# 

DOCUMENTATION

# Online Food Ordering using foodies

**Group No.:** 12

**Contributors:**

Urvish Mehta (202312010) (Group Leader)

Mansi Dhokiya(202312034)(group member)

Vedansh pandey (202312091) (group member)

Rahul gupta(202312085) (group member)

**Table of Contents**

1. Introduction  
   1.1 Purpose  
   1.2 Introduction to Food Delivery App  
      1.2.1 Product Description  
      1.2.2 Product Features  
   1.3 Scope of System  
   1.4 Assumptions

2. User Stories  
   2.1 Admin  
   2.2 User

3. Use Cases  
   3.1 Admin Use Cases  
      3.1.1 Admin Login  
      3.1.2 Manage Restaurant Listings  
      3.1.3 View and Manage Pending Orders  
      3.1.4 Analyze Popular Food Iteam

3.1.5 Register Restaurant  
      3.1.6 Manage Restaurant Menu  
      3.1.7Receive Order Notifications and Update Status  
      3.1.8 View Customer Reviews and Ratings

   3.2 User Use Cases  
      3.2.1 User Registration  
      3.2.2 View Ratings and Reviews  
      3.2.3 Select Food Type and Filter Restaurants  
      3.2.4 Track Real-time Order Status  
      3.2.5 Provide Reviews and Ratings  
      3.2.6 Add Favorites for Future Orders  
      3.2.7 User Profile Deletion

4. Functional Requirements  
   4.1 Admin Module  
      4.1.1 Admin Login  
      4.1.2 Manage Restaurant Listings  
      4.1.3 View and Manage Pending Orders  
      4.1.4 Analyze Popular Food Item   
      4.1.5 Register Restaurant  
      4.1.6 Manage Restaurant Menu  
      4.1.7 Receive Order Notifications and Update Status  
      4.1.8 View Customer Reviews and Ratings

   4.2 User Module  
      4.2.1 User Registration  
      4.2.2 View Ratings and Reviews  
      4.2.3 Select Food Type and Filter Restaurants  
      4.2.4 Track Real-time Order Status  
      4.2.5 Provide Reviews and Ratings  
      4.2.6 Add Favorites for Future Orders  
      4.2.7 User Profile Deletion

5. Non-Functional Requirements  
   5.1 Performance  
      5.1.1 Response Time  
   5.2 Reliability  
      5.2.1 System Availability  
      5.2.2 Data Backup  
   5.3 Security  
      5.3.1 User Authentication  
      5.3.2 Data Encryption  
   5.4 Scalability  
      5.4.1 User Growth  
      5.4.2 Performance under Load  
   5.5 Usability  
      5.5.1 Intuitive User Interface  
   5.6 Compatibility  
      5.6.1 Browser and Device Compatibility  
   5.7 Maintainability  
      5.7.1 Modular Design  
      5.7.2 Coding Standards

6. diagrams

6.1 use case diagram  
   6.2 E-R diagram  
   6.3 class diagram  
   6.4 activity diagram

6.5sequence diagrm

7.DFD

6.1 level 0  
  6.2 level 1  
  6.3 level 2

## 1. Introduction

### 1.1 Purpose

The purpose of this document is to outline the requirements for the development of a Food Delivery Application.

### 1.2 Introduction to Food Delivery App

#### 1.2.1 Product Description

The Food Delivery App is a platform that enables users to browse, order, and receive food from various restaurants through a user-friendly interface. The application also provides features for restaurant owners to manage their menus, receive orders, and track performance.

#### 1.2.2 Product Features

* User registration and authentication
* Restaurant management
* Menu browsing and selection
* Order placement and payment processing
* Real-time order tracking

### 1.3 Scope of System

The system encompasses user interfaces for customers, restaurant owners, and administrators, facilitating interactions and transactions between these entities. The application will be available on web and mobile platforms.

### 1.4 Assumptions

* Users have access to a stable internet connection.
* Payment gateway integration relies on third-party services.

## 2. User Stories

### 2.1 Admin

1. As an admin, I want to log in to access various features.
2. As an admin, I want to manage restaurant listings, including adding, updating, or removing them.
3. As an admin, I want to view and manage pending orders from different restaurants.
4. As an admin, I want to analyze popular food items and trending restaurants.
5. As a admin, I want to register my restaurant with the app.
6. As a admin, I want to manage my restaurant's menu, including adding new items and updating prices.
7. As a admin, I want to receive order notifications and update order statuses.

### 2.2 User

1. As a first-time customer, I want to create an account easily and securely, with the ability to save payment and delivery information for future orders, reducing the need for repetitive data entry.
2. As a customer exploring new restaurants, I want to view ratings and reviews from other users, empowering me to make informed decisions about what to order.
3. As a hungry customer, I want to select what type of food I want to eat and filter restaurants accordingly.
4. As a hungry customer, I want to select restaurants which are closest to me so I can get my food faster.
5. As a customer, I want to add food items which I want to eat from a particular restaurant to my cart to review later and order.
6. As a customer, I want to edit my added items in the cart, add, delete, or edit quantity.
7. As a user, I want to see what offers are there so I can get discounts on my order.
8. As a customer ordering food for delivery, I want to track the status of my order in real-time and receive updates on the estimated delivery time, ensuring I know when to expect my food.
9. As a user, I want to give a review to my previous order and give a rating to food and restaurants.
10. As a user, I want to add a particular food item or restaurants to my favorite list for future orders.
11. As a user, I want to delete my profile for not further using it.

### 3. Use Cases

### 3.1 Admin Use Cases

#### 3.1.1 Admin Login

* Primary Actor: Admin
* Description: The admin logs in to access the admin dashboard.
* Preconditions: Admin credentials are valid.
* Postconditions: Admin gains access to administrative features.

#### 3.1.2 Manage Restaurant Listings

* Primary Actor: Admin
* Description: The admin adds, updates, or removes restaurant listings.
* Preconditions: Admin is logged in.
* Postconditions: Restaurant listings are updated accordingly.

#### 3.1.3 View and Manage Pending Orders

* Primary Actor: Admin
* Description: The admin views and manages pending orders from various restaurants.
* Preconditions: Admin is logged in.
* Postconditions: Order statuses are updated as per admin actions.

#### 3.1.4 Analyze Popular Food Items

* Primary Actor: Admin
* Description: The admin analyzes data to identify popular food items and trending restaurants.
* Preconditions: Admin is logged in.
* Postconditions: Analytical insights are provided.

#### 3.1.5 Register Restaurant

* Primary Actor: Admin Description: A restaurant owner registers their restaurant with the app.
* Preconditions: Restaurant owner has necessary information.
* Postconditions: Restaurant is successfully registered.

#### 3.1.6 Manage Restaurant Menu

* Primary Actor: Admin
* Description: A restaurant owner manages the restaurant's menu, including adding new items and updating prices.
* Preconditions: Restaurant owner is logged in.
* Postconditions: Menu is updated accordingly.

#### 3.1.7Receive Order Notifications and Update Status

* Primary Actor: Admin
* Description: A restaurant owner receives order notifications and updates the status of orders.
* Preconditions: Restaurant owner is logged in.
* Postconditions: Order status is updated.

#### 3.1.8 View Customer Reviews and Ratings

* Primary Actor: Admin
* Description: A restaurant owner views customer reviews and ratings for their restaurant.
* Preconditions: Restaurant owner is logged in.
* Postconditions: Reviews and ratings are visible.

### 3.2 User Use Cases

#### 3.2.1 User Registration

* Primary Actor: User
* Description: A user creates an account securely for the first time.
* Preconditions: User has access to the registration page.
* Postconditions: User receives confirmation of successful registration.

#### 3.2.2 View Ratings and Reviews

* Primary Actor: User
* Description: A user explores ratings and reviews for restaurants and food items.
* Preconditions: User is logged in.
* Postconditions: User makes informed decisions based on reviews.

#### 3.2.3 Select Food Type and Filter Restaurants

* Primary Actor: User
* Description: A user selects a food type and filters restaurants accordingly.
* Preconditions: User is logged in.
* Postconditions: User views a curated list of restaurants.

#### 3.2.4 Track Real-time Order Status

* Primary Actor: User
* Description: A user tracks the real-time status of their order and receives estimated delivery updates.
* Preconditions: User has placed an order.
* Postconditions: User is informed about the order status.

#### 3.2.5 Provide Reviews and Ratings

* Primary Actor: User
* Description: A user provides reviews and ratings for previous orders.
* Preconditions: User has completed an order.
* Postconditions: User feedback is submitted.

#### 3.2.6 Add Favorites for Future Orders

* Primary Actor: User
* Description: A user adds favorite food items and restaurants for future orders.
* Preconditions: User is logged in.
* Postconditions: Favorites are saved for easy access.

## 4. Functional Requirements

### 4.1 Admin Module

#### 4.1.1 Admin Login:

* **Description:**
  + The system shall provide secure login functionality for administrators to access the admin dashboard.
* **Preconditions:**
  + Admin credentials are valid.
* **Postconditions:**
  + Admin gains access to administrative features.

#### 4.1.2 Manage Restaurant Listings:

* **Description:**
  + The system shall allow admins to add, update, or remove restaurant listings.
* **Preconditions:**
  + Admin is logged in.
* **Postconditions:**
  + Restaurant listings are updated accordingly.

#### 4.1.3 View and Manage Pending Orders:

* **Description:**
  + The system shall enable admins to view and manage pending orders from various restaurants.
* **Preconditions:**
  + Admin is logged in.
* **Postconditions:**
  + Order statuses are updated as per admin actions.

#### 4.1.4 Analyze Popular Food Items:

* **Description:**
  + The system shall provide analytical tools for admins to analyze data and identify popular food items and trending restaurants.
* **Preconditions:**
  + Admin is logged in.
* **Postconditions:**
  + Analytical insights are provided.

#### 4.1.5 Register Restaurant:

* **Description:**
  + The system shall allow restaurant owners to register their restaurants with the app.
* **Preconditions:**
  + Restaurant owner has necessary information.
* **Postconditions:**
  + Restaurant is successfully registered.

#### 4.1.6 Manage Restaurant Menu:

* **Description:**
  + The system shall enable restaurant owners to manage the restaurant's menu, including adding new items and updating prices.
* **Preconditions:**
  + Restaurant owner is logged in.
* **Postconditions:**
  + Menu is updated accordingly.

#### 4.1.7 Receive Order Notifications and Update Status:

* **Description:**
  + The system shall notify restaurant owners of new orders and allow them to update the status of orders.
* **Preconditions:**
  + Restaurant owner is logged in.
* **Postconditions:**
  + Order status is updated.

#### 4.1.8 View Customer Reviews and Ratings:

* **Description:**
  + The system shall allow restaurant owners to view customer reviews and ratings for their restaurant.
* **Preconditions:**
  + Restaurant owner is logged in.
* **Postconditions:**
  + Reviews and ratings are visible.

### 4.2 User Module

#### 4.2.1 User Registration:

* **Description:**
  + The system shall allow users to create accounts securely for the first time.
* **Preconditions:**
  + User has access to the registration page.
* **Postconditions:**
  + User receives confirmation of successful registration.

#### 4.2.2 View Ratings and Reviews:

* **Description:**
  + The system shall enable users to explore ratings and reviews for restaurants and food items.
* **Preconditions:**
  + User is logged in.
* **Postconditions:**
  + User makes informed decisions based on reviews.

#### 4.2.3 Select Food Type and Filter Restaurants:

* **Description:**
  + The system shall allow users to select a food type and filter restaurants accordingly.
* **Preconditions:**
  + User is logged in.
* **Postconditions:**
  + User views a curated list of restaurants.

#### 4.2.4 Track Real-time Order Status:

* **Description:**
  + The system shall allow users to track the real-time status of their orders and receive estimated delivery updates.
* **Preconditions:**
  + User has placed an order.
* **Postconditions:**
  + User is informed about the order status.

#### 4.2.5 Provide Reviews and Ratings:

* **Description:**
  + The system shall enable users to provide reviews and ratings for previous orders.
* **Preconditions:**
  + User has completed an order.
* **Postconditions:**
  + User feedback is submitted.

#### 4.2.6 Add Favorites for Future Orders:

* **Description:**
  + The system shall allow users to add favorite food items and restaurants for future orders.
* **Preconditions:**
  + User is logged in.
* **Postconditions:**
  + Favorites are saved for easy access.

#### 4.2.7 User Profile Deletion:

* **Description:**
  + The system shall provide users with the option to delete their profiles if they no longer intend to use the platform.
* **Preconditions:**
  + User has an account.
* **Postconditions:**
  + User no longer has access to the application using the same credentials.

## 5. Non-Functional Requirements

### 5.1 Performance:

* **5.1.1 Response Time:**
  + The system must respond promptly to user interactions, including registration, login, and dashboard access.
  + Dashboard loading times should be optimized to provide a seamless user experience.
  + Updates and saves of information should occur with minimal response time.

### 5.2 Reliability:

* **5.2.1 System Availability:**
  + The system should ensure high availability, allowing users to access and utilize it without experiencing unplanned downtime.
* **5.2.2 Data Backup:**
  + User data, including card details and registration information, must be regularly backed up to prevent data loss.
  + Backups should be stored securely to maintain data integrity.

### 5.3 Security:

* **5.3.1 User Authentication:**
  + The system must implement robust user authentication procedures to prevent unauthorized access to user accounts.
  + Multi-factor authentication should be incorporated for enhanced security.
* **5.3.2 Data Encryption:**
  + The system must utilize HTTPS encryption to secure data transfer between the user's browser and the server.
  + Encryption protocols must align with industry standards to protect sensitive information.

### 5.4 Scalability:

* **5.4.1 User Growth:**
  + The system should support an increasing number of user registrations, logins, and concurrent dashboard visits.
  + Infrastructure must be scalable to accommodate rising traffic and user data storage requirements.
* **5.4.2 Performance under Load:**
  + The system should maintain optimal performance even under heavy loads, ensuring responsiveness during peak usage times.

### 5.5 Usability:

* **5.5.1 Intuitive User Interface:**
  + The user interface must be intuitive and user-friendly, especially for controlling website content, changing personal information, and choosing themes.
  + Clear instructions and tooltips should be provided to assist users in navigating the dashboard, login procedures, and the registration process.

### 5.6 Compatibility:

* **5.6.1 Browser and Device Compatibility:**
  + The system must seamlessly work with a variety of web browsers and devices to ensure accessibility for all users.
  + Compatibility testing should encompass various platforms and screen sizes to confirm functionality.

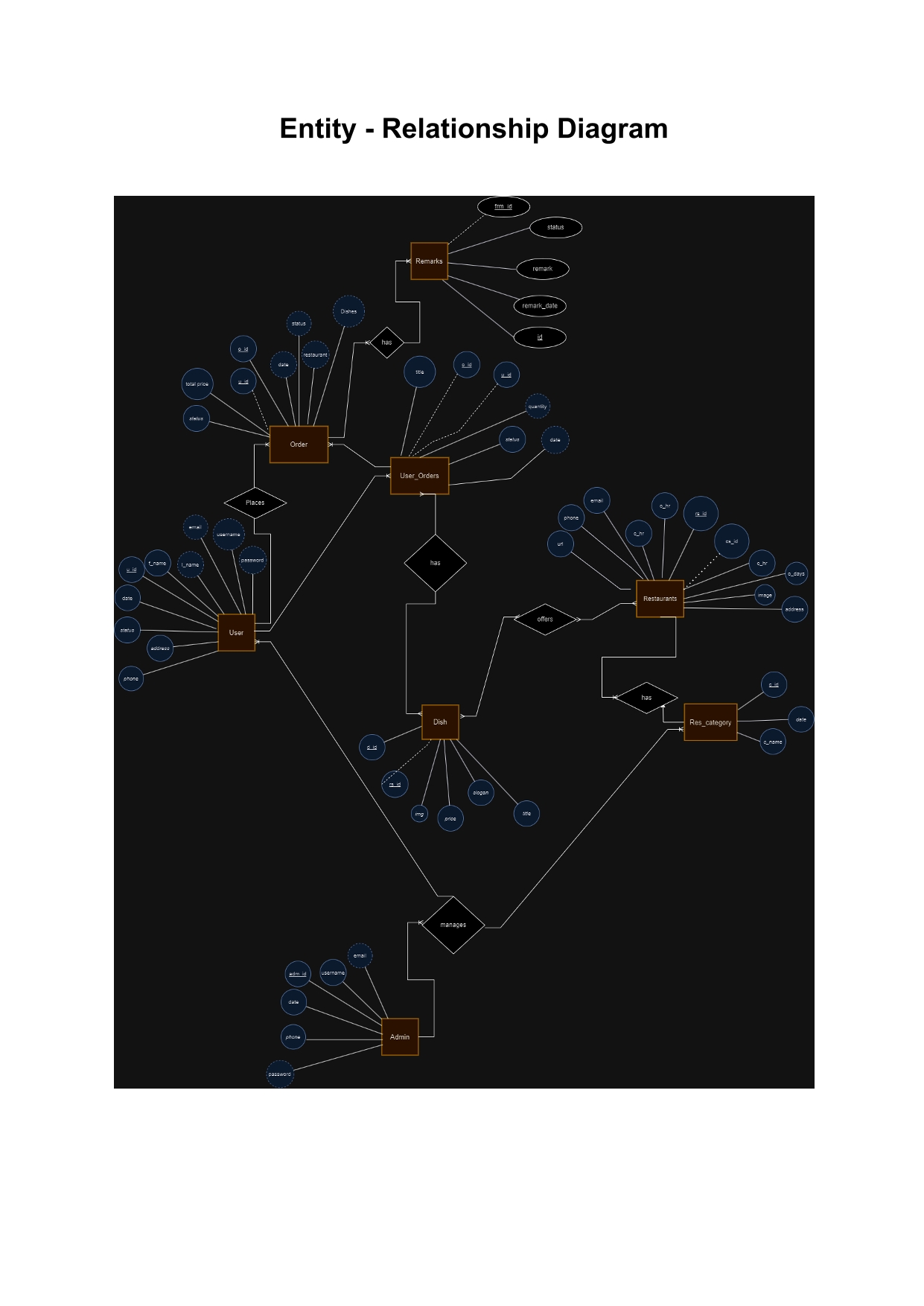
### 5.7 Maintainability:

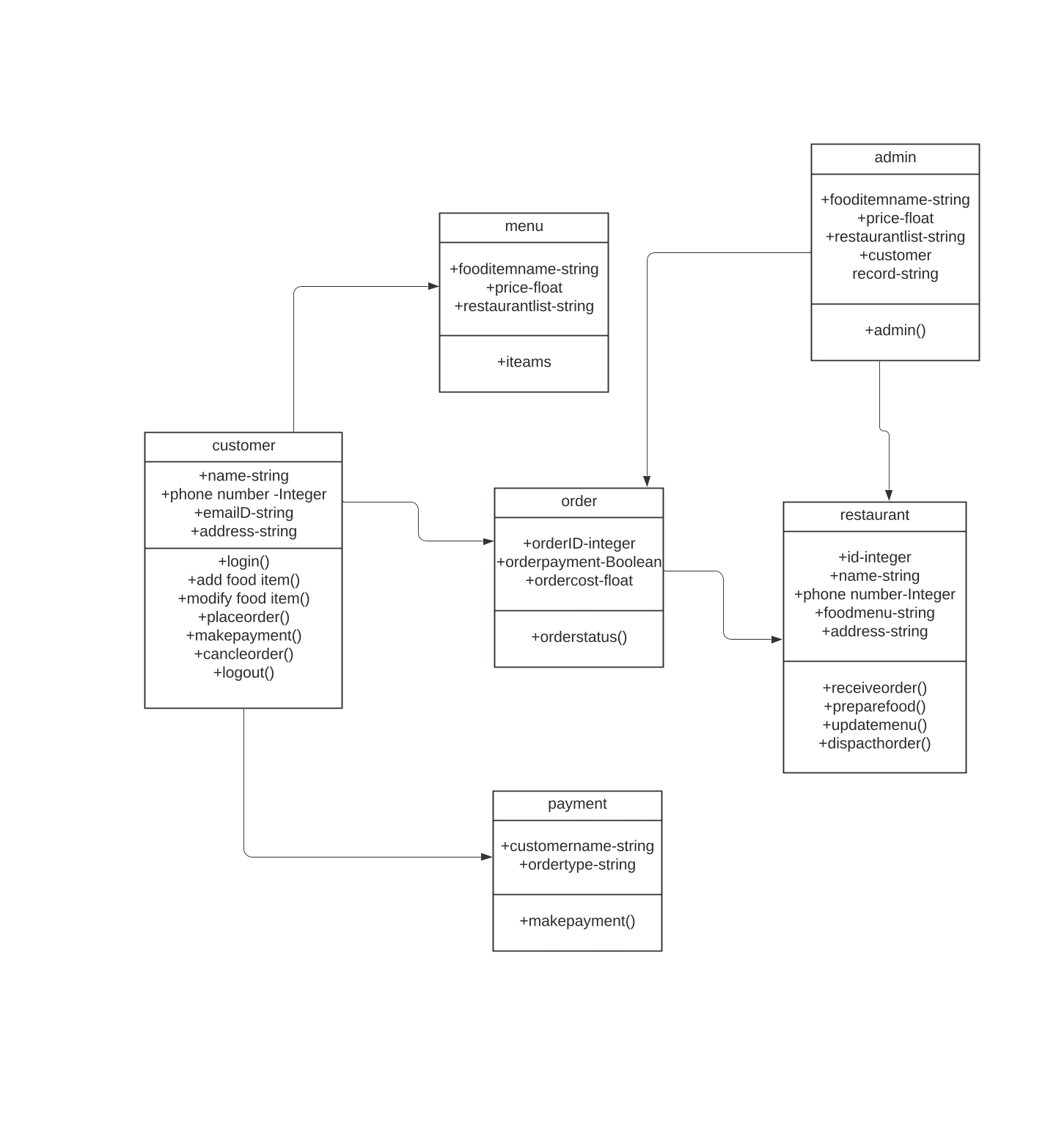
* **5.7.1 Modular Design:**
  + The system design should be modular to facilitate easy maintenance and future modifications.
  + Each module or component must be independent and well-documented for clarity.
* **5.7.2 Coding Standards:**
  + The codebase must adhere to coding standards and best practices to simplify troubleshooting and debugging procedures.
  + Thorough documentation should accompany the codebase to aid in future development and maintenance efforts.

6. diagrams

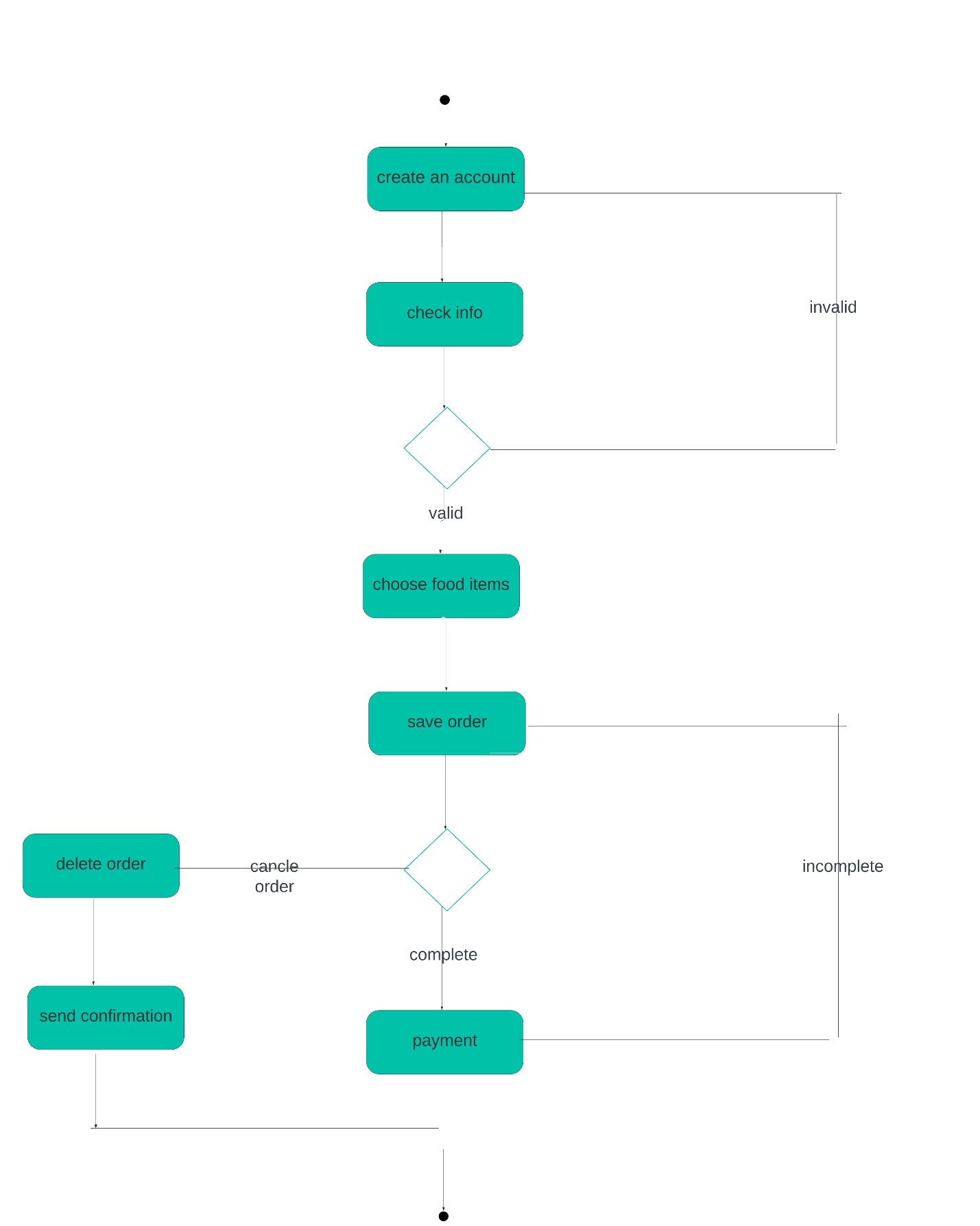
6.1 use case diagram



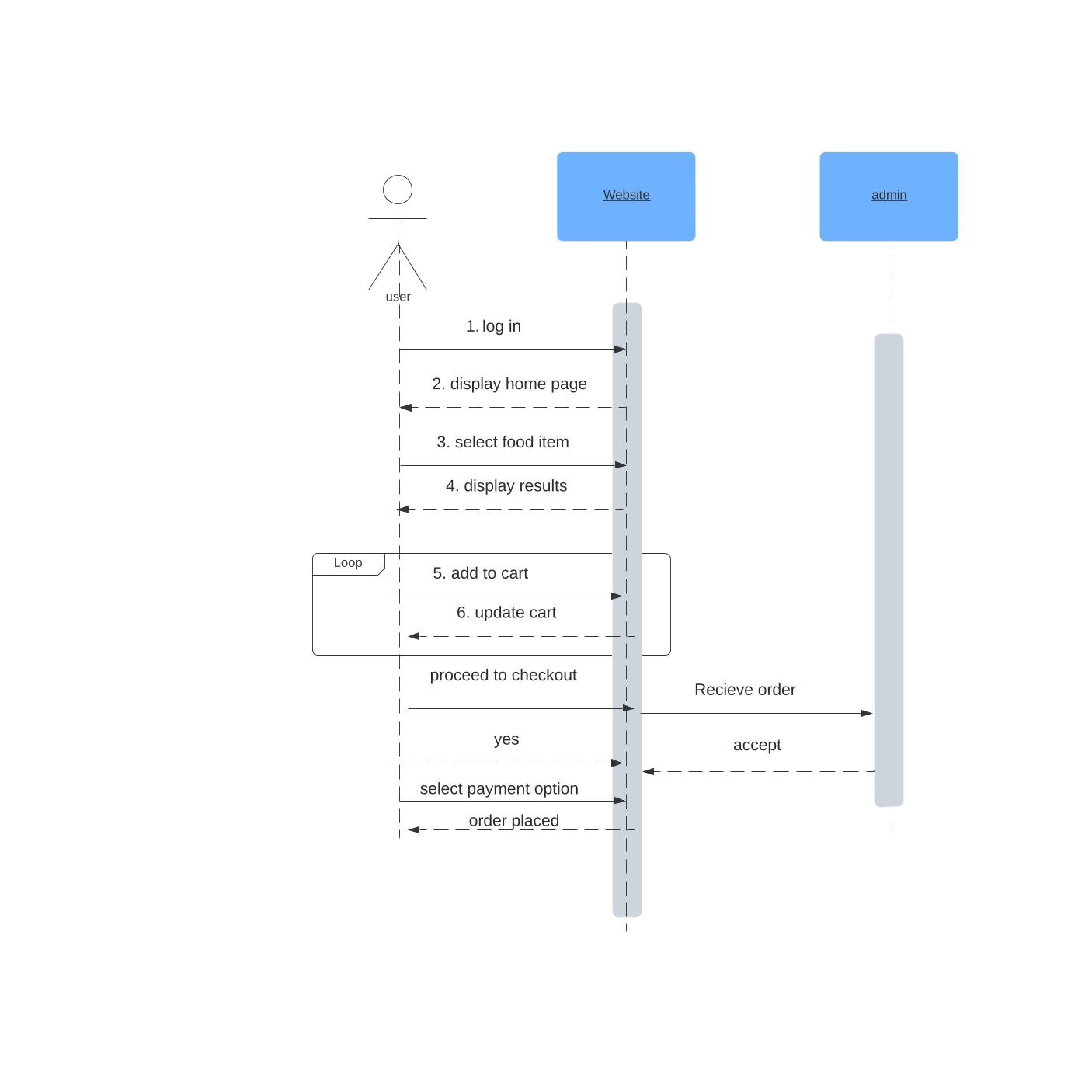
  
   6.3 class diagram



 6.4 activity diagram

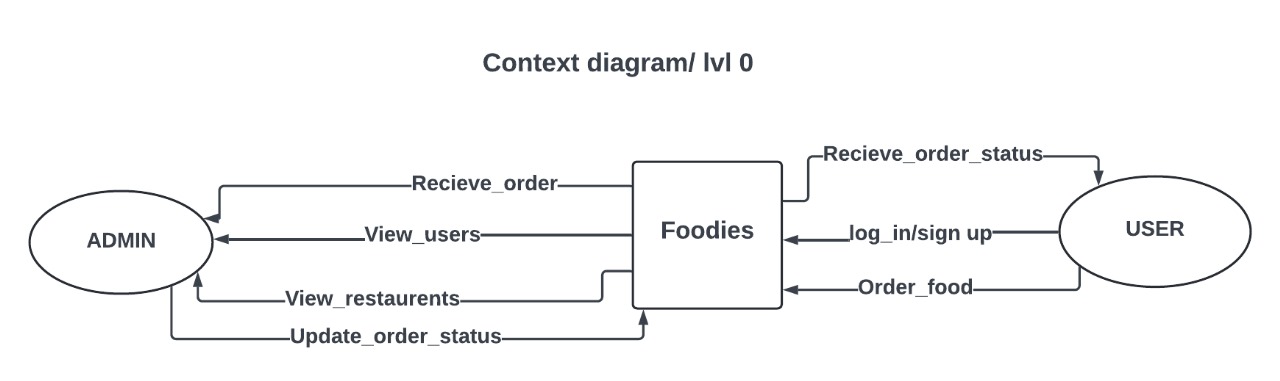


6.5 sequence diagram

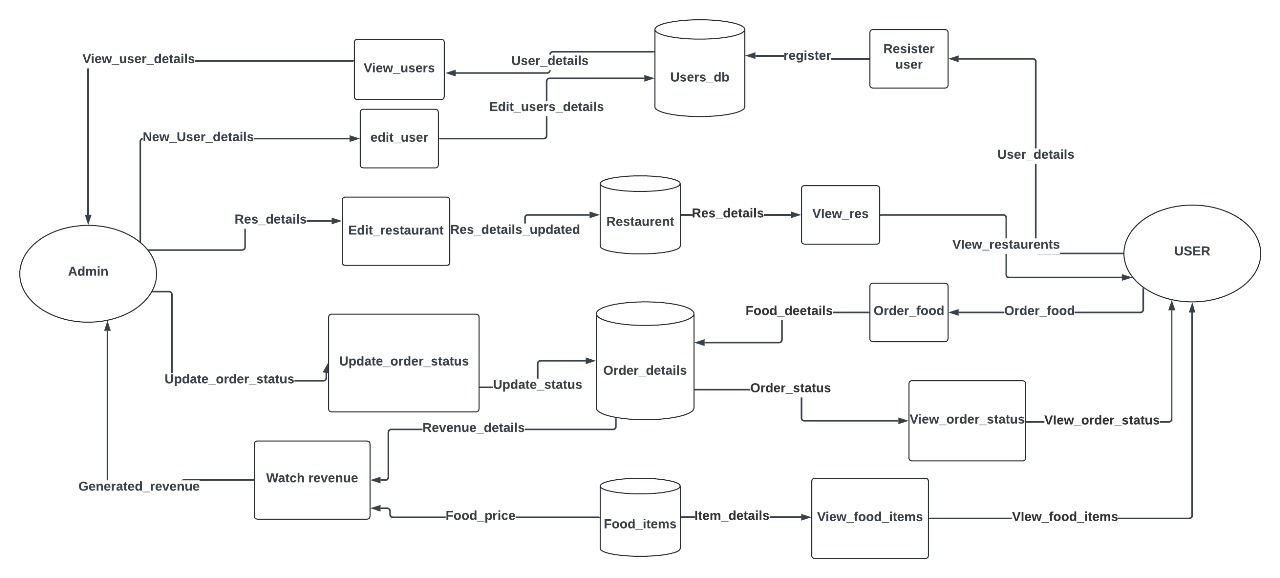


6.DFD

6.1 level 0



6.2 level 1



6.3 level 2

