PLANADAY : SUGGEST ONE DAY TRIP APPLICATION

MR. SITIPORN WIMOLPUNYAKUL

MS. KANYARANT PREMPRAPAPONG

MS. THANAKORN CHOTTHANIGARN

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF BACHELOR OF SCIENCE (COMPUTER SCIENCE)

SCHOOL OF INFORMATION TECHNOLOGY

KING MONGKUT'S UNIVERSITY OF TECHNOLOGY THONBURI

2024

PLANADAY : SUGGEST ONE DAY TRIP APPLICATION

MR. SITIPORN WIMOLPUNYAKUL

MS. KANYARANT PREMPRAPAPONG

MS. THANAKORN CHOTTHANIGARN

A PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF BACHELOR OF SCIENCE (COMPUTER SCIENCE)

SCHOOL OF INFORMATION TECHNOLOGY

KING MONGKUT'S UNIVERSITY OF TECHNOLOGY THONBURI

2024

PLANADAY : SUGGEST ONE DAY TRIP APPLICATION

Mr. Sitiporn Wimolpunyakul

MS. Kanyarant Premprapapong

MS. Thanakorn Chotthanigarn

A Project Submitted in Partial Fulfillment of the Requirements for

The Degree of Bachelor of Science (Computer Science)

School of Information Technology

King Mongkut's University of Technology Thonburi

Academic Year 2024

Project Committee

..... Advisor

(Asst.Prof. Worarat Krathu)

..... Committee

(Dr. Watanyoo Suksa-ngiam)

..... Committee

(Dr. Vithida Chongsuphajaisiddhi)

Project Title PLANADAY : SUGGEST ONE DAY TRIP APPLICATION
Project Credits 6
Candidate Mr. Sitiporn Wimolpunyakul
MS. Kanyarant Premprapapong
MS. Thanakorn Chotthanigarn
Project Advisor Asst.Prof. Worarat Krathu
Program Bachelor of Science
Field of Study Computer Science
Faculty School of Information Technology
Academic Year 2024

Abstract

PlanADay is a mobile application designed to generate personalized one-day trip plans with minimal effort based on user preferences and input data. The application utilizes two recommendation strategies: Most-Related Places From Location Area and Based-on-Preferences Interests. Users can specify details such as their categories of interest (e.g., gyms, parks, cafes, restaurants, museums, theaters, or art galleries), the starting date and time, preferred location area, and the number of places to visit. Additionally, PlanADay allows users to view detailed attraction information, bookmark plans, and access their plan history.

This project demonstrates the practicality of implementing a plan suggestion system by leveraging user inputs to tailor recommendations. The app provides an intuitive and engaging experience, enabling users to efficiently discover and plan their ideal day while enhancing satisfaction and utility.

Keywords : Plan/ Interests / Mobile Application / Suggestion

Acknowledgement

We received assistance and guidance from several respected persons, who deserve our deepest gratitude. This project would not have been feasible without the support of Asst.Prof.Dr. Worarat Krathu contributed invaluable insight and experience.

In addition, we would also like to express our appreciation to Mr. Thanatat Wongabut and Ms. Sukanya Chinwicha, and Mr. Ukrit Ruckcharti who generously contributed their time and expertise to our project.

Contents

CHAPTER	PAGE
ABSTRACT	**iii**
ACKNOWLEDGEMENT	**iv**
CONTENTS	**v**
LIST OF TABLES	**vii**
LIST OF FIGURES	**viii**
Chapter 1 Introduction	**1**
1.1 Background	1
1.2 Objectivs	1
1.3 Scope	1
1.4 Expected Benefits	2
Chapter 2 Feasibility	**3**
2.1 Introduction	3
2.2 Problem Statement	3
2.3 Related research and projects	3
2.4 Requirement specifications for the new system	4
2.4.1 Function requirements	4
2.4.2 Data requirements	4
2.4.3 System requirements	4
2.5 Implementation techniques	5
2.5.1 Frontend	5
2.5.2 Backend	5
2.5.3 Infrastructure	5
2.5.4 Other	6
2.6 Implementation Plan	7
Chapter 3 Analysis and Design	**8**
3.1 Introduction	8
3.2 Analysis of the existing system	8

Contents(CONT.)

CHAPTER **PAGE**

3.3 User requirement analysis 9

3.4 System Design..... 10

3.4.1 Use case diagram 10

3.4.2 Context diagram..... 11

3.4.3 Activity Diagram 12

3.4.4 User Interface Design..... 14

3.5 Database design..... 25

3.5.1 Relational Database..... 25

3.5.2 non-relational database..... 26

Chapter 4 System functionality 27

4.1 Introduction..... 27

4.2 4.2 System architecture..... 27

4.3 4.3 Test Plan..... 27

4.4 Test plan and test result 34

Chapter 5 Summary and suggestions..... 36

5.1 Introduction..... 36

5.2 Project summary..... 36

5.3 Problems encountered and solutions..... 36

5.4 Suggestions for further development 37

5.4.1 Integration of AI in our system 37

5.4.2 Responsive design and implementation..... 37

5.4.3 High coupling of the data 37

REFERRENCES..... 37

List of Tables

Table **PAGE**

Table 2.1 Table of the project task 7

Table 3.1 Table of the feature comparisons 9

Table 4.1 Table 4-1. Test Plan 35

List of Figures

Figure **PAGE**

Figure 3.1 Use case diagram..... 10
Figure 3.2 Context diagram..... 11
Figure 3.3 Activity Diagram of Authentication Process..... 12
Figure 3.4 Activity Diagram of Generate Plan Process..... 13
Figure 3.5 Login Page..... 14
Figure 3.6 Register Page..... 15
Figure 3.7 Persona Page..... 16
Figure 3.8 Home Page..... 17
Figure 3.9 Home Page..... 18
Figure 3.10 Generated Page..... 19
Figure 3.11 Place Detail Page..... 20
Figure 3.12 Edit Plan Page..... 21
Figure 3.13 Suggest Plan Page..... 22
Figure 3.14 Profile Plan Page..... 23
Figure 3.15 History Page..... 24
Figure 3.16 Relational Database Schema..... 25
Figure 3.17 Non-Relational Database Schema..... 26
Figure 4.1 System architecture..... 27
Figure 4.2 Welcome screen..... 28
Figure 4.3 Home screen..... 29
Figure 4.4 Choosing destination screen..... 30
Figure 4.5 Choosing route screen..... 31
Figure 4.6 Route options..... 32
Figure 4.7 Selected route detail screen..... 33
Figure 4.8 Real-time navigation screen..... 34

Chapter 1

Introduction

1.1 Background

In today's fast-paced world, people rely more on technology to plan their daily activities and their free time. However, many planning tools provide general suggestions that may not be suitable for individual preferences or specific situational requirements such as time, location, or interest activities. The growing demand for personalized planning in mobile applications gives an opportunity to create a smart solution that personalizes recommendations to each user's specific preferences. Users today expect convenience, accuracy, and relevance, especially when it comes to planning their day or deciding where to go for activities. Personalized services are no longer an option but rather a requirement for improving user satisfaction and engagement in an increasingly digital era.

1.2 Objectivs

1. To provide the plan which is suitable to the user's preference.
2. To provide up-to-date information on opening hours and availability.

1.3 Scope

This project aims to develop a mobile application that suggests one-day trip plans based on user interests, such as cafes, restaurants, shopping, parks, and more. Users can also provide input, such as starting location, date and time, and the number of places to visit, to refine the suggestions and include more relevant locations in their plans. Users of the PlanADay application

- **can create** personalized one-day trip itineraries based on their interests and preferences.
- **can discover** nearby attractions that align with their preferences and current location.
- **can adjust** their generated plans by rearranging locations, deleting stops, or regenerating.
- **can view** detailed information for each suggested location.
- **can share** their trip plans with others on the platform.
- **can bookmark** plans created by other users for easy access and reference in the future.
- **can explore** their plan history to revisit past itineraries or gain inspiration for new trips.

1.4 Expected Benefits

1. **Seamless Trip Planning:** The main features of the application will help users to create their personal preferences, ensuring a tailored and enjoyable experience.

2. **User Satisfaction and Flexibility:** Users will appreciate both the thoughtfully generated plans and the ease of customizing them to suit their needs, ensuring a more satisfying experience.

3. **Business Partnerships:** By featuring related points of interest, such as shops and cafes, PlanADay opens opportunities for businesses to partner and connect with a highly engaged audience.

Chapter 2

Feasibility

2.1 Introduction

PlanADay Application: Suggest One Day Trip is a mobile application that helps people who want to go on a one-day trip without wasting time to create the plan that has many steps to do such as searching for places, available hours on each place, finding the route path and so on. The application will help reduce the time spent on organizing a one-day trip by suggesting the place due to the user's preferences and providing the place detail

2.2 Problem Statement

Planning a one-day trip can be challenging and time-consuming. It involves multiple steps, including searching for places of interest, checking operating hours, determining suitable routes, and aligning schedules. Many people lack the time or abilities to plan an efficient trip, which often results in frustration, inefficiency, or missed opportunities to explore exciting places. Current solutions may offer individual recommendations or maps, but they do not take a comprehensive approach to creating a personalized, structured plan. This gap highlights the need for an application like PlanADay, which streamlines the process by providing personalized recommendations to help users maximize their travel experience with minimal effort.

2.3 Related research and projects

Location based services: ongoing evolution and research agenda

Explores the evolution and research trends of Location-Based Services (LBS) in the context of the mobile information era, highlighting their significance in delivering location-dependent information to users. It identifies key research challenges, including advancements in positioning, modeling, communication, and the evaluation of LBS-generated data. Additionally, the article addresses social, ethical, and behavioral considerations associated with LBS integration into daily life, offering a comprehensive research agenda to shape the future of LBS in a positive and impactful way.

Personalized Day-Trip Planning: A TSP-TW-Based Multimodal Multicriteria Optimisation Approach

Innovative approach to creating personalized one-day travel itineraries by enhancing the Traveling Salesman Problem with Time Windows (TSP-TW). It incorporates multi-criteria optimization, flexible activities, park-and-ride options, and multiple transport modes to address diverse mobility preferences. Using choice experiments, the authors derive utility functions to optimize itineraries, achieving a 16.19solely on travel time. The method was validated through simulations in a medium-sized German city, showcasing its effectiveness in real-world scenarios.

2.4 Requirement specifications for the new system

2.4.1 Function requirements

1. Preference setting: Users should be able to select their own preference of categories for the application to provide the suggestion for the user.
2. User Input: The application should allow users to input trip details with plan name, categories of interested place, location area, start time, start date, and the number of places to visit.
3. Location suggestion: The application should suggest places based on user preferences and user input to find the appropriate place.
4. Itinerary Generation: The system must provide a sequential trip plan with details like order of visit, travel time between locations, and routing path to each place.
5. Plan Customization: Users should be able to customize the plan including edit plan name. Delete or add more places, reorder the place, and regenerate the plan.
6. Attraction Details: The application should display detailed information for each attraction, including photos, descriptions, rating, and opening hours.

2.4.2 Data requirements

1. Location data from Google Map API

2.4.3 System requirements

1. Smartphone with Android OS or iOS
2. Has geo-positioning sensor (GPS)

2.5 Implementation techniques

2.5.1 Frontend

- Language
  - Dart
- Framework
  - Flutter

2.5.2 Backend

- Language
  - Typescript
- Framework
  - Bun
  - Elysia
- External Service
  - GoogleMap API
- Testing
  - Jest
  - Postman

2.5.3 Infrastructure

- Operating System
  - Debian Linux
- Hosting
  - SSD Node (Virtual Private Server)
- Container Management
  - Docker
- Image Storage
  - Docker Registry

- External Service
  - Google Cloud Platform
- Web-server
  - Nginx
- Database
  - mongoDB
  - Redis
  - PostgreSQL

2.5.4 Other

- Editor Development: Visual Studio Code
- DevOps: Github Action
- Project Management: Trello
- Online meetings: Microsoft Teams and Discord
- Design: Figma and Canva
- Clipart and Icon: Freepik, Scale by Flexiple, Flaticon and Feather
- Diagram Design: Lucid Chart

2.6 Implementation Plan

No.	Task Name	Duration	Start Date	End Date
Phase 1: Project Initiation				
1.	Define topic	1 week	25 January 2024	1 Febuary 2024
2.	Define scope	1 week	25 January 2024	1 Febuary 2024
3.	Conduct feasibility study	1 week	25 January 2024	1 Febuary 2024
4.	Develop business model	1 week	25 January 2024	1 Febuary 2024
Phase 2: Project Acquisition				
1.	Project Profile	2 weeks	11 March 2024	25 March 2024
2.	Project proposal preparation	2 weeks	11 March 2024	25 March 2024
3.	Submit project approval	1 day	25 March 2024	25 March 2024
Phase 3: Project Planning				
1.	Manage scope and requirements changes	2 weeks	4 April 2024	19 April 2024
2.	Develop project management plan	3 weeks	10 August 2024	3 Setember 2024
Phase 4: Project Execution				
1.	Presentation Project Exam 1	1 day	2 April 2024	2 April 2024
2.	User Interface Design	1 month	24 April 2024	6 June 2024
3.	Presentation Project Exam 2	1 day	4 June 2024	4 June 2024
4.	Proof of Concept (POC)	3 weeks	23 July 2024	15 August 2024
5.	Database design	2 weeks	30 July 2024	16 August 2024
6.	Development	3 months	10 August 2024	9 November 2024
	- Frontend	3 months	10 August 2024	9 November 2024
	- Backend	3 months	10 August 2024	9 November 2024
7.	Alpha Exam: Midterm Examination	1 day	11 October 2024	11 October 2024
8.	Testing	20 days	20 October 2024	9 November 2024
9.	Deployment	3 days	6 November 2024	9 November 2024
10.	Beta Exam: SIT's Undergraduate Student Demonstration Project Day	1 day	11 November 2024	11 November 2024
Phase 5: Project Closeout				
11.	Planning to do Project Report	1 day	13 November 2024	13 November 2024
12.	Project Report preparation	1 week	18 November 2024	25 November 2024

Table 2.1: Table of the project task

Chapter 3

Analysis and Design

3.1 Introduction

This chapter provides an explanation of our project design in the following sections: analysis of the existing system, user requirement analysis, and system design. First is analysis of the existing system, this section evaluates the current systems and applications available in the market, identifying their features, limitations, and areas for improvement along with the PlanA-Day application. Second,the user requirement analysis section that explains the features of the project. Third, the system design user diagrams consist of a context diagram, data flow diagram, activity diagram, use case diagram, system sequence diagram, flow chart, and key-valued database diagram.

3.2 Analysis of the existing system

Currently, several applications are available in the market that assist users in planning trips, such as Google Maps, Strippl, Gethergo, and ChatGPT. These platforms offer various features, including route navigation, attraction suggestions, and itinerary planning. However, they differ in their ability to cater to specific user needs.

For instance, most existing applications, like Google Maps and Strippl, are well-suited for users who already know where they want to go, as they provide efficient routing paths and location-based directions. Similarly, Gethergo offers recommendations based on user-defined interests but requires users to input detailed preferences or destinations. On the other hand, PlanADay offers catering to users who may not have a clear idea of where they want to go. The application generates personalized, one-day trip plans based on broad user interests, preferences, and current location, offering a complete itinerary without requiring extensive input or prior knowledge of specific destinations. The comparison table below summarizes the key features of each application.

Feature	Strippl	gethergo	Google Map	ChatGPT	PlanADay
Require less input	✓	-	✓	-	✓
Suggest plan base on user interests and preferences	-	✓	-	✓	✓
Suggest route based on location	✓	✓	✓	-	✓
Real-time place Details	-	✓	✓	-	✓
Provide routing path	✓	✓	✓	-	✓
Generate/ Regenerate plan	-	-	-	✓	✓
Save related plan from others	✓	✓	✓		✓
Overview Detail	-	✓	✓	-	✓

Table 3.1: Table of the feature comparisons

The PlanADay application aims to address these limitations by offering a more personalized and flexible experience. It not only generates customized one-day trip plans based on user interests but also allows users to adjust itineraries, share plans with others in the community, and bookmark suggestions for future use. By bridging these gaps, PlanADay seeks to provide a more user-focused and collaborative solution compared to existing systems.

3.3 User requirement analysis

The user requirement analysis aims to identify the expectations, preferences, and functional needs of users who seek efficient and personalized one-day trip planning. Below is a breakdown of the key user requirements that they might found:

1. Users do not get the satisfied plan from the user's input.
    - Customization plan: Users can edit the plan which is delete the place, add more places, and reorder the place on their own again to make the most appropriate plan for the user.
2. Users want to change their preferences.
    - Preference setting: Users can reselect their preferences all the time in application.
3. Users do not have any specific input.
    - Suggestion plan: The application provides the plan that are generated from other users, so the user can use other's plan to travel.

3.4 System Design

3.4.1 Use case diagram

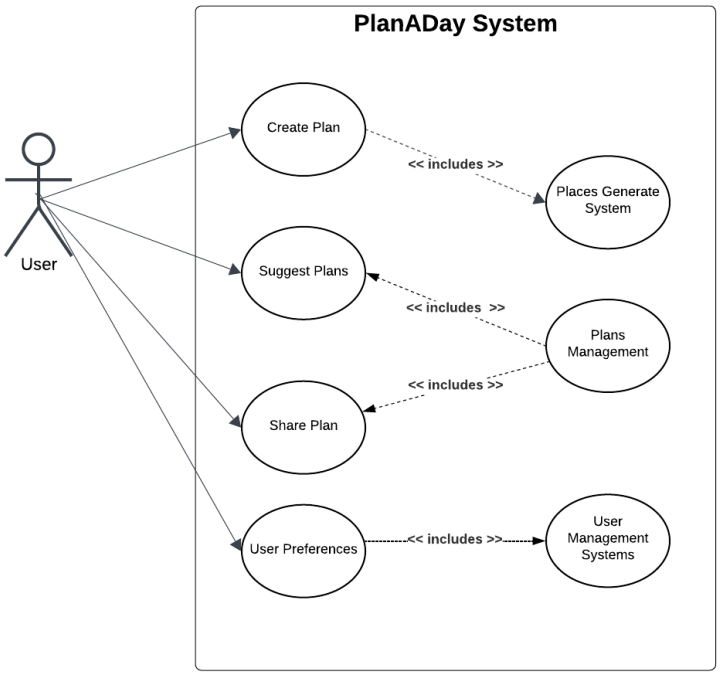

Figure 3.1: Use case diagram

The Actors who are in the PlanAday System. The user must create the account to authenticate in the system. After the user input all field forms to create a plan, including the plan name, place type, location, date and time and amount of places, The system will generate the plan and display it to the user. if the user is not satisfied they can customize the plan by rearrange, regenerate. The suggestion plans will show the public plans that mean each plan is shared by other users. and filter the plan by user preferences.

3.4.2 Context diagram

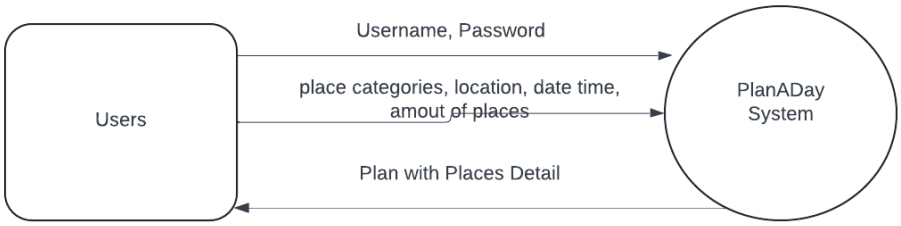

Figure 3.2: Context diagram

The context diagram shows the overall of PlanADay System that requires the username, password that need to be used in authentication service, place categories, location, date and time, amount of places that need to be used in generating the plan. Then the system will provide a plan that matches with input that includes details.

3.4.3 Activity Diagram

Authentication process

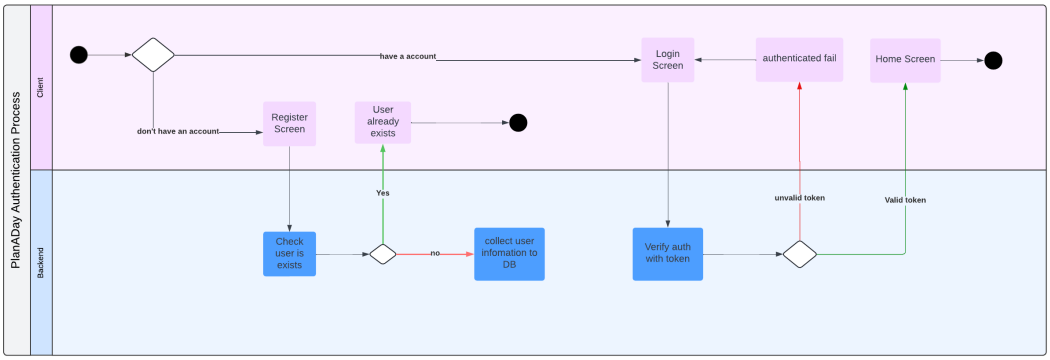

Figure 3.3: Activity Diagram of Authentication Process

The PlanADay Authentication Process illustrates the workflow for user registration and login, divided into client-side and backend interactions. The process begins with a decision: if the user does not have an account, they are directed to the registration screen. On registering, the backend checks if the user already exists. If the user exists, they are informed and redirected to the login screen; otherwise, their information is collected and stored in the database. For users with an account, the login screen allows them to authenticate. The backend verifies the authentication using a token. If the token is valid, the user is granted access to the home screen. In the event of an invalid token, authentication fails. This flow ensures a secure and user-friendly mechanism for handling account creation and login.

Generate Plan Process

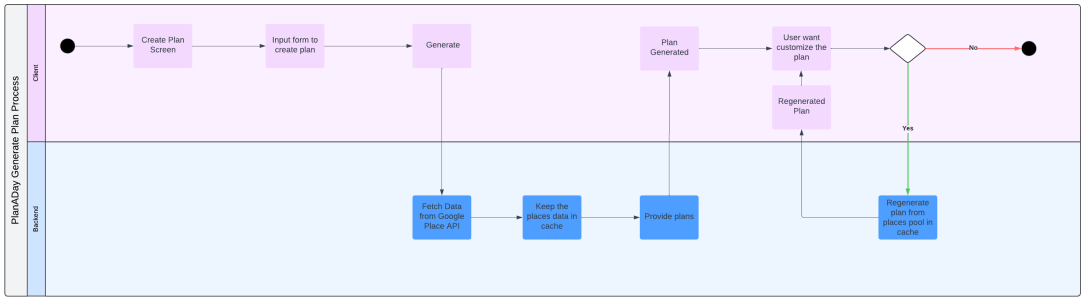

Figure 3.4: Activity Diagram of Generate Plan Process

The PlanADay Generate Plan Process outlines the workflow for creating and customizing plans. The process begins on the client side, where the user accesses the ”Create Plan” screen, fills out an input form, and initiates the plan generation. The backend fetches relevant data from the Google Places API, caches the data for future use, and provides the generated plan to the client. After the plan is displayed, the user has the option to customize it. If they choose to do so, the backend regenerates the plan using the cached data without making additional API calls. If no customization is needed, the process ends. This workflow ensures efficiency through caching and provides a seamless user experience for generating personalized plans.

3.4.4 User Interface Design

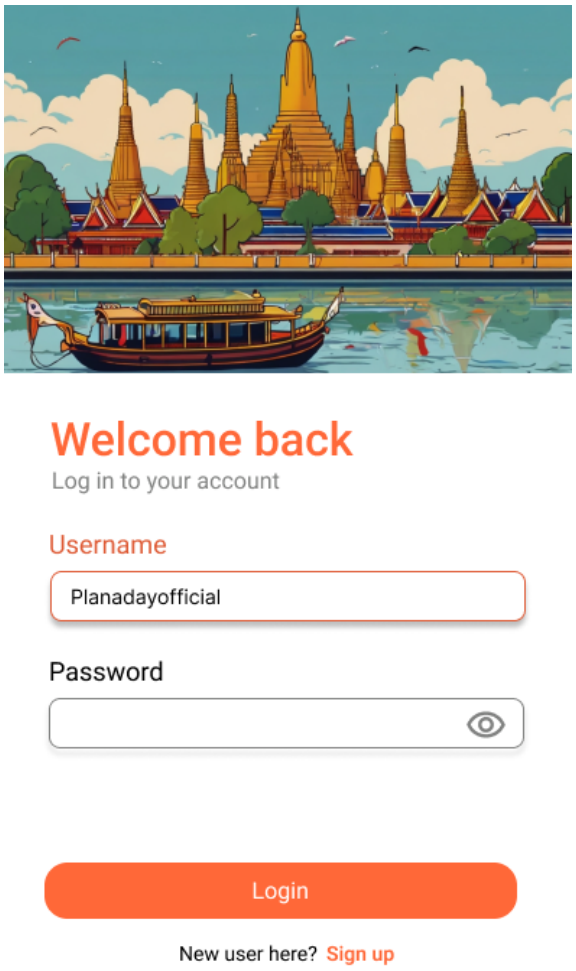

Figure 3.5: Login Page

Figure 3-5. display the Login page of PlanADay application. The system requires the user to login using username and password for authentication.

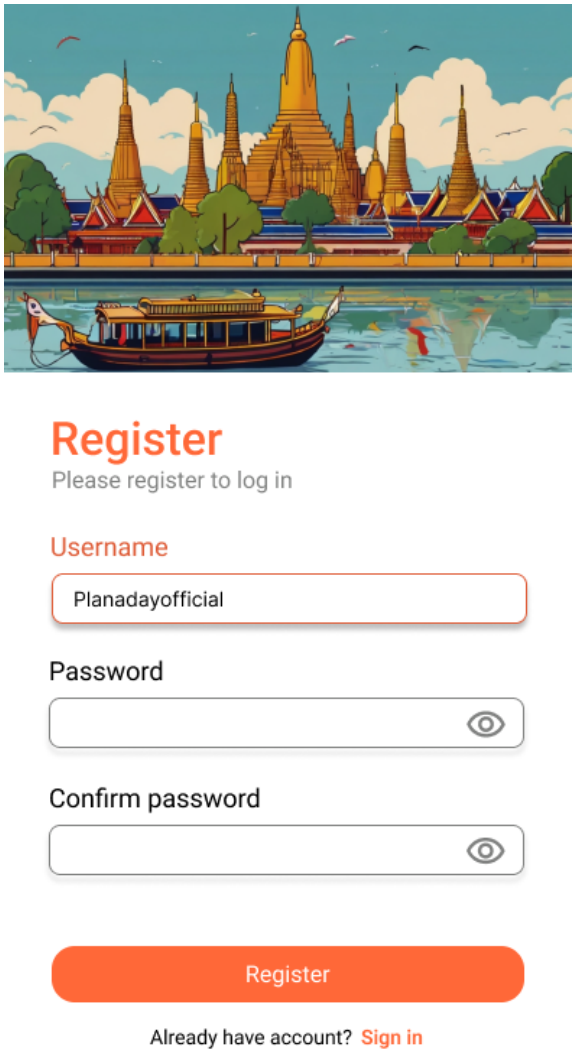

Figure 3.6: Register Page

Figure 3-6. display the Register page of PlanADay application. When users get in the application for the first time, they are required to create the account to use in the system.

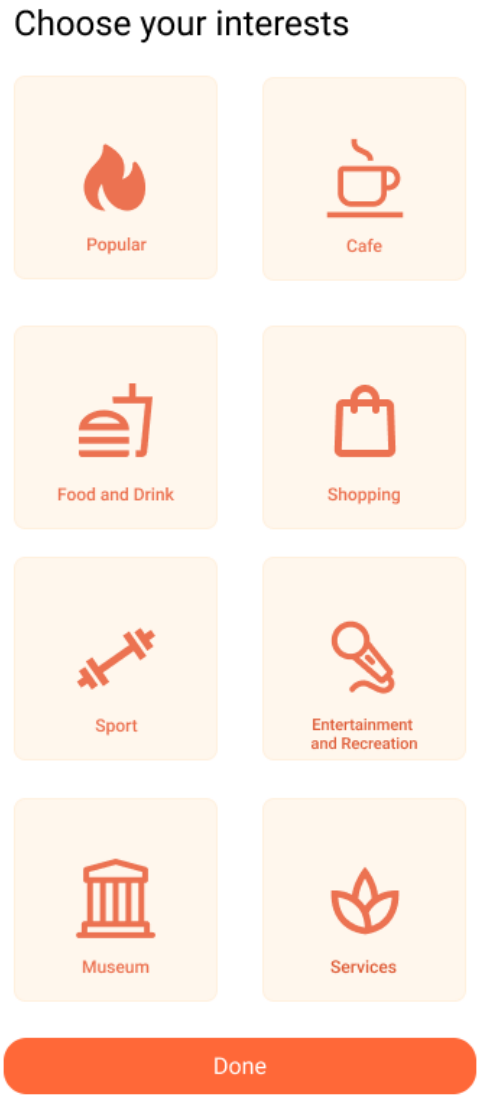

Figure 3.7: Persona Page

Figure 3-7. display the Persona page of PlanADay application. Users can set their preferences in this page and it will be used to suggest the plan in the system.

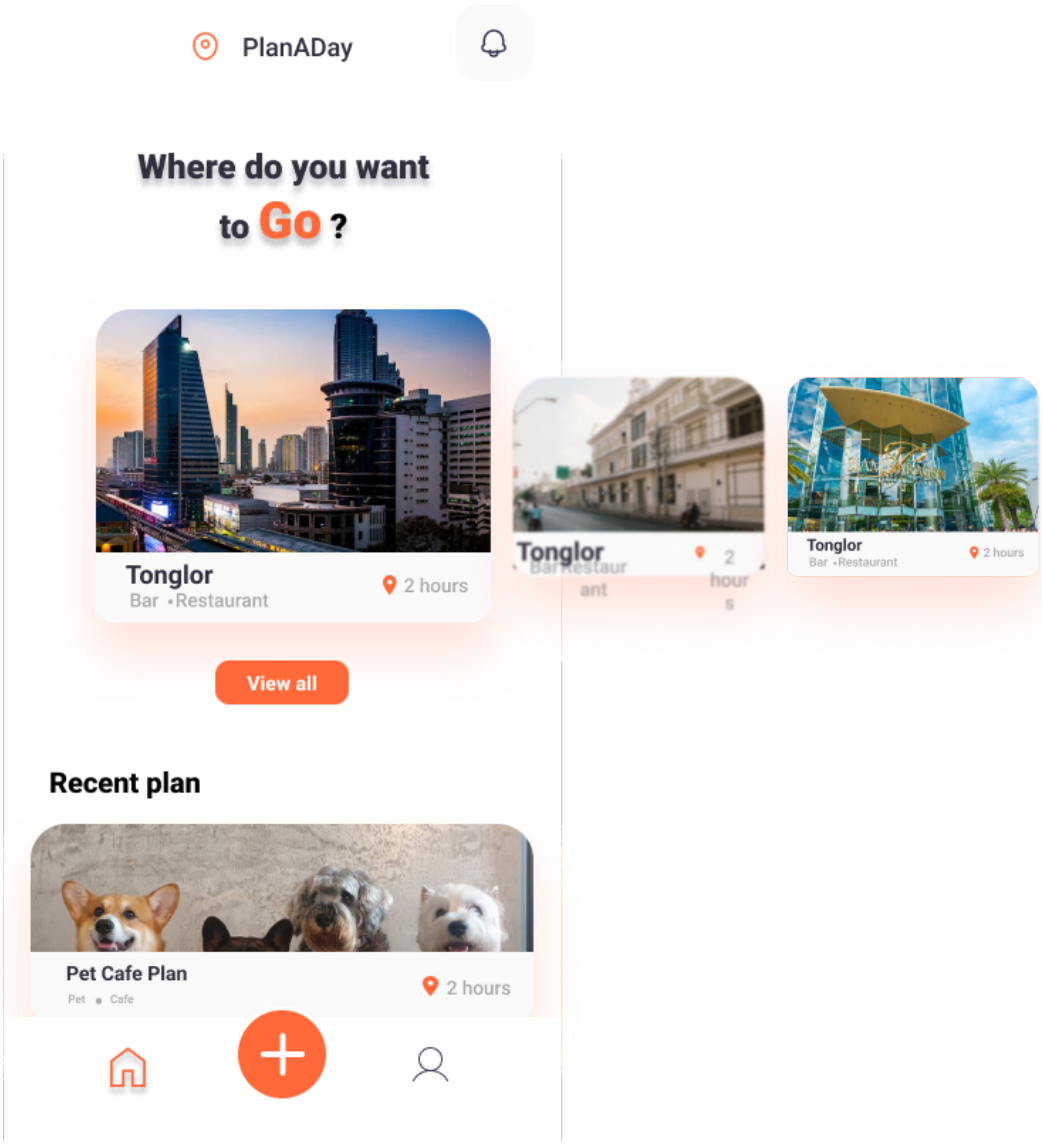

Figure 3.8: Home Page

Figure 3-8. display the Homepage of PlanADay application. The application will show the plan suggestion and the plan that the user created in this page.

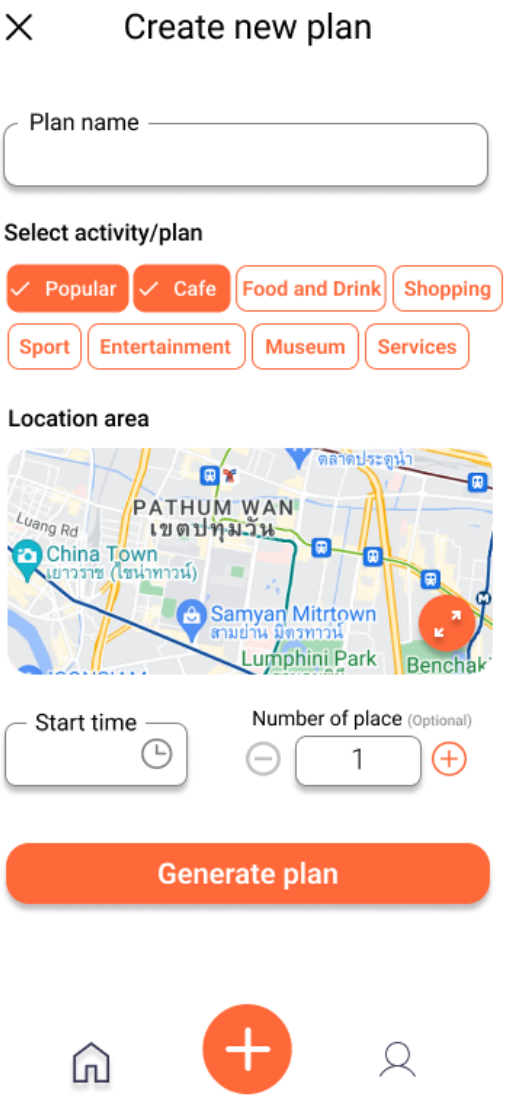

Figure 3.9: Home Page

Figure 3-9. display the Create Plan page of PlanADay application. Users can create their own plan on this page. The input are plan name, categories, location area, start date, start time, and number of places.

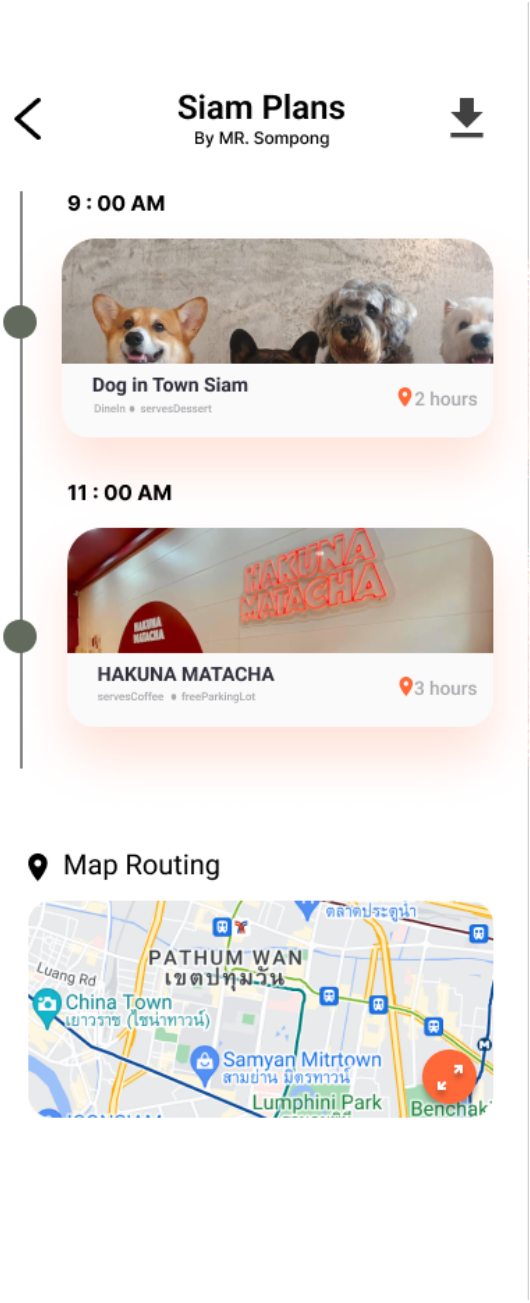

Figure 3.10: Generated Page

Figure 3-10. display the Generated Plan page of PlanADay application. After the user selects their input and clicks on the generate plan button, the system will suggest the plan according to the input, and all the plan details will show in this page.

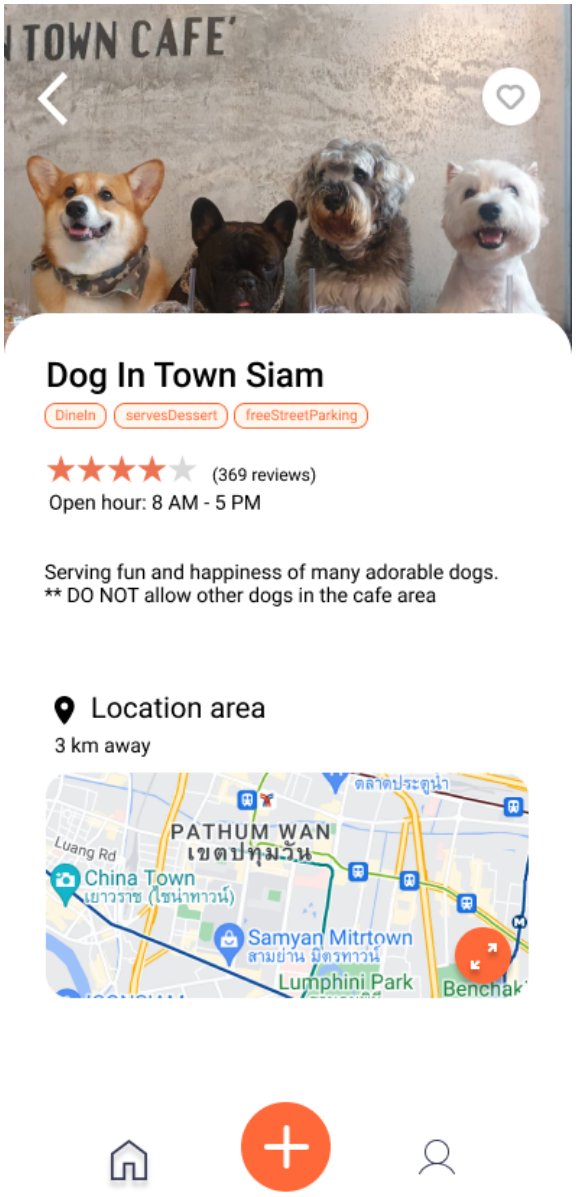

Figure 3.11: Place Detail Page

Figure 3-11. display the Place Detail page of PlanADay application. This page will show after the user clicks the place card in the Generated Plan page. The details that this page shows are place name, service types, rating, available open hours, and location on google map.

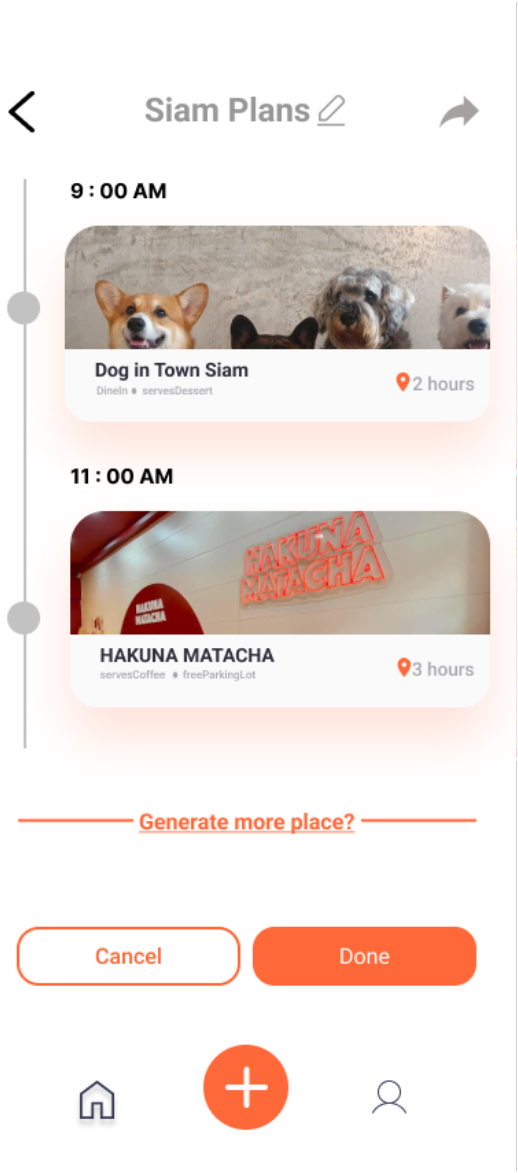

Figure 3.12: Edit Plan Page

Figure 3-12. display the Edit plan page of PlanADay application. If the user is not satisfied with the generated plan, they can click on the edit plan button and it will navigate to this page. Users can edit the plan name, delete each place, regenerate the place, and add more places.

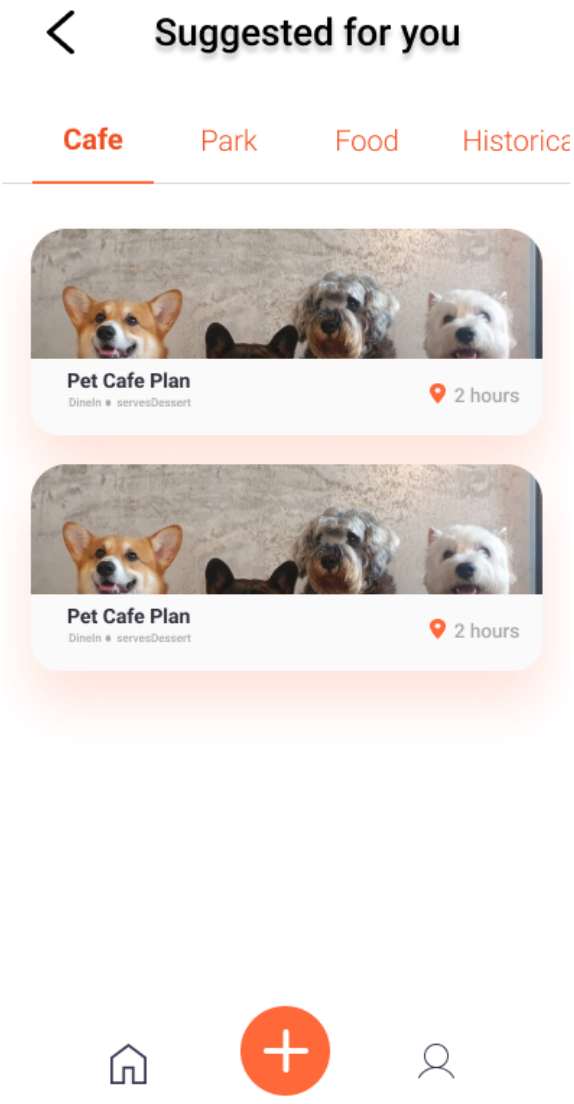

Figure 3.13: Suggest Plan Page

Figure 3-13. display the Suggest Plan page of PlanADay application. This page can get on from the Homepage of the application. This page will show the plan that was generated from others user in the application and the user can click on each plan to see the details.

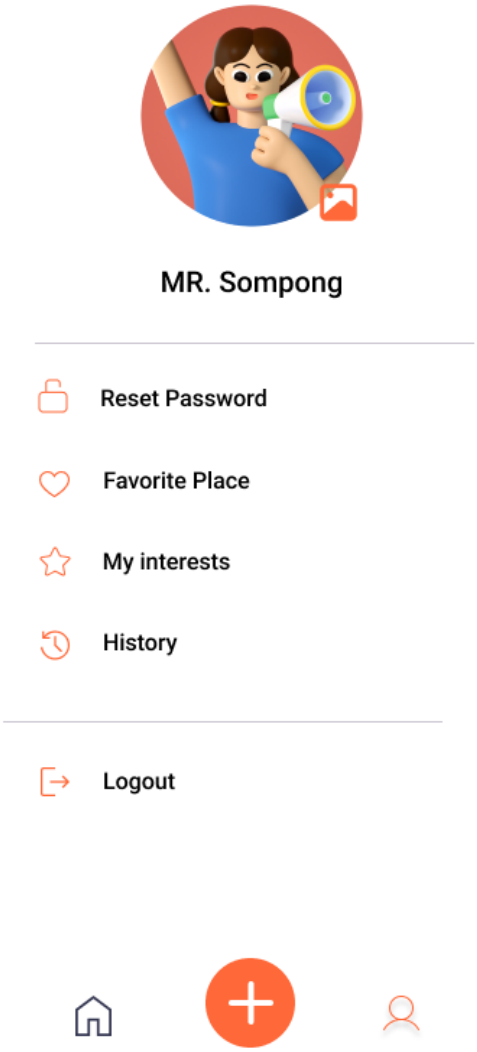

Figure 3.14: Profile Plan Page

Figure 3-14. display the Profile page of PlanADay application. Users can access the history page, persona page, and logout from their account on this profile page.

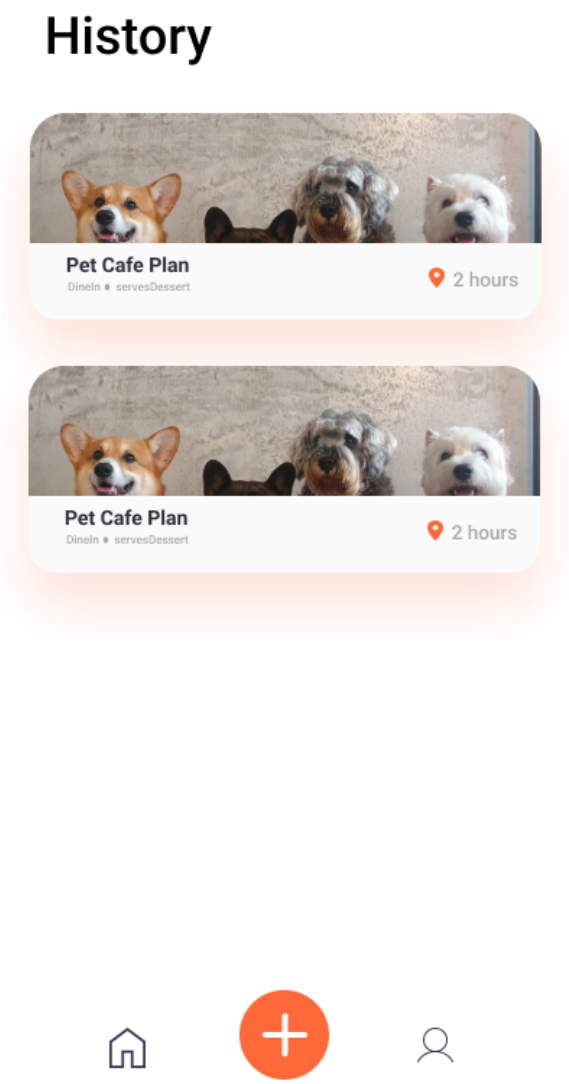

Figure 3.15: History Page

Figure 3-15. display the History page of PlanADay application. This page shows all the plans that the user generated.

3.5 Database design

3.5.1 Relational Database

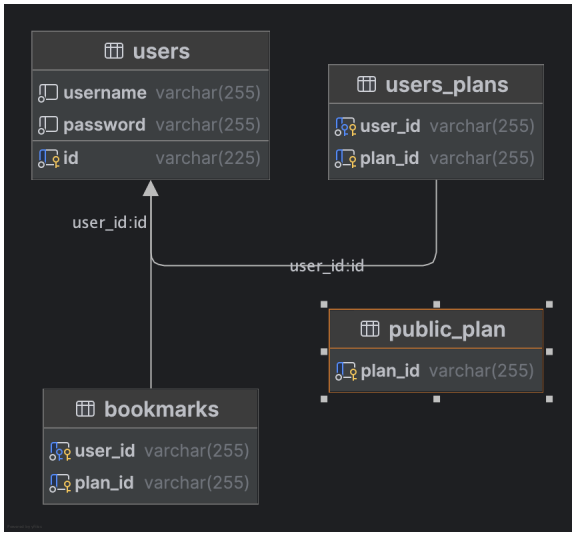

Figure 3.16: Relational Database Schema

Our database relies on PostgreSQL. The ER diagram illustrates a system for managing public plans and user interactions with them. Public plans are stored in the public-plan entity, identified by their unique plan-id. Users are represented in the users entity with their username, password, and id. The users-plans entity establishes a many-to-many relationship between users and public plans, allowing users to be associated with multiple plans and vice versa. The bookmarks entity also represents a many-to-many relationship, enabling users to bookmark multiple public plans and plans to be bookmarked by multiple users.

3.5.2 non-relational database

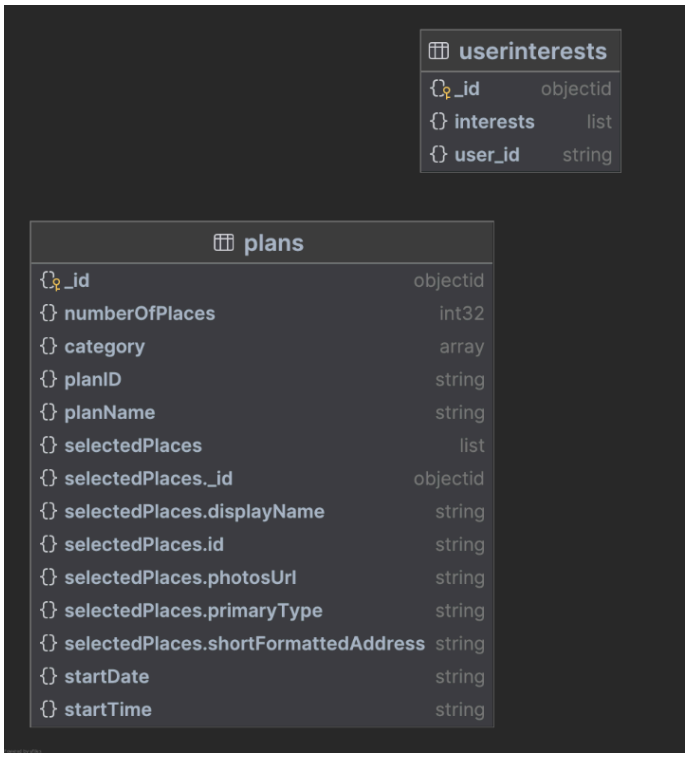

Figure 3.17: Non-Relational Database Schema

The ER diagram depicts a database schema for managing travel plans and user interests. The user interests table stores user IDs and their associated interests. The plans table contains detailed information about travel plans, including the number of places, category, plan ID, name, selected places with their attributes, and start date/time.

Chapter 4

System functionality

4.1 Introduction

This chapter describes the system architecture, the plan, and the test result. First, the chapter begins with the system architecture which consists of the system functionality and main functions. The system functionality describes the components of the system and the overall structure of the system. The main function section describes all the functions in the system. Lastly, the test plan and the results explain the processes and procedures used for testing the system.

4.2 4.2 System architecture

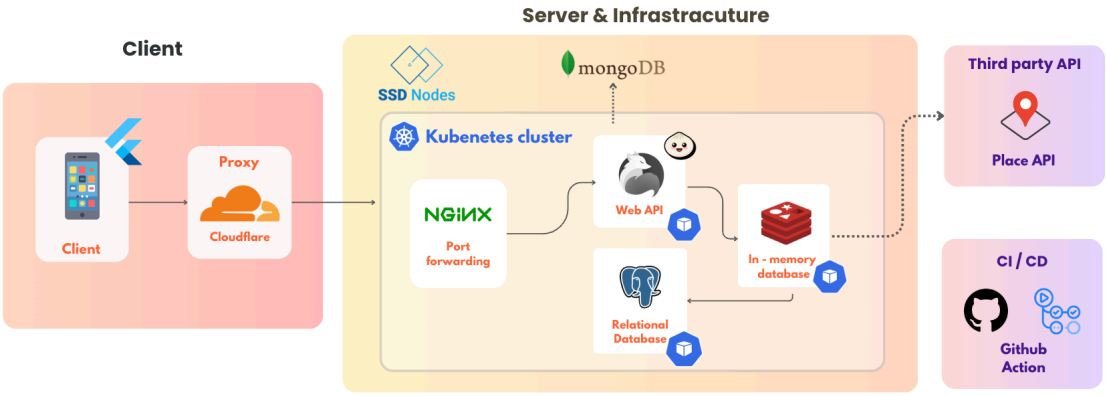

Figure 4.1: System architecture

The PlanADay software architecture employs a client-server model, with the frontend handling user interaction through screens and forms, and the backend managing core logic, API integrations, and database operations. The frontend allows users to create plans, customize them, or authenticate via registration and login, sending input to the backend for processing. The backend retrieves data from external APIs like Google Places, caches it for efficient reuse, verifies user credentials using token-based authentication, and stores user data securely. This architecture prioritizes separation of concerns, efficiency through caching, and security, ensuring a scalable and user-friendly system for generating and customizing plans.

4.3 4.3 Test Plan

This figure shows the welcome message and login button, Users can log in by Google to register and log into our application.

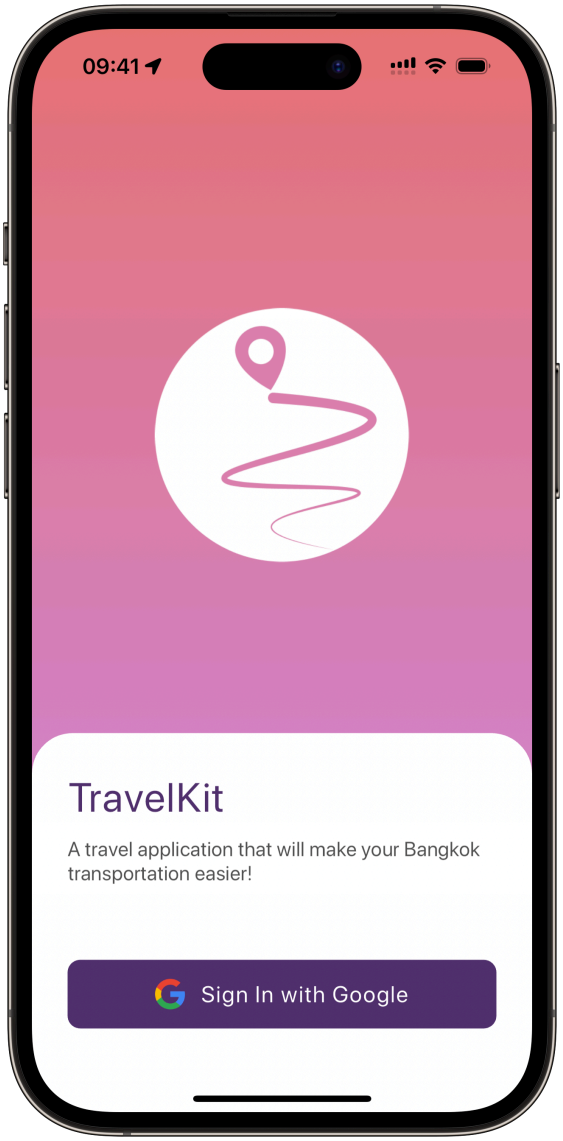

Figure 4.2: Welcome screen

This figure shows the map, current location, the name the current location after logging in and the user can tap the location icon to update the map screen to show their current location. The user can switch between the home screen and the community screen using the bottom navigation bar below.

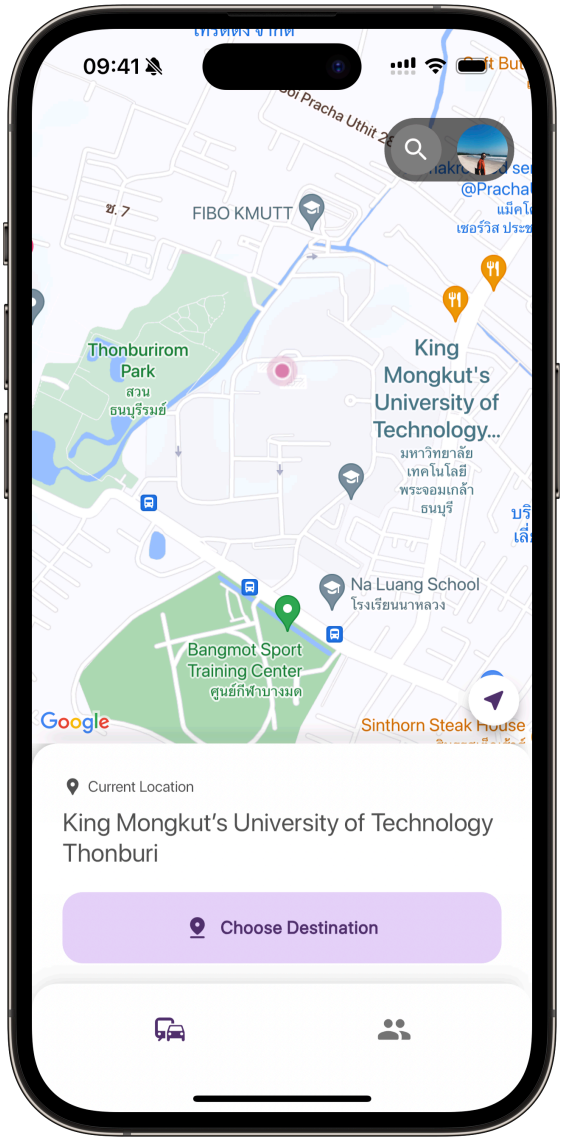

Figure 4.3: Home screen

Users can search routes by specifying both the starting location and the destination to find routes for users.

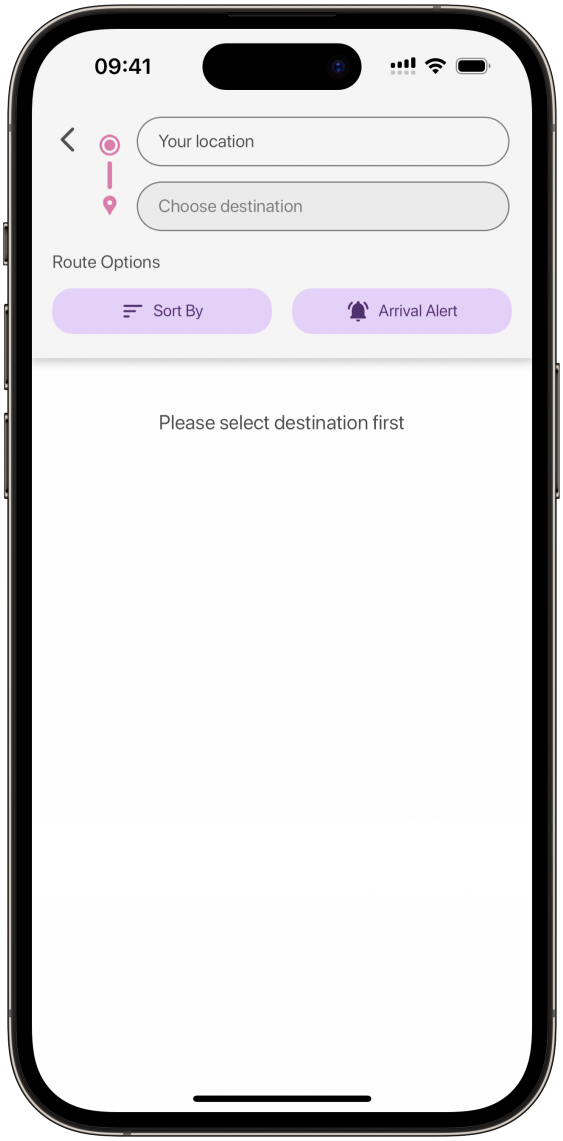

Figure 4.4: Choosing destination screen

This figure shows the search route results from the starting location to the destination. All routes are sorted by estimated time of arrival by default and users can see multiple routes to go to the destination.

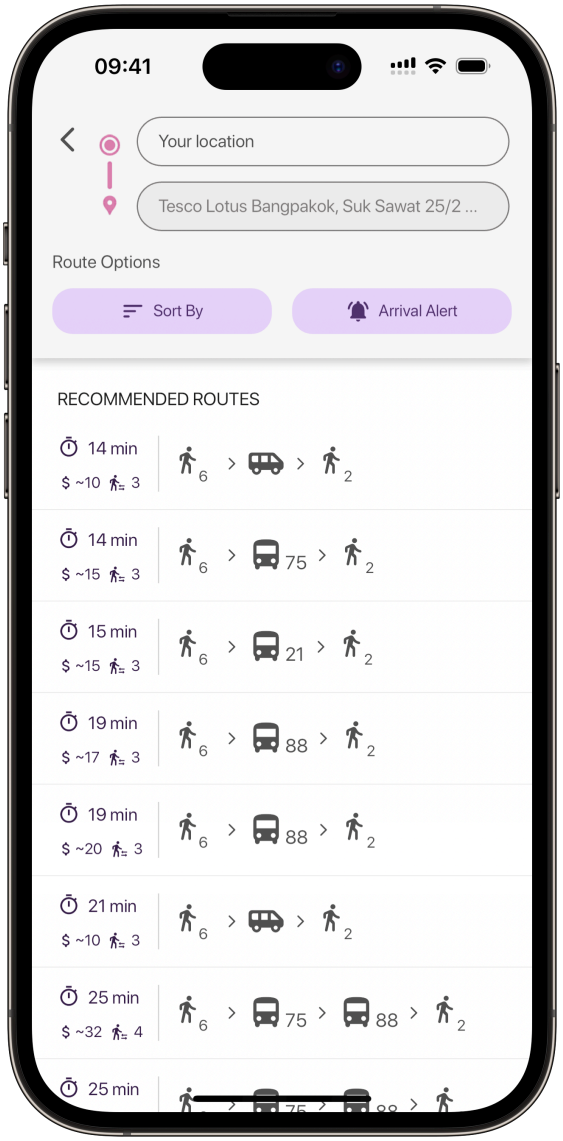

Figure 4.5: Choosing route screen

The recommended routes can be sorted according to the user's preferences, such as the fastest arrival time, the cheapest price, the shortest transfer, and the avoid ferry option.

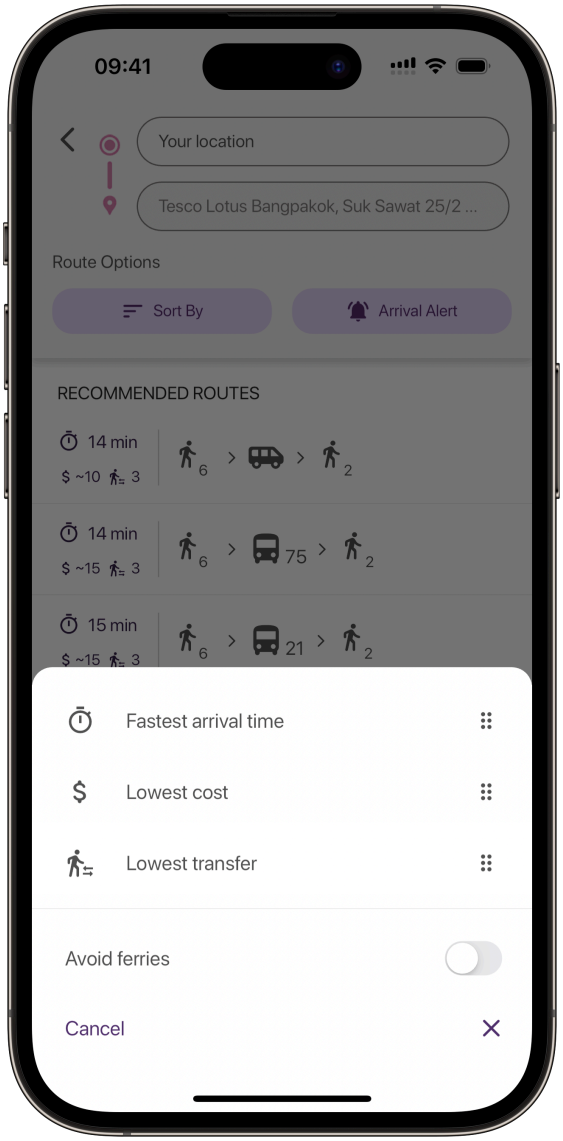

Figure 4.6: Route options

After swiping up the bottom sheet, users will see the details of the selected route including cost, estimated time of arrival, and passed stop of each transport type. Users can tap the Start button for starting to real-time navigation or the Share icon to share the selected route with others.

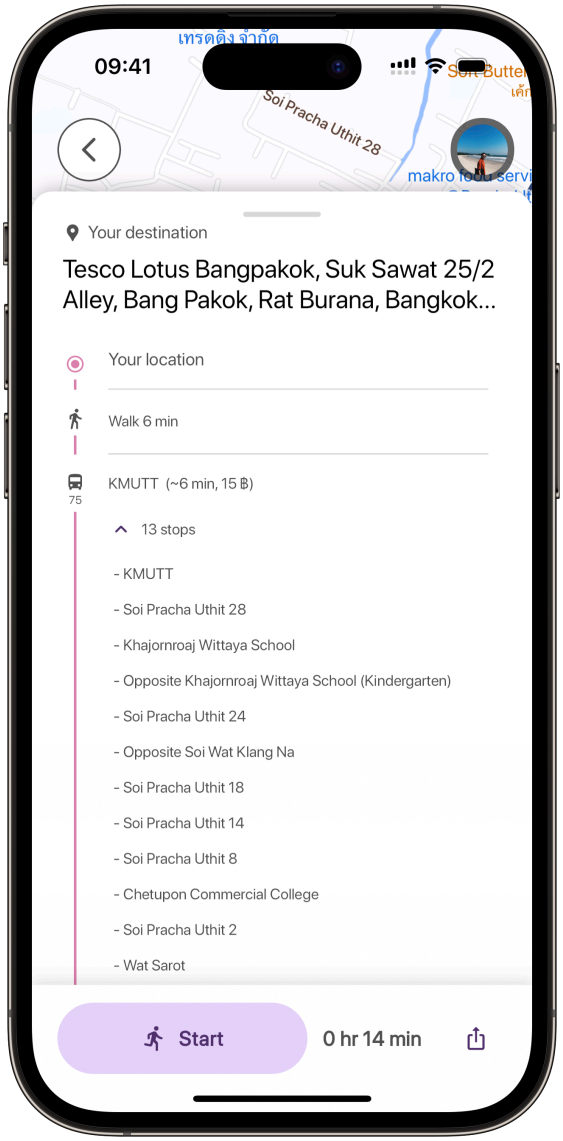

Figure 4.7: Selected route detail screen

This screen shows the details of the route in real-time including the current location, estimated arrival time, and the navigation overlay showing the current step with time and the next step at the top of the screen.

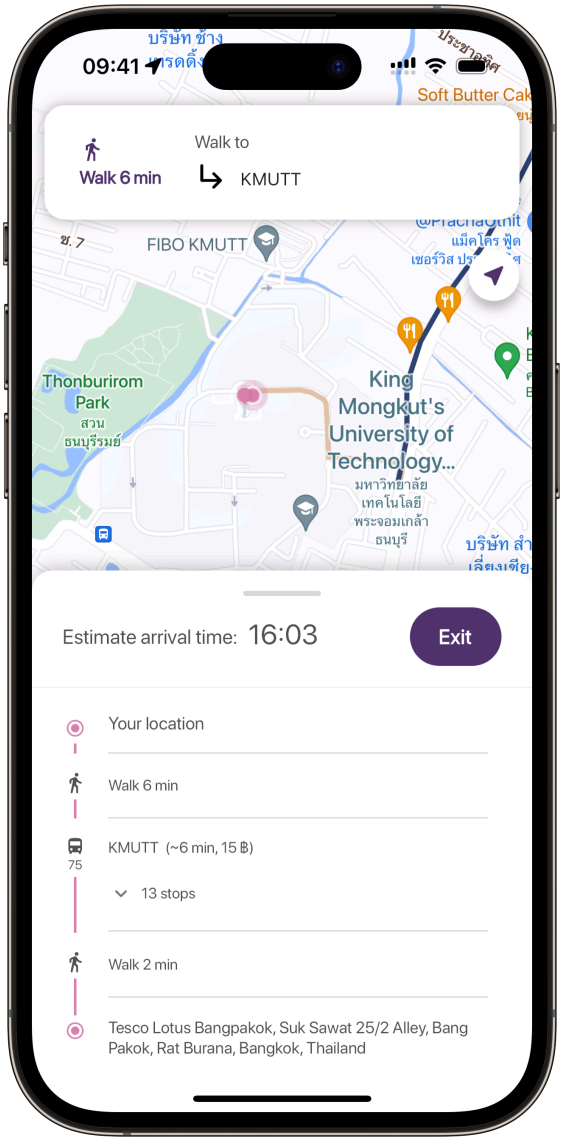

Figure 4.8: Real-time navigation screen

4.4 Test plan and test result

Module	Test expectation	Expected result	Result
Authentication	Login with Google	Successful login to the system with user information provided	Success
Home page	Able to navigate between home page and community page	Can change tab between two page	Success
	Able to see the map with the current user location	See the map and correct current location	Success
	Able to logout from profile	User is logged out	Success
	Able to navigate to search page/ choose destination	Navigate to search page correctly	Success
Routing	Able to search for places	User sees the list of places from Google API	Success
	Able to sort the route options	User can change the sort by choices to change the travel preferences based on time, cost, and number of transfer	Success
	Able to see the suggested routes when entering start and end location	User sees the list of suggested routes correctly with the details	Success
	Able to navigate to see each route detail	User sees the route detail on the map and modal with travel time and cost	Success
	Able to share the route	User gets a 6-digit code for sharing the route with other users	Success
	Able to start the trip	User navigates to the routing screen with a map and directions provided	Success
	Able to receive notification when an event happens	The system alerts the correct notification	Success
Community	Able to get the exact route when searching by code	User navigates to the route detail page correctly	Success
	Able to see the suggested places	List of suggested places displays correctly when entering the community page	Success
	Able to navigate to suggested routes page from suggested places	User navigates to the routes selection page correctly	Success

Table 4.1: Table 4-1. Test Plan

Chapter 5

Summary and suggestions

5.1 Introduction

This chapter provides a summary of the project, which comprises three parts: project summary, problems encountered and solutions, and suggestions for further development. The first section summarizes and provides the overall results of the project. The second section outlines the problems encountered and proposes solutions to the limitations of the project. Finally, suggestions are provided for the future improvement of the project.

5.2 Project summary

TravelKit is a mobile application that assists individuals in navigating through public transportation options like buses, sky trains, subways, boats, and mini trucks to reach their desired destinations. Users can access information such as pricing, travel duration, and the number of transfers involved in their chosen route. Additionally, the application includes a feature enabling users to share their routes with one another. From the project's objective and scope, we can create a software that can calculate travel routes, distances, and estimate travel costs with optimizing the suggested routes based on user personal choices.

5.3 Problems encountered and solutions

There are two main problems that we encountered during the project. First problem that we find difficult to cope with is the process of developing mobile application that uses Dart with Flutter as a programming language and framework that we are not quite familiar with. So, we need to learn some new programming styles for the development. The second challenge involves incorporating Neo4j as a graph database, a tool with which we lack prior implementation experience. Consequently, we must familiarize ourselves with the database's query language in order to effectively build our system.

5.4 Suggestions for further development

5.4.1 Integration of AI in our system

Currently, our application has a feature that suggest routes based on the users preferences but we can use artificial intelligence to enhance this feature by collecting data from user. We can use that data to predict what sorting preferences user is likely to choose. Furthermore, we can suggest more accurate recommended places to user if we have a proper ai model.

5.4.2 Responsive design and implementation

Currently, our application is working correctly with only certain devices. To support wide variety of mobile and desktop screens, it need revisions in design and implementation to support the responsive design.

5.4.3 High coupling of the data

Currently, our service has accurate data that we can return route data correctly, but to modify route data, we need to update both in the MongoDB and Neo4j separately which is consequence of manually inserting data to all databases and all relation in Neo4j needs to changed for the updated route.