CSCE 5430: Software Engineering
Sprint-3 Report
Student Trading Connection

Group Number	8

Team Member	Member ID
Kishan Kumar Zalavadia	11685261
Ajay Jatprol	11689329
Gowtham Sai Lendhra Narayanam	11709752
Pavanipriya Vuppalapati	11661784
Mohammed Shakeel	11701737

Scrum Master: Kishan Kumar Zalavadia

GitHub: https://github.com/Kishan-Kumar-Zalavadia/Student_Trading_Connection

GitHub Project Board: https://github.com/users/Kishan-Kumar-Zalavadia/projects/2/views/1

Progress along with the Project Sprints - Sprint 3

During Sprint 3, our team had a clear set of objectives, which primarily included the completion of the next two user stories. We are pleased to report that not only were these goals met successfully, but the sprint also proceeded without any notable deviations from the initial plan.

Planned Objectives:

1. **User Story-5: Bonus Points System**
   Objective: As a student, I want to earn bonus points with every sale or purchase that I make, so that I can boost my profile to levels like gold, silver, or platinum, showcasing my achievements and may unlock rewards benefits.
   Progress: The team successfully executed User Story-5, implementing a bonus points system. Students now earn points with each sale or purchase, enhancing their profiles and may unlock various achievement levels.

2. **User Story-6: Admin Dashboard**
   Objective: As an admin, I want access to an admin page where I can manage user accounts, monitor platform activity, and ensure the platform's smooth operation.
   Progress: User Story-6 was executed seamlessly, providing administrators with a dedicated dashboard for efficient management of user accounts, real-time monitoring of platform activity, and ensuring overall operational smoothness.

Additional Achievements:

1. **Confirmation Options**: User actions like product deletion, and product edit prompt confirmations for added security and clarity.

Deviation from Plan:
We are pleased to report that there were no significant deviations from our planned objectives during Sprint 3. The project progressed smoothly, and the team's efforts aligned with the initial sprint goals.

This successful sprint sets a positive precedent for the project, and we are motivated to continue our development journey in Sprint 3.

Scrum Meetings in Sprint 3: Dates and Summaries:

Scrum Meeting	Date, Time	Medium	Scrum Meeting Summary
Scrum Planning	11/6/2023 – 09:30	Google Meet	1. Discuss how to start the sprint. 2. Discussed ways to make sprint-3 better than sprint-2 by being punctual and working iteratively.
Daily Scrum	11/9/2023 – 09:30	Google Meet	**1. Making updates:** Modifying the existing code to add new code. **2. Analyze the data bugs in existing code if any.** Fixing the minor bugs.
Daily Scrum	11/13/2023 – 09:30	Google Meet	**1. Bugs fixed:** Modifies and improves the existing code. **2. Creating HTTP requests for Bonus points.** Creating the PUT request in Spring Boot.
Daily Scrum	11/15/2023 – 09:30	Google Meet	**1. Done with HTTP request.** **2. Testing the HTTP request.** Testing is done using Postman.
Daily Scrum	11/17/2023 – 09:30	Google Meet	**1. Completed the Testing.** Successfully tested the HTTP requests. **2. Working on user story 5:** Creating a Spring Boot business logic to update the user's bonus points.
Daily Scrum	11/21/2023 – 09:30	Google Meet	**1. Done with Business Logic for User Story 5:** Also tested using Postman. **2. Working on the angular service for bonus points.** Created an angular service that talks to the spring boot controller to update the bonus points for a user when he buys a product.
Daily Scrum	11/23/2023 – 09:30	Google Meet	**1. Done with angular service for user story 5:** Also done manual testing. **2. Working on admin page business logic** Created HTTP request for admin page and codded the business logic.
Daily Scrum	11/24/2023 – 09:30	Google Meet	**1. Done with backend for user story 6:** Also tested using Postman **2.Working on angular service for the admin page.** Creating Admin component.
Daily Scrum	11/27/2023 – 09:30	Google Meet	**1.Created the angular component for the admin page.** This include html, css, and ts files. **2.Creating admin service** Creating an admin service which will interact with Backend.
Daily Scrum	11/28/2023 – 09:30	Google Meet	**1.Done with user story 6.** Created and Tested the admin services. **2.Final Project Integration** Performing the final integration.
Sprint-3 Review	11/29/2023 – 09:30	Physical	**3.Done on complete project integration.** **4.Product manager review:** Got a positive review from the product manager.

Tasks assigned to each member:

All the tasks below are self-assigned tasks based on one's interest.

Kishan Kumar	Ajay	Gowtham	Pavanipriya	Shakeel
Analyze the Bonus point storage in the database.	HTTP Bonus Point Testing: Execute rigorous HTTP tests to affirm the bonus point system's functionality and adherence to requirements.	HTTP API testing	Logo designing	Created a GET request which accepts user ID
Creating a POST request for Bonus Points. Backend Spring boot HTTP request is created.	MVC and Docker Architecture: Craft precise MVC and Docker architecture diagrams to delineate the system's blueprint.	Working on CSS	Testing of the project	Created a POST request which accepts bonus points
Creating a GET request for Bonus Points. Backend Spring boot HTTP request is created.	Logo Creation: Conceptualize and design a distinct logo that reflects the project's essence.	Testing the project workflow	Understanding the interrelation between Database tables	Worked on updating the profile level based on bonus points.
Creating a PUT request for Bonus Points. Backend Spring boot HTTP request is created for bonus point update.	Database Setup for Bonus Points: Establish the bonus point database for secure and effective data handling.			
Get request access in angular service i.e., frontend.	All-Encompassing Project Evaluation: Thoroughly test the project to assure its comprehensive performance and stability.			
Post request access in angular service i.e., frontend.	Deployment Plan Development: Formulate an explicit deployment guide, ensuring a smooth transition to production			
Create POST request for admin – Spring Boot				
Create PUT request for admin – Spring Boot				
Create GET request for admin – Spring Boot				
Create admin Component				
Designing front-end for the admin page.				
Users can edit Profiles. Created angular and Spring boot request to handle user edits.				
HTTP request testing using Postman.				
Project Integration to ensure all the functions are properly working.				

Sprint-3 Milestones: User Stories and Descriptions:

Project Workflow:

The implementation of the user story involved a comprehensive project workflow:

1. **Technology Stack**: Angular for the frontend, Spring Boot for the backend, and MySQL for the database.

2. **Pages and Features**: Different pages were developed within each portal to support specific actions:
   - **All User Page:**
     Description: Admin-exclusive page showcasing all users. Enables admins to view, edit, and delete user profiles for efficient user management.
   - **All Product Page:**
     Description: Admin and seller page for comprehensive product management. Admins can view, edit, and delete products, ensuring accurate and up-to-date listings.
3. **Data Flow:**
   - **Angular Frontend**: Handles the user interface and user interactions.
   - **Spring Boot Backend**: Manages the application logic and data processing.
   - **MySQL Database**: Stores and retrieves data.

4. **Data Communication:**
   - **HTTP Requests**: Spring Boot communicates with the Angular frontend through HTTP GET POST, PUT, PATCH, and DELETE requests.

5. **Controllers and Services:**
   - **Controllers**: Spring Boot controllers handle HTTP requests and route them to the appropriate services.
   - **Services**: Spring Boot services manage business logic and interact with external data sources, including the MySQL database.

6. **Workflow:**
   - Data flows from the user interface in Angular to Spring Boot using HTTP requests.
   - Spring Boot controllers route requests to the relevant services.
   - Spring Boot services handle business logic and interact with the MySQL database.
   - MySQL Database stores and retrieves data efficiently.
   - The processed data is presented in the frontend UI, allowing users to interact with the system.

User Story-5:

As a student, I want to earn bonus points with every sale or purchase to boost my profile to levels like gold, silver, or platinum, showcasing my achievements and may unlock rewarding benefits.

Implementation:

- **Frontend Interface:**
  - The user profile section now prominently displays earned bonus points and the corresponding level (gold, silver, platinum).
  - Bonus points are visually represented within the user's profile, offering a clear indication of the user's progress.
  - Users can observe their bonus points and level without the use of a progress bar, ensuring a seamless and intuitive viewing experience.

- **Data Interaction:**
  - Bonus points are calculated and updated in real time based on completed sales or purchases.
  - Backend logic dynamically determines the student's level, reflecting their achievements on the platform.

Explanation:

- The implementation introduces a gamification element, motivating students to actively engage in platform activities to achieve higher levels and unlock associated benefits.
- It enhances the user profile as a dynamic and personalized space, reflecting the user's progress and accomplishments.
- The bonus points system encourages active participation, contributing to a vibrant and engaged user community.

User Story-6:

As an admin, I want access to an admin page where I can manage user accounts, monitor platform activity, and ensure the platform's smooth operation.

Implementation:

- **Frontend Implementation:**
  - An exclusive admin page was developed with sections for user account management, activity monitoring, and overall platform health.
  - User Management: Admins can view, edit, and delete user accounts directly from the admin page, streamlining administrative tasks.
  - Activity Monitoring: Real-time charts or graphs display platform activity trends, helping admins identify patterns and potential issues.

- **Data Integration:**
  - **User Accounts**: Admin actions on the front end trigger requests to the backend to update user accounts or retrieve user-related data.
  - **Platform Metrics**: Data regarding platform activity is continuously collected and presented on the admin page, enhancing the admin's ability to make informed decisions.

Explanation:

- **Comprehensive User and Product Tracking**: The backbend's dedicated SQL table plays a crucial role in tracking and maintaining records of all users and products purchased within group purchases, ensuring transparency and organized group participation.
- **Seller-Customer Interaction:** The system allows sellers to view and interact with customers who have purchased their products through group purchases. This feature promotes effective communication between sellers and their customers, fostering a sense of community and trust.
- **Seamless Group Purchase Process**: The implementation emphasizes the ease and convenience of buying a product within a group purchase. Students can do so from the product details page, making it as straightforward as purchasing any regular product.
- **User-Friendly Experience**: By indicating group eligibility and providing a direct process for participation, the platform enhances the user experience, making it hassle-free for students to access educational resources at a lower cost.

GitHub Project Page:

GitHub Project Board: https://github.com/users/Kishan-Kumar-Zalavadia/projects/2/views/1

List of Issues worked on:

- Completed Issues in Sprint-3:
  https://github.com/Kishan-Kumar-Zalavadia/Student_Trading_Connection/issues?q=is%3Aissue+is%3Aclosed+milestone%3ASprint-3

Database Schema:

Table 1: User

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
address	varchar(255)	YES		NULL	
email_id	varchar(255)	YES		NULL	
password	varchar(255)	YES		NULL	
phone	varchar(255)	YES		NULL	
profile_level	varchar(255)	YES		NULL	
user_name	varchar(255)	YES		NULL	
bonus_points	int	NO		NULL	
user_type	int	NO		NULL	

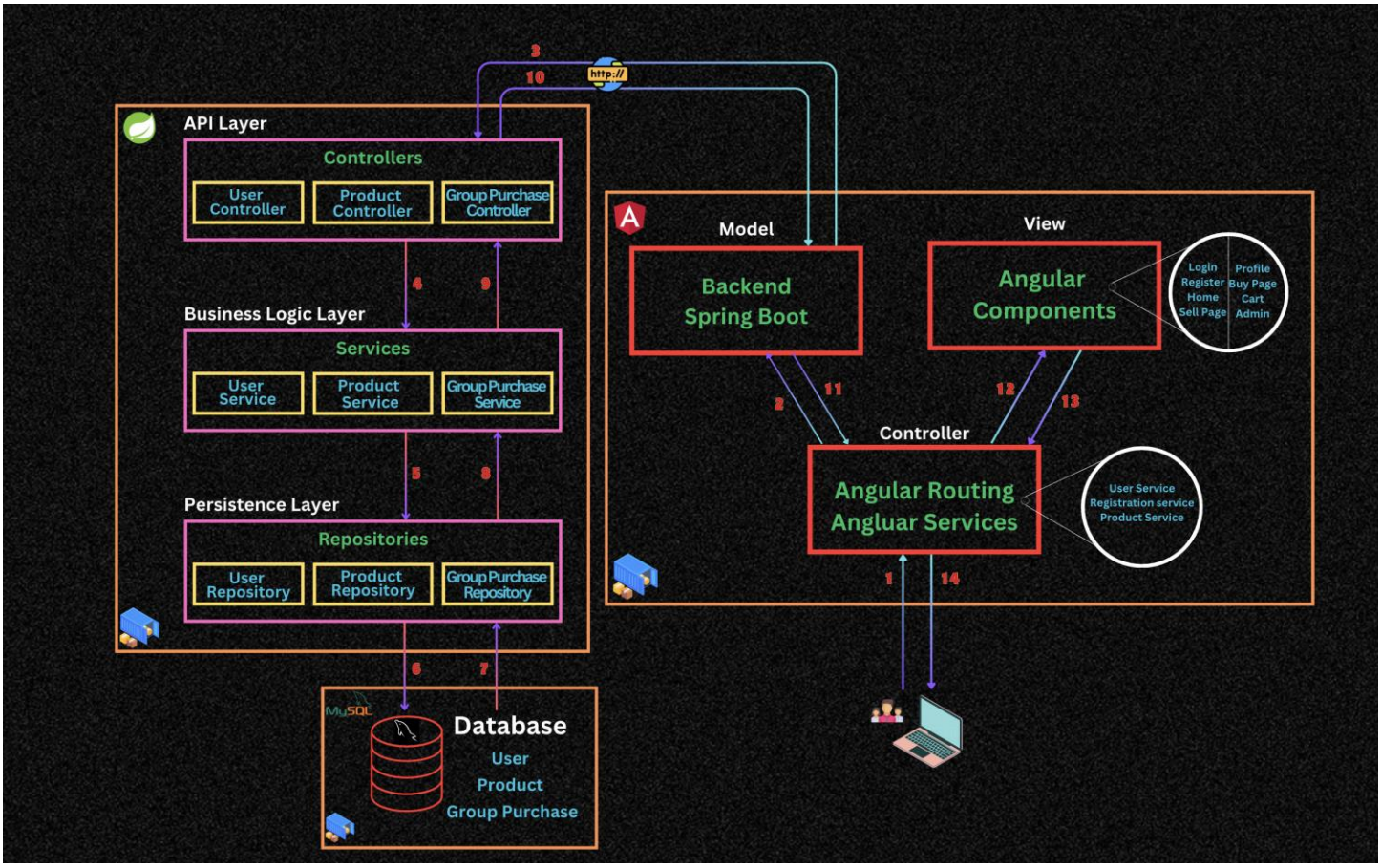
Table 2: Product

Field	Type	Null	Key	Default	Extra
productid	int	NO	PRI	NULL	auto_increment
buyerid	int	NO		NULL	
category	varchar(255)	YES		NULL	
description	varchar(255)	YES		NULL	
is_group_purchase	int	NO		NULL	
price	int	NO		NULL	
sellerid	int	NO		NULL	
status	int	NO		NULL	
title	varchar(255)	YES		NULL	

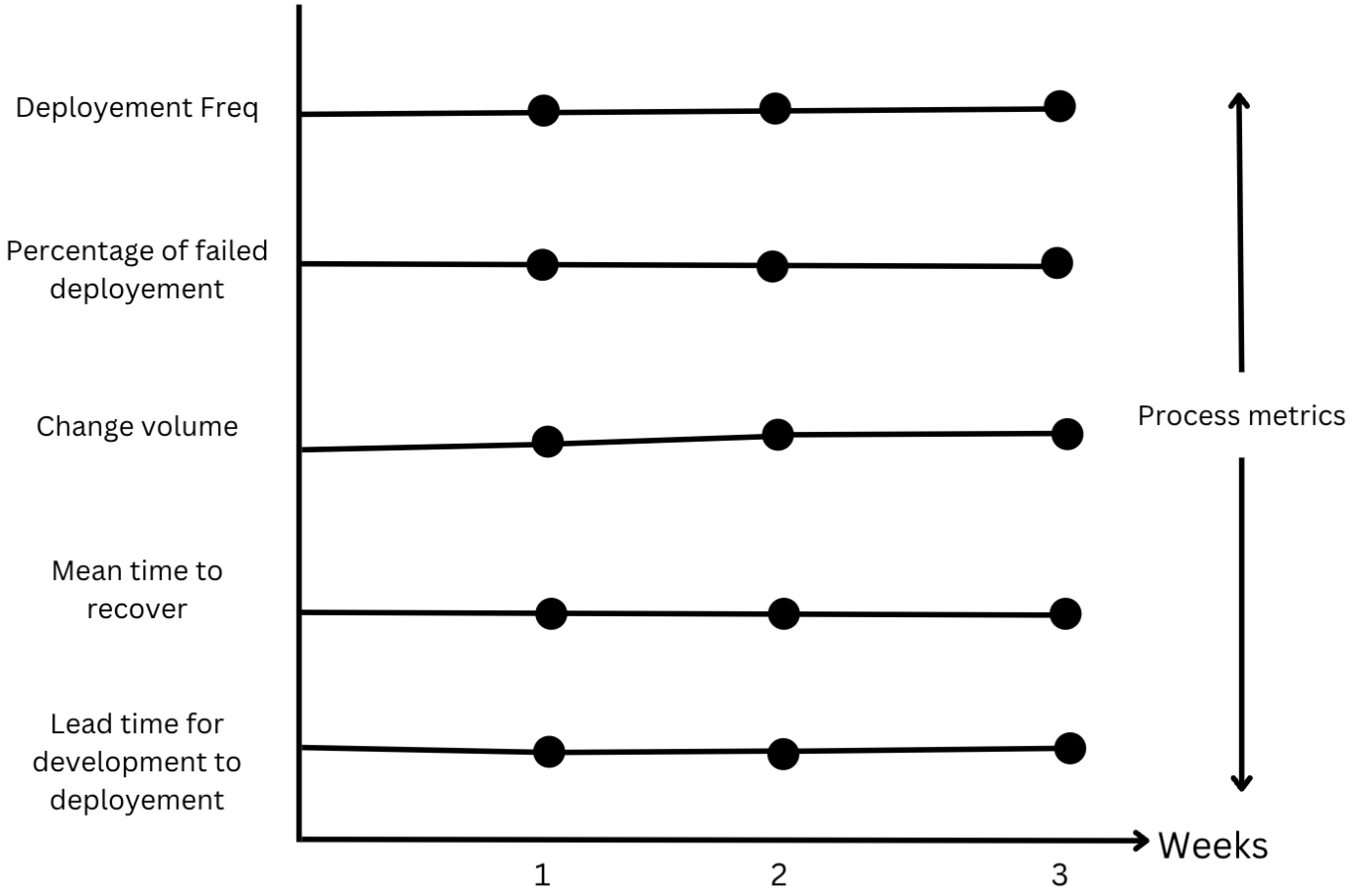
Table 3: Group Purchases

Field	Type	Null	Key	Default	Extra
group_purchase_id	int	NO	PRI	NULL	auto_increment
buyerid	int	NO		NULL	
productid	int	NO		NULL	

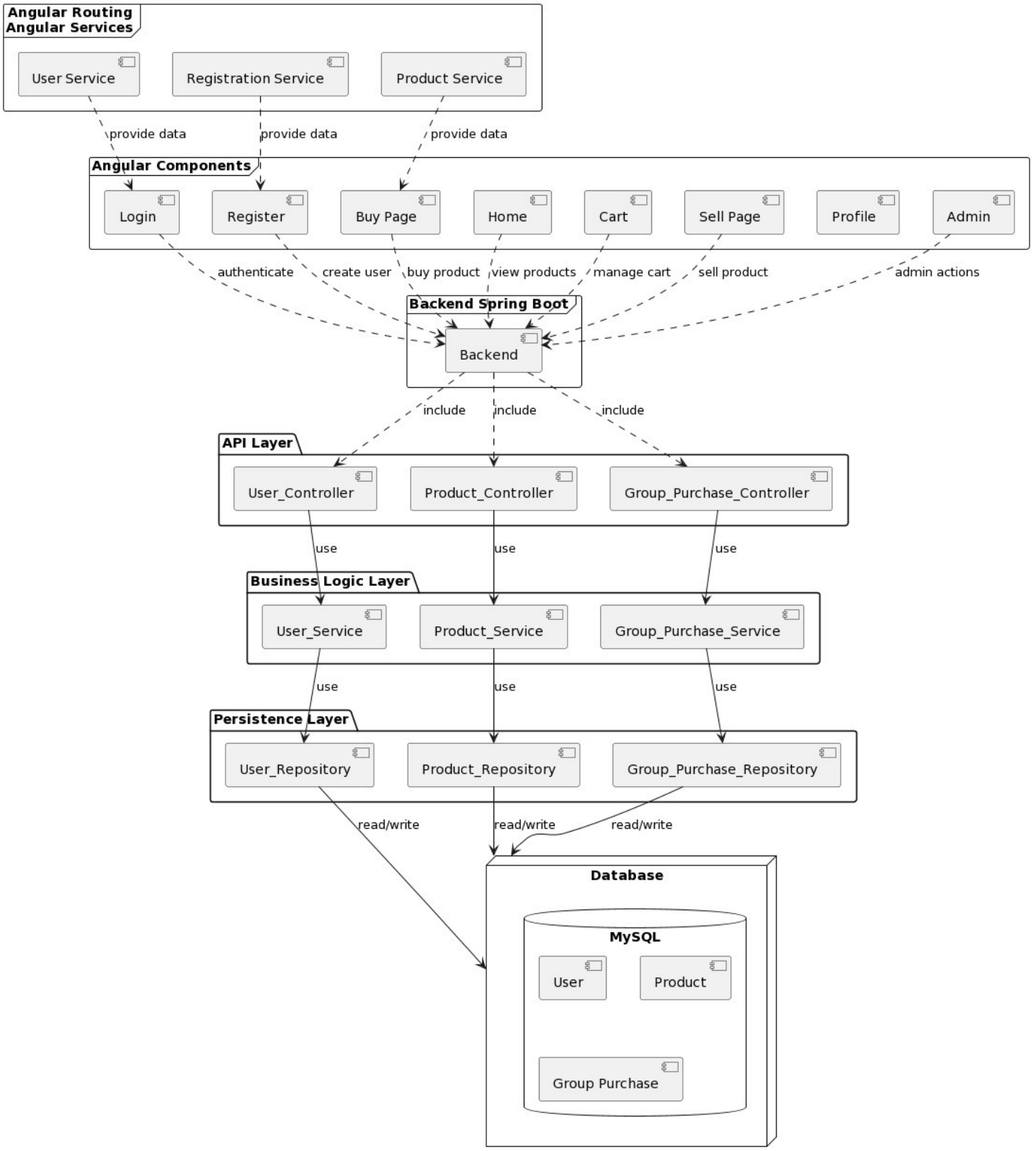
Architecture:


3
10
http://
API Layer
Controllers
User Controller
Product Controller
Group Purchase Controller
4
9
Business Logic Layer
Services
User Service
Product Service
Group Purchase Service
5
8
Persistence Layer
Repositories
User Repository
Product Repository
Group Purchase Repository
6
7
MySQL
Database
User
Product
Group Purchase
Model
Backend Spring Boot
View
Angular Components
Login
Register
Home
Sell Page
Profile
Buy Page
Cart
Admin
2
11
12
13
Controller
Angular Routing
Angluar Services
User Service
Registration service
Product Service
1
14

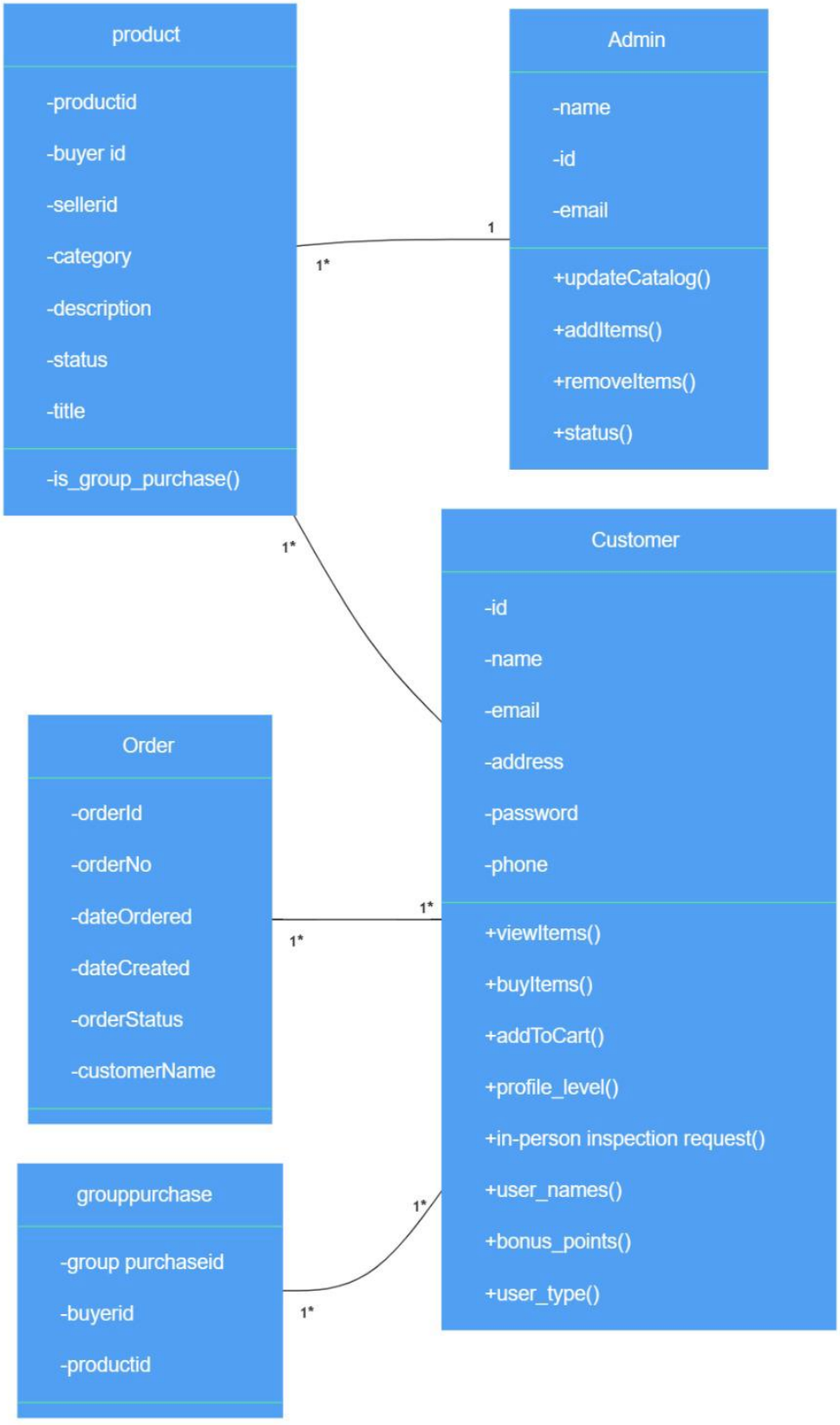
Metrics:



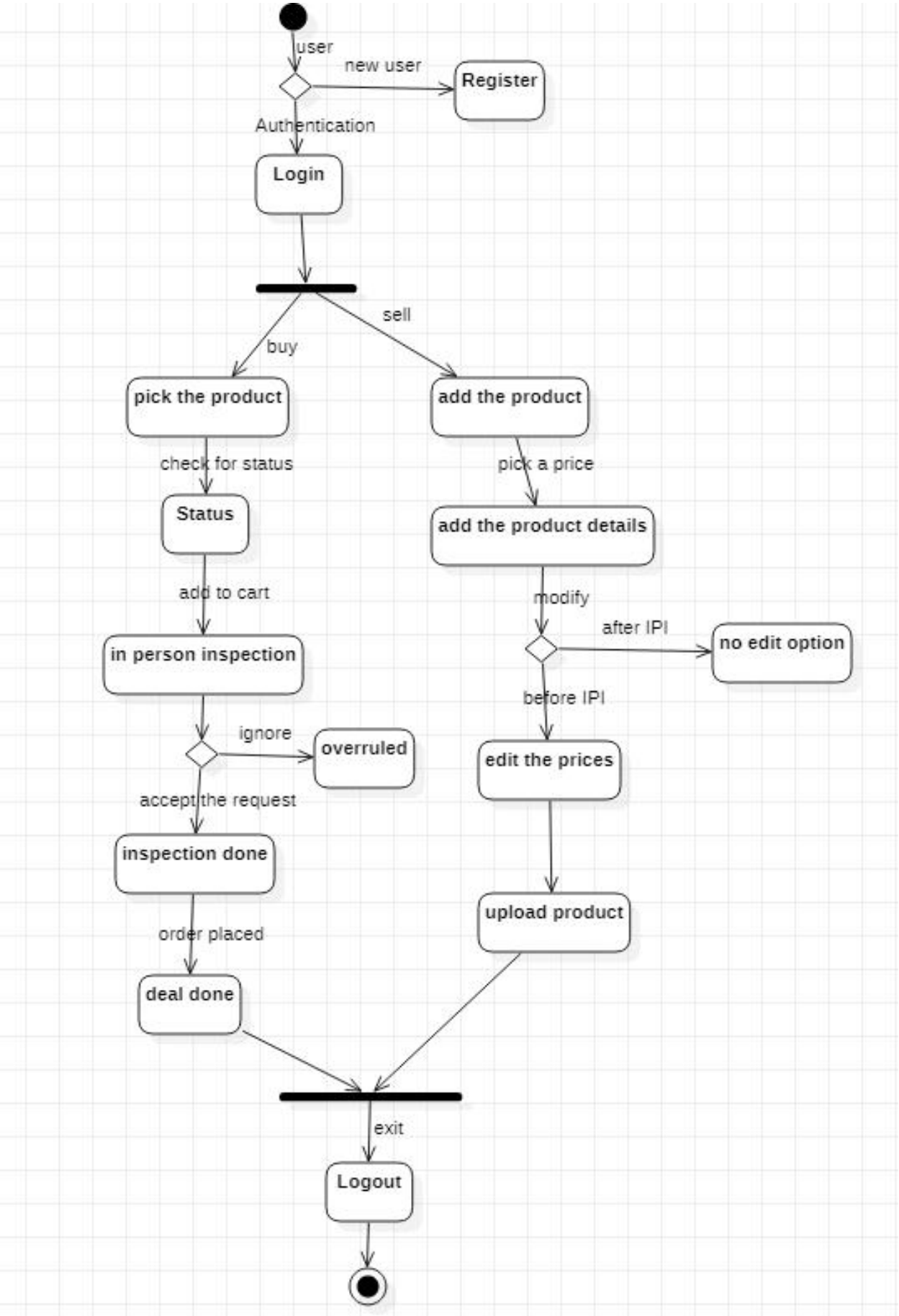
UML Diagrams:



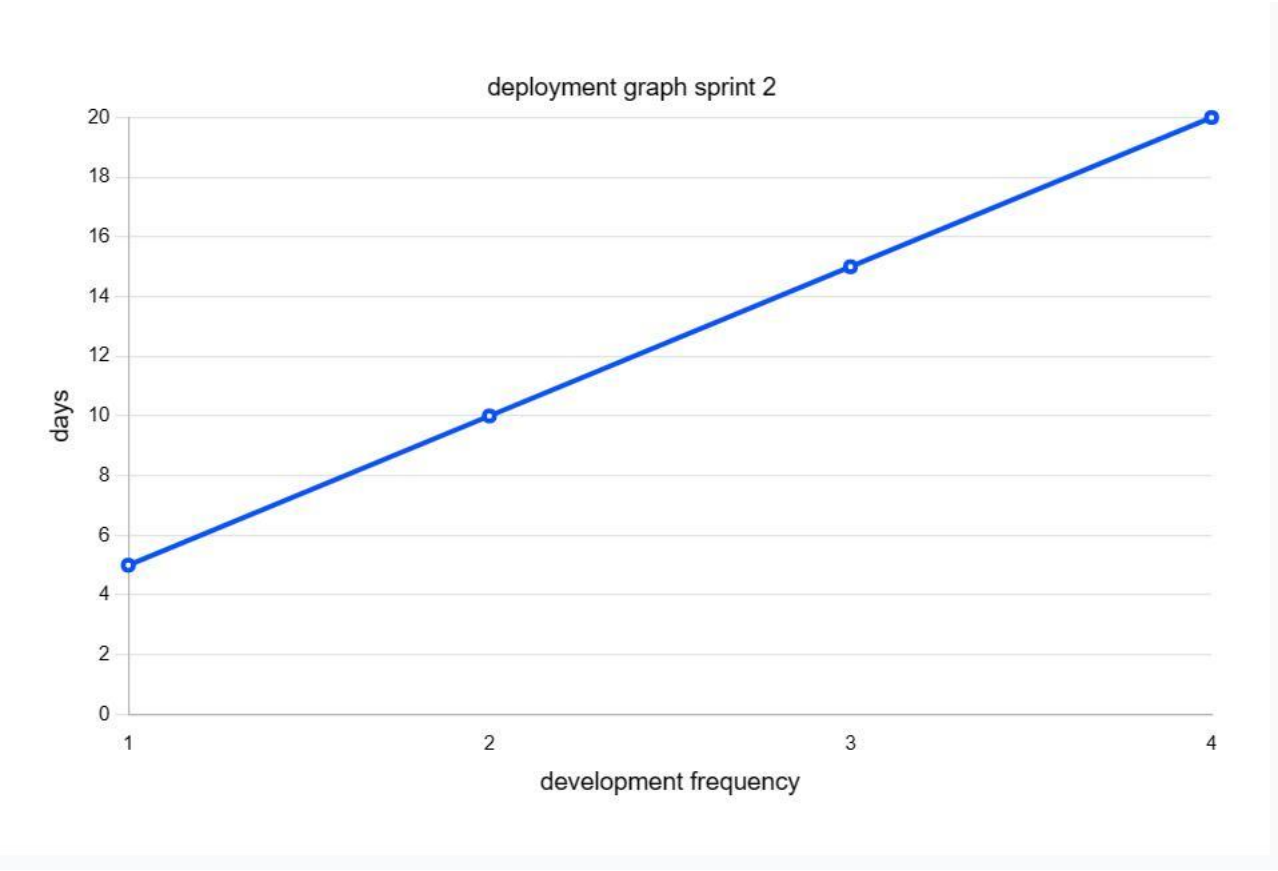
Class Diagram:



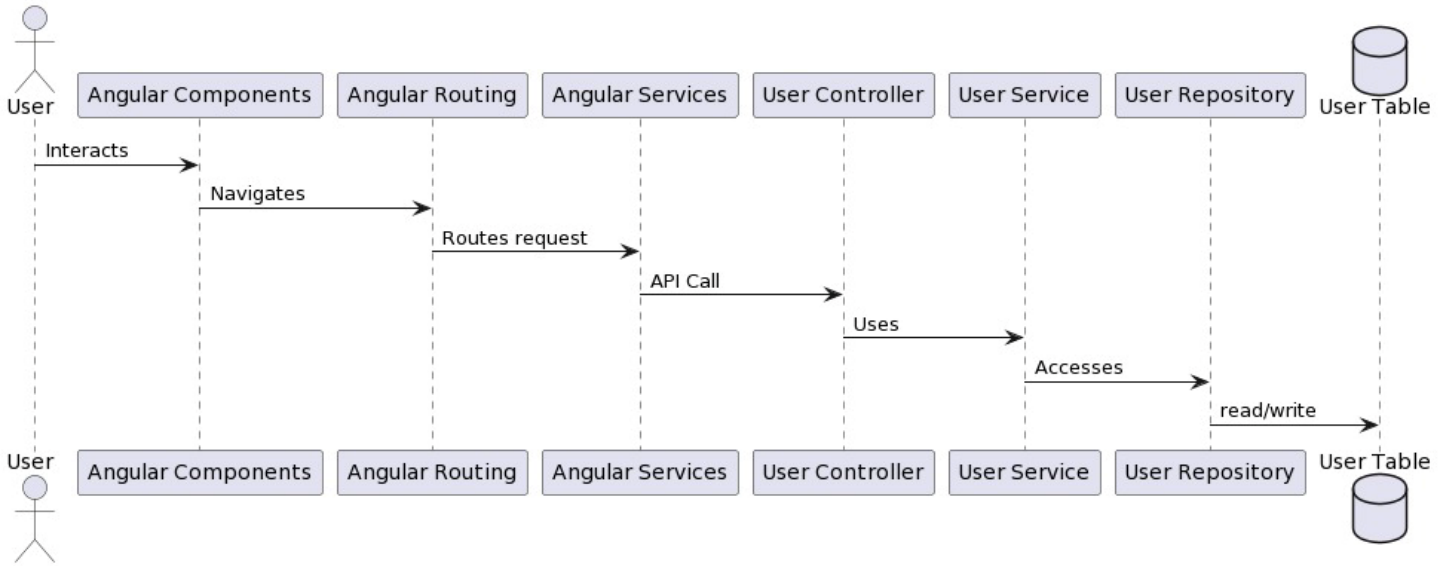
Use case Diagram:



Development Graph:

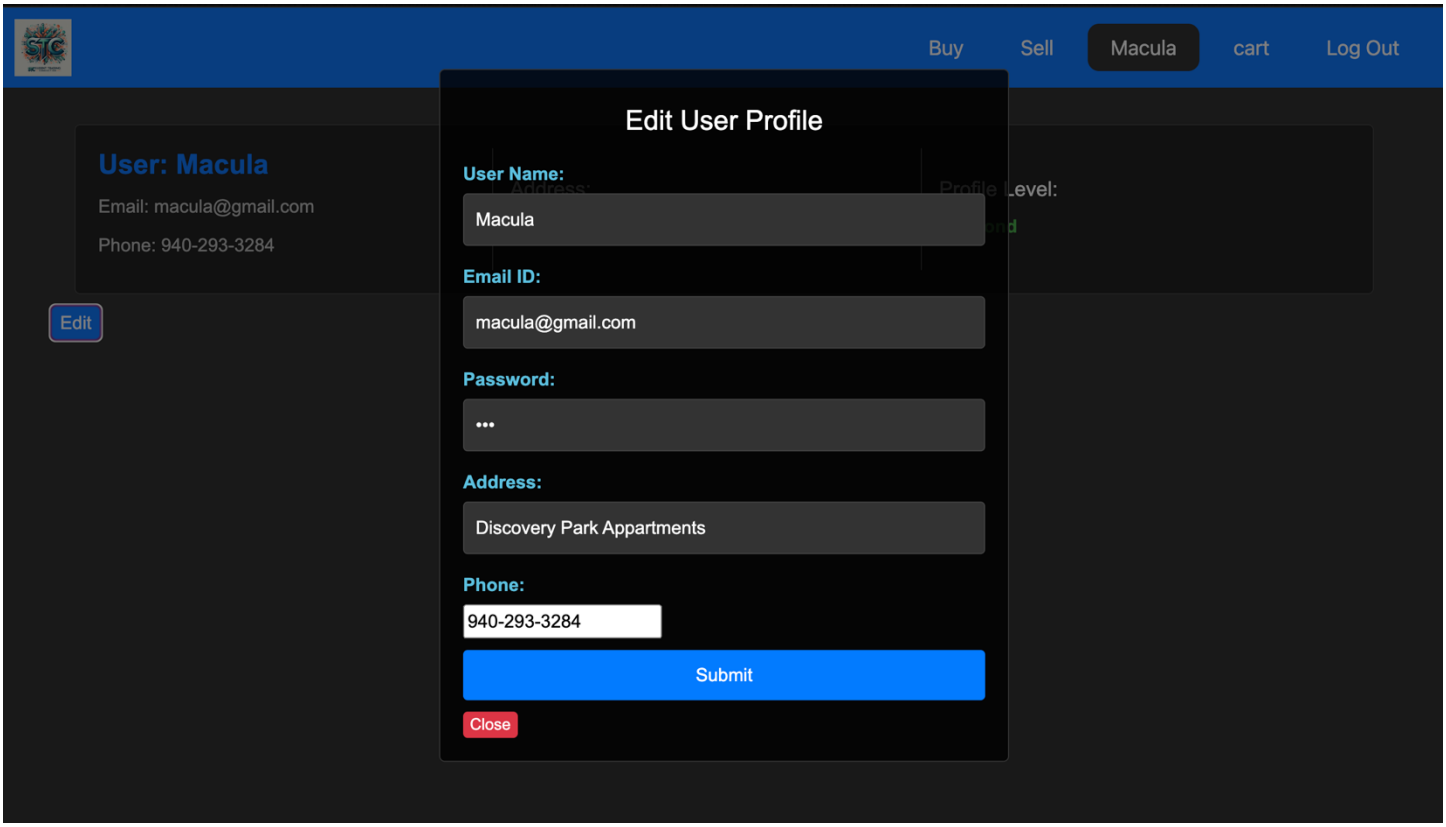


Workflow Diagram:

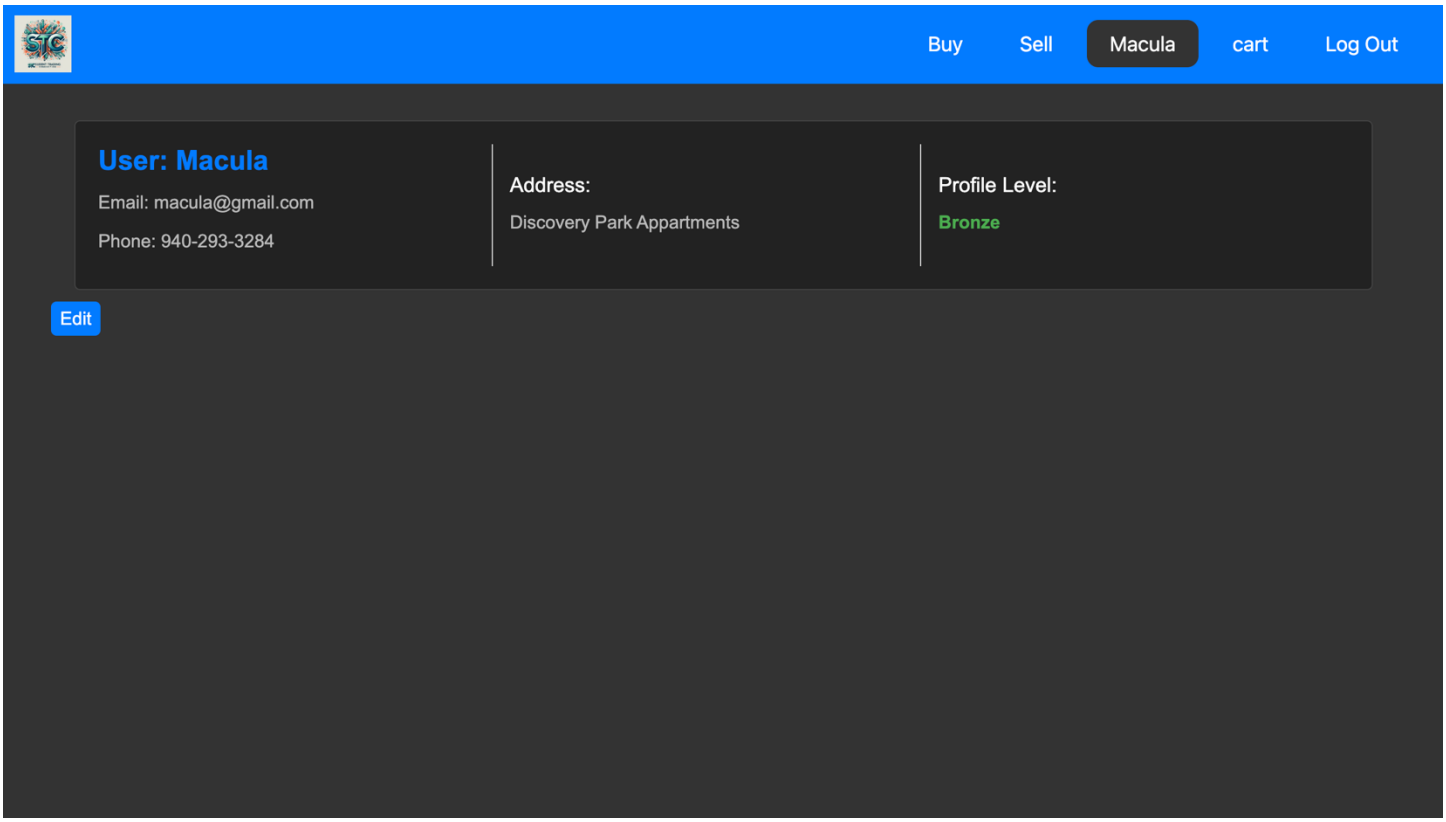


Demo Project Images:

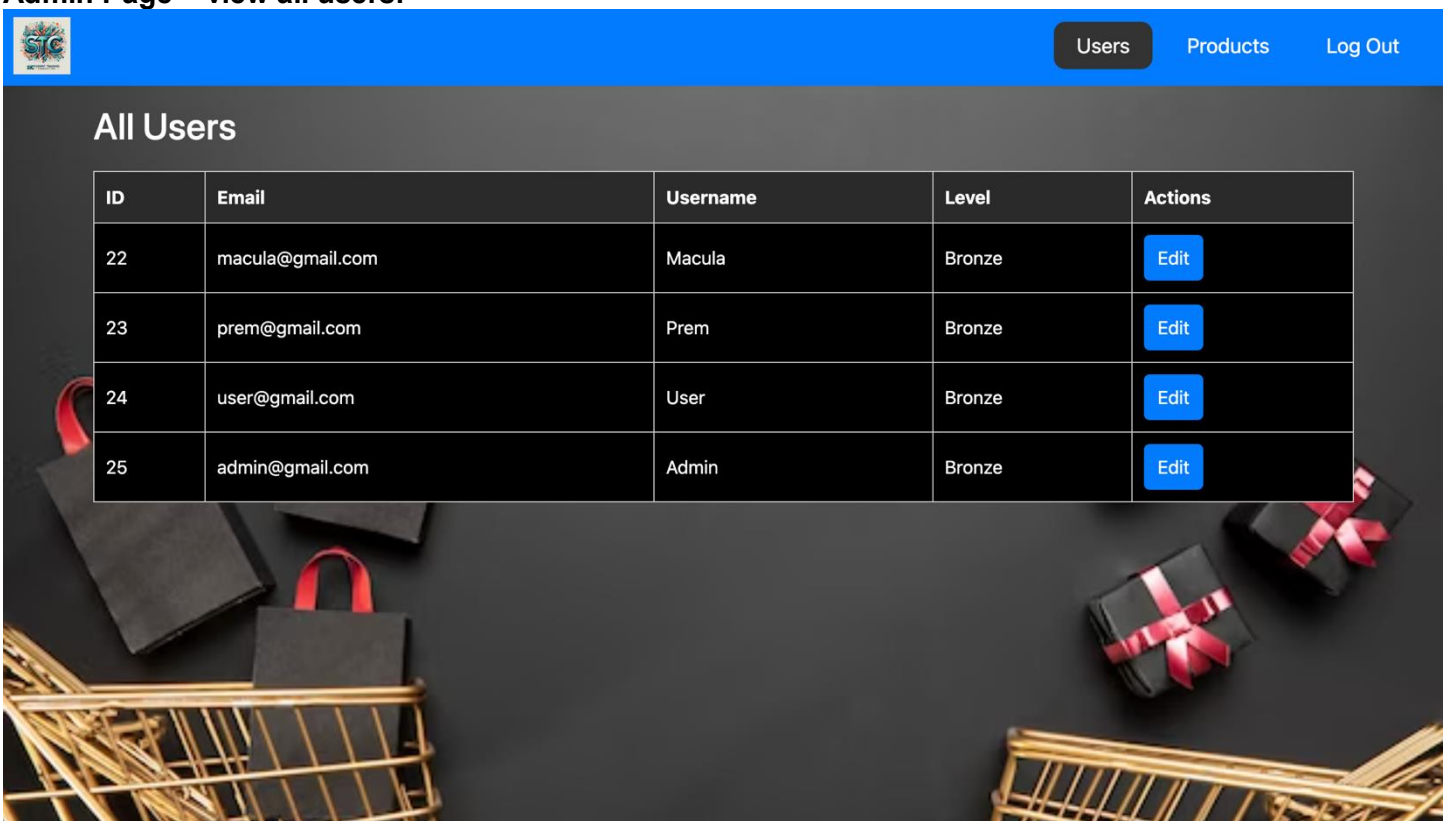
Edit User Profile



User Profile Level:

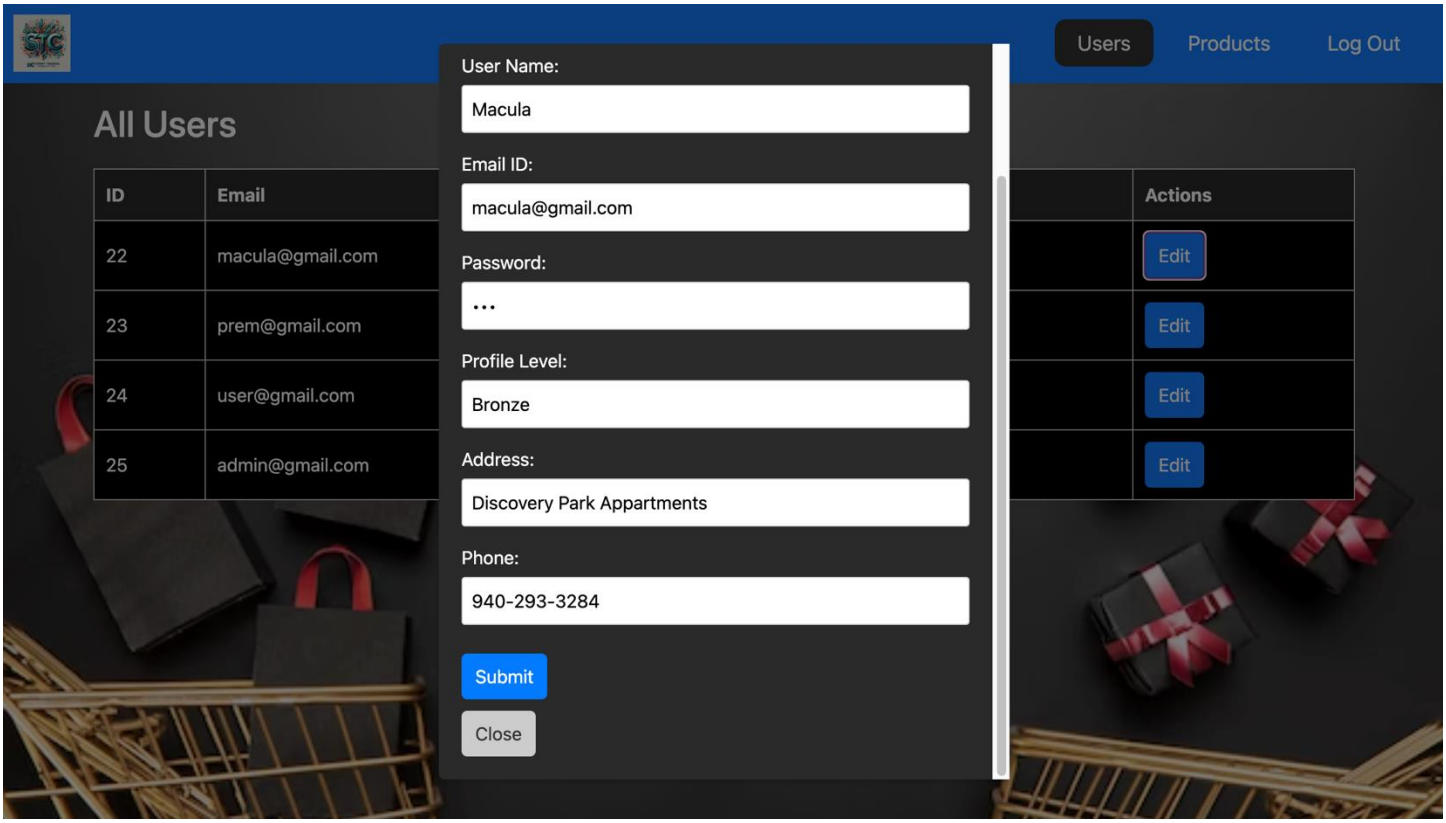


Admin Page – view all users:



Admin Page – view all products:

Admin Page – Edit users



Admin Page - Edit Product: