

**Analysis Modelling Document**

-Truong Nguyen Viet Quang-

-HE182422-

– Hanoi, Sep 2024 –

**Table of Contents**

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

[II. Static Modeling 4](#_heading=h.30j0zll)

[1. Context Modeling 4](#_heading=h.1fob9te)

[2. Entity Classes 4](#_heading=h.3znysh7)

[III. Object & Class Structuring 6](#_heading=h.2et92p0)

[1. Validate PIN 6](#_heading=h.tyjcwt)

[2. User Case Name2 7](#_heading=h.1t3h5sf)

[IV. Dynamic Interaction Modeling 7](#_heading=h.4d34og8)

[1. Validate PIN 7](#_heading=h.2s8eyo1)

[2. User Case Name2 8](#_heading=h.17dp8vu)

[V. Finite State Machine 8](#_heading=h.3rdcrjn)

[1. PIN Validation 8](#_heading=h.lnxbz9)

[2. State Chart Name2 9](#_heading=h.35nkun2)

[3. State Chart Name3 9](#_heading=h.1ksv4uv)

# I. Record of Changes

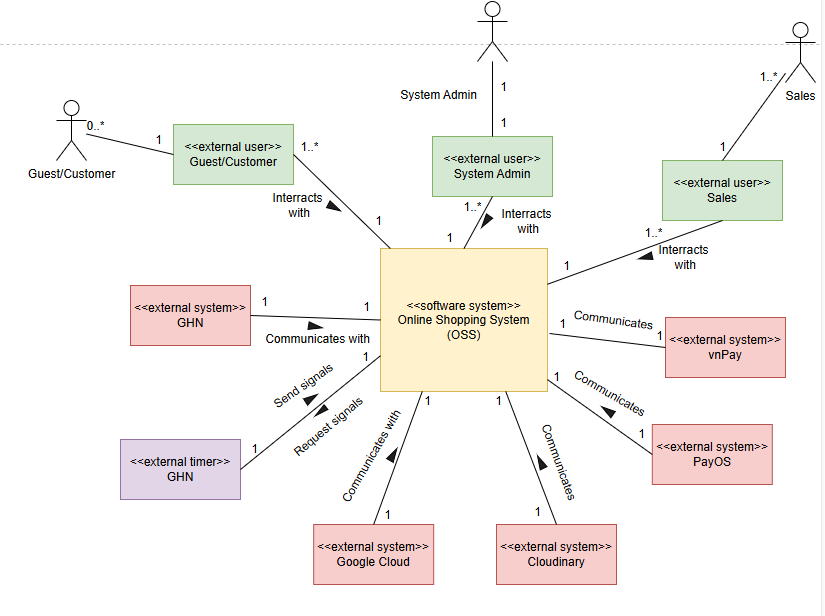
| **Date** | **A\* M, D** | **In charge** | **Change Description** |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

\*A - Added M - Modified D - Deleted

# II. Static Modeling

## 1. Context Modeling

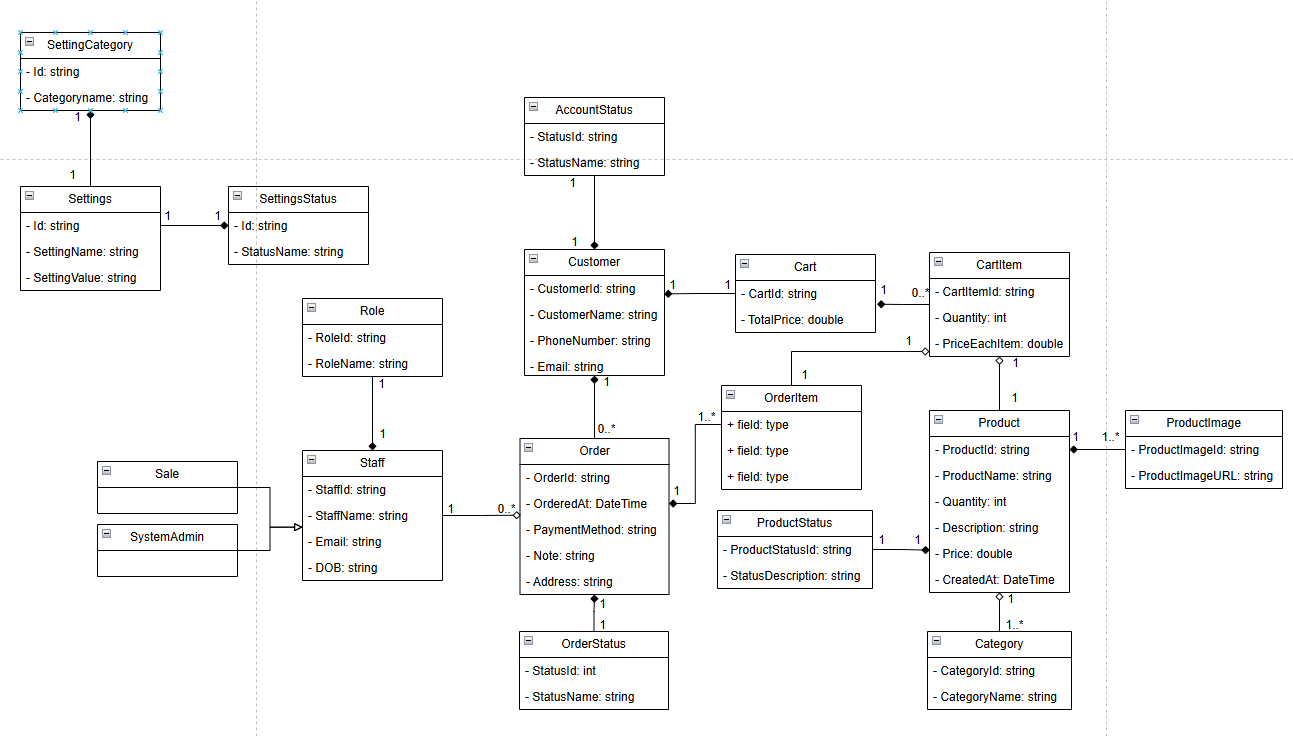
*[Provide the software system context class diagram (with suitable stereotypes) and the description for each of the class in the diagram like the sample below]*



This is a context model for an Online Shopping System. The online shopping system provides items for sale, which integrates the use of external systems such as vnPay and PayOS to support online payments, the Google Cloud system for account authentication and email sending, the Cloudinary system for image storage, and the GHN system for order tracking support.

## 2. Entity Classes

*[Provide entity classes and the relationships among those classes by entity class diagram like the samples below]*

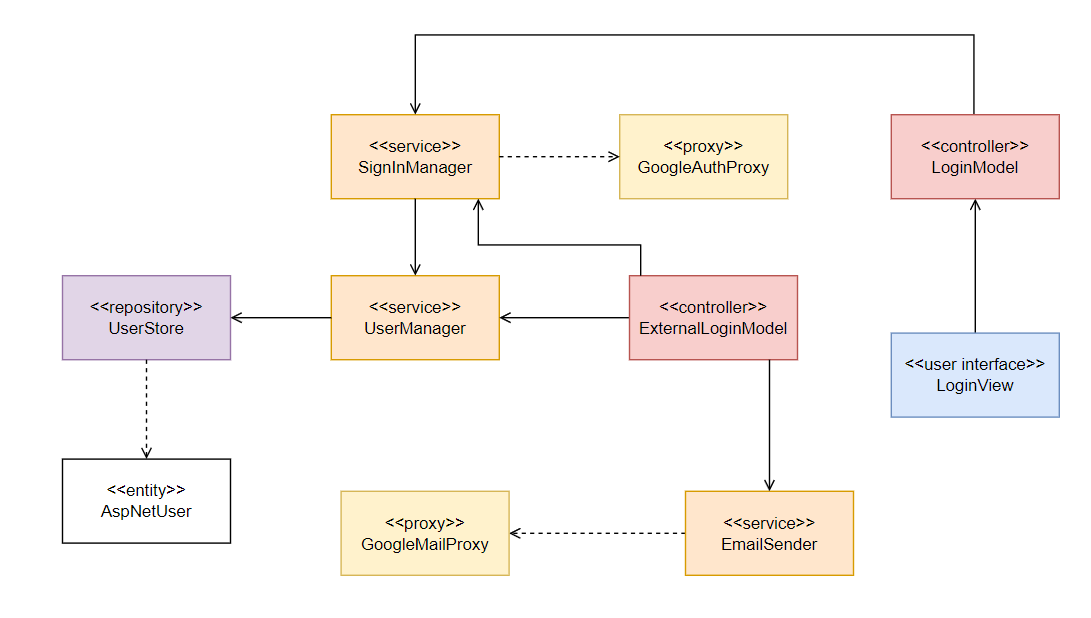


# 

# III. Object & Class Structuring

## 1. Login System

### a. Class Diagram



### b. Class Description

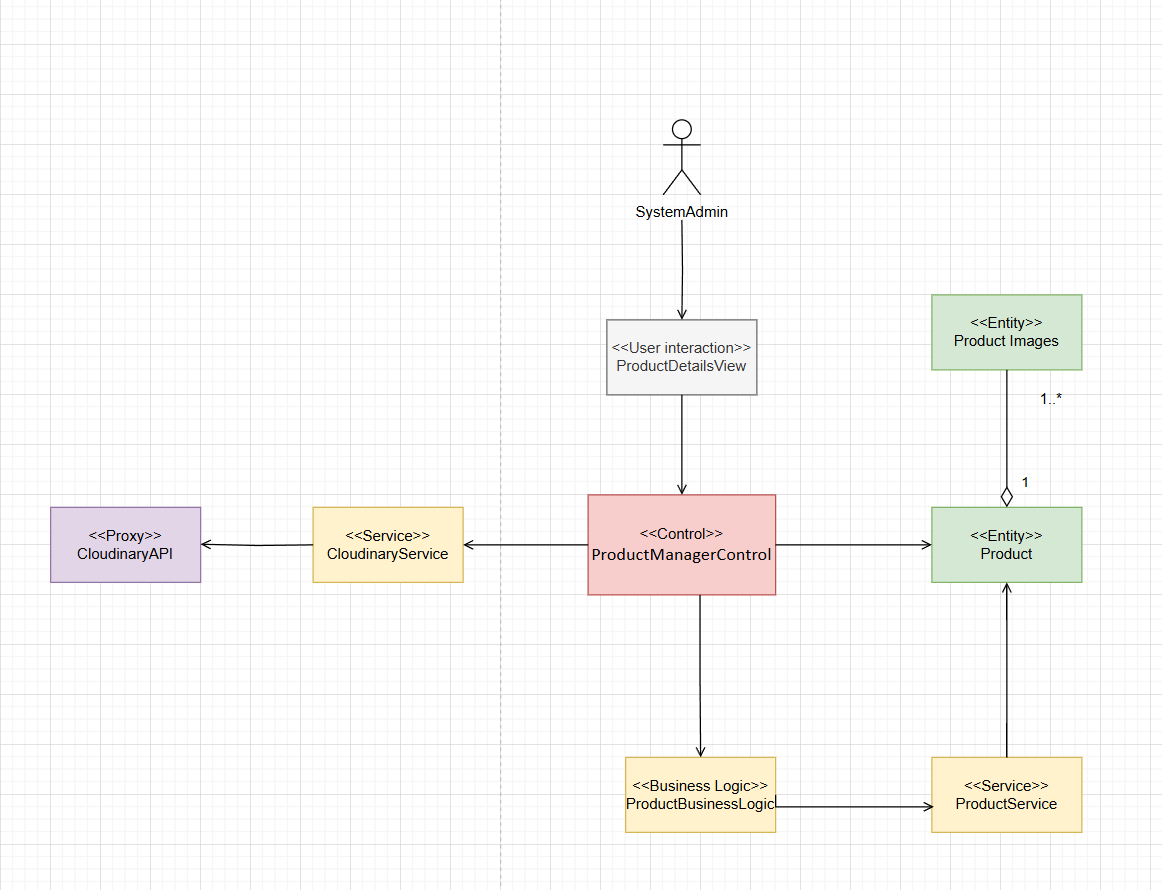
### 

| **#** | **Class** | **Category** | **Description** |
| --- | --- | --- | --- |
| 1 | LoginView | User Interface | The login interface where users enter their credentials or choose Google authentication. |
| 2 | LoginModel | Controller | Handles login logic, calls SignInManager for authentication and GoogleAuthService for Google login. |
| 3 | UserStore | Repository | Internal component of Identity Framework to get IdentityUser record |
| 4 | GoogleAuthProxy | Proxy | Communicates directly with Google’s API for user authentication, retrieves and processes responses. |
| 5 | SignInManager | Service | Manages user authentication, verifies credentials, and logs in users if valid. |
| 6 | AspNetUser | Entity | inherited from IdentityUser of Identity framework but have extended attribute, Represents a user account in the system, storing necessary authentication details. |
| 7 | ExternalLoginModel | Controller | Handles new user registration, receives user data from the registration form, and calls UserManager to create an account |
| 8 | UserManager | Service | Manages user accounts, handles account creation, authentication, and account status verification |
| 9 | EmailSender | Service | Handles sent email after login as new user successfully |
| 10 | GoogleMailProxy | Proxy | Handle send mail for confirmation |

## 

## 2. Add New Product

### a. Class Diagram



### 

### 

### 

### b. Class Description

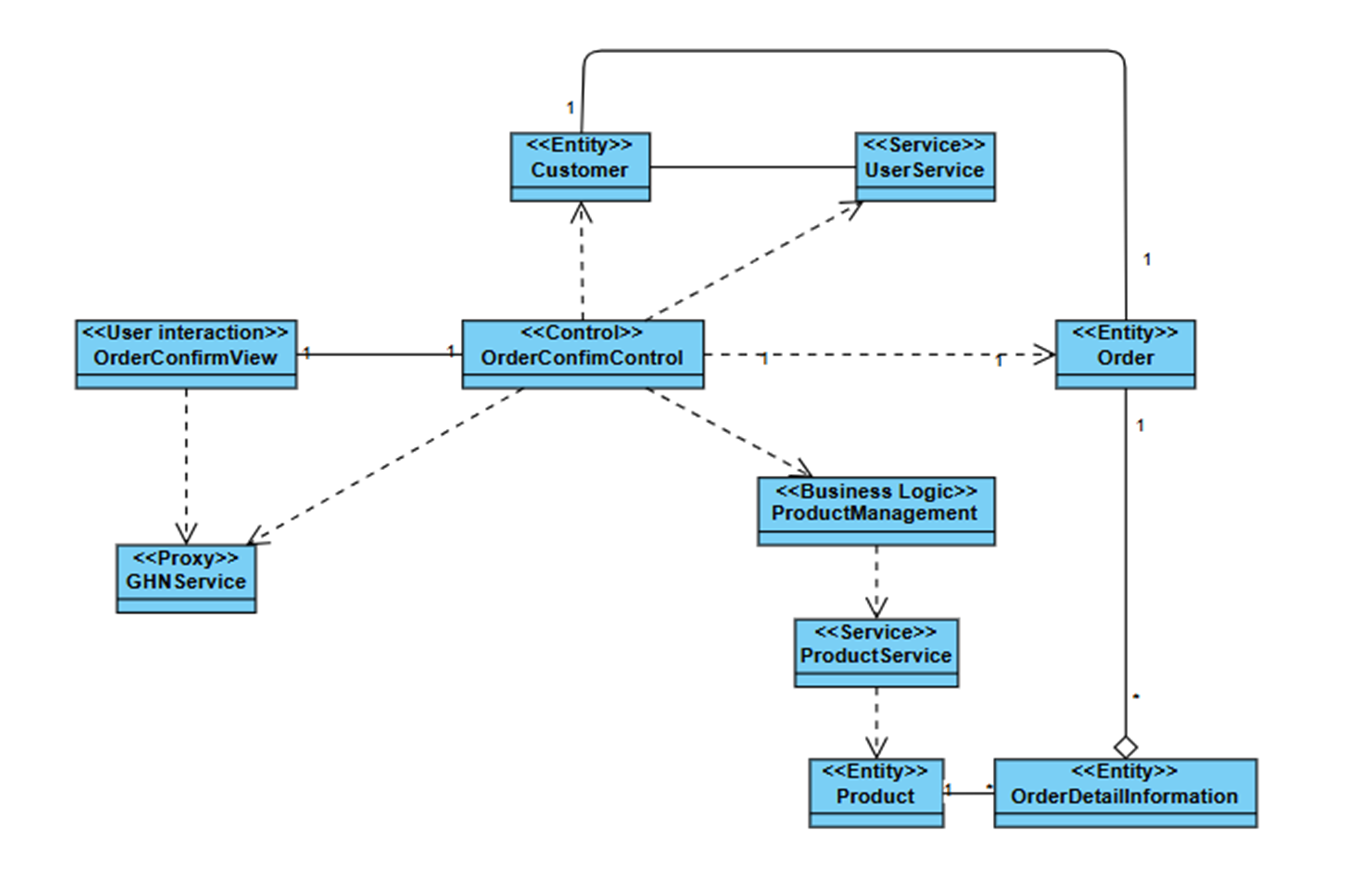
### 

| **#** | **Class** | **Category** | **Description** |
| --- | --- | --- | --- |
| 1 | SystemAdmin | Actor | Represents the system user who initiates the process of adding a new product. They interact with ProductManagerControl to start the product addition process. |
| 2 | AddProductPage | User Interaction | The boundary class that allows SystemAdmin to input product details and images. This class acts as the UI through which the admin interacts with the system. |
| 3 | ProductManagerControl | Control | Responsible for managing the entire process of adding a new product. This class collects product information from Product, processes the image through ImageProcessing, and communicates with AddProductUseCase to store product details. |
| 4 | ProductBusinessLogic | Business Logic | Handles the business rules and validation before sending product data to AddProductUseCase. It ensures data consistency and integrity before persisting. |
| 5 | AddProductUseCase | Service | The core service class that saves the product details into the system. It interacts with Product to persist data and ensures proper linking with images. |
| 6 | Product | Entity | Represents the main product entity, including attributes such as product name, description, price, and associated images. |
| 7 | ProductImages | Entity | Represents the product images that are stored in CloudinaryService. Each product can have multiple associated images. |
| 8 | ImageProcessing | Algorithm | Handles image preprocessing before storing in CloudinaryService. It includes resizing, compression, and format conversion for optimization. |
| 9 | CloudinaryService | Proxy | An external system used to store product images. ProductManagerControl interacts with CloudinaryService to upload product images and retrieve their URLs. |

## 

## 3. Confirm Order

### a. Class Diagram



### b. Class Description

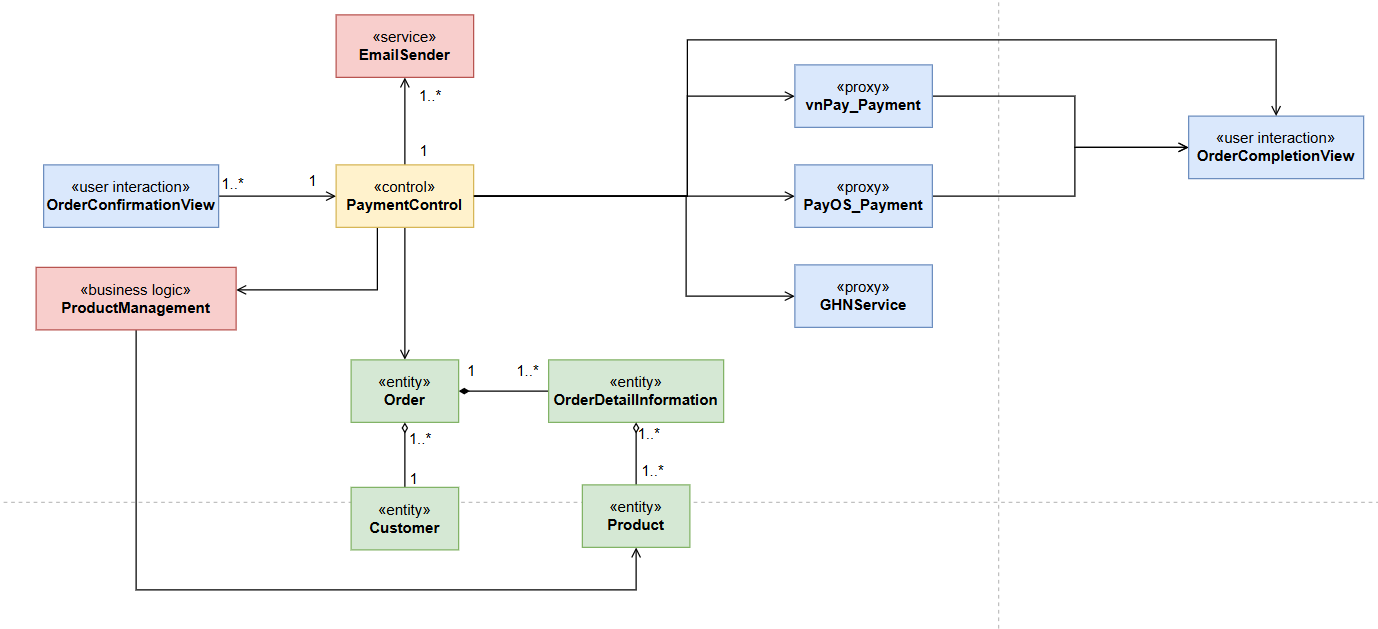
### 

| **#** | **Class** | **Category** | **Description** |
| --- | --- | --- | --- |
| 1 | Customer/Guest | Actor | Represents the user or guest who places an order. They interact with OrderConfirmView to confirm their order. |
| 2 | OrderConfirmView | User interaction | Responsible for displaying order details, including order summary, order form, and payment confirmation. |
| 3 | OrderConfirmController | Control | Coordinates the entire order confirmation process. It collects user data, checks login status, validates delivery information, and processes payments. |
| 4 | UserService | Service | Verifies the user's login status and authenticates their credentials. |
| 5 | Customer | Entity | Represents a registered user, containing details like ID, name, email, and phone number. |
| 6 | Product | Entity | Product representation includes Image, product name, product ID, Quantity, Selling price |
| 7 | ProductService | Service | Provides product-related operations such as retrieving product details, checking inventory levels, and updating product information. |
| 8 | GHNService | Proxy | Calculates the shipping fee based on delivery details and shipping policies. |
| 9 | Order | Entity | Represents an order in the system with all relevant order details. Contains relationships to Customer who placed it and OrderDetailInformation. |
| 10 | OrderDetailInformation | Entity | Contains detailed information about each order including specific product selections, quantities, pricing, and other order-specific metadata. |
| 11 | ProductManagement | Business Logic | Handles core business rules related to products including inventory management, pricing rules, and product availability checks |

## 

## 4. Checkout Order

### a. Class Diagram



### b. Class Description

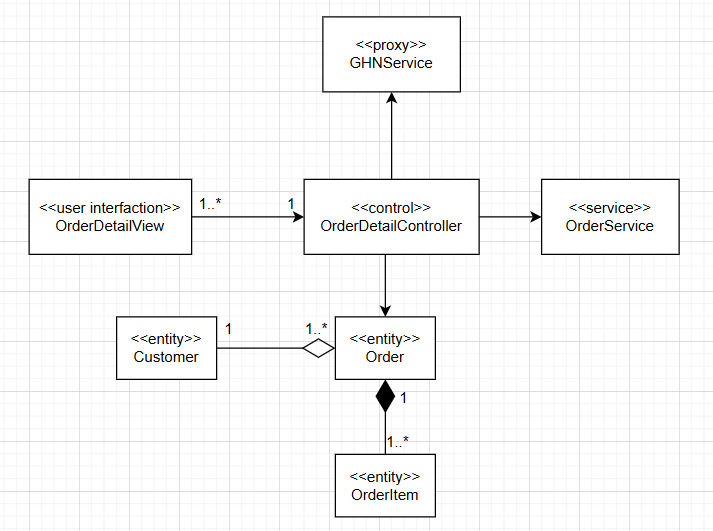
### 

| **#** | **Class** | **Category** | **Description** |
| --- | --- | --- | --- |
| 1 | OrderConfirmationView | User interaction | This is the class that interacts with the user. After the user has confirmed the order and selected the payment method, the user will press the "Checkout" button. |
| 2 | OrderCompletionView | User interaction | This is a user-interactive class that will return order information that the user has placed. |
| 3 | PaymentControl | Control | This is the control class after the user sends a checkout request to the server, this control class will process depending on the payment method the user has chosen and proceed to process the payment, send email and create a new order to GHN. |
| 4 | EmailSender | Service | This is a Service class configured with Google SMTP to send emails to users after successfully creating an order and is called from the Control class. |
| 5 | ProductManagement | Business logic | This is a class to manage the logic of the quantity of goods, such as handling the subtraction of the quantity of goods in stock if an order is successfully placed or adding it if there is a returned order or failed payment. |
| 6 | Order | Entity | This is the class to store order information such as orderId, customerId, orderDetailId, paymentMethod,.... |
| 7 | OrderDetailInformation | Entity | This is the class to store order detail information such as orderId, productId, quantities,.. |
| 8 | Product | Entity | This is the class to store product information such as productName, productId, productQuantities, productImage,.. |
| 9 | Customer | Entity | This class stores customer information such as customerId, customerName, address,.. |
| 10 | vnPay\_Payment | Proxy | This is a proxy class configured to return the payment page of the vnPay system. |
| 11 | PayOS\_Payment | Proxy | This is a proxy class configured to return the payment page of the PayOS system with QR code. |
| 12 | GHNService | Proxy | This is a proxy class to call GHN's API with the purpose of calling the API to create a new order and send information to GHN after the user has successfully created an order. |

## 

## 5. Track Order Details

### a. Class Diagram



### b. Class Description

### 

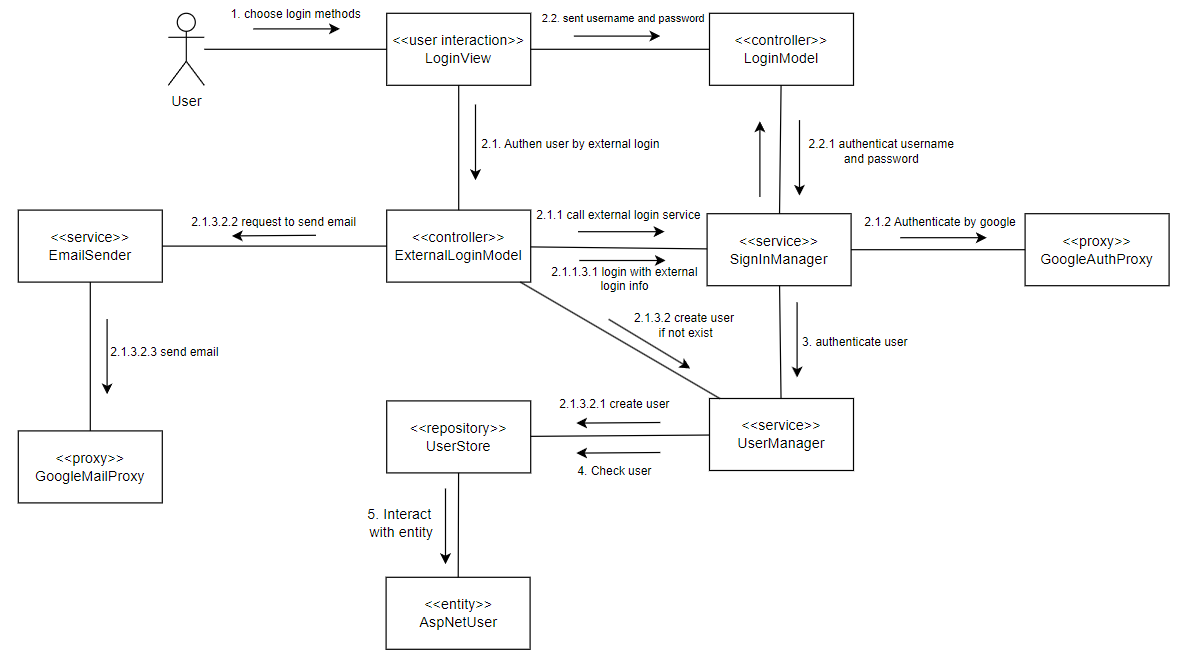
| **#** | **Class** | **Category** | **Description** |
| --- | --- | --- | --- |
| 1 | OrderDetailView | User interaction | Responsible for displaying detailed information about the order, including order summary, order cost, and shipping process |
| 2 | OrderDetailController | Control | Responsible for managing order information and status. It accesses and displays order data, changes details and status of submitted orders. |
| 3 | Customer | Entity | Represents registered users such as customers and sales staff, including details such as ID, name, email and phone number. |
| 4 | Order | Entity | Represents user orders including address, payment method, status, notes |
| 5 | OrderItem | Entity | Represents the products in the user's order including product name, quantity, price |
| 6 | OrderService | Service | Displays order information, changes details of submitted orders, updates order status with notes |
| 7 | GHNService | Proxy | This is a proxy class to call GHN's API for the purpose of updating order details after submission and passing the information to GHN after the user successfully creates the update. |

## 

# IV. Dynamic Interaction Modeling

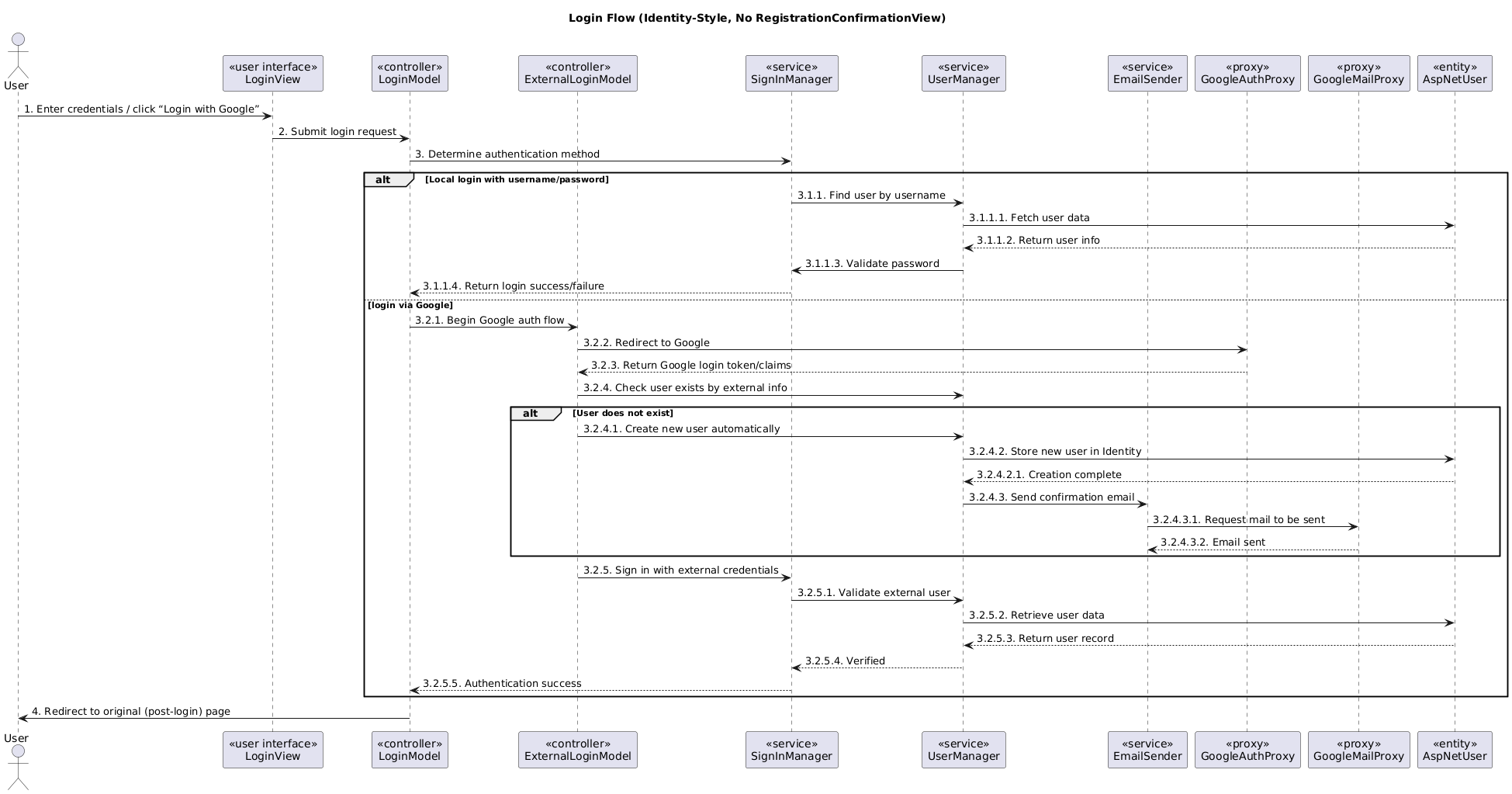
## 1. Login System

### a. Communication Diagram



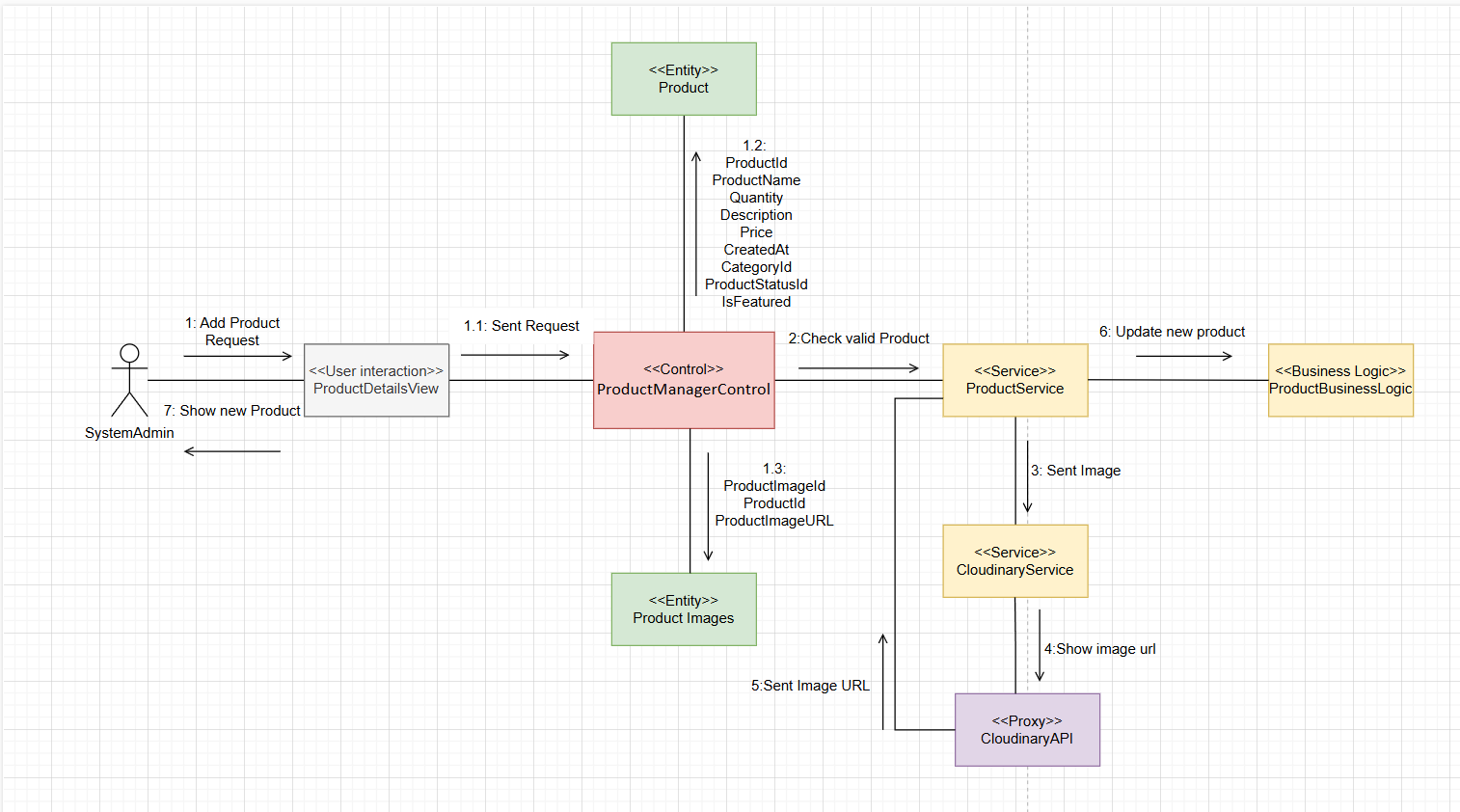
### 

### b. Sequence Diagram

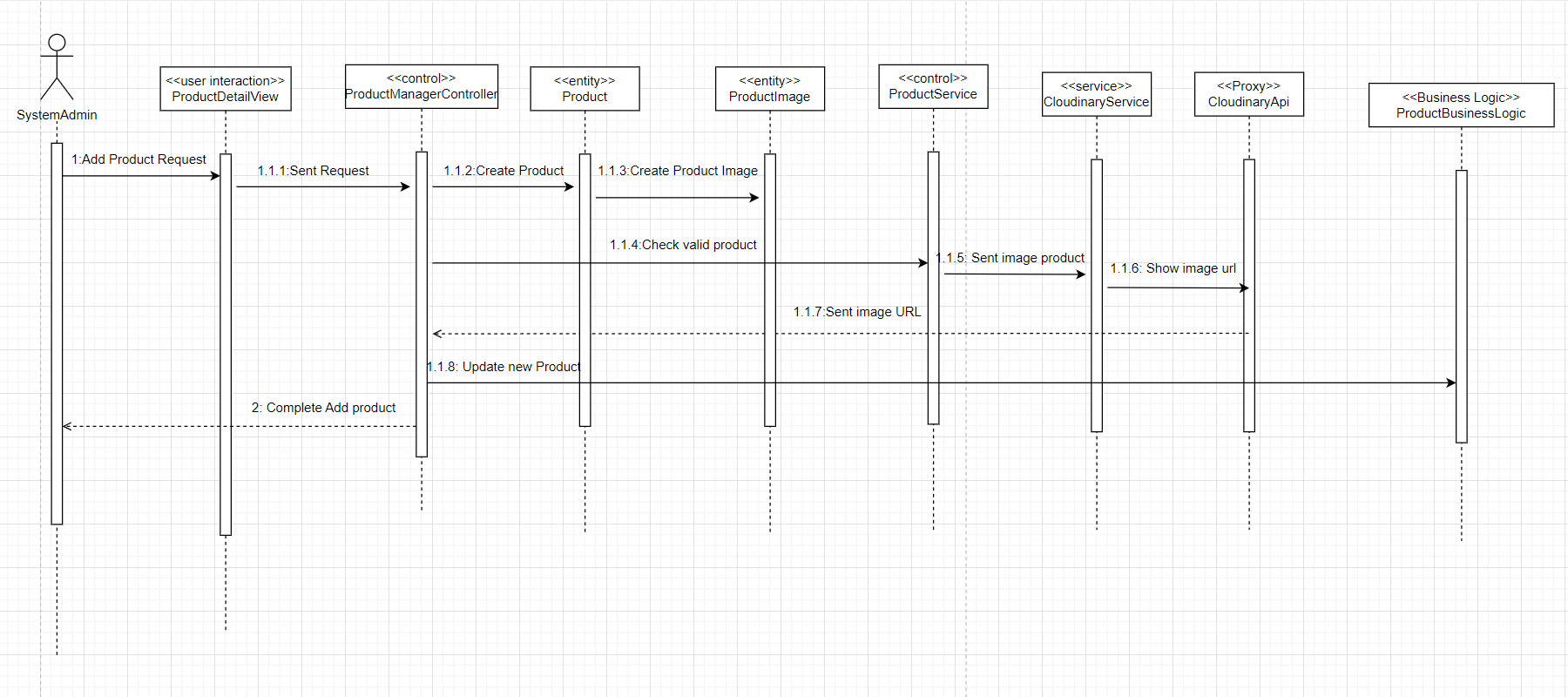


## 2. Add New Product

### a. Communication Diagram



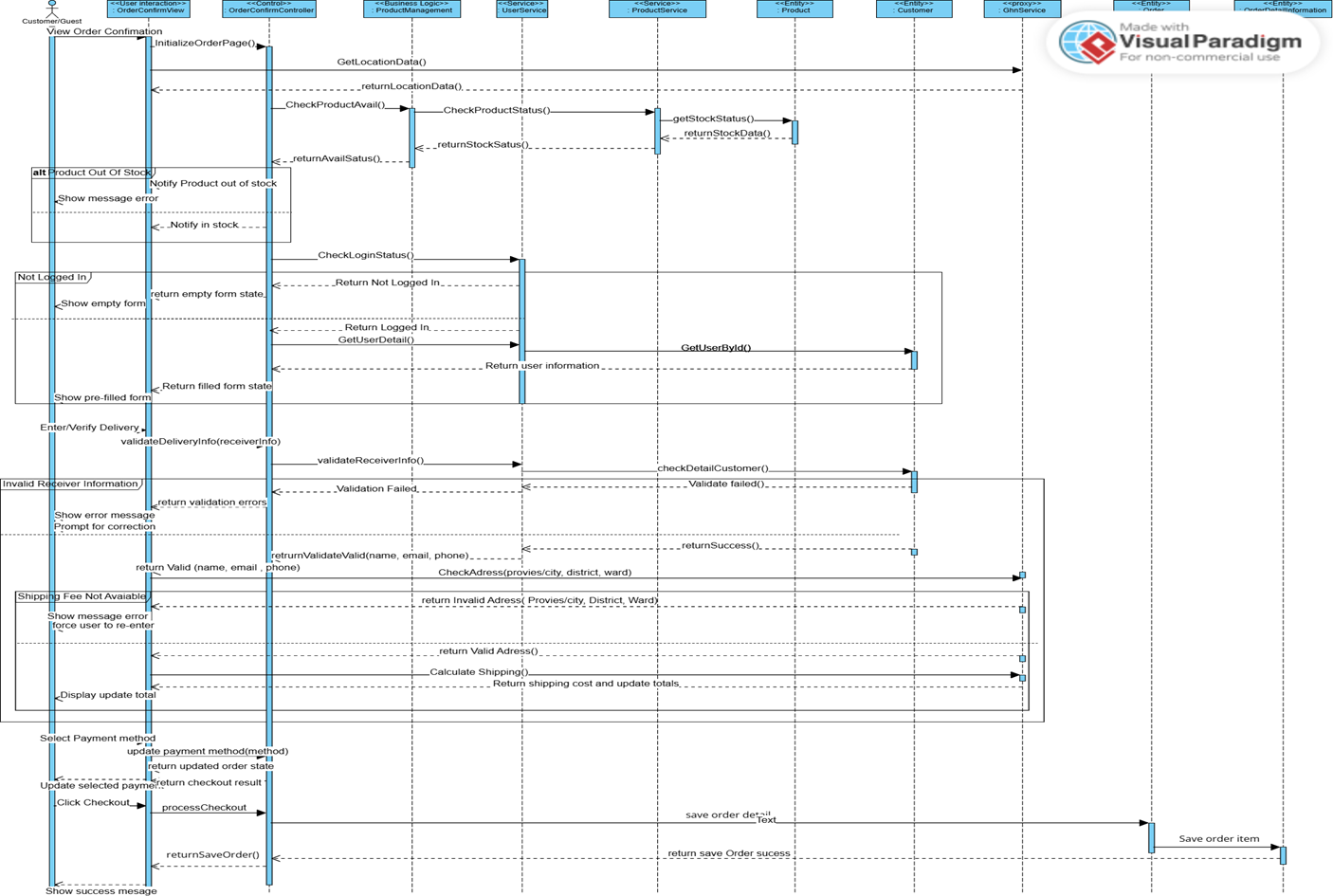
### b. Sequence Diagram



## 3. Confirm Order

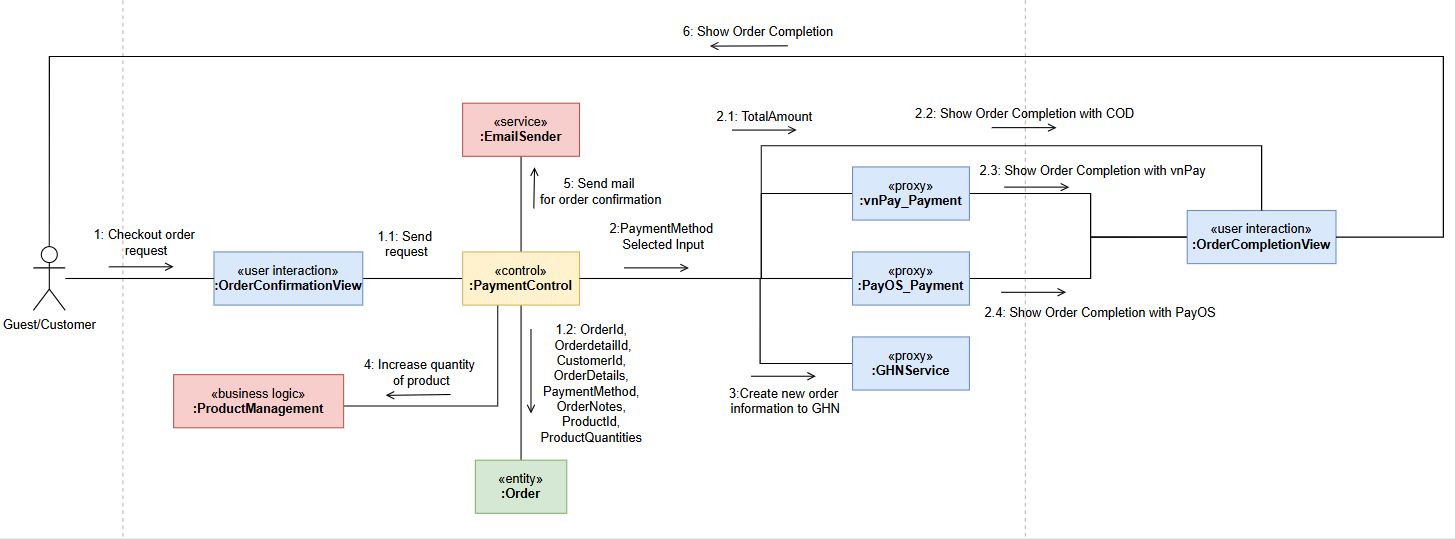
### a. Communication Diagram

### b. Sequence Diagram



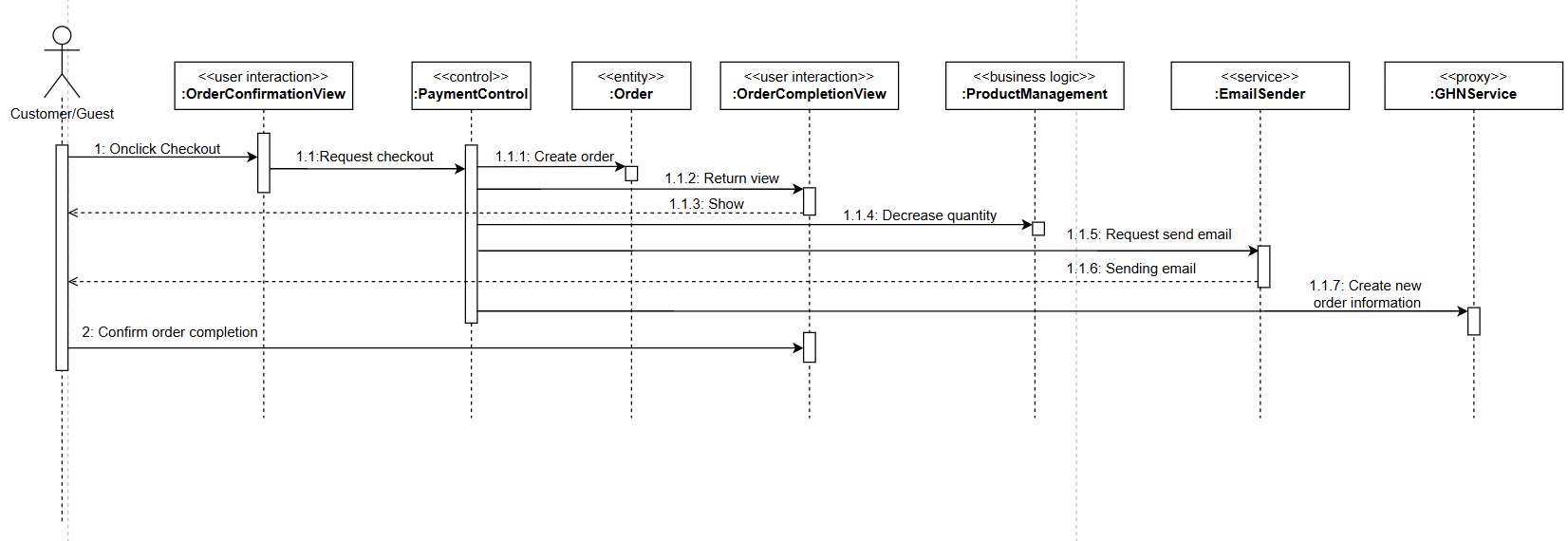
## 4. Checkout Order

### a. Communication Diagram

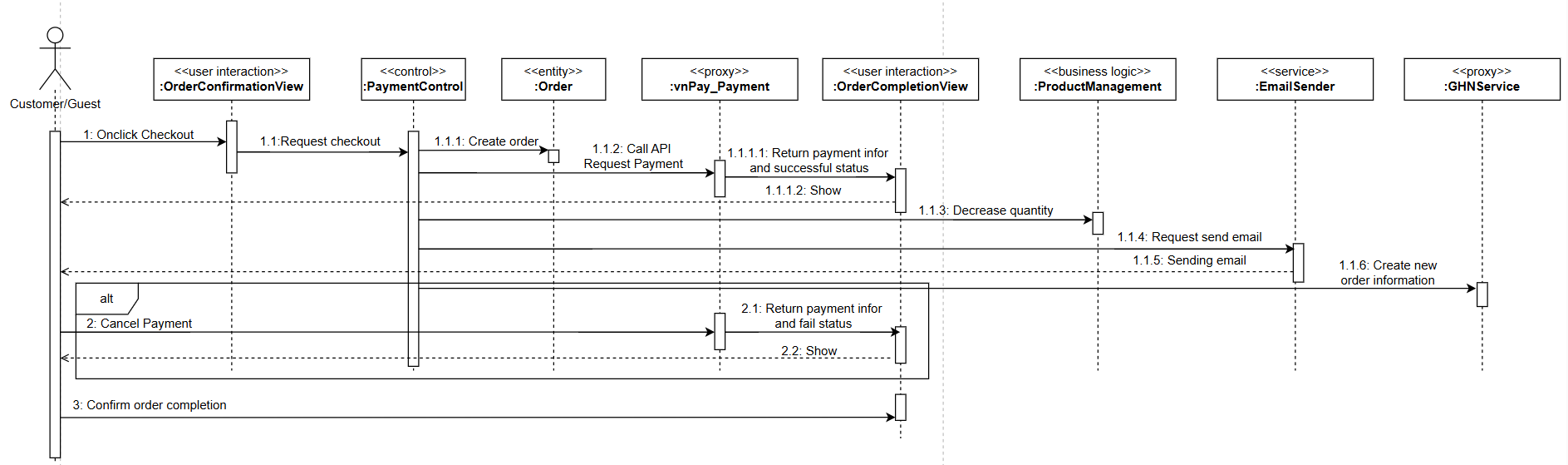


### b. Sequence Diagram

***Checkout by COD method Sequence Diagram:***



***Checkout by Online Payment method (vnPay) Sequence Diagram:***

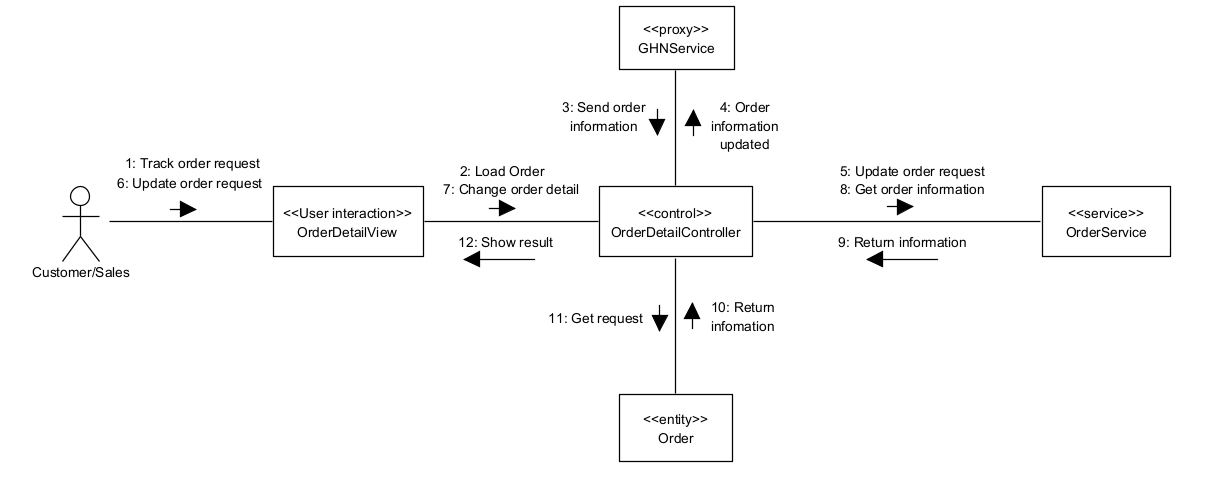
******

***Checkout by Bank Transfer method (PayOS) Sequence Diagram:***

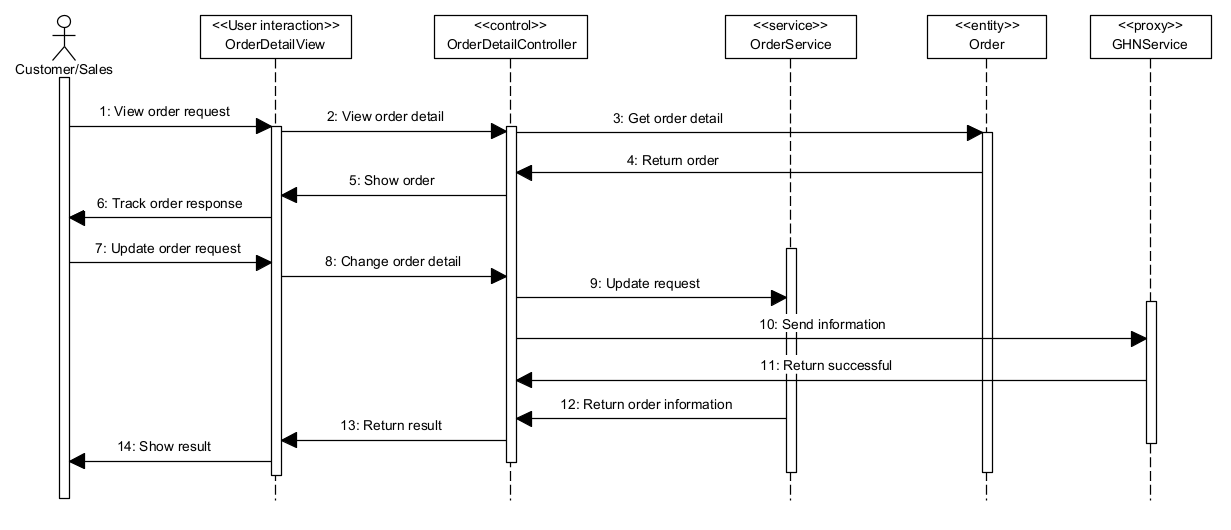
## 

## 5. Track Order Details

### a. Communication Diagram

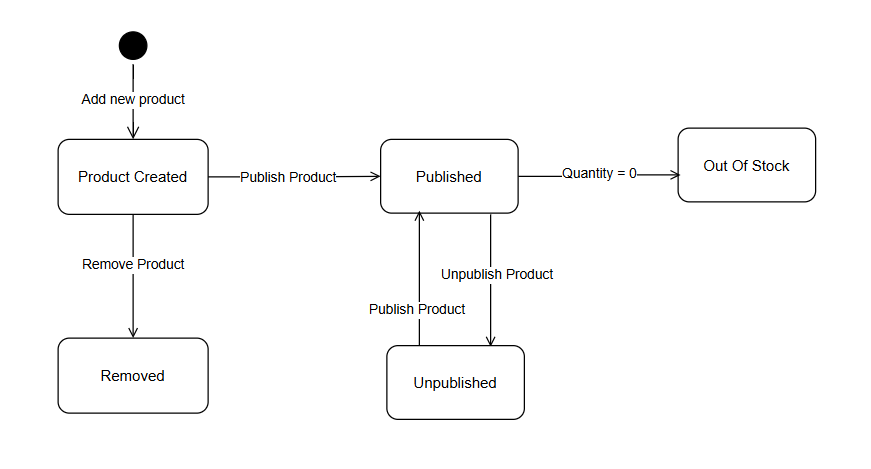


### b. Sequence Diagram

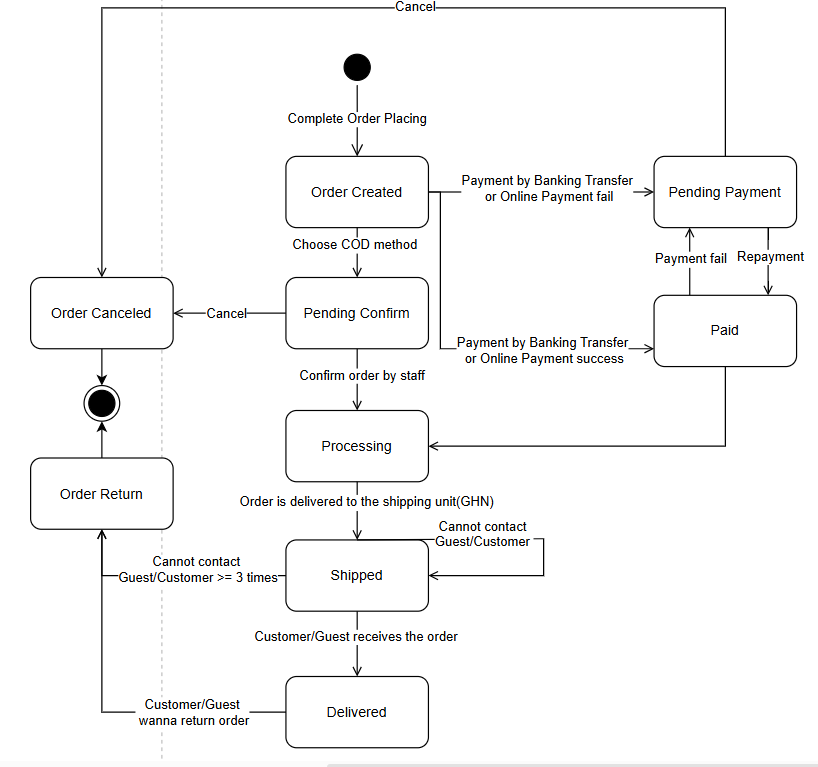
******

# V. Finite State Machine

## 1. Product



## 2. Order



## 3. User(Customer/Guest)

