G40_R

by Sneh Aswal

Submission date: 13-Oct-2021 01:48PM (UTC+0530)
Submission ID: 1672720464
File name: G40_Final_Report1.1.doc (3.85M)
Word count: 3881
Character count: 20406

FINAL PROJECT REPORT (CS421)

ON

GEOLOCATOR APPLICATION

A report submitted in partial fulfilment of the requirement for the award of

The degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

Submitted to:

Mohd Zia Ur Rehman

Assistant Professor

Submitted by:

Siddhant Garg-180102194-C
Mohd Ahad-18102257-D
Mohd Mustafa Anwar-180102148-C
Aditya Verma-180102251-D

SCHOOL OF COMPUTING

DIT UNIVERSITY, DEHRADUN

(State Private University through State Legislature Act No. 10 of 2013 of Uttarakhand and approved by UGC)

Mussoorie Diversion Road, Dehradun, Uttarakhand - 248009, India.

Nov 2021

CANDIDATES DECLARATION

I hereby certify that the work, which is being presented in the Report, entitled **Geolocator Application**, in partial fulfilment of the requirement for the award of the Degree of **Bachelor of Technology** and submitted to the DIT University is an authentic record of my work carried out during the period ***17/08/2020*** to ***4/11/2021*** under the guidance of Mohd Zia Ur Rehman.

Date: 03/11/2020

Siddhant Garg

Mohd Ahad

Mohd Mustafa Anwar

Aditya Verma

Signature of Candidate's

Guide: Mohd Zia Ur Rehman

Signature of Guide

ACKNOWLEDGEMENT

We take this opportunity to express our profound gratitude and deep regards to our guide **Mohd Zia Ur Rehman** (Assistant Professor) for his exemplary guidance, monitoring and constant encouragement throughout the course of this thesis. The blessing, help and guidance given by him time to time shall carry us a long way in the journey of life on which we are about to embark.

We are grateful to our Head of Department – **Dr Vishal Bharti** Sir for their support and guidance in our project. We started out this project with exhilaration of embark on a new and long journey, with exploring new course of action that the filed of computerized operations offers.

Lastly, we thank almighty, our parents, brother, and friends for their constant encouragement without which this project would not be possible.

NAME	**ROLL NO**
Siddhant Garg	180102194
Mohd Ahad	180102257
Mohd Mustafa Anwar	180102148
Aditya Verma	180102251

ABSTRACT

The location-based services apps are already in trend and have been dyed-in-the-wool in the mobile app market. Geo-location marks in mobile apps have benefited these types of new applications start-up ideas and to set up businesses with this technology. The knotty probe of today is “Where are you, now?”

This question can be answered with the help of a GPS based smartphone having a Geo-location app. There are several courses of action to integrate location-based services in an existing mobile app. At every spot, a person is widely using Geo location apps to update his/her present position on social media and also for many other purposes. These types of apps are often helpful for users and money making for creators to make improvements. Geo location feature is now a requisite as it will help to provide the services easily and improve all-embracing user experience.

Besides it, navigation and survey apps can also suggest different venues to the user which will depend on the place they want to go. It delivers a review of those places and tells users the right path to choose while going on. The users could place recommendations in the place which they have visited. We are usually unable to visit any place shown in advertisements. Therefore, to get speculation, recommendation apps can be used before implementation of Geo-location features.

List of Abbreviations:

GPS	Global Positioning System
API	Application Program Interface
Wi-fi	Wireless Fidelity
LAC	Location Area Code
IDE	Integrated Development Environment
RAM	Random Access Memory
IP	Internet Protocol
XML	eXtensible Markup Language
OS	Operating System

Chapter 1

INTRODUCTION

Android is a cell working device that is based on model of the Linux kernel and different open-source software program. those have been designed predominantly for touchscreen cell devices together with smartphone and tablet. Android gadgets encompass many elective hardware additives, video cameras, GPS, orientated sensors and devoted gaming controls, accelerometers, strain sensors, touchscreens and so forth. For working systems, clever telephones and tablets, there wished a substitute. consequently, after the 12 months 2000, OS as Android and Blackberry have been citified. With the lolling of time, amongst all the finest and extensively used cellular running structures there is android. Android Inc. became constituted through Andy Rubin, wealthy miner, and Chris White in Palo Alto of California, U.S. in early 2000. Then, in 2005, Android became picked up by way of Google and with the evolution in Android working system, endless variations were introduced. Now there have been a big range of improvements in the unique model of Android. Android running system offers the pliancy for both the customers and the builders.

The phrase Geo-place suggests to the identity of the geographic place of person and additionally a computing tool through an expansion of statistics series mechanisms. on the whole, Geo-vicinity offerings use inner GPS gadgets or network routing addresses to decide the area. Geo-region is a tool-unique API, i.e., browsers or gadgets assist Geo-region if you want to use it via web programs. If permission is accord, the Geo-location facts is on the market for your scripts and third-celebration scripts protected inside the web page and letting your software sway the place and maintain because the person movements round. site statistics is approximate and has been collected from IP cope with, cellular towers, wireless network, GPS or maybe the manual records entries. here the Geo-place API does not care how the patron sway the area or the data is obtained in not unusual. considering ascertainment of the vicinity can take time, the Geo-place API controls the time of operations.

CHAPTER 2

PROJECT DESCRIPTION

2.1 PURPOSE

Nowadays, we have various applications which uses location to show our results as per our requirements and use.

For example, we can use app for tickets for movies, it uses our location to track the theatres present in our proximity.

There are different cases of apps like coupon application which show us offers and discount on the places we visit rather than the places we don’t visit

In vast majority, geo-location features in social-media websites/apps, which helps us to punch in our as well as our friend’s location.

Several applications like OLA, UBER, Maps by Google, etc uses this feature to help you to reach your destination by avoiding roads that are under construction, blocked, heavily trafficked, because it shows you the optimal or the best path/road to move to your destination

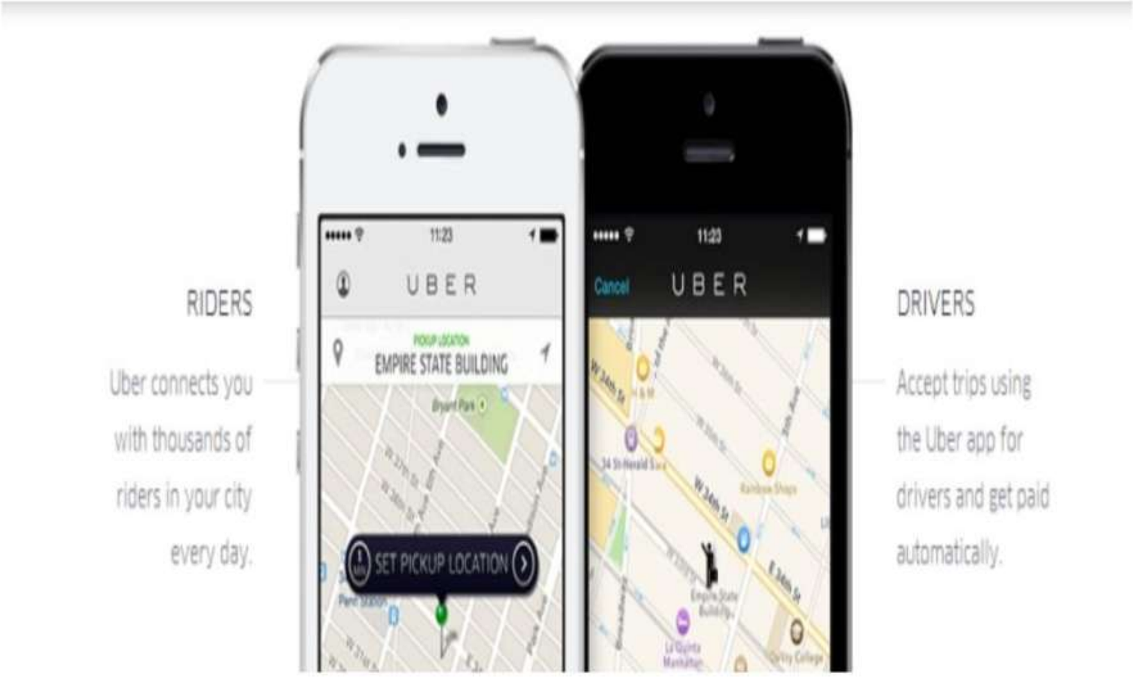

RIDERS
Uber connects you with thousands of riders in your city every day.
11:23
UBER
PICKUP LOCATION
EMPIRE STATE BUILDING
SET PICKUP LOCATION
Cancel
UBER
DRIVERS
Accept trips using the Uber app for drivers and get paid automatically.

FIG 2.1.1: Detecting current location

Geolocator feature helps us to detect our current location and take us to our destined location in an optimal way

Earlier, only military used to access gps to trace enemy's location. Merchant navy were the first one's to use gps for surveying underwater and to navigate hazard location. But now, every new car and even bikes come with installed gps navigation system which helps them to go from "point A to point B".

FIG 2.1.2: GPS in Merchant Navy

FIG 2.1.3 : GPS in Athletics

Many athletes use GPS for different purposes.

2.2 MOTIVATION

The main motivation behind creating geolocation application is to establish communication between our phone/smartphones and the satellites.

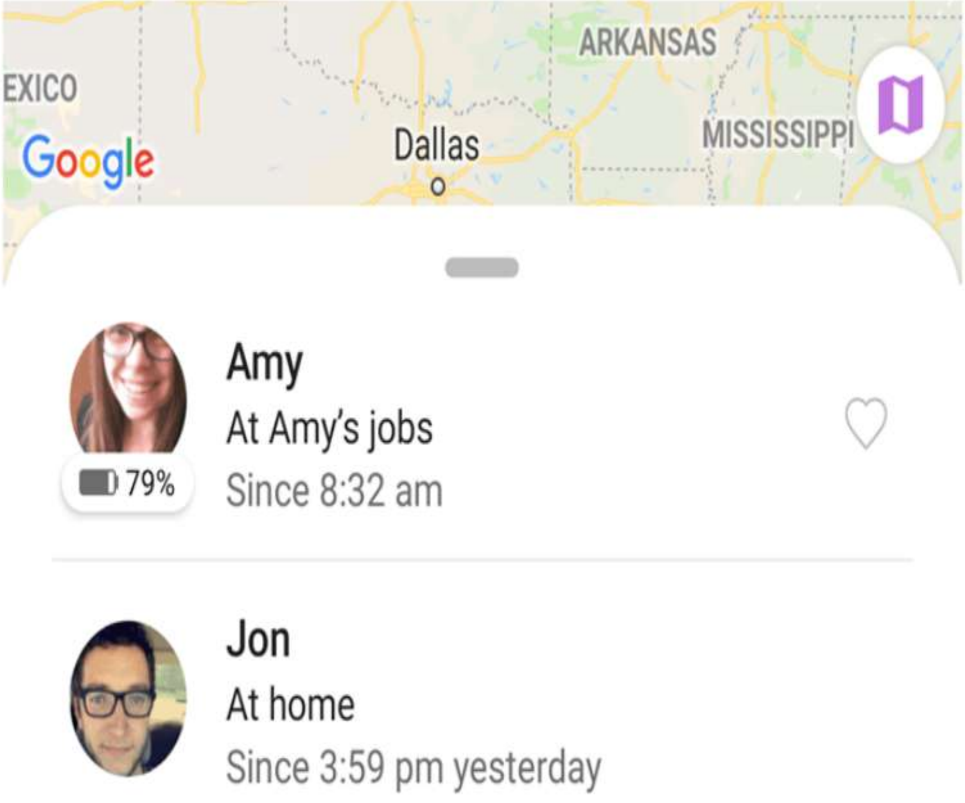
Geolocation uses technology of G.P.S which stands for Global Positioning System.

This motivation helps to create the best sync between the two i.e., GPS and the satellites.

This combination allows us to collect the data of the user or any dynamic thing/object.

This allows to make us dwell ourself into more scientific and advanced researches.

As we know google maps is a web mapping service offered by google. Geolocation helps one to trace their family member. You can trace your loved ones with just a click. So, you no longer need to worry if your loved one is safe or not because there are many apps that extends geolocator API and helps you to show the location of someone you care. So, this also motivates us to build something which contains this beautiful feature.



Amy
At Amy's jobs
Since 8:32 am
79%

Jon
At home
Since 3:59 pm yesterday

FIG 2.2.1: Collecting the data of the user

2.3 PROBLEM STATEMENT

The Study Project topic is Geo-location App. We will work on Android Studio to create the app. we are able to offer a framework for authentic wise combining records to a vicinity while presenting sturdy assurances in opposition to cloud garage providers that try to re-discover cloud information. Geo-location App which is connected to cloud storage facility, can be used to store current location of the user. In our app we will basically use Firebase which will provide us the facility to store the user's data. We are creating a Restaurant review app which will on location of the user will tell about the nearby restaurants, distance from the user location and the reviews about the restaurants from the app users. We are creating it as simple as for the users so that no problem occurs in signing and looking for the restaurants.

2.4 PROJECT PERSPECTIVE

Nowadays there is a huge follow up and trend for geolocation which has created new chances, advantages and start-up beliefs.

It provided road to the businesses that were never run using this as a technology.

However, it has given new life to thrilling levels of retailing strategies and it has changed the working flow of how to interact with the customer.

The achievability to attach geolocation in to an app has given space for new blend of consumer assistance for already alive goods and services.

Exploration has assumed new dimensions with application based on geolocation.

Apart from the usual or same map, apps based on location give more than one or optimal path from one place to the other.

They single out places which are of our interest and also seek answers to questions about our daily life like:

- "Which restaurant serves best momos?"
- "Where is the nearest petrol station?"

- “Can you show me directions to ahad’s premium shoe shop??”
- “From where I can buy fresh vegetables?”

The important thing about geolocation-based application is how they collect information about the location. There are various applications that are completely based on this tech and some uses or extends its feature in the application making that application more advance and robust.

Our app will be providing the user current location, details of the restaurant, location of the restaurants, address, opening and closing time, distance from the current location of user to the restaurant location and the reviews about the restaurants. In present world, mostly people follow up the reviews by the customers who before visited the place, so this app will help them to know about their restaurants better and would encourage to try new restaurants and cuisines.

2.5 SYSTEM REQUIREMENTS

1). **GPS (Global Positioning System):** Global Locating tool is the technology that is used by geolocation application. It is used to predict the location of a person or thing via GPS signals.

GPS does not transmit any user data.

FIG 2.5.1: GPS is used by Geolocation Application

2). **Cell Id**

A cell id is an exclusive number which analyse transceiver base station (BTS) or Area's location code (LAC).

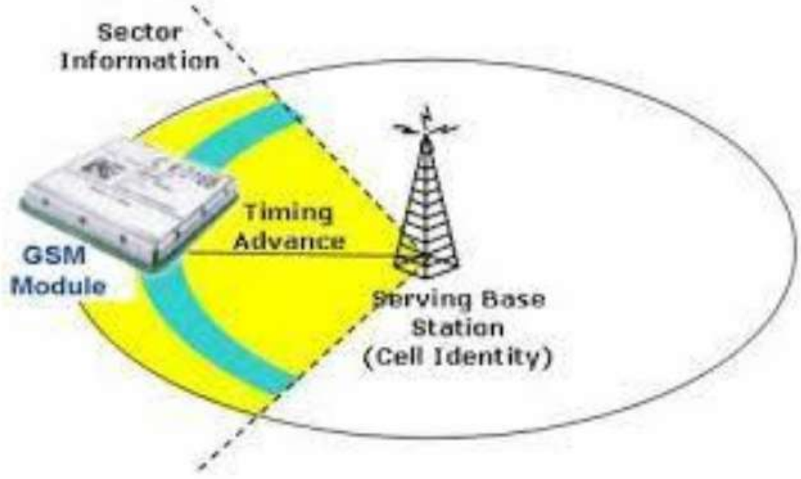
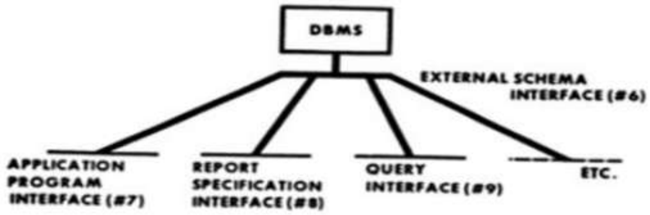


FIG 2.5.2: Cell Id

3). **APIs that essential for location's positioned application**:

API are interfaces of computer and these are used to deliver data from the hosting site to the client site as per their delivery of their criteria.

Application Programmer Interface


DBMS
EXTERNAL SCHEMA INTERFACE (#6)
APPLICATION PROGRAM INTERFACE (#7)
REPORT SPECIFICATION INTERFACE (#8)
QUERY INTERFACE (#9)
ETC.

An alternative which has several advantages is to make the API sufficiently rich to enable programs to be written in support of query, report generation, etc. (Figure 6).

FIG 2.5.3 : Application Programmer Interface

Enriched Application Programmer Interface

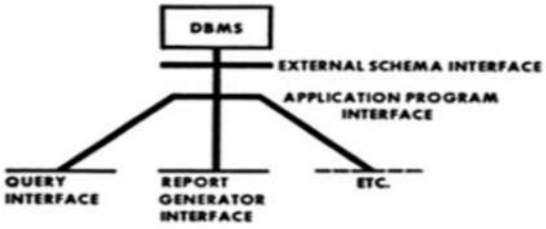
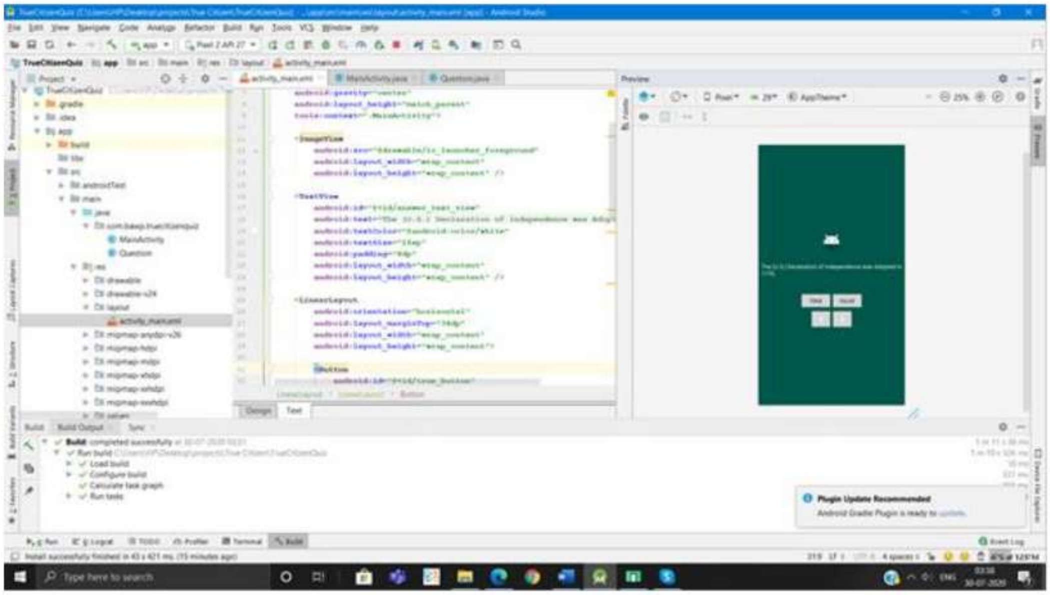

DBMS
EXTERNAL SCHEMA INTERFACE
APPLICATION PROGRAM INTERFACE
QUERY INTERFACE
REPORT GENERATOR INTERFACE
ETC.

FIG 2.5.4: Enriched Application Programmer Interface

4). **Android studio:**

Android Studio Is an IDE or integrated improvement surroundings that's advanced with the aid of google.It is primarily acting as a workshop which contains all the tools to create android applications.



```
    android:gravity="center"
    android:layout_height="match_parent"
    tools:context=".MainActivity">

    <ImageView
        android:src="@drawable/ic_launcher_foreground"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <TextView
        android:id="@+id/answer_text_view"
        android:text="The [illegible] Declaration of Independence was Adopt
        android:textColor="@android:color/white"
        android:textSize="[illegible]"
        android:padding="[illegible]"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

    <LinearLayout
        android:orientation="horizontal"
        android:layout_marginTop="[illegible]"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content">

        <Button
            android:id="@+id/true_button"
```

FIG 2.5.5: Android Studio

Android Studio Is an IDE or integrated improvement surroundings that's advanced with the aid of google.It is primarily acting as a workshop which contains all the tools to create android applications.

5). **Android Emulator:**

It is an emulating device which emulate other devices on PC when there is no available mobile device to be connected.

FIG 2.5.6: Mobile View

6). **Firebase:**

It is a Backend-as-a-Service (BaaS). It enables user with different widgets and help the programmers which provide support to make good application, develop their customer end and gain profitable trust. It has been listed as a No S Q L database management module, which collects records in JSON format.

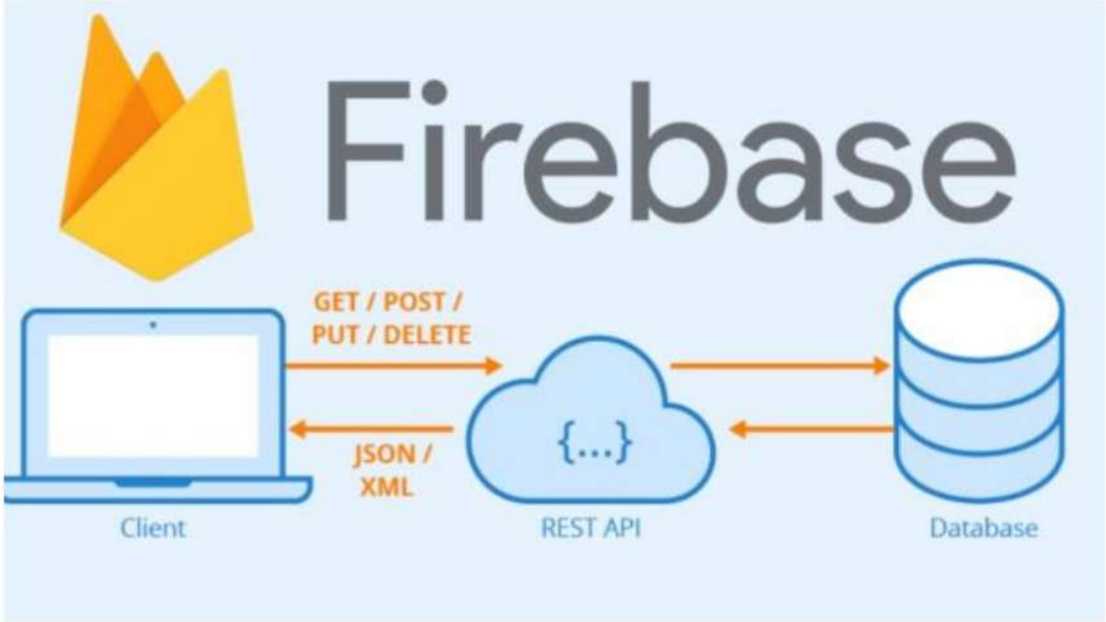


FIG 2.5.7: Firebase

CHAPTER 3

TOOLS AND TECHNOLOGIES

In preparing this project there are some basic requirements and a procedure to complete the Android project which provided many features that helped us to construct material layout application:

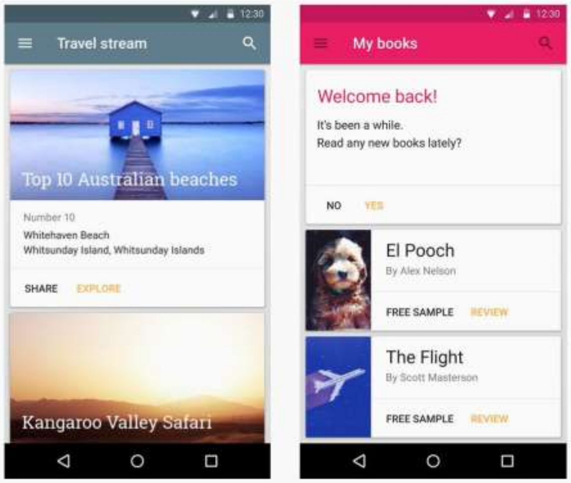
- We require a material layout app subject to style all our UserInterface tools.
- We require widgets for complicated perspectives like lists and card views.
- For user defined shadows and animations, we required new APIs.

We will apply a materialistic-layout theme on our application so that we can take benefit of the materialistic charactersitics like tailoring and designing for qualitu user interface tools.

We can also apply two types of themes for our application, black materialistic layout and light materialistic layout. We will also provide a Floating Action Button (FAB) to promote our user interface main action. There will also be an app bar, that show our app name, navigation, search and other actions.

We can use many materialistic parts for the application theme and traversing like fall down toolbars, tabs, a base searching bar, and much more.

Working for this project, card-based layout can also be applied to the app which help us to present significant pieces of data inside cardview that supply a materialistic outcome. When card view is applied, default elevation is created.



Travel stream

Top 10 Australian beaches

Number 10
Whitehaven Beach
Whitsunday Island, Whitsunday Islands

Kangaroo Valley Safari

My books

Welcome back!

It's been a while.
Read any new books lately?

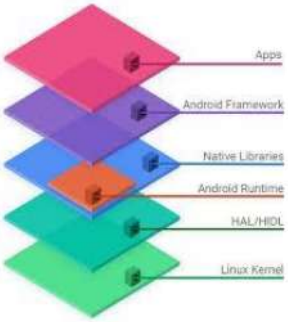
El Pooch
By Alex Nelson

The Flight
By Scott Masterson

FIG 3.1: CARD VIEW

We start by creating user interface using XML. As we move further, we can add map API's to list our map on the user screen. Different methods and activities will be created to make the app as attractive as possible. Then the different activities will be linked together so that one activity directly or indirectly starts the new activity. Each activity will have different functions. There will also be different kind of buttons that have their own functions.

We can get recommendations of almost every available category. There will be some reviews available. User will be able to view recent photos on the go and we can even share our experience with our friends. User can even search for the places he or she may like to visit. We can even reach out to the place with all connectivity options and they can even see the recommendations on the google map.



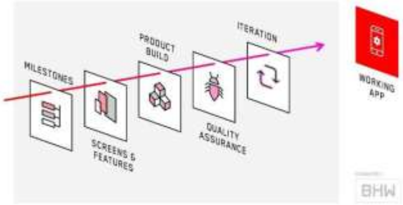


FIG 3.2 : Android Framework

FIG 3.3: Flow Chart to Build an Android App

The basic requirements for our app will be an Android device and this device should have at least 1GB RAM for our app to work perfectly. For android version we can use KitKat version. There will be google maps and lastly internet connection will also be required.

So, we first have to link our app in Android Studio to one of the cloud storage systems like Firebase. The purpose of our work is to build a data service that will provide benefits to users. It will help the users to retrieve the data of the places where they have checked in or last visited. Whenever we update the app user will be notified.

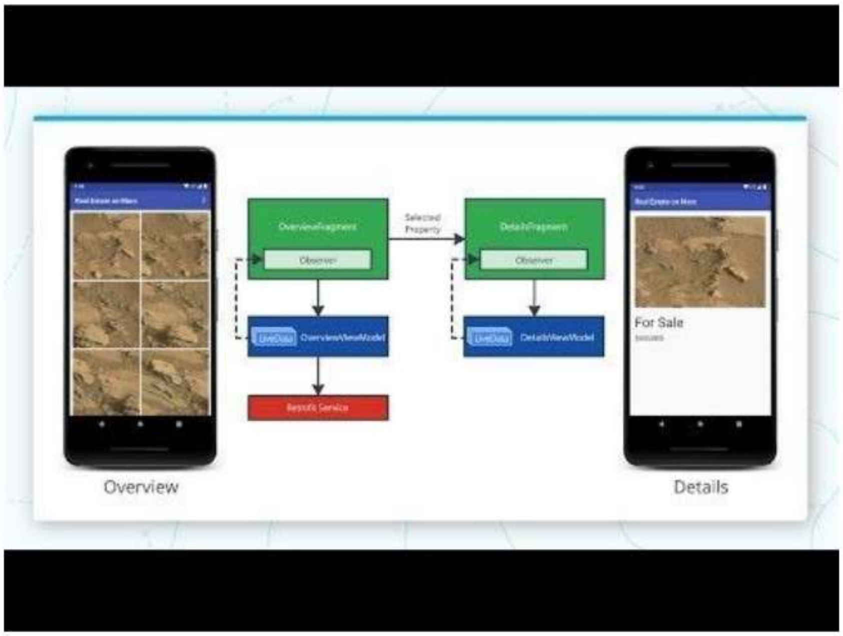
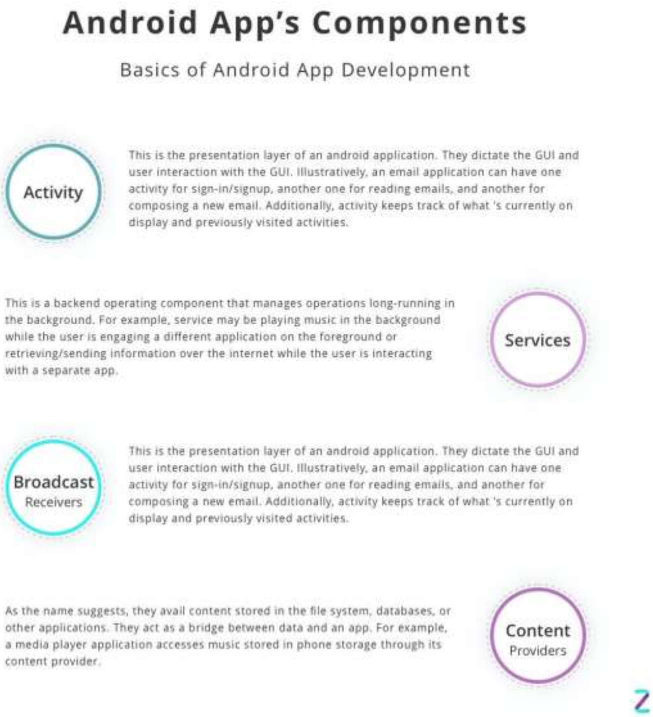


FIG 3.4: Different Fragments

We will basically work on two languages JAVA and XML. Java is a preferred-motive programming language. iI is classes dependent and object orientation dependent. The Java code can execute on all systems with a view to assist Java. there may be no need to compile code again some other language is XML which stands for Extensible Marking Up Language. It defines a hard and fast of guidelines for encoding files in a layout that may be examine with the aid of each human and machines

All user interface is done using XML. This language is responsible to make the interface as attractive as possible. Mostly all work on backend is done through Java but what the user sees on his or her screen is XML which is the user interface. Android apps have various building blocks known as app components.

We will then put some new Android apps such as Room Persistence. Room persistence is the library that helps us to make a temporary storage data of our application record on a device that is executing our application. All this work will be done on Android Studio which is a platform where we write all kinds of code using different languages and combine them.



Android App's Components

Basics of Android App Development

This is the presentation layer of an android application. They dictate the GUI and user interaction with the GUI. Illustratively, an email application can have one activity for sign-in/signup, another one for reading emails, and another for composing a new email. Additionally, activity keeps track of what 's currently on display and previously visited activities.

This is a backend operating component that manages operations long-running in the background. For example, service may be playing music in the background while the user is engaging a different application on the foreground or retrieving/sending information over the internet while the user is interacting with a separate app.

This is the presentation layer of an android application. They dictate the GUI and user interaction with the GUI. Illustratively, an email application can have one activity for sign-in/signup, another one for reading emails, and another for composing a new email. Additionally, activity keeps track of what 's currently on display and previously visited activities.

As the name suggests, they avail content stored in the file system, databases, or other applications. They act as a bridge between data and an app. For example, a media player application accesses music stored in phone storage through its content provider.

FIG 3.5: Android App Components

We will also use Android Emulator which is just like a fully fledged mobile on our PC or laptop screen to test our apps.

We can register all our activity in the manifest. Every activity that we will create or every window or view that we create that users can actually interact with, we must register that activity inside of our Android manifest. Android contains ART which stands for Android Runtime Libraries which connects the runtime of the entire Android system.

Some tools or some common requirements for our app are 8GB RAM on our laptops or personal computers, Android Studio which is the official IDE for Google's Android OS, Emulator for running and testing our apps on our PC's, MAPS API by google, and Places API by Google. We also have to add Intents for Gallery that will include some features that will provide what exact class we want to run first and next.

Figma:

It is a internet-dependent graphical editing and user interface (UI) design application. All types of graphical editing work from framing websites, to design mobile application interfaces, initial stage designs. In this, teams can create design among themselves and helps the entire design team to create, test, and ship better application design. The Figma Mirror has various companion apps for android and ios which enables viewing figma initial design on mobile devices. Figma can works on every operating system that has a web browser. It is the only prototype software that does this. Figma uses many communicating medium like slack. It also enables permissions-dependent sharing of any records, page, or file. It shares live attached code part to paste in iFrame in third-party tools.

FIG 3.6: Figma

CHAPTER 4

IMPLEMENTATION MODULES AND SCREEN SHOTS

4.1. HOME SCREEN

This screen will show your current location on map. We implemented map fragment by using maps api provided by google. In order to implement map in the fragment, first, we had to copy the unique key when we signed on console.cloud.google.com/google/maps in our project. Then, we wrote the code which allows the application to fetch user's current location. The green marker, you can see in the screenshot down below points to the current location. Also, we used View Flipper which shows you some food pictures on the top of the map. It has a flip interval of 2.5 seconds i.e.; Pictures will keep changing every 2.5 seconds.

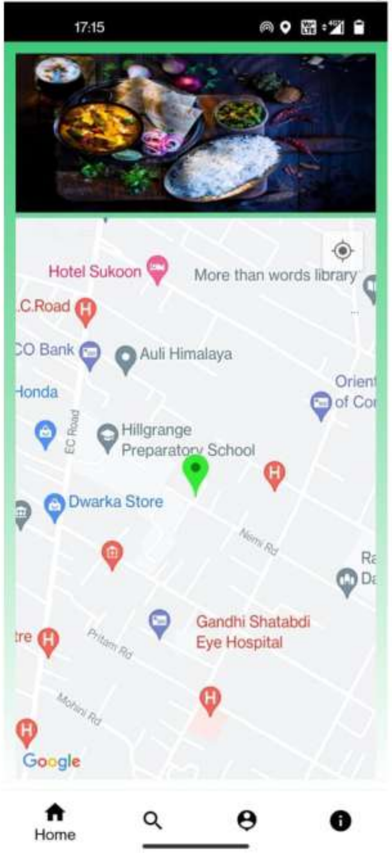
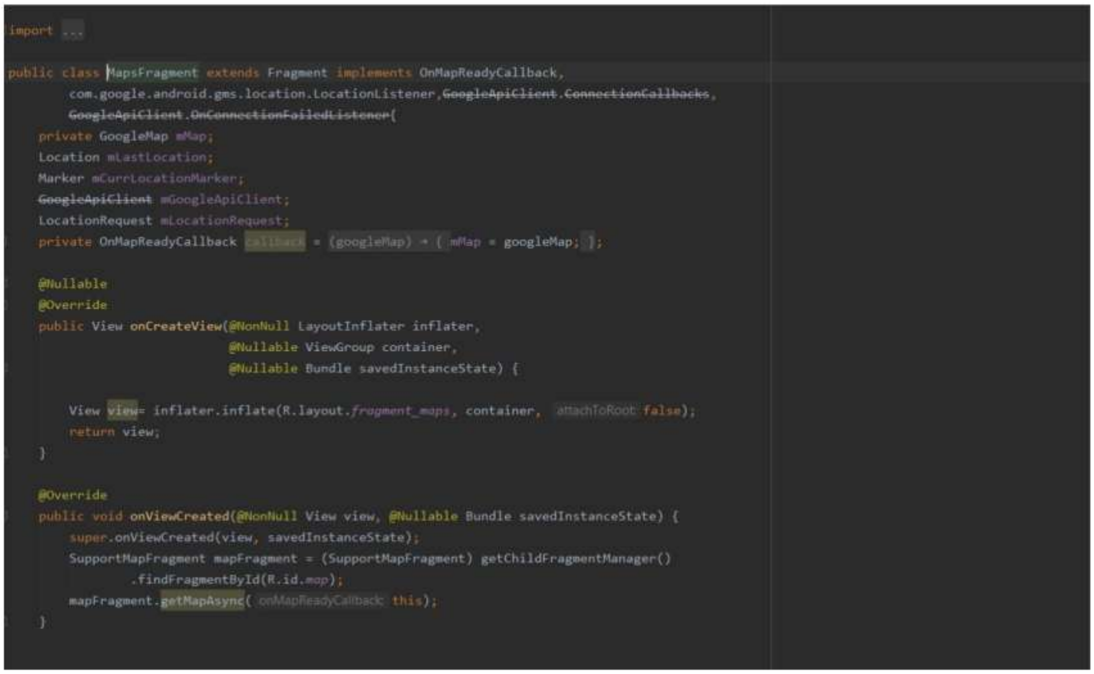


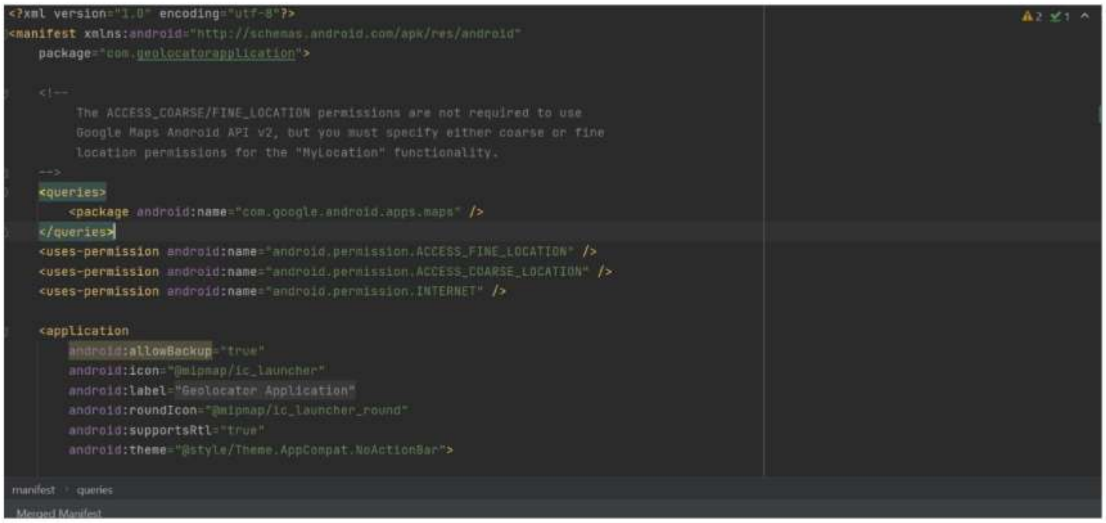
FIG 4.1.1: Home Screen



```
import ...

public class MapsFragment extends Fragment implements OnMapReadyCallback,
        com.google.android.gms.location.LocationListener,~~GoogleApiClient.ConnectionCallbacks~~,
        ~~GoogleApiClient.OnConnectionFailedListener~~{
    private GoogleMap mMap;
    Location mLastLocation;
    Marker mCurrLocationMarker;
    ~~GoogleApiClient~~ mGoogleApiClient;
    LocationRequest mLocationRequest;
    private OnMapReadyCallback callback = (googleMap) → { mMap = googleMap; };

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater,
                             @Nullable ViewGroup container,
                             @Nullable Bundle savedInstanceState) {

        View view= inflater.inflate(R.layout.fragment_maps, container,  attachToRoot: false);
        return view;
    }

    @Override
    public void onViewCreated(@NonNull View view, @Nullable Bundle savedInstanceState) {
        super.onViewCreated(view, savedInstanceState);
        SupportMapFragment mapFragment = (SupportMapFragment) getChildFragmentManager()
                .findFragmentById(R.id.map);
        mapFragment.getMapAsync( onMapReadyCallback: this);
    }
```

FIG 4.1.2: Home Screen Code

We have also enabled google map navigation in our app. When user searches for a particular restaurant of his or her choice then he is provided with the location of that restaurant with a marker on a map. Now when user hits on the marker on the map he is provided with the latitude and longitude of that particular restaurant. Also then he will also have the advantage of using map navigation which directs him towards the restaurant from his current location. This helps the user to identify the exact location of the restaurant.

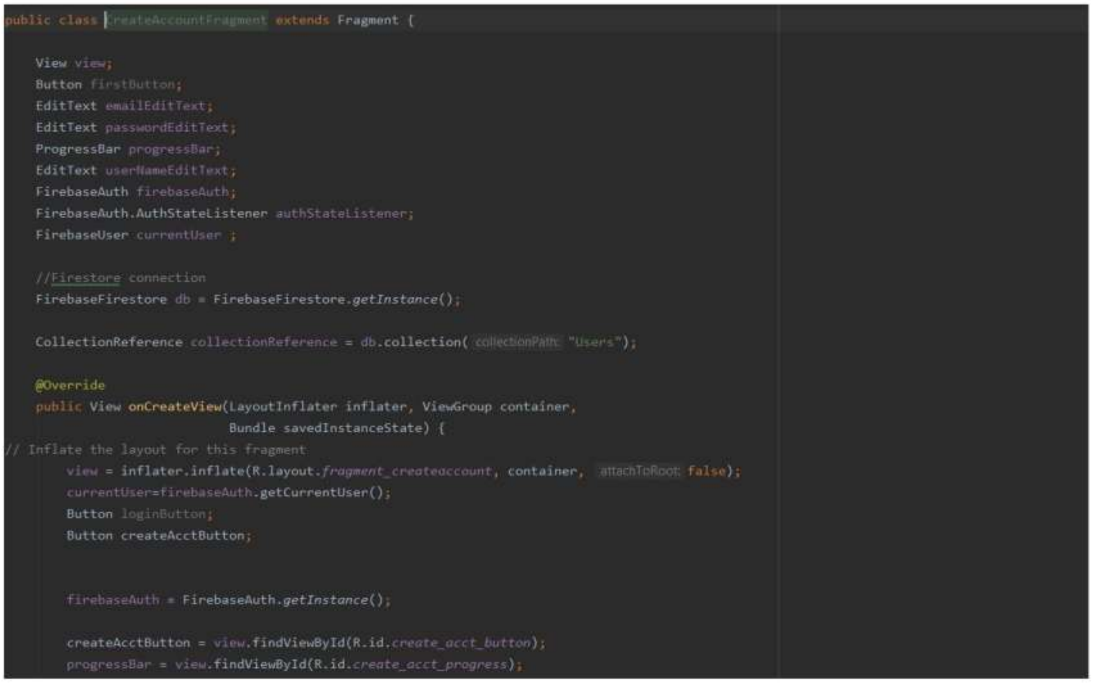


```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.geolocatorapplication">

    <!--
        The ACCESS_COARSE/FINE_LOCATION permissions are not required to use
        Google Maps Android API v2, but you must specify either coarse or fine
        location permissions for the "MyLocation" functionality.
    -->
    <queries>
        <package android:name="com.google.android.apps.maps" />
    </queries>
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
    <uses-permission android:name="android.permission.INTERNET" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="Geolocator Application"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/Theme.AppCompat.NoActionBar">
```

FIG 4.1.3: Home Screen Code

4.2. USER ACCOUNT'S SCREEN

This screen deals with creating an account for a user. This screen will be visible to the user after the Splash Screen. As soon as this screen loads the user will be provided with two options to choose from. The first option will be "sign-up" and the second one will be "log-in". If we talk about the first scenario, where the user is using the app for the very first time, he/she will then have to create an account by clicking the "sign-up" button. This button will in turn provide the user with a small details section to be filled. This section will ask the user to fill out his personal details in username, email and password fields. After filling out the details the user will then need to click on "Done" button. As soon as user hits on "Done" the data of the user will get fetched from here and will be stored in Firestore. We will create collection and documents in Firestore for storing and fetching these details. Finally, the main screen of our app will load. Now if we talk about the second scenario, where the user has already made an account then he/she will then have to just log-in by clicking the "log-in" button. This button will then provide the user with a details section which will have username and password fields only. As soon as the user has filled these details and hit on "Done" button, his/her data will be fetched from Firestore and checked whether the inputs entered by the user in the given fields match with his data which is already stored in Firestore in his first attempt. If the condition is true then the main screen of our app will load, otherwise “incorrect username” or “incorrect password” options will be popped and the user will be asked again to enter these respective fields. Since our app is already connected to the Firebase which has Firestore so a database will be created in the Firestore which will have collections, and these collections will have documents from where we will fetch up the data. This will be used both, for storing data (at the time of creating account) and fetching data (at the time of logging in).



```
public class CreateAccountFragment extends Fragment {

    View view;
    Button firstButton;
    EditText emailEditText;
    EditText passwordEditText;
    ProgressBar progressBar;
    EditText userNameEditText;
    FirebaseAuth firebaseAuth;
    FirebaseAuth.AuthStateListener authStateListener;
    FirebaseUser currentUser ;

    //Firestore connection
    FirebaseFirestore db = FirebaseFirestore.getInstance();

    CollectionReference collectionReference = db.collection( collectionPath: "Users");

    @Override
    public View onCreateView(LayoutInflater inflater, ViewGroup container,
                             Bundle savedInstanceState) {
// Inflate the layout for this fragment
        view = inflater.inflate(R.layout.fragment_createaccount, container,  attachToRoot: false);
        currentUser=firebaseAuth.getCurrentUser();
        Button loginButton;
        Button createAcctButton;


        firebaseAuth = FirebaseAuth.getInstance();

        createAcctButton = view.findViewById(R.id.create_acct_button);
        progressBar = view.findViewById(R.id.create_acct_progress);
```

FIG 4.2.1: User Account Code

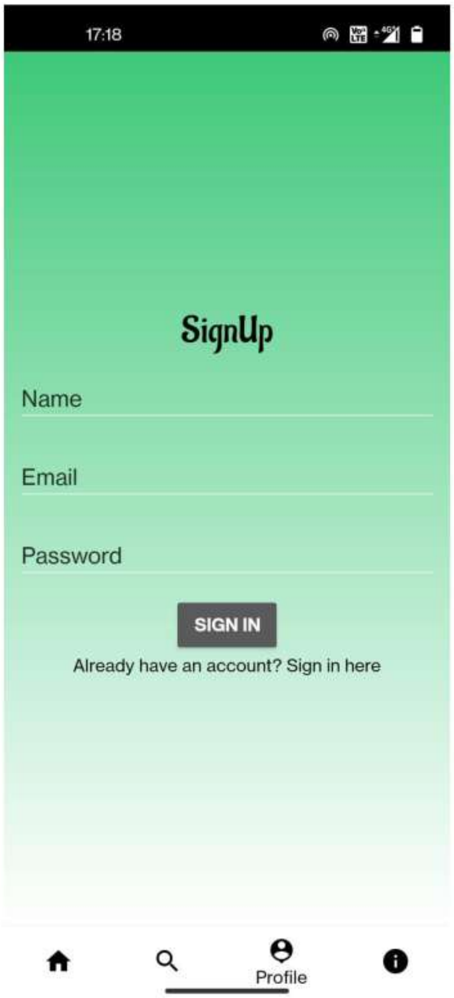
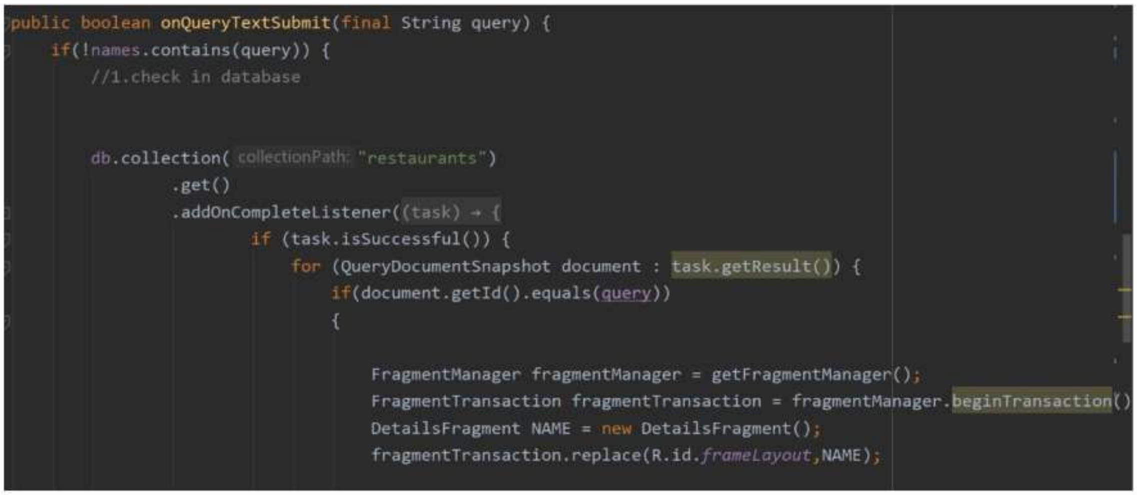


FIG 4.2.2: Sign Up Screen

4.3. SEARCH SCREEN

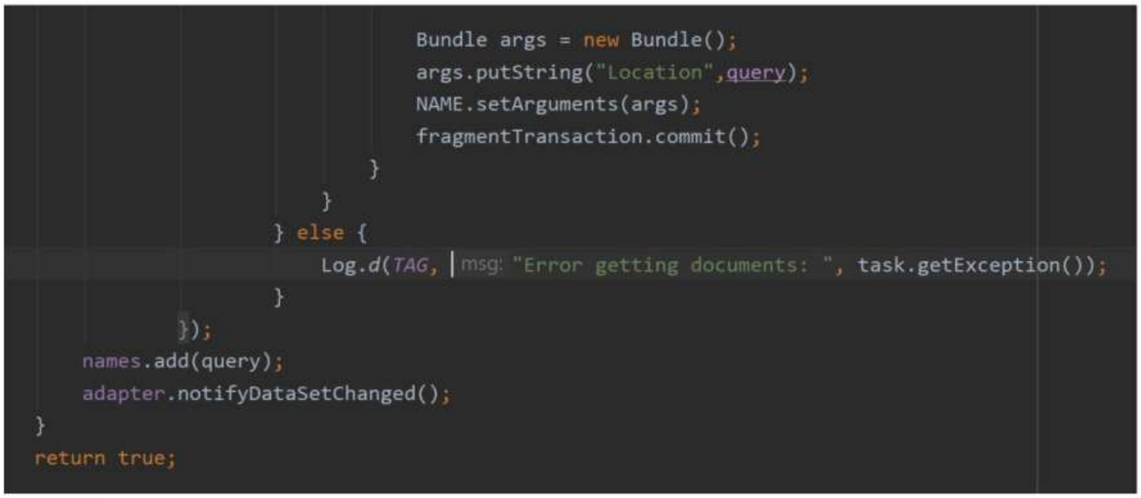
This fragment allows you to search the particular restaurant. We used searchView to search the restaurant. If the restaurant that user searched exists in database then the user will be redirected to restaurant details page. Otherwise, a toast message would appear that the restaurant user searched does not exist in database. We used firebase firestore database to store the restaurant details. As the user clicks on the search button, the searched restaurant will also get stored in

ArrayList. To save the list, we used shared preferences which allows you to save the list even after the app gets terminated. We used recyclerView in order to show the stored restaurants in a grid format.



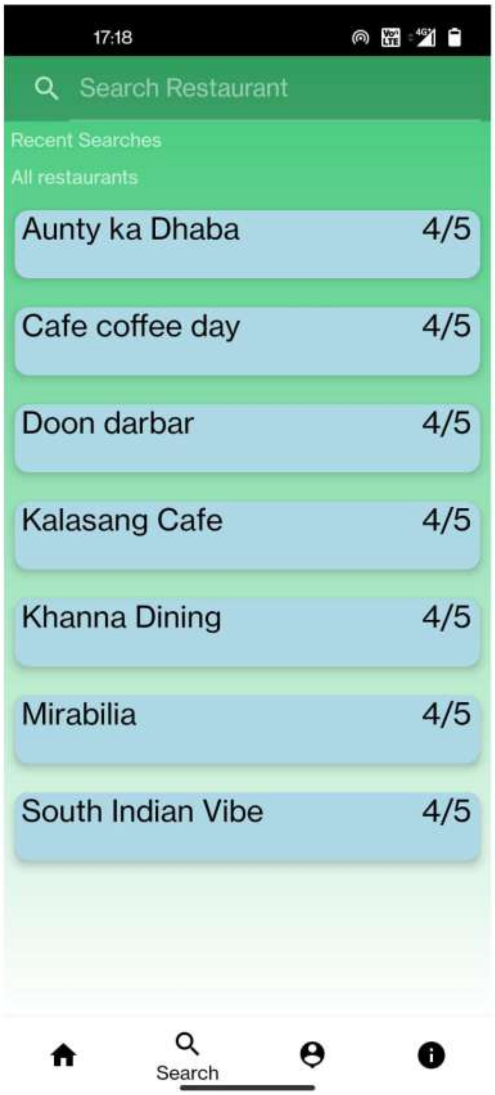
```
public boolean onQueryTextSubmit(final String query) {
    if(!names.contains(query)) {
        //1.check in database


        db.collection( collectionPath: "restaurants")
                .get()
                .addOnCompleteListener((task) -> {
                    if (task.isSuccessful()) {
                        for (QueryDocumentSnapshot document : task.getResult()) {
                            if(document.getId().equals(query))
                            {

                                FragmentManager fragmentManager = getFragmentManager();
                                FragmentTransaction fragmentTransaction = fragmentManager.beginTransaction()
                                DetailsFragment NAME = new DetailsFragment();
                                fragmentTransaction.replace(R.id.frameLayout,NAME);
```

FIG 4.3.1: Search Code



```
                                Bundle args = new Bundle();
                                args.putString("Location",query);
                                NAME.setArguments(args);
                                fragmentTransaction.commit();
                            }
                        }
                    } else {
                        Log.d(TAG, msg: "Error getting documents: ", task.getException());
                    }
                });
        names.add(query);
        adapter.notifyDataSetChanged();
    }
    return true;
```

FIG 4.3.2: Search Code



17:18

Search Restaurant

Recent Searches

All restaurants

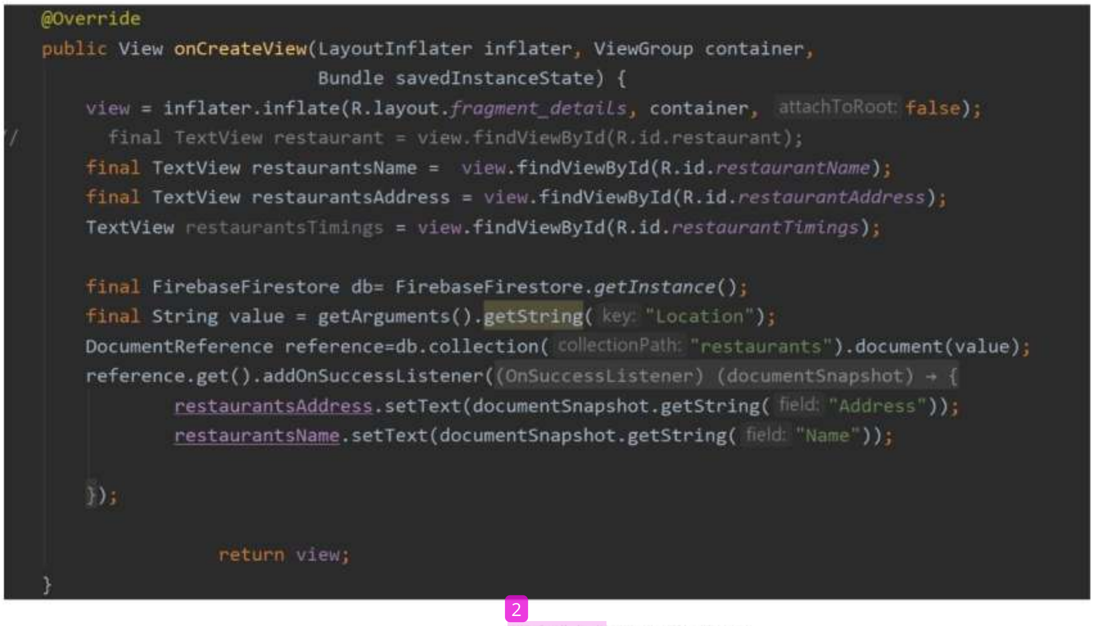
Aunty ka Dhaba	4/5
Cafe coffee day	4/5
Doon darbar	4/5
Kalasang Cafe	4/5
Khanna Dining	4/5
Mirabilia	4/5
South Indian Vibe	4/5

FIG 4.3.3: Search Screen

4.4. DETAILS SCREEN

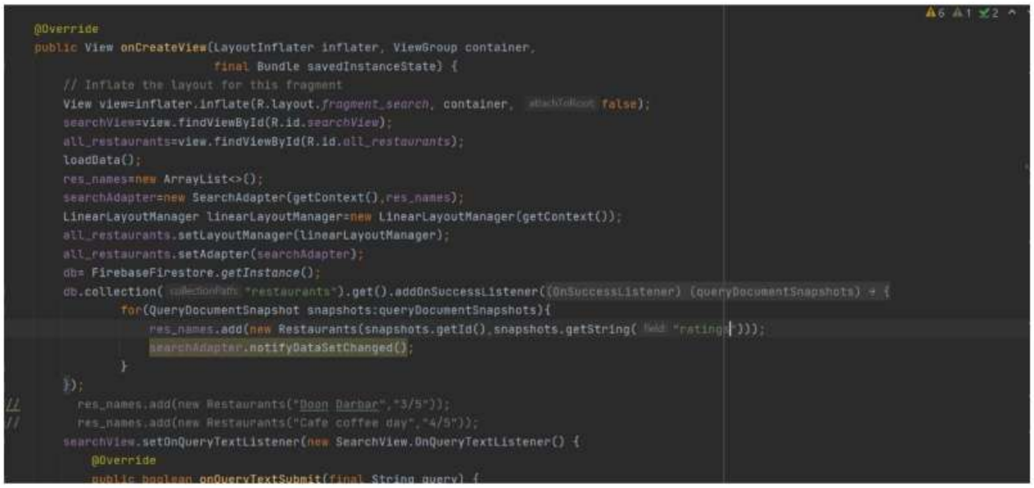
In details fragment we will fetch the data from the search fragment and display in the text views provided in this fragment. We have made use of Firebase Firestore database to store the details of the local restaurants near DIT University which are generally not available in the Google Maps. We have made a Firestore database that contains fields such as Restaurant's Name, Address, Timings, Latitude and Longitude. In Details Fragment we have fetched the restaurants details which is searched in Search Fragment by the user and displayed the related data of that restaurants in this fragment i.e.; Details Fragment.

In this we have fetched the ratings dynamically from firebase. Each restaurant has different dynamic ratings which will be shown for each restaurant alongside its name. These ratings are useful for the customers to understand the credibility of the restaurants. We have to fetch the attributes from document using document snapshot and get the string name as ratings from firebase. Dynamic ratings are stored as a document attribute in firebase.



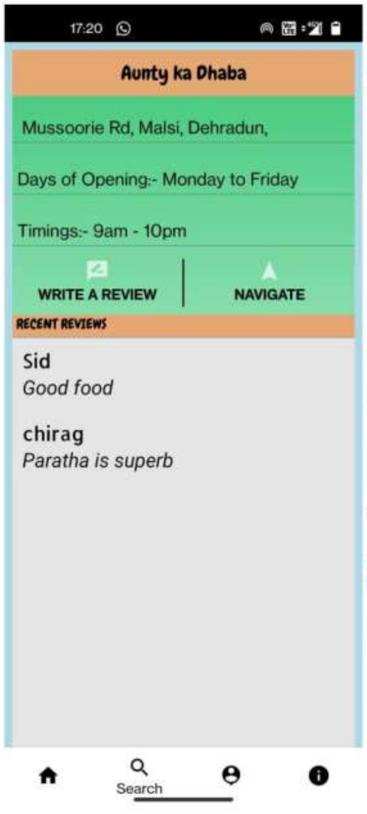
```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                         Bundle savedInstanceState) {
    view = inflater.inflate(R.layout.fragment_details, container,  attachToRoot: false);
/     final TextView restaurant = view.findViewById(R.id.restaurant);
    final TextView restaurantsName =  view.findViewById(R.id.restaurantName);
    final TextView restaurantsAddress = view.findViewById(R.id.restaurantAddress);
    TextView restaurantsTimings = view.findViewById(R.id.restaurantTimings);

    final FirebaseFirestore db= FirebaseFirestore.getInstance();
    final String value = getArguments().getString( key: "Location");
    DocumentReference reference=db.collection( collectionPath: "restaurants").document(value);
    reference.get().addOnSuccessListener((OnSuccessListener) (documentSnapshot) → {
            restaurantsAddress.setText(documentSnapshot.getString( field: "Address"));
            restaurantsName.setText(documentSnapshot.getString( field: "Name"));

    });

                return view;
}
```

FIG 4.4.1: Details Code



```
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                         final Bundle savedInstanceState) {
    // Inflate the layout for this fragment
    View view=inflater.inflate(R.layout.fragment_search, container,  attachToRoot: false);
    searchView=view.findViewById(R.id.searchView);
    all_restaurants=view.findViewById(R.id.all_restaurants);
    loadData();
    res_names=new ArrayList<>();
    searchAdapter=new SearchAdapter(getContext(),res_names);
    LinearLayoutManager linearLayoutManager=new LinearLayoutManager(getContext());
    all_restaurants.setLayoutManager(linearLayoutManager);
    all_restaurants.setAdapter(searchAdapter);
    db= FirebaseFirestore.getInstance();
    db.collection( collectionPath: "restaurants").get().addOnSuccessListener((OnSuccessListener) (queryDocumentSnapshots) → {
            for(QueryDocumentSnapshot snapshots:queryDocumentSnapshots){
                res_names.add(new Restaurants(snapshots.getId(),snapshots.getString( field: "rating")));
                searchAdapter.notifyDataSetChanged();
            }
    });
//    res_names.add(new Restaurants("Doon Darbar","3/5"));
//    res_names.add(new Restaurants("Cafe coffee day","4/5"));
    searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {
        @Override
        public boolean onQueryTextSubmit(final String query) {
```

FIG 4.4.2: Details Code



17:20

Aunty ka Dhaba

Mussoorie Rd, Malsi, Dehradun,

Days of Opening:- Monday to Friday

Timings:- 9am - 10pm

RECENT REVIEWS

Sid
Good food

chirag
Paratha is superb

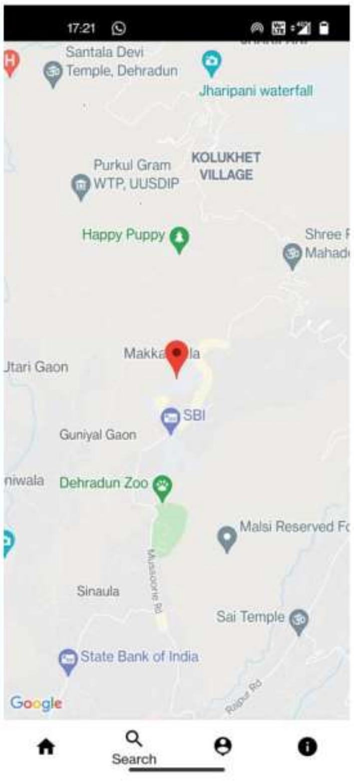


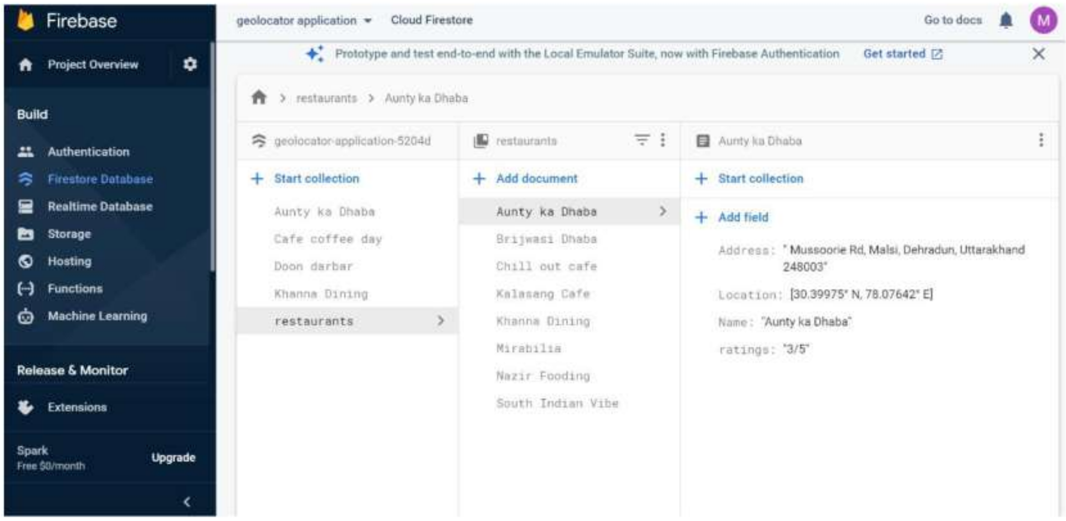
FIG 4.4.3:Review Screen

FIG 4.4.4: Restaurant Location Screen

4.5. FIREBASE AUTHENTICATION

Firebase Authentication has created a functionality to identify a particular user.When a user's identity is known this calls for saving the data in the cloud more securely and this experience is share with all the other users.Firebase Authentication provide services which work at backend , it has UI – libraries which performs authentication of the user.Firebase Authentication provides authentication to various sites like google, facebook, etc.

In Firebase we have created a collection of restaurants which are local to DIT university. These restaurants are not found on Google Maps. All these restaurants have details such as address, name, location along with latitude and longitude and ratings. The ratings are fetched dynamically from Firebase. These collection of restaurants provide usefulness to the customers who are not aware of these restaurants and also helps the sellers to increase their business. Firebase stores these details and these details are updated dynamically.



restaurants > Aunty ka Dhaba

geolocator-application-5204d
- Aunty ka Dhaba
- Cafe coffee day
- Doon darbar
- Khanna Dining
- restaurants

restaurants
- Aunty ka Dhaba
- Brijwasi Dhaba
- Chill out cafe
- Kalasang Cafe
- Khanna Dining
- Mirabilia
- Nazir Fooding
- South Indian Vibe

Aunty ka Dhaba

Address: " Mussoorie Rd, Malsi, Dehradun, Uttarakhand 248003"

Location: [30.39975° N, 78.07642° E]

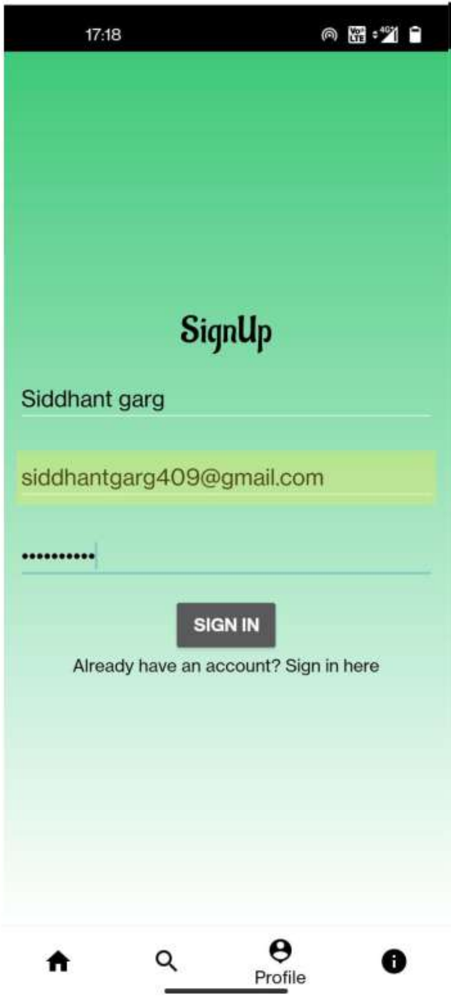
Name: "Aunty ka Dhaba"

ratings: "3/5"

FIG 4.5.1: Firebase

4.6. LOGIN SCREEN

In this login screen user will be asked to sign in to their repective accounts which they have created previously when they registered using the create account fragment.When the user creates the account, their credentials are stored in the firebase firestore authentication database which stores the user information their email Id’s etc.When a user enter his/her credentials for the login screen , its authentication is done at backend at the cloud storage.



SignUp

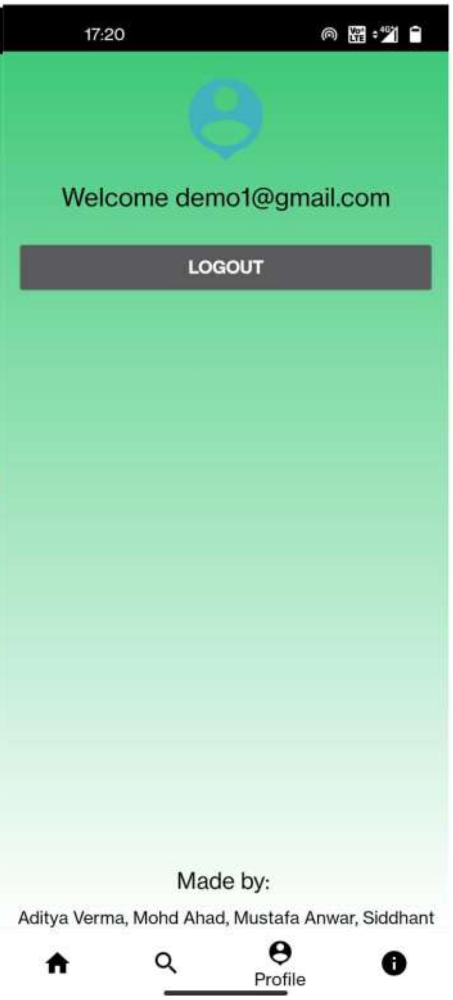
Siddhant garg

siddhantgarg409@gmail.com

••••••••••

Already have an account? Sign in here

FIG 4.6.1: Login Screen



Welcome demo1@gmail.com

Made by:

Aditya Verma, Mohd Ahad, Mustafa Anwar, Siddhant

FIG 4.6.2: Signed Screen

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

Android is a complete, open and unfastened cellular device platform, with its influential characteristic and better host encounter revel in fast developed into the most popular cell operation system in the international market

This report gives the overview of the different challenges and issues faced in creating the android application. The problems faced by many students during college time mainly focused is "Food". We try to overcome this situation by helping them to know good and affordable food shops near college.Map projectioning and geometrical node in plotting are important for place naming in the world, that's moved to flat map. The geometrica machine permits us to understand material's location and position on earth via Meridian strip cut up the earth into elements assemble on angle based path or meridian referred to as Geographical Coordinates.. Map helps too much for the navigation purpose and nowadays featuring location of any thing near us gets the world into new position.

FUTURE SCOPE

The project has a very extensive scope in future. The application can be executed on intranet in future and can be helpful for the students all over the places and for the stalls and shops near college. In future we will try to implement direction functionality onto our main location map and try to improve the user interface of the application.

Additionally, we will try to reduce the Apk size by utilizing the code more efficiently and try to expand the categories of from restaurant to local shopping malls, general stores, barber shops and all needful things needed by students in daily life.

BIBLIOGRAPHY

1. **http://www.standardista.com/html5/introduction-to-geolocation/#:~:text=Especially%20useful%20in%20social%20networking,the%20location%20of%20the%20user.**
2. **https://en.wikipedia.org/wiki/Android_(operating_system)**
3. **https://developers.google.com/maps/documentation/javascript/geolocation**
4. **https://www.spaceotechnologies.com/geolocation-app-ideas-integrate-location-based-services/**
5. **https://www.androidauthority.com/history-android-os-name-789433/**
6. **https://en.wikipedia.org/wiki/Google_Maps#:~:text=Google%20Maps%20is%20a%20web,in%20beta)%20and%20public%20transportation.**
7. **https://medium.com/firebase-developers/what-is-firebase-the-complete-story-abridged-bcc730c5f2c0**
8. **https://www.toptal.com/designers/ui/figma-design-tool**
9. **https://www.gistda.or.th/main/en/node/1114**
10. **https://www.vogella.com/tutorials/AndroidLocationAPI/article.html**
11. **https://www.projecttopics.info/android/android-application-final-year-project-report.php**
12. **https://www.docsity.com/en/android-app-report-format/5099203/**

G40_R

ORIGINALITY REPORT

1%	1%	0%	0%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.toptal.com Internet Source	<1%
2	"Advanced Reservoir Characterization and Evaluation of CO2 Gravity Drainage in the Naturally Fractured Spraberry Trend Area, Class III", 'Office of Scientific and Technical Information (OSTI)' Internet Source	<1%
3	igreigrice247.com Internet Source	<1%
4	www.hindawi.com Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	On		