IZMIR UNIVERSITY OF ECONOMICS
FACULTY OF ENGINEERING
SOFTWARE AND COMPUTER ENGINEERING

FENG 498 PROJECT REPORT

Restaurant Management System

Author(s):

Ahmet Berkay Kürkçü
Ali Doğan Güllü
Mursel Yetim
Umut Eren Sürücü
Yiğit Koçyiğit

Supervisor:
Asst. Prof. Gazihan Alankuş

OUTLINE

1. Abstract

2. Introduction

    2.1. Problem Statement
    2.2. Why Is This Project Worth Doing

3. Literature Review

4. Methodology

    4.1. Scope Of Work

    4.2. Tools

5. Risk Analysis

6. Test

7. Conclusion

8. References

1. Abstract

In our senior project, we designed and developed a comprehensive Restaurant System Application aimed at streamlining the dining experience for customers, waitstaff, and restaurant managers. Our software system encompasses web-based, mobile, and desktop applications, built using Flutter. This multi-platform approach ensures versatile access, allowing users to interact with our system in the most convenient way for them.

Our motivation was to address common challenges in the restaurant industry, such as ordering inefficiencies and payment complications. The application boasts several innovative features, including the ability for customers to leave feedback on their dining experiences and an integrated payment system. We've enhanced the overall customer experience and reduced the operational burdens on restaurant staff.

Throughout the development process, we adopted Incremental Model. This strategy involved breaking the application into sub-programs and progressively adding functionalities. By doing so, we managed the project's complexity efficiently and mitigated potential development risks.

This Restaurant System Application represents our vision for the future of the restaurant industry. By integrating the convenience of digital technology into the dining experience, we believe our system offers a robust solution for both customers and restaurant businesses, transforming the way they interact and conduct operations.

2. Introduction

In recent years, restaurants have become more and more popular. The typical procedure in a restaurant is for the customer to place his order and then wait for the food. The delay in fulfilling the customer's request is just one of several factors that contribute to this feeling of dissatisfaction. The emergence of new communication technologies can help to solve these problems.The limitations of traditional food ordering procedures are overcome with the use of our integrated, networked system. This program enables customers to conveniently order food from restaurants using their mobile devices

Customers can leave comments on dishes and read reviews from others to gain insights about the product. In addition, they have the opportunity to follow their favorite restaurants and view posts shared by these establishments. The posting feature of our restaurant application enables these businesses to reach a larger customer base. The app's user-friendly and dynamic structure also allows for quick and efficient menu updates.

In restaurants, customers do not want to be disturbed when they are consuming something.

They want to be complete and satisfied with their orders. Likewise, restaurants want their customers to be satisfied with their orders. To solve this, we decided to make this application. As we know, such applications have started to become widespread gradually nowadays. The main purpose of these applications is to reduce confusion, possible misunderstandings, and unnecessary workload in restaurants. Most applications in the industry can already do this. Our aim is to use these powerful parts in our application and to minimize the factors that will cause unnecessary trouble for the customer while using the application. In addition, adding the comments of the previous customers in the same restaurant about the meals to the application and the option to pay their bills directly through their phone.

We used an incremental model approach in our system. We constructed our application by parts and added functionalities step by step. First, we consider almost the entire system. Then, we divide the project into sub-programs. Each member of the project group worked on a different application. Because of this, management of the entire system became easier. This procedure reduced the risks and made testing easier. But for this approach to work properly, collaboration needed to be stronger and planning requirements more challenging.

2.1 Problem Statement

Originally, clients would choose their menu preferences and give them to the waiter, who would then take their order. He personally then delivers the meal item to the customer after bringing the order to the culinary department. Consequently, the process took a long time. It wastes paper and necessitates reprinting each menu card. Additionally, it is frequently not practical to keep printing all menu cards whenever a tiny modification needs to be made. simply stating that the menu card cannot be modified once it has been produced. After a few days, the menu card lost its respectable appearance and appeal. Also, in most of the digital menus, it only shows price information and content, they do not contain any feedback and comment section. However, reviews of any product are also very important for customers when online shopping nowadays. Similarly, we designed the system to be more interactive and social.

2.2 Motivation

Many restaurants handle their workflow and services manually and using paperwork, which requires a lot of management time and money. As people's schedules and values of their time grow more and more essential in today's busy world, Smart Restaurant manages the system of digital ordering and booking, saving the restaurant staff's time and manpower. It's simple to generate a KOT (Kitchen Order Ticket) by application, which helps you save time, money, and paper. Looking at the benefits of the Smart Restaurant System from the customer's point of

view, order waiting time has decreased. In addition, there will be no need to wait in queue or wait for a waiter to come to the table for payment, because of that, the online payment method is offered within the system

3. Literature Review

Nowadays, especially after the Covid-19 pandemic, lots of restaurants have started to use a digital menu system. Although the main purpose is to increase social distance, research has shown that a management system integrated with the digital menu also significantly reduces the delivery time of the order given by the customer (See Figure 1.a And 1.b).

Summary of Average Service Time of Waiter's Activity	
Activity	**Mins.**
Assist customer to table	0.83
Give Menu to Customer	0.97
Get Customer's Order	4.42
Give Customer's Order to Kitchen Staff	2.04
Give Customer's Order to Cashier	0.96
Give Customer's Order	6.13
Get Customer's Bill from Cashier	1.56
Give Bill to Customer	1.04
Get Customer's Payment	2.16
Give Receipt to Customer	1.63
TOTAL AVERAGE SERVICE TIME	**21.73**

Figure 1.a - Service time without application

Summary of Average Service Time of Waiter's Activity (mins) with RIS	
Activity	**Mins**
1. Assist customer to table	0.83
2. Give Customer's Order	6.13
3. Give Receipt to Customer	**1.63**
TOTAL AVERAGE SERVICE TIME	**8.58**

Figure 1.b - Service time with application

Due to the efficiency provided by the new generation digital menus, many restaurant management systems have been produced. These are most related previous studies:

- **Minimal.Menu**

  Minimal.Menu is a digital menu software that allows restaurants to design and alter menus from a computer, tablet, or smartphone. It creates and prints a basic QR code that may be displayed in restaurants and instantly scanned by consumers. Main features of the system are:

  - Creating a menu with different sections and products.
  - Sharing menu with a web address on social media, by email or directly access it via QR code.
  - Allow customers to browse your menu on their smartphone.

  Minimal.Menu is only a website system.

- **Orwi**

  Orwi is an application that offers food and beverage experience in restaurants, allows you to make orders, pay with Masterpass assurance, which gives you a continuous discount point. Main features of the system are:

  - Creating a menu with different sections and products.
  - Viewing and invoicing orders through the management website for waiters.
  - Ordering through the website or mobile application which are accessed with a QR code.
  - Making payments on the website or mobile application.
  - Ordering to the restaurant for takeaway delivery.

  Orwi has been produced to be used both on the website and on the mobile application.

- **Restaurant Information System (RIS)**

  RIS is a software that enables the customer to access the menu prepared by the manager in restaurants with a QR code and to place an order through the same system. Main features of the system are:

  - Creating a menu with different sections and products.
  - Viewing and calculating the prices of orders through the management part for waiters.
  - Ordering through the application which is accessed with a QR code.

C#, HTML, CSS, JavaScript and SQL are used in the implementation of RIS.

- **Our Solution**

  We implemented the main features in the mentioned projects. These main features are:

  1- Creating a menu with different sections and products.

  2- Viewing and invoicing orders through the management application for waiters.

  3- Ordering through the website which is accessed with a QR code.

4- Making payments on the website or mobile application.

5- Providing sales statistics for the manager.

6- Managing the social profile of the restaurant.

In addition to these features, we made the restaurant's menu interactive and social. We provided features so that customers could post comments, and other customers could see previous reviews. Moreover, we released separate applications for management, waiters, and customers. With this, we aim to reduce complexity and make the user experience much easier.

We used Flutter for the development of these applications on different platforms (Windows, Android, iOS, Web), and we made use of Firebase to establish the connection between them.

4. Methodology

In this project, we tried to adhere to the Incremental Build Model. It is a software development approach where the software is developed and delivered in small, incremental releases. This approach allows for a more flexible and responsive process, as well as early delivery of a working product. Because this project consists of three different sub-programs, we developed them individually and then we combined these sub-programs. We discussed daily to improve the features of our project and we visited restaurants to review existing products to find out how we can improve the digital menu. After those daily meetings, we added features to our program, then we tested and tried to find risks and bugs from these features.

4.1 Scope of Work

The scope of the work can be divided into three parts.

Customer:

- Customers would be able to check the restaurant menu with a web application.
- Customers could be able to pay to order in a web application.

- Customers could be able to comment about foods and restaurants with mobile application.
- Customers could be able to give feedback about foods.

Waitress/Chief:

- Waitress could be able to look at the order from the customer.
- Waitress could be able to accept and update order status.

Restaurant Admin/Manager:

- Manager could be able to change the menu via mobile application.
- Manager could able to view sales statistics via a mobile application.
- Manager could manage the social profile of restaurant.

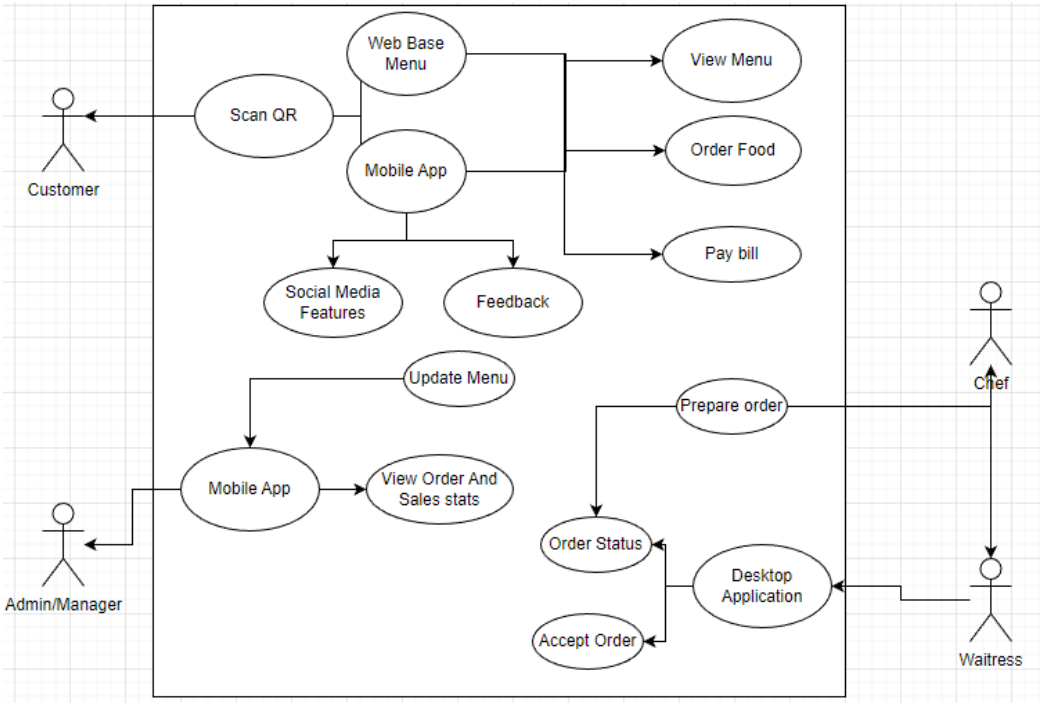

Figure 3- General Use Case Diagram

4.2 System Features and Usages

Mobile Application Sign-In Process:

The mobile application used by both restaurant customers and managers incorporates a secure and user-friendly sign-in process. Upon launching the app, users are prompted to sign in to access the system's features and functionality.

The sign-in process follows phone number verification with Firebase Authentication; to ensure the security of user accounts and system, the app requires users to provide their phone

number. Upon entering the phone number, the system sends a one-time password (OTP) to the provided phone number for verification purposes. Users then enter the received SMS OTP within the app to complete the verification process (See Figure 4.a). This step adds an extra layer of security and ensures that only authorized users can access the app.

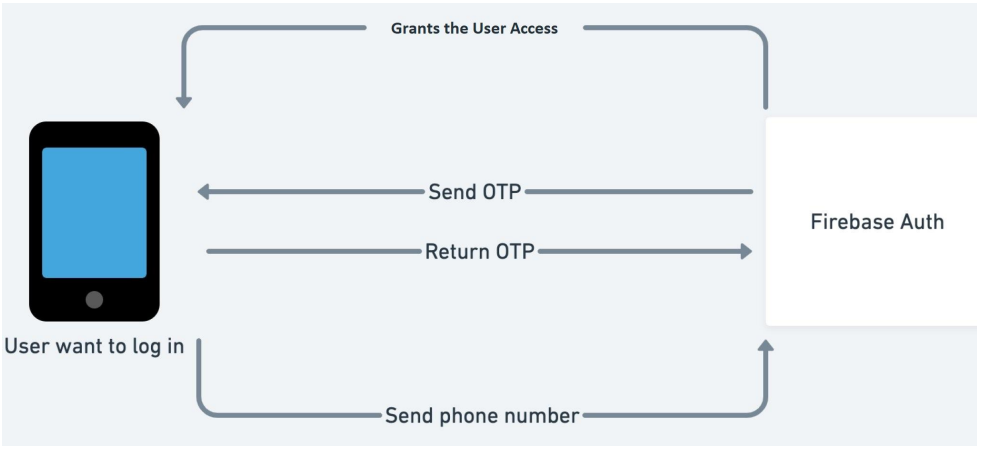

Figure 4. a- Flow Chart of Firebase Phone Number Authentication [6]

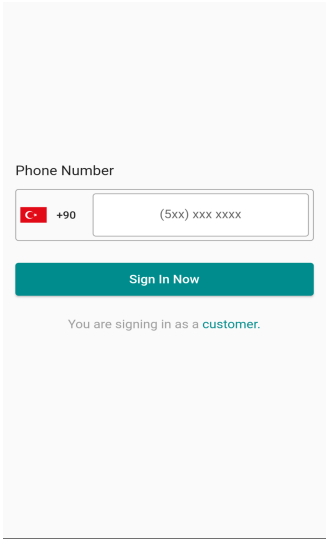

Figure 4.b- Sign In Screen

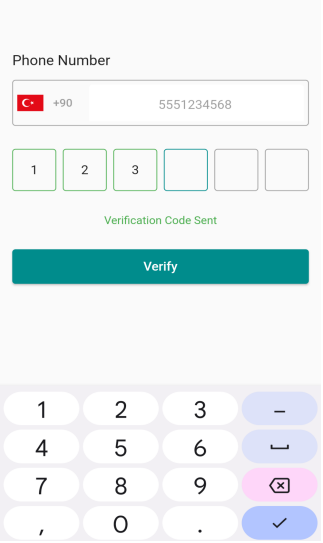

Figure 4.c- Phone Verification Screen

For New Users; If the phone number entered is associated with a new user, the app directs them to a sign-up screen. Here, new users are prompted to provide additional information, including their name and surname, to create a personalized account.We use Firebase Authentication libraries for this purpose (See Figure 4.b And 5.a).

For Existing Users; If the phone number entered is associated with an existing user, the app proceeds to the login process, where users enter their credentials (phone number and OTP) to access the app's features and personalized content. (See Figure 5.a And 4.c).

After the sign-in process, the application can store the state of a user signed-in or signed-out, also signed-in as a customer or as a manager.

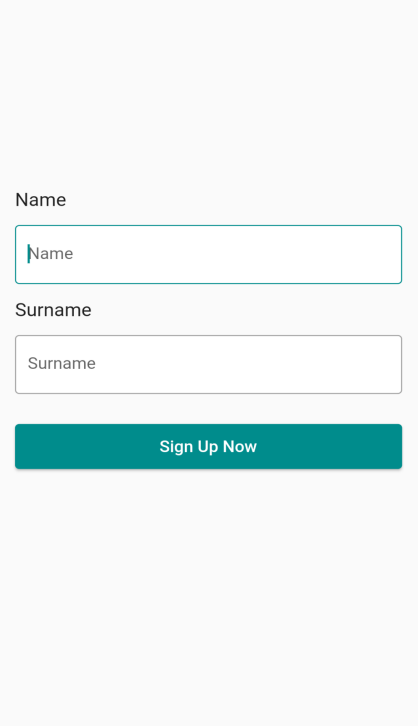

Figure 5.a- Name And Surname Sign-Up Screen

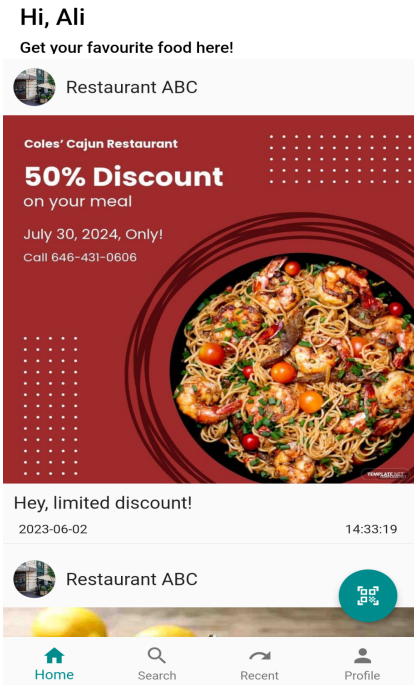

Figure 5.b- Home Screen

Restaurant Manager's Mobile Application:

A restaurant manager, upon opening the application, first needs to add a restaurant to manage on the management screen. Also managers can add multiple restaurants to manage and all the manageable restaurants visible at this screen. They need to input the necessary information for adding a new restaurant, including its name, location, image, wifi security option, and other managers.(See Figure 6.a And 6.b)

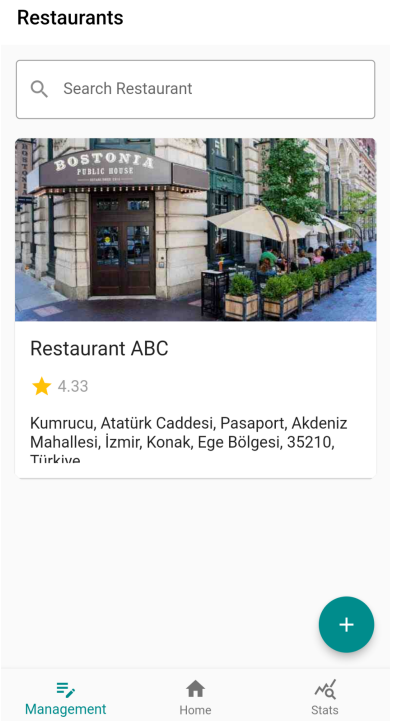

Figure 6.a.- Restaurant Search

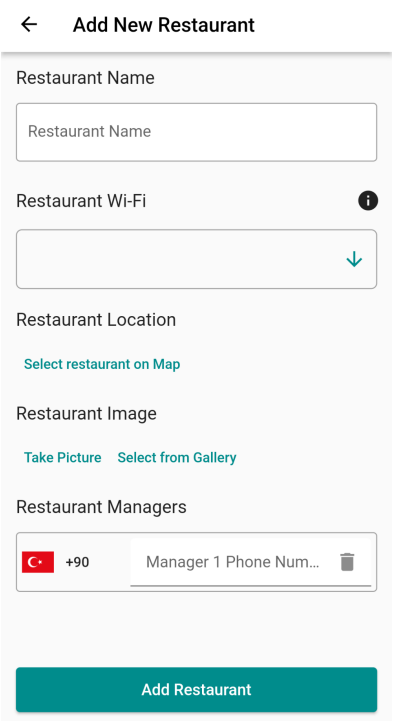

Figure 6.b- Adding New Restaurant

After creating the new restaurant profile, the manager designs the restaurant menu through the menu management feature. They can input various categories; then dishes with their descriptions, prices, and relevant photos with an option of sharing this new item as a post (See Figure 7.a And 7.b).

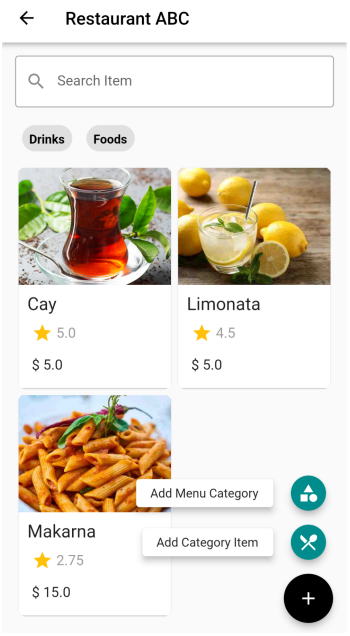

Figure 7.a- Restaurant Menu

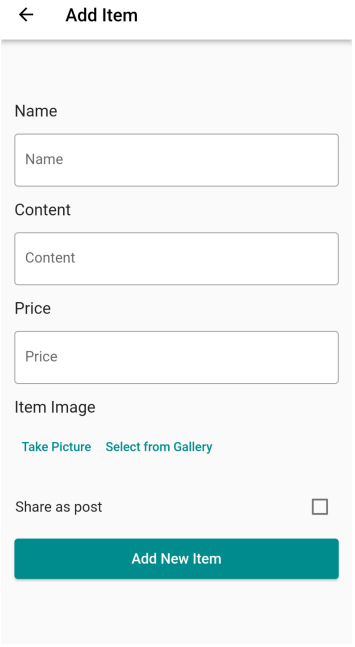

Figure 7.b- Adding New Item

With the menu set, the manager moves onto other features of the app. A stats section gives an overview of the restaurant's performance, including income, customer feedback, top rated products. By evaluating these insights, the manager can make informed decisions about the restaurant's operation and marketing strategies (See Figure 8.a And 8.b).

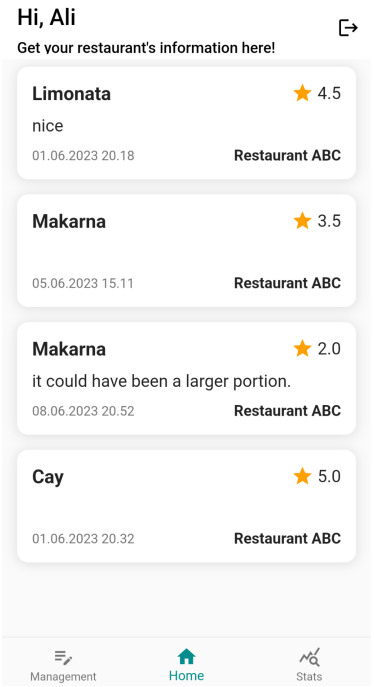

Figure 8.a- User Comments

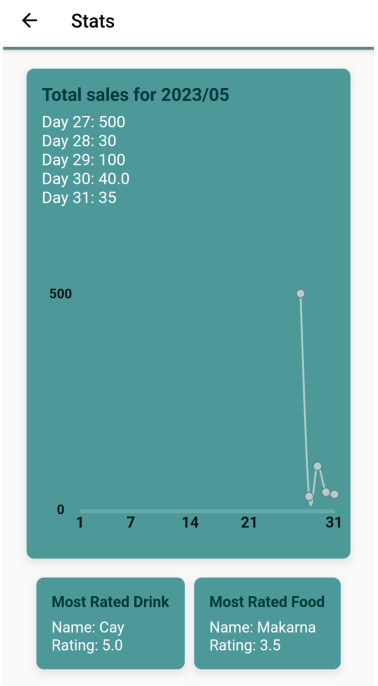

Figure 8b- Sale Stats

One such strategy could involve leveraging the app's social media integration. Here, the manager can create posts promoting popular dishes or special offers. This feature creates a digital bridge between the restaurant and its customers, fostering engagement and boosting visibility (See Figure 9.a And 9.b).

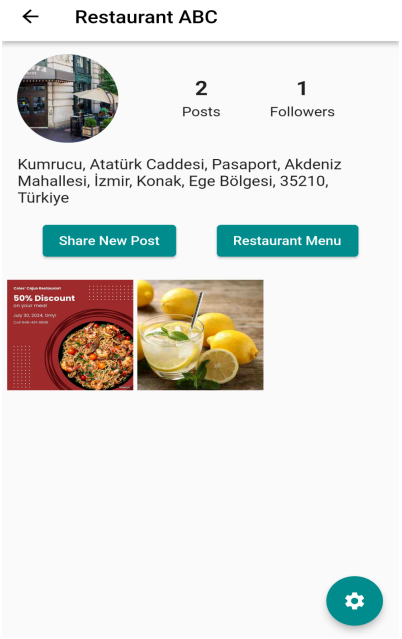

Figure 9.a- Restaurant Profile Screen For Manager

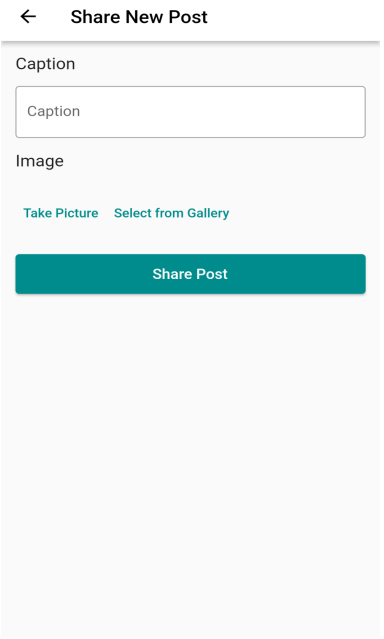

Figure 9.b- New Post Sharing Screen

Finally, managers can generate QR codes for each table through the app, QR codes contain a URL that includes the unique identifier for the restaurant and the corresponding table number. Manager can easily import this for printing. This feature enables customers to access the restaurant's menu and place orders directly from their mobile devices, enhancing the dining experience's efficiency and convenience (See Figure 10.a And 10.b).

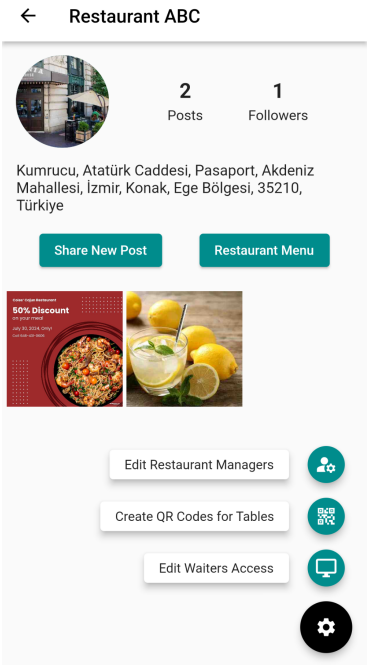

Figure 10.a- QR Code Generator Sceen

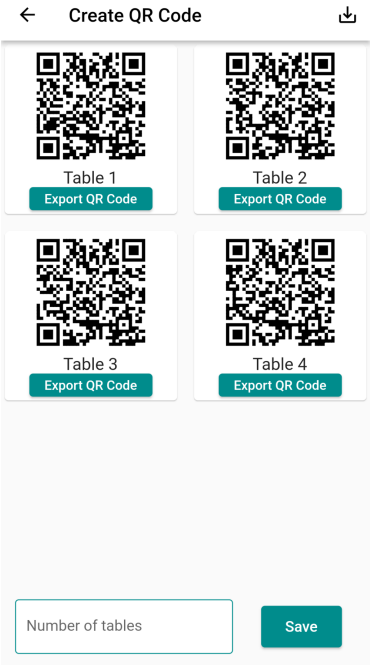

Figure 10.b- QR Code Management

Restaurant Customer's Mobile Application

A customer planning to dine at the restaurant opens the mobile app. Utilizing the app's search function, they explore nearby restaurants, looking for a specific ambiance or cuisine. They also see the various posts made by restaurants, learning about trending dishes, special offers, and customer favorites. (See Figure 11.a ,11.b And 11.c)

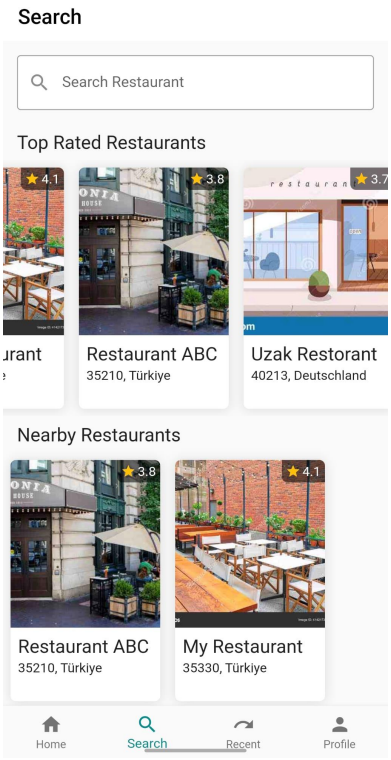

Figure 11.a- Restaurant Suggestion Screen

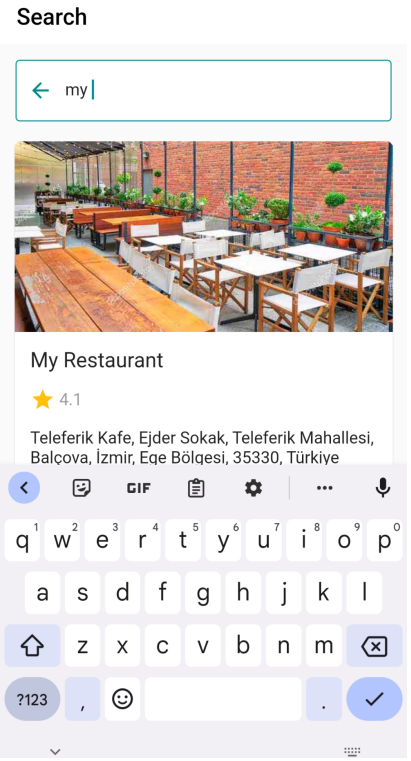

Figure 11.b- Restaurant Search Bar

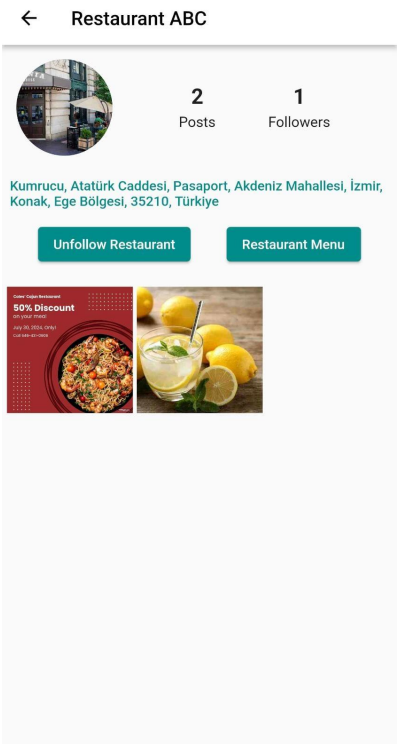

Figure 11.c- Restaurant Profile Page For Customer

Deciding on a particular restaurant, they arrive and use the app to scan the QR code (containing a URL that includes the unique identifier for the restaurant and the corresponding table number) placed on their table using Scanner on the homepage. Instantly, the restaurant's menu appears on their screen, with detailed information about each dish, including previous customer feedback. The customer selects their desired dishes and places an order directly from the app (See Figure 12.a, 12.b And 12.c).

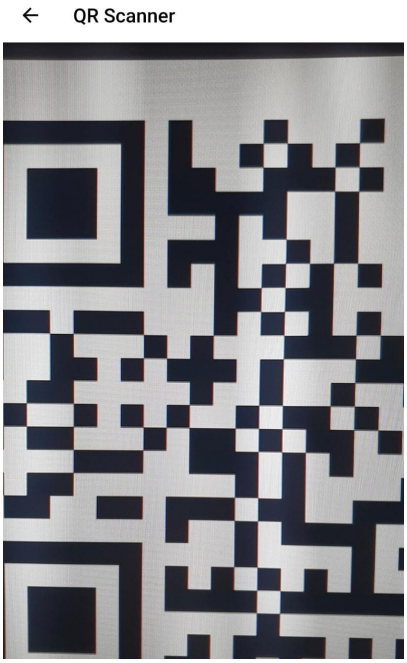

Figure 12.a -QR Code Scanner

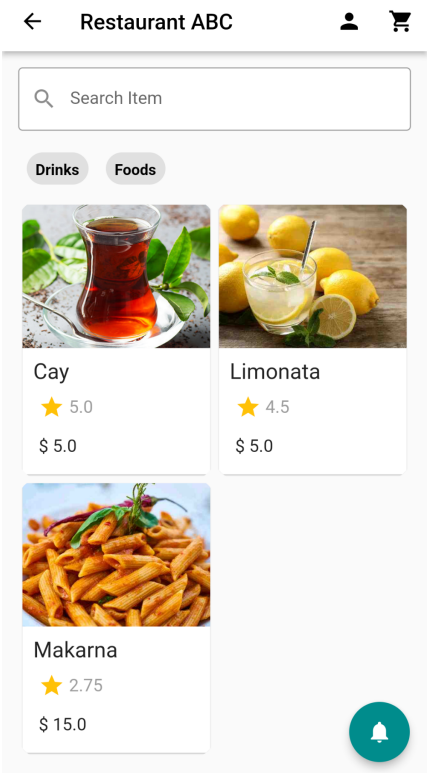

Figure 12.b- Restaurant Menu

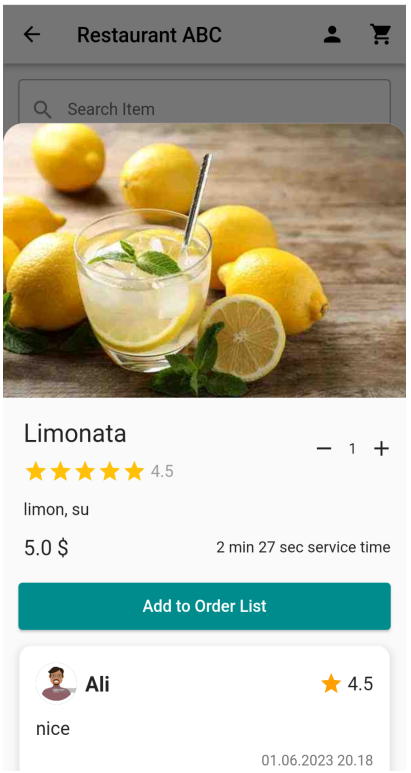

Figure 12.c- Detailed Item Screen

As they wait for their meal, they can check order status or estimated time, continue exploring the app, checking out other restaurants, and engaging with their posts. After enjoying their meal, they complete the payment process directly through the app (See Figure 13.a ,13.b ,13.c And 13.d).

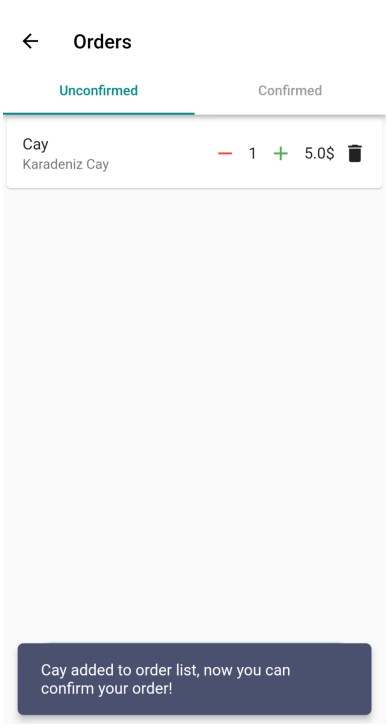

Figure 13.a- Unconfirmed Order Screen

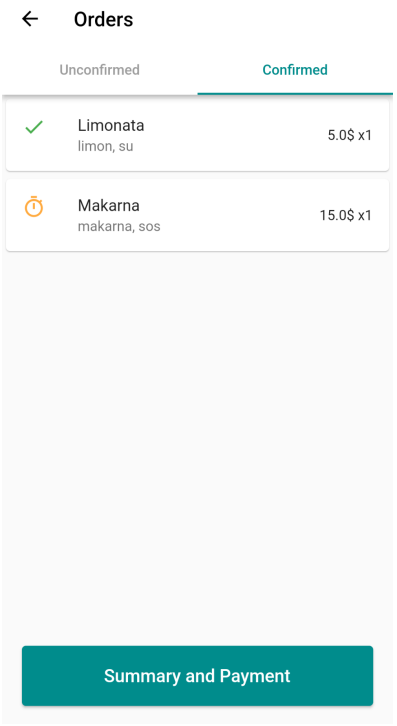

Figure 13.b- Confirmed Order Screen

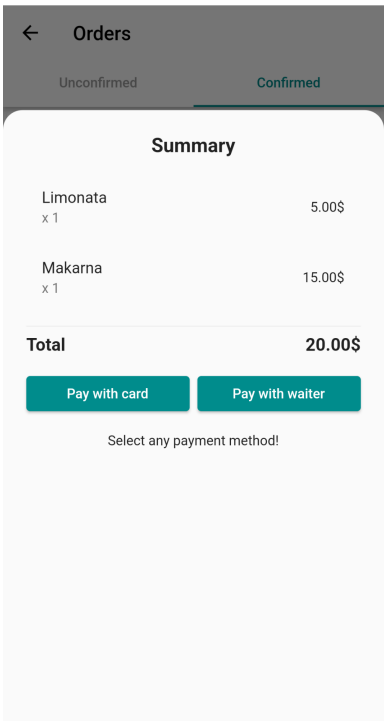

Figure 13.c- Payment Options

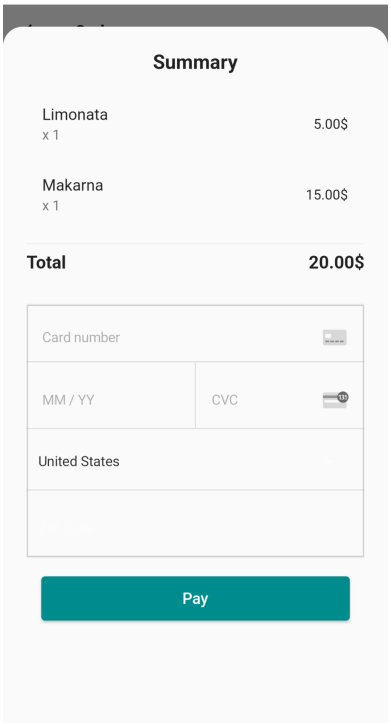

Figure 13.d- Payment With Credit Card

The customer then leaves their feedback as comment and as rating on the dishes they ordered on the Recent page, contributing to the community-driven review system within the app. Anytime customer can edit previous feedback on Profile page also this page provides opportunities for customers to edit their profile settings (See Figure 14.a ,14.b And 14.c).

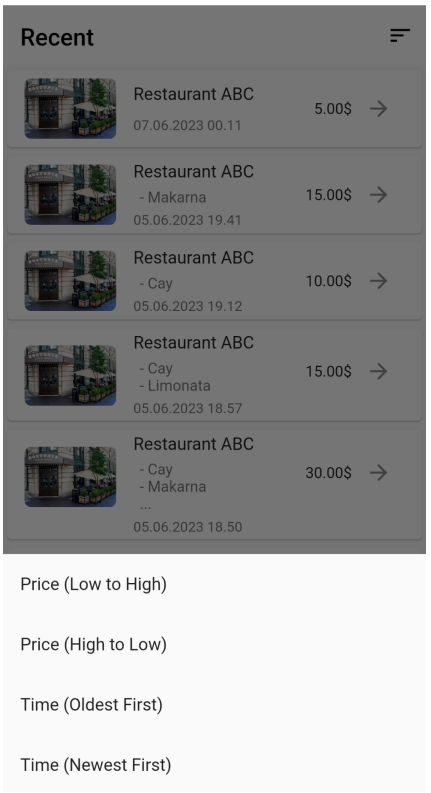

Figure 14.a- Recent Order Search Filter

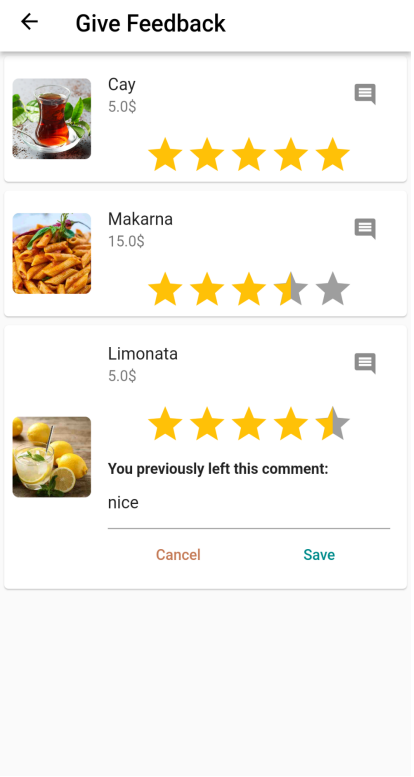

Figure 14.b- Item Feedback For Customer

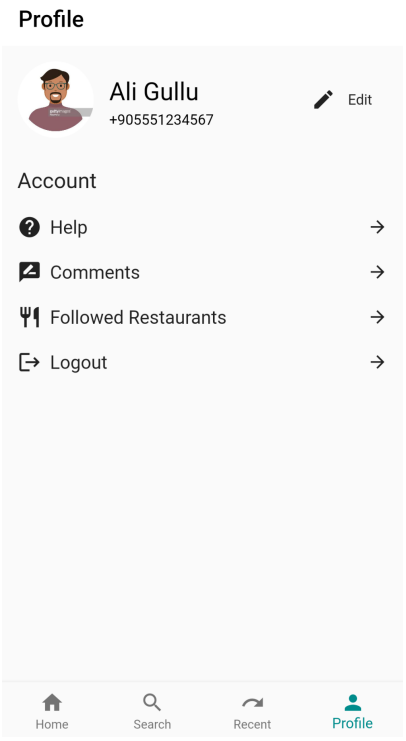

Figure 14.c- Customer Profile Screen

Restaurant Customer’s Web Application

At the same time, another customer can scan the restaurant's QR code (same QR with mobile application which contains a URL that includes the unique identifier for the restaurant and the corresponding table number) with any phone's camera, intending to place an order. The restaurant's menu displays on the web application. They can place an order, complete the online payment, the same way on mobile application without giving any feedback (See Figure 15.a ,15.b And 15.c).

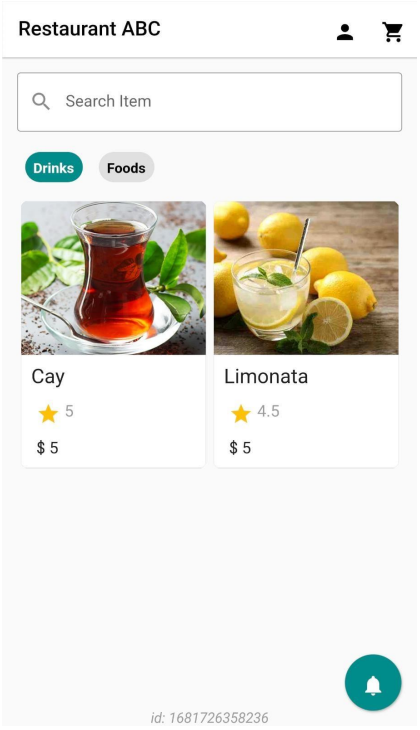

Figure 15.a-Restaurant Menu In Web

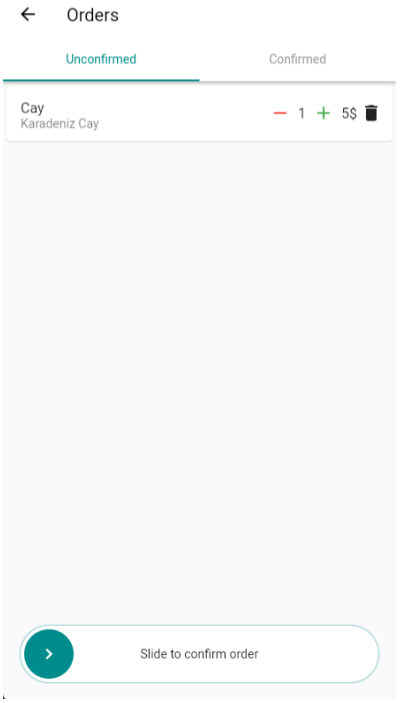

Figure 15.b- Unconfirmed Order In Web

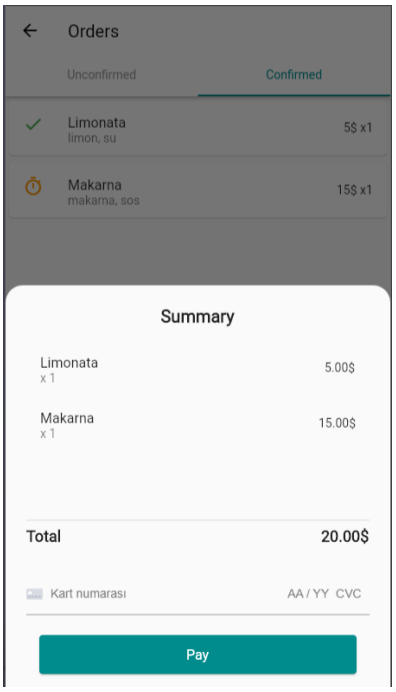

Figure 15.c-Payment Screen In Web

Waiter’s Desktop Application

At the restaurant, waitstaff can manage customers, customers' orders and receive notifications of restaurant’s tables on the desktop application. The system automatically assigns orders to the respective tables, and the waitstaff need only confirm and send them to the kitchen (See Figure 16.a And 16.b).

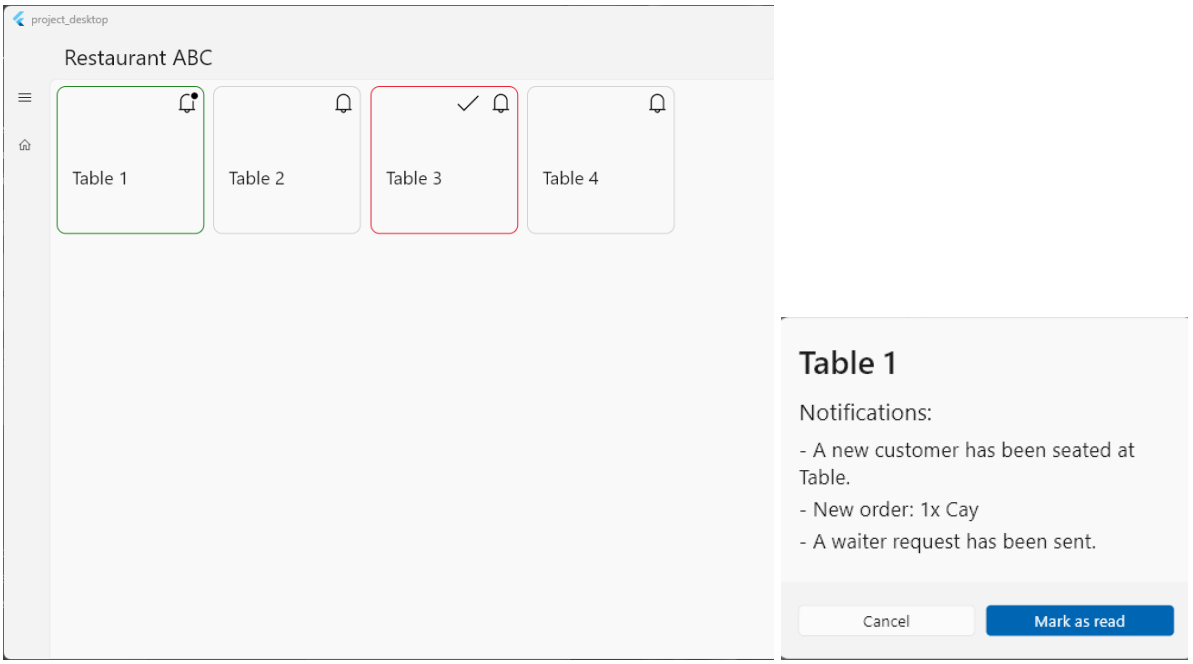

Figure 16.a- List of Restaurant Tables and Notifications

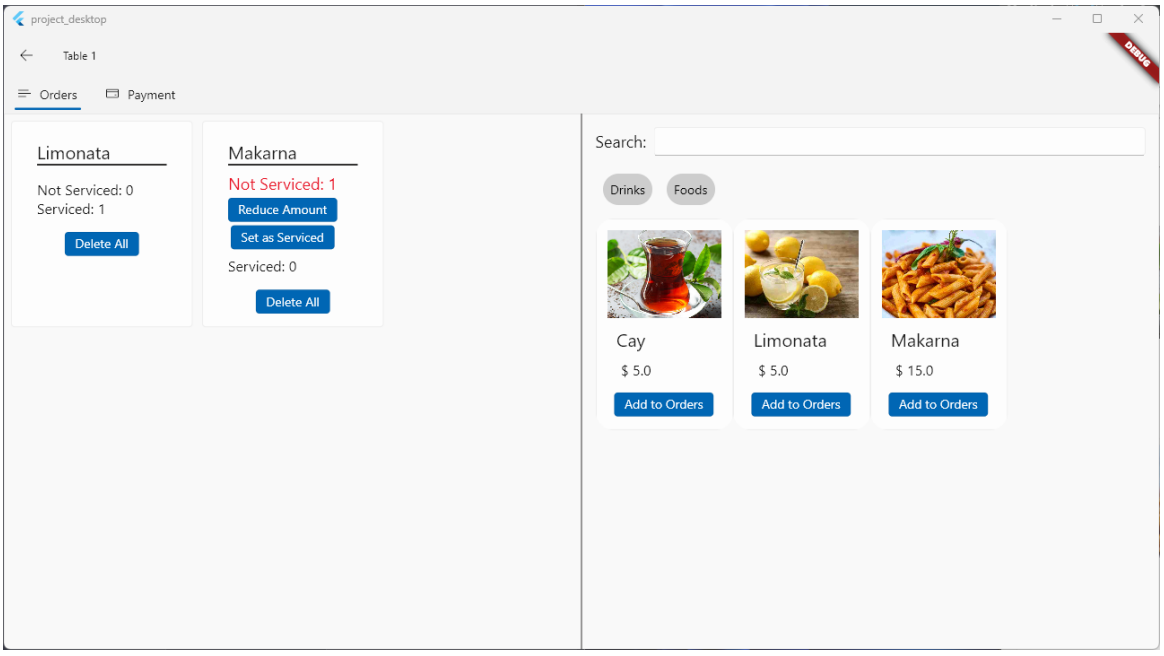

Figure 16.b- Customer Order For Restaurant

Once customers finish their meals, the system enables waitstaff to print bills with ease. As the payment process is handled through the app, customers receive only the receipt, making the entire process smooth and efficient or they can use traditional payment options like cash or POS (See Figure 17).

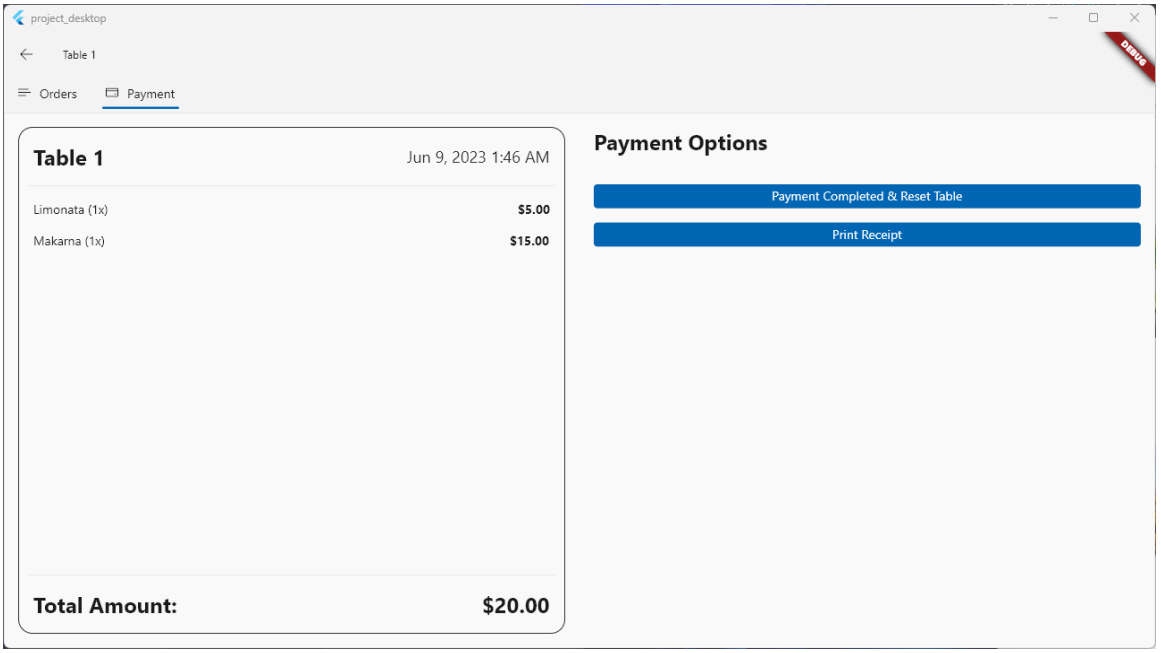

Figure 17- Bill Print Screen

Customer Payment Confirmation

After customers have successfully completed the payment process within the mobile application or web application, they are presented with a Payment Successful screen. This screen serves as a confirmation and acknowledgment of their successful transaction (See Figure 18).

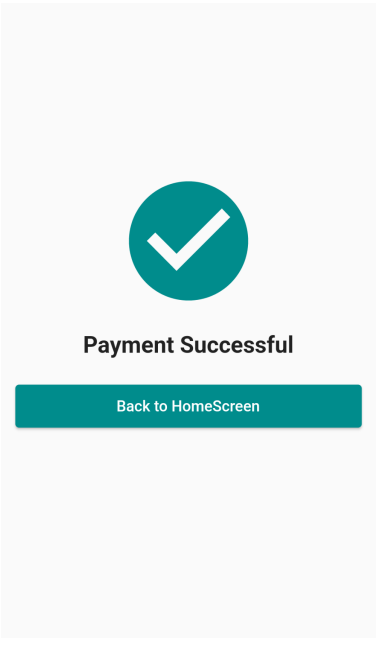

Figure 18- Payment Successful

4.3 Limitations

Naturally, this project has limitations.

The system is only intended for use in restaurants and cannot be used to order food outside of the restaurant boundaries.

The system only accepts credit cards or cash as payment methods, and it does not include other forms of payment like cryptocurrency or coupons.

The mobile application has some social media features like giving comments and scores to the restaurant, but it does not allow for adding other persons as friends or follow other people.

Like any software system, our solution requires regular maintenance, updates, and technical support to ensure its optimal performance and address any issues that arise. However, as this is a senior project, we can not manage these types of maintenance.

The feature implemented in our system, which accesses the nearby device's SSID and compares it with authorized SSIDs for security purposes, is implemented using the android_flutter_wifi package. However, it is essential to note that this feature is exclusively available on Android devices due to limitations imposed by Apple on iOS.

The System utilizes Firebase as its backend service. While Firebase offers various features and scalability, it also has some limitations, particularly when it comes to high usage levels. This must be reevaluated in case of excessive usage when the system is deployed.

The desktop application is only available for the Windows operating system, while the mobile application is specifically designed for Android and iOS platforms. Also, the web application is designed specifically for the aspect ratio of mobile devices.

We have used the esc_pos_printer Flutter package for only WiFi/Ethernet printers. The package has been tested on limited printer models with different results[7].

4.4 Tools

We established communication with the team through Slack, a messaging and collaboration platform. Slack provided a convenient and efficient way for us to communicate, share updates, discuss ideas, and address any project-related issues. It served as a central hub for team members and our supervisor to connect, collaborate, and stay updated throughout the development process. Additionally, GitHub and Git as essential tools for version control and collaboration were used for development of the system.

As mentioned this project consists of three parts. Web application, mobile application, desktop application. These three parts need to work together with the same database, and to achieve these; the Flutter framework was chosen for its flexibility and cross-platform capabilities and online database system Firebase was chosen because Firebase is compatible with Flutter also it has a lot of free features until certain use.

By using Flutter and Firebase, we were able to develop a unified application that works seamlessly across web, mobile, and desktop platforms from a single codebase. Additionally, to enhance the functionality and user experience of the application, various Flutter Packages were integrated. Some of these packages include:

firebase_core, firebase_auth, cloud_firestore, firebase_storage: Firebase packages used for user authentication, real-time data synchronization, database management, and cloud storage.

intl_phone_number_input, pinput, simple_speed_dial, flutter_rating_bar, slide_action, open_street_map_search_and_pick, fl_chart: These packages were integrated into the project to enhance the user interface and interaction elements. The intl_phone_number_input package handles phone number inputs by providing validation and formatting options for correct international formatting on sign-in sign-up and management screens. The pinput package allowed the implementation of a pin code input field within the sign-in signup screen. With the simple_speed_dial package, customizable floating action buttons were incorporated into the user interface. These buttons provided quick access to frequently used actions, enhancing the usability and convenience of the application for Managers. The flutter_rating_bar package facilitated the integration of a customizable rating bar widget. This feature allowed users to provide their ratings and feedback on dishes or their overall restaurant experience, enhancing user engagement and providing valuable insights for restaurant owners. The slide_action package enabled the implementation of slide actions within the application's user interface. This feature allowed customers to perform placing order actions. The open_street_map_search_and_pick package was utilized in the project to enable restaurant managers to select the location of a restaurant, package is a Flutter library

that integrates with the OpenStreetMap API to provide search and selection functionalities for locations. The fl_chart package is a Flutter library that provides a wide range of customizable chart options, including line charts, bar charts, pie charts, and more and it was utilized in the project to create interactive and visually appealing charts for the managers' stats screen.

shared_preferences: Stores and retrieves key-value pairs on the device for saving user preferences, session data, and other relevant information, such as storing user type locally for sign-in.

qr_flutter, mobile_scanner: These packages Generate and scan QR codes features, used for various purposes such as table identification and order tracking.

image_picker, image, path_provider, flutter_image_compress, cached_network_image: Packages for efficient image handling, including image selection, compression, and caching of network images.

geolocator, location, safe_device, android_flutter_wifi: Packages for the application incorporate security measures by checking device security parameters and validating user location, respectively. These features contribute to a more secure and reliable experience for both users and the system. In our system, web and mobile applications include a location verification feature. When a customer places an order, the system receives the user's location to verify that the customer is indeed in the restaurant. This can be achieved in Flutter with the use of a geolocator package, which allows access to the device's location and its use within the app. To check if the location is correct, the restaurant location can be set as a reference point and compared with the location received from the user's device. If the user's location is within a certain distance of the reference point, the system verifies that the user is indeed in the restaurant and allows the order to be placed. By adding this feature, we can ensure that only customers physically present in the restaurant can place orders. Additionally, checking for fake locations is important to ensure that the user's device isn't providing a fake location using spoofing applications. We added this using the safe_device package. This ensures that the location data used by the application is accurate and can't be manipulated by malicious actors, making the system more secure and reliable.

path_provider, share_plus, archive: Packages were utilized in the project to handle file paths and enable content(for our system contents are QR Codes) sharing functionality on QR Code Creating screen of Managers.

geoflutterfire2: Package was used in the project to implement geospatial querying and

enable location-based features, specifically for finding nearby restaurants on the search screen.

http, url_launcher: The http package provides convenient methods for making HTTP requests from the Flutter application. The url_launcher package allows for opening URLs in the device's default browser or other related applications. These two packages were utilized in the project to open the related map application based on the restaurant's address.

flutter_stripe, flutter_bloc, flutter_paypal_checkout: These packages were utilized in the project to enable secure payment options for customers on both mobile and web platforms. The flutter_stripe package provides integration with the Stripe payment gateway, enabling secure payment processing within the application. Stripe is a popular payment platform that offers a robust set of APIs and tools for accepting and managing online payments. The flutter_bloc package is a state management library based on the BLoC (Business Logic Component) pattern. It provides a structured approach to manage the application's state and handle complex business logic. The flutter_paypal_checkout package offers integration with the PayPal payment gateway, allowing customers to make secure payments using their PayPal accounts. PayPal is a widely recognized and trusted platform for online payments but we have huge restrictions to test this in Turkey.

By harnessing the capabilities of these packages and integrating them into our Flutter-based Restaurant System Application, we created a solution that addressed the needs of both customers and restaurant staff.

5. Risk Analysis

There are several risks that can occur in this project, down below we identified possible problems and solutions. Since we are considering three separate applications, some applications have limitations in their implementation. Due to these limitations, we offered more than one solution to a problem and used the most suitable solution for implementation.

- We are using an online database (Firebase Firestore) so there might be connection issues, with regards to that connection may down. As a solution, we are going to use Firebase offline database support when the system is deployed, which will be updated at certain time intervals.
- For the system to work integrated with each other, we had to connect a single online database to more than one application. We solve this problem using Firebase.
- One of the main risks associated with this project is the potential for security breaches. To address this concern, we implemented multiple layers of security measures, including the use of Firebase Rules. These rules are designed to restrict data access to authorized users only. However, we recognize that relying solely on authorized checks may not be sufficient. Therefore, we have taken additional steps to enhance security by implementing different sets of rules for web users, authenticated managers, authenticated customers, and authenticated waiters. So, we can ensure that each user has limited access to the database based on their respective privileges. This approach helps mitigate the risks associated with unauthorized actions within the database.
- Given the potential complexity of the user interface (UI) that may hinder user usability, we propose a solution where we actively gather feedback from users. Based on this feedback, we will make necessary changes to the UI to enhance its user-friendliness and address any usability issues.
- We designed applications for more than one platform, for that reason the user interface may cause problems on some devices, we can find a solution by using the responsive UI features provided by Flutter.
- To mitigate the risk of malicious individuals (hackers, etc.) placing orders without being in the restaurant, we added location verification to our web and mobile applications. If a customer wants to place an order, the system will receive the user's location and verify that the customer is indeed in the restaurant.
- Malicious individuals can trick the location verification system, so to solve that problem, we added mock location (fake location) detection to our mobile applications.
- Malicious people can trick location verification, so to solve that problem on mobile application, the application can scan the restaurant's Wi-Fi hotspot so that people can be detected within the certain area. This solution can be used optionally as it requires extra hardware in the restaurant and it uses the SSID of

the device to verify that the customer's device is in the vicinity of the restaurant's Wi-Fi network. By comparing the SSID of the device with a list of authorized SSIDs , the system can confirm that the device is near to the restaurant's network, thereby providing an additional layer of security. This feature is implemented using the android_flutter_wifi package that grants access to the device's SSID and allows comparison with the authorized SSIDs. However, it is important to note that due to limitations imposed by Apple on iOS, this feature is exclusively available on Android devices. On iOS devices, this security measure is automatically bypassed.

- To ensure that there is no unauthorized access to an activated restaurant table, our system includes an algorithm that there is no unauthorized access to an activated restaurant table. The algorithm works as follows:

    1. When a customer arrives at a table, they scan a QR code (contains url that include restaurant id and table number) that is placed on the table, this QR code activates the table.
    2. The initial person who scans the QR code at the table is granted privileges, such as the ability to approve access of other customers (See Figure 19).
    3. Subsequent individuals who arrive at the table will only be able to place orders if the initial person grants them access to do so. This can be done by a minimal pop-up alert that asks for new customer confirmation.
    4. Once the initial person grants access, subsequent individuals will be able to view the menu, place orders, and make payments as part of the group.
    5. This algorithm ensures that only authorized individuals have access to an activated table, preventing unauthorized access and protecting the security and privacy of customers.

    This feature adds an extra layer of security, privacy, and flexibility to the ordering system, enabling customers to have full control over their order and who can join their table.

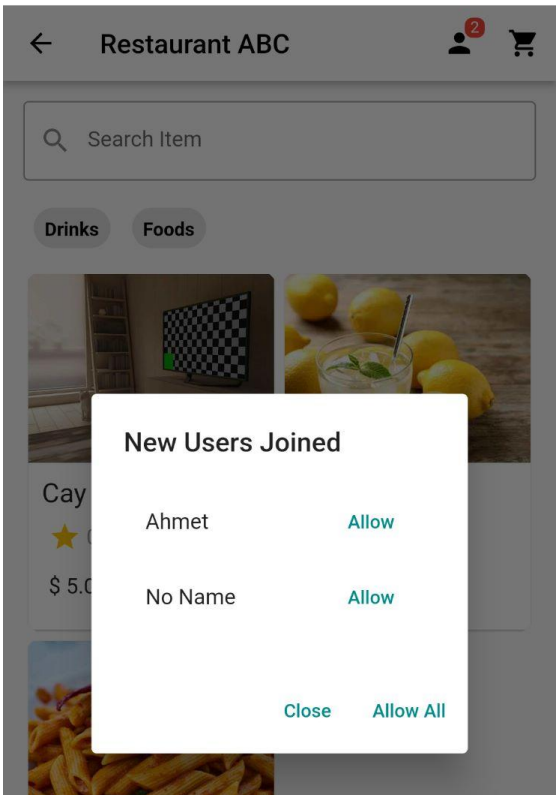

Figure 19- Giving Permission to New Customer who Join Table

- Malicious individuals might use fake locations to trick the location verification system. To address this problem, we added a manual order option to the web application which is more unsafe, as mentioned in Figure 20. Customers will only be able to place orders when the waiter opens the table for ordering from the desktop application (See Figure 20).

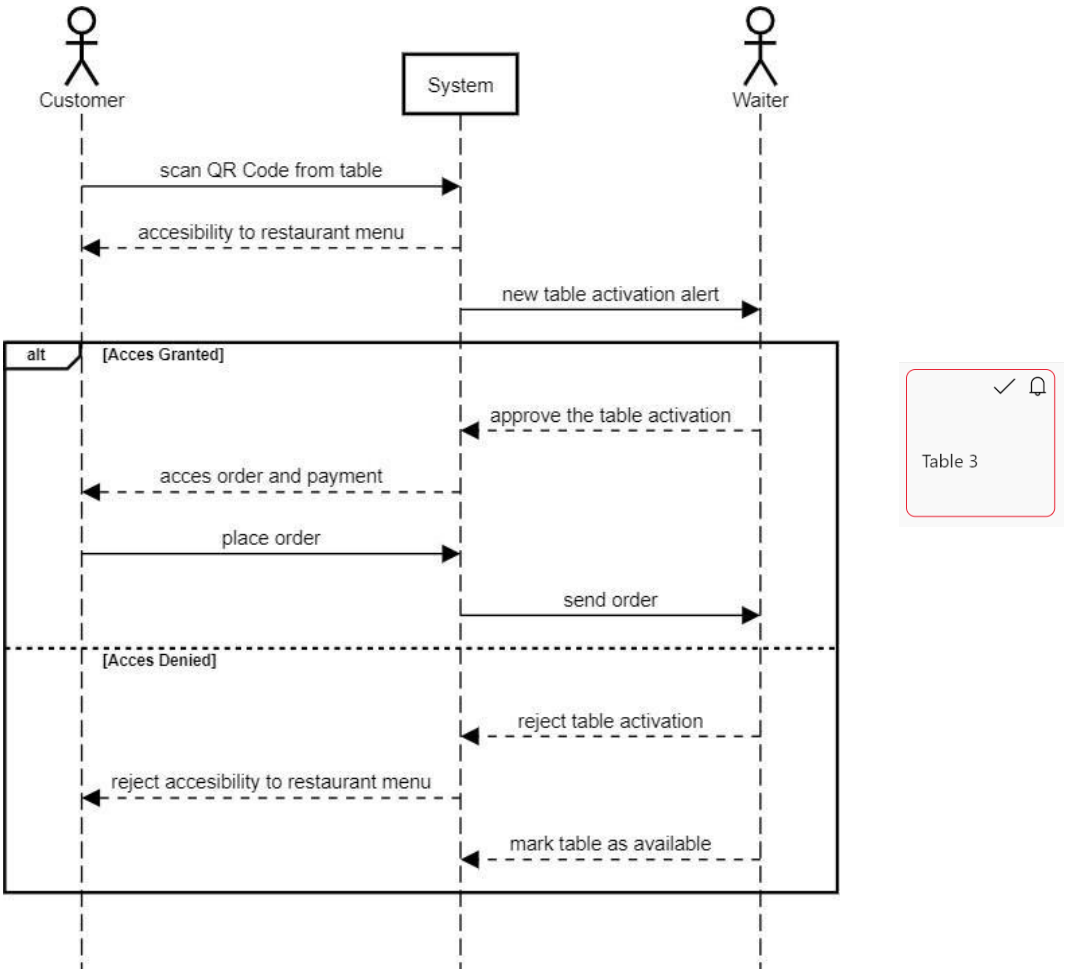

Figure 20- Sequence Diagram of Web Menu Manual Waiter Validation & Related UI on Desktop Application

6. Tests

The system underwent a comprehensive testing process to ensure its functionality, usability, and compatibility across different devices and platforms. The following tests were conducted to evaluate the performance, security and reliability of the system.

6.1 Real Device Testing

Real device testing was performed to assess the system's behavior and performance on actual mobile devices and desktop computers. The system was tested on various devices, including smartphones running different operating systems (Android and iOS), as well as desktop computers running Windows. This testing approach allowed us to identify any device-specific issues and ensure a seamless user experience across different platforms.

6.2 Web Page Testing

Web page testing was conducted to validate the system's functionality and user interface on different web browsers after deploying on Firebase. The system was tested on popular mobile browsers such as Google Chrome, Safari, and Samsung Browser. This testing ensured that the web pages rendered correctly, and all features and functionalities worked as intended across different browsers.

6.3 Commit Testing

Throughout the development process, commit testing was performed to ensure the stability and integrity of the codebase. Developers performed unit tests, integration tests, and regression tests on each code commit to identify and resolve any potential issues early on. This iterative testing approach helped maintain the reliability and quality of the system's codebase throughout the development lifecycle.

6.4 Stripe Payment Testing

Stripe payment integration was thoroughly tested in developer mode to ensure the successful processing of payments within the system. In developer mode, a test scenario was executed to simulate a successful payment transaction. The purpose of this test was to ensure that the system accurately communicates with the Stripe API and handles the payment process without any issues. It was verified that the payment was successfully processed, and the corresponding amount was correctly charged to the customer.(See Figure 21)

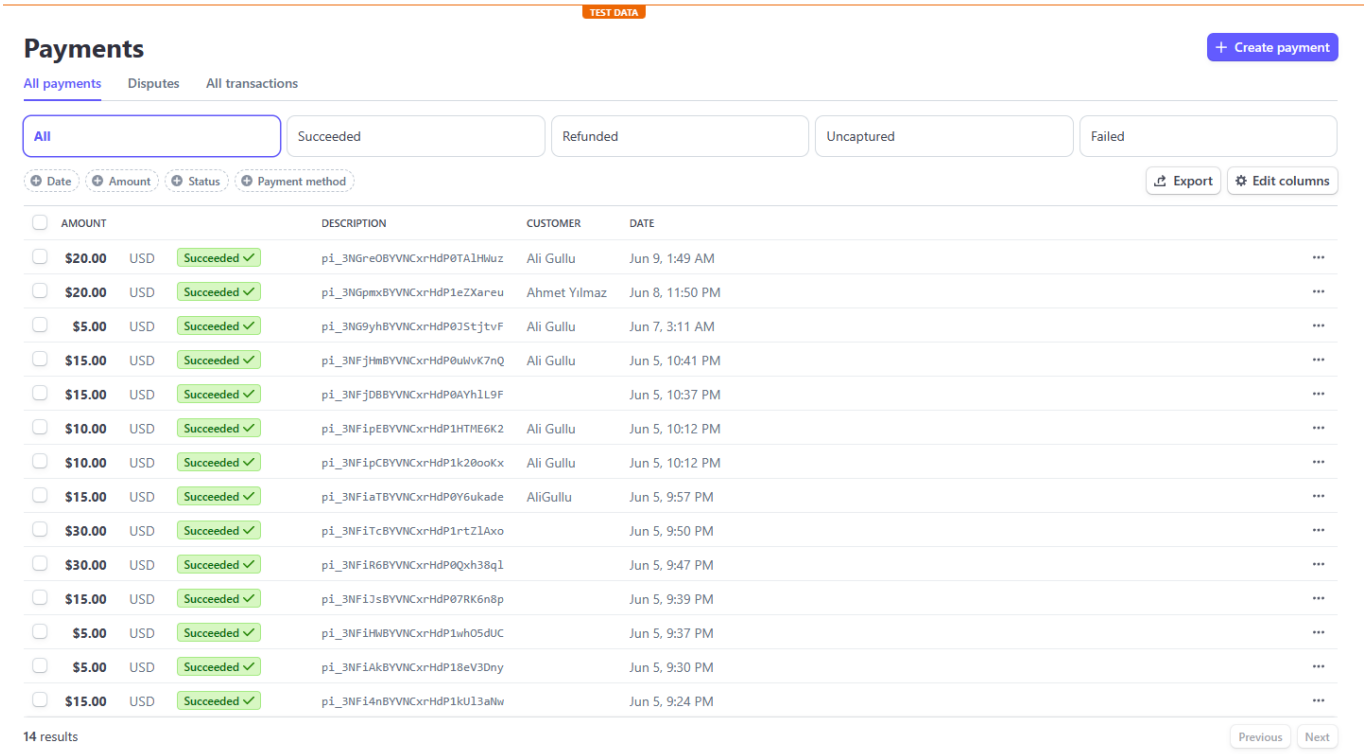

Figure 21- Stripe Payment Testing

7. Conclusions

In this project, we proposed a Restaurant Management Application that aims to improve the customers, waiters and managers experience in restaurants by providing a convenient and efficient way to order, pay and review for food using a web-based, mobile, and desktop application. The project team planned to use the Flutter to develop the application as it allows for building, testing, and deploying mobile, web, and desktop apps from a single codebase. The application was also designed to include features such as customer reviews and the option to pay directly through the application.

The team used Incremental Model in the development of the system, where the application was constructed in parts and functionalities were added step by step. This approach allowed the team to manage the project efficiently and reduced risks associated with the development process.

In the future, to improve the results of this project, the team could consider incorporating additional payment methods, and allowing for ordering food outside of the restaurant boundaries. Additionally, we are only planning a social media with commenting on past orders, but they could also include more social media features in the mobile application.

8. References

[1] Grace L. I. & Jan D. P. & Louise M. F. & Blanca M. D. (2020). Restaurant Information System (RIS) with QR Code to Improve Service Operations of Casual Fine Dining Restaurant, Retrieved November 15. 2022 from https://www.researchgate.net/publication/341691351_Restaurant_Information_System_RIS_with_QR_Code_to_Improve_Service_Operations_of_Casual_Fine_Dining_Restaurant

[2] Minimal Menu. Retrieved November 15, 2022 from https://minimal.menu/en

[3] Orwi Application. Retrieved November 15, 2022 from https://orwi.app/

[4] Flutter development framework. Retrieved November 15, 2022 https://flutter.dev/

[5] Stripe Payment Method. Retrieved June 9, 2023 https://www.stripe.com

[6] Firebase Authentication Diagram Retrieved June 12, 2023 from Diagram

[7] Tested printers for esc_pos_printer Flutter Package Retrieved June 12, 2023 from Tested Printers