ByteXpress - Team 9
ByteXecom E-commerce System

Aphiwe Shozi	Kyle van Eeden	Nomusa Vumisa (TL)	Ofhani Mungani	Thenjiwe Ntsonda (PM)
u19363967	u18035176	u17254061	u18022571	u18139958
u19363967@tuks.co.za	u18035176@tuks.co.za	u17254061@tuks.co.za	u18022571@tuks.co.za	u18139958@tuks.co.za
0827135234	0614700577	0662254267	0766495693	0817473388

ITERATION - 3

Technical Specification - For this iteration, the team intends to demonstrate the technical interactions of various parts of our proposed system in detail.

The document includes the latest requirements list, the complete set of physical use case diagrams, physical entity relationship diagram along with a preliminary description of the hardware and software needed for the development and the eventual operation of our system, the initial SQL database setup and a description of the design principles used in our interfaces.

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 Figures

Table of Figures

1. Iteration Introduction

For iteration three we the ByteXpress team have compiled the technical specifications of our proposed system that we are developing for our client, Natuurlik. Upon the completion of the functional specifications the team is prepared to analyse the ByteXecom E-commerce system in more detail and incorporate the feedback we received from our previous iteration to produce a stronger system. This analysis of proposed system includes our updated requirements list, a technical use case diagram and entity relationship diagram in 3NF as well as a preliminary description for our design principles and hardware and software requirements.

2. Hardware and Software Requirements

Introduction

In this section of the iteration ByteXpress provides a comprehensive overview of the minimum and recommended hardware and software requirements needed for the proposed ByteXecom system. For the ByteXecom system to perform optimally we have provided the following hardware recommendation based on what is readily available for the client side of our system.

Software Requirements	Client Side		Server Side	
	Minimum	Recommended	Minimum	Recommended
Database Management server	N/A	N/A	SQL Server 2014 or and Azure SQL	QL Server 2019 Or Azure SQL
Operating system	Microsoft Windows 8.1 and Mac OS	Microsoft Windows 11 and Mac OS	Windows Server 2012	Windows Server 2012
Web Browser	Chromium v6, Firefox 74,Opera, Internet Explorer 9	Google Chrome, Microsoft Edge, Safari and FireFox76	N/A	N/A
PDF Viewer	Embedded PDF viewer in web browser	Latest version of ADOBE Acrobat Reader DC	N/A	N/A
Development Environment	N/A	N/A	Visual Studio 2015 community.Net Framework 5	Visual Studio 2022 Community Edition, .NET Framework 6
Email Client	Gmail (Google Email Service)	Gmail (Google Email Service) or Microsoft Outlook 365	Internal Email services provided by Visual Studios 2015	Internal Email services provided by Visual Studios 2022
Web Server	N/A	N/A	Microsoft Azure	Microsoft Azure
Interactive Data			Latest Power BI version, Windows Server 2016, Windows 10, 32-bit platform	Latest Power BI version, Windows Server 2019, Windows 11, 64-bit platform

Table 1- Software requitements

Hardware Requirements	Client Side		Server Side	
	Minimum	Recommended	Minimum	Recommended
HDD	2GB Available hard drive space	3GB Available hard drive space	70GB Available hard drive space	100GB Available hard drive space
RAM	2GB	4GB	8GB	16GB
CPU	Intel celeron or Ryzen 3-1300	Intel Core i3-7020U or Ryzen 5 - 2500	Intel Core i5 - 6600 or Ryzen 5 - 1500x	Intel Core i5 - 9400F or Ryzen 5 - 2500
GPU	DirectX 9 compatible device	DirectX 11 compatible device	N/A	N/A
Input devices	Keyboard and mouse	Keyboard and mouse	Keyboard, mouse and a document scanner	Keyboard, mouse and a document scanner
Output Devices	Computer/Laptop monitor or mobile device screen	Computer/Laptop monitor or mobile device screen	Computer/Laptop monitor or mobile device screen and a printer	Computer/Laptop monitor or mobile device screen and a printer
Internet connection	5Mb/s upload and download speed	10Mb/s upload and download	40Mb/s upload and download	80Mb/s upload and download
Networking Hardware	Built-In Network Controller	Built-In Network Controller	Built-In Network Controller	Built-In Network Controller

Table 2 - Hardware requirements

2.1. Development Environment Requirements

The following software is to be used during the development phase of the proposed ByteXecom system:

- Microsoft SQL Server Management Studio 2018
- Web Browsers (Chrome, IE and Firefox)
- Microsoft Visual Studio 2022 IDE
- Microsoft Office 365
- Microsoft Power BI Desktop 2022
- Visual Studio Code
- Postman
- Microsoft SQL Server 2019
- Microsoft Azure

Visual Studio 2022 must have the following additional software installed and configured to enable the successful development of the ByteXecom system's required functionality:

- .NET 6 (Long-term-support)
- Angular (v 11 or later)
- Microsoft SQL Server Reporting Services (SSRS)

2.2. Production Environment Requirements

The following software is used to configure and set up the production environment's database:

- Azure SQL Database
- SQL Server 2019 or later versions

To publish the web project to Microsoft cloud so it is accessible for the intended end-users, the following software services will be used for implementing the deployment of the ByteXecom system:

- **Azure App Service** (Windows version only) with .NET 6 as the target framework.
- Publish business intelligence reporting solution to an application in **Power BI Service**.

2.3. Recommended Hardware and Software for System Configuration

2.3.1. Server-Side Setup

- Reliable Internet Connection
- 8 GB RAM
- 3.40 GHz CPU or faster 64-bit processor
- 64-bit OS
- 512 GB SSD
- Keyboard and mouse
- Windows 10 version 1909 or later releases
- VGA supported display
- Windows Server 2019 or later

2.3.2. Client-Side Setup

- 2.60 GHz or faster CPU
- 512 GB HDD
- GPU with minimum of 1 GB VRAM
- Wired or wireless internet connection
- 8 GB RAM
- Mouse and keyboard
- HDMI Display with a resolution of 1920 x 1080

2.4. Payment Gateway Design Using Stripe

The following will give a high-level overview into how Stripe is to be integrated with the ByteXecom web application to enable customers to pay for goods they wish to order.

In order to configure and start testing payments on the system, default details for payment testing will be used which is provided by Stripe. This includes card details that allow for successful as well as unsuccessful payments.

All customers are expected to pay for goods requested before placing an order on the system. Once, successful a successful payment has been made, the customer’s cart will be checked out and their order’s details will be created and stored in the Order entity.

The three endpoints to be used for processing payments on the system include:

- Initiate Payment: Initiate payment processing by submitting a request to make an online payment using the Stripe payment gateway.
- Redirect: Redirect end-users of the web application to the correct pages based on certain events. The landing page and payment page will be used by the ByteXecom solution.
- Transaction Response: All transactional information can be retrieved, including the transaction status, result code and other relevant information pertaining to a particular transaction so that a transaction can be validated and confirmed.

Payment Processing Overview using Stripe:

The customer will have the option to initiate the process of paying for the order they wish to place on the system. The relevant page should have an option that can capture the request to make the payment for the order.

The user is then to be redirected to the “Stripe Payment Page” where the following details is displayed within the allocated sections:

Payment Overview Header:

- Merchant Name
- Payment Reference
- Product Details (Including the prices of all products)
- Delivery Fee
- VAT Inclusive amount
- Total Amount
- Cardholder Details:
- Card Holder Name (as per card to be used)
- Card Number
- Expiry Month
- Expiry Year
- CVV Code

A session is then to be created with Stripe for payment processing using either Visa or MasterCard as the only valid payment options. The only currency to be supported is “ZAR” and only South African card holders can submit a payment request.

The following cart details of the customer requesting to make a payment are retrieved from the **Cart** entity:

- CartItemPrice: The price per each distinct product in the customer’s cart.
- CartItemQuantity: The quantity of each distinct product in the customer’s cart being checked out.
- The details of all products in the customer’s cart are retrieved from the Product entity and include the following attribute values:
- ProductName
- ProductDescription
- CustomerPrice (Used to assign the CartItemPrice value when adding items to cart initially)

Upon checking out a customer’s cart, a courier has to be selected, and the following attribute value is retrieved from the Courier entity to determine the courier service charge to be added to the order’s total:

DeliveryFee: The fee that is to be charged based on the chosen courier service.

The inclusive vat amount is also to be calculated by the ByteXecom system and added to the order’s total which is to be processed by Stripe. The current VAT details applicable on the system at the time of requesting to make a payment is retrieved from the VAT entity:

VATAmount: The total of the order, including the delivery fee, will be multiplied by the VATAmount attribute value retrieved and then divided by a 100.

The OrderTotal value calculated by the ByteXecom system includes the sum of all products’ prices, the quantities ordered, the delivery fee and the total value added tax amount applicable to the order. This order total is then sent to stripe for processing.

Payment Outcomes:

- Payment Successful: Redirect the customer to Order Confirmation page so that the application can confirm that the payment was successful, and the order has been captured on the system. The customer’s cart is cleared upon successfully making a payment for their order.
- Payment Unsuccessful: In the event of payment failing the customer will be redirected to the “Payment Failed” page which will inform the user that the payment details they have provided were not able to be used for successful payment of the order total due.
- Cancel Payment: Redirect the customer back to their current shopping cart if they wish to cancel the payment for the order.

A payment intent ID will be stored for each order which has been successfully placed, to allow Natuurlik staff to query whether payment has indeed been received by confirming the payment in the Stripe account’s portal or by querying a payment transaction.

In order to successfully make a payment for an order, a valid MasterCard or Visa card number is required, along with the expiry date of the card (MM/YYYY) and a CVV number.

Integrated South African Banks that are supported in terms of making an online payment include any banks that process MasterCard and Visa cards.

Information to be sent to Stripe for payment processing:

- Order Total
- Cardholder Name
- Card Number
- Card Expiry Date
- CVV Code
- Customer Email Address

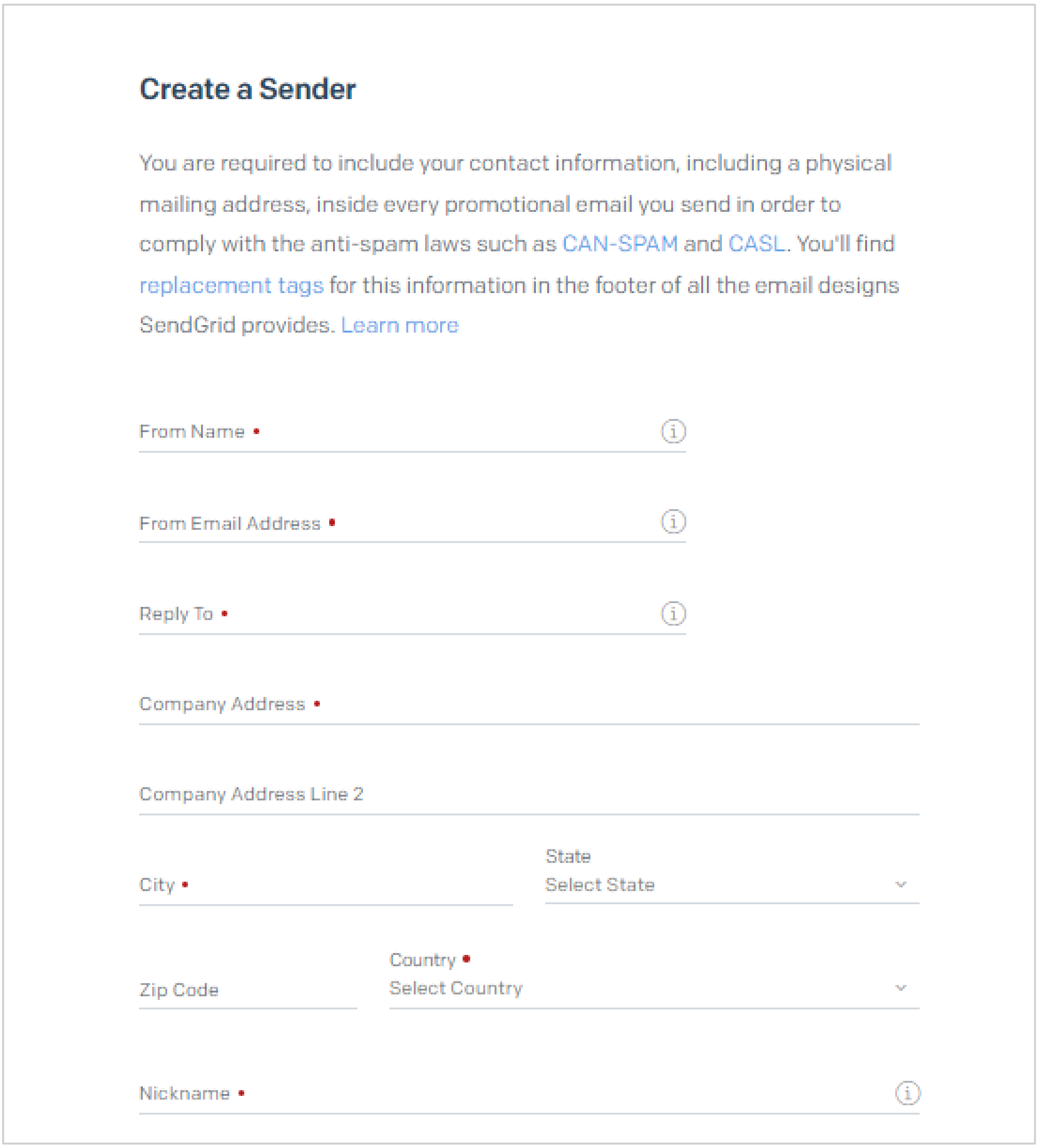
Information to be received from Stripe and captured on the system for all payment requests processed:

- Payment Intent ID
- Session ID

2.5. Email Service

The emails in the ByteXcom system will be sent out to satisfy business operations such as sending an email after an order has been placed or when a user wants to reset their password.

The email service we'll be using for this functionality is the Twilio SendGrid SMTP Service which uses a sender identity which includes information from on the server side. The information received by the API includes the senders name, email, company address, etc., as depicted in the image below.



Create a Sender

You are required to include your contact information, including a physical mailing address, inside every promotional email you send in order to comply with the anti-spam laws such as CAN-SPAM and CASL. You'll find replacement tags for this information in the footer of all the email designs SendGrid provides. Learn more

From Name •

From Email Address •

Reply To •

Company Address •

Company Address Line 2

City •

State
Select State

Zip Code

Country •
Select Country

Nickname •

Figure 1 – Twilio create sender page

This information is then used to send emails to users of the system using the company's email address. It also allows a separate email to be used for replies with the "Reply To" field in the form. This is used to control the traffic on the business' main email account so that no important emails are missed.

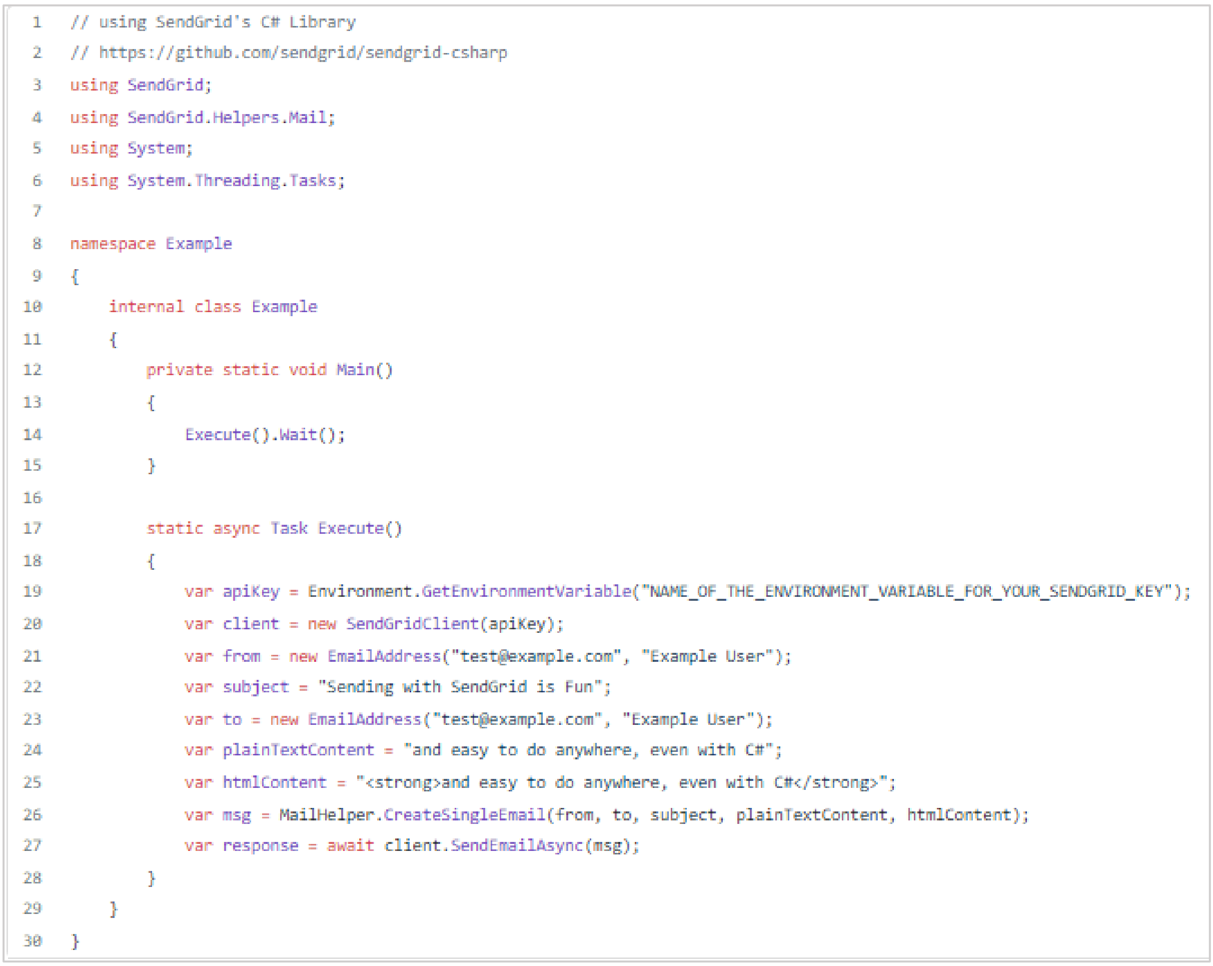
After setting up the sender identity, the WEP API integration process will be followed, this will be done through C#. Using the C# method requires the application used (Visual Studio) to have version .NET 4.5.2 in order to support the library used.

The following command will be used on the NuGet package manager console to download the package required:

Install-Package SendGrid

An API key is then created, this will allow the ByteXcom application to authenticate the SendGrid API and allow it to send emails.

This is the minimum code required by SendGrid to send an email. This is used to verify the connection between the ByteXcom system and the SendGrid API.



```csharp
// using SendGrid's C# Library
// https://github.com/sendgrid/sendgrid-csharp
using SendGrid;
using SendGrid.Helpers.Mail;
using System;
using System.Threading.Tasks;

namespace Example
{
    internal class Example
    {
        private static void Main()
        {
            Execute().Wait();
        }

        static async Task Execute()
        {
            var apiKey = Environment.GetEnvironmentVariable("NAME_OF_THE_ENVIRONMENT_VARIABLE_FOR_YOUR_SENDGRID_KEY");
            var client = new SendGridClient(apiKey);
            var from = new EmailAddress("test@example.com", "Example User");
            var subject = "Sending with SendGrid is Fun";
            var to = new EmailAddress("test@example.com", "Example User");
            var plainTextContent = "and easy to do anywhere, even with C#";
            var htmlContent = "<strong>and easy to do anywhere, even with C#</strong>";
            var msg = MailHelper.CreateSingleEmail(from, to, subject, plainTextContent, htmlContent);
            var response = await client.SendEmailAsync(msg);
        }
    }
}
```

Figure 2 - Twilio setup code example

The task used to trigger an email can then be configured to be sent to all of the consumers of the business, e.g., customers and resellers by referencing their entities in the Natuurlik database. The body of the email can also be configured to include the business' logo and make the email reflect the business' overall image. This task can be configured after the following activities/usecases have been completed on the system.

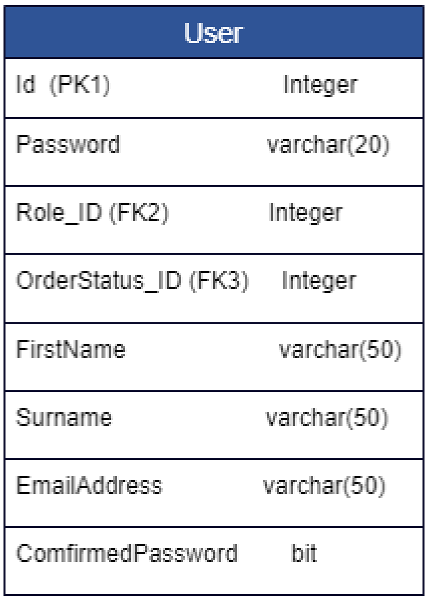
- When a customer/reseller clicks on the "forgot password link",
- When a customer/reseller places an order

These usecases will be used to trigger the task which executes the sending of an email to the user initiating the usecase.

The following usecases and entities with their attributes will be using the SendGrid API in order to send emails:

User Management Subsystem

1. Forgot Password
2. Register Customer
3. Update Email Address

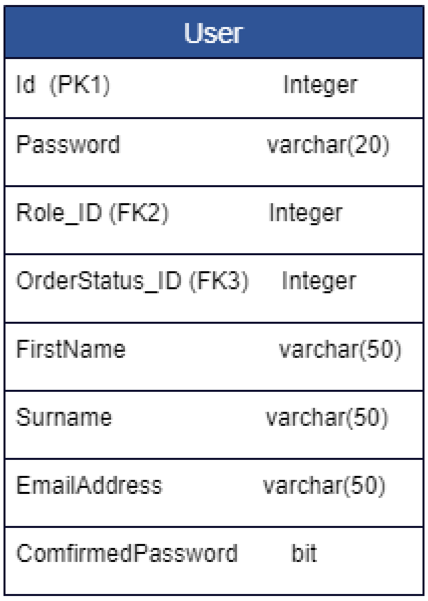


User	
Id (PK1)	Integer
Password	varchar(20)
Role_ID (FK2)	Integer
OrderStatus_ID (FK3)	Integer
FirstName	varchar(50)
Surname	varchar(50)
EmailAddress	varchar(50)
ComfirmedPassword	bit

Figure 3 - User Entity for User Management Subsystem

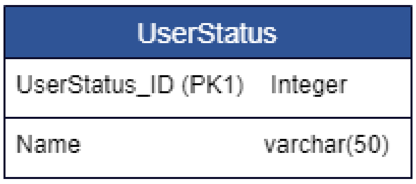
Administrative Subsystem

1. Register User



User	
Id (PK1)	Integer
Password	varchar(20)
Role_ID (FK2)	Integer
OrderStatus_ID (FK3)	Integer
FirstName	varchar(50)
Surname	varchar(50)
EmailAddress	varchar(50)
ComfirmedPassword	bit

Figure 4 - User Entity for Administrative Subsystem

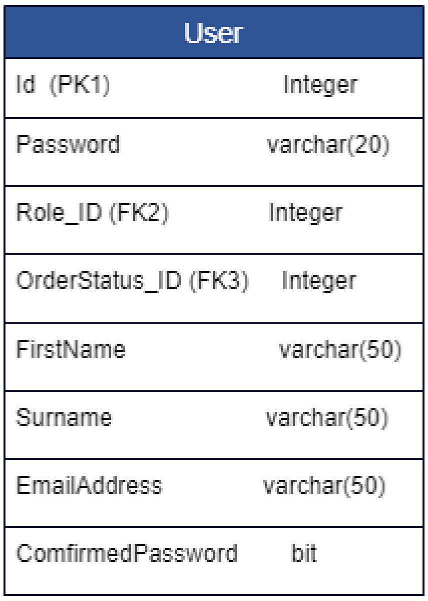


UserStatus	
UserStatus_ID (PK1)	Integer
Name	varchar(50)

Figure 5 - UserStatus Entity for Administrative Subsystem

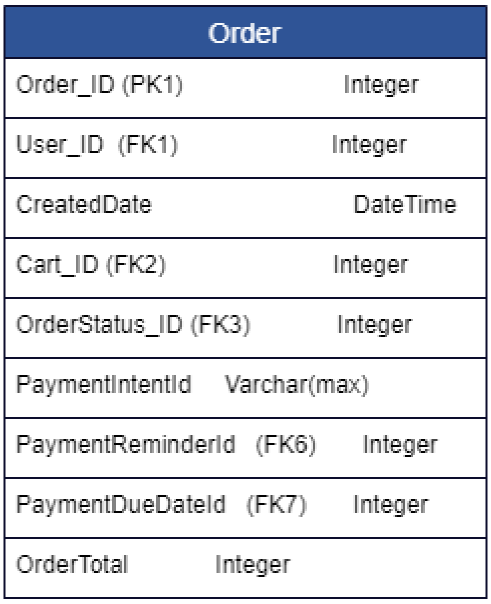
Reseller Subsystem

1. Send Payment Reminder
2. Capture Proof of Payment



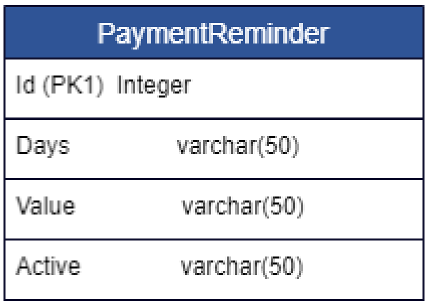
User	
Id (PK1)	Integer
Password	varchar(20)
Role_ID (FK2)	Integer
OrderStatus_ID (FK3)	Integer
FirstName	varchar(50)
Surname	varchar(50)
EmailAddress	varchar(50)
ComfirmedPassword	bit

Figure 6 - User Entity for Reseller Subsystem

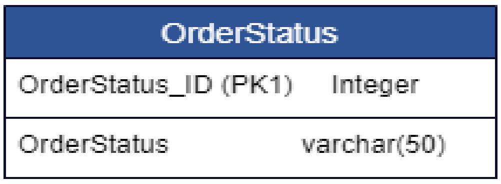


Order	
Order_ID (PK1)	Integer
User_ID (FK1)	Integer
CreatedDate	DateTime
Cart_ID (FK2)	Integer
OrderStatus_ID (FK3)	Integer
PaymentIntentId	Varchar(max)
PaymentReminderId (FK6)	Integer
PaymentDueDateId (FK7)	Integer
OrderTotal	Integer

Figure 7 - Order Entity for Reseller Subsystem



PaymentReminder	
Id (PK1) Integer	
Days	varchar(50)
Value	varchar(50)
Active	varchar(50)



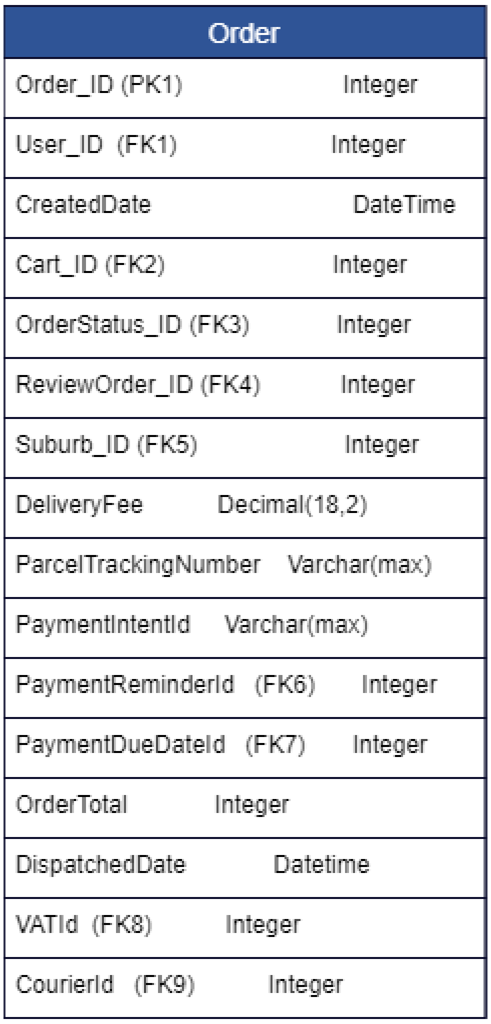
OrderStatus	
OrderStatus_ID (PK1)	Integer
OrderStatus	varchar(50)

Figure 8 - OrderStaus Entity for Reseller Subsystem

Figure 9 - PaymenReminder Entity for Reseller Subsystem

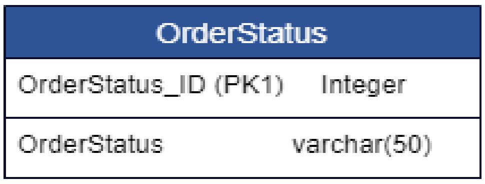
Order Subsystem

1. Make Payment
2. Cancel Placed Order
3. Cancel Reseller Order
4. Approve/Reject Reseller Order
5. Log Back Order
6. Confirm Order
7. Send Order Confirmation Reminder



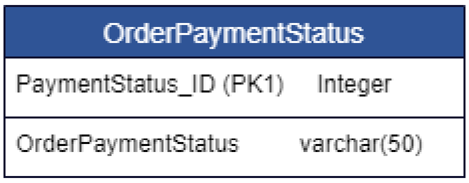
Order	
Order_ID (PK1)	Integer
User_ID (FK1)	Integer
CreatedDate	DateTime
Cart_ID (FK2)	Integer
OrderStatus_ID (FK3)	Integer
ReviewOrder_ID (FK4)	Integer
Suburb_ID (FK5)	Integer
DeliveryFee	Decimal(18,2)
ParcelTrackingNumber	Varchar(max)
PaymentIntentId	Varchar(max)
PaymentReminderId (FK6)	Integer
PaymentDueDateId (FK7)	Integer
OrderTotal	Integer
DispatchedDate	Datetime
VATId (FK8)	Integer
CourierId (FK9)	Integer

Figure 10 - Order Entity for Order Subsystem



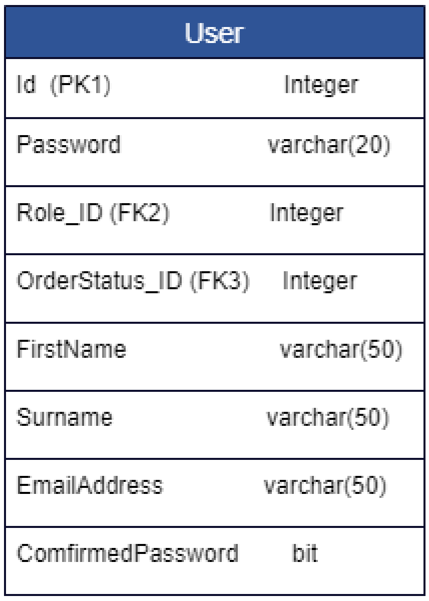
OrderStatus	
OrderStatus_ID (PK1)	Integer
OrderStatus	varchar(50)

Figure 11 - OrderStatus Entity for Order Subsystem

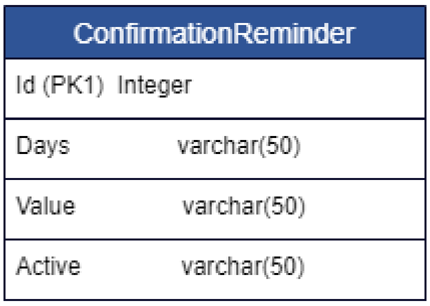


OrderPaymentStatus	
PaymentStatus_ID (PK1)	Integer
OrderPaymentStatus	varchar(50)

Figure 12 - OrderPaymentStatus Entity for Order Subsystem



User	
Id (PK1)	Integer
Password	varchar(20)
Role_ID (FK2)	Integer
OrderStatus_ID (FK3)	Integer
FirstName	varchar(50)
Surname	varchar(50)
EmailAddress	varchar(50)
ComfirmedPassword	bit



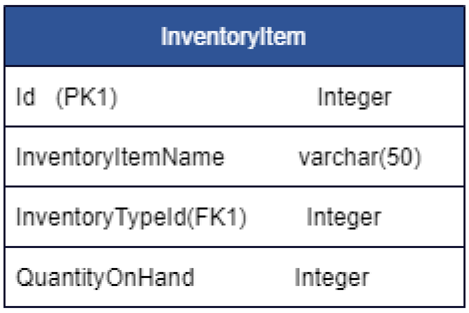
ConfirmationReminder	
Id (PK1)	Integer
Days	varchar(50)
Value	varchar(50)
Active	varchar(50)

Figure 13 - ConfirmationReminder Entity for Order Subsystem

Figure 14 - User Entity for Order Subsystem

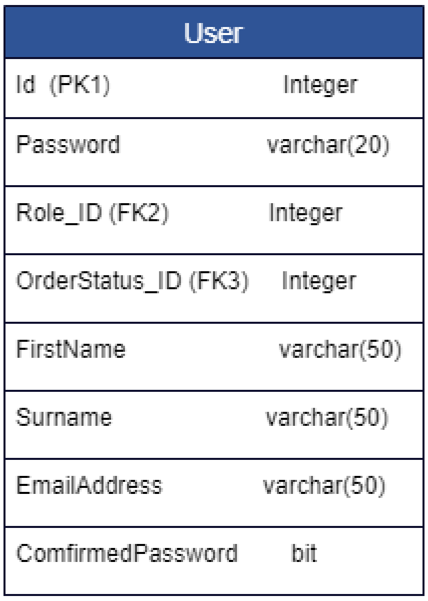
<u>Inventory Management Subsystem</u>

1. Send Low Inventory Alert
2. Send Inventory Out of Stock Alert



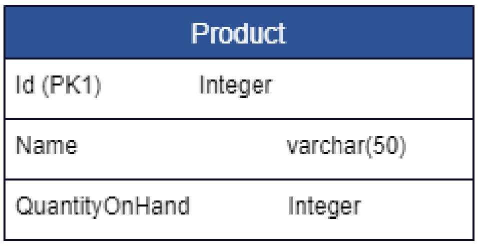
InventoryItem	
Id (PK1)	Integer
InventoryItemName	varchar(50)
InventoryTypeId(FK1)	Integer
QuantityOnHand	Integer

Figure 15 - InventoryItem Entity for Inventory Management Subsystem



User	
Id (PK1)	Integer
Password	varchar(20)
Role_ID (FK2)	Integer
OrderStatus_ID (FK3)	Integer
FirstName	varchar(50)
Surname	varchar(50)
EmailAddress	varchar(50)
ComfirmedPassword	bit

Figure 16 - User Entity for Inventory Management Subsystem

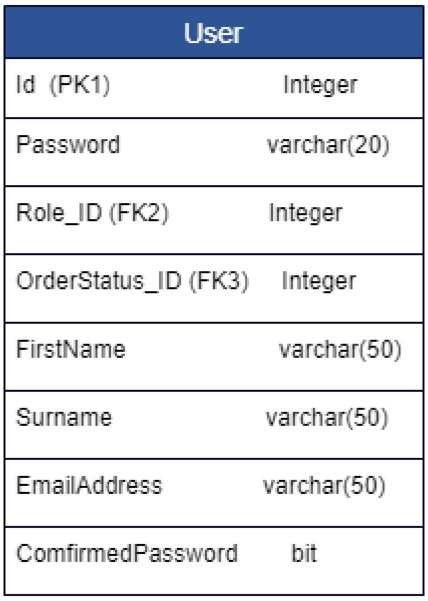


Product	
Id (PK1)	Integer
Name	varchar(50)
QuantityOnHand	Integer

Figure 17 - Product Entity for Inventory Management Subsystem

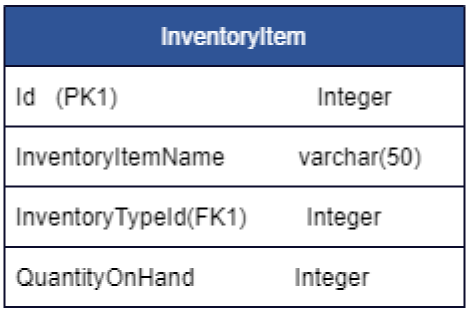
Supplier Subsystem

1. Send Supplier Order



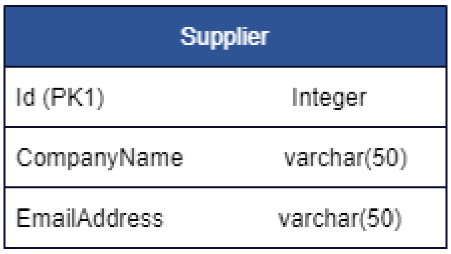
User	
Id (PK1)	Integer
Password	varchar(20)
Role_ID (FK2)	Integer
OrderStatus_ID (FK3)	Integer
FirstName	varchar(50)
Surname	varchar(50)
EmailAddress	varchar(50)
ComfirmedPassword	bit

Figure 19 - User Entity for Supplier Subsystem



InventoryItem	
Id (PK1)	Integer
InventoryItemName	varchar(50)
InventoryTypeId(FK1)	Integer
QuantityOnHand	Integer

Figure 18 - InventoryItem Entity for Supplier Subsystem

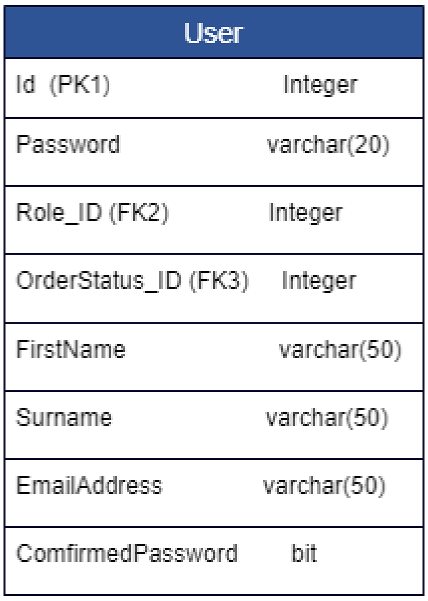


Supplier	
Id (PK1)	Integer
CompanyName	varchar(50)
EmailAddress	varchar(50)

Figure 20 - Supplier Entity for Supplier Subsystem

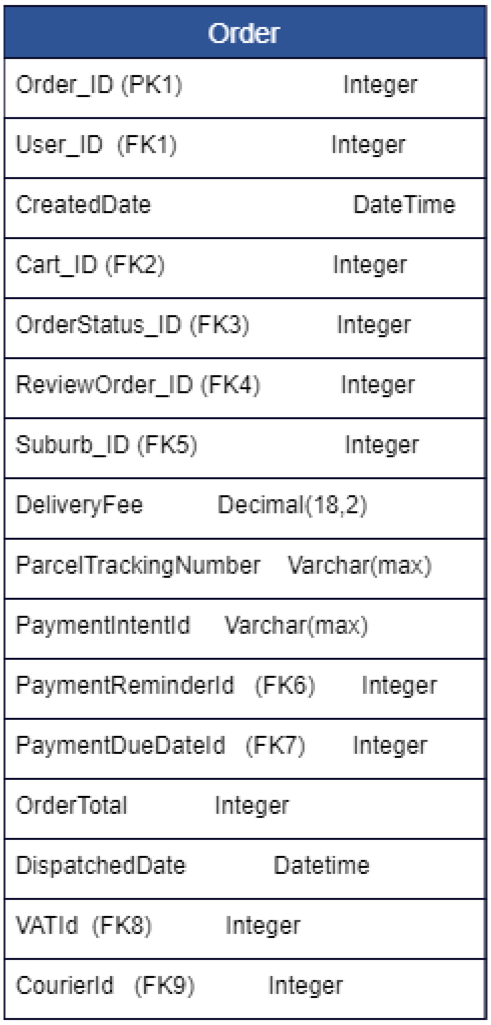
2.6. Twilio (SMS)

The ByteXcom system will be using Twilio to send their SMSs whenever an order is dispatched. This web service will be used for the Dispatch Order use case and will use the following entity from the following entities:



User	
Id (PK1)	Integer
Password	varchar(20)
Role_ID (FK2)	Integer
OrderStatus_ID (FK3)	Integer
FirstName	varchar(50)
Surname	varchar(50)
EmailAddress	varchar(50)
ComfirmedPassword	bit

Figure 21 - User Entity for Twilio SMS

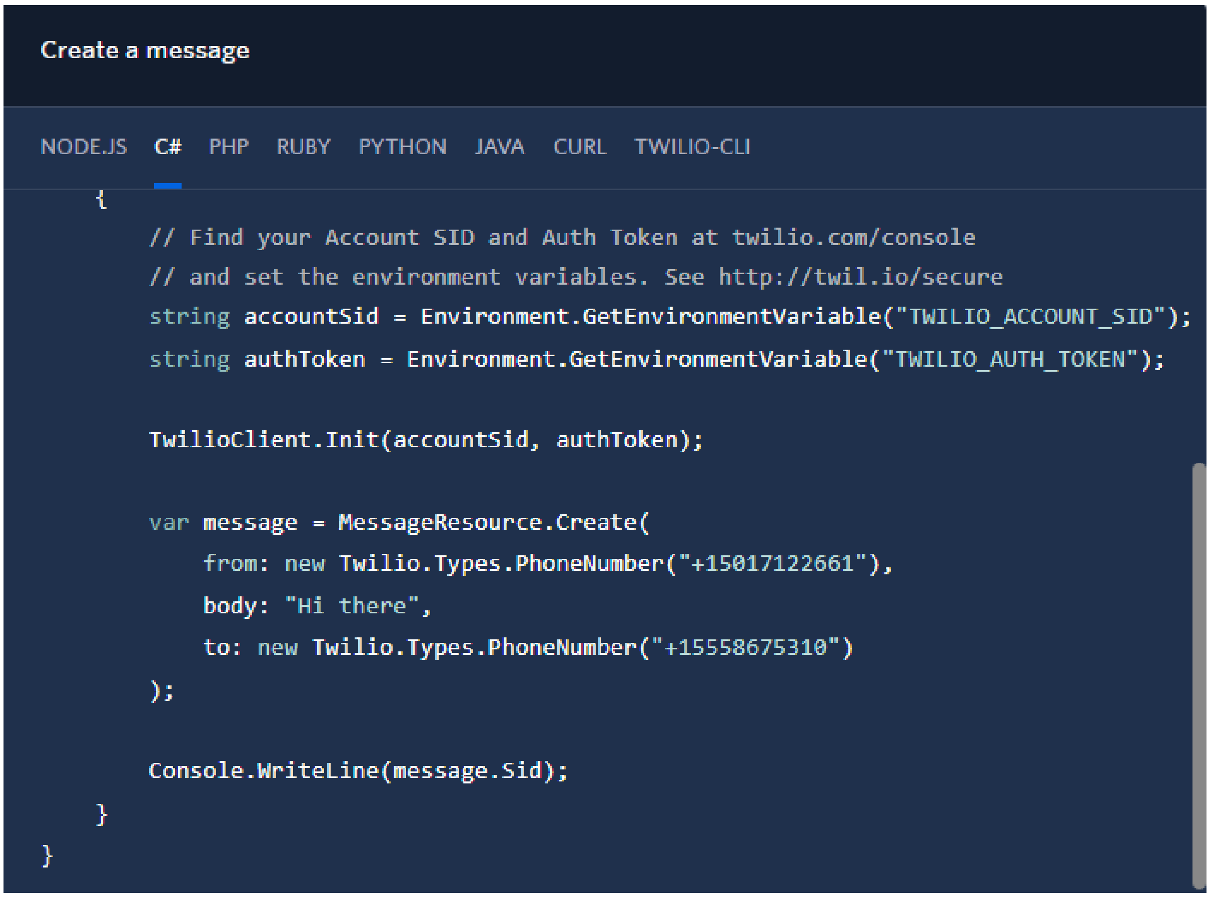


Order	
Order_ID (PK1)	Integer
User_ID (FK1)	Integer
CreatedDate	DateTime
Cart_ID (FK2)	Integer
OrderStatus_ID (FK3)	Integer
ReviewOrder_ID (FK4)	Integer
Suburb_ID (FK5)	Integer
DeliveryFee	Decimal(18,2)
ParcelTrackingNumber	Varchar(max)
PaymentIntentId	Varchar(max)
PaymentReminderId (FK6)	Integer
PaymentDueDateId (FK7)	Integer
OrderTotal	Integer
DispatchedDate	Datetime
VATId (FK8)	Integer
CourierId (FK9)	Integer

Figure 22 - Order Entity for Twilio SMS

Twilio is installed using the solution explorer on the project. The name of the package is called *Twilio.AspNet.Mvc*

This is an example for how we set up Twilio to be able to send SMS messages to Customers and Resellers after their order is dispatched.



```
    {
        // Find your Account SID and Auth Token at twilio.com/console
        // and set the environment variables. See http://twil.io/secure
        string accountSid = Environment.GetEnvironmentVariable("TWILIO_ACCOUNT_SID");
        string authToken = Environment.GetEnvironmentVariable("TWILIO_AUTH_TOKEN");

        TwilioClient.Init(accountSid, authToken);

        var message = MessageResource.Create(
            from: new Twilio.Types.PhoneNumber("+15017122661"),
            body: "Hi there",
            to: new Twilio.Types.PhoneNumber("+15558675310")
        );

        Console.WriteLine(message.Sid);
    }
}
```

Figure 23 - Twilio Set Up Example

2.7. Generating Reports and Dashboard using Power BI

The following will outline the details that are retrieved from the ByteXecom system's database in order to facilitate reporting using the Power BI Reporting tool. Once the system has been deployed, a direct connection between the system's database and Power BI Service will be established to enable real-time reporting on business-related information that is important to our client and that could potentially aid in better informed decision-making in running the business on a day-to-day basis.

The following reports will be generated using the Power BI Service integration:

1. Generate Orders By Status Report: A transactional report that breaks all orders placed on the system down by Status and Payment Status.

2. Generate Orders By Region Report: A transactional report that groups all orders where a payment has been received by Natuurlik by the Province and the City that is associated with each order for which a payment has been received. This will allow the owner to identify regions that drive higher revenues for their business.

3. View Dashboard Overview: A report that contains mission-critical data such as product returns, top-selling products, sales by geographical location and the couriers that are used most often. These numerous visuals should allow our client to get a high-level overview of how the business is performing.

The following data will be retrieved from the ByteXecom system's SQL Server database to generate these reports:

Order entity:

- OrderTotal
- OrderStatus
- OrderPaymentStatus
- IsResellerOrder (Boolean value to identify whether owner of an order is a reseller or a customer)

User entity:

- FirstName
- LastName

Product entity:

- ProductName

- ResellerPrice
- CustomerPrice

This information is used to identify top-selling products as well as what products are returned most often.

- **Country** entity:
  CountryName

- **Province** entity:
  ProvinceName

- **City** entity:
  CityName

- **Suburb** entity:
  SuburbName

These entities are used to break orders down by regions and to identify the areas where sales are highest.

Conclusion

In conclusion, ByteXpress compiled an extensive overview of the hardware and software that Natuurlik will require in order to operate the ByteXecom point of sales system.

3. Iteration Conclusion

In conclusion this iteration three document compiled by ByteXpress detailed the technical specification of our ByteXecom point of sales and business intelligence system for our client Natuurlik. Our team has worked tersely to accurately illustrate the technical components of our proposed system. We have provided the physical use case diagrams and physical entity relationship diagram for our entire as well as the preliminary descriptions of our design principles and required software and hardware. Thus, it can be concluded the abovementioned aspects of this technical specifications iteration will contribute to the success of on our iteration and the overall completion of our ByteXecom point of sales and business intelligence system.