Software Requirements and Design Document

for

Zarooriyaat Platform

Prepared by:

Mahad Rehman Durrani 22I-0792

Muhammad Masab Hammad 22I-1004

Muhammad Hashim Awan 22I-0976

23 March 2025

1. Introduction

1.1 Purpose

This document specifies the software requirements for Zarooriyaat Platform, version It details the system architecture and functionalities aimed at supporting small to medium-sized businesses in launching online stores without initial costs. This SRS will guide the development, deployment, and ongoing maintenance of the platform, targeting accessibility for emerging businesses.

1.2 Product Scope

Our project targets small to medium-sized businesses globally, providing a cost-effective solution for launching online stores. Unlike previous existing tools such as Shopify, which impose monthly fees, our platform will allow businesses to operate freely without initial charges. We aim to serve businesses that want to avoid upfront costs, particularly in their early stages, while introducing a small percentage-based charge only after they surpass a sales threshold of Rs. 40,000

1.3 Title

Zarooriyaat Platform: Unlocking E-Commerce for Everyone

1.4 Objectives

Our project targets small to medium-sized businesses globally, providing a cost-effective solution for launching online stores. Unlike previous existing tools such as Shopify, which impose monthly fees, our platform will allow businesses to operate freely without initial charges. We aim to serve businesses that want to avoid upfront costs, particularly in their early stages, while introducing a small percentage-based charge only after they surpass a sales threshold of Rs. 40,000

1.5 Problem Statement

Many existing e-commerce platforms, such as Shopify, require sellers to pay a monthly fee, which can be discouraging for new businesses with limited initial revenue. These recurring fees present a barrier, especially for startups and small businesses, making it difficult for them to gain traction without additional financial pressure

2. System's Functional Requirements

2.1.1.1 Story #1: User Registration

1.1 The system shall allow users to register with an email, username, and password.
1.2 The system shall validate the uniqueness of the email to prevent duplicate accounts.
1.3 The system shall send a verification email to activate the user account.
1.4 The system shall prevent registration with weak passwords and enforce password complexity rules.

2.1.1.2 Story #2: Secure Login

2.1 The system shall allow users to log in using their email and password.
2.2 The system shall hash and securely store passwords.
2.3 The system shall lock an account after three consecutive failed login attempts.
2.4 The system shall allow users to reset passwords via a secure email link.
2.5 The system shall support two-factor authentication (optional).

2.1.1.3 Story #3: Seller Profile Creation

3.1 The system shall allow registered users to create a seller profile.
3.2 The system shall require sellers to provide business details (name, address, contact info).
3.3 The system shall allow sellers to upload business verification documents.
3.4 The system shall notify sellers upon successful verification of their profile.

2.1.1.4 Story #4: Manage Profile

4.1 The system shall allow users (both buyers and sellers) to update their profile information.
4.2 The system shall allow users to update their email, password, and contact details.
4.3 The system shall provide a history of recent account activities.
4.4 The system shall allow users to delete their accounts permanently.

2.1.1.5 Story #5: Product Search

5.1 The system shall allow users to search products by name, category, and brand.
5.2 The system shall display search suggestions as users type in the search bar.
5.3 The system shall return relevant products based on keyword matching.
5.4 The system shall allow users to refine search results with filters.

2.1.1.6 Story #6: Wishlist

6.1 The system shall allow users to add products to their wishlist.
6.2 The system shall allow users to remove products from their wishlist.
6.3 The system shall notify users when a wishlist item goes on sale.

2.1.1.7 Story #7: View Product Details

7.1 The system shall display product details including images, description, price, and seller information.

7.2 The system shall show product reviews and ratings.
7.3 The system shall allow users to share product links via social media.

2.1.1.8 Story #8: Product Catalog Management (Seller Side)

8.1 The system shall allow sellers to add, edit, and delete products.
8.2 The system shall allow sellers to upload product images.
8.3 The system shall allow sellers to manage inventory (stock availability).

2.1.1.9 Story #9: Secure Checkout

9.1 The system shall allow users to add items to their cart and proceed to checkout.
9.2 The system shall allow users to enter shipping and billing information.
9.3 The system shall support multiple payment options (Credit/Debit Card, PayPal, Cash on Delivery).
9.4 The system shall securely process payments and generate invoices.
9.5 The system shall display an order confirmation message upon successful checkout.

2.1.1.10 Story #10: Order Tracking

10.1 The system shall allow users to track their order status (Pending, Shipped, Delivered).
10.2 The system shall provide estimated delivery dates.
10.3 The system shall allow users to cancel an order before it is shipped.

2.1.1.11 Story #11: Email Order Updates

11.1 The system shall send email notifications for order confirmation, shipping, and delivery.
11.2 The system shall send an email receipt to users after a successful purchase.
11.3 The system shall notify users if an order is delayed or canceled.

2.1.2 Story #12: Sales Analytics

12.1 The system shall generate sales reports based on daily, weekly, and monthly performance.
12.2 The system shall provide graphical representations of sales trends.
12.3 The system shall allow sellers to filter sales data by product, category, or date range.
12.4 The system shall display top-selling products and revenue statistics.

2.1.3 Story #13: Product Feedback

13.1 The system shall allow users to leave feedback in the form of ratings and reviews.
13.2 The system shall moderate and filter inappropriate reviews using predefined policies.
13.3 The system shall allow sellers to respond to customer feedback.
13.4 The system shall display average product ratings based on user feedback.

2.1.4 Story #14: Inventory Management

14.1 The system shall allow sellers to add, edit, and remove products from inventory.
14.2 The system shall track stock levels in real-time.
14.3 The system shall prevent customers from purchasing out-of-stock items.
14.4 The system shall generate stock reports for sellers.

2.1.5 Story #16: Product Comparison

16.1 The system shall allow users to select multiple products for comparison.
16.2 The system shall display product specifications side by side.
16.3 The system shall highlight key differences such as price, features, and ratings.
16.4 The system shall allow users to save comparisons for future reference.

2.1.6 Story #17: Promotions and Discounts

17.1 The system shall allow sellers to create and manage discount campaigns.
17.2 The system shall apply discounts automatically during checkout.
17.3 The system shall display discounted prices on product listings.
17.4 The system shall notify users of ongoing promotions via email or dashboard alerts.

2.1.7 Story #18: Read Customer Reviews

18.1 The system shall allow users to view customer reviews on product pages.
18.2 The system shall provide sorting options for reviews (e.g., most recent, highest-rated).
18.3 The system shall verify that reviews come from authenticated buyers.
18.4 The system shall allow users to report inappropriate reviews.

2.1.8 Story #19: Low Stock Alerts

19.1 The system shall send automatic alerts to sellers when stock levels drop below a defined threshold.
19.2 The system shall display a "low stock" label on affected products.
19.3 The system shall provide an inventory dashboard summarizing low-stock products.
19.4 The system shall allow sellers to set custom stock alert thresholds.

2.1.9 Story #20: Real-time Shipment Tracking

20.1 The system shall allow customers to track their shipments in real-time.
20.2 The system shall integrate with third-party courier APIs to fetch tracking details.
20.3 The system shall send automatic shipment status updates via email and dashboard notifications.
20.4 The system shall display estimated delivery dates based on courier data.

3. System's Non-Functional Requirements

3.1 Product Requirements

These requirements define the quality attributes and constraints of the system.

3.1.1 Performance Requirements

- The system shall handle up to **1,000 concurrent users** without performance degradation.
- The response time for **search queries** shall not exceed **2 seconds** under normal load.
- The checkout process shall complete within **5 seconds** after the payment is initiated.

3.1.2 Security Requirements

- The system shall enforce **multi-factor authentication (MFA)** for sellers.
- All sensitive data (e.g., passwords, payment details) shall be **encrypted** using **AES-256**.
- The system shall implement **role-based access control (RBAC)** to restrict access to admin functionalities.
- The system shall protect against **SQL injection.**

3.1.3 Availability Requirements

- The system shall have **99.9% uptime**, allowing no more than **8.76 hours of downtime per year**.
- The system shall support **automatic failover** in case of server failure.
- The system shall perform **automated database backups** every **6 hours**.

3.1.4 Usability Requirements

- The system shall have a **responsive UI** to ensure compatibility with **mobile, tablet, and desktop**.
- The system shall follow **WCAG 2.1 Level AA** accessibility guidelines.
- The system shall provide **tooltips and help sections** to assist users.

3.1.5 Maintainability & Scalability Requirements

- The system shall be **modular**, allowing easy addition of new features.
- The system shall support **horizontal scaling**, allowing additional servers to be added when traffic increases.
- Codebase shall follow **Google Java Coding Standards** for maintainability.

3.2 Organizational Requirements

These requirements define constraints based on company policies, development standards, and internal procedures.

3.2.1 Development Requirements

- The system shall be developed using **Agile methodology** with **3 iterations (Sprints)**.
- The project shall use **GitHub for version control**, with a minimum of **one commit per feature**.
- The development shall follow **CI/CD (Continuous Integration/Continuous Deployment)** using GitHub Actions.

3.2.2 Compliance Requirements

- The system shall comply with **GDPR (General Data Protection Regulation)** for user data protection.
- The system shall follow **ISO 27001** standards for security management.
- The payment module shall be **PCI DSS (Payment Card Industry Data Security Standard) compliant**.

3.2.3 Training Requirements

- The system shall provide **training documentation** for sellers on how to list products and manage orders.
- Admins shall undergo **security training** to handle user data safely.

3.3 External Requirements

These requirements define dependencies on external factors like legal, regulatory, and system integrations.

3.3.1 Regulatory Requirements

- The system shall comply with **local e-commerce laws** in the target operating countries.
- The system shall generate invoices that meet **taxation requirements (e.g., VAT, GST)**.
- The system shall follow **consumer protection laws** for refunds and returns.

3.3.2 Integration Requirements

- The system shall integrate with **third-party payment gateways** like **Stripe, PayPal, and Razorpay**.
- The system shall support **third-party shipping APIs** for real-time tracking (e.g., FedEx, DHL, UPS).
- The system shall allow integration with **social media platforms** for login (Google, Facebook, Apple).

3.3.3 Environmental Requirements

- The database shall be **hosted on a high-availability server** to prevent data loss.
- The system shall be compatible with **all modern browsers (Chrome, Edge, Firefox, Safari)**.

4. User Stories

4.1.1.1 Story #1: User Registration

As a new user,
I want to register an account using my email, username, and password,
So that I can access the system's features.

Pre-conditions:

- The user must provide a unique email.
- The user must enter a strong password that meets complexity requirements.

Post-conditions:

- A verification email is sent to activate the account.
- The system prevents duplicate registrations.

4.1.1.2 Story #2: Secure Login

As a registered user,
I want to log in securely using my email and password,
So that I can access my account and system features.

Pre-conditions:

- The user must have a registered account.

Post-conditions:

- The system securely hashes and stores passwords.
- The system locks the account after three failed login attempts.
- Users can reset passwords through email.
- Two-factor authentication (if enabled) is required.

4.1.1.3 Story #3: Seller Profile Creation

As a registered user,
I want to create a seller profile with my business details,
So that I can sell products on the platform.

Pre-conditions:

- The user must have a verified account.
- The user must provide business details and verification documents.

Post-conditions:

- The system notifies the user upon successful profile verification.
- The seller gains access to product management features.

4.1.1.4
Story #4: Manage Profile

As a registered user,
I want to update or delete my profile information,
So that I can keep my account up to date or remove my presence from the platform.

Pre-conditions:

- The user must be logged in.

Post-conditions:

- The profile information is updated.
- Account deletion is permanent and irreversible.

4.1.1.5

Story #5: Product Search

As a buyer,
I want to search for products by name, category, or brand,
So that I can quickly find the items I am looking for.

Pre-conditions:

- The user must enter a valid search term.

Post-conditions:

- The system displays relevant results.
- Search filters can refine the results.

4.1.1.6

Story #6: Wishlist

As a buyer,
I want to add products to my wishlist,
So that I can easily track items I want to purchase later.

Pre-conditions:

- The user must be logged in.

Post-conditions:

- The wishlist is updated.
- Users receive notifications when wishlist items go on sale.

4.1.1.7

Story #7: View Product Details

As a buyer,
I want to view detailed product information,
So that I can make informed purchasing decisions.

Pre-conditions:

- The product must be available in the system.

Post-conditions:

- The system displays product details, images, reviews, and seller info.
- Users can share product links.

4.1.1.8 Story #8: Product Catalog Management (Seller Side)

As a seller,
I want to manage my product catalog,
So that I can add, edit, or remove products as needed.

Pre-conditions:

- The seller must have a verified account.

Post-conditions:

- The catalog updates reflect in the product listings.
- Inventory levels are maintained.

4.1.1.9
Story #9: Secure Checkout

As a buyer,
I want to purchase items securely,
So that I can complete my transactions without issues.

Pre-conditions:

- The user must have items in the cart.
- The user must provide valid payment details.

Post-conditions:

- Payment is securely processed.
- An order confirmation message is displayed.

4.1.1.10

Story #10: Order Tracking

As a buyer,
I want to track my order status,
So that I can stay updated on delivery progress.

Pre-conditions:

- The user must have placed an order.

Post-conditions:

- The system provides real-time order status.
- Users can cancel orders before shipment.

4.1.1.11

Story #11: Email Order Updates

As a buyer,
I want to receive email updates on my orders,
So that I am informed about my purchase status.

Pre-conditions:

- The user must have placed an order.

Post-conditions:

- The system sends order confirmation, shipping, and delivery emails.

4.1.1.12

Story #12: Sales Analytics

As a seller,
I want to view sales analytics,
So that I can track my business performance.

Pre-conditions:

- The seller must have sales data.

Post-conditions:

- The system generates reports and trends.

4.1.1.13

Story #13: Product Feedback

As a buyer,
I want to leave reviews and ratings,
So that I can share my experience with other users.

Pre-conditions:

- The user must have purchased the product.

Post-conditions:

- The review appears on the product page.

4.1.1.14

Story #14: Inventory Management

As a seller,
I want to manage my inventory,
So that I can ensure stock levels are accurate.

Pre-conditions:

- The seller must have products listed.

Post-conditions:

- The system updates stock levels.

4.1.1.15
Story #16: Product Comparison

As a buyer,
I want to compare multiple products,
So that I can choose the best one.

Pre-conditions:

- The user must select products for comparison.

Post-conditions:

- A side-by-side comparison is displayed.

4.1.1.16 Story #17: Promotions and Discounts

As a seller,
I want to create discounts,
So that I can attract more buyers.

Pre-conditions:

- The seller must define a discount campaign.

Post-conditions:

- The system applies discounts automatically.

4.1.1.17

Story #18: Read Customer Reviews

As a buyer,
I want to read customer reviews,
So that I can make informed purchase decisions.

Pre-conditions:

- The product must have reviews.

Post-conditions:

- Reviews are displayed and sortable.

4.1.1.18 Story #19: Low Stock Alerts

As a seller,
I want to receive alerts when stock is low,
So that I can restock items before they run out.

Pre-conditions:

- The seller must have products in inventory.

Post-conditions:

- The system sends low-stock alerts.

4.1.1.19 Story #20: Real-time Shipment Tracking

As a buyer,
I want to track my shipment in real-time,
So that I know when to expect my delivery.

Pre-conditions:

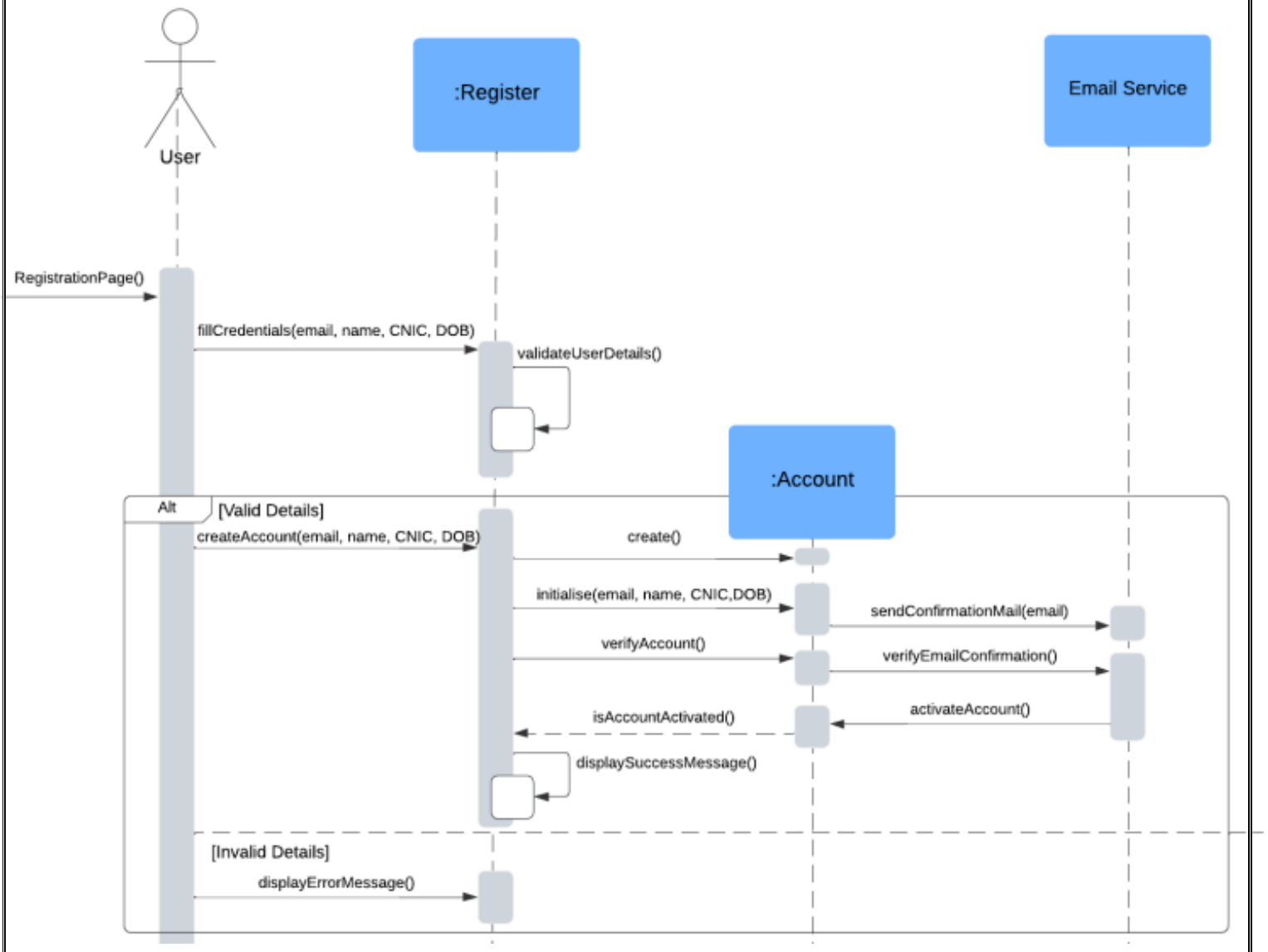
- The order must be shipped.

Post-conditions:

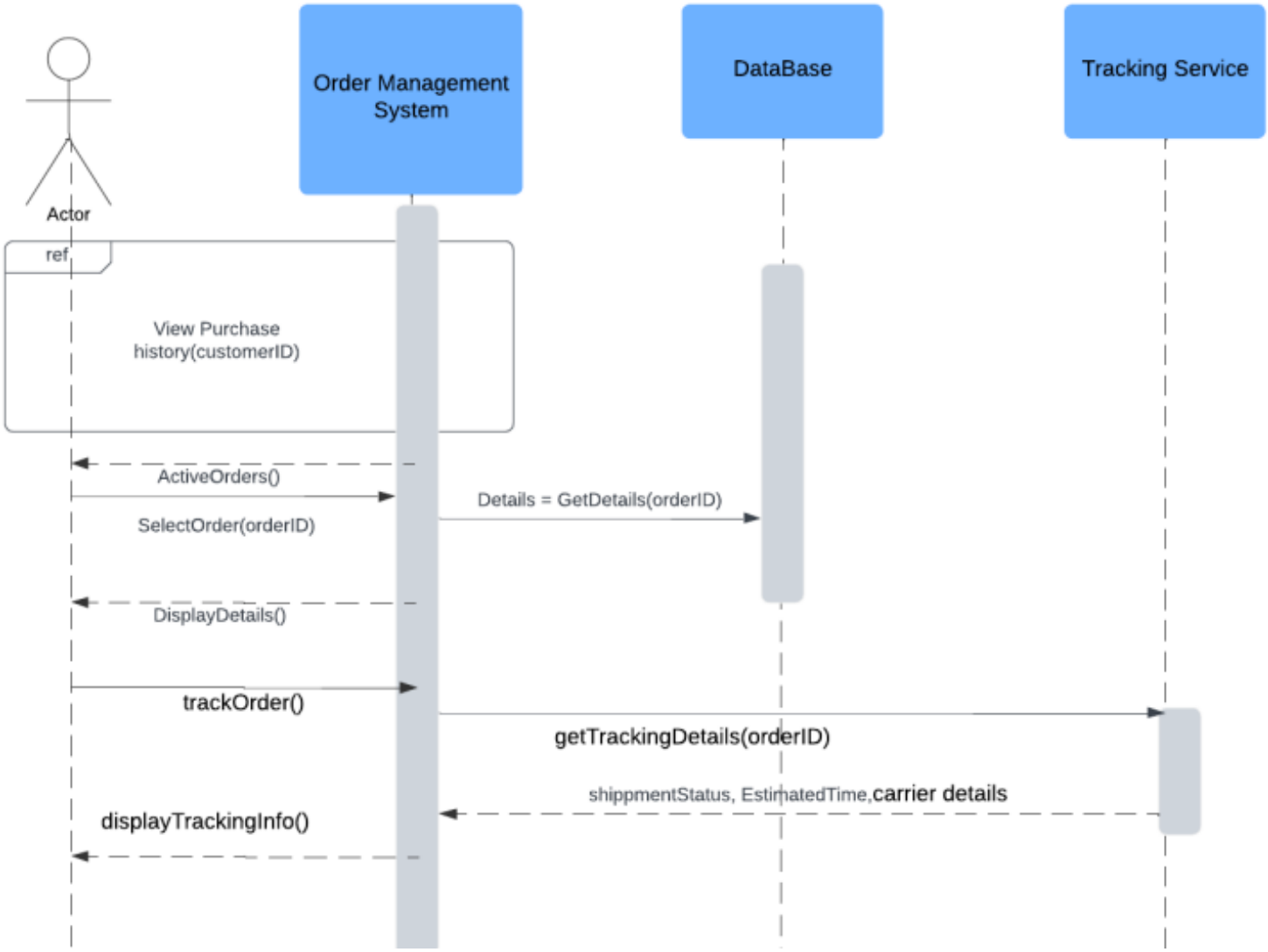
- The system displays real-time tracking updates.

5. Use Case Diagrams

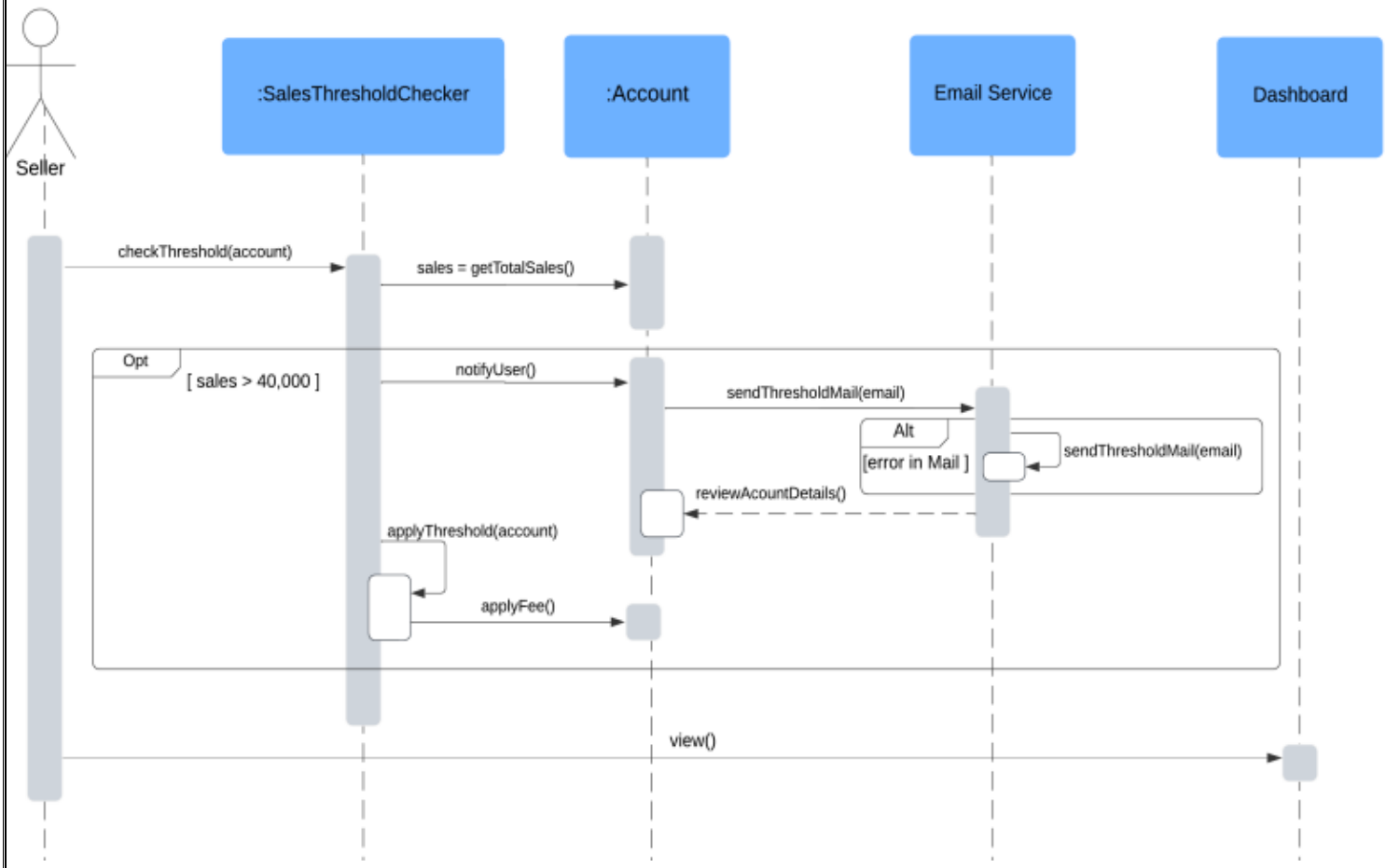
- **Register User**



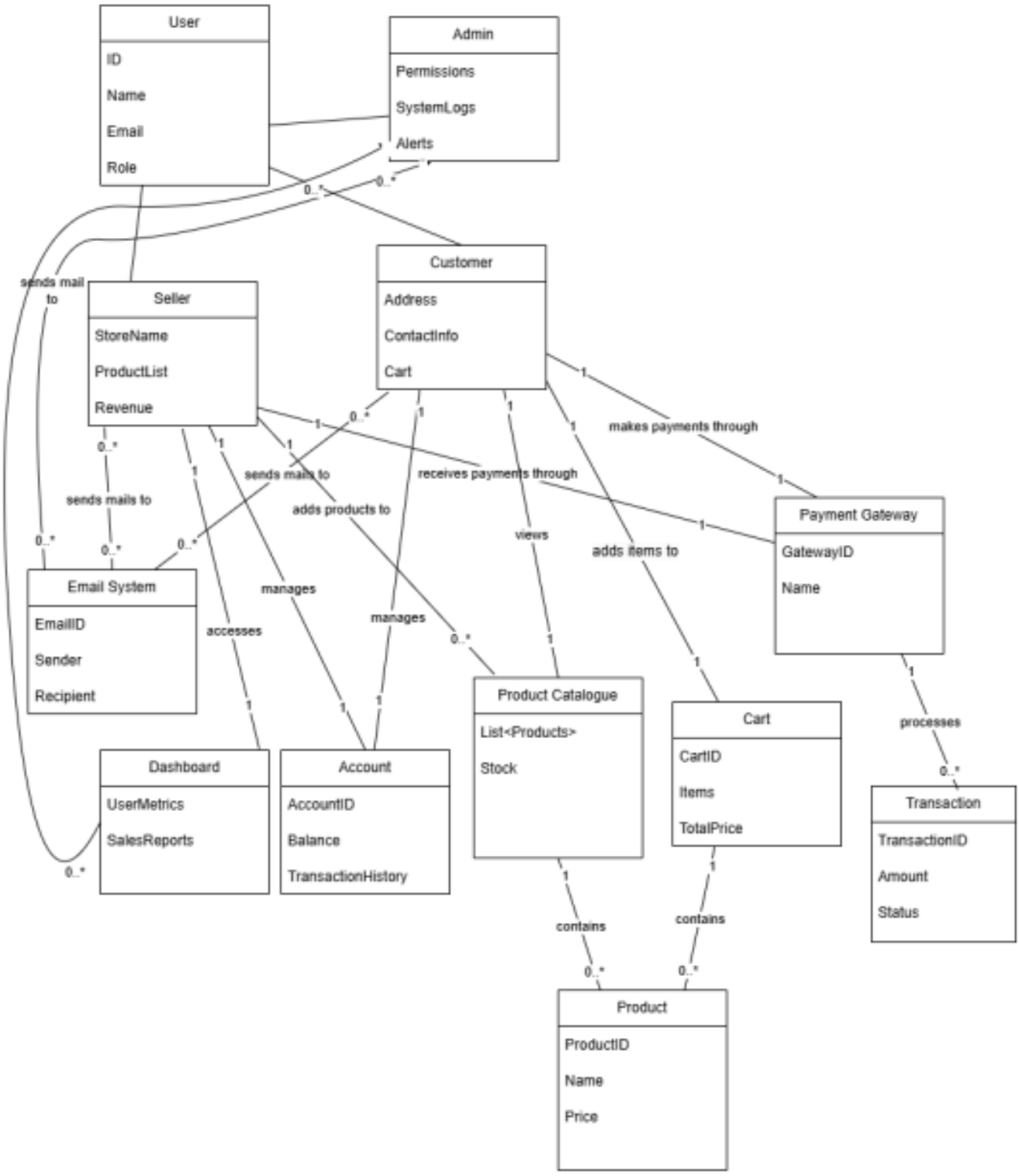
Order Management



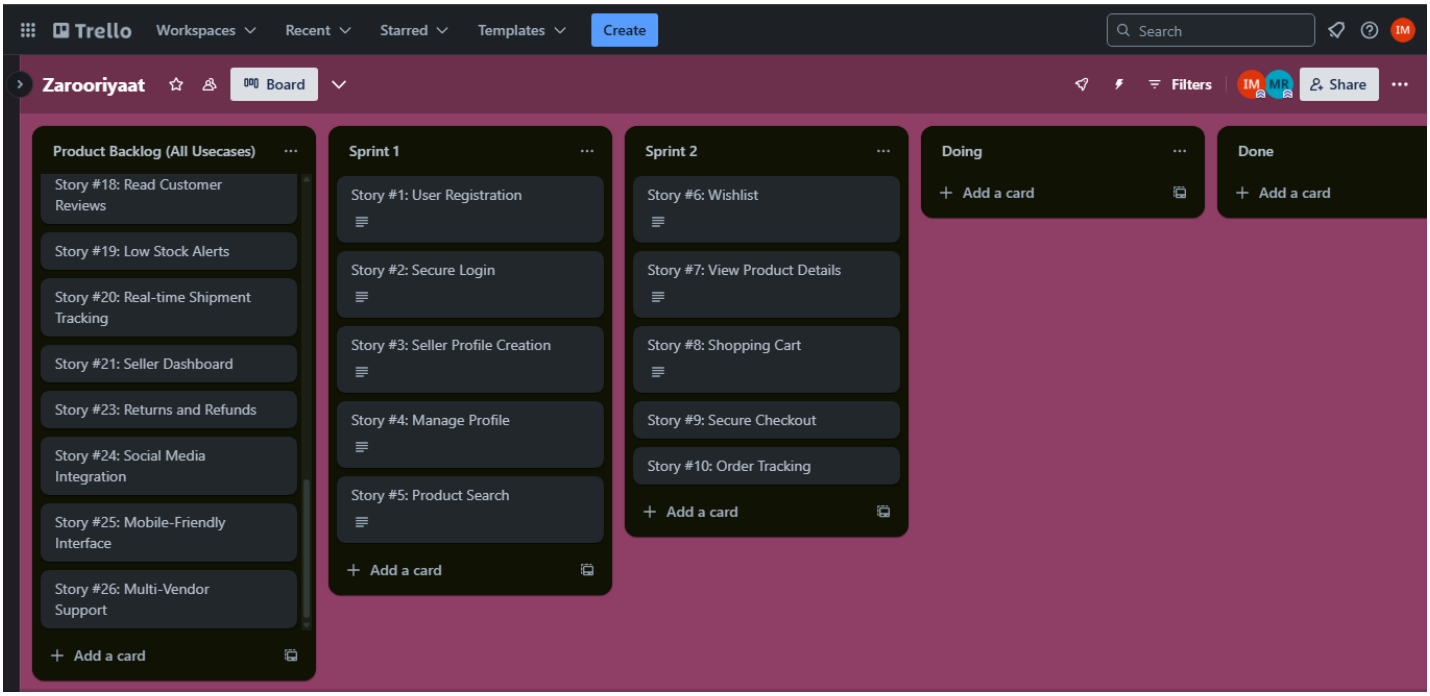
Apply Seller Fee

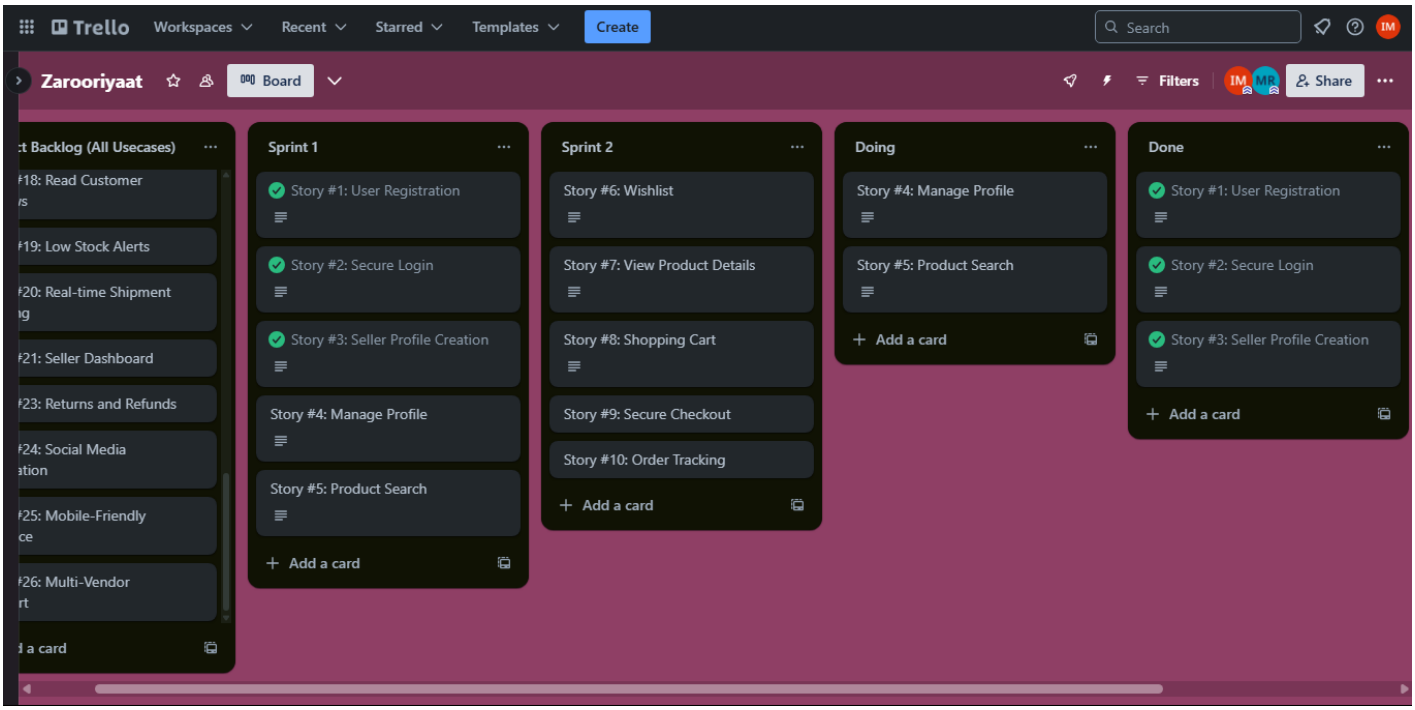


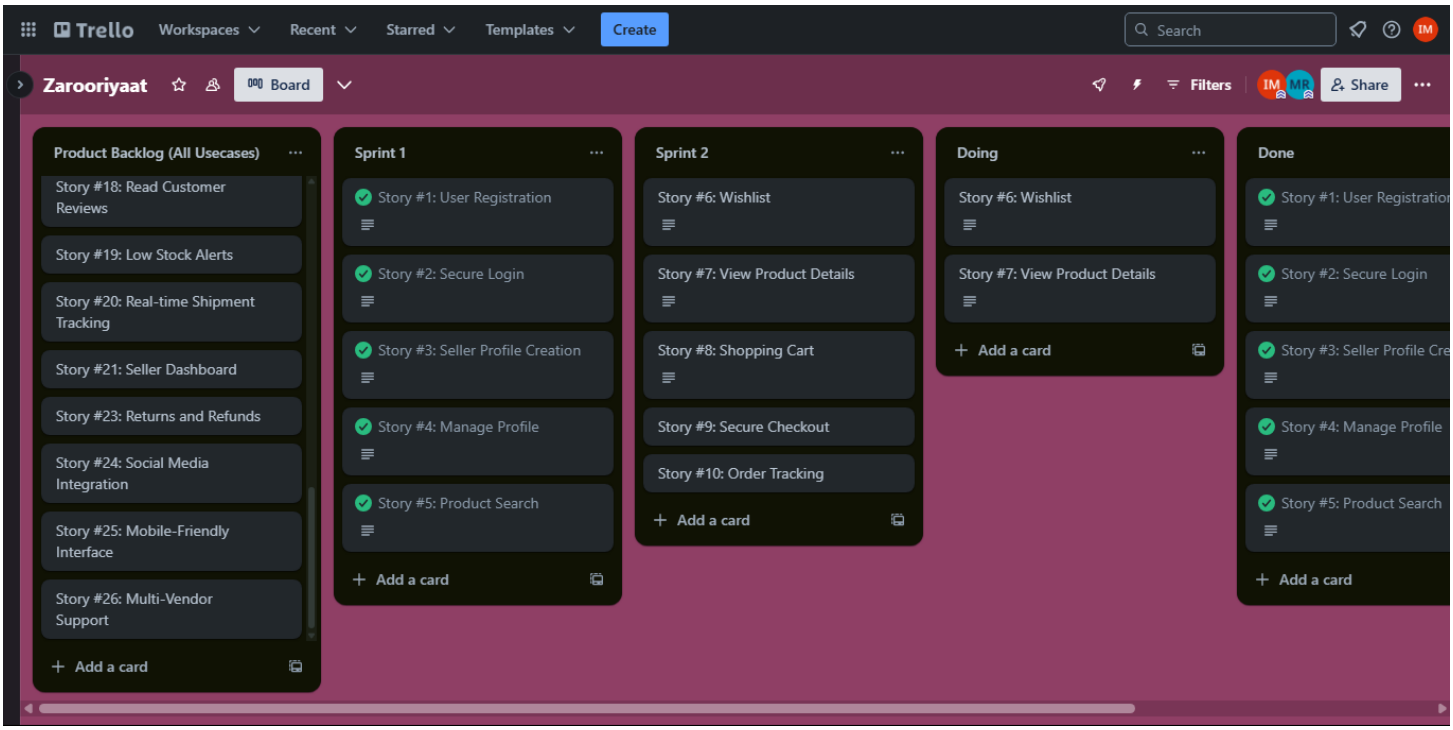
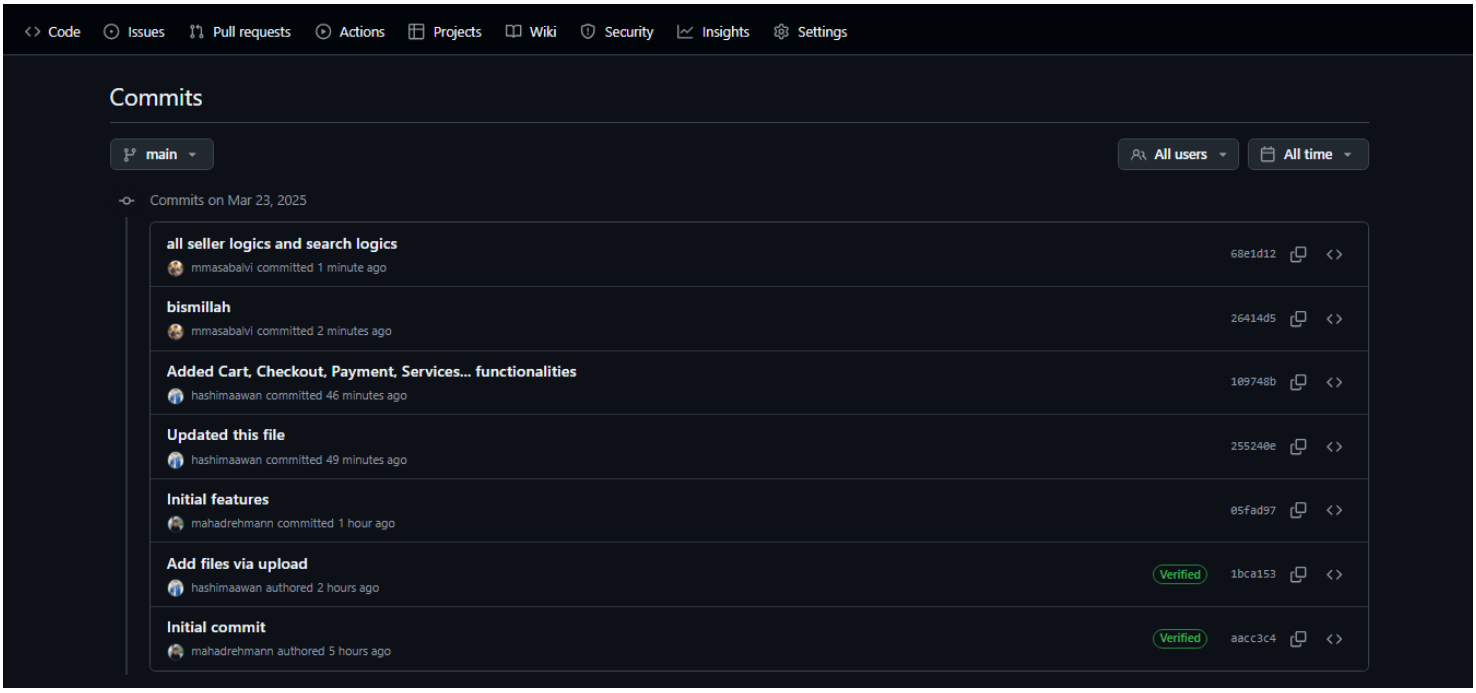
Class Diagram


User
ID
Name
Email
Role
Admin
Permissions
SystemLogs
Alerts
Customer
Address
ContactInfo
Cart
Seller
StoreName
ProductList
Revenue
Email System
EmailID
Sender
Recipient
Payment Gateway
GatewayID
Name
Dashboard
UserMetrics
SalesReports
Account
AccountID
Balance
TransactionHistory
Product Catalogue
List<Products>
Stock
Cart
CartID
Items
TotalPrice
Transaction
TransactionID
Amount
Status
Product
ProductID
Name
Price
sends mail to
sends mails to
receives payments through
makes payments through
adds products to
views
adds items to
manages
accesses
processes
contains
0..*
1

Trello Screenshots







GitHub Screenshots