ÇANKAYA UNIVERSITY

FACULTY OF ENGINEERING

COMPUTER ENGINEERING DEPARTMENT

PROJECT REPORT

CENG 407

INOVATITE SYSTEM AND DEVELOPMENT 1

RECOMMENDATION SYSTEM FOR TRAVELERS

PROJECT ID: 2022-05

TAHA KOYUTÜRK	201811044
HÜSEYİN DİKBASAN	201611014
PELİN SU GÖK	202011405
ABDULLAH ÖMÜR ŞENOCAK	201811403
CEREN EROĞLU	201811029

ADVISOR: PROF. DR. AHMET COŞAR

Contents

Abstract

In general, people like to travel, see new places, explore and share the memories they have spent there with people by photographing them. We see this a lot on social media. Based on this idea, we thought to collect and develop these activities together thanks to our mobile application Dec. Users using this application can recommend the places they visit to other users by voting, sharing photos, commenting. They can photograph their memories of these places and share them with people on their profile. In addition, our application is similar to the places that users visit, it can recommend places that the user may like depending on their location and, if desired by the user, it can draw a road map so that October can easily go to the proposed location. In this way, users in the system can see more places, discover more different places and also share the locations they have visited with users.

Özet

İnsanlar genellikle gezmeyi, yeni yerler görmeyi, keşfetmeyi ve oralarda geçirdikleri anıları fotoğraflayarak insanlarla paylaşmayı severler. Buna sosyal medyada da çok gez görmekteyiz. Bu düşünceden yola çıkarak bu aktiviteleri mobil uygulamamız sayesinde bir araya toplayıp geliştirmeyi düşündük. Bu uygulamayı kullanan kullanıcılar gezdiği yerleri oylayıp, fotoğraf paylaşıp, yorum yaparak diğer kullanıcılara önerebilirler. Bu yerlerdeki anılarını fotoğraflayıp profilinde insanlarla paylaşabilirler. Ek olarak uygulamamız kullanıcıların gezdiği yerlere benzer, kullanıcının beğenebileceği yerleri konumlarına bağlı olarak önerebilir ve eğer kullanıcı tarafından istenirse önerilen konuma rahatlıkla gidebilmesi için bir yol haritası çizebilir. Bu sayede sistemdeki kullanıcılar daha fazla yer görebilir, daha farklı yerler keşfedebilir ve ayriyeten ziyaret ettikleri konumları kullanıcılarla da paylaşabilirler.

1. INTRODUCTION

1.1 Motivation

Over the years, sociability among people, recommendations on the internet and interest and sharing in places evaluated, the more widespread social media, travel planning have increased a lot. In this regard, there is no mobile application that combines all these features and more. This situation made us think about the mobile application we made during the project process.

1.2 Problem Statement

Due to the lack of a mobile application that is widely used today, which contains most of the requirements, the users cannot make their travel plans suitable for them, and the places they go afterwards can be liked with a correct and appropriate planning, but they are not liked. In addition, situations caused by the inactivity of user communication, the uncertainty of the good or bad features of that place, due to the fact that other users' opinions about the places they have been to before are not stated. Finally, due to the lack of a mobile application that contains most

of the requirements, the lack of a fully efficient and fast service to the users, therefore, the users may not notice the places suitable for them in a particular region, causing them to search for places suitable for them for a longer period of time, causing many users to give up on travel. may cause. As a result, it may indirectly cause a decrease in travel rates. As an example of some commonly used applications; Travel Shop Turkey, Google Maps, City Mapper, Live Trekker, Time Out, Zomato.

1.3 Solution Statement

In the common applications we have designed and similar, it will maintain the existence of all necessary features, sociability and communication, as well as make recommendations according to the preferences and tastes of the users, make the users happier from the place they travel, express their thoughts about the places visited within the framework of certain etiquette rules and go there. By making an application that will give ideas to other users who want, in short, that will both increase sociality and help plan the travel better and more comfortable than users, users can be used faster and more actively.

2. LITERATURE REVIEW

Abstract

Travel has always been an innate desire of human beings. These days travel and adventure have become the most trending entertainment as well. With the development of information technology and social media, there are numerous possibilities and opportunities in fetching suited information that can yield to setting up an appropriate travel plan and hence enhance the quality of travel. The significance of a Recommender System (RS) has become extremely common in recent years. It helps users to discover information and settle on choices that they prefer. It is a software tool and technique that provide a suggestion based on the customer's taste to come upon a new appropriate thing for them by filtering personalized information based on the user's preferences from a large volume of information. Recommendation Systems (RS) have been extensively utilized as a means of reducing information overload and offering travel recommendations to tourists. [1] The aim of recommendation systems for tourism is to enrich the tourists' experience by recommending rich multimedia content, context-aware services, and views/ratings of peer users. This survey reviews mobile tourism recommendation systems and provides insights into their offered services. We conducted a detailed review and chronological evolutions of various methods and techniques used in the travel and tourism sector and compared them to our project.

Keywords: Recommendation system, Types of the recommendation system, Travel, and Tourism

2.1 Introduction

The explosive growth of online environments has made the issue of information search and selection increasingly cumbersome; users are overwhelmed by options that they may not have the time or knowledge to assess. . According to Google statistics, about 80 percent of people seek information from the web for planning their upcoming holidays. Another analysis uncovers that 52% remain positive while 48% reexamine their travel arrangements based on reviews on online platforms. [2] Recommender Systems (RSs) have proven to be a valuable tool for online users to cope with information overload. [3] A relatively recent development recommendation system lies in the use of mobile devices as a primary platform for information access. The unique characteristics of mobile tourism bring forward new challenges and opportunities for the evolution of innovative personalized services. Recommender Systems were implemented in many domains like social websites like Facebook, Entertainment like Netflix, and YouTube, Ecommerce like Amazon, and Flipkart, In Travel domains like TripAdvisor, and Trivago. [4]

2.2 Recommendation Systems

Recommendation systems have become extremely common in recent years. It aids the customer in gathering knowledge and making decisions. By filtering individualized information based on the user's preferences from a massive volume of information, it is a software tool and approach that offer suggestions based on the customer's taste to discover new relevant things for them. Because they assist users in finding products they might not have otherwise found, recommender systems are a helpful alternative to search algorithms.

It's important to note that recommender systems are frequently built utilizing search engines that index unconventional data.

The First RS was created by Goldberg, Nichols, and Oki&Terry in 1992. A recommendation system is a solution to the problem of giving the consumer the right items aside from looking through many different options.

Although people's likes differ from one another, there are some patterns that they all share.

There are multiple types of recommendation systems.

2.2.1 Personalized Recommendation System

Personalization is the process of matching the appropriate users with the appropriate services, goods, or content. When done properly, it enhances user engagement, which refers to any interaction’s users have with a service, website, app, etc. There are 3 types of filtering when personalizing a recommendation system.

2.2.1.1 Content-Based Filtering

By identifying similarities, the content-based filtering (CBF) algorithm makes suggestions based on particular item features. These systems build data profiles based on description data, which may include user or item characteristics. The profiles that have been built are then utilized to suggest products that are comparable to those that the user has previously enjoyed, purchased, watched, or listened to.

2.2.1.2 Collaborative Filtering

Collaborative filtering (CF), the algorithm that is used the most frequently, generates suitable recommendations based on how various users interact with the target items.

These recommender systems collect data on prior user activity and mine it to determine which products to show to other active users who have similar likes.

This may include things like the music they've listened to, items they've added to their shopping cart, adverts they've clicked on, movies they've reviewed in the past, etc.

Such a system aims to anticipate how a person will respond to things they haven't yet engaged with.

2.2.1.3 Hybrid Filtering

To address the drawbacks and shortcomings of pure recommendation system models, hybrid filtering was developed.

The hybrid models combine numerous recommendation strategies under one roof to produce recommendations with a higher degree of accuracy and fewer drawbacks than any single one.

Collaborative filtering is typically combined with other techniques to address the cold start issue.

However, this is not a requirement since techniques can be integrated into various ways.

2.2.2 Recommender System Feedback Techniques

Information feedback is the fundamental component of a recommendation system since it provides the information the RS need to make appropriate recommendations to the customers based on their preferences. Generally speaking, there are three categories of feedback techniques.

2.2.2.1 Explicit Feedback Technique

The system must prompt users to rate products to gather explicit feedback from them. The system determines how relevant or similar an item is to users' choices after gathering feedback. Although this enables the recommender to understand the user's actual viewpoint, it is frequently difficult to gather because it involves direct user interaction.

2.2.2.2 Implicit Feedback Technique

Contrary to explicit feedback, gathering implicit input doesn't require user involvement. By observing activities taken by users, such as which things they viewed, where they clicked, what they bought, or how long they spent on a web page, the system automatically analyzes users' preferences.

2.2.2.3 Hybrid Feedback Technique

To improve prediction quality, hybrid feedback employs both explicit and implicit feedback. The system must be able to gather both explicit and implicit feedback from users to employ the hybrid method.

2.3 Mobile Application Development

A mobile application, more known simply as "an app," is a category of application software created specifically to run on mobile devices like smartphones and tablets. Similar services to those accessed on PCs are routinely made available to consumers through mobile applications. Mobile devices have grown in popularity in recent years thanks to their cheaper prices and compact, stylish design. Developers now face additional challenges as a result of the rise in popularity of mobile devices, including how to handle increases in bandwidth and how to adapt old codebases to run on a screen that is ten times smaller than it was intended to.

2.3.1 Java

A variety of apps can be created using the flexible and strong programming language Java. It can be used on any platform, including iOS and Android, because it is platform-independent. It's vital to understand that iPhones do not run the Java Virtual Machine due to iOS. Compiling the Java input to the Objective-C code allows you to get around this restriction, though.

2.3.2 Dart

Dart is a programming language that forms the basis of Flutter, it's created for client development, including web and mobile app development. It was created by Google and may be used to create desktop and server apps.

It is a class-based, garbage-collected, object-oriented language with C-style syntax. It supports type inference, interfaces, mixins, abstract classes, and reified generics, and can be compiled to either machine code or JavaScript.

2.3.3 Objective-C

A general-purpose, object-oriented programming language called Objective-C enhances the C programming language with Smalltalk-like messaging. Early in the 1980s, Brad Cox and Tom Love created the original version. Up to the release of the Swift programming language in 2014, Objective-C was the preferred programming language utilized, supported, and pushed by Apple for the creation of macOS and iOS applications.

2.3.4 Swift

Swift is a multi-paradigm, general-purpose programming language that was created by Apple Inc. and the open-source community. Swift was created as a replacement for Apple's previous programming language Objective-C, which had been mostly untouched since the early 1980s and lacked modern language capabilities. Swift was first released in 2014.

2.4 Related Work Overview

2.4.1 Google Maps

Navigation is the first thing that comes to mind when it comes to route planning, Google Maps is an indispensable part of our travels. Features such as adding your destination to the favorites on the application, and adding multiple stops to your route are our favorite aspects. You can also use Google Maps as an offline map. If there is no internet at your destination, you download the map of this place where there is the internet and you can use this map offline while you are there. [5] Google Maps is excellent, but there are no downsides. Sometimes he can take you off the beautiful route and take you to the mountain, to the forest. This can also allow you to encounter unusual things. At this point, it is suggested to look at the general route in general while saying the route is OK. And sometimes he tries to change the route itself, so don't accidentally choose and take a different route. Finally, you can look at the photos and comments of the places you will go on the application. Especially in the comments, you can find very useful things. You can also sort comments by date. Google Maps is the most important application to have on a phone. Apple and Yandex maps are also other alternatives. As stated, the application is good, but you should not completely give up control over the route, especially when it comes to the application.

When we created our project we were inspired by Google Maps. However, we wanted to design our application by developing Google maps in slightly different directions. That's why Google maps, is one of the applications that most fits our original logic. We will talk about the extra features we have added here; First, user comments, route determination, an indication of how long it will take to reach this route, and an indication of which vehicles can be used in our application, as well as in Google Maps. We have added interaction with platforms such as Facebook and Instagram to increase socialization to have socialization. Additionally, by selecting a specific range, the display of sights and historical monuments at a distance will again be selected. However, here Google maps only show all petrol offices, shopping malls, etc., including all the places in the region.

2.4.2 Travel Shop Turkey

Three alternatives are presented here, aiming for a straightforward cultural trip without confusing people too much. Firstly, tours are being organized to historical monuments located in various regions of Turkey (at a distance that is more well-known) and places where there is a lot of interest. Secondly, cabin trips are provided at the designated locations separately from the tours. Thirdly, it determines the routes that can be traveled by high-speed train in advance and then leaves it to the preference of passengers who want to attend these routes. [6] The difference from our project is; since we do not want a situation where people will be confused here, we thought of making an application so that they can easily make their preferences and not be undecided about issues such as where what is there or which places are more popular. Thus, after the application users have determined the desired range, they will be informed about many more places such as restaurants, historical places, and entertainment venues located within that range, and will also make recommendations from them, including the closest to the user's age and favorite things. Additionally, it can be linked to applications such as Facebook and Instagram to increase sociability. Finally, we have designed the application interface as convenient and understandable as possible for users to be satisfied with the application.

2.4.3 City Mapper

City Mapper is a must-have application for city transportation. The application, which has a very large database in which all vehicles used in public transportation are connected and synchronized, also cooperates with many other applications such as Google, Foursquare, and Uber. [7] It both takes you from point A to point B with the directions it offers and supports the processes from places to your means of transportation. You can find out your journey time by vehicle or how far you will walk, what alternatives there are by bus or metro on the route, and their fares. There are both android and iOS-supported versions. [8] In our project, it is progressing in the same process, only different, it can be thought of as a slightly higher model. While city mapper provides access only to certain entered points, many places are determined according to the range chosen in the application, depending on the person. Then it is ranked according to its popularity.

2.4.4 LiveTrekker

The purpose of the LiveTrekker application is to record the route you took during your trip, then show where you can go for how long on the interactive map and share it on social networks. You can also create a travel diary by uploading photos, notes, audio recordings, and video content to the application. The application we will develop is one of the most similar applications among the applications. The route is determined by travel time, similarly, it is used in an integrated way with platforms such as Instagram and Facebook in case you want to share it on social networks. In the application mentioned here, many features such as photos, notes, voice recordings, etc. can be uploaded inside the application.

2.4.5 Time Out

Time Out, an application that lists restaurants, cafes, and entertainment venues located on the route you designate, also calculates the distance by showing the address information of the listed places on the map. In addition, you can easily make plans with the Time Out application, which also shows current cultural, art, and music events. This application is one of the closest applications to the application we will make. However, unlike our application, it can interact with platforms such as Instagram and Facebook to ensure sociability. However, although determining places via a specific route is similar to its features, such as showing the popularity of these places, addition, our application also determines historical places.

2.4.6 Zomato

In a city where you are going for the first time, you don't know where to eat and how much to pay. Zomato, which comes to the rescue of users in this regard, saves you from a big burden by listing the businesses closest to your location, their concepts, menus, and other user experiences. Which also offers the ability to make reservations, and stands out as a very useful application. On the other hand, we see that this application answers more questions such as what is on the menu and where we should I eat, but our application it is a little different in this regard. There is no routing based on the location of the user in this application. And also, Zomato doesn’t integrate with any type of social networking application.

2.5 Summary

Our goal in this project is to plan and develop the popular recommendation that systems examine our positive aspects are included in the practice, while other applications on the negative aspects and think about how we can fix it based on user feedback, and develop software for developing the negative aspects. Specially to provide personalization, unlike other recommendation systems.

3 SOFTWARE REQUIREMENT SPECIFICATION(SRS)

3.1 Introduction

A recommendation system for travelers is a mobile application that learns the location of the user, is located in the area between the distance values entered by the user, shows and informs the location of the place or places that the travelers want to go, and if desired, guides the user to the place accepted in the system. This system includes Live Location Tracking and Navigator. Additionally, the system uses Google API.

3.1.1 Purpose

The purpose of the document is to give a detailed description of the requirements for the "Recommendation System for Travelers" project. This application aims that users who will use the application called "Travelers" can easily find the places they can visit, share and evaluate from the people who are interested in traveling. This document describes the requirement of the project in detail.

3.1.2 Scope of Project

"Recommendation system for Travelers" is a GPS-Based mobile application that helps users to find and discover new historical places or natural areas based on the user's current GPS location and recommending places to go. There are other features such as the application drawing a route for the specified location and users can upload and share photos to the system. This information will form the basis for the recommended results displayed to the user. The administrator also uses this application to manage the system and limit inappropriate or prohibited posts by the social platform and delete users from the platform when necessary. In addition, if there is a malfunction in the application structure, the administrator can directly handle and manage the situation. System information is stored in a web-based data store. System information is stored in a web-based data store. The stored data is filtered and transferred to the user interface and used. In addition, the GPS of the device used by the user must be turned on for the user's instant location to be known by the system. While the GPS shows the instant location, the system suggests and shows the details of the nearby areas by taking the instant location data from the user according to the radius range determined by the user.

3.1.3 Glossary

- API –API stands for Application Programming Interface.
- Geolocation – the process or technique of identifying the geographical location of a person or device using digital information processed via the internet.
- User-A person who uses or operates something.
- Flutter-Flutter is an Open-Source UI SDK developed by Google.
- IOS- A mobile operating system created by Apple.
- Android- A mobile operating system created by America; Google.
- Dart-Dart is an Open-Source, client-side programming language.

3.2 Overall Description

3.2.1 Product Perspective

Recommendation System for Travelers project that has the purpose of recommending places to visit such as historical places, natural environments, etc. This section will give an overview of the system. The system is made to specify and organize the details to be done in the project.

3.2.1.1 Development Methodology

For developing the project, we have planned to use the Waterfall software development methodology, as known as the Waterfall model. The waterfall model uses a logical progression, similar to the direction water flows over the edge of a cliff. We planned to divide the project into equal parts in our group and to unite it after everyone did the task given in their field according to the determined requirements.

3.2.2 User Characteristics

There are two types of users that interact with the system: admin and user.

3.2.2.1 Admin

Admin should know the system well and be able to intervene when necessary. • Admin must know how to use a computer and phone. • Admins must have internet access and any mobile device to use the application.

3.2.2.2 User

Users must have internet access and any mobile device to use the application. • The user must activate GPS for giving the system to the current location.

3.3 Requirements Specification

3.3.1 External Interface Requirements

3.3.1.1 User Interface

The user interface must be easy-to-use and clear, because the app's target users are general users, and for that reason, the possible errors should be minimized. In case of an error, error messages should be clear and understandable. The user interface should be designed in such a way that user can clearly show their current location on the map and the places they want to go or are recommended by the app. The target user of this app are people who are indecisive about where to see when traveling.

Flexibility and efficiency, Aesthetic and minimalist design, Help and documentation, as well as a design for the user to better use the given design, are being applied.

3.3.1.2 Hardware Interfaces

The project is a mobile-based application. The client-side application will be developed as an Android and IOS application. To use the application user must have an Android or IOS-based operating system device. Also, if the user wants to add a photo of the places they visited, a camera may be needed.

3.3.1.3 Software Interfaces

To develop the application, Flutter is the main programming language for the app because the application will support Android and IOS. To develop this application, Google's own database Google API will be used. Also, we are using GPS for location services.

3.3.1.4 Communications Interfaces

In order to send the user's instant location to the system, the user's internet and GPS system must be enabled.

3.3.2 Functional Requirements

3.3.2.1 Open Page

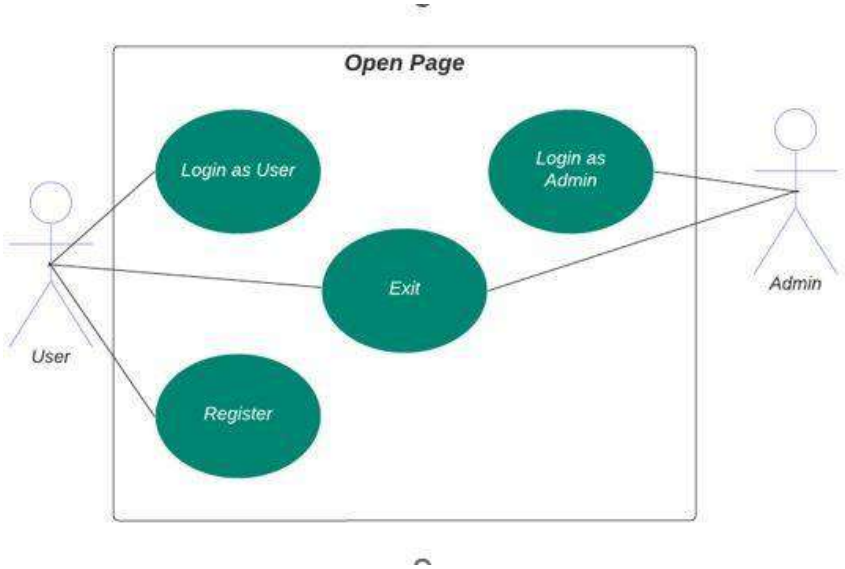
• Register

• Login as a User

• Login as Admin

• Exit

Use Case Diagram:



Use Case Name: Open Page

Use Case Number: UC-1

Actors: User, Admin

Overview: Users will use this use case to log in to the page.

Pre-Conditions:

- An Internet connection is required to open the page.

Post-Conditions:

- Users must log in to use the app. Flow Description:
- The user clicks the “Login” button to register to the application.
- When the user clicks the “Login” button, they are redirected to the login screen.

Alternative Flow Description:

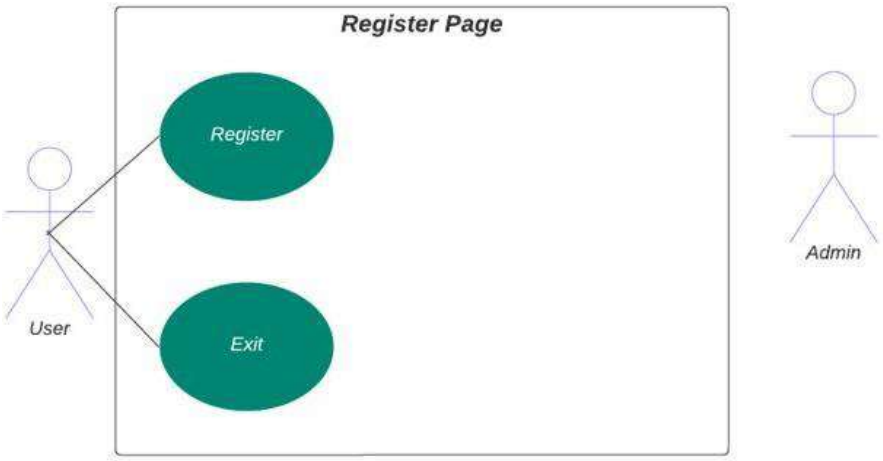
- If the person entering the application is “Admin”, then click the “Login as Admin” button.
- If the person entering the application is a “User”, they click the “Login as User” button.

3.3.2.2 Register Page

• Register

• Exit

Use Case Diagram:



Use Case Name: Register Page

Use Case Number: UC – 2

Actors: User

Overview: In this use case, users register to the system.

Related Use Cases: UC – 1

Pre-Conditions:

- Users must log in to the register page.
- Users should not be registered in the system before.

Post-Conditions:

- When users fill in the required information correctly, they are successfully registered in the system.

Flow Description:

- The system expects users to enter their “Username”, “Email” and “Password” correctly.
- The system checks whether the users have this information in the database.
- If there is an error, it is indicated on the screen. Re-enters user information.

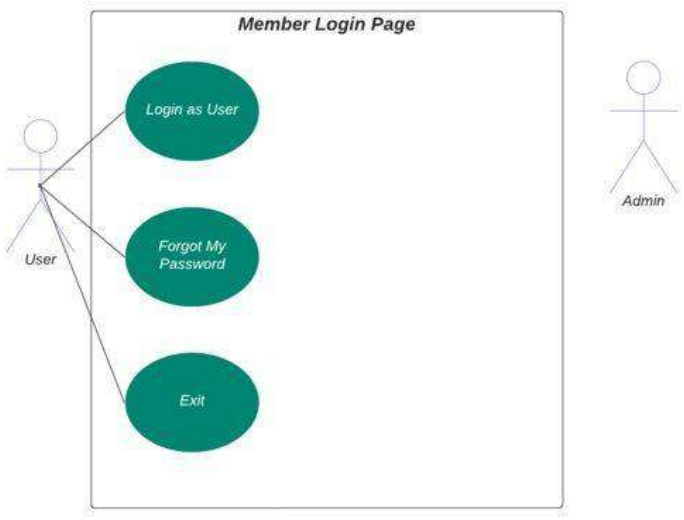
3.3.2.3 Member Login Page

• Login

• Forgot My Password

• Exit

Use Case Diagram:



Use Case Name: Member Login Page

Use Case Number: UC – 3

Actors: Users

Overview: In this use case, users log into the system.

Related Use Cases: UC – 1

Pre-Conditions:

- Users must enter user information correctly.

Post-Conditions:

- If the user has entered the correct “Username” and “Password”, she/he logs into the system.

Flow Description:

- Users fill in the username and password section on the screen after pressing the login button.
- The system checks the information entered by the users.
- If there is an error, it is indicated on the screen. Re-enters user information.

Alternative Flow Description:

- If the user has forgotten his password, click the “Forgot My Password” button.
- The system prompts the user to enter the e-mail registered to the system.

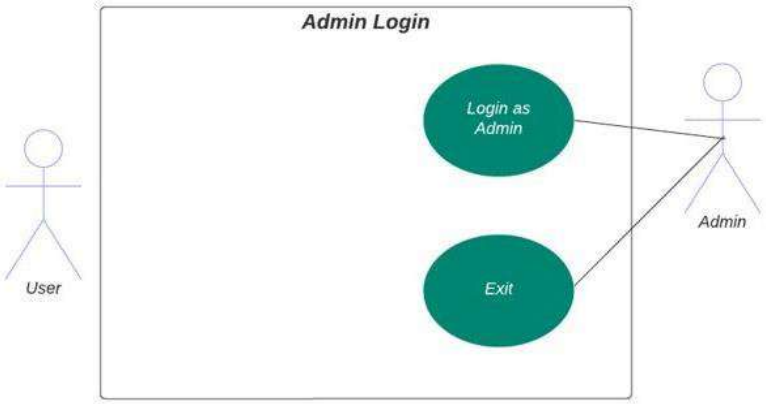
- The system sends a password change mail to the user's e-mail.
- The user can reset his password in line with the incoming mail.

3.3.2.4 Admin Login

• Login as Admin

• Exit

Use Case Diagram:



Use Case Name: Admin Login

Use Case Number: UC-4

Actors: Admin, System

Overview: In this use case, admins log into the system.

Related Use Cases: UC-1

Pre-Conditions:

- Admins must have an internet connection.

Post-Conditions:

- If the admin information is entered correctly, it logs into the admin page.

Flow Description:

- Admin must press "Login as Admin" on the login screen
- Admin enters "Username" and "Password" correctly.
- If there is an error, it is indicated on the screen. Re-enters admin information.

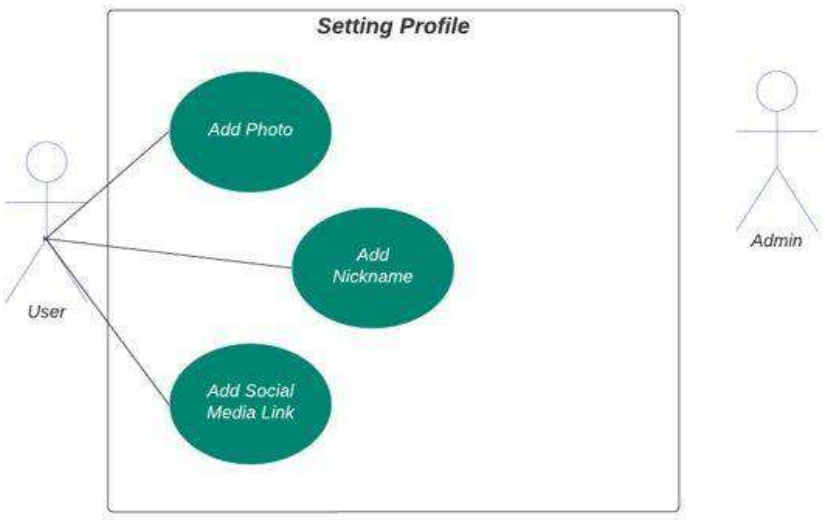
3.3.2.5 Setting Profile

• Add Photo

• Add Nickname

• Add Social Media Link

Use Case Diagram:



Use Case Name: Setting Profile

Use Case Number: UC-5

Actors: Users

Overview: In this use case, users are redirected to the “Settings Profile” page, where users can share their photos, add a nickname for themselves and share links to other social media platforms.

Related Use Case: UC-1, UC-3

Pre-Conditions:

- Users must be registered in the system.

Post-Conditions:

- When the users click the setting profile button, they are directed to the setting profile page.

Flow Description:

- User presses the setting profile button.

- If the user wants to add a photo, he/she clicks the “add photo” button.
- If the user wants to change their username, they click the “change username” button.
- When the users make changes to their profile or add something new, they should press the save button.

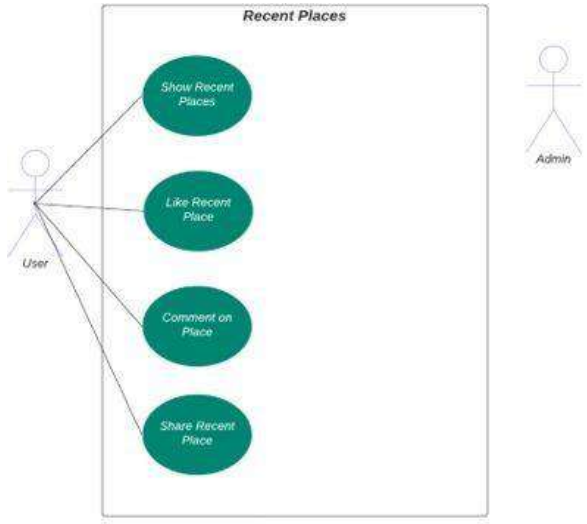
Alternative Flow Description

- If the user does not press the “save” button on the settings they have changed, the system will continue to use the user’s default settings.

3.3.2.6 Recent Places

• Showing Recent Places

Use Case Diagram:



Use Case Name: Recent Places

Use Case Number: UC-6

Actors: Users, System

Overview: In this use case, users can see the recent places visited by other users or themselves.

Related Use Cases: UC-1, UC-3

Pre-Conditions:

- Users must be registered in the system.
- Users must have an internet connection.

- Users must have the GPS turned on, on their phones.

Post-Conditions:

- When the users click “Recent Places”, they can see the last places of the user.

Flow Description:

- The user should enter their own or another user’s profile and click the “Recent Places” button.
- The user can like the recent places he/she has been to by pressing the like button under the photo or location.
- The user can write his/her comment in the comment box about this place and share it with the share button.
- If users want, they can stop sharing with the close “Share Recent Places” button in the profile settings section.

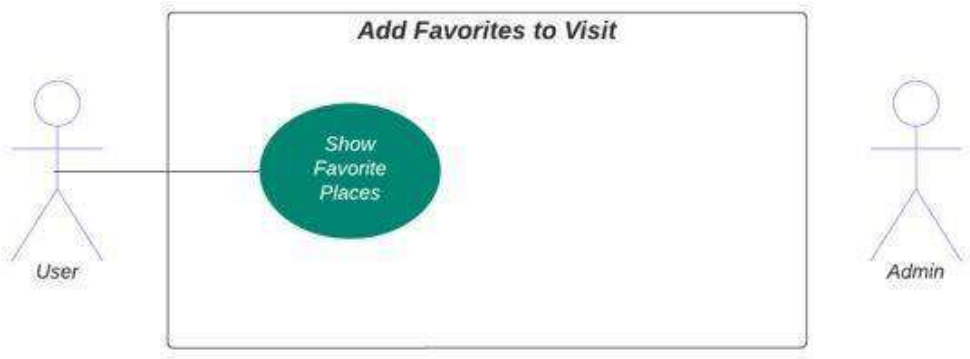
Alternative Flow Description:

- Users cannot see the last visited GPS feature and the internet connection on their phones is not turned on.

3.3.2.7 Add Favorites To Visit

• Adding Places to Visit

Use Case Diagram:



Use Case Name: Add Favorites To Visit

Use Case Number: UC-7

Actors: Users

Overview: In this use case, users can create collages and share them on the “Add Favorites to Visit” page.

Related Use Cases: UC-1, UC-3

Pre-Conditions:

- Users must be registered in the system.
- Users must have an internet connection.
- Users must have the GPS turned on, on their phones.

Post-Conditions:

- When the users click “My Favorite Places”, they can see the added favorite places.

Flow Description:

- Users should click on the “Recent Places” button.
- Users should click on the heart icon above the place they like.
- When the user clicks on the heart symbol, the system automatically creates a “My Favorites Places” folder in the user’s profile.

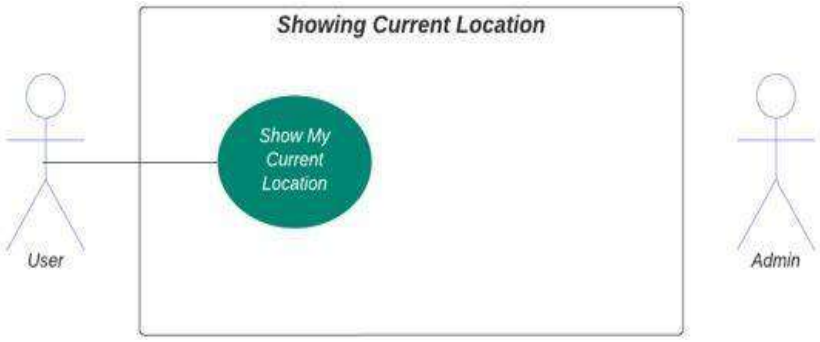
Alternative Flow Description:

- Users cannot see the last visited GPS feature and the internet connection on their phones is not turned on.

3.3.2.8 Showing Current Location

• Current Location

Use Case Diagram:



Use Case Name: Showing Current Location

Use Case Number: UC-8

Actors: Users, System

Overview: In this use case, after logging into the application, users can find their location on the map on the main screen with the GPS feature on their phones.

Related Use Cases: UC-1, UC-3

Pre-Conditions:

- Users must be registered in the system.
- Users must have an internet connection.
- Users must have the GPS turned on, on their phones.

Post-Conditions:

- When the user clicks on the “show my current location” button on the map, it shows the current location and its surroundings.

Flow Description:

- After the user logs into the application, the map appears on the main screen.
- The users press the “show my location” button on the map to find their location on the map.
- After pressing the button, the users can see their location and surroundings on the map.

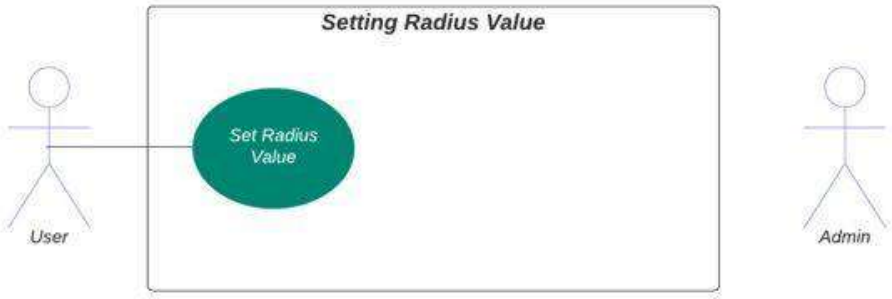
Alternative Flow Description:

- If the user does not have an internet connection, they cannot see their location on the map.
- If the GPS feature of the users is not turned on, they cannot see their location.

3.3.2.9 Setting Radius Value

• Setting value

Use Case Diagram:



Use Case Name: Setting Radius Value

Use Case Number: UC-9

Actors: Users, System

Overview: In this use case, the user sets how many kilometers in diameter they want to see places around them.

Related Use Cases: UC-1, UC-3, UC-8

Pre-Conditions:

- Users must be registered in the system.
- Users must have an internet connection.
- Users must have the GPS turned on, on their phones.

Post-Conditions:

- When the user clicks on the “Set Radius Value” button on the map, they enter the desired value, and the surrounding places are shown according to the entered value.

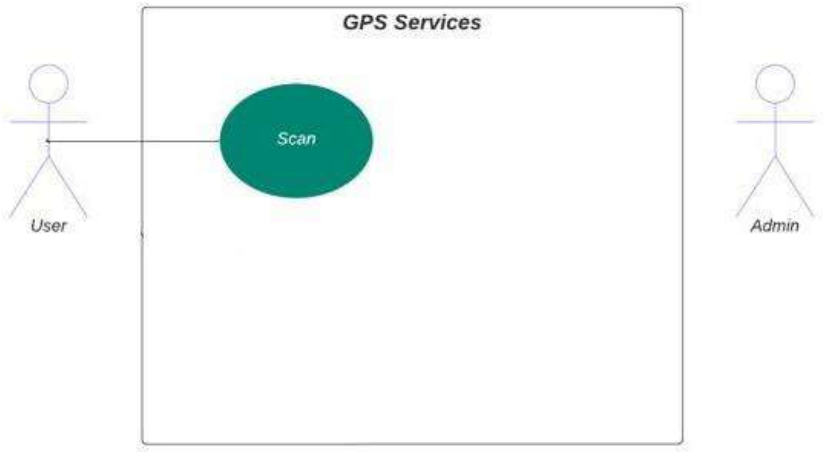
Flow Description:

- After the user logs into the application, the map appears on the main screen. -The user's current location is shown on the map. -The user chooses settings and “Set Radius Value”, and enters the desired Radius value.
- The application shows the desired area. Alternative Flow Description: -If there is no place to show within the selected radius, a warning message is shown to the user to select a wider radius.

3.3.2.10 GPS Services

• Scanning Places

Use Case Diagram:



Use Case Name: GPS Services

Use Case Number: UC-10

Actors: Admin, Users, System, Google Maps

Overview: In this use case, after the user opens the application and logs in, the GPS service activation warning message appears on the screen of the system.

Related Use Cases: UC-1, UC-3, UC-4, UC-6, UC-7, UC-8

Pre-Conditions:

- Users must be registered in the system.
- Users must have an internet connection.
- Google Maps should be accessible.

Post-Conditions:

- The system prompts users to turn on GPS services to use the app.

Flow Description:

- User clicks Scan.
- The system asks the user where she/he wants to go.
- After the user chooses the place to go, the system draws the shortest distance to be traveled and can see the nearby recommended places with the GPS services.

Alternative Flow Description:

- If the user's GPS service is not turned on, the system cannot create a route to the destination and cannot scan for nearby places.

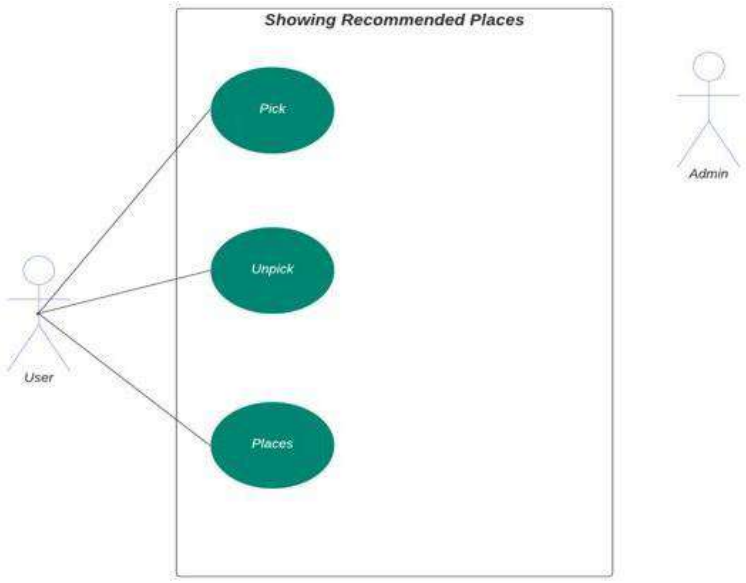
3.3.2.11 Showing Recommended Places

• Pick

• Unpick

• Places

Use Case Diagram:



Use Case Name: Showing Recommended Places

Use Case Number: UC-11

Actors: Users, System

Overview: In this use case, the system recommends places to users according to the places they visit.

Related Use Cases: UC-1, UC-3, UC-10, UC-9, UC-8, UC-6

Pre-Conditions:

- Users must be registered in the system
- Users must have an internet connection.
- Google Maps should be accessible.

- The user must find the places they visit through the application.

Post-Conditions:

- The system of the application learns by machine learning by keeping the places visited by the user in its database.
- The system sends warning messages for the location of similar places on the map used by the application, according to the type of places the user visits the most.

Flow Description:

- After the user logs into the application, the map appears on the login screen.
- While the user is traveling, the application sends the location of the places s/he may like to the user in the form of a warning message as a pop-up.
- If the user accepts to go to this place suggested by the system, s/he should click the "Yes" button in the warning message that appears.
- After the user clicks the "Yes" button, the shortest route that the user can go to this recommended place is created.

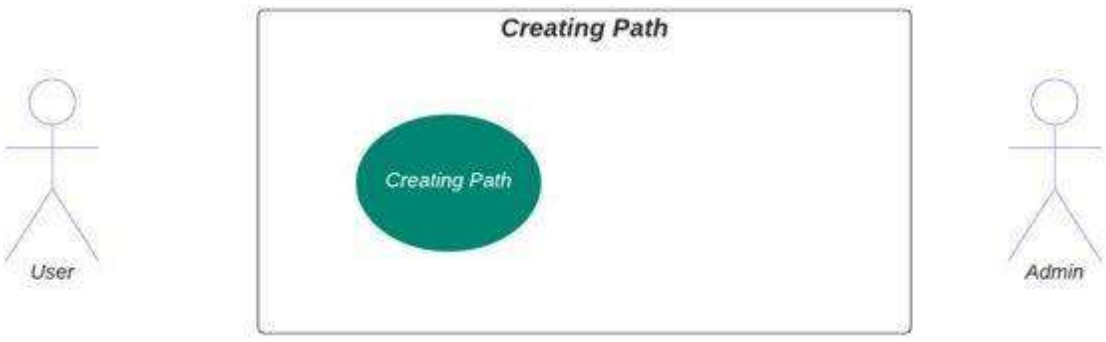
Alternative Flow Description:

- If the GPS service and internet connection on the user's phone is not turned on, the system cannot recommend places to visit

3.3.2.12 Creating Path

• Creating path

Use Case Diagram:



Use Case Name: Creating Path

Use Case Number: UC-12

Actors: System

Overview: In this use case, the system creates the route of the place where the user wants to go on the map used by the application in the shortest way.

Related Use Cases: UC-1, UC-3, UC-6, UC-8, UC-9, UC-10

Pre-Conditions:

- Users must be registered in the system
- Users must have an internet connection.
- Google Maps should be accessible.
- GPS service must be turned on.

Post-Conditions:

- The user should write the name of the place at the bottom of the map where s/he wants to go on the map used by the application and click the “Create Path” button.

Flow Description:

- After logging into the application, the user should open the slider at the bottom of the map to choose the place s/he wants to go.
- When it's a slider, the user should write the location of the user and the place s/he wants to go into the spaces.
- When the locations are entered correctly the “Create Path” button should be clicked.
- After pressing the button, the system creates the shortest route from the user’s current location to the location they want to go and displays it on the map.
- When the user reaches the place s/he wants to go, the message that s/he has reached that place.

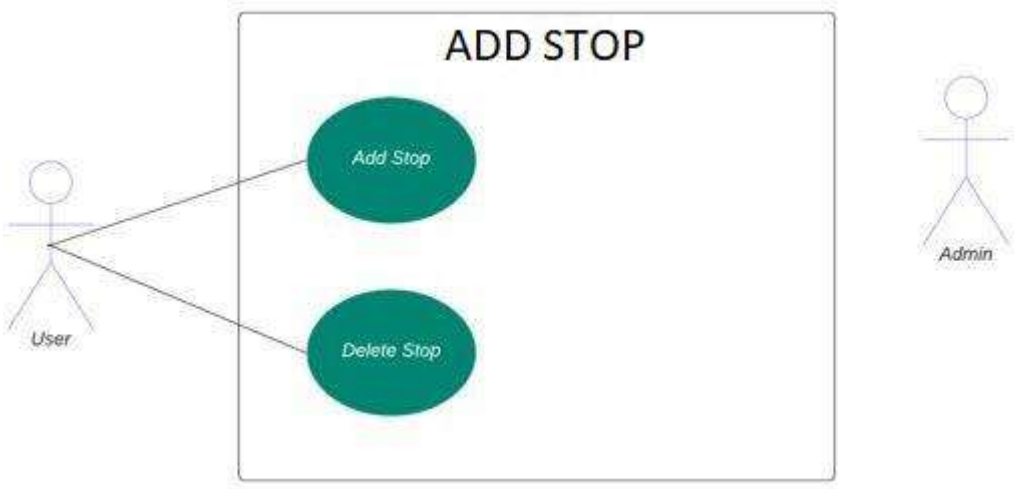
Alternative Flow Description:

- If the user does not have an internet connection, the route of the place they want to go cannot be created.
- If the GPS service is not turned on, it cannot see the location of the destination and the location of the user.

3.3.2.13 Add Stop

• Adding wanted stop location

Use Case Diagram:



Use Case Name: Add Stop

Use Case Number: UC-13

Actors: Users

Overview: In this use case, the user can add stop points on the creating path s/he has created by pressing and holding the selected locations with the “Add stop point” button.

Related Use Cases: UC-1, UC-3, UC-6, UC-8, UC-9, UC-10, UC-12

Pre-Conditions:

- Users must be registered in the system
- Users must have an internet connection.
- Google Maps should be accessible.
- GPS service must be turned on.
- Users must create a path to go.

Post-Conditions:

- After the user creates the path to go, if there are places on the path that s/he wants to go, s/he should press the “add stop” button.

Flow Description:

- The user should click on the “Create Path” button using the slider.

- If there is another place that the user wants to go on the path created after creating the paths/he should click on the “add stop” button after pressing and holding on to the place where these locations are on the map.
- After the stops are added, these points appear as a red dot on the map.
- After stops can be deleted by pressing the “delete stop” button.

Alternative Flow Description:

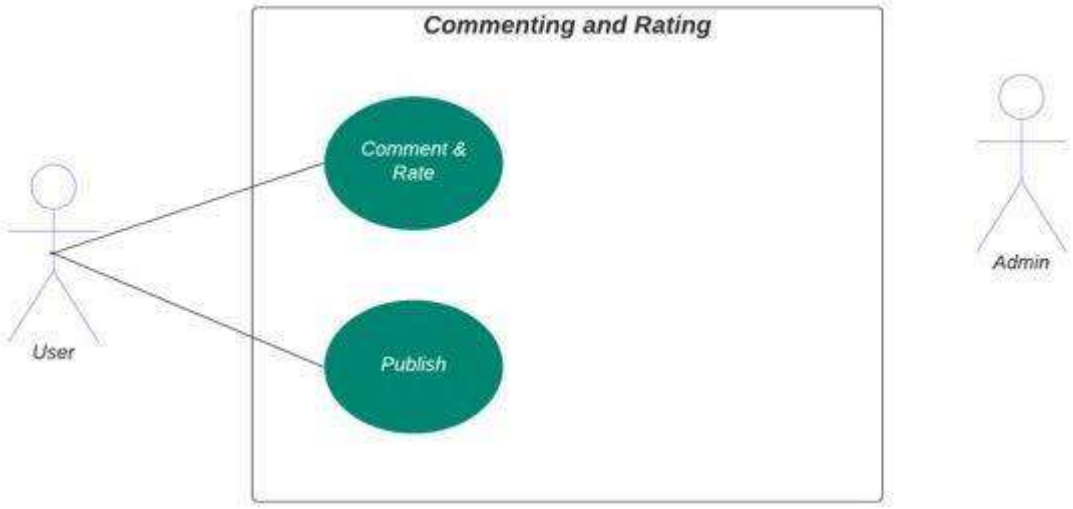
- If the user has not created a path, s/he cannot add stops.

3.3.2.14 Commenting and Rating

• Comment & Rate

• Publish

Use Case Diagram:



Use Case Name: Commenting and Rating

Use Case Number: UC-14

Actors: Users, System

Overview: In this use case, users can comment and rate the places they go and publish these comments, and rate with other users.

Related Use Cases: UC- 1, UC-3

Pre-Conditions:

- Users must be registered in the system

- Users must have an internet connection.
- GPS service must be turned on.
- Users must have gone to the place to comment and rating.

Post-Conditions:

- If was in location update database, refresh is called and exit activity.
- If it wasn't in location send an error message and exit activity.

Flow Description:

- If the user wants to comment and rate the place s/he went, s/he should press the "Comment & Rate" button.
- After pressing the button, the user can write a comment in the box that appears.
- The user can rate the place one to five stars.
- When the user presses the "Publish" button, their comment and rating are published so that everyone can see them.

Alternative Flow Description:

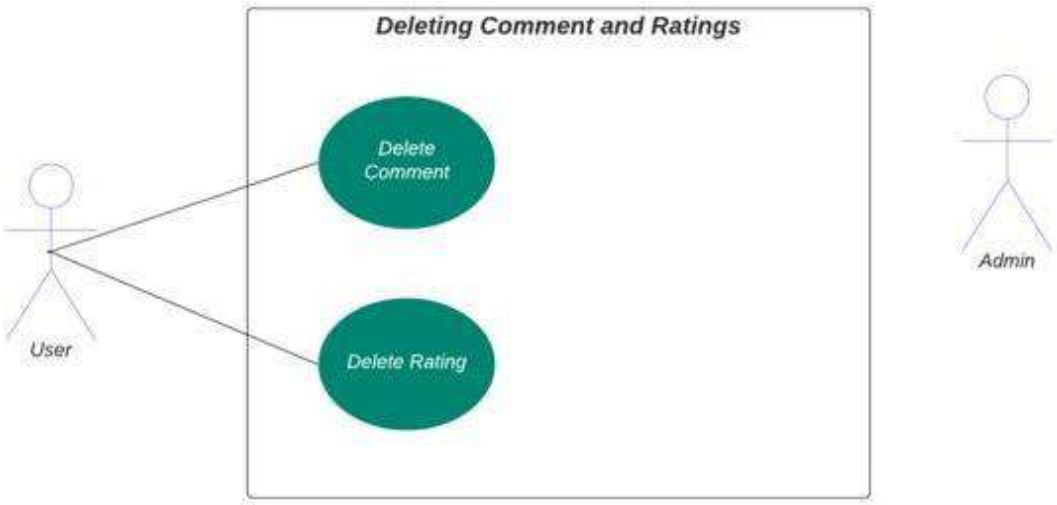
- If the user has not visited that place before, they cannot make comments or give ratings.

3.3.2.15 Deleting Comments and Ratings

• Delete Comment

• Delete Rating

Use Case Diagram:



Use Case Name: Deleting Comments and Ratings

Use Case Number: UC-15

Actors: User, System

Overview: In this use case, users can delete the comments they have made before or the ratings they have given.

Related Use Cases: UC-1, UC-3, UC-14

Pre-Conditions:

- The user must be registered in the system.
- User must have an internet connection
- The user must have previously commented on that place or rated that place.

Post-Conditions:

- When the user presses the “Delete Comment” or “Delete Rating” button, the selected content will be deleted.

Flow Description:

- User clicks the "Delete Comment" or "Delete Rating" button.
- A message is shown to the user asking, "Are you sure you want to delete the content".
- The selected content will be deleted if the user confirms their choice.

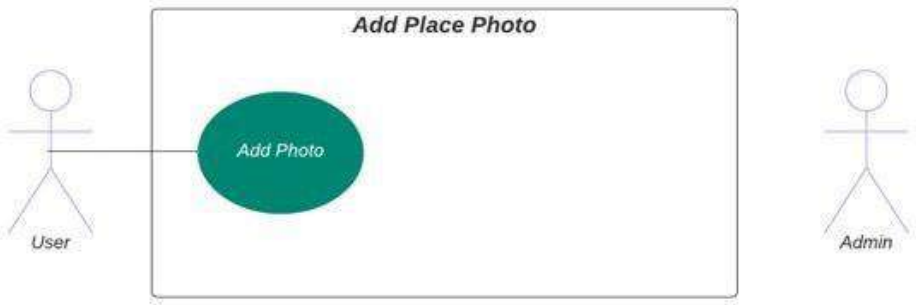
Alternative Flow Description:

- If the user's comment has been deleted by the admin, a message should be sent to the user stating this.

3.3.2.16 Add Place Photo

• Add Place Photo

Use Case Diagram:



Use Case Name: Add Place Photo

Use Case Number: UC-16

Actors: User

Overview: In this use case, the user adds a photo to the location they have visited.

Related Use Cases: UC-1, UC-3, UC-6

Pre-Conditions:

- The user must be logged in to the system.
- The user must have an internet connection.
- The user must allow camera access.
- The user must have visited that place before adding a photo.

Post-Conditions:

- When the user clicks on the “Add Photo” button, the user can add the photo they have taken.
- If the user clicks “Add Photo”, the selected photo will be uploaded and can be seen on the app.

Flow Description:

- User clicks the "Add New Photo" button.
- User either picks the photo from their gallery or they can take the Picture from their camera.
- User uploads the photo to the system.

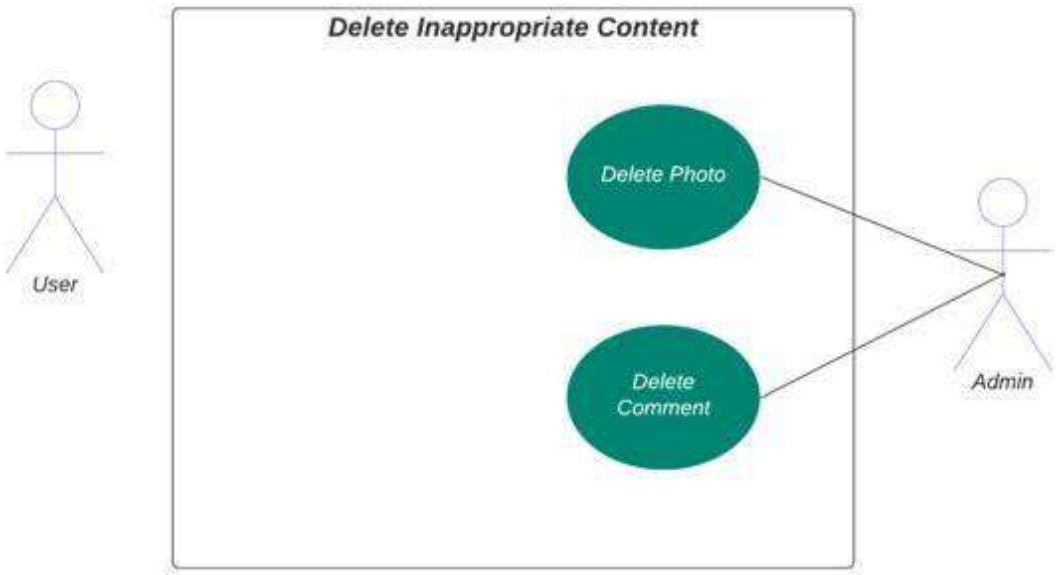
Alternative Flow Description:

- If the photo cannot be uploaded due to internet connection problems or the resolution of the photo an error message will appear on the screen.

3.3.2.17 Delete Inappropriate Content

• Deleting inappropriate content

Use Case Diagram:



Use Case Name: Delete Inappropriate Content

Use Case Number: UC-17

Actors: Admin

Overview: In this use case, the admin can delete inappropriate content that may disturb other users.

Related Use Cases: UC-4

Pre-Conditions:

- Admin must be logged into the system.
- Admin must have an internet connection.
- When the admin clicks on the "Delete Comment" or "Delete Photo" button, the chosen content will be deleted.

Post-Conditions:

- If the admin clicks the button, the selected comment or photo will be deleted, and the user will be added to the blacklist.

Flow Description:

- Admin clicks "Delete Comment/Photo"
- The selected comment or photo will be deleted from the system.
- The user will be added to the blacklist.

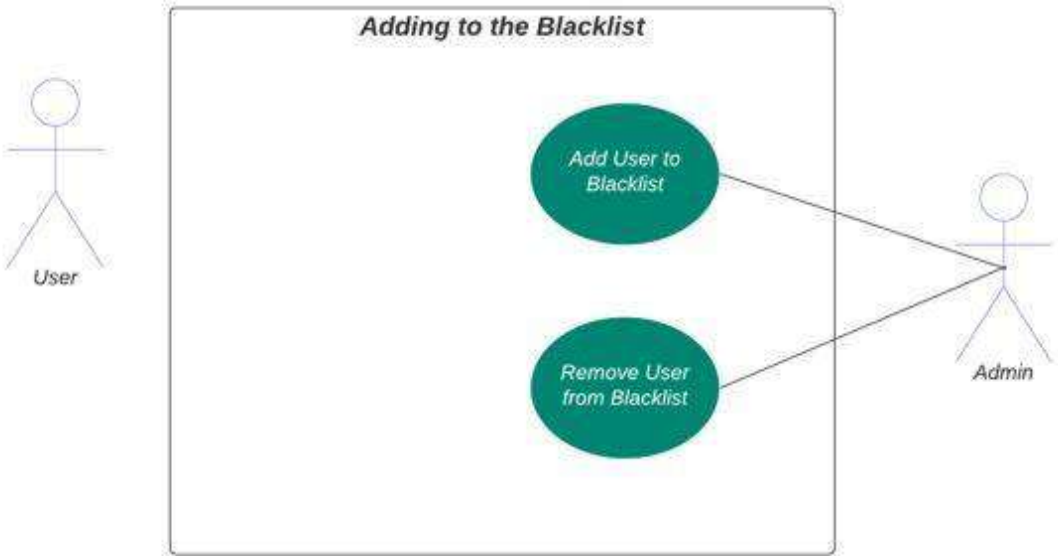
Alternative Flow Description:

- If the user has already been blacklisted, the user's account will be deleted.

3.3.2.18 Adding to the Blacklist

• Add the user to the Blacklist

Use Case Diagram:



Use Case Name: Adding to the Blacklist

Use Case Number: UC-18

Actors: Admin, System

Overview: In this case, Admin can delete users if the user makes an inappropriate action on the system.

Related Use Cases: UC-4, UC-17

Pre-Conditions:

- Admin must be logged into the system.

- Admin must have an internet connection.
- The admin must have previously deleted the user's inappropriate content, or the user must have been reported to the admin.

Post-Conditions:

- If the admin clicks the "Blacklist User" button, the user's account will be suspended.

Flow Description:

- Admin clicks the “Blacklist User” button.
- The admin specifies why the user's account wants to be blacklisted.
- The user's account is suspended.

Alternative Flow Description:

- If the user's account has already been suspended, the admin will be shown an error message indicating this.

3.3.3 Software System Attributes

3.3.3.1 Portability

The system will be a mobile application, it will be able to be used on all mobile devices.

3.3.3.2 Usability

This application is user-friendly.

3.3.3.3 Reliability

Reliability is one of the key attributes of the system. Back-ups will be made regularly so that restoration with minimal data loss is possible in the event of unforeseen events. The system will also be thoroughly tested by all team members to ensure reliability.

3.3.3.4 Availability

This application will be available on Google Play Store and also AppStore.

3.3.3.5 Security

The system shall be designed with a level of security appropriate for the sensitivity of information enclosed in the database. More interaction is needed with the client about the volatility of the information. The only requirements that could be implemented are encrypting the database and/or making the database password-protected, by the user’s request.

4 SOFTWARE DESIGN DOCUMENTATION(SDD)

4.1 Introduction

A recommendation system for travelers is a mobile application that learns the location of the user, is located in the area between the distance values entered by the user, shows and informs the location of the place or places that the travelers want to go, and if desired, guides the user to the place accepted in the system. This system includes Live Location Tracking and Navigator.

4.1.1 Purpose

The purpose of this document is to determine the requirements of the project to be carried out, to determine and explain the internal and external design used in it, and to define and design the layout of the system before the project has been completed.

4.1.2 Overview

In the introduction, information about the purpose of use of our product, what it contains, and its design is given. In the Approach section, the approach we will use while designing the product is emphasized. In the System Design section, the current class diagram, sequence diagram, and flowcharts of the product are designed.

4.1.3 Definitions and Acronyms Abbreviations

- Android: A mobile operating system based on a modified version of the Linux kernel and other open-source software.
- IOS: A mobile operating system created and developed by Apple.
- User: Any user character that is yet to register to the system.
- Administrator: The user character that supervised the system and is a registered administration user.
- Firebase: Provides detailed documentation and cross-platform SDKs to help you build and ship apps on Android and iOS.
- GPS: The Global Positioning System is a satellite-based radio navigation system owned by the US government.
- Geolocation: The process or technique of identifying the geographical location of a person or device by means of digital information processed via the internet.
- Flutter: Mobile app development platform created by Google.
- Dart: Client-optimized language for developing fast apps on any platform.

- UI: UI design stands for "user interface".
- UX: UX design stands for "user experience".

4.2 Architecture Design

4.2.1 Approach

'Recommendation system for travelers'; It is a GPS-based mobile application that helps the user find and explore new historical places or natural areas and suggest places to go based on their current GPS location. It also has other features such as the application that draws a route for the determined location and the ability for users to upload and share photos to the system. This information will form the basis of the recommended results displayed to the user. The administrator also uses this application to manage the system and limit inappropriate or prohibited posts by the social platform and delete users from the platform when necessary. In addition, the administrator can add and delete places and update places' information. In addition, in case of a malfunction in the application structure, the administrator can directly handle and manage the situation. System information is stored in a cloud-based data warehouse. The stored data is filtered and transferred to the user interface and used. In addition, as a result of logging into the system, it is possible for the user to write comments and evaluate the places he/she has actually visited. In addition, the GPS of the device used by the user must be turned on in order for the user's instant location to be known by the system. While the GPS shows the instant location, the system suggests and shows the details of the nearby areas by taking the instant location data from the user according to the radius range determined by the user.

Briefly:

• Admin can view/modify/delete User, upload places information.

• Admin can view all feedback posted by customers and identify negative comments and users to be restricted.

• The user can log in to the system after entering the registration information.

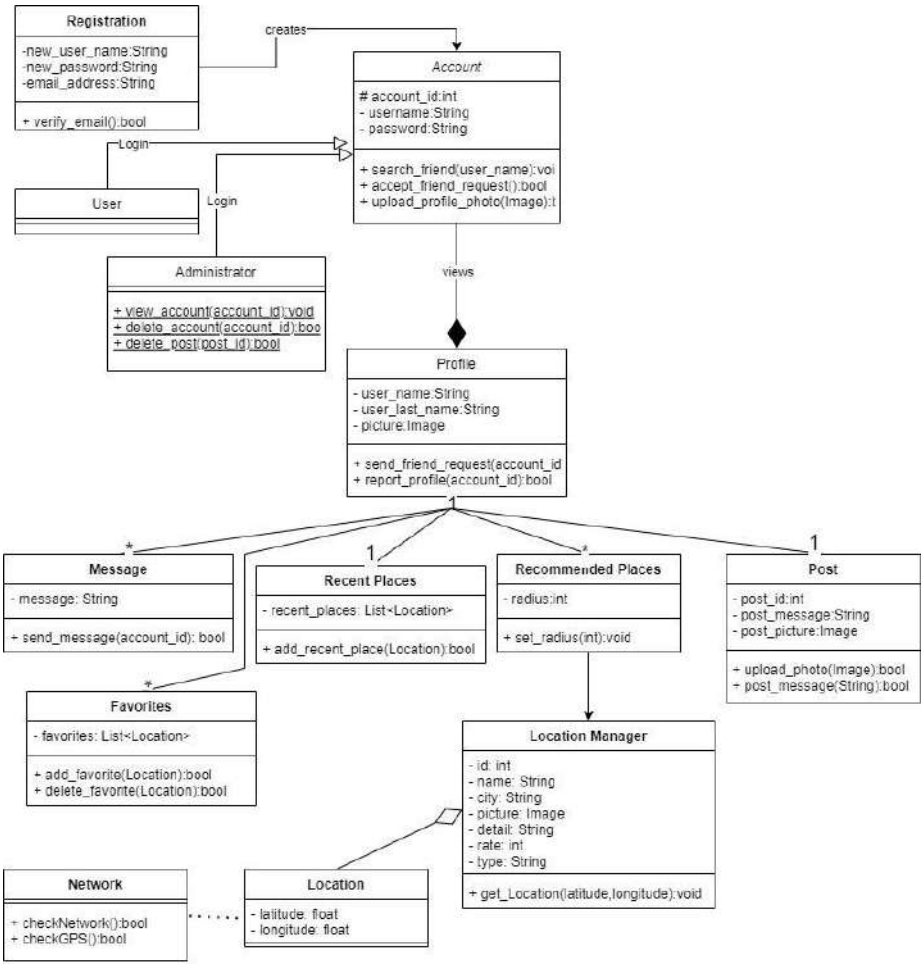
• Users can search places by rating and place type.

• When the user is going somewhere, GPS must be turned on.

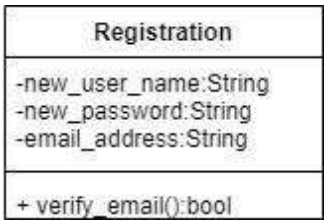
• The user can evaluate and write a comment about that place according to where he/she went.

• The user can post comments and reviews about a particular place. It can also provide photo sharing related to that place.

4.2.2 Class Diagram



4.2.2.1 Registration



Fields:

new_user_name: This is the name of the user.

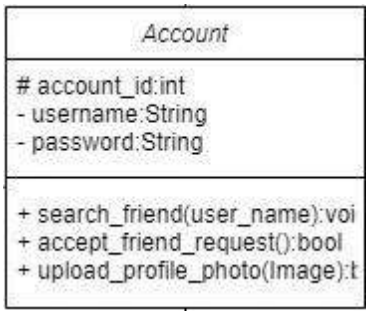
new_password: This is the password of the user.

email_address: This is the email of the user.

Functions:

verify_email(): This function checks the connection sent by email from the system and provides the transition.

4.2.2.2 Account



Fields:

account_id: The field that holds the private number of everyone randomly assigned by the system.

username: The field is an actual email address for login to the system.

password: The field where the user specifies his/her password.

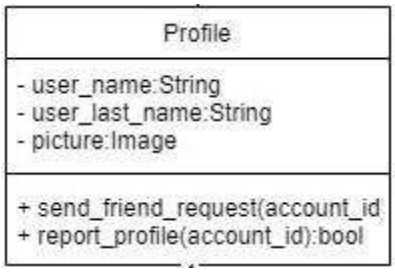
Functions:

search_friend(): Member can search for friends from the system and can send requests.

accept_friend_request(): Member can accept the sent request.

upload_profile_photo(): Member can upload a photo to the system.

4.2.2.3 Profile



Fields:

user_name: where the user determines their name in the profile.

user_last_name: where the user determines their last name in the profile.

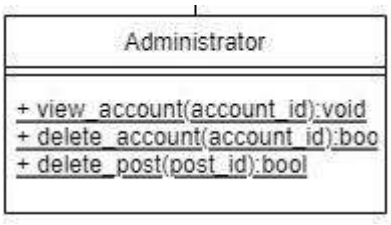
picture: where the user's profile picture is determined.

Functions:

send_friend_request(): A function where a user can send requests to other users.

report_profile(): Function where the user can report other users to the system.

4.2.2.4 Administrator



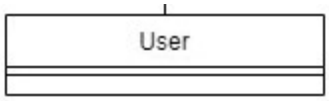
Functions:

view_account(): Admin can see the accounts in the system.

delete_account(): Admin can delete an account from the system when it must.

delete_post(): Admin can delete inappropriate posts that came from any user.

4.2.2.5 User



4.2.2.6 Favorites

Fields:

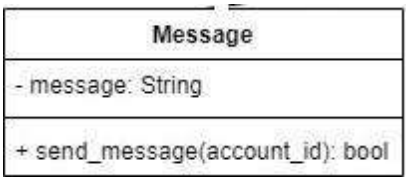
favorites: The field where the user pre-determines the places they want to go and adds them.

Functions:

add_favorites(): The function where the user pre-determines the places they want to go and adds them to the list.

delete_favorites(): A function that allows the user to pre-determine the places they want to go and to extract the places they add to the list.

4.2.2.7 Message



Fields:

message: chat of the user with other users.

Functions:

send_message(): function that allows the user to chat with other users.

4.2.2.8 Recent Places

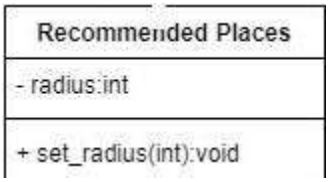
Fields:

recent_places: a list where the user has added previously visited places.

Functions:

add_recent_places(): Function where the user adds previously visited places.

4.2.2.9 Recommended Places



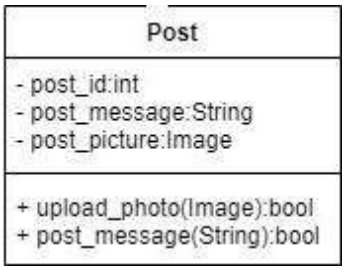
Fields:

radius: The radius of the area that the user will evaluate and suggest by the system.

Functions:

set_radius(): The function that determines the radius of the area that the user will evaluate and suggest by the system.

4.2.2.10 Post



Fields:

post_id: The system-issued unique number of the user-submitted post.

post_message: The message written by the user in the post sent by the user.

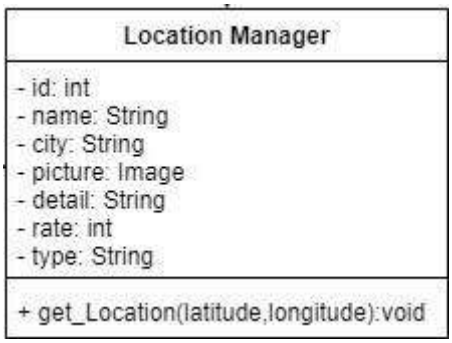
post_picture: User-submitted image inside the user-submitted post.

Functions:

upload_photo(): The function of the user-submitted image inside the user-submitted post.

post_message(): The function of the user-submitted message inside the user-submitted post.

4.2.2.11 Location Manager



Fields:

id: unique number of the location.

name: the name of the location.

city: city of location.

picture: picture of the location.

detail: details of the location previously written by another system.

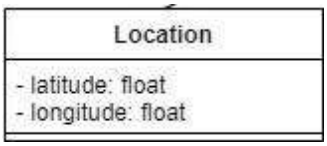
rate: prior evaluation of the location by another system.

type: the type of place found in the location.

Functions:

get_Location(): function that returns the location of the location.

4.2.2.12 Location

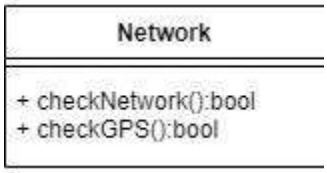


Fields:

latitude: the latitude of the location.

longitude: the longitude of the location.

4.2.2.13 Network



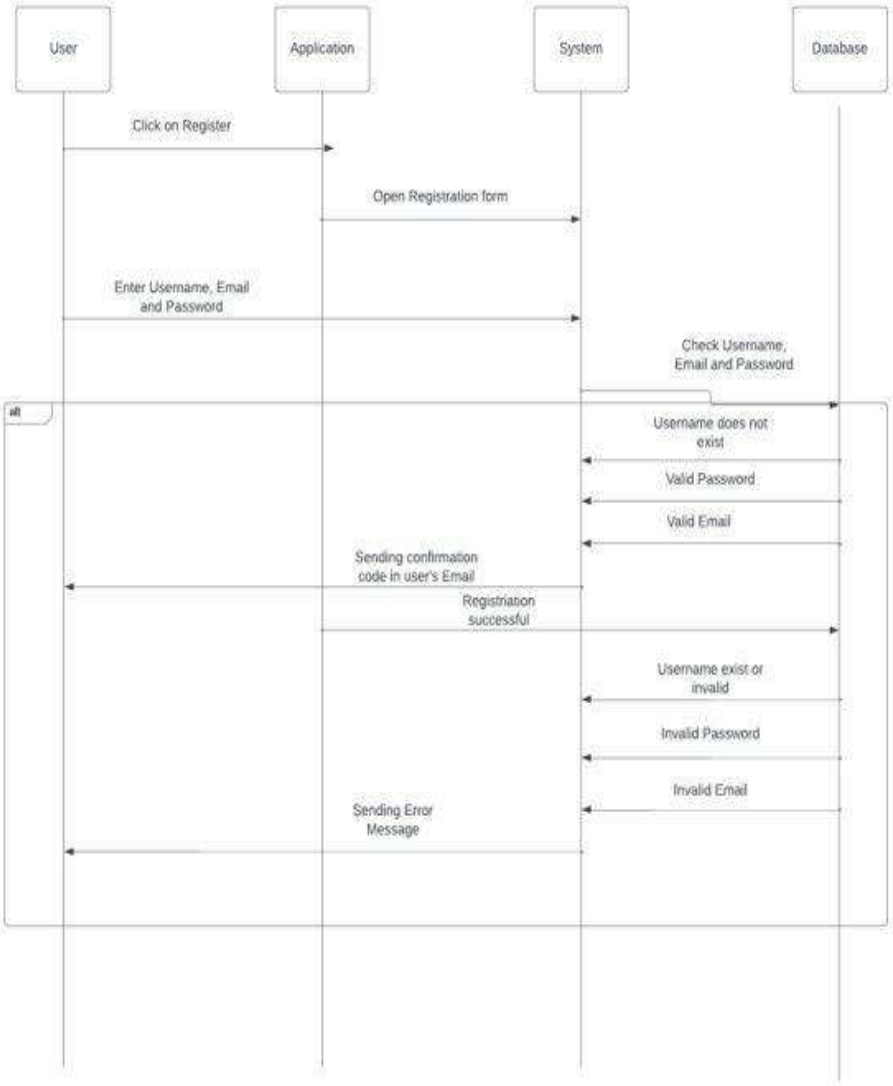
Functions:

checkNetwork(): Function that controls the system's internet access.

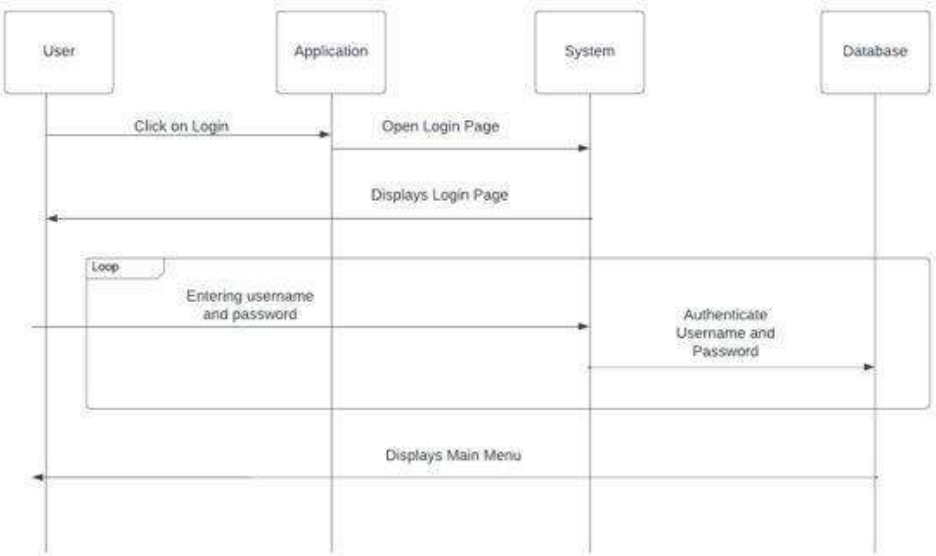
check GPS (): a function that controls the location access of the system.

4.2.3 Sequence Diagrams

4.2.3.1 Register Page

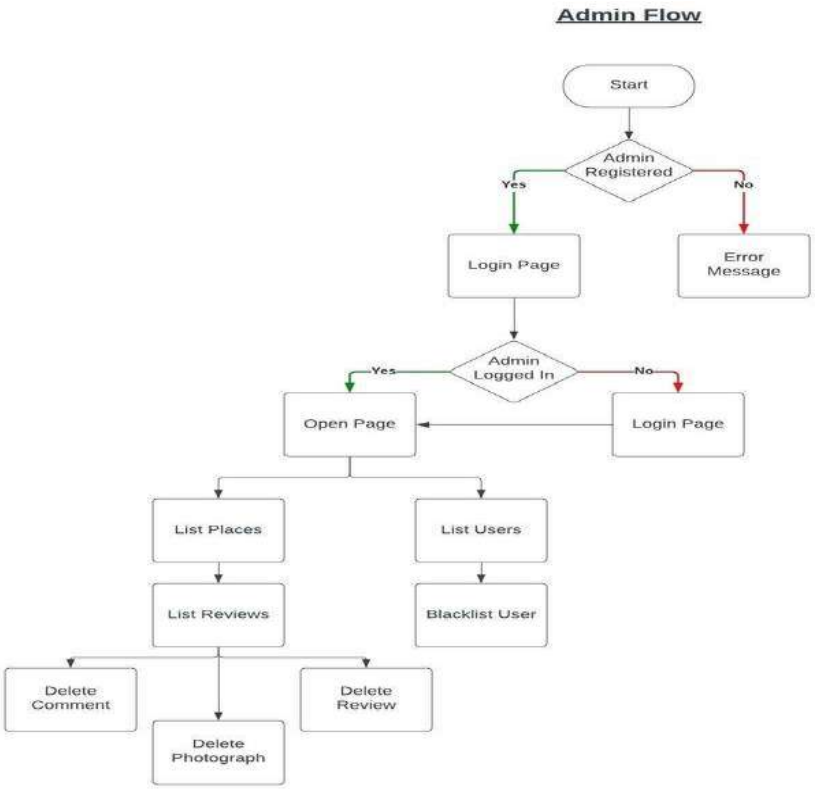


4.2.3.2 Login Page

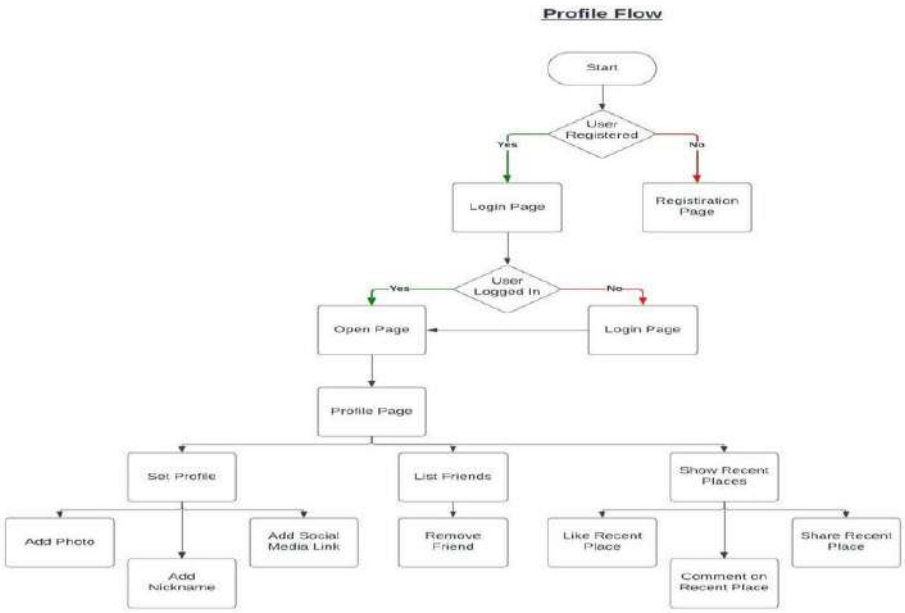


4.2.4 Activity Diagram

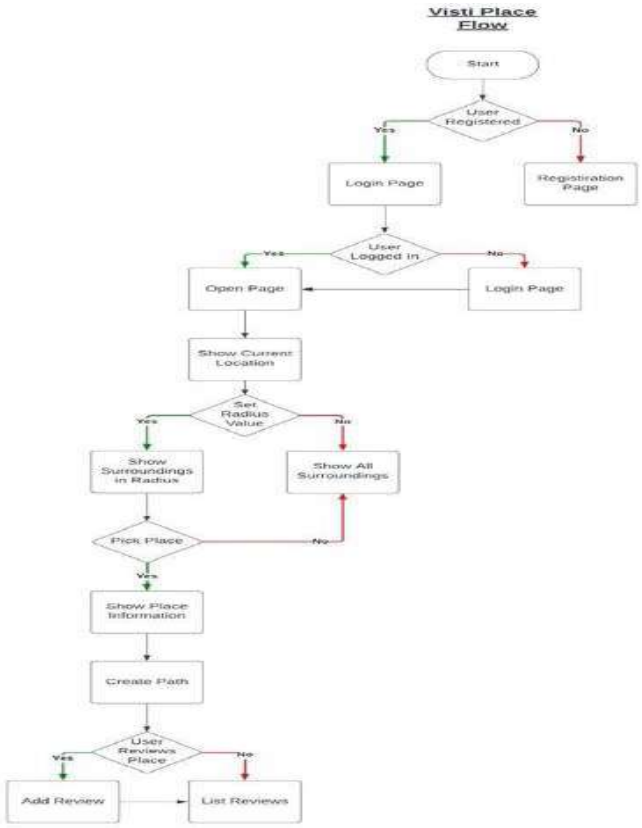
4.2.4.1 Admin



4.2.4.2 Profile

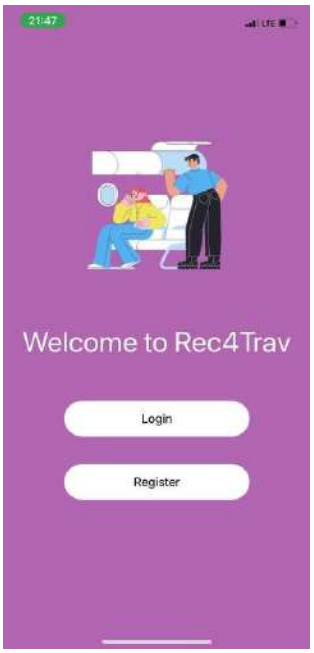


4.2.4.3 Visit Place

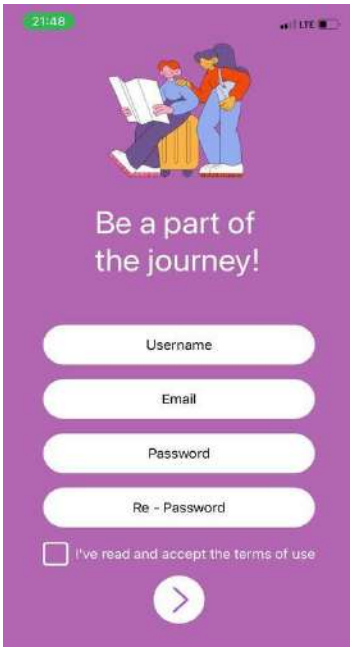


4.3 User Interface Design

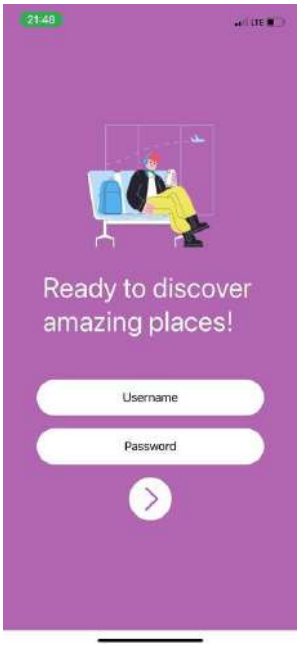
4.3.1 Open Page



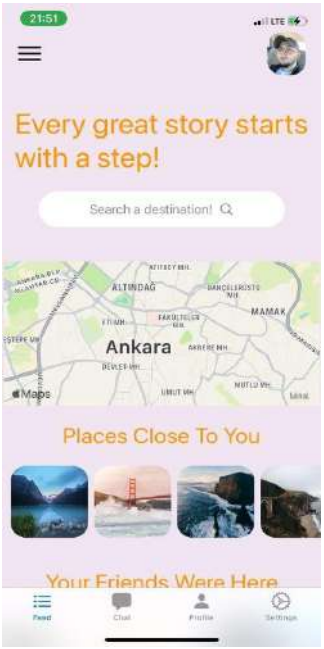
4.3.2 Register Page



4.3.3 Login Page



4.3.4 Main Page Top

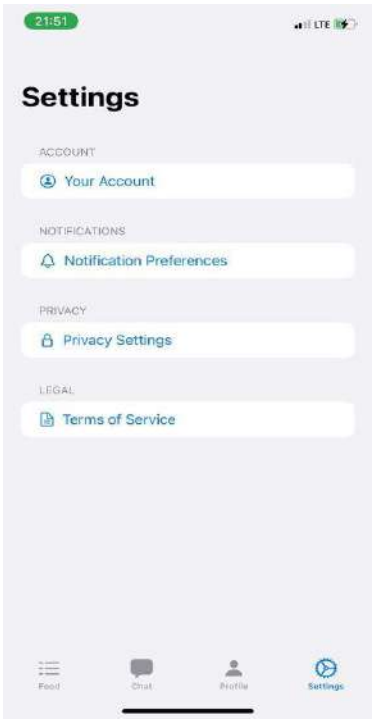


4.3.5 Main Page Down

4.3.6 Message

4.3.7 Profile

4.3.8 Settings



4.4 Constraints

4.4.1 Time

We got limited Time to Create this project so we will be using 3 steps finish method.

- 1st step is creating the general parts of the project (Creating UI, adding new users activating the mail system, etc..) which should take 1 week,
- 2nd step is Adding details step which is Creating under layers of the project (Giving votes, calculating the average, banning users etc.) which should again take 1 week,
- 3rd step is Completing the hardest part (which is Creating a map, showing restaurants and places around users, etc...) and completing the project which should take around 2 weeks.

4.4.2 Performance

As our project is a mobile project that is going to run on users' phones our project's performance will be dependent on the user's internet, Android, and IOS phone Memory, and their Android, and IOS versions.

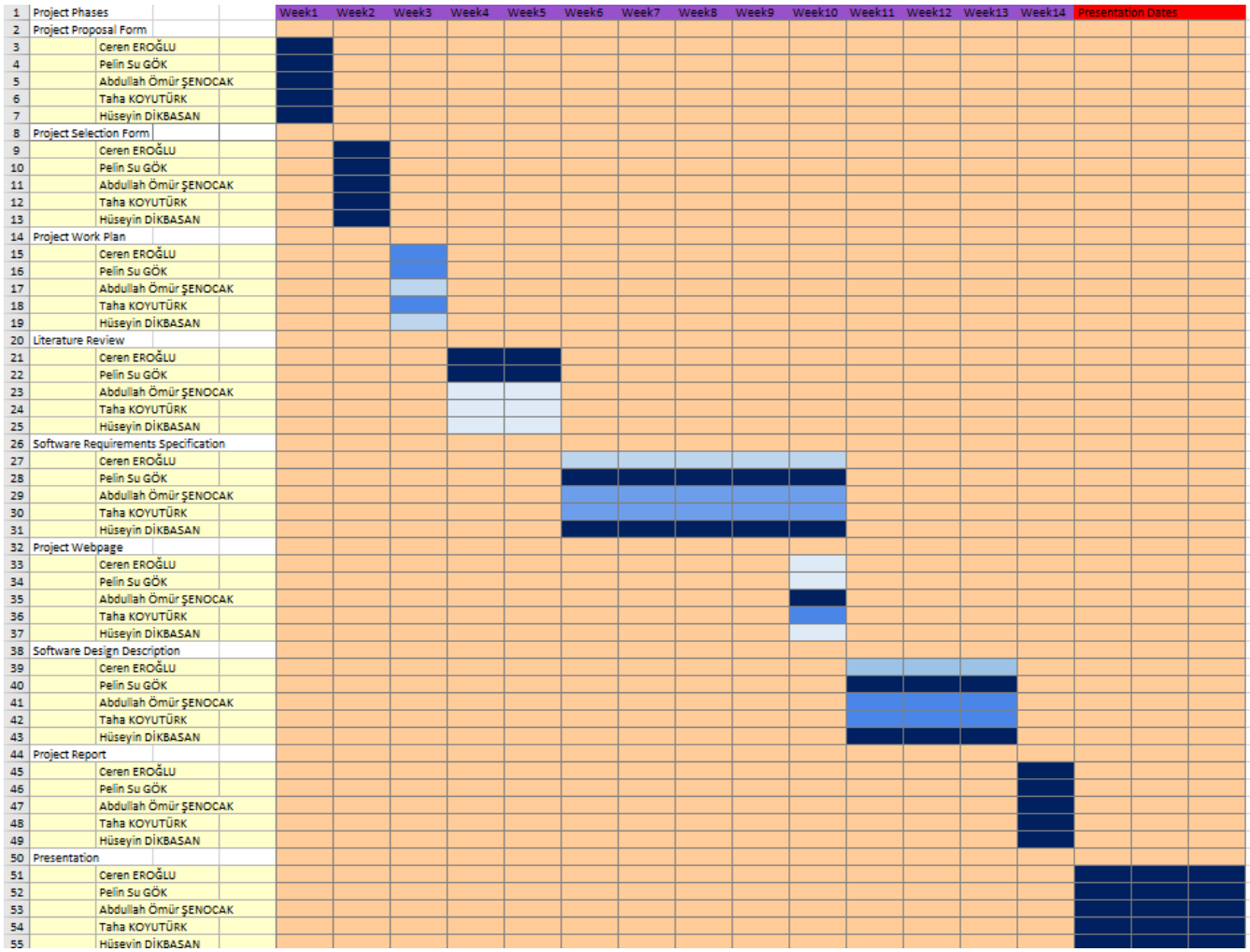
4.4.3 Application Constraints

- User can only use approximately 5000 words when giving a review. The real value of the word is to be confirmed later.
- The maximum distance that a map can be drawn is approximately 20 kilometers. The real value of the distance is to be confirmed later.

5 CONCLUSIONS

In our CENG 407 project, we researched and explained academic research recommendation for traveler systems. This report details the design phase of our recommendation for traveler systems application. Our mobile application suggests places within a certain distance specified by the user, according to the preferences of the users. In addition, it is also possible to make comments and evaluations about the places traveled so that users can communicate and socialize with other users. While designing this project, we also found such widely used travel recommendation applications useful for users. In addition, we aimed to improve the negative aspects of other applications and use their positive aspects in our own application. Thus, we have made it easier for users to determine a much more convenient and faster travel and to make recommendations based on individuals. We have observed that thanks to these applications, it has become easier for users to travel. During the project process, we researched the Flutter and Swift platforms in detail while creating the design of our mobile application. While Swift software language is faster than Flutter and has open source codes like Flutter, Swift language has many advantages, while its inability to be used on some platforms (Android,...etc) has a more important negative value for us. On top of that, we chose to go with Flutter, which we thought was more suitable for our project going forward. Another reason we chose the Flutter platform is that it is an open source platform created by Google that can develop UI software. At the same time, the fact that it can be used on many platforms (IOS, Android, ...etc) has been one of the main reasons for us to choose Flutter.

6 PROJECT WORK PLAN



7 REFERENCES

[1] V.K, Muneer & K P, Dr. (2020). The evolution of travel recommender systems: A comprehensive review. Malaya Journal of Matematik. 8. 10.26637/MJM0804/0075.

[2] C. A. Gomez-Uribe and N. Hunt, The Netflix recommender system: Algorithms, business value, and innovation, TMIS, 2016.

[3] Kenteris, Michael & Gavalas, Damianos & Mpitziopoulos, Aristides. (2010). A Mobile tourism recommender system. Proceedings - IEEE Symposium on Computers and Communications. 840-845. 10.1109/ISCC.2010.5546758.

[4] L R, Roopesh & Bomatpalli, Tulasi. (2019). A Survey of Travel Recommender System. 10.13140/RG.2.2.34775.32168.

[5] https://bujuyollarda.com/en-iyi-seyahat-uygulamalari/

[6] https://travelshopturkey.com/

[7] https://www.neredekal.com/blog/her-gezginin-mutlaka-kullanmasi-gereken-16-seyahat-uygulamasi/

[8] https://www.bizevdeyokuz.com/en-iyi-seyahat-uygulamalari/

[9] Gavalas, Damianos & Konstantopoulos, Charalampos & Mastakas, Konstantinos & Pantziou, Grammati. (2014). Mobile Recommender Systems in Tourism. Journal of Network and Computer Applications. 39. 319–333. 10.1016/j.jnca.2013.04.006.

[10] Park, Deuk-Hee & Kim, Hyea-Kyeong & Choi, Il-Young & Kim, Jae Kyeong. (2011). A Literature Review and Classification of Recommender Systems on Academic Journals. Journal of Intelligence and Information Systems. 17.

[11] Gironacci, Irene. (2021). Literature Review of Recommendation Systems. 10.4018/978-1-7998-4339-9.ch009.

[12] Google Maps Platform | https://developers.google.com/maps/documentation

[13] Overview of GPS-Based Mobile Apps | https://www.cogniteq.com/blog/overview-gps-based-mobile-apps-types-features-technologies#:~:text=GPS%20technology%20in%20mobile%20app,Uber%2C%20Lyft%2C%20and%20Curb.

[14] SDLC Waterfall Model | https://www.projectmanager.com/guides/waterfall-methodology

[15] Flutter Documentation | https://docs.flutter.dev/

[16] IEEE Standard for Information Technology - Systems Design - Software Design Descriptions | https://cengproject.cankaya.edu.tr/wp-content/uploads/sites/10/2017/12/SDD-ieee-1016-2009.pdf

[17] SOFTWARE DESIGN DOCUMENT | https://senior.ceng.metu.edu.tr/2014/such/documents/SDD.pdf

[18] Lucid | https://lucid.app/documents#/documents?folder_id=home

[19] Debashis Das & Laxman Sahoo & Sujoy Datta. (2017). A Survey on Recommendation System. International Journal of Computer Applications (0975 – 8887)

[20] F.O. Isinkaye, Y.O. Folajimi, B.A. Ojokoh. (2015) Recommendation systems: Principles, methods and evaluation