

Project Title : Cookies Shop

Problem formulation :

- **Problem:**
- The absence of a user-friendly e-commerce platform for cookie sales hampers business growth. Current solutions lack efficiency, risking suboptimal user experiences, inadequate product management, and potential security issues.
- **Solution:**
- Develop a robust e-commerce platform for cookie sales, prioritizing user and administrator satisfaction. The platform will excel in product management, user security, and communication, setting a new standard in the online cookie market.
- **Objectives:**
- **User Experience Excellence:**
 - Intuitive design for easy navigation.
 - Responsive interface for various devices.
- **Efficient Product Management:**
 - Simple product management for easy updates.
 - Categorization for product organization.
- **Communication Facilitation:**
 - User-friendly feedback system.
 - customer support.
- **Scalability:**
 - Handle growth in products and users.

Stakeholders

- Admin
- User

System Architecture

- **Microservices Architecture:** Utilizing Spring Boot for backend services.
- **Frontend:** HTML, CSS, JavaScript, Bootstrap, jQuery.
- **Backend:** Spring Boot.
- **Database:** MySQL.
- **Service Discovery:** Eureka Server.

Component Design

- **Frontend Components:** HTML templates, CSS stylesheets, JavaScript scripts.
- **Backend Components:** Spring Boot microservices for user management, product management, order management, etc.
- **Database Schema:** Tables for users, products, orders, etc.

Security Design

- **Authentication and authorization mechanisms:** using Spring Security and JWT packages.
- **Secure communication:** between frontend and backend.

Communication Protocols

- **Communication protocols used between frontend and backend components:** RESTful APIs.
- **Description of data formats exchanged between client and server :** JSON.

Logging Mechanisms

- **Implementation of logging mechanisms using Aspect-Oriented Programming (AOP):** to track system activities, such as user interactions, API calls, database operations, and error handling.
- **Integration with logging frameworks:** to log messages with different severity levels and appenders for writing logs to the terminal.

Requirements :

- **Functional requirements :**
- **Admin :**
- **Product Management:**
 - 1- Add new cookies to the inventory with details.
 - 2- Edit/update existing product information.
 - 3- Remove products.
- **User Management:**
 - 1- View and manage user accounts.
 - 2- Remove accounts if necessary.
 - 3- Reset user passwords.
- **Order Management:**
 - 1- View and process incoming orders.
- **Feedback Management:**

- 1- Access and review customer feedback messages.
- **User :**
- **User Registration and Authentication:**
 - 1-Allow users to create/delete accounts and log in/log out.
 - 2-Update personal profile information (name, shipping address...).
- **Shopping Cart and Check-out:**
 - 1-Add cookies to a shopping cart and adjust quantities.
 - 2-Calculate order totals, including shipping costs.
 - 3-Proceed through a checkout process with multiple payment options.
 - 4-Apply discounts and promotions at checkout.
 - 5-Select the preferred shipping method and delivery address.
- **Wishlist:**
 - 1-Users can create and manage Wishlist.
 - 2-User can transfer items from Wishlist to cart.
 - 2-Users can stop notifications whenever they want.
- **Order History:**
 - 1-View order history and track the status of current orders(shipping updates, and delivery information).
- **Product Reviews and Ratings:**
 - 1-Leave reviews and ratings for purchased cookie products.
 - 2-send messages for inquiries, issues, or feedback.
- **Non-Functional Requirements:**
- **Performance:**
 - 1-The website should be able to handle a high volume of traffic.
 - 2-The website should be able to process transactions quickly.
 - 3-The website should be able to load pages quickly.
- **Security:**
 - 1-The website should be secure from hackers.
 - 2-The website should protect customer data.
 - 3-The website should use secure payment processing methods.
 - 4-The website should protect user reviews from unauthorized access.
- **Usability:**
 - 1-The website should be easy to use.
 - 2-The website should provide a clear and concise product description.
 - 3-The website should provide a way to easily leave reviews.
- **Reliability:**
 - 1-The website should be reliable and available 24/7.
 - 2-The website should be able to recover from failures quickly.
 - 3-The website should be able to handle a large volume of user reviews.

4-The website should be able to recover from data loss quickly.

- **Scalability:**

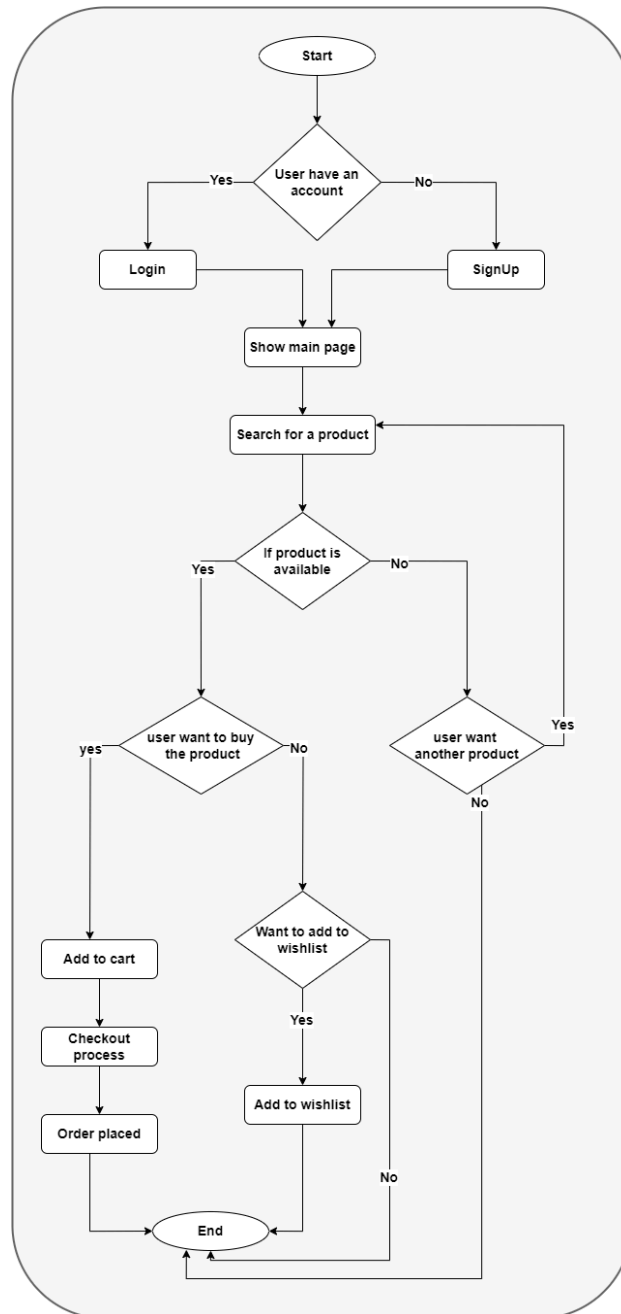
1-The website should be able to accommodate a growing product catalog.

2-The website should be able to handle increasing user review volumes.

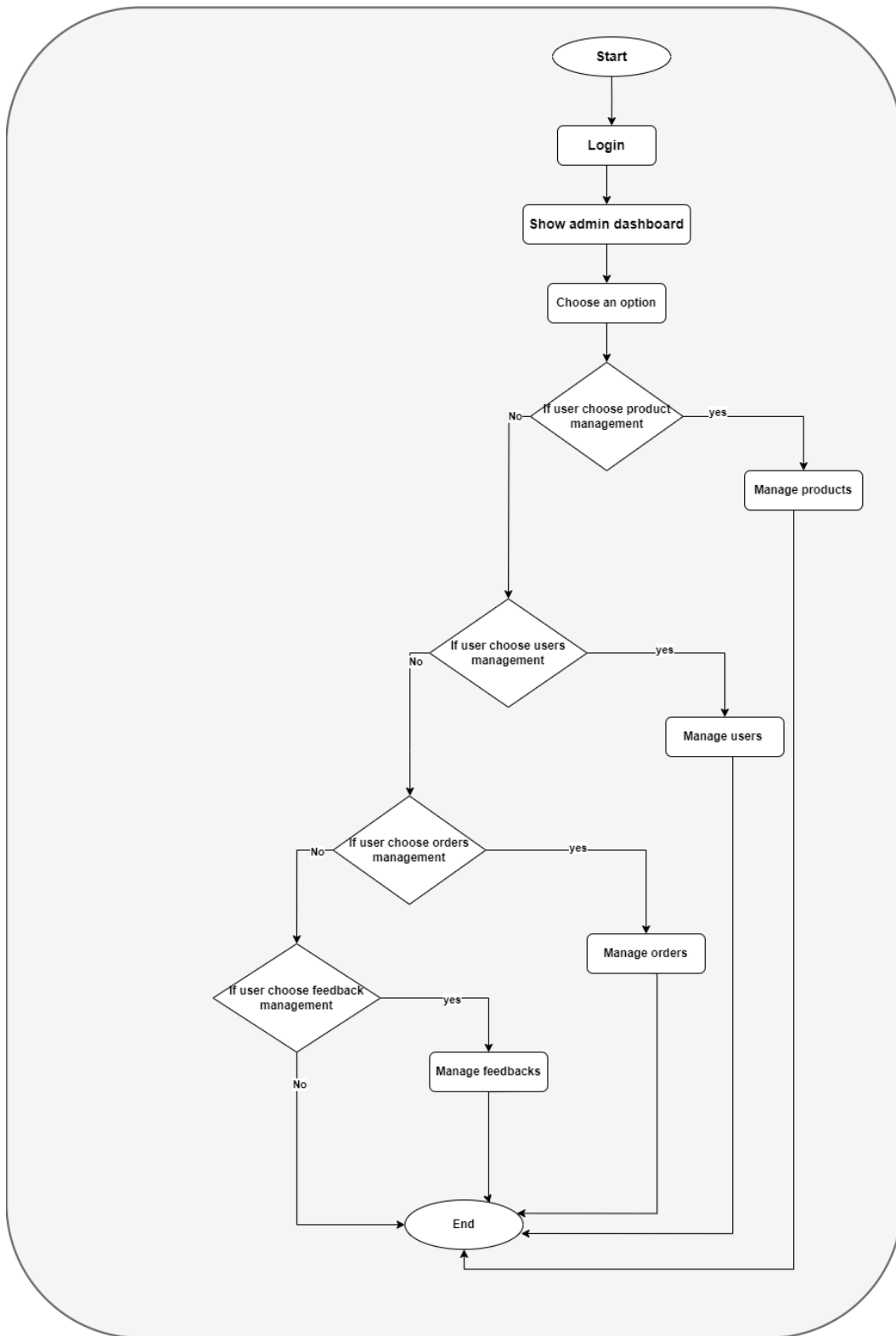
3-The website should be able to handle increasing traffic volumes.

- **Activity diagrams :**

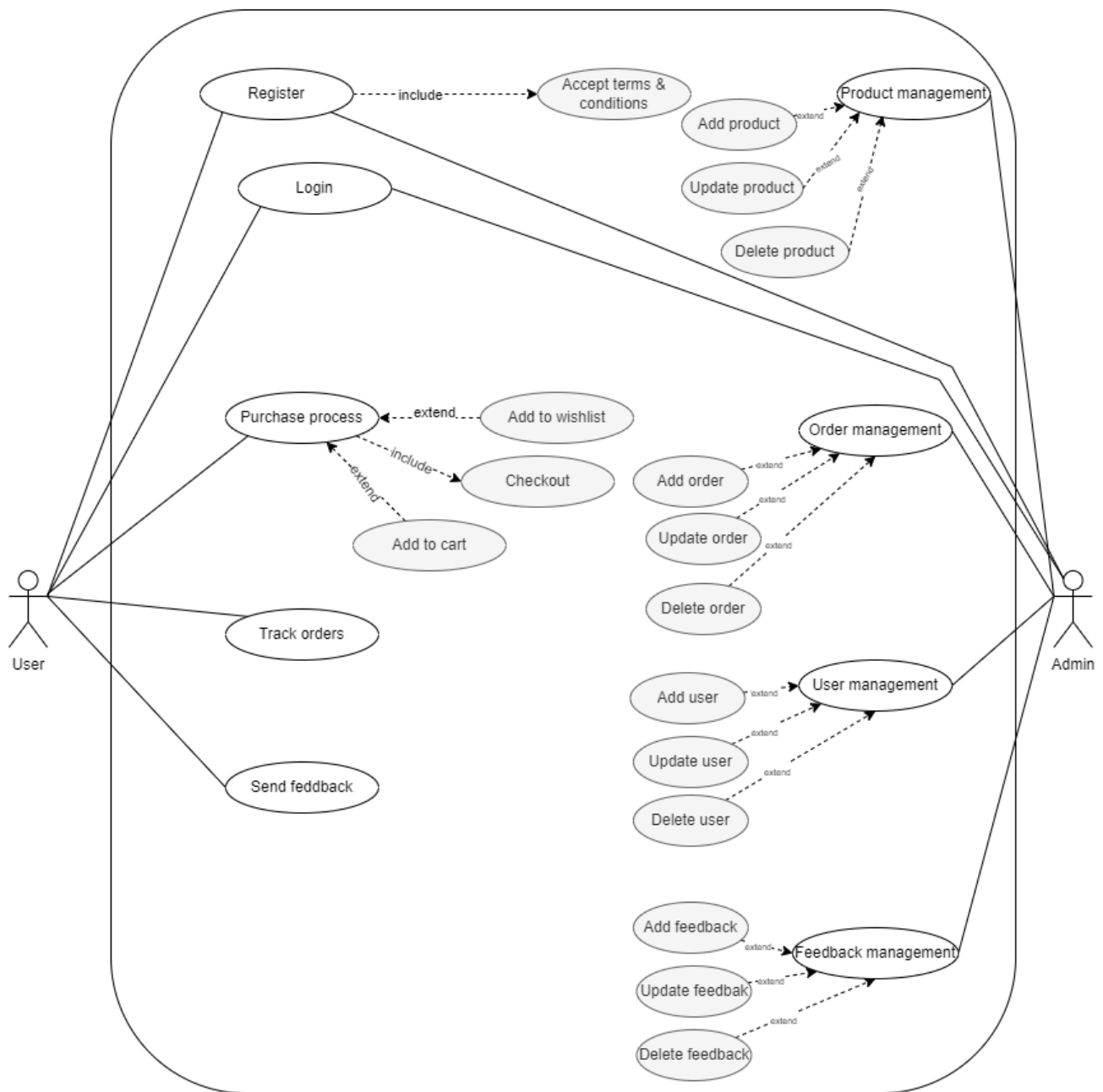
- **Customer :**



- Admin :



-Use case diagram:



-Use case descriptions:

1-Admin login

Use case ID	UC-01
Use case name	Admin Log In
Preconditions:	-Admin credentials (username and password) are valid and registered in the system.
Postconditions:	-Admin is successfully authenticated and gains access to the admin dashboard. - In case of unsuccessful login attempts, the system provides appropriate error messages.
Actor:	Admin
Trigger:	-Admin navigates to the login page and enters valid credentials.
Goal:	-To authenticate the admin and grant access to the admin dashboard.
Standard Process (Main Success Scenario):	<ul style="list-style-type: none">- Admin navigates to the login page.- The system presents a login form requesting the admin's username and password.-Admin enters valid credentials.-System verifies the credentials against the stored data. <p>If the credentials are valid:</p> <ul style="list-style-type: none">a. System grants access to the admin dashboard.b. Admin gains the ability to perform admin-level functions (product management, user management, order management, feedback management).c. System logs the successful login for security purposes. <p>If the credentials are invalid:</p> <ul style="list-style-type: none">a. System displays an error message indicating the login failure.b. Admin is prompted to re-enter the credentials.

2- Product Management

Use case ID	UC-02
Use case name	Admin Product Management
Preconditions:	<ul style="list-style-type: none">-Admin is logged into the system.-The admin has the necessary permissions for product management.
Postconditions:	<ul style="list-style-type: none">-Product information is successfully updated in the system, changes are reflected in the user interface.
Actor:	Admin
Trigger:	<ul style="list-style-type: none">-Admin selects the "Product Management" option from the admin dashboard.
Goal:	<ul style="list-style-type: none">-To manage and update product information in the system.
Standard Process (Main Success Scenario):	<ul style="list-style-type: none">-Admin logs into the system using valid credentials.-Admin navigates to the admin dashboard.-Admin selects the "Product Management" option.-System displays a list of existing products.-Admin selects a specific product to manage.-System presents options for product management, including:<ul style="list-style-type: none">a. View product details.b. Edit product information (name, price, description, etc.).c. Add new products to the inventory.d. Delete existing products.- Admin chooses an action and provides the necessary details.-System updates the product information.- Changes are reflected in the product list.

3- User management

Use case ID	UC-03
Use case name	Admin Users Management
Preconditions:	<ul style="list-style-type: none">-Admin is logged into the system.-The admin has the necessary permissions for user management
Postconditions:	<ul style="list-style-type: none">-User information is successfully updated in the system, and Changes are reflected in the user interface.
Actor:	Admin
Trigger:	<ul style="list-style-type: none">-Admin selects the "User Management" option from the admin dashboard.
Goal:	<ul style="list-style-type: none">-To manage and update user information in the system.
Standard Process (Main Success Scenario):	<ul style="list-style-type: none">-Admin logs into the system using valid credentials.-Admin navigates to the admin dashboard.-Admin selects the "User Management" option.-System displays a list of existing users.-Admin selects a specific user to manage.-System presents options for user management, including:<ul style="list-style-type: none">a. View user details.b. Edit user information (name, email, etc.).c. Add new users to the system.d. Deactivate or delete existing user accounts.Admin chooses an action and provides the necessary details.The system updates the user information.Changes are reflected in the user list.

4- Orders management

Use case ID	UC-04
Use case name	Admin Orders Management
Preconditions:	<ul style="list-style-type: none">-Admin is logged into the system.-The admin has the necessary permissions for order management
Postconditions:	<ul style="list-style-type: none">-Order information is successfully updated in the system.-Changes are reflected in the order status and relevant user accounts.
Actor:	Admin
Trigger:	-Admin selects the "Order Management" option from the admin dashboard.
Goal:	-To manage and update order information in the system.
Standard Process (Main Success Scenario):	<ul style="list-style-type: none">-Admin logs into the system using valid credentials.-Admin navigates to the admin dashboard.-Admin selects the "Order Management" option.-System displays a list of existing orders, including order details (order number, total, etc.).-Admin selects a specific order to manage.- System presents options for order management, including:<ul style="list-style-type: none">a. View detailed order information.b. Update order status (e.g., processing, shipped, delivered, canceled).c. Add or remove products from the order.Admin chooses an action and provides the necessary details.The system updates the order information.Changes are reflected in the order list and relevant user accounts.

5-Feedback management

Use case ID	UC-05
Use case name	Admin Feedback Management
Preconditions:	<ul style="list-style-type: none">-Admin is logged into the system.-The admin has the necessary permissions for feedback management.
Postconditions:	<ul style="list-style-type: none">-Feedback information is successfully processed and may influence improvements in the system.
Actor:	Admin
Trigger:	<ul style="list-style-type: none">-Admin selects the "Feedback Management" option from the admin dashboard.
Goal:	<ul style="list-style-type: none">-To manage feedback provided on the website.
Standard Process (Main Success Scenario):	<ul style="list-style-type: none">-Admin logs into the system using valid credentials.-Admin navigates to the admin dashboard.-Admin selects the "Feedback Management" option.-System displays a list of user feedback, including details such as the user's name, date, and feedback content.-Admin selects a specific feedback entry to manage.-System presents options for feedback management, including:<ul style="list-style-type: none">a. View detailed feedback content.b. Categorize the feedback (e.g., positive, negative, suggestion).Admin chooses an action and provides the necessary details.If necessary, the admin communicates with the user for clarification or additional information.Changes are reflected in the feedback list.

6-User registration

Use case ID	UC-06
Use case name	User Registration
Preconditions:	<ul style="list-style-type: none">-The user is accessing the website for the first time.-The user has a valid email address.
Postconditions:	<ul style="list-style-type: none">-User account is successfully created.-User is logged into the system.
Actor:	User
Trigger:	-User clicks on the "Sign Up" button.
Goal:	-To create a new user account on the website.
Standard Process (Main Success Scenario):	<ul style="list-style-type: none">-User navigates to the registration page.-System presents a registration form, requesting information such as name, email, password, etc.-User enters valid registration details.-System validates the entered information.- If validation is successful:<ul style="list-style-type: none">a. System creates a new user account. <p>The user account is activated, and the user is redirected to the login page.</p> <p>The user logs in with the newly created credentials.</p>

7-User login

Use case ID	UC-07
Use case name	User Login
Preconditions:	<p>The user account is successfully registered.</p> <p>The user has valid login credentials.</p>
Postconditions:	<p>The user is successfully authenticated and logged into the system.</p> <p>-User gains access to personalized features.</p>
Actor:	User
Trigger:	-User clicks on the "Log In" button.
Goal:	-To access the user account on the website.
Standard Process (Main Success Scenario):	<p>-User navigates to the login page.</p> <p>-System presents a login form, requesting the user's email and password.</p> <p>-User enters valid login credentials.</p> <p>-System verifies the credentials against the stored data.</p> <p>If the credentials are valid:</p> <ol style="list-style-type: none">System logs the user into the system.User gains access to personalized features (cart, wish list, etc.).User is redirected to the main user page (cookies page). <p>If the credentials are invalid:</p> <ol style="list-style-type: none">System displays an error message indicating the login failure.The user is prompted to re-enter the credentials.

8-User feedback

Use case ID	UC-08
Use case name	User Feedback
Preconditions:	The user is logged into the website. -The user is on the "Contact Us" page.
Postconditions:	-Feedback is successfully submitted through the contact form. -The feedback is shown in the admin dashboard in the feedback management part.
Actor:	User
Trigger:	-User clicks on the "Contact Us" link or button.
Goal:	-To provide feedback or inquiries through the website's contact form.
Standard Process (Main Success Scenario):	-User navigates to the "Contact Us" page. -System presents a contact form with fields such as name, email, and message. -User fills in the required information and provides feedback in the message field. -User clicks on the "Submit" or "Send" button. -System validates the entered information. If validation is successful: a. System records the feedback in the admin dashboard.

10-User add product to the cart

Use case ID	UC-10
Use case name	User Add Product to Cart
Preconditions:	<ul style="list-style-type: none">-User is logged into the website.-User is on the product details page or a page displaying the product.
Postconditions:	<ul style="list-style-type: none">-Product is successfully added to the user's cart.-User has the option to proceed to checkout.
Actor:	User
Trigger:	-User clicks the "Add to Cart" button on the products page.
Goal:	-To add a specific product to the user's shopping cart.
Standard Process (Main Success Scenario):	<ul style="list-style-type: none">-User navigates to the product details page or a page displaying the product.-User clicks the "Add to Cart" button.-System adds the selected product to the user's shopping cart.-System updates the cart icon to reflect the added product.-Optionally, User can continue shopping or proceed to the cart for checkout.

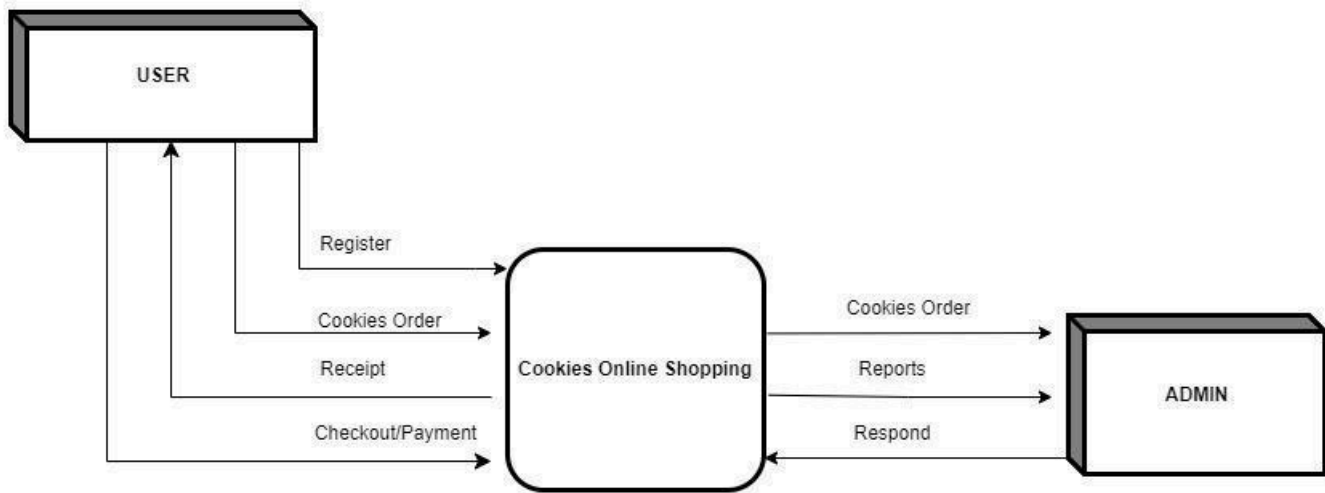
11-User add product to the Wishlist

Use case ID	UC-11
Use case name	User Add Product to Wishlist
Preconditions:	<ul style="list-style-type: none">-User is logged into the website.-User is on the product details page or a page displaying the product.
Postconditions:	<ul style="list-style-type: none">-Product is successfully added to the user's Wishlist.-User can view and manage products in their Wishlist.
Actor:	User
Trigger:	<ul style="list-style-type: none">-User clicks the "Add to Wishlist" button on the products page.
Goal:	<ul style="list-style-type: none">-To add a specific product to the user's Wishlist for future reference.
Standard Process (Main Success Scenario):	<ul style="list-style-type: none">-User navigates to the product details page or a page displaying the product.-User clicks the "Add to Wishlist" button.-System adds the selected product to the user's Wishlist.-Optionally, user can continue shopping or view their Wishlist for later reference.

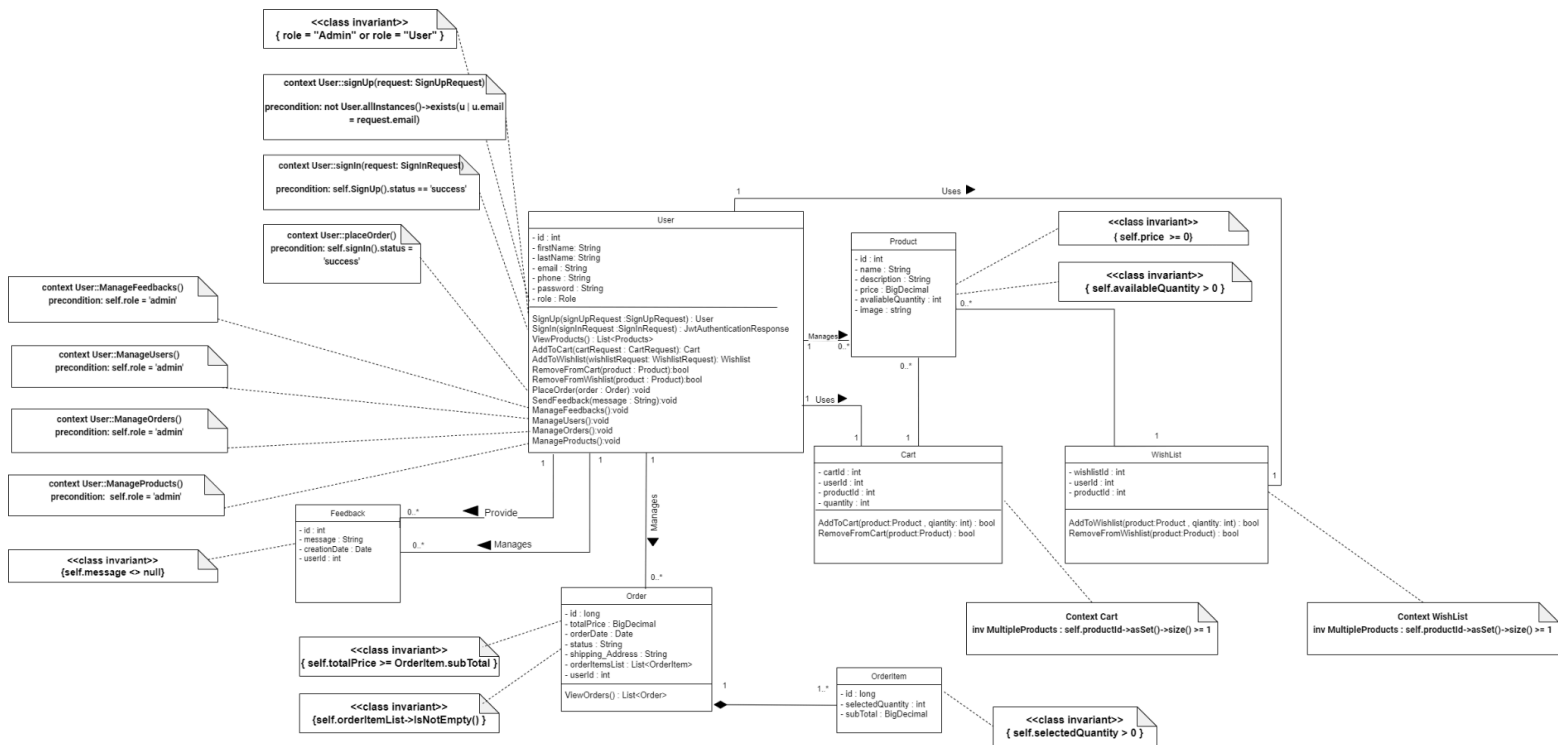
12-User track the order

Use case ID	UC-12
Use case name	User Track the Order
Preconditions:	<ul style="list-style-type: none">-User is logged into the website.-User has placed an order recently.
Postconditions:	<ul style="list-style-type: none">-The user is provided with information on the order status.-User may receive notifications for significant order updates.
Actor:	User
Trigger:	<ul style="list-style-type: none">-User clicks on the "orders" option in the user account or order confirmation page.
Goal:	<ul style="list-style-type: none">-To check the status and details of a placed order.
Standard Process (Main Success Scenario):	<ul style="list-style-type: none">-Users logs into their account on the website.-User navigates to the "Order History" or "Track Order" or "My Orders" section.-System displays a list of the user's recent orders.-User selects the specific order they want to track. <p>System presents order information, including status, estimated delivery date.</p>

- Data flow diagrams:
- Context Diagram:



- Class Diagram



OCL :

1- Ensure that email is not exist before signup

context User::signup(request: SignUpRequest)

precondition: not User.allInstances()->exists(u | u.email = request.email)

2- Ensure user have account before signing

context User::signIn(request: SignInRequest)

precondition: self.SignUp().status == 'success'

3-Ensure user have 'admin' role to manage products

context User::manageProducts()

precondition: self.role = 'admin'

4-Ensure user have 'admin' role to manage users

context User::manageUsers()

precondition: self.role = 'admin'

5-Ensure user have 'admin' role to manage feedbacks

context User::manageFeedbacks()

precondition: self.role = 'admin'

6-Ensure user have 'admin' role to manage orders

context User::manageOrders()

precondition: self.role = 'admin'

7-Ensure user is signedIn to place an order

context User::placeOrder()

precondition: self.signIn().status = 'success'

8-Ensure that user choose quantity > 0 when adding to cart

context User::addToCart(product: Product)

precondition: product.availableQuantity > 0

9-Ensure that user choose quantity > 0 when adding to wishlist

context User::addToWishlist(product: Product)

precondition: product.availableQuantity > 0

10-Ensure that user is signedIn to send feedback

context User::sendFeedback(message: String)

precondition: self.signIn().status == 'success'

11-Ensure that message in Feedback is not null

Context Feedback

inv FeedbackMessageNotNull : self.message <> null

11-Ensure that price in Product is not negative

Context Product

inv PriceNotNegative : self.price >= 0

12-Cart may contain multiple products

Context Cart

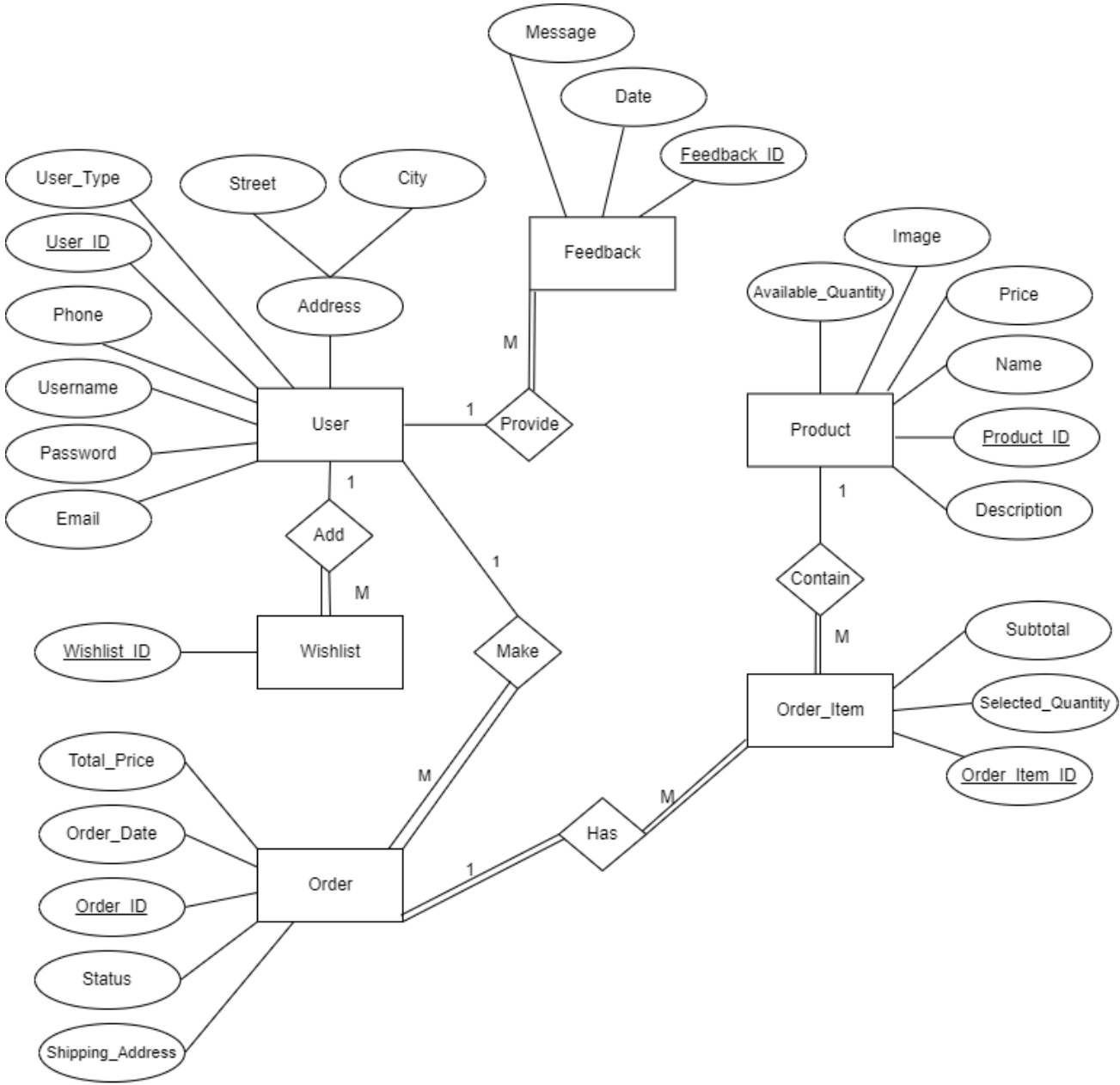
inv MultipleProducts : self.productId->asSet()->size() >= 1

13-Wishlist may contain multiple products

Context Wishlist

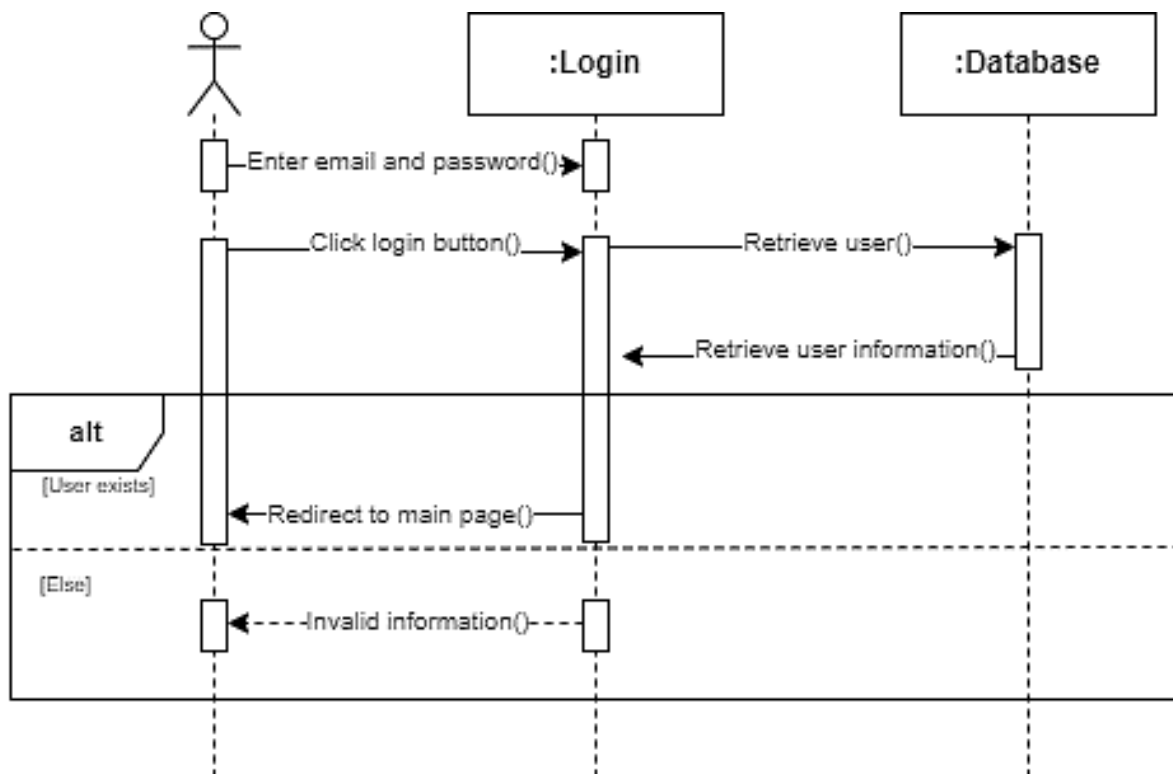
inv MultipleProducts : self.productId->asSet()->size() >= 1

ERD:

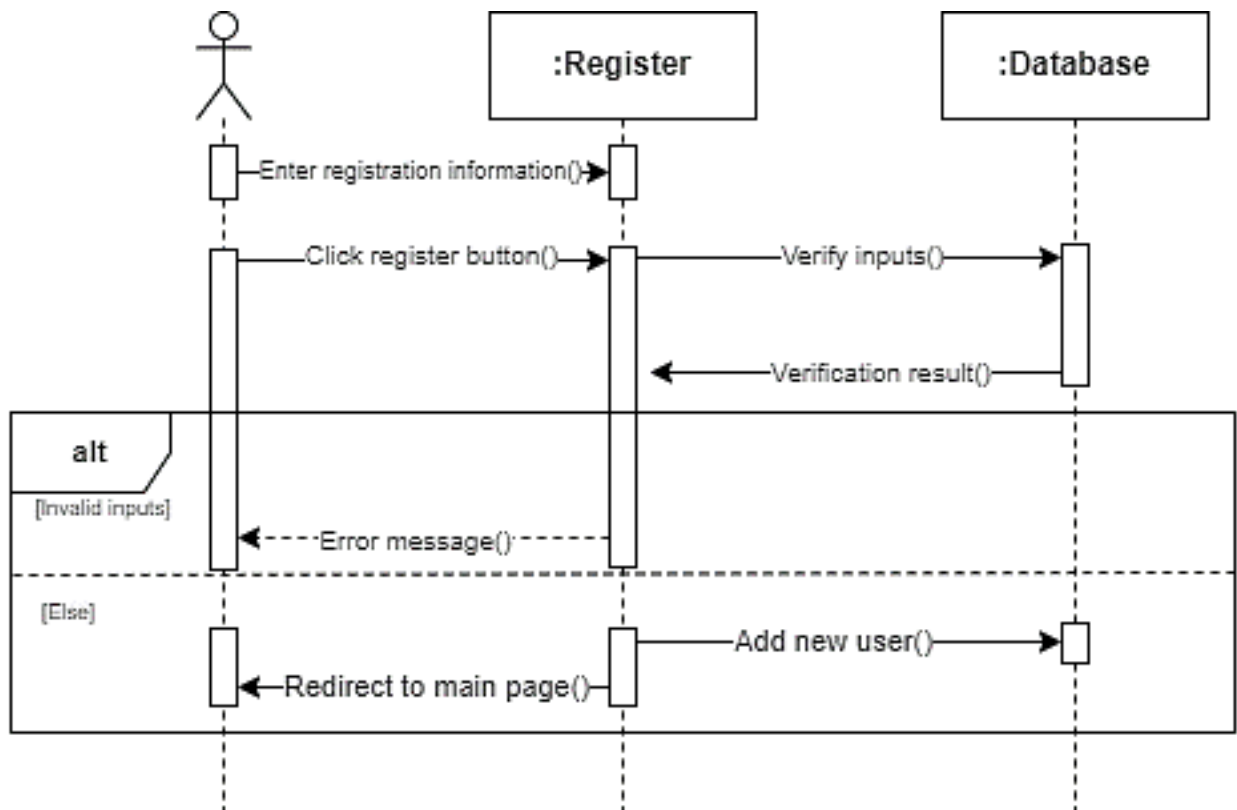


-Sequence diagrams:

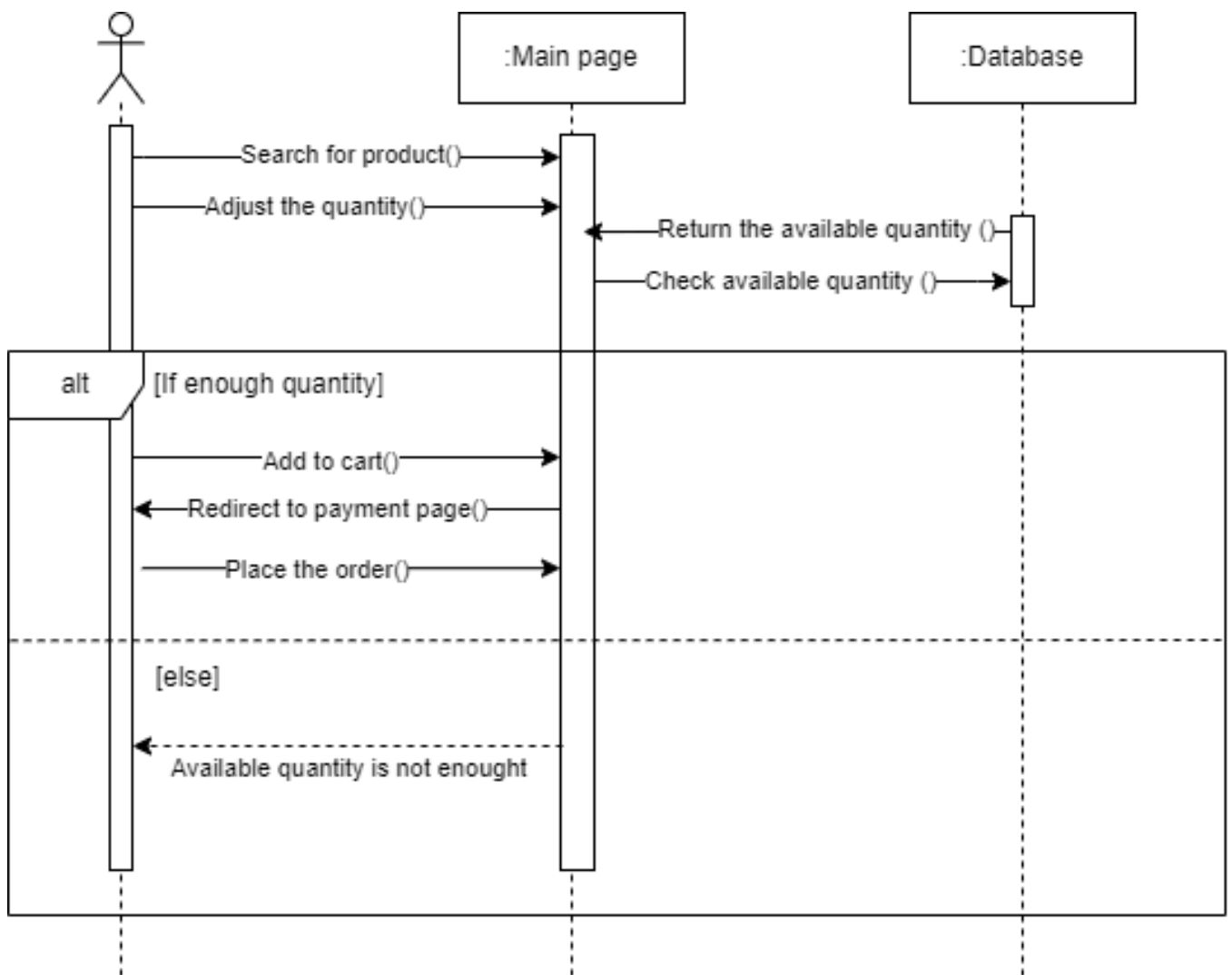
-User : login



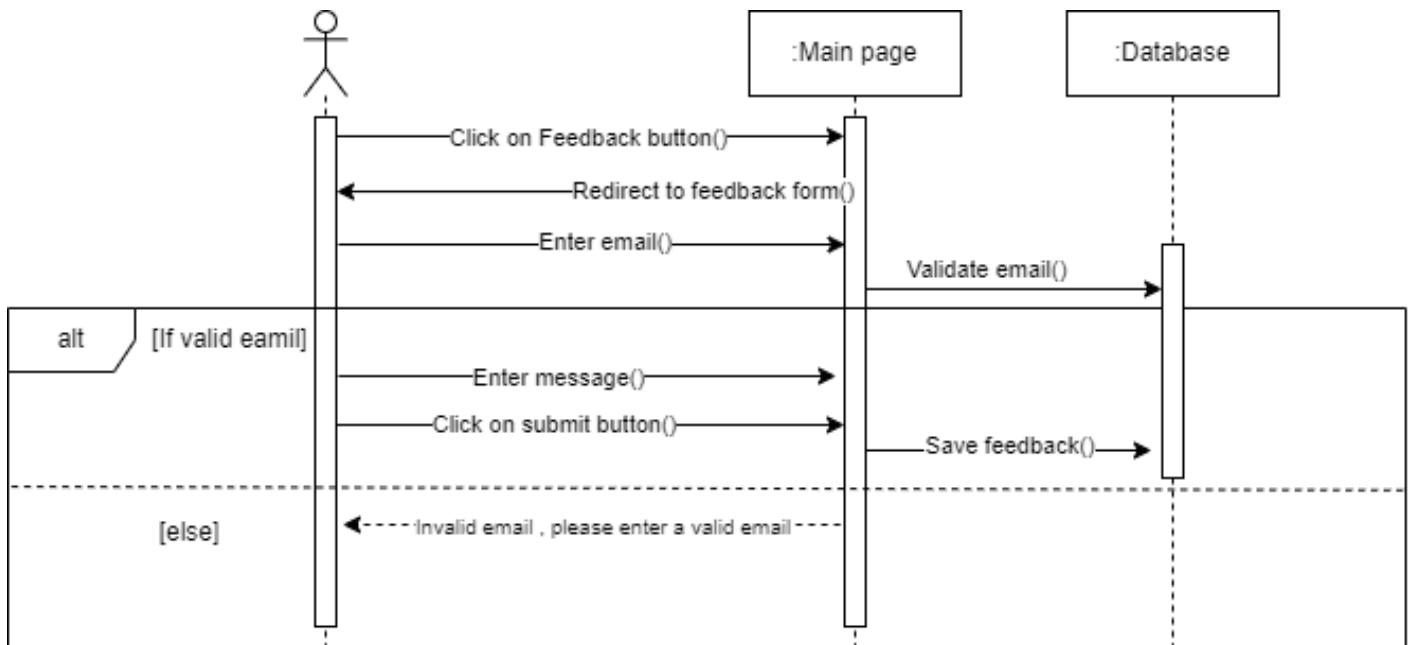
-Customer : registration



-Customer : Place an order



-customer : Send feedback



-Admin : Dashboard management

