**Shopping Cart System**

**Low Level Design (LLD)**



**DOCUMENT APPROVAL**

**Approvers of this document**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Department** | **Role** | **Signature** | **Date** |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**DOCUMENT CHANGE HISTORY**

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Version #** | **Author** | **Date** | **Description** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**TABLE OF CONTENTS**

1. ***Document Purpose***
2. ***Intended Audience***
3. ***Project Background & Objective***
4. ***Design Pattern***
5. ***Solution Diagram***
6. ***Solution Steps***
7. ***UML Diagrams***

***1.0 Document Purpose***

This document describes the solution architecture for Shopping Cart System.

***2.0 Intended Audience***

This document is intended as a reference for the following roles and stakeholders who are interested in the Shopping Cart System technical architecture.

|  |  |
| --- | --- |
| Role | Nature of Engagement in WB Classics Portal Technical Architecture |
| Product Owners/SME | Key stakeholder to ensure that the architecture is aligned with business goals. |
| Business Analysts | Business analysts are one of the stakeholders who are informed with the key architectural decisions. |
| Enterprise Architects | To enforce Shopping Cart Platform Architecture is aligned to business goals and architecture, architectural guidelines. |
| Solution Architects | To ensure solution design and architecture is aligned to business requirements, architectural guidelines. |
| Developers | Use Technical Architecture Document as the guiding document for detail design and implantation approach to align with Shopping Cart System |

***3.0 Project Background & Objective(s)***

***3.1* *Problem with Manual shopping***

The current system for shopping is to visit the shop manually and from the available product choose the item customer want and buying the item by payment of the price of the item.

* It is less user-friendly.
* User must go to shop and select products.
* It is difficult to identify the required product.
* Description of the product limited.
* It is a time consuming process.
* Not in reach of distant users.

***3.2* *Solution to the above issue***

In the proposed system customer need not go to the shop for buying the products. User can order the product he wish to buy at anytime and from anywhere. The shop owner will be admin of the system.

* The motive of this Shopping Cart Web Application is to allow the user to play with the search tool and create different combinatorial search criterion to perform exhaustive search.
* Provide Interactive interface through which a user can interact with different areas of application easily.
* A search engine that provides an easy and convenient way to search for products specific to their needs. The search engine would list a set of products based on the search term and the user can further filter the list based on various parameters.
  1. ***Scope of Application***
* The current system can be extended to allow the users to create accounts and add products in to the cart.
* This system can be implemented to any shop in the locality or to multinational branded shops having retail outlet chains. The system recommends a facility to accept the orders 24\*7 and a home delivery system which can make customers happy.
* If shops are providing an online portal where their customers can enjoy easy shopping from anywhere, the shops won’t be losing any more customers to the trending online shops such as flipkart or ebay.

***3.4 Project Objectives***

* Shopping Cart System leads to perform where seller can sell their products and customer can view or buy their products with the help of different payment options like cash on delivery, card payment, etc.
* Shopping Cart System will perform various operations like listing, creation, update and delete products for seller and customers can view without login. But, if a customer wants to buy items then he/she required to login himself/herself and then add to cart or buy them.

***4.0 Design Pattern***

|  |  |  |
| --- | --- | --- |
| # | Name | Description |
| 1. | API | Using HTTP requests, we will use the respective action to trigger various operations |
| 2. | Angular | To create and design the frontend |
| 3. | Database | To store and retrieve the information |

***5.0 Solution Diagram***

JWT Authentication

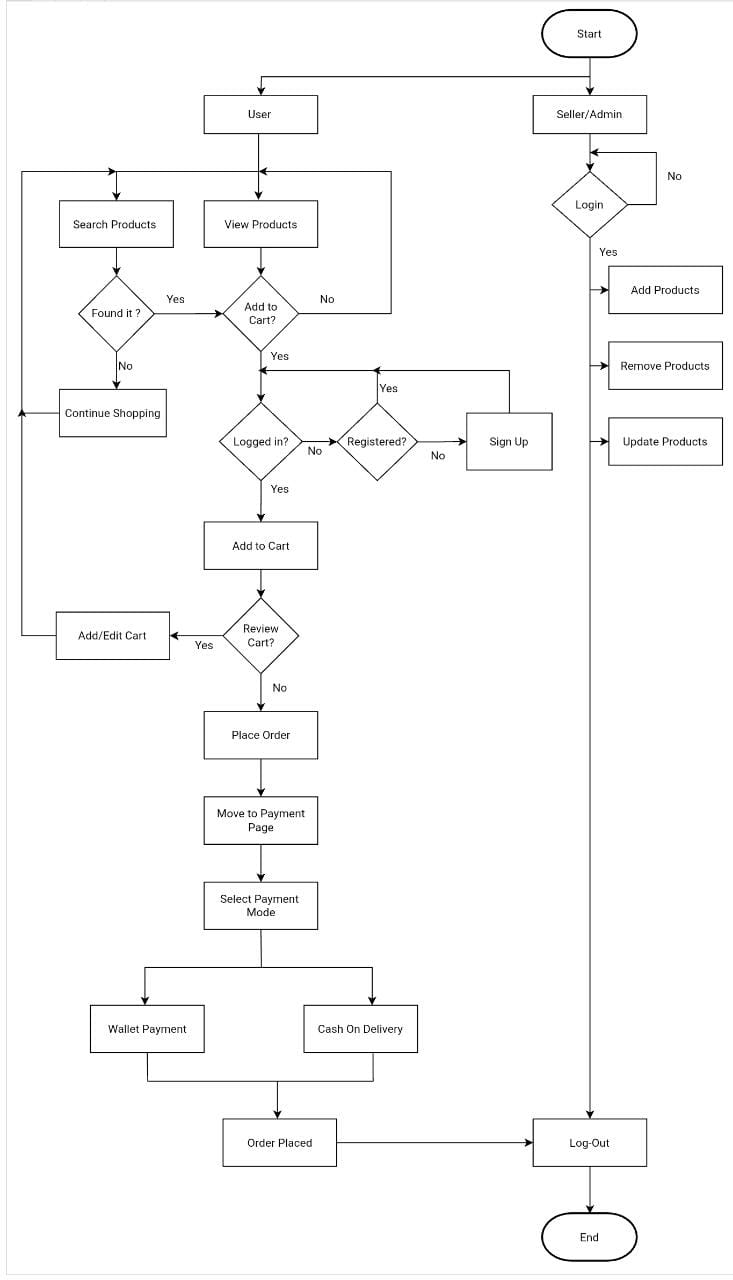
Entity Framework

Core

Web API Core

Database

Angular (Front End)

***6.0 Flowchart***

***6.0 Solution Steps:***

**User Registration:**

This is for customer and seller to login/sign up for the web app.

* For login, we are going to use email and password
* For sign up, we are going to use name, email, phone number, password and address

**Product details:**

This is about the product details in the web app. Some of the fields in the product table are product id, product name, price etc.

* The product can be seen by every user without login.
* Only seller can add and delete products.

**Cart Details:**

This is about the cart details in the web app. Some of the fields in the cart table are product id, product name, quantity etc.

* The user /customer has to sign up/ login to add products to the cart.
* The user can view and delete products from the cart.

**Order details:**

This is about the order details in the web app. Some of the fields in the order table are order id, product id, order date, delivery date, total amount etc.

* For placing the order user has to login or create the account if he do not have active account.
* User can view their previous orders and transactions.

**Payment details:**

This is about the payment details in the web app. Some of the fields in the payment table are payment id, order id, payment mode, amount etc.

* The user can make wallet payment and also cash on delivery.

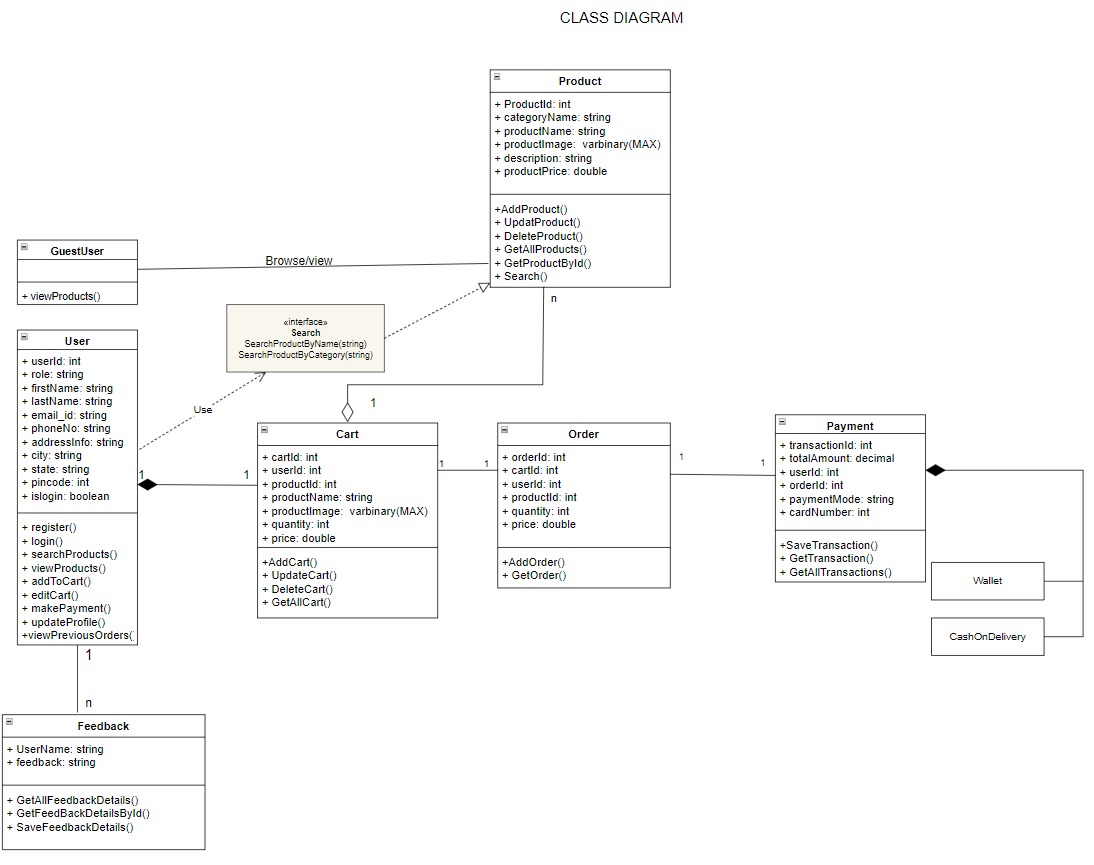
**Admin details:**

This is about the seller details in the web app. Some of the fields in the seller table are seller id, username, email, password etc.

* The seller can add, update and remove the products.

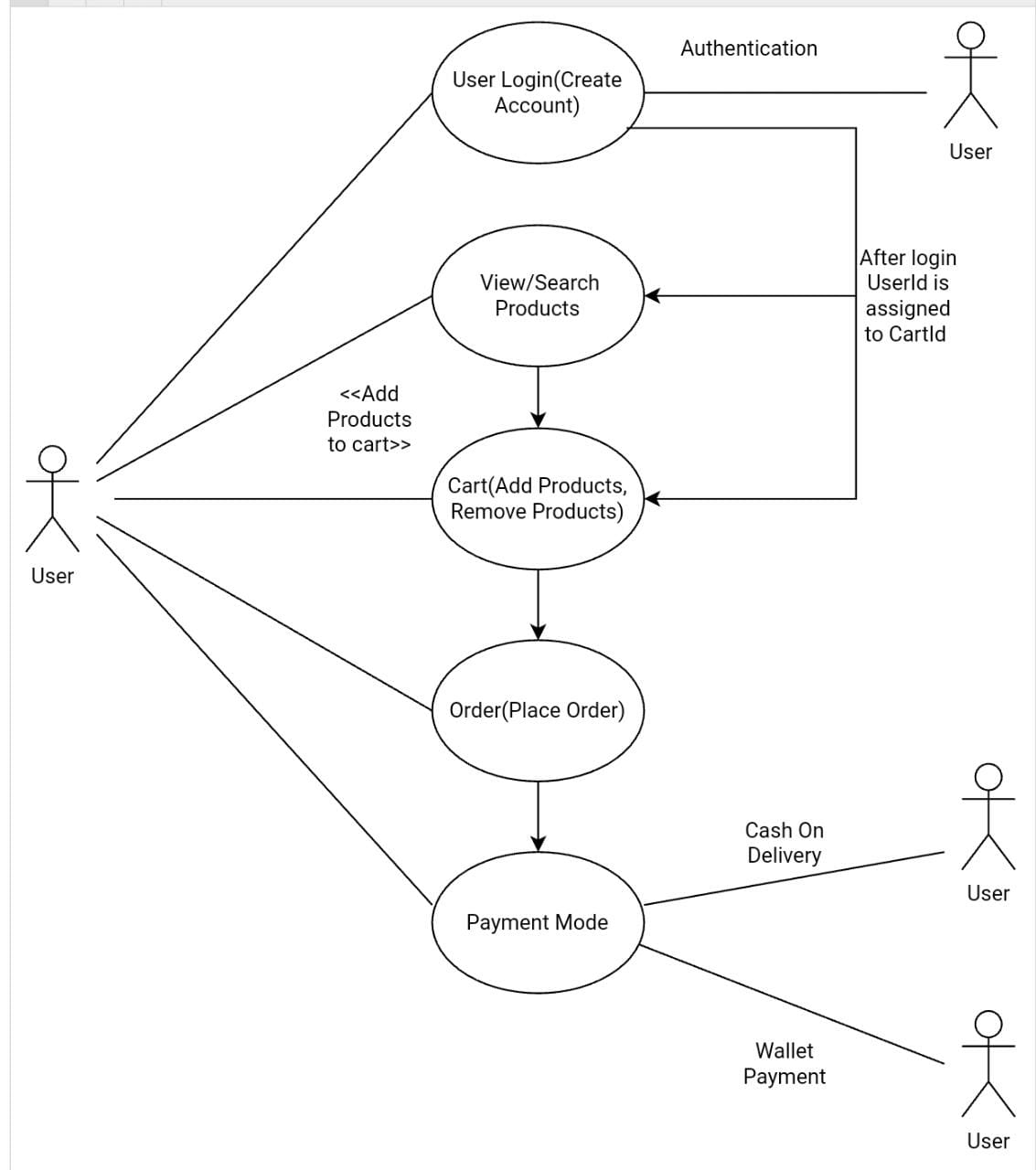
**7.0 UML Diagrams**

***Class Diagram*:**

****

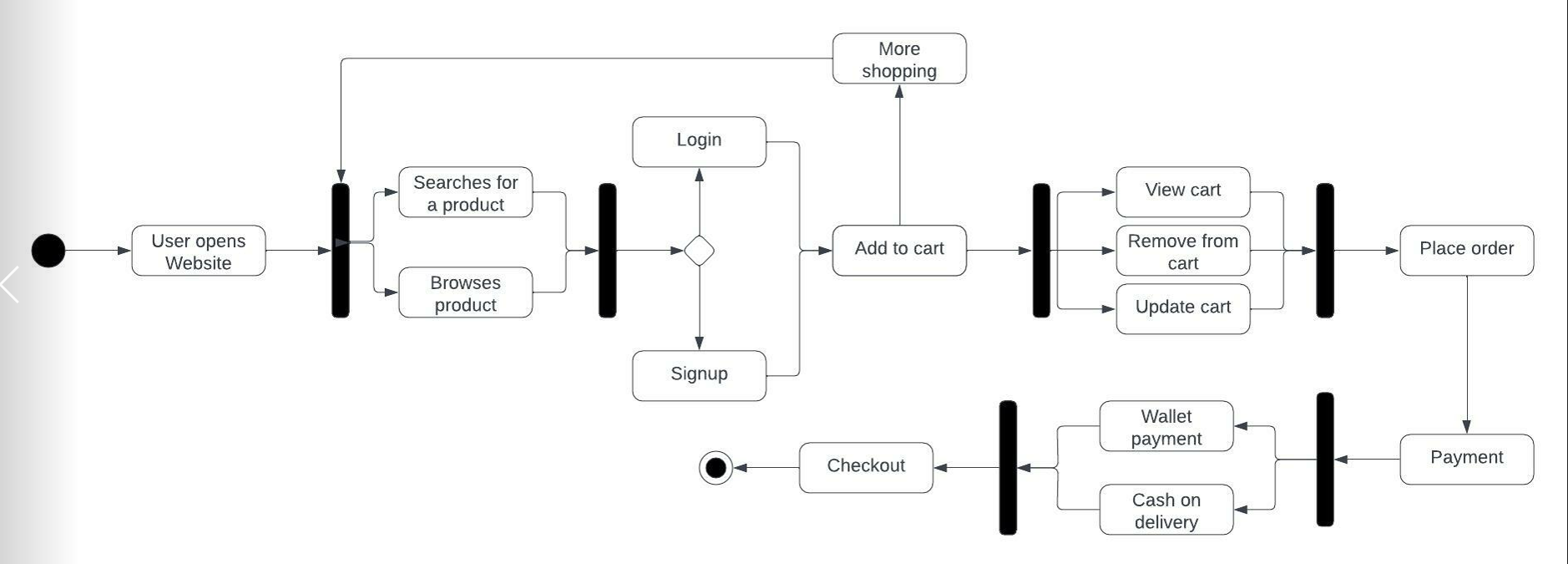
***Use case Diagram****:*

***Use case Diagram for User****:*

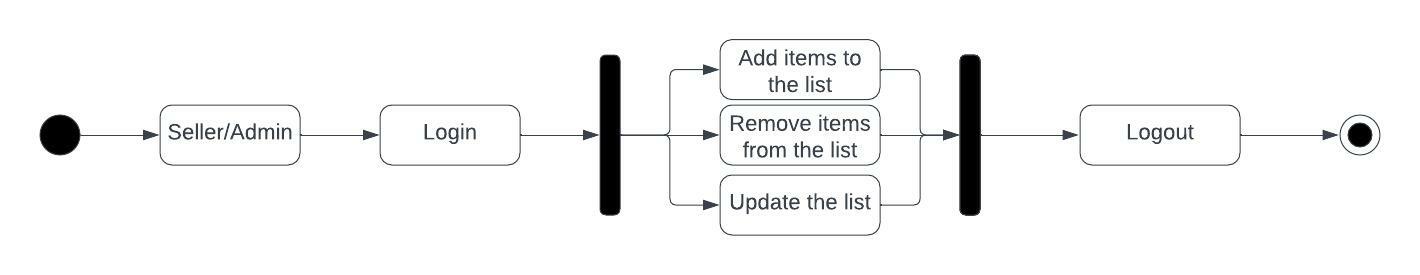
**

***Activity Diagram:***

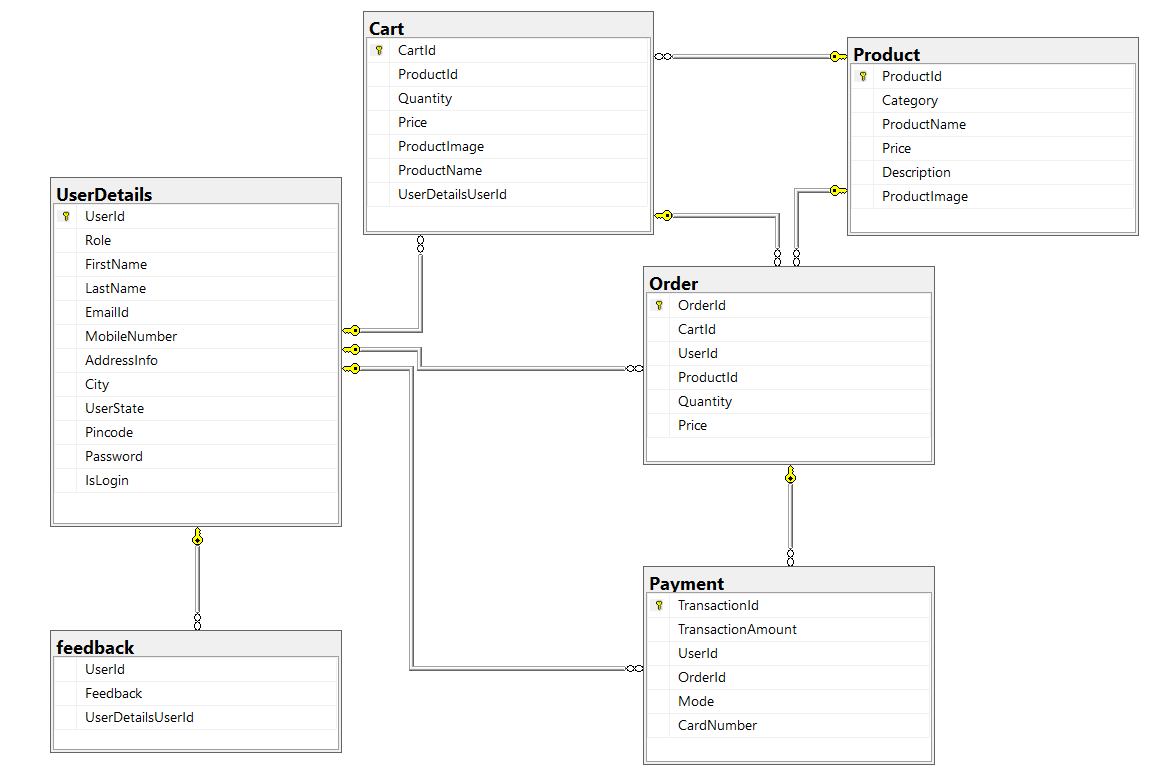
***Activity Diagram for user:***

****

***Activity Diagram for seller/Admin*:**

****

***Database Diagram***

******