**Shopping Cart System**

**Low Level Design (LLD)**



Date: 13/07/2022

Current Document Version: [*1.0*]

DOCUMENT APPROVAL

**Approvers of this document**

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

**Document Change History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Document Version #** | **Author** | **Date** | **Description** |
| 1.0 | Bhushan Vishwas Phulpagar | 13/07/2022 | Shopping Cart System LLD |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[1.0 Document Purpose 4](#_Toc94636300)

[2.0 Intended Audience 4](#_Toc94636301)

[3.0 Project Background, Objective(s) 4](#_Toc94636302)

[3.1 Project Background 4](#_Toc94636303)

[3.2 Project Objective 4](#_Toc94636304)

[4.0 Design Pattern 5](#_Toc94636305)

[5.0 Requirements 5](#_Toc94636306)

[6.0 Solution diagram 5](#_Toc94636307)

[7.0 Flow diagram 6](#_Toc94636308)

[8.0 Solution steps 6](#_Toc94636309)

[9.0 Classes/Functions 6](#_Toc94636310)

[10.0 Data model/table 6](#_Toc94636311)

[11.0 Env variables 6](#_Toc94636312)

[12.0 Integrations 7](#_Toc94636313)

[13.0 AWS Role 7](#_Toc94636314)

[14.0 HTTP Status Code 7](#_Toc94636314)

[15.0 Unit Testing 7](#_Toc94636314)

[16.0 Request 7](#_Toc94636314)

[17.0 Response 7](#_Toc94636314)

# Document Purpose

This document describes the solution architecture for Shopping Cart System.

# 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 System 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 |

# Project Background, Objective(s)

## Project Background

Shopping Cart System leads to perform where merchant 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.

## Project Objective

Shopping Cart System will perform various operations like listing, creation,

Update and delete products for merchant and customers can view them without log in. But if a customer wants to buys items then he/she required to login himself/herself and can add to cart or buy them.

# 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 front end |
| 3 | Database | To store and retrieve the information |

**5.0 Requirements**

**Hardware requirements: -** 64 GB Windows System

**Software requirements: -** 2019 Visual Studio, Microsoft SQL Server Management Studio, Visual Studio Code.

# 6.0 Solution Diagram

**WEB BROWSER CLIENT**

Angular (Front End)

**WEB APPLICATION**

Entity Framework

Web API in .NET CORE

HTML views,

Views models,

Controllers, Services

Authentication

SQL Database

**7.0 Flow Diagram**

USER/CUSTOMER

Home Page

(Open Website)

Choose Products

Cart

(Item add to cart)

Login

(If registered User)

Sign Up

(After registration redirect to the Login Page)

Order details

Order Checkout

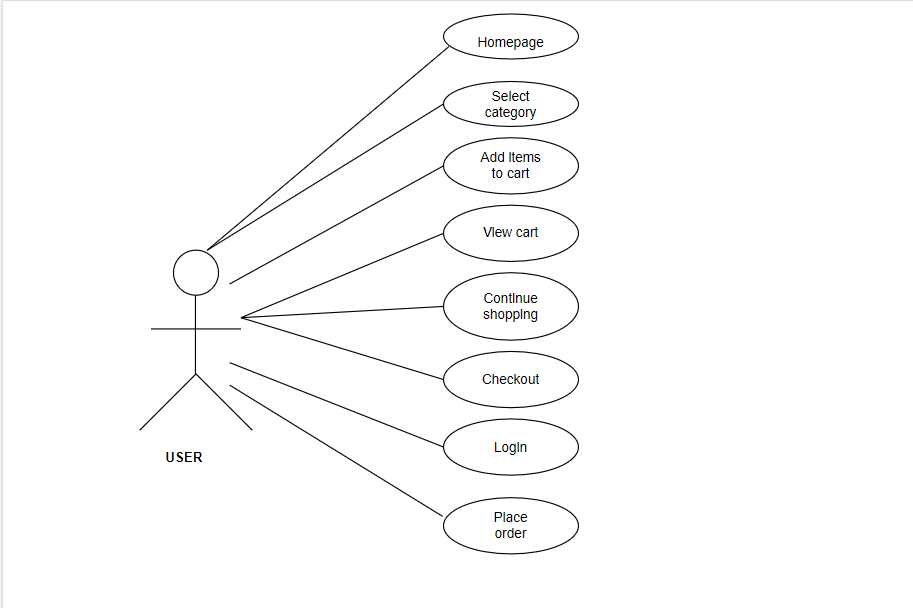
Address

(Add address details)

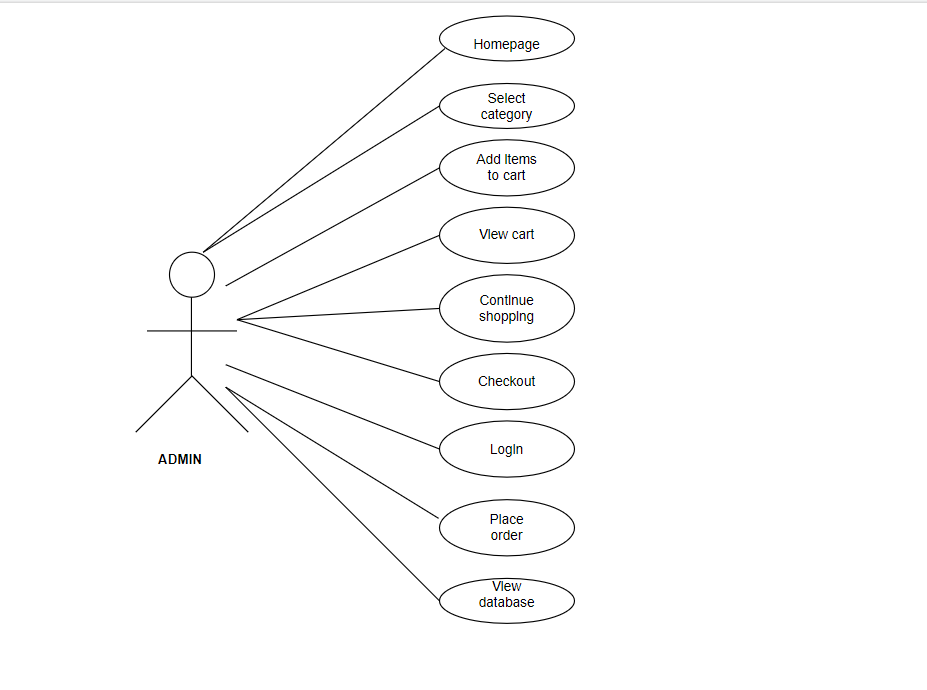
Payment

(Cash on delivery or Online payment)

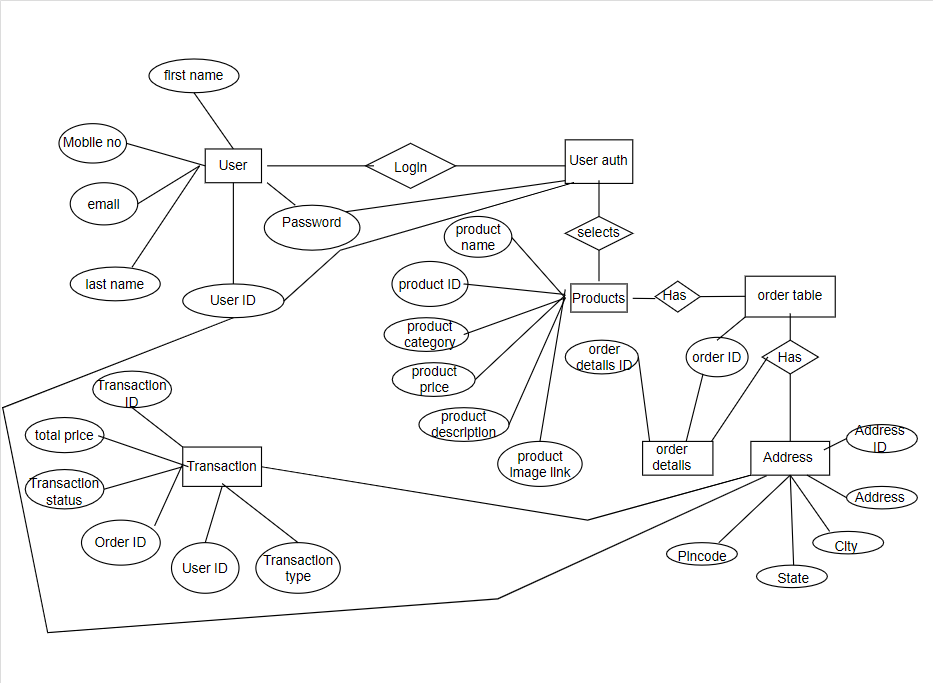
USER

****

ADMIN

****

**ERD Diagram:**

****

# 8.0 Solution Steps

**User Registration:**

This is for the Customer, Merchant and Admin to login/signup for the WebApp

* For login, we’re going to use an email id and password.
* For signup, we’re going to use the details like First Name, Last Name, email, etc. (Full Details in the table section)

**Product Details:**

This is about the product details which are available in WebApp. Here some fields include like Product ID, Product name, Price, etc. (Full details in the table section)

* This product can be seen by every user without logging in or register
* Only merchant can add and delete the products
* Admin can delete any product inside the WebApp. (If any inappropriate products are there.)

**Cart Details:**

The cart will display the total price of products those were added by the user. User will be able to remove item from the cart.

**Address Details:**

In this section, the user needs to add the proper address with the fields like flat no./house no., street /road, locality, pincode. (Full details in the table section)

- Once the address is added this user details can be used for the future references.

**Payment Details:**

In this section after addition of address user can make payment by using online payment mode or by using Cash on delivery mode of payment.

# 9.0 Classes/function

**Controllers** - contains application logic and passing user input data to service

For eg.

ProductController.cs

CartController.cs

UserDetailController.cs

**Services** - The middleware between controller and repository. Gather data from controller, performs validation and business logic, and calling repositories for data manipulation.

For eg.

ProductServices.cs

CartServices.cs

UserDetailServices.cs

**Repositories** - layer for interaction with models and performing DB operations.

**Models** – models declare the all the attributes of the database.

|  |  |  |
| --- | --- | --- |
| **#** | **Class** | **Description** |
| 1 | user.cs | Model holds the users schema details |
| 2 | AuthenticationController.cs | Has the functionality of Register, Login, Logout.  It uses Dto and user to do the functionality |
| 3 | Product.cs | It contains details about the product |
| 4 | OrderTable.cs  OrderDetailsTable.cs | It Has the functionality for the cart details and the full order detail are there is the order detail table |
| 5 | Payment.cs | Has the detail for the various payment method that is availableblob:file: |
| 6 | AddressTable.cs | It contains detail about the address and it is connected with the user table by using the foreign key “UserID” |
| 7 | ProductController.cs | Has all the API operation for product table |
| 8 | OrderTableController.cs  OrderDetailsTableController.cs | Has the API Details for the corresponding |
| 9 | AddressController.cs | Has the API Details for the use of Address table |
| 10 | PaymentController.cs | Has the API Details for the use of Payment table |

# 10.0 Data model/Table

**1. USER TABLE:**

|  |  |
| --- | --- |
| UserID | INT |
| FirstName | VARCHAR |
| LastName | VARCHAR |
| Email | VARCHAR |
| Phone | INT |
| Password | VARCHAR |

**2. ADMIN TABLE:**

|  |  |
| --- | --- |
| Email | VARCHAR |
| Password | VARCHAR |

**3. ORDER TABLE:**

|  |  |
| --- | --- |
| OrderID | INT |
| CartID | INT |
| UserID | INT |
| TotalPrice | DECIMAL |
| ProductID | INT |
| ProductName | VARCHAR |
| Quantity | INT |

**4. PRODUCT TABLE**

|  |  |
| --- | --- |
| ProductID | INT |
| ProductCategory | VARCHAR |
| ProductName | VARCHAR |
| Price | DECIMAL |
| Description | VARCHAR |
| Image | VARCHAR |

**5.CART TABLE:**

|  |  |
| --- | --- |
| CartID | INT |
| ProductID | INT |
| ProductName | varchar |
| UserID | INT |
| Quantity | INT |
| TotalPrice | DECIMAL |

**6. ADDRESS TABLE:**

|  |  |
| --- | --- |
| UserID | INT |
| AddressID | INT |
| Address | VARCHAR |
| City | VARCHAR |
| State | VARCHAR |
| PIncode | INT |

**7. TRANSACTION TABLE:**

|  |  |
| --- | --- |
| TransactionID | INT |
| TotalPrice | DECIMAL |
| TransactionStatus | VARCHAR |
| OrderID | INT |
| UserID | INT |
| Mode | VARCHAR |
| Code | VARCHAR |

# 11.0 ENV variables

**12.0 Integration**

N/A

# 14.0 HTTP Status Code

201 – Customer Registered

200 - Request succeeded

400 – Inputs are invalid

404 – Customer Not found

502 – Bad gateway

# 15.0 Unit Testing

|  |  |
| --- | --- |
| Project Name | Shopping Cart System |
| Created by | Bhushan Vishwas Phulpagar |
| Date of Creation | 15/07/22 |
| Date of review |  |

**For Registration of customers**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test CASE ID | TEST CASE SCENARIO | TEST CASE | PRE CONDITION | TEST STEPS | TEST DATA | EXPECTED RESULT | Actual Result |
| TC\_o1 | Customer registration | Enter the valid data to get registered | Customer needs to enter all the valid details | 1) Enter  customer\_first\_name: John  customer\_last\_name:doe  emai:customer@gmail.com  Phone:0123456789  age:22  address:delhi  gender:male  2) Enter Submit | <Valid Details | Successful registration | Successful registration |
| TC\_o2 | Customer registration | Enter invalid data to get registered | Customer needs to enter the valid details with wrong phone number type | 1) Enter  customer\_first\_name: John  customer\_last\_name:do  emai:customer@gmail.com  Phone:01234567  age:22  address:delhi  gender:male  2) Enter Submit | <invalid phone number> | Phone number should be of 10 digits | Phone number should be 10 digits |
| TC\_o3 | Customer registration | Enter all the required fields to get registered | Customer must enter all the required field as per specified in schema | 1) Enter  customer\_first\_name: John  customer\_last\_name:doe  emai:customer@gmail.com  Phone:0123456789  age:22  address:delhi  gender:male  2) Enter Submit | <All the Required fields are available> | Successful registration | Successful registration |
| TC\_o4 | Customer registration | Enter all the required fields to get registered | If customer misses one of the fields during registration which is marked as required in schema | 1) Enter  customer\_first\_name: John  customer\_last\_name:doe  Phone:0123456789  age:22  address:delhi  gender:male  2) Enter Submit | <email is missing> | You need to enter email | You need to enter email |

**For customers Updation**

# 16.0 Request

**Register Customer**

/customers

{

“customer\_first\_name”: “John”,

“customer\_last\_name” :”Doe”,

“email”:”customer@gmail.com”

“age” : 22,

“customer\_phone\_number” :” 0123456789”,

“gender”: “Male”,

“customer\_address”:”Jamshedpur”,

}

**17.0 Response:**

**Register Customer**

**If valid details**

{

"message": "Customer registered successfully"

}

status code: 201

**If invalid details**

{

“message”: “Inputs are not valid”

}, status code: 400

**If server encounters unexpected error**

{

“message”:” Internal server error”

}, status code: 500