



**Bilkent University**

**Department of Computer Engineering**

---

# **CS 491 - Senior Design Project I**

**Foodie**

## **Analysis Report**

Utku Gökçen 21703746

İlhan Koç 21603429

Emre Erciyas 21600991

Yiğit Erkal 21601521

**Supervisor:** Fazlı Can

**Jury Members:** Shervin Arashloo and Hamdi Dibeklioglu

**Innovation Expert:** Murat Ergun

November 15, 2021

<b>1. Introduction</b>	<b>2</b>
<b>2. Current System</b>	<b>3</b>
<b>3. Proposed System</b>	<b>4</b>
3.1 Overview	4
3.2 Functional Requirements	4
3.3 Nonfunctional Requirements	5
3.3.1 Security	5
3.3.2 Usability	5
3.3.3 Accessibility	5
3.3.4 Scalability	5
3.3.5 Performance	6
3.3.6 Extendibility	6
3.4 Pseudo Requirements	6
3.5 System Models	7
3.5.1 Scenarios	7
3.5.2 Use Case Model	16
3.5.3 Object-Class Model	17
3.5.4 Dynamic Models	18
3.5.4.1 Sequence Diagrams	18
3.5.4.2 Activity Diagrams	21
3.5.4.3 State Diagrams	23
3.5.5 User Interface - Navigational Paths and Screen Mock-ups	25
3.5.5.1 Navigational Path	25
3.5.5.2 User Interface	27
<b>4. Other Analysis Elements</b>	<b>52</b>
4.1. Consideration of Various Factors in Engineering Design	52
4.2. Risks and Alternatives	54
4.3. Project Plan	55
4.4. Ensuring Proper Teamwork	57
4.5. Ethics and Professional Responsibilities	58
4.6. Planning for New Knowledge and Learning Strategies	58
<b>5. References</b>	<b>59</b>

## 1. Introduction

Food waste has become a major problem globally in recent years. According to the research conducted by the United Nations Environment Program, more than 7.7 million tons of food is wasted in Turkey every year. In addition, according to the 2021 United Nations Food Waste Index Report, 93 kilograms of food per person is thrown away every year in Turkey. This means that 17 percent of ready-to-eat food in retail outlets, homes and restaurants globally goes straight to waste[1].

Before researching the solutions to this problem, it is necessary to talk about the main causes of the problem. The main reasons for food waste in mass consumption areas are the inability of restaurants, hotels or cafeterias to adjust the size of the meals offered to their customers, buying more food than they need, not having enough storage space for the food they have and inability to track the expiration dates of the food. Restaurants, hotels or cafeterias should avoid these situations that cause food waste and the leftover food should be re-evaluated in the restaurants and hotels, if it cannot be re-evaluated, it should be donated to the people, if it cannot be donated, it should be considered as animal feed, if it cannot be used as animal feed, it should be directed to composting and ultimately sent to landfill as a last resort[2].

The purpose of our senior design project is to create a platform where restaurants, hotels or cafeterias can sell leftover food at lower prices to people or animal farms when they cannot re-evaluate the leftover food. In this way, leftover meals will not be wasted and both customers and businesses will be profitable by adopting a win-win approach.

With this analysis report, we aimed to explain the description of our project, its requirements, system models, screen mockups and other analysis elements.

## 2. Current System

### Too Good To Go

Too Good To Go is a mobile application used especially in America and Europe, that connects customers to restaurants and stores that have unsold food surplus. This application provides users to a service in which users can buy cheap meals from restaurants and markets close to their location [3].

### Food For All

Food For All is an application for mobile platforms to decrease the rate of the wasted foods. This project started in 2016 as a Kickstarter campaign. This application has three main goals. One of them was helping everyone to get high quality meals. Another one is reducing the rate of food waste and their last goal is making food prices more optimized [4].

Functionality	Foodie	Too Good To Go	Food For All
Location Based Service	+	+	+
Business Review	+	+	-
Pay Through Application	+	+	+
Notification System According to Customer Location	+	-	-
Discount Rate According to Customer Location	+	-	-

### **3. Proposed System**

#### **3.1 Overview**

Foodie is a mobile application where people can buy leftover food from restaurants or cafes at a lower price. There are two different types of users in the application. The first is restaurants and cafes which sell leftover food and the second is customers who want to buy leftover food. Customers can list restaurants close to their location that sell leftover food and also view those restaurants on a map. Customers can add the meals they want to buy to their cart, pay through the application and receive the food from the restaurant. In addition, customers can view their pending purchases and past purchases. In this way, they can control their spending and see how much money they have saved. Besides, customers will be able to add their favorite restaurants to their "Favorites" list. This way, they will receive notifications when their favorite restaurants list discounted meals.

Restaurants list the leftover food in the application by uploading a photo of the food, selecting the recommended expiry date and writing the name and description. In addition, restaurants will be able to see pending orders and previously delivered leftover food. In this way, they will be able to see how much food and money they have saved.

#### **3.2 Functional Requirements**

- The application will require a signing up procedure for the business owners, and customers.
- Customers can list restaurants with deals near their location or sell discounted food on the map.
- Restaurants will list their discounted foods.
- Restaurants can see orders placed by customers.
- It will be possible for customers to chat with the restaurants(and vice versa) when they place an order.
- Customers can display the total savings.
- Customers can use different payment methods.
- Customers will be able to evaluate restaurants.
- Customers will be able to add their favorite restaurants to the "Favorites" list.
- Restaurants choose the time interval in which they sell the food.

- Business owners choose which payment methods are available in their restaurants/cafes. (online payment via Credit card or cash payment).
- view their previous sales so they can see how much food and money they have saved.
- Users can edit their profile information.

### **3.3 Nonfunctional Requirements**

#### **3.3.1 Security**

- Authentication system will be used to login to the application. Therefore, users' email, phone numbers and passwords will not be visible to other users.
- Users' passwords will be stored in Firebase in an encrypted format.
- User's card information will not be stored after in-app payment.

#### **3.3.2 Usability**

- After opening the application for the first time, a user manual will be displayed explaining how to use the application.
- Foodie will provide its users a simple and convenient interface that they can use easily.
- Foodie will be developed to run on different phone and tablet resolutions.

#### **3.3.3 Accessibility**

- Foodie should be available for all people who have an android device and are located in Turkey.
- Foodie can be downloaded for free from the Google Play Store.

#### **3.3.4 Scalability**

- According to the free version of Firebase, 10.000 users can sign up for the app. If the number of users exceeds 10.000, the paid version of Firebase is available.

### **3.3.5 Performance**

- Foodie will be implemented using the most trendy and efficient layouts and views to increase the performance of the application.

### **3.3.6 Extendibility**

- Foodie's layered architecture will make it easier to implement new features and improve the interface.
- In order for Foodie to be a maintainable application, a procedural way should be followed when naming variables, classes, layout files, activities and layers.

## **3.4 Pseudo Requirements**

- Foodie must run on android devices.
- Foodie should use Firebase for storage.
- Foodie should operate with every existing android mobile system written at least 5 years before.
- Java programming language will be used to write Foodie.
- Application's server side will use REST API services.
- For version control purposes, Git will be used and it will be hosted on Github.
- For project development control purposes, Trello will be used.

### 3.5 System Models

#### 3.5.1 Scenarios

##### Scenario 1

<b>Use Case Name</b>	Sign up for the application
<b>Participating Actors</b>	Customer user
<b>Entry Conditions</b>	<ul style="list-style-type: none"><li>• Application must be open.</li><li>• Users should have a valid phone number and mail address.</li></ul>
<b>Exit Conditions</b>	<ul style="list-style-type: none"><li>• User submit the sign-up form</li></ul>
<b>Flow of Events</b>	<ol style="list-style-type: none"><li>1. User opens the application</li><li>2. The user chooses whether s/he is a business or a customer.</li><li>3. User clicks on the sign-up button.</li><li>4. User fills the registration form.</li><li>5. The form gets validated.</li><li>6. User signs up to the application.</li></ol>



## Scenario 2

<b>Use Case Name</b>	Register the business to the application
<b>Participating Actors</b>	Business owner
<b>Entry Conditions</b>	<ul style="list-style-type: none"><li>• Application must be open.</li><li>• Users should have a valid phone number and email address.</li></ul>
<b>Exit Conditions</b>	<ul style="list-style-type: none"><li>• User submit the registration form</li></ul>
<b>Flow of Events</b>	<ol style="list-style-type: none"><li>1. User opens the application</li><li>2. The user chooses whether s/he is a business or a customer.</li><li>3. User clicks on the “register new business” button.</li><li>4. User fills the registration form.</li><li>5. The form gets validated.</li><li>6. User’s business signs up to the application.</li></ol>

### Scenario 3

<b>Use Case Name</b>	Sign in to the application
<b>Participating Actors</b>	Customer user, Business Owner
<b>Entry Conditions</b>	<ul style="list-style-type: none"><li>• Application must be open.</li><li>• Users should have a valid phone number and password.</li></ul>
<b>Exit Conditions</b>	<ul style="list-style-type: none"><li>• User submit the sign-in form</li></ul>
<b>Flow of Events</b>	<ol style="list-style-type: none"><li>1. User signs up for the application</li><li>2. User fills the sign-in form.</li><li>3. The form gets validated.</li><li>4. User signs in to the application.</li></ol>
<b>Alternative Flow of Events</b>	<ol style="list-style-type: none"><li>1. User clicks the "Sign in with Google" button.</li><li>2. User verifies his/her Google account.</li><li>3. User signs in to the application.</li></ol>

#### Scenario 4

<b>Use Case Name</b>	Add Meal to Cart
<b>Participating Actors</b>	Customer user
<b>Entry Conditions</b>	<ul style="list-style-type: none"><li>• User must be signed in to the application</li></ul>
<b>Exit Conditions</b>	<ul style="list-style-type: none"><li>• User clicks the "order" button.</li></ul>
<b>Flow of Events</b>	<ol style="list-style-type: none"><li>1. User chooses one of the restaurants in the map interface or listed on the home page.</li><li>2. User displays the meals that the restaurant sells at a discount.</li><li>3. User selects the food they want to buy and adds it to their cart.</li><li>4. After adding the product to the cart, the user goes to the cart and presses the "confirm cart" button and is directed to the page where the payment method will be selected.</li><li>5. User makes the payment through the application and clicks the "order" button.</li></ol>
<b>Alternative Flow of Events</b>	<ol style="list-style-type: none"><li>5. User prefers to pay at the venue and clicks the "order" button.</li></ol>

## Scenario 5

<b>Use Case Name</b>	Add a business to your favorites
<b>Participating Actors</b>	Customer user
<b>Entry Conditions</b>	<ul style="list-style-type: none"><li>• User must be signed in to the application, and businesses should be listed</li></ul>
<b>Exit Conditions</b>	<ul style="list-style-type: none"><li>• User clicks the heart icon on the business's page.</li></ul>
<b>Flow of Events</b>	<ol style="list-style-type: none"><li>1. The user clicks on the heart icon next to the businesses they display on the homepage and adds the business to their favourites.</li></ol>
<b>Alternative Flow of Events</b>	<ol style="list-style-type: none"><li>1. User clicks the business s/he wants to add to his/her favourites from the map interface or the home page and displays the business' page.</li><li>2. User clicks the heart icon on the business page and adds the business to their favourites.</li></ol>

## Scenario 6

<b>Use Case Name</b>	Edit profile
<b>Participating Actors</b>	Customer user and business owner
<b>Entry Conditions</b>	<ul style="list-style-type: none"><li>• User must be signed in to the application</li></ul>
<b>Exit Conditions</b>	<ul style="list-style-type: none"><li>• User clicks the "save" button after updating their profile information.</li></ul>
<b>Flow of Events</b>	<ol style="list-style-type: none"><li>1. User clicks on the "profile" icon in the bottom navigation bar.</li><li>2. The user clicks the "edit my profile" button.</li><li>3. User edits his/her profile information.</li><li>4. User presses the "save" button and updates their profile.</li></ol>

## Scenario 7

<b>Use Case Name</b>	Display completed and pending orders
<b>Participating Actors</b>	Customer user
<b>Entry Conditions</b>	<ul style="list-style-type: none"><li>• User must be signed in to the application</li></ul>
<b>Exit Conditions</b>	<ul style="list-style-type: none"><li>• User displays their completed and pending orders.</li></ul>
<b>Flow of Events</b>	<ol style="list-style-type: none"><li>1. User clicks on the "My orders" button on the bottom navigation bar.</li><li>2. User can see their pending and completed orders by clicking the "Pending" and "Completed" buttons on the "My orders" page.</li></ol>

### Scenario 8

<b>Use Case Name</b>	Display total savings
<b>Participating Actors</b>	Customer user
<b>Entry Conditions</b>	<ul style="list-style-type: none"><li>• User must be signed in to the application</li></ul>
<b>Exit Conditions</b>	<ul style="list-style-type: none"><li>• User displays their total savings.</li></ul>
<b>Flow of Events</b>	<ol style="list-style-type: none"><li>1. User clicks on the "profile" icon in the bottom navigation bar.</li><li>2. User sees how much money and food they have saved in the profile tab.</li></ol>

### Scenario 9

<b>Use Case Name</b>	Communicate
<b>Participating Actors</b>	Customer user and business owner
<b>Entry Conditions</b>	<ul style="list-style-type: none"><li>• Users must be signed in to the application</li><li>• Customers should have made a purchase from the business.</li></ul>
<b>Exit Conditions</b>	<ul style="list-style-type: none"><li>• Conversation is terminated by the customer or business owner.</li></ul>
<b>Flow of Events</b>	<ol style="list-style-type: none"><li>1. Customer user clicks the "my orders" button on the bottom navigation bar.</li><li>2. Customer user clicks on the "pending orders" tab.</li><li>3. Customer user clicks on one of the pending orders and reaches the order</li></ol>

	<p>detail page.</p> <p>4. Customer user starts the communication by pressing the "communicate" button on the order detail page.</p>
--	---

### Scenario 10

<b>Use Case Name</b>	Set Preferences
<b>Participating Actors</b>	Business owner
<b>Entry Conditions</b>	<ul style="list-style-type: none"> <li>User must be signed in to the application</li> </ul>
<b>Exit Conditions</b>	<ul style="list-style-type: none"> <li>Business owner clicks the "set" button.</li> </ul>
<b>Flow of Events</b>	<ol style="list-style-type: none"> <li>Business owner clicks “My Business” button on the bottom navigation bar</li> <li>The business owner clicks on the "Business Preferences" button.</li> <li>The business owner can view and change the pick-up time of the meals, business information and address on this page.</li> </ol>

### Scenario 11

<b>Use Case Name</b>	Add New Meal
<b>Participating Actors</b>	Business owner
<b>Entry Conditions</b>	<ul style="list-style-type: none"> <li>User must be signed in to the application</li> </ul>

<b>Exit Conditions</b>	<ul style="list-style-type: none"> <li>Business owner clicks the "add" button.</li> </ul>
<b>Flow of Events</b>	<ol style="list-style-type: none"> <li>Business owner clicks the “Add New Meal” button on the bottom navigation bar.</li> <li>Business owner enters the information about the food.</li> <li>Business owner clicks the “Add” button.</li> </ol>

## Scenario 12

<b>Use Case Name</b>	Make Payment
<b>Participating Actors</b>	Customer User
<b>Entry Conditions</b>	<ul style="list-style-type: none"> <li>User must be signed in to the application</li> <li>User must select the food they want to buy.</li> </ul>
<b>Exit Conditions</b>	<ul style="list-style-type: none"> <li>User click “Make Payment” button.</li> </ul>
<b>Flow of Events</b>	<ol style="list-style-type: none"> <li>User chooses the food they want to buy and reaches the payment page.</li> <li>User enters card information and presses the "Make Payment" button.</li> </ol>



### 3.5.2 Use Case Model

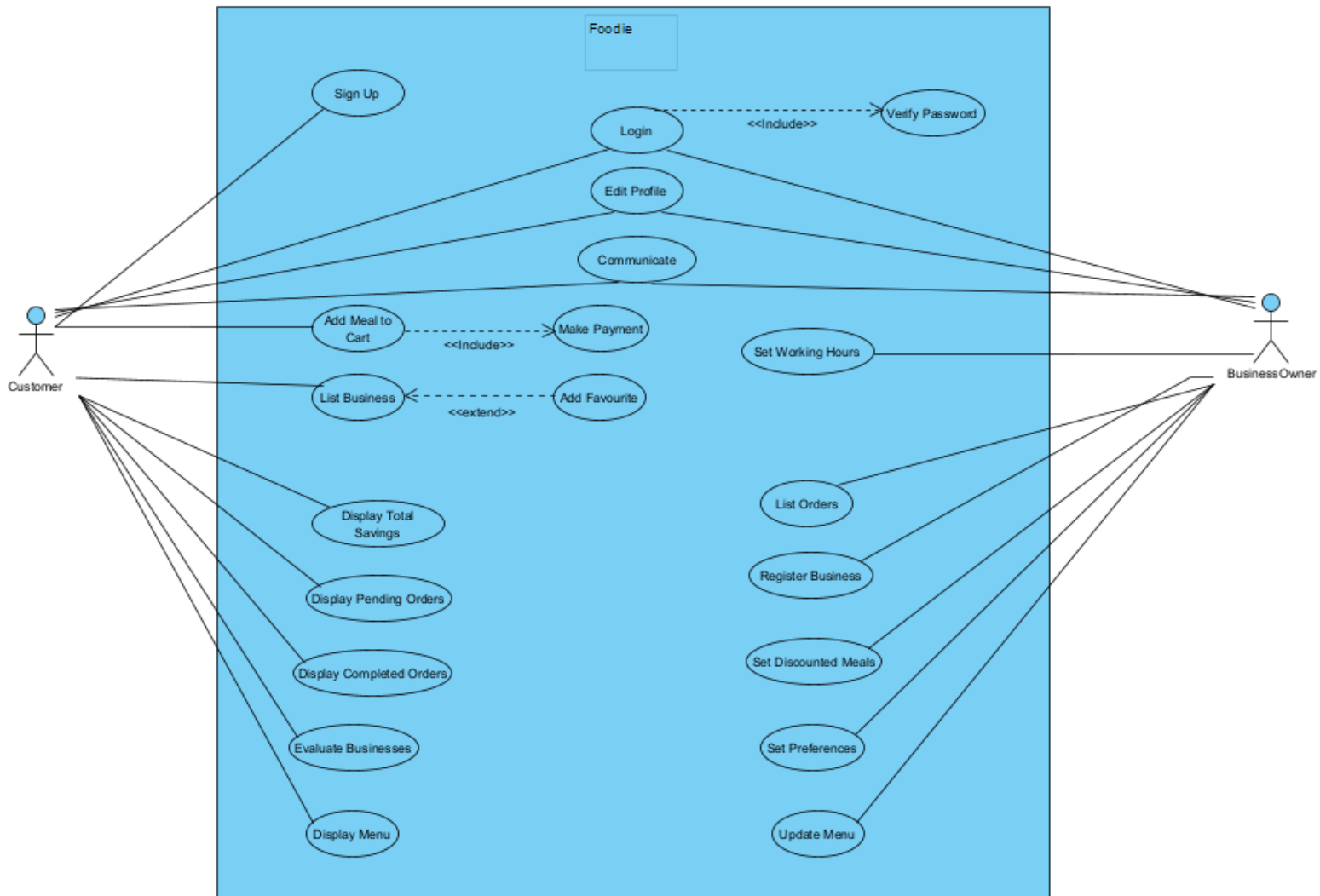


Figure 1: User Case Diagram of Foodie

### 3.5.3 Object-Class Model

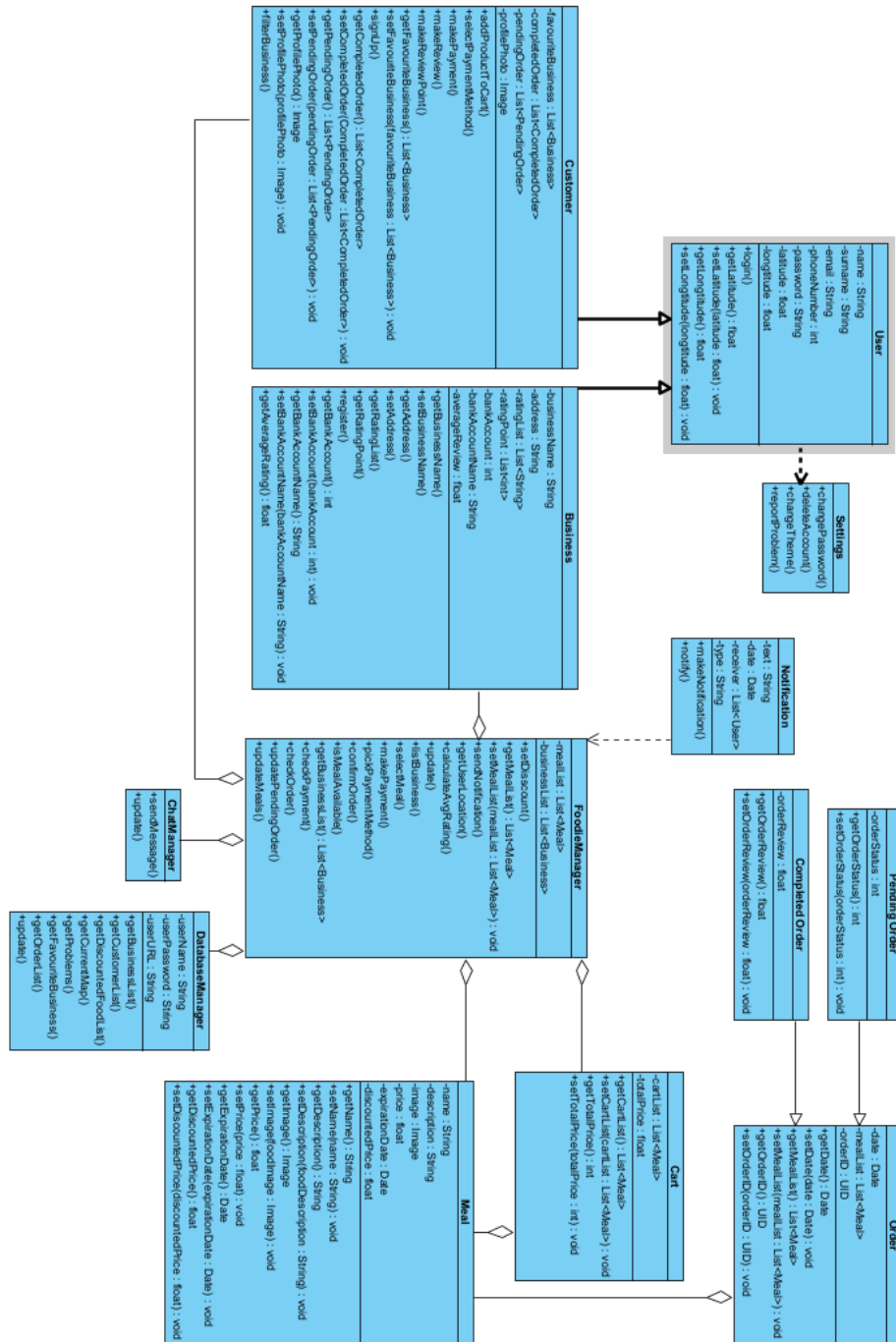


Figure 2: Object-Class Model of Foodie

<https://imgur.com/a/IXtDKiN>

## 3.5.4 Dynamic Models

### 3.5.4.1 Sequence Diagrams

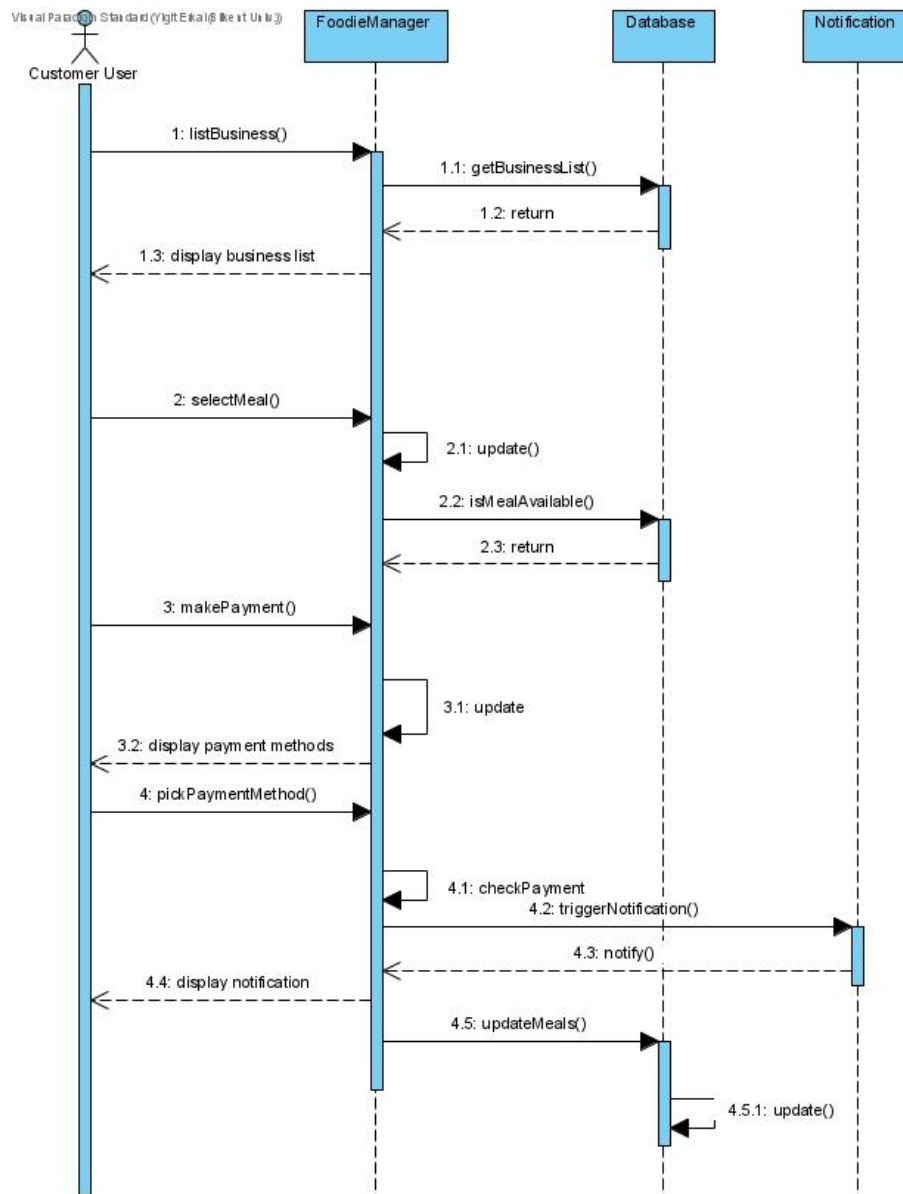


Figure 3: Sequence Diagram of Buying Food

In this sequence, the user presses the button for the list of business and FoodieManager gets the business list from the database and it returns with the list. As a result, the application displays the business list to the customer user. Then, the user selects the desired meal from the discounted menu. FoodieManager updates the cart and checks the availability of the meal. Lastly, the customer user selects the payment button to make a payment, and FoodieManager displays the payment methods page. Customer user selects a method and makes the payment. FoodieManager checks if the payment is done or not and it runs the triggerNotification method to prepare a notification for completed order.

Then, the user sees the notification about the completed order. As a last, after the completion of the order, FoodieManager updates the availability of the meal from the database.

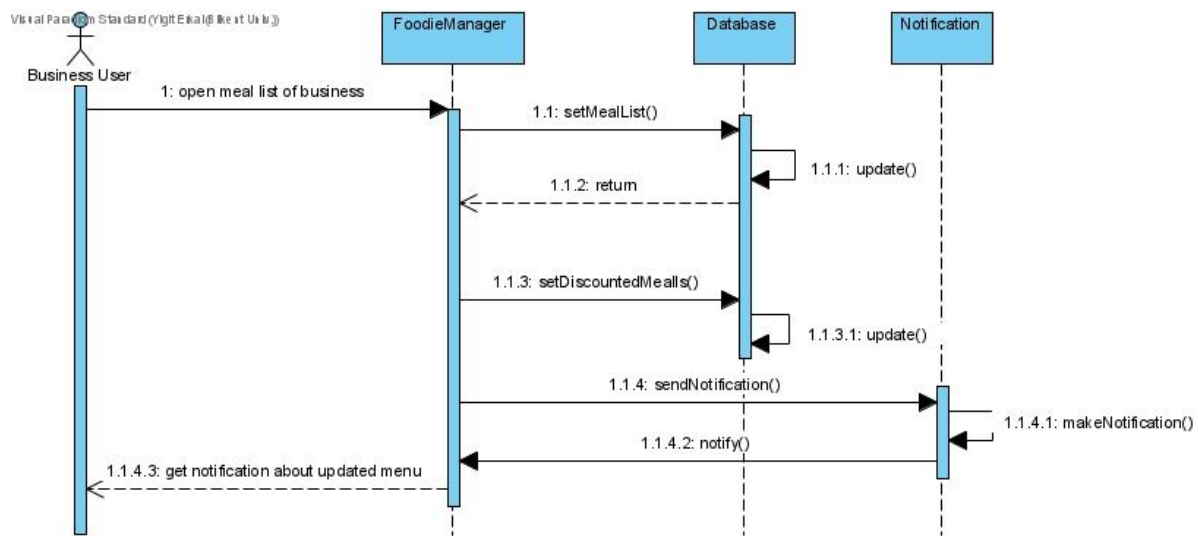


Figure 4: Sequence Diagram of Updating Menu

In this sequence, the business user selects the update menu button to change the menu. Then, the business user sets the meal list and this action updates the database. Then, the user prepares the discounted meals after the completion of the menu. Finally, FoodieManager runs the sendNotification() function to notify business users and the application displays the notification about the successfully updated menu.

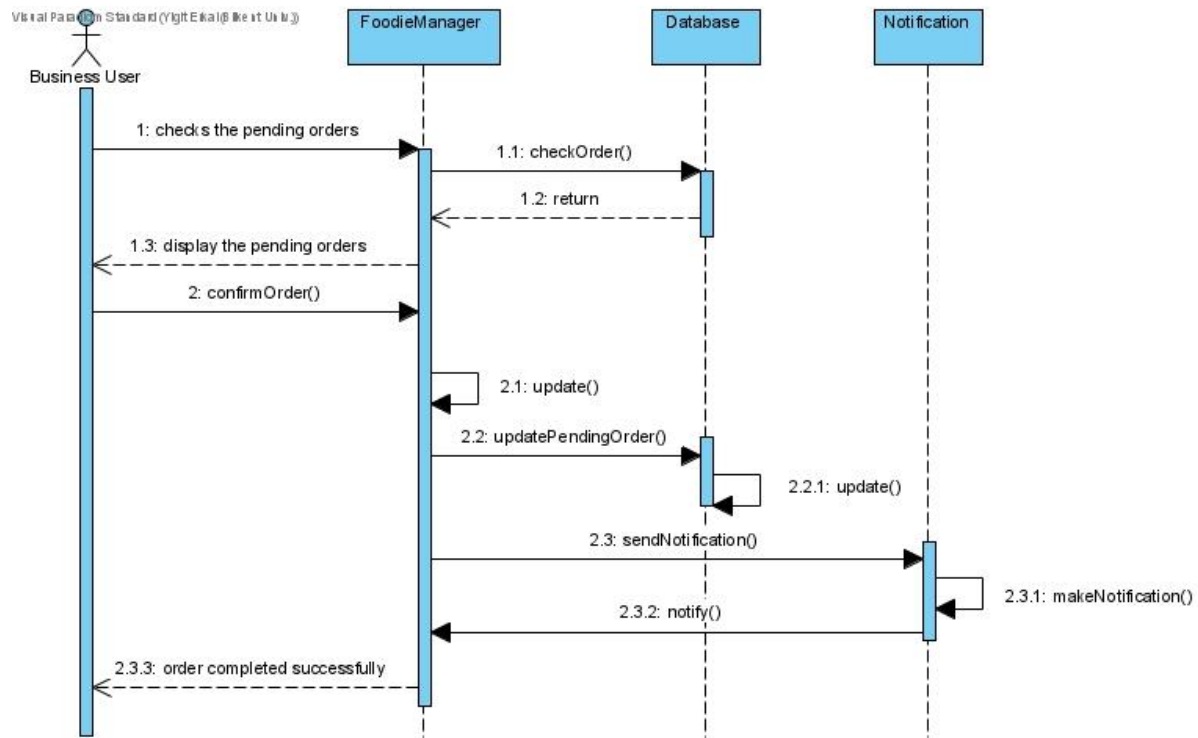


Figure 5: Sequence Diagram of Accepting Pending Order

In this sequence, a business user presses the my orders button to check pending orders. Then, FoodieManager runs the `checkOrder()` function to get data from the database. As a result of this action, user sees the list of pending orders. For each order, there is a confirm or reject order button. To accept the pending order, user selects the accept button and it runs the `confirmOrder()` function. After this action, FoodieManager runs the `updatePendingOrder()` function to update the database. Lastly, FoodieManager runs the `sendNotification()` function for business users and the application displays the successful acceptance of the order.

### 3.5.4.2 Activity Diagrams

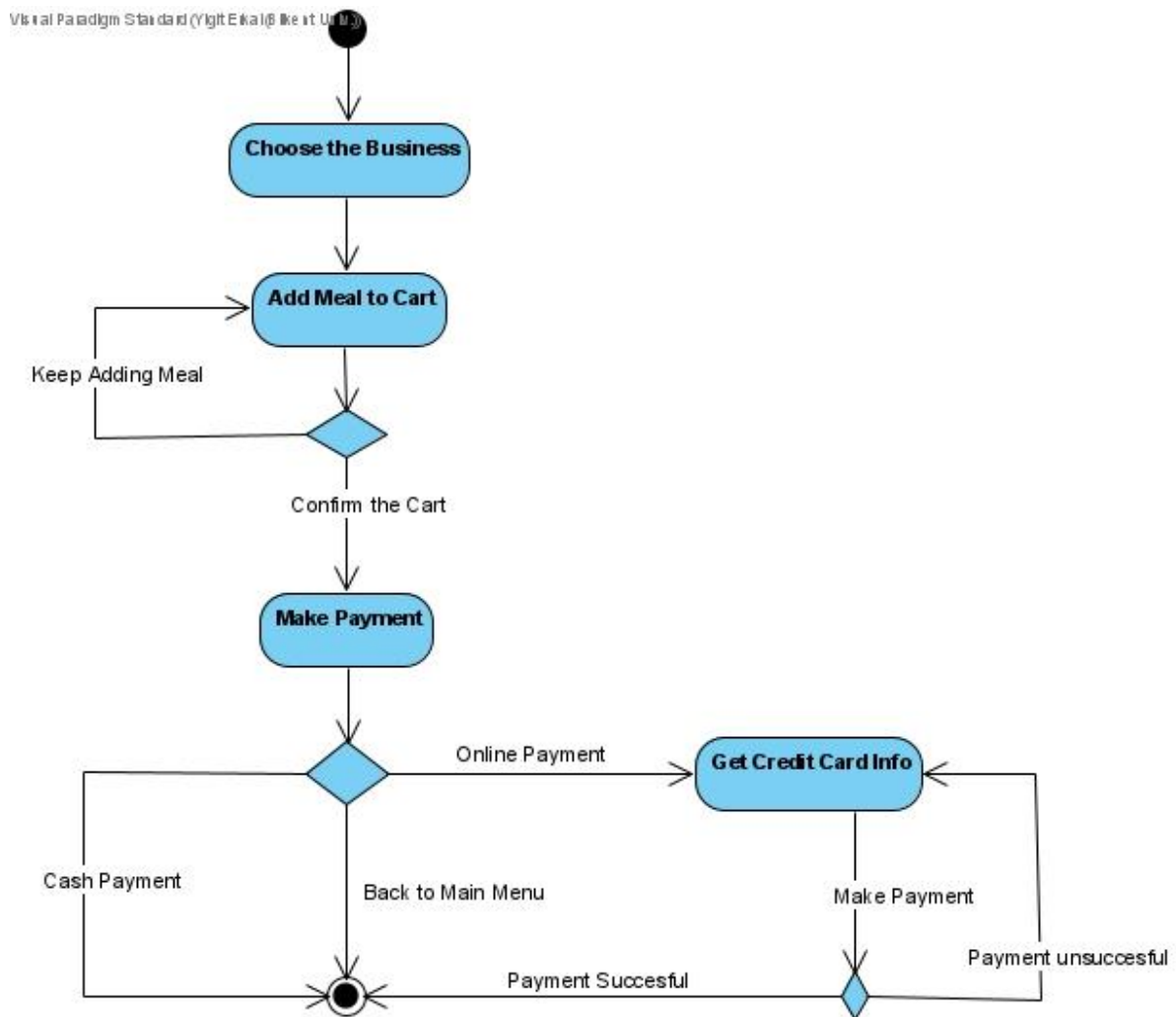


Figure 6: Activity Diagram of Making Payment

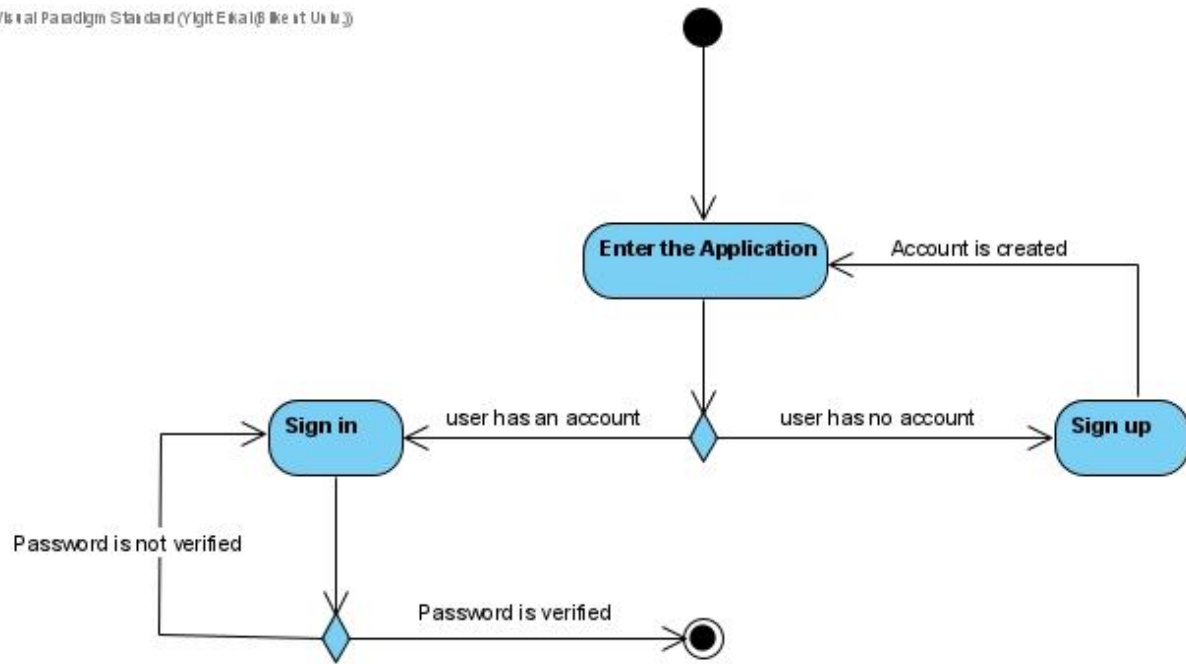


Figure 7: Activity Diagram of Entering the Foodie

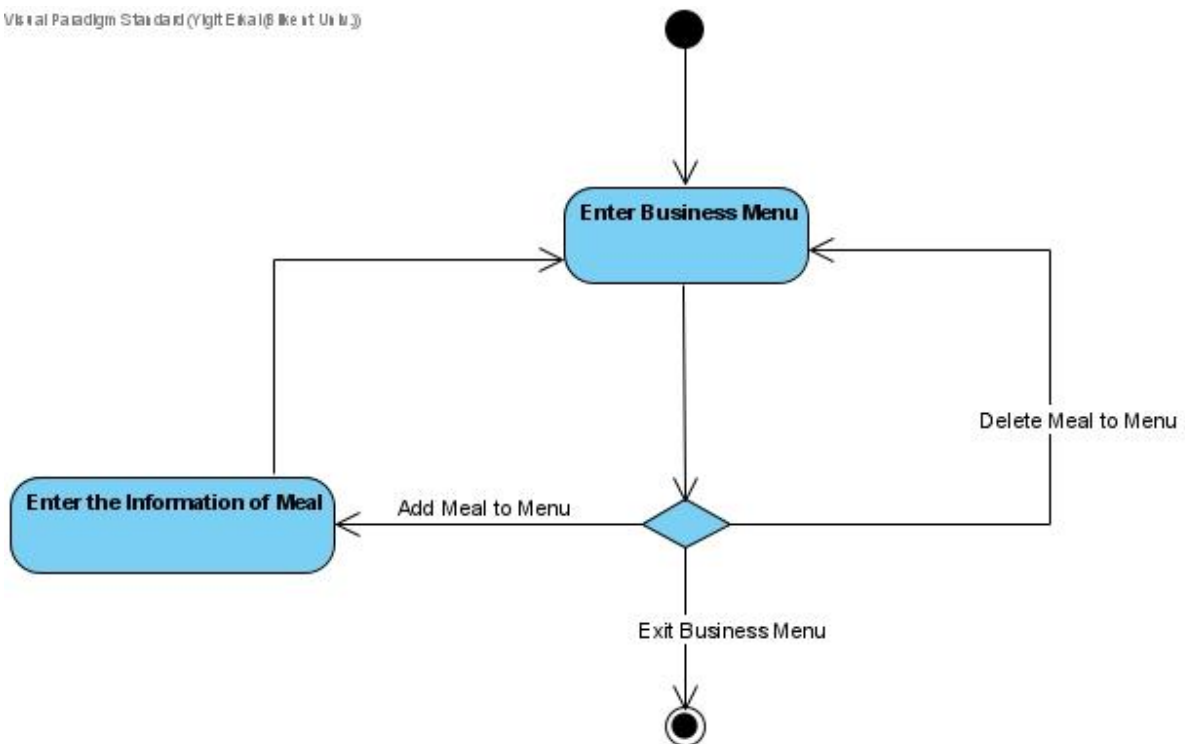


Figure 8: Activity Diagram of Updating Menu

### 3.5.4.3 State Diagrams

Visual Paradigm Standard (Vijit Eka Istitusi)

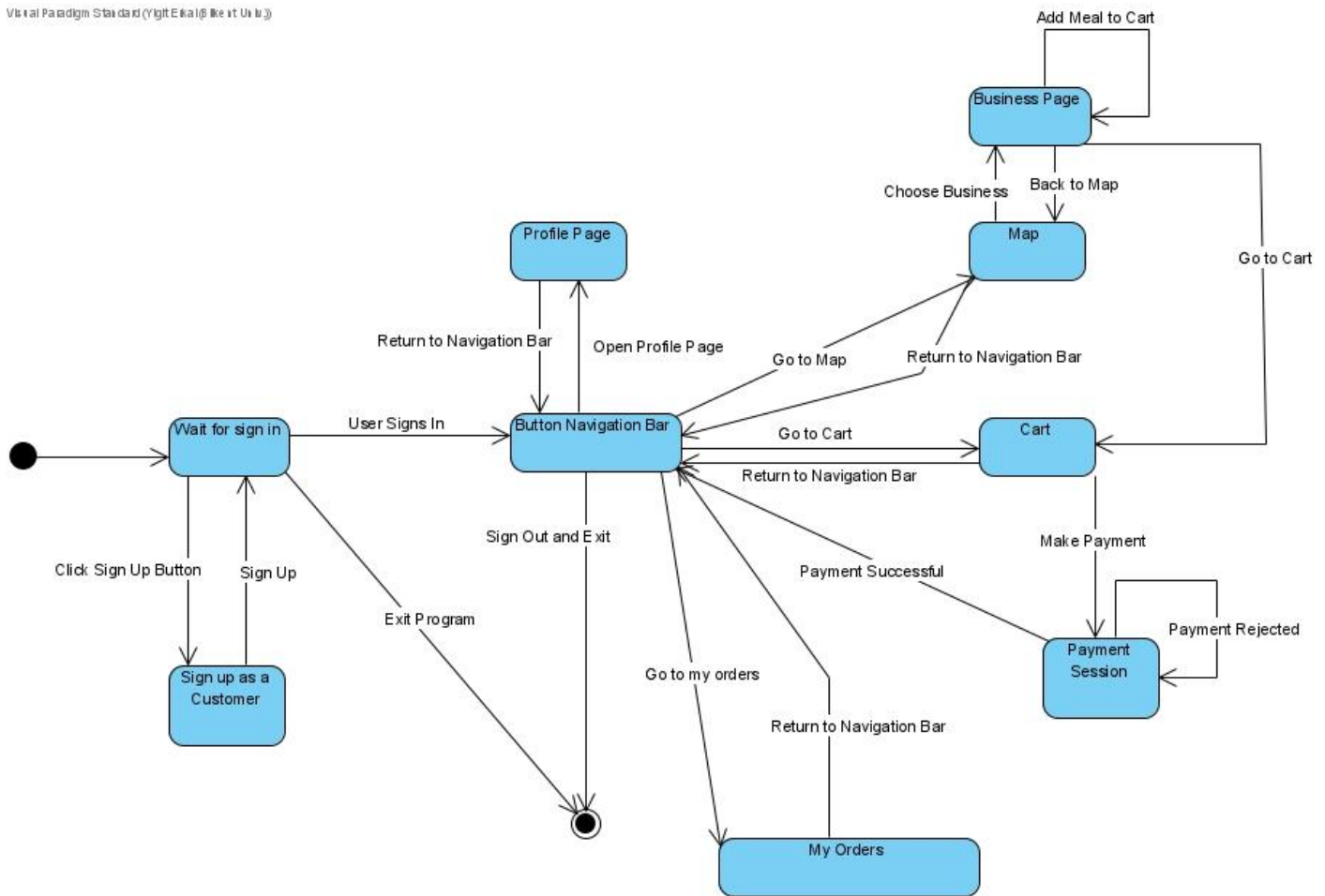


Figure 9: State Diagram of Customer User



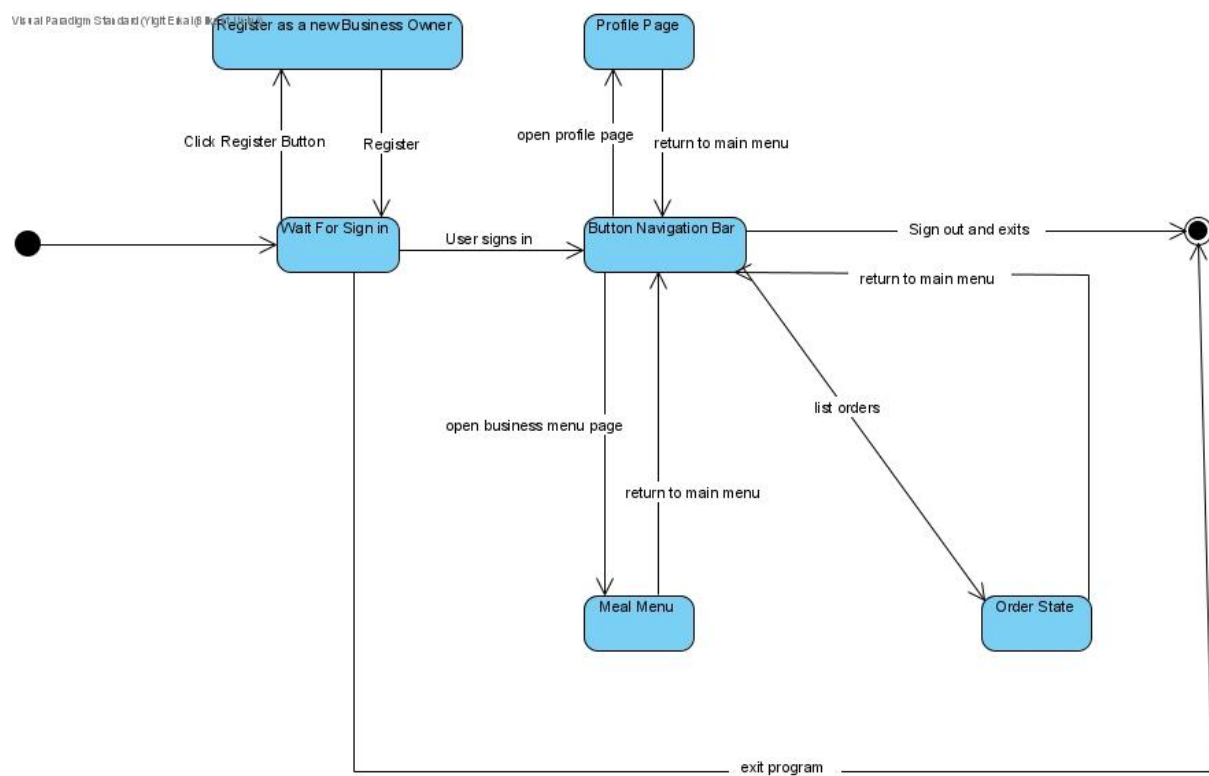


Figure 10: State Diagram of Business User

### 3.5.5 User Interface - Navigational Paths and Screen Mock-ups

#### 3.5.5.1 Navigational Path



Figure 11: Customer Navigational Path

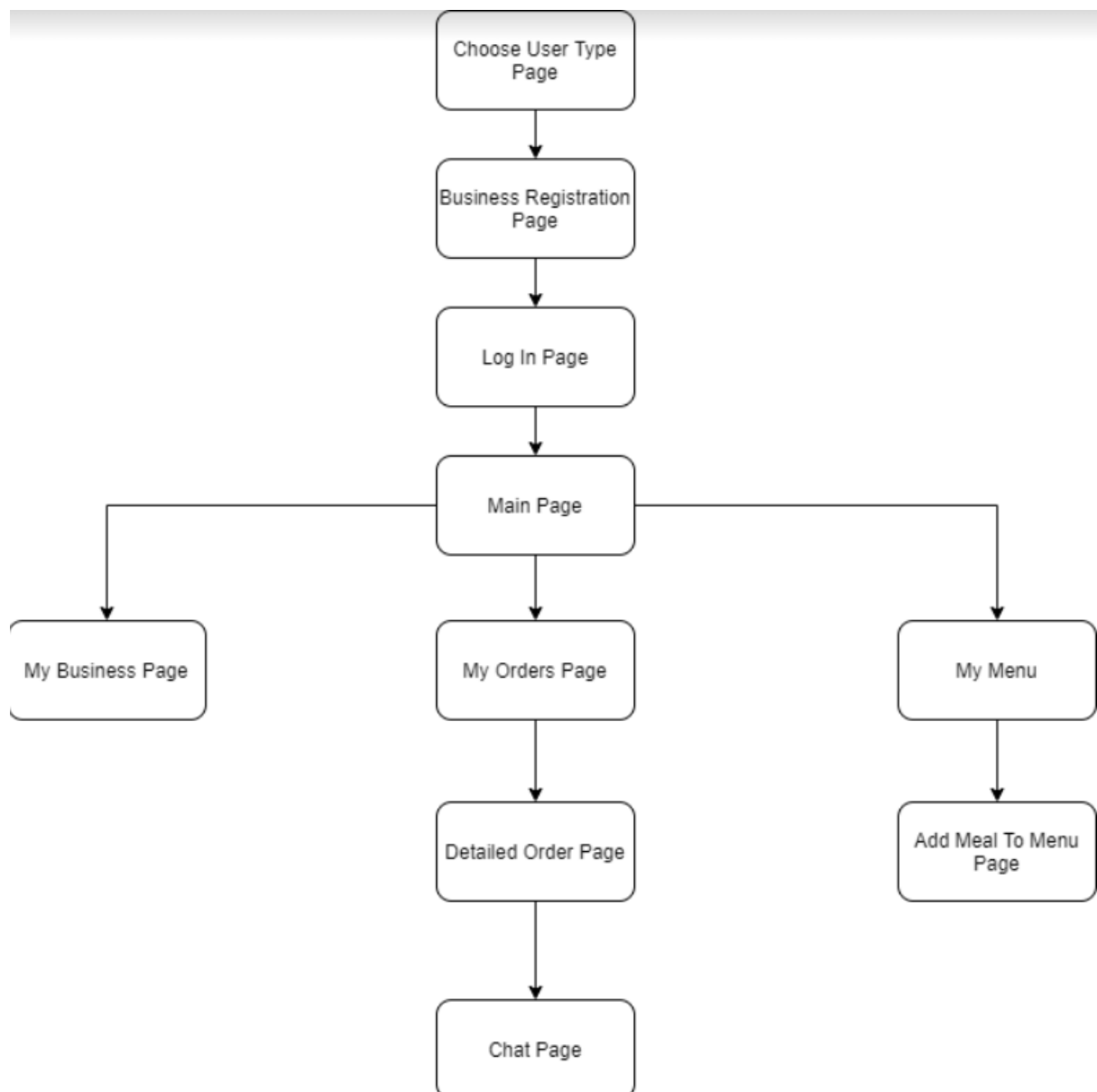


Figure 12: Business Navigational Path

### 3.5.5.2 User Interface

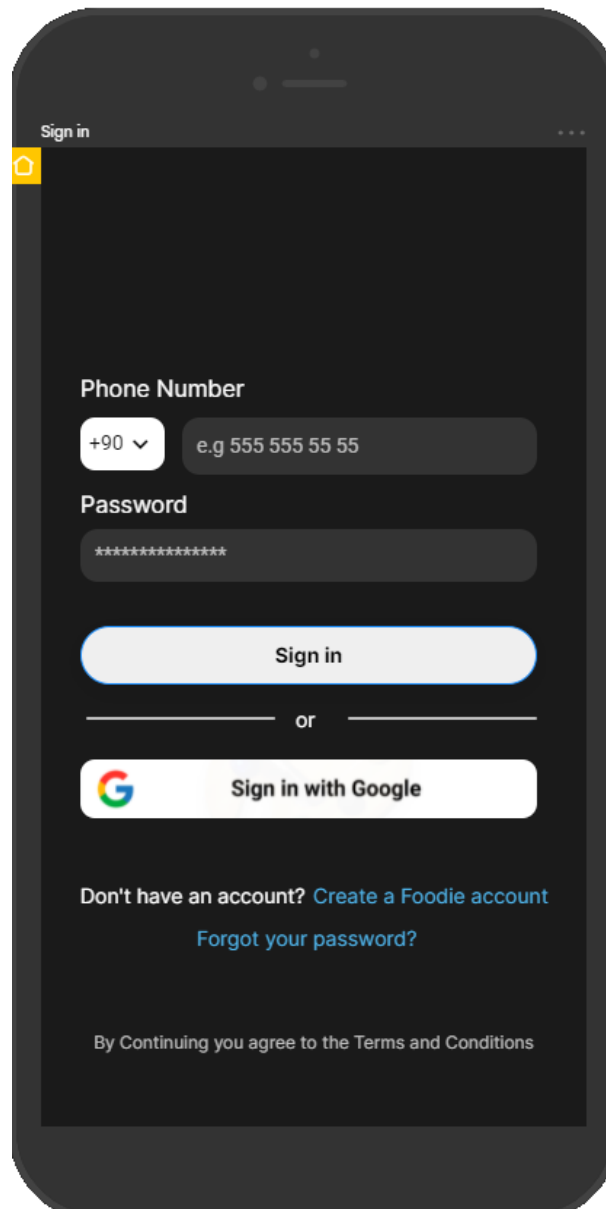


Figure 13: Sign In Page

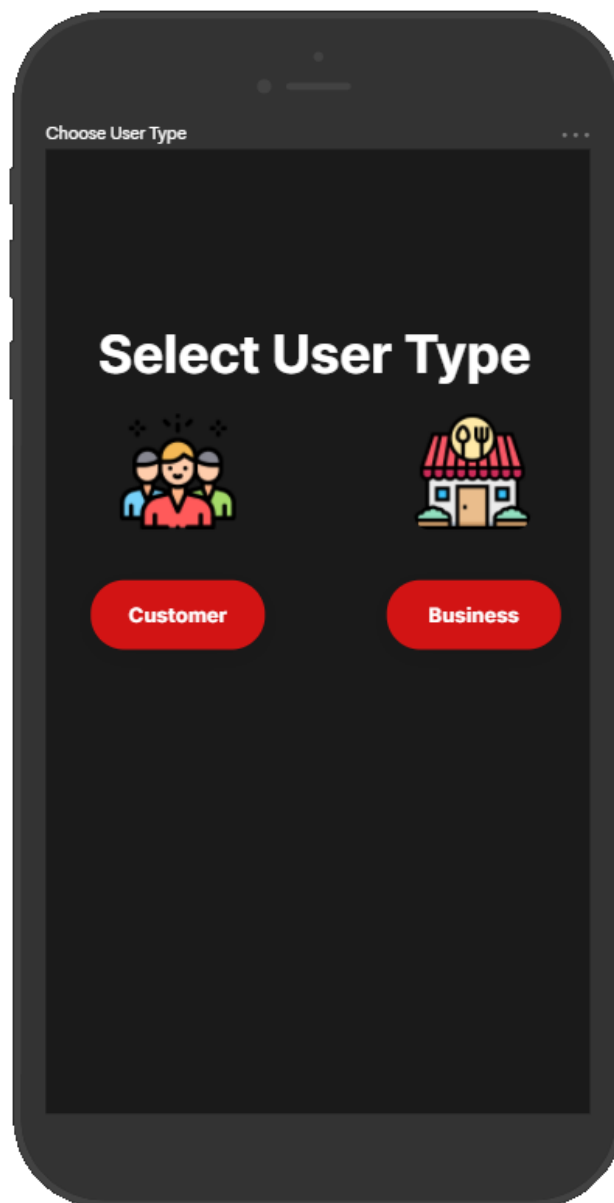


Figure 14: Select User Type Page

Sign Up Page

Name

Surname

+90 ▼ e.g 555 555 55 55

e-mail

Password

Re-enter password

**Sign Up**

Figure 15: Sign Up Page

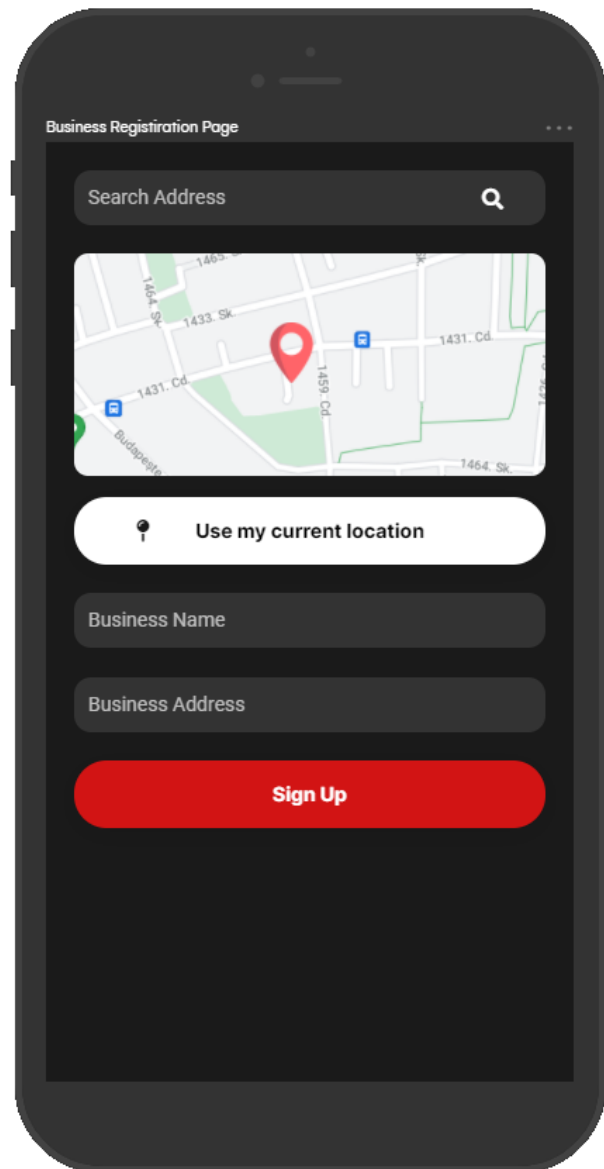


Figure 16: Business Registration Page

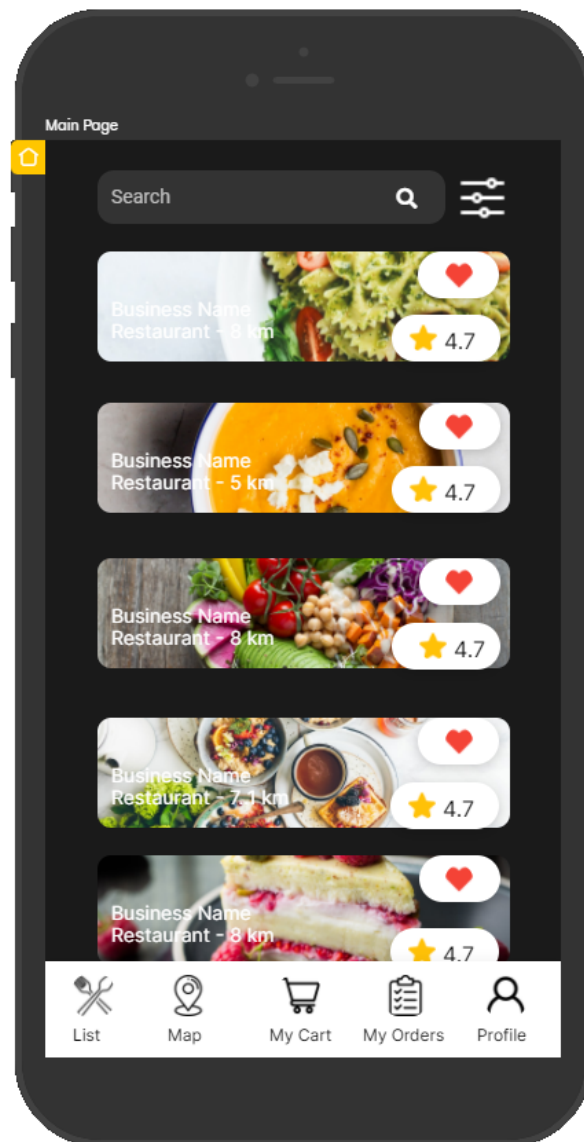


Figure 17: Main Page For Customers



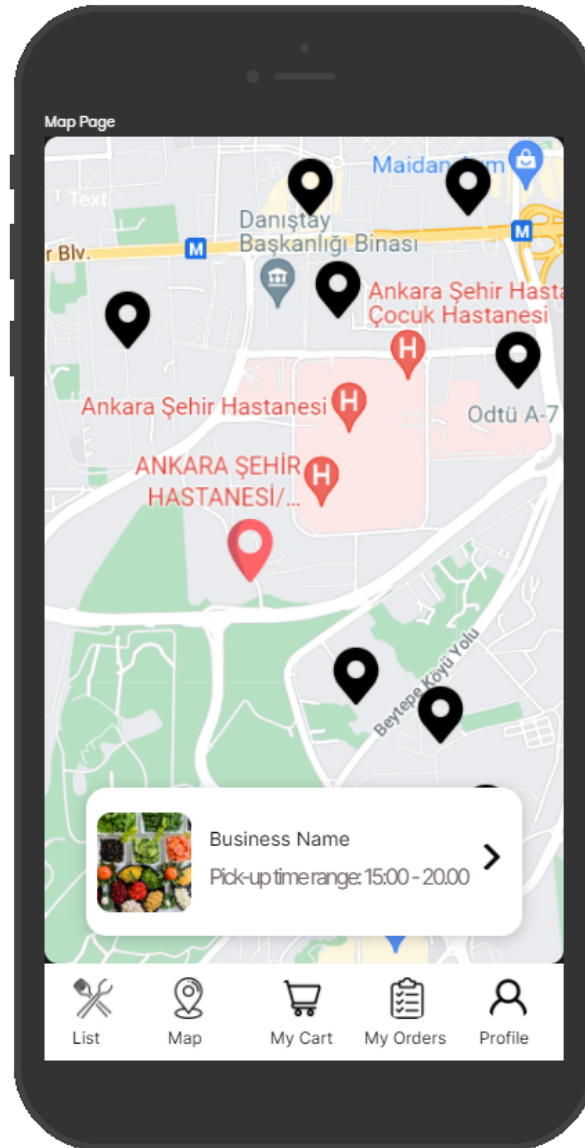


Figure 18: Map Page For Customers

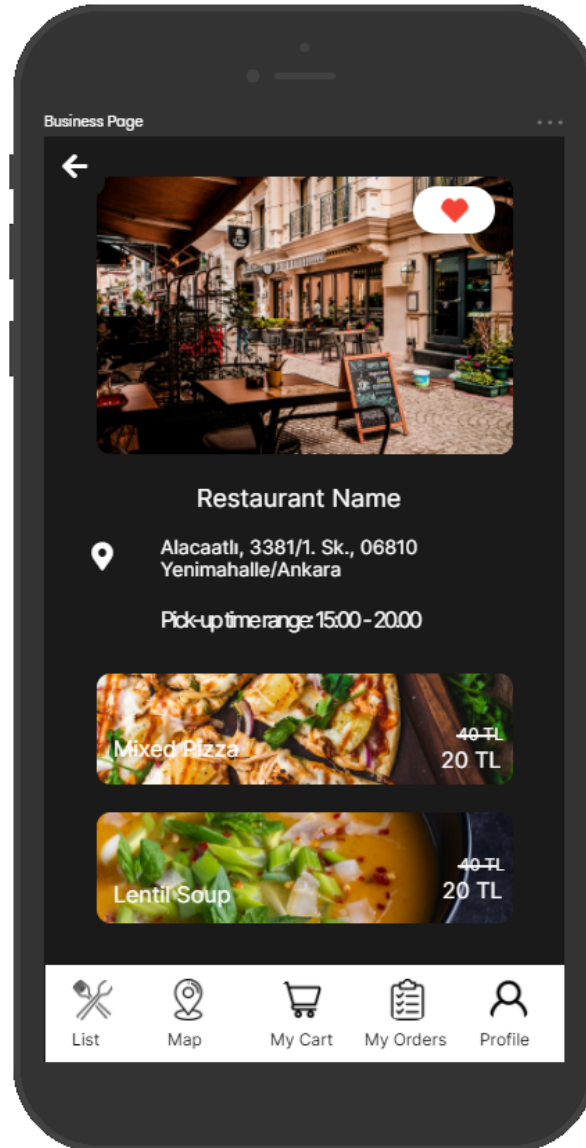


Figure 19: Business Page For Customers

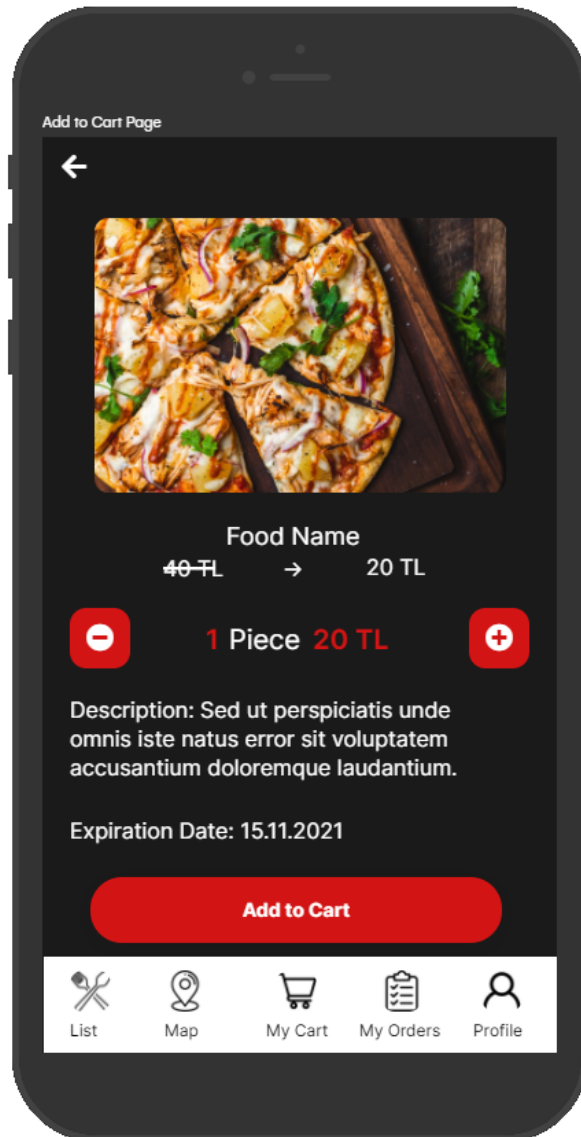


Figure 20: Add To Cart Page For Customers

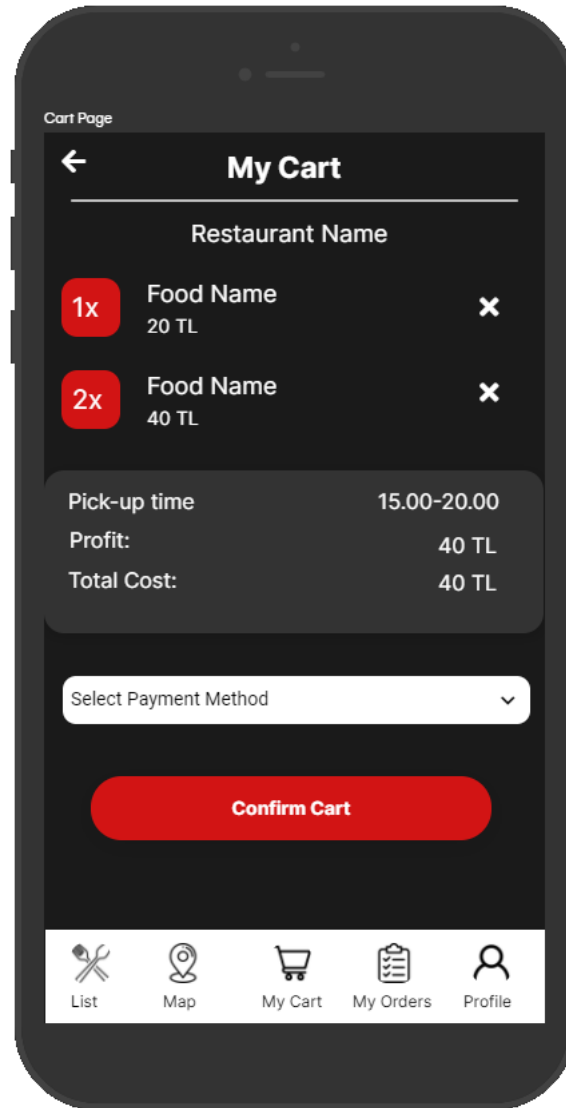


Figure 21: Cart Page For Customers

Payment Page

← **Payment**

---

Total Cost 40 TL

---

First Name Last Name

Card Number

MM / YY CVV

☐ I have read and approved the pre-information conditions and the distance sales contract.

**Purchase**

Figure 22: Payment Page

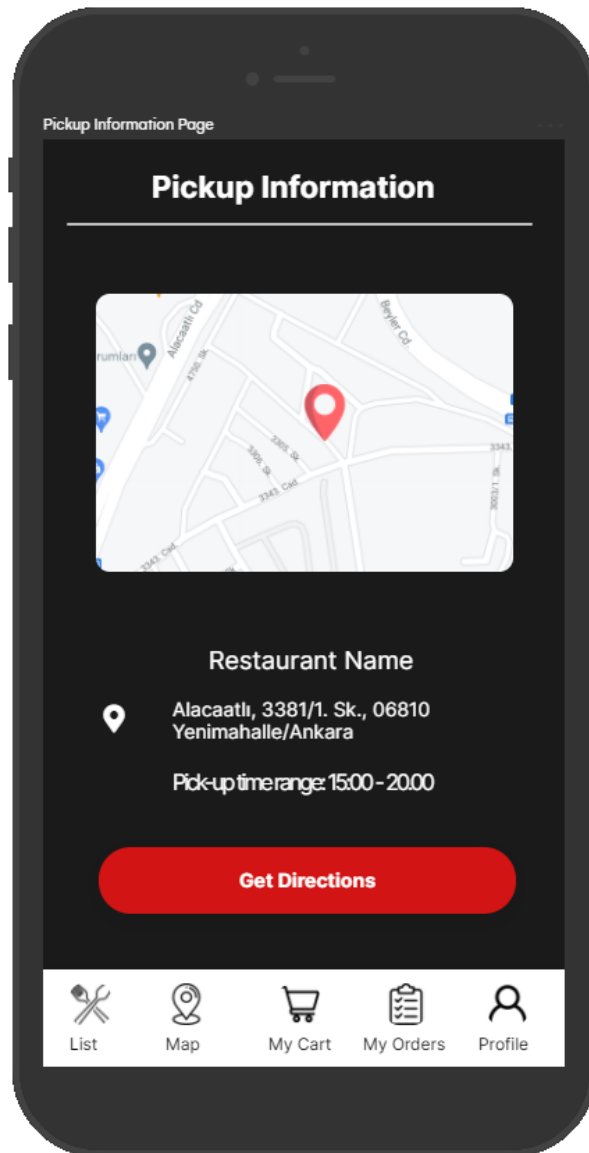


Figure 23: Pick-up Information Page For Customers

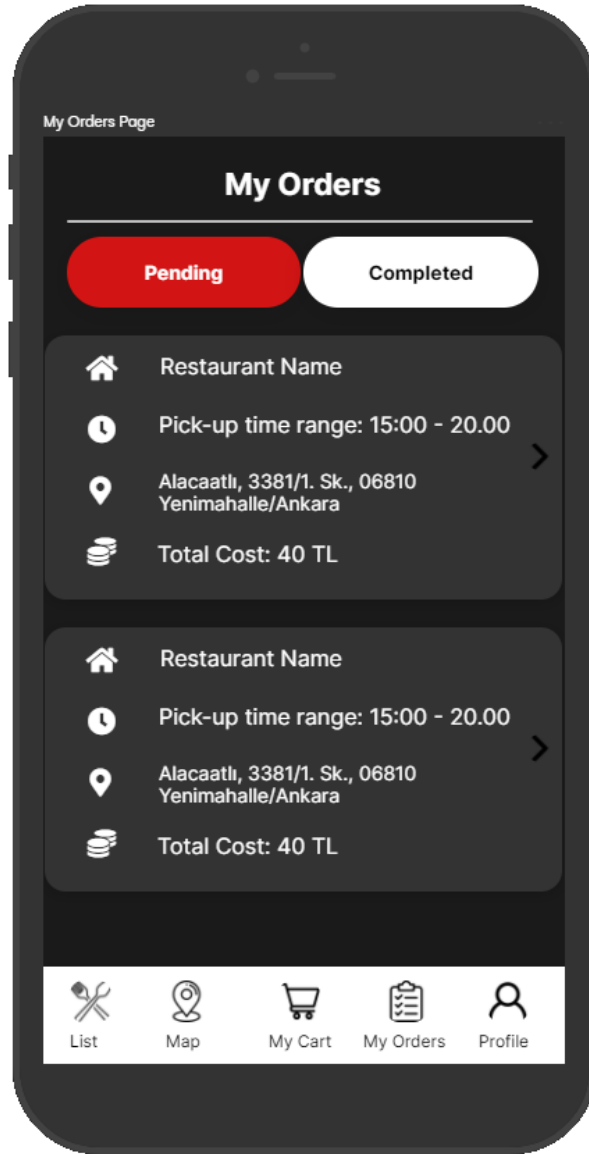


Figure 24: My Orders Page For Customers

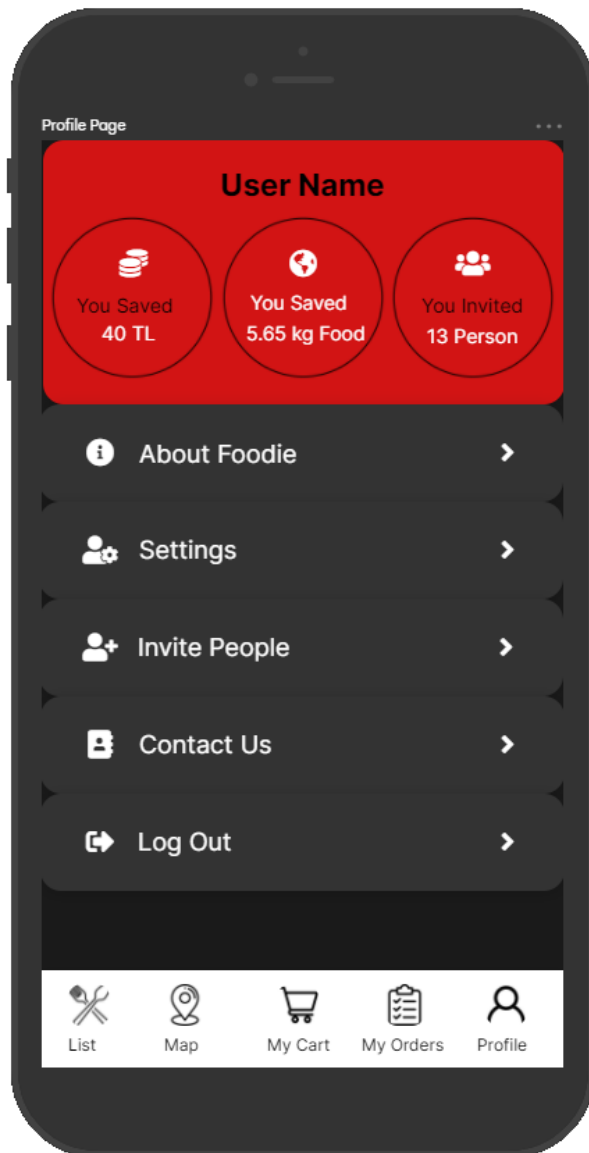


Figure 25: Profile Page For Customers



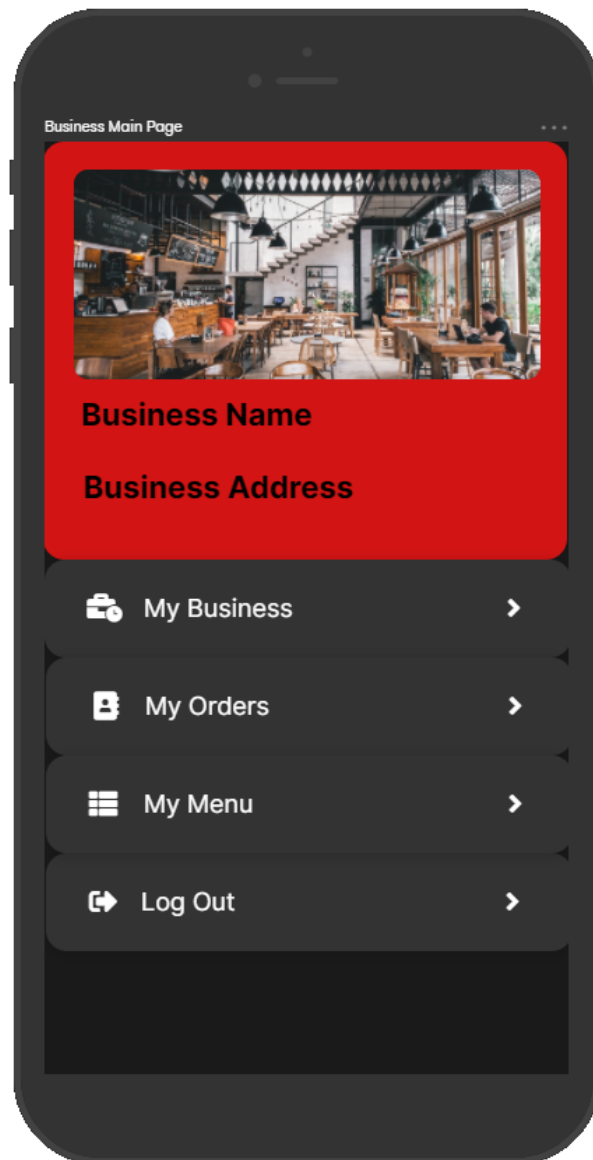


Figure 26: Main Page For Businesses

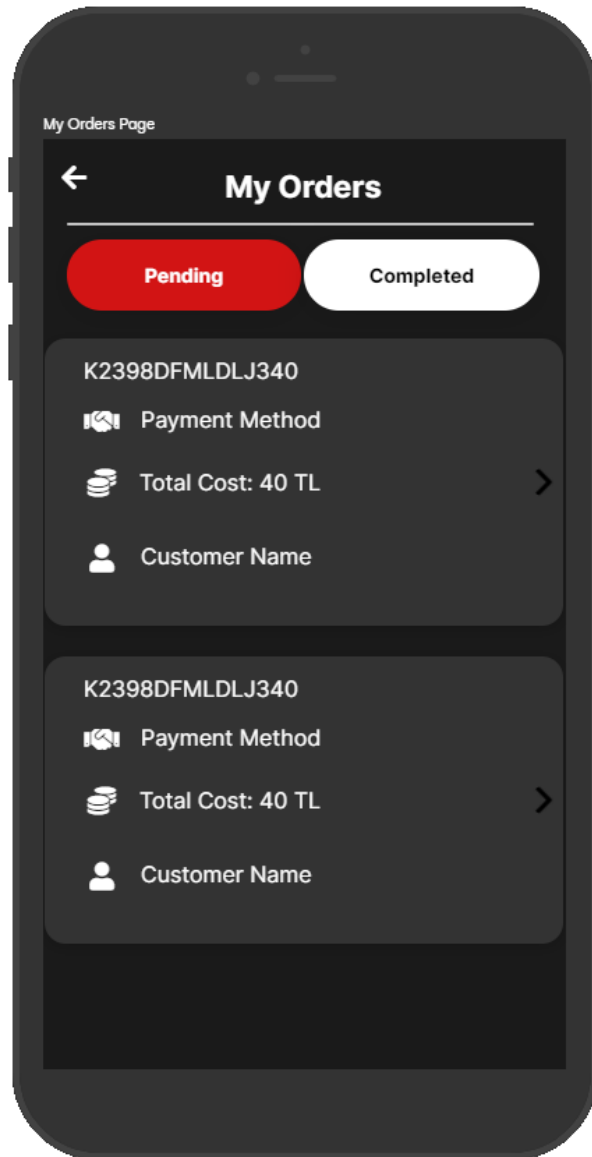


Figure 27: My Orders Page For Businesses

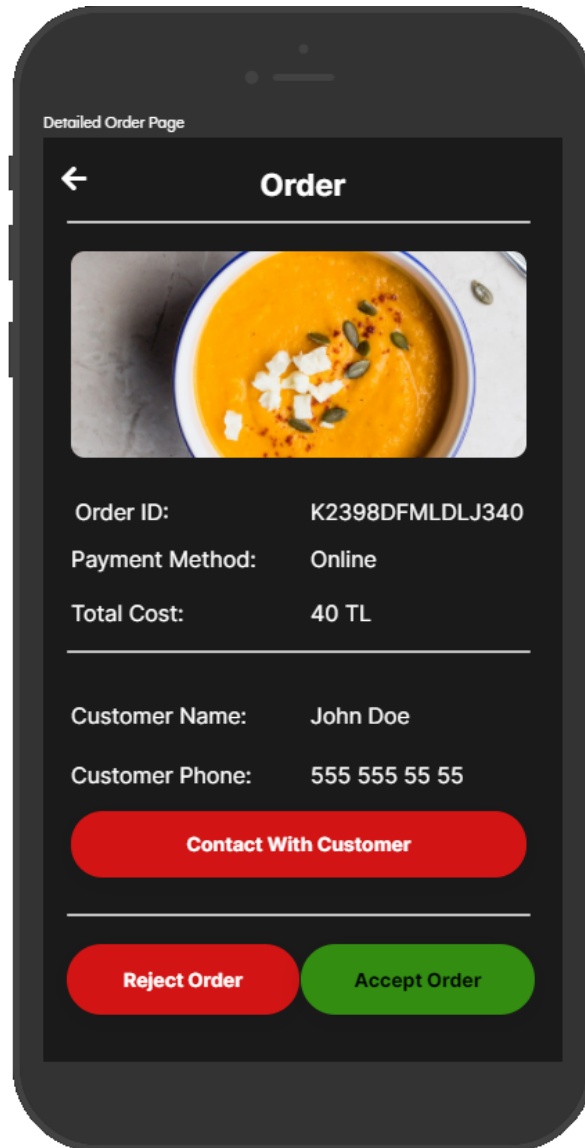


Figure 28: Detailed Order Page For Business

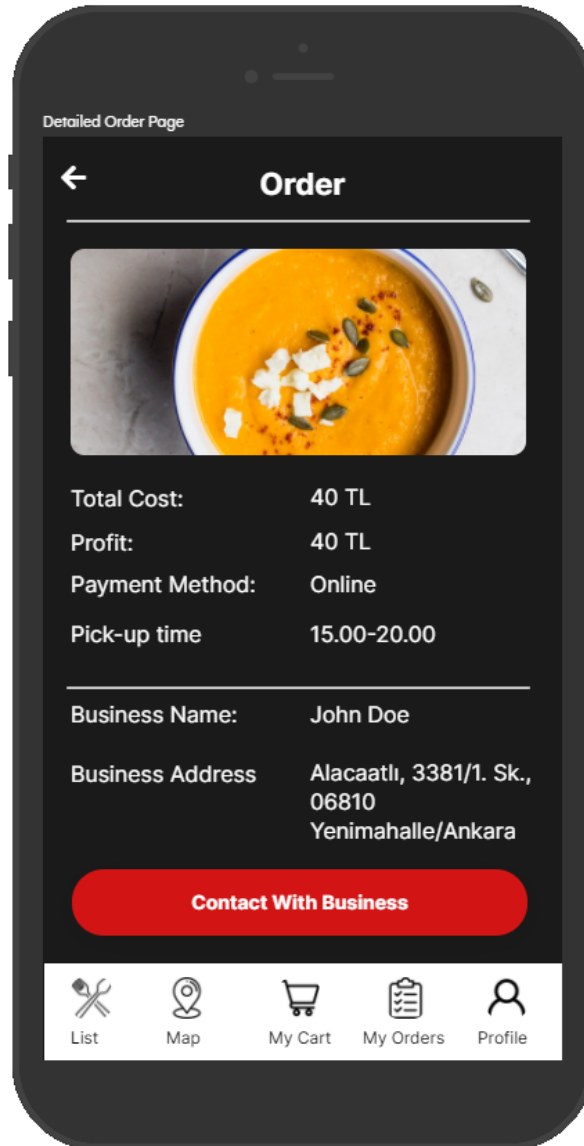


Figure 29: Detailed Order Page For Customers

My Menu

← **Business Name** ✎

---

**Business Address** ✎

**Bank Information** ✎

Name Surname

IBAN No

**Pick-up Time: 17.00-20.00** ✎

**Save**

Figure 30: My Business Page

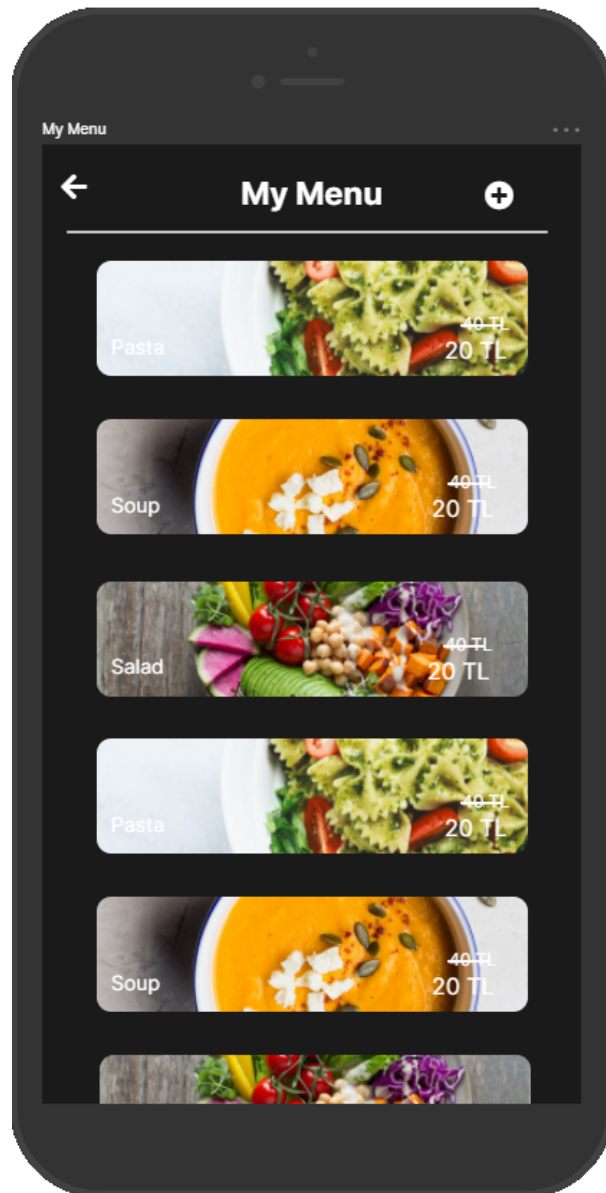


Figure 31: My Menu Page

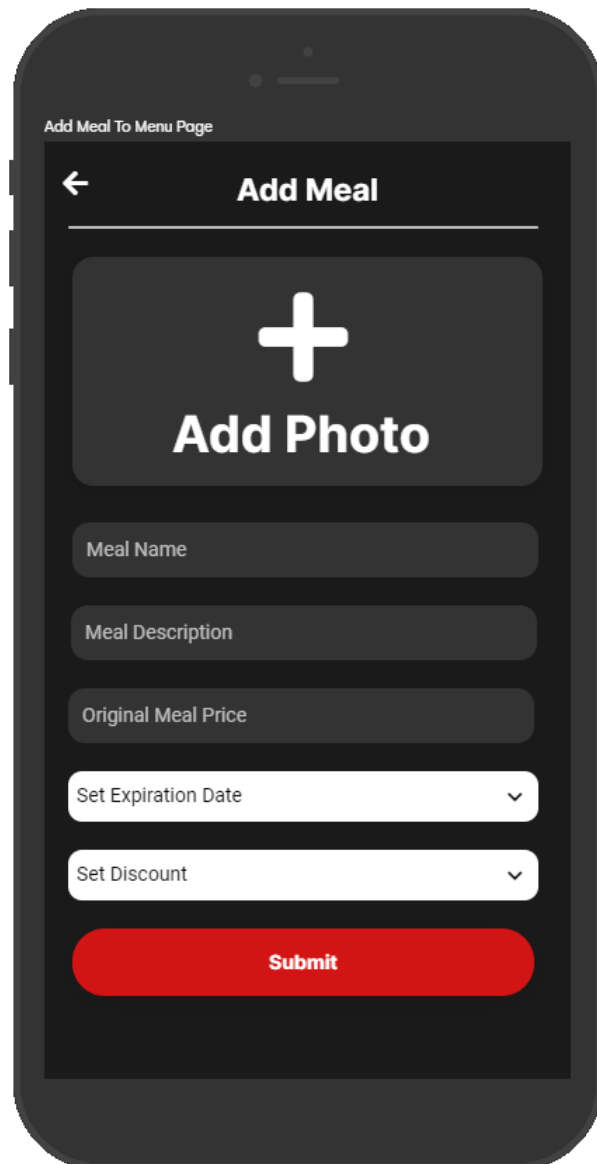


Figure 32: Add Meal To the Menu Page

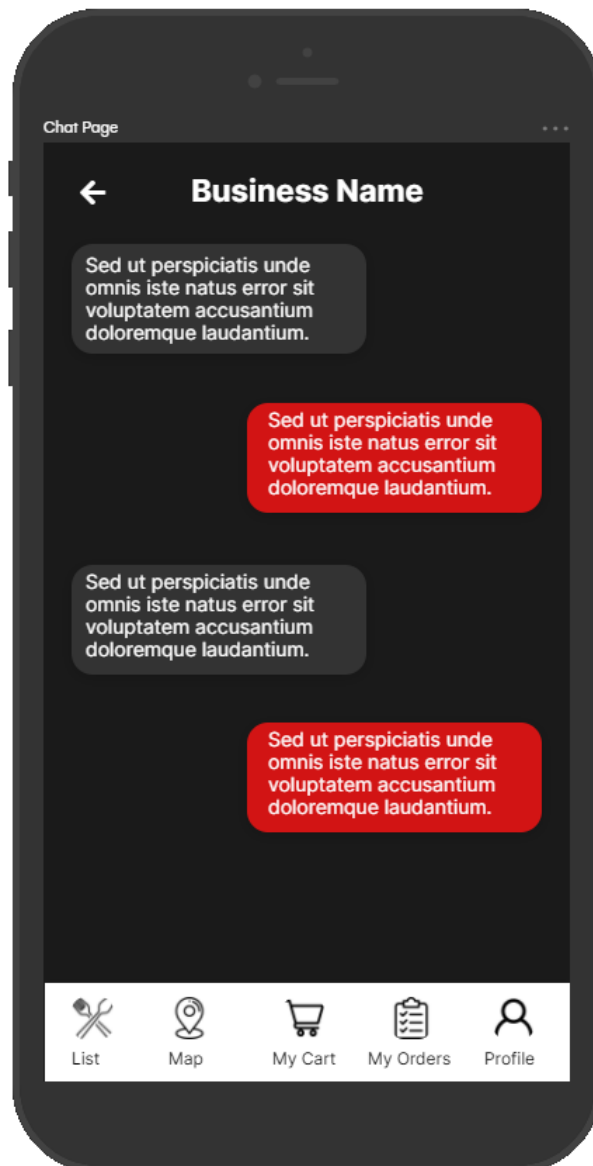


Figure 33: Chat Page



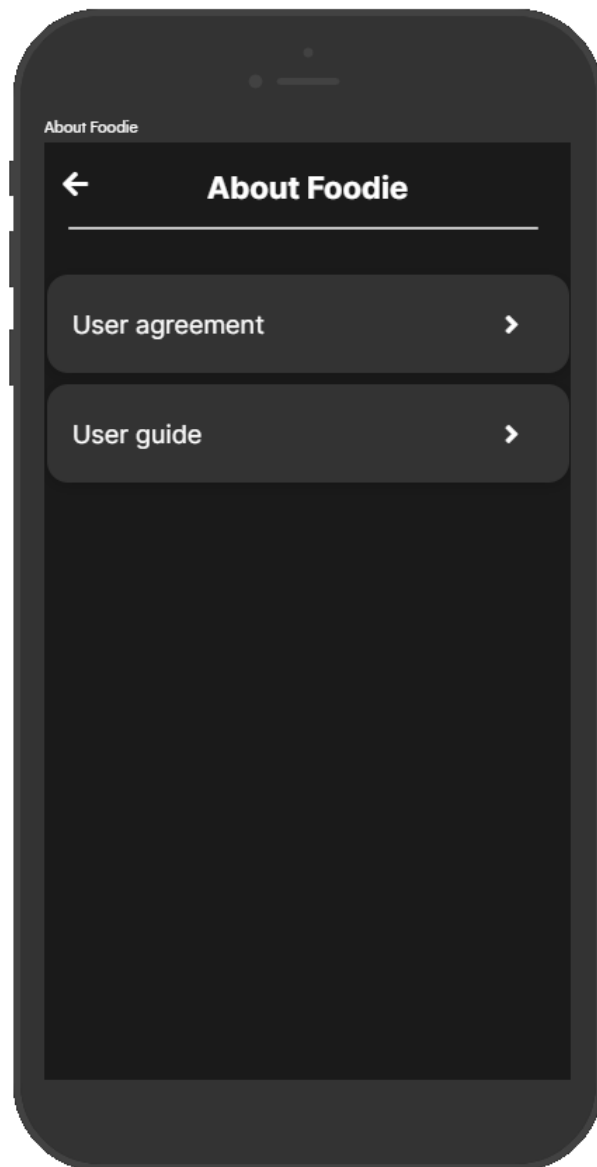


Figure 34: About Page

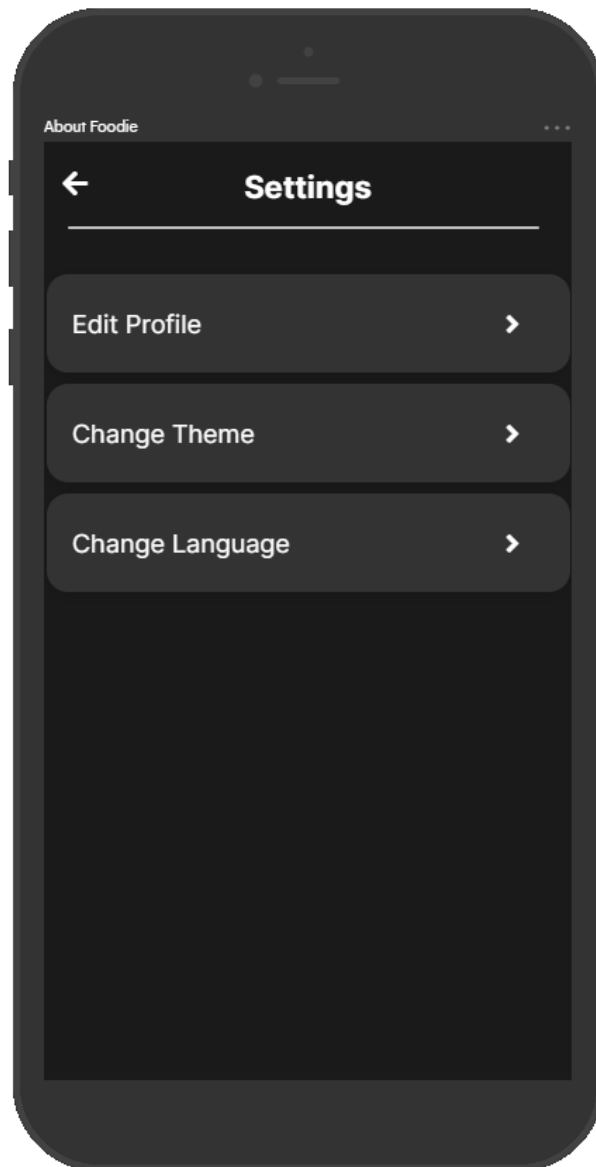


Figure 35: Edit Profile Page

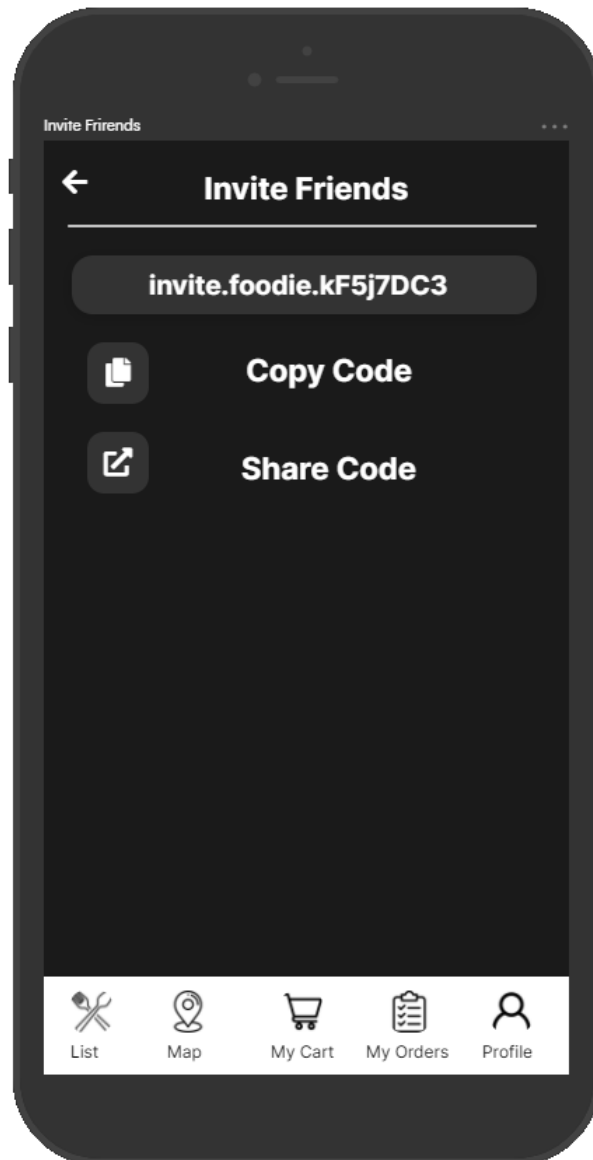


Figure 36: Invite Friends Page

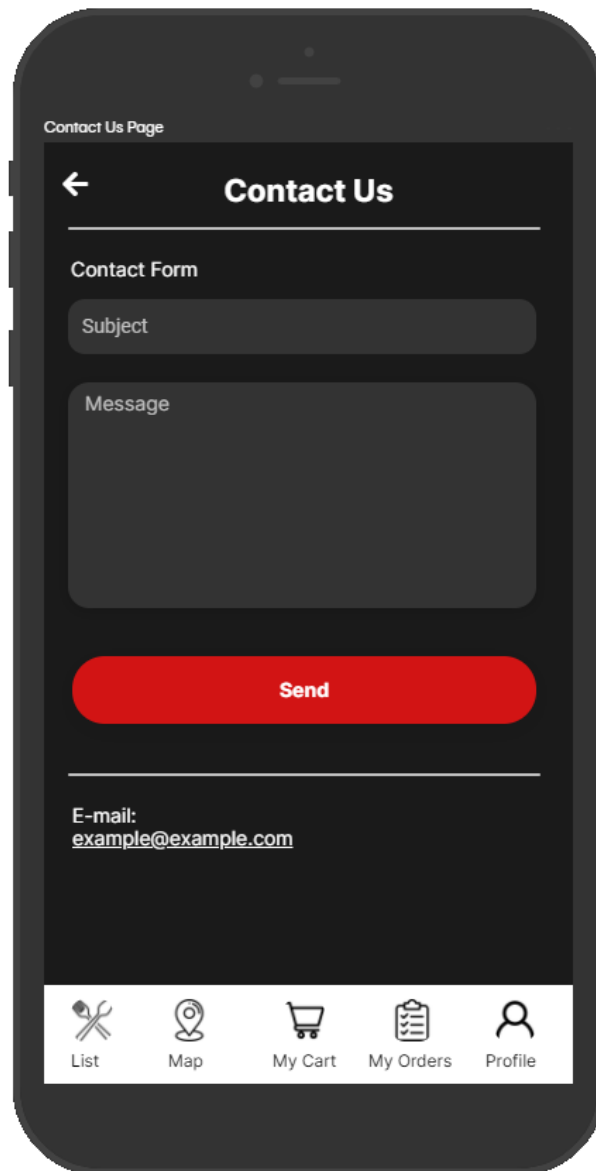


Figure 37: Contact Us Page

## **4. Other Analysis Elements**

### **4.1. Consideration of Various Factors in Engineering Design**

#### **Public Health**

Restaurants may want to sell their outdated food. This situation is a risk factor for public health so we will check the comments of the restaurants regularly. If there are comments about outdated foods than usual, we will suspend these restaurants strictly.

#### **Public Safety**

Our application uses some personal information such as location information. This kind of personal information may be a problem for public safety. Hence, this information will not be shared by any third-party groups.

#### **Public Welfare**

Our application is aimed to enhance the public welfare by giving a chance to sell leftover foods with lower prices. Hence, leftover foods will not be wasted and people in need will have a chance to buy discounted foods.

#### **Global Factors**

Our application will not have any effect on global factors. Foodie will be a platform especially created for Turkey. Furthermore, we will give an option to change the language to English because of the foreign people who live in Turkey.

#### **Cultural Factors**

There will be a cultural interaction with foods. People will get a chance to buy foods from other cultures with lower prices.

#### **Social Factors**

Our mobile application aims to create social awareness by finding a way to use leftover foods. Furthermore, people in need will get a chance to buy discounted foods easily.

## Environmental Factors

Leftover food is thrown away by most of the restaurants. Restaurant's wastes may cause more damage to the environment. Our application will find a solution for this waste.

## Economic Factors

Our mobile application also aims to find a solution for people in need. With the help of discounted leftover food, these people will reach the food with very optimal prices.

	Effect Level	Effect
Public Health	10	Possibility of selling outdated food
Public Safety	6	Keeping personal data confidential
Public Welfare	8	People who are in need will have a chance to get discounted food.
Global Factors	0	Global factors have no effect on our solution.
Cultural Factors	6	Possibility of eating foods from different cultures.
Social Factors	7	Our application aims to create a social awareness about leftover foods.
Environmental Factors	10	Application will be environmentally friendly and it will aim to minimize leftover foods.
Economic Factors	10	Discounted foods will be an opportunity to people who are in need

## 4.2. Risks and Alternatives

Due to wrong location detection and abusing the price levels of the discounted foods, there are probable risks. Hence, our project group needs to solve these possible risks.

### **Wrong user location detection**

**Risk Definition:** Detecting the restaurant locations incorrectly.

**Explanation:** Each restaurant must mark their location from a map. However, there may be mistakes during this marking period. Therefore, restaurant locations can be saved with wrong location information.

**Likelihood:** High

**Effect on the project:** High

**B plan summary:** In our project, we want to give highly accurate locations, therefore, our project group needs to solve this problem. As a plan B, we will give an option to restaurants to enter their address by writing directly. Hence, we will avoid such wrong location risks.

### **Fraudulent price display**

**Risk Definition:** Restaurants may abuse the discounted prices.

**Explanation:** Restaurants have their own price list and this list is open to abuse with entering the higher prices than expected values. Hence, they may pretend it's a discounted price and sell that food with higher prices.

**Likelihood:** High

**Effect on the project:** High

**B plan summary:** In our project, we want to avoid any abusing action. Hence, restaurants have to enter their menu prices firstly to start to sell discounted leftover foods. This plan B action will reduce the probability of price abusing actions.

### 4.3. Project Plan

WP#	Work Package Title	Leader	Start Date	End Date
WP1	Documentation	Utku Gökçen	09.11.2021	27.05.2022
WP2	Front-end Development	Utku Gökçen	01.12.2021	20.12.2021
WP3	Back-end Development	Yiğit Erkal	12.12.2021	01.04.2022
WP4	Testing	Emre Erciyas	08.04.2022	08.05.2022
WP5	Launching the application	Utku Gökçen	12.05.2020	19.05.2020

<b>WP1: Documentation</b>
<b>Start Date:</b> 09.11.2021 <b>End Date:</b> 27.05.2022
<b>Leader:</b> Utku Gökçen <b>Members Involved:</b> Utku Gökçen, Yiğit Erkal, Emre Erciyas, İlhan Koç
<b>Objectives:</b> Planning all the steps of the project with documentation and making a detailed description of the developed product.
<b>Tasks:</b> <b>Task 1</b> Writing High Level Design Report <b>Task 2</b> Writing Low Level Design Report <b>Task 3</b> Writing Final Report
<b>Deliverables:</b> <b>Deliverable 1:</b> High Level Design Report <b>Deliverable 2:</b> Low Level Design Report <b>Deliverable 3:</b> Final Report



<b>WP2: Frontend Development</b>
<b>Start Date:</b> 01.12.2021 <b>End Date:</b> 20.12.2021
<b>Leader:</b> Utku Gökçen <b>Members Involved:</b> Utku Gökçen, Yiğit Erkal
<b>Objectives:</b> Implementation of user interface and navigational paths.
<b>Tasks:</b> <b>Task 1</b> Developing user interface with XML <b>Task 2</b> Creating navigational path of user interfaces

<b>WP3: Backend Development</b>
<b>Start Date:</b> 12.12.2021 <b>End Date</b> 01.04.2022
<b>Leader:</b> Yiğit Erkal <b>Members Involved:</b> Utku Gökçen, Yiğit Erkal, Emre Erciyas
<b>Objective:</b> Creating the architectural design of the application.
<b>Tasks:</b> <b>Task 1</b> Creating the architectural layers of the application <b>Task 2</b> Establishing the connection between the database and the application. <b>Task 3</b> Writing REST API Service.

<b>WP4: Testing</b>
<b>Start Date:</b> 08.04.2022 <b>End Date:</b> 08.05.2022

<b>Leader:</b> Emre Erciyas
<b>Members Involved:</b> Utku Gökçen, Yiğit Erkal, Emre Erciyas, İlhan Koç
<b>Objectives:</b> Testing the user interface and architectural structure.
<b>Tasks:</b> <b>Task 1</b> Testing the user interface of the application with JUnit. <b>Task 2</b> Finding and fixing bugs.

<b>WP4:</b> Launching the Application
<b>Start Date:</b> 12.05.2020 <b>End Date:</b> 19.05.2020
<b>Leader:</b> Utku Gökçen <b>Members Involved:</b> Utku Gökçen, Yiğit Erkal, Emre Erciyas, İlhan Koç
<b>Objectives:</b> Getting the application ready to work and uploading it to the Google Play Store.
<b>Tasks:</b> <b>Task 1</b> Preparing well-documented source code <b>Task 2</b> Preparing the website. <b>Task 3</b> Publishing the app on the Google Play Store.

#### 4.4. Ensuring Proper Teamwork

We have several paths to follow to ensure proper teamwork. First of all, the tasks to be done according to a certain plan will be distributed equally to the people in this group project. Next, we'll use some collaborative applications and management tools like Github. Since Github allows us to see what each member is doing, we can follow the work done here and combine these works in a common place. In addition, thanks to Google Docs, we are able to monitor the joint work of each team member and the preparation of reports. Zoom or Discord meetings also allow us to further our work and discuss these studies.

#### **4.5. Ethics and Professional Responsibilities**

Foodie is a protected application that stores and manages users' personal information in its database. Home and work addresses, e-mail addresses, phone numbers, credit card and identity information are examples of this data. Responsibility for the protection of this data rests with us, as the application owners, in accordance with the personal data protection law. Such data is never shared with other parties. However, we strive to provide protection to avoid cyber attacks. Therefore, our most important ethical and professional responsibility when developing this application is to provide an error-free application that will work as intended.

#### **4.6. Planning for New Knowledge and Learning Strategies**

In order to implement our application and develop it in a quality way, we need strong knowledge about the Android operating system. Since we will give our first priority to developing the application on Android, we have decided to use Android Studio and Java programming Language to learn this operating system for this purpose. Java and Android studio have become a widely used Software Development Kit (SDK) in recent years. Thus, there are many and varied resources in these programs as it has an active community that provides many libraries and data for different uses. We will also use some educational apps and websites such as Udemy and video resources.

## 5. References

- [1] <https://www.nrdc.org/resources/food-waste-restaurant-challenge-guide>
- [2] [https://gidanikoru.com/\\_uploads/G%C4%B1dan%C4%B1%20Koru%20Horeca%20K%C4%B1lavuz.pdf](https://gidanikoru.com/_uploads/G%C4%B1dan%C4%B1%20Koru%20Horeca%20K%C4%B1lavuz.pdf)
- [3] <https://toogoodtogo.co.uk/en-gb/blog/the-food-app-helping-you-save-the-planet>
- [4] <https://foodforall.com/>