Palestine Technical University-Kadoorie

College of Engineering and Technology

Department of Computer Systems Engineering

Course name:

Introduction to Graduation Project

Project title:

DEVELOPING PUBLIC TRANSPORTATION BOOKING APPLICATION IN BEIT LEED

By:

Mahmoud Draidi - 201910433

Muath Khasati - 201910283

Obada Shobaki - 201911078

Supervisor:

Dr. Rami Yousef

Tulkarm, Palestine

13 June 2023

DEDICATION

To our families who have never failed to give us financial and moral support, giving all our needs during the time we developed our app and teaching us that even the largest task can be accomplished if it is done one step at a time. We dedicate this Project to all the people who have worked hard to help us complete the project.

ACKNOWLEDGMENT

First and foremost, we would like to thank the supervisor of our project, Dr. Rami Yousef, for his support, outstanding guidance, and encouragement throughout our senior project.

We would also like to express our gratitude and appreciation to Dr. Yazeed Sleet and Dr. Yousef Daraghmi for the help and guidance they provided throughout our education. Your useful advice and suggestions were really helpful to us during the project's completion. In this aspect, we are eternally grateful to you.

Many thanks to our friends and colleagues at our university.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABSTRACT

The project is simply an application for booking seats on seven-seat buses available in Beit Leed. Organizing public transportation is very helpful to all people living in the area, especially for saving time, fuel consumption, and money. The app provides a system to manage the list of available drivers and their turns so passengers can book properly. The booking process includes a specific time, place, and number of seats as book options for passengers to reserve their seats. In addition to booking, the app gives information to passengers to assist them in traveling as well as to drivers to assist them at their work. The app has two different interfaces depending on the type of user: the first is for the driver, allowing him to control his bus seats and access special features, and the other is for passenger to reach the booking process and app features.

الملخص

المشروع ببساطة عبارة عن تطبيق لحجز مقاعد في الباصات العمومية المتوفرة في بيت ليد. يعد تنظيم وسائل النقل العام مفيدًا جدًا لجميع الأشخاص الذين يعيشون في المنطقة، وخاصة توفير الوقت واستهلاك الوقود والمال. يوفر التطبيق نظامًا لإدارة قائمة السائقين المتاحين وأدوارهم حتى يتمكن الركاب من الحجز بشكل صحيح. تتضمن عملية الحجز وقتًا ومكانًا محددين وعدد المقاعد كخيارات للركاب لحجز مقاعدهم، والتي يتم تنظيمها من خلال الادوار. بالإضافة إلى الحجز، يقدم التطبيق معلومات للركاب لمساعدتهم في السفر وللسائقين لمساعدتهم في العمل. يحتوي التطبيق على واجهتين مختلفتين حسب نوع المستخدم: الأولى للسائق تسمح له بالتحكم في مقاعد الحافلة الخاصة به والوصول إلى خصائص مميزة في التطبيق والواجهة الثانية للركاب للوصول إلى عملية الحجز وخصائص التطبيق.

CHAPTER ONE: INTRODUCTION

1.1 Background

People in Beit Leed use public transportation to go to the Tulkarm governorate. They ride available seven-seat buses for this transportation line. The app in Spring 2023 provides a booking system that people can use to book a seat on the bus at a specific time. However, each bus follows the turn to take passengers at specific times, and the app organizes the turn between available buses.

In peak times, available buses can't handle all the number of passengers who go and waste their time waiting at the bus station, so the app saves people time by showing there are no available buses at that time. On the other side, buses drivers waste money and time searching for and collecting passengers. The app shows the driver the location and number of waiting passengers or passengers that have already booked their seats on his bus.

1.2 Literature Review

Through our search, we found Plenty of applications that organize and give different services for transportation. One of them was Grap which provides services to connect passengers with private hire, taxi, and coach drivers (Tan&Ling, 2012). In our application, we provide services specifically for available public transportation in Beit Leed including only buses instead of private transportation.

Another one is Uber app, which is the biggest private transportation application around the world, it's much easier than taxis and quicker too. You can also choose the type of vehicle to fit your needs. The app is simple to use and gives you up-to-date information on where your driver is and when they are there to pick you up (Khosrowshahi, 2009). The only problem is that you can't pay with cash and sometimes that's all you got or can get. In our application, we look to provide all features just like Uber, but to public transportation, in which all the cars are identical, and you can pay cash to our drivers.

In the Arabic world, we have Careem company, a Dubai-based super app with operations in over 100 cities, covering 12 countries across the Middle East, Africa, and South Asia regions. The company, which was valued at over US$2 billion in 2018, became a wholly-owned subsidiary of Uber after being acquired for $3.1 billion in January 2020. Careem expanded into the food delivery business with Careem Now in November 2018 and launched a digital payment platform, Careem Pay in April 2022. Careem said they would be launching bus services, starting with cities in Egypt in December of the same year. The service was discontinued in early 2020 (Sheikha et al., 2012).

Locally, there is no app but (مواصلاتي) app which was developed for public transportation, it doesn't organize transportation like our app but is used for Driving license renewal (Palestinian Ministry of Transportation and Communications, 2021). As a result, our app comes with new services that solve our local transportation problems.

Apps	Payment ways	Trak the driver by map	Chat with drivers	Rating system for drivers	Public / Private
Uber	Credit card only	✓	✓	✓	Private
Careem	Cash and credit card	✗	✗	✗	Private
Grab	Credit card only	✗	✗	✗	Private
Our app	Cash only	✓	✓	✓	Public

1.3 Objectives

- To organize public transportation under the supervision of the Palestinian Ministry of Transportation.
- To make the operation of booking a seat on a seven-seat public transportation bus easier and available in any possible case (from the garage, home, etc...).
- To reduce the traveling time from the garage to the destination And vice versa.
- To solve the problem of traffic produced by the lack of efficiency in collecting people from their waiting spots.
- To increase the total wage of the drivers significantly by reducing costs resulting from fuel consumption, wasted time, etc...

1.4 Motivations

- Time and cost savings for people.
- Passengers waiting time takes more than the traveling time.
- Drivers don’t know any information about traveling passengers and they work randomly.
- Every single one in Beit Leed needs the service and can easily use the app.

1.5 Problem Identification

Passengers struggling with when and where they can use public transportation buses. They are stuck in traffic and can't handle all the traveling passengers, who have to wait long periods until the buses become available. On the other side, drivers faced the same situation in how they waste long periods collecting the passengers who are distributed in several spots. Therefore, the financial costs increase. Moreover, drivers are responsible for booking the passengers.

At peak times most buses exist only in one location, the other side is full of passengers. Buses don't return to that location because of the lack of passengers in their current location, they lose money when returning empty seats, then passengers are stuck in one location. So, providing a way to organize getting into public buses and getting out of them is a demand to get a partial solution to this problem. So, our application is responsible for taking a step forward to help both passengers and bus drivers have better traveling.

1.6 Research Questions

1. How does the system affect fuel consumption?
2. How much time is saved for both drivers and passengers?
3. How effective is this system in available public transportation?

CHAPTER TWO: METHODOLOGY

2.1 Process Model

Our application is designed for organizing transportation basically between Tulkarm and Beit Leed. Both drivers and passengers can use the application to get the best experience during their transportation.

Agile processes were used in this project. Both drivers and passengers are involved in the project's development. Changes happen each period of time, and functions are added. Incremental model so the system will be developed incrementally, allowing users to access some services while other services undergo development. For example, adding Map and location services to be accessed.

2.2 Software Components

2.2.1 Figma

Figma is a highly regarded cloud-based design and prototyping tool utilized by designers and teams for creating user interfaces (UI) and facilitating collaborative design projects. This platform provides a comprehensive set of features, empowering designers to seamlessly generate, share, and iterate on designs in real-time. Figma's user-friendly interface, robust design tools, and functionalities such as version control and commenting foster efficient teamwork among various stakeholders.

Moreover, Figma's cloud-based infrastructure eliminates the need for software installations and allows designers to access their projects from any device. Its reputation stems from its ability to enhance collaboration, streamline design workflows, and maintain design consistency across teams. We used Figma to design our app interfaces, making them friendly-use and choosing a suitable color scheme.

2.2.2 UML diagrams tools

UML diagram tools are software applications designed to assist in the creation and visualization of Unified Modeling Language (UML) diagrams. These tools offer a wide array of features and capabilities, aiding in the development, documentation, and communication of system architectures and software designs.

UML diagram tools commonly provide various diagram types, including class diagrams, sequence diagrams, and activity diagrams, among others. With user-friendly interfaces, drag-and-drop functionality, and automated layout options, these tools simplify the process

of diagram creation. By utilizing UML diagram tools, software developers, architects, and stakeholders can effectively represent and analyze intricate systems, fostering improved collaboration and communication throughout the software development lifecycle. We used SmartDraw for use case and sequence diagrams needed in our project.

2.2.3 Flutter

We used Flutter for Front-End development which is an open-source UI (User Interface) software development kit (SDK) developed by Google. It allows developers to build cross-platform applications with a single codebase. Flutter uses the Dart programming language, which is also developed by Google, to create high-performance applications for various platforms, including Android, iOS, web, and desktop.

- Widget-Based Development

  Flutter follows a widget-based approach, where everything is a widget. Widgets are reusable UI elements that compose the user interface. There are two types of widgets in Flutter: stateless widgets and stateful widgets. Stateless widgets are immutable and don't have internal state, while stateful widgets can change dynamically based on user interactions or other factors. Understanding and effectively using widgets is crucial for developing Flutter applications.

- Single Codebase, Multiple Platforms

  One of the primary advantages of Flutter is its ability to build cross-platform applications. With Flutter, you can write code once and deploy it on multiple platforms. This approach saves development time and effort, as there is no need to write platform-specific code. The methodology should focus on how you utilized this feature to ensure consistency across different platforms and optimize code reuse.

- Material Design and Cupertino

  Flutter provides built-in UI libraries called Material Design and Cupertino. Material Design follows Google's design principles and is suitable for Android applications, while Cupertino provides an iOS-style UI. Depending on the target platform, developers can choose the appropriate UI library to create a consistent user experience. Explaining your design choices and adherence to platform-specific guidelines can be a valuable part of the methodology.

For instance, we used these packages:

- cupertino_icons: A package that provides the iconic Apple-style Cupertino icons, allowing developers to enhance the visual appeal and consistency of their Flutter applications.
- provider: A powerful state management package that simplifies data sharing and access between widgets, promoting efficient code organization and reducing

boilerplate code. Provider state management in Flutter simplifies data sharing and access across widgets, reducing code repetition. It optimizes state updates and rebuilds only necessary components. Provider follows a reactive approach, ensuring widgets respond to changes in data. It supports dependency injection for managing complex dependencies. With its simplicity and versatility, Provider is a popular choice for scalable and maintainable Flutter applications.

- google_fonts: Allows developers to use a wide range of beautiful and customizable fonts from the Google Fonts collection.
- firebase_core: Provides essential Firebase initialization and configuration, allowing seamless integration of various Firebase services into Flutter applications.
- cloud_firestore: A Flutter plugin that enables real-time data synchronization and storage using Firebase Cloud Firestore, allowing developers to build powerful and scalable cloud-based applications.
- firebase_auth: Simplifies user authentication and authorization using Firebase Authentication, providing secure login, registration, and user management capabilities.
- shared_preferences: A simple package that enables developers to store and retrieve key-value pairs locally, providing a straightforward solution for managing small amounts of persistent data in Flutter apps.
- awesome_dialog: Offers customizable and visually appealing dialog boxes and pop-ups for Flutter applications, enhancing the user experience and interaction.
- date_time_picker: A package that offers a simple and customizable date and time picker for Flutter, providing a user-friendly way to select dates and times.

2.2.4 Firebase

We used Firebase as Back-End development which is a comprehensive mobile and web development platform offered by Google. It provides a wide range of cloud-based services and tools that simplify backend development tasks, such as data storage, authentication, analytics, and more. With Firebase, developers can focus on building high-quality applications without worrying about server management or infrastructure. It offers seamless integration with Flutter, making it an ideal choice for scalable and feature-rich mobile and web applications.

- Firebase Authentication

  Firebase Authentication is a service provided by Firebase that enables developers to easily add user authentication and authorization capabilities to their applications. It supports various authentication methods, including email/password, phone number, social media logins (Google, Facebook, Twitter), and more.

  With Firebase Authentication, developers can authenticate users securely, manage user accounts, and implement access controls effortlessly. It ensures a smooth and secure user authentication experience, enhancing the overall user satisfaction and app security. In our app, we used email/password and phone number to register a user and save his id to be able to reach him while interacting with the system.

- Firebase Firestore

  Firebase Firestore is a powerful and flexible NoSQL document database provided by Firebase. It offers real-time data synchronization and a scalable cloud-based storage solution for applications. Firestore allows developers to store and retrieve data in a structured manner, making it easy to build responsive and collaborative applications.

  It offers real-time listeners that enable automatic updates to the user interface whenever data changes, ensuring a seamless user experience. Firestore's scalable and serverless architecture makes it a robust choice for building data-driven applications in Flutter. We used Firestore to store users' data and retrieve them when needed.

2.3 REQUIREMENTS

2.3.1 User functional requirements

Functional requirements define the capabilities and functions that a system must be able to perform successfully. The functional requirements of our app include:

Admin:

- Register drivers

Driver:

- Sign in
- Sign out
- Control bus
    - Approve or delete passengers
    - Book for passengers
    - See booked passengers
    - View seats
- Main screen
- View information
- View History
- View profile
- View Frequently asked questions

Passenger:

- Sign up
- Sign in
- Sign out
- Book seats
- Cancel book
- Main screen
- View book details
- View information
- View History
- View profile
- View Frequently asked questions

2.3.2 System functional requirements

- Registration

  Passengers must register first to be able to access the app but no need for drivers to register because they were added by admins into Firebase. A passenger can only have one account. Before registration completion, the app checks whether he already has an account. When passengers first register, they need to enter the following data:

  - Full name

- Phone number
- Email
- Password
- Repeat password

- Sign in

  When passengers or drivers sign in, they need to enter the following data:

  - Phone number
  - Password

- Control bus

  View the seven seats. A driver can change each seat state (free, booked, or taken seat). Each seat can have only one passenger and his information.

- Book seats

  Passenger can book a seat at present or in the future; he can view the booking details and confirm his booking. If he books at the present time, he can cancel his booking for only a specific time after the confirmation. And if he books at a future time, he can cancel the book before the current time equals booked time.

  Passenger chooses:

  - Location of waiting

- Time of travel
- Number of seats to book.

- View screens

  Each driver or passenger has own screens and data. Retrieving data from Firebase Firestore and viewing them properly.

2.3.3 Non-functional requirements

A non-functional requirement specifies how the system should behave and they specify constraints upon the behavior of the system. One could also think of non-functional requirements as quality attributes of a system.

2.4 Inputs

Mobile users open the app and interact with interfaces developed by Flutter. Many buttons may be pressed in the interfaces and considered as inputs for the system.

- Sign up

  A passenger enters his full name, phone number, email, and password.

  A driver is registered via the admin by entering his full name, phone number, email, and password.

- Sign in

  each driver or passenger enters his phone number and password.

- Booking process

  Passenger chooses the location of waiting, time of travel, and number of seats to book. When a driver reaches specific locations, the system determines his turn and adds him to the available drivers list on that side.

- Seats states

  The driver can change each seat state (free, booked, or taken seat).

2.5 Software Diagrams

2.5.1 Use case diagrams

For drivers, we make it easier to book for passengers and know when and where they are. They can also browse the app. Use cases are shown in Figure 1.

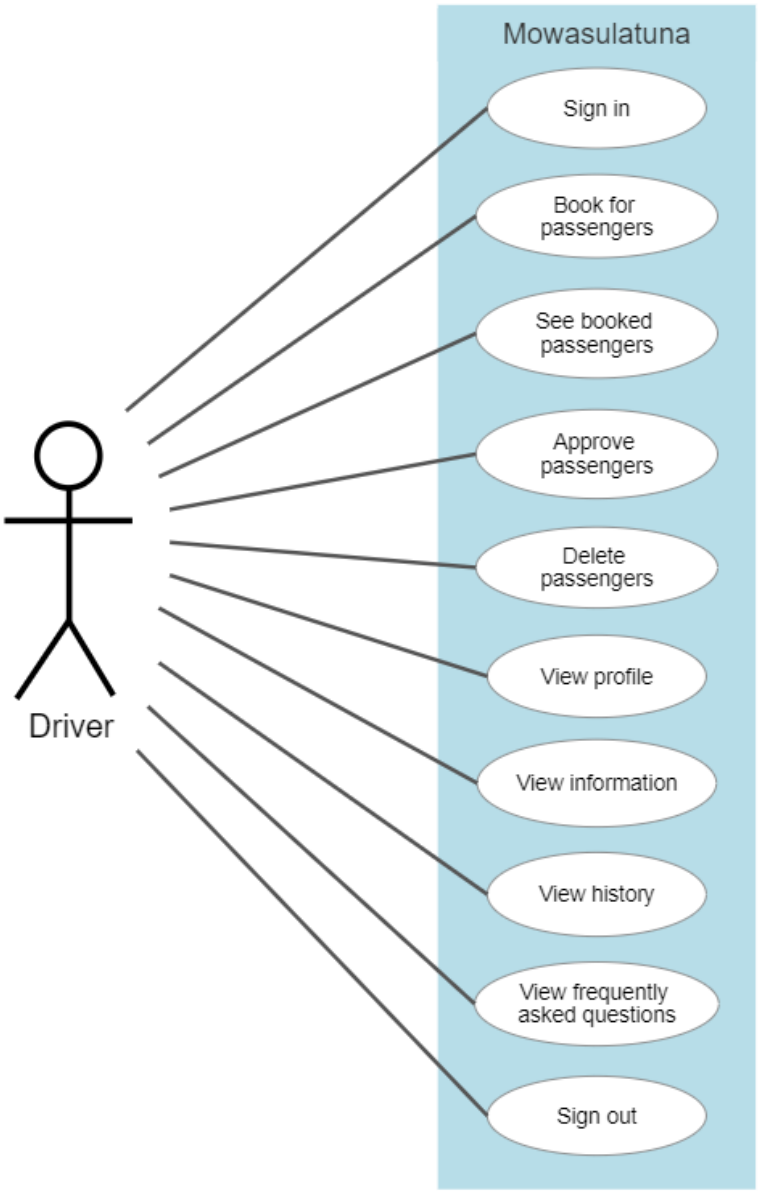

Figure 1: A use case diagram for the drivers.

For passengers, we make it easier for them to book their seats and view the interfaces they need. Use cases are shown in Figure 2.

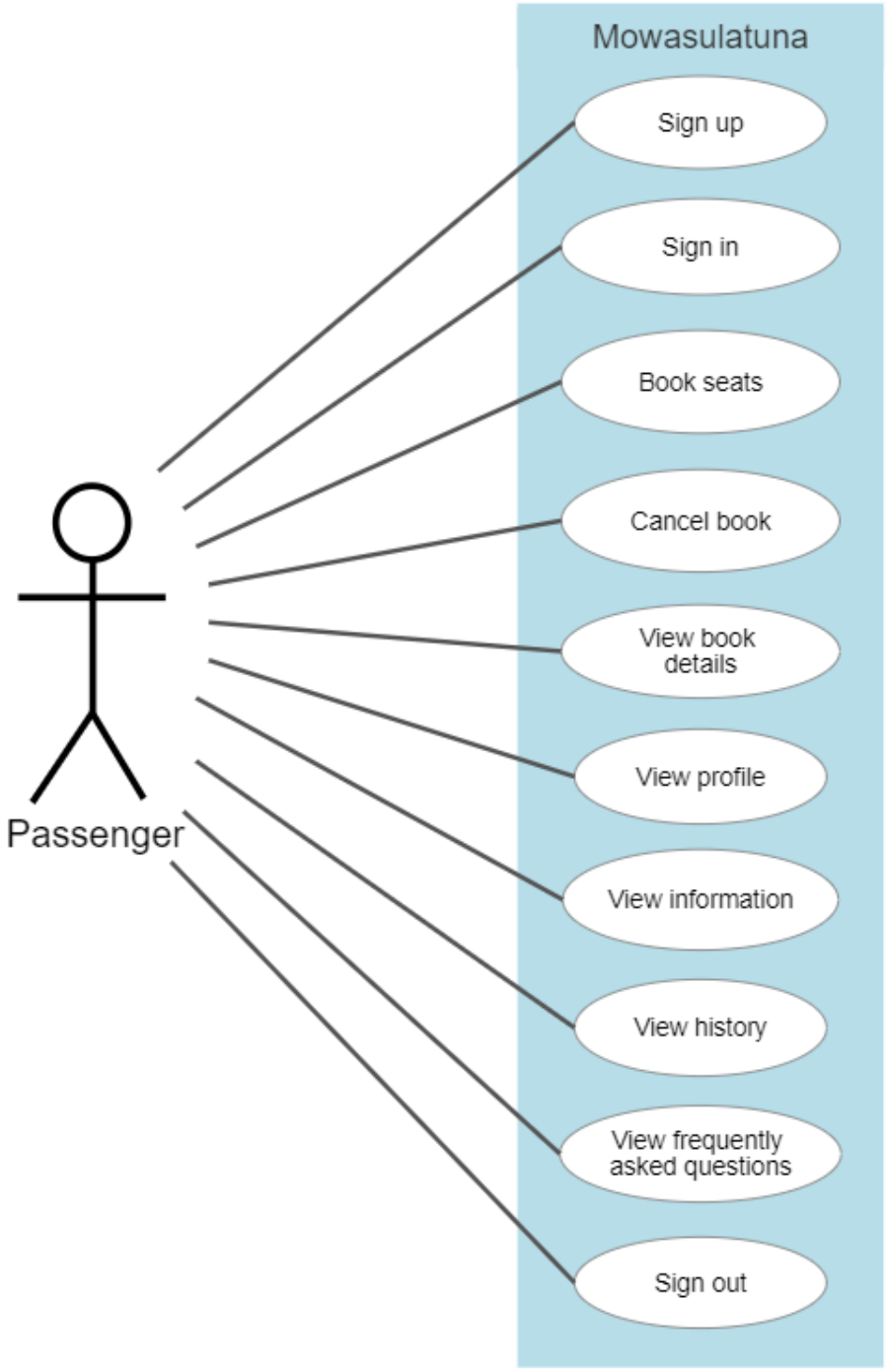

Figure 2: A use case diagram for the Passengers.

For admins, the main thing they're responsible for is registering drivers, as shown in Figure 3.

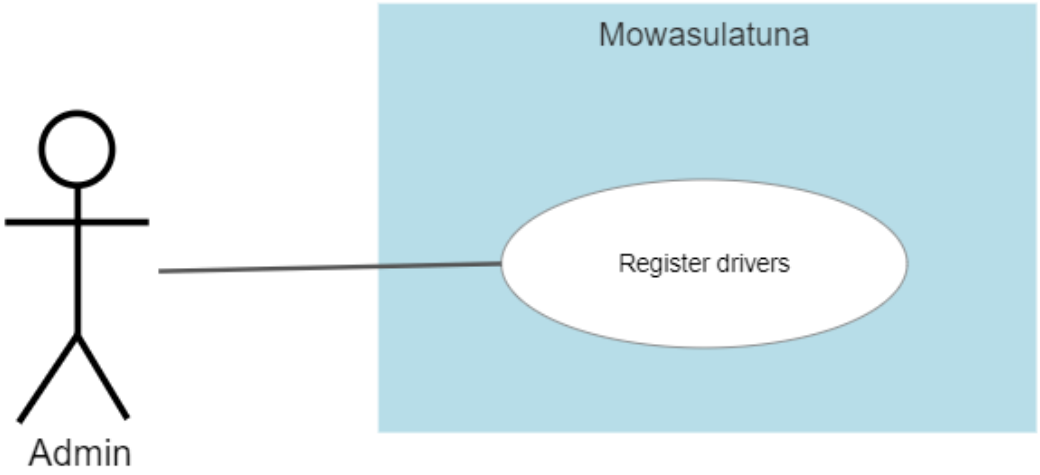

Figure 3: A use case diagram for the admins.

2.5.2 Algorithms and sequence diagrams

In the application, every operation has its technique and mechanism. We're going to show these operations in detail and the corresponding sequence diagrams. These sequence diagrams are shown in Figures 4-18.

Sign in:

When pressing sign in button, the app sends the entered data to Firebase Authentication to check if the user exists and has entered the correct phone and password. The app goes to the bus screen if the user is a driver or to the main screen if he is a passenger.

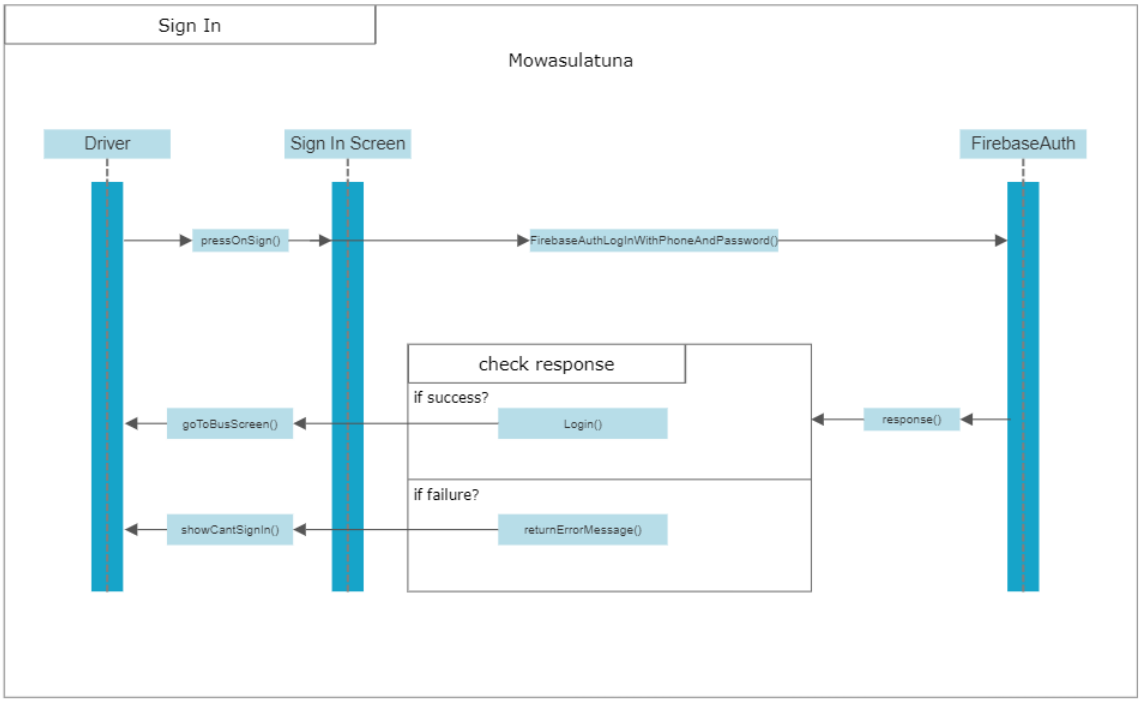

Figure 4: A sequence diagram demonstrates the sign in operation for the driver.

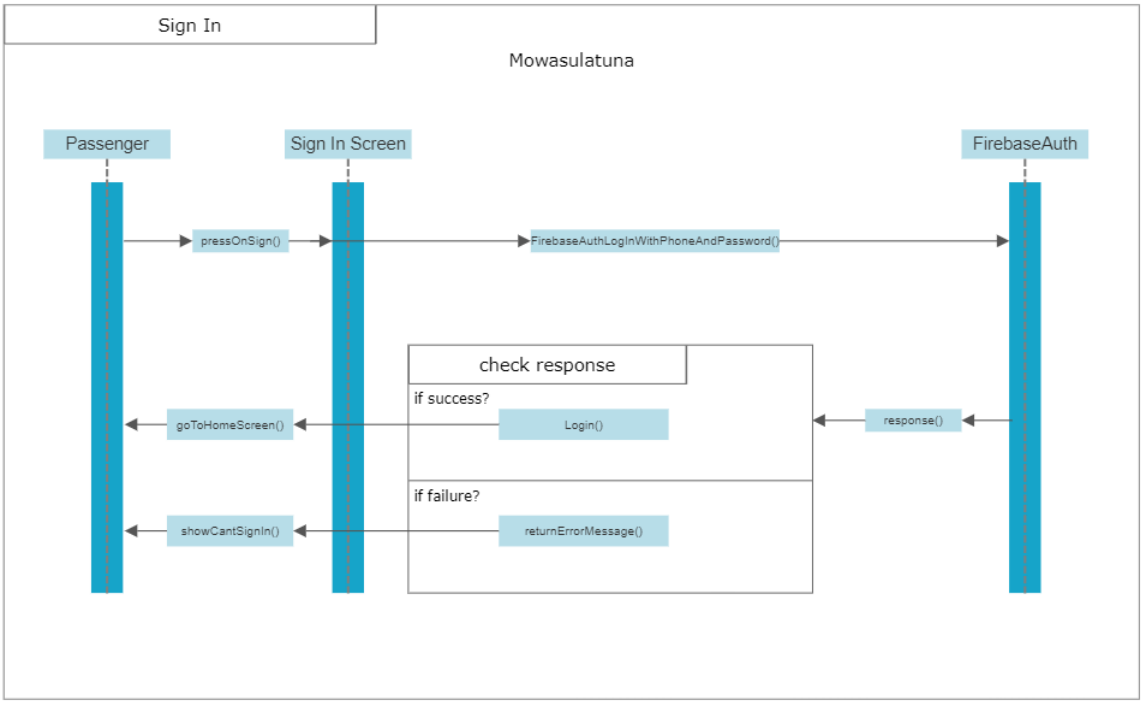

Figure 5: A sequence diagram demonstrates the sign in operation for the passenger.

Sign up:

When pressing the sign up button, the app validates entered data. After that, the app sends them to Firebase to check the existence of the user. If the user does not exist, Firebase sends the verification code to the passenger's phone and the app goes to verify code screen. If he entered the correct verification code, a new passenger will be added to Firebase.

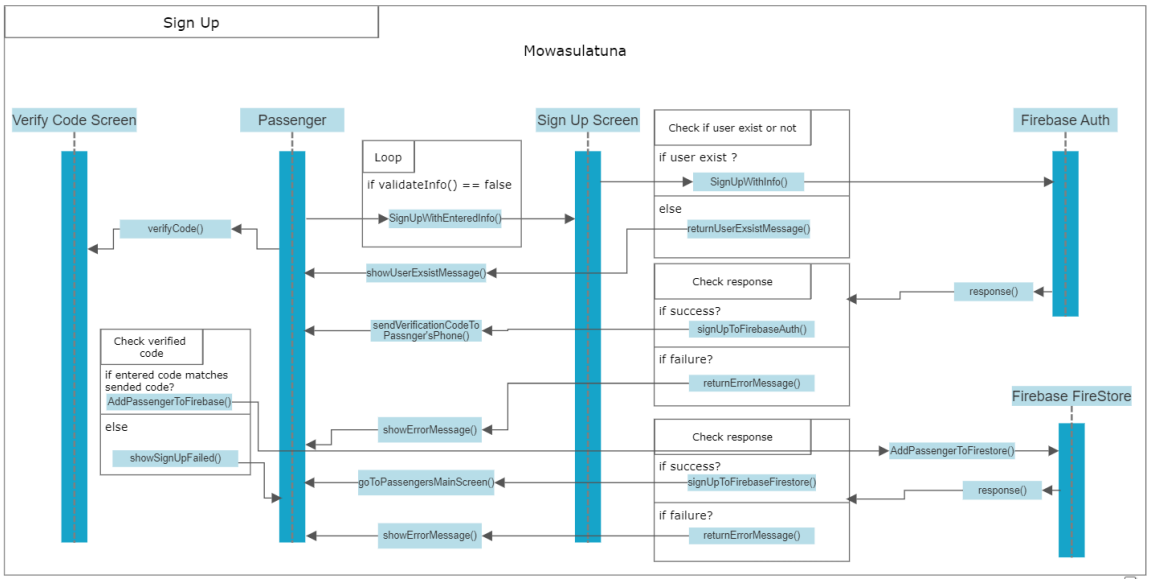

Figure 6: A sequence diagram demonstrates the sign up operation for the passenger.

Book seats:

Passenger chooses the location of waiting, time of travel, and number of seats to book. Passengers can only have one book at a time, and book time must be present or future. Then app goes to the book details screen to confirm or cancel the book. If the book is confirmed, it will be added to Firestore and the book appears to the passenger. If there is an available driver and if the current time is after or equal to book time, the book appears on his bus screen as an orange booked seat.

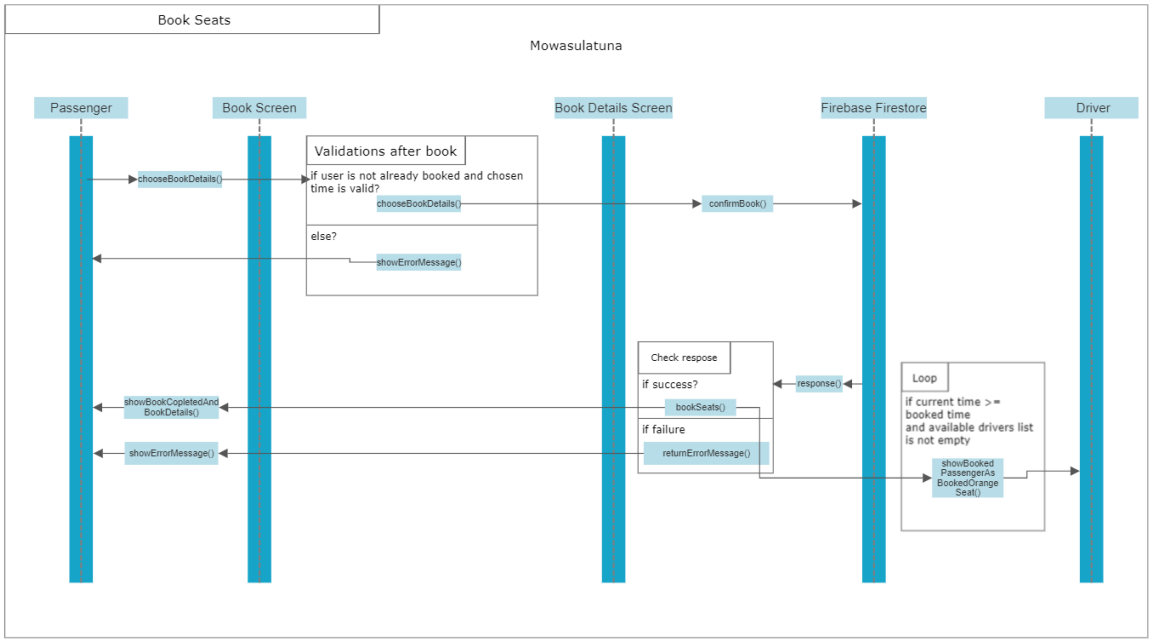

Figure 7: A sequence diagram demonstrates book seats operation.

Cancel book:

If the Passenger books at the present time, he can cancel his booking for only a specific time after the confirmation. And if he books at a future time, he can cancel the book before the current time equals booked time.

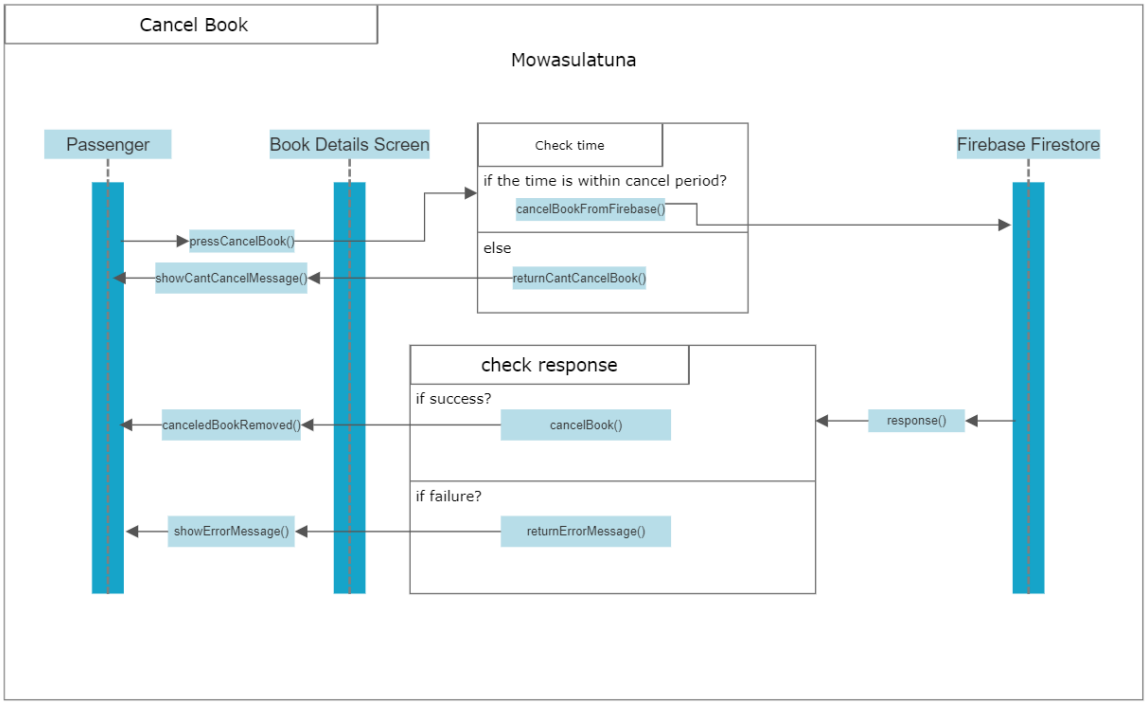

Figure 8: A sequence diagram demonstrates cancel book operation.

Book for passengers:

If the driver finds a passenger not booked in the app, he can add him to his bus by pressing the empty seat in his bus screen then this seat becomes a green taken seat along with seat state is changed in Firestore.

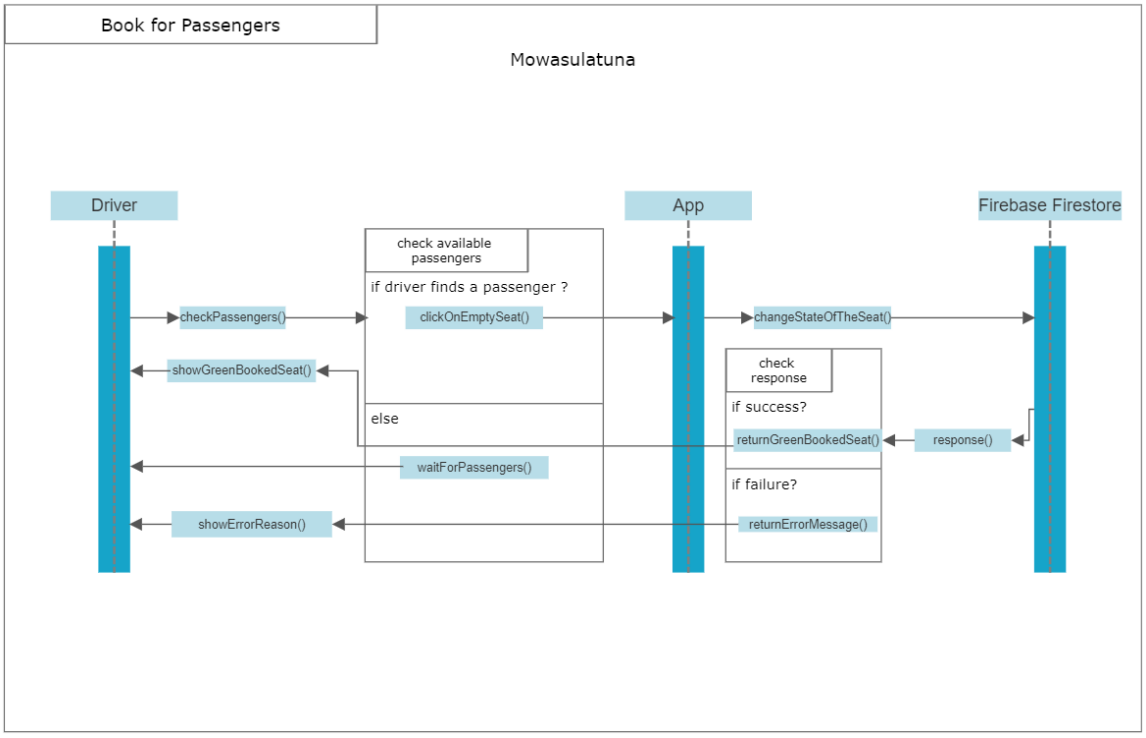

Figure 9: A sequence diagram demonstrates book for passengers operation.

See booked passengers:

When a driver has the turn, booked passengers assigned to him appear as orange booked seats in his bus screen.

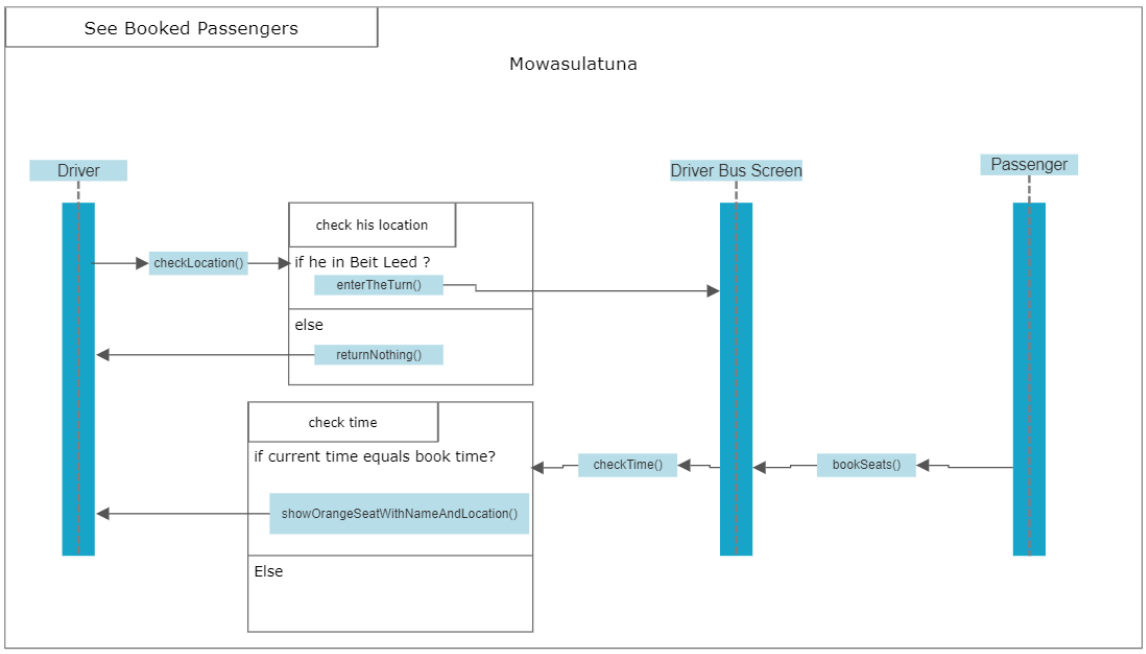

Figure 10: A sequence diagram demonstrates see booked passengers operation.

Approve passengers:

When the driver's bus screen has booked passengers appears as orange booked seats, he can approve this book by pressing on this seat when booked passenger rides the bus. This orange seat becomes a green taken seat.

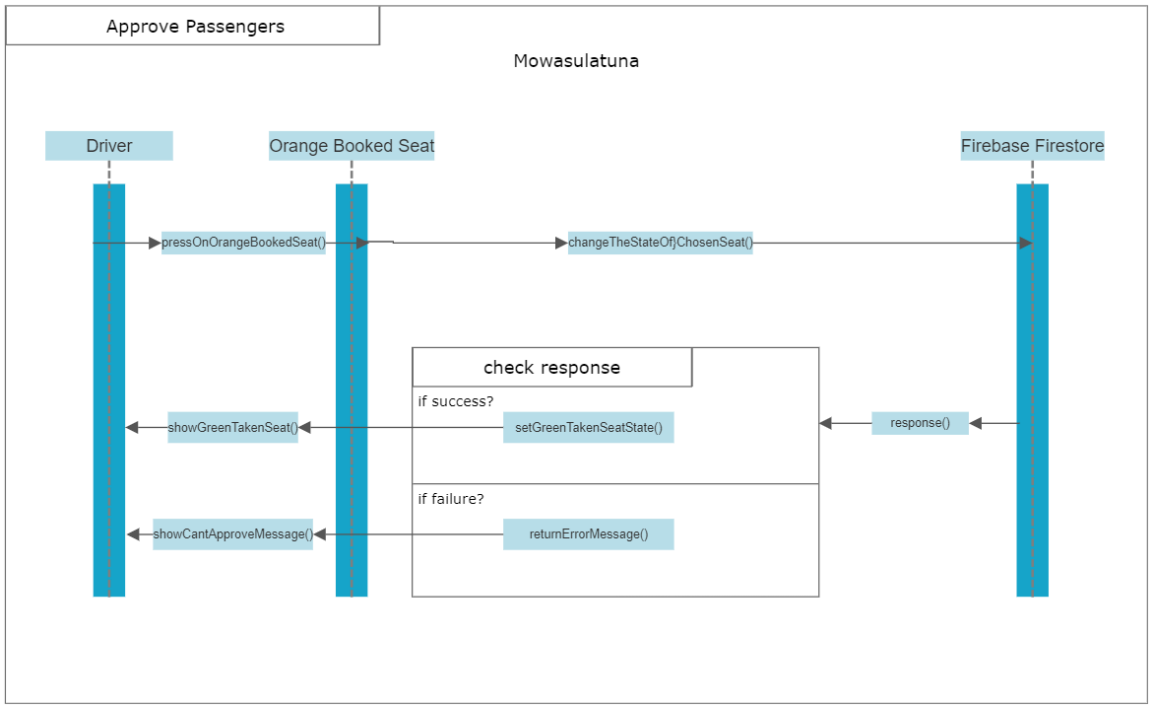

Figure 11: A sequence diagram demonstrates approve passengers operation.

Delete passengers:

In the case the passenger doesn't commit to his book location and time, the driver can remove the passenger's book by pressing on the orange booked seat. The orange seat becomes a gray free seat.
In addition to that, when passengers get off the bus green taken seats become gray free seats by pressing on them.

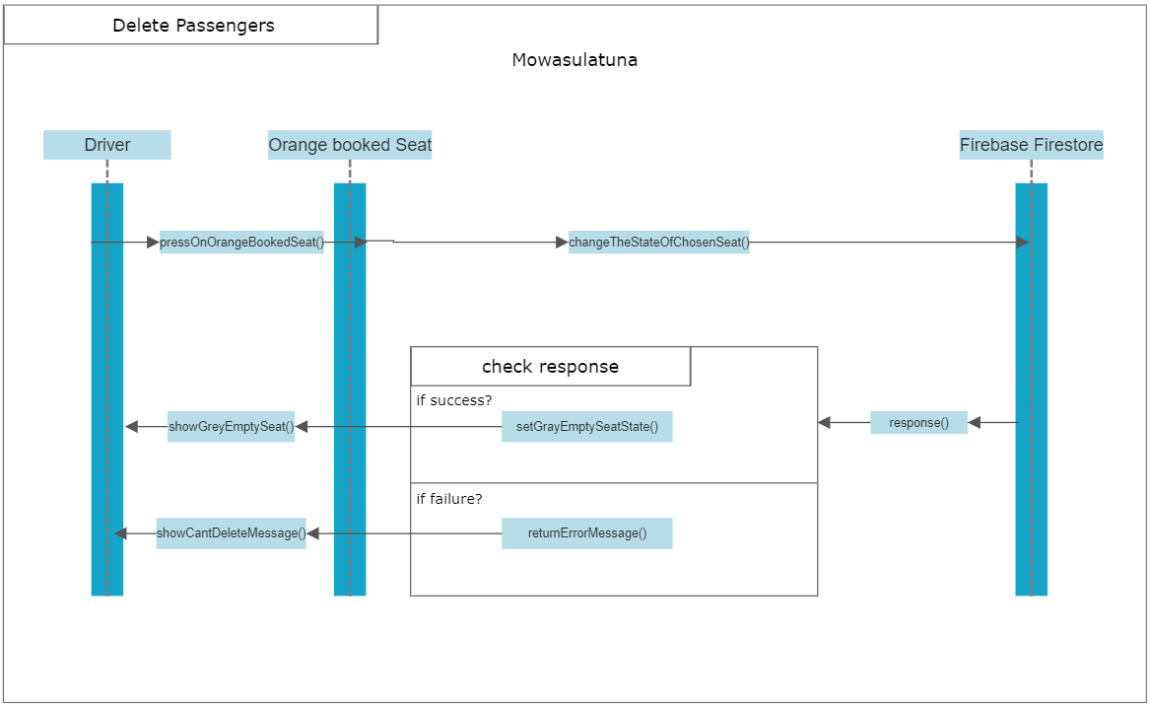

Figure 12: A sequence diagram demonstrates delete passengers operation.

View profile, history, information, and frequently asked questions:

Users can browse the app and view various interfaces and data retrieved from Firebase Firestore.

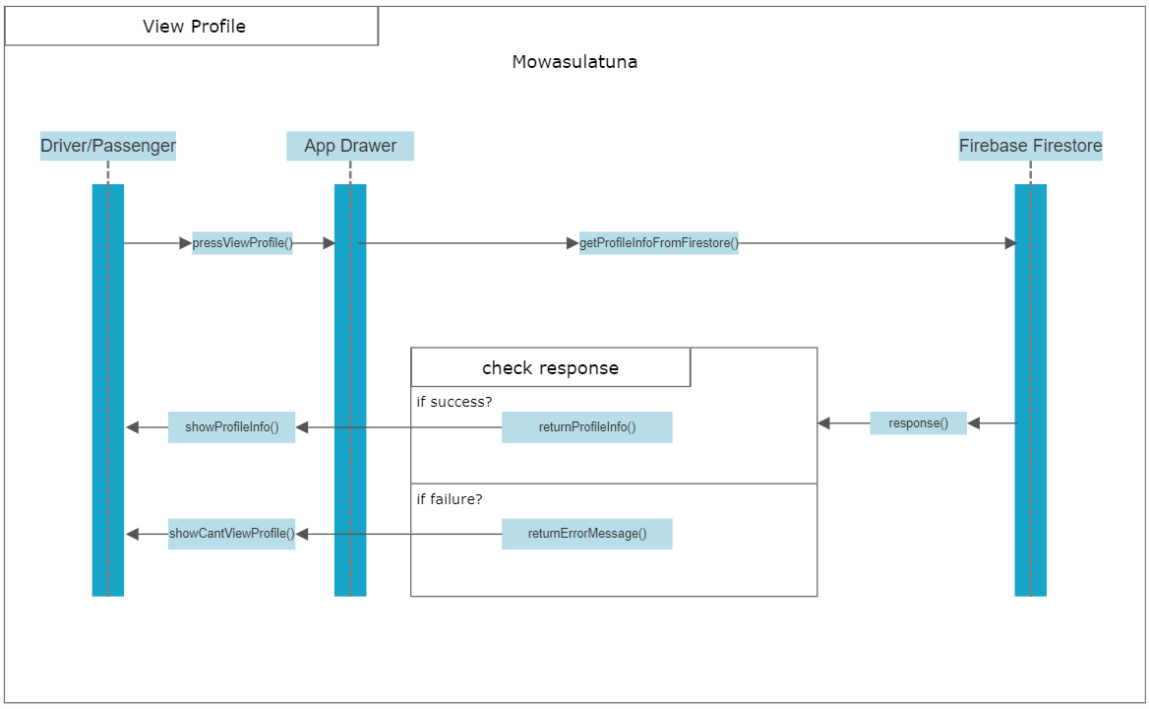

Figure 13: A sequence diagram demonstrates view profile operation.

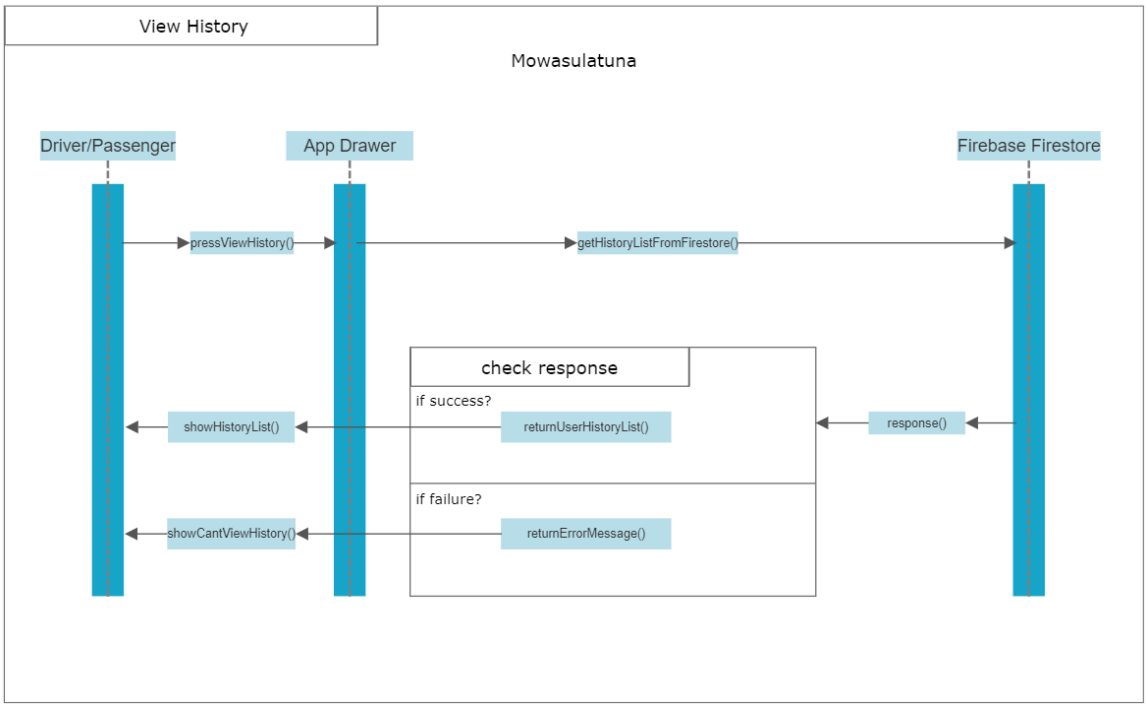

Figure 14: A sequence diagram demonstrates view history operation.

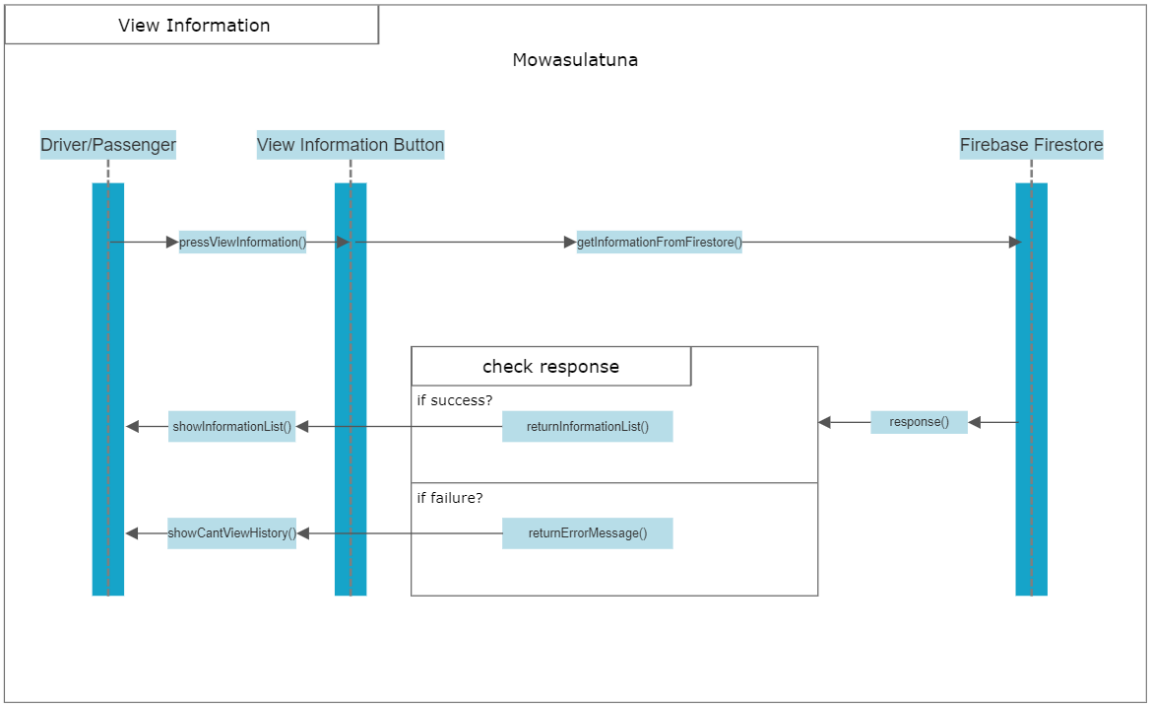

Figure 15: A sequence diagram demonstrates view information operation.

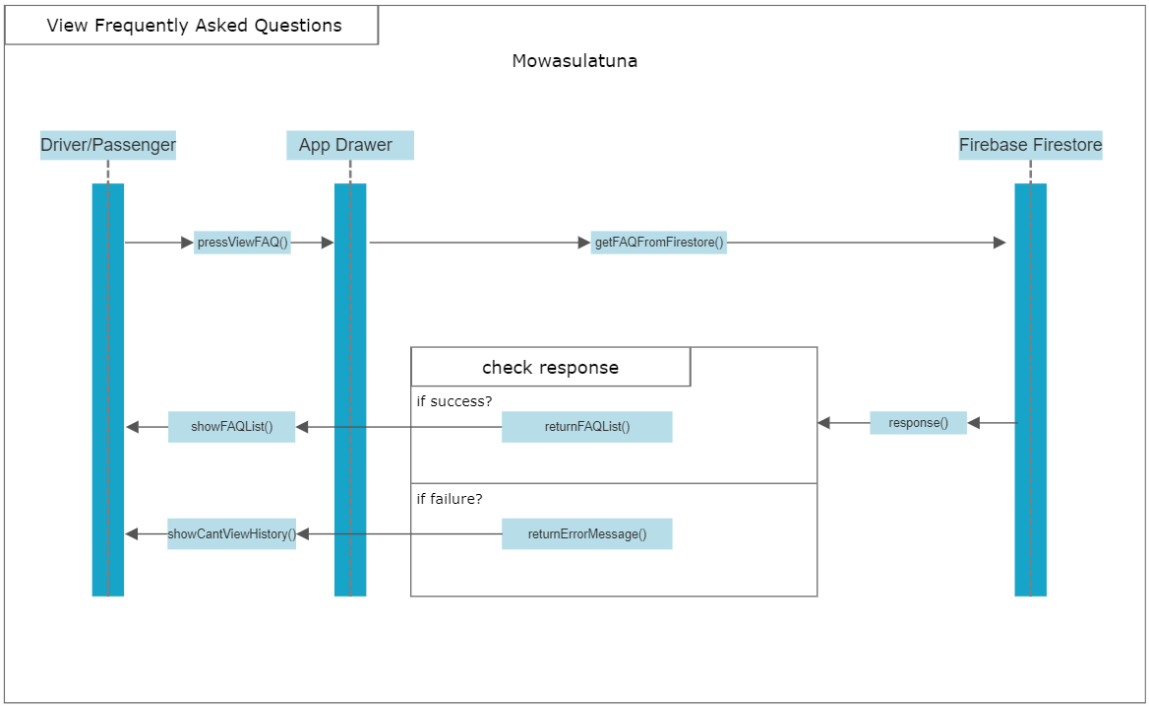

Figure 16: A sequence diagram demonstrates view frequently asked questions operation.

View book details:

A passenger can view his book details from the main screen.

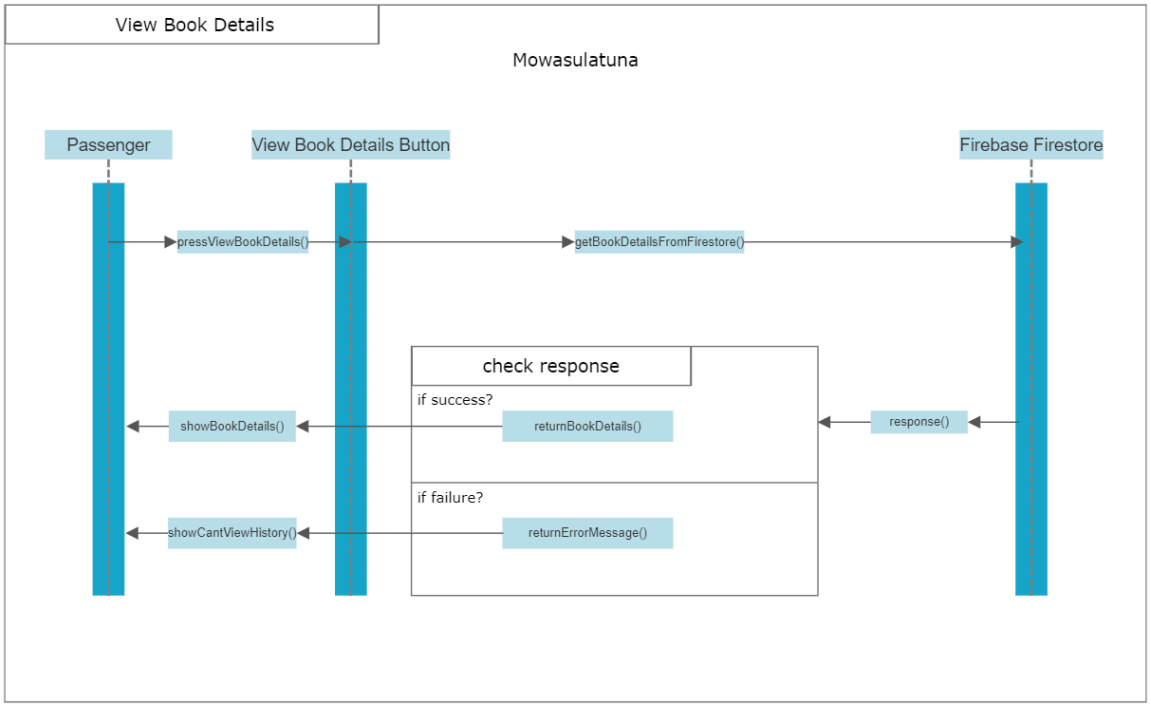

Figure 17: A sequence diagram demonstrates view book details operation.

Sign out:

Drivers or passengers can sign out from the app by pressing sign out in the app drawer.

The app goes to the sign in screen.

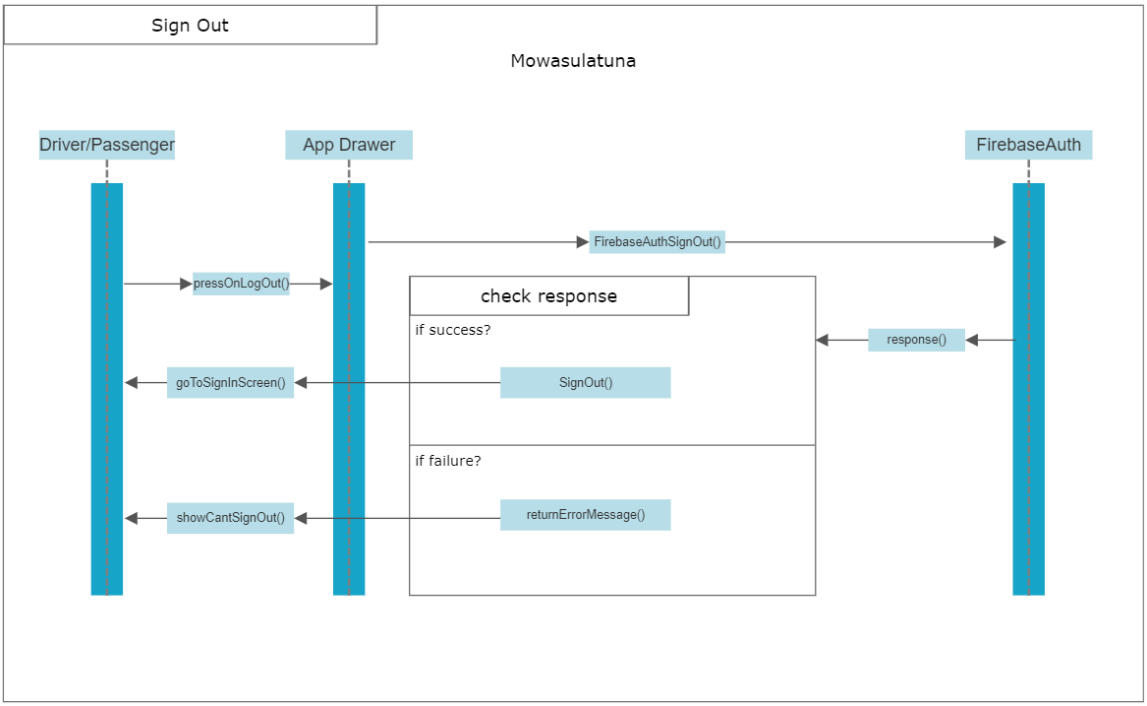

Figure 18: A sequence diagram demonstrates sign out operation.

CHAPTER THREE: RESULTS AND DISCUSSION

3.1 User Interfaces

In this Section, we will show the app screens, and we will summarize what each interface in the system will do.

3.1.1 Introduction

In these Screens, we give an introduction and information about the app for the user. They will appear for the first session only and will not reappear again unless the application is Reinstalled or its data cleared.

Figure 19: introduction_1 screen

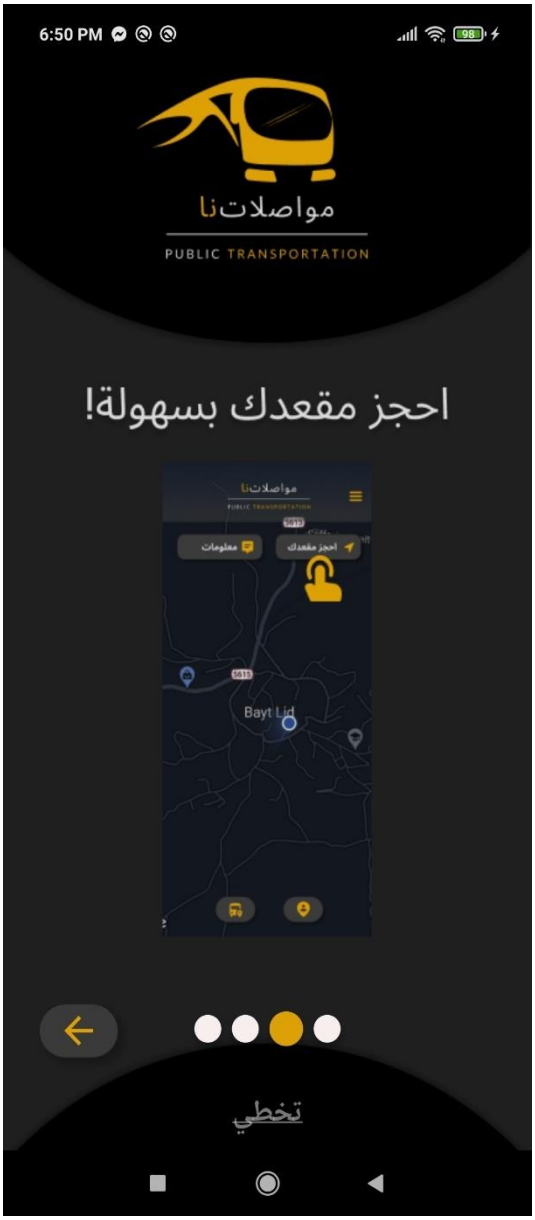

Figure 20: introduction_2 screen

Figure 21: introduction_3 screen

Figure 22: introduction_4 screen

3.1.2 Sign in:

This screen will ask the user to sign in using his phone number and password. If the user doesn't have an account, he can move to the sign-up screen by clicking on (مستخدم جديد).

Figure 23: sign in screen

3.1.3 Sign up:

In this Screen, the user can create a new account by inserting his full name, phone number, email, and password.

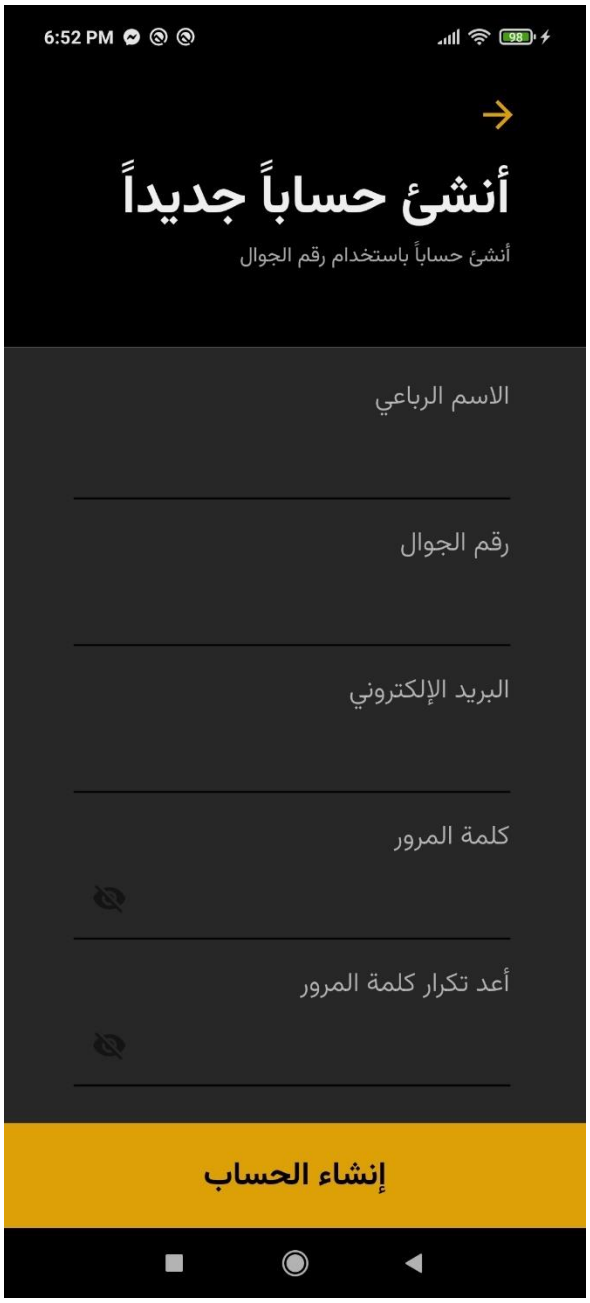

Figure 24: sign up screen

A confirmation code will be sent to the inserted phone number via SMS (Short Messaging Service). After inserting the code, the account will be created successfully.

Figure 25: confirmation code screen

3.1.4 Home Screen:

The home screen has two different appearances, the passenger home screen and the driver home screen.

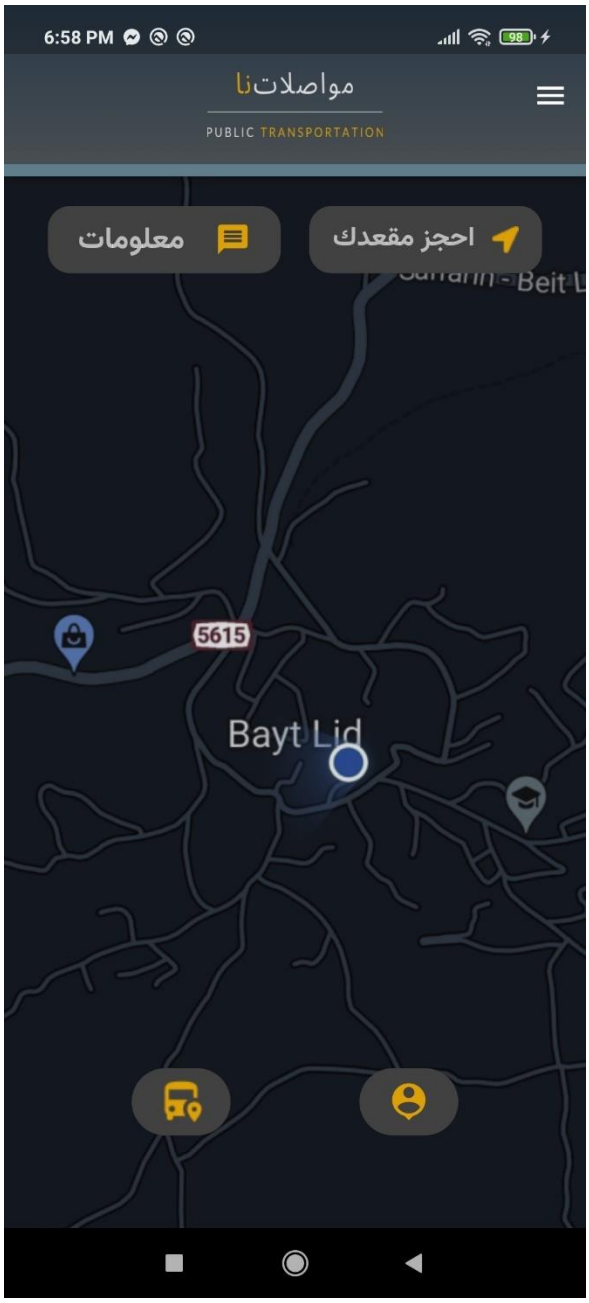

Figure 26: passenger home screen

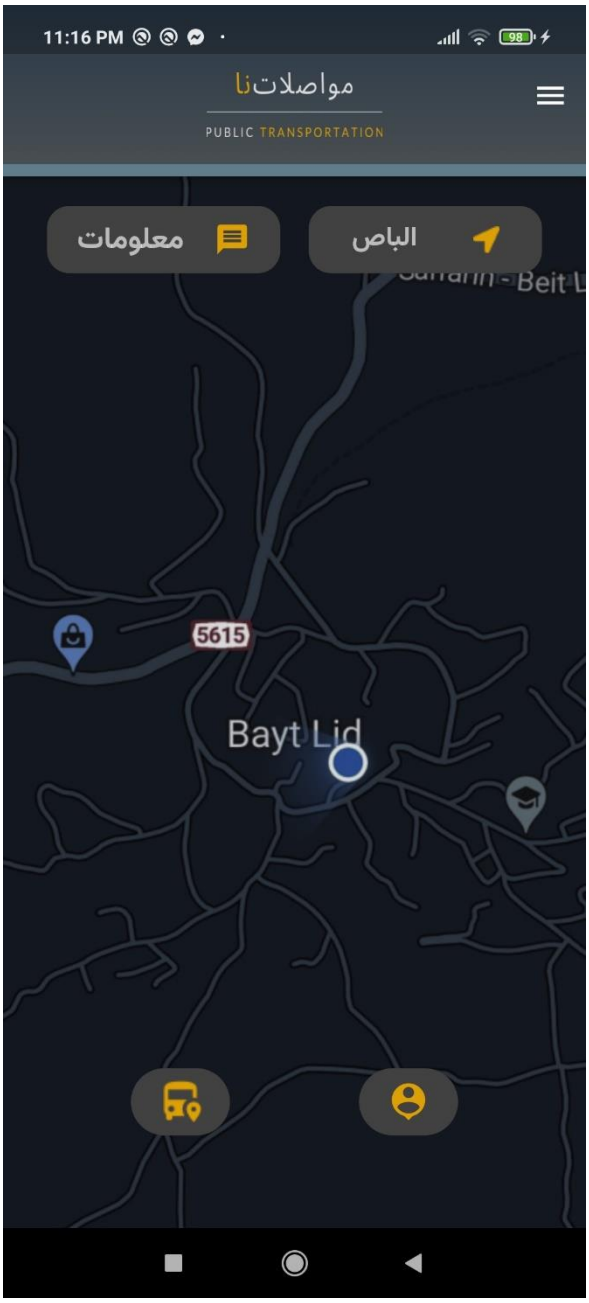

Figure 27: driver home screen

Passenger Home Screen:

In the passenger side, the home Screen provides the booking service by clicking on the button (احجز مقعدك), and it gives information about the journey by clicking on the (معلومات) button.

Driver Home Screen:

In the driver side, the button (الباص) takes the driver into "my bus" screen which allows him to control each seat in the bus as shown in the figure, and it gives information about the journey by clicking on the (معلومات) button.

In "my bus" screen there are three cases for the seat, (free, booked, or taken), which can be accessed and changed by the driver.

Figure 28: my bus screen

The figure below shows the booking operation in the passenger side. It requires a place, a valid time, and the number of people.

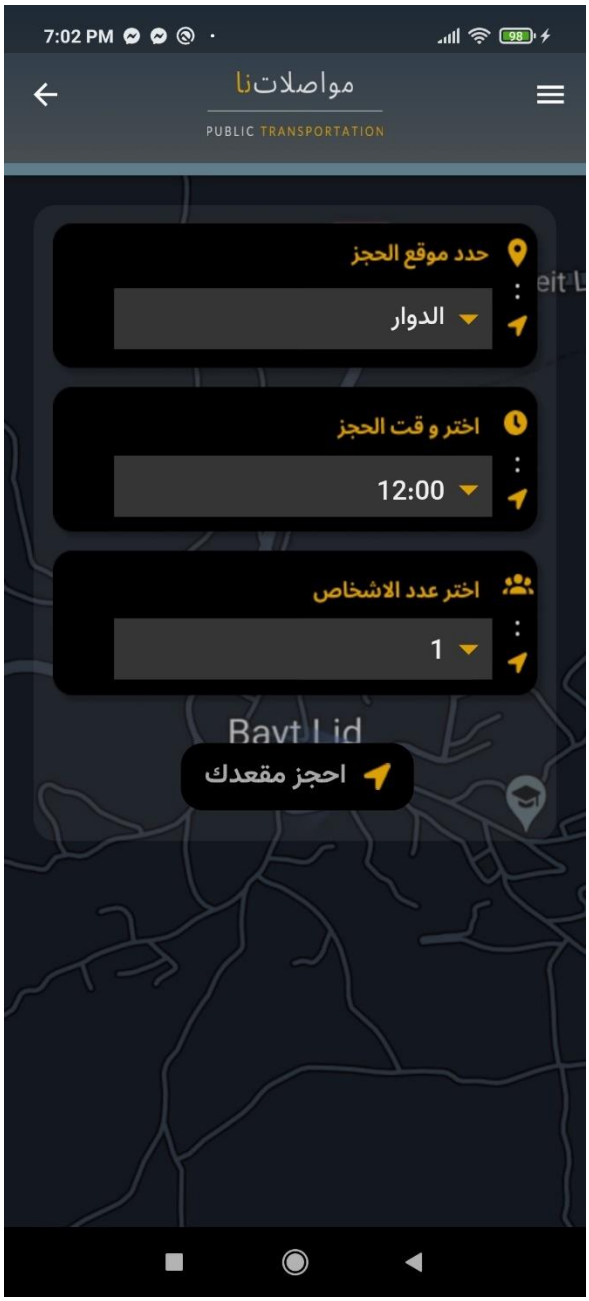

Figure 29: book screen

The home screen also contains a drawer, which allows the user to sign-out, and reach other features as shown in the figure.

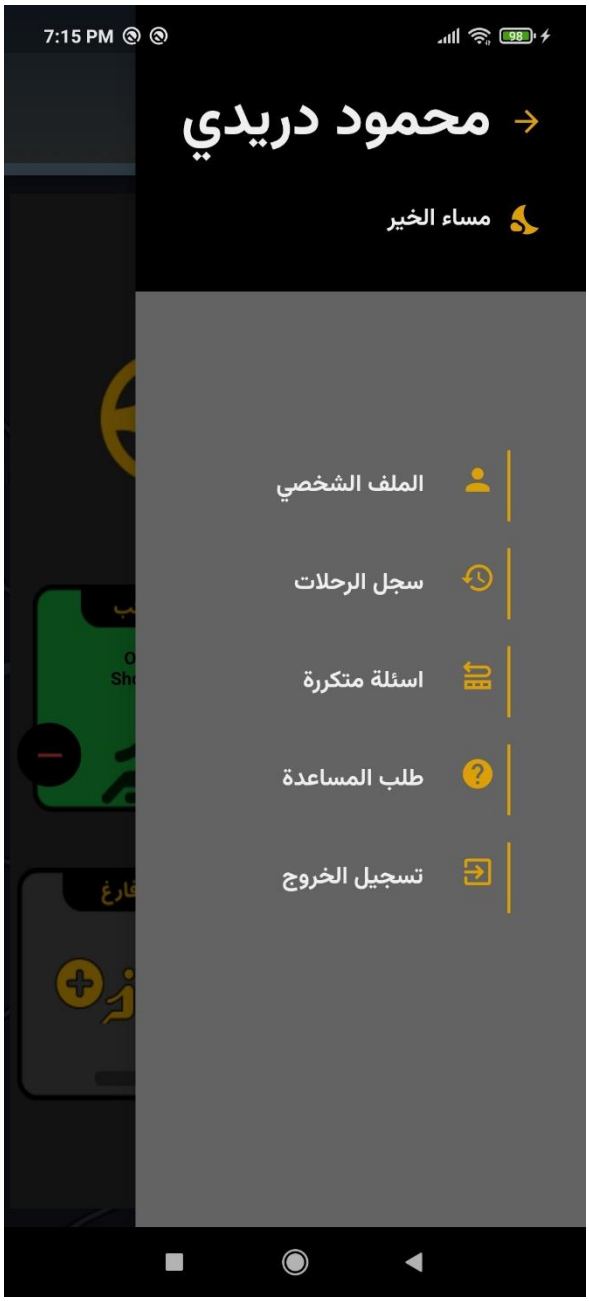

Figure 30: drawer screen

3.2 Output

- Passenger side:

  The passenger sends a book request, the output will be a confirmation that the operation has been done successfully or a failure message if the inserted data contains wrong data (invalid time, invalid number of passengers, etc..).

  The figures next page show the details of the booking operation.

- Driver side:

  After the passenger creates a seat reservation request, the driver is notified, and this appears through the seat state on his “my bus” screen. Then, the driver can control the seats state and every change will be shown immediately on the “my bus” screen.

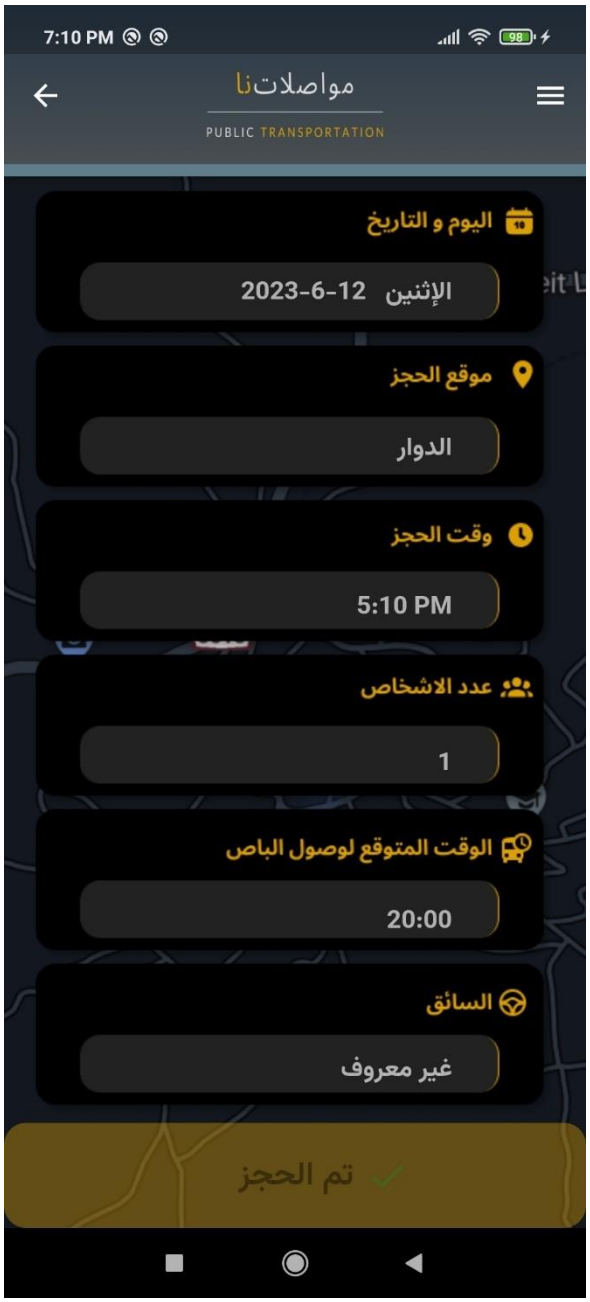

Figure 31: successful booking screen

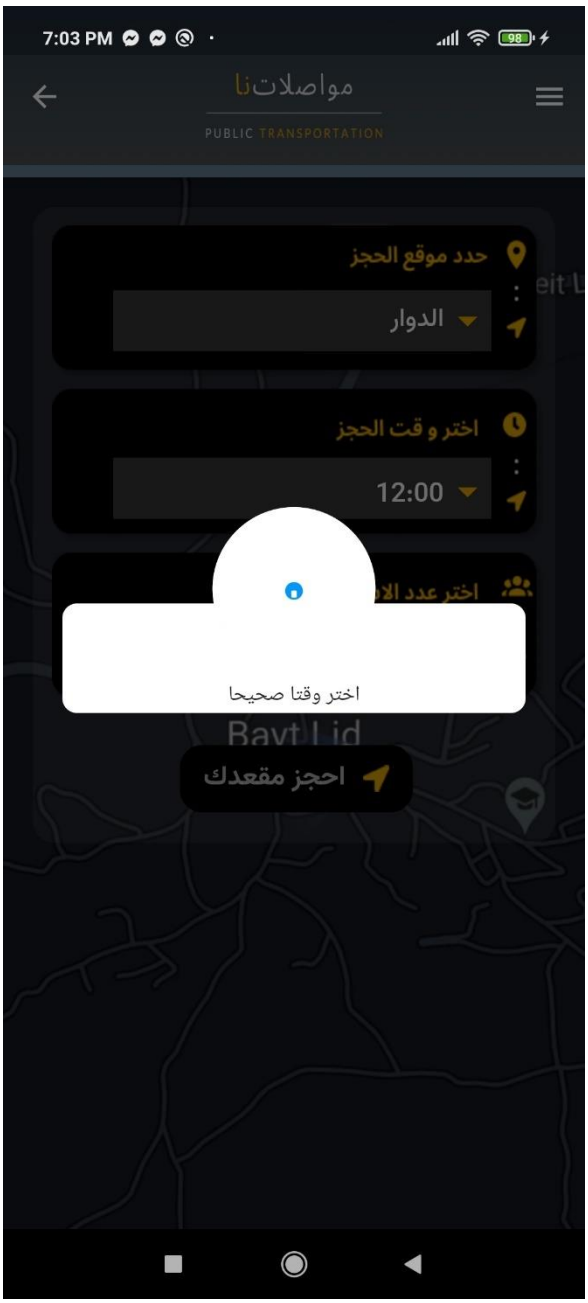

Figure 32: failure message screen

3.3 Challenges and Limitations:

1. The work on the principle of the turn and how the system is able to achieve its goals while respecting this principle.

2. The lack of information. There is no previous data that the Palestinian Ministry of Transportation can provide us where we were informed of this by them through an interview with them.

3. Convince drivers of the feasibility of using the app and how the system won't affect those who don't use the app.

4. Drivers always have to turn on the location to manage the turn.

5. Drivers and passengers have to be connected to the internet, which is not available at all locations.

CHAPTER FOUR: CONCLUSION AND RECOMMENDATIONS

4.1 CONCLUSION

- **Output**

  The available drivers list is generated by applying different turn algorithms and priorities. It shows available drivers with specific locations and times.

  The passenger can see the available drivers list depending on his location and selected time so he can book his next travel. Also, he can get from the app the expected time for his next travel.

  Depending on receiving data from passengers, drivers, and Google Maps, the app provides general information about the streets' state, and locations of each driver and passenger and show this on a map.

Expected Results

1. Saving more time by providing information for both sides.
2. Reduce fuel consumption and distance traveled.
3. Passengers and drivers know the exact location of each other.

Project Timeline

We are supposed to start working on this by the beginning of the Fall 2022 semester and finish it by the end of the Spring 2023 semester. The proposed project timeline shown in Table 3 serves as a guide to follow during the period of accomplishing this task. The project timeline is flexible to accommodate any changes or surprises in the schedule. The

suggested timeline of the project will be updated as the further the group progresses in time and the picture becomes clearer.

Table 3: The proposed timeline of the project.

Month	February				March				April				May			
Task / Week	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Interviews																
Design and Analysis																
Implementation																
Testing																
Development																
Writing thesis																
Deployment																

References

Khosrowshahi, D. (2009). Uber. Retrieved from https://www.uber.com/il/en/

Palestinian Ministry of Transportation and Communications (2021).مواصـلاتي . Retrieved from http://www.mot.gov.ps/

Sheikha, M., Olsson, M., & Elyas, A. (2012). Careem. Retrieved from https://www.careem.com/

Tan, A., & Ling, T.H. (2012). Grab. Retrieved from https://www.grab.com/sg/

Appendices

Interviews

Meetings with Drivers and passengers to make them answer the pre-detailed questions. Interviews with passengers and drivers are shown in Table 1 and Table 2.

Table 1: Interview 1 questions and answers for passengers.

Questions to Passengers	Their Answers
What do passengers do when they want to go to Tulkarm or Beit Leed?	They go to the bus station which is a gathering point where buses exist
How do passengers go to the bus station?	They walk 5 to 15 mins to reach the station
What steps do passengers follow to travel?	There are no steps only walk to the bus station and wait there
How could passengers travel differently?	They book using a phone call or not going to the bus station
What information do passengers use?	No information was used so they don't know if there is an available driver or not

Table 2: Interview 2 questions and answers for drivers.

Questions to drivers	Their Answers
What initial turn do drivers follow?	They have an ordered turn list. A new driver becomes at the top of the list at the beginning of each day.
What steps do drivers take to travel?	They go to the bus station, wait their turn, wait bus to be full of passengers, and start driving.
Which driver is available?	The driver on the turn from 6 am to 10 am and then they don't follow any turn.
What information do drivers use?	Some of them said they call each other to know if there are passengers on each side
What ideas may help the driver?	Know the number and location of passengers. Recommendations about waiting at the bus station or going to the other side of the transportation line with free seats.