Bilkent University

Department of Computer Engineering

Senior Design Project

Carpus

High-Level Design Report

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

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

High-Level Design Report
December 24, 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.

Table of Contents

1. Introduction

1.1 Purpose of the System

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. A typical car emits about 4.6 tons of carbon dioxide a year [1]. In addition, most of a car's environmental impact, around 85 percent, comes from fuel consumption and emissions of air pollution and greenhouse gasses [2]. Moreover, Ankara is among six most air-polluted cities in Turkey.

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

1.2 Design Goals

Our aim is to have an easily accessible, modifiable, user-friendly and secure application.

1.2.1 Accessibility

Carpus should be accessible by students who use a device which has a proper version of Android. Since users who will be carpooling are going to use the app concurrently, it is essential to have a sufficient number of users using the application.

1.2.2 Modifiability

As Carpus will only be a helper to find and communicate with other people who search partners to carpool, the rest is the responsibility of the users. In order to prevent encountering unpleasant situations, Carpus may need to be modified from time to time. Also state regulations about transportation and university regulations may force Carpus to be modified. Therefore, Carpus should be easily modifiable. The system and components should be compact, documentation should be convenient and updates should not limit basic functionalities of the application and should be suitable with other design goals.

1.2.3. User Friendliness

Carpus aims to provide its users a basic user interface and schedule carpools easily with other users in a short amount of time. Therefore, the application must provide a comfortable experience to users and user interface design should avoid confusing concepts. Users should be directed to proper tabs when they are using Carpus.

1.2.4 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.

1.3 Definitions, Acronyms, and Abbreviations

App: Abbreviated version for the word “application”. In the context of this report, an app is a smartphone software that is designed for users.

App Store: Stands for the market where applications can be purchased for free or for charge by users via their smartphones. As our application will be an Android application, we would mean Google Play Store most of the time.

Carpus: The name of the application. It is a concatenated version of words “Car” and “Campus”.

Carpool: An arrangement between people to make a journey in a single vehicle.

Client: A single user of the application.

Server: Back-end part of the application where related information is stored.

API: Application Programming Interface

GPS: Global Positioning System

Route: The chosen way taken from a starting point to a destination.

Trip: A journey to a place, which is from a starting point to the campus or vice versa in our case.

Passenger: User of the Carpus who gets in any other user's car to travel.

Driver: User of the Carpus who drives his own car to carpool.

1.4 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.

This report is the high-level design report for Carpus which provides detailed information about high-level design of the Carpus and related models, diagrams and architectures. It also contains how various factors in engineering design are considered and teamwork details as well.

2. Current Software Architecture

Today, many carpooling applications have been in the software market which are Carma Carpooling, Sidecar, Uber , Rideout etc. [3] Those of these applications can give an opportunity to make a profit to drivers, but some of these don't give this opportunity to drivers. Common aim of the applications is to reduce usage of vehicles for transportation.These applications are mobile applications and service on the web.There is no specific security software architecture to control users who use these applications. These tools use map APIs to indicate the users and drivers , which are Google Map API , Yahoo Map API and use cloud computing software architecture to store data.To sum up, there is not a carpooling tool for university students. Therefore, Carpus is an innovative mobile application for the students in order to reduce usage of vehicles. Carpus can be implemented with these similar current software architectures.

3. Proposed Software Architecture

3.1 Overview

We show the main parts of the system in the system decomposition. 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.

3.2 Subsystem Decomposition

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

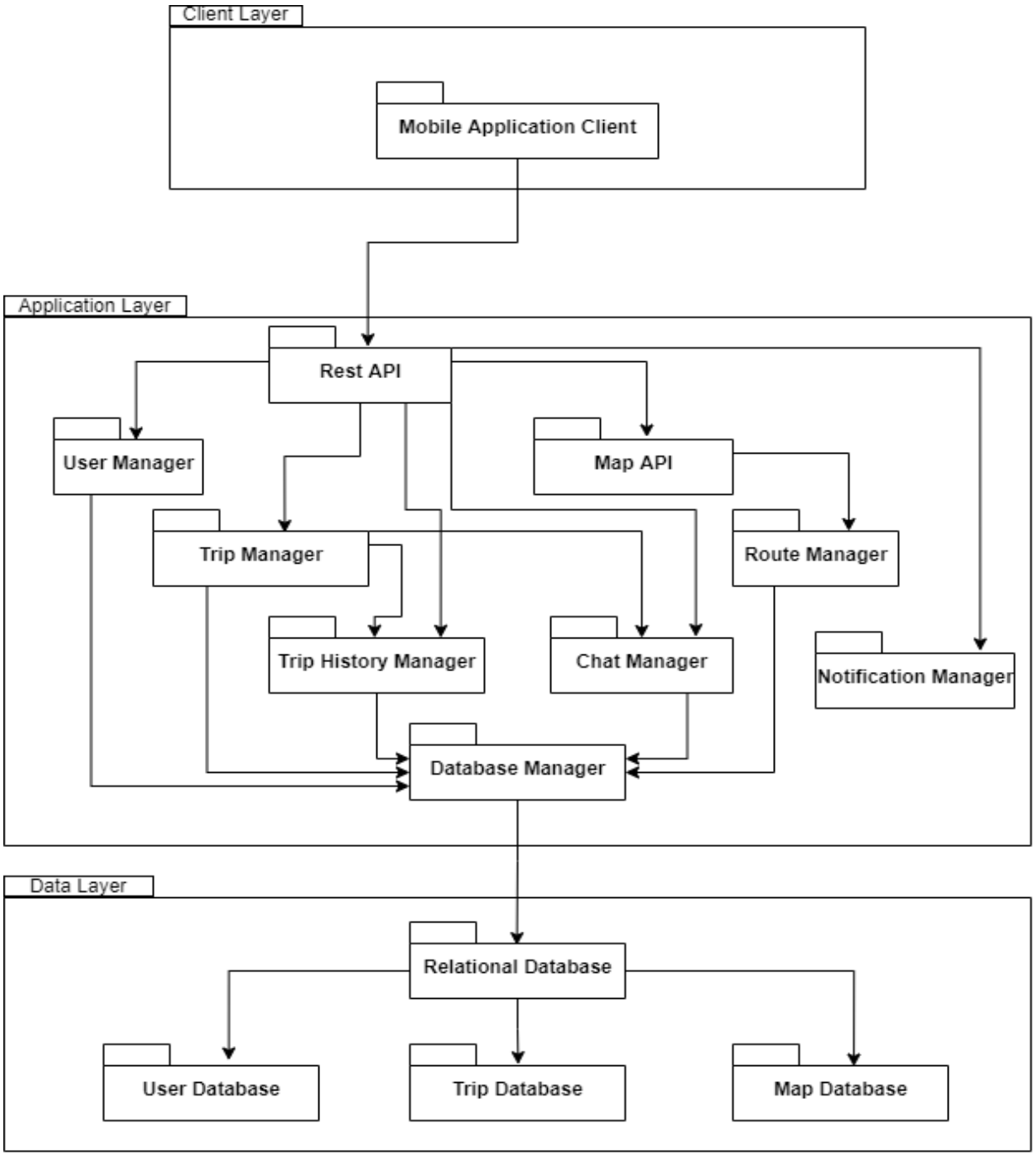

Figure 1: Subsystem Decomposition

3.3 Hardware/Software Mapping

We have a mapping diagram that can be seen in Figure 2. Carpus provides the service with HTTP requests from the server to show the app to users. Mobile application client for users to see what is happening in the app. The application uses REST API to retrieve the information from the database controller. Database is responsible for the storage and presentation of the data of the users and trips.

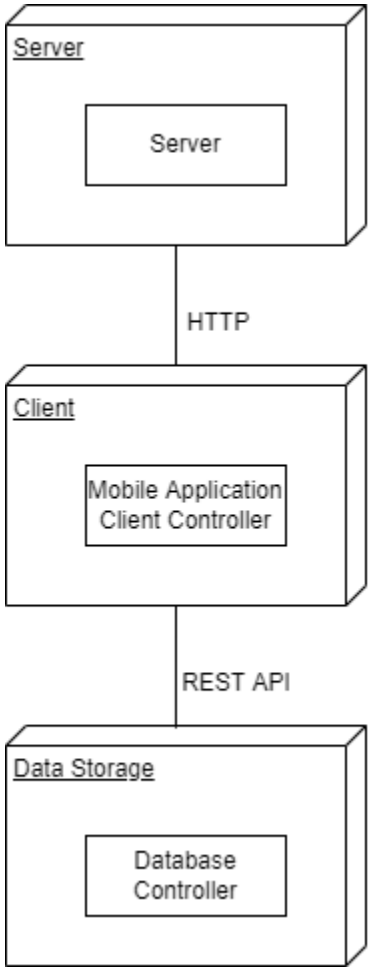

Figure 2: Hardware/Software Mapping

3.4 Persistent Data Management

We need to store all of the users and trips data so that we can reach them to keep the application up to date. We may need to store huge data according to the number of trips and number of users as we expect the number of users of the application to increase day by day. Therefore, the data will be stored with a text-based and structured format to make them lightweight and fasten to reach.

We need to store userID, userName, userSurname, userPlateNumber, userAddress, userEmail and password for clients. We need to store tripID, route, group, time, direction, ratings and comments for trips.

We will use the database to store and retrieve data which is mentioned above.

3.5 Access Control and Security

Carpus will store a lot of information about the users which are student identification number, their national identification number, name etc.Therefore, Carpus must be securable application and user's information must be kept safely. 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 used 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 will 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.The users can evaluate their past trips, continue doing carpooling with their carpooling group.For example of usage of the Carpus carpooling; a user who study at Bilkent University , if the user has a vehicle , the user can start a trip to go the university or other place 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 and estimated travel time. After the trip, the group of carpooling can continue other trips or the group can be closed. Users can evaluate the trip and the owner of the vehicle which is used for carpooling can be awarded.

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 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. Subsystem Services

4.1 Client Layer

4.1.1 User Interface

4.1.1.1 Register Page

This page is shown only for the first time to the user when the application is first started (if the user has logged in before, the main page will be loaded automatically). The functionality of the page basically registers the user to the carpus system.

4.1.1.2 Main Page

The functionality of the page is to direct the user to the specific page. It will contain buttons to direct pages such as ‘Create Trip’, ‘Join Trip’, ‘Profile’, ‘My Groups’, etc.

4.1.1.3 Create Trip Page

This page is for car owners only. A driver can create a trip for an arrival time. The selected time (e.g. 9:30 A.M.) will be used by algorithm to schedule the route.

4.1.1.4 Join Trip Page

This page gets a time input from a user, then it lists the available trips at that time.

4.1.1.5 Profile Page

Users can change their information by this page such as name, car plate, etc.

4.2 Application Layer

4.2.1 Rest API

Rest API will be used for intercommunication between services.

4.2.2 User Manager

User manager is responsible for users’ interactions with other services.

4.2.3 Trip Manager

Trip manager is responsible for arranging trips with relevant users.

4.2.4 Trip History Manager

This service manages past trips and allows users to review these trips.

4.2.5 Map API

Map API will be used to show specific map views to individual users and update the map accordingly.

4.2.6 Route Manager

Route manager is responsible for finding the optimal route for a trip.

4.2.7 Chat Manager

Chat manager is responsible for arranging the group chats and individual chats between users.

4.2.8 Notification Manager

This service is responsible for notifying the users about trips, reviews, chat messages and other necessary information.

4.2.9 Database Manager

Database manager establishes the relation between data layer and app layer.

4.3 Data Layer

4.3.1 Relation Database

4.3.1.1 User Database

The database will be used for users' data (e.g. car plate, id, name).

4.3.1.2 Trip Database

The database will be used for trips' data (e.g. routes, schedules for specific days, etc.).

4.3.1.3 Map Database

The database will be used for map’s data (e.g. university locations, names etc.).

5. Consideration of Various Factors in Engineering Design

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.

5.1 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.

5.2 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.

5.3 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.

5.4 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.

5.5 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.

5.6 Social Factors

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

5.7 Cultural Factors

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

5.8 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

6. Teamwork Details

6.1 Contributing and Functioning Effectively on the Team

As the Carpus project group, we planned to meet periodically and discuss our progress, future tasks and which changes could be done regarding the project. Throughout the semester, each group member proposed his/her idea for the upcoming task and we made decisions in an equal manner. We divided each task into subtasks and equally distributed each task. We decided on deadlines which are before the actual deadline to finish subtasks and discuss possible changes and improvements. In the analysis, design and implementation stages of the project, some tasks that are connected are handled by the same person. Some relatively major tasks assigned to mini teams consist of two or three people throughout these stages. In the Carpus team, each team member has been respectful and kind to each other, contributed to every stage of the project, proposed solutions when needed and reviewed other's work at the end of each stage.

6.2 Helping Creating a Collaborative and Inclusive Environment

In order to have a collaborative and inclusive environment throughout the project development process, we frequently shared our ideas face to face or via online meetings. We established a healthy, respectful and comfortable work environment during this process. Tasks assigned to group members according to their strengths and weaknesses. For the cases a member or few members could not handle the given task, other members attempted to help and compensate for the weaknesses of their teammates. Thus, we keep up with the deadlines and comply with our schedule.

6.3 Taking Lead Role and Sharing Leadership on the Team

As mentioned before in the analysis report, we have six work packages: Analysis, High-Level Design, Implementation I, Low-Level Design, Implementation II and Final Report. We schedule our work according to these packages and due dates. Each member is assigned to all work packages and everyone leads at least one package. Thus, we can say that team leadership is shared among the team members.

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 2: 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 3: 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 4: 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 5: 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 6: Work Package 4

<table>
<tr><td colspan="4">WP5: Implementation II</td></tr>
<tr><td colspan="4">Start Date: 4th week of the spring semester End Date: 13th week of the spring semester</td></tr>
<tr><td>Leader:</td><td>İbrahim Furkan Aygar</td><td>Members Involved:</td><td>Veli Can Mert

Deniz Çalkan

Mehmet Yiğit Harlak

Murat Sevinç</td></tr>
<tr><td colspan="4">Objectives: To finalize the implementation of the project</td></tr>
<tr><td colspan="4">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.</td></tr>
</table>

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

Table 7: 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 8: Work Package 6

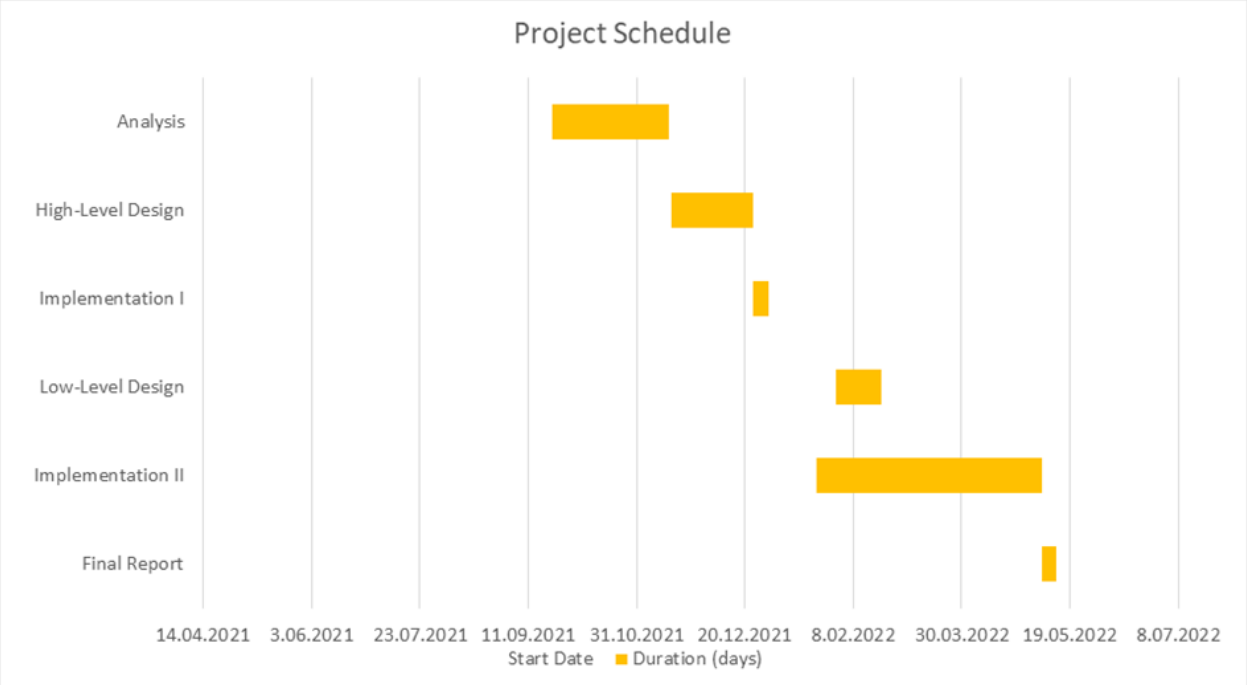

Figure 2: Gantt Chart for the Project Schedule

7. References

[1] ”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].

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

[3] “10 Rideshare apps”, [Online]. Available : https://www.techrepublic.com/article/10-rideshare-apps-to-crowdsource-your-commute/ [Accessed : 23- Dec-2021]