**High Level Solution Design Document**

**For an application like Zomato**

Contents

1.Overview and Introduction

2.Requirements

3. Assumptions and Pre-requisites

4. High Level Design

4.1. Architectural Modules

4.2. Use Case diagram

**1.Overview and Introduction:**

**High-level design**or**HLD**refers to the overall system, a design that consists description of the system architecture and design and is a generic system design that includes:

* System architecture
* Database design
* Brief description of systems, services, platforms, and relationships among modules.

HLD document consists of **data flows, flowcharts, and data structures** to help developers in understanding and implement how the current system is being designed intentionally to function.

High-Level Design (HLD) document for our online food ordering application, Zomato. This document serves as a blueprint for understanding the architecture and key components of our system. By outlining the high-level design, stakeholders can grasp the structure, interactions, and technologies driving our platform, ensuring a robust and efficient food ordering experience for users and restaurants alike.

**2.Requirements:**

To create a high-level design (HLD) for an application like Zomato, you would need to consider several key requirements:

**1. User Interface Design:** Design intuitive interfaces for customers, restaurants, and delivery partners. Include features for Login, viewing menus, placing orders, tracking deliveries, and providing feedback.

**2. User Authentication and Authorization:** Implement secure authentication mechanisms for users, including email/phone verification, social media login, and role-based access control for different user types.

**3. Order Management:** Create a system to manage orders from placement to delivery, including order tracking, payment processing, and notifications to users and delivery partners.

**4. Delivery Management:** Implement features for assigning orders to delivery partners, optimizing delivery routes, and tracking delivery status in real-time.

**5. Payment Gateway Integration:** Integrate secure payment gateways to facilitate online transactions, including credit/debit cards, digital wallets, and cash on delivery options.

**6. Location-Based Services:** Utilize GPS and mapping APIs to provide accurate location-based services, including restaurant discovery, order tracking, and delivery estimation.

**3.Assumptions and Pre-requisites :**

Creating a high-level design (HLD) for an application like Zomato involves considering various assumptions and prerequisites. Here are some key ones:

**1. Functional Requirements:** The core functionalities of Zomato applications, such as user registration, menu display, ordering, payment processing, delivery tracking, user reviews, etc.

**2. Scalability:** Assume the application will need to handle a large volume of concurrent users, restaurant listings, orders, and reviews. Design the system to scale horizontally and vertically to meet increasing demand.

**3. Performance**: Design the system to be responsive, with low latency for critical operations like search, order placement, and payment processing.

**4. Security:** Implement robust security measures to protect user data, payment information, and the integrity of the platform. Consider encryption, authentication, authorization, and secure communication protocols.

**5. Integration:** Assume integration with various third-party services such as payment gateways, mapping services for location-based features, SMS/email services for notifications, etc.

**6. Data Management :** Design a robust data model to efficiently store and retrieve information related to users, restaurants, menus, orders, reviews, etc. Consider database sharing, replication, and caching strategies.

**4.High Level Design :**

**4.1. Architectural Modules:**

To build an application like Zomato, you would need several architectural modules:

**1. User Interface :** This includes screens, navigation, and user interactions for customers, restaurants, and delivery personnel.

**2. User Authentication and Authorization:** Module for managing user registration, login, and permissions.

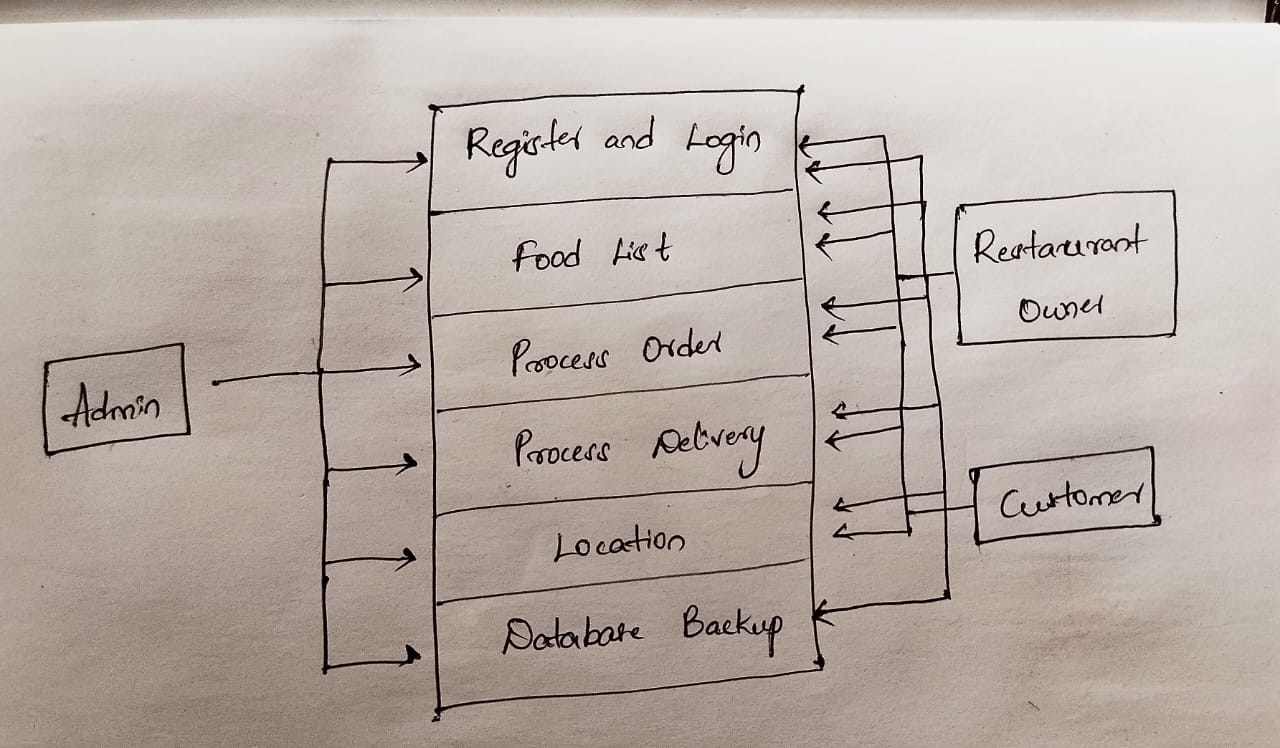
**3. Order and Delivery Management:** Handles the process of placing, tracking, and managing orders from customers to restaurants and delivery personnel.

**6. Payment Gateway Integration:** Integration with payment gateways to facilitate secure transactions for orders.

**9. Geolocation Services:** Integration with mapping services to provide location-based features like restaurant search, delivery tracking, etc.

Each of these modules would have its own set of functionalities and would interact with others to provide the overall experience of the application.

**4.2. Use Case diagram :**

****