

**User Requirement Document (URD)**

-Truong Nguyen Viet Quang-

-HE182422-

– Hanoi, Mar 2025 –

**Table of Contents**

[I. Record of Changes 3](#_heading=h.3as4poj)

[II. Software Requirement Specification 4](#_heading=h.1pxezwc)

[1. Overall Requirements 4](#_heading=h.49x2ik5)

[1.1 Context Diagram 4](#_heading=h.2p2csry)

[1.2 System Features 5](#_heading=h.147n2zr)

[2. Use Case Modelling 6](#_heading=h.3o7alnk)

[2.1 Actors 6](#_heading=h.23ckvvd)

[2.2 UC Diagrams 7](#_heading=h.ihv636)

[2.3 Activity Diagrams 8](#_heading=h.32hioqz)

[3. Use Case Specifications 9](#_heading=h.1hmsyys)

[3.1 <<Feature Name1>> 9](#_heading=h.41mghml)

[4. Non-Functional Requirements 11](#_heading=h.2grqrue)

[4.1 External Interfaces 11](#_heading=h.vx1227)

[4.2 Quality Attributes 11](#_heading=h.3fwokq0)

# I. Record of Changes

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

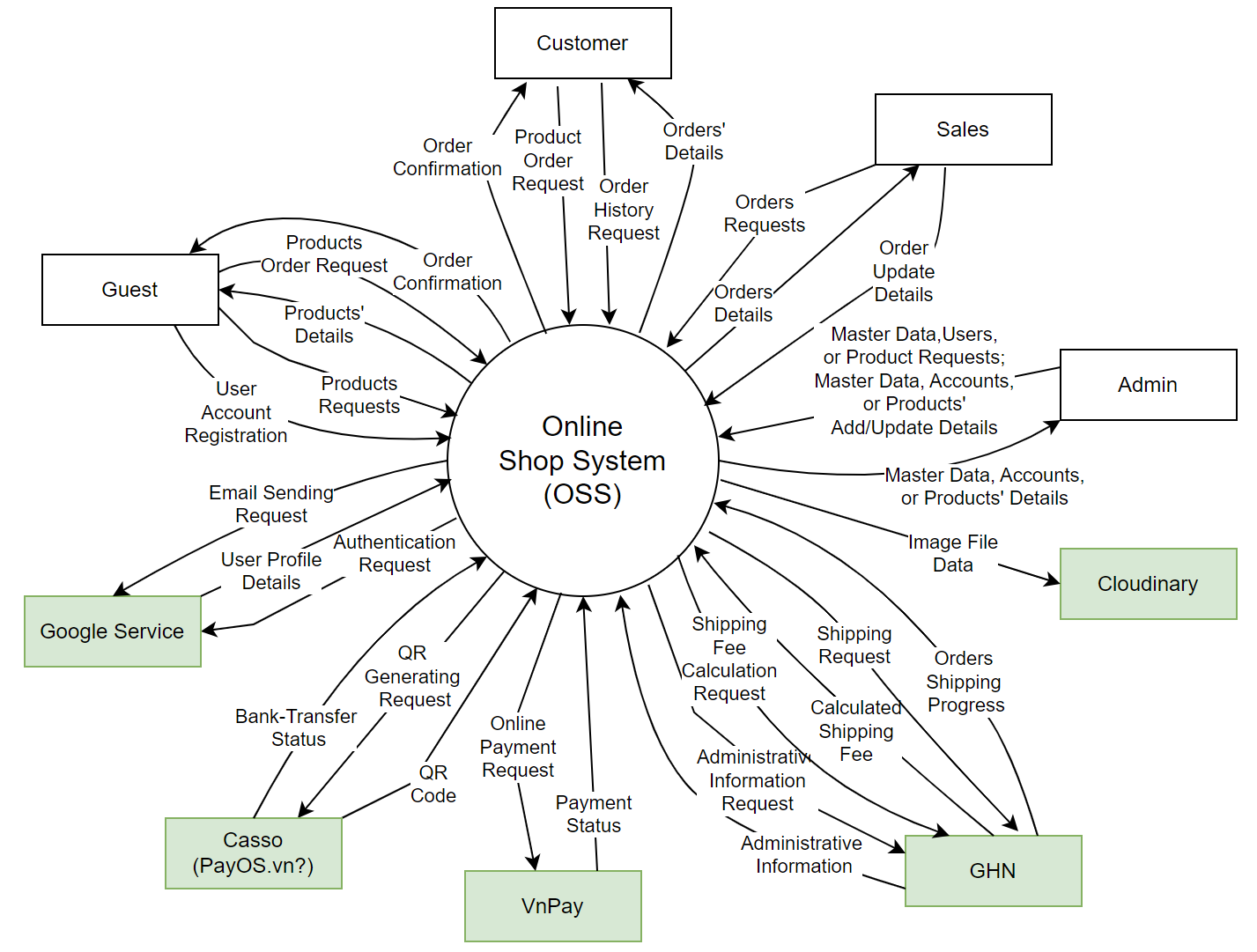
**\*A - Added M - Modified D - Deleted**

# II. Software Requirement Specification

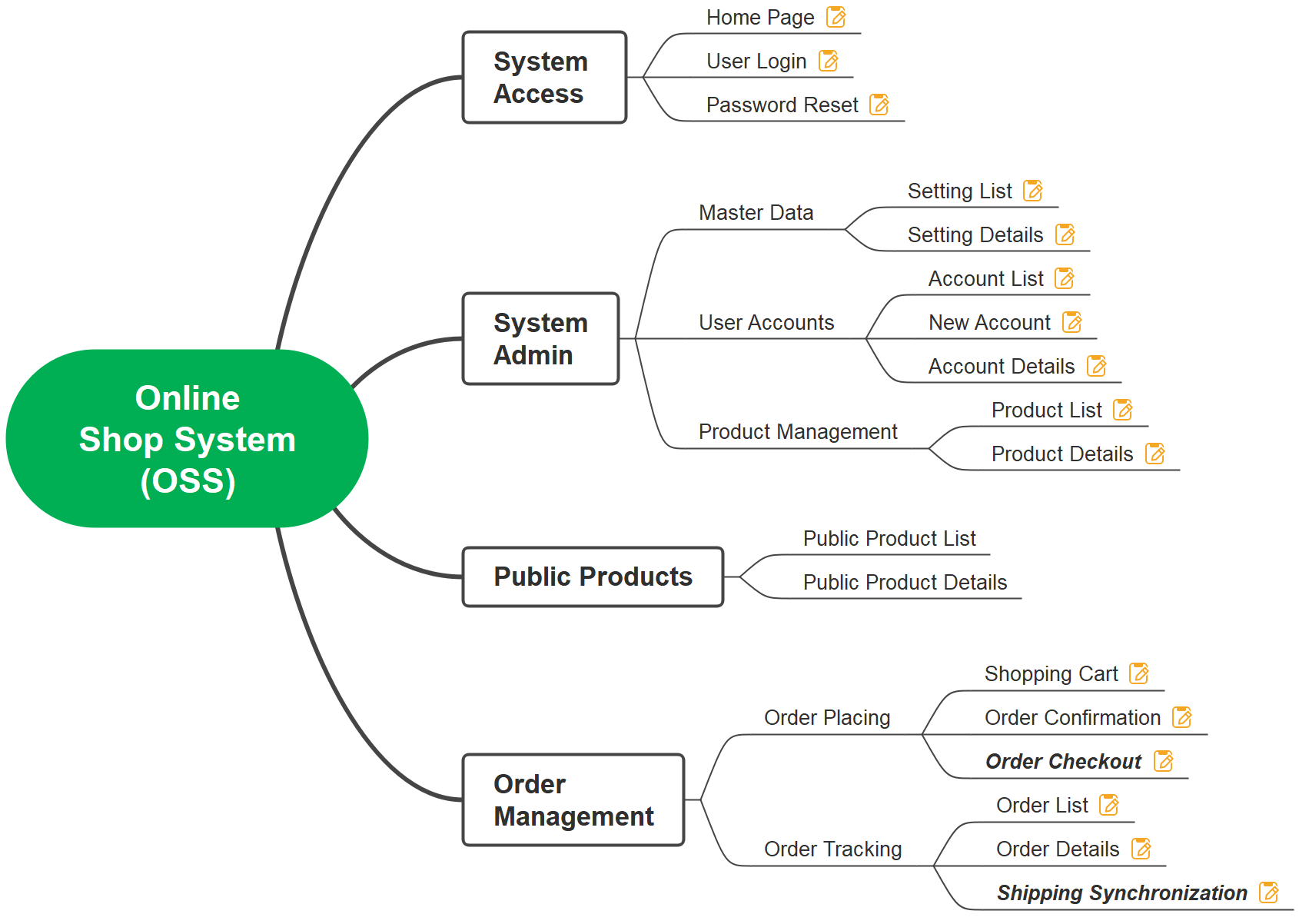
## 1. Overall Requirements

### 1.1 Context Diagram

The Online Shopping System is a new software system that replaces the current manual and telephone processes for ordering and delivering products in the ABC shop. The context diagram below illustrates the external entities and system interfaces for release 1.0. The system is expected to evolve over several releases, ultimately connecting to the Internet ordering services for any shops with one or more shop branches.



### 1.2 System Features



| **#** | **Feature** | **Function/Screen** | **Description** |
| --- | --- | --- | --- |
| 1 | System Access | Home Page | Show featured products, latest products |
| 2 | System Access | User Login | Authenticate users to the system. For the first time user to login with Gmail, new user account is registered to the system with relevant notification to his/her email |
| 3 | System Access | Reset Password | Allow users to reset their forgot password |
| 4 | Master Data | Setting List | Show system master data: user roles, system pages, product categories, shop’s bank accounts, etc. |
| 5 | Master Data | Setting Details | Allow System Admin to add new or update master data |
| 6 | User Accounts | Account List | Show registered user accounts, from that system admin can activate/deactivate a specific account |
| 7 | User Accounts | New Account | Allow System Admin to add new account, relevant email notification would be sent to that new account |
| 8 | User Accounts | Account Details | Allow System Admin to view and/or update details for a specific user account |
| 9 | Product Management | Product List | Show registered product, from that system admin can publish/unpublish/remove a specific product |
| 10 | Product Management | Product Details | Allow System Admin to add new or update a product details, product’s image(s) are stored on the Cloudinary |
| 11 | Public Products | Public Product List | Allow user to query and view the list of published products, from that he/she can choose to view details, add product to shopping cart or buy that right away |
| 12 | Public Products | Public Product Details | Show the details of a specific product, from that user can add product to shopping cart or buy that right away |
| 13 | Order Placing | Shopping Cart | Show the products that user has selected. On this page, user can choose to change the product quantity, remove, or to place the order |
| 14 | Order Placing | Order Confirmation | Show the list of selected products with relevant costs.  Allow user to input/update receiver information, including the administrative details (city/province, district, ward) for defining the shipping fee;  Allow user to select payment method (Online Payment, Bank Transfer, or COD) before checking out the order |
| 15 | Order Placing | Order Checkout | Allow the user to check out the order details, show and send the order placing notification message to the user. Before that, extra payment details can be shown if the selected payment method is not COD:  \* For Online Payment: show the payment details and get payment status via integrating with VnPay  \* For Bank-Transfer: show QR codes and updating back with the bank-transfer status (via Casso/PayOS system) |
| 16 | Order Tracking | Order List | Allow Customer or sales to query the list of placed orders, from that allowing them to cancel a specific submitted orders |
| 17 | Order Tracking | Order Details | Allow Customer or Sales to query the details of a specific order (including shipping progress if any), change the submitted order details  Allow Sales to update the order status with notes |
| 18 | Order Tracking | Shipping Synchronization | This is system functions to   * Upload/submit the shipping-ready orders to request external shipping service (GHN) to deliver the orders. This is done hourly (as configured by System Admin) * Get the shipping progress and status (from GHN) |

## 2. Use Case Modelling

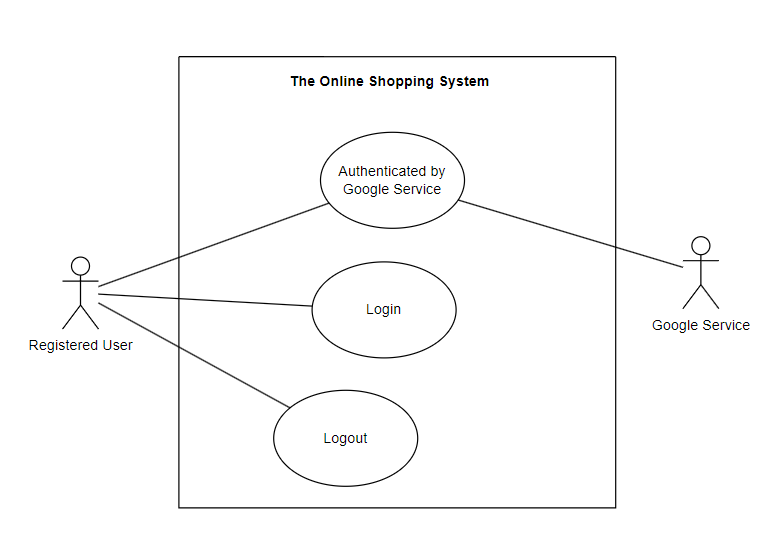
### 2.1 Actors

| **#** | **Actor** | **Description** |
| --- | --- | --- |
| 1 | Guest | These are people who do not have an account to log into the system, but they can still make purchases normally. |
| 2 | Customer | These are people who already have an account to log into the system, they will have some more special features than Guests in addition to the online shopping feature. |
| 3 | Sale | Process and track order quantity, can receive or cancel orders from customers, update item types and quantity increases to the system |
| 4 | System Admin | As the person with the highest access rights to the system, can customize the system configuration parameters, view registered user accounts, Activate/Deactivate a specific account, Add new accounts, View and/or update details for a specific user account. |
| 5 | Google Authentication Service | This service provides authentication functionality using Google accounts. It allows users to log in to the OSS securely using their Google credentials, enhancing the security and convenience of the authentication process.s |
| 6 | Casso (PayOS.vn) | This is an external system service that supports authentication and QR payment generation for the OSS system. |
| 7 | VnPay | This is a third party service that facilitates payments and returns payment status to OSS |
| 8 | GHN | This is also a service of the GHN system that returns information to OSS such as shipping fees, shipping information and shipping status. |

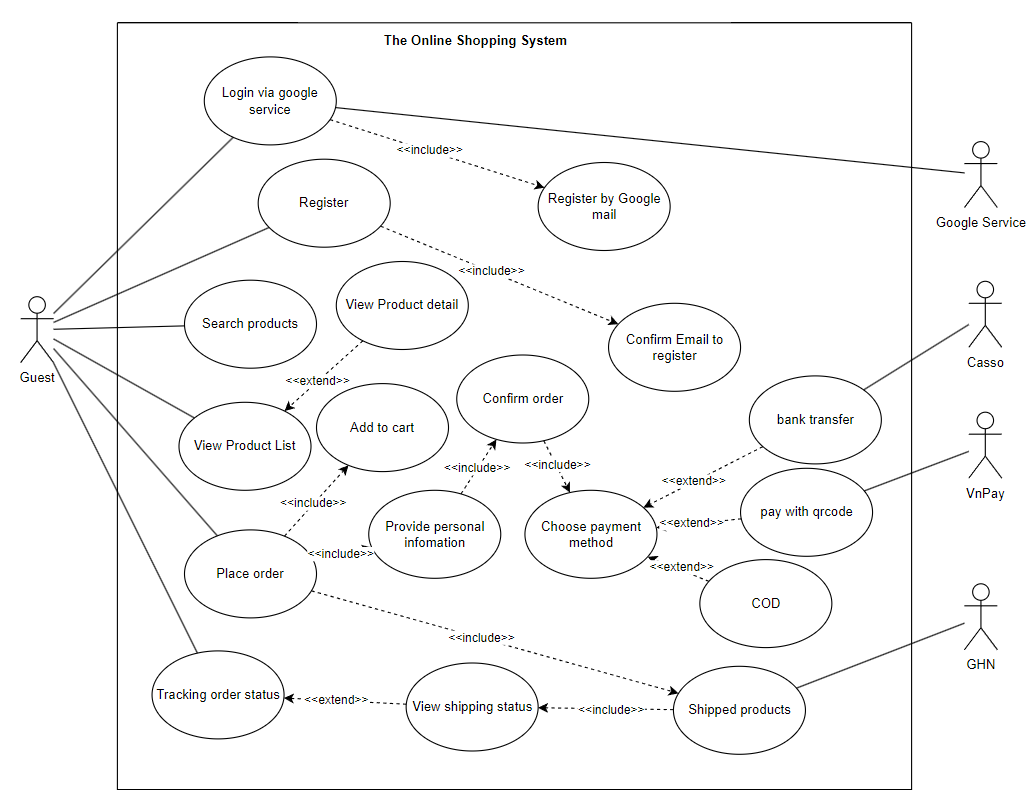
### 

### 2.2 UC Diagrams

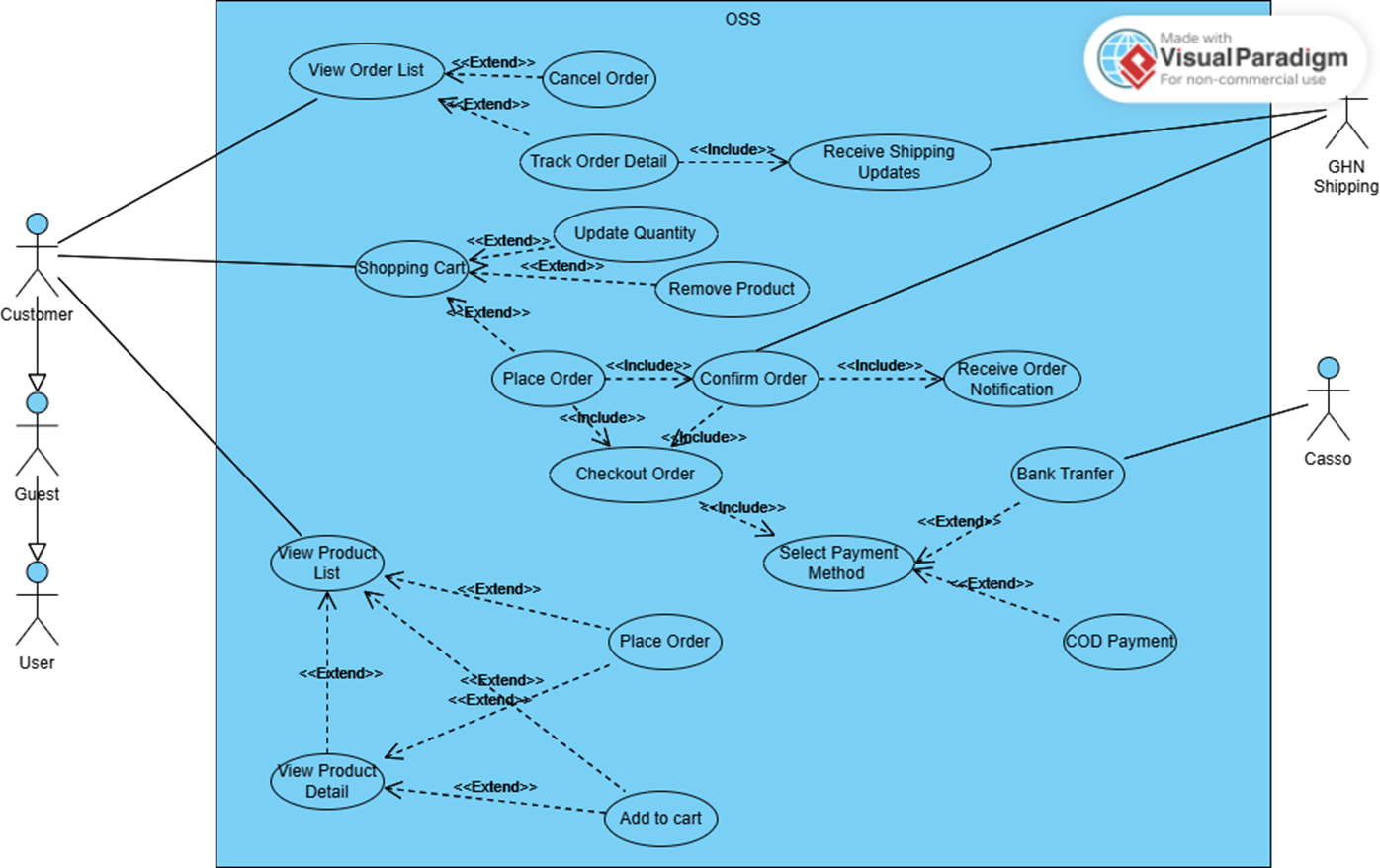
#### 2.2.1 UCs for Registered User

**

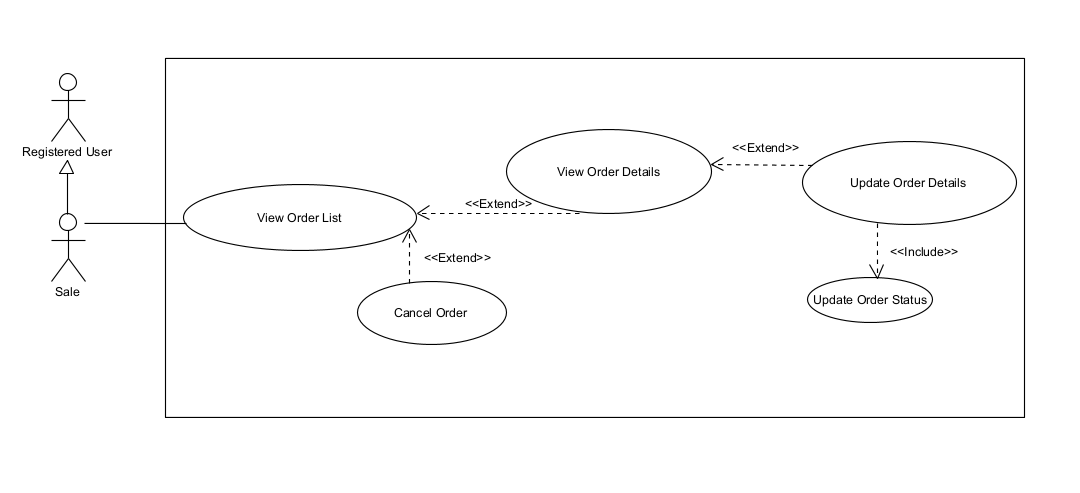
#### 2.2.2 UCs for Guest

**

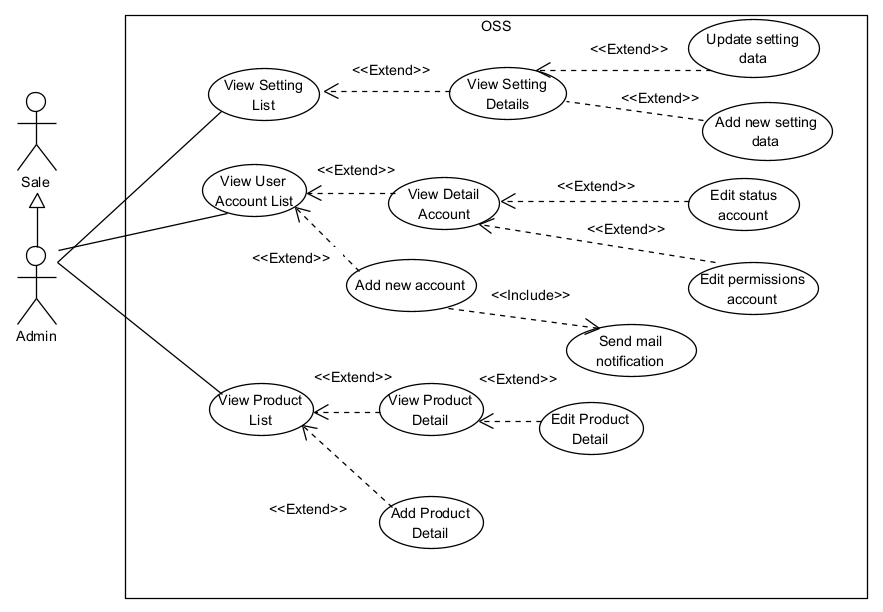
#### 2.2.3 UCs for Customer



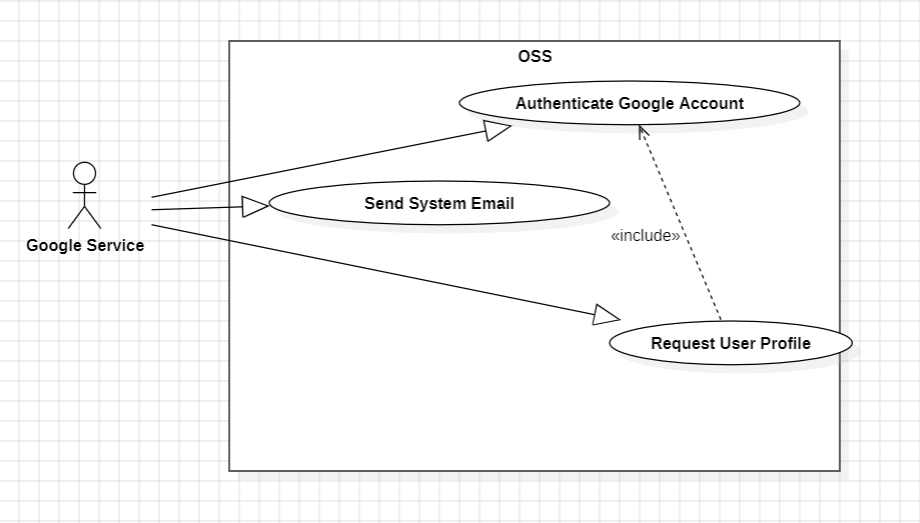
#### 2.2.4 UCs for Sale



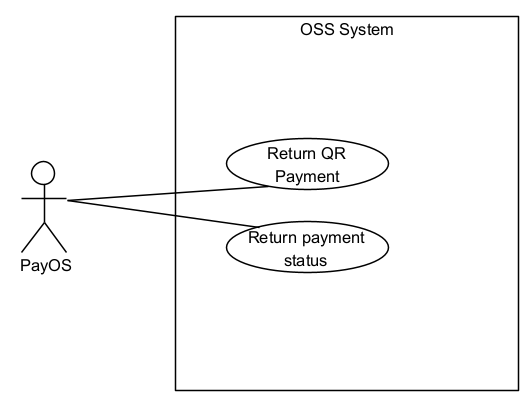
#### 2.2.5 UCs for Admin



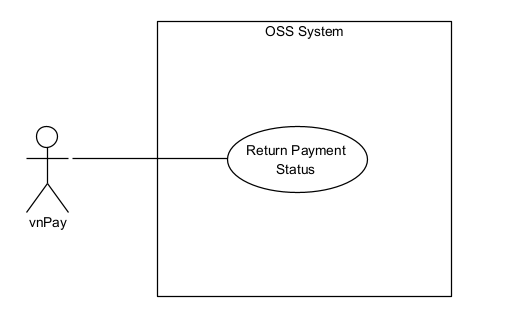
#### 2.2.6 UCs for Google Authentication Service



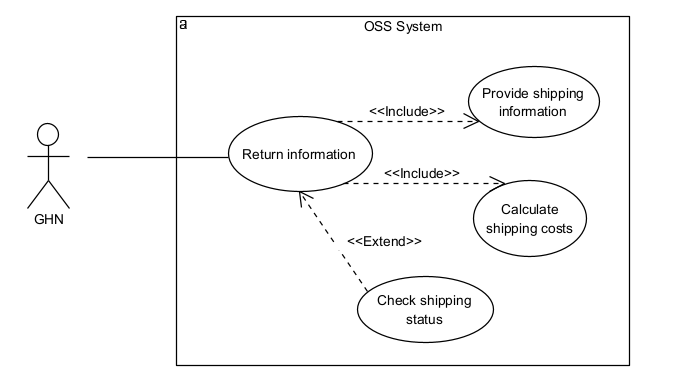
#### 2.2.7 UCs for Casso (PayOS)



#### 2.2.8 UCs for vnPay



#### 2.2.9 UCs for GHN



### 2.3 Activity Diagrams

#### 2.3.1. Login System (Customer)

#### Summary: This use case allows four types of actors—Guest, Customer, Sale, and Admin—to log into the system. Users can choose either Traditional Login (username/password) or Google Login (OAuth). For Traditional Login, the system validates credentials and either creates a session token if valid or displays an error if invalid. For Google Login, the system redirects to Google OAuth, retrieves an authentication token, and verifies it. If the token is valid and the user exists, the login is successful; if the user does not exist, the system auto-registers them using Google data. Successful logins redirect the actor to an appropriate dashboard or homepage. If login fails (wrong password, invalid OAuth token, or user denies authorization), the system displays an error and may prompt the user to retry.

#### Dependency:

#### The system must be operational and have internet connectivity.

#### Google Service integration must be configured for Google Login.

#### The user must be on the login page or have clicked a “Login” button/link.

#### If using Traditional Login, the user must have valid credentials stored in the system.

#### If using Google Login, the system depends on Google OAuth to provide a valid authentication token.

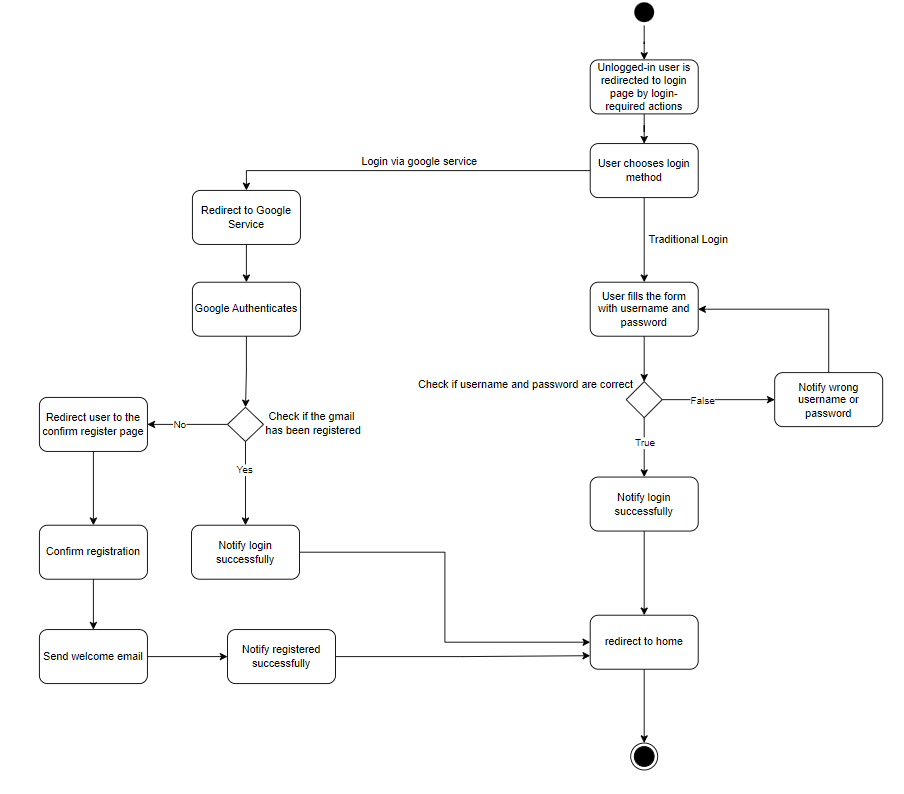
#### Actors:

* Primary: Guest, Customer, Admin, Sale
* Secondary: Google Service

#### Preconditions:

* The user has navigated to the login page (or triggered the login flow).
* The system is online and connected to external services if needed (Google).
* Google OAuth settings are properly configured (for Google login).

#### Activity Diagram:



#### 2.3.2. Add New Product (System Admin)

#### Summary: The "Add New Product" use case allows a System Admin to create and add a new product to the system by providing product details such as name, description, category, price, and uploading product images. Once completed, the new product becomes available for display to the public.

#### Dependency:

#### This use case depends on the "Product Management" functionality, which includes managing product categories, and may be affected by any system configuration changes made by the System Admin in the "Master Data" settings.

#### It also relies on the Cloudinary system for storing product images.

#### Actors:

* Primary: System Admin
* Secondary Actors:

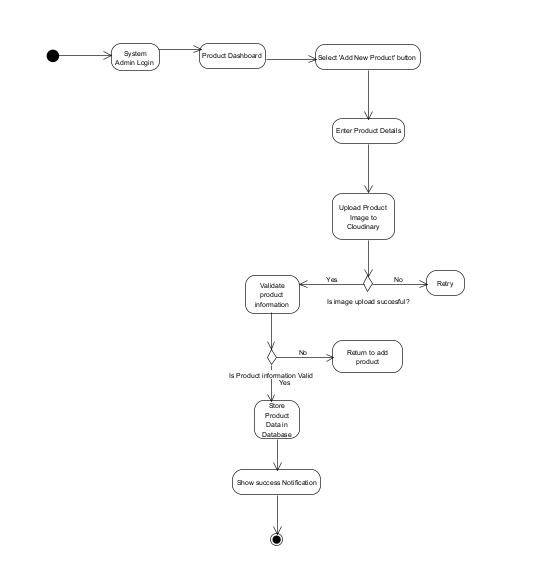
Cloudinary (for storing product images)

System (for validation and adding data to the database)

#### Preconditions:

* The System Admin must be logged into the system with proper administrative privileges.
* The system should have the necessary product categories already configured.
* The Cloudinary service must be operational for image uploads.

#### Activity Diagram:



#### 2.3.3. Confirm Order (Guest, Customer)

**Summary:**

This use case describes an order confirmation process where the user reviews the selected products, enters or updates recipient information, administrative details (city/province, district, ward) to determine shipping costs, selects payment method and completes the order.

**Dependency:**

* This use case may extend "Select Products" if the user needs to update the product list before confirming the order.
* It may also include "Shipping Fee Calculation" through integration with GHN

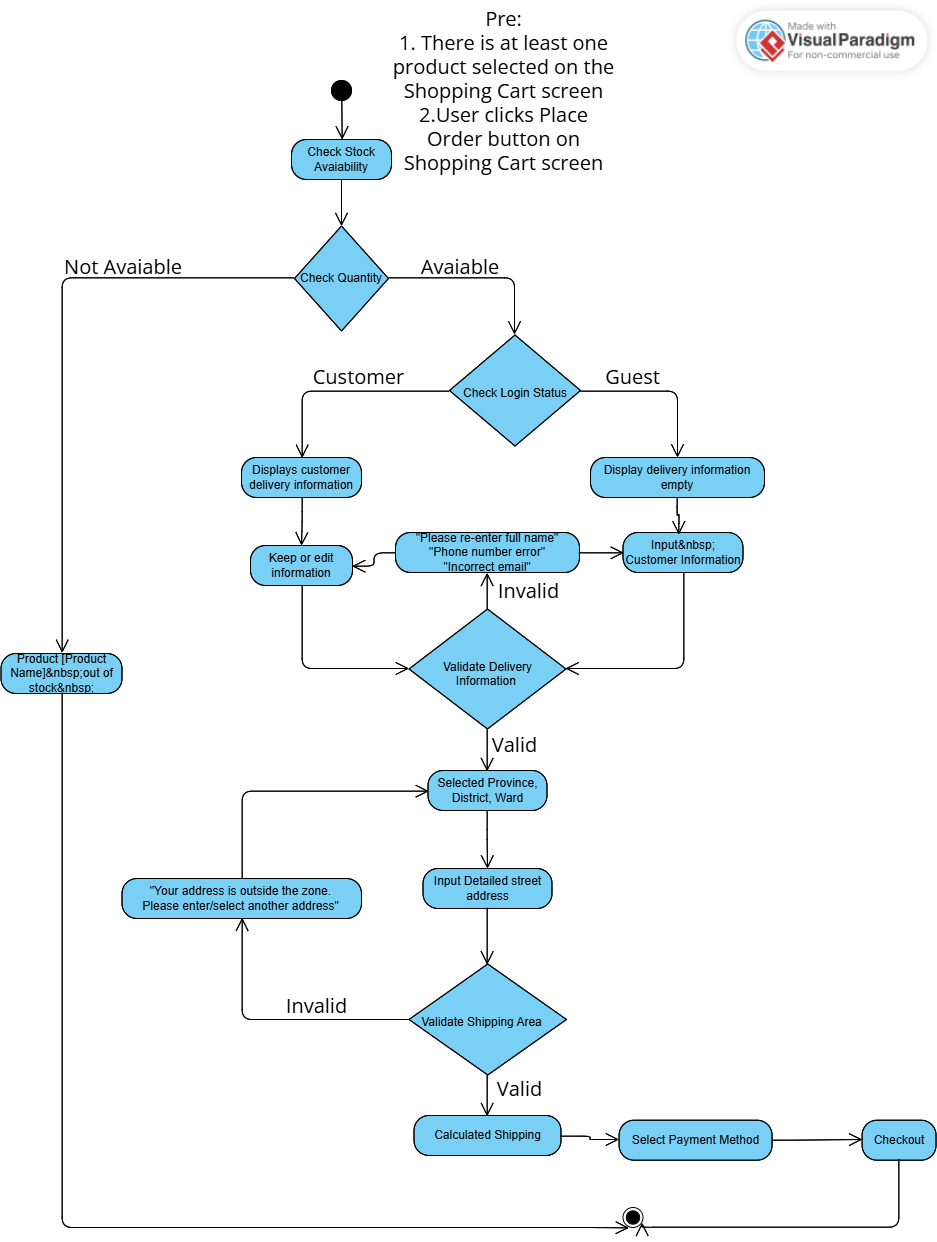
**Actors:**

* **Primary Actor**: Guest, Customer
* **Secondary Actor**: GHN (External shipping fee calculation system)

**Preconditions:**

* At least one product is selected on the Cart screen
* The user clicks the Order button on the Cart screen, either on the product list page or the product details page, clicking the "Buy Now" button
* Products in stock

**Activity Diagram**:

****

#### 2.3.4. Checkout Order (Guest, Customer)

**Summary:**

The user initiates the checkout process after adding items to their shopping cart. The system facilitates the order completion process by collecting necessary information such as shipping details, payment method, and order confirmation. A guest or customer can proceed with the checkout, where they either log in or proceed as a guest. After completing the checkout, an order is created, and the customer receives an order confirmation

**Dependency:**

* Product Availability (Ensure that the items in the cart are available in stock)
* User Authentication (if the user is logged in as Customer)
* Payment Gateway (for processing payments)
* Shipping System (for calculating shipping costs and delivery options)

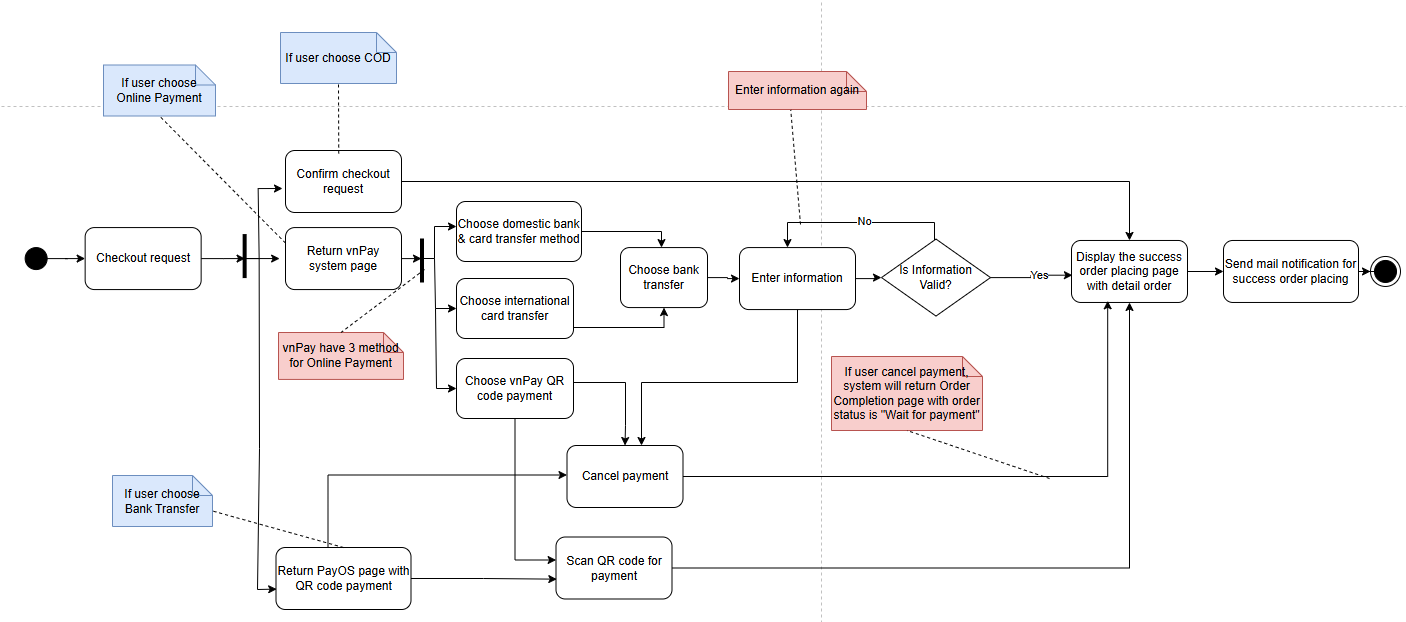
**Actors:**

* Guest: A user who browses the system without logging in, completing checkout without an account.
* Customer: A user who has an account, logs in, and completes the checkout with their saved information (address, payment method, etc.).

**Preconditions:**

* The user has added items to their shopping cart.
* If the user is a guest, they have selected the "Checkout as Guest" option. If the user is a registered customer, they have logged in with their credentials.
* The shopping cart contains valid products available for purchase.
* The user has selected a payment method.
* The shipping address is either entered by the user or retrieved from the customer’s account if logged in.
* The system verifies that all items in the cart are available and the payment gateway is functional.

**Activity Diagram**:



#### 

#### 

#### 

#### 2.3.5. Track Order Details (Customer, Sales)

**Summary:** This use case describes the process of a customer or sales department querying the details of a specific order including delivery progress if any. This allows changing the details of the submitted order such as recipient information, product quantity. This use case also allows the sales department to update the order status with notes.

**Dependency:**

* Identity of the Customer or sales department when accessing order information.
* Order is in shipping status.

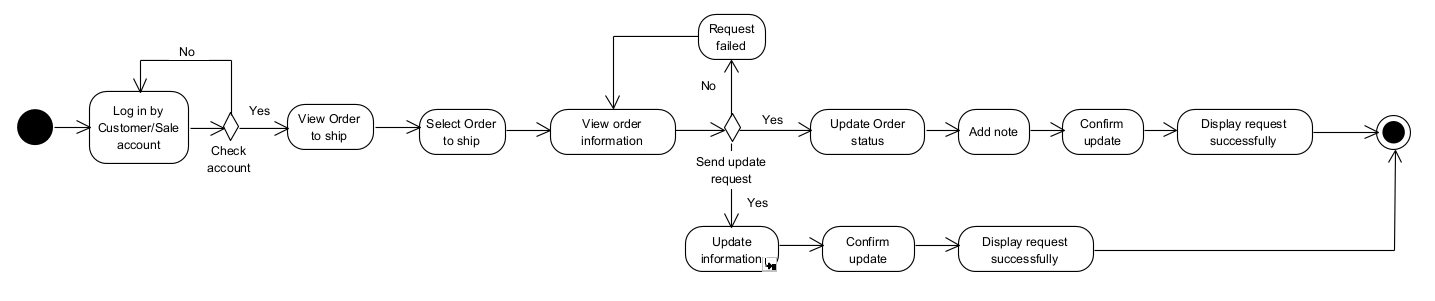
**Actors:**

* **Primary Actor**: Customer, Sales
* **Secondary Actor**: GHN

**Preconditions:**

* Sales staff need to log in to the system to have access to order information.
* The system must have information about the order being queried.
* Sales can only change order information if the order is not completed or canceled.
* The system needs to connect and query data from GHN to get complete information.

**Activity Diagram**:



## 3. Use Case Specifications

#### 

#### 3.1.1 Login System

#### 

| Primary Actors | Customer, Guest, Admin, Sale | Secondary Actors | Google Service |
| --- | --- | --- | --- |
| Description | This use case describes how a Guest, Customer, Sale, or Admin logs into the system. Two common login methods are provided:  • Traditional Login: Using a username and password.  • Google Login: Redirecting to Google for OAuth authentication and returning to the system. | | |
| Preconditions | 1. The user is on the login page (or has clicked a “Login” button/link).  2. The system is operational and connected to the internet.  3. Google Service integration (if using Google login) is properly configured. | | |
| Postconditions | 1. Successful Login: The user is authenticated and redirected to a page/functionality appropriate to their role.  2. Failed Login: The system displays an error message (e.g., incorrect credentials, Google authentication failed). | | |
| Main Sequence | 1. Actor selects the “Login” functionality.  2. System displays login options:   * Traditional Login (username/password). * Google Login.   3. Actor selects login method:   * Traditional Login → Enters username/password. * Google Login → The system redirects the user to Google OAuth.   4. System processes authentication:  Traditional Login:   * The system checks the username and password. * If valid, a session/token is created. * If invalid, an error message is shown.   Google Login:   * The system redirects to Google OAuth. * User approves access. * Google returns an authentication token. * The system verifies the token and retrieves user data. * If the user exists in the system → Login successful. * If the user does NOT exist → System auto-registers them with Google data. * A session/token is created.   5. Actors are redirected to their respective dashboard/homepage. | | |
| Alternative Sequences | • Invalid username or password (traditional login):  • System shows an error message: “Incorrect credentials.”  • Actor retries or selects “Forgot Password.”  • Google authentication fails:   * Google or System displays an error (e.g., invalid token, user denied authorization). * Actor is prompted to retry or use an alternative method. | | |

#### 3.1.2 Add New Product

#### 

| Primary Actors | System Admin | Secondary Actors | None |
| --- | --- | --- | --- |
| Description | As a System Admin, I want to add a new product to the system so that it is available for customers to view, purchase, or manage. The System Admin manages user permissions, while Sales staff performs the actual addition of products. | | |
| Preconditions | -The System Admin must be logged into the system with the correct role/permissions.  -Product details (name, price, description, etc.) must be provided by the System Admin team. | | |
| Postconditions | -The new product is added to the system and available for customers to view and purchase. | | |
| Main Sequence | -The System Admin logs into the system.  -The System Admin navigates to the Product Management section and selects "Add New Product."  -The System Admin enters the required product details (name, description, price, etc.).  -The System Admin uploads product images, which are automatically stored in Cloudinary.  -The System Admin submits the new product information.  -The System adds the product to the database and marks it as published or available for customers. | | |
| Alternative Sequences | -Missing Information: If any required fields (name, description, price) are missing, the System will prompt the Sales user to fill in the missing details.  -Image Upload Failure: If the image fails to upload to Cloudinary, the System notifies the user of the failure and allows the user to retry. | | |

#### 3.1.3 Confirm Order

#### 

| Primary Actors | Guest, Customer | Secondary Actors | GHN |
| --- | --- | --- | --- |
| Description | The use case allows Customers and Guests to view orders before payment, provide shipping information, select payment method and complete the order by integrating with a delivery service (GHN) and selecting a payment method (VNPay, Casso/PayOS). | | |
| Preconditions | * At least one product is selected on the Cart screen * The user clicks the Order button on the Cart screen, either on the product list page or the product details page, clicking the "Buy Now" button * Products in stock | | |
| Postconditions | * The order is transferred to the payment page and the order is successfully created and stored in the system. * The user receives an order confirmation. * The order is marked for processing and completion. | | |
| Main Sequence | -**The system** checks product availability before displaying the **Order Confirmation page**:   * If any product is out of stock, **the system notifies the customer** and prevents the order from proceeding. * Otherwise, the system displays:   + Selected product(s) with **image, name, category, quantity, and price**   + **Stock status** of each product   + **Individual amount** and **total amount**   **-The customer** fills in the **delivery information**, including:   * **Full name** * **Email address** * **Mobile phone number** * **City/Province** (drop-down selection) * **District** (drop-down selection) * **Ward** (drop-down selection) * **Detailed street address** * **Delivery notes** (optional)   **-The system** calculates and displays:   * **Subtotal** * **Shipping fee** * **Total cost**   **-The customer** selects a **payment method** from the available options:   * **Cash on delivery** * **Online banking** * **Bank transfer**   **-The customer** reviews all the information and clicks **"Confirm"** to finalize the order.  **-The system** processes the order and:   * **Validates the input data** * **Saves the order details** in the database * **Updates stock levels**   **-The system** displays a **confirmation message** with order details and estimated delivery time. | | |
| Alternative Sequences | **A1: Product Out of Stock**  **1a**. The system checks product availability before displaying the Order Confirmation page.  **2a**. If any selected product is out of stock, the system:   * Displays a notification informing the customer that the product is no longer available. * Prevents the customer from proceeding with the order. * Suggests alternative actions, such as:   + Removing the out-of-stock item and proceeding with the remaining items.   + Browsing for similar products.   **3a**. If the customer removes or replaces the unavailable product, the system updates the order details and proceeds with the main sequence.  **A2: Invalid Receiver Information**  **1a**. The system detects invalid receiver details (e.g., missing required fields, incorrect format).  **2a**. The system displays an error message and highlights the incorrect fields.  **3a**. The customer corrects the information and re-submits.  **4a**. The system revalidates the information and proceeds with the main sequence.  **A3: Shipping Fee Not Available**  **1a**. The system requests a shipping fee from GHN but fails to retrieve a response.  **2a**. The system alerts the customer about the issue.  **3a**. The customer chooses one of the following options:   * Enter a manual shipping fee (if allowed). * Select an alternative shipping method.   **4a**. The system updates the total cost and proceeds with the main sequence. | | |
|  |  | | |

#### 3.1.4 Checkout Order

| Primary Actors | Guest, Customer | Secondary Actors |  |
| --- | --- | --- | --- |
| Description | The user proceeds to checkout after adding items to their shopping cart. The system requests the user to enter necessary information (shipping address, payment method) and completes the order. The user can check out as a guest or login to use their saved details. After the checkout is completed, an order is created, and the user receives an order confirmation notification. | | |
| Preconditions | * User access to website * The user has added items to the shopping cart. * If the user is a guest, they have selected "Checkout as Guest". If the user is a registered customer, they have logged into the system. * The shopping cart contains valid and in-stock products. * The user has selected a payment method (credit card, e-wallet, bank transfer, etc.). * The shipping address is either entered by the guest user or retrieved from the customer’s account if logged in. * The system verifies the availability of products in the cart and confirms payment options. | | |
| Postconditions | * An order is created and confirmed. * The user receives an email notification or an on-screen confirmation of the order being processed successfully. * The payment transaction is recorded, and the payment is completed (if applicable). * The order is sent to the processing and shipping department. | | |
| Main Sequence | **User choose COD method**   1. The user selects "Checkout" with COD. 2. The user confirms on the system using the COD method. 3. The system redirects the user to the Order Completion page with detailed information about the placed order and its status. 4. The system sends a confirmation email to the user notifying them of the successful order placement. | | |
| Alternative Sequences | **Alternative 1: User choose Online Payment (vnPay)**   1. The user selects "Checkout" with Online Payment (vnPay). 2. The system redirects the user's page to the vnPay payment page. 3. The user selects the payment method from vnPay. 4. The user enters their account number, bank code, etc., and proceeds with the payment. 5. If the user enters incorrect information, they must re-enter it. 6. If the user enters the correct information, vnPay returns to the Order Completion page of the OSS system. 7. The system sends a confirmation email to the user notifying them of the successful order placement.   **Alternative 2: User choose Banking Transfer (QR code by PayOS)**   1. The user selects "Checkout" with Banking Transfer (QR code by PayOS). 2. The system redirects the user's page to the PayOS payment page. 3. The user scans the QR code displayed on the PayOS page. 4. The user successfully scans the code, and PayOS redirects the user to the Order Completion page. 5. The system sends a confirmation email to the user notifying them of the successful order placement.   **Alternative 3: User Cancels Payment**   1. The user selects "Checkout" with Banking Transfer or Online Payment. 2. The system redirects the user's page to the PayOS or vnPay payment page. 3. The user cancels the payment during the process. 4. The system redirects the user's page to the Order Completion page with the order status as "Pending Payment." 5. The system sends a confirmation email to the user notifying them of the successful order placement. | | |

#### 

#### 3.1.5 Track Order Details

#### 

| Primary Actors | Customer, Sales | Secondary Actors | GHN |
| --- | --- | --- | --- |
| Description | The customer or Sales Department accesses information about the order submitted for delivery to the external shipping service (GHN). The Sales Department updates the order status and adds notes to the order. The GHN service then processes and is responsible for delivering the order to the customer. | | |
| Preconditions | User is logged in and granted access.List of Orders exists in the system.The system has integrated API with GHN. | | |
| Postconditions | 1. View order information  * Order information is displayed correctly (product name, quantity, price, order status,...). * If the order has been delivered to GHN, the delivery status is updated from the GHN API.  1. Update order information: Order status is updated in Order Database. | | |
| Main Sequence | **Main Sequence:**  1. User logs into the system.  2. User accesses the order list.  3. User selects an order to view.  4. User accesses delivery information for each order: Recipient name, delivery address, phone number, product, payment method, order status, notes  5. User updates information of the submitted order  6. User updates the status of the order with notes  7. Click the Save Changes button to confirm the change.  8. The system sends a request to GHN and receives a response from GHN  9. The system updates the order and sends a success notification to the user | | |
| Alternative Sequences | **A1: User does not have permission to query orders**   1. The system checks the user's access rights. 2. If no rights are granted, the system displays an error message: "You do not have access to this order." 3. The system may suggest that the user contact support if necessary.   **A2: Order does not exist in the system**   1. The system searches for the Order ID in the Order Database. 2. If it is not found, the system displays the message: "The order does not exist. Please check the order code again." 3. The user re-enters the Order ID to display the correct order   **A3: Order has been cancelled or completed, cannot be edited**   1. User requests to edit order. 2. The system checks order status in the Order Database. 3. If the order has a status of "Canceled" or "Completed", no updates are allowed. 4. Displays a message: "Order is completed or canceled, cannot be edited."   **A4: Error connecting system to GHN API**   1. User requests to view order details. 2. The system sends a request to GHN API to get delivery information. 3. The GHN API does not respond or returns an error. 4. The system reports an error: "Unable to get delivery information at this time, please try again later". | | |

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |
|  |  | | |

## 4. Non-Functional Requirements

### 4.1 External Interfaces

***User Interface (UI):***

*-The system provides an intuitive and easy-to-navigate user interface for customers, allowing them to browse products, place orders, and track delivery status.*

*-The UI will be designed to be responsive, ensuring compatibility across different devices such as desktops, tablets, and smartphones.*

*-User inputs such as order details, payment information, and delivery preferences will be captured through forms and buttons in the UI.*

***Software Interfaces:***

*-The system will integrate with existing backend databases to retrieve product information, order status, and customer data.*

*-The system will communicate with payment gateways via API for processing customer transactions securely.*

*-A shipping and delivery service interface will be implemented to send order details for fulfillment and tracking. This may involve third-party logistics providers with their own API systems.*

***Hardware Interfaces:***

*-The system may interface with hardware devices such as barcode scanners or point-of-sale (POS) terminals for inventory management and order processing in the shop.*

*-Devices such as receipt printers may also be connected for order confirmation and invoices.*

***Network Interfaces:***

*-The system will utilize secure internet connections for communicating with external services, including payment gateways and delivery tracking systems.*

*-A reliable network infrastructure will ensure smooth communication between the Online Shopping System and external entities, such as shops with multiple branches and external logistics services.*

*-For future releases, the system will be designed to interface with internet ordering services, allowing seamless integration with various online shopping platforms.*

### 4.2 Quality Attributes

#### 4.2.1 Usability

**User-Centric Design:**

-The system will feature a clean, intuitive interface that is easy to navigate, even for users with minimal technical experience.

-The layout will be consistent across different screens, providing clear instructions and visual cues to guide users through the process of browsing, ordering, and managing their accounts.

**Accessibility:**

-The system will be designed to be accessible to a wide range of users, including those with disabilities. Features like screen reader compatibility, keyboard navigation, and high-contrast visual design will be implemented to enhance accessibility.

-The platform will comply with relevant accessibility standards, ensuring that it meets or exceeds requirements for users with various needs.

**Responsiveness:**

-The system will be fully responsive, ensuring a seamless experience across devices, including desktops, tablets, and mobile phones.

-The design will adapt to different screen sizes, providing users with an optimal viewing and interaction experience regardless of their device.

**Ease of Use:**

-The process of browsing products, adding items to the cart, and completing a purchase will be straightforward and user-friendly. Clear navigation paths and simplified forms will minimize user effort.

*-Features like search functionality, product filters, and order tracking will be easy to use, allowing customers to quickly find products and check the status of their orders.*

***Error Prevention and Feedback:***

*-The system will incorporate error prevention mechanisms to guide users toward correct actions and avoid common mistakes, such as incorrect payment details or missing information during order placement.*

*-Informative and helpful feedback will be provided in real-time, including error messages, confirmation alerts, and progress indicators, to keep users informed throughout their interactions with the system.*

***User Support:***

*-A help section or FAQ will be accessible from the system, providing users with answers to common questions and troubleshooting tips.*

*-For more complex issues, users will have the option to contact customer support through various channels, such as live chat or email, ensuring timely assistance.*

#### 4.2.2 Performance

1. **Response Time**
   * Average:
     + *Search Products*: Results displayed within **2 seconds** under normal load (100 concurrent users).
     + *Checkout Order*: Checkout screen loads in **under 3 seconds**.
   * Maximum: Response time for critical tasks does not exceed **5 seconds** under high load (500-1,000 concurrent users).
2. **Throughput**
   * Handles **50 payment transactions/second** under normal conditions.
   * Scales to **200 transactions/second** during peak events (e.g., sales campaigns).
3. **Capacity**
   * Supports up to **10,000 concurrent users**, with **3,000 active checkouts** simultaneously.
   * Database designed for **1 million products** and **10 million transactions annually**.
4. **Resource Utilization**
   * **CPU/Memory**: Utilization below 70% under normal conditions.
   * **Disk I/O**: No bottlenecks for log writing or order updates.
   * **Network Latency**: Interactions with external services (e.g., Payment System, shipping) have latency under **200ms**.

#### 4.2.3 Scalability

1. **Horizontal Scaling**
   * System supports **adding instances** for application servers with automatic load balancing to handle increased traffic.
2. **Vertical Scaling**
   * Resources like **CPU, RAM, and SSD** can be upgraded to support larger data and transaction volumes.
3. **Database Scaling**
   * Implements **Primary-Replica clustering** for scalability.
   * Includes a **caching layer** (e.g., Redis) to reduce database load for frequent queries.
4. **Cloud-based Scalability**
   * Uses cloud services for storage, payment processing, and shipping to dynamically scale based on traffic.
5. **Monitoring and Auto-Scaling**
   * Monitors resource usage in real-time and triggers automatic scaling (scale out/in) during traffic spikes (e.g., flash sales).

#### 4.2.4 Portability

* The system must operate smoothly on all popular web browsers such as Chrome, Firefox, Safari, Edge with versions released within the last 2 years.
* The website must have a responsive design, compatible and displaying well on mobile devices (phones, tablets) with different screen sizes.
* Page loading time must be fast and stable on both 3G/4G and wifi connections.
* The system must allow data export/import in common formats such as CSV, Excel for easy data migration.

#### 4.2.5 Compatibility

* The system must be compatible with popular payment gateways in Vietnam such as VNPay, Momo, ZaloPay and domestic bank cards.
* The system's API must comply with RESTful standards and support common data formats such as JSON, XML.
* The system must be able to integrate with third-party services such as shipping providers (Giao Hang Nhanh, Giao Hang Tiet Kiem), email marketing systems, and analytics tools.
* Data must be stored in UTF-8 standard to fully support Vietnamese and special characters.
* The system must be compatible with popular accounting/inventory management software for easy data synchronization.

#### 4.2.6 Reliability

#### 4.2.6.1 High Availability

* The system must be available 24/7 so that customers can search for products, place orders, pay, and track delivery status at any time.
* Uptime must be at least 98% per year. This equates to a maximum of 8.76 hours of downtime per year.

***4.2.6.2 Fault Tolerance***

* The system must be able to continue operating even if part of it fails.
* Redundancy mechanisms should be implemented to protect data and service operations in the event of hardware, network, or software failures. If the main server is interrupted, the system must automatically switch to the backup server without affecting customers.

***4.2.6.3 Data Accuracy***

* Ensure that transactions, order information, payment and shipping status are processed accurately.
* Zero errors in checkout processes, order status updates, or shipping calculations.
* If the customer pays via PayOs or VnPay, the "paid" status must be updated correctly in the system.

#### 4.2.7 Maintainability

***4.2.7.1 Modular Design***

* The system should be designed in such a way that it is divided into independent modules, each module handling a specific function.
* This makes it easy to change or upgrade individual modules without affecting other parts.
* The Payment module can be upgraded to integrate new methods such as Momo without affecting the Order module.

***4.2.7.2 Comprehensive Documentation***

* The entire system needs to be fully documented, including:
* **Architecture document**: Describes the system design, relationships between modules.
* **Source Code Documentation**: Notes explaining each section of the source code.
* **User Manual**: Helps maintenance staff understand how to operate the system.
* If the delivery management system needs to be upgraded, documentation on API communication with GHN will help the maintenance team understand and make changes easily.

***4.2.7.3 Scalable Architecture***

* The system needs to be designed to easily add new features or change processes without having to modify the entire system.
* When adding a new payment provider, simply extend the payment module without modifying the entire system.

***4.2.7.4 Automated Testing***

* Use automated testing to ensure that changes don't break existing features.
* Have at least 80% of source code covered by automated tests.
* Use unit tests to test individual functions, integration tests to test interactions between modules.

#### 4.2.8 Availability

#### 4.2.8.1. Uptime Guarantee

* The system must be available 99.9% of the time, which allows for only 8.76 hours of downtime per year.

#### 4.2.8.2. Redundancy for Critical Systems

* The system must implement redundancy for critical services like:
  + Payment processing.
  + Book catalog browsing and search.
  + Customer account management.

#### 4.2.8.3. Maintenance Windows

* Planned maintenance should be conducted during non-peak hours with prior user notification.

#### 4.2.8.4. Real-Time Health Monitoring

* The system must have real-time monitoring tools to detect failures and performance bottlenecks.

#### 4.2.9 Security

#### 4.2.9.1. User Authentication and Authorization

* The system must ensure only authorized users can access restricted areas (e.g., admin dashboards).

#### 4.2.9.2. Data Encryption

* All sensitive data must be encrypted both in transit and at rest.

#### 4.2.9.3. Secure Payment Processing

* The system must comply with the **Payment Card Industry Data Security Standard (PCI DSS)**.

#### 

#### 