HANOI UNIVERSITY OF SCIENCE AND TECHNOLOGY

School of Information and communications technology

Software Requirement Specification

Version 1.2

CLOUDY STORE

Subject: Product selling website

<Group 25>

<List of participants>

Nguyễn Ngọc Trường Xuân-20225958

Lê Hoàng Thành Vinh-20225778

Phạm Tiến Đại-20225799

Nguyễn Đức Tín-20225935

*Hanoi,* *< July, 2025>*

Table of contents

Table of contents 1

1 Introduction 3

1.1 Objective 3

1.2 Scope 3

1.3 Glossary 3

1.4 References 4

2 Overall Description 5

2.1 Survey 5

2.2 Overall requirements 5

2.3 Business process 7

3 Detailed Requirements 9

3.1 Use case 1 9

3.2 Use case 2 10

3.3 Use case 3 11

3.4 Use case 4 13

3.5 Use case 5 14

3.6 Use case 6 15

3.7 Use case 7 16

3.8 Use case 8 17

3.9 Use case 9 19

3.10 Use case 10 20

3.11 Use case 11 21

3.12 Use case 12 22

3.13 Use case 13 24

3.14 Use case 14 24

3.15 Use case 15 25

3.16 Use case 16 27

3.17 Use case 17 27

3.18 Use case 18 28

3.19 Use case 19 30

3.20 Use case 20 31

3.21 Use case 21 32

3.22 Use case 22 33

3.23 Use case 23 34

3.24 Use case 24 35

3.25 Use case 15 36

4 Supplementary specification 38

4.1 Functionality 38

4.2 Usability 38

4.3 Reliability 38

4.4 Performance 38

4.5 Supportability 38

4.6 Other requirements 38

# Introduction

## Objective

This document provides the overall requirements of the software system project. The main purpose of this document is to provide an overview of the final project’s behavior for the class lecturers. Other than that, this document provides future users or developers a description about the current state of the system.

## Scope

Cloudy Store is a application software used to manage an online store. The system has some basic function like add and manage selling items to store owner, as well as add, adjust items in cart and place order for the customers.

With the rapid improvement of technology and the increasing demands of online shopping. Cloudy Store is built to be a platform in which the shop owner can sell their product to the customer regardless of distance.

## Glossary

| ***No*** | ***Term*** | ***Explanation*** | ***Example*** | ***Note*** |
| --- | --- | --- | --- | --- |
| 1 | Software Requirement Specification (SRS) | Description of the system requirement |  |  |
| 2 | Use Case(s) | Interaction(s) between the users and the system | Login, Add items to cart |  |
| 3 | Cart | List of product personalized for each customer |  |  |
| 4 | Check out | Buying all items in cart |  |  |
| 5 | Navbar | Navigation bar locate on top of the screen used to switch tab |  |  |
| 6 | Database | System’s memory about user account, items in cart,… |  |  |

## References

*<Listing the referenced material used in this documents, including the one related to the project>*

# Overall Description

## Survey

The actors of the system are guests, store owners and customers. Guests are the users that has not login. Customers are users that logged in with customer account. Store owners Aare provided with authority to manage customer accounts as well as selling product in the platform.

## Overall requirements

*A diagram of a customer

AI-generated content may be incorrect.*

**Figure 2-1: Use Case Diagram**

**A diagram of a diagram

AI-generated content may be incorrect.**

**Figure 2-2: Break down Use Case View shop**

**A diagram of a product

AI-generated content may be incorrect.**

**Figure 2-3: Break down Use Case View cart**

**A diagram of a person

AI-generated content may be incorrect.**

**Figure 2-4: Break down owner Use Case**

## Business process

Customers can log in to start working with the system. If they don’t have an account, the system allow them to register a new account. If they forgot their account’s password, they can reset their password. After that, they are given a personal cart to store the product they want to buy. The customer can check out, this will empty out their cart and place an order to their home.

A diagram of a login

AI-generated content may be incorrect.

**Figure 2-5: Activity Diagram**

# Detailed Requirements

## Use case 1

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Log in”**   1. **Use case code**   UC001   1. **Brief Description**   This use case describes the interaction between visitor and Cloudy Store when the visitor wishes to log in their account   1. **Actors**    * Visitor 2. **Preconditions**   **None**   1. **Basic Flow of Events** 2. The visitor presses the log in button in navbar 3. The software navigates visitor to the log in tab 4. The visitor enters log in information (see Table A) 5. The visitor presses the log in button in log in tab 6. The software checks log in information 7. The software displays function according to visitor account 8. **Alternative flows**   Table N-Alternative flows of events for UC Log in   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | |  | At Step 3 | If visitor press register button | * Call “Register” Use Case | Resumes at Step 6 | |  | At Step 3 | If visitor press forgot password button | * Call “Reset password” Use Case | Resumes at Step 3 | |  | At Step 5 | If the email or password is incorrect | * Alert “Wrong email or password” | Resumes at Step 3 | |  | At Step 6 | If the visitor using admin account | * Navigate to admin tab | Use Case ends |  1. **Input data**   Table A-Input data of Step 3   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | |  | Email |  | yes | At least 6 character, contain ‘@’ | vinhle@gmail.com | |  | Password |  | yes | None | 123456 |  1. **Output data** 2. **Postconditions**  * Visitor logged in their account |

## Use case 2

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Register”**   1. **Use case code**   UC002   1. **Brief Description**   This use case describes the interaction between visitor and Cloudy Store when the visitor wishes to create new account   1. **Actors**    * Visitor 2. **Preconditions**   **None**   1. **Basic Flow of Events** 2. The visitor presses register button in the log in tab 3. The software navigates visitor to register tab 4. The visitor enters log in information (see Table A) 5. The visitor presses register button 6. The software checks validity of the account 7. The software displays function according to visitor account 8. **Alternative flows**   Table N-Alternative flows of events for UC Register   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | |  | At Step 3 | If visitor press log in button | * Call “Log In” Use Case | Use Case ends | |  | At Step 5 | If the input information is invalid | * Alert “Invalid information” | Resumes at Step3 |  1. **Input data**   Table A-Input data of Step 3   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | |  | First name |  | yes | None | Vinh | |  | Last name |  | yes | None | Le | |  | Email |  | yes | At least 6 character, contain ‘@’ | abcd@e | |  | Password |  | yes | None | 123456 |  1. **Output data**   **None**   1. **Postconditions**  * Visitor logged in their account * Visitor’s account are archived in Database |

## Use case 3

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Reset password”**   1. **Use case code**   UC003   1. **Brief Description**   This use case describes the interaction between visitor and Cloudy Store when the visitor wishes to reset account’s password   1. **Actors**  * Visitor  1. **Preconditions**   **None**   1. **Basic Flow of Events** 2. The visitor presses reset password button 3. The software requests registered email address 4. The visitor enters email address 5. The software sends verify account email 6. The visitor presses the email to go to reset password tab 7. The visitor enters new password form (see Table A) 8. The software checks new password validity 9. The software resets account password 10. **Alternative flows**   Table N-Alternative flows of events for UC Reset password   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | |  | At Step 4 | If the address is invalid | * Alert “Invalid email” | Resumes at Step 3 | |  | At Step 7 | If password and confirm password field does not match | * Alert “Password and confirm password must be identical” | Resumes at Step 6 |  1. **Input data**   Table A-Input data of Step 6   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | |  | Password | New password | yes | None | 654321 | |  | Confirm password | Password confirmation | yes | None | 654321 |  1. **Output data**   **None**   1. **Postconditions**  * Account is saved with new password |

## Use case 4

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “View shop”**   1. **Use case code**   UC004   1. **Brief Description**   This use case describes the interaction between guest and Cloudy Store when the guest wishes see shop product   1. **Actors**    * Guest 2. **Preconditions**   **None**   1. **Basic Flow of Events** 2. The guest presses cart icon 3. The software navigate guest to shop tab (See table A) 4. **Alternative flows**   **None**   1. **Input data**   **None**   1. **Output data**   Table A-Output data of Step 2   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Display format** | **Example** | |  | Product image | Image of product posted by seller | Image |  | |  | Product name |  | String | Iphone | |  | Product cost |  | Float with VND currency (₫ symbol) | 1,200,000 ₫ | |  | Product description | Seller’s information about product | String | Newest phone on the market |  1. **Postconditions**   **None** |

## Use case 5

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Find item”**   1. **Use case code**   UC005   1. **Brief Description**   This use case describes the interaction between guest and Cloudy Store when the guest wishes to search a product   1. **Actors**    * Guest 2. **Preconditions**   **None**   1. **Basic Flow of Events** 2. The guest clicks search box on the navbav 3. The guest inputs the name of the product (See Table A) 4. The guest clicks the find button 5. The software displays list of products that match the input (See Table B) 6. **Alternative flows**   Table N-Alternative flows of events for UC Find item   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | |  | At Step 4 | If there is no product match input | * Display text “No product found” on screen | Use case ends |  1. **Input data**   Table A-Input data of Step 2   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | |  | Search input |  | no | None | Smart Phone |  1. **Output data**  |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Display format** | **Example** | |  | Product image | Image of product posted by seller | Image |  | |  | Product name |  | String | Iphone | |  | Product cost |  | Float with VND currency (₫ symbol) | 1,200,000 ₫ | |  | Product description | Seller’s information about product | String | Newest phone on the market |  1. **Postconditions**   **None** |

## Use case 6

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Find item by catergory”**   1. **Use case code**   UC006   1. **Brief Description**   This use case describes the interaction between guest and Cloudy Store when the guest wishes to search a product of a specific catergory   1. **Actors**    * Guest 2. **Preconditions**    * The user is in the product tab 3. **Basic Flow of Events** 4. The guest clicks on button that have the category they want 5. The software lists all items of that category (See table A) 6. **Alternative flows**   Table N-Alternative flows of events for UC Find item by category   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | |  | At Step 2 | If there is no item min that category | * Display text “No product found” on screen | Use case ends |  1. **Input data**   **None**   1. **Output data**   Table A-Output data of Step 2   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Display format** | **Example** | |  | Product image | Image of product posted by seller | Image |  | |  | Product name |  | String | Iphone | |  | Product cost |  | Float with VND currency (₫ symbol) | 1,200,000 ₫ | |  | Product description | Seller’s information about product | String | Newest phone on the market |  1. **Postconditions**   **None** |

## Use case 7

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “See item detail”**   1. **Use case code**   UC007   1. **Brief Description**   This use case describes the interaction between guest and Cloudy Store when the guest wishes to see more information about item   1. **Actors**    * Guest 2. **Preconditions**    * The guest is in the product tab 3. **Basic Flow of Events** 4. The guest clicks the “See detail” button 5. The software presents product’s information (See table A) 6. **Alternative flows**   **None**   1. **Input data**   **None**   1. **Output data**   Table A-Output data of Step 2   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Display format** | **Example** | |  | Product image | Image of product posted by seller | Image |  | |  | Product name |  | String | Iphone | |  | Product category |  | String | electronic | |  | Product cost |  | Float with VND currency (₫ symbol) | 1,200,000 ₫ | |  | Product description | Seller’s information about product | String | Newest phone on the market |  1. **Postconditions**   **None** |

## Use case 8

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Add item to cart”**   1. **Use case code**   UC008   1. **Brief Description**   This use case describes the interaction between customer and Cloudy Store when the customer wishes to add item to cart   1. **Actors**    * Customer 2. **Preconditions**    * Customer is in the see detail tab 3. **Basic Flow of Events** 4. The customer select quantity of product that they want (See table A) 5. The customer presses the “Add to cart” button 6. The software adds product in the cart 7. The software navigates to shop tab 8. **Alternative flows**   Table N-Alternative flows of events for UC Add item to cart   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | |  | At Step 1 | If the customer presses “Return” button | * Navigate to shop tab | Use case ends | |  | At Step 3 | If the customer has not logged in | * Call “Log in” Use Case | Resumes at Step 3 |  1. **Input data**   Table A-Input data of Step 1   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | |  | Quantity |  | yes | Integer, greater or equal to 1 | 5 |  1. **Output data**   **None**   1. **Postconditions**    * The item is added to customer’s cart |

## Use case 9

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “View cart”**   1. **Use case code**   UC009   1. **Brief Description**   This use case describes the interaction between customer and Cloudy Store when the customer wishes see their cart   1. **Actors**    * Customer 2. **Preconditions**   **None**   1. **Basic Flow of Events** 2. The customer presses cart icon 3. The software navigate customer to cart tab (See table A) 4. **Alternative flows**   **None**   1. **Input data**   **None**   1. **Output data**   Table A-Output data of Step 2   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Display format** | **Example** | |  | Product image | Image of product posted by seller | Image |  | |  | Product name |  | String | Iphone | |  | Product quantity | Number of product in cart | Integer | 5 | |  | Product cost |  | Float with VND currency (₫ symbol) | 1,200,000 ₫ | |  | Product total cost | Cost multiply by quantity | Float with VND currency (₫ symbol) | 6,000,000 ₫ |  1. **Postconditions**   **None** |

## Use case 10

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “See cart item detail”**   1. **Use case code**   UC010   1. **Brief Description**   This use case describes the interaction between customer and Cloudy Store when the customer wishes to see more information about item in cart   1. **Actors**    * Customer 2. **Preconditions**    * Customer is in cart tab 3. **Basic Flow of Events** 4. The customer clicks “see detail” button 5. The software display item detail (See table A) 6. **Alternative flows**   **None**   1. **Input data**   **None**   1. **Output data**   Table A-Output data of Step 2   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Display format** | **Example** | |  | Product image | Image of product posted by seller | Image |  | |  | Product name |  | String | Iphone | |  | Product category |  | String | electronic | |  | Product cost |  | Float with VND currency (₫ symbol) | 1.200.000 ₫ | |  | Product description | Seller’s information about product | String | Newest phone on the market |  1. **Postconditions**   **None** |

## Use case 11

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Adjust cart item quantity”**   1. **Use case code**   UC011   1. **Brief Description**   This use case describes the interaction between customer and Cloudy Store when the customer wishes to change cart item’s quantity   1. **Actors**    * Customer 2. **Preconditions**    * Customer is in cart item detail tab 3. **Basic Flow of Events** 4. The customer update cart item quantity (See table A) 5. The customer presses the “Update quantity button” 6. The software adjust cart item’s quantity 7. The software navigates to cart tab 8. **Alternative flows**   Table N-Alternative flows of events for UC Adjust cart item quantity   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | |  | At Step 1 | If customer presses “Return” button | * Navigate to cart tab | Use case ends |  1. **Input data**   Table A-Input data of Step 1   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | |  | Quantity |  | yes | Integer, greater or equal to 1 | 5 |  1. **Output data**   **None**   1. **Postconditions**    * The item in cart is updated with new quantity |

## Use case 12

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Place order”**   1. **Use case code**   UC012   1. **Brief Description**   This use case describes the interaction between customer and Cloudy Store when the customer wishes to buy items   1. **Actors**    * Customer 2. **Preconditions**    * The customer is in cart tab    * There is at least 1 item in cart 3. **Basic Flow of Events** 4. The customer presses “Place order” button 5. The software navigates to checkout tab 6. The customer enters required information (see Table A) 7. The customer presses “Pay with VN Pay” button 8. The software checks information 9. The software connects with VN Pay to process payment 10. VN Pay confirms payment 11. The software empties customer’s cart 12. **Alternative flows**   Table N-Alternative flows of events for UC Place order   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | |  | At Step 3 | If information is invalid | * Alert “Confirmation failed” | Resumes at Step 3 | |  | At Step 3 | If customer selects coupon | * Recaculate total cost * Print new cost on screen | Resumes at Step 4 | |  | At Step 7 | If VN Pay does not confirm payment | * Alert “Payment canceled” | Use case ends |  1. **Input data**   Table A-Input data of Step 3   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | |  | First name |  | yes | String | Tuan | |  | Last name |  | yes | String | Tran | |  | Address |  | yes | None | 83, Hai Ba Trung | |  | Alternate address | Another address for unexpected situation | no | None | 84, Ba Trieu | |  | City |  | yes | None | Ha Noi | |  | Postal code |  | yes | 6 digits | 123456 | |  | Email |  | yes | At least 6 character, includes ‘@’ | qdas@f | |  | Phone number |  | yes | 10 digits, begin with 0 | 0328934643 | |  | Note | Other request for the seller | no | None | Fragile product |  1. **Output data**   **None**   1. **Postconditions**    * Customer’s cart is empty |

## Use case 13

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “View product category”**   1. **Use case code**   UC013   1. **Brief Description**   This use case describes the interaction between owner and Cloudy Store when the owner wishes to view item catergory   1. **Actors**    * Owner 2. **Preconditions**    * The owner logged in with admin account 3. **Basic Flow of Events** 4. The owner selects “Manage category” button 5. The software navigates to category tab (See table A) 6. **Alternative flows**   **None**   1. **Input data**   **None**   1. **Output data**   Table A-Output data of Step 2   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Display format** | **Example** | |  | Category name |  | String | electronic |  1. **Postconditions**    * The owner is in manage category tab |

## Use case 14

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Create new category”**   1. **Use case code**   UC014   1. **Brief Description**   This use case describes the interaction between owner and Cloudy Store when the owner wishes to make new item catergory   1. **Actors**    * Owner 2. **Preconditions**    * The owner logged in with admin account 3. **Basic Flow of Events** 4. The owner presses “Add category” button 5. The software navigates the owner to add category tab 6. The owner enters category name (see Table A) 7. The software add new category 8. **Alternative flows**   Table N-Alternative flows of events for UC Create new category   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | |  | At Step 3 | If owner presses “Return” button | * Navigate to category tab | Use case ends | |  | At Step 4 | If the category name is empty | * Alert “Enter category name” | Resumes at Step 3 |  1. **Input data**   Table A-Input data of Step 3   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | |  | Category name |  | yes | None | electronic |  1. **Output data**   **None**   1. **Postconditions**    * New category is saved |

## Use case 15

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Manage category”**   1. **Use case code**   UC015   1. **Brief Description**   This use case describes the interaction between owner and Cloudy Store when the owner wishes to change item category name   1. **Actors**    * Owner 2. **Preconditions**    * The owner logged in with admin account    * There is at least one category 3. **Basic Flow of Events** 4. The owner selects the “Modify” button corresponding with modifying category 5. The software navigates the owner to modify tab 6. The owner enters new category name (see Table A) 7. The software displays new category name 8. **Alternative flows**   Table N-Alternative flows of events for UC Manage category   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | |  | At Step 3 | If owner presses “Return” button | * Navigate to category tab | Use case ends | |  | At Step 4 | If the category name is empty | * Alert “Enter category name” | Resumes at Step 3 |  1. **Input data**   Table A-Input data of Step 3   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | |  | Category name |  | yes | None | electronic |  1. **Output data**   **None**   1. **Postconditions**    * New category name is saved |

## Use case 16

|  |
| --- |
| **Use Case “Remove category”**   1. **Use case code**   UC015   1. **Brief Description**   This use case describes the interaction between owner and Cloudy Store when the owner wishes to remove item category   1. **Actors**    * Owner 2. **Preconditions**    * The owner logged in with admin account    * There is at least one category 3. **Basic Flow of Events** 4. The owner selects the “Delete” button corresponding with modifying category 5. The software remove that category 6. **Alternative flows**   **None**   1. **Input data**   **None**   1. **Output data**   **None**   1. **Postconditions**    * The selected category is deleted |

## Use case 17

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “View product”**   1. **Use case code**   UC017   1. **Brief Description**   This use case describes the interaction between owner and Cloudy Store when the owner wishes to view item in shop   1. **Actors**    * Owner 2. **Preconditions**    * The owner logged in with admin account 3. **Basic Flow of Events** 4. The owner selects “Manage product” button 5. The software navigates to product tab (See table A) 6. **Alternative flows**   **None**   1. **Input data**   **None**   1. **Output data**   Table A-Output data of Step 2   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Display format** | **Example** | |  | Product name |  | String | Laptop | |  | Category |  | String | electronic | |  | Preview | Product image | Image |  |  1. **Postconditions**    * The owner is in manage product tab |

## Use case 18

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Add new product”**   1. **Use case code**   UC018   1. **Brief Description**   This use case describes the interaction between owner and Cloudy Store when the owner wishes to add new item in shop   1. **Actors**    * Owner 2. **Preconditions**    * The owner logged in with admin account 3. **Basic Flow of Events** 4. The actor(s) presses “Add product” button 5. The software navigates the owner to add product tab 6. The actor enters product information (see Table A) 7. The software add new product 8. **Alternative flows**   Table N-Alternative flows of events for UC Add new product   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | |  | At Step 3 | If owner presses “Return” button | * Navigate to product tab | Use case ends | |  | At Step 4 | If any product information field is empty | * Alert “Enter all required field” | Resumes at Step 3 |  1. **Input data**   Table A-Input data of Step 3   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | |  | Product name |  | yes | String | Smart phone | |  | Category |  | yes | Selected from a list | electronic | |  | Price |  | yes | Vietnamese currency | 1,000,000 ₫ | |  | Weight |  | yes | Float, gram unit | 3.4 gram | |  | Product image |  | yes | Image |  | |  | Description |  | no | String | New product |  1. **Output data**   **None**   1. **Postconditions**    * New product is saved |

## Use case 19

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Manage product property”**   1. **Use case code**   UC019   1. **Brief Description**   This use case describes the interaction between owner and Cloudy Store when the owner wishes to change item property   1. **Actors**    * Owner 2. **Preconditions**    * The owner logged in with admin account    * There is at least one product 3. **Basic Flow of Events** 4. The owner selects the “Modify” button corresponding with modifying product 5. The software navigates the owner to modify tab 6. The actor enters new product property (see Table A) 7. The software displays new product property 8. **Alternative flows**   Table N-Alternative flows of events for UC Manage product property   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | |  | At Step 3 | If owner presses “Return” button | * Navigate to product tab | Use case ends | |  | At Step 4 | If any product information field is empty | * Alert “Enter all required field” | Resumes at Step 3 |  1. **Input data**   Table A-Input data of Step 3   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | |  | Product name |  | yes | String | Smart phone | |  | Category |  | yes | Selected from a list | electronic | |  | Price |  | yes | Vietnamese currency | 1,000,000 ₫ | |  | Weight |  | yes | Float, gram unit | 3.4 gram | |  | Product image |  | yes | Image |  | |  | Description |  | no | String | New product |  1. **Output data**   **None**   1. **Postconditions**    * New product property is saved |

## Use case 20

|  |
| --- |
| **Use Case “Remove product”**   1. **Use case code**   UC020   1. **Brief Description**   This use case describes the interaction between owner and Cloudy Store when the owner wishes to remove product   1. **Actors**    * Owner 2. **Preconditions**    * The owner logged in with admin account    * There is at least one product 3. **Basic Flow of Events** 4. The owner selects the “Delete” button corresponding with modifying product 5. The software remove that product 6. **Alternative flows**   **None**   1. **Input data**   **None**   1. **Output data**   **None**   1. **Postconditions**    * The selected product is deleted |

## Use case 21

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “View user account”**   1. **Use case code**   UC021   1. **Brief Description**   This use case describes the interaction between owner and Cloudy Store when the owner wishes to view user account   1. **Actors**    * Owner 2. **Preconditions**    * The owner logged in with admin account 3. **Basic Flow of Events** 4. The owner selects “Manage user” button 5. The software navigates to user tab (See table A) 6. **Alternative flows**   **None**   1. **Input data**   **None**   1. **Output data**   Table A-Output data of Step 2   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Display format** | **Example** | |  | Email |  | String | Vinhle@gmail.com |  1. **Postconditions**    * The owner is in manage user tab |

## Use case 22

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Add new account”**   1. **Use case code**   UC022   1. **Brief Description**   This use case describes the interaction between owner and Cloudy Store when the owner wishes to add new user account   1. **Actors**    * Owner 2. **Preconditions**    * The owner logged in with admin account 3. **Basic Flow of Events** 4. The actor(s) presses “Add user” button 5. The software navigates the owner to add user tab 6. The actor enters user information (see Table A) 7. The software add new user 8. **Alternative flows**   Table N-Alternative flows of events for UC Add new account   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | |  | At Step 3 | If owner presses “Return” button | * Navigate to user tab | Use case ends | |  | At Step 4 | If any user information field is empty | * Alert “Enter all required field” | Resumes at Step 3 |  1. **Input data**   Table A-Input data of Step 3   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | |  | Email |  | yes | At least 6 digit, include ‘@’ | Vinh@gmail.com | |  | Password |  | yes | None | 123456789 | |  | First name |  | yes | String | Nguyen | |  | Last name |  | yes | String | Xuan | |  | Role | User’s role | yes | User or admin | user |  1. **Output data**   **None**   1. **Postconditions**    * New user is saved |

## Use case 23

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Manage account”**   1. **Use case code**   UC023   1. **Brief Description**   This use case describes the interaction between owner and Cloudy Store when the owner wishes to change account property   1. **Actors**    * Owner 2. **Preconditions**    * The owner logged in with admin account    * There is at least one account 3. **Basic Flow of Events** 4. The owner selects the “Modify” button corresponding with modifying account 5. The software navigates the owner to modify tab 6. The actor enters new account property (see Table A) 7. The software displays new account property 8. **Alternative flows**   Table N-Alternative flows of events for UC Manage account   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | |  | At Step 3 | If owner presses “Return” button | * Navigate to user tab | Use case ends | |  | At Step 4 | If any user information field is empty | * Alert “Enter all required field” | Resumes at Step 3 |  1. **Input data**   Table A-Input data of Step 3   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **No** | **Data fields** | **Description** | **Mandatory** | **Valid condition** | **Example** | |  | Email |  | yes | At least 6 digit, include ‘@’ | Vinh@gmail.com | |  | Password |  | yes | None | 123456789 | |  | First name |  | yes | String | Nguyen | |  | Last name |  | yes | String | Xuan | |  | Role | User’s role | yes | User or admin | user |  1. **Output data**   **None**   1. **Postconditions**    * New account property is saved |

## Use case 24

|  |
| --- |
| **Use Case “Remove account”**   1. **Use case code**   UC024   1. **Brief Description**   This use case describes the interaction between owner and Cloudy Store when the owner wishes to remove account   1. **Actors**    * Owner 2. **Preconditions**    * The owner logged in with admin account    * There is at least one account other than current account 3. **Basic Flow of Events** 4. The owner selects the “Delete” button corresponding with modifying account 5. The software remove that account 6. **Alternative flows**   **None**   1. **Input data**   **None**   1. **Output data**   **None**   1. **Postconditions**    * The selected account is deleted |

## Use case 15

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Use Case “Log out”**   1. **Use case code**   UC015   1. **Brief Description**   This use case describes the interaction between user and Cloudy Store when the user wishes to log out   1. **Actors**    * User 2. **Preconditions**    * The user logged in 3. **Basic Flow of Events** 4. The user presses “Log out” button in navbar 5. The software send alert “Do you wish to log out” 6. The user presses “Confirm” 7. The software logs out of user account 8. **Alternative flows**   Table N-Alternative flows of events for UC Log out   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **No** | **Location** | **Condition** | **Action** | **Resume location** | |  | At Step 3 | If user press “Cancel” | None | Use case ends |  1. **Input data**   **None**   1. **Output data**   **None**   1. **Postconditions**    * The user logged out |

# Supplementary specification

## Functionality

* Anyone should be able to see product without logging in
* The system need to verify user before adding item to cart or paying
* When user add a product that already have in cart, the system will increase the amount
* The system should archive customer’s cart for every customer even when logged out
* The system needs to verify admin before modify shop or account

## Usability

* The shop will display total cost by quantity for user
* The cart need to display product cost in unit, multiply by quantity, cart total cost before and after VAT
* The buttons need to be clearly displayed, with proper instruction

## Reliability

* User’s data need to remain after system malfunction
* User’s password need to be encrypted and protected
* The system need to be stable with 99% uptime
* Mean Time Between Failures (MTBF) need to be greater than 200 hours

## Performance

* The system need to respond within 2 seconds
* The system can provide service for at least 100 people at a time

## Supportability

* Source code need to be easy to understand and easy for maintenance
* Updating each module should not affecting other modules performance

## Other requirements

* Database stored using PostgreSQL, backup data daily