
Software Requirements and Design Document

for

HawkTU-eCommerce

Prepared by Arqam Zia i22-1170 | Tabish Noman Khan i22-0831 | Danish Haroon i22-0955

HawkTU-eCommerce

Date Created : 26-11-24

Table of Contents

Table of Contents	ii
1. Introduction	1
1.1 Purpose	1
1.2 Product Scope	1
1.3 Title	1
1.4 Objectives	1
1.5 Problem Statement	1
2. Overall Description	1
2.1 Product Perspective	1
2.2 Product Functions	2
2.3 List of Use Cases	2
2.4 Extended Use Cases	2
2.5 Use Case Diagram	2
3. Other Nonfunctional Requirements	2
3.1 Performance Requirements	2
3.2 Safety Requirements	2
3.3 Security Requirements	2
3.4 Software Quality Attributes	2
3.5 Business Rules	3
3.6 Operating Environment	3
3.7 User Interfaces	3
4. Domain Model	3
5. System Sequence Diagram	3
6. Sequence Diagram	3
7. Class Diagram	4
8. Package Diagram	4
9. Deployment Diagram	4

1. Introduction

1.1 Purpose

The purpose of this document is to detail the software requirements and design for the HawkTU-eCommerce system. HawkTU is a logistics and delivery management platform that integrates e-commerce functionalities, providing businesses with a unified solution for managing order processing, inventory tracking, delivery scheduling, and real-time tracking. This document outlines the specifications for the system, ensuring alignment with the needs of its target industries.

1.2 Product Scope

HawkTU-eCommerce is a scalable and integrated platform for e-commerce, retail, manufacturing, and third-party logistics providers. It streamlines logistics operations and enhances e-commerce functionalities, allowing businesses to efficiently manage inventory, process orders, and track deliveries in real time. The solution aims to optimize operations, reduce operational costs, and improve customer satisfaction.

1.3 Title

HawkTU-eCommerce: An integrated logistics and e-commerce management platform designed to streamline order processing, inventory management, and delivery operations.

1.4 Objectives

Provide a comprehensive, scalable solution for logistics and e-commerce management.
Simplify workflows by integrating order processing, product management, and real-time tracking.
Enhance customer experience with transparent, real-time delivery tracking.
Ensure secure data management and interoperability with third-party services.
Deliver a unified platform with user-friendly desktop and web interfaces.

1.5 Problem Statement

Businesses face inefficiencies and increased costs when managing logistics and e-commerce through fragmented systems. HawkTU addresses these issues by providing an all-in-one platform that integrates key processes, minimizes errors, and enhances customer satisfaction through seamless and real-time tracking capabilities. The e-commerce functionalities further empower sellers to manage their operations centrally, optimizing the entire supply chain.

2. Overall Description

2.1 Product Perspective

HawkTU is a next-generation logistics and e-commerce platform designed to replace fragmented systems with an integrated, efficient solution. Built on a modular architecture with a Java Spring Boot backend and a React-based frontend, HawkTU supports secure and scalable operations. The platform is designed to interoperate with third-party services, including dropshipping and e-commerce APIs, ensuring maximum flexibility for diverse business needs.

2.2 Product Functions

Order Processing and Management: Enables businesses to process and update customer orders seamlessly.
Inventory Management: Provides real-time tools for managing stock levels, product details, and listings.
Real-Time Order Tracking: Offers customers live updates on delivery status through order IDs.
E-Commerce Seller Dashboard: Provides a user-friendly interface for sellers to manage products and orders.
Secure Data Handling: Ensures robust protection for business and customer data.

2.3 List of Use Cases

1. Manage Profile Information
2. Add Product Listing
3. Earn and Redeem Loyalty Points
4. Submit Product Review and Rating
5. Place Order
6. Modify or Cancel Order
7. Process Payment
8. Handle Refunds and Exchanges
9. Monitor Inventory Levels
10. Edit or Delete Product Listing
11. Track Orders by ID

2.4 Extended Use Cases

Manage Profile Information

Use Case Name	Manage Profile Information
Scope	HawkTU Logistics and Delivery Management System
Level	User Goal
Primary Actor	Customer/Seller
Stakeholders and Interests	<ul style="list-style-type: none"> • Customer/Seller: Wants to easily manage personal details and security settings. • System Administrator: Ensures data integrity and security of personal information.
Preconditions	<ul style="list-style-type: none"> • The user is logged into the system. • The system displays the profile management interface.
Post-conditions	<ul style="list-style-type: none"> • User profile is updated with the new information. • Changes are saved securely in the database.
Main Success Scenario	
Actor Action (or Intention)	System Responsibility
1. User navigates to the profile section.	2. System displays the profile management interface.
3. User updates their personal details (e.g., name, address, phone number).	4. System validates the inputs and checks for any required fields.
5. User modifies account security settings (e.g., password).	6. System enforces password strength rules and provides feedback.
7. User submits the updated information.	8. System saves the changes to the database and confirms the update.
Extensions	
Actor Action (or Intention)	System Responsibility
3a. If the password does not meet security standards, the user enters a stronger password.	3a. System prompts the user to enter a stronger password.
5a. If the system fails to update the profile due to server issues, the user tries again later.	5a. System notifies the user of the failure and advises retrying.

Add Product Listings

Use Case Name	Add Product Listings
Scope	HawkTU Logistics and Delivery Management System
Level	User Goal
Primary Actor	Seller
Stakeholders and Interests	<ul style="list-style-type: none"> Seller: Wants to easily manage their product listings, adding new products, updating prices, and removing outdated or unavailable items. Customer: Expects the catalogue to be up-to-date with accurate information. System Administrator: Ensures data integrity and prevents errors in product listings.
Preconditions	<ul style="list-style-type: none"> The seller is logged into the system. The seller has access to the product management interface.
Postconditions	<ul style="list-style-type: none"> The product listings are updated with new, modified, or removed items. Changes are saved securely in the database and reflected in the system.
Main Success Scenario	
Actor Action (or Intention)	System Responsibility
1. Seller navigates to the product management section.	2. System displays the product listings interface and prompts for user action
3. Seller chooses desired option. (add, delete, modify listing)	4. System prompts for product details and validates input.
5. Seller enters relevant information.	6. System authenticates the information and updates the catalogue.
Extensions	
Actor Action (or Intention)	System Responsibility
3a. If the seller tries to add existing item.	3a. System prompts error.
5a. Seller Fails to enter all required details.	5a. System prompts user "Fill in Required Details"

Delete and Edit Product Listings

Use Case Name	Delete and Edit Product Listings
Scope	HawkTU Logistics and Delivery Management System
Level	User Goal
Primary Actor	Seller
Stakeholders and Interests	<ul style="list-style-type: none"> Seller: Wants to efficiently manage product listings by removing unavailable products and updating product details to maintain accuracy. Customer: Expects an accurate and updated product catalogue to make informed purchasing decisions. System Administrator: Ensures data integrity and prevents errors in the product catalogue.
Preconditions	<ul style="list-style-type: none"> The seller is logged into the system. The seller has access to the product management interface. The product exists in the catalogue for deletion or editing.
Postconditions	<ul style="list-style-type: none"> The product listing is either updated or removed from the system. Changes are saved securely in the database and reflected in the catalogue.
Main Success Scenario	
Actor Action (or Intention)	System Responsibility
1. Seller navigates to the product management section.	2. System displays the product listings interface with available options.
3. Seller selects a product to delete or edit.	4. System displays the product details.
For Deletion: Seller confirms the deletion action.	For Deletion: System removes the product from the database and updates the catalogue.
For Editing: Seller updates relevant product details.	For Editing: System validates the new details and updates the product in the database.
Extensions	
Actor Action (or Intention)	System Responsibility
3a. If the seller selects a non-existent product for deletion or editing.	3a. System displays an error message stating "Product not found."
4a. If the product details entered during editing are incomplete or invalid.	4a. System prompts the seller to provide complete and valid details.
4b. If the system fails to update or delete due to server issues.	4b. System notifies the seller of the failure and advises retrying later.

Earn and Redeem Loyalty Points

Use Case Name	Earn and Redeem Loyalty Points
Scope	HawkTU Logistics and Delivery Management System
Level	User Goal
Primary Actor	Customer
Stakeholders and Interests	<ul style="list-style-type: none"> • Customer: Wants to earn points from purchases and redeem them for rewards or discounts on future purchases. • Seller: Encourages customer loyalty by offering points for purchases and providing incentives to redeem them. • System Administrator: Ensures the accuracy of points tracking and redemption, maintaining data integrity.
Preconditions	<ul style="list-style-type: none"> • The customer is logged into the system. • The customer has made at least one eligible purchase to earn loyalty points.
Postconditions	<ul style="list-style-type: none"> • Loyalty points are updated in the customer's account after a purchase. • Points are redeemed, generating a coupon code that can be applied during future checkouts.
Main Success Scenario	
Actor Action (or Intention)	System Responsibility
1. Customer completes a purchase.	2. System calculates and adds loyalty points to the customer's account.
3. Customer views their loyalty points balance.	4. System displays the current balance of earned points.
5. Customer chooses to redeem points.	6. System validates the redemption request and deducts the appropriate number of points.
7. Customer receives a coupon code.	8. System generates a unique coupon code and stores it in the customer's account for future use.
Extensions	
Actor Action (or Intention)	System Responsibility
3a. If there is an error in calculating points, the customer contacts support.	3a. System logs the error and notifies support for correction.
5a. If the customer tries to redeem more points than available, they enter a valid amount.	5a. System prompts the customer to enter a valid redemption amount.
6a. Points deduction fails.	6a. System notifies the customer of the failure and advises retrying.
7a. Coupon generation fails.	7a. System logs the failure and notifies the customer to contact support.
8a. Customer attempts to use an invalid coupon code at checkout.	8a. System notifies the customer of the invalid code and prompts them to try again or contact support.

Submit Product Reviews and Ratings

Use Case Name	Submit Product Reviews and Ratings
Scope	HawkTU Logistics and Delivery Management System
Level	User Goal
Primary Actor	Customer
Stakeholders and Interests	<ul style="list-style-type: none"> • Customer: Wants to provide honest feedback on products, helping others make informed decisions and ensuring quality products are highlighted. • Seller: Gains insights into customer satisfaction and areas for improvement, enhancing product quality and reputation. • Other Customers: Depend on reviews and ratings to evaluate product quality before purchasing. • System Administrator: Monitors reviews to ensure compliance with community standards and prevents inappropriate content.
Preconditions	<ul style="list-style-type: none"> • The customer is logged into the system. • The customer has purchased or received the product being reviewed.
Postconditions	<ul style="list-style-type: none"> • The product review and rating are submitted, stored in the database, and displayed under the respective product. • The seller and other customers can view the review and rating.
Main Success Scenario	
Actor Action (or Intention)	System Responsibility
1. Customer navigates to the product review section for a purchased product.	2. System displays the product review submission form.
3. Customer writes a review and assigns a rating (e.g., 1-5 stars).	4. System validates the input to ensure the review is within acceptable length and rating is within range.
5. Customer submits the review and rating.	6. System saves the review and rating in the database and displays a success message.
7. Customer views the submitted review under the product details.	8. System updates the product's overall rating and adds the new review to the product's review list.
Extensions	
Actor Action (or Intention)	System Responsibility
3a. If the review contains inappropriate content, the customer revises it.	3a. System detects inappropriate content using filters and prompts the customer to edit.
5a. If the customer tries to submit without a rating, they are prompted to provide one.	5a. System alerts the customer that a rating is mandatory.
5b. If the system fails to save the review due to server issues, the customer tries again later.	5b. System notifies the customer of the failure and advises retrying.

Place Order

Use Case Name	Place Order
Scope	HawkTU Logistics and Delivery Management System
Level	User Goal
Primary Actor	Customer
Stakeholders and Interests	<ul style="list-style-type: none"> • Customer: Wants a seamless ordering process to purchase products efficiently. • Seller: Expects to receive confirmed orders for inventory management. • System Administrator: Ensures the order process is secure and efficient.
Preconditions	<ul style="list-style-type: none"> • The customer is logged into the system. • The customer has selected products to purchase.
Postconditions	<ul style="list-style-type: none"> • An order is created and stored in the database. • The customer receives an order confirmation.
Main Success Scenario	
Actor Action (or Intention)	System Responsibility
1. Customer selects products and adds them to the cart.	2. System displays the cart with selected products.
3. Customer confirms delivery details.	4. System validates the delivery information.
5. Customer proceeds to payment.	6. System displays payment options (e.g., cash, card).
7. Customer completes the payment process.	8. System processes the payment and stores the order.
Extensions	
Actor Action (or Intention)	System Responsibility
3a. If delivery details are incomplete, the customer corrects them.	3a. System prompts the customer to complete missing information.
7a. If payment processing fails, the customer retries or selects a different payment method.	7a. System notifies the customer of the failure and provides options to retry.

Modify or Cancel Order

Use Case Name	Manage Order
Scope	HawkTU Logistics and Delivery Management System
Level	User Goal
Primary Actor	Customer/Seller
Stakeholders and Interests	<ul style="list-style-type: none"> • Customer: Wants to easily modify or cancel orders based on changing needs. • Seller: Needs to manage inventory and understand changes to orders. • System Administrator: Ensures modifications and cancellations follow established rules.
Preconditions	<ul style="list-style-type: none"> • The customer is logged into the system. • The order is placed and can be modified or canceled based on its status.
Postconditions	<ul style="list-style-type: none"> • The order is modified or canceled in the system. • Notifications are sent to the customer and seller regarding the changes.
Main Success Scenario	
Actor Action (or Intention)	System Responsibility
1. Customer views their order history.	2. System displays current orders with options to modify or cancel.
3. Customer selects an order to modify or cancel.	4. System validates the current status of the order.
5. Customer makes changes or confirms cancellation.	6. System processes the request and updates the order status.
Extensions	
Actor Action (or Intention)	System Responsibility
1a. No Orders are displayed.	1a. System detects that no orders have been made.
3a. If the order cannot be modified/canceled due to status, the customer is informed.	3a. System notifies the customer of the order's status and restrictions.
5a. If modifications are invalid, the customer corrects them.	5a. System prompts the customer to provide valid changes.

Process Payment

Use Case Name	Process Payment
Scope	HawkTU Logistics and Delivery Management System
Level	Subfunctional
Primary Actor	Customer
Stakeholders and Interests	<ul style="list-style-type: none"> • Customer: Wants a secure and efficient payment process. • Seller: Expects timely confirmation of payments for orders. • Payment Processor: Requires accurate transaction handling and security.
Preconditions	<ul style="list-style-type: none"> • The customer is logged into the system. • The customer has selected a payment method.
Postconditions	<ul style="list-style-type: none"> • Payment is processed successfully and recorded in the system. • The customer receives a payment confirmation.
Main Success Scenario	
Actor Action (or Intention)	System Responsibility
1. Customer checks out and is redirected to Payment Handler	2. System presents payment option : Cash, Card, 3rd Party
3. Customer selects an option.	4. System Prompts for required details
5. User enters required details.	6. System processes the payment with the selected payment processor.
	7. System confirms the payment and updates the order status.
	8. System sends a confirmation to the customer.
Extensions	
Actor Action (or Intention)	System Responsibility
5a. If payment details are invalid, the customer corrects them.	5a. System prompts the customer to enter valid payment information.
6a. If payment processing fails, the customer is notified.	6a. System notifies the customer of the failure and advises retrying.

Handle Refunds and Exchanges

Use Case Name	Handle Refunds and Exchanges
Scope	HawkTU Logistics and Delivery Management System
Level	User Goal
Primary Actor	Customer/Seller
Stakeholders and Interests	<ul style="list-style-type: none"> • Customer: Wants a simple process for returns, refunds, or exchanges. • Seller: Needs to manage returns efficiently and understand customer feedback. • System Administrator: Ensures compliance with return policies and manages records.
Preconditions	<ul style="list-style-type: none"> • The customer is logged into the system. • The order is eligible for a refund or exchange based on the policy.
Postconditions	<ul style="list-style-type: none"> • The refund or exchange is processed and recorded in the system. • Notifications are sent to the customer and seller regarding the outcome.
Main Success Scenario	
Actor Action (or Intention)	System Responsibility
1. Customer requests a refund or exchange for a product.	2. System asks for relevant information.
3. Customer provides required information (e.g., order number, reason).	4. System processes the request and updates order status.
	5. System confirms the refund/exchange and sends notifications.
Extensions	
Actor Action (or Intention)	System Responsibility
1a. Requests cannot be made.	1a. System Detects there are no previous orders.
3a. If the required information is incomplete, the customer is prompted to provide it.	3a. System requests the missing information from the customer.
4a. If the product is not eligible for a refund/exchange, the customer is informed.	4a. System notifies the customer of eligibility criteria.

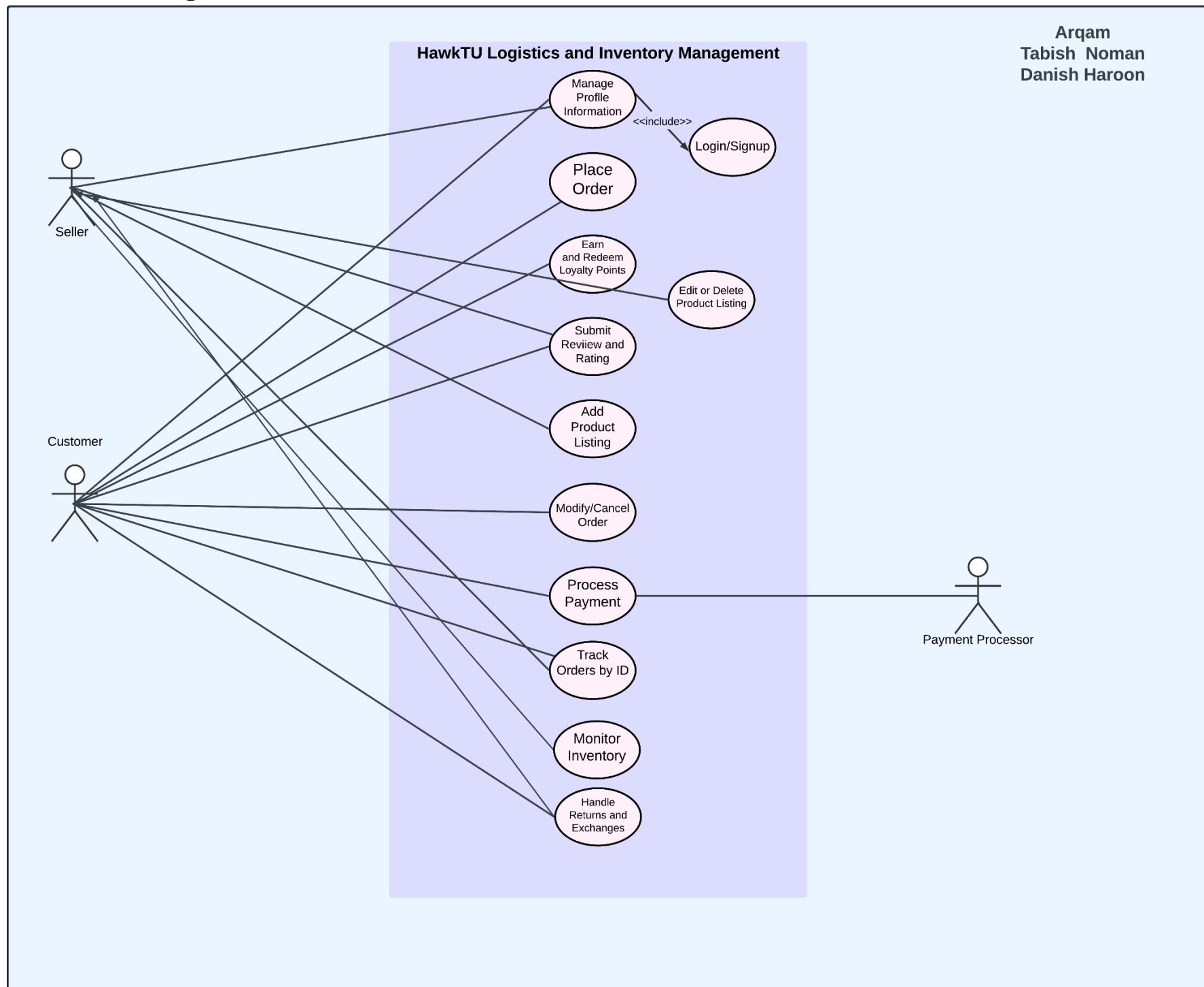
Monitor Inventory Levels

Use Case Name	Monitor Inventory Levels
Scope	HawkTU Logistics and Delivery Management System
Level	System Level
Primary Actor	Automated Management System
Stakeholders and Interests	<ul style="list-style-type: none"> • Sellers: Rely on the system to maintain optimal stock levels for customer demand. • System Administrators: Ensure the system operates effectively to manage inventory.
Preconditions	<ul style="list-style-type: none"> • The system is initialized and running. • Inventory data is current and accurately reflects stock levels.
Postconditions	<ul style="list-style-type: none"> • Inventory levels are continuously monitored and updated by the system. • Notifications are generated for items that require restocking.
Main Success Scenario	
Actor Action (or Intention)	System Responsibility
1. System continuously monitors current inventory levels.	2. System displays inventory status, including stock quantities and alerts.
3. System records updates to inventory levels based on sales or new stock.	4. System triggers notifications for low-stock items.
Extensions	
Condition	System Responsibility
1a. If inventory data is outdated, the system prompts for a refresh.	1a. System notifies stakeholders of the need to refresh data.
3a. If an update fails, the system logs the error.	3a. System alerts stakeholders about the failure and suggests retrying.

Track Orders by ID

Use Case Name	Track Orders by ID
Scope	HawkTU Logistics and Delivery Management System
Level	User Goal
Primary Actor	Customer/Guest
Stakeholders and Interests	<ul style="list-style-type: none"> • Customer: Wants to monitor the status and location of their order for timely updates. • Guest: Aims to check order status without needing to log in. • System Administrator: Ensures the tracking system functions correctly and securely.
Preconditions	<ul style="list-style-type: none"> • The order has been placed and assigned a unique tracking ID.
Postconditions	<ul style="list-style-type: none"> • The customer or guest can view the current status and location of the order.
Main Success Scenario	
Actor Action (or Intention)	System Responsibility
1. Customer or guest enters the tracking ID on the tracking page.	2. System retrieves the order details associated with the tracking ID. 3. System displays the current status and location of the order.
4. Customer or guest reviews the order status and estimated delivery time.	
Extensions	
Actor Action (or Intention)	System Responsibility
1a. If the tracking ID is invalid, the system notifies the user.	1a. System displays an error message indicating the tracking ID is not found.
3a. If there is a system error, the customer or guest is informed.	3a. System alerts the user of any issues retrieving order details.

2.5 Use Case Diagram



3. Other Nonfunctional Requirements

3.1 Performance Requirements

- The system must handle up to 1,000 concurrent users with minimal latency (<2 seconds per operation).
- Real-time tracking updates should reflect within 1 second of status change.

3.2 Safety Requirements

- System data backups must occur daily to ensure data recoverability.
- All critical operations must include a fail-safe mechanism to prevent data corruption.

3.3 Security Requirements

- Secure User authentication through Password Hashing
- Data encryption for all stored and transmitted information using AES-256 standards.

3.4 Software Quality Attributes

- **Usability:** The interface should be intuitive and require minimal training.
- **Reliability:** System uptime must exceed 99.9%.
- **Scalability:** The system should scale efficiently to support up to 1000 users.
- **Maintainability:** Modular architecture allows updates without disrupting operations.

3.5 Business Rules

- Only authenticated users can access order and inventory management functionalities.
- Customers must provide valid order IDs to access tracking information.

3.6 Operating Environment

Backend: Java (Spring Boot) running on Linux-based servers.

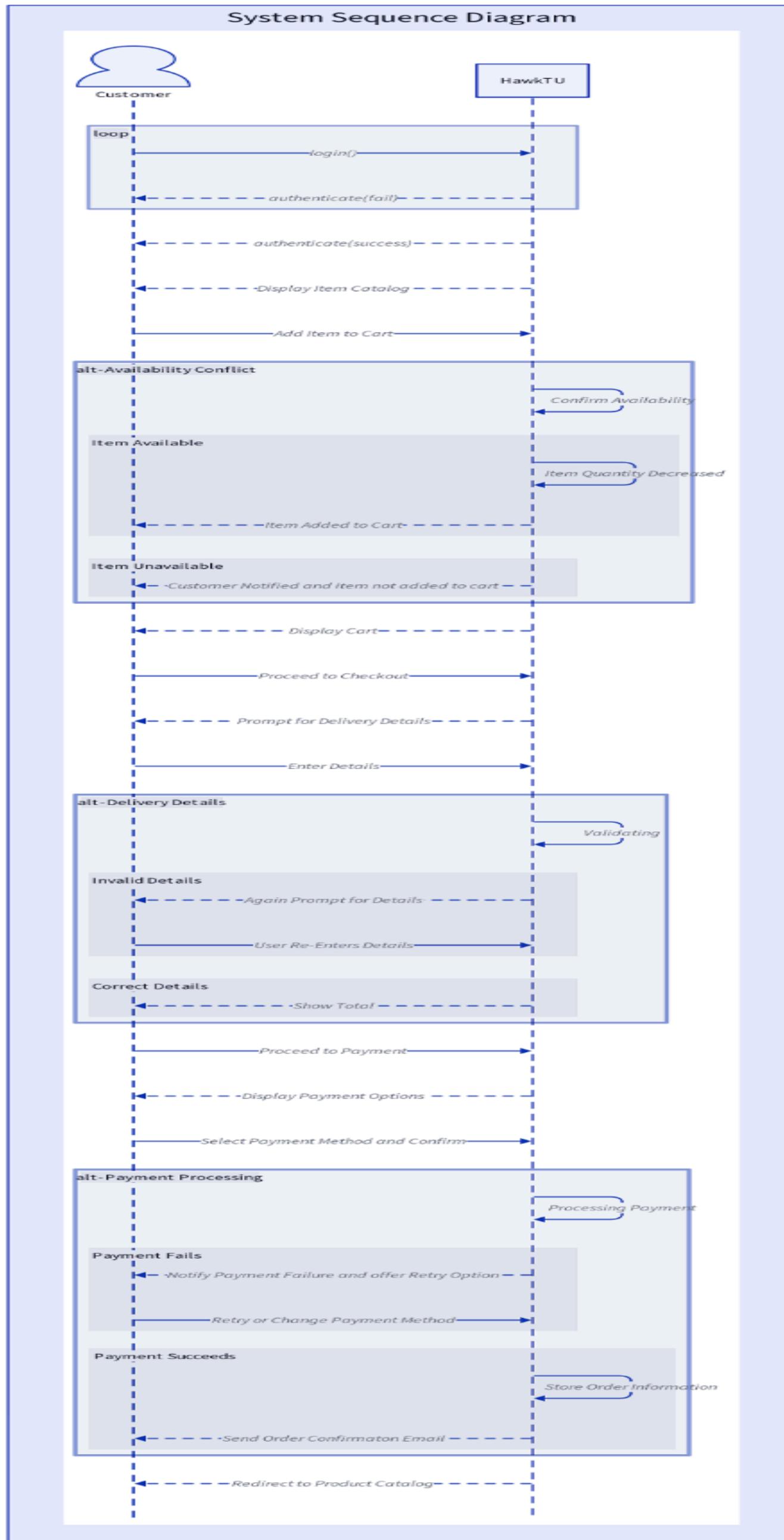
Database: PostgreSQL

Frontend: React-based web interface, compatible with major browsers (Chrome, Firefox, Safari).

3.7 User Interfaces

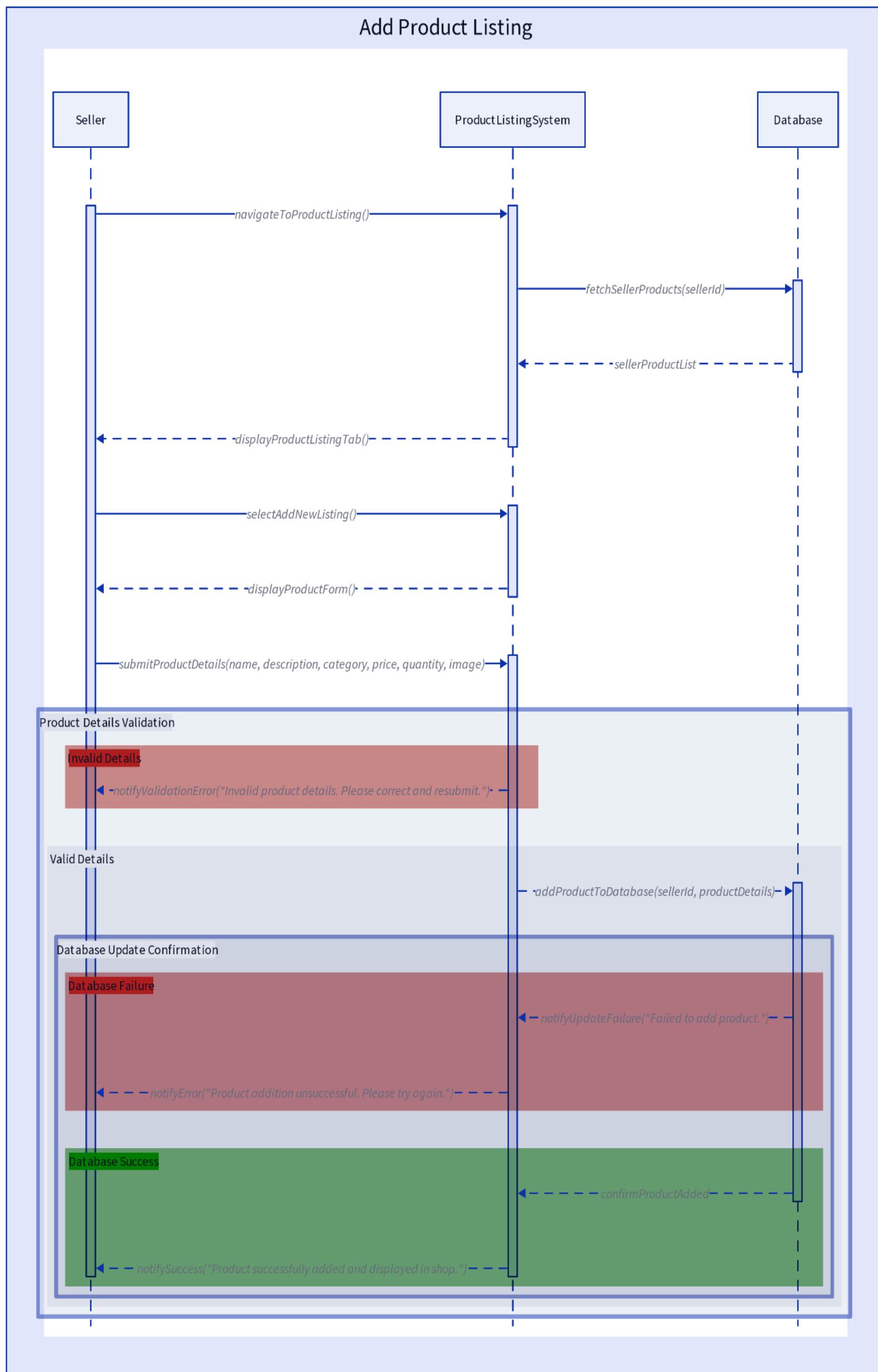
- Seller Dashboard: Includes product listing, order management, and analytics panels.
- Customer Dashboard: Supports order tracking via unique order IDs, Ordering Products, Submit Reviews etc.

4. System Sequence Diagram

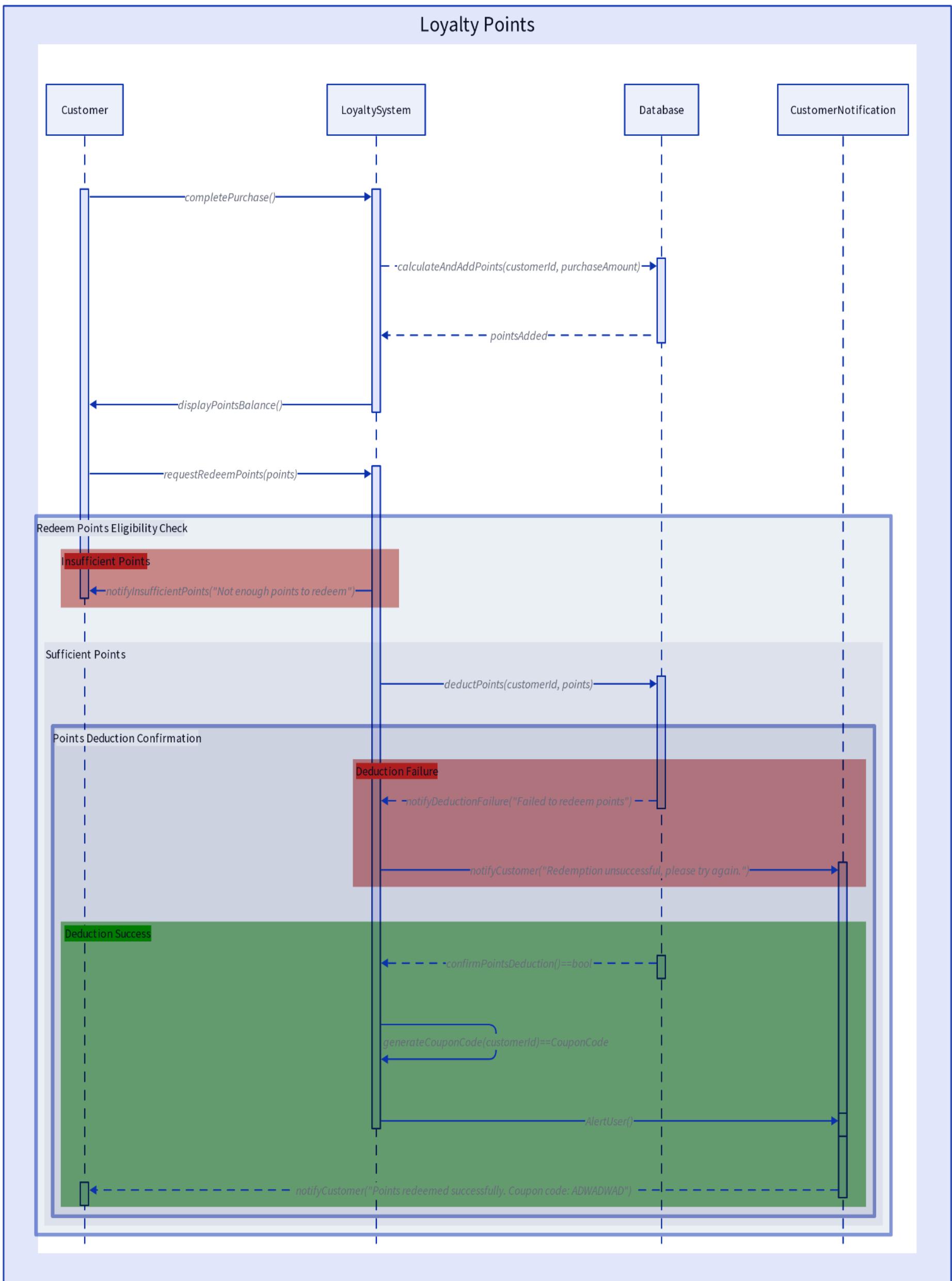


5. Sequence Diagram

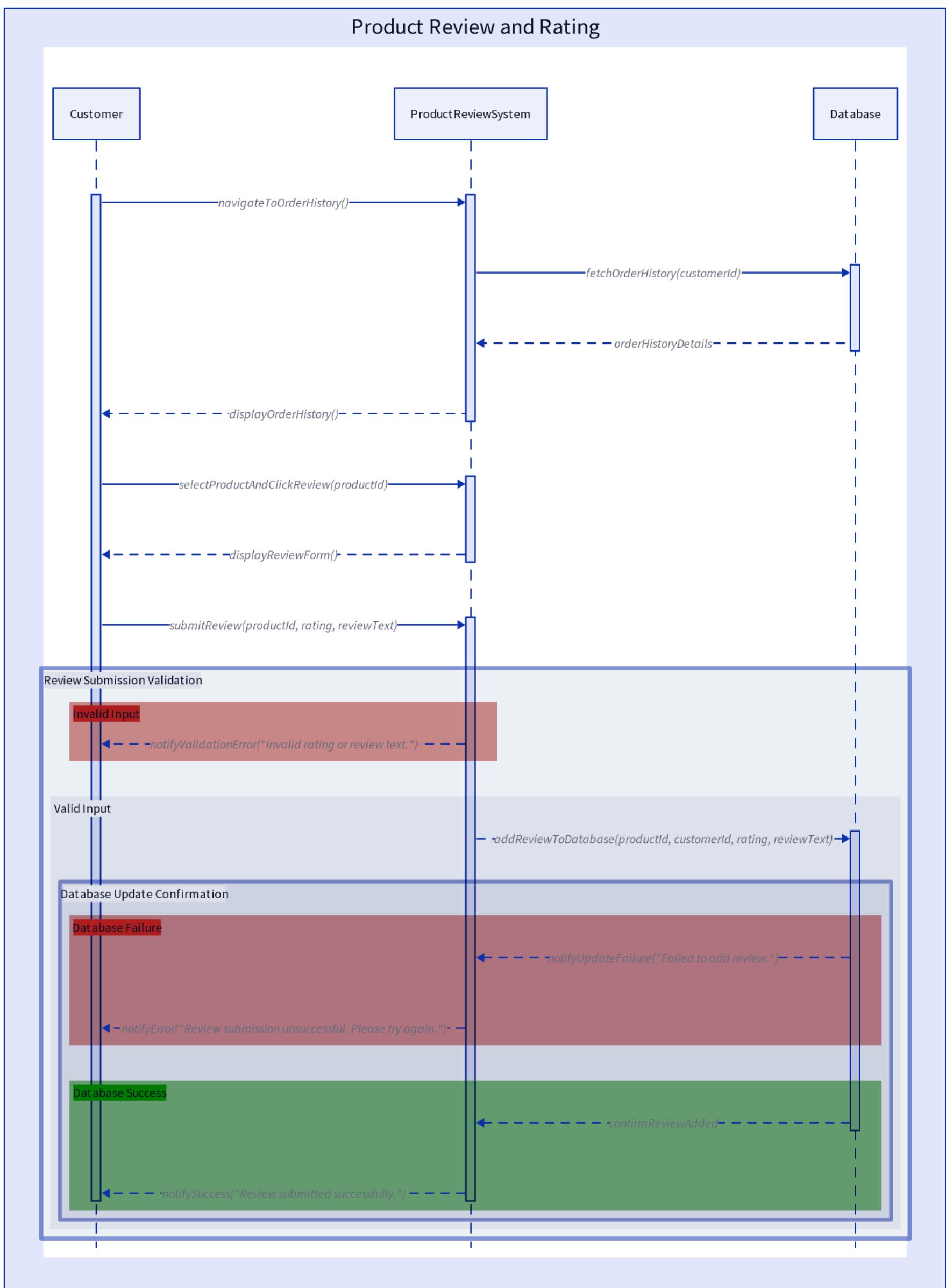
1. Add Product Listing



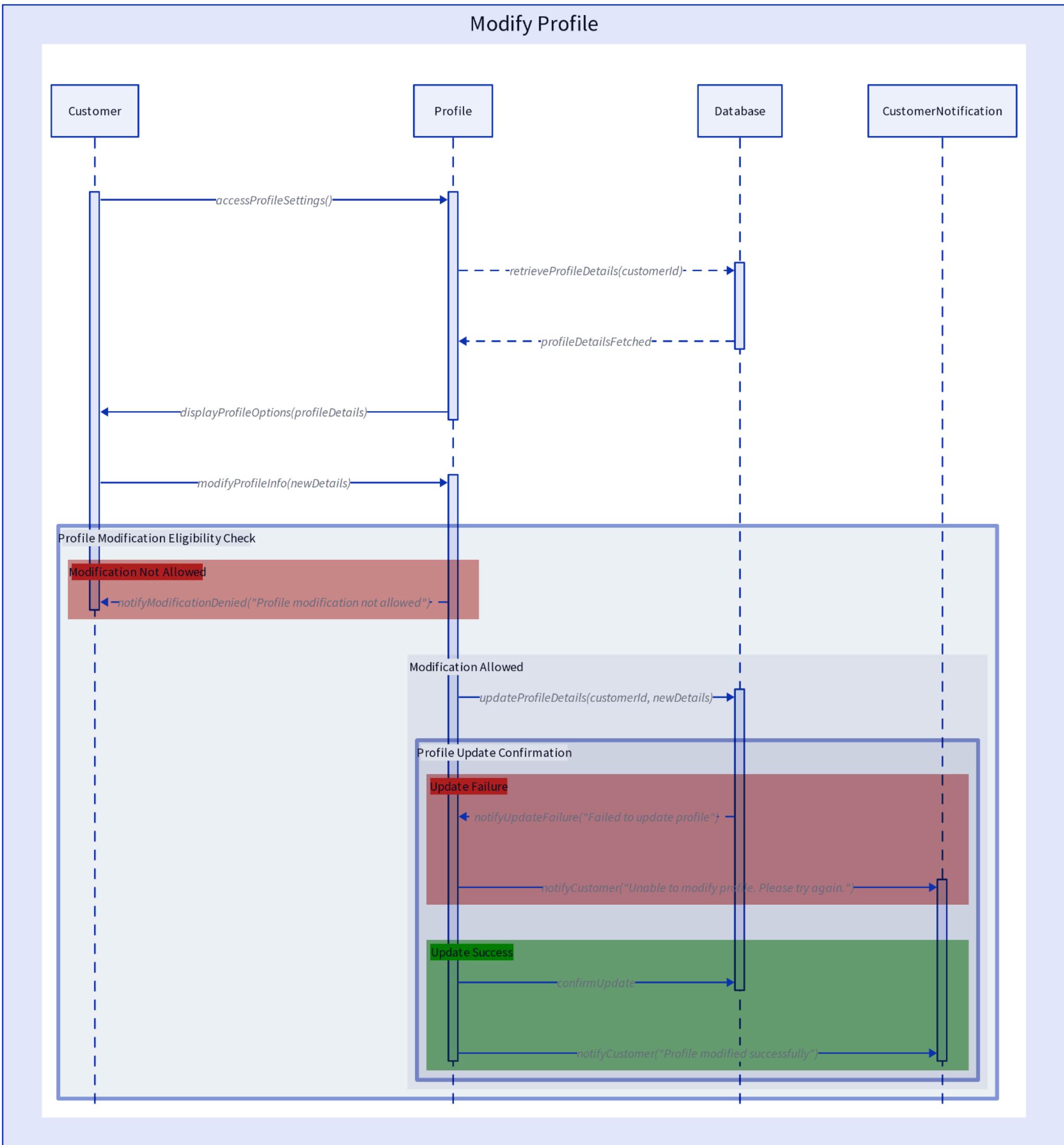
2. Loyalty Points



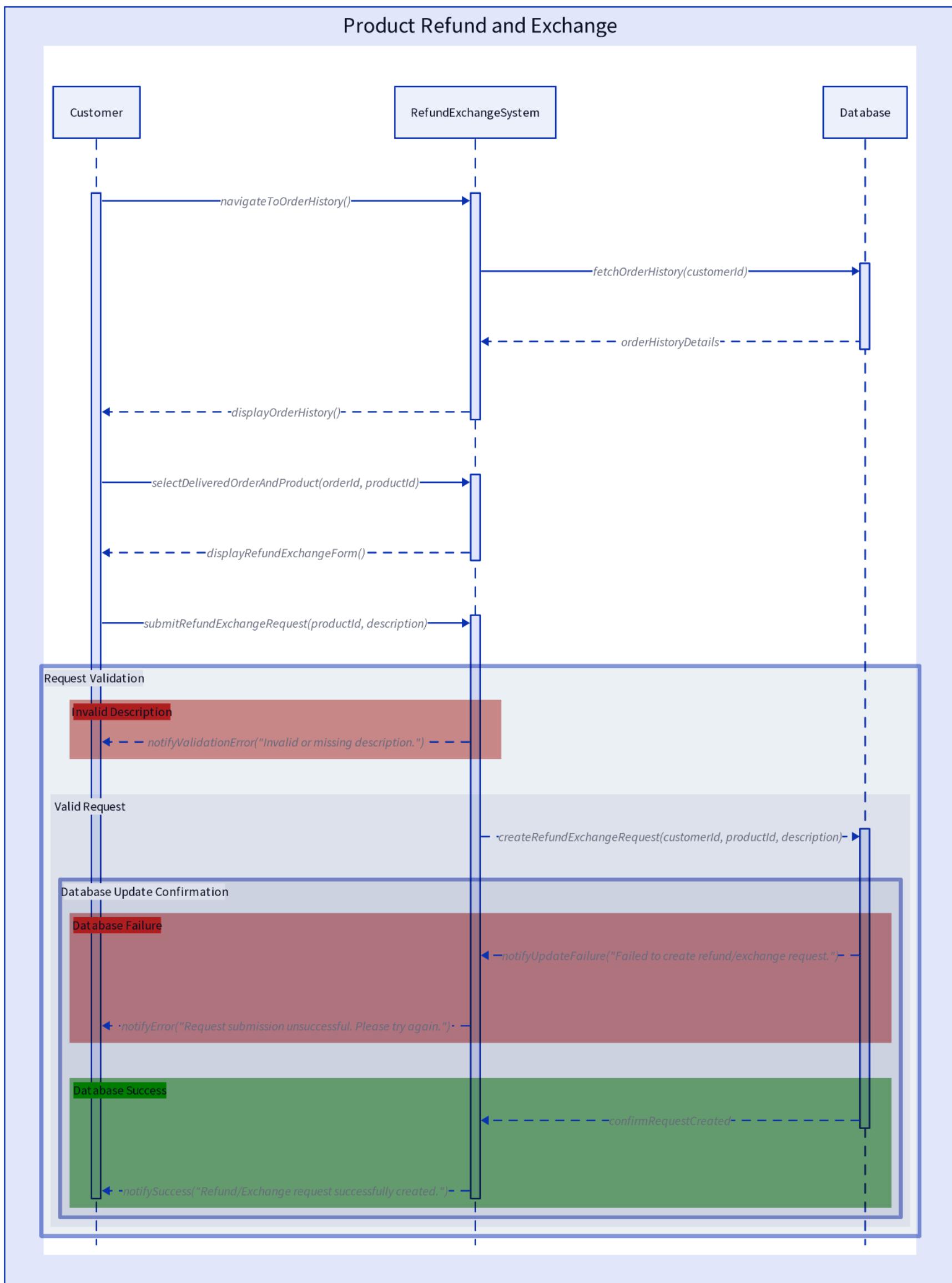
3. Review and Ratings



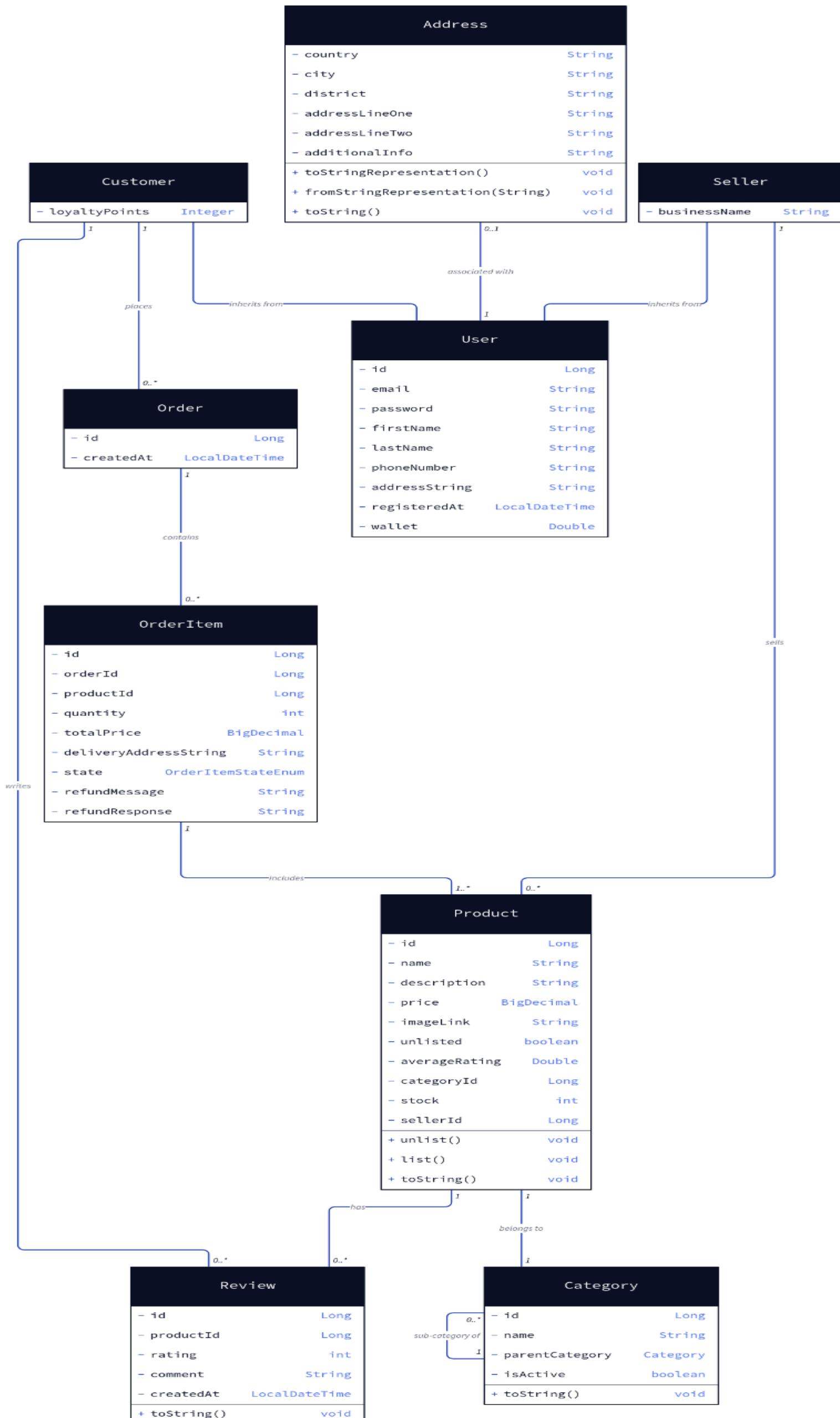
4.Modify Profile



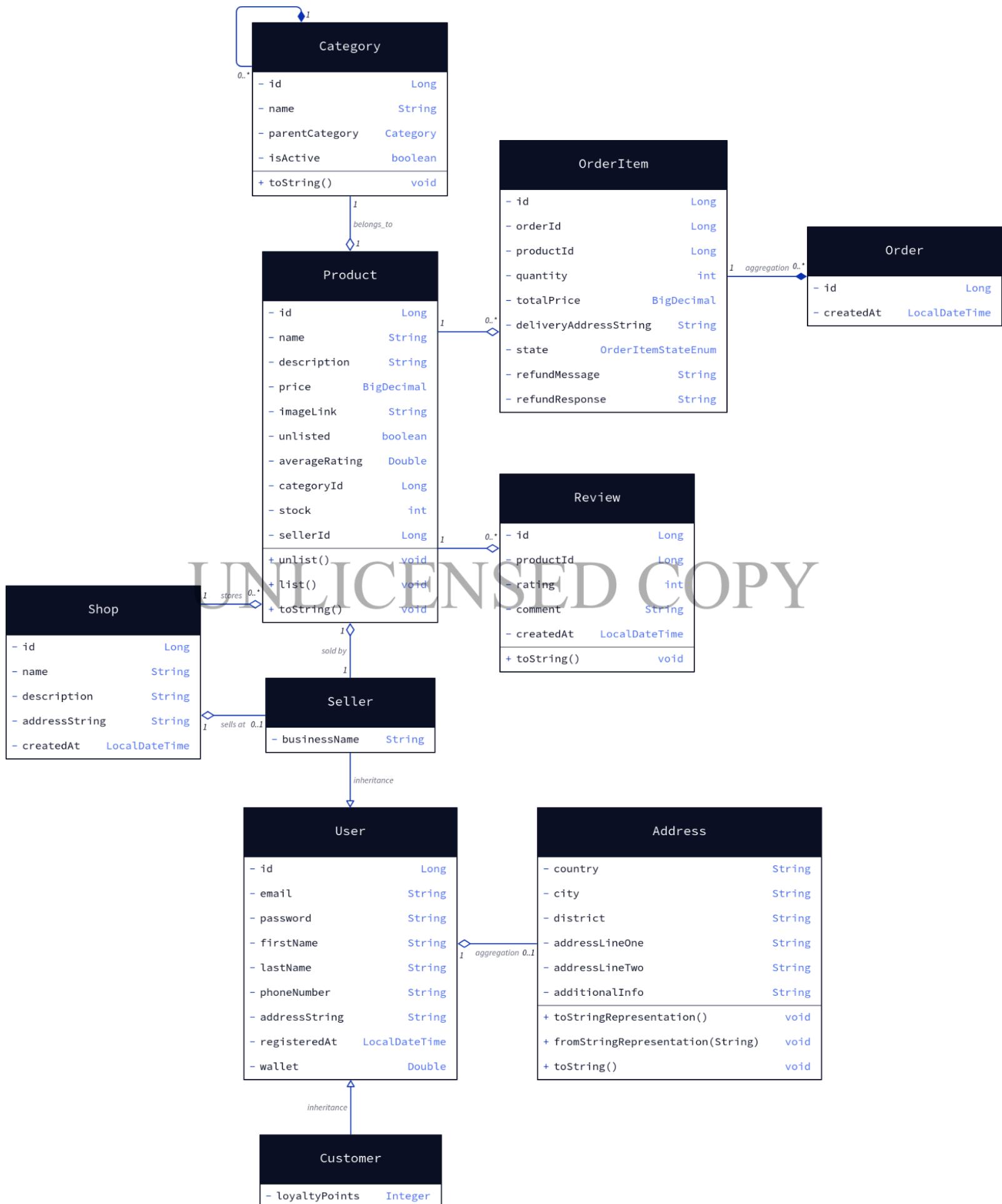
5. Refund and Exchange



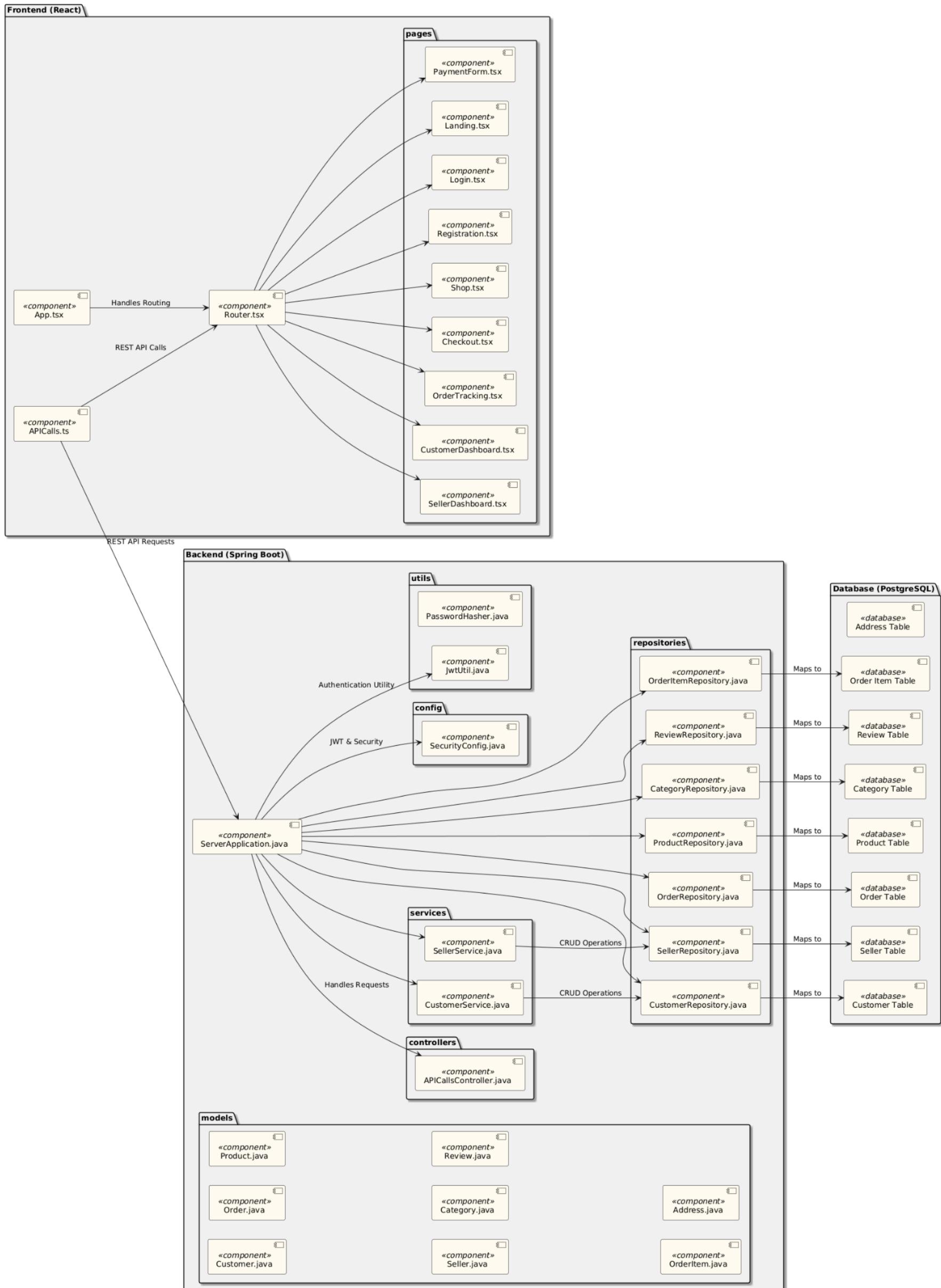
6. Domain Model



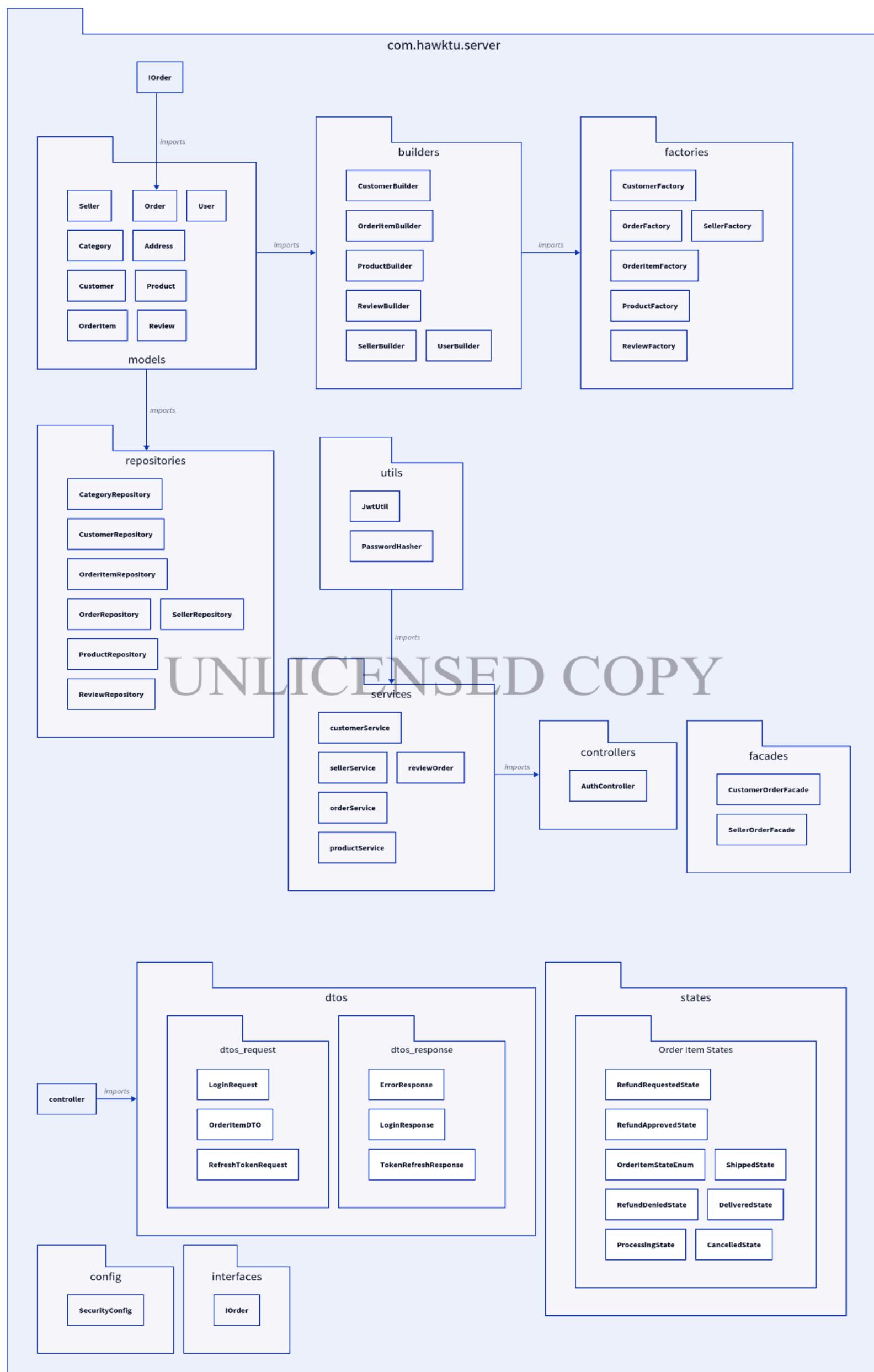
7. Class Diagram



8. Component Diagram



9. Package Diagram



10. Deployment Diagram

