HIGH LEVEL DOCUMENT FOR FOOD ODERING APPLICATION

This project presents the development of Food ordering Application, an innovative e-commerce platform engineered to compete with industry frontrunners in the online retail sector. Food Ordering Application is equipped with a comprehensive suite of features, including User Authentication, Product Discovery, Order Management, and Payment Handling, tailored to meet the diverse needs of modern consumers .The platform prioritizes security and efficiency, ensuring seamless transactions and order fulfilment processes.

**1.Introduction:**

Food Ordering Application is an advanced e-commerce platform offering a seamless shopping experience across diverse product categories. With a user-friendly interface, it provides easy navigation, personalized recommendations, and secure payment options. Food Ordering Application prioritizes security, partnering with trusted payment gateways for safe transaction. Food Ordering Application aims to revolutionize online retail by delivering convenience, reliability, and quality service to modern consumers.

**1.1.Purpose of this Document :**

This High-Level Design (HLD) document specifies the implementation, including inter-component dependencies, and provides sufficient design detail that any product based on this HLD will satisfy the product requirements.

**1.2.Document Scope:**

Anyone interested in understanding the Food Ordering Application Enhancements internal design should read this document.

**2.Requirement Summary:**

The new requirements for the Food Ordering Application platform encompass key aspects of network infrastructure, system functionalities, and user interactions. This breakdown provides a high-level overview to guide the development process effectively:

**2.1 Network Infrastructure:**

• Scalability: The platform should support an increasing number of users and products without compromising performance.

• Security Measures: Implement robust security protocols to safeguard user data, transactions, and ensure a secure browsing environment.

* 1. **System Functionalities:**

• User Authentication Enhancement: Strengthen user authentication mechanisms to ensure secure access and protect user accounts.

• Order Processing Optimization: Streamline order processing for faster checkout, order confirmation, and tracking.

* 1. **User Interactions:**

• Interactive User Interface: Enhance the user interface for intuitive navigation and an efficient shopping experience.

• Personalized Recommendations: Implement algorithms for personalized product recommendations based on user preferences and browsing history.

* 1. **Performance and Reliability:**

• Performance Optimization: Optimize system performance for efficient product browsing, order placement, and payment processing.

• System Reliability: Ensure high system availability and reliability to minimize downtime and enhance user trust.

**2.5 Accessibility:**

• Cross-Platform Compatibility: Ensure the platform is accessible and functions seamlessly across various devices and operating systems.

• Inclusive Design: Implement design features that enhance accessibility for users with different abilities.

**2.6 User Education and support:**

• User Guidance: Provide informative tooltips, guides, and support features to assist users in navigating and utilizing the platform effectively.

• Customer Support Integration: Implement efficient customer support mechanisms, including live chat and helpdesk features.

**3.Assumptions and Prerequisites :**

This document outlines key assumptions and foundational elements integral to the design process of Food Ordering Application. The purpose is to ensure clarity and alignment among stakeholders regarding the fundamental principles guiding the platform's development.

1. Internet Connectivity: It is assumed that users will have reliable internet access to engage with the Application seamlessly across various devices.
2. Device Compatibility: The Application is designed to be compatible with smartphones, tablets, and computers, ensuring accessibility for a wide range of users.
3. Product Availability: The assumption is made that products showcased on the Application are readily available for purchase, maintaining user trust and satisfaction.
4. Security Measures: Robust security measures will be implemented to safeguard user data and ensure secure transactions, maintaining the integrity of the Application's platform.
5. Payment Gateway Integration: Food Ordering Application's functionality relies on seamless integration with trusted payment gateways for efficient transaction processing and user convenience.
6. Server Infrastructure: Availability of stable server infrastructure is a prerequisite to support platform operations and ensure consistent performance.
7. Database Management: Food Ordering Application depends on a reliable database management system for efficient data storage, retrieval, and management, ensuring seamless user experiences.
8. Third-party Integrations: Integration with third-party services such as shipping providers and analytics platforms enhances Food Ordering Application's functionality and user experience.

These assumptions and foundational elements will be regularly reviewed and assessed throughout the design process to ensure their validity and alignment with Food Ordering Application's objectives.

**4.High-Level Design:**

**4.1. Multi Factor Authentication:**

Software Components:

• Frontend (Mobile/Web Interface)

• Backend Server

• Database (User Information)

Workflow:

• Users initiate the process by submitting login credentials through the frontend interface.

• Backend server validates credentials by cross-referencing with the user information database and verifies the user using highly efficient IAM using two factor Authentication.

• Upon successful verification, the backend generates a token and transmits it to the frontend for efficient session management.

**4.2. Secure Product Browsing :**

Software Components:

• Frontend (Mobile/Web Interface)

• Backend Server

• Database (Product Display)

Workflow:

• After successful login the User interface sends a request to retrieve available product data from the backend.

• Backend server retrieves comprehensive product information from the database.

• Product data is transmitted back to the frontend, facilitating an immersive display for users.

**4.3. Shopping Cart Management:**

Software Components:

• Frontend (Mobile/Web Interface)

• Backend Server • Database (User Cart)

Workflow:

• Users seamlessly add products to their carts through the frontend.

• Frontend communicates cart updates to the backend server.

• Backend ensures real-time updates to the user's cart within the database.

**4.4. Order Placement:**

Software Components:

• Frontend (Mobile/Web Interface)

• Backend Server

• Database (Orders)

Workflow:

• Users confirm their orders through the frontend interface.

• Frontend transmits order details to the backend server.

• Backend stores comprehensive order information securely in the database.

**4.5. Payment Processing:**

Software Components:

• Frontend (Mobile/Web Interface)

• Payment Gateway

• Backend Server

Workflow:

• Users select their preferred payment method and initiate the payment process through the frontend.

• Frontend communicates the payment request to the backend server.

• Backend forwards the request seamlessly to the designated payment gateway.

• The payment gateway processes the transaction securely.

• Confirmation of the successful payment is relayed back to both the backend and frontend.

This High-Level Design provides a holistic view of Food Ordering Application's foundational processes, aiding in the early assessment of assumptions, validations, and potential areas for further review. The integration 8 aspects with other systems are seamlessly incorporated, ensuring a robust and efficient design structure for the Food Ordering Application platform.

**5.System Architecture :**

**5.1 Frontend**

Mobile App: Cross-platform mobile application using AngularJS Native for a consistent user experience.

**5.2 Backend**

Server-side Logic: NodeJS and ExpressJS for handling server-side logic and API endpoints.

Database: MongoDB for efficient storage and retrieval of event and user data.

Authentication: JWT (JSON Web Tokens) for secure user authentication.

**5.3 Third-Party Integrations**

Payment Gateway: Integration with Stripe API for secure payment processing.

Notification Services: Utilize Twilio for SMS, SMTP for email, and Firebase Cloud Messaging for push notifications.

A diagram of a computer network

Description automatically generated