**Software Requirements Specification (SRS)**

**Table of Contents**

1. Introduction  
   1.1 Purpose  
   1.2 Scope  
   1.3 Definitions, Acronyms, and Abbreviations  
   1.4 References  
   1.5 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 Order Management  
   3.2 Restaurant Management  
   3.3 Delivery Tracking  
   3.4 Payment Integration  
   3.5 Notification System
4. External Interface Requirements  
   4.1 User Interfaces  
   4.2 Hardware Interfaces  
   4.3 Software Interfaces  
   4.4 Communication Interfaces
5. System Attributes  
   5.1 Performance Requirements  
   5.2 Security Requirements  
   5.3 Usability Requirements  
   5.4 Reliability and Availability  
   5.5 Maintainability  
   5.6 Portability
6. Other Non-Functional Requirements
7. Appendices  
   7.1 Diagrams and Flowcharts  
   7.2 Glossary

**1. Introduction**

**1.1 Purpose**

This Software Requirements Specification (SRS) document is intended to outline and define the functionalities, design constraints, and features of a **Food Delivery Application (FDA)**. This document will be used by the development team, stakeholders, and testers to understand and agree upon the system’s requirements and ensure the final product meets the expectations of all users and stakeholders.

**1.2 Scope**

The **Food Delivery Application** will provide users the ability to browse menus, order food, and track their deliveries in real-time. It will serve three primary user classes: customers, restaurant partners, and delivery agents. The app will support user registration, secure payments, order tracking, restaurant management, and customer feedback. It will be available for both Android and iOS devices as a native mobile application and also accessible via a web interface.

**1.3 Definitions, Acronyms, and Abbreviations**

* **FDA**: Food Delivery Application
* **API**: Application Programming Interface
* **UI**: User Interface
* **OTP**: One-Time Password
* **GPS**: Global Positioning System
* **CRUD**: Create, Read, Update, Delete
* **POS**: Point of Sale system

**1.4 References**

* IEEE Standard 830-1998: Software Requirements Specifications
* Android and iOS Development Guidelines
* PCI DSS (Payment Card Industry Data Security Standard) for secure payments
* Mobile App UI/UX Design Standards

**1.5 Overview**

This SRS includes a comprehensive description of the system’s overall architecture, functionalities, user interfaces, system features, non-functional requirements, and external interfaces. Diagrams and charts are included to describe data flow and system architecture.

**2. Overall Description**

**2.1 Product Perspective**

The **Food Delivery Application (FDA)** will serve as an intermediary between customers, restaurants, and delivery agents. It will allow customers to browse through restaurant menus, place orders, make payments, and track their deliveries. Restaurants will use the app to manage their menus, accept orders, and track deliveries. Delivery agents will use the app to pick up and deliver orders while using GPS tracking for optimized routes. The FDA follows a **client-server architecture**, leveraging RESTful APIs to process orders, payments, and communications between all parties.

**2.1.1 System Interfaces**

* **Cloud Server**: The FDA will use cloud services to store user, order, and restaurant data.
* **Payment Gateway**: The system will interface with secure third-party payment gateways (e.g., Stripe, PayPal) for processing customer payments.
* **GPS Tracking**: The system will interface with Google Maps API for real-time tracking and optimized delivery routes.

**2.1.2 User Interfaces**

* **Mobile Application**: Native mobile apps for Android and iOS, built with a responsive and intuitive UI.
* **Web Application**: A browser-based version for desktop users, featuring the same functionality as the mobile app.

**2.1.3 Hardware Interfaces**

* **Mobile Devices**: The app will run on smartphones and tablets with GPS and internet connectivity.
* **POS Systems**: Restaurants with Point of Sale (POS) systems can integrate with the app to streamline orders.

**2.1.4 Software Interfaces**

* **Operating Systems**: Android 7.0+ and iOS 12.0+
* **Web Browsers**: Chrome, Safari, Firefox (latest versions)
* **Third-party APIs**: Google Maps API for GPS tracking, Payment Gateway APIs (e.g., Stripe)

**2.1.5 Communications Interfaces**

* **HTTP/HTTPS**: The system will communicate with the server through RESTful web services using HTTPS for secure data transmission.

**2.2 Product Features**

* **User Registration and Login**: Secure registration and login using email, phone, and OTP verification.
* **Menu Browsing**: Customers can browse through restaurants, filter based on cuisine, location, and price.
* **Order Placement**: Users can add items to their cart, place orders, and track delivery status.
* **Real-time Tracking**: Delivery tracking using GPS for real-time location updates.
* **Payment Processing**: Multiple payment options including credit cards, debit cards, and mobile wallets.
* **Restaurant Management**: Restaurants can update menus, accept/reject orders, and track the delivery status.
* **Reviews and Feedback**: Customers can leave reviews and ratings for restaurants and delivery agents.

**2.3 User Classes and Characteristics**

* **Customer**: Users who browse the app, place orders, and track deliveries. They expect a smooth, intuitive user experience.
* **Restaurant Partner**: Restaurants that list their menu on the app and manage incoming orders.
* **Delivery Agent**: Individuals responsible for delivering orders to customers, utilizing GPS tracking for optimized delivery.
* **Admin**: Responsible for managing the overall system, including user accounts, restaurants, and order flow.

**2.4 Operating Environment**

* The mobile app will operate on smartphones with internet access and GPS functionality.
* The web version will run on any modern web browser (Chrome, Firefox, Safari) on desktop or laptop devices.
* The backend will be hosted on a cloud platform (e.g., AWS or Azure) to ensure scalability and availability.

**2.5 Design and Implementation Constraints**

* **Security**: The system must comply with PCI DSS for payment processing security and use encryption for sensitive data.
* **Scalability**: The system must handle up to 100,000 concurrent users and support expansion to multiple cities.
* **Latency**: The system should ensure low-latency GPS updates for accurate delivery tracking.

**2.6 Assumptions and Dependencies**

* Users are assumed to have internet access and location services enabled on their mobile devices.
* The system will rely on third-party payment gateways for handling secure transactions.
* GPS functionality is dependent on accurate location services provided by mobile devices and Google Maps API.

**3. System Features**

**3.1 Order Management**

**3.1.1 Description and Priority**

This feature allows customers to search restaurants, browse menus, and place food orders. It is a **high priority** feature and forms the core functionality of the FDA.

**3.1.2 Stimulus/Response Sequences**

* **Stimulus**: A user selects items from a restaurant’s menu and adds them to the cart.
* **Response**: The system calculates the total cost and displays the cart to the user for confirmation.
* **Stimulus**: A user confirms the order and initiates payment.
* **Response**: The system processes the payment and forwards the order to the selected restaurant for preparation.

**3.1.3 Functional Requirements**

* **REQ-1**: The system shall allow users to search for restaurants by location, cuisine, or price.
* **REQ-2**: The system shall allow users to add items to their cart and modify the cart before checkout.
* **REQ-3**: The system shall calculate taxes and delivery fees based on the user’s location.
* **REQ-4**: The system shall notify the restaurant of a new order in real time.

**3.2 Restaurant Management**

**3.2.1 Description and Priority**

This feature allows restaurant partners to manage their menus, accept orders, and track the delivery process. It is a **medium priority** feature, but critical for restaurant operations.

**3.2.2 Functional Requirements**

* **REQ-5**: The system shall allow restaurant managers to update and modify their menus.
* **REQ-6**: The system shall provide real-time notifications to restaurant partners for new orders.
* **REQ-7**: The system shall allow restaurant partners to accept or reject orders.

**3.3 Delivery Tracking**

**3.3.1 Description and Priority**

This feature allows customers to track their order’s status in real time. It is a **high priority** feature for enhancing the user experience.

**3.3.2 Functional Requirements**

* **REQ-8**: The system shall provide real-time GPS tracking of the delivery agent’s location.
* **REQ-9**: The system shall notify the customer when their food is picked up and when it is approaching the delivery location.

**3.4 Payment Integration**

**3.4.1 Description and Priority**

This feature allows users to securely make payments for their orders. It is a **high priority** feature as it involves financial transactions.

**3.4.2 Functional Requirements**

* **REQ-10**: The system shall support multiple payment methods including credit cards, debit cards, and digital wallets.
* **REQ-11**: The system shall use a third-party payment gateway for secure transactions.
* **REQ-12**: The system shall send a payment confirmation to the user once the payment is successfully processed.

**3.5 Notification System**

**3.5.1 Description and Priority**

This feature handles notifications sent to customers, restaurants, and delivery agents. It is a **medium priority** feature.

**3.5.2 Functional Requirements**

* **REQ-13**: The system shall send push notifications to the customer for order status updates.
* **REQ-14**: The system shall notify the delivery agent when an order is ready for pickup.

**4. External Interface Requirements**

**4.1 User Interfaces**

* **Mobile UI**: Must be responsive, intuitive, and support features like push notifications and location services.
* **Web UI**: Should mirror the mobile app’s functionality and have a user-friendly design for desktop users.

**4.2 Hardware Interfaces**

* **Mobile Devices**: The app should operate on any device with a minimum of 2GB RAM and GPS support.

**4.3 Software Interfaces**

* **Google Maps API**: For real-time GPS tracking and route optimization.
* **Payment Gateway API**: For secure payment processing (e.g., Stripe, PayPal).

**4.4 Communication Interfaces**

* **Push Notifications**: Delivered to users’ mobile devices to inform them of order status changes.
* **Email/SMS**: For registration confirmation and receipt of orders.

**5. System Attributes**

**5.1 Performance Requirements**

* The system shall handle up to 500 orders per minute.
* The system shall support up to 100,000 concurrent users.

**5.2 Security Requirements**

* User data shall be encrypted using AES-256 encryption.
* Payment transactions shall comply with PCI DSS standards.

**5.3 Usability Requirements**

* The UI shall be intuitive and require minimal user training.
* The system shall be accessible to users with disabilities.

**5.4 Reliability and Availability**

* The system shall provide 99.9% uptime and auto-scaling features to handle high traffic periods.

**5.5 Maintainability**

* The codebase shall follow microservices architecture for scalability and maintainability.
* The system shall support continuous integration and automated testing for bug fixes and feature updates.

**5.6 Portability**

* The mobile app shall be deployable on both Android and iOS.
* The web version shall be compatible with all modern web browsers.

**6. Other Non-Functional Requirements**

* **Backup and Recovery**: Daily automated backups shall be stored in a secure cloud environment.
* **Scalability**: The system shall auto-scale based on user demand, ensuring consistent performance during peak times.

**7. Appendices**

**7.1 Diagrams and Flowcharts**

* Use Case Diagram: To be added.
* System Architecture Diagram: To be added.

**7.2 Glossary**

* **FDA**: Food Delivery Application
* **GPS**: Global Positioning System
* **OTP**: One-Time Password
* **CRUD**: Operations for Create, Read, Update, Delete