**Low Level Solution Design Document**

**For an application like Zomato**

Meghamala Alajangi

Contents

1.Overview and Introduction

2.Requirements

3. Assumptions and Pre-requisites

4. Low Level Design

4.1. Architectural Modules

**1.Overview and Introduction:**

A Low-level Design or an LLD application typically refers to the LLVM linker, which is a tool used to link object files and libraries together to create an executable or shared library. LLD stands LLVM Linker Driver. It’s a component of the LLVM project, providing a faster and more modern alternative to traditional linkers like GNU Id.

An LLD document, or Low-Level Design document, is a detailed technical document that outlines the design specifications of a software system or application at a granular level. It typically follows the High-Level Design (HLD) document and provides a more in-depth view of the system's architecture, components, modules, data structures, algorithms, interfaces, and interactions.

The LLD document serves as a guide for developers to implement the system accurately according to the specified design. It is often created by software architects or senior developers and serves as a reference for the development team throughout the implementation phase.

**Trivial Small:**

Small Scale

Low uncertainty

**Novel Large:**

Small Scale

High uncertainty

High customer interactions

**2.Requirements:**

To create a high-level design (LLD) 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 (LLD) 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.Low Level Design :**

**4.1. Architectural Modules :**

For a low-level design (LLD) of an application like Zomato, you'd focus more on the detailed design of individual components within the system. Here are the architectural modules you might consider:

**User Management Module:** Responsible for handling user registration, login, and profile management. Includes functionalities for authentication, authorization, and user data storage.

**Order Processing Module:** Facilitates the process of placing, tracking, and managing orders. Coordinates communication between users, restaurants, and delivery personnel.

**Payment Processing Module:** Integrates with payment gateways to handle secure transactions. Manages payment processing and transaction records.

**Geolocation Module:** Integrates with mapping services to provide location-based functionalities.Facilitates features like restaurant search, delivery tracking, etc.