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)

Copyright reserved

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

Contents(CONT.)

List of Tables

List of Figures

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

Our transportation application aims to enhance user experience by providing seamless navigation from the starting point to the destination. In addition to providing efficient routes, we also offer transparent pricing information, allowing users to select the transportation method that best suits their budget. Our navigator displays the actual cost of each route, enabling users to make informed decisions and pay only for the transportation they use. In addition, we provide the mini social media in the application, users can also share their recent transportation methods to the mini community so that other users can view them.

2.2 Problem Statement

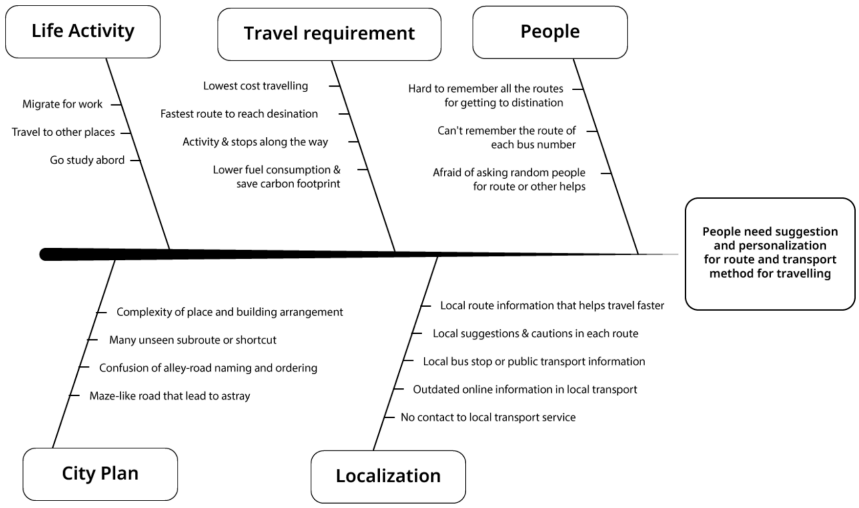

Figure 2.1: Problem diagram

In people's everyday life, they need to transport to places for doing things, from daily commute to work, go on a trip or even going abroad. However, there's many concerning and obstacle for transportation. People themselves not able to remember all the routes (i.e., transportation line) that bring him to destination or sometimes there might be more better (in terms of cost, duration, or transfer) transportation route for them that they don't know. These problem may influenced by city plan that have complex arrangements or some localization that might leads to misinformation. As transportation, especially public transportation has a important impact to people, improving transport information to be more accurate, accessible, and personalized will give an opportunity to people for more ease of life, so all these reasons lead to why we need a better routing application.

2.3 Related Research Projects

2.3.1 Google Maps

This application is designed for regular users who use to find the location of the place, and transportation method between one point to another point, Google Maps allows users to see the detail of the traffic and estimate the time of transporting in each way that user select.

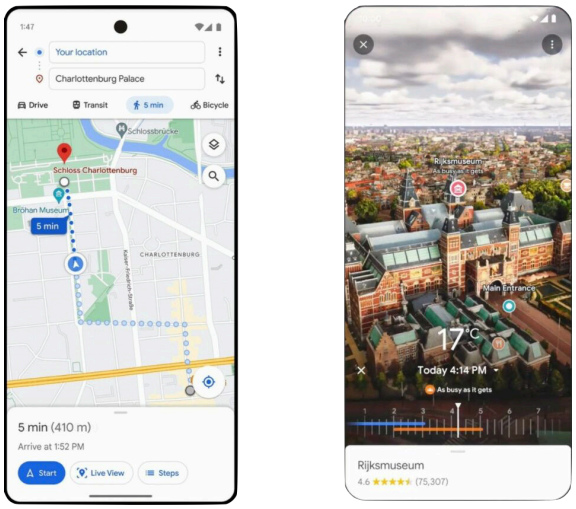

Figure 2.2: The Google Maps application

2.3.2 Apple Maps

This application is for Apple users that allows users to find the location, the way to go to their destination, estimate the time of transportation, see the details of traffic, and Apple Maps also provides the bicycle method, so cyclists can use Apple Maps to find the best way to get to their destination by bicycle.

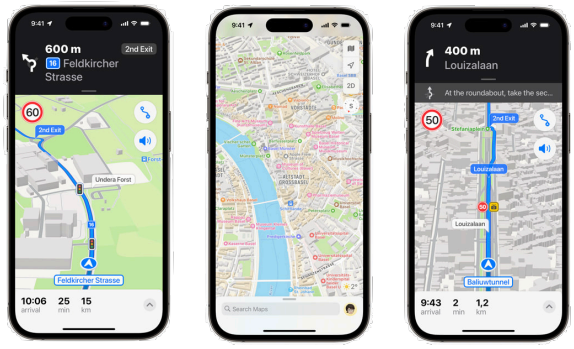

Figure 2.3: The Apple Maps application

2.3.3 ViaBus

ViaBus application is the application that provides user with the bus stops and routes, real-time bus locations approaching your stop, and recommended routes to travel from one place to another place which include buses, trains, and boats.

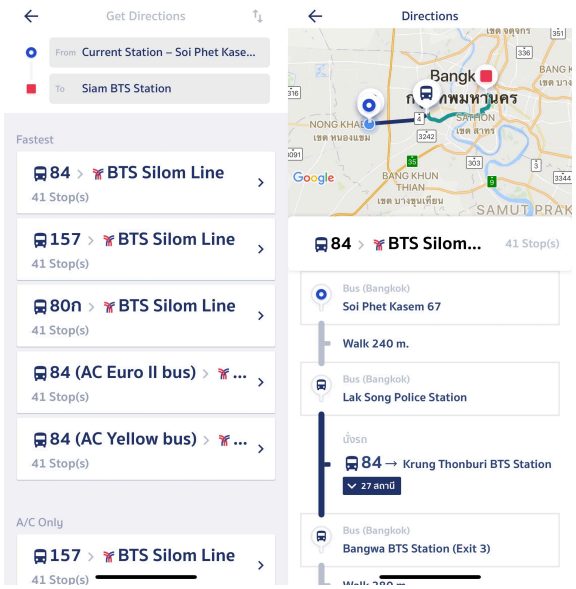

Figure 2.4: The ViaBus application

2.3.4 Existing Functions

Applica-tions	Map route	Search for destination	Routes recom-mendation	Estimate travel time	Estimate travel cost	Record travel info
Google Maps	Yes	Yes	Yes	Yes	No	Yes
Apple Maps	Yes	Yes	Yes	Yes	No	Yes
ViaBus	Yes	Yes	No	No	No	No
TravelKit	Yes	Yes	Yes	Yes	Yes	Yes

Table 2.1: Existing functions of related research application and TravelKit

2.4 Requirement Specifications

2.4.1 System requirements

- Users select the destination after that the map view shows the path to travel to the destination, the estimated cost of traveling, and the estimated time to get there and the user can select the best way for them.
- Record the selected path from the user.
- Create the graph that provides the way to travel to the destination.
- Estimate the time from the one place to another place based on the path that is selected by the user.
- Estimate the cost on the selected path.
- Users can start and stop their trip.
- After a trip has been recorded, the application will record it in the database.

2.4.2 Mobile application requirements

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

2.5 Implementation Technique

- Frontend
  - Programming language: Dart
  - UI SDK: Flutter
  - HTTP Client: Dio
- Backend
  - Go Runtime
  - GoFiber for API server
- Database Server
  - MySQL: Persistent and relational data store
  - MongoDB: Document Database for storing transport data
  - Neo4j: Graph database for routing and path finding

- 3rd party API
  - Firebase Authentication: Sign-in and credential management
  - Google Maps API: Geographic information retrieval
  - OSRM: Routing engine for shortest paths in road networks
- Infrastructure
  - Container management: Docker
  - DNS: Cloudflare
- Development Software
  - Visual Studio Code
  - Goland
  - DataGrip
  - Postman
  - iOS, Android simulator
- Others
  - Version Control: GitLab
  - User interface design: Figma

2.6 Deliverables

2.6.1 Application

- Source code of application
- Mobile application

2.6.2 Documentation

- Project report
- Meeting log
- Poster

2.7 Implementation Plan

Task Number	Tasks	Duration in hours	Jan		February				March				April				May				June				July				August				September				October				November			
			Wk1	Wk2	Wk1	Wk2	Wk3	Wk4	Wk1	Wk2	Wk3	Wk4	Wk1	Wk2	Wk3	Wk4	Wk1	Wk2	Wk3	Wk4	Wk1	Wk2	Wk3	Wk4	Wk1	Wk2	Wk3	Wk4	Wk1	Wk2	Wk3	Wk4	Wk1	Wk2	Wk3	Wk4	Wk1	Wk2	Wk3	Wk4	Wk1	Wk2	Wk3	Wk4
1	**Proposal**																																											
1.1	Team Discussion 1	4																																										
1.2	Summarize the topic	8																																										
1.3	Team Discussion 2	2																																										
1.4	Proposal Sumbmission	6																																										
2	**Document Chapter 1 - 2**																																											
2.1	Draft chapter 1	4																																										
2.2	Team Discussion 2	2																																										
2.3	Draft chapter 2	6																																										
2.4	Team Discussion 3	2																																										
2.5	Finalize chapter 1 -2	6																																										
2.6	Advisor consultation	2																																										
2.7	Chapter 1 - 2 submission	1																																										
2.8	Preparing feasibility presentation	8																																										
3	**Document Chapter 3**																																											
3.1	Draft chapter 3	4																																										
3.2	Team Discussion 4	2																																										
3.3	Draft UX/UI	8																																										
3.4	Advisor consultation	2																																										
3.5	Finalize UX/UI	10																																										
3.6	Advisor consultation	2																																										
3.7	Chapter 3 submission	1																																										
3.8	Preparing design presentation	8																																										
4	**Project Implementation**																																											
4.1	Developing	400																																										
4.2	Unit Testing	80																																										
4.3	Deployment (alpha)	10																																										
4.4	Advisor consultation	2																																										
4.5	Integration test	10																																										
4.6	Deployment (beta)	2																																										
4.7	Acceptance test	20																																										
4.8	Deployment (Production)	3																																										
4.9	D-Day presentation	8																																										

Figure 2.5: Gantt Chart of the implementation plan

Our Implementation plan is divided into 4 phases that includes:

- **Proposal Phase** During this initial stage, we engage in discussions to determine and finalize the topic for our project.
- **Document Chapter 1-2** In this phase, we delve into the conceptualization of ideas, feasibilities, conduct research, identify target users, and consult with advisors to gather essential information.
- **Document Chapter 3** This phase is dedicated to the creation of the project document, encompassing the development of diagrams, user personas, and the design of our application's user interface.
- **Project Implementation** The final phase involves a concentrated effort on coding, testing, and deploying our application.

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

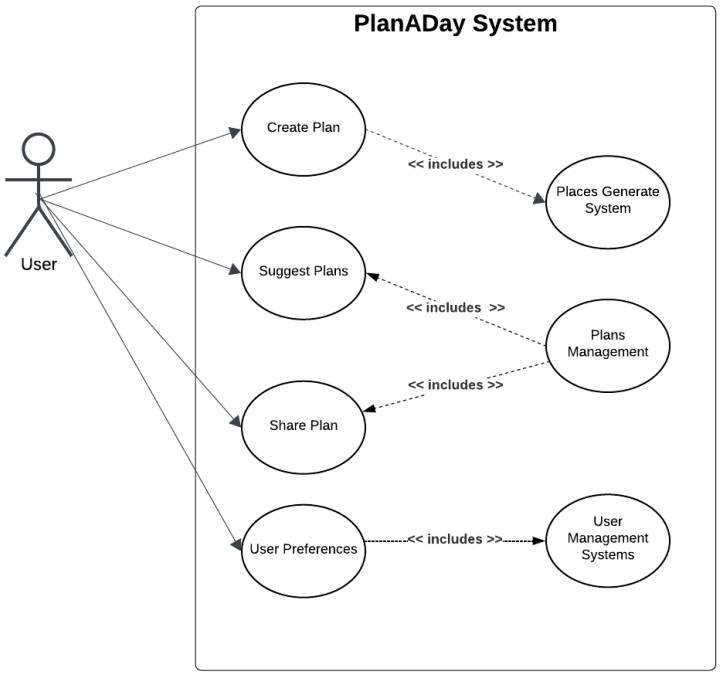

Figure 3.1: Use case diagram

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

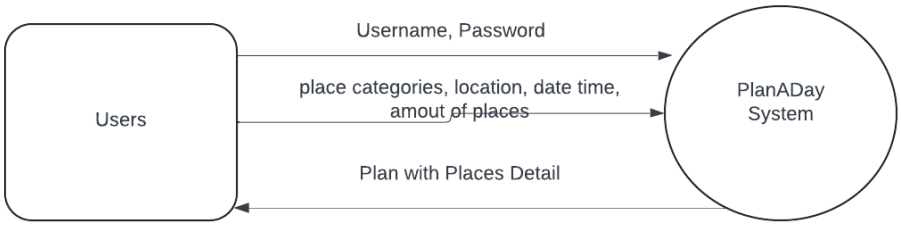

Figure 3.2: Context diagram

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

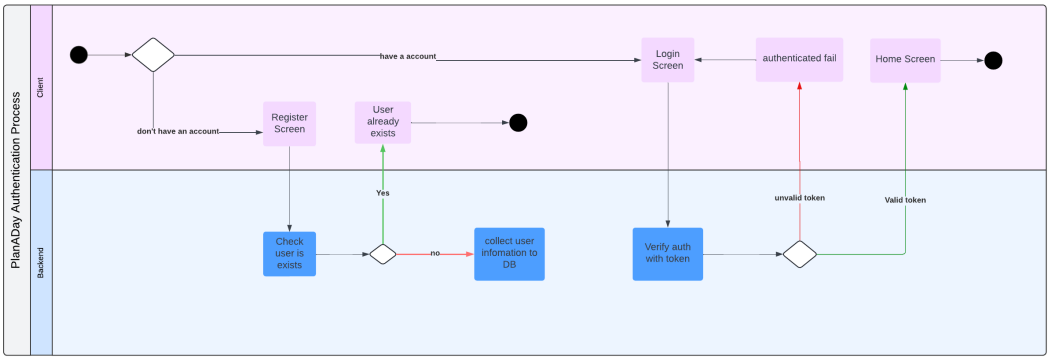

Figure 3.3: Activity Diagram of Authentication Process

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

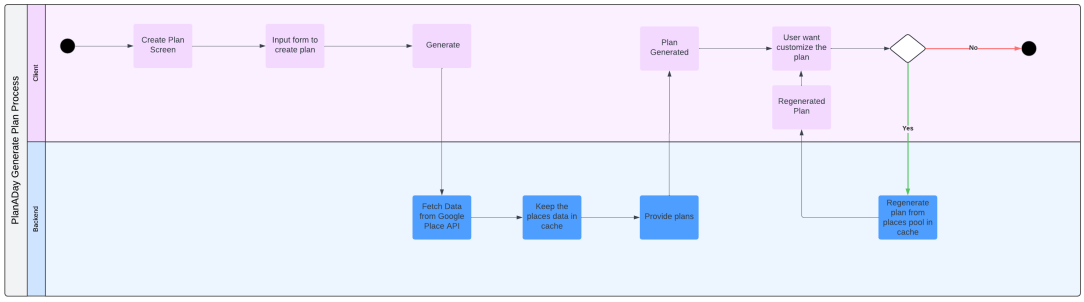

Figure 3.4: Activity Diagram of Generate Plan Process

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

3.5 Database design

3.5.1 Relational Database

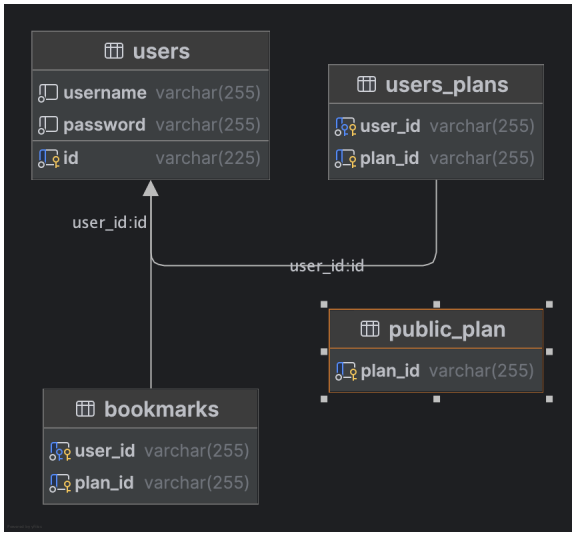

Figure 3.5: Relational Database Schema

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

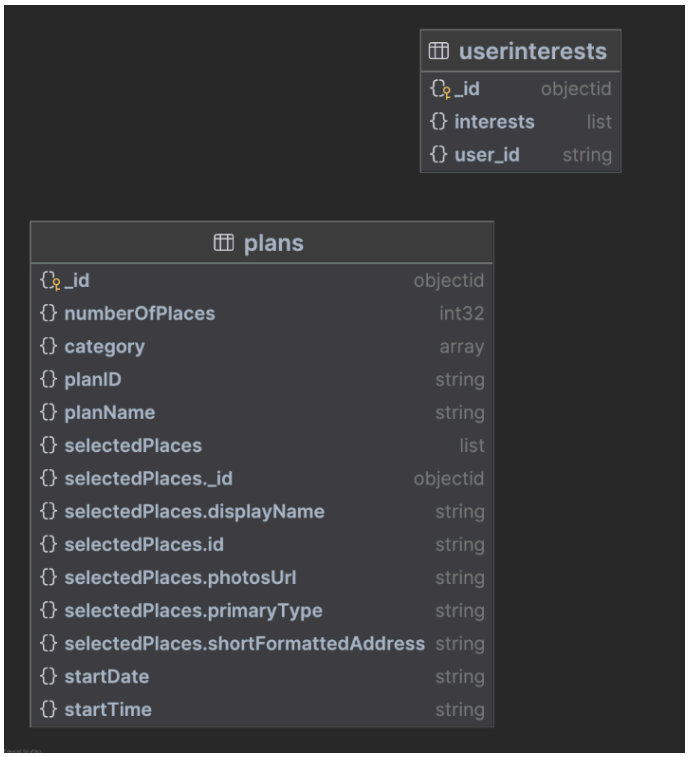

Figure 3.6: Non-Relational Database Schema

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 core aspects of the system's functionality, covering its system architecture, primary functions, planning, and testing results that defined the application's capabilities. The first part covers the architecture of the system in this project and the second part covers the main functionality of the system.

4.2 System architecture

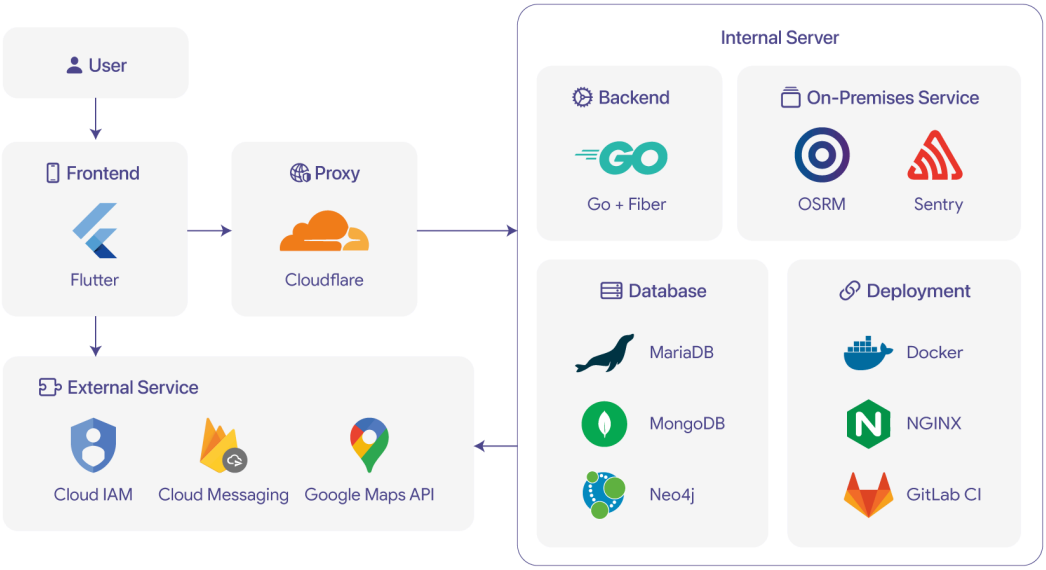

Figure 4.1: System architecture

The system will consist of two main components. The first component is frontend, the user-interaction part written as Flutter application which has functionalities of login, using Google OAuth API and has the map preview for user to search for places and navigate through routes. It will have to retrieve raw location data from the device GPS and call to backend using REST API. The second component is backend, which consists of the data manipulation and algorithmic part, which written using Golang. The backend use for retrieve route request from frontend, then All the connections between frontend and backend are proxies trough Cloudflare and internal NGINX reverse proxy. For the backend process, we use MariaDB for relational data store, Neo4j [?] for graph data store, The maps library, Google Maps API and self-hosted OSRM API, is used gather places, routes and any geographic information.

4.3 Main function

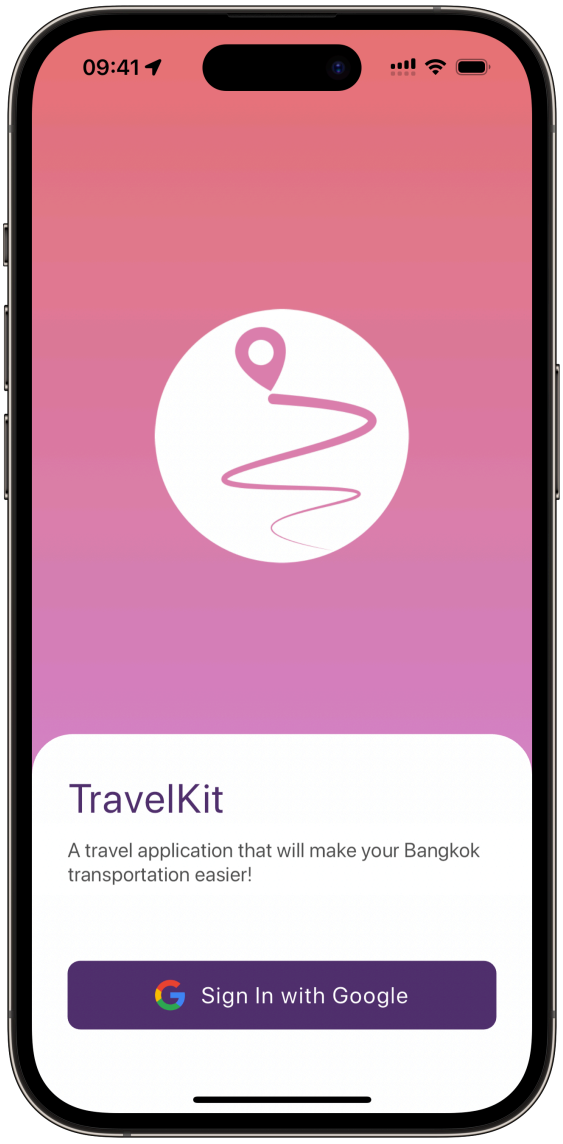

Figure 4.2: Welcome screen

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

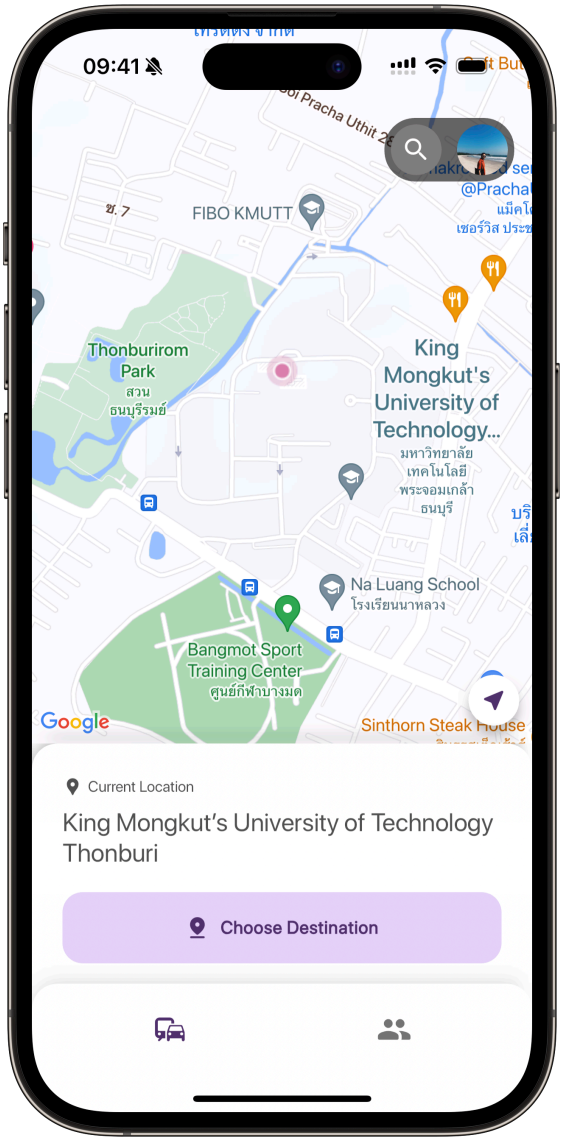

Figure 4.3: Home screen

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.

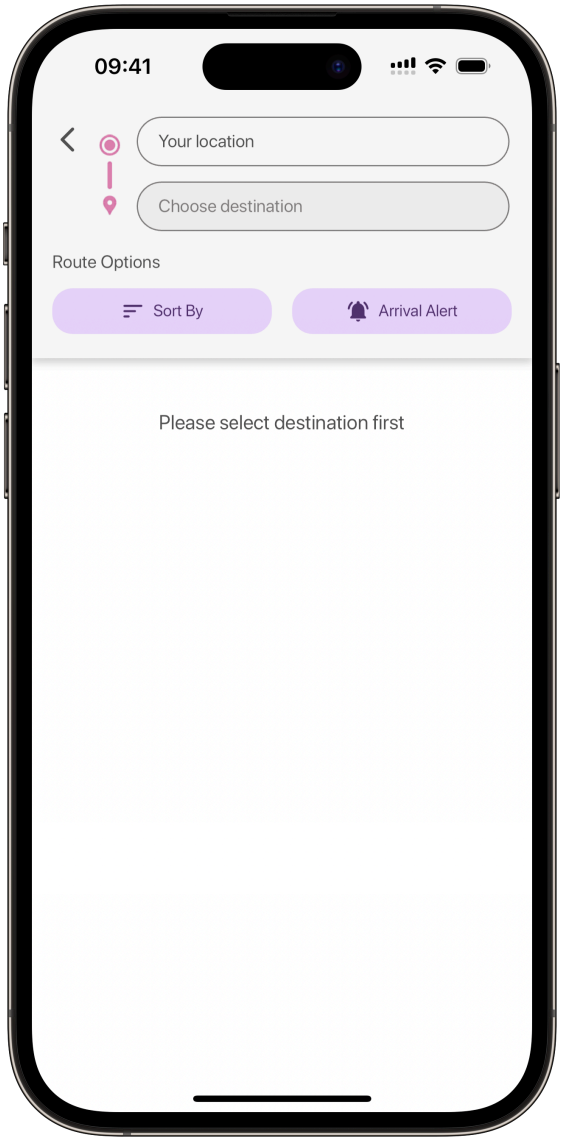

Figure 4.4: Choosing destination screen

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

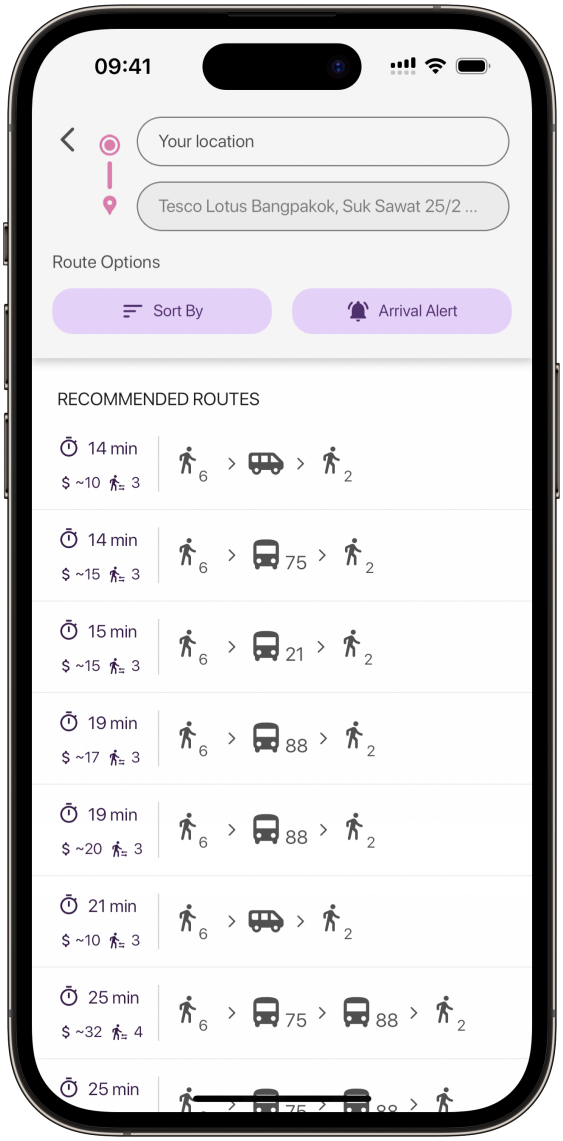

Figure 4.5: Choosing route screen

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.

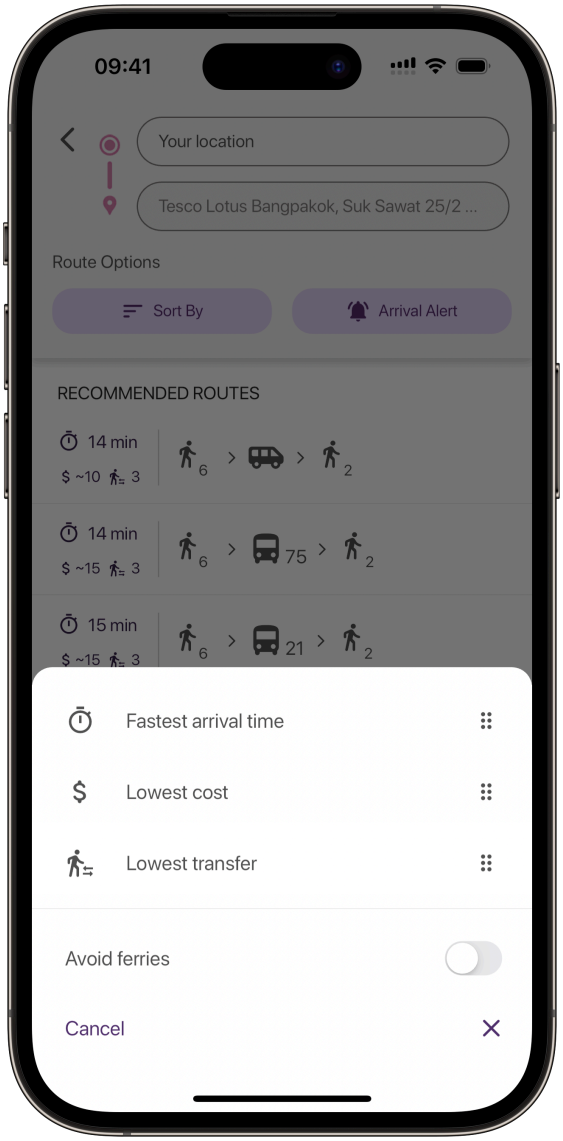

Figure 4.6: Route options

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.

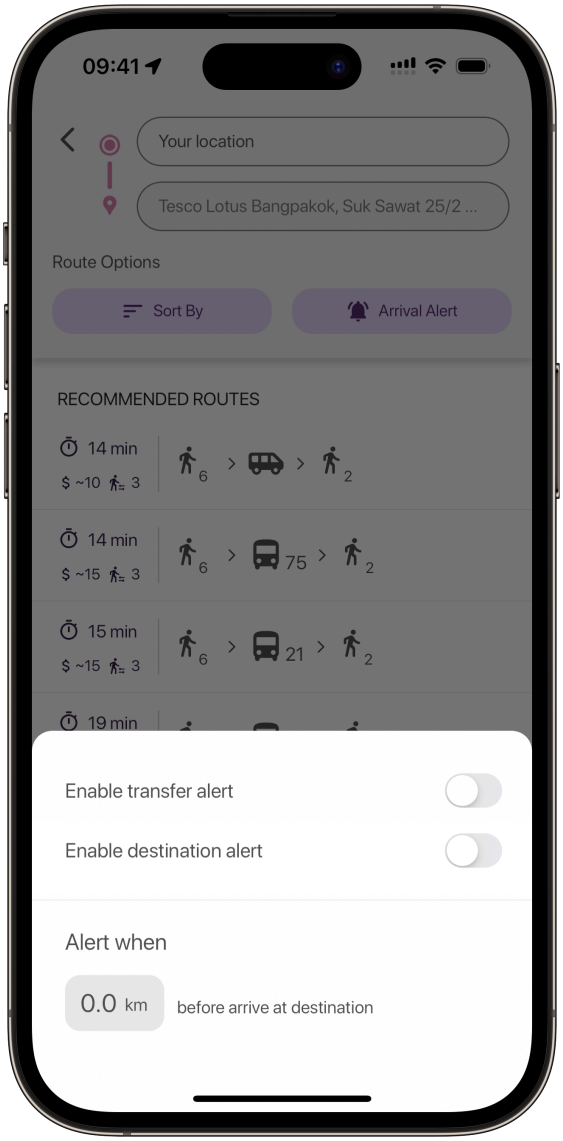

Figure 4.7: Arrival alert options

There are alerts that can be set when users are nearing the transfer point or the destination which help them reach their destination accurately.

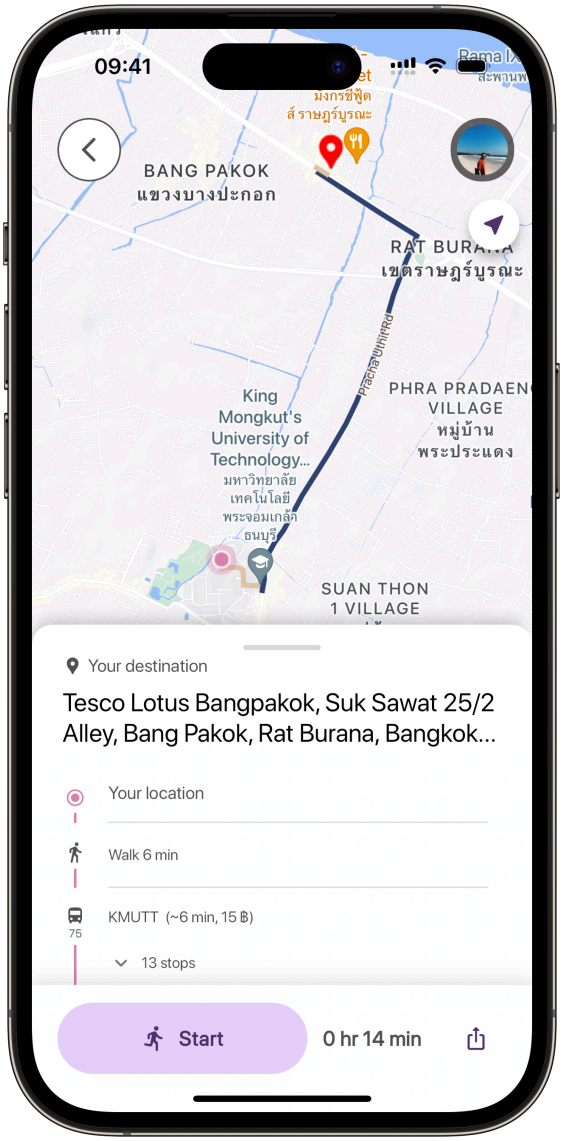

Figure 4.8: Selected route screen

Users will see the details of the selected route on the map once they select a route from the search route results. From the starting point to the destination, the route polyline is color-coded separately.

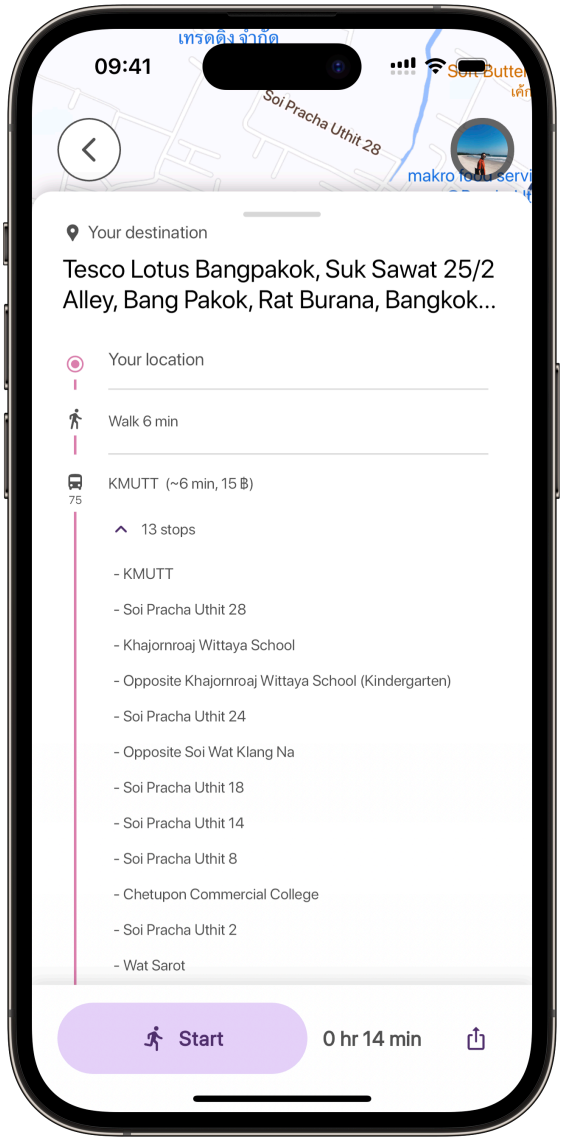

Figure 4.9: Selected route detail screen

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.

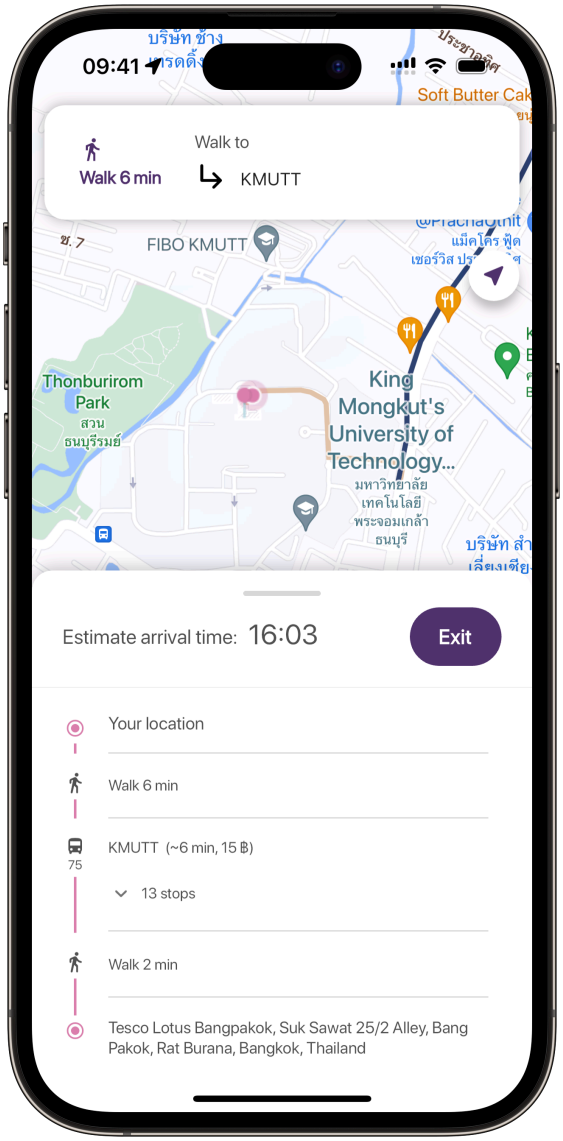

Figure 4.10: Real-time navigation screen

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.

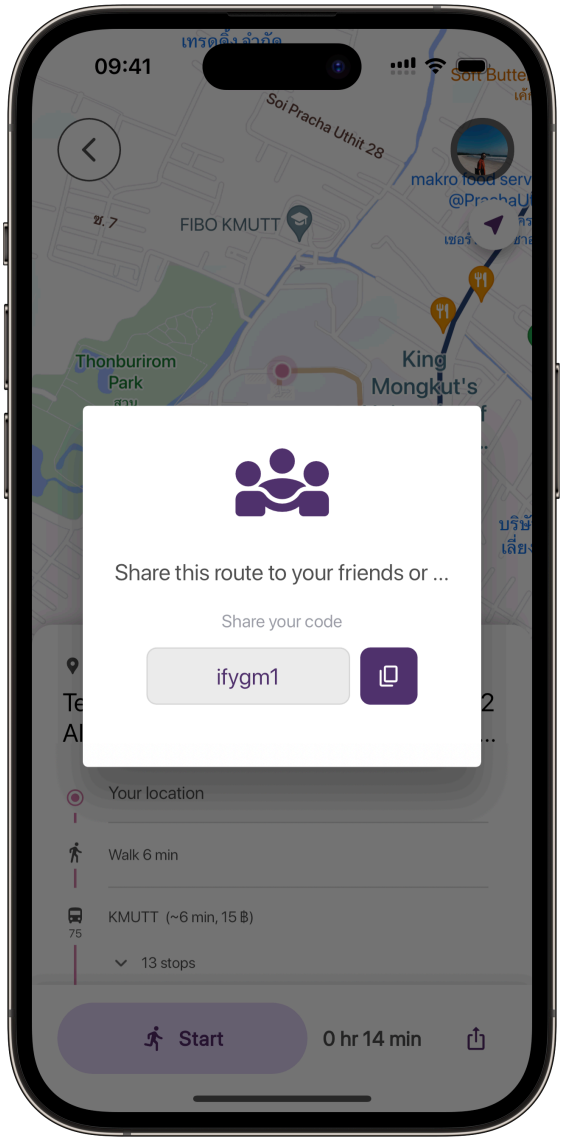

Figure 4.11: Share route screen

When users tap the Share button, they will receive the generated code and be able to share the code with others without having to search for the route again.

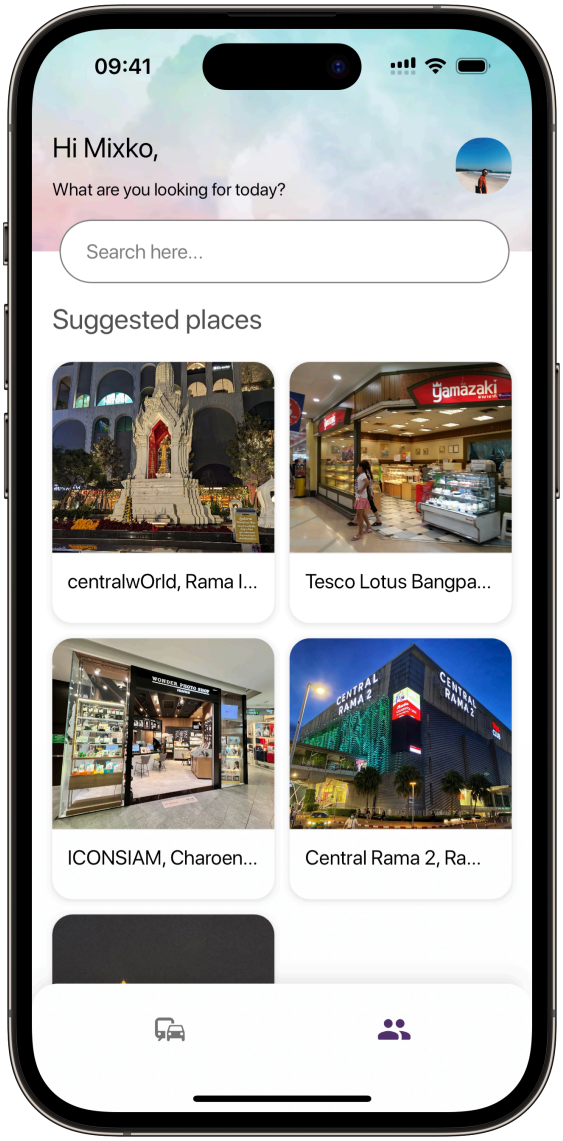

Figure 4.12: Community screen

This figure shows the suggested places, which are sorted by how often they are searched, and users can search routes based on the code that is generated by others that will navigate them to the shared routes.

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: Test plan and result

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.