

LONGMILE: THE TRIP PLANNER APP

Submitted in partial fulfillment of the requirements for
the award of
Bachelor of Engineering degree in Computer Science and Engineering

By

KUSAM ANJALI DIVITHA (37110398)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade "A" by NAAC
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600 119**

MARCH - 2021



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with "A" grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119

www.sathyabama.ac.in



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **KUSAM ANJALI DIVITHA (37110398)** who carried out the project entitled "**LONGMILE: THE TRIP PLANNER**" under my supervision from **November 2020** to **March 2021**.

Internal Guide

Dr.S.Vigneshwari, M.E., Ph.D.,

Head of the Department

Dr. L. Lakshmanan, M.E., Ph.D.,

Dr.S.Vigneshwari, M.E., Ph.D.,

Submitted for Viva voce Examination held on _____

Internal Examiner

External Examiner

DECLARATION

I **KUSAM ANJALI DIVITHA (37110398)** hereby declare that the Project Report entitled “**LONGMILE: THE TRIP PLANNER APP**” is done by me under the guidance of **Dr.S.Vigneshwari, M.E., Ph.D.**, Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in Computer Science and Engineering.

DATE:

PLACE: CHENNAI

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala, M.E., Ph.D., Dean, School of Computing, Dr. S. Vigneshwari, M.E., Ph.D., and Dr. L. Lakshmanan, M.E., Ph.D., Heads of the Department** of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide, **Dr. S. Vigneshwari, M.E., Ph.D.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

In a hectic lifestyle, every individual desire to travel and explore their dream destinations but does not have adequate time to do the required groundwork in organizing a complete vacation. For enhancing performance, a travel planning approach has been the focus of interest. Most travelers need to plan prior to their traveling for energy-saving and visiting maximum places of interest within the specified travel time.

A powerful, simple and, easy-to-use android application-based system to plan on-the-go is proposed to plan a vacation much quicker and efficiently. The users can view the near-by suggested tourist attractions, restaurants to dine-in, hotels for accommodation all together in the given travel budget using an advanced shortest path algorithm which saves time and money efficiently. It lists a comprehensive CMS of all the go-to places in that particular locality. Users will be given an option to manage destinations by adding or removing destinations depending upon user interests using Google Maps. Also, the proposed system estimates and provides various travel options from any part of the country and corresponding expenses. Thus, planning on-the-go is finally achievable.

TABLE OF CONTENTS

	ABSTRACT	v
	LIST OF FIGURES	ix
	LIST OF TABLES	xi
CHAPTER NO.	TITLE	PAGE NO.
1.	INTRODUCTION	
	1.1 INTRODUCTION	1
	1.2 OUTLINE OF THE THESIS	2
2.	LITERATURE SURVEY	
	2.1 TRAVELER'S REQUIREMENTS	3
	2.2 ANDROID PLATFORM TO MEET REQUIREMENTS	3
	2.3 STATIC MAP APPROACHES	3
	2.3.1 PERSONAL ACTIVITY TRAVEL SURVEYS	3
	2.3.2 THE SERVICES BASED ON LOCATION	4
	2.1.3 SECURITY	4
	2.4 CHALLENGES	5
	2.5 EXISTING SOLUTIONS	5
	2.6 USE OF GOOGLE MAP API	6
	2.7 LOCATION DETECTION	7
	2.6 CALCULATE DISTANCE BETWEEN 2 LOCATIONS	8
3.	AIM AND SCOPE OF THE PRESENT INVESTIGATION	
	3.1 PROBLEM STATEMENT	10
	3.2 OBJECTIVES	10
	3.3 LIMITATION AND SCOPE	11

3.3.1	SCOPE	11
3.3.2	LIMITATIONS	11
4.	SOFTWARE REQUIREMENTS AND METHODS, ALGORITHMS USED:	
4.1	SOFTWARE REQUIREMENTS.....	12
4.2	HARDWARE REQUIREMENTS	12
4.3	ANDROID STUDIO	13
4.3.1	WHY USE ANDROID STUDIO	13
4.4	ANDROID ARCHITECTURE	14
4.5	LIFE CYCLE OF ANDROID APPLICATION	15
4.5.1	ACTIVE AND RUNNING STATE	15
4.5.2	PAUSED STATE	15
4.5.3	STOPPED STATE	16
4.5.4	DESTROYED/DEAD STATE.....	16
4.6	ANDROID VERSIONS.....	18
4.7	PROJECT CONTRIBUTION.....	18
4.7.1	DIRECTION.....	19
4.7.2	LOCATIONS.....	19
4.7.3	NEARBY PLACES.....	19
4.7.4	WEATHER.....	19
4.7.5	SOCIAL NETWORKING.....	20
4.7.6	MAP MARKER INFORMATION	20
4.8	FUNDAMENTALS OF APPLICATION.....	20
4.9	FIREBASE.....	21
4.10	APIs OF GOOGLE MAPS.....	21
4.11	APPLICATION ARCHITECTURE	21
4.11.1	USER.....	21

4.11.2	FRONT END.....	21
4.11.3	ALGORITHM	22
4.11.4	SERVICES	22
4.11.5	RESPONDENTS	22
4.11.5.1	LOCATION MODERATOR	22
4.11.5.2	STORAGE MANAGEMENT.....	22
4.11.6	CONNECTIVITY MANAGER	23
4.12	SHOWING DISTANCE BETWEEN LOCATIONS.....	23
4.13	DEVELOPMENT OF SOFTWARE	23
4.13.1	REQUIREMENT SPECIFICATION.....	23
4.13.2	ANDROID DEVELOPMENT TOOLS.....	24
4.13.3	DESIGNING THE LAYOUTS.....	24
4.13.4	DEVELOPING THE LAYOUTS.....	24
4.13.5	DESINING AND IMPLENTING CLASSES	26
4.14	SUMMARY OF GUIDE	26
5.	RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS	
5.1	STARTING OF APPLICATION.....	28
5.1.1	LOCATION BASED SERVICES	28
5.1.2	DIRECTION BETWEEN TWO PLACES.....	28
5.2	LOCATION ON MAP BY MARKERS.....	28
5.3	WEATHER FORECASTING.....	29
5.4	LOCATION	29
6.	SUMMARY AND CONCLUSIONS	
6.1	SUMMARY	30
6.2	CONCLUSION.....	30
6.3	LIMITATIONS	31

6.4 FUTURE ENHANCEMENTS	31
REFERENCES	33
APPENDIX	
A. JOURNAL PAPER	34
B.PLAGIARISM REPORT	40
C. SCREENSHOTS	
C.1 CODES SCREENSHOTS	41
C.2 OUTPUTS SCREENSHOTS	43
D. SOURCE CODE	48

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO.
2.1	FIND A LOCATION USING GPS BASED ANDROID SMART WATCH	4
2.2	GPS WITH 4 SATELLITE CONNECTED OUT OF 24	6
2.3	HAVERSINE FORMULA	8
2.4	DIJKSTRA'S ALGORITHM'S ANALYSIS	9
4.1	ANDROID ARCHITECTURE	15
4.2	ANDROID APPLICATION LIFE CYCLE	16
4.3	MODULE IMPLEMENTATION	19
4.4	LONGMILE APPLICATION ARCHITECTURE	22
6.2	PLAGIARISM REPORT	40
6.3	ANDROID MANIFEST FILE	41
6.4	XML FILE	41
6.5	MAIN ACTIVITY MODULE	42
6.6	ADAPTERS MODULE	42
6.7	DATABASE INTEGRATION	43
6.8	HOME SCREEN	44
6.9	INDIVIDUAL CITY DESCRIPTION	44
6.10	CITY MAP ATTARCTIONS	45
6.11	CITY MAP F&B	45
6.12	CITY MAP UTILITIES	45
6.13	HOW TO REACH	46
6.14	BEST TIME TO TRAVEL	46
6.15	NAVIGATION PANEL	46

6.16	WISHLIST	46
6.17	SHARE	47
6.18	PERSONAL INFORMATION	47

LIST OF TABLES

TABLE NO	TABLE NAME	PAGE NO.
4.1	SOFTWARE REQUIREMENTS	12
4.2	HARDWARE REQUIREMENTS	13
4.3	ANDROID VERSIONS AND COMPARISION	18

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The goal of the project's design and creation is to create a tourist guide that will support both domestic and foreign visitors. Tourists face various difficulties due to the lack of a suitable tourist guide. When a tourist visits a city, they are required to hire skilled tourist guides, as is customary. The guides are useful for learning about the city. To receive the services of a knowledgeable guide, we must pay a substantial sum of money. The majority of tourists find it to be prohibitively costly. Most guides are not professionals because they work part-time as guides during the summer season and work in other fields during the winter. As a result, the guide will be unable to provide accurate details to the travelers because, by default, people cannot recall facts and figures that are necessary for making decisions, such as temperature, heights above sea level, weather conditions, dates and historical significance, and so on.

Using an Android-based smart phone, this tourist guide can display a map of the desired area, measure distance between two locations, and display basic information about a tourist attraction. It is accessible at any time when a visitor needs it. This tourist guide project is an Android application that allows use of Google Map API, GPS, and the Internet. The machine calculates the location's latitude and longitude and plots it on a map. It also determines the distance between the user's current position and the target location. The application would aid in the provision of new technology to the tourism industry, as well as the promotion of tourism. The following are some of the main features that the proposed tourist guide system benefits from:

- Reliable guide:

Tourists may use printed maps and hire a guide, but this does not meet their needs. We can use accurate and interactive maps of the region by using mobile features such as the global positioning system and the Internet. The application provides accurate and reliable data that aids decision-making.

- Efficiency:
Laptops and personal computers (PCs) are difficult to use due to their limited portability. For reliability and portability, the program uses smartphones instead of laptops.
- Profit:
The tourist guide app is free to download and works for almost all Android phones, so it saves tourists money on hiring a skilled guide.

1.2 OUTLINE OF THE THESIS

The current smartphone and Android city tourist guides are described in Chapter 2. It also includes Google Maps API creation, location-based sensing technologies, the global positioning system, and Android Studio. Finally, it discusses the algorithms that are used in the implementation of these technologies. The problem statement, goal, reach, and limitation of the trip planner are all included in Chapter 3.

Methodology, Android architecture, Android life cycle, trip planner architecture, and project participation are all covered in Chapter 4. The fifth chapter presents the findings of experiments conducted with a smartphone. The chapter conclusion and potential work illustrate the project's progress as well as the areas that need to be strengthened.

CHAPTER 2

LITERATURE SURVEY

The features of tourist guide types were discussed in this chapter, and various current applications for finding locations, calculating distances, and other functions were examined.

2.1 TRAVELER'S REQUIREMENTS

Despite the fact that people can get some general information about travelling on the internet, it can be difficult for newcomers to adjust to their new surroundings. Essentially, they have difficulty communicating and locating correct routing information and related costs for different routes.

2.2 ANDROID PLATFORM TO MEET THE REQUIREMENTS

People nowadays are so engrossed in modern technology that they crave an intelligent living space, as well as intelligent objects that contain powerful infrastructure with the most desirable features. As a result, Android mobile apps have become extremely common among smartphone users.

2.3 STATIC MAP APPROACHES

Some applications use static maps to provide detail. Since these maps aren't interactive, users won't be able to find real location-based services. For proper navigation, the user needs dynamic maps rather than static maps.

There are a variety of map applications available that display data in the following formats:

- Textual information
- Pictorial information
- Static maps
- Graphs
- Tables

2.3.1 *Personal Activity Travel Surveys*

Abdulazim et al. [1] introduced a data collection system and prototype

application using current smartphone sensors for personal activity travel surveys. The framework's key components run on smartphones and are supported by cloud-based (on-line) services for data storage, information sharing, and decision-making assistance.



Figure 2.1: Find a Location using GPS based Android Smart Watch

Machine-learning approaches are used to automatically assess the type of operation and travel routes with the least amount of inconvenience to the recipient. The importance of smartphone positioning data as a data source for travel behavior studies has been addressed by Wang et al. The scope for studies of travel behavior dynamics is of particular interest to the researchers. As an alternative data collection method, the authors addressed the relative advantages of mobile positioning over travel diary/GPS monitoring. Finally, they came to the conclusion that smart-phone positioning data is an important data source for studying travel behavior. [1]

2.3.2 The Services Based on Location

The Location-Based Services offer a personalized alternative to clients based on geographic location information collected by the database. This facility makes use of geographic information systems, which can be accessed from the customer side, such as GPS, or from the server side, such as positioning services for cellular network operators, and communication technologies to relay data to an application that can process and respond to the service request. [2]

2.3.3 Security

Android's architecture is stable, and the software is written in Java and XML using Android's Software Development Kit (SDK) tools. The code is then compiled

into an Android package kit (.apk), which can be installed on Android devices. Every program is treated as a new Linux client by the operating system, and each one runs in its own procedure that runs in isolation from the others. [2]

2.4 CHALLENGES

The capability of obtaining exact position from the desired location, user's location map, measuring distance between two locations, predicting weather conditions, and finding multimedia information were the challenges solved by smartphone. Many technological programs have been developed, but the specific information that a tourist requires is mostly obtained from newspapers, advertising, and magazines. These sources do not provide reliable information while tourists are travelling. [3]

2.5 EXISTING SOLUTIONS

- With these concerns in mind, various writers have recommended a variety of applications over the last decade to effectively and efficiently regulate the tourism industry by providing various solutions.
- One of the city guides is Islamabad City Guide. It does not use Google Maps or GPS (Global Positioning System), instead relying on static maps and photographs to represent locations. It does not offer services based on venue. [4]
- Smart Tourist Guide developed the guide and converted it into a smartphone app called smart travel guide, where users can look for tourism-related details that matches their needs. Travelers can get succinct information about an attraction, such as textual, pictorial, and video, using a smart travel guide. The smart travel guide, in particular, will provide travellers with knowledge about a place that can be found on a map. After measuring the distance between the traveler's position and the tourist attraction, the traveller may find nearby attractions and visit them. The application updates the user's location on the map as they pass. [3]
- • BE-ON-THE-ROAD program is mostly used for navigational purposes. BE-ON-THE-ROAD is an application that offers this functionality. It uses Google Maps, but the interface is unintuitive and lacks written and visual knowledge

about the area. Another guide, Navigation, displays all major city routes based on GPS coordinates. The user can easily use this feature in the proposed project, and the interface is user-friendly.

- BRouter also provides navigational assistance. However, the issue with this application is that it requests to download the map to the computer every time we start it. This might not be the safest way to offer navigational assistance to a visitor. As a result, my proposed project fixes this problem by allowing direct access to the map.
- GPS Route Finder is a useful app for calculating distances between two points, but it doesn't provide any other features apart from distance calculations. The proposed project contains this feature as well.

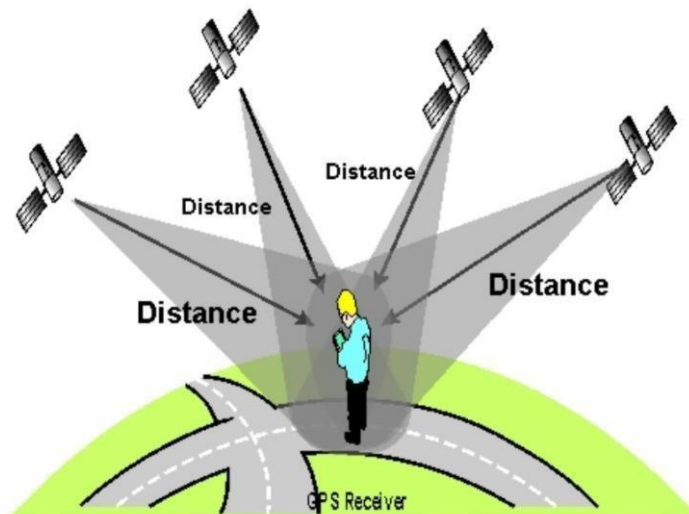


Figure 2.2: Global Positioning System (GPS) with 4 satellite connected out of 24

GPS is used in transportation and carriage systems all over the world. It provides accurate site information to as many people as possible around the world. GPS satellites send signals into the sky that are picked up by receivers, which can be a single receiver or built-in to a system. It offers 3D data in the form of altitude, latitude, and longitude. The user's location can be easily and reliably calculated if the client computer has a G.P.S (Global Positioning System), which is needed in modern smartphones. [5]

2.6 USE OF GOOGLE MAP API

- The following are essential tasks to complete while using the Google Maps

API effectively:

- To show directions from point A to point B.
- To place a marker on map based on GPS coordinates.
- Place multiple markers based on the data retrieved from the GPS.
- Add layers (such as the traffic layer).
- Fit the map to the bounds.

2.7 LOCATION DETECTION

The global positioning system (GPS) is a navigation system component that consists of a network of 24 outer earth satellites (as shown in figure 2.2) moving into an orbit launched by the US military. Initially, the army used GPS for a variety of purposes, but in the 1980s, it was revealed that the general public could also use it. At an altitude of around 10,600 miles, 21 dedicated GPS satellites and three spare satellites rotate around the world. The satellites are located so that four satellites can be seen from every point on the globe. Each satellite is equipped with an atomic clock, a radio, and a custom computer. The satellites relay their location and time readings in real time. Each satellite checks its location and time with an earth station at least once a day to search for any minor errors in the information broadcast. [6]

This software's basic function is location detection, which involves detecting the user's location on a map. Latitude and longitude coordinates are used for classification. It is important to locate the system on the maps in the correct place. Location discovery is an easy and fast method of determining the device's location on a map. Each position has a unique latitude and longitude number, which can be used to detect the location/region on a map. The main and most important reason to use this approach is that it makes use of the global positioning system (GPS), which is a worldwide and free service.

The device's accuracy is about 10 meters, which is normally enough to locate the GPS device. As shown in figure 2.1, the program automatically discovers and points the position on Google maps, which is already integrated with this application.

Haversine formula:

$$a = \sin^2(\Delta\varphi/2) + \cos \varphi_1 \cdot \cos \varphi_2 \cdot \sin^2(\Delta\lambda/2)$$

$$c = 2 \cdot \operatorname{atan2}(\sqrt{a}, \sqrt{1-a})$$

$$d = R \cdot c$$

where φ is latitude, λ is longitude, R is earth's radius (mean radius = 6,371km); note that angles need to be in radians to pass to trig functions!

Figure 2.3: Haversine Formula

2.8 CALCULATE DISTANCE BETWEEN TWO LOCATIONS

During the literature review, it was discovered that several algorithms for navigating (routing) between two points have been developed in recent years. However, in this post, we'll look at two common algorithms: Dijkstra's algorithm and the haversine formula. The 'haversine' formulae are useful for manipulating distances between direct points. It may be the shortest route across the planet's surface. The distance between the points is often referred to as the crow fly distance (it ignores the turning points, ups and downs). As a consequence, the algorithm we choose will be dependent on our requirements. Below is a brief summary of each algorithm.

Dijkstra's algorithm:

Dijkstra's algorithm was developed by computer science specialist Dijkstra in 1956 and published in 1959. It was a graph searched based algorithm that can overcome the one starting point shortest path problem of a positive graph, displaying a shortest path tree. This method is most useful when looking for a direction. This approach takes a starting node in the graph and calculates the shortest path between that node and other nodes, as shown in figure 2.3. We can also calculate the costs of a single node to a single target node for the shortest path using this approach. To accomplish this, we must split the algorithm after it has determined the shortest path to the target node. This algorithmic approach could be used to find

the shortest route between a city and all other cities, for example, if the node in the graph shows multiple places/cities and the corners shows the walking/driving distances between the places in pairs by line as path. As a result, shortest path first (I.S Intermediate System and I.S Intermediate System) and open shortest path first are commonly used in networking as routing/searching protocols.[7]

The Dijkstra Algorithm is used to process the following diagram: It is important to understand the functions of algorithm before delving into the specifics of the algorithmic-code. To solve the sub-part m, which processes the shortest path from the source to nodes among the m nearest nodes to the source, Dijkstra's Algo meets expectations.

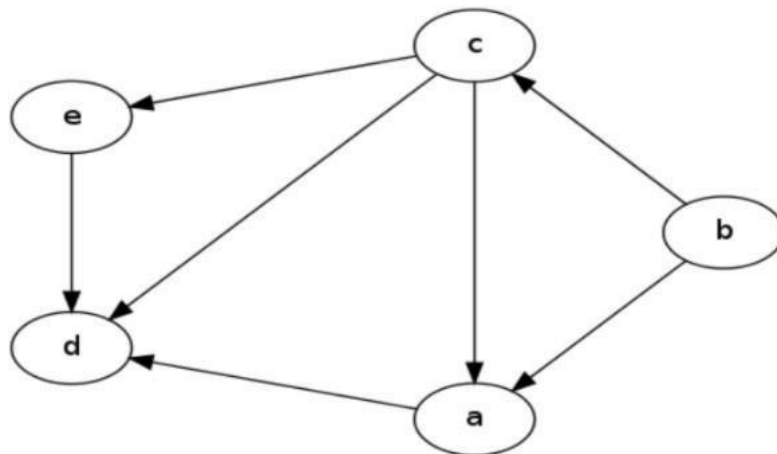


Figure 2.4: Example image for the following Dijkstra's algorithm's Analysis

It must be a coordinated weighted diagram with positive edges in order for Dijkstra's Algorithm to work. It is difficult to find the shortest path if the corners are not positive.

CHAPTER 3

AIM AND SCOPE OF THE PRESENT INVESTIGATION

3.1 PROBLEM STATEMENT

To build an Android application that will include tourist services. A tourist needs new technology that can assist them. As a result, the Android application for managing travel activities should be simple to use and effective. Tourists are having difficulties travelling due to the lack of such an application. To obtain details, they must pay a large portion of their travel budget to local guides and agents.

A proper tourist guide that can detect a current position, measure distance, and provide proper guide-lines using polylines is not available. Different issues occur when tourists visit tourist attractions, which can be addressed by developing a system that provides precise global positioning system (GPS) coordinates, proper real-time direction, and textual and pictorial information about the area. The following is a brief overview of some of the main challenges:

3.2 OBJECTIVES

- The aim of the project is to design and create a tourist guide for any metropolitan city; the application would be a user-friendly tourist guide for metropolitan cities running on the Android operating system, which is currently inaccessible for smart phones. Due to the high volume of domestic and foreign tourists in any major city, such an application is beneficial to visitors. People nowadays tend to use Android applications because they are simple to use and transport.
- Using printed maps and written knowledge to replace conventional human guide techniques. Due to a lack of knowledge from the guide, the conventional technique can cause problems when making decisions.
- To stop having to pay for the services of people that we need to employ as guides. With the aid of interactive Google Maps, the app acts as a virtual guide for tourists.

- To provide visitors with location-related facilities based on GPS coordinates.
- To determine the difference between current and ideal locations.
- To provide detailed information on tourist attractions.
- To improve tourism by facilitating the tourism industry through the use of new technologies.
- To make preferred direction facilities and a compass for correct direction accessible.

3.3 LIMITATION AND SCOPE

3.3.1 Scope

- As seen in the table, it supports nearly 90% of Android phones (Figure 4.3).
- We can use location data off-line after searching for it on the Internet.
- It can be used without registering.

3.3.2 Limitations

- It needed a computer (smartphone) with the Android operating system pre-installed.
- It could only find the positions that the creator had marked with custom markers on the map.
- Constant access to the internet, the global positioning system (GPS), and cellular data is needed.
- It is not possible to run it on a personal computer without the use of a virtual system.

CHAPTER 4

SOFTWARE REQUIREMENTS AND METHODS, ALGORITHMS USED

The proposed project makes use of Google Maps APIs to locate a spot, measure distance between two locations, and display location information.

4.1 SOFTWARE REQUIREMENTS

The program specifications are a list of the target system's features and functions. Users' perceptions of the software product are expressed in requirements. From the client's perspective, the specifications may be obvious or concealed, known or unknown, planned or unexpected.

Table 4.1: Software Requirements of the application

SOFTWARE	SPECIFICATIONS
SOFTWARE	JAVA DEVELOPMENT KIT (JDK)
	SOFTWARE DEVELOPMENT KIT (SDK)
	REAL TIME COMMUNICATION API
ANDROID SOFTWARE	ANDROID STUDIO
APPLICATIONS	ANDROID DEVELOPMENT TOOLS (ADT)

4.2 HARDWARE REQUIREMENTS

All computer software requires the existence of certain hardware components or other software resources in order to operate properly. These specifications are referred to as (computer) device requirements, and they are often used as a guideline rather than an absolute law. Most software has two sets of minimum and recommended device specifications. Device specifications begin to rise over time as demand for more computing power and resources in newer versions of software grows. According to industry experts, this trend is more important than technical advances in driving upgrades to existing computer systems. A generalization of the

first concept, the term "system specifications" refers to the requirements that must be met in the design of a system or sub-system.

Table 4.2: Software Requirements of the application

HARDWARE	SPECIFICATIONS
OPERATING SYSTEM	WINDOWS 7 (64 BIT)
RAM	4.00 GB (3.88 GB USABLE)
HARD DISK	512 MB
TOOLS REQUIRED	ANDROID STUDIO & EMULATOR
DEBUGGING	USB CABLE
PHONE	ANDROID MOBILE

4.3 ANDROID STUDIO

Google's Android software kit includes an Android Studio integrated development environment (IDE), which is based on JetBrains' IntelliJ programming code and customised to android development. In Android External, there are nine Java backport options. Android Studio does not support all published Java versions, including Java 12. IntelliJ says that it does. It's unclear how far Android Studio supports Java versions up to Java 12.

4.3.1 Why use Android Studio?

- A feature-rich mortal with a versatile Gradle-based build system are additional options in Android Studio that increase productivity when creating android apps.
- A single environment in which you can build applications for all Android devices.
- A diverse set of research methodologies and structures.
- Incorporate tools to detect performance, usability, version compatibility, and other issues.

- C++ and the NDK are supported.
- Support for Google Cloud Platform is pre-installed, making it simple to integrate Google Cloud communication and App Engine.

4.4 ANDROID ARCHITECTURE

As shown in Figure 4.1, the Android operating system is made up of five layers. These layers included an application, various libraries, the Linux Kernel, which is the OS's heart, an application system, and the Android runtime. In every Android framework, the top-level layer is known as the android application layer. Some essential features can be found here, such as quick messaging service applications, e-mail applications, calls, calendars, browsers, maps, and contacts. The most popular language used to build these components is Java.

The application framework is the next and second layer of the Android architecture. Developers of software applications use this outline or structure. For development purposes, application programming interfaces are available. It consists of simple tools that can be used to build more complex Android applications. The layer that various Android device components use. The available libraries in the context of the application are followed by the developers. These libraries are used by developers to perform basic tasks or to reuse machine functions for their own purposes. The Android Runtime is the fourth layer of this architecture. Android uses the Dalvik virtual machine in the same way as Java does to provide a runtime environment for applications.

It's a unique piece of software that creates new processes for Android apps on its own. In android architecture, the Linux kernel is the fifth and final layer. The android runs on the Linux kernel version 2.6. Memory management, power management tools, file system access, networking, inter-process communication, security settings, and many hardware drivers are all used by it. The Kernel conceals the hardware from the program user.

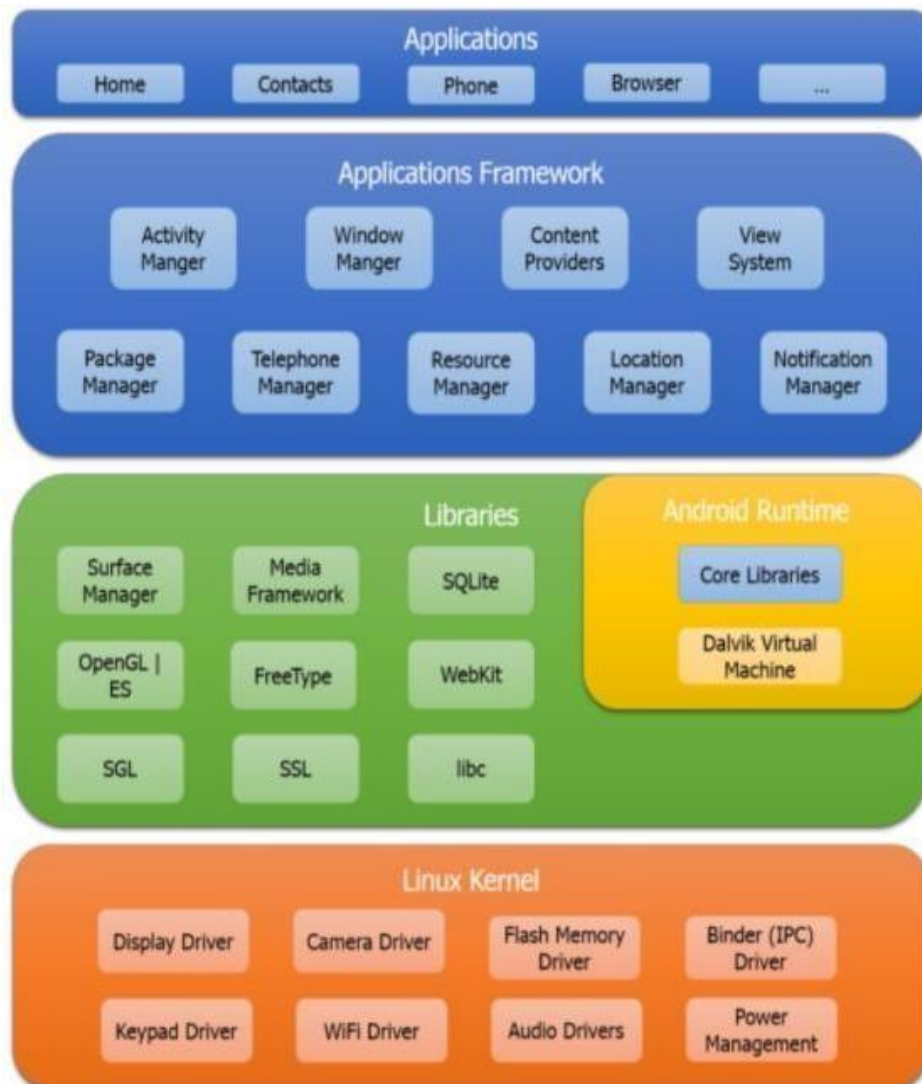


Figure 4.1: Android Architecture Reprinted from androidhaxies.com

4.5 LIFE CYCLE OF ANDROID APPLICATION

In Android, an activity can occur in one of four states, as shown below: An operation in Android can be used in four states, as shown in figure 4.2.

4.5.1 Active and Running state

In this state, the operation takes precedence and is the centre of attention. It is constantly used by the consumer and is totally visible.

4.5.2 Paused state

The software is partially visible to the user in this state, but it is not visible and unfocused. This occurs when another activity is running that does not occupy the entire screen or has some clarity, allowing the partially visible activity to be the

original Activity.

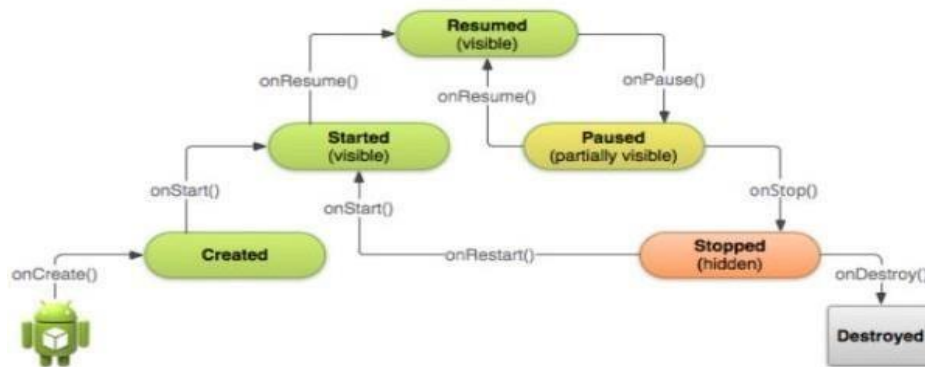


Figure 4.2: Android Application Life Cycle reprinted from strawflower.com

The paused operation is fully functional and retains its state, but it can be terminated using the task manager's option to clear currently running applications.

4.5.3 Stopped state

The operation is said to be in a stopped state when the main screen is not available. It totally obscures the view and prevents you from doing something else. In this stopped state, the activity continues to run in the background and consumes resources such as memory, but it can be stopped using the task manager's option to clear currently running applications.

4.5.4 Destroyed/Dead state

A dead/destroyed state occurs when a running program is no longer in memory. It's possible that the program was not started or that after it was started, it was cleared from memory to save memory resources. The android operation life cycle flow chart is shown in Figure 4.2. It is important for developers to investigate because it is a critical state. The life cycle of an Android application is depicted below. The flow diagram above depicting the life cycle of an Android application can be explained as follows:

- When an Activity is launched on an Android device, the onCreate() function is named. It generates the User Interface and initialises the data necessities during this time.

- Even though the consumer hasn't seen the operation yet, the onStart() function is named. We should know why the operation is still halted.
- The onResume() feature allows the user to communicate with the operation and display it on the screen. The task will be at the top of the activity stack. Now that the programme is up and running, the user may provide feedback.
- While the application is in the Active state, the onPause() function may be used to determine if the application is ready to restart when the user presses the home button or whether another operation is running on top of it.
- Only because one operation or warning is on top of it doesn't mean the visibility of actual activity is completely obscured. The mobile is switched off.
- • If one of the following three conditions occurs, the operation will be paused: The user resumes the Activity by closing the new Activity or notification, and the paused Activity becomes Active/Runs when the onResume() functions are called. If the battery is fully depleted, it will be destroyed. The Operation can't be called any more functions until it's killed, so it has to be started over by calling onCreate() and restoring the last configuration from the bundle item. Otherwise, it uses the onStop() feature to bring it to a complete stop.
- This operation is done by default when the user clicks the back button. Alternatively, it can resume at the top by being fully covered by a new operation.

In a stopped state, there are three potential scenarios:

- To free memory resources, the OS must kill: An operation in the stopped state is more likely than one in the paused state to be cleared from memory by the machine. It is necessary to restart the restart with onCreate (). If the user presses the back button to the Operation again, it is properly restarted by calling onRestart(), onStart(), and onResume() functions. In this case, the User Interface is in its proper place and no restart is needed. The activity is killed/destroyed when the onDestroy() function is called. Before the Activity dies, we may call this function one last time.

This occurs in two situations:

1. The role assigned to the operation has been accomplished.
2. To conserve memory, the programme is momentarily cleated.

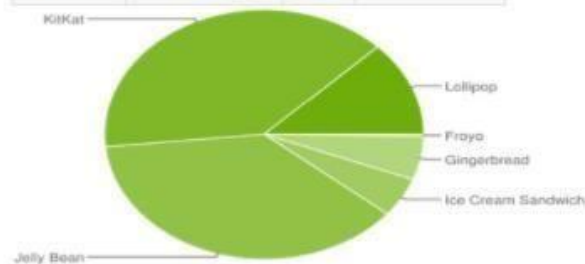
4.6 ANDROID VERSIONS AND ITS LATEST COMPARISON

The various versions of the Android operating system are a significant feature. These versions are getting better and introducing new functionality for users. As mobile hardware advances and becomes more capable of providing more services, the Android operating system improves as well. The most recent utilisation comparison is shown below.

Table 4.3: Android Versions and Its latest comparison

(Data collected during a 7-day period ending on June 1, 2015)

Version	Codename	API	Distribution
2.2	Froyo	8	0.3%
2.3.3 - 2.3.7	Gingerbread	10	5.6%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	5.1%
4.1.x	Jelly Bean	16	14.7%
4.2.x		17	17.5%
4.3		18	5.2%
4.4	KitKat	19	39.2%
5.0	Lollipop	21	11.6%
5.1		22	0.8%



4.7 PROJECT CONTRIBUTION

This project encapsulated basic city guide features such as showing Google maps, locating user's location, getting information about tourist spots, basic information, showing different routes by directions, and choosing off-line information about different facilities such as hospitals, shopping centres, car rentals, checking the latest weather conditions, keeping in touch on social networking, and looking at different routes by directions.

The following are the six main sections of this project:

4.7.1 Direction

This operation retrieves Google maps and allows the user to pick a position and calculate distances between them. To complete these tasks, it makes use of Google's path API. By drawing lines, it also portrays the tourist's journey from point A to point B. It has the ability to measure distances in kilometres. It also uses GPS coordinates to indicate the user's current position. It necessitates the use of the Internet and GPS.

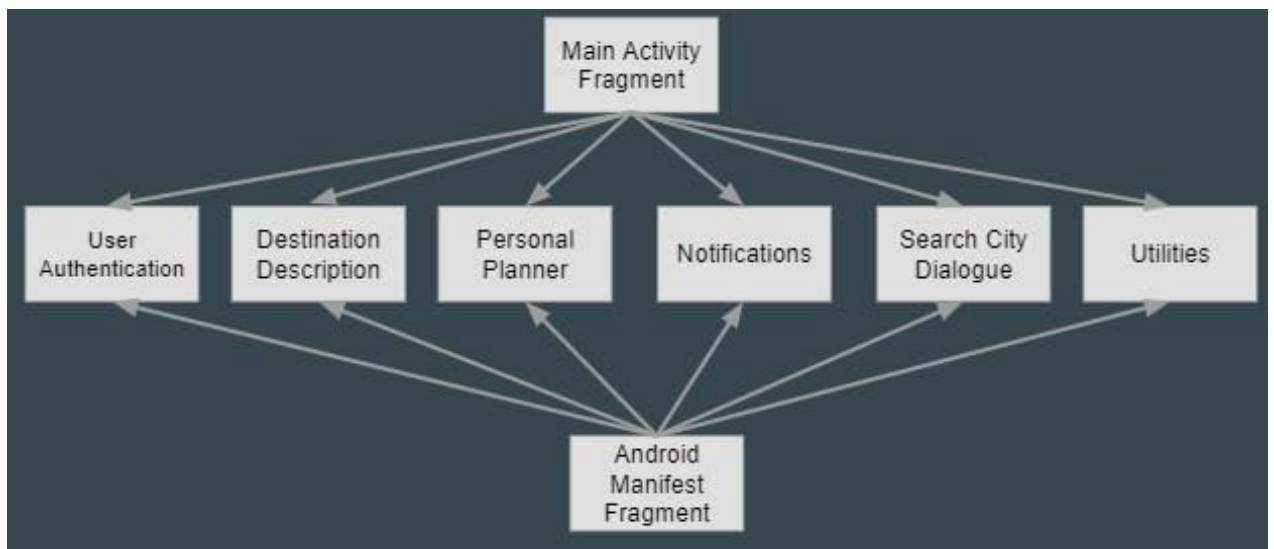


Figure 4.3: Module Implementation

4.7.2 Locations

It's an essential feature of this app that displays offline locations such as hospitals, car rentals, pharmacies, shopping centres, and other places in a region. It includes contact details such as phone numbers, addresses, email addresses (if available), and website addresses (if available).

4.7.3 Nearby Places

Google has another essential API called Google Places API, which detects current tourist coordinates and displays nearby places based on their position, assisting users in locating nearby places.

4.7.4 Weather

This section will provide tourists with the most up-to-date weather

information. To assist the user of this guide, it displays hourly, regular, and weekly weather forecasts.

4.7.5 Social Networking

Since social networking plays such an important role nowadays, this section links visitors to a special Facebook community page. Where they can talk about tourism and share their travel experiences in any region.

4.7.6 Map Marker Information

This choice displays a map with customizable markers that display the necessary details about a region. A map position is represented with a picture and some text.

4.8 FUNDAMENTALS OF APPLICATION

This portion of the application highlights important aspects of application growth. It is divided into four parts. Services, receivers, conference providers, operation, and broadcast are all terms that can be used to describe services, receivers, conference providers, activity, and broadcast. One or more of these component forms may be present in an Android application.

- The operation displays user interfaces that will be used for communication. One operation in the short messaging component, for example, shows users the user interface for sending a short message to others. The Activity main (base) class is used to write all tasks.
- Services aren't running in the background, and the user interface isn't closing. For example, if we play music in the background. Service is an extension of the Service base class.
- Radio Receivers: This section receives broadcast announcements and responds to them. In other words, if the battery is low, it is important to inform mobile users. Broadcast Receiver base class is expanded by the antennas.
- Service Provider: This portion collects and stores data. It extends the Content Provider base class, but instead of calling Content Provider's methods directly, applications call methods in the object Content Resolver, which call Content Provider.

For retrieving and storing data, Android employs the following process. Database, files, and interests are all part of the network. Creating and using SQLite databases is supported by the Android API. Each database is only accessible by the app that created it. The SQLite Database object represents a database and includes methods for constructing the database, running queries, and handling the data.

4.9 FIREBASE

Google acquired Firebase, a web and mobile development application platform, in 2004. It offers cloud-based applications as well as real-time databases. Developers may use the API services which allows information from the framework to synchronize among users and store in cloud.

It is easy to work with Firebase since it has no tables or help requests, providing a leg up on the competition across conventional databases. When an update to the information is made (for example, a new statement is added), all linked purchasers can now be updated as soon as possible.

4.10 APIS OF GOOGLE MAPS

APIs of Google Maps permits maps to be supplemented and adds map details to the Android application that is provided by Google Maps. The API has connections to Google Maps servers and is responsible for uploading data as well as displaying and handling gestures on maps. To add markers, API calls are often used and apply it to a plain map, which improves interaction with the user.

4.11 LONGMILE TRIP PLANNER APPLICATION ARCHITECTURE

LongMile Trip Planner is working as the following architecture. It starts with a menu which will appear where a user can select different locations like tourist spots, market, schools, hospital etc. The system checks the Internet connection and provide the map for that location. It is shown in figure 4.4.

4.11.1 User

The one who communicates with the Android application in order to obtain a daily travel schedule or some other additional details.

4.11.2 Front End

This is the software on the screen. To display preferences and other components, interaction between backend of the application and the user has been communicated, as well as adjustments were made to the displayed output as a result.

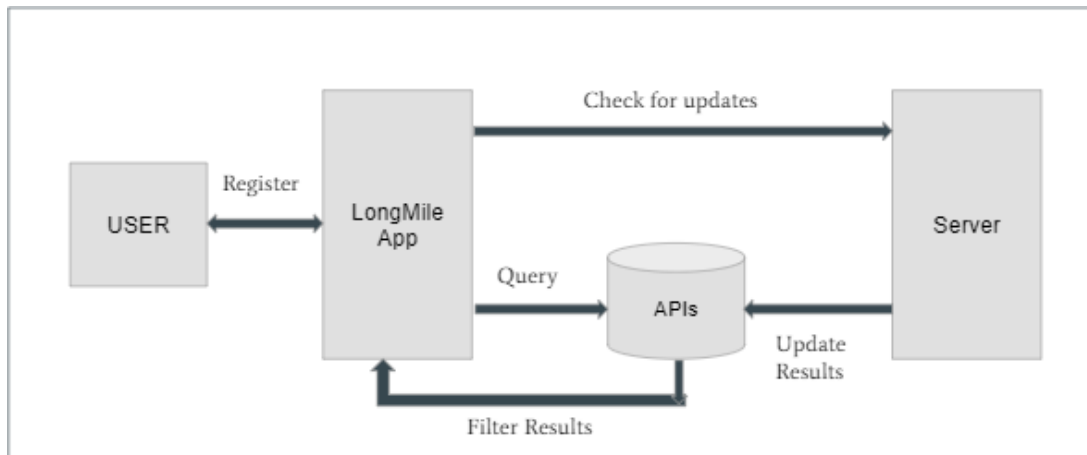


Figure 4.4: LongMile Application Architecture

4.11.3 Algorithm

In this section software's choices are made based on user reviews, as well as backend resources, are considered, as they may have an impact on the actual output to be shown.

4.11.4 Services

Services are the name for all of the application's context operations. This entails obtaining information from the server, calculating approximate distances, and determining the shortest path, all of which the server is capable of providing.

4.11.5 Respondents

4.11.5.1 Location Moderator

A utility that is tasked with gathering the user's approximate status and, as a result, displaying nearby amenities.

4.11.5.2 Storage Management

The system does not have access to the internet on a continuous basis. This permission is required in order to store information regionally in order to keep track of future plans.

4.11.6 Connectivity Manager

The group responds to concerns about network access. As the network link shifts, it also notifies the applications.

4.12 SHOWING DISTANCE BETWEEN TWO LOCATIONS BY DRAWING POLYLINES

We can measure distance between two points using latitude and longitude. When the user selects the source and destination, a polyline will be drawn. Typically, the system position is the source location, which the app senses and displays on a map. We may, however, alter both the source and destination positions.

4.13 DEVELOPMENT OF SOFTWARE

Software development starts with a software process. The software model illustrates how a developer can build software and which approach is best for a specific application. The Rapid Application Development (RAD) model is the one I've chosen. The key reason for choosing this model is that it focuses on application creation rather than planning activities. It is a paradigm that is adaptable. Other models, such as the water fall, are rigid, and a comprehensive plan for software development is needed. I was new to Android Software Development and had no idea how to build this app. We can create a software prototype, demonstrate it, and refine it until the necessary implementation goals are met using this model. Then we put it into effect.

4.13.1 Requirement Specification

- i. Functional requirements:
 - The application can display a map of the city.
 - The map can be zoomed in and out.
 - The application can show the path between various locations on the map.
 - The application can provide language training.

ii. Non-functional requirements:

- The user interface should be simple to use and comprehend.

4.13.2 Android Development Tools

For open-source mobile apps, Android is the most relevant platform. The default operating system is Linux kernel-based, and user applications are developed in Java using the Android SDK. As Android can be built on both Mac and Windows systems, it becomes more versatile [2]. We can not only build but also run and test the application using the Android SDK add-on on emulators and real Android devices.

4.13.3 Designing the layouts

First view:

The first view contains two buttons and a user greeting. The buttons allow users to switch to a different view.

Phrases button:

To see three different scenarios that users can encounter while interacting with residents, click the Phrases button. To see a map and route transportation information, click the second button, Search transportation medium.

View of Phrases:

There will be three keys, arranged in the following order: Hotel conversation, Restaurant conversation, and Bus conversation. Each of these conversations' subpages is a scrollview.

View from the Route:

GPS is used for automatic localization.

4.13.4 Developing the layouts

Android applications uses XML layout to display its contents. Each tag in XML layout performs given property. By defining the id, these tags can be used in java code.

Main view:

Main view will have simple look using RelativeLayout statement for easy customizing.

Example of RelativeLayout statement for XML document should look like:

```
<RelativeLayout xmlns:android = "http://schemas.android.com/apk/res/android"
    xmlns:tools = "http://schemas.android.com/tools"
    android:layout_width = "match_parent"
    android:layout_height = "match_parent"
</RelativeLayout>
```

Layout_width / layout_height is used to custom the view in mobile device. In this example, "match_parent" is used so the view will be as big as its parent. Layout_height is kind of similar to width but it uses vertical scale to display the view.

The child of RelativeLayout can be any view, TextView, Button, etc. In the project, TextView is used to show the name of the application and the ImageView for displaying the background. There are also two Button for menu options.

ScrollView:

ScrollView is a special kind of layout, it's used to hold view that is larger than its actual size. Once the view size is larger than the ScrollView size, it will add scroll bar that can be scrolled vertically [5].

```
<ScrollView xmlns:android = "http://schemas.android.com/apk/res/android"
    android_width = "fill_parent"
    android_height = "fill_parent" >
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
    >
    </LinearLayout>
</ScrollView>
```

By default, the ScrollView has a LinearLayout as its only content. However, this LinearLayout can give a simple display of contents in either vertically or horizontally [4].

4.13.5 Designing and implementing the classes

On an Android project, an Activity is the collection of elements that communicate with the user [5]. All of the operations are categorized into groups, which are stored in the src folder. The main activity class in any Android program is MainActivity.java.

Intent is used to navigate to the user's option in the MainActivity.java class. An purpose is a type of object that describes a task to be completed. It is the most common method for developers to begin new activities within an application. It can also be used to interact between various activities [4].

```
public void onClick (View v){  
    Intent i = new Intent (getApplicationContext(), Phrase.class);  
    startActivity(i);  
}
```

This method will take as a parameter view, which will cause a new intention to be triggered. Essentially, it will execute code from a particular class. This Purpose will run the Phrase class once it is clicked in the example above. All intents are described in the project's AndroidManifest file [3].

```
<activity  
    android:name = "packet_name.MainActivity"  
    <intent-filter>  
        <action android:name="android.intent.action.MAIN"/>  
        <category android:name="android.intent.category.LAUNCHER" />  
    </intent-filter>  
</activity>
```

For all activities, we must declare the intent filters in the manifest file.

4.14 SUMMARY OF GUIDE

The operations, as well as the navigational and location-based services offered to the user, are depicted in all of the snap shots and other information. The

icons reflect various ties to the city tourist guide application's functionality. The first operation provides a user with a main menu from which they can choose various choices. The other activities will result in a listing of the said functionalities based on the position and coordinates that the user has requested, as well as the user being able to fulfil their guidance needs by providing criteria. Tourists may use the app to get different types of information such as markers, pictures, charts, and text information on their smartphones.

CHAPTER 5

RESULTS AND DISCUSSION, PERFORMANCE ANALYSIS

5.1 STARTING OF APPLICATION ON ANDROID SMARTPHONE

The programme ran smoothly on a Samsung Galaxy Tab 3 with Android Kit Kat 4.4.2 as the operating system. The first screen that the application displayed was the menu screen, which was configured during the application's creation. The navigation panel then appeared, with several screen switch options.

5.1.1 Location Based Services

The app has three key features: it displays directions, calculates distance between two locations, and locates famous landmarks. These functionalities are depicted in the screen shots below. Custom markers provide both text and pictorial information about the spot. By clicking on the marker, typing the name of the position in the location search text box, and entering GPS coordinates, the location can be found. Instead of using an emulator, these tests were carried out on a real computer. The findings accurately depicted the location and provided the desired outcome.

5.1.2 Direction Between two Places

The experiment's next step is to test the direction interface, which successfully displayed the direction between two cities by drawing a polyline on the map. My position was determined using GPS coordinates, which can be selected by pressing the default button on the chart. It displayed the amount of time it took to fly between the cities as well as the distance in kilometers.

5.2 LOCATION ON MAP BY MARKERS

The markers prominently display the spot, allowing the user to quickly find the desired location. It also includes an image and some basic information about the venue. It displays the developer's customized text and photographs. The default markers could not show comprehensive city details, but we can feed any information based on the user's needs using custom markers.

5.3 WEATHER FORECASTING AND SOCIAL NETWORKING

Both modules are operational and display the necessary details.

5.4 LOCATION

This module properly displays offline information about important locations, such as name, address, phone number, and so on.

CHAPTER 6

SUMMARY AND CONCLUSIONS

6.1 SUMMARY

Everyone wants to fly and visit their dream destinations in today's hectic world, but they don't have enough time to do the necessary groundwork to plan a complete vacation. Most travelers must prepare ahead of time to save energy and visit as many points of interest as possible within the allotted travel time. To make vacation planning much faster and more effective, a powerful, quick, and easy-to-use android application-based framework is proposed.

Using an advanced shortest path algorithm, users can view nearby recommended tourist attractions, restaurants to dine-in, and hotels to stay in all within the given travel budget. This saves time and money. The proposed system measures and offers different travel options and costs from any place in the world. Prioritizing the budget allocated to different categories is also feasible.

6.2 CONCLUSION

For a visitor, the tourist guide application is an important tool. Most conventional approaches are time consuming and necessitate the use of professional human resources to direct tourists in the region. The main aim of the proposed project is to ensure that visitors save time by offering sufficient instructions and directions.

The tourist guide system has an easy-to-use menu where users can pick from a selection of buttons depending on their requirements. The chooses directions, positions, distances, and other choices depending on their requirements. These services can be accessed via the Internet, Global Positioning Systems (GPS), and Google Maps. The Google maps are interactive, allowing the user to quickly identify locations and draw a line to indicate proper direction. These lines assist in getting to the desired spot. The consumer can also get the most up-to-date weather forecast for the region. This tourist guide makes use of the most up-to-date charts, which can

show locations based on user-provided latitude and longitude.

6.3 LIMITATIONS

Following our project's assessment, we discovered several areas that need extra focus and work. The phrases we've created so far aren't interactive in real time. As a consequence, it may not be as beneficial to the target users (foreigners) because it does not assist them in real-time language translation. Another problem is that we do not use a GPS location tracker to keep track of the user's current location. A consumer must include his current position and destination to the map in our system, which is inefficient. Furthermore, there could be a different number of local buses across the area, which the user is unaware of but needs to be aware of so that he or she can get a sense of which routes are served by local buses.

6.4 FUTURE ENHANCEMENTS

- Adding Google Street view

The proposed system displays pictures and photos of a spot, but we can use Google Street View map API to display three-dimensional images of the location. It allows the user to see the location from all sides. Which is extremely useful in making some decision about that position and visualizing it clearly.

- Off Line version

The proposed framework could be adapted to an offline version in the future. This guide would then be more useful for tourists who do not have access to the Internet. Making it off-line would boost its storage space, but it could be jeopardized because visitors will reap more benefits.

- Search Nearby

Allow users to find specific locations near their current geographic position, such as hotels, restaurants, and book stores [7].

- Real Time Interactive Conversation

Instant language translations can be used to make conversations in hotels, taxis, and restaurants real-time interactive.

- GPS Location Tracker

Instead of giving user's current location to the map, the user's location will be

automatically followed by GPS location tracker which will make the app more powerful and systematic.

- More Detailed Routing Information

The user can be given more detailed routing information for nearby areas in the region.

- Whole Country Coverage

Since we're using Google Maps to locate the user's destination and route information, this app can be used across the entire world.

I hope that by completing this project, other students and developers can gain a deeper understanding of Android programming, Java, and XML. They will use Android application creation in their study, and they will use it and improve it in the future by finding more research topics in these directions.

REFERENCES

- [1] Y Liu, Y Hu, and X Li. A smartphone-based prototype system for incident/work zone management driven by crowd- sourced data. Digital Repository Iowa State University, 2(2), 2015.
- [2] S Muzumil. Fastfinder android application. A Thesis Presented to the Department of Software Engineering Mehran University of Engineering & Technology, Jamshore, 2(2), 2014.
- [3] Jinendra, D Bhagyashri, and A Pranav. Smart travel guide. 1st International Conference on Recent Trends in Engineering & Technology, 2012.
- [4] Umair Ahmed. Pakistan travel guide. <http://intellisenseglobal.com/>, December 2014.
- [5] N Xu J Meng. A mobile tourist guide system based on mashup technology. IEEE, 2010.
- [6] S Garmin. What is gps? Wikipedia, 2015.
- [7] A Rosetta. Dijkstra’s algorithm. rosettacode.org, 2015.
- [8] M Puthuparampil. Report Dijkstra’s algorithm. Technical report, New York University, 2007.
- [9] L Storm. Disadvantages & advantages of using google maps website. <http://science.opposingviews.com/>, 2001.

APPENDIX

A. JOURNAL PAPER

A TRIP PLANNER ANDROID APPLICATION

¹KUSAM ANJALI DIVITHA, ²S.VIGNESHWARI

¹Department of Computer Science and Engineering, Sathyabama Institute of Science and Technology, Chennai, Tamil Nadu

²Faculty of Computing, Sathyabama Institute of Science and Technology, Chennai, Tamil Nadu

E-mail: kusam.anjali@gmail.com, vikiraju@gmail.com

ABSTRACT

In a hectic lifestyle, every individual desires to travel and explore their dream destinations but does not have adequate time to do the required groundwork in organising a complete vacation. For enhancing performance, a travel planning approach has been the focus of interest. Most travellers need to plan prior to their traveling for energy-saving and visiting maximum places of interest within the specified travel time. A powerful, simple and, easy-to-use android application-based system to plan on-the-go is proposed to plan a vacation much quicker and efficiently. The users can view the near-by suggested tourist attractions, restaurants to dine-in, hotels for accommodation all together in the given travel budget using an advanced shortest path algorithm which saves time and money efficiently. It lists a comprehensive CMS of all the go-to places in that particular locality. Users will be given an option to manage destinations by adding or removing destinations depending upon user interests using Google Maps. Also, the proposed system estimates and provides various travel options from any part of the country and corresponding expenses. Thus, planning on-the-go is finally achievable.

Keywords: Travel planning, Android application, Comprehensive CMS, Google Maps.

1. INTRODUCTION

Travelers typically waste time deciding or planning their holidays. Given the situation, the proposed android application focuses on identifying the most critical support for travellers in order to improve and provide them with a useful tool to schedule their trip in a timely manner. In general, tourists want to see the most famous monuments and historical sites in a given region.

To accomplish this, the proposed android application will automatically display and set up user's direction of travel. By pre-selecting a central location, the system aids a newcomer to town or someone who wishes to explore the city. During the login process, the user is expected to enter their interests and preferences. After that, the user can choose to enter his or her current location manually or let the device do so. Then, they will personalise their plan of action by wish listing the locations they might wish to go to. Currently, the app can easily analyse the user's desires and expectations, as well as the fundamental location the user wants to visit, and design a route with the easiest traveller spot around the chosen location. The device employs the Google Maps API to generate a list of all the places in the vicinity of the chosen site, along with all relevant details. Before being shown to the user, all of the position area units

are sorted based on scores, distance, and a variety of alternative constraints.

2. LITERATURE REVIEW

There are variety of alternative connected surveys that have already been printed. The aim of this section is to provide a detailed analysis of different approaches to exploring tourist trip applications that are designed and produced for a broad audience.

In [4], there is a closely related survey that compares a few of the current tour planning framework approaches. Furthermore, tourer recommender applications are seeing an increase in the availability of auto-mated routing. The importance of offering bespoke touristic tour design with appropriate latency is emphasised.

The City Trip Planner described in [5] is the most innovative integration framework for web-based tour-guide support. This paper analysis is extended by the proposed approach, which adds additional functionalities.

In [1], minimal survey is given for analysis of 4 representative mobile tour guides. These applications area units compared supported their ability to supply information regarding transport, security, POIs and weather. The system proposed a framework that

includes a classification of tour services, and delivery aspects. But it doesn't offer specific trip design.

Another survey [3], it evaluates 8 travel systems that support a catalogue. It includes modules such as community options, security, personalization. But, none of the offer functionalities for touristic trip designing, that stands out from other papers.

The two smart cities IoT technologies [2]: the first is energy and heat management, which uses a variety of tools to optimise energy consumption both in regions where citizens reside and industrial areas. The following is the second submission for public transportation regulation of the cruise, which uses a variety of tools to include driving recommendations that are environmentally friendly.

Smart Intelligent tourism systems [6] describes how business enterprise destinations, their industries, and their visitors are increasingly reliant on emerging types of ICT that allows the transformation of large amounts of data into value propositions. The text of the document begins by defining good business enterprise, then goes over current good tourism patterns before laying out its technical and business foundations. This is often accompanied by a review of the benefits and drawbacks of running a successful company. The term was introduced to cities for identifying campaigns aimed at promoting the use of victimisation technology in novel ways to enhance resource management, efficient and transparent governance, property, and quality of life.

3. OBJECTIVE AND AIM

Aim is to develop a system that analyses the customer's preferences, as well as the approximate distance that the user is willing to travel, and to provide them with a variety of options in Google Maps from which the user can choose a specific location to maximise their time.

Objective: The proposed Android application is primarily intended to assist a newcomer to town or someone who wishes to uncover the locality on a regular basis at their leisure by pre-selecting a central location.

4. MOTIVATION

Any business enterprise or established agencies may use this method to assist their customers or agents in achieving a competitive advantage over paper-based maps. When a user first logs in, they are asked a series of questions that help them to narrow down their

search for locations. The locations are shown on charts, which provide a clear picture of the situation and directions from one location to the next.

If the user is new to town and has no idea where to go, the system will display all nearby hotels, tourist attractions, and hotels on the map. The customer receives a complete read of the plan right away, with the highest level of accuracy. The need for an android application would significantly shorten the period spent hunting for locations and procuring feedback for each of the spots, making it less difficult and allowing for faster decision-making.

5. WORK DONE/PROPOSED SYSTEM

5.1 PROBLEM DESCRIPTION

As the name implies, "A Trip Planner Android Application" takes the approach of analysing the likes and dislikes of the user and thus estimating distance that the consumer is curious about the area and wants to learn more about it and offer entirely distinct options under Google Maps starting from which the user has an ability to choose specific location in order to get the best out of their time. The proposed method can be primarily used for assist a newcomer to town or someone who wishes to uncover the locality on a regular basis at their leisure.

An Android application is suggested as part of the proposed scheme. The user interface was designed and developed using Android Studio. The recently posted application had number of issues with trail optimizations. This subject was chosen after being inspired by a project called Android based Smart City, which can be implemented or used by any company or establishment agency associated with business enterprise to assist their customers or agents by acquiring advantage more than a map printed on paper. Several aspects of the cited paper have been improved, and all data is stored in Firebase for database connectivity.

5.2 PROPOSED FRAMEWORK

5.2.1 Android Studio

Google's Android app package comes with an integrated development environment (IDE) called Android Studio, based on JetBrains' IntelliJ computer code and tailored to android development. There are nine Java backport options in Android External.

Although IntelliJ claims that Android Studio supports all released Java versions, including Java 12, Android Studio does not. The extent to which Android Studio supports Java versions up to Java 12 is unknown.

5.2.2 Why use Android Studio?

Additional options in Android Studio that improve productivity when developing android apps include:

- A feature-rich mortal with a versatile Gradle-based build framework.
- A seamless environment where you can build for all Android devices.
- A wide range of research methods and frameworks.
- Connect tools to detect issues with performance, usability, version compatibility, and other factors.
- Support for C++ and the NDK.
- Google Cloud Platform support is built-in, making it easy to incorporate Google Cloud communication and App Engine.

5.2.3 Android Overview



Fig.1 Android Component View

The higher layers of the framework are used often when compared to the lower layers. Figure-1 purpose is to understand the working components with less code.

5.2.4 Firebase

Google acquired Firebase, a web and mobile development application platform, in 2004. It offers cloud-based applications as well as real-time databases. Developers may use the API services which allows information from the framework to synchronise among users and store in cloud.

It is easy to work with Firebase since it has no tables or help requests, providing a leg up on the competition across conventional databases. When an update to the information is made (for example, a new statement is added), all linked purchasers can now be updated as soon as possible.

5.2.5 APIs of Google Maps

APIs of Google Maps permits maps to be supplemented and adds map details to the Android application that is provided by Google Maps. The API has connections to Google Maps servers and is responsible for uploading data as well as displaying and handling gestures on maps. To add markers, API calls are often used and apply it to a plain map, which improves interaction with the user.

5.3 SYSTEM ARCHITECTURE

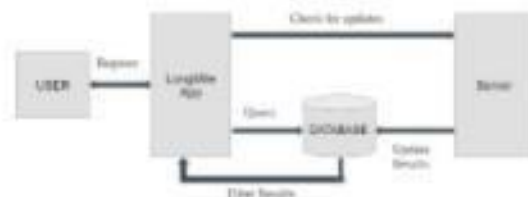


Fig.2 System Architecture

USER:

The one who communicates with the Android application in order to obtain a daily travel schedule or some other additional details.

FRONT END:

This is the software on the screen. To display preferences and other components, interaction between backend of the application and the user has been communicated, as well as adjustments were made to the displayed output as a result.

ALGORITHM

In this section software's choices are made based on user reviews, as well as backend resources, are considered, as they may have an impact on the actual output to be shown.

SERVICES:

Services are the name for all of the application's context operations. This entails obtaining information

from the server, calculating approximate distances, and determining the shortest path, all of which the server is capable of providing.

RESPONDENTS

LOCATION MODERATOR:

A utility that is tasked with gathering the user's approximate status and, as a result, displaying nearby amenities.

STORAGE MANAGEMENT:

The system does not have access to the internet on a continuous basis. This permission is required in order to store information regionally in order to keep track of future plans.

CONNECTIVITY MANAGER:

The group responds to concerns about network access. As the network link shifts, it also notifies the applications.

6. RESULTS & DISCUSSION

The proposed trip planner android application helps the users to view the near-by suggested tourist attractions, restaurants to dine-in, hotels for accommodation all together in the given travel distance which saves time and money efficiently.

This Android application is enforced in some modules as mentioned below:

6.1 Google Map Analysis

Three categories are considered i.e., Tourist attractions, restaurants and utilities. Depending on the current or specified location, nearby locations are shown as per preferences differentiating between the three categories.

Figure-3 is the application differentiating all the three categories



Fig.3 Google Map Analysis Module

6.2 City CMS

Every city or locality is provided with a description which includes Food & Restaurants section, how to reach section, Hotels & accommodation and best to visit section. Figure-4 module is maintained to put information all together about a particular location.



Fig.4 City CMS Module

6.3 How to Reach

This feature helps in providing information for people who would want to check on transportation details from one city to another by three means i.e., air-plane, train and bus.



Fig.5 How to Reach Module

6.4 Wishlist & Recently Browsed

As per Figure-6, users are given an option to wish list certain localities and view them later according to their convenience.



Fig.6 Wishlist Module

And, as per Figure-7, the list of recently viewed cities or localities are displayed which makes it easy for the user to review the places later.



Fig.7 Recently Browsed Module

6.5 Navigation Panel

Navigation Panel is used to switch between different modules available in the application. Figure-8 gives

the details about all the modules that are involved in the android application.

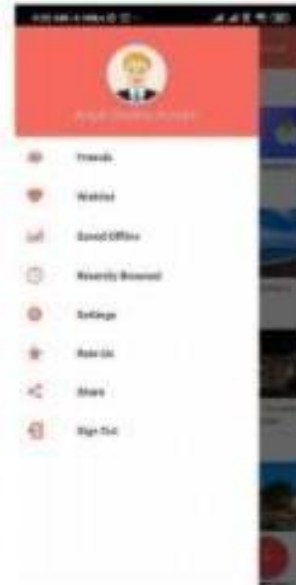


Fig.8 Navigation Panel Module

7. CONCLUSION AND FUTURE WORK

The majority of people do not use cutting-edge software and spend a significant amount of time manually preparing their vacations. As a result, the proposed Android application enables visitors to make the most of their limited time while also planning their stopover.

Under Future work, the proposed system can be further integrated with popular social medias in order to make it easy for the users to share the travel plan and further details.

REFERENCES

- [1] Andrea Rossi, Joshua Cooper, Andrea Rossi, Dimosthenis Kyriazis, Theodora Varvarigou - "Sustainable smart city applications", 2009
- [2] U. Gretzel, "Intelligent systems in tourism: A social science perspective. *Annals of Tourism Research*", vol. 38, no. 3, pp. 757-779, 2011.
- [3] Dippelreiter, B., Grün, C., Pöttler, M., Seidel, I., Berger, H., Dittenbach, M., Pesenhofer, A. (2007). *Online Tourism Communities on the Path to Web 2.0 - An Evaluation*. Submitted for *Virtual communities in Travel and Tourism*, Special Issue, *Journal on Information Technology & Tourism*.

- [4] Vineet Singh, Akeshnil Bali, Avinesh Adhikthikar, Rohitash Chandra, "Web and mobile based tourist travel guide system for fiji's tourism industry", Computer Science and Engineering (APWC on CSE) 2014 Asia-Pacific World Congress on, pp. 1-7, 2014.
- [5] F.B. Zhan, C. Noon, "Shortest Path Algorithms: An Evaluation Using Real Road Networks", Transportation Science, vol. 32, no. 1, pp. 65-73, November 1996.
- [6] Soufria, W. and Vansteenwegen, P. (2010) Tourist Trip Planning Functionalities: State-of-the- Art and Future, Proceedings from International Workshop on Web Engineering & Tourism, Vienna, Austria.
- [7] P. Vansteenwegen, W. Souriau, G. Vanden Berghe, and D. Van Oudheusden. The city trip planner: an expert system for tourists. Expert Systems with Applications, 2010.
- [8] C. Chung-Hua, H. Chenyang, "A Platform for Travel Planning by using Google Maps", 16th IEEE Int. Conf. Mobile Data Management, vol. 2, pp. 120-125, Jun. 2015.
- [9] D. Gavalas, C. Konstantopoulos, K. Mastakas, G. Pantziou, "A survey on algorithmic approaches for solving tourist trip design problems", Journal of Heuristics, vol. 20, no. 3, pp. 291-328, 2014.
- [10] Larisa Kuznetsova, Arthur Zhigalov, Natalia Yanishevskaya, Denis Parfenov, Irina Bolodurina, Advances in Intelligent Systems, Computer Science and Digital Economics, vol. 1127, pp. 346, 2020.

Figure 6.1: Journal paper

B. PLAGIARISM REPORT

A TRIP PLANNER ANDROID APPLICATION

ORIGINALITY REPORT

1%

SIMILARITY INDEX

1%

INTERNET SOURCES

1%

PUBLICATIONS

1%

STUDENT PAPERS

PRIMARY SOURCES

1

Submitted to Indian Institute of Technology,
Madras

Student Paper

1%

2

"International Conference on Intelligent Data
Communication Technologies and Internet of
Things (ICICI) 2018", Springer Science and
Business Media LLC, 2019

Publication

<1%

Exclude quotes Off

Exclude matches Off

Exclude bibliography Off

Figure 6.2: Plagiarism Report

C.1 CODES SCREENSHOTS

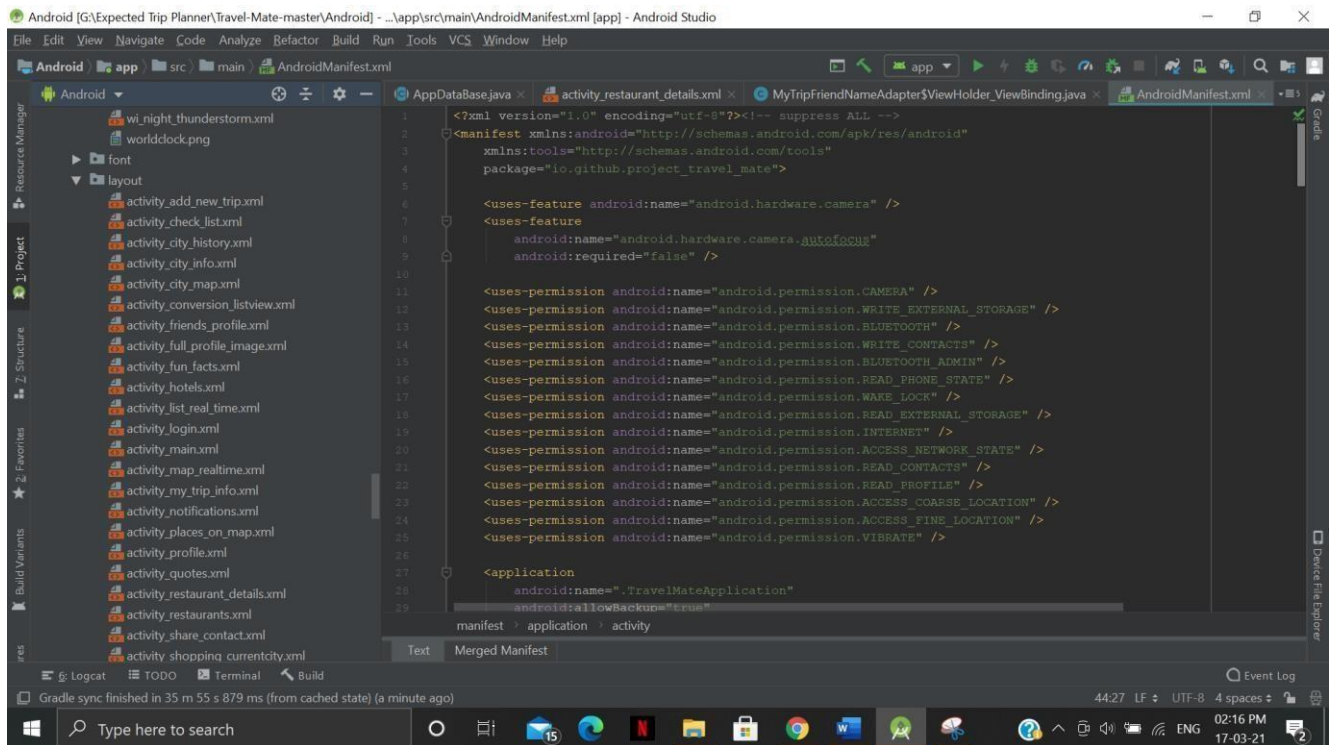


Figure 6.3: Android Manifest File

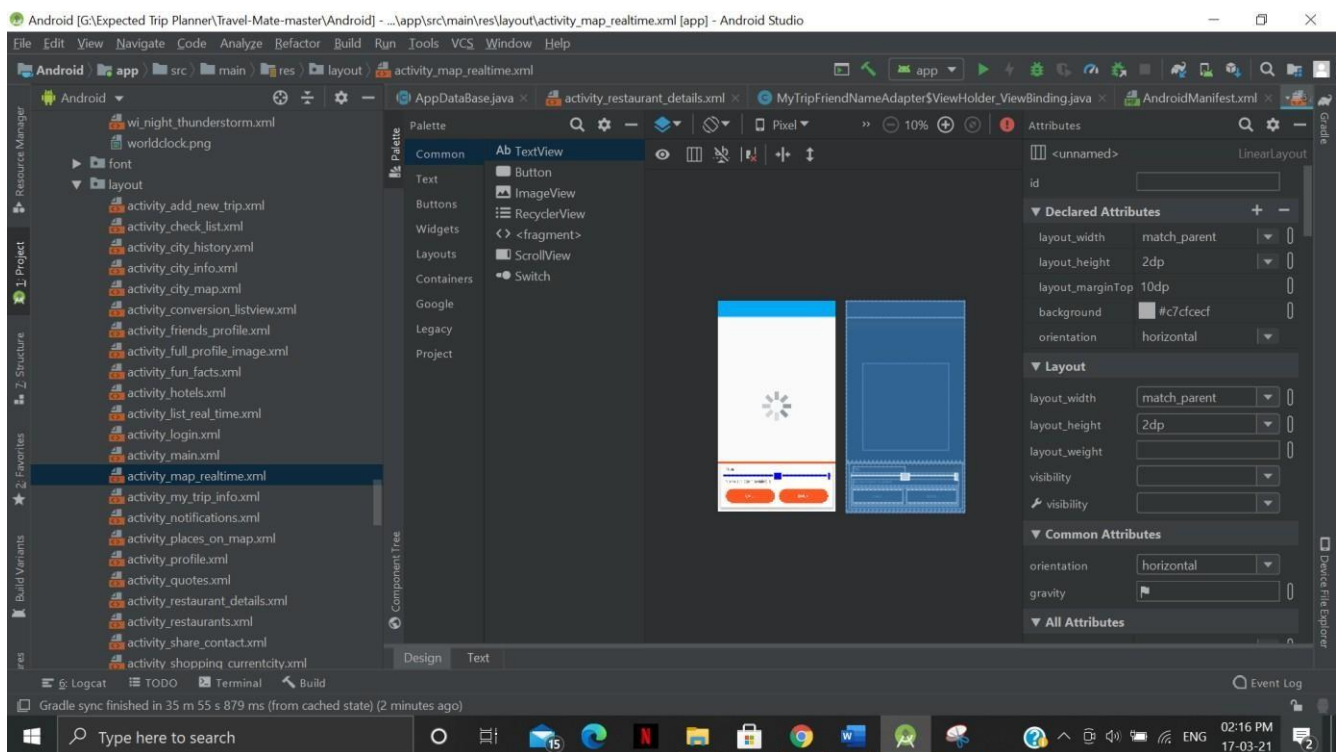


Figure 6.4: XML File

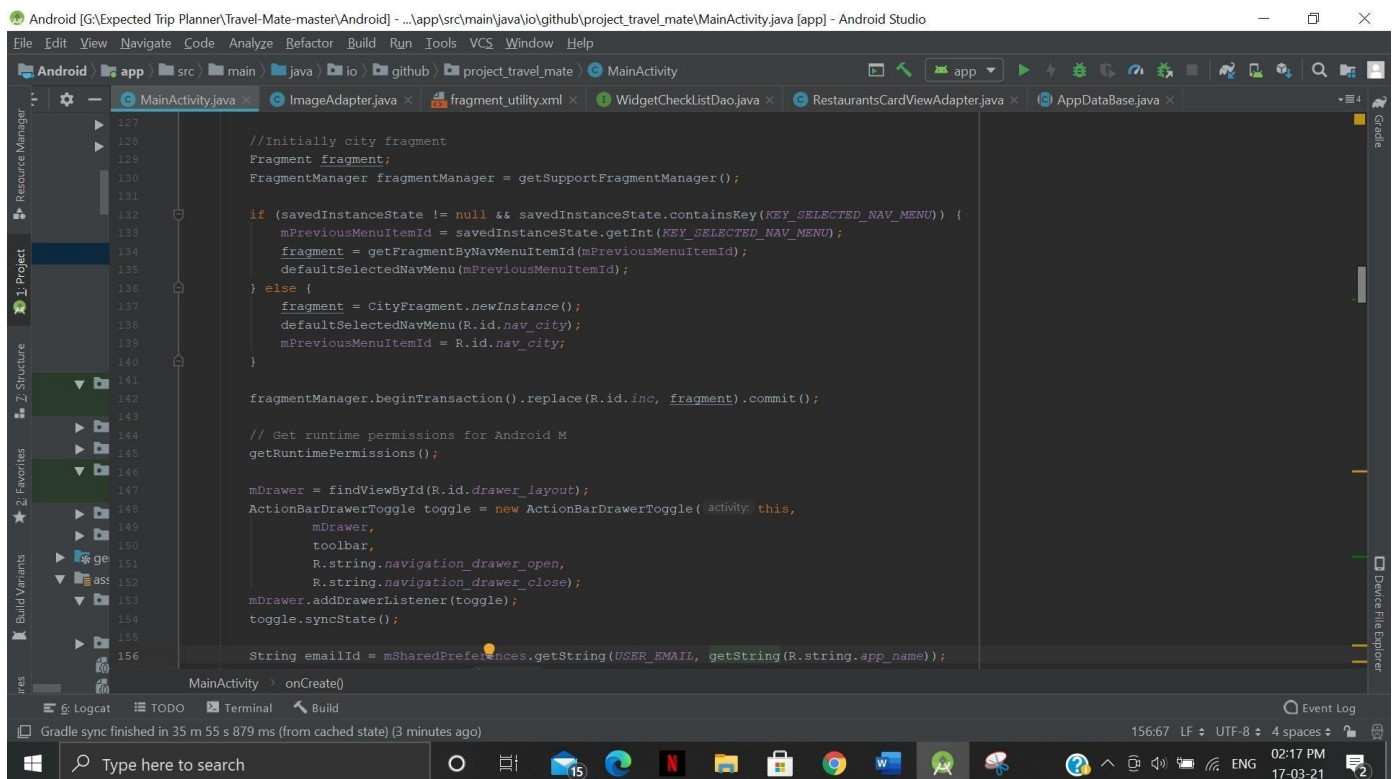


Figure 6.5: MainActivity Module

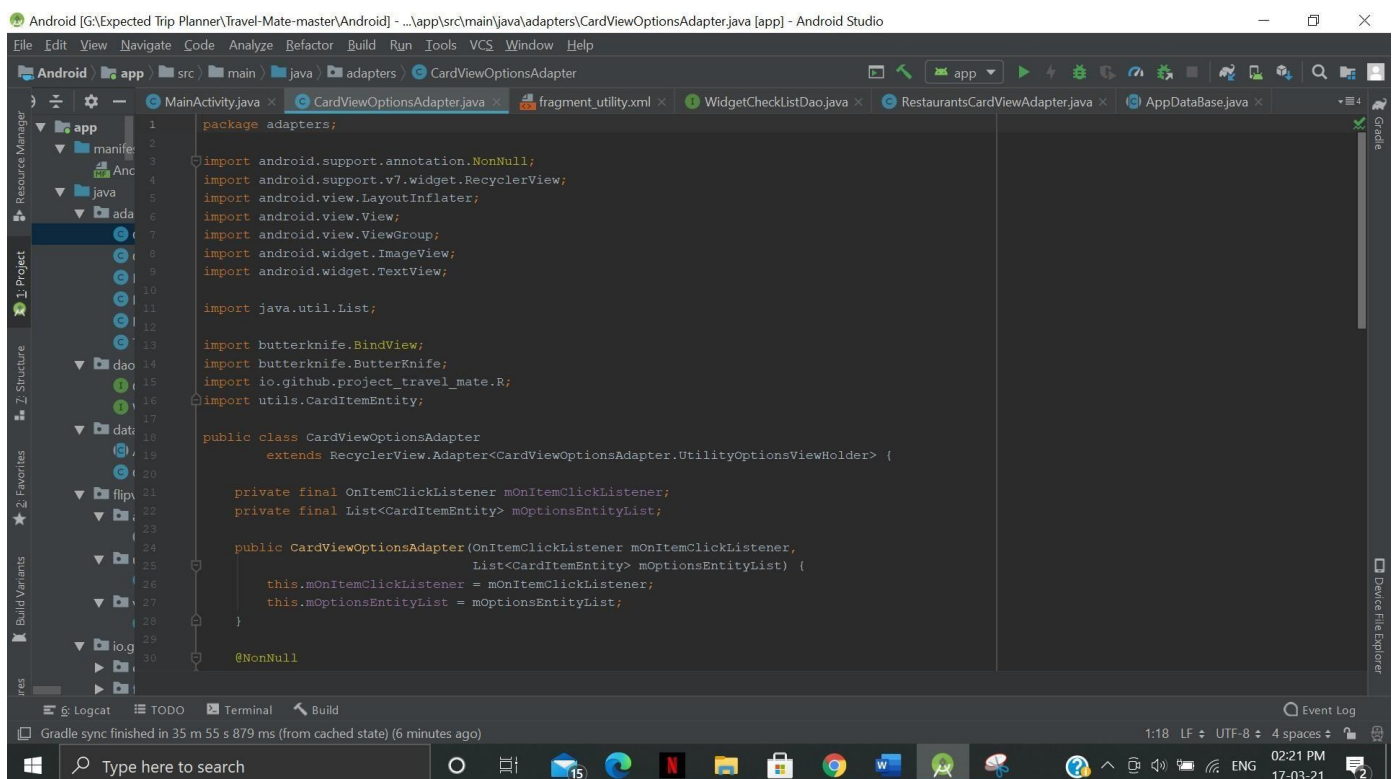


Figure 6.6: Adapters Module

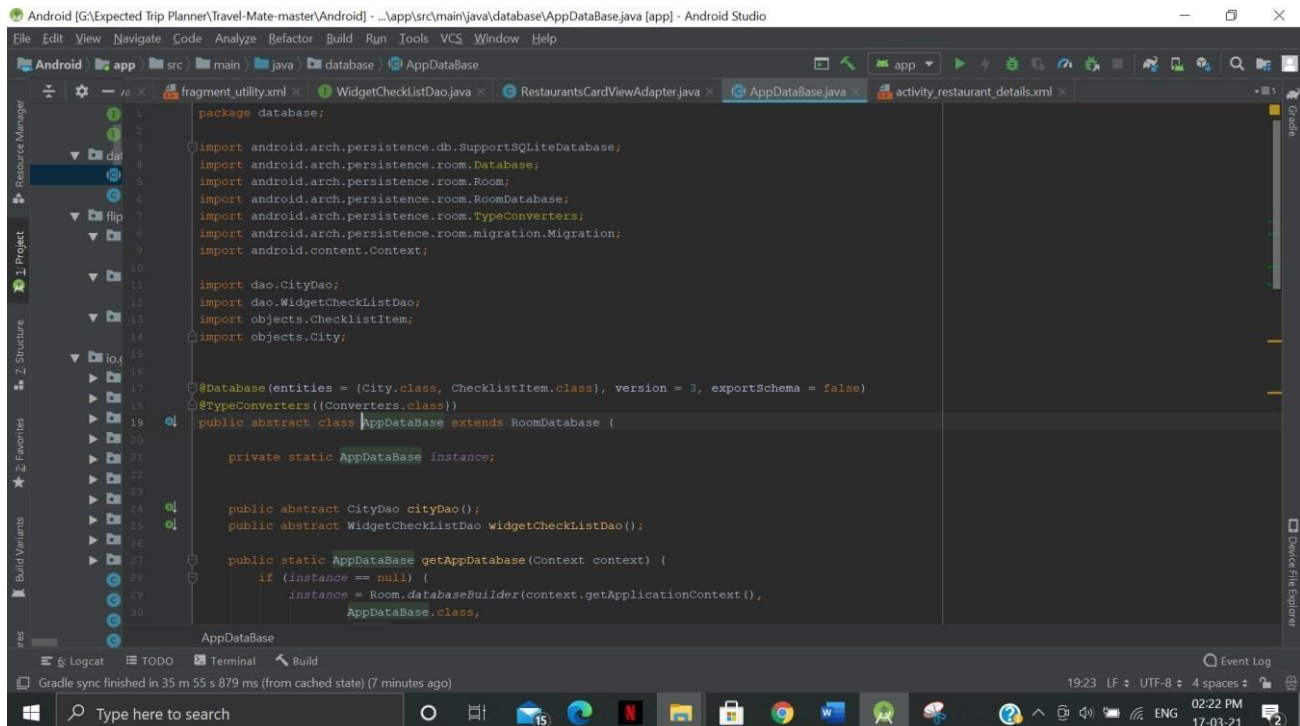


Figure 6.7: Database Integration

C.2 APPLICATION OUTPUT SCREENSHOTS

- Three categories are considered i.e., Tourist attractions, restaurants and utilities. Depending on the current or specified location, nearby locations are shown as per preferences differentiating between the three categories. Figure-6.10, 6.11, 6.12 are the application screenshots differentiating all the three categories
- Every city or locality is provided with a description which includes Food & Restaurants section, how to reach section, Hotels & accommodation and best to visit section. Figure 6.9 module is maintained to put information all together about a particular location.
- Figure 6.13 feature helps in providing information for people who would want to check on transportation details from one city to another by three means i.e., airplane, train and bus.
- As per Figure-6.16, users are given an option to wish list certain localities and view them later according their convenience.
- Navigation Panel is used to switch between different modules available in the application. Figure-6.15 gives the details about all the modules that are involved in the android application.

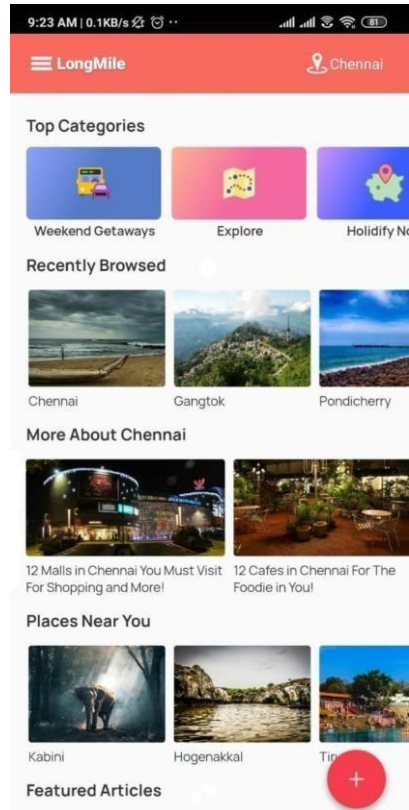


Figure 6.8: Home Screen

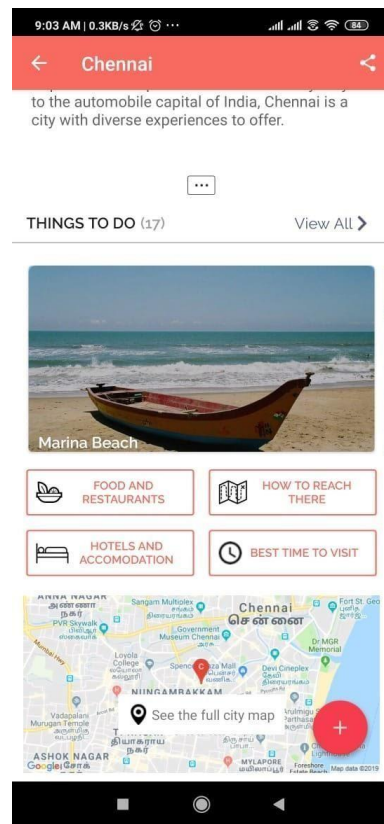
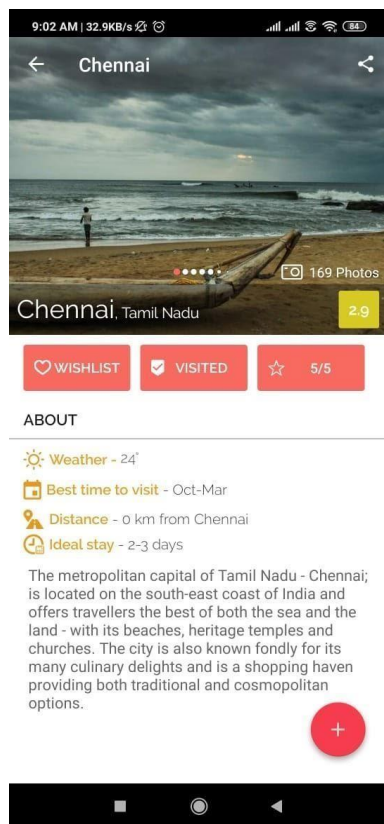


Figure 6.9: Individual City Description

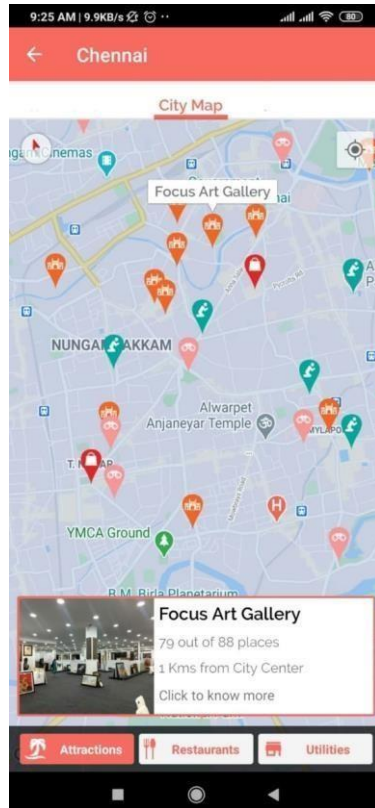


Figure 6.10: City Map Attractions

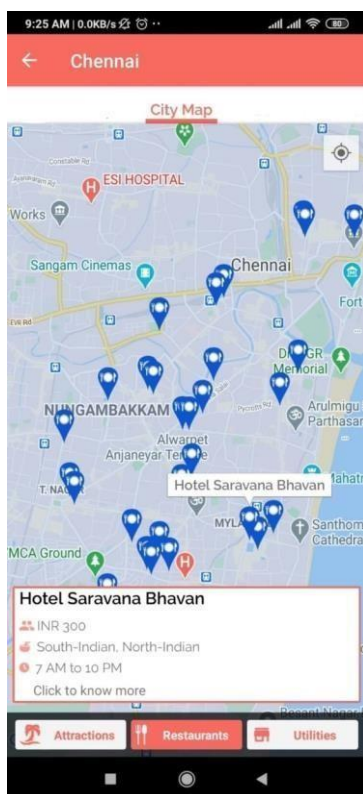


Figure 6.11: City Map F&B

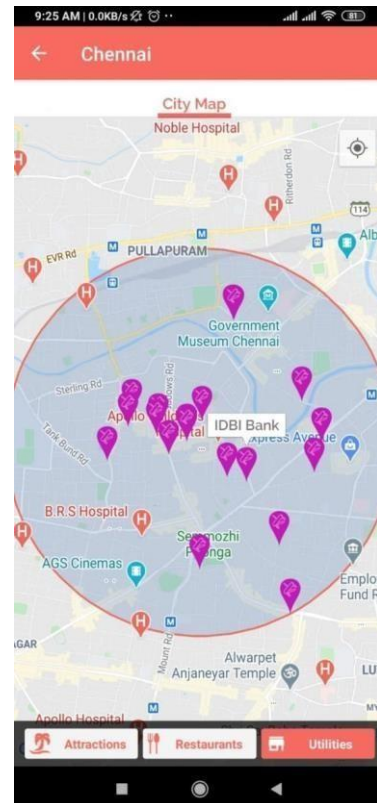


Figure 6.12: City Map Utilities

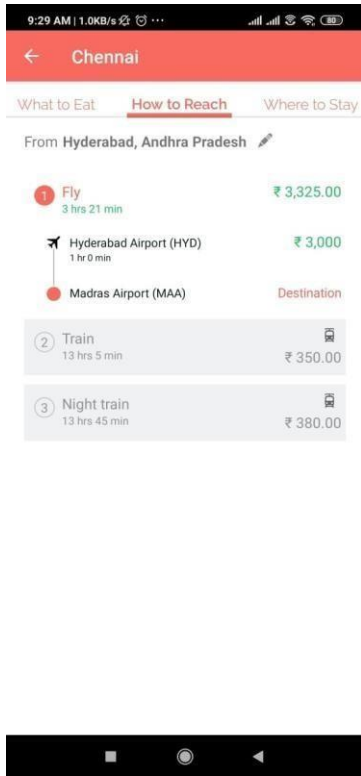


Figure 6.13: How To Reach



Figure 6.14: Best Time To Travel

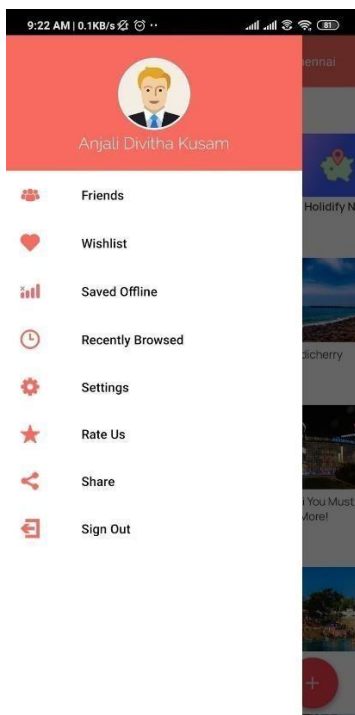


Figure 6.15: Navigation Panel

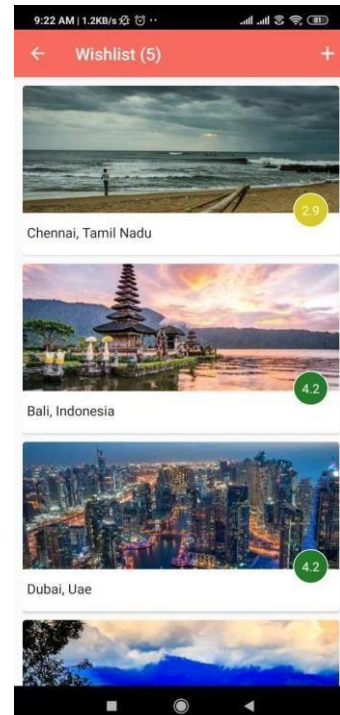


Figure 6.16: Wishlist

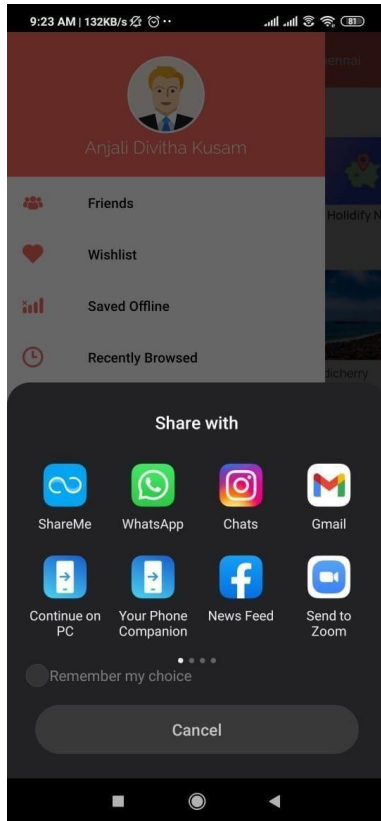


Figure 6.17: Share

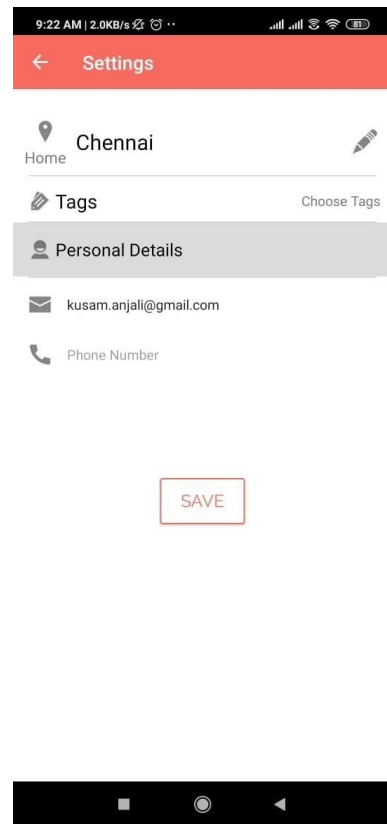


Figure 6.18: Personal Information

D. SOURCE CODE

D.1 Android Manifest File

```
<?xml version="1.0" encoding="utf-8"?><!-- suppress ALL -->
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="io.github.project_travel_mate">

    <uses-feature android:name="android.hardware.camera" />
    <uses-feature
        android:name="android.hardware.camera.autofocus"
        android:required="false" />

    <uses-permission android:name="android.permission.CAMERA" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.BLUETOOTH" />
    <uses-permission android:name="android.permission.WRITE_CONTACTS" />
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
    <uses-permission android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.WAKE_LOCK" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
    <uses-permission android:name="android.permission.READ_CONTACTS" />
    <uses-permission android:name="android.permission.READ_PROFILE" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.VIBRATE" />

    <application
        android:name=".TravelMateApplication"
        android:allowBackup="true"
        android:fullBackupContent="@xml/backup_descriptor"
        android:icon="@drawable/google_travel_logo"
```

```

        android:label="@string/app_name"
        android:largeHeap="true"
        android:screenOrientation="portrait"
        android:supportsRtl="true"
        android:theme="@style/AppTheme"
        tools:replace="android:fullBackupContent">
        <activity
            android:name=".mytrips.TripImageActivity"
            android:configChanges="orientation|screenSize"
            android:theme="@style/AppTheme" />
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name"
            android:theme="@style/AppThemeNoBar"
            android:configChanges="orientation|screenSize"
            android:windowSoftInputMode="adjustResize">

<intent-filter>
    <action android:name="android.intent.action.VIEW" />

    <category android:name="android.intent.category.DEFAULT" />
    <category android:name="android.intent.category.BROWSABLE" />

    <data android:host="project-travel-mate.github.io" />
    <data android:scheme="https" />
    <data android:scheme="http" />
    <data android:pathPattern=".*" />
</intent-filter>
</activity>

<service
    android:name="utils.GPSTracker"
    android:enabled="true" />

<activity
    android:name=".utilities.ChecklistActivity"
    android:label="@string/title_activity_select_city"
    android:theme="@style/AppTheme_picker" />
<activity
    android:name=".travel.ShoppingCurrentCityActivity"
    android:label="@string/title_activity_shopping_currentcity"
    android:theme="@style/AppTheme_picker" />
<activity
    android:name=".travel.HotelsActivity"
    android:configChanges="orientation|screenSize"
    android:label="@string/title_activity_hotels"
    android:theme="@style/AppTheme_picker" />

```

```

<activity
    android:name=".travel.HotelsActivity"
    android:configChanges="orientation|screenSize"
    android:label="@string/title_activity_hotels"
    android:theme="@style/AppTheme_picker" />
<activity
    android:name=".travel.MapViewRealTimeActivity"
    android:label="@string/title_activity_map"
    android:theme="@style/AppTheme_picker" />
<activity
    android:name=".SplashActivity"
    android:label="@string/app_name"
    android:theme="@style/AppThemeNoBar">
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>

    <meta-data
        android:name="android.app.shortcuts"
        android:resource="@xml/shortcuts" />
</activity>
<activity
    android:name=".destinations.funfacts.FunFactsActivity"
    android:label="@string/title_activity_fun_facts"
    android:theme="@style/AppTheme_picker" />
<activity android:name=".utilities.ShareContactActivity" />
<activity
    android:name=".login.LoginActivity"
    android:label="@string/title_activity_login"
    android:theme="@style/AppThemeNoBar"
    android:windowSoftInputMode="adjustPan"/>
<activity
    android:name=".destinations.description.FinalCityInfoActivity"
    android:theme="@style/AppThemeNoBar" />
<activity android:name=".destinations.description.PlacesOnMapActivity" />
<activity android:name=".mytrips.AddNewTripActivity" />
<activity
    android:name=".mytrips.MyTripInfoActivity"
    android:theme="@style/AppThemeNoBar"
    android:windowSoftInputMode="stateHidden" />
<activity android:name=".destinations.description.TweetsActivity" />
<activity
    android:name=".VerifyEmailActivity"
    android:label="@string/verify_email"
    android:theme="@style/AppThemeNoBar" />

```

```

<provider
    android:name="android.support.v4.content.FileProvider"
    android:authorities="io.github.project_travel_mate.shareFile"
    android:exported="false"
    android:grantUriPermissions="true">
    <meta-data
        android:name="android.support.FILE_PROVIDER_PATHS"
        android:resource="@xml/provider_paths" />
</provider>

<activity android:name=".ProfileActivity"
    android:configChanges="orientation|screenSize" />
<activity android:name=".friend.FriendsProfileActivity" />
<activity android:name=".notifications.NotificationsActivity" />
<activity
    android:name=".FullScreenImage"
    android:theme="@style/FullProfileImageTheme" />
<activity android:name=".utilities.WeatherForecastActivity" />
<activity android:name=".utilities.CompassActivity" />
<activity android:name=".utilities.CurrencyActivity" />
<activity android:name=".utilities.WorldClockActivity" />
<activity android:name=".utilities.CurrencyListViewActivity" />
<activity android:name=".utilities.TimezoneListViewActivity" />
<activity android:name=".destinations.description.WeatherActivity" />
<activity android:name=".utilities.UpcomingWeekendsActivity" />
<activity android:name=".destinations.description.CityHistoryActivity" />
<activity android:name=".destinations.description.CityMapActivity" />
<activity android:name=".travel.ListViewRealTimeActivity" />
<activity android:name=".destinations.description.RestaurantsActivity" />
<activity android:name=".destinations.description.TweetsDescriptionActivity" />
<activity android:name="com.theartofdev.edmodo.cropper.CropImageActivity" />
<activity android:name=".utilities.QuotesActivity"|
    android:theme="@style/Theme.AppCompat.NoActionBar"
/>
<activity
    android:name=".DeleteProfilePictureActivity"
    android:icon="@drawable/ic_delete_round"
    android:label="@string/remove_photo"
    android:theme="@android:style/Theme.NoDisplay">
    <intent-filter>
        <action android:name="android.intent.action.DELETE" />

        <category android:name="android.intent.category.DEFAULT" />
    </intent-filter>
</activity>
<activity
    android:name=".destinations.description.RestaurantDetailsActivity"
    android:theme="@style/AppThemeNoBar" />

```

```

        <receiver android:name="widget.ClockWidget">
            <intent-filter>
                <action android:name="android.appwidget.action.APPWIDGET_UPDATE" />
            </intent-filter>

            <meta-data
                android:name="android.appwidget.provider"
                android:resource="@xml/clock_widget_info" />
        </receiver>

        <activity android:name="widget.ClockWidgetConfigureActivity">
            <intent-filter>
                <action android:name="android.appwidget.action.APPWIDGET_CONFIGURE" />
            </intent-filter>
        </activity>
    </application>

</manifest>

```

D.2 XML Map Module

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.design.widget.CoordinatorLayout xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:id="@+id/map_real_time"
    android:fitsSystemWindows="true">

    <android.support.design.widget.AppBarLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:theme="@style/AppTheme.AppBarOverlay">

        <android.support.v7.widget.Toolbar
            android:id="@+id/toolbar"
            android:layout_width="match_parent"
            android:layout_height="?android:attr/actionBarSize"
            android:background="?android:attr/colorPrimary"
            app:popupTheme="@style/AppTheme.PopupOverlay" />

    </android.support.design.widget.AppBarLayout>
    <com.airbnb.lottie.LottieAnimationView
        android:id="@+id/animation"
        android:layout_width="300dp"
        android:layout_height="300dp"
        android:layout_gravity="center"
        app:lottie_autoPlay="true"
        app:lottie_loop="true" />

    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

```

```

xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/layout"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical"
app:layout_behavior="@string/appbar_scrolling_view_behavior"
tools:context="io.github.project_travel_mate.MainActivity">

<org.osmdroid.views.MapView
    android:id="@+id/map"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_weight="0.4"/>

<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="5dp"
    android:background="@color/colorAccent">

</LinearLayout>

<ScrollView
    android:id="@+id/data"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"



---


<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@drawable/shadow_1"
    android:gravity="center_horizontal">

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="@color/white"
    android:orientation="vertical"
    android:padding="10dp">

    <TextView
        android:id="@+id/item_title"
        android:layout_width="182dp"
        android:layout_height="wrap_content"
        android:layout_gravity="top"
        android:layout_marginStart="5dp"
        android:layout_marginTop="0dp"
        android:gravity="start"
        android:text="@string/text_title"
        android:textColor="#ff4a4a4a"
        android:textSize="15sp"
        android:textStyle="bold" />

```

```

    <LinearLayout
        android:layout_width="match_parent"

```

```

    android:layout_height="match_parent"
    android:orientation="vertical">

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="2dp"
    android:layout_marginTop="10dp"
    android:background="#c7cfcecf"
    android:orientation="horizontal" />

```

```

<TextView
    android:id="@+id/item_description"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="5dp"
    android:layout_marginTop="10dp"
    android:maxLines="10"
    android:text="@string/empty_description"
    android:textAppearance="?android:attr/textAppearanceSmall"
    android:textSize="15sp" />

```

```

<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="10dp"
    android:orientation="horizontal">

    <Button
        android:id="@+id/call"
        style="?android:attr/buttonBarButtonStyle"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_margin="4dp"
        android:layout_weight="1.0"
        android:background="@drawable/round_shape"
        android:text="@string/text_call"
        android:textColor="#ffffff" />

    <Button
        android:id="@+id/book"
        style="?android:attr/buttonBarButtonStyle"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_margin="4dp"
        android:layout_weight="1.0"
        android:background="@drawable/round_shape"
        android:text="@string/text_book"
        android:textColor="@color/white" />

</LinearLayout>

        </LinearLayout>

    </LinearLayout>

        </LinearLayout>
    </LinearLayout>

</ScrollView>

</LinearLayout>

</android.support.design.widget.CoordinatorLayout>

```

The project code involves 30 modules which is difficult to mention in the document. For reference, above is the code for Android Manifest File and XML Map Module.