

Software Requirements Specification

for

Restaurant Food Ordering System

Prepared By

Saheli Das

Harshavardhan V

Rithik Voolla

Tadepalli Yaswanth

Table of Contents

Table of Contents

Revision History

1. Introduction

1.1 Purpose

1.2 Scope

1.3 References

1.4 Overview

2. Overall Description

2.1 Product Perspective

2.2 Product Features

2.3 User Classes and Characteristics

2.4 Operating Environment

2.5 Design and Implementation Constraints

2.6 Assumptions and Dependencies

3. System Features

3.1 System Feature 1

3.2 System Feature 2 (and so on)

4. External Interface Requirements

4.1 User Interfaces

4.2 Hardware Interfaces

4.3 Software Interfaces

4.4 Communications Interfaces

5. Other Nonfunctional Requirements

5.1 Performance Requirements

5.2 Safety Requirements

5.3 Security Requirements

5.4 Software Quality Attributes

6. Other Requirements

Appendix A: Glossary

Appendix B: Analysis Models

Appendix C: Issues List

Revision History

Name	Date	Reason For Change	Version

1. Introduction

1.1 Purpose

The purpose of this SRS document is to define the requirements and specifications for developing a food delivery application. It aims to outline the functionality, features, and constraints of the system to ensure a clear understanding of the project's objectives.

1.2. Scope of Project

The food delivery application will provide a platform for users to browse restaurants, view menus and place orders. It will also include features for restaurant owners to manage their menus, orders, and deliveries. The application will target both customers and restaurants, offering a seamless and efficient food ordering and delivery experience.

1.3 References

- C. Larman, APPLYING UML AND PATTERNS An Introduction to Object-Oriented
- Analysis and Design and Iterative Development, 3rd ed., Massachusetts: Pearson Education, 2005.
- D. Carrington, CSSE3002 Course Notes, School of ITEE University of Queensland, 2008
- IEEE Recommended Practice for Software Requirements Specifications, IEEE Standard 830, 1998.

1.4 Overview

The food delivery application responds to the rising need for convenient and reliable food ordering and delivery services. It provides users with a user-friendly platform to browse menus, browse restaurants, and place orders seamlessly. For restaurants, the application offers efficient menu management, order processing, and delivery person management. The aim of the application is to enhance the overall food delivery experience for both users and restaurants by ensuring convenience, efficiency, and satisfaction.

2. Overall Description

2.1 Product Perspective

The food delivery application described in this SRS operates within a dynamic and interconnected ecosystem, seamlessly integrating users, restaurants, delivery personnel, and administrative functionalities. This comprehensive system not only ensures data consistency and security but also facilitates smooth interactions between stakeholders. With a design focused on scalability, adaptability to technological advancements, and compliance with industry standards, the application offers a robust and reliable solution for efficient food ordering and delivery services.

2.2 Product Features

The food delivery application offers a range of key features to enhance the user experience and streamline operations for restaurants:

- **User-friendly Interface:** An intuitive interface for users to easily browse restaurants, view menus and place orders.
- **Restaurant Management:** Tools for restaurants to manage menus, update item availability, track orders, and assign delivery personnel.
- **Payment Integration:** Seamless integration with payment gateways for secure and convenient online transactions.
- **Admin Dashboard:** A comprehensive dashboard for administrators to monitor system performance, manage user accounts, and analyze data for insights and decision-making.

These features collectively ensure a smooth and enjoyable food ordering and delivery experience for both users and restaurants, promoting customer satisfaction and operational efficiency.

2.3 User Classes and Characteristics

Admin:

- Has administrative privileges and access to the backend system.
- Can manage user accounts, restaurant profiles, and system settings.
- Responsible for monitoring system performance, resolving issues, and ensuring compliance with regulations.

- May generate reports, analyze data, and make strategic decisions to improve the platform.

Customer:

- Represents the end-users who use the application to order food.
- Can create and manage their profiles, including personal information and payment methods.
- Browses restaurants, menus, and places orders for delivery or pickup.
- Tracks order status, receives notifications, and provides feedback on the service and food quality.
- May access loyalty programs, discounts, and promotions offered by the platform.

Restaurant:

- Represents the businesses registered on the platform to showcase their menus and accept orders.
- Manages their restaurant profile, including menu items, pricing, availability, and promotions.
- Receives and processes orders, manages order fulfillment, and assigns delivery personnel.
- Monitors performance metrics, customer feedback, and ratings to improve service quality.
- Utilizes tools provided by the platform for marketing, analytics, and customer engagement

2.4 Operating Environment

The food delivery management system is designed to operate in a distributed environment leveraging Spring Boot for backend services, MySQL for data storage, and React for the frontend user interface.

2.5 Design and Implementation Constraints

The system's design is constrained by the chosen technology stack of Spring Boot, MySQL, and React. Any limitations or updates in these technologies could potentially affect system functionality and performance. It must also support parallel operations without introducing scalability issues for the number of connected devices, ensuring a seamless experience for users across various devices. Additionally, the system must

exhibit high reliability, capable of running crash-free indefinitely or recovering from errors without impacting end-user experience, thereby maintaining a smooth and glitch-free operation.

2.6 Assumptions and Dependencies

- **Assumption: Availability of Internet Connectivity**

The system assumes that users, restaurants, and delivery personnel have access to reliable internet connectivity for seamless communication and real-time updates.

- **Assumption: User Device Compatibility**

The system assumes that users' devices (e.g., smartphones, tablets, computers) meet minimum compatibility requirements for accessing the web-based application built with React.

- **Assumption: Database Reliability**

The system relies on the assumption of consistent and reliable performance of the MySQL database for storing and retrieving data related to user profiles, orders and menus.

- **Assumption: User Input Accuracy**

The system assumes that users provide accurate and valid information during registration, order placement, and payment processes to ensure smooth operations and data integrity.

- **Assumption: User Training and Support**

The system assumes that users, especially restaurant owners and administrators, receive adequate training and support to effectively use the platform's features and functionalities.

- **Dependency: Framework Updates**

The system's dependency on frameworks like Spring Boot and React necessitates regular updates and maintenance to incorporate new features, security patches, and performance improvements.

- **Dependency: Development Tools and Libraries**

The system depends on development tools such as IDEs, version control systems, build automation tools, and front-end libraries for efficient development and deployment.

- **Dependency: Hosting and Infrastructure**

The system's availability and performance depend on the reliability and scalability of the hosting infrastructure, including servers, cloud services, and network infrastructure.

3. System Features

3.1 Authentication and Authorization:

- Users (Admin, Restaurant, Customer, Delivery Person) can sign up, log in, and log out with email and password authentication.
- Admin has default credentials and can view, update, and delete users and restaurants.

3.2 Restaurant Management:

- Restaurants can manage their profiles (name, phone, address) and login with email and password.
- They can manage their menu by adding, updating, and deleting items (name, description, price, category, quantity, preparation time).

3.3 Customer Experience:

- Customers can manage their profiles (name, phone, address) and login with email and password.
- They can view all restaurants, search for restaurants/items, add items to cart, and place orders.
- Customers can track order status and view delivery person details.

3.4 Order Management:

- Once a customer places an order, the restaurant can accept or reject it, updating the order status.
- Delivery persons can view and update the status of orders as accepted, rejected, or delivered.

3.5 Cart and Payment:

- Customers can add items to their cart, view cart details (item names, quantity, price), and make payments.

- Orders include item details, customer information, total price, and order/delivery status.

3.6 Admin Dashboard:

- Admin has access to a dashboard to view all restaurants, customers, and delivery persons.
- Admin can manage users, including deleting restaurants.
- These features collectively provide a comprehensive food delivery management system with functionalities tailored for each user role, ensuring a smooth and efficient ordering and delivery experience.

4. External Interface Requirements

4.1 User Interfaces

Admin Interface:

- Login interface with username and password fields.
- Dashboard displaying options to view all restaurants, customers, delivery persons, and perform delete operations.

Restaurant Interface:

- Signup/login interface with email and password fields.
- Profile management interface for updating restaurant details.
- Menu management interface for adding, updating, and deleting items.
- Order acceptance/rejection interface for managing customer orders.

Customer Interface:

- Signup/login interface with email and password fields.
- Profile management interface for updating customer details.
- Restaurant/item browsing interface with search functionality.
- Cart management interface for adding/removing items and making payments.
- Order tracking interface to view order status and delivery person details.

Delivery Person Interface:

- Signup/login interface with email and password fields.
- Profile management interface for updating delivery person details.
- Order status interface for viewing and updating order delivery status.

4.2 Hardware Interfaces

Surface Computers:

- Touch-sensitive display for manipulating food items, dietary requirements, tips, and menus.
- Robust and liquid-resistant design with industrial aesthetics suitable for restaurant tables.

Wireless Tablets:

- Used by delivery persons for order status updates and navigation.
- Robust design with good industrial aesthetics and liquid resistance.

Touch Displays:

- Used in place of normal restaurant tables for customer interactions.
- Touch-sensitive with robust design and liquid resistance.

4.3 Software Interfaces

Database Management System (DBMS):

- Interface with MySQL DBMS for storing and retrieving data.
- Provide low-latency access to restaurant menu, employee details, dietary requirements, and order information.
- Secure access to data with password encryption and authentication mechanisms.

APIs:

- RESTful APIs for communication between frontend (React) and backend (Spring Boot) services.
- Endpoints for POST, PUT, GET, DELETE operations for all user roles (Admin, Restaurant, Customer, Delivery Person).
- Integration with payment gateways for secure online transactions.

These interface requirements ensure seamless interaction between users and the system, robust hardware functionality, and efficient data management through software interfaces.

4.4 Communications Interfaces

Local Area Network (LAN):

- The food delivery application will communicate over a Local Area Network (LAN) to connect all its devices.
- It should utilize a reliable-type IP protocol such as TCP/IP or reliable-UDP/IP for maximum compatibility and stability.
- All devices (servers, surface computers, tablets, displays, external payment system) should have standard Ethernet-compatible, software-accessible LAN cards to maintain communication within the LAN.
- This communication interface ensures seamless connectivity and data exchange between the server and all devices involved in the food delivery application, maintaining compatibility, stability, and reliability over the LAN.

5. Other Nonfunctional Requirements

The food delivery application must adhere to various nonfunctional requirements beyond functional features. These include aspects such as performance, safety, security, and software quality attributes, ensuring a robust and reliable system that meets user expectations and industry standards.

5.1 Performance Requirements:

Performance requirements for the food delivery application dictate efficient response times, scalability to handle concurrent users, and optimal resource utilization. The system must maintain responsiveness, even during peak usage periods, to provide users with a seamless and fast experience.

5.2 Safety Requirements:

Safety requirements focus on data integrity, fault tolerance, and system reliability. The application should have mechanisms in place to prevent data loss or corruption, recover from failures gracefully, and ensure uninterrupted service to users.

5.3 Security Requirements:

Security requirements are critical to protect user data, transactions, and system integrity. The application must employ strong authentication methods, encryption protocols, access controls, and compliance with privacy regulations to safeguard sensitive information.

5.4 Software Quality Attributes:

Software quality attributes encompass various aspects such as reliability, maintainability, usability, and performance optimization. The application should be reliable with minimal downtime, easy to maintain and update, user-friendly, and efficient in resource usage to deliver a high-quality user experience.

6. Other Requirements

Appendix A: Glossary

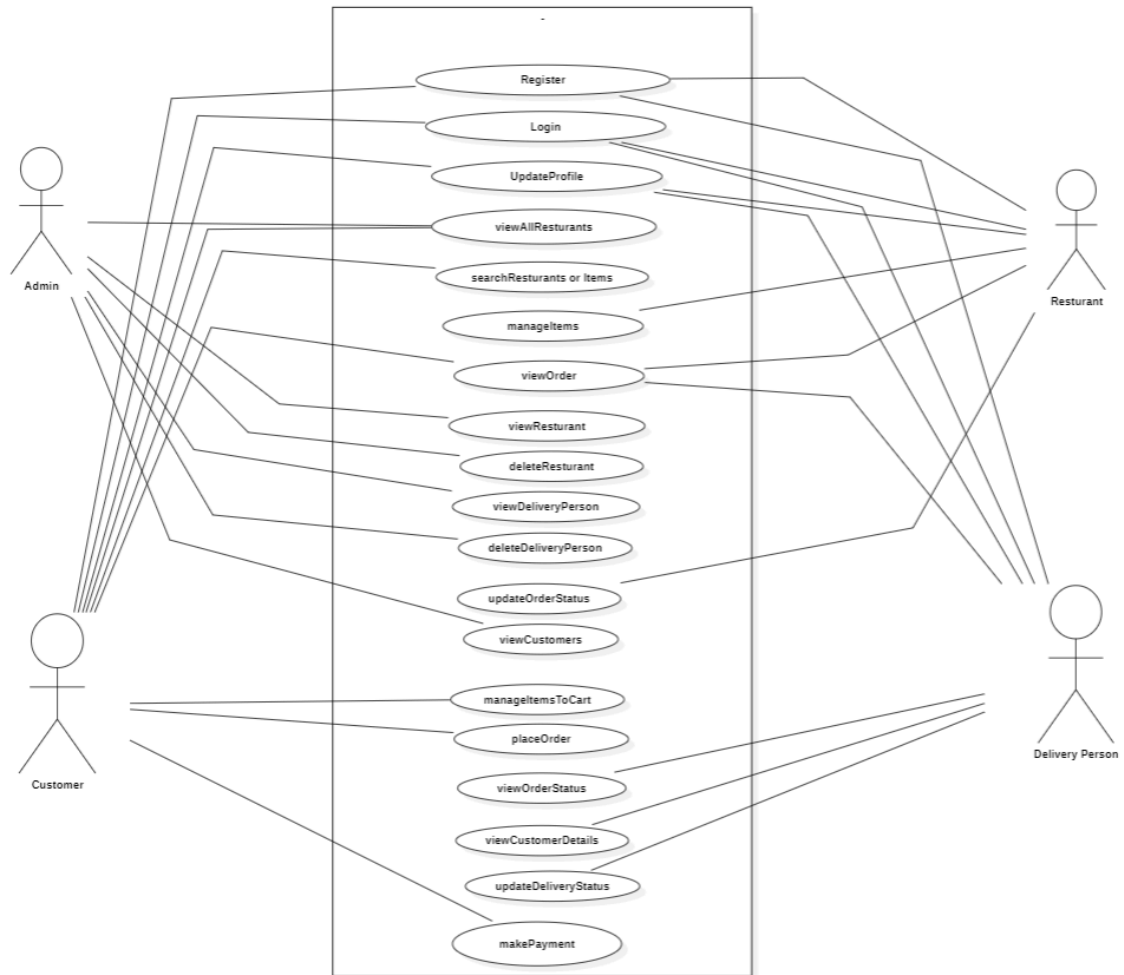
- Admin: Refers to the administrative user role in the food delivery application with privileges for managing users, restaurants, and delivery personnel.
- Restaurant: Represents a business entity that offers food items for delivery through the application. Restaurants can manage their menus, accept/reject orders, and update their profiles.
- Customer: Refers to users of the application who browse restaurants, view menus, add items to cart, place orders, and track order status.
- Delivery Person: Represents individuals responsible for delivering orders from restaurants to customers. They can update order status and manage their profiles.
- Item: Refers to food items offered by restaurants, including details such as name, description, price, category, quantity, and preparation time.
- Cart: Represents the virtual shopping cart where customers can add selected items before proceeding to checkout and payment.
- Order: Represents a customer's request for food items from a restaurant, including item details, customer information, total price, and order status.
- Order Status: Indicates the current status of an order, such as accepted, rejected, sent for delivery, or delivered.
- Delivery Status: Indicates the status of order delivery, such as accepted delivery, rejected delivery, or successfully delivered to the customer.
- Login: The process of authenticating users with valid credentials (username/email and password) to access their respective accounts.

- **Signup:** The process of creating a new user account in the system, typically requiring personal information and account credentials.
- **Profile Management:** Refers to the functionality for users, restaurants, and delivery personnel to update their personal information, contact details, and preferences.
- **Menu Management:** Refers to the functionality for restaurants to add, update, and delete items from their menus, including item details and prices.
- **Payment Processing:** The system's integration with payment gateways to facilitate secure online transactions for customer orders.
- **Feedback and Rating:** Allows customers to provide feedback, ratings, and reviews for restaurants and delivery experiences, contributing to service improvement and customer satisfaction.
- **Security Configuration:** Refers to the setup of security measures such as authentication, authorization, encryption, and data privacy to protect user information and system integrity.
- **APIs (Application Programming Interfaces):** Interfaces that allow communication and data exchange between different components of the system, including frontend (React) and backend (Spring Boot) services.
- **LAN (Local Area Network):** Refers to the network infrastructure used for communication between system devices, including servers, surface computers, tablets, displays, and external systems.

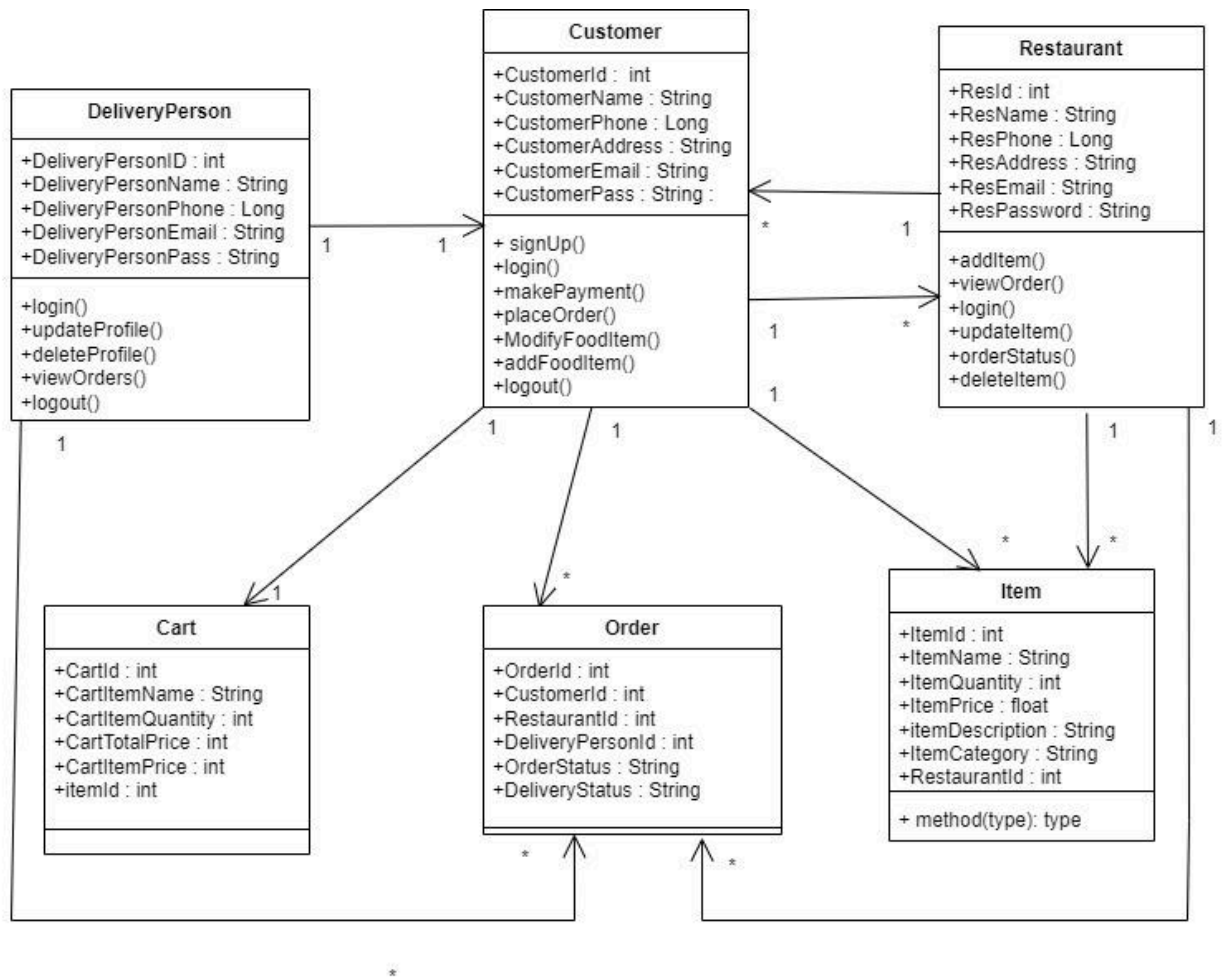
This glossary provides definitions for key terms and concepts used in the food delivery application, ensuring clarity and understanding of the system's functionalities.

Appendix B: Analysis Model

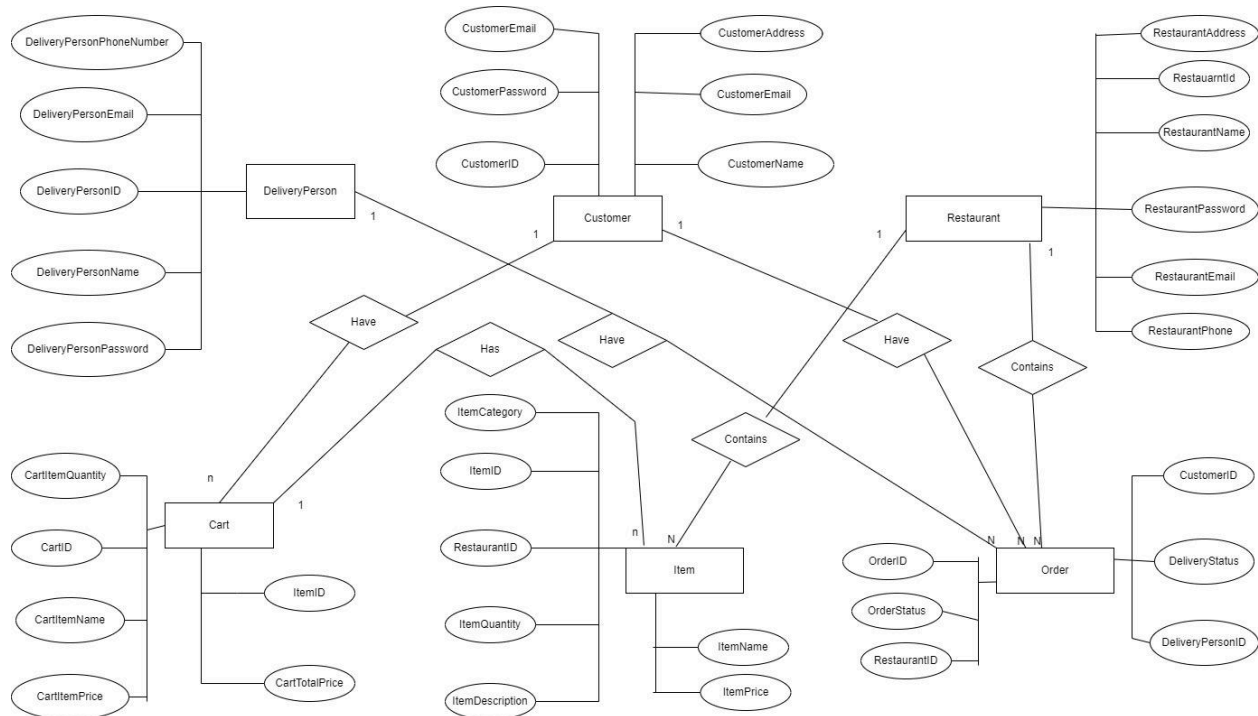
Use Case Diagram:



Class Diagram:



ER Diagram:



Appendix C: Issues List

- Payment Gateway Integration

Description: Payment gateway integration is not functioning properly, causing payment failures.

Priority: High

Status: Pending

- User Interface Improvements

Description: Customers and restaurants have provided feedback on UI improvements for better usability.

Priority: Low

Status: Pending

- Order Acceptance Workflow

Description: Refining the order acceptance workflow for restaurants to streamline operations.

Priority: Medium

Status: Pending

- Delivery Person Assignment

Description: Implementing a feature for assigning delivery persons to orders efficiently.

Priority: Medium

Status: Pending

This issue list helps track and prioritize various issues, bugs, and feature requests within the food delivery application, ensuring that they are addressed in a timely manner to improve the overall system performance and user experience.