

**Software Requirements**

**Specification**

**for**

**Food Delivery Service Management Software (FDSMS)**

**Version 1.0 approved**

**Prepared by:**

**Shshank Mittal(22CS30051)**

**Sujan Jain(22CS10075)**

**Ashish Kumar(22CS30014)**

**Organization:**

**Indian Institute of Technology (IIT) Kharagpur**

**Date:**

## **Table of Contents**

1. **Introduction**
   * 1.1 Purpose
   * 1.2 Document Conventions
   * 1.3 Intended Audience and Reading Suggestions
   * 1.4 Product Scope
   * 1.5 References
2. **Overall Description**
   * 2.1 Product Perspective
   * 2.2 Product Functions
   * 2.3 User Classes and Characteristics
   * 2.4 Operating Environment
   * 2.5 Design and Implementation Constraints
   * 2.6 User Documentation
   * 2.7 Assumptions and Dependencies
3. **External Interface Requirements**
   * 3.1 User Interfaces
   * 3.2 Hardware Interfaces
   * 3.3 Software Interfaces
   * 3.4 Communications Interfaces
4. **System Features**
   * 4.1 Register a Customer/Delivery Agent/Restaurant
     + 4.1.1 Description
     + 4.1.2 Stimulus/Response Sequences
     + 4.1.3 Functional Requirements
   * 4.2 Menu of a Restaurant
     + 4.2.1 Description
     + 4.2.2 Stimulus/Response Sequences
     + 4.2.3 Functional Requirements
   * 4.3 Order by a Customer
     + 4.3.1 Description
     + 4.3.2 Stimulus/Response Sequences
     + 4.3.3 Functional Requirements
   * 4.4 Promotional offers
     + 4.4.1 Description
     + 4.4.2 Stimulus/Response Sequences
     + 4.4.3 Functional Requirements
   * 4.5 Provide feedback
     + 4.5.1 Description
     + 4.5.2 Stimulus/Response Sequences
     + 4.5.3 Functional Requirements
   * 4.6 Restaurant
     + 4.6.1 See available delivery agent
       - 4.6.1.1 Description
       - 4.6.1.2 Stimulus/Response Sequences
       - 4.6.1.3 Functional Requirements
     + 4.6.2 Set the available orders for pickup in an area
       - 4.6.2.1 Description
       - 4.6.2.2 Stimulus/Response Sequences
       - 4.6.2.3 Functional Requirements
     + 4.6.3 See pending orders
       - 4.6.3.1 Description
       - 4.6.3.2 Stimulus/Response Sequences
       - 4.6.3.3 Functional Requirements
   * 4.7 Delivery Agent
     + 4.7.1 Mark their location
       - 4.7.1.1 Description
       - 4.7.1.2 Stimulus/Response Sequences
       - 4.7.1.3 Functional Requirements
     + 4.7.2 See Delivery Requests available in an area
       - 4.7.2.1 Description
       - 4.7.2.2 Stimulus/Response Sequences
       - 4.7.2.3 Functional Requirements
     + 4.7.3 Provide estimated pick-time from a restaurant and delivery time to a customer
       - 4.7.3.1 Description
       - 4.7.3.2 Stimulus/Response Sequences
       - 4.7.3.3 Functional Requirements
   * 4.8 Management
     + 4.8.1 Provide Restaurant and Food Recommendations to Customers
       - 4.8.1.1 Description
       - 4.8.1.2 Stimulus/Response Sequences
       - 4.8.1.3 Functional Requirements
     + 4.8.2 Can manage list of customers, restaurants and delivery agents
       - 4.8.2.1 Description
       - 4.8.2.2 Stimulus/Response Sequences
       - 4.8.2.3 Functional Requirements
5. **Other Nonfunctional Requirements**
   * 5.1 Performance Requirements
   * 5.2 Safety Requirements
   * 5.3 Software Quality Attributes
   * 5.4 Business Rules

### **1. Introduction**

#### **1.1 Purpose**

The aim of this document is to offer a comprehensive depiction of the Food Delivery Service Management system. It serves as a detailed guide outlining the system's purpose, functionalities, interfaces, and operational constraints. Our application caters to a diverse user base, including restaurants, delivery agents, and customers, facilitating various services such as menu selection, delivery location designation, and order management. Additionally, the system incorporates value-added features like promotional offers and user ratings, enhancing the overall user experience. Notably, this platform serves as a valuable resource for individuals unable to venture outside due to safety concerns or limited resources, while also providing restaurants with an avenue to expand their customer base..

#### **1.2 Document Conventions**

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 comprehensively covers both technical and non-technical aspects of the software, serving as a valuable resource for developers and end-users alike. It provides insights into the underlying motivations driving the development of the software and delves into the intricate details of its implementation. Anybody who wants to use the software can read the appropriate parts of the document, a list of which is given 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:

* Direct food delivery from restaurants to customers
* 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.
* Assign delivery agents to orders.
* 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.

**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 statuses accordingly.

**Management:**

* Oversee the statuses 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, CSS, and JavaScript, while the back-end relies on the Flask framework in Python. Data is managed through a Firebase database.

#### **2.5 Design and Implementation Constraints**

The use of Firebase as the database imposes certain constraints, such as compliance with Firebase's usage policies. Any changes in these policies may impact system functionality.

#### **2.6 User Documentation**

The application provides intuitive user interfaces, supplemented by FAQs for common queries. Technical documentation for developers includes API documentation 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. Dependencies include Google's Material.io for front-end design and Firebase for database management.

### **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 interacts with Google's Material.io for front-end design and Firebase 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, where they can view available items categorized by type (e.g., appetizers, main courses, beverages). 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, prices, and images where available.
* Allow customers to search for specific items or filter by categories.
* Provide an intuitive interface for adding and removing items from the order list.
* Calculate and display the total order amount including taxes and delivery charges, if applicable.
* Securely handle payment transactions and confirm orders upon successful payment.

##### **4.1.2 View Past Orders**

**Description:** Customers can access a list of their previous orders for reference and reordering purposes.

**Stimulus/Response Sequences:** Customers navigate to the past orders section in their user dashboard, where they can view a chronological list of their previous orders. Each order entry includes details such as order ID, restaurant name, ordered items, order status, and timestamps. Customers can select individual orders to view more detailed information.

**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, allowing customers to easily identify and select orders of interest.
* Provide options for sorting and filtering past orders based on different criteria (e.g., date, order status).
* Allow customers to reorder items from past orders with a single click, optionally modifying quantities or adding additional items.

##### **4.1.3 View 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, where they can view a list of orders awaiting processing and delivery. Each order entry includes details such as order ID, restaurant name, ordered items, estimated delivery time, and current status. Customers can select individual orders to view real-time updates on their status.

**Functional Requirements:**

* Display pending orders prominently in the customer dashboard, prioritizing visibility and accessibility.
* 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.
* Allow customers to request assistance or make inquiries regarding pending orders through the customer support interface.

##### **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 user dashboard, where they can provide ratings and comments for specific orders, restaurants, or delivery agents. They can rate various aspects such as food quality, delivery speed, and overall satisfaction on a predefined scale (e.g., star ratings). Customers can also provide written feedback to elaborate on their experiences and suggest improvements.

**Functional Requirements:**

* Capture feedback submissions securely and associate them with the corresponding orders, restaurants, or delivery agents.
* Present feedback forms with clear instructions and intuitive interfaces to encourage customer participation.
* Validate and sanitize feedback entries to prevent misuse or inappropriate content.
* Aggregate and analyze feedback data to generate performance metrics and identify areas for improvement.
* Notify restaurants and delivery agents of received feedback and provide mechanisms for response or follow-up actions.

##### **4.1.5 View Promotional Offers**

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

**Stimulus/Response Sequences:** Customers navigate to the promotions section in their user dashboard, where they can view a list of available offers and discounts. Each promotion entry includes details such as offer description, validity period, and applicable terms and conditions. Customers can select individual offers to apply them to their current order, triggering corresponding adjustments in the total order amount.

**Functional Requirements:**

* Retrieve and display promotional offers from the database, ensuring accuracy and consistency with current promotions.
* Highlight featured offers and new additions to attract customer attention and encourage utilization.
* Allow customers to redeem applicable offers during checkout by applying relevant promo codes or selecting offers from the available list.
* Validate and enforce offer eligibility criteria to prevent misuse and ensure fair distribution of benefits.
* Track and record the usage of promotional offers to assess their effectiveness and inform future marketing strategies.

#### **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, description, category, price, and optional image. They can also edit existing items to update their information or remove items that are no longer available.

**Functional Requirements:**

* Provide intuitive interfaces for adding, editing, and deleting menu items, incorporating form validation and error handling mechanisms.
* Support categorization and organization of menu items into logical groups (e.g., appetizers, entrees, desserts) to facilitate navigation and browsing.
* Validate input data to ensure consistency and accuracy of menu information, enforcing constraints such as maximum item length and valid price formats.
* Store menu data securely in the database, ensuring integrity and confidentiality of restaurant-specific information.
* Enable version control and revision history for menu changes, allowing restaurants to revert to previous versions if needed.

##### **4.2.2 View Past Orders**

**Description:** Restaurants can access a comprehensive history of orders placed to their establishment, providing insights into past customer interactions and preferences.

**Stimulus/Response Sequences:** Restaurant staff navigate to the past orders section in the restaurant dashboard, where they can view a chronological list of orders received over a specified time period. Each order entry includes details such as order ID, customer name, ordered items, order status, and timestamps. Restaurants can select individual orders to view additional information or perform actions such as printing receipts or generating reports.

**Functional Requirements:**

* Retrieve and display past order data from the database, allowing restaurants to filter and search for specific orders based on criteria such as date range, order status, or customer details.
* Present order information in a clear and organized manner, prioritizing readability and accessibility for restaurant staff.
* Include options for exporting order data in various formats (e.g., PDF, CSV) for record-keeping and analysis purposes.
* Implement role-based access controls to restrict access to sensitive order information and ensure compliance with privacy regulations.
* Provide mechanisms for archiving and purging old order records to optimize database performance and storage utilization.

##### **4.2.3 View 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, and requested delivery time. Restaurants can review each order to determine availability of items, assess preparation time, and allocate resources accordingly.

**Functional Requirements:**

* Display pending orders prominently in the restaurant dashboard, prioritizing visibility and accessibility for timely processing.
* Provide interactive features for reviewing order details, such as expanding order summaries or viewing item descriptions.
* 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, informing customers and delivery agents of expected service intervals.
* Facilitate communication between restaurants and customers/delivery agents regarding order status updates, delivery delays, or special instructions.

##### **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 agents and select suitable candidates for order fulfillment. Restaurants initiate delivery requests by assigning selected delivery agents to specific orders and providing relevant order details and estimated delivery times.

**Functional Requirements:**

* Integrate location-based services to identify and display nearby delivery agents within a specified radius of the restaurant's location.
* Present delivery agent profiles and availability status, including details such as distance from the restaurant, current workload, and average delivery times.
* Enable restaurants to send delivery requests to selected agents, including order-specific instructions and contact information for coordination purposes.
* Track delivery request status and response times, updating order statuses 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:** Restaurant staff access the order management interface in the restaurant dashboard, where they can locate accepted orders awaiting pickup and delivery. Restaurants mark orders as "out for delivery" once preparation is complete and inform assigned delivery agents of order readiness.

**Functional Requirements:**

* Display accepted orders prominently in the restaurant dashboard, distinguishing them from pending and completed orders.
* 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.
* Track the progress of orders marked as "out for delivery," monitoring delivery agent movements and estimated delivery times.
* 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 location tracking feature in the delivery agent dashboard, where they can input their current location manually or via GPS tracking. The system updates the delivery agent's location in real-time and displays it on the map interface.

**Functional Requirements:**

* Provide an intuitive interface for delivery agents to input their current location, allowing for manual entry or automatic detection using GPS.
* Update the delivery agent's location in the system database and display it on the map interface for easy reference by restaurants and management.
* Ensure accuracy and reliability of location tracking, minimizing errors and discrepancies in position reporting.
* Implement privacy controls to allow delivery agents to enable or disable location sharing based on their preferences and availability.

##### **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, confirming their availability and readiness to proceed with order pickup and delivery. 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 prominently in the delivery agent dashboard, prioritizing orders based on proximity and urgency.
* Provide options for delivery agents to accept or reject individual delivery requests, with accompanying reasons for rejection if applicable.
* Facilitate communication between delivery agents and restaurants/customers to coordinate order pickup and delivery, including real-time messaging and notifications.
* Update the status of each delivery task (e.g., order picked up, en route, delivered) in the system database and notify relevant stakeholders of status changes.
* Implement safeguards to prevent double booking and ensure delivery agents can effectively manage their workload without overcommitting.

##### **4.3.3 Provide Estimated Pickup and Delivery Time**

**Description:** Delivery agents can provide estimated pickup times for accepted orders from restaurants and estimate 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, and order preparation time. They also provide estimated delivery times to customers upon order pickup, updating the delivery status with revised time estimates as needed.

**Functional Requirements:**

* Include fields for delivery agents to input estimated pickup times and delivery times during order acceptance and pickup stages.
* Calculate and display estimated delivery times based on factors such as distance, route optimization, and historical delivery data.
* Continuously update estimated delivery times based on real-time conditions and progress updates, adjusting for delays or unexpected events.
* Communicate estimated pickup and delivery times to relevant stakeholders (e.g., restaurants, customers) via notifications and order tracking interfaces.
* Monitor and analyze actual pickup and delivery times to refine estimation algorithms and improve accuracy over time.

##### **4.3.4 Rate Customers**

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

**Stimulus/Response Sequences:** Delivery agents access the rating and feedback section in their dashboard, where they can assign ratings and optionally provide comments for individual customers. Ratings may be based on factors such as punctuality, clarity of delivery instructions, and overall cooperation.

**Functional Requirements:**

* Present customers' information and order history to delivery agents, allowing for informed ratings based on past interactions.
* Include a rating scale or scoring system (e.g., star ratings) for delivery agents to assign ratings to customers, with corresponding descriptions for each rating level.
* Capture and store rating data securely in the system database, associating ratings with specific orders and delivery agent accounts.
* Provide options for delivery agents to provide written feedback or comments alongside numerical ratings, allowing for detailed explanations and context.
* Aggregate and analyze customer ratings and feedback to identify patterns and trends, informing customer service improvements and policy adjustments.

### **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 add new entries, update existing information, or deactivate accounts as needed. Management personnel may also perform bulk actions such as importing/exporting user data or generating reports.

**Functional Requirements:**

* Provide comprehensive user management functionalities, including user creation, modification, and deletion, with appropriate access controls and permissions.
* Ensure data integrity and security by enforcing validation rules, such as mandatory fields and data format checks, during user management operations.
* Implement search and filtering capabilities to facilitate navigation and retrieval of specific user records based on criteria such as user type, status, or geographic location.
* Enable management personnel to review and approve new user registrations, verifying authenticity and compliance with platform policies.
* Log all user management activities and changes for auditing and accountability purposes, maintaining an audit trail of administrative actions.

##### **4.4.2 Maintain Ratings for Customers, Restaurants, and Delivery Agents**

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

**Stimulus/Response Sequences:** Management staff access the ratings management section in the management dashboard, where they can view aggregated rating data for customers, restaurants, and delivery agents. They analyze ratings trends over time, identifying high-performing entities and areas for improvement. Management may also intervene in cases of inconsistent or unfair ratings to ensure fairness and accuracy.

**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 or orders.
* Calculate and display average rating scores for each entity category, providing a summary of overall performance based on user feedback.
* Enable management personnel to drill down into rating details, viewing individual ratings and associated comments to gain deeper insights into user experiences.
* Implement data visualization tools and analytics dashboards to present rating data in an intuitive and actionable format, supporting informed decision-making.
* Establish performance benchmarks and quality standards based on rating metrics, using them to recognize top performers and identify areas for targeted improvement initiatives.

##### **4.4.3 Provide Restaurant and Food Recommendations for Customers**

**Description:** Management personnel can curate and provide personalized restaurant and food recommendations to customers based on their preferences and past interactions.

**Stimulus/Response Sequences:** Management staff access the recommendation engine in the management dashboard, where they can analyze customer data and historical order patterns to generate tailored recommendations. Recommendations may include suggested restaurants, popular menu items, or special promotions aligned with customer preferences. Management may also leverage machine learning algorithms or collaborative filtering techniques to enhance recommendation accuracy and relevance.

**Functional Requirements:**

* Aggregate and analyze customer data, including order history, preferences, and demographic information, to identify relevant patterns and preferences.
* Implement recommendation algorithms and models to generate personalized recommendations for restaurants and food items based on customer profiles and behavior.
* Present recommendations to customers through various channels, such as the user dashboard, email newsletters, or push notifications, ensuring visibility and engagement.
* Allow customers to provide feedback and ratings on recommended restaurants and food items, enabling continuous refinement and improvement of recommendation algorithms.
* Measure the effectiveness of recommendations through metrics such as click-through rates, conversion rates, and customer satisfaction scores, iteratively refining recommendation strategies based on performance insights.

##### **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 the promotions management section in the management dashboard, where they can identify and segment customers based on criteria such as order frequency, order value, or loyalty status. They create targeted promotional offers and discounts tailored to each customer segment, incentivizing repeat business and fostering customer loyalty. Management may also track redemption rates and campaign performance to assess effectiveness and refine promotional strategies over time.

**Functional Requirements:**

* Segment customers into distinct groups or cohorts based on predefined criteria, such as order history, spending behavior, or engagement level.
* Design and deploy promotional campaigns targeting specific customer segments, utilizing channels such as email marketing, in-app notifications, or social media promotions.
* Personalize promotional offers and incentives to align with each customer segment's preferences and interests, maximizing relevance and effectiveness.
* Monitor and track customer response to promotional offers, capturing metrics such as redemption rates, conversion rates, and revenue impact.
* Analyze promotional campaign performance and customer behavior data to optimize future campaigns, iterating on messaging, timing, and offer types to maximize ROI and customer engagement.

### **5. Other Nonfunctional Requirements**

#### **5.1 Performance Requirements**

### The system should handle database queries efficiently to ensure quick response times. Balancing performance and accuracy is crucial, achieved through optimization techniques.

#### **5.2 Safety Requirements**

### The application operates within web browsers, minimizing risks to user devices. However, data storage and server integrity are vital to prevent potential damage during heavy usage.

#### **5.3 Software Quality Attributes**

### **Maintainability:** The system should be easy to maintain, allowing for seamless updates and addition of new features. Cost-effective maintenance and upgradability are essential.

### **Usability:** The application should be user-friendly, easy to learn, and 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.

#### **5.4 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.

### **Conclusion**

### The Food Delivery Service Management Software (FDSMS) aims to revolutionize food delivery by providing a convenient platform for users to order food from restaurants and manage deliveries efficiently. With a focus on user experience, performance, and safety, FDSMS seeks to enhance the food delivery experience for customers, restaurants, delivery agents, and management alike.

### This Software Requirements Specification serves as a comprehensive guide for developers and stakeholders, outlining the system's functionalities, interfaces, and nonfunctional requirements. By adhering to these specifications, the development team can ensure the successful implementation of FDSMS, meeting the needs and expectations of its users.

### **Appendix**

### **Glossary:** Defines technical terms and acronyms used in the document.

### **References:** Lists resources consulted during the development of this document.

### **This concludes the Software Requirements Specification for the Food Delivery Service Management Software (FDSMS). For any further inquiries or clarifications, please refer to the contact information provided in the document.**

### 