HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

School of Information and communications technology

Software Requirement Specification

AIMS – An Internet Media Store

ITSS Software Development

**Group 08**

Vũ Quyền Gia Linh - 20225734

Chu Vĩnh Khang - 20225863

Cao Quang Huy - 20225724

Nguyễn Đức Nghĩa - 20225895

Nguyễn Trần Việt Bách - 20198206

*Hanoi,* *June 2025*

Contents

[Table of contents 1](#_Toc201701177)

[1 Introduction 2](#_Toc201701178)

[1.1 Objective 2](#_Toc201701179)

[1.2 Scope 2](#_Toc201701180)

[1.3 Glossary 3](#_Toc201701181)

[1.4 References 4](#_Toc201701182)

[2 Overall Description 5](#_Toc201701183)

[2.1 Survey 5](#_Toc201701184)

[2.2 Overall requirements 5](#_Toc201701185)

[2.3 Business process 6](#_Toc201701186)

[*2.3.1* *Business operation – “Place an order”* 7](#_Toc201701187)

[*2.3.2* *Business operation – “Manage products.”* 8](#_Toc201701188)

[*2.3.3* *Business operation – “Manage users.”* 9](#_Toc201701189)

[3 Detailed Requirements 10](#_Toc201701190)

[3.1 Use case “Place order” 10](#_Toc201701191)

[3.2 Use case “Pay order” 13](#_Toc201701192)

[3.3 Use case “Place rush order” 14](#_Toc201701193)

[3.4 Use case “Create product” 15](#_Toc201701194)

[3.5 Use case “Update product” 17](#_Toc201701195)

[3.6 Use case “Delete product” 18](#_Toc201701196)

[3.7 Use case “Update order” 19](#_Toc201701197)

[3.8 Use case “Update account” 20](#_Toc201701198)

[3.9 Use case “Create account” 22](#_Toc201701199)

[3.10 Use case “Search Product” 23](#_Toc201701200)

[3.11 Use case “Sort Product” 24](#_Toc201701201)

[3.12 Use case “Add product in cart” 25](#_Toc201701202)

[3.13 Use case “Delete product in cart” 26](#_Toc201701203)

[3.14 Use case “View cart” 27](#_Toc201701204)

[3.15 Use case “Login” 27](#_Toc201701205)

[3.16 Use case “Change profile informations” 28](#_Toc201701206)

[4 Supplementary specification 30](#_Toc201701207)

[4.1 Functionality 30](#_Toc201701208)

[4.2 Usability 30](#_Toc201701209)

[4.3 Reliability 30](#_Toc201701210)

[4.4 Performance 31](#_Toc201701211)

[4.5 Supportability 31](#_Toc201701212)

[4.6 Other requirements 31](#_Toc201701213)

# 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.

## 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 program is capable of catering to 1,000 clients concurrently with minimal impact on performance and can run uninterrupted for 300 hours without any issues. Moreover, it can return to regular functioning within a maximum of 1 hour following an incident. The software's response time ranges from 2 seconds under typical circumstances to 5 seconds during periods of peak activity.

In AIMS, customers can not only search for products, but also sort products as their desire, they can place order or rush order for necessary cases. AIMS is supported for 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 privilege problems of users.

Besides, for a desktop website, the need for graphical user interface is also under consideration, which can meet the requirements of end users and enhance the experience of users. Throughout the development stage, every document is also recorded for the future maintenance and upgrading. We keep our focus on every stage to supervise the timeline of the client provided and the quality the software may deliver. If any change is made, our team will adapt quickly to revise our work.

## Glossary

| ***No*** | ***Term*** | ***Explanation*** | ***Example*** | ***Note*** |
| --- | --- | --- | --- | --- |
| 1 | Session | A session is a temporary period of interaction or engagement between a user and a system, during which the user accesses and interacts with software or a website, and the system maintains relevant settings and informations. | software session |  |
| 2 | VAT (Value-added tax) | It is a type of consumption tax that is levied on the value added to goods and services at each stage of production or distribution. VAT is typically implemented as a percentage of the final selling price of a product or service, and it is collected by businesses on behalf of the government. |  |  |
| 3 | API (Application Programming Protocol) | API is a set of rules, protocols, and tools that allows different software apps to communicate with each other. | VNPay API | AIMS connects API of VNPay for transaction. |
| 4 | Payment gateway | 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) | 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, and windows. |  |  |
| 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

This section describes the survey of overall description of AIMS software, which includes the stakeholders, the main functionalities of the software. At the same time, this also delivers the main business processes of the software by illustrating activity diagrams.

## 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.

In the systems includes three main 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. 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. In addition, they can also modify the orders’s status.

- Administrator: They can gain access to mange user in the system. They can also block or unblock the user (customer, product manager). 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.

Besides, VNPay is also a stakeholder in the system when they supply the API to make the transaction in the AIMS software.

## Overall requirements

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

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.

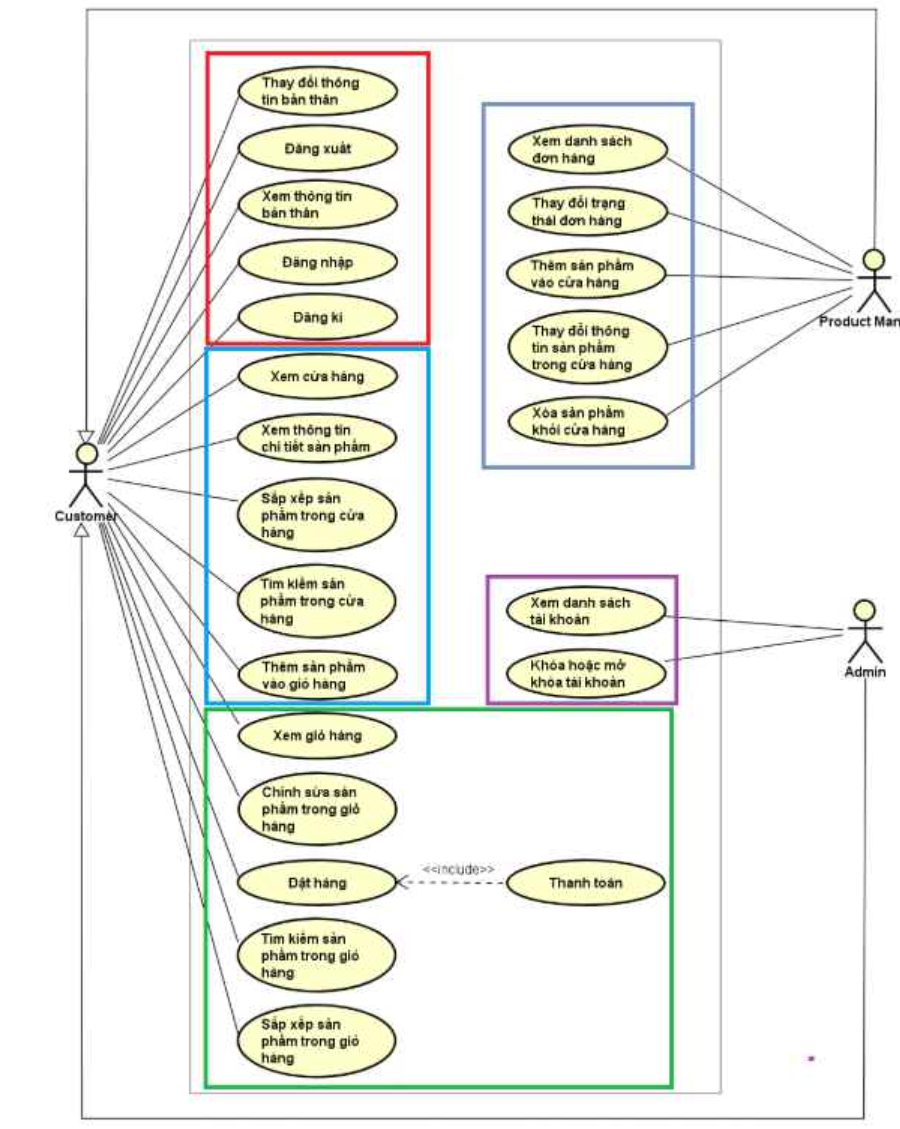


Figure 1. General Use Case Diagram

## Business process

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

Activity diagrams are used for business processes because they provide a clear visual representation of workflows, decision points, and resource allocation. They help stakeholders understand, analyze, and optimize complex processes by illustrating the sequence of activities, concurrency, error handling, and integration with other models. They serve as effective documentation tools and facilitate communication and collaboration among stakeholders.

### ***Business operation – “Place an order”***

A diagram of a project

Description automatically generated

Figure 2: Business process - "Place an order" by customer.

### ***Business operation – “Manage products.”***

A screenshot of a computer

Description automatically generated

Figure 3: Business process - "Manage products" by product manager.

### ***Business operation – “Manage users.”***

A screenshot of a diagram

Description automatically generated

Figure 4: Business process - "Manage users" by admin.

# Detailed Requirements

Detailed requirements typically include specific descriptions of the functionalities, features, user interactions, system behavior, performance criteria, constraints, and dependencies of the software system. These requirements serve as the foundation for software development and encompass both functional and non-functional aspects of the system.

## Use case “Place order”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “PLACE ORDER”**   1. **Use case code**   UC001   1. **Brief Description**   This use case describes the interaction between the software and the customer when the customer wants to place an order.   1. **Actors**   - Customer   1. **Preconditions**   Customer must be in their working session and the cart is not empty.   1. **Basic Flow of Events** 2. Customer requests to place order in the cart 3. AIMS software displays the form of delivery information 4. Customer enters and submits delivery information (see Table 1) 5. AIMS software calculates and displays order information and total cost included shipping fee (see Table 2) 6. The customer choose payment method 7. The AIMS software calls UC “Pay order” 8. The AIMS software creates a new order 9. The AIMS software displays the successful order notification, the order and the transaction information (see Table 3). 10. **Alternative flows**   Table A-Alternative flows of events for UC “Place Order”   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | |  | At Step 3 | If the delivery info is invalid | * AIMS software notifies that the delivery info is invalid (blank or wrong format) | At Step 2 | |  | At Step 3 | If the user chooses to place a rush order | * AIMS software calls use case “Place rush order” | At Step 4 | |  | At Step 5 | If the user chooses payment method as ‘Cash on Delivery’ | * AIMS software skip Step 6 |  | |  | At Step 6 | If the order payment is not successul or goes back from payment |  | At Step 1 |  1. **Input data**   Table 1 - Input data of delivery information   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | |  | Recipient name |  | Yes |  | Chu Vinh Khang | |  | Email |  | Yes | * Email format | khang@gmail.com | |  | Phone number |  | Yes | * 10 digits * Start with ‘0’ | 0912258368 | |  | Delivery Address |  | Yes |  | Số 1 Đại Cồ Việt, Hai Bà Trưng | |  | Province/City |  | Yes |  | Hà Nội | |  | Rush Delivery | Checkbox | No |  |  | |  | Rush Delivery Time |  | No | * yyyy-MM-dd HH:mm | 2025-05-05 12:22 |  1. **Output data**   Table 2-Output data of order information and shipping fee   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Display format** | **Example** | |  | Recipient |  |  | Chu Vinh Khang | |  | Email |  |  | khang@gmail.com | |  | Phone Number |  |  | 0912258368 | |  | Address |  |  | Số 1 Đại Cồ Việt, Hai Bà Trưng, Hà Nội | |  | Rush Delivery | Rush Delivery status | * ‘Yes’ or ‘No’ | No | |  | Delivery Fee | Total delivery fee | * Positive integer * Currency | 7500 VND | |  | Items | List of order’s items |  |  | |  | Price | Price of the corresponding media product | * Positive integer * Currency | 123000 VND | |  | Quantity | Quantity of the corresponding media | * Positive integer | 2 | |  | Amount | Total money of the order | * Positive integer * Currency | 246000 VND | |  | Payment Method | Option for payment method include ‘VNPay’ and ‘Cash On Delivery’ | * Drop-down list | VNPay |   Table 3-Output data of general information of order and transaction info   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Display format** | **Example** | | 1. | Phone Number |  |  | 0912258368 | | 2. | Address |  |  | Số 1 Đại Cồ Việt, Hai Bà Trưng, Hà Nội | | 3. | Rush Delivery | Rush Delivery status | * ‘Yes’ or ‘No’ | No | | 4. | Delivery Fee | Total delivery fee | * Positive integer * Currency | 7500 VND | | 5. | Items | List of order’s items |  |  | | 6. | Price | Price of the corresponding media product | * Positive integer * Currency | 123000 VND | | 7. | Quantity | Quantity of the corresponding media | * Positive integer | 2 | | 8. | Amount | Total money of the order | * Positive integer * Currency | 246000 VND | | 9. | Payment Method | Option for payment method include ‘VNPay’ and ‘Cash On Delivery’ |  | VNPay |  1. **Postconditions**   A new order is created with pending status. |

## Use case “Pay order”

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “PAY ORDER”**   1. **Use case code**   UC002   1. **Brief Description**   This use case describes the interaction between the software, VNPay and the customer when the customer wants to pay for an order.   1. **Actors**    1. **Customer**    2. **VNPay** 2. **Preconditions**   The cart contains products. The user has entered shipping information (recipient name, address, email, phone number). The system is ready to calculate shipping costs.   1. **Basic Flow of Events** 2. AIMS software redirects to VNPay with payment information 3. VNPay notifies the transaction result 4. AIMS software saves the payment transaction 5. **Alternative flows**   Table A-Alternative flows of events for UC “Pay Order”   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | | 1. | At Step 1 | If the customer cancels the payment transaction | * AIMS software records it as failed transaction | End |  1. **Input data** 2. **Output data** 3. **Postconditions**   A record in the orders table (initial state PENDING). The products are stored in order\_items. If payment is made via VNPay, transactions record the transaction code and amount. The user's shopping cart is deleted. |

## 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 software and the customer when the customer wants to place an order with rush order.   1. **Actors**    1. **Customer** 2. **Preconditions**   Customer must be in their working session; the cart is not empty, and the customer’s information is valid.   1. **Basic Flow of Events** 2. Customer requests to place rush order 3. AIMS software checks whether the delivery address supports this service and the eligibility of the products. 4. AIMS software displays the form for choosing products for rush order (Table 1) 5. Return to UC “Place Order” 6. **Alternative flows**   Table A-Alternative flows of events for UC “Place Rush Order”   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | | 1. | At Step 2 | If the no products are not available or address is not eligible for rush order | * The AIMS software notifies that the products in the cart are not available or address is not supported and stay at the use case “View cart” | End |  1. **Input data**   Table 1-Input data of choosing product   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | | 1. | List of Products | Checkbox for choosing product for rush order | Yes |  | “Pulp Fiction” |  1. **Output data** 2. **Postconditions**   None. |

Activity diagram for use case “Place order with rush order”

## Use case “Create product”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Create product”**   1. **Use case code**   UC004   1. **Brief Description**   This use case describes the interaction between the product manager and the product when there is a need to create more 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 requests the creation of a new product 3. The system displays a form for the manager to enter product information (Table 1) 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 “Create product”   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | | 7. | At Step 6 | 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 1-Input data   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | | 1. | Title | Title of a media product | Yes |  | Sample DVD | | 2. | Price | Price of the corresponding media product | Yes |  | 20.000 | | 3. | Stock Quantity | Quantity of the corresponding media | Yes |  | 10 | | 4. | Weight | Weight of the product | Yes |  | 2 | | 5. | Rush Eligible | Does the product support fast delivery? | No |  |  | | 6. | Value |  | Yes |  |  | | 10. | Category | Drop-down list of valid category | Yes |  |  | | 11. | Special information fields for the product type |  | Yes |  |  |  1. **Output data**   No   1. **Postconditions**   A new product will be created |

## Use case “Update product”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Update product”**   1. **Use case code**   UC005   1. **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**    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 and the manager can edit it (Table 1) 4. The manager fills in the information that needs to be corrected 5. The system checks the validity of the entered information 6. The system checks the validity of entered product information 7. The system updates information, announces success and returns to the product management screen 8. **Alternative flows**   Table A-Alternative flows of events for UC “Update Product”   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | |  |  |  |  |  |  1. **Input data**   Table 1-Input data   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | | 1. | Title | Title of a media product | Yes |  | Sample DVD | | 2. | Price | Price of the corresponding media product | Yes |  | 20.000 | | 3. | Stock Quantity | Quantity of the corresponding media | Yes |  | 10 | | 4. | Weight | Weight of the product | Yes |  | 2 | | 5. | Rush Eligible | Does the product support fast delivery? | No |  |  | | 6. | Value |  | Yes |  |  | | 10. | Category | Drop-down list of valid category | Yes |  |  | | 11. | Special information fields for the product type |  | Yes |  |  |  1. **Output data**   No   1. **Postconditions**   New product updated successfully |

## Use case “Delete product”

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Delete product”**   1. **Use case code**   UC006   1. **Brief Description**   This use case describes the interaction between the product manager and the product when there is a need to delete product   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 some products he wants to delete 3. The system displays the delete button 4. The manager presses the delete button 5. The system notifies that the product has been deleted successfully 6. **Alternative flows**   Table A-Alternative flows of events for UC “Delete product”   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | |  |  |  |  |  |  1. **Input data**   No   1. **Output data**   No   1. **Postconditions**   No |

## Use case “Update order”

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Update order”**   1. **Use case code**   UC007   1. **Brief Description**   This use case describes the interaction between the product manager and the order when there is a need to update order   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 order to update 3. The manager updates the order status 4. The manager presses the update button 5. The system updates order information into the database 6. The system sends an email to the email address of the updated order 7. **Alternative flows**   Table A-Alternative flows of events for UC “Update Order”   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | |  |  |  |  |  |  1. **Input data**   Update order status   1. **Output data**   No   1. **Postconditions**   No |

## Use case “Update account”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Update account”**   1. **Use case code**   UC008   1. **Brief Description**   This use case describes the interaction between the admin and the account when there is a need to update the account.   1. **Actors**    1. **Admin** 2. **Preconditions**   You must log in with an admin account.   1. **Basic Flow of Events** 2. The administrator selects an account in the system 3. The administrator chooses to update the account 4. Administrator enters updated information (Table 1) 5. The system checks the information and reports an error if the information is invalid 6. The system updates account information in the database 7. The system sends an email to the account's email address that has just been updated 8. The system notifies the user that their account has been successfully updated 9. **Alternative flows**   Table A-Alternative flows of events for UC “Update account”   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | |  | At Step 4 | The administrator entered incorrect update information | * The system opens a dialog with incorrect information when the administrator presses the update button. | Use case ends |  1. **Input data**   Table 1-Input data   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | | 1. | Password |  | No |  | 12345 | | 2. | Email |  | No |  | a@gmail.com | | 3. | Role | Drop-down list of available roles | Yes |  |  | | 4. | Blocked Status | Account is blocked or not | Yes |  | 1 |  1. **Output data**   No   1. **Postconditions**   An account will be updated and its information will be sent via email or nothing if the update fails. |

## Use case “Create account”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Create account”**   1. **Use case code**   UC009   1. **Brief Description**   This use case describes the interaction between the admin and the account when there is a need to create account.   1. **Actors**    1. **Admin** 2. **Preconditions**   You must log in with an admin account.   1. **Basic Flow of Events** 2. The system displays the account creation form (Table 1) 3. The administrator enters information into the form 4. The administrator presses the submit form button 5. The system create account information in the database 6. **Alternative flows**   Table A-Alternative flows of events for UC “Create account”   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | |  |  |  |  |  |  1. **Input data**   Table 1-Input data   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | | 1. | Username |  | Yes |  | name | | 2. | Password |  | Yes |  | 12345 | | 3. | Email |  | Yes |  | a@gmail.com | | 4. | Role | Drop-down list of available roles | Yes |  | Customer |  1. **Output data**   No   1. **Postconditions**   No |

## Use case “Search Product”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Search Product”**   1. **Use case code**   UC011   1. **Brief Description**   This use case describes the interaction between the customer and the system when there is a need to search product.   1. **Actors**    1. **Customer** 2. **Preconditions** 3. **Basic Flow of Events** 4. Customer enters the keyword to search in the search box 5. Customer press Search 6. The system processes search requests and displays search results to users 7. **Alternative flows**   Table A-Alternative flows of events for UC “Search product”   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | |  |  |  |  |  |  1. **Input data**   Table 1-Input data   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | | 1. | Product Name |  | Yes |  | “Pulp Fiction” |  1. **Output data**   No   1. **Postconditions**   No |

## Use case “Sort Product”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Sort Product”**   1. **Use case code**   UC012   1. **Brief Description**   This use case describes the interaction between the customer and the system when there is a need to sort product.   1. **Actors**    1. **Customer** 2. **Preconditions**   You must log in with a guest account.   1. **Basic Flow of Events** 2. Customer Click on Dropdown list Sort 3. Customer selects Price 4. The system handles arrangement requests 5. The system displays search results by price from low to high 6. **Alternative flows**   Table A-Alternative flows of events for UC “Sort product”   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | |  |  |  |  |  |  1. **Input data**   Table 1-Input data   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | |  |  |  |  |  |  |  1. **Output data**   No   1. **Postconditions**   No |

## Use case “Add product to cart”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Add product to cart”**   1. **Use case code**   UC014   1. **Brief Description**   This use case describes the interaction between the customer and the system when there is a need to add product in cart.   1. **Actors**    1. **Customer** 2. **Preconditions**   You must log in with a guest account.   1. **Basic Flow of Events** 2. Customer launches AIMS system 3. The system displays a list of products for customers to choose from 4. Customers choose to add products to the cart 5. Customer clicks on the shopping cart icon 6. The system displays a list of products the customer has added to the cart 7. **Alternative flows**   Table A-Alternative flows of events for UC “Add product to cart”   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | |  |  |  |  |  |  1. **Input data**   Table 1-Input data   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | |  |  |  |  |  |  |  1. **Output data**   No   1. **Postconditions**   No |

## Use case “Delete product in cart”

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Delete product in cart”**   1. **Use case code**   UC015   1. **Brief Description**   This use case describes the interaction between the customer and the system when there is a need to delete product in cart.   1. **Actors**    1. **Customer** 2. **Preconditions**   You must log in with a guest account.   1. **Basic Flow of Events** 2. Customer Click on the shopping cart icon on the main screen 3. The system displays the products in the shopping cart 4. Customers click on the delete button next to the product 5. The system will delete the product from the shopping cart 6. **Alternative flows**   Table A-Alternative flows of events for UC “Delete product in cart”   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | |  |  |  |  |  |  1. **Input data**   No   1. **Output data**   No   1. **Postconditions**   No |

## Use case “View cart”

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “View cart”**   1. **Use case code**   UC016   1. **Brief Description**   This use case describes the interaction between the customer and the system when there is a need to view cart.   1. **Actors**    1. **Customer** 2. **Preconditions**   You must log in with a guest account.   1. **Basic Flow of Events** 2. Customer Click on the shopping cart icon on the main screen 3. The system displays the products in the shopping cart 4. **Alternative flows**   Table A-Alternative flows of events for UC “View cart”   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | |  |  |  |  |  |  1. **Input data**   No   1. **Output data**   No   1. **Postconditions**   No |

## Use case “Login”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Login”**   1. **Use case code**   UC017   1. **Brief Description**   This use case describes the interaction between the customer and the system when there is a need to login.   1. **Actors**    1. **Admin**    2. **Product manager**    3. **Customer** 2. **Preconditions** 3. **Basic Flow of Events** 4. When running the program, the system will be on the login page 5. The user will fill in the account and password 6. When pressing the login button, the system will switch to the admin interface if it is an admin account or the product manager interface if it is a product manager account, otherwise it will switch to the customer interface 7. **Alternative flows**   Table A-Alternative flows of events for UC “Login”   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | | 4 | 3.1 | If the user fills in missing information or the account is wrong | The system will report an error | Continue step 2 |  1. **Input data**  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | | 1 | Username |  | yes |  | user | | 2 | password |  | yes |  | 1234 |  1. **Output data**   No   1. **Postconditions**   No |

## Use case “Update profile information”

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Update profile information”**   1. **Use case code**   UC018   1. **Brief Description**   This use case describes the interaction between the user and the system when there is a need to change profile information.   1. **Actors**    1. **Customer** 2. **Preconditions** 3. **Basic Flow of Events** 4. User click on the profile button on main screen 5. User field the profile information form 6. The system checks the information and reports an error if the information is invalid 7. The system updates account information into the database 8. The system sends an email to the email address of the newly updated account 9. **Alternative flows**   Table A-Alternative flows of events for UC “Update profile information”   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | |  |  |  |  |  |  1. **Input data**  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | | 1 | Username |  | Yes |  | user | | 2 | Email |  | Yes |  | 1234 | | 3 | New Password |  | Yes |  | 1234 |  1. **Output data**   No   1. **Postconditions**   No |

# 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

- Use cases related to transactional operations, if errors occur during the connection or operation process, need to provide corresponding error notifications so that actors know that the error is related to the system and not the user's fault.

- Use cases related to steps involving registration, account recovery, and password change need to request accurate, secure, reliable confirmation, with at least 2 methods to confirm the user's account. Avoid situations where there is insufficient information to confirm the account.

- Use cases used by Product manager and Customer, guest users need to log in with corresponding roles.

## Usability

- The functions need to be designed for ease of operation.

- Convenient layout, easy to operate.

- The language used should be understandable, avoiding the use of too much technical jargon.

## 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: Robust error handling mechanisms detect and log errors, with alerts sent to administrators for immediate resolution. The system gracefully handles errors to prevent service disruptions and provide a seamless user experience.

- Testing and Verification: Rigorous testing procedures, including unit tests, integration tests, and stress tests, verify the system's reliability under various conditions. Automated monitoring tools continuously monitor system performance and alert administrators to potential issues.

- Documentation and Reporting: Comprehensive documentation outlines system architecture, failover procedures, and incident response protocols. Incident reports document reliability incidents and resolutions, facilitating continuous improvement and accountability.

## Performance

- Concurrent Capacity: The program can handle up to 1,000 clients simultaneously with minimal performance impact.

- Continuous Operation: It can run continuously for 300 hours without encountering any issues.

- Recovery Time: In the event of an incident, the program can return to regular functioning within a maximum of 1 hour.

- Response Time: The software's response time varies from 2 seconds under typical circumstances to 5 seconds during periods of peak activity.

## Supportability

This software is supported for Windows platform desktop, and computer. At the same time, whenever clients need to upgrade or maintain any module, then the development team will support them.

## Other requirements

Beautiful graphical user interface, easy to use, fast processing speed, accurate. Reasonable product categorization, easy to search.