Requirement Gathering and Analysis Phase for Food Ordering Application

Date	05 -07-2024
Team ID	SWTID1720150432
Project Name	EagerEats - Food Ordering App
Maximum Marks	

Solution Architecture:

Solution Architecture for Food Ordering Application

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

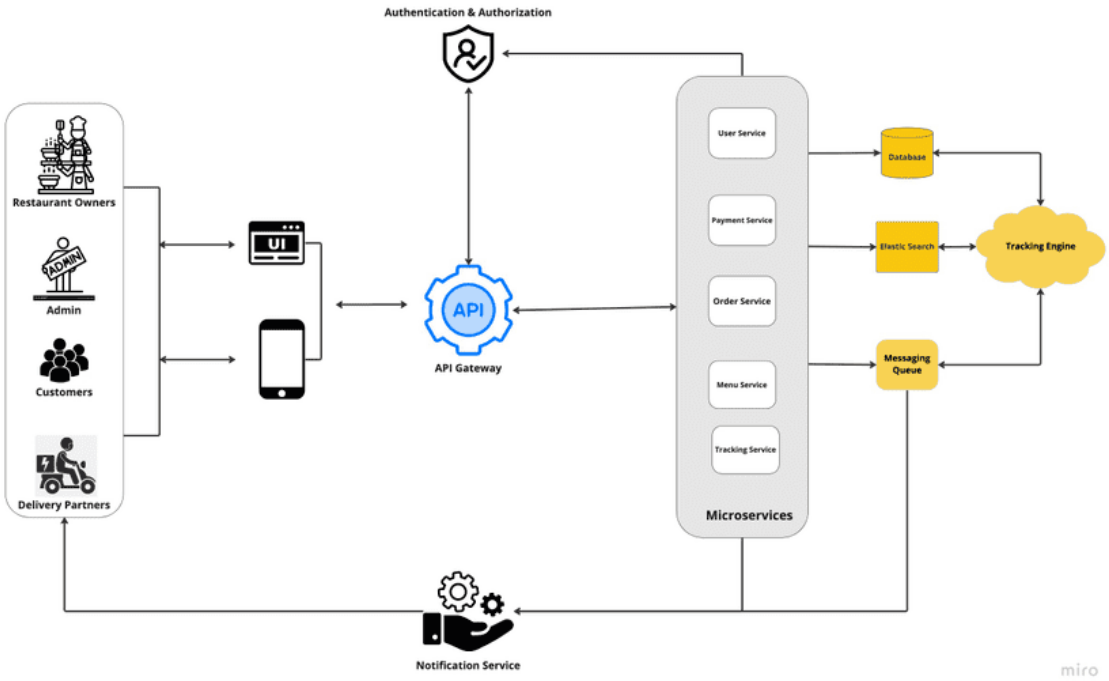
1. **Find the Best Tech Solution to Solve Existing Business Problems:**
    - **Business Problem:** Customers face difficulty in ordering food, tracking orders, and getting timely deliveries. Restaurants need an efficient way to manage orders, inventory, and deliveries.
    - **Tech Solution:** Develop a comprehensive food ordering application that includes:
        - **Customer App:** For customers to browse menus, place orders, make payments, and track deliveries.
        - **Restaurant Management System:** For restaurants to manage orders, update menus, track inventory, and handle customer feedback.
        - **Delivery Personnel App:** For delivery staff to receive delivery assignments, navigate to delivery locations, and update delivery statuses.
        - **Admin Dashboard:** For administrators to monitor the overall system, generate reports, and manage user roles and permissions.

2. **Describe the Structure, Characteristics, Behaviour, and Other Aspects of the Software to Project Stakeholders:**
    - **Structure:**
        - **Frontend:** Mobile apps (iOS and Android) for customers and delivery personnel, web interface for restaurant management.
        - **Backend:** Cloud-based server infrastructure to handle data processing, storage, and application logic.
    - **Characteristics:**
        - **Scalable:** Able to handle increased number of users and orders, especially during peak times.
        - **Responsive:** Quick load times and smooth user interactions.
        - **Secure:** Protect user data and ensure secure transactions.
    - **Behaviour:**
        - **Real-time Updates:** Orders and delivery statuses are updated in real-time.
        - **Push Notifications:** Customers and delivery personnel receive updates on order status changes.
        - **Data Analytics:** Provide insights into user behaviour, order patterns, and operational efficiency.

3. **Define Features, Development Phases, and Solution Requirements:**
    - **Features:**
        - **User Registration and Authentication:** Secure sign-up and login for customers, restaurant staff, and delivery personnel.
        - **Menu Browsing and Search:** Allow customers to browse restaurant menus and search for specific items.
        - **Order Placement and Payment:** Enable customers to place orders and pay through multiple payment options.
        - **Order Tracking:** Provide real-time tracking of orders from preparation to delivery.
        - **Feedback and Ratings:** Allow customers to rate their experience and provide feedback.
        - **Inventory Management:** Enable restaurants to manage inventory and receive alerts for low stock.
        - **Delivery Management:** Assign orders to delivery personnel and optimize delivery routes.
    - **Development Phases:**
        - **Phase 1:** Requirements gathering, feasibility study, and initial design.
        - **Phase 2:** Development of customer app and basic backend services.
        - **Phase 3:** Development of restaurant management system and integration with customer app.
        - **Phase 4:** Development of delivery personnel app and integration with the system.
        - **Phase 5:** Testing, security audits, and performance optimization.
        - **Phase 6:** Deployment, user training, and support.
    - **Solution Requirements:**
        - **Functional Requirements:** Specific functionalities that the application must provide.
        - **Non-functional Requirements:** Performance, security, usability, and scalability considerations.
        - **Technical Requirements:** Specific technologies, frameworks, and tools to be used in the development.

4. **Provide Specifications According to Which the Solution is Defined, Managed, and Delivered:**
    - **Specifications:**
        - **System Architecture Document:** Detailed documentation of the system architecture, including diagrams and component descriptions.
        - **API Documentation:** Specifications for APIs used for communication between frontend and backend.
        - **Security Plan:** Detailed plan outlining security measures and compliance requirements.
        - **Deployment Plan:** Step-by-step plan for deploying the application in a live environment.
        - **Maintenance Plan:** Guidelines for ongoing maintenance, updates, and support.

Solution Architecture Diagram:



Reference: https://pratapsharma.com.np/architecture-of-food-delivery-app