-



**DESIGN MODELLING DOCUMENT**

-Truong Nguyen Viet Quang-

-HE182422-

– Hanoi, Sep 2024 –

**Table of Contents**

[I. Record of Changes 3](#_heading=h.gjdgxs)

[II. User Interface Design 4](#_heading=h.30j0zll)

[1. Feature Name1 4](#_heading=h.1fob9te)

[1.1 SubFeature Name1.1 4](#_heading=h.3znysh7)

[1.2 SubFeature Name1.2 4](#_heading=h.2et92p0)

[2. User Authentication 4](#_heading=h.tyjcwt)

[2.1 User Login 4](#_heading=h.3dy6vkm)

[2.2 User Register 4](#_heading=h.1t3h5sf)

[2.3 User Profile 4](#_heading=h.4d34og8)

[3. System Administration 5](#_heading=h.2s8eyo1)

[3.1 Master Data 5](#_heading=h.17dp8vu)

[III. High-Level Design 6](#_heading=h.3rdcrjn)

[1. Software Architecture 6](#_heading=h.26in1rg)

[1.1 Sub-Systems/Components 6](#_heading=h.lnxbz9)

[1.2 External Interfaces 6](#_heading=h.35nkun2)

[2. Package Diagram 7](#_heading=h.1ksv4uv)

[3. Database Design 7](#_heading=h.2jxsxqh)

[IV. Detailed Design 8](#_heading=h.z337ya)

[1. <Feature/Function Name1> 8](#_heading=h.3j2qqm3)

[1.1 Class Diagram 8](#_heading=h.1y810tw)

[1.2 <Sequence Diagram Name1> 9](#_heading=h.4i7ojhp)

[1.3 <Sequence Diagram Name2> 9](#_heading=h.2xcytpi)

[1.4 … 9](#_heading=h.1ci93xb)

[2. <Feature/Function Name2> 9](#_heading=h.3whwml4)

# I. Record of Changes

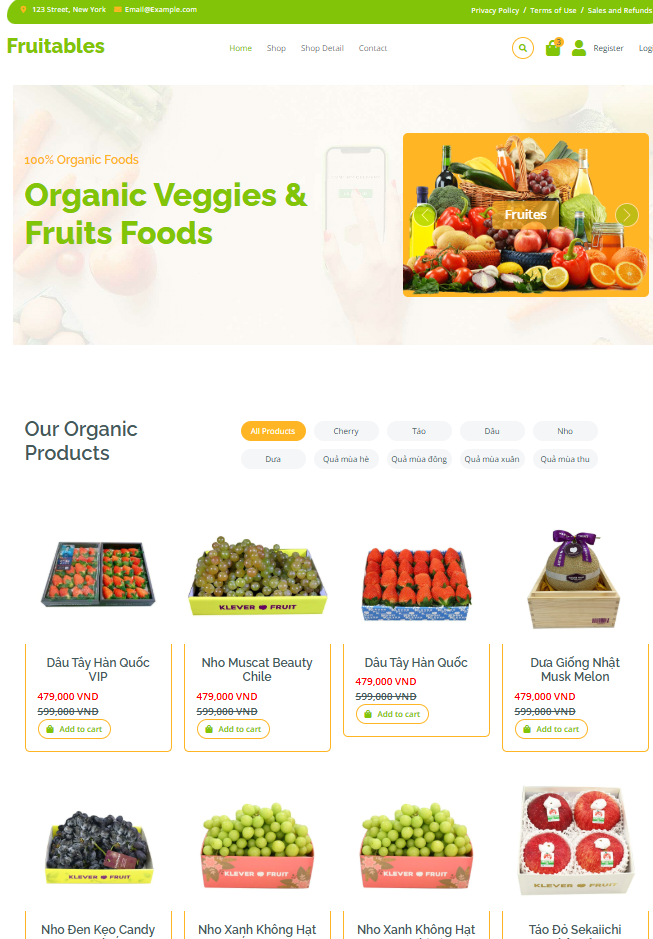
| **Date** | **A/M/D\*** | **In charge** | **Description** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

\*A - Added M - Modified D - Deleted

# II. User Interface Design

## 1. System Access

### 1.1 Homepage

**

*This screen allows users to perform the following functions:*

*-View Product List: Display features and latest products.*

*-Search Products: Enter keywords to search for products by name or category.*

*-Add Products to Cart: Users can select products and add them to the cart for purchase.*

*-Navigate Between Categories: Users can browse products through navigation links.*

*-Access Cart and Checkout: Users can view their cart and proceed to checkout if logged in.*

*-View Customer Reviews: Display testimonials from previous buyers.*

*-Limited Time Offers: Display promotional banners and discounts.*

*On this screen, users can also:*

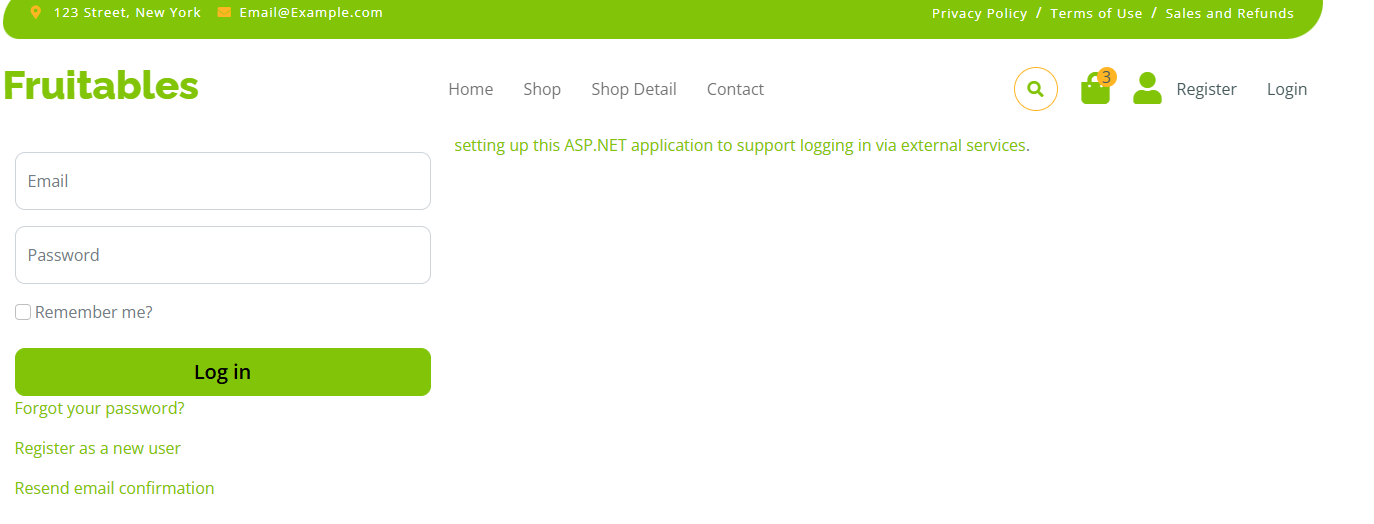
*-Login/Register: Required for adding products to the cart and making purchases.*

*-Edit Cart: Logged-in users can update product quantities or remove items from the cart.*

*-Admin Management: Admins can add, edit, or delete products from the system.*

| **Field Name** | **Description** |
| --- | --- |
| Product Name | Data type: string, max length of 100 characters |
| Product Image | Data type: URL string, must be a valid image link |
| Price | Data type: decimal, formatted as currency (VND) |
| Category | Data type: string, max length of 50 characters |
| Add to Cart Button | Functionality: Adds product to the cart (available for logged-in users) |
| Navigation Menu | Functionality: Allows navigation to Home, Shop, Categories, Contact, Register, and Login pages |
| Shopping Cart | Functionality: Displays the number of items in the cart and allows checkout |
| Banner Section | Data type: HTML content, includes promotional text and a call-to-action button |
| Testimonials | Data type: list of customer feedback, each with a name, image, and review text |

### 1.2 User Login



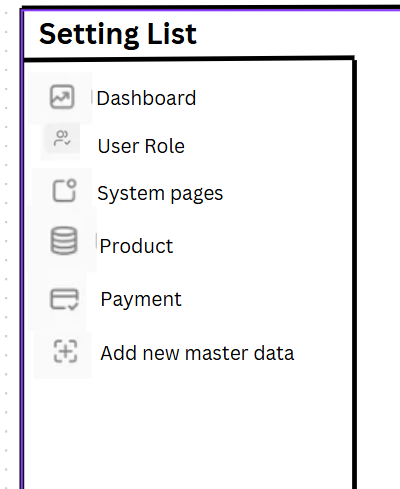
| **Field Name** | **Description** |
| --- | --- |
| ***Traditional Login*** | |
| Email | User input email |
| Password | User input password |
| Remember Me | Checkbox to remember user login info |
| Login Button | Button to send info to login |
| Forgot Password | A link to redirect to the page for user to take the forgotten password |
| Register as a new user | A link button to redirect to the register page |
| Message | A line to show error notification based on user input, submit |
| ***Google Login*** | |
| Google Button | Google login button |

### 

### 

## 2. Master Data

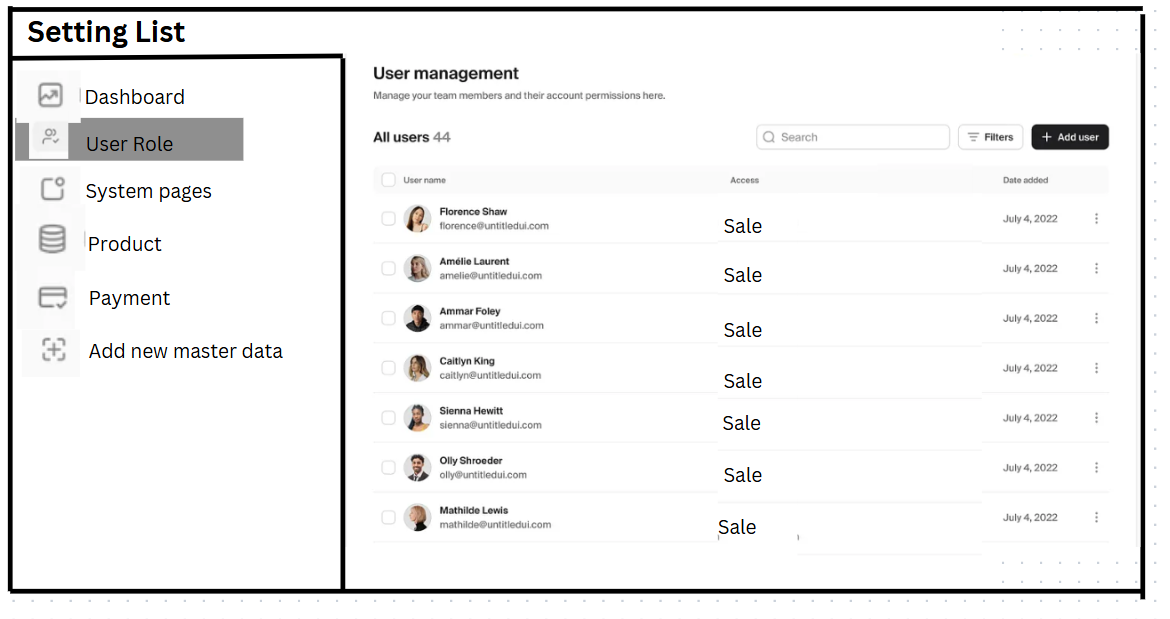
### 2.1 Setting List

**

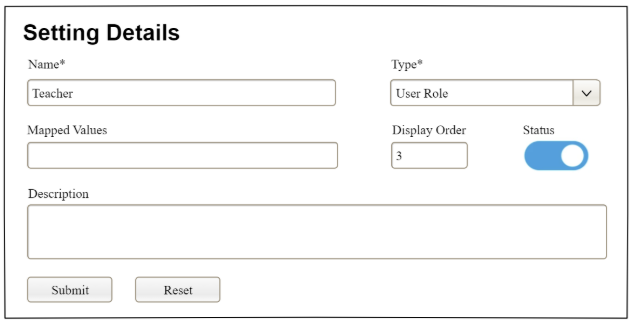
* This settings list is presented as a sidebar navigation, with specific fixed settings items.
* By default this page allows admin to edit master data of dashboard, user role, system pages, product, payment and add new master data
* When you click on each setting item in the list, the details of that setting will appear next to it.

| **Field Name** | **Description** |
| --- | --- |
| Dashboard | Redirect to dashboard page |
| User Role | Manage user roles and permissions |
| System Pages | Configure visibility of system pages |
| Product | Manage product settings and categories |
| Pament | Configure payment methods and gateways |
| Add New Master Data | Add new configurable master data |

### 2.2 Setting Detail



* When the admin selects a specific type of setting, the details of that setting will appear next to the navigation bar, as shown in the picture. For example, when the admin clicks on the "User role" setting, the details of that setting will appear next to it.

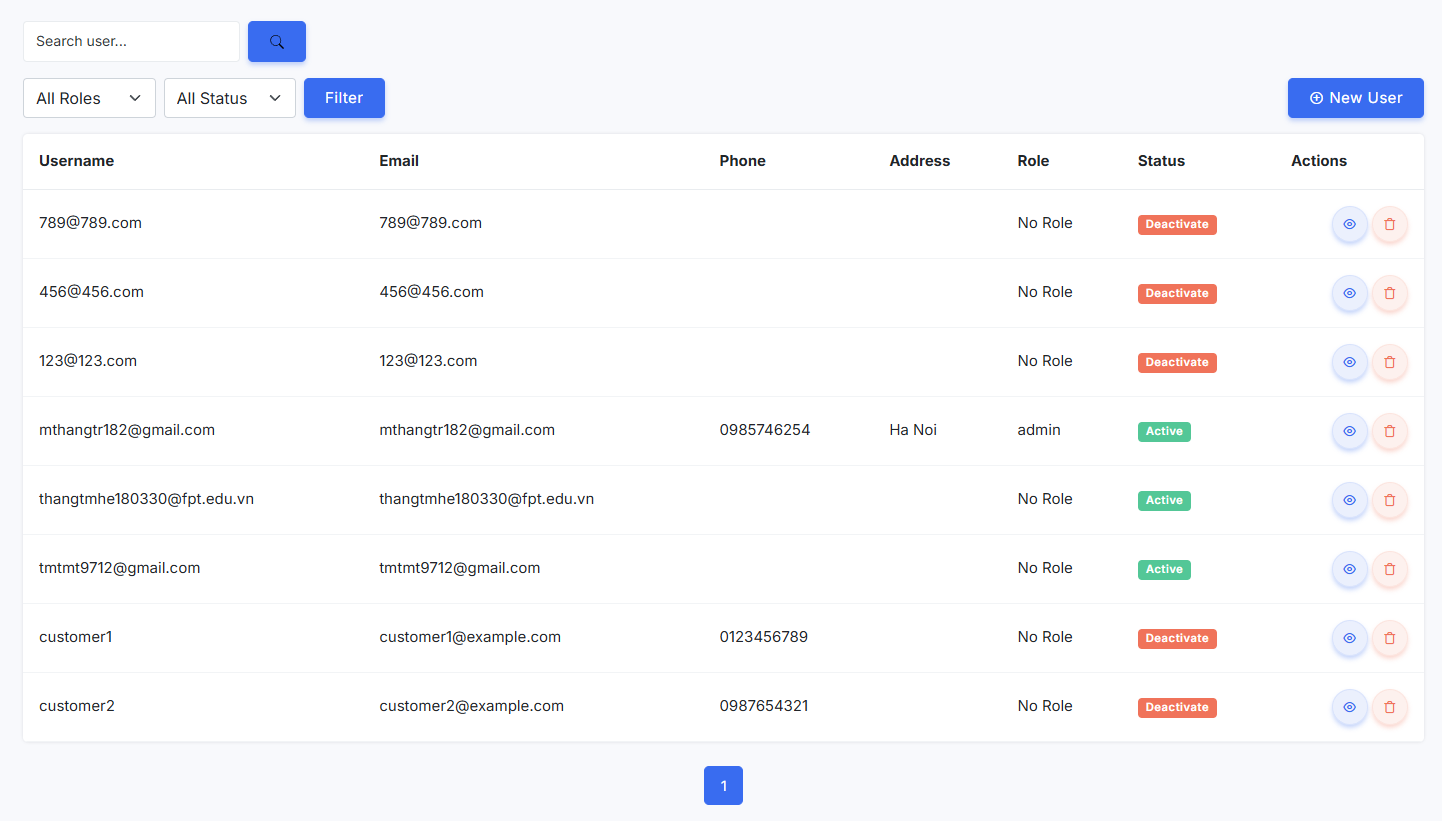


| **Field Name** | **Description** |
| --- | --- |
| Name | Name of the setting |
| Type | Type of the setting, filled with the list of setting types  Default value: the first type in the list |
| Mapped Values | Supplementary information for the setting (if any) |
| Order | Display order of the setting: the order of the setting type, displayed among the list of settings with the same type |
| Status | Status of the setting: Active or Inactive  Default value: Active |
| Description | Description of the setting |
| Submit | Click to store new or updated setting details |
| Reset | Click to reset the changes use has made on the screen fields back to the initial values when the screen is loaded |

### 

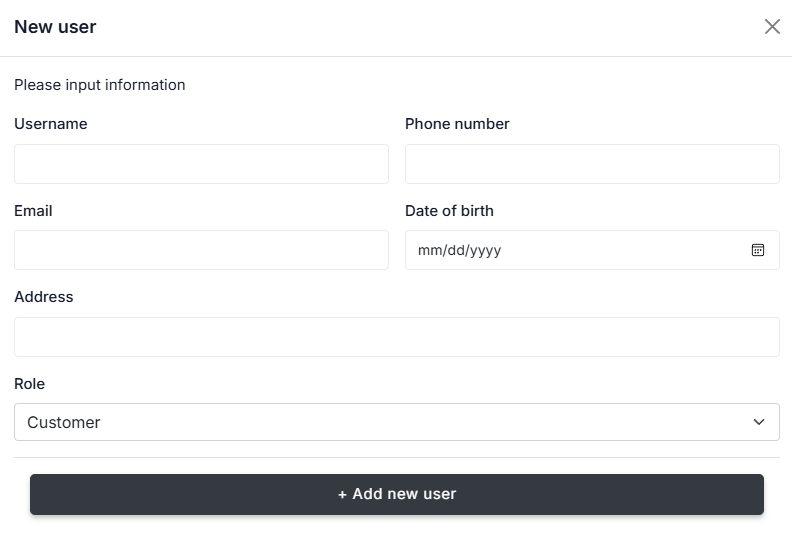
## 3. User Accounts

### 3.1 Account List



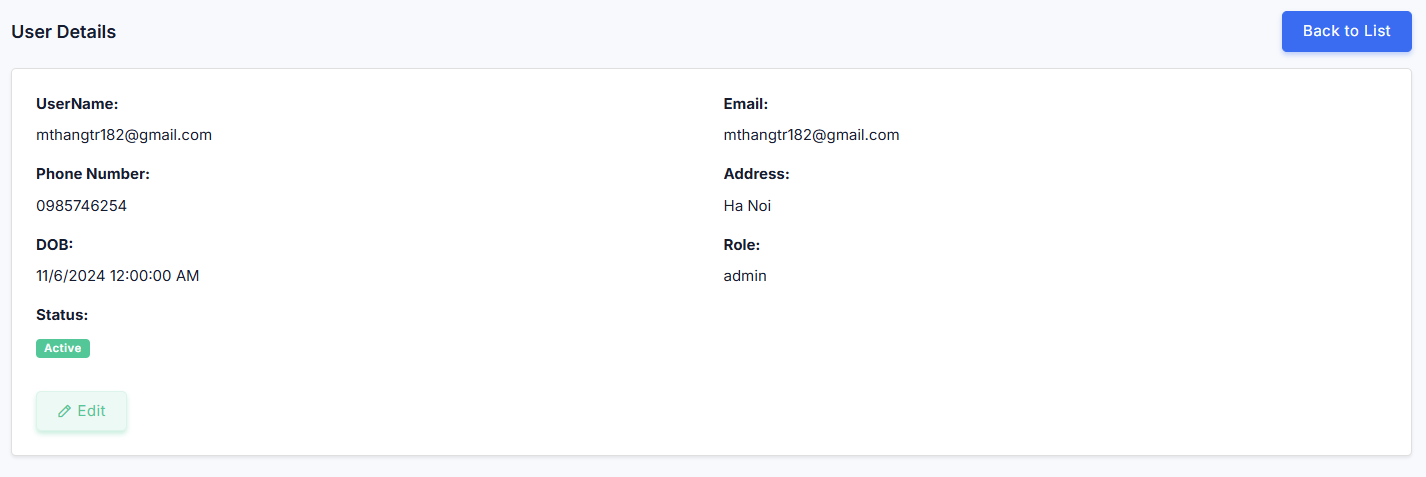
| **Field Name** | **Description** |
| --- | --- |
| Pagination | To handle the amounts of data to show |
| Search Bar | Input field to search for accounts by name, email, or ID |
| Add new user button | A button to add new account |
| ***Account Table*** | |
| Table Headers | The table consists of the following headers:  - **Username**  - **Email**  - **Phone**  - **Address**  - **Role**  - **Status**  - **Actions** |
| User Rows | Each row displays user details such as email, phone number, address, role, and status. |
| Status Badges | Two types of status indicators:  - **"Active"** (Green badge)  - **"Deactivate"** (Red badge) |
| Actions Column | Contains two action buttons for each user:  - **View Button (Eye Icon)**: Blue with a circular border.  - **Delete Button (Trash Icon)**: Red with a circular border. |
| Color Scheme | - **Background:** White  - **Text:** Black or dark gray  - **Buttons:** Blue for search, filter, and new user  - **Status Badges:** Green for active, red for deactivated  - **Action Icons:** Blue for view, red for delete. |

### 3.2 New Account



| **Field Name** | **Description** |
| --- | --- |
| ***New Account*** | |
| Title | A small "X" button at the top-right corner for closing the form. |
| Instruction Text | "Please input information" displayed below the title. |
| Username Field | An empty text input field for entering the username, positioned in the first column. |
| Phone Number Field | An empty text input field for entering the phone number, positioned in the second column. |
| Email Field | An empty text input field for entering the email address, positioned in the first column. |
| Date of Birth Field | An empty text input field with a calendar icon for selecting the date of birth, positioned in the second column. Format: mm/dd/yyyy. |
| Address Field | A full-width text input field for entering the user's address. |
| Role Dropdown | A dropdown menu with "Customer" selected by default, allowing the user to choose their role. |
| Add User Button | Submit button to create a new accountA dark gray button at the bottom with the label "+ Add new user" for submitting the form. |
| Color Scheme | - **Background:** White  - **Text:** Black or dark gray  - **Input Fields:** Light gray border with white background  - **Dropdown:** White with dark gray text  - **Submit Button:** Dark gray with white text. |

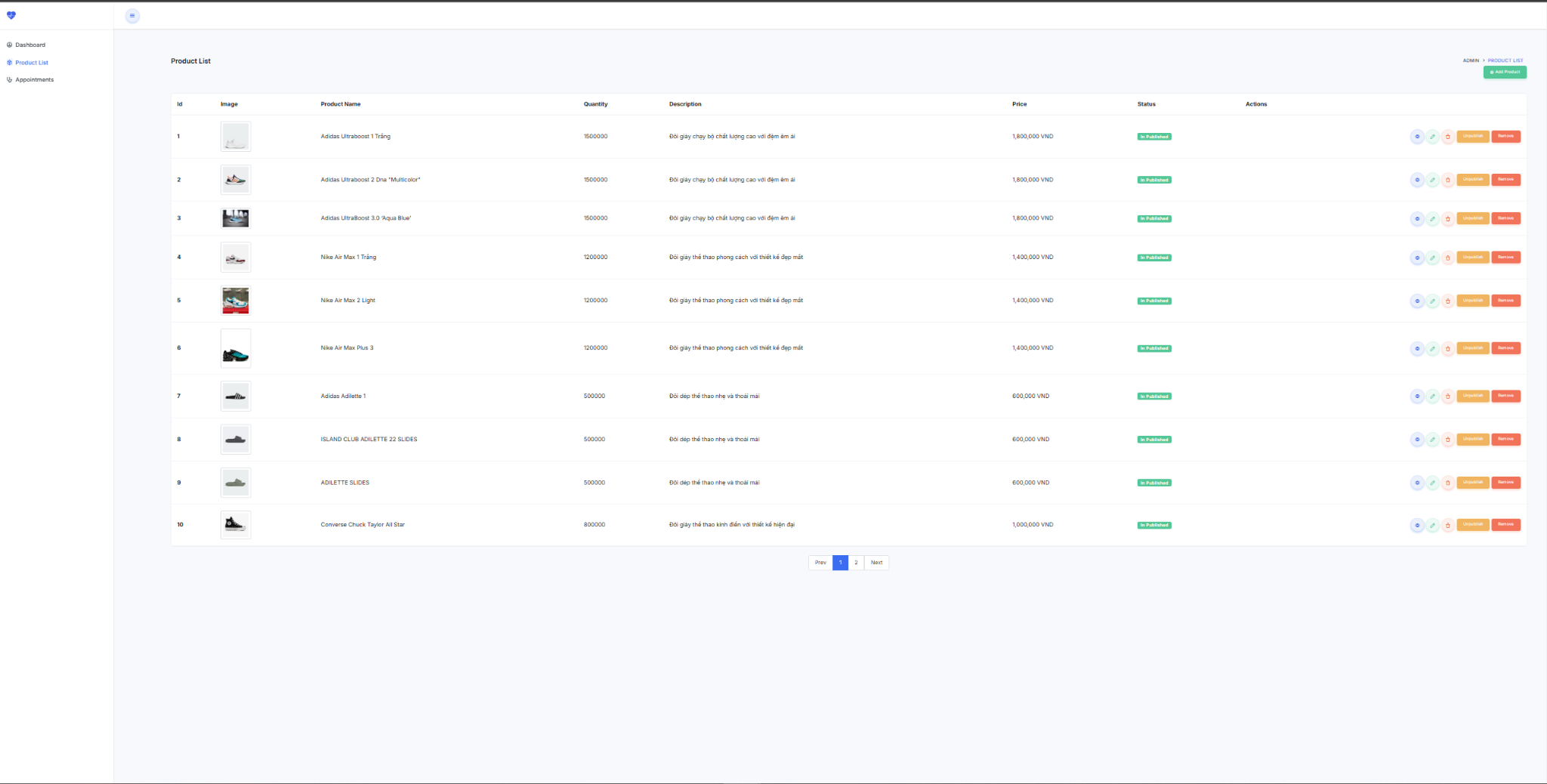
### 3.3 Account Detail



| **Field Name** | **Description** |
| --- | --- |
| ***Account Details*** | |
| Title | "User Details" displayed at the top-left corner. |
| Main Container | A bordered box displaying user details in two columns. |
| Button Actions | - **"Edit" Button**: Located at the bottom left, light green.  - **"Back to List" Button**: Positioned at the top right, dark blue. |
| Left Column | Contains personal information:  - **UserName:** mthangtr182@gmail.com  - **Phone Number:** 0985746254  - **DOB:** 11/6/2024 12:00:00 AM  - **Status:** "Active" (Green badge) |
| Right Column | Contains address and role information:  - **Email:** mthangtr182@gmail.com  - **Address:** Ha Noi  - **Role:** admin |
| Color Scheme | - **Background:** White  - **Text:** Black or dark gray  - **"Edit" Button:** Light green  - **"Back to List" Button:** Dark blue  - **Status Badge:** Green with white text |

## 4. Product Management

### 4.1 Product List



This screen allows administrators to manage products by performing the following actions:

-View Product List: Displays a paginated table containing product details such as ID, name, image, quantity, description, price, and status.

-Change Product Status: Administrators can publish, unpublish, or remove a product from the store.

-View Product Details: Admins can click on a product to view its detailed information.

-Edit Product Information: Admins can modify product details such as name, price, quantity, and description.

-Delete Product: Allows administrators to permanently remove a product from the system.

-Pagination Support: Enables navigation through multiple pages of product records.

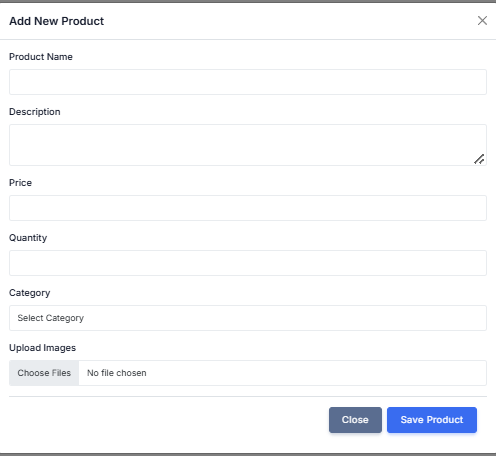
On this screen, administrators can also:

-Filter Products by Category: View products based on their assigned category.

-Quick Access to Product Details and Edit Page: Easily navigate to the edit or details page of a product.

| **Field Name** | **Description** |
| --- | --- |
| Product ID | Data type: integer, unique identifier for each product. |
| Product Image | Data type: URL string, must be a valid image link. Displays product image preview. |
| Product Name | Data type: string, max length of 100 characters. Displays the name of the product. |
| Quantity | Data type: integer, represents the available stock for the product. |
| Description | Data type: string, max length of 500 characters. Provides a short product overview. |
| Price | Data type: decimal, formatted as currency (VND). Displays product price. |
| Status | Data type: integer, represents whether the product is published, unpublished, or removed. |
| View Button | Functionality: Redirects to the Product Details Page for more information. |
| Edit Button | Functionality: Redirects to the Product Edit Page for modifying product details. |
| Delete Button | Functionality: Removes a product from the system after confirmation. |
| Status Action Buttons | Functionality: Allows admins to publish, unpublish, or remove a product. |
| Pagination Controls | Functionality: Allows navigation through multiple pages of products. |

### 4.2 Product Details



This screen allows administrators to perform the following actions:

-Add New Product: Admins can input product details, including name, description, price, quantity, and category.

-Edit Existing Product: Modify existing product details from the database.

-Upload Product Images: Supports multiple image uploads using Cloudinary API.

-Select Product Category: Assign a product to a specific category from the dropdown.

-Save Product Changes: Updates product information and stores it in the database.

-Cancel or Close Modal: Admins can discard changes before saving.

On this screen, administrators can also:

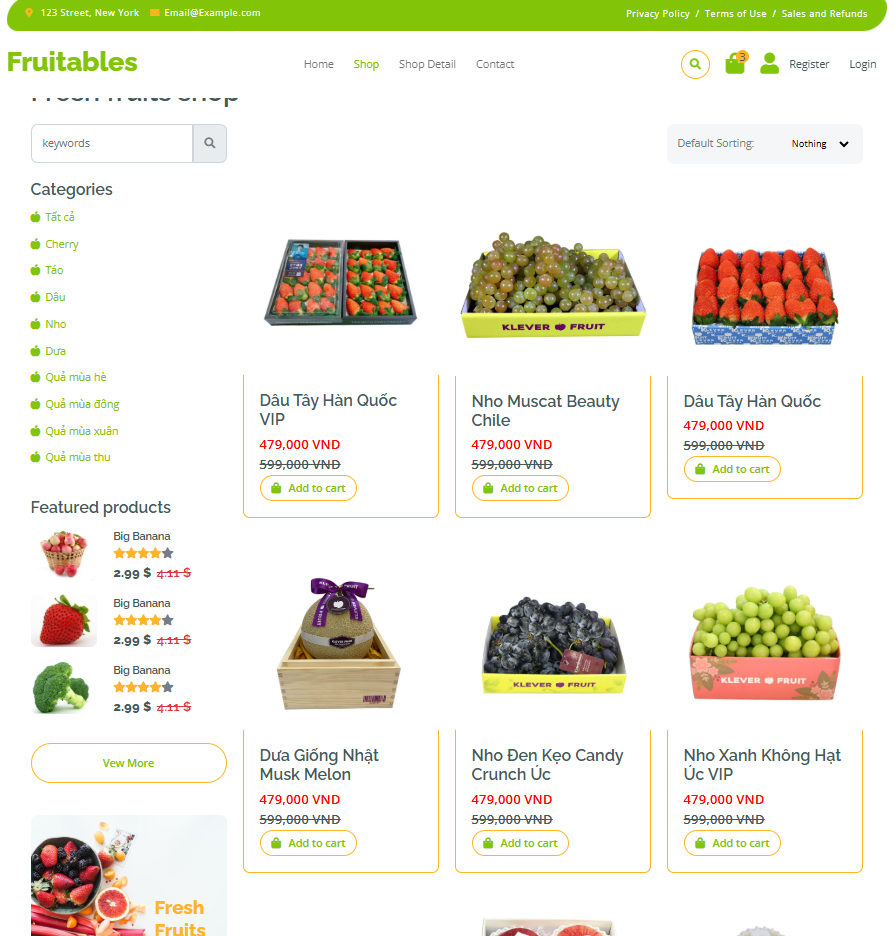
-Validate Input Fields: Ensure product name, price, quantity, and images are correctly entered before submission.

-Integrate Cloudinary API: Upload and store product images in the cloud storage.

| **Field Name** | **Description** |
| --- | --- |
| Product Name | Data type: string, max length of 100 characters. Displays the name of the product. |
| Description | Data type: string, max length of 500 characters. Provides a short product overview. |
| Price | Data type: decimal, formatted as currency (VND). Represents the selling price of the product. |
| Quantity | Data type: integer, represents the available stock for the product. |
| Category Selector | Data type: dropdown list, allows assigning a category to the product. |
| Upload Images | Data type: file input, supports multiple image uploads via Cloudinary API. |
| Save Product Button | Functionality: Saves the product details and updates the database. |
| Close Button | Functionality: Closes the form without saving changes. |

## 5. Public Product

### 5.1 Public Product List



This screen allows users to perform the following actions:

-View Product List: Displays all available products with their images, names, prices, and "Add to Cart" options.

-Filter Products by Category: Users can select a category from the category filter menu to display relevant products.

-Search for Products: Users can browse and refine their selections based on categories.

-Pagination Support: Users can navigate between different pages to browse more products.

-Add Products to Cart: Logged-in users can add selected products to their shopping cart for purchase.

On this screen, users can also:

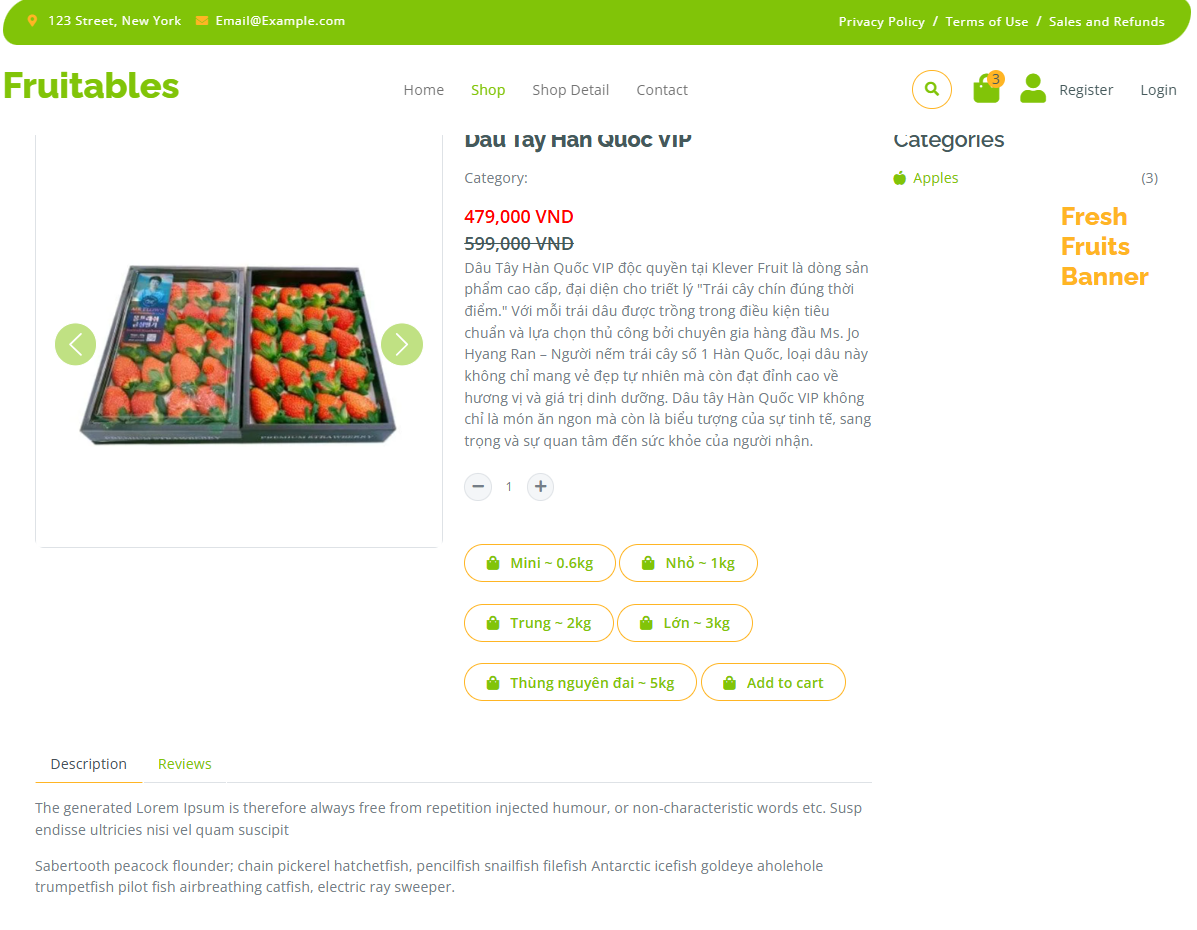
-Click on a Product: Redirect to the Product Details Page for more information.

-Modify Cart Contents: Users can add multiple products and update quantities in the cart.

-Admins can Manage Products: Admins have access to add, edit, or delete products from the system.

| **Field Name** | **Description** |
| --- | --- |
| Product Name | Data type: string, max length of 100 characters. Displays the name of the product. |
| Product Image | Data type: URL string, must be a valid image link. Shows product image preview. |
| Price | Data type: decimal, formatted as currency (VND). Displays product price. |
| Category Filter | Data type: dropdown list, allows filtering products based on their category. |
| Add to Cart Button | Functionality: Adds product to the cart (only available for logged-in users). |
| Pagination Controls | Functionality: Allows navigation through multiple pages of products. |
| Navigation Menu | Functionality: Allows users to navigate between Home, Shop, Categories, Contact, Register, and Login pages. |
| Shopping Cart | Functionality: Displays the number of items in the cart and allows checkout. |

### 5.2 Public Product Details



This screen allows users to perform the following actions:

-View Detailed Product Information: Displays product name, image, price, description, and available quantity.

-Select Product Quantity: Users can choose the number of items they wish to purchase, within the available stock limits.

-Add Product to Cart: Logged-in users can add the selected product to their shopping cart.

-Share Product on Social Media: Users can share product details via Facebook, Twitter, Google Plus, and LinkedIn.

-View Related Products: Displays similar products that may interest the user.

On this screen, users can also:

-Navigate Back to Product List: Users can browse more products.

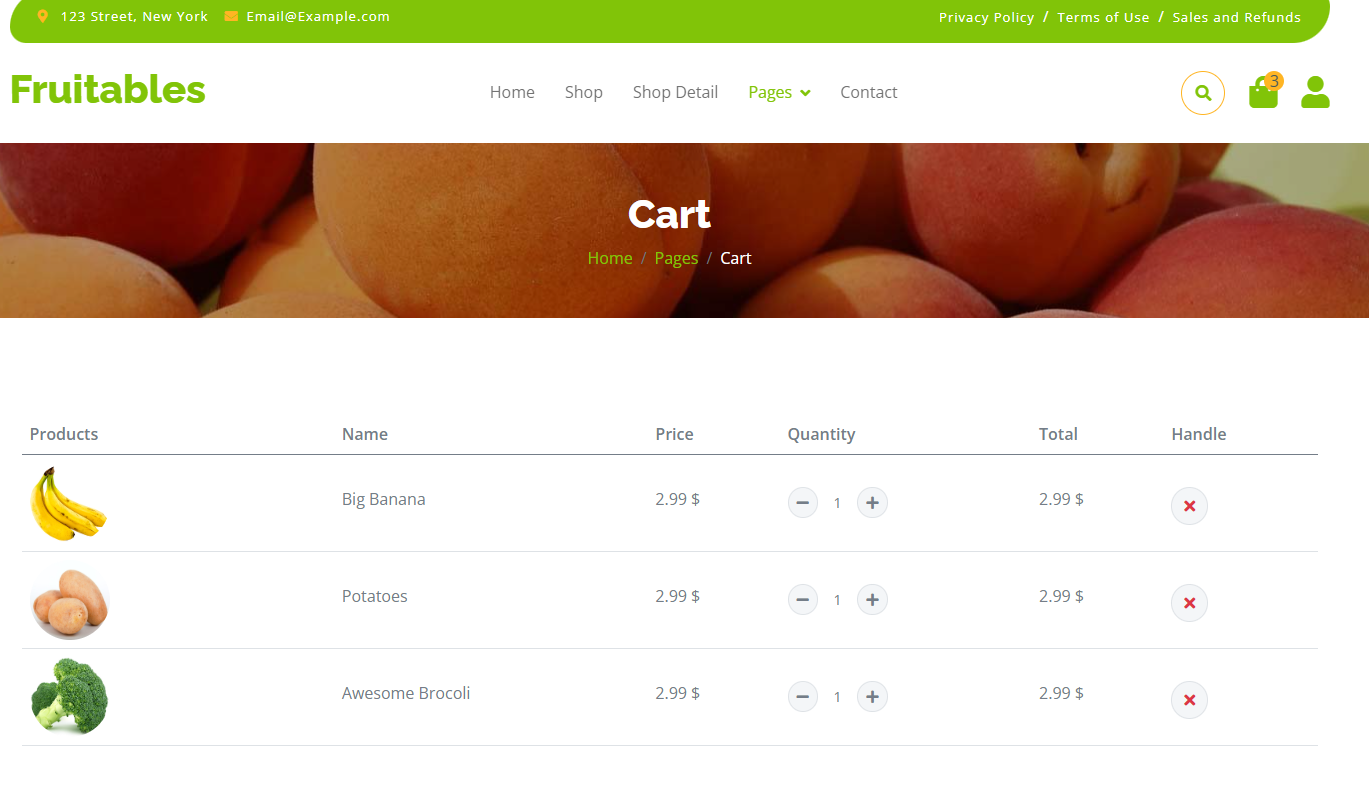
-Modify Cart Contents: Users can update quantities or remove items after adding them to the cart.

-Admins can Edit Product Details: Admins have access to modify product information such as name, price, description, and availability.

| **Field Name** | **Description** |
| --- | --- |
| Product Name | Data type: string, max length of 100 characters. Displays the name of the product. |
| Product Image | Data type: URL string, must be a valid image link. Shows product image preview. |
| Price | Data type: decimal, formatted as currency (VND). Displays product price. |
| Description | Data type: string, max length of 500 characters. Provides a short product overview. |
| Available Quantity | Data type: integer, represents the stock availability of the product. |
| Quantity Selector | Data type: number input, allows users to select how many items to add to the cart. |
| Add to Cart Button | Functionality: Adds selected quantity to the cart (only available for logged-in users). |
| Product Category | Data type: string, displays the category to which the product belongs. |
| Social Media Share Buttons | Functionality: Users can share product details on Facebook, Twitter, Google Plus, and LinkedIn. |
| Related Products Section | Functionality: Displays products that are similar to the viewed product. |

## 6. Order Placing

### 6.1 Shopping Cart



This screen allows customers or guest to manage their shopping cart by performing the following actions:

* View Cart Items: Displays a table containing product details such as name, unit price, quantity, and subtotal.
* Update Item Quantity: Customers can modify the quantity of each product in their cart.
* Remove Items: Allows customers to delete unwanted products from their cart.
* Select Items for Checkout: Customers can choose which products to include in their order.
* View Total Amount: Displays the cumulative cost of all items in the cart.
* Place Order: Enables customers to proceed to checkout with their selected items.

On this screen, customers can also:

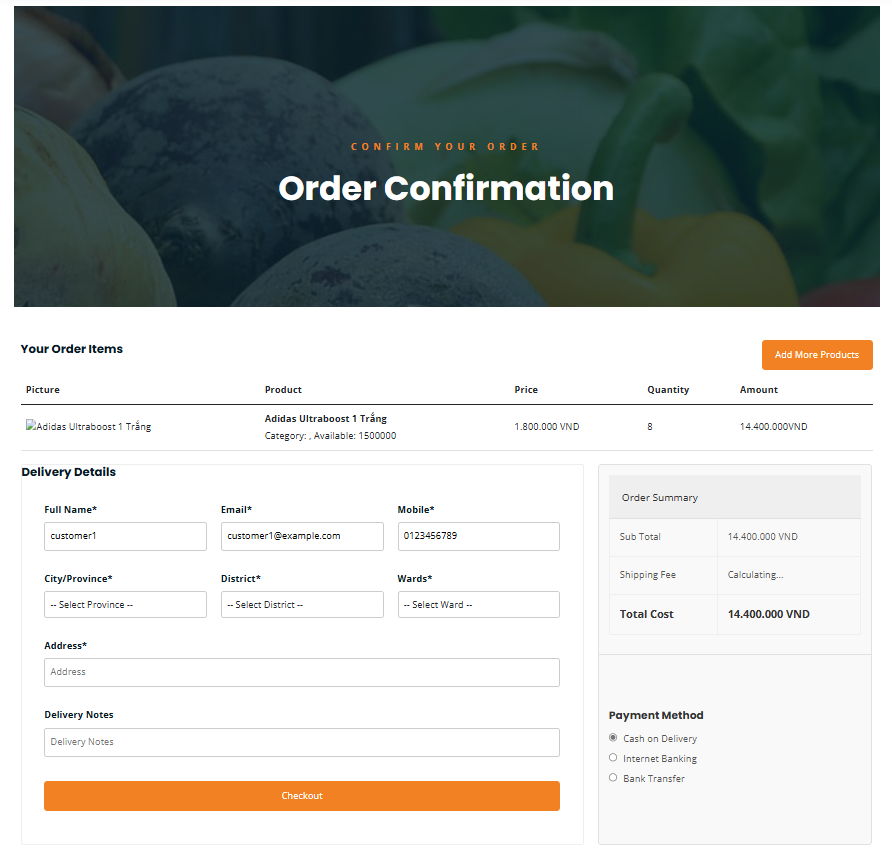
* Manage Multiple Items: Add and manage various products in a single cart.
* View Product Calculations: See automatic price calculations based on quantity changes.
* Partial Checkout: Select specific items to include in the final order.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Chọn | Data type: boolean checkbox, allows selection of individual products for purchase. |
| Sản Phẩm | Data type: string, displays the name and model of the footwear product. |
| Giá | Data type: decimal, formatted as currency (VND). Shows the unit price of each product. |
| Số lượng | Data type: integer input field, allows customers to specify the quantity of each product. |
| Tổng | Data type: decimal, formatted as currency (VND). Displays the subtotal for each product (unit price × quantity). |
| Delete Button | Functionality: Removes a product from the cart after clicking the red trash icon. |
| Số lượng | Data type: integer input field, allows customers to specify the quantity of each product. The system automatically saves the updated quantity when customers change the value, instantly reflecting the new subtotal and cart total. |
| Tổng tiền | Data type: decimal, formatted as currency (VND). Shows the cumulative total of all items in the cart (16,600,000 VND). |
| Đặt hàng Button | Functionality: Proceeds to the checkout process with selected items. |

### 6.2 Order Confirmation

### 



This screen allows customers or guests to confirm their order by performing the following actions:

* Review Order Items: Displays a detailed table of selected products with comprehensive information.
* Modify Quantity: Customers can adjust the number of items before final purchase.
* Add More Products: Option to continue shopping and include additional items.
* Enter Delivery Information: Provide complete shipping and contact details.
* Select Delivery Location: Choose precise delivery address through province, district, and ward selection.
* Add Delivery Notes: Include special instructions for delivery.
* Select Payment Method: Choose preferred payment option.

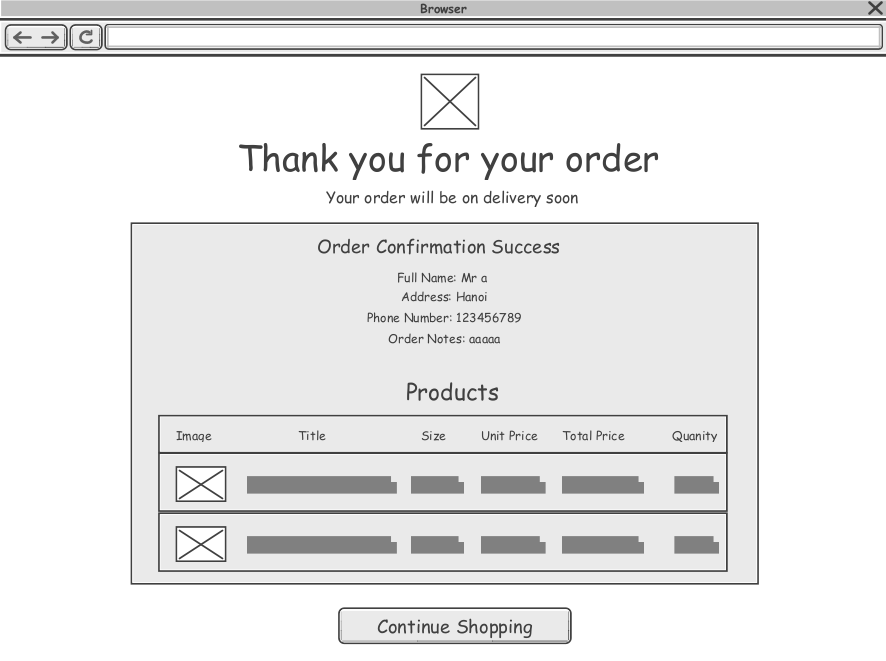
On the screen, they can also:

* Review Product Details: see the name, price, quantity, and total amount for each product.
* Check Availability: view product stock information.
* View Shipping Costs: see the delivery fee added to their order.
* Choose Delivery Options: enter their complete delivery address.
* Select Payment Options: choose between Cash on Delivery, Internet Banking, or Bank Transfer.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Picture | Data type: image URL, displays product thumbnail for visual reference. |
| Product | Data type: string, shows product name, category, and available stock information. |
| Price | Data type: decimal, formatted as currency (VND). Displays unit price of each product. |
| Quantity | Data type: integer input field, allows customers to modify the number of items. System automatically updates subtotal when quantity changes. |
| Amount | Data type: decimal, formatted as currency (VND). Calculates total price for specific product line (unit price × quantity). |
| Full Name\* | Data type: string, required field for customer's complete name. |
| Email\* | Data type: string, required field for customer's contact email. |
| Mobile\* | Data type: phone number, required field for customer contact. |
| City/Province\* | Data type: dropdown selection, allows choosing delivery location. |
| District\* | Data type: dropdown selection, specifies detailed district for delivery. |
| Wards\* | Data type: dropdown selection, provides most precise delivery location. |
| Address\* | Data type: string, detailed street address for delivery. |
| Delivery Notes | Data type: string, optional field for additional delivery instructions. |
| Sub Total | Data type: decimal, formatted as currency (VND). Shows total cost of products before shipping. |
| Shipping Fee | Data type: decimal, calculated shipping cost based on delivery details. |
| Total Cost | Data type: decimal, formatted as currency (VND). Final total including products and shipping. |
| Payment Method | Data type: radio button selection, allows choosing from available payment options. |
| Add More Products Button | Functionality: Allows customers to return to product selection and add more items. |
| Checkout Button | Functionality: Completes the order process and proceeds to payment. |

### 6.3 Order Checkout

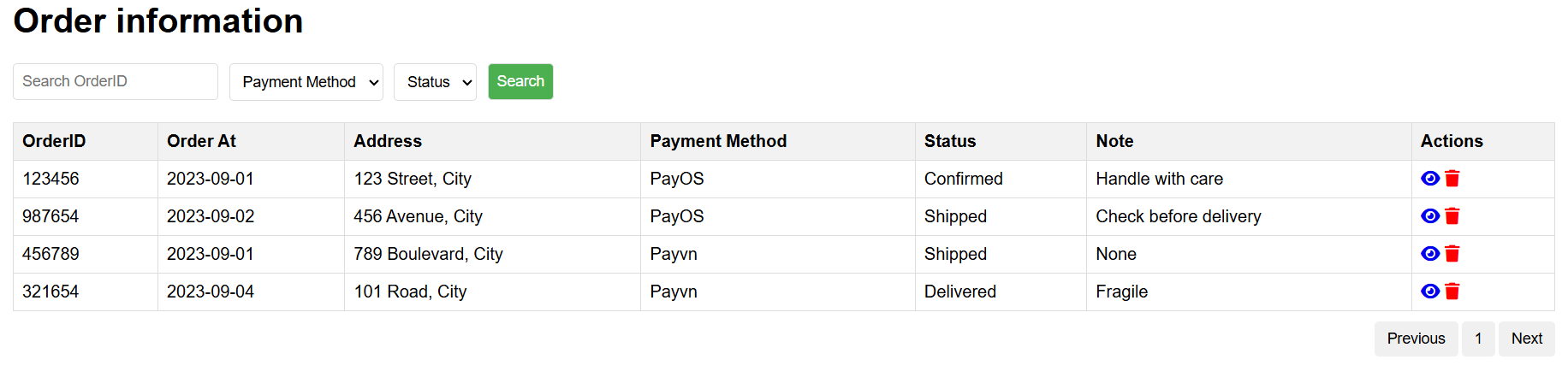


* This is the screen after the customer/guest has completed selecting the payment method and proceeded to pay. This page will display information about the order that the customer/guest has ordered.

| **Field Name** | **Description** |
| --- | --- |
| Thank You Message | Displays "Thank you for your order" with a confirmation note. |
| Full Name | Displays the customer’s full name. |
| Phone number | Displays the customer’s contact number. |
| Address | Displays the shipping address. |
| Order Notes | Shows any special notes the customer provided. |
| Products Section | A table listing purchased products with detail information |
| Continue Shopping Button | A button allowing users to return to shopping. |

## 7. Order Tracking

### 7.1 Order Tracking



This screen allows the Customer/Sales to:

* View order information: View list of current order
* Filter order list: Filter orders by pay method, status
* Search order: enter keywords to search orders by OrderID

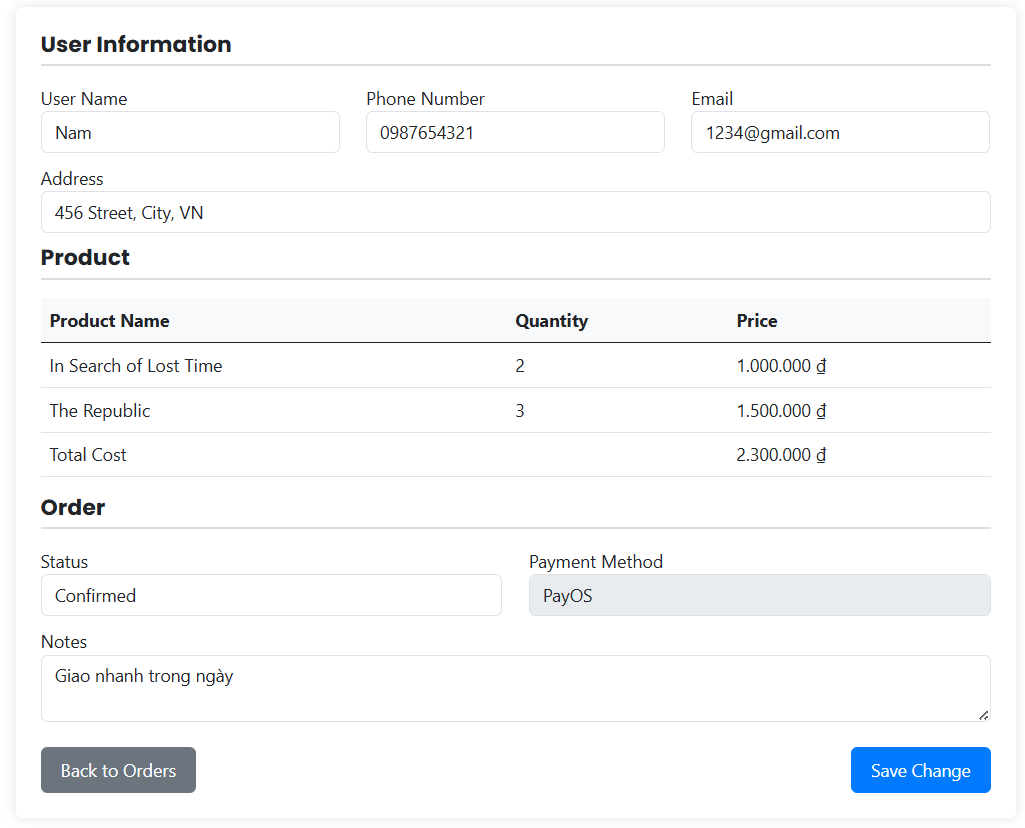
On the screen, users can also

* View order details: Click the view icon in the list to view order details or edit that order information.
* Delete order: Delete an order by clicking the Delete icon corresponding to that order.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| Search OrderID | Initial values: All the orders at with null or blank type  Enter keyword to display product name |
| Payment MeThod | Initial values: All the orders at with null or blank type  Hover the mouse to show the field name: “Payment MeThod” |
| Status | Initial values: All the orders at with null or blank type  Hover the mouse to show the field name: “Status” |
| Search button | Search orders by product name in order, Order At and order status |
| OrderID | Column display OrderID of the order |
| Order At | Column describes the order placement time |
| Address | Column showing the location of the order delivered to |
| Payment MeThod | Column describing the method to pay for the order |
| Status | Column showing order shipping status |
| Note | Column displays notes related to the order |
| View button | View the order details corresponding to it |
| Delete button | Delete the order corresponding to it |

### 7.2 Order Details



This screen allows the Customer/Sales to:

* View Order Details: View user information, products in the order, status, payment method, and order notes
* Update user information: Change user name, phone number, address, email of the placed order
* Select Status: Assign order status to specific order from drop down list.
* Update Notes: Change notes of submitted order
* Save Order Changes: Updates order information and stores it in the database.
* Cancel or Close Modal: Customer/Sales can discard changes before saving.

On this screen, users can also

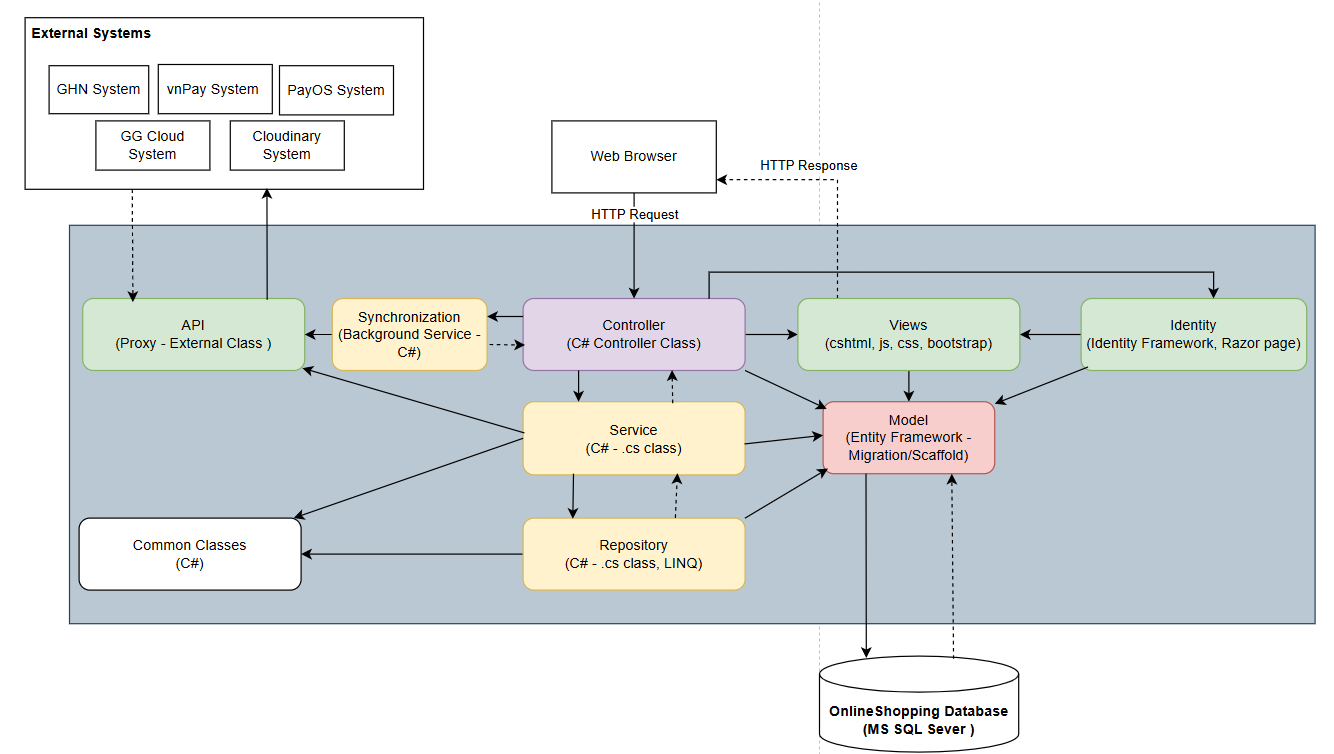
* Validate Input Fields: Ensure user name, phone number, address, notes are correctly entered before submission.

**Field Description**

| **Field Name** | **Description** |
| --- | --- |
| User name | Data type: string, max length of 100 characters. Displays the recipient name of the order. |
| Phone Number | Data type: integer, represents the user's phone number. |
| Address | Data type: string, max length of 500 characters. Provide address for delivery. |
| Email | Data type: string, max length of 20 characters. Provide email of user. |
| Status Selector | Data type: dropdown list, allows assigning status to the order. |
| Notes | Data type: string, max length of 500 characters. Show notes related to order. |
| Close Button | Functionality: Closes the form without saving changes. |
| Save Change Button | Functionality: Saves the order details and updates the database. |

# III. High-Level Design

## 1. Software Architecture

**

### 1.1 Sub-Systems/Components

#### 1.1.1 Identity

* This is an identity framework supported by .NET that helps verify and authenticate users, roles,...etc.

#### 1.1.2 Views

* This is the part responsible for displaying the interface for users to interact with the system.

#### 1.1.3 Controller

* This is the part that coordinates other classes to perform an action or logic sequence requested by the user.

#### 1.1.4 Service

* This is the part responsible for calling from the Repository, API, and Common Classes layers to coordinate the execution of business logic and pass it to the Controller.

#### 1.1.5 Repository

* This is the part that directly retrieves data called from DBContext to Service.

#### 1.1.6 Model

* This is the part that maps and stores data directly with the Database, including the part that connects directly to the Database using DBContext.

#### 1.1.7 Common Classes

* This section contains common methods for the Repository and Service classes to use.

#### 1.1.8 Synchronization

* This is the timer part that uses the Worker Service inherited from BackGround Service of .NET to schedule a fixed time to send data update signals to API.

#### 1.1.9 API

* This is the proxy layer responsible for interacting and connecting directly to external systems.

#### 1.1.10 Online Shopping Database

* This is the database part that contains the system's relational tables and data.

### 1.2 External Interfaces

#### 1.2.1 GHN System

* This is a system that returns data from provinces and cities in the country, receives shipping data and returns shipping data to OSS.

#### 1.2.2 vnPay System

* This is a system that supports online payment using many different methods such as QR code, domestic account or international account.

#### 1.2.3 PayOS System

* This is a system that supports online payment using QR code.

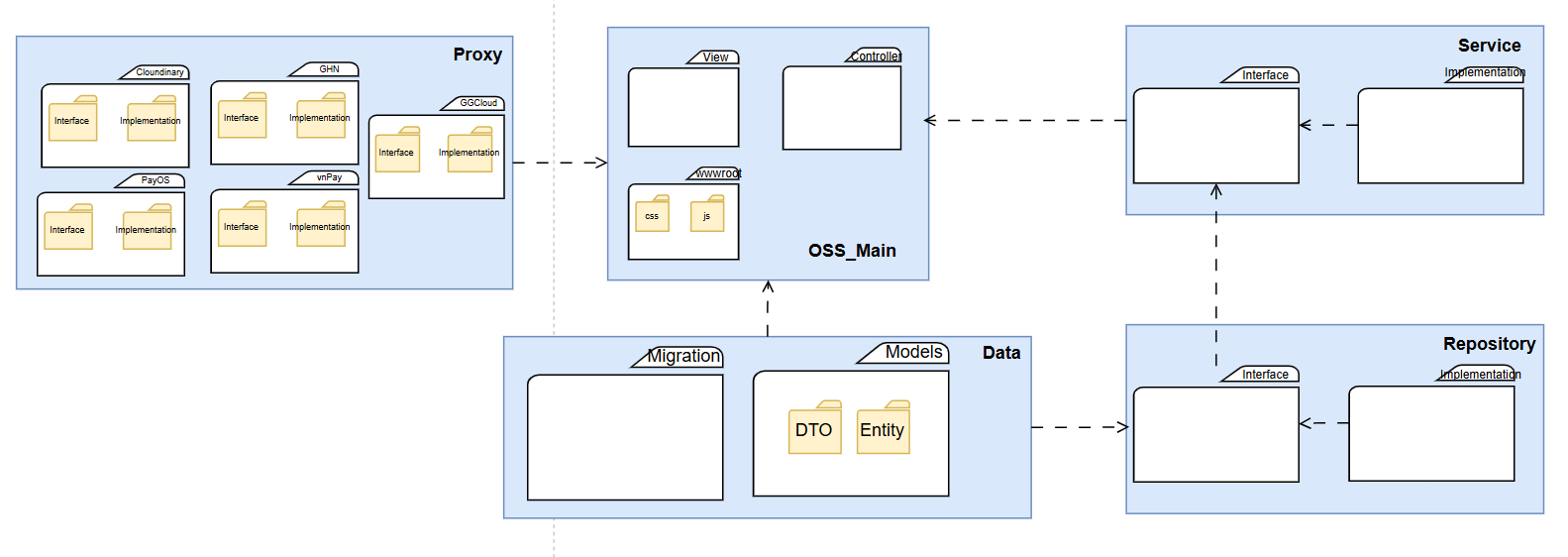
#### 1.2.4 Cloudinary System

* This is an image storage system for OSS, when uploading images to Cloudinary and the system returns OSS a URL to save to the database.

#### 1.2.5 GG Cloud System

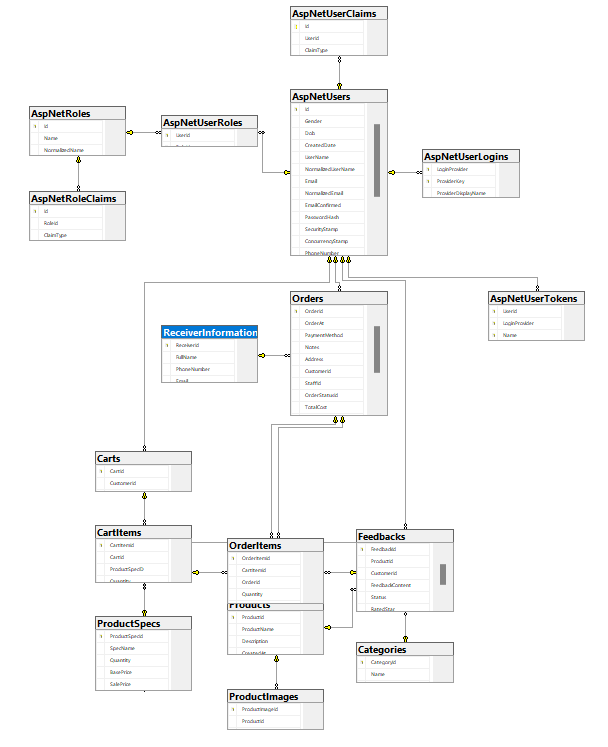
* This is a system that supports quick login authentication and sends emails to users.

## 2. Package Diagram



| **No** | **Package** | **Description** |
| --- | --- | --- |
| *01* | *Controllers* | *Contains all controllers that handle HTTP requests and responses.*  *Serves as the primary entry point for user interactions.*  *Delegates requests to the Service Layer for business logic processing.* |
| *02* | *Services* | *Includes business logic components that process requests and communicate with the Repository layer.*  *Ensures data integrity and enforces application-specific rules.*  *Acts as an intermediary between Controllers and Repositories.* |
| *03* | *Repositories* | *Responsible for data access and manipulation.*  *Interacts directly with the database using Entity Framework or other ORM tools.*  *Provides structured data queries to be used in the Service Layer.* |
| *04* | *Models* | *Defines data structures and Entity Classes that represent database tables.*  *Used across Repositories, Services, and Controllers.*  *Serves as the foundation for data representation and validation.* |
| *05* | *API* | *Manages integration with external APIs.*  *Handles requests to third-party services, such as payment gateways, external authentication providers, or data services.* |
| *06* | *Views* | *Contains Razor view (.cshtml) files responsible for rendering UI components.*  *Displays data received from Controllers and Services.*  *Supports front-end user interactions with dynamic content.* |
| *07* | *Identity* | *Manages authentication and user-related data.*  *Provides user login, logout, authorization, and role-based access control.*  *Handles Database Context for managing user sessions and security configurations.* |
| *08* | *AuthController* | *Manages authentication (login, logout, password reset, and token handling).* |
| *09* | *UserController* | *Handles user profile management (update profile, change password, and user details retrieval).* |
| *10* | *RoleController* | *Manages user roles and permissions (assigning and updating roles).* |
| *11* | *ProductController* | *Handles CRUD operations for products (adding, updating, deleting, and retrieving products).* |
| *12* | *CategoryController* | *Manages product categories (adding, updating, and deleting categories).* |
| *13* | *CartController* | *Manages shopping cart operations (adding/removing items, updating quantities).* |
| *14* | *OrderController* | *Handles order management (placing orders, updating order status, retrieving order history).* |
| *15* | *SettingController* | *Manages system-wide settings and configurations.* |

## 3. Database Design



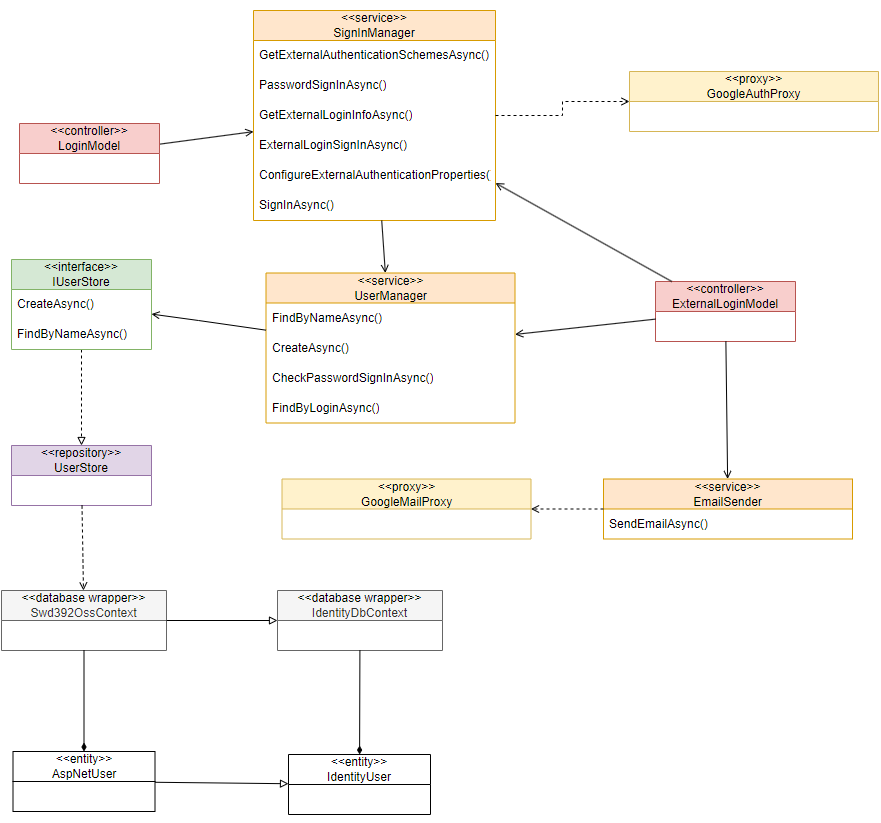
***Table Descriptions***

| **No** | **Table** | **Description** |
| --- | --- | --- |
| 01 | AspNetRoleClaims | * Stores the claims that are assigned to ASP.NET Identity roles. * Primary keys: Id (auto-incremented identity) * Foreign keys: RoleId (references AspNetRoles.Id) |
| 02 | AspNetRoles | * Stores role information for ASP.NET Identity. * Primary keys: Id (unique role identifier) * Foreign keys: None |
| 03 | AspNetUserClaims | * Stores claims assigned to users in ASP.NET Identity. * Primary keys: Id (auto-incremented identity) * Foreign keys: UserId (references AspNetUsers.Id) |
| 04 | AspNetUserLogins | * Tracks external login providers for users. * Primary keys: LoginProvider, ProviderKey (composite key of login provider and key) * Foreign keys: UserId (references AspNetUsers.Id) |
| 05 | AspNetUserRoles | * Maps users to roles in a many-to-many relationship * Primary keys: UserId, RoleId (composite key of user and role identifiers) * Foreign keys: UserId (references AspNetUsers.Id), RoleId (references AspNetRoles.Id) |
| 06 | AspNetUsers | * Stores user account information for ASP.NET Identity. * Primary keys: Id (unique user identifier) * Foreign keys: None |
| 07 | AspNetUserTokens | * Stores authentication tokens for users. * Primary keys: UserId, LoginProvider, Name (composite key) * Foreign keys: UserId (references AspNetUsers.Id) |
| 08 | Cart | * Stores shopping cart information for customers. * Primary keys: CartId (unique cart identifier) * Foreign keys: CustomerId (references AspNetUsers.Id) |
| 09 | CartItem | * Stores individual items in a customer's shopping cart. * Primary keys: CartItemId (unique cart item identifier) * Foreign keys: CartId (references Cart.CartId), ProductId (references Product.ProductId) |
| 10 | Category | * Stores product categories. * Primary keys: CategoryId (unique category identifier) * Foreign keys: None |
| 11 | Order | * Stores customer order information. * Primary keys: OrderId (unique order identifier) * Foreign keys: CustomerId (references AspNetUsers.Id), StaffId (references AspNetUsers.Id), OrderStatusId (references OrderStatus.StatusId) |
| 12 | OrderItem | * Stores individual items within an order. * Primary keys: OrderItemId (unique order item identifier) * Foreign keys: OrderId (references Order.OrderId), ProductId (references Product.ProductId) |
| 13 | OrderStatus | * Defines possible statuses for orders. * Primary keys: StatusId (unique status identifier) * Foreign keys: None |
| 14 | Product | * Stores product information. * Primary keys: ProductId (unique product identifier) * Foreign keys: CategoryId (references Category.CategoryId), ProductStatusId (references ProductStatus.ProductStatusId) |
| 15 | ProductImage | * Stores image URLs for products. * Primary keys: ProductImageId (unique image identifier * Foreign keys: ProductId (references Product.ProductId) |
| 16 | ProductStatus | * Defines possible statuses for products. * Primary keys: ProductStatusId (unique status identifier * Foreign keys: None |

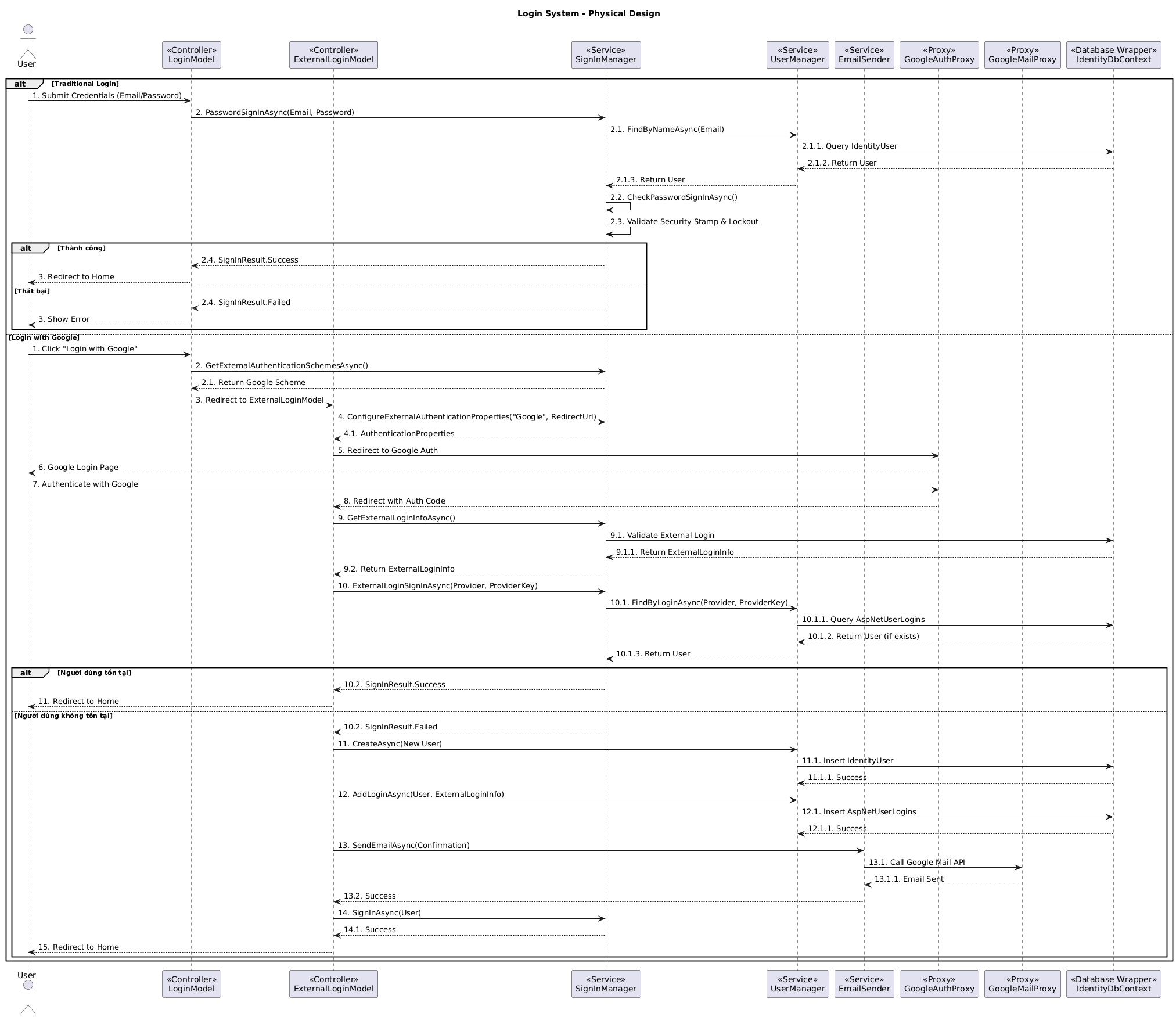
# IV. Detailed Design

## 1. Login System - (Customer)

### 1.1 Class Diagram



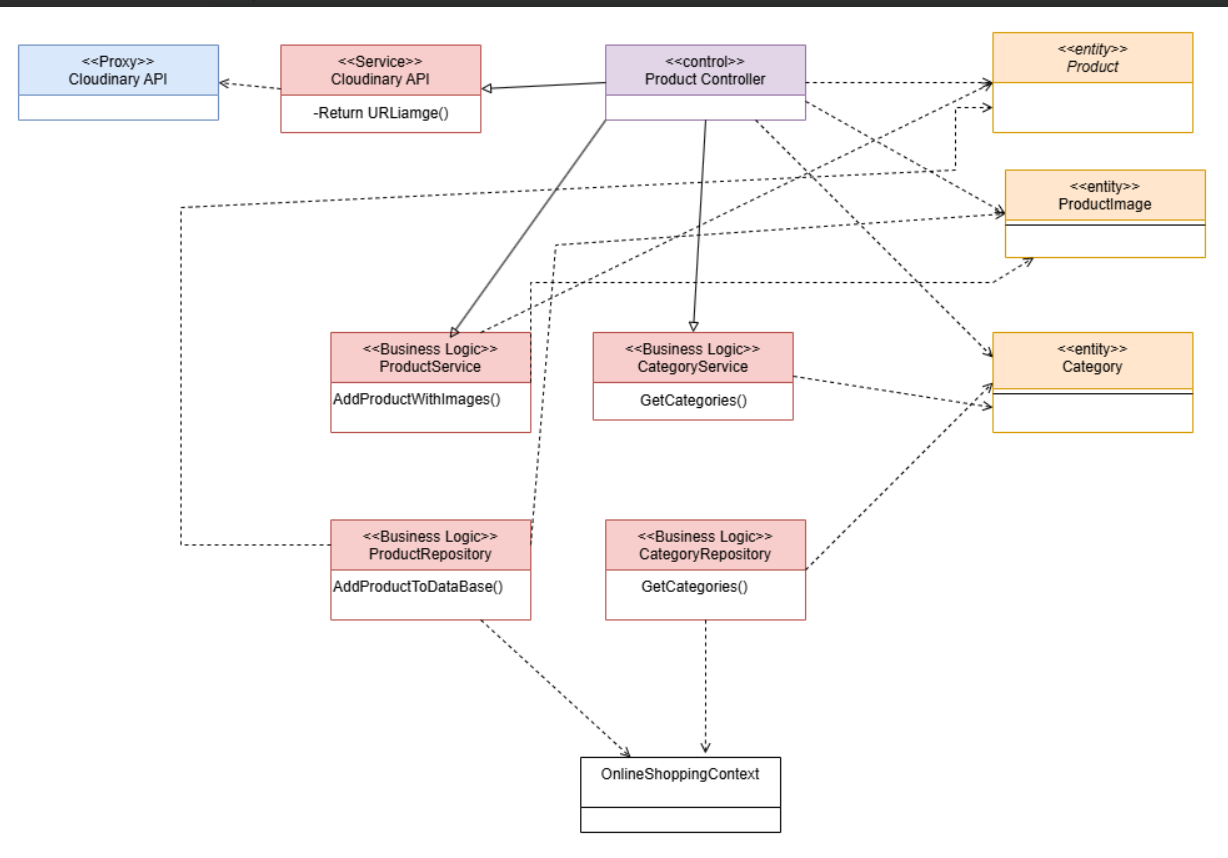
### 1.2 Sequence Diagram



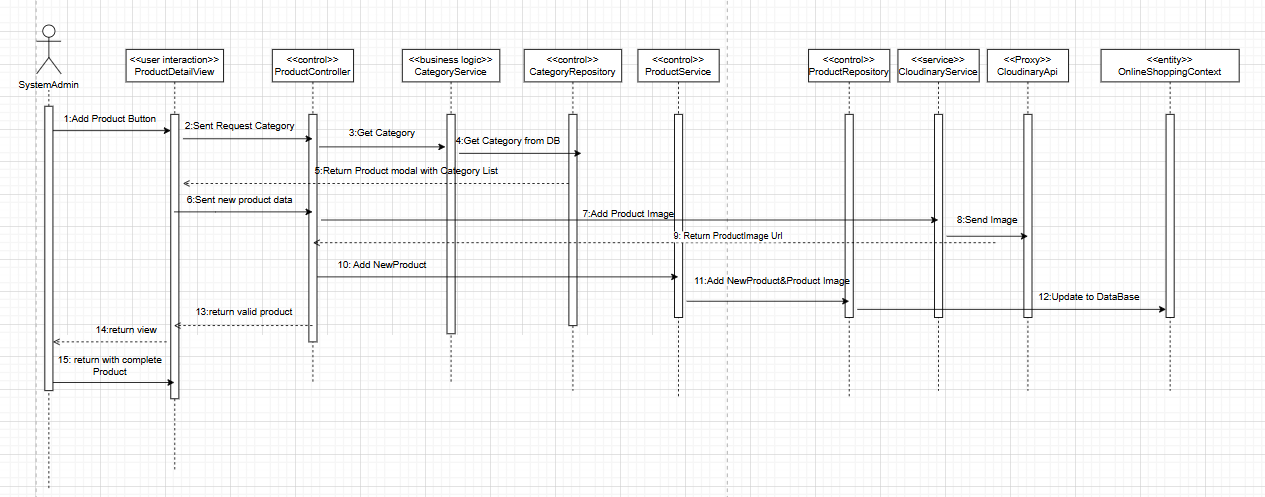
### 1.3 Google Login Sequence Diagram

## 2. Add New Product - (System Admin)

### 2.1 Class Diagram

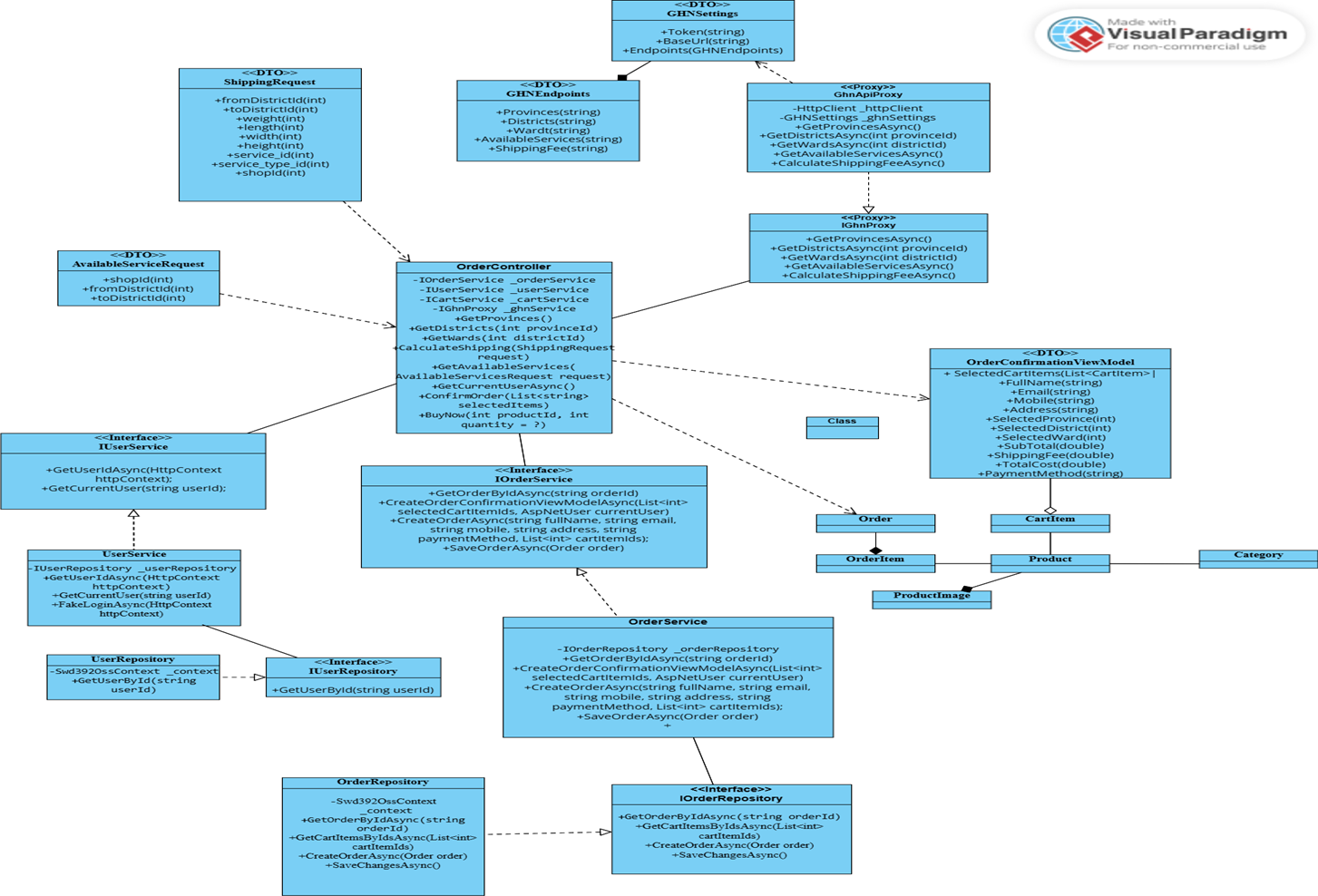


### 2.2 Add Product Sequence Diagram



## 3. Confirm Order

### 3.1 Class Diagram

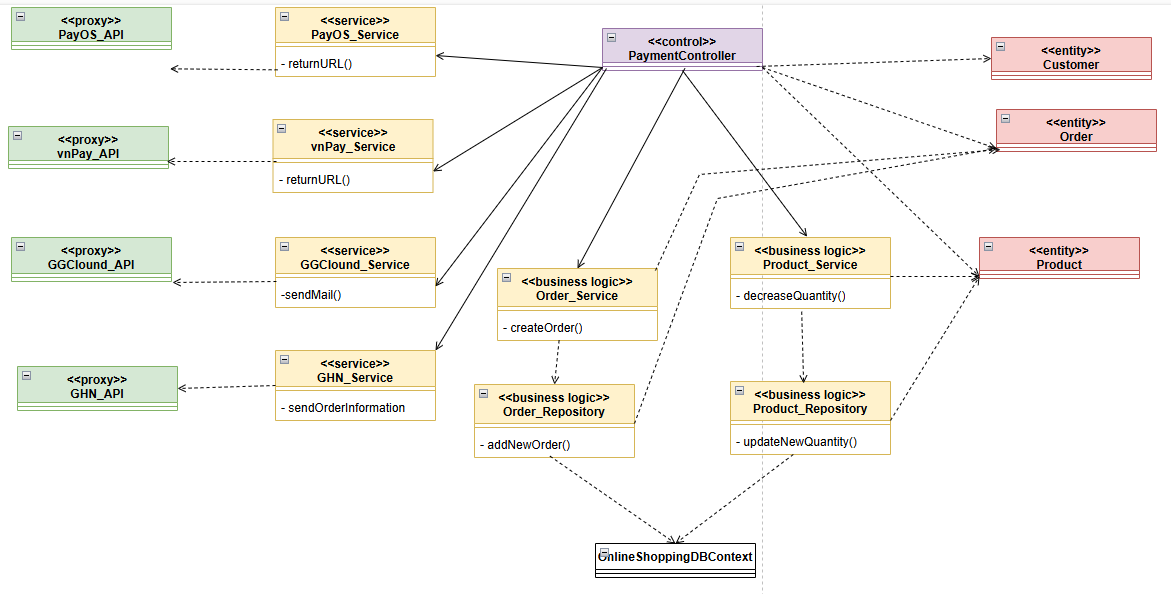


### 3.2 Sequence Diagram Confirm Order

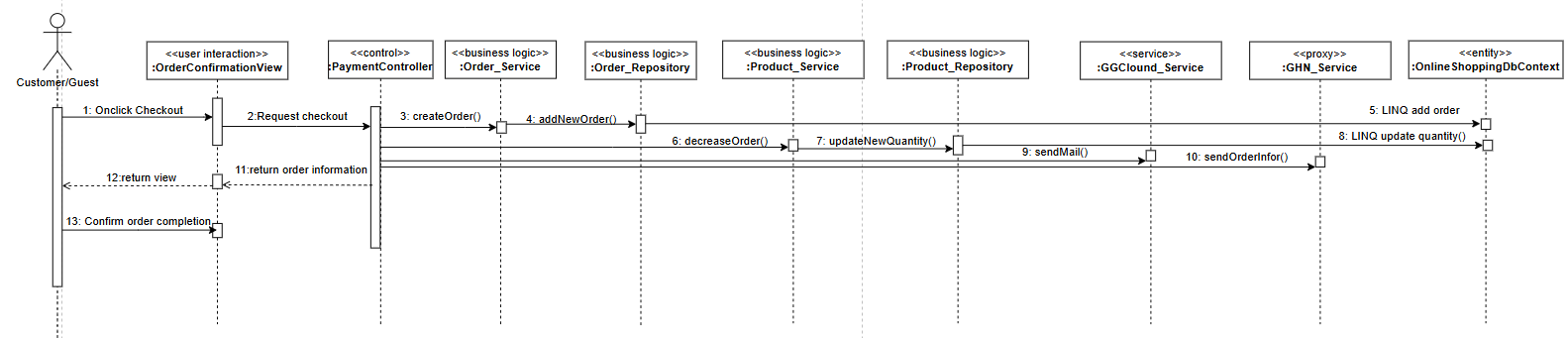


## 4. Checkout Order

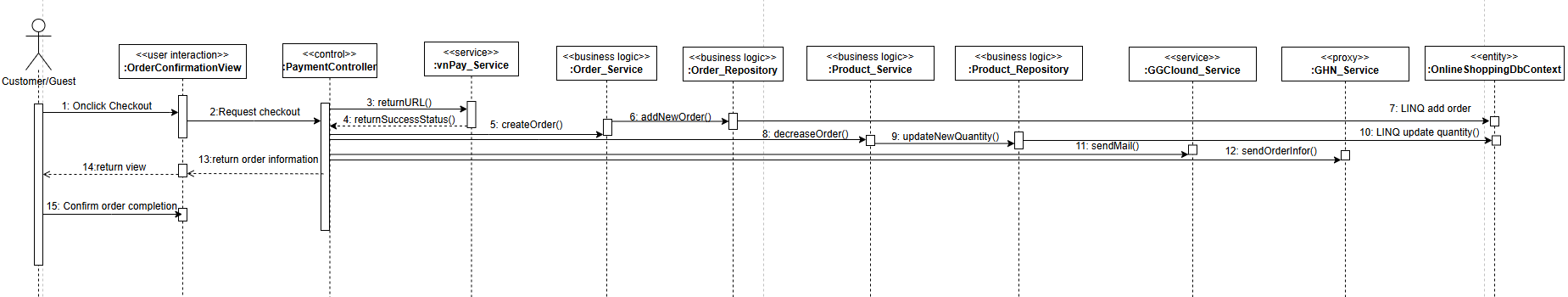
### 4.1 Class Diagram



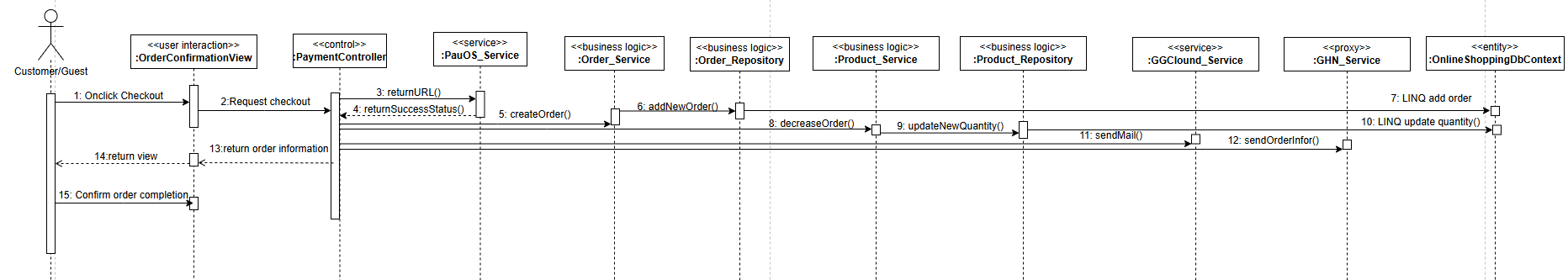
### 4.2 Sequence Diagram COD Method



### 4.3 Sequence Diagram vnPay Method

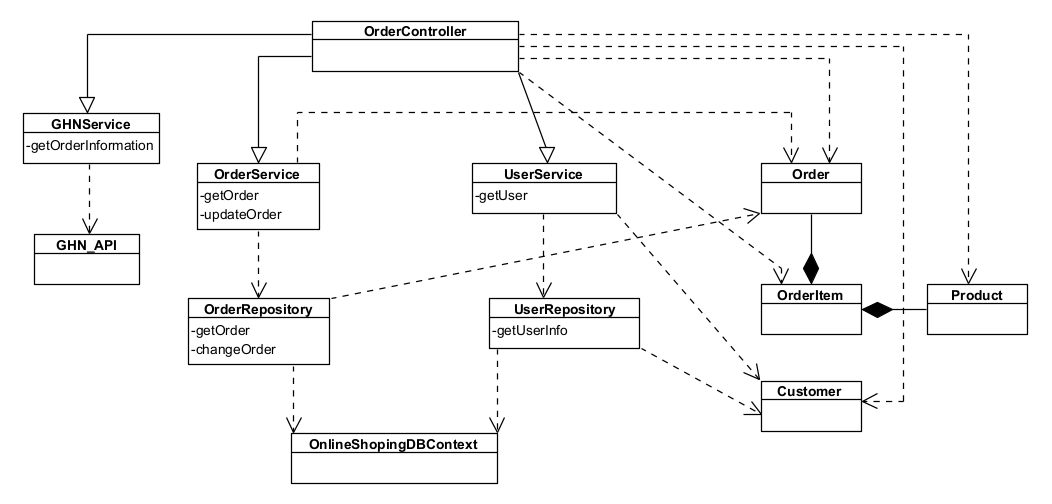


### 4.4 Sequence Diagram PayOS Method

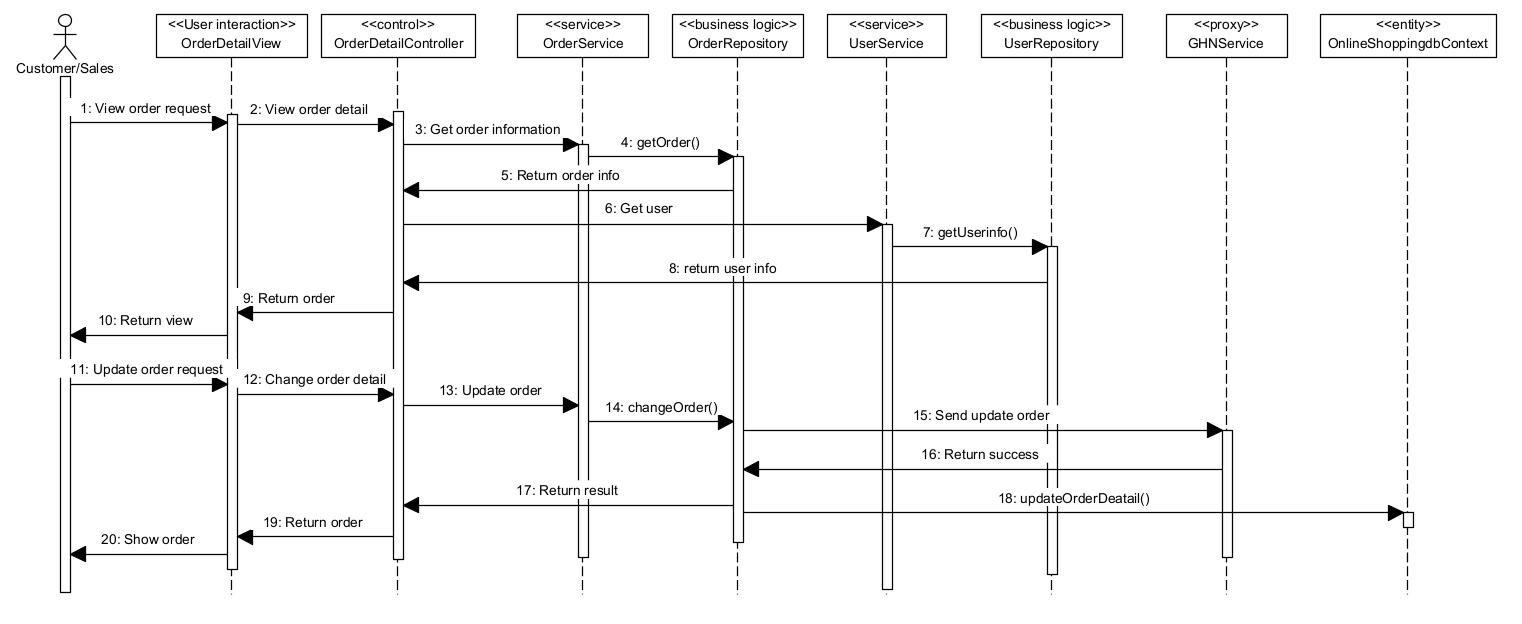


## 5. Track Order Details

### 5.1 Class Diagram



### 5.2 Sequence Diagram Track Order Details

******