SOFTWARE DESIGN DOCUMENT

for

FOODKAMP APP (P005)

Version 1.0

Prepared by : 1. Devansh Sahu (190001012)
2. Mayank Raj (190001035)
3. Nimish Bansal (190001040)

Submitted to : Dr. Puneet Gupta
Course Instructor
(CS258)

May 6, 2021

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to describe the implementation of the ‘Foodkamp’ app. This document explicitly talks about the design, processes and working of the app. This document contains design which satisfies all the requirements described in the Software Requirements and Specification (SRS) document. ‘Foodkamp’ app will act as a mediator between the user and restaurant and enhance the quality of interaction and increase the efficiency of both the user and the restaurant.

1.2 Scope

This document describes the implementation details of the ‘Foodkamp’ app. The app will consist of two major interafaces. First, the user side, and second, the restaurant side. This document will only give information about the design, processes and working of the app. The coding and testing part will not be talked about in this document.

1.3 Definitions, Acronyms, Abbreviations

1.3.1 User

User User is the customer who will make food orders, which will be sent to the restaurant via ’FoodKamp’ app.

Cart Cart is the list of items that the user wants to order.

Order An order will contain the list of items, quantity and total amount the user has to pay.

1.3.2 Restaurant

Menu Menu is the list of items that are available to order at the chosen restaurant.

Order Verification It is the process by which restaurant knows that the person who came to take the order is the same person who ordered the food. Also, this allows the user to be assured that no one else can take their food order.

2 Design Overview

2.1 Description of Problem

Food ordering is one of the most difficult tasks when it comes to crowdy places. You never know how much time it would take for you to even get your order confirmed. Physically being present at the restaurant and doing all the tasks manually is a very time consuming and tiring process.

Also, because of the new norms, where people are trying to avoid physical contact and moving towards virtual methods also brings a need for a service which will digitize the user and restaurant interaction. This will not only save time but also physical and mental energy. Allowing the user to order food as per their convenience will not only attract more number of users, but also decrease the cost of maintenance of restaurants. Food pick-ups, deliveries and even dine-ins will be fast and on point.

2.2 Technologies Used

The app completely works on software components and needs no external hardware other than the usual devices such as phones, tablets, etc. The app will be developed using Dart, Flutter and Firebase. The app will work on android devices and an IOS version of the app will also be released in near future.

2.3 System Architecture

Figure 1 depicts the high-level system architecture. The system will be constructed from multiple distinct components:

- **User Interface** — This interface will contain all the information about nearby restaurants and their availability and will allow user to make orders to their preferred restaurants. Also, it will contain information about the user and their order history.
- **Restaurant Interface** — This interface will contain information of the orders received to the respective restaurant. Also, the restaurant can make changes to their menu, accept orders and verify them at time of delivery.
- **Data Model** — The classes such as cart, order, item will be required to perform user operations. Also the same classes will be used by the restaurant system for verifying the orders.

- **Data Storage** — The interface for storing, importing and exporting the data model and raw collected data.

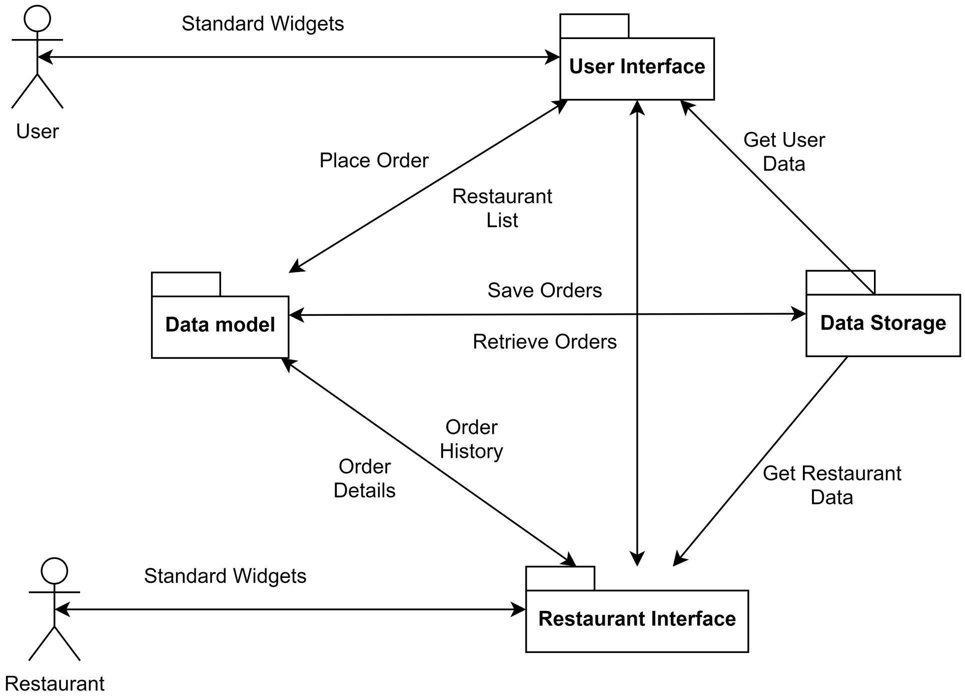

Figure 2.1: FoodKamp Architecture

2.4 System Operation

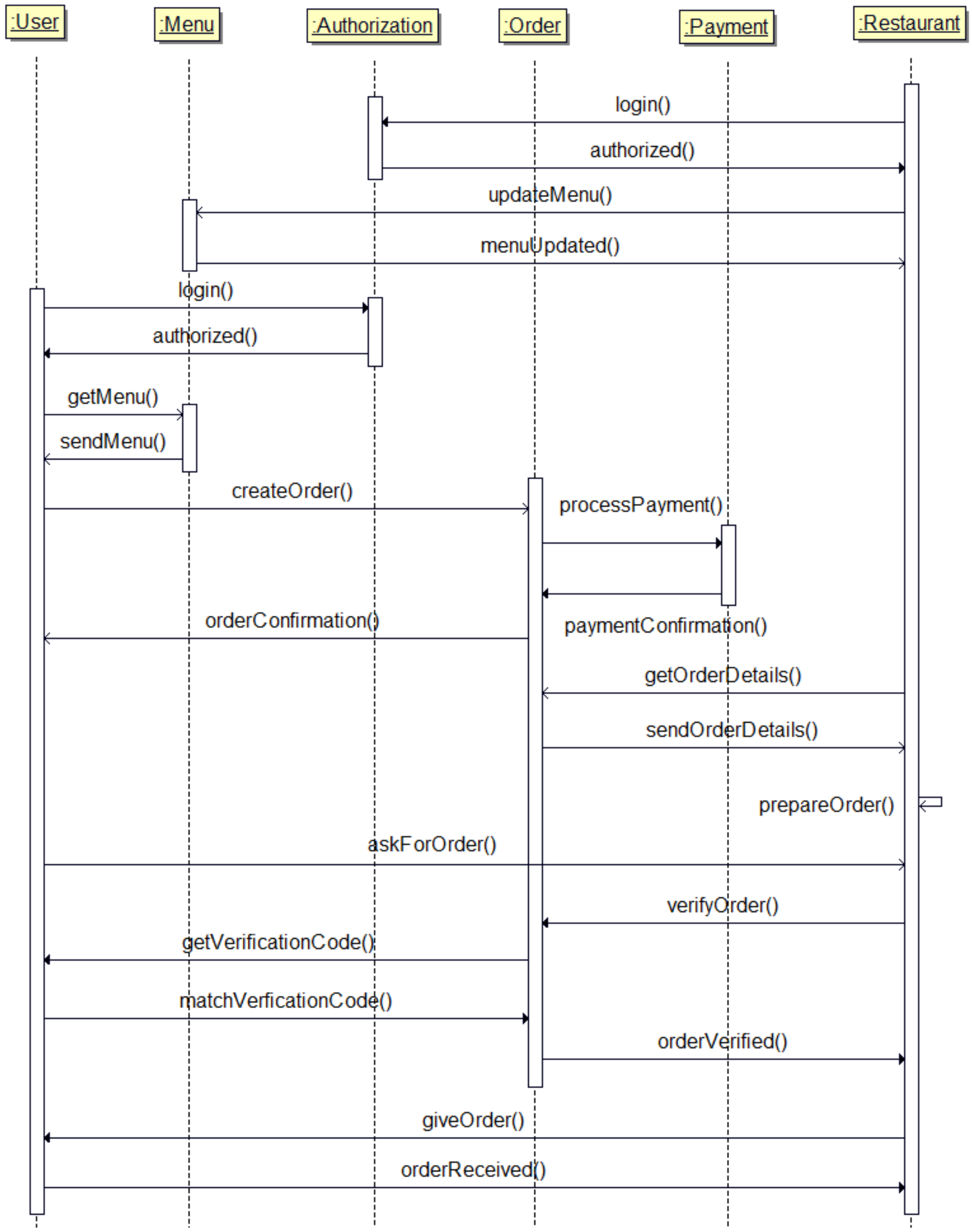

Figure 2.2: Sequence Diagram

3 Classes and Objects Structure

The next figure depicts the class diagram of the ”FoodKamp” app. There are two main classes, user and restaurant. Other classes are generated with the data model and the information is displayed on the user and restaurant interface.

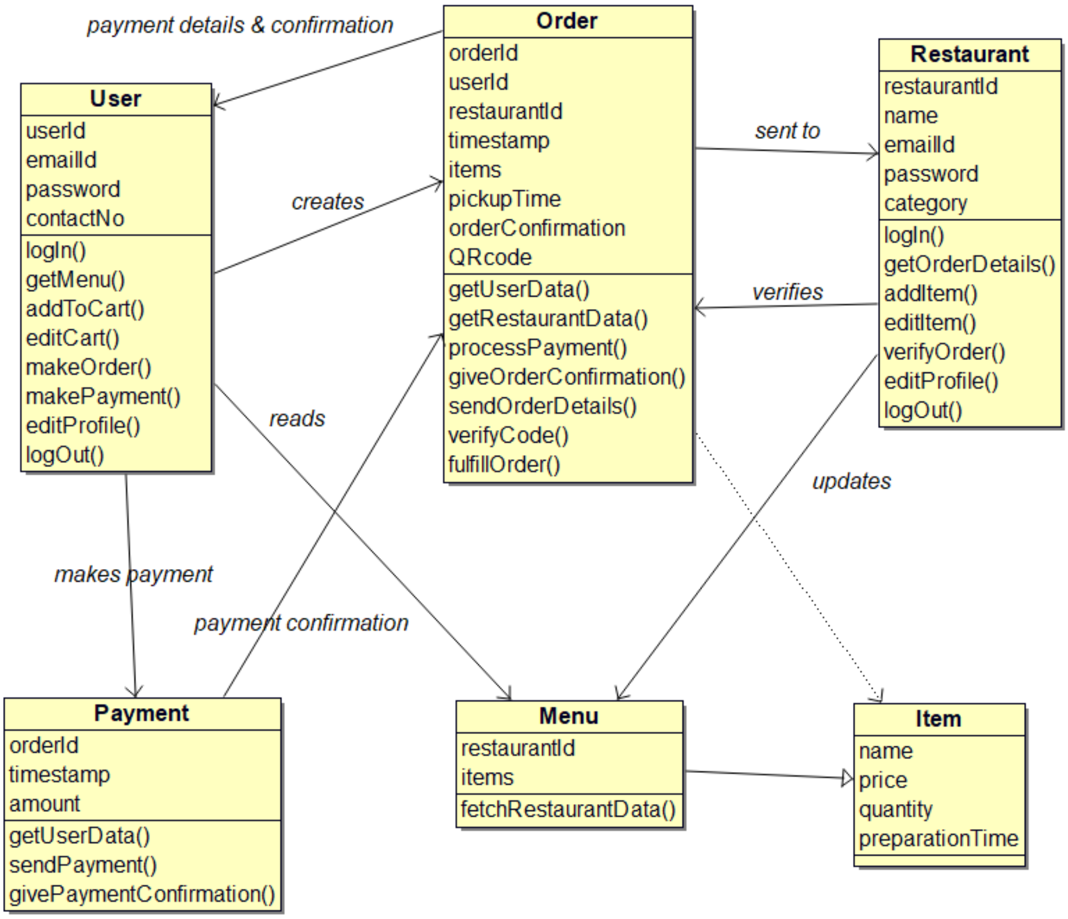

Figure 3.1: Class Diagram

4 Use-Case Diagram

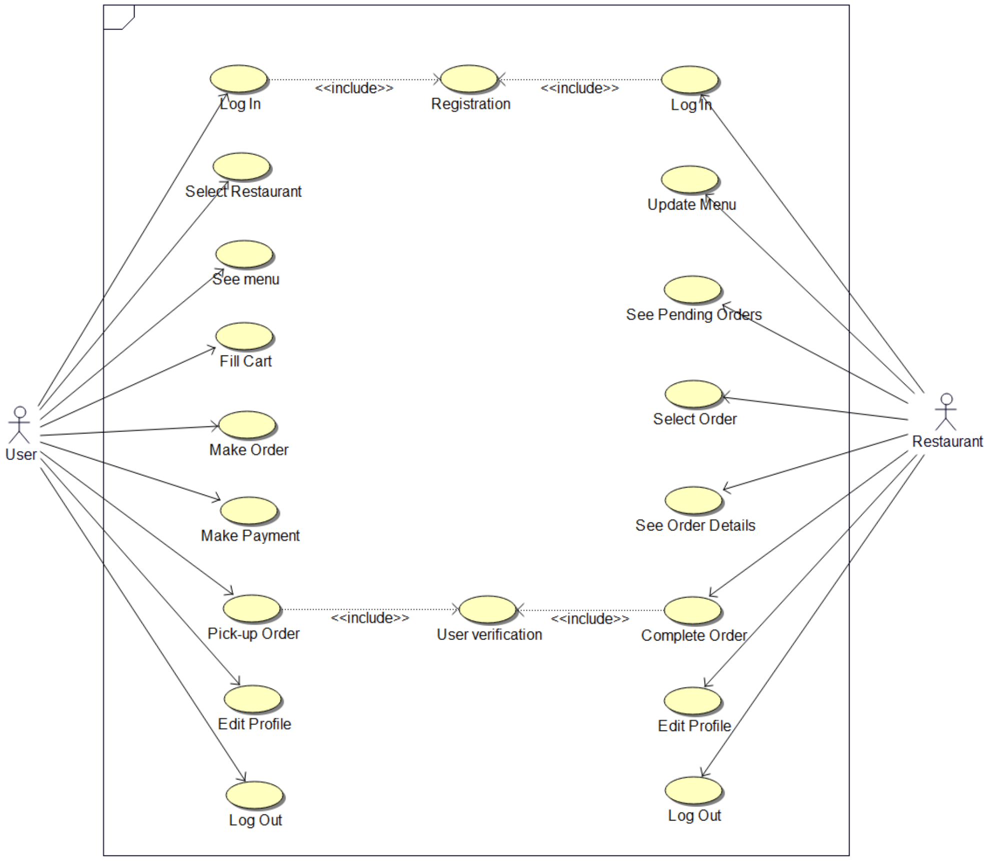

Figure 4.1: Use-case Diagram

5 User Interface

5.1 Login and Sign Up

Here we have the interface and the working of the Login and Sign Up feature

5.1.1 UI

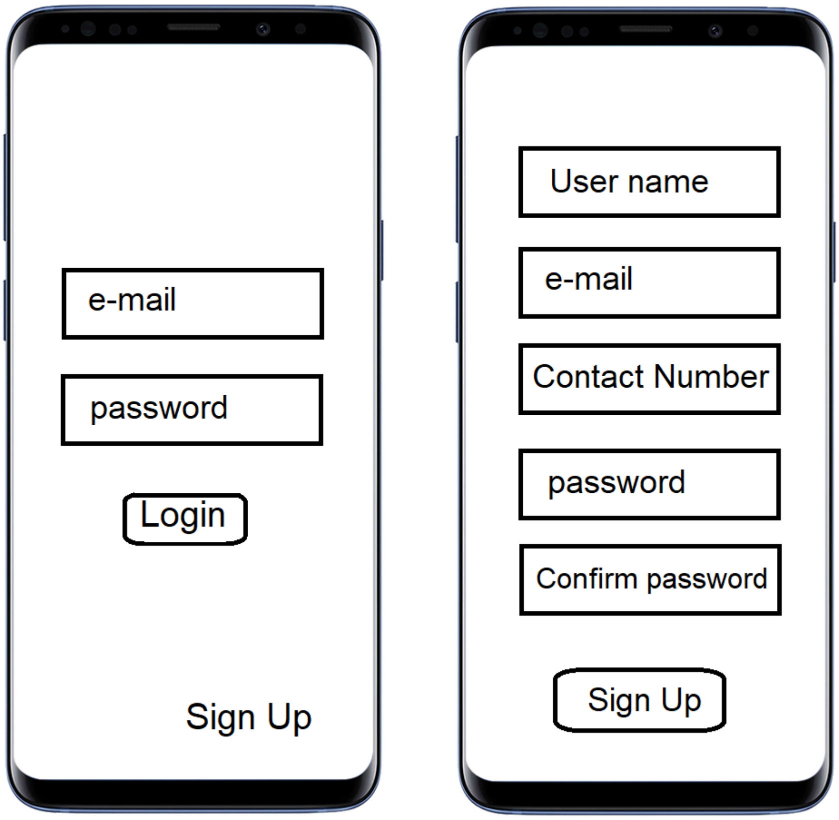

Figure 5.1: Login and Sign Up UI

5.1.2 Activity Diagram

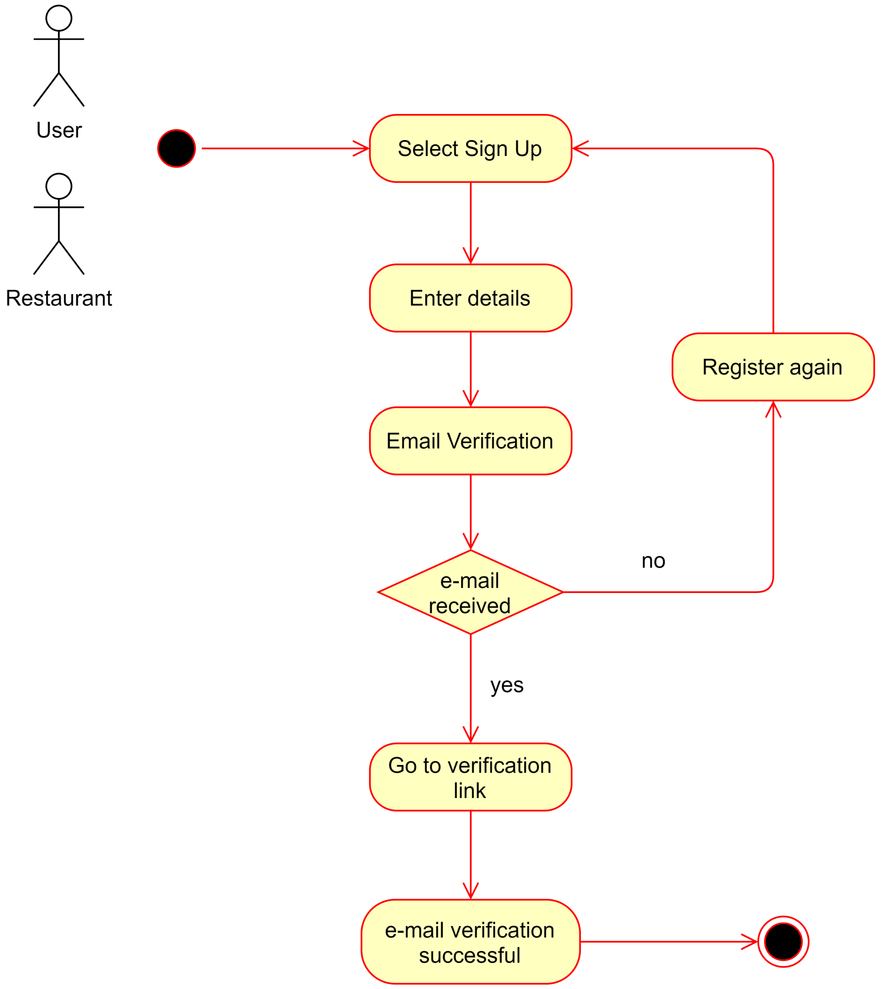

Figure 5.2: Sign Up Process

5.2 Profile and Wallet

Here we have the interface for the profile section, process of editing profile and adding money to wallet

5.2.1 UI

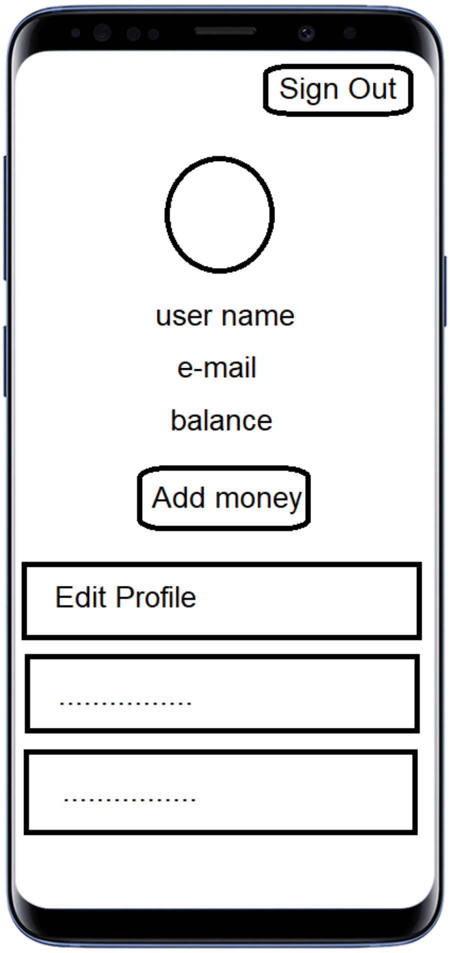

Figure 5.3: Profile Page

5.2.2 Activity Diagram

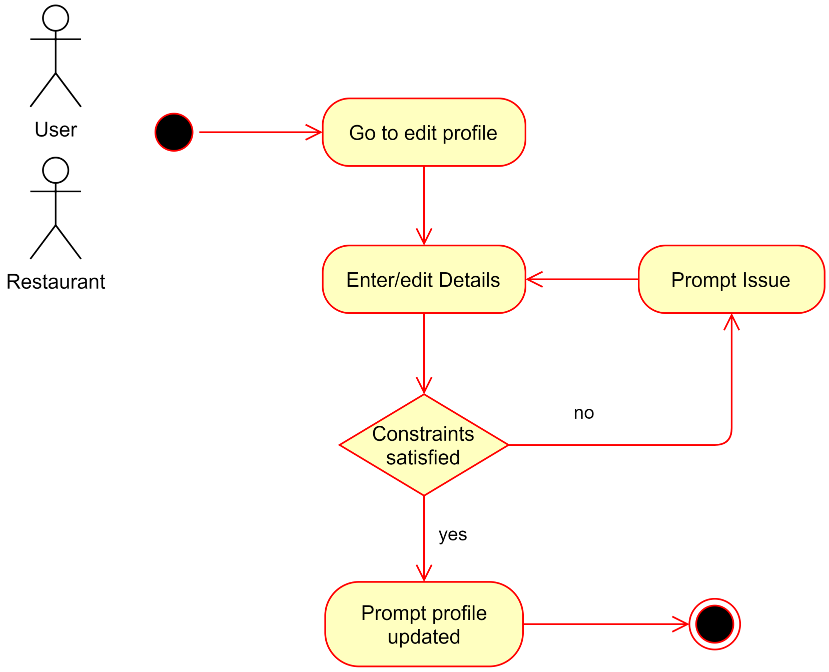

Figure 5.4: Edit Profile Process

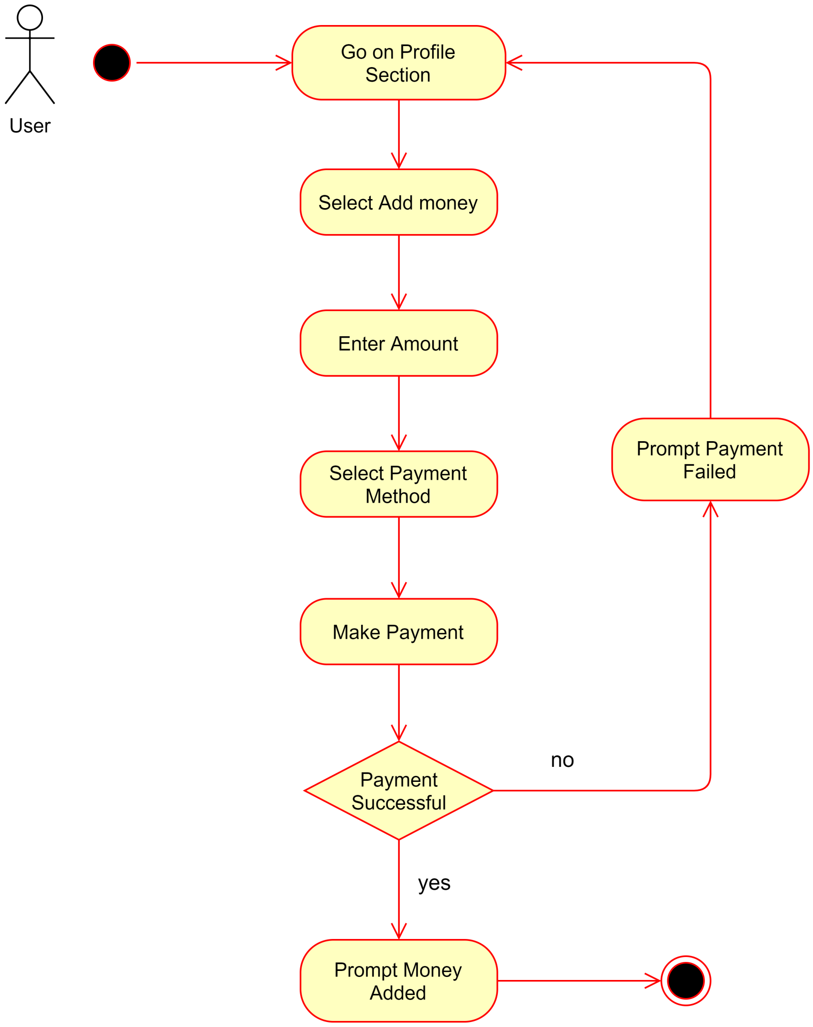

Figure 5.5: Adding Money to Wallet

5.3 Ordering Food

Here we have the interface for food order pages. Also, the process of ordering food is depicted with the help of an activity diagram

5.3.1 UI

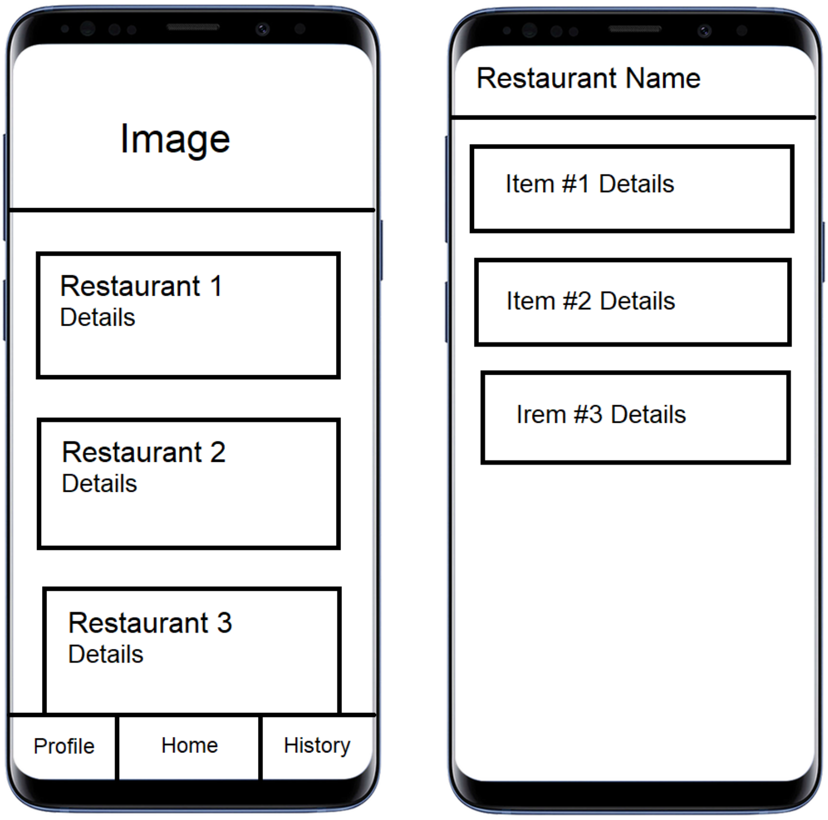

Figure 5.6: Restaurant List and Restaurant Menu

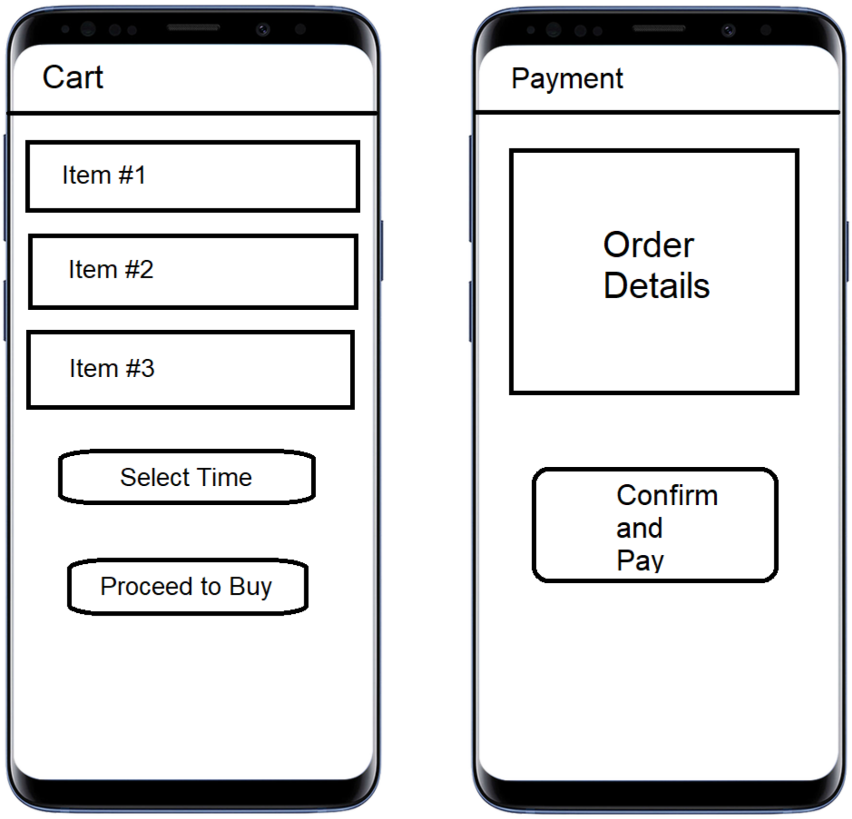

Figure 5.7: Cart and Payment Page

5.3.2 Activity Diagram

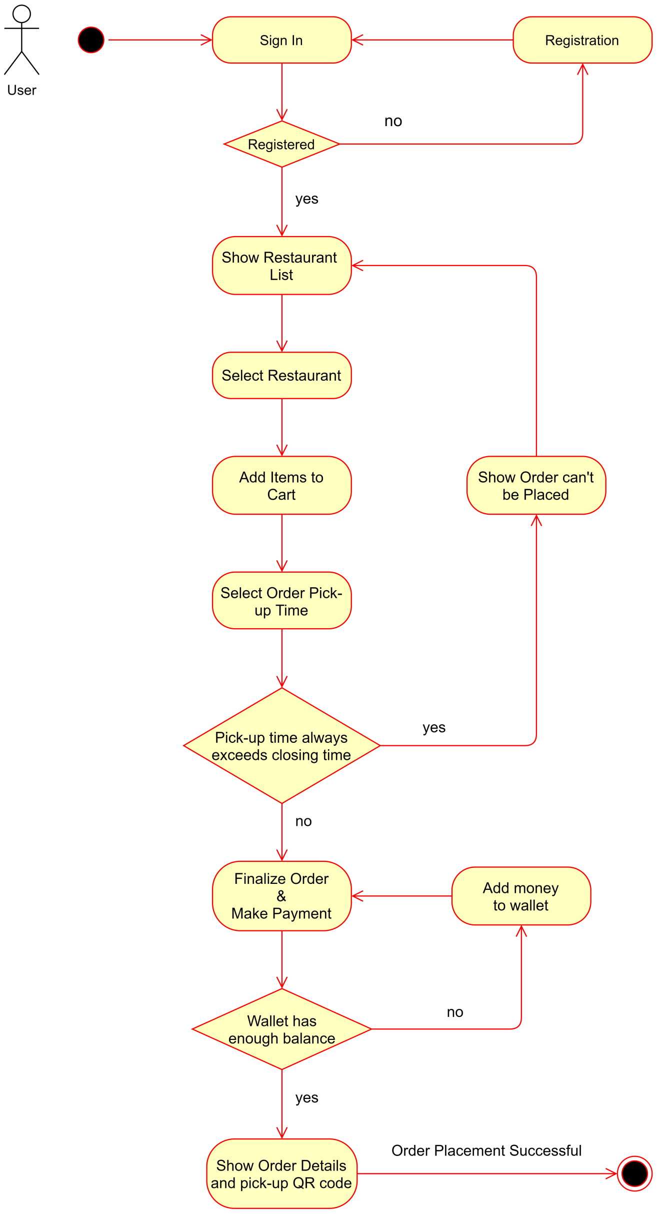
User
Sign In
Registration
no
Registered
yes
Show Restaurant List
Select Restaurant
Add Items to Cart
Show Order can't be Placed
Select Order Pick-up Time
yes
Pick-up time always exceeds closing time
no
Finalize Order & Make Payment
Add money to wallet
no
Wallet has enough balance
yes
Order Placement Successful
Show Order Details and pick-up QR code

Figure 5.8: Food Order Process

5.4 Order History

5.4.1 UI

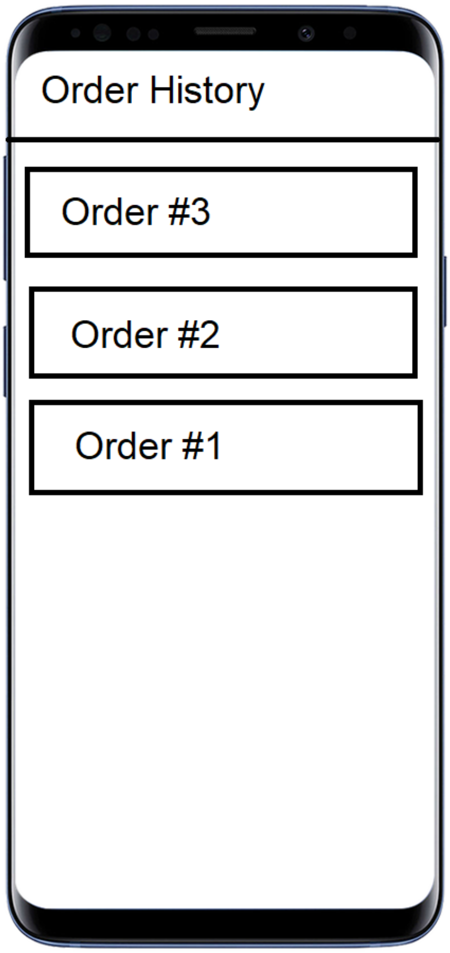

Figure 5.9: Order History Section

6 Restaurant Interface

6.1 Login and Sign Up

Here we have the interface and the working of the Login and Sign Up feature for Restaurant

6.1.1 UI

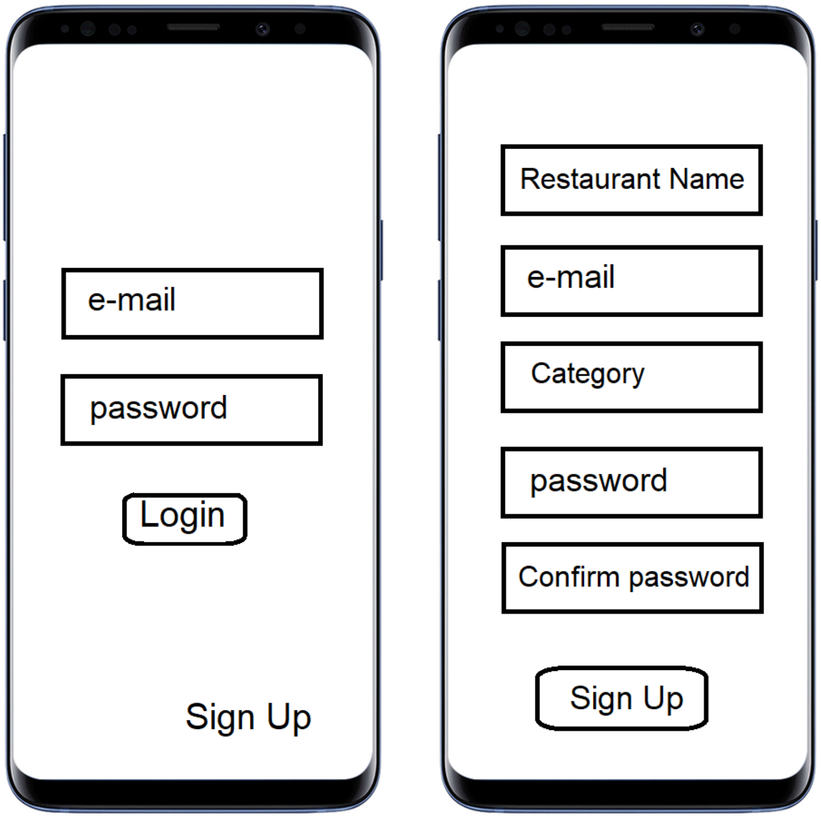

Figure 6.1: Login and Sign Up UI

6.1.2 Activity Diagram

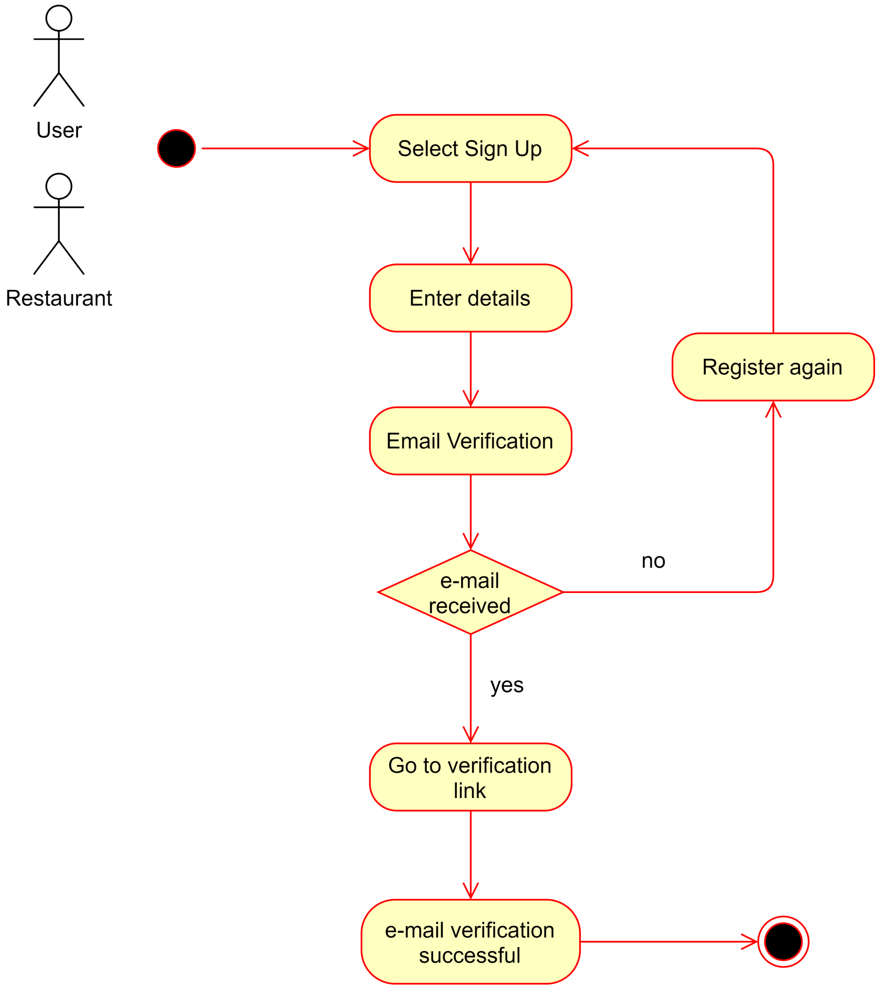

Figure 6.2: Sign Up Process

6.2 Menu

Here we have the interface of the menu as seen by the restaurant. Restaurant can add or delete items and edit items.

6.2.1 UI

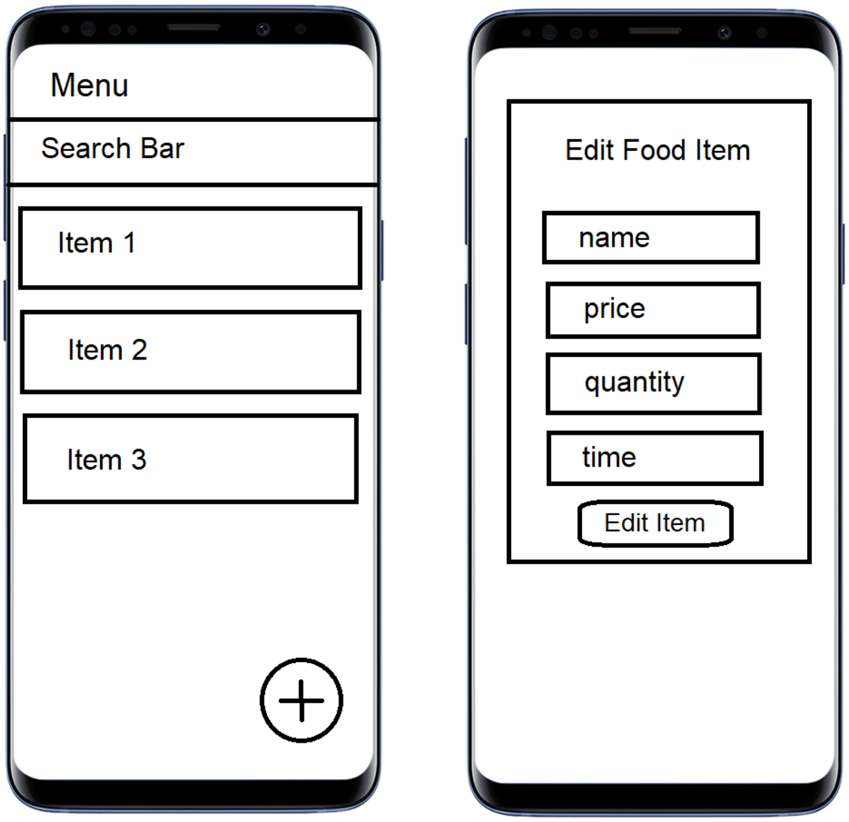

Figure 6.3: Menu and Edit Item Section

6.2.2 Activity Diagram

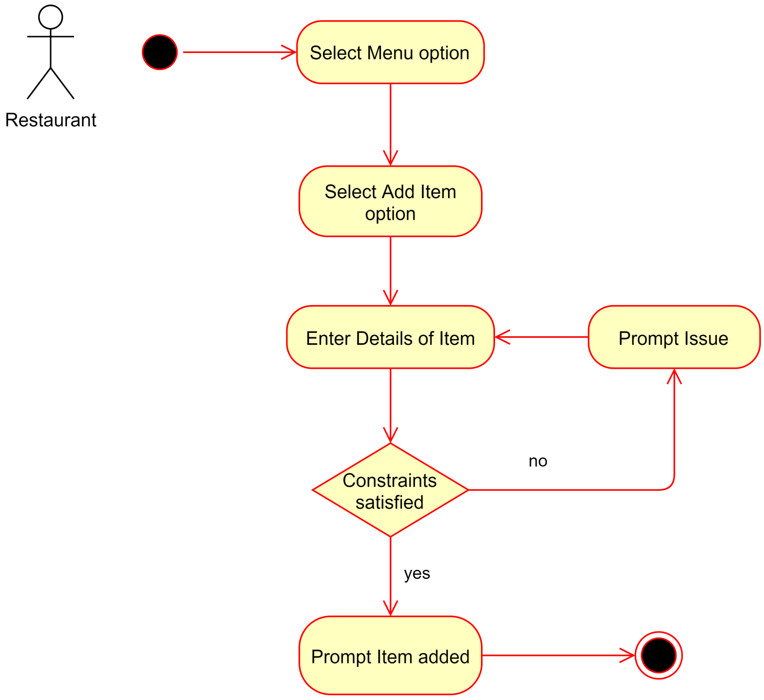

Figure 6.4: Adding an item to the menu

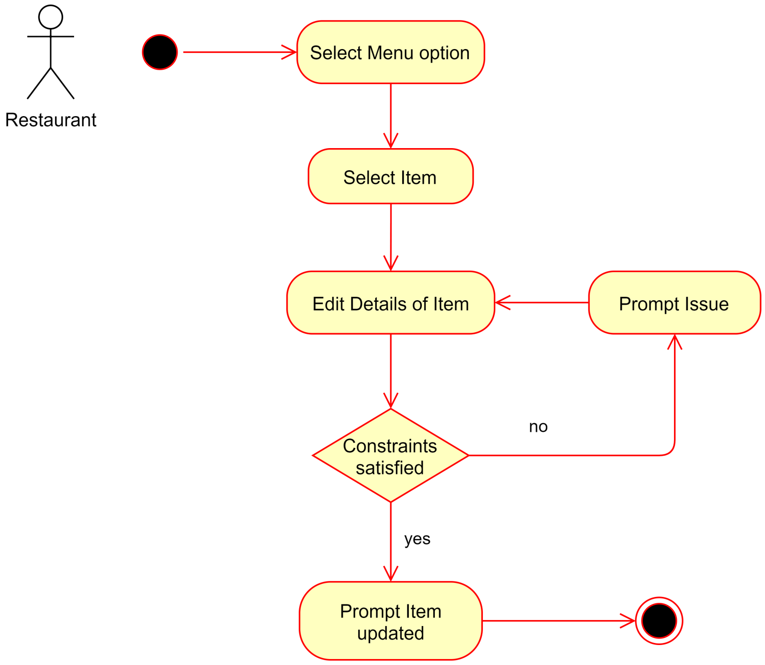

Figure 6.5: Updating an item from the menu

6.3 Completing Order

Here we have the interface for pending orders for restaurant. Restaurant can select any order and view the order details. Also, we have the process followed by restaurant for fulfilling any order.

6.3.1 UI

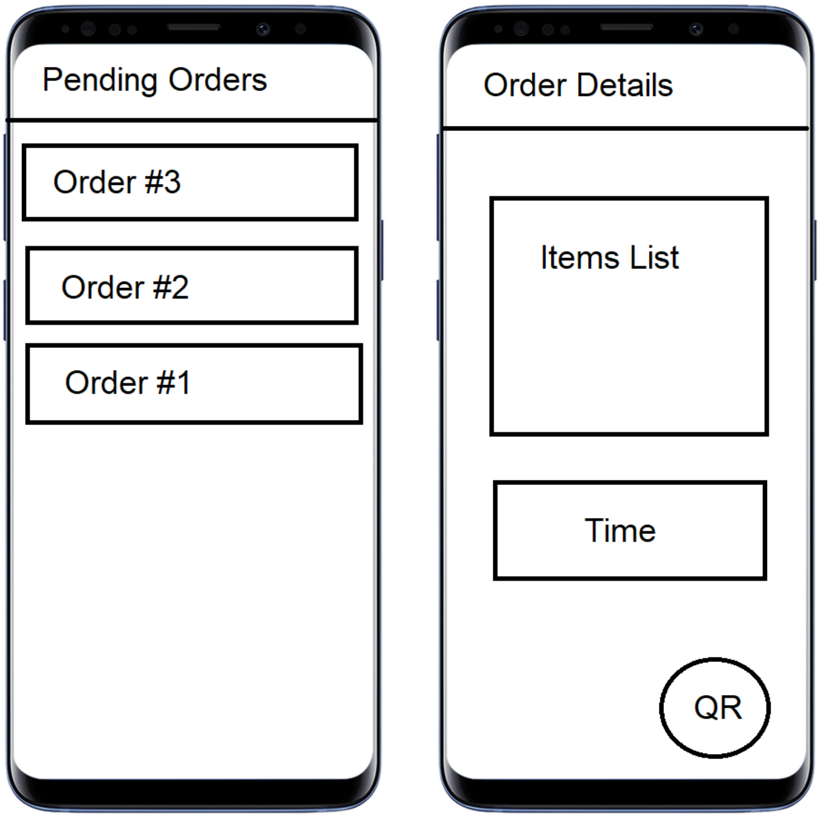

Figure 6.6: Pending Order and Order Details Section

6.3.2 Activity Diagram

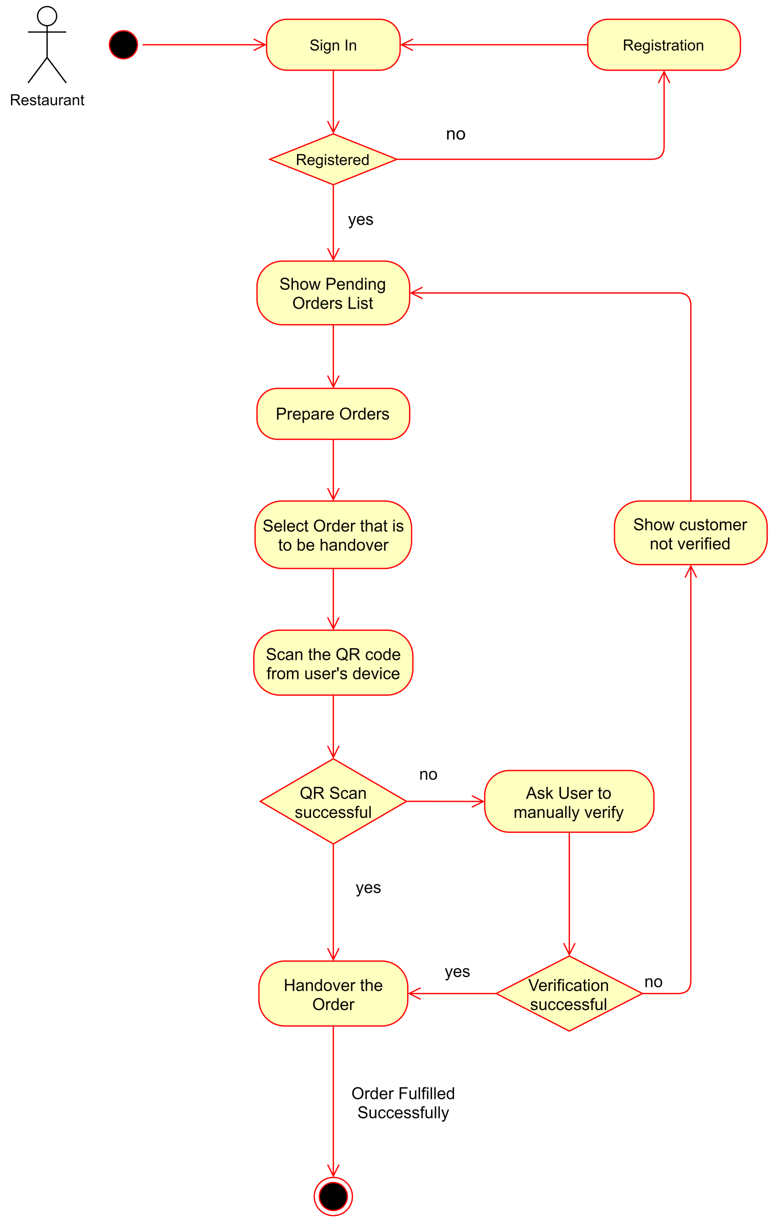
Restaurant
Sign In
Registration
Registered
no
yes
Show Pending Orders List
Prepare Orders
Select Order that is to be handover
Show customer not verified
Scan the QR code from user's device
QR Scan successful
no
Ask User to manually verify
yes
Handover the Order
yes
Verification successful
no
Order Fulfilled Successfully

Figure 6.7: Process of completing the order by restaurant

6.4 Order History

6.4.1 UI

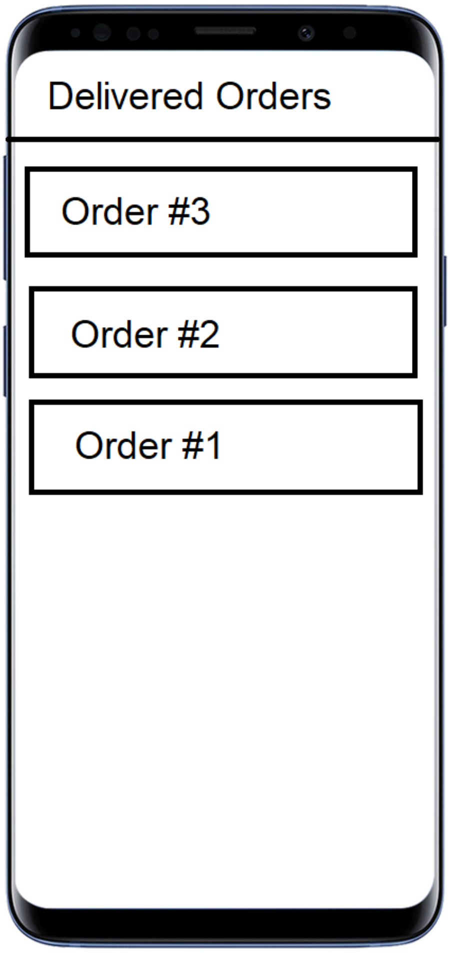

Figure 6.8: Delivered/Completed Orders Section

7 Attributes and Methods

7.1 User

In this section, we have details about the attributes to be used in the User section, as well as all the methods that are used by system to perform user functions.

7.1.1 User Attributes

Attribute	**Type**
userId	string
emailId	string
contactNo	string
password	hash string

7.1.2 User Methods

login()	
Input	email and password
Output	Authorize the user, and unlock all features
Description	Input is checked in the database. If the input matches the data, user is authorized

signUp()	
Input	email, userId, contactNo, password
Output	E-mail verification link is sent
Description	The system checks if e-mail is previously registered, and if not registered then, it sends the verification mail

getUserDetails()	
Input	userId
Output	All the attributes of the user except password
Description	The system searches the database and retrieves the details of the user associated with the given userId

removeCartItem()	
Input	userId, item, buildContext
Output	item list decreses in size
Description	The system will remove the given item from the user with given userId and update the database

editCartItem()	
Input	userId, itemId, quantity, and build context
Output	Cart is updated
Description	The system updates the data of cart associated to the given userId

addMoney()	
Input	userId, amount(int), and buildContext
Output	Balance update in user profile
Description	The system updates the balance of the given userId

orderReceived()	
Input	userId, orderId, qrCode, buildContext
Output	bool output, whether the order receiving was completed or not
Description	The system verifies the code and checks whether the order matches the user in the database and reflects on the screen about the result

logOut()	
Input	buildContext
Output	session logged out
Description	The user will be logged out of the system and will go back to the login screen

7.2 Restaurant

7.2.1 Restaurant Attributes

Attribute	**Type**
restaurantId	string
name	string
emailId	string
password	hash string
category	string

7.2.2 Restaurant Methods

login()	
Input	email and password
Output	Authorize the user, and unlock all features
Description	Input is checked in the database. If the input matches the data, user is authorized

signUp()	
Input	email, userId, contactNo, password
Output	E-mail verification link is sent
Description	The system checks if e-mail is previously registered, and if not registered then, it sends the verification mail

getUserDetails()	
Input	userId
Output	All the attributes of the user except password
Description	The system searches the database and retrieves the details of the user associated with the given userId

addNewItem()	
Input	itemName(string), price(int), quantity(int), and time(int), restaurantId
Output	new item will be added in the list of items
Description	The system will take the new item and add it to the menu of the restaurant with the given restaurantId

editItem()	
Input	itemId, itemName(string), price(int), quantity(int), and time(int), restaurantId
Output	Item with the given id will be updated in the list of items
Description	The system will find the item in the menu of the restaurant with given restaurantId and update its details

deleteItem()	
Input	itemId, restaurantId, buildContext
Output	The size of the menu list will be decreased
Description	The system will find the item with the given id from the menu associated to the restaurant with given restaurantId and delete that item

orderReceived()	
Input	restaurantId, orderId, qrCode buildContext
Output	bool output, whether the order receiving was completed or not
Description	The system verifies the code and checks whether the order matches the user in the database and reflects on the screen about the result

scanQRCode()	
Input	userId, buildContext
Output	scanned qr code is converted into string data
Description	The system converts the scanned qr code into string data and stores it for the verification

myOrders()	
Input	restaurantId
Output	list of all the pending orders for the restaurant with given restaurantId
Description	The system checks the database and selects all the orders for the given restaurant whose status is pending

logOut()	
Input	buildContext
Output	session logged out
Description	The restaurant will be logged out of the system and will go back to the login screen