

T.R.
GEBZE TECHNICAL UNIVERSITY
FACULTY OF ENGINEERING
DEPARTMENT OF COMPUTER ENGINEERING

MOBILE TRAVEL TALKING PAL

ZEYNEP ÇİĞDEM PARLATAN

SUPERVISOR
ASSOC. PROF. DR. MEHMET GÖKTÜRK

GEBZE
2024

T.R.
GEBZE TECHNICAL UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT

MOBILE TRAVEL TALKING PAL

ZEYNEP ÇİĞDEM PARLATAN

SUPERVISOR
ASSOC. PROF. DR. MEHMET GÖKTÜRK

2024
GEBZE

| | |
|--|--|
|  <p>GEBZE TECHNICAL UNIVERSITY</p> | <p>GRADUATION PROJECT JURY APPROVAL FORM</p> |
|--|--|

This study has been accepted as an Undergraduate Graduation Project in the Department of Computer Engineering on 02/10/2024 by the following jury.

JURY

Member

(Supervisor) : Assoc. Prof. Dr. Mehmet GÖKTÜRK

ABSTRACT

Artificial intelligence has shown extraordinary development in recent years and has deeply affected many aspects of our daily lives. Smartphones, AI-powered applications like personal assistants have become an integral part of our modern lifestyle. This project has been prepared within the scope of the GTU CSE496 Graduation Project. The goal of the project is to offer a unique travel companion experience for vehicle drivers, transforming the car journey into an informative and entertaining conversational experience.

The project, named 'Mobile Travel Talking Pal', is designed to initiate a fluid conversation about topics relevant to the user's geographical location, weather conditions, time, and the speed of the vehicle. By using a fuzzy algorithm approach for content creation, the application is tailored to provide a unique experience specific to each user's situation. A mobile application and a backend service have been developed for the project. The necessary functions of the project have been tested and reported with the developed application and backend service.

This report provides the user with information on how the 'Mobile Travel Talking Pal' application works, the communication between the application and the backend service, and how the user experience is shaped.

ÖZET

Yapay zeka son yıllarda olağanüstü bir gelişme göstermiş ve günlük yaşamımızın birçok yönünün derinden etkilemiştir. Akıllı telefonlar, kişisel asistanlar gibi yapay zeka destekli uygulamalar modern yaşam tarzımızın ayrılmaz bir parçası haline gelmiştir. Bu proje GTU CSE496 Bitirme projesi kapsamında hazırlanmıştır. Projenin amacı, araç sürücüleri için benzersiz bir yol arkadaşı deneyimi sunmak, araba yolculuğunu bilgilendirici ve eğlendirici bir sohbet deneyimine dönüştürmektir.

'Mobile Travel Talking Pal' adı verilen proje, kullanıcının coğrafi konumu, hava durumu, zamanı ve aracın hızı dikkate alınarak, bu bilgilere uygun konular hakkında akıcı bir sohbet başlatmak üzere tasarlanmıştır. Fuzzy algoritması kullanılarak içerik oluşturma yaklaşımıyla, uygulamanın her kullanıcının durumuna özgü bir deneyim sunmasını sağlamıştır. Proje kapsamında bir mobile uygulama ve backend servisi hazırlanmıştır. Hazırlanan uygulama ve backend servisiyle, projenin gerekli işlevleri test edilmiş ve raporlanmıştır.

Bu rapor, kullanıcıya, 'Mobile Travel Talking Pal' uygulamasının nasıl çalıştığını, uygulama ile backend servisinin haberleşmesi ve kullanıcı deneyiminin nasıl şekillendiği hakkında bilgi vermektedir.

ACKNOWLEDGEMENT

I would like to thank my dear teacher Mehmet Göktürk, who helped me in the making of this graduation project, answered my questions and guided me, for his contributions.

I would also like to express my respect and love to my family and friends, who have always been by my side and have always helped me throughout my education life.

Zeynep Çiğdem Parlatan

LIST OF SYMBOLS AND ABBREVIATIONS

Symbol or

Abbreviation : Explanation

GTU : Gebze Technical University

API : Application Programming Interface

HTTP : Hyper-Text Transfer Protocol

CONTENTS

| | |
|---|------------|
| Abstract | iv |
| Özet | v |
| Acknowledgement | vi |
| List of Symbols and Abbreviations | vii |
| Contents | ix |
| List of Figures | x |
| List of Tables | xi |
| | |
| 1 INTRODUCTION | 1 |
| 1.1 Project Definition | 1 |
| 1.2 Purpose and Uses of the Project | 2 |
| 1.2.1 Usage Area | 2 |
| 1.3 Project Requirements | 2 |
| 1.4 Project Planning | 3 |
| | |
| 2 Project Design and Processes | 4 |
| 2.1 Project Schemas | 4 |
| 2.1.1 Application Flowchart | 4 |
| 2.1.2 Process Flow | 5 |
| 2.2 Software Processes | 6 |
| 2.2.1 Development Environments | 6 |
| 2.2.2 Required Libraries and API's | 7 |
| 2.2.2.1 API's | 7 |
| 2.2.2.2 Required Libraries | 10 |
| 2.2.3 Fuzzy Algorithm | 11 |
| 2.2.4 Mobil Application Interface | 12 |
| | |
| 3 WORKING SYSTEM AND EVALUATION | 17 |
| 3.1 Working System | 17 |
| 3.2 Difficulties Encountered During the Project | 19 |

| | |
|---|-----------|
| 3.3 Evaluating the Success Criteria | 19 |
| 4 CONCLUSIONS | 21 |
| Bibliography | 22 |

LIST OF FIGURES

| | | |
|------|---|----|
| 1.1 | Schematic representation of the project. | 1 |
| 1.2 | Design of the project | 3 |
| 2.1 | Application flowchart | 4 |
| 2.2 | Application process flow | 6 |
| 2.3 | Developments environments used in project | 7 |
| 2.4 | ChatGPT API request | 8 |
| 2.5 | Places API request | 8 |
| 2.6 | Places API request for restaurants | 9 |
| 2.7 | News API request | 9 |
| 2.8 | OpenWeather API request | 10 |
| 2.9 | Required libraries | 10 |
| 2.10 | Fuzzy Schema | 11 |
| 2.11 | Fuzzy Schema | 11 |
| 2.12 | SplashScreen and MainPage | 12 |
| 2.13 | Registration Page | 13 |
| 2.14 | Login Page | 13 |
| 2.15 | Home Page | 14 |
| 2.16 | User Statistic Page and Logout | 15 |
| 2.17 | Home Page with added button | 16 |
| 3.1 | chatGPT Response | 17 |
| 3.2 | chatGPT Response | 17 |
| 3.3 | chatGPT Response | 18 |
| 3.4 | chatGPT Response | 18 |
| 3.5 | chatGPT Response | 18 |
| 3.6 | chatGPT Response | 18 |

LIST OF TABLES

1. INTRODUCTION

People want to explore their surroundings, gain new information and share these experiences during travels. Nowadays, with the development of technology, getting information about the places we go or the conditions around us has become important. Likewise, artificial intelligence has gained serious importance in our lives, and artificial intelligence-supported applications (such as personal assistants) have made great progress. The 'Mobile Travel Talking Pal' project aims to combine these technological advances and people's interest in discovering new information during travel. At the end of the project, the application collected the user's data in the background, processed this data and presented content to the user. In the following sections, the project's design and software development processes will be explained in detail.[1] [2]

1.1. Project Definition

The aim of this project is to collect data such as the user's location, time, weather conditions, and speed, analyze this information, and present various contents based on the obtained data. The fuzzy algorithm will be used in determining the contents, ensuring that the most suitable and personalized information is selected for the user's current situation. The obtained contents will be conveyed to the user audibly, allowing them to acquire information without taking their eyes off the road during driving.1.1

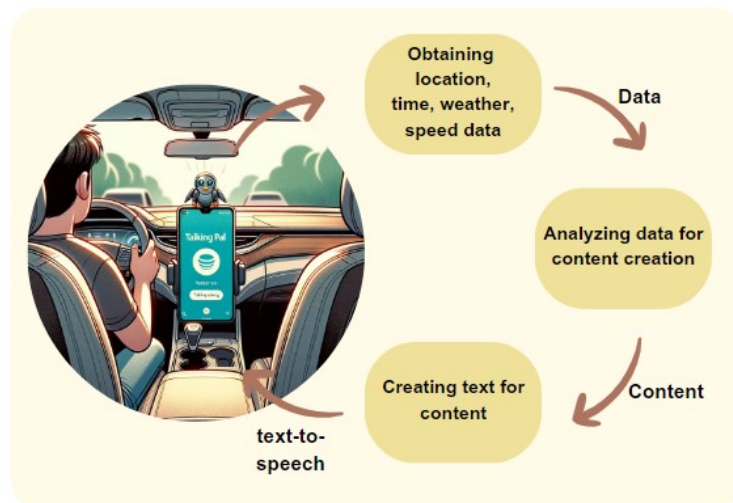


Figure 1.1: Schematic representation of the project.

1.2. Purpose and Uses of the Project

This project was chosen due to the integration of mobile application and fuzzy algorithm (artificial intelligence) fields. The project has provided experience in facilitating communication between the mobile application and the backend service. It offers a different alternative for users to obtain information during car journeys.

1.2.1. Usage Area

The 'Mobile Travel Talking Pal' project offers an innovative travel companion for travel enthusiasts. Essentially, the project is designed to provide information about your location, surrounding factors, and interests during a car journey, much like a friend sitting next to you. However, the project is not limited to use inside a vehicle; it can also be used while walking in nature or strolling through the streets to receive interesting information and enhance the walking experience. In short, the project is available for use at any time and provides personalized information.

1.3. Project Requirements

- Location Tracking and Location-Based Content Creation:
 - The application must be able to track the user's current location.
 - The application must update the user's location at a certain periodicity.
 - The application should be able to produce content about tourist attractions or points of interest depending on location.
- Time Tracking:
 - The application should be able to keep track of the current time
 - Must be able to present content according to time and day.
- Vehicle Speed Monitoring:
 - The application should be able to track the speed information of the vehicle
 - The application can give friendly warnings when the user exceeds certain speed limits.
- Weather tracking:
 - The application should be able to follow the weather information in the user's location and present content accordingly.

- Fuzzy Algorithm and Voice Communication:
 - The application should be able to determine audio content by analyzing location, time, weather and speed data with a fuzzy algorithm.
- Tracking Travel Data:
 - The application should be able to track the time the user travels.
 - The application should be able to track and present information to the user, such as how many time the user has accessed the application on which day.
- Specific Voice Command Control:
 - The application should be able to provide control with certain voice commands (for example, stopping the chat with the "stop" command, restarting the chat mode with the "start" command).

1.4. Project Planning

The application will be developed with Flutter and will collect data such as location, time, weather and speed from various sensors. The Python backend service will analyze this data, make decisions on the content using a fuzzy logic algorithm, and create content with the ChatGPT API. HTTP requests will be sent from the Flutter application to the backend service. Other APIs can be used for real-time data (like nearby places, news) and can be integrated with the ChatGPT API. Additionally, the flutter-tts will be used for text-to-speech functionality, thus enabling the application to provide audio content to the user.1.2

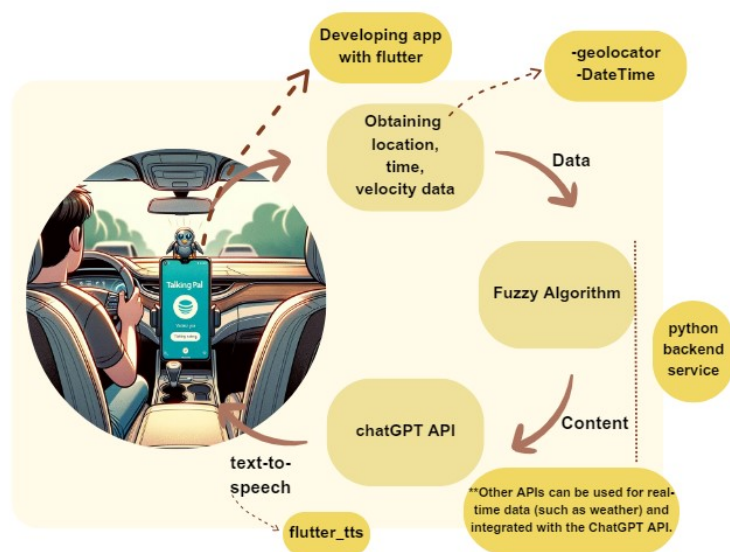


Figure 1.2: Design of the project

2. PROJECT DESIGN AND PROCESSES

In this section, the design schemes and software processes of the project are mentioned. The flow process of the project and the software operations performed are explained.

2.1. Project Schemas

We will discuss the application flowchart that illustrates the user's interaction with the application, and the flow of processes that occur in the background.

2.1.1. Application Flowchart

This flowchart illustrates the interactions of users with the 'Mobile Travel Talking Pal' application. It shows the processes of user registration and logging into the application. Fundamentally, this is how the user interacts with the application. The app is designed to be opened by the user when they get into the car and start their journey. This means that no further interaction is required from the user after they have logged into the app.^{2.1}

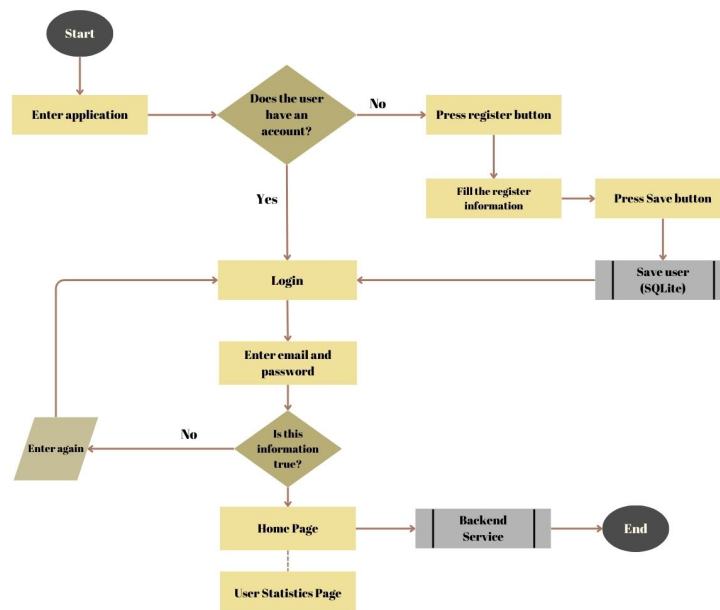


Figure 2.1: Application flowchart

The process is summarized as follows:

- If the user does not have an account, they need to register in the application. They fill in the necessary information for registration and complete the process. In the background, this registration request is sent to the backend written in Python Flask, and the user's information is stored in the database.
- If the user already has an account, they can log in to the application using the registered email and password.
- Upon logging in, the main screen opens (from where the user can navigate to the user statistics page), and from there on, processes are automatically carried out in the background.
- In the background, the user's data is collected at regular intervals and sent to the backend service, where the content is prepared.

2.1.2. Process Flow

This process diagram shows the processes that happen in the background after the user logs into the application. The different components of the application and the data flow between these components are detailed and it is shown how the application is integrated between various services. The main focus of application design is to make the user experience automatic and fluid.2.2[3]

The processes can be summarized as follows:

- When the user logs into the application, the homepage opens. At this point, requests are sent at regular intervals to the Geolocator library to obtain location information (latitude and longitude). Requests are also made to the Geocoding library based on the location information to determine the city.
- Once the location information is obtained, it is sent to the Weather Service API to get weather information for the current location. Weather information is requested every 30 minutes in the background.
- Speed information is also obtained using location libraries.
- All collected information (weather, speed, time, location) is sent to the backend service written in Flask.
- In the backend service, a topic is chosen based on the fuzzy algorithm.

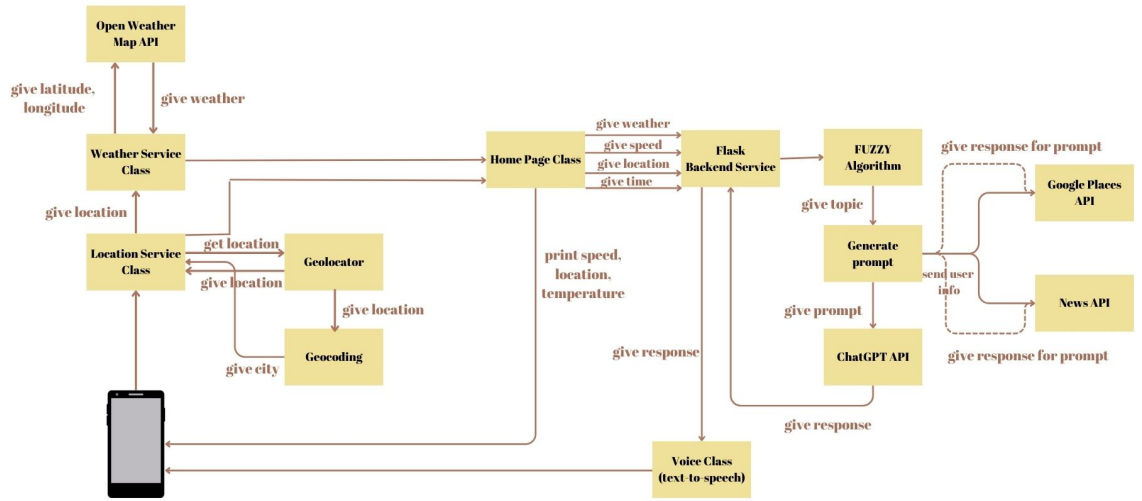


Figure 2.2: Application process flow

- The selected topic is sent to a prompt generation function. Additional APIs are also used in this part. The Places API is used to find nearby tourist attractions, shopping malls, and restaurants. The NewsAPI is used to present news related to the user's location and the topics chosen during registration.
- After the prompt is created, it is sent to the ChatGPT API, and a response is generated
- The generated response is sent back to the Flutter application and added to a queue. It is then conveyed to the user audibly in the order of the queue.

2.2. Software Processes

In this section, project development environments, the libraries required for the operations performed, fuzzy algorithm (membership functions, rules) and application interface are mentioned.

2.2.1. Development Environments

- To develop the application, Android Studio environment was established and the application interface was prepared using Flutter.

- SQLite was used to store registered user data.
- Python Flask was used for the backend part. For this purpose, PyCharm environment was established.
- Python was also used in the Pycharm environment for the Fuzzy algorithm and API communications.
- Postman was used to test endpoints written in Flask.
- Additionally, Ngrok was used to use the application on the real phone.2.3

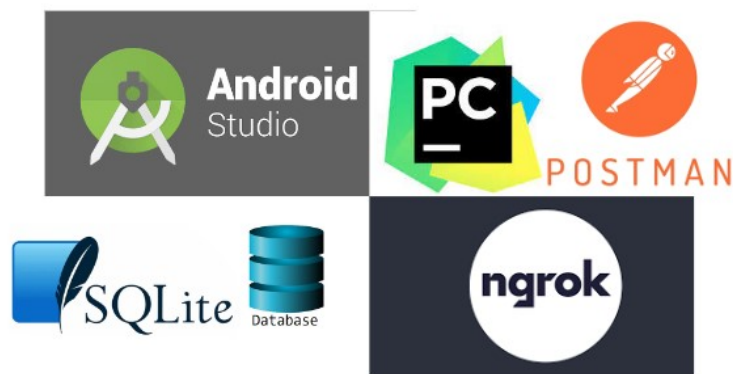


Figure 2.3: Developments environments used in project

2.2.2. Required Libraries and API's

2.2.2.1. API's

4 different APIs are used in the application.

1. ChatGPT API:

The ChatGPT API is an AI-based chatbot interface developed by OpenAI. It has been used to generate content based on written prompts. This approach allows for dynamic responses instead of fixed answers, by making requests to the API each time, aiming to create a fluid conversation. An apiKey was created to use the API. Different conversational models from OpenAI were tested (such as text-davinci-003 and text-babbage-001). Ultimately, the 'gpt-4' model was

chosen for its more accurate responses. The request format used for the API is as follows:2.4

```
def send_to_chatgpt(prompt):
    try:
        response = openai.ChatCompletion.create(
            model="gpt-4",
            messages=[
                {"role": "system", "content": prompt},
            ]
        )
        return response['choices'][0]['message']['content']

    except Exception as e:
        print(f"Bir hata oluştu: {e}")
        return "Bir hata oluştu ve cevap alınamadı."
```

Figure 2.4: ChatGPT API request

The created prompt is sent to this function and a request is sent to the API.

2. Google Places API

The Places API is a service provided by Google that offers developers tools to enable users to search and discover places in their surroundings. In the project, it has been used to find places near the user's latitude and longitude. Based on the location information sent at regular intervals, nearby tourist attractions are found. Additionally, to increase the variety of topics, it has been used to provide information to the user about nearby restaurants during lunch hours or shopping malls in the evening. An apiKey has also been created for this API. The request format used for the API is as follows:2.5

```
def find_nearby_places(latitude, longitude):
    base_url = "https://maps.googleapis.com/maps/api/place/nearbysearch/json"
    params = {
        "location": f"{latitude},{longitude}",
        "radius": 1000, # Arama yarıçapı metre cinsinden
        "type": "tourist_attraction", # Turistik yerler
        "key": apiKey,
    }

    response = requests.get(base_url, params=params)
    return response.json()
```

Figure 2.5: Places API request

This request found the user tourist attractions within 1000 meters of user location.2.6

```
def find_nearby_restaurants(latitude, longitude):
    base_url = "https://maps.googleapis.com/maps/api/place/nearbysearch/json"
    params = {
        "location": f"{latitude},{longitude}",
        "radius": 2000, # Arama yarıçapı metre cinsinden
        "type": "restaurant", # Restoranlar
        "key": apiKey,
    }

    response = requests.get(base_url, params=params)
    return response.json()
```

Figure 2.6: Places API request for restaurants

This request found the user restaurants within 2000 meters of his location.

3. News Api

The News API is a tool that allows developers to programmatically access news articles from various sources around the world. It can be used to obtain real-time news headlines for different regions, categories, and details of individual articles. In this project, it has been used to provide users with news related to their location. Additionally, to ensure a diversity of topics, further enhancements have been made. When users register for the application, they are asked to select topics of interest, and these responses are recorded in the database. These recorded topics are sent to the API, enabling the application to provide summaries of news articles on those subjects to the user. An apiKey has also been created for this API. The request format used for the API is as follows:2.7

```
def get_news_about_location(location):
    url = "https://newsapi.org/v2/everything"
    params = {
        "q": location,
        "apiKey": apiKey,
        "language": "tr" # Türkçe haberler için
    }
    response = requests.get(url, params=params)
    return response.json()
```

Figure 2.7: News API request

4. OpenWeather Api

The OpenWeather API is a service that allows developers to programmatically access weather data. In this project, it has been used to periodically retrieve weather information upon logging into the application. The obtained weather data is sent to the backend service and, along with other data, is analyzed to determine the content. For example, prompts like 'What should we pay attention to while driving in snowy and rainy weather at night?' have been created. An apiKey has also been created for this API. The request format used for the API is as follows:

```
Future<Map<String, dynamic>> getWeatherData(double latitude, double longitude) async {  
    final url = 'https://api.openweathermap.org/data/2.5/weather?lat=$latitude&lon=$longitude&appid=$apiKey&units=metric&lang=tr';  
    print(latitude);  
    print(longitude);  
    final response = await http.get(Uri.parse(url));  
  
    if (response.statusCode == 200) {  
        var data = json.decode(response.body);  
        data['weather'][0]['description'] = translateWeatherCondition(data['weather'][0]['description']);  
        return data;  
    } else {  
        throw Exception('Hava durumu bilgisi alınamadı');  
    }  
}
```

Figure 2.8: OpenWeather API request

2.2.2.2. Required Libraries

The necessary libraries, especially in the backend, are as follows:

```
from flask import Flask, request, jsonify  
import sqlite3  
from flask_cors import CORS  
import numpy as np  
import skfuzzy as fuzz  
from skfuzzy import control as ctrl  
import openai  
import random  
import requests  
from datetime import datetime  
import time
```

Figure 2.9: Required libraries

These libraries have been used for creating endpoints (Flask), communicating

with the database (sqlite3), performing calculations for the fuzzy logic system (numpy, skfuzzy), making HTTP requests (requests), and accessing the ChatGPT API (openai).

2.2.3. Fuzzy Algorithm

Fuzzy algorithm or fuzzy logic is an approach used to process ambiguous and imprecise information. The basis of fuzzy logic is based on concepts called fuzzy sets. A fuzzy set expresses the degree to which its elements belong to the set with a numerical value ranging from 0 to 1.2.10

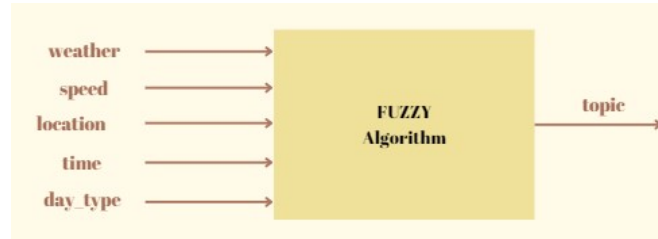


Figure 2.10: Fuzzy Schema

Fuzzy logic uses a set of rules to produce an output based on given inputs. A fuzzy logic system typically consists of four main components: input variables, membership functions, rules, and an output variable. The input and output variables have been defined as follows.2.11

```
##fuzzy
# Giriş Değişkenleri
time = ctrl.Antecedent(np.arange(0, 24, 1), label='time')
weather = ctrl.Antecedent(np.arange(0, 51, 1), label='weather')
speed = ctrl.Antecedent(np.arange(0, 200, 1), label='speed')
proximity = ctrl.Antecedent(np.arange(0, 5, 1), label='proximity')
day_type = ctrl.Antecedent(np.arange(0, 2, 1), label='day_type')

# Çıktı Değişkeni
topic = ctrl.Consequent(np.arange(0, 10, 1), label='topic')
```

Figure 2.11: Fuzzy Schema

Membership functions were determined for the input variables, and each variable was divided into different categories.(For example, the 'time' variable was divided into time periods like 'night', 'morning', 'noon', 'afternoon', 'evening').

Rules were established, and the topic of conversation was determined based on these rules.For example, if the time is late afternoon (time['evening']) and the weather is rainy

(weather['rainy']), the system can suggest movie recommendations (topic['movies']) as content. Or, if the user is near a tourist spot (proximity['near']) and it's not a special day (day_type['normal']), the system suggests topics of conversation about tourist places and interesting points in the vicinity (topic['touristic']). To avoid repetition and increase diversity in topics, random topic assignments were also made in the rules. For instance, when a user registers, they select their interests (such as stock market, science), and summaries of news related to the areas chosen by the user are made among random topics. In this way, by assigning random topics, an effort was made to create a more fluid and non-repetitive conversation.

2.2.4. Mobil Application Interface

For the project, a mobile application was made using flutter in the Android studio environment.

I designed a splash screen to appear when the application is first opened. If the screen is touched, it directly redirects to the main page. If the screen is not touched, it automatically redirects to the main page after 10 seconds.2.12

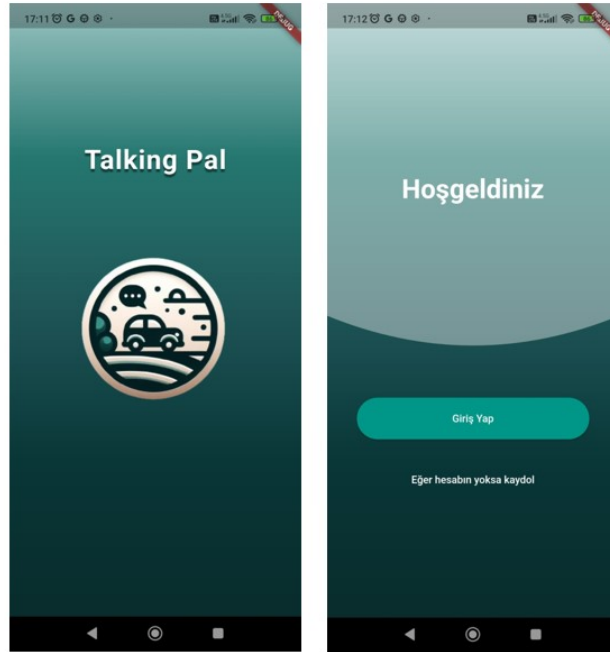


Figure 2.12: SplashScreen and MainPage

In the application, the mainPage section includes options for logging in and registering. In the registration part, users are additionally asked about their topics of interest, and the responses are saved in the database. This information is later used to

increase topic diversity when selecting subjects with the fuzzy algorithm.2.13

The figure shows three sequential screenshots of a mobile application's registration page. The first screenshot, titled 'Kaydol', displays input fields for 'İsim' (Name), 'Soyisim' (Surname), 'Yaş (isteğe bağlı)' (Age - optional), 'Email', and 'Şifre' (Password), with a green arrow button at the bottom right. The second screenshot, titled 'Bu konulardan hangisi ilginizi çekiyor?' (Which of these topics interest you?), shows a list of topics: 'Borsa' (Stocks), 'Bilim' (Science), 'Astroloji' (Astrology), 'Ekonomi' (Economy), 'Teknoloji' (Technology), and 'Uzay' (Space), each with an unchecked checkbox. The third screenshot shows the same list of topics, but with checkboxes for 'Borsa', 'Bilim', and 'Uzay' checked, indicating user selection.

Figure 2.13: Registration Page

If the user has previously registered, they can log in on the loginPage with their registered email and password.2.14

The figure shows a single screenshot of a mobile application's login page, titled 'Giriş Yap'. It features two input fields: 'Email' and 'Password', followed by a green arrow button at the bottom right. The page has a dark green background with a lighter green header area.

Figure 2.14: Login Page

When the application is logged in, the homeMainPage appears like this. The information collected from the user (weather, location, speed) is displayed on the screen. From the moment this page is opened, requests are automatically sent to the backend service at regular intervals, and the responses are conveyed to the user audibly.2.15

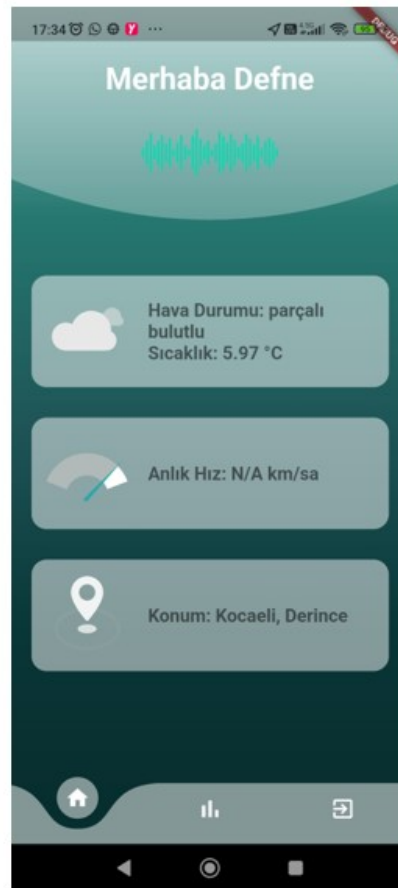


Figure 2.15: Home Page

Below the homeMainPage, there is a bottomBar. This leads to the user statistics page. There is also a logout button. On the user statistics page, it was considered to record some data (such as how many times the app was accessed on a particular day, total travel time data) to later provide a summary to the user. Currently, only the data of how many times the app was accessed on each day is graphically displayed.2.17

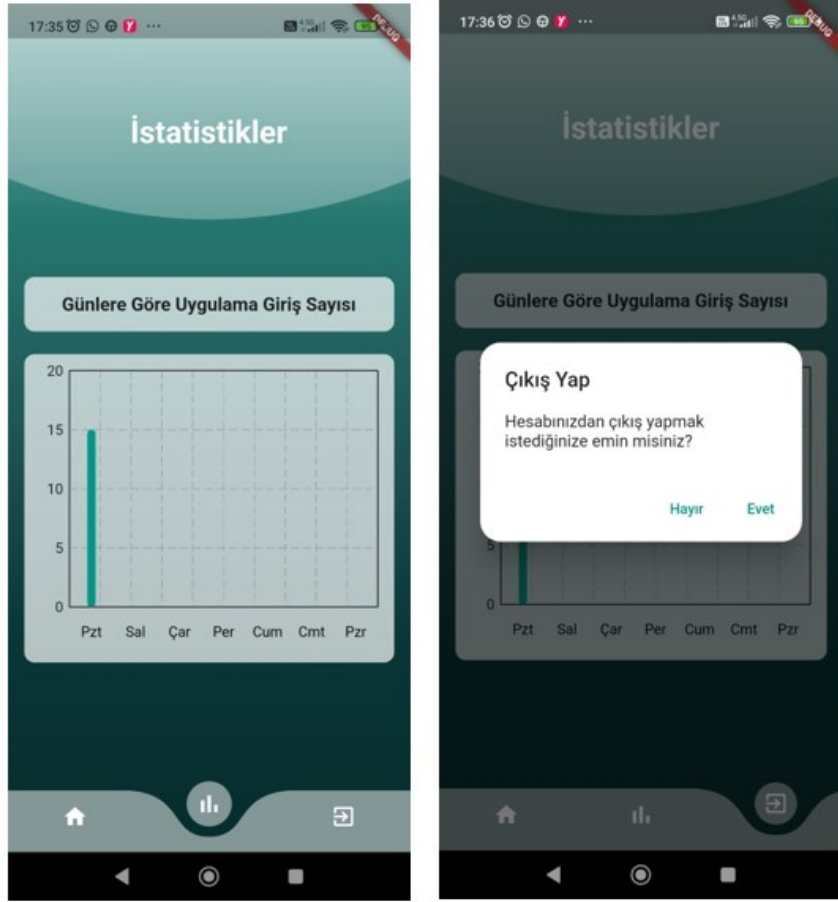


Figure 2.16: User Statistic Page and Logout

Additionally, two buttons were added to the home Page later on. The first button was added to stop or start the conversation by receiving voice commands ('stop', 'start'). The second button was similarly used to manually stop and start the conversation. The sound wave on the screen changes depending on whether or not the conversation is happening.2.17

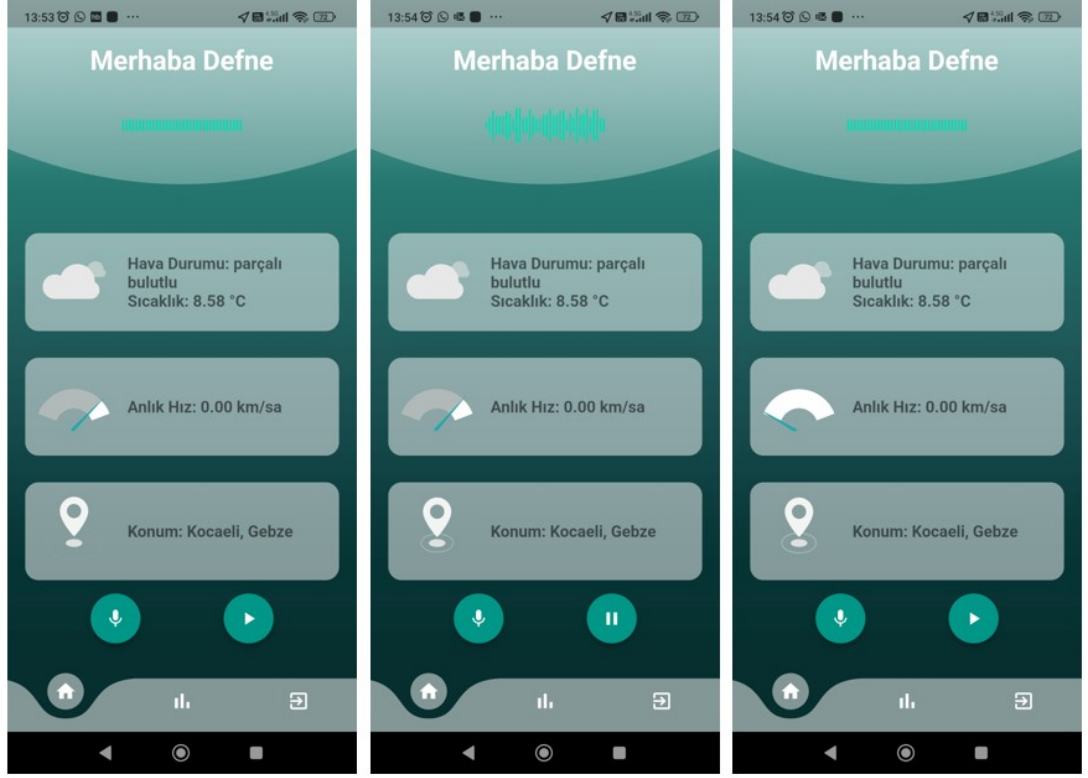


Figure 2.17: Home Page with added button

3. WORKING SYSTEM AND EVALUATION

3.1. Working System

In the project, in summary, user data is collected and sent to the backend service. There, it is processed through the fuzzy algorithm, and a topic is determined with the help of rules. Based on the topic, a prompt is created and sent to the ChatGPT API. The response received is then relayed back to the user. The test cases for the responses to be created are as follows;

Example data:

```
{ "weather": "yağışlı", "speed": 121.0, "location": "Kocaeli, Derince", "latitude": 40.7553141, "longitude": 29.8069233, "time": "14:10:02", "date": "2024-4-23", "nearby_places": ["Çınarlıdere Cami"] }
```

- Warning when speed exceeds 120 km/h.3.1

```
{
  "chat_response": "Hızın oldukça yüksek, 121.0 km/sa! Bu hızda sürüş yaparken dikkatli olmak çok önemli. Unutma ki güvenli sürüş, sadece senin hayatını değil, aynı zamanda diğer yol kullanıcılarının hayatını da korur.\n\nGüvenli sürüş yapmak, sadece trafik kurallarına uymakla elde edilmez. Aynı zamanda dikkat, öngörü ve karar verme becerilerini de gerektirir. Hızını kontrol altında tut ve hız limitlerine uyma konusunda disiplinli ol. Yüksek hızlar sadece reaksiyon süreni azaltmakla kalmaz, aynı zamanda bir kaza durumunda yaralanmaların daha ciddi olma ihtimalini de artırır.\n\nAyrıca, hızını trafik, yol ve hava koşullarına göre ayarlamana gerektiğini unutma. Hava koşulları kötüyse veya yoğun trafik varsa, hızını bu durumlara göre düşürmelisin. İyi bir sürücü, her zaman rahat ve kontrollü bir sürüş stili benimser.\n\nSon olarak, emniyet kemeri her zaman bağla ve cep telefonu gibi dikkat dağıtıcı unsurlardan kaçın. Kemeri bağlamak, bir kaza durumunda hayatını kurtarabilir ve cep telefonu kullanmak hem senin hem de diğerlerinin güvenliğini tehlikeye atar.\n\nUnutma, temel amacın her zaman en hızlı şekilde gitmek değil, güvenli bir şekilde varış yerine ulaşmaktır."
  "message": "Veriler başarıyla alındı"
}
```

Figure 3.1: chatGPT Response

- Talking about it when there is a tourist area nearby. 3.2

```
{
  "chat_response": "Çınarlıdere Cami, Kocaeli'nin Derince ilçesi içerisinde bulunan önemli bir dini yapılarındandır. Bu cami, doğal ve huzurlu bir ortamda bulunmasıyla bilinir.\n\nCaminin adı olduğu gibi, çevredeki çok eski çınar ağaçlarından gelir. Çınarlıdere Cami'nin çevresi çok sayıda çınar ağacı ile çevrilidir ve bu ağaçlar, caminin peyzajını ve çevresini eşsiz bir şekilde güzelleştirir. Bu ağaçlar, camiye sadece bir dini ibadet yeri olmaktan çıkarak, aynı zamanda ziyaretçilere huzurlu ve dingin bir alan sunar.\n\nCami, geleneksel Osmanlı mimari tarzında inşa edilmiştir ve bu özellik, onu Derince'deki diğer dini yapılarından ayırır. İçerisi, sanatsal detaylar ve süslemelerle donatılmış olup ziyaretçilere muhteşem bir görüntü sunar. Caminin iç tasarımı, ışığın zarif bir şekilde kullanılmasını sağlayan vitray pencereler ile doludur.\n\nÇınarlıdere Cami, özellikle Cuma ve Ramazan gibi dini günlerde topluluğun buluşma noktasıdır. Bu zamanlarda, bölge sakinleri ve farklı yerlerden gelen ziyaretçilerle birlikte cami, eşsiz bir atmosfere sahip olur.\n\nSonuç olarak, Çınarlıdere Cami, eşsiz mimarisi, çevresiyle uyumlu peyzajı ve huzurlu atmosferi ile Derince'nin en önemli dini ve turistik mekanlarından biridir. Ayrıca, cami ve çevresindeki çınar ağaçları, ziyaretçilere doğanın ve mimarinin mükemmel birlikteliğini sunar. Bu yüzden, bir cami ziyaretinin ötesinde, Çınarlıdere Cami'yi ziyaret etmek, doğa ile iç içe bir huzur deneyimi yaşamak anlamına da gelir."
  "message": "Veriler başarıyla alındı"
}
```

Figure 3.2: chatGPT Response

- Informative speech for health according to the weather. 3.3

```
{
  "chat_response": "Soğuk havalarda sağlıklı kalmak için çeşitli yollar mevcuttur. Öncelikle bağışıklık sisteminizi güçlendirmek önemlidir. Sağlıklı ve dengeli bir beslenme, yeterli miktarda uyku ve günlük egzersiz bağışıklık sisteminizi güçlendirebilir. Özellikle vitamin ve mineral bakımından zengin yiyecekler tüketmek önemlidir.\n\nBunun yanı sıra, vücut ısını koruyacak şekilde giyinmek de vazgeçilmezdir. Soğuk hava, vücutun enerjisini koruması için ekstra çaba sarf etmesini gerektirir, bu yüzden sıcak kıyafetler ve tabakalar sayesinde vücudunuz gereksiz enerji tüketimini önleyebilir. Ayrıca, vücutun soğuk hava nedeniyle kuruyan derisini nemlendirmeyi unutmalısınız.\n\nAyrıca, soğuk algınlığı ve grip gibi tipik kış hastalıklarından korunmak için ellerinizi düzenli olarak yıkamalısınız. Eğer hasta olan biriyle temas ettiyseniz, gözlerinizi, ağrınıza veya burnunuza dokunmama dikkat edin çünkü bu, virüslerin vücuda girmesine yol açabilir.\n\nSon olarak, evin içerisinde düzenli olarak havalandırmak da iyi bir fikirdir. Bu, evin nem seviyesini dengede tutarak mevcut hastalıkların yayılmasını önlemeye yardımcı olabilir.",
  "message": "Veriler başarıyla alındı"
}
```

Figure 3.3: chatGPT Response

- View and summarize the latest news on a topic of interest to the user.3.4

```
{
  "chat_response": "Borsa İstanbul'da bugün pek de iyi bir tablo çizilmedi ne yazık ki. Satış ağırlıklı seyreten Borsa İstanbul BIST 100 endeksi, günü yüzde 2,67 değer kaybıyla kapattı. Endeks, 7.557,56 puanla günü tamamladı. Kısacası, borsa bugün düşüşe geçti diyebiliriz. Bu durum, yatırımcıları biraz üzebilir. Havanın biraz kasvetli olduğu bu borsa gününde, dikkatli adımlar atmakta fayda var. Eminim daha iyi günler de göreceğiz. Bunu unutmayın!",
  "message": "Veriler başarıyla alındı"
}
```

Figure 3.4: chatGPT Response

- Provides information to the user about the food of the region and tells the restaurants nearby during lunch hours.3.5

```
{
  "chat_response": "Kocaeli, Derince bölgesi, birçok lezzetli bir arada sunabilen bir gastronomi cennetidir. Yöresel yemeklere olan sevgisi ve bu lezzetlerin kaliteli ve özgün şekillerde sunulduğu mekanları ile bilinir. Derince'deki popüler yemekler arasında etli ekme, pide, iskender kebab ve döner öne çıkar.\n\nYakınızdaki İkrar Kebab Salonu ve Öz Anadolu Kebab Salonu adlı iki farklı restoran bulunuyor. İkrar Kebab Salonu, özellikle kebabları ile bilinen bir yer. Burada yapılan kebablar, yerel baharatlar ve özgün tarifler ile hazırlanır. Hem ızgara hem de aşkım yemekleri için uygun bir ortam sunuyor. Kebab çeşitleri arasında Adana, Urfa, Beyrit ve Iskender gibi seçenekler bulunuyor.\n\nÖz Anadolu Kebab Salonu ise adından da anlaşılacağı üzere Anadolu'nun çeşitli bölgelerine ait kebabları sunuyor. Özellikle Adana ve Urfa kebabları oldukça popüler. Burası da genelde hem ızgara hem de aşkım yemekleri için tercih ediliyor.\n\nHer iki restoran da, geleneksel Türk yemeklerini sevenler için ideal seçenekler sunuyor. Karar vermede zorlanıyorsanız, hangi tür kebabi veya ne tür bir lezzet denemek istediğinizi düşünerek karar vermeniz faydalı olabilir. Peki, siz genellikle Türk mutfağında hangi tür yemekleri tercih ediyorsunuz?\n",
  "message": "Veriler başarıyla alındı"
}
```

Figure 3.5: chatGPT Response

- Talking about current news related to the user's location.3.6

```
{
  "chat_response": "Konumunuzla alakalı son haberleri bir gözden geçirdim ve dikkat çeken bir haber buldum. AK Parti Genel Başkanı ve Cumhurbaşkanı Recep Tayyip Erdoğan, 26 belediye başkan adayını Haliç Kongre Merkezi'nde düzenlenen AK Parti Aday Tanıtım Toplantısı'nda açıkladı. Bu isimler arasında Kocaeli Büyükşehir Belediye Başkan adayı olarak Tahir Büyükkakın'ın adı da yer alıyor. İlgili haberin detaylarına bakarsak; Tahir Büyükkakın'ın kim olduğunu, politik sıfatını ve bu süreçte neler yapmayı hedeflediğinden bahsediliyor. Kesinlikle takip etmekte fayda var.\n",
  "message": "Veriler başarıyla alındı"
}
```

Figure 3.6: chatGPT Response

3.2. Difficulties Encountered During the Project

The project started with research on APIs. After that, the development of the application began. Flutter was chosen due to some prior experience in application development. There was no previous experience in writing algorithms for fuzzy logic systems, so Python was identified as the most suitable language for this purpose. Therefore, in the Flutter part, data was collected and requests were made to an endpoint written in Python Flask. Although the codes worked correctly when tested in the emulator with Android Studio, results could not be obtained, and continuous errors occurred when the application was loaded onto a phone and requests were attempted to the service. This issue was resolved by using ngrok after some research. Subsequently, there were difficulties in finding tourist attractions based on location. While location information was accurately retrieved from the application, the ChatGPT API did not provide information about specific locations, so the nearest tourist attractions could not be found. It only provided information about all the tourist areas in a region, for instance, if the location was in Kocaeli, it would mention all tourist areas in Kocaeli. Therefore, it was decided to use another API for this part. The Google Places API was researched. Consequently, the location information from the incoming data was sent to the Places API to obtain information about nearby places, and this place name was sent to the ChatGPT API to provide information about the place.

Another challenge encountered was related to the ChatGPT API. The initial model used in the project was discontinued, and changes had to be made in the code to use the newly released models. Furthermore, responses from the first new model used were not as fast as the previous one. The response time becomes even longer if the internet connection is slow.

3.3. Evaluating the Success Criteria

The success criteria of the project were determined as follows.

1. A survey should be conducted with 10 users, and the results of the survey should yield over 80% satisfaction. (for contents)
2. A survey should be conducted with 10 users, and the results of the survey should yield over 80% satisfaction. (for auditory output)
3. The application should correctly recognize and respond to the user's voice commands, "dur", and "başla".

At the end of the project;

- Communication between the Flutter application and the Python Flask backend was established.
- Collection of user data in the background and its regular transmission to the backend system were provided.
- Content and prompts were determined based on the data received from the user, and a response was generated by making requests to the API.
- Providing specific content based on the information filled out by the user during registration was provided.
- Audio output for the user was implemented. When multiple messages returned from the service accumulated, simultaneous audio reading of these messages was controlled and prevented to ensure no overlapping of sound.
- Additionally, a button was added to detect voice commands ('stop', 'start') and to manually initiate and halt the conversation.
- The application design and content creation were well-received by the users.

4. CONCLUSIONS

The necessary requirements for the project were determined. APIs to be used in the application were researched, and decisions were made on which APIs should be used. After the research phase, a mobile application focused on the user interface was developed.

When the application is accessed, the collection of user data in the background and its periodic transmission to the backend service were ensured. API keys were generated for the chosen APIs, and they were integrated into the application. The incoming data was evaluated with the fuzzy algorithm, a topic was decided upon, and content was created and sent as an API request. The resulting output was sent back to the application and conveyed to the user audibly. Adjustments were made to prevent messages from overlapping and to avoid delays in sound transmission.

Since API requests are made from the moment the application is accessed, it was observed that the application slows down and responses are delayed when the internet connection is slow. In the future, a structure that automatically operates under poor internet conditions could be implemented.

In conclusion, the application has been designed to be easily used in daily life. During the development phase, optimizations could be made regarding the speed of delivering responses to the user for more widespread usage.

BIBLIOGRAPHY

- [1] J. C. Ferreira, “A personal travel assistance for electrical vehicles in smart cities,” 2011.
- [2] A. Kodre, “Smart and efficient personal car assistant system,” 2018.
- [3] Fernando Putra. “Get current location-geolocator.” (), [Online]. Available: <https://medium.com/@fernandoptr>.