ABSTRACT

We are in technology era, everything is trying to automatic or reduce most of human intervention. The e-commerce is something that very familiar with most of people from all in the world. The customer gains many advantages when using e-commerce form, such as reduce time, save energy, etc. Not only customer but also the business from small to big gain the advantages from this form, reduce price of renting place for the store, reduce price of hiring staff to supervising the store, save time on managing the products, bills, or anything else, reduce most of risk that data can be lost. We can see that it win-win situation that both side is gain more advantages than normal traditional commerce.

But, in current day, both customer and business are required more than that, such as the business want to automatic the process of packing products, print bills, reports, auto backup data, etc.; While the customer also need to know when the product available, when product come to them, what event currently occurs, etc. All of that events cannot do by just a people, if need, it will require many people to join in the process to make it available, but that will increase many risks and problems, while we want to reduce it. To solve that, the higher requirements for e-commerce project are placed. Business want to control everything in the process with most of it can run automatic without intervention of human, the software more secure to prevent the hijacking or any destructive actions, if the process still need to human such as delivery, it need to reduce the act of human to increase time of processing. All of that features, are the aims of every e-commerce projects and every developer.

PREFACE

As a junior developer, my aim is building the e-commerce system that can do all of that features. The reason I’m choosing this topic is its popular, and therefore, if I have any problem with the system, I still have a place to base on and solve the puzzle. The technology used in the project is ASP.Net Core 2 for back-end side and basic HTML5 CSS3 JS for front-end side, MSSQL server for database technology. The front-end technology is very common, so I just point the reason why I’m using the ASP.Net Core 2 for the back-end. First, I am confident on my knowledge of ASP.Net MVC. Second, my topic is very old in current day and every developer should know how to build the e-commerce system, because it very popular, example in everywhere, easy to do, easy to learn. So, I have to choose the new technology to balance with the old topic.

The ASP.Net Core framework is not too new, while it is release on Jul 8, 2014 in GitHub as a Microsoft open source project, the core is very familiar with the ASP.Net MVC, but the most valuable of this framework is it allow developer host the side cross platforms. It means developers and the business can reduce the money of renting the server to host their site or their service. But 4 years for a framework, is not a long time to it development. Developers still choose the ASP.Net MVC 5 over ASP.Net Core because the MVC 5 is more stable than current Core. Because it can run cross platform, it means some of feature run only in window must be removed, these missing features is something that developer cannot or not easy to replace. The extension and package support for ASP.Net Core is not enough as the MVC 5. Lack of those things and ASP.Net Core is still now not quite popular in development.

Those disadvantages of ASP.Net Core is some how can affect to my project but I still decide to choose it because I want to challenge my self as new framework and what I can learn from it. It can gain me more experience on handling problems, debug processes, solve or trying to figure the alternative way to solve problems.

1. **INTRODUCTION**
   1. **Overview**

- This project topic is about creating the e-commerce site that do basic tasks (add product, login, buy product, payment online) and expand to higher tasks (email notification). The site can mainly divide into two main parts: one for internal staff to manage their site, and one for customer to buy and browse for the products. This project can run online and need the network to send email or process payment. Other technologies were mentioned above is also include in project with specific role in operating the system.

* 1. **Goals**

- The goals of this project are building the new e-commerce system with ASP.Net Core technology and relate technologies. Also, when building the system, I will research on the ASP.Net Core, and other technologies. The research, the knowledge and problem I have met through the developing project will also discuss in detail in this document.

- The final goals are applying any knowledge to build the ASP.Net Core with basic e-commerce site requirements and some higher features.

* 1. **Scope of the project**

- The purpose of this project is creating the e-commerce system with basic functions and include some higher functions. The system should be interacted with the user and not occur any fatal error to lose the user data or information. Because it can interact with many customer and need extra information about their identity and private info, the security of the site is also in the requirement list.

- The system not only, provide a way to business manage their data, but also, it provides the effective way of browse data, store data and securing data. It also increases the user experience when customer use the site.

- The final scope is gain the experience on practical project, research new technologies and way of handling any error that can occur while system are using. In other word, this project is a challenge that can help me on increase the view of current e-commerce trends, and also new view in the development process.

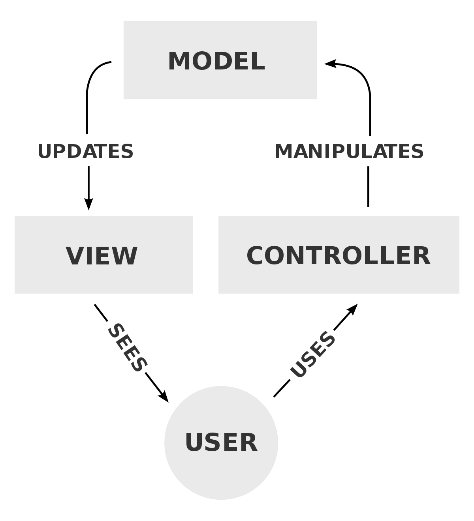
1. **LITERATURE REVIEW**
   1. **Introduction**

- The main purpose of the project is creating an e-commerce web application using ASP.Net Core framework as server side and HTML5-CSS3-JS for the client side. This part will introduce the technologies used in the project and the knowledge I have after researching. Along with the back-end and front-end technologies, I will also introduce the kind of database used in this project.

* Back-end technologies: belong with the ASP.Net core, I’ve also used the entity framework for performing the tasks with the server database such as: create, update, delete, select records, open database connection or transaction, etc., ASP.Net Core Identity framework is a famous framework for the membership system of ASP.Net website including (create membership, login, and user data security), LINQ will have the developer can customer the data and/or perform the task in database more easier, both of them is developing with C# programming language.
* Front-end technologies: design a website interface. I’m using the core technologies that almost every other website does, HTML5 (Hypertext Markup Language) for displaying the data to the user, CSS3 (Cascading Style Sheets 3) to format the interface for responsive, increase user experience and look well. JavaScript (JS) is used for interacting between client and website. It also helps on delivery client information to server side or validate the user input fields. There are other libraries which supporting for the front-end, the most using is: jQuery, Bootstrap, Font-Awesome, Bootstrap Notify, etc.
* Database: MSSQL Server (Microsoft Structured Query Language Server) database is using for this project, over other RDBMS (Relational database management systems) database, MSSQL Server developed by Microsoft and used widely in Asp.Net project. Both server side and database technologies are developing and supporting by Microsoft so I have no doubt about it performance and connection.
  1. **Web API and Model – View – Controller pattern**

- The ASP.Net Core is the combination of ASP.Net Web API project and ASP.Net MVC framework. The web API used in this project is for connecting to the database server and return the value or result as a JSON (JavaScript Object Notation). The ASP.Net Core MVC framework used for displaying the data, get user input data, and returning to server database by web API connection. The MVC pattern is using widely nowadays, it can reduce the duplicate code, reuse the code and make it more efficient. Like it name, it has three parts:

* Model: for storing the data, logic or any rule of the application. It also independent from the user interface.
* Controller: is the intermediate for the communication between model and view. It receives the input or data from client (through view) and pass it to the Model.
* View: is displaying the data for the user and also allow user input their data to the server.



Picture 1 - The process of MVC pattern

The when the user come to the website, the controller will send information to the model to do the appropriate task and then return the view to display the site for user. When user input data, one again, the input will go to controller, and it will validate in here, if something go wrong, it will redisplay the view for user and the message to correct it. If it passes the validation step, the input will come to model and execute the task. The MVC can reduce the dependency in each part of the project, instead, it allows developers divide the project into smaller parts, to manipulate and control the development process.

* 1. **Back-end technologies**
     1. **C# Programming Language**

- C# is a high-level programming language like Java, Python, PHP, etc. It becomes popular nowadays with the development of the Microsoft software and operating. The first thinking of any developer that want to develop a software run on Windows operation and Window for Mobile is C#. Within the same Microsoft technologies, the compatible of those technologies is ensure. The support and community on solving issue and future features is very huge. The C# syntax is also similarly with other languages, readable and understandable in sematic. Therefore, C# is suitable for any kind of developers from beginners to expects of other languages that want to learn C# to develop and website, windows app or mobile app that can run on Windows operating system.

* + 1. **ASP.Net Core**

- This is the first open source web framework that can run cross-platform. This is the next generation of ASP.Net MVC 5, and a big step of Microsoft that allow .Net developers expanding their development environment. By the .Net Core runtime, the app can run on any popular operating system such as OSX, Linux, and also Windows itself. Also, it not only running, but it also can be developing through those environment by Microsoft provide the Visual Studio 2017 that work in OSX and Visual Studio Code work cross-platform. Back to ASP.Net core MVC, its syntax is not too new, in other words, its syntax is very familiar with ASP.Net MVC 5. So, the developers like me that get acquainted with ASP.Net MVC 5 before, is also easy to know or learn about ASP.Net Core MVC. The big changes of ASP.Net MVC 5 to ASP.Net Core MVC is the supporting of Tag helper. Microsoft has described that Tag Helper is *“An HTML-friendly development experience”*, *“A way to make you more productive and able to produce more robust, reliable, and maintainable code using information only available on the server”*. I have a chance to research and use the Tag Helper through this project and I think this is awesome. Instead of old HTML Helper that look not familiar if developer come from front-end, the new Tag Helper provide a way that both back-end and front-end developers work together in the same find view razor. The combination of IntelliSense from Visual Studio and the Tag Helper, the need of working in razor view is reduce. Below are two pictures that Microsoft provides to compare the view of two syntax in razor view:

Picture 2 – Razor View with HTML Helper syntax

Picture 3 – Razor View with Tag Helper syntax

- We can easily see the difference and how easy it is to read the code within the front-end developers’ eyes.

* + 1. **ASP.NET Core Identity**

- The membership system that Microsoft build for ASP.Net Core web application. It allows developer to create a website that include the security system such as login, register, forgot password, or email confirm, etc. All data is store in tables created by Identity itself. It contains many helpful features but still provide a way that developer can customize it for suitable to each of the project type. The security of a website is a first aim of any developer, and special important with the e-commerce website where customer input their data information and their credit card/bank account etc. Therefore, I decide to use Identity framework to reduce the risk that can be happened. The framework will automatic create the salt and use one-way encryption to encrypt the sensitive data such as password. That can prevent the internal staff steal the customer info or the database being attacked. If anything, go wrong, the thing they stolen is just a encrypted string and almost impossible to decrypt.

* + 1. **Entity Framework Core**

- It is an object relational mapper (O/RM) that allow .Net developers work with the database, it reduce most of unnecessary code to create or query the task to database. It can support many database providers such as: MS SQL Server, MySQL, SQLite, and InMemory. The Entity Framework Core (EF Core) allow developers to auto-generate model class for the existing database or create database base on existing model. The EF Core also has migration features that enable developer to change field or table and update database. This feature is very useful when developers cannot ensure the final database, and while the development process, the requirements also changes, so the database must change to. I have use this features through developing this project and I glad that I used it, the requirements change overtime when deadline coming but features not ready, or some mistake in the first stage of creating database and need to change.

* + 1. **Language Integrated Query (LINQ)**

- It is the technology that help developer reduce time when perform filtering, ordering, and grouping operations on data sources with a minimum of code. It can work on combination entity framework core to get the customize data back with minimum of code without writing any SQL query. LINQ can also work in XML documents and streams, and .NET collections. LINQ has two syntax types: the lambda syntax (or method syntax) and the query (comprehension) syntax. Depend on each developer will choose their own syntax. Both syntax can run as the same performance and at least code. In the project, because of using web API to return JSON data from database to client side, the using of LINQ technology is very important. It allows me to customize the output data, get the kind of data I want, perform the specific condition and return it as a JSON object.

* 1. **Front-end technologies**
     1. **Hypertext Markup Language 5 (HTML 5)**

- The HTML5 is a mark-up language that used to create the webpage. Its syntax usually has a tag inside the open and close angle brackets (<body>, <div>, etc.). This mark-up language allows we displaying the media data to end-user. The data can be text, text-formatted, video, music, etc. It defines a structure for a webpage and entire website. The HTML5 is a new version of the HTML standard, it increases the speed of render page, reduce the resources need to render it, allow more media type to be render to the end-user. The HTML (HTML5) is a standard that almost every website need to have. It can represent metadata, allow the website has better SEO (Search engine optimization) result.

* + 1. **Cascading Style Sheets 3 (CSS 3)**

- If the HTML define the structure of a website, then the CSS will be the paint or the outside-color for that structure. Nowadays, the responsive trends is so popular, because the increasing of mobile user, the website must follow and has a friendly mobile design to allow the interaction with the customer more easily. If any web developer doesn’t know the work of responsive, they can use the Bootstrap CSS library which is free and open-source, there are other CSS libraries free and open-source and can match of any kind of difference projects. These libraries will help developer create responsive website, friendly user interface, good user experience for the website, with the minimum of knowledge about HTML or CSS. In my project, I’ve use many CSS libraries (Bootstrap, Font-Awesome, Bootstrap Select, etc.) which support for responsive, friendly interface, and any other task that relate to displaying the user interface to get higher chance of interaction with end-user.

* + 1. **JavaScript (JS)**

- If the HTML is a structure, the CSS is a paint for colorful that structure, then the JS is the way that customer can interact with that colorful structure or vice versa. The JS is a programming language running on a client browser, it executes that code or command help user interact with the website. For example, the user input fields will be validate in the client side to help user input correct data before sending it to server. The JS can also helpful on the search or auto complete input. Depend on each of the JS library, it will have difference usage and result. JS can also allow developer store some information about that customer in the client browser and reuse it if necessary. One of the most famous JS library is jQuery. It is JS based but with many useful methods to allow developer can reduce the code line and save time with the same task. In the project, I have used many other type of JS libraries (jQuery, bootstrap-notify, js-select, js-datatable, etc.) to perform a simple task for each of page.

1. **REQUIREMENT ANALYSIS**
   1. **Functional Requirements**

- The project is about building the e-commerce web application and also management system for the admin. The system has four types of user: Administrator, Product Manager, Customer and Anonymous Customer. Each group has a specific role. The Administrator, Product Manager and Customer need to be login to use their function in the website. While the Anonymous Customer is the client with non-authenticated and allowed to add product into cart and checkout. The Admin is the only person can add the staff (Product Manager). The customer can login through the email registered, or by their Facebook account. The differences between Customer and Anonymous Customer is the Customer can add their address and use it for next time checkout, but the anonymous customer will need to provide their address every time they checkout. Also, the authenticated customer can access to any other new features such as add to wish list or review the product.

- The product manager will have the ability to add the category, brand and product to bring new product for their customer. Also, the manager can perform some relate tasks for product, brand, category such as search, update information, etc. In this project scope, I don’t allow any user can be deleted or has a right to delete any other things. If necessary, the manager is the only person can hide the product, brand, category.

- The product in this system is added by the product manager and it must belong to brand and category. Product has two types of quantity is the current quantity and maximum quantity. Current quantity provides the current quantity of the product can have in store. Maximum quantity provides the maximum quantity of a product that can only be placed in the cart.

- This version of project has all basic features for a simple e-commerce website. The limitation of time made me decide to move remain features into next update version.

|  |  |  |
| --- | --- | --- |
| No. | Roles | Functions |
| 1 | Administrator | Login to the admin site  Staff Management   * See list of current Staffs * Reset staff password * Add new Staff |
| 2 | Product Manager | Login to the admin site  Brand Management   * See list of current Brands * Search Brands * Add new Brand * Update current information of Brand   Category Management   * See list of current Categories * Search Categories * Add new Category * Update current information of Category   Promotion Management   * See list of current Promotions * Search Promotions * Add new Promotion * Update current information of Promotion   Product Management   * See list of current Products * Search Products * Add new Product * Update current information of Product   Order Management   * See list of current Orders * Search Orders |
| 3 | Customer | Account Management   * Add/Update current user information * See list of orders has been placed * Add address for checkout * Change password * Forgot Password   Use promotion code for checkout  Add products to cart and checkout  Receive the notify of product available |
| 4 | Anonymous Customer | Register by Email or Facebook account  Login by Email or Facebook account  Use promotion code for checkout  Add products to cart and checkout  Receive the notify of product available |

* 1. **Non-functional Requirements**

- Beside the basic requirement of an e-commerce website, any other requirement is also important.

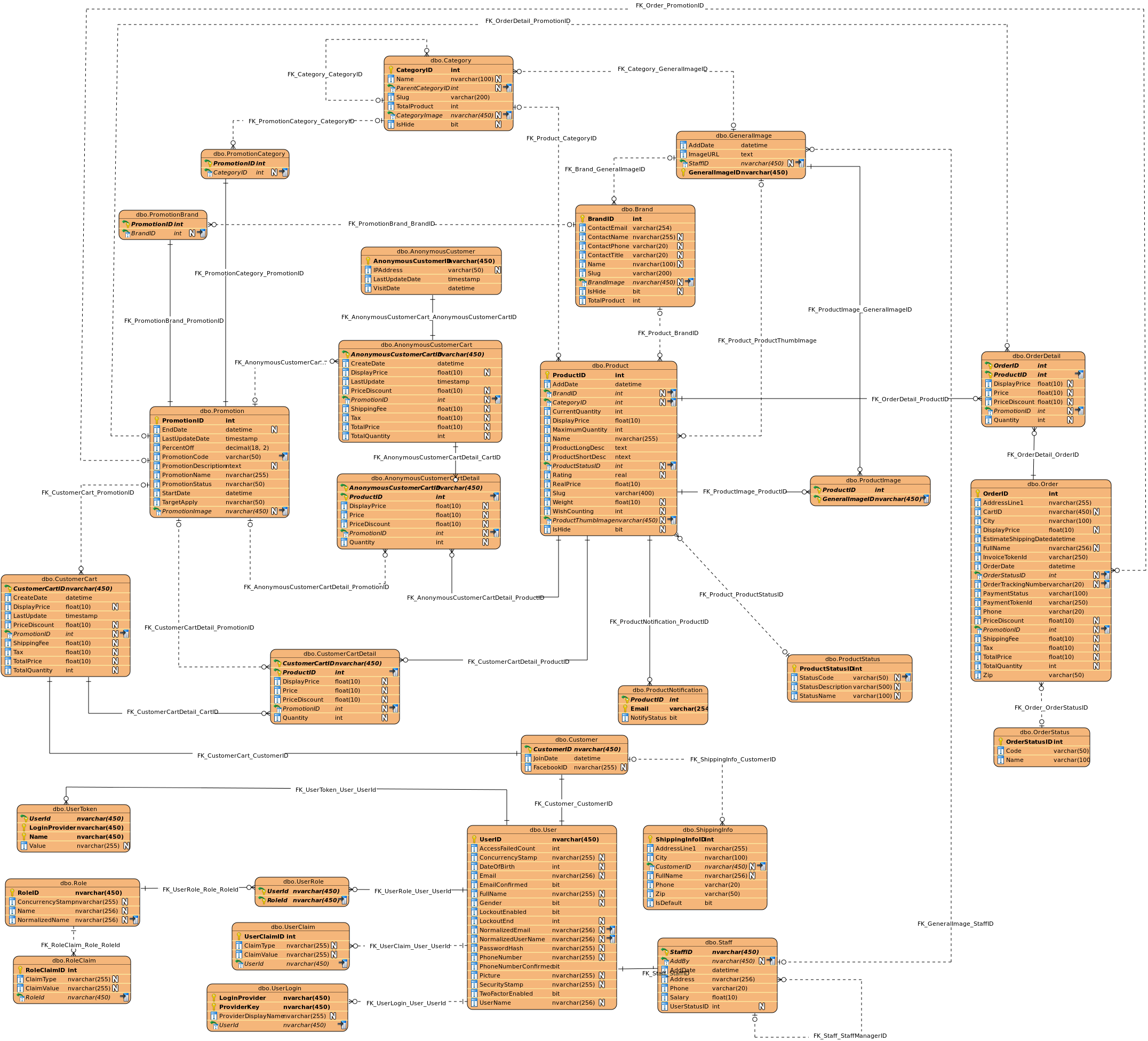
* Security: the security must be ensuring to keep the user information from stolen or prevent the hacker can go inside the database to destructive info.
* Interface: the interface has to friendly with customer and made them feel comfortable, not too much info, but simple to avoid confusing of customer. The interface also should be run smoothie in the mobile browser.
* User guide and installation guide need to be produced after this document to allow the business can run the website and use it without any trouble.
  1. **Development Environment**

- This system is developed using the ASP.Net Core framework, I choose Visual Studio 2017 is the IDE to develop the .Net Core project because its compatibility.

- For the database development and storage, I choose MSSQL Server 2014, even I’m not using this too much to developing the project.

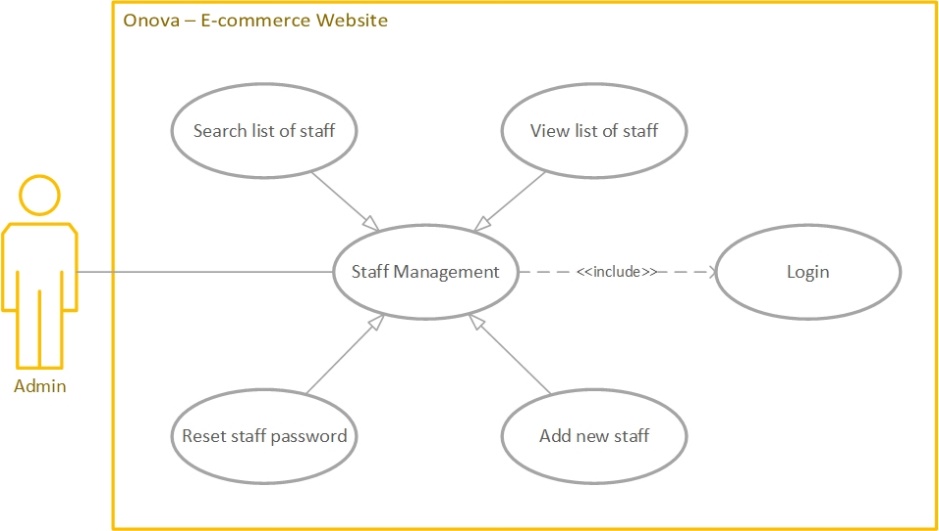
- Operating system currently, I’ve used to develop the project is Windows 10.

1. **DESIGN**
   1. **Entity Relationship Diagram (ERD)**



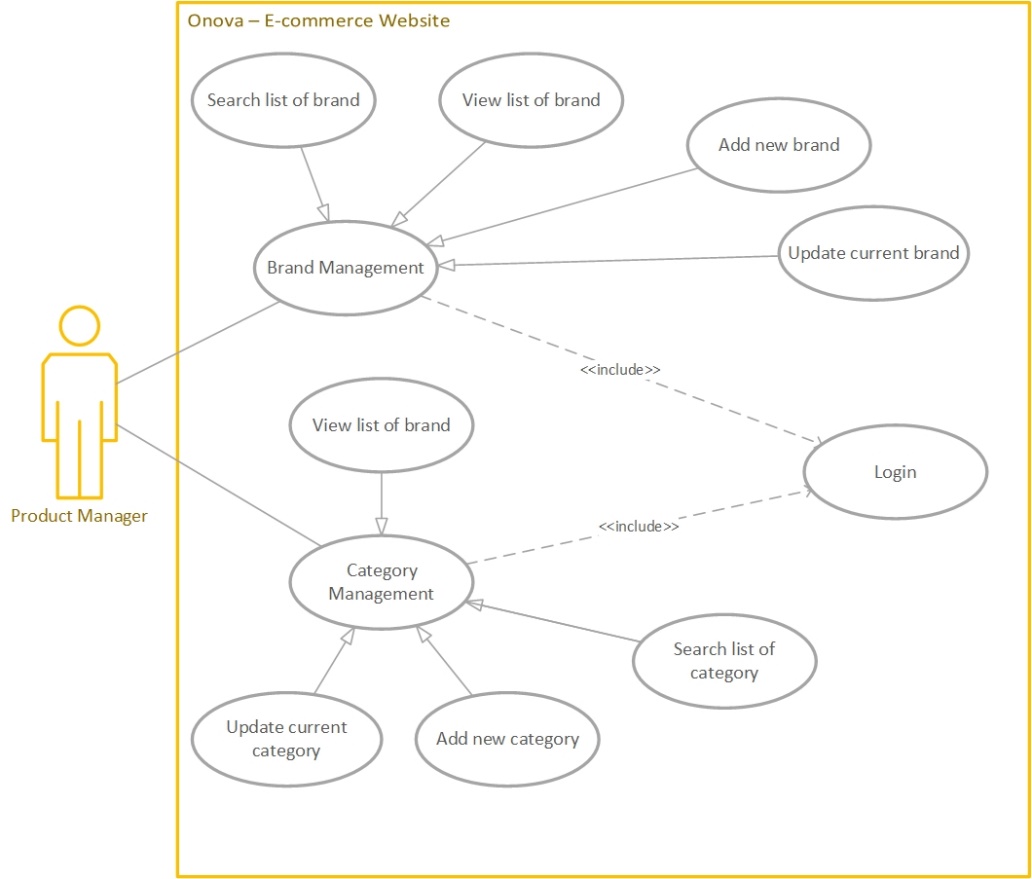
Picture 4 – Entity Relationship Diagram

* 1. **Use Case Diagram (UCD)**
     1. **Administrator use case diagram**

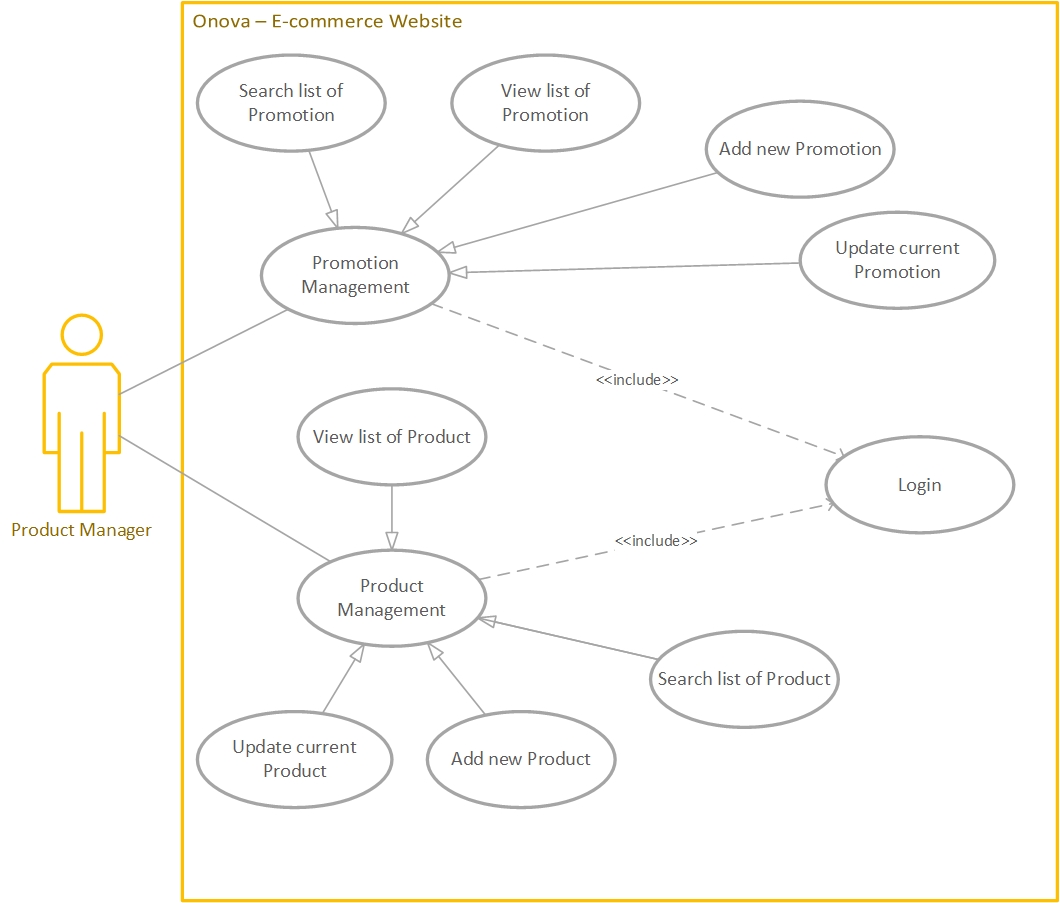


Picture 5 – Admin Use Case Diagram

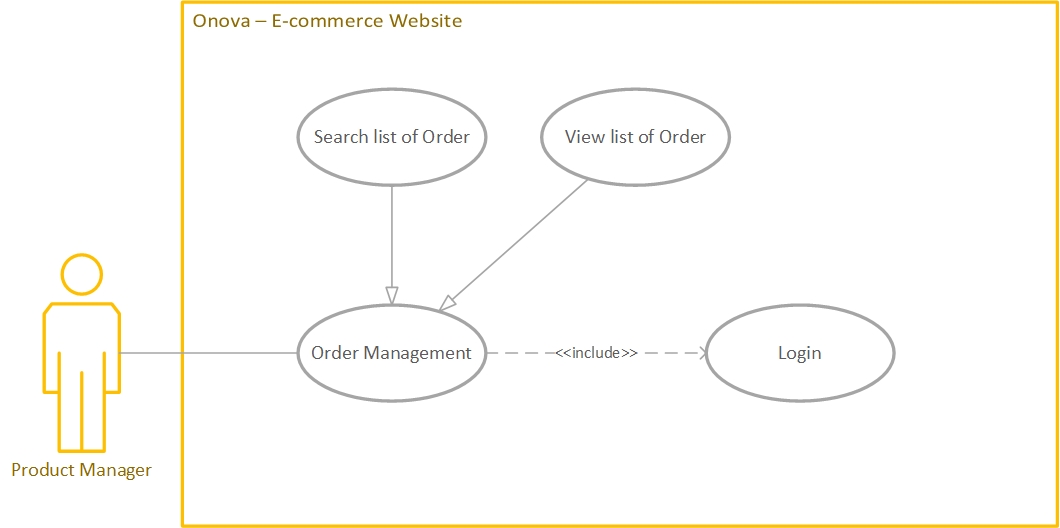
* + 1. **Product Manager use case diagram**



Picture 6.1 – Product Manager Use Case Diagram

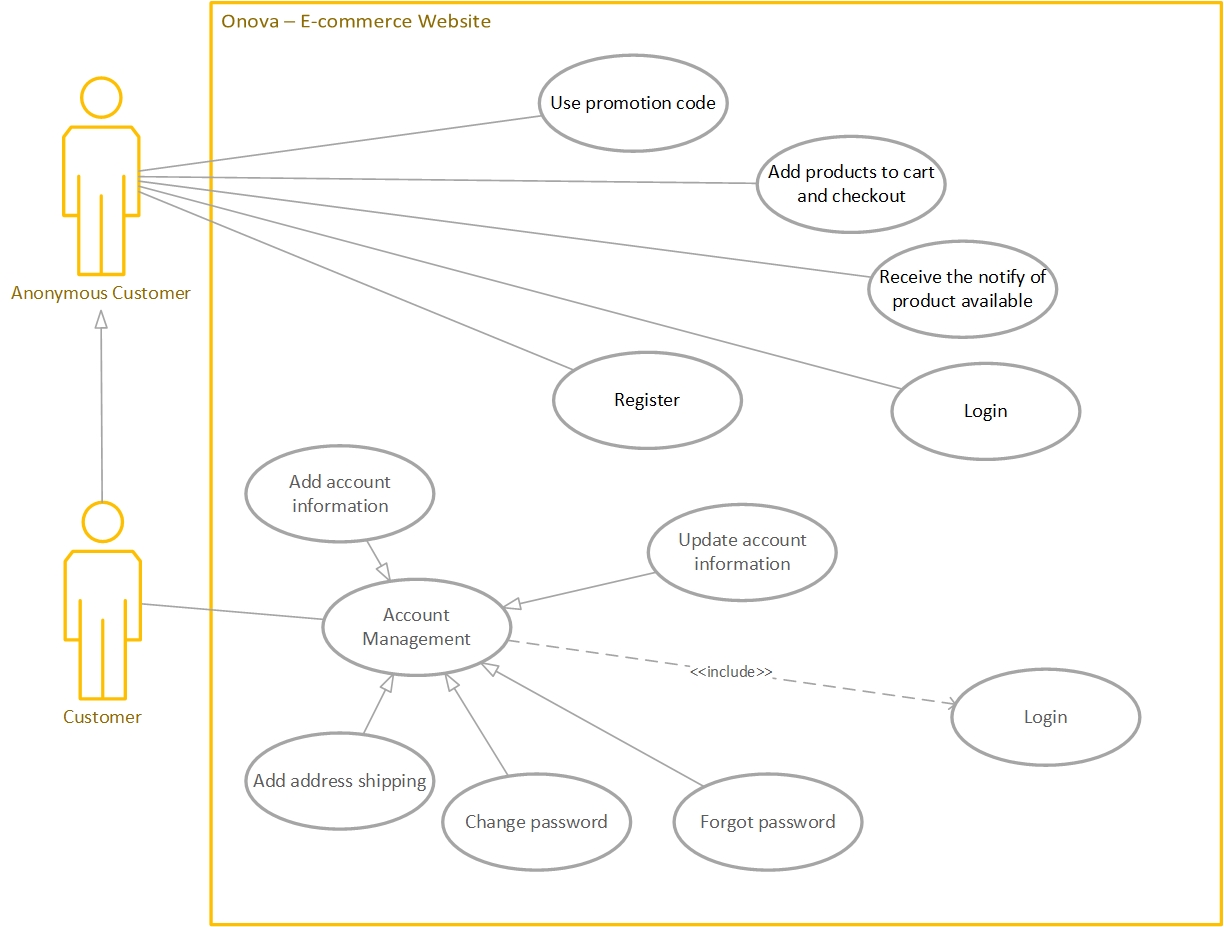


Picture 6.2 – Product Manager Use Case Diagram



Picture 6.3 – Product Manager Use Case Diagram

* + 1. **Customer and Anonymous Customer use case diagram**



Picture 7 – Customer and Anonymous Customer Use Case Diagram

* 1. **Use Case Explanation** 
     1. **Use case – Log In**

|  |  |
| --- | --- |
| **Title** | Log in use case |
| **Description** | Log in to the system |
| **Actors** | Administrator, Product Manager, Customer, Anonymous Customer |
| **Initial status & Preconditions** | User must have the account for the system |
| **Basic flow** | |
| 1. User input the email and password 2. The server will validation the input 3. The server will redirect user after login successful | |
| **Post condition** | |
| * No | |
| **Alternative flow** | |
| * No | |

|  |  |
| --- | --- |
| **Title** | Register use case |
| **Description** | Register the account of the system |
| **Actors** | Anonymous Customer |
| **Initial status & Preconditions** | * No |
| **Basic flow** | |
| 1. User input the information 2. The server will validation the input 3. The server will redirect user after register successful | |
| **Post condition** | |
| * No | |
| **Alternative flow** | |
| 1. Click on the login with Facebook button 2. User input the Facebook account 3. The server will redirect user after register successful | |

* + 1. **Use case – Register**

|  |  |
| --- | --- |
| **Title** | Brand management use case |
| **Description** | Manage brand of the store |
| **Actors** | Product Manager |
| **Initial status & Preconditions** | User must login to the system with the Product Manager role |
| **Basic flow** | |
| 1. Log in 2. View the list of current brands 3. Search list of brands    1. Input value to search field    2. Enter 4. Add new brand    1. Click add new brand link    2. Enter new brand info into the fields    3. Click Add new brand button 5. Update brand information    1. Click on the specific brand action menu in the last right column    2. Choose Edit/Update    3. Enter any value that need to update    4. Click update button | |
| **Post condition** | |
| * No | |
| **Alternative flow** | |
| * No | |

* + 1. **Use case – Brand Management**

|  |  |
| --- | --- |
| **Title** | Category management use case |
| **Description** | Manage category of the store |
| **Actors** | Product Manager |
| **Initial status & Preconditions** | User must login to the system with the Product Manager role |
| **Basic flow** | |
| 1. Log in 2. View the list of current categories 3. Search list of categories    1. Input value to search field    2. Enter 4. Add new category    1. Click add new category link    2. Enter new category info into the fields    3. Click Add new category button 5. Update category information    1. Click on the specific category action menu in the last right column    2. Choose Edit/Update    3. Enter any value that need to update    4. Click update button | |
| **Post condition** | |
| * No | |
| **Alternative flow** | |
| * No | |

* + 1. **Use case – Category Management**

|  |  |
| --- | --- |
| **Title** | Product management use case |
| **Description** | Manage Product of the store |
| **Actors** | Product Manager |
| **Initial status & Preconditions** | User must login to the system with the Product Manager role |
| **Basic flow** | |
| 1. Log in 2. View the list of current products 3. Search list of products    1. Input value to search field    2. Enter 4. Add new Product    1. Click add new product link    2. Enter new product info into the fields    3. Click Add new product button 5. Update Product information    1. Click on the specific product action menu in the last right column    2. Choose Edit/Update    3. Enter any value that need to update    4. Click update button | |
| **Post condition** | |
| * No | |
| **Alternative flow** | |
| * No | |

* + 1. **Use case – Product Management**

|  |  |
| --- | --- |
| **Title** | Promotion management use case |
| **Description** | Manage Promotion of the store |
| **Actors** | Product Manager |
| **Initial status & Preconditions** | User must login to the system with the Product Manager role |
| **Basic flow** | |
| 1. Log in 2. View the list of current promotions 3. Search list of promotions    1. Input value to search field    2. Enter 4. Add new Promotion    1. Click add new promotion link    2. Enter new promotion info into the fields    3. Click Add new promotion button 5. Update Promotion information    1. Click on the specific promotion action menu in the last right column    2. Choose Edit/Update    3. Enter any value that need to update    4. Click update button | |
| **Post condition** | |
| * No | |
| **Alternative flow** | |
| * No | |

* + 1. **Use case – Promotion Management**

|  |  |
| --- | --- |
| **Title** | Order management use case |
| **Description** | Manage Order of the store |
| **Actors** | Product Manager |
| **Initial status & Preconditions** | User must login to the system with the Product Manager role |
| **Basic flow** | |
| 1. Log in 2. View the list of current orders 3. Search list of orders    1. Input value to search field    2. Enter 4. Add new Orders    1. Click add new Order link    2. Enter new Order info into the fields    3. Click Add new order button 5. Update Order information    1. Click on the specific order action menu in the last right column    2. Choose Edit/Update    3. Enter any value that need to update    4. Click update button | |
| **Post condition** | |
| * No | |
| **Alternative flow** | |
| * No | |

* + 1. **Use case – Order Management**

|  |  |
| --- | --- |
| **Title** | Use promotion code use case |
| **Description** | Apply promotion code to cart to get discount |
| **Actors** | Customer, Anonymous Customer |
| **Initial status & Preconditions** | Have promotion code |
| **Basic flow** | |
| 1. Go to checkout page 2. Click on Apply promotion code    1. Input code to the field    2. Click Apply | |
| **Post condition** | |
| * No | |
| **Alternative flow** | |
| * No | |

* + 1. **Use case – Use Promotion Code**
    2. **Use case – Add Product to cart**

|  |  |
| --- | --- |
| **Title** | Add product to cart use case |
| **Description** | Add product into customer cart |
| **Actors** | Customer, Anonymous Customer |
| **Initial status & Preconditions** | * No |
| **Basic flow** | |
| 1. Go to home page/product detail page 2. Click add to cart symbol or button | |
| **Post condition** | |
| * No | |
| **Alternative flow** | |
| * No | |

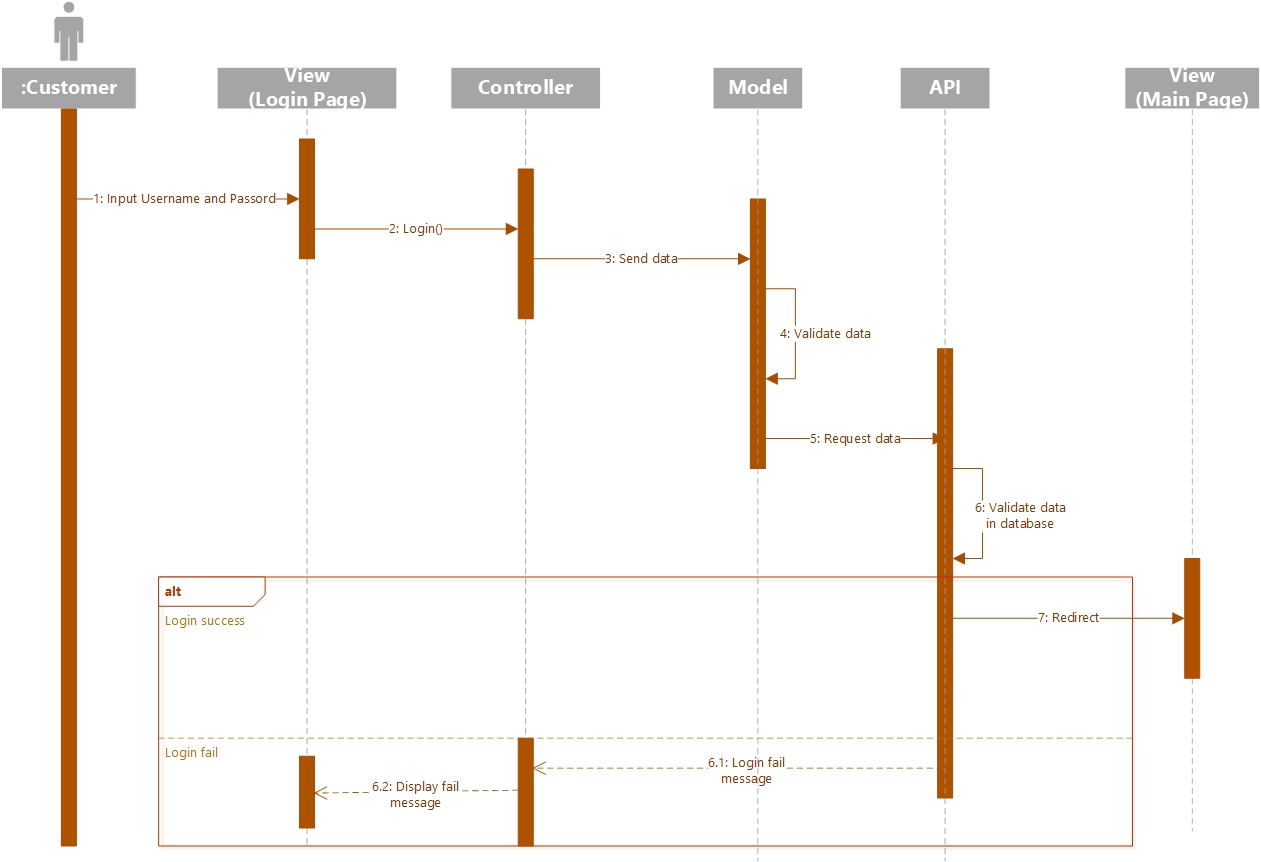
* + 1. **Use case – Checkout to cart**

|  |  |
| --- | --- |
| **Title** | Checkout cart |
| **Description** | Checkout and payment the cart |
| **Actors** | Customer, Anonymous Customer |
| **Initial status & Preconditions** | Cart have at least one product  Have a credit card (VISA) |
| **Basic flow** | |
| 1. Go to Checkout page 2. Click checkout button after review the cart 3. Input the fields if necessary 4. Input the card info to process payment 5. Check email to get order placed confirmation | |
| **Post condition** | |
| * No | |
| **Alternative flow** | |
| * No | |

* + 1. **Use case – Receive Notification of Product**

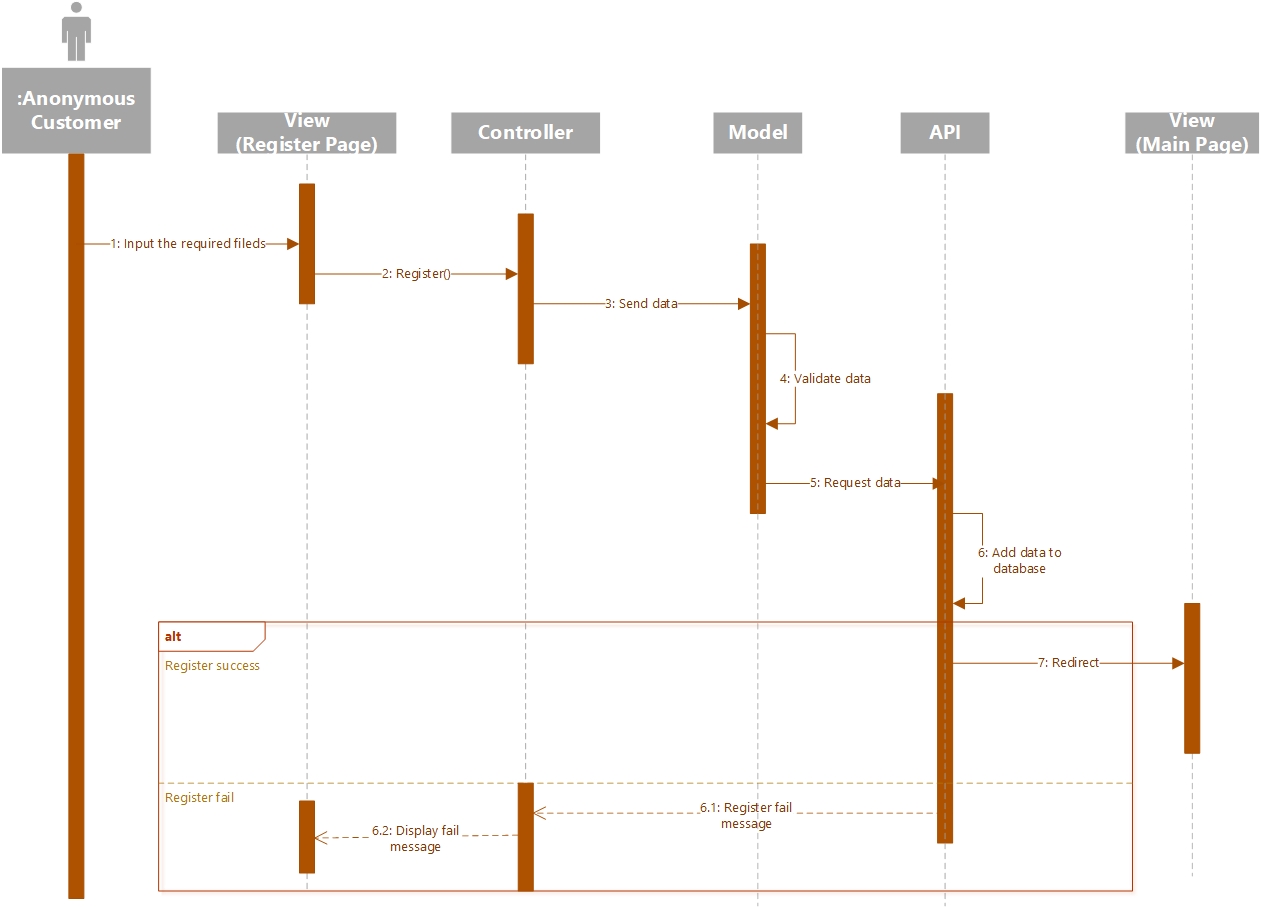
|  |  |
| --- | --- |
| **Title** | Receive notification of the product |
| **Description** | Get email of the available product |
| **Actors** | Customer, Anonymous Customer |
| **Initial status & Preconditions** | Product in state Sold Out or Stop Selling |
| **Basic flow** | |
| 1. Go to the product detail page 2. Input the email to the input field 3. Click get notification button | |
| **Post condition** | |
| * No | |
| **Alternative flow** | |
| * No | |

* 1. **Sequence Diagram**
     1. **Log in Sequence Diagram**



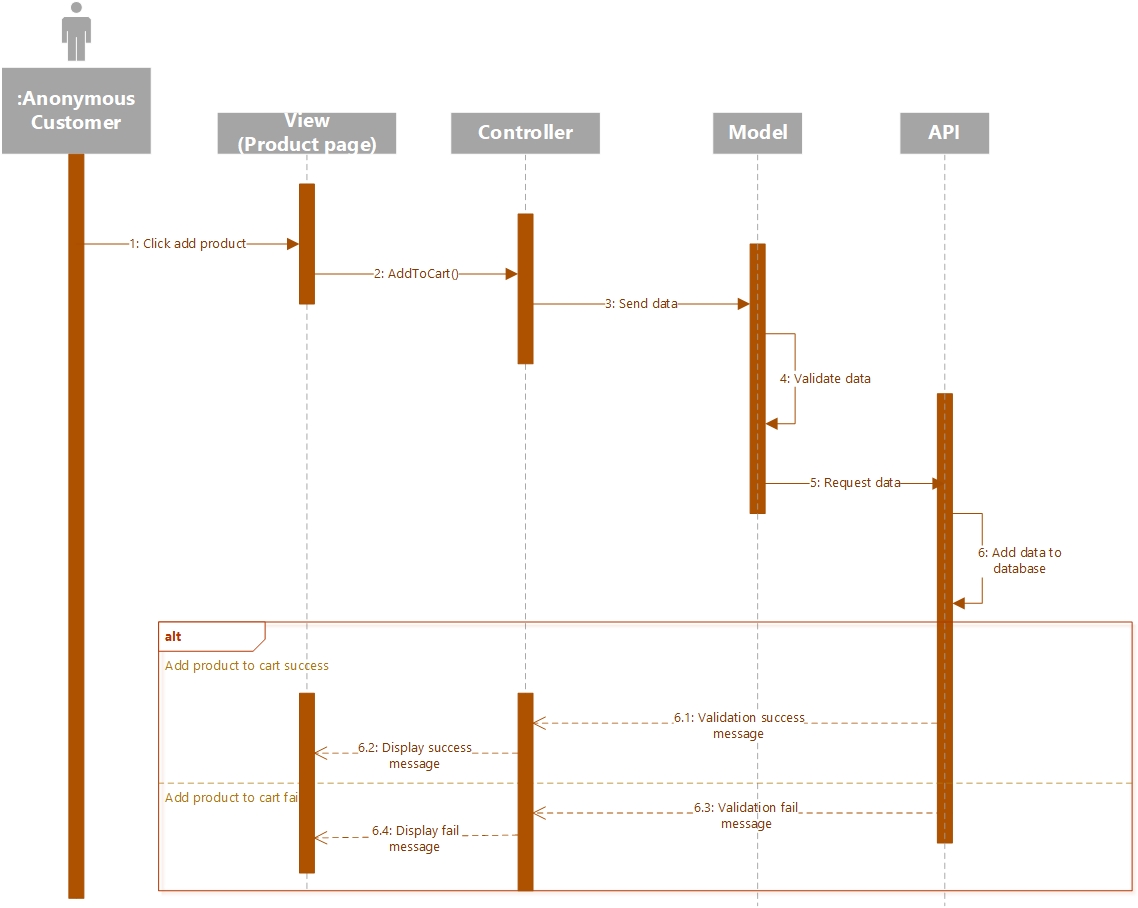
Picture 8 – Sequence Diagram Log In

* + 1. **Register Sequence Diagram**



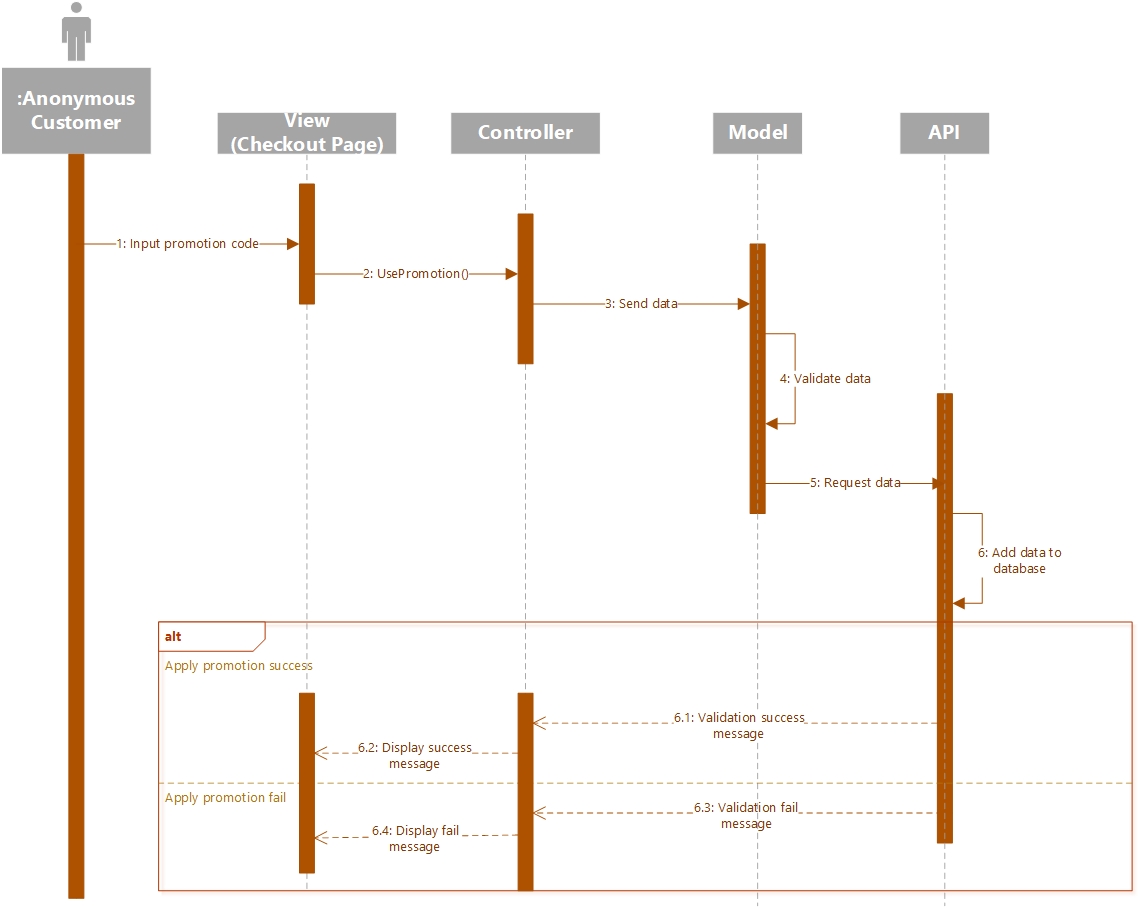
Picture 9 – Sequence Diagram Register

* + 1. **Add Product to Cart Sequence Diagram**



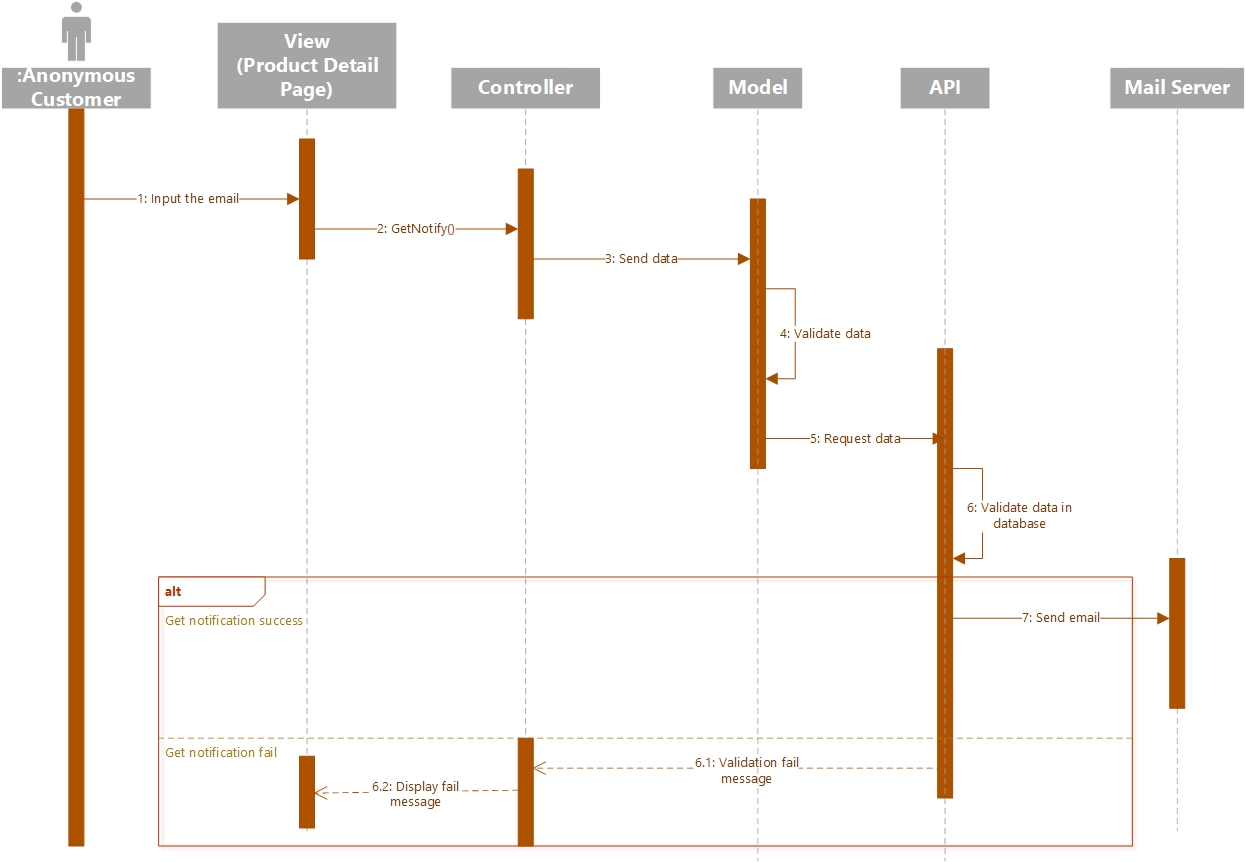
Picture 10 – Sequence Diagram Add product to Cart

* + 1. **Apply Promotion Code Sequence Diagram**



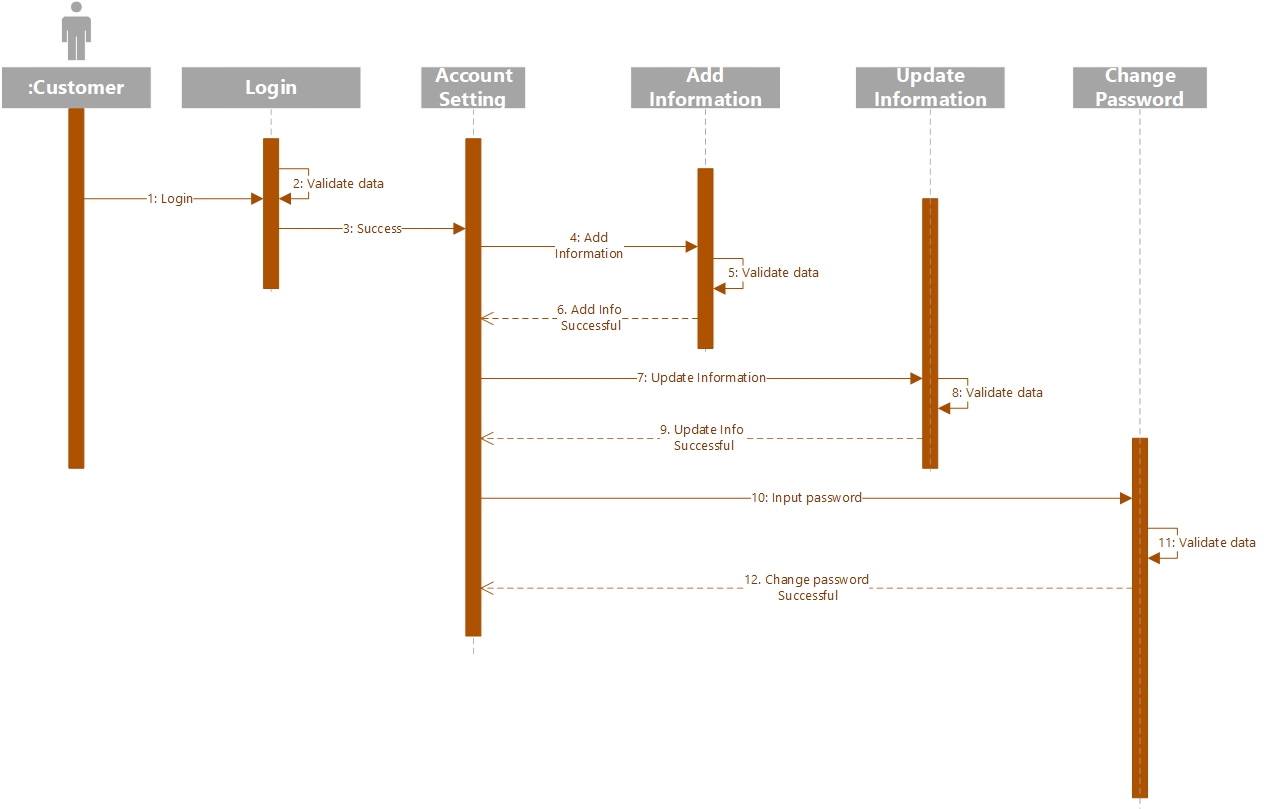
Picture 11 – Sequence Diagram Apply promotion code

* + 1. **Get Product Notification Sequence Diagram**



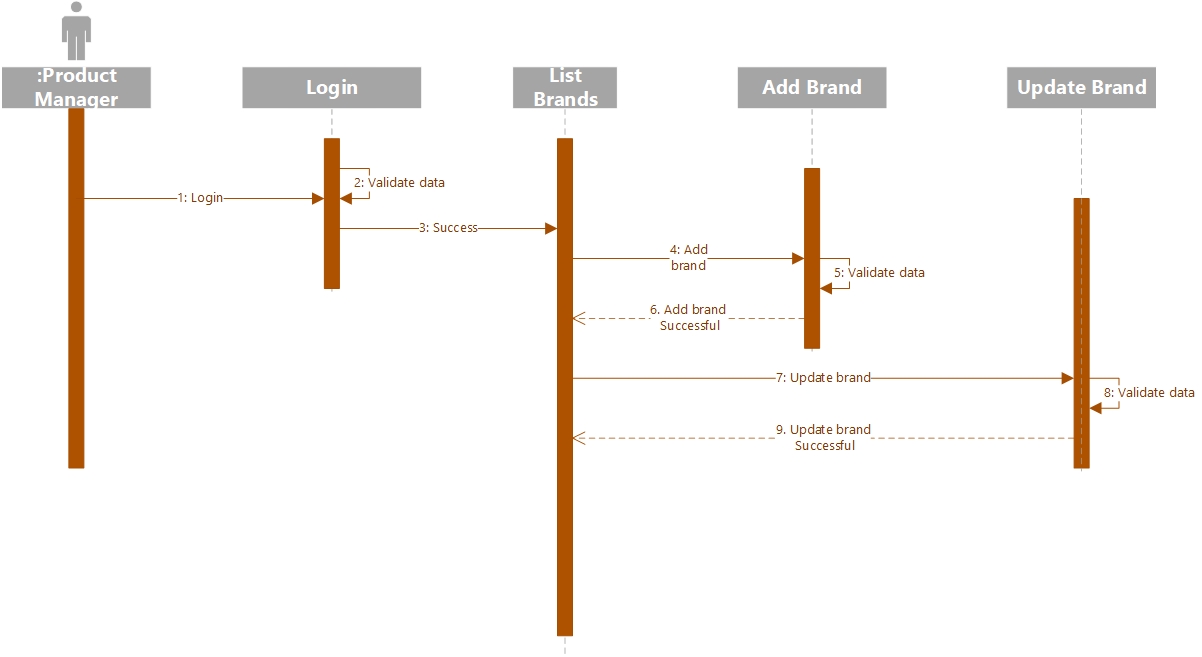
Picture 12 – Sequence Diagram Apply promotion code

* + 1. **Account Management Sequence Diagram**



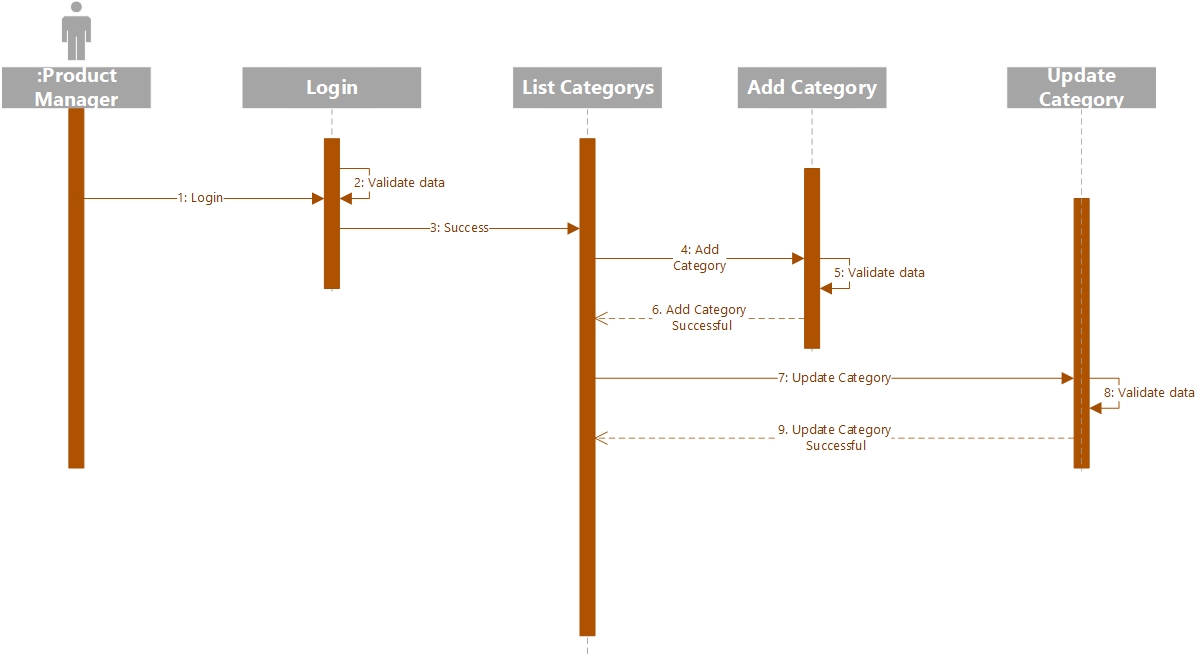
Picture 13 – Sequence Diagram Account Management

* + 1. **Brand Management Sequence Diagram**



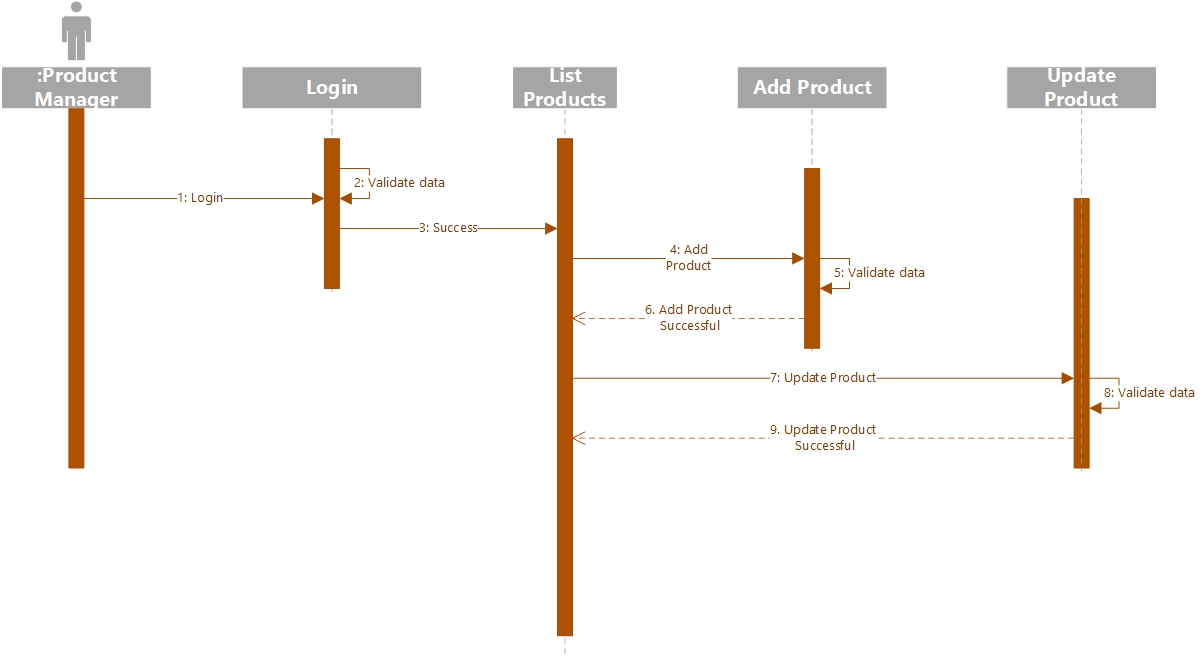
Picture 14 – Sequence Diagram Brand Management

* + 1. **Category Management Sequence Diagram**



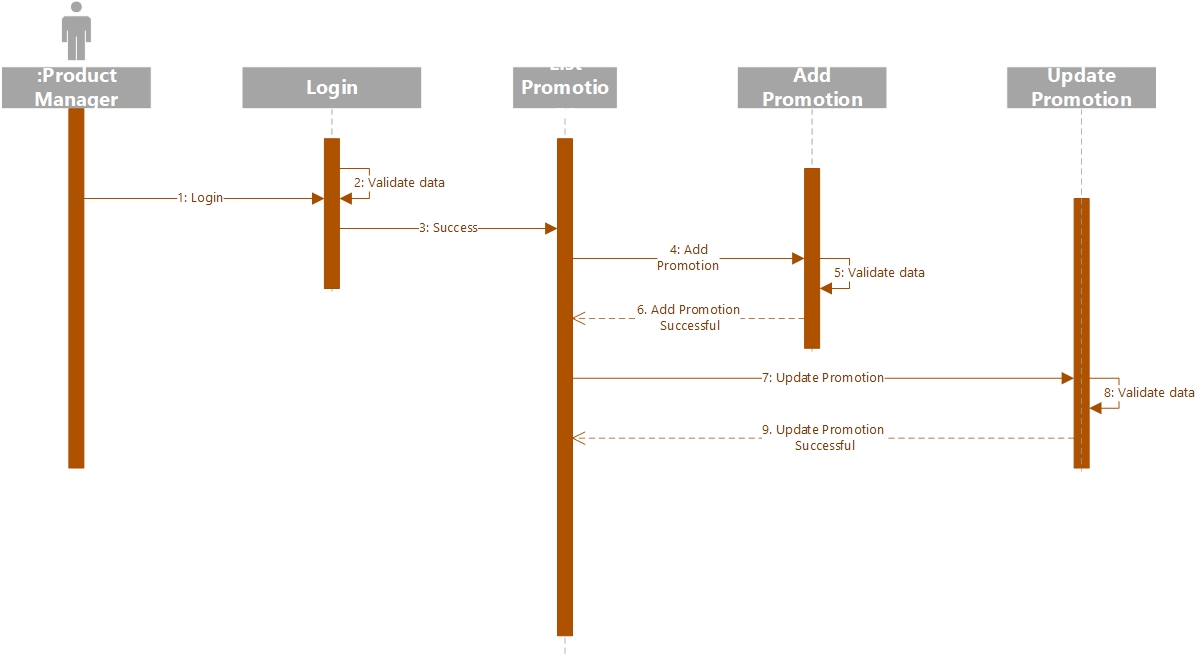
Picture 15 – Sequence Diagram Category Management

* + 1. **Product Management Sequence Diagram**



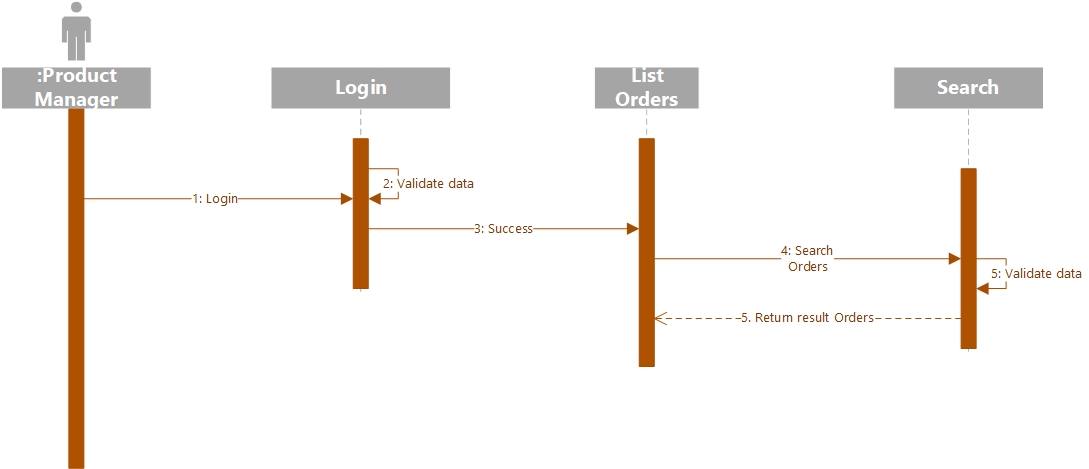
Picture 16 – Sequence Diagram Product Management

* + 1. **Promotion Management Sequence Diagram**



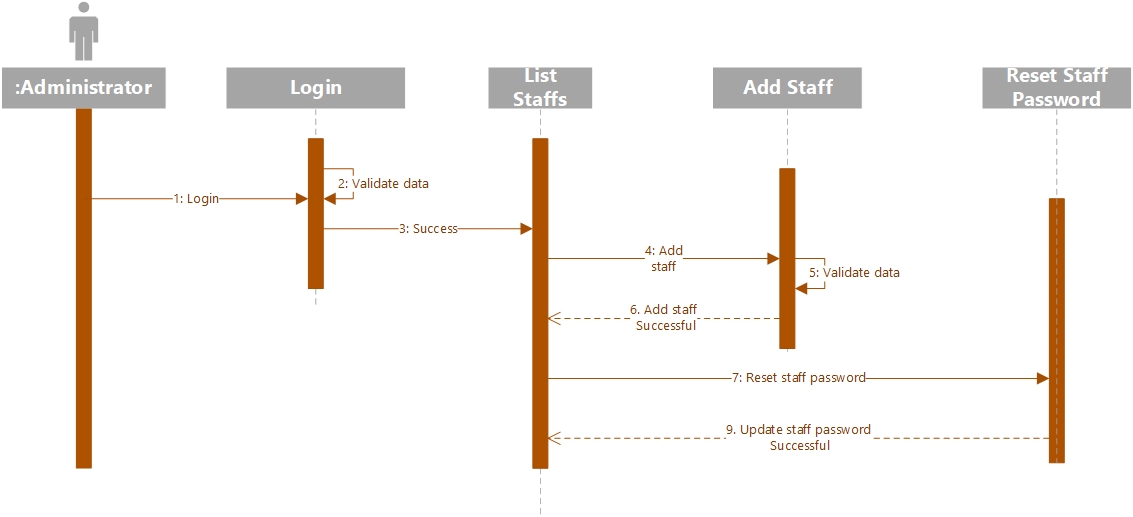
Picture 17 – Sequence Diagram Promotion Management

* + 1. **Order Management Sequence Diagram**



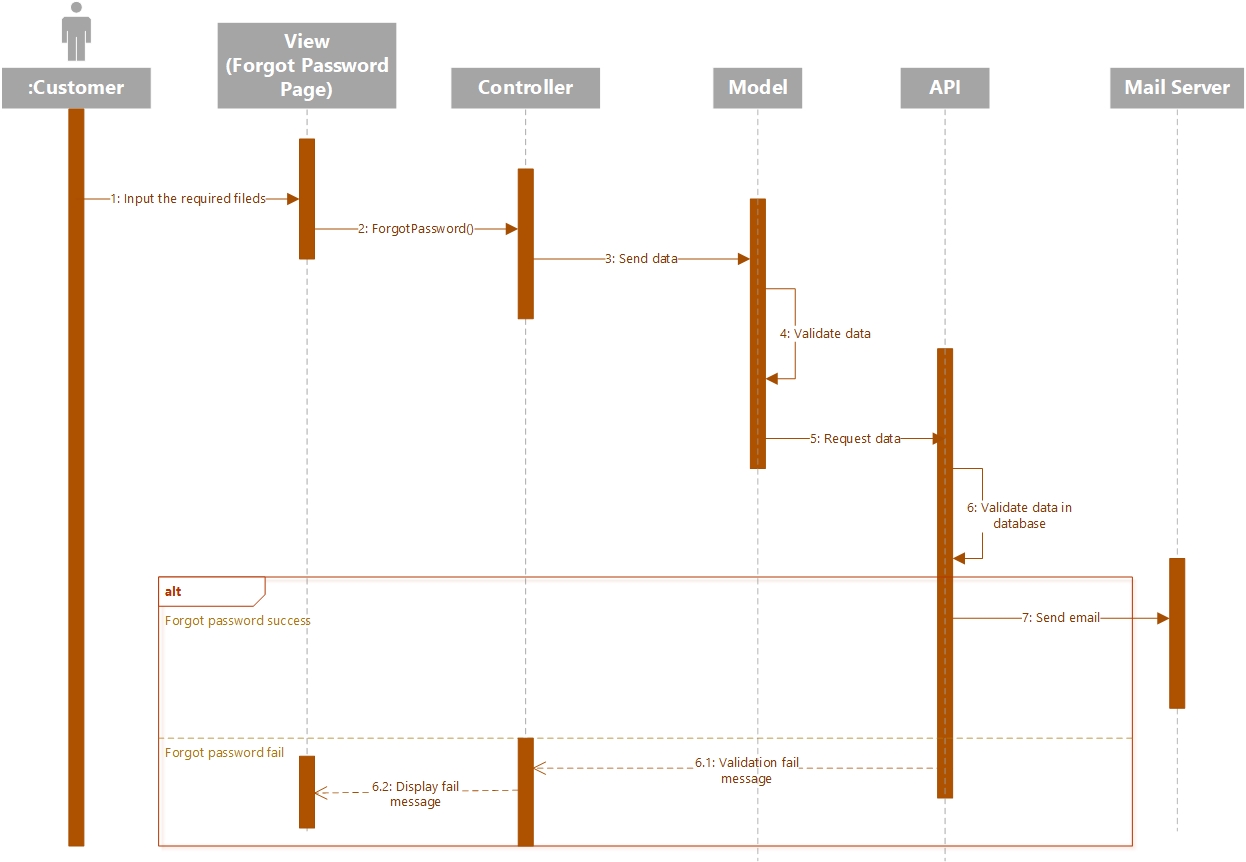
Picture 18 – Sequence Diagram Order Management

* + 1. **Staff Management Sequence Diagram**



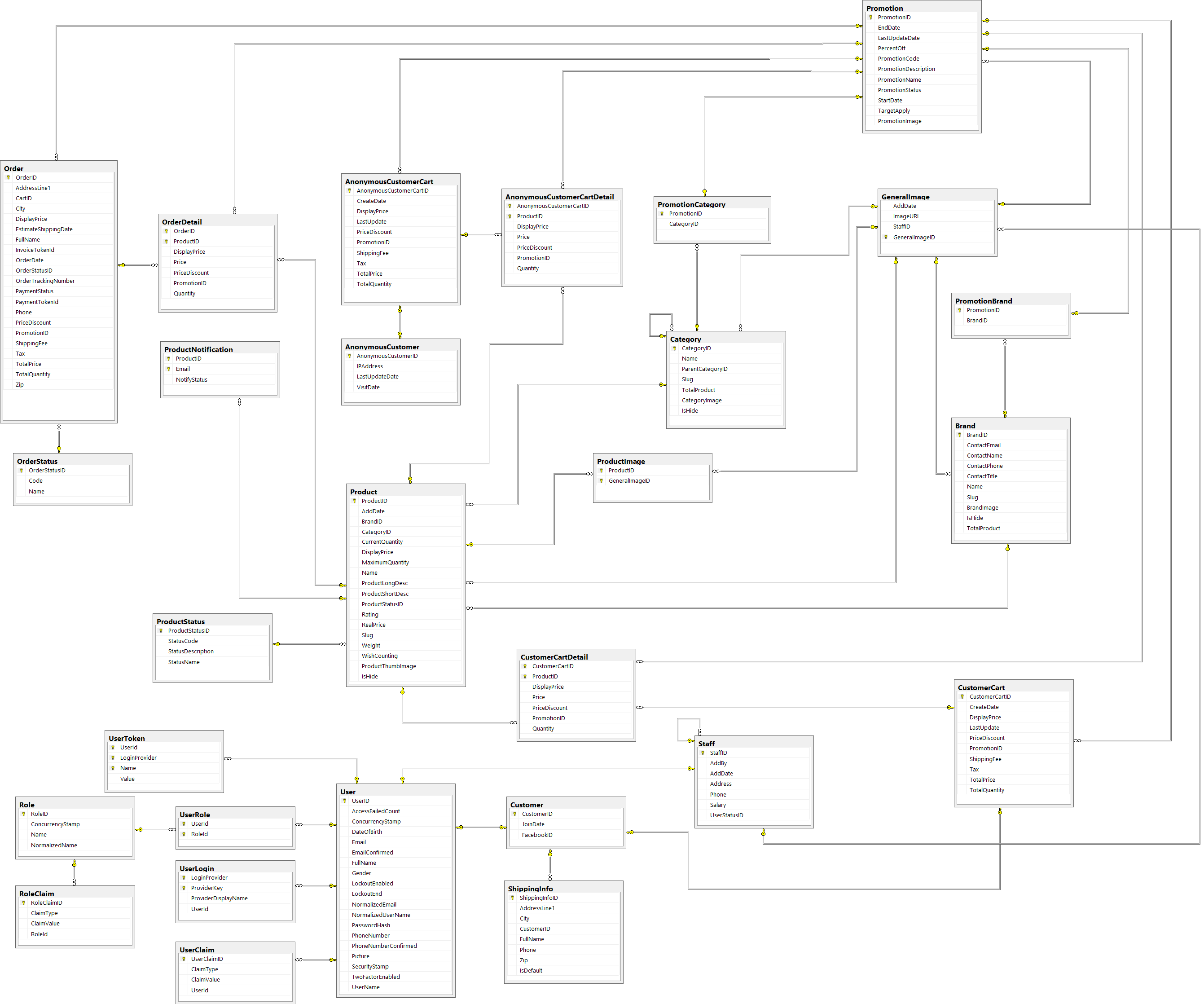
Picture 19 – Sequence Diagram Staff Management

* + 1. **Forgot Password Sequence Diagram**



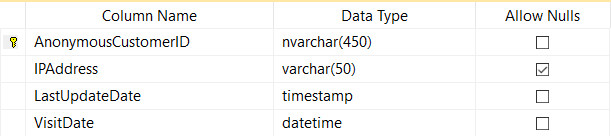
Picture 20 – Sequence Diagram Forgot Password

* 1. **Database Design**



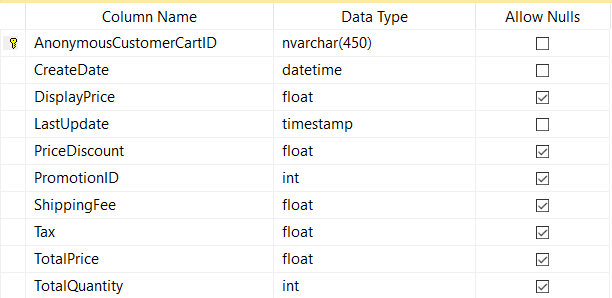
Picture 21 – Database Design Diagram

* + 1. **Table Anonymous Customer**



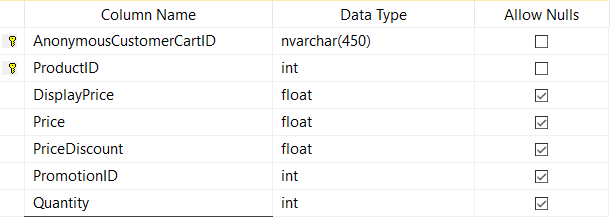
Picture 22 – Table Anonymous Customer

* + 1. **Table Anonymous Customer Cart**



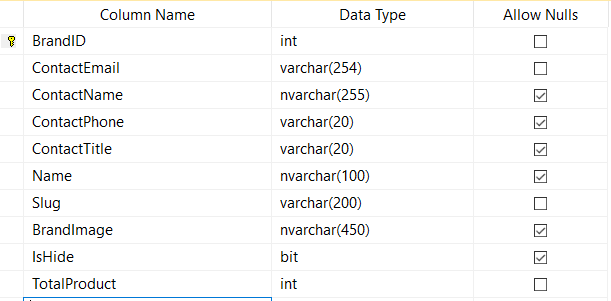
Picture 23 – Table Anonymous Customer Cart

* + 1. **Table Anonymous Customer Cart Detail**



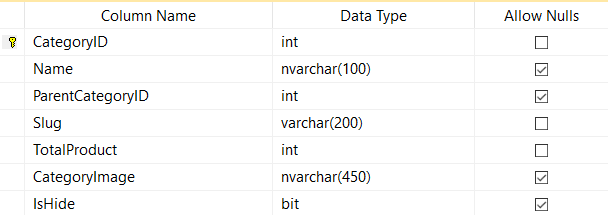
Picture 24 – Table Anonymous Customer Cart Detail

* + 1. **Table Brand**



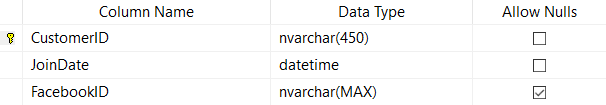
Picture 25 – Table Brand

* + 1. **Table Category**

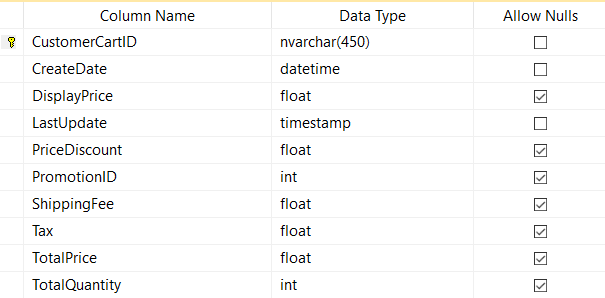


Picture 26 – Table Category

* + 1. **Table Customer**

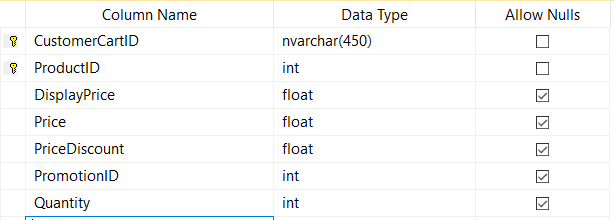


Picture 27 – Table Customer

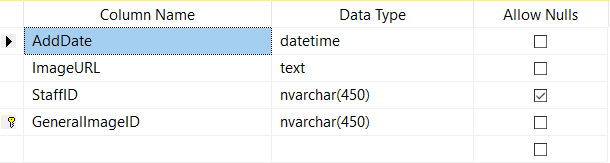
* + 1. **Table Customer Cart**

Picture 28 – Table Customer Cart

* + 1. **Table Customer Cart Detail**

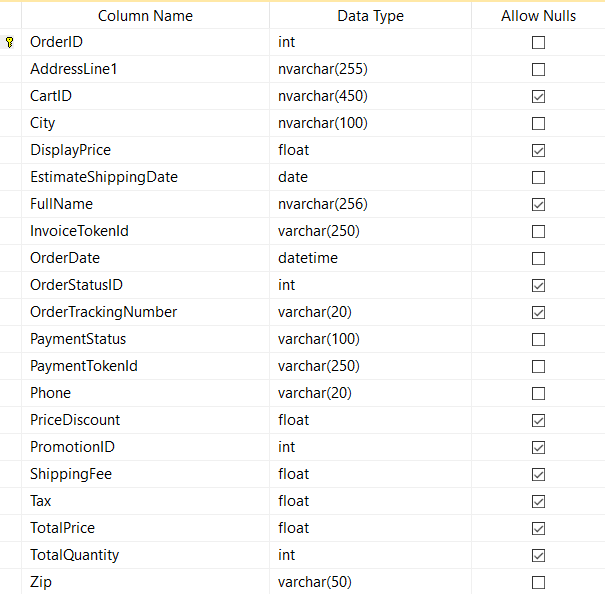


Picture 29 – Table Customer Cart Detail

* + 1. **Table General Image**

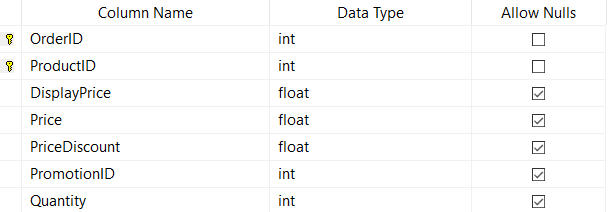
Picture 30 – Table General Image

* + 1. **Table Order**



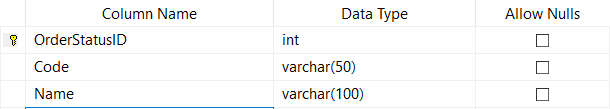
Picture 31 – Table Order

* + 1. **Table Order Detail**



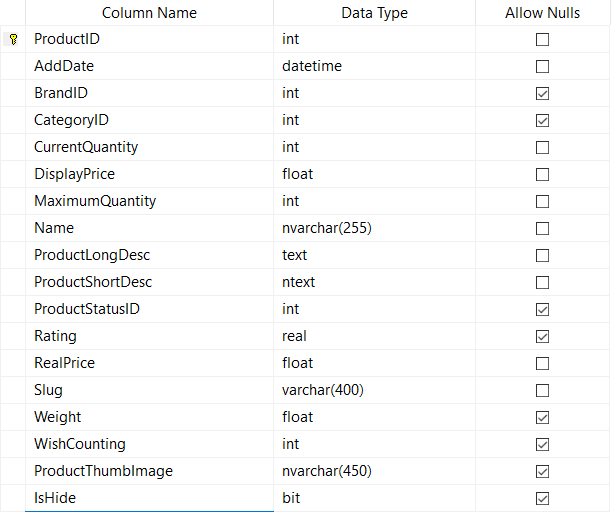
Picture 32 – Table Order Detail

* + 1. **Table Order Status**



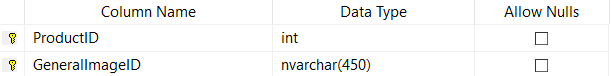
Picture 33 – Table Order Status

* + 1. **Table Product**



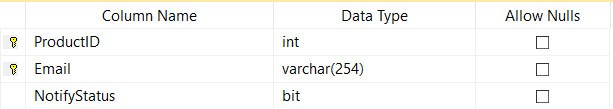
Picture 33 – Table Product

* + 1. **Table Product Image**

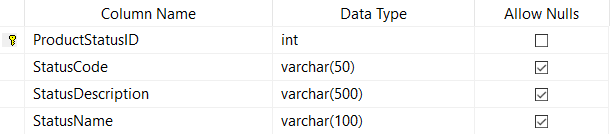


Picture 34 – Table Product Image

* + 1. **Table Product Notification**

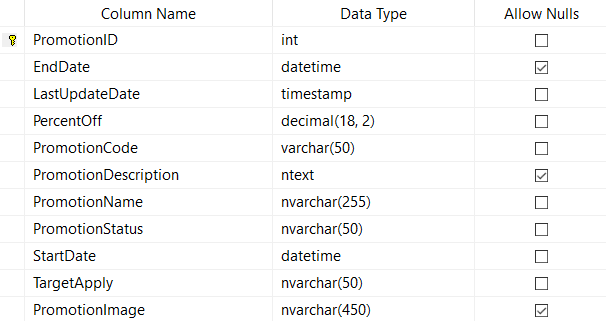


Picture 35 – Table Product Notification

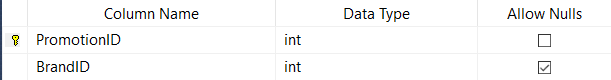
* + 1. **Table Product Status**

Picture 36 – Table Product Status

* + 1. **Table Promotion**



Picture 37 – Table Promotion

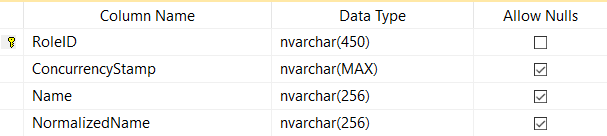
* + 1. **Table Promotion Brand**

Picture 38 – Table Promotion Brand

* + 1. **Table Promotion Category**

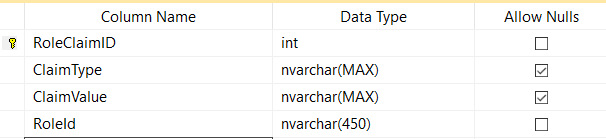


Picture 39 – Table Promotion Category

* + 1. **Table Role**

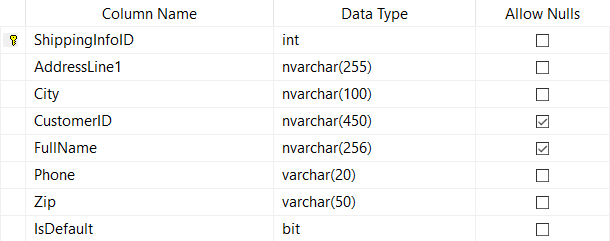
Picture 40 – Table Role

* + 1. **Table Role Claim**



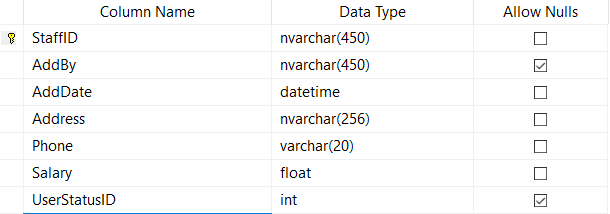
Picture 41 – Table Role Claim

* + 1. **Table Shipping Info**



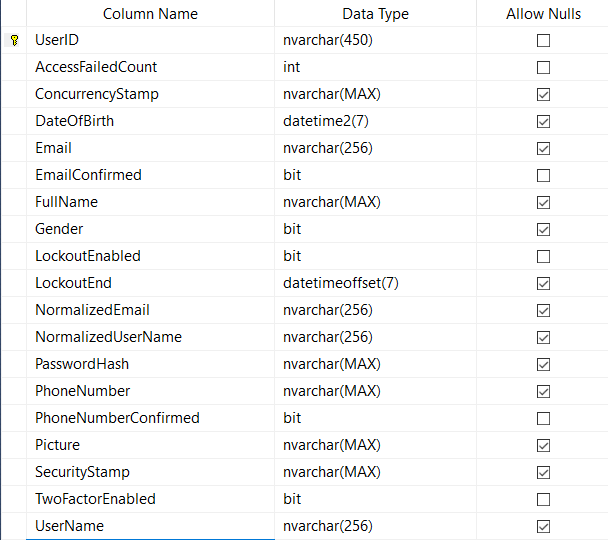
Picture 42 – Table Shipping Info

* + 1. **Table Staff**



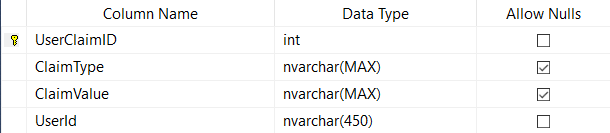
Picture 43 – Table Staff

* + 1. **Table User**



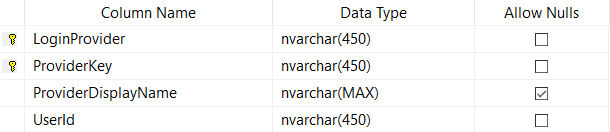
Picture 44 – Table User

* + 1. **Table User Claim**

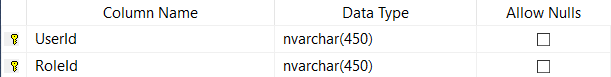


Picture 45 – Table User Claim

* + 1. **Table User Login**

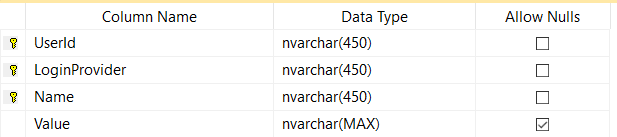


Picture 45 – Table User Login

* + 1. **Table User Role**

Picture 46 – Table User Role

* + 1. **Table User Token**



Picture 46 – Table User Token

1. **IMPLEMENTATION**
   1. **Create ASP.Net Core MVC**