Bilkent University

Department of Computer Engineering

Senior Design Project

Carpus

Final Report

Deniz Çalkan, İbrahim Furkan Aygar, Mehmet Yiğit Harlak, Murat Sevinç, Veli Can Mert

Supervisor: Özgür Ulusoy
Jury Members: Shervin Arashloo, Hamdi Dibeklioglu

Final Report
May 6, 2022

Contents

1. Introduction

As thousands of vehicles travel across the city of Ankara everyday. Population of the city, geographical conditions, health and time constraints urge people to drive their own cars if possible. Number of vehicles, travel times and type of cars or motorcycles harms the environment day by day [1]. Carpus is created as a solution which aims to reduce both traffic and gas emissions in Ankara by allowing Bilkent University students to share vehicles alternately via a mobile application. Carpus is designed as a user-friendly and simple mobile application for Bilkent University students to use their personal vehicles together with other students to get to the campus. Our aim in this project is to reduce the number of vehicles used, reduce carbon emissions and the resulting traffic density. By using Carpus, the vehicle owners determine the time and date they will go to the campus. Other participants who are close to the vehicle owner will meet on a starting point through the route created by integrated mapping service by Google Maps and head to the campus through the shortest route created.

Drivers and users who joined the trip would communicate through the group chat feature of the application before and after the trip. For the safety of the users and validation constraints, the vehicle owner must provide license plate information of his/her vehicle and all users must provide their Bilkent University email addresses during the registration.

2. Requirements Details

In this section, currently satisfied requirements of Carpus are listed along with details.

2.1 Functional Requirements

2.1.1 User Functionalities

- Users can create an account using their Bilkent University email.
- Users can login using their registered Bilkent University email.
- Users can register via bilkent email addresses to the system.
- Users can register their cars to the system through its license plate number.

- Users can pin their location on the map.
- Users can mark their destination and departure time.
- Users can create trips.
- Users can join trips.
- Users who are going to carpool on the same vehicle can send and receive messages via group chat.
- Passengers who request a carpool can view available cars with license plate information.
- Passengers who have chosen an available car can view the shortest route to the vehicle on the map.

2.1.2 System Functionalities

- Receive user mail address and password as inputs for the registration process.
- Ask user permission to access the user’s current location.
- Receive mail address and password for the authentication.
- Receive driver’s license plate number while creating a trip.
- Receive the driver's trip date and time while creating a trip.
- Receive the driver's location while creating a trip.
- Receive passenger’s locations while joining a trip.
- Display the shortest route to the vehicle for passengers.
- Display the shortest route to the campus.
- Receive group chat messages only from users who are going to carpool on the same vehicle.
- Show other participants messages in the group chat to the user.
- Keep the past group chat messages and display these messages.
- Keep registered mail addresses and passwords as encoded strings.

2.2 Non-Functional Requirements

2.2.1 Accessibility

- The application is suitable for devices which has a proper version of Android.

2.2.2 Usability

- The application is user friendly, simple and easily adaptable in terms of user interface.

2.2.3 Availability

- The system, currently, is available and operating consistently.

2.2.4 Security

- Since only the university students are permitted to use the application, license plate number and university mail address is verified and sensitive information such as passwords are encrypted.

2.2.5 Privacy

- Home addresses of the users are not received. Users can pin their addresses manually. So neither other users, nor any person are able to obtain the exact address information in case the user does not enter.

2.2.6 Extensibility

- The documentation of the application is systematic and open to any upgrades due to changing functional and non-functional requirements or user interface changes.

2.2.7 Performance

- The application operates fast in terms of launch and login time.
- The application displays available trips in a short amount of time.

2.2.8 Response Time

- Response time for the application is at most 5 seconds which improves user experience.

2.2.9 Maintainability

- Application modules are correctly separated and any update in the application would not interfere with any other non-related module.
- It is possible to increase the maximum number without applying any major changes to the system.

3. Final Architecture and Design Details

3.1 Overview

The final architecture of Carpus is shown in this section. We show the system and subsystems. How our subsystems interact with each other and layers of the subsystem are shown in the subsystem decomposition diagram. Analyzing the subsystems before starting the implementation phase in the project is important to lower the errors and draw a road map. We have 3 layers in the subsystem decomposition. Client layer is for interacting with users. Application layer is for carrying out the logic of our application. Data layer is for securing and storing necessary data for our application. The mapping of the hardware and the software is explained in this section. Moreover, persistent data management, access control and security, and boundary conditions are explained in this section.

3.2 Subsystem Decomposition

We constructed a 3-tier architecture diagram for Carpus [2]. Our diagram consists of Client Layer, Application Layer, Data Layer and the decomposition of subsystems in each layer is represented in Figure 1.

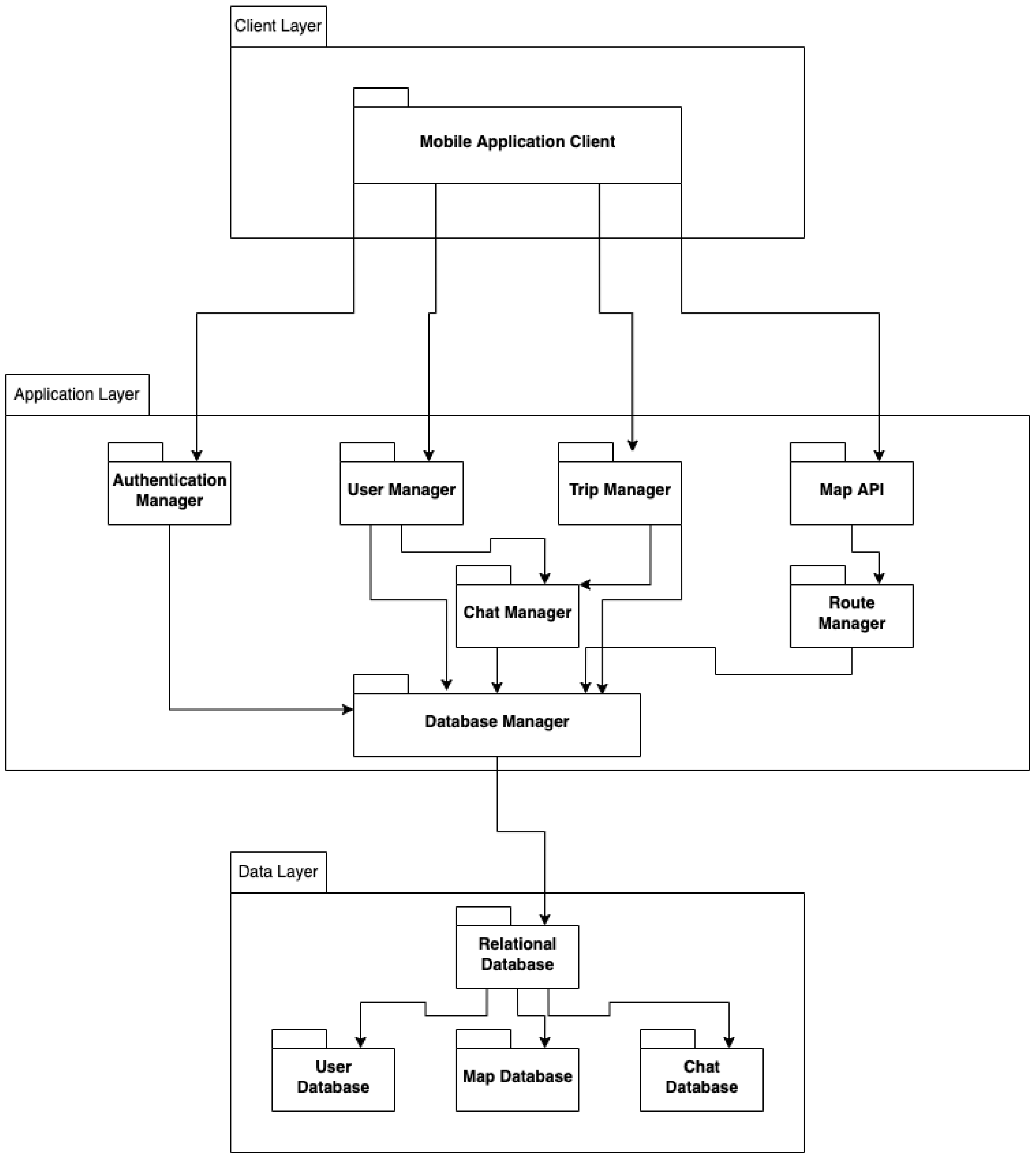


Figure 1: Subsystem Decomposition

3.3 Hardware/Software Mapping

Carpus runs on Android. It uses the GPS and mobile data of the smartphone.Carpus uses mobile data to reach the database for retrieving relative information about the user and it uses the GPS of the smartphone to reach the user's location.

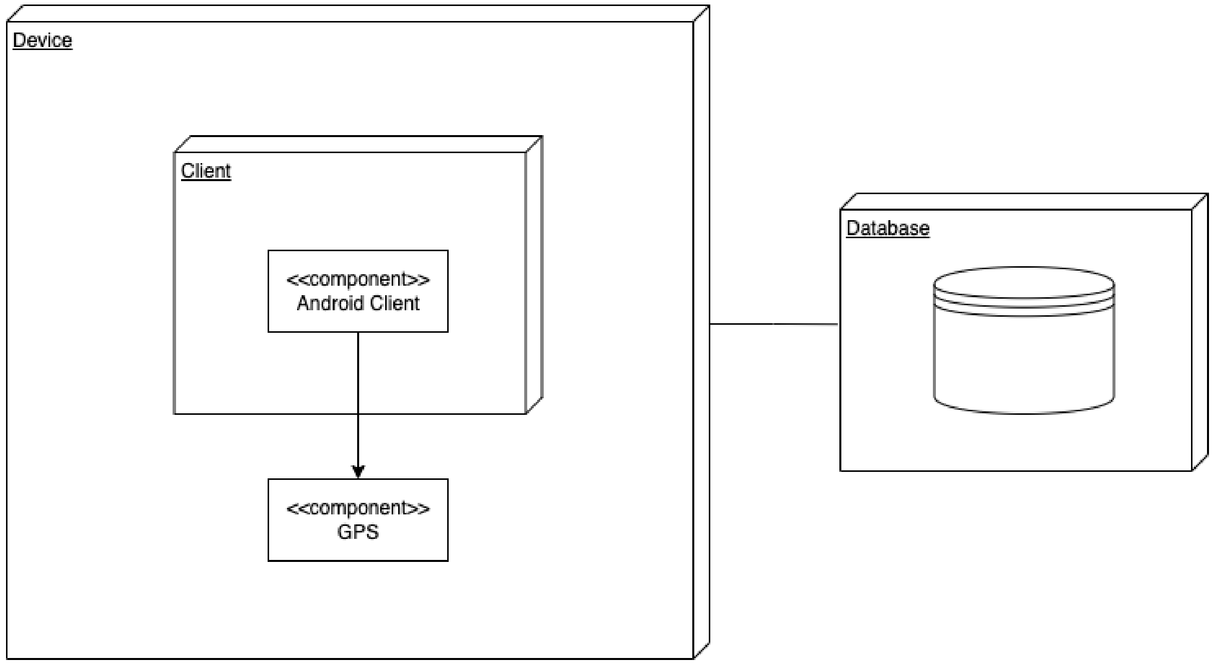


Figure 2: Hardware/Software Mapping

3.4 Persistent Data Management

We need to store all of the users, map and chat data so that we can reach them to keep the application up to date. We need to store huge data according to the number of users as we expect the number of users of the application to increase day by day. Therefore, the data is stored with a structured format to make them lightweight and fasten to reach.

We need to store username, user ID, email and password for clients. We need to store chat, user location and route for trips.We will use the database to store and retrieve data which is mentioned above.

3.5 Access Control and Security

Carpus stores a lot of information about the users which are username, user ID, email, password etc.Therefore, Carpus must be securable application and user’s information must be kept safe. Many security software architectures can be used to keep user and application data safely. If the application does not have enough securable software, the users' information can be stolen and their private data can be shared with third parties. Hashing password, encryption and e-mail or sms verification systems can be added to make Carpus a more securable application. In addition, a face recognition feature can be added to Carpus in order to verify real users and prevent fake usage of fake users.

3.6 Global Software Control

Carpus is a carpooling application for students of Bilkent University so, the users interact with each other to do carpooling. When doing carpooling, users can communicate with each other thanks to the chat feature of Carpus, see the online map and efficient route. For example of the usage of the Carpus carpooling; a user who studies at Bilkent University , if the user has a vehicle , the user can start a trip to go to the university with other users. The users can communicate with each other thanks to the chat feature and they can see their location with the map feature, the map provides the shortest path of the route. After the trip, the group of carpooling can continue other trips or the group can be closed.

Therefore, Carpus has a software architecture that depends on the user. The steps and states that will occur in the application take place depending on the choices of the users. The system aims to present the most suitable carpooling route to the user by providing an output to the users according to the inputs entered by the users.

3.7 Boundary Conditions

Carpus will have 3 boundary conditions consisting of initialization, termination and failure.

3.7.1 Initialization

The user must have the application on his mobile phone, a functioning GPS and the internet connection of the mobile phone must be enabled to initialize the application.

3.7.2 Failure

If there is insufficient internet connection, network failures will be encountered. If login credentials are wrong, an authorization error will occur.

3.7.3 Termination

Users can terminate the application by closing the application or signing out from the application.

4. Development/Implementation Details

4.1 General Details

Carpus is coded in Android Studio with Java version 1.8. Minimum SDK version for users to run Carpus is 21 (Android 5.0) which serves around 98.5% of android users.

4.2 Authentication

The authentication feature of Carpus is provided by Google's Firebase Authentication which provides 10.000 authentications per month which is more than enough for our application. Firebase is chosen for two reasons: it is easy to implement and you can easily integrate different methods of login or signup (such as logging in with email, google, facebook, github, etc.). Our application only provides authentication with email since our application only serves Bilkenters.

Signup information is internally checked to ensure:

- Provided email address is a valid bilkent university email address
- Password length is between 6-15 characters
- All credentials are filled
- Unique username and student identification number

Firebase Authentication
Login
Firebase

Figure 3: Firebase Authentication

4.3 Car Registration

Driver users are obliged to register their car to the system. We store their car information by Firestore Database which is a NoSQL database. Car plate information is used for join trip activity. As authentication, the business logic layer checks whether a car plate is valid or not.

4.4 Creating a Trip

To create a trip, users are first obliged to register their cars to the system. After a successful car registration, they are able to create trips.

In the ‘create trip’ screen, a map and an input area is provided for the driver to initialize the start time of the trip and easily see the closest path to Bilkent campus.

In this part, we used Google Maps API to provide a map and the closest path to the destination. Trips are also stored in the Firestore database to demonstrate existing ones to the joining users.

4.5 Joining to a Trip

Any user can join an available trip. In this screen, the application basically shows the available trips from the database and their previews so that joining users can choose closer trips to their locations. Moreover, the application provides the closest path to the starting location of the trip, thus joiners can easily reach the driver. Application again uses Google Maps API for live tracking.

4.6 Live Chat

There are specific chat rooms created for each trip for participants of the trip. They can talk about meeting details, delay information, etc. Once a user leaves the trip or a trip is deleted, the chat room also disappears / changes.

To provide real-time chatting, firebase’s snapshot listener is used so that the application updates the view once it detects a change in the content.

5. Testing Details

5.1 Automated Tests

Automated testing is a process that validates if software is functioning appropriately and meeting requirements before it is released into production[3].Carpus is tested via automated end-to-end tests. Appium, which is an open source test automation framework, is used for these tests. These tests generally verifies the fundamental features of carpus such as signing up, logging in, etc. For a requirement or a feature, both successful and unsuccessful test scenarios are applied. Below, you can find detailed test scenarios.

Test Case #	Test Case Description	Expected Result	Actual Result	Pass / Fail
1	Check a successful sign up	Signed Up Successfully!	Signed Up Successfully!	Pass
2	Check an unsuccessful signup due to duplicate	The username is already taken!	The username is already taken!	Pass
3	Check an unsuccessful signup due to non-bilkent mail address	You must enter Bilkent mail!	You must enter Bilkent mail!	Pass
4	Check a successful login	Logged In Successfully!	Logged In Successfully!	Pass
5	Check an unsuccessful login	Login Failed!	Login Failed!	Pass
6	Get error when creating trip without registering a car	You should register a car to create trip	You should register a car to create trip	Pass
7	Check a successful car registration	06 CAR 06	06 CAR 06	Pass

5.2 Manual Tests

Manual Testing is a type of software testing in which test cases are executed manually by a tester without using any automated tools. The purpose of Manual Testing is to identify the bugs, issues, and defects in the software application[4].Some parts of our application are tested manually. This is because the parts containing the map and GPS are too complicated for automation. Moreover; we have observed that quality attributes (usability, sustainability, etc.) of the app are better tested via manual tests. Below, you can find detailed test scenarios.

Test Case #	Test Case Description	Expected Result	Actual Result	Pass / Fail
8	Check whether GPS locates correctly	39.922580, 32.822318	39.922580, 32.822318	Pass
9	Check if the algorithm displays the optimum (closest) path	True	True	Pass
10	Check if a chat room saves and displays the whole chat history	True	True	Pass
11	Check if the user can create a trip and it is displayed in the trips page.	True	True	Pass
12	Check if the user can join a trip and this will be displayed in the trips section.	True	True	Pass
13	Check if the user can remove a user from the trip.	True	True	Pass
14	Check if the user can remove a	True	True	Pass

	car plate in my cars page.			

6. Maintenance Plan and Details

Maintenance for Carpus would be required depending on feedback given or new ideas and enhancements planned in the future. Moreover, adjustments on the code base may be required as plug-ins used in projects are updated. Our maintenance plan is detailed below.

6.1 Feedback Related Maintenance

User feedback and expert comments on Carpus will be taken seriously in case application will be public in the future. Those comments and feedback will be used in order to detect problems on the earlier stages and our team would benefit from user experience while solving the problems. According to our plan, users will be welcomed to comment on their user experiences via the comments tab on Google Play Store.

6.2 Google Play Store and Version Updates

As Carpus is implemented on Android Studio, our team is planning to make Carpus available on Google Play Store. User comments, feedback and rating system on Google Play Store would determine where and how to update our system.

7. Other Project Elements

7.1.Consideration of Various Factors in Engineering Design

7.1.1 Public Health:

As creation of the Carpus project began during COVID-19 pandemic, our team also considered the health aspect of the application. Carpus protects users from alternative public transportation options and provides an isolated campus trip.

7.1.2 Cultural Factors:

Carpus is only for Bilkent University students and its cultural impact is limited to this context. Therefore its cultural impact is limited to cultural interactions between Bilkent University students.

7.1.3 Global Factors:

Since Carpus is an application created specifically for Bilkent University students, it doesn’t have an impact on global factors.

7.1.4 Social Factors:

As Carpus will have a wide range of users from different departments or classes, it is possible to increase social interaction between students by carpooling. It also may increase social awareness on preventing air pollution since less cars may trip through the Bilkent University campus through the application.

7.1.5 Public Welfare:

Our application will be free to download and use. Version updates and upgrades will not be charged in any case. Therefıre it will not affect the budget and has no effect on public welfare.

7.1.6 Economic Factors:

Carpus mobile application utilizes Google Firebase platform for database related operations. If the total number of users that will use the application would exceed a certain number, our team needs to pay to handle more users. Also, some charges may apply to our side to utilize APIs for commercial use. However, those payments and charges will not be reflected on the customer side in any case.

7.2.Ethics and Professional Responsibilities

As the Carpus team we value ethics and also will demand our users to be responsible during the usage of the application as well. The application would have some confidential information about users but this information will not be shared with any people or company without user’s permission. Location information which will be required before and during the trips will also not be shared as well.

Moreover, in order to avoid any copyright issues, Carpus team preferred open-source products to develop the application and used all the products according to their licenses. We also will kindly ask users to be careful during the trip to the campus and to be respectful to each other.

Also, as we report our work in certain periods, we cited different sources following IEEE Citation Style guidelines [5].

7.3.Judgements and Impacts to Various Contexts

Discussed in part 7.1 and analysis report, Carpus is considered to have impacts mostly on environmental, public health, economic and social contexts. As less number of vehicles are planned to travel to Bilkent University campus, reducing environmental pollution, specifically air pollution, due to harmful gasses that vehicles emit is essential. Environmental impact of the application will also have an impact on public health context as those gasses are harmful for people. Also interaction between driver and users have an impact on social context. Finally we can mention that users can save money using Carpus as many users will travel to campus by a single car. Judgements and impacts to various contexts are summarized in the table below:

Judgements	Impact Level	Impact
Environmental Context	10	Reduce CO2 emission
Public Health	9	Physical health
Public Safety	8	Protection of private information
Public Welfare	8	Raise welfare level of students
Economic Context	5	Non-profit application
Social Context	5	Affected by social limitations
Cultural Context	5	Provide cultural interaction
Global Context	0	None

7.4 Teamwork Details

7.4.1 Contributing and functioning effectively on the team

From the beginning of the fall semester till the end of spring semester, we established a healthy communication and met regularly. In these meetings every group member is assigned to tasks related to the project considering an equal workload. Also every team member contributed to decisions made about analysis, design, implementation and feasibility.Tasks are done according to a schedule. We gave proper deadlines for each task and tried to obey our schedule as much as possible. We have discovered each other's strengths and weaknesses and divided tasks according to these. Each group member reviewed the other's work in order to increase the quality of the work done. We also cooperated in bigger tasks that cannot be divided into smaller subparts.

7.4.2 Helping creating a collaborative and inclusive environment

As indicated previously, tasks are distributed to group members according to their weaknesses and strengths. Throughout the development process, all group members acted considering these points and tried to compensate lacknesses regarding the project.

Each group member expressed their ideas and feedback clearly and regularly. Every opinion has been discussed and the optimum option in the given conditions has been chosen. Therefore, it can be stated that we have worked in a respectful and comfortable environment throughout the process.

7.4.3 Taking lead role and sharing leadership on the team

In order to provide an equal workload and encourage everyone to participate in leadership, we divided our project into six work packages and made each team member the leader of at least one package. Everyone has been included in each package to maintain the equal workload and awareness of each individual.

Face to face and online meetings have been periodically arranged. Each task in the project has been controlled by all members in the group and necessary changes and modifications have been done afterwards. Progress, code and implementation details are available for group members in a GitHub repository.

Six packages mentioned above are Analysis, High-Level Design, Implementation I, Low-Level Design, Implementation II and Final Report. Summary of the work packages given below and work plan is detailed using a Gantt chart.

WP#	**Work Package Title**	**Leader**	**Members Involved**
WP1	Analysis	Mehmet Yiğit Harlak	All
WP2	High-Level Design	Deniz Çalkan	All
WP3	Implementation I	Murat Sevinç	All
WP4	Low-Level Design	Veli Can Mert	All
WP5	Implementation II	İbrahim Furkan Aygar	All
WP6	Final Report	Murat Sevinç	All

Table 1: List of Work Packages

WP1: Analysis

Start Date: 22 September 2021 **End Date:** 15 November 2021			
Leader:	Mehmet Yiğit Harlak	**Members Involved:**	Deniz Çalkan İbrahim Furkan Aygar Murat Sevinç Veli Can Mert
Objectives: To determine requirements and have a comprehensive analysis of the project			
Tasks: **Task 1.1 Initial Research:** Put on a research to collect information about the current system **Task 1.2 Requirement Analysis:** Analyze and determine all requirements and specifications of the project. **Task 1.3 Analysis Report:** Write a report indicating the use cases, dynamic models and development plan of the project.			

Deliverables: **D1.1:** Specification Report **D1.2:** Analysis Report

Table 2: Work Package 1

WP2: High-Level Design			
Start Date: 16 November 2021 **End Date:** 24 December 2021			
Leader:	Deniz Çalkan	**Members Involved:**	İbrahim Furkan Aygar Murat Sevinç Mehmet Yiğit Harlak Veli Can Mert
Objectives: To have a comprehensive high-level design for the product which would help in the implementation stage			

Tasks: **Task 2.1 Analysis Evaluation:** Evaluate the analysis that has been done before to have a proper design. **Task 2.2 High-Level Design:** Decide on the design goals of the project and divide the project into subsystems.
Deliverables: **D2.1:** High-Level Design Report

Table 3: Work Package 2

WP3: Implementation I			
Start Date: 24 December 2021 **End Date:** End of the fall semester			
Leader:	Murat Sevinç	**Members Involved:**	İbrahim Furkan Aygar Deniz Çalkan Mehmet Yiğit Harlak Veli Can Mert

Objectives: To obtain a product which meets initial requirements and help users to present ideas and give feedbacks			
Tasks: **Task 3.1 Database Creation:** Create the database that will be used in the project. **Task 3.2 Front-End Implementation:** Implement the user interface for the Carpus. **Task 3.3 Back-End Implementation:** Implement some initial features for the first prototype.			
Deliverables: **D3.1:** First Prototype **D3.2:** First Presentation and Demo			

Table 4: Work Package 3

WP4: Low-Level Design
Start Date: 1st week of the spring semester **End Date:** 3rd week of the spring semester

Leader:	Veli Can Mert	**Members Involved:**	İbrahim Furkan Aygar Deniz Çalkan Mehmet Yiğit Harlak Murat Sevinç
Objectives: To have a comprehensive low-level design for the product which would help in the implementation stage			
Tasks: **Task 4.1 Architecture:** Decide on the architecture that will be used to implement the project. **Task 4.2 System Models:** Indicate the system models will be used to implement the project.			
Deliverables: **D4.1:** Low-Level Design Report			

Table 5: Work Package 4

WP5: Implementation II			
Start Date: 4th week of the spring semester **End Date:** 13th week of the spring semester			
Leader:	İbrahim Furkan Aygar	**Members Involved:**	Veli Can Mert Deniz Çalkan Mehmet Yiğit Harlak Murat Sevinç
Objectives: To finalize the implementation of the project			
Tasks: **Task 5.1 Implementation of Map Features:** Implementing map related features. **Task 5.2 Back-End Implementation:** Implementing all features and complete back-end implementation of the project. **Task 5.3 User Interface Implementation:** Complete front-end implementation of the project making necessary modifications and additions.			

Deliverables: **D5.1:** Final Product

Table 6: Work Package 5

WP6: Final Report			
Start Date: 10th week of the spring semester **End Date:** 13th week of the spring semester			
Leader:	Murat Sevinç	**Members Involved:**	İbrahim Furkan Aygar Deniz Çalkan Mehmet Yiğit Harlak Veli Can Mert
Objectives: To provide a detailed final report which includes comprehensive information about the product and project environment			

Tasks: **Task 6.1 Final Report:** Write a final report which explains the final product which gives detailed information about the system from all aspects. **Task 6.2 Maintenance and Testing:** Provide a maintenance and testing plan for the product
Deliverables: **D6.1:** Final Report

Table : Work Package 6

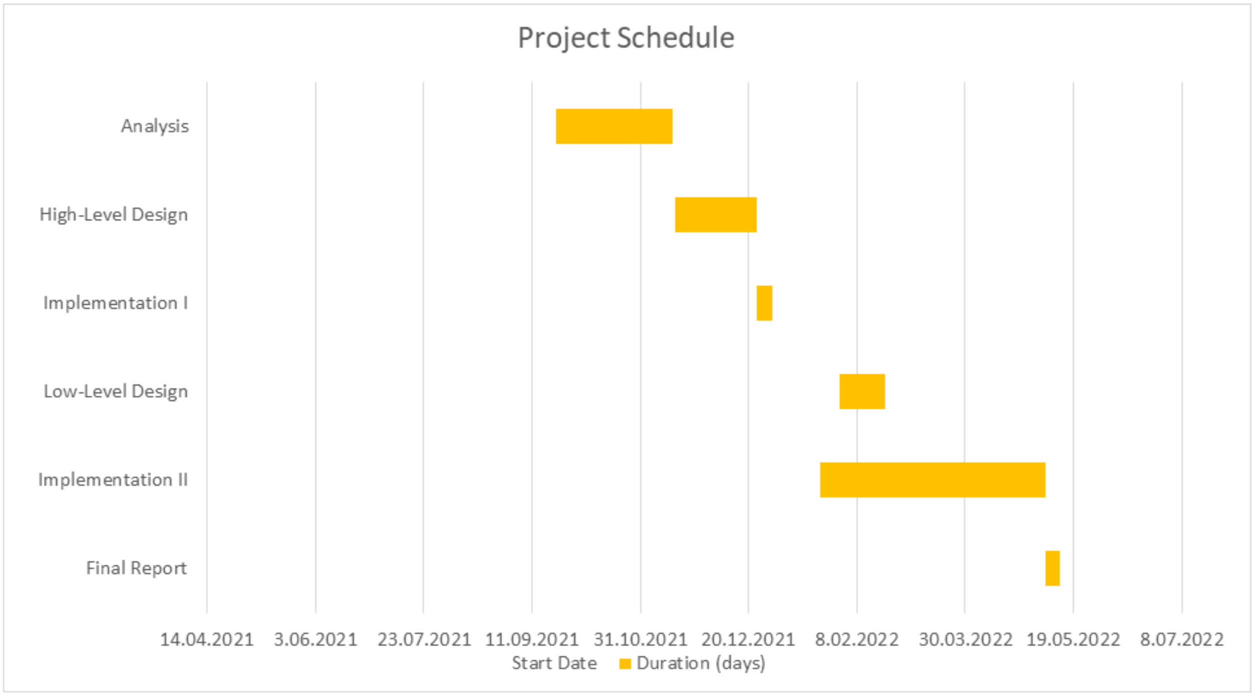


Figure 4: Gantt Chart for the Project Schedule

7.4.4 Meeting objectives

The objectives have been planned to be completed in previous stages of the project are mostly completed. Integration of weekly schedules of students to our application have not been done. Consequently, Carpus is now a successfully working mobile application that works on various versions of Android systems which provides a simple and responsive user interface. Precisely, the objectives that we met can be listed as follows:

- Implement a mobile application that includes create and join trip functionalities
- Implement a system meets the users via location services and routes to the campus
- Implement a group chat functionality for users to be able to communicate before the trip
- Implement necessary email validation steps in order to restrict the application only for Bilkent University students
- Implement a simple and well designed user interface

In conclusion, our project development process suited the schedule and we followed rules and necessities of the software development lifecycle in the correct way. Despite the limited time, we implemented most of the functionalities and have met our objectives as we obtained a finished product in the end.

7.5 New Knowledge Acquired and Applied

As our team decided to implement an Android application at the beginning of the project, we decided to use Android Studio as an IDE for both backend and frontend implementation and design stages. We utilized Firebase to handle the database and authentication part and used Gradle as the build automation tool.

Some of us were familiar with those tools before while others needed to be acknowledged how to use those tools in the project. Therefore, we benefited from a wide range of resources on the web such as tutorials and software documentation. Details can be found below:

7.5.1 Android Application Development

Although some of our team members were familiar with Android Studio and mobile application development, it took some time for others to learn the knowledge required to adapt the analysis and design to a mobile app. We have watched related tutorials and taken online courses in order to know what to do and how to do it.

7.5.2 Firebase Cloud

As we need to have a database to store user required user credentials, we have chosen Google's Firebase to authenticate users and apply required queries [6]. It has been beneficial for us to learn how to use Firebase in our projects as we can make use of it in our other projects.

8. Conclusion and Future Work

Consequently, we have designed a mobile Android application to provide Bilkent University students an environment where they can communicate, meet and travel through to the campus. Our team has planned the work will be done through two semesters and we completed the software development lifecycle almost respecting our plan. We considered sustainability and public health aspects of the project while thinking about our project topic.

For the future of the project, we can convert our application to a cross platform application and make it available for different operating systems. As more users download the application from the store we also need to make upgrades on the server side, therefore we need to invest on the application to be available to purchase. We may integrate the weekly schedule of the students by obtaining necessary permissions from the university administration. We can also make the application available for other students or make it entirely a public application. We may set a timeout in order to reduce the wait amount of the users. Number of users which will travel on the same vehicle can be limited through the app in the future as well.

9. Glossary

Appendix A - User Manual

After installing Carpus, sign up with your Bilkent Mail and Bilkent ID.

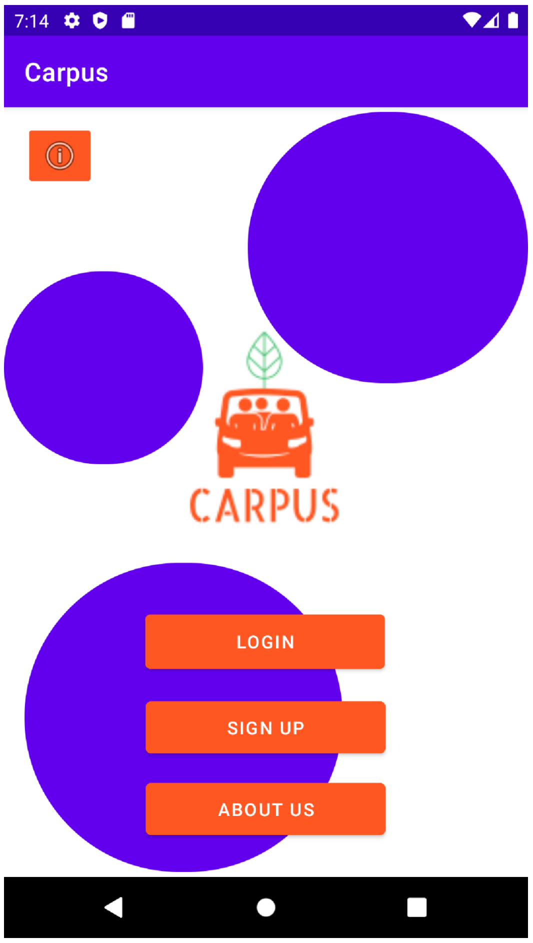


Figure 5: Main Page

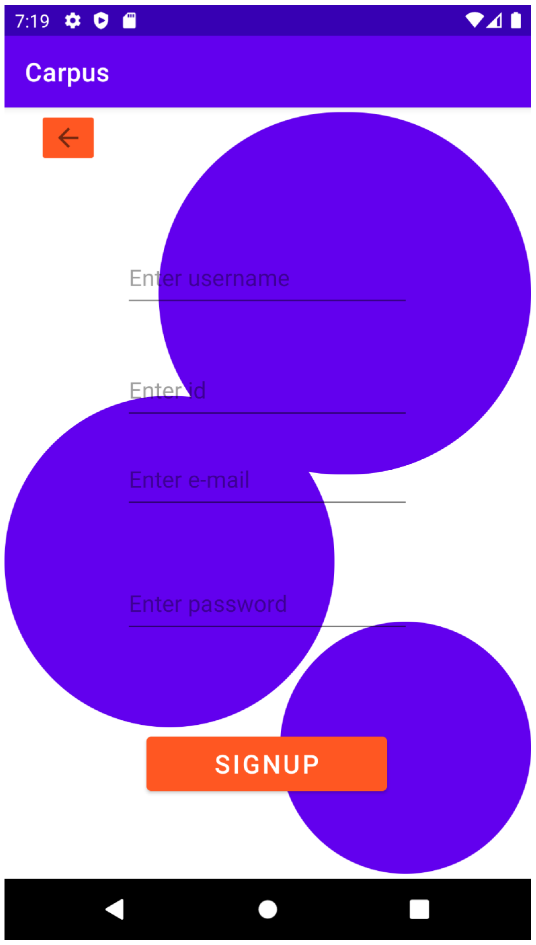


Figure 6: Signup Page

After signing up, login with your Bilkent Mail and password.

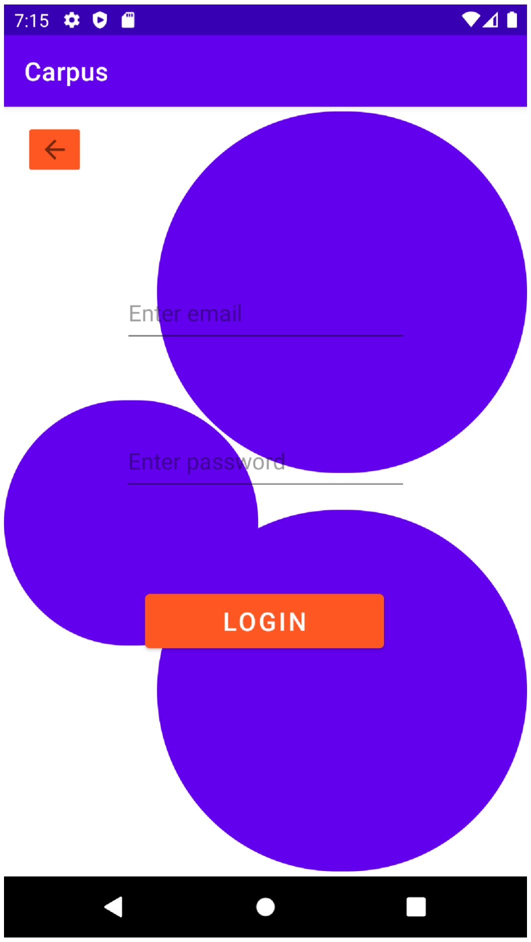


Figure 7: Login Page

After you login, the Main Menu will be ready for you to do various activities.

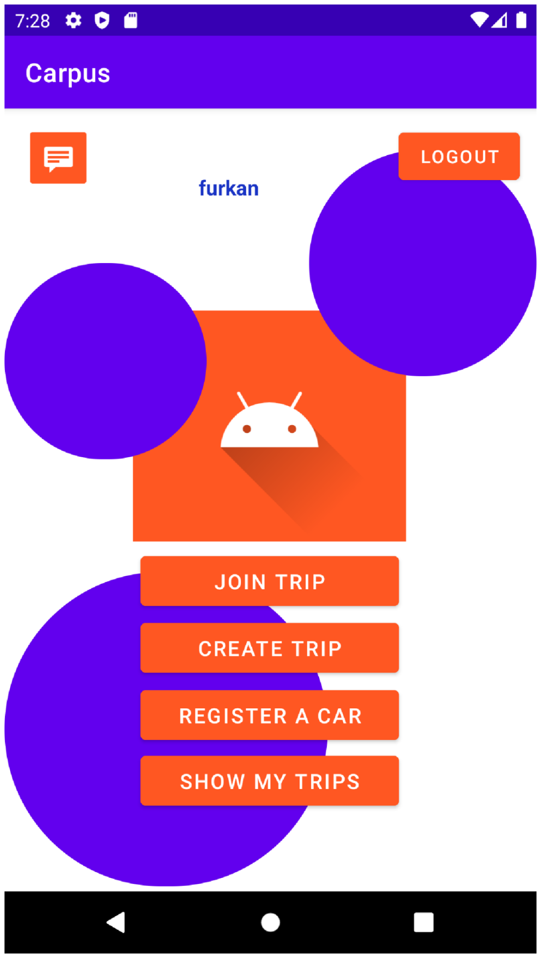
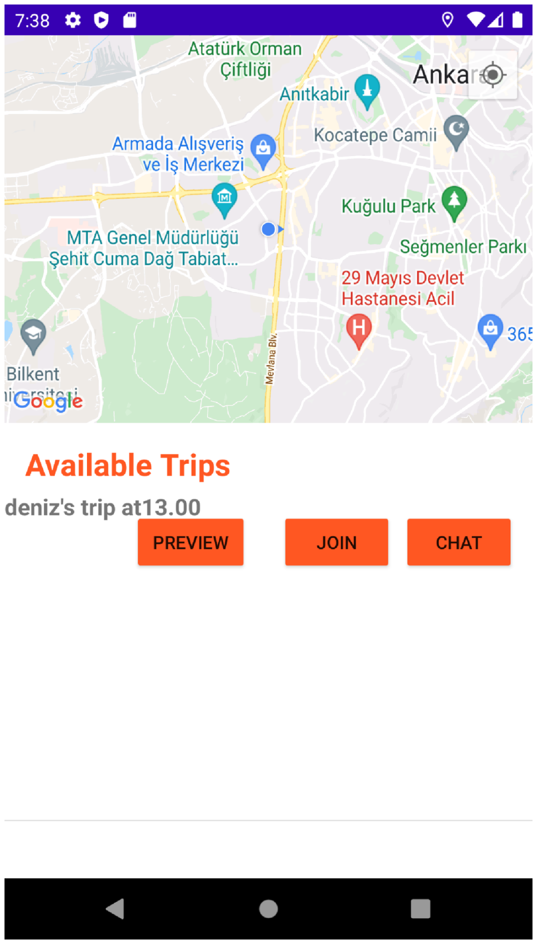


Figure 8: Main Menu Page

You can join existing trips. To do that, click on the “Join Trip” button. After that, available trips will be listed for you to join. You can preview the trip or even chat with the people that are registered to that trip. Furthermore, when you click on the “Join” button you will be registered to that trip and will be directed to the Main Menu Page.



Available Trips

deniz's trip at13.00

Figure 9: Join Trip Page

You can view your trips. To do that click on the “Show My Trips” button. After that, you can get information about specific trips if you click on the “Info” button or abandon a trip by clicking the “Delete” button.

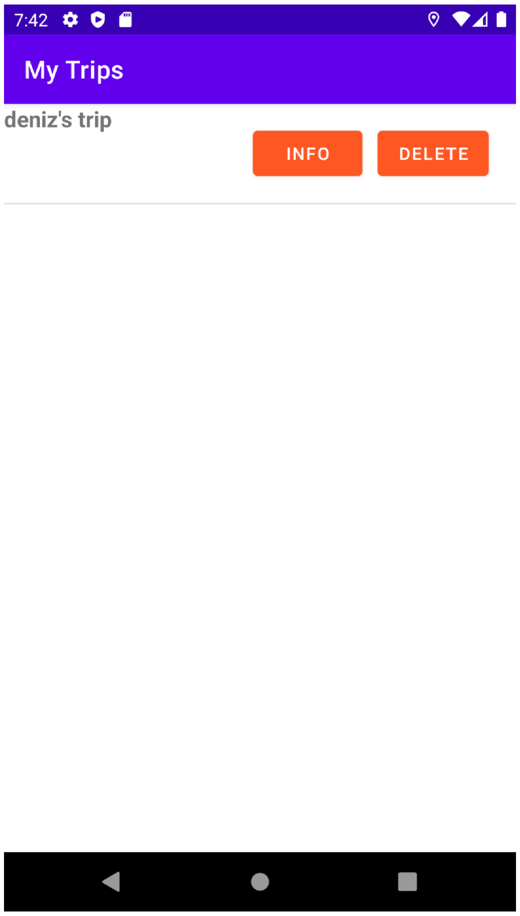


Figure 10: My Trips Page

You can create a new trip. To do that, you need to register a car. Click on the “Register a Car” button to register your car.

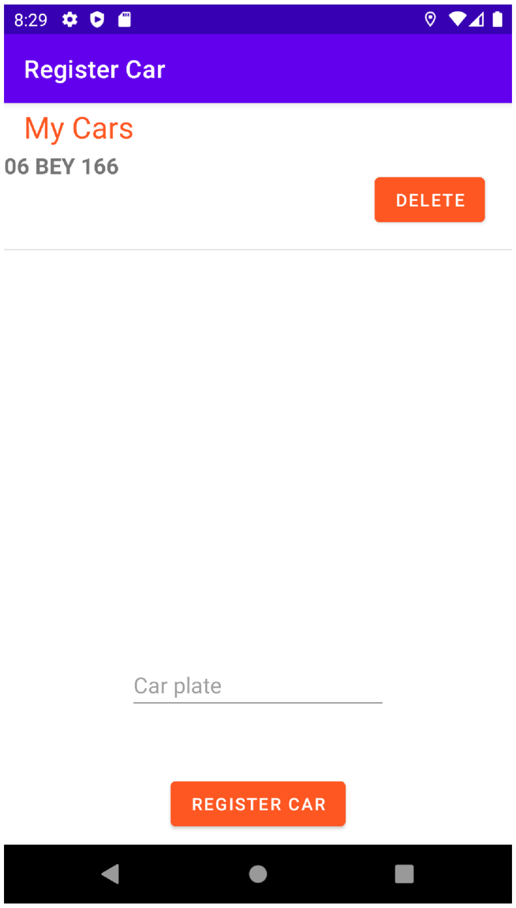
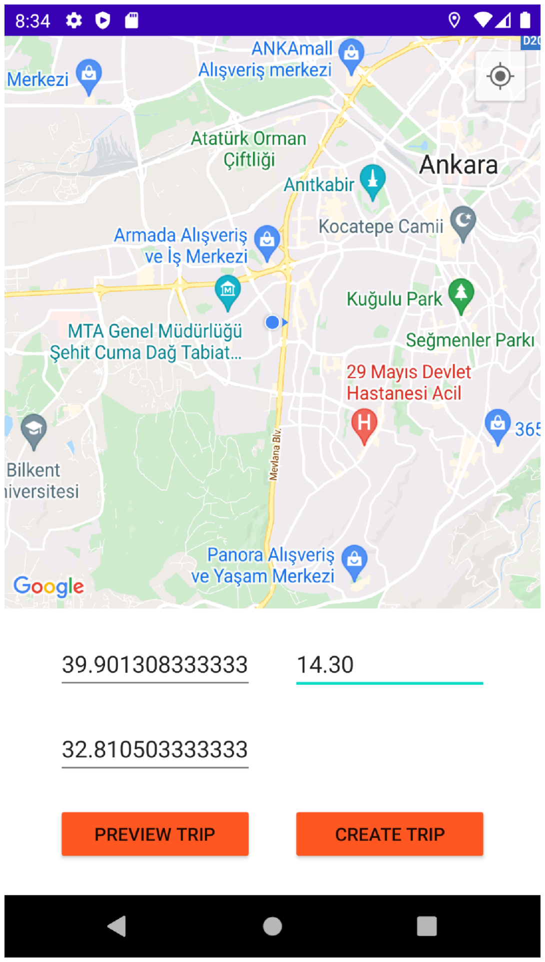


Figure 11: Register a Car Page

Now, by clicking the “Create Trip” button you can create a trip and set a starting location else the app will use your current location. Also you should provide the start time of the trip. Once you create the trip, other users will be able to see and join your trip.


8:34
ANKAmall Alışveriş merkezi
Merkezi
Atatürk Orman Çiftliği
Ankara
Anıtkabir
Kocatepe Camii
Armada Alışveriş ve İş Merkezi
Kuğulu Park
MTA Genel Müdürlüğü Şehit Cuma Dağ Tabiat...
Seğmenler Parkı
29 Mayıs Devlet Hastanesi Acil
Bilkent iversitesi
Panora Alışveriş ve Yaşam Merkezi
Google
PREVIEW TRIP
CREATE TRIP

39.901308333333

14.30

32.810503333333

Figure 12: Create Trip Page

You can preview your trip by clicking the “Preview Trip” button.

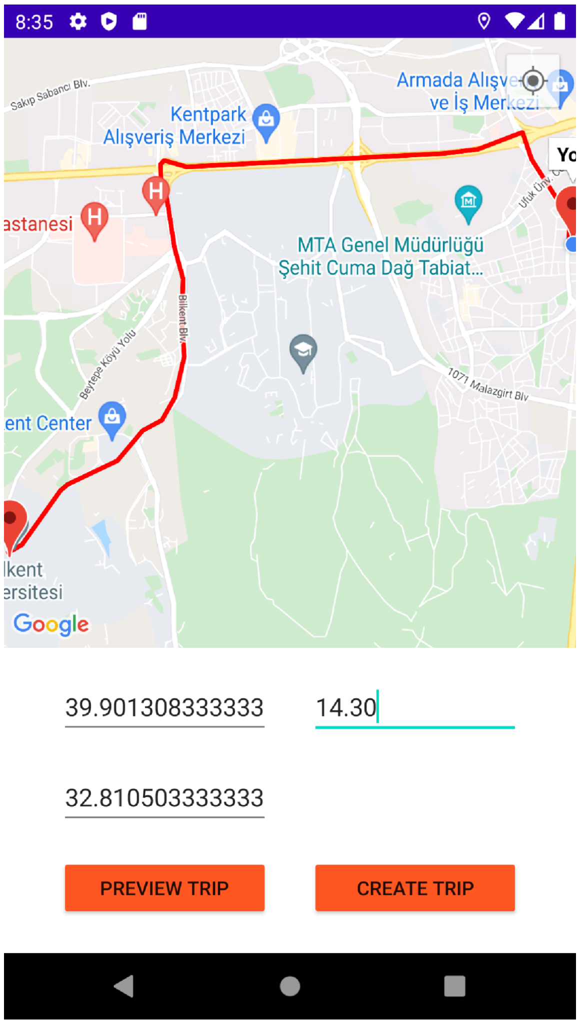


Figure 13: Preview Trip

You will be directed to the Main Menu after creating the trip. You can see your trip from the My Trips page and see the registered users.

10. References

[1] "The environmental impacts of cars explained", 2019. [Online]. Available: https://www.nationalgeographic.com/environment/article/environmental-impact. [Accessed: 3- May- 2022].

[2] "What Is Three-Tier Architecture". *Ibm.Com*, 2022. [Online]. Available: https://www.ibm.com/in-en/cloud/learn/three-tier-architecture. [Accessed: 3- May- 2022]

[3] "What Is Automated Testing ? ". [Online]. Available: https://www.techtarget.com/searchsoftwarequality/definition/automated-software-testing [Accessed: 3- May- 2022]

[4] "What Is Manual Testing ?". [Online]. Available: https://www.guru99.com/manual-testing.html [Accessed: 3- May- 2022]

[5] IEEE Periodicals Transactions/Journals Department, "IEEE Reference Guide - IEEE Author Center," IEEE Author Center. [Online]. Available: https://ieeeauthorcenter.ieee.org/wp-content/uploads/IEEE-Reference-Guide.pdf. [Accessed: 3- May- 2022].

[6] Google Firebase [Online]. Available: Firebase (google.com)

[Accessed: 3- May- 2022].