Technical Story Card

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Version No. | Date | Prepared by / Modified by | Significant Changes |
| 0.1 | 10-Feb-2022 | Sakthi | Draft version |
| 1.0 | 19-Feb-2022 | Sakthi | Reworked based on Review comments |
| 1.1 | 04-Mar-2022 | Sakthi | Reworked based on Review comments |

Glossary

|  |  |
| --- | --- |
| Abbreviation | Description |
| UI | User Interface |
| DB | Database |
| UML | Unified Modeling Language |
| DD | Detailed Design |

Table of Contents

[1 Introduction 2](#_Toc83671932)

[2 Scope 2](#_Toc83671933)

[3 Non Scope of Change 2](#_Toc83671934)

[4 Design and Detailed technical updates 3](#_Toc83671935)

[4.1 Process model 3](#_Toc83671936)

[4.1.1 Use case Model 3](#_Toc83671937)

[5 Technical Architecture Diagram 4](#_Toc83671938)

[6 Class Diagram 4](#_Toc83671939)

[7 Sequence diagram 4](#_Toc83671940)

[8 ER Diagram for database Design 4](#_Toc83671941)

[9 Other Technical changes 4](#_Toc83671942)

[9.1 CI / Build relates tasks 4](#_Toc83671943)

[9.2 Non-functional Requirements and Design 4](#_Toc83671944)

[10 Additional details 5](#_Toc83671945)

[10.1 Open Questions / clarifications / Assumptions 5](#_Toc83671946)

[10.2 Additional notes to technical team 5](#_Toc83671947)

[11 References 5](#_Toc83671948)

# Introduction

* This Application is built using ASP.NET MVC as front-end and SQL Server as back-end.
* The objective of this project is to build an E-Commerce platform that enables product management.
* This system will allow the user to book order for products by taking inputs which includes:
* Product type
* Product seller
* County of Origin
* Price
* Details
* About of product
* User can also see the wallet where he/she can check coupons and discounts on the dashboard.
* This application also allows the user to check product brief description and later gives option to add it in the cart, proceeding in placing order for particular product.
* It will automatically set delivery duration.

# Scope

* In the modern age. Many people don’t have time to purchase physically.
* They have a hectic schedule throughout the day.
* There are many small sellers who want to expand their business, by acquiring more customers.
* With the help of computerized system, we can deliver products of good quality directly at door step of customers.
* E-Commerce platform is an automatic system which delivers data processing in very high speed in systematic manner.
* By using our system, they can maintain records very easily. Our system covers every area of online purchasing. In coming future there will be excessive need of E-Commerce platforms.

# Non Scope of Change

a) User Profile details (like Name, Age, Password, etc.).

b) Encryption and Decryption of Password stored in Database.

c) Upload of data using files.

# Design and Detailed technical updates

## Process model

### Use case Model

**UI Screen Flow**

### 

User

|  |  |
| --- | --- |
| Brief Description | Product Display |
| Basic Flow | At Home scree user select Our Product. Where he/she check the details and all about the product listed on the website. User can also check the details of specific product. |
| Alternate Flow | If user is unable to see the product he/she must need to contact to admin. |
| Validation | The details will be validated against all the validation rules and upon successful validation products will be successfully displayed. |
| Pre-Conditions | User clicked on Our Product. |
| Post-Conditions | User can check details about each and every product listed on the cart. |

**UI Screen Flow**

### 

User

### 

|  |  |
| --- | --- |
| Brief Description | Checkout |
| Basic Flow | At product page or cart user click on Checkout. Where he/she do final confirmation and proceed to choose payment mode. |
| Alternate Flow | If user is unable to do checkout he/she must need to contact to admin. |
| Validation | The details will be validated against all the validation rules and upon successful validation product will be successfully ordered. |
| Pre-Conditions | User clicked for Checkout at cart or product page. |
| Post-Conditions | User places successful order of chosen product. |

**UI Screen Flow**

### 

User

|  |  |
| --- | --- |
| Brief Description | **User Login** |
| Basic Flow | The user will enter its credentials to login to the system and upon successful validation the user will be able to access the system. |
| Alternate Flow | If the user is not register he/she needs to click on New user link to register or if user is already register and forgot his/her password, he can reset his password by clicking on Reset password link. |
| Validation | The user credentials will be validated against the database records. |
| Pre-Conditions | The user must be a registered user in order to access the system. He/she must provide his/her valid email id and password. |
| Post-Conditions | The user will be able to access dashboard and he/she can place order. |

**UI Screen Flow**

### 

User

|  |  |
| --- | --- |
| Brief Description | **User Register** |
| Basic Flow | 1. Click on Register Tab 2. Enter details like Email, Name, Address, password and confirm password 3. After entering all required details click on Register button |
| Alternate Flow | If user is unable to register he/she can contact Admin in order to get registered. |
| Validation | The entered details will be validated against all the validation rules and upon successful validation user account will be successfully created. |
| Pre-Conditions | The user must fill all the details with confirm valid data and email should be a valid, password should be greater than or equals to 8-digit and password and confirm password should be same. |
| Post-Conditions | The user details will get stored into the database and the user will be able to login using his/her credentials. |

**UI Screen Flow**

### 

### U

User

|  |  |
| --- | --- |
| Brief Description | Cart |
| Basic Flow | The user will select the product and add it in the cart where he or she can add more than one product which they will currently want to purchase. |
| Alternate Flow | If the user has not entered his or her credentials before or not registered in the application, he or she must register to proceed further. |
| Validation | The user credentials will be validated against the database records. |
| Pre-Conditions | The user must be registered user and must provide valid mobile number and email address. |
| Post-Conditions | The user will be able to add more products, remove the products and proceed to order the product. |

**UI Screen Flow**

### 

### U

User

|  |  |
| --- | --- |
| Brief Description | Order |
| Basic Flow | The user will select the product and add it in the cart where he or she can order the product by entering details and address. |
| Alternate Flow | If the user has not entered his or her credentials before or not registered in the application, he or she must register to proceed further. |
| Validation | The user credentials will be validated against the database records. |
| Pre-Conditions | The user must be registered user and must provide valid mobile number and email address along with correct address. |
| Post-Conditions | The user will be able to order the product, return the product and exchange the product. |

**UI Screen Flow**

### 

### U

User

|  |  |
| --- | --- |
| Brief Description | My Order |
| Basic Flow | The user will go to My Order and check the product which he/she have been ordered |
| Alternate Flow | If the user has not entered his or her credentials before or not registered in the application, he or she must register to proceed further. |
| Validation | The user credentials will be validated against the database records. |
| Pre-Conditions | The user must be registered user and must provide valid mobile number and email address along with correct address. |
| Post-Conditions | The user will be able to check his/her orders. |

**UI Screen Flow**

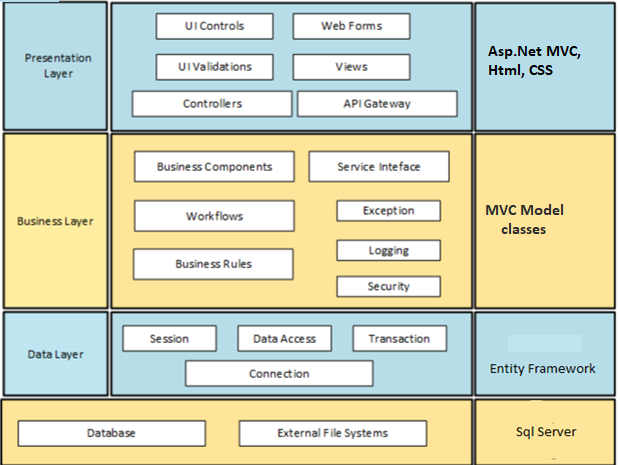
### 

### U

User

|  |  |
| --- | --- |
| Brief Description | Wallet |
| Basic Flow | The user will select the product and add it in the cart where he or she can pay through the and get some rewards. |
| Alternate Flow | If the user has not entered his or her wallet credentials he or she will not able to take rewards. |
| Validation | The user wallet will be validated against the database records. |
| Pre-Conditions | The user must be registered user and must provide valid mobile number and email address. |
| Post-Conditions | The user will be able to add more money, buy things and get more exciting rewards. |

**Technical Architecture Diagram**

****

|  |  |  |
| --- | --- | --- |
| Layer | Component | Example(s) |
| Presentation Layer | Views | User Registration()  Login()  Update() |
| Business Layer | Business components(Controller) | UserDetails ()  ProductDetails() |
| Data Layer | Transaction(Model) | LoginViewModel()  RegistrationViewModel()  SignOut()  AboutUs() |

**Presentation Layer**

The presentation layer has two main parts:

a) Client side code and b) Server side code.

The client side code store data from server (if needed) and does validation of input parameters. It also collects meta data information of the client which will be used by the application. The server side code is responsible for the UI components displayed on the screen. It interacts with the “Business Layer” which contains the business logic

Various Components in this layer are:

C.1.1. UI Controls: Interactive components in user interface like buttons, textbox, etc.

C.1.2. Web Pages: Forms with various fields to submit data to the server.

C.1.3. UI Validations: Validations to be done at UI level. E.g. Password validation, date validation, etc.

C.1.4. Controllers: User interface (UI) controllers serve as a connection between your UI and any business logic in your application that controls, or is instructed by, that UI

C.1.5. Views: Basic building block of UI. Customized for each screen or page.

**Business Layer**

This layer has the business logic. It collects data from the Presentation Layer, receives the request from the client, processes the requests, connects to the database layer if needed and sends the response back to the presentation layer.

Various Components in this layer are:

C.1. Workflows: A Workflow is a sequence of tasks that processes a set of data.

C.2. Business Rules: A business rule defines or specify constraints some aspect of business and always resolves to either true or false.

C.3. Exception: Exception Handling is a mechanism to handle runtime errors such as ResourceNotFoundException, etc.

C.5. Security: Security is a vast topic that encompasses many areas. These are available as services, which include data encryption like hashing of password, authentication, and authorization.

**Data Layer**

This layer will access the data stored in the database and files. When the “Business Layer” requests for information, it searches the tables in the database and provides the information. It also stores the data for each session. It stores the log of all the steps followed by the user in a given session.

Various Components in this layer are:

C.3.1. Session: It represents the connection between an application and the relational database that stores its persistent objects.

C.3.2. Data Access: It is a pattern that provides an abstract interface to some type of database or other persistence mechanism. By mapping application calls to the persistence layer, the DAO provides some specific data operations without exposing details of the database.

C.3.4. Connection: A Database connection is required to send commands and receive response usually in the form of a result set.

**Schema Diagram**

|  |
| --- |
| **New User** |
| Email |
| Name |
| Password |
| Confirm Password |
| Address |

|  |
| --- |
| **User Login** |
| Email |
| Password |
| Submit |

|  |
| --- |
| **Cart/Order** |
| Product Name |
| Product Seller |
| Price |
| Country of origin |
| About |
| Details |

|  |
| --- |
| **Products** |
| Product Name |
| Product Seller |
| Price |
| Country of origin |
| About |
| Details |

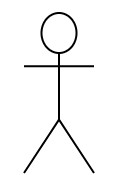
Contact

About

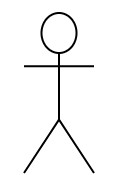
Wallet

Log Out

**Use Case Diagram**



Admin



User

# Class Diagram

|  |
| --- |
| Product |
| * Name * Seller * Price * C.O.I * About * Details |

|  |
| --- |
| Cart/Order |
| * Product Details * Date * Cost |

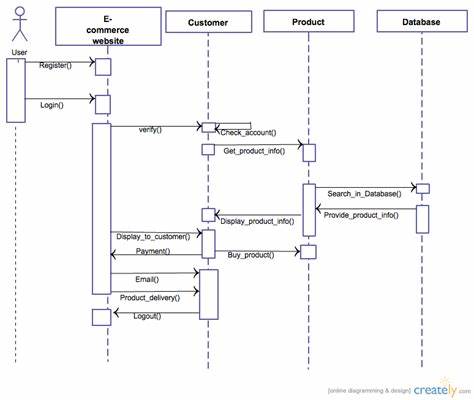
|  |
| --- |
| Admin login |
| * Email * Password |

|  |
| --- |
| User Registration |
| * Email * Name * Address * Password * Confirm Password |

|  |
| --- |
| User login |
| * Email * Password |

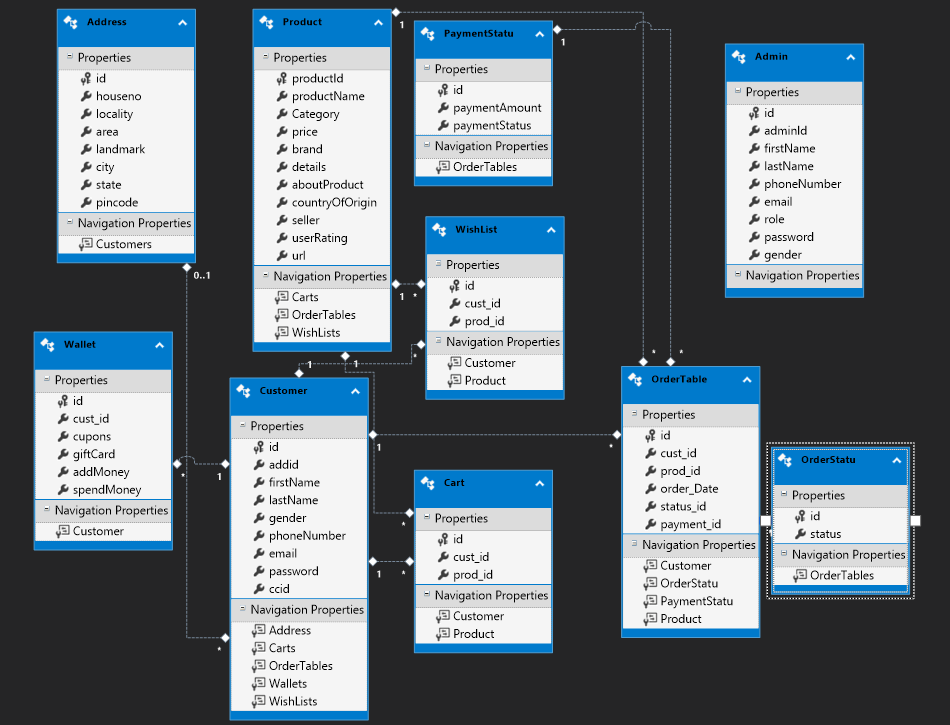
|  |
| --- |
| Update Password |
| * Email * Current Password * New Password * Confirm Password |

# Sequence diagram



Proceed with the Payment

# ER Diagram for database Design



# Other Technical changes

# Non-functional Requirements and Design

* The central repository should be platform independent so that it can be accessible and store application data via the web application.
* The server should be able to handle concurrent requests from different users.
* The system should provide confidentiality for user data using database encryption.
* The web application provides high availability and high accuracy in finding the locations.

# Open Questions / clarifications / Assumptions

**Assumptions:**

User Profile page is omitted from the scope because it is assumed that most people don’t care about their profile page once they login to the system.

# References

* <https://dotnet.microsoft.com/learn/csharp>
* <https://www.w3schools.com/css/default.asp>