NEW HORIZON
COLLEGE OF ENGINEERING
Autonomous College, Affiliated to VTU | Approved by AICTE New Delhi & UGC
Accredited by NAAC with 'A' Grade & Accredited by NBA

A MINI PROJECT REPORT

for

Mini Project in Mobile Application Development (20CSE77A)

on

TOUCH TO CURE

Submitted by

J S JAYASREE

USN: 1NH18CS077, Sem-Sec: 7-A

In partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

Academic Year: 2021-22

NEW HORIZON
COLLEGE OF ENGINEERING

Autonomous College, Affiliated to VTU | Approved by AICTE New Delhi & UGC
Accredited by NAAC with 'A' Grade & Accredited by NBA

CERTIFICATE

This is to certify that the mini project work titled

TOUCH TO CURE

submitted in partial fulfillment of the degree of Bachelor of Engineering in Computer Science and Engineering by

J S JAYASREE
USN:1NH18CS077

DURING

ODD SEMESTER 2021-2022

for

Course: Mini Project in Mobile Application Development-20CSE77A

Signature of Reviewer

Signature of HOD

SEMESTER END EXAMINATION

Name of the Examiner	*Signature with date*
1. _____	_____
2. _____	_____

ABSTRACT

Access to an affordable and working healthcare system is one of the aspects by which a country can be classified upon. Today, healthcare in India has become of the fastest growing sectors with respect to both employment and revenue.

But as this sector continues to grow, the poor and downtrodden sections of our society are being left behind. They have to make do with the crumbling public healthcare system which by all means would mostly fail due to lack of adequate infrastructure and doctors.

The COVID 19 pandemic has also taught the world about the importance of blood/plasma/organ banks that have played a key role in this battle against the virus. Ayurvedic medicines have proven to be really efficient in dealing with COVID 19 because of its lasting effects without causing any harm to us.

This mini-project solves the above-mentioned problems through the development of a mobile application. This mobile application would maintain a database of the users and other relevant information and would have a graphical user interface that would be simple and easy to use.

The entire program has been developed in Java and uses the Android Studio for running the Java application.

The mini-project is completely based on the high-level language, Java and the XML Scripting Language and uses GUI programming to provide a simple and easy to understand platform for the users.

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be impossible without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success.

I have great pleasure in expressing gratitude to **Dr. Mohan Manghnani**, Chairman, New Horizon Educational Institutions, for providing necessary infrastructure and creating good environment.

I take this opportunity to express my profound gratitude to **Dr. Manjunatha,** Principal, New Horizon College of Engineering, for his constant support and encouragement.

I would also like to thank **Dr. B. Rajalakshmi**, Professor and HOD, Department of Computer Science and Engineering, for her constant support.

I also express my gratitude to **Ms. Manjula A K,** Assistant Professor, Department of Computer Science and Engineering, my mini project reviewer, for constantly monitoring the development of the project and setting up precise deadlines. Her valuable suggestions were the motivating factors in completing the work.

J S Jayasree

USN: 1NH18CS077

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 ANDROID

Android is an open source and Linux-based Operating System for mobile devices such as smartphones and tablet computers. Android was developed by the Open Handset Alliance, led by Google, and other companies.

Android offers a unified approach to application development for mobile devices which means developers need only develop for Android, and their applications should be able to run on different devices powered by Android.

The first beta version of the Android Software Development Kit (SDK) was released by Google in 2007 where as the first commercial version, Android 1.0, was released in September 2008.

On June 27, 2012, at the Google I/O conference, Google announced the next Android version, 4.1 Jelly Bean. Jelly Bean is an incremental update, with the primary aim of improving the user interface, both in terms of functionality and performance.

The source code for Android is available under free and open-source software licenses. Google publishes most of the code under the Apache License version 2.0 and the rest, Linux kernel changes, under the GNU General Public License version 2.

Android Activity Lifecycle is controlled by 7 methods of android.app.Activity class. The android Activity is the subclass of ContextThemeWrapper class.

An activity is the single screen in android. It is like window or frame of Java.

By the help of activity, you can place all your UI components or widgets in a single screen.

The 7-lifecycle method of Activity describes how activity will behave at different states.

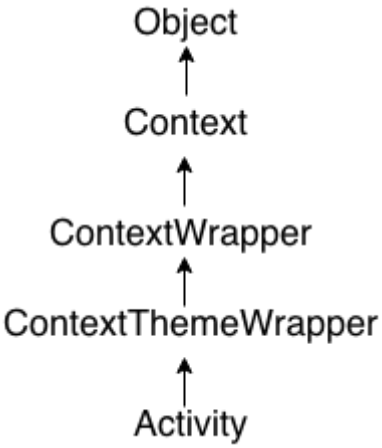

Fig 1.1a Behaviour of activity at different states

Android Activity Lifecycle methods

Let's see the 7 lifecycle methods of android activity.

Method	Description
onCreate	called when activity is first created.
onStart	called when activity is becoming visible to the user.
onResume	called when activity will start interacting with the user.
onPause	called when activity is not visible to the user.
onStop	called when activity is no longer visible to the user.
onRestart	called after your activity is stopped, prior to start.
onDestroy	called before the activity is destroyed.

Table 1 Lifecycle methods

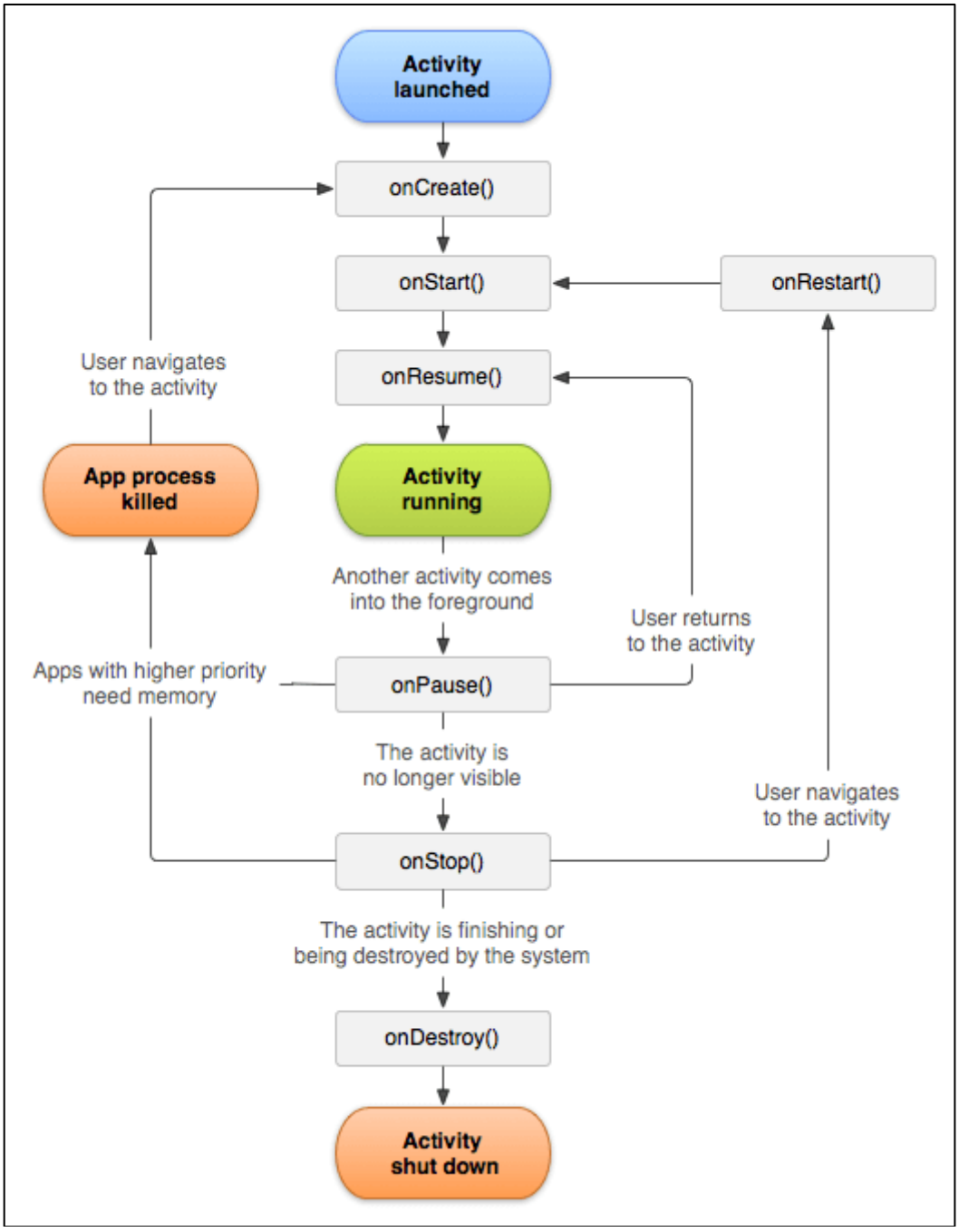

Fig 1.1b Android Activity Life Cycle

1.2 ANDROID ARCHITECTURE

Android architecture contains different number of components to support any android device needs. Android software contains an open-source Linux Kernel having collection of number of C/C++ libraries which are exposed through an application framework services.

Among all the components Linux Kernel provides main functionality of operating system functions to smartphones and Dalvik Virtual Machine (DVM) provide platform for running an android application.

The main components of android architecture are following:-

- Applications
- Application Framework
- Android Runtime
- Platform Libraries
- Linux Kernel

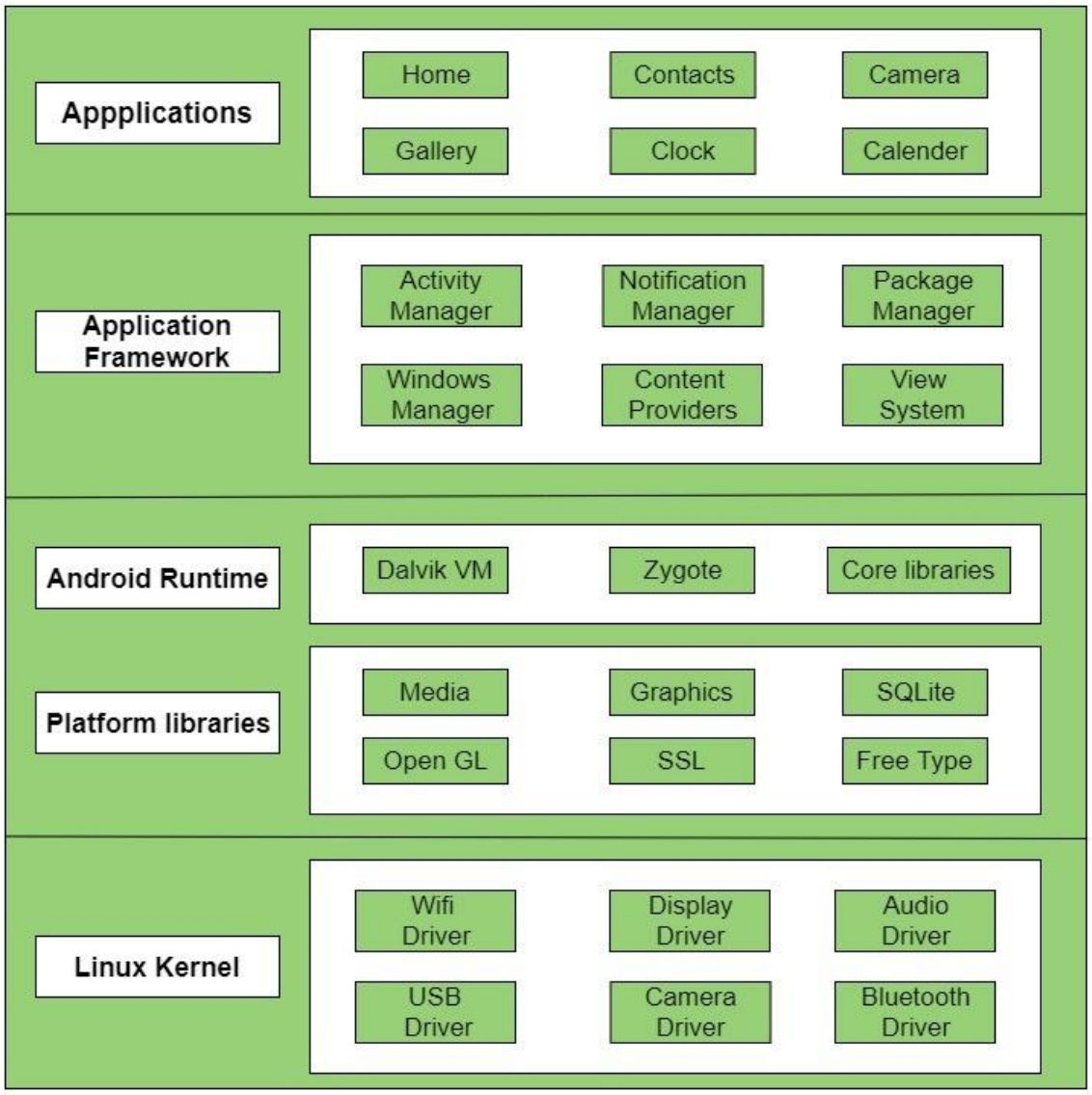

Fig 1.2a Pictorial representation of android architecture with several main components and their sub components

Applications –

Applications is the top layer of android architecture. The pre-installed applications like home, contacts, camera, gallery etc and third-party applications downloaded from the play store like chat applications, games etc. will be installed on this layer only. It runs within the Android run time with the help of the classes and services provided by the application framework.

Application framework –

Application Framework provides several important classes which are used to create an Android application. It provides a generic abstraction for hardware access and also helps in managing the user interface with application resources. Generally, it provides the services with the help of which we can create a particular class and make that class helpful for the Applications creation.

It includes different types of services activity manager, notification manager, view system, package manager etc. which are helpful for the development of our application according to the prerequisite.

Application runtime –

Android Runtime environment is one of the most important parts of Android. It contains components like core libraries and the Dalvik virtual machine (DVM). Mainly, it provides the base for the application framework and powers our application with the help of the core libraries.

Like Java Virtual Machine (JVM), Dalvik Virtual Machine (DVM) is a register-based virtual machine and specially designed and optimized for android to ensure that a device can run multiple instances efficiently. It depends on the layer Linux kernel for threading and low-level memory management. The core libraries enable us to implement android applications using the standard JAVA or Kotlin programming languages.

Platform libraries –

The Platform Libraries includes various C/C++ core libraries and Java based libraries such as Media, Graphics, Surface Manager, OpenGL etc. to provide a support for android development.

- **Media** library provides support to play and record an audio and video formats.
- **Surface manager** responsible for managing access to the display subsystem.

- **SGL** and **OpenGL** both cross-language, cross-platform application program interface (API) is used for 2D and 3D computer graphics.
- **SQLite** provides database support and **FreeType** provides font support.
- **Web-Kit** This open-source web browser engine provides all the functionality to display web content and to simplify page loading.
- **SSL (Secure Sockets Layer)** is security technology to establish an encrypted link between a web server and a web browser.

Linux Kernel –

Linux Kernel is heart of the android architecture. It manages all the available drivers such as display drivers, camera drivers, Bluetooth drivers, audio drivers, memory drivers, etc. which are required during the runtime.

The Linux Kernel will provide an abstraction layer between the device hardware and the other components of android architecture. It is responsible for management of memory, power, devices etc.

The features of Linux kernel are:

- **Security:** The Linux kernel handles the security between the application and the system.
- **Memory Management:** It efficiently handles the memory management thereby providing the freedom to develop our apps.
- **Process Management:** It manages the process well, allocates resources to processes whenever they need them.
- **Network Stack:** It effectively handles the network communication.

1.3 PROBLEM STATEMENT

"TOUCH TO CURE" Project emphasises that the Ayurvedic medicines are the best for illness. It lays down an inspiration down the line to consume herbal medicines. We all know that Ayurveda is based on the principle that to achieve and maintain health over the course of your life, you must re-balancing emotions, improve diet, practice yoga and "pranayama" (breathing exercises), and make lifestyle adjustments. This 5,000-year-old medicine system that originated in India has many life-altering benefits.

The main feature of this Project includes to provide a platform to buy and sell Ayurvedic medicines, to provide doctor consultation, to provide different diet plans and to provide Online Yoga. The recent witness of the Pandemic inspired me to adopt this idea to build an app for Ayurvedic medicines.

Significant amount of time is allocated for writing the customer details and category details as he/she needs to go through the records available and make rough estimate of the amount spent for current and previous services in a month. Also, the details of already availed user have to be maintained to recommend the necessary services to the Customer.

Thus, Touch to Cure allows for database server application to efficiently manage the data to ensure an elevated performance of the organisation.

Also, data security cannot be fortified by the manual copies.

Thus, the objective of project is to generate a clear database and to satisfy the customers with all the services and give them the best experience. It proves to be a great help for an individual in making the place where we live to be safe.

1.4 OBJECTIVES

The main purpose is to build mobile application which satisfies the customer with the services it provides. The recent witness of pandemic has brought the world to think about their cleanliness and hygiene right from home. But it would be tedious for some-one to do it all by themselves.

So, this application aims to provide best service in providing ayurvedic medicines to all customers. It believes “A satisfied customer is the best business strategy of all “.

The Project “Touch to Cure “uses the mobile framework developing languages to suc-cessfully build and deploy the application. The user can select the required services he wants and also can book it.

Also, Customer Detail has kept paper record in filing cabinets, it proves that monitoring individual customers on a monthly basis is difficult then managing a very large corporate, business agency with records on papers will be tedious and difficult to keep track of inventories with regards to the cost incurred and their details. The individual has to look for manual records for every operation or action to be performed. Thus the objectives of the Project include to develop a dynamic and engaging App for Ayurvedic medicines.

1.5 ADVANTAGES OF PROJECT

"Touch to Cure" project is mainly developed for promoting the use of Ayurvedic medicines and natural way of recovering from ailments like involving in Yoga activities.

The project also has lot of additional functionalities like to Book Doctor Appointment, to set reminders for consuming medicines, to teach basic exercises and also it helps to find nearby stores.

In comparison with the website Mobile Apps are easy to use and it is very much convenient to navigate to different functionalities.

The design of the user interfaces is very attractive and encourages the App users to gain best experience with the Mobile Application.

The usage of Notifications, Timer and Google Maps API based functionalities drags the customers into usage of the App and provides the best user experience.

It enables the customers to buy the product and the efficient use of Database allows to store all the user details.

CHAPTER 2

REQUIREMENT SPECIFICATION

2.1 MOBILE REQUIREMENTS

Android Version: 11

Hardware Version: RMX2027_11

Kernel Version: 4.14.186+

ColorOS Version : V11

Build Number : RMX2020_11_C.07

Processor : G70

RAM : 4.00 GB

2.2 SYSTEM REQUIREMENTS

- **Hardware Requirement:**
    1. Processor: Pentium processor, i3 or i5(sixth generation) (any one)
    2. Ram :4 GB or 8 GB (recommended)
    3. Other support(monitor(1080p), camera, etc)
- **Software Support:**
    1. Operating System: Windows 10 Home
    2. Developing Tool: Android Studio Arctic Fox (Version: 2020.3.1 Patch 3)

       Programming Language: Java Internet Language: XML
    3. Database Tool:

       MySQL Community Edition (Version :8.0)

CHAPTER 3

ANALYSIS AND DESIGN

3.1 ALGORITHM

The algorithm used for the development of the Project are as follows:

Step 1: The project incepts with the Home Page which has either LOGIN or REGISTRATION support asking the customer either to login or validate the credentials.

Step 2: The Home Page proceeds with the following actions:

Scenario 1: The new user has to go for registration after which the available services are displayed.

Scenario 2: The existing user should be authenticated with the password after which he can provide the ratings for the previous service done.

Step 3: Proceeding further we have all the services displayed on the Dashboard Activity page with different functionalities. Here the possible actions are:

a) To check the account page of the user.

b) To book doctor appointment

c) To buy medicines

d) To do diet plans

e) To promote Yoga practices

Step 4: Selecting any of the actions navigates through different activities.

Step 5: The personal details of the user their name, date of birth is collected and efficiently stored in a SQL (Structured Query Language) database. This proper management of resources results in successful operation of the Clinic.

Step 6: The effective utilization of the database will result in an efficient maintenance of the Customer details and thus ensuring Customer Relationship.

3.2 FLOW CHART

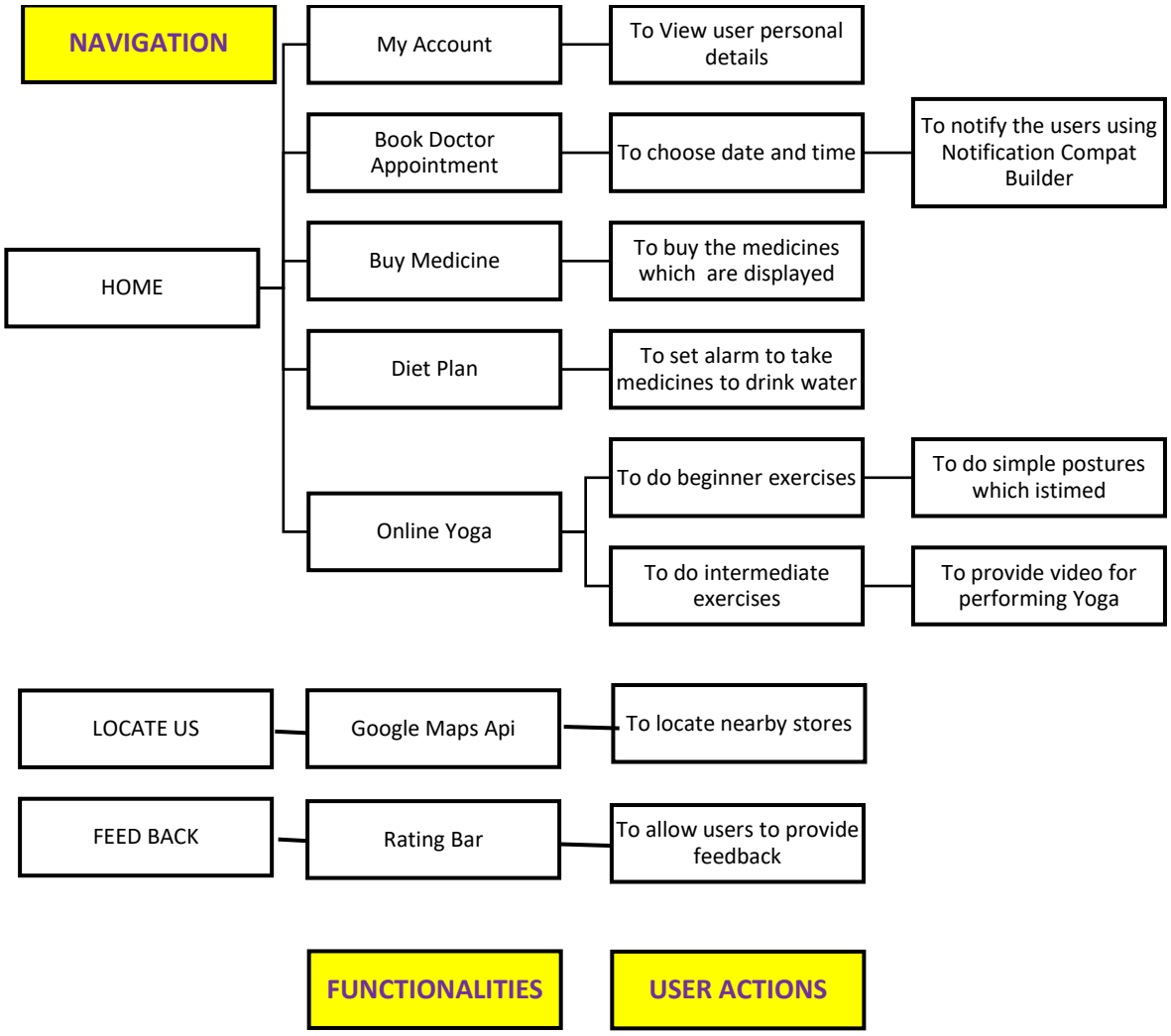

Fig 3.2a Flowchart showing the workflow of Project

CHAPTER 4

IMPLEMENTATION

4.1 ANDROID CONCEPTS USED

The basic building block for user interface is a View object which is created from the View class and occupies a rectangular area on the screen and is responsible for drawing and event handling. View is the base class for widgets, which are used to create interactive UI components like buttons, text fields, etc.The ViewGroup is a subclass of View and provides invisible container that hold other Views or other ViewGroups and define their layout properties.At third level we have different layouts which are subclasses of ViewGroup class and a typical layout defines the visual structure for an Android user interface and can be created either at run time using View/ViewGroup objects or you can declare your layout using simple XML file activity_main.xml which is located in the res/layout folder of your project.

“Touch to Cure” uses different layouts efficiently to generate the best user interface. Activities are properly synchronized to give best functionality to the user. The different User Interface elements are used. Notifications are used by the Mobile App to alert the user with the confirmation of booking Appointment. Google Map Location based API Service is used to locate near by stores. Alarm Manager class is used to set the alarm for the functionality which provides diet to the App users. Explicit intents are used to navigate from one activity to another activity within the App. Android Fragment is the part of activity, it is also known as sub-activity. There can be more than one fragment in an activity. Fragments represent multiple screens inside one activity. Android fragment lifecycle is affected by activity lifecycle because fragments are included in activity. Each fragment has its own life cycle methods that is affected by activity life cycle because fragments are embedded in activity.

The FragmentManager class is responsible to make interaction between fragment objects. Here the above concepts are explained in detail.

Android Notification provides short, timely information about the action happened in the application, even it is not running. The notification displays the icon, title and some amount of the content text. The properties of Android notification are set using **NotificationCompat.Builder** object. Some of the notification properties are mention below:

- **setSmallIcon()**: It sets the icon of notification.
- **setContentTitle()**: It is used to set the title of notification.
- **setContentText()**: It is used to set the text message.
- **setAutoCancel()**: It sets the cancellable property of notification.
- **setPriority()**: It sets the priority of notification.

Android location APIs make it easy for you to build location-aware applications, without needing to focus on the details of the underlying location technology. This becomes possible with the help of Google Play services, which facilitates adding location awareness to your app with automated location tracking, geofencing, and activity recognition. The Location object represents a geographic location which can consist of a latitude, longitude, time stamp, and other information such as bearing, altitude and velocity. To get the current location, create a location client which is LocationClient object, connect it to Location Services using connect() method, and then call its getLastLocation() method. This method returns the most recent location in the form of Location object that contains latitude and longitude coordinates and other information.

Android Intent is the *message* that is passed between components such as activities, content providers, broadcast receivers, services etc. It is generally used with startActivity() method to invoke activity, broadcast receivers etc.The **dictionary meaning** of intent is *intention or purpose*. So, it can be described as the intention to do

action.The LabeledIntent is the subclass of android.content.Intent class. Types of Android Intents.There are two types of intents in android: implicit and explicit.

1) Implicit Intent

Implicit Intent doesn't specifiy the component. In such case, intent provides information of available components provided by the system that is to be invoked.

2) Explicit Intent

Explicit Intent specifies the component. In such case, intent provides the external class to be invoked.

Broadcast in android is the system-wide events that can occur when the device starts, when a message is received on the device or when incoming calls are received, or when a device goes to airplane mode, etc. Broadcast Receivers are used to respond to these system-wide events. Broadcast Receivers allow us to register for the system and application events, and when that event happens, then the register receivers get notified. There are mainly two types of Broadcast Receivers:

- **Static Broadcast Receivers:** These types of Receivers are declared in the manifest file and works even if the app is closed.
- **Dynamic Broadcast Receivers:** These types of receivers work only if the app is active or minimized.

Since from API Level 26, most of the broadcast can only be caught by the dynamic receiver, so we have implemented dynamic receivers in our sample project given below. There are some static fields defined in the Intent class which can be used to broadcast different events.

4.2 FUNCTIONALITY OF THE PROJECT

"Touch to Cure" Project uses three different navigation section which includes Home, Locate Us and Feedback. Home Activity screen incepts the user with sign in for the existing user and register for the new user. Once after successful login the following functionalities are implemented:

My Accounts – This Activity Page consists of the user information who has registered with the Mobile Application.

Book Doctor Appointment – This activity screen involves guiding the user to book the doctor appointment with the date and time. Once the appointment is booked a notification is displayed to the user confirming the appointment.

Buy Medicine – This Page allows the App users to buy the required medicines.

Diet Plan – This allows the users to set the Alarm for consuming medicines, drinking water etc

Online Courses – This guides the users with the different postures of the Yoga.

Locate Us- This functionality uses Google Apps API to guide the users to locate nearby Ayurvedic Clinic.

Feedback – It allows users to provide feedback about the App usage.

4.3 SQLite Database Connectivity

SQLite is an open-source relational database i.e. used to perform database operations on android devices such as storing, manipulating or retrieving persistent data from the database.It is embedded in android by default. So, there is no need to perform any database setup or administration task. ndroid SQLite native API is not JDBC, as JDBC might be too much overhead for a memory-limited smartphone. Once a database is created successfully its located in **data/data//databases/** accessible from Android Device Monitor.SQLite is a typical **relational database**, containing tables (which consists of rows and columns), indexes etc. We can create our own tables to hold the data accordingly. This structure is referred to as a **schema**.

Create a Database

Simple steps to create a database and handle are as follows.

1. Create "SQLiteDatabase" object.
2. Open or Create a database and create a connection.
3. Perform insert, update or delete operation.
4. Create a Cursor to display data from the table of the database.
5. Close the database connectivity.

SQLiteOpenHelper class provides the functionality to use the SQLite database.

The android.database.sqlite.SQLiteOpenHelper class is used for database creation and version management. For performing any database operation, you have to provide the implementation of onCreate() and onUpgrade() methods of SQLiteOpenHelper class. SQLiteOpenHelper wraps up this logic to create and upgrade a database as per our specifications. For that we'll need to create a custom subclass of SQLiteOpenHelper implementing at least the following three methods.

1. **Constructor**: This takes the Context (e.g., an Activity), the name of the database, an optional cursor factory (we'll discuss this later), and an integer representing

the version of the database schema you are using (typically starting from 1 and increment later).

2. **onCreate(SQLiteDatabase db)** : It's called when there is no database and the app needs one. It passes us a SQLiteDatabase object, pointing to a newly-created database, that we can populate with tables and initial data.

3. **onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)** : It's called when the schema version we need does not match the schema version of the database, It passes us a SQLiteDatabase object and the old and new version numbers. Hence, we can figure out the best way to convert the database from the old schema to the new one.

Android SQLite Cursor

A Cursor represents the entire result set of the query. Once the query is fetched a call to cursor.moveToFirst() is made. Calling moveToFirst() does two things:

- It allows us to test whether the query returned an empty set (by testing the return value)
- It moves the cursor to the first result (when the set is not empty)

Sr.No	Method & Description
1	openDatabase(String path, SQLiteDatabase.CursorFactory factory, int flags, DatabaseErrorHandler errorHandler) This method only opens the existing database with the appropriate flag mode. The common flags mode could be OPEN_READWRITE OPEN_READONLY
2	openDatabase(String path, SQLiteDatabase.CursorFactory factory, int flags) It is similar to the above method as it also opens the existing database but it does not define any handler to handle the errors of databases
3	openOrCreateDatabase(String path, SQLiteDatabase.CursorFactory factory) It not only opens but create the database if it not exists. This method is equivalent to openDatabase method.
4	openOrCreateDatabase(File file, SQLiteDatabase.CursorFactory factory) This method is similar to above method but it takes the File object as a path rather then a string. It is equivalent to file.getPath()
5	execSQL(String sql, Object[] bindArgs) This method not only insert data , but also used to update or modify already existing data in database using bind arguments
6	getColumnCount() This method return the total number of columns of the table.
7	getColumnIndex(String columnName) This method returns the index number of a column by specifying the name of the column
8	getColumnName(int columnIndex) This method returns the name of the column by specifying the index of the column
9	getColumnNames() This method returns the array of all the column names of the table.
10	getCount() This method returns the total number of rows in the cursor
11	getPosition() This method returns the current position of the cursor in the table
12	isClosed() This method returns true if the cursor is closed and return false otherwise

Table 2 Depicts the different methods to handle SQLite events

CHAPTER 5

SAMPLE OUTPUT

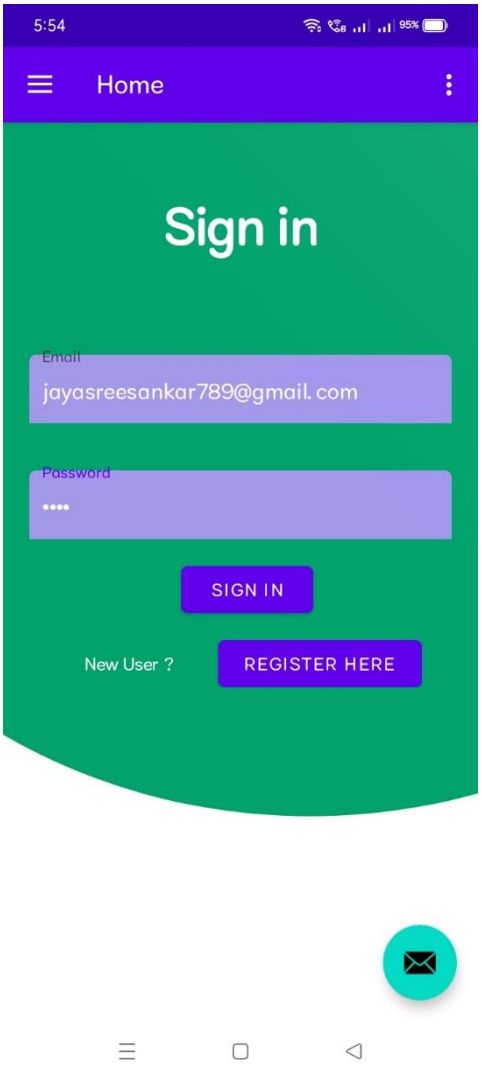

Fig 5.1a Output showing the login

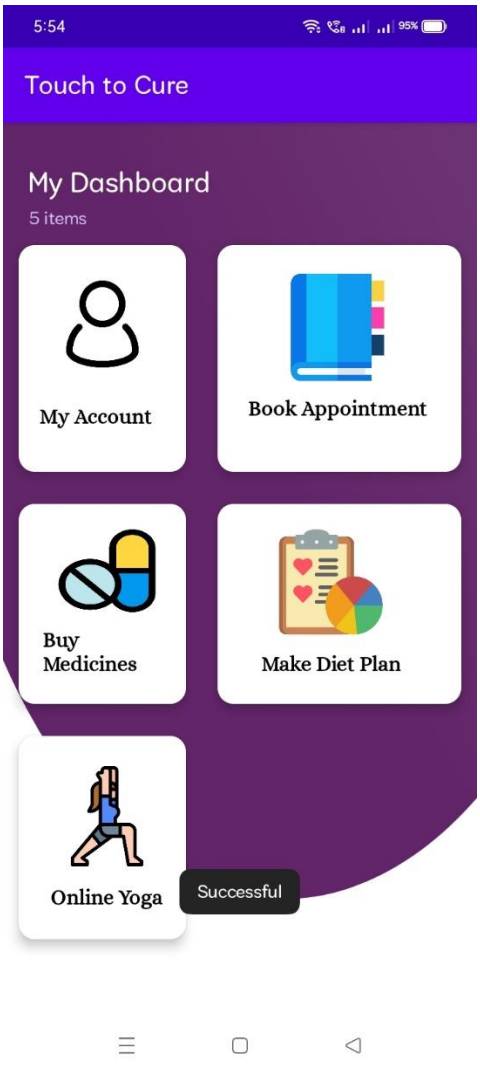

Fig 5.1b Output showing the Dashboard

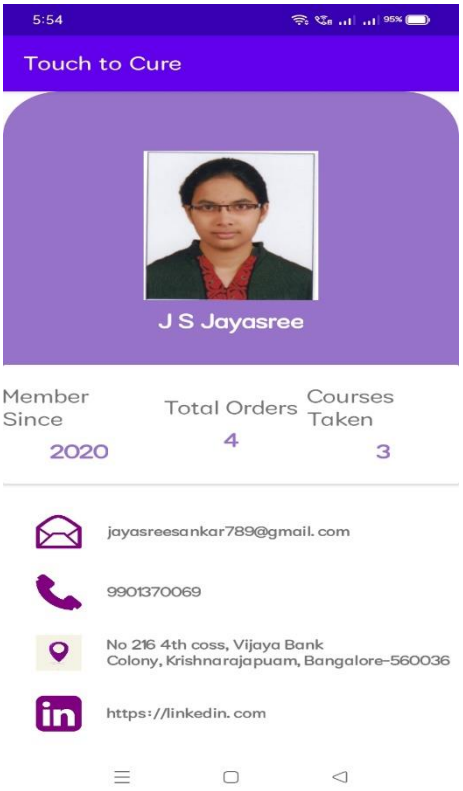

Fig 5.1c Output showing My Accounts Page

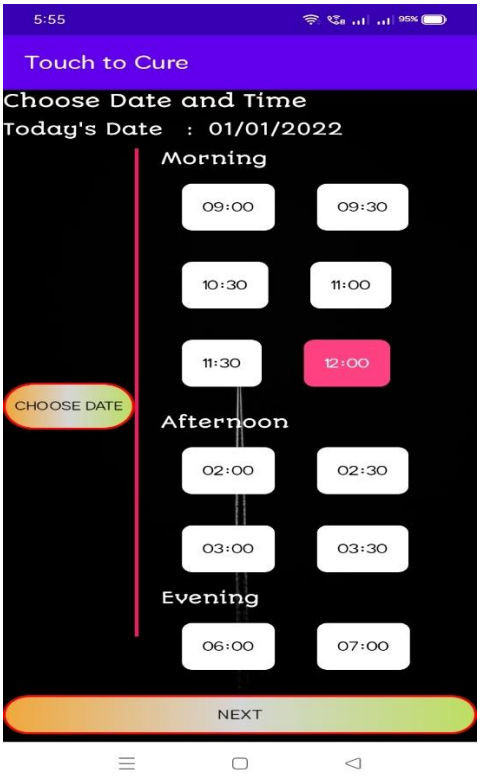

Fig 5.1d User Action to book Appointment

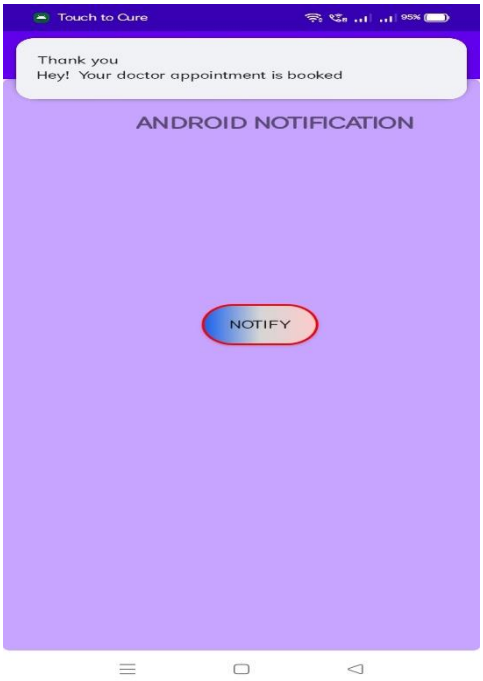

Fig 5.1e Output showing the Notification

Fig 5.1f Output showing the Add Cart Activity

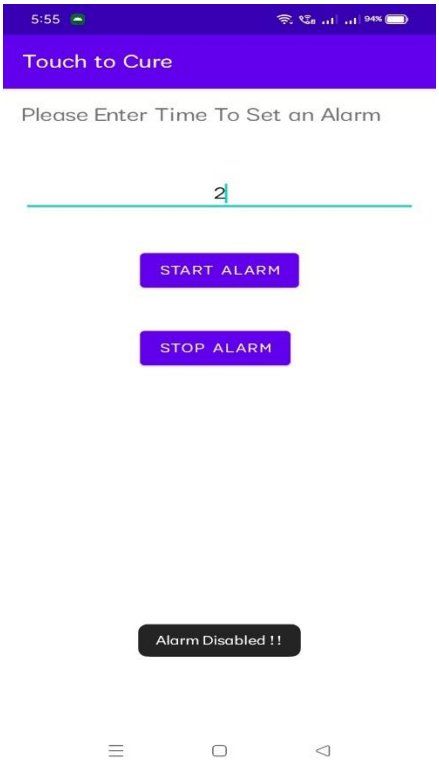

Fig 5.1g Output showing the Alarm

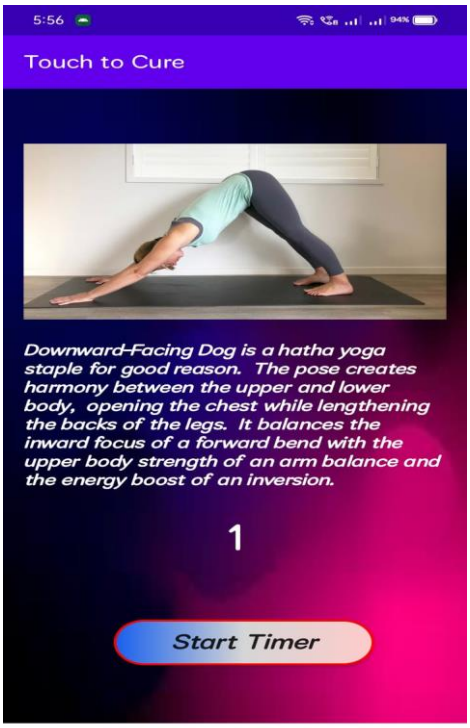

Fig 5.1h Output showing the start of timer

Fig 5.1i Output showing the end of timer

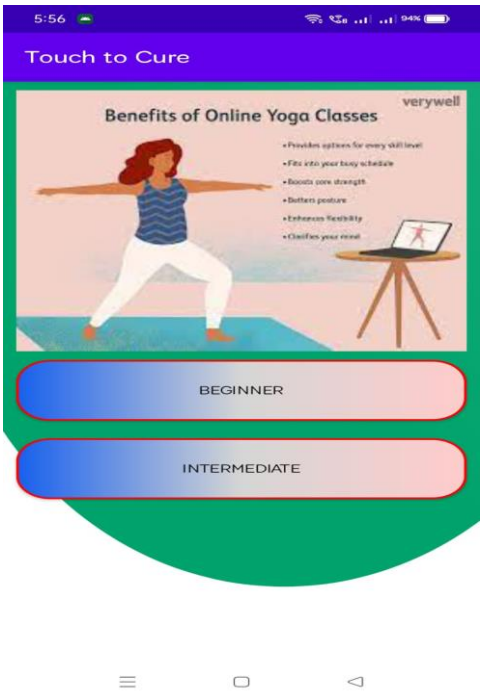

Fig 5.1j Output showing the Course Page

Fig 5.1k Output showing the video

Fig 5.1l Output Showing the Navigation bar

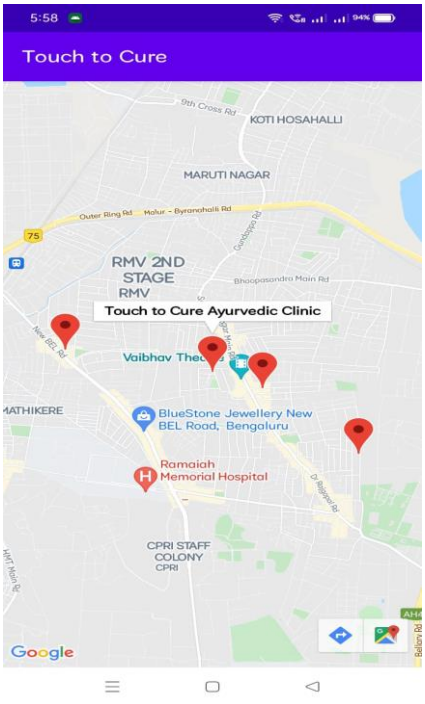

Fig 5.1m Output showing the google map

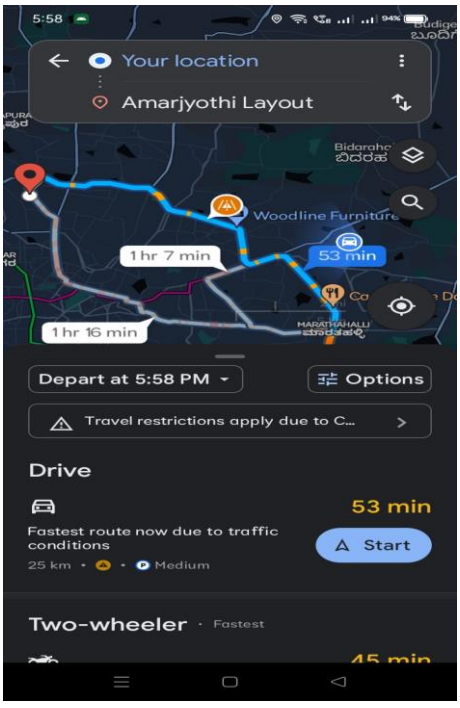

Fig 5.1n Output showing the directions

CONCLUSION

Thus, the following conclusions can be drawn from the Project" Touch to Cure":

- Touch to Cure was developed to ensure the safety of everyone considering the current situation of pandemic.

- The information gathered during the data collection was properly analyzed and the results provided the basis for the new system.

- The system was tested and found to be functional and the outputs produced by this system were encouraging.

- The application will hence reduce the loss of information unlike the existing system and also information will be processed fast.

- Effective implementation of this software will take care of the basic requirements of the customer management system because it is capable of providing easy and effective storage of information related to activities happening in the stipulated area. With these, the objectives of the system design will be achieved.

- In order to allow for future expansion, the system has been designed in such a way that will allow possible modification as it may deem necessary by the user system, whenever the idea arises.

REFERENCES

[1] https://www.jackrutorial.com/2018/03/android-login-with-sqlite-database-tutorial.html

[2] https://developer.android.com/reference/android/content/Intent

[3]https://developers.google.com/android/reference/com/google/android/gms/location/LocationServices

[4] https://developer.android.com/guide/topics/ui/notifiers/notifications

[5] https://developer.android.com/guide/fragments