Bilkent University
Department of Computer Engineering

CS 491 - Senior Design Project

Fall 2024

Analysis and Requirement Report

T2419

Bora Haliloğlu - 22101852

Burak Oruk - 22102443

Emir Tuğlu - 22003165

Gökalp Gökdoğan - 22102936

Tevfik Emre Sungur - 22102377

Table Of Contents

1. Introduction

SürDur is a mobile application that offers users a pleasant road trip experience by recommending personalized stopovers along the route. After selecting a destination, users can explore a range of recommended points of interest (POIs) drawn from various databases and travel blogs, guaranteeing a customized and rich trip experience. Through its social media feature, which allows users to vote on routes, follow other travelers, and discuss their travel experiences, SürDur promotes community involvement in addition to route planning. This social component improves user engagement and offers feedback loops for personalized recommendations.

In this report, we will first discuss the current market situation and analyze our competitors. Then, we will elaborate on our proposed system by explaining its requirements and providing system models, which include various diagrams and the application's user interface. Lastly, we will discuss other analysis elements, including considerations in engineering design, risks and alternatives, project plan, and other elements. By the end of the report, readers will have a complete understanding of the analysis and requirements of the SürDur project.

2. Current System

In the current system, most of the potential SürDur users plan their route using either Google Maps or Yandex Maps. These apps also have the feature of searching locations along the route; however, this is not the main feature of the apps, therefore this feature is not commonly used since these apps don’t have a recommendation mechanism that SürDur will implement. The current mechanism in these apps is searching locations based on the search keyword that the user enters. All locations that have the entered keyword in their name are displayed without any filtering.

There are also other applications that try to solve the same problem with SürDur. However, some of these apps are not supported in Türkiye, some others lack critical features like navigation, and some have quite inferior user interfaces that makes it a hassle to use. A list of the most important of these competitor applications can be found below.

Contrary to the competitors, SürDur will provide precise place recommendations that users would like to add to their route. Moreover, SürDur will have navigation functionality and provide all of them in an easy-to-use and simple user interface. For these reasons, it is

inevitable that SürDur will reach a large market penetration and will be the go-to app for anyone looking for stopovers along their route.

2.1. Market & Competitor Analysis

2.1.1. Roadtrippers

- Available in only the USA, Australia, Canada, and New Zealand.
- Creating road trips is free, but navigating along them requires a premium.
- Has some pre-created road trip suggestions.
- Users can choose what categories of recommendations they want.
- There is a map-saving option for offline usage.
- Has a chatbot that generates road trips by asking questions as a premium feature.
- Has 1M+ downloads.

2.1.2. Roadie

- Has no navigation option, it exports the route to Google Maps for navigation.
- Locations can be searched by category.
- Waypoints can be added, deleted, and reordered.
- Users can save routes, but there is no sharing option.
- Free mode is restricted.
- Has 100K+ downloads.

2.1.3. Sygic Travel

- Has no navigation option, it exports the route to Google Maps for navigation.
- Dynamically retrieves locations as the user moves on the map.
- Has offline maps mode.
- Poor UI/UX.
- Has 1M downloads.

2.1.4. Google Maps

- Has an explore feature which shows all locations by category as the user moves on the map.
- Has search along the route feature. Locations are retrieved by search keywords, which is not useful.

2.1.5. Yandex Maps

- Has a separate explore option which shows locations of selected categories as the user moves on the map.
- The explore option is separated from the route option, suggested locations are not shown on the route.

2.2. Comparison with Competitors

	SürDur	Roadtrippers	Roadie	Sygic Travel	Google Maps	Yandex Maps
Navigation	+	+	-	-	+	+
Displaying the POIs and the route together on a map	+	+	+	+	+	-
Retrieving POIs from multiple sources	+	?	?	?	-	-
Personalized POI Recommendations	+	-	-	-	+	+
Sharing routes with friends	+	-	-	+	+	+
Has a chatbot assistant	-	+	-	-	-	-
Available on Türkiye	+	-	+	+	+	+

Table 1: Comparison with Competitors

3. Proposed System

3.1. Overview

SürDur is an application that provides personalized stopover recommendations on users' routes. The purpose of the application is to connect long-distance travelers with the best places to stop along the way. After selecting start and destination locations, users are provided with points of interests (POI) such as restaurants, natural beauties, and historical sites. These POIs can be added and removed from the route. Users can also filter POIs according to categories and get more specific recommendations. After

finalizing the route, users can navigate through each stopover and reach their destination by using the navigation feature of the application.

In SürDur, there is also a social media functionality. Users can follow their friends and other people using the application. They can share their routes, see other posted routes, and upvote or downvote them. Users can also clone a shared route which enables them to modify and navigate it. In this way, users can visit the locations that people they follow are satisfied with.

SürDur will provide place recommendations according to the preferences of the users. Recommendations will be based on three main data sources: users' responses to questions about their stopover preferences, places that they previously visited, and POIs in the routes that they have upvoted on social media. In this way, we aim to provide users with personalized and accurate recommendations that they will be pleased with.

3.2. Functional Requirements

This section will describe the functional requirements that SürDur application should include for proper implementation.

3.2.1. Sign Up & Login

Application should:

- Allow users to log in to the application using their account information.
- Allow users to register manually by providing account details.
- Allow users to add their personal information, such as name, age, and preferences.
- Enable users to add place category preferences for road trips.

3.2.2. Destination Selection

Application should:

- Display the user's current location on an interactive map.
- Allow users to move around the interactive map.
- Provide a feature to return to the user's current position on the map.
- Allow users to select a destination location for route planning.

3.2.3. Point of Interest Suggestion

Application should:

- Display suggested points of interest (POIs) close to the given route.
- Allow users to zoom in or out to a particular region to get an adequate number of suggestions in that area.
- Allow users to filter POIs by category to display only the POIs with the selected categories.
- Display the preview information about the suggested places.
- Display detailed information about the selected place among the list of suggested places.
- Allow users to add places to the route from the given suggestions.
- Allow users to remove places from the route among all of the previously selected places.
- Reconstruct the route dynamically after each place addition and removal.
- Display the estimated time to reach the destination and the percentage of how off the new route is from the original route.
- Allow users to finalize the route with selected places and save it on the ‘Planned’ routes of the user.

3.2.4. Social Media

Application should:

- Display the shared route posts from other users on the application, with the components:
    - Starting and Destination Location
    - Interactive Map Overview of The Route
    - Title
    - Description
    - Author Username and Profile Picture
    - Upvote and downvote counts
- Allow users to display detailed description of the selected route among shared routes.
- Open the selected shared route by the user on the interactive map.
- Allow users to upvote or downvote the route post.

- Allow users to follow and unfollow other users
- Allow users to save the selected shared routes on the ‘Saved Routes’ folder of the user.
- Send users notifications when there are upvotes or downvotes on their posts.
- Send users notifications when the users that are followed share their routes.

3.2.5. Archives

Application should:

- Display all the personal routes of the user that are divided into four route categories:
    - In Draft: Routes that have not been totally completed on planning by the user and require further route planning completion from the user.
    - Planned: Routes that have been assigned as ‘completed planning’ by the user and are ready to be traveled afterward.
    - Saved: Routes from other users that have been saved on the local archive from the social media
    - Finalized: Routes that have been traveled and finished by the user in real life.
- Allow users to filter personal routes based on their category, destination location, and start location.
- Open the selected route from the archive on the interactive map for further route editing or to start the road trip.
- Allow users to delete present routes from the user archive.
- Allow users to share their routes on the social media part of the application.
- Allow users to flag certain routes for further distinction from other routes.

3.2.6. Live Route Navigation

Application should:

- Allow users to start live navigation throughout the selected route.
- Perform real-time navigation throughout the road trip.
- Display real-time road directions (next maneuver’s turn and distance etc.) on the screen.

- Display the total expected time remaining.
- Allow users to select emergency 'gas station' and 'electric vehicle charging station' stops.
- Add the nearest and most convenient gas station or electric vehicle charging station dynamically into the current route if chosen.
- Adjust live directions dynamically when the driver gets off the planned route.
- Allow users to exit the live navigation of the current route.

3.2.7. Profile Editing & Setting Adjustment

Application should:

- Allow users to edit their personal account information, such as username, password, etc.
- Allow users to adjust application experience settings, such as theme, distance and speed units, notification enablement, etc.

3.3. Nonfunctional Requirements

3.3.1. Usability

The experience of the usage of SürDur will be evaluated based on both application usage time, and personalized place suggestion satisfaction. Hence, SürDur must provide a user-friendly interface by including simple, easy-to-use also comprehensive components to provide a smooth state transaction from the main page to the completion of the planned route. In that sense, SürDur should provide a proper number of place suggestions along the main route in order to not exhaust the user and also to prevent a lack of suggestions. The application must also execute smoothly for both mobile devices (including Android and iOS) and CarPlay devices.

3.3.2. Reliability

SürDur must provide an uninterrupted navigation performance throughout the road trip. Hence, the application should handle network errors by caching the

map information on the local machine, and continue to serve navigation, which is independent from the network. Also, the application should include navigation services in the case of a failed satellite connection, and inform the driver about the issue safely. The application should continue its live navigation services in any possible server downtime coming from maintenance and update processes. Other than error handling criteria, the application should store backup of the main database in the case of system crash and information loss.

3.3.3. Performance

SürDur must have an advanced data-retrieval service that would minimize the response time in each route planning suggestion. The application should cache the most related places from the enormous database of place information based on the location, relevance and personal choices of the user. Other than data retrieval performance, live navigation should have minimized lagging for the driver to have a real-time driving experience for both satisfaction and safety.

3.3.4. Supportability

SürDur must have a globally compatible service and database structure. The development environment should allow the addition or removal of microservices that alter the place of information on the database without interrupting the main functionality of the application. Also, the development environment should include environment containerization services, such as Docker, to easily deploy new libraries and configurations globally with one main structure. Other than a compatible development environment, the system should include logging services to keep track of the occurring events regularly, including errors and warnings.

3.3.5. Scalability

SürDur must handle enormous workloads on large-scale user scenarios by considering storage space, user request overloading, and performance efficiency. Hence, the application should include fast and asynchronous backend services, information storage optimization, and efficient caching services. Other than large-scale user demand, SürDur also should consider vast amounts of

place data, and how to store them and retrieve them efficiently. This issue also shows the importance of information storage optimization and efficient caching services.

3.4. Pseudo Requirements

- SürDur will be available as both iOS and Android applications.
- The app will require internet connection.
- The app will need permission to reach the location of users.
- Frontend of the app will be implemented with JavaScript and React Native.
- Backend of the app will be implemented with Python and FastAPI.
- MySQL will be used as RDBMS.
- Git and GitHub will be used for version control.
- The app will utilize Google Maps API to display the map on the interface.
- The app will utilize OpenStreetMap API to create routes.
- AWS will be used to host the backend of the application.
- Initially, the app will be available only in Türkiye.

3.5. System Models

3.5.1. Scenarios

Use Case Name	Get Place Recommendation
Participating Actor	User
Flow of Events	1. User selects a destination 2. Recommendations along the route are given to user
Entry Condition	● User is logged in ● Destination is selected
Exit Condition	● User completes the route ● User leaves the navigation page

Use Case Name	Create Post
Participating Actor	User

Flow of Events	1. User opens up the saved routes page 2. Selects one of his/her routes 3. Clicks on create post button 4. Enters post description 5. Adds visuals 6. Clicks on create button
Entry Condition	● User is logged in ● User has at least one route created
Exit Condition	● User clicks on back button

Use Case Name	Select Destination
Participating Actor	User
Flow of Events	1. User opens up the main page 2. Clicks on search bar 3. Types the destination 4. Selects the destination
Entry Condition	● User is logged in
Exit Condition	● User removes destination

Use Case Name	Add Place to Route
Participating Actor	User
Flow of Events	1. User opens up the main page 2. Selects a destination 3. Adds one of the recommended places to route
Entry Condition	● User is logged in ● User selected a destination
Exit Condition	-

Use Case Name	Remove Place from Route
Participating Actor	User
Flow of Events	1. User selects one of the places in route 2. Clicks on remove button
Entry Condition	● User is logged in ● User has a route including at least a place

Exit Condition	-

Use Case Name	Filter Recommended Places
Participating Actor	User
Flow of Events	1. User opens up the main page 2. Selects a destination 3. Filters recommended places by category by clicking on a category
Entry Condition	● User is logged in ● User selected a destination
Exit Condition	● User clicks again on the same category

Use Case Name	Interact with World Map
Participating Actor	User
Flow of Events	1. User opens up the main page 2. Interacts with the World Map
Entry Condition	● User is logged in ● User is on the main page
Exit Condition	● User leaves main page

Use Case Name	See User Profile
Participating Actor	User
Flow of Events	1. User opens up the social media page 2. Clicks on profile picture of a user 3. Clicks on follow button (optional)
Entry Condition	● User is logged in ● User is on the social media page
Exit Condition	● User leaves the profile page

Use Case Name	See Post
Participating Actor	User

Flow of Events	1. User opens up the social media page 2. Views posts 3. Clicks on a post to view its details 4. Upvotes/downvotes post (optional)
Entry Condition	● User is logged in ● User is on the social media page
Exit Condition	● User leaves social media page

Use Case Name	Save Post
Participating Actor	User
Flow of Events	1. User opens up details page of a post 2. Clicks on the save button
Entry Condition	● User is logged in ● User is on details page of a post
Exit Condition	● User clicks on the save button again

Use Case Name	See Saved Posts
Participating Actor	User
Flow of Events	1. User opens up the saved routes page
Entry Condition	● User is logged in
Exit Condition	● User leaves saved routes page

Use Case Name	Login
Participating Actor	Non logged-in User
Flow of Events	1. User opens up the application 2. Enters credentials 3. Clicks on login button
Entry Condition	● User is not logged in
Exit Condition	● User logs in

Use Case Name	Sign Up

Participating Actor	Non logged-in user
Flow of Events	1. User opens up the application 2. Clicks on sign up button 3. Enters credentials 4. Clicks on submit button
Entry Condition	● User is not logged in
Exit Condition	● User registers ● User logs in

Use Case Name	Reset Password
Participating Actor	User
Flow of Events	1. User opens up the application 2. Clicks on forgot password button 3. Enters email address 4. Clicks on reset password button
Entry Condition	● User is not logged in
Exit Condition	● User resets password ● User logs in

Use Case Name	Change Credentials
Participating Actor	User
Flow of Events	1. User opens up his/her own profile 2. Changes his/her credentials 3. Clicks on save button
Entry Condition	● User is logged in
Exit Condition	● User leaves his/her profile

3.5.2. Use Case Model

The following use case diagrams demonstrate how the user actors interact with the SürDur app's different states, and what functionalities are available on each page.

Introductory Page Use Cases

This diagram shows how the User can use the suggestion page. This is the initial screen for entering the application. The figure shows how the User can proceed with the app.

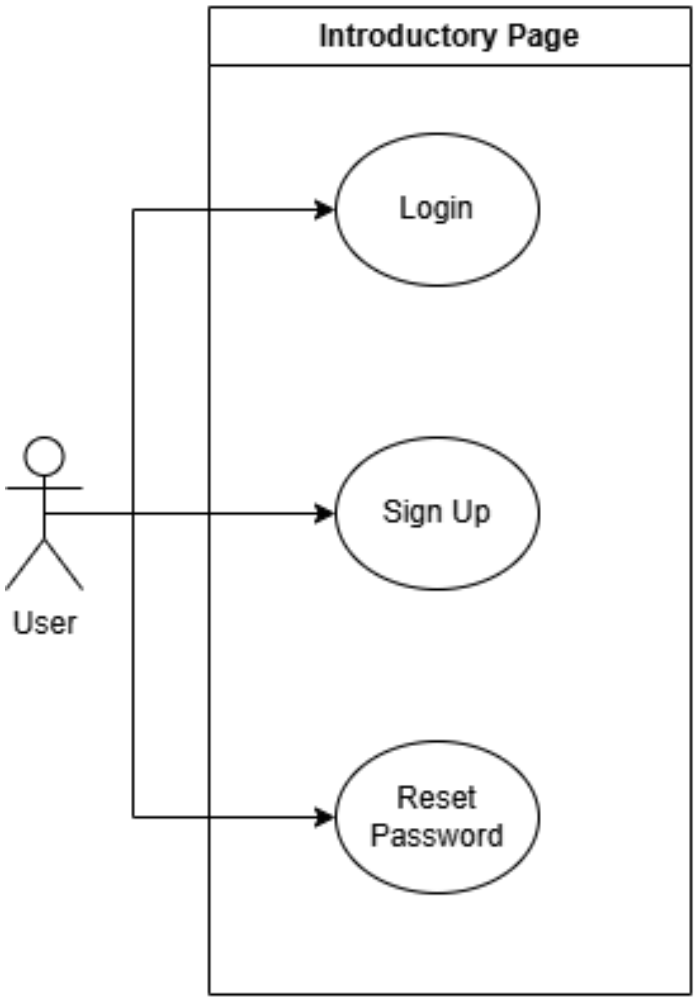

Figure 1: Use Case for Suggestion Page

Social Page Use Cases

This diagram shows how the User can use the social page. This is where the publicly shared routes of the other users will be displayed. The figure shows how the User can interact with the posts.

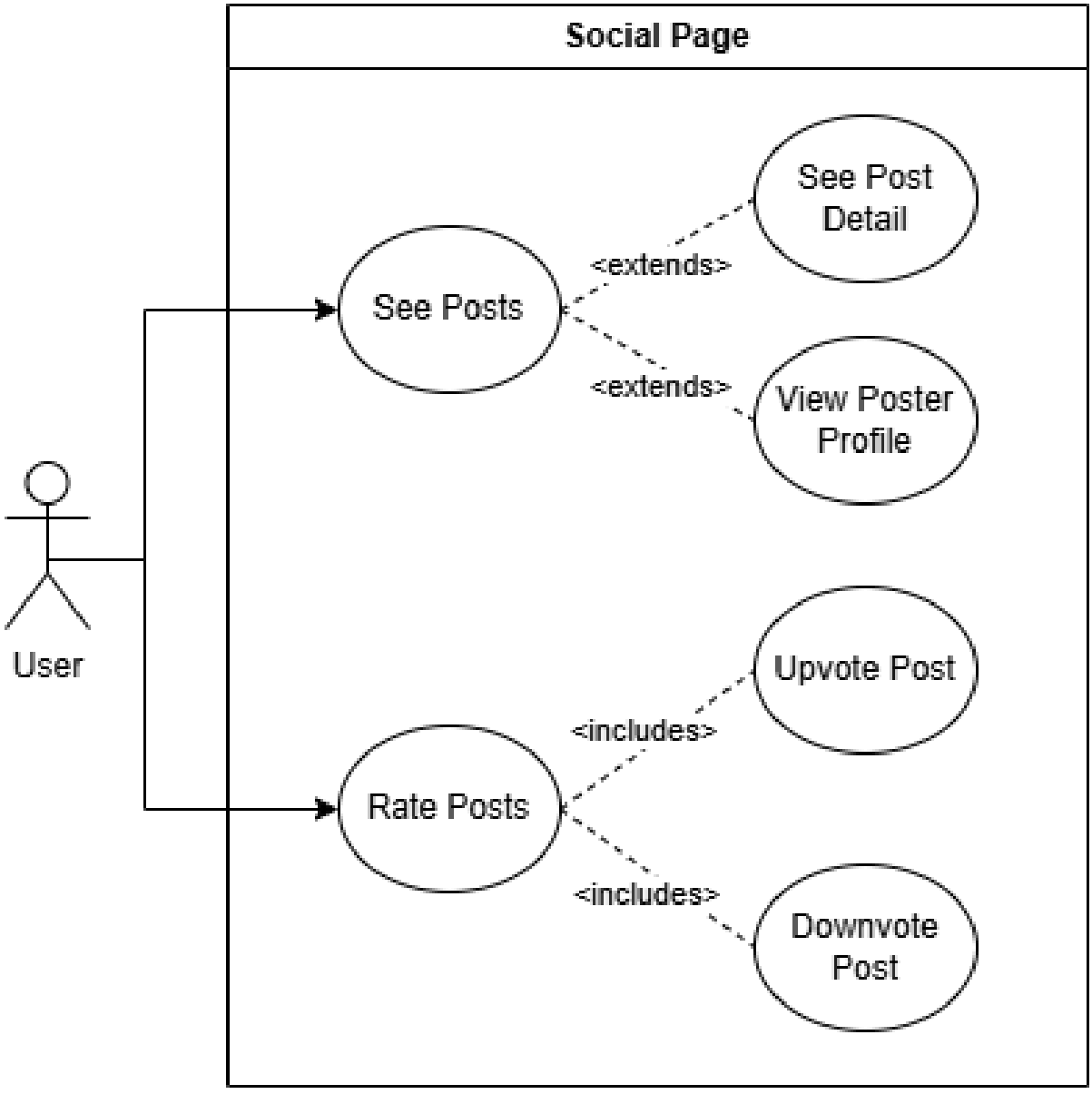

Figure 2: Use Case for Suggestion Page

Saved Routes Page Use Cases

This diagram shows how the User can use the saved routes page. Previously done or planned are showcased here. The figure shows how the User can interact with saved routes.

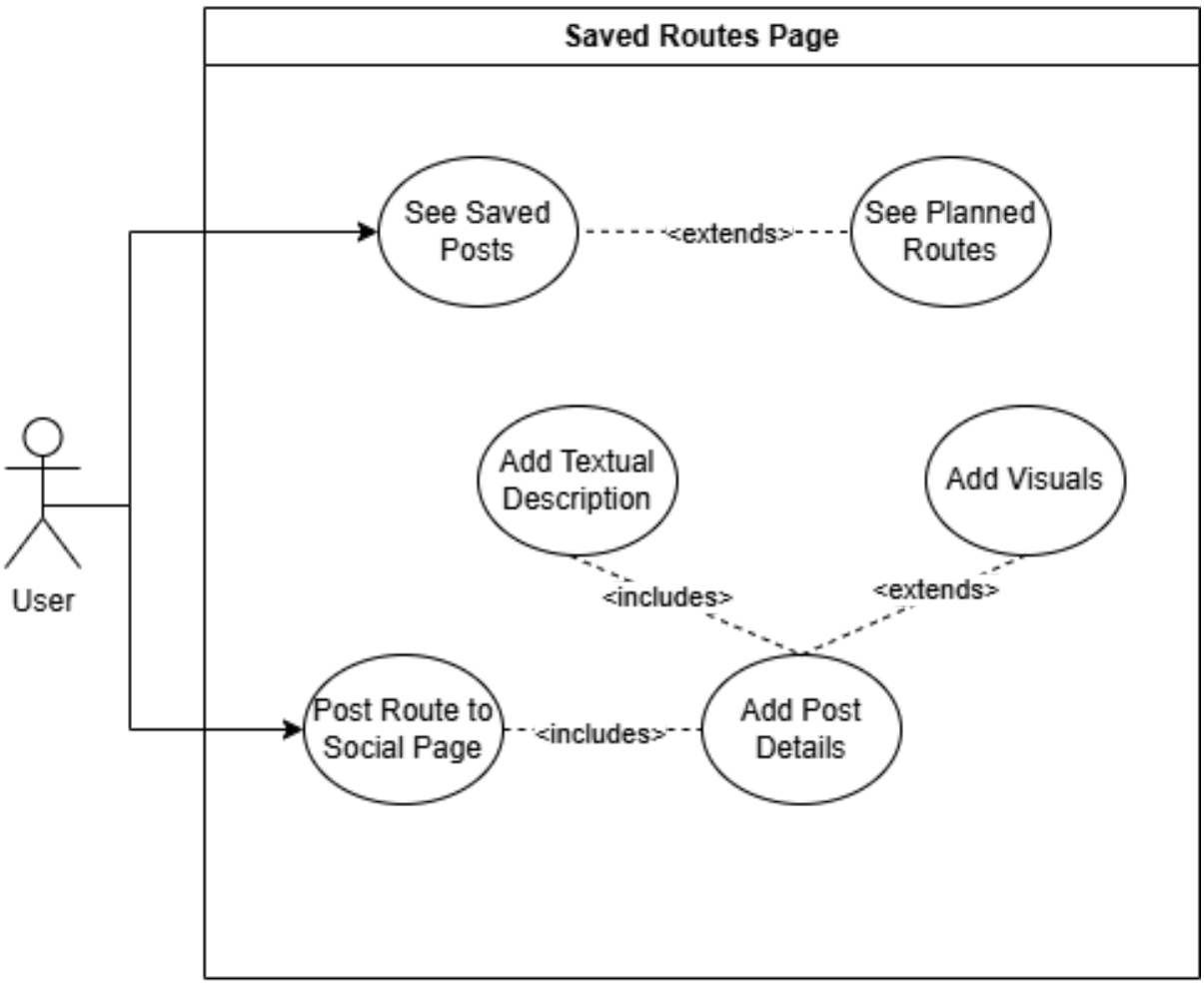

Figure 3: Use Case for Suggestion Page

Suggestion Page Use Cases

This diagram shows how the User can use the suggestion page. The suggestion page is where the system will suggest stopovers to the user. The figure shows how the User can interact with those suggested pinpoints

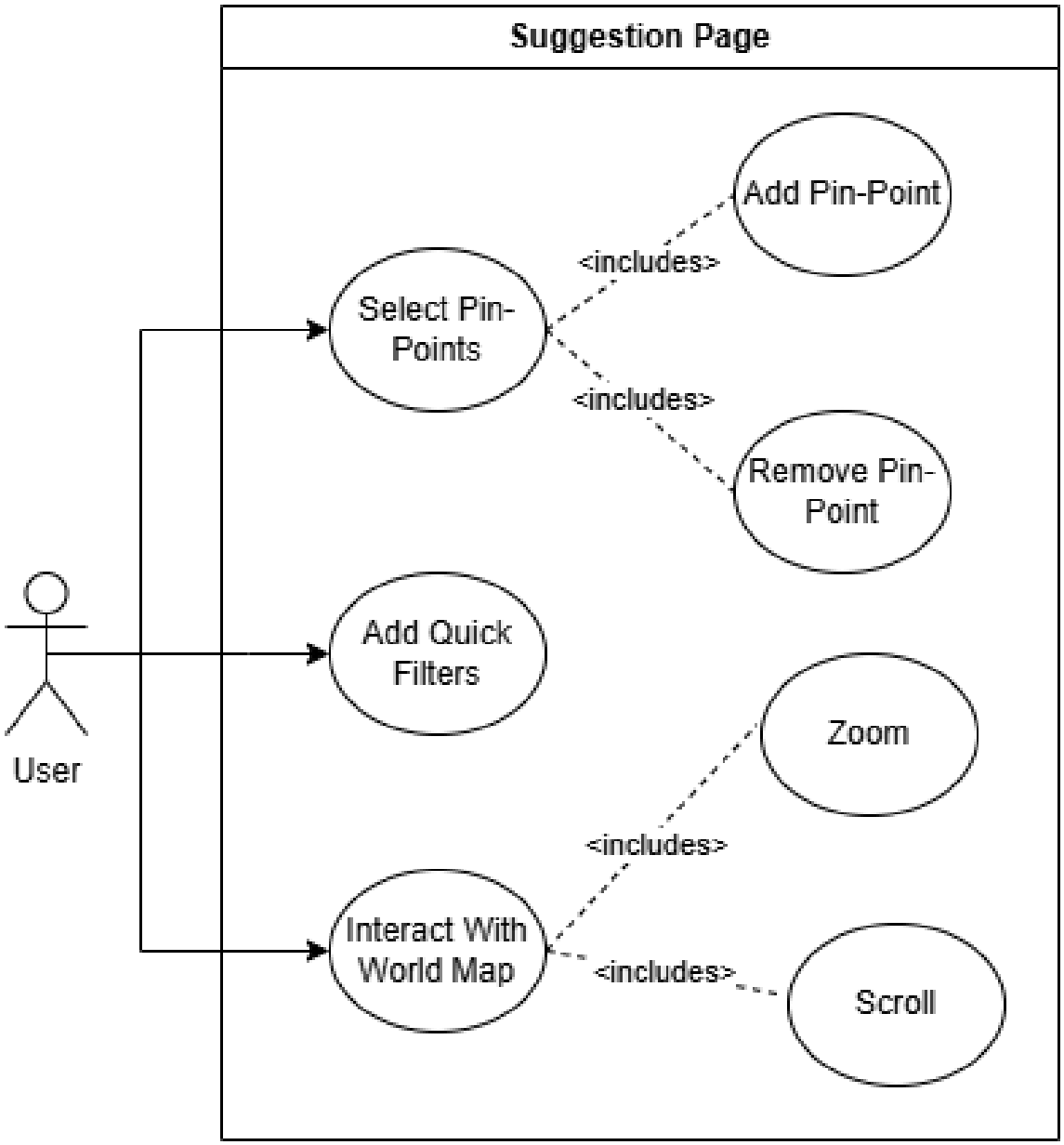

Figure 4: Use Case for Suggestion Page

Profile Settings Page Use Cases

This diagram shows how the User and Visiting Users can use the profile page. While having different uses for different users, its purpose is to display profile information of the related user. The figure shows how the User and Visiting User can interact with the profile.

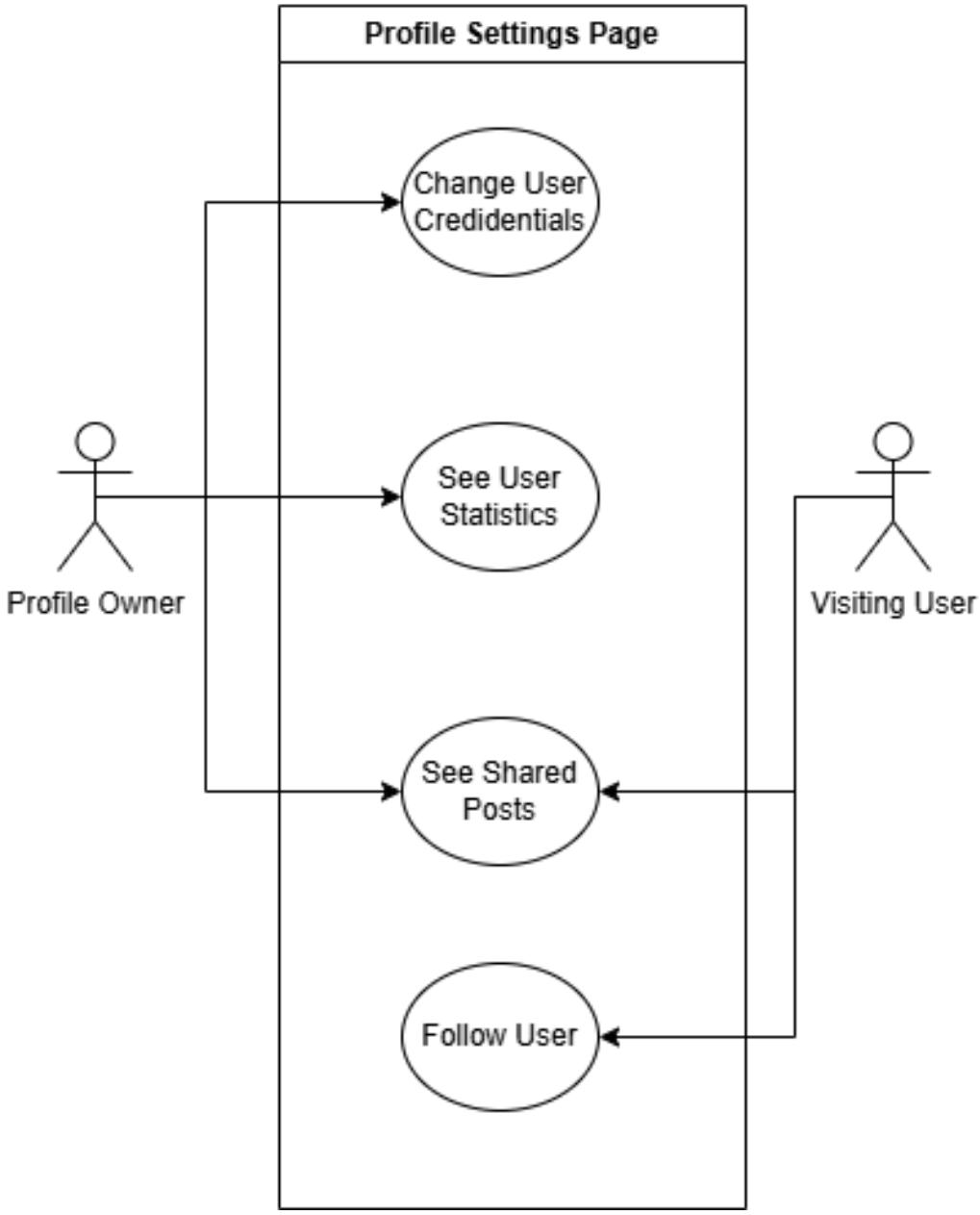

Figure 5: Use Case for Suggestion Page

3.5.3. Object and Class Model

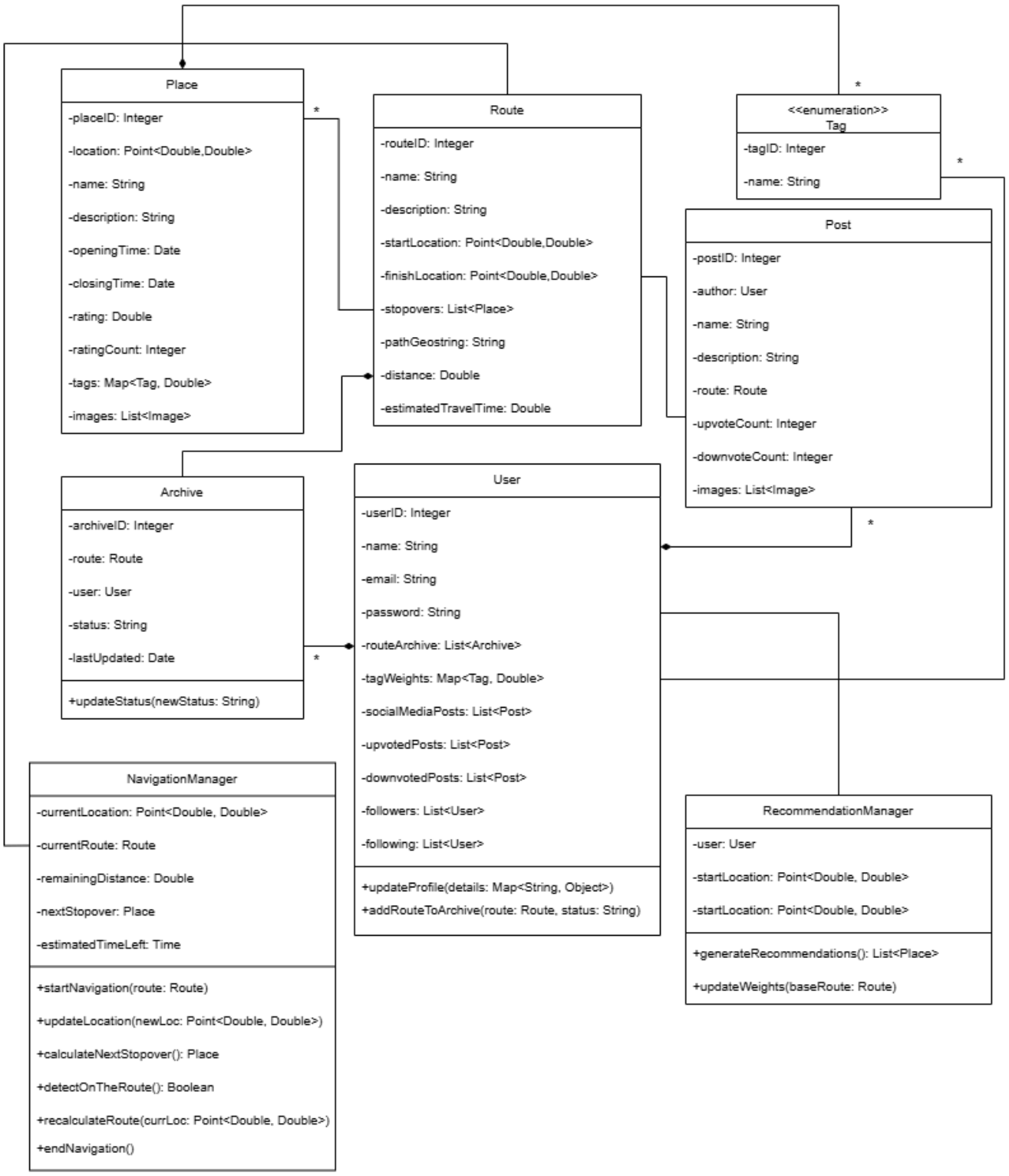

Figure 6: Class Diagram

3.5.4. Dynamic Models

3.5.4.1. Activity Diagrams

Route Creation

Route Creation Activity Diagram explains how a route can be created. It explains how our layered route filtering works to ensure best recommendations for our users. It ensures scalability with our layered design.

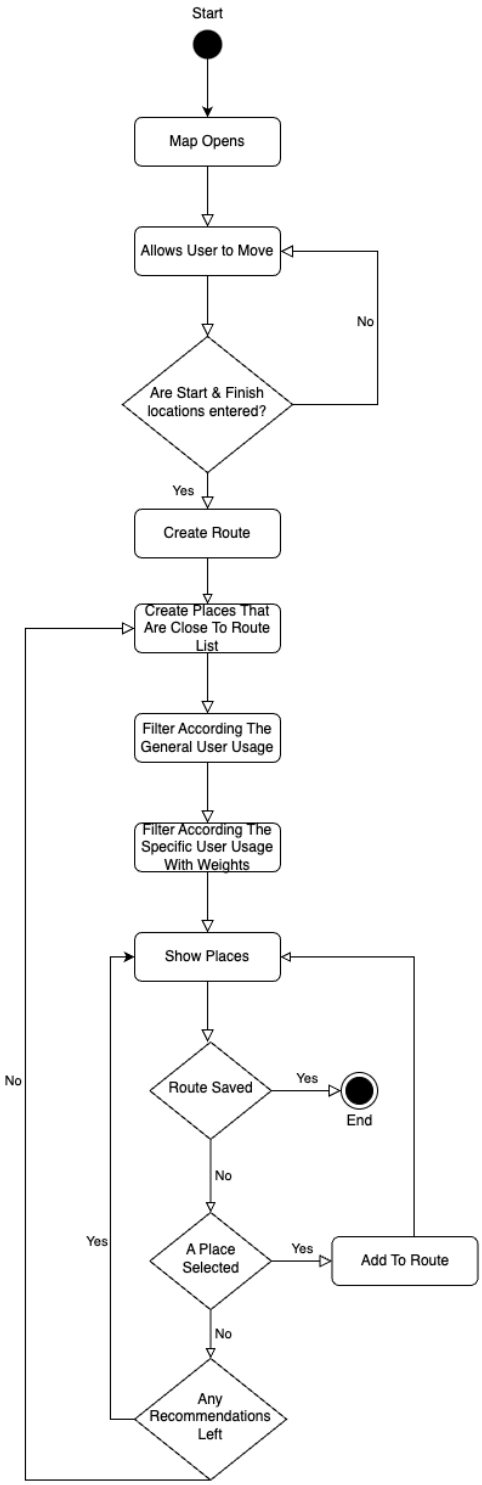

Figure 7: Route Creation Activity Diagram

Post Creation

Post Creation Activity Diagram explains how a post can be created. It explains how uploaded user images are handled and how posts are saved in the database. It ensures scalability with our layered design.

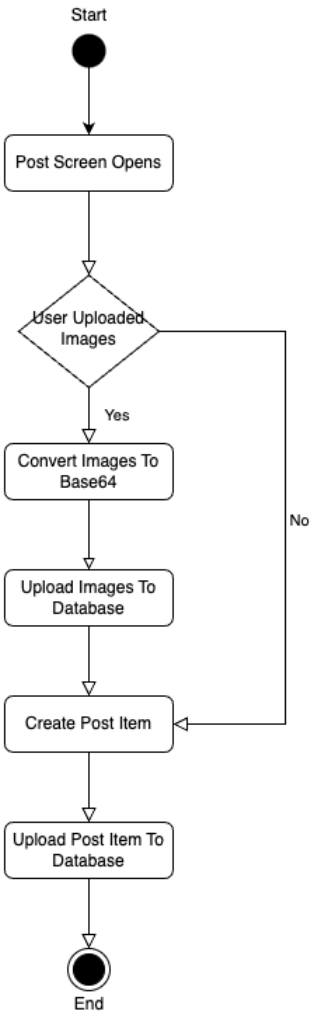

Figure 8: Post Creation Activity Diagram

3.5.4.2. State Diagrams

Dynamic Route Formation:

This state diagram shows the state transitions of the route which the user is forming in the suggestion page of the application. If the user did not choose any stopovers, the route is assigned as Empty Route,

representing the optimal route without any stopovers. If there are added stopovers but not finalized, the route is in the Updated Route state. If the user decides to finalize and see the overall route, the route is in the Completed Route state. Finally if the user decides to save the completed route to the archive, the user leaves the route formation state.

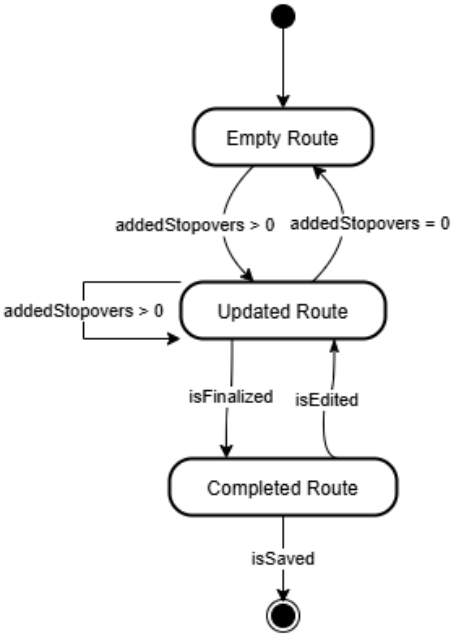

Figure 9: Dynamic Route Formation State Diagram

User Archive Route Classification:

This state diagram shows the state transitions of the route from the user's personal archive. The initial state directs to the In-Draft Route where the route has not been completed by the user yet. If the user finalizes the route, the route state is Completed Route. Finally, if the completed route is travelled by the user, the state becomes Travelled Route. In any state, if the route is deleted, the state of the current route is finalized.

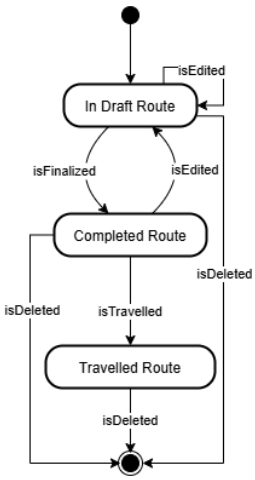

Figure 10: User Archive Route Classification State Diagram

3.5.4.3. Sequence Diagrams

Post Creation

The following is a diagram of how users send a post. Other users, namely those following the said user, will receive a notification. The flow begins when the user sends a post(description) request to the page. The Page then calls sends_post(description) to forward this request to the PostForm. The PostForm uses a second sends_post(description) method to forward this to the Backend. After processing the request, the backend notifies the user via a notice string to confirm the post. If processing issues arise, the PostForm receives an error response from the Backend, relaying these errors to the Page and, ultimately, to the User. A simple posting procedure with an emphasis on success and error handling is depicted in this diagram.

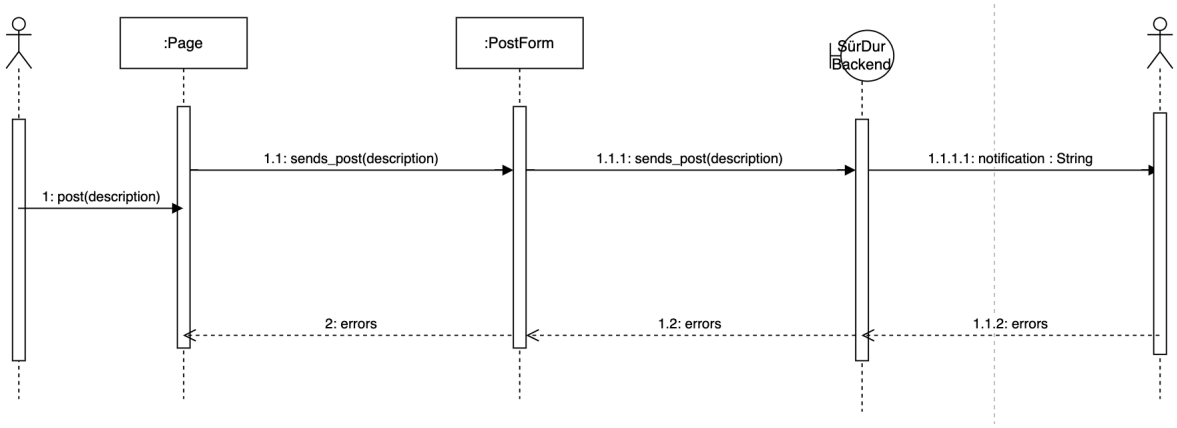

Figure 11: Post Creation Sequence Diagram

Route Generation

A more intricate interaction for creating customized routes between a User, the GenerateRoutePage, and the Backend is depicted in the second sequence diagram. To request create_personalized_route(destination) from the Backend, the user first calls generate_route(destination) on the GenerateRoutePage. To customize the route, the backend modifies user weights and provides the user with a list of places of interest (POIs). Two extra flows are available: by calling generate_new_places(), the user can request new locations, which causes the Backend to create and deliver alternate locations displayed to the user. Additionally, by using show_saved_routes(), which retrieves and shows the saved routes from the Backend, the user can examine stored routes. With optional features for finding alternate and saved routes, this figure highlights customized route development.

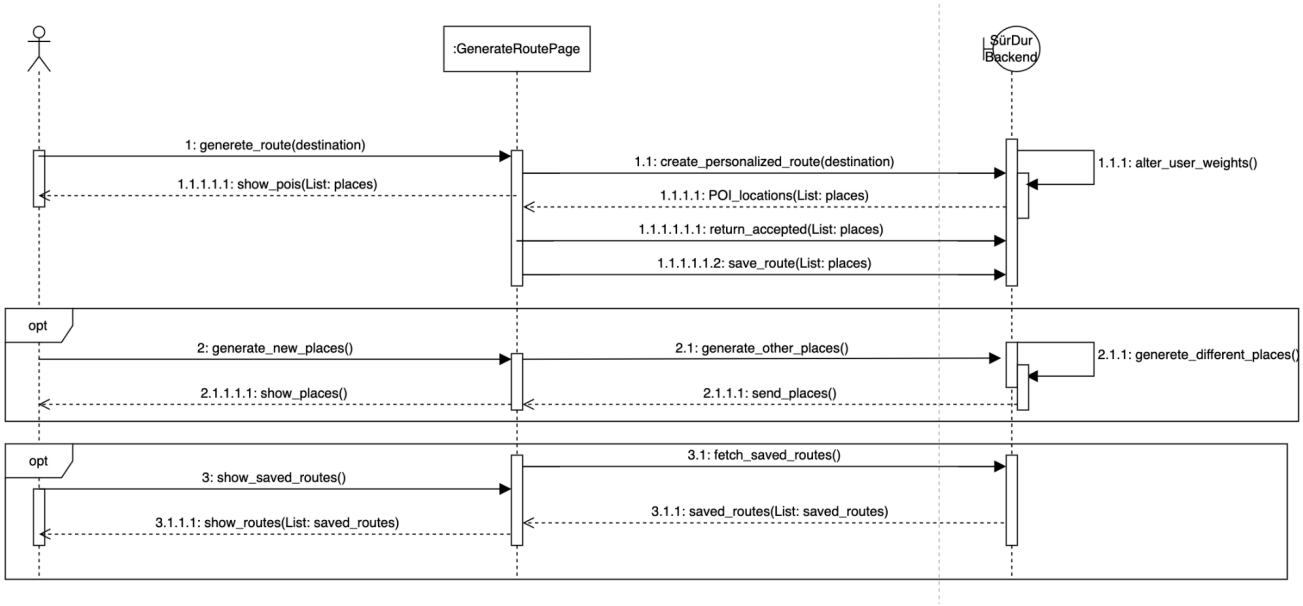

Figure 12: Route Generation Sequence Diagram

3.5.5. User Interface - Navigational Paths and Screen Mock-ups

Onboarding Page

The onboarding page serves as a welcome screen for SürDur, providing users with two options to get started: logging in for returning users and signing up for new users. Its purpose is to offer a simple entry point, allowing users to quickly access or register for the app while setting the tone for our brand.

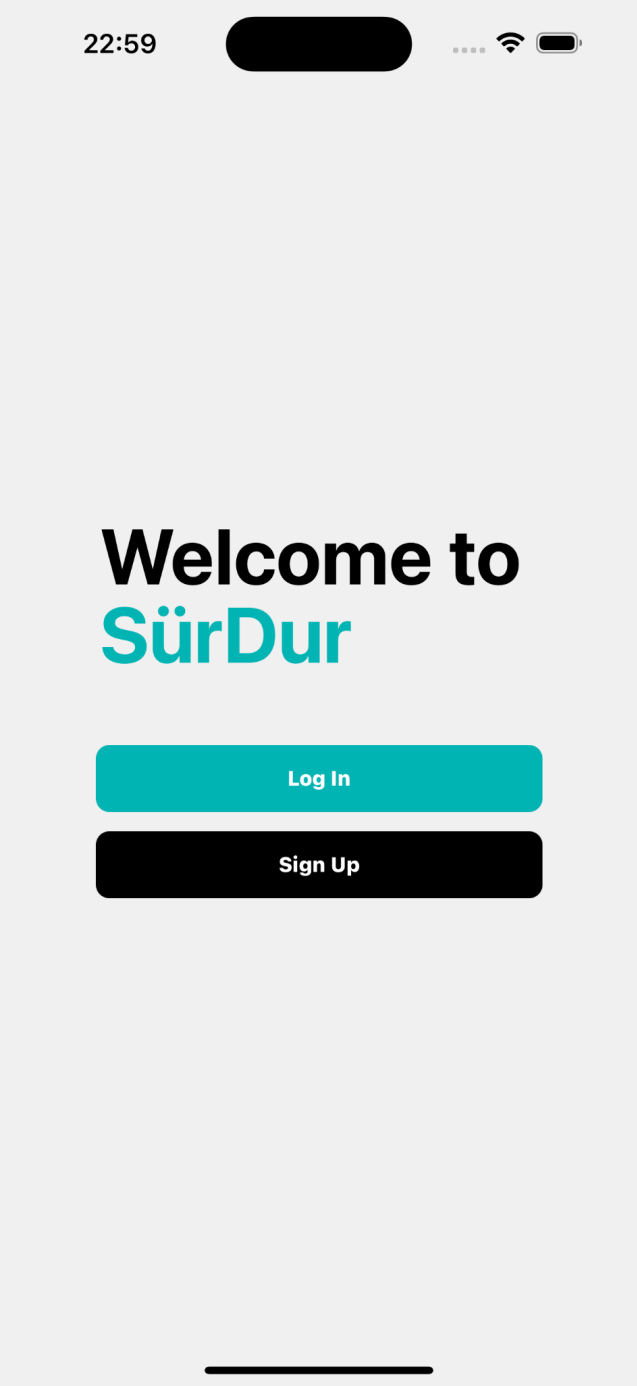

Figure 13: Onboarding Page

Login Page

Login page for SürDur provides returning users with input fields for their email and password to access their account. It includes a visible "Login" button

for account access and a secondary "Go To Sign Up" button to redirect new users to the registration process.

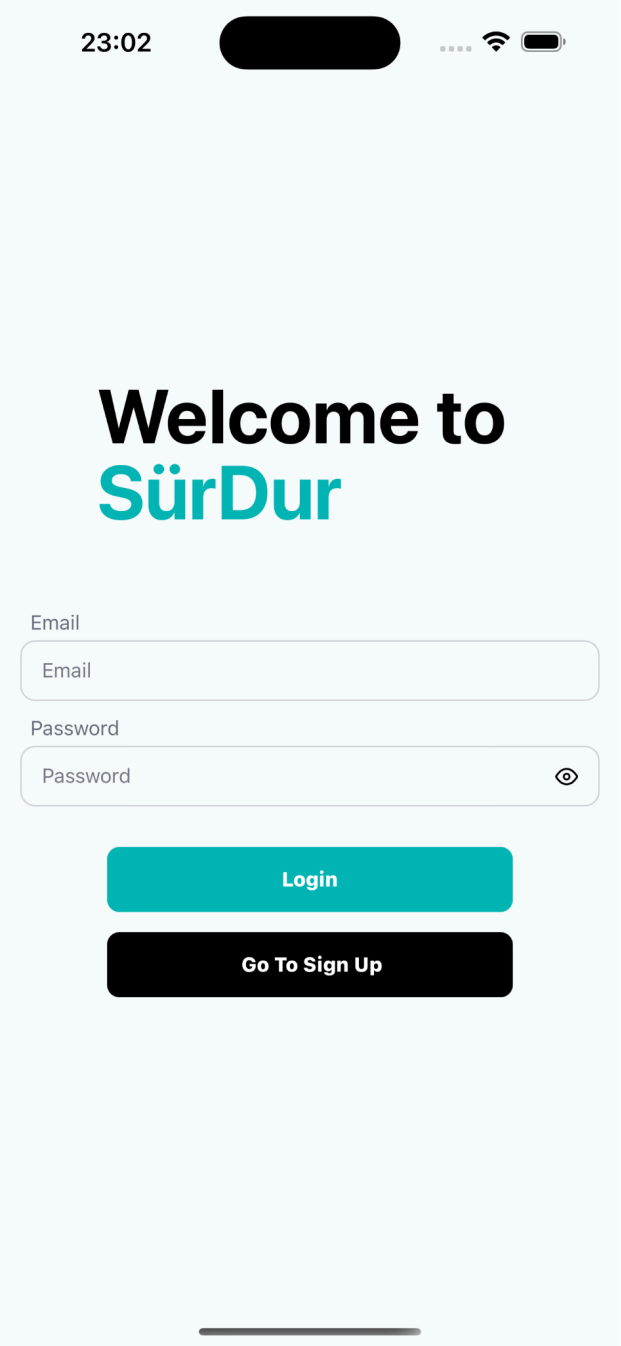

Figure 14: Login Page

Sign-up Page

Sign-up page of SürDur is designed to allow new users to create an account by providing essential information, including email, password (with

confirmation), phone number, date of birth, and country. It features a "Sign Up" button to submit the details and a "Go To Login" button to redirect users who already have an account.

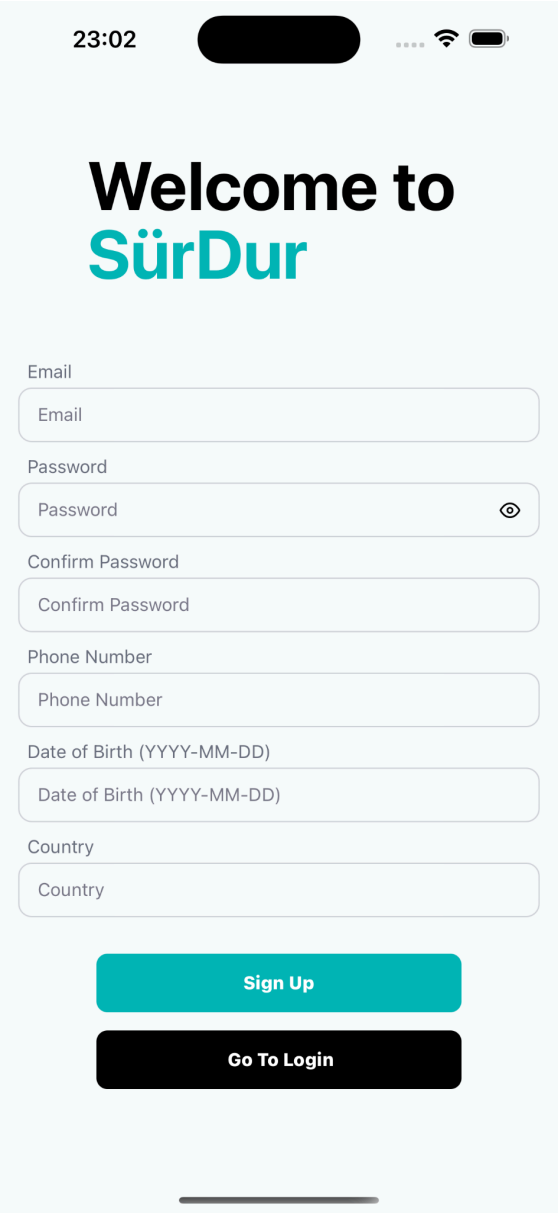

Figure 15: Sign-up Page

Main Page

Main page of the SürDur application, allowing users to explore geographical locations. It includes a search bar for finding specific locations, a profile icon for user settings or account access, and map interaction tools like

zoom or location focus buttons. The interface appears user-friendly, focusing on exploration and ease of navigation.

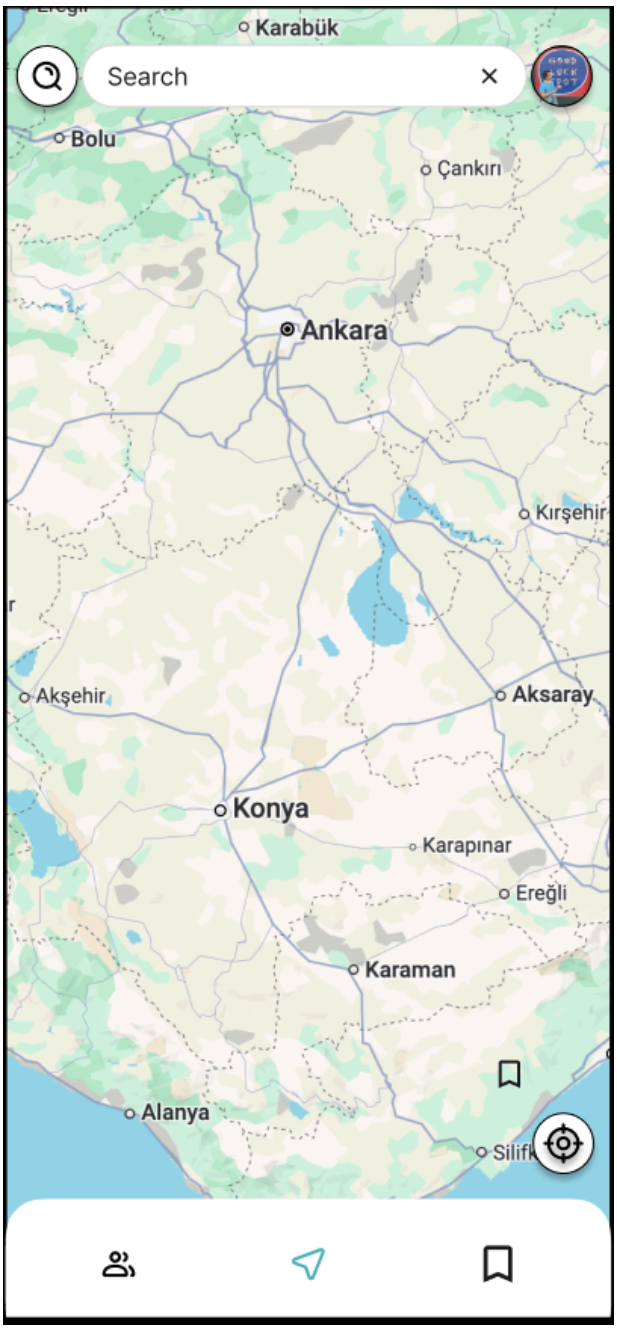

Figure 16: Main Page

Search Screen

Search interface allows users to plan a route or find destinations in the SürDur application by inputting a starting point (Ankara in the example) and a

destination. The suggestions below provide options related to the entered query, such as places or landmarks, each accompanied by navigation icons to explore further.

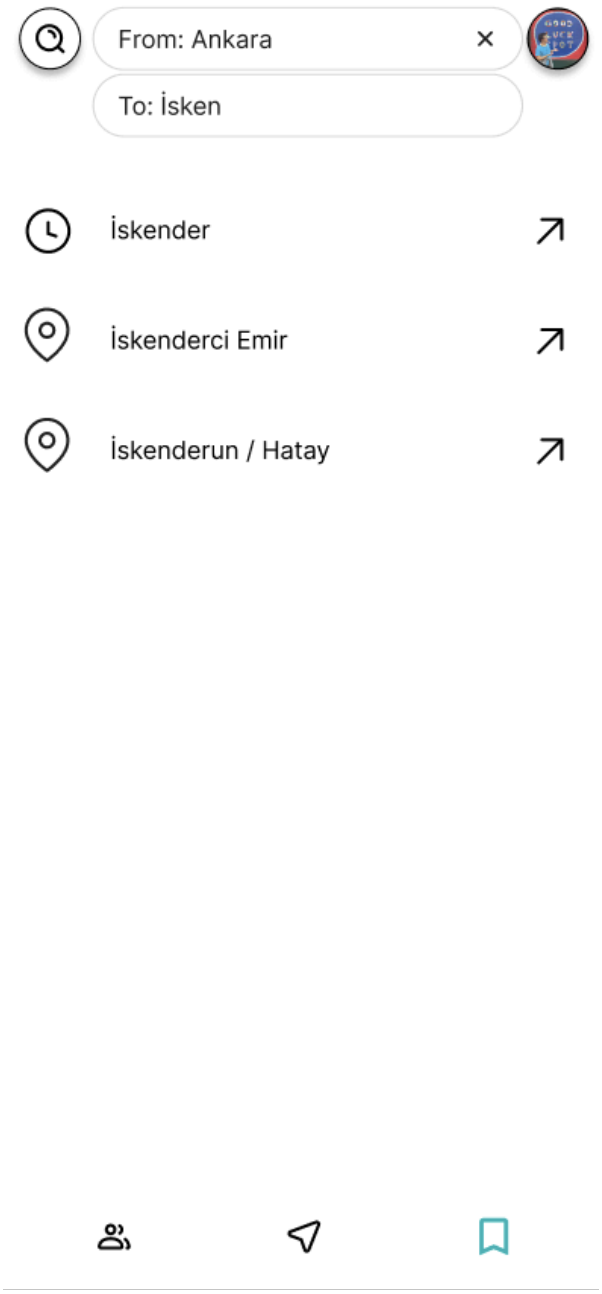

Figure 17: Search Screen

Suggestion Screen

Suggestion screen displays a route and suggested waypoints (pinpoints on the map) along the journey. It features category filters such as "Fuel Station" and "Shopping" to assist users in locating relevant services along the route. Additionally, at the bottom, suggestions for specific places provide personalized or popular destinations that users might want to visit during their trip. In the following example, the user has accepted all four of the suggestions and the route is drawn accordingly. Also, users can see the selected place on the map with the blue highlighted marker.

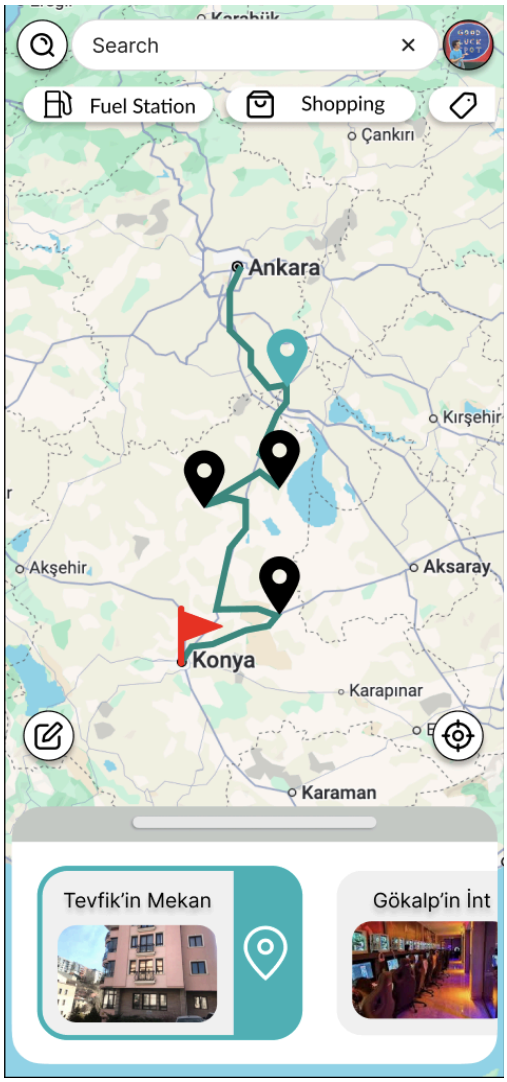

Figure 18: Suggestion Screen

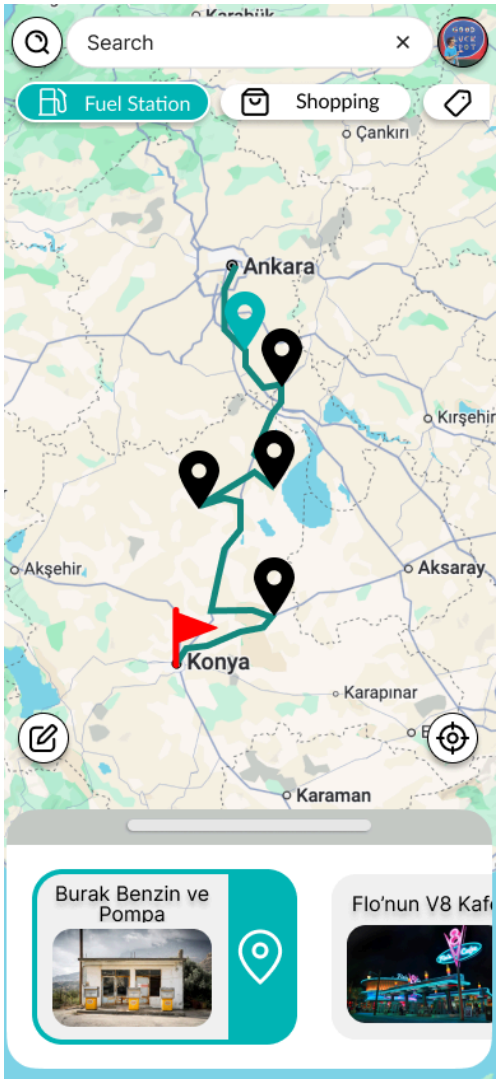

Figure 19: Suggestion with Quick Filter

Navigation Screen

Navigation screen provides turn-by-turn guidance. The map highlights the user's route, with a directional prompt at the top for the next turn. The current location is indicated, and at the bottom, the "NEXT" section highlights the upcoming destination. The design focuses on the safety of our user by minimizing user interactions while driving.

Figure 20: Navigation Screen

Saved Routes Page

The Saved Routes Page is used to display the user's saved routes. Here, the routes are separated into two categories, completed and drafted. Those categories are indicated within the saved routes. Each route has a title and description to make different routes distinguishable. Using quick filters at the top, routes can be seen separately by categories.

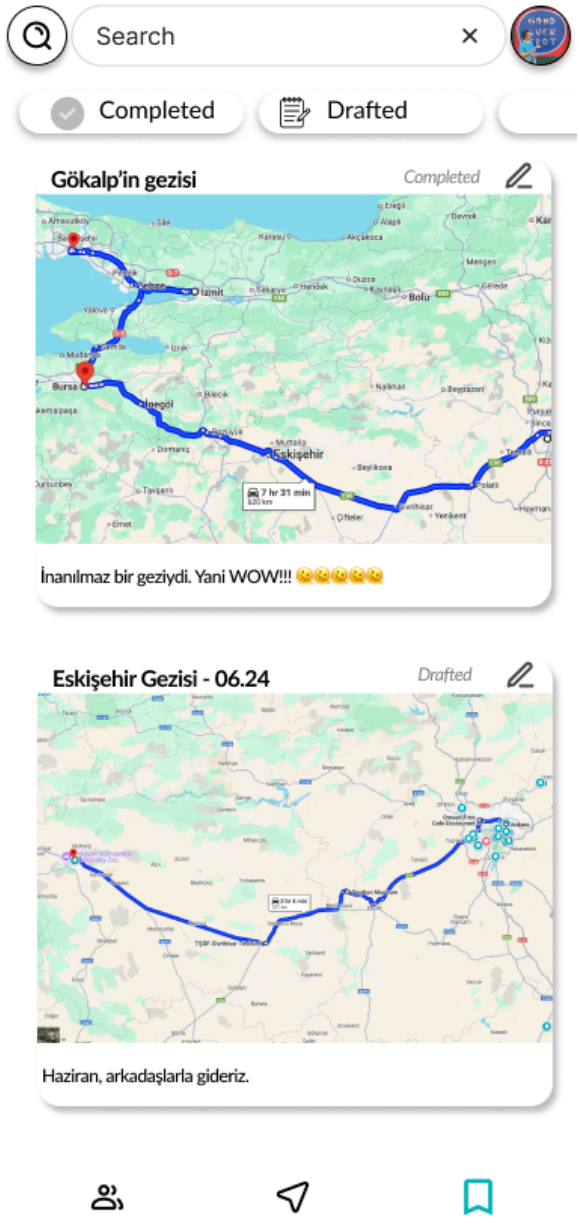

Figure 21: Saved Routes Page

Saved Route Details Page

This is the detailed pop-up view of a saved post entry. The title and the description can be edited, and the log can be marked as “completed”, and can be saved via the “save” button. The route can be shared on the social page via the “share” button. If the user wishes to see the route on map and start navigation, “start route” button should be clicked.

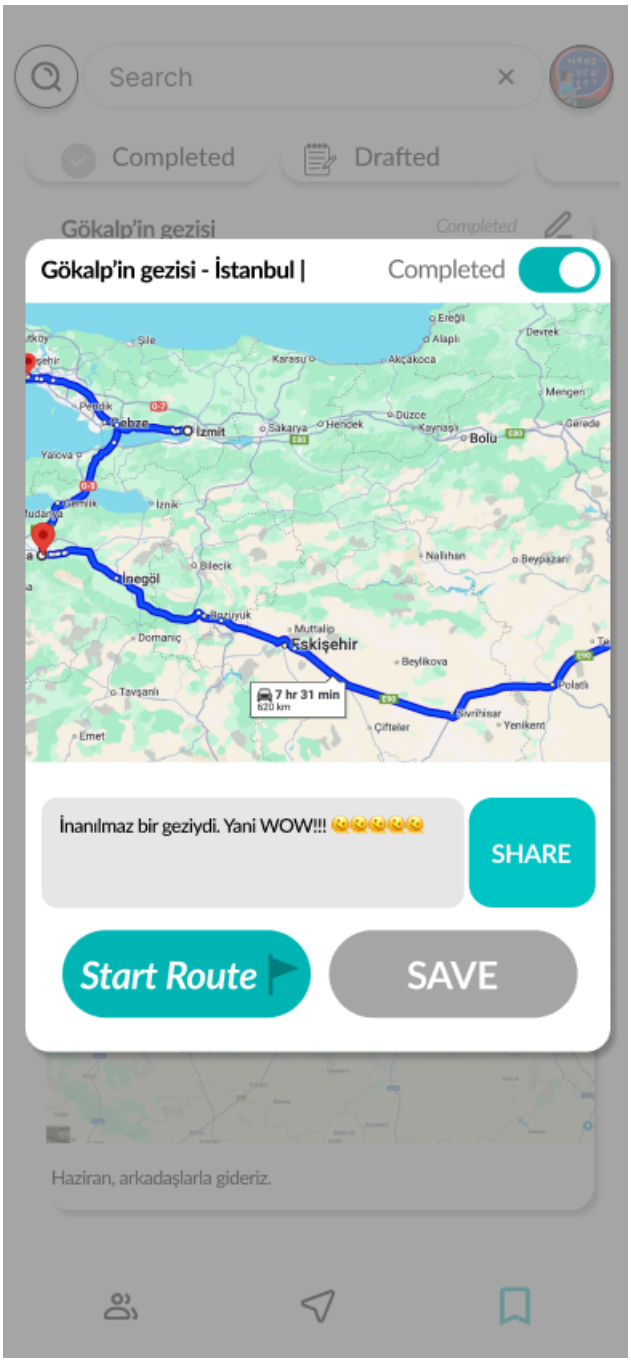

Figure 22: Saved Route Details Page

Social Page

Here a feed of routes from multiple users is displayed. Each post contains a mapped route with a title and description provided by the sharing user. Other users can interact by liking or disliking.

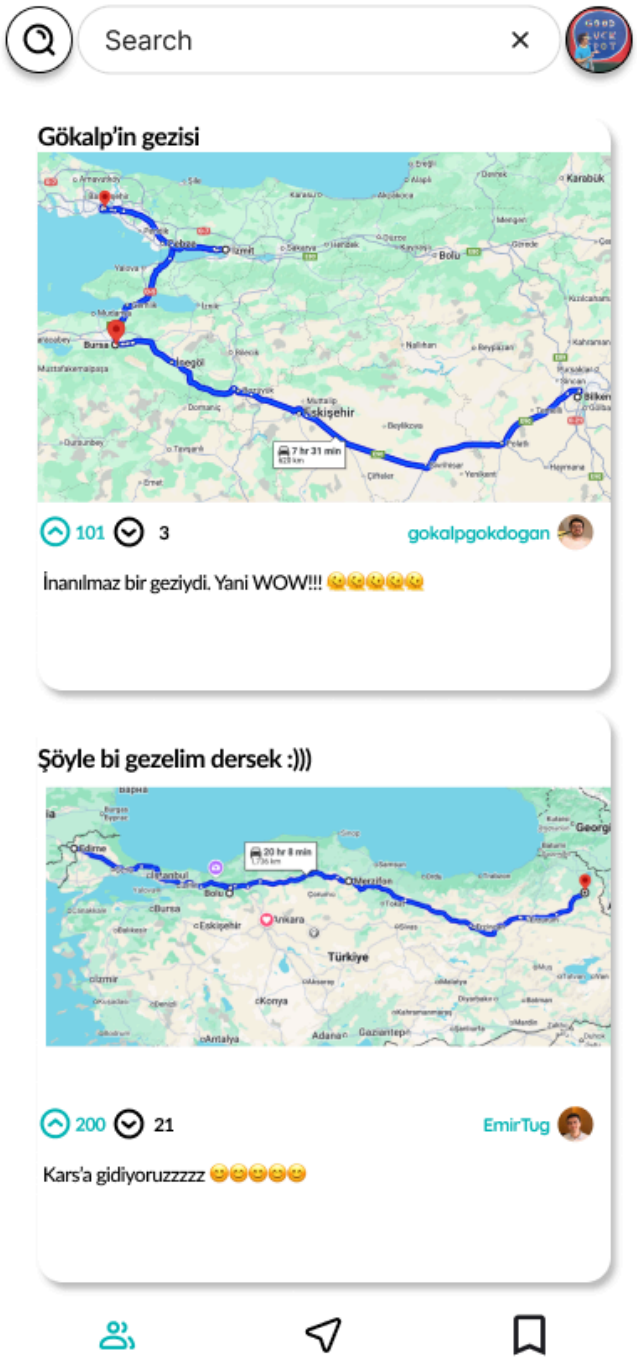

Figure 23: Social Page

Post Details Page

Tapping on a post opens a detailed pop-up view, showing the route on a map. Here the post has all the utilities seen on the social page alongside "See on Map" or "Copy to Log" which are the features for saving or replicating other users' routes.

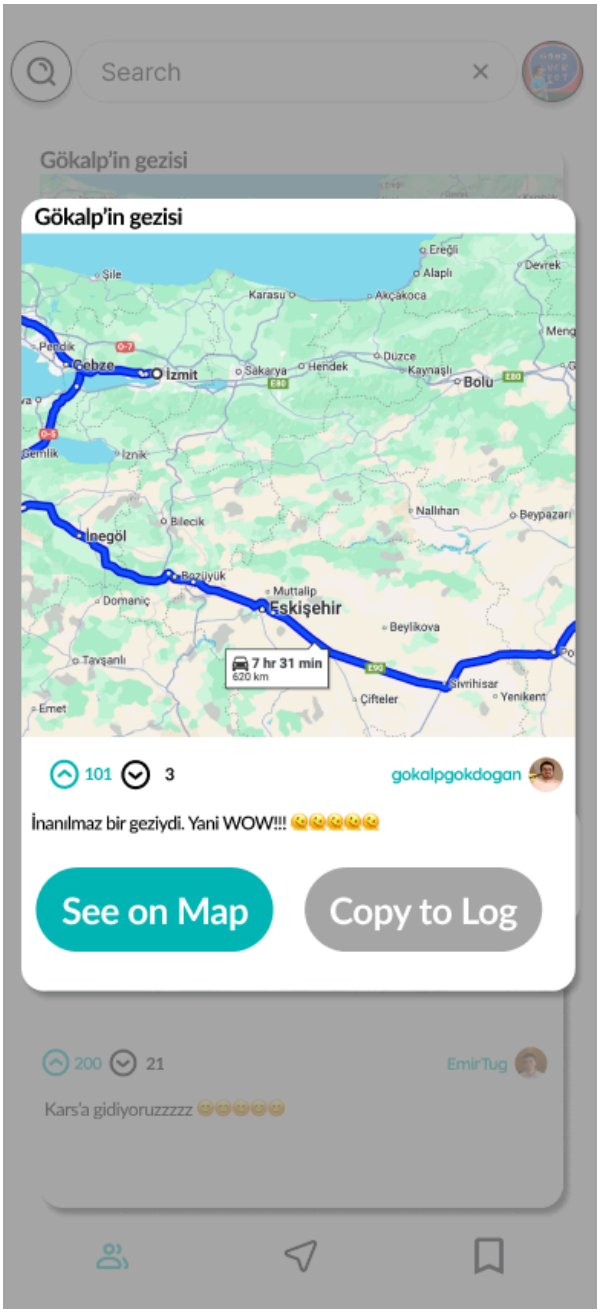

Figure 24: Post Details Page

Profile Page

A user profile consists of key details like profile picture, username, bio, number of posts, followers, and following. It includes a "Follow" button for visiting and "Edit Profile" button for the owner of the profile. Displays recent posts in a similar manner to the social page with the exception of posts belonging only to the related user.

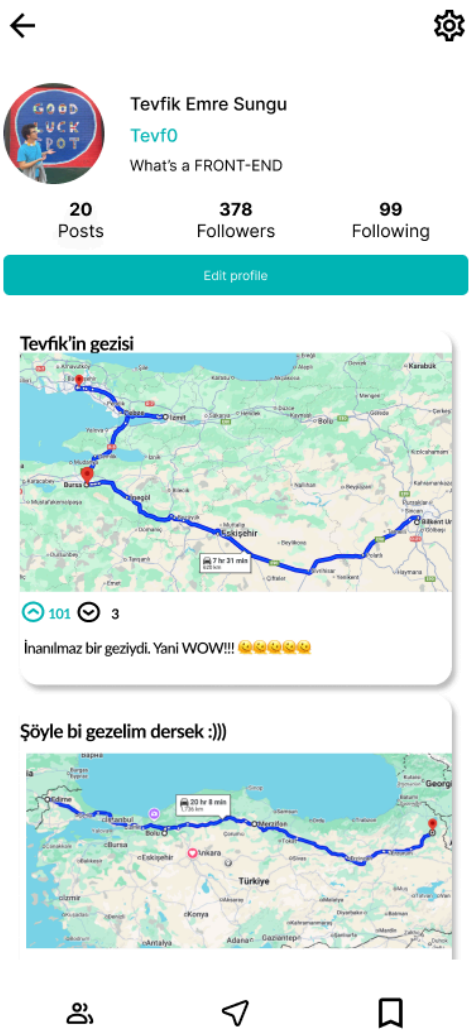

Figure 25: My Profile Page

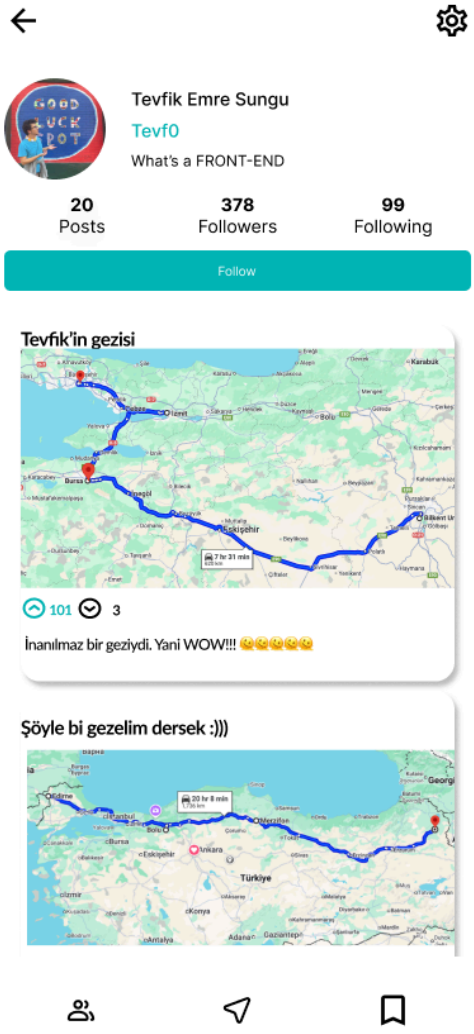

Figure 26: Others' Profile Page

4. Other Analysis Elements

4.1. Consideration of Various Factors in Engineering Design

4.1.1. Constraints

This section will discuss the SürDur project's constraints in detail on aspects of development, economic, technological, social, safety, and sustainability. Also, the effects of global, cultural, social, environmental and economic factors on the app is given in the following table:

	Effect Level	Effect
Global	9	● Take different roads, places into consideration. ● Limitations of sources, passes from one country to another may affect the whole design.
Cultural	9	● Cultural preferences should be taken into consideration in place recommendation. ● Local cultural point of interests should be taken into consideration.
Social	9	● Social post interactions between users should be considered on personal preference loggings. ● Follower-following system should be considered on social content display.
Environmental	5	● Less fuel consumption should be taken into consideration in route generation.
Economic	7	● Budget availability by the user should be taken into consideration in place suggestions. ● Server maintenance fees should be evaluated to decide how large amount of place data the application can hold.

Table 2: Factors that can affect analysis and design

4.1.1.1. Development Constraints

- The project app will be available for both iOS and Android.
- In a broader scope, the project’s UI also will be implemented for Apple CarPlay and Android Auto.
- The mobile side of the project will be developed using React Native as it’s compatible with both IOS and Android.

- The application will be developed with Python and FastAPI for the back end and React-Native for the front end.
- OpenAI API will be used to extract categories of POIs from online blogs by using language models.
- In categorizing POIs, NLP methodologies and tools will be used to help mapping many categories into predetermined categories.
- Git and Github will be used as our version control system.
- MySQL will be used to store and access database components related to both the application engine and the user data.
- The decision engine and the database will be kept on AWS servers.
- Notion will be used to keep track of the development process.

4.1.1.2. Economic Constraints

- The database will be kept on AWS servers. Annual payment for reserving a micro-sized server space (1 GB of data space) from Stockholm servers requires $94 [1].
- Publishing the app on mobile platforms has two economic constraints. One is the $25 one-time registration fee on the Google Play Store (Android), and the other is the annual $99 fee on the App Store (IOS) [2].
- Frameworks and libraries that will be used to implement the project such as FastAPI, React Native, and Expo, are free to use.

4.1.1.3. Technological Constraints

- The application will need an internet connection for all the functionalities such as route creation, navigation, social functions, and profile operations.
- The application has to access the user's location on route creation, and navigation functions.

4.1.1.4. Social Constraints

- The application will allow the sharing of previously followed routes.
- The public route posts will include a title and a header which allows a detailed explanation of the route. However, there is no further text-based communication allowed on those posts.
- The posts have a voting system that shows the public appreciation of users' posts.

4.1.1.5. Safety Constraints

- Mobile app decisions will be made to minimize user interaction during a car ride.
- The user will be asked to make choices before starting the ride.
- The live navigation service of SürDur will be designed primarily concerning driver safety. Therefore, the effect level of safety is **10** out of 10.

4.1.1.6. Sustainability Constraints

- Server and publishing services need to be paid annually.
- An increase in the number of users may result in database enlargement, which will result in higher server space costs.

4.1.2. Standards

4.1.2.1. IEEE 1471

Purpose:

IEEE 1471, a software-intensive system architecture documentation standard, is ISO/IEC/IEEE 42010. It outlines developing an architecture description that offers a shared comprehension of the system's structure, functionality, and essential characteristics [8]. We can more easily comprehend system components and their relationships thanks to IEEE 1471's assistance in clarifying the architecture. This entails outlining the architecture's background, perspectives, interested parties, and the reasoning behind essential choices [3].

Key Elements:

We document architectural choices in development; this outlines essential decisions made during the design process, supporting information, and other factors. Also, documenting architectural views in the project enables us to represent the project's physical, process, development, and logical aspects.

4.1.2.2. UML 2.5.1

Purpose

A widely used modeling language for describing, building, visualizing, and recording the structure and behavior of software systems is UML 2.5.1 [9]. It provides a consistent method for drawing diagrams that explain various system

components. We can understandably display the system's structural and functional elements using UML. This standard facilitates the creation of models for different views (such as class, sequence, and activity diagrams), which helps with system design and communication [4].

Key Aspects

Class, Component, and Deployment aspects define the system's static structure. Use Case, Sequence, and Activity represent dynamic aspects of the system, including interactions and workflows.

4.1.2.3. IEEE 830

Purpose

Writing Software Requirements Specifications (SRS) is standardized by IEEE 830. It establishes a thorough framework for recording functional and non-functional requirements, guaranteeing accuracy, consistency, and comprehensiveness [10]. IEEE 830 offers a systematic style for specs reports that assists teams in organizing requirements for easy understanding and verification by stakeholders, developers, and testers. Project objectives, scope, requirements, assumptions, and restrictions are all covered in this standard [5].

Key Aspects

This standard addresses the project's background, goal, and extent. Additionally, it gives a summary of the operating environment, user attributes, and product capabilities.

4.1.2.4. ISO 31000

Purpose

One standard that offers recommendations for efficient risk management is ISO 31000. It aids businesses in recognizing, evaluating, and reducing risks, which enhances decision-making and reduces uncertainty [11]. This standard exemplifies proactive risk management by addressing potential project risks (technical, operational, and financial) and mitigation techniques. Risk assessment frameworks, prioritization, and controls are a few examples [6].

Key Aspects

Identifying potential risks that have an impact on the project. Assessing the impact and probability of hazards that have been discovered. establishing measures to reduce or eliminate risks. We can identify possible problems and dangers by implementing risk management.

4.1.3. IEEE Citation Style

Purpose

IEEE Citation Style is a widely standardized approach for citing sources from engineering, information technology, and allied fields. It increases the traceability and dependability of the information by ensuring that sources are consistently mentioned. IEEE Citation Style provides a uniform method of referring to external sources (including research papers, technical publications, and standards) that ensures accuracy and lucidity. When citations are appropriately formatted, readers may locate sources for further context and proof [7].

Key Aspects

References match the list of references and are numbered in brackets (e.g., [1], [2])[12]. provides comprehensive information for every source and arranges citations in numerical order.

4.2. Risks and Alternatives

Most possible risk factors, their likelihood, their overall effect and their Plan B summary is given below:

	Likelihood	Effect on the project	B Plan Summary
Delays in third party API's	High	Low user performance by late response time of the app	● Requests will be sent to alternative route formation API's. ● Current cached map and location information will be used for live navigation.
Data loss and privacy concerns	Low	Total app may crash and reliability of the application will reduce.	● Periodically updated backup database will be activated. ● Whole data storage structure will be transformed.

Lack of data source	Medium	Less accurate and personal route recommendations will be given.	● Reduce the total feature number stored. ● Reduce the suggested place amount, but in a high variant manner.
Data feature mismatch between multiple sources	High	Shrinking of total features list or place list.	● Relevance levels between mismatched features will be evaluated. ● Based on the observations, one of the features will be selected to represent both, or one of the sources will not be used anymore.

Table 3: Risks

4.2.1. Delays in third-party API's

- SürDur application relies on multiple external third-party API's for both route recommendation and live navigation.
- OpenRouteService API will be used for route formation and Google Maps API will be used for live navigation management and display.
- The likelihood for any possible delay on any possible third-party API will be considered high, as these delays may occur periodically with high consequences as our app relies majorly on these services.
- Alternative plans will be explained separately for initial route formation and live navigation.
    - Dynamic route formation based on user selections majorly relies on the response from open source OpenRouteService API. Alternative third-party route formation API's will be pinged by numerous requests with the same purpose of route formation. If no response has been obtained from any alternative services, a kindly warning message of 'Please try again' will be displayed.
    - Live navigation will be handled with Google Maps API throughout the roadtrip. Any possible delay on this service will be handled by using only the navigation information of the user and the locally cached map to keep track of the road for the driver.

4.2.2. Data loss and privacy concerns

- SürDur requires permission for current location, in order to execute the core functionalities of recommending route planning and live navigation.

- SürDur requires to keep track of the personal preferences of each user based on past routes and social media interactions.
- Personal information stored and fetched by the application should be stored and executed safely for potential information leakage (e.g. using encryption for data storage).
- The likelihood of any sort of information leakage and data loss is relatively low, as the potential hosting and storage platforms are highly-established and commonly-used platforms (Microsoft Azure, AWS etc.).
- The alternative plan is to recover potential data loss from our periodically-updated backup storage. Then each user should be informed of any potential data loss and they should be able to reach out to the staff in any case of data loss.
- Alternatively, if any form of information leakage occurs, the structure of the leaked data will be converted to a new structure which would make the previously encrypted data inadmissible.

4.2.3. Lack of data source

- SürDur mainly relies on a high variety of place data for the most suitable route recommendation. This demand requires plenty of data sources that should be combined to one large database.
- The lack of data sources most likely to conclude with less accurate and personal route recommendation across the available map.
- The likelihood of collecting an insufficient amount of data is relatively medium. Because, the sufficient amount of place data will be mostly determined by the accuracy of the AI-powered recommendation system itself and the sufficient amount is unpredictable at the moment.
- The alternative plan is to reduce the total feature number for user preferences and recommend less places in order to keep the variety of choices proportional to the total amount of places available.

4.2.4. Data feature mismatch between multiple sources

- SürDur stands out from its competitors with the idea of merging multiple sources to form one single route recommendation service.

- Each external travelling API service contains a different features list for their place database. One single features list should be formed to represent all of the multiple sources.
- Any unavoidable feature misalignment between any two data sources may end up with shrinking the total features list, or maybe removing one of the sources which may damage the precision and the variety of the recommendations.
- The likelihood of any feature mismatch between sources is high. It is mostly likely to have misalignments between two sources, since there is no universal features list for all travelling and place data sources.
- The action plan is to evaluate the relevance levels of mismatched features using language processing techniques to determine whether one of the features can be used to represent both. If not, one of the sources will not be used for further data collection.

4.3. Project Plan

WP#	Work package title	Leader
WP1	Project Specification Document	Gökalp Gökdoğan
WP2	Analysis and Requirement Report	Tevfik Emre Sungur
WP3	Frontend Development	Gökalp Gökdoğan
WP4	Backend Development	Burak Oruk
WP5	Setting up the Database	Tevfik Emre Sungur
WP6	Recommendation System Development	Emir Tuğlu
WP7	Demo	Bora Haliloğlu
WP8	Detailed Design Report	Emir Tuğlu
WP9	Design Project Final Report	Bora Haliloğlu
WP10	App Launch	Burak Oruk
WP11	Final Demo	Gökalp Gökdoğan

WP1: Project Specification Document			
Start Date: 12 November 2024 End Date: 22 November 2024			
Leader	Gökalp Gökdoğan	Members Involved	All Members
Objectives: Prepare and deliver the Project Specification Document.			
Tasks: **Task 1.1 Writing an Introduction:** Describe the project in detail. Describe the type of innovation that is being sought. Identify the limitations and ethical and professional concerns. **Task 1.2 Writing Requirements:** Describe the functional and non-functional requirements in your writing. **Task 1.3 Writing Ongoing Discussions:** Provide information on any ambiguities in the project's specifics and outline potential future directions. **Task 1.4 Writing References:** Use the proper citation formats and include references for all sources used in the report.			
Deliverables: **D1.1:** Project Specification Document			

WP2: Analysis and Requirement Report			
Start Date: 3 December 2024 End Date: 16 December 2024			
Leader	Tevfik Emre Sungur	Members Involved	All Members
Objectives: Prepare and deliver the Analysis and Requirement Report.			
Tasks: **Task 2.1 Scenarios** **Task 2.2 Creation of Use-Case Diagram** **Task 2.3 Creation of Object and Class Model** **Task 2.4 Creation of Dynamic Models:** Create Activity, Sequence, and State Diagrams. **Task 2.5 Creation of UI Designs** **Task 2.6 Other Analysis Elements:** Identify the options and hazards, and describe the elements that influenced the design. Additionally, describe the professional and ethical obligations. Provide a thorough project strategy as well as a road map for gaining the technical know-how required for the future. **Task 2.7 References:** Use the proper citation formats and include references for all sources used in the report.			
Deliverables: **D2.1:** Analysis and Requirement Report			

WP3. Frontend Development			
Start Date: 29 November 2024 End Date: May 2025			
Leader	Gökalp Gökdoğan	Members Involved	Bora Haliloğlu Gökalp Gökdoğan Emir Tuğlu
Objectives: Implementation of the front-end of the application according to the UI Designs created for the Analysis Report.			
Tasks: **Task 3.1 Implementation of Log-in & Sign-up pages** **Task 3.2 Implementation of the Onboarding Page** **Task 3.3 Implementation of the Main Page** **Task 3.4 Implementation of Suggestion Page** **Task 3.5 Implementation of the Navigation Page** **Task 3.6 Implementation of the Search Page** **Task 3.7 Connect front-end to back-end** **Task 3.8 Optimizing performance of application**			
Deliverables: **D3.1:** The Frontend of the app.			

WP4: Backend Development			
Start Date:16 November 2024 End Date: May 2025			
Leader	Burak Oruk	Members Involved	Tevfik Emre Sungur Burak Oruk Emir Tuğlu
Objectives:Implementation of the back-end of the application according to the design proposed in the Analysis Document.			
Tasks: **Task 2.1: Dockerizing the back-end application** **Task 2.2: Initializing FastAPI project with correct dependencies** **Task 2.3: Implementation of basic classes according to class diagram** **Task 2.4: Implementing service layer** **Task 2.5: Connecting the external services to service classes** **Task 2.6: Testing controller endpoints via postman** **Task 2.7: Connecting the back-end with front-end** **Task 2.8: Deployment to AWS by Dockerizing the app**			

Deliverables: **D2.1:** The back-end application **D2.2:** Docker container

WP5: Setting up the Database			
Start Date:16 November 2024 End Date: February 2025			
Leader	Tevfik Emre Sungur	Members Involved	Tevfik Emre Sungur
Objectives:Designing and creating a database that can store high volume of location data and allows low latency data retrieval.			
Tasks: **Task 5.1: Design the database schema** **Task 5.2: Create tables according to the design** **Task 5.3: Collect and standardize data from different APIs and blogs**: Write scripts to collect data from the POI APIs and blogs. Then, standardize this data to the same format and eliminate duplicate data before storing in the database. **Task 5.4: Populate tables with the retrieved data:** Save collected and standardized data into the database.			
Deliverables: **D5.1:** The database that contains POI information.			

WP6: Recommendation System Development			
Start Date: January 2025 End Date: May 2025			
Leader	Emir Tuğlu	Members Involved	Tevfik Emre Sungur Burak Oruk Gökalp Gökdoğan Emir Tuğlu
Objectives:Developing a recommendation system algorithm to provide users personalized POI recommendations.			
Tasks: **Task 6.1 Define POI Labels:** Each API has its own labels for POIs. We need our own set of labels and map labels of different APIs to our labels. **Task 6.2 Assign weights to POIs in the database:** Each POI in the database should have a weight value for each of our labels. **Task 6.3 Implement the algorithm to update user weights:** Each user should also have a weight value for each label. These weights should be			

updated according to user's choices and labels of their previously visited POIs
Deliverables: **D6.1:** The recommendation system that provides personalized recommendations according to users' preferences.

WP7: Demo			
Start Date: 16 December 2024 End Date: 20 December 2024			
Leader	Bora Haliloğlu	Members Involved	All Members
Objectives: Prepare and deliver the Demo			
Tasks: **Task 7.1 Prepare Slides:** Prepare slides about the project, the problem that project solves, market and competitor analysis, business model, etc. **Task 7.2 Prepare Demo:** Prepare a demo to display implemented functionality of the system. **Task 7.3 Present:**			
Deliverables: **D7.1:** Demo			

WP8: Detailed Design Report			
Start Date: February 2025 End Date: March 2025			
Leader	Emir Tuğlu	Members Involved	All Members
Objectives: Prepare and deliver the Detailed Design Report			
Tasks: **Task 8.1 Determine design goals:** Usability, performance, reliability, marketability, etc. **Task 8.2 Sketch the architecture of the system** **Task 8.3 Explain subsystem services** **Task 8.4 Define functional and non-functional test cases** **Task 8.5 Discuss teamwork details**			
Deliverables:D8.1: Detailed Design Report			

WP9: Design Project Final Report			
Start Date: April 2025 End Date:May 2025			
Leader	Bora Haliloğlu	Members Involved	All Members
Objectives: Prepare and deliver the Design Project Final Report			
Tasks: **Task 9.1 Write down requirements details:** **Task 9.2 Sketch the final architecture** **Task 9.3 Provide development and implementation details** **Task 9.4 Give information about test cases and results** **Task 9.5 Discuss maintenance plan** **Task 9.6 Discuss other project elements:** Constraints, standards, ethics and professional responsibilities, teamwork details etc.			
Deliverables: **D9.1:** Design Project Final Report			

WP10: App Launch			
Start Date: May 2025 End Date: May 2025			
Leader	Burak Oruk	Members Involved	All Members
Objectives: Launch the app			
Tasks: **Task 10.1 Test the app:** Ensure each functionality is working as expected **Task 10.2 Launch the app on the App Store** **Task 10.3 Launch the app on the Play Store**			
Deliverables: **D10.1:** The app that can be downloaded by iOS and Android devices			

WP11: Final Demo			
Start Date: May 2025 End Date: May 2025			
Leader	Gökalp Gökdoğan	Members Involved	All Members
Objectives: Prepare and deliver the Final Demo			
Tasks:			

Task 11.1 Prepare Slides: Prepare slides about the project, the problem that project solves, market and competitor analysis, business model, etc. **Task 11.2 Prepare Demo:** Prepare a demo in which functionalities of the app are displayed. **Task 11.3 Present**
Deliverables: **D1.1:** Final Demo

4.4. Ensuring Proper Teamwork

- Each group member has and will have an equal say in any matter throughout and further beyond the development process.
- In times when there is no consensus, the matter will be discussed professionally until the discussion converges to a mutual conclusion.
- Weekly progress meetings will be held among group members. Logs will be recorded to track the overall process and the performance of the group members.
- An equal workload will be put on each group member. This will be achieved by taking into consideration every member's schedule separately.
- Notion is used assign tasks to each member separately, and to keep track of overall progress.

4.5. Ethics and Professional Responsibilities

4.5.1. Professional Responsibilities

- To ensure transparency and inclusivity, end users will be kept informed throughout the project's lifecycle. This will be achieved through surveys and notifications regarding any design changes or finalizations. Their valuable feedback will be actively incorporated into the development process.
- A demo version of the application will be presented to end users to collect their opinions and tailor the app to align with market expectations.
- The construction of the database will prioritize equal representation across all cities and regions of Turkey. However, due to the natural non-uniform distribution of human-made stops such as restaurants, the density of solutions may vary depending on the region.

4.5.2. Ethical Responsibilities

- User geolocation data will strictly be used for generating routes and navigation purposes. This data will neither be stored nor shared under any circumstances.
- Sensitive personal information, such as recommendation details and past trip data, will remain confidential and inaccessible to third parties.
- Publicly visible information, including name, surname, profile photo, and shared routes, will only be displayed within the app interface. Outside of the app, this information will be safeguarded to ensure user privacy.

4.6. Planning for New Knowledge and Learning Strategies

- The team is currently in the process of deciding the main hosting service in order to configure and connect all components of the app.The team will decide between Microsoft Azure or AWS services mainly, or any other candidates. Additional hosting service research should be conducted based on price, performance, and robustness.
- The team will build the foundation for the feature extraction from multiple data sources and merging them into one large database. This foundation requires advanced data preprocessing, feature selection steps, multiple NLP applications (e.g. sentiment analysis, TF-IDF). Multiple academic sources and previous notebook applications will be traversed and the most suitable combination of applications will be formed.
- The team will be focusing on mastering several main third-party location and mapping API services, such as OpenRouteService and Google Maps API. Each documentation will be analyzed further along with the conduction of several test cases with several scenarios.
- The team will conduct extended market research on data source extraction. More reliable place information portals and travelling related blog portals will be found and merged with previously found sources.

5. Glossary

UI: User Interface.

POI: Point of Interest. Locations to be recommended to the user.

API: Application Programming Interface. An interface that a software provides for the use of another software.

NLP: Natural Language Processing. A subfield of computer science and artificial intelligence (AI) that uses machine learning to enable computers to understand and communicate with human language. [14]

6. References

[1] Amazon Web Services, "Amazon RDS for MySQL pricing," [Online]. Available: https://aws.amazon.com/tr/rds/mysql/pricing/?pg=pr&loc=2. [Accessed: Nov. 19, 2024].

[2] Sphinx Solution, "Cost to put an app on the App Store," [Online]. Available: https://www.sphinx-solution.com/blog/cost-to-put-an-app-on-the-app-store/. [Accessed: Nov. 19, 2024].

[3] IEEE, "IEEE Standard 1471: Recommended practice for architectural description of software-intensive systems," [Online]. Available: https://standards.ieee.org/ieee/1471/2187/. [Accessed: Nov. 19, 2024].

[4] Object Management Group, "Unified Modeling Language (UML), version 2.5.1," [Online]. Available: https://www.omg.org/spec/UML/2.5.1/About-UML/. [Accessed: Nov. 19, 2024].

[5] G. Booch, J. Rumbaugh, and I. Jacobson, "The Unified Modeling Language user guide," IEEE, 1999. [Online]. Available: https://ieeexplore.ieee.org/document/720574. [Accessed: Nov. 19, 2024].

[6] International Organization for Standardization, "ISO 31000: Risk management," [Online]. Available: https://www.iso.org/iso-31000-risk-management.html/. [Accessed: Nov. 19, 2024].

[7] New Jersey Institute of Technology, "IEEE citation style guide," [Online]. Available: https://researchguides.njit.edu/ieee-citation/ieeereferencing/. [Accessed: Nov. 19, 2024].

[8] W. Pree, "Design patterns for object-oriented software development," IEEE, 1995. [Online]. Available: https://ieeexplore.ieee.org/document/875998/. [Accessed: Nov. 19, 2024].

[9] UML Diagrams, "UML 2.5 diagrams overview," [Online]. Available: https://www.uml-diagrams.org/uml-25-diagrams.html. [Accessed: Nov. 19, 2024].

[10] IEEE, "IEEE Standard 830: Recommended practice for software requirements specifications," [Online]. Available: https://standards.ieee.org/ieee/830/1222/. [Accessed: Nov. 19, 2024].

[11] International Organization for Standardization, "ISO 9001: Quality management systems," [Online]. Available: https://scc.isolutions.iso.org/obp/ui#iso:pub:PUB100464. [Accessed: Nov. 19, 2024].

[12] George Mason University, "IEEE style citation guide," [Online]. Available: https://infoguides.gmu.edu/ieee_style#s-lg-box-29326431. [Accessed: Nov. 19, 2024].

[13] “Content-based Filtering | Recommendation Systems,” Google Developers. https://developers.google.com/machine-learning/recommendation/content-based/basics. [Accessed: Nov. 21, 2024].
[14] IBM, “What is NLP (Natural Language Processing)?,” IBM, https://www.ibm.com/topics/natural-language-processing (Accessed Dec. 15, 2024).