HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

School of Information and communications technology

Software Requirement Specification

AIMS – An Internet Media Store

Subject: ITSS Software Development

Group 13

|  |  |
| --- | --- |
| Trương Công Thái Đức | 20225815 |
| Vũ Ngọc Lâm | 20225645 |
| Phan Sỹ Hùng | 20225631 |
| Nguyễn Mạnh Tuấn | 20225679 |

*Hanoi, March* *2025*

**Contents**

[1 Introduction 2](#_Toc2099208049)

[1.1 Objective 3](#_Toc813535418)

[1.2 Scope 3](#_Toc565613051)

[1.3 Glossary 4](#_Toc1419196569)

[1.4 References 4](#_Toc876943683)

[2 Overall Description 4](#_Toc1126759197)

[2.1 Survey 5](#_Toc1513884540)

[2.2 Overall requirements 5](#_Toc1448689389)

[2.3 Business process 6](#_Toc770440277)

[2.3.1 Business operation - “Place Order” 6](#_Toc1808778648)

[2.3.2 Business operation - “Manage products” 7](#_Toc1005555274)

[2.3.3 Business operation - “Manage users” 8](#_Toc631277440)

[3 Detailed Requirements 9](#_Toc663405141)

[3.1 Use case “Place Order” 10](#_Toc479580060)

[3.2 Use case “Login” 12](#_Toc474676971)

[3.3 Use case “View Cart” 13](#_Toc2005822651)

[3.4 Use case “Manage products in cart” 15](#_Toc1649343055)

[3.5 Use case “Search Product” 18](#_Toc1938472854)

[3.6 Use case “Add Users” 20](#_Toc1715672497)

[3.7 Use case “Update Users” 22](#_Toc1852616724)

[3.8 Use case “Add Product” 23](#_Toc2116557376)

[3.9 Use case “Update Product” 25](#_Toc1378651775)

[3.10 Use case “Pay order” 26](#_Toc325222209)

[4 Supplementary specification 28](#_Toc2027958183)

[4.1 Functionality 29](#_Toc682854572)

[4.2 Usability 29](#_Toc1438113035)

[4.3 Reliability 29](#_Toc280010413)

[4.4 Performance 29](#_Toc1212360467)

[4.5 Supportability 29](#_Toc1667548013)

[4.6 Other requirements 29](#_Toc1132994166)

# Introduction

This SRS document serves as a comprehensive guide outlining the requirements, features, and constraints of the software project. It is intended to provide a clear understanding of the scope and objectives of the project for all stakeholders involved, including clients, users, and development team members.

## Objective

The primary purpose of this document is to establish a common understanding of the software requirements among all stakeholders. It defines the functionalities, features, and performance expectations of the software system to be developed. By documenting these requirements in detail, this document serves as a reference point throughout the software development lifecycle, guiding the design, implementation, testing, and deployment phases. The intended audience includes project stakeholders, development teams, system architects, UI/UX designers, quality assurance testers, end-user representatives, and regulatory officers, all of whom rely on this document to ensure the successful realization of AIMS.

## Scope

This AIMS – ‘An Internet Media Store’ software is developed to be a desktop platform e-commerce software, which helps users to order media products on the Internet, and the store managers, at the same time, are easier to manage their store as well as the orders.

This software can serve up to 1,000 customers simultaneously without significantly reducing performance and can operate continuously for 300 hours without failure. Additionally, the software can resume normal operation within a maximum of 1 hour after an incident. The maximum response time of the software is 2 seconds under normal conditions or 5 seconds during peak hours.

In AIMS, customers can not only search for products, but also sort products as they desire, they can place order or rush order for necessary cases. AIMS is supported by VNPay transactions; thus, customers can easily pay for their order. Moreover, customers can review their order and modify any information during the processing order stage. While shopkeepers can many their store by managing products directly in the system. They, meanwhile, can process the orders of the customers. For administrators, they are capable of managing users and privileges problems of users.

Additionally, for a desktop website, the graphical user interface (GUI) is carefully considered to meet end-user requirements and enhance the overall user experience. Throughout the development process, all documentation is systematically recorded to facilitate future maintenance and upgrades. We maintain a strong focus on each stage, ensuring adherence to the client’s timeline and the delivery of high-quality software. If any modifications are required, our team swiftly adapts to revise and refine our work accordingly.

## Glossary

| ***No*** | ***Term*** | ***Explanation*** | ***Example*** | ***Note*** |
| --- | --- | --- | --- | --- |
| 1 | Session | a session is a temporary interaction period between a user and the software, starting when the application is launched and ending when it is closed | Shopping Cart Session |  |
| 2 | VAT  (Value-added tax) | A **10% tax** applied to the product price, which is not included in the base value set by the product manager |  |  |
| 3 | API (Application Programming Protocol | API is a set of rules, protocols, and tools that allows different software apps to communicate with each other. |  |  |
| 4 | VNPay | An online payment gateway integrated with AIMS for online transaction |  |  |
| 5 | Shipping fees | Costs associated with delivering a product, calculated based on weight and location |  |  |

## References

# Overall Description

**AIMS Project** is a desktop e-commerce software that operates 24/7, allowing new users to easily familiarize themselves. The software can serve up to 1,000 customers simultaneously without a significant decrease in performance and can operate continuously for 300 hours without failure. In the event of an incident, the software can resume normal operation within a maximum of 1 hour. The maximum response time of the software is 2 seconds under normal conditions and 5 seconds during peak hours.

AIMS software supports transactions only for physical media products (books, CDs, LP records, and DVDs), and requires information such as barcode, product description, quantity, warehouse entry date, product dimensions, and weight. The software allows product managers to add, view, edit, or delete products and stores a history of product addition, editing, and deletion operations. It will notify the product manager if any operation is invalid.

## Survey

**Software Overview**:

AIMS is a desktop e-commerce software that supports the management of media products like books, CDs, LP records, and DVDs. It allows product managers to add, edit, and delete products from the database. Customers are able to view products, order products and pay orders through VNPays.

**Actors**:

* **Product Manager**: Has the rights to add, edit, and delete products in the system.
* **Administrator**: Manages user accounts, can create, edit, delete users, and assign roles to users.
* **Customer**: Searches for products, views product details, adds products to the shopping cart, and proceeds with the checkout process.
* **VNPay:** facilitates online transactions

## Overall requirements

The figure below shows the general use case diagram of AIMS software, which includes the actors and use cases that are involved in the systems.

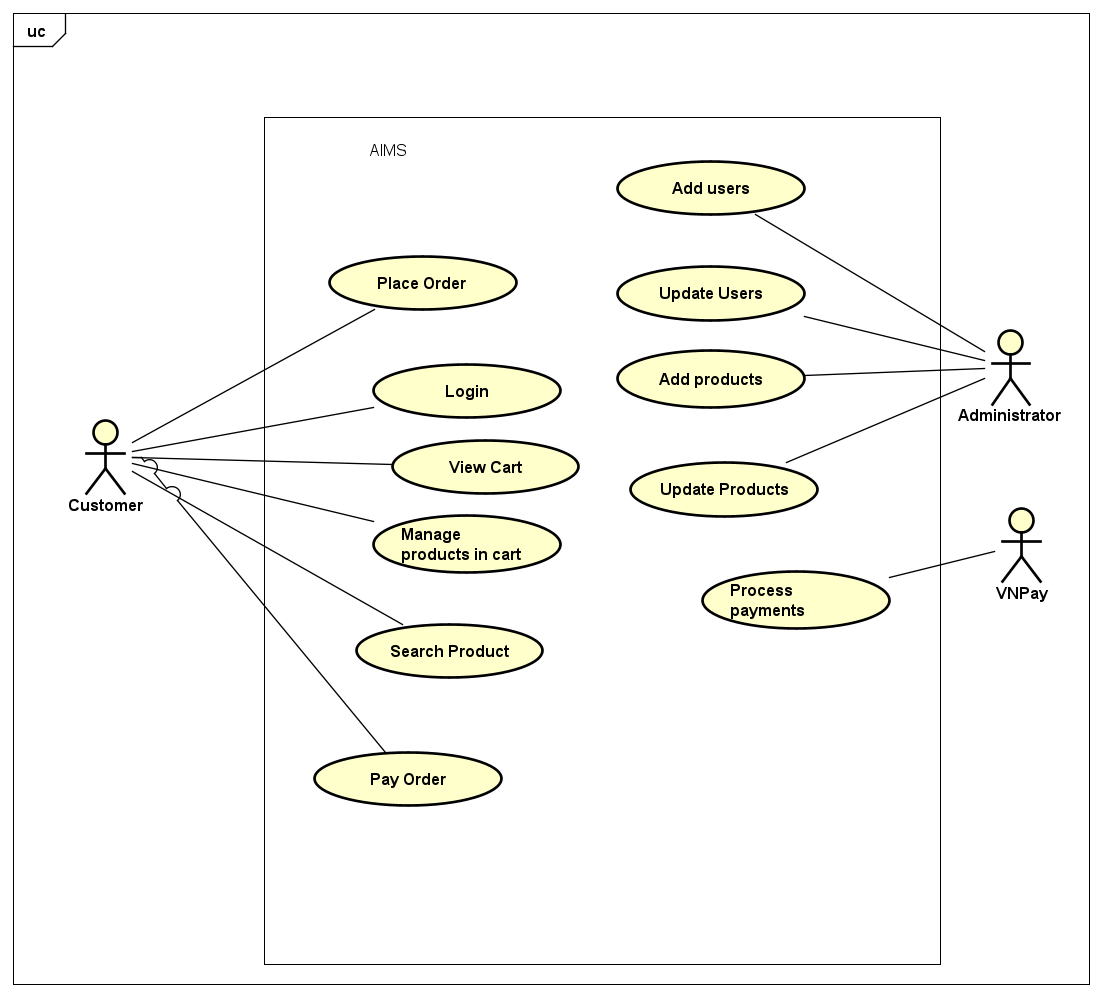
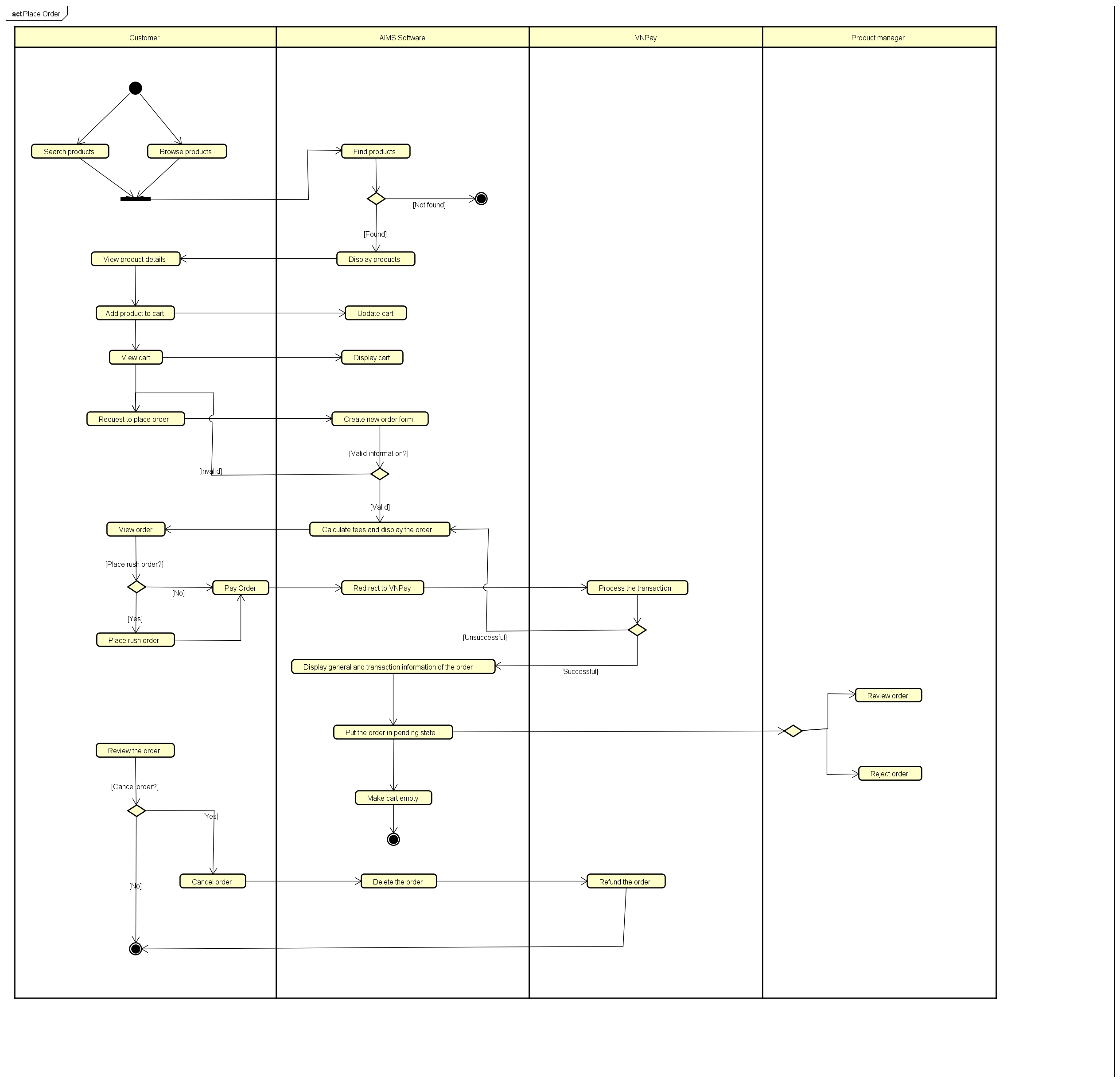


Figure 1. General use case diagram

## Business process

In AIMS software, there are three main business operations: business operation – “Place order” by the customer, business operation – “Mange products” by product managers and business operation “Manage users” by administrator. The details of each business process are described by an activity diagram in each section of each one.

### Business operation - “Place Order”

Figure 2. Business process - ”Place Order”

### Business operation - “Manage products”

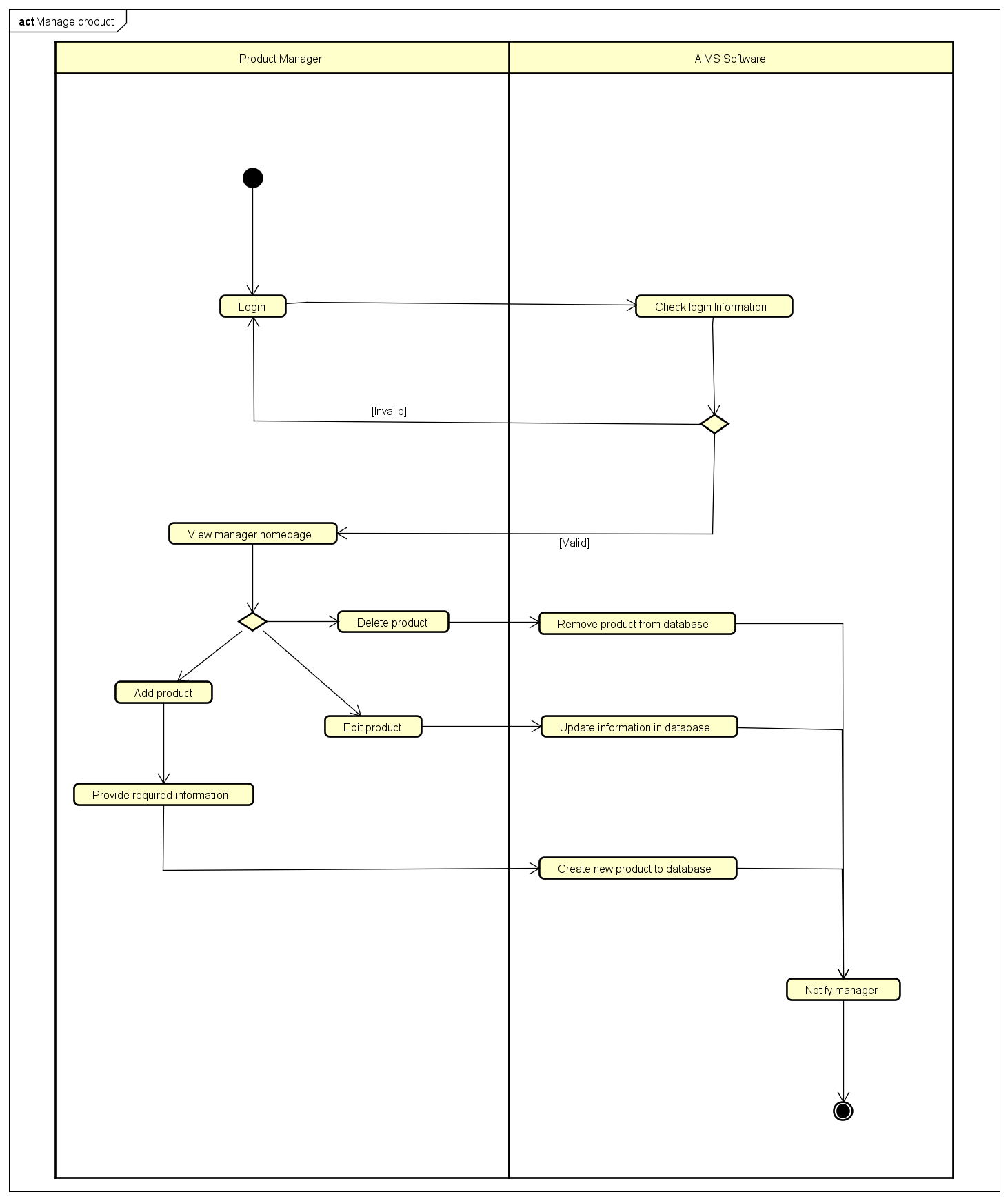


Figure 3. Business process - “Manage products”

### Business operation - “Manage users”

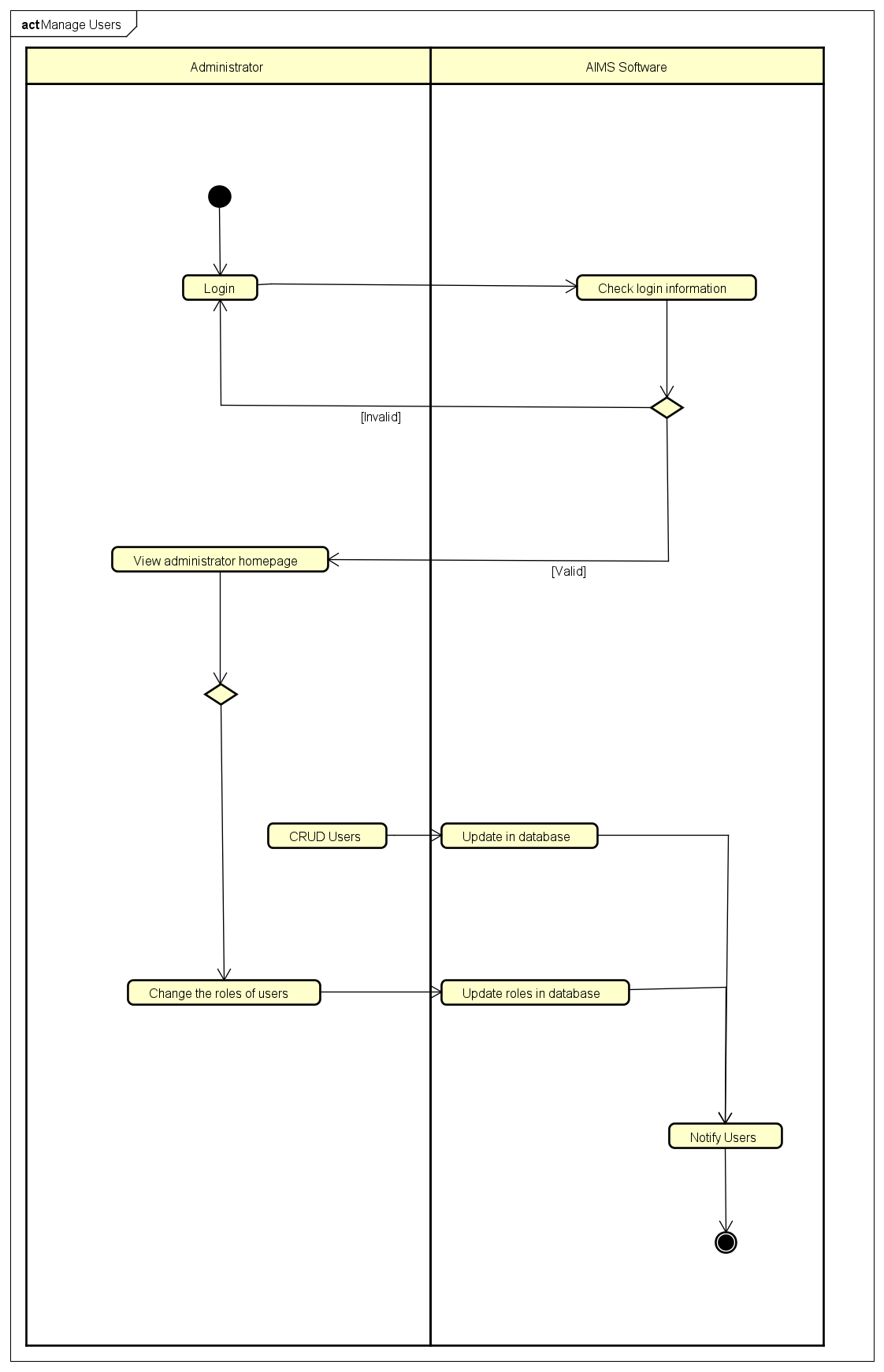


Figure 4. Business process - “Manage users”

# Detailed Requirements

## Use case “Place Order”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use case “Place Order”**  **1. Use case code** UC001  **2. Brief Description**  This use case describes the interaction between customers and system. Customers have to fill in some information to place a successful order.  **3. Actors:** Customers  **4. Preconditions**   * The quantity of products is sufficient * Customers need to provide several delivery information * The AIMS software connects to VNPay for further payment   **5. Basic Flow of Events**  • **Step 1.** Customer requests to place order in the cart  • **Step 2.** AIMS software checks the availability of products in the cart  • **Step 3.** AIMS software displays the form of delivery information with order information  • **Step 4.** Customer enters and submits delivery information  • **Step 5.** AIMS software calculates and updates order information with shipping fees  • **Step 6.** The customer asks to pay order  • **Step 7.** **The AIMS software calls UC “Pay order”**  • **Step 8.** The AIMS software creates and saves a new order  • **Step 9.** The AIMS software makes the cart empty  • **Step 10.** The AIMS software sends email about the order notification and information  • **Step 11.** The AIMS software displays the successful order notification, the order and the transaction information  **6. Alternativeflows**  **Table A - Alternative flow of events for UC “Place Order”**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Ressume Location** | | 1 | At step 2 | If the products are not available | The Aims software notifies that the products in the cart are not available and stay at the UC “View cart” | Use case ends | | 2 | At step 4 | If the delivery info is invalid | AIMS software notifies that the delivery info is invalid | At step 3 | | 3 | At step 5 | If the user chooes to place rush order | Aims software inserts UC “place rush order” | At step 6 | | 4 | At step 7 | If the order payment is not successful or goes back from payment |  | At step 6 |   **7. Input Data**   * For delivery information:  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | No | Data fields | Description | Mandatory | Valid Condition | Example | | 1 | Receipient name |  | Yes |  | Phan Sy Hung | | 2 | Email |  | Yes |  | [hung05324@gmail.com](mailto:hung05324@gmail.com) | | 3 | Phone number |  | Yes | 10 digits | 0357257667 | | 4 | Province/City |  | Yes |  | Hanoi | | 5 | Delivery address |  | Yes |  |  |   **8. Output Data**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Data Fields** | **Description** | **Display Format** | **Example** | | 1 | Title | Title of media product |  | DVD phim | | 2 | Price | Price of the product | * Comma for thousands separtor * Positive Interger | 123.000 | |

## Use case “Login”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Login”**  **1. Use Case Code** UC002  **2. Brief Description**  This use case describes the interaction between Customers and Aims System  **3. Actors:** Customers, System  **4. Preconditions**   * The customer has an active account * The customer is on the login page of web   **5. Basic Flow of Events**   * **Step 1.** **Customer** accesses the login page. * **Step 2.** **AIMS software** prompts the **Customer** to enter their username/email and password * **Step 3.** The **Customer** enters their username/email and password * **Step 4.** Customer clicks the Login button * **Step 5.** AIMS software validates the entered username/email and password * If the credentials are correct, the system authenticates the Customer and grants access to the dashboard or main page of the system. * If the credentials are incorrect, the system displays an error message (e.g., “Invalid username or password. Please try again.”) * **Step 6.** The system logs the Customer into the application and redirects them to the appropriate page (usually the Customer’s dashboard or homepage). * **Step 7.** The use case ends successfully with the Customer logged in.   **6. Alternative flows**   |  |  |  |  | | --- | --- | --- | --- | | **No** | **Location** | **Action** | **Resume action** | | 1 | At step 3 | User enters invalid credentials (either incorrect username/email or password) | Step 2 | | 2 | At step 3 | The Customer attempts to log in after several failed attempts | Use case ends |   **7. Input Data**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid conditions** | **Example** | | **1** | **Username** |  | **yes** |  |  | | **2** | **Password** |  | **yes** |  |  |   **8. Output data** |

## Use case “View Cart”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “View Cart”**  **1. Use Case Code** UC003  **2. Brief Description**  This Use Case describes the process in which the customer reviews the list of products in their shopping cart, including product details, quantity, price, and total payment amount.  **3. Actors** Customers  **4. Preconditions**   * The customer has added at least one product to the shopping cart. * The shopping cart data is stored in the system.   **5. Basic Flow of Events**   * **Step 1**: The customer accesses the shopping cart. * **Step 2**: AIMS displays the list of products in the cart. * **Step 3**: The customer can change the product quantity or remove products from the cart. * **Step 4**: AIMS updates the total order value. * **Step 5**: The customer can continue shopping or proceed to checkout.   **6. Alternative flows**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Condition** | **Location** | **Action** | **Resume location** | | 1 | At Step 2 | The shopping cart is empty. | AIMS displays the message "Your cart is empty." | Use Case ends. | | 2 | At Step 3 | The product is out of stock. | AIMS displays a notification and automatically removes the product from the cart. | Return to Step 2. |   **7. Input data**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid Conditions** | **Example** | | 1 | Customer ID | The unique ID of the logged-in customer. | Yes | Must be a unique identifier. | Customer1 | | 2 | Product List | List of products in the cart | Yes | Must contain at least one product or display an empty cart message | SP1 ,SP2, | | 3 | Product Quantity | The quantity of each product in the cart. | Yes | Positive integer. | 1,2,3,4,5 | | 4 | Total Price | The total order value before tax. | Yes | Positive integer. | 10,000 VND | | 5 | VAT Tax | Tax applied to the order. | Yes |  | 5,000 VND | | 6 | Product Status | Checks whether the product is still in stock. | Yes | In stock/Out of stock. | Out of stock | | 7 | Cart Update Date | The most recent time the customer modified the cart. | Yes | Format must be **dd/mm/yyyy HH:mm**. | 28/02/2025 15:30 |   **8. Output data**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Display format** | **Example** | | 1 | Product Name | The name of the product in the shopping cart. |  | Shampoo | | 2 | Price | The price of the product. | Positive integer | 123,000 | | 3 | Quantity | The quantity of the product selected by the customer. | Positive integer | 2 | | 4 | Total Price | The total value of the order before tax and fees. | Positive integer. | 246,000 | | 5 | Total Payment | The final amount the customer needs to pay (including tax and fees). |  | 500,000 | |

## Use case “Manage products in cart”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use case code: UC004****Brief Description:** This use case describes the interaction between the customer and the shopping cart system when a customer performs CRUD (Create, Read, Update, Delete) operations on products in their cart. **Actors:** Customer (end user), Shopping Cart System **Preconditions:**  * The customer must have products available to add to their cart.  **Basic Flow of Events:**  1. The customer views available products on the website. 2. The customer selects a product they wish to add to the cart. 3. The system prompts the customer to select the quantity (if applicable) and then adds the product to the cart. 4. The customer can view the cart at any time to review products added. 5. The customer can:    1. **Create**: Add a new product to the cart.    2. **Read**: View the list of products in the cart, including details such as quantity, price, and total.    3. **Update**: Modify the quantity of a product already in the cart.    4. **Delete**: Remove a product from the cart. 6. The system updates the cart accordingly after each CRUD operation. 7. The system calculates the total cost of items in the cart after each modification.  **Alternative Flows:**  |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | | 1 | At Step 3 | If the customer tries to add an out-of-stock product | The system notifies the customer that the product is out of stock and cannot be added | Return to Step 3 | | 2 | At Step 5 (Update) | If the customer tries to update quantity to a negative value | The system reports an error and asks for a valid quantity | Return to Step 5 | | 3 | At Step 5 (Delete) | If the customer tries to delete a product that's not in the cart | The system reports an error indicating the product is not in the cart | Return to Step 5 |  **Input Data:**  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | | 1 | Product ID | The identifier of the product | Yes | The product must exist in the catalog | 12345 | | 2 | Quantity | The quantity of the product to be added or updated | Yes | Quantity must be a positive integer | 2 | | 3 | Action (CRUD) | The action to perform on the cart (Create, Update, Delete) | Yes | Must be one of "Create", "Update", or "Delete" | Create | | 4 | Cart ID (if applicable) | The identifier of the user's shopping cart | Yes | The cart ID must exist if applicable | ABC123 | |

## Use case “Search Product”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Search Product”**   1. **Use case code** UC005 2. **Brief Description**   This Use Case describes the process of a customer searching for a product in the AIMS system by entering a search keyword or filtering by category.   1. **Actors: Customers** 2. **Preconditions**  * The system has a stable internet connection to perform search queries. * The product database is updated to ensure accurate search results.  1. **Basic Flow of Events**   **Step 1:** The customer enters a keyword or selects a search filter.  **Step 2:** AIMS displays a list of products matching the keyword or filter.  **Step 3:** The customer can view product details or add items to the cart.   1. **Alternative flows**  |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume Location** | | 1 | At Step 2 | No matching products found | AIMS displays the message "No matching products found” | UC ends | | 2 | At Step 2 | Network connection error | AIMS displays an error message and asks the customer to retry | Back to Step 1 |  1. **Input data**  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid Conditions** | **Example** | | 1 | Search Keyword | Text string to search for products | No | No special characters | Dầu gội Romano | | 2 | Sort Results | Sort by price, name, rating | No | Price: Asc/Desc, Name: A-Z/Z-A, Rating: High/Low | 4/5 | | 3 | Min Price | Lowest price range of product | No | Positive integer | 5,000,000 | | 4 | Max Price | Highest price range of product | No | Positive integer | 10,000,000 | | 5 | Brand | Filter by product brand | No |  | Romano | | 6 | Stock Status | Filter by product availability | No | In stock / Out of stock | Out of stock |  1. **Output data**  |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Display format** | **Example** | | 1. | Product Name | Name matching the search keyword | Text | Máy tính DELL | | 2 | Product Price | Price of the product | Number | 11,234,500 | | 3 | Stock Status | Remaining product quantity | Integer (≥0) | 10 | | 4 | Brand | Product brand | Text | Dell | | 5 | Rating | Average product rating | Stars (1-5) | 4/5 | | 6 | Product Image | Product thumbnail image | Image | Ảnh laptop DELL | |

## Use case “Add Users”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Add users”**   1. **Use Case Code**: UC006 2. **Brief Description**:  This use case describes the process of adding a new user to the system by the Admin. The Admin provides the necessary user details, and the system creates a new user account. 3. **Actors**:  * **Admin**  1. **Preconditions**:  * The Admin must be logged in to the system.  1. **Basic Flow of Events**: 2. The Admin accesses the "Add User" page. 3. The Admin fills in the required user details, including User ID, Username, Role, Email, and Password. 4. The Admin submits the form to create the new user. 5. The system validates the user details (e.g., checking for duplicate usernames or invalid email formats). 6. If all details are valid, the system creates the new user account and stores it in the database. 7. The system displays a success message, confirming the user was added successfully. 8. **Alternative Flows**:  |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume Location** | | 1 | Step 3 (Add) | If the Admin tries to add a user with missing required information | The system notifies the Admin that the information is missing and asks to provide it | Return to Step 3 | | 2 | Step 3 (Add) | If the Admin tries to add a user with an already existing username | The system informs the Admin that the username is already taken and prompts for a new one | Return to Step 3 | | 3 | Step 3 (Add) | If the Admin tries to add a user with an invalid email format | The system reports an error indicating the email is invalid and requests a valid format | Return to Step 3 |  1. Input Data  |  |  |  |  |  | | --- | --- | --- | --- | --- | | No | Data fields | Mandatory | Valid Condition | Example | | 1 | Username | Yes | The username must be unique | test | | 2 | Role | Yes | The role must be valid (Admin, Customer, etc.) | Admin | | 3 | Email | Yes | Must be a valid email format | Demo@example.com | | 4 | Password | Yes | Password must meet security criteria (minimum length, etc.) | P@ssw0rd123 |  1. Postconditions:  * A new user is successfully added to the system. * The new user's information is stored in the user database. * The Admin can view and manage the new user. |

## Use case “Update Users”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Update Users”**   1. **Use Case Code**: UC007 2. **Brief Description**:  This use case describes the process of updating the details of an existing user in the system by the Admin. The Admin can modify user information such as username, role, email, and password. 3. **Actors**:  * **Admin**  1. **Preconditions**:  * The Admin must be logged in to the system. * The Admin must have the necessary permissions to update user information. * The user to be updated must already exist in the system. * The system must be running and accessible.  1. **Basic Flow of Events**: 2. The Admin accesses the "Manage Users" page and selects the user they want to update. 3. The system displays the current details of the selected user. 4. The Admin modifies the required fields, such as username, role, email, or password. 5. The Admin submits the updated details. 6. The system validates the new details (e.g., ensuring the username is unique and the email is valid). 7. If all details are valid, the system updates the user information in the database. 8. The system displays a success message, confirming that the user's details were successfully updated. 9. **Alternative Flows**:  |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume Location** | | 1 | Step 3 (Update) | If the Admin tries to update a user with missing required information | The system notifies the Admin that the information is missing and asks to provide it | Return to Step 3 | | 2 | Step 3 (Update) | If the Admin tries to update a user with an already existing username | The system informs the Admin that the username is already taken and prompts for a new one | Return to Step 3 | | 3 | Step 3 (Update) | If the Admin tries to update a user with an invalid email format | The system reports an error indicating the email is invalid and requests a valid format | Return to Step 3 |  1. **Input Data**:  |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Data Fields** | **Mandatory (Yes/No)** | **Valid Condition** | **Example** | | 1 | Username | No | The username must be unique | test | | 2 | Role | No | The role must be valid (Admin, Customer, etc.) | customer | | 3 | Email | No | Must be a valid email format | demo@example.com | | 4 | Password | No | If updated, the password must meet security criteria (minimum length, etc.) | 1234567 |  1. **Postconditions**:  * The user's details are successfully updated in the system. * The new user information is stored in the user database. * The Admin can view the updated user details. |

## Use case “Add Product”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Add product”**   1. **Use case code** UC008 2. **Brief Description**   This use case describes the interaction between the product manager and the product when there is a need to add new products   1. **Actors:** Admin 2. **Preconditions**   You must log in with a product manager account.   1. **Basic Flow of Events** 2. The manager requests the creation of a new product 3. The system displays a form for the manager to enter product information 4. The manager needs to select the type of product he wants to add so that the specific information form for that product will appear 5. The manager enters information 6. The system checks the validity of entered product information 7. The system notifies the successful creation of new products. 8. **Alternative flows**   **Table A-Alternative flows of events for UC “**Add product**”**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | | 1. | At Step 5 | If the user leaves the required information blank | The system reports an error and requires the manager to fill in all the information. | Continue at step 4 |  1. **Input data**   **Table A-Input data** of UC “Add product”   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | | 1. | Title | Title of a media product | Yes |  | Sample DVD | | 2. | Category |  | Yes |  |  | | 3. | Price | Price of the corresponding media product | Yes |  | 20.000 | | 4. | Quantity | Quantity of the corresponding media | Yes |  | 10 | | 5. | Barcode | product barcode | Yes |  | 312321 | | 4. | Weight | Weight of the product | Yes |  | 2 | | 5. | Description |  | Yes |  | This is a sample DVD | | 6. | Special information fields for the product type |  | Yes |  |  |  1. **Output data**   No   1. **Postconditions**   A new product will be added |

## Use case “Update Product”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use case “Update Product”**   1. **Use case code** UC009 2. **Brief Description**   This use case describes the interaction between the product manager and the product when there is a need to update products   1. **Actors:** Product manager 2. **Preconditions**   You must log in with a product manager account.   1. **Basic Flow of Events** 2. The manager selects the product that needs fixing and chooses to fix it 3. The system displays the product information form and the manager can edit it 4. The manager fills in the information that needs to be updated 5. The system checks the validity of the entered information 6. The system updates information, announces success and returns to the product management screen 7. **Alternative flows**   **Table A-Alternative flows of events for UC “**Create product**”**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | |  |  |  |  |  |  1. **Input data**  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | | 1. | Title | Title of a media product | Yes |  | Sample DVD | | 2. | Category |  | Yes |  |  | | 3. | Price | Price of the corresponding media product | Yes |  | 20.000 | | 4. | Quantity | Quantity of the corresponding media | Yes |  | 10 | | 5. | Weight | Weight of the product | Yes |  | 2 | | 6. | Barcode | product barcode | Yes |  | 312321 | | 7. | Description |  | Yes |  | This is a sample DVD | | 8. | Special information fields for the product type |  | Yes |  |  |  1. **Output data**   **No**   1. **Postconditions**   New product updated successfully |

## Use case “Pay order”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use case “Pay order”**   1. **Use case code** UC010 2. **Brief Description**   This use case describes the interaction between customers, AIMS software and VNPay when customers to pay order   1. **Actors:**   3.1 Customers  3.2 VNPay   1. **Preconditions**   The AIMS software has generated an invoice, including product details, VAT, shipping fees, and total payment.   1. **Basic Flow of Events**    * + 1. AIMS software displays the invoice        2. Customer asks to pay the invoice        3. AIMS software redirects to VNPay        4. VNPay proceeds the transaction        5. AIMS software saves invoice and payment transaction        6. AIMS software notifies that the payment was successful 2. **Alternative flows**  |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | | 1. | At Step 5 | If the customer cancels the payment transaction |  | At Step 1 |  1. **Input data**   Input data of order information and shipping fee   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Valid condition** | **Example** | | 1. | Title | Title of a media product |  | DVD Phim Vượt  ngục | | 2. | Price | Price of the corresponding media  product | Comma for thousands separator  Positive integer  Right alignment | 123,000 | | 3. | Quantity | Quantity of the corresponding media | Positive integer  Right alignment | 2 | | 4. | Amount | Total money of the corresponding media | Comma for thousands separator  Positive integer  Right alignment | 246,000 |  1. **Output data**   Output data of invoice   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Display format** | **Example** | | 1. | Title | Title of a media product |  | DVD Phim Vượt  ngục | | 2. | Price | Price of the corresponding media product | * Comma for thousands separator * Positive integer * Right alignment | 123,000 | | 3. | Quantity | Quantity of the corresponding media | * Positive integer * Right alignment | 2 | | 4. | Amount | Total money of the corresponding media | * Comma for thousands separator * Positive integer * Right alignment | 246,000 | | 5. | Subtotal Before VAT | Total price of products in the cart before VAT | * Comma for thousands separator * Positive integer * Right alignment | 2,106,000 | | 6. | Subtotal | Total price of products in the cart with VAT | 2,316,600 | | 7. | Shipping fees |  | 30,000 | | 8. | Total | Sum of subtotal and shipping fees |  | 2,346,600 | | 9. | Currency |  |  | VND | | 10. | Name |  |  | Vu Ngoc Lam | | 11. | Phone number |  |  | 0987654321 | | 12. | Province |  |  | Hanoi | | 13. | Address |  |  | 12, 34 Alley of Tran Thai Tong street, Cau Giay district |  1. **Postconditions**   The payment is completed, the invoice and transaction are recorded, the order status is updated, the customer is notified of the payment result, and stock levels are adjusted. |

# Supplementary specification

## Functionality

* The system must support searching by keyword, product category, price range, brand, and rating.
* Users can filter results using multiple criteria at the same time.

## Usability

* The search interface must be simple and user-friendly.

## Reliability

* Availability: The system is expected to be available most of the time, with scheduled maintenance windows limited to no more than 1 hour per month.
* Error handling: The system must ensure error-free product searches. If an error occurs, the system must provide a clear notification and guide the customer to retry.

## Performance

* Search results must be displayed as quickly as possible.
* Product data should be stored in cache to speed up query processing.

## Supportability

* The software should support Windows platform desktop.

## Other requirements

* Support intelligent search features such as keyword suggestions and spell correction.