Qidrah Delivery App

فول *Stack Team*

Team Members:

Mohammed Mahmoud Mohammed
Doha Mahmoud Ramadan Mohammed
Karim Hussam Al-Din El-Sayed
Sylvia Sobhy Ibrahim Rezk
Zienab Abdelsamea Mohamed Ahmed

- **Introduction:**
  Food Delivery Application is a hybrid application [mobile and web] application for ordering food online from various restaurants.
  This platform will provide a convenient way to browse through the different restaurant menus and order without going to their individual sites.
- **Goal:**

    - The goal of this application is to save user’s efforts by providing a single platform to browse and order from nearby restaurants.
    - Qidrah is an app that delivers fresh and delicious ful medames straight to your doorstep. With Qidrah, you can easily find and order from the best ful medames shops in your area, all displayed on a convenient map. Whether you're a busy student or just craving some authentic Egyptian food, Qidrah makes it easy and convenient to enjoy this beloved dish anytime, anywhere.
- **Overview:**
- Idea: Delivery Application
- Type of Project: Interactive transaction
- Type of software: Generic
    - A user will have an account for the app service. It will have menus from restaurants that have subscribed to this service.
    - A user can browse through different menus and place an order. The customer has an option to pay the bill using the app itself or pay on delivery.
    - The order will be placed at a restaurant nearest to the customer and the subscribed delivery service will be notified.
    - For delivering food charges will be applied. An agent from the delivery service will pick up the order and deliver it to the customer
- **Objectives of the Project:**

  The objective of this project is to develop a web application that allows the customers to order their desired food item from their working location.

  The following points simplify the objectives of this project more clearly:
    - To increase efficiency and improve services provided to the customers through better application of technology in daily operations.
    - To enable customers to order their desired food through the internet.
    - To make all the work easier and computerised.
    - To improve efficiency of the restaurant's staff.

- To increase speed of service, sales volume and customer satisfaction.

- **Methodology:**

  The Software Development Model

  ➢ *We use agile methodology to develop our software as:*
  - *Agile methodology is a flexible and iterative approach that focuses on delivering working software in short iterations.*
  - *It involves continuous collaboration between the development team and the stakeholders, allowing for frequent feedback and adjustments to be made throughout the development process.*
  - *For a food delivery application, this approach would be ideal as it allows for quick adaptation to changing customer needs and preferences. So, we are always able to ensure that the application meets the customer requirements.*
  - *Agile methodology emphasises testing and quality assurance throughout the development process, ensuring that the final product is reliable and user-friendly. Which is crucial for a food delivery application as it involves handling sensitive customer information such as payment details and delivery addresses.*

- ***Requirements Analysis:***
  - **User Requirements:**

1. Easy to use interface: The application should have a user-friendly interface that is easy to navigate and understand.
2. Menu and restaurant selection: The app should provide a wide range of restaurants and menus to choose from, with clear descriptions of the dishes, prices, and availability.
3. Customization options: Users should be able to customize their orders according to their preferences, such as adding or removing ingredients or selecting the level of spiciness.

4. Payment options: The app should offer multiple payment options, including credit/debit cards, mobile wallets, and cash on delivery.
5. Real-time tracking: Users should be able to track their order in real-time from the moment it is placed until it is delivered.
6. Delivery time estimation: The app should provide an estimated delivery time based on the distance between the restaurant and the user's location.
7. Order history: Users should be able to view their order history for future reference or reordering.
8. Customer support: The app should have a customer support system in place to address any issues or concerns that users may have.
9. Ratings and reviews: Users should be able to rate and review restaurants based on their experience with the food quality, delivery time, and overall service.
10. Promotions and discounts: The app can offer promotions and discounts to users as an incentive for using the service regularly.

- **System Requirements:**

The system requirements for a food delivery application may vary depending on the specific features and functionalities of the app. However, some common system requirements include:

1. Operating System: The app should be compatible with both iOS and Android operating systems.

2. Internet Connectivity: The app should require a stable internet connection to function properly.

3. GPS Tracking: The app should have GPS tracking capabilities to locate the user's current location and track the delivery driver's location.

4. Payment Gateway Integration: The app should integrate with popular payment gateways like PayPal, Stripe, or Braintree to facilitate secure online payments.

5. Push Notifications: The app should have push notification functionality to keep users updated about their order status, delivery time, and other important information.

6. User Authentication: The app should have a secure login system that requires users to create an account using their email address or social media accounts like Facebook or Google.

7. Menu Management System: The app should have a menu management system that allows restaurants to update their menu items, prices, and availability in real-time.

8. Order Management System: The app should have an order management system that allows users to place orders, track their status, and cancel orders if necessary.

9. Customer Support System: The app should have a customer support system that allows users to contact customer support via chat or phone in case of any issues or queries.

10. Analytics Dashboard: The app should have an analytics dashboard that provides insights into user behavior, order trends, revenue generation, etc., to help businesses make informed decisions about their operations and marketing strategies.

The system requirements are classified into: functional requirements and non-functional requirements:

- **Functional Requirements:**

1. User registration and login: The application should allow users to register and create an account with their personal details,

including name, email address, phone number, and password. Users should be able to log in using their credentials.

2. Menu display: The application should display the menu of various restaurants with their prices and descriptions.

3. Ordering system: The application should allow users to place orders from the menu of different restaurants. Users should be able to select the items they want to order, specify any special instructions or dietary requirements, and add them to their cart.

4. Payment gateway integration: The application should integrate a secure payment gateway that allows users to pay for their orders online using various payment methods such as credit/debit cards, net banking, or digital wallets.

5. Order tracking: The application should provide real-time updates on the status of the order, including confirmation of receipt of the order by the restaurant, preparation time, dispatch time, delivery time estimation, and delivery confirmation.

6. Delivery management: The application should have a delivery management system that assigns delivery personnel based on availability and proximity to the restaurant and customer's location. Delivery personnel should be able to update their status on the app once they pick up the order from the restaurant and deliver it to the customer.

7. Rating and review system: The application should allow users to rate restaurants based on food quality, delivery speed, customer service, etc., and leave reviews about their experience.

8. Customer support: The application should provide customer support through chatbots or live chat options for any queries or complaints related to orders or payments.

9. Loyalty programs: The application can offer loyalty programs such as discounts or cashback offers for frequent customers or referrals.

10. Restaurant management system: The application can provide a dashboard for restaurant owners/managers that allows them to manage their menus, update prices/availability of items in real-time, track orders received, and manage their delivery personnel.

- **Non-functional Requirements:**

  1. Performance: The application should be able to handle a large number of users and orders without any lag or delay.

  2. Reliability: The application should be reliable and available 24/7 without any downtime or crashes.

  3. Security: The application should have robust security measures in place to protect user data, payment information, and other sensitive information.

  4. Usability: The application should be easy to use and navigate for both customers and delivery personnel.

  5. Scalability: The application should be able to scale up or down depending on the demand for the service.

  6. Compatibility: The application should be compatible with different devices, operating systems, and browsers.

  7. Accessibility: The application should be accessible to people with disabilities, including those who are visually impaired or have mobility issues.

  8. Localization: The application should support multiple languages and currencies to cater to a global audience.

  9. Integration: The application should integrate seamlessly with other systems such as payment gateways, inventory management systems, and customer relationship management tools.

  10. Maintenance: The application should be easy to maintain and update regularly with new features and bug fixes.

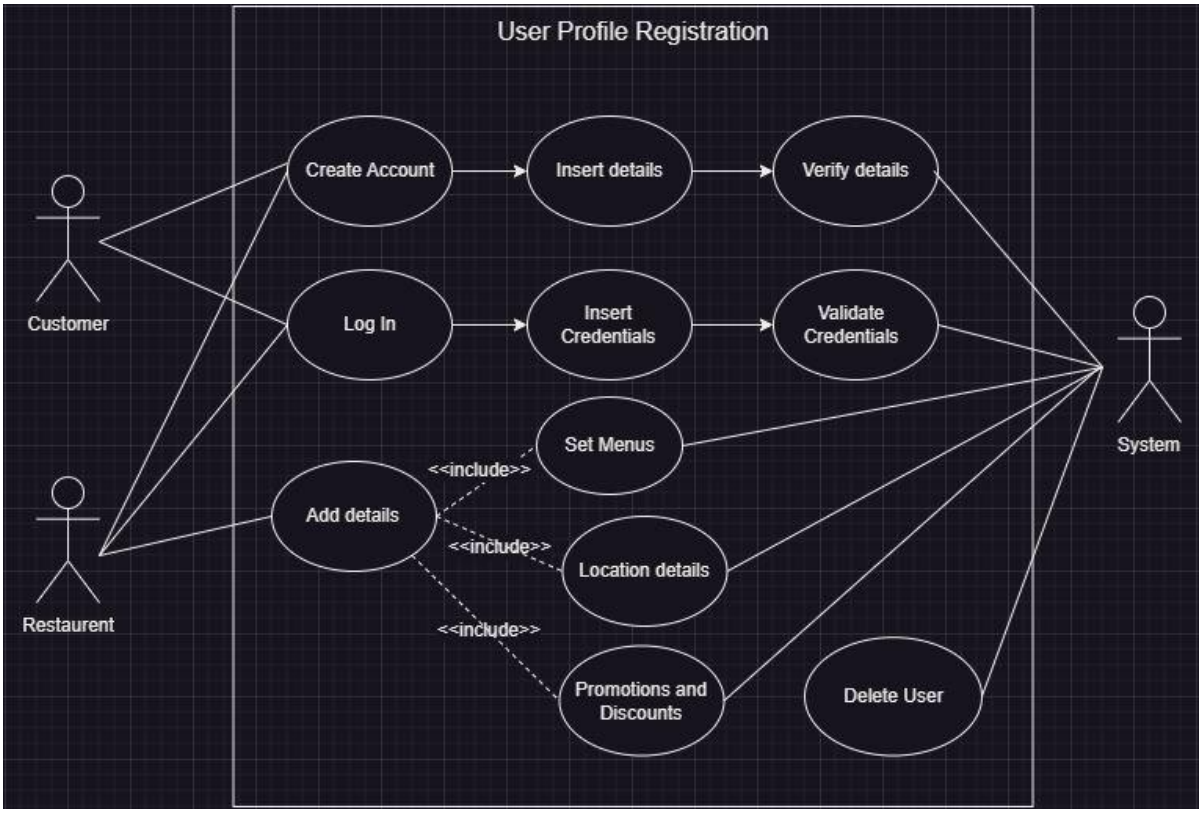
- **Assumption and constraints**:
    - The app should be available on the phone.
    - Restaurant will be registered for the app service.
    - Delivery service has a contract with the App service and restaurants.
    - The selected restaurants should be within the range of delivery service.
    - The order should be placed at least 30 minutes before restaurant closing time

- **Analysis**

Use Case Diagram

This will include 3 use case diagrams for the system.

1. User profile registration



Use Case Description for User Registration:

Use case name	User profile registration
Actors	User, System, Restaurant
Description	The user logs in using this app and orders food from the nearest restaurant.
Trigger	The user must be an existing one or properly registered in the case of new user to place the order.
Relationships	Association: User, System, App
Normal Flow of Events	1. The user will be displayed the nearest restaurants based on his/her location. 2. A new user will register on the app or an existing user will log in to order food. 3. The credentials will be validated by the system. 4. User will select an item from menu. 5. Items will be added or deleted from the cart. 6. User will provide payment details for the transaction. 7. The order will be validated by the system; first, whether the order is placed at least 30 minutes before restaurant closing time; second, the type of order whether it's a pickup or delivery; and third, for checking whether the selected restaurant is within the range of delivery service (in case of delivery order). 8. Final order will be placed.
Exceptions	2a. If the credentials are invalid, the user will enter the login / register

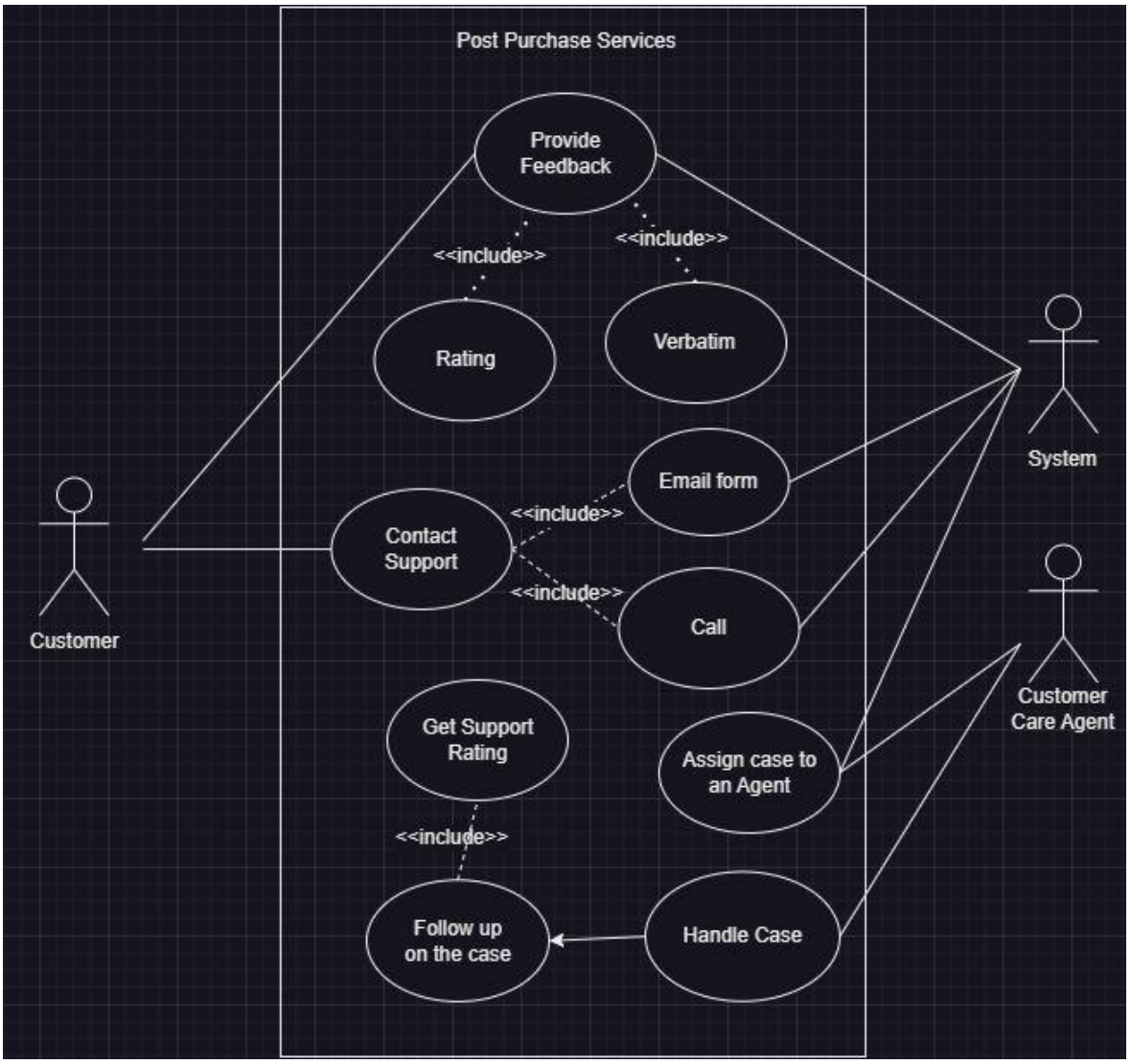
	details again. 1. The selected item must be available. 2. If the order is out of range of the delivery service or the order is placed less than 30 min of the restaurant closing time, the order will be discarded. 6a. If provided payment details are incorrect then transaction will be cancelled.

Use Case Description for placing order:

Use case name	Placing order
Actors	System , Restaurant ,Authentication

	services
Stakeholder and Investor	Customer
Description	This use case describe how delivery order services get placed
Trigger	While ordering food, the user has requested for home delivery.
Relationships	Association: Customer, System, Restaurant, Delivery System Generalisation: Cash on delivery will depend on customer's choice.
Normal flow of Events	1. Admin will notify the Delivery system about an order 2. Delivery system will notify the nearest driver about the order to pickup 3. Driver will pick up the order from specified restaurant and notification will be sent to Admin and Delivery System. 4. The Order will be delivered to customer and notification will be sent to Admin and Delivery System
Alternate/Exceptional Flow	4a. If cash on delivery, the Driver has to collect cash as indicated on the receipt.

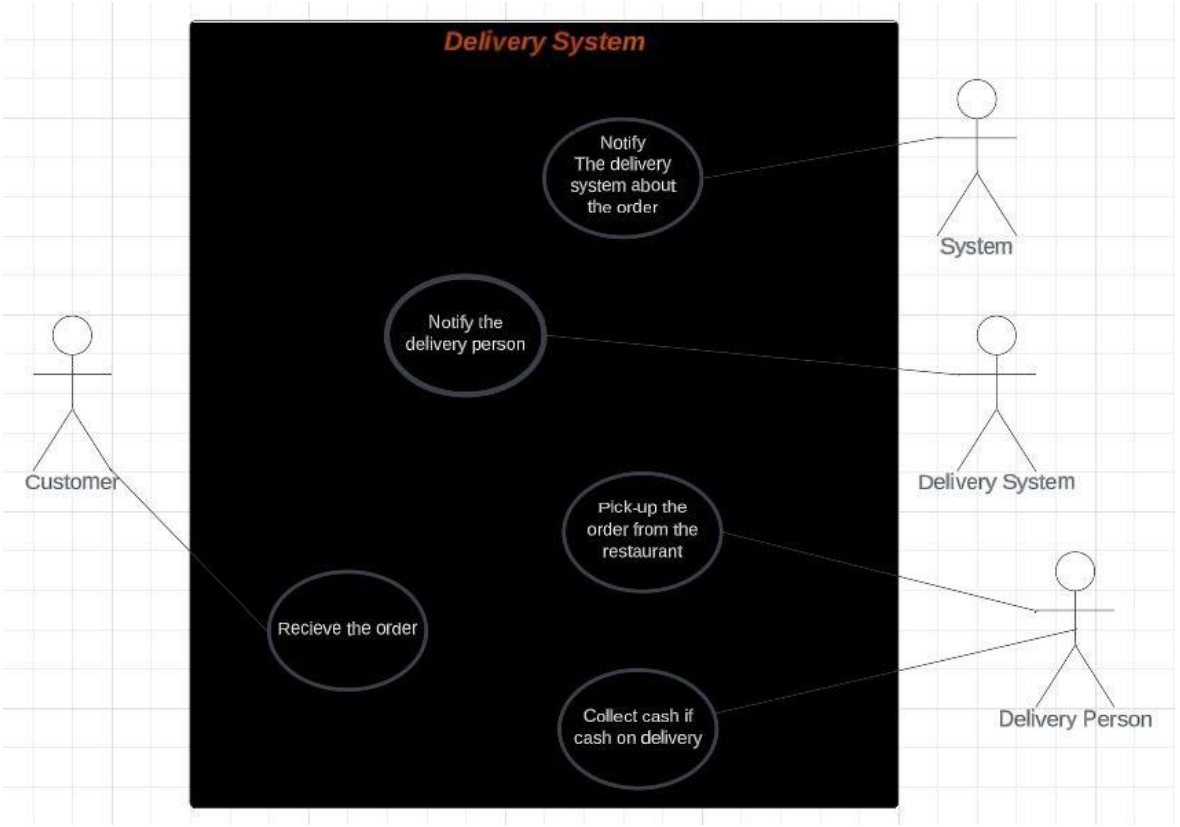
Use case : Post purchase services



Use case name	Post purchase services
Actors	Customer, Care agent, System
Description	The customer may contact customer care for support and provide his feedback.
Trigger	When the customer contacts the customer care.
Relationships	Association
Normal flow of Events	1. The customer will contact the

	customer care service using their email or phone number. 2. Customer care will receive the customer feedback. 3. cIn case of the existence of an issue, customer care will work on it and try to do their best to handle and get customer satisfaction.
Exceptions	Customer care doesn't respond.

Use case : Delivery System:

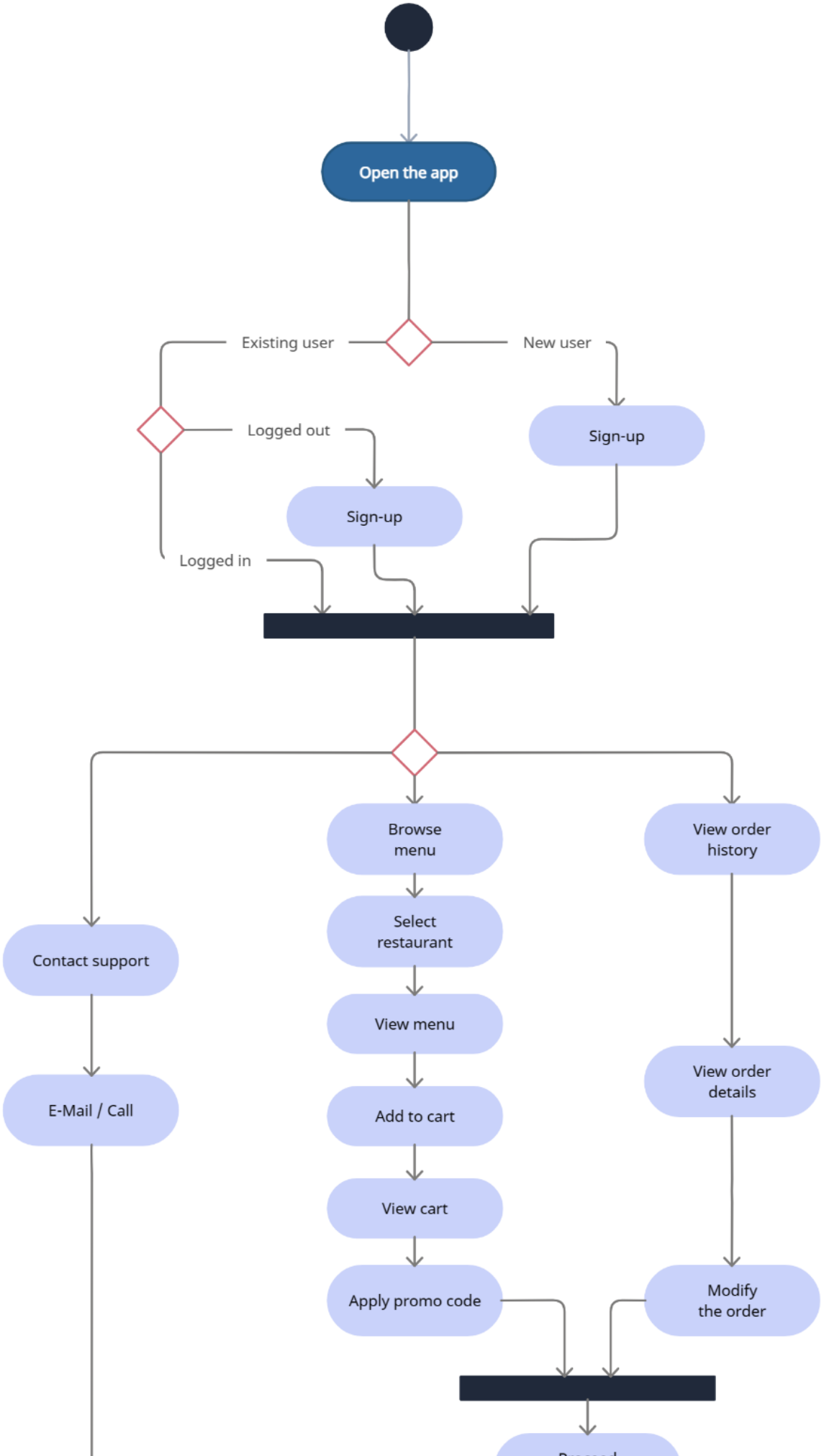


Use case name	Delivery System
Actors	Customer, System, Delivery System, Delivery Person

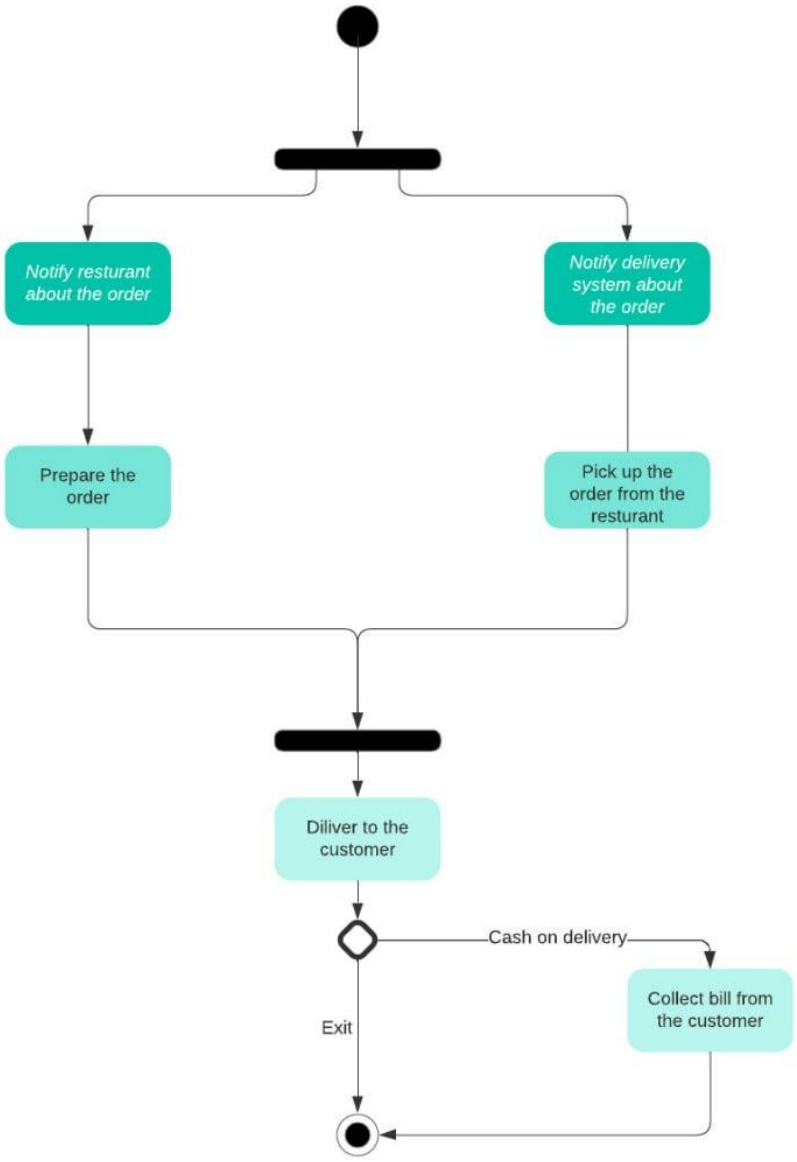
Description	The driver will pick up the order from the restaurant and deliver it to the customer.
Trigger	When the customer makes a delivery order.
Relationships	Association
Normal flow of Events	1. The customer will make an orde. 2. The system will notify the delivery system about the order. 3. The delivery system will notify the delivery person (driver). 4. The driver will pick-up the order from the restaurant. 5. The customer will receive the order. 6. The delivery person will collect cash if cash on delivery.
Exceptions	The customer chose to buy by credit card or bank account.

ERD - (EER) Diagram

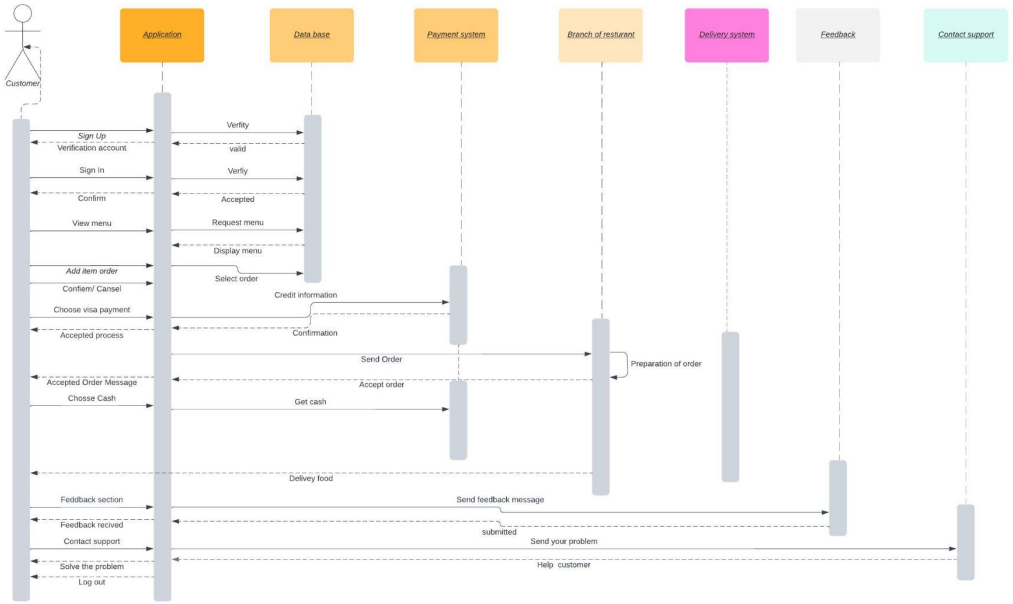
Activity Diagram: System Flow


Open the app
Existing user
New user
Logged out
Sign-up
Sign-up
Logged in
Contact support
Browse menu
View order history
Select restaurant
View menu
View order details
E-Mail / Call
Add to cart
View cart
Apply promo code
Modify the order

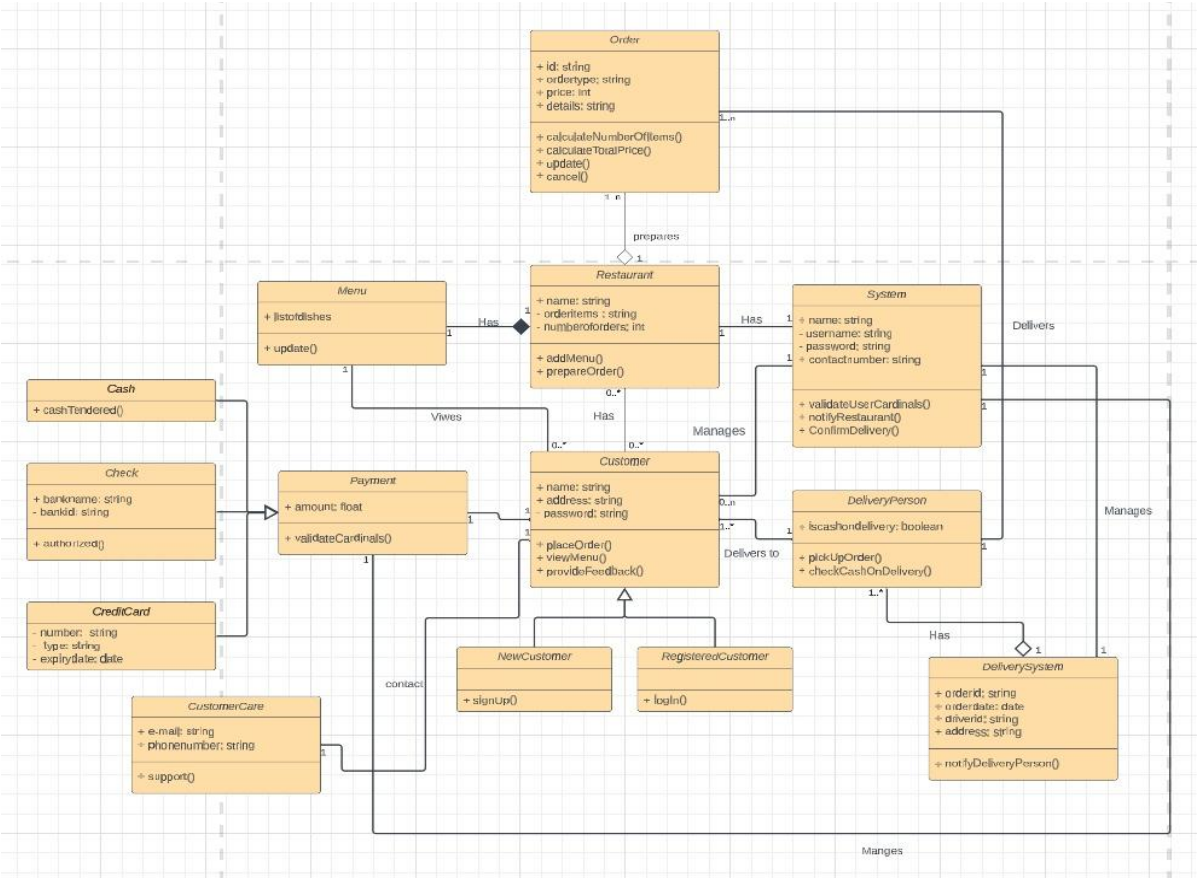
Activity Diagram: Delivery System



Sequence Diagram



Class Diagram



- **Functional Test Plan**

  It would include all the functional test cases for the project.

  1. Test Cases: This section will provide an idea about how the functionality works after the software is ready to use.
  2. The test cases provide what are the valid or invalid data input for a particular feature. Based on the inputs of the user the respective response will get.

➢ Sign UP &Login test case for both User and Restaurant Panel

Action	**Input**	**Expected Output**	**Description**
Valid Sign up	Username, Password	Navigated to next screen	User Successfully signup into Food Order app.
Valid Login	Username, Password	Navigated to next screen	User Successfully Login into Food Order app.
Invalid Login	Username, Password	Alert box with message “Invalid Credentials” and ask for credentials again Unauthorized	user trying to login.

➢ Test Case for Admin Panel

Action	**Input**	**Expected Output**	**Description**
Orders Approval	Orders approved from user	List of Orders in Queue which are out for Delivery	Orders approved

Feedback	Select the Accessories	Display Feedback by user so far.	Feedback Successfully
Payments bifurcation	Select the Orders separate bills	Payment online through banking	Transactions successfully.