Software Engineering

for

Gas Station Project

Version 1.1 approved

Prepared by Gizem Turan

SRS Documentation

Table of Contents

Table of Contents ii

1. Introduction 1

1.1 Purpose 1

1.2 Project Members 1

1.3 Technical Assumption and Constraints 1

1.4 Naming Conventions 1

1.5 References 1

2. Overall Description 2

2.1 System Environment 2

3. Requirements 2

3.1 Functional Requirements and Specifications 3

3.2 User Characteristics 4

3.3 Non-Functional Requirements 4

3. System Architecture and Architectural Design 7

3.1 Use Case View 5

3.2 Use Case Descriptions 5

3.3 Logical View 5

5. Entity Relationship Diagram 7

6. Design and Implementation 7

7. References 8

**1.Introduction**

In this section contains the general information about our project, its improvement stages and the reason why people need it.

**1.1.Project Purpose and Scope**

Our purpose is helping local drivers for finding gas stations via a mobile app which is a reliable, robust and easy-to-use tool for users. Also, with a click, clients will be able to see the locations and gasoline prices of the closest gas stations to supply a comfortable experience to clients.

This software system provides accessibilty all over Turkey via the application’s maintenance of daily bases datas. The system also contains a relational database which includes the most known gasoline brands.

**1.2.Project Members**

The students who submitted these team projects are,

1. Gizem TURAN: SRS Author, Web Site, Web Scraping, Database Setup

2. Buğra Meriç DEĞİRMENCİ: Mobile Application and GUI Developer with Kotlin

3. Osman Can İSKİT: Web Scraping, Database Setup, Firebase Developer, Web Site

4. Dursun Alperen ÇOBAN: Price Prediction (Machine Learning), Documentation

5. Aziz Doğay EKİNCİ: Price Prediction (Machine Learning)

**1.3.Technical assumptions and Constraints**

Our application’s Technical Assumptions are based on basic specifications.

System requirements are:

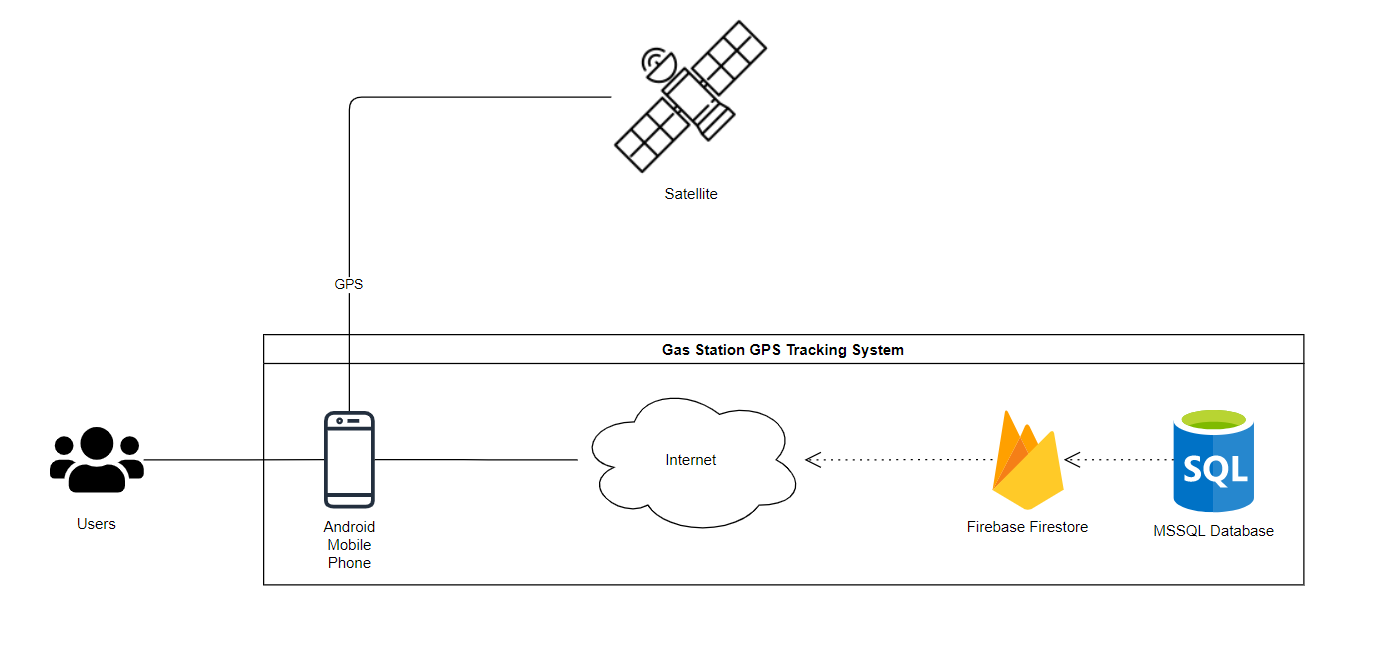
* Android operating system 9.0 or higher
* 256 MB ram or higher
* 100 MB free space or higher
* Internet connection
* GPS

The system requires firebase and api’s for database adjustments. Our team have chosen to use Kotlin, C#, Python for coding language.

**1.4.Naming Conventions**

Our team have chosen to utilize camelCase, Pascal Case and snake\_case conventions according to our requirements.

**2.Overall Description**

**2.1.System Environment**

The users are connected to the internet through the satelitte via their smart phones. They can access/see the nearest gas stations and their prices coming through cloud firestore which is backed up to MSSQL.

**3.Functional Requirements**

**3.1 Functional Requirements**

1) User’s mobile phone must be connected to the internet

2) User must turn on GPS location on their mobile phone.

3) User must give GPS access permission to our application.

**3.2.Non- Functional**

**Requirements**

**3.2.1 Performance Requirements**

1) Launcing our app should not take more than 3 seconds including our splash screen. Our app has minimal UI elements and it is optimized to launch relatively fast.

2) Getting the location of the user should not take more than a second depending on the device, the strenght of the GPS line.

3) Loading the price results of individual stations in our prices screen should noy take more than 2 seconds depending on the internet connection strenght.

4) The functionality of the “Route” button should take less than a second to work, depending on the speed of the mobile device.

5) Settings button should take less than a second to work.

6) The refresh button should refresh the app and apply the settings in two seconds depending on the internet connection.

7) Increasing the search area to the maximum value can cause lag, depending on the mobile device.

**3.2.2 Safety and Security Requirements**

We do not have any safety and security requirements.

**3.2.3 Software Quality Attributes**

**3.2.3.1 Portability**

Our application is designed for mobile devices, so it is relatively easy to use and it is very portable

**3.2.3.2 Reusability**

Our application is desgined to be reusable. We update our data every day to ensure our users the current prices and predictions.

**3.2.3.3 Maintainability**

Our application is easy to maintain. The application is easy to debug and implement new features. Unless the web sites we are getting our price information fails to update or doesn’t work, we can maintain our app as long as it exists.

**3.2.3.4 Usability**

Our application has minimal and user-friendly interface. Anyone can learn how to use our app.

**3.2.3.5 Reliablity**

Our app does not hold any sensetive personal data. So any data loss or theft would not harm our app or our users.

**4. System Architecture and Architectural Design**

**4.1.Use Case Overview**

This part includes outlines of the use cases.

**4.1.1 USE CASE: Search Gas Stations**

**Diagram:**

User

Search Gas Stations

**Brief Description:** The user can see the nearest gas station based on their location.

**4.1.2 USE CASE: Search Gasoline Prices**

**Diagram:**

User

Search Gasoline Prices

**Brief Description:** The user can find out the individual gasoline prices in their nearest gas stations.

**4.1.3 USE CASE: Search Future Gasoline Prices**

**Diagram:**

User

Search Future Gasoline Prices

**Brief Description:** The user is able to see the estimation of the future gasoline prices.

**4.1.4 USE CASE: Update for The Future Gasoline Prices**

**Diagram:**

System

Update Future Gasoline Prices

**Brief Description:** The system must display updated version of the gasoline prices based on the current prices which is coming from gas station sites.

**4.1.5 USE CASE: Route to Selected Gas Station**

**Diagram:**

User

Route to Selected Gas Station

**Brief Description:** The user is redirected to Google Maps application after pressing the route button. The app automatically plots the route to the selected gas station.

**4.2.Use Case Descriptions**

### 4.2.1 Search Gas Stations

|  |  |
| --- | --- |
| **Use Case Name** | Search Gas Stations |
| **Trigger** | The user opens the application with internet and GPS on. |
| **Precondition** | The user must have GPS and internet connection. |
| **Postcondition** | The user has successfully connected to the internet. |
| **Exception** | The user might not access internet connection. |

### 4.2.2 Search Gasoline Prices

|  |  |
| --- | --- |
| **Use Case Name** | Search Gasoline Prices |
| **Trigger** | The user presses the marker on a map. It displays certain gas station brand icon. |
| **Precondition** | The user must have GPS and internet connection. |
| **Postcondition** | The user has successfully connected to the internet. |
| **Exception** | The user might not access internet connection. |

### 4.2.3 Search Future and Current Gasoline Prices

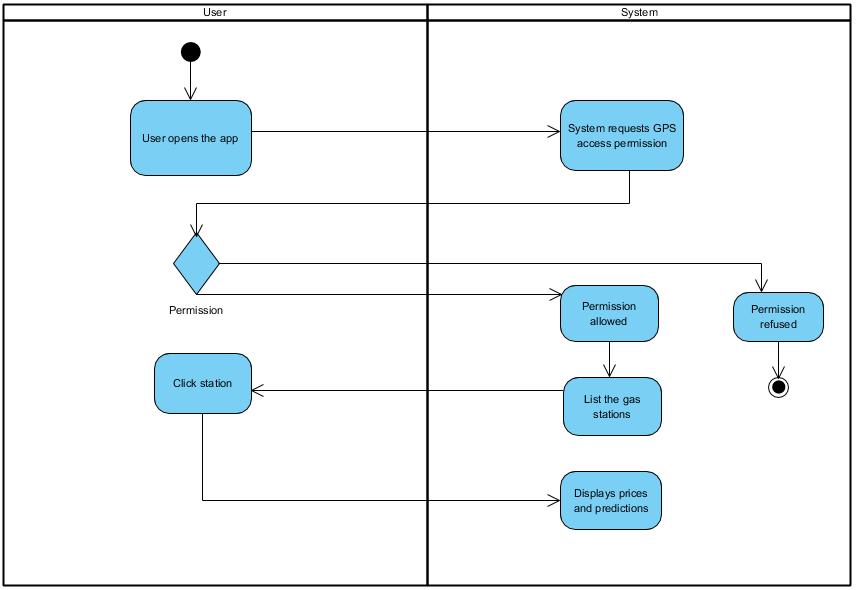
|  |  |
| --- | --- |
| **Use Case Name** | Search Future and Current Gasoline Prices |
| **Trigger** | The user presses the marker on a map. It displays certain gas station brand icon. |
| **Precondition** | The user must have GPS and internet connection. |
| **Postcondition** | The user has successfully connected to the internet. |
| **Exception** | The user might not access internet connection. |

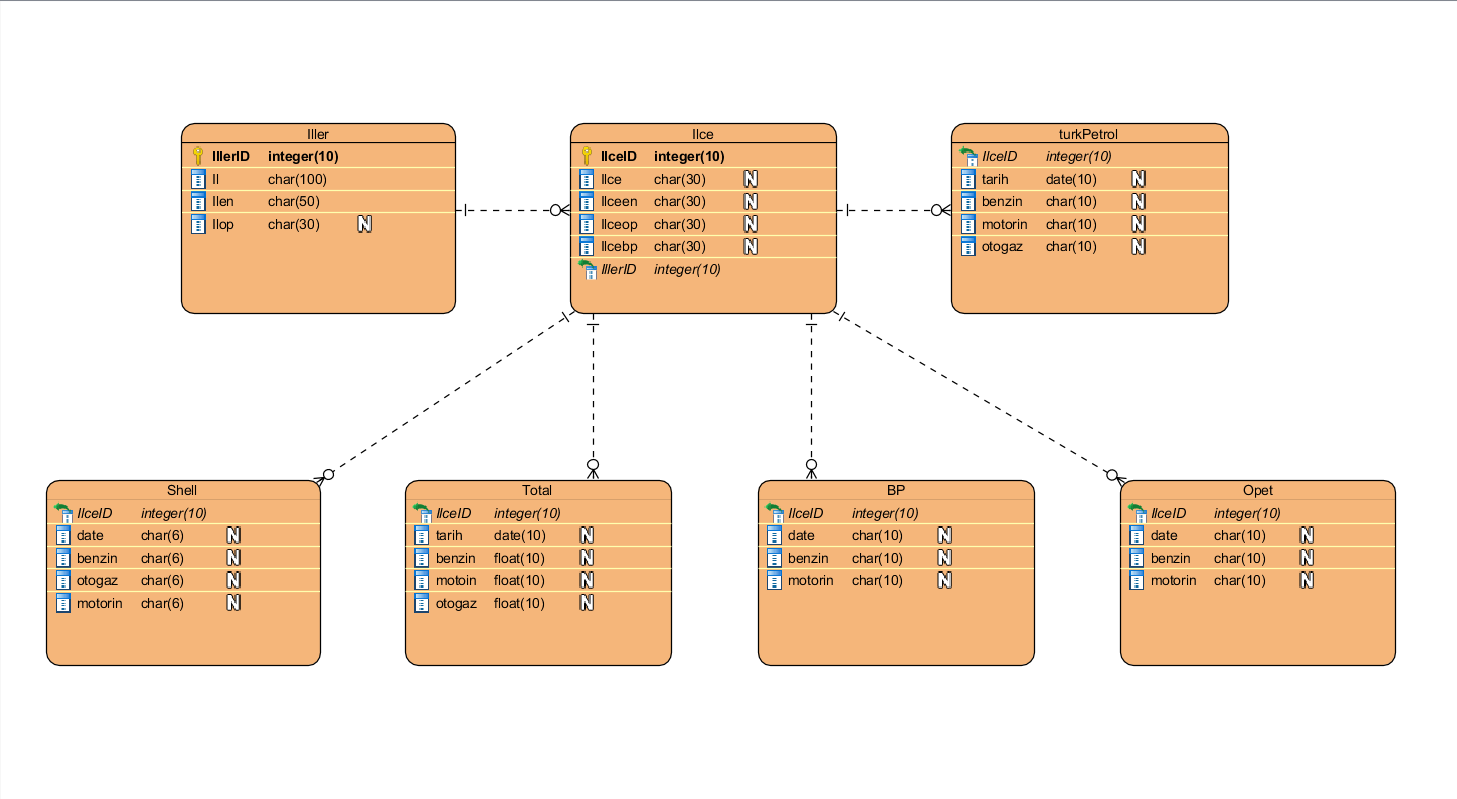
### 4.2.4 Update for the Future and Current Gasoline Prices

|  |  |
| --- | --- |
| **Use Case Name** | Update for the Future and Current Gasoline Prices |
| **Trigger** | The database refreshes itself day by day according to current datas. |
| **Precondition** | The database must be connected to the internet and to the site that was scraped datas. |
| **Postcondition** | The database has been updated. |
| **Exception** | The python script that updated to the database cannot connected to the site. |

### 4.2.5 Route to Selected Gas Station

|  |  |
| --- | --- |
| **Use Case Name** | Route to Selected Gas Station |
| **Trigger** | The user presses the route button. |
| **Precondition** | The user must select a gas station before pressing the route button. |
| **Postcondition** | The user is redirected to Google Maps application. |
| **Exception** | The user have not selected a gas station before pressing the button. |

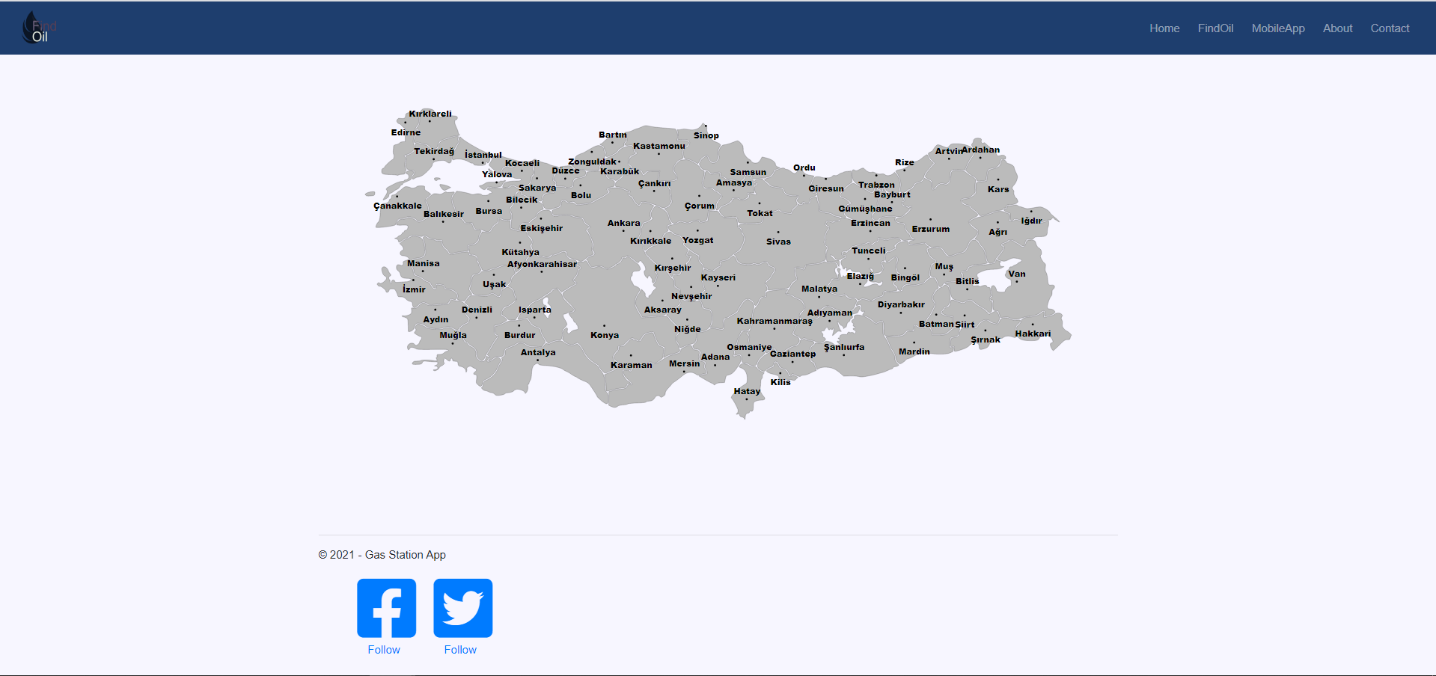
**4.3.Logical View**

**5.Entity Relationship Diagram**

**6.Design and Implementations**

Web Site Home Page:



Web Site Interactive Map:

This is an interactive map of Turkey which users can select a city and a district to see the gas price results.

**Application Splash Screen:**



This is our splash screen. It displays our logo. This appears everytime our application is launched.

**Application Home Screen:**

This is our application’s home screen. This is the first view after the pslash screen. We can see the nearest gas stations displayed on the map as their respective logos.



This is our navigation bar including the “Route” , “Show Prices” “Settings” buttons. The “Route” button redirects the user to an external map application and draws a route from their location to the selected gas station. To use this button, user must select a gas station from the map first.

The “Show Prices” button opens a prices menu that shows our prices and predictions for a selected gas station on the map.

The “Settings” button redirects the user to the settings page.

**Application Price Menu:**



Refresh Button

The below menu is our Prices and Predictions screen. After selecting a gas station on the map, you can Access themenu and close it at any time.

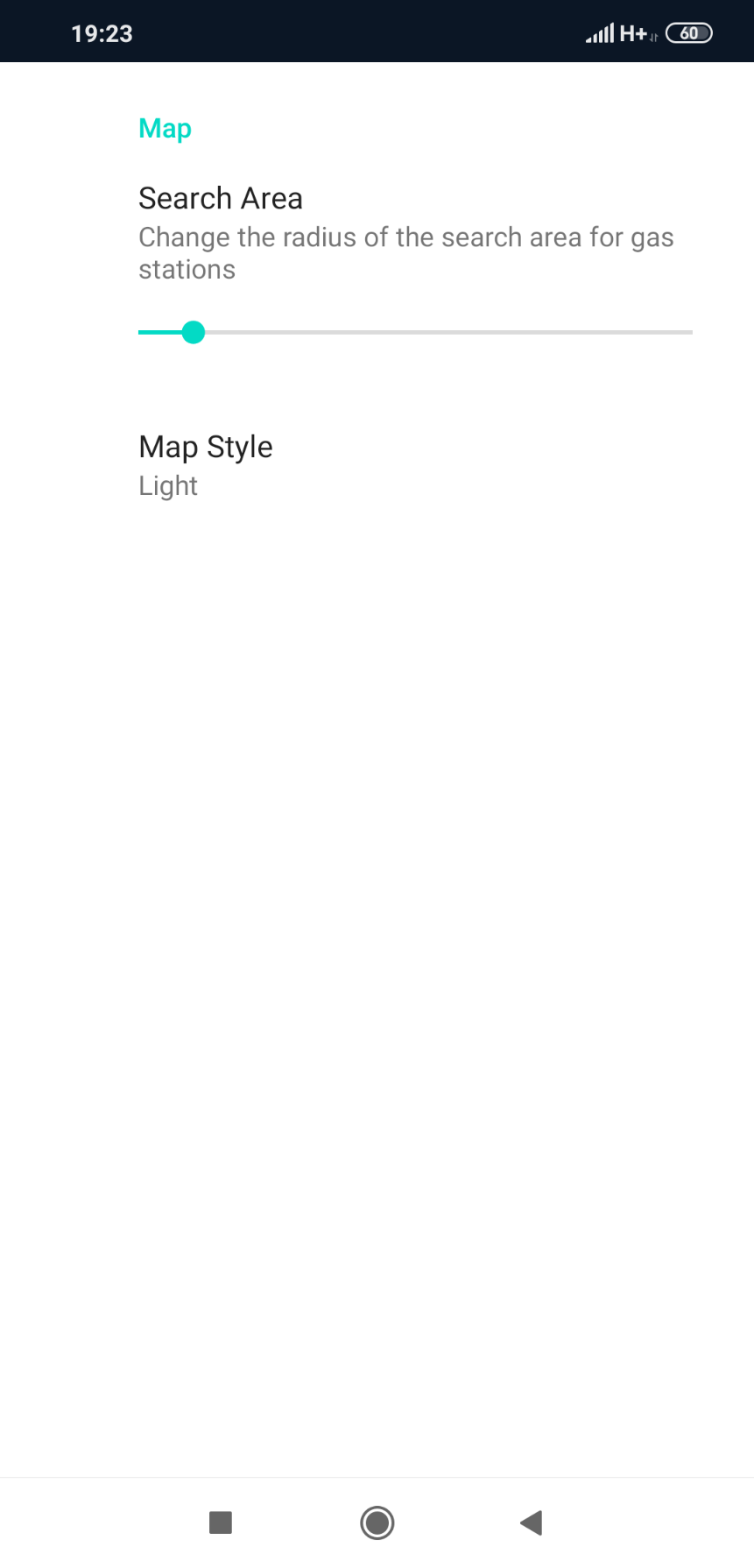
This menu shows the user, the selected gas station’s name, the selected gas station’s address, tomorrow’s date, the current gas prices, the future gas prices, the error margins of those price predictions.

Price prediction’s error margin

Future price prediction for “Benzin”

Current price for “Benzin”

**Application Settings Menu:**



This is the settings menufor setting up user preferences. The preferences are stored so everytime user opens our application, the preferences are stored.

Search area slider sets the search area fort he nearest gas stations.

User must restart the app or use the “Refresh” button to see the effect of the “Map Style” preference.

**7.References**

**Machine Learning:** <https://www.youtube.com/watch?v=QIUxPv5PJOY&list=WL&index=12&t=1466s>

**HTML,CSS:** <https://www.youtube.com/watch?v=KcdBOoK3Pfw&ab_channel=DevEd>

**Kotlin:**

<https://developers.google.com/maps/documentation/android-sdk/marker#maps_android_markers_add_a_marker-kotlin>

<https://www.youtube.com/watch?v=TN97RRR-7bk&ab_channel=BintuHarwani>

**Encoding:**

<https://forum.yazbel.com/t/locale-setlocale-locale-lc-all-tr-tr-yapisinda-sorun-yasiyorum-turkce-karakter-sorunu/476/5>

**Web Scraping:**

<https://www.youtube.com/watch?v=o83WYkwNslU&t=448s&ab_channel=Aporlorxl23>

**Python and Machine Learning:**

<https://www.btkakademi.gov.tr/portal/course/python-ile-makine-ogrenmesi-11800#!/about>

**Firebase and Project Development:**

<https://www.btkakademi.gov.tr/portal/course/firebase-ile-proje-gelistirme-15059#!/about>

**Introduction to Data Science with R**

<https://www.btkakademi.gov.tr/portal/course/r-ile-veri-bilimine-giris-13050#!/about>