the product cannot be produced given the production requirements configured and the available quantity of each required inventory item: • The system informs the administrator or inventory manager that the desired quantity of the product requested cannot be produced. • Return to **Step 2.**	
CONCLUSION:	This use case concludes when the system displays an alert confirming that the finished products produced have been captured successfully.	
POST-CONDITION:	The quantity on hand of the product produced and its required inventories is updated in the **Product** and **InventoryItem** entities, respectively.	
BUSINESS RULES	• Only the system administrator and inventory manager roles should be allowed to produce finished products on the system. • The amount of finished products which can be produced at any time should be constrained by the quantity of required inventory items available as per what is stored in the system's database.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• The system should not produce the quantity of the final products requested if there are insufficient inventory items available as per the database. • The quantity of finished products to be produced should always at least be 1.	
ASSUMPTIONS:	• The product's production configuration has already been set up.	
OPEN ISSUES:	• None	

Table 84 - 8.16 Search Procured Inventory

INVENTORY SUBSYSTEM			
AUTHORS (s): Thenjiwe Ntsonda		**DATE: 08/06/2022**	
VERSION: 1.0		**LAST REVIEW DATE: 08/06/2022**	
USE CASE NAME:	Search Procured Inventory	**USE CASE TYPE**	
USE CASE ID:	8.16	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 searches for procured inventory 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 procured inventory on the system.		
TYPICAL COURSE OF EVENTS:	**Actor Action**	**System Response**	
	Step 1: The administrator wants to search for procured inventory.		
	Step 2: The administrator enters the search parameter.	**Step 3:** The system searches using the following attributes in the **InventoryProcured** table: • **CompanyName** • **InventoryItemName** • **QuantityReceived** • **DateLogged**	
		Step 4: The system displays the search results **[ALT]**	
ALTERNATE COURSES:	**ALT Step 4**: The search query entered does not match any existing instances of procured inventory in the database. • Return to Step 2 or terminate the use case		
CONCLUSION:	This use case concludes when the desired procured inventory 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	• None		
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None		
ASSUMPTIONS:	• None		
OPEN ISSUES:	• None		

Table 85 -8.17 View Procured Inventory Details

<table>
<tr><th colspan="4">INVENTORY SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Thenjiwe Ntsonda</td><td colspan="2">DATE: 08/06/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 08/06/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">View Procured Inventory Details</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">8.17</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">Medium</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use case is triggered when the administrator wishes to view the details of the inventory received from a supplier.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wants to view the details of inventory item(s) received from a supplier.</td><td colspan="2">Step 2: The system invokes 18.5 Search Procured Inventory</td></tr>
<tr><td>Step 3: The administrator selects the desired procured inventory item.</td><td colspan="2">Step 4: The system retrieves and displays the procured inventory item details which have the following attributes:

From the InventoryProcured Table
• InvoiceNo
• QuantityReceived

From the Supplier Table
• CompanyName

From the InventoryItem Table
• InventoryItemName
[ALT]</td></tr>
<tr><td>ALTERNATE COURSES:</td><td colspan="3">ALT Step 4: The administrator wants to return to the previous page
• The administrator clicks on the return button
• The system redirects the administrator to the previous page.</td></tr>
</table>

CONCLUSION:	This use case concludes when the inventory item details are displayed on the system.
POST-CONDITION:	• The selected inventory item details are displayed on the system.
BUSINESS RULES	• Only the administrator can view the procured inventory item details.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 86 -8.18 Capture Procured Inventory

<table>
<tr><th colspan="4">INVENTORY SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Thenjiwe Ntsonda</td><td colspan="2">DATE: 07/18/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 07/18/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>Capture Returned Product</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>8.18</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator or Sales Assistant</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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”.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator or sales assistant must be logged in.
• The order status of the order must be “Dispatched” or “Delivered”</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">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.</td></tr>
<tr><td rowspan="2">TYPICAL COURSE</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator or sales assistant wishes to log a return on the system.</td><td colspan="2">Step 2: The system requires the following information from the ReturnedProduct entity with the following attributes:
• QuantityReceived

From the Product entity</td></tr>
</table>

		• **Name** From the **ReturnReason** entity • **ReturnReasonName**
	Step 3: The administrator or sales assistant provides the relevant information. **[ALT]**	**Step 4:** The system requires the administrator to confirm the entered details.
		Step 5: The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. **[ALT]**
		Step 6: The system increases the **QuantityOnHand** value in the Product entity with the value received from the QuantityReceived attribute in the **ReturnedProduct** entity. The system then updates the **OrderStatus** attribute in the Order entity to "Includes Returned Product(s)".
		Step 7: The system saves its information to the **ReturnedProduct** entity with the following attributes: • **ReturnedProductId** • **QuantityReceived** • **ProductName** • **DateLogged** • **ReturnReasonName**
		Step 8: The system displays a success message to the administrator or sales assistant.
ALTERNATE COURSES:	**ALT Step 3:** The administrator or sales assistant does not wish to log the return of a product anymore • Terminate use case.	
	ALT Step 5A: The system fails to validate the information provided because of missing fields or incorrectly entered information. • Return to Step 3.	

	ALT Step 5B: The product selected for return was not in the selected order • Display error message **ALT Step 5B**: The quantity entered is more than what was ordered • Display error message
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 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 Natuurlik can be logged for a return. • Only the administrator and sales assistant have access to this function.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• The administrator and sales assistant have a stable internet connection.
OPEN ISSUES:	• None

Table 87- 8.19 Capture Returned Product

INVENTORY SUBSYSTEM			
AUTHORS (s): Thenjiwe Ntsonda		**DATE: 07/18/2022**	
VERSION: 1.0		**LAST REVIEW DATE: 07/18/2022**	
USE CASE NAME:	Capture Returned Product	**USE CASE TYPE**	
USE CASE ID:	8.16	Business Requirements: ☐	
PRIORITY:	High	System Analysis:	
SOURCE:	ByteXpress Requirements List	System Design: ☐	
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"		
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	**Actor Action**	**System Response**	
	Step 1: The administrator or sales assistant wishes to log a return on the system.	**Step 2:** The system requires the following information from the **ReturnedProduct** entity with the following attributes: • **QuantityReceived** From the **Product** entity • **Name** From the **ReturnReason** entity • **ReturnReasonName**	
	Step 3: The administrator or sales assistant provides the relevant information. **[ALT]**	**Step 4:** The system requires the administrator to confirm the entered details.	
		Step 5: The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. **[ALT]**	

		Step 6: The system increases the **QuantityOnHand** value in the Product entity with the value received from the QuantityReceived attribute in the **ReturnedProduct** entity. The system then updates the **OrderStatus** attribute in the Order entity to "Includes Returned Product(s)".
		Step 7: The system saves its information to the **ReturnedProduct** entity with the following attributes: • **ReturnedProductId** • **QuantityReceived** • **ProductName** • **DateLogged** • **ReturnReasonName**
		Step 8: The system displays a success message to the administrator or sales assistant.
ALTERNATE COURSES:	**ALT Step 3:** The administrator or sales assistant does not wish to log the return of a product anymore • Terminate use case.	
	ALT Step 5A: The system fails to validate the information provided because of missing fields or incorrectly entered information. • Return to Step 3. **ALT Step 5B:** The product selected for return was not in the selected order • Display error message **ALT Step 5B**: The quantity entered is more than what was ordered • Display error message	
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 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 Natuurlik can be logged for a return. • Only the administrator and sales assistant have access to this function.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• The administrator and sales assistant have a stable internet connection.	
OPEN ISSUES:	• None	

Table 88- 8.20 Package Order

INVENTORY MANAGEMENT SUBSYSTEM		
AUTHORS (s): Kyle van Eeden	**DATE: 13/07/2022**	
VERSION: 1.1	**LAST REVIEW DATE: 26/07/2022**	
USE CASE NAME:	Package Order	**USE CASE TYPE**
USE CASE ID:	8.20	Business Requirements: ☐
PRIORITY:	High	System Analysis: ☒
SOURCE:	ByteXpress Requirements List	System Design: ☐
PRIMARY BUSINESS ACTOR	• Administrator/ Sales Assistant	
PRIMARY SYSTEM ACTOR	• None	
OTHER PARTICIPATING ACTORS:	• None	
OTHER INTERESTED STAKEHOLDERS:	• None	
DESCRIPTION:	This use case begins when an Administrator or Sales Assistant wishes to capture an order's products which has been prepared before shipping the parcel off to the couriers. The system user will request the system to package products for a particular order and the system will display details on all products which have been packaged at the time of the request. The user can then request the system to package additional products required. The system will then prompt the administrator or sales assistant to select a product from all options made available and to provide the quantity which has been packaged. After providing the required packaging details, the system will validate the provided details to ensure the selected product and its quantity specified is correct as per what is stored in the **OrderLine** entity for the relevant order. The system then determines whether the order has been placed by a reseller or customer. For orders placed by a customer user, the system will add the package order details to the **OrderProduct** entity. For orders placed by a reseller user, the system will also validate whether there are sufficient levels of inventory available of the requested product, and if the desired quantity can be packaged, the system will deduct the quantity captured from the current quantity on hand of the product and	

<table>
<tr><td></td><td colspan="2">save the updated quantity to the Product entity. The system will then add the packaged order details to the OrderProduct entity.

The use case concludes with the system displaying a message to confirm that the package activity has been completed successfully.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="2">• 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.</td></tr>
<tr><td>TRIGGER:</td><td colspan="2">This use case is called when a system administrator or sales assistant wishes to capture products that have been packaged for a particular order.</td></tr>
<tr><td>TYPICAL COURSE</td><td>Actor Action</td><td>System Response</td></tr>
<tr><td></td><td>Step 1: The system Administrator or Sales Assistant wishes to capture a product which have been packaged for a particular order on the system.</td><td>Step 2: The system will retrieve the following details for all products which have been packaged for the requested order from the OrderProduct entity:

• OrderNumber
• ProductId
• ProductQuantity
• TransactionDate

The following is retrieved from the Product entity for each product packaged:

• ProductName

The system will then display all the package order details retrieved with an option provided to package an additional product.</td></tr>
<tr><td></td><td>Step 3: The user selects the option to capture an additional</td><td>Step 4: The system retrieves the following details from the entities listed below:</td></tr>
</table>

	product which has been packaged for the order.	From the **Product** table: • ProductName From the **Order** table: • OrderId (Order Reference Number of the requested order) The system then prompts the user to enter the following required details: • ProductName • ProductQuantity
	Step 5: The user selects a product and enters the quantity which has been packaged. **[ALT]**	**Step 6:** The system validates the captured packaging details based on the validation requirements of the **OrderProduct** table. The system also validates that the product requested for packaging is associated with the current order and if the correct quantity has been specified by retrieving the following details: From the **OrderLine** entity: • OrderID • ProductID • Count **[ALT]**
		Step 7: The system retrieves the following details from the **Order** entity for the requested order to determine if it is a reseller order: • IsResellerOrder

		[ALT]
		Step 8: If the order has been placed by a reseller, the system will then determine if there is sufficient stock on hand to package the specified quantity of the requested product. The following details are retrieved from the **Product** table to perform the check: • QuantityOnHand 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. The system will save the updated quantity on hand value for the particular product using the following attribute in the **Product** entity: • QuantityOnHand [ALT]
		Step 9: The system retrieves the user's claims details from the **User** entity to retrieve the following details for the user who has effected the packaging operation: • FirstName The system adds the following captured package order transaction details to the **OrderProduct** table: • OrderId • ProductId • ProductQuantity • CreatedDate • PackagedBy

		Step 10: The system displays a confirmation alert which informs the system user that the package order details were created successfully.
ALTERNATE COURSES:	**ALT Step 5:** The user does not wish to capture products packaged for an order anymore: • The use case is terminated.	
	ALT Step 6 a: The captured 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 5.**	
	ALT Step 6 b: The product requested for packaging is not associated with the current order's stored order line details: • The validation error details is displayed to the system user. • The use case concludes.	
	ALT Step 6 c: The quantity of the product requested to be packaged is not correct according to what is stored for the order in the **OrderLine** entity: • The validation error details is displayed to the system user. • The use case concludes.	
	ALT Step 7 : The order has been placed by a customer: • Proceed to **Step 9**.	
	ALT Step 8: The system detects that there are insufficient stock of the selected product available on hand to complete the packaging activity: • The validation error details is displayed to the system user. • 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 for the order.	
POST-CONDITION:	The new transactional record's details is added to the **OrderProduct** entity. For a reseller order which has been packaged, the quantity on hand value of the product which has been packaged will be updated in the **Product** 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 value of the selected product for packaging should only be reduced for reseller orders. • The quantity on hand of all ordered products should be reduced at the time of placing a customer order and not when packaging its required products.	

IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• The user has no internet connection and are unable to capture products packaged for a specific order using the system. • Users should be restricted to capture a package activity if there are insufficient stock levels to fulfil the request. • Users will only be allowed to capture one product packaged for an order at a time. • The administrator or sales assistant should not be allowed to capture products packaged if they are not associated with the order for which the packaging operation has been requested. • The system should only allow for the correct quantity to be packaged of each required product as per the order details stored on the system.
ASSUMPTIONS:	• The key stakeholders identified would first like to see what products have already been packaged for the order of interest.
OPEN ISSUES:	• None

Table 89 - 8.21 Send Low Inventory Alert

<table>
<tr><th colspan="4">INVENTORY MANAGEMENT SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Nomusa Vumisa</td><td colspan="2">DATE: 06/06/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 16/06/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Send Low Inventory Alert</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">8.21</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Time</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• Administrator and Inventory Manager</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The threshold value has been reached</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the quantity on hand has reached the specified threshold.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The threshold value has been reached</td><td colspan="2">Step 2: The system reads the following attributes in the Inventory Item table to verify that the quantity on hand has reached the specified threshold

• QuantityOnHand
• ThresholdValue</td></tr>
<tr><td></td><td colspan="2">Step 3: The system generates a low inventory email using the following attributes:

From the User table
• Email

From the Inventory Item table
• InventoryItemName
• QuantityOnHand</td></tr>
<tr><td></td><td colspan="2">Step 4: The system sends a low inventory alert to the administrator and inventory manager</td></tr>
<tr><td>ALTERNATIVE STEPS:</td><td colspan="3"></td></tr>
<tr><td>CONCLUSION:</td><td colspan="3">This use case concludes when alert has been sent</td></tr>
</table>

POST-CONDITION:	The administrator and inventory manager are informed of the stock levels
BUSINESS RULES	• None
IMPLEMENTATION CONTRAINTS 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 90- 8.22 Send Inventory Out of Stock Alert

<table>
<tr><th colspan="4">INVENTORY MANAGEMENT SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Nomusa Vumisa</td><td colspan="2">DATE: 06/06/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 16/06/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>Send Inventory Out of Stock Alert</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>8.22</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Time</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• Administrator and Inventory Manager</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The quantity on hand has run out</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the quantity on hand is out of stock.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The quantity on hand has run out</td><td colspan="2">Step 2: The system reads the following attributes in the Inventory Item table to verify that the quantity on hand has run out

• QuantityOnHand</td></tr>
<tr><td></td><td colspan="2">Step 3: The system generates an out-of-stock email using the following attributes:

From the User table
• Email

From the Inventory Item table
• InventoryItemName
• QuantityOnHand</td></tr>
<tr><td></td><td colspan="2">Step 4: The system sends an out-of-stock alert to the administrator and inventory manager</td></tr>
<tr><td>ALTERNATIVE STEPS:</td><td colspan="3"></td></tr>
<tr><td>CONCLUSION:</td><td colspan="3">This use case concludes when alert has been sent</td></tr>
<tr><td>POST-CONDITION:</td><td colspan="3">The administrator and inventory manager are informed of the stock levels</td></tr>
<tr><td>BUSINESS RULES</td><td colspan="3">• None</td></tr>
</table>

IMPLEMENTATION CONTRAINTS 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 91- 9.1 Add Country

Location SUBSYSTEM			
AUTHORS (s): Ofhani Mungani		**DATE: 06/04/2022**	
VERSION: 3.0		**LAST REVIEW DATE: 21/08/2022**	
USE CASE NAME:	Add Country	**USE CASE TYPE** Business Requirements: ☐ System Analysis: ☒ System Design: ☐	
USE CASE ID:	9.1		
PRIORITY:	High		
SOURCE:	ByteXpress Requirements List		
PRIMARY BUSINESS ACTOR	• Administrator		
PRIMARY SYSTEM ACTOR	• None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• Customers and Resellers – The newly added country is displayed to the customers and resellers during their address registration process.		
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 wishes to add a new country onto the system.		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: The administrator wishes to add a new country.	**Step 2:** The system requests the new country details which have the following attribute: From the **Country** table:	

		• CountryName
	Step 3: The administrator inputs the details. **[ALT]**	**Step 4:** The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. **[ALT]**
	Step 4: The administrator selects the option to add a new country onto the system.	**Step 5**: The system reads from the **Country** table to ensure that the record being added does not match any existing records using the following attribute: • CountryName . **[ALT]**
		Step 6: The system saves the new country details which have the following attributes: • Id • CountryName
		Step 7: The system displays a success message on the screen to confirm the addition of a new country.
ALTERNATE COURSES:	**ALT Step 3**: The administrator wishes not to procced to add a new country. • Terminate Use Case	
	ALT Step 4: The system fails to validate the information provided because of missing fields or incorrectly entered information • Return to Step 3	
	ALT Step 5: The country's name already exists on the system. • Terminate Use Case	
CONCLUSION:	A new instance of Country has been successfully added to the system database.	
POST-CONDITION:	The system now has a new country for customers to choose from when capturing their addresses.	
BUSINESS RULES	Only the administrator can add a new country to the system.	

IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 92- 9.2 Search Country

Location SUBSYSTEM			
AUTHORS (s): Ofhani Mungani		**DATE: 06/04/2022**	
VERSION: 3.0		**LAST REVIEW DATE: 21/08/2022**	
USE CASE NAME:	Search Country	**USE CASE TYPE**	
USE CASE ID:	9.2	Business Requirements:	☐
PRIORITY:	High	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 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 into the system.		
TRIGGER:	The wishes to search for a country onto the system.		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: The administrator requests to search for a country onto the system.		
	Step 2: The administrator inputs the search parameter.	**Step 3:** The system searches the following attributes in the **Country** entity for a match: • CountryName	

		Step 4: The system displays the results to the administrator. **[ALT]**
ALTERNATE COURSES:	**[ALT] Step 4**: The search query entered does not match any existing instances of country in the system database. • Terminate use case	
CONCLUSION:	This use case concludes when a list of matching results from the search query is displayed on the screen.	
POST-CONDITION:	The Administrator can see the country matched from the search query.	
BUSINESS RULES	Only the administrator can search for a country to the system	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	
	•	

Table 93- 9.3 Update Country

<table>
<tr><th colspan="4">Location SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Ofhani Mungani</td><td colspan="2">DATE: 06/04/2022</td></tr>
<tr><td colspan="2">VERSION: 3.0</td><td colspan="2">LAST REVIEW DATE: 21/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>Update Country</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>9.3</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• Customers and Resellers – The newly updated country is displayed to the customers and resellers during their address registration process.</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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 country'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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use is triggered when the administrator wishes to update a country.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE

OF EVENTS:</td><td colspan="2">Actor Action</td><td>System Response</td></tr>
<tr><td colspan="2">Step 1: The administrator requests to update a country on the system</td><td>Step 2: The system invokes use case 9.2 "Search Country".</td></tr>
<tr><td colspan="2">Step 3: The administrator selects the desired country.</td><td>Step 4: The system retrieves the desired country details from the following tables:

From the Country table:</td></tr>
</table>

		• CountryName .
	Step 5: The administrator inputs the details required. **[ALT]**	**Step 6**: The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. **[ALT]**
		Step 7: The system prompts the administrator to confirm the update of a country with the details provided
	Step 8: The administrator confirms the information provided. **[ALT]**	**Step 9:** The system reads from the **Country** table to ensure that the record being updated does not match any existing records. **[ALT]**
		Step 10: The system saves the updated country details which have the following attribute: • CountryName .
		Step 11: The system displays a success message on the screen to confirm the update of a country.
ALTERNATE COURSES:	**ALT Step 5:** The administrator wishes not to procced to update a name of a country. • Terminate Use Case	
	ALT Step 6: The system fails to validate the information provided because of missing fields or incorrectly entered information. • Return to Step 5	
	ALT Step 8: The administrator does not confirm the updating of the country's details: • Return to Step 5 or terminate the use case	

	ALT **Step 9:** The entered country's name already exists on the system. • Terminate Use Case
CONCLUSION:	An instance of Country has been successfully updated to the system database.
POST-CONDITION:	• The system now has an updated country name for customers to choose from when capturing their addresses.
BUSINESS RULES	• Only the administrator can search for a country to the system
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 94- 9.4 Delete Country

Location SUBSYSTEM			
AUTHORS (s): Ofhani Mungani		**DATE: 06/04/2022**	
VERSION: 3.0		**LAST REVIEW DATE: 21/08/2022**	
USE CASE NAME:	Delete Country	**USE CASE TYPE**	
USE CASE ID:	9.4	Business Requirements:	☐
PRIORITY:	High	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 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:	This use case is called when the administrator wishes to delete a country.		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: The administrator wishes to delete a country.	**Step 2:** The system invokes Use Case 9.2 “Search Country”.	
	Step 3: The administrator selects the desired country	**Step 4:** The system retrieves the country information from the database with the following attributes: From the **Country entity** • **CountryName**	

		.
	Step 5: The administrator selects the delete option **[ALT]**	**Step 6:** The system prompts the administrator to confirm the deletion of the selected country.
	Step 7: The administrator confirms the deletion of the country from the system **[ALT]**	**Step 8:** The system removed all the details related to the country from the database **[ALT]**
		Step 9: The system informs the administrator of the successful deletion of the country
ALTERNATE COURSES:	**[ALT] Step 5**: The administrator wishes not to procced to delete a country. • Terminate use case	
	[ALT] Step 7: The administrator does not confirm the deletion of the country. • Return to step 5	
	ALT Step 8: The country cannot be deleted as it has an association with the **province** table • The system displays an error message • Terminate use case	
CONCLUSION:	This use case concludes when the administrator is notified of the successful deletion of a country.	
POST-CONDITION:	• The system deletes all of the details related to the selected country from the **Country entity**.	
BUSINESS RULES	• Only the administrator can delete a country.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 95- 9.5 Create Province

<table>
<tr><th colspan="4">LOCATION SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): APHIWE SHOZI</td><td colspan="2">DATE: 21/08/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 22/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Create Province</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">9.5</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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 country. The administrator will add all of the pertinent information regarding the new province and save it to the business database.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.
• Only province that fall under countries that are already in the system can be added to the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use case is triggered when the administrator wants to create a new Province.</td></tr>
<tr><td rowspan="2">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wants to create a new Province.</td><td colspan="2">Step 2: The system requires the admin to add information about the province in the province entity such as:
• Province Name
• Country Name</td></tr>
</table>

	Step 3: The administrator enters the details. **[ALT]**	**Step 4**: The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. **[ALT]**
		Step 5: The system reads from the **Province** table to ensure that the record being added does not match any existing records using the following attribute: • Name **[ALT]**
		Step 6: The system saves the new Province details which have the following attributes: From the **Province** Table • Id • Name • Country
		Step 7: : The system displays a success message on the screen to confirm the addition of the new province
ALTERNATE COURSES:	**ALT Step 5**: The administrator does not want to add a province anymore • The use case is terminated **ALT Step 4:** The entered details are not in the correct format or the required fields are empty • The system displays a validation error message • Return to step 3 **ALT Step 5:** The province already exists on the system • The system displays a validation error message • The use case is terminated	
CONCLUSION:	This use case concludes when a new province is created.	

POST-CONDITION:	• The Province details are added to the database.
BUSINESS RULES	• Only the administrator can add a new Province.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 96- 9.6 Search Province

<table>
<tr><th colspan="4">LOCATION SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): APHIWE SHOZI</td><td colspan="2">DATE: 22/08/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 22/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>Search Province</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>9.6</td></tr>
<tr><td>PRIORITY:</td><td>Medium</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process in the administrator wishes to search through the existing provinces on the system. The administrator can use the province details for a search query and the system will retrieve the province that matches</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The administrator must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the administrator wants to search for a Province on the system.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wants to search for a Province.</td><td colspan="2"></td></tr>
<tr><td>Step 2: The administrator enters the search parameter.</td><td colspan="2">Step 3: The system searches using the following attributes in the Province table:

• Name</td></tr>
<tr><td></td><td colspan="2">Step 4: The system displays the search results [ALT]</td></tr>
</table>

ALTERNATE COURSES:	**ALT Step 4**: The search query entered does not match any existing instances of Provinces in the database. • Return to Step 2 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	• None
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 97- 9.7 Update Province

LOCATION SUBSYSTEM			
AUTHORS (s): APHIWE SHOZI		**DATE: 22/08/2022**	
VERSION: 1.0		**LAST REVIEW DATE: 22/08/2022**	
USE CASE NAME:	Update Province	**USE CASE TYPE**	
USE CASE ID:	9.7	Business Requirements:	☐
PRIORITY:	High	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 in which the administrator wishes to update the details of province to the system. 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 saves it to the business database..		
PRE-CONDITION:	The administrator must be logged in to the system.		
TRIGGER:	The use is triggered when the administrator updates the details of a Province that already exists on the system.		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: The administrator wishes to update a province's details	**Step 2:** The system invokes Use Case **9.6 Search Province**.	
	Step 3: The administrator selects the desired province	**Step 4**: The system retrieves the desired provinces' details from the following tables From the **Province** Table • Name	

		• Country Name •
	Step 5: The administrator enters the updated details. **[ALT]**	**Step 6**: The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. **[ALT]**
		Step 7: The system requests the administrator to confirm the updating of the province details
	Step 8: The administrator confirms the updating of the Province details **[ALT]**	**Step 9:** The system reads from the **Province** table to ensure that the record being updated does not match any existing records using the following attribute: • Name **[ALT]**
		Step 10: The system saves the updated Province details which have the following attributes: From the **Province** Table • Name • Country Name
		Step 11: The system displays a success message on the screen to confirm the update of a new Province.
ALTERNATE COURSES:	**ALT Step 5**: The administrator does not want to add a province anymore • The use case is terminated **ALT Step 6:** The entered details are not in the correct format or the required fields are empty • The system displays a validation error message • Return to step 5 **ALT Step 8**: The administrator does not confirm the updating of the province details	

	• Return to Step 5 or terminate the use case **ALT Step 9:** The province already exists on the system • The system displays a validation error message • The use case is terminated
CONCLUSION:	This use case concludes when the details of the respective province have been updated successfully on the system
POST-CONDITION:	• The Province details are updated on the database.
BUSINESS RULES	• Only the administrator can update a province.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 98- 9.8 Delete Province

<table>
<tr><th colspan="4">LOCATION SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): APHIWE SHOZI</td><td colspan="2">DATE: 22/08/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 22/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>Delete Province</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>9.8</td></tr>
<tr><td>PRIORITY:</td><td>Medium</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The administrator must be logged in to the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the administrator wishes to delete a Province.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE

OF EVENTS:</td><td colspan="2">Actor Action</td><td>System Response</td></tr>
<tr><td colspan="2">Step 1: The administrator wishes to delete a province.</td><td>Step 2: The system invokes Use Case 9.6 Search Province.</td></tr>
<tr><td colspan="2">Step 3: The administrator selects the desired province</td><td>Step 4: The system retrieves the desired provinces’ details from the following tables

From the Province Table
• Name</td></tr>
</table>

ByteXpress

		• Country Name
	Step 5: The administrator selects the delete option **[ALT]**	**Step 6:** The system prompts the administrator to confirm the deletion of the Province.
	Step 7: The administrator confirms the deletion of the province from the system. **[ALT]**	**Step 8:** The system removes all the details related to the Province from the database from the following tables: From the **Province** Table • Name • Country Name
		Step 9: The system informs the administrator of the successful deletion of the Province
ALTERNATE COURSES:	**ALT Step 5**: The administrator does not want to delete the province • Terminate this use case. **ALT Step 7**: The administrator does not confirm the deletion of the province. • 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.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 99- 9.9 Create City

<table>
<tr><th colspan="4">LOCATION SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): APHIWE SHOZI</td><td colspan="2">DATE: 21/08/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 22/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Create City</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">9.9</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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 products and services to. The use case begins when the administrator adds the new city under its respective province. The administrator will add all of the pertinent information regarding the new city and save it to the business database.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.
• Only city that fall under provinces that are already in the system can be added to the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use case is triggered when the administrator wants to create a new city.</td></tr>
<tr><td rowspan="2">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wants to create a new city.</td><td colspan="2">Step 2: The system requires the admin to add information about the city in the City entity such as:
• City Name
• Province Name</td></tr>
</table>

	Step 3: The administrator enters the details. **[ALT]**	**Step 4**: The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. **[ALT]**
		Step 5: The system reads from the **City** table to ensure that the record being added does not match any existing records using the following attribute: • Name **[ALT]**
		Step 6: The system saves the new city details which have the following attributes: From the **City** Table • Id • Name • Province
		Step 7: : The system displays a success message on the screen to confirm the addition of the new city
ALTERNATE COURSES:	**ALT Step 5**: The administrator does not want to add a city anymore • The use case is terminated **ALT Step 4:** The entered details are not in the correct format or the required fields are empty • The system displays a validation error message • Return to step 3 **ALT Step 5:** The city already exists on the system • The system displays a validation error message • The use case is terminated	
CONCLUSION:	This use case concludes when a new city is created.	
POST-CONDITION:	• The city details are added to the database.	

BUSINESS RULES	• Only the administrator can add a new city.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 100- 9.10 Search City

LOCATION SUBSYSTEM			
AUTHORS (s): APHIWE SHOZI		**DATE: 22/08/2022**	
VERSION: 1.0		**LAST REVIEW DATE: 22/08/2022**	
USE CASE NAME:	Search City	**USE CASE TYPE**	
USE CASE ID:	9.10	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 in the administrator wishes to search through the existing cities on the system. The administrator can use the city details for a search query and the system will retrieve the city that matches		
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	
	Step 1: The administrator wants to search for a city.		
	Step 2: The administrator enters the search parameter.	**Step 3:** The system searches using the following attributes in the **City** table: • Name	
		Step 4: The system displays the search results **[ALT]**	
ALTERNATE COURSES:	**ALT Step 4**: The search query entered does not match any existing instances of cities in the database.		

	• Return to Step 2 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	• None
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 101- 9.11 Update City

<table>
<tr><th colspan="4">LOCATION SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): APHIWE SHOZI</td><td colspan="2">DATE: 22/08/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 22/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>Update City</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>9.11</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process in which the administrator wishes to update the details of city to the system. 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 saves it to the business database..</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The administrator must be logged in to the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use is triggered when the administrator updates the details of a city that already exists on the system.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wishes to update a city’s details</td><td colspan="2">Step 2: The system invokes Use Case 9.10 Search City.</td></tr>
<tr><td>Step 3: The administrator selects the desired city</td><td colspan="2">Step 4: The system retrieves the desired cities’ details from the following tables

From the City Table
• Name</td></tr>
</table>

		• Province Name •
	Step 5: The administrator enters the updated details. **[ALT]**	**Step 6**: The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. **[ALT]**
		Step 7: The system requests the administrator to confirm the updating of the city details
	Step 8: The administrator confirms the updating of the city details **[ALT]**	**Step 9:** The system reads from the **City** table to ensure that the record being updated does not match any existing records using the following attribute: • Name **[ALT]**
		Step 10: The system saves the updated City details which have the following attributes: From the **City** Table • Name • Province Name
		Step 11: The system displays a success message on the screen to confirm the update of a new city.
ALTERNATE COURSES:	**ALT Step 5**: The administrator does not want to add a city anymore • The use case is terminated **ALT Step 6:** The entered details are not in the correct format or the required fields are empty • The system displays a validation error message • Return to step 5 **ALT Step 8**: The administrator does not confirm the updating of the city details	

	• Return to Step 5 or terminate the use case **ALT Step 9:** The city already exists on the system • The system displays a validation error message • The use case is terminated
CONCLUSION:	This use case concludes when the details of the respective city have been updated successfully on the system
POST-CONDITION:	• The City details are updated on the database.
BUSINESS RULES	• Only the administrator can update a city.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 102- 9.12 Delete City

LOCATION SUBSYSTEM			
AUTHORS (s): APHIWE SHOZI		**DATE: 22/08/2022**	
VERSION: 1.0		**LAST REVIEW DATE: 22/08/2022**	
USE CASE NAME:	Delete City	**USE CASE TYPE**	
USE CASE ID:	9.12	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 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	
	Step 1: The administrator wishes to delete a city.	**Step 2:** The system invokes Use Case **9.10 Search City**.	
	Step 3: The administrator selects the desired city	**Step 4**: The system retrieves the desired cities' details from the following tables From the **City** Table • Name • Province Name	

	Step 5: The administrator selects the delete option **[ALT]**	**Step 6:** The system prompts the administrator to confirm the deletion of the city.
	Step 7: The administrator confirms the deletion of the city from the system. **[ALT]**	**Step 8:** The system removes all the details related to the city from the database from the following tables: From the **City** Table • Name • Province Name
		Step 9: The system informs the administrator of the successful deletion of the city
ALTERNATE COURSES:	ALT **Step 5**: The administrator does not want to delete the city • Terminate this use case. ALT **Step 7**: The administrator does not confirm the deletion of the city. • Return to step 4	
CONCLUSION:	This use case concludes when the administrator is notified of the successful deletion of the 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.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 103- 9.13 Create Suburb

LOCATION SUBSYSTEM			
AUTHORS (s): APHIWE SHOZI		**DATE: 21/08/2022**	
VERSION: 1.0		**LAST REVIEW DATE: 22/08/2022**	
USE CASE NAME:	Create Suburb	**USE CASE TYPE**	
USE CASE ID:	9.13	Business Requirements:	☐
PRIORITY:	High	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 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 business database.		
PRE-CONDITION:	• The administrator must be logged onto the system. • Only suburb that fall under cities that are already in the system can be added to the system		
TRIGGER:	The use case is triggered when the administrator wants to create a new suburb.		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: The administrator wants to create a new suburb.	**Step 2**: The system requires the admin to add information about the suburb in the **Suburb entity** such as: • **Suburb Name** • **City Name**	

	Step 3: The administrator enters the details. **[ALT]**	**Step 4**: The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. **[ALT]**
		Step 5: The system reads from the **Suburb** table to ensure that the record being added does not match any existing records using the following attribute: • Name **[ALT]**
		Step 6: The system saves the new suburb details which have the following attributes: From the **Suburb** Table • Id • Name • City
		Step 7: : The system displays a success message on the screen to confirm the addition of the new suburb
ALTERNATE COURSES:	**ALT Step 5**: The administrator does not want to add a suburb anymore • The use case is terminated **ALT Step 4:** The entered details are not in the correct format or the required fields are empty • The system displays a validation error message • Return to step 3 **ALT Step 5:** The suburb already exists on the system • The system displays a validation error message • The use case is terminated	
CONCLUSION:	This use case concludes when a new suburb is created.	

POST-CONDITION:	• The suburb details are added to the database.
BUSINESS RULES	• Only the administrator can add a new suburb.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 104- 9.14 Search Suburb

<table>
<tr><th colspan="4">LOCATION SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): APHIWE SHOZI</td><td colspan="2">DATE: 22/08/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 22/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Search Suburb</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">9.14</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">Medium</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process in the administrator wishes to search through the existing cities on the system. The administrator can use the suburb details for a search query and the system will retrieve the suburb that matches</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The administrator must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the administrator wants to search for a suburb on the system.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wants to search for a suburb.</td><td colspan="2"></td></tr>
<tr><td>Step 2: The administrator enters the search parameter.</td><td colspan="2">Step 3: The system searches using the following attributes in the Suburb table:
• Name</td></tr>
<tr><td></td><td colspan="2">Step 4: The system displays the search results [ALT]</td></tr>
<tr><td>ALTERNATE COURSES:</td><td colspan="3">ALT Step 4: The search query entered does not match any existing instances of cities in the database.</td></tr>
</table>

	• Return to Step 2 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	• None
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 105- 9.15 Update Suburb

LOCATION SUBSYSTEM			
AUTHORS (s): APHIWE SHOZI		**DATE: 22/08/2022**	
VERSION: 1.0		**LAST REVIEW DATE: 22/08/2022**	
USE CASE NAME:	Update Suburb	**USE CASE TYPE**	
USE CASE ID:	9.15	Business Requirements:	☐
PRIORITY:	High	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 in which the administrator wishes to update the details of suburb to the system. 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 saves it to the business database..		
PRE-CONDITION:	The administrator must be logged in to the system.		
TRIGGER:	The use is triggered when the administrator updates the details of a suburb that already exists on the system.		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: The administrator wishes to update a suburb's details	**Step 2:** The system invokes Use Case **9.14 Search Suburb**.	
	Step 3: The administrator selects the desired suburb	**Step 4**: The system retrieves the desired cities' details from the following tables From the **Suburb** Table • Name	

		• City Name •
	Step 5: The administrator enters the updated details. **[ALT]**	**Step 6**: The system captures and validates the information provided by ensuring the format is adhered to and no required text fields are left empty. **[ALT]**
		Step 7: The system requests the administrator to confirm the updating of the suburb details
	Step 8: The administrator confirms the updating of the suburb details **[ALT]**	**Step 9:** The system reads from the **Suburb** table to ensure that the record being updated does not match any existing records using the following attribute: • Name **[ALT]**
		Step 10: The system saves the updated Suburb details which have the following attributes: From the **Suburb** Table • Name • City Name
		Step 11: The system displays a success message on the screen to confirm the update of a new suburb.
ALTERNATE COURSES:	**ALT Step 5**: The administrator does not want to add a suburb anymore • The use case is terminated **ALT Step 6:** The entered details are not in the correct format or the required fields are empty • The system displays a validation error message • Return to step 5 **ALT Step 8**: The administrator does not confirm the updating of the suburb details	

	• Return to Step 5 or terminate the use case ALT **Step 9:** The suburb already exists on the system • The system displays a validation error message • The use case is terminated
CONCLUSION:	This use case concludes when the details of the respective suburb have been updated successfully on the system
POST-CONDITION:	• The Suburb details are updated on the database.
BUSINESS RULES	• Only the administrator can update a suburb.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 106- 9.16 Delete Suburb

LOCATION SUBSYSTEM			
AUTHORS (s): APHIWE SHOZI		DATE: 22/08/2022	
VERSION: 1.0		LAST REVIEW DATE: 22/08/2022	
USE CASE NAME:	Delete Suburb	USE CASE TYPE	
USE CASE ID:	9.16	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 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:	Actor Action	System Response	
	Step 1: The administrator wishes to delete a suburb.	**Step 2:** The system invokes Use Case **9.10 Search Suburb**.	
	Step 3: The administrator selects the desired suburb	**Step 4**: The system retrieves the desired cities' details from the following tables From the **Suburb** Table • Name • City Name	

	Step 5: The administrator selects the delete option **[ALT]**	**Step 6:** The system prompts the administrator to confirm the deletion of the suburb.
	Step 7: The administrator confirms the deletion of the suburb from the system. **[ALT]**	**Step 8:** The system removes all the details related to the suburb from the database from the following tables: From the **Suburb** Table • Name • City Name
		Step 9: The system informs the administrator of the successful deletion of the suburb
ALTERNATE COURSES:	ALT **Step 5**: The administrator does not want to delete the suburb • Terminate this use case. ALT **Step 7**: The administrator does not confirm the deletion of the suburb. • Return to step 4	
CONCLUSION:	This use case concludes when the administrator is notified of the successful deletion of the suburb.	
POST-CONDITION:	• The system deletes all the details related to the selected suburb from the **Suburb table**.	
BUSINESS RULES	• Only the administrator can delete a suburb.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

2.10. Reporting Subsystem

Table 107- 10.1 Generate Inventory List Report

<table>
<tr><th colspan="4">REPORTING SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): APHIWE SHOZI</td><td colspan="2">DATE: 07/08/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 07/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Generate Inventory List Report</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">10.1</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator
• Inventory Manager</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">•</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The administrator or inventory manager must be logged on the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use case is triggered when the administrator or inventory manager wants to generate an inventory list report</td></tr>
<tr><td>TYPICAL COURSE</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
</table>

OF EVENTS:	**Step 1**: The administrator or inventory manager wishes to generate an inventory list report on the system	**Step 2:**The system retrieves the following attributes: From the **InventoryItem** entity: • Name • Quantity on Hand From the **InventoryType** entity: • Name And the following user claims details from the **User** table for the administrator or inventory manager who made the request: • FirstName • Surname
		Step 3: The system calculates the number of Inventory on the system.
		Step 4: The system generates and displays the report with the retrieved attributes and calculated number of Inventory.
	Step 5: The administrator selects the export option	**Step 6:** The system exports the report
ALTERNATE COURSES:	• None	
CONCLUSION:	The use case concludes when the inventory list report is downloaded from the system	
POST-CONDITION:	None	
BUSINESS RULES	• Only the administrator and the inventory manager can generate an inventory list report	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• The administrator and the inventory manager have a stable internet connection	
OPEN ISSUES:	• None	

Table 108- 10.2 Generate Sales Report

<table>
<tr><th colspan="4">REPORTING SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Ofhani Mungani</td><td colspan="2">DATE: 08/08/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 25/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>Generate Sales (by Top 5 Clients) Report</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>10.2</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator/Sales Assistant</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes an event where the administrator or sales assistant wants to generate sales (by top five clients) report on the system. The use case begins when the administrator/ sales assistant requests to generate sales (by top five clients) report and the system responds by retrieving the order details. The use case concludes when the report is displayed on the screen.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator or sales assistant must be logged on the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use case is called when the administrator or sales assistant wants to generate sales (by top five clients) report</td></tr>
<tr><td rowspan="2">TYPICAL COURSE</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator or sales assistant wishes to generate sales (by top five clients) report on the system</td><td colspan="2">Step 2: The system retrieves the following attributes:

from the OrderLine entity:

• Price
• Count</td></tr>
</table>

		from the **Order** entity: • FirstName • Surname • OrderTotal • DeliveryFee • CreatedDate from the **Product** entity: • Name
		Step 3: The system calculates the dashboard sales values on the system
		Step 4: The system displays the sales dashboard.
	Step 5: The administrator or sales assistant selects an option to display sales (by top 5 clients) report	**Step 6:** The system retrieves the following attributes: from the **Order** entity: • FirstName • Surname • OrderTotal • DeliveryFee And the following claims details from the **User** entity for the administrator or sales assistant who made the request • FirstName • Surname
		Step 7: The system calculates the sales by top 5 client's values on the system
		Step 8: The system generates and displays the report with the retrieved attributes and sales by top 5 client's values.

	Step 9: The administrator or sales assistant selects an option to export the report.	**Step 10:** The system exports the sales (by top 5 clients) report
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 CONTRAINTS 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 109 - 10.3 Generate Product List Report

<table>
<tr><th colspan="4">REPORTING SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Thenjiwe Ntsonda</td><td colspan="2">DATE: 13/08/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 13/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Generate Product List Report</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">10.3</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator or Inventory Manager</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process in which the administrator 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 downloads the report.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The administrator must be logged onto the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use case is triggered when the administrator wants to generate a product list report</td></tr>
<tr><td rowspan="2">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator or inventory manager wishes to generate a product list report</td><td colspan="2">Step 2: The system retrieves the following attributes from the Product table:
• Name
• QuantityOnHand
• ResellerPrice
• CustomerPrice

From the ProductBrand table:
• Name

From the ProductCategory table:
• Name</td></tr>
</table>

		And the following user claims details from the **User** table for the administrator or inventory manager who made the request: • FirstName • Surname
		Step 3: The system calculates the total product quantity on the system using the **QuantityOnHand** attribute from the Product table.
		Step 4: The system generates and displays the report with the retrieved attributes and calculated product quantity
	Step 5: The administrator or inventory manager selects the export option	**Step 6:** The system exports the report
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 CONTRAINTS AND SPECIFICATIONS	Only the administrator or inventory manager has access to this function	
ASSUMPTIONS:	The administrator or inventory manager will always export the report	
OPEN ISSUES:	None	

Table 110- 10.4 Generate Customer List Report

<table>
<tr><th colspan="4">REPORTING SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Nomusa Vumisa</td><td colspan="2">DATE: 08/08/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 08/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Generate Customer List Report</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">10.4</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process in which the administrator wishes to generate a customer 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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The administrator must be logged onto the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use case is triggered when the administrator wants to generate a customer report</td></tr>
<tr><td rowspan="2">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wishes to generate a customer report</td><td colspan="2">Step 2: The system retrieves the following attributes from the User table:
• FirstName
• Surname
• PhoneNumber
• EmailAddress

From the UserRole table:
• Name</td></tr>
</table>

		And the following user claims details from the **User** table for the administrator who made the request: • FirstName • Surname
		Step 3: The system calculates the number of customers on the system.
		Step 4: The system generates and displays the report with the retrieved attributes and calculated number of customers.
	Step 5: The administrator selects the export option	**Step 6:** The system exports the report
ALTERNATE COURSES:	• None	
CONCLUSION:	This use case concludes when the report is downloaded	
POST-CONDITION:	The **User** and **UserRole** table were read	
BUSINESS RULES	None	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	Only the administrator has access to this function	
ASSUMPTIONS:	The administrator will always export the report	
OPEN ISSUES:	None	

Table 111- 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 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 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 report		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: The administrator wishes to generate a reseller report	**Step 2**: The system retrieves the following attributes from the **User** table: • FirstName • Surname • PhoneNumber • EmailAddress From the **UserRole** table: • Name	

		And the following user claims details from the **User** table for the administrator who made the request: • FirstName • Surname
		Step 3: The system calculates the number of resellers on the system.
		Step 4: The system generates and displays the report with the retrieved attributes and calculated number of resellers.
	Step 5: The administrator selects the export option	**Step 6:** The system exports the report
ALTERNATE COURSES:	• None	
CONCLUSION:	This use case concludes when the report is downloaded	
POST-CONDITION:	The **User** and **UserRole** table were read	
BUSINESS RULES	None	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	Only the administrator has access to this function	
ASSUMPTIONS:	The administrator will always export the report	
OPEN ISSUES:	None	

Table 112- 10.6 Generate Produced Products Report

REPORTING SUBSYSTEM			
AUTHORS (s): Kyle van Eeden		DATE: 06/08/2022	
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:	☐
PRIORITY:	Low	System Analysis:	☒
SOURCE:	ByteXpress Requirements List	System Design:	☐
PRIMARY BUSINESS ACTOR	• Administrator or Inventory Manager		
PRIMARY 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 requests the system to generate a report for all finished products produced on the system. The administrator or inventory manager will be allowed to optionally specify a filter criteria to be applied to the report and the system filters the production records retrieved from the database, if any, to retrieve only the details which are to be used for the report to be generated. The use case concludes when the system displays the generated production report which is based on the filter criteria entered.		
PRE-CONDITION:	The administrator or inventory manager must be logged into the system.		
TRIGGER:	The administrator or inventory manager requests the system to display the latest production report details.		
TYPICAL COURSE OF EVENTS:	Actor Action	System Response	
	Step 1: The administrator or inventory manager requests the system to display the latest production report.		

	Step 2: The administrator or inventory manager requests the system to generate the report. **[ALT]**	**Step 3:** The system retrieves the following details from the **ProductionTransaction** entity for all produced products transactions: • ProductId • QuantityBefore • QuantityAfter • TransactionDate The system also retrieves the following details from the **Product** entity for each product referenced in a retrieved production transaction record: • ProductName The system also retrieves the following user claims details from the **User** entity for the administrator or inventory manager who made the request: • Name • Surname **[ALT]**
		Step 4: The system calculates the production quantity for each production transaction record retrieved by subtracting the retrieved **QuantityAfter** attribute value from the **QuantityBefore** value.
		Step 5: The system displays the production report, which is populated with the details of all production transactions retrieved from the database.
	Step 6: The administrator or inventory manager selects the option to export the report.	**Step 7:** The system exports the generated report details to the user's local machine.
ALTERNATE COURSES:	**ALT** Step 2: The administrator or inventory manager wishes wishes to apply a filter criteria to the production report: • The user specifies the desired filter criteria using the filter options provided.	

	• The administrator or inventory manager selects the option provided to generate the report. • Proceed to **Step 3**.
	ALT Step 3: The administrator or inventory manager has specified a filter criteria for the report to be generated: • The system retrieves only the production transaction records from the **ProductionTransaction** entity which has has met the report filter criteria specified. • Proceed to **Step 4**.
CONCLUSION:	The system displays the production report's details which are filtered according to the filter criteria specified.
POST-CONDITION:	• The production transaction records which meet the filter criteria specified are retrieved from the **ProductionTransaction** entity. • The generated report is populated with all the retrieved product production transaction records.
BUSINESS RULES	• Only the inventory manager and administrator users should be allowed to the view this report. • The administrator or inventory manager should be allowed to export the production report in PDF format.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• The system should allow the administrator or inventory manager to specify a filter criteria for the report to be generated. • This report should be exported upon the user's request, and as a PDF document only. • The production report cannot be generated if the system is offline.
ASSUMPTIONS:	• The administrator or inventory manager has access to the internet. • The administrator or inventory manager would want to export the report after viewing it on the system.
OPEN ISSUES:	• None

Table 113- 10.7 Generate Orders (By Status) Report

Reporting Subsystem			
AUTHORS (s): Kyle van Eeden		**DATE: 06/08/2022**	
VERSION: 1.1		**LAST REVIEW DATE: 25/08/2022**	
USE CASE NAME:	Generate Orders (By Status) Report	**USE CASE TYPE**	
USE CASE ID:	10.7	Business Requirements:	☐
PRIORITY:	Low	System Analysis:	☒
SOURCE:	ByteXpress Requirements List	System Design:	☐
PRIMARY BUSINESS ACTOR	• Administrator		
PRIMARY 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 latest orders by status report on the system. All orders will be retrieved from the database and grouped by status and order payment status. The total count of orders per status will be displayed along with accumulated totals for each grouping of orders. The use case concludes when the system displays the generated report details.		
PRE-CONDITION:	The administrator must be logged into the system. The administrator must be logged into their Power BI account. The administrator's Power BI account should have access to the Natuurlik Reporting workspace.		
TRIGGER:	The administrator user wishes to view the latest orders by status report on the system.		
TYPICAL COURSE	Actor Action	System Response	

OF EVENTS:	**Step 1**: The administrator requests the system to display the latest report on orders categorized by various statuses.	**Step 2**: The system retrieves the following details for all orders that have been placed on the system from the **Order** entity: • OrderStatus • OrderPaymentStatus • OrderTotal
		Step 3: The system displays the generated report that is retrieved and containing the following report details which have been generated: The total number of orders assigned to each distinct **OrderStatus** attribute value are displayed. The accumulated totals are displayed for each grouping of orders, with [OrderStatus] as the parent group and [OrderPaymentStatus] as the child group, using the following attribute: OrderTotal
ALTERNATE COURSES:	• None	
CONCLUSION:	The system displays the generated report details which gives an overview the status all orders are assigned to currently.	
POST-CONDITION:	• The latest orders data is retrieved from the **Order** entity.	
BUSINESS RULES	• Only the administrator of the system should be allowed to view this report. • The administrator should be able to view a report where all orders are grouped by status and payment status.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• The orders should be categorized by order status and order payment status and the accumulated totals per order grouping should be calculated and displayed in the generated report. • The report's data will only be refreshed every 2 hours until the ByteXecom system has been deployed.	

	• Only administrators with the required Power BI Service license and access to the Natuurlik Reporting workspace will be able to view this report.
ASSUMPTIONS:	• The administrator has access to the internet and the system is not offline. • The administrator has their Power BI Service account credentials readily available. • The administrator would like to see a detailed report on which orders are still within a specific status and/or what the payment status of each order is.
OPEN ISSUES:	• None

Table 114- 10.8 Generate Sales (By Region) Report

<table>
<tr><th colspan="4">Reporting Subsystem</th></tr>
<tr><td colspan="2">AUTHORS (s): Kyle van Eeden</td><td colspan="2">DATE: 30/08/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 30/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Generate Sales (By Region) Report</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">10.8</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">Low</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process in which the administrator wishes to view the latest sales by region report on the system. All orders will be retrieved from the database along with associated location details stored in the Province and City entities, respectively, and the orders are grouped by province and city. The system displays the generated report on screen using the latest available details retrieved.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The administrator must be logged into the system.
The administrator must be logged into their Power BI account with a Premium Per User capacity.
The administrator’s Power BI account should have access to the Natuurlik Reporting workspace that has been set up.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The administrator user wishes to view the latest sales by region report
on the system.</td></tr>
<tr><td>TYPICAL COURSE</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
</table>

OF EVENTS:	**Step 1**: The administrator requests the system to display the latest report on all orders categorized by various regions.	**Step 2**: The system retrieves the following details from the below listed entities in the database: From the **Order** entity: • OrderID • OrderTotal • OrderStatus From the **Province** entity: • ProvinceName From the **City** entity: • CityName
		Step 3: The system displays the generated report containing the following report details which have been generated: The total sales revenue calculated for each province is displayed. The total count of orders are calculated by using the following attributes to group orders by: • ProvinceID • CityID The accumulated totals are displayed for each grouping of orders by province and city by using the following attribute : • OrderTotal

		Only orders where the OrderStatus attribute value is equal to "Processing" or "Dispatched"
ALTERNATE COURSES:	• None	
CONCLUSION:	The system displays the generated transactional report details where all orders placed on the system are grouped by province and city.	
POST-CONDITION:	• The latest orders data is retrieved from the **Order** entity. • A report is generated using the retrieved orders data where all orders are grouped by	
BUSINESS RULES	• Only the administrators of the system who has access to the Power BI workspace should be allowed to view this report. • The administrators should be able to view a report where all orders are grouped by provinces and cities.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• The orders should be categorized by province and city. The accumulated totals per order grouping should be calculated and displayed in the generated report. • The report's data will only be refreshed every 5 minutes until the ByteXecom system has been deployed. • Only administrators with the required Power BI Service license and access to the Natuurlik Reporting workspace will be able to view this report.	
ASSUMPTIONS:	• The administrator has access to the internet and the system is not offline. • The administrator has their Power BI Service account credentials readily available. • The administrator would like to see a detailed report on which orders are still within a specific status and/or what the payment status of each order is.	
OPEN ISSUES:	• None	

Table 115- 10.9 Generate Sales (by Product) Report

<table>
<tr><th colspan="4">REPORTING SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Ofhani Mungani</td><td colspan="2">DATE: 08/08/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 25/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Generate Sales (by Product) Report</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">10.9</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator/Sales Assistant</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator or sales assistant must be logged on the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use case is called when the administrator or sales assistant wants to generate sales (by Product) report</td></tr>
<tr><td rowspan="2">TYPICAL COURSE</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator or sales assistant wishes to generate sales (by Product) report on the system</td><td colspan="2">Step 2: The system retrieves the following attributes:

from the OrderLine entity:

• Price
• Count

from the Order entity:</td></tr>
</table>

		• FirstName • Surname • OrderTotal • DeliveryFee • CreatedDate from the **Product** entity: • Name
		Step 3: The system calculates the dashboard sales values on the system
		Step 4: The system displays the sales dashboard.
	Step 5: The administrator or sales assistant selects an option to display sales (by Product) report	**Step 6:** The system retrieves the following attributes: from the **OrderLine** entity: • Price • Count from the **Product** entity: • Name And the following claims details from the **User** entity for the administrator or sales assistant who made the request • FirstName • Surname
		Step 7: The system calculates the sales by product values on the system
		Step 8: The system generates and displays the report with the retrieved attributes and sales by product values.
	Step 9: The administrator or sales assistant selects an option to export the report.	**Step 10:** The system exports the sales (by product) report
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 CONTRAINTS 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 116- 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	
USE CASE ID:	10.10	Business Requirements:	☐
PRIORITY:	High	System Analysis:	☒
SOURCE:	ByteXpress Requirements List	System Design:	☐
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 Product) report		
TYPICAL COURSE	Actor Action	System Response	
	Step 1: The administrator or sales assistant wishes to generate sales (by month) report on the system	**Step 2:** The system retrieves the following attributes: from the **OrderLine** entity: • Price • Count from the **Order** entity:	

		• FirstName • Surname • OrderTotal • DeliveryFee • CreatedDate from the **Product** entity: • Name
		Step 3: The system calculates the dashboard sales values on the system
		Step 4: The system displays the sales dashboard.
	Step 5: The administrator or sales assistant selects an option to display sales (by month) report	**Step 6:** The system retrieves the following attributes: from the **Order** entity: • OrderTotal • DeliveryFee • CreatedDate And the following claims details from the **User** entity for the administrator or sales assistant who made the request • FirstName • Surname
		Step 7: The system calculates the sales by product values on the system
		Step 8: The system generates and displays the report with the retrieved attributes and sales by product values.
	Step 9: The administrator or sales assistant selects an option to export the report.	**Step 10:** The system exports the sales (by month) report
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 CONTRAINTS 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 117- 10.11 View Dashboard Overview

<table>
<tr><th colspan="4">Reporting Subsystem</th></tr>
<tr><td colspan="2">AUTHORS (s): Kyle van Eeden</td><td colspan="2">DATE: 30/08/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 02/09/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>View Dashboard Overview</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>10.11</td></tr>
<tr><td>PRIORITY:</td><td>Low</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• Power BI Service (ESA)</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process in which the administrator wishes to view the Natuurlik Business Intelligence Dashboard using the latest data retrieved from the database. A report will be generated using the retrieved data and the use case concludes once the system displays the dashboard overview.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The administrator must be logged into the system.
The administrator must be logged into their Power BI account with a Premium Per User capacity.
The administrator’s Power BI account should have access to the Natuurlik Reporting workspace that has been set up.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The administrator user wishes to view the Natuurlik dashboard overview.</td></tr>
<tr><td>TYPICAL COURSE</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
</table>

OF EVENTS:	**Step 1**: The administrator requests the system to display the latest dashboard.	**Step 2**: The system retrieves the following details from the below listed entities in the database: From the **Order** entity: • OrderID • OrderTotal • OrderStatus • OrderPaymentStatus • IsResellerOrder • BackOrderDate • DispatchedDate From the **OrderLine** entity: • OrderID • ProductID • CartItemQuantity • CartItemPrice From the **OrderProduct** entity: • OrderID • ProductID • ProductQuantity • TransactionDate From the **OrderQuery** entity: • OrderID • QueryFeedback • QueryStatus • LoggedDate From the **QueryReason** entity: • Description From the **ReturnedProduct** entity: • OrderID • ProductID • QuantityReceived • ReturnReasonID • DateLogged

		From the **ReturnReason** entity: • ReturnReasonName From the **User** entity: • FirstName • Surname From the **Country** entity: • CountryName From the **Province** entity: • ProvinceName From the **City** entity: • CityName From the **Suburb** entity: • SuburbName From the **Product** entity: • ProductName From the **Category** entity: • CategoryName From the **Brand** entity: • BrandName From the **Courier** entity: • CourierName • CourierFee
		Step 3: The system displays the retrieved dashboard.

ALTERNATE COURSES:	• None	
CONCLUSION:	The system displays the generated dashboard details.	
POST-CONDITION:	• The latest orders data is retrieved from the **Order** entity. • All associated orders transaction details is retrieved from the **Courier**, **OrderLine** and **Product** entities. • 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 administrators of the system who has access to the Power BI workspace should be allowed to view this report. • The administrators should be able to view a dashboard overview which summarizes all business activities.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• The report's data will only be refreshed every 2 hours until the ByteXecom system has been deployed. • Only administrators with the required Power BI Service license and access to the Natuurlik Reporting workspace will be able to view this dashboard.	
ASSUMPTIONS:	• The administrator has access to the internet and the system is not offline. • The administrator has their Power BI Service account credentials readily available if not logged in already.	
OPEN ISSUES:	• None	

2.11. Supplier Subsystem

Table 118 - 11.1 Create Supplier

<table>
<tr><th colspan="4">SUPPLIER SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Nomusa Vumisa</td><td colspan="2">DATE: 20/08/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 20/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Create Supplier</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">11.1</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.
• The supplier must not already exist on the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use case is triggered when the administrator wants to create a new supplier.</td></tr>
<tr><td rowspan="2">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wants to create a new supplier.</td><td colspan="2">Step 2: The system requests the new supplier details which have the following attributes:

From the Supplier Table
• CompanyName
• EmailAddress
• PhoneNumber</td></tr>
</table>

		• StreetName From the **Suburb** Table • SuburbName From the **City** Table • CityName From the **Province** Table • ProvinceName From the **Country** Table • CountryName
	Step 3: The administrator enters the details. **[ALT]**	**Step 4**: The system captures and validates the entered details **[ALT]**
		Step 5: The system reads from the **Supplier** table to ensure that the record being added does not match any existing records using the following attribute: • CompanyName • EmailAddress **[ALT]**
		Step 6: The system saves the new supplier details which have the following attributes: From the **Supplier** Table • Id • CompanyName • EmailAddress • PhoneNumber • StreetName • SuburbId

		Step 7: The system displays a successful creation message
ALTERNATE COURSES:	ALT **Step 5**: The administrator does not want to add a Supplier anymore • The use case is terminated ALT **Step 4:** The entered details are not in the correct format or the required fields are empty • The system displays a validation error message • Return to step 3 ALT **Step 5:** The supplier already exists on the system • The system displays a validation error message • 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 CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 119 - 11.2 Search Supplier

<table>
<tr><th colspan="4">SUPPLIER SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Nomusa Vumisa</td><td colspan="2">DATE: 14/05/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 14/05/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Search Supplier</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">11.2</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">Medium</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The administrator must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the administrator wants to search for a supplier on the system.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wants to search for a supplier.</td><td colspan="2"></td></tr>
<tr><td>Step 2: The administrator enters the search parameter.</td><td colspan="2">Step 3: The system searches using the following attributes in the Supplier table:
• CompanyName
• EmailAddress
• PhoneNumber</td></tr>
<tr><td></td><td colspan="2">Step 4: The system displays the search results [ALT]</td></tr>
<tr><td>ALTERNATE COURSES:</td><td colspan="3">ALT Step 4: The search query entered does not match any existing instances of suppliers in the database.</td></tr>
</table>

	• Return to Step 2 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 CONTRAINTS AND SPECIFICATIONS	• None
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 120 - 11.3 Update Supplier

SUPPLIER SUBSYSTEM			
AUTHORS (s): Nomusa Vumisa		**DATE: 14/05/2022**	
VERSION: 1.0		**LAST REVIEW DATE: 14/05/2022**	
USE CASE NAME:	Update Supplier	**USE CASE TYPE**	
USE CASE ID:	11.3	Business Requirements: ☐	
PRIORITY:	High	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 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	
	Step 1: The administrator wishes to update a supplier's details	**Step 2:** The system invokes Use Case **11.2 Search Supplier**.	
	Step 3: The administrator selects the desired supplier	**Step 4**: The system retrieves the desired suppliers' details from the following tables From the **Supplier** Table • CompanyName	

		• EmailAddress • PhoneNumber • StreetName From the **Suburb** Table • SuburbName From the **City** Table • CityName From the **Province** Table • ProvinceName From the **Country** Table • CountryName
	Step 5: The administrator enters the updated details. **[ALT]**	**Step 6**: The system captures and validates the entered details **[ALT]**
		Step 7: The system requests the administrator to confirm the updating of the supplier details
	Step 8: The administrator confirms the updating of the supplier details **[ALT]**	**Step 9:** The system reads from the **Supplier** table to ensure that the record being updated does not match any existing records using the following attribute: • CompanyName • EmailAddress **[ALT]**
		Step 10: The system saves the updated supplier details which have the following attributes: From the **Supplier** Table • CompanyName • EmailAddress • PhoneNumber • StreetName

		• SuburbId
		Step 11: The system displays a successful update message
ALTERNATE COURSES:	**ALT Step 5**: The administrator does not want to add a Supplier anymore • The use case is terminated **ALT Step 6:** The entered details are not in the correct format or the required fields are empty • The system displays a validation error message • Return to step 5 **ALT Step 8**: The administrator does not confirm the updating of the supplier details • Return to Step 5 or terminate the use case **ALT Step 9:** The supplier already exists on the system • The system displays a validation error message • The use case is terminated	
CONCLUSION:	This use case concludes when the supplier details are updated.	
POST-CONDITION:	• The supplier details are updated on the database.	
BUSINESS RULES	• Only the administrator can update a supplier.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 121 - 11.4 Delete Supplier

<table>
<tr><th colspan="4">SUPPLIER SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Nomusa Vumisa</td><td colspan="2">DATE: 14/05/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 14/05/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Delete Supplier</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">11.4</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">Medium</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The administrator must be logged in to the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the administrator wishes to delete a supplier.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wishes to delete a supplier.</td><td colspan="2">Step 2: The system invokes Use Case 11.2 Search Supplier.</td></tr>
<tr><td>Step 3: The administrator selects the desired supplier</td><td colspan="2">Step 4: The system retrieves the desired suppliers’ details from the following tables

From the Supplier Table</td></tr>
</table>

		• CompanyName • EmailAddress • PhoneNumber • StreetName From the **Suburb** Table • SuburbName From the **City** Table • CityName From the **Province** Table • ProvinceName From the **Country** Table • CountryName
	Step 5: The administrator selects the delete option **[ALT]**	**Step 6:** The system prompts the administrator to confirm the deletion of the supplier.
	Step 7: The administrator confirms the deletion of the supplier from the system. **[ALT]**	**Step 8:** The system reads from the **InventoryProcured** table to ensure that the record being deleted does not have an association with Procured Inventory using the following attribute: • Supplier_Id **[ALT]**
		Step 9: The system removes all the details related to the supplier from the database from the following tables: From the **Supplier** Table • CompanyName • EmailAddress • PhoneNumber • StreetName • SuburbId

		Step 10: The system informs the administrator of the successful deletion of the supplier
ALTERNATE COURSES:	**ALT Step 5**: The administrator does not want to delete the supplier • Terminate this use case. **ALT Step 7**: The administrator does not confirm the deletion of the supplier. • Return to step 4 **ALT Step 8**: The supplier cannot be deleted as it has an association with the Procure Inventory Table • The system displays an error message • 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 CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 122 - 11.5 Send Supplier Order

SUPPLIER SUBSYSTEM			
AUTHORS (s): Kyle van Eeden		**DATE: 27/06/2022**	
VERSION: 1.1		**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:	ByteXpress Requirements List	System Design:	☐
PRIMARY BUSINESS ACTOR	• Administrator		
PRIMARY SYSTEM ACTOR	• None		
OTHER PARTICIPATING ACTORS:	• None		
OTHER INTERESTED STAKEHOLDERS:	• None		
DESCRIPTION:	The use case begins when a user with the Administrator role assigned wishes to send a supplier an email that contains all of the required inventories and their required quantities. The administrator select a specific supplier to send the request to and the system displays a form on which all the supplier order details is to be captured. The administrator should be able to search all inventories defined on the system to add all relevant items along with their required quantities. After the input provided has been validated, the system sends an email containing all the supplier order details to the selected supplier's email address. The use case concludes when the system confirms that the email has been sent successfully.		
PRE-CONDITION:	At least one Inventory Item has to be defined on the system to send a supplier order email. The supplier to send the supplier order to must be defined on the system.		
TRIGGER:	This use case is called when a system administrator wishes to send a supplier order email to a specific supplier within Natuurlik's suppliers list.		
TYPICAL COURSE	Actor Action	System Response	
	Step 1: The system administrator wishes to	**Step 2:** The system displays the following details for all the suppliers retrieved from the **Supplier** entity and prompts the system administrator to	

	request inventories from one of Natuurlik's suppliers.	select a single supplier to send the email to: Details to be read / Attributes in the table / Notes: Supplier Name / CompanyName / Name of each supplier retrieved. Supplier Email Address / SupplierEmail / Email address of each supplier.
	Step 3: The administrator wishes to search for the specific supplier they wish to contact and enters a value to search by.	**Step 4:** The system filters the suppliers retrieved from the **Supplier** entity by only displaying the details of suppliers that have matched the search criteria specified.
	Step 5: The system administrator selects the option to send a supplier order email to a specific supplier. **[ALT]**	**Step 6:** The system displays the following details retrieved from the the **Supplier** entity for the specific supplier requested: Details to be read / Attributes in the table / Notes: Supplier Name / CompanyName / Name of each supplier retrieved. Supplier Email Address / SupplierEmail / Email address of each supplier. Supplier Contact Number / SupplierPhoneNumber / Phone number for each supplier. The system also retrieves all inventory items that are associated with the selected supplier's last order request sent, if any, by retrieving the following details from the **SupplierInventoryItem** entity: Details to be read / Attributes in the table / Notes: Inventory Item Name / InventoryItemID / Referenced foreign key that uniquely identifies each associated

		inventory item.
Inventory Item Order Quantity	OrderQuantity	Quantity of each inventory item last ordered.

The system retrieves the names of all associated inventory items from the **InventoryItem** entity:

Details to be read	Attributes in the table	Notes
Inventory Item Name	InventoryItemName	Name of each inventory item retrieved.

Step 7: The administrator enters a search criteria value to look up an Inventory Item they wish to order and that is not included in the list of inventory items displayed.

[ALT]

Step 8: The system displays the following details for all inventory items where the name attribute value matched the search query entered Items from the **InventoryItem** entity:

Details to be read	Attributes in the table	Notes
Inventory Item Name	InventoryItemName	Name of each inventory item that matched the search query.

Step 9: The administrator selects the Inventory Item they wish to include in the supplier order email.

Step 10: The system adds the selected inventory item to the supplier's list of inventories if it was not included before and retrieves the updated list to display the name of the added inventory item.

[ALT]

Step 11: The administrator enters all required details and the quantities of each inventory item they wish to request from the selected supplier and selects the option to send the supplier order email.

[ALT]

Step 12: The system validates that all the provided details are correct as per the identified validation requirements.

		[ALT]
		Step 13: The system generates an email containing the following captured supplier order details: • Supplier Email (Recipient) • Subject • Message • Inventory Item Name • Inventory Item Quantity
		Step 14: The system sends the supplier order email to the selected supplier's email address.
		Step 15: The system displays an alert confirming that the supplier order request email has been sent successfully.
ALTERNATE COURSES:	**ALT Step 5:** The administrator no longer wishes to send a supplier order request to an existing supplier or the required supplier's details is not yet defined on the system: • The administrator clicks the option provided to cancel the operation: • The use case is terminated.	
	ALT Step 7 a: The administrator does not wish to search for any other Inventory Item to add to the supplier order request: • Proceed to **Step 11**.	
	ALT Step 7 b: The administrator wishes to remove an Inventory Item included in the table of inventories and selects the option provided to remove a specific item: • The system removes the Inventory Item temporarily from the list of Inventory Items. • Proceed to **Step 11**.	
	ALT Step 10: The system detects that the Inventory Item selected already exists in the inventories list: • Proceed to **Step 11**.	
	ALT Step 11: The administrator no longer wishes to send the supplier order email and selects the option to cancel the operation: • Return to **Step 2**.	
	ALT Step 12: The system detects validation errors in the details captured on the form submitted and raises validation errors for all required fields without a value and/or fields containing incorrect data: • The system displays the validation errors raised for each field. • Return to **Step 11.**	

CONCLUSION:	This use case concludes when the system displays a confirmation notification which informs the administrator that the supplier order email has been sent successfully.
POST-CONDITION:	The updated inventory items list is saved to the **SupplierInventoryItem** entity.
BUSINESS RULES	• Only the system administrator roles should be permitted to send emails to suppliers using this functionality. • The list of inventory items to be requested from a supplier should be specified by the system administrators.
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• Remember the inventory items list that has been requested from a specific supplier and populate the list automatically for the next supplier order. • The supplier order email cannot be sent as the system is offline or the administrator has no stable internet connection.
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

2.12. Driver Subsystem

Table 123- 12.1 View Deliveries

<table>
<tr><th colspan="4">DRIVER SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Ofhani Mungani</td><td colspan="2">DATE: 30/08/2023</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 30/08/2023</td></tr>
<tr><td>USE CASE NAME:</td><td>View Deliveries</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>12.1</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Delivery Driver</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The main system server should be online.
• The order status should be dispatched.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use case is called when the delivery driver wants to view orders to deliver.</td></tr>
<tr><td rowspan="2">TYPICAL COURSE

OF EVENTS:</td><td colspan="2">Actor Action</td><td>System Response</td></tr>
<tr><td colspan="2">Step 1: The delivery driver wants to view orders to deliver on the mobile application.</td><td>Step 2: The system displays the order details using the following attributes:
From the Order Table
• OrderId
• FirstName
• Surname
• PhoneNumber
• StreetAddress
• CreatedDate</td></tr>
</table>

		From the **Suburb** Table • SuburbName • PostalCode From the **City** Table • CityName From the **Province** Table • ProvinceName From the **Country** Table • CountryName
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 CONTRAINTS 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 124- 12.2 View Previous Deliveries

<table>
<tr><th colspan="4">DRIVER SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Ofhani Mungani</td><td colspan="2">DATE: 30/08/2023</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 30/08/2023</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">View Previous Deliveries</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">12.2</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Delivery Driver</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The main system server should be online.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use case is called when the delivery driver wants to view previous deliveries.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The delivery driver wants to view previous deliveries on the mobile application.</td><td colspan="2">Step 2: The system invokes Use Case 12.1 View Delivery</td></tr>
<tr><td>Step 3: The delivery driver selects an option to view previous deliveries.</td><td colspan="2">Step 4: The system displays the delivery details using the following attributes:
From the Order Table
• OrderId
From the Delivery Table</td></tr>
</table>

		Date
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.	
IMPLEMENTATION CONTRAINTS 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 125- 12.3 Confirm Deliveries

<table>
<tr><th colspan="4">DRIVER SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Ofhani Mungani</td><td colspan="2">DATE: 30/08/2023</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 30/08/2023</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Confirm Deliveries</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">12.3</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Delivery Driver</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The main system server should online.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use case is called when the delivery driver wants to confirm the completion of an order delivery.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The delivery driver wants to confirm the completion of an order delivery.</td><td colspan="2">Step 2: The system invokes Use Case 12.1 View Delivery</td></tr>
<tr><td>Step 3: The delivery driver selects an option to complete a delivery.</td><td colspan="2">Step 4: The system updates order status to delivered using the following attributes:
From the Order table
• OrderStatus</td></tr>
</table>

		Step 5: The system Creates a new delivery using the following attributes: Into the **Delivery** table • Id • Date • OrderId and displays a success message on the screen to confirm the completion of an order delivery.
	Step 6: The delivery driver selects an option to dismiss the success message	
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 CONTRAINTS AND SPECIFICATIONS	• The delivery driver cannot confirm deliveries 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 126 - 13.1 Create VAT Amount

<table>
<tr><th colspan="4">VAT SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Aphiwe Shozi</td><td colspan="2">DATE: 06/062022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 06/062022</td></tr>
<tr><td>USE CASE NAME:</td><td>Create VAT Amount</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>13.1</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process in which the administrator wishes to add a new VAT Amount onto the system. The use case begins when the administrator wishes to add a VAT Amount. The administrator will proceed to capture the VAT Amount details and save it to the system database.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use case is triggered when the administrator wants to create a new VAT Amount.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wants to create a new VAT Amount.</td><td colspan="2">Step 2: The system requires the admin to add information about the VAT Amount in the VAT Table such as :
• Created Date
• Percentage
• Is Active</td></tr>
<tr><td>Step 3: The administrator provides the required the details. [ALT]</td><td colspan="2">Step 4: The system captures and validates the entered details
[ALT]</td></tr>
<tr><td></td><td colspan="2">Step 5: The system reads from the VAT table to ensure that the record being created does not match any existing records using the following attribute:</td></tr>
</table>

		• Id • Created Date • Percentage • Is Active **[ALT]**
		Step 6: The system saves the new VAT Amount details which have the following attributes: From the **VAT Table** • Id • Created Date • Percentage • Is Active **[ALT]**
		Step 7: The system save the new VAT Amount to the VAT Table
ALTERNATE COURSES:	**ALT Step 3**: The administrator does not want to add a VAT Amount anymore • The use case is terminated **ALT Step 4:** The entered details are not in the correct format or the required fields are empty • The system displays a validation error message • Return to step 3 **ALT Step 5:** The VAT Amount is already exists on the system • The system displays a validation error message • The use case is terminated	
CONCLUSION:	This use case concludes when a new VAT Amount is created.	
POST-CONDITION:	• The VAT Amount details are added to the database.	
BUSINESS RULES	• Only the administrator can add a new VAT Amount.	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Table 127 - 13.2 Search VAT Amount

VAT SUBSYSTEM			
AUTHORS (s): Aphiwe Shozi		**DATE: 06/06/2022**	
VERSION: 1.1		**LAST REVIEW DATE: 24/07/2022**	
USE CASE NAME:	Search VAT	**USE CASE TYPE**	
USE CASE ID:	13.2	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 in the administrator wishes to perform a search on all the existing VAT instances' details on the system. The administrator can search for VAT details defined on the system by entering a search query based on either the VAT Value or VAT Status values assigned to all instances. The use case concludes once the system has filtered the records and displays only the details of those VAT records which have matched the specified 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 VAT Amount on the system.		
TYPICAL COURSE OF EVENTS:	**Actor Action**	**System Response**	
	Step 1: The administrator wishes to search for existing VAT details defined on the system.	**Step 2**: The system displays the following details for all VAT instances retrieved from the VAT table: • VAT Value • VAT Status	
	Step 3: The administrator enters the search parameter in the field provided.	**Step 4**: The system filters the details by only displaying the details of those VAT instances where the details have matched the search query provided. **[ALT]**	
ALTERNATE COURSES:	**ALT Step 4**: The search query entered does not match the details of any existing instances defined on the system: • The system indicates that no results have been found. • The use case is terminated.		
CONCLUSION:	This use case concludes when only the details are displayed of those VAT instances where its details have matched the search query specified.		
POST-CONDITION:	Only the details of VAT instances that have matched the search query provided is displayed.		

BUSINESS RULES	• Only administrator users should be allowed to search for VAT details defined on the system.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• The system is offline or the administrator user does not have access to the internet, resulting in the user being unable to search for VAT details on the system.
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 128 - 13.3 Update VAT

<table>
<tr><th colspan="4">VAT SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Aphiwe Shozi</td><td colspan="2">DATE: 06/06/2022</td></tr>
<tr><td colspan="2">VERSION: 1.1</td><td colspan="2">LAST REVIEW DATE: 24/07/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Update VAT Details</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">13.3</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• System Owner</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">This use case describes the process in which the administrator wishes to update the details of existing VAT instances on the system. The use case begins when the administrator requests the system to update the details of an existing VAT instance. The system will then prompt the administrator to enter the updated details and after having entered the details, the system will prompt the administrator to confirm the updated details. The captured input is then validated and if validation has passed, the system will save the updated details to the VAT table. This use case concludes when the system displays an alert confirming that the updated VAT details have been saved successfully.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">• The administrator must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use is triggered when the administrator requests the system to update the details of an existing VAT instance.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wishes to update a VAT instance’s details.</td><td colspan="2">Step 2: The system invokes Use Case 13.2 Search VAT.</td></tr>
<tr><td>Step 3: The administrator selects the option provided to update the desired VAT record’s details.</td><td colspan="2">Step 4: The system retrieves the VAT instance’s details which has been requested to be updated from the following tables:

From the VAT Table:

• VAT Value
• VAT Status

The system also prompts the administrator to capture the updated details for the VAT instance.</td></tr>
</table>

	Step 5: The administrator enters the updated VAT details. **[ALT]**	**Step 6**: The system captures and validates the entered details. **[ALT]**
		Step 7: The system prompts the administrator to confirm the updated VAT details provided.
	Step 8: The administrator selects the option to proceed with the update operation. **[ALT]**	**Step 9:** The system reads the following attribute values for all VAT instances from the **VAT** table: • VAT Value • VAT Status The retrieved details is used to ensure that the updated VAT Value captured is unique across all existing VAT instances. The system will also perform a check to ensure that only one instance has a status of 'Active' assigned if the VAT instance's status has been changed from 'Inactive' to 'Active'. **[ALT]**
		Step 10: The system saves the updated VAT details for the requested instance in the **VAT** table: • VAT Value • VAT Status
		Step 11: The system displays an alert that confirms the VAT details have been updated successfully.
ALTERNATE COURSES:	**ALT Step 5**: The administrator does not wish to update the VAT details anymore: • The administrator selects the option provided to cancel the update operation. • The use case is terminated.	
	ALT Step 6: The entered details are not in the correct format or the required fields are empty: • The system displays a validation error message for each field for which a validation error was raised. • Return to **Step 5**.	
	ALT Step 8: The administrator does not confirm to proceed with the updating of the VAT instance's details: • The administrator selects the option provided to cancel the update operation. • The use case is terminated.	

	ALT Step 9 a: The VAT Value value already exists on the system for another VAT instance: • The system displays a validation error message informing the administrator that the VAT Value already exists. • The use case is terminated
	ALT Step 9 b: Another VAT instance already has a status value of 'Active' assigned: • The system displays a validation error message informing the administrator that only one VAT instance may be active at a time. • The use case is terminated
CONCLUSION:	This use case concludes when the details of the respective VAT record has been updated successfully on the system and the system has displayed a message confirming the update was performed successfully.
POST-CONDITION:	The requested VAT instance's details is updated in the **VAT** table.
BUSINESS RULES	• Only administrator users should be allowed to update a VAT instance's details.
IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS	• The system is offline or the administrator user does not have access to the internet, resulting in the user being unable to update the details of a VAT instance defined on the system.
ASSUMPTIONS:	• None
OPEN ISSUES:	• None

Table 129 - 13.4 Delete VAT

<table>
<tr><th colspan="4">VAT SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Aphiwe Shozi</td><td colspan="2">DATE: 06/06/2022</td></tr>
<tr><td colspan="2">VERSION: 1.1</td><td colspan="2">LAST REVIEW DATE: 24/07/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>Delete VAT</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>13.4</td></tr>
<tr><td>PRIORITY:</td><td>Medium</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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 instance to be deleted and request the system to remove its details. The system will display the current details stored for the requested VAT instance and the administrator be prompted to confirm the removal of the VAT instance. Once the administrator confirms to proceed with the removal, the system will remove all of the details stored for the particular instance from the VAT table. The use case concludes once the system displays an alert which confirms that the VAT instance’s details has been removed successfully.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The administrator must be logged in to the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the administrator wishes to delete a VAT instance from the system.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wishes to delete a VAT instance.</td><td colspan="2">Step 2: The system invokes Use Case 11.2 Search VAT Amount.</td></tr>
<tr><td>Step 3: The administrator selects option to delete the desired VAT instance from the system.</td><td colspan="2">Step 4: The system retrieves the requested VAT instance’s details from the following table

From the VAT Table :
• VAT Value
• VAT Status
• Created Date</td></tr>
<tr><td>Step 5: The administrator selects the delete option provided.</td><td colspan="2">Step 6: The system prompts the administrator to confirm the removal of the requested VAT instance.</td></tr>
</table>

<table>
<tr><td rowspan="4"></td><td>[ALT]</td><td></td></tr>
<tr><td>Step 7: The administrator selects the option to confirm the removal of the VAT details from the system.
[ALT]</td><td>Step 8: The system retrieves the following details from the Order table for all existing orders:
• VAT ID

The retrieved information is used to determine whether the VAT instance requested for removal has been associated with any orders on the system.

[ALT]</td></tr>
<tr><td></td><td>Step 9: The system removes the following details for the requested VAT instance from the VAT table:

• VAT ID
• VAT Value
• VAT Status
• Created Date</td></tr>
<tr><td></td><td>Step 10: The system informs the administrator of the successful deletion of the VAT details by displaying a confirmation alert.</td></tr>
<tr><td rowspan="3">ALTERNATE COURSES:</td><td colspan="2">ALT Step 5: The administrator does not wish to delete the VAT instance from the system anymore:
• The administrator selects the option provided to cancel the delete operation.
• The use case is terminated.</td></tr>
<tr><td colspan="2">ALT Step 7: The administrator does not wish to proceed with the removal of the requested VAT instance’s details:
• The administrator selects the option provided to cancel the delete operation.
• The use case is terminated.</td></tr>
<tr><td colspan="2">ALT Step 8: The VAT instance’s details cannot be removed from the system since it is associated with an order which has been placed on the system:
• The system displays a validation error message informing the administrator why the VAT details cannot be removed.
• The use case is terminated</td></tr>
<tr><td>CONCLUSION:</td><td colspan="2">This use case concludes when the system displays an alert which confirms the VAT details has been removed successfully.</td></tr>
<tr><td>POST-CONDITION:</td><td colspan="2">• The system has removed all the VAT record’s details from the VAT table.</td></tr>
<tr><td>BUSINESS RULES</td><td colspan="2">• Only the administrator should be allowed to delete a VAT instance from the system.</td></tr>
</table>

IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• The system should restrict administrator users from removing VAT details which have been associated with orders placed on the system to ensure referential integrity. • The system is offline or the administrator does not have access to the internet, resulting in the administrator 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 130 - 14.1 Update Reseller Credit Settlement Time

<table>
<tr><th colspan="4">RESELLER CREDIT SETTLEMENT TIME SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Nomusa Vumisa</td><td colspan="2">DATE: 14/05/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 14/05/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>Update Reseller Credit Settlement Time</td><td colspan="2" rowspan="4">USE CASE TYPE

Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>14.1</td></tr>
<tr><td>PRIORITY:</td><td>High</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The administrator must be logged onto the system</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">The use case is triggered when the administrator wants to update the reseller credit settlement time</td></tr>
<tr><td rowspan="4">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator wishes to update the reseller credit settlement time</td><td colspan="2">Step 2: The system retrieves the following attribute from the PaymentReminder table:
• Days</td></tr>
<tr><td>Step 3: The administrator selects their desired settlement time interval</td><td colspan="2">Step 4: The system requests the administrator to confirm the updating of the settlement time</td></tr>
<tr><td>Step 5: The administrator confirms the updating of the settlement time
[ALT]</td><td colspan="2">Step 6: The system reads from the PaymentReminder table to ensure that the record being updated is not already active using the following attribute:
• IsActive
[ALT]</td></tr>
</table>

		Step 7: The system activates the selected settlement time using the following attributes: From the **PaymentReminder** Table • IsActive
		Step 8: The system displays a successful creation message
ALTERNATE COURSES:	**ALT Step 5**: The administrator does not confirm the updating of the settlement time • The use case is terminated	
	ALT Step 6: The settlement time is already active on the system • The system displays an error message • The use case is terminated	
CONCLUSION:	This use case concludes when the settlement time is updated.	
POST-CONDITION:	The settlement time was updated in the PaymentReminder Table	
BUSINESS RULES	• Only the administrator can update the settlement time	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• Only one settlement time can be active at a time • The settlement time cannot be updated without an internet connection	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

2.15. Video Subsystem

Table 131- 15.1 Add Video

<table>
<tr><th colspan="5">VIDEO SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Thenjiwe Ntsonda</td><td colspan="3">DATE: 18/04/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="3">LAST REVIEW DATE: 23/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Add Video</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☐
System Design: ☒</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">15.1</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXcom Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="4">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="4">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="4">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="4">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="4">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="4">• The administrator must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="4">The use case is triggered when the administrator wants to add a new video.</td></tr>
<tr><td rowspan="3">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION</td><td colspan="3">SYSTEM RESPONSE</td></tr>
<tr><td>MANUAL ACTION</td><td colspan="2">AUTOMATED ACTION</td></tr>
<tr><td>Step 1: The administrator navigates to the Video Screen under the Video tab on the side bar</td><td></td><td colspan="2">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:

<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>Video Link</td><td>VideoUrl</td></tr></table>
And displays the Video Screen with the following controls:

<table><tr><th>Control Name</th><th>Control</th><th>Notes</th></tr></table></td></tr>
</table>

				Type	
			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 “” 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 Video** button

Step 4: The system displays the **Create Video** screen with the following controls:

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

<table>
<tr>
<td rowspan="3"></td>
<td></td>
<td></td>
<td>
<table>
<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>

Note: All controls are enabled by default unless stated otherwise.
</td>
</tr>
<tr>
<td>Step 5: The system administrator enters the details and clicks on the Create button
[ALT]</td>
<td></td>
<td>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>
</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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 Notes</th></tr>
<tr><td>Name</td><td>Name</td></tr>
<tr><td>Video Link</td><td>VideoUrl</td></tr>
</table>
</td>
</tr>
</table>

Step 8: The system saves the information into the **Video** table using an INSERT SQL query:

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:

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
-

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

Control Name	Control Name	Notes
Name	Textbox	This control will be outline in red to show a validation error

<table>
<tr><td></td><td>
<table>
<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>
Return to step 3
</td></tr>
<tr><td>CONCLUSION:</td><td>This use case concludes when a video has been added on the system</td></tr>
<tr><td>POST-CONDITION:</td><td>A new user has been added to the Video table in the database</td></tr>
<tr><td>BUSINESS RULES</td><td>Only the administrator has access to this function.</td></tr>
<tr><td>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</td><td>The system is offline, and the system administrator will not be able to add a new video.</td></tr>
<tr><td>ASSUMPTIONS:</td><td>None</td></tr>
<tr><td>OPEN ISSUES:</td><td>None</td></tr>
</table>

Table 132- 15.2 Search Video

<table>
<tr><th colspan="5">VIDEO SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Thenjiwe Ntsonda</td><td colspan="3">DATE: 18/04/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="3">LAST REVIEW DATE: 23/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Search Video</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☐
System Design: ☒</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">15.2</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">Medium</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXcom Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="4">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="4">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="4">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="4">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="4">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="4">• The administrator must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="4">This use case is called when the administrator wants to search for a video on the system.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION</td><td colspan="3">SYSTEM RESPONSE</td></tr>
<tr><td>MANUAL ACTION</td><td colspan="2">AUTOMATED ACTION</td></tr>
<tr><td>Step 1: The administrator wishes to search for a video on the system.</td><td></td><td colspan="2"></td></tr>
<tr><td>Step 2: The administrator navigates to the Video Screen under the Video tab on the side bar</td><td></td><td colspan="2">Step 3: The system retrieves following information using an SQL READ QUERY

From the Video 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>Video Link</td><td>VideoUrl</td></tr></table>
And displays the Video Screen with the following controls:
<table><tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr></table></td></tr>
</table>

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 "" 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 **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 133- 15.3 Update Video

<table>
<tr><th colspan="5">VIDEO SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Thenjiwe Ntsonda</td><td colspan="3">DATE: 18/04/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="3">LAST REVIEW DATE: 23/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">Update Video</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☐
System Design: ☒</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">15.3</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">Medium</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXcom Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="4">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="4">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="4">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="4">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="4">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="4">• The administrator must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="4">The use case is triggered when the administrator wants to update a video.</td></tr>
<tr><td rowspan="4">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION</td><td colspan="3">SYSTEM RESPONSE</td></tr>
<tr><td>MANUAL ACTION</td><td colspan="2">AUTOMATED ACTION</td></tr>
<tr><td>Step 1: The administrator wishes to update a video in the video catalogue.</td><td></td><td colspan="2">Step 2: The system invokes use case 15.2 Search Video</td></tr>
<tr><td>Step 2: The administrator clicks on the Edit button</td><td></td><td colspan="2">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:
<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>Video Link</td><td>VideoUrl</td></tr></table>
And displays the Update Video Screen with the following controls:</td></tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td>
<table>
<tr><th>Control Name</th><th>Control Type</th><th>Notes</th></tr>
<tr><td>Update 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 update 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 update a video link of the video</td></tr>
<tr><td>Update</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 edit of the video</td></tr>
</table>
Note: All controls are enabled by default unless stated otherwise.
</td>
</tr>
<tr>
<td></td>
<td>Step 4: The system administrator updates the video information and clicks on the Update button [ALT]</td>
<td></td>
<td>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>
</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Step 6: The system uses an SQL READ query to check if the video already exists in the Video table using the following attributes</td>
</tr>
</table>

<table>
<tr>
<td rowspan="3"></td>
<td></td>
<td></td>
<td>
[ALT]
<table>
<tr><th>Details to be updated</th><th>Attributes in the table Notes</th></tr>
<tr><td>Name</td><td>Name</td></tr>
<tr><td>Video Link</td><td>VideoUrl</td></tr>
</table>
</td>
</tr>
<tr>
<td></td>
<td></td>
<td>
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>
</td>
</tr>
<tr>
<td>Step 8: The system administrator clicks on the Confirm button [ALT]</td>
<td></td>
<td>
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>
</td>
</tr>
</table>

			Step 10: The system redirects the system administrator to the **Video 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 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

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
-

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 134- 15.4 Delete Video

<table>
<tr><th colspan="5">VIDEO SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Thenjiwe Ntsonda</td><td colspan="3">DATE: 18/04/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="3">LAST REVIEW DATE: 23/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="3">Delete Video</td><td rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☐
System Design: ☒</td></tr>
<tr><td>USE CASE ID:</td><td colspan="3">15.4</td></tr>
<tr><td>PRIORITY:</td><td colspan="3">Medium</td></tr>
<tr><td>SOURCE:</td><td colspan="3">ByteXcom Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="4">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="4">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="4">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="4">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="4">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="4">• The administrator must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="4">The use case is triggered when the administrator wants to delete a video.</td></tr>
<tr><td rowspan="5">TYPICAL COURSE OF EVENTS:</td><td rowspan="3">ACTOR ACTION</td><td colspan="3">SYSTEM RESPONSE</td></tr>
<tr><td colspan="3"></td></tr>
<tr><td>MANUAL ACTION</td><td colspan="2">AUTOMATED ACTION</td></tr>
<tr><td>Step 1: The administrator wishes to delete a video</td><td></td><td colspan="2">Step 2: The system invokes use case 15.2 Search Video</td></tr>
<tr><td></td><td></td><td colspan="2">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><th>Details to be read</th><th>Attributes in the table Notes</th></tr><tr><td>Name</td><td>Name</td></tr><tr><td>Video Link</td><td>VideoUrl</td></tr></table></td></tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td></td>
<td>
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>
</td>
</tr>
<tr>
<td></td>
<td>Step 4: The administrator clicks on the Delete button</td>
<td></td>
<td>
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</td><td>Label</td><td>This control is used to request</td></tr>
</table>
</td>
</tr>
</table>

<table>
<tr><td></td><td></td><td></td><td>
<table>
<tr><td>delete this video?</td><td></td><td>confirmation of the deletion of the video</td></tr>
<tr><td>Confirm</td><td>Button</td><td>This control is used to confirm the deletion of the video details</td></tr>
<tr><td>Cancel</td><td>Button</td><td>This control is used to close the message box deletion of the video</td></tr>
</table>
</td></tr>
<tr><td></td><td>Step 6: The system administrator clicks on the Confirm button [ALT]</td><td></td><td>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 Notes</th></tr>
<tr><td>VideoId</td><td>VideoId</td></tr>
<tr><td>Name</td><td>Name</td></tr>
</table>
</td></tr>
<tr><td></td><td></td><td></td><td>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>
</td></tr>
<tr><td>ALTERNATE COURSES:</td><td colspan="3">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</td></tr>
<tr><td>CONCLUSION:</td><td colspan="3">This use case concludes when a video has been deleted on the system</td></tr>
<tr><td>POST-CONDITION:</td><td colspan="3">The video has been removed from the system database.</td></tr>
<tr><td>BUSINESS RULES</td><td colspan="3">Only the administrator has access to this function.</td></tr>
<tr><td>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</td><td colspan="3">None</td></tr>
<tr><td>ASSUMPTIONS:</td><td colspan="3">None</td></tr>
<tr><td>OPEN ISSUES:</td><td colspan="3">None</td></tr>
</table>

Table 135- 15.5 View Video Details

<table>
<tr><th colspan="5">VIDEO SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Thenjiwe Ntsonda</td><td colspan="3">DATE: 08/06/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="3">LAST REVIEW DATE: 23/08/2022</td></tr>
<tr><td>USE CASE NAME:</td><td colspan="2">View Video Details</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis:
System Design:</td></tr>
<tr><td>USE CASE ID:</td><td colspan="2">15.5</td></tr>
<tr><td>PRIORITY:</td><td colspan="2">High</td></tr>
<tr><td>SOURCE:</td><td colspan="2">ByteXcom Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="4">• User</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="4">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="4">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="4">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="4">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="4">• None</td></tr>
<tr><td>TRIGGER:</td><td colspan="4">The use case is triggered when the user wishes to view the video(s) on the system</td></tr>
<tr><td rowspan="3">TYPICAL COURSE OF EVENTS:</td><td rowspan="2">ACTOR ACTION</td><td colspan="3">SYSTEM RESPONSE</td></tr>
<tr><td>MANUAL ACTION</td><td colspan="2">AUTOMATED ACTION</td></tr>
<tr><td>Step 1: The user clicks on the About Us nav link</td><td></td><td colspan="2">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

<table><tr><th>Details to be read</th><th>Attributes in the table</th></tr><tr><td>Name</td><td>Name</td></tr><tr><td>Video Link</td><td>VideoUrl</td></tr></table>

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</td></tr></table></td></tr>
</table>

					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 retrieved from the

<table>
<tr><td></td><td></td><td></td><td><table><tr><td></td><td></td><td>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></td></tr>
<tr><td></td><td>Step 3: The user clicks on the video they wish to play</td><td></td><td>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.</td></tr>
<tr><td>ALTERNATE COURSES:</td><td colspan="3">• None</td></tr>
<tr><td>CONCLUSION:</td><td colspan="3">This use case concludes when the inventory procured details are displayed on the system</td></tr>
<tr><td>POST-CONDITION:</td><td colspan="3">The selected video is played on the system.</td></tr>
<tr><td>BUSINESS RULES</td><td colspan="3">• None</td></tr>
<tr><td>IMPLEMENTATION CONSTRAINTS AND SPECIFICATIONS</td><td colspan="3">• None</td></tr>
<tr><td>ASSUMPTIONS:</td><td colspan="3">• The user will always choose to play the selected video</td></tr>
<tr><td>OPEN ISSUES:</td><td colspan="3">• None</td></tr>
</table>

2.16 Help Subsystem

Table 136- 16.1 View Help Document

<table>
<tr><th colspan="4">HELP SUBSYSTEM</th></tr>
<tr><td colspan="2">AUTHORS (s): Nomusa Vumisa</td><td colspan="2">DATE: 03/09/2022</td></tr>
<tr><td colspan="2">VERSION: 1.0</td><td colspan="2">LAST REVIEW DATE: 03/09/2022</td></tr>
<tr><td>USE CASE NAME:</td><td>View Help Document</td><td colspan="2" rowspan="4">USE CASE TYPE
Business Requirements: ☐
System Analysis: ☒
System Design: ☐</td></tr>
<tr><td>USE CASE ID:</td><td>16.1</td></tr>
<tr><td>PRIORITY:</td><td>Medium</td></tr>
<tr><td>SOURCE:</td><td>ByteXpress Requirements List</td></tr>
<tr><td>PRIMARY BUSINESS ACTOR</td><td colspan="3">• Administrator</td></tr>
<tr><td>PRIMARY SYSTEM ACTOR</td><td colspan="3">• None</td></tr>
<tr><td>OTHER PARTICIPATING ACTORS:</td><td colspan="3">• None</td></tr>
<tr><td>OTHER INTERESTED STAKEHOLDERS:</td><td colspan="3">• None</td></tr>
<tr><td>DESCRIPTION:</td><td colspan="3">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.</td></tr>
<tr><td>PRE-CONDITION:</td><td colspan="3">The administrator must be logged onto the system.</td></tr>
<tr><td>TRIGGER:</td><td colspan="3">This use case is called when the administrator wants to view a help document on the system.</td></tr>
<tr><td rowspan="5">TYPICAL COURSE

OF EVENTS:</td><td>Actor Action</td><td colspan="2">System Response</td></tr>
<tr><td>Step 1: The administrator or sales assistant want to view a help document</td><td colspan="2"></td></tr>
<tr><td>Step 2: The administrator enters the search parameter</td><td colspan="2">Step 3: The system searches for a match</td></tr>
<tr><td></td><td colspan="2">Step 4: The system displays the search results [ALT]</td></tr>
<tr><td>Step 5: The administrator selects the help document they wish to view</td><td colspan="2">Step 6: The system retrieves the help document</td></tr>
</table>

		Step 7: The system displays the selected help document
ALTERNATE COURSES:	**ALT Step 4**: The search query entered does not match any existing instances of an order. • Return to Step 2 or terminate the use case	
CONCLUSION:	This use case concludes when the desired help document is displayed on the system.	
POST-CONDITION:	The data table was searched and the help document was retrieved	
BUSINESS RULES	• None	
IMPLEMENTATION CONTRAINTS AND SPECIFICATIONS	• None	
ASSUMPTIONS:	• None	
OPEN ISSUES:	• None	

Conclusion

In conclusion, the ByteXpress team compiled a comprehensive list of the logical flow of their use cases through the logical narratives. These narratives include all of the business processes carried out by the system.

3. Iteration Conclusion

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