

**Phone-Online-SHop**

**Software Requirement Specification**

– Hanoi, March 2023

–

Record of changeS

|  |  |  |  |
| --- | --- | --- | --- |
| Date | A\* M, D | In charge | Change Description |
| 2/2 | A | Linhtq | Function Requirement(II) |
| 15/2 | A | Linhtq | Add ERD(I.3) |
| 19/2 | M | Linhtq | Update context diagram |
| 20/2 | A | Linhtq | Add screen flow(I.2.a) |
| 20/2 | A | Linhtq | Add screen Detail(I.2.b) |
| 21/2 | M | Thanhdq | Update context diagram |
| 27/2 | M | Linhtq | Update Screen Flow |
| 28/2 | M | Thanhdq | Update context diagram |
| 28/2 | A | Thanhdq | Add Entities Description |
| 16/3 | A | LinhTQ | Add All Funtional requirement in iter1 |
| 18/3 | M | Linhtq | Finish Funtional requirement of all iters |

\*A - Added

\*M - Modified

\*D - Deleted

Table of Contents

[I. Overview 6](#_Toc130292347)

[1.1. Introduction 6](#_Toc130292348)

[1.2. System Functions 7](#_Toc130292349)

[1.3. Entity Relationship Diagram 10](#_Toc130292350)

[II. User Requirements 14](#_Toc130292351)

[2.1. Actors 14](#_Toc130292352)

[2.2. Uses Case Diagram 15](#_Toc130292353)

[2.3. Uses Case Description 16](#_Toc130292354)

[III. Functional Requirements 19](#_Toc130292355)

[3.1 Create comment. 19](#_Toc130292356)

[3.2 View comment. 20](#_Toc130292357)

[3.3 Update comment. 21](#_Toc130292358)

[3.4 Delete comment. 22](#_Toc130292359)

[3.5 View profile account. 24](#_Toc130292360)

[3.6 Update profile account. 25](#_Toc130292361)

[3.7 Change password account. 27](#_Toc130292362)

[3.8 Login. 29](#_Toc130292363)

[3.9 Logout. 30](#_Toc130292364)

[3.10 View home page. 31](#_Toc130292365)

[3.11 View shopping cart. 32](#_Toc130292366)

[3.12 Create order. 33](#_Toc130292367)

[3.13 Update order. 35](#_Toc130292368)

[3.14 Register. 36](#_Toc130292369)

[3.15 Manage Comment. 37](#_Toc130292370)

[3.16 Manage Product. 39](#_Toc130292371)

[3.17 Create Product. 41](#_Toc130292372)

[3.18 View Product Change. 42](#_Toc130292373)

[3.19 Update Product Detail. 43](#_Toc130292374)

[3.20 Update Status. 45](#_Toc130292375)

[3.21 Delete Product. 47](#_Toc130292376)

[3.22 Manage Account. 48](#_Toc130292377)

[3.23 Customer Account. 49](#_Toc130292378)

[3.24 Employee Account. 50](#_Toc130292379)

[3.25 Create Account. 51](#_Toc130292380)

[3.26 Update Account. 52](#_Toc130292381)

[3.27 Manage Order. 54](#_Toc130292382)

[3.28 Update Order Status. 56](#_Toc130292383)

[3.29 View Order. 57](#_Toc130292384)

[3.30 View Order Change. 58](#_Toc130292385)

[IV. Non-functional requirements 59](#_Toc130292386)

[4.1 External Interface 59](#_Toc130292387)

[4.2 Quality Attributes 60](#_Toc130292388)

[V. Requirements Appendix 61](#_Toc130292389)

[1.Business Rules 61](#_Toc130292390)

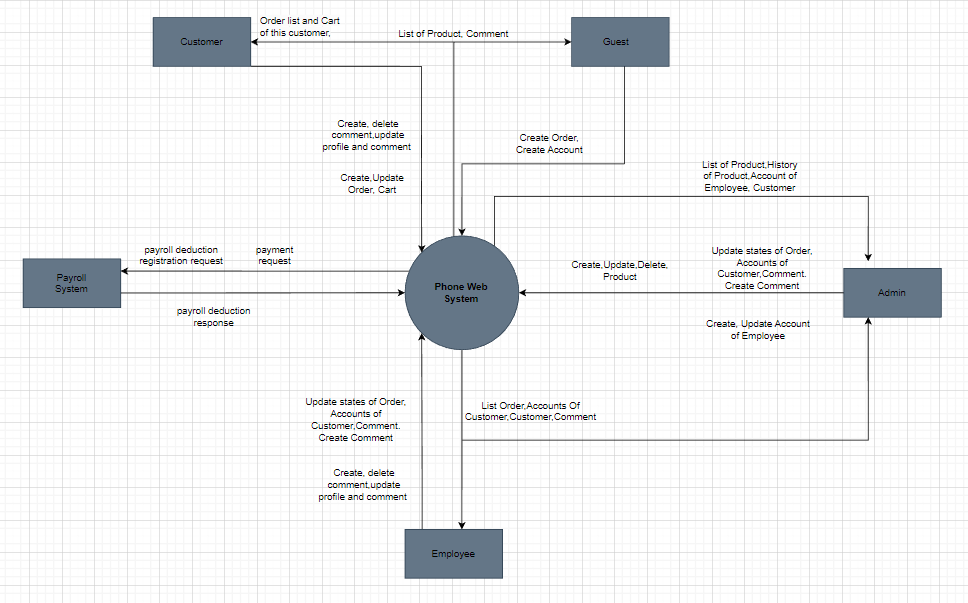
[2.Common Requirement 62](#_Toc130292391)

[3.Application Messages List 62](#_Toc130292392)

# I. Overview

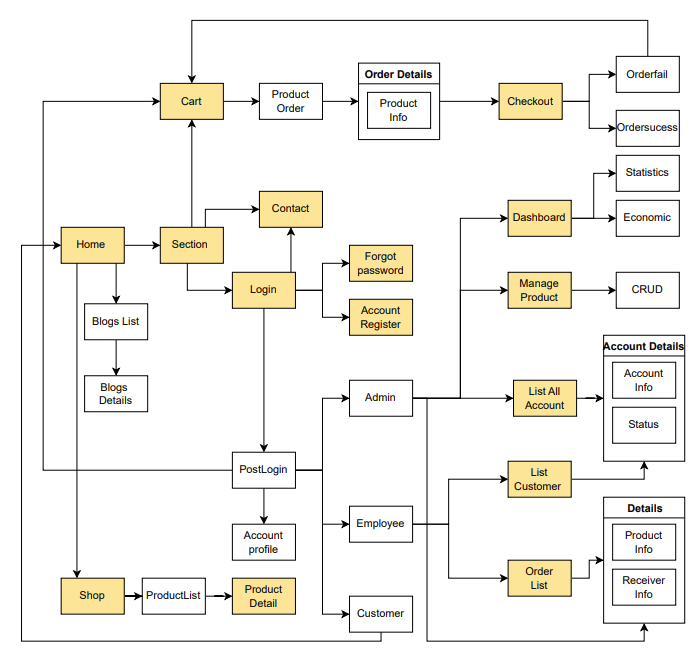
## 1.1. Introduction

Sell phone website can replace the current manual and help users buy easier instead of going showroom. The system is expected to be developed after several releases and updates. Besides, the website will link with phone brands to release the latest products.



## 1.2. System Functions

#### a. Screen Flow



#### b. Screen Details

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Feature** | **Screen** | **Description** |
| 1 | Show Product | Shop | Show all products. |
| 2 | Storage Product | Cart | Store the product which user add in order. |
| 3 | User authentication | Login | User can sign in by their account. |
| 4 | User authentication | Forgot password | User can change password when they don’t remember the password. |
| 5 | User authentication | Register | Users create an account. |
| 6 | Contact | Contact | User can send feedback to help improving quality of service. |
| 7 | Payment | Checkout | User can complete order in checkout screen by using credit card, E-wallet or Cash, v.v. |
| 8 | Show Infor | Product Detail | Show the information of the product. |
| 9 | Show Infor | Account Profile | Show the information of the account and they can UR the information. |
| 10 | Order Product | Order Detail | User can CRUD the product in order |
| 11 | Admin | List All Account | Admin can CRUD account status (exam: ban account). |
| 12 | Admin | Dashboard | Admin can see the view order statistics and economic on that time. |
| 13 | Employee | List Customers | Employee can view all Customer mail |
| 14 | Employee | Order List | Employee can CRUD order. |

#### c. User Authorization

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Screen** | **Guest** | **Customer** | **Employee** | **Admin** |
| Home | X | X | X | X |
| Sign In |  | X | X | X |
| Sign Up | X | X | X |  |
| Checkout | X | X | X |  |
| Shopping cart | X | X | X |  |
| Contact | X | X |  |  |
| Product Detail | X | X | X | X |
| Forgot password |  | X | X | X |
| Account Profile | X | X | X |  |
| Order Detail | X | X | X |  |
| List All Account |  |  |  | X |
| Dashboard |  |  |  | X |
| List Customers |  |  | X |  |
| Order List |  |  | X | X |

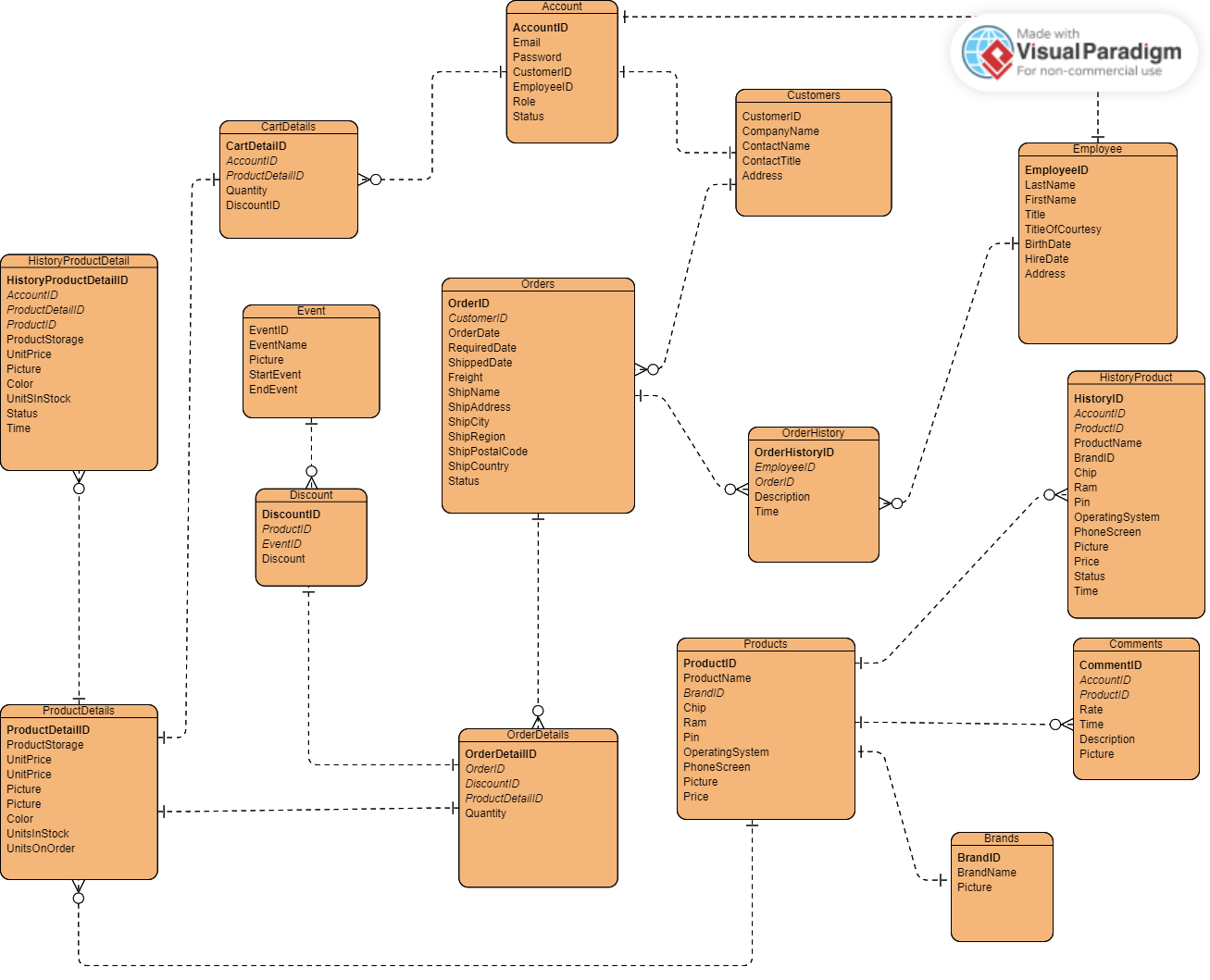
In which:

* Guest: User who does not have the account
* Customer: User have account
* Employee: Who can manage the products store
* Admin: The owner

#### d. Non-Screen Functions

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Feature** | **System Function** | **Description** |
| 1 | **External Interfaces** | **User interface** | -UI-1: The GUI is suitable for display on the PC and laptop.  -UI-2: The layout of the screen is streamlined, easy to  operate, and implement for the user.  -UI-3: For the creator screen, the system must show the confirmation modal (Y / N) for operations add, edit, delete, clone survey.  -UI-4: Use icons to provide visual insights to users.  -UI-5: Encourage vertical scrolling, minimize horizontal scrolling.  -UI-6: All error messages must provide troubleshooting instructions to the user.  -UI-7: Important commands and messages are displayed as buttons and labels with distinct background colors.  -UI-8: For student screen, the system must show confirmation modal (Y / N) for survey submission activity. |
| 2 | … |  |  |

## 1.3. Entity Relationship Diagram



**Entities Description**

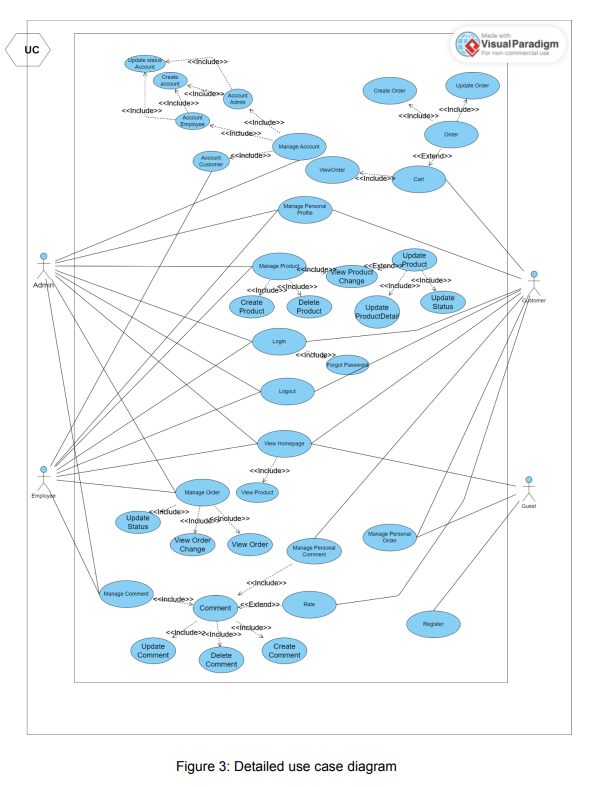
|  |  |  |
| --- | --- | --- |
| **#** | **Entity** | **Description** |
| 1 | Accounts | Accounts (Entity): The account of someone when accessing an online shopping website.  AccountID: Used to identify the account record.  Email: The account of the person who owns the account.  Password: The password of the person who owns the account.  CustomerID: ID of customer in Customers table.  EmployeeID: ID of employee in Employees table.  Role: Used to determine the user's function in the online shopping website (0 Admin, 1 Employee, 2 Customer).  Status: The status of the account (1: Active,2: Banned). |
| 2 | Customers | Customers (Entity): Customers who own an account can sign in to the store's website.  CustomerID: Used to identify the  customer record in the Customers table.  CompanyName: The company name of the customer. ContactName: The name of the customer.  ContactTitle: The position of the customer. Address: The address of the customer. |
| 3 | Employees | EmployeeID: Used to identify the employee record in the Employees table.  LastName: The last name of the employee.  FirstName: The first name of the employee.  Title: The job title or position of the employee.  ReportsTo: The ID of the employee's supervisor.  BirthDate: The birth date of the employee.  HireDate: The date when the employee was hired.  Address: The address of the employee. |
| 4 | Comments | Comments (Entity): Customers' comments on the products available in the store.  CommentID: Used to identify the comment record in the Comments table.  ProductID: ID of the product that the customer comments on in the Products table.  AccountID: ID of the account that comments in the Accounts table.  Status: The status of the comment (1: Active, 2: Delete).  Rate: The customer's rating for this product, each account can rate a product only once.  Time: The time the user posted this comment.  Description: The user's comment on the product.  Picture: The picture that the user posts. |
| 5 | Products | Products (Entity): Products sold on the online shopping website  ProductID: Used to identify the product record in the Products table  ProductName: The name of the product  BrandID: The ID of the product's brand  Chip: The name of the chip in the product  Pin: The type of battery in the product  OperatingSystem: The operating system of the product  PhoneScreen: The specifications of the product's screen  Picture: The picture describing the product  Price: The product's price (Take the lowest product price in the ProductDetails table). |
| 6 | ProductDetails | ProductDetails (Entity): Detailed information about the product  ProductDetailsID: Used to identify the product detail record in the ProductDetails table  ProductID: The ID of the product that has this ProductDetail  ProductStorage: The capacity of the product  UnitPrice: The price of the product  Picture: The illustration image of the product with the properties in this table  Color: The color of the product  UnitsInStock: The number of items in stock  UnitsOnOrder: The number of items on order. |
| 7 | HistoryProducts | HistoryProducts (Entity): The history of updating, deleting, and creating a product  HistoryID: Used to identify the HistoryProduct record in the HistoryProducts table  AccountID: The ID of the account that performed this history  ProductID: The ID of the product that was updated, created, or deleted  ProductName: The name of the product  BrandID: The ID of the product's brand  Chip: The name of the chip in the product  Pin: The type of battery in the product  OperatingSystem: The operating system of the product  PhoneScreen: The specifications of the product's screen  Picture: The picture describing the product  Price: The product's price (Take the lowest product price in the ProductDetails table). |
| 8 | HistoryProductDetails | HistoryProductDetails (Entity): History of updates, deletions, and creations of ProductDetail  HistoryProductDetailID: Used to identify the HistoryProductDetail record in the HistoryProductDetailIDs table  AccountID: ID of the account that performed this history  ProductID: ID of the product that has this ProductDetail  ProductStorage: Storage capacity of the product  UnitPrice: Price of the product  Picture: Illustration picture of the product with its attributes  Color: Color of the product  UnitsInStock: Quantity in stock  UnitsOnOrder: Quantity on order. |
| 9 | Events | Events (Entity): The events within a certain time period of the store  EventID: Used to identify the Event record in the Events table  EventName: The name of the event  Picture: The image of the event  StartEvent: The start time of the event  EndEvent: The end time of the event. |
| 10 | Discounts | Discounts (Entity): The discount of a product in the store  DiscountID: Used to identify the Discount record in the Discounts table  ProductID: The ID of the product that has this discount  EventID: The ID of the event of the discount  Discount: The percentage of discount. |
| 11 | Orders | Orders (Entity): The customer's order when purchasing a product  OrderID: Used to identify the Order record in the Orders table  CustomerID: The ID of the customer who placed this order  OrderDate: The time the order was approved  RequiredDate: The time the product is requested to be shipped to the customer  ShippedDate: The actual time the product was shipped to the customer  Freight: The weight of the order  ShipName: The name of the customer who receives the shipment  ShipAddress: The address of the shipment  ShipCity: The shipping company  ShipRegion: The province of the shipment  ShipPostalCode: The shipment code  ShipCountry: The country of the shipment  Status: The status of the order (0: Order Deleted, 1: Approved, 2: Pending Approval) |
| 12 | OrderDetails | OrderDetails (Entity): The details of the order  OrderDetailID: Used to identify the OrderDetail record in the OrderDetails table  OrderID: The ID of the order that owns this OrderDetail  DiscountID: The ID of the Discount of this OrderDetail  ProductDetailID: The ID of the product in this OrderDetail  Quantity: The quantity of the product. |
| 13 | OrderHistories | OrderHistories: The history of changes in the status of an Order  OrderHistoryID: Used to identify the OrderHistory record in the OrderHistories table  EmployeeID: The ID of the employee who changed the status  Description: Approve or disapprove the order  Time: The time when the Order status was changed  OrderID: The ID of the order that was changed. |
| 14 | CartDetails | CartDetails (Entity):  CartDetailID:  DiscountID:  ProductDetailID:  Quantity: |
| 15 | Brands | Brands (Entity): The brand of the products in the shop  BrandID: Used to identify the Brand record in the Brands table  BrandName: The name of the brand  Picture: The logo of the brand. |
|  |  |  |

# II. User Requirements

## 2.1. Actors

|  |  |  |
| --- | --- | --- |
| # | Actor | Description |
| 1 | Admin | The person who manages all accounts, product, order, comment in the system. |
| 2 | Employee | The person who manages all accounts of customer, order, comment in the system. |
| 3 | Customer | The registered user who create and update order , comment, profile of this user. |
| 4 | Guest | People who do not have an account can create order but can’t comment, rate, and save cart. |

## 2.2. Uses Case Diagram



## 2.3. Uses Case Description

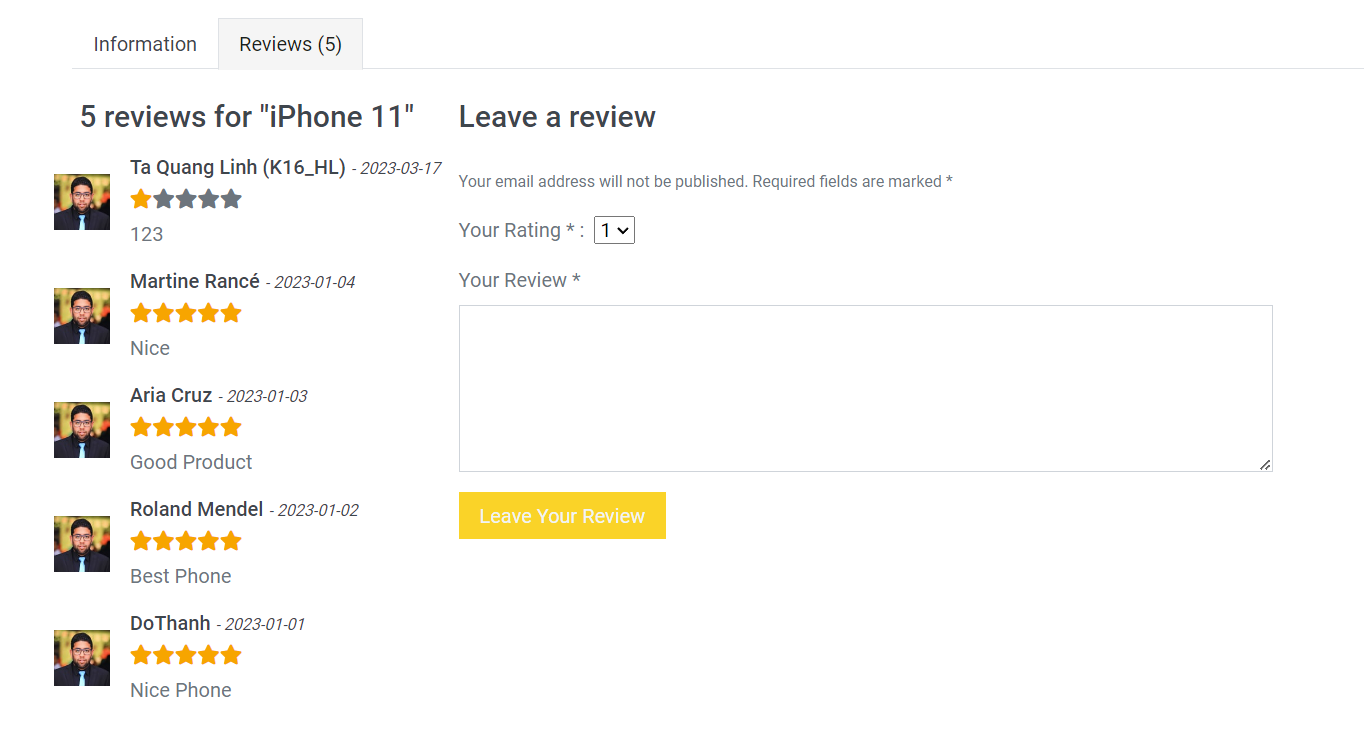
|  |  |  |  |
| --- | --- | --- | --- |
| No. | Use Case | Actors | Use Case Description |
| 1 | Create comment | Customer,  Employee, Admin | This function allows commenting when an actor has logged into the system. |
| 2 | View comment | Customer, Guest, Employee, Admin | This function allows actors to view comment of everybody |
| 3 | Update comment | Customer, Employee, Admin | This function allows users to edit their comment when they have logged into the system and comment before |
| 4 | Delete comment | Customer, Employee, Admin | This function allows actors to delete their comment when they have logged into the system and comment before |
| 5 | View profile account | Customer, Employee | This function allows actors to view their profile account |
| 6 | Update profile account | Customer, Employee | This function allows actor to edit their profile when they have logged into the system |
| 7 | Change password account | Customer, Employee | This function allows actor to change their profile when they have logged into the system |
| 8 | Login | Customer, Employee, Admin | This function allows actors to login the system to access their function’s role |
| 9 | Logout | Customer, Employee, Admin | This function allows actors to logout the system |
| 10 | View home page | Guest, Customer | This function allows actors to view home page |
| 11 | View shopping cart | Guest, Customer | This function allows actors to view their order cart |
| 12 | Create order | Customer | This function allows actors to create an order when actor has logged in the system and create shopping cart before |
| 13 | Update order | Customer | This function allows actors to update their orders when logged in the system and create shopping cart before |
| 14 | Register | Guest | This function allows guest to register a new account in the system |
| 15 | Manage Comment | Admin | This function allows admin to manage comment of customers. |
| 16 | Manage Product | Admin | This function allows admin to view all product’s information in the system by clicking the Show Product button from the Manage. |
| 17 | Create Product | Admin | This function admin to create a new product into the system |
| 18 | View Product Change | Admin ,Employee | This function allows admin to view history of changed product information |
| 19 | Update Product Detail | Employee | This function allows actors to change information of product in the system |
| 20 | Update Status | Employee | This function allows actors to suspend business items |
| 21 | Delete Product | Admin | This function allows admin to delete product(s) out of the system |
| 22 | Manage Account | Admin | This function allows admin to choose one of customer or employee role to manage. |
| 23 | Customer Account | Admin | This function allows admin to view all Customer’s account in the system by clicking the account management button from the Manage. |
| 24 | Employee Account | Admin | This function allows admin to view all Employee’s account in the system by clicking the account management button from the Manage. |
| 25 | Create Account | Admin | This function allows admin to add a new account into the system by clicking the “Create” button and filling out the form with account information. |
| 26 | Update Account | Admin | This function allows admin to edit information of an account in the system by clicking on the “Update” column on the account management page. |
| 27 | Manage Order | Admin, Employee | This function allows actors to view all Orders in the system by clicking the Show Order button from the sidebar. |
| 28 | Update Status | Admin, Employee | This function allows actors to update order’s status |
| 29 | View Order | Admin, Employee | This function allows actors to view detail of Customer’s order |
| 30 | View Order Change | Admin, Employee | This function allows actors to view history of order’s status |

# III. Functional Requirements

## 3.1 Create comment.

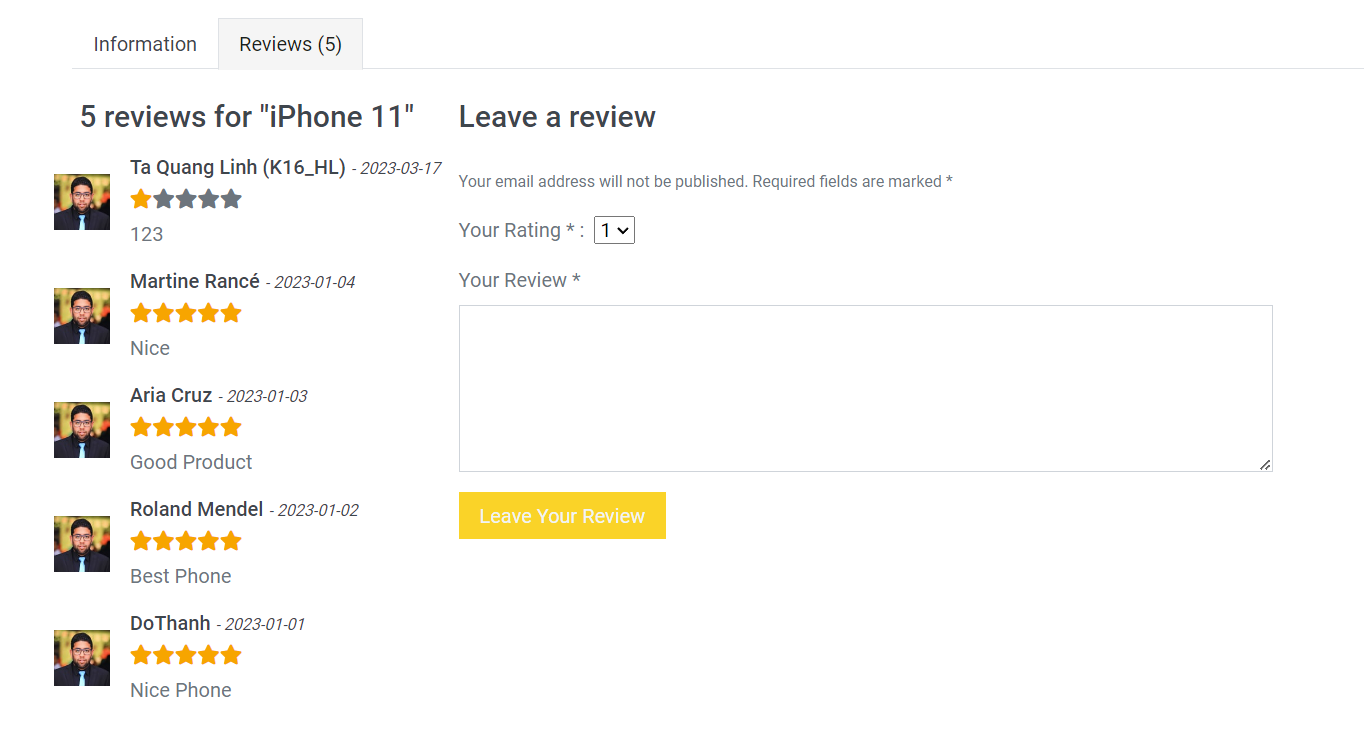
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **01 - Create comment** | Actors: | | Customer, Employee, Admin |
| Description: | A brief explanation of what creating a comment involves, its purpose and the expected outcome. | | | |
| Trigger: | The event or action that initiates the process of creating a comment. For example, clicking on the "add comment" button. | | | |
| Preconditions: | PRE-1.The user must be logged in to the system.  PRE-2.The user must have permission to create comments.  PRE-3The user must be viewing a page or post that allows comments.  PRE-4Another precondition might be that the user has already read the content they wish to comment on. | | | |
| Post-conditions: | POST-1.The comment should be visible to the user and other users who can view the content.  POST-2.The user should receive a confirmation message indicating that the comment has been successfully created.  POST-3.The user's comment count should be incremented.  POST-4.Another post-condition might be that the user's comment appears in their profile page if the system has one. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 1.The user enters text for the comment.  3.The user selects the "add comment" button.  4.The user submits the comment.  8.The user views the newly created comment. | | 2.The system displays a comment form.  5.The system saves the comment to the database.  6.The system displays a confirmation message to the user indicating that the comment has been successfully created.  7.The system displays the newly created comment on the page or post. | |
| Alternative Flows: | The sequence of steps that are taken in alternative scenarios when creating a comment. | | | |
| Exception Flows | The sequence of steps that are taken when an unexpected or error scenario occurs when creating a comment. | | | |
| Business Rules: |  | | | |

## 3.2 View comment.



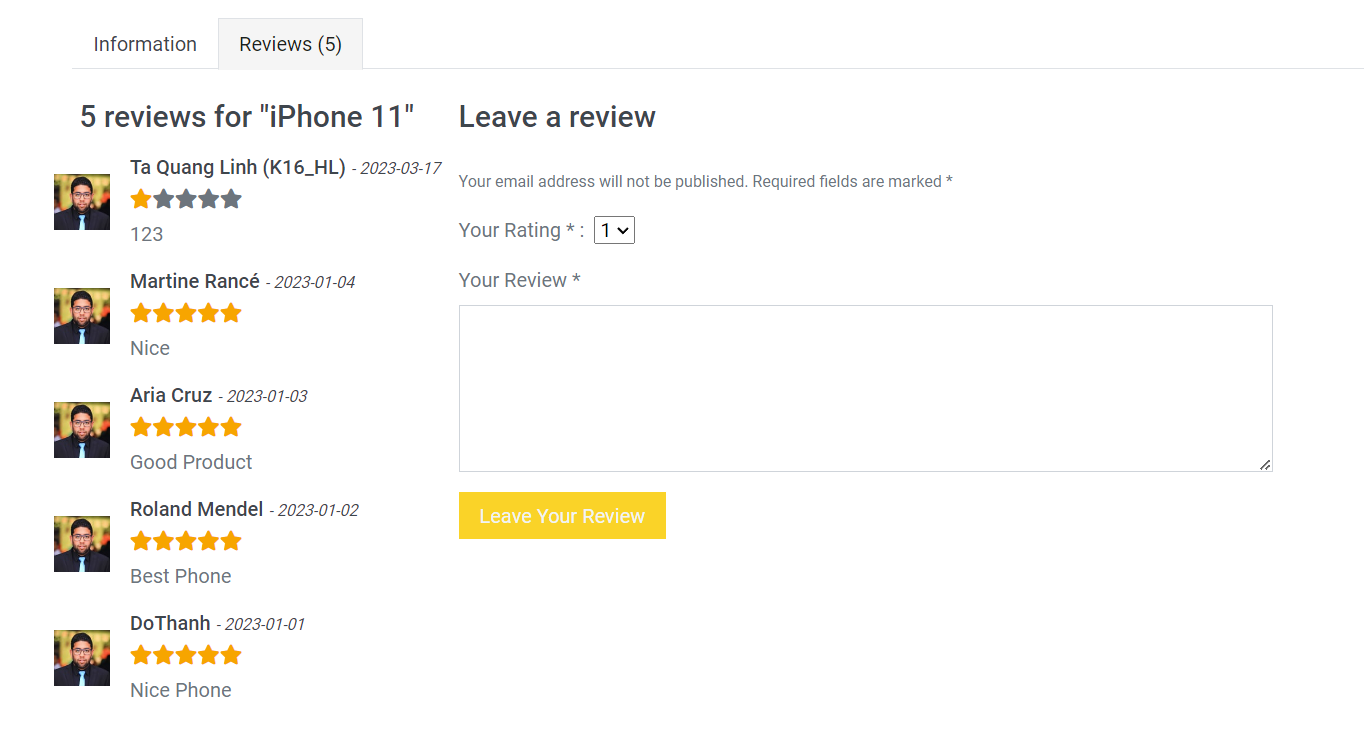
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **02 - View comment** | Actors: | | Customer, Guest, Employee, Admin |
| Description: | Creating a comment refers to the process of adding a comment to an existing post or article on a website or social media platform. | | | |
| Trigger: | The trigger for creating a comment could be when a user reads an article or post and wants to share their thoughts or feedback with others. | | | |
| Preconditions: | PRE-1: The user must be logged into their account on the website or social media platform.  PRE-2: The user must have navigated to the page containing the post or article where they want to add a comment. | | | |
| Post-conditions: | POST-1: The comment is successfully added to the post or article.  POST-2: The user is redirected to the page containing the post or article with the new comment displayed. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 1. The user clicks on the "Add Comment" or "Write a Comment" button.  3. The user types in their comment and clicks on the "Post" or "Submit" button. | | 2. The website or social media platform displays a comment box or field where the user can type in their comment.  5. The comment is added to the post or article and the user is redirected to the page containing the post or article with the new comment displayed. | |
| Alternative Flows: | If the user decides not to add a comment after clicking on the "Add Comment" or "Write a Comment" button, they can click on the "Cancel" button to return to the post or article without adding a comment. | | | |
| Exception Flows | If the user is not logged into their account, they will be prompted to log in or create an account before they can add a comment.  If there is an error in the process of adding the comment, the website or social media platform will display an error message and prompt the user to try again. | | | |
| Business Rules: |  | | | |

## 3.3 Update comment.



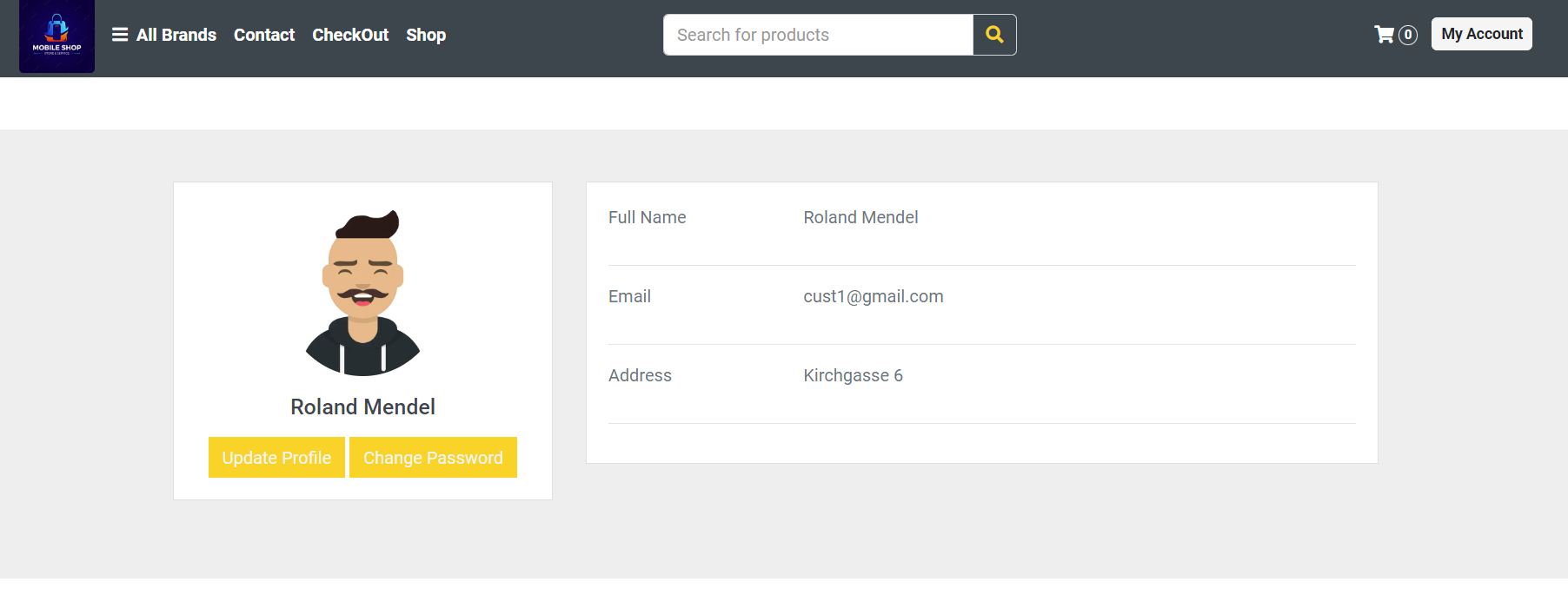
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **03 -** Update comment | Actors: | | Customer, Employee, Admin |
| Description: | Updating a comment refers to the process of editing or modifying an existing comment on a website or social media platform. | | | |
| Trigger: | The trigger for updating a comment could be when a user wants to correct a mistake, add additional information, or clarify their previous comment. | | | |
| Preconditions: | PRE-1: The user must be logged into their account on the website or social media platform.  PRE-2: The user must have previously created a comment on the post or article that they want to update. | | | |
| Post-conditions: | POST-1: The updated comment is successfully saved and displayed on the post or article.  POST-2: The user is redirected to the page containing the post or article with the updated comment displayed. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 1. The user clicks on the "Edit" or "Update" button next to their comment.  3. The user modifies the comment text and clicks on the "Save" or "Update" button. | | 2. The comment text is displayed in an editable field or box.  4. The updated comment is saved and displayed on the post or article, and the user is redirected to the page containing the post or article with the updated comment displayed. | |
| Alternative Flows: | If the user decides not to update the comment after clicking on the "Edit" or "Update" button, they can click on the "Cancel" button to return to the post or article without modifying their comment. | | | |
| Exception Flows | If the user is not logged into their account, they will be prompted to log in or create an account before they can update their comment.  If there is an error in the process of updating the comment, the website or social media platform will display an error message and prompt the user to try again. | | | |
| Business Rules: |  | | | |

## 3.4 Delete comment.



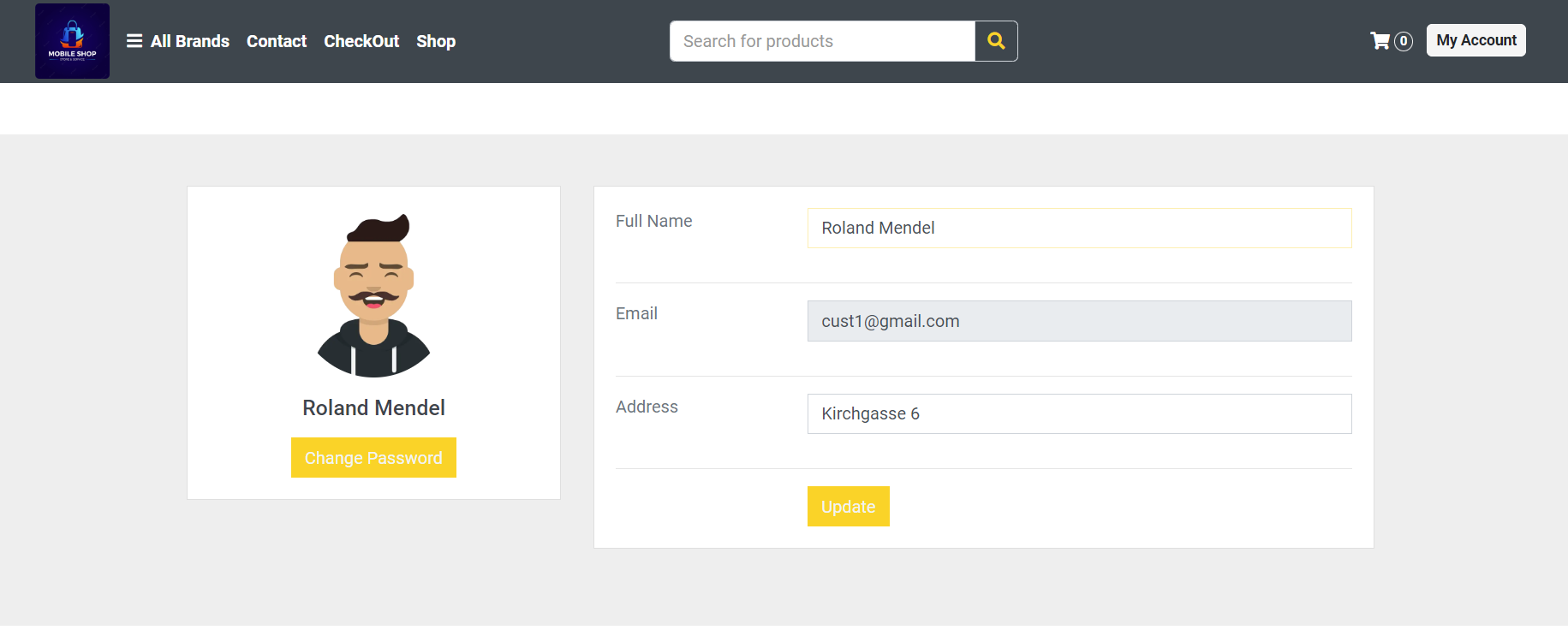
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **04 -** Delete comment | Actors: | | Customer, Employee, Admin |
| Description: | Deleting a comment refers to the process of removing a previously created comment on a website or social media platform. | | | |
| Trigger: | The trigger for deleting a comment could be when a user wants to remove their comment, they no longer want to have their comment associated with the post, or they have posted something they regret. | | | |
| Preconditions: | PRE-1: The user must be logged into their account on the website or social media platform.  PRE-2: The user must have previously created a comment on the post or article that they want to delete.  PRE-3: The user must have the necessary permissions to delete the comment, as some websites or social media platforms may restrict the deletion of certain types of comments. | | | |
| Post-conditions: | POST-1: The comment is successfully deleted from the post or article.  POST-2: The user is redirected to the page containing the post or article without the deleted comment displayed. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 1. The user clicks on the "Delete" or "Remove" button next to their comment.  3. The user confirms the deletion by clicking on the "Yes" or "Delete" button in the confirmation message. | | 2. The website or social media platform displays a confirmation message asking the user to confirm the deletion of their comment.  4. The comment is deleted from the post or article, and the user is redirected to the page containing the post or article without the deleted comment displayed. | |
| Alternative Flows: | If the user decides not to delete the comment after clicking on the "Delete" or "Remove" button, they can click on the "Cancel" button in the confirmation message to return to the post or article without deleting their comment. | | | |
| Exception Flows | If the user is not logged into their account, they will be prompted to log in or create an account before they can delete their comment. | | | |
| Business Rules: |  | | | |

## 3.5 View profile account.



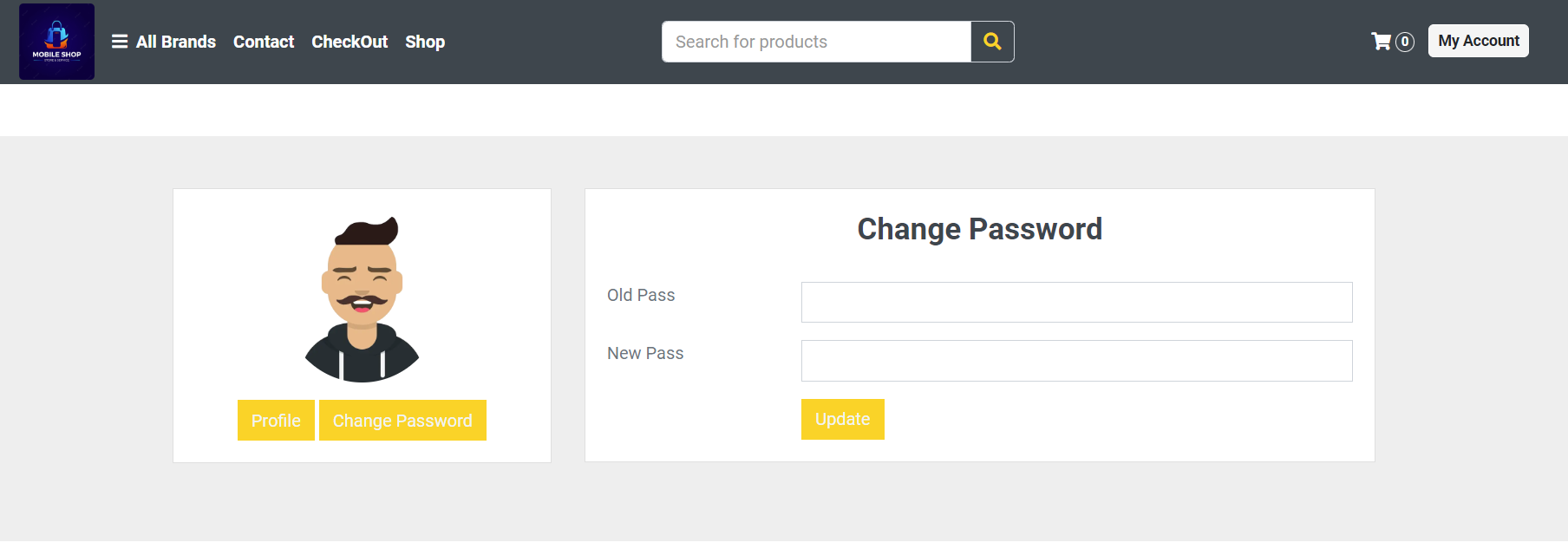
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **05 -** View profile account | Actors: | | Customer, Employee |
| Description: | Viewing a profile account refers to the process of accessing a user's profile page on a website or social media platform, where their personal information and posts are displayed. | | | |
| Trigger: | The trigger for viewing a profile account could be when a user wants to know more about another user's profile or to view their own profile. | | | |
| Preconditions: | PRE-1: The user must be logged into their account on the website or social media platform.  PRE-2: The user must have a valid username or user ID to access the profile page. | | | |
| Post-conditions: | POST-1: The user can view the profile account information, including the user's name, profile picture, bio, and posts.  POST-2: The user can interact with the profile account by sending messages or following the user. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 1. The user navigates to the profile account page by entering the username or user ID into the search bar or clicking on a link to the profile.  3. The user can interact with the profile account by sending messages, following the user, or performing other actions available on the page. | | 2. The profile account page is displayed, including the user's name, profile picture, bio, and posts. | |
| Alternative Flows: | If the user enters an invalid username or user ID, the system will display an error message and prompt the user to try again.  If the profile account is set to private, the user may be required to send a request to follow the user or be approved by the user before they can view the profile account. | | | |
| Exception Flows | If the user is not logged into their account, they will be prompted to log in or create an account before they can view the profile account.  If there is an error in the process of accessing the profile account, the website or social media platform will display an error message and prompt the user to try again. | | | |
| Business Rules: |  | | | |

## 3.6 Update profile account.



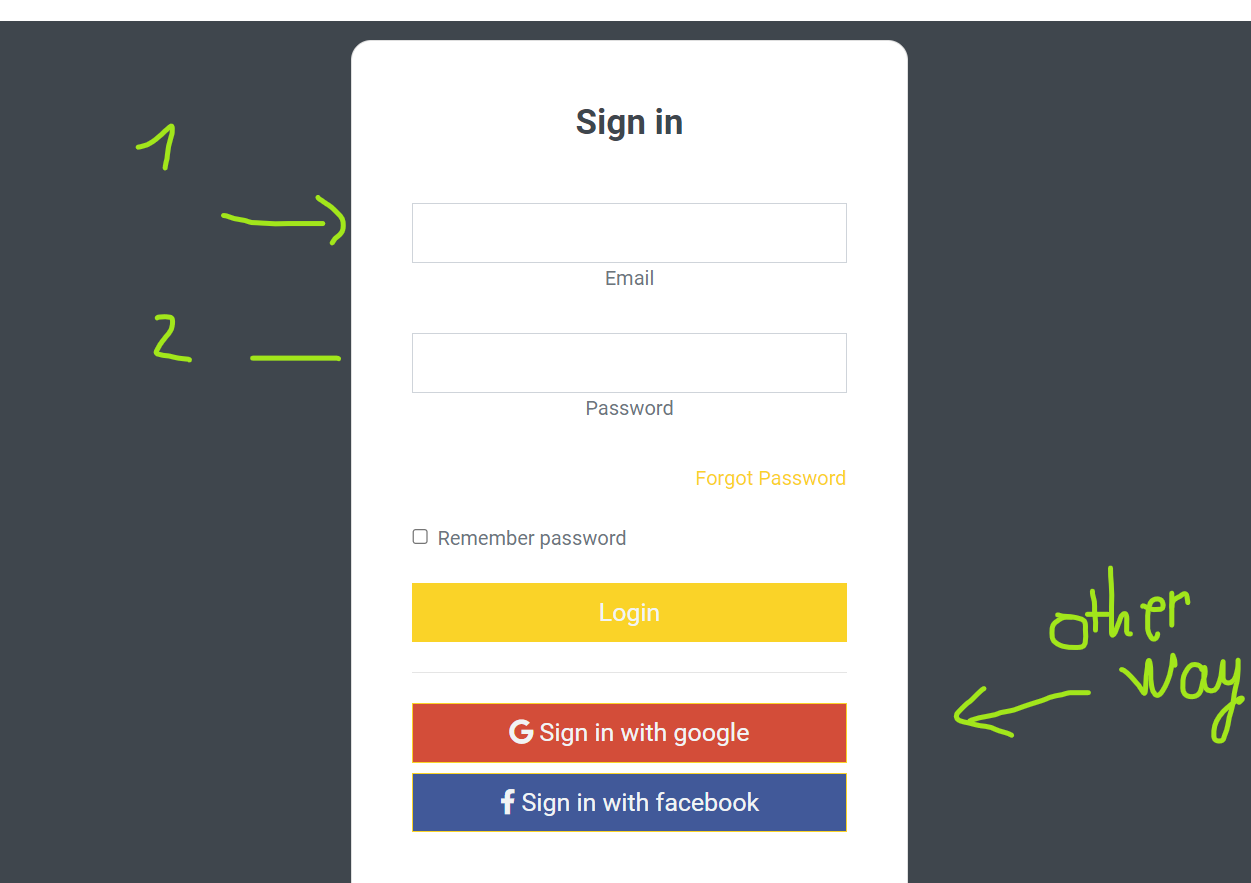
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **06 -** Update profile account | Actors: | | Customer, Employee |
| Description: | Updating a profile account refers to the process of changing or modifying the personal information, settings, or posts on a user's profile page on a website or social media platform. | | | |
| Trigger: | The trigger for updating a profile account could be when a user wants to change their profile picture, update their bio or personal information, or modify their privacy settings. | | | |
| Preconditions: | PRE-1: The user must be logged into their account on the website or social media platform.  PRE-2: The user must have a valid profile account with information and settings that can be modified. | | | |
| Post-conditions: | POST-1: The user's profile account information, settings, or posts are updated and saved.  POST-2: The updated information is displayed on the user's profile page. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 1. The user navigates to their profile account page.  3. The user selects the option to update their profile account.  5. The user modifies their personal information, settings, or posts.  7. The user navigates back to their profile account page.  System response: The website or social media platform displays the updated profile information. | | 2. The user's profile page is displayed with their current information and settings.  4. The website or social media platform displays the profile editing interface.  6. The website or social media platform updates and saves the changes to the user's profile account. | |
| Alternative Flows: | If the user encounters an error while updating their profile account, the system will display an error message and prompt the user to try again.  If the user tries to update information that violates the website or social media platform's terms of service, the system will display an error message and prevent the user from updating their profile account. | | | |
| Exception Flows | If the user is not logged into their account, they will be prompted to log in or create an account before they can update their profile account.  If the user does not have a valid profile account with information and settings that can be modified, the website or social media platform will display an error message and prevent the user from updating their profile account. | | | |
| Business Rules: |  | | | |

## 3.7 Change password account.



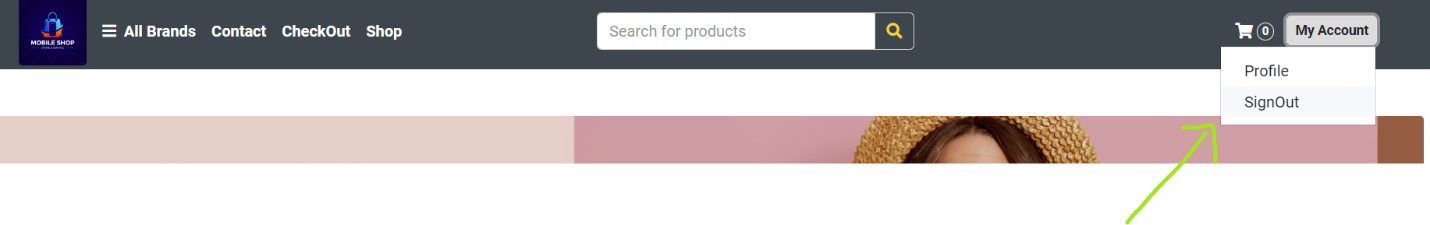
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **07 -** Change password account | Actors: | | Customer, Employee |
| Description: | Changing a password for an account refers to the process of updating the password associated with the user's account on a website or application. | | | |
| Trigger: | The trigger for changing a password could be when a user wants to update their password for security reasons or if they have forgotten their current password. | | | |
| Preconditions: | PRE-1: The user must be logged into their account on the website or application.  PRE-2: The user must know their current password.  PRE-3: The new password must meet the website or application's password requirements. | | | |
| Post-conditions: | POST-1: The user's account password is updated and saved.  POST-2: The user is logged out of their account and prompted to log back in with their new password. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 1. The user navigates to their account settings page.  3. The user selects the option to change their password.  5. The user enters their current password and new password.  7. The user is logged out of their account. | | 2. The website or application displays the account settings interface.  4. The website or application prompts the user to enter their current password and new password.  6. The website or application verifies that the new password meets their password requirements and updates and saves the user's account password.  8. The website or application displays a message prompting the user to log back in with their new password. | |
| Alternative Flows: | If the user enters an incorrect current password, the system will display an error message and prompt the user to try again.  If the new password does not meet the website or application's password requirements, the system will display an error message and prompt the user to choose a different password. | | | |
| Exception Flows | If the user is not logged into their account, they will be prompted to log in or create an account before they can change their password.  If the user does not know their current password, they will be prompted to reset their password through email or other means. | | | |
| Business Rules: |  | | | |

## 3.8 Login.



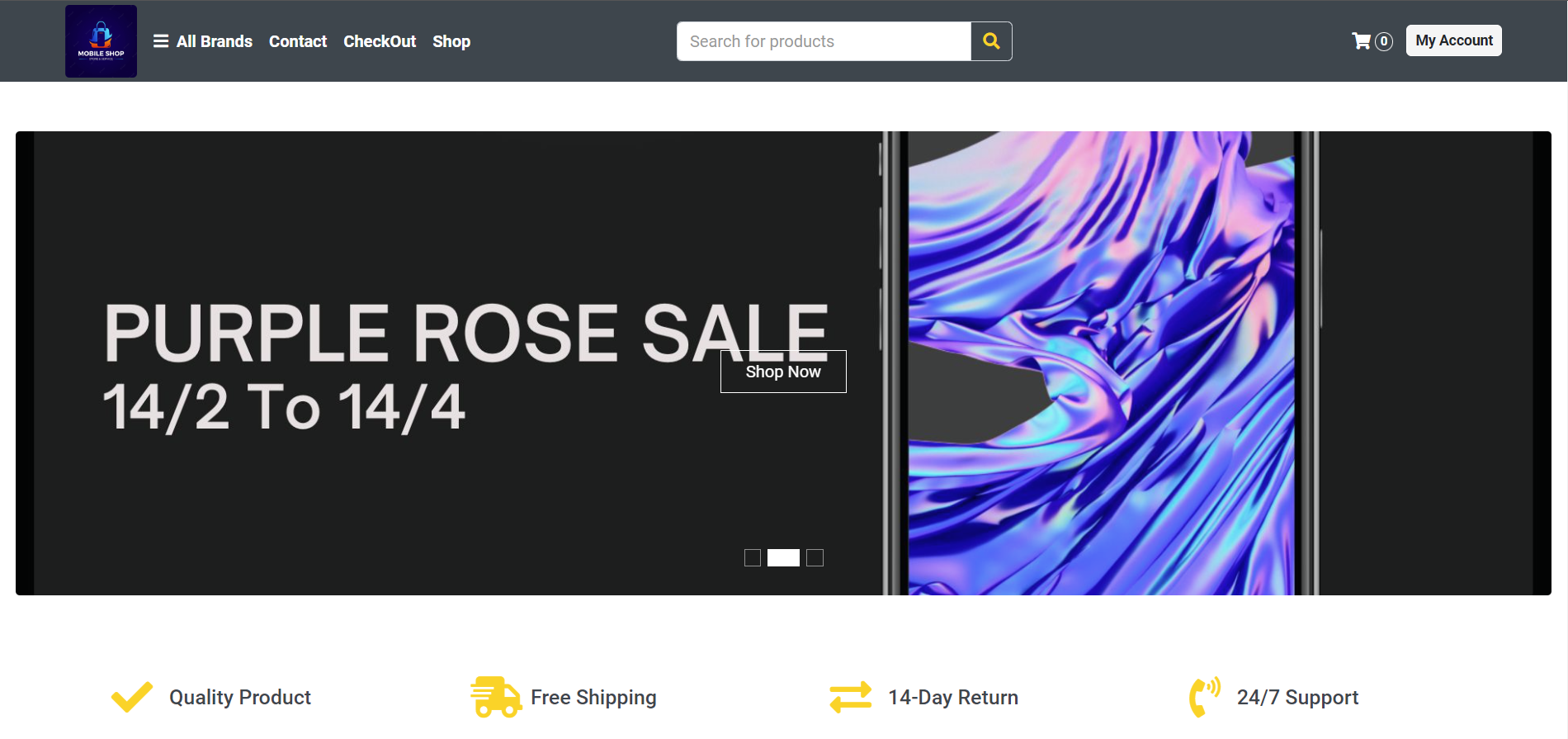
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **08 -** Login | Actors: | | Customer, Employee, Admin |
| Description: | This use case describes the steps involved when a user logs into the system. | | | |
| Trigger: | The user wishes to access the system and needs to authenticate themselves. | | | |
| Preconditions: | PRE-1: The user must have an account on the system.  PRE-2: The user must have a valid username and password. | | | |
| Post-conditions: | POST-1: The user is successfully logged into the system and can access their account. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 1. The user navigates to the login page and enters their username and password.  4. The user is now logged in and can access their account. | | 2. The system validates the user's credentials.  3. The system creates a session for the user and redirects them to the main page of the system. | |
| Alternative Flows: | If the user enters invalid credentials, the system displays an error message and prompts the user to enter their credentials.  If the user forgets their password, they can click on the "Forgot Password" link and follow the steps to reset their password. | | | |
| Exception Flows | If the system encounters an error during the login process, it displays an error message and prompts the user to try again later. | | | |
| Business Rules: |  | | | |

## 3.9 Logout.



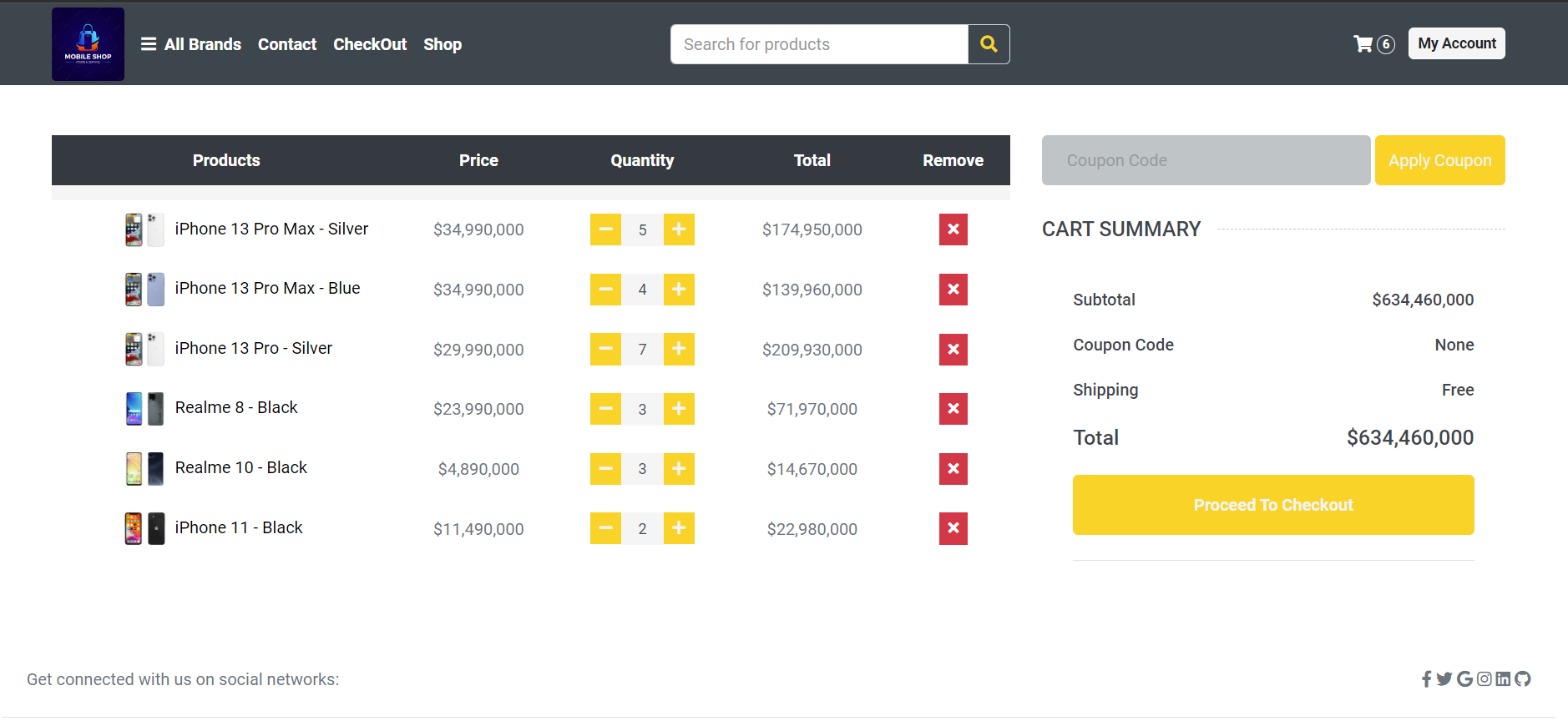
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **09 -** Logout | Actors: | | Customer, Employee, Admin |
| Description: | This use case describes the steps involved when a user logs out of the system. | | | |
| Trigger: | The user wants to end their current session and log out of the system. | | | |
| Preconditions: | PRE-1: The user must be logged into the system. | | | |
| Post-conditions: | POST-1: The user is successfully logged out of the system and returned to the login page. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 1. The user clicks on the "Logout" button or link.  4. The user know they were logged out. | | 2. The system logs the user out and returns them to the login page.  3. The system displays a message confirming the user has been logged out. | |
| Alternative Flows: | N/A | | | |
| Exception Flows | N/A | | | |
| Business Rules: |  | | | |

## 3.10 View home page.



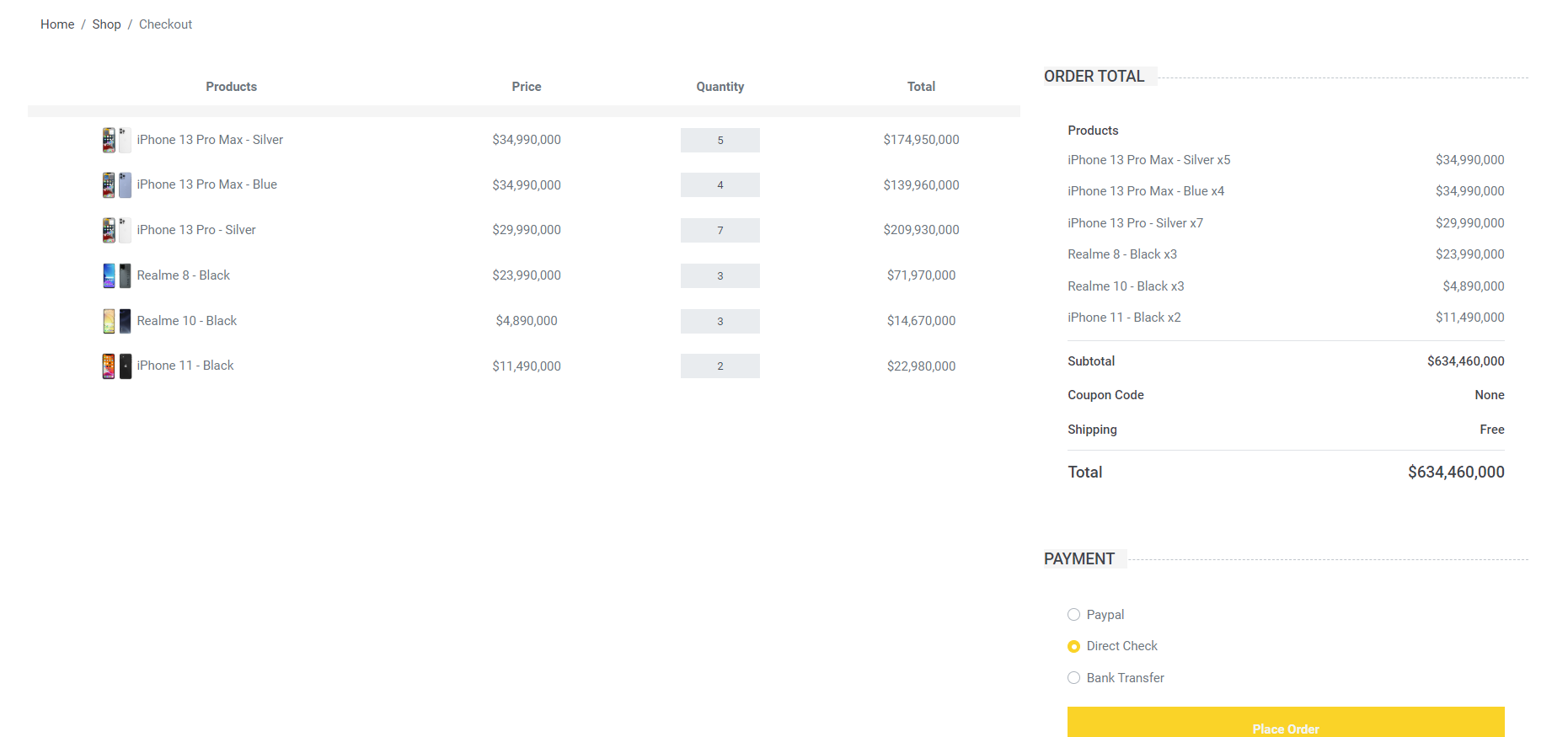
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **10 -** View home page | Actors: | | Guest, Customer |
| Description: | This use case describes the steps involved when a user views the home page of the system. | | | |
| Trigger: | The user wants to view the home page of the system. | | | |
| Preconditions: | PRE-1: The user must be logged into the system.  PRE-2: The user must have appropriate permissions to view the home page. | | | |
| Post-conditions: | POST-1: The user is successfully able to view the home page of the system. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 1. The user clicks on the "Home" button or link. | | 2. The system displays the home page of the system.  3. The system displays any relevant information or notifications on the home page. | |
| Alternative Flows: | N/A | | | |
| Exception Flows | N/A | | | |
| Business Rules: |  | | | |

## 3.11 View shopping cart.

****

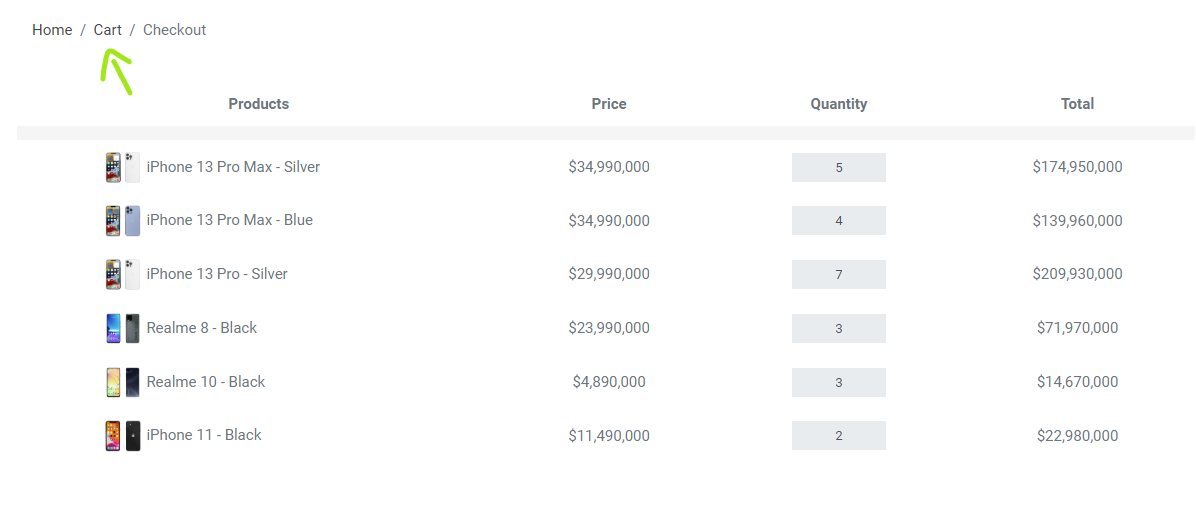
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **11 -** View shopping cart | Actors: | | Guest, Customer |
| Description: | The "View Shopping Cart" feature allows the user to view their current shopping cart, which includes all the items they have added to their cart during their current shopping session. | | | |
| Trigger: | The user clicks on the "View Cart" button/link on the website or app. | | | |
| Preconditions: | PRE-1: The user must have already added at least one item to their cart.  PRE-2: The user must be logged in to their account. | | | |
| Post-conditions: | POST-1: The user is presented with the contents of their shopping cart.  POST-2: The user can choose to proceed to checkout or continue shopping. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 1. The user clicks on the "View Cart" button/link.  2. The user can choose to adjust the quantity of any item, remove an item, or continue shopping.  6. The user can choose to proceed to checkout or continue shopping. | | 2. The system verifies that the user is logged in and has at least one item in their cart.  3. The system displays the contents of the user's shopping cart, including product names, quantities, prices, and subtotals.  4. The system displays the total cost of the items in the cart.  5. If the user chooses to adjust the quantity or remove an item, the system updates the cart and recalculates the total cost. | |
| Alternative Flows: | ALT-1: If the user is not logged in, the system prompts them to log in or create an account.  ALT-2: If the user has not added any items to their cart, the system displays a message indicating that the cart is empty.  ALT-3: If the user chooses to remove all items from the cart, the system displays a message confirming that the cart is empty and prompts the user to continue shopping.  ALT-4: If the user chooses to continue shopping, the system redirects them to the home page or a product category page. | | | |
| Exception Flows | EXC-1: If the system encounters an error while retrieving the contents of the user's shopping cart, it displays an error message and prompts the user to try again later. | | | |
| Business Rules: |  | | | |

## 3.12 Create order.

****

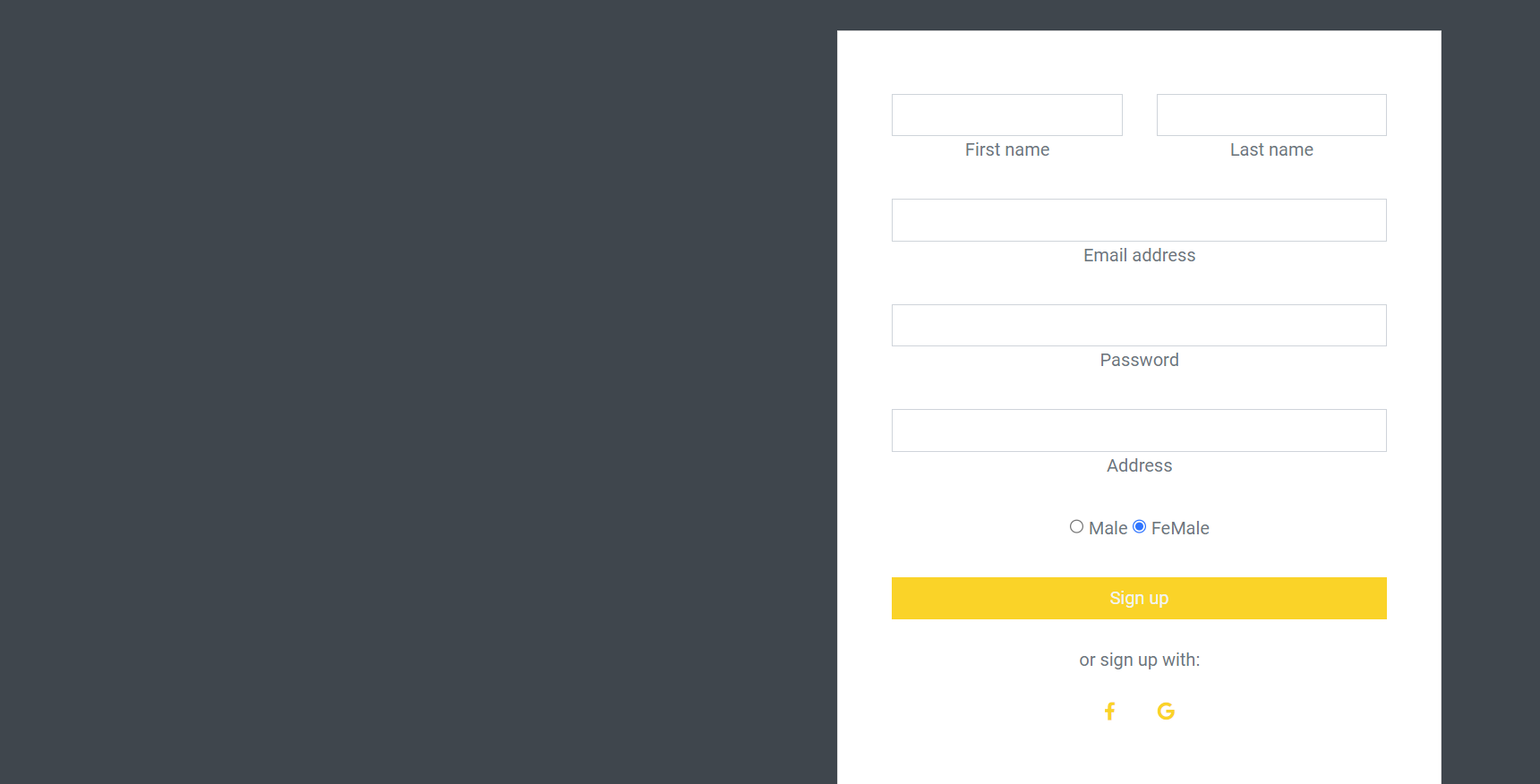
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **12 -** Create order | Actors: | | Customer |
| Description: | The use case describes the process of creating a new order for a customer on an e-commerce platform. | | | |
| Trigger: | The trigger for this use case is a customer's desire to purchase a product or products from the e-commerce platform. | | | |
| Preconditions: | PRE-1: The customer must be registered and logged into their account on the e-commerce platform.  PRE-2: The customer must have selected the product or products they wish to purchase and added them to their shopping cart.  PRE-3: The customer must have entered their billing and shipping information. | | | |
| Post-conditions: | POST-1: The customer's order is stored in the system's database.  POST-2: The customer's shopping cart is cleared of the purchased items.  POST-3: The customer is sent an order confirmation email.  POST-4: The customer is redirected to a confirmation page. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 1. The customer selects the "checkout" option from their shopping cart page.  5. The customer confirms the order and provides payment information.  . | | 2. The system displays a summary of the order, including the selected products, quantities, prices, and total cost.  3. The system verifies the payment information and displays a confirmation page.  4. The system sends an order confirmation email to the customer and updates the order status in the system. | |
| Alternative Flows: | ALT-1: If the customer needs to update their billing or shipping information, they can select the "edit" option to make changes before confirming the order.  ALT-2: If the customer does not have a registered account, they will be prompted to create one before proceeding with the order.  ALT-3: If there is an issue with the payment information, the system will display an error message and prompt the customer to update their payment details. | | | |
| Exception Flows | EXC-1: If the selected product is out of stock, the system will display an error message and prompt the customer to remove the product from their cart or choose a different product.  EXC-2: If there is a system error during the checkout process, the system will display an error message and prompt the customer to try again later. | | | |
| Business Rules: |  | | | |

## 3.13 Update order.

****

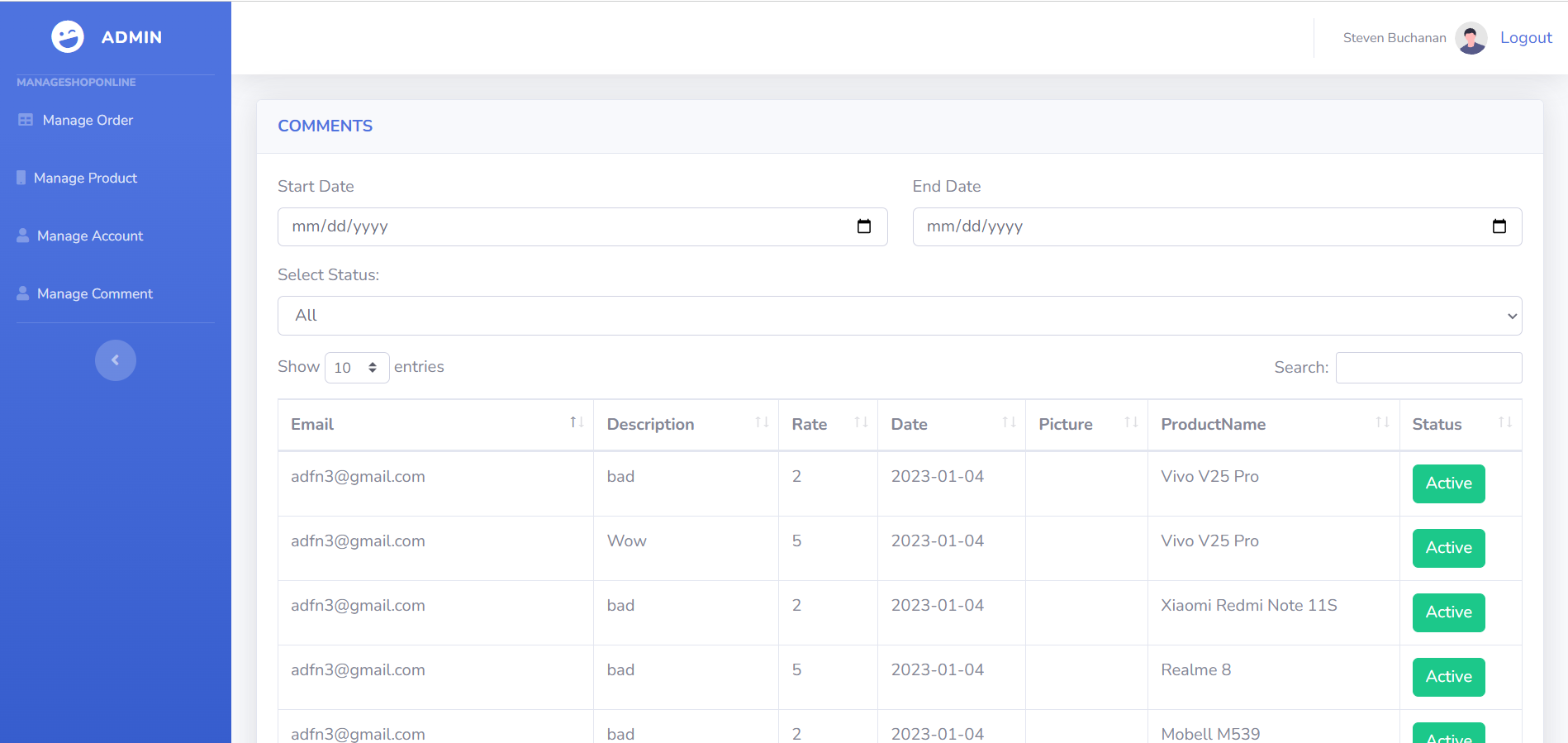
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **13 -** Update order | Actors: | | Customer |
| Description: | This use case describes the process of updating an existing order in the system. | | | |
| Trigger: | The user wishes to modify the details of an existing order. | | | |
| Preconditions: | PRE-1. The user must be logged into the system.  PRE-2. The user must have previously created at least one order. | | | |
| Post-conditions: | POST-1. The order details have been updated in the system.  POST-2. The user is presented with a confirmation message. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 1. The user navigates to the "My Orders" page.  3. The user selects the order they wish to update.  .  5. The user modifies the desired fields.  6. The user confirms the changes and submits the updated order. | | 2. The system displays a list of the user's orders.  3. The system verifies the payment information and displays a confirmation page.  4. The system displays the details of the selected order.  7. The system updates the order details and displays a confirmation message. | |
| Alternative Flows: | N/A | | | |
| Exception Flows | EXC-1. If the user does not have any orders in the system, the system displays an error message.  EXC-2. If the system encounters an error while updating the order details, the system displays an error message. The user may be prompted to try again or contact customer support for assistance. | | | |
| Business Rules: |  | | | |

## 3.14 Register.

****

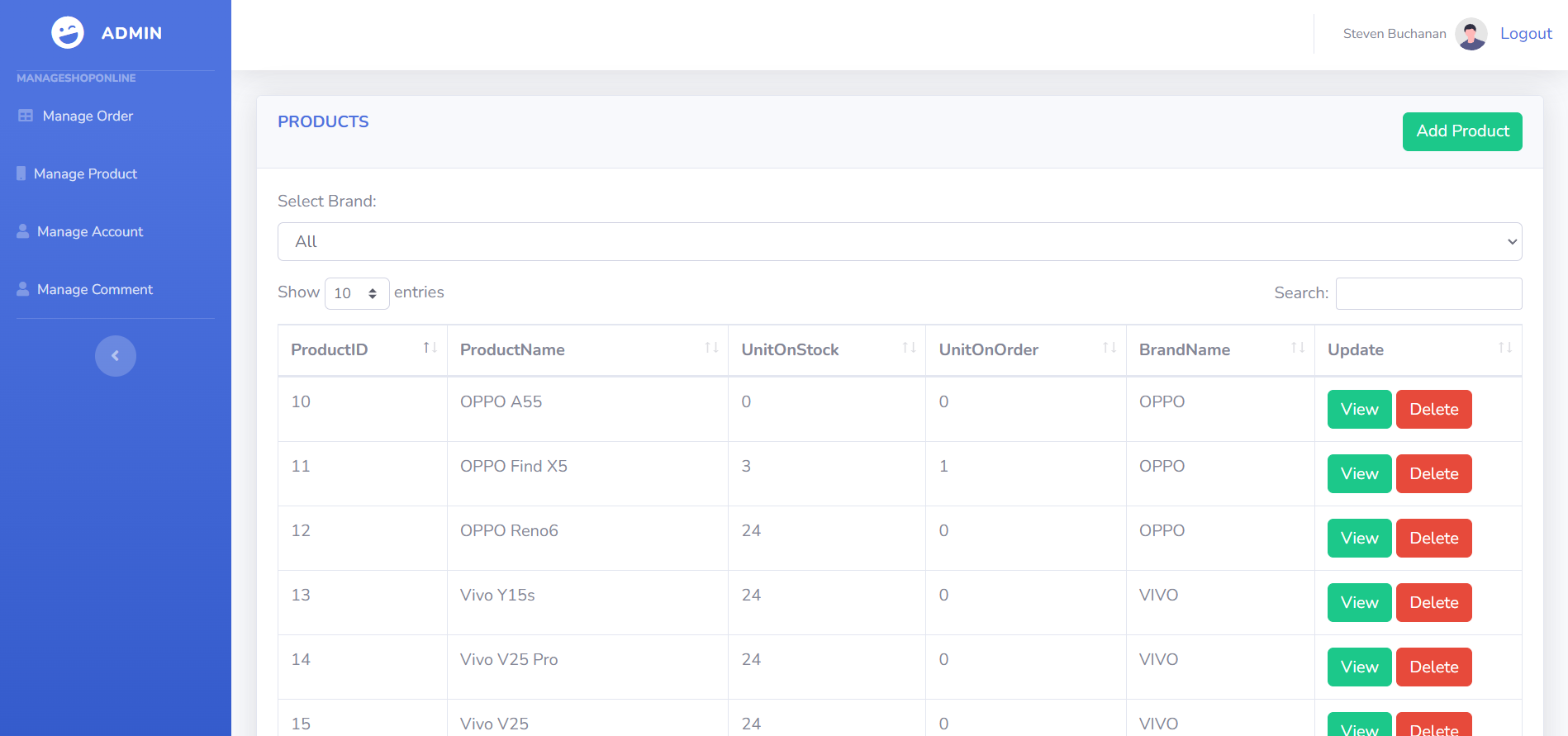
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **14 -** Register | Actors: | | Guest |
| Description: | This use case describes the process of a new user registering for an account on a website. | | | |
| Trigger: | The user clicks the "Register" button on the website. | | | |
| Preconditions: | PRE-1: The user has access to the internet and a web browser.  PRE-2: The website's registration feature is functional and accessible to the user.  PRE-3: The user does not already have an account on the website. | | | |
| Post-conditions: | POST-1: The user is successfully registered for an account on the website.  POST-2: The user is redirected to their account dashboard or the website's homepage.  POST-3: The user receives a confirmation email to verify their account. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 1. The user clicks the "Register" button on the website.  3. The user fills out the registration form and clicks the "Submit" button.  5. The user receives a confirmation email to verify their account. | | 2. The website displays a registration form, prompting the user to enter their name, email, password, and other required information.  3. The website validates the user's input and creates a new account for the user.  4. The website redirects the user to their account dashboard or the website's homepage. | |
| Alternative Flows: | ALT-1: If the user inputs invalid data in the registration form, the website displays an error message and prompts the user to correct their input.  ALT-2: If the user already has an account on the website, the website displays an error message and prompts the user to log in instead. | | | |
| Exception Flows | EXC-1: If the website's registration feature is down or inaccessible, the website displays an error message and prompts the user to try again later.  EXC-2: If the user does not receive a confirmation email, they can contact the website's customer support for assistance. | | | |
| Business Rules: |  | | | |

## 3.15 Manage Comment.

****

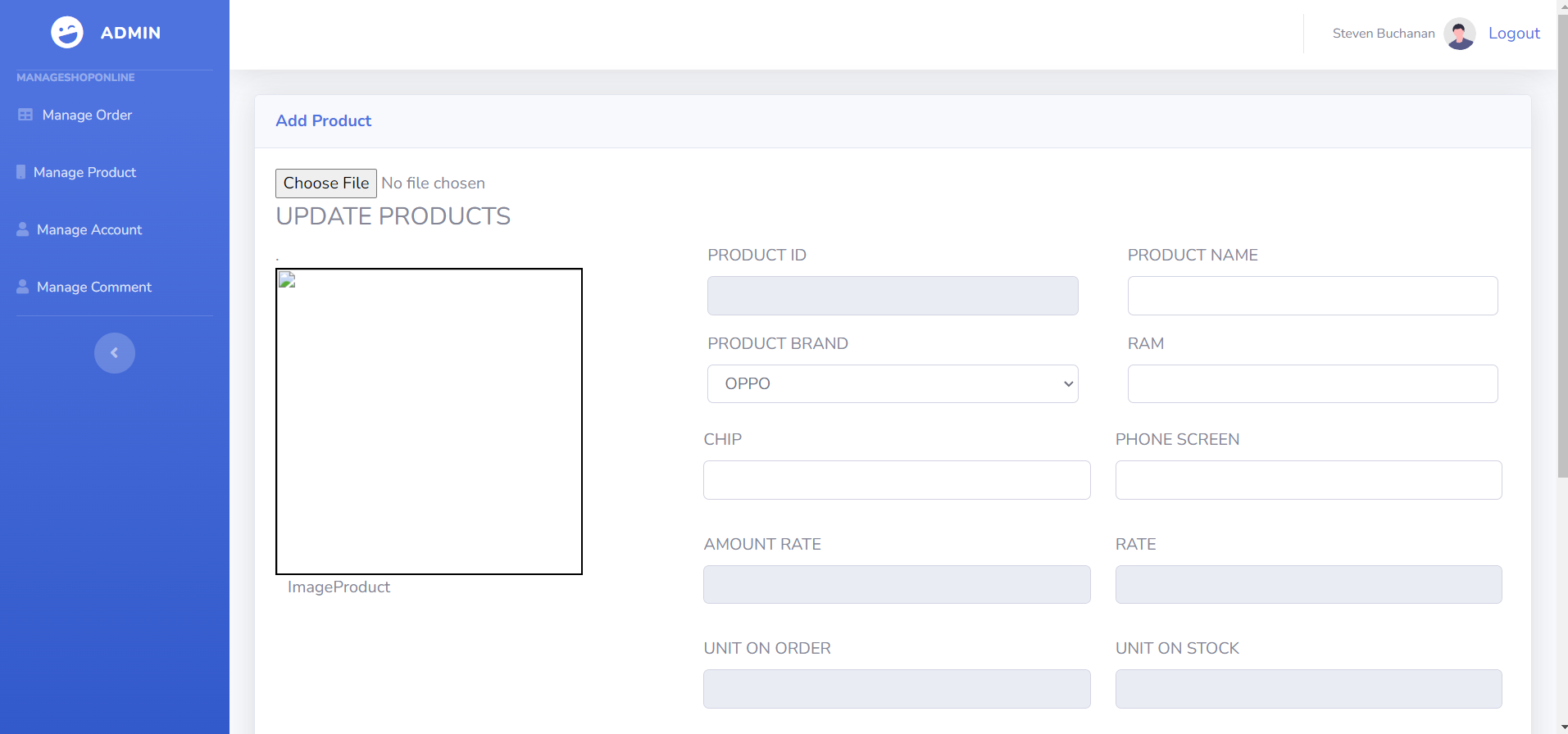
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **15 -** Manage Comment | Actors: | | Admin |
| Description: | This use case describes the process of managing comments on the website by the admin. The admin can manage comments from both customers and employees. This includes viewing, approving, rejecting, and deleting comments. | | | |
| Trigger: | The admin wants to manage the comments on the website. | | | |
| Preconditions: | PRE-1: The admin has logged in to the system.  PRE-2: There are comments submitted by customers and employees on the website. | | | |
| Post-conditions: | POST-1: The comments are either approved, rejected, or deleted.  POST-2: Customers and employees are notified of any changes to their comments.  POST-3: The admin logs out of the system. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 1. The admin selects the "Manage Comments" option from the dashboard.  3. The system displays the full comment along with the user's information (name, email, etc.).  The admin selects an action to take on the comment (approve, reject, or delete).  7. The admin logs out of the system. | | 2. The system presents a list of all comments with their status (approved, rejected, or pending).  The admin selects a comment to view.  4. If the admin approves the comment, the system updates the status to "approved".  5. If the admin rejects the comment, the system updates the status to "rejected" and prompts the admin to provide a reason for rejection.  6. If the admin deletes the comment, the system removes it from the list and prompts the admin to confirm the deletion. | |
| Alternative Flows: | ALT-1. If there are no comments to manage, the system displays a message indicating that there are no comments to manage.  ALT-2. If the admin selects a comment that has already been approved or rejected, the system displays the comment with its current status and disables the approve/reject/delete options.  ALT-3. If the admin selects to reject a comment, but does not provide a reason, the system prompts the admin to provide a reason.  ALT-4. If the admin cancels the deletion of a comment, the system returns to the comment list without making any changes. | | | |
| Exception Flows | EXC-1. If the system fails to retrieve the list of comments, it displays an error message and allows the admin to try again or cancel.  EXC-2. If the system fails to update the status of a comment, it displays an error message and allows the admin to try again or cancel.  EXC-3. If the system fails to delete a comment, it displays an error message and allows the admin to try again or cancel. | | | |
| Business Rules: |  | | | |

## 3.16 Manage Product.

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **16 -** Manage Product | Actors: | | Admin |
| Description: | This use case describes the process of managing products on a website by an admin. | | | |
| Trigger: | The admin logs in to the website and selects the "Manage Products" option from the admin panel. | | | |
| Preconditions: | PRE-1: The admin has valid login credentials and is authorized to access the admin panel.  PRE-2: There are products already listed on the website. | | | |
| Post-conditions: | POST-1: The product information is updated in the database.  POST-2: The admin can view the updated product details on the website. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 2. The admin selects a product from the list to manage.  4. The admin can update any of the product details, including price, quantity, description, or image.  6. The admin logs out of the system. | | 1. The system displays a list of all the products on the website.  3. The system displays the product details such as name, description, image, price, quantity, and other relevant information.  4. The system validates the changes made by the admin and updates the product information in the database.  5. The system displays a confirmation message indicating that the product information has been updated successfully. | |
| Alternative Flows: | ALT-1: If there are no products listed on the website, the system displays a message indicating that there are no products to manage.  ALT-2: If the admin selects a product that does not exist in the database, the system displays an error message indicating that the product could not be found.  ALT-3: If the admin enters invalid product information, the system displays an error message indicating that the changes could not be saved. | | | |
| Exception Flows | EXC-1. If the system encounters an error while updating the product information in the database, the system displays an error message indicating that the changes could not be saved. | | | |
| Business Rules: |  | | | |

## 3.17 Create Product.

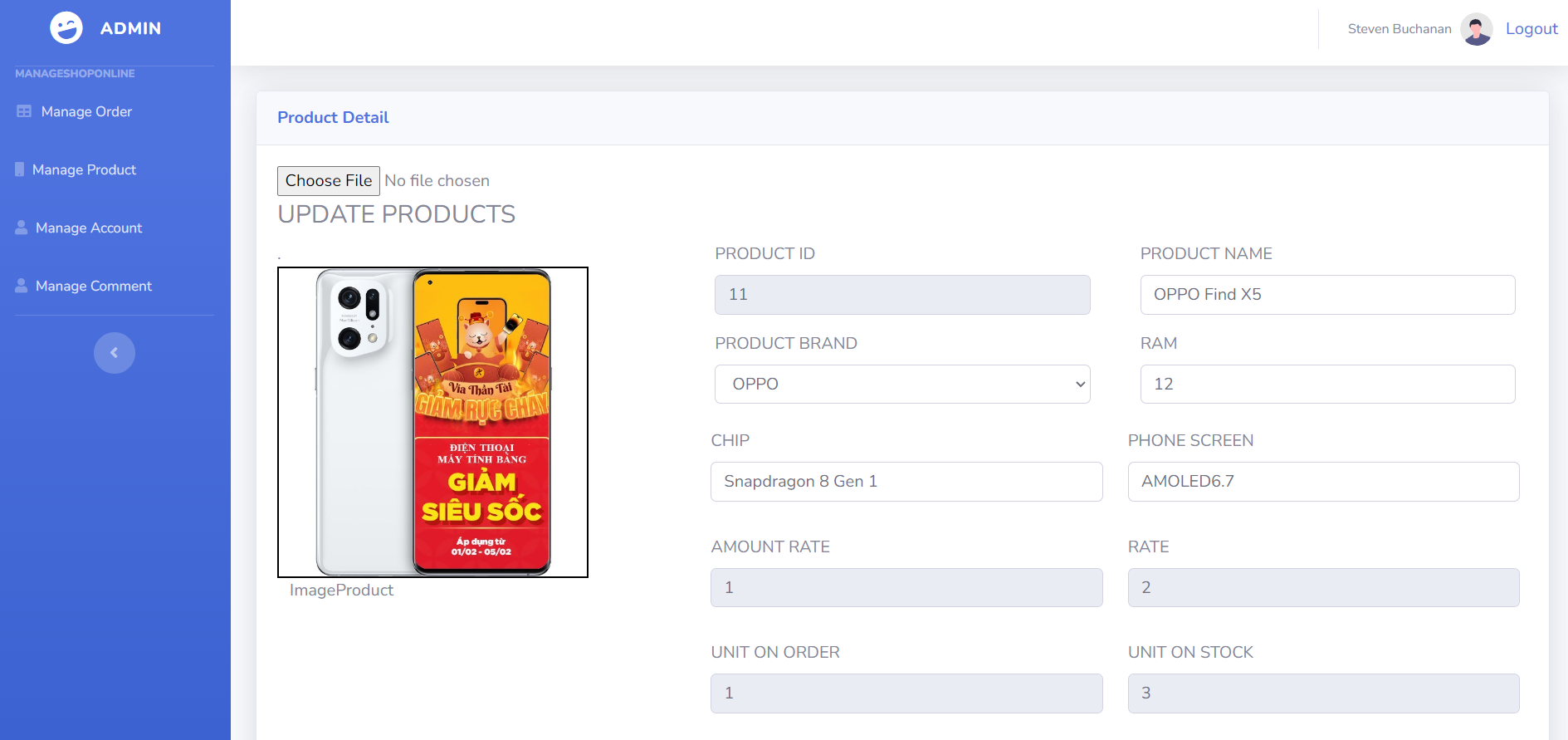
****

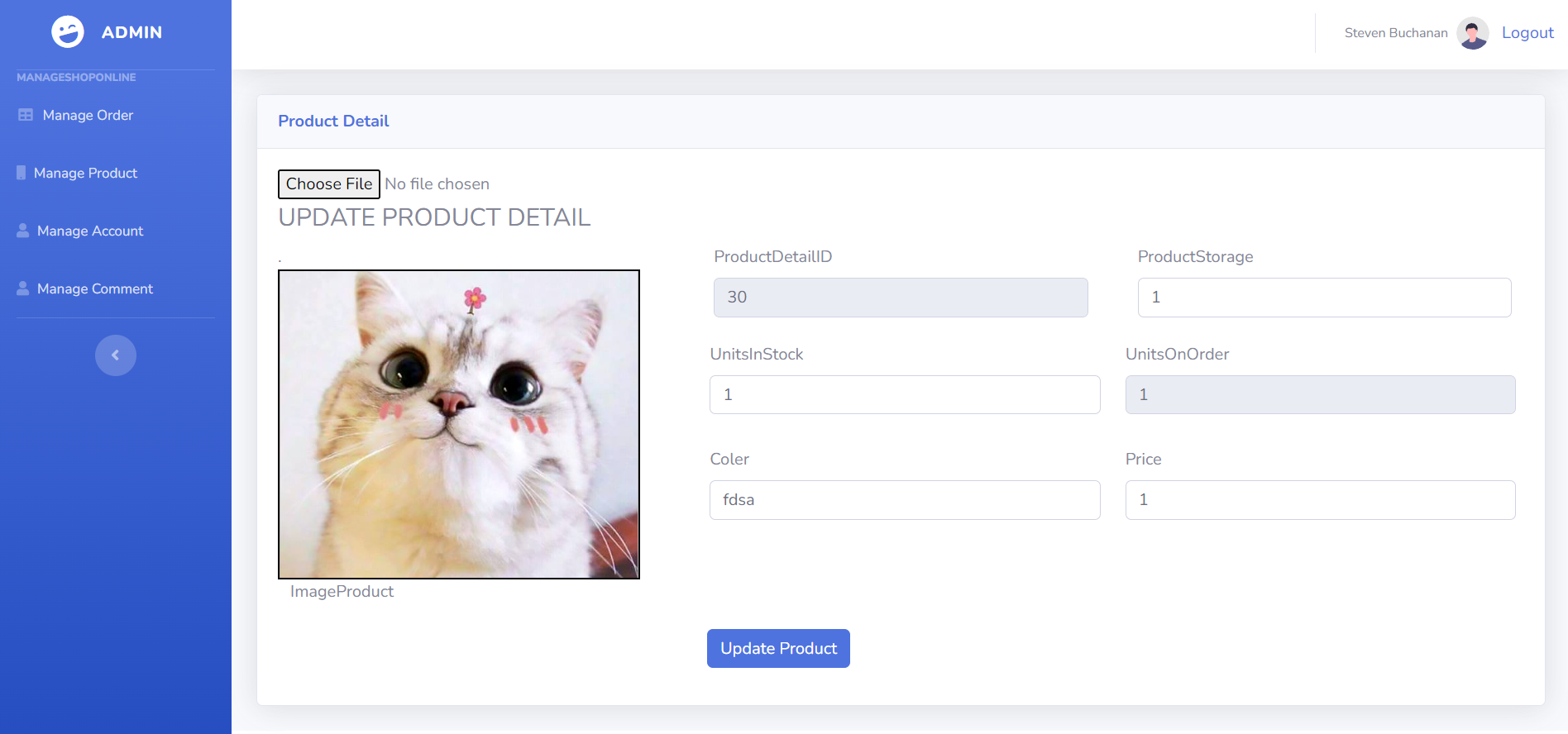
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **17 -** Create Product | Actors: | | Admin |
| Description: | The "Create new Product" use case refers to the process of creating a new product for the website's collection. The admin is responsible for managing the website's collection and can add new products to it as needed. This use case involves the creation of a new product with all necessary details and adding it to the collection. | | | |
| Trigger: | The trigger for this use case is the admin's decision to add a new product to the website's collection. | | | |
| Preconditions: | PRE-1: The admin must be logged in to the system.  PRE-2: The admin must have the necessary privileges to create a new product.  PRE-3: The admin must have all necessary information about the product, such as product name, description, price, and images. | | | |
| Post-conditions: | POST-1: A new product has been added to the website's collection.  POST-2: The admin can view the new product in the collection. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 1. The admin selects the "Create new Product" option from the admin panel.  3. The admin fills in the form with the necessary details of the product.  4. The admin selects the appropriate category for the product. | | 2. The system displays a form with fields for the product name, description, price, and images.  5. The system validates the input data and checks for any errors or missing information.  - If there are any errors or missing information, the system displays an error message, and the admin is prompted to correct the information.  - If the input data is valid, the system creates a new product and adds it to the website's collection.  6. The system displays a success message indicating that the new product has been created and added to the collection. | |
| Alternative Flows: | ALT-1: If the admin decides to cancel the process of creating a new product, the system redirects the admin to the previous page. | | | |
| Exception Flows | EXC-1: If the system encounters an error while creating the new product, it displays an error message and prompts the admin to try again.  EXC-2: If the admin does not have the necessary privileges to create a new product, the system displays an error message informing the admin that they do not have permission to perform this action. | | | |
| Business Rules: |  | | | |

## 3.18 View Product Change.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **18 -** View Product Change | Actors: | | Admin ,Employee |
| Description: | This use case describes the process of viewing the history of changes made to a product in the Website's collection by an admin or employee. | | | |
| Trigger: | The admin or employee selects the option to view the history of a product's changes. | | | |
| Preconditions: | PRE-1: The admin or employee is authenticated and authorized to view the history of product changes.  PRE-2: The product for which the history is to be viewed exists in the Website's collection. | | | |
| Post-conditions: | POST-1: The admin or employee is able to view the history of changes made to the product. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 1. The admin or employee selects the product for which the history is to be viewed.  3. The admin or employee selects the option to view the product's change history. | | 2. The system displays the product details.  4. The system retrieves and displays the history of changes made to the product, including the date and time of each change and the user who made the change. | |
| Alternative Flows: | ALT-1: If the admin or employee is not authenticated or authorized to view the history of product changes, the system displays an error message and the use case ends.  ALT-2: If the product for which the history is to be viewed does not exist in the Website's collection, the system displays an error message and the use case ends. | | | |
| Exception Flows | EXC-1: If there is an error retrieving the history of changes made to the product, the system displays an error message and the use case ends. | | | |
| Business Rules: |  | | | |

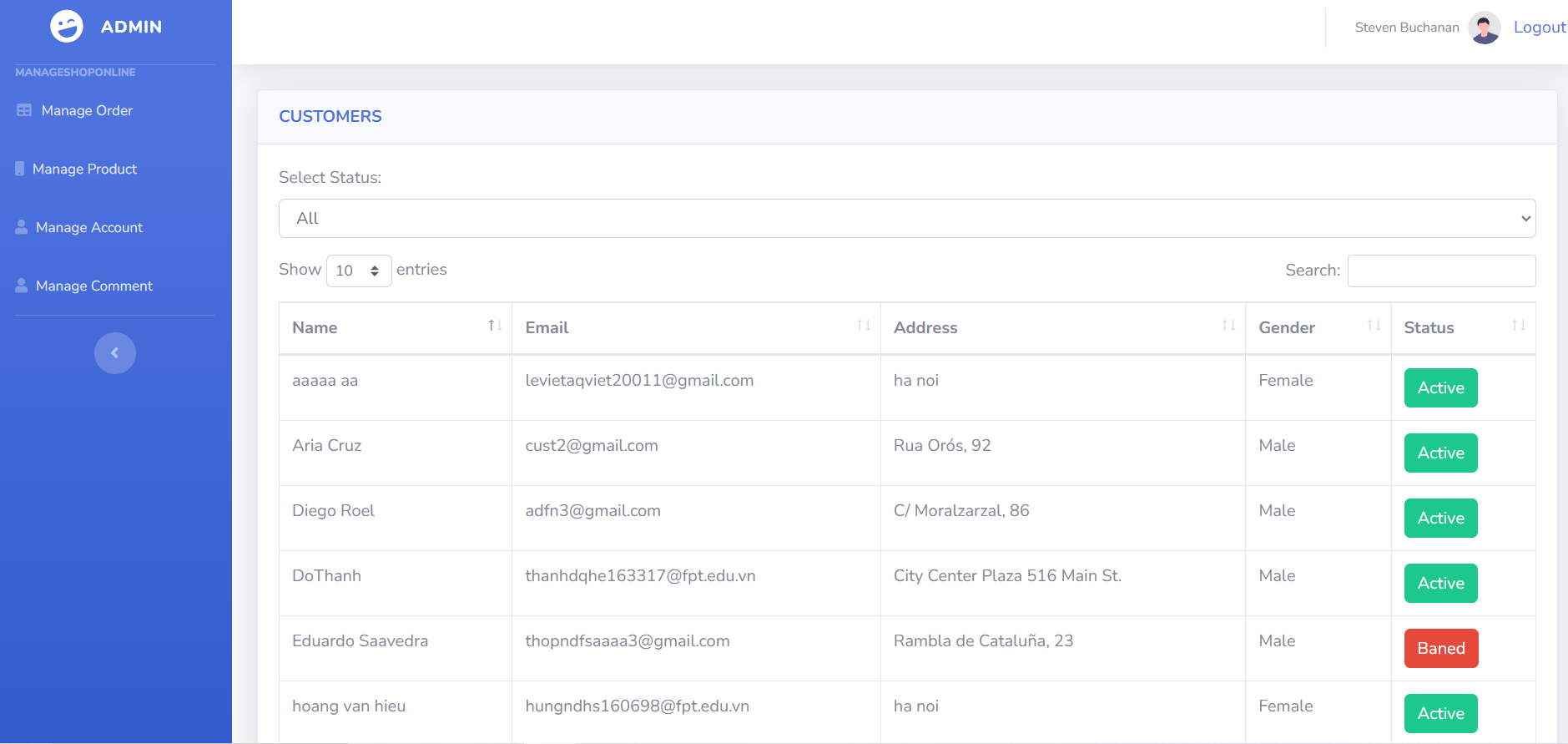
## 3.19 Update Product Detail.

****

****

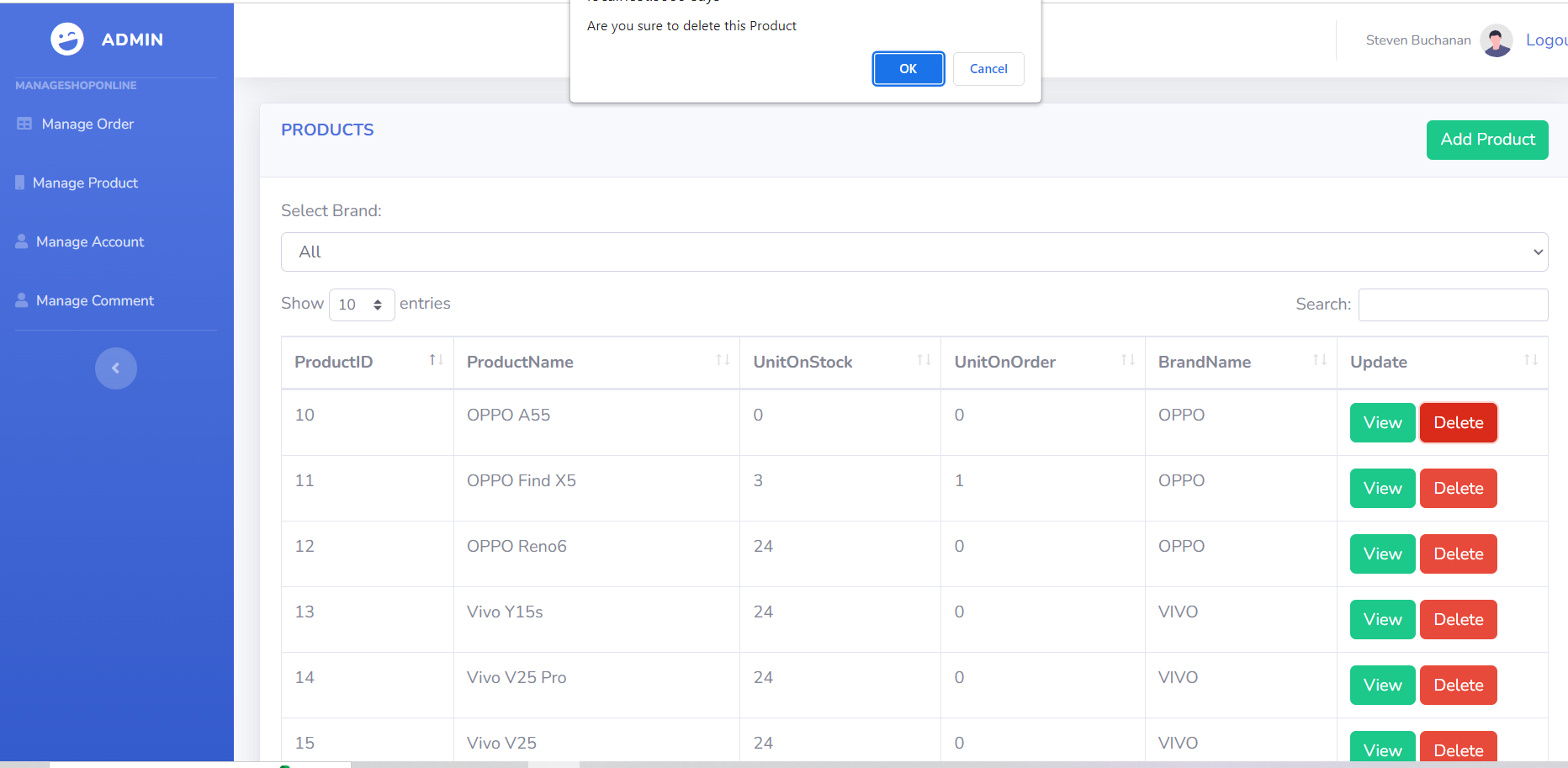
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **19 -** Update Product Detail | Actors: | | Employee |
| Description: | This use case describes the process of updating product details by an admin or employee on a website. | | | |
| Trigger: | The trigger for this use case is when an admin or employee decides to update a product detail. | | | |
| Preconditions: | PRE-1: The admin or employee must be logged in to the website with the necessary permissions to update product details.  PRE-2: The product being updated must already exist in the system. | | | |
| Post-conditions: | POST-1: The product details are updated in the system.  POST-2: The system generates a notification to indicate that the product details have been updated. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 1. The admin or employee navigates to the product management section of the website.  3. The admin or employee selects the product they want to update.  5. The admin or employee modifies the necessary fields and saves the changes. | | 2. The system displays a list of all existing products.  4. The system displays the product details, including fields that can be updated.  6. The system validates the input and updates the product details in the database.  7. The system displays a success message to indicate that the product details have been updated. | |
| Alternative Flows: | ALT-1: If the admin or employee provides invalid input, the system displays an error message and prompts them to correct the input. | | | |
| Exception Flows | EXC-1: If the admin or employee does not have the necessary permissions to update product details, the system displays an error message and does not allow the update to proceed. | | | |
| Business Rules: |  | | | |

## 3.20 Update Status.

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **20 -** Update Status | Actors: | | Employee |
| Description: | This use case allows an employee to update the status of an account to either ban or active. | | | |
| Trigger: | The employee selects an account to update from a list of accounts. | | | |
| Preconditions: | PRE-1: The employee is logged in to the system.  PRE-2: The account to be updated exists in the system. | | | |
| Post-conditions: | POST-1: The status of the account is updated to either ban or active.  POST-2: The system displays a success message to the employee. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 1. The employee selects the "Manage Accounts" option from the main menu.  3. The employee selects the account to be updated.  5. The employee selects the "Update Status" button.  7. The employee selects either "Ban" or "Active" from a drop-down menu.  8. The employee clicks the "Update" button. | | 2. The system displays a list of accounts.  4. The system displays the account details, including the current status.  6. The system displays a form for updating the status.  9. The system updates the status of the account and displays a success message to the employee. | |
| Alternative Flows: | ALT-1: If the employee selects an account that doesn't exist in the system, the system displays an error message and returns the employee to the list of accounts. | | | |
| Exception Flows | EXC-1: If there is an error updating the status of the account, the system displays an error message and prompts the employee to try again later. | | | |
| Business Rules: |  | | | |

## 3.21 Delete Product.

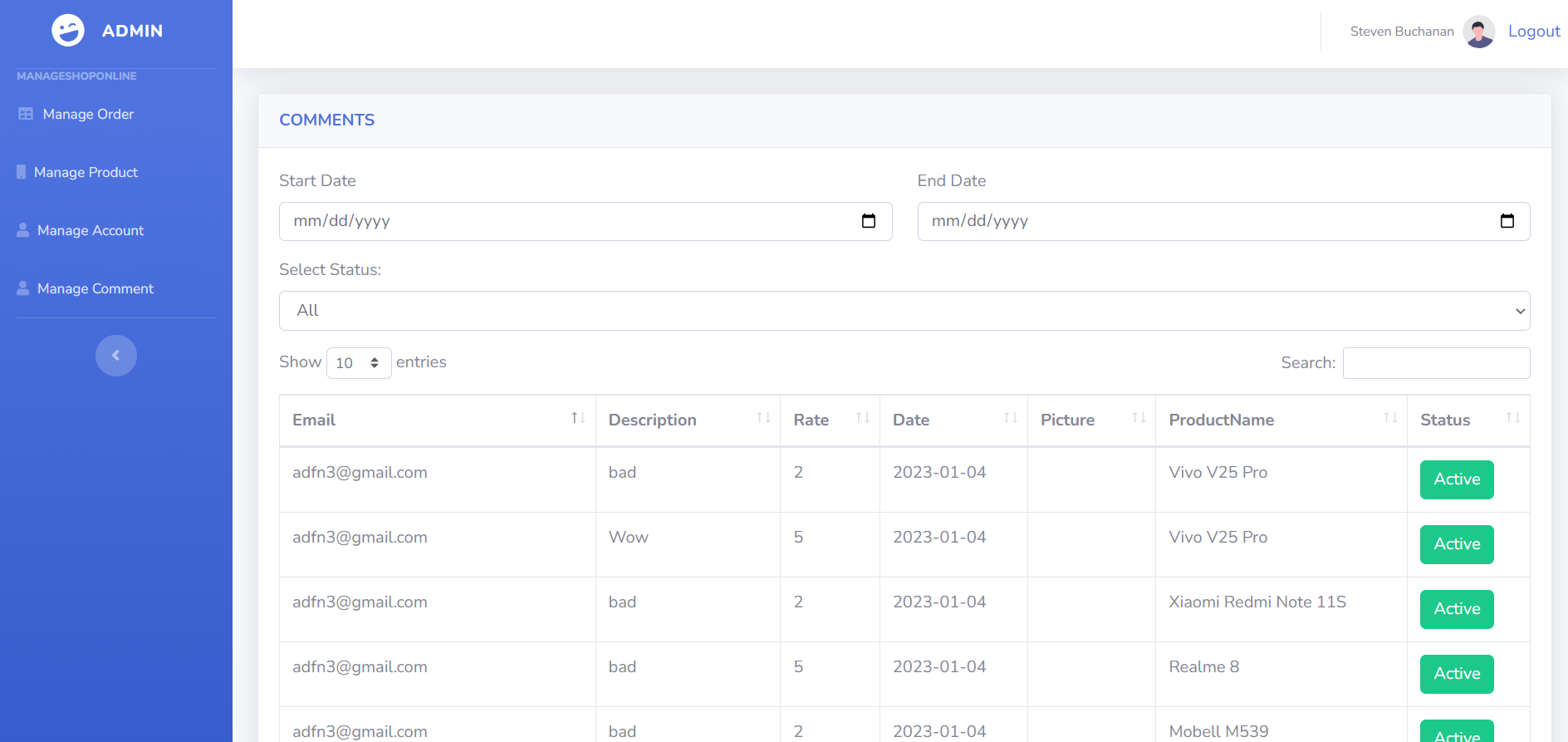
****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **21 -** Delete Product | Actors: | | Admin |
| Description: | This use case describes the process of deleting a product from the website's collection by the admin. | | | |
| Trigger: | The admin wants to remove a product from the website's collection. | | | |
| Preconditions: | PRE-1: The admin is logged in to the admin panel of the website.  PRE-2: The product to be deleted is currently present in the website's collection.  PRE-3: The admin has the necessary permissions to delete products. | | | |
| Post-conditions: | POST-1: The product is no longer present in the website's collection.  POST-2: If the product was associated with any orders or customers, that information is retained in the database. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 1. The admin navigates to the product management section of the admin panel.  2. The admin locates the product they wish to delete and selects the "delete" option.  4. The admin confirms that they want to delete the product. | | 3. The system prompts the admin to confirm that they want to delete the product.  5. The system removes the product from the website's collection and updates the database accordingly.  6. The system displays a message to the admin confirming that the product has been deleted successfully. | |
| Alternative Flows: | ALT-1: If the product cannot be deleted due to it being associated with orders or customers, the system displays an error message to the admin.  ALT-2: If the admin does not have the necessary permissions to delete products, the system displays an error message to the admin. | | | |
| Exception Flows | EXC-1: If there is an unexpected error during the deletion process, the system displays an error message to the admin and may log the error for future debugging. | | | |
| Business Rules: |  | | | |

## 3.22 Manage Account.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **22 -** Manage Account | Actors: | | Admin |
| Description: | This use case involves the admin changing the role of a customer or employee account in the system. | | | |
| Trigger: | The admin selects the "Manage Account" option from the admin dashboard. | | | |
| Preconditions: | PRE-1: The admin is logged into the system with appropriate privileges.  PRE-2: The customer or employee account exists in the system. | | | |
| Post-conditions: | POST-1: The account role has been successfully changed in the system. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 2. The admin selects the account they wish to manage.  4. The admin selects the "Change Role" option.  6. The admin selects the new role for the account.  8. The admin confirms the role change. | | 1. The system displays a list of all customer and employee accounts in the system.  3. The system displays the account details, including the current role.  5. The system displays a list of available roles.  7. The system prompts the admin to confirm the role change.  9. The system updates the account with the new role and displays a success message. | |
| Alternative Flows: | ALT-1: If the admin selects an account that does not exist in the system, the system displays an error message and prompts the admin to try again. | | | |
| Exception Flows | EXC-1: If there is an error updating the account with the new role, the system displays an error message and prompts the admin to try again. | | | |
| Business Rules: |  | | | |

## 3.23 Customer Account.

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **23 -** Customer Account | Actors: | | Admin |
| Description: | This use case describes the process of an admin viewing a list of all customer accounts registered on the website. | | | |
| Trigger: | The admin wants to view a list of all customer accounts. | | | |
| Preconditions: | PRE-1: The admin has logged into the system with their credentials.  The website has at least one registered customer account. | | | |
| Post-conditions: | POST-1: The admin has successfully viewed a list of all customer accounts on the website. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 1. The admin navigates to the "Customers" section of the admin panel.  3. The admin can view the following details for each customer account:  -Account ID  -First name  -Last name  -Email address  -Registration date  -Account status  4. The admin can sort the list by any of the above details.  5. The admin can use a search function to find a specific customer account by entering their account ID, first name, last name, or email address. | | 2. The system displays a list of all customer accounts registered on the website. | |
| Alternative Flows: | ALT-1: If there are no registered customer accounts, the system displays a message indicating that there are no accounts to display. | | | |
| Exception Flows | EXC-1: If the admin does not have the necessary permissions to view customer accounts, the system displays an error message and denies access. | | | |
| Business Rules: |  | | | |

## 3.24 Employee Account.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **24 -** Employee Account | Actors: | | Admin |
| Description: | This use case describes the process of an admin viewing all employee accounts in the system. | | | |
| Trigger: | The admin wants to view a list of all employee accounts in the system. | | | |
| Preconditions: | PRE-1: The admin is logged in to the system.  PRE-2: The admin has the necessary permission to view employee accounts.  PRE-3: There is at least one employee account in the system. | | | |
| Post-conditions: | POST-1: The admin has successfully viewed the list of all employee accounts in the system.  POST-2: The system has displayed the necessary information about each employee account. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 1. The admin navigates to the "View All Employees" page.  3. The admin reviews the list of employee accounts and their details.  4. The admin can click on each employee account to view more detailed information. | | 2. The system displays a list of all employee accounts, including their names, email addresses, and roles. | |
| Alternative Flows: | ALT-1: If there are no employee accounts in the system, the system displays a message indicating this and offers the option to create a new employee account. | | | |
| Exception Flows | EXC-1: If the admin does not have permission to view employee accounts, the system displays an error message and denies access to the "View All Employees" page. | | | |
| Business Rules: |  | | | |

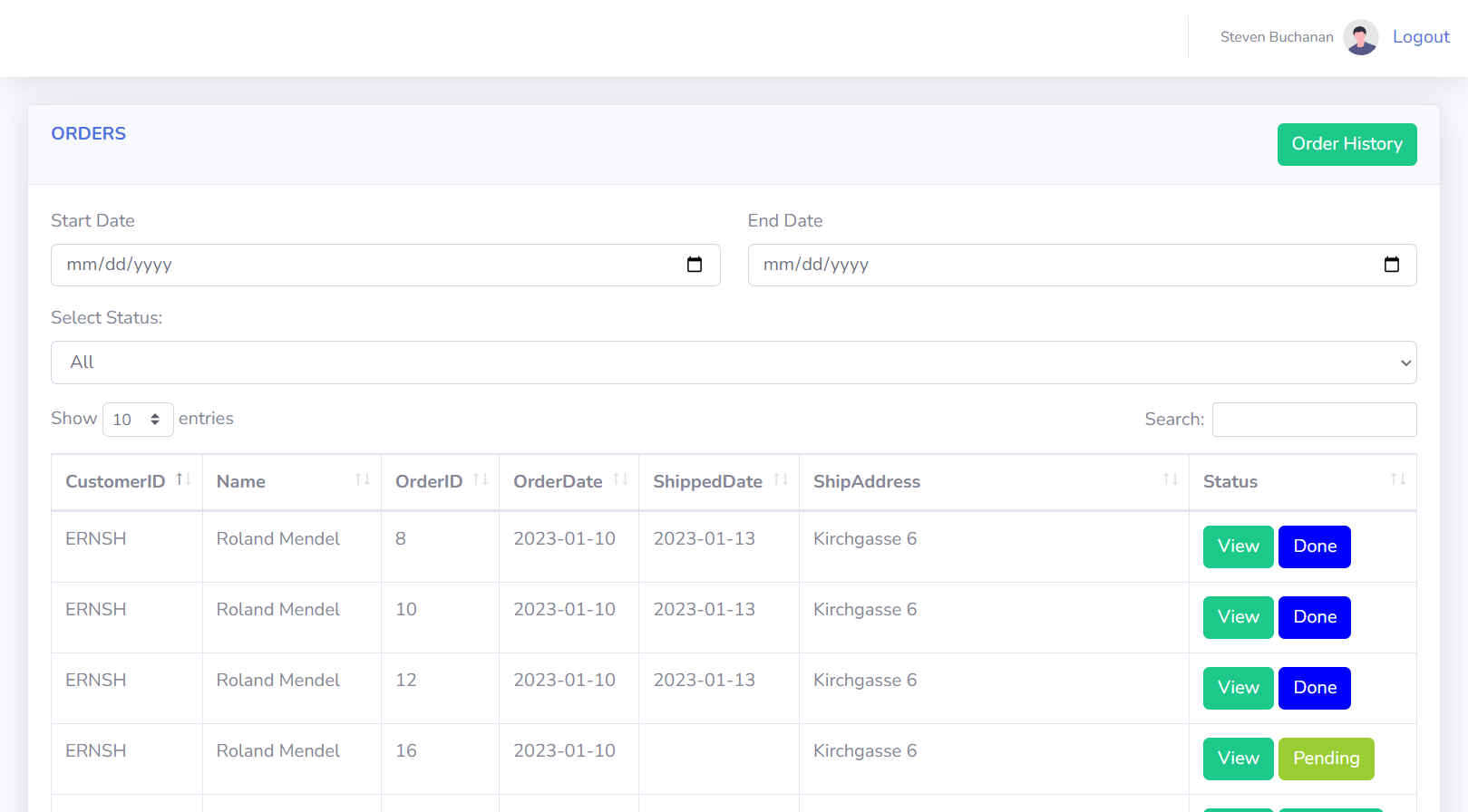
## 3.25 Create Account.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **25 -** Create Account | Actors: | | Admin |
| Description: | This use case describes the process of creating a new employee or customer account by an admin user in the system. | | | |
| Trigger: | An admin user needs to create a new employee or customer account in the system. | | | |
| Preconditions: | PRE-1: The admin user must be logged in to the system.  PRE-2: The admin user has the necessary permissions to create a new account.  PRE-3: The system is functioning properly. | | | |
| Post-conditions: | POST-1: The new employee or customer account has been created and is now available in the system.  POST-2: The new account credentials are sent to the user's email address. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 1. The admin user selects the option to create a new account.  3. The admin user fills out the form with the necessary information. | | 2. The system presents a form for the admin user to fill out with the user's information (e.g., name, email, password).  4. The system validates the information and creates the new account.  5. The system sends an email to the new user's email address with their account credentials. | |
| Alternative Flows: | ALT-1: If the system detects that the information entered in step 3 is invalid, the system presents an error message and prompts the admin user to correct the information.  ALT-2: If the system encounters an error while creating the new account in step 4, the system presents an error message and prompts the admin user to try again later. | | | |
| Exception Flows | EXC-1: If the admin user does not have the necessary permissions to create a new account, the system presents an error message and does not allow the admin user to proceed.  EXC-2: If the system is not functioning properly, the system presents an error message and does not allow the admin user to proceed. | | | |
| Business Rules: |  | | | |

## 3.26 Update Account.

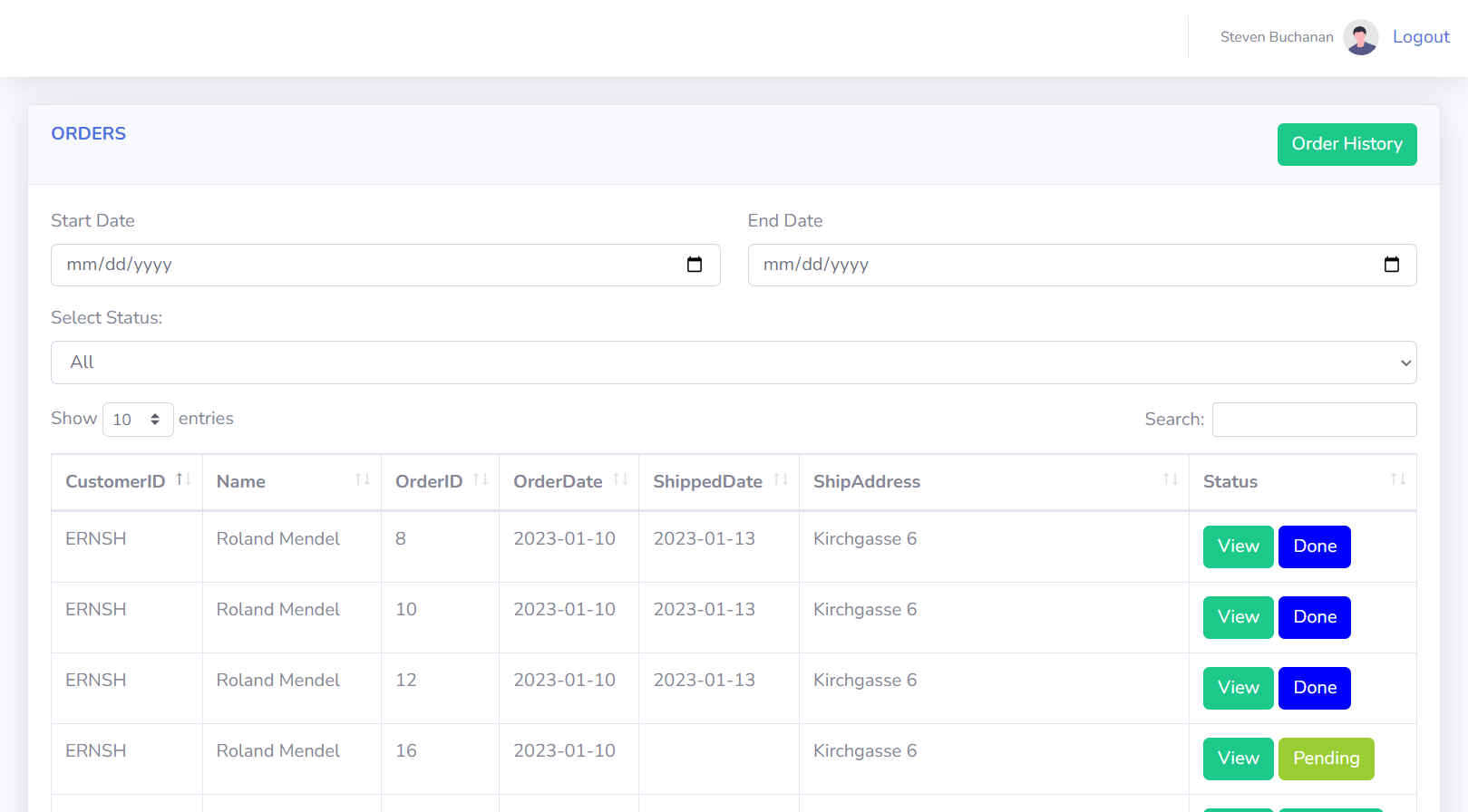
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **26 -** Update Account | Actors: | | Admin |
| Description: | This use case describes the process of updating the information of an employee or customer account by an admin on the website. | | | |
| Trigger: | The trigger for this use case is when the admin logs into the website and navigates to the "Manage Accounts" section. | | | |
| Preconditions: | PRE-1: The admin is logged into the website.  PRE-2: The admin has the necessary permissions to manage employee and customer accounts.  PRE-3: The employee or customer account exists in the system. | | | |
| Post-conditions: | POST-1: The employee or customer account information is updated in the system.  POST-2: The admin receives a confirmation message that the update was successful. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 1. The admin navigates to the "Manage Accounts" section of the website.  2. The admin searches for the employee or customer account they want to update.  3. The admin selects the account and clicks on the "Edit" button.  5. The admin updates the necessary fields and clicks on the "Save" button. | | 4. The system displays the account information in an editable form.  6. The system validates the changes and updates the account information in the system.  7. The system displays a confirmation message that the update was successful. | |
| Alternative Flows: | ALT-1: If the admin cannot find the account they want to update:  The admin can search again using different search criteria.  ALT-2: If the account still cannot be found, the admin can contact the technical support team for assistance. | | | |
| Exception Flows | EXC-1: If the admin tries to update an account with invalid information:  The system displays an error message indicating which field(s) contain invalid information.  The admin corrects the invalid information and tries to save the changes again.  EXC-2: If the system encounters an error while trying to update the account information:  The system displays an error message indicating the reason for the failure.  The admin can retry the update or contact the technical support team for assistance. | | | |
| Business Rules: |  | | | |

## 3.27 Manage Order.

****

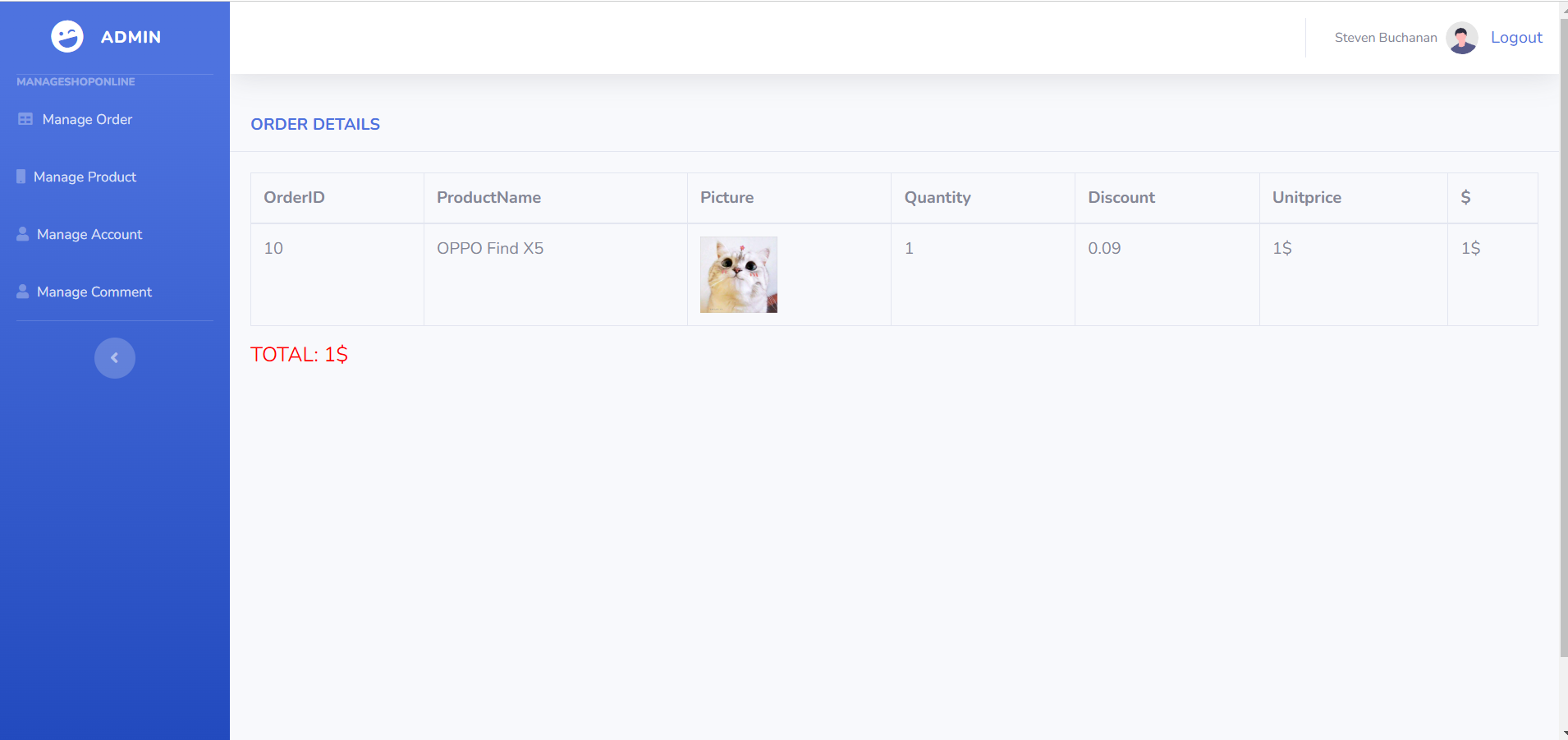
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **27 -** Manage Order | Actors: | | Admin, Employee |
| Description: | This use case involves the management of orders by an admin or employee of an e-commerce website. It includes the ability to view, edit, and delete orders as well as manage the order status. | | | |
| Trigger: | The trigger for this use case is when an admin or employee logs into the system and navigates to the order management section. | | | |
| Preconditions: | PRE-1: The admin or employee has a valid login and access to the order management section.  PRE-2: There are existing orders in the system that need to be managed. | | | |
| Post-conditions: | POST-1: The system updates the order information based on the actions taken by the admin or employee.  POST-2: The updated order information is saved in the system. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 1. The admin or employee logs into the system and navigates to the order management section.  3. The admin or employee selects an order to manage.  3. The admin or employee can perform one or more of the following actions:  - Edit the order details (e.g., change the customer's address)  -Cancel the order  - Mark the order as shipped or delivered  -Refund the order  -After making changes, the admin or employee clicks "Save" to update the order information in the system. | | 2. The system displays a list of all orders in the system.  5. The system confirms that the changes have been saved. | |
| Alternative Flows: | ALT-1: If there are no orders to manage, the system displays a message indicating that there are no orders to manage.  ALT-2: If the admin or employee attempts to edit an order that has already been shipped or delivered, the system displays a message indicating that the order cannot be edited.  ALT-3: If the admin or employee attempts to cancel an order that has already been shipped or delivered, the system displays a message indicating that the order cannot be canceled.  ALT-4: If the admin or employee attempts to mark an order as shipped or delivered before the order has been paid for, the system displays a message indicating that the order cannot be marked as shipped or delivered.  ALT-5: If the admin or employee attempts to refund an order that has already been refunded, the system displays a message indicating that the order cannot be refunded again. | | | |
| Exception Flows | EXC-1: If there is an error saving the updated order information, the system displays an error message and prompts the admin or employee to try again.  EXC-2: If the system crashes while the admin or employee is managing orders, the system displays an error message and prompts the admin or employee to log in again and resume order management. | | | |
| Business Rules: |  | | | |

## 3.28 Update Order Status.

****

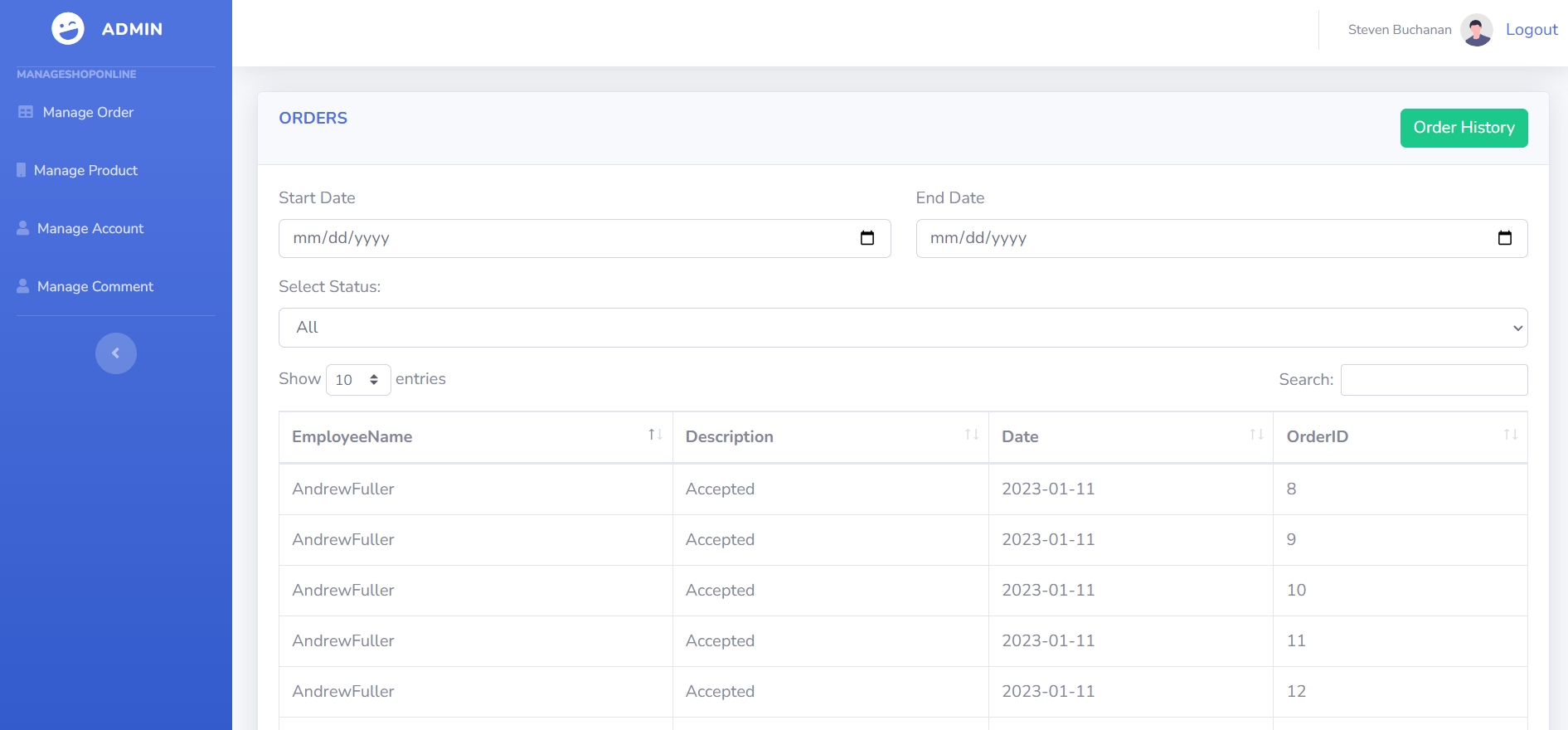
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **28 –** Update Order Status | Actors: | | Admin, Employee |
| Description: | The action of updating the status of an order from one state to another. | | | |
| Trigger: | The trigger for this action is an admin or employee logging into the system and navigating to the order management section. | | | |
| Preconditions: | PRE-1: The admin or employee has appropriate permissions to modify the status of the order.  PRE-2: The order exists in the system and its current status can be updated. | | | |
| Post-conditions: | POST-1: The status of the order is successfully updated.  POST-2: The customer associated with the order is notified of the status update. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 1. The admin or employee navigates to the order management section.  3. The admin or employee selects the order they want to update the status for.  5. The admin or employee selects the new status from a dropdown list. | | 2. The system displays a list of all orders with their current status.  4. The system displays the details of the order, including the current status.  6. The system updates the status of the order to the new value.  7. The system notifies the customer associated with the order of the status update. | |
| Alternative Flows: | ALT-1: If the admin or employee does not have permission to modify the order status, the system displays an error message and the flow ends.  ALT-2: If the order does not exist in the system or the status cannot be modified, the system displays an error message and the flow ends. | | | |
| Exception Flows | EXC-1: If there is an error while updating the status of the order, the system displays an error message and the flow ends. The status of the order remains unchanged. | | | |
| Business Rules: |  | | | |

## 3.29 View Order.

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **29 –** View Order | Actors: | | Admin, Employee |
| Description: | This use case describes the process by which an Admin or Employee user can view the details of a customer's order. | | | |
| Trigger: | The user clicks on an order in the system's list of orders. | | | |
| Preconditions: | PRE-1: The user must be logged in as an Admin or Employee.  PRE-2: There must be at least one order in the system.  PRE-3: The selected order must exist and have been placed by a customer. | | | |
| Post-conditions: | POST-1: The user is able to view the details of the selected order.  POST-2: The system returns the order details, including the customer's name, contact information, order date, items purchased, and order status. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 1. The Admin or Employee clicks on an order in the list of orders.  3. The Admin or Employee can view the order details, including the customer's name, contact information, order date, items purchased, and order status. | | 2. The system retrieves the order details and displays them to the user. | |
| Alternative Flows: | ALT-1: If the selected order does not exist, the system displays an error message and returns the user to the list of orders. | | | |
| Exception Flows | EXC-1: If the user does not have permission to view order details, the system displays an error message and returns the user to the list of orders. | | | |
| Business Rules: |  | | | |

## 3.30 View Order Change.

****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID and Name: | **30 –** View Order Change | Actors: | | Admin, Employee |
| Description: | As an admin or employee, I want to view the history of order changes so that I can see the past changes made to an order's status. | | | |
| Trigger: | The admin or employee selects an order and clicks on the "view order history" button. | | | |
| Preconditions: | PRE-1: The user is logged in as an admin or employee.  PRE-2: The user has access to view the order history.  PRE-3: The order exists in the system. | | | |
| Post-conditions: | POST-1: The system displays the order history.  POST-2: The user can navigate back to the order list page by clicking on the "cancel" button. | | | |
| Normal Flows: | Actor actions | | System responses | |
| 1. The admin or employee selects an order from the list of orders.  2. The admin or employee clicks on the "view order history" button. | | 3. The system retrieves and displays a list of all changes made to the order's status, including the date and time of the change and the user who made the change. | |
| Alternative Flows: | ALT-1: If the user selects an order that does not exist, the system displays an error message.  ALT-2: If the user does not have access to view the order history, the system displays an error message. | | | |
| Exception Flows | EXC-1: If the system encounters an error while retrieving the order history, the system displays an error message.  EXC-2: If the user clicks on the "cancel" button, the system returns to the order list page. | | | |
| Business Rules: |  | | | |

# IV. Non-functional requirements

### 4.1 External Interface

#### 4.1.1 User Interfaces

UI-1: Header: This is the topmost part of the website, containing the website's logo and important links such as Cart, Login, Register, Customer Support, etc.   
UI-2: Menu: Usually placed horizontally or vertically on the left-hand side of the website, the menu contains product categories to help users easily find what they're looking for.   
UI-3: Banner: This is a promotional section, typically located at the center of the website, with attractive images and messages.   
UI-4: Search box: This allows users to search for products based on keywords or categories.  
UI-5:  Product listing: Displays products based on the selected category or search keywords, with information such as name, price, image, ratings, etc.  
UI-6:  Product details: Displays detailed information about a specific product, including images, description, price, ratings, shipping information, etc.   
UI-7: Cart: Displays products that have been added to the cart, allowing users to perform actions such as adding or removing products, or proceeding to checkout.   
UI-8: Login/Register: Allows users to log in or register an account to use the website's features. Customer UI-9: Support: Provides contact information for users to get in touch with the website's customer support team.  
UI-10:  Footer: This is the bottommost part of the website, containing important links such as Home, About Us, Privacy Policy, Terms of Use, etc.

#### 4.1.2 Hardware Interfaces

No hardware interfaces have been identified.

#### 4.1.3 Communications Interfaces

CI-1: User Interface (UI) - The online shopping website provides a user interface for customers to access, search for products and place orders.

CI-2: Sign up/Login Interface: The sign up/login interface helps customers create an account to place orders and manage personal information and shipping addresses.

CI-3: Order Management Interface: The order management interface provides order management features including order status, delivery tracking, and cancellation.

CI-4: Product Review and Comment Interface: The product review and comment interface allows customers to rate the products, write reviews, or comments to share their experiences with others.

CI-5: The system shall send an email to the user’s email to get new password when user forgot her/his account password.

CI-6: User use mouse to communicate with application to handle functions that use need.

CI-7: The system shall display a text message to notify about the result of the function that the user performed.

### 4.2 Quality Attributes

#### 4.2.1 Usability

* User-friendly interface design: The website interface needs to be designed to be simple, easy to understand, and easy to use so that users can easily search and purchase products quickly and conveniently.
* User-friendliness: The website needs to be user-friendly, which can be achieved through user interaction, quality of customer service, website latency, page load speed, etc.
* Compatibility with devices and browsers: The website needs to be designed to be compatible with various devices and popular web browsers in Vietnam, such as Google Chrome, Firefox, Safari, etc.
* Easy-to-use search function: The search function on the website needs to be designed to allow users to search for products easily and conveniently.
* Secure payment: The website needs to provide secure and reliable payment methods so that users can purchase products safely and securely.
* Product reviews and ratings: Reviews and ratings from previous customers will help users get more information about the products before making a purchase and create trust with the website.

#### 4.2.2 Availability

Zappy shall be available at least 98% of the time between 5:00 A.M and midnight local time, and at least 90% of the time between midnight and 5:00 A.M local time, excluding scheduled maintenance servers.

#### 4.2.3 Performance

All pages have loading speed less than 3 seconds on average • The system can support up to 500 requests at the same time; The system allows up to 1200 students to register in system per semester. • The system shall display confirmation messages to users within an average of 1.5 seconds and a maximum of 3 seconds after the user submits information to the system.

#### 4.2.4 Security

HTTPs between client and server.

 • Token-based authentication using JWT.

 • Encrypt password with BCrypt algorithm.

#### 4.2.5 Safety

The system shall not spread virus into the devices of users. • The system shall not include any harm information for users. • The system shal notify the user to confirm when performing any function that affects the database.

#### 4.2.6 Support Documents

* <https://s3-sgn09.fptcloud.com/fustorage/files/attachfiles/Report3_Software_Requirement_Specification_fc88de8c78384fb9996b8a5e29a14196.pdf?fbclid=IwAR1GrsmL1pfu0Y4F9UWO1nUzFQIQMMsCHe9z0zpAD4uoDIefSyVnDKBNtYE>.

# V. Requirements Appendix

## 1.Business Rules

|  |  |
| --- | --- |
| **ID** | **Rule Definition** |
| BR-01 | Only verified users can log in successfully and access the system |
| BR-02 | Users only can access information that belongs to their own, except admin. |
| BR-03 | Employee only login with an account that is provided by admin |
| BR-04 | Each user has exactly one role. |
| BR-05 | Each email is only registered for one user. |
| BR-06 | Email must be the right format: “abc123@example.com”. |
| BR-07 | The username must be unique. |
| BR-08 | Passwords must be encrypted. |
| BR-09 | When registering or changing a password, the user must enter the password twice. |
| BR-10 | The date of birth must be before the current time. |
| BR-11 | All lists must be displayed with the pagination. |
| BR-12 | The admin is only allowed to reset password when the admin receives a request by the user. |
| BR-13 | Only create order when have product in cart and logged in the system |
| BR-14 | Each pet only have only one |

## 2.Common Requirement

* The pop-up form is used for functions such as viewing, adding, editing, and deleting.
* Account in the database is not deleted when using the delete function; instead, it is changed from active to inactive.
* The system must display the result of performing a data-related function at the end of the function.

## 3.Application Messages List

|  |  |  |
| --- | --- | --- |
| **ID** | **Type** | **Context** |
| M-01 | In red, under the text box | The user does not select or enter required field |
| M-02 | In red, under the text box | The user entered an email invalid |
| M-03 | In red, under the text box | The user entered an existing email in the database |
| M-04 | In red, under the text box | The user entered an existing username in the database |
| M-05 | In red, under the text box | The user entered a username invalid or wrong form |
| M-06 | In red, under the text box | The user entered password invalid |
| M-07 | In red, under the text box | The user entered phone invalid |
| M-08 | In red, under the text box | The user entered a password has length less than 8 characters or greater than 20 characters |
| M-09 | In red, under the text box | The user entered a re-password does not match with password |
| M-10 | In green, on alert | User edited account successfully |
| M-11 | In green, on alert | User reset password of account successfully |
| M-12 | In green, on alert | Create order successful |