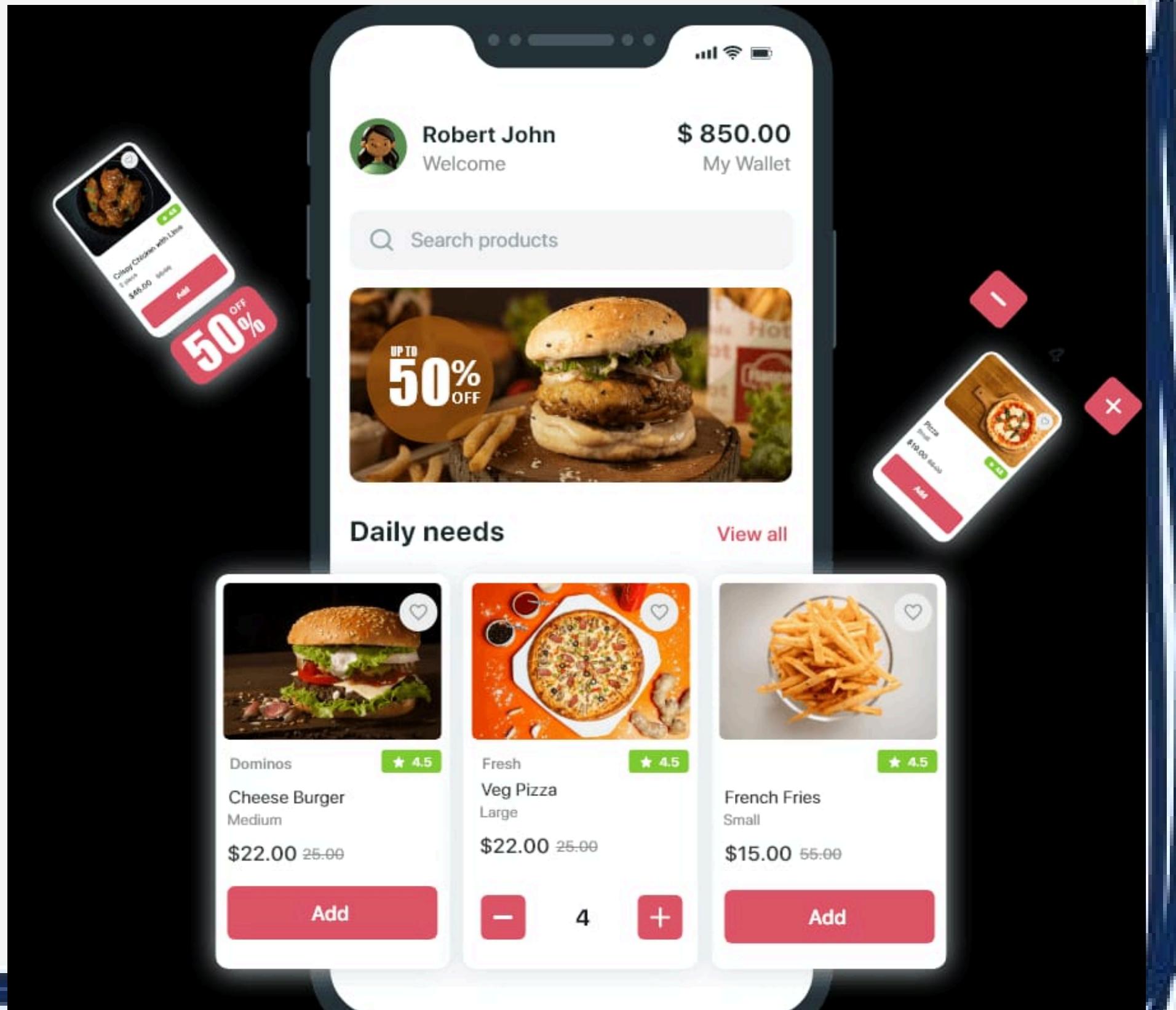
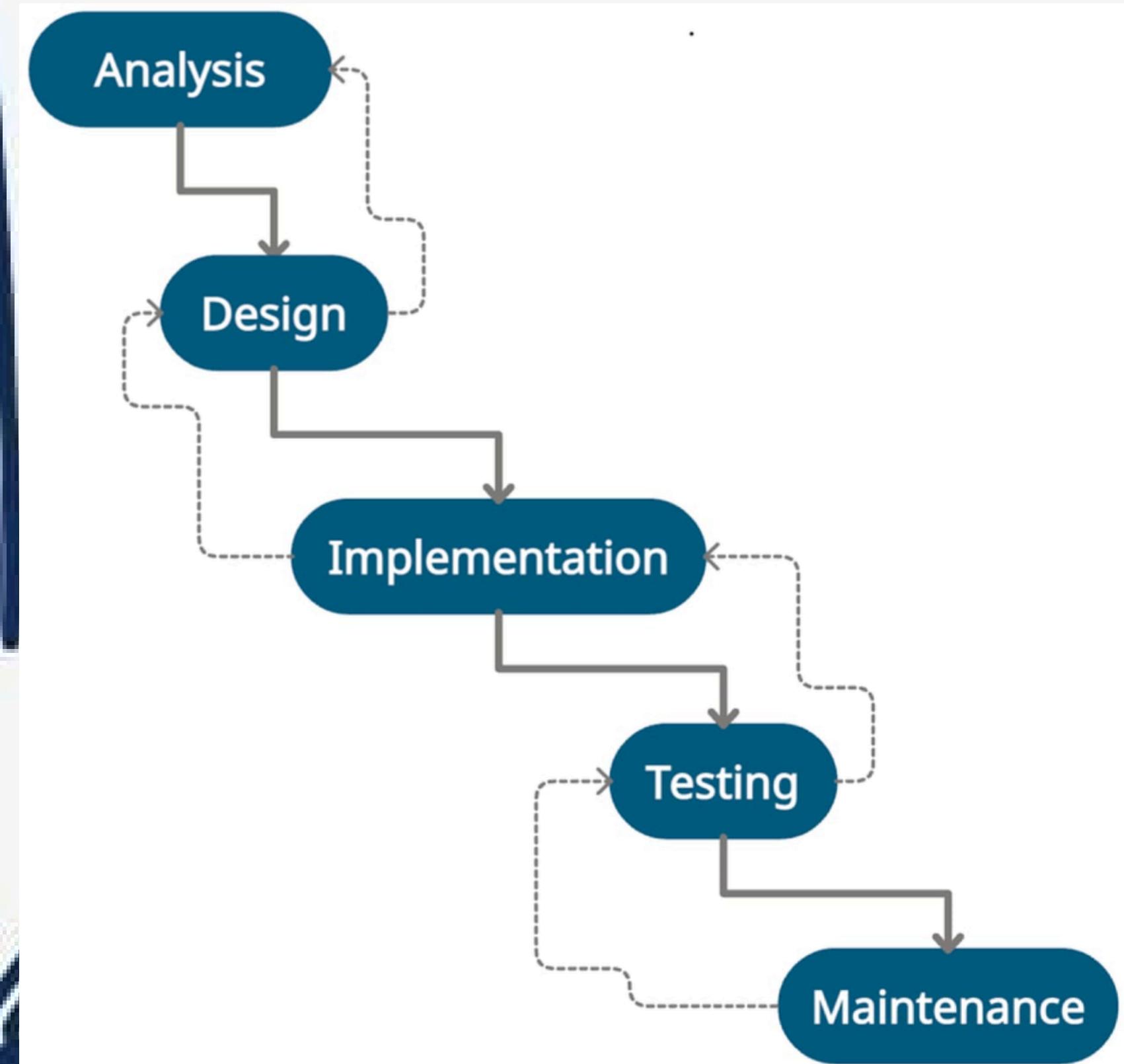


Task 03 :- Waterfall Development Model

TOPIC:- To create a waterfall development model for a new product

TEAM:- Cyber Tribe

Waterfall Model :-



ANALYSIS :-

1. Functional Requirement:-

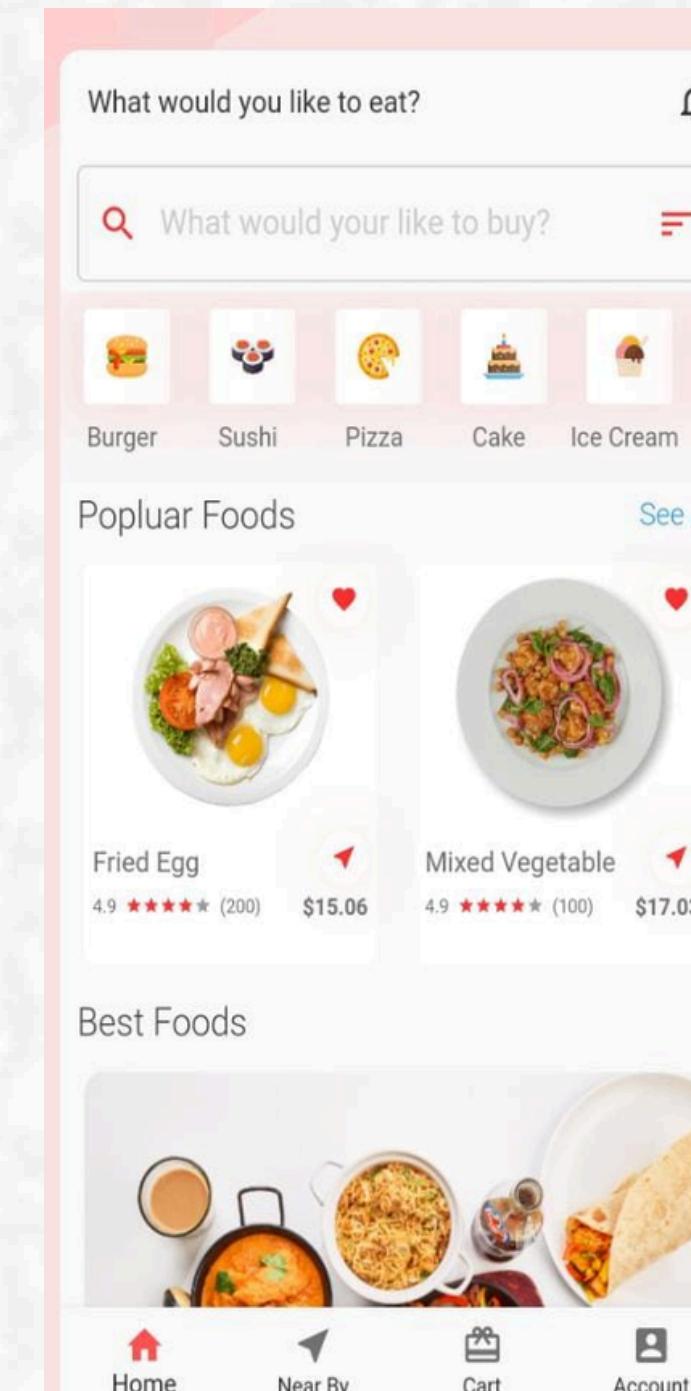
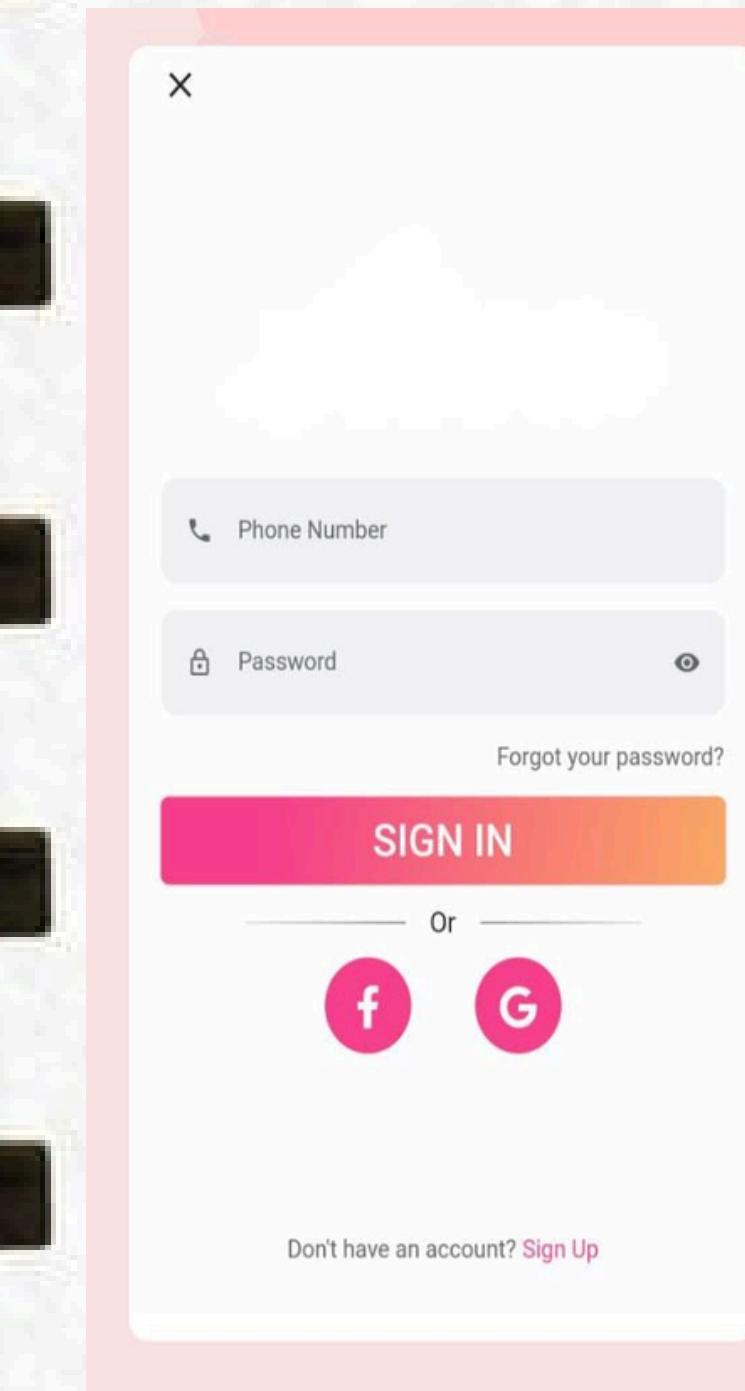
1. User registration and login
2. Restaurant listing and menu browsing
3. Food ordering and gateway integration
4. Order tracking and status updates
5. User profile management

2. Non-Functional requirements:-

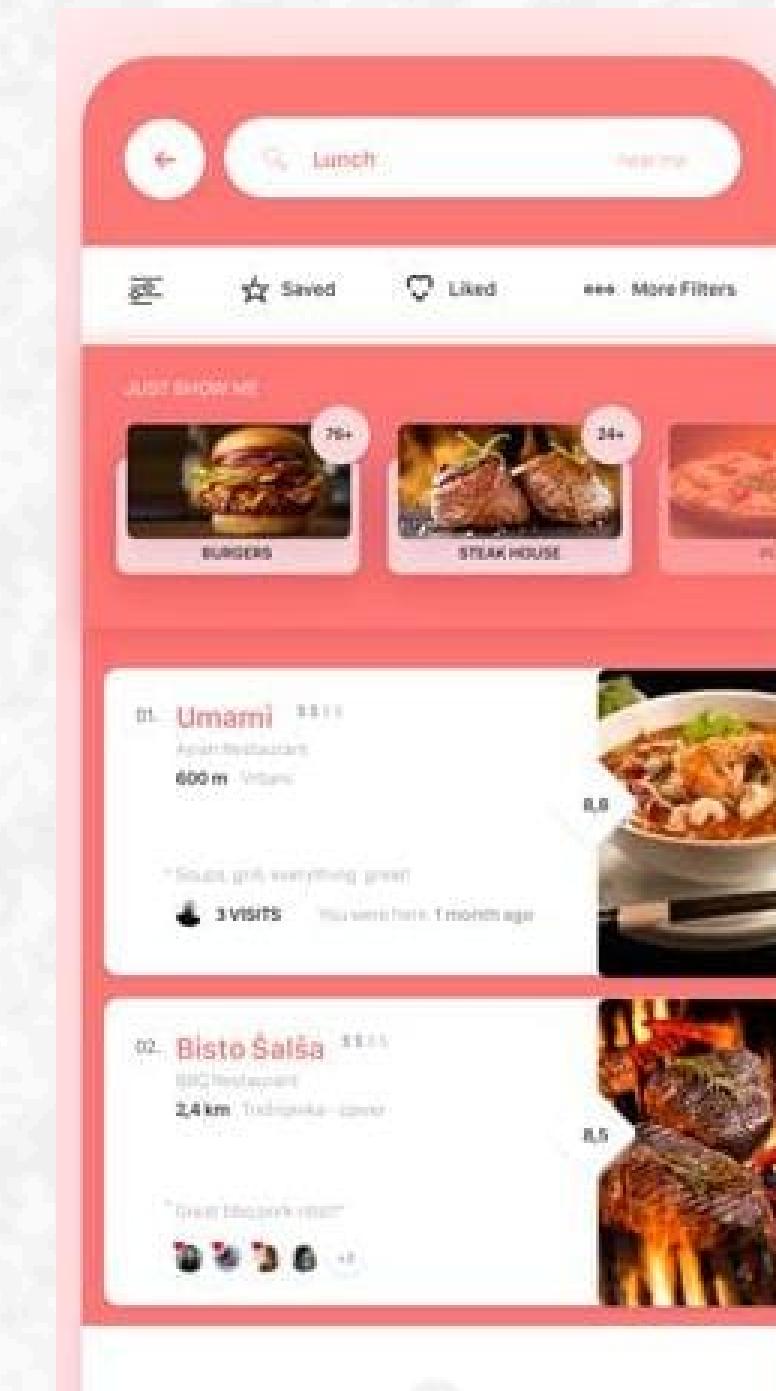
- 1. Performance:-** Fast loading times and responsive UI
- 2. Security:-** Encryption and secure payment processing
- 3. Usability:-** Intuitive navigation and clear instructions
- 4. Scalability:-** Support for increasing user base and restaurant partnerships

DESIGN :-

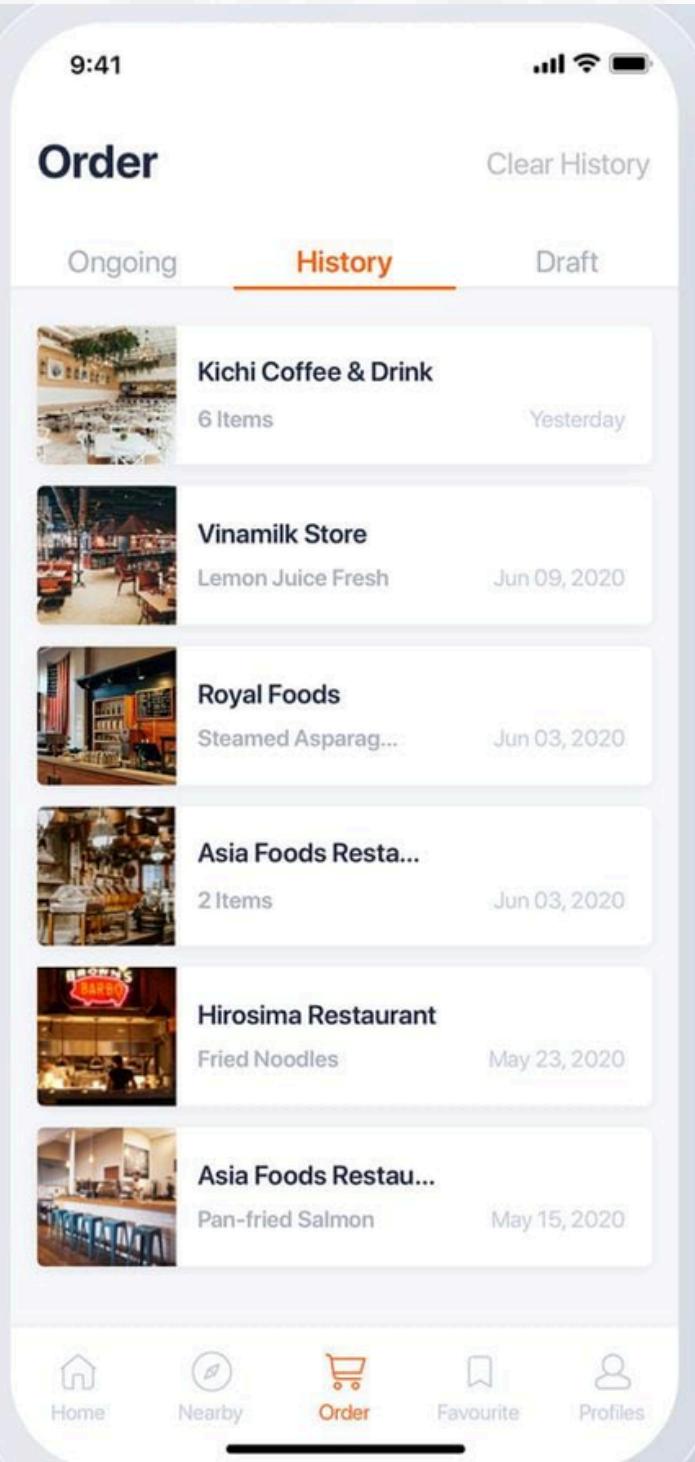
Login screen:-



Menu Page:- Restaurant list:-



Order History



Implementation :-

1. Backend Development (APIs, Server-side Logic)

Tools/Technologies:

- Node.js: It is used to handle server-side logic.
- Express.js: It is used for the framework for building APIs.
- MongoDB: NoSQL database for storing app data

Key Tasks:

- Set up a Node.js server.
- Create RESTful APIs for user authentication, data management, and integration of third-party services.
- Ensure database design aligns with app needs.
- Third-Party Services:
 - **Stripe integration for payment processing.**
 - **Google Maps API for location services.**

2. Frontend Development (Mobile App UI/UX)

Tools/Technologies:

- React Native: Cross-platform mobile app development.
- Platforms: iOS and Android.

Key Tasks:

- Design and implement the UI/UX for the mobile app.
- Ensure seamless user experience and responsive design.
- Set up navigation and interactions between screens.
- Integrate APIs developed in the backend.

3. Third-Party Service Integration

Stripe Payment Gateway:

- Enable secure and seamless payments.
- Integrate subscription models and one-time payment options.

Google Maps API:

- Enable location-based services, such as displaying user locations and nearby services.

4. Testing Phase

Unit Testing:

- Test individual components of both the backend and frontend.
- Mock external services (e.g., payment gateways, location services) for testing.
-

Integration Testing:

- Test the integration between backend and frontend.
- Test integration with third-party services like Stripe and Google Maps.

Testing:-

- Conduct unit testing for individual backend components.
- Conduct unit testing on individual components.
- Ensure app functions properly across both iOS and Android platforms.

MAINTENANCE :-

- Monitor app performance and analytics
- Fix defects and issues
- Implement new features and updates
- Conduct regular security audits

THANK

YOU!