Bilkent University

Department of Computer Engineering

Senior Design Project

Carpus

Project Analysis Report

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

Supervisor: Özgür Ulusoy

Jury Members: Shervin Arashloo, Hamdi Dibeklioglu

Project Analysis Report
November 15, 2021

This report is submitted to the Department of Computer Engineering of Bilkent University in partial fulfillment of the requirements of the Senior Design Project course CS491/2.

1. Introduction

Nowadays, transportation is a big part of people's lives. People can travel easily from one place to another using different types of vehicles. Some of the most popular vehicles are undoubtedly cars and motorcycles [1]. Almost 67 million cars are produced per year and the number keeps getting bigger [2]. Recently, the increase in the use of these vehicles has started to harm the environment and the resulting traffic corrupts the daily lives of individuals. A typical car emits about 4.6 tons of carbon dioxide a year [3]. In addition, most of a car's environmental impact, around 85 percent, comes from fuel consumption and emissions of air pollution and greenhouse gases [4].

Our goal in creating Carpus is to reduce both traffic and gas emissions in Ankara by allowing university students to share vehicles alternately via a mobile application.

2. Current System

Currently, there are many ride sharing applications that are widely used in Europe and the US. BlaBlaCar is one of the most popular carpooling apps where drivers post a trip and few passengers join them [5]. Expenses can be shared by passengers and drivers and drivers can be rated by their talkativeness. ”Bla'' is used for quiet drivers, “bla bla” and “bla bla bla” is used for talkative drivers. There is also a “women only” option for women drivers. Carma Carpooling is another app which lets users share commuting costs [5]. Riders pay a standard price per mile and drivers do not make profit. Trees for Cars is one of the carpooling applications which aims to decrease CO2 emissions via carpooling [5]. Also there are other applications such as Uber and Lyft which are widely known. Full-time drivers are included in these apps and they make profit as passengers pay a certain amount of money depending on their route and location [5]. Zimride, Hitch and RideScout are some of the other carpooling applications.

As it can be seen from the examples above, carpooling apps are designed for a more general scope than our scope and targeted groups are different. Some applications permit drivers to make profit while others do not. Some applications aim to decrease CO2 emission while others do not. Aforementioned applications are not available in Turkey and none of them especially target university students in their scope. Also, some infrastructural features such as carpool lanes in the US are not available in Turkey as well.

3. Proposed System

3.1 Overview

Carpus will be designed as a mobile application for university students to use their personal vehicles together with other students to get to the university 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 for each weekday. According to the provided schedule, our algorithm will create groups from 3 or 4 users who live close to each other, then assign routes for each participant in a group to pick up the other participants and go to the university campus. Since a group's schedule is the same, next week this route will be assigned to another participant.

Active map service will be provided to the application users to keep track of where the car is. In this system, the shortest and most economical route to the campus will be provided. Also, for the safety of the users, the users must provide their university email addresses during the registration and the face of each user must be provided as a profile picture so that group members can recognize each other.

3.2 Functional Requirements

3.2.1 User Functionalities

- Users can register their cars to the system through its license plate number.
- Users can register their university ID numbers to the system.

- Users can mark their location on the map.
- Users can view other registered cars on the map.
- Users can share their destination and departure time on the map.
- Users can register for shared trips.
- Users can rate car owners which they have travelled with.
- Users can make weekly planning in certain groups.

3.2.2 Server Functionalities

- Server saves license plate numbers to the database.
- Server saves university ID numbers to the database.

3.2.3 University Administration Functionalities

- University administration can check whether car registered to the system is allowed to the campus or not.
- University administration can check whether the student is allowed to the campus or not.

3.3 Nonfunctional Requirements

3.3.1 Accessibility

- The application should be accessible from devices which has a proper version of Android.

3.3.2 Usability

- The application should be user friendly and easily adaptable for users and university administration in terms of user interface.

3.3.3 Availability

- The system should be available and operate consistently.

3.3.4 Security

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

3.3.5 Scalability

- The application should operate within 10.000 users.

3.3.6 Legality

- The application should have necessary permissions from university administration to include student ID numbers.

3.3.7 Performance

- The application should operate fast in terms of launch and login time.
- The application should display available trips in at most 15 seconds.
- Battery consumption for the application should be low.

3.3.8 Response Time

- Response time for the application should be at most 5 seconds in order to improve user experience.

3.4 Pseudo Requirements

3.4.1 Implementation Requirements

- Object-oriented programming will be used as the programming paradigm of the application.
- Android Studio will be the integrated development environment for the team.
- Git will be used as the version control system and Android Studio project will be authenticated with GitHub.
- Java will be used as the programming language for the implementation of the application.
- Google Maps API will be used for location and destination related features.
- Object classes are subject to change depending on the implementation.

3.4.2 Economic Requirements

- The application will be free for all university students.
- Development tools will be utilized for free.

3.4.3 Sustainability Requirements

- In the further stages of the development, feedback from other parties may be taken into account and necessary changes or updates can be made.

3.4.4 Administrative Constraints

- Necessary permissions need to be obtained from Bilkent University administration and Metropolitan Municipality of Ankara if needed.
- GPS permissions need to be given by users in order to access locations of users and draw routes.

3.4.5 Time Constraints

- The implementation of the application needs to be done before May 2022 and necessary updates on feedback given must be made in order to have a fully-functioning application before CSFair.
- A prototype version must be implemented before the end of the 2021 Fall Semester.

3.5 System Models

3.5.1 Scenarios

Scenario 1	Sign Up
Actors	Driver, Passenger
Entry Conditions	User opens the app and clicks on the Sign Up button.
Exit Conditions	User returns to the main menu or closes the app.
Flow of Events	1. The Sign Up page opens. 2. User enters the necessary information. 3. User clicks on the Sign Up button. 4. System creates user’s account.

Scenario 2	Log In
Actors	Driver, Passenger
Entry Conditions	User opens the app and clicks on the Login button.
Exit Conditions	User returns to the main menu or closes the app.
Flow of Events	1. The Login page opens. 2. User enters username and password. 3. System checks the entered information. 4. User logs in.

Scenario 3	Log Out
Actors	Driver, Passenger
Entry Conditions	User opens the app.
Exit Conditions	User returns to the main menu or closes the app.
Flow of Events	1. User clicks on the Logout button. 2. User logs out.

Scenario 4	Edit Profile
Actors	Driver, Passenger
Entry Conditions	User opens the app and logs in.
Exit Conditions	User returns to the main menu or closes the app.
Flow of Events	1. User clicks on the Edit Profile button.

	2. The Edit Profile page opens. 3. User makes the desired changes. 4. System saves the changes.

Scenario 5	Apply for a Trip
Actors	Driver, Passenger
Entry Conditions	User opens the app and logs in.
Exit Conditions	User returns to the main menu or closes the app.
Flow of Events	1. User clicks on the View On Map button. 2. User clicks on the Apply for a Trip button. 3. User chooses time and date. 4. If there are suitable trips, the app offers these trips to the user. 5. Else, the app offers alternative trips. 6. User selects the trip. 7. System adds user to the trip.

Scenario 6	Create Trip
Actors	Driver
Entry Conditions	User opens the app and logs in.
Exit Conditions	User returns to the main menu or closes the app.
Flow of Events	1. User clicks on the View On Map button.

	2. User clicks on the Create Trip button. 3. System checks whether the user has a registered car. 3. User chooses time and date. 4. System creates the trip.

Scenario 7	Rate Driver and Comment on a Trip
Actors	Passenger
Entry Conditions	User opens the app and logs in.
Exit Conditions	User returns to the main menu or closes the app.
Flow of Events	1. User clicks on the Past Trips button. 2. User selects a past trip. 3. User rates the driver and comments on the trip. 4. System updates driver’s rating.

Scenario 8	Join a Group Chat
Actors	Driver, Passenger
Entry Conditions	User opens the app and logs in.
Exit Conditions	User returns to the main menu or closes the app.
Flow of Events	1. User clicks on the Groups button. 2. User selects a chat. 3. User reads, writes and sends messages to the chat.

3.5.2 Use Case Model

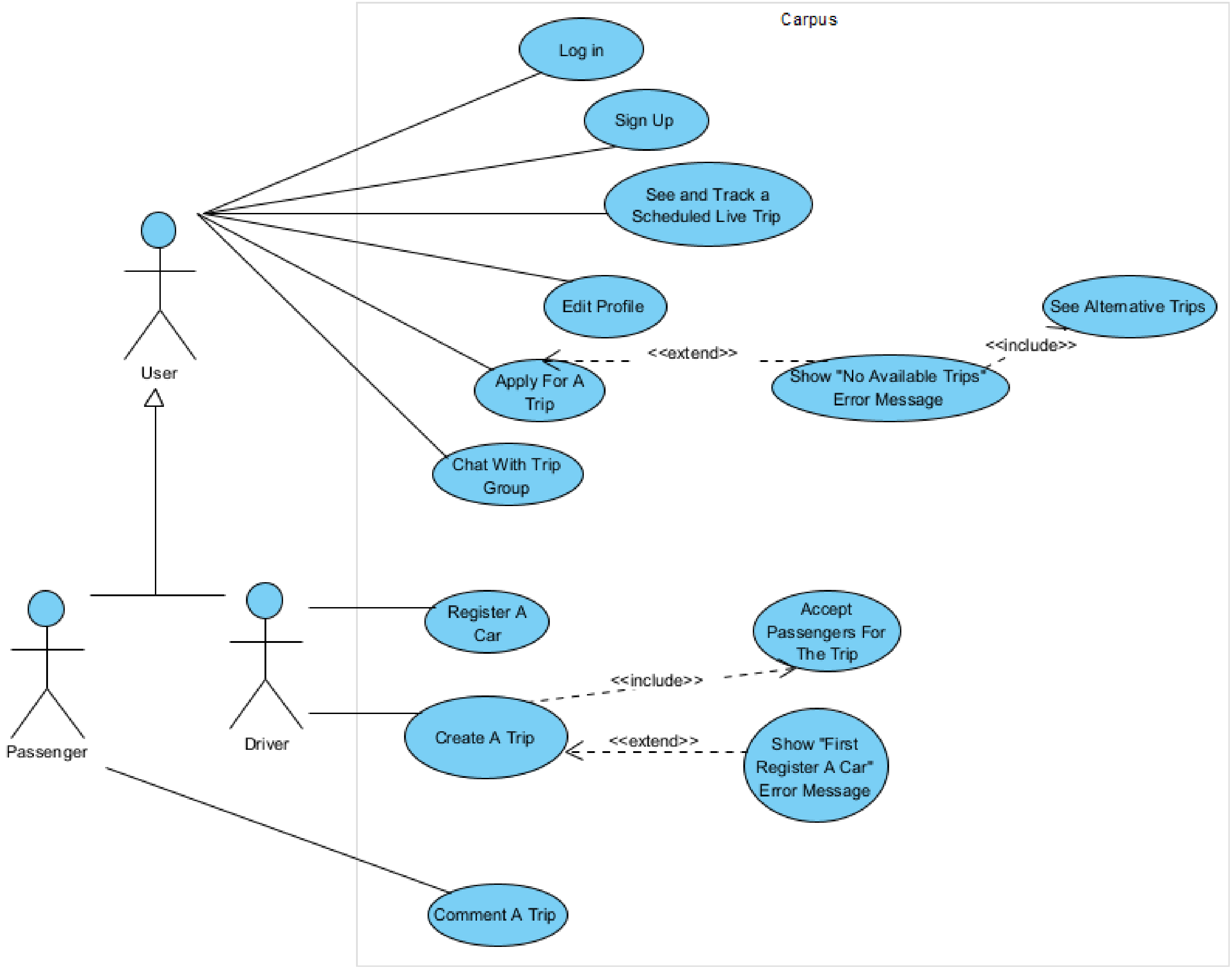


Figure 1: Use-Case Model of Carpus

3.5.3 Object and Class Model

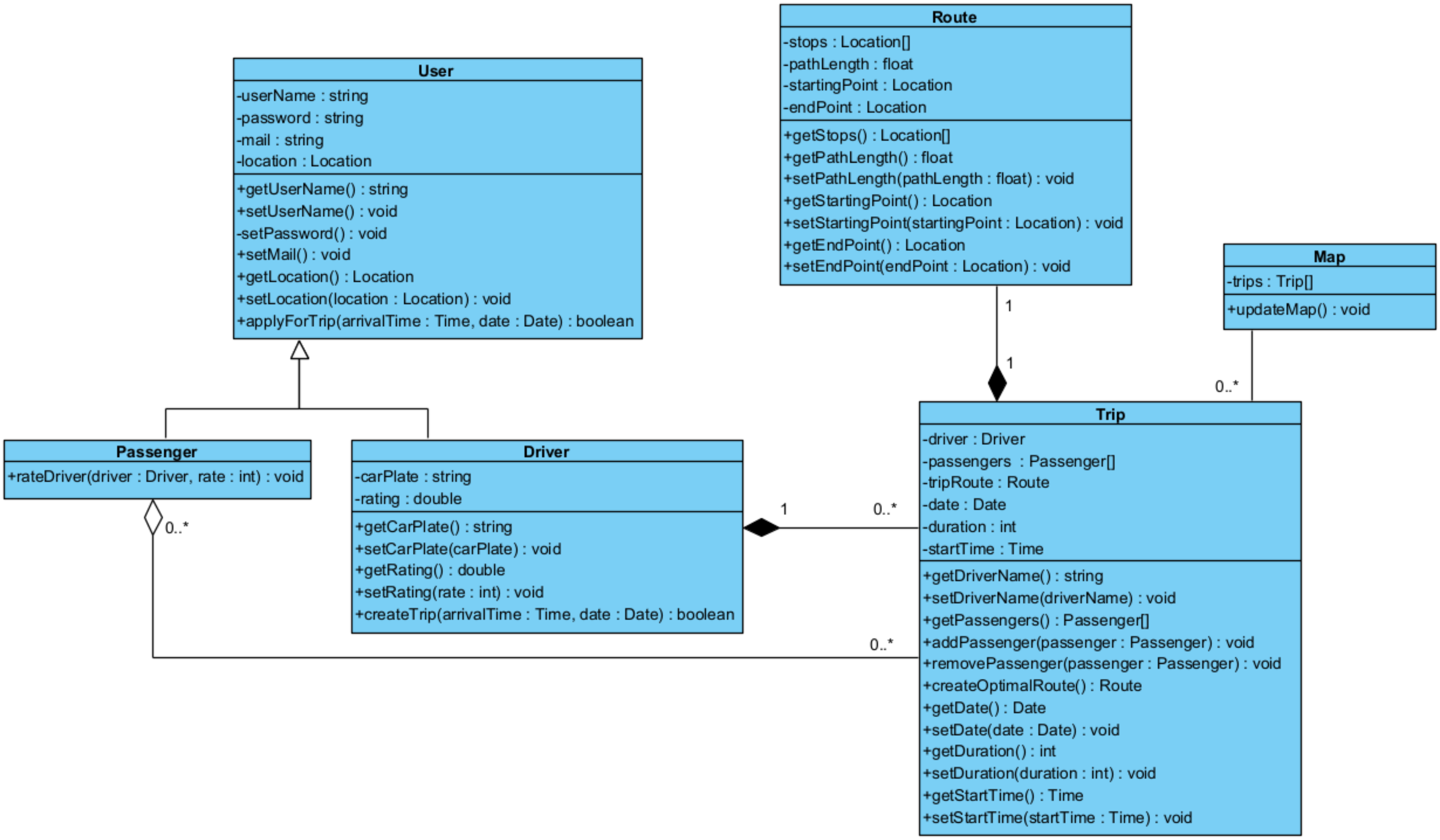

Route
-stops : Location[]
-pathLength : float
-startingPoint : Location
-endPoint : Location
+getStops() : Location[]
+getPathLength() : float
+setPathLength(pathLength : float) : void
+getStartingPoint() : Location
+setStartingPoint(startingPoint : Location) : void
+getEndPoint() : Location
+setEndPoint(endPoint : Location) : void
User
-userName : string
-password : string
-mail : string
-location : Location
+getUserName() : string
+setUserName() : void
-setPassword() : void
+setMail() : void
+getLocation() : Location
+setLocation(location : Location) : void
+applyForTrip(arrivalTime : Time, date : Date) : boolean
Map
-trips : Trip[]
+updateMap() : void
1
1
0..*
Trip
-driver : Driver
-passengers : Passenger[]
-tripRoute : Route
-date : Date
-duration : int
-startTime : Time
+getDriverName() : string
+setDriverName(driverName) : void
+getPassengers() : Passenger[]
+addPassenger(passenger : Passenger) : void
+removePassenger(passenger : Passenger) : void
+createOptimalRoute() : Route
+getDate() : Date
+setDate(date : Date) : void
+getDuration() : int
+setDuration(duration : int) : void
+getStartTime() : Time
+setStartTime(startTime : Time) : void
Passenger
+rateDriver(driver : Driver, rate : int) : void
0..*
Driver
-carPlate : string
-rating : double
+getCarPlate() : string
+setCarPlate(carPlate) : void
+getRating() : double
+setRating(rate : int) : void
+createTrip(arrivalTime : Time, date : Date) : boolean
1
0..*
0..*

Figure 2: Object and Class Model of Carpus

3.5.4 Dynamic Models

3.5.4.1 Activity Diagram

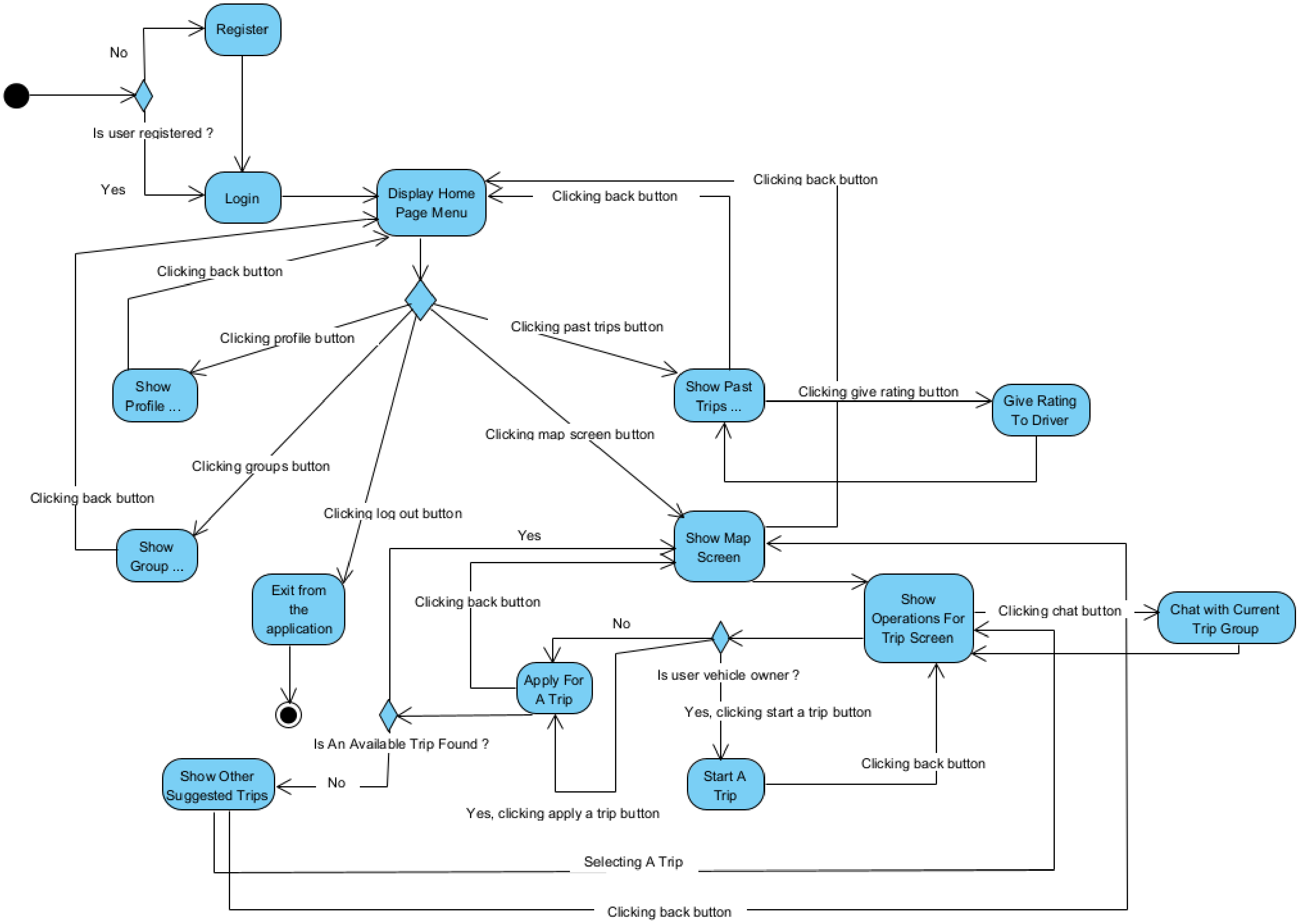


Figure 3: Activity Diagram of Carpus

3.5.4.2 State Diagram

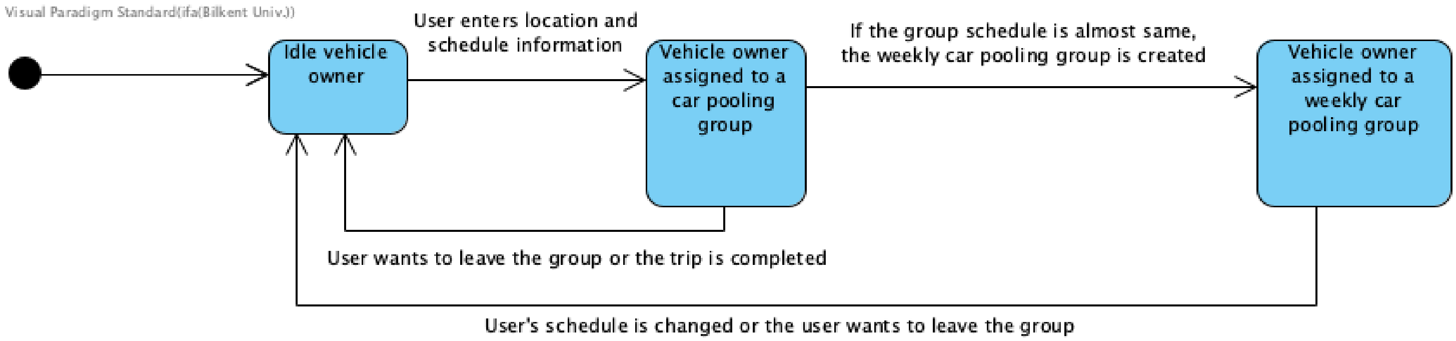

Visual Paradigm Standard(ifa(Bilkent Univ.))
Idle vehicle owner
User enters location and schedule information
Vehicle owner assigned to a car pooling group
If the group schedule is almost same, the weekly car pooling group is created
Vehicle owner assigned to a weekly car pooling group
User wants to leave the group or the trip is completed
User's schedule is changed or the user wants to leave the group

Figure 4: Driver State Diagram

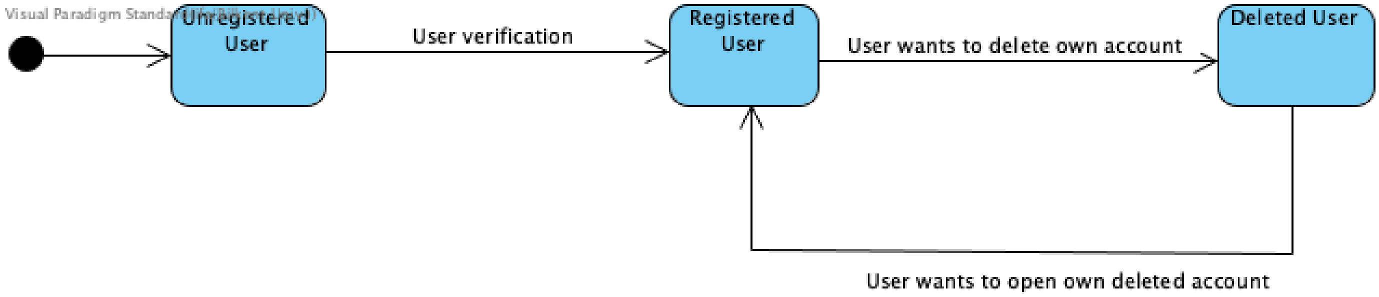

Visual Paradigm Standard(ifa(Bilkent Univ.))
Unregistered User
User verification
Registered User
User wants to delete own account
Deleted User
User wants to open own deleted account

Figure 5: Registered User State Diagram

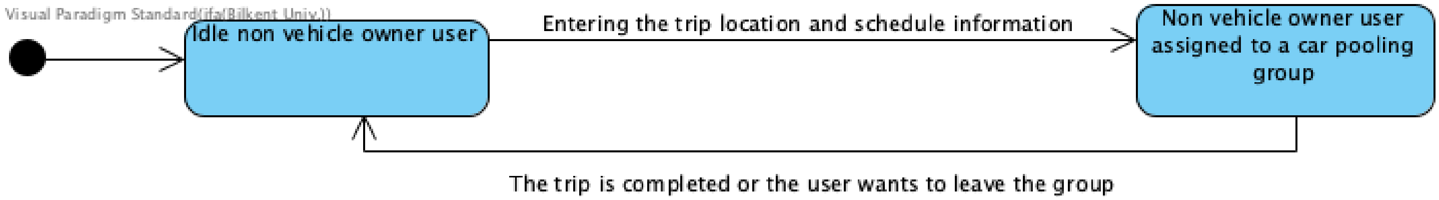

Visual Paradigm Standard(ifa(Bilkent Univ.))
Idle non vehicle owner user
Entering the trip location and schedule information
Non vehicle owner user assigned to a car pooling group
The trip is completed or the user wants to leave the group

Figure 6: Passenger State Diagram

3.5.4.3 Sequence Diagram

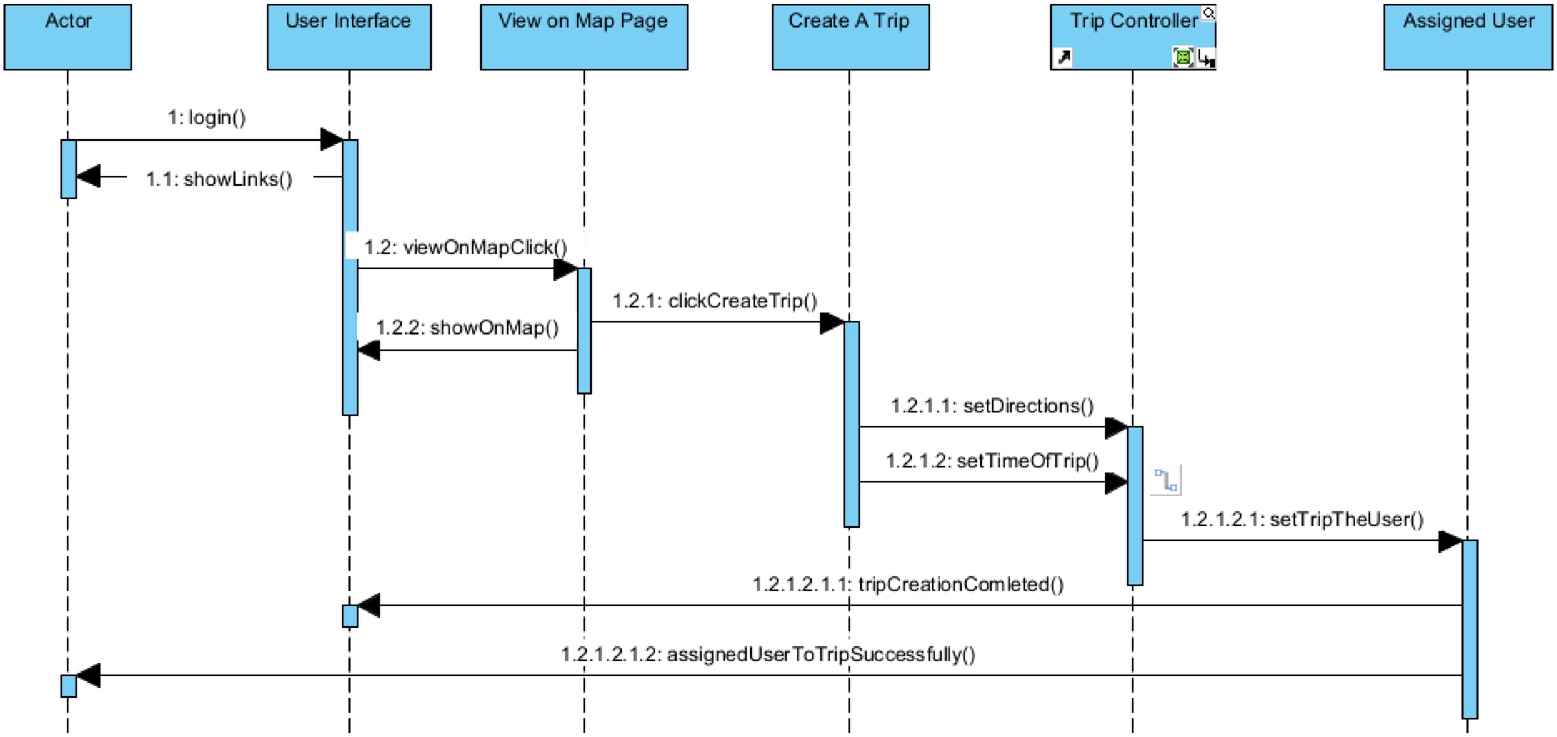


Figure 7: Sequence Diagram of Starting a Trip

3.5.5 User Interface - Navigational Paths and Screen Mock-ups

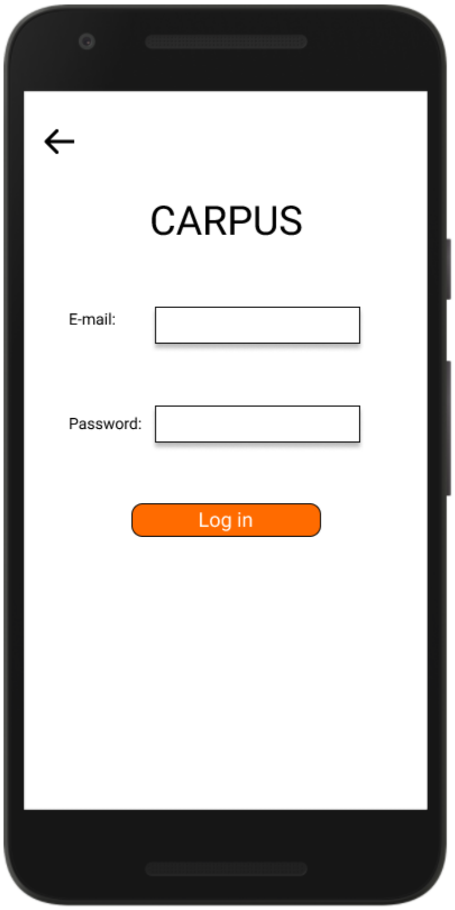


Figure 8: Login Page Mockup

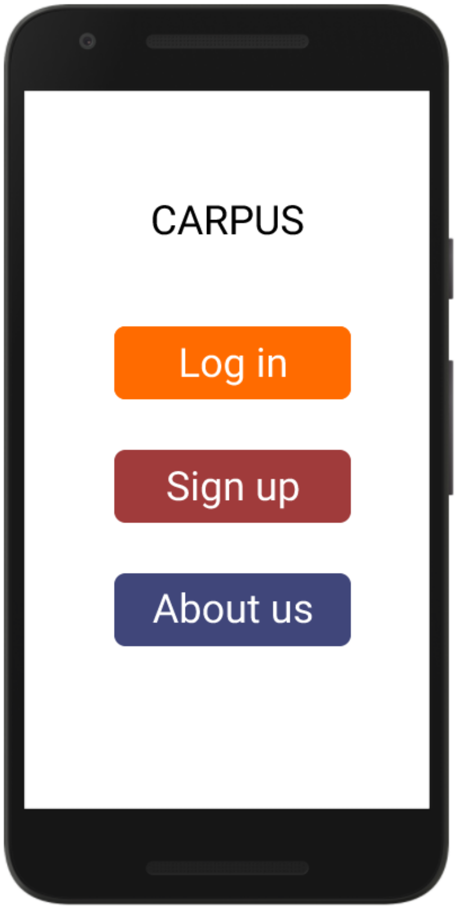


Figure 9: Main Page Mockup

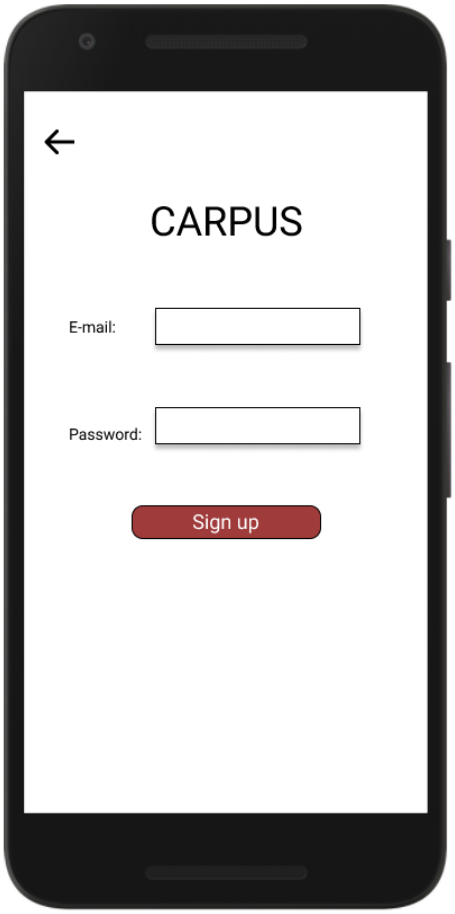


Figure 10: Sign Up Page Mockup

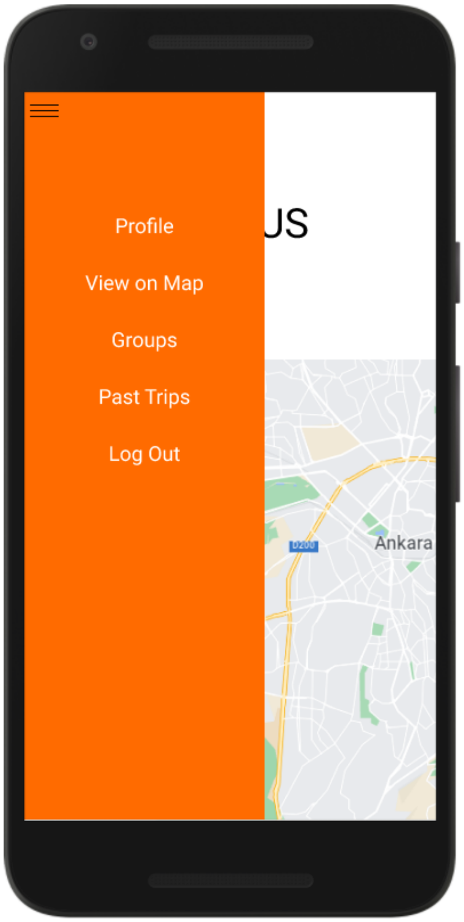
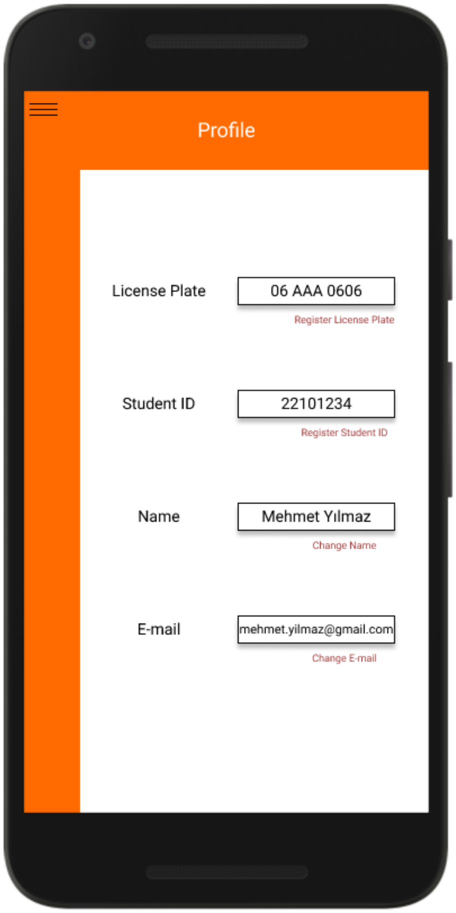


Figure 11: Menu View Mockup



License Plate: 06 AAA 0606

Student ID: 22101234

Name: Mehmet Yılmaz

E-mail: mehmet.yilmaz@gmail.com

Figure 12: Profile Page Mockup

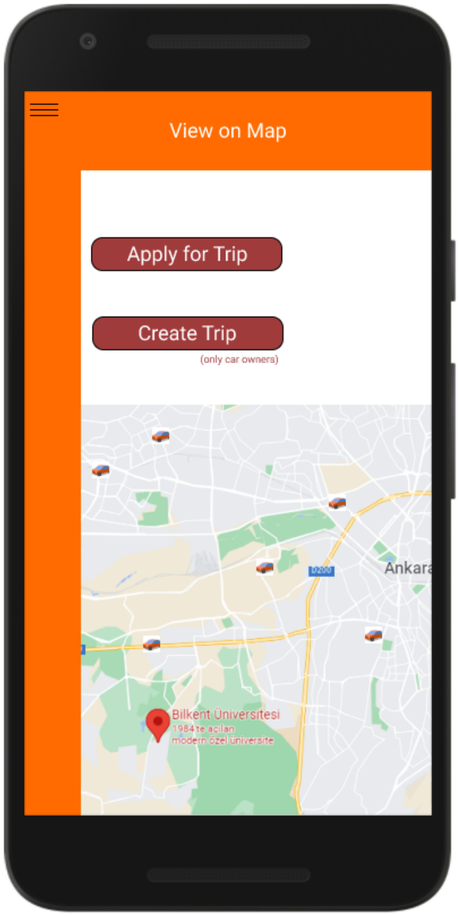
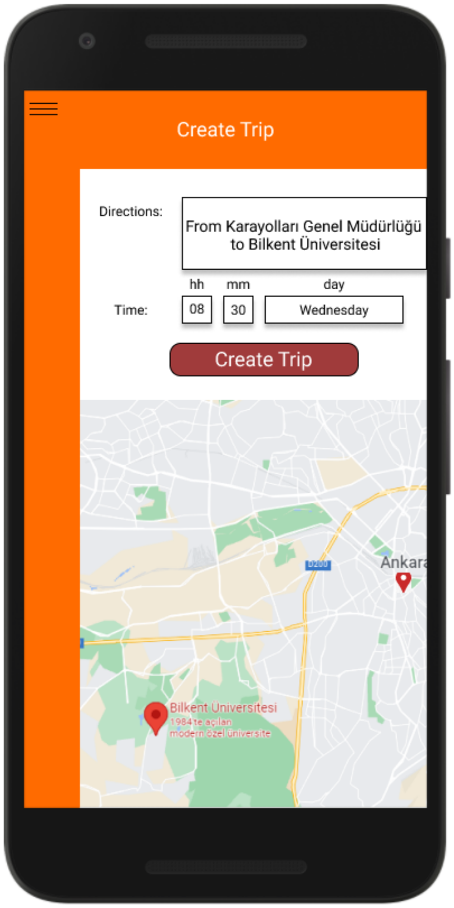


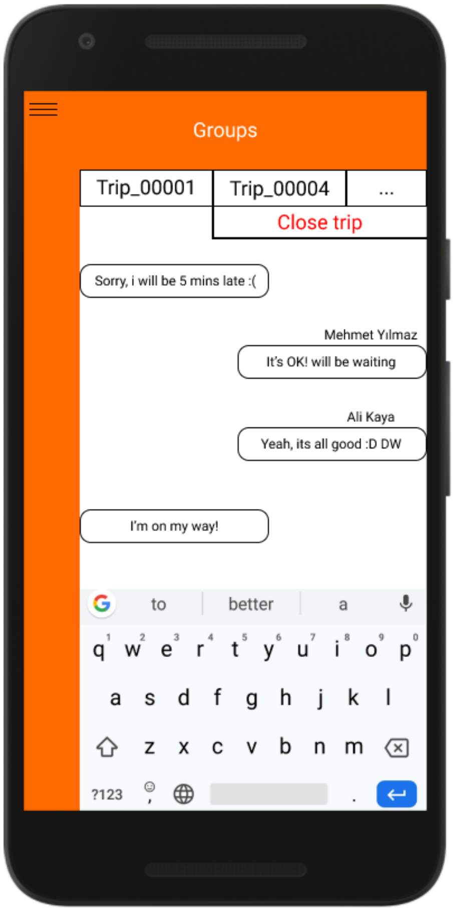
Figure 13: Map View Mockup



Directions: From Karayolları Genel Müdürlüğü to Bilkent Üniversitesi

Time: 08 30 Wednesday

Figure 14: Create Trip Page Mockup



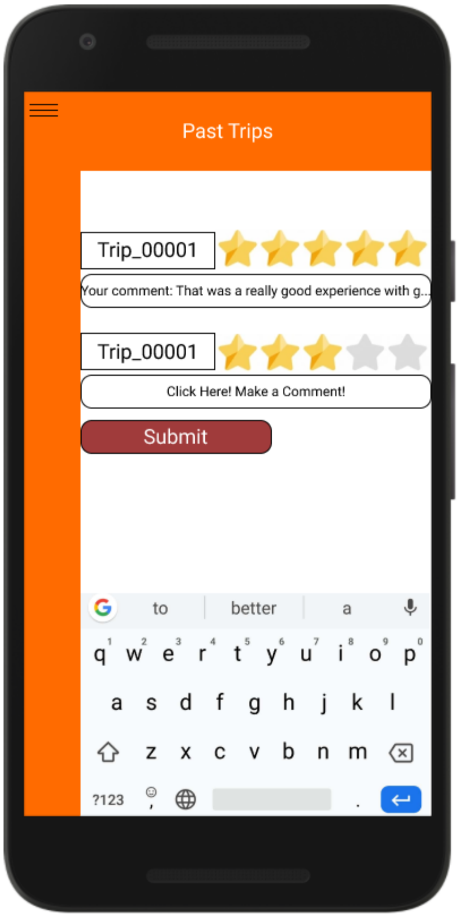
Sorry, i will be 5 mins late :(

Mehmet Yılmaz
It's OK! will be waiting

Ali Kaya
Yeah, its all good :D DW

I'm on my way!

Figure 15: Groups Page Mockup


Past Trips

Trip_00001

Your comment: That was a really good experience with g...

Trip_00001

Figure 16: Past Trips Page Mockup

4. Other Analysis Elements

4.1. Consideration of Various Factors in Engineering Design//yiğit

There are several factors that may affect the implementation and development process of Carpus that have to be taken into account. These factors are explained in the sections below.

4.1.1 Environmental Factors

The most important goal of Carpus is to reduce gas and traffic emissions as mentioned before. Carpooling will be a sustainable alternative for the environment.

Moreover, privileges that cars that join Carpus would get, such as private parking slots, would affect the design process.

4.1.2 Public Health

As carpooling via Carpus provides less CO2 emission with decreasing number of vehicles in city and campus transportation, to have less polluted air is not only beneficial for the environment but also essential for public health of people who live on campus.

Additionally, as COVID-19 measures taken are still valid in Turkey and Bilkent University campus, eligibility of potential users in terms of COVID-19 may be taken into consideration.

4.1.3 Public Safety

Private and sensitive information will be kept confidential and will not be shared with any third party application. Required permissions will be obtained from the user in case it is necessary.

4.1.4 Public Welfare

Carpus will also help people who do not own private cars to travel between city and campus in a more comfortable manner. Therefore it will contribute to public welfare.

4.1.5 Economic Factors

Carpus will be an application that is free to download and use. It will not be for profit for its users or its developers.

4.1.6 Social Factors

As people who carpool will meet before the trip, Carpus would be a way to socialise and also carpooling will be affected by social constraints.

4.1.7 Cultural Factors

Carpus could provide a cultural interaction between users who travel together in case people from different cultures come together.

4.1.8 Global Factors

We cannot mention any global impact that Carpus would have as users will only be Bilkent University students but our development process may be affected by the carpooling applications that are available all around the world.

	Effect Level	Effect
Environmental Factors	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 Factors	5	Non-profit application
Social Factors	5	Affected by social limitations
Cultural Factors	5	Provide cultural interaction

Global Factors	0	None

Table 1: Factors and Effects

4.2. Risks and Alternatives

Carpus aims to provide a social platform for Bilkent University students to set time and location before the trip. Basically, drivers post a trip which features date, time and location and other students who are near to that location attend it. In order to have a properly working application, locations of both driver and participants should be detected precisely. In case of wrong or irrelevant detection of location people may struggle to communicate and find each other. To overcome this issue, adding a chat feature for users who want to travel on the same trip may be a solution.

If many people apply for the same trip, that would also raise problems as the number of passengers will exceed the capacity of the vehicle. Also having many passengers could be inappropriate and against to COVID-19 measures. Therefore, drivers should be able to accept or reject people who apply for their trips, which can be added as a feature to the application.

	Likelihood	Effect on the project	B Plan Summary
Incorrect location detection	5	Passengers and driver could not meet	Enabling users to communicate via chat
Excessive number of passengers	5	Some passengers may not be able to	Enabling drivers to accept or reject passengers

		travel despite meeting criteria	

Table 2: Risks

4.3. Project Plan

As the project team, we will follow a formal software development cycle which includes analysis, design, implementation, testing and maintenance stages. We are going to analyze and design our project and plan to have a prototype by the end of the fall semester. Afterwards, we will test our product according to the feedback we get from users and make sure our product meets the requirements and functionalities that have been indicated before.

We will have six work packages: Analysis, High-Level Design, Implementation I, Low-Level Design, Implementation II and Final Report.

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 3: 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 4: 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 5: 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 6: 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 7: 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 8: 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 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 9: Work Package 6

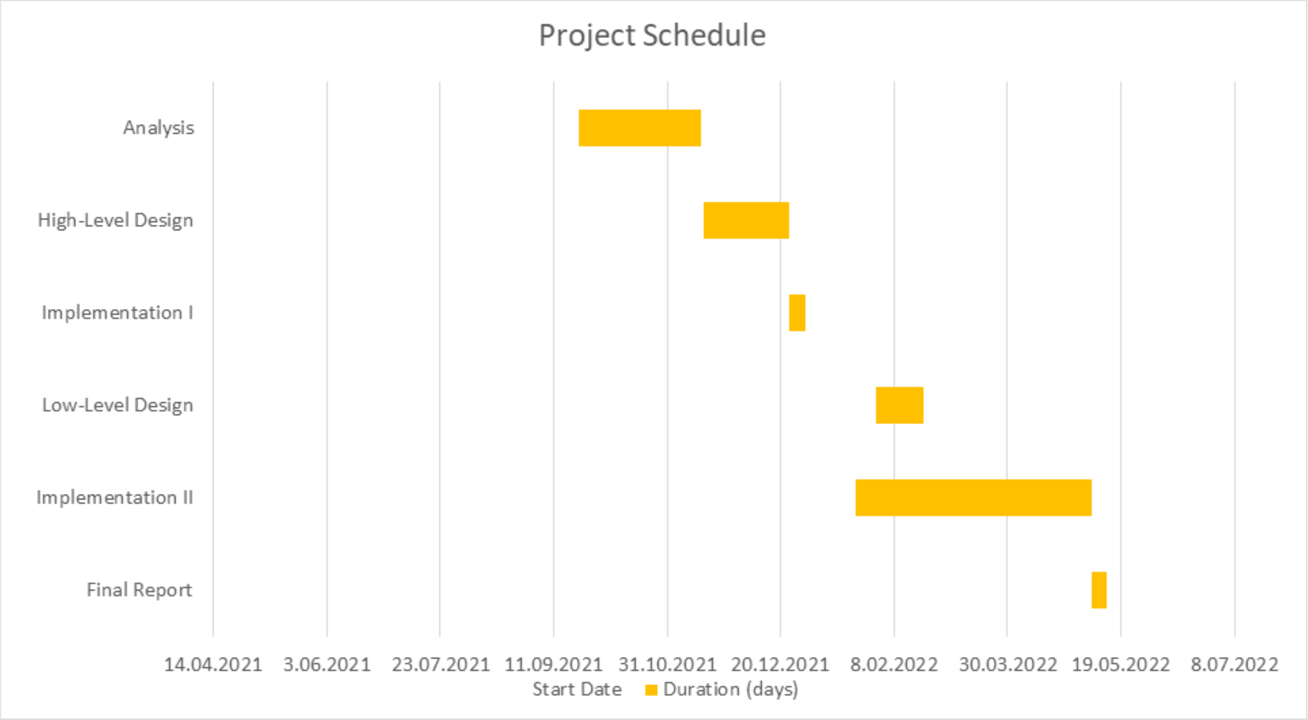


Figure 17: Gantt Chart for the Project Schedule

4.4. Ensuring Proper Teamwork

Overall work of the project has been divided into six packages and each team member is marked as the leader of one package at least one time. By doing this, our goal is to encourage each team member to lead a stage. Each team member will have equal workload as everybody in the group is responsible for every stage. In this way everybody would contribute to each stage by the work they do or feedback they give. Face to face and online meetings will be periodically arranged. Each task in the project will be controlled by all members in the group and necessary changes and modifications will be done after discussions. Progress, code and implementation details will be available for group members in a GitHub repository.

4.5. Ethics and Professional Responsibilities

Carpus needs to access some confidential information about users. This process will be done by getting necessary permissions from the user and this information will not be shared with the third parties. Obtained location information will also not be shared with any third party application or device. Moreover, in order to avoid copyright issues, open source tools will be used in the development process.

4.6. Planning for New Knowledge and Learning Strategies

As the development team, our goal is to enhance our Java skills and to develop an Android application from scratch by applying all formal software development steps. Developing an application that can be used in the real-life is new for some members in our team and requires new knowledge. Also tools and software we will utilize during the development process require a technical background.

Self-learning and researching will be two main learning strategies that we will use. We will benefit from academic resources to learn the project environment and common needs of the people and watch tutorials and read online articles to achieve technical information.

5. Glossary

API: Application Programming Interface.
GPS: Global Positioning System.
,

6. References

[1] "10 Most Common Modes Of Transportation - 10 Most Today", *10 Most Today*, 2018. [Online]. Available: https://10mosttoday.com/10-most-common-modes-of-transportation/. [Accessed: 10- Oct- 2021].

[2] "The World Counts", *Theworldcounts.com*, 2021. [Online]. Available: https://www.theworldcounts.com/challenges/consumption/transport-and-tourism/cars-impact-on-the-environment/story. [Accessed: 10- Oct- 2021].

[3] ”Greenhouse Gas Emissions from a Typical Passenger Vehicle”, [Online] Available: https://www.epa.gov/greenvehicles/greenhouse-gas-emissions-typical-passenger-vehicle, [Accessed: 10 - Oct- 2021].

[4]]"The environmental impacts of cars explained", 2019. [Online]. Available: https://www.nationalgeographic.com/environment/article/environmental-impact. [Accessed: 10- Oct- 2021].

[5]E. Carson, “10 rideshare apps to Crowdsource your commute,” *TechRepublic*, 10-Apr-2014. [Online]. Available: https://www.techrepublic.com/article/10-rideshare-apps-to-crowdsource-your-commute/. [Accessed: 13-Nov-2021].