Assignment 2: Documentation

M.A. Jayaratne - CB014637

Staffordshire University MSc - Information Technology Management

Use Case

Use Case ID: 1.1

Use Case Name: Shipment Status Updates for Customer

Actors: Driver, Customer

Steps:

1. Driver logs into the web app and selects a shipment.
2. Driver updates the status
3. Backend processes the update and triggers notifications.
4. Customer receives an email/SMS with the updated status.

Postcondition: Customer is informed of the shipment status in real time.

Design Decisions

Frontend – React.js

React.js was chosen because of its component-based architecture and reusability

It's free and easy to learn

Used Axios for API calls making sure a seamless communication with the backend

React enables real time updates without requiring full page reload. This is important for displaying real-time shipment status updates to customers.

React supports responsive design which work well on both desktop and mobile devices.

Backend – Spring Boot

The reason for selecting Spring Boot was its scalability and its easy to integrate with MySQL

To handle CRUD operations and notifications RESTful APIs were designed.

Spring boot is easy to integrate with email gateways

Spring Boot processes status updates from drivers and triggers notifications to customers in real-time.

Database – MySQL

MYSQL is reliable and it’s easy to use with spring boot

MySQL is easy to set up and integrates well with Spring Boot using Spring Data JPA.

Technical Approach

Frontend (React.js)

Developed reusable components for shipment tracking and status updates. So that drivers can login to the web application and change the status of the shipment to PENDING, IN_TRANSIT, DELIVERED and CANCELLED

Implemented API request functions using Axios for handling API calls to the backend

Backend

Implemented controllers for handling HTTP requests.

Created REST APIs for shipment management and notifications.

Implemented services to handle business logic

Integrated with JavaMail (Email) for notifications

Implemented Entities and Repositories for database modeling and operations

Database

Designed tables for shipments, customers, drivers, and notifications.

Used JPA for ORM (Object-Relational Mapping) in Spring Boot.

Testing Methods

Unit Testing

End-to-End Testing

Challenges Faced

Limited familiarity with the technologies.

Additional Features and Non-Functional Requirements

Dashboards for managing customers and drivers

Dashboard for analytics

Responsive design for mobile and desktop access.

Way Forward

Mobile App:

Mobile App for drivers and customers can be developed

SMS Notifications:

SMS notifications can be implemented

Whole System Integration:

This can be integrated with other use cases