System Architecture Diagram

Actors / Clients

- **Customer** (Login & Registration)
- **User** (Restaurant Listing & Search)
- **Backend Team** (Restaurant Listing & Search)
- **User** (Order & Tracking)
- **Delivery Executive** (Notification for Order)
- **User** (Review & Rating)

Devices

- **UI**
- **Phone**

API GATE WAY

- **Authentication & Authorization**

Services

- Login Service
- Registration Service
- Search Service
- Restaurant Listing Service
- Location Service
- Ordering Service
- Payment Service
- Location Update Service
- Order notification Service
- Delivery Assignment Service
- Review & Rating Service

Databases

- Data base (Login)
- Search Data base
- User Data base
- Data base (Order notification)
- Data base (Review & Rating)

Feature 1: User Management

- **Trivial + Small**:
  - Factor Explanation: User management is a common feature in most applications, including authentication and user profile management.
  - Handles user **registration** and **login** processes securely.
  - Utilizes industry-standard authentication protocols like OAuth 2.0 or JWT.
  - Integrates with social media login options for user convenience.
  - Why?: Implementing user management involves well-established practices and existing libraries/frameworks for authentication and profile management. The scope of this feature is relatively small compared to other complex functionalities.

Feature 2: Restaurant Listing and Search

- **Novel + Large**:
  - Factor Explanation: While restaurant listing and search are common features, the scale and complexity of implementing them in a platform like Zomato can be substantial.
  - Implement a search functionality **using filters for location**, cuisine ,etc
  - Use **Google map API** for location based search.
  - Why?: Integrating various data sources for restaurant information, implementing advanced search algorithms, and ensuring real-time updates for a large number of restaurants contribute to the novelty and scale of this feature.

Feature 3: Ordering and Tracking

- **Novel + Large**:
  - Factor Explanation: Handling online orders and tracking logistics at scale involves complex real-time processing and integration with external services.
  - Facilitates the process of placing, updating, and cancelling orders.
  - Allows users to track the status of their orders in real-time from placement to delivery.
  - Why?: Managing order placement, modification, and tracking in real-time, along with coordinating with delivery partners and ensuring timely delivery, makes this feature both novel and large in scope.

Feature 4: Notifications

- **Trivial + Small**:
    - Factor Explanation: Sending notifications to users based on specific events is a common feature, and there are established solutions and services available for this purpose.
    - Sends real-time notifications to delivery executives regarding new orders. Utilizes push notifications or SMS for instant communication.
    - Stores information about available delivery executives.
    - Handles assignment and tracking of delivery executives to orders.
    - Why?: Implementing basic notification functionalities such as sending emails, push notifications, or SMS messages based on predefined triggers is relatively straightforward and doesn't involve significant complexity or innovation.

Feature 5: Reviews and Ratings

- **Trivial + Large**:
    - Factor Explanation: While reviews and ratings are standard features, managing a large volume of user-generated content and ensuring data integrity can be challenging.
    - Why?: The core functionality of allowing users to leave reviews and ratings is straightforward. However, implementing features like moderation, spam detection, and ensuring the accuracy and trustworthiness of reviews add complexity to the feature.

- **Trivial + Small**: Features that are common and have well-established solutions, with a relatively small scope in terms of implementation effort.
- **Trivial + Large**: Features that are standard but involve managing a large scale or volume of data, leading to complexity in implementation.
- **Novel + Small**: Features that introduce new or innovative functionality but have a relatively small scope or impact on the overall system.
- **Novel + Large**: Features that are both innovative and have a significant scope, requiring substantial effort to implement due to their complexity and scale.