**Zomato App High Level Design & Low Level Design**

**High Level Design:**

1)onboard a restaurant

2)onboard client

3)inventory management

4)order management

5)delivery management

6)notification

Inventory management

Order management service

User service

Delivery management service

Payment

Notification

**High level diagram:**

Start

Browse Restaurants

View Restaurant Details

Browse Menu & Order

Review Order & Checkout

Make Payment

Track Order & Delivery

Receive Order & provide Feedback

End

1. **Start**: The user launches the application.
2. **Browse Restaurants**: The user browses through a list of available restaurants.
3. **View restaurant Details**: The user selects a restaurant and views its details, including menu items and reviews.
4. Browse Menu & place order: The user selects menu items and places an order.
5. **Review order & checkout:** The user reviews the order details and proceeds to checkout.
6. **Make Payment:** The user makes payment for the order.
7. **Track order & delivery:** The user tracks the status of the order and delivery.
8. **Receive order & provide feedback:** The user receives the order, provides feedback, and completes the transaction.
9. **End:** The user completes the interaction with the application.

**Low Level Design**

**Requirements:**

1. Users needs to login into our application.

1. Search restaurants based on location.
2. List all the food items for a particular restaurant.
3. Placing a particular order.
4. Make payment.
5. Giving notification to the user.
6. Live tracking of the food

Non-functional Requirements

1. System should be scalable.

2. Reliable/Modular/Maintainable

Main entity

1. User(Account)
2. Customer
3. Restaurant
4. Food Items
5. Cart
6. Payment
7. Orders
8. Notification System

Low level design

class Restaurant {

String name;

String Description;

Address address;

double rating;

…

Menu menu;

+updateRestaurantInfo()

+updatemenu(Menu menu);

..

}

class FoodItem{

String name;

String description;

float rating;

int price;

}

class Address{

string address;

long pincode;

long latitude;

long longitude;

}

class Order {

long orderId;

Date orderDate;

Time orderTime;

Cart id;

Card details;

Status status;

…

}

Notification Server

class FoodOrdermanagementSysem {

List<Restaurant>restaurants;

List<Customers>customers;

..

HashMap<Location,Restaurant> rest;

+signUpsignIn();

+searchRestaurantByLoc(Location address);

+listAllFoodItems(Restaurant rest)

+placeOrder(Order order)

+sendNotification(Order details);

}

class FoodItemDetails{

FoodItem food;

String description;

Int qty;

Restaurant ;

}

class Cart {

long cartId;

List <FoodItemDetails> foods;

}

class Customer{

User user;

List <Order> orders;

Cart cart;

.

+updateInfo();

+placeOder(Order id);

..

}

class User {

string name;

Address address;

String email;

String password;

longphoneNumber;

}

class Menu {

List <FoodItems> menuItem;

}