HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

School of Information and Communications Technology

Software Requirement Specification

Version 1.2

AIMS – An Internet Media Store

Subject: ITSS Software Development

Group 2

Đặng Văn Nhân – 20225990

Hà Việt Khánh – 20225979

Hồ Bảo Thư – 20226003

Trần Cao Phong – 20226061

Nguyễn Lan Nhi - 20225991

*Hanoi,* *March 2025*

Table of contents

Table of contents 1

1 Introduction 3

1.1 Objective 3

1.2 Scope 3

1.3 Glossary 4

1.4 References 6

2 Overall Description 7

2.1 Survey 7

2.2 Overall requirements 7

2.3 Business process 8

2.3.1 Business Operation – Place Order 9

2.3.2 Business Operation – Manage Products 10

2.3.3 Business Operation – Manage Users 10

3 Detailed Requirements 11

3.1 Use case “Place order” 11

3.2 Use case “Pay order” 14

3.3 Use case “Place rush order” 16

3.4 Use case “Cancel order” 18

3.5 Use case “View order details” 20

3.6 Use case “Create products” 21

3.7 Use case “Update products” 23

3.8 Use case “Approve order” 24

3.9 Use case “Reject order” 26

3.10 Use case “Create users” 27

4 Supplementary specification 29

4.1 Functionality 29

4.2 Usability 29

4.3 Reliability 29

4.4 Performance 29

4.5 Supportability 29

4.6 Other requirements 30

# Introduction

## Objective

An Internet Media Store (AIMS) is an e-commerce desktop application designed to facilitate the buying and selling of physical media products such as books, CDs, LP records, and DVDs. This system provides a seamless shopping experience for customers while enabling product managers to efficiently manage inventory. The software is designed to handle up to 1,000 concurrent users without performance degradation and can operate continuously for 300 hours without failure.

This Software Requirements Specification (SRS) document defines the functional and non-functional requirements of the AIMS software, serving as a reference for developers, testers, product managers, and other stakeholders involved in its development and deployment.

## Scope

The AIMS software allows users to browse, search, purchase, and manage physical media products in a structured and user-friendly manner. The system supports both normal and rush orders, integrates with VNPay for payment processing, and provides robust inventory management capabilities. Key functionalities include:

* For Customers:
  + Searching and filtering products by attributes.
  + Viewing product details and adding items to the cart.
  + Normal or rush order options.
  + Making payment through VNPay.
  + Viewing and managing past orders, including order cancellations.
* For Product Managers:
  + Adding, editing, and removing products from the inventory.
  + Monitoring and updating product prices.
  + Reviewing and approving customer orders.
  + Managing stock availability.
* For Administrators:
  + Creating and managing user accounts.
  + Assigning roles and permissions to users.

## Glossary

| ***No*** | ***Term*** | ***Explanation*** | ***Example*** | ***Note*** |
| --- | --- | --- | --- | --- |
| 1 | session | A session is a temporary period of interaction between a user and a system, during which the user accesses and interacts with software, a website, or an application. The system maintains session-specific data, such as user authentication, preferences, and activity history, for the duration of the session. Once the session ends—either through user logout, session timeout, or browser closure—the stored session data is typically cleared unless explicitly saved. | software session |  |
| 2 | VAT (Value-added tax) | VAT (Value-Added Tax) is a consumption tax applied to the value added at each stage of production or distribution of goods and services. It is typically calculated as a percentage of the final selling price and is collected by businesses on behalf of the government. |  |  |
| 3 | API (Application Programming Protocol) | An API (Application Programming Interface) is a set of rules, protocols, and tools that enables different software applications to communicate and interact with each other. | VNPay API | AIMS connects API of VNPay for transaction. |
| 4 | Payment gateway | A payment gateway is a technology service that facilitates the secure transmission of payment information between a merchant's website or application and the financial institutions involved in processing payment transactions. |  |  |
| 5 | GUI (Graphical user interface) | A Graphical User Interface (GUI) refers to the visual elements and interactive components of a software application that allow users to interact with the system using graphical icons, buttons, menus, windows, and other visual elements. |  |  |
| 6 | Credit card | A credit card is a payment card issued by a financial institution, such as a bank or credit union, that allows cardholders to borrow funds up to a predetermined limit to make purchases or pay for goods and services. |  | AIMS currently supports for credit card payment through VNPay |
| 7 | Authentication | Authentication is the process of verifying the identity of a user, device, or system attempting to access a resource or service. |  |  |
| 8 | Response time | Response time refers to the amount of time it takes for a system to respond to a user's request or input. |  |  |

## References

# Overall Description

## Survey

The system under consideration is an internet media store designed to facilitate the desire to purchase digital media products online by the customers. This software serves as a comprehensive platform for not only customers but also the shop managers or product managers.

The system includes four actors:

- **Customer**: They can view, search for or sort in order by many factors the products that are available in the store. To place an order, they must add, update the products in the cart, and provide delivery information to the system. If the information is available, the customer needs to pay for the order through the VNPay platform. If the order is created successfully, the customer will get an email from AIMS about the order invoice. Moreover, customers can place orders with rush orders in some cases.

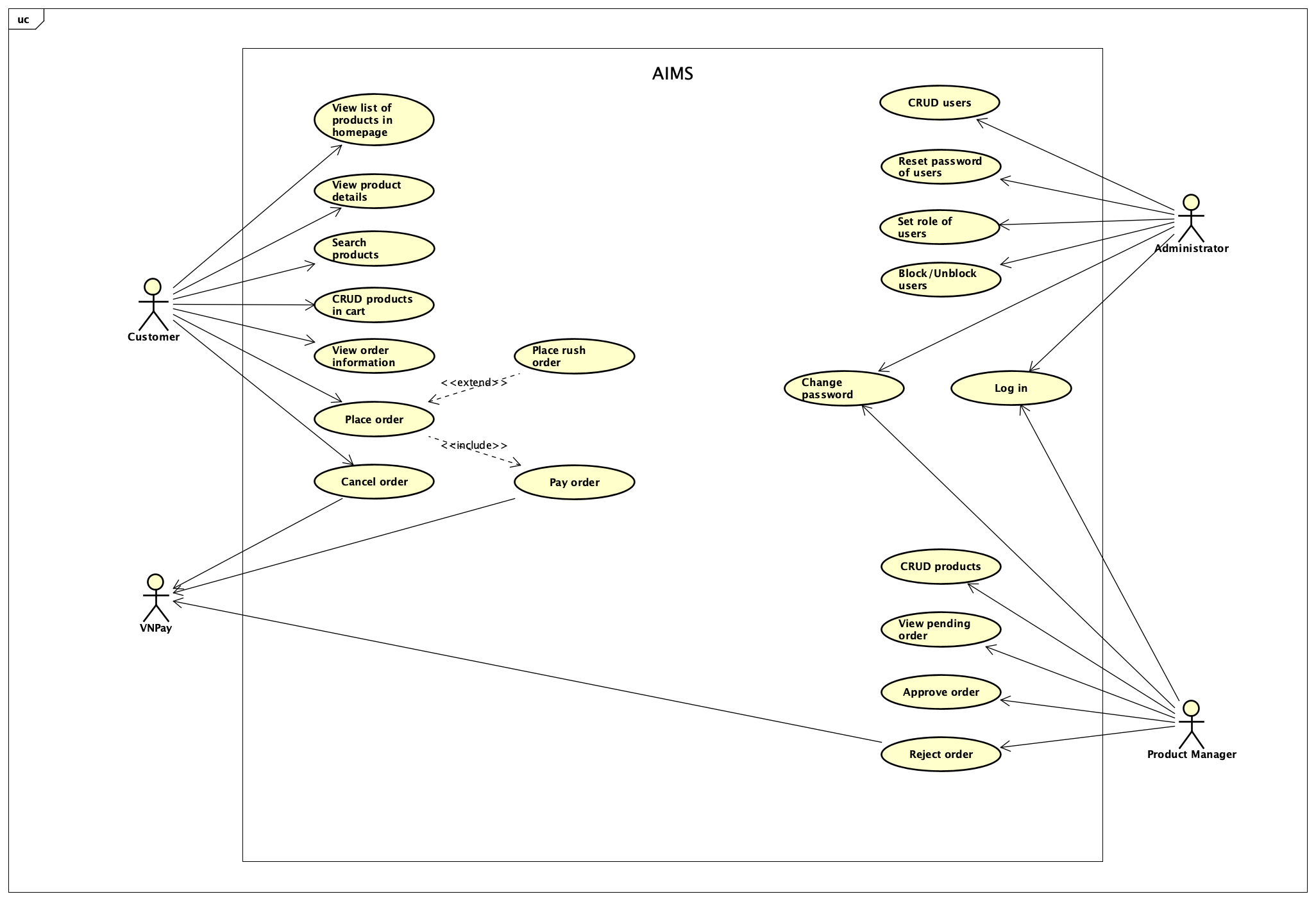
- **Product Manager**: They can manage their products in their shop through the user interface of manager supplied by AIMS software. They can add, remove, update information about the products. Moreover, under some circumstances, they can apply sales for many products.

- **Administrator**: They can gain access to mange user in the system. They can also block or unblock the user (customer, product manager) and the system will send the mail to that account. In addition, admin can also change the role of the user like a user can be a customer and a product manager at the same time.

- **VNPay**: A stakeholder in the system when they supply the API to make the transaction in the AIMS software.

## Overall requirements

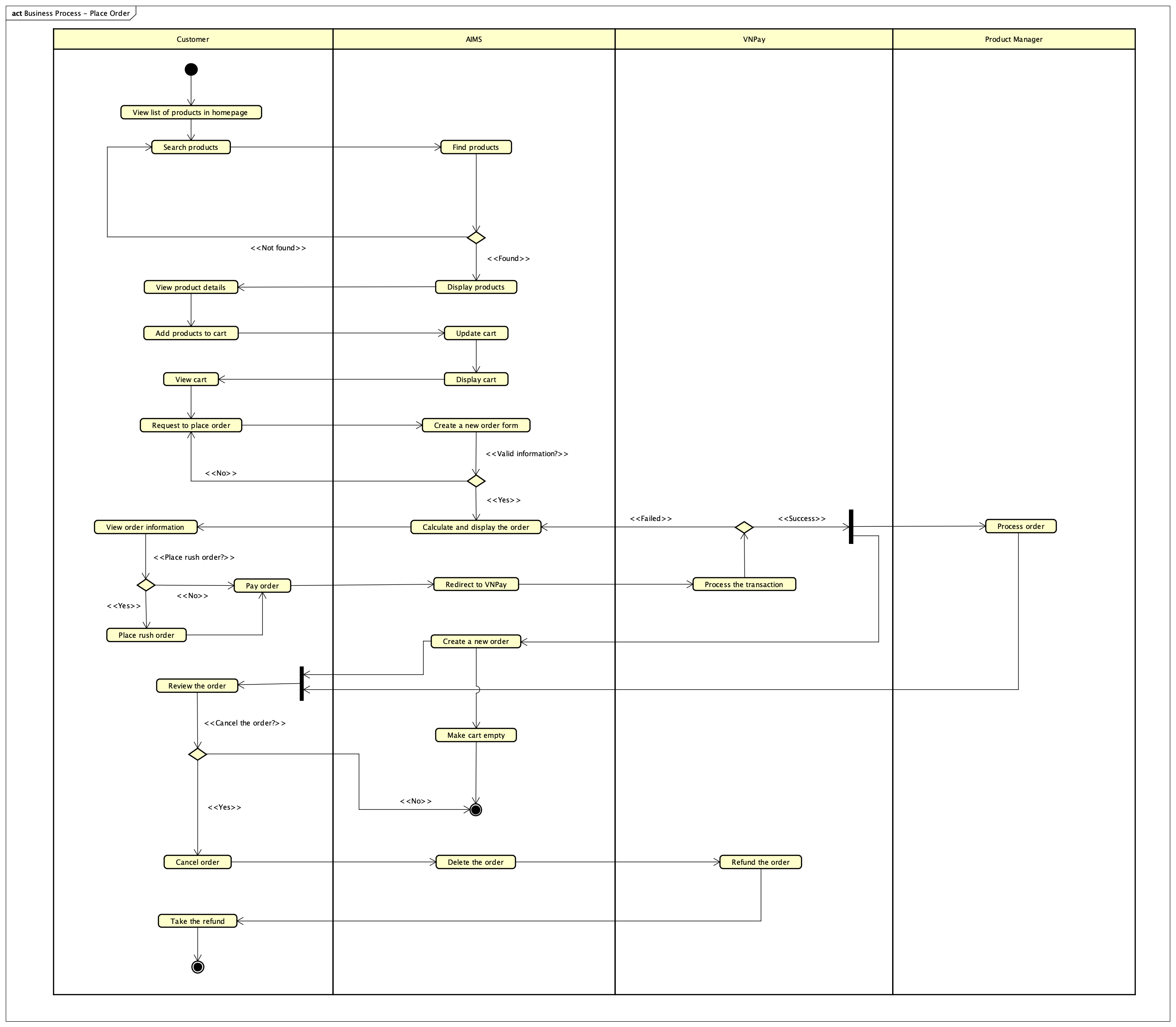
Use case diagram represents the interactions between actors and use cases. It represents the functional requirements of the system, showing the interaction between external and internal actors with the system.

**

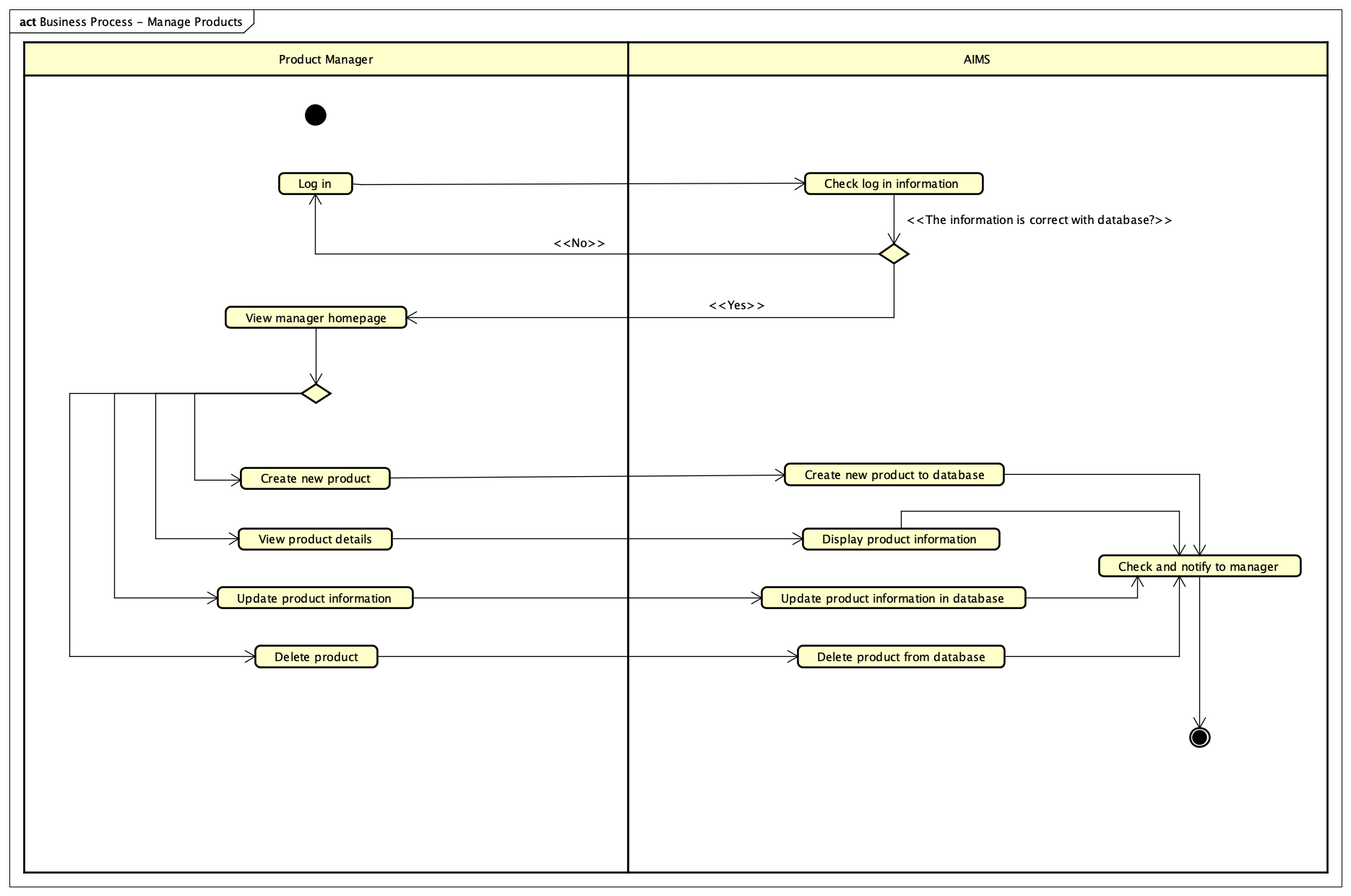
## Business process

In the AIMS software, there are three primary business operations: "**Place Order**" carried out by Customer, "**Manage Products**" handled by Product Managers, and "**Manage Users**" by Administrators. Each of these business processes is detailed through activity diagram in their respective sections.

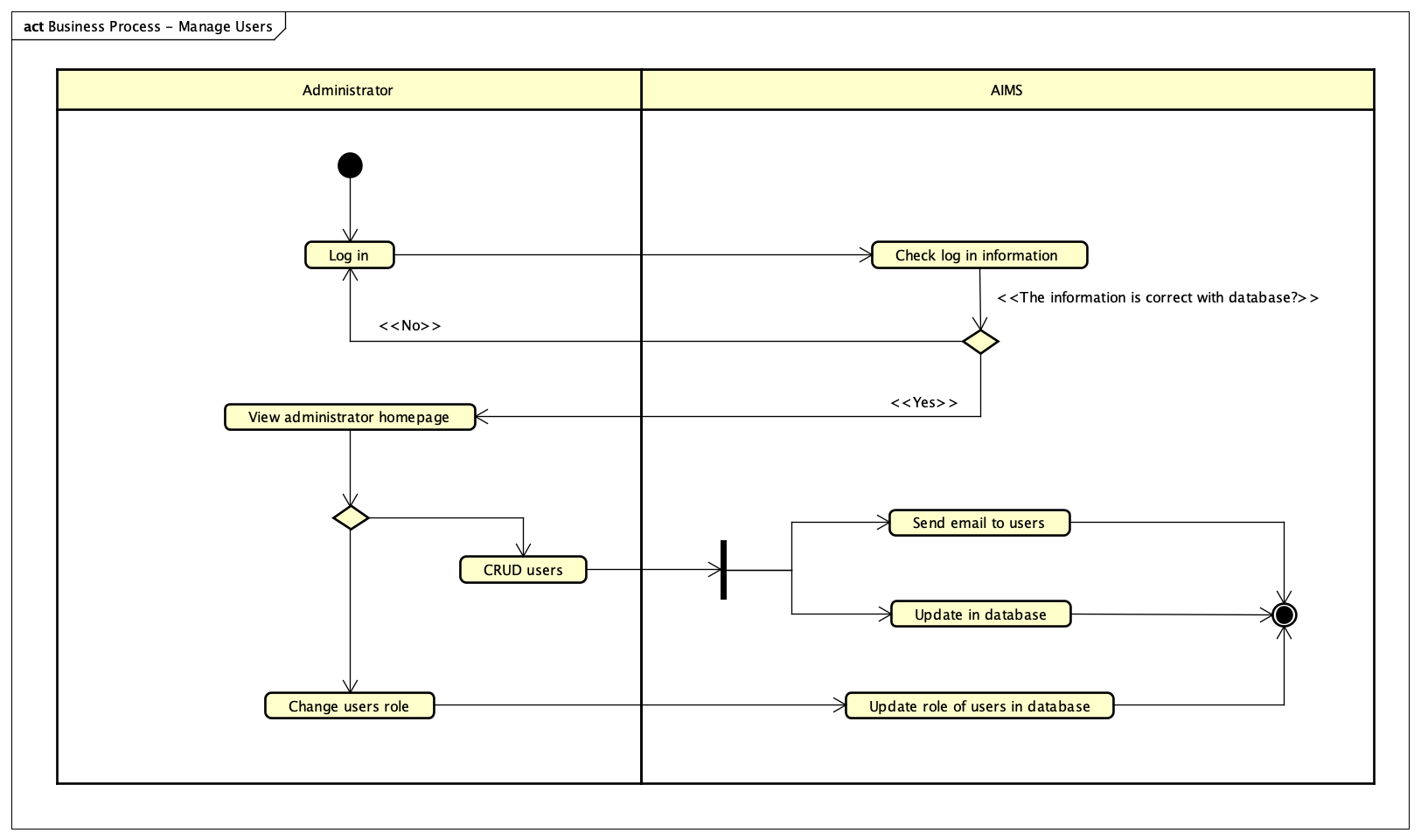
### Business Operation – Place Order



### Business Operation – Manage Products



### Business Operation – Manage Users



# Detailed Requirements

## Use case “Place order”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Place order”**   1. **Use case code**   UC001   1. **Brief Description**   This use case describes the interaction between Customer and AIMS software when Customer wishes to place an order.   1. **Actors**    1. **Customer** 2. **Preconditions**   Cart is not currently empty; All selected items are in stock and available in the requested quantity; At least 1 item is selected by Customer.   1. **Basic Flow of Events** 2. Customer chooses products they want to purchase and requests to place an order 3. Software checks the availability of products in the cart 4. Software displays the form of delivery information with order information 5. Customer enters and submits delivery information (see Table 1) 6. Software check the submitted information 7. Software calculates and updates order information with shipping fee 8. Software displays and saves invoice information (see Table 2) 9. Customer chooses payment method and asks for payment 10. Software calls Use Case “Pay order” 11. Software creates and saves a new order 12. Software sends email to customer for the success order 13. Software makes the cart empty 14. Software displays the successful order notification, the order and the transaction information (see Table 3) 15. **Alternative flows**   Table A1 - Alternative flows of events for UC “Place order”   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | | 1. | At Step 3 | If the products are not available | The software displays the inventory quantity of each unmet product | Use case ends | | 2. | At Step 5 | If the delivery information is invalid | The software notifies that the delivery information is invalid | At Step 3 | | 3. | At Step 5 | If the user chooses to place a rush order | The software inserts use case “Place rush order” | At Step 6 | | 4. | At Step 6 | If the user adjusts the delivery method or the items | The software recalculates the delivery fees and display the corresponding invoice | At Step 7 | | 5. | At Step 8 | If the order payment is not successful or goes back from payment |  | At Step 5 |  1. **Input data**   Table 1 - Input data of Customer   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | |  | Receiver Name |  | Yes |  | Ho Bao Thu | |  | Phone Number |  | Yes | 10 digits | 0123456789 | |  | Province | Choose from a list | Yes |  | Hanoi | |  | Address |  | Yes |  | 123 ABC street, DEF district | |  | Shipping address |  | Yes |  |  | |  | Address type | Choose from a list | Yes |  | Office | |  | Email |  | Yes | Valid domain part with @ | abc@gmail.com | |  | Shipping instruction | Any additional instruction for shipping preocess | No |  |  |  1. **Output data**   Table 2 - Output data of order information and shipping fee   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Display format** | **Example** | |  | Title | Title of a media product |  | Book Cây cam ngọt của tôi | |  | Price | Price of the corresponding media product | * Comma for thousands separator * Positive integer * Right alignment | 200,000 | |  | Quantity | Quantity of the corresponding media | * Positive integer * Right alignment | 1 | |  | Amount | Total money of the corresponding media | * Comma for thousands separator * Positive integer * Right alignment | 200,00 | |  | Subtotal before VAT | Total amount of all products in the order excluding VAT |  | 1,263,000 | |  | Subtotal after VAT | Total amount of all products in the order with VAT |  | 1,389,300 | |  | Shipping fee |  |  | 20,000 | |  | Total |  | * Comma for thousands separator * Positive integer * Right alignment | 1,409,300 | |  | Concurrency |  |  | VND |   Table 3 - Output data of general information of order and transaction information   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Display format** | **Example** | |  | Customer name |  |  | Ho Bao Thu | |  | Phone Number |  | 10 digits | 0123456789 | |  | Province |  |  | Hanoi | |  | Address |  |  | 123 ABC street, DEF district | |  | Total amount |  | * Comma for thousands separator * Positive integer * Right alignment | 1,283,004 | |  | Transaction ID |  |  | 00001 | |  | Transaction content |  |  | Order #00001 | |  | Transaction date |  | dd/mm/yyyy | 23/02/2025 | |  | Currency |  |  | VND | |  | Order status |  |  | Pending | |  | Order email |  |  | abc@gmail.com |  1. **Postconditions**   If the payment is successful, the cart is emptied; order, payment, and delivery information are stored; the user receives a confirmation, and stock levels are updated. If the payment fails, no changes occur. |

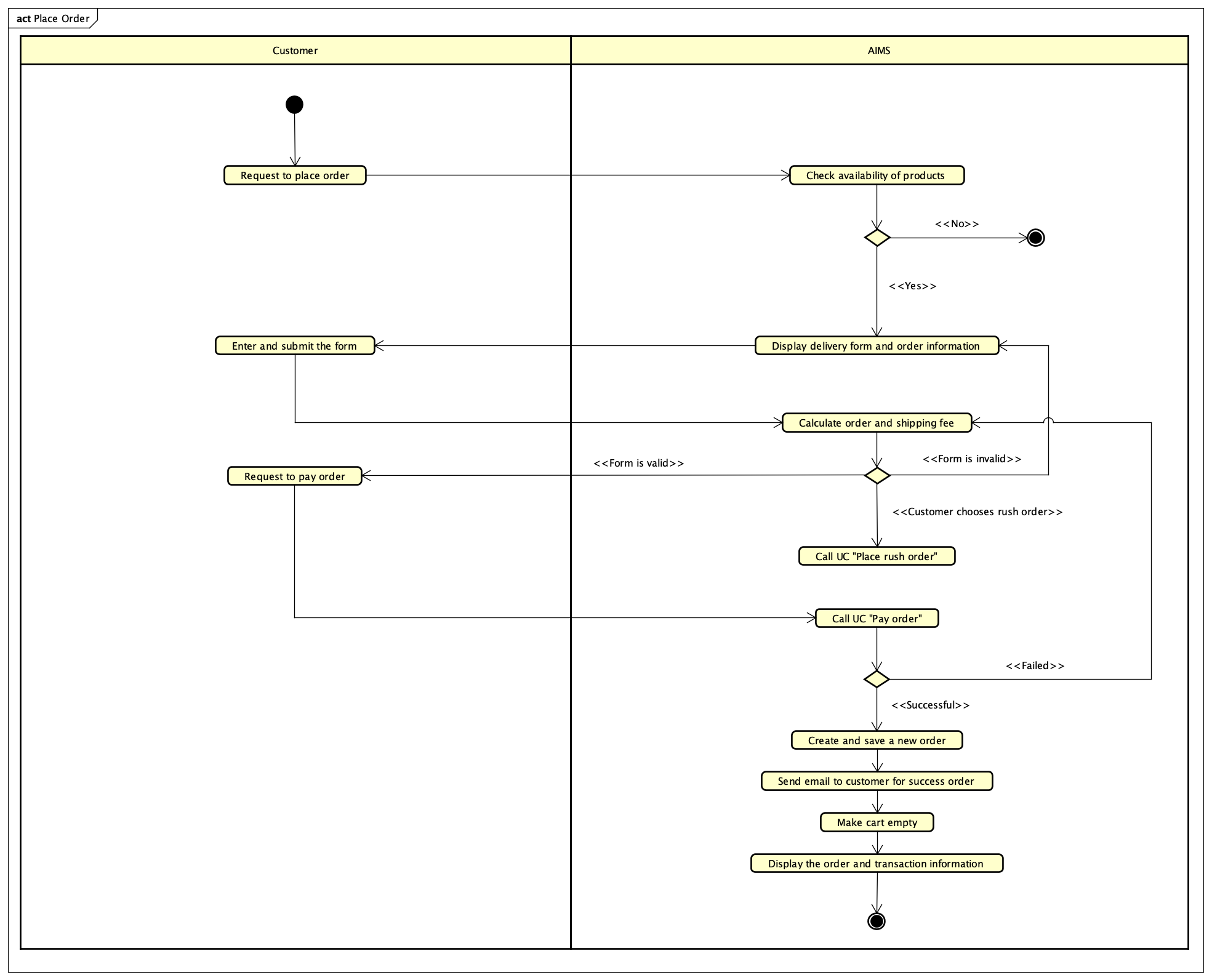
****

Figure : Activity Diagram for "Place Order"

## Use case “Pay order”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Pay order”**   1. **Use case code**   UC002   1. **Brief Description**   This use case describes how software interacts with Customer and VNPay to pay the previous order.   1. **Actors**    1. **Customer**    2. **VNPay** 2. **Preconditions**   The order is in payment state and software can interact with VNPay.   1. **Basic Flow of Events**   1. Software displays the invoice (see Table 1)  2. Software asks Customer to pay order  3. Software redirects Customer to VNPay with payment information  4. VNPay sends payment result to VNPay  5. Software update saves the invoice and payment transaction  6. VNPay notifies the payment result   1. **Alternative flows**   Table A1 - Alternative flows of events for UC “Pay order”   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | | 1 | At Step 3 | Customer cancels payment | Software notices the cancellation. | At Step 1 | | 2 | At Step 4 | Payment fails | Software notifies the Customer of failure. | At Step 1 |  1. **Input data** 2. **Output data**   Table 1 - Output data of invoice   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Display format** | **Example** | |  | Title | Title of the media product | Text | DVD Phim Vượt ngục | |  | Price | Price of the media product | Positive integer, right alignment | 123,000 | |  | Quantity | Quantity of the media product | Positive integer, right alignment | 2 | |  | Amount | Total money of the media product | Positive integer, right alignment | 246,000 | |  | Subtotal before VAT | Total price of products in the cart before VAT | Positive integer, right alignment | 2,106,000 | |  | Subtotal with VAT | Total price of products in the cart with VAT | Positive integer, right alignment | 2,316,600 | |  | Shipping fee | Shipping fee | Positive integer, right alignment | 30,000 | |  | Total | Total price of subtotal and shipping fee | Positive integer, right alignment | 2,346,600 | |  | Currency |  |  | VND |  1. **Postconditions**   If the payment is successful, the customer is notified and receives a bill detail email. If it fails, nothing happens. |

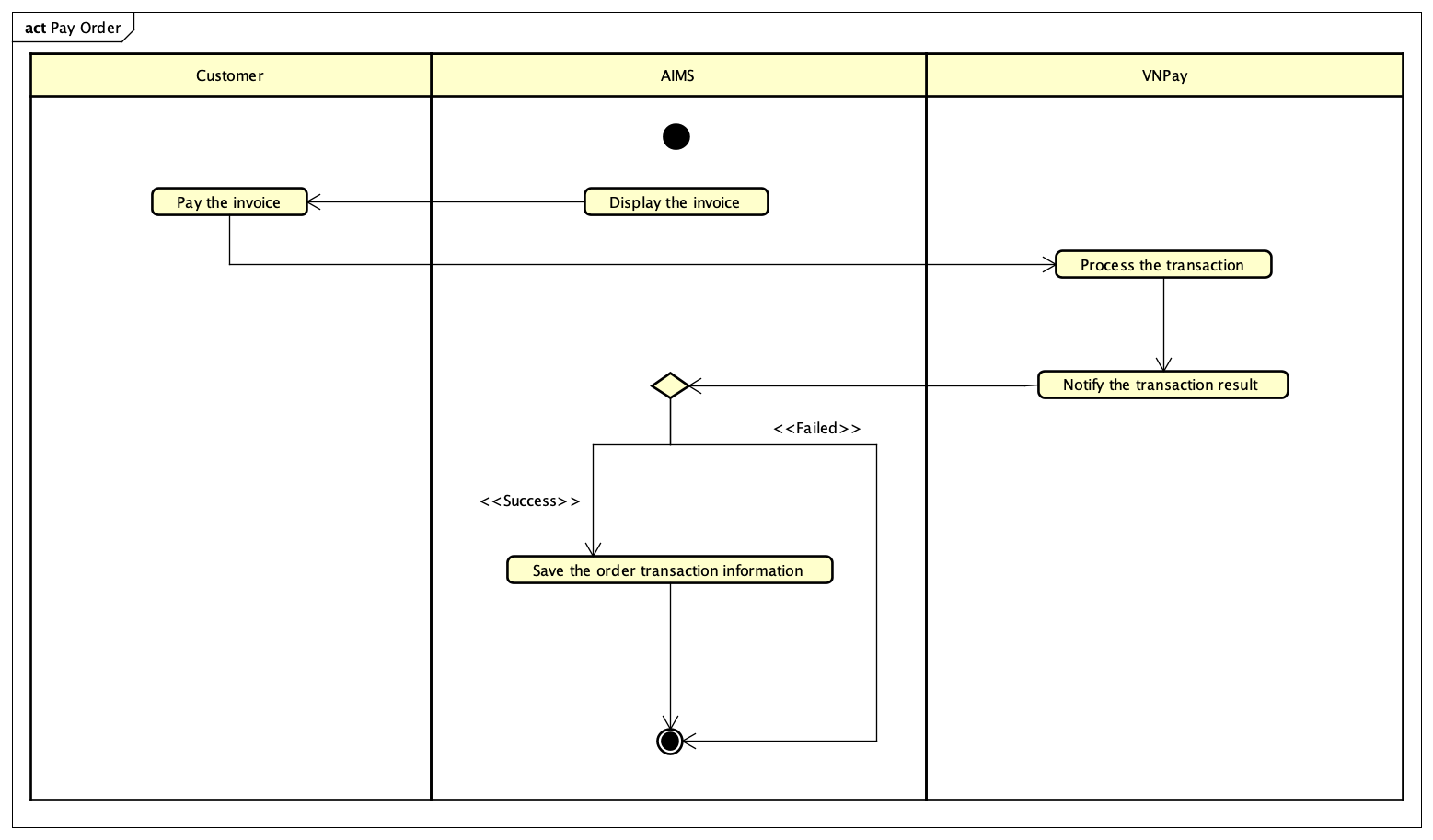


Figure : Activity diagram for "Pay Order"

## Use case “Place rush order”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Place rush order”**   1. **Use case code**   UC003   1. **Brief Description**   This use case describes the interaction between the customer and software when the customer wants to place an order with rush order.   1. **Actors**    1. **Customer** 2. **Preconditions**   Customer’s cart is not empty, the order is being placed by customer.   1. **Basic Flow of Events** 2. Customer requests to place rush order when placing the order 3. The software checks whether the delivery address supports these services and the availability of products 4. The software asks for additional information (see Table 1) 5. Customer enters and submits additional information 6. AIMS software calculates, updates, and displays the total order information with shipping fee (see Table 2) 7. **Alternative flows**   Table A1 - Alternative flows of events for UC “Place rush order”   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | | 1. | At Step 2 | If the products and the address is not available for rush order | The software displays: Invalid product/Invalid address and returns to UC001 “Place order” | Use case ends | | 2. | At Step 7 | If the customer cancels the payment |  | At Step 3 |  1. **Input data**   Table 1 - Input data of Customer   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | | 1. | Order instructions | Additional notes from customer | No |  |  | | 2. | Delivery time |  | Yes | hh:mm, 24-hour based. Delivery time within 2 hours. | 14:20 |  1. **Output data**   Table 2 - Output data of shipping fee   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Display format** | **Example** | |  | Standard shipping fee | Shipping fee for non-eligible express shipping products | * Comma for thousands separator * Positive integer * Right alignment | 30,000 | |  | Express shipping fee | Express shipping fee for eligible products | * Comma for thousands separator * Positive integer * Right alignment | 50,000 |  1. **Postconditions** |

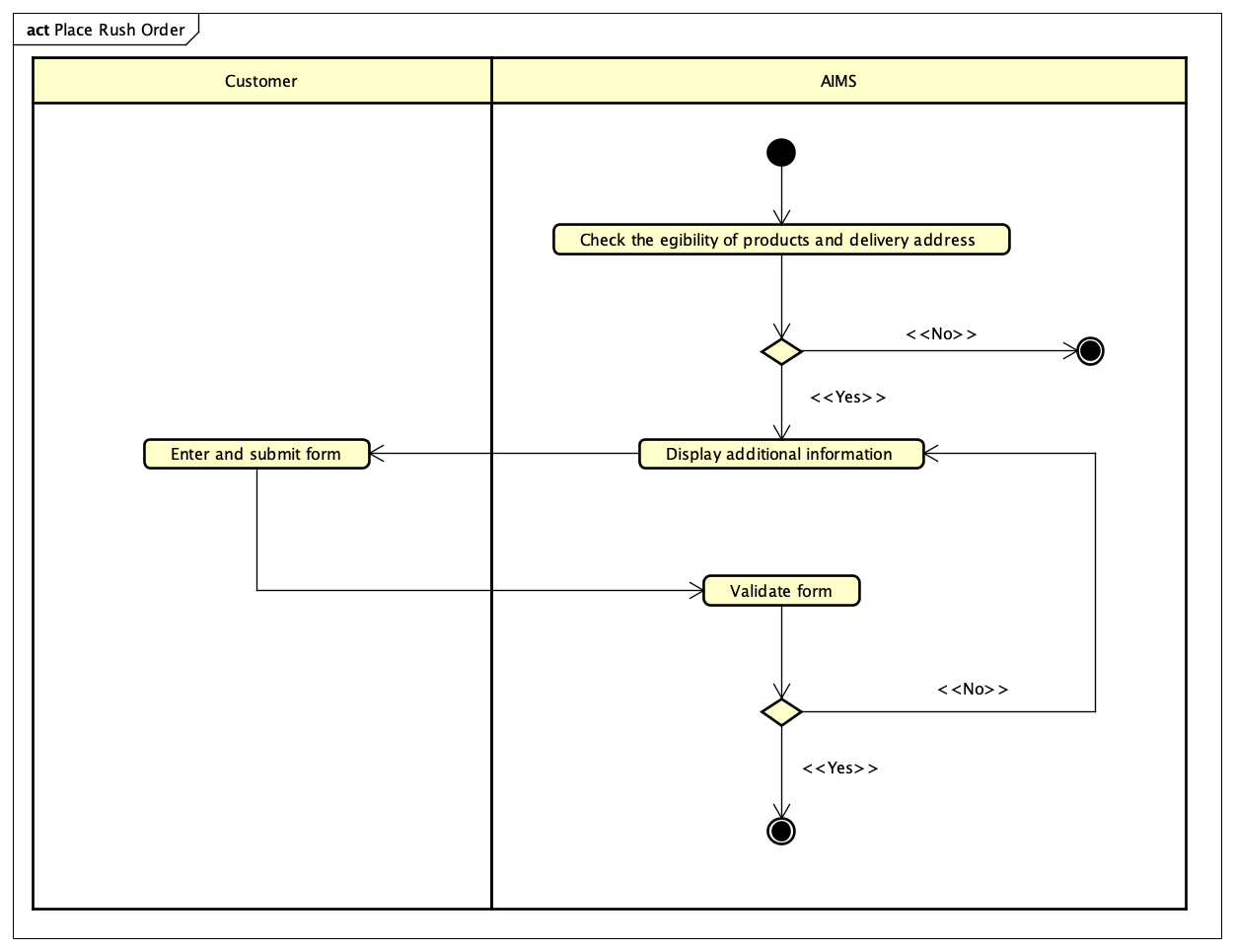


Figure : Activity Diagram for "Place Rush Order"

## Use case “Cancel order”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Cancel order”**   1. **Use case code**   UC004   1. **Brief Description**   This use case describes the interaction between the customer and software when the customer wants to cancel an order.   1. **Actors**    1. **Customer** 2. **Preconditions**   Customer selects to view an order successfully.   1. **Basic Flow of Events** 2. Customer selects to cancel a specific order 3. The software displays all relevant order details (see Table 1) 4. The software changes the order’s status to “Cancelled” 5. The software calls VNPay to refund the amount of that order. 6. **Alternative flows**   Table A1 - Alternative flows of events for UC “Cancel order”   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | | 1. | At Step 1 | The order has been accepted/rejected by Product Manager before | The software informs to customer: “This order can not be cancelled” | Use case ends |  1. **Input data** 2. **Output data**   Table 1 - Output data of order details   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Display format** | **Example** | |  | Order ID | Unique identifier for the order | Numeric | 000018 | |  | Customer’s name |  | Text | Nguyen Lan Nhi | |  | Itemized list of products |  | Text | DVD | |  | Quantity | Number of units for each product | Positive integer | 1 | |  | Price | Price of each product | Positive integer, comma for thousands separator | 200,000 | |  | Total amount | Total cost including tax and shipping | Positive integer, comma for thousands separator | 260,000 | |  | Tax |  | Positive integer, comma for thousands separator | 30,000 | |  | Shipping fee |  | Positive integer, comma for thousands separator | 30,000 | |  | Shipping address | Delivery address provided by the customer | Text | Hoang Mai, Ha Noi | |  | Payment method |  | Text | VNPay | |  | Transaction ID | Unique identifier for the transaction | Numeric | 00812 | |  | Amount paid | Total amount paid by the customer | Positive integer, comma for thousands separator | 260,000 | |  | Refund status | Status of the refund process | Text (e.g., "Pending", "Failed", "Successful") | Successfully |  1. **Postconditions** |

## Use case “View order details”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “View order details”**   1. **Use case code**   UC005   1. **Brief Description**   This use case describes the interaction between Customer/Product Manager and software when Customer/Product Manager wants to view product details.   1. **Actors**    1. **Product Manager/Customer** 2. **Preconditions**   Customer/Product Manager has already accessed theo product listing screen.   1. **Basic Flow of Events** 2. Customer/Manager selects a product from the product listing screen 3. Software retrieves the product details from database and displays the product details (see Table 1) 4. **Alternative flows** 5. **Input data** 6. **Output data**   Table 1 - Output data of product details   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Display format** | **Example** | |  | Title of product |  |  | Movie | |  | Category | Category of products |  | DVD | |  | Description | Description of product |  |  | |  | Price | Price of each product | Positive integer, comma for thousands separator | 200,000 | |  | Availability status | In stock/Out of Stock |  | In stock | |  | Product images |  | Image | 30,000 | |  | Rating & Reviews | Star Rating + Review Count |  |  |  1. **Postconditions** |

## Use case “Create products”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Create products”**   1. **Use case code**   UC006   1. **Brief Description**   This use case describes the interaction between Product Manager and AIMS software when Product Manager wishes to create a new producet.   1. **Actors**    1. **Product Manager** 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 (see Table 1) 6. The system checks the validity of entered product information 7. The system notifies the successful creation of new product 8. **Alternative flows**   Table A1 - Alternative flows of events for UC “Create products”   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | | 1. | At Step 5 | If the user leaves any of the required information blank | The software reports an error and requires the manager to fill in all the information | At Step 4 |  1. **Input data**   Table 1 - Input data of Product Manager   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | |  | Title | Title of media product | Yes |  | SHINee WORLD VI | |  | Price |  | Yes | Numeric | 20.000 | |  | Total quantity |  | Yes |  | 10 | |  | Weight | Weight of a product | Yes | Numeric, in kilogram unit | 2kg | |  | Rush order support |  | Yes | Yes/No | Yes | |  | Image URL |  | Yes |  |  | |  | Barcode |  | Yes |  | 250508 | |  | Description |  | Yes |  |  | |  | Dimension |  | Yes |  |  | |  | Category |  | Yes |  |  | |  | Special information fields for the product type |  | Yes |  |  |  1. **Output data** 2. **Postconditions**   A new product will be created. |

## Use case “Update products”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Update products”**   1. **Use case code**   UC007   1. **Brief Description**   This use case describes the interaction between Product Manager and Software when there is a need to update products.   1. **Actors**    1. **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 for the manager (see Table 1) 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 the manager to the product manager screen 7. **Alternative flows**   Table A1 - Alternative flows of events for UC “Update products”   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | | 1. | At Step 3 | If the user leaves any of the required information blank | The software reports an error and requires the manager to fill in all the information | At Step 2 |  1. **Input data**   Table 1 - Input data of Product Manager   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | |  | Title | Title of media product | Yes |  | SHINee WORLD VI | |  | Price |  | Yes | Numeric | 20.000 | |  | Total quantity |  | Yes |  | 10 | |  | Weight | Weight of a product | Yes | Numeric, in kilogram unit | 2kg | |  | Rush order support |  | Yes | Yes/No | Yes | |  | Image URL |  | Yes |  |  | |  | Barcode |  | Yes |  | 250508 | |  | Description |  | Yes |  |  | |  | Dimension |  | Yes |  |  | |  | Category |  | Yes |  |  | |  | Special information fields for the product type |  | Yes |  |  |  1. **Output data** 2. **Postconditions**   The product will be updated. |

## Use case “Approve order”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Approve order”**   1. **Use case code**   UC008   1. **Brief Description**   This use case describes the interaction between Product Manager, VNPay and software when the product manager wants to approve a pending order.   1. **Actors**    1. **Product Manager**    2. **VNPay** 2. **Preconditions**   The list of pending orders is not empty, product manager logged in before accessing the pending orders list.   1. **Basic Flow of Events** 2. Product manager views the list of pending orders and selects a specific order 3. Product manager views the details of this order 4. Product manager approves the order 5. The software checks the approved order 6. The software sends email of order information to customers to notice that the order was approved 7. **Alternative flows**   Table A1 - Alternative flows of events for UC “Approve order”   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | | 1. | At Step 4 | If there are not enough products in stock | The software rejects accepting order and sends notification to customer, and asks VNPay to refund to customer | Use case ends |  1. **Input data** 2. **Output data**   Table 1 - Output data of general information of order and transaction information   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Display format** | **Example** | |  | Customer name |  |  | Ho Bao Thu | |  | Phone Number |  | 10 digits | 0123456789 | |  | Province |  |  | Hanoi | |  | Address |  |  | 123 ABC street, DEF district | |  | Total amount |  | * Comma for thousands separator * Positive integer * Right alignment | 1,283,004 | |  | Transaction ID |  |  |  | |  | Transaction content |  |  |  | |  | Transaction date |  | dd/mm/yyyy | 23/02/2025 | |  | Currency |  |  | VND |  1. **Postconditions** |

## Use case “Reject order”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Reject order”**   1. **Use case code**   UC009   1. **Brief Description**   This use case describes the interaction between Product Manager, VNPay and software when the product manager wants to reject a pending order.   1. **Actors**    1. **Product Manager**    2. **VNPay** 2. **Preconditions**   The list of pending orders is not empty, product manager logged in before accessing the pending orders list.   1. **Basic Flow of Events** 2. Product manager views the list of pending orders and selects a specific order 3. Product manager views the details of this order 4. The product manager rejects the order and provides the reason for the rejection 5. The software sends email of order information to customers to notice that the order was rejected 6. The software asks VNPAY to refund the customer 7. **Alternative flows** 8. **Input data** 9. **Output data**   Table 1 - Output data of general information of order and transaction information   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Display format** | **Example** | |  | Customer name |  |  | Ho Bao Thu | |  | Phone Number |  | 10 digits | 0123456789 | |  | Province |  |  | Hanoi | |  | Address |  |  | 123 ABC street, DEF district | |  | Total amount |  | * Comma for thousands separator * Positive integer * Right alignment | 1,283,004 | |  | Transaction ID |  |  |  | |  | Transaction content |  |  |  | |  | Transaction date |  | dd/mm/yyyy | 23/02/2025 | |  | Currency |  |  | VND |  1. **Postconditions** |

## Use case “Create users”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Create users”**   1. **Use case code**   UC010   1. **Brief Description**   This use case describes how Administrator creates a new user account.   1. **Actors**    1. **Administrator** 2. **Preconditions**   Admin must be logged in. User email/username haven’t been used.   1. **Basic Flow of Events** 2. Admin navigates to the Create User screen 3. Admin enters user details (see Table 1) 4. Software validates the input 5. Software creates the new user account 6. Software displays a success message to Administrator 7. Software sends email to the new user, notifies that the account has been created successfully 8. **Alternative flows**   Table A1 - Alternative flows of events for UC “Create users”   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | | 1. | At Step 3 | Email is already in use | Software notifies admin | Resumes at Step 2 | | 2. | At Step 3 | Invalid input (e.g., weak password) | Software prompts for correction | Resumes at Step 2 |  1. **Input data**   Table 1 - Input data of Administrator   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | |  | Full name | User's full name | Yes | Non-empty | Viet Khanh Ha | |  | Email | User's email address | Yes | Valid email format | Havietkhanh49@gmail.com | |  | Role | User’s role in the system | Yes | Must in a list | Adminstrator | |  | Password | User's initial password | Yes | Have at least 8 character | 123456789 |  1. **Output data**   Table 2 - Output data of user’s information   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Display format** | **Example** | |  | User ID | Unique for user | Text | U1234 | |  | Registration Date | Date when the account was created | dd/mm/yy | 04/09/2024 | |  | Confirmation Notify | State message | Text | Successful |  1. **Postconditions**   If the account is successfully created, an email verification is sent, and the user can log in after verification. If validation fails or the email is not verified, the account remains inactive. |

# Supplementary specification

Supplementary specifications are additional details that complement the main Software Requirements Specification (SRS). They often include information about external interfaces, performance requirements, quality attributes, design constraints, and other supplementary details that are necessary for a comprehensive understanding of the software system.

## Functionality

* The software should validate all user inputs for correct data formats and fill required fields.
* The software should maintain a record of product modifications, order transactions, and administrative activities for audit trails to support traceability and compliance

## Usability

* AIMS software must be of an intuitive layout so that new users can familiarize themselves with ease.
* The functions need to be designed for ease of operation.
* The system must have tooltips and help documentation for essential features.

## Reliability

* The system should run 24/7 and support up to 1,000 customers concurrently without degradation in performance.
* AIMS should run for at least 300 hours continuously before maintenance is needed.
* In the event of failure, the system should be back to normal within 1 hour.
* Ordering and payment processing should be atomic to avoid inconsistencies in data.

## Performance

* Response time should not exceed 2 seconds per action under normal conditions. During peak hours, response time should not exceed 5 seconds.
* Searching for products should produce results in 2 seconds.
* Processing an order, from checkout to payment confirmation, should complete within 10 seconds.
* The system should be able to handle product stock checks effectively, updating real-time inventory.

## Supportability

* The software should support database backups and restoration.
* At the same time, whenever customers need to upgrade or maintain any module, the development team will support them.

## Other requirements

* Payment transactions must be securely processed via VNPay.
* Shipping fees should be dynamically calculated based on product weight and customer location.
* Rush order delivery should only be available for eligible products within Hanoi's inner districts.
* Refunds for canceled/rejected orders must be processed via VNPay within a predefined timeframe.