



IT-632 Software Engineering  
Group ID: 26  
Project Title: Stellar Food Ordering System

### Team Members

Student ID	Name
202312101	Hitesh Agarwal
202312037	Varshil Shah
202312044	Krupesh Modi
202312027	Anmol Rangwani

# INDEX

<b>Description</b>	<b>Page No.</b>
<a href="#"><u>Introduction</u></a>	3
<a href="#"><u>Proposed System</u></a>	4
<a href="#"><u>Scope of System</u></a>	5
<a href="#"><u>Technologies Used</u></a>	6
<a href="#"><u>Modules</u></a>	7
<a href="#"><u>System Perspective</u></a>	11
<a href="#"><u>SA/SD</u></a>	14
<a href="#"><u>UML Diagrams</u></a>	21
<a href="#"><u>Screenshots</u></a>	31

# Introduction

Welcome to Stellar, where culinary delights are just a tap away! We understand that your cravings know no bounds and your time is precious. That's why we've crafted the ultimate food delivery experience, bringing the world's flavors to your doorstep with just a few clicks.

At Stellar, we believe in connecting you with a diverse array of restaurants, ensuring that every craving, from the comfort food classics to exotic international cuisines, is satisfied.

Our user-friendly interface makes ordering a breeze, allowing you to explore menus, customize your selections, and track your delivery in real-time.

Whether you're seeking a quick bite, planning a family feast, or hosting a virtual gathering, our app seamlessly caters to your every culinary need.

We prioritize convenience without compromising on quality, partnering with top-rated restaurants to deliver a gastronomic experience that exceeds expectations.

# Proposed System

- A food delivery application is a digital platform designed to facilitate the ordering and delivery of a variety of food items from local restaurants and eateries.
- Users can access the application through their smartphones, browse menus from participating restaurants, customize their orders, and place requests for home or office delivery.
- These applications typically provide a user-friendly interface, real-time order tracking, secure payment options, and reviews/ratings for restaurants, ensuring a seamless and convenient experience for customers seeking a diverse range of culinary options without leaving their homes.
- Food delivery applications serve as an intermediary between consumers and local eateries, streamlining the process of food ordering and delivery.

# Scope of System

## Market Accessibility

- **Local and Global Reach:** Food delivery apps have the potential to connect local restaurants and eateries with a broader customer base, transcending geographical boundaries.
- **Diverse Cuisines:** Users can access a wide variety of cuisines, ranging from local favorites to international specialties, thereby catering to diverse taste preferences.

## User Convenience

- **Timesaving:** Food delivery apps offer users a convenient and time-saving alternative to traditional dining or cooking, making it easier for individuals with busy lifestyles.
- **Customization:** Users can personalize their orders, specify dietary preferences, and add special instructions, enhancing the overall dining experience.

## Technological Integration

- **Real-Time Tracking:** The inclusion of real-time order tracking features keeps users informed about the status of their orders, providing transparency and peace of mind.
- **Secure Transactions:** Integration with secure payment gateways ensures safe and hassle-free financial transactions within the app.

## Business Opportunities

- **Restaurant Partnerships:** The app opens up new avenues for local restaurants to expand their customer base and increase sales by partnering with the platform.
- **Marketing and Promotion:** Food delivery apps often provide marketing and promotional opportunities for partnered restaurants, helping them reach a larger audience.

## Community Engagement

- **User Reviews and Ratings:** The interactive nature of these apps allows users to share their dining experiences through reviews and ratings, fostering a sense of community and trust.

# Technologies Used

## Flutter

Flutter is an open-source UI software development kit created by Google. It is used to develop cross platform applications for Android, iOS, Linux, Mac, Windows, Google Fuchsia, and the web from a single codebase. Firebase app distribution integration for Flutter developers in Codemagic just got better. Publish Flutter Apps to Firebase App Distribution. Intuitive UI helps to get the job done. It's a software development kit (SDK) with prewritten code, consisting of ready-to-use and customizable widgets, as well as libraries, tools, and documentation that together serve to build cross-platform apps.

## Google Firebase

Google Firebase is a Google-backed application development software that enables developers to develop iOS, Android and Web apps. Firebase provides tools for tracking analytics, reporting and fixing app crashes, creating marketing and product experiments. Firebase is a platform developed by Google for creating mobile and web applications. Easily integrate Firebase into your team's favorite tools. Trusted by the largest apps. Firebase helps teams from startups to global enterprises build & run successful apps. Boost App Engagement. Monitor App Performance. Cross-Platform Solutions.

## Cloud Firestore

Cloud Firestore is a cloud-hosted, NoSQL database that your iOS, Android, and web apps can access directly via native SDKs. Query & structure data the way you like. Sync data across devices, on or offline. Powered by Google's storage infrastructure, Cloud Firestore helps scale your business. Build Fast For Any Device. For Mobile or Web Apps. Backed by Google. Build Extraordinary Apps.

# Modules

## Register/Login

- **Actor:** User (Customer, Restaurant Owner, Admin)
- **Description:** Users register for an account or log in using existing credentials to access the platform's features.
- **Preconditions:** The user has access to the internet and the app is installed on their device.
- **Basic Flow:**
  1. The user opens the app.
  2. The app displays the login or registration screen.
  3. The user enters their credentials or selects the option to register.
  4. The system verifies the credentials or guides the user through the registration process.
  5. Upon successful authentication, the user gains access to the platform.

## Place Order

- **Actor:** Customer
- **Description:** Customers browse restaurants, select items, customize orders, and place them for delivery or pickup.
- **Preconditions:** The customer is logged into the platform and has selected a restaurant.
- **Basic Flow:**
  1. The customer selects a restaurant from the list.
  2. The app displays the restaurant's menu.
  3. The customer browses the menu, selects items, and customizes their order.
  4. The customer adds items to their cart and proceeds to checkout.
  5. The customer selects delivery or pickup options and provides necessary details.

6. The system confirms the order and sends it to the restaurant for preparation.

## Manage Menu

- **Actor:** Restaurant Owner
- **Description:** Restaurant owners add, update, or remove menu items and manage their availability on the platform.
- **Preconditions:** The restaurant owner is logged into their account.
- **Basic Flow:**
  1. The restaurant owner navigates to the menu management section of the app.
  2. The app displays options to add, edit, or delete menu items.
  3. The restaurant owner selects the desired action (add, edit, delete).
  4. The restaurant owner provides details for new items or updates existing ones.
  5. The system updates the menu accordingly and reflects changes on the platform.

## Track Order

- **Actor:** Customer, Delivery Driver
- **Description:** Customers track the status of their orders, while delivery drivers view assigned orders and update their status upon delivery.
- **Preconditions:** The customer has placed an order, and the delivery driver has accepted it.
- **Basic Flow:**
  1. The customer opens the app and navigates to the order tracking section.
  2. The app displays the status of the order (e.g., preparing, out for delivery).
  3. The delivery driver receives the order assignment and navigates to the pickup location.
  4. The delivery driver updates the order status upon pickup and delivery.



5. The system updates the order status in real-time and notifies the customer accordingly.

## Provide Feedback

- **Actor:** User (Customer, Restaurant Owner, Delivery Driver)
- **Description:** Users provide feedback about their experience with the platform, a specific order, or service provided.
- **Preconditions:** The user is logged into the platform and has completed a transaction.
- **Basic Flow:**
  1. The user navigates to the feedback section of the app or website.
  2. The system presents options for providing feedback, such as rating and comments.
  3. The user selects the type of feedback they want to provide.
  4. The user provides feedback and submits it to the system.
  5. The system records the feedback and may notify relevant parties (e.g., platform administrators, service providers).

## View Order History

- **Actor:** Customer
- **Description:** Customers view a history of their past orders placed through the platform.
- **Preconditions:** The customer is logged into the platform.
- **Basic Flow:**
  1. The customer navigates to their account settings or order history section.
  2. The system displays a list of past orders, including order details and status.
  3. The customer selects a specific order to view additional details if needed.
  4. The system provides options to filter or search for specific orders.
  5. The customer can review past orders and their status for reference or reordering.

## Check Active Deliveries

- **Actor:** Delivery Person
- **Description:** Delivery personnel can monitor and check the status of their currently active deliveries in real-time.
- **Preconditions:** The delivery person is logged into the platform and has at least one active delivery assignment.
- **Basic Flow:**
  1. The delivery person navigates to the main dashboard or active deliveries section in their account settings.
  2. The system displays a real-time list of all currently active deliveries assigned to the delivery person, including relevant details such as order information, delivery status, and customer details.
  3. The delivery person can select a specific active delivery to access additional details, such as the customer's location, order specifics, and any special instructions provided by the customer.
  4. The system continuously updates the delivery status, providing live tracking information for each active delivery.
  5. The delivery person has the option to mark certain milestones within the delivery process, such as "en route," "arrived at destination," or "delivered," to keep both the system and the customer informed about the progress.

# System Perspective

## System Functions

### User Registration and Authentication

- Users should be able to register with the system using email, phone number, or social media accounts.
- Upon registration, users must verify their accounts through email or SMS.
- Users should be able to log in securely using their registered credentials.

### Browse and Search Restaurants

- Users should be able to browse a list of restaurants based on their location.
- Users must have the ability to search for restaurants by cuisine, name, or location.

### View Menus and Place Orders

- Users should be able to view menus of restaurants including prices and item descriptions.
- Users must be able to add items to their cart and place orders for delivery or pickup.

### Order Tracking

- Customers should be able to track the status of their orders in real-time.
- Delivery drivers must have access to navigation features to reach pickup and delivery locations efficiently.

## Restaurant Management

- Restaurant owners should be able to manage their menus, including adding, editing, and deleting items.
- Restaurant owners must receive orders promptly and have the ability to confirm or reject them.

## Admin Dashboard

- Admins should have access to a dashboard to manage users, restaurants, and delivery personnel.
- Admins must be able to view analytics and reports on order volumes, revenue, and user engagement.

## Software System Attributes

### Performance

- The system must be able to handle concurrent user requests efficiently, even during peak hours.
- Response times for actions such as placing orders and updating menus should be fast to ensure a smooth user experience.

### Security

- User data, including personal information and payment details, must be encrypted, and stored securely.

- Access to sensitive functionalities (e.g., admin dashboard) should be restricted to authorized users.

## Reliability

- The system should be available and operational 24/7, with minimal downtime for maintenance or updates.
- Measures should be in place to ensure the integrity of orders and user information, preventing data loss or corruption.

## Scalability

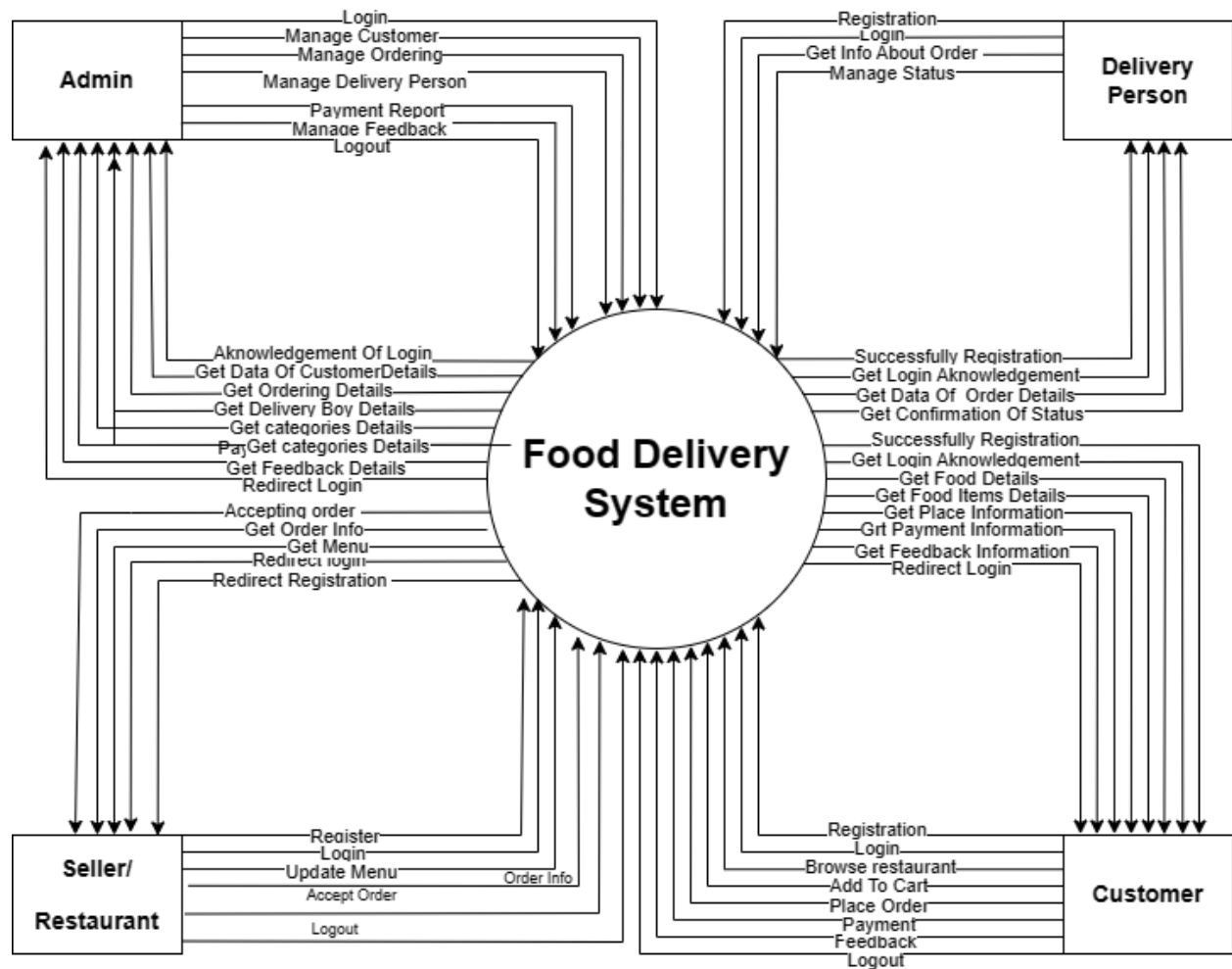
- The system architecture should be designed to scale horizontally to accommodate increases in user traffic and order volumes.
- Cloud-based infrastructure and auto-scaling mechanisms should be utilized to handle spikes in demand.

## Usability

- The user interface should be intuitive and easy to navigate, catering to users of all technical levels.
- The platform should support multiple languages and provide accessibility features to accommodate diverse user needs.

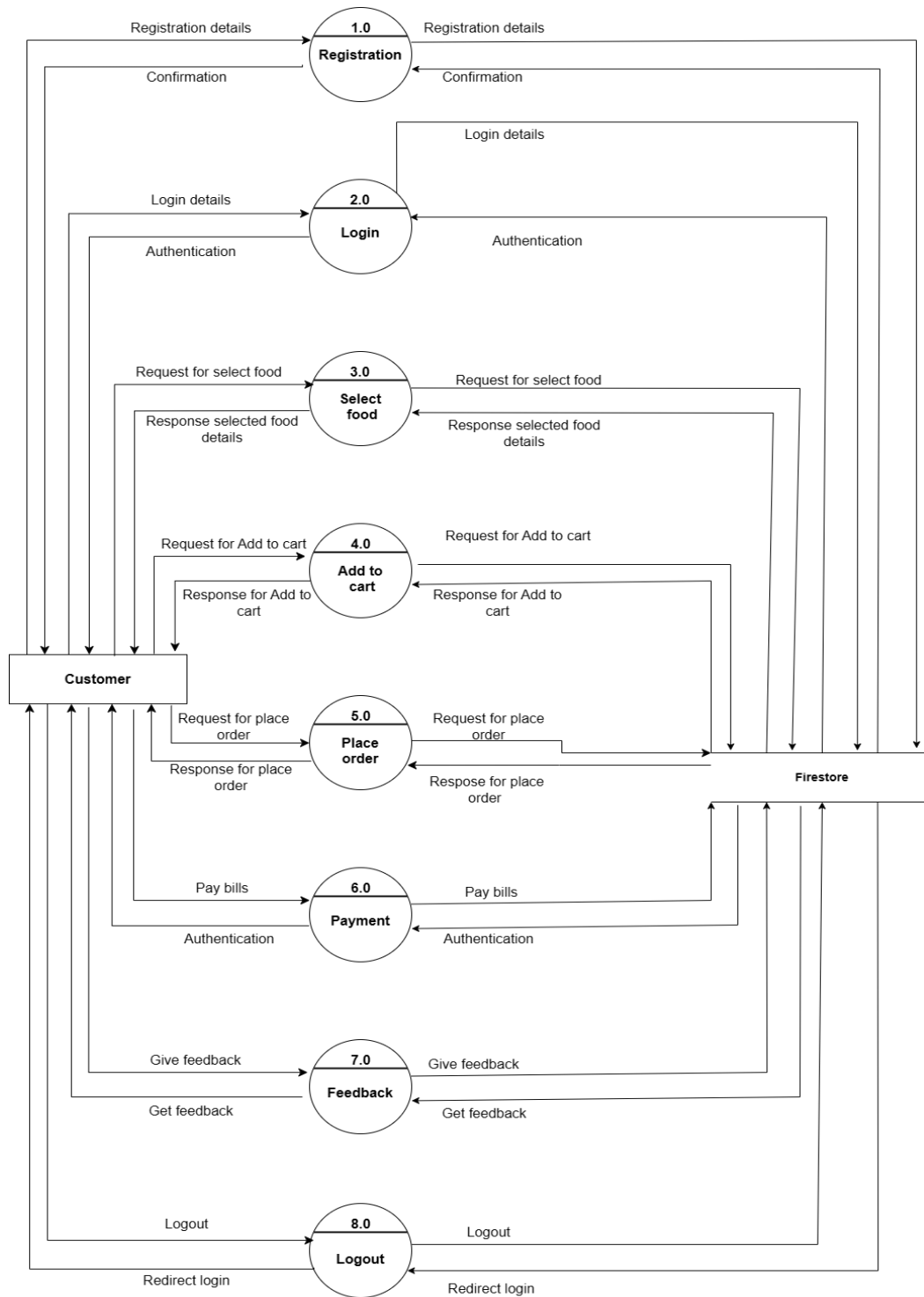
# SA/SD

## DFD Level 0

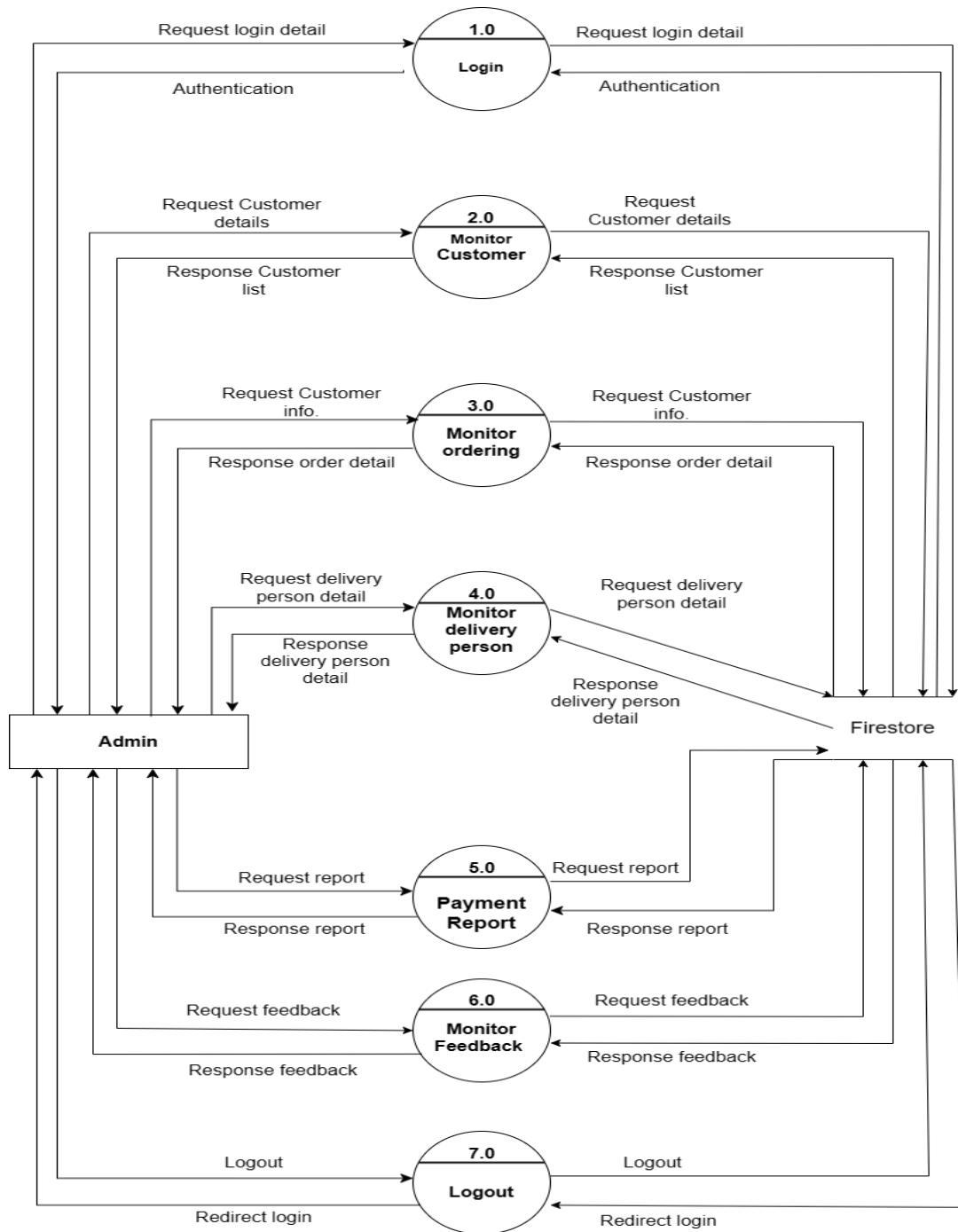


# DFD Level 1

## Customer

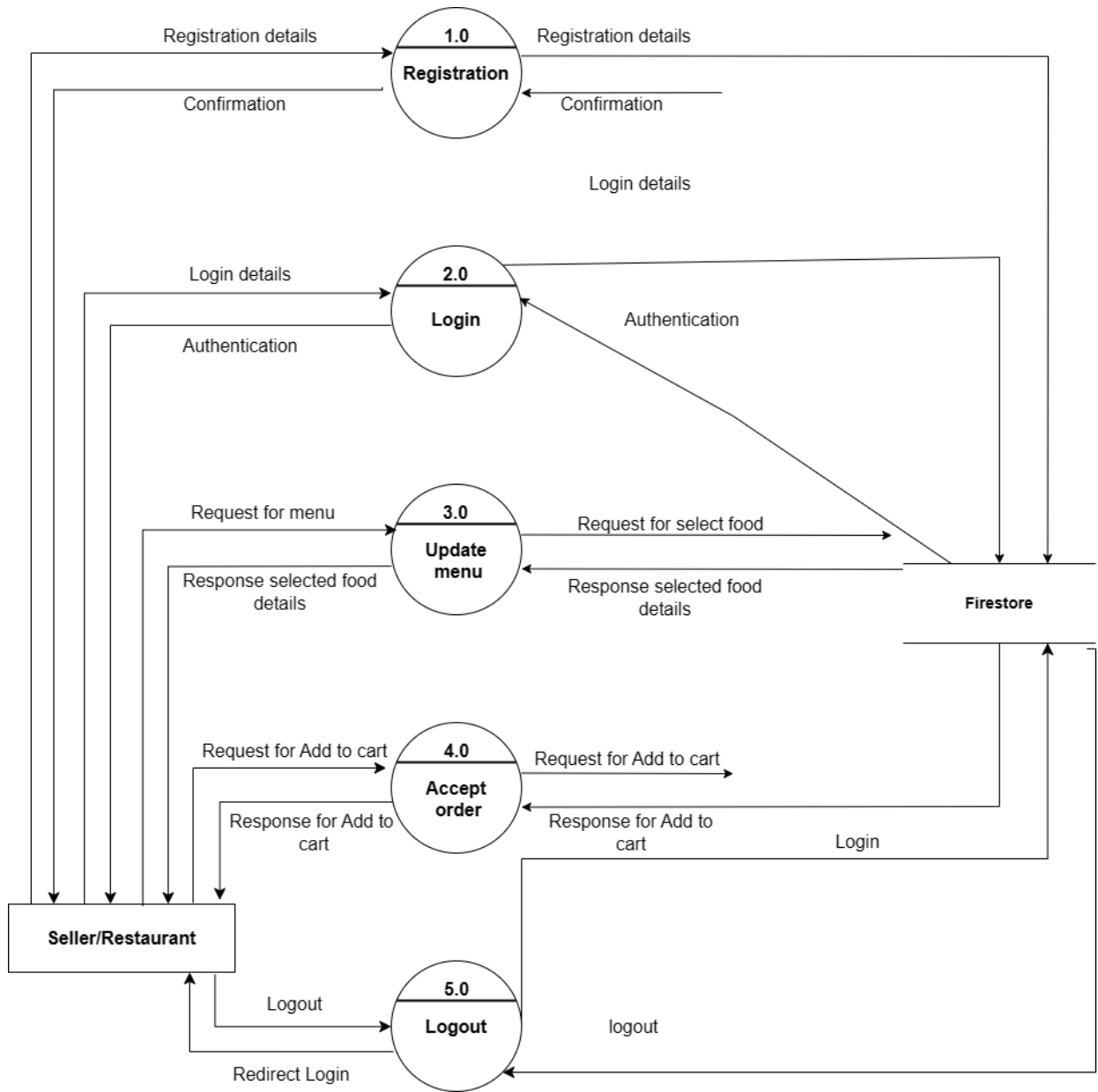


## Admin

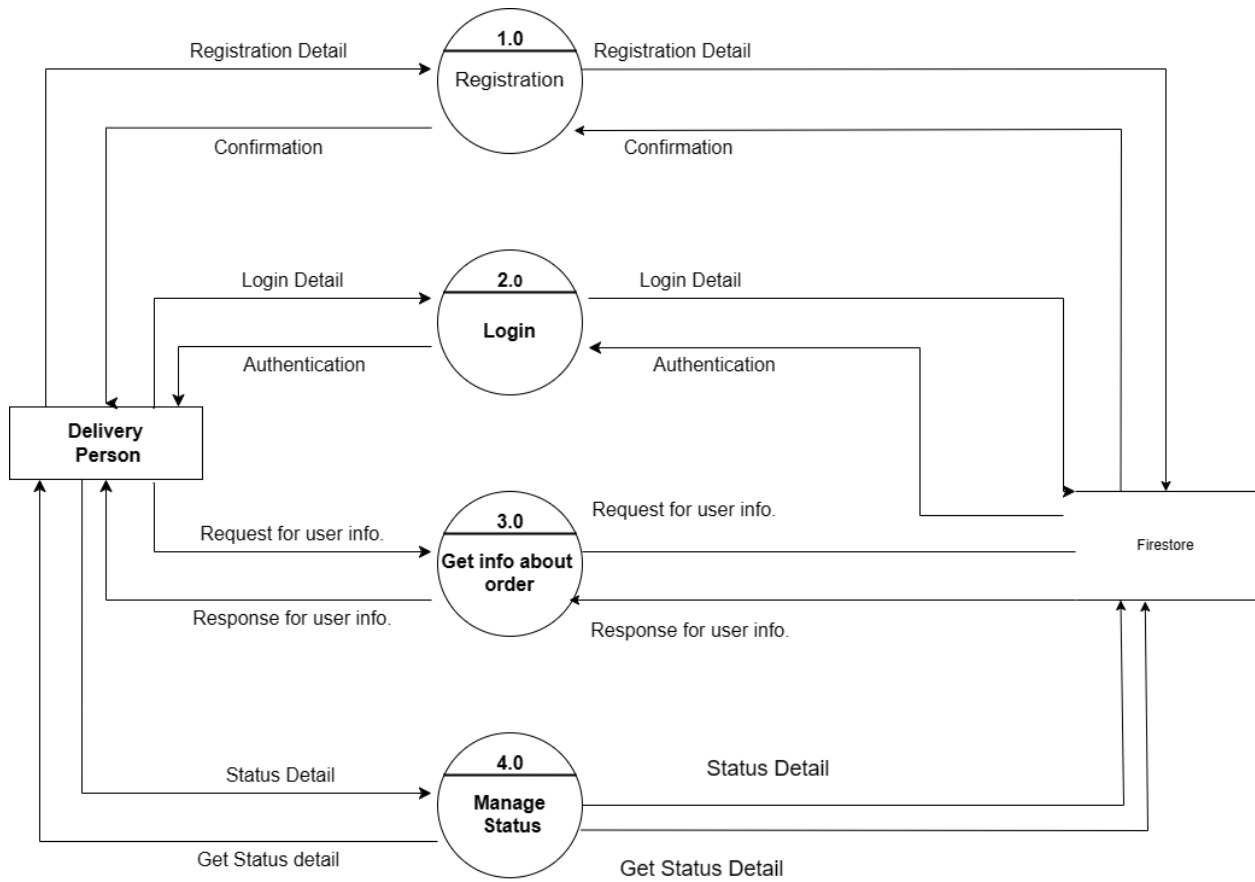




## Seller

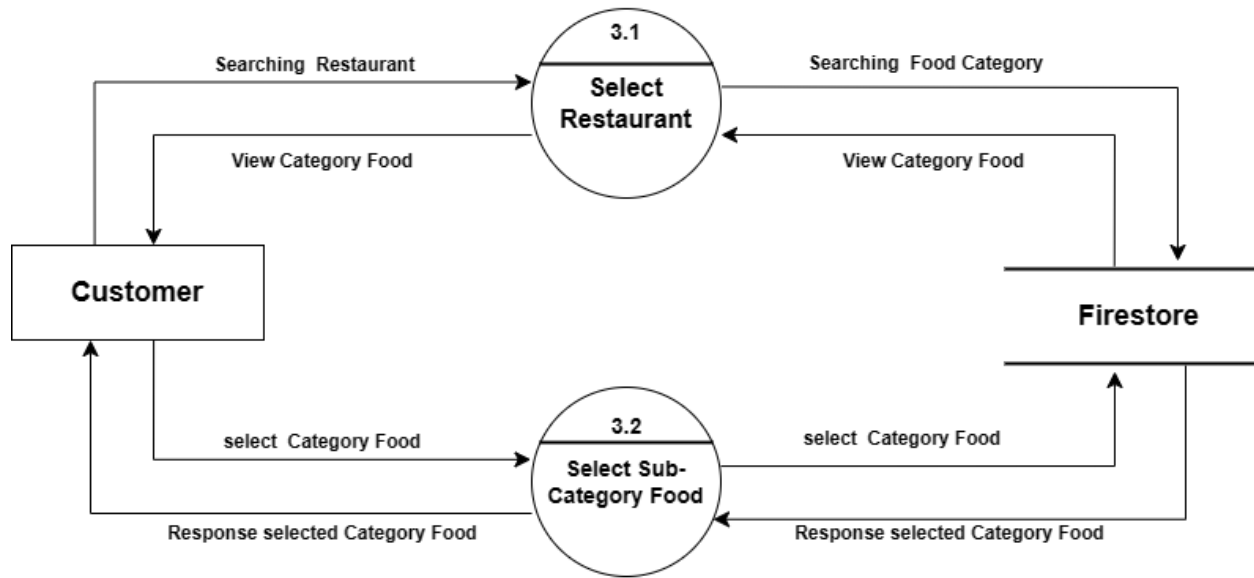


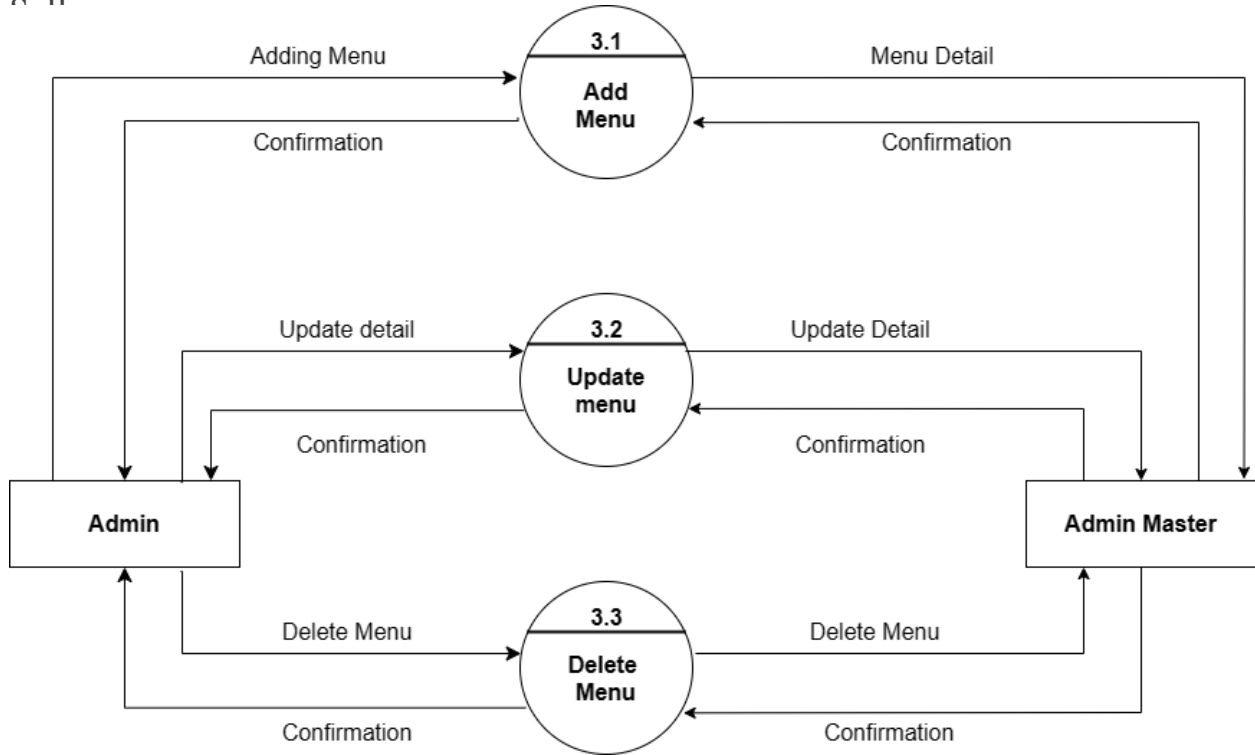
## Delivery Person



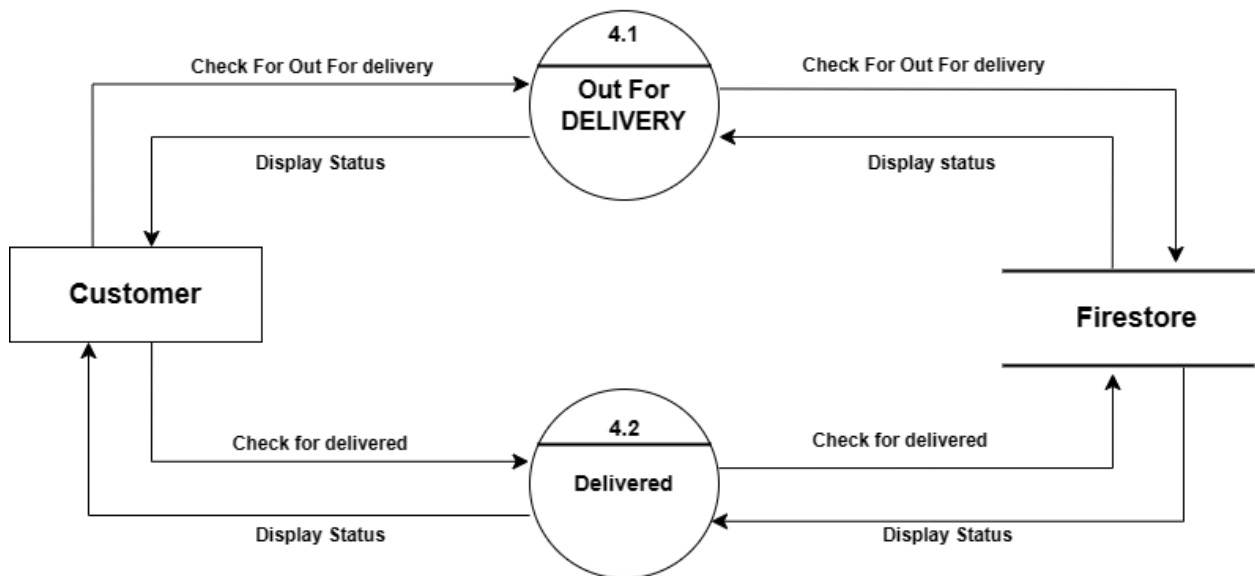
## DFD Level 2

Customer



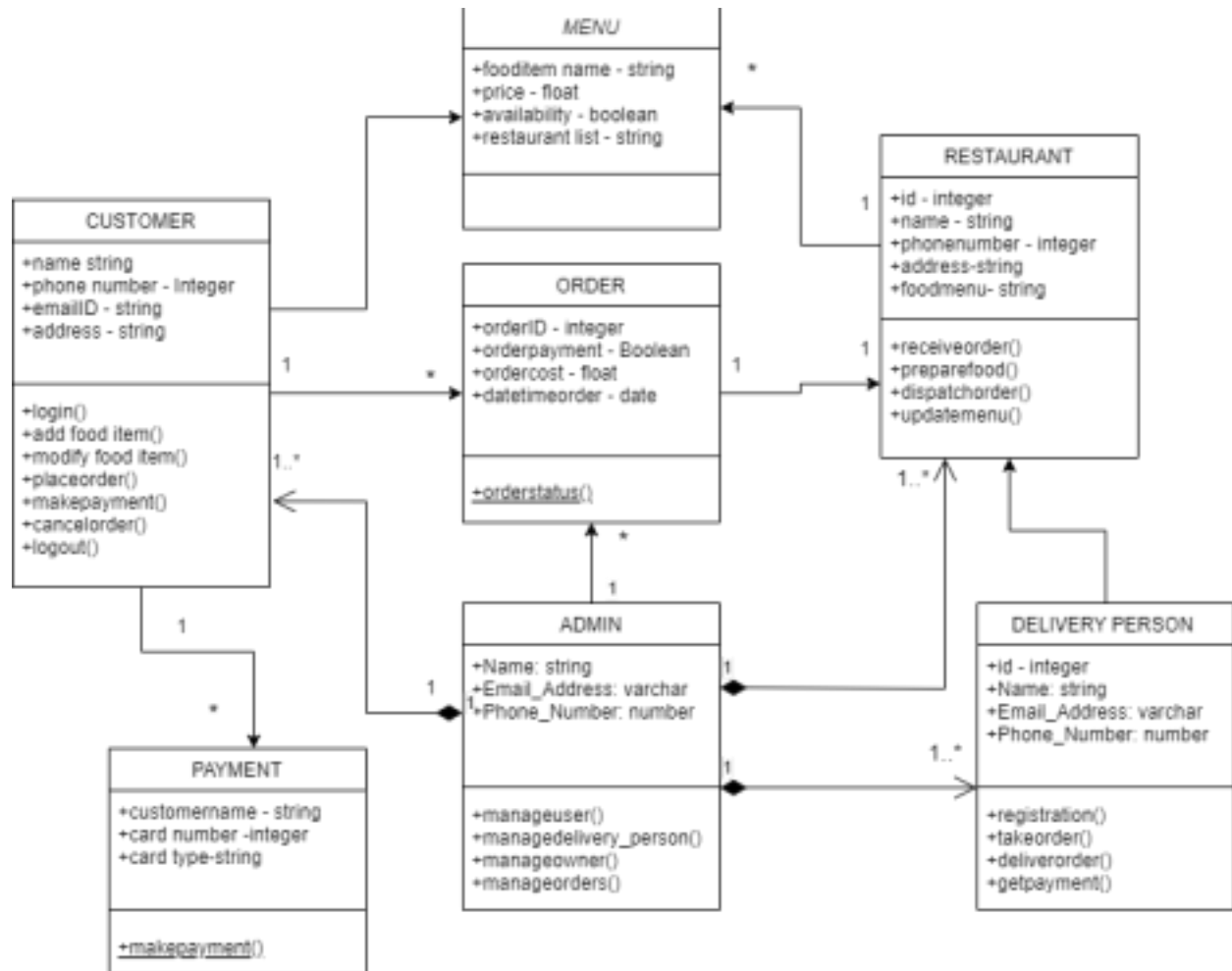


## Delivery Person



# UML Diagrams

## Class Diagram

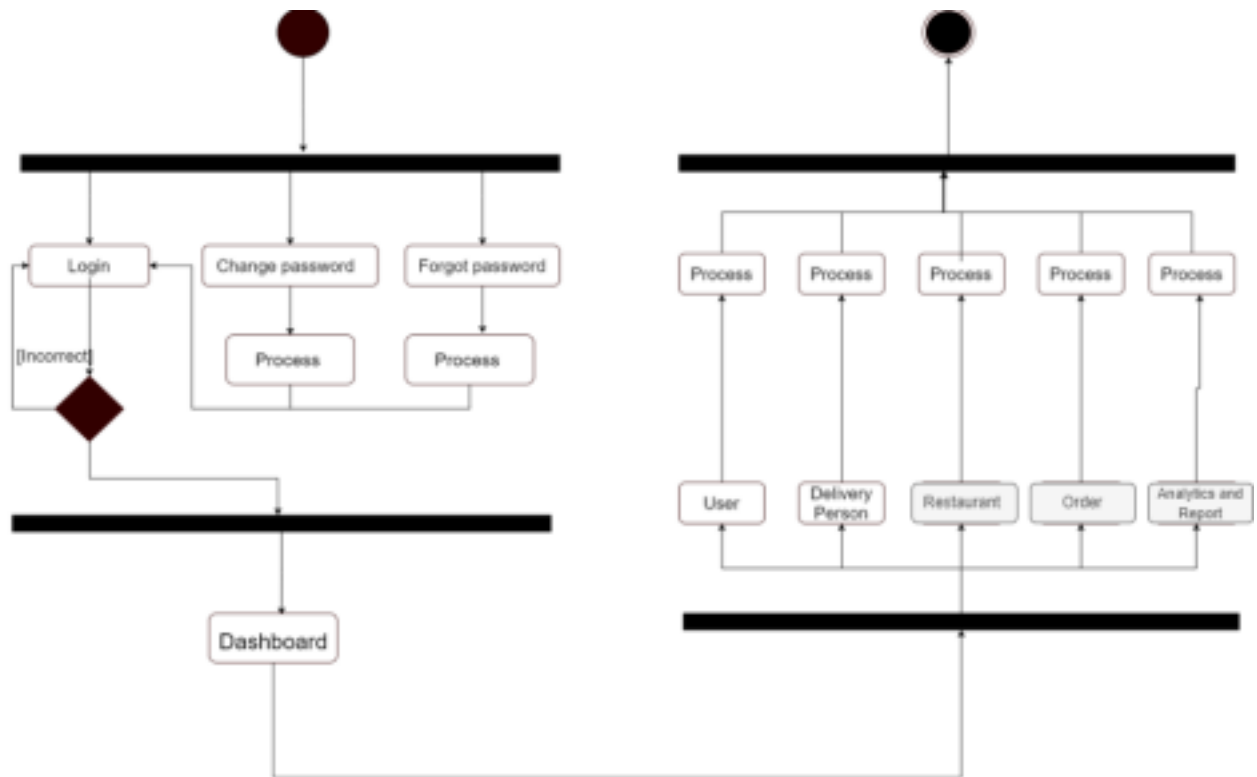


## Use Case Diagram



# Activity Diagram

Admin

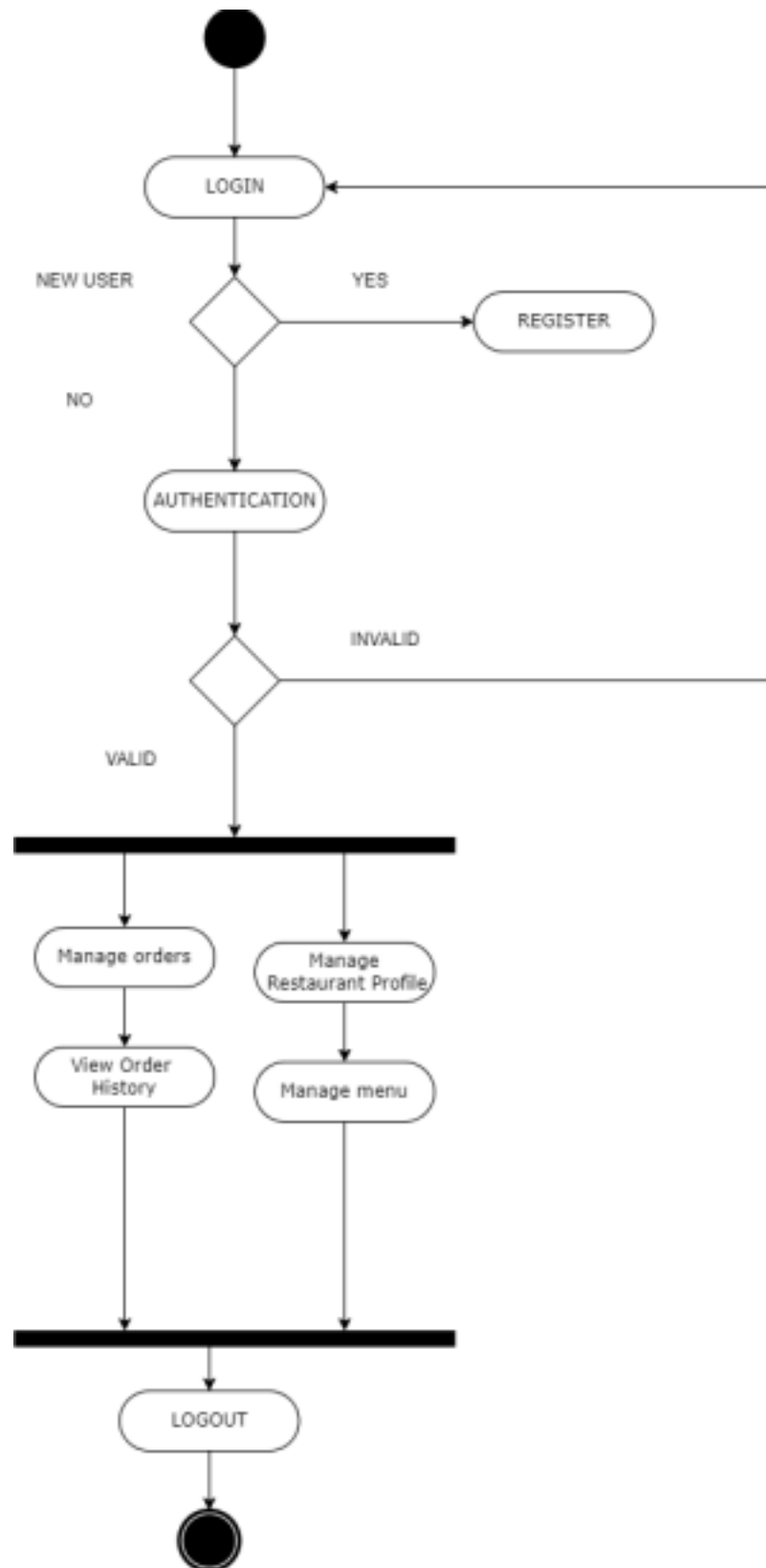


## Customer

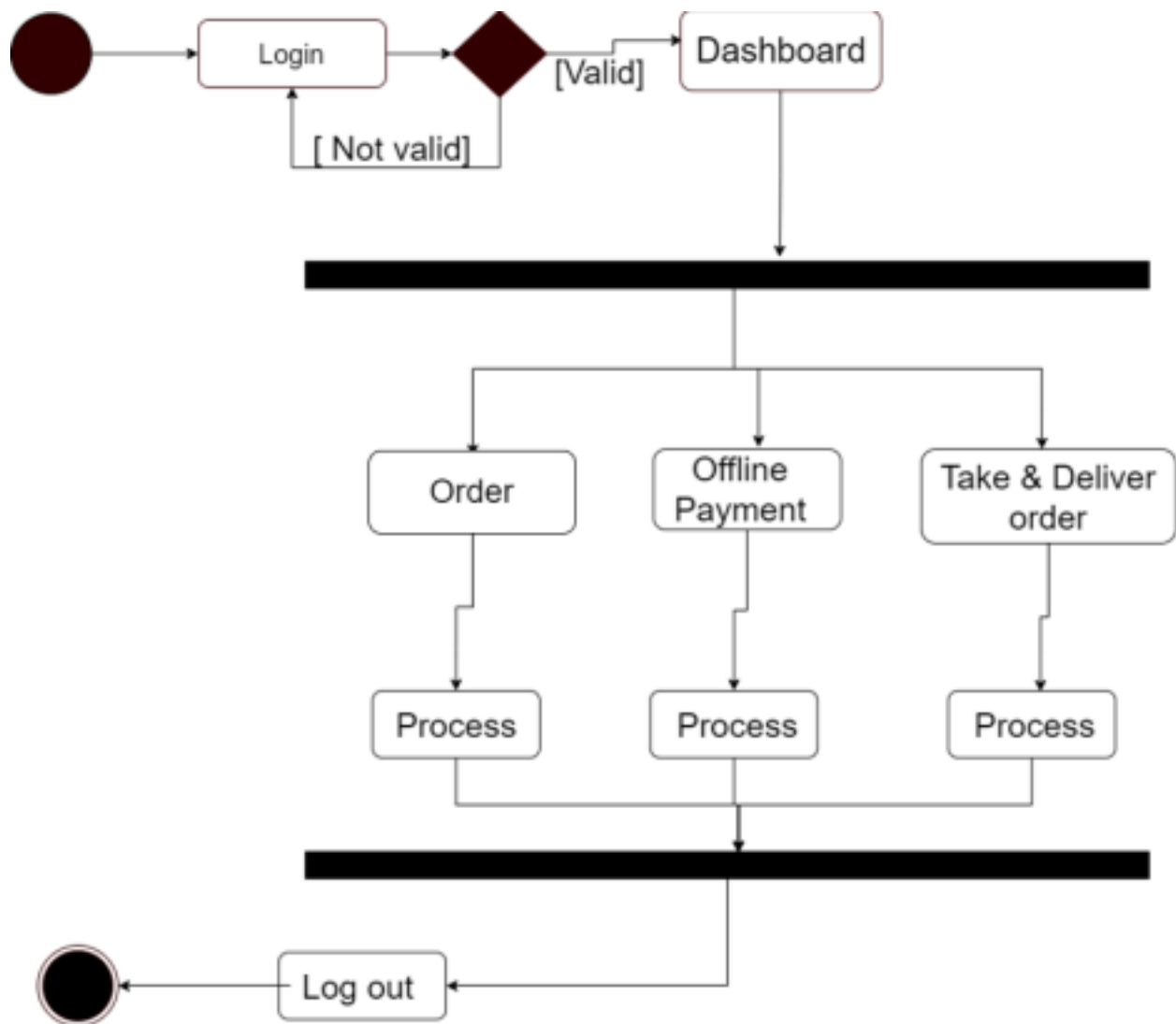




## Restaurant Owner

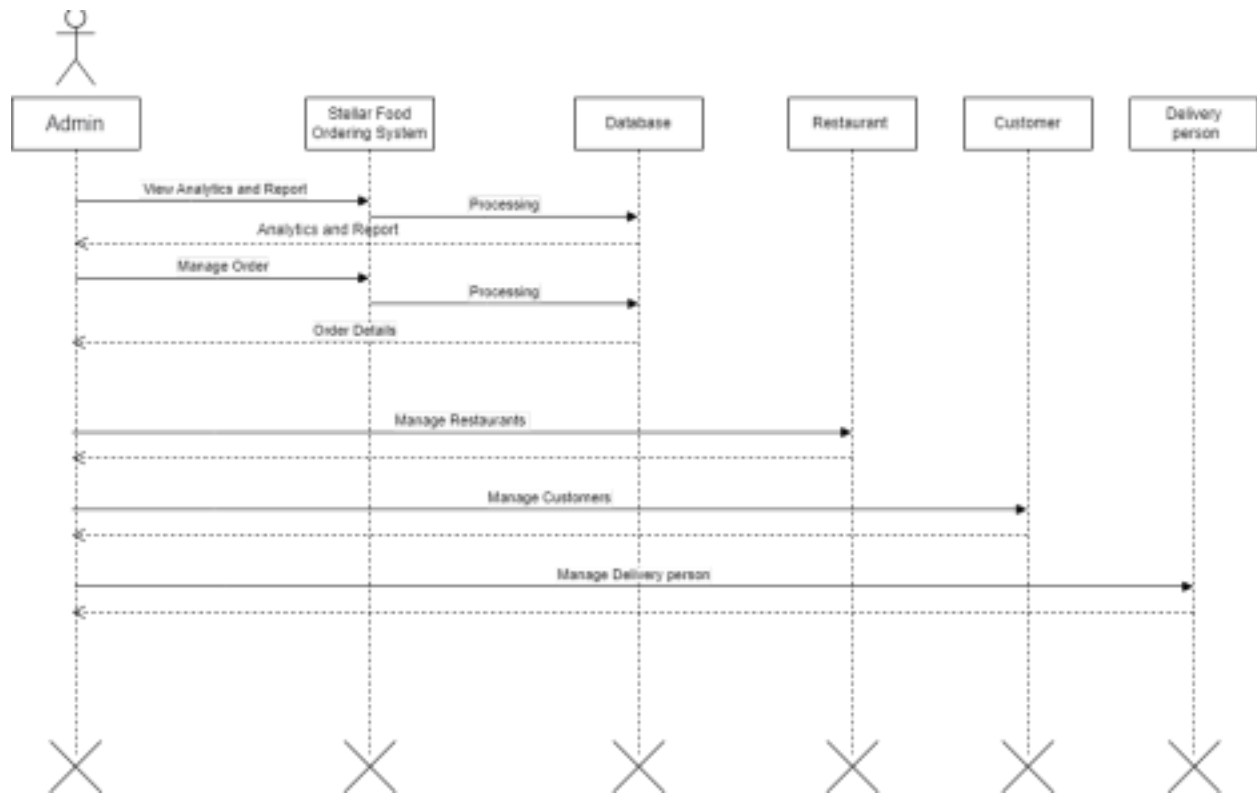


## Delivery Person

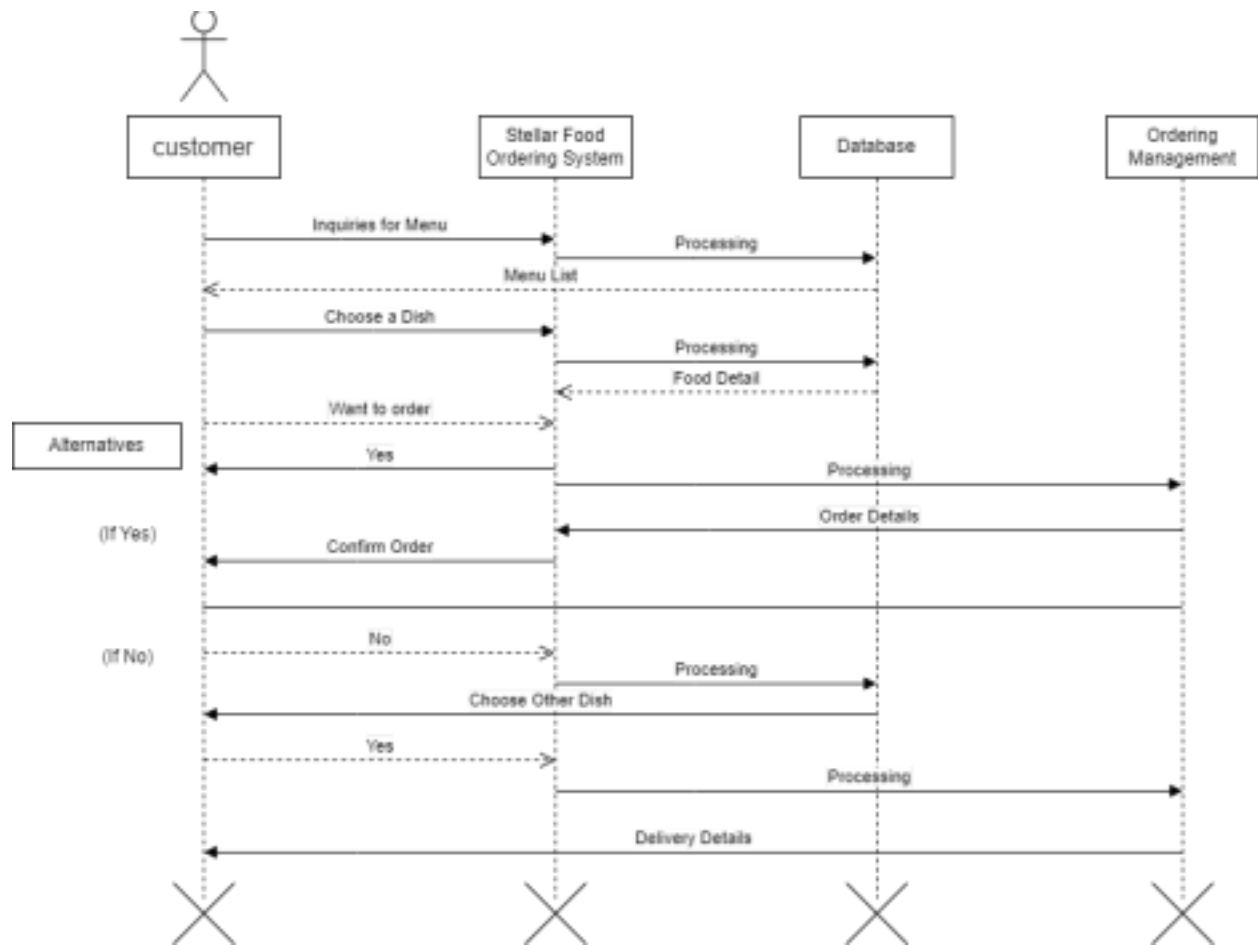


# Sequence Diagram

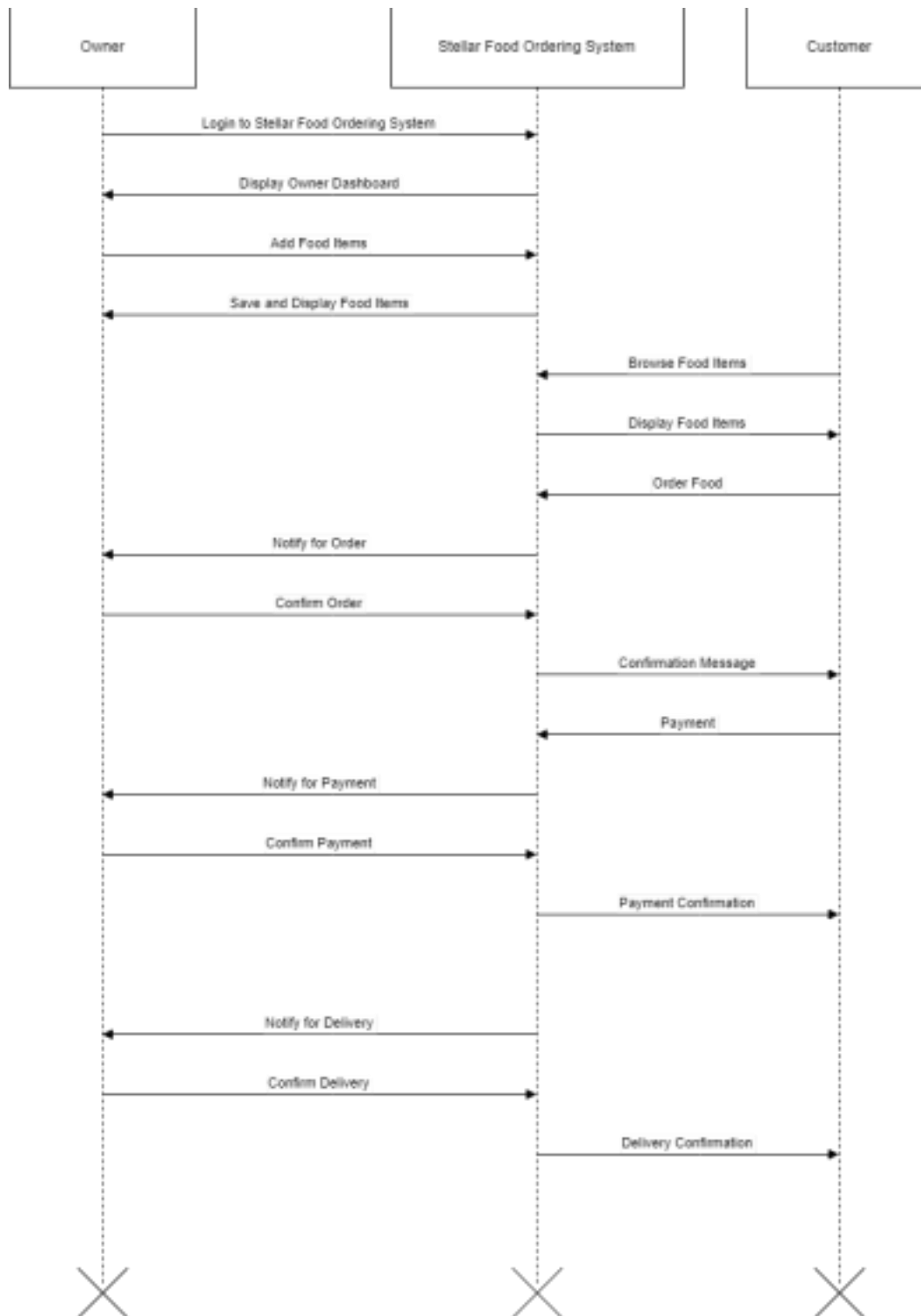
## Admin



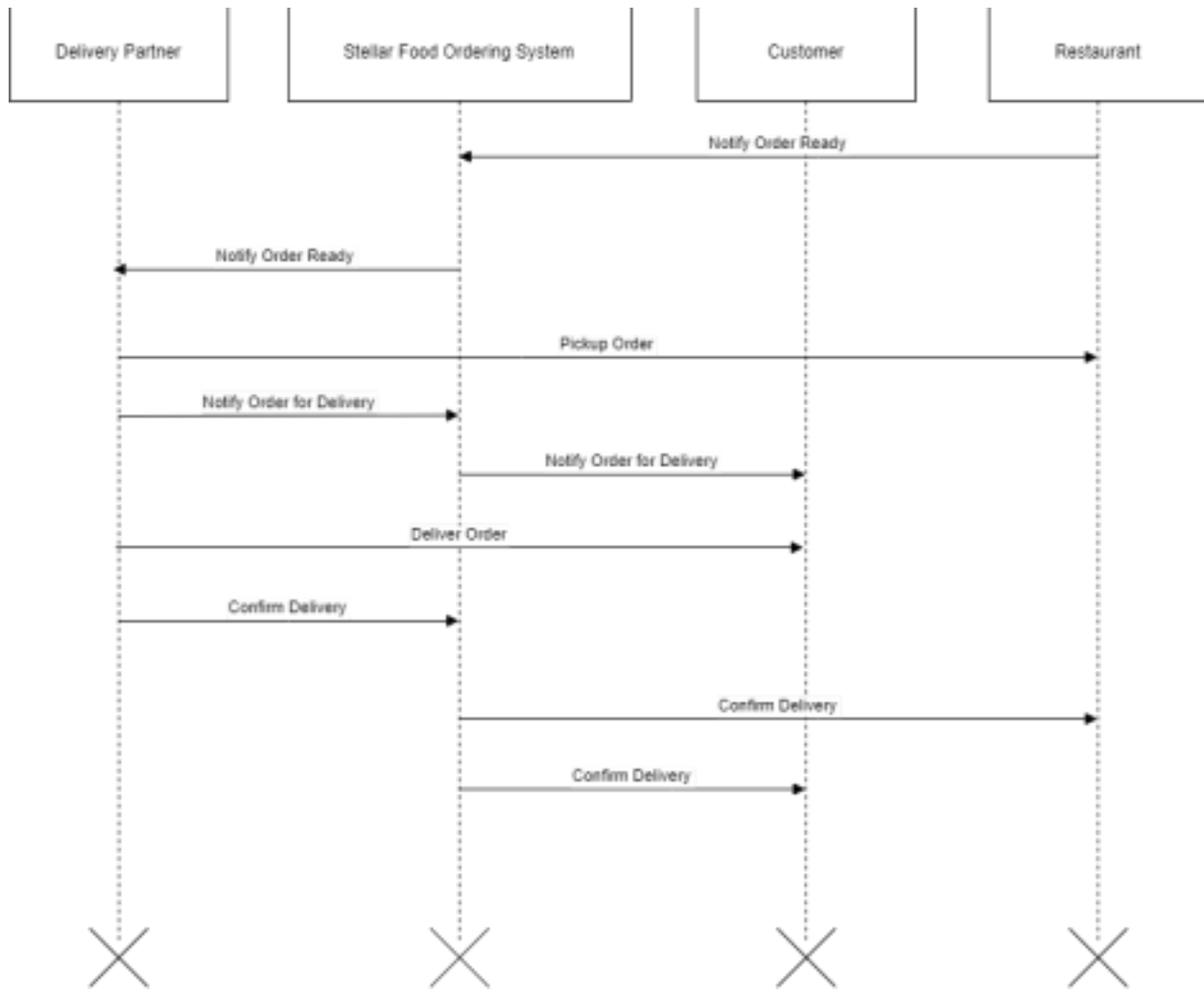
## Customer



## Restaurant Owner

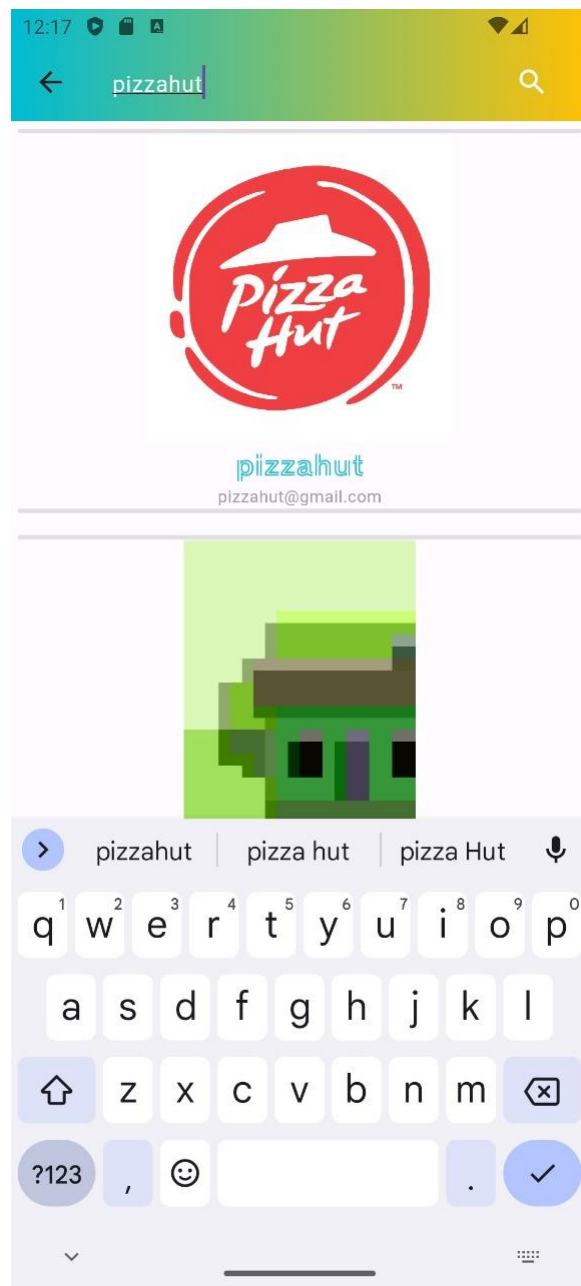


## Delivery Person



# Screenshots

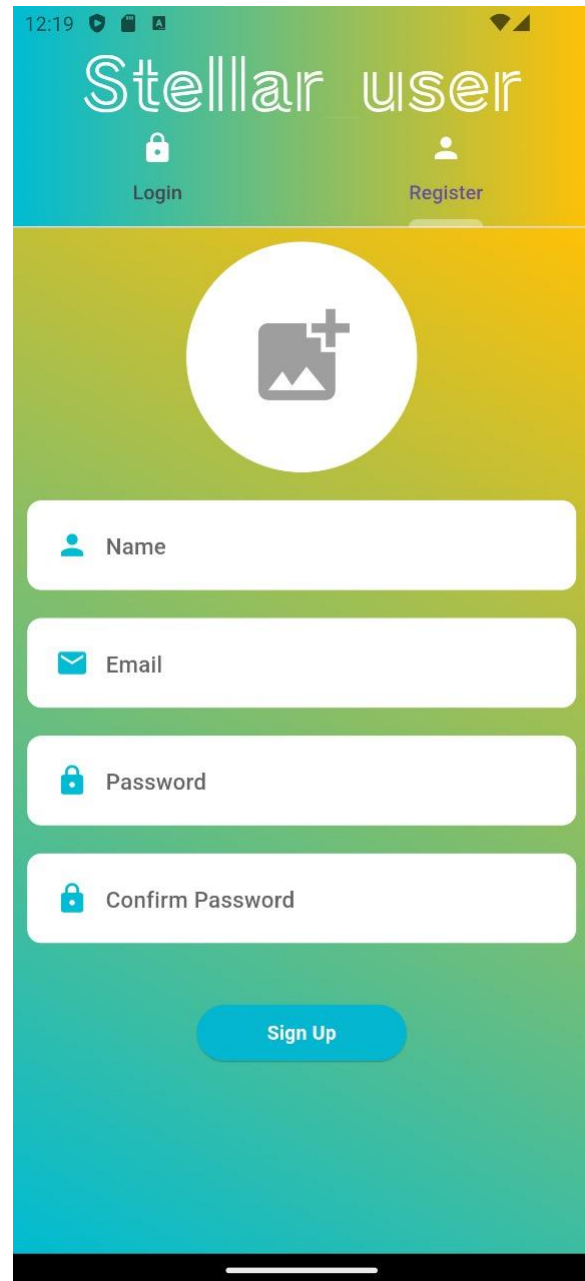
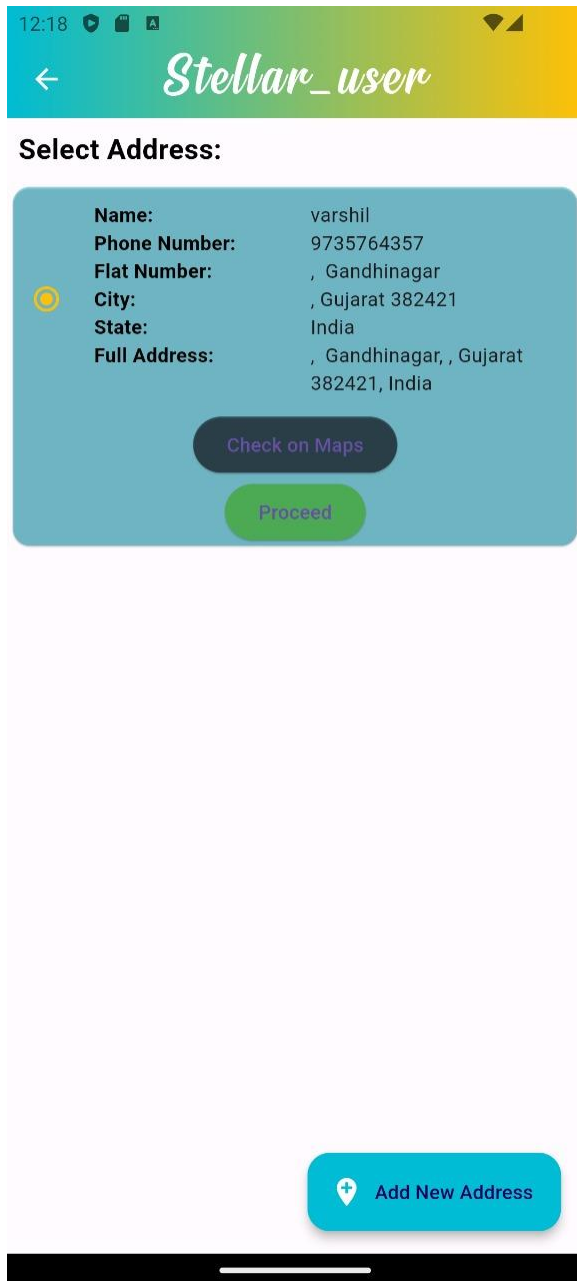
## User

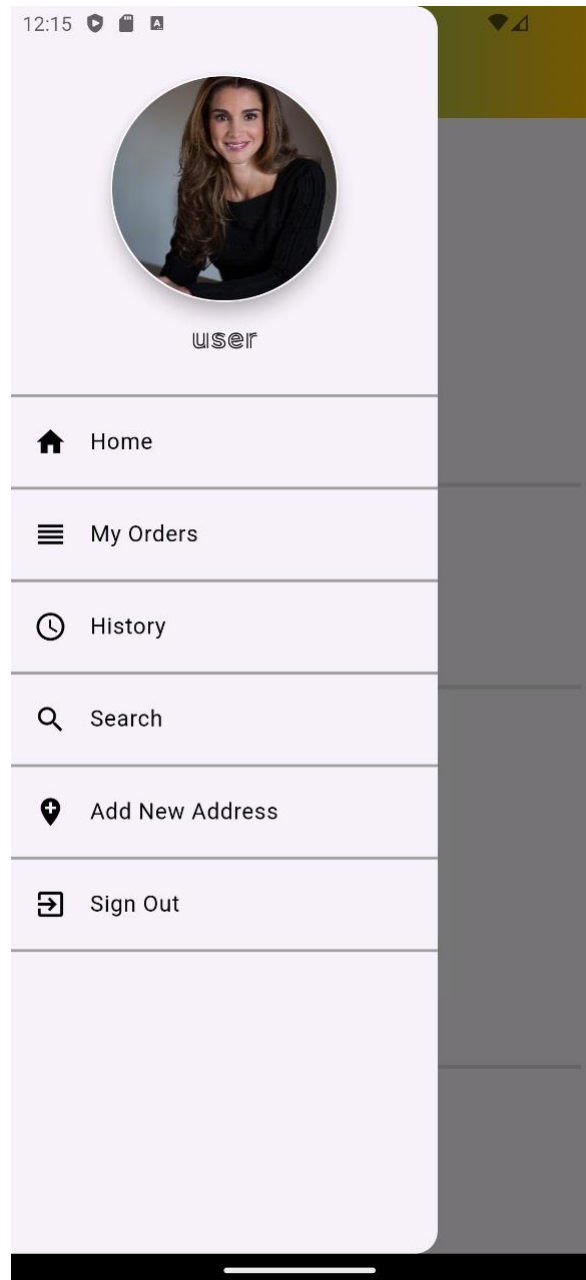
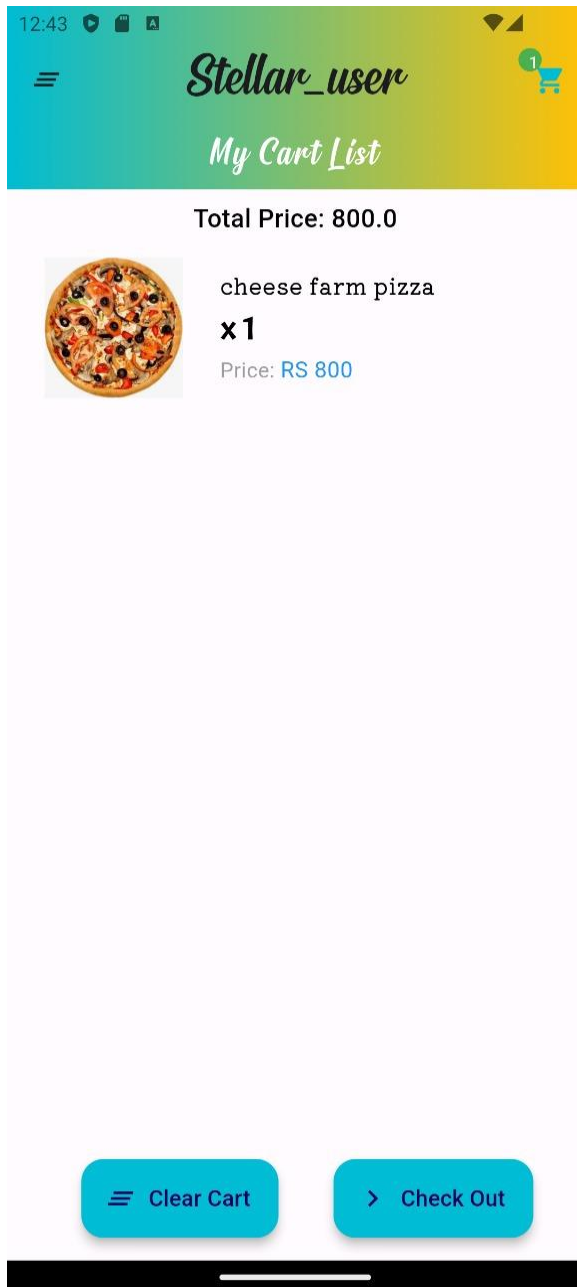


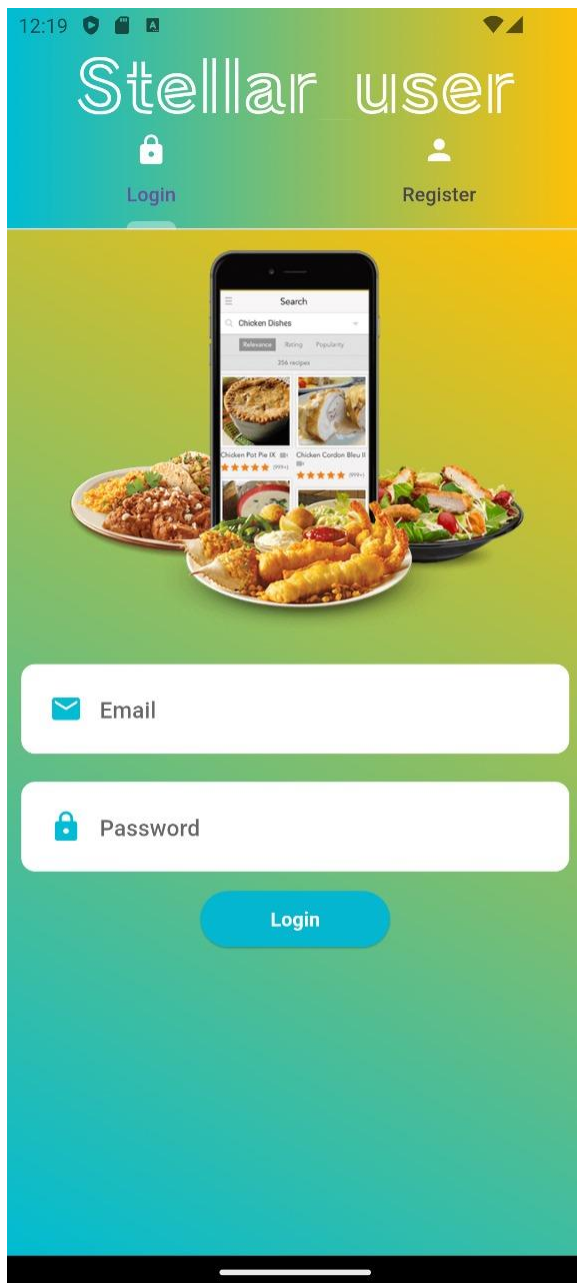




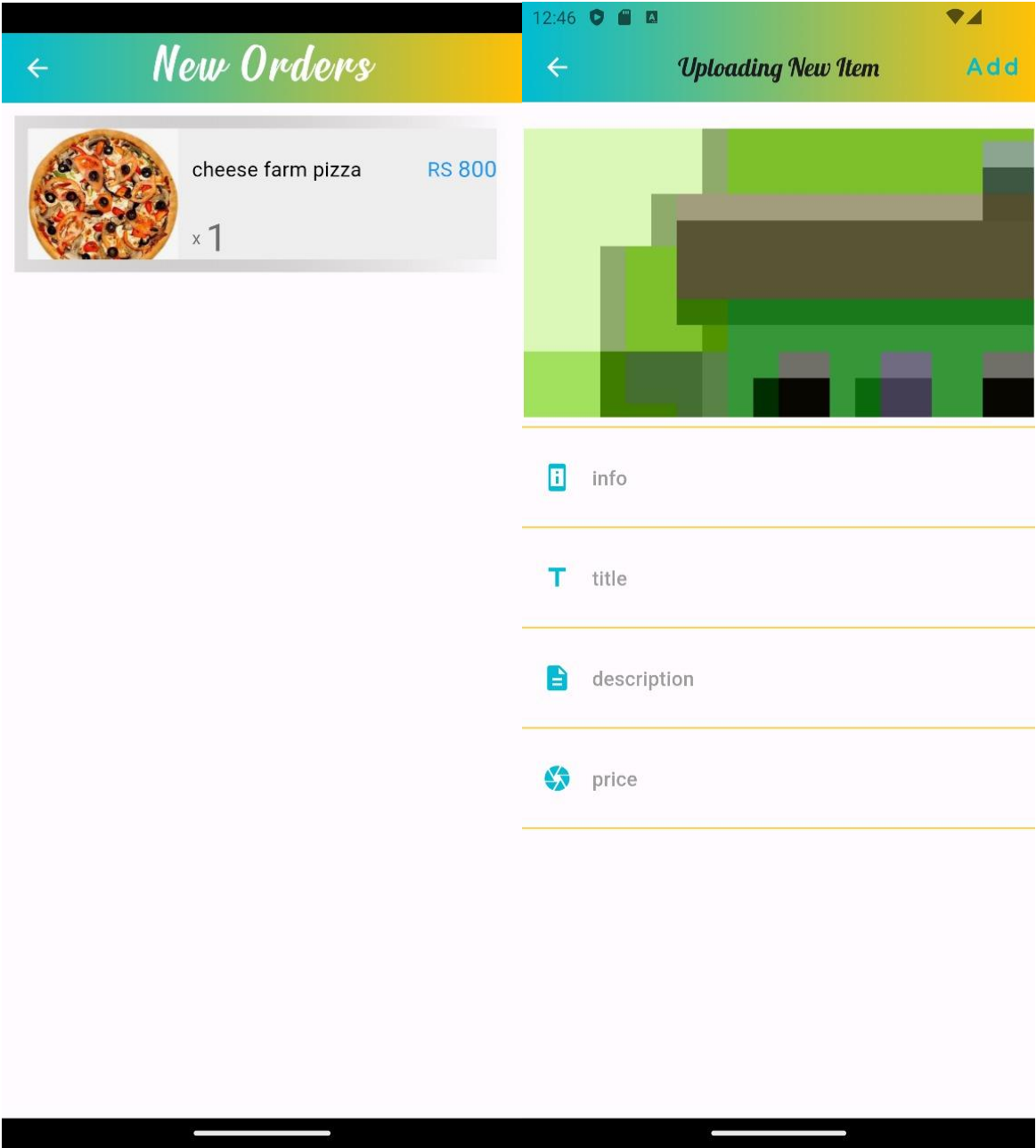




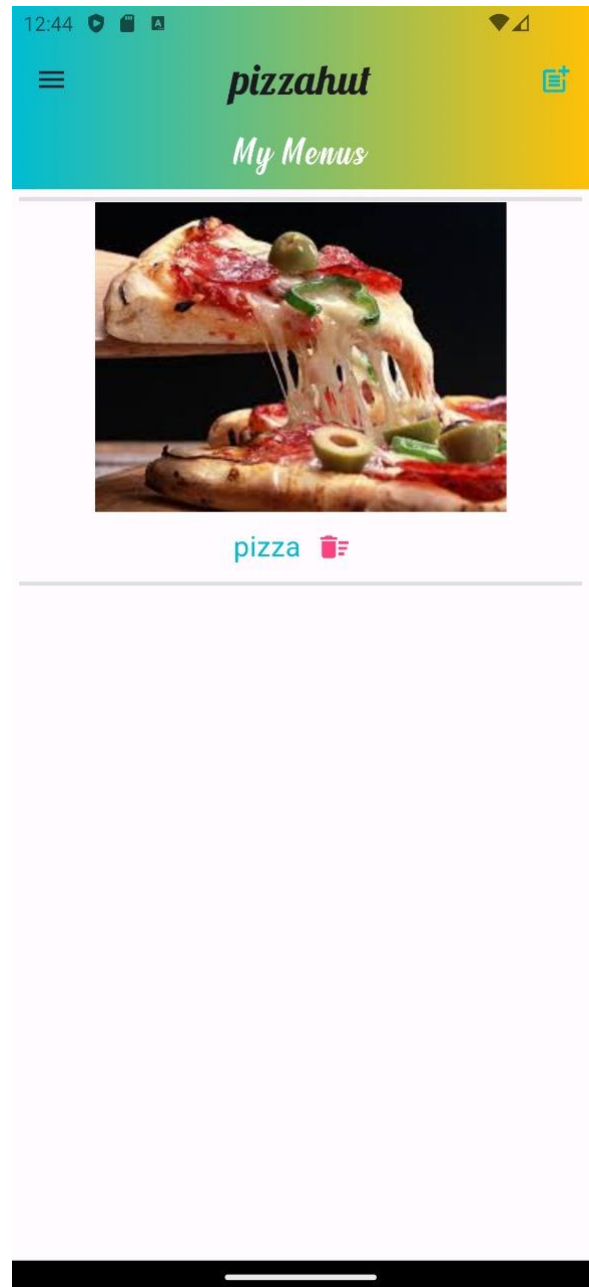
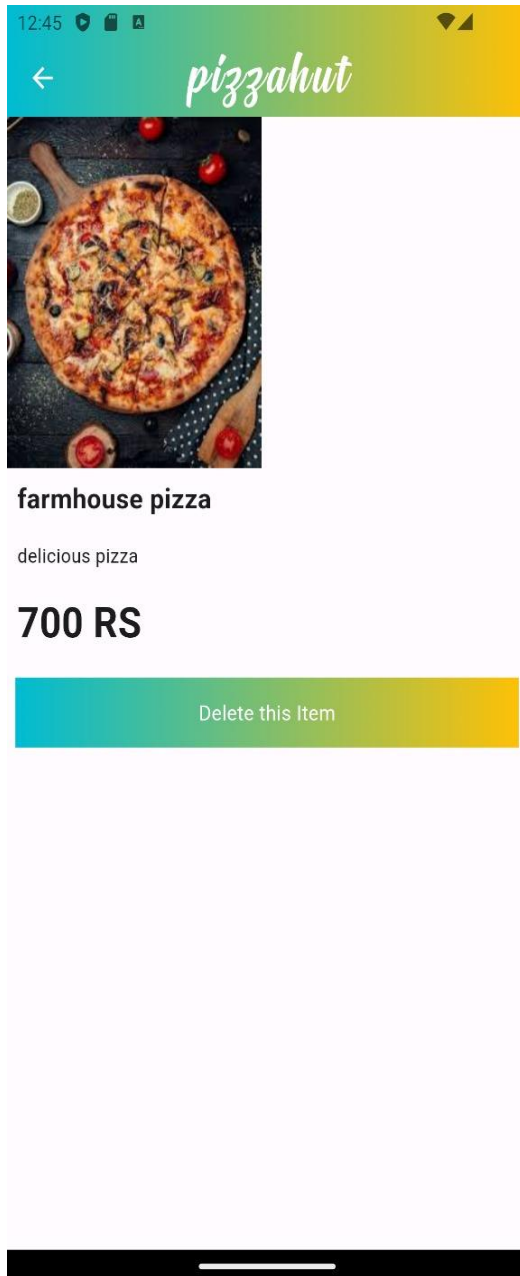


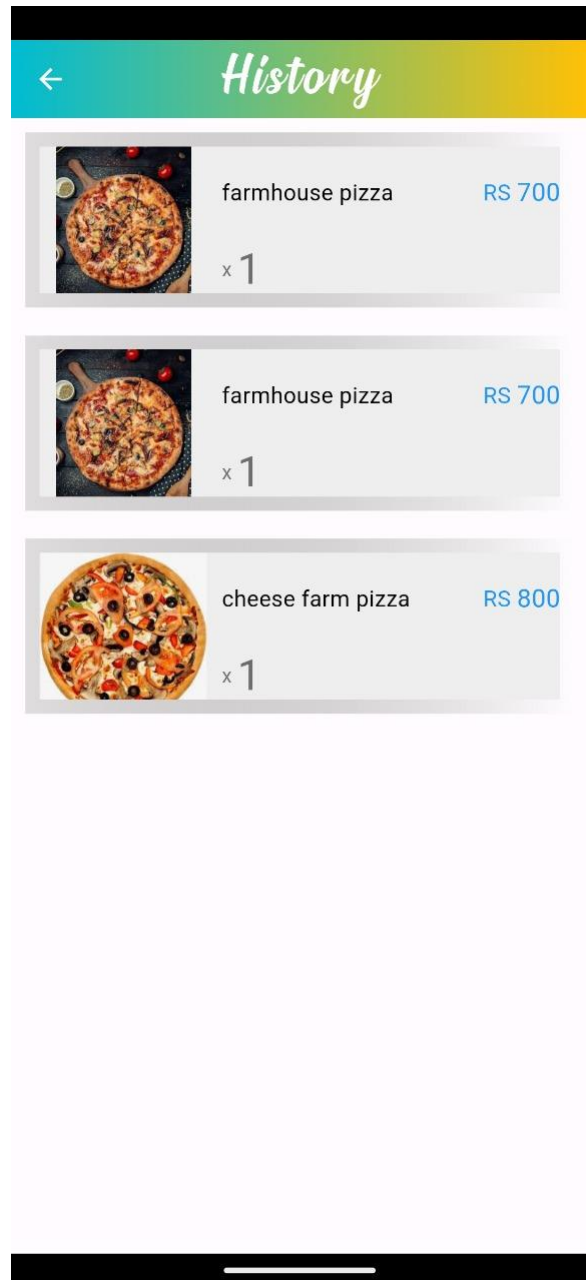
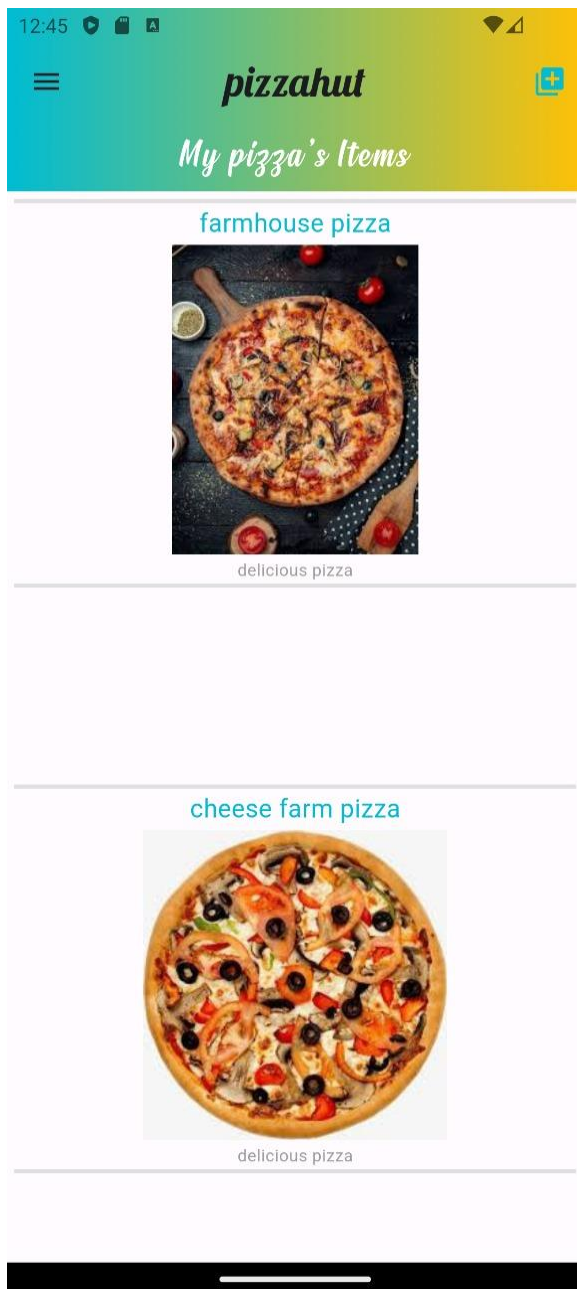


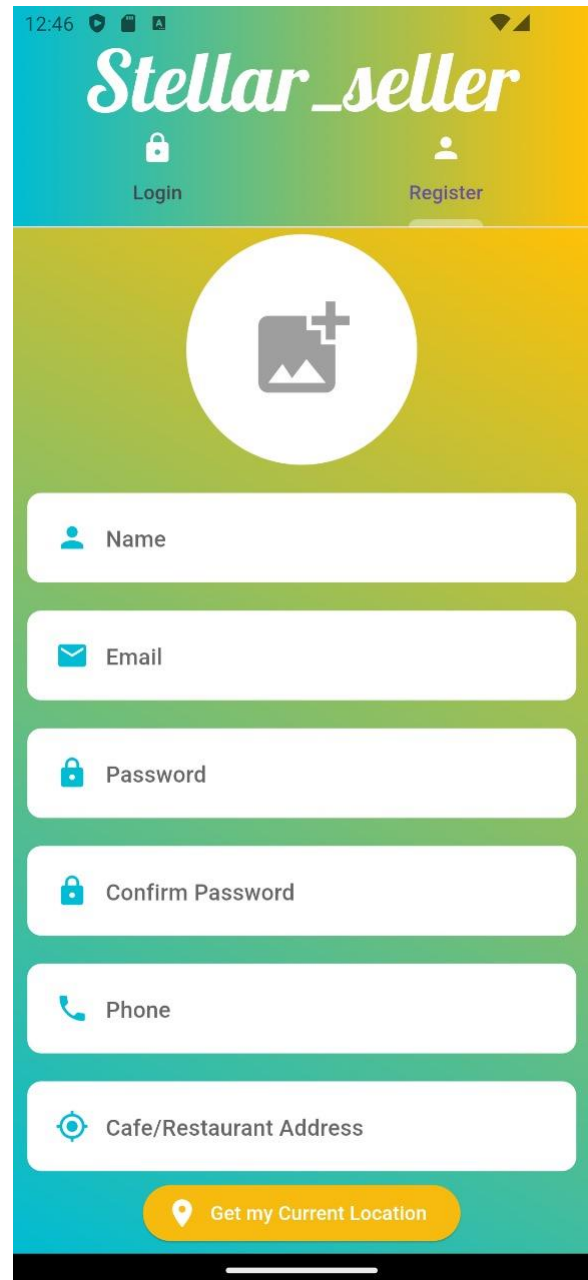
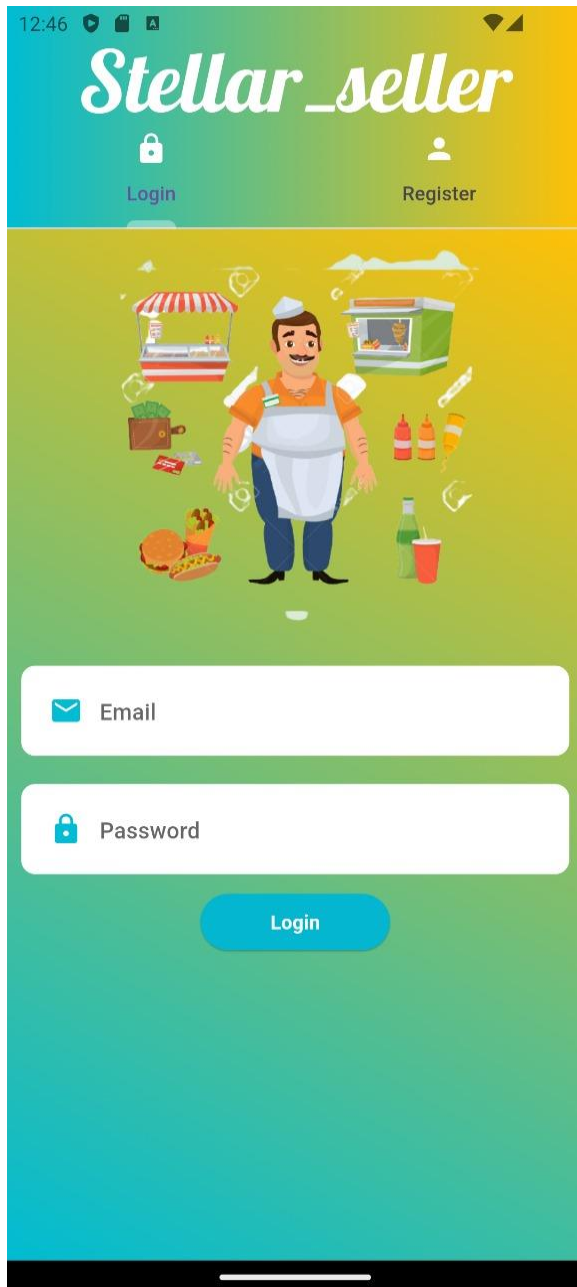
Seller



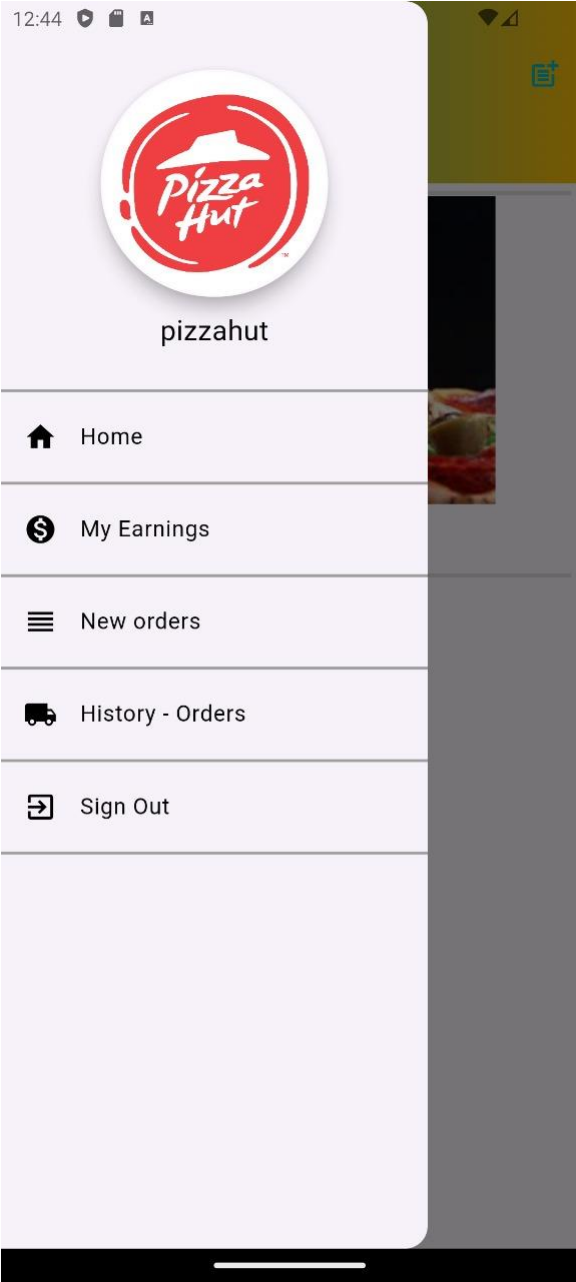
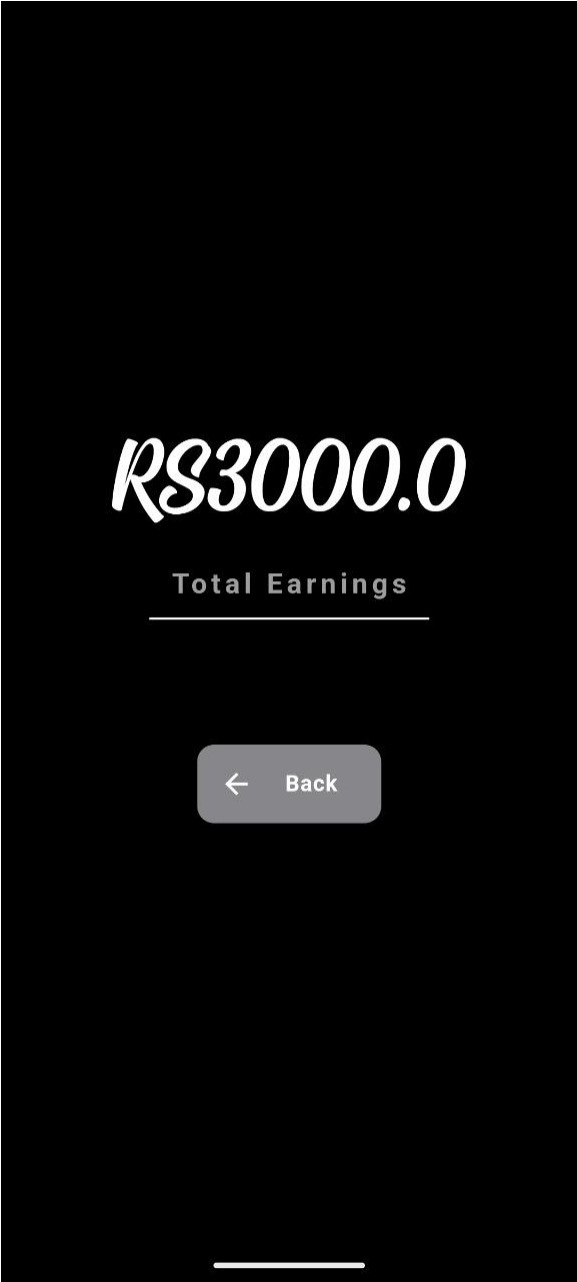
S



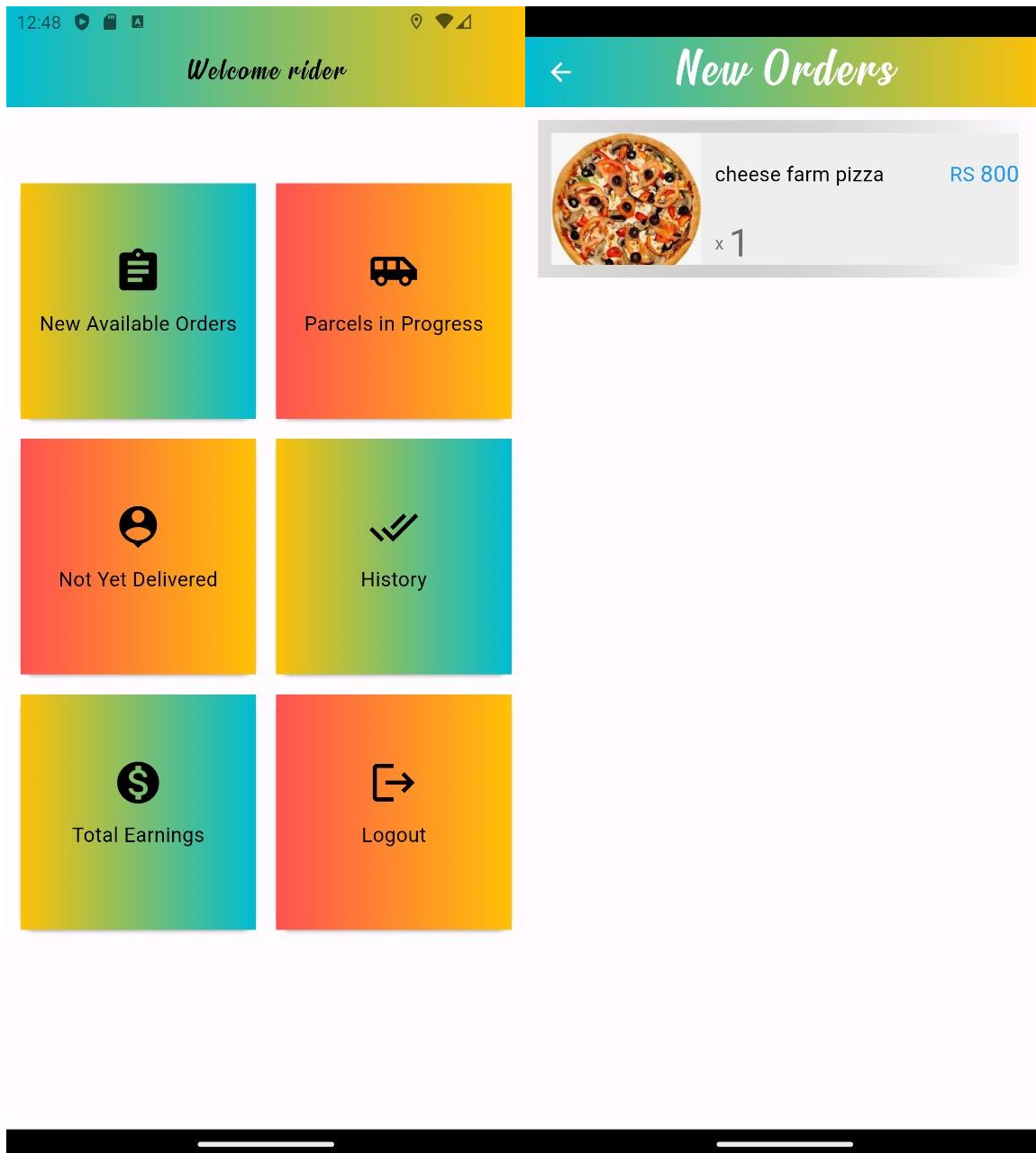


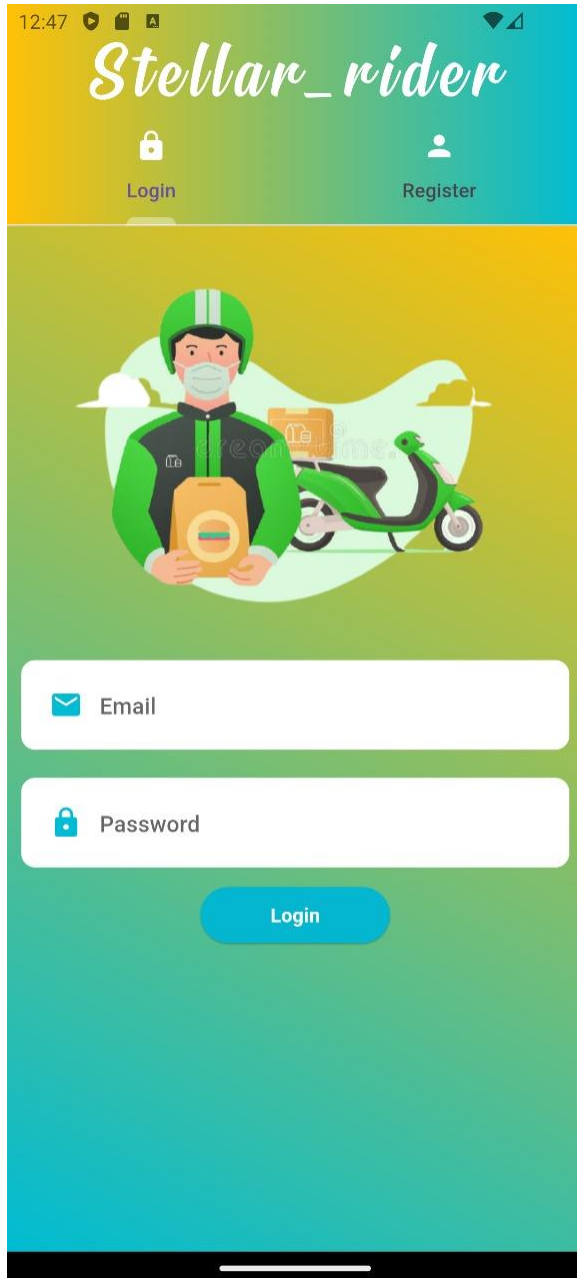






# RIDER








12:47


# Stellar\_rider


Login Register





 Name


 Email

 Password

 Confirm Password


 Phone

 My Current Address

 Get my Current Location

*RS 20.0*

Total Earnings

 Back

# ADMIN

