Bilkent University
Department of Computer Engineering

Senior Design Project

T2320

BilRide

Analysis and Requirement Report

21702587, Turgut Alp Edis, alp.edis@ug.bilkent.edu.tr
21703556, İdil Yılmaz, idil.yilmaz@ug.bilkent.edu.tr
21602059, Dilay Yiğit, dilay.yigit@ug.bilkent.edu.tr
21702331, Doğukan Ertunga Kurnaz, ertunga.kurnaz@ug.bilkent.edu.tr
21801861, Funda Tan, funda.tan@ug.bilkent.edu.tr
Prof. Dr. İbrahim Körpeoğlu
Erhan Dolak and Tağmaç Topal

Nov 13, 2022
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.

Contents

1. Introduction

The primary purpose of BilRide is to create a mobile app to solve the issues caused by the lack of ring shuttles at Bilkent University and the lack of hitchhiking culture. It also aims to transform the transportation environment of Bilkent and its individuals, and it is willing to create an opportunity for people who are cautious about hitchhiking, enabling people to socialize with others. In the meantime, the driver will not suffer from high gas prices, and the passengers will be able to arrive at the campus on time and in a much more comfortable way. Also, the carbon emissions due to private car usage will decrease.

This project analysis report will provide a detailed analysis of the application. We will start by giving information about existing or similar systems and their weak points. Then, we will give a brief overview of BilRide and its functionalities. After that, we will examine the Functional Requirements from both the Application's and Server's perspective. Also, we will explain our non-functional requirements in detail. Lastly, pseudo requirements will be listed so that everyone involved in the development of the BilRide Application will have the same idea about the scope of the application and the technologies we have to use. After that, we will show the system models and diagrams we draw for our application. Then, possible scenarios will be discussed to cover all user stories from the design phase. Lastly, screen

mock-ups and navigational paths will be provided. Some other analysis elements will also be covered in this report.

2. Current System

There are some alternatives to our application, for example, BlaBlaCar. Even though they are already in the market, we believe that they have some weaknesses in terms of functionality and usability. We want to focus on these points more. We will bring a new option to the pickup choice of the passenger. BlaBlaCar offers only pickup points to the users[1, 2]. However, in our application, the passenger can also be picked up from the current location and the pickup points. Besides, we want to bring to the market a sustaining innovation by adding unique features. These are checking if the user is a Bilkent University member to provide more safety, providing both carpooling options and ring bus ETA's for a complete solution to our users, and creating an environment for socializing with exchanging information about their ride, Spotify playlists, and gamification the experience with a badge system.

3. Proposed System

3.1. Overview

BilRide app has additional features compared to its competitors, such as the passengers can rate the driver, give feedback, and view the drivers' upcoming

routes schedule if they set it. Also, they can filter for their preferences; for example, they can choose to filter only rides with fewer people in one car. Drivers can view which route they will take, how many passengers they want, whether there will be a tip (a certain percentage of the gasoline price), and the passengers' information, such as their profile photos, names, and pickup points. Besides, there will be a reward system for drivers and passengers who can earn cool badges or points for each completed quest. They can show off their badges on their profile or use the points to get a higher rank among the other participants. Even the passengers and drivers can form small teams and use them to exchange information related to their travel, or they can choose or create a Spotify playlist that will be played during the journey.

In addition to carpooling, we wanted to touch on the ring problem. The excessive occupancy of the rings due to the increase in the student population in the number of rings in Bilkent and the delay caused by the harsh weather conditions and traffic make it difficult for the students to use the rings. With the ring tracking system, we can determine which station the rings are at, the data we will collect, the ETA times between the stops (using ML), the number of people in the rings, and the occupancy rate. Thus, when the passenger who wants to do carpooling cannot find a car and/or misses it, s/he will be able to choose to get on the ring, thanks to the BilRide app.

Customer engagement innovation will be used to enhance the BilRide application. The passengers can improve the application via feedback about the timing of the shuttles, which creates more precise data for the ETA of shuttles, about the behavior of the driver, which creates a more secure application for Bilkent students, and about the application itself, which leads to the changes that make the application better. In other words, we will use the customer data to improvise our application.

3.2. Functional Requirements

In this report section, we will discuss the project's functional requirements in detail. This part can be divided into two sides: functionalities of the application and the server. Since there are two user types, application functionalities can also be divided into the driver and passenger sides.

3.2.1 Functionalities of the Application

3.2.1.1 All Users Functionalities

- The application should provide users with the logging-in and/or signing-up options.
- The application should provide users with a map view, including some information, such as routes, and the current location of the cars, passengers, and shuttles.

- The application should provide a customizable user profile page for all users.
- The application should show the progress for each badge to each user.
- The application should show the earned badges in the user profiles.
- The application should show users how much gasoline they save after each ride and how much gasoline they save in total on the profile page.
- The application should allow all users to create a team with other users to create a common Spotify playlist and chat with each other before the ride.
- The application should allow all users to join open or private teams with an invitation.

3.2.1.2 Passenger Functionalities

- The application should show nearby routes of the drivers to the user from the map.
- The application should show the route, available quota, gender of the car's other passengers (if it is a problem for the user), whether the gasoline price is needed, the driver's rating, picture, and name for each nearby car to the user.
- The application should allow passengers to filter the routes according to their gender preferences, the gas price and the total number of passengers in the car, and the driver's rating.
- The application should allow passengers to choose the gender of the other passengers (if they need to) they wish to travel with.

- The application should allow passengers to send requests to nearby drivers with a selection of pickup points or their current locations.
- The application should allow passengers to rate the driver and give feedback about the journey after the ride.
- The application should show the user's current approximate location of the shuttle.
- The application should show the occupancy rate of the upcoming shuttle.

3.2.1.3 Driver Functionalities

- The application should show the current requests from the passengers to the driver with selected options, such as their pickup points, the gender they prefer to travel with, and whether they pay some portion of the gasoline price.
- The application should suggest the driver's route with the least gas consumption.
- The application should allow drivers to choose the number of people they want to travel with and the route, then inform the system about the choices.
- The application should allow drivers to see the passengers' Covid-19 status.

3.2.2 Functionalities of the Server

- The system should keep information about all drivers, such as name, Bilkent ID number, rating, and routes informed to the app.
- The system should keep information about all passengers, such as name, Bilkent ID number, and stops where they want to join the ride.
- The system should keep information about the shuttle, such as the estimated location and occupancy rate.
- The system should keep the information about the teams, such as the member information.
- The system should keep the requests for joining the team.
- The system should keep the location information about each user in case of emergency, such as accidents.

3.3. Non-functional Requirements

In this section of the report, we will discuss the non-functional requirements of the project in detail. We choose the non-functional requirements in the scope of our application’s domain.

3.3.1 Usability

Usability is an important non-functional requirement for our application because our application aims to reach everyone in the university. The application should be easy to use by design. All of the main functionalities of our application should be accessible by at most 3 touch gestures. The user interface will be designed to be friendly to people from all backgrounds, regardless of their understanding of technology. The application should not require any previous experience, and it should be easy to learn for novice users. The user should feel satisfied while using the application with intuitiveness.

3.3.2 Supportability

Supportability is another important non-functional requirement for our application because our application should be able to work as designed on different platforms. The application should be able to run on both Android (10.0 and above) and iOS (11.0 and above) operating systems. The application should be able to run smoothly with average hardware requirements. The application may require external installation of Google App Services for Chinese versions of some Android Mobile Phones.

3.3.3 Maintainability

The application should be developed with future improvements and extensions in mind. Object Oriented Software Development techniques will be used during implementation. The application’s codebase will be utilized with the version control tools like Git, which enables our team to work on multiple branches. That's how our team can continue development while the application is still usable. The deployment of our application will be automated with the help of GitHub Actions; when a new release happens, the action will automatically distribute the app package to the remote servers where our users can update their apps on the air. That is how the application will be easier to maintain.

3.3.4 Performance

The application should be loaded from a cold boot in less than thirty seconds. In-app pages should load in under three seconds. The application should send the user’s information to the application's backend servers in less than 3 seconds.

3.3.5 Security

In our application, there will be a database in which we store all the user’s personal data, such as; student id, full name, password, mail address, route information, etc. Sensitive ones, such as passwords, should be stored encrypted

with a non-reversible hash function in the application database. Other sensitive information related to the user itself will be stored and processed according to the law requirements, such as KVKK and GDPR. The backend functionality of the application should be only accessible by the application itself. The application should sanitize dangerous codes from the user-inputted fields and have a rate-limit functionality to prevent DoS attacks. The password should be at least eight characters in length. The application should automatically disable the user's account if the user tries to log in with the wrong password three times. The application should require re-login if the user has not opened the application for more than five days or the user changes the password. Since the application will not process transactions over it, there will be no further security checks other than the implementation guide of 3rd party payment provider. The application package will be compiled with some obfuscation techniques as best practices to prevent the decompilation of our application package.

3.3.6 Scalability

Since the application's implementation process was designed considering the scale-out approach, scalability would not be difficult. The only thing we may have to scale is the backend API, and the database in the future depends on the user count we will have. The application backend will not use microservice for simplicity, but we will try to Dockerize the API to enhance security and scalability.

3.4. Pseudo Requirements

- GitHub will be used throughout the project as a main Version Control platform.
- Dart programming language and Flutter UI SDK will be used in the project's frontend.
- Application should be able to run on both Android (+10.0) and iOS (+13.0) platforms.
- Python programming language and Django Rest Framework will be used for the backend side of the project.
- To provide real-time communication (chatting), WebSockets will be used while implementing the chat feature.
- To provide location services and maps, Leaflet and its dart package flutter_map will be used, an open-source library that uses OpenStreetMap.

3.5. System Models

3.5.1. Scenarios

Scenario 1	Signin
Participating Actor	All Users

Main Flow of Events	1. The actor opens the application. 2. The actor enters the required information. 3. The actor enters the verification code. 4. System checks the information. 5. The actor logs in successfully.
Entry Conditions	The actor opens the application.
Exit Conditions	The actor closes the application.

Scenario 2	Signup
Participating Actor	All Users
Main Flow of Events	1. The actor opens the application. 2. The actor clicks the signup button. 3. The actor enters the required information for signing up. 4. System checks the information. 5. The actor signs up successfully.
Entry Conditions	The actor opens the application and clicks the signup button.
Exit Conditions	The actor closes the application.

Scenario 3	Create Chat
Participating Actor	All Users
Main Flow of Events	1. The actor selects the ride. 2. The actor clicks the new chat button. 3. The actor writes the message. 4. The actor sends the message.
Entry Conditions	The actor selects the ride and clicks the new chat button.
Exit Conditions	The actor clicks the back button or closes the application.

Scenario 4	Create Team
Participating Actor	All Users
Main Flow of Events	1. The actor goes to the profile page. 2. The actor clicks the new team button in the team section. 3. The actor writes the required information for creating the team. 4. The actor clicks the create button. 5. The actor creates the team successfully.
Entry Conditions	The actor goes to the profile page and clicks the new team button in the team section.

Exit Conditions	The actor clicks the back button or closes the application.

Scenario 5	Reply Chat
Participating Actor	All Users
Main Flow of Events	1. The actor goes to the chat page. 2. The actor selects the chat. 3. The actor writes the reply message. 4. The actor clicks the send button.
Entry Conditions	The actor goes to the chat page and selects the chat.
Exit Conditions	The actor clicks the back button or closes the application.

Scenario 6	Delete Team
Participating Actor	All Users and Admin

Main Flow of Events	1. The actor goes to the profile page. 2. The actor clicks the delete button of the team in the team section. 3. System shows a confirmation message. 4. The actor confirms the message.
Entry Conditions	The actor goes to the profile page and clicks the delete button in the team section.
Exit Conditions	The actor clicks the back button.

Scenario 7	Edit Profile
Participating Actor	All Users
Main Flow of Events	1. The actor goes to the profile page. 2. The actor selects the edit option for the attribute. 3. The actor writes the new attribute. 4. The actor clicks the save button.
Entry Conditions	The actor goes to the profile page and selects the edit option.
Exit Conditions	The actor clicks the back button or clicks the discard button or closes the application.

Scenario 8	Join Team

Participating Actor	All Users
Main Flow of Events	1. The actor goes to the profile page. 2. The actor clicks the invitations button in the team section. 3. The actor clicks the approve button. 4. The actor clicks the approve button.
Entry Conditions	The actor goes to the profile page and clicks the invitations button in the team section.
Exit Conditions	The actor clicks the back button or closes the application.

Scenario 9	Delete Chat
Participating Actor	All Users
Main Flow of Events	1. The actor goes to the chat page. 2. The actor selects the chat. 3. The actor selects the messages that will be deleted. 4. The actor clicks the delete button. 5. System shows a confirmation message. 6. The actor confirms the message.
Entry Conditions	The actor goes to the chat page and selects the chat.
Exit Conditions	The actor clicks the back button or closes the application.

Scenario 10	Check Ring Schedule
Participating Actor	All Users
Main Flow of Events	1. The actor goes to the ring page. 2. The actor sees the current location of buses.
Entry Conditions	The actor goes to the ring page.
Exit Conditions	The actor clicks the back button or closes the application.

Scenario 11	Read Feedback
Participating Actor	All Users
Main Flow of Events	1. The actor goes to the profile page. 2. The actor sees the feedback in the feedback section.
Entry Conditions	The actor goes to the profile page.
Exit Conditions	The actor clicks the back button or closes the application.

Scenario 12	Give Feedback
Participating Actor	All Users

Main Flow of Events	1. The actor goes to the profile page 2. The actor goes to my trips page. 3. The actor selects the ride. 4. The passenger writes feedback about and rates the driver. 5. The driver selects the passenger, writes feedback about, and rates the passenger.
Entry Conditions	The actor goes to the profile page, then goes to my trips page and selects the ride.
Exit Conditions	The actor clicks the back button or closes the application.

Scenario 13	See Achievements
Participating Actor	All Users
Main Flow of Events	1. The actor goes to the profile page. 2. The actor sees the achievements in the achievement section.
Entry Conditions	The actor goes to the profile page.
Exit Conditions	The actor clicks the back button or closes the application.

Scenario 14	Delete User

Participating Actor	Admin
Main Flow of Events	1. The actor signs in. 2. The actor goes to the managed user page. 3. The actor selects the users and clicks the delete button. 4. System shows a confirmation message. 5. The actor confirms the message.
Entry Conditions	The actor signs in and goes to the managed user page.
Exit Conditions	The actor clicks the back button or closes the application.

Scenario 15	Cancel Ride
Participating Actor	All Passengers
Main Flow of Events	1. The actor goes to the profile page 2. The actor goes to my trips page. 3. The actor clicks the delete button of the rides in the upcoming rides sections. 4. System shows a confirmation message. 5. The actor confirms the message.
Entry Conditions	The actor goes to the profile page and goes to my trips page.

Exit Conditions	The actor clicks the back button or closes the application.

Scenario 16	Pay
Participating Actor	All Passengers
Main Flow of Events	1. The actor selects the ride. 2. The actor sees the tip amount. 3. The actor pays the tip after the ride.
Entry Conditions	The actor selects the ride and sees the tip amount.
Exit Conditions	The actor clicks the back button or closes the application.

Scenario 17	Create Ride Request
Participating Actor	All Passengers
Main Flow of Events	1. The actor selects the ride. 2. The actor shows the current information about the ride. 3. The actor sends the request to join.
Entry Conditions	The actor selects the ride.
Exit Conditions	The actor clicks the back button or closes the application.

Scenario 18	Delete Ride
Participating Actor	All Drivers
Main Flow of Events	1. The actor goes to the profile page. 2. The actor goes to my trips page. 3. The actor clicks the delete button of the rides in the upcoming rides sections. 4. System shows a confirmation message. 5. The actor confirms the message.
Entry Conditions	The actor goes to the profile page and goes to my trips page.
Exit Conditions	The actor clicks the back button or closes the application.

Scenario 19	Create Ride
Participating Actor	All Drivers
Main Flow of Events	1. The actor goes to the profile page. 2. The actor goes to my trips page. 3. The actor clicks the new ride button 4. The actor enters the required information for creating the ride. 5. The actor creates the ride.

Entry Conditions	The actor goes to the profile page, then goes to my trips page and clicks the new ride button.
Exit Conditions	The actor clicks the back button or closes the application.

Scenario 20	Approve Ride Request
Participating Actor	All Drivers
Main Flow of Events	1. The actor goes to the profile page. 2. The actor goes to my trips page. 3. The actor selects the ride. 4. The actor sees the requests. 5. The actor selects the requests that will be approved. 6. The actor approves the request.
Entry Conditions	The actor goes to the profile page, then goes to my trips page and selects the ride.
Exit Conditions	The actor clicks the back button or closes the application.

Scenario 21	Delete Ride Request
Participating Actor	All Drivers
Main Flow of Events	1. The actor goes to the profile page 2. The actor goes to my trips page. 3. The actor selects the ride. 4. The actor sees the requests. 5. The actor selects the requests that will be deleted. 6. The actor deletes the requests.
Entry Conditions	The actor goes to the profile page, then goes to my trips page and selects the ride.
Exit Conditions	The actor clicks the back button or closes the application.

Scenario 22	Verify Driver Profile
Participating Actor	All Drivers
Main Flow of Events	1. The actor goes to the profile page. 2. The actor clicks the verify button. 3. The actor uploaded the photo with driver 's license.
Entry Conditions	The actor goes to the profile page and clicks verify.
Exit Conditions	The actor clicks the back button or closes the application.

Scenario 23	Confirm Payment
Participating Actor	All Drivers
Main Flow of Events	1. The actor goes to the profile page. 2. The actor goes to my trips page. 3. The actor selects the ride from past rides. 4. The actor clicks the confirm payment button.
Entry Conditions	The actor goes to the profile page, then goes to my trips page and selects the ride.
Exit Conditions	The actor clicks the back button or closes the application.

Scenario 24	Add Song
Participating Actor	All Team Members
Main Flow of Events	1. The actor goes to the profile page. 2. The actor selects the team in the team section. 3. The actor clicks the add song button. 4. The actor copies and pastes the song link.
Entry Conditions	The actor goes to the profile page and selects the team.
Exit Conditions	The actor clicks the back button or closes the application.

Scenario 25	Exit Team
Participating Actor	All Team Members
Main Flow of Events	1. The actor goes to the profile page. 2. The actor selects the team in the team section. 3. The actor clicks the leave button.
Entry Conditions	The actor goes to the profile page and select the team.
Exit Conditions	The actor clicks back button or closes the application.

Scenario 26	Upload Photo

Participating Actor	All Team Members
Main Flow of Events	1. The actor goes to the profile page. 2. The actor selects the team. 3. The actor clicks the upload photo button. 4. The actor uploads the photo.
Entry Conditions	The actor goes to the profile page and selects the team.
Exit Conditions	The actor clicks the back button or closes the application.

Scenario 27	Crate Playlist
Participating Actor	All Team Members
Main Flow of Events	1. The actor goes to the profile page. 2. The actor selects the team. 3. The actor clicks the new playlist button. 4. The actor copies and pastes the playlist link.
Entry Conditions	The actor goes to the profile page and selects the team.
Exit Conditions	The actor clicks the back button or closes the application.

3.5.2. Use-Case Model

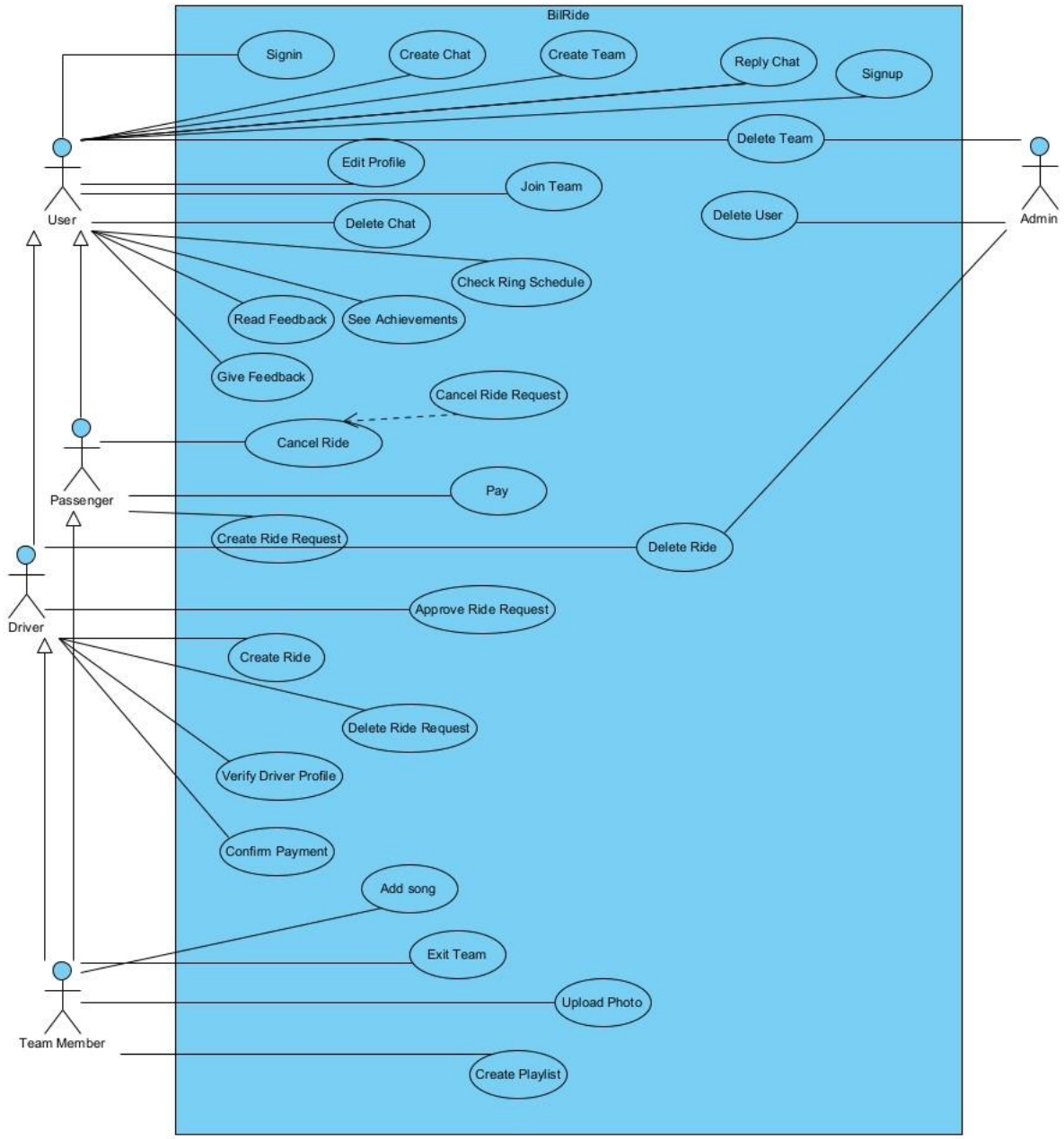

Figure 1. Use Case Diagram of BilRide

3.5.3. Object and Class Model

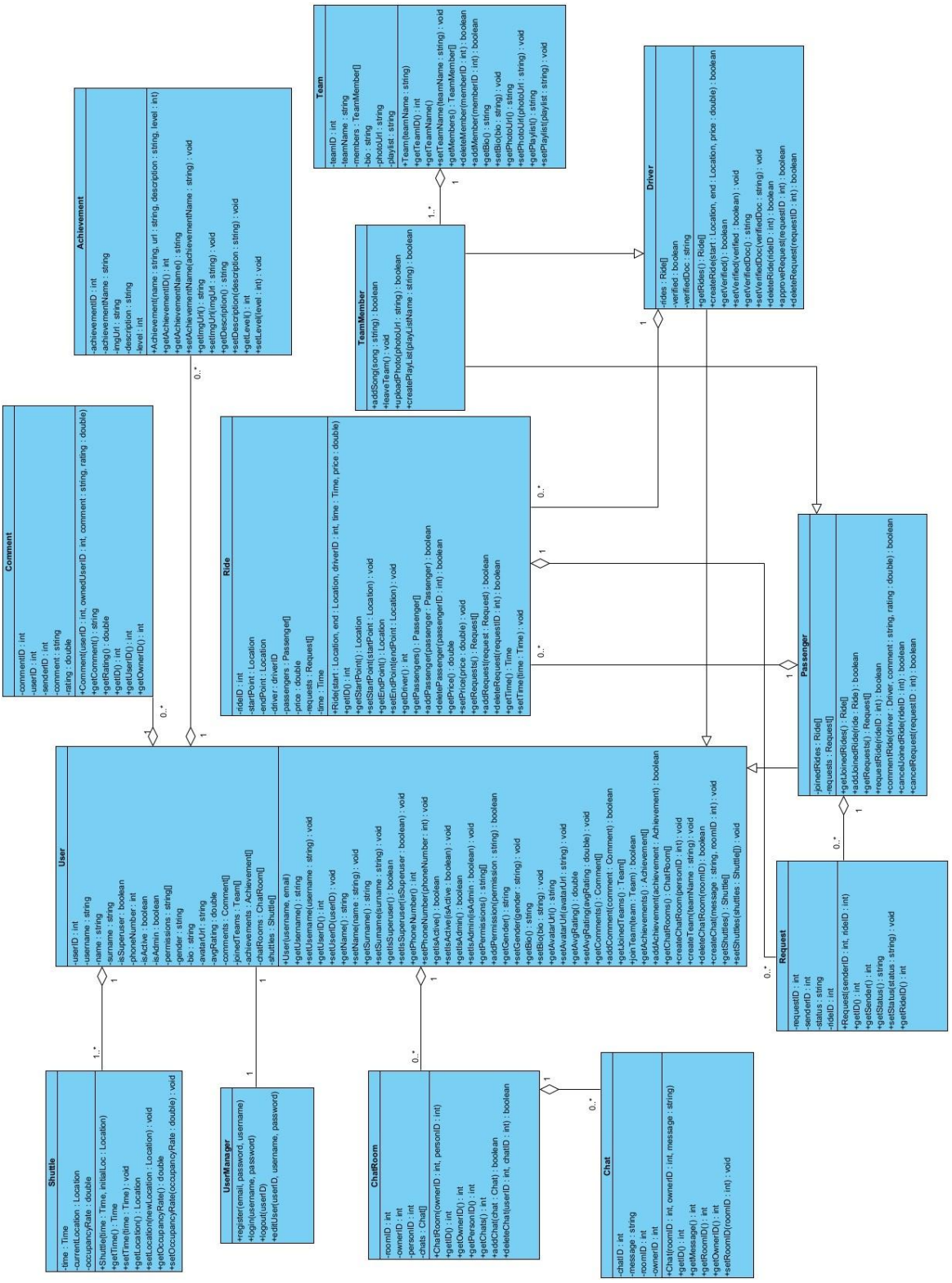

Figure 2. Object and Class Diagram of BilRide

3.5.4. Dynamic Models

3.5.4.1. Activity Diagram

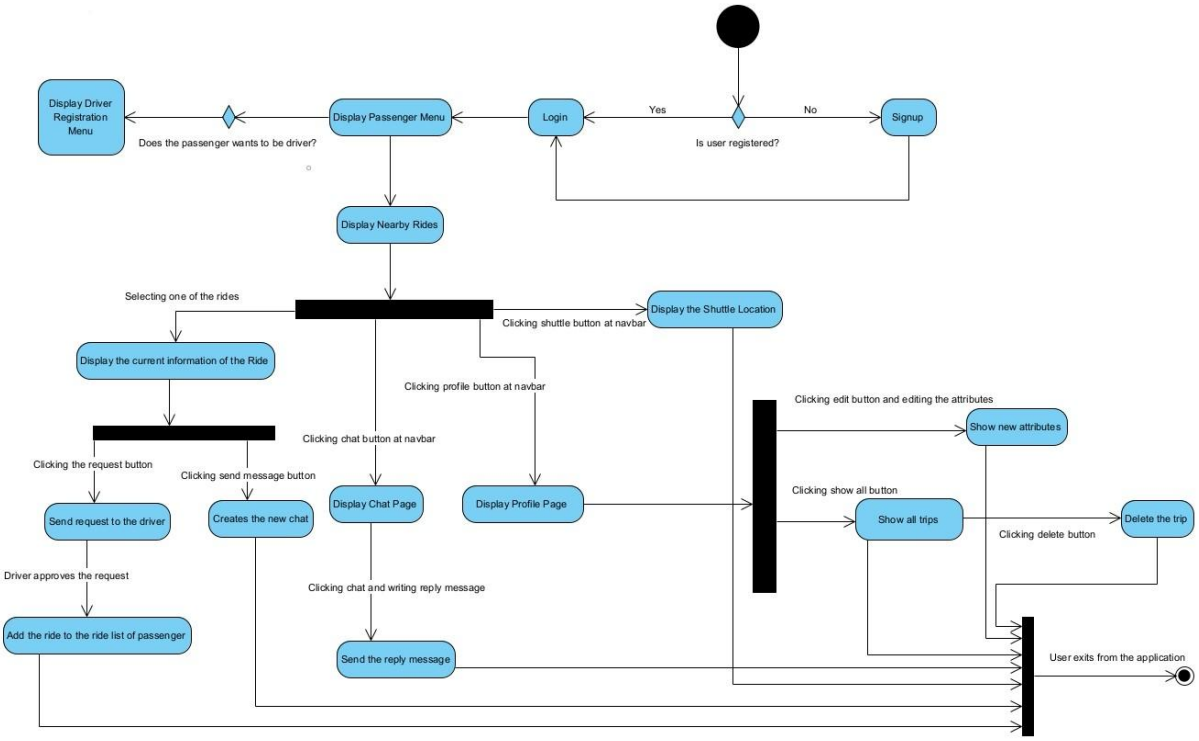

Figure 3. Activity Diagram of Main Passenger Activities

3.5.4.2. State Diagram

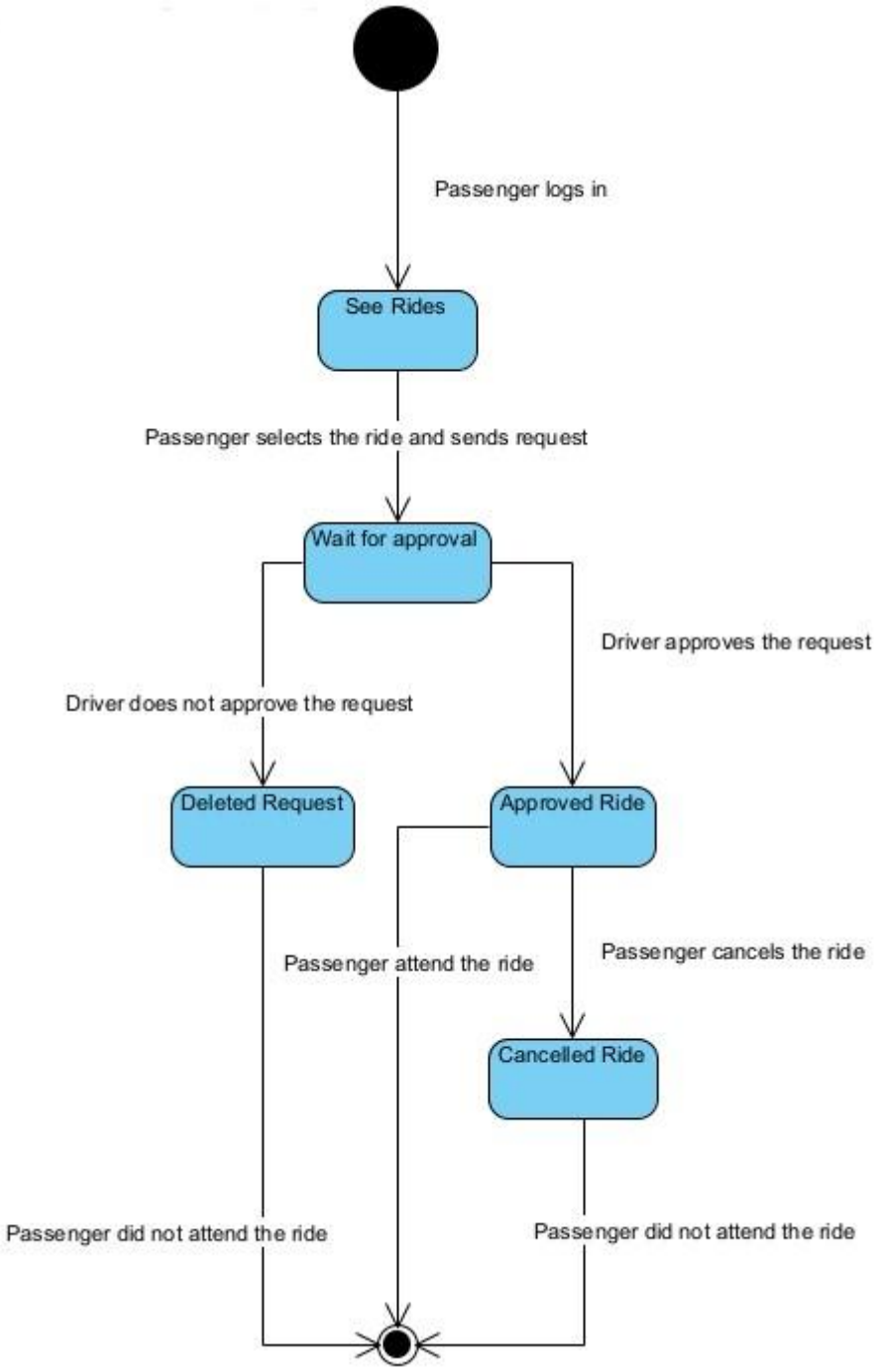

Figure 4. State Diagram of Passenger requesting to join a ride

3.5.4.3. Sequence Diagram

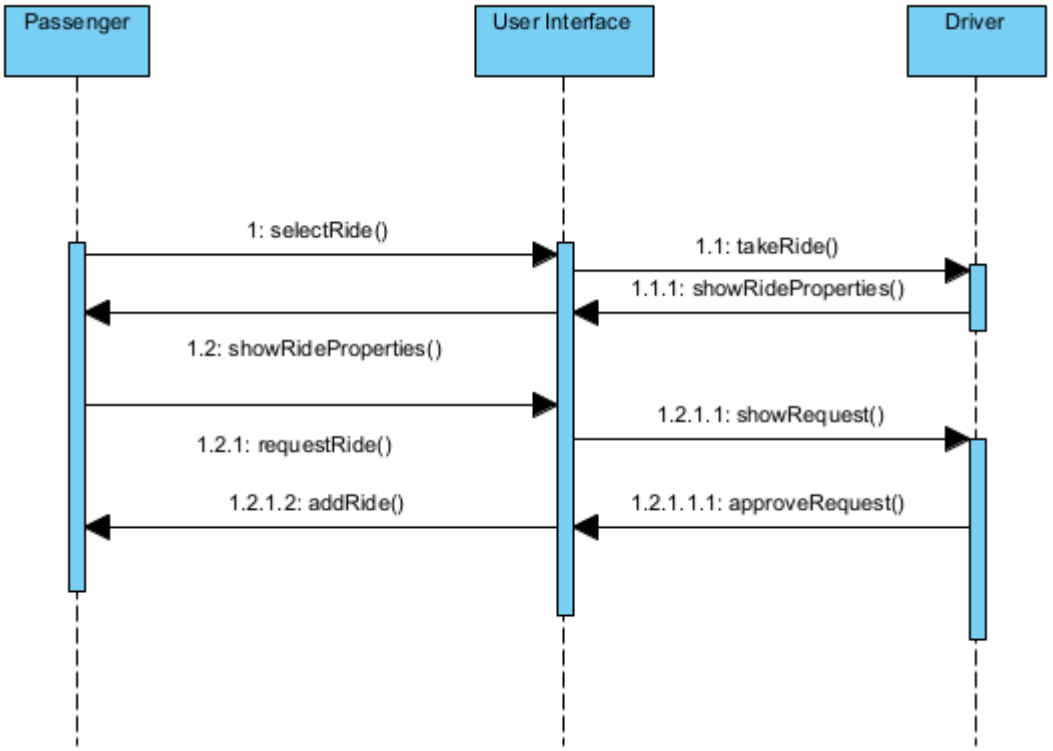

Figure 5. Sequence Diagram of Passenger Requesting to Join a Ride

3.5.5.User Interface-Navigational Paths and Screen Mockups

3.5.5.1. User Interface

3.5.5.1.1. Login Page

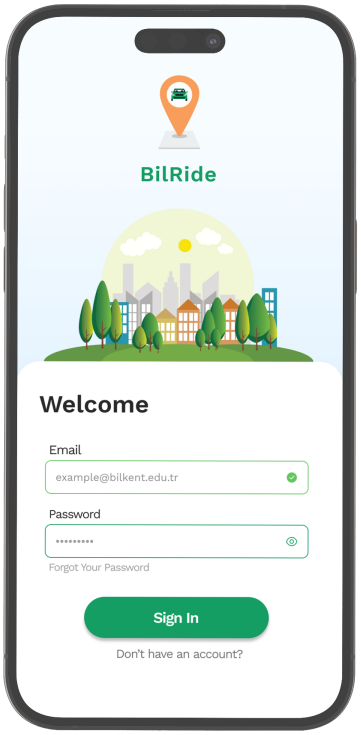

Figure 6. Login Page

In the login page, the users can authenticate via email and password. If the user forgets the password, he/she can click forget your password to go to the reset password page. Also, If the user does not have an account, he/she can easily create an account by clicking “Don’t have an account?” button. After the user types the correct credentials, the user navigates to the verification page.

3.5.5.1.2. Verification Page

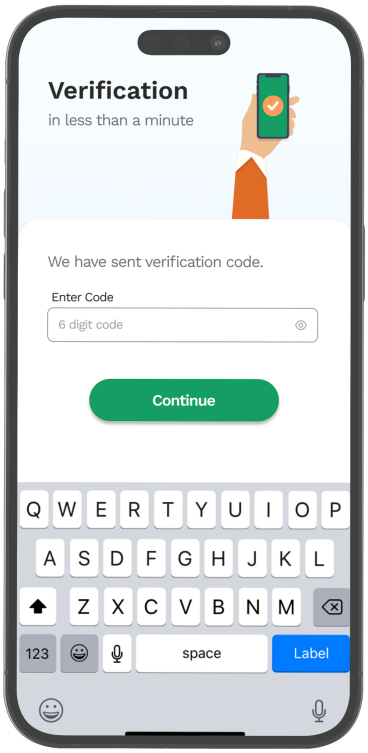

Figure 7. Verification Page

In the verification page, the user types the verification code provided to his/her email. After typing the correct code, the user navigates to the home page.

3.5.5.1.3. Signup Page

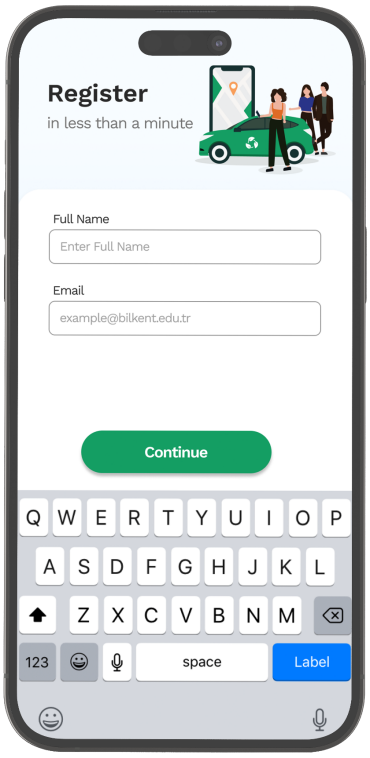

Figure 8. The Signup Page

In the signup page, the user can register to the app by just providing a full name and email. After the user clicks to the continue button, the system sends an email to the user that contains a link to the create a password page.

3.5.5.1.4. Home Page

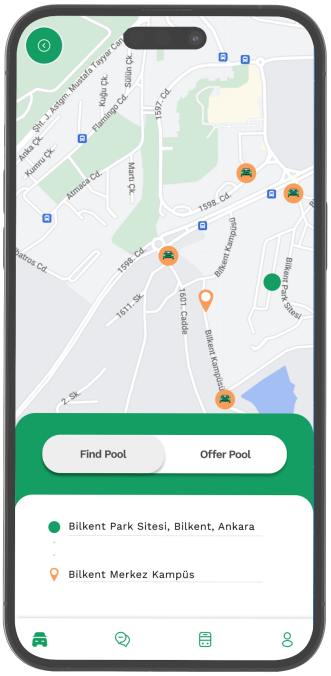

Figure 9. The Home Page with Available Rides

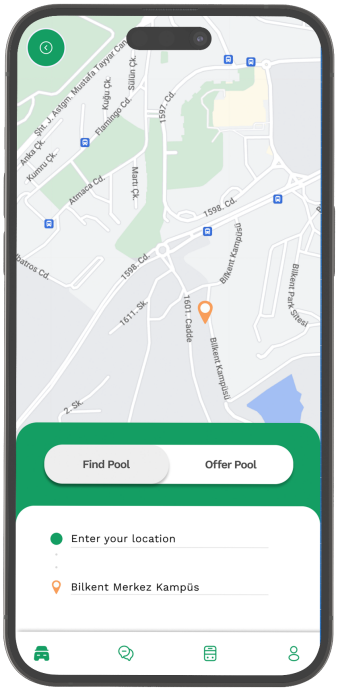

Figure 10. The Home Page with no Available Rides

In the home page, the available rides will show to the user, as in Figure 9. The user can select one of the rides and the detailed information about the ride will show to the user, as in Figure 11. From the detailed ride information, the user can send a message to the driver by clicking the send message button. After the send message button is pressed, the user will be navigated to the chat page, as shown in Figure 17. In addition, the user can send a request for joining the ride. Also, the user can click the chat button to go to the my chats page, as shown in Figure 16, or click profile icon to go to the profile page, as shown in Figure 14. Sometimes, there will be no available rides, as in Figure 10. If there are no available routes, the user can click the bus button at navbar to navigate to the ring page, as shown in Figure 12.

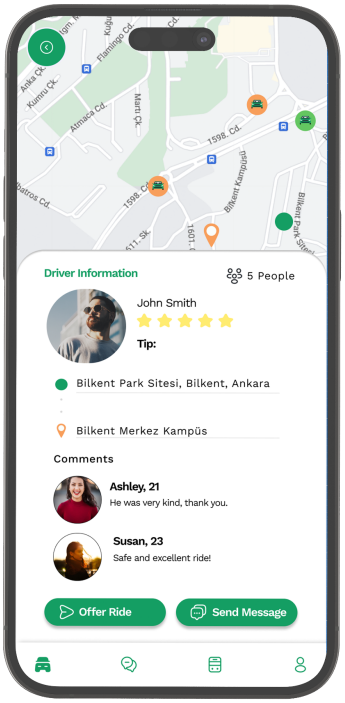

Figure 11. The Detailed Ride Information

3.5.5.1.5. Ring Page

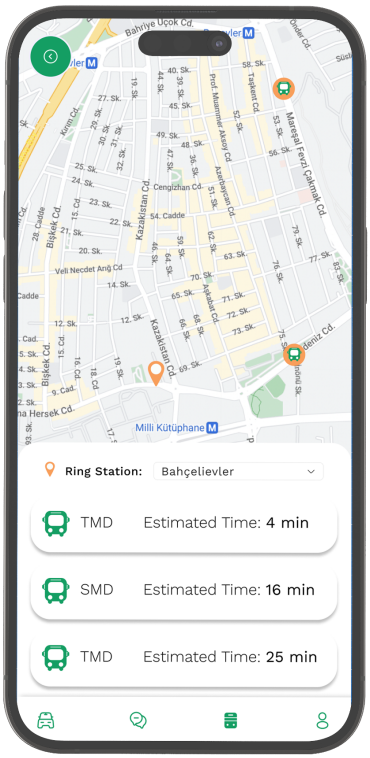

Figure 12. The Ring Page

In the ring page, the user can select the nearest bus stop to see the upcoming buses. The estimated time will be calculated with the official time schedule. Later on the project, the ETA is calculated using the most suitable ML model.

3.5.5.1.6. Offer Ride Page

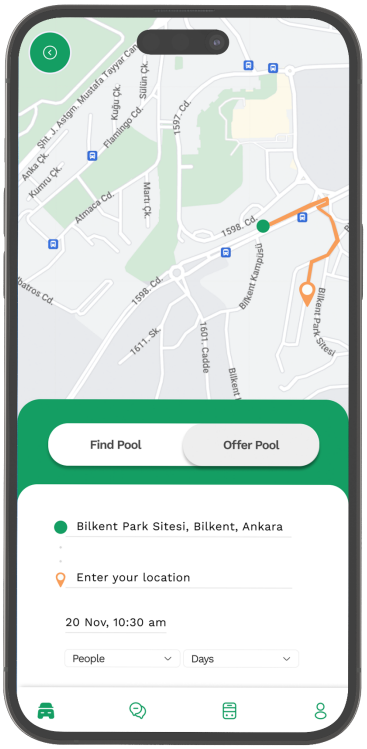

Figure 13. The Offer Ride Page

In the offer ride page, the drivers can create a ride with selecting start point, end point, maximum passengers and time. After the ride is created, every nearby passenger can see the route and request for joining the ride.

3.5.5.1.7. Profile Page

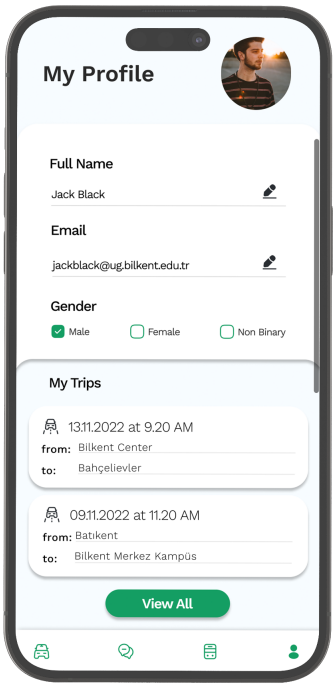

Figure 14. Profile Page

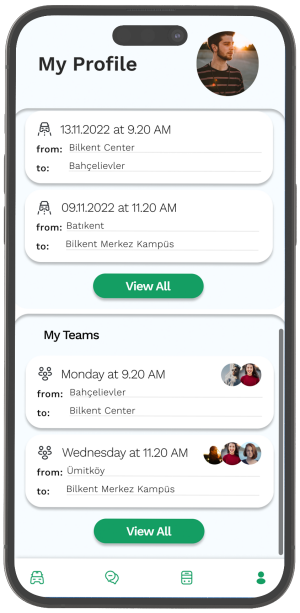

Figure 15. Scrolled Profile Page

In the profile page, the user can edit his/her information. Also, the user can see his/her trips and teams. After clicking the view all button in the my trips section, the user can see all his/her rides and can delete the ride from that page.

3.5.5.1.8. My Chats Page

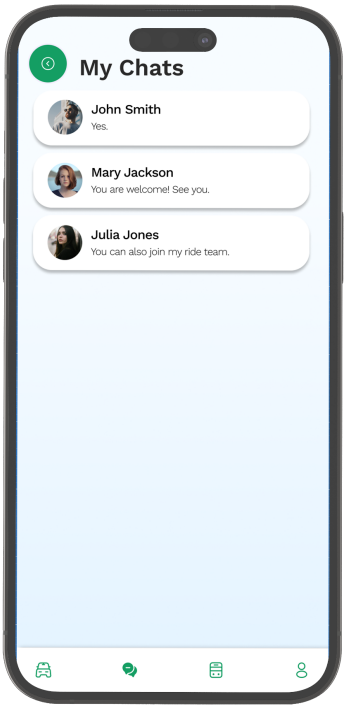

Figure 16. My Chats Page

In the my chats page, the user can see the chats with other users. After selecting one of the chats, the user can see the chat page, as shown in Figure 17. Also, if the user clicks the back button, he/she returns to the previous page.

3.5.5.1.9. Chat Page

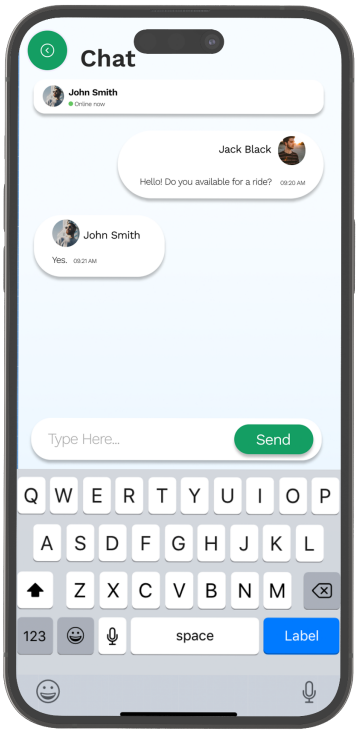

Figure 17. Chat Page

In the chat page, the user can reply to the chat or write a new message to the other user. After the user completes chatting, he/she can turn back to the previous page by clicking the back button.

3.5.5.2. Navigational Paths

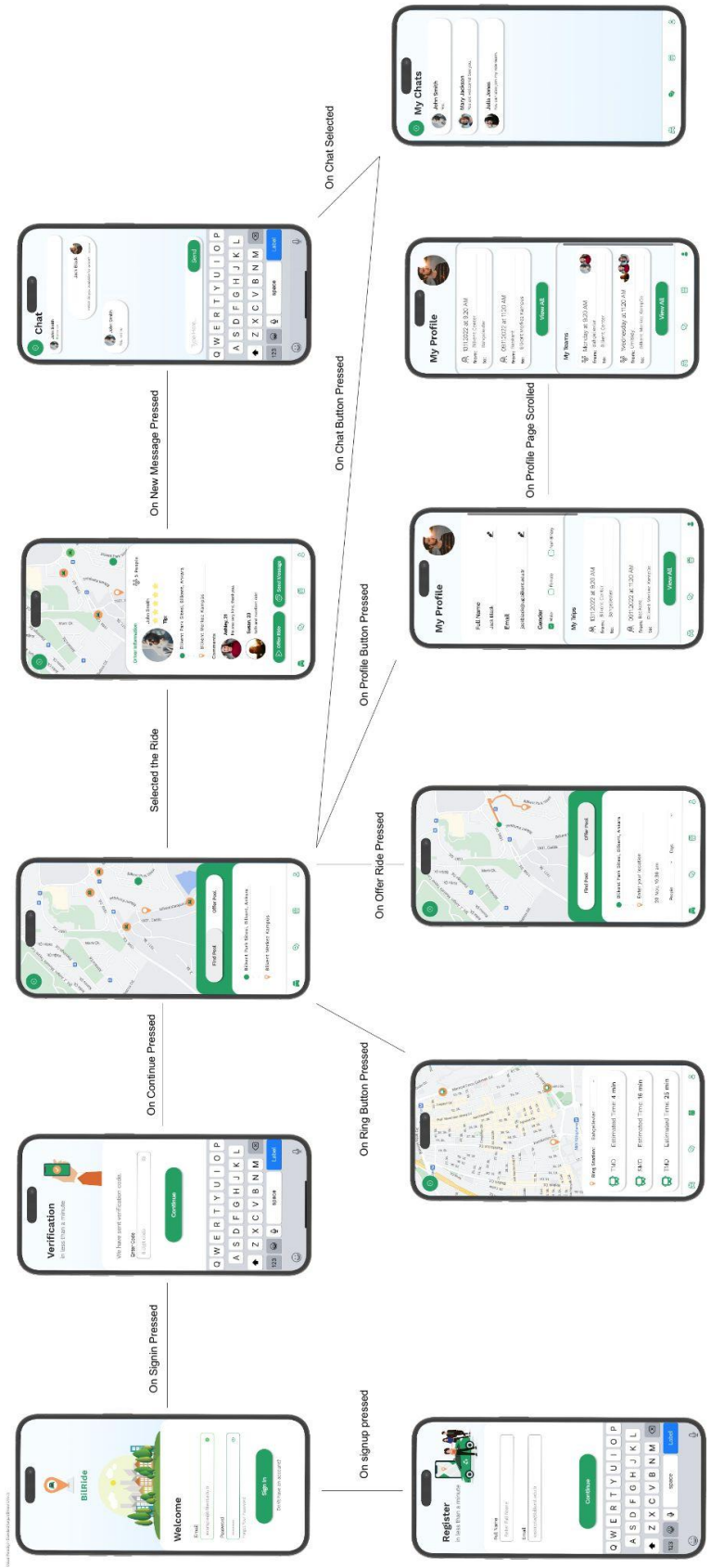

Figure 18. Navigational Path of BilRide

4. Other Analysis Elements

4.1. Consideration of Various Factors in Engineering Design

4.1.1 Environmental Factors

Our project addresses environmental factors as environmental sustainability and benefits. In our analysis, we took what is best in terms of environmental benefit into consideration.

4.1.2 Public Safety Factors

Safety is a factor that is highly important for BilRide. In our thinking and consideration phase, we highly considered protection of personal data of users which are protected by “Kişisel Verilerin Korunumu Kanunu” (KVKK). Our project must be highly secure because the user data is confidential and protected by data protection regulations. Also, we considered situations that may put our users’ safety at risk like careless drivers, users performing unacceptable behaviors when participating in carpooling and users with serious disciplinary actions. We came up with solutions to these kinds of situations when analyzing and designing our project.

4.1.3 Economical Factors

As an economic point of view, we need to address how much a ride will possibly cost and a recommended price to share between rider and passengers. Our

application will be free to use, and we provided our business model with an optional payment between riders and passengers.

4.1.4 Social Factors

We also took social factors into consideration. In analysis and requirements, we came up with some features which can simulate a social platform like sharing playlists, commenting and rating, achievements, chat rooms and forming teams.

4.1.5 Public Health Factors

Public health is also a factor that affects our thinking. Since we are affected by a pandemic, we may put more importance on public health if there are situations which puts users at risk later considering the situation of pandemic.

4.1.6 Global Factors

Since our project focused on transportation of Bilkent University specifically, global factors are not as important when compared to other factors that are discussed.

	Effect level	Effect
Public health	8	Users should be notified when there is a concern of health.
Public safety	10	Users need to be notified and protected in any case that disrupts safety that may be caused by the driver or passenger.
Public welfare	4	Public welfare is connected with public health and public safety in terms of our topic.
Global factors	1	Our scope is Bilkent University and its individuals.

Cultural factors	1	Our users consist of only Bilkent University users but if we experience any cultural factor we have to consider, it can affect our analysis and design later on.
Social factors	9	There are many features that will boost sociality of the application as discussed above.
Economical factors	7	Payment between drivers and passengers will be optional and the amount will be decided by users.
Environmental factors	10	Main focus is to do what is best for the environment.

Table 1: Factors that can affect analysis and design.

4.2. Risks and Alternatives

We have two risks for our project. First risk is being unable to get data from the ring firm. Likelihood of this risk is high because the firm is independent from Bilkent University. Instant information on rings like arrival and departure times will not be accurate if this happens. This will affect our project at a medium rate because we can get the help of users to collect data and apply machine learning methods to get information about where the ring passed from each station. Our second risk is being unable to reach Bilkent University Database for identity check of users. Likelihood and effect of this risk is medium. If we cannot reach the database, we cannot check if the user is being a part of disciplinary action or if there are previous complaints about the user. However, we can check the identity of the user asking their Bilkent University email address to confirm that they are a part of Bilkent University as our B plan.

	Likelihood	Effect on the project	B Plan Summary
Unable to get data from ring firm	High	Medium	Collect ring data with the help of users and machine learning methods to calculate ETA.
Unable to reach Bilkent University System for ID Check	Medium	Medium	Checking identity of users with Bilkent university mail addresses.

Table 2: Risks

4.3. Project Plan

In order for the project to progress in a systematic way, the planned activities should be created as a list and divided into work package groups(WPs).Each of these work packages can be assigned to a leader in the form of a task, and a share of at least 2-3 students is given to the people.For each WP, the start and end dates, major milestones, objectives and deliverables are planned.

WP#	Work package title	Leader	Members involved
WP1	Project Specification Report	Doğukan	Everyone
WP2	Analysis Report	Turgut	Everyone
WP3	Backend Implementation	Doğukan	Turgut
WP4	Permission for the Data of Bilkent Members	İdil	Everyone
WP5	Frontend Implementation	Dilay	Funda, İdil
WP6	Mid Testing	Funda	Everyone
WP7	Demo Presentation	İdil	Everyone
WP8	ETA Data Collection for Services	Turgut	İdil
WP9	Training of Model	Funda	Turgut

WP10	Implementation of Shuttle Schedule	Turgut	Everyone
WP11	Detailed Design Report	Dilay	Everyone
WP12	Final Testing	Turgut	Everyone
WP13	Final Report	Funda	Everyone
WP14	Final Presentation	Doğukan	Everyone

Table 3: List of work packages

WP 1:*Project Specification Report*			
Start date: *5/10/2022* **End date:** *17/10/2022*			
Leader:	*Doğukan Ertunga Kurnaz*	**Members involved:**	*İdil Yılmaz, Turgut Alp Edis, Dilay Yiğit, Funda Tan*
Objectives: *Name and briefly describe the project. Identify initial project requirements, constraints and professional & ethical responsibilities.*			
Tasks: ***Task 1.1 Introduction and Field Research*:** *The project name, its description and requirements will be determined. Then, field research will be conducted for other projects that have similar project ideas and have been implemented.* ***Task 1.2 Constraints and Issues:** The constraints and the issues of the project will be discussed and written to the report.*			
Deliverables ***D1.1:** Project Specification Report*			
WP 2: *Analysis Report*			
Start date: *7/11/2022* **End date:** *13/11/2022*			
Leader:	*Turgut Alp Edis*	**Members involved:**	*İdil Yılmaz, Turgut Alp Edis, Dilay Yiğit, Funda Tan*
Objectives: *This report will show the detailed analysis of the project. The diagrams, tables, scenarios and other analysis elements will help the detailed analysis of the project.*			
Tasks: ***Task 2.1 Proposed System Models*:** *System models, such as scenarios, use case model, object and class model and dynamic models will be created according to the fundamental features of the project.*			

Task 2.2 System Screen Mockups* :** *Screen mockups will be sketched according to the design model which will be discussed before the report.* ***Task 2.3 Other Analysis Elements: *Other elements, such as various factors, risks, plan, proper teamwork, ethical issues and planning for new knowledge, will be discussed in detail and written to the report.* ...
Deliverables ***D2.1:*** *Analysis Report*
WP 3: *Backend Implementation*
Start date: *Mid Of November* **End date:** *Mid of December*

Leader:	*Doğukan Ertunga Kurnaz*	**Members involved:**	*Turgut Alp Edis*

Objectives: *Implementing the backend in Python. The use and processing of the data that is obtained.*
Tasks: ***Task 3.1 SDK Research for Backend* :** *Finding a platform for the backend in order to create the architecture of the system, plan the database management and ensure that the system works at maximum efficiency and speed.* ***Task 3.2 Choosing the Language to be Used for Backend* :** *Choosing an object-oriented, interpretive, modular and interactive, high-level language is important when choosing the language to use for Backend. At the same time, the desired work should be done quickly with fewer lines of code and should be rich in libraries.* ***Task 3.3 Creating E/R Diagram for Database:*** *Entity-relation diagram will be created to ease the database work.* ***Task 3.4 Applying E/R Diagram to the Code:*** *The code for managing the database using E/R diagram will be written and the fundamental part of the backend will be completed.* ***Task 3.5 Implementing Docker:*** *Docker will be used to cease the errors caused by the environment.* ***Task 3.6 Completing Backend:*** *The other pieces that are required for backend and API will be completed.*
Deliverables ***D3.1:*** *Complete API of the project* ***D3.2:*** *Entity-Relation Diagram of the project*
WP 4: Permission for the Data of Bilkent Members
Start date: *Mid of November* **End date:** *Beginning of December*

Leader:	*İdil Yılmaz*	**Members involved:**	*Turgut Alp Edis, Dilay Yiğit, Funda Tan, Doğukan Ertunga Kurnaz*
Objectives: *Data acquisition permission for the ID and driver information of people from Bilkent University.*			
Tasks: ***Task 4.1 Determining the Data to be Received* :** *It will be determined which personal data should be obtained. It will be examined which data is required for authentication.* ***Task 4.2 Requesting permission from BCC:*** *The project will be briefly explained to BCC and the permission of read data of Bilkent University students will be requested with the help of instructors and supervisor* ***Task 4.2 Testing the Received Data from BCC*:** *The obtained data are tested in the environment. Any access permission problems and bugs are fixed.*			
Deliverables ***D4.1:*** *Permission document required for Bilkent data.* ***D4.2:*** *Read-only database of Bilkent University students*			
WP 5: Frontend Implementation			
Start date: *Mid Of November* **End date:** *Mid Of December*			
Leader:	*Dilay Yiğit*	**Members involved:**	*Idil Yılmaz, Funda Tan*
Objectives: *Implementing the frontend required for the interface in the Flutter app and implementing the design of the app thanks to the UI created with the help of mock ups.*			
Tasks: ***Task 5.1 Library Research for Creating interfaces*:** *A selection of libraries that work well on both Android and iOS, providing the ability to optimize all reusable components allowing them to incorporate new parts into the development of their applications*			

Task 5.2 Implementation of User Interface*: *Implement UI with the help of screen mockups. ***Task 5.3 Merging frontend with backend:* *Merge frontend with backend using API.***			
Deliverables ***D5.1:* *Complete UI of the project***			
WP 6: Mid Testing			
Start date: *Beginning of December* **End date:** *Mid of December*			
Leader:	*Funda Tan*	**Members involved:**	*Turgut Alp Edis, Dilay Yiğit, İdil Yılmaz, Doğukan Ertunga Kurnaz*
Objectives: *Testing the current system to solve bugs if there are any.*			
Tasks: ***Task 6.1 Testing the project:* *Test the current system.*** ***Task 6.2 Debugging:* *Debug the project if there are any errors or bugs.***			
Deliverables ***D6.1:* *The project with no bugs or errors***			
WP 7: Demo Presentation			
Start date: *Mid of December* **End date:** *Mid of December*			
Leader:	*İdil Yılmaz*	**Members involved:**	*Turgut Alp Edis, Dilay Yiğit, Funda Tan, Doğukan Ertunga Kurnaz*
Objectives: *Presentation of the current system and discussion about the project*			
Tasks: ***Task 7.1 Preparing presentation:* *Prepare a presentation of the project.*** ***Task 7.2 Presenting the demo:* *Present the current system***			
WP 8: ETA Data Collection for Buses			
Start date: *Beginning of January* **End date:** *End of January*			
Leader:	*Turgut Alp Edis*	**Members involved:**	*İdil Yılmaz*

Objectives: *Collecting the time and location data of the buses.*			
Tasks: ***Task 8.1 Requesting for tracing from bus firm* :** *Request a permission from the bus firm of Bilkent to trace buses for one month.* ***Task 8.2 Implementing backup plan:*** *Trace the buses for one month with the help of volunteer students of team members and create a dataset.*			
WP 9: Training of Model			
Start date: *Beginning of February* **End date:** *Mid of February*			
Leader:	*Funda Tan*	**Members involved:**	*Turgut Alp Edis*
Objectives: *Training the selected ML model to predict ETA of the buses.*			
Tasks: ***Task 9.1 Determining the ML model*:** *Select the best suitable ML model that can be trained with the current dataset.* ***Task 9.2 Training the model:*** *Train the ML model with the proportion of current dataset.* ***Task 9.3 Testing the model*:** *Test the model with the proportion of current dataset.*			
Deliverables ***D9.1:*** *The trained ML model*			
WP 10: Implementation of Shuttle Schedule			
Start date: *Beginning of March* **End date:** *Mid of March*			
Leader:	*Turgut Alp Edis*	**Members involved:**	*İdil Yılmaz, Dilay Yiğit, Funda Tan, Doğukan Ertunga Kurnaz*
Objectives: *The detailed shuttle schedule showing the current location of the bus with great accuracy.*			
Tasks: ***Task 10.1 Implementing the ML model to the project* :** *Implement the ml model to the project to calculate the ETA of the bus.* ***Task 10.2 Implementing Tracing Feature:*** *Implement the live bus tracing feature to improve the ML model*			

Deliverables ***D10.1:*** *The ETA and bus tracing features*			
WP 11: Detailed Design Report			
Start date: *TBD* **End date:** *TBD*			
Leader:	*Dilay Yiğit*	**Members involved:**	*Turgut Alp Edis, İdil Yılmaz, Funda Tan, Doğukan Ertunga Kurnaz*
Objectives: *The detailed design report indicates the design and implementation decisions in detail.*			
Tasks: ***Task 11.1 Creating Detailed Design:*** *Prepare the details of the design of the project.*			
Deliverables ***D11.1:*** *The detailed design report*			
WP 12: Final Testing			
Start date: *TBD* **End date:** *TBD*			
Leader:	*Turgut Alp Edis*	**Members involved:**	*İdil Yılmaz, Dilay Yiğit, Funda Tan, Doğukan Ertunga Kurnaz*
Objectives: *Test the final version of the application*			
Tasks: ***Task 12.1 Testing the project:*** *Test the final version of the project.* ***Task 12.2 Debugging:*** *Debug the problems if there will be any.*			
Deliverables ***D12.1:*** *The final version of the project with no bugs*			
WP 13: Final Report			

Start date: *TBD* **End date:** *TBD*			
Leader:	*Funda Tan*	**Members involved:**	*Turgut Alp Edis, Dilay Yiğit, İdil Yılmaz, Doğukan Ertunga Kurnaz*
Objectives: *Final version of the project.*			
Tasks: ***Task 13.1 Preparing Final Architecture*** **:** *Prepare the final architecture according to the experience throughout the year.* ***Task 13.2 Discussing the Final Version of Project:*** *Discuss the properties of the final project and write to the final project.*			
Deliverables ***D13.1:*** *The final report of the project.*			
WP 14: Final Presentation			
Start date: *TBD* **End date:** *TBD*			
Leader:	*Doğukan Ertunga Kurnaz*	**Members involved:**	*Turgut Alp Edis, Dilay Yiğit, Funda Tan, İdil Yılmaz*
Objectives: *Present the final version of the project and discuss the project.*			
Tasks: ***Task 14.1 Preparing the presentation*** **:** *Prepare a presentation of the project* ***Task 14.2 Preparing the video:*** *Prepare a video about explaining and demonstrating the final version of project*			
Deliverables ***D14.1:*** *Presentation of the project.* ***D14.2:*** *Video about the project*			

4.4. Ensuring Proper Teamwork

4.4.1 Documentation & Report Writing

To ensure teamwork while writing our project reports or our documentation, we use Google Drive. We created a folder for CS491/2 where we can put our reports, documentations, user manuals that are in Google Docs form, and some templates for them. By this way, team members can see how far we progressed for each element, and how each team member contributed. Also, we can check versions of documents from Google Docs. We can give feedback by writing comments to a particular paragraph or sentence and the team member who is responsible for that part can quickly read the feedback and make changes accordingly.

4.4.2 Git & GitHub

For developing and version control tools, we use GitHub. Because of GitHub usage, contributions of each member will be visible in the repository. By proper branching strategy, we will make our contributions easier and less conflicts. We are using GitHub Issues to divide and assign development work among team members and tracking. For tracking those, we take advantage of the GitHub Projects. GitHub Projects ensure the trackability of teamwork with providing a project table and gives the ability to filter, sort and group issues and pull requests. We have two projects in GitHub Projects as Backend and Front-end. This provides us a way to organize, track closed and open issues, meaning our overall progress in the project.

4.4.3 Meetings

We prefer face to face meetings, but when we are unable to do it face to face, we prefer Google Teams or Zoom. If we planned a development task in a meeting, we log them into GitHub issues and make assignments.

4.5. Ethics and Professional Responsibilities

During the analysis phase, we considered two potential ethical and professional issues regarding our application. First one is the privacy of users which we see as both a professional and ethical responsibility of ours. Any private user data should be safe and secure to data leakages. This confidential data that is protected by “KVKK” should be safe and secure in any case of data leakage. Also, there may be some cases in which users participate in a ride from or to their houses. To reduce any problems caused by this inevitable sharing of private information, we decided to provide a feedback and complaint system in order to detect problematic users. In addition, we decided to get information from Bilkent University database if we can to detect if any user has a complaint or disciplinary action in the university database in order to prevent any harm to our users. Our second responsibility is providing users to choose which users they want to travel with. We find it useful to show some information about users, especially ratings and complaints. Additionally, we are planning to put a gender option because users can have some preferences related to the person who they will share a ride with.

4.6. Planning for New Knowledge and Learning Strategies

While developing BilRide, we use Flutter and not all of the team members have experience on Flutter. We prepared YouTube playlists which include educational videos for Flutter. We are going to benefit from these playlists, official documentation and tutorials while learning. Experienced team members can give quick training to those who are not experienced in any case. We will apply this strategy whenever we will need to learn new technologies. For the machine learning part, again, some of the team are experienced and others are not. Experienced team members can organize tutorials for those who are not in terms of libraries, models or used techniques. If we face with a situation where no one has any experience, we can always benefit from documentation, tutorials, online courses and educational videos and walkthroughs. We are planning to get help from our supervisors, innovation experts when we need to make a quick start. Then, we can deep dive into the topic that we need to learn.

5. Glossary

6. References

[1] “Hakkımızda”. https://blog.blablacar.com.tr/about-us. [Accessed: Oct 16, 2022]

[2] “BlaBlaCar Hakkında”. https://support.blablacar.com/hc/tr/categories/360002754379-BlaBlaCar-hakk%C4%B1nda. [Accessed: Oct 16, 2022]