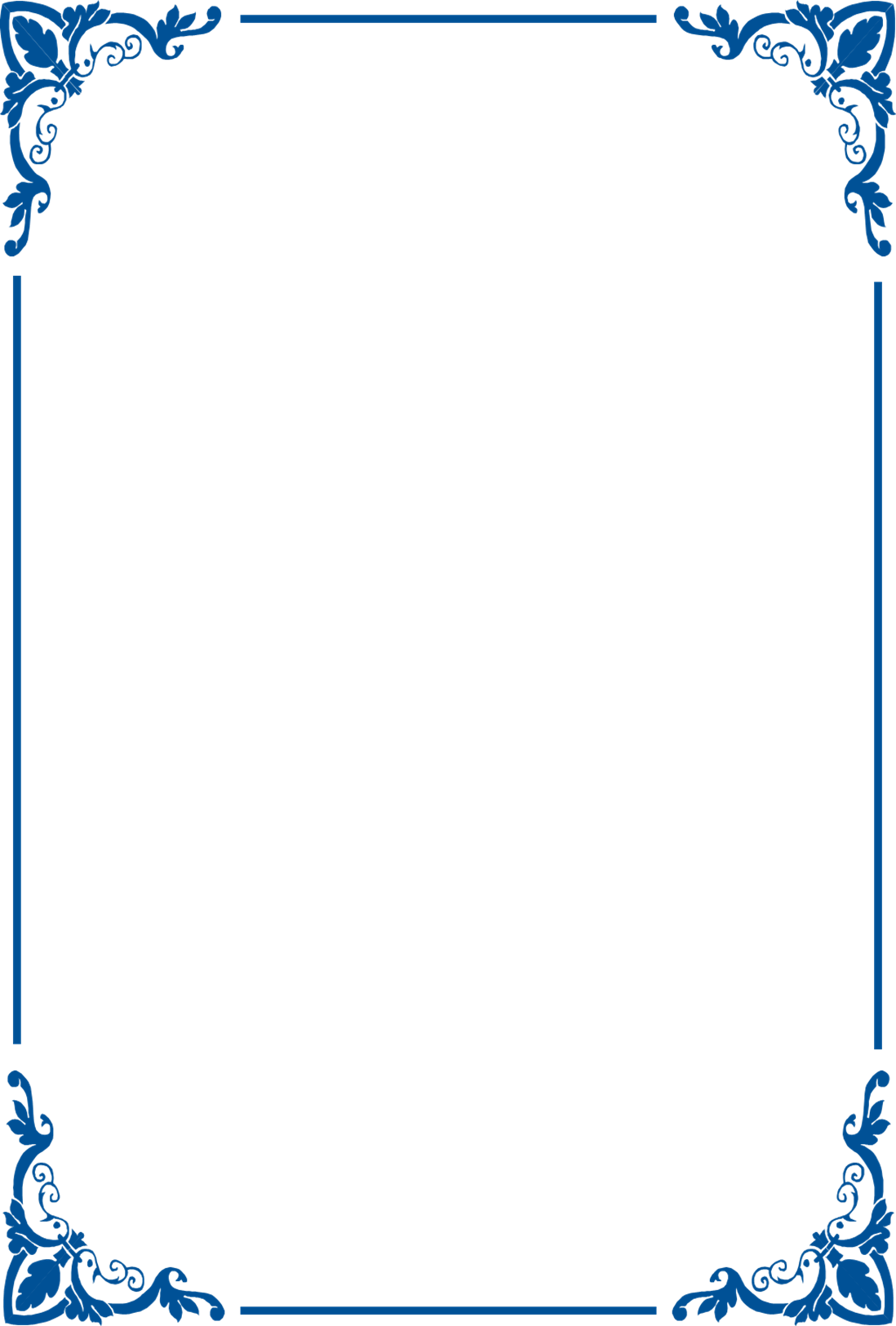
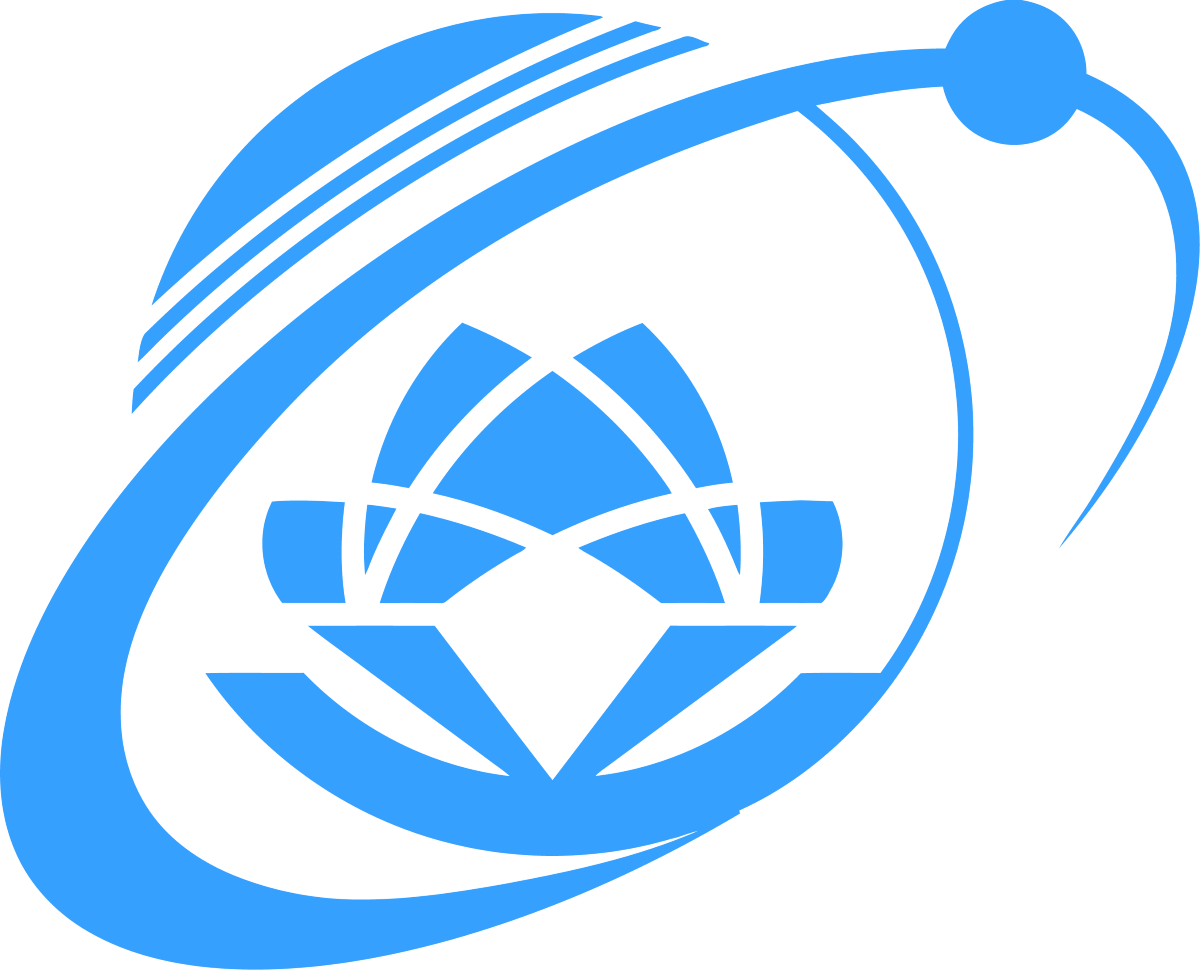
**UNIVERSITY OF INFORMATION TECHNOLOGY**

**ADVANCED PROGRAM IN INFORMATION SYSTEMS**



**LAB 01 REPORT**

**SOFTWARE ENGINEER**

**Class:**   **CS4273.O11.CTTT**

**Lectures:**  **Nguyen Tuan Nam**

**Tran Vinh Khiem**

**Members: Nguyen Chu Nguyen Chuong - 22520165**

**Vo Ha Minh Huy - 22520585**

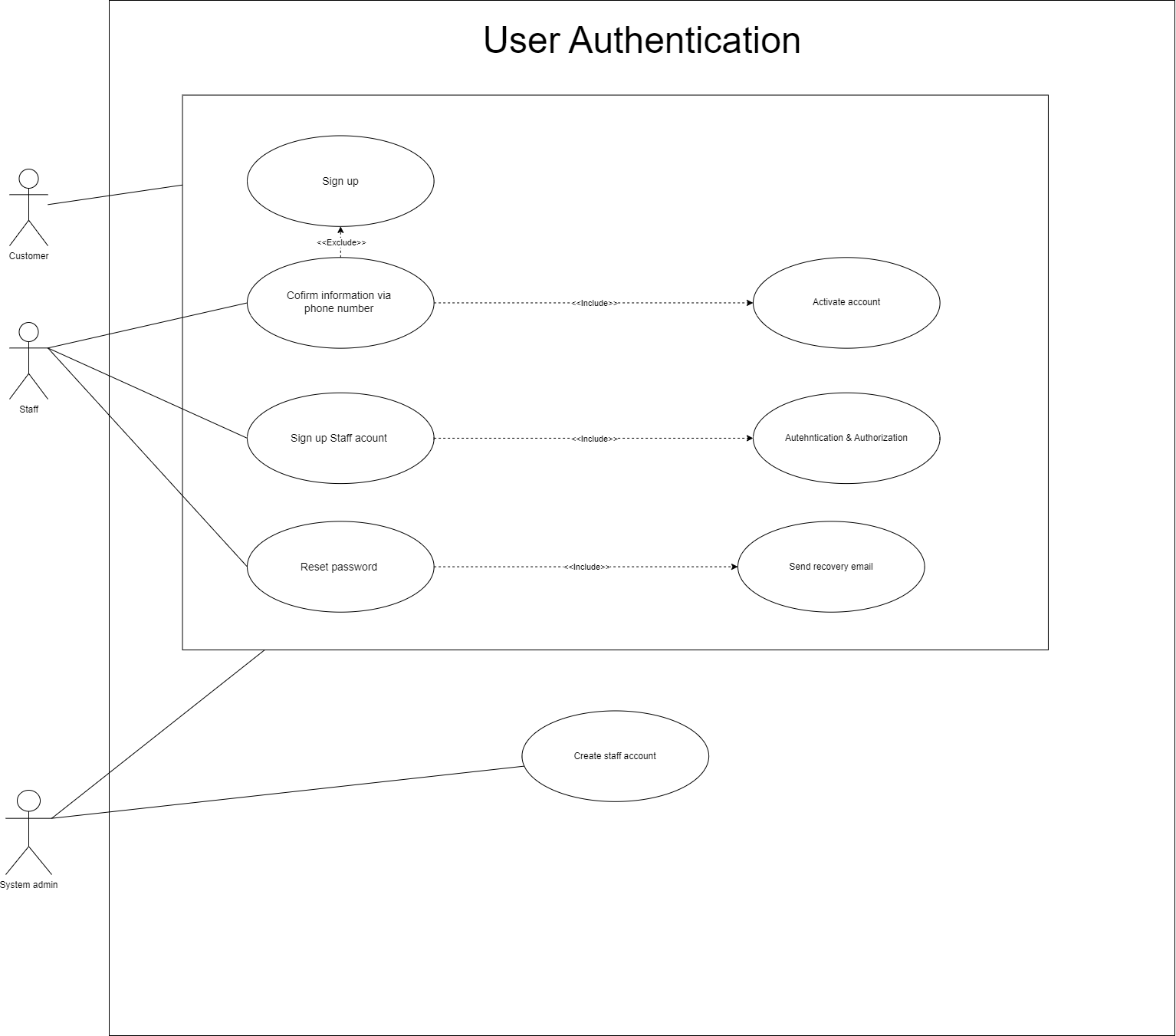
**Ho Tan Loc - 22520786**

**Ngo Thi Hong Ly - 22520836**

**Le Hoang Thien Phu - 22521099**

**Le Minh Sang - 22521240**

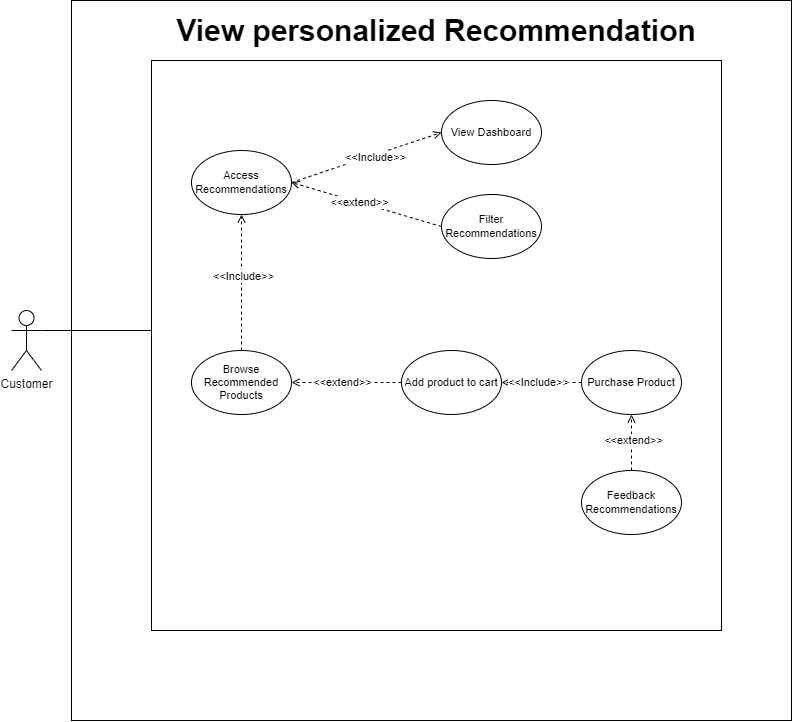
1. User Authentication
   1. UML Diagram



* 1. Table Description

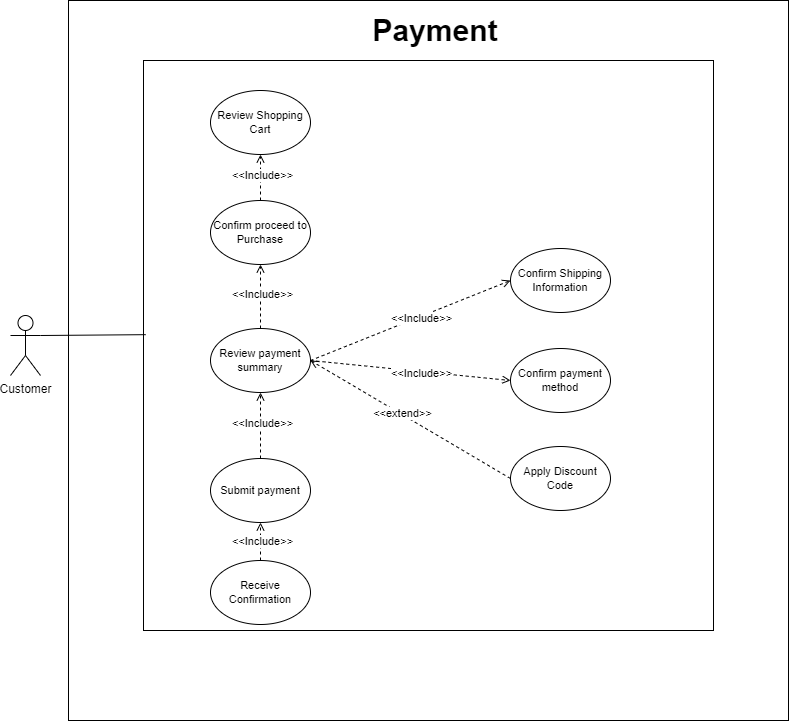
|  |  |
| --- | --- |
| Name | Sign up |
| Scenario | Create an account for new user who want to buy needed products |
| Description | This use case describes the process of creating an account in the system, which includes email confirmation to verify user information. After completing the sign-up process, the user will be redirected to the system's homepage. |
| Actor | Customers |
| Trigger | The customer clicks the sign-up button, enters their email, password, and basic information (such as name and gender), then submits the form. |
| Related use cases | Confirm information via email, Activate account |
| Pre-condition | The user has accessed the system's website and is able to access the sign-up function. |
| Post-condition | * **Success:** Users have system access rights * **Fail:** The system notifies an invalid account and requires to create an account again. |
| Flow of events | 1. User navigates to the store website. 2. User clicks the "Sign Up" button. 3. User is redirected to the sign-up page. 4. User inputs their username, password, and basic information, then submits the form. 5. The system checks the validity of the entered email, password, and other information. 6. The system displays a waiting page for email verification. 7. If the user is activated, they are redirected to the homepage of the system. |
| Exception conditions | * If the entered email or password already exists, the system notifies the user to log in instead. * If any of the entered information is invalid, the system prompts the user to try again. |

1. View Personalized Recommendations
2. UML Diagram



1. Table Description

|  |  |
| --- | --- |
| Name | View personalized product recommendations |
| Scenario | Customers view personalized product recommendations based on their previous browsing and purchase behavior |
| Description | This use case describes the process where a customer logs into the system and access product recommendations based on their browsing history, purchase history, and interactions. The system analyzes user behavior and generates personalized suggestions, displaying relevant products on the user's homepage |
| Actor | Customers |
| Trigger | The customer navigates to the dashboard, access to the recommendation page where the system automatically generates and displays personalized product recommendations. |
| Related use cases | Browse Products: Customer browsing activity contributes to future recommendations.  Purchase Product: Customer purchase history influences recommendations. |
| Pre-condition | * The customer has logged into the system. * The system has previously tracked user behavior, including browsing or purchase history. |
| Post-condition | * Success: Personalized recommendations are displayed on the screen, and the user can browse, add to the cart, or purchase recommended products. * Fail: If insufficient data is available, the system displays default or trending products. |
| Flow of events | 1. The user logs into the system or navigates to the homepage/dashboard. 2. The system retrieves the customer’s browsing history, purchase history, and interactions. 3. The system analyzes the collected data to identify relevant products. 4. The system generates a list of recommended products personalized to the customer. 5. The recommendations are displayed on the homepage/dashboard. 6. The customer can interact with the recommendations by clicking on a product to view details, adding products to the cart, or purchasing them. |
| Exception conditions | * Insufficient Data: If the system does not have enough data on the customer (e.g., a new user), it displays default recommendations or popular products. * Data Retrieval Error: If the system cannot retrieve the customer’s data, it notifies the customer and displays a generic list of products. * Inconsistent Behavior Data: If conflicting data is detected (e.g., the customer’s browsing and purchase history show different patterns), the system adjusts the recommendations or offers a broader set of products. |

1. Payment
2. UML Diagram
3. Table Description

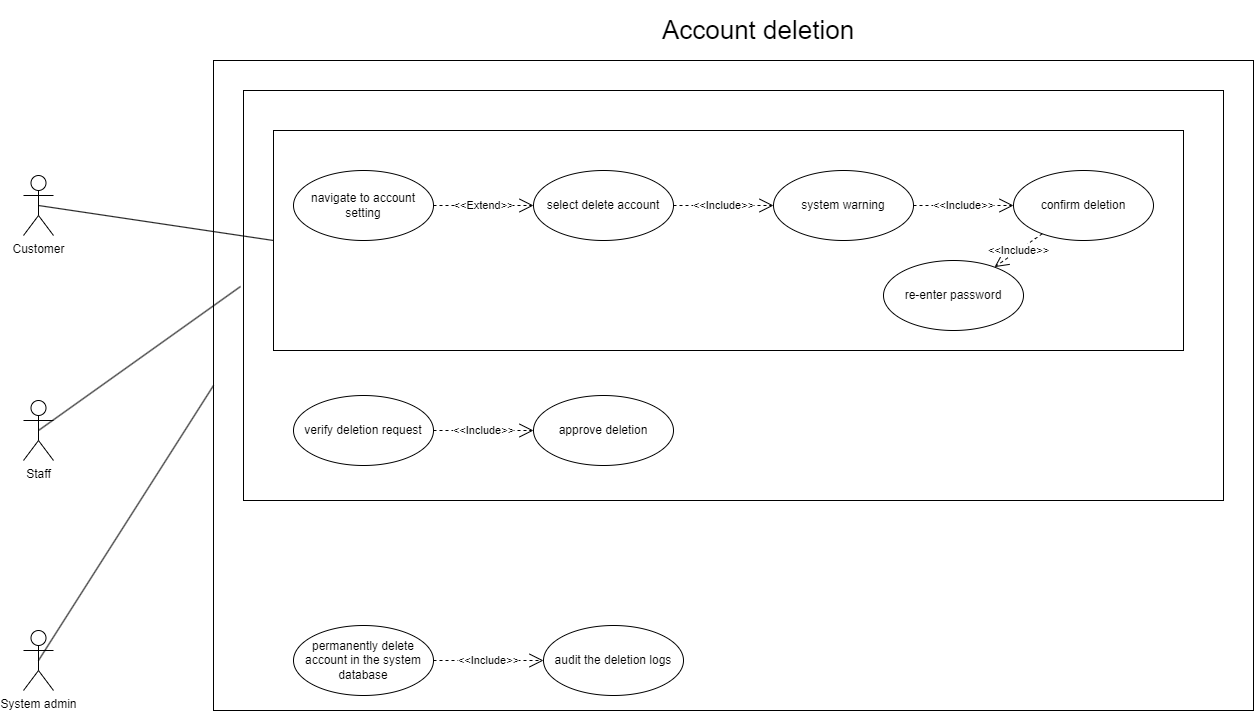
|  |  |
| --- | --- |
| Name | Payment |
| Scenario | A customer completes the payment process after selecting products to purchase. |
| Description | This use case describes the process where a customer initiates and completes payment for products they have selected in their shopping cart. The system processes payment details, confirms payment. The customer is provided with a receipt confirmation. |
| Actor | Customers |
| Trigger | The customer proceeds to checkout after selecting items to purchase from the shopping cart. |
| Related use cases | **Apply Discount**: (Optional) Customer can apply a coupon or promotional code before completing payment. |
| Pre-condition | * The customer is logged in and has products in the shopping cart. * The customer has valid payment information ready to be processed. |
| Post-condition | * Success: The payment is successfully processed, and the order is confirmed. The customer receives a receipt. * Failure: The system notifies the customer of payment failure (e.g., insufficient funds, invalid card) and prompts them to retry or use another payment method. |
| Flow of events | 1. The customer navigates to the shopping cart and clicks the "Proceed to Checkout" button. 2. The system displays the order summary, including product details, prices, and shipping costs. 3. The customer inputs or confirms payment details (e.g., credit card, PayPal). 4. (Optional) The customer applies a discount code if available. 5. The system validates the payment details and processes the payment. 6. The system confirms successful payment. 7. The customer receives a receipt, and the system updates the order status. |
| Exception conditions | * Invalid Payment Information: If the customer enters invalid payment details, the system prompts the customer to correct the information. * Insufficient Funds: If the payment fails due to insufficient funds, the system notifies the customer and suggests using another payment method. * Expired Discount Code: If the customer applies an expired or invalid discount code, the system notifies the customer and allows them to continue without it. |

1. Voucher Interactions
   1. A black background with white ovals

      Description automatically generatedUML Diagram
   2. Table Description

|  |  |
| --- | --- |
| Name | Voucher Interaction |
| Scenario | The customer uses a voucher to apply discounts on purchases, while staff create and manage vouchers, and the system admin oversees the process. |
| Description | The customer can view, redeem, or apply a voucher. Staff create vouchers for their store's products, and the system admin manages and audits voucher usage. |
| Actor | Customer, Staff, System Admin |
| Trigger | The customer initiates a purchase and decides to apply for a voucher. |
| Related use cases | - View Available Vouchers  - Redeem Voucher  - Apply Voucher  - Create Voucher  - Expire Voucher  - Audit Voucher Usage |
| Pre-condition | - Customer is logged in and has an available voucher.  - Staff has created vouchers.  - System policies are in place for voucher usage. |
| Post-condition | - The voucher is successfully applied to the customer’s purchase.  - The voucher is marked as redeemed.  - Transaction is complete. |
| Flow of events | 1. Customer logs in.  2. Customer views available vouchers.  3. Customer proceeds to checkout.  4. Customer selects or manually enters a voucher code.  5. System validates the voucher.  6. Voucher is applied if valid, and total order amount is updated.  7. Voucher is marked as redeemed in the system |
| Exception conditions | - Voucher Invalid: The voucher code is incorrect, expired, or not applicable.  - Voucher Already Used: The voucher has already been redeemed by the customer.  - Minimum Spend Not Met: The customer’s order does not meet the conditions to use the voucher.  - Voucher Not Applicable: The voucher is not valid for the items in the cart. |

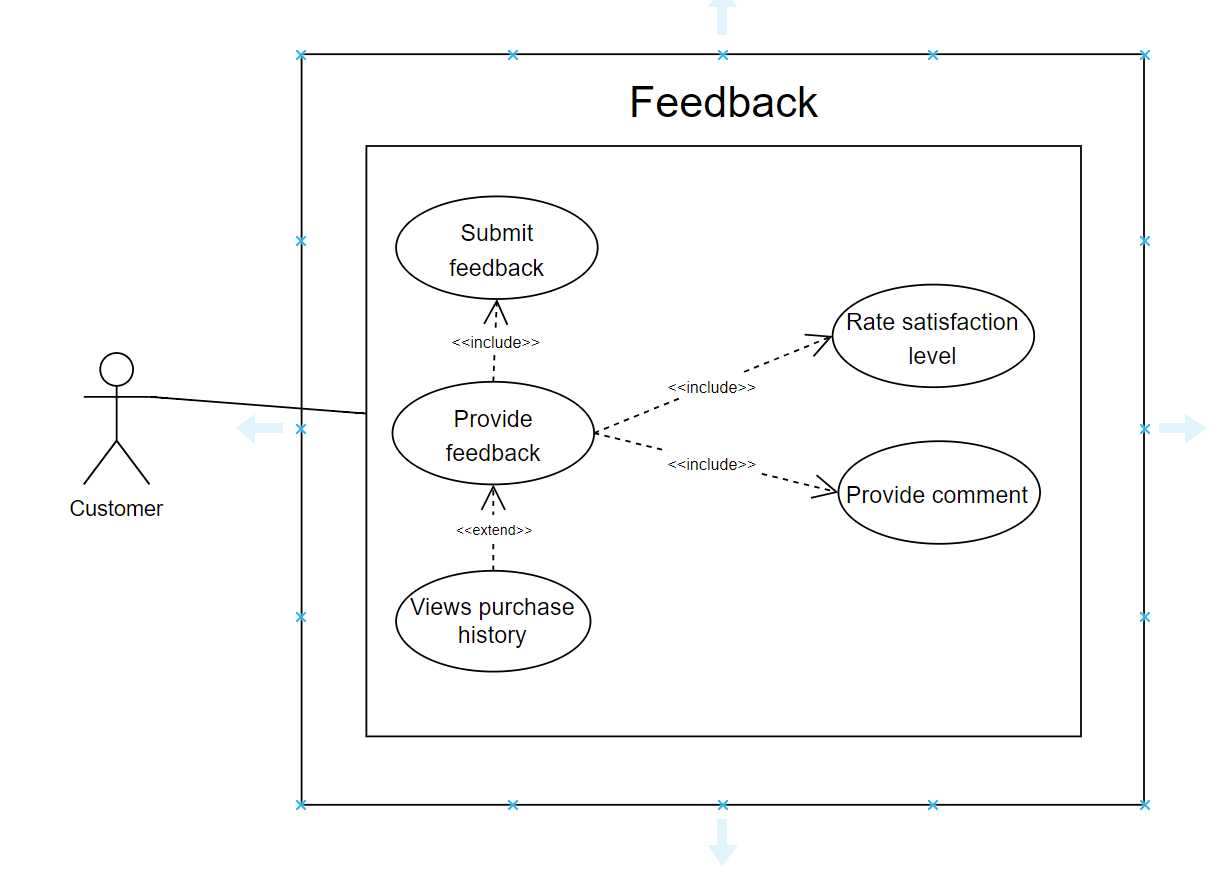
1. Account Deletion
   1. UML Diagram



* 1. Table Description

|  |  |
| --- | --- |
| Name | Delete account |
| Scenario | The customer requests account deletion, staff verify and approve the request, and the system admin handles the final deletion and audit of the process. |
| Description | The customer initiates the process of deleting their account. Staff verify the request to ensure that any pending transactions are completed, and the system admin executes the permanent deletion of the account. |
| Actor | Customer, Staff, System Admin |
| Trigger | The customer decides to delete their account, triggering the deletion process. |
| Related use cases | - Request Account Deletion  - Verify Deletion Request  - Approve Deletion  - Permanently Delete Account  - Audit Deletion Logs |
| Pre-condition | - The customer is logged in.  - The customer has no unresolved or active transactions.  - Staff are available to review deletion requests. |
| Post-condition | - The customer’s account is permanently deleted.  - All associated data, including order history and personal information, is removed from the system.  - The system logs the deletion for auditing purposes. |
| Flow of events | 1. Customer logs in.  2. Customer navigates to the account settings and selects "Delete Account".  3. Customer confirms the deletion request and enters their password.  4. Staff receives the request, verifies that no pending transactions are associated with the account.  5. Staff approves the deletion.  6. System admin permanently deletes the account and audits the deletion logs. |
| Exception conditions | - Pending Transactions: The account cannot be deleted due to unresolved transactions.  - Verification Failure: The customer’s identity could not be verified, and the deletion request is rejected.  - Cancellation: The customer cancels the deletion request during the grace period.  - Staff Rejection: Staff reject the deletion request if any issues are found, such as disputes or unresolved refunds. |

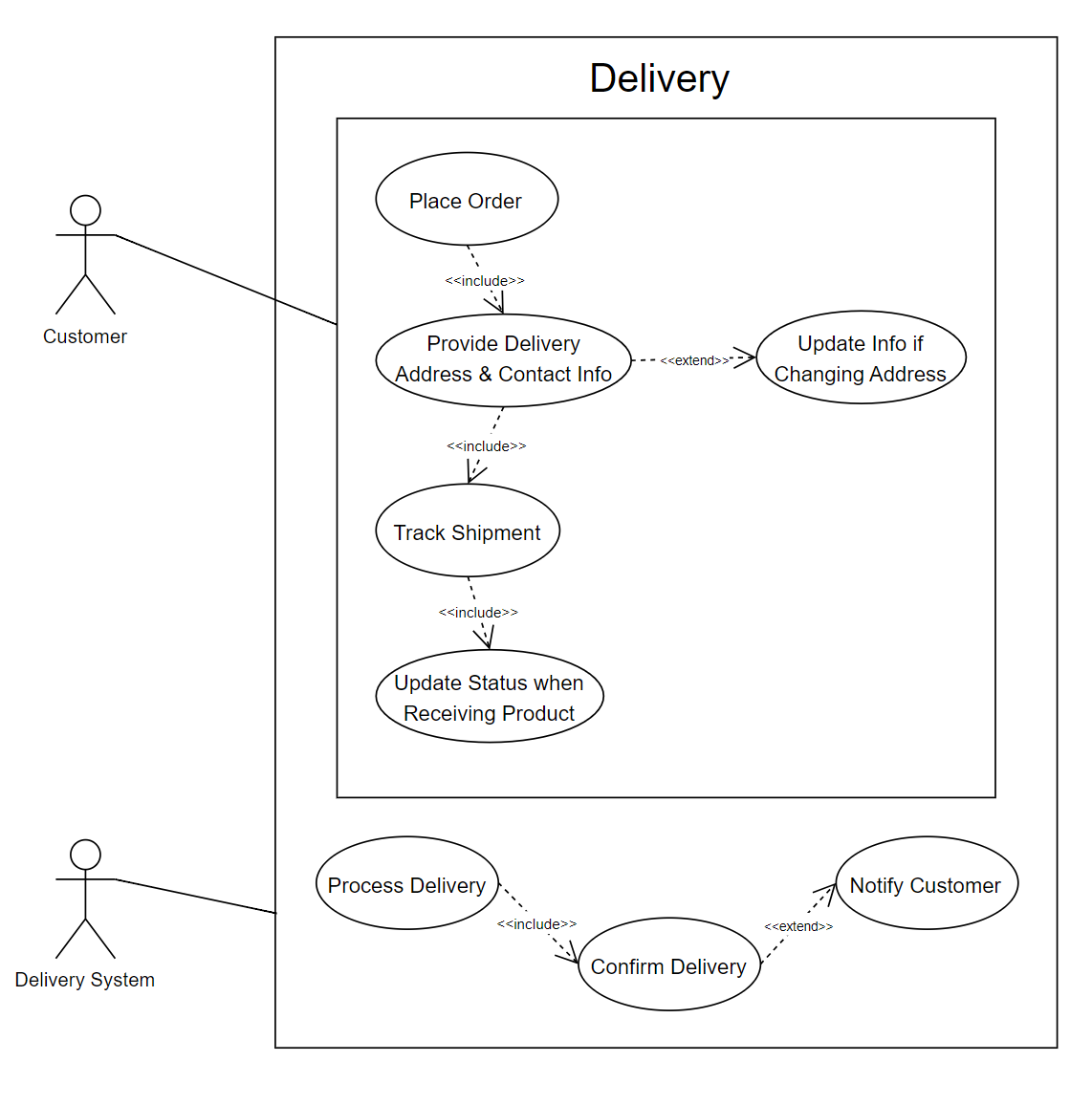
1. Feedback
   1. UML Diagram



* 1. Table Description

|  |  |
| --- | --- |
| Name | Give feedback |
| Scenario | Customer users want to provide feedback after purchasing items. |
| Descriptions | This use case occurs when the users want to give evaluations about products they have bought. A feedback form appears, where the users are asked to rate satisfaction level and provided additional comments rely on that rating. The system receives the feedback submission and stores it for enhancing products. |
| Actor | Customer |
| Trigger | Customers purchase products and decide to share their opinion based on their experiences. |
| Related use cases | Purchase product, submit feedback. |
| Pre-condition | * The customers have purchased products before. * The customers log in the system’s website and access the Feedback function of those products. |
| Post-condition | * The system already receives the Feedback form and notify for customers the feedback is successfully sent. * The system stores feedback from multiple users to improve its processes or communicate with suppliers about products. |
| Flow of Events | 1. The customer logs into the platform. 2. The customer decides to share opinions after experiencing products, they navigate to the list of delivered products and select Feedback options. 3. The system displays Feedback form including satisfaction level and asks customers to rate, then they need to provide additional comments. 4. The customer submits the feedback. 5. The system confirms receipt of the feedback and stores it in a database for analysis. 6. Over time, the system uses many feedbacks from customers to improve processes or discuss with suppliers about products. |
| Exception conditions | * No purchase history: The system prevents the action and informs customers that feedback is only for purchased products. * Incomplete feedback: If the user provides sufficient information (e.g no rating or comment) |

1. Delivery:
   1. UML Diagram

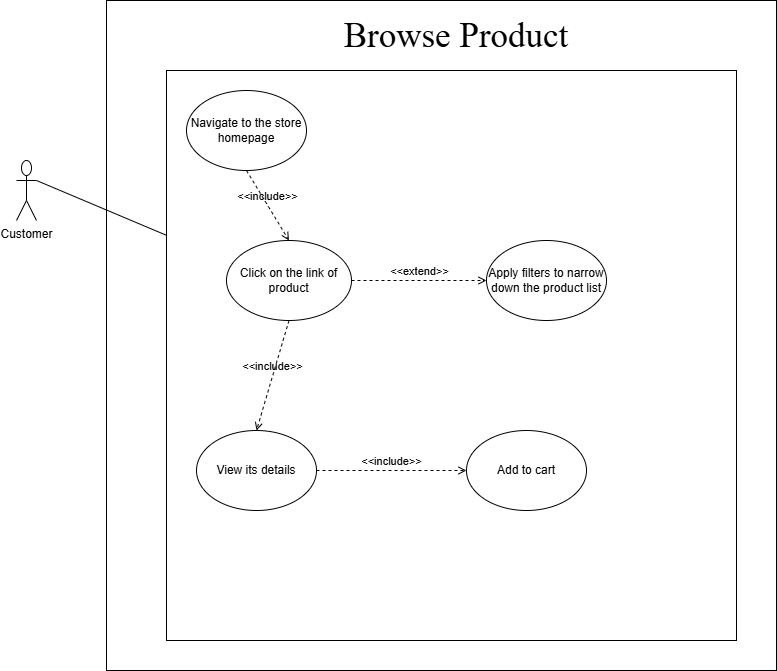


* 1. Table Description

|  |  |
| --- | --- |
| Name | Show delivery |
| Scenario | When a user places an order for a product, they have to choose a delivery method to ensure the product is shipped to their address. |
| Description | This use case handles the logistics of delivering products purchased by users, including order processing, shipment tracking, and notifying users about their delivery status. |
| Actor | Customer, delivery system |
| Trigger | The customer has completed the process of purchasing products and is requested delivery to their specified address. |
| Related use cases | Place order, track shipment, update delivery address, confirm delivery |
| Pre-condition | * The customer has successfully purchased a product. * The customer has provided a valid delivery address. |
| Post-condition | * The customer is notified of the delivery status. * The order status is updated in the system. * The product is successfully delivered to the customer. |
| Flow of Events | 1. The customer completes the purchase and selects the delivery option. 2. The system verifies the delivery address and contact information provided by the customer. 3. The system calculates the estimated delivery time and shipping costs. 4. The customer confirms the order details, including delivery options. 5. The system processes the order and notifies the user of the order confirmation. 6. The Delivery System prepares the product for shipment. 7. The system generates a tracking number and provides it to the customer. 8. The customer can track the shipment through the system. 9. Once the product is delivered, the customer receives a notification confirming delivery. 10. The order status is updated to “Delivered” in the system. |
| Exception conditions | * Invalid Delivery Address or Contact Information: the system prompts the user to enter a valid one. * Payment Issues: the system notifies the customer and does not proceed with delivery. * Shipping Delays: (due to unforeseen circumstances, such as bad weather, logistics issues), the system notifies the customer and provides an updated estimated delivery time. * Delivery Failure: (due to recipient not available) The system attempts a redelivery or notifies the user to reschedule. |

8. Browse Product

a. UML Diagram

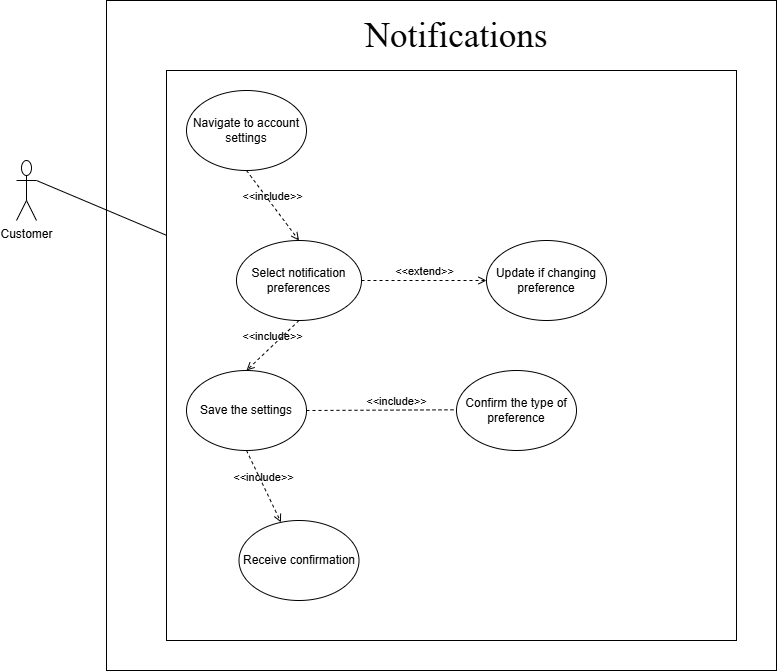


b. Table Description

|  |  |
| --- | --- |
| Name | Browse Product |
| Scenario | Users want to search and view available products in the store. |
| Description | This use case describes how users can navigate the product catalog, filter products, and view details of each product. |
| Actor | Customers |
| Trigger | The customer accesses the store's homepage and clicks on the "Browse Products" link. |
| Related use cases | None |
| Pre-condition | The user has successfully accessed the store's homepage. |
| Post-condition | Success: Users can view product details and add items to their cart. Fail: The system displays an error if there are no products available. |
| Flow of events | 1. User navigates to the store homepage.  2. User clicks on the "Browse Products" link.  3. The system displays the product catalog.  4. User can apply filters to narrow down the product list.  5. User selects a product to view its details. |
| Exception conditions | If no products are found, the system displays a message indicating the lack of products. |

9. Notifications

a. UML Diagram

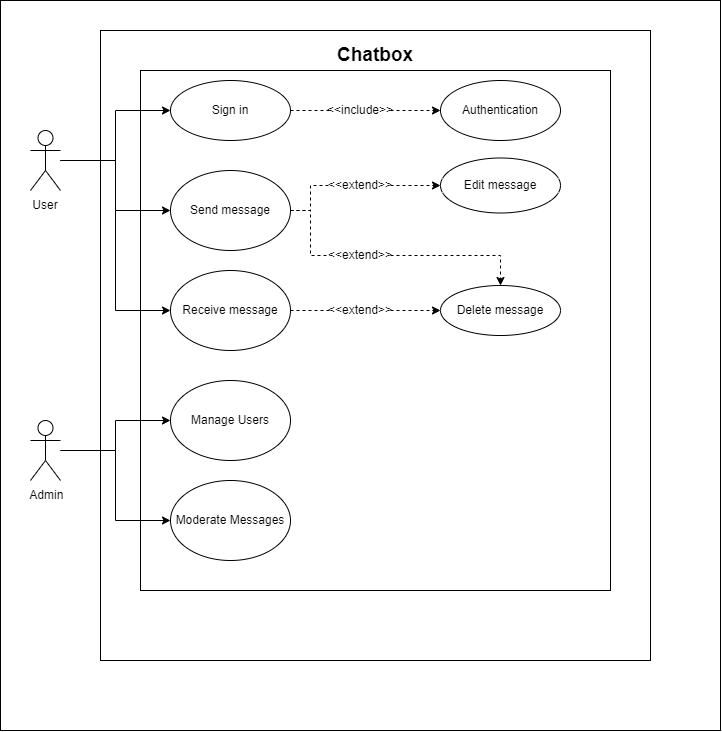


b. Table Description

|  |  |
| --- | --- |
| Name | Check notifications |
| Scenario | Users want to receive updates about their orders and promotions. |
| Description | This use case describes how users can opt-in to receive notifications via email or app regarding order updates and special promotions. |
| Actor | Customers |
| Trigger | The customer chooses to enable notifications in their account settings. |
| Related use cases | None |
| Pre-condition | The user must have an account and be signed in to enable notifications. |
| Post-condition | **Success:** Users receive notifications as per their preferences.  **Fail:** The system informs users of any issues in enabling notifications. |
| Flow of events | 1. User navigates to account settings.  2. User selects notification preferences.  3. The system saves the user's preferences.  4. The system confirms that notifications are enabled. |
| Exception conditions | If the system fails to save preferences, it notifies the user of the error. |

10. Chatbox

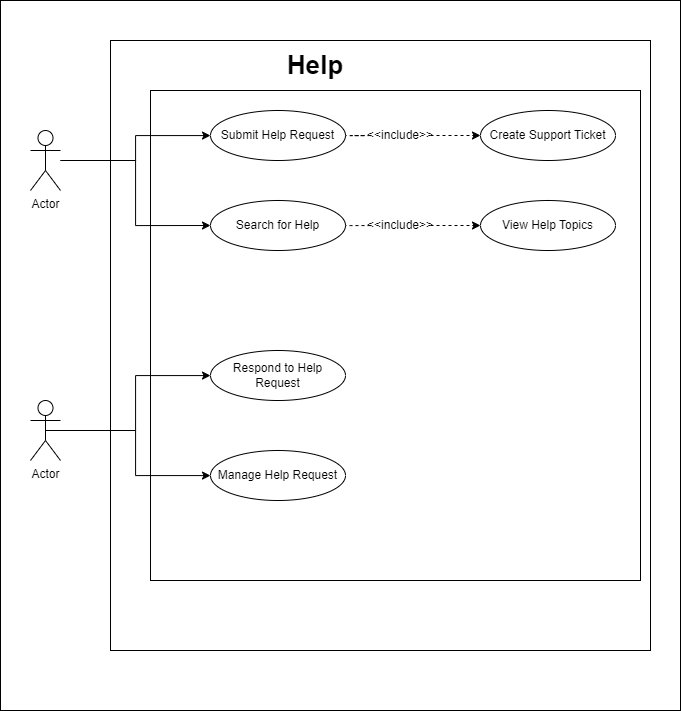
1. UML



1. Description

|  |  |
| --- | --- |
| Name | Show chatbot |
| Scenario | User-to-User Communication |
| Description | This use case represents the process where a user signs in to a chat system, sends, receives, edits, or deletes messages. Admin users manage user accounts and moderate chat messages. |
| Actor | User, Admin |
| Trigger | User opens the chat interface or Admin manages the chat system. |
| Related use cases | Sign In, Authentication & Authorization, Manage Users, Moderate Messages |
| Pre-condition | - The system is online.  - User has an active account.  - Admin has the necessary permissions. |
| Post-condition | - Messages are successfully sent, received, edited, or deleted.  - Admin moderates or manages users. |
| Flow of events | 1. User signs in to the system.  2. User sends a message.  3. The message is received by the intended recipient.  4. User can edit or delete the message.  5. Admin can manage users and moderate messages if necessary. |
| Exception conditions | 1. If the user is not authenticated, they cannot send or receive messages.  2. If the message fails to send, the user is notified of the error.  3. Admin fails to moderate due to permissions error. |

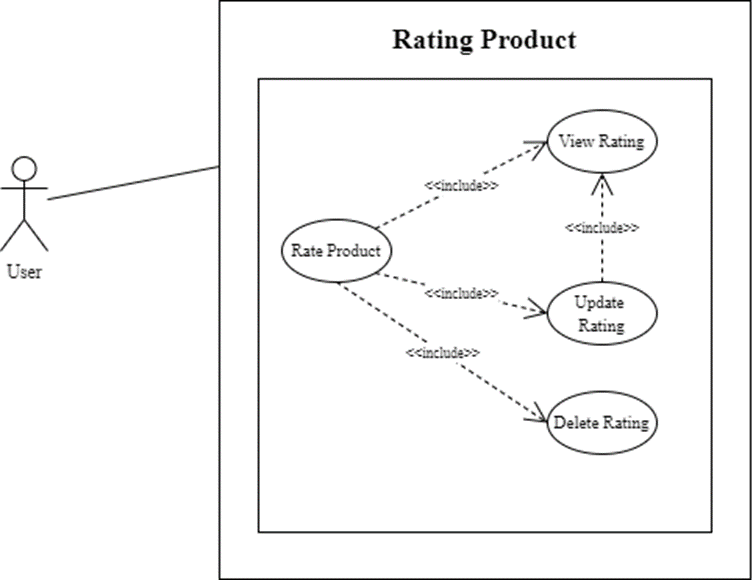
1. Help
2. UML



1. Table Description

|  |  |
| --- | --- |
| Name | Show help |
| Scenario | User seeking help or Admin managing help requests and topics |
| Description | This use case represents the interaction where a user seeks help through viewing topics, searching for specific help, or submitting a help request. Admins respond to requests and manage the help topics. |
| Actor | User, Admin |
| Trigger | User needs assistance or Admin managing help system functionality |
| Related use cases | View Help Topics, Search for Help, Submit Help Request, Respond to Help Request, Manage Help Topics |
| Pre-condition | - The user is logged in.  - Help topics are available.  - Admin has appropriate permissions. |
| Post-condition | - Help topics are displayed.  - User help requests are submitted or responded to.  - Admin manages topics. |
| Flow of events | 1. User signs in to the system.  2. User views help topics or searches for specific help.  3. If needed, the user submits a help request.  4. Admin reviews and responds to the request.  5. Admin updates or manages help topics if necessary. |
| Exception conditions | 1. If no help topics are available, an error message is displayed.  2. If the help request fails, the user is notified.  3. Admin fails to respond due to permission errors. |

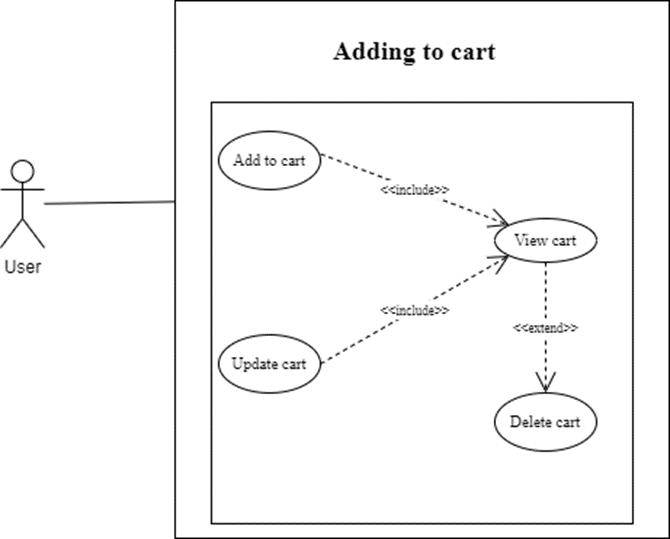
1. Rating product
2. UML



1. Table Description

|  |  |
| --- | --- |
| Name | Rate Product |
| Scenario | The user rates a product |
| Description | The user can provide a rating for the product, view, update, or delete their rating |
| Actor | User |
| Trigger | |  | | --- | | The user selects a product and wants to rate it |  |  | | --- | |  | |
| |  | | --- | | Related Use Cases |  |  | | --- | |  | | |  | | --- | | View Rating, Update Rating, Delete Rating |  |  | | --- | |  | |
| Pre-condition | |  | | --- | | The user is logged into the system and has permission to rate the product |  |  | | --- | |  | |
| |  | | --- | | Post-condition |  |  | | --- | |  | | |  | | --- | | The rating is stored in the system and can be viewed, updated, or deleted later |  |  | | --- | |  | |
| Flow of Events | 1. The user selects the product. 2. The user enters a rating (star rating and comment) 3. The system confirms and saves the rating. |
| Exception Conditions | - If the user does not enter a new rating, display an error message |

1. Add to cart
2. UML



1. Table Description

|  |  |
| --- | --- |
| Name | Add to Cart |
| Scenario | User adds a product to their shopping cart |
| Description | The user selects a product and adds it to their cart. The system checks product availability and updates the cart accordingly. |
| Actor | User |
| Trigger | |  | | --- | | User selects the option to add a product to the cart |  |  | | --- | |  | |
| |  | | --- | | Related Use Cases |  |  | | --- | |  | | |  | | --- | | View Cart, Update Cart, Delete Cart, |  |  | | --- | |  | |
| Pre-condition | |  | | --- | | The product must be available in stock; the user must be browsing the product list or detail page. |  |  | | --- | |  | |
| |  | | --- | | Post-condition |  |  | | --- | |  | | |  | | --- | | The product is successfully added to the cart, and the cart's total is updated. |  |  | | --- | |  | |
| Flow of Events | 1. The user selects a product.  2. The user clicks "Add to Cart."  3. Added to the cart.  4. The cart is updated and displayed to the user. |
| Exception Conditions | 1. Product is out of stock (system displays an error message).  2. Cart cannot be updated due to a system error (error message displayed). |