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 possible without the support of Asst. Prof. Dr. Worarat Krathu and Dr. Watanyoo Suksa-ngaim contributed invaluable insight and experience.

Contents

Contents(CONT.)

List of Tables

Table **PAGE**

List of Figures

List of Figures(CONT.)

Figure **PAGE**

Chapter 1

Introduction

1.1 Background

Due to the large number of people who want to travel to remote areas, whether it be long-distance trips to different provinces or short trips to unfamiliar places, all of them need information for their trips. Especially for public transportation, which is complex and has multiple routes to choose from, some of which may lead to slower trips due to uncertainty in travel time. Despite the current technology such as Google Maps and Apple Maps, they only process the shortest route based on available data, but in some cases, there may be alternative methods of travel that is more convenient and match the user's travel preference, like lowest cost, minimize multi stop, or find the route that has activity along the way. [1]

1.2 Objectivs

1. To respond to the growing demands for travel among individuals in the society, particularly to unfamiliar destinations or unknown routes.
2. To improve the accessibility to travel information, allowing people to easily obtain the necessary details for their journeys. The information can best meet the needs of users by providing directions and recommendations. The information is diverse and collected to provide complete information for the general area.
3. To notify and suggest routes for navigating, attracting users to use our app. This in turn will contribute to the development of the local economy.

1.3 Expected Benefits

1. Users can use our app to receive notifications when they are approaching their destination.
2. Users can choose their own preferred travel route and the best type of transportation.
3. Users can share and recommend their own travel information on the app's community.

1.4 Scope

1. It is a mobile application that designed to help improve efficiency in traveling from one point to another within Bangkok.

2. It is suitable for users who can use mobile phones and understand how to use map applications.

3. It calculates distances, travel routes, and estimate travel costs so that users can choose their own mode of transportation.

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

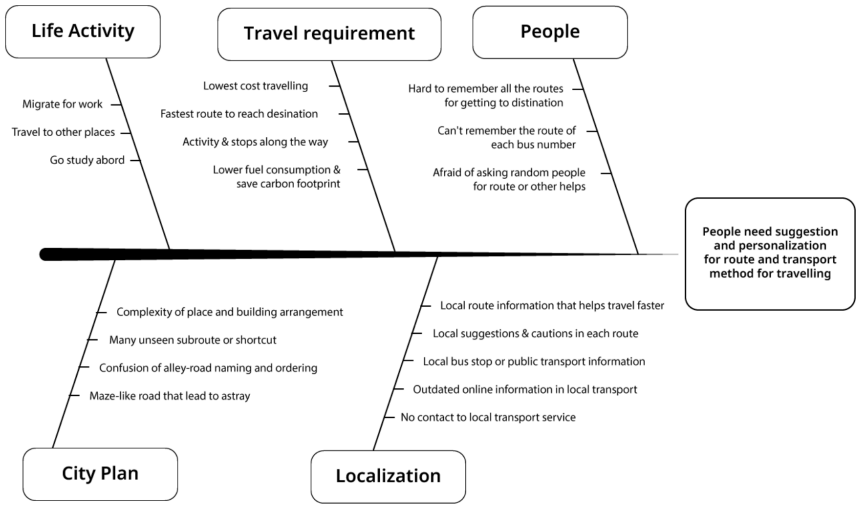

Figure 2.1: Problem diagram

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.

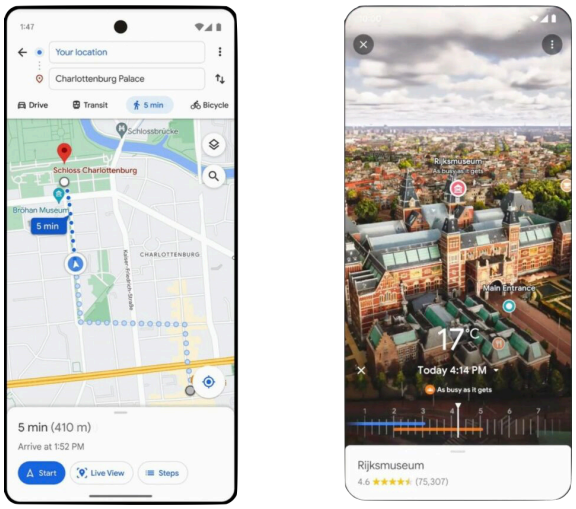

Figure 2.2: The Google Maps application

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.

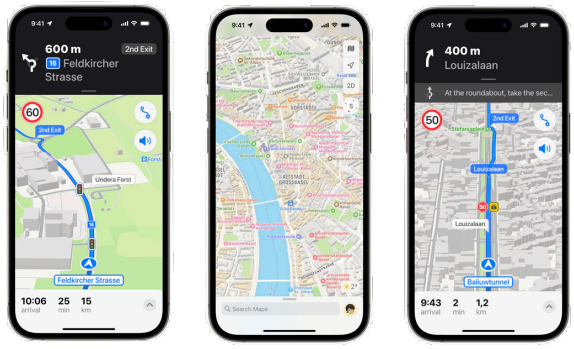

Figure 2.3: The Apple Maps application

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.

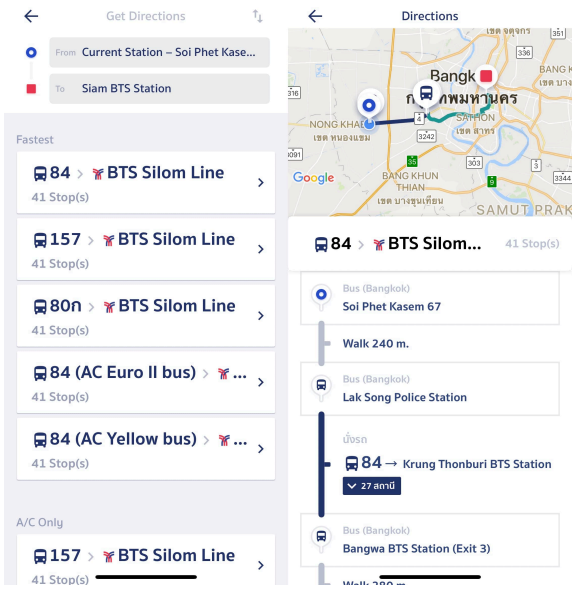

Figure 2.4: The ViaBus application

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 chapters covers three topics which are user requirement, business process designs, and system design. The user requirement part discusses the business needs for what users require from the system, The business process design part discusses the activities diagram, use case diagram, and context diagram of the application. Lastly, the system design consists of database design, and system architecture.

3.2 User definition

- People who aged around 12 - 40 years old and familiar with English language
- Lived in Bangkok and in the city crowded with technology-accessible people
- People who trying to travel into newly visited place without known directions
- People who want to browse alternative transportation method to reach their destination with their own condition like cost savings, faster route, or less transfer

3.2.1 User persona

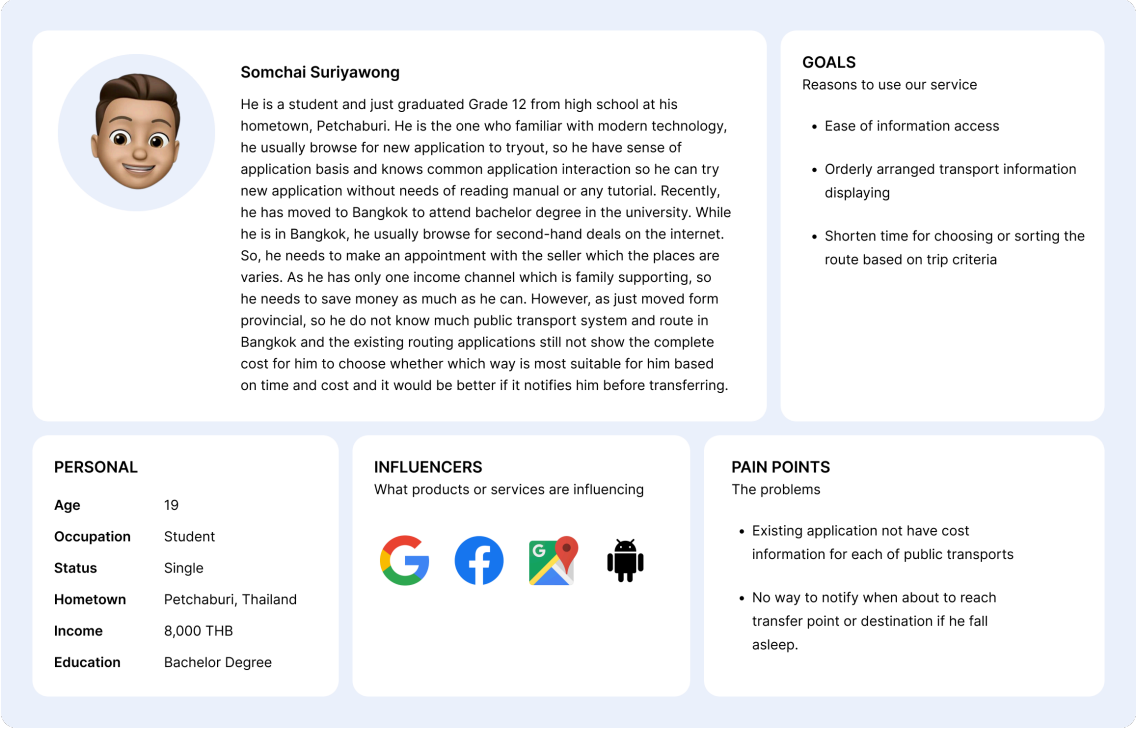

Figure 3.1: User one persona

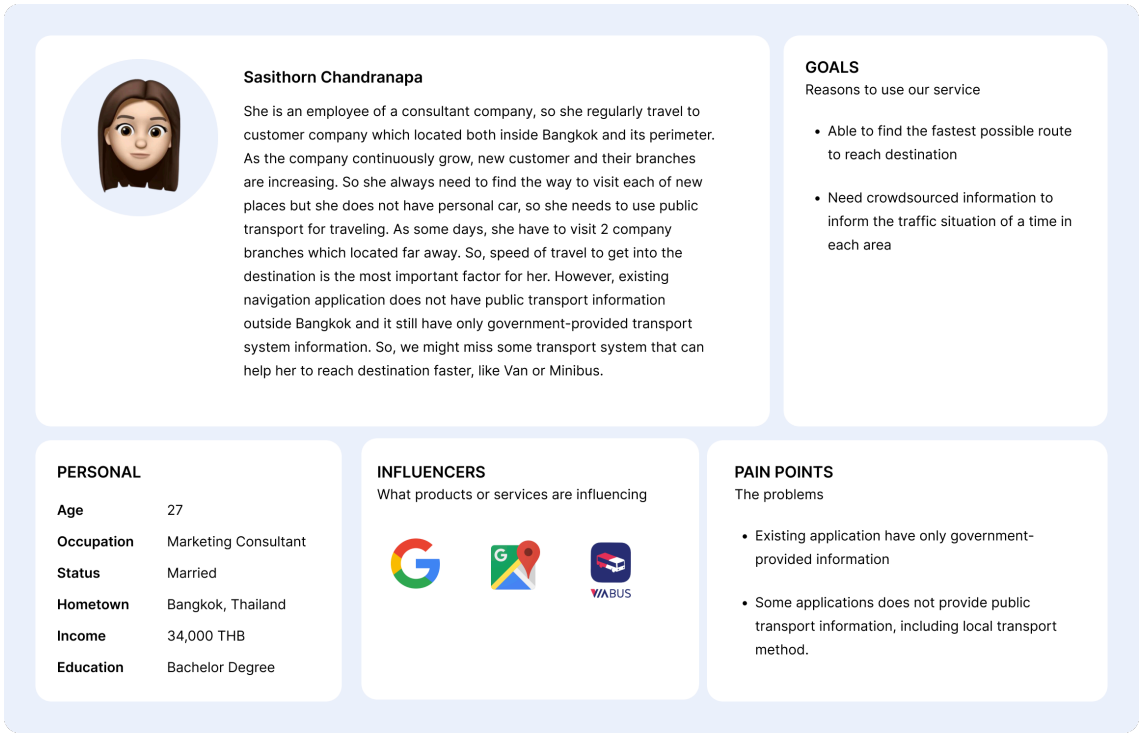

Figure 3.2: User two persona

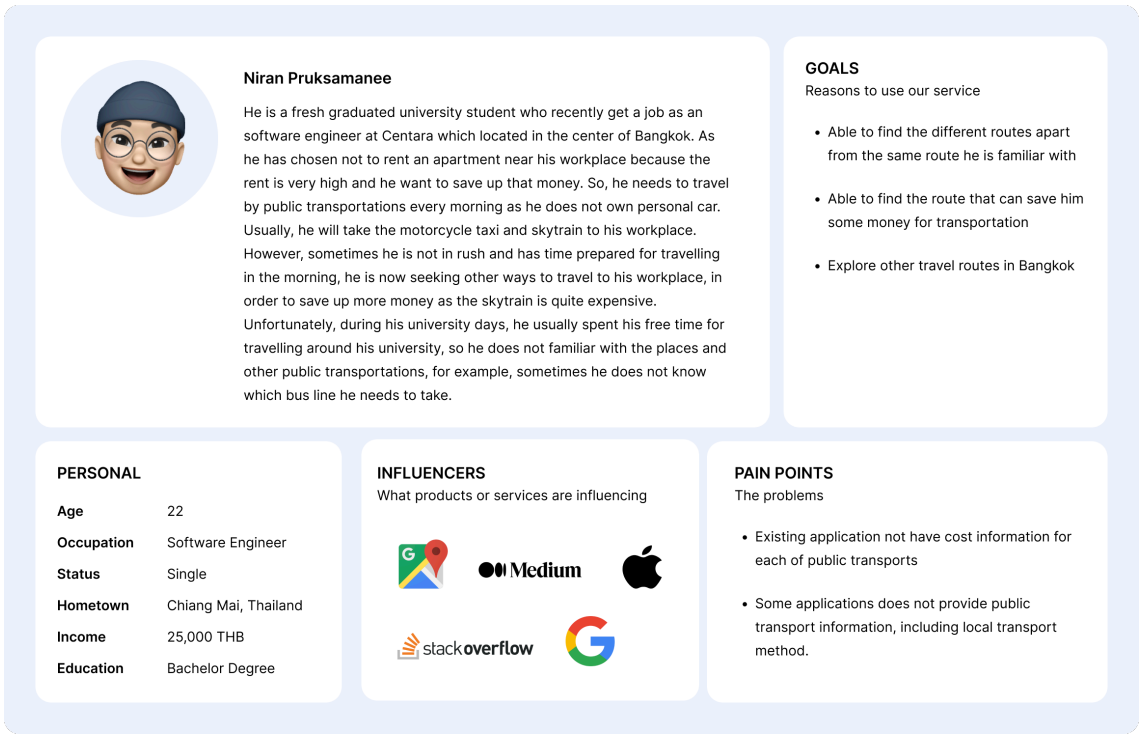

Figure 3.3: User three persona

3.2.2 User requirement analysis

- This application helps users to find the best route for traveling.
- This application uses map service provider and our public transportation data such as

vans, and mini truck to suggest the route for users.

- This application provides the community to share the routes of traveling of users.
- This application will provide a mobile application for users that use smartphones.

3.3 Use case diagram

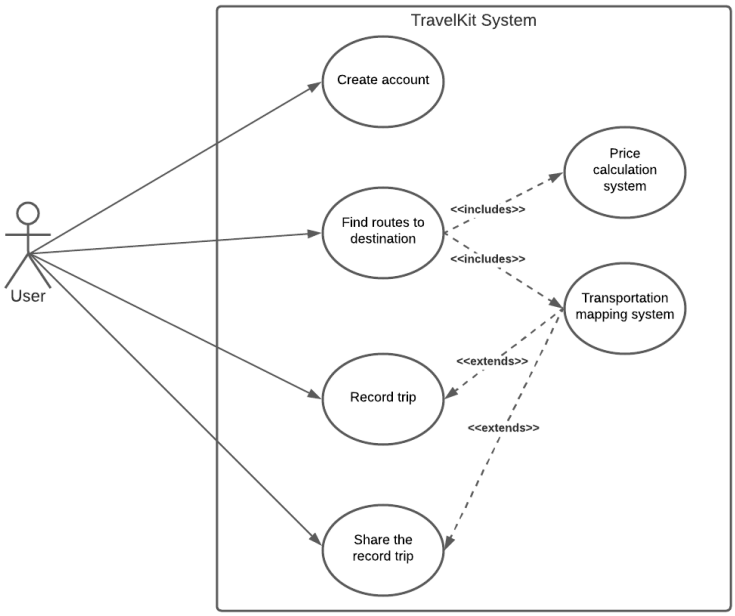

Figure 3.4: Use case diagram

The actor who is user in the routing system. The user must create an account to login to the system. After the user finds routes to the destination, the system will calculate the cost of traveling and display alternative routes using public transport data, such as vans and mini trucks, so that the user can make informed decisions. In addition, the user can record their trip after reaching the destination. Lastly, the user can share their trip to the community, along with the details of their travel, as well as comment and recommend other users.

3.4 Context diagram

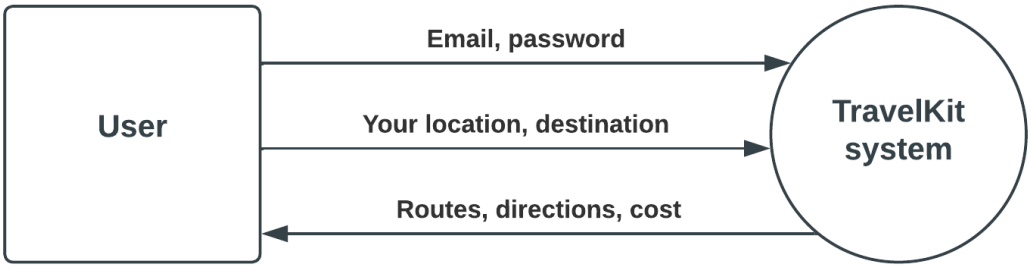

Figure 3.5: Context diagram

The context diagram shows the overall of the TravelKit system that required the name, password, user's location, destination. Then, the system will create choices of the routes, directions and calculate the cost of traveling to the destination.

3.5 Activity diagram

3.5.1 Travel process

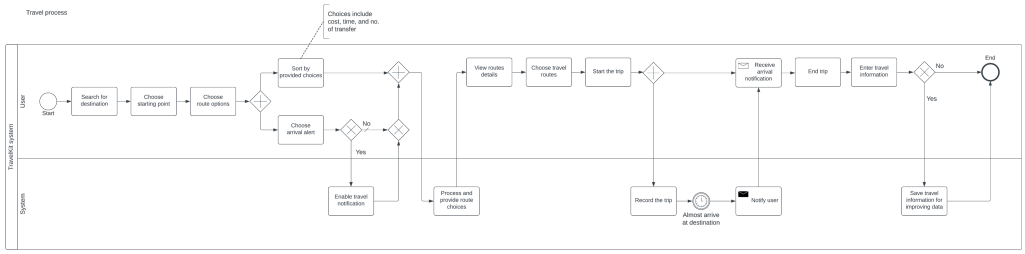

Figure 3.6: Travel process

The travel process for the TravelKit system starts when the user searches for the designated destination. Then, the user will be presented with the default starting point, which the user can either use their current location or select it again. After that, the user will choose their route options, which include the travel cost, time, number of transfers, and the arrival alert. If the user desires to have an arrival alert, the system will enable travel notification. Subsequently, the user can see the provided routes details, choose the route and then start the trip. Meanwhile, the system will start recording the trip and send the notification to the user when they are almost arriving at the destination. Finally, the user can choose either to share their travel route or not. If they choose to do so, the system will save the travel information to improve our system data and user can share to other people in community.

3.5.2 Community process

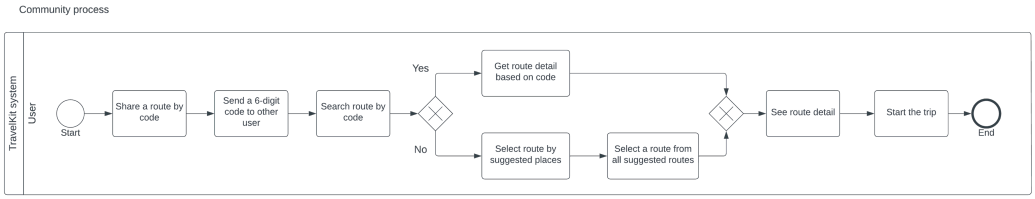

Figure 3.7: Community process

The community processes for the TravelKit system can be divided into two sub-processes. The first process is for the users who desire to search for routes, which can enter the code to get exact route detail. The second process is for users who wish to find for routes or places recommendations. The process starts when the user select the destination. Afterwards, the user can view the recommended routes and then select the route they preferred.

3.6 Database design

3.6.1 Relational Database

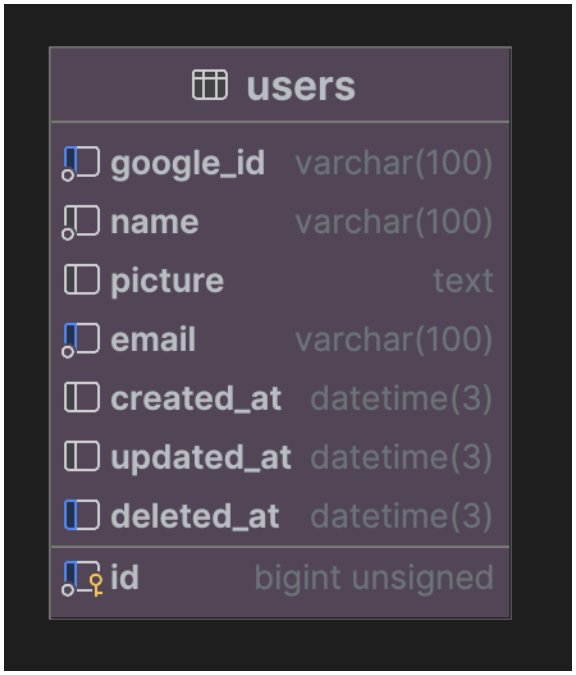

Figure 3.8: Relational Database Schema

Our database relies on MySQL to store user data efficiently. It is intricately connected with MongoDB, particularly in managing community-related information. This dual-database approach enhances functionality, ensuring seamless integration between individual user data and community features, contributing to a robust and scalable system.

3.6.2 non-relational database

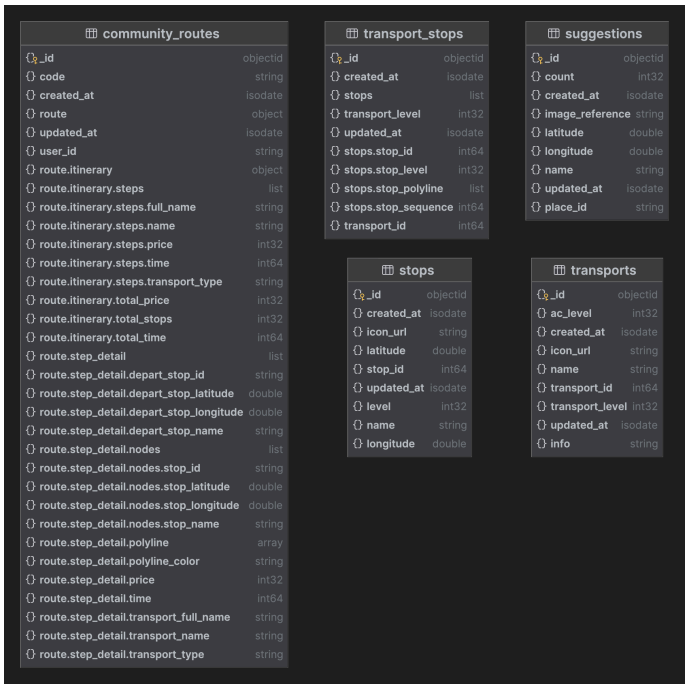

Figure 3.9: Non-Relational Database Schema

MongoDB is used for storing transport data, stops, and each stop related to each route. It includes:

- **Transports** for storing all data about transport types and transports
- **Stops** for storing stop data, such as bus stops, train stations, and ports
- **Transport Stops** for storing all stops that relate to each transport sequentially
- **Community Routes** for storing shared routes from each users
- **Suggestion** for storing the most frequently selected destinations by users

3.6.3 Graph Database

The graph database used for storing routes and stops relationship for using the algorithm to find shortest path based on user queries. The graph consist of nodes and relations which is the following node types:

- **STOP** For storing stop or station of all transportation type (e.g. National Theatre, Sanam Luang Bus Terminal, Hua Lamphong, Opposite Tha Phra Chan)
- **ARL** For each line of Airport Rail Link system
- **BRT** For each line of Bus Rapid Transit system
- **BTS** For each line of BTS Skytrain system
- **BUS** For each route of BMTA bus line
- **CHE** For each route of Chao Phraya Express Boat
- **KPS** For each route of minibus and local mini truck
- **MRT** For each line of Metropolitan Rapid Transit
- **SRT** For each line of train

Concerning relationships, the project incorporates various types tailored for tasks such as shortest path finding [3], performance optimization, and conditional querying. These relationships are categorized as follows:

3.6.3.1 GOTO relationship

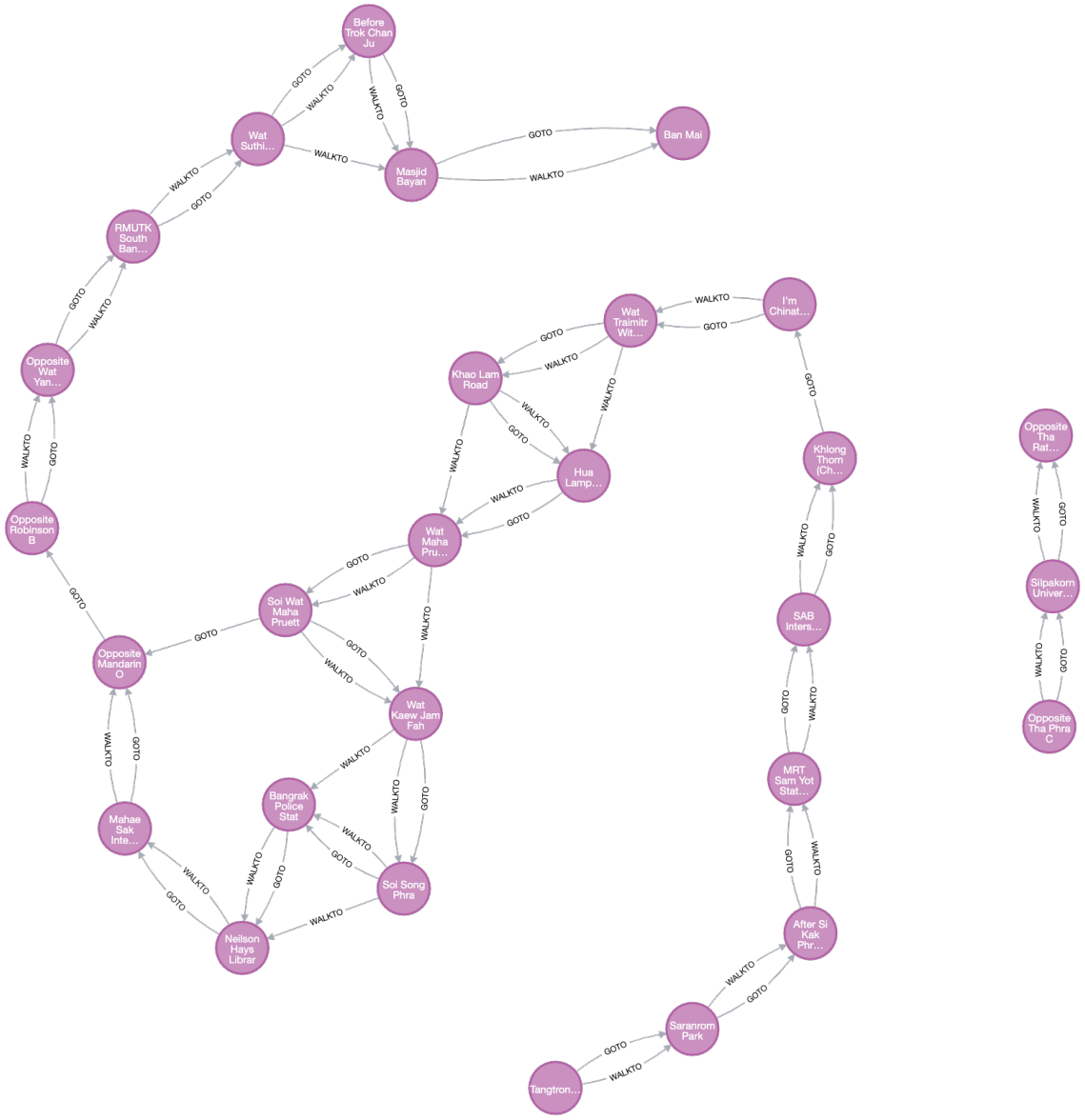

Figure 3.10: GOTO relationship

The relation of each two stops that are next to each other and have at least one transportation that pass from one to another (for example, from Sanam Luang stop to National Theatre stop) consider as one-way direction of each side of the road. This relation used for specifying the stop that are available for each bus line to transfer to another

3.6.3.2 WALKTO relationship

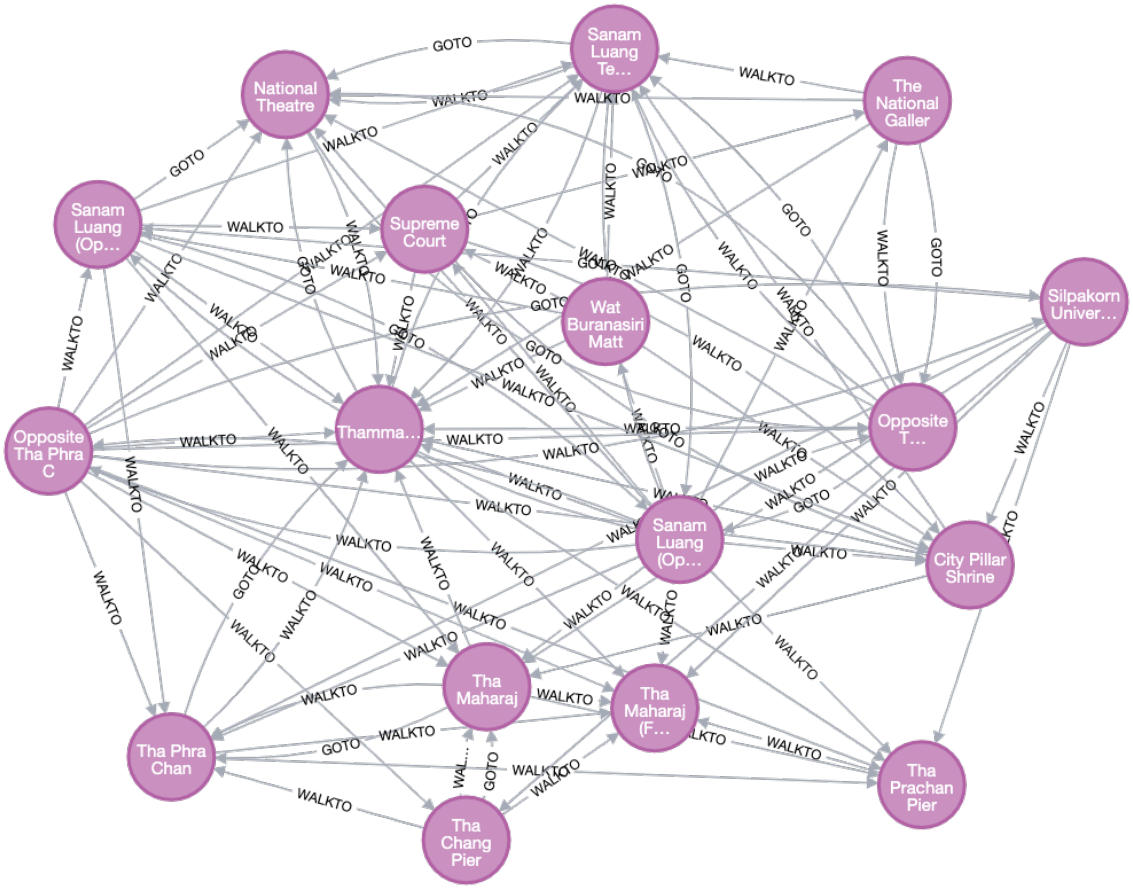

Figure 3.11: WALKTO relationship

As GOTO relation might not enough for finding the best suitable route, since some stops may not next to each other but user can transfer, walk or reroute between bus line that are in opposite direction for shorter distance. So we create WALKTO relation for all pair of stops that are walkable within 500 meters for querying the route which user can transfer to the nearby stop which are not directly next to each other.

3.6.3.3 STOPAT/STOPBY relationship

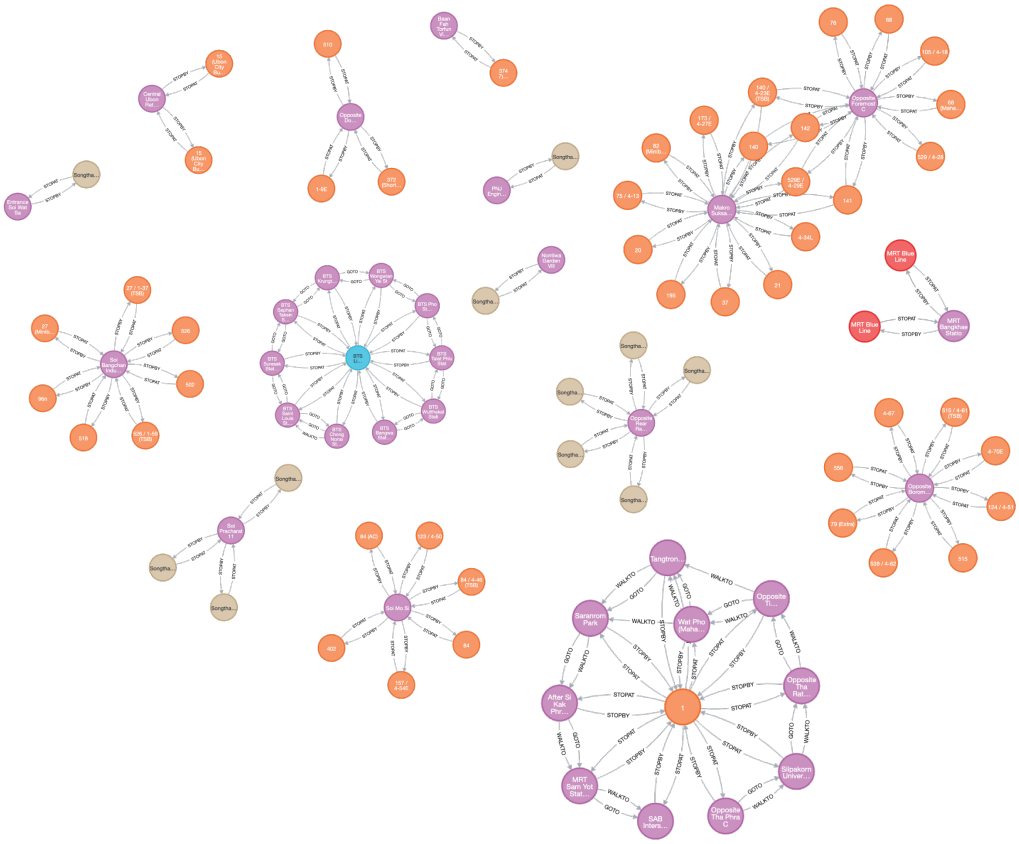

Figure 3.12: STOPAT/STOPBY relationship

The relationship represent that each transportation line (e.g., bus line, BTS line) are stop at each stop/stations. Each stop are stoppable from many transportation line and each transportation line can pass trough and stoppable at many stop as well.

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

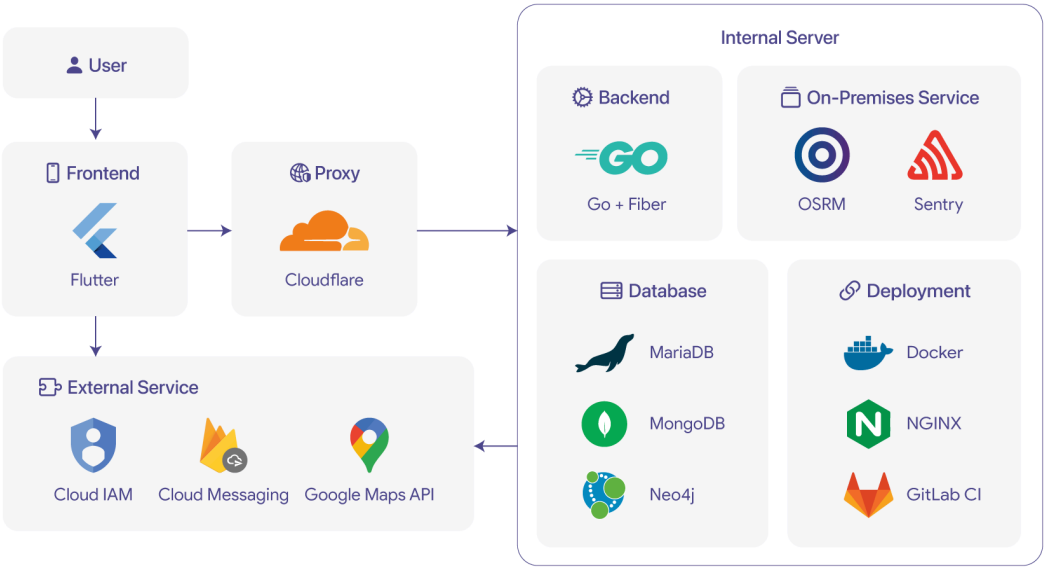

Figure 4.1: System architecture

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 [2] 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

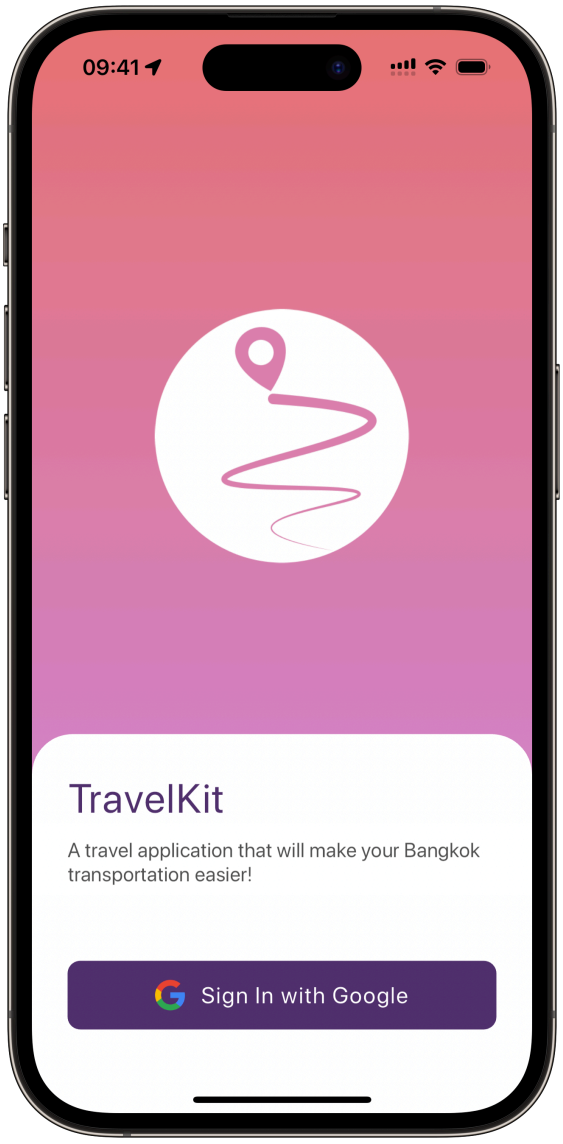

Figure 4.2: Welcome screen

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

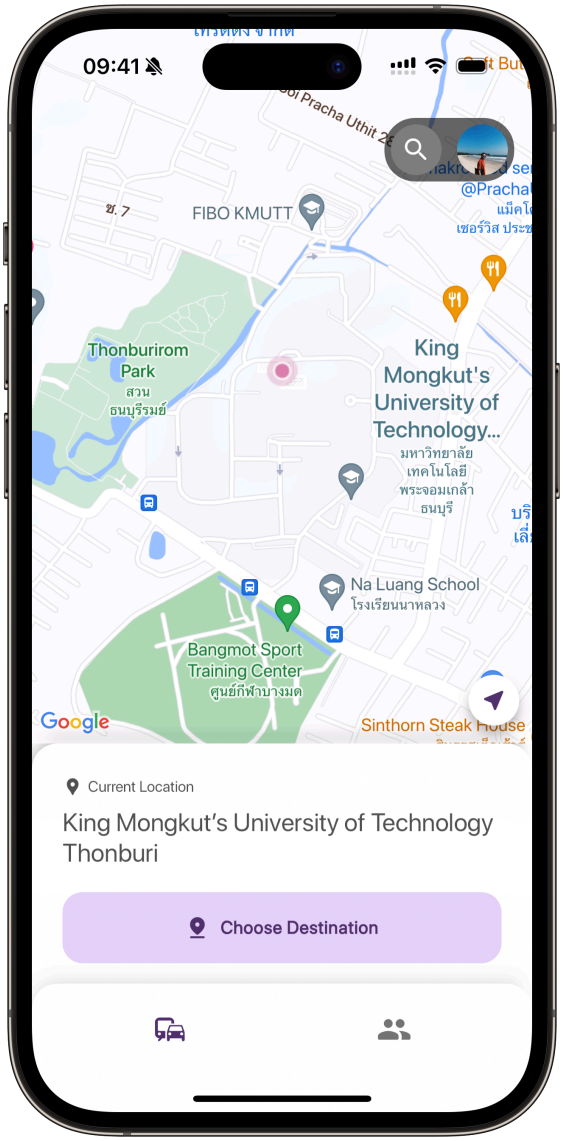

Figure 4.3: Home screen

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.

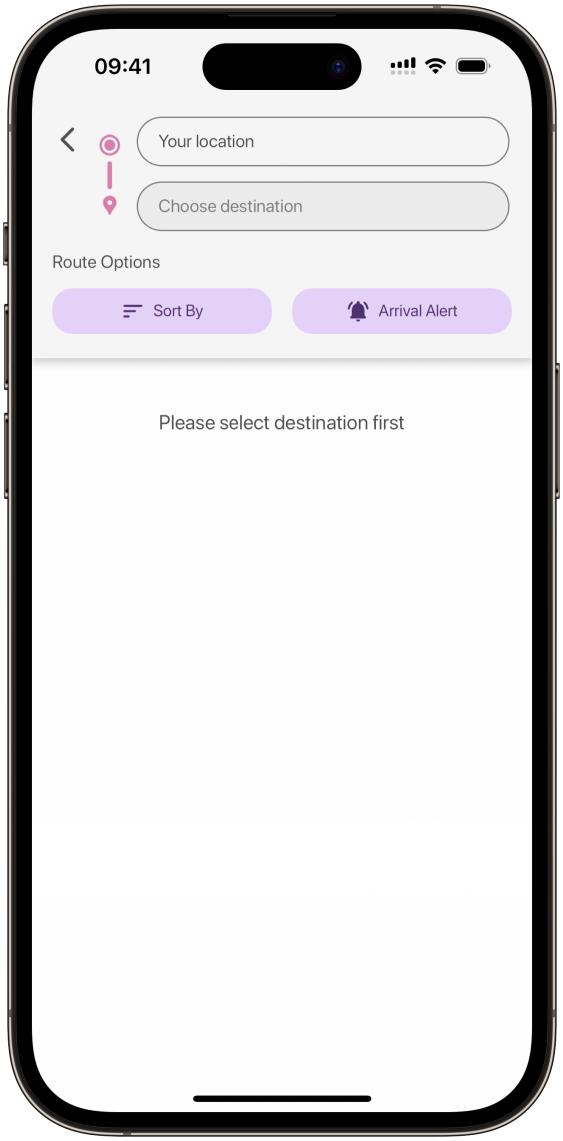

Figure 4.4: Choosing destination screen

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

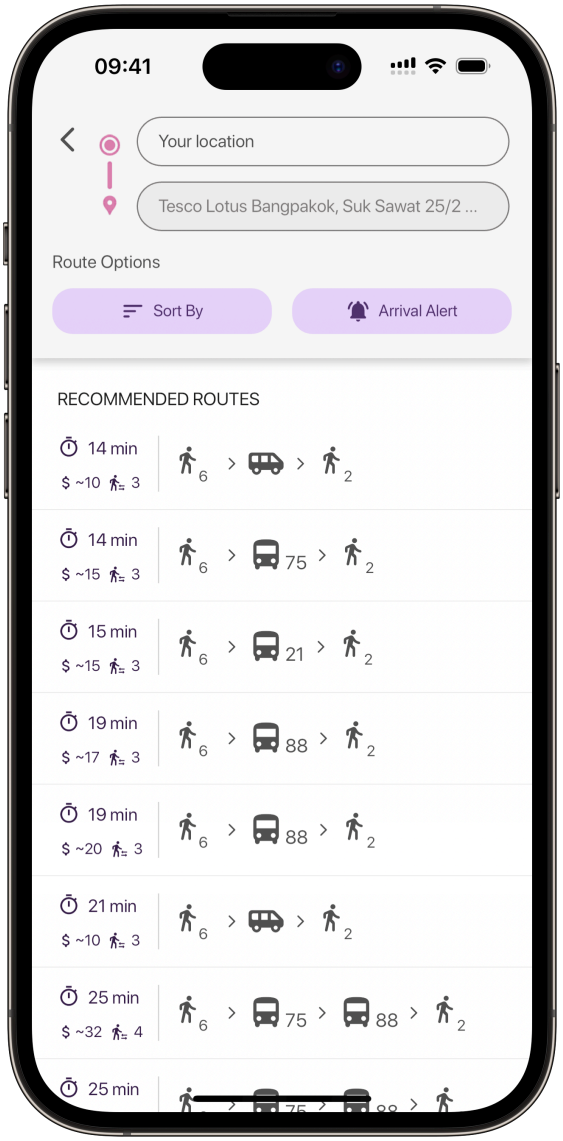

Figure 4.5: Choosing route screen

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.

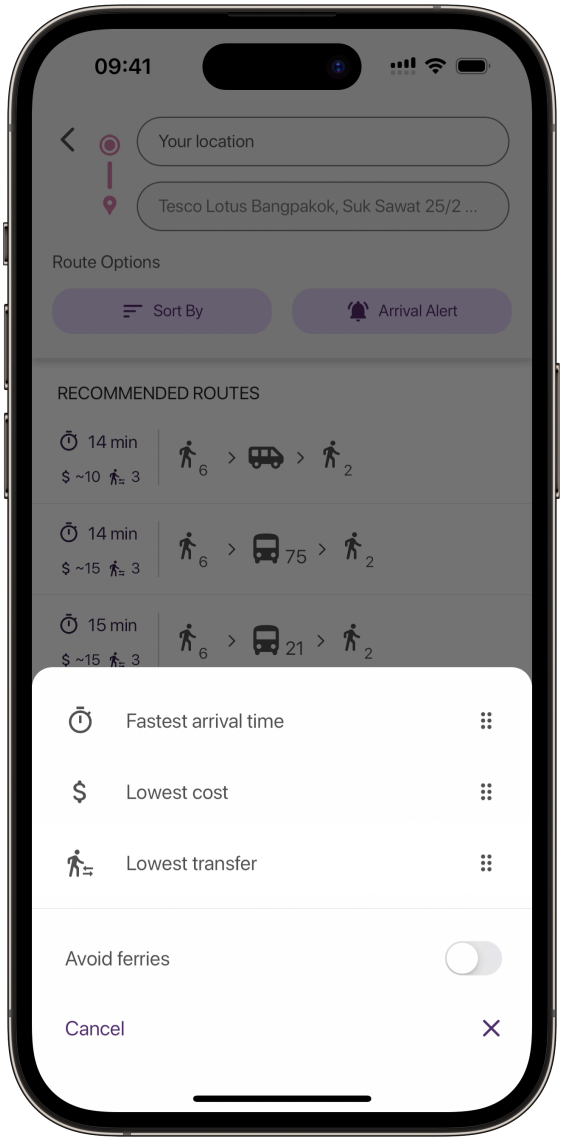

Figure 4.6: Route options

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.

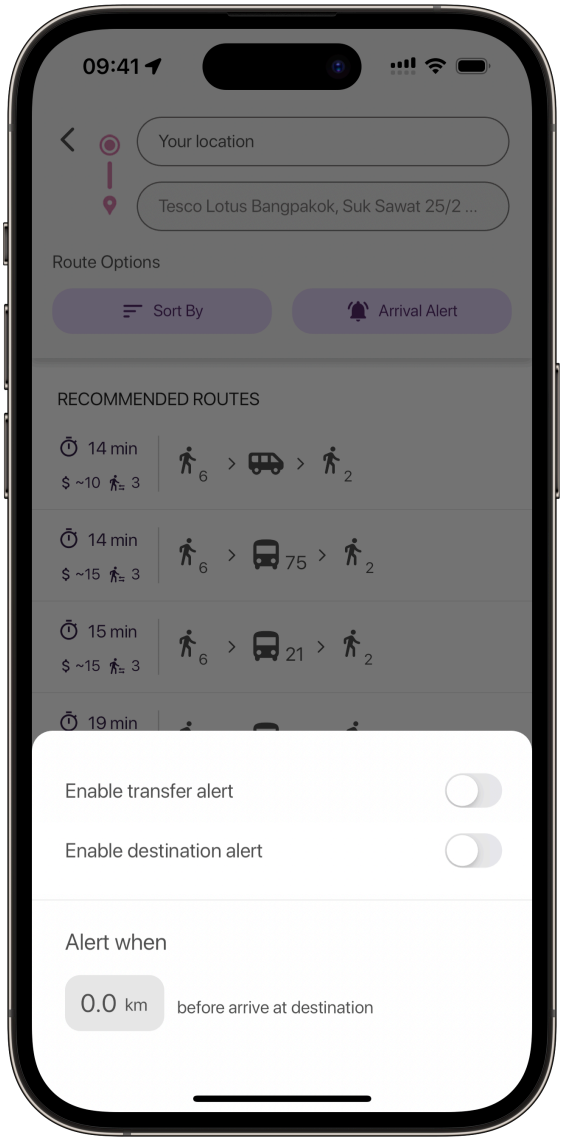

Figure 4.7: Arrival alert options

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

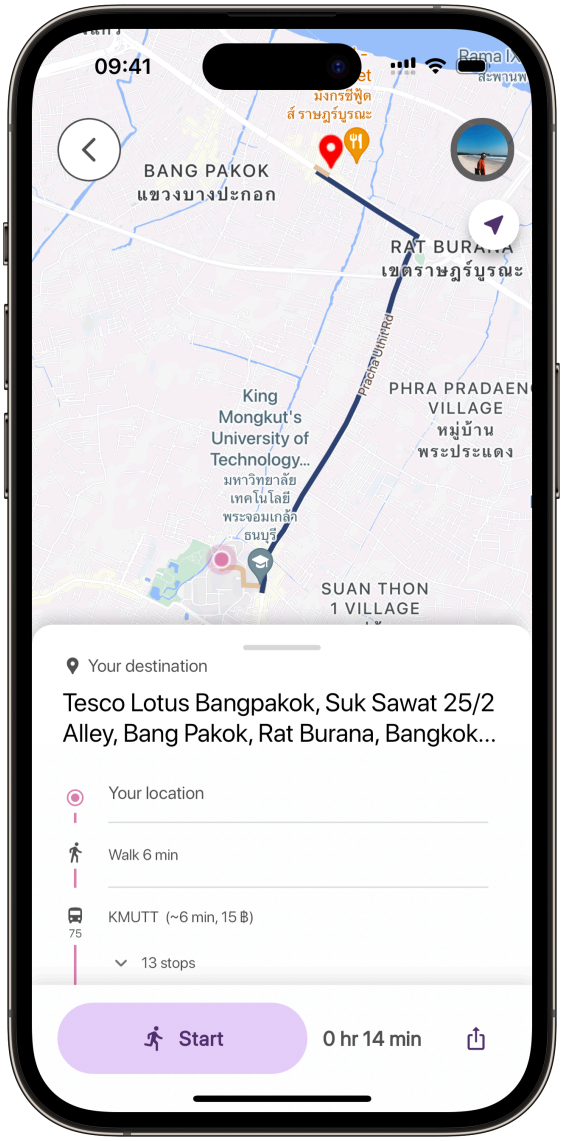

Figure 4.8: Selected route screen

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.

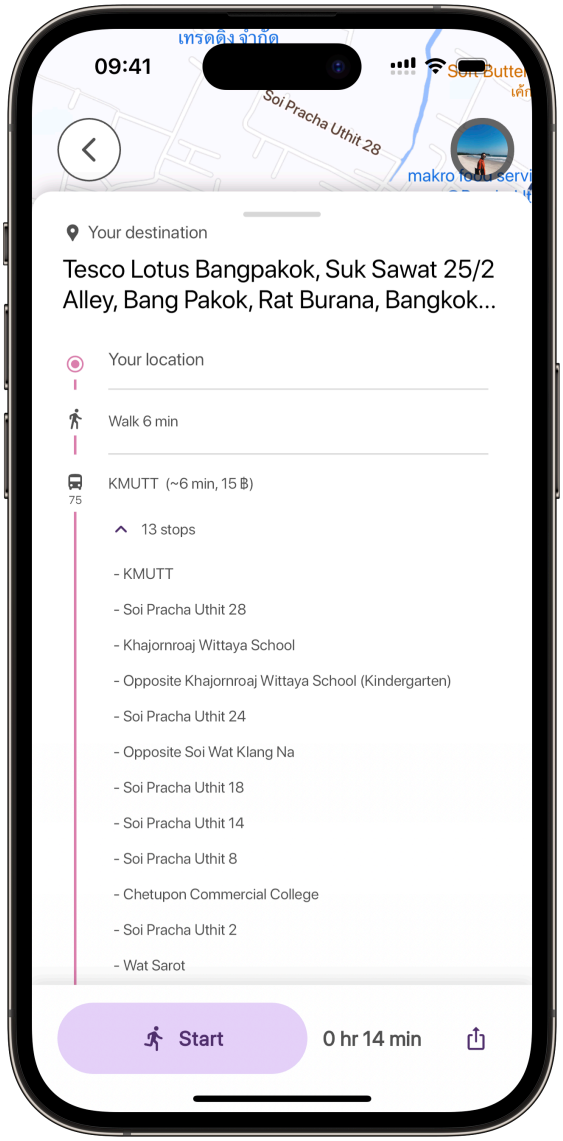

Figure 4.9: Selected route detail screen

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.

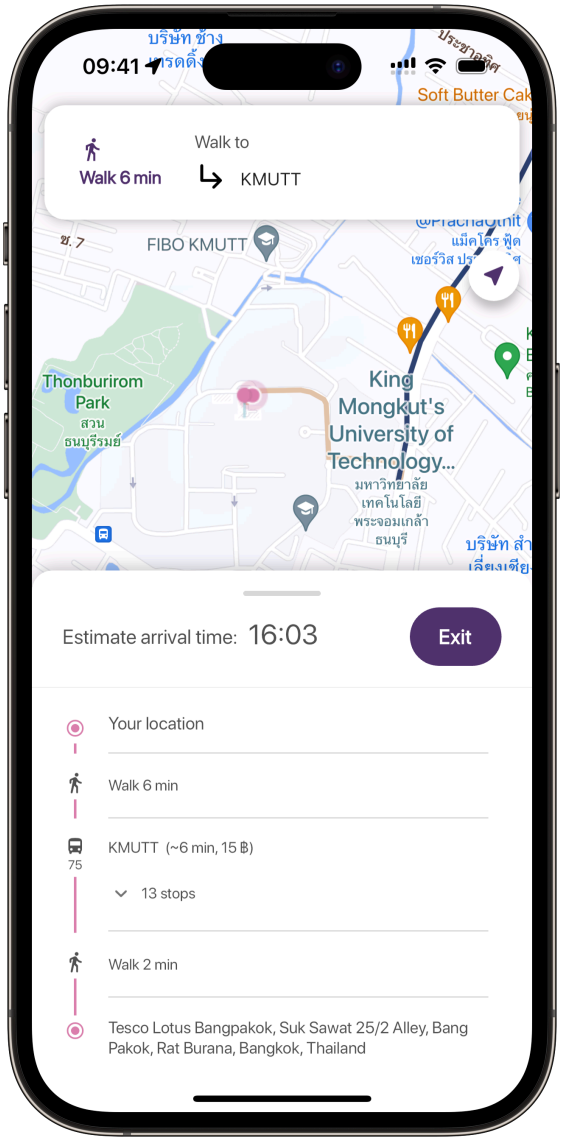

Figure 4.10: Real-time navigation screen

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.

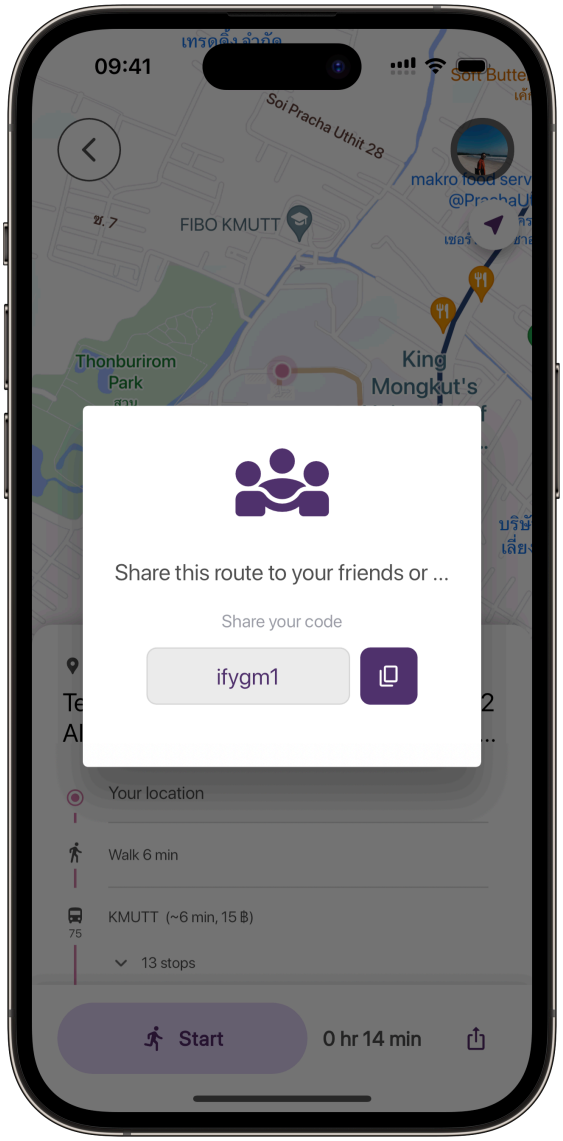

Figure 4.11: Share route screen

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.

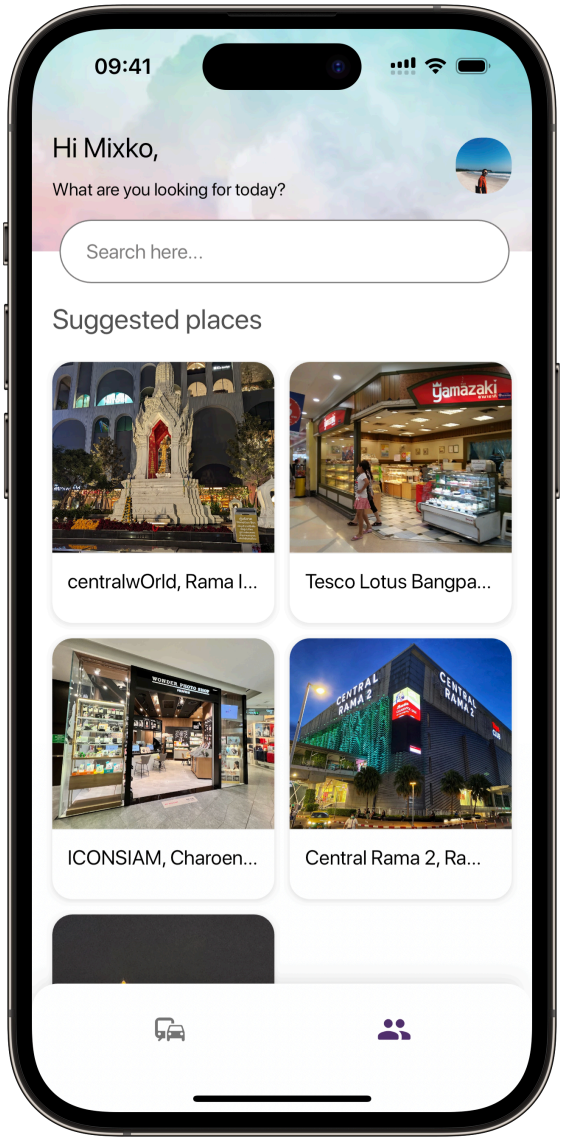

Figure 4.12: Community screen

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.

References

[1] BLT. Statistics of using public transport of bangkok people. URL: https://www.bltbangkok.com/poll/4055, October 2017.

[2] Adam Cowley. Implementing neo4j with a map. URL: https://neo4j.com/blog/journey-planning-why-i-love-cypher, November 2018.

[3] Ng Wai Foong. Build a subway journey planner using neo4j. URL: https://bit.ly/subway-journey-using-neo4j, September 2020.