**Project Description**

The Food Delivery App is a mobile-based application that enables customers to order food from nearby restaurants and have it delivered to their doorstep. The system provides a simple and user-friendly interface where customers can register, browse menus, add items to the cart, place orders, make payments, and track deliveries in real time.

Restaurants can receive and manage customer orders, update order status, and maintain menu details, while delivery personnel are assigned tasks to pick up and deliver orders efficiently. The application also includes a secure payment module for processing online transactions.

Additionally, an admin panel manages users, restaurants, and delivery staff, ensuring smooth system operations. The project demonstrates the practical implementation of software engineering concepts, including modular design, data flow, and interaction between multiple stakeholders.

**Premble**

Module Description :

1. User Authentication Module : secure login/register (user/admin)
2. Product Module : Add/View/Edit/Delete products (admin) Product categories, descriptions, images, stock
3. Cart : Add to cart, update quantity
4. Orders Module : Place orders

5. Admin Module : Dashboard to manage products and orders

**Litrature Review**

1. Ease of Use and Usefulness

According to the Technology Acceptance Model (TAM), people are more likely to use apps if they are simple and useful. In food apps, this means menus should be clear, the cart should update quickly, and checkout should be easy. Features like “Add to Cart” buttons, visible prices, and quick navigation improve user satisfaction.

2. Trust and Security

Users worry about payment failures, data privacy, and wrong or late orders. Trust is very important for adoption. Secure payment gateways, cash-on-delivery options, order tracking, and clear refund policies make customers feel safe. Without trust, people hesitate to order online.

3. User Interface (UI) and Cart Flow

A good user interface keeps users engaged. Best practices include showing a cart icon at all times, quantity +/– buttons, and a clear total price with taxes and fees. Allowing guest checkout (without forcing sign-up first) reduces drop-offs. In your app design, the simple menu and cart are good examples of this.

4. Payment Options

Multiple payment choices—like UPI, credit/debit cards, wallets, and cash-on-delivery—help users complete their orders. A retry option for failed transactions is also useful. If payments are smooth, users are more likely to return.

5. Personalization and Recommendations

Many studies show that personalized features, such as “Recent Orders” or “Recommended Items,” increase repeat usage. Your app screenshots already show “Recent Buy Items,” which is a strong feature to improve loyalty.

6. Service Quality and Delivery

The SERVQUAL model highlights that reliability (correct and on-time delivery) and responsiveness (quick support, refunds, and updates) are most important. Real-time tracking and accurate delivery time estimates reduce customer frustration.

**TECHNICAL DESCRIPTION**

**Hardware Requirements:**

**Processor:-** Intel Core i3 or higher

**Ram:** Minimum 4 Gb

**Hard-disk :-** Minimum 500 Gb HDD or 128 Gb SSD

**Software Requirements:**

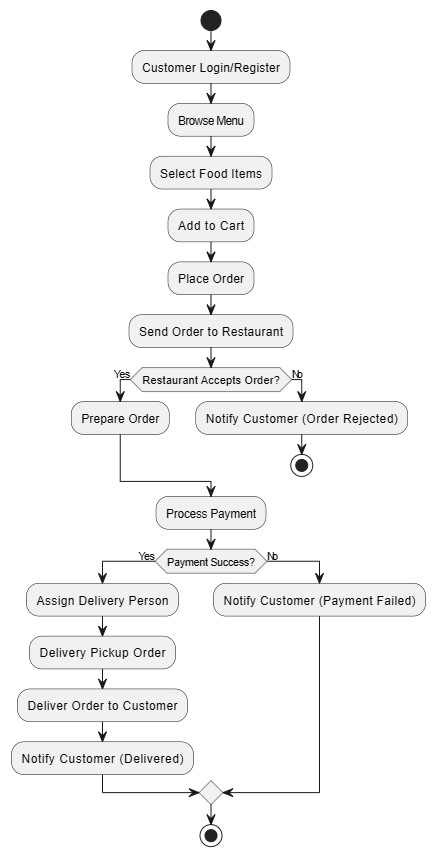
**Operating System:-** windows 10 or later

**Programing Language:** Android Using Java

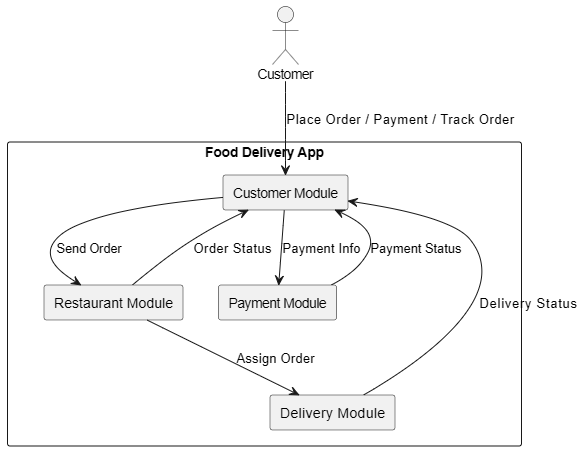
**Database:-** Inbuilt Phone

Development Tools:- Android Studio

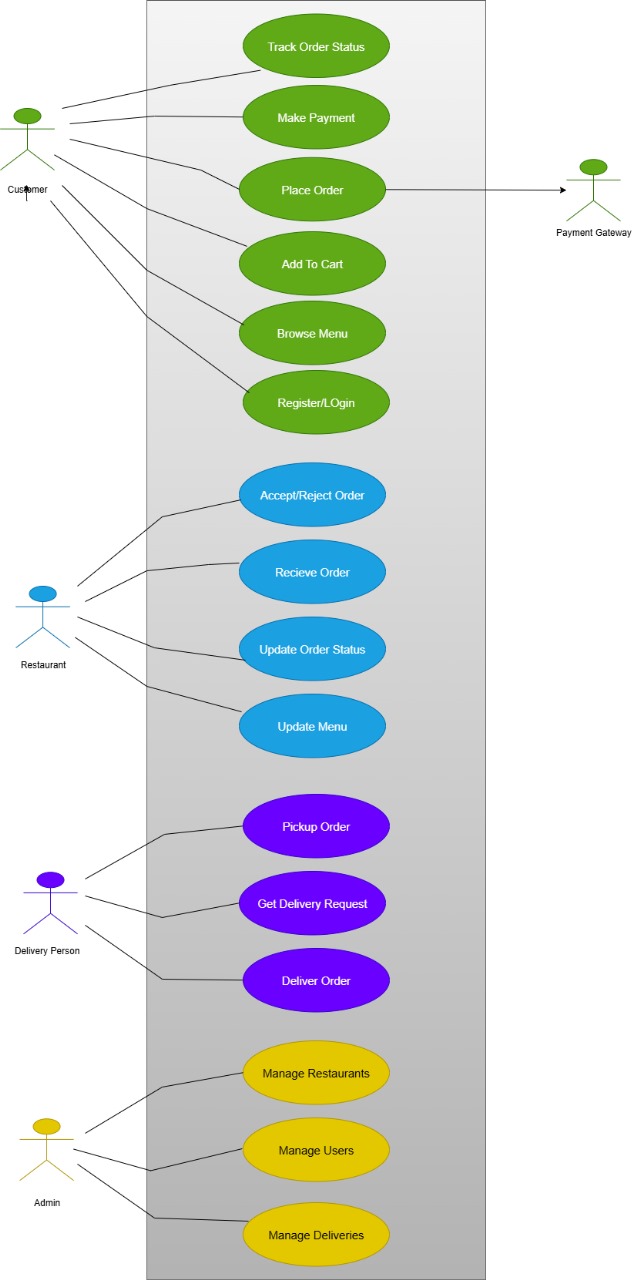
**Flow Chart**



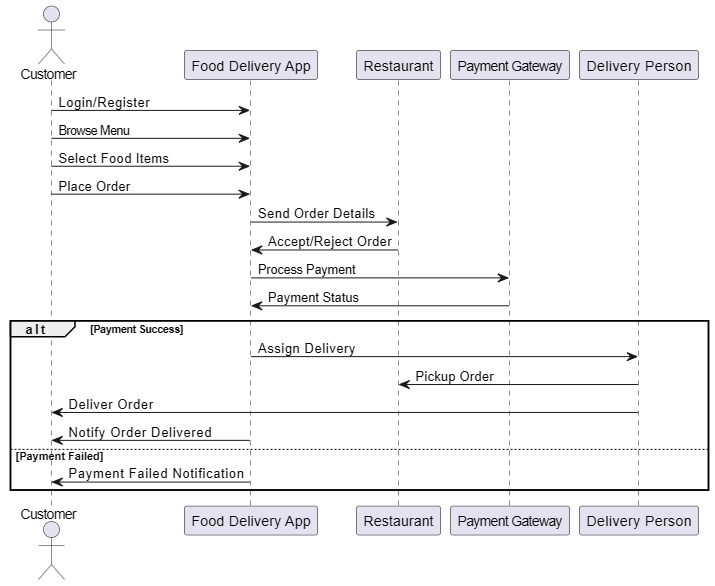
**Data Flow Diagram**



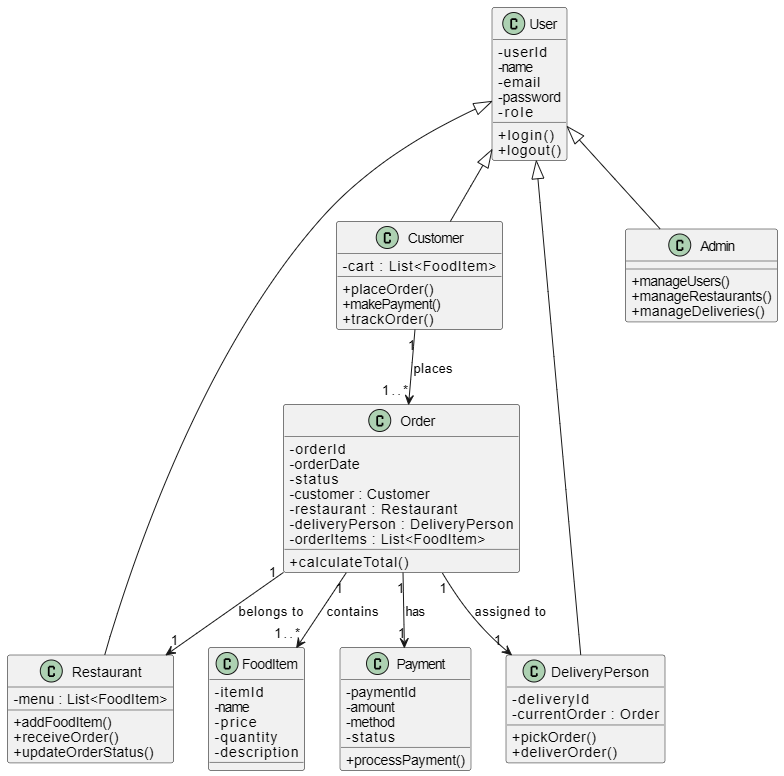
**Use Case Diagram**



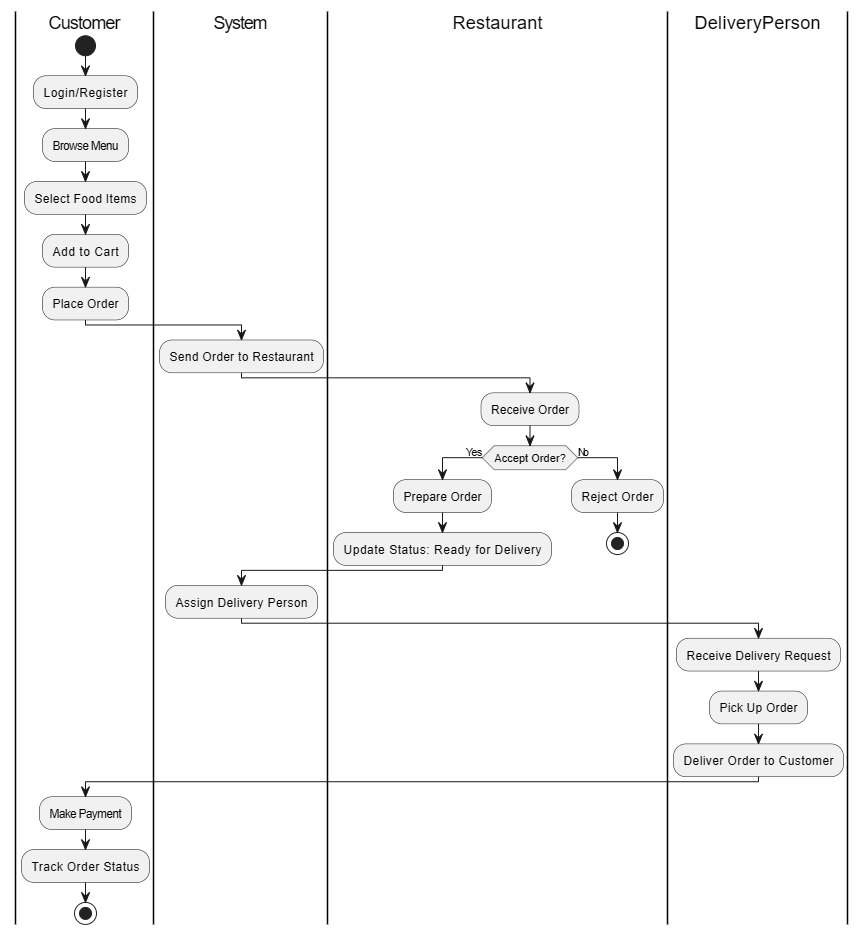
**Sequential Diagram**

****

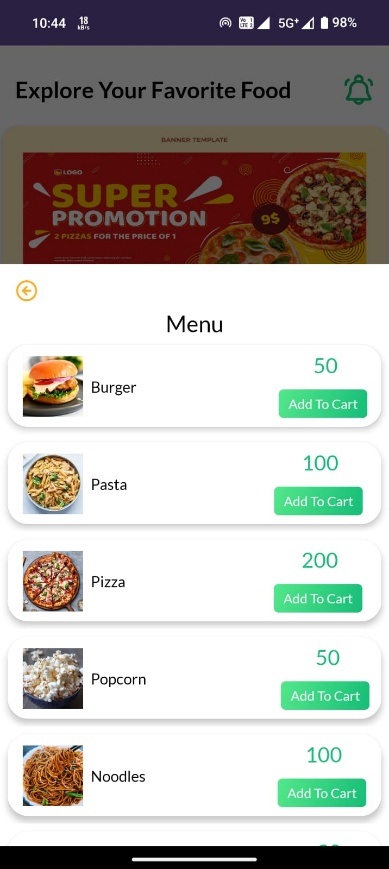
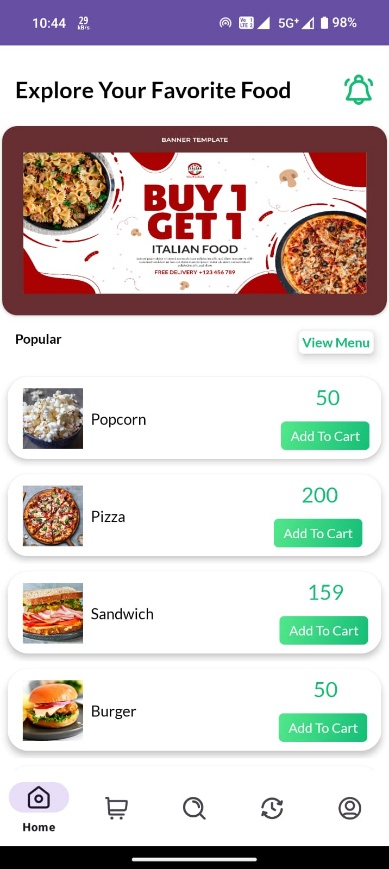
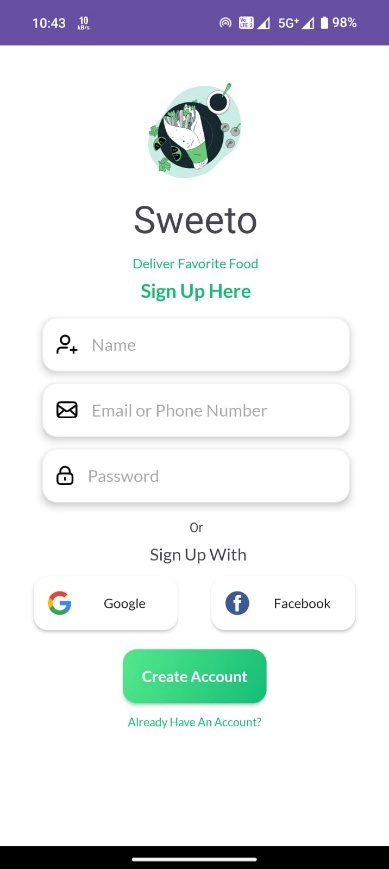
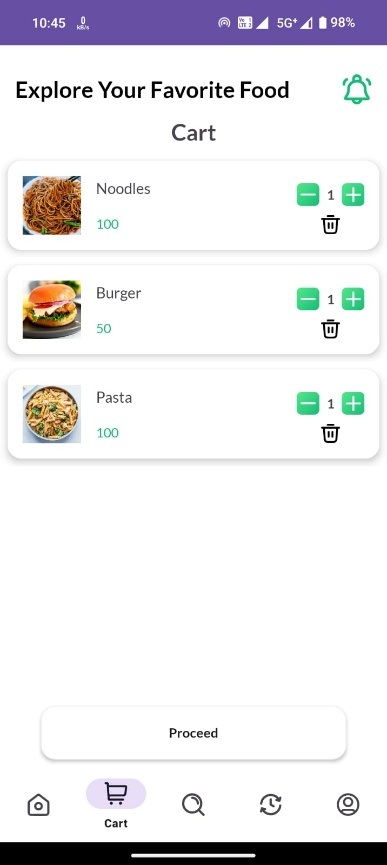
**Class Diagram**

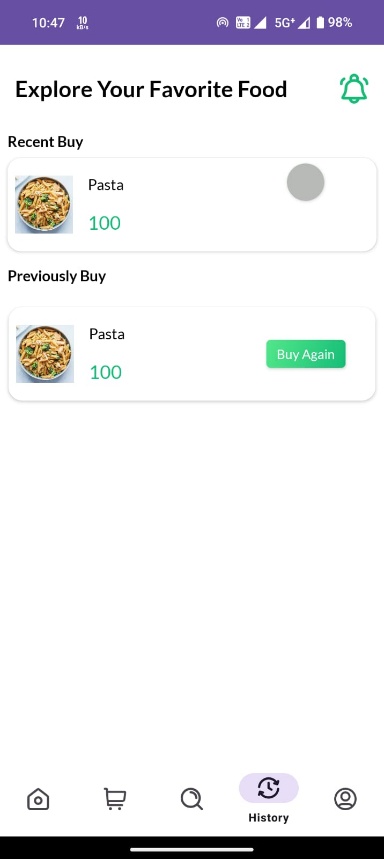
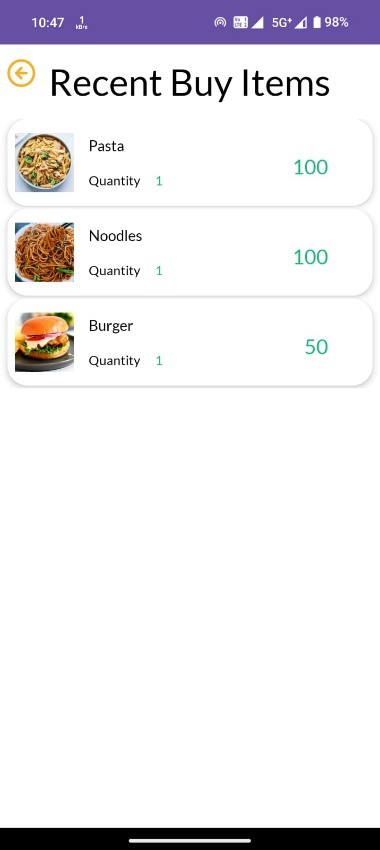
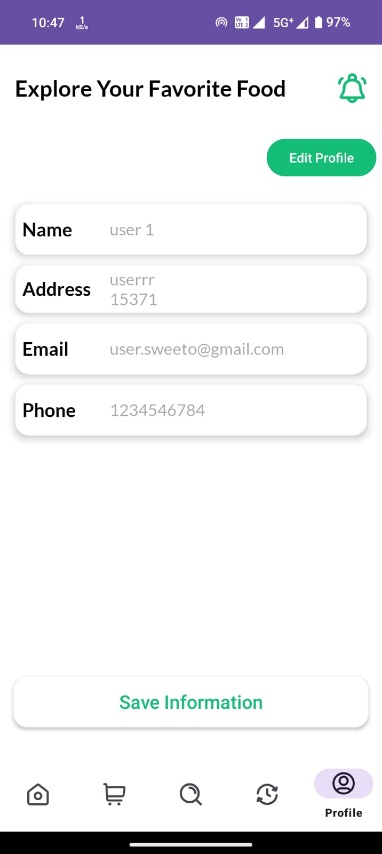


**Activity Diagram**

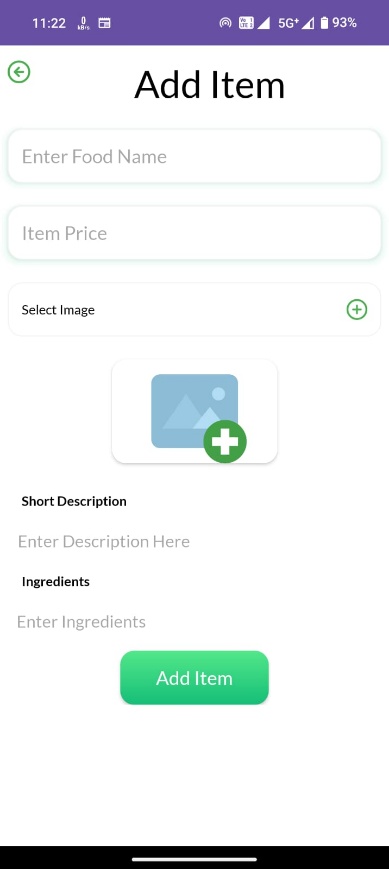
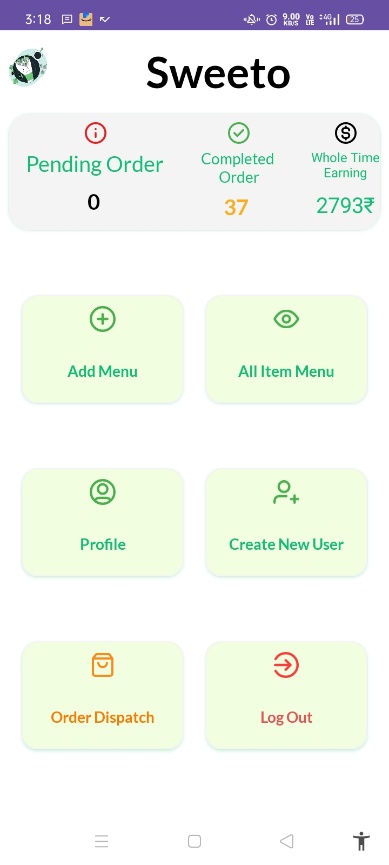
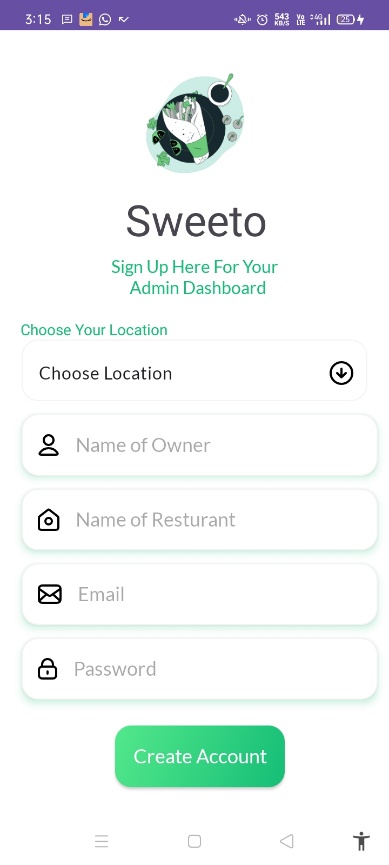
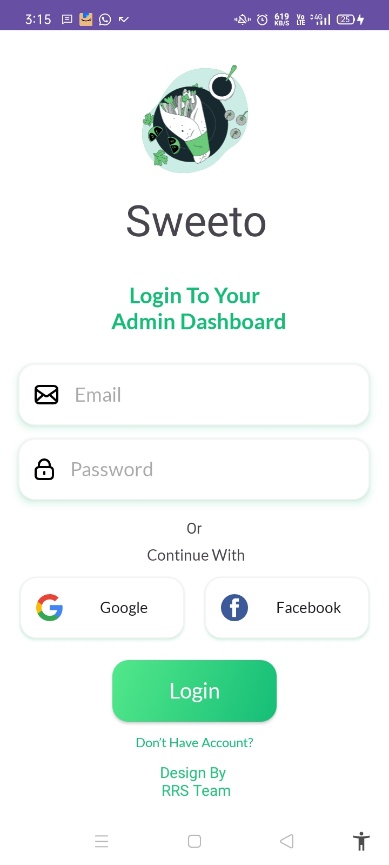


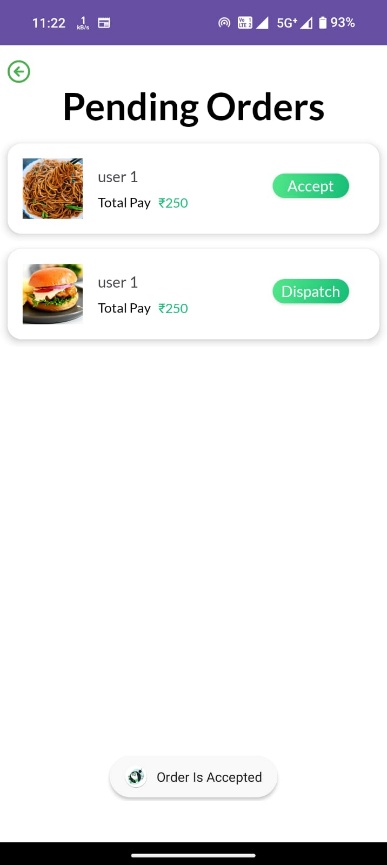
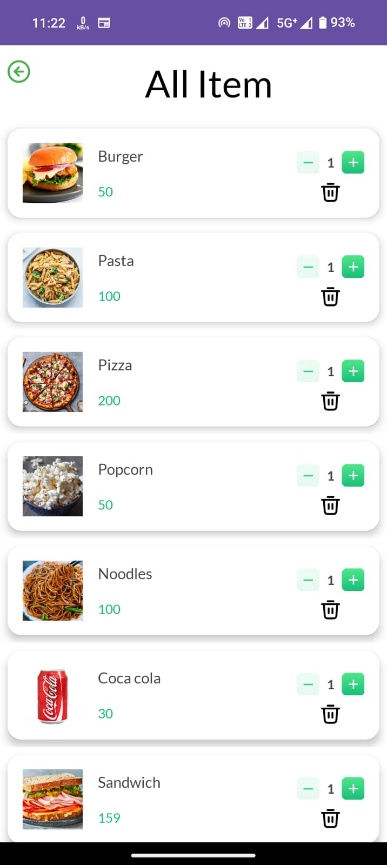
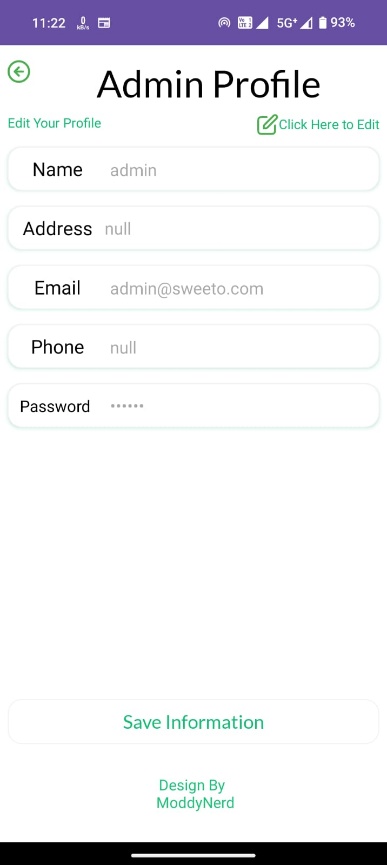
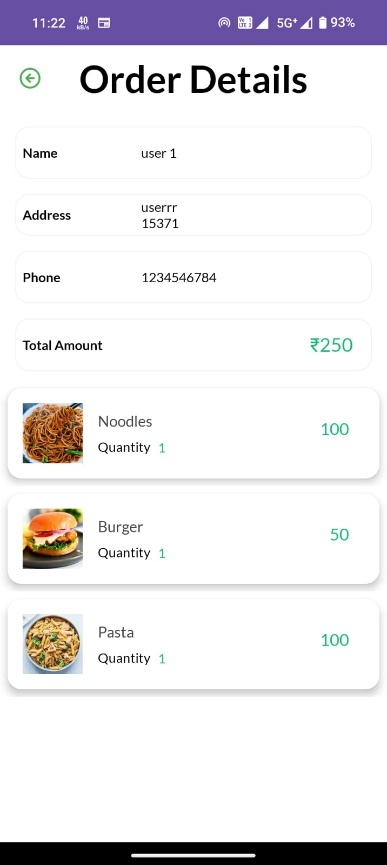
**Design (USER)**





**Design (ADMIN)**





# SYSTEM TESTING

# 1. Functional Testing

# Ensures that each function of the software operates in accordance with the requirement like CRUD operations.

# 2. Integration Testing

# Tests the interaction between different modules like the GUI, database, and backend logic.

# 3. Validation Testing

# Checks whether user inputs are properly validated.

# 4. Performance Testing

# Ensures the system responds quickly and can handle operations efficiently.

# 5. Usability Testing

# Checks whether the application is easy to use for non-technical users.

# 6. Security Testing

# Tests whether data is secure and unauthorized access is prevented.

# CONCLUSION

The Food Delivery App successfully provides a digital platform that connects customers, restaurants, and delivery personnel in a seamless manner. It simplifies the process of food ordering by offering features like **online ordering, secure payment, and real-time delivery tracking**. Restaurants benefit from efficient order management, while customers enjoy convenience and transparency.

This project demonstrates how software engineering concepts can be applied to solve real-world problems by integrating **user-friendly design, database management, and modular functionality**. In conclusion, the Food Delivery App offers an effective, reliable, and scalable solution for modern food ordering and delivery services.

# LEARNING DURING PROJECT WORK

Working on FOOD DELIVERY APP project. We learning like app development skills in learned how to design and build GUI based applications using Android.

Also we learned how to perform CRUD operations through SQL queries.

We enhanced analytical skill while troubleshooting bugs and logic errors.

# BIBLIOGRAPHY

## Online references:

1. [www.youtube.com](http://www.youtube.com/)
2. [www.geeksforgeeks.org](http://www.geeksforgeeks.org/)
3. Many Other Websites

## Offline references:

1. Android Programming For Beginners
2. Android programming : Pushing the limits