Itinerary Planner for Tourism Mobile Application with Smart Trip Generation and Social Platform

BY

LOH WAI LEONG

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

MAY 2022

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: Itinerary Planner for Tourism Mobile Application with Smart Trip ____ Generation and Social Platform________________________________

Academic Session: MAY 2022

I LOH WAI LEONG

(CAPITAL LETTER)

declare that I allow this Final Year Project Report to be kept in Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

[illegible] [illegible]

(Author's signature)	(Supervisor's signature)
Address:	
No,26, Jalan Kobis, Taman Sejaterah,	
Mambang Diawan, 31950, Kampar,	Dr. Liew Soung Yue
Perak	Supervisor's name
Date: 08 Sept 2022	**Date**: 8/9/2022

Universiti Tunku Abdul Rahman			
Form Title : **Sample of Submission Sheet for FYP/Dissertation/Thesis**			
Form Number: **FM-IAD-004**	Rev No.: **0**	Effective Date: **08 SEPT 2022**	Page No.: **1 of 1**

Faculty OF Information and Communication Technology

UNIVERSITI TUNKU ABDUL RAHMAN

Date : 08 Sept 2022

SUMMISION OF FINAL YEAR PROJECT/DISSERTATION/THESIS

It is hereby certified that **Loh Wai Leong** (ID No: **18ACB06900**) has completed this final year project entitled **"Itinerary Planner for Tourism Mobile Application bwith Smart Trip Generation and Social Platform"** under the supervision of **Ts Dr Liew Soung Yue** (Supervisor) from the Department of **Computer and Communication Technology** , Faculty/Institute* of **Information and Communication Technology.**

I understand that University will upload softcopy of my final year project in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Your truly,

[illegible]

Mr Loh Wai Leong

DECLARATION OF ORIGINALITY

I declare that this report entitled "**Itinerary Planner for Tourism Mobile Application with Smart Trip Generation and Social Platform**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature	:	[illegible]
Name	:	Loh Wai Leong
Date	:	08 Sept 2022

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Dr. Liew Soung Yue, for providing a good opportunity for me to participate in the itinerary planner mobile application development. He tries his best to give us suggestions when we run into problems, to provide a clear direction for us to complete the FYP 1 project, and to encourage me throughout the project development process.A million thanks to you, Dr. Liew.

In addition, I would like to thank my friends who were willing to hear my problem and provide suggestions for me. Thanks for your patience and love. Finally, I must say thanks to my parents and my family for their love, support, and continuous encouragement throughout the course.

ABSTRACT

Itinerary planning is an important process before a traveler starts a trip. This is because travelers would wish their trips to go smoothly and so that they could enjoy themselves as much as possible. However, it is a complicated and time-consuming process if travelers do not know how to make the plans. Although travelers can search through the online resources, they may not fully explore the attraction or other worthwhile resources in the destination. Thus, the purpose of this project is to build an itinerary planner mobile application to provide a proper and easy tool for all travelers to make their plans before trips. The mobile application allows users to create a trip by entering the necessary data and managing it. Furthermore, there is an auto build route function that will help users compile the route of their trip. As a result, users can get the best route in the short time. If users are not satisfied with the arrangement, they are able to manage it by themselves. Nevertheless, the social platform is available in the itinerary planner mobile application. The social platform allows users to create posts, view posts and react toward the posts by liking the posts or commenting on them. This is a good start for all the users to get to know each other and share their travel experiences. Moreover, users are allow to promote their business by creating an attraction and publish to others.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

GPS	Global Positioning System
API	Application Programming language
SDK	Software Development Kit

Chapter 1: Introduction

1.1 Problem Statement and Motivation

As time passes, advanced technology evolves. The Internet will not be excepted. With the internet, there are many travelers starting to travel around the world. This is due to the internet making human life easier and simplifying the process of travel. Traveles are able to perform searching on the online platform to explore some of the information regarding the places that they are interested in visiting. Thus, they are able to gain some of the knowledge regarding the preparation process as well as explore the attractions available in the destination before the trips. In addition, they are able to make the reservation process such as hotel reservation through the online platform instead of calling to the airline or with the help of the travel agency. Since most of the things needed to do during travel preparation can be figured out from online resources, travelers are able to plan their trips based on their interest instead of being controlled by the travel agency. However, **travelers may not have the experience to explore the destination information available on the online resource**. This is because there may have incorrect and outdated information available on the online resources. Thus, users are required to filter out those inaccurate resources among all the available resources in order to plan the trip well. If travelers lack of experience to filter the information, the opportunity to encounter the problem which out of the users expectation will dramatically increase. Moreover, travelers may be overwhelmed by the plenty of information provided on the online platform during searching. As a result, they may miss out or misunderstand something.

Furthermore, **travelers may not be able to explore some of the hidden attractions available in the destination**. This is because the result of searching depends on how often the search keyword is in the web page [1]. Thus, travelers may miss some great information when searching from scratch. Although there are some guidelines provided to guide the traveler on how to plan a trip, it may be insufficient for fresh travelers because users may **not fully understand the description provided**. Furthermore, users may **require to do the extra work in order to record some of the detail of the planning**. Thus, the time taken to develop an itinerary plan will be longer and cumbersome. Nevertheless, all the activities planned for each day of travel **require**

users to imagine the path instead of visualizing it. As a result, the failure rate to comply with the planning will be higher. If the whole planning is not visualized such as pining the location on the map, listing out all the activities with the accurate estimated time, it may be difficult to explain the whole plan to other tripmates. As a result, argument and misunderstanding may arise during discussion. In addition, users may also have no idea to arrange all the activities wisely.

With the great advancement of technologies, itinerary planning is starting to evolve to the digital form such as mobile application and web-based application. At the current moment, there are few applications such as Booking.com, Go Holidays available on the market which further simplify the planning process. Users are able to make the reservation by using the application and the details of the reservation will be stored on the application. In addition, users are able to explore some of the attractions in the application by entering the destination on the application. However, there are some look holes on the applications. Most of the applications available in the market are **performing specific functions** like hotel reservation instead of integrating all the necessary functions together. Based on Smirnov[2], all the services offered in the itinerary planner application can be classified into four classes which are “Online booking”, “Information Resource”, “Location-based Services” and “Trip Journal”. In other words, these categories must integrate together and make them available on a single application.

1.2 Project Objectives

There are three main objectives of this project in order to develop an effective and user-friendly mobile application for all citizens. The first objective is to **analyse the existing mobile application, which is related to the itinerary planner on the Google Play Store or Apple Store, in order to identify the strengths, weaknesses, and limitations of the application**. With the change of technology, tourism is starting to be digitalized in the market. There are a few existing applications that assist travelers in managing their trips. Before we start the development process, we must study the existing applications in the market in order to have a better understanding of the trend and the requirements of the application. In addition, we are able to understand more about the users' needs through the analysis in order to develop an application that meets most of the travelers’ requirements.

Moreover, we aim **to develop an itinerary planner mobile application that assists users in managing their itinerary plans well by adopting the strengths of the existing application and resolving the weaknesses and limitations of the existing application with some of the novelty concept to enhance the mobile application**. The application will help travelers explore the vacation destination by providing them with information tailored to their needs. Users are able to add several activities to the itinerary plan if they are interested. The system will help users arrange their activities based on the path between one location and another. As the result, it able to arrange all the activities effectively.

Last but not least, we are going **to evaluate the functionalities and efficiency of both the itinerary planner mobile application by using verification plan and validation plan**. We are going to test each of the modules by using valid and invalid data. By doing so, we are able to ensure the application works as we expected. In addition, we are also share the mobile application to several users and let them experience it. After that, they will requested to fill up the google form in order to get the feedback from them.

1.3 Project Scope

This project is aimed to develop an itinerary planner mobile application for android users which provide a channel to them to manage and plan their trip. On the other hand, the information provided on the mobile application is focusing on Ipoh, Perak and Kampar, Perak (For demo purpose). In the following section (1.3.1 – 1.3.6), we will discuss the scope of each module.

1.3.1 Itinerary plan

Users are able to create a new itinerary plan or modify the existing itinerary plan on the application. When users create a new itinerary plan, they are required to enter some of the details, such as vacation destination, hotel location information, duration of trip, and so on. The system will be based on the information entered by the user to retrieve the related information for users. Next, they are able to explore the activities available at the vacation destination by selecting the categories provided, such as sightseeing, family, restaurants, and so on. The maximum number of activities to be arranged in one day is 6, of which 3 are for the main activities such as location visitations and the other 3 are for food and drink. After users finish the selection, users are able to allow the system automatically arrange the activities based on the path. Users are able to modify the itinerary plan if they are unhappy with it.

1.3.2 Social Platform

A social platform provides a way for all the users to interact with each other. Users are able to post some of the posts during the trip or share the experience to others through the social platform. So, other users are able to give a like or provide some of the comments under the post. In addition, users are also able to share the itinerary plan with others by posting the plan on the social platform. Thus, for those who have no idea and think the plan is nice, they are able to import the plan directly to their account.

1.3.3 Attraction Management

The mobile application provide a channel for all the users to promote their business in term of attractions as well as food and drink. Users able to add, delete and manage the attractions information which created by herself or himself. Then, the attraction added by users will expose to other users who are using this itinerary mobile application.

1.3.4 Attraction Map

If users intend to explore the attractions without create a trip, they are able to do so. The application provide a map with all the attractions available in the application. As the result, it provide a visual view to users regarding the attractions location as well. In addition, it also allow users to explore the attraction around users current location. Furthermore, users are able to filter the attractions appear in the map based on the category selected.

1.4 Impact, significance and contribution

With an itinerary planner mobile application, travelers are able to create an initial plan in the proper way. By doing so, users are able to carry out the planning efficiently because they do not have to spend more time figuring out what to do during planning. In addition, the system will connect to a real time database that holds all the necessary information in real time. So, users are able to use the advanced filter function in the application in order to search for more relevant information. It has become one of the reasons why travelers are able to build a plan quickly when an itinerary planner application is used. In addition, the mobile application also offer a fantastic function which is route optimization which help users to arrange all the selected activities in the plan smartly by creating a shortest path. As the result, it is able to prevent users arrange the activities wrongly and cause extra time is needed to travel from one location to another.

Nevertheless, itinerary mobile application is not only used by those who are preparing a tip, it is also provide a sharing channel to all the users to share their experience or happiest in the application. Furthermore, users able to expand their network by using the social platform in the mobile application because users are able to interact with each other. In addition, users are able to share their plan to social platform in order to allow other users have some idea how to plan the trip. By doing so, if any users are interested with the plan, they are able to import the plan into their account and further adjustment toward the imported plan is allow as well.

Last but not least, users are able to promote their business in term of attractions or food and beverage by creating the attraction in their account. As the result, the added attraction will expose to other users during the creation of trip. Moreover, users are able to manage the created attractions such as delete or modify the existing attraction(Only can manage those attraction created by the himself or herself).

1.5 Background information

The tourism industry is also known as the travel industry, in which people travel to other places, either locally or globally, for leisure or business purposes [5]. It is not a standalone industry but connects with other industries such as the hospitality industry, transport industry, hotel industry, and so on. Before the age of the internet, planning a trip was a **costly and time-consuming process**. This is because many of the processes have to be done manually before travel. For example, travelers are required to call the airline or travel agent to confirm the availability of the flight ticket and wait for them to reply. In addition, citizens seldom travelled by themselves in that century because there was **not much information available** to the citizens regarding the destination. In addition, the **information available is in book form instead of online**. Thus, travellers may have difficulties finding the resources they need. Moreover, there is another option at that moment, which is a travel agency. A travel agency will help the travelers arrange and plan everything. However, **travelers have no right to rearrange the schedule** based on their interests. Thus, the schedule provided by the travel agency may not fully suit all the travelers. Since the travel is handled by the agency, travelers will be charged an extra fee. In other words, the total cost of travel will be increased compared to planning the trip by ourselves.

As technology becomes more advanced, it actually **simplifies many processes** and makes them simple and efficient, especially in terms of itinerary planning. There are a few mobile applications on the market that help citizens plan their trips, such as Sygic Travel, KAYAK, and others.These applications will **guide the users step-by-step** in order to develop trips wisely and efficiently. Nonetheless, users can explore the destination using a single application rather than starting from scratch with a search engine.Thus, travelers able to avoid getting some wrong or unrelated information from the mobile application. Since the itinerary mobile application focuses on the attraction information, users have a high chance of exploring those hidden attractions compared to searching from scratch. As a result, it was able to help the fresh attraction grow the business.

Based on the domestic tourism survey 2020, although the total tourism expenditure dropped from 103.2 billion (2019) to 40.4 billion (2020) due to the COVID-19 pandemic, Perak will still remain the second most visited state by domestic visitors in 2020 [6]. This is due to the fact that Perak contains many natural resources and historical heritage, especially the capital of Perak, Ipoh, and the educational city, Kampar. With the travel application, more travellers will be exposed to the attractions and beauties of the country. As a result, more travelers will be attracted to visiting your country, which will boost the economy of the country. This is because tourism is able to boost the revenue of the economy, develop the infrastructure of the country, create more job opportunities, and so on, allowing the economy to be successful [6].

1.6 Report Organization

There are total 7 Chapters in the reports in order to explain the project in detail. In Chapter 1, it provide some of the introduction and the intention of the project such as problem statement, objectives, project scope and so on. In chapter 2, the comparison among several mobile applications which related to tourism is documented. In chapter 3, the system design in order to achieve the objectives is clearly explained by using some of the diagram such as use case diagram. In chapter 4, it discuss about how the prokect is developed in detail. For chapter 5 and 6, they mainly discuss about the prototype implementation as well as software testing plan and result. In the last chapter, it make a conclusion about the project as well as provide some of the future work of the project.

Chapter 2: Literature Review

There are several itinerary plan applications that have been launched in the market. However, those applications still contain some of the flows and limitations. In this chapter, we will discuss the strength and limitation or weakness of the existing application no matter web-based application or mobile application.

2.1 Mobile Application for Guiding Tourist Activities: Tourist Assistant – TAIS

Based on (Smirnov et al. 2014), the author has analyzed several applications which were related to travel. The author has classified the application into four categories which were "Location-Based Services", "Online Booking", "Information Resource" and "Trip Journal" [2]. Furthermore, there are another three categories which were the combination of the above categories which were travel guide, tour operator, hotel and hotel chain. The main purpose of this journal paper is to propose an application of travel which is related to travel guides. Travel guide is the combination of "Location-Based Services" and "Information resource".

Tourist Assistant – TAIS is a mobile application developed by using Java KPI library and using Android Java Development Kit on the client site. In order to simplify the development of the system as well as to make a dynamic and scalable system, TAIS adopts Smart-M3 platform to achieve it. Tourist Assistant – TAIS provides a range of services to citizens in order to offer a convenient channel for them to plan their trip well in an efficient way. The services offered include client application, attraction information service, recommendation service, region context service ride sharing service as well as public transport services [2].

In the main menu of TAIS, the system is able to locate the current location of users by showing the pointer on the dynamic map which allows users to enlarge or minimize it and the context data such as weather show on the screen also (Fig 2-1). In addition, some of the attractions will be displayed to users as well. If users intend to search the attraction based on the categories such as city or region, they are able to do so. If users were interested with the attraction in the filter list, they were able to enter the respective page to view the detailed information (Fig 2-2). The information is retrieved from internet resources such as Wikipedia. In addition, users are able to provide the feedback on the pictures whether like, dislike or the image is not applicable. By doing so, the system will reorder the images in order to prevent the negative emotion to the users.

The system adopts OpenStreetMap-based web mapping service which is able to calculate the attraction reaching path from the user's current location. It enables the tourists to search the public transports to reach the intended location by using Yandex.Schedule API. There are several functions to search the public transport to reach the destination. However, TAIS only applied two functions which are nearest station searching and searching route between stations. It will search the nearest station based on the user's current location as well as the destination and link them together. The result will show to the tourists in ascending order in terms of departure time.

Author also conducted several testing and evaluation to determine the performance of the system. Author found that TAIS only takes a few seconds to perform every operation, which most of the time is in searching. The following sections(2.1.1 and 2.1.2) will discuss about the strength and weakness of the application.

2.1.1 Strength

<u>Dynamization</u> – It allows users to resize the map either enlarge it or minimize it. Dynamization is able to improve the user's experience because point of view may be different from one another. Thus, the visualization of the map is able to be adjusted by the users depending on their view. In addition, the size of the devices which are used to launch the system will be different. So, dynamization can ensure the display looks great under any environment.

<u>Fast searching and processing algorithm</u> – TAIS applies an efficient algorithm which is able to process the data and display the result to users in a shorter time. Since the increasing number of tourists will linearly affect the query transaction time. Thus, testing was done by the author to evaluate the mobile application by generating triples that describe the tourist in smart space and the execution time required is calculated. It only took 0.3 second transaction response time in Smart-M3 platform for 1000 tourists (approximately 28 thousand of triples and one thousand of subscribe transactions (Smirnov et al. 2014). With the fast response time, it is able to avoid the challenge of the users' patients.

<u>Provide realistic view</u> - Users able to view the current location on the map instead of description. Furthermore, the path from one location to another will also show on the map when users perform public transport searching (Fig 2-3). So, users are able to have a solid view about the location and able to observe the actual distance between two or more locations marked on the map.

Re-order the image – TAIS allows users to provide the feedback toward the image. If users dislike the picture, the system will re-order the picture on this or next trip. It is able to prevent showing the same images to users again and bring the negative impression to users again.

Up-to-date information - The system will get the information from online such as Wikipedia. It is able to get the most updated information and no need to have a big database which is managed by the admin.

2.1.2 Weakness/Limitation

Feedback constraint – the system only allows users to provide feedback toward the image only. Tourists unable to provide the feedback in terms of other perspectives such as security, accuracy of information and so on. In addition, fresh users are unable to get some feedback from the floor before using the application.

Unreliable information – information from online resources may not be accurate and trustable such as Wikipedia. This is because information from Wikipedia can be edited by everyone (Hafner, 2006). In addition, there are no filtering methods to filter out the unreliable information before showing it to users. Thus, users may get the wrong information from the application.

Solid presentation – the presentation of information is solid with less decoration. For example, most of the divisions between the sections are using solid boxes which are unable to bring the comfortable view to the users. The application is used by tourists along their travels. Thus, interface design is important in order to improve their travel experience.

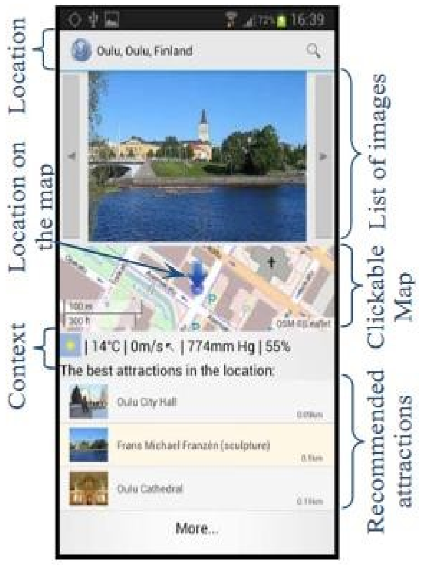
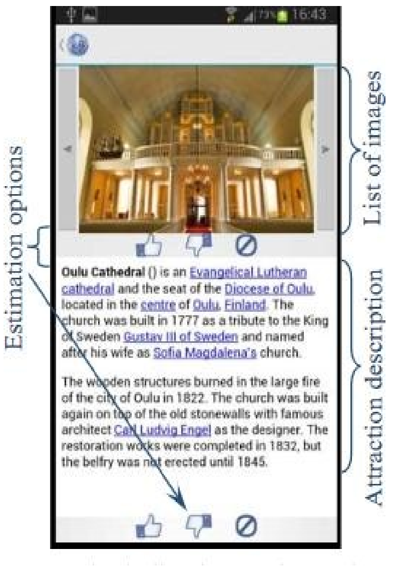


Fig2-1
Attraction suggestion (TAIZ)



Oulu Cathedral () is an Evangelical Lutheran cathedral and the seat of the Diocese of Oulu, located in the centre of Oulu, Finland. The church was built in 1777 as a tribute to the King of Sweden Gustav III of Sweden and named after his wife as Sofia Magdalena's church.

The wooden structures burned in the large fire of the city of Oulu in 1822. The church was built again on top of the old stonewalls with famous architect Carl Ludvig Engel as the designer. The restoration works were completed in 1832, but the belfry was not erected until 1845.

Fig 2-2
Description of attraction (TAIZ)

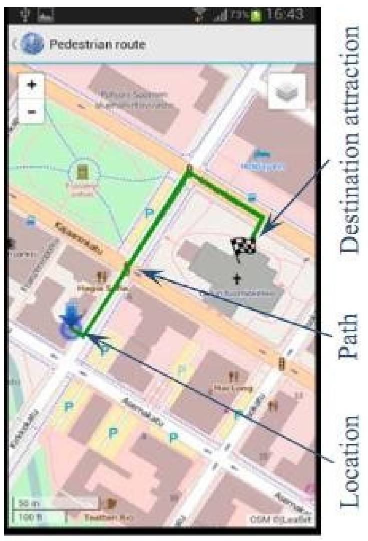


Fig 2-3
Path show on map (TAIZ)

2.2 Kayak.com

Kayak was a travel search engine which co-founded by Steve Hafner and Paul English in 2004 (Everything You Ever Wanted to Know About Kayak.com - DPO blog, 2021). Kayak has been recognized as for business travel and one of the best overall application for traveller. It has been awarded by several certification such as the World's Leading Fight Comparison Award in 2013 and so on. Based on the statistics, kayak application was available in more than 30 countries and adopt Alexa search engine which rank 571 in global and 166 in United States (Everything You Ever Wanted to Know About Kayak.com - DPO blog, 2021). Kayak has two version which were web-based system and mobile application. Both of the channels able to sign in by using email or link with Facebook account or Booking.com account.

Basically, Kayak offered three main functions to the tourists which were flight ticket booking, hotel booking and car rental booking. In the flight ticket booking sub-system, users are able to search the flight ticket by specifying the departure airport location, the destination's airport location, the date of flight and other detailed information (Fig 2-4). Then, the system will display the result and users are able to perform advanced filtering to filter out the unrelated result in terms of the price or the departure time, payment method and so on (Fig 2-5). In addition, users are able to purchase the flight ticket in one way trip, round trip or multiple trips. Other than searching for a particular flight, the system provides some suggestions about which country allows travel now. Furthermore, Kayak also develops for users to expose some travel in the world. Users are able to view the travel list by specifying departure. The travel list shows all the details of the trip such as the price of the flight ticket currently, estimated ticket price for a certain month, weather prediction and so on. System provides an overall view to users which display all the destinations as well as the price of a flight ticket for each destination on a single map (Fig 2-6). Users are able to filter the result by selecting the categories such as themes, distance and so on. The information of accommodation and the car rental was obtained from other well ranking sites such as Booking.com, Priceline and so on. It allows users to compare the price between several sites and choose one of the sites that users are comfortable with.

Users are able to add the selection into a collection during planning. If users intend to modify the decision, they are able to do so by modifying from the collection. In addition, users are able to open the alert function on the system. For example, users are able to set the alert function to alert the users when the price of certain accommodation is lower than current. In addition, a flight tracker is available on Kayak. It allows users to manage and check the information of the confirmed flight. Moreover, the system offered a special function which is measuring the size of the luggage. The purpose of this function is to help the tourists to estimate the price.

2.2.1 Strength

Advance search – Users able further filter the result by specific some of the characteristics of the result. If there is no advance search available on the system, users may not be able to find the intended result from the large database. Thus, advanced search enables users to search something as users expected in an effective and efficient way.

Notification – Notification able to notify the users whenever there are new announcements. On the Kayak system, there are two types of notifications which are push notification and the alert notification. Push notification occurs when there are new or special events such as special promotion. For alert notification, it will alert the users when the event set by the users occurs such as the price of the hotel is lower than certain value. Thus, users are able to get the newest update and without the need to keep checking it.

Travel suggestion – System provides some of the suggestions for users such as the countries that can travel now. It is a great feature in the travel guide system especially during COVID-19 pandemic. Thus, users are able to get some idea of where they can travel to.

Multiple currency – users able to change the currency according to their preference. The system will change the price based on the current rate by using currency converter API. Based on (The 10 Best Travel Planner Apps of 2019,2021), the author claim that global currency support is an essential and important feature in itinerary planner application.

Google map API – Kayak adopts google map API to show the location to users in the form of pointing on the map. Other than viewing a list of suggested travel locations in the list format, the system also shows the location as a pointer as well as the flight ticket price on the map. By doing so, users are able to have a simple view when they are planning a trip.

2.2.2 Weakness/Limitation

Wasting users' time – When users perform searching such as flight ticket searching by specific the departure and destination location as well as the date, the result show to the users may be empty when there are not any flights available on the particulars specification. It will waste users time when waiting the unexpected result. Thus, users may struggle in searching the available fright on certain period. If the system is able to show or prohibit users from selecting the unavailable flight on a certain date, it is able to show all the available flights to users clearly and easily.

Insensitive luggage scanning – Although the luggage scanning function is useful to predict the potential fees, but it is insensitive to detect the luggage. Thus, some of the users refuse to use this function.

Ineffective display lengthy information – Kayak displays the lengthy information all aligned in the left-hand side as shown on Fig 2-7. Users may refuse to read it when they are overwhelmed by the information.

Loss of control over updating the information – Since all the data of accommodation is obtained from other applications such as Booking.com. When the vendors remove the business from the market, the information will no longer update and require finding other vendors again.

<u>Missing data</u> – Based on "The 10 Best Travel Planner Apps of 2019", it said that some of the flights on specific routes may not show to the users. Thus, users have the probability of missing a great deal.

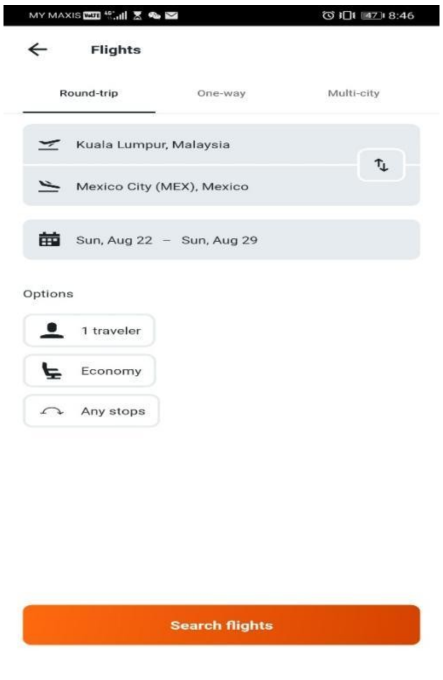


Fig 2-4
Flight ticket searching (Kayak)

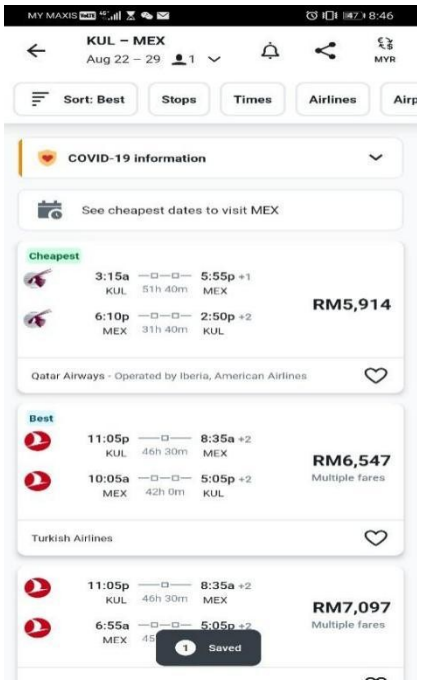

MY MAXIS 8:46
KUL – MEX
Aug 22 – 29 1
MYR
Sort: Best
Stops
Times
Airlines
COVID-19 information
See cheapest dates to visit MEX
Cheapest
3:15a KUL 51h 40m 5:55p +1 MEX
6:10p MEX 31h 40m 2:50p +2 KUL
RM5,914
Qatar Airways · Operated by Iberia, American Airlines
Best
11:05p KUL 46h 30m 8:35a +2 MEX
10:05a MEX 42h 0m 5:05p +2 KUL
RM6,547
Multiple fares
Turkish Airlines
11:05p KUL 46h 30m 8:35a +2 MEX
6:55a MEX 5:05p +2
RM7,097
Multiple fares
1 Saved

Fig 2-5
Result of searching with advance search (Kayak)

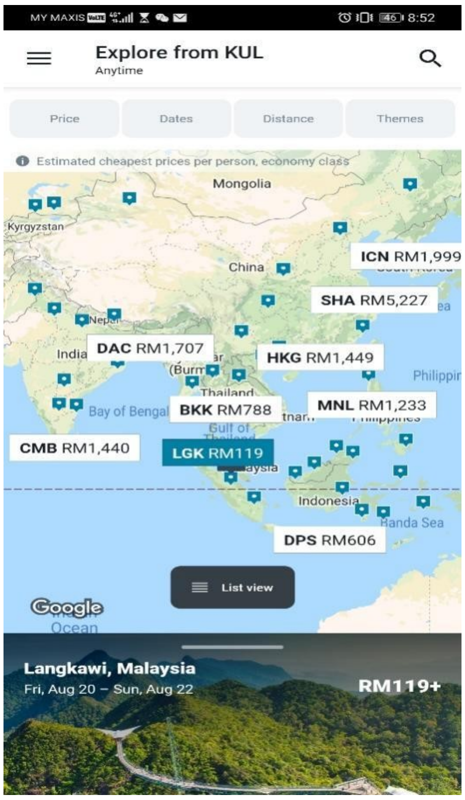
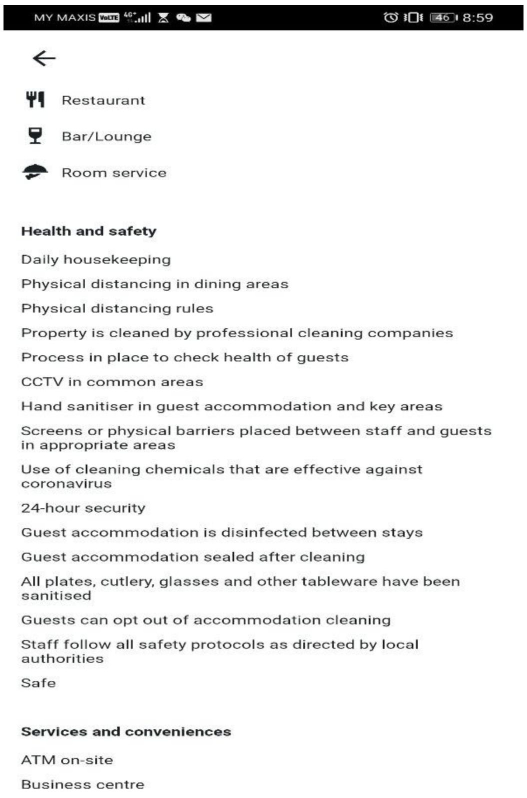


Fig 2-6 Map with all selection (Kayak)


MY MAXIS 8:59

Restaurant
Bar/Lounge
Room service

Health and safety
Daily housekeeping
Physical distancing in dining areas
Physical distancing rules
Property is cleaned by professional cleaning companies
Process in place to check health of guests
CCTV in common areas
Hand sanitiser in guest accommodation and key areas
Screens or physical barriers placed between staff and guests in appropriate areas
Use of cleaning chemicals that are effective against coronavirus
24-hour security
Guest accommodation is disinfected between stays
Guest accommodation sealed after cleaning
All plates, cutlery, glasses and other tableware have been sanitised
Guests can opt out of accommodation cleaning
Staff follow all safety protocols as directed by local authorities
Safe

Services and conveniences
ATM on-site
Business centre

Fig 2-7 Lengthy information (Kayak)

2.3 Sygic Travel Maps Offline & Trip Planner

Sygic Travel is an all-in-one trip planner application which allows users to discover tourist attractions and things to do anywhere users are as well as plan their trip in one application. In addition, it covered 50 million places around the world which included sights, museums, parks, cafes, hotels, and so on. It is available to both Android users and Apple users. In addition, users are able to browse the Sygic website through the web browser.

After users successfully enter into the system, there are three subsections available on the main page which are scheduled, past and trash. All the planned trips' details will show on the scheduled section. After users delete the scheduled trip, the particular detail will show on the trash section. In the past section, it will store the pass data which users plan before.

Furthermore, users are able to create a new trip by entering the specific country or city or even a town with the duration of the trip (Fig 2-8). Moreover, users are able to further explore the particular places on the application in two ways. The first method is by using searching and filtering. Users are able to search a place by entering the keyword such as Ipoh, the system will provide the recommendation and show all on the map by using a small icon (Fig 2-9). Users are able to filter the result based on some categories such as sport, sightseeing and so on (Fig 2-10). In addition, users are able to explore all the attractions available on the specified location in the list format instead of on the map (Fig 2-11). The second method is a sightseeing tour or activity available with charge. Virtual talk is available to users in order to give a brief introduction on a certain topic. Some talks will include the ticket fee for certain entry. Users are able to register the activity based on their interest. In addition, the system will be based on the location that users set during the trip creation to filter all the accommodation and car rental service (Fig 2-12). Users are able to book the interesting hotel or rent the car directly on the application. All the features above allow users to add into their itinerary planning as well as directly navigate users from the current location to the destination. Users are able to manage the itinerary plan by removing or adding the new event on the trip. Furthermore, estimated time moved from one direction to another will show on the plan as well (Fig 2-13). System will plot all the planned activities' location on the map and allow users to have a path view on the journey (Fig 2-14). There is a unique feature in this application which is the location map. System allows users to download the location map and refer it offline. Moreover, the system will provide the basic information about the location such as history, culture and so on. There are some other features only available on specific locations such as 360◦video which take a look around top sight in exclusive 360◦ video. In the following section (2.3.1 and 2.3.2), we will discuss the strength and the weakness of the application.

2.3.1 Strength

Advance trip planner – Sygic travel allows users to make an itinerary plan before the trip and be able to refer to the itinerary plan during the trip. By doing so, travellers are able to find all the correct and critical information from the application instead of searching from scratch. In addition, users plan the trip visually on the application rather than make some notes on other applications. Thus, it provides a better view and clear understanding about the trip to the travellers.

Powerful searching and filtering – The system provides a great search function to the users. Instead of searching from the large database, users are able to enter the address or the name of the location to find the intended location and are able to further filter the result.

Advance map function – The detailed map available on the application is based on the OpenStreetMap.org data adjusted for walking and allows users to explore the destination. In addition, the map enables GPS-based walking directions and is integrated with Sygic GPS Navigation. It further strengthens the functionalities of the map and provides a realistic and solid view to the users.

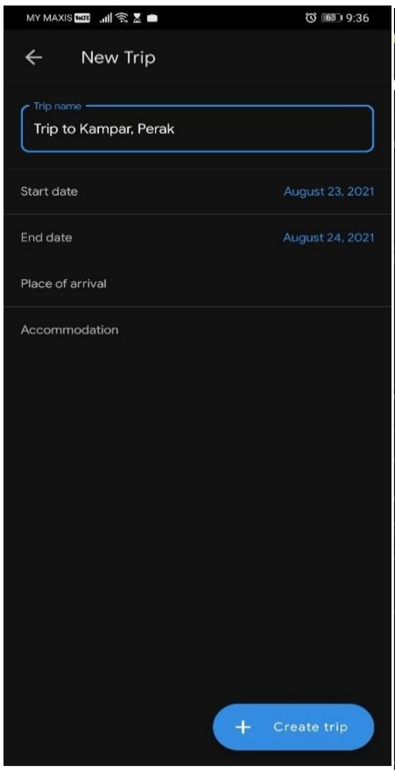
Offline functionality – Users able to use the application offline. In other words, users are able to refer to the itinerary plan during the trip even though there is no internet connection. However, users have to buy the premium version in order to grant this function.

Store pass data – When the itinerary plan is either completed or deleted, the plan will not disappear from the application. Users are still able to refer to the itinerary plane either on the pass section or trash section. Thus, users are able to restore the trip again and apply in the future.

2.3.2 Weakness / Limitation

Unreliable data – The information provided in the application such as description, opening hour, photos and so on are retrieved from either Wikipedia, or other databases. Since Wikipedia allows users to edit the information and there is not any authentication toward the correctness of the information. Thus, the information available on the application may not be correct.

Premium version with charge – users able to download the application free of charge. However, there are some functionalities that require users to buy the premium version such as applications available offline.


New Trip

Trip name: Trip to Kampar, Perak

Start date: August 23, 2021

End date: August 24, 2021

Place of arrival

Accommodation

Fig 2-8
Create a new trip (Sygic)

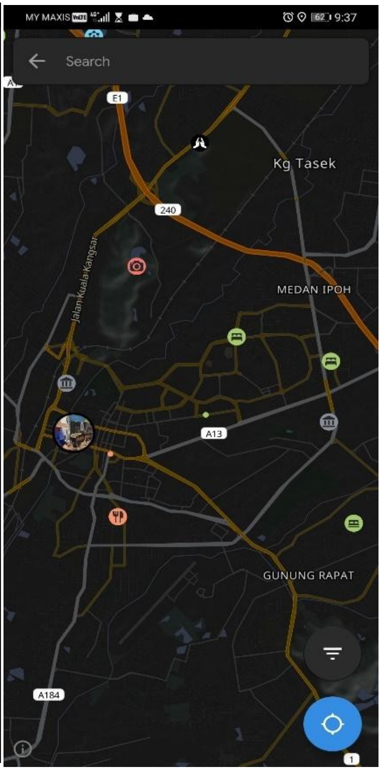


Fig 2-9
Attraction around destination (Sygic)

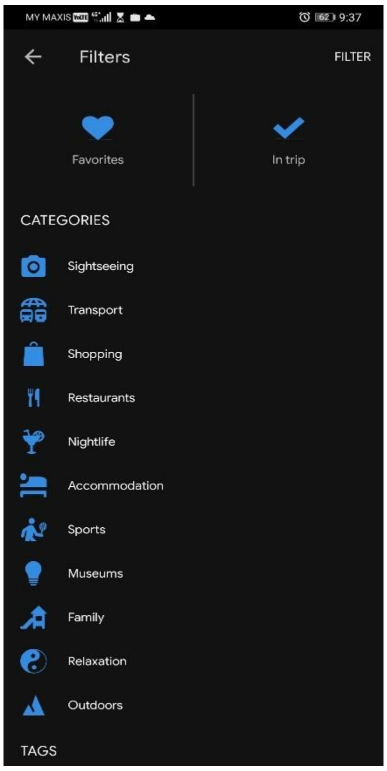
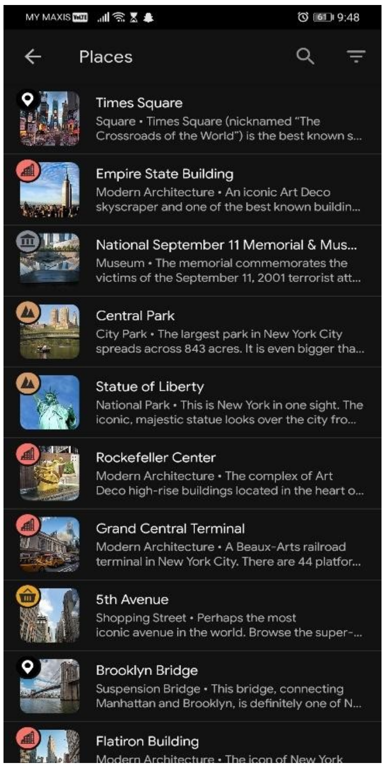


Fig 2-10
Further filter the result (Sygic)


Places

Times Square
Square • Times Square (nicknamed "The Crossroads of the World") is the best known s...

Empire State Building
Modern Architecture • An iconic Art Deco skyscraper and one of the best known buildin...

National September 11 Memorial & Mus...
Museum • The memorial commemorates the victims of the September 11, 2001 terrorist att...

Central Park
City Park • The largest park in New York City spreads across 843 acres. It is even bigger tha...

Statue of Liberty
National Park • This is New York in one sight. The iconic, majestic statue looks over the city fro...

Rockefeller Center
Modern Architecture • The complex of Art Deco high-rise buildings located in the heart o...

Grand Central Terminal
Modern Architecture • A Beaux-Arts railroad terminal in New York City. There are 44 platfor...

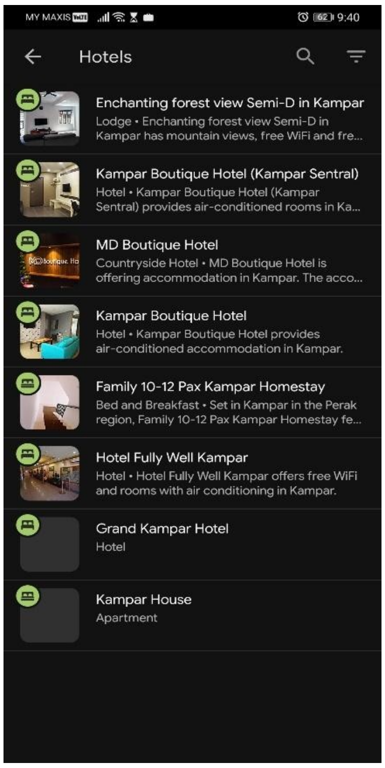
5th Avenue
Shopping Street • Perhaps the most iconic avenue in the world. Browse the super-...

Brooklyn Bridge
Suspension Bridge • This bridge, connecting Manhattan and Brooklyn, is definitely one of N...

Flatiron Building
Modern Architecture • The icon of New York

Fig 2-11

Attraction around destination (Sygic)


Hotels

Enchanting forest view Semi-D in Kampar
Lodge • Enchanting forest view Semi-D in Kampar has mountain views, free WiFi and fre...

Kampar Boutique Hotel (Kampar Sentral)
Hotel • Kampar Boutique Hotel (Kampar Sentral) provides air-conditioned rooms in Ka...

MD Boutique Hotel
Countryside Hotel • MD Boutique Hotel is offering accommodation in Kampar. The acco...

Kampar Boutique Hotel
Hotel • Kampar Boutique Hotel provides air-conditioned accommodation in Kampar.

Family 10-12 Pax Kampar Homestay
Bed and Breakfast • Set in Kampar in the Perak region, Family 10-12 Pax Kampar Homestay fe...

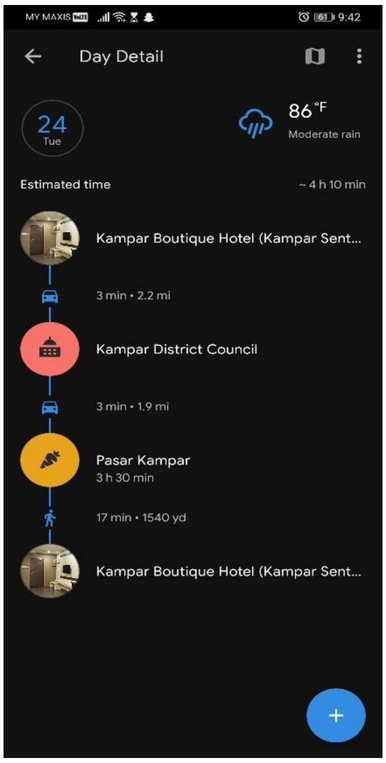
Hotel Fully Well Kampar
Hotel • Hotel Fully Well Kampar offers free WiFi and rooms with air conditioning in Kampar.

Grand Kampar Hotel
Hotel

Kampar House
Apartment

Fig 2-12

Hotel around the destination (Sygic)


Day Detail

24 Tue

86 °F Moderate rain

Estimated time ~ 4 h 10 min

Kampar Boutique Hotel (Kampar Sent...

3 min • 2.2 mi

Kampar District Council

3 min • 1.9 mi

Pasar Kampar
3 h 30 min

17 min • 1540 yd

Kampar Boutique Hotel (Kampar Sent...

Fig 2-13

Itinerary planner (Sygic)

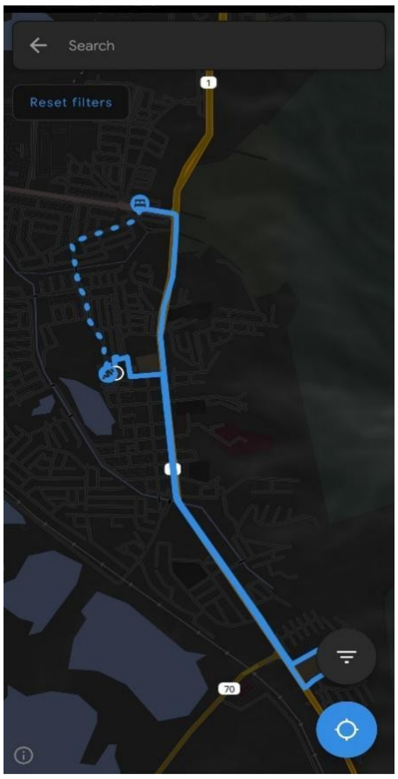


Fig 2-14

Itinerary plan show on map (Sygic)

2.4 Wanderlog – Travel Itinerary & Road Trip Planner

Wanderlog is an easiest-to-use and completely free travel application for planning any kind of trip such as group travel or road travel. It allows users to create an itinerary plan, view places to visit, organize flight and hotel reservations and collaborate with tripmates. In addition, it is available on Google Play Store and Apple Store as well as providing the web-based system.

Users are able to create an itinerary plan on Wanderlog by entering the destination and the duration of the trip (Fig 2-15). Wanderlog allows users to add the tripmates in the same itinerary plan and plan the trip together. Any changes made on one side will synchronise on the other side. In addition, users are able to forward the reservation details such as flight ticket, car rental service and hotel reservation to the specific email provided by the system in order to add the detail on the itinerary plan on the application. For hotel reservations, users are able to make the reservation directly in the application.
During the creation of a new trip, there are two main parts which let users plan their trip well which are the "Overview" section and "Itinerary" section. In the "Overview" section, it allows users to explore some of the famous tourist attractions around the destination. System will sort the attraction from the higher ranking to the lower ranking with a number label. Each of the labels will pin on the map. Furthermore, users are able to understand in depth about the interest attraction by clicking the specific attraction (Fig 2-16). System will show some basic information about the attraction such as opening hours, contact numbers, or a link that directs users to the attraction web page. In addition, users are able to add the interest attraction to the planning. All the attractions added to the planning will fall under "place to visit" in the "Overview" section. Moreover, users are able to add the new subsection under "Overview" such as restaurant, entertainment and so on. It allows users to group similar attractions together.

Under the "Itinerary" section, it allows travellers to plan the activity for each day. Travellers are able to arrange the interesting attractions which they choose before under "Overview" in the itinerary plan. If users intend to write some short note under the attraction such as things to do during the trip, users are able to add the notes under the intended attraction. After arranging all the activities in the itinerary plan, the system provides an optimization function which assists users to optimize the route among the location of the activities (Fig 2-17). In the following section (2.4.1 and 2.4.2), we will discuss the strength and weakness of the Wanderlog application.

2.4.1 Strength

Support group travel - users able to add all the tripmates in the same itinerary plan and plan the trip together. It allows users to plan their trip together virtually and provide a clear image to all the tripmates.

Import the reservation detail - Travellers may not make all the reservations in the itinerary planner application. Thus, Wanderlog allows users to import the detail in the itinerary plane by sending the detail to an email. As a result, it allows users to manage everything in this application.

Comparable price - Since Wanderlog allows users to make the hotel reservation in the application, the price offered in the application may vary from others due to promotion or other offer event. Thus, Wanderlog allows users to compare the price with other booking applications such as Kayak and Booking.com. It allows users to deal with the best deal at a reasonable price with the best service.

Route optimization – After users complete the itinerary plan, users are able to select the optimization function offered in the application. It will assist users to rearrange the sequence of the activities in order to provide a smooth sequence to visit all the attractions. Users may not have a clear understanding about the location of the attraction. Thus, back and forth to visit the attractions may occur during the trip. Route optimization is able to solve this problem and is able to optimize the transportation time.

Itinerary Plan available offline – traveller may not have internet access during travel. But they are still able to refer to the itinerary plan in the application even if there is no internet access. Thus, users are able to refer to the application any time with taking the internet into consideration. This feature is significant especially when traveling overseas, because not all locations offer free internet access.

2.4.2 Weakness / Limitation

Ineffective searching - Users only able to search the attraction by using the address of the location. If users intend to search the attraction by using the keyword such as restaurant or cafes, there are no results produced. It will limit users to explore more attractions in the destination.

Ineffective map information management – all the attractions will be associated with a number label in a certain color. System will plot all the labels on the map in order to provide a solid view to users. However, it may provide a messy view to users because users require refer to the attractions list in order to know what the label represents.

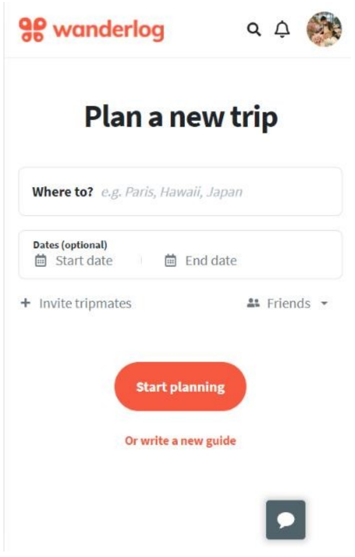
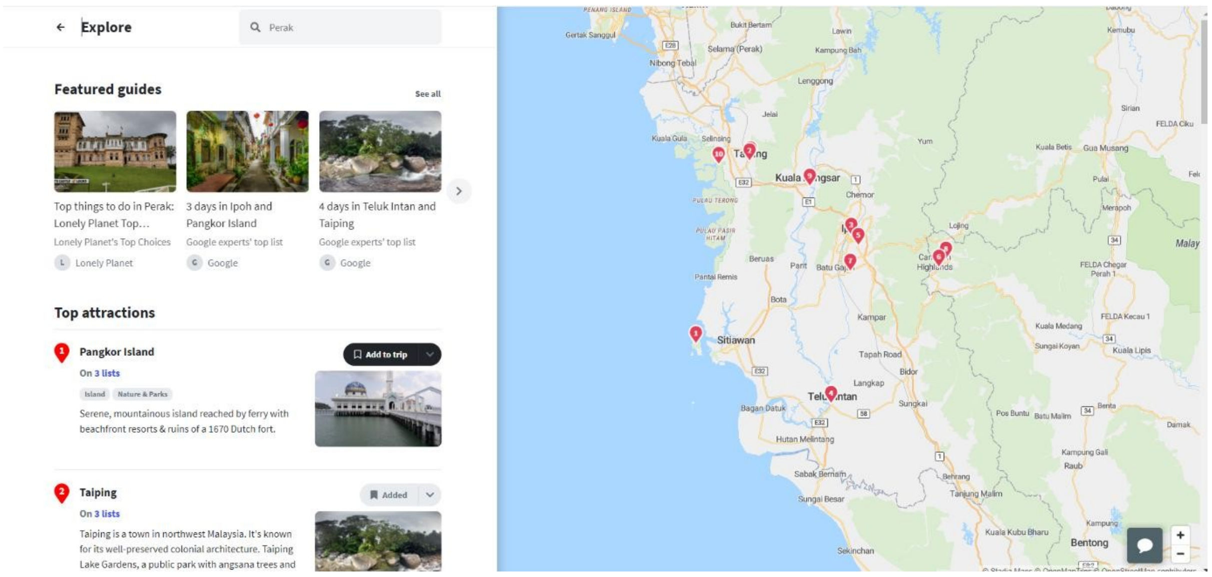


Fig 2-15 Create new itinerary plan (wenderlog)



1 Pangkor Island

On 3 lists

Serene, mountainous island reached by ferry with beachfront resorts & ruins of a 1670 Dutch fort.

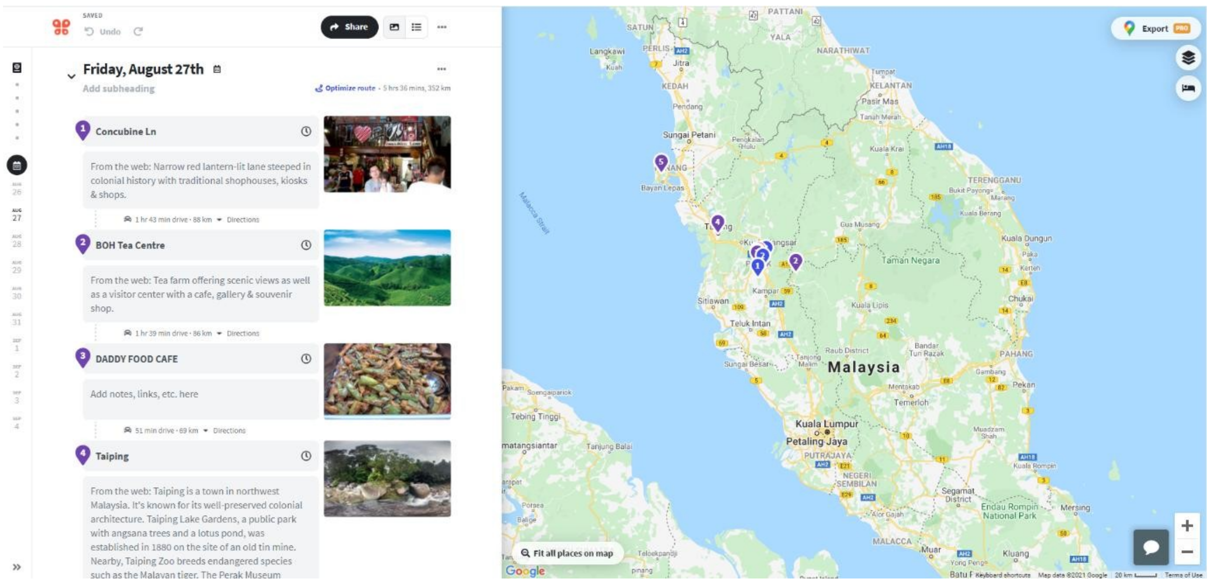
2 Taiping

On 3 lists

Taiping is a town in northwest Malaysia. It's known for its well-preserved colonial architecture. Taiping Lake Gardens, a public park with angsana trees and

Fig 2-16

Attraction exploration (Wanderlog)


SAVED
Undo
Share

Friday, August 27th
Add subheading
Optimize route · 5 hrs 36 mins, 352 km

1 Concubine Ln
From the web: Narrow red lantern-lit lane steeped in colonial history with traditional shophouses, kiosks & shops.
1 hr 43 min drive · 88 km · Directions

2 BOH Tea Centre
From the web: Tea farm offering scenic views as well as a visitor center with a cafe, gallery & souvenir shop.
1 hr 39 min drive · 86 km · Directions

3 DADDY FOOD CAFE
Add notes, links, etc. here
51 min drive · 69 km · Directions

4 Taiping
From the web: Taiping is a town in northwest Malaysia. It's known for its well-preserved colonial architecture. Taiping Lake Gardens, a public park with angsana trees and a lotus pond, was established in 1880 on the site of an old tin mine. Nearby, Taiping Zoo breeds endangered species such as the Malayan tiger. The Perak Museum

Fig 2-17

Itinerary plan and route optimization (Wanderlog)

2.5 NextTripPlan

NextTripPlan is an itinerary planner which allows users to plan the trip before the trip. It is only available on the Google Play store but not the Apple store.

It is similar to other itinerary planner applications which allows users to create new itinerary plans and plan the travel (Fig 2-18). On the main page of the itinerary plan, there is a timer to countdown the travel date (Fig 2-19). In addition, it allows users to organize a few things regarding the trip which are itinerary plan, packing list, to do list, bookings, transit, and documents during the creation of a new plan. Under each category, users are able to add a new item to it by entering the necessary information (Fig 2-20).

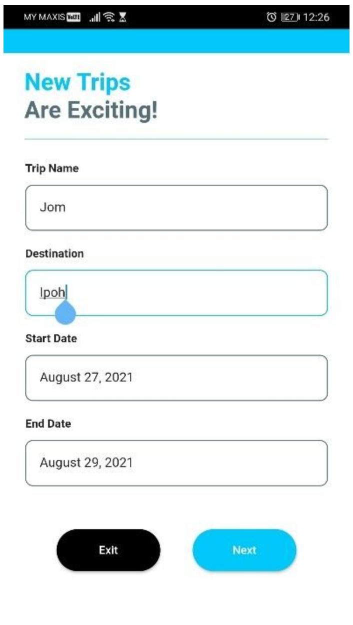
2.5.1 Strength

Take note everything about the trip – users are able to figure out anything regarding the plan that was planned before, they are able to refer to the data in the application. It is able to prevent users from losing the data about the planning.

2.5.2 Weakness / Limitation

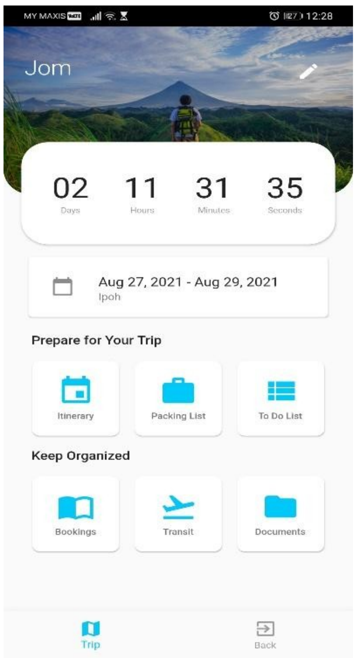
No recommendation provided – the system does not provide any recommendation to the users. Users are required to explore the destination by themselves from the internet and enter the necessary information into the system.

Excessive user input – users have to enter all the require information into the system step-by-step rather than input automatically by providing the selection. Users may fell annoying and troublesome when filling the detail.



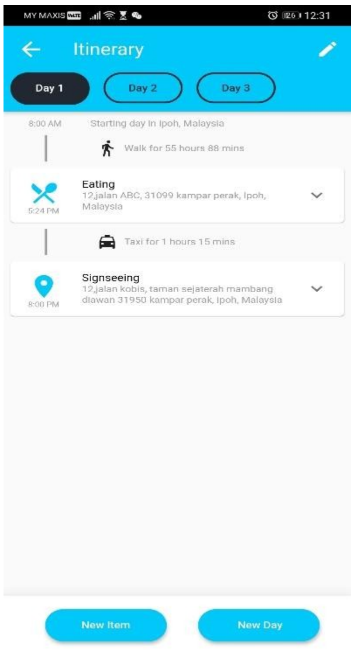
Jom
Ipoh
August 27, 2021
August 29, 2021

Fig 2-18
Itinerary plan creation
(NextTripPlan)



02 11 31 35
Aug 27, 2021 - Aug 29, 2021
Ipoh

Fig 2-19
Itinerary plan main page
(NextTripPlan)



8:00 AM Starting day in Ipoh, Malaysia
Walk for 55 hours 88 mins
Eating 12,jalan ABC, 31099 kampar perak, Ipoh, Malaysia 5:24 PM
Taxi for 1 hours 15 mins
Signseeing 12,jalan kobis, taman sejaterah mambang diawan 31950 kampar perak, Ipoh, Malaysia 8:00 PM

Fig 2-20
Manage itinerary plan
(NextTripPlan)

2.6 Itinerary App

Itinerary App is an itinerary planner mobile application which develop by the UTAR student who is Tong Guo Wei. In the home page of the application, all the existing trip will show in a slider manner(Fig 2-21). The application allow users to create a trip and mange it through the mobile application.

2.6.1 Strength

Recommendation system- The application will do the analysis first to figure out the type of current login user. So, the attraction recommendation system will based follow users' favorite. By doing so, users able to complete the planning faster because they able to explore the attraction they like.

Auto route generation- Users are able to use the auto route generation to arrange the attractions during the trip. By doing so, users are able to get the best route in the automatic way.

2.6.2 Weakness/Limitation

UI design - The user interface design in the application is very solid. This is because most of the thing are design in a box with shard edge. It may affect user the mood of the users when they are using it.

Data not updated - There are not any admin management at the back end. So, the information in the application may outdated from time to time. So, users may have the wrong information when they use it in future.

Hotel selection limitation- During the trip creation, users have to choose the hotel from the map. A list of hotel is predefined and store in the storage. There may have the possibility that the hotel book by the users is not available in the application. If this case occur, the creation is not success.

2.6 Summarization of Literature Review

Application	**Strength**	**Weakness or Limitation**
Tourist Assist – TAIS	- Dynamization - Fast searching and processing algorithms - Provide realistic view - Re-order the image - Up-to-date information	- Feedback constraint - Unreliable information - Solid presentation
Kayak	- Advance search - Notification - Travel suggestion - Multiple currency - Google API	- Wasting users' time - Insensitive luggage scanning - Ineffective display lengthy information - Loss of control over updating the data - Missing data
Sygic Travel Maps Offline & Trip	- Advance trip planner - Powerful searching and	- Unreliable data - Premium version with

Planner	filtering - Advance map function - Offline functionality - Store pass data	charge
Wanderlog – Travel Itinerary & Travel Itinerary & Road Trip Planner	- Support group travel - Import the reservation detail - Comparable price - Route optimization - Itinerary Plan available offline	- Ineffective searching - Ineffective map information management
NextTripPlan	- Take note of everything about the trip	- No recommendation provided - Excessive user input
Itinerary App	- Recommendation system - Auto route generation	- UI design - Data not updated - Hotel selection limitation

Table 2-1 Literature Review Summarization

Among all the travel applications above, every application has strengths and weaknesses in many aspects. However, features and functionalities have to be available in an itinerary planner application in order to meet the requirement of the users. The first criteria are that an itinerary planner application must integrate with the reservation service such as hotel reservation, flight ticket purchasing and car rental service. This is because users are able to perform all the actions in a single application and include the detailed information in the itinerary plan. Thus, they can avoid searching a trustable platform to perform reservations and import the details into the application. In addition, an itinerary planner application must provide plenty of up-to-date information to the users. This is because most of the users may not have a clear understanding about the destination. They require exploring the destination in an effective way in order to plan their trip well before travel. Thus, an itinerary planner application must make the accurate and up-to-date information available to the users. Furthermore, the information should be expressed in an effective way in order to let the users explore the destination easily. For example, we are able to express the information in table format which integrates with some animation and pictures to improve the user's experience instead of presenting the information in paragraphs. By doing so, users are able to make the decision faster during planning.

Moreover, advanced searching and sorting become a significant function when dealing with the large database. With the effective searching and sorting, users are able to search the intended result either by using the keyword or the category fileting. In addition, the system should provide some suggestions to the users. This is due to the reason that some of the hidden attractions may not be explored by the users during exploration. Then, the application is able to provide the ranking list which rank the attraction based on the location set by users. Thus, they are able to explore all the attractions from the ranking list. Last but not least, visualization is something ever better than the description. We should fully utilize the available functions such as Google maps to visualize the attraction instead of list all the attractions. In addition, we should visualize all the paths between the destinations of the activities on the map. Thus, users are able to observe the arrangement of the activities on the map instead of imaging it

Chapter 3: System Methodology/Approach

3.1 System Design Diagram/Equation

3.1.1 System Architecture Diagram

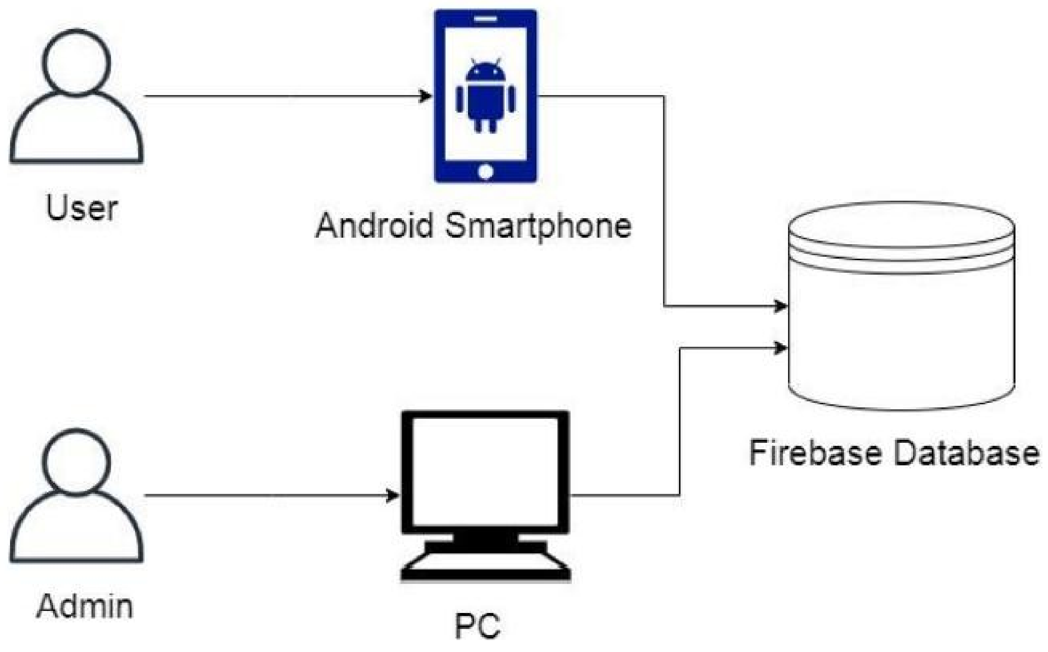
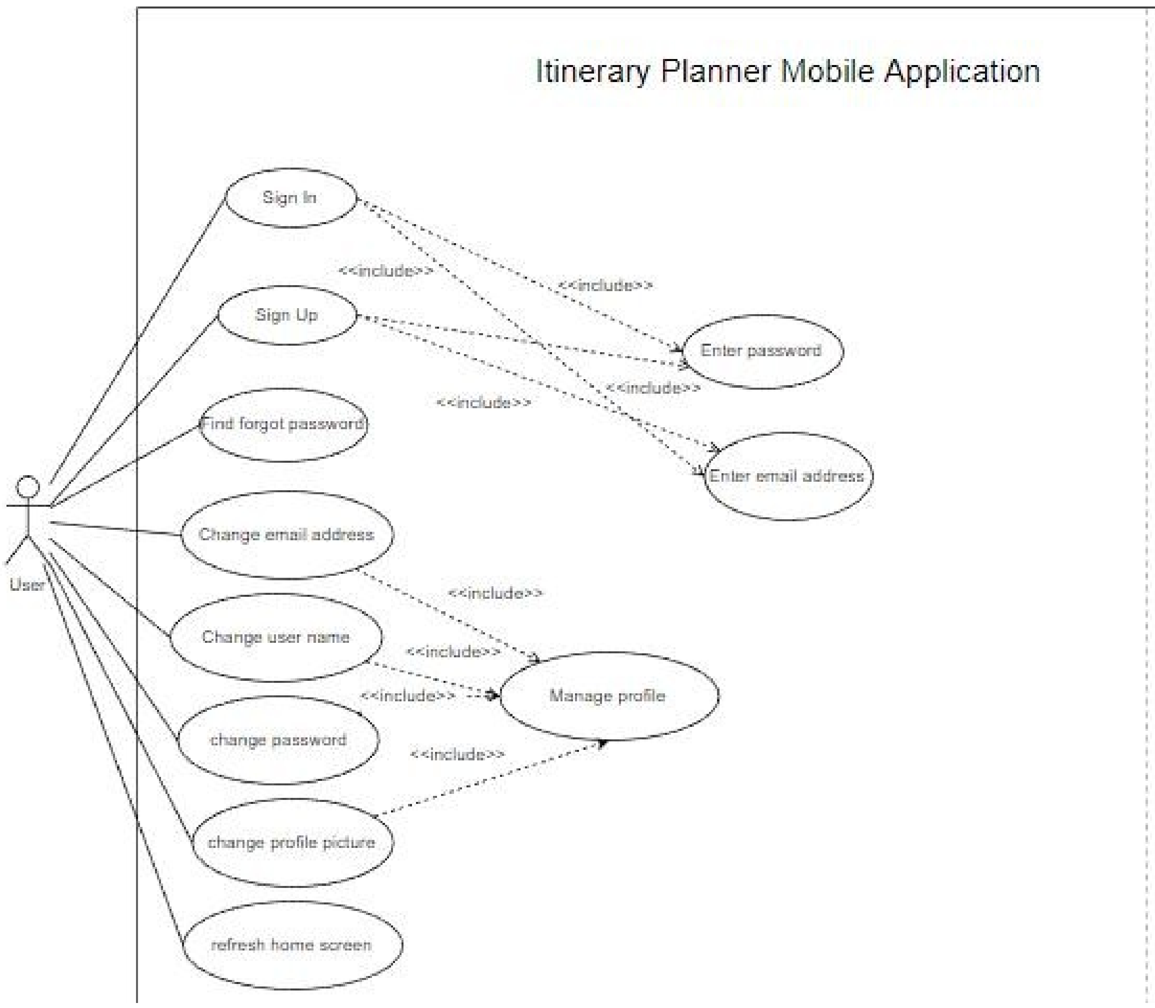
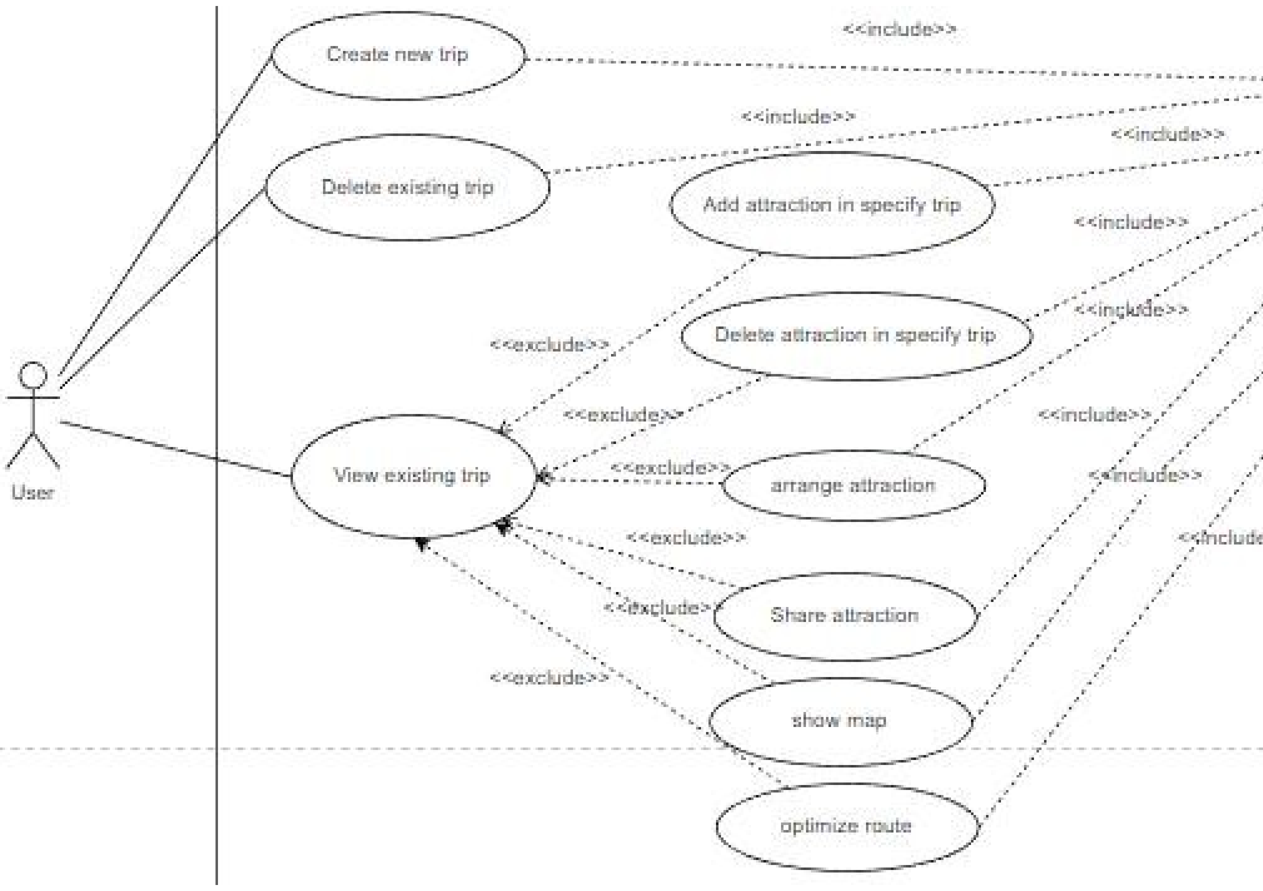


Fig 3-1 System Architecture Diagram

The architecture applied in the itinerary planner mobile application is a client-server architecture in which the server used is the Google Firebase server. First, the system will adopt the service from Google, which is the Google Firebase Authentication service, to manage the login and sign up staff. Based on Fig. 3.1, both the user and the admin will access the same database. Thus, all the data can be synchronized. When users log in to the application via the mobile app, all of the user's data, such as the attractions list and trip list, is first retrieved from Firebase. If users create new trip or make the modification of the existing trip, the information will upload and store on the Firebase instantly. On the other hand, the admin is able to add, modify, and delete the attractions from the Google Firebase console. All the changed data will be updated on Firebase Database. Thus, users are able to obtain the updated data after the admin has updated it.

3.1.2 Use Case Diagram




User
Create new trip
Delete existing trip
View existing trip
Add attraction in specify trip
Delete attraction in specify trip
arrange attraction
Share attraction
show map
optimize route
<<include>>
<<exclude>>

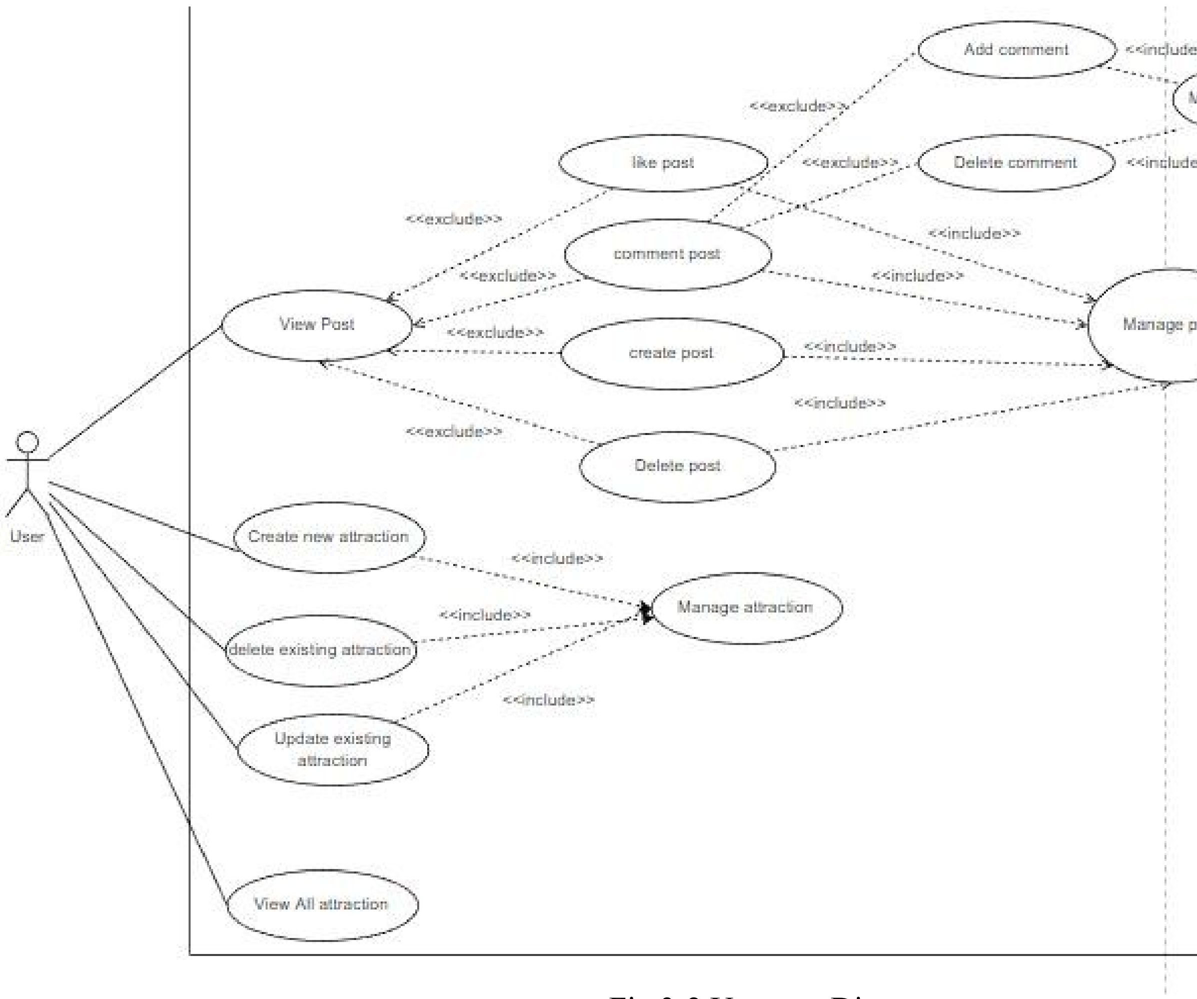


Fig 3-2 Use case Diagram

3.1.3 Use Case Description

Use Case ID	UseCase0001
Use Case Name	Sign Up
Brief Description	To allow users to sign up an account for itinerary planner mobile application
Actor	User
Trigger	When users click the "Sign Up" button in the login screen
Include Relationship	Enter email address, Enter password
Exclude Relationship	-
Normal Flow	1. Users click "Sign up" button in "login" screen. 2. The system navigates users to "sign up" screen. 3. User enter nick name. 4. Users enter email. 5. User enter password. 6. User select one of the gender check box. 7. Users click "Sign Up" button in "sign up" screen. 8. The system validates users input. 9. The system navigates users to "home" screen.
Sub-flow	-
Alternate/Exception flow	8a. The system will display "Invalid nick name!" error message when empty nick name is provided. 8b. The system will display "Invalid email!" error message when empty email is provided or the provided email does not contain @. 8c. The system will display "Password is too shot!" error message when the provided password is shorter than 6 letter. 8d. The system will display "Please select a gender!!!" error message when users do not check any of the gender check box.

Table 3-1 Sign Up Use Case

Use Case ID	UseCase0002
Use Case Name	Sign In
Brief Description	To allow users login into the itinerary planner mobile application.
Actor	User
Trigger	When users click the “Sign In” button in the “login” screen.
Include Relationship	Enter email address, Enter password
Exclude Relationship	-
Normal Flow	1. Users enter email address. 2. Users enter password. 3. Users click the “Sign In” button. 4. The system validates users input. 5. The system navigates users to “home” screen.
Sub-flow	-
Alternate/Exception flow	4a. The system will display “Invalid email!” error message when empty email is provided or the provided email does not contain @. 4b. The system will display “Password is too shot!” error message when the provided password is shorter than 6 letter. 4c. The system will pop out an alert dialog box to display “Could not find a user with that email” error message when the provided email is not register yet.

Table 3-2 Sign In Use Case

Use Case ID	UseCase0003
Use Case Name	Find forgot password
Brief Description	To allow user figure out the forgot password
Actor	User
Trigger	When user click “Forgot Passwor” button in the “login” Screen
Include Relationship	Enter email address
Exclude Relationship	-

Normal Flow	1. Users click "Forgot Password" button in the "login" screen 2. The system pop out a dialog box. 3. Users enter email address, which is the forgot password email. 4. The system validates users input. 5. The system pop out an message to inform users check the email. 6. The system triggers Google Firebase Authentication service. 7. Google Firebase Authentication service send an email to the provided email. 8. Users change the password from the email.
Sub-flow	-
Alternate/Exception flow	4a. The system will display "Invalid email!" error message when empty email is provided or the provided email does not contain @. 4b. The system will pop out an alert dialog box to display "Could not find a user with that email" error message when the provided email is not register yet.

Table 3-3 Find Forgot Password Use Case

Use Case ID	UseCase0004
Use Case Name	Change email address
Brief Description	To allow users change the email address of the existing account
Actor	User
Trigger	When users click the "Change E-mail" button in the "Setting" screen.
Include Relationship	Manage profile
Exclude Relationship	-
Normal Flow	1. Users click "Setting" button in the drawer of home

	screen. 2. The system will navigate users to "Setting" screen. 3. Users click "Change E-mail" button in the "Setting" screen. 4. The system pop out a dialog box. 5. Users enter new email address. 6. The system validates users input. 7. The system remove the dialog box. 8. The system display "Change Successfully" message on the bottom of the screen.
Sub-flow	-
Alternate/Exception flow	6a. The system will display "Invalid email!" error message when empty email is provided or the provided email does not contain @.

Table 3-4 Change Email Address Use Case

Use Case ID	UseCase0005
Use Case Name	Change User Name
Brief Description	To allow users change the user name of the existing account
Actor	User
Trigger	When users click the "Change User Name" button in the "Setting" screen.
Include Relationship	Manage profile
Exclude Relationship	-
Normal Flow	1. Users click "Setting" button in the drawer of home screen. 2. The system will navigate users to "Setting" screen. 3. Users click "Change User Name" button in the "Setting" screen. 4. The system pop out a dialog box. 5. Users enter new user name.

	6. The system validates users input. 7. The system remove the dialog box. 8. The system display “Change Successfully” message on the bottom of the screen.
Sub-flow	-
Alternate/Exception flow	6a. The system will display “Please enter a new user name” error message when empty user name is provided.

Table 3-5 Change User Name Use Case

Use Case ID	UseCase0006
Use Case Name	Change password
Brief Description	To allow users change the password of the existing account
Actor	User
Trigger	When users click the “Change password” button in the “Setting” screen.
Include Relationship	Manage profile
Exclude Relationship	-
Normal Flow	1. Users click “Setting” button in the drawer of home screen. 2. The system will navigate users to “Setting” screen. 3. Users click “Change Password” button in the “Setting” screen. 4. The system pop out a dialog box. 5. Users enter new password address. 6. Users enter confirm password address. 7. The system validates users input. 8. The system remove the dialog box. 9. The system display “Change Successfully” message on the bottom of the screen.
Sub-flow	-
Alternate/Exception flow	5a. The system will display “Password is too shot!” error message when the provided password is shorter than 6 letter.

	6a. The system will display “Password is too shot!” error message when the provided confirm password is shorter than 6 letter. 6b. The system will display “Password not match” error message when the provided confirm password is different from the new password.

Table 3-6 Change Password Use Case

Use Case ID	UseCase0007
Use Case Name	Change profile picture
Brief Description	To allow users change the profile picture of the existing account
Actor	User
Trigger	When users click the exiting profile picture in the “Setting” screen.
Include Relationship	Manage profile
Exclude Relationship	-
Normal Flow	1. Users click “Setting” button in the drawer of home screen. 2. The system will navigate users to “Setting” screen. 3. Users click profile picture in the “Setting” screen. 4. The system will navigate users to the file selection screen. 5. Users choose a picture as his/her profile picture. 6. The system uploads the picture to Google Firebase Storage. 7. Loading spinner is show while loading. 8. New profile picture is shown while upload is success. 9. The system display “Change Successfully” message on the bottom of the screen.
Sub-flow	-
Alternate/Exception flow	5a if users choose nothing, the system will not make any

	action.

Table 3-7 Change profile picture Use Case

Use Case ID	UseCase0008
Use Case Name	Refresh home screen
Brief Description	To allow users refresh the home screen in the itinerary planner mobile application.
Actor	User
Trigger	When user pull down the home screen.
Include Relationship	-
Exclude Relationship	-
Normal Flow	1. Users pull down the home screen 2. The system shows the loading spinner. 3. The system request data from the Google Firebase. 4. The system remove the loading spinner in home screen. 5. The system display the original UI of home screen.
Sub-flow	-
Alternate/Exception flow	-

Table 3-8 Refresh Home Screen Use Case

Use Case ID	UseCase0009
Use Case Name	Create new trip
Brief Description	To allow users create new trip in itinerary planner mobile application.
Actor	Users
Trigger	When users click the "Add" button in the bottom right corner of the home screen.
Include Relationship	Manage Trip
Exclude Relationship	-
Normal Flow	1. Users click "Add" button in the home screen. 2. The system navigate users to "Create New Trip" screen. 3. Users select an image from the phone's storage as trip

	profile picture. 4. The system show the selected image on top of the screen which retrieve from phone' storage 5. Users enter trip name of the trip. 6. Users select city of the trip. 7. Users select state of the trip. 8. Users choose the location of hotel/hostel 9. Users choose the data range of the trip. 10. Users click "Create" button in the "Create New Trip" screen. 11. The system validates users input. 12. The system shows the spinner in the button location while validating. 13. The system shows successfully created message on the bottom of the screen. 14. The system navigates users to "Attraction List Screen".
Sub-flow	-
Alternate/Exception flow	8a. User choose the hotel from the hotel list 1. The system retrieves the location of the selected hotel/hostel 2. . The systems updates the longitude and latitude 8b. Users enter the longitude and latitude 8c. Users click "Find My Location" button 1. The system check the permission of GPS enable setting 2. If the GPS is enable, the system retrieve the current location of user and updates the longitude and latitude 3. If GPS is not enable, the system request the permission from user, retrieve the current location of user and updates the longitude and latitude 11a. The system will display "Please provide a trip name!!!" error message when empty trip name is provided.

	11b. The system will display "Please provide a Hotel/Hostel Name!!!" error message when empty hotel name is provided. 11c. The system will display "Please provide a city name!!!" error message when empty city name is provided. 11d. The system will display "Please provide a state name!!!" error message when empty state name is provided. 11e. The system will display "Please enter image URL!!!" error message when empty image URL name is provided. 11f. The system will display "Please valid image URL!!!" error message when the provided image URL does not in the proper format. 11g. The system will display "Please choose the trip date!!!" error message when the date range is not set.

Table 3-9 Create New Trip Use Case

Use Case ID	UseCase0010
Use Case Name	Delete existing trip
Brief Description	To allow users delete the existing trip in itinerary planner mobile application.
Actor	User
Trigger	When users click the "delete" icon on the trip entry in the home screen.
Include Relationship	Manage trip
Exclude Relationship	-
Normal Flow	1. User click the "delete" icon on the trip entry. 2. The system pop out the dialog box to confirm the deletion process
Sub-flow	2a. Users click "Yes" button on the dialog box. 1. The system will delete the respective trip entry in the Google Firebase. 2. The system pop away the dialog box 3. The system show successful message at the bottom

	of the screen. 2b. Users click "No" button on the dialog box. 1. . The system pop away the dialog box.
Alternate/Exception flow	2a4-a. When users click the "undo" button on the right hand side of the message, the system will trigger the add function to add back the respective trip to Google Firebase.

Table 3-10 Delete Existing Trip Use Case

Use Case ID	UseCase0011
Use Case Name	Add Attraction in specify Trip
Brief Description	To allow users add the attraction to the respective trip.
Actor	User
Trigger	When user click the "Add" icon on the attraction entry in "Attraction List" Screen.
Include Relationship	Manage trip
Exclude Relationship	View existing trip
Normal Flow	1. Users click trip entry in the home screen. 2. The system navigate users to "Trip Detail" screen. 3. Users click the "Add" icon on the top right of "Trip Detail" screen. 4. The system navigates users to "Attraction List" screen. 5. Users click the "Add" button on the attraction in "Attraction List" screen. 6. The system pop out a dialog box with the day selection. 7. Users choose a day to add the attraction. 8. The system add the attraction into the day choose by the users in the respective trip. 9. The system display successfully message at the bottom of the screen.
Sub-flow	-
Alternate/Exception flow	7a. If the trip already contain the respective attraction, error message will show to the user .

	7b. If users add a new main activity into the trip and the trip is full of main activity which is 3, the system will show the error message. 7c. If users add a new F&B activity into the trip and the trip is full of F&B activity which is 3, the system will show the error message.

Table 3-11 Add Attraction in specify Trip Use Case

Use Case ID	UseCase0012
Use Case Name	Delete attraction in specify Trip
Brief Description	To allow users delete the attraction on the respective trip.
Actor	User
Trigger	When user click the “Delete” icon on the attraction entry in “Trip Detail” Screen.
Include Relationship	Manage trip
Exclude Relationship	View existing trip
Normal Flow	1. Users click trip entry in the home screen. 2. The system navigates users to “Trip Detail” screen. 3. Users click the “delete” icon on the attraction entry in “Trip Detail” screen. 4. The system remove the attraction from the respective day of the trip by triggering Google Firebase service. 5. The system display delete successfully message at the bottom of the screen.
Sub-flow	-
Alternate/Exception flow	5a. When users click the “undo” button on the right hand side of the message, the system will trigger the add function to add back the respective attraction into the original day of the trip to Google Firebase.

Table 3-12 Delete Attraction in specify Trip Use Case

Use Case ID	UseCase0013

Use Case Name	Arrange attraction
Brief Description	To allow users arrange the sequence of the attraction of the trip.
Actor	User
Trigger	When user long press a entry and drag to the position users intend to move.
Include Relationship	Manage trip
Exclude Relationship	View existing trip
Normal Flow	1. Users click trip entry in the home screen. 2. The system navigates users to “Trip Detail” screen. 3. Users long press a entry and drag to the position users intend to move. 4. The system make the changes on the list of the respective day of the trip.
Sub-flow	-
Alternate/Exception flow	-

Table 3-13 Arrange Attraction Use Case

Use Case ID	UseCase0014
Use Case Name	Share attraction
Brief Description	To allow users to move the attraction from one day to another on the same trip.
Actor	User
Trigger	When user click the “share” icon on the attraction entry in “Trip Detail” Screen.
Include Relationship	Manage trip
Exclude Relationship	View existing trip
Normal Flow	1. Users click trip entry in the home screen. 2. The system navigates users to “Trip Detail” screen. 3. Users click the “share” icon on the attraction entry in “Trip Detail” screen. 4. The system pop out a dialog box with the day selection.

	5. Users choose a day to add the attraction. 6. The system remove the attraction from the current day of the trip. 7. The system add the attraction into the day choose by the users in the respective trip. 8. The system display successfully message at the bottom of the screen.
Sub-flow	-
Alternate/Exception flow	12a. When users click the "undo" button on the right hand side of the message, the system will trigger the add function to add back the respective attraction into the original day of the trip as well as remove the attraction from the chosen day and update to the Google Firebase.

Table 3-14 Share Attraction Use Case

Use Case ID	UseCase0015
Use Case Name	Share map
Brief Description	To allow users to have the visual view on the selected attractions of the day
Actor	User
Trigger	When user click the "toggle map" icon in "Trip Detail" Screen.
Include Relationship	Manage trip
Exclude Relationship	View existing trip
Normal Flow	1. Users click trip entry in the home screen. 2. The system navigates users to "Trip Detail" screen. 3. Users click the "toggle map" icon on the attraction entry in "Trip Detail" screen. 4. The system get all the selected attraction of the day 5. The system create markers 6. The system shows the map with all the markers
Sub-flow	-

Alternate/Exception flow	-

Table 3-15 Show Map Use Case

Use Case ID	UseCase0016
Use Case Name	Optimize Route
Brief Description	To help users arrange the selected activities by just a click of a button.
Actor	User
Trigger	When user click the "smart routing" icon in "Trip Detail" Screen.
Include Relationship	Manage trip
Exclude Relationship	View existing trip
Normal Flow	1. Users click trip entry in the home screen. 2. The system navigates users to "Trip Detail" screen. 3. Users click the "smart routing" icon on the attraction entry in "Trip Detail" screen. 4. The system get all the selected attractions of the trip. 5. The system calculate the shortest path to connect all the attractions. 6. The system updates the trip schedule in Google Firebase. 7. The system show the updated schedule to user
Sub-flow	-
Alternate/Exception flow	-

Table 3-16 Optimize Route Use Case

Use Case ID	UseCase0017
Use Case Name	Like post
Brief Description	To allow users like the post in the "Social Platform" screen.
Actor	User
Trigger	When users click the "like" icon of the post in "Social Platform" screen.
Include Relationship	Manage post

Exclude Relationship	-
Normal Flow	1. Users click the "Social Platform" button in the drawer of home screen. 2. The system navigates users to the "Social Platform" screen. 3. Users click the "like" button of the post. 4. The system update the post like list.
Sub-flow	4a. If users do not like the post yet 1. The system adds users to the like list of the post. 2. The system change the "like" icon to blue. 4b. If users liked the post 1. The system removes users from the like list of the post. 2. The system change the "like" icon to grey color.
Alternate/Exception flow	-

Table 3-17 Like Post Use Case

Use Case ID	UseCase0018
Use Case Name	Comment post
Brief Description	To allow users comment the post in the "Social Platform" screen.
Actor	User
Trigger	When users click the "comment" icon of the post in "Social Platform" screen.
Include Relationship	Manage post
Exclude Relationship	Add comment, Delete comment
Normal Flow	1. Users click the "Social Platform" button in the drawer of home screen. 2. The system navigates users to the "Social Platform" screen. 3. Users click the "comment" button of the post. 4. The system navigates users to the "Comment" screen.

Sub-flow	-
Alternate/Exception flow	-

Table 3-18 Comment Post Use Case

Use Case ID	UseCase0019
Use Case Name	Add comment
Brief Description	To allow users add new comment in the “Comment” screen which belong to respective post.
Actor	User
Trigger	When users click the “add comment” bar at the bottom of the “comment” screen.
Include Relationship	Manage comment
Exclude Relationship	-
Normal Flow	1. Users click the “add comment” bar at the bottom of the “comment” screen. 2. The system shows the keyboard on the screen. 3. Users enter the comment. 4. The system validates the users input. 5. The system add the comment to the Google Firebase. 6. The system display the success message at the bottom of the “comment” screen.
Sub-flow	-
Alternate/Exception flow	4a. The system will display “Please enter a new comment” error message when empty comment is provided.

Table 3-19 Add comment Use Case

Use Case ID	UseCase0020
Use Case Name	Delete comment
Brief Description	To allow users delete comment in the “Comment” screen which belong to respective post.
Actor	User
Trigger	When users click the “delete” selection after longer pressing

	on the post entry in the "comment" screen.
Include Relationship	Manage comment
Exclude Relationship	-
Normal Flow	1. Users click the "delete" button on the most right of the comment. 2. The system remove the comment from the comment list of the post. 3. The system display deletion success message on the bottom of the "comment" screen.
Sub-flow	-
Alternate/Exception flow	2a. If the created comment user is not the user who currently login, the system will not show the "delete" icon to the users. 3a. When users click the "undo" button on the right hand side of the message, the system will trigger the add function to add back the respective comment to the post and update to the Google Firebase.

Table 3-20 Delete comment Use Case

Use Case ID	UseCase0021
Use Case Name	Create Post
Brief Description	To allow users create new post in the "Social Platform" screen.
Actor	User
Trigger	When users click the "add " icon on the bottom right corner of the "Social Platform" screen.
Include Relationship	Manage Post
Exclude Relationship	-
Normal Flow	1. Users click the "add" icon on the bottom right corner of the "Social Platform" screen. 2. The system navigates users to the "Create New Post" screen. 3. Users enter the caption of the new post.

	4. Users upload the picture of the new post. 5. Users upload the itinerary plan from his/her account by clicking the "import" button. 6. Users click the "create" button in the "Create New Post screen. 7. The system validates the users input. 8. The system add the new post to the Google Firebase. 9. The system display the success message at the bottom of the "comment" screen. 10. The system navigate users back to the "Social Platform" screen.
Sub-flow	-
Alternate/Exception flow	5a. The system shows a list of existing trip name. 5b, Users select the trip intend to upload. 5c. The system directs users to preview the selected trip. 1. If users select "import" button in the "preview" screen, the selected trip will be imported into the creating post. 2. If users select "cancel" button, the trip will not be imported in the creating post. 5d. The system will return users back to "Create New Post" screen. 7a. The system will display "Please provide the content of the post" error message when empty caption is provided, no photo uploaded, and not itinerary plan uploaded at the same time.

Table 3-21 Create Post Use Case

Use Case ID	UseCase0022
Use Case Name	Delete Post
Brief Description	To allow users delete post in the "Social Platform" screen.
Actor	User

Trigger	When users click the "delete" selection of the hidden menu on the post entry in the "Social Platform" screen.
Include Relationship	Manage Post
Exclude Relationship	-
Normal Flow	1. Users click the hidden menu(three dot at the top rigth corner of the post). 2. The system show the option in the hidden menu. 3. Users click the "delete" button. 4. The system pops out a dialog box for deletion confirmation.
Sub-flow	-
Alternate/Exception flow	2a. If the created post user is not the user who currently login, the system will not show the hidden menu to the users. 4a. Users click "Yes" button on the dialog box. 1. The system will delete the respective post entry in the Google Firebase. 2. The system pop away the dialog box 3. The system show successful message at the bottom of the screen. 4. When users click the "undo" button on the right hand side of the message, the system will trigger the add function to add back the respective post to the post list and update to the Google Firebase. 4b. Users click "No" button on the dialog box. 1. The system pop away the dialog box.

Table 3-22 Delete Post Use Case

Use Case ID	UseCase0023
Use Case Name	Create new attraction
Brief Description	To allow users create new attraction in the "Create New Attraction" screen.
Actor	User

Trigger	When users click the "add " icon on the top right corner of the "Manage Attraction" screen
Include Relationship	Manage attraction
Exclude Relationship	-
Normal Flow	1. Users select "Manage Attraction" screen in the drawer. 2. Users click the "add" icon on the top right corner of the "Manage Attraction" screen. 3. The system navigates users to the "Create New Attraction" screen. 4. Users select an image from phone's storage to become the representative image of the attraction. 5. User enter new attraction necessary information. 6. Users click the "create" button in the "Create New Attraction" screen. 7. The system validates the users input. 8. The system add the new attraction to the Google Firebase. 9. The system display success message at the bottom of the "Create New Attraction" screen. 10. The system navigate users back to the "Manage Attraction" screen.
Sub-flow	-
Alternate/Exception flow	7a. The system will display "Invalid Input" error message when empty field is provided.

Table 3-23 Create New Attraction Use Case

Use Case ID	UseCase0024
Use Case Name	Update existing attraction
Brief Description	To allow users update attraction in the "Update Attraction" screen.
Actor	User
Trigger	When users click the attraction entry in the "Manage

	Attraction" screen.
Include Relationship	Manage attraction
Exclude Relationship	-
Normal Flow	1. Users select "Manage Attraction" in the drawer 2. Users click the "edit" icon of the attraction entry in the "Manage Attraction" screen. 3. The system fetch the data of the attraction. 4. The system navigates users to the "Update Attraction" screen. 5. Users update the attraction. 6. Users click the "update" button in the "Update Attraction" screen. 7. The system validates the admins input. 8. The system update the attraction to the Google Firebase. 9. The system display success message at the bottom of the "Update Attraction" screen. 10. The system navigate users back to the "Manage Attraction" screen.
Sub-flow	-
Alternate/Exception flow	6a. The system will display "Invalid Input" error message when empty field is provided.

Table 3-24 Update Existing Attraction Use Case

Use Case ID	UseCase0025
Use Case Name	Delete Existing Attraction
Brief Description	To allow user delete attraction in the "Manage Attraction" screen.
Actor	User
Trigger	When users click the "delete" icon of the attraction entry in the "Manage Attraction" screen.
Include Relationship	Manage attraction
Exclude Relationship	-

Normal Flow	1. Users select "Manage Attraction" in the drawer. 2. Users click the "delete" icon of the attraction entry. 3. The system pop out the delete selection. 4. Users click the "delete" button. 5. The system will show the dialog box for delete confirmation.
Sub-flow	-
Alternate/Exception flow	5a. Users click "Yes" button on the dialog box. 1. The system will delete the respective attraction entry in the Google Firebase. 2. The system pop away the dialog box 3. The system show successful message at the bottom of the screen. 4. When users click the "undo" button on the right hand side of the message, the system will trigger the add function to add back the respective attraction to the attraction list and update to the Google Firebase. 5b. Users click "No" button on the dialog box. 1. The system pop away the dialog box.

Table 3-25 Delete Existing Attraction Use Case

Use Case ID	UseCase0026
Use Case Name	View All Attraction
Brief Description	To allow user view all the existing attraction in the application.
Actor	User
Trigger	When users select "Attraction Map" in the drawer
Include Relationship	-
Exclude Relationship	-
Normal Flow	1. Users select "Attraction Map" in the drawer.

	2. The system will retrieve all the existing attractions data from Google Firebase. 3. The system creates markers of all the attraction. 4. The system shows map to users with all the markers. 5. User click the marker. 6. The system will direct users to the respective “Attraction detail” screen.
Sub-flow	-
Alternate/Exception flow	-

Table 3-26 View All Attraction Use Case

Chapter 4: System Design

4.1 System Block Diagram

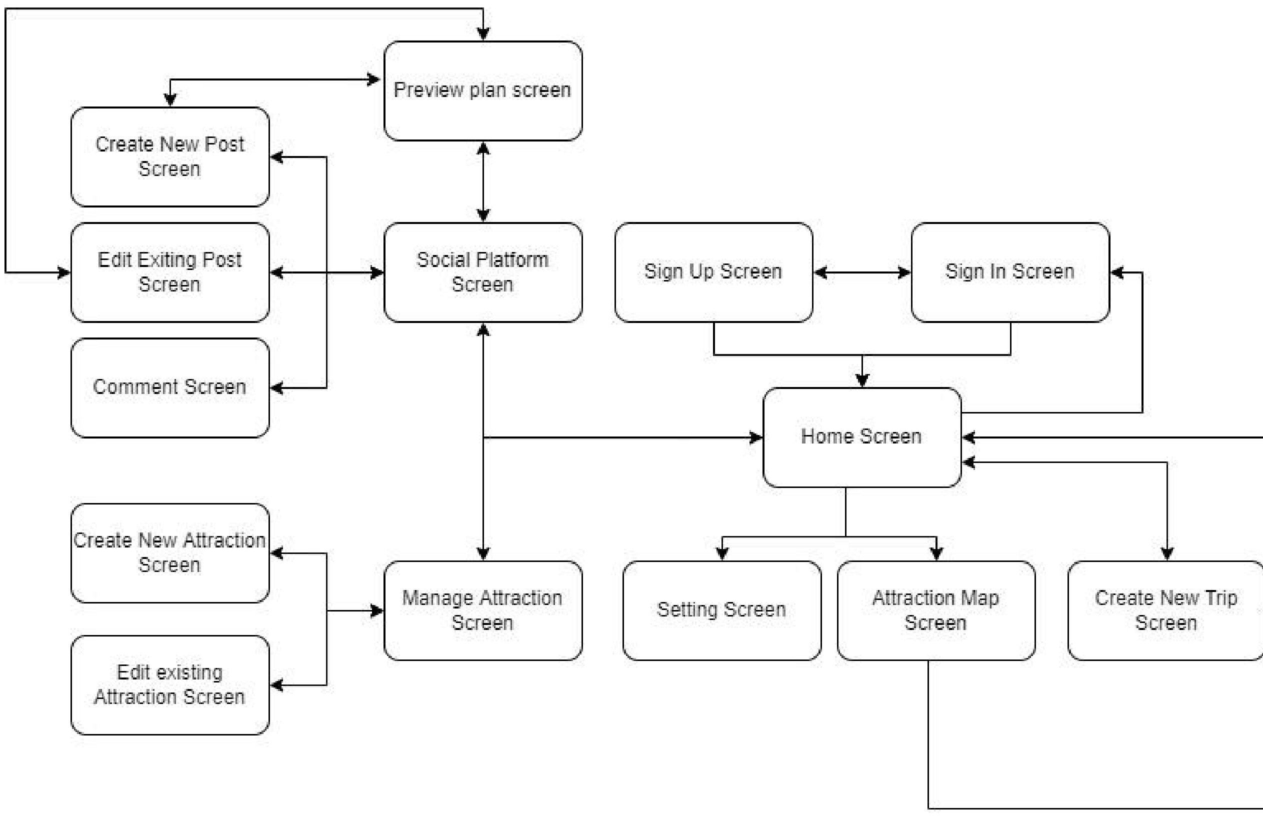


Fig 4-1 System Block Diagram

When users launch the mobile application, a sign-in screen will appear for users. To access the itinerary planner mobile application, users must enter the correct email address and password of an existing account. If users do not have an account, they are able to create a new account by clicking the "sign up" button on the sign in screen. If users successfully login or create a new account, the system will navigate them to the home screen.

On the home screen, there are six main functions, which are setting management, social platform, attraction map, attraction management, existing trip management, and creating a new trip function. When users click the "setting" button on the drawer of the home screen, the system will navigate them to the settings screen. Users are able to manage their profiles, such as changing their email address, changing their password, or changing their user name. In addition, they are also able to view the terms and conditions as well as the policy of the application.

Furthermore, if users intend to create a new trip, they can click the "add" button on the bottom right corner of the home screen. In order to successfully create a trip, users have to input all the valid data in the "Create New Trip" screen. If the creation is successful, the system will navigate users to the "Attraction List" screen. On the "Attraction List" screen, users are able to filter the attractions by choosing a filter option such as "family", "relaxation", "sightseeing", and so on. If users are interested in a particular attraction, they are able to click on the respective entry and the system will bring them to the attraction detail screen. In order to add the attraction to the trip, users are able to click the "Add" button either in the respective entry on the "Attraction List" screen or the "Add" button on the "Attraction Detail" screen. Users have to choose which day they intend to add to. It is forbidden to visit the same attractions twice during the trip. In the top right corner of the "Attraction List" screen, there is a "show details" button. As users click it, the system will navigate them to the trip detail screen. The "Trip Detail" screen will show all the information about the respective trip, such as the attraction arrangement of each day, either in list view or map view. A user is able to move the sequence of the attraction arrangement by dragging and dropping. Furthermore, they are able to move the respective attraction from one day to another by clicking the share button and choosing which day the attraction will move to. Moreover, users are able to delete the respective attraction by clicking the "Delete" button on the entry. If users intend to know more about the specified attraction, the system will navigate users to the respective "Attraction Detail" screen as soon as they click on it. In addition, the system also offers a smart routing function that helps users arrange the attractions of their trip by just clicking a button. Nevertheless, the trip detail screen and attraction list screen are changeable by clicking the button on the top right corner. Moreover, all the created trips will show up on the home screen in the slider format display, which is categorised by "on going trip" and "history plan." The user is still able to manage them by clicking on them, and the system will navigate them to the respective "trip detail" screen.

If the users intend to explore all the attractions available in the application, they are able to select the "Attraction Map" button on the drawer of the main page. After that, the system will direct users to the "Attraction Map" screen, on which all the attractions are represented by a marker on the map. Users are also able to filter out the attractions shown on the map by selecting the categories on the filter option provided. Moreover, all the markers on the map are clickable. After clicking it, the system will direct users to the respective "Attraction Detail" screen.

A social platform is available on the drawer of the main page. Users are able to view all the available posts that were created by others. In addition, users are able to add comments and likes on the respective posts. Furthermore, users are able to create a new post by clicking the "add" button on the social platform screen and filling in the details required. If users intend to modify the content of an existing post that was created by him/her, they are able to do so. The social platform allows users to share their plans with others by attaching them to them. If someone is interested in the plan after previewing it, they are able to download the plan into their account to carry out further management.

Last but not least, users are able to promote their attraction by adding it to the mobile application. Users are able to click the "Manage Attraction" button on the drawer in order to access the "Manage Attraction" screen. Furthermore, users are also allowed to perform further management on the existing attraction in order to provide up-to-date information to other users.

4.2 System Design Principle

During the design of the user interface, several rules are applied in order to meet the user's requirements and be able to design the interface that favours the end user. The following will discuss all the laws applied.

4.2.1 KISS Rule

KISS Rule stand for "Keep it Simple, Stupid". The design principle tell us that end users will not concern about how clever the developers and designers are and how good the skill they have, the one and only one matter concern by users are how the output of the application able to provide something is useful to the users in the effective and efficient way[9]. As we based on this rule during design, the main objective is to make all the task simple and clear as well as allow users to complete the task in a shortest time and obtain the output they intended to.

By following the rules, we keep everything simple and clear in our design. We provide some of the smart ways to reduce the work of users and enable them to obtain the intended result at the same time, which is the smart routing system. In addition, we also use the drawer to hide the selection on the main page instead of displaying it all in one shot. A simple background is applied in order to provide a clear screen to all the users.

4.2.2 3 Click Rule

In order to allow users to complete the task in a faster way, the fewer the number of clicks, the better it is. In other words, it is better to keep the number of clicks lower than or equal to three in order to make the thing simple and easy.

In the itinerary planner application, most of the tasks are able to be completed within three clicks. We take the new trip creation as an example. Users have to click the "add" button on the main page, enter all the necessary information, and then click the "create" button to do it.

4.2.3 Fitts' Law

Fitts' Law states that the amount of time required for end users to interact directly impacts the size of the target [10]. In other words, the size of the icon used in the application must be large enough in order to allow users to click easily. In addition, the principle also tells us that the space between the icons should be enough to prevent users from missing the icon.

In the application of our project, the small size of the icon we use is 24.0 X 24.0, which is the default setting of the icon in flutter. In addition, there is 10 pixel padding around the icon to prevent miss-clicking problems.

4.2.4 Law of Common Region

The Law of Common Region states that the elements on a page that are placed close to each other will be perceived as a group and that they are interconnected with each other [11]. For example, if there is a group of buttons at the bottom of the main page, users will immediately think that these are the functions of the tab.

It is applied in our project, such as the drawer contains a list of the module buttons which allow users to go to the respective module.

4.2.5 Jakob's Law

During the design phase of the project, we should design the user interface to follow universal practise instead of the opposite [12]. For example, a trash icon in red will tell the users that it is a function that will delete something, and the colour red is a warning to users to be careful with it.

In our project, we also follow all the universal manners, such as fork and spoon representing a restaurant, the hamburger icon representing a hidden menu in it, and it should be placed on the top of the screen as a universal manner, and so on. Instead of requiring users to learn in your application, this allows it to make the process more efficient.

Chapter 5: System Implementation

5.1 Hardware Setup

1. Laptop

Operating System	Window 10 64-bit
Processor	AMD Ryzen 7 350H Radeon Vega Mobile Gfx 2300Mhz
RAM	8GB DDR memory

Table 5-1 Laptop criterion

2. Smartphone
   - We require a smartphone in order to run and test the system prototype

Operating System	Android 10
Processor	Huawei Kirin 980
RAM	8GB

Table 5-2 Smartphone criterion

5.2 Software Setup

1. Visual Studio Code

Visual Studio Code is a lightweight yet capable source code editor for Windows, macOS, and Linux that runs on your desktop. It contains built-in support for JavaScript, TypeScript, and Node.js, as well as a large ecosystem of extensions for additional languages and runtimes.

2. Google Firebase

It is a noSQL database which is free of charge available on Google. The reason why Google Firebase is chosen is because it provides several built- in functions such as authentication and real-time database.

3. Google Cloud technology

Since our application requires direct users to the destination and visualize the location on the map, Google Direction API and Google Map will be used on the project in order to achieve the objective.

4. Visual Paradigm Online

It is a useful drawing tool which is free of charge. All the diagrams such as UML diagrams are drawn by using this online tool.

5.3 Programming Language

1. Dart
   It is an object-oriented programming language which allows the programme to structure the coding in terms of classes. In addition, it extensively supports code reuse. For example, we are able to adopt the inheritance concept to structure our code well. However, it only support single inheritance.

5.4 System Operation

5.4.1 Sign In Screen

When users launch the mobile application, the system will navigate users to the login screen (UI refer to A-1). The account management process is handled by the Google Firebase Authentication service. Users have to enter a valid email address and a corresponding password in order to access the mobile application. As users click the "Sign In" button, the system will validate their input (code refer to B-1). If the email field entry is empty or the length of the password field is less than 6, an error message will be shown to users (UI refer to A-2). In addition, Google Firebase Authentication service also validates the email and password (code refer to B-2). If the error exists, the system will pop out an alert dialog box in order to warn the users (UI refer to A-3). If all the validations pass, users are eligible to login into the itinerary planner mobile application (UI refer to A-4, A-5).

5.4.2 Sign Up Screen

If users click the "Sign Up" button on the "Sign In" screen, the system will navigate users to the "Sign Up" screen (UI refer to A-6). All the input is handled in a form where all the entries are checked by the system when users click the "Sign Up" button on the "Sign Up" screen (code refer to B-3). The system will show the error message in the respective field when the input data is invalid (UI refer to A-7). In addition, Google Firebase Authentication also checks the register data before users can create an account successfully. If an error occurs, a alert dialog box will appear to alert users (UI refer to A-8).If all the validations are passed, the system will create a new account by adopting the Google Firebase Authentication service and navigate users to the home screen (UI refer to A-4, A-5).

5.4.3 Find Forgot Password

If users forgot the password of the register account, users able to press the “Forgot Password” button on the “Sign In” screen. Then, system will pop out the dialog box for users to enter the email address (UI refer to A-9). The data entry will check by the system when users press the “Verify” button in the dialog box (code refer to B-4). In addition, the Google Firebase Authentication also check whether the email provided is registered. If not, alert dialog box with error message will display (code refer to B-5, UI refer to A-10). If all the validations are pass, the system will show the success message and inform users check their email address to change the password (UI refer to A-11).

5.4.4 Home Screen

On the home screen, the greeting section will show on top of the screen. The background of the greeting section and the greeting will be based on the current time. There are three types of greeting, which are morning session, afternoon session, and evening session. In addition, there is the name shown in the greeting section as well, which is based on the current login user.

Furthermore, the system also displays all the exciting trips on the home screen into two groups, which are "On-going Trip" and "History Trip." For those trips who start date is after the current date, they will fall into the "On-going Trip" group. For those trips whose end date is before the current date, they will fall into the "History Trip" group.

Sometimes the internet connection may not be good. So, users are able to refresh the home screen by pulling the screen down. During refreshing, a loading spinner will appear on the screen.

5.4.5 Setting Screen

Users are able to manage their account by clicking the "setting" button in the home screen drawer (UI refer to A-12). Then, the system will navigate users to the "Setting" screen (UI refer A-13). In the "Settings" screen, users' information such as user name and email address as well as the profile picture will show on top of the screen. There are 5 entries in the "Settings" screen, which are "Change User Name", "Change Password", "Change E-mail", Privacy, and Terms & Conditions. As users click the first three entries, which change their personal details, the system will pop out the dialog box with a form for users to enter the new data (UI refer A-14, A-15, A-16, Code refer B-6, B-7, B-8). All the users' input will be validated by the system. If there is an error, an error message will be shown to the users. If there are no errors, a success message will appear at the bottom of the "Setting" screen.

Furthermore, users able to logout the system by clicking either the “logout” button in the “Setting” screen or in the drawer. Then, users will be navigated back to the “Sign In” screen.

5.4.6 Create New Trip Screen

Users are able to create a new trip by clicking the "Add" icon on the bottom right corner of the home screen. Then, the system will navigate users to the "Create New Trip" screen (UI refer to A-17). In the "Add New Trip" screen, there is a form which allows users to enter the necessary data (code refer B-9). In addition, users are able to select an image from the phone storage to become the trip profile image. After an image is selected, the image will show on top of the form(UI refer to A-18). In addition, each entry in the form will be validated by the system. If an error occurs, an error message will be shown to the users (UI refer to A-19). If no errors are found, the system will generate a new trip and upload the data to Google Firebase (code refer to B-10). In addition, a success message will display at the bottom of the screen, as creation is success. Then, the system will navigates users to the “Attraction List” screen (UI refer A-22)

5.4.7 Trip Detail Screen

Users are able to manage the trip details by clicking on the respective trip on the home screen. Then, the system will fetch all the data of the trip and navigate users to the "Trip Detail" screen (UI refer to A-20).

In the "Trip Detail" screen, there is a "Show Map" toggle bar at the top right of the screen. Initially, the toggle bar is in the "Off" state. As users press on it, the toggle bar will change to an "On" state and the map with the attraction markers will display at the top of the screen (UI refer A-21). The map is handled by the Google Cloud Map API (code refer to B-11). All the pink color markers pinned on the map are based on the attractions listed on the respective day of the trip. In addition, there is a "Map Recenter" button on the bottom left of the map. When users press on it, the map will be centred to include all the markers in the map display (code refer to B-12). Furthermore, the system will direct users to the respective "Attraction Detail" screen (UI refer to A- 24) when users either click the pin on the map or the entry in the trip list below the days' selection.

In addition, users are able to see the details of each day of the trip by clicking on the day selection provided on the screen. The selection is based on the start date and end date of the trip, which are entered by the users during the creation (code refer to B-13). Moreover, users are able to move the sequence of the attraction by long pressing it and dragging it to the intended position. The system will then send the most recent sequence to Google Firebase (code B-14).

Furthermore, users are also able to delete the attraction entry in the "Trip Detail screen" by clicking the "delete" icon in the respective entry. After pressing the "delete" icon, the system will remove the attraction from the list in the Google Firebase (code refer to B-15). After that, the system will display a success message with an "UNDO" button at the bottom of the screen. If users press the "UNDO" button, the system will add the attraction back to the original sequence of the original day (code refer to B-16). If the attraction is not schedule in the selected day, it will not be pinned as pink marker. So, the pink marker will change back to the original category representative marker after deletion.

Moreover, users are able to move the attraction from one day to another within the same trip by clicking the "share" button on the respective entry. If "share" button is clicked, the system will pop out a selection dialog box to allow users to choose which day they intend to move. After moving the attraction to the other day, the marker of that attraction will no longer in pink color on the current day.

Last but not least, users can still add more attractions to the trip by clicking the "add" button on the right hand side of the app bar. Then, the system will navigate users to the "Attraction List" screen (UI refers to A-22).

5.4.8 Attraction List Screen

In the "Attraction List" screen, there are a few filter options on the top of the screen in order for the users to search for the intended attraction faster. Initially, the system will show all the attractions to the users. The selected filter option and the number of the attractions that fall into the respective category will be shown below the filter option. If users select another filter option, the selected filter option turns blue, and the list of attractions changes based on the filter option (UI refer to A-23).In addition, users are able to add the attraction to the trip by clicking the "add" button on the respective entry. When a user clicks the "add" button, the system will show the days selection dialog box for users to choose the day they intend to add (refer code B-16). If the the trip already contains the attraction, an error message will be displayed and unable to add the attraction to the trip. By default, each day in the trip maximum can allocate 3 normal attractions and 3 F&B attraction. If the limit is reached, the system will prohibit users add the respective attractions to the trip anymore. If users are interested in a particular attraction, they are able to click on the respective attraction. The system will direct users to the respective "Attraction Detail" screen (UI refer to A-24).

5.4.9 Attraction Detail Screen

In the "Attraction Detail" screen, users are able to check the details of the attraction. In addition, they are able to click on the phone number (if available) in order to call the management team of the attraction. Furthermore, users are able to open the Google Maps application on their mobile phones by clicking the "Press Me to Open Google Direction" button. Then, the system will navigate users to the Google Map application and direct users from the current location to the respective attraction location. Nevertheless, users are also able to expand the description by clicking on it in order to know more about the location (UI refer to A-25).

5.4.10 Smart Routing Function

We have made some assumptions on it, such as the total number of activities in a day being no more than six (3 for normal activities and 3 for F & B). If there are fewer F & B activities than normal activities, F & B activities cannot stick together because we assume users are full after one meal. This assumption is vise versa.

In order to trigger this function, users have to click the "Smart Routing" button on the "Trip Detail" screen (refer UI Fig A-21). After the button is clicked, the "build route" function is triggered (refer code B-17). The route optimization function is explained in the following.

First, the function will get all the selected attractions on the trip. After that, all the selected attractions are added into a list where the hotel location is the first element, followed by all normal activities and then F&B activities. Moreover, a matrix consisting of all the possible paths is built.

$$\begin{array}{r|ccccc|}
 & & \multicolumn{2}{c}{\text{Normal Attraction}} & \multicolumn{2}{c}{\text{F\&B activity}} \\
\text{Hotel/Hostel} \rightarrow & 0 & 3 & 7 & 1 & 3 \\
\text{Normal Attraction} & 3 & 0 & 2 & 8 & 8 \\
 & 7 & 2 & 0 & 9 & 6 \\
\text{F\&B activity} & 1 & 8 & 9 & 0 & 7 \\
 & 3 & 8 & 6 & 7 & 0
\end{array}$$

Let take the above matrix as an example. As we can see that, the diagonal of the matrix is zero because one location to to itself is zero. In addition, the distance between two points are fixed no matter go to or back from the point. Thus, row i is same as column i. Let take the element which is enclosed by a circle as example. It indicate that the distance from element 3 (the second normal attraction) to element 2 (the first normal attraction) is 2 or vice versa.

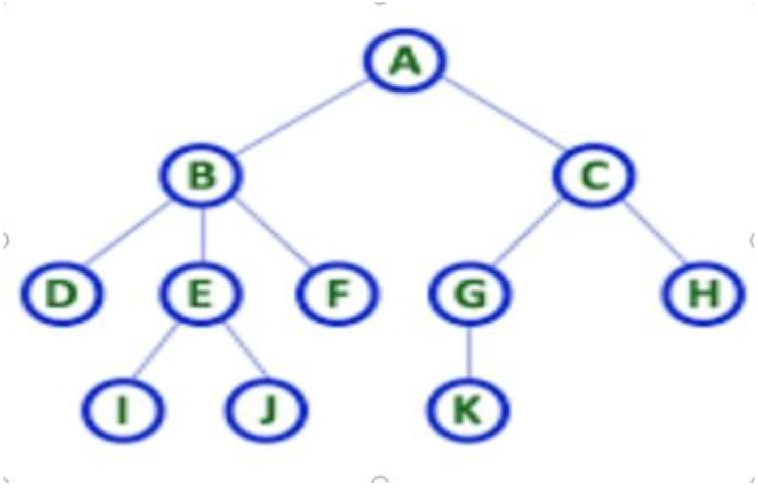
Node Class

Data Type	Name	Function
Integer	Vertex	index number
Integer	Level	record which level the node in
List of integer	Path	record all the vertex along the path
Nested list of double	Matrix_reduced	2D vector of distance in order to know where the point and parcel can go
Integer	Total distance	record the total distance along the path
Integer	numAttrLeave	Number of normal attraction leave
Integer	numFAndBLeave	Number of F&B activities Leave

Boolean	isFAndB	Check whether the node is F&B or not
Boolean	isAttr	Check whether the node is normal attraction or not

Table 5-3 Node Class Parameter

Brief idea to find the optimal path



All the explanations in this section will be based on the tree above. For example, A is a starting point (Hotel/Hostel). The system will search all the points to figure out where node A can move to. Based on the tree above, node A can move to node B and node C. So, node B and node C are created as nodes and all the information is generated. The node B node and node C node will store the total distance from A->B and A->C, respectively, and add them to the priority queue. Next, the system will remove the minimum-cost node from the priority queue and assign it as the minimum-cost node. For example, if node B is the minimum node, it will search all the possible points where the minimum node (node B) can go to. In this case, it should be node D, node E and node F. All these three points are generated and the cost will be the cost of the minimum node (node B) added with the cost from node B to itself (B->D, B-> E, B-> F). The process is repeated until the minimum node has not more path it can go. Then, the path of minimum node is the shortest path among other.

Find Shortest Path function (refer code B-18)

It is responsible to find the shortest path which trigger by "smart route" function. It receive three parameters which are starting node(Hotel/Hostel), number of normal attraction, number of F&B activities and total length of matrix (number of normal attraction + number of F&B activities + 1).

First, it will create a priority list which sort all the node in the list in ascending order in term of the total distance. Then, a while loop is start and stop when the shortest path is found.

Inside the while loop, it will remove the first element in the priority list (the element with the smallest distance) and assign it as the minimum node. The minimum node will be evaluated as follows:

1. If minimum node is at the level of last level (total length -1)
    - Return the path of minimum node
    - Function end
2. if minimum node is F&B activity
    - If the number of F&B activities leaves smaller or equal to number of normal attraction leave
        - Expand the tree with all the possible normal attraction that minimum node can go to (but not the F&B activities because we want find the best location to allocate the normal attractions)
        - Add the child node to the priority list
    - If not
        - Expand the tree with all the possible node that minimum node can go to (include all normal attraction and F&B activities)
        - Add the child node to the priority list
3. if minimum node is normal attraction
    - If the number of normal attraction leave smaller or equal to F&B activities leaves
        - Expand the tree with all the possible F&B activities that minimum node can go to (but not the normal attraction because we want find the best location to allocate the F&B activities)
        - Add the child node to the priority list
    - If not
        - Expand the tree with all the possible node that minimum node can go to (include all normal attraction and F&B activities)
        - Add the child node to the priority list

At the end of the loop, we have to sort the priority list in term of the total distance of the loop

The while loop continue loop until condition 1 is reached and return the shortest path.

<u>How to find the possible path?</u>

$$\begin{array}{r|ccccc|}
 & & \multicolumn{2}{c}{\overbrace{\text{Normal Attraction}}} & \multicolumn{2}{c}{\overbrace{\text{F\&B activity}}} \\
\rightarrow & 0 & 3 & 7 & 1 & 3 \\
\text{Normal Attraction} \left\{ \right. & 3 & 0 & 2 & 8 & 8 \\
 & 7 & 2 & 0 & 9 & 6 \\
\text{Normal Attraction} \left\{ \right. & 1 & 8 & 9 & 0 & 7 \\
 & 3 & 8 & 6 & 7 & 0
\end{array}$$

Let's assume the matrix above is the matrix reduced of a minimum node and the vertex of that minimum node is 2. When we want to find the possible path that vertex 2 can go down, we should refer to the third row (index starts from 0). If the path from vertex 2 to the other vertex is not 0, indicate that it is a possible path (2,0), (2,1), (2,3), (2,4)). If the minimum node can only go for F&B activities, then the scanning will start from column 3.

<u>Create a new node function (refer to code Fig 6-19)</u>

The parameter receive from caller are show on the following :

Data Type	Name	How to generate ?
Integer	Child Vertex	The vertex going to generate
Integer	Parent Vertex	Vertex of parent node
Integer	Parent Level	Level of parent node
List of integer	Parent Path	The path of parent node
Nested list of double	Parent Matrix	Matrix reduce of parent node
Integer	Parent distance	total distance of parent node
Integer	Parent numAttrLeave	Number of normal attraction leave of parent node
Integer	Parent numFAndBLeave	Number of F&B activities

		Leave of parent node
Boolean	Child is normal attraction	Check whether the child node which going to create is normal attraction or not

Fig 5-4 Create New Node Parameter

The information of Child node going to create :

Data Type	Name	How to generate ?
Integer	Vertex	Child vertex
Integer	Level	Parent level + 1
List of integer	Path	Parent path with it self (child vertex)
Nested list of double	Matrix_reduced	Further reduce from parent matrix.(block the path from the parent to the child and all the path from other to this parent again)
Integer	Total distance	Parent distance + the distance from parent to child
Integer	numAttrLeave	If child is normal attraction, then reduce the parent numAttrLeave by 1. if not, remain as parent numAttrLeave.
Integer	numFAndBLeave	If child is normal attraction, remain as parent numFAndBLeave. If not, reduce the parent numAttrLeave by 1.
Boolean	isFAndB	Opposite of "Child is normal attraction"

Boolean	isAttr	Same as "Child is normal attraction"

Fig 5-5 Child Node Information

5.4.11 Create New Attraction

Users able to promote their business by adding the attraction in the mobile application. In order to navigate to the “Manage Attraction” screen, users have to click “Manage Attraction” button in the drawer. In “Manage Attraction” screen (refer UI Fig A-26, “No attraction own by you” caption is show in the middle of the screen when there are not any attraction created by you. Users have to click the “add” button at the top right corner to create an attraction. In “Attraction creation” screen (refer UI Fig A-27), users have to enter all the require information in the correct format before successfully create the attraction. Once “Create Attraction” button is clicked, all the entry will be evaluated by the system (refer code Fig B-17). If there are any entry do not fulfill the requirement, error message will show (refer UI Fig A-28). If the creation is success, the success message will show and the system will navigate users back to “Manage Attraction” screen. In addition, the attraction which just created also present at the top of the attraction list (refer UI Fig A-29).

5.4.12 Manage Existing Attraction

After users create the attraction, they are able to update the information of the existing attraction even though delete the existing attraction. For editing the attraction, users have to click the “edit” button on the respective entry. Then, the system will navigate users to “Update Attraction” screen with same as “Create Attraction” screen(refer UI Fig A-27) but the system will auto fill in all the information of the attraction. When users click “Update Attraction” button, the system will evaluate all the entry again and update the attraction accordingly. If there are any invalid input, the error message will show.

If users intend to delete the attraction, they are able to do so by clicking the “delete” button on the respective attraction. Then, the system will pop out the confirmation dialog box which require users to further confirm the deletion process (refer UI FigA-30). If users click yes, the system will delete the attraction from the Google Firebase. If no, nothing will happen.

5.4.13 Attraction Map

When users click "Attraction Map" in the drawer, the system will navigate users to a map screen with all the attractions in the application in the markers format (refer UI Fig-31). Users able to get more information about particular attraction by clicking the marker. Then, the system will navigate users to "Attraction Detail" screen with all the information of selected attraction. But, there is not "Add" button on the screen. Furthermore, users are able to perform filter on "Attraction Map" screen by selecting the category (refer UI Fig A-32)(refer code Fig B-20). After filter is applied, the markers on the map only show those attractions related to the selected category (refer UI Fig A-33). If all the filters are unchecked, all the attraction will be displayed.

5.4.14 Social Platform

Users are able to navigate to social platform in the mobile application by clicking "Social Platform" button in the drawer (refer UI Fig A-34). If users wish to create a new post, they have to click the "add" button at the top right corner of "Social Platform" screen. Users have to enter one of the information in correct format (Among pictures, description and itinerary plan) in order to create the post successfully. If not, error message will show(ref UI Fig A-35). If users wish to upload a plan in the post, they able to select which itinerary plan they going to upload (refer UI Fig A-36). After the selection of itinerary plan, the system will navigate users to preview the selected trip (refer UI Fig_37). Users can choose either cancel the importation or continue to import the plan. After the post is created successfully, the system will navigate users back to "Social Platform" screen. In "Social Platform" screen, you should see the the post you just created is at the top of the post list.

For each post, if there is a plan integrated with the post, users should see a paper clip icon on the top right corner. Inside the paper clip icon, users can either preview the itinerary plan first or straightly import the plan. If users decide to download the itinerary plan, the system will navigate users to "Trip Creation" screen (refer UI Fig A-17). However, the number of day in the creation must same as the imported trip duration. If not, error message will show (refer UI Fig A-38).

In addition, users are allow to edit or delete the post if the post is created by him/her. The creator of the post should see three dot at the top right corner of the post.

Three dot icon have two selections which are edit the post and delete the post. If users choose edit the post, the system will direct user to "Edit Post" screen which same as "Create Post" screen. If users choose the delete option, a delete confirmation will pop out. Users able to click "Yes" button to confirm the deletion.

Nevertheless, users able to like any post in social platform by clicking the like icon. If the post is liked by users, the like icon will be in blue color else, it will be in gray color.

Comment toward the post also can be done by users by clicking the comment icon. Then, the system will direct users to comment screen of the post. If there are not any comment, a tag "No Comment Available" is shown in the center of the screen (refer UI Fig A-39). Users are able to post the comment by typing it on the space provide at the bottom of the "Comment" screen. The latest comment will be at the bottom of the list (refer UI Fig A-40). Furthermore, users are able to delete the comment which is created by themselves by clicking the "delete" button on the comment.

5.5 Implementation Issues and Challenge

These are a few of the implementation issues and challenges during the development of the itinerary planner mobile application. Since the itinerary planner application requires real data in order to make the application realistic. As a result, we will need more time and patience to collect all of the information and import it into Google Firebase.Since the application adopts some of the external APIs, such as Google Firebase, Google Cloud Platform, and so on. The version of the external API may be upgraded from time to time, which is uncontrollable by us. The mobile application may be incompatible with the upgraded version. Thus, we have to figure out the problem and check the documentation of the upgraded version of the API in order to solve it. This process is time-consuming. Furthermore, the algorithm of the attraction arrangement is complicated and has to take many conditions into account, such as shortest path, the arrangement of activities and so on. Thus, we have to try to put many solutions into the application and adopt the best solution. Since we are using Google Firebase as the database of the mobile application and there is a limit in Google Firebase Storage in each day, it is often to use up the quota provided. Thus, we have to wait until the quota is available and continue test our system.

Chapter 6: System Evaluation and Discussion

6.1 System Testing and Performance Metric

In order to test the mobile application deeply, both system verification and system validation plans have been used. For system verification plans, unit testing and integration testing are applied in the itinerary mobile application in order to make sure that the system works as expected. After a unit of a module is developed, unit testing is applied to the developed unit only. As a result, we are able to check whether the module is well developed or not. In addition, it is also able to prevent the integrated version of the application from being corrupted when the module with bugs is integrated with it. After the unit testing, the module will be integrated with the other modules that have been developed before. Thus, the new version of the integrated mobile application is produced. Furthermore, integration testing also applies to the new version of the integrated mobile application. After performing both unit testing and integration testing throughout the development process, it was able to test the reliability of the mobile application.

After the mobile application is developed, a validation plan is carried out in order to get some of the feedback from the floor. The well developed mobile application is sent to 20 users for testing purposes. After users test out the application, they have to fill in a survey form regarding their feedback towards the itinerary mobile application.

6.2 Verification Plan Result

Test Case	Test Description	Test Data	Expected Outcome	Actual Outcome	Pass / Fail
1	Check response when valid email and password is enter	Email : jas@gmail.com Password : 123456	Login success and the system direct user to home page	Login success and the system direct user to home page	Pass
2	Check response when invalid	Email : jasmine@gmail.com	Login fail and the error	Login fail and the error	Pass

	email and valid password is enter	Password : 123456	message is shown	message is shown	
3	Check response when valid email and invalid password is enter	Email : jas@gmail.com Password : 78906	Login fail and the error message is shown	Login fail and the error message is shown	Pass
4	Check response when invalid email and invalid password is enter	Email : jasmine@gmail.com Password : 78906	Login fail and the error message is shown	Login fail and the error message is shown	Pass

Table 6-1 Test Scenario: Check Login Functionality

Test Case	Test Description	Test Data	Expected Outcome	Actual Outcome	Pass / Fail
1	Check the response when all the information require is enter in correct format	Nick Name: Jasmine E-mail : chongjasmine0827@gmail.com Password : 123456 Gender : Ms	Create an account successfully and the system direct user to home page	Create an account successfully and the system direct user to home page	Pass
2	Check the response when all the information require is enter in correct format except nick name is empty	Nick Name: “Empty” E-mail : chongjasmine0827@gmail.com Password : 123456 Gender : Ms	Create an account unsuccessfully and error message is shown	Create an account unsuccessfully and error message is shown	Pass
3	Check the	Nick Name: Jasmine	Create an	Create an	Pass

	response when all the information require is enter in correct format except email is invalid (without @ or empty)	E-mail : chongjasmine0827 Password : 123456 Gender : Ms ------------------------ Nick Name: Jasmine E-mail : "Empty" Password : 123456 Gender : Ms	account unsuccessfully and error message is shown	account unsuccessfully and error message is shown	
4	Check the response when all the information require is enter in correct format except password is invalid (less than 6 digit or empty)	Nick Name: Jasmine E-mail : chongjasmine0827@gmail.com Password : 123 Gender : Ms ----------------------- Nick Name: Jasmine E-mail : chongjasmine0827@gmail.com Password : "Empty" Gender : Ms	Create an account unsuccessfully and error message is shown	Create an account unsuccessfully and error message is shown	Pass
5	Check the response when all the information require is enter in correct format except gender is not selected	Nick Name: Jasmine E-mail : chongjasmine0827@gmail.com Password : 123456 Gender : "Empty"	Create an account unsuccessfully and error message is shown	Create an account unsuccessfully and error message is shown	Pass

Table 6-2 Test Scenario: Sign Up Functionality

Test Case	Test Description	Test Data	Expected Outcome	Actual Outcome	Pass / Fail
1	Check response when invalid email is enter (not existing account)	Email : efbwji@gmail.com	A dialog box with error message is pop out	A dialog box with error message is pop out	Pass
2	Check response when empty email is enter	Email : "Empty"	Error message is shown	Error message is shown	Pass
3	Check response when valid email is enter	Email : chongjasmine0827@gmail.com	A dialog box with message is shown	A dialog box with message is shown	Pass

Table 6-3 Test Scenario: Forgot Password Functionality

Test Case	Test Description	Test Data	Expected Outcome	Actual Outcome	Pass / Fail
1	Check the response when clicking the profile picture and without select any image	Without select any images	the system do nothing and direct users back to "Setting" page	the system do nothing and direct users back to "Setting" page	Pass
2	Check the response when clicking the profile picture and select an image as profile picture	select an image as profile picture	Selected image upload to Google store and it become the user' profile picture.	Selected image upload to Google store and it become the user' profile picture.	Pass

3	Check the response when user enter empty during changing the user name	New User Name: "Empty	Error message is shown	Error message is shown	Pass
4	Check the response when user enter new user name during changing the user name	New User Name : newJasmine	1. Success message is shown at the bottom of the screen after updating 2. User name below the profile image is change to new user name	1. Success message is shown at the bottom of the screen after updating 2. User name below the profile image is change to new user name	Pass
5	Check the response when user enter empty new password and empty confirm password	New Password : "Empty" Confirm Password : "Empty"	Error message is shown	Error message is shown	Pass
6	Check the response when user enter the new password which different from confirm password	New Password : 12345678 Confirm Password : 123	Error message is shown	Error message is shown	Pass
7	Check the response when user enter the new password same as confirm password	New Password : 12345678 Confirm Password : 12345678	- Success message is shown at the bottom of the screen	- Success message is shown at the bottom of the screen	Pass

	in correct format		- able to login into the account by using the new password	- able to login into the account by using the new password	
8	Check the response when user enter the empty new email	New Email : "empty"	Error message is shown	Error message is shown	Pass
9	Check the response when user enter the new email in correct format	New Email : chongjasmine99 @gmail.com	- Success message is shown at the bottom of the screen - able to login into the account by using the new email	- Success message is shown at the bottom of the screen - able to login into the account by using the new email	Pass

Table 6-4 Test Scenario: Setting Functionality

Test Case	Test Description	Test Data	Expected Outcome	Actual Outcome	Pass / Fail
1	Check the response when hotel is selected	Select Lost World of Tambun as hotel	Auto inset longitute (101.151917 63474004) and latitude (4.64153737 89157)	Auto inset longitute (101.151917 63474004) and latitude (4.64153737 89157)	Pass
2	Check the	Click "Find My Location"	Auto inset	Auto inset	Pass

	response when "Find My Location" is clicked	button	longitute (101.1467299) and latitude (4.2848135)	longitute (101.1467299) and latitude (4.2848135)	
3	Check response when "Choose Date" or the selected date is clicked	Click "Choose Date" button -------------------------------- Click the selected date	The system pop out the calender for data selection	The system pop out the calender for data selection	Pass
4	Check response when all the required information is input in a correct format	Image : select from phone storage Trip name : Happy Family State : Perak City : Ipoh Hotel : Lost World of Tambun Longitude : "input automatically" Latitude : "Input Automatically" Date : 2022/09/27 - 2022/09/29	- Success message is shown at the bottom of the screen - The system direct user to the "Attraction List" screen	- Success message is shown at the bottom of the screen - The system direct user to the "Attraction List" screen	Pass
5	Check response when one or more the required information are input in invalid format	Image : select from phone storage Trip name : "Empty" State : Perak City : Ipoh Hotel : Lost World of Tambun Longitude : "input automatically"	Creation is fail and error message is shown	Creation is fail and error message is shown	Pass

		Latitude : “Input Automatically” Date : “Empty			
6	Check response when image is not selected and other information is input in correct format	Image : “Empty” Trip name : Happy Family State : Perak City : Ipoh Hotel : Lost World of Tambun Longitude : “input automatically” Latitude : “Input Automatically” Date : 2022/09/27 - 2022/09/29	- Success message is shown at the bottom of the screen - The system direct user to the “Attraction List” screen	- Success message is shown at the bottom of the screen - The system direct user to the “Attraction List” screen	Pass

Table 6-5 Test Scenario: Create a New Trip Functionality

Test Case	Test Description	Test Data	Expected Outcome	Actual Outcome	Pass / Fail
1	Check the response when the existing trip in home page is clicked	Click exiting trip in home page (three day in the trip)	- The system direct user to the “Manage trip detail” screen - three selection of the day is available	- The system direct user to the “Manage trip detail” screen - three selection of the day is available	Pass
2	Check the response when the “Show Map” toggle button is click	Click “Show Map” toggle button	The map with markers(all the selected attractions in the selected day) is shown.	The map with markers(all the selected attractions in the selected day) is shown.	Pass
3	Check the response when	Click the marker on the	The system directs users to respective	The system directs users to respective	Pass

	the marker or the entry in the list of attraction is clicked.	map ------------- Click the name of the attraction in the list	"Attraction Detail" screen	"Attraction Detail" screen	
4	Check the response when the focus button on the left corner of the map is clicked	Click the focus button on the left corner of the map	The map will cover all the markets in the fit region	The map cover all the markets in the fit region	Pass
5	Check the response when the attraction is drag from last to top	Drag the last attraction in the selected day to the top	The attraction will move it to the top of the list	The attraction move it to the top of the list	Pass
6	Check the response when share button is clicked and share from day 1 to day 2	Click share button and select day 2	- The attraction will move from day 1 to day 2 and display the success message at the bottom of the screen - the marker of the attraction change from pick color to the respective representation marker	- The attraction move from day 1 to day 2 and display the success message at the bottom of the screen - the marker of the attraction change from pick color to the respective representation marker	Pass
7	Check the response when	Click delete button	- The attraction will be removed	- The attraction is removed and the	Pass

	delete button is clicked		and the success message is shown at the bottom of the screen - the marker of the attraction change from pick color to the respective representation marker	success message is shown at the bottom of the screen - the marker of the attraction change from pick color to the respective representation marker	
8	Check the response when the "Undo" button in the success message (after delete or share) is click	Click the "undo" button in the right most of the success message	- The arrangement back to previous. - the marker of the attraction change from respective representation marker to pink color marker	- The arrangement back to previous. - the marker of the attraction change from respective representation marker to pink color marker	Pass
9	Check the response when "Smart Routing" button is clicked	Click the "Smart Routing" button	- The system auto rearrange all the selected attraction in the trip	- The system auto rearrange all the selected attraction in the trip	Pass
10	Check the response when click the "Add" button at the top right corner of "Trip Detail" screen.	Click the "Add" button at the top of the "Trip Detail" screen	The system direct user to "Attraction List" screen	The system direct user to "Attraction List" screen	Pass
11	Check the	Click	The system will	The system only	Pass

	response when user choose the filter of the attraction list	"Restaurant" filter	only show all the attractions under restaurant category.	show all the attractions under restaurant category.	
12	Check the response when user click the entry in the attraction list	Click one of the entry in attraction list	The system will direct user to the respective "Attraction detail" screen.	The system direct user to the respective "Attraction detail" screen.	Pass
13	Check the response when user add the attraction to the trip (attraction in the trip still have available slot for F&B and attraction)	Click the "Add" button	The system add the attraction to the selected day and success message is shown	The system add the attraction to the selected day and success message is shown	Pass
14	Check the response when user add the attraction to the trip (attraction in the trip still have available slot for F&B but not attraction)	Click the "Add" button of the attraction (Not under F&B categoty)	Error message is shown	Error message is shown	Pass
15	Check the response when user add the	Click the "Add" button of the	The system add the attraction to the selected day and	The system add the attraction to the selected day and	PAss

	attraction to the trip (attraction in the trip still have available slot for F&B but not attraction)	attraction (under F&B categoty)	success message is shown	success message is shown	
16	Check the response when user clicks the "Map" button on the top right corner of the" Attraction List" screen	Click the "Map" button	The system will direct user to "Trip Detail" screen	The system direct user to "Trip Detail" screen	Pass

Table 6-6 Test Scenario: Manage Trip Functionality

Test Case	Test Description	Test Data	Expected Outcome	Actual Outcome	Pass / Fail
1	Check the response when input all the data in correct format	Image : Pick from phone storage Name : Happy Playground Fee : 5.00 Rating : 3.3 Start Time : 1800 End Time : 2400 Tel : 0127997773 State : Perak City : Kampar Longitude :	Successfully created the attraction and success message is shown	Successfully created the attraction and success message is shown	Pass

		101.1467253 Latitude : 4.2848182 Category : Night Life			
2	Check the response when input some of the data are invalid	Image : Pick from phone storage Name : "Empty" Fee : 5.00 Rating : 3.3 Start Time : 1800 End Time : 2400 Tel : 0127997773 State : Perak City : Kampar Longitude : "Empty" Latitude : "Empty" Category : Night Life	Fail to create new attraction and error message is shown	Fail to create new attraction and error message is shown	Pass
3	Check the response when "Find My Location" is clicked	Click "Find My Location" button	Auto inset longitute (101.1467299) and latitude (4.2848135)	Auto inset longitute (101.1467299) and latitude (4.2848135)	Pass

Table 6-7 Test Scenario: Create New Attraction Functionality

Test Case	Test Description	Test Data	Expected Outcome	Actual Outcome	Pass / Fail
1	Check the response when editing the existing attraction with valid data	Change the Name from Happy Playground to New Happy Playground	Change is made and success message is shown	Change is made and success message is shown	Pass

2	Check the response when editing the existing attraction with invalid data	Change the Name from Happy Playground to “Empty”	Fail to change and error message is shown	Fail to change and error message is shown	Pass
3	Check the response when “Delete” icon is clicked	Click the “delete” icon	Confirmation dialog box is shown. If click “No”, nothing happen If click “Yes”, the respective attraction will be deleted.	Confirmation dialog box is shown. If click “No”, nothing happen If click “Yes”, the respective attraction is deleted.	Pass

Table 6-8 Test Scenario : Manage Attraction Functionality

Test Case	Test Description	Test Data	Expected Outcome	Actual Outcome	Pass / Fail
1	Check the response when “Attraction Map” is enter by clicking the button in drawer	Click “Attraction Map” button in the drawer	The system direct user to “Attraction Map” screen. The Map contain all the attraction available in the application in the form of marker	The system direct user to “Attraction Map” screen. The Map contain all the attraction available in the application in the form of marker	Pass
2	Check the response when filter is applied	Apply the “restaurant” filter	Only those attractions under restaurant category are	Only those attractions under restaurant category are	Pass

			shown	shown	
3	Check the response when unchecked all the filter	Unchecked all the filter	All the attractions in the application are shown on the map	All the attractions in the application are shown on the map	Pass

Table 6-9 Test Scenario : Attraction Map Functionality

Test Case	Test Description	Test Data	Expected Outcome	Actual Outcome	Pass / Fail
1	Check the response when create a post with valid data	Image : Pick from phone storage Description: it is a great trip Import Plan : Happy Trip	Create the post successfully and the success message is shown	Create the post successfully and the success message is shown	Pass
2	Check the response when create a post with no data	Image : “Empty” Description: "Empty” Import Plan : “Empty”	Fail to create and error message is shown	Fail to create and error message is shown	Pass
3	Check the response when the post going to import a plan	Click “Import Plan” button to import the plan	A dialog box is pop out with all the existing plan selection. Then, the system will navigate user to preview the plan	A dialog box is pop out with all the existing plan selection. Then, the system will navigate user to preview the plan	Pass

			when user select one of it	when user select one of it	
4	Check the response when edit the existing post with valid data	Change description to “Welcome to Kampar”	The change is mad and success message is shown	The change is mad and success message is shown	Pass
5	Check the response when edit the existing post with empty description	Change description to empty	The change is mad and success message is shown	The change is mad and success message is shown	Pass
6	Check the response when edit the existing post with all empty input	Change all input to empty	The change is fail and error message is shown	The change is fail and error message is shown	Pass
7	Check the response when like the post	Click the “Like” button which is gray color	The “Like” button change to blue and the number of like increase by 1	The “Like” button change to blue and the number of like increase by 1	Pass
8	Check the response when dislike the post	Click the “like” button which in blue color	The “Like” button change to gray and the number of like decrease by 1	The “Like” button change to gray and the number of like decrease by 1	Pass
9	Check the response when a comment is created	Create a new comment	The comment is publish under the post and the number of comment increase by 1	The comment is publish under the post and the number of comment increase by 1	Pass

10	Check the response when a comment is deleted	Click the "delete" button under the comment	A confirmation dialog box is shown If click "No", nothing happen If click "Yes", the comment is deleted and the amount of comment is decrease by 1	A confirmation dialog box is shown If click "No", nothing happen If click "Yes", the comment is deleted and the amount of comment is decrease by 1	Pass
11	Check the response when a post is deleted	Click the "delete" button under the post	A confirmation dialog box is shown If click "No", nothing happen If click "Yes", the post will be deleted	A confirmation dialog box is shown If click "No", nothing happen If click "Yes", the post is deleted	Pass

Table 6-10 Test Scenario : Manage Post Functionality

6.3 Validation Plan Result

A customer acceptance test is conducted in the validation plan in order to verify how good the system is toward the users and what their comments are toward our system in Kampar, Perak. In total, there were 18 users conducting the survey. Firstly, they will try out the application we have built. After they have some experience with the application, they will be requested to fill up the google form, which consists of eight questions, which are gender, age, employment status, how satisfied with the system they are, what is the module they are most satisfied with, what is the opinion that the mobile application is able to boost the tourism of the rural area and why they said so, and what the opinion toward the application for future improvement.

Among all the responses, 50% were male and 50% female (Fig. C–1). As a result, it is necessary to strike a balance with regard to the survey. For age range, there are 55.6% in the range between 26 and 40 years old, which consists of the most people among other age ranges 27.8 % in the range between 18 years old and 25 years old and 16.7% in the range between 40 years old and 50 years old (Fig C-2). In addition, there are four employment statuses which are employed(27.8%), self-employed(27.8%), student(27.8%) and unemployed(16.7%)(Fig C-3).

Based on Fig C-4, we are able to observe that most of the users are satisfied with our system. The most welcome models are the trip creation and management module and the social platform module, which consist of 33.3% each. This indicated that these two modules were able to produce the outcome they intended to and satisfy most of the users' needs.

The reason why we require to know about the employment status and the age of users is that we are able to know what type of module is favoured for which category of users. Based on the survey, 83.3% of users said that the itinerary planner mobile application is able to boost tourism in rural areas, while 16.7% said maybe, and no one rejected this idea. Based on the reasons they provided to justify their selection, most of the self-employed respondents said that the application was able to help them promote their business and allow tourists to see them. In addition, some of the respondents also claim that the application gathers all the information of interest and they do not have to search for the information from scratch. The smart routing system improves the user experience. This kind of feedback comes from employed respondents most of the time. Nonetheless, unemployed and student respondents lean more toward social media platforms. They claim that the social platform allows them to have interaction with other travellers and allows them to share the experience with others as well. Some of the users also claim that sharing the plans is a fantastic function that allows them to immediately adopt those well-planned plans for their trip.

There is much feedback from the respondents for the future improvement of the application. Most of the respondents claim that the information available in the application should have even more and be presented in a realistic way. In addition, they also said that we should have a group of users edit the itinerary plan together instead of a user alone. Although the smart routing function is cool, it still requires further improvement to make it more efficient. In addition, some of the respondents also suggest that we provide some games inside the application in order to improve the user experience of tourists. Furthermore, some of the users also claim that weather problems should be considered during planning as well as guide them through the trip.

6.4 Objective Evaluation

In the project, there are total three objectives we have to achieve which are

1. **To analyse the existing mobile application, which is related to the itinerary planner on the Google Play Store or Apple Store, in order to identify the strengths, weaknesses, and limitations of the application**
2. **To develop an itinerary planner mobile application that assists users in managing their itinerary plans well by adopting the strengths of the existing application and resolving the weaknesses and limitations of the existing application with some of the novelty concept to enhance the mobile application**
3. **To evaluate the functionalities and efficiency of itinerary planner mobile application by using verification plan and validation plan.**

We are going to evaluate each of the objective at the following.

We have achieved the first objective at an early stage by analysing a total of six existing mobile applications that are related to itinerary planning, no matter in the Google Play store or the App Store. In addition, we also performed comparisons among these mobile applications in order to figure out what the needs of the users were. All the details of each mobile application are documented in chapter 2.

Furthermore, we have developed an itinerary mobile application by adopting the strength and resolving the limitations of an existing mobile application as well as enhancing the application by using some of the novelty concepts to achieve the second objective. For example, we adopted the route optimization technique in Wanderlog to help users optimise their route with just a click. In addition, effective searching techniques in Sygic Travel Maps Offline & Trip Planner, Kayak, and Tourist Assist-TAIS are adopted to resolve the searching problem in Wandelog. Nevertheless, the Google Maps API is used in our proposed mobile application in order to provide a visual view to users and make the map more informative and useful. By doing so, ineffective map information in Wandelog is solved. Since our application allows users to create an attraction and publish it to others, If they wish to update the information in order to provide up-to-date information to users, they are able to do so. As a result, the application has no problem losing control over updating the data which is present in the Kayak application. In the proposed application, a social platform is included, which allows all the users to interact with each other. As a result, they are able to widen their network through the application. Moreover, users are able to upload and download the itinerary plan on the social platform. This is a great function which is used by the surveyors because it allows them to adopt the plan directly by downloading it into their account.

After the development phase, we also performed system evaluation by using the verification plan and validation plan. In the verification plan, each function in the system has been tested with valid and invalid data. In addition, we also conducted a survey for the users after they tried out the itinerary mobile application we developed. All the evaluation data and analysis are documented in sections 6.2 and 6.3.

In conclusion, all the objectives we set initially have been achieved through all the phases of the Rapid Application Development process.

Chapter 7: Conclusion and Recommendation

7.1 Conclusion

To sum up, the main purpose of this project is to build the itinerary mobile application to solve most of the problems that existed during trip planning. In the age of technology, **most of the questions can be figured out through searching using a search engine. However, it is only useful to those who are familiar with searching, not to those who are not**. If the travelers do not know how to explore the online resource, they may have difficulties during planning as well as dramatically increase the chances of problems occurring during the trip. This is because the information from the internet resource may not always be correct due to nobody at the back end verifying the correctness of the resource. In addition, travelers **may not be able to explore some of the hidden or classical foods or attractions in the destination**. This is because some of the rural areas, such as Gopeng, Perak or Kampar, Perak, may have some attractions organised by those who cannot follow the pace of the current age. Thus, they may not know how to use the internet to promote their products. So, the famous attractions are only known by the people who are living in that city. Furthermore, travelers may not fully understand the geographical location of all the attractions on their trip. So, they may have **difficulties arranging the sequence of the attractions during the trip**. They have to think of a way to arrange it and, based on their feelings, decide which routes are the best routes physically. This **process is very time-consuming and has a high chance of change.** The route may not be the best route. This is due to the fact that they do not have a proven way to guide them through completing the route formation. As a result, they may not have enough time to travel around all the attractions they intend to visit. Nevertheless, travelers **do not have the proper tool to record all the details of their trip** during planning. So, they have to plan the location of the attraction from scratch.

This project will solve all the problems mention above. Travelers can use the mobile application to manage their trip in a easy manner. First, itinerary planner mobile application **provide a channel for all the traveler to manage their trip easily**. They are able to create a new trip by entering the necessary information during creation. After that, they can **explore to most of the attraction information in destination**. In addition, users are able to add, modify and delete the attractions in every trips. Nevertheless, the mobile application provide a very cool function which is **auto build route function**. It will help the users arrange the attractions which users intend to visit in a best way. The auto build route function will group all the closer attractions together and figure out the shortest path to connect them. Of course, there are several consideration during the route formation which are breakfast, lunch and dinner consideration as well as Hotel location. If users are not satisfy with the arrangement, they **able to manage by themselves as well**. All the trip information will be save. So, travelers can refer to the application when they have any inquires regarding the trip.

Nevertheless, the application use the power of users to boots the attraction information in the application. In other word, users **able to create an attractions and publish to others to promote their business**. By doing so, some of the hidden attraction are able to promote on this mobile application.

Moreover, there is a **social platform available** in the application for all the users interact with each other. Users able to view the post which post by other users in the social platform of the application. In addition, users also able to create a post which may contain some caption or photos. Furthermore, users able able to post the itinerary plan to the social platform. Thus, for those who interest with the plan able to **import the itinerary plan from the social platform to theirs account**. So, they able to manage the imported itinerary plan and adopt it for travelling. In addition, users able to **react to the post** either by like or comment on the post. As the result, all the users able to **share their point of view and improve the knowledge of travel.**

7.2 Future Work

Due to the time constrain, we are not able to build the comprehensive itinerary planner mobile application. For future work, we able to improve the smart routing function which make it more effective and efficient. In addition, we are also able to provide some of the suggestion during planning. The suggestion can based on users preference. As the result, it able to provide the useful suggestion to users. Nevertheless, weather consideration can be included as one factor during planning. In addition, the system are also able to guide the users during the trip. This is because there are many unpredictable circumstance may occur during the trip such as traffic jam, raining and so on. Last but not least, admin portal can be created to perform some of the administration work such as update attraction information, create new attraction and so on. Furthermore, the creation of attraction done by users should pass to admin to verify it instead of publish directly. This is because there may have some offenders purposely build some of the fake attraction and publish to public. In addition, admin are also able to delete those unethical post or comment in social platform.

Reference

[1] Hafner, K. (2006, June 17). Growing wikipedia refines its 'anyone can edit' policy. The New York Times. https://www.nytimes.com/2006/06/17/technology

[2] Smirnov, A., Kashevnik, A., Shilov, N., Teslya, N. & Shabaev, A., 2014, Mobile application for guiding tourist activities: Tourist assistant - TAIS, Conference of Open Innovation Association, FRUCT, vols. 2014-December, 95–100, IEEE Computer Society.

[3] The 10 Best Travel Planner Apps of 2019,2021. [online] Available at: <https://www.spaceotechnologies.com/travel-booking-app-development-features-usa-booking-com/> [Accessed 15 August 2021].

[4] DPO blog. 2021. Everything You Ever Wanted to Know About Kayak.com - DPO blog. [online] Available at: <https://blog.dpogroup.com/everything-you-ever-wanted-to-know-about-kayak-com/> [Accessed 15 August 2021].

[5] Tourism industry: Everything you need to know about tourism (2020) Revfine.com. Available at: https://www.revfine.com/tourism-industry/ (Accessed: August 24, 2021).

[6] Yehia, Y. (2019) “The importance of tourism on economies and businesses,” Msu.edu. globalEDGE, 26 March. Available at: https://globaledge.msu.edu/blog/post/55748/the-importance-of-tourism-on-economies-a (Accessed: August 24, 2021).

[7] rad methodology - Google Search (no date) Google.com. Available at: https://www.google.com/search?q=rad+methodology&source=lnms&tbm=isch&sa=X&ved=2ahUKEwiPzO6M7NfyAhWnxDgGHaC2CooQ_AUoAXoECAEQAw&biw=1536&bih=754 (Accessed: August 30, 2021).

[8] 4 phases of rapid application development methodology (2018) Lucidchart.com. Available at: https://www.lucidchart.com/blog/rapid-application-development-methodology (Accessed: August 30, 2021).

[9] “Kiss (keep it simple, stupid) - a design principle,” The Interaction Design Foundation. [Online]. Available: https://www.interaction-design.org/literature/article/kiss-keep-it-simple-stupid-a-design-principle. [Accessed: 05-Sep-2022].

[10] “What is Fitts' law?,” The Interaction Design Foundation. [Online]. Available: https://www.interaction-design.org/literature/topics/fitts-law. [Accessed: 05-Sep-2022].

[11] “The laws of proximity and common region in UX design: Software country,” The Laws of Proximity and Common Region in UX Design | Software Country. [Online]. Available: https://softwarecountry.com/company/our-blog/laws-of-proximity-in-ui/. [Accessed: 05-Sep-2022].

[12] Y. Zane, “Jakob's law in UX design,” Medium, 30-Jul-2021. [Online]. Available: https://bootcamp.uxdesign.cc/jakobs-law-in-ux-design-48661c775a1b. [Accessed: 05-Sep-2022].

Appendix A

User Interface Screenshot

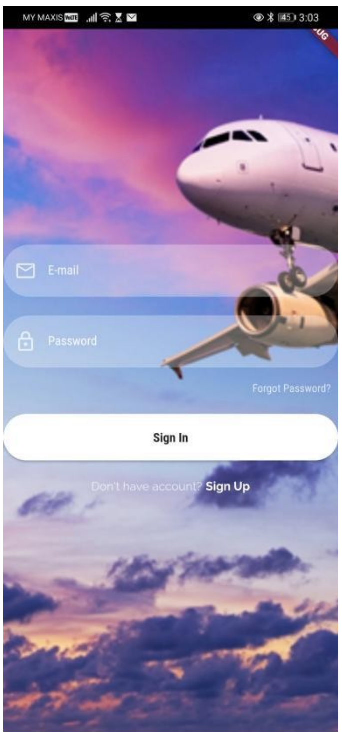


Fig A-1 Sign In Scree (No Error Message)

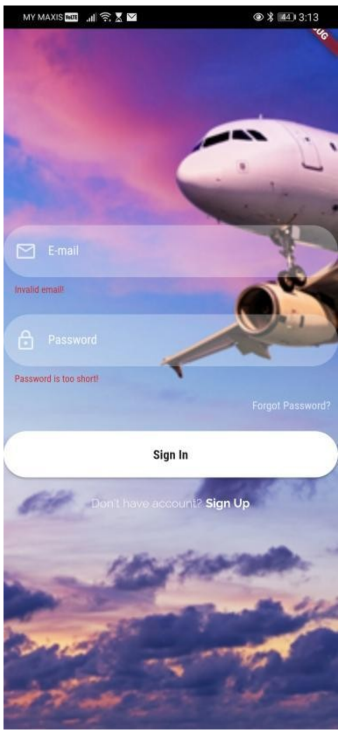


Fig A-2 Sign In Scree (Error Message)

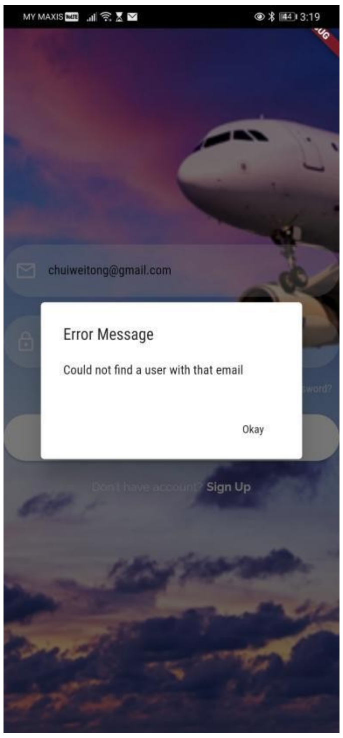

chuiweitong@gmail.com
Error Message
Could not find a user with that email
Okay
Don't have account? Sign Up

Fig A-3 Invalid Email Error Message

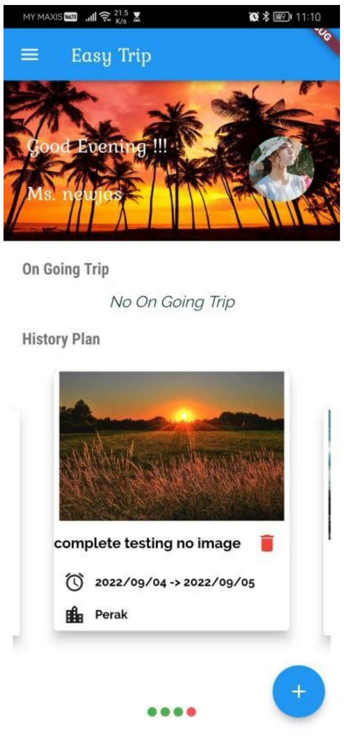


Fig A-4 Home Screen Part 1

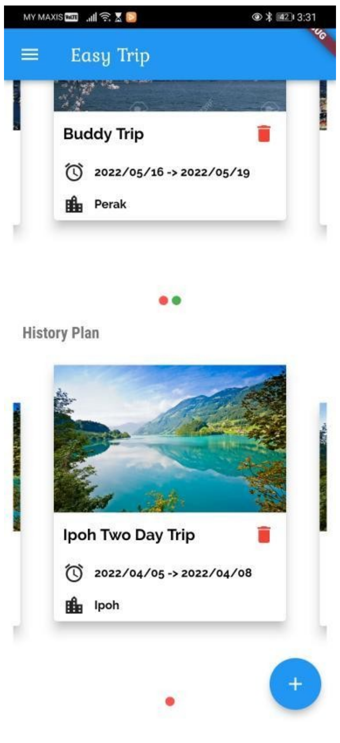


Fig A-5 Home Screen Part 2

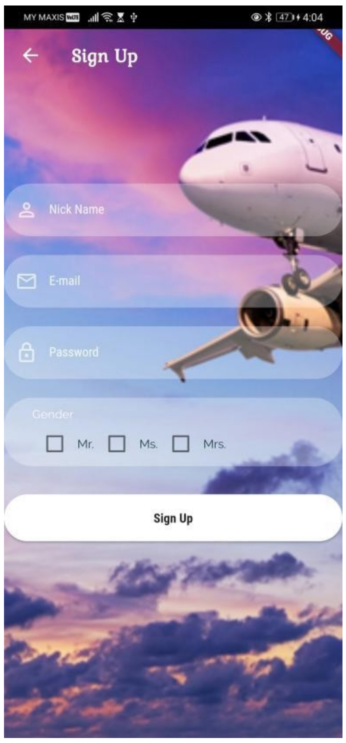


Fig A-6 Sign Up Screen(No Error Message)

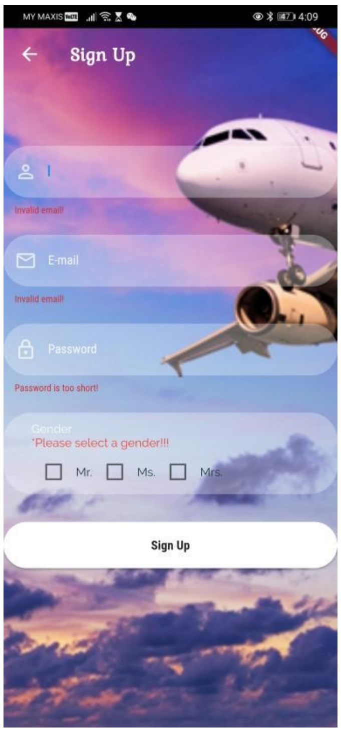
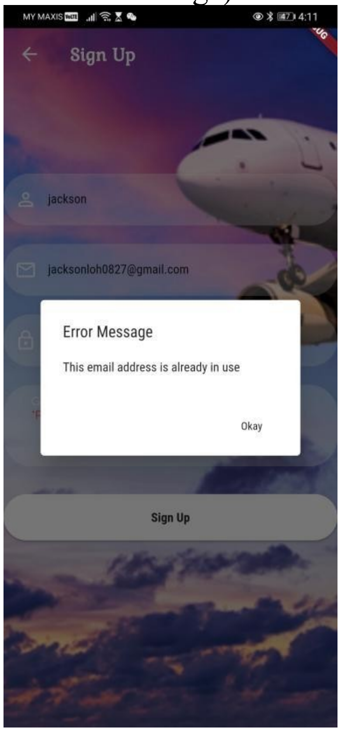


Fig A-7 Sign Up Screen(With Error Message)



Error Message

This email address is already in use

Okay

Fig A-8 Sign Up an Email already Exist

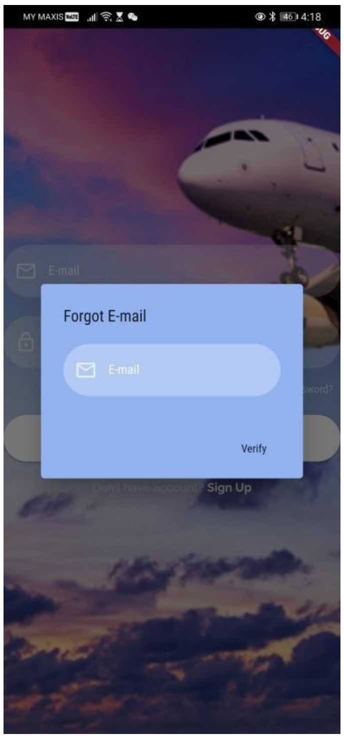
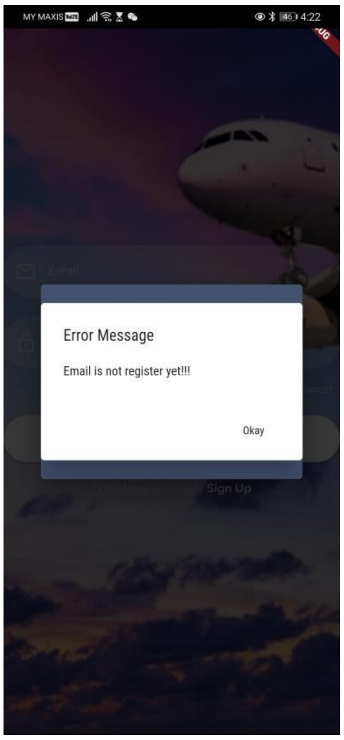


Fig A-9 Forgot Password Dialog Box Entry Form


Error Message

Email is not register yet!!!

Fig A-10 Find the Password of an Email which is not register yet

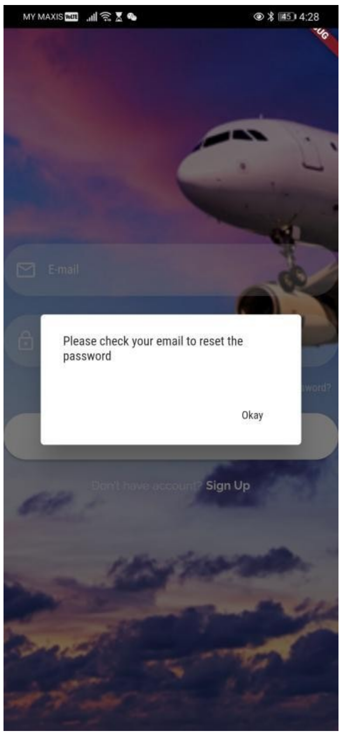

Please check your email to reset the password

Fig A-11 Success Message after Input Register Email in Forgot Password Dialog Box

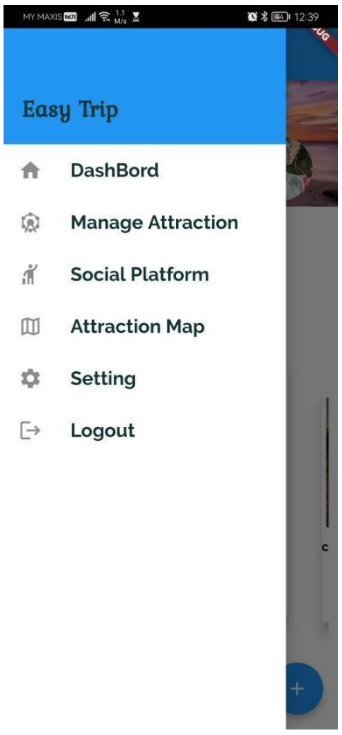


Fig A-12 Drawer on home screen

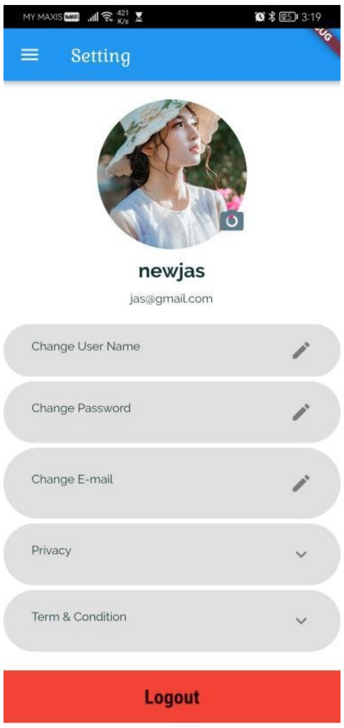


Fig A-13 Setting Screen

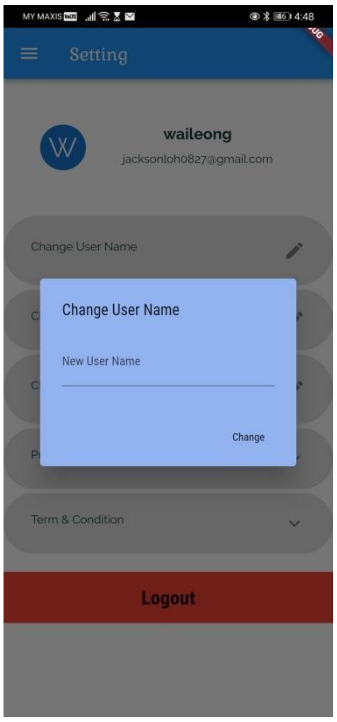


Fig A-14 Change User Name Dialog Box Form

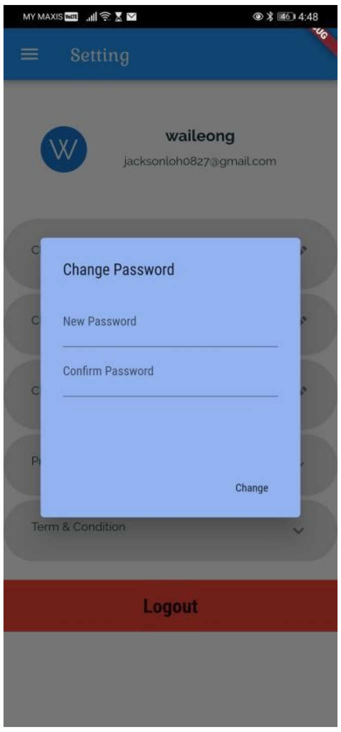


Fig A-15 Change Password Dialog Box Form

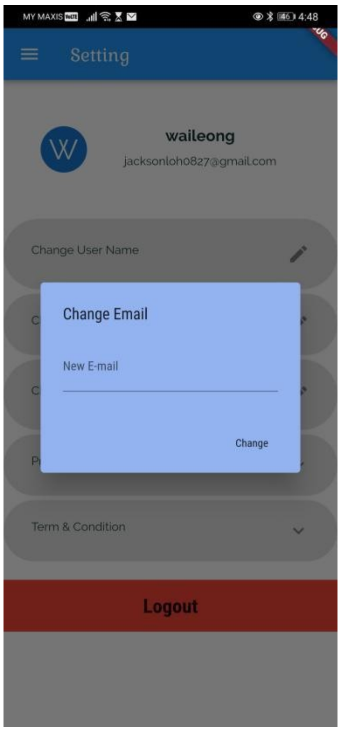


Fig A-16 Change Email Dialog Box Form

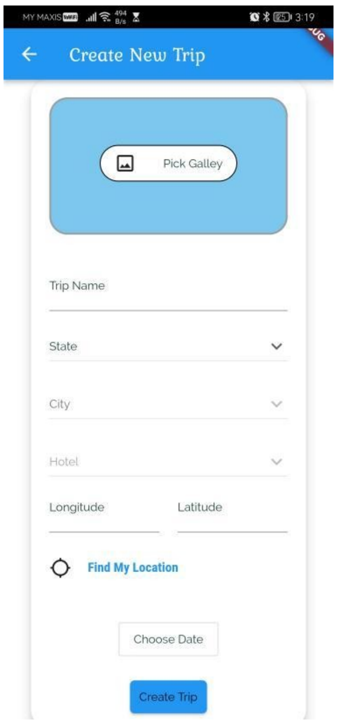


Fig A-17 Create New Trip Form (No Error Message)

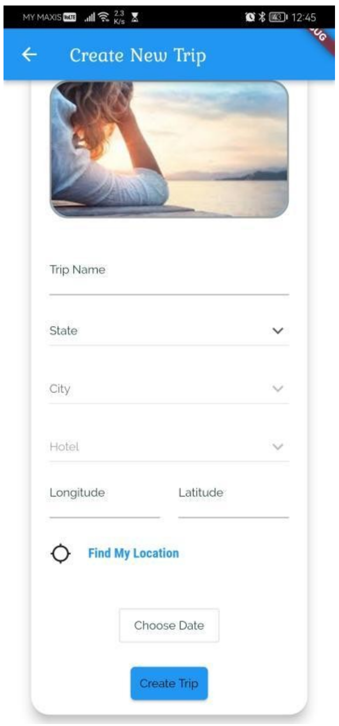


Fig A-18 Create New Trip Form (When user select an image from storage)

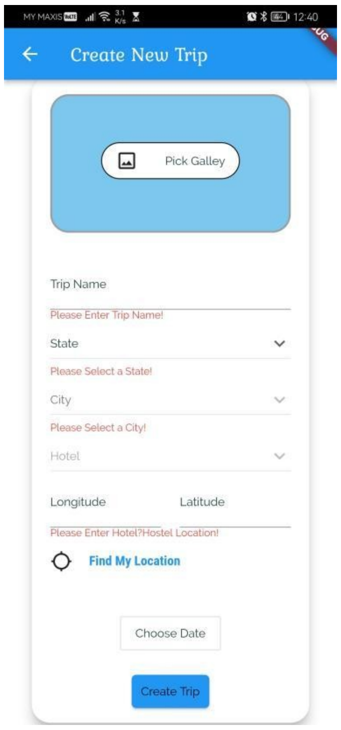


Fig A-19 Create New Trip Form (With Error Message)

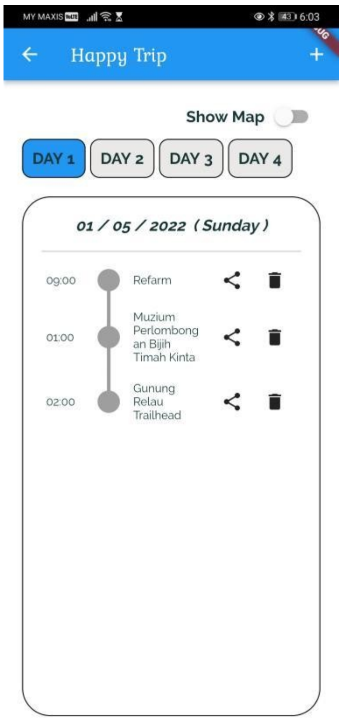


Fig A-20 "Trip Detail" Screen

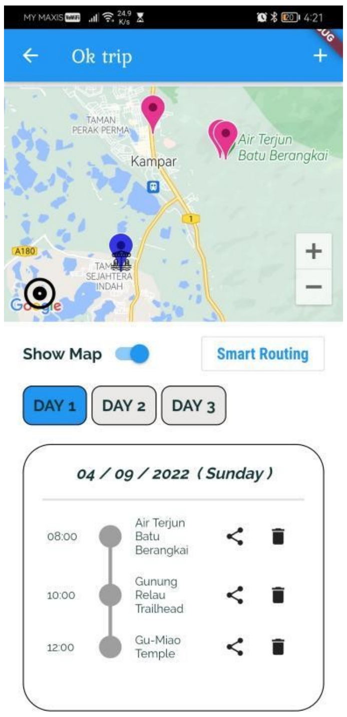

Ok trip
Air Terjun Batu Berangkai
Kampar
TAMAN PERAK PERMAI
TAMAN SEJAHTERA INDAH
Show Map
Smart Routing
DAY 1
DAY 2
DAY 3
04 / 09 / 2022 (Sunday)
08:00 Air Terjun Batu Berangkai
10:00 Gunung Relau Trailhead
12:00 Gu-Miao Temple

Fig A-21 "Trip Detail" Screen (when map is toggle)

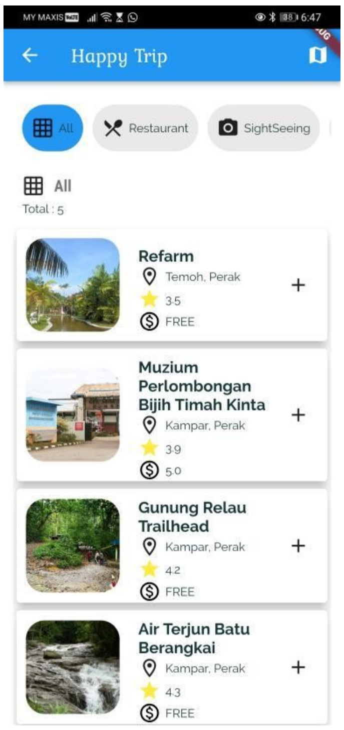


Fig A-22 "Attraction List" Screen

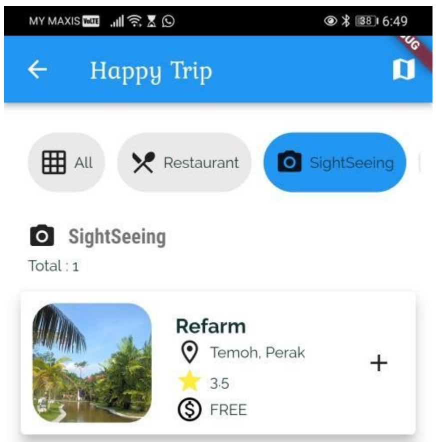


Fig A-23 "Attraction List" Screen (after filter)

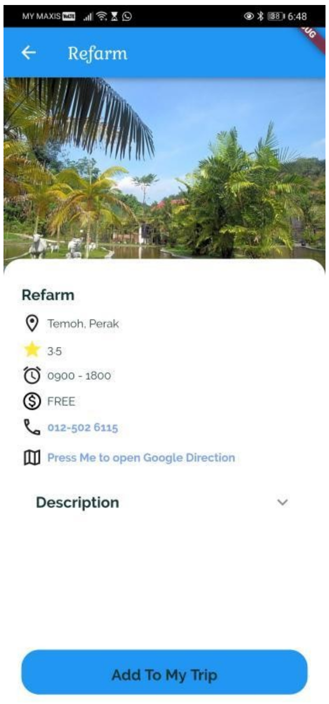
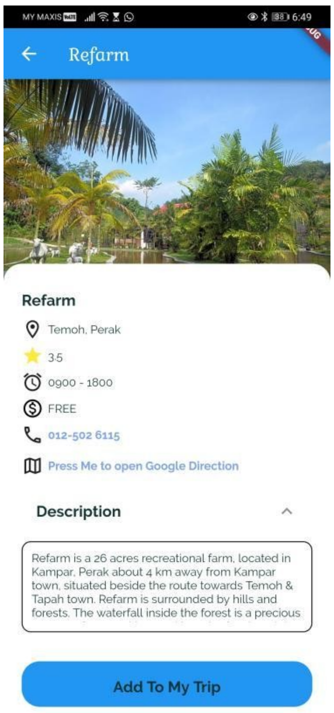


Fig A-24 "Attraction Detail" screen



Refarm is a 26 acres recreational farm, located in Kampar, Perak about 4 km away from Kampar town, situated beside the route towards Temoh & Tapah town. Refarm is surrounded by hills and forests. The waterfall inside the forest is a precious

Fig A-25 "Attraction Detail" screen (After expanded the description)

Fig A-26 "Manage Attraction" screen (without any attraction)

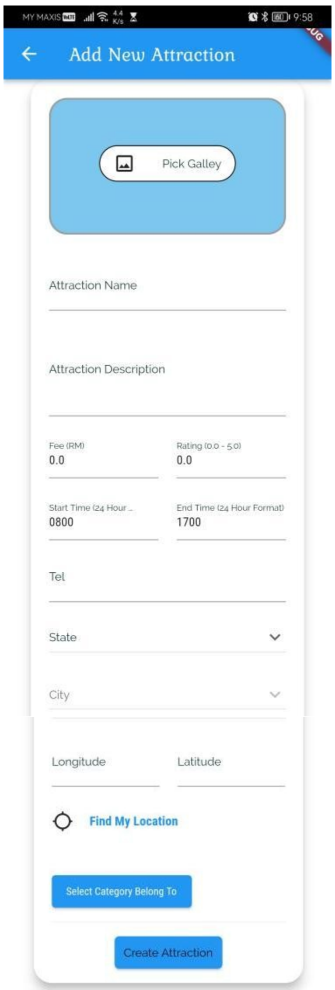
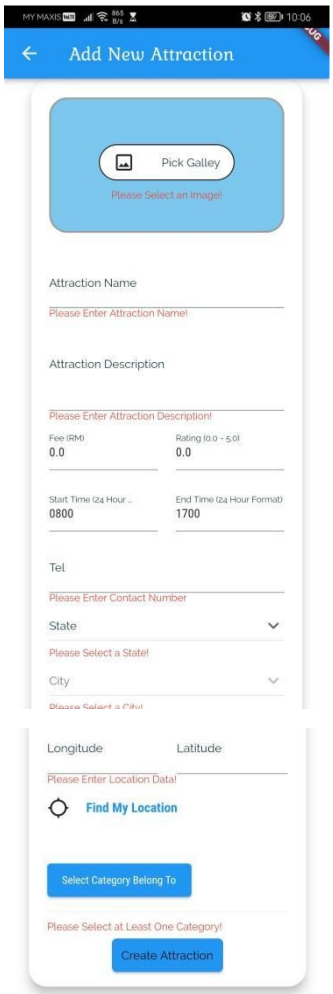
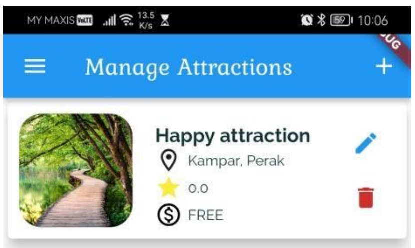


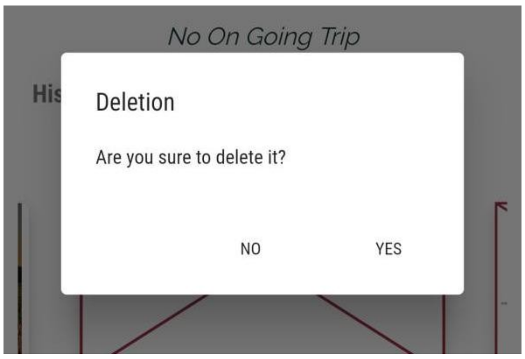
Fig A-27 “Create Attraction” screen (without error message)



Please Select an Image!
Please Enter Attraction Name!
Please Enter Attraction Description!
Please Enter Contact Number
Please Select a State!
Please Select a City!
Please Enter Location Data!
Please Select at Least One Category!

Fig A-28 “Create Attraction” screen (with error message)





Deletion

Are you sure to delete it?

Fig A-29 “Manage Attraction” screen (with attraction)

Fig A-30 Confirmation Dialog Box

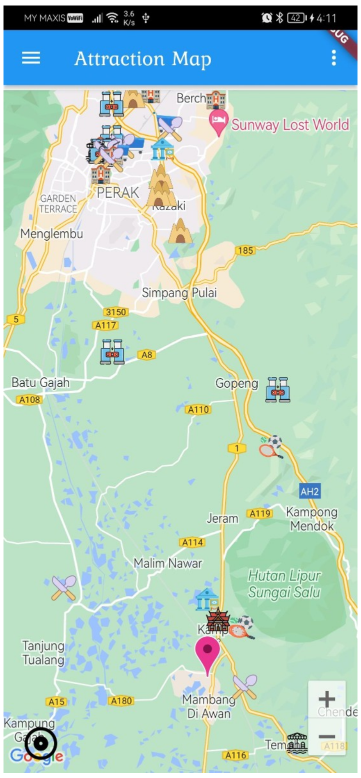


Fig A-31 “Attraction Map” screen (without filter is applied)

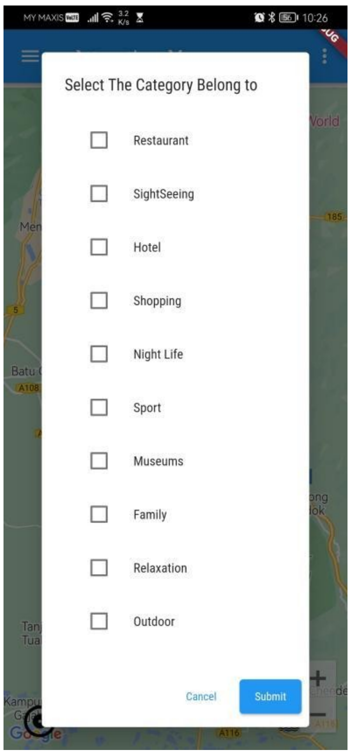


Fig A-32 Filter in “Attraction Map” screen

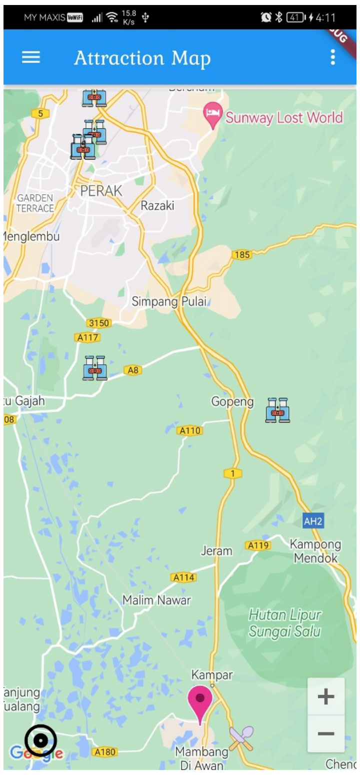


Fig A-33 “Attraction Map” screen (with filter is applied)

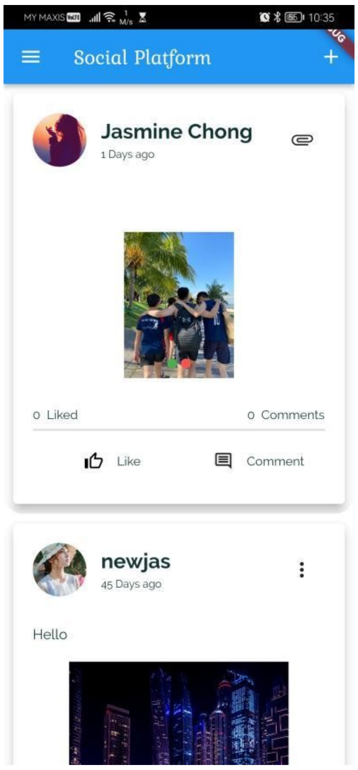
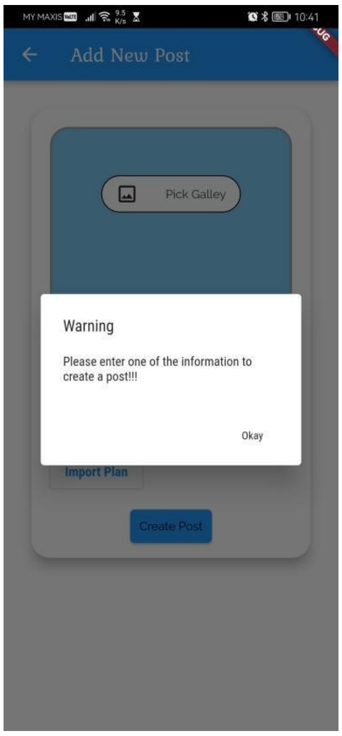


Fig A-34 “Social Platform” screen

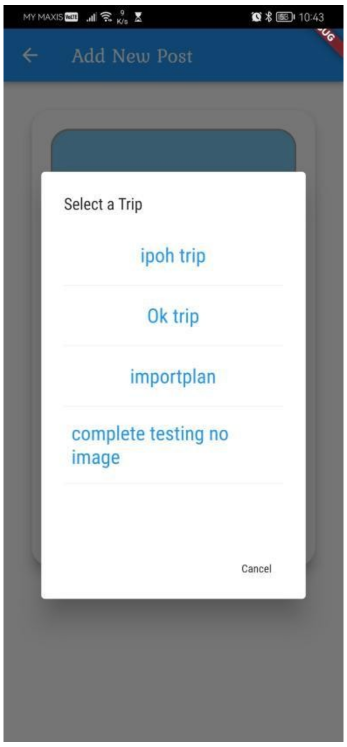


Warning

Please enter one of the information to create a post!!!

Okay

Fig A-35 Error Message during Post Creation



Select a Trip

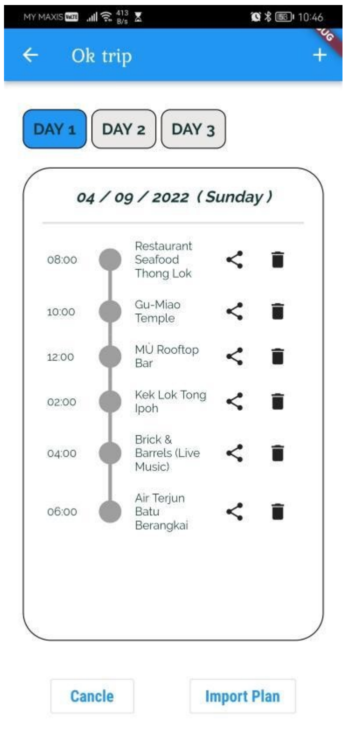
ipoh trip

Ok trip

importplan

complete testing no image

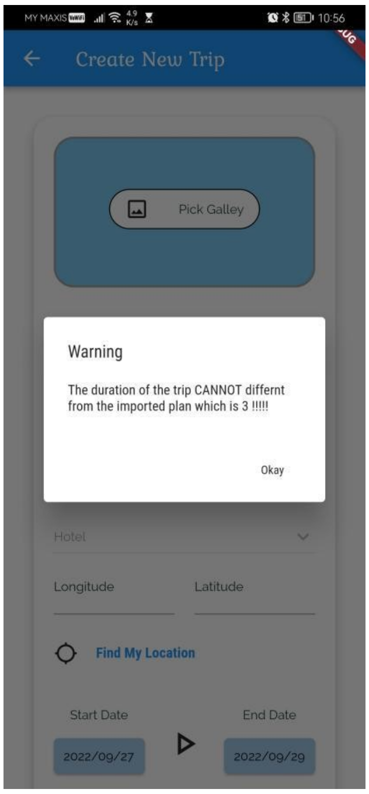
Fig A-36 Itinerary Plan selection when Creating Post



04 / 09 / 2022 (Sunday)

08:00	Restaurant Seafood Thong Lok
10:00	Gu-Miao Temple
12:00	MÚ Rooftop Bar
02:00	Kek Lok Tong Ipoh
04:00	Brick & Barrels (Live Music)
06:00	Air Terjun Batu Berangkai

Fig A-37 Itinerary Plan Preview



Warning

The duration of the trip CANNOT differnt from the imported plan which is 3 !!!!!

Okay

Fig A-38 Error message during importing the itinerary Plan

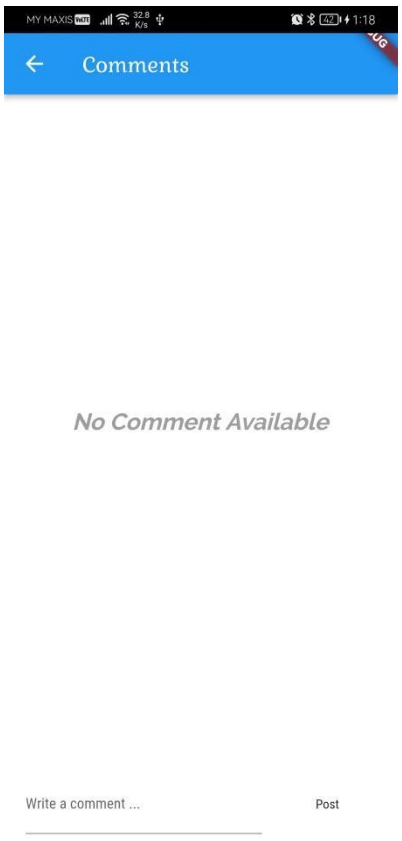


Fig A-39 Comment screen (without any Comment)

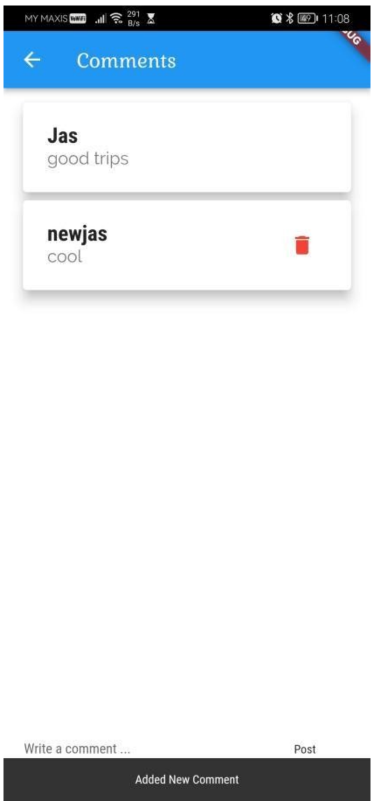
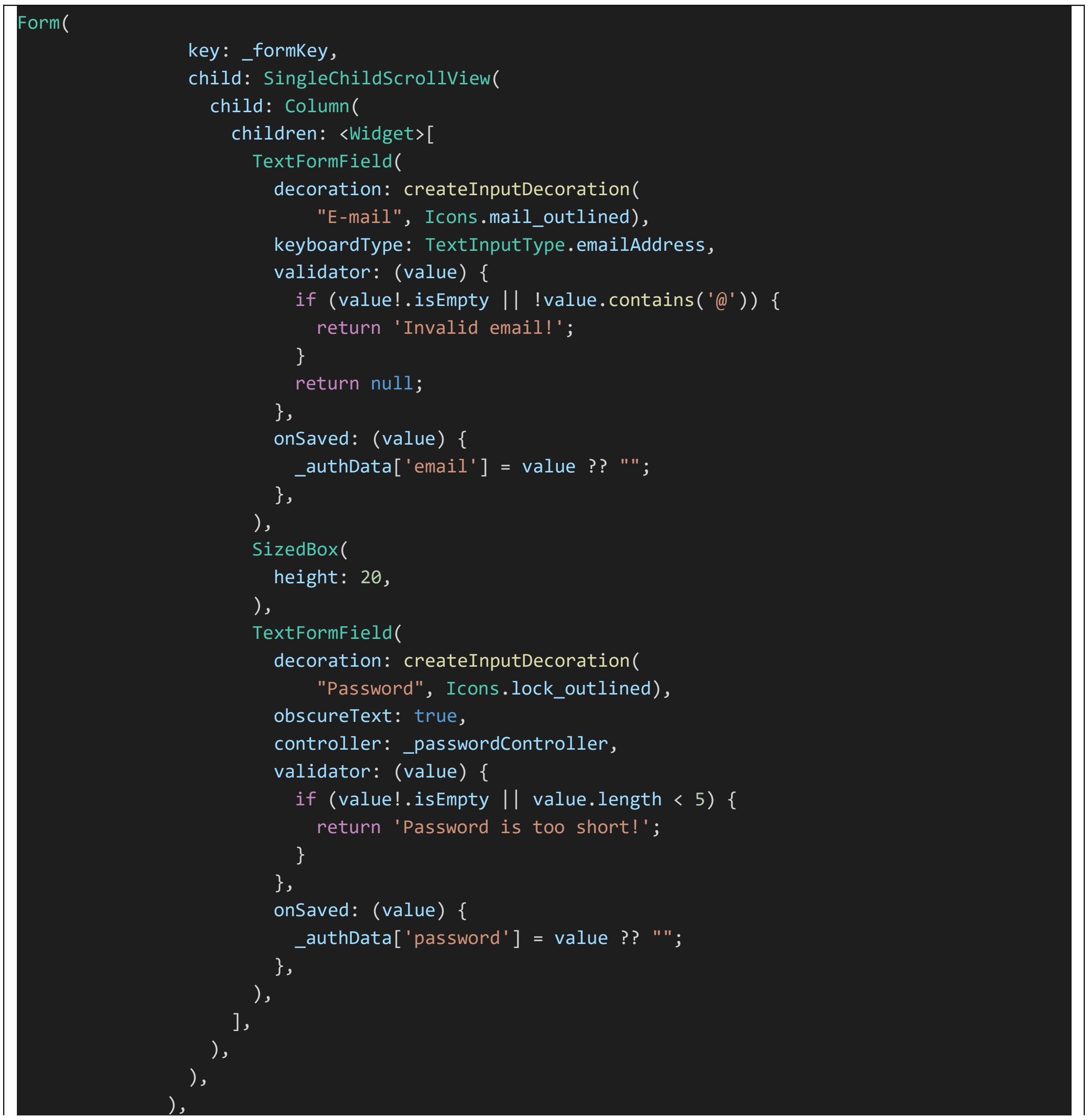


Fig A-40 Comment screen (with Comment)

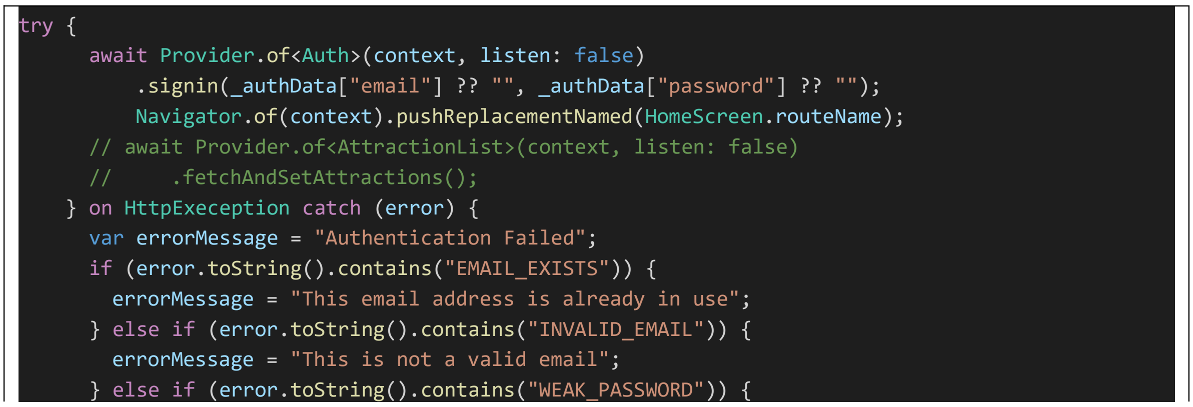
Appendix B

Coding Screenshot

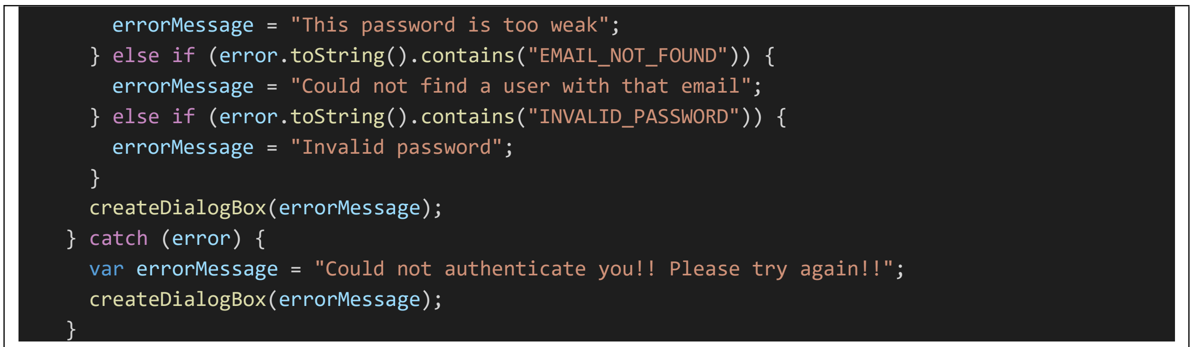


```
Form(
                key: _formKey,
                child: SingleChildScrollView(
                  child: Column(
                    children: <Widget>[
                      TextFormField(
                        decoration: createInputDecoration(
                            "E-mail", Icons.mail_outlined),
                        keyboardType: TextInputType.emailAddress,
                        validator: (value) {
                          if (value!.isEmpty || !value.contains('@')) {
                            return 'Invalid email!';
                          }
                          return null;
                        },
                        onSaved: (value) {
                          _authData['email'] = value ?? "";
                        },
                      ),
                      SizedBox(
                        height: 20,
                      ),
                      TextFormField(
                        decoration: createInputDecoration(
                            "Password", Icons.lock_outlined),
                        obscureText: true,
                        controller: _passwordController,
                        validator: (value) {
                          if (value!.isEmpty || value.length < 5) {
                            return 'Password is too short!';
                          }
                        },
                        onSaved: (value) {
                          _authData['password'] = value ?? "";
                        },
                      ),
                    ],
                  ),
                ),
              ),
```

Fig B-1 Login Entry Form Validation

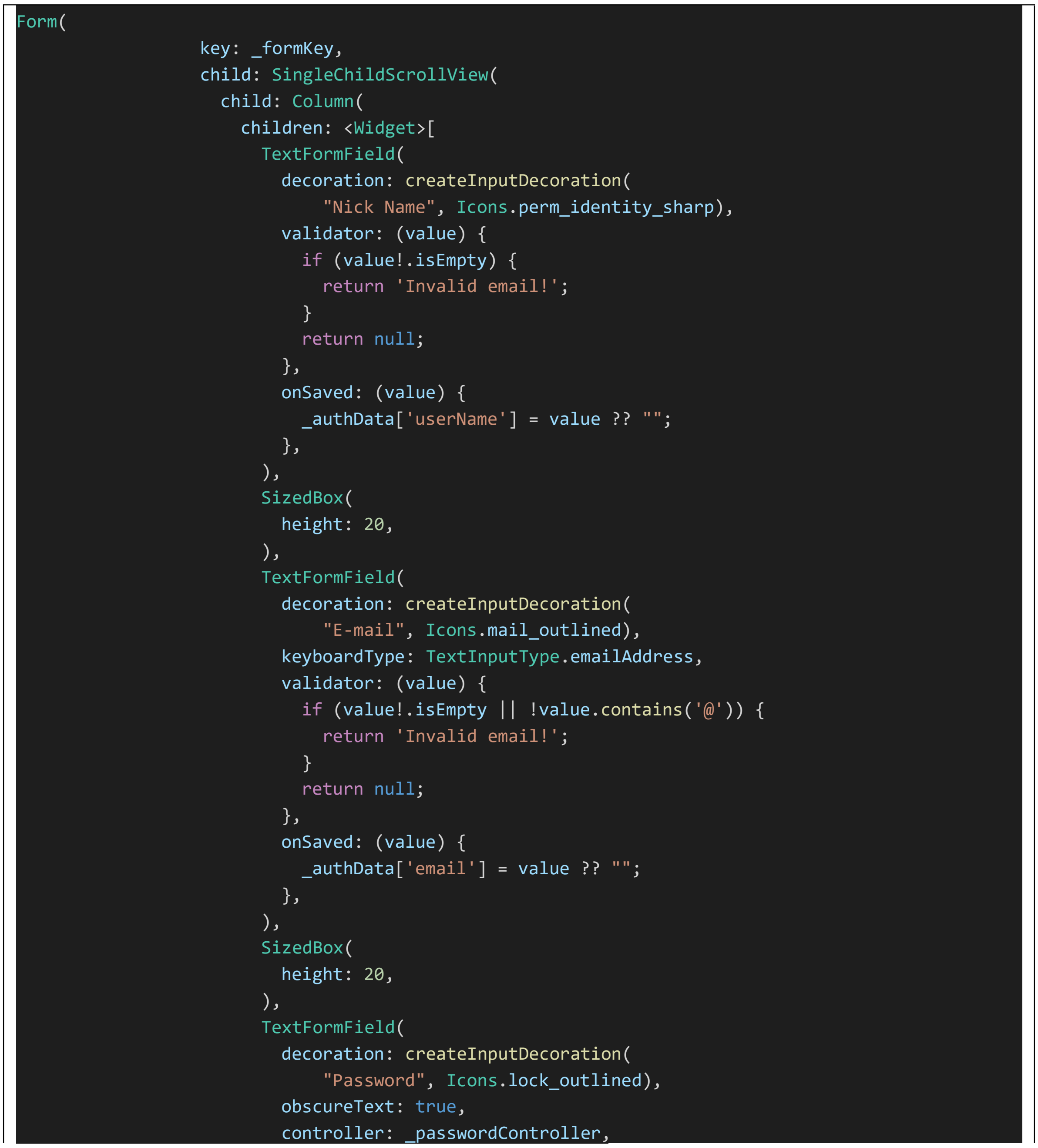


```
try {
      await Provider.of<Auth>(context, listen: false)
          .signin(_authData["email"] ?? "", _authData["password"] ?? "");
          Navigator.of(context).pushReplacementNamed(HomeScreen.routeName);
      // await Provider.of<AttractionList>(context, listen: false)
      //     .fetchAndSetAttractions();
    } on HttpExeception catch (error) {
      var errorMessage = "Authentication Failed";
      if (error.toString().contains("EMAIL_EXISTS")) {
        errorMessage = "This email address is already in use";
      } else if (error.toString().contains("INVALID_EMAIL")) {
        errorMessage = "This is not a valid email";
      } else if (error.toString().contains("WEAK_PASSWORD")) {
```

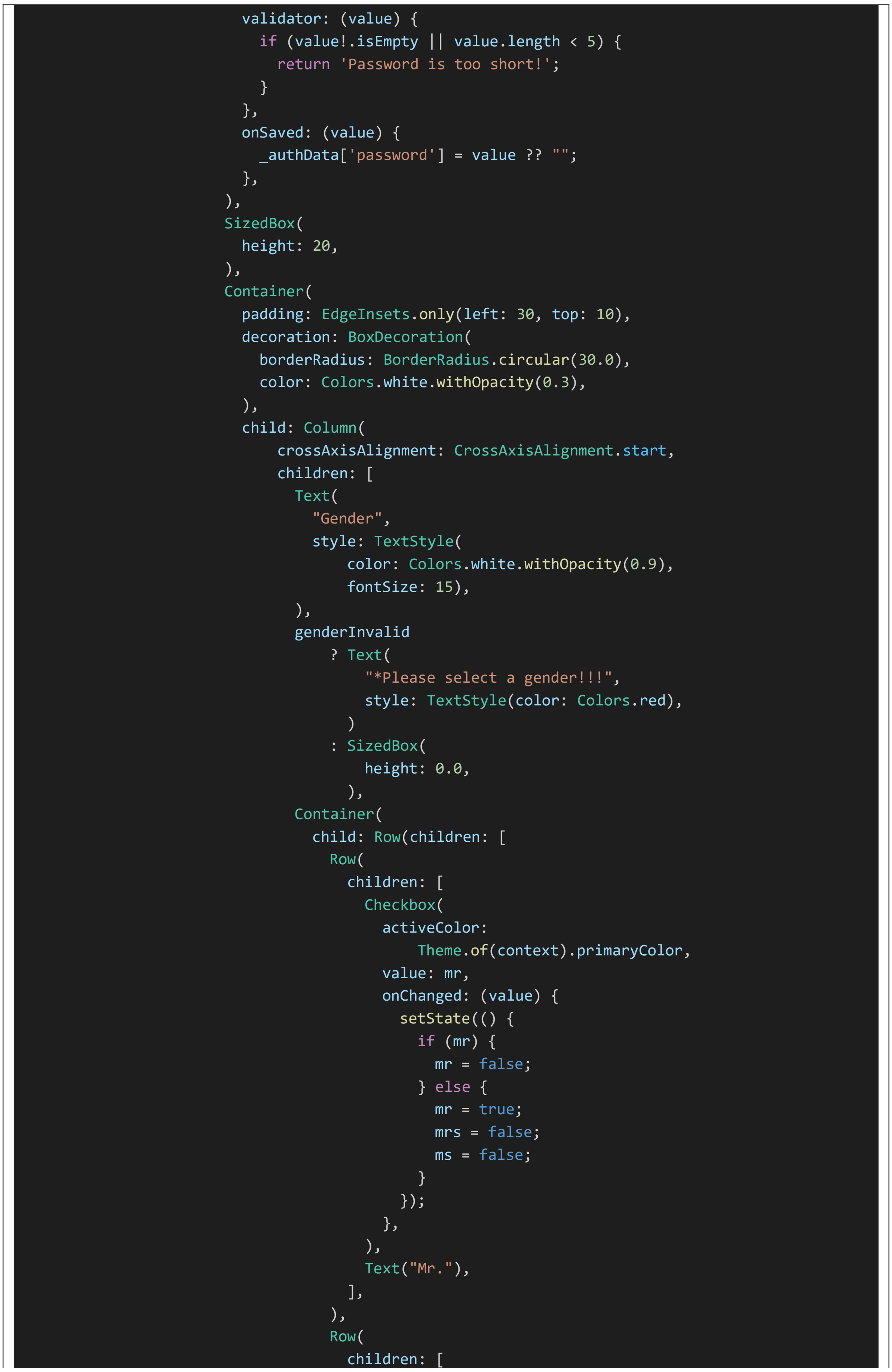


```
        errorMessage = "This password is too weak";
      } else if (error.toString().contains("EMAIL_NOT_FOUND")) {
        errorMessage = "Could not find a user with that email";
      } else if (error.toString().contains("INVALID_PASSWORD")) {
        errorMessage = "Invalid password";
      }
      createDialogBox(errorMessage);
    } catch (error) {
      var errorMessage = "Could not authenticate you!! Please try again!!";
      createDialogBox(errorMessage);
    }
```

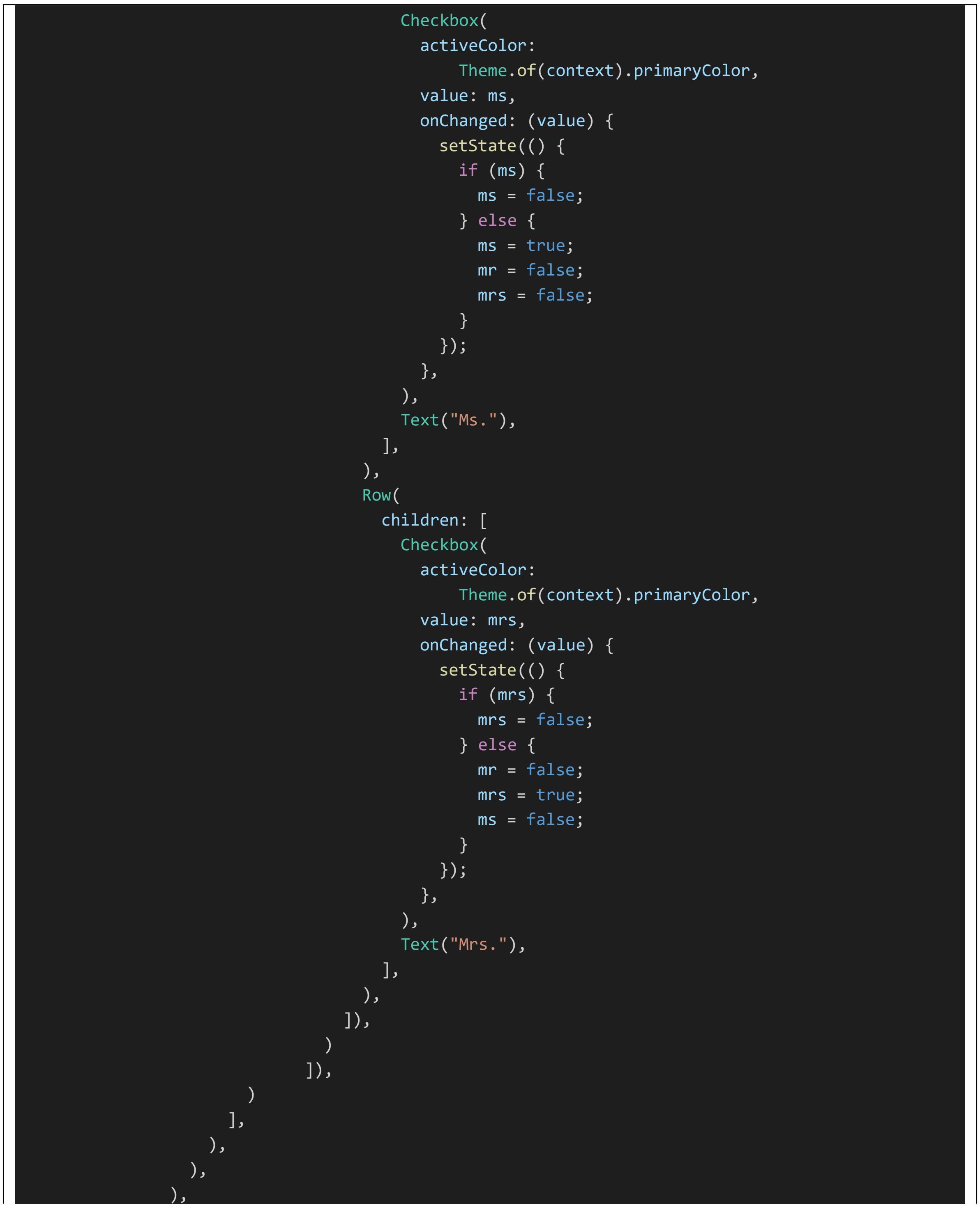
Fig B-2 Google Firebase Authentication Service



```
Form(
                  key: _formKey,
                  child: SingleChildScrollView(
                    child: Column(
                      children: <Widget>[
                        TextFormField(
                          decoration: createInputDecoration(
                              "Nick Name", Icons.perm_identity_sharp),
                          validator: (value) {
                            if (value!.isEmpty) {
                              return 'Invalid email!';
                            }
                            return null;
                          },
                          onSaved: (value) {
                            _authData['userName'] = value ?? "";
                          },
                        ),
                        SizedBox(
                          height: 20,
                        ),
                        TextFormField(
                          decoration: createInputDecoration(
                              "E-mail", Icons.mail_outlined),
                          keyboardType: TextInputType.emailAddress,
                          validator: (value) {
                            if (value!.isEmpty || !value.contains('@')) {
                              return 'Invalid email!';
                            }
                            return null;
                          },
                          onSaved: (value) {
                            _authData['email'] = value ?? "";
                          },
                        ),
                        SizedBox(
                          height: 20,
                        ),
                        TextFormField(
                          decoration: createInputDecoration(
                              "Password", Icons.lock_outlined),
                          obscureText: true,
                          controller: _passwordController,
```

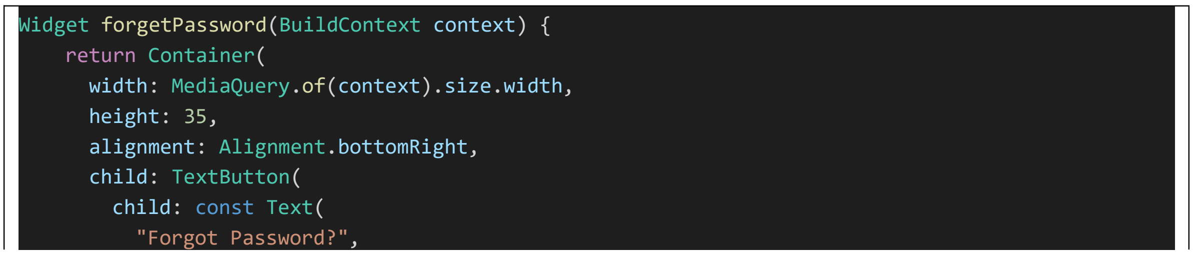


```
                          validator: (value) {
                            if (value!.isEmpty || value.length < 5) {
                              return 'Password is too short!';
                            }
                          },
                          onSaved: (value) {
                            _authData['password'] = value ?? "";
                          },
                        ),
                        SizedBox(
                          height: 20,
                        ),
                        Container(
                          padding: EdgeInsets.only(left: 30, top: 10),
                          decoration: BoxDecoration(
                            borderRadius: BorderRadius.circular(30.0),
                            color: Colors.white.withOpacity(0.3),
                          ),
                          child: Column(
                              crossAxisAlignment: CrossAxisAlignment.start,
                              children: [
                                Text(
                                  "Gender",
                                  style: TextStyle(
                                      color: Colors.white.withOpacity(0.9),
                                      fontSize: 15),
                                ),
                                genderInvalid
                                    ? Text(
                                        "*Please select a gender!!!",
                                        style: TextStyle(color: Colors.red),
                                      )
                                    : SizedBox(
                                        height: 0.0,
                                      ),
                                Container(
                                  child: Row(children: [
                                    Row(
                                      children: [
                                        Checkbox(
                                          activeColor:
                                              Theme.of(context).primaryColor,
                                          value: mr,
                                          onChanged: (value) {
                                            setState(() {
                                              if (mr) {
                                                mr = false;
                                              } else {
                                                mr = true;
                                                mrs = false;
                                                ms = false;
                                              }
                                            });
                                          },
                                        ),
                                        Text("Mr."),
                                      ],
                                    ),
                                    Row(
                                      children: [
```

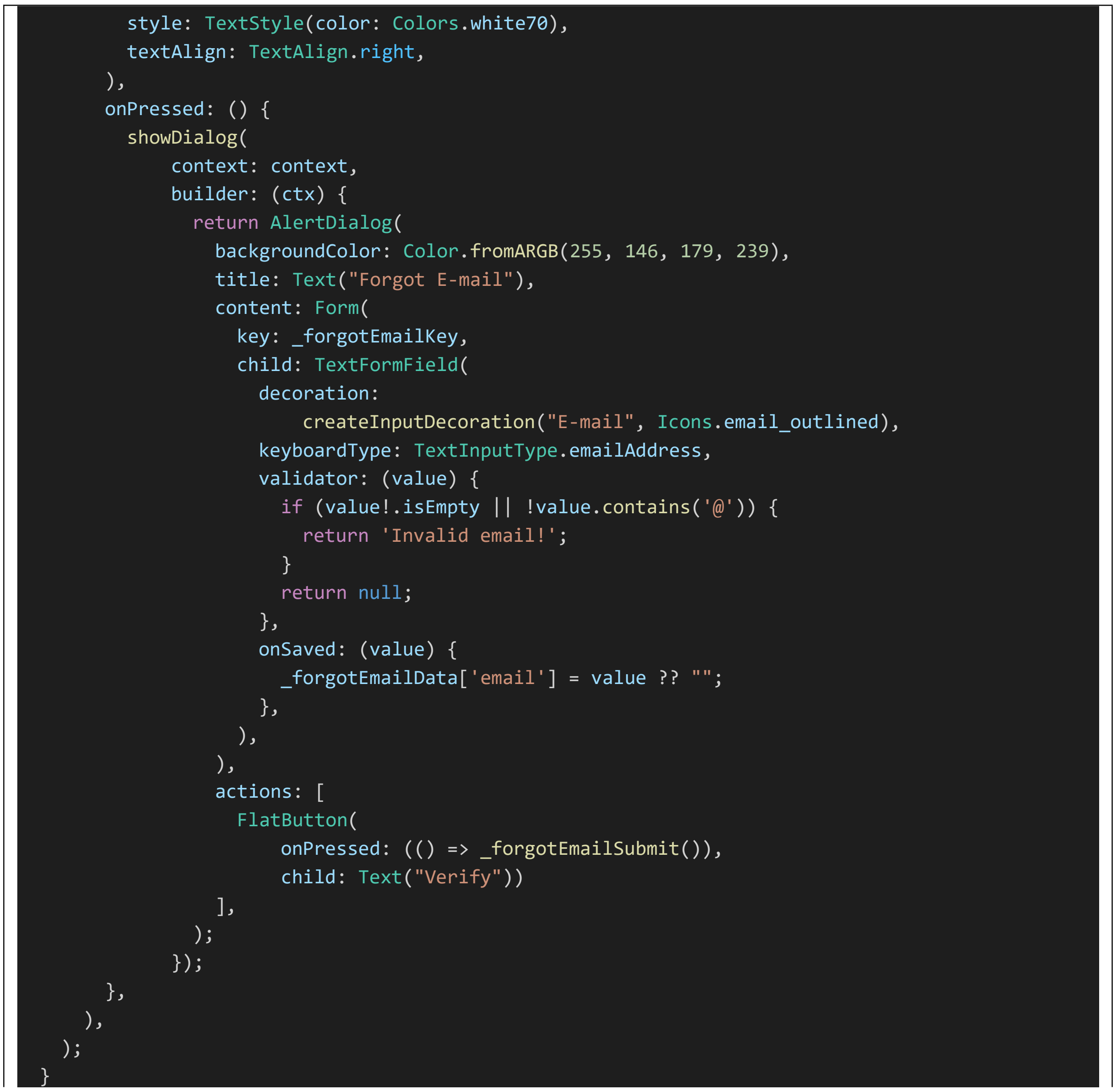


```
                                        Checkbox(
                                          activeColor:
                                              Theme.of(context).primaryColor,
                                          value: ms,
                                          onChanged: (value) {
                                            setState(() {
                                              if (ms) {
                                                ms = false;
                                              } else {
                                                ms = true;
                                                mr = false;
                                                mrs = false;
                                              }
                                            });
                                          },
                                        ),
                                        Text("Ms."),
                                      ],
                                    ),
                                    Row(
                                      children: [
                                        Checkbox(
                                          activeColor:
                                              Theme.of(context).primaryColor,
                                          value: mrs,
                                          onChanged: (value) {
                                            setState(() {
                                              if (mrs) {
                                                mrs = false;
                                              } else {
                                                mr = false;
                                                mrs = true;
                                                ms = false;
                                              }
                                            });
                                          },
                                        ),
                                        Text("Mrs."),
                                      ],
                                    ),
                                  ]),
                                )
                              ]),
                        )
                      ],
                    ),
                  ),
                ),
```

Fig B-3 Sign Up Form with Validation

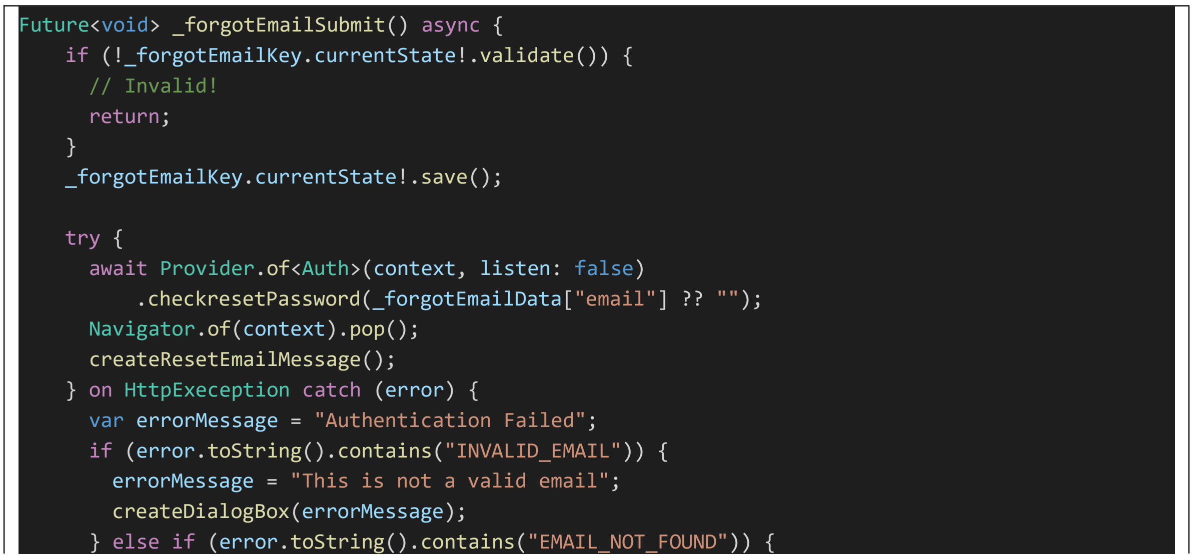


```
Widget forgetPassword(BuildContext context) {
    return Container(
      width: MediaQuery.of(context).size.width,
      height: 35,
      alignment: Alignment.bottomRight,
      child: TextButton(
        child: const Text(
          "Forgot Password?",
```

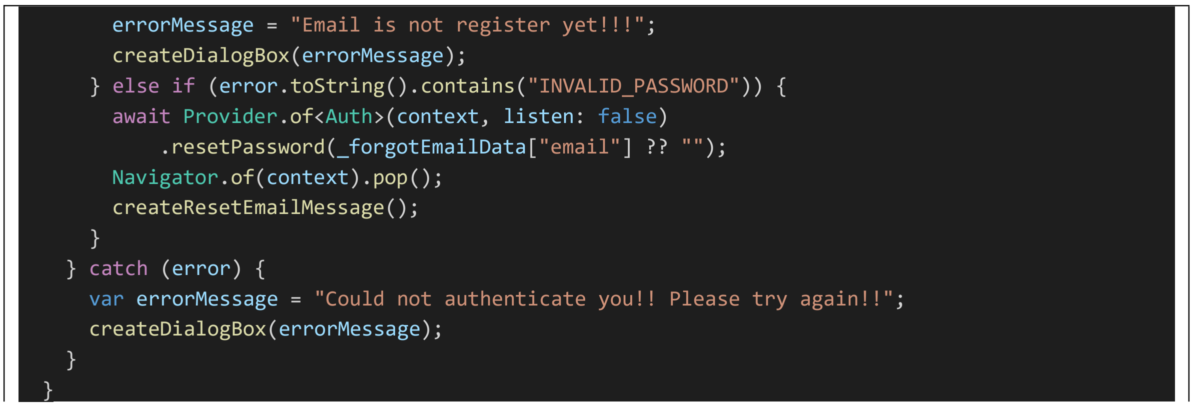


```
          style: TextStyle(color: Colors.white70),
          textAlign: TextAlign.right,
        ),
        onPressed: () {
          showDialog(
              context: context,
              builder: (ctx) {
                return AlertDialog(
                  backgroundColor: Color.fromARGB(255, 146, 179, 239),
                  title: Text("Forgot E-mail"),
                  content: Form(
                    key: _forgotEmailKey,
                    child: TextFormField(
                      decoration:
                          createInputDecoration("E-mail", Icons.email_outlined),
                      keyboardType: TextInputType.emailAddress,
                      validator: (value) {
                        if (value!.isEmpty || !value.contains('@')) {
                          return 'Invalid email!';
                        }
                        return null;
                      },
                      onSaved: (value) {
                        _forgotEmailData['email'] = value ?? "";
                      },
                    ),
                  ),
                  actions: [
                    FlatButton(
                        onPressed: (() => _forgotEmailSubmit()),
                        child: Text("Verify"))
                  ],
                );
              });
        },
      ),
    );
  }
```

Fig B-4 Forgot Password Dialog Box Form

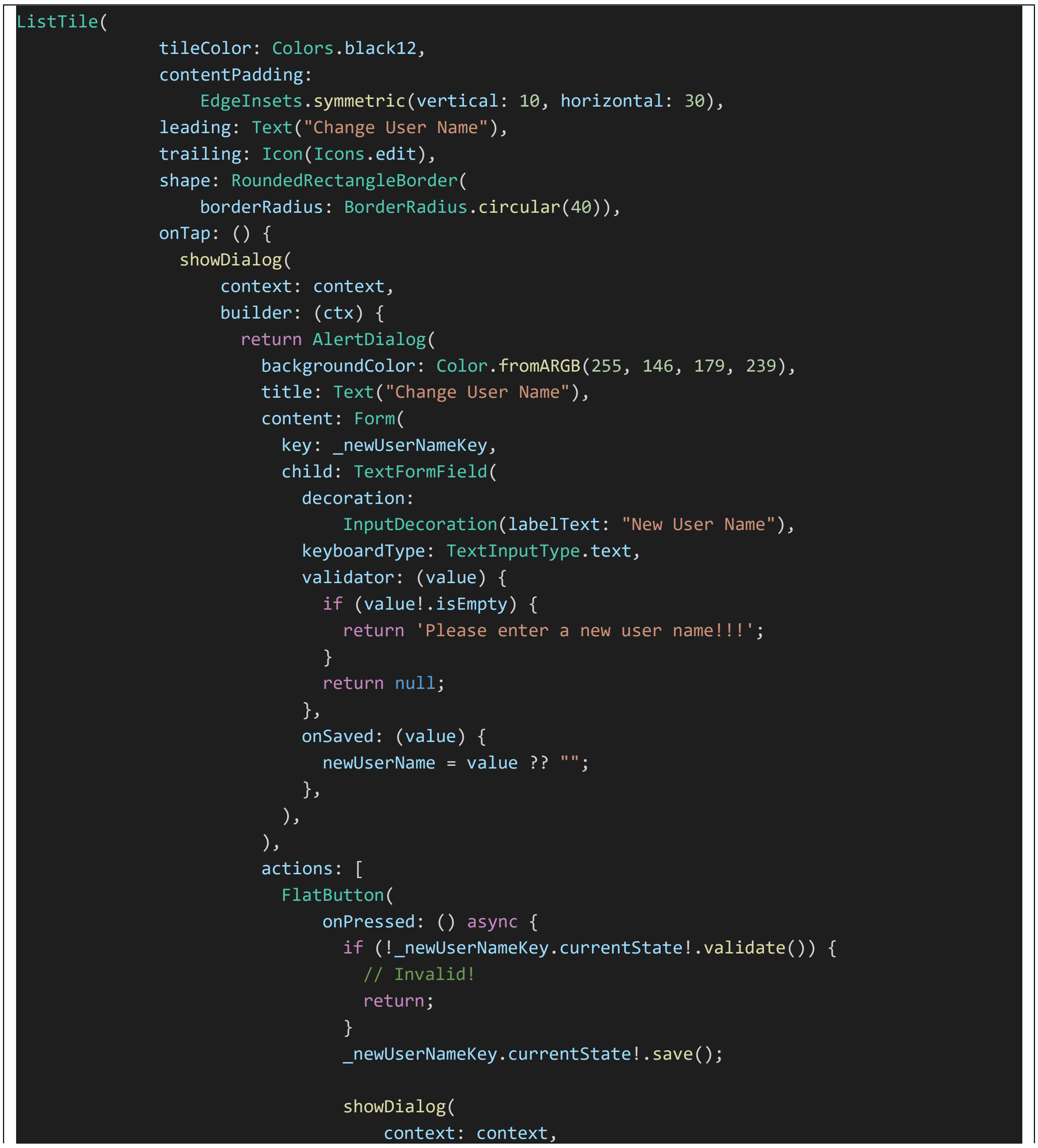


```
Future<void> _forgotEmailSubmit() async {
    if (!_forgotEmailKey.currentState!.validate()) {
      // Invalid!
      return;
    }
    _forgotEmailKey.currentState!.save();

    try {
      await Provider.of<Auth>(context, listen: false)
          .checkresetPassword(_forgotEmailData["email"] ?? "");
      Navigator.of(context).pop();
      createResetEmailMessage();
    } on HttpException catch (error) {
      var errorMessage = "Authentication Failed";
      if (error.toString().contains("INVALID_EMAIL")) {
        errorMessage = "This is not a valid email";
        createDialogBox(errorMessage);
      } else if (error.toString().contains("EMAIL_NOT_FOUND")) {
```

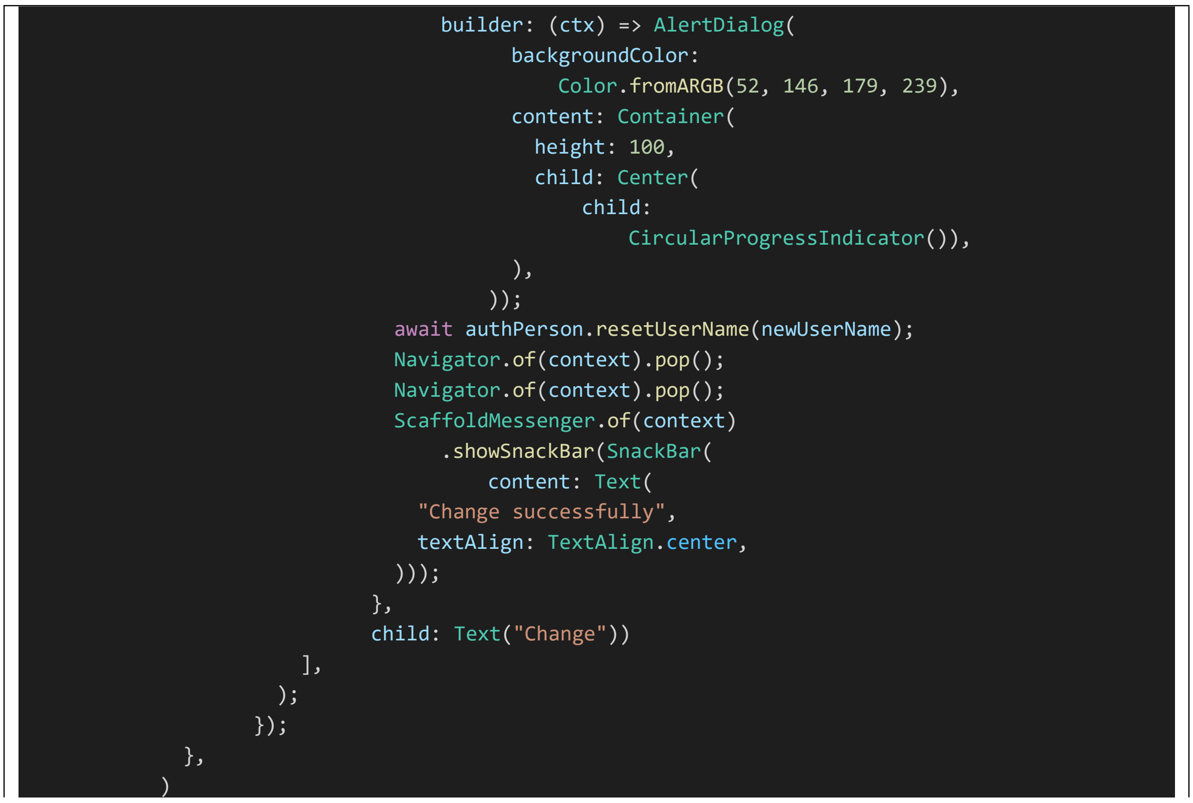


```
        errorMessage = "Email is not register yet!!!";
        createDialogBox(errorMessage);
      } else if (error.toString().contains("INVALID_PASSWORD")) {
        await Provider.of<Auth>(context, listen: false)
            .resetPassword(_forgotEmailData["email"] ?? "");
        Navigator.of(context).pop();
        createResetEmailMessage();
      }
    } catch (error) {
      var errorMessage = "Could not authenticate you!! Please try again!!";
      createDialogBox(errorMessage);
    }
  }
```

Fig B-5 Google Firebase Validate the Email Provided

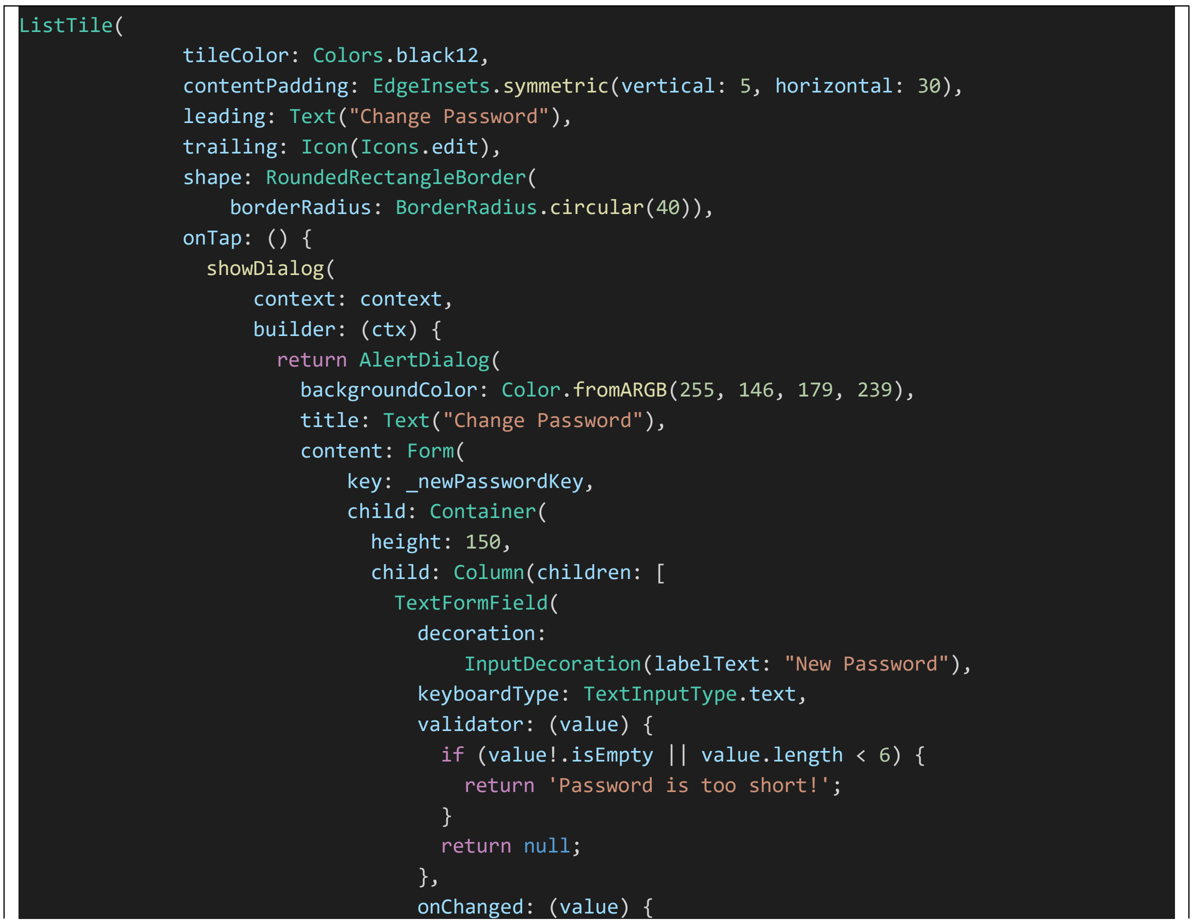


```
ListTile(
              tileColor: Colors.black12,
              contentPadding:
                  EdgeInsets.symmetric(vertical: 10, horizontal: 30),
              leading: Text("Change User Name"),
              trailing: Icon(Icons.edit),
              shape: RoundedRectangleBorder(
                  borderRadius: BorderRadius.circular(40)),
              onTap: () {
                showDialog(
                    context: context,
                    builder: (ctx) {
                      return AlertDialog(
                        backgroundColor: Color.fromARGB(255, 146, 179, 239),
                        title: Text("Change User Name"),
                        content: Form(
                          key: _newUserNameKey,
                          child: TextFormField(
                            decoration:
                                InputDecoration(labelText: "New User Name"),
                            keyboardType: TextInputType.text,
                            validator: (value) {
                              if (value!.isEmpty) {
                                return 'Please enter a new user name!!!';
                              }
                              return null;
                            },
                            onSaved: (value) {
                              newUserName = value ?? "";
                            },
                          ),
                        ),
                        actions: [
                          FlatButton(
                              onPressed: () async {
                                if (!_newUserNameKey.currentState!.validate()) {
                                  // Invalid!
                                  return;
                                }
                                _newUserNameKey.currentState!.save();

                                showDialog(
                                    context: context,
```

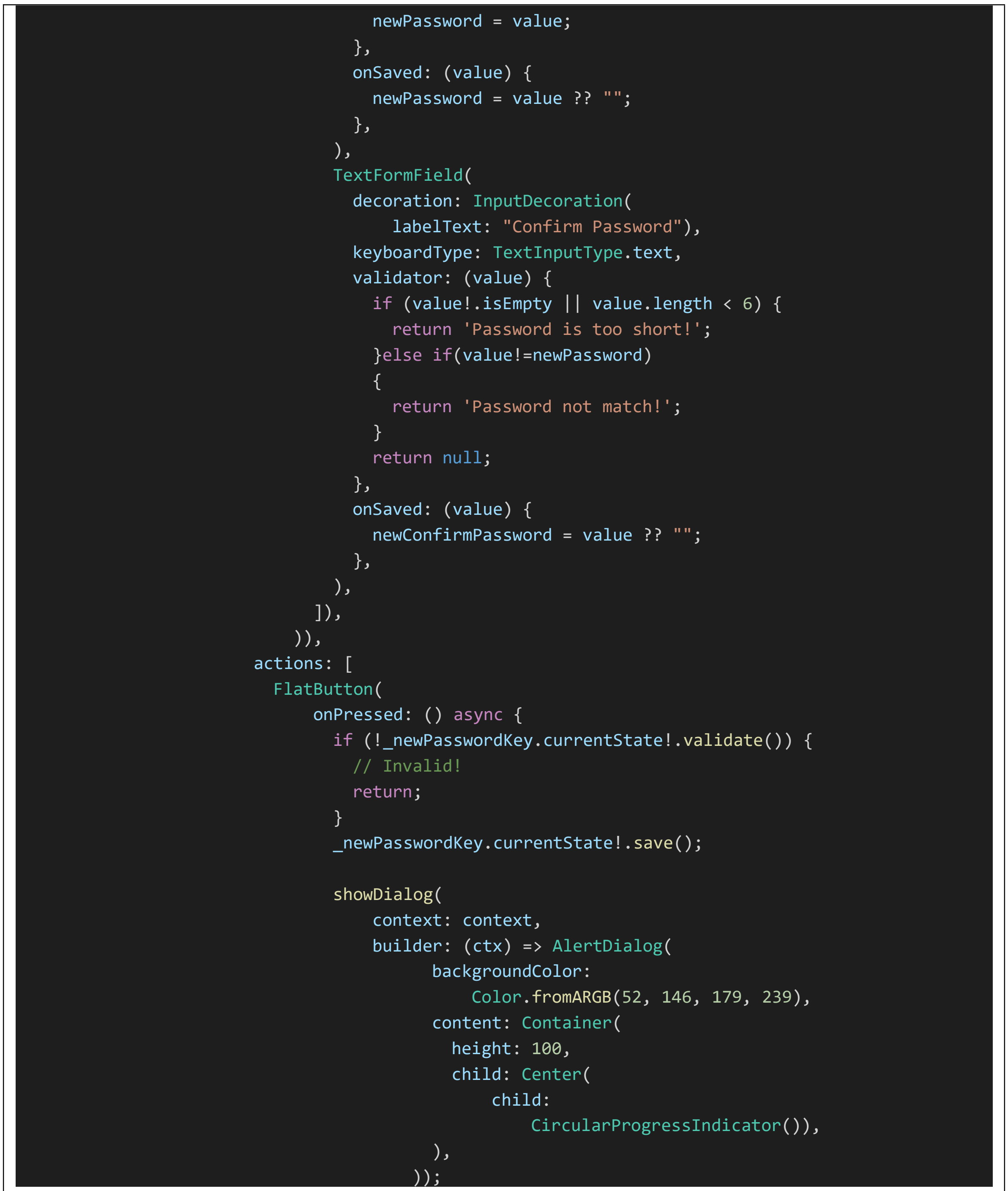


```
                                    builder: (ctx) => AlertDialog(
                                          backgroundColor:
                                              Color.fromARGB(52, 146, 179, 239),
                                          content: Container(
                                            height: 100,
                                            child: Center(
                                                child:
                                                    CircularProgressIndicator()),
                                          ),
                                        ));
                                await authPerson.resetUserName(newUserName);
                                Navigator.of(context).pop();
                                Navigator.of(context).pop();
                                ScaffoldMessenger.of(context)
                                    .showSnackBar(SnackBar(
                                        content: Text(
                                  "Change successfully",
                                  textAlign: TextAlign.center,
                                )));
                              },
                              child: Text("Change"))
                        ],
                      );
                    });
              },
            )
```

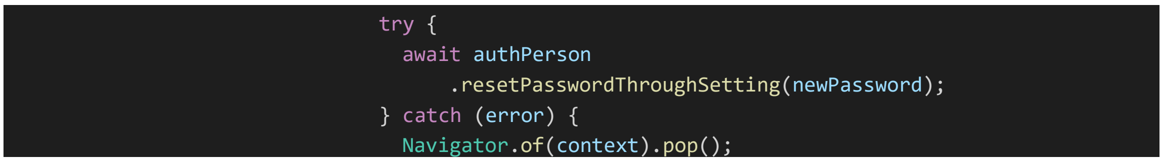
Fig B-6 Change User Name Dialog Box Form



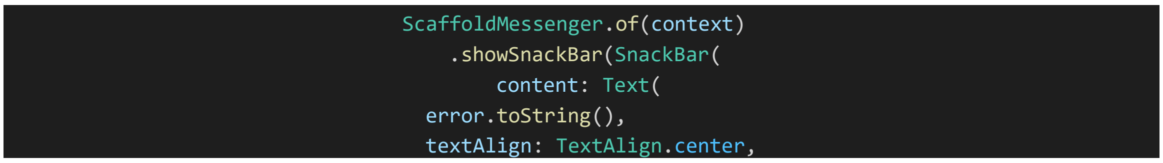
```
ListTile(
              tileColor: Colors.black12,
              contentPadding: EdgeInsets.symmetric(vertical: 5, horizontal: 30),
              leading: Text("Change Password"),
              trailing: Icon(Icons.edit),
              shape: RoundedRectangleBorder(
                  borderRadius: BorderRadius.circular(40)),
              onTap: () {
                showDialog(
                    context: context,
                    builder: (ctx) {
                      return AlertDialog(
                        backgroundColor: Color.fromARGB(255, 146, 179, 239),
                        title: Text("Change Password"),
                        content: Form(
                            key: _newPasswordKey,
                            child: Container(
                              height: 150,
                              child: Column(children: [
                                TextFormField(
                                  decoration:
                                      InputDecoration(labelText: "New Password"),
                                  keyboardType: TextInputType.text,
                                  validator: (value) {
                                    if (value!.isEmpty || value.length < 6) {
                                      return 'Password is too short!';
                                    }
                                    return null;
                                  },
                                  onChanged: (value) {
```



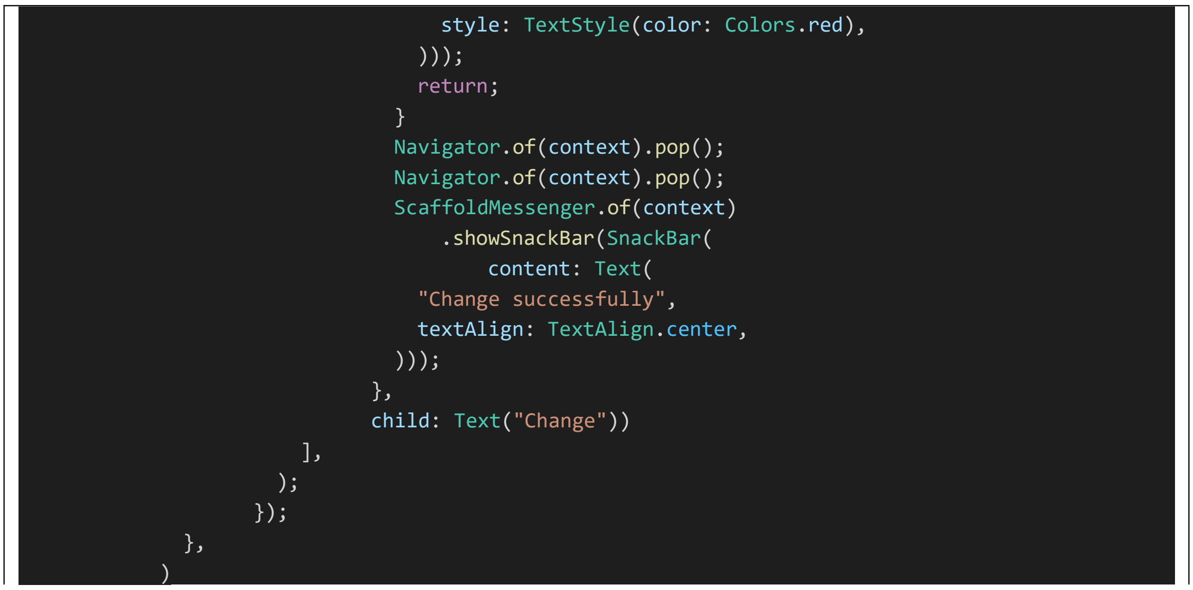
```
            newPassword = value;
          },
          onSaved: (value) {
            newPassword = value ?? "";
          },
        ),
        TextFormField(
          decoration: InputDecoration(
              labelText: "Confirm Password"),
          keyboardType: TextInputType.text,
          validator: (value) {
            if (value!.isEmpty || value.length < 6) {
              return 'Password is too short!';
            }else if(value!=newPassword)
            {
              return 'Password not match!';
            }
            return null;
          },
          onSaved: (value) {
            newConfirmPassword = value ?? "";
          },
        ),
      ]),
    )),
actions: [
  FlatButton(
      onPressed: () async {
        if (!_newPasswordKey.currentState!.validate()) {
          // Invalid!
          return;
        }
        _newPasswordKey.currentState!.save();

        showDialog(
            context: context,
            builder: (ctx) => AlertDialog(
                  backgroundColor:
                      Color.fromARGB(52, 146, 179, 239),
                  content: Container(
                    height: 100,
                    child: Center(
                        child:
                            CircularProgressIndicator()),
                  ),
                ));
```



```
        try {
          await authPerson
              .resetPasswordThroughSetting(newPassword);
        } catch (error) {
          Navigator.of(context).pop();
```

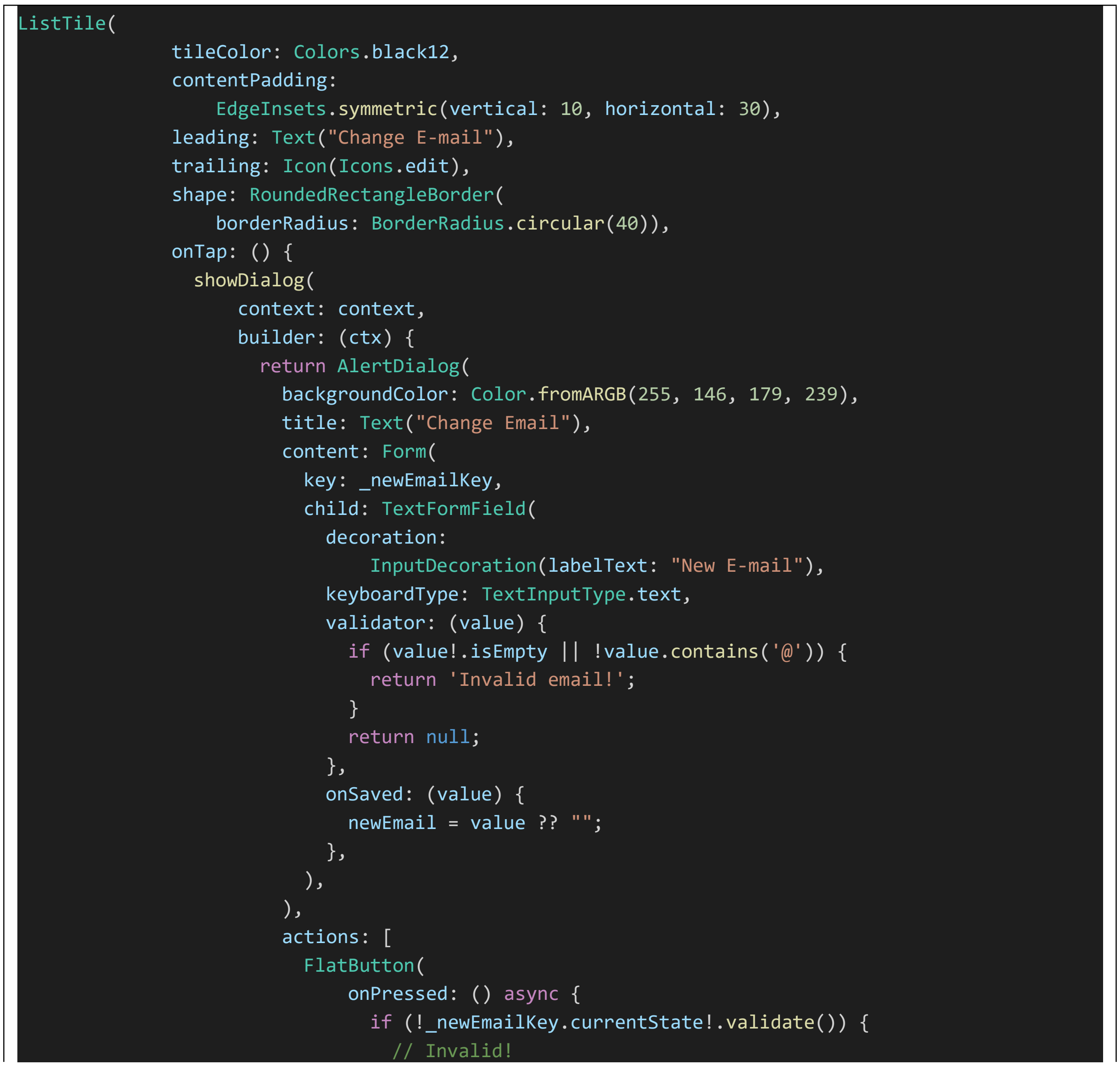


```
          ScaffoldMessenger.of(context)
              .showSnackBar(SnackBar(
                  content: Text(
            error.toString(),
            textAlign: TextAlign.center,
```

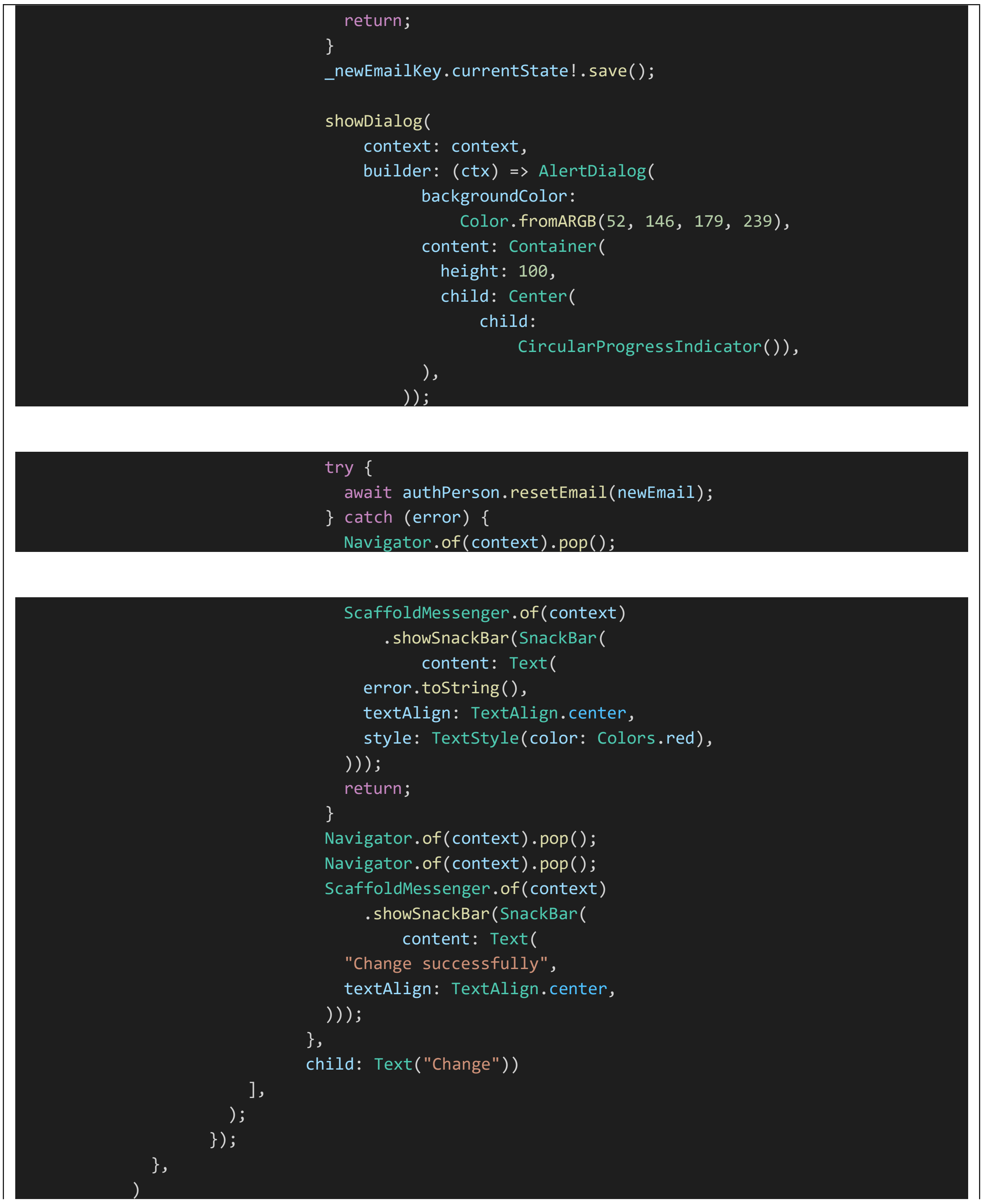


```
                                    style: TextStyle(color: Colors.red),
                                  )));
                                  return;
                                }
                                Navigator.of(context).pop();
                                Navigator.of(context).pop();
                                ScaffoldMessenger.of(context)
                                    .showSnackBar(SnackBar(
                                        content: Text(
                                  "Change successfully",
                                  textAlign: TextAlign.center,
                                )));
                              },
                              child: Text("Change"))
                        ],
                      );
                    });
              },
            )
```

Fig B-7 Change Password Dialog Box Form



```
ListTile(
              tileColor: Colors.black12,
              contentPadding:
                  EdgeInsets.symmetric(vertical: 10, horizontal: 30),
              leading: Text("Change E-mail"),
              trailing: Icon(Icons.edit),
              shape: RoundedRectangleBorder(
                  borderRadius: BorderRadius.circular(40)),
              onTap: () {
                showDialog(
                    context: context,
                    builder: (ctx) {
                      return AlertDialog(
                        backgroundColor: Color.fromARGB(255, 146, 179, 239),
                        title: Text("Change Email"),
                        content: Form(
                          key: _newEmailKey,
                          child: TextFormField(
                            decoration:
                                InputDecoration(labelText: "New E-mail"),
                            keyboardType: TextInputType.text,
                            validator: (value) {
                              if (value!.isEmpty || !value.contains('@')) {
                                return 'Invalid email!';
                              }
                              return null;
                            },
                            onSaved: (value) {
                              newEmail = value ?? "";
                            },
                          ),
                        ),
                        actions: [
                          FlatButton(
                              onPressed: () async {
                                if (!_newEmailKey.currentState!.validate()) {
                                  // Invalid!
```

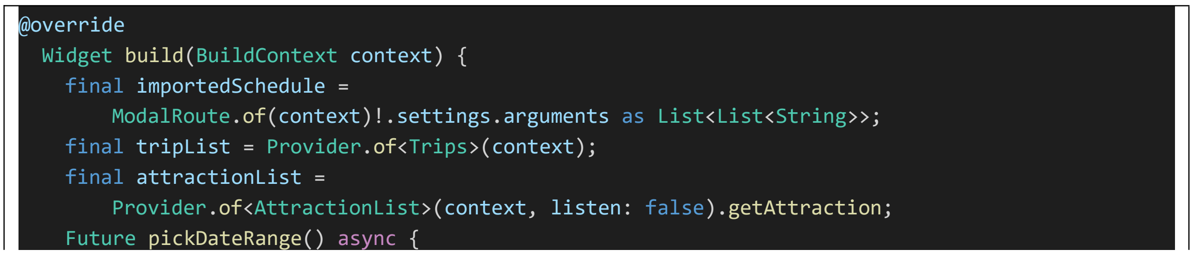


```
                                  return;
                                }
                                _newEmailKey.currentState!.save();

                                showDialog(
                                    context: context,
                                    builder: (ctx) => AlertDialog(
                                          backgroundColor:
                                              Color.fromARGB(52, 146, 179, 239),
                                          content: Container(
                                            height: 100,
                                            child: Center(
                                                child:
                                                    CircularProgressIndicator()),
                                          ),
                                        ));
```

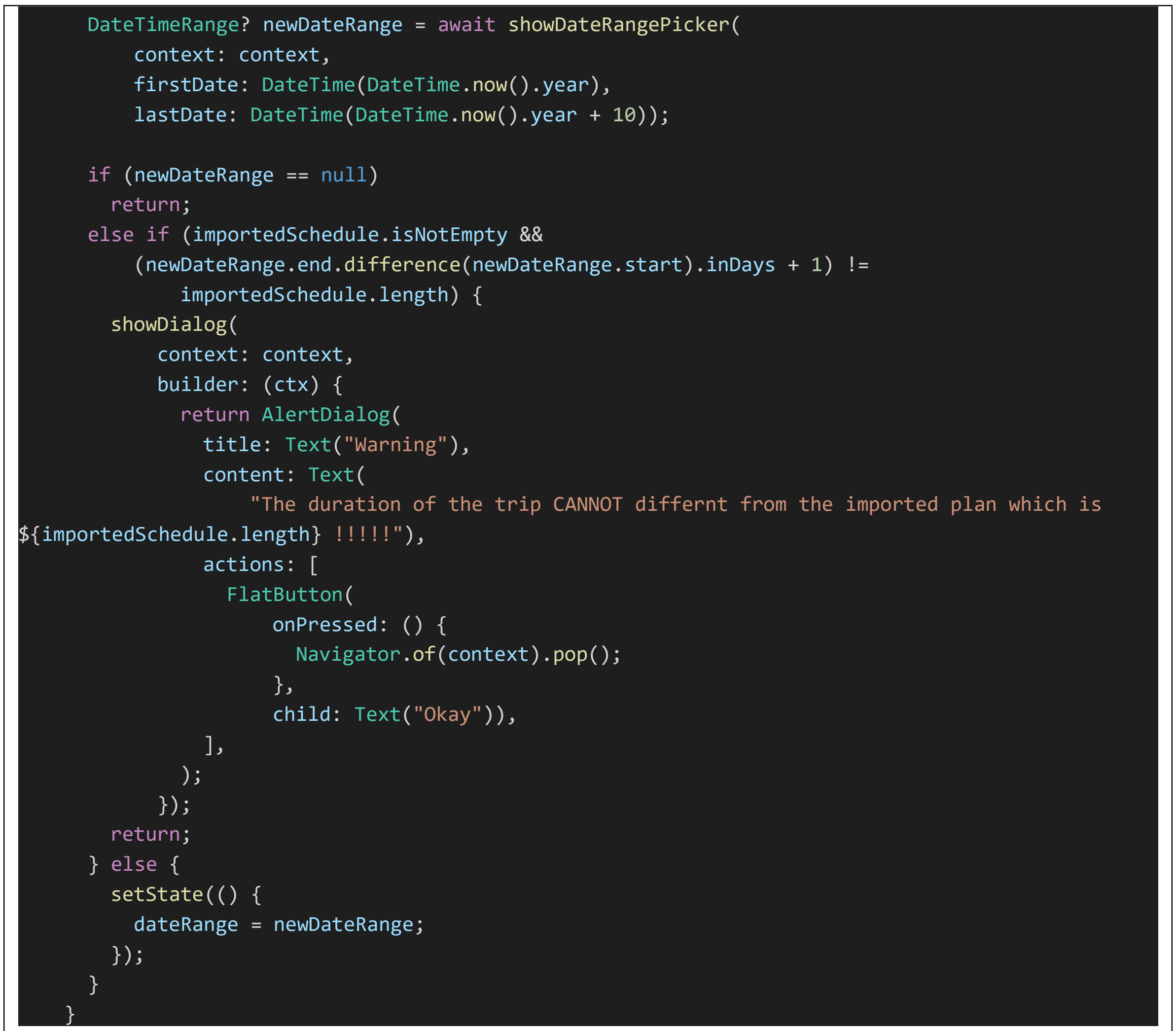
```
                                try {
                                  await authPerson.resetEmail(newEmail);
                                } catch (error) {
                                  Navigator.of(context).pop();
```

```
                                  ScaffoldMessenger.of(context)
                                      .showSnackBar(SnackBar(
                                          content: Text(
                                    error.toString(),
                                    textAlign: TextAlign.center,
                                    style: TextStyle(color: Colors.red),
                                  )));
                                  return;
                                }
                                Navigator.of(context).pop();
                                Navigator.of(context).pop();
                                ScaffoldMessenger.of(context)
                                    .showSnackBar(SnackBar(
                                        content: Text(
                                  "Change successfully",
                                  textAlign: TextAlign.center,
                                )));
                              },
                              child: Text("Change"))
                        ],
                      );
                    });
              },
            )
```

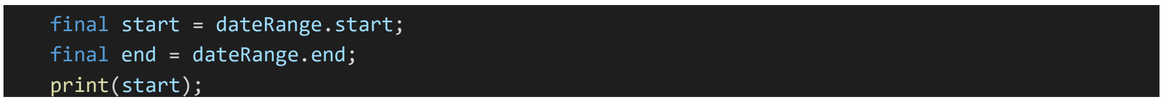
Fig B-8 Change Password Dialog Box Form



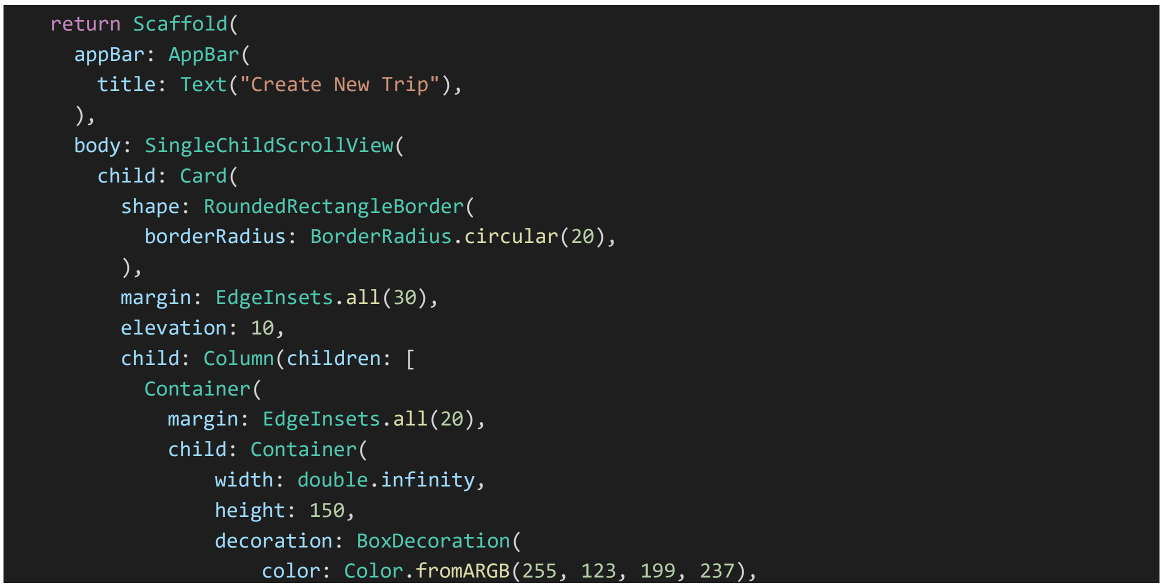
```
@override
  Widget build(BuildContext context) {
    final importedSchedule =
        ModalRoute.of(context)!.settings.arguments as List<List<String>>;
    final tripList = Provider.of<Trips>(context);
    final attractionList =
        Provider.of<AttractionList>(context, listen: false).getAttraction;
    Future pickDateRange() async {
```



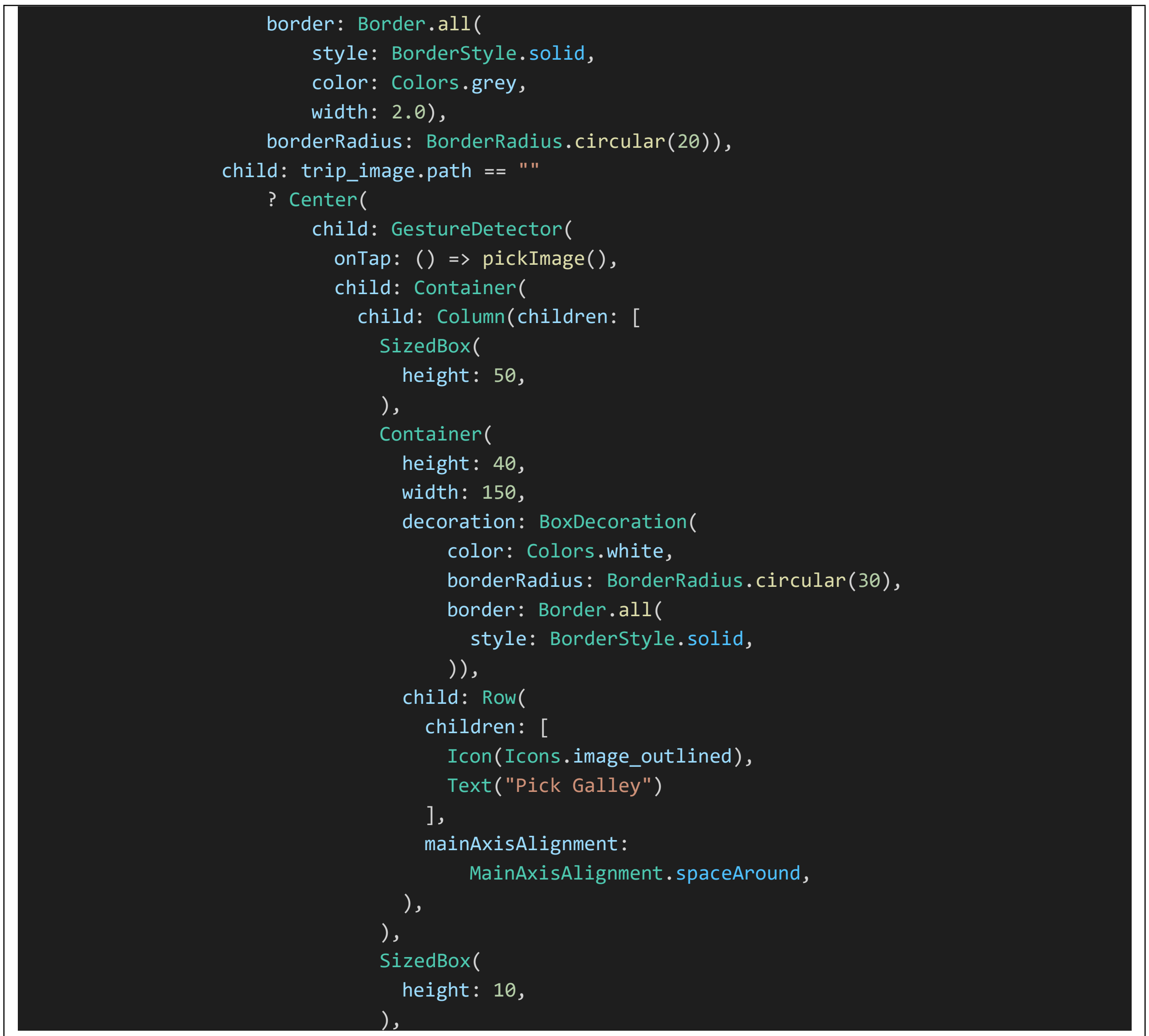
```
      DateTimeRange? newDateRange = await showDateRangePicker(
          context: context,
          firstDate: DateTime(DateTime.now().year),
          lastDate: DateTime(DateTime.now().year + 10));

      if (newDateRange == null)
        return;
      else if (importedSchedule.isNotEmpty &&
          (newDateRange.end.difference(newDateRange.start).inDays + 1) !=
              importedSchedule.length) {
        showDialog(
            context: context,
            builder: (ctx) {
              return AlertDialog(
                title: Text("Warning"),
                content: Text(
                    "The duration of the trip CANNOT differnt from the imported plan which is 
${importedSchedule.length} !!!!!"),
                actions: [
                  FlatButton(
                      onPressed: () {
                        Navigator.of(context).pop();
                      },
                      child: Text("Okay")),
                ],
              );
            });
        return;
      } else {
        setState(() {
          dateRange = newDateRange;
        });
      }
    }
```



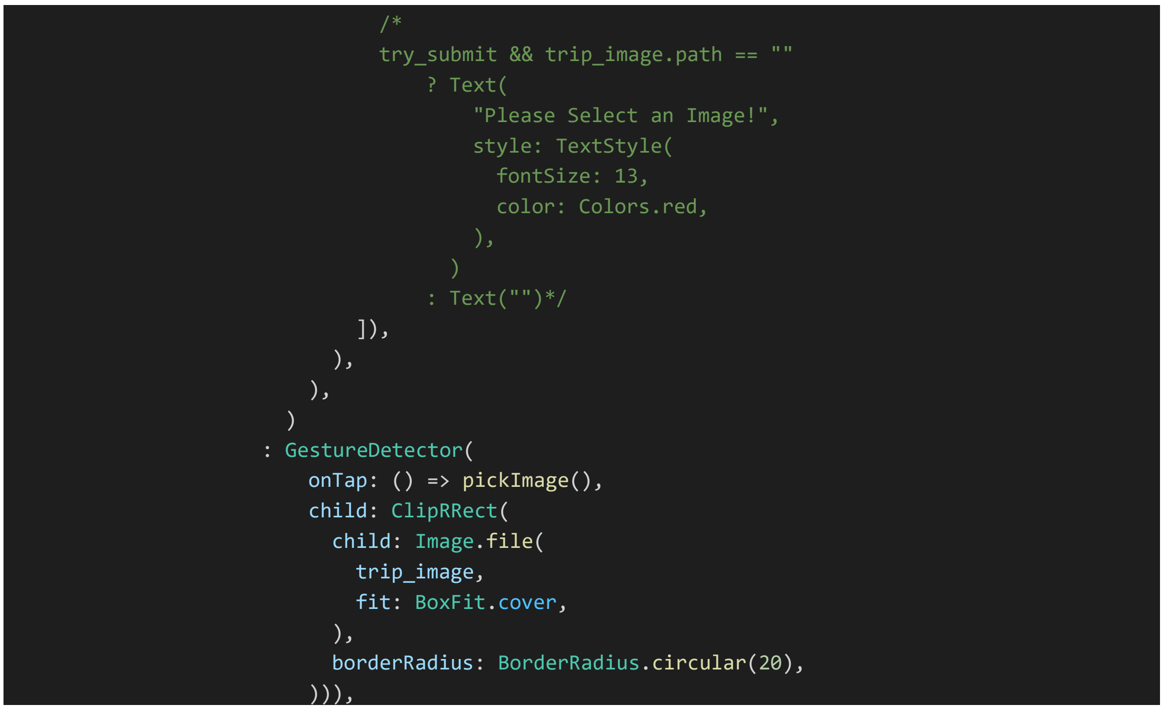
```
    final start = dateRange.start;
    final end = dateRange.end;
    print(start);
```



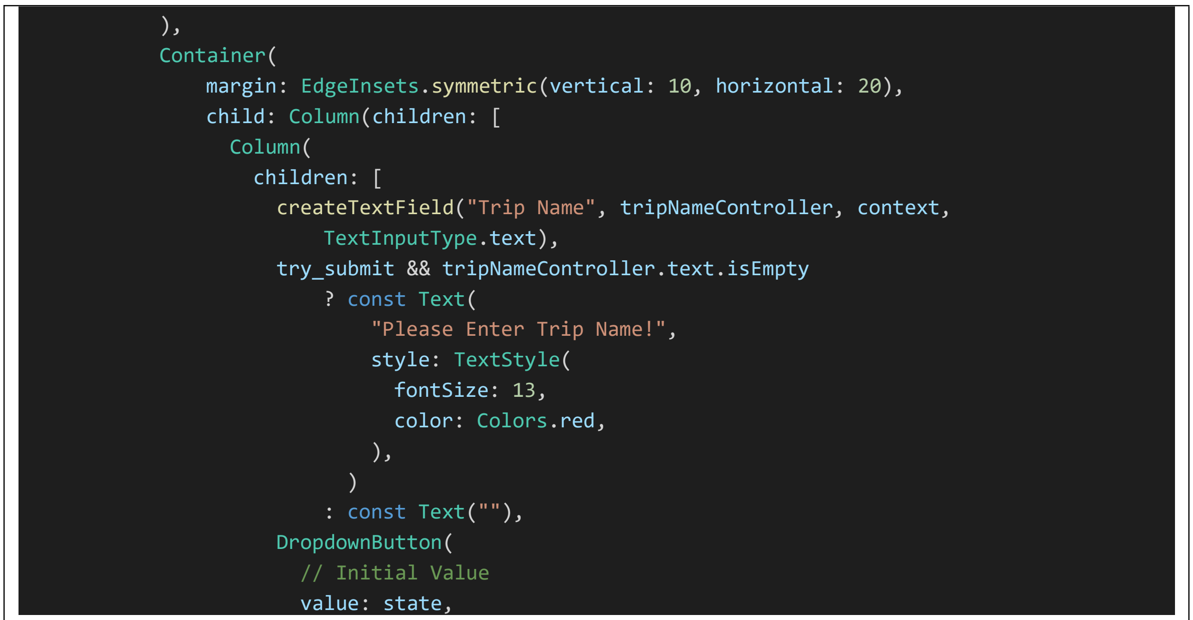
```
    return Scaffold(
      appBar: AppBar(
        title: Text("Create New Trip"),
      ),
      body: SingleChildScrollView(
        child: Card(
          shape: RoundedRectangleBorder(
            borderRadius: BorderRadius.circular(20),
          ),
          margin: EdgeInsets.all(30),
          elevation: 10,
          child: Column(children: [
            Container(
              margin: EdgeInsets.all(20),
              child: Container(
                  width: double.infinity,
                  height: 150,
                  decoration: BoxDecoration(
                      color: Color.fromARGB(255, 123, 199, 237),
```



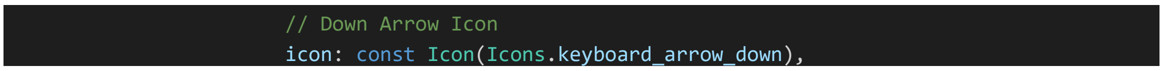
```
                  border: Border.all(
                      style: BorderStyle.solid,
                      color: Colors.grey,
                      width: 2.0),
                  borderRadius: BorderRadius.circular(20)),
              child: trip_image.path == ""
                  ? Center(
                      child: GestureDetector(
                        onTap: () => pickImage(),
                        child: Container(
                          child: Column(children: [
                            SizedBox(
                              height: 50,
                            ),
                            Container(
                              height: 40,
                              width: 150,
                              decoration: BoxDecoration(
                                  color: Colors.white,
                                  borderRadius: BorderRadius.circular(30),
                                  border: Border.all(
                                    style: BorderStyle.solid,
                                  )),
                              child: Row(
                                children: [
                                  Icon(Icons.image_outlined),
                                  Text("Pick Galley")
                                ],
                                mainAxisAlignment:
                                    MainAxisAlignment.spaceAround,
                              ),
                            ),
                            SizedBox(
                              height: 10,
                            ),
```



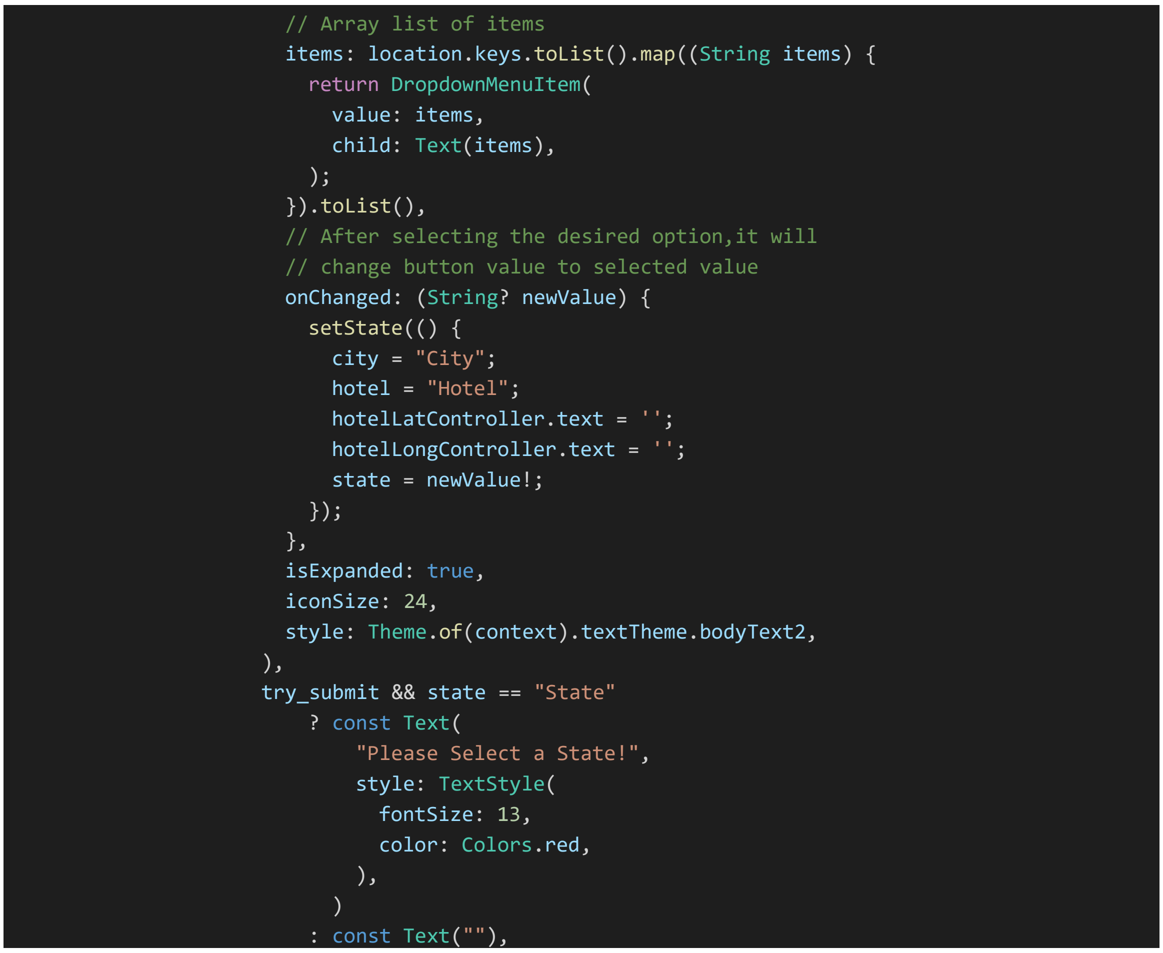
```
                            /*
                            try_submit && trip_image.path == ""
                                ? Text(
                                    "Please Select an Image!",
                                    style: TextStyle(
                                      fontSize: 13,
                                      color: Colors.red,
                                    ),
                                  )
                                : Text("")*/
                          ]),
                        ),
                      ),
                    )
                  : GestureDetector(
                      onTap: () => pickImage(),
                      child: ClipRRect(
                        child: Image.file(
                          trip_image,
                          fit: BoxFit.cover,
                        ),
                        borderRadius: BorderRadius.circular(20),
                      ))),
```



```
            ),
            Container(
                margin: EdgeInsets.symmetric(vertical: 10, horizontal: 20),
                child: Column(children: [
                  Column(
                    children: [
                      createTextField("Trip Name", tripNameController, context,
                          TextInputType.text),
                      try_submit && tripNameController.text.isEmpty
                          ? const Text(
                              "Please Enter Trip Name!",
                              style: TextStyle(
                                fontSize: 13,
                                color: Colors.red,
                              ),
                            )
                          : const Text(""),
                      DropdownButton(
                        // Initial Value
                        value: state,
```

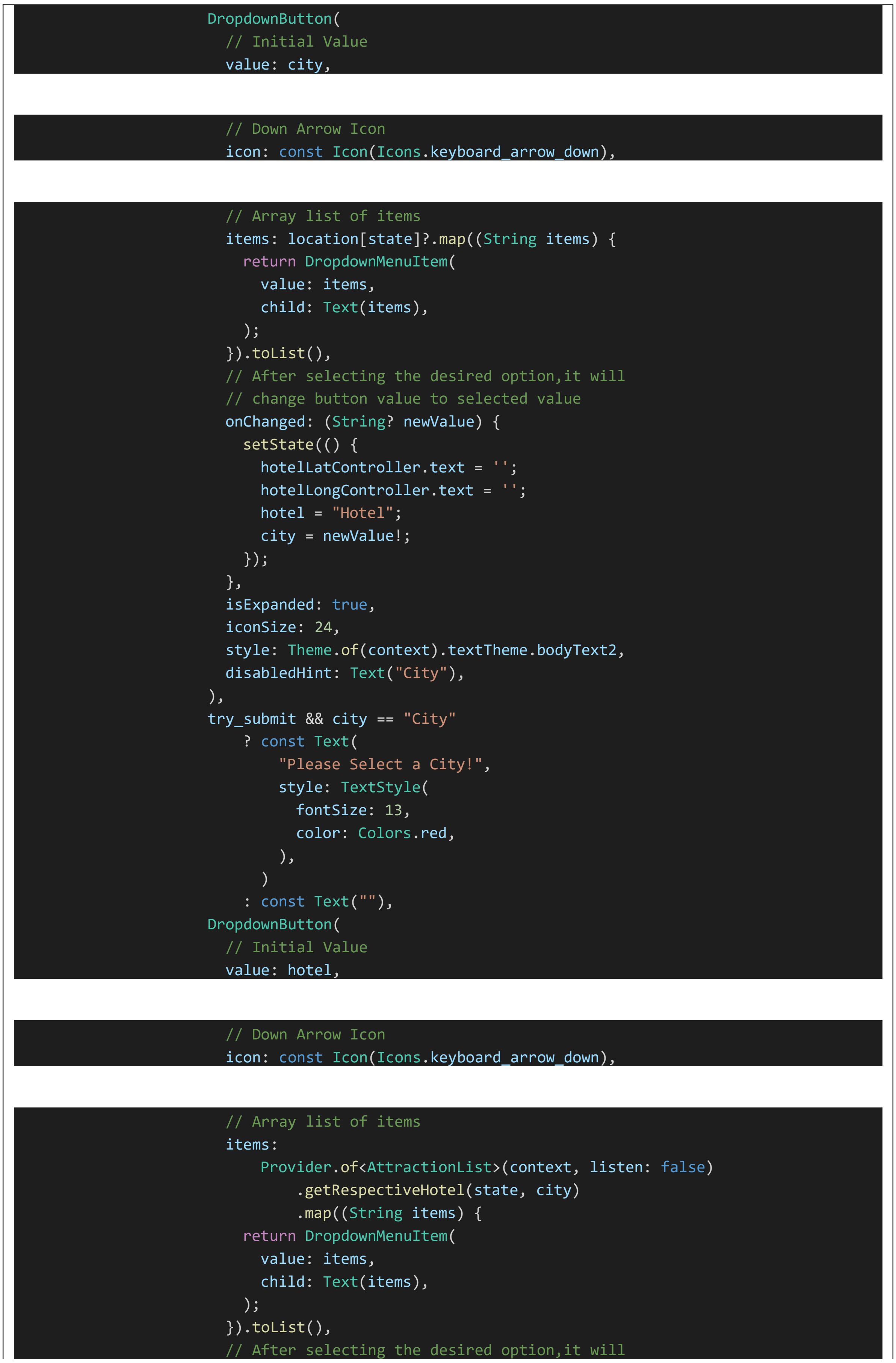


```
                        // Down Arrow Icon
                        icon: const Icon(Icons.keyboard_arrow_down),
```



```
                        // Array list of items
                        items: location.keys.toList().map((String items) {
                          return DropdownMenuItem(
                            value: items,
                            child: Text(items),
                          );
                        }).toList(),
                        // After selecting the desired option,it will
                        // change button value to selected value
                        onChanged: (String? newValue) {
                          setState(() {
                            city = "City";
                            hotel = "Hotel";
                            hotelLatController.text = '';
                            hotelLongController.text = '';
                            state = newValue!;
                          });
                        },
                        isExpanded: true,
                        iconSize: 24,
                        style: Theme.of(context).textTheme.bodyText2,
                      ),
                      try_submit && state == "State"
                          ? const Text(
                              "Please Select a State!",
                              style: TextStyle(
                                fontSize: 13,
                                color: Colors.red,
                              ),
                            )
                          : const Text(""),
```

```
                      //city
```



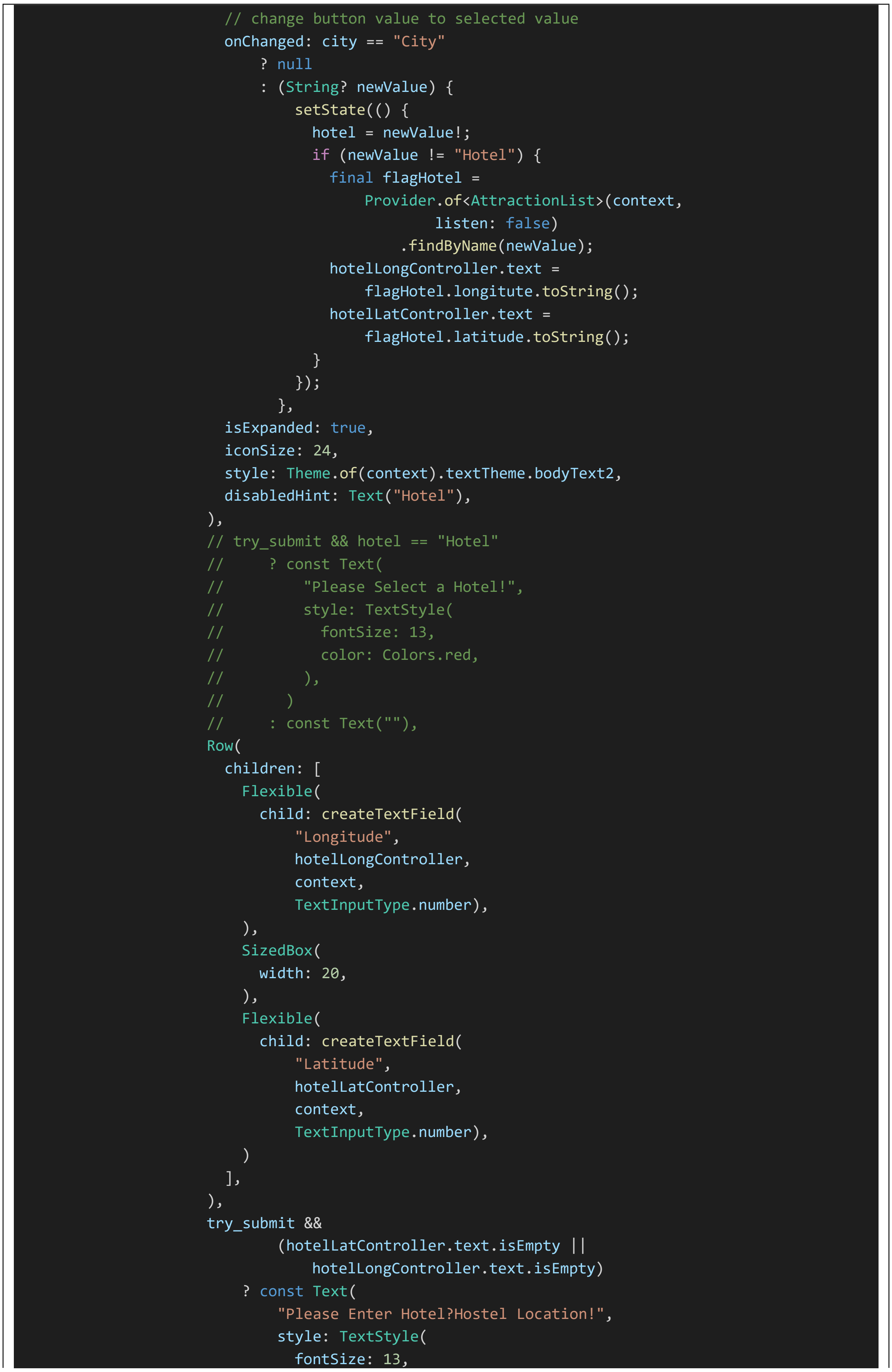
```
DropdownButton(
  // Initial Value
  value: city,
```

```
  // Down Arrow Icon
  icon: const Icon(Icons.keyboard_arrow_down),
```

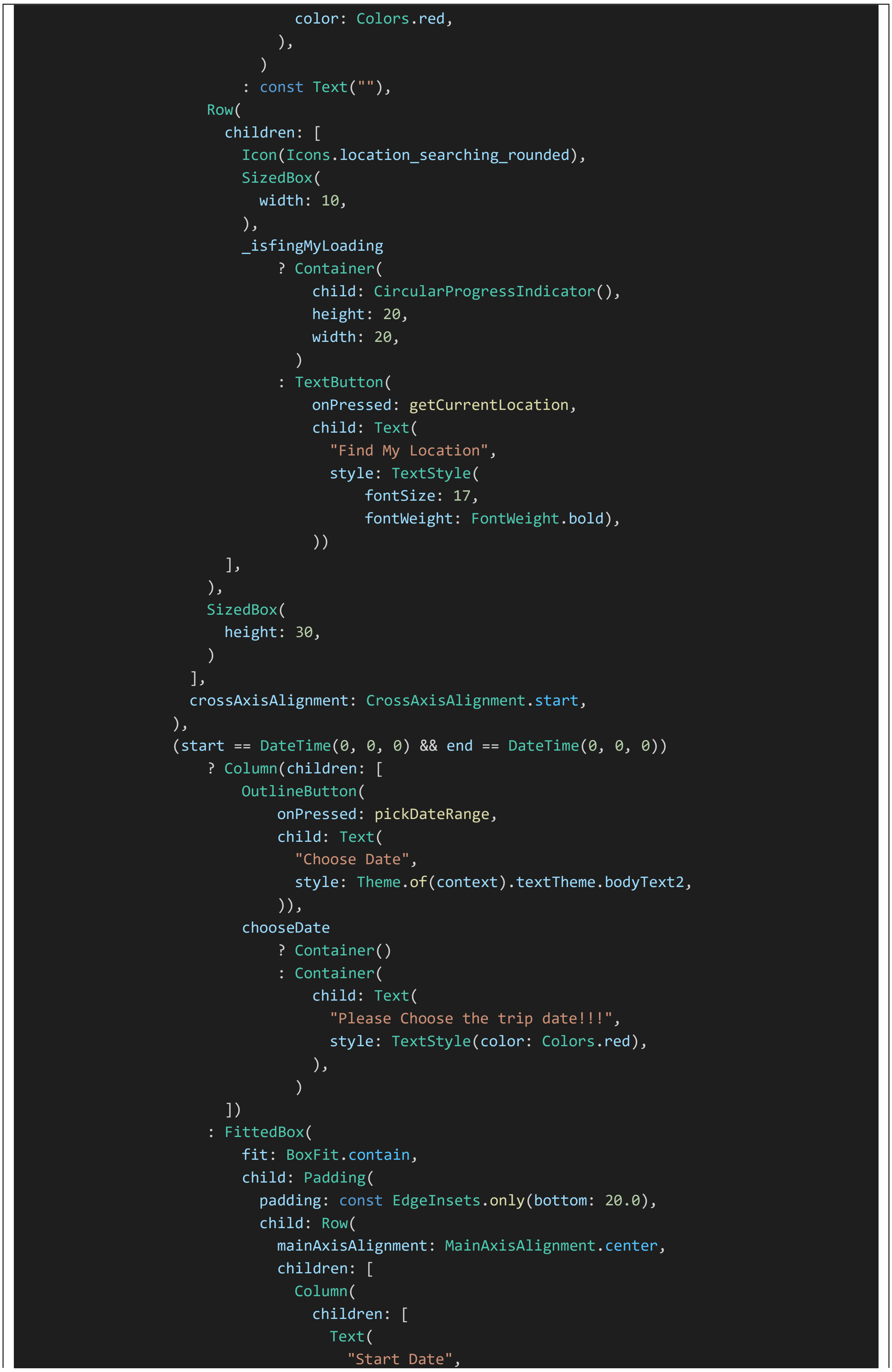
```
  // Array list of items
  items: location[state]?.map((String items) {
    return DropdownMenuItem(
      value: items,
      child: Text(items),
    );
  }).toList(),
  // After selecting the desired option,it will
  // change button value to selected value
  onChanged: (String? newValue) {
    setState(() {
      hotelLatController.text = '';
      hotelLongController.text = '';
      hotel = "Hotel";
      city = newValue!;
    });
  },
  isExpanded: true,
  iconSize: 24,
  style: Theme.of(context).textTheme.bodyText2,
  disabledHint: Text("City"),
),
try_submit && city == "City"
    ? const Text(
        "Please Select a City!",
        style: TextStyle(
          fontSize: 13,
          color: Colors.red,
        ),
      )
    : const Text(""),
DropdownButton(
  // Initial Value
  value: hotel,
```

```
  // Down Arrow Icon
  icon: const Icon(Icons.keyboard_arrow_down),
```

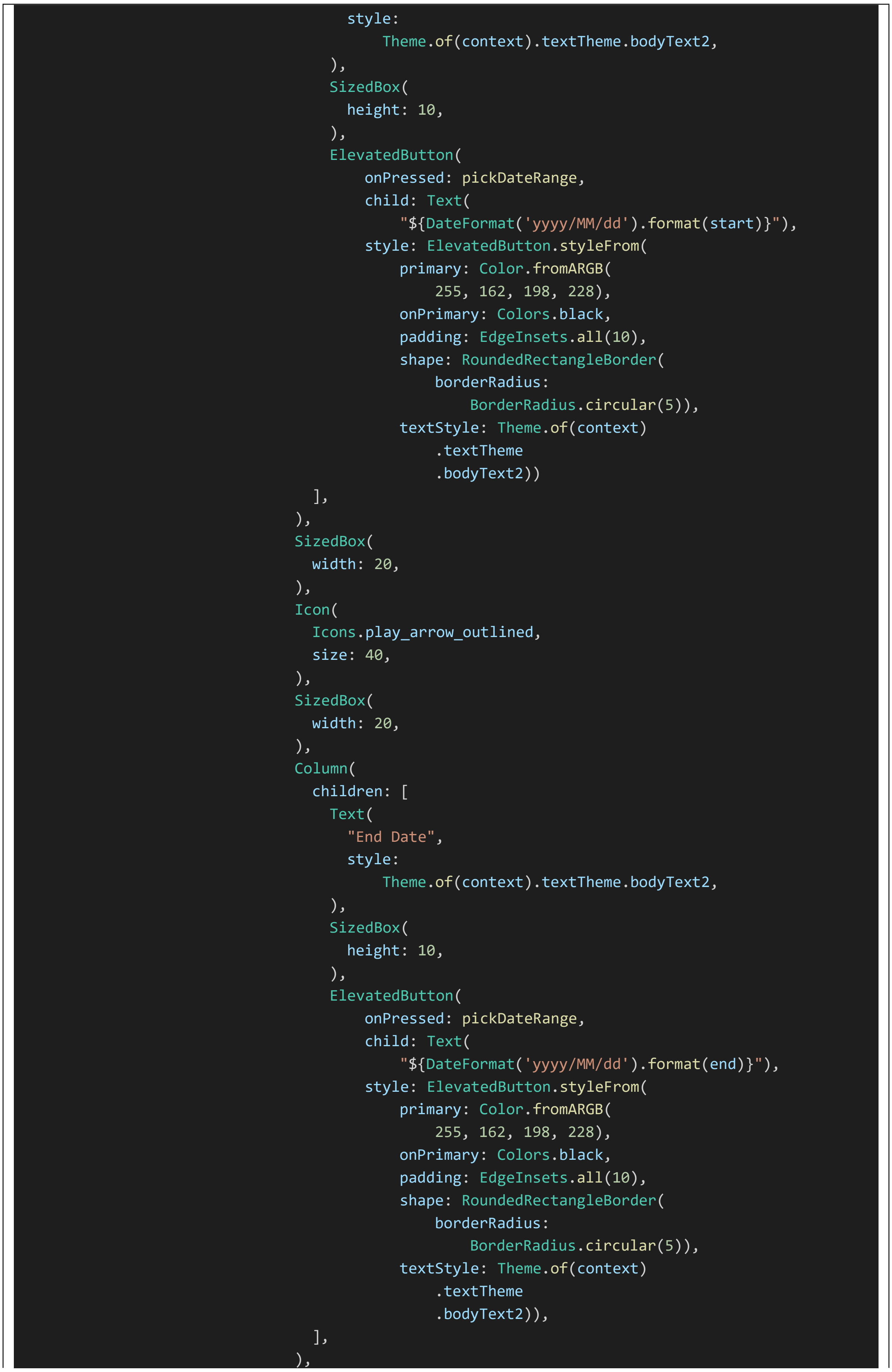
```
  // Array list of items
  items:
      Provider.of<AttractionList>(context, listen: false)
          .getRespectiveHotel(state, city)
          .map((String items) {
    return DropdownMenuItem(
      value: items,
      child: Text(items),
    );
  }).toList(),
  // After selecting the desired option,it will
```



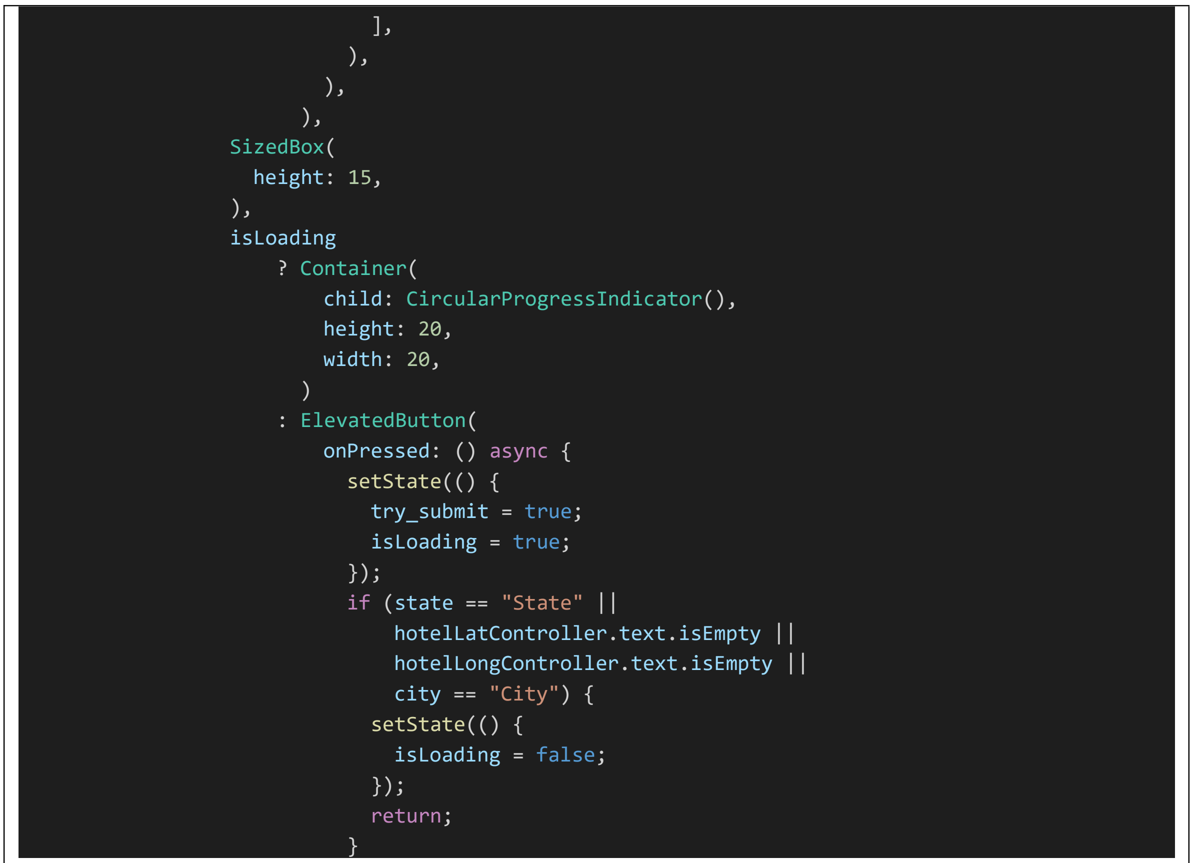
```
  // change button value to selected value
  onChanged: city == "City"
      ? null
      : (String? newValue) {
          setState(() {
            hotel = newValue!;
            if (newValue != "Hotel") {
              final flagHotel =
                  Provider.of<AttractionList>(context,
                          listen: false)
                      .findByName(newValue);
              hotelLongController.text =
                  flagHotel.longitute.toString();
              hotelLatController.text =
                  flagHotel.latitude.toString();
            }
          });
        },
  isExpanded: true,
  iconSize: 24,
  style: Theme.of(context).textTheme.bodyText2,
  disabledHint: Text("Hotel"),
),
// try_submit && hotel == "Hotel"
//     ? const Text(
//         "Please Select a Hotel!",
//         style: TextStyle(
//           fontSize: 13,
//           color: Colors.red,
//         ),
//       )
//     : const Text(""),
Row(
  children: [
    Flexible(
      child: createTextField(
          "Longitude",
          hotelLongController,
          context,
          TextInputType.number),
    ),
    SizedBox(
      width: 20,
    ),
    Flexible(
      child: createTextField(
          "Latitude",
          hotelLatController,
          context,
          TextInputType.number),
    )
  ],
),
try_submit &&
        (hotelLatController.text.isEmpty ||
            hotelLongController.text.isEmpty)
    ? const Text(
        "Please Enter Hotel?Hostel Location!",
        style: TextStyle(
          fontSize: 13,
```



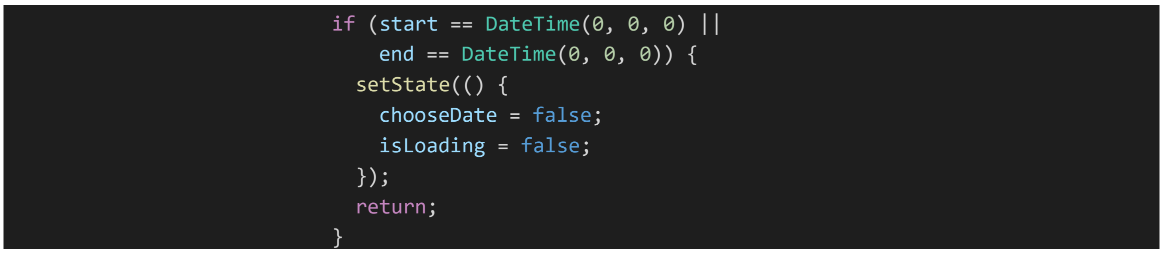
```
                            color: Colors.red,
                          ),
                        )
                      : const Text(""),
                  Row(
                    children: [
                      Icon(Icons.location_searching_rounded),
                      SizedBox(
                        width: 10,
                      ),
                      _isfingMyLoading
                          ? Container(
                              child: CircularProgressIndicator(),
                              height: 20,
                              width: 20,
                            )
                          : TextButton(
                              onPressed: getCurrentLocation,
                              child: Text(
                                "Find My Location",
                                style: TextStyle(
                                    fontSize: 17,
                                    fontWeight: FontWeight.bold),
                              ))
                    ],
                  ),
                  SizedBox(
                    height: 30,
                  )
                ],
                crossAxisAlignment: CrossAxisAlignment.start,
              ),
              (start == DateTime(0, 0, 0) && end == DateTime(0, 0, 0))
                  ? Column(children: [
                      OutlineButton(
                          onPressed: pickDateRange,
                          child: Text(
                            "Choose Date",
                            style: Theme.of(context).textTheme.bodyText2,
                          )),
                      chooseDate
                          ? Container()
                          : Container(
                              child: Text(
                                "Please Choose the trip date!!!",
                                style: TextStyle(color: Colors.red),
                              ),
                            )
                    ])
                  : FittedBox(
                      fit: BoxFit.contain,
                      child: Padding(
                        padding: const EdgeInsets.only(bottom: 20.0),
                        child: Row(
                          mainAxisAlignment: MainAxisAlignment.center,
                          children: [
                            Column(
                              children: [
                                Text(
                                  "Start Date",
```



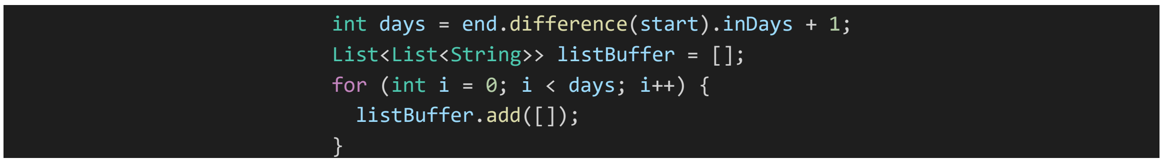
```
                                    style:
                                        Theme.of(context).textTheme.bodyText2,
                                  ),
                                  SizedBox(
                                    height: 10,
                                  ),
                                  ElevatedButton(
                                      onPressed: pickDateRange,
                                      child: Text(
                                          "${DateFormat('yyyy/MM/dd').format(start)}"),
                                      style: ElevatedButton.styleFrom(
                                          primary: Color.fromARGB(
                                              255, 162, 198, 228),
                                          onPrimary: Colors.black,
                                          padding: EdgeInsets.all(10),
                                          shape: RoundedRectangleBorder(
                                              borderRadius:
                                                  BorderRadius.circular(5)),
                                          textStyle: Theme.of(context)
                                              .textTheme
                                              .bodyText2))
                                ],
                              ),
                              SizedBox(
                                width: 20,
                              ),
                              Icon(
                                Icons.play_arrow_outlined,
                                size: 40,
                              ),
                              SizedBox(
                                width: 20,
                              ),
                              Column(
                                children: [
                                  Text(
                                    "End Date",
                                    style:
                                        Theme.of(context).textTheme.bodyText2,
                                  ),
                                  SizedBox(
                                    height: 10,
                                  ),
                                  ElevatedButton(
                                      onPressed: pickDateRange,
                                      child: Text(
                                          "${DateFormat('yyyy/MM/dd').format(end)}"),
                                      style: ElevatedButton.styleFrom(
                                          primary: Color.fromARGB(
                                              255, 162, 198, 228),
                                          onPrimary: Colors.black,
                                          padding: EdgeInsets.all(10),
                                          shape: RoundedRectangleBorder(
                                              borderRadius:
                                                  BorderRadius.circular(5)),
                                          textStyle: Theme.of(context)
                                              .textTheme
                                              .bodyText2)),
                                ],
                              ),
```



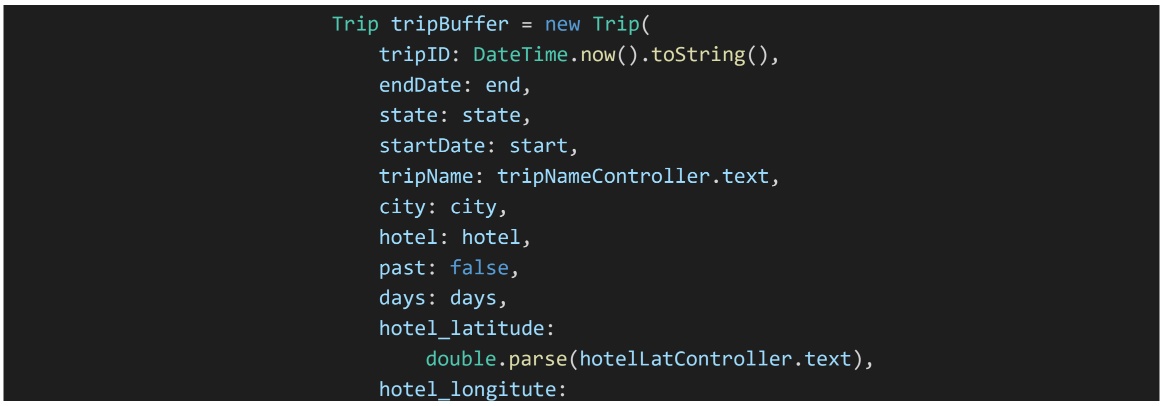
```
            ],
          ),
        ),
      ),
SizedBox(
  height: 15,
),
isLoading
    ? Container(
        child: CircularProgressIndicator(),
        height: 20,
        width: 20,
      )
    : ElevatedButton(
        onPressed: () async {
          setState(() {
            try_submit = true;
            isLoading = true;
          });
          if (state == "State" ||
              hotelLatController.text.isEmpty ||
              hotelLongController.text.isEmpty ||
              city == "City") {
            setState(() {
              isLoading = false;
            });
            return;
          }
```



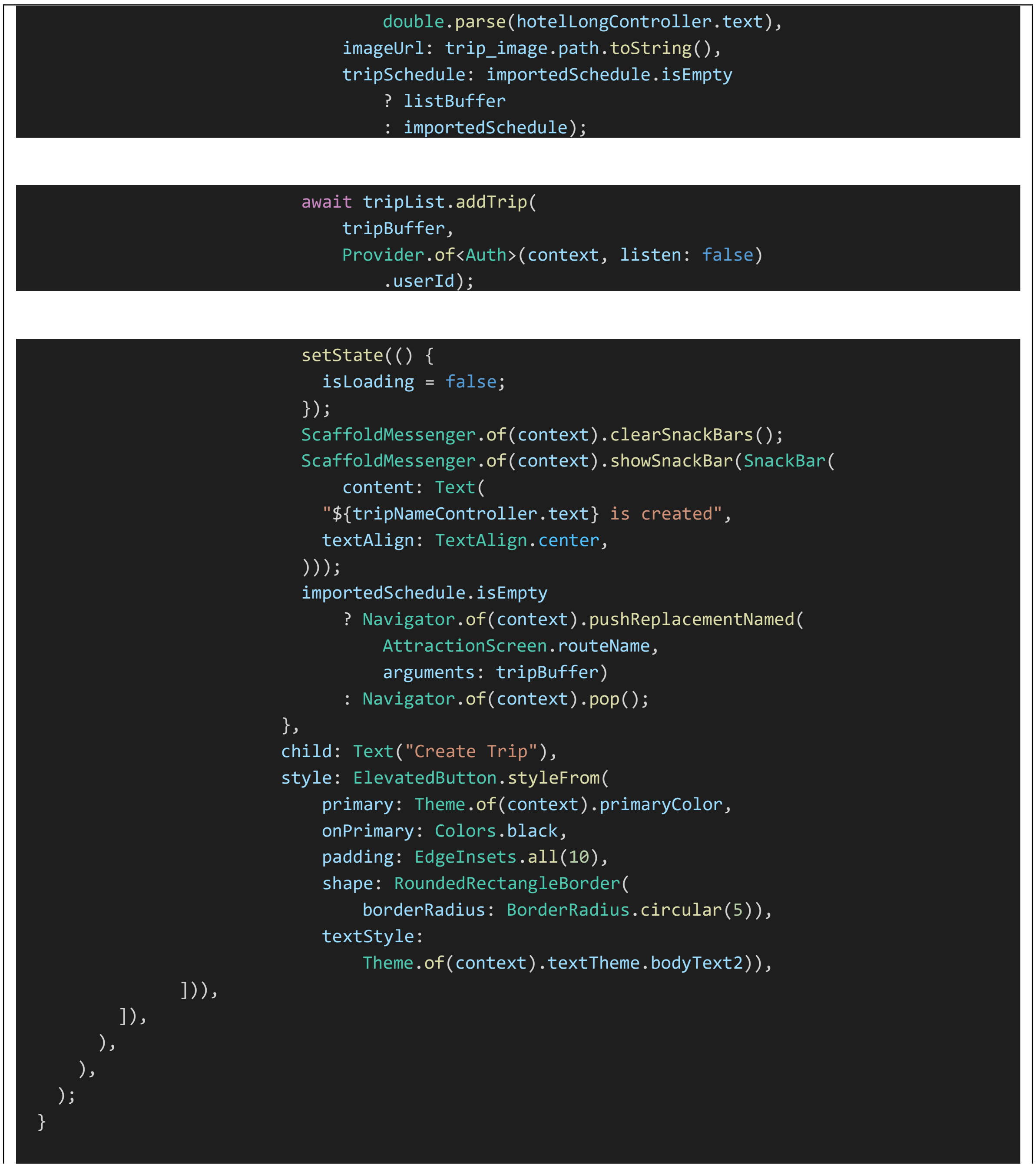
```
          if (start == DateTime(0, 0, 0) ||
              end == DateTime(0, 0, 0)) {
            setState(() {
              chooseDate = false;
              isLoading = false;
            });
            return;
          }
```



```
          int days = end.difference(start).inDays + 1;
          List<List<String>> listBuffer = [];
          for (int i = 0; i < days; i++) {
            listBuffer.add([]);
          }
```



```
          Trip tripBuffer = new Trip(
              tripID: DateTime.now().toString(),
              endDate: end,
              state: state,
              startDate: start,
              tripName: tripNameController.text,
              city: city,
              hotel: hotel,
              past: false,
              days: days,
              hotel_latitude:
                  double.parse(hotelLatController.text),
              hotel_longitute:
```

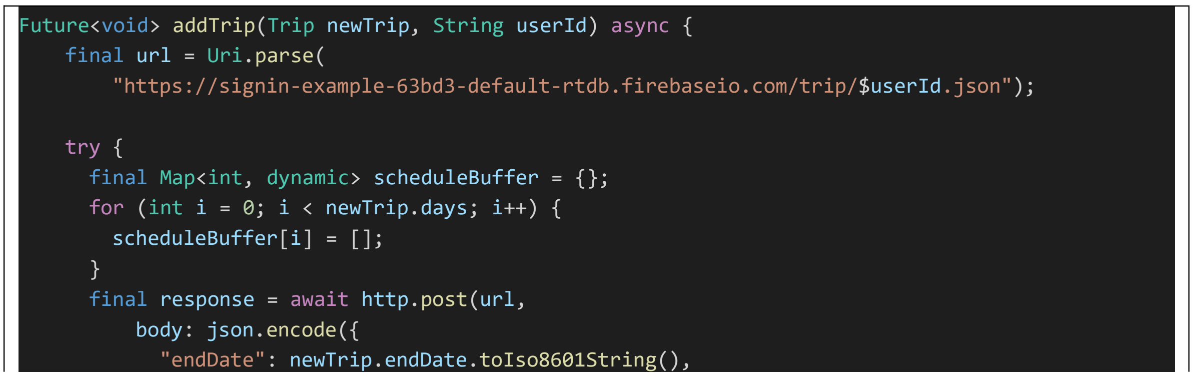


```
                  double.parse(hotelLongController.text),
                imageUrl: trip_image.path.toString(),
                tripSchedule: importedSchedule.isEmpty
                  ? listBuffer
                  : importedSchedule);
```

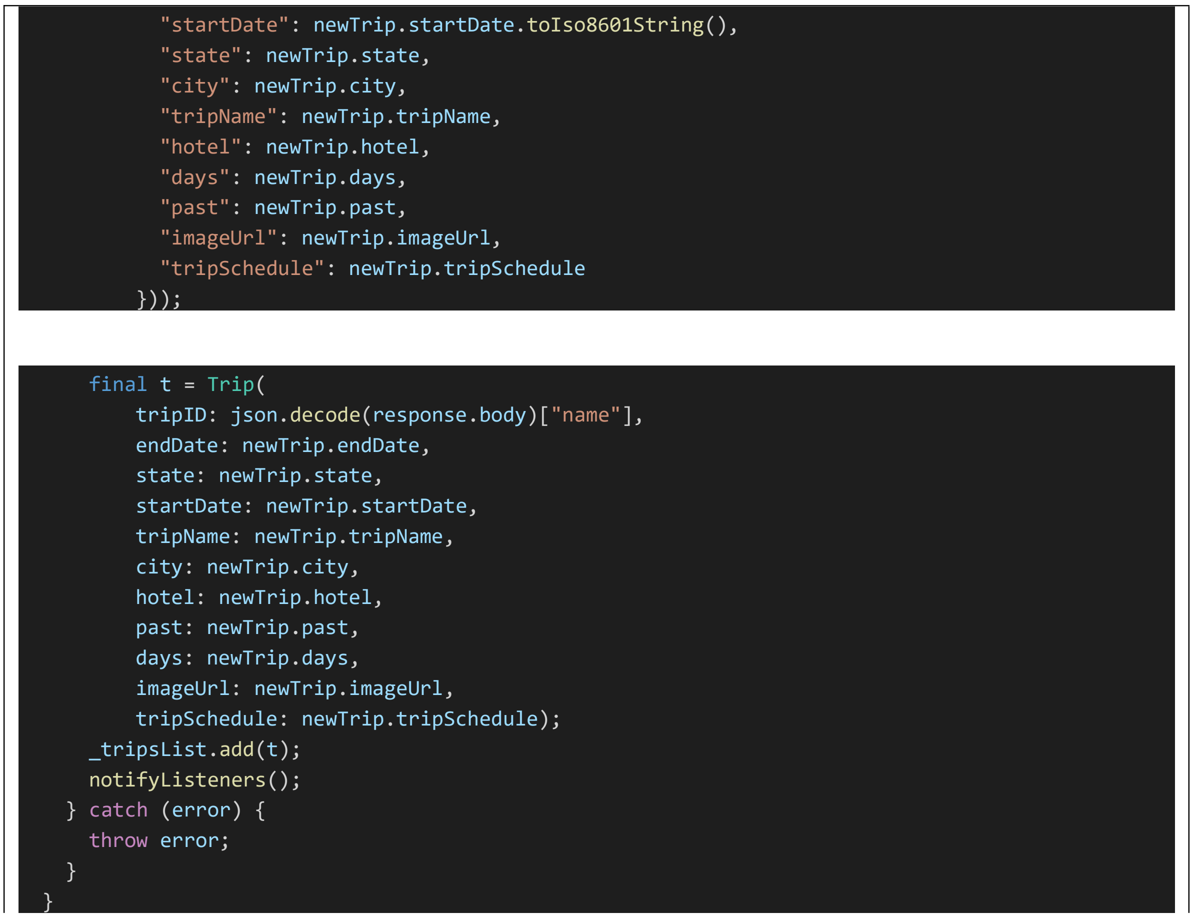
```
              await tripList.addTrip(
                tripBuffer,
                Provider.of<Auth>(context, listen: false)
                  .userId);
```

```
              setState(() {
               isLoading = false;
              });
              ScaffoldMessenger.of(context).clearSnackBars();
              ScaffoldMessenger.of(context).showSnackBar(SnackBar(
                content: Text(
               "${tripNameController.text} is created",
               textAlign: TextAlign.center,
              )));
              importedSchedule.isEmpty
                ? Navigator.of(context).pushReplacementNamed(
                  AttractionScreen.routeName,
                  arguments: tripBuffer)
                : Navigator.of(context).pop();
             },
             child: Text("Create Trip"),
             style: ElevatedButton.styleFrom(
               primary: Theme.of(context).primaryColor,
               onPrimary: Colors.black,
               padding: EdgeInsets.all(10),
               shape: RoundedRectangleBorder(
                 borderRadius: BorderRadius.circular(5)),
               textStyle:
                 Theme.of(context).textTheme.bodyText2)),
        ])),
     ]),
    ),
   ),
  );
 }
```

Fig B-9 Create New Trip Form

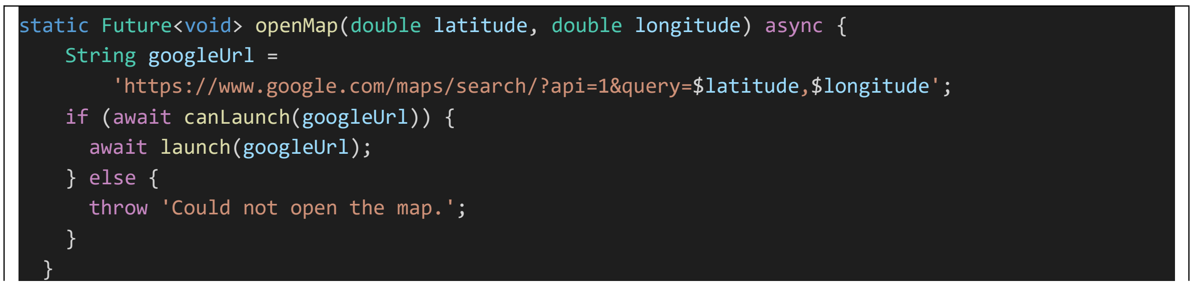


```
Future<void> addTrip(Trip newTrip, String userId) async {
  final url = Uri.parse(
    "https://signin-example-63bd3-default-rtdb.firebaseio.com/trip/$userId.json");

  try {
   final Map<int, dynamic> scheduleBuffer = {};
   for (int i = 0; i < newTrip.days; i++) {
    scheduleBuffer[i] = [];
   }
   final response = await http.post(url,
     body: json.encode({
      "endDate": newTrip.endDate.toIso8601String(),
```



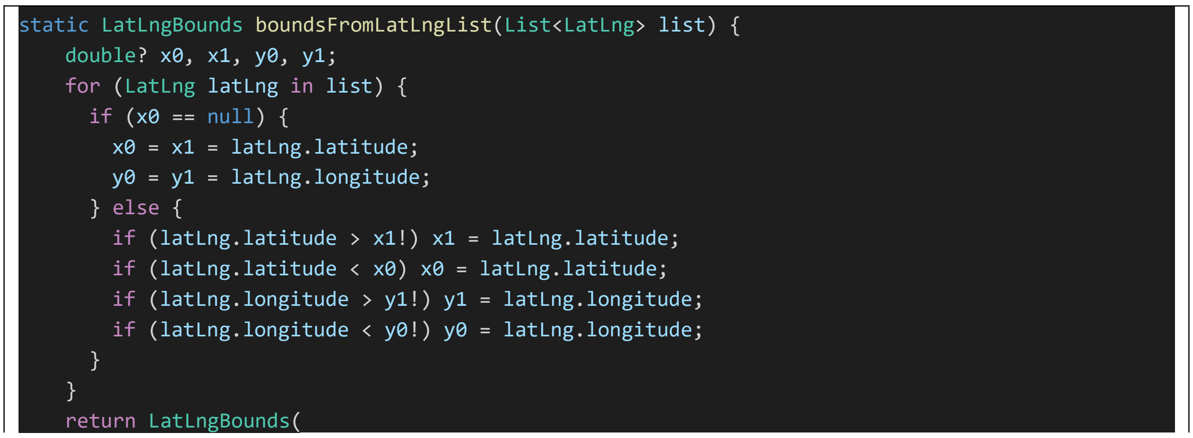
```
            "startDate": newTrip.startDate.toIso8601String(),
            "state": newTrip.state,
            "city": newTrip.city,
            "tripName": newTrip.tripName,
            "hotel": newTrip.hotel,
            "days": newTrip.days,
            "past": newTrip.past,
            "imageUrl": newTrip.imageUrl,
            "tripSchedule": newTrip.tripSchedule
          }));

      final t = Trip(
          tripID: json.decode(response.body)["name"],
          endDate: newTrip.endDate,
          state: newTrip.state,
          startDate: newTrip.startDate,
          tripName: newTrip.tripName,
          city: newTrip.city,
          hotel: newTrip.hotel,
          past: newTrip.past,
          days: newTrip.days,
          imageUrl: newTrip.imageUrl,
          tripSchedule: newTrip.tripSchedule);
      _tripsList.add(t);
      notifyListeners();
    } catch (error) {
      throw error;
    }
  }
```

Fig B-10 Create New Trip

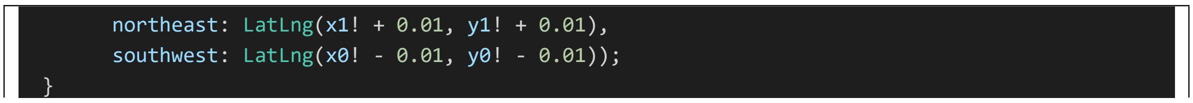


```
static Future<void> openMap(double latitude, double longitude) async {
    String googleUrl =
        'https://www.google.com/maps/search/?api=1&query=$latitude,$longitude';
    if (await canLaunch(googleUrl)) {
      await launch(googleUrl);
    } else {
      throw 'Could not open the map.';
    }
  }
```

Fig B-11 Open Google Map

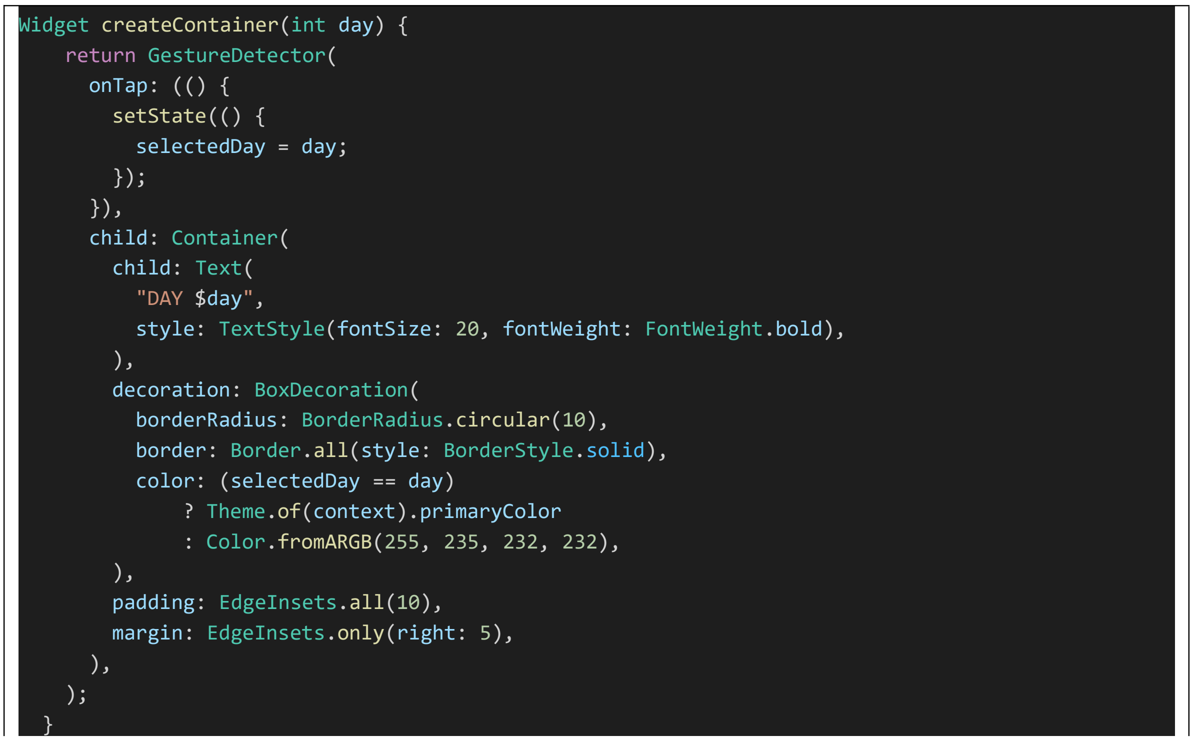


```
static LatLngBounds boundsFromLatLngList(List<LatLng> list) {
    double? x0, x1, y0, y1;
    for (LatLng latLng in list) {
      if (x0 == null) {
        x0 = x1 = latLng.latitude;
        y0 = y1 = latLng.longitude;
      } else {
        if (latLng.latitude > x1!) x1 = latLng.latitude;
        if (latLng.latitude < x0) x0 = latLng.latitude;
        if (latLng.longitude > y1!) y1 = latLng.longitude;
        if (latLng.longitude < y0!) y0 = latLng.longitude;
      }
    }
    return LatLngBounds(
```



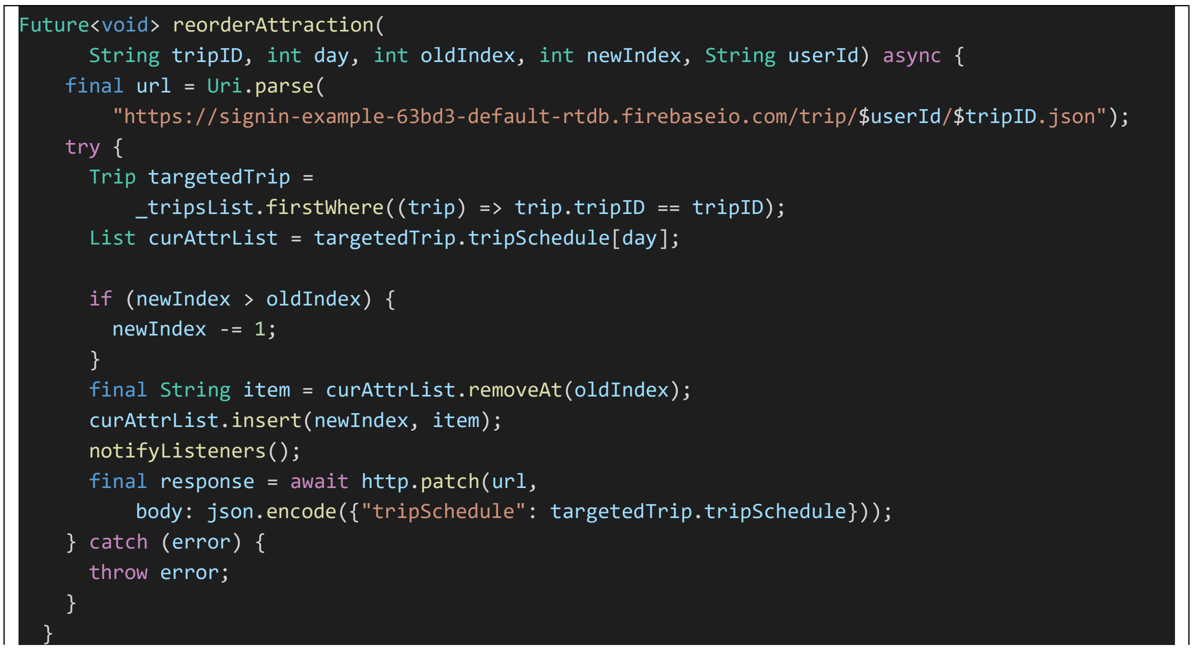
```
        northeast: LatLng(x1! + 0.01, y1! + 0.01),
        southwest: LatLng(x0! - 0.01, y0! - 0.01));
  }
```

Fig B-12 Recenter Function



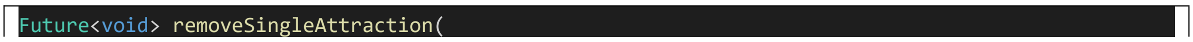
```
Widget createContainer(int day) {
    return GestureDetector(
      onTap: (() {
        setState(() {
          selectedDay = day;
        });
      }),
      child: Container(
        child: Text(
          "DAY $day",
          style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold),
        ),
        decoration: BoxDecoration(
          borderRadius: BorderRadius.circular(10),
          border: Border.all(style: BorderStyle.solid),
          color: (selectedDay == day)
              ? Theme.of(context).primaryColor
              : Color.fromARGB(255, 235, 232, 232),
        ),
        padding: EdgeInsets.all(10),
        margin: EdgeInsets.only(right: 5),
      ),
    );
  }
```

Fig B-13 Create days selection on "Trip Detail" screen



```
Future<void> reorderAttraction(
      String tripID, int day, int oldIndex, int newIndex, String userId) async {
    final url = Uri.parse(
        "https://signin-example-63bd3-default-rtdb.firebaseio.com/trip/$userId/$tripID.json");
    try {
      Trip targetedTrip =
          _tripsList.firstWhere((trip) => trip.tripID == tripID);
      List curAttrList = targetedTrip.tripSchedule[day];

      if (newIndex > oldIndex) {
        newIndex -= 1;
      }
      final String item = curAttrList.removeAt(oldIndex);
      curAttrList.insert(newIndex, item);
      notifyListeners();
      final response = await http.patch(url,
          body: json.encode({"tripSchedule": targetedTrip.tripSchedule}));
    } catch (error) {
      throw error;
    }
  }
```

Fig B-14 Reorder the attraction list in the "Trip Detail" screen



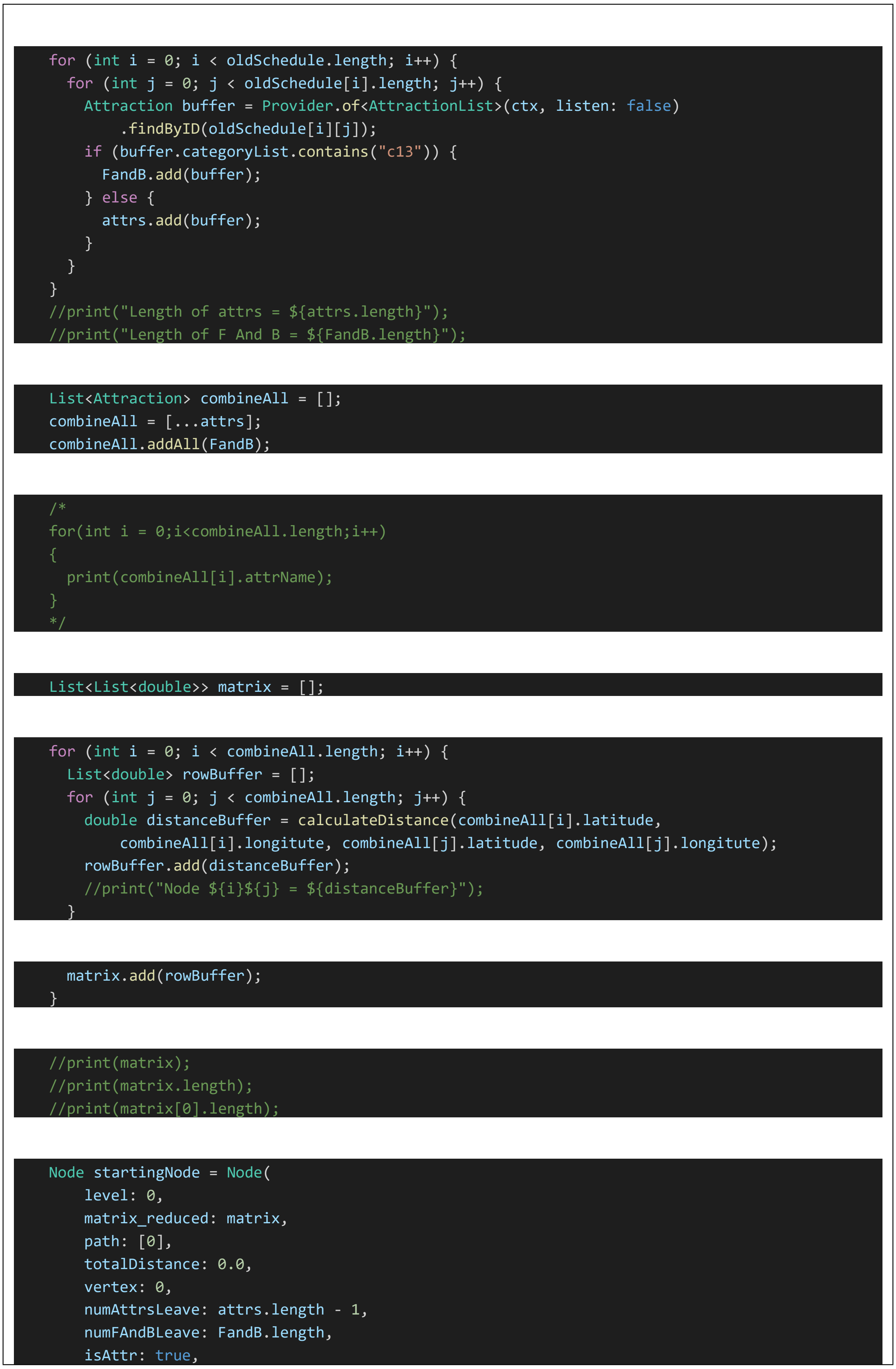
```
Future<void> removeSingleAttraction(
      String tripID, int day, int attrsequence, String userId) async {
    final url = Uri.parse(
        "https://signin-example-63bd3-default-
rtdb.firebaseio.com/trip/$userId/$tripID/tripSchedule/$day/$attrsequence.json");
    try {
      Trip targetedTrip =
          _tripsList.firstWhere((trip) => trip.tripID == tripID);
      targetedTrip.tripSchedule[day].removeAt(attrsequence);
      notifyListeners();
      final response = await http.delete(url);
    } catch (error) {
      throw error;
    }
  }
```

Fig B-15 Remove Single Attraction from the list of the day in “Trip Detail” screen

```
Future<void> addSingleAttraction(String tripID, int day, int attrsequence,
      String attrid, String userId) async {
    final url = Uri.parse(
        "https://signin-example-63bd3-default-rtdb.firebaseio.com/trip/$userId/$tripID.json");
    try {
      Trip targetedTrip =
          _tripsList.firstWhere((trip) => trip.tripID == tripID);
      targetedTrip.tripSchedule[day].insert(attrsequence, attrid);
      notifyListeners();
      final response = await http.patch(url,
          body: json.encode({"tripSchedule": targetedTrip.tripSchedule}));
    } catch (error) {
      throw error;
    }
  }
```

Fig B-16 Add Single Attraction to the list of the day in “Trip Detail” screen

```
static Future<void> buildRoute(Trip curTrip, BuildContext ctx) async {
    List<List<String>> newSchedule = [];
    List<List<String>> oldSchedule = curTrip.tripSchedule;
    double startLat = curTrip.hotel_latitude;
    double startLong = curTrip.hotel_longitute;
    Attraction hotel = Attraction(
        attrDesc: "",
        attrId: "Accomodation",
        attrName: "Accomodation",
        fee: 0.0,
        url: "",
        city: '',
        rate: 0.0,
        state: "",
        endTime: "",
        startTime: "",
        tel: "",
        longitute: startLong,
        latitude: startLat,
        createdAt: DateTime.now(),
        createdBy: "",
        categoryList: []);

    List<Attraction> attrs = [hotel];
    List<Attraction> FandB = [];
```



```
for (int i = 0; i < oldSchedule.length; i++) {
  for (int j = 0; j < oldSchedule[i].length; j++) {
    Attraction buffer = Provider.of<AttractionList>(ctx, listen: false)
        .findByID(oldSchedule[i][j]);
    if (buffer.categoryList.contains("c13")) {
      FandB.add(buffer);
    } else {
      attrs.add(buffer);
    }
  }
}
//print("Length of attrs = ${attrs.length}");
//print("Length of F And B = ${FandB.length}");
```

```
List<Attraction> combineAll = [];
combineAll = [...attrs];
combineAll.addAll(FandB);
```

```
/*
for(int i = 0;i<combineAll.length;i++)
{
  print(combineAll[i].attrName);
}
*/
```

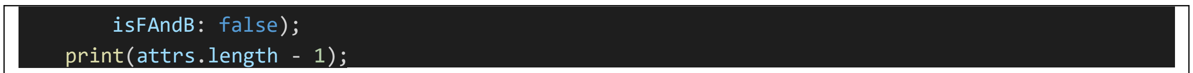
```
List<List<double>> matrix = [];
```

```
for (int i = 0; i < combineAll.length; i++) {
  List<double> rowBuffer = [];
  for (int j = 0; j < combineAll.length; j++) {
    double distanceBuffer = calculateDistance(combineAll[i].latitude,
        combineAll[i].longitude, combineAll[j].latitude, combineAll[j].longitude);
    rowBuffer.add(distanceBuffer);
    //print("Node ${i}${j} = ${distanceBuffer}");
  }
```

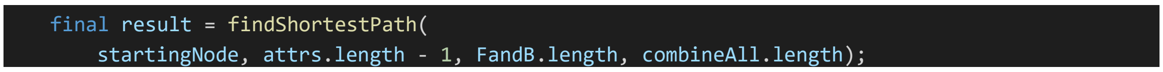
```
  matrix.add(rowBuffer);
}
```

```
//print(matrix);
//print(matrix.length);
//print(matrix[0].length);
```

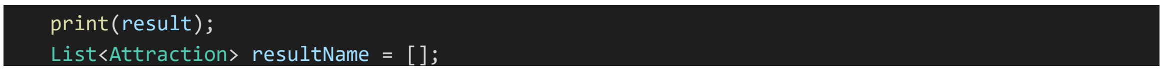
```
Node startingNode = Node(
    level: 0,
    matrix_reduced: matrix,
    path: [0],
    totalDistance: 0.0,
    vertex: 0,
    numAttrsLeave: attrs.length - 1,
    numFAndBLeave: FandB.length,
    isAttr: true,
```



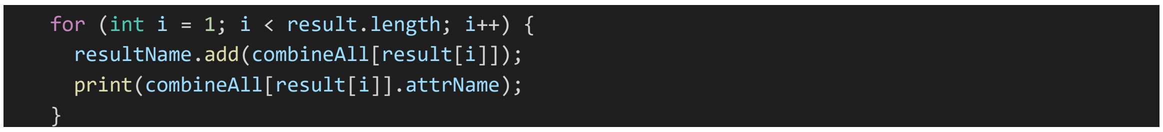
```
        isFAndB: false);
    print(attrs.length - 1);
```



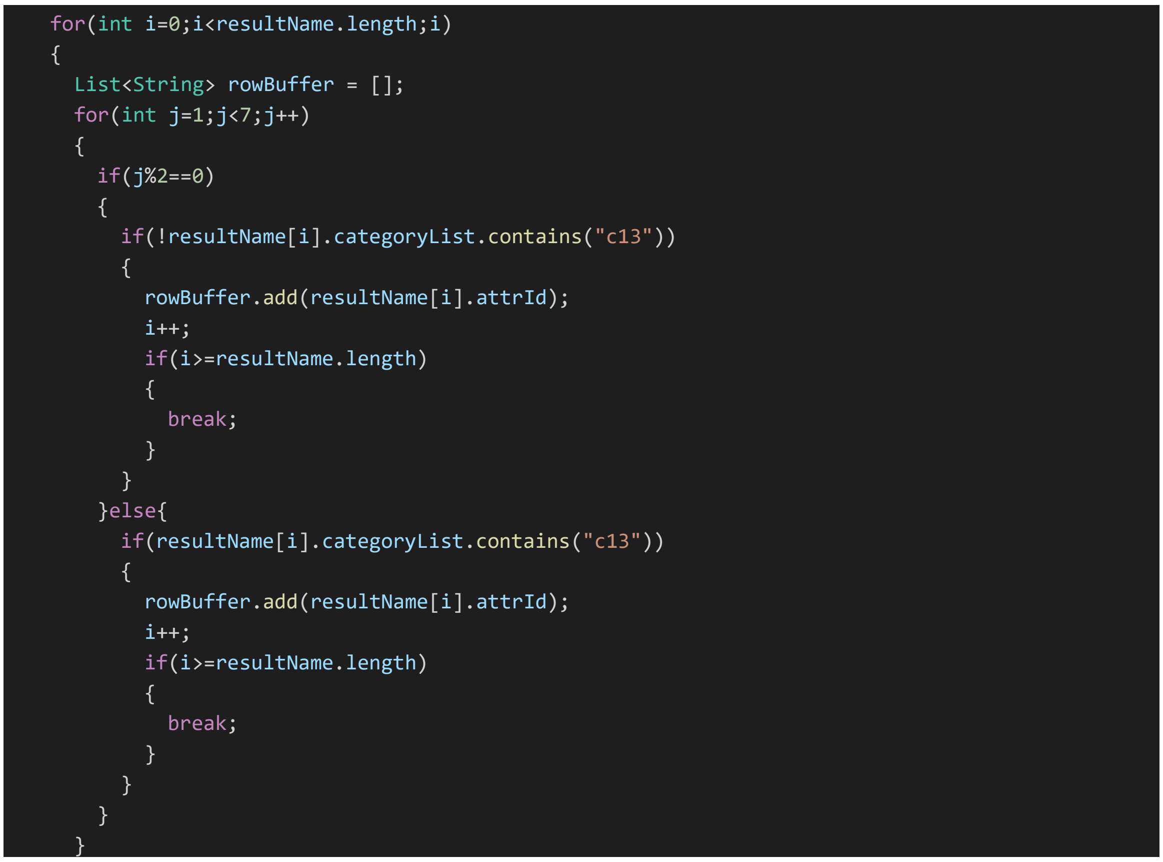
```
    final result = findShortestPath(
        startingNode, attrs.length - 1, FandB.length, combineAll.length);
```



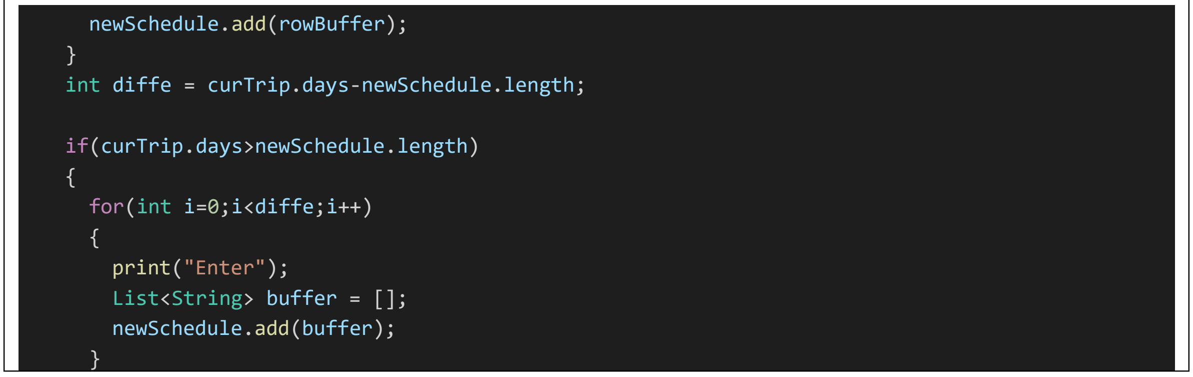
```
    print(result);
    List<Attraction> resultName = [];
```



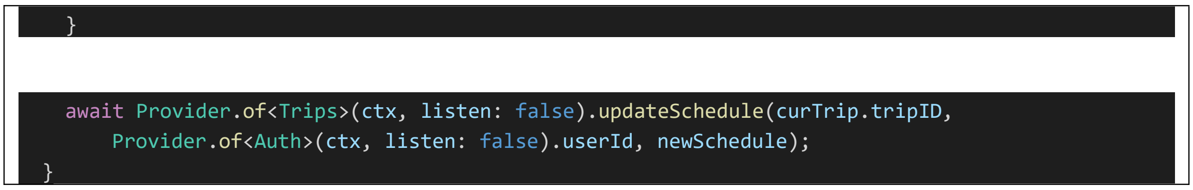
```
    for (int i = 1; i < result.length; i++) {
      resultName.add(combineAll[result[i]]);
      print(combineAll[result[i]].attrName);
    }
```



```
    for(int i=0;i<resultName.length;i)
    {
      List<String> rowBuffer = [];
      for(int j=1;j<7;j++)
      {
        if(j%2==0)
        {
          if(!resultName[i].categoryList.contains("c13"))
          {
            rowBuffer.add(resultName[i].attrId);
            i++;
            if(i>=resultName.length)
            {
              break;
            }
          }
        }else{
          if(resultName[i].categoryList.contains("c13"))
          {
            rowBuffer.add(resultName[i].attrId);
            i++;
            if(i>=resultName.length)
            {
              break;
            }
          }
        }
      }
```

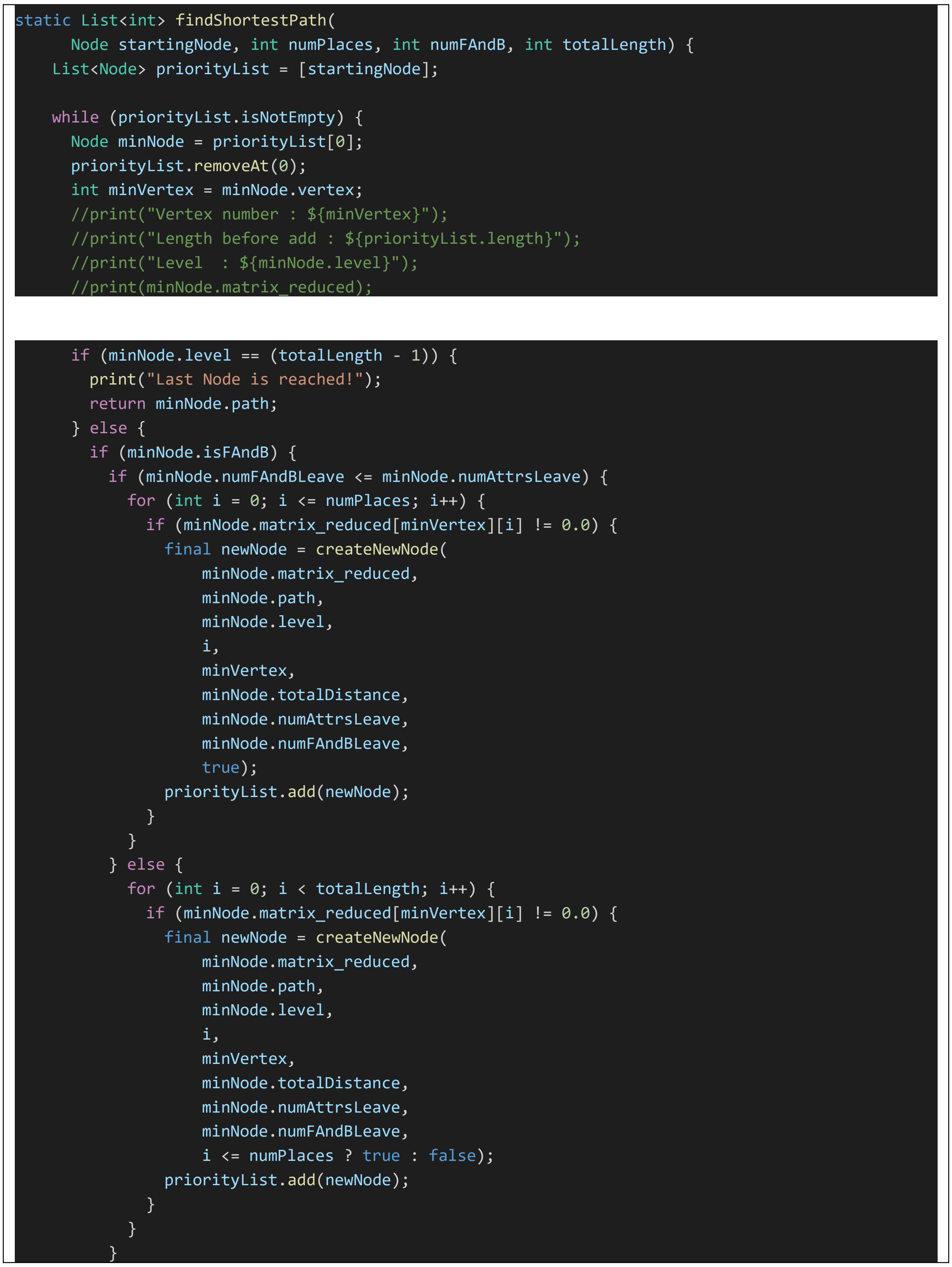


```
      newSchedule.add(rowBuffer);
    }
    int diffe = curTrip.days-newSchedule.length;

    if(curTrip.days>newSchedule.length)
    {
      for(int i=0;i<diffe;i++)
      {
        print("Enter");
        List<String> buffer = [];
        newSchedule.add(buffer);
      }
```

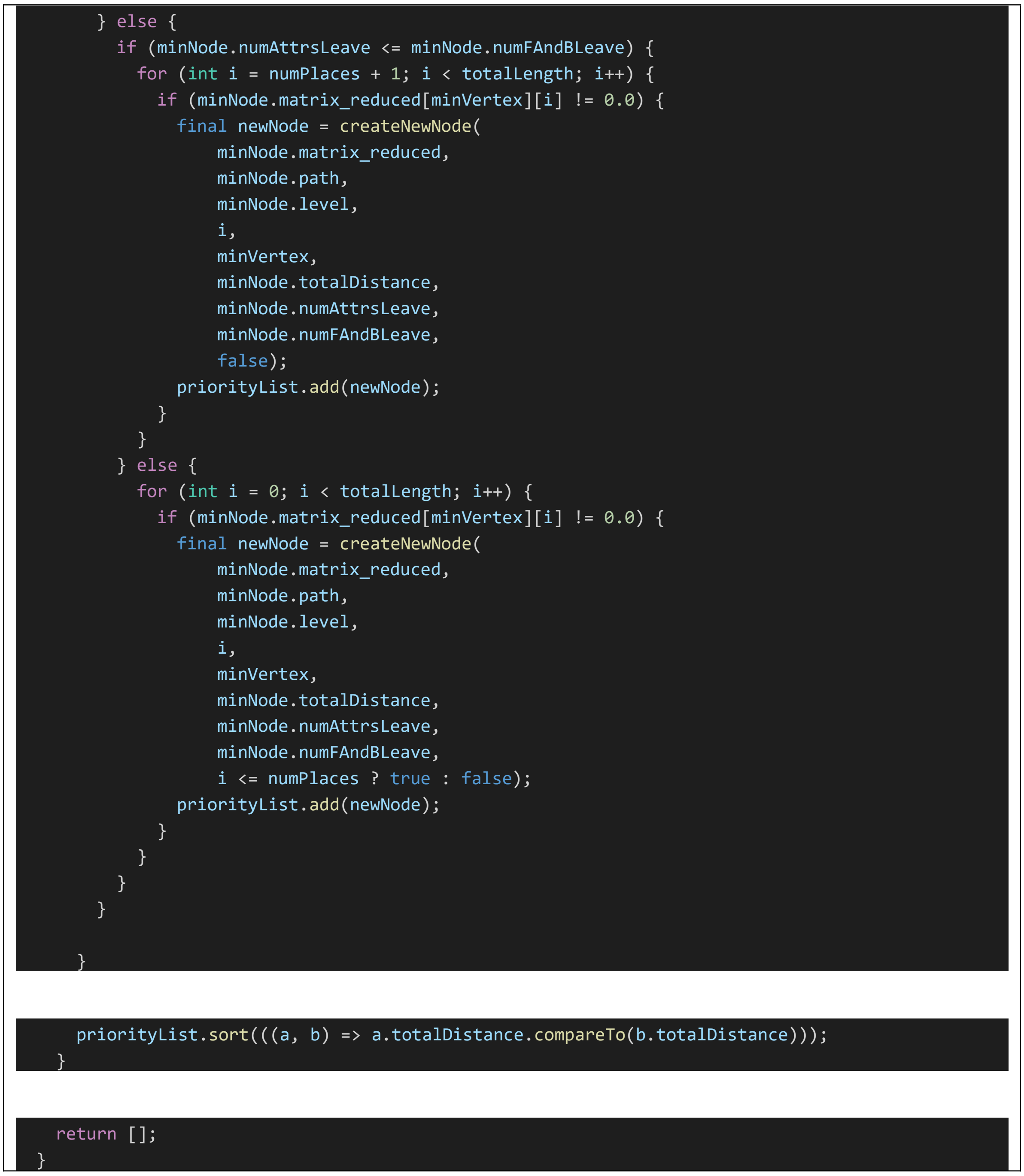


```
    }

    await Provider.of<Trips>(ctx, listen: false).updateSchedule(curTrip.tripID,
        Provider.of<Auth>(ctx, listen: false).userId, newSchedule);
  }
```

Fig B-17 Smart Routing function

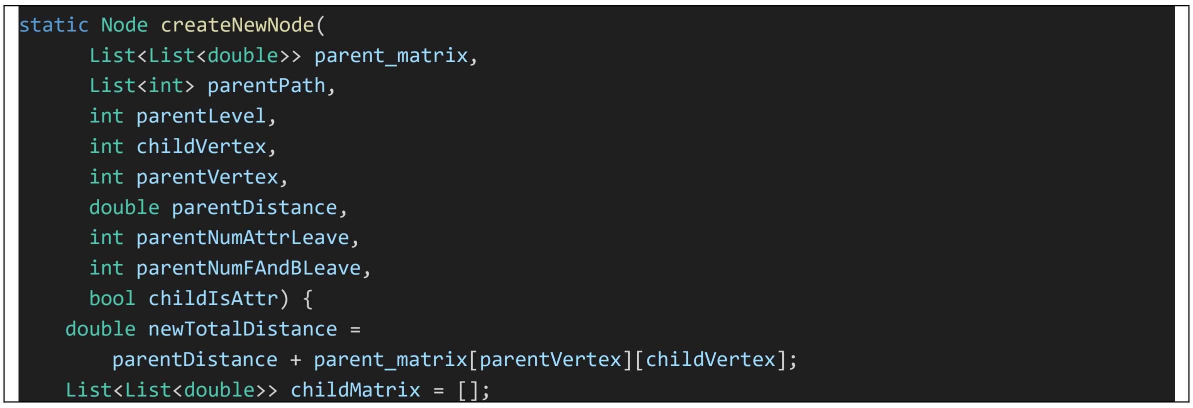


```
static List<int> findShortestPath(
      Node startingNode, int numPlaces, int numFAndB, int totalLength) {
    List<Node> priorityList = [startingNode];

    while (priorityList.isNotEmpty) {
      Node minNode = priorityList[0];
      priorityList.removeAt(0);
      int minVertex = minNode.vertex;
      //print("Vertex number : ${minVertex}");
      //print("Length before add : ${priorityList.length}");
      //print("Level  : ${minNode.level}");
      //print(minNode.matrix_reduced);

      if (minNode.level == (totalLength - 1)) {
        print("Last Node is reached!");
        return minNode.path;
      } else {
        if (minNode.isFAndB) {
          if (minNode.numFAndBLeave <= minNode.numAttrsLeave) {
            for (int i = 0; i <= numPlaces; i++) {
              if (minNode.matrix_reduced[minVertex][i] != 0.0) {
                final newNode = createNewNode(
                    minNode.matrix_reduced,
                    minNode.path,
                    minNode.level,
                    i,
                    minVertex,
                    minNode.totalDistance,
                    minNode.numAttrsLeave,
                    minNode.numFAndBLeave,
                    true);
                priorityList.add(newNode);
              }
            }
          } else {
            for (int i = 0; i < totalLength; i++) {
              if (minNode.matrix_reduced[minVertex][i] != 0.0) {
                final newNode = createNewNode(
                    minNode.matrix_reduced,
                    minNode.path,
                    minNode.level,
                    i,
                    minVertex,
                    minNode.totalDistance,
                    minNode.numAttrsLeave,
                    minNode.numFAndBLeave,
                    i <= numPlaces ? true : false);
                priorityList.add(newNode);
              }
            }
          }
```

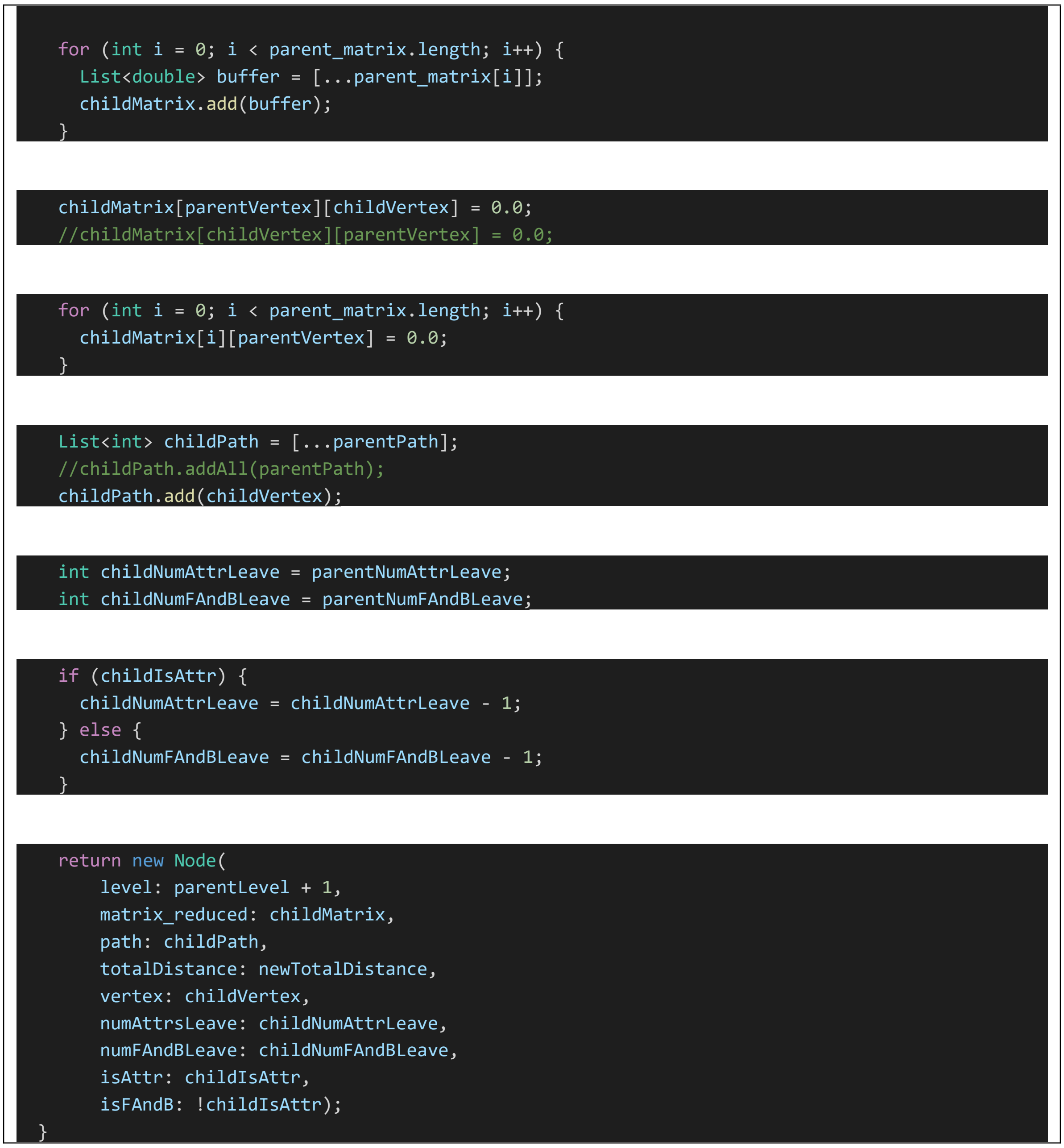


```
        } else {
          if (minNode.numAttrsLeave <= minNode.numFAndBLeave) {
            for (int i = numPlaces + 1; i < totalLength; i++) {
              if (minNode.matrix_reduced[minVertex][i] != 0.0) {
                final newNode = createNewNode(
                    minNode.matrix_reduced,
                    minNode.path,
                    minNode.level,
                    i,
                    minVertex,
                    minNode.totalDistance,
                    minNode.numAttrsLeave,
                    minNode.numFAndBLeave,
                    false);
                priorityList.add(newNode);
              }
            }
          } else {
            for (int i = 0; i < totalLength; i++) {
              if (minNode.matrix_reduced[minVertex][i] != 0.0) {
                final newNode = createNewNode(
                    minNode.matrix_reduced,
                    minNode.path,
                    minNode.level,
                    i,
                    minVertex,
                    minNode.totalDistance,
                    minNode.numAttrsLeave,
                    minNode.numFAndBLeave,
                    i <= numPlaces ? true : false);
                priorityList.add(newNode);
              }
            }
          }
        }

      }

      priorityList.sort(((a, b) => a.totalDistance.compareTo(b.totalDistance)));
    }

    return [];
  }
```

Fig B-18 findShortestPath function

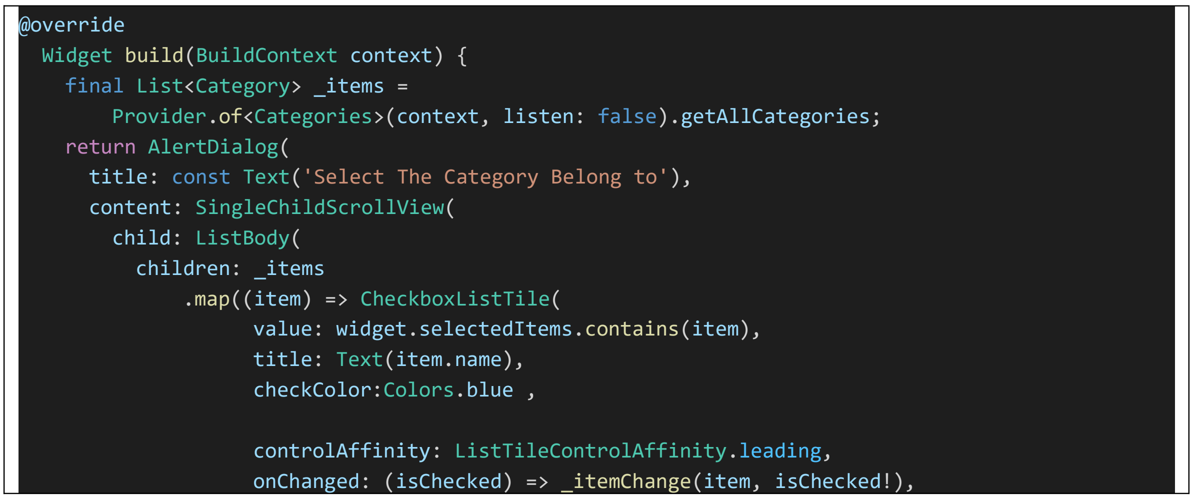


```
static Node createNewNode(
      List<List<double>> parent_matrix,
      List<int> parentPath,
      int parentLevel,
      int childVertex,
      int parentVertex,
      double parentDistance,
      int parentNumAttrLeave,
      int parentNumFAndBLeave,
      bool childIsAttr) {
    double newTotalDistance =
        parentDistance + parent_matrix[parentVertex][childVertex];
    List<List<double>> childMatrix = [];
```

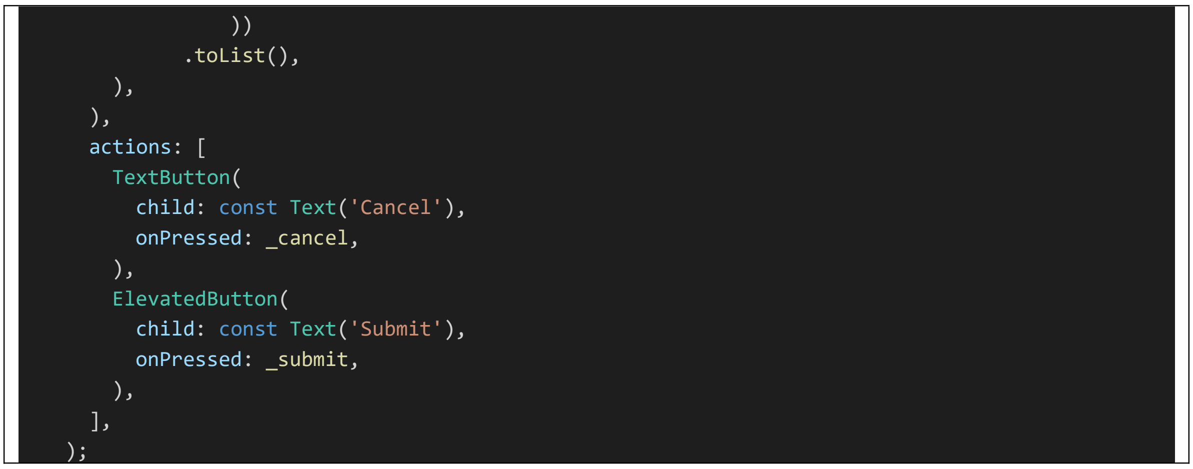


```
    for (int i = 0; i < parent_matrix.length; i++) {
      List<double> buffer = [...parent_matrix[i]];
      childMatrix.add(buffer);
    }

    childMatrix[parentVertex][childVertex] = 0.0;
    //childMatrix[childVertex][parentVertex] = 0.0;

    for (int i = 0; i < parent_matrix.length; i++) {
      childMatrix[i][parentVertex] = 0.0;
    }

    List<int> childPath = [...parentPath];
    //childPath.addAll(parentPath);
    childPath.add(childVertex);

    int childNumAttrLeave = parentNumAttrLeave;
    int childNumFAndBLeave = parentNumFAndBLeave;

    if (childIsAttr) {
      childNumAttrLeave = childNumAttrLeave - 1;
    } else {
      childNumFAndBLeave = childNumFAndBLeave - 1;
    }

    return new Node(
        level: parentLevel + 1,
        matrix_reduced: childMatrix,
        path: childPath,
        totalDistance: newTotalDistance,
        vertex: childVertex,
        numAttrsLeave: childNumAttrLeave,
        numFAndBLeave: childNumFAndBLeave,
        isAttr: childIsAttr,
        isFAndB: !childIsAttr);
  }
```

Fig 6-19 Create New Node function



```
@override
  Widget build(BuildContext context) {
    final List<Category> _items =
        Provider.of<Categories>(context, listen: false).getAllCategories;
    return AlertDialog(
      title: const Text('Select The Category Belong to'),
      content: SingleChildScrollView(
        child: ListBody(
          children: _items
              .map((item) => CheckboxListTile(
                    value: widget.selectedItems.contains(item),
                    title: Text(item.name),
                    checkColor:Colors.blue ,

                    controlAffinity: ListTileControlAffinity.leading,
                    onChanged: (isChecked) => _itemChange(item, isChecked!),
```



```
                  ))
              .toList(),
        ),
      ),
      actions: [
        TextButton(
          child: const Text('Cancel'),
          onPressed: _cancel,
        ),
        ElevatedButton(
          child: const Text('Submit'),
          onPressed: _submit,
        ),
      ],
    );
```

Fig B-20 Multi Select Function to Display the Filter Selection

Appendix C

Survey Response

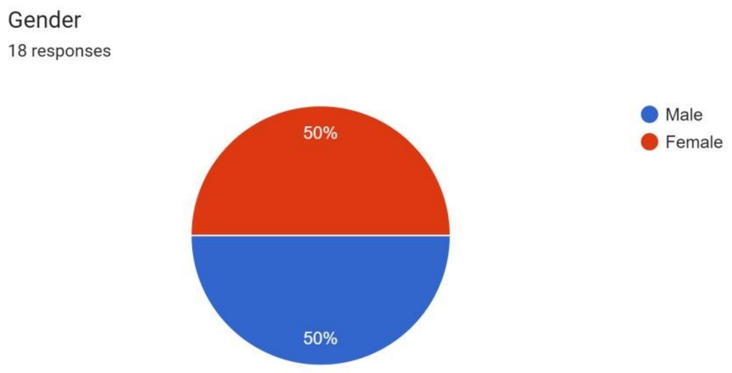


Fig C-1 Gender Classification

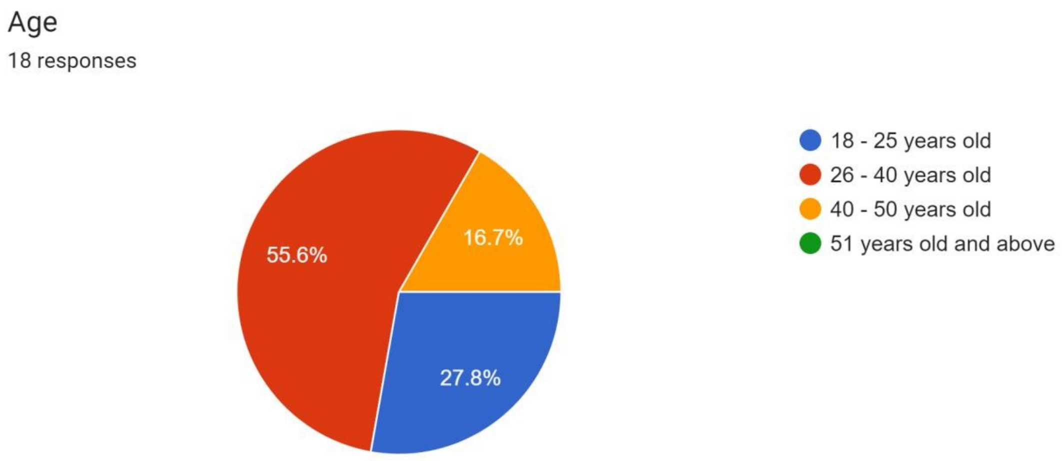


Fig C-2 Age Classification

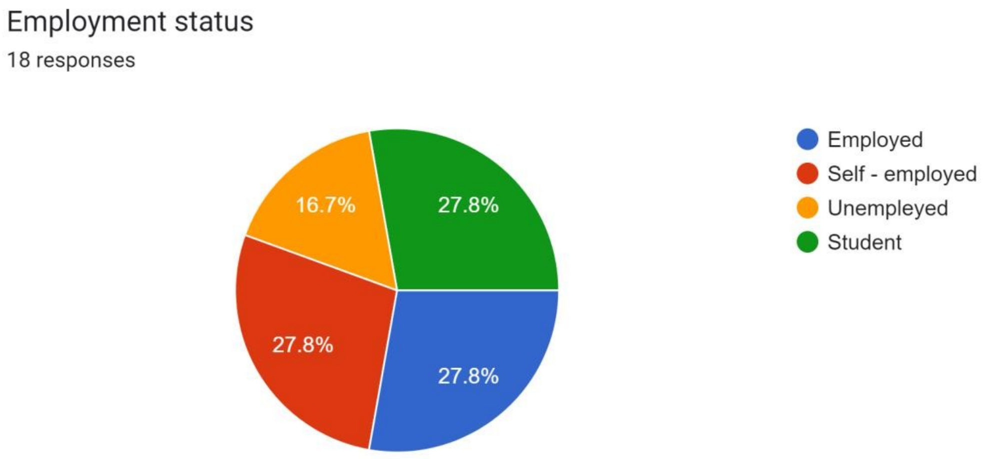


Fig C-3 Empleyment Classification

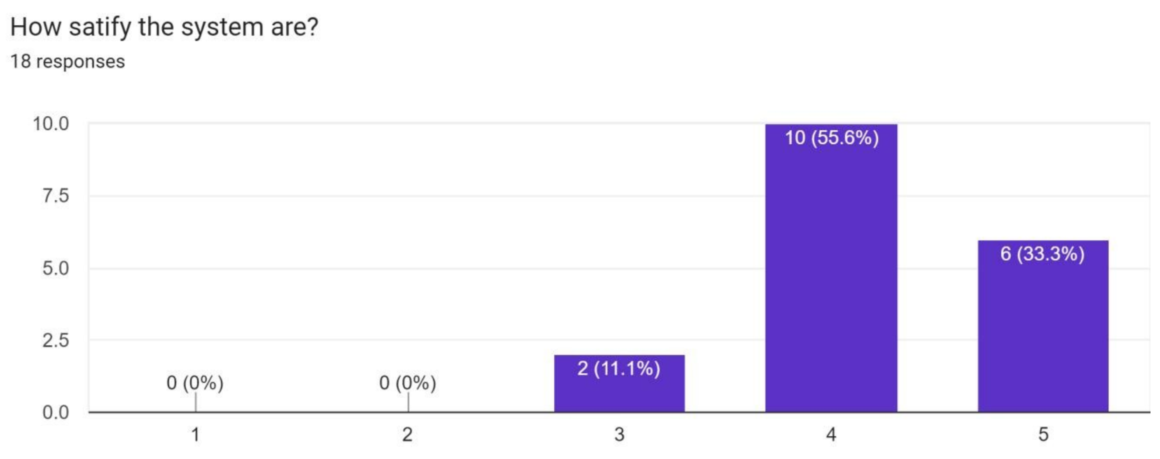


Fig C-4 Sanctification toward Itinerary Planner Mobile Application

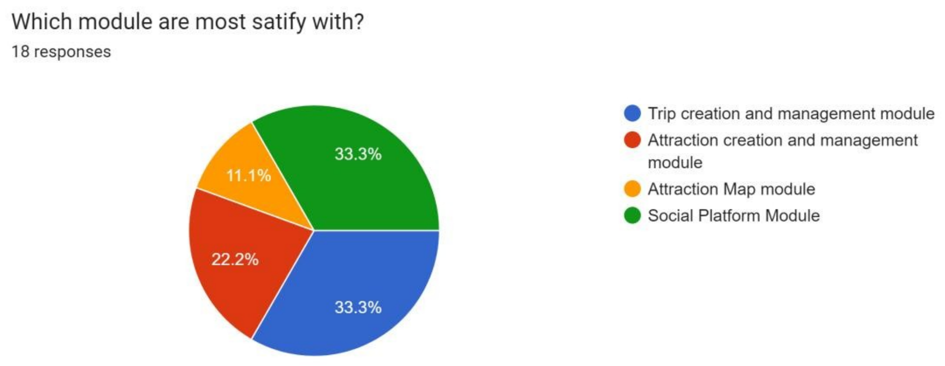


Fig C-5 Stratification in Each Module

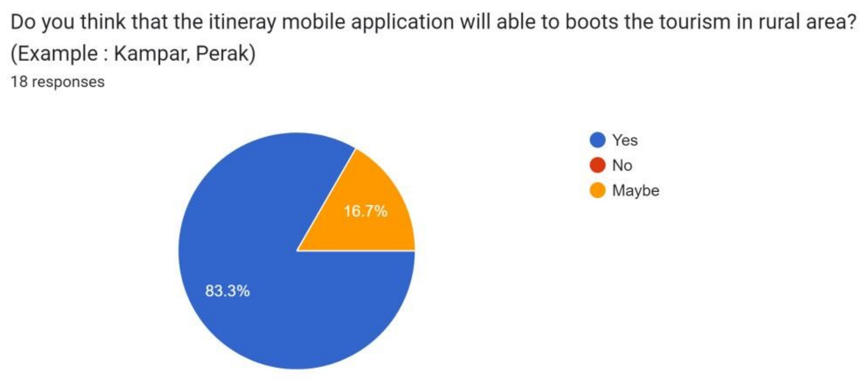
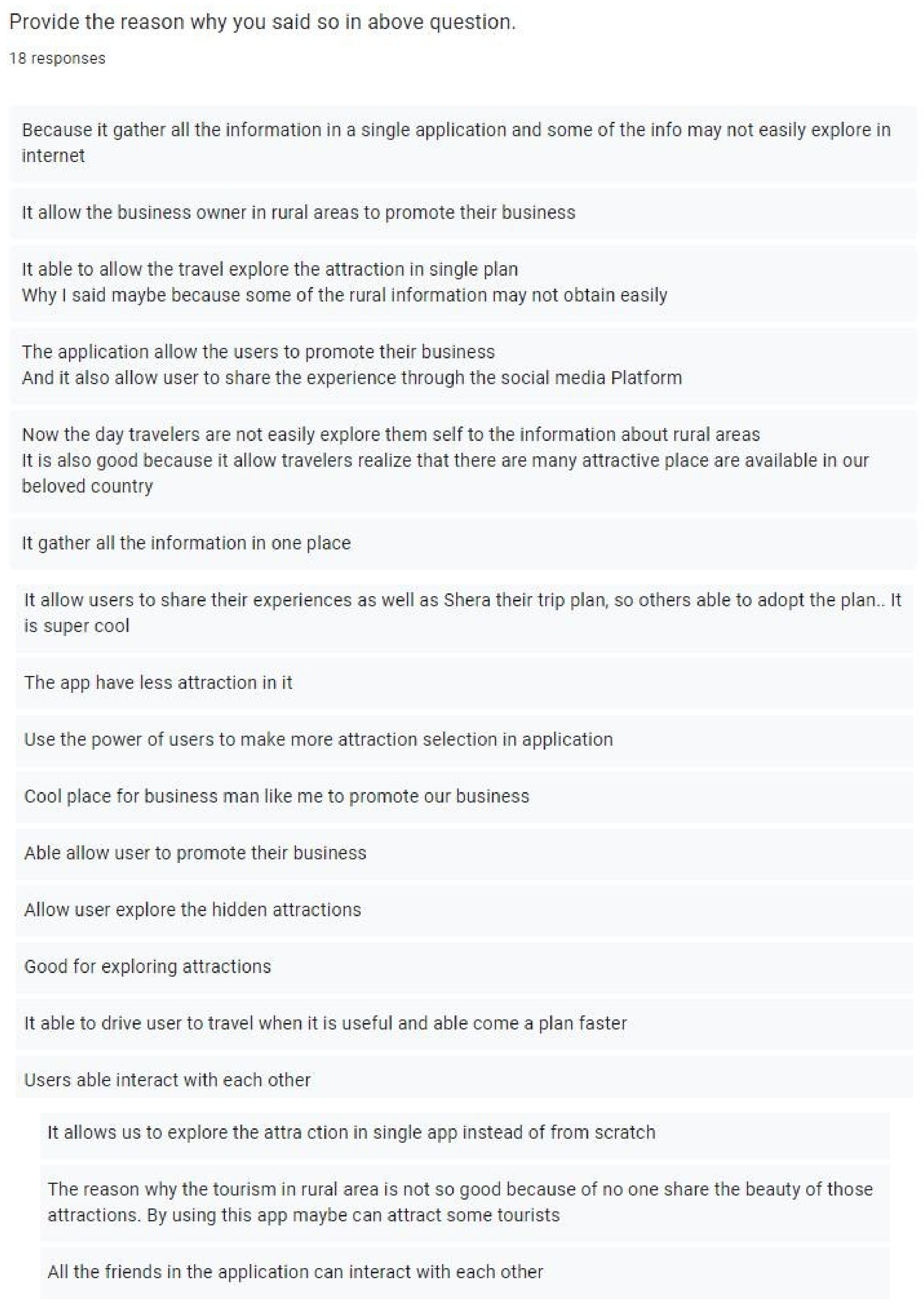


Fig C-6 Will Itinerary Planner Mobile Application Help Tourism in Rural Area?



Provide the reason why you said so in above question.

18 responses

Because it gather all the information in a single application and some of the info may not easily explore in internet

It allow the business owner in rural areas to promote their business

It able to allow the travel explore the attraction in single plan
Why I said maybe because some of the rural information may not obtain easily

The application allow the users to promote their business
And it also allow user to share the experience through the social media Platform

Now the day travelers are not easily explore them self to the information about rural areas
It is also good because it allow travelers realize that there are many attractive place are available in our beloved country

It gather all the information in one place

It allow users to share their experiences as well as Shera their trip plan, so others able to adopt the plan.. It is super cool

The app have less attraction in it

Use the power of users to make more attraction selection in application

Cool place for business man like me to promote our business

Able allow user to promote their business

Allow user explore the hidden attractions

Good for exploring attractions

It able to drive user to travel when it is useful and able come a plan faster

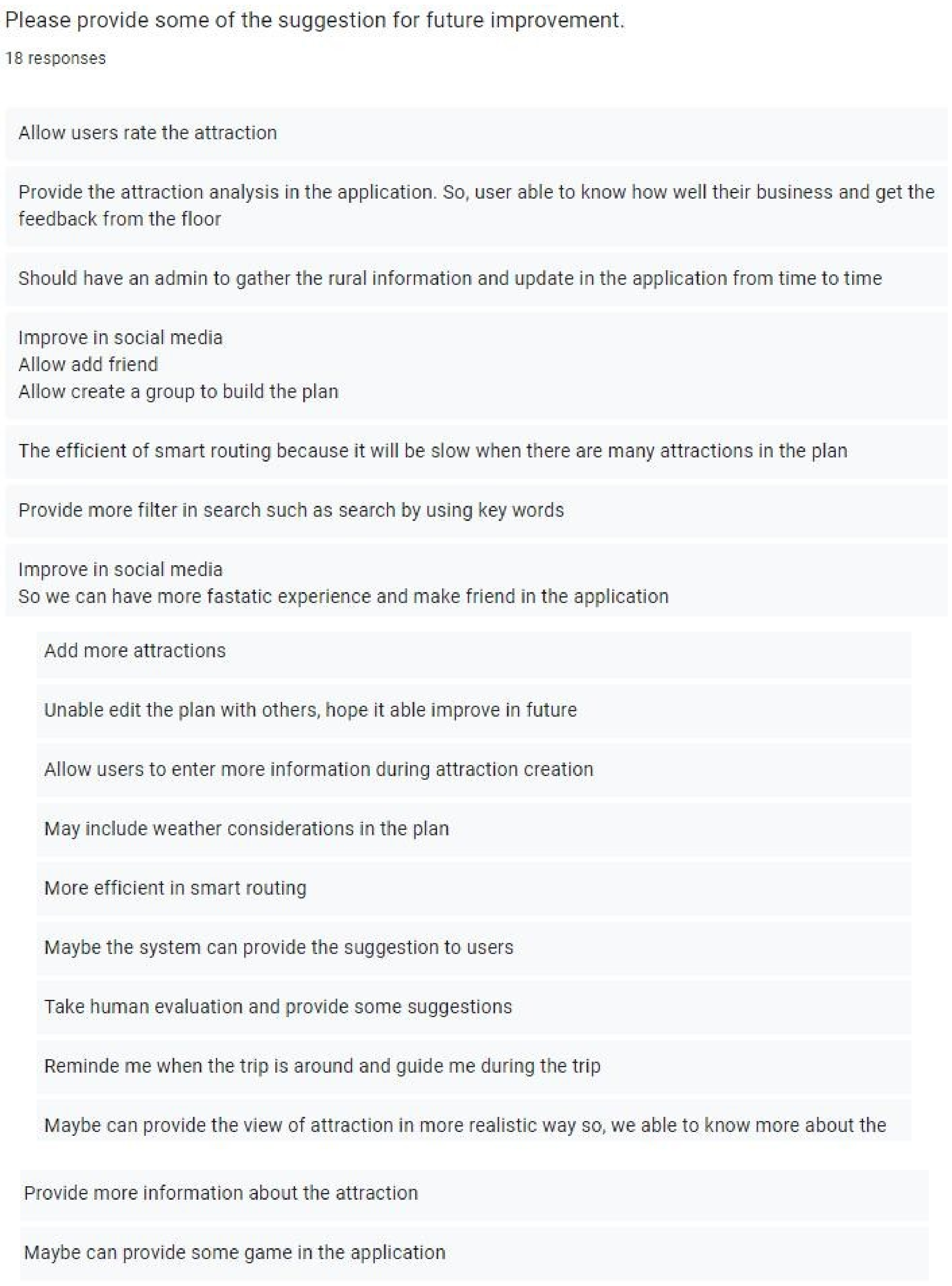
Users able interact with each other

It allows us to explore the attra ction in single app instead of from scratch

The reason why the tourism in rural area is not so good because of no one share the beauty of those attractions. By using this app maybe can attract some tourists

All the friends in the application can interact with each other

Fig C-7 Reason about the select in Fig C-6



Please provide some of the suggestion for future improvement.

18 responses

Allow users rate the attraction

Provide the attraction analysis in the application. So, user able to know how well their business and get the feedback from the floor

Should have an admin to gather the rural information and update in the application from time to time

Improve in social media
Allow add friend
Allow create a group to build the plan

The efficient of smart routing because it will be slow when there are many attractions in the plan

Provide more filter in search such as search by using key words

Improve in social media
So we can have more fastatic experience and make friend in the application

Add more attractions

Unable edit the plan with others, hope it able improve in future

Allow users to enter more information during attraction creation

May include weather considerations in the plan

More efficient in smart routing

Maybe the system can provide the suggestion to users

Take human evaluation and provide some suggestions

Reminde me when the trip is around and guide me during the trip

Maybe can provide the view of attraction in more realistic way so, we able to know more about the

Provide more information about the attraction

Maybe can provide some game in the application

Fig C-8 Suggestion for Future Improvement

Appendix D

FINAL YEAR PROJECT WEEKLY REPORT

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3Y3	**Study week no.: Week 1**
Student Name & ID: Loh Wai Leong (18ACB06900)	
Supervisor: Dr. Liew Soung Yue	
Project Title: Itinerary Planner Mobile Application	

1. WORK DONE

- Revised the FYP1

2. WORK TO BE DONE

- Build out the plan to complete FYP2

3. PROBLEMS ENCOUNTERED

- **-**

4. SELF EVALUATION OF THE PROGRESS

Satisfied

[illegible]

Supervisor's signature

[illegible]

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3Y3	**Study week no.: Week 2**
Student Name & ID: Loh Wai Leong (18ACB06900)	
Supervisor: Dr. Liew Soung Yue	
Project Title: Itinerary Planner Mobile Application	

1. WORK DONE

- Plan to complete FYP2 is done

2. WORK TO BE DONE

- Brainstorm the idea how to build the social platform function
- Do research on social platform

3. PROBLEMS ENCOUNTERED

- -

4. SELF EVALUATION OF THE PROGRESS

Satisfied

[illegible]

Supervisor's signature

[illegible]

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3Y3	**Study week no.: Week 3**
Student Name & ID: Loh Wai Leong (18ACB06900)	
Supervisor: Dr. Liew Soung Yue	
Project Title: Itinerary Planner Mobile Application	

1. WORK DONE

- Create post function
- Display all the post

2. WORK TO BE DONE

- Continue complete Social Platform module

3. PROBLEMS ENCOUNTERED

- Encounter when build the form by using flutter

4. SELF EVALUATION OF THE PROGRESS

Satisfied

[illegible]

Supervisor's signature

[illegible]

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3Y3	**Study week no.: Week 4**
Student Name & ID: Loh Wai Leong (18ACB06900)	
Supervisor: Dr. Liew Soung Yue	
Project Title: Itinerary Planner Mobile Application	

1. WORK DONE

- Able to like the post
- Able to make a comment on the post
- Able to edit the existing post

2. WORK TO BE DONE

- Decorate the social platform module

3. PROBLEMS ENCOUNTERED

-

4. SELF EVALUATION OF THE PROGRESS

Satisfied

[illegible]

Supervisor's signature

[illegible]

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3Y3	**Study week no.: Week 5**
Student Name & ID: Loh Wai Leong (18ACB06900)	
Supervisor: Dr. Liew Soung Yue	
Project Title: Itinerary Planner Mobile Application	

1. WORK DONE

- Complete Social Platform Module

2. WORK TO BE DONE

- Enable user to upload image to Google Firebase Storage

3. PROBLEMS ENCOUNTERED

-

4. SELF EVALUATION OF THE PROGRESS

Satisfied

[illegible]

Supervisor's signature

[illegible]

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3Y3	**Study week no.: Week 6**
Student Name & ID: Loh Wai Leong (18ACB06900)	
Supervisor: Dr. Liew Soung Yue	
Project Title: Itinerary Planner Mobile Application	

1. WORK DONE ● Users able to upload image to Google Firebase Storage
2. WORK TO BE DONE ● Update "Create Trip" form and allow users to upload profile picture
3. PROBLEMS ENCOUNTERED ● Configuration Google Firebase Storage
4. SELF EVALUATION OF THE PROGRESS **Satisfied**

[illegible] [illegible]

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3Y3	**Study week no.: Week 7**
Student Name & ID: Loh Wai Leong (18ACB06900)	
Supervisor: Dr. Liew Soung Yue	
Project Title: Itinerary Planner Mobile Application	

1. WORK DONE

- Create markers with icons and show on the map

2. WORK TO BE DONE

- Attraction Map Module

3. PROBLEMS ENCOUNTERED

-

4. SELF EVALUATION OF THE PROGRESS

Satisfied

[illegible]

Supervisor's signature

[illegible]

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3Y3	**Study week no.: Week 8**
Student Name & ID: Loh Wai Leong (18ACB06900)	
Supervisor: Dr. Liew Soung Yue	
Project Title: Itinerary Planner Mobile Application	

1. WORK DONE

- Map in Attraction Map screen with all the attraction markers

2. WORK TO BE DONE

- Filter in Attraction Map Module

3. PROBLEMS ENCOUNTERED

- Google Firebase Storage out of quota
- Unable retrieve image from Google Firebase Storage

4. SELF EVALUATION OF THE PROGRESS

Not so Satisfied

[illegible] [illegible]

Supervisor's signature Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3Y3	**Study week no.: Week 9**
Student Name & ID: Loh Wai Leong (18ACB06900)	
Supervisor: Dr. Liew Soung Yue	
Project Title: Itinerary Planner Mobile Application	

1. WORK DONE

- Complete Attraction Map Module

2. WORK TO BE DONE

- Allow users to add new attraction
- Allow user to edit existing attraction

3. PROBLEMS ENCOUNTERED

-

4. SELF EVALUATION OF THE PROGRESS

Satisfied

[illegible]

Supervisor's signature

[illegible]

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3Y3	**Study week no.: Week 10**
Student Name & ID: Loh Wai Leong (18ACB06900)	
Supervisor: Dr. Liew Soung Yue	
Project Title: Itinerary Planner Mobile Application	

1. WORK DONE

- Done Manage Attraction Module

2. WORK TO BE DONE

- Perform research on how to build smart routing

3. PROBLEMS ENCOUNTERED

-

4. SELF EVALUATION OF THE PROGRESS

Satisfied

[illegible]

Supervisor's signature

[illegible]

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3Y3	**Study week no.: Week 11**
Student Name & ID: Loh Wai Leong (18ACB06900)	
Supervisor: Dr. Liew Soung Yue	
Project Title: Itinerary Planner Mobile Application	

1. WORK DONE

- Chapter 1, Chapter 2 and half of Chapter 3 in the report

2. WORK TO BE DONE

- Build Smart routing function

3. PROBLEMS ENCOUNTERED

- The idea of smart routing function not work as expected

4. SELF EVALUATION OF THE PROGRESS

Satisfied

[illegible]

Supervisor's signature

[illegible]

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3Y3	**Study week no.: Week 12**
Student Name & ID: Loh Wai Leong (18ACB06900)	
Supervisor: Dr. Liew Soung Yue	
Project Title: Itinerary Planner Mobile Application	

1. WORK DONE

- Complete smart routing function

2. WORK TO BE DONE

- Complete the rest of report

3. PROBLEMS ENCOUNTERED

4. SELF EVALUATION OF THE PROGRESS

Satisfied

[illegible]

Supervisor's signature

[illegible]

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: T3Y3	**Study week no.: Week 13**
Student Name & ID: Loh Wai Leong (18ACB06900)	
Supervisor: Dr. Liew Soung Yue	
Project Title: Itinerary Planner Mobile Application	

1. WORK DONE

- Completed the Report

2. WORK TO BE DONE

- Upload report to Turnitin

3. PROBLEMS ENCOUNTERED

4. SELF EVALUATION OF THE PROGRESS

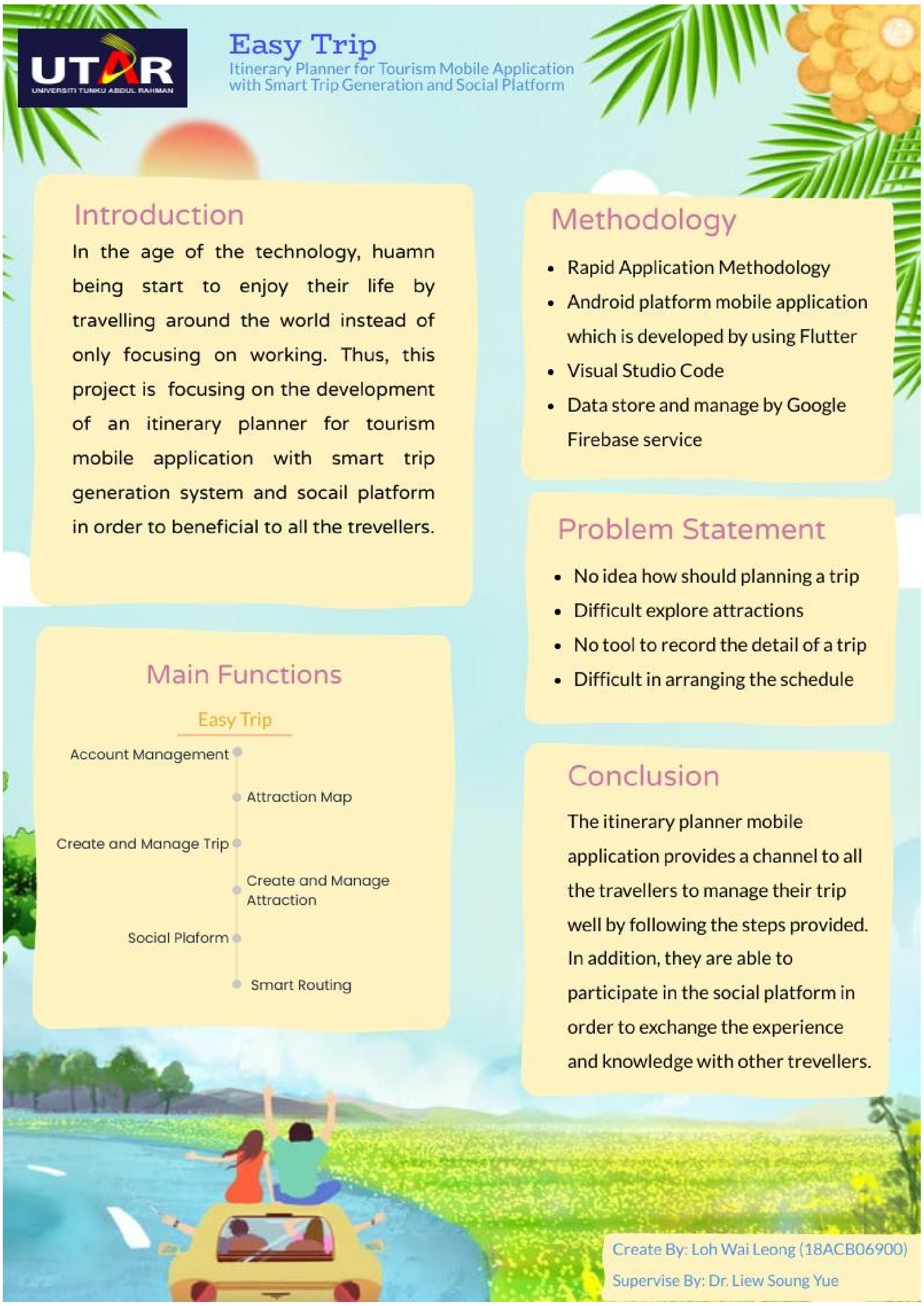
Satisfied

[illegible] [illegible]

Supervisor's signature   Student's signature

Appendix E

Poster



UTAR UNIVERSITI TUNKU ABDUL RAHMAN

Easy Trip

Itinerary Planner for Tourism Mobile Application with Smart Trip Generation and Social Platform

Introduction

In the age of the technology, huamn being start to enjoy their life by travelling around the world instead of only focusing on working. Thus, this project is focusing on the development of an itinerary planner for tourism mobile application with smart trip generation system and socail platform in order to beneficial to all the trevellers.

Methodology

- Rapid Application Methodology
- Android platform mobile application which is developed by using Flutter
- Visual Studio Code
- Data store and manage by Google Firebase service

Problem Statement

- No idea how should planning a trip
- Difficult explore attractions
- No tool to record the detail of a trip
- Difficult in arranging the schedule

Main Functions

Easy Trip

- Account Management
- Attraction Map
- Create and Manage Trip
- Create and Manage Attraction
- Social Plaform
- Smart Routing

Conclusion

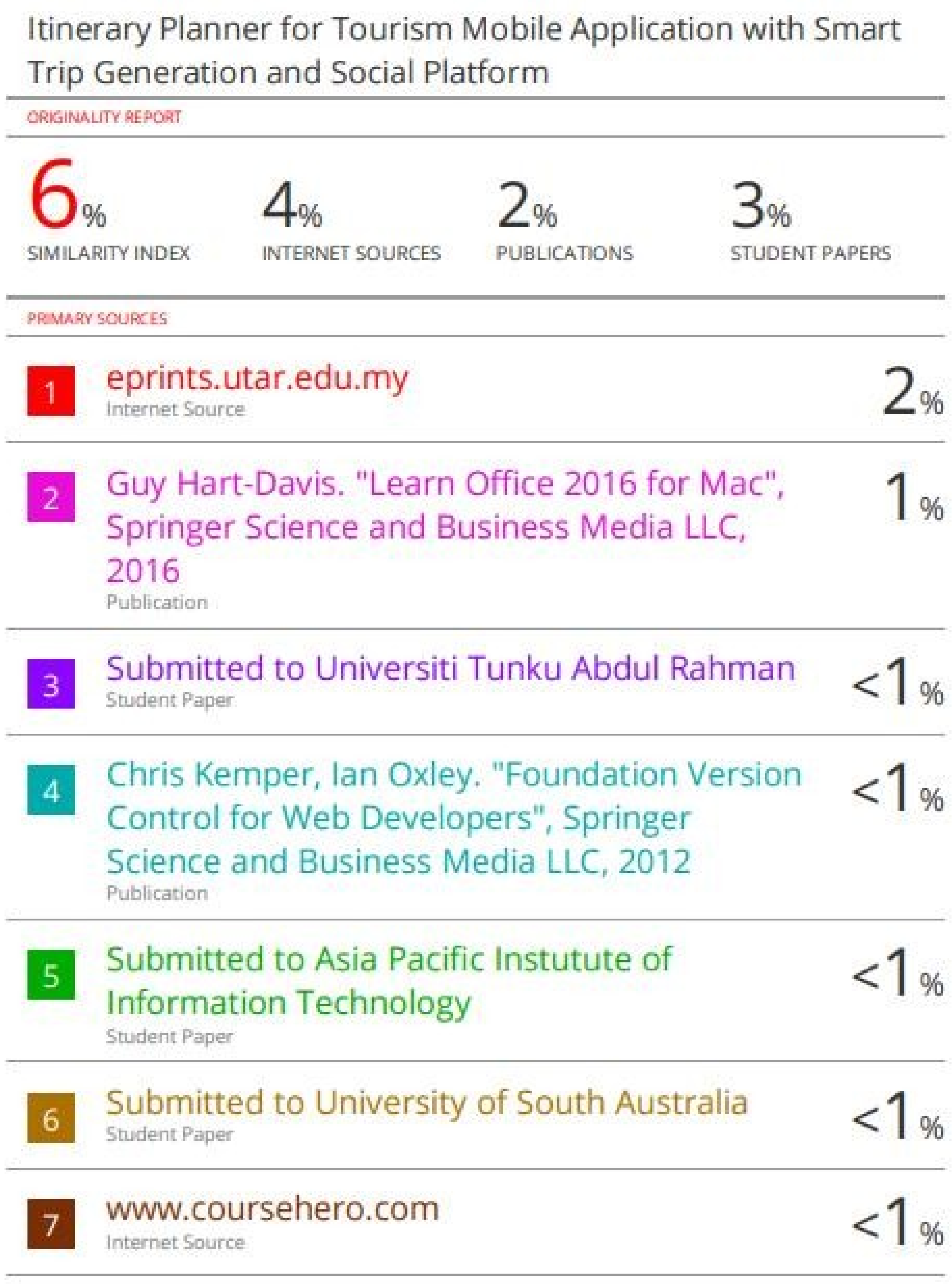
The itinerary planner mobile application provides a channel to all the travellers to manage their trip well by following the steps provided. In addition, they are able to participate in the social platform in order to exchange the experience and knowledge with other trevellers.

Create By: Loh Wai Leong (18ACB06900)

Supervise By: Dr. Liew Soung Yue

Appendix F

Plagiarism check result



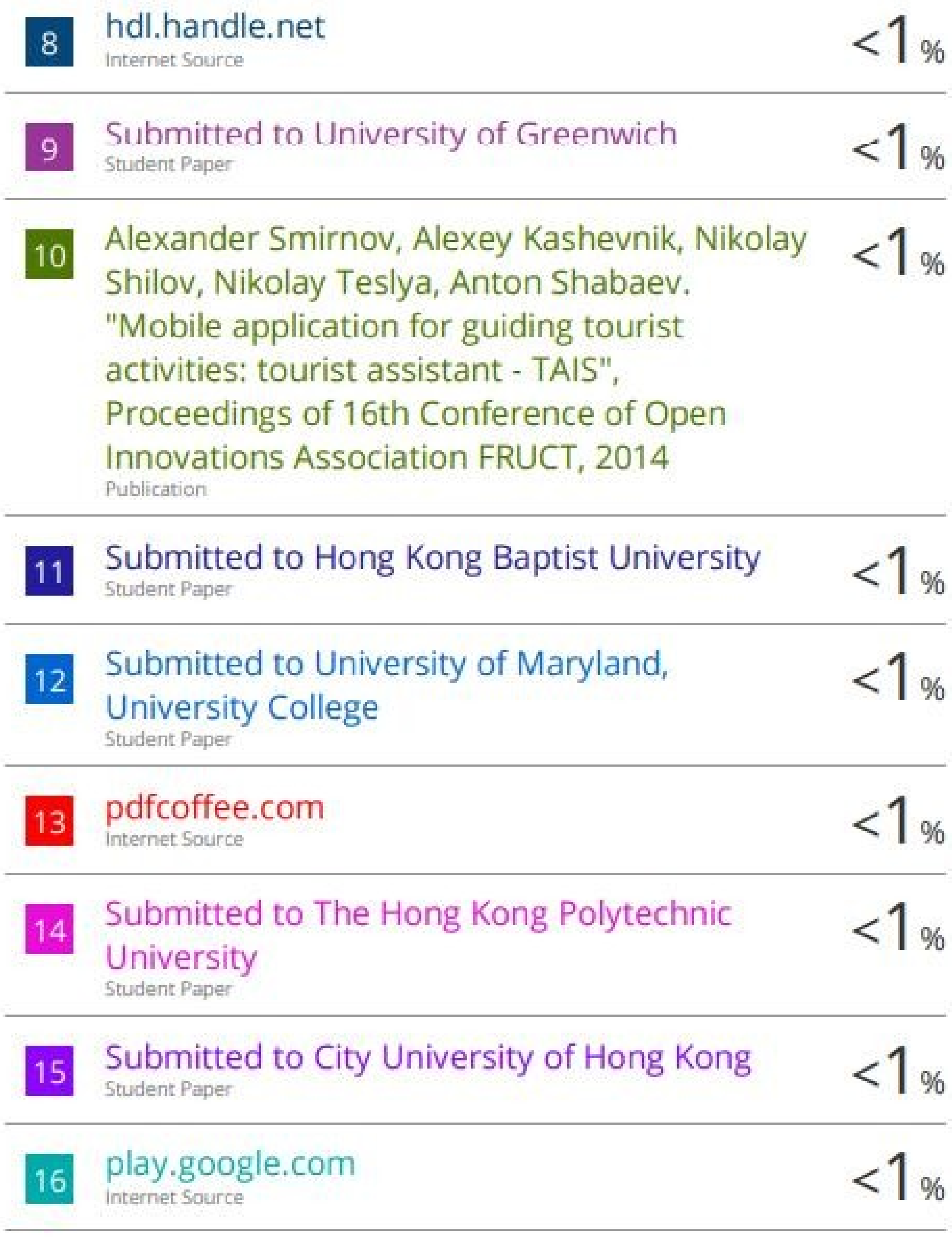
Itinerary Planner for Tourism Mobile Application with Smart Trip Generation and Social Platform

ORIGINALITY REPORT

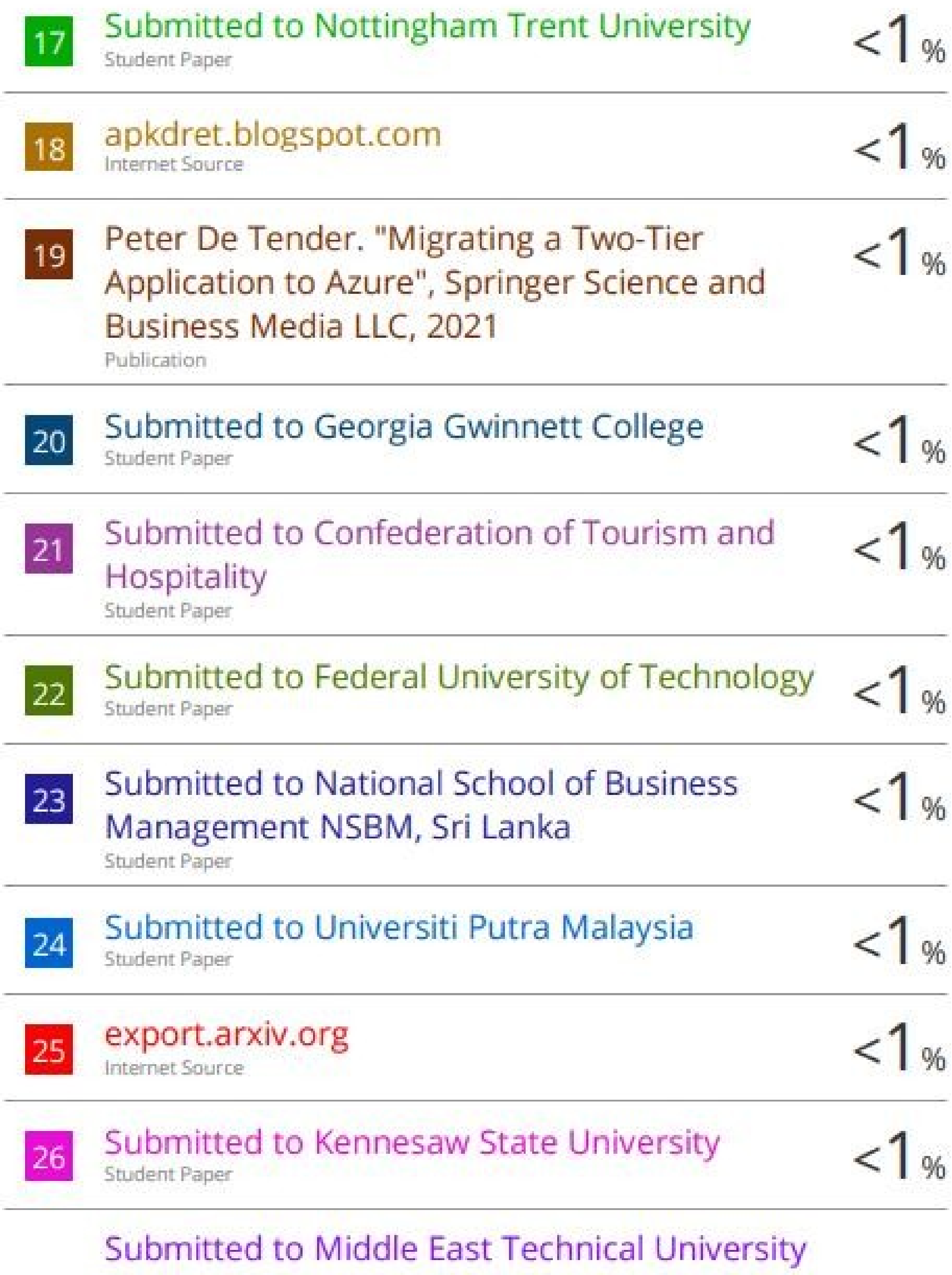
6%	4%	2%	3%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

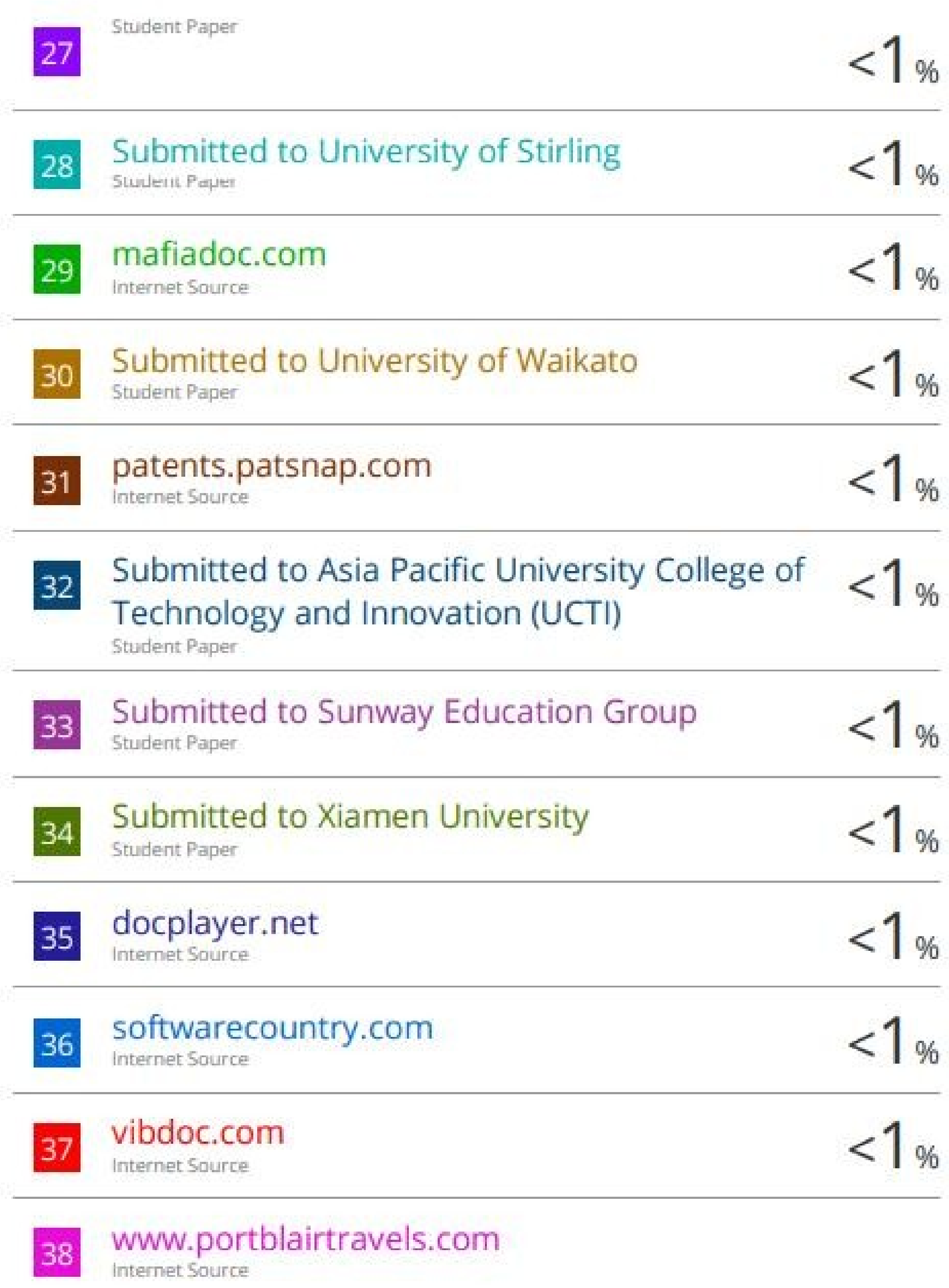
1	eprints.utar.edu.my Internet Source	2%
2	Guy Hart-Davis. "Learn Office 2016 for Mac", Springer Science and Business Media LLC, 2016 Publication	1%
3	Submitted to Universiti Tunku Abdul Rahman Student Paper	<1%
4	Chris Kemper, Ian Oxley. "Foundation Version Control for Web Developers", Springer Science and Business Media LLC, 2012 Publication	<1%
5	Submitted to Asia Pacific Instutute of Information Technology Student Paper	<1%
6	Submitted to University of South Australia Student Paper	<1%
7	www.coursehero.com Internet Source	<1%



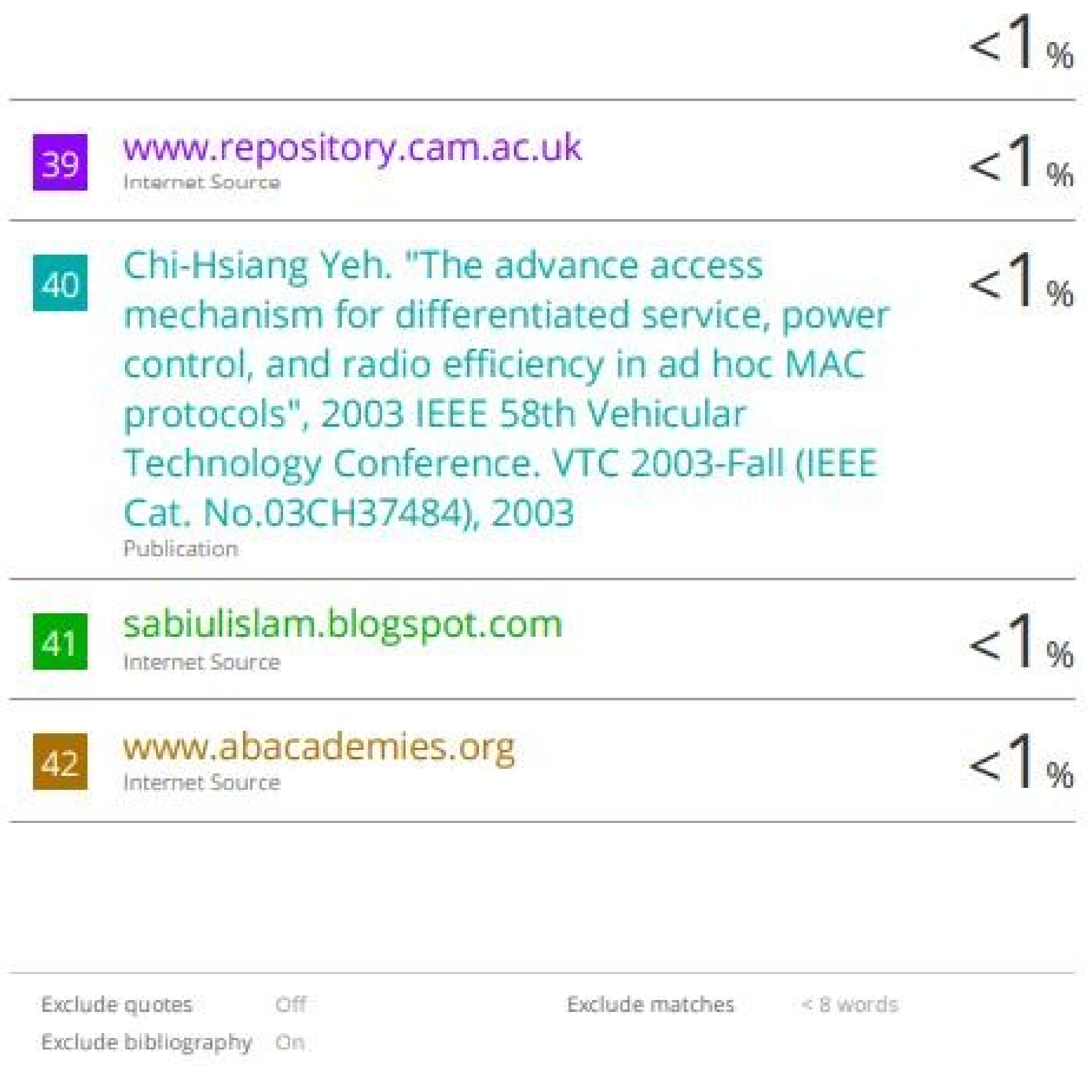
8	hdl.handle.net Internet Source	<1 %
9	Submitted to University of Greenwich Student Paper	<1 %
10	Alexander Smirnov, Alexey Kashevnik, Nikolay Shilov, Nikolay Teslya, Anton Shabaev. "Mobile application for guiding tourist activities: tourist assistant - TAIS", Proceedings of 16th Conference of Open Innovations Association FRUCT, 2014 Publication	<1 %
11	Submitted to Hong Kong Baptist University Student Paper	<1 %
12	Submitted to University of Maryland, University College Student Paper	<1 %
13	pdfcoffee.com Internet Source	<1 %
14	Submitted to The Hong Kong Polytechnic University Student Paper	<1 %
15	Submitted to City University of Hong Kong Student Paper	<1 %
16	play.google.com Internet Source	<1 %



17	Submitted to Nottingham Trent University Student Paper	<1 %
18	apkdret.blogspot.com Internet Source	<1 %
19	Peter De Tender. "Migrating a Two-Tier Application to Azure", Springer Science and Business Media LLC, 2021 Publication	<1 %
20	Submitted to Georgia Gwinnett College Student Paper	<1 %
21	Submitted to Confederation of Tourism and Hospitality Student Paper	<1 %
22	Submitted to Federal University of Technology Student Paper	<1 %
23	Submitted to National School of Business Management NSBM, Sri Lanka Student Paper	<1 %
24	Submitted to Universiti Putra Malaysia Student Paper	<1 %
25	export.arxiv.org Internet Source	<1 %
26	Submitted to Kennesaw State University Student Paper	<1 %
	Submitted to Middle East Technical University	



27	Student Paper	<1 %
28	Submitted to University of Stirling Student Paper	<1 %
29	mafiadoc.com Internet Source	<1 %
30	Submitted to University of Waikato Student Paper	<1 %
31	patents.patsnap.com Internet Source	<1 %
32	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1 %
33	Submitted to Sunway Education Group Student Paper	<1 %
34	Submitted to Xiamen University Student Paper	<1 %
35	docplayer.net Internet Source	<1 %
36	softwarecountry.com Internet Source	<1 %
37	vibdoc.com Internet Source	<1 %
38	www.portblairtravels.com Internet Source	



		<1%
39	www.repository.cam.ac.uk Internet Source	<1%
40	Chi-Hsiang Yeh. "The advance access mechanism for differentiated service, power control, and radio efficiency in ad hoc MAC protocols", 2003 IEEE 58th Vehicular Technology Conference. VTC 2003-Fall (IEEE Cat. No.03CH37484), 2003 Publication	<1%
41	sabiulislam.blogspot.com Internet Source	<1%
42	www.abacademies.org Internet Source	<1%

Exclude quotes Off
Exclude bibliography On
Exclude matches < 8 words

Universiti Tunku Abdul Rahman
Form Title : Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)

Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Loh Wai Leong
ID Number(s)	18ACB06900
Programme / Course	Bachelor of Computer Science (HONOURS)
Title of Final Year Project	Itinerary Planner Mobile Application

Similarity	**Supervisor's Comments** **(Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 6 **%** **Similarity by source** Internet Sources: 4 % Publications: 2 % Student Papers: 3 %	Within the required range.
Number of individual sources listed of more than 3% similarity: 0	Within the required range.

Parameters of originality required and limits approved by UTAR are as Follows:

(i) Overall similarity index is 20% and below, and

(ii) Matching of individual sources listed must be less than 3% each, and

(iii) Matching texts in continuous block must not exceed 8 words

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

[illegible]	
Signature of Supervisor	Signature of Co-Supervisor
Name: Liew Soung Yue	Name:
Date: 8/9/2022	Date:

Project II Checklist

UNIVERSITI TUNKU ABDUL RAHMAN
FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY
(KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	18ACB06900
Student Name	Loh Wai Leong
Supervisor Name	Dr Liew Soung Yue

TICK (√)	**DOCUMENT ITEMS** Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
N/A	Front Plastic Cover (for hardcopy)
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
N/A	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

[illegible]

____

(Signature of Student)

Date: 08 Sept 2022