Software Requirements Specification

for

Food Delivery Service Management Software (FDSMS)

Version 1.0 approved

Prepared by:

Sujan Jain(22CS10075)
Ashish Kumar(22CS30014)
Shshank Mittal(22CS30051)

Organization:

Indian Institute of Technology (IIT) Kharagpur

1 April,2024

Table of Contents

---

1. Introduction

1.1 Purpose

Providing a thorough overview of the Food Delivery Service Management system is the goal of this document. It functions as a thorough manual that describes the functions, interfaces, goal, and operational limitations of the system. Our application supports a wide range of users, including customers, delivery agents, and restaurants. It allows for the selection of menu items, designating delivery locations, and managing orders. The system also includes value-added elements that improve the user experience overall, such as user ratings and promotional offers.

1.2 Document Conventions

This document has been written using free writing tools such as MS Word.The document follows standard formatting conventions, employing Arial font for text and Times New Roman for headings. Proper indentation and numbering are used for clarity.

1.3 Intended Audience and Reading Suggestions

This document covers everything about the software, both technical and non-technical. It's useful for both developers and users, explaining why the software was made and how it works. You can find what you need in the Table of Contents.

1.4 Product Scope

FDSMS is a web-based application designed to streamline food delivery services, catering to customers, restaurants, and delivery agents. It offers features for menu management, order tracking, and user feedback.

1.5 References

The document draws inspiration from IEEE Std 830-1998, adhering to recommended practices for Software Requirements Specification.

2. Overall Description

2.1 Product Perspective

FDSMS is an independent software product developed to address the growing demand for efficient food delivery services. It operates within the client-server model, where users interact with a web-based interface served by both front-end and back-end components.

2.2 Product Functions

FDSMS offers a range of functions aimed at facilitating food delivery services:

- Food delivery from restaurants to customers through delivery agents
- Third-party delivery services
- Menu management for restaurants
- Real-time order status updates
- Order history tracking
- User rating and feedback mechanisms

2.3 User Classes and Characteristics

The system caters to four primary user classes:

Restaurant:

- Create and manage the menu for all available items.
- Review and decide whether to accept or reject received orders, providing status updates for accepted orders and assigning estimated service times.
- Send Delivery request to Delivery Agents.
- Mark orders as "Out for Delivery" once prepared.

Customer:

- Browse menus of various restaurants and place orders for food items.
- Receive updates on order status, informed by both restaurant and delivery agent activities.
- Rate restaurants and delivery personnel based on service quality.

- Use Promotional Offers during ordering Food Items

Delivery Agent:

- Set their current location for order pickup and view incoming delivery requests from restaurants.
- Accept delivery requests, specifying estimated delivery times, and update order status accordingly.

- Provide rating to customer upon successful delivery.

Management:

- Oversee the status of customers, restaurants, and delivery agents, managing ratings and feedback.
- Provide food and restaurant recommendations to customers.

- Offer promotions to loyal customers.

2.4 Operating Environment

FDSMS operates as a web application, accessible through standard web browsers. The front-end interface is developed using HTML(specifically Jinja2 templating) and CSS. While the backend relies on the **Flask framework** in **Python**. Data is managed through a **MongoDB database**.

2.5 Design and Implementation Constraints

MongoDB, as the database, imposed constraints on the types of queries that can be efficiently executed. Flask as the backend framework limits the available libraries or integrations compared to other frameworks like NodeJS and Django.

2.6 User Documentation

The app offers user-friendly interfaces, supported by FAQs to address common questions. For developers, there's technical documentation including API guides and online resources.

2.7 Assumptions and Dependencies

The system assumes users have access to a stable internet connection and possess a valid email address for registration.

3. External Interface Requirements

3.1 User Interfaces

The system offers separate interfaces for each user class, featuring login/sign-up functionalities and personalized dashboards tailored to specific user roles.

- **Homescreen**: Provides options for login and signup.
- **Management Dashboard**: Displays user details, lists of users, ratings, recommendations, and promotions.
- **Delivery Agent Dashboard**: Allows marking current location, viewing order lists, order details, and providing feedback.
- **Restaurant Dashboard**: Facilitates menu creation, order management, and viewing past orders.
- **Customer Dashboard**: Enables browsing restaurants, menus, placing orders, tracking order status, and accessing past orders.

3.2 Hardware Interfaces

The system does not have specific hardware requirements for users. However, the backend server requires sufficient processing power to handle incoming requests effectively.

3.3 Software Interfaces

FDSMS uses Bootstrap for front-end design and MongoDB for database management. It relies on the Flask framework for server-side processing and communicates with clients via standard HTTPS protocol.

3.4 Communications Interfaces

All communication within the system occurs via web browsers using HTTPS protocol, ensuring secure data transmission.

4. System Features

4.1 Customer Features

4.1.1 View Menus and Place Orders

Description: Customers can browse through the menus of available restaurants and select items to place orders.

Stimulus/Response Sequences: Customers navigate to the restaurant menu section. They can select desired items, specify quantity, and add them to their order list. Upon completion, customers proceed to the checkout process to finalize their order.

Functional Requirements:

- Display restaurant menus with clear item descriptions, price where available.
- Provide an intuitive interface for adding and removing items from the order list.
- Calculate and display the total order amount including delivery charges, if applicable.

4.1.2 View Past Orders

Description: Customers can access a list of their previous orders for sharing their experience.

Stimulus/Response Sequences: Customers navigate to the past orders section in their user dashboard. Each order entry includes details such as order ID, restaurant name, ordered items, order status, and timestamps.

Functional Requirements:

- Store order history for each customer in the database, linked to their user profile.
- Display past orders in a clear and organized manner.

4.1.3 View Current Pending Orders

Description: Customers can monitor the status of their pending orders, tracking the progress from submission to delivery.

Stimulus/Response Sequences: Customers access the pending orders section in their user dashboard to view a list of orders awaiting processing and delivery. Customers can select individual orders to view real-time updates on their status.

Functional Requirements:

- Display pending orders in the customer dashboard.
- Provide clear and informative status updates for each order, indicating stages such as order received, preparing, out for delivery, and delivered.
- Update order statuses in real-time based on notifications received from restaurants and delivery agents.

4.1.4 Provide Feedback

Description: Customers can submit feedback and ratings to evaluate the quality of service provided by restaurants and delivery agents.

Stimulus/Response Sequences: Customers access the feedback section in their past order section, where they can provide ratings for specific orders, restaurants, or delivery agent.

Functional Requirements:

- Capture feedback submissions securely and associate them with the corresponding orders, restaurants, or delivery agents.

4.1.5 View Promotional Offers

Description: Customers can access promotional offers provided by management and apply them during the order placement process.

Stimulus/Response Sequences: Customers can view a list of available offers while placing their order. Each promotion entry includes details such as offer description and applicable terms and conditions. Customers can select individual offers to apply them to their current order.

Functional Requirements:

- Retrieve and display promotional offers from the database, ensuring accuracy and consistency with current promotions.

4.2 Restaurant Features

4.2.1 Create a Menu

Description: Restaurants can create and manage their menus by adding, modifying, or removing food items and associated details.

Stimulus/Response Sequences: Restaurant staff access the menu management section in the restaurant dashboard, where they can add new items by providing details such as item name, price.

Functional Requirements:

- Provide intuitive interfaces for adding, editing, and deleting menu items.
- Store menu data securely in the database, ensuring integrity and confidentiality of restaurant-specific information.

4.2.2 View Past Orders

Description: Restaurants can see a full history of orders made to their place, which helps them understand past customer interactions and preferences.

Stimulus/Response Sequences: Restaurant staff navigate to the past orders section in the restaurant dashboard. Each order entry includes details such as order ID, customer name, ordered items, order status.

Functional Requirements:

- Retrieve and display past order data from the database.
- Present order information in a clear and organized manner, prioritizing readability and accessibility for restaurant staff.

4.2.3 View Current Pending Orders

Description: Restaurants can monitor incoming orders in real-time, reviewing details such as order contents, customer information, and requested delivery times.

Stimulus/Response Sequences: Restaurant staff access the pending orders section in the restaurant dashboard, where they can view a list of orders awaiting processing and confirmation. Each order entry includes details such as order ID, customer name, ordered items, delivery address.

Functional Requirements:

- Display pending orders in the restaurant dashboard.
- Enable restaurants to accept or reject pending orders based on availability of items, kitchen capacity, and staffing constraints.
- Allow restaurants to specify estimated preparation times for accepted orders.

4.2.4 Manage Delivery Requests

Description: Restaurants can coordinate delivery logistics by sending requests to available nearby delivery agents for accepted orders.

Stimulus/Response Sequences: Restaurant staff access the delivery request section in the restaurant dashboard, where they can view a list of available nearby delivery agent.

Functional Requirements:

- Track delivery request status and response times, updating order status accordingly based on delivery agent acceptance or rejection.
- Provide mechanisms for restaurants to communicate with assigned delivery agents, facilitating coordination and resolving issues during the delivery process.

4.2.5 Mark Orders as "Out for Delivery"

Description: Restaurants can update the status of accepted orders to indicate that they are ready for pickup and out for delivery by assigned delivery agents.

Stimulus/Response Sequences: Restaurants mark orders as "out for delivery" once preparation is complete and inform assigned delivery agents of order readiness.

Functional Requirements:

- Provide options for updating order statuses and sending notifications to assigned delivery agents regarding order readiness.
- Implement real-time updates and notifications to alert delivery agents of new orders available for pickup.
- Notify customers of order status changes, including updates on order preparation, pickup, and estimated delivery times.

4.3 Delivery Agent Features

4.3.1 Mark Their Location

Description: Delivery agents can update their current location within the application to indicate their availability for order pickup and delivery.

Stimulus/Response Sequences: Delivery agents access the mark location feature in the delivery agent dashboard, where they input their current location manually. The system updates the delivery agent's location in real-time and displays it on the list of nearby delivery agents of respective restaurants of that region.

Functional Requirements:

- Provide a drop down menu for delivery agents to input their current location.
- Update the delivery agent's location in the system database.

4.3.2 Accept and Manage Delivery Requests

Description: Delivery agents can view and accept delivery requests from nearby restaurants, manage order pickup and delivery, and update the status of each task in real-time.

Stimulus/Response Sequences: Delivery agents access the delivery request section in their dashboard, where they can view a list of available delivery requests from nearby restaurants. They select specific orders to accept, by giving the estimated time to reach the restaurant and subsequently to the customer. Throughout the process, delivery agents update the status of each task (e.g., order accepted, order picked up, order delivered) to keep stakeholders informed.

Functional Requirements:

- Display available delivery requests in the delivery agent dashboard.
- Provide options for delivery agents to accept individual delivery requests.
- Update the status of each delivery task (e.g., order picked up, delivered) in the system database.
- Once an order is accepted by a Delivery Agent it is removed from the list of delivery requests.

4.3.3 Provide Estimated Pickup and Delivery Time

Description: Delivery agents have to provide estimated pickup times for accepted orders from restaurants and estimated delivery times for orders in transit to customers.

Stimulus/Response Sequences: Delivery agents input estimated pickup times when accepting orders from restaurants, considering factors such as travel distance, traffic conditions.

Functional Requirements:

- Include fields for delivery agents to input estimated pickup times and delivery times during order acceptance and pickup stages.

4.3.4 Rate Customers

Description: Delivery agents can provide ratings and feedback to evaluate the behaviour and cooperation of customers during the delivery process.

Stimulus/Response Sequences: Delivery agents give rating to customers in See current orders section in their dashboard, where they can assign for individual customers. Ratings may be based on factors such as clarity of delivery instructions, and overall cooperation.

Functional Requirements:

- Capture and store rating data securely in the system database, associating ratings with specific customer and delivery agent accounts.
- Aggregate and analyse customer ratings to identify patterns and trends, and therefore accordingly management can give promotional offers to customers.

4.4 Management Features

4.4.1 Manage Lists of Customers, Restaurants, and Delivery Agents

Description: Management personnel can oversee and administer the lists of customers, restaurants, and delivery agents registered within the system.

Stimulus/Response Sequences: Management staff access the administrative interface in the management dashboard, where they can view and modify lists of registered users, including customers, restaurants, and delivery agents. They have the authority to delete accounts as needed.

Functional Requirements:

- Provide comprehensive user management functionalities, including viewing user profile and user deletion.

4.4.2 Maintain Ratings for Customers, Restaurants, and Delivery Agents

Description: Management personnel can maintain and analyse ratings provided by customers for restaurants, and delivery agents , and by delivery agents for customers, to assess performance and quality of service.

Stimulus/Response Sequences: Management staff can access the rating of all customers, restaurants and delivery agents in the respective category list. They analyse ratings trends over time, identifying high-performing entities and areas for improvement.

Functional Requirements:

- Collect and store rating data securely in the system database, categorizing ratings by entity type (customers, restaurants, delivery agents) and associating them with specific user accounts.
- Calculate and display average rating scores for each entity category.

4.4.3 Provide Restaurant and Food Recommendations for Customers

Description: Management personnel can curate and provide personalised restaurant and food recommendations to customers based on their preferences and past interactions and/or the performance of the restaurant in terms of rating given by its customers.

Stimulus/Response Sequences: Management staff access the Give Restaurant/Food Recommendation in the management dashboard, where based on their analysis on customer data and historical order patterns they can give restaurant, popular menu items recommendations.

Functional Requirements:

- Present recommendations to customers through View Recommended Restaurant feature in their dashboard.

4.4.4 Provide Promotional Offers for Good Customers

Description: Management personnel can identify and reward loyal or high-value customers by providing exclusive promotional offers and incentives.

Stimulus/Response Sequences: Management staff access create offer section in the management dashboard, where they can create promotional offer by specifying its name, allowed percentage discount and upper limit of offer. Management staff then can go to the customer list through their dashboard and give offer to customers of their choice.

Functional Requirements:

- Personalise promotional offers and incentives to align with each customer segment's preferences and interests, maximising relevance and effectiveness.

- Each customer's data in the database includes the Offer id’s of all offers which are currently awarded to him.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The system should handle database queries efficiently to ensure quick response times.

5.2 Safety Requirements

The application operates within web browsers, minimising risks to user devices.

5.3 Security Requirements

Data storage and server integrity are vital to prevent potential damage after deployment.

5.4 Software Quality Attributes

- **Maintainability:** The system should be easy to maintain, allowing for seamless updates and addition of new features.
- **Usability:** The application should be user-friendly, easy to navigate. Simple, intuitive design enhances user experience**.**
- **Flexibility:** The system should be flexible, allowing for modifications and adaptations to new technologies and third-party components seamlessly.

5.5 Business Rules

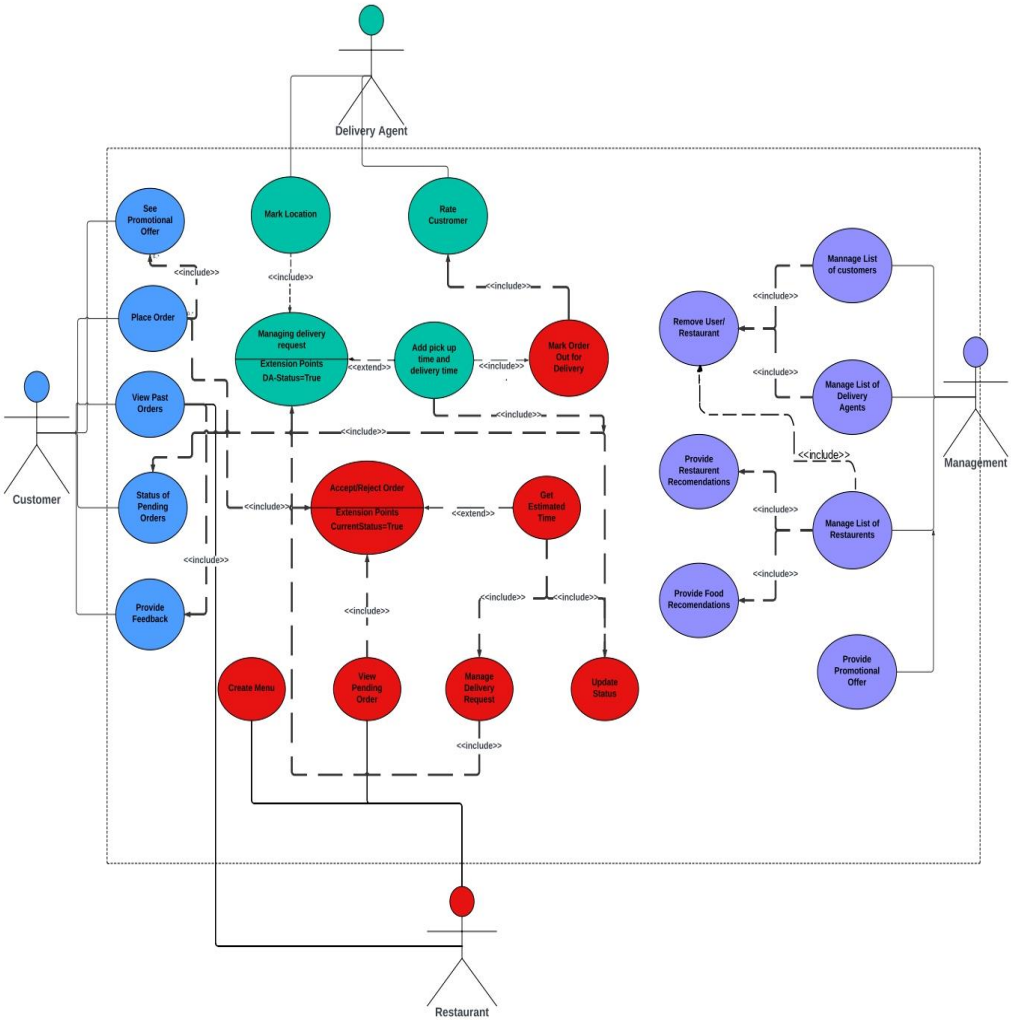
The software is freely available for public use, and its source code is open for modification and use. This fosters collaboration and innovation within the development community.

6. Other Requirements

There are some DataBase requirements for the person who will run the program. There should be MongoDB Atlas account and there should be a required structure of DataBase.

Appendix B : Analysis Models

Use-Case Diagram



UML Class Diagram