1. INTRODUCTION

Mobile application development is the set of processes and procedures involved in writing software for small, wireless computing devices such as smartphones or tablets.

Mobile application development is similar to Web application development and has its roots in more traditional software development. One critical difference, however, is that mobile applications (apps) are often written specifically to take advantage of the unique features a particular mobile device offers. For instance, a gaming app might be written to take advantage of the iPhone's accelerometer.

One way to ensure that applications show optimum performance on a given device is to develop the application (app) natively on that device. This means that at a very low level, the code is written specifically for the processor in a particular device. When an app needs to run on multiple operating systems, however, there is little -- if any -- code that can be reused from the initial development. The application must essentially be rewritten for each specific device.

In the future, it's expected that a majority of mobile application development efforts will focus on creating browser-based applications that are device-agnostic. Browser-based applications are simply websites that are built for mobile browsers. Such sites are built to load quickly over a cellular network and have finger-friendly navigation.

1.1 What is Mobile Application Development?

A mobile application, most commonly referred to as an app, is a type of application software designed to run on a mobile device, such as a smartphone or tablet computer. Mobile applications frequently serve to provide users with similar services to those accessed on PCs. Apps are generally small, individual software units with limited function. This use of software has been popularized by Apple Inc. and its App Store, which sells thousands of applications for the iPhone, iPad and iPod Touch.

1.2 JAVA

Java is a general-purpose computer-programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture.

Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform.

The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them. The original and reference implementation Java compilers, virtual machines, and class libraries were originally released by Sun under proprietary licenses.

The Java language is a key pillar in Android, an open source mobile operating system. Although Android, built on the Linux kernel, is written largely in C, the Android SDK uses the Java language as the basis for Android applications. The bytecode language supported by the Android SDK is incompatible with Java bytecode and runs on its own virtual machine, optimized for low-memory devices such as smartphones and tablet computers. Depending on the Android version, the bytecode is either interpreted by the Dalvik virtual machine or compiled into native code by the Android Runtime.

1.3 XML

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The W3C's XML 1.0 Specification and several other related specifications—all of them free open standards—define XML.

The design goals of XML emphasize simplicity, generality, and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, the language is widely used for the representation of arbitrary data structures such as those used in web services.

Several schema systems exist to aid in the definition of XML-based languages, while programmers have developed many application programming interfaces (APIs) to aid the processing of XML data.

1.4 Android Studio

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as primary IDE for native Android application development.

Android Studio supports all the same programming languages of IntelliJ, and PyCharm e.g. Python, and Kotlin; and Android Studio 3.0 supports "Java 7 language features and a subset of Java 8 language features that vary by platform version." External projects backport some Java 9 features.

It has built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine Android Virtual Device (Emulator) to run and debug apps in the Android studio.

It provides Gradle-based build support . Android-specific refactoring and quick fixes. Lint tools to catch performance, usability, version compatibility and other problems . ProGuard integration and app-signing capabilities . Template-based wizards to create common Android designs and components . A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations. Support for building Android Wear apps.

1.5 OBJECTIVE

The main objective of this mini project is to track the location of lost mobile without depending any external agencies by installing this application in their mobile.

2. EXISTING SYSTEM

The existing system works only if it has internet connection to locate lost mobile using some tracking application or using IMEI number when ever the lost mobile is used to make a call. In latter case it requires government support and may take a very long time to recover the mobile. The probability of recovering the lost mobile in both cases is very low.

2.1 METHODOLOGY

We intend to develop an application which tracks the mobiles location and user can login to the app using his/her mobile phone id. There are two apps working to complete the process.

One is **Tracer** and other one is **Runner**. if the mobile is lost we want to send signal to the lost mobile by using Tracer. When the Runner receives the signal from the Tracer it will send the latitude and longitude of it location for every five second or for every 5 meters movement.

Using tracer app we can get the location of the mobile from server by using the mobile id.

2.2 LIMITATIONS

- Only registered mobile can find the lost mobile.
- We can track the mobile only when it is power on.
- GPS must be turned on both devices to track.

3. DESIGN METHODOLOGY

The proposed system follows a digital level architecture, which involves structuring the data, and manipulation of the geo location with respective to the each android that has the mobile application installed in prior.

3.1 Problem definition

Since our project is a offline cloud based one, it does not requires constant and high amount of internet data on the users side to monitor the location of the mobile.

We want our project to be easy accessible to almost all the users regardless of their net connectivity.

We will use all latest methodologies in evaluating the last release to help us improve our processes.

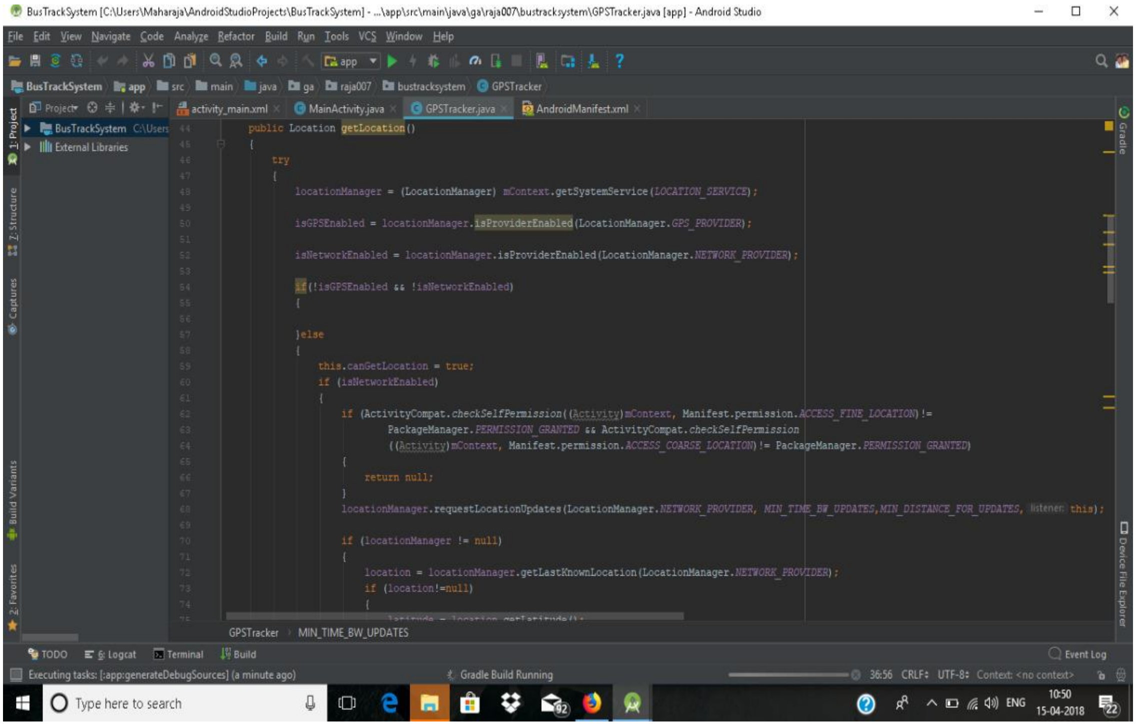
3.2 SYSTEM OVERVIEW

The system consist of two application namely the trace back and chaser application installed in two different mobiles. One sends the location while the other tracks it. They are connected through Google Firebase database.

3.3 MODULO DESCRIPTION

3.3.1 Trackback application and Chaser application

First module is to create trackback application and chaser application using java and xml language in android studio. Trackback application is one that’s installed in lost mobile and continuously sends location to the chaser via firebase database. Chaser application is the one that receives the location of the lost mobile that has trackback application installed in it, for every ten seconds.

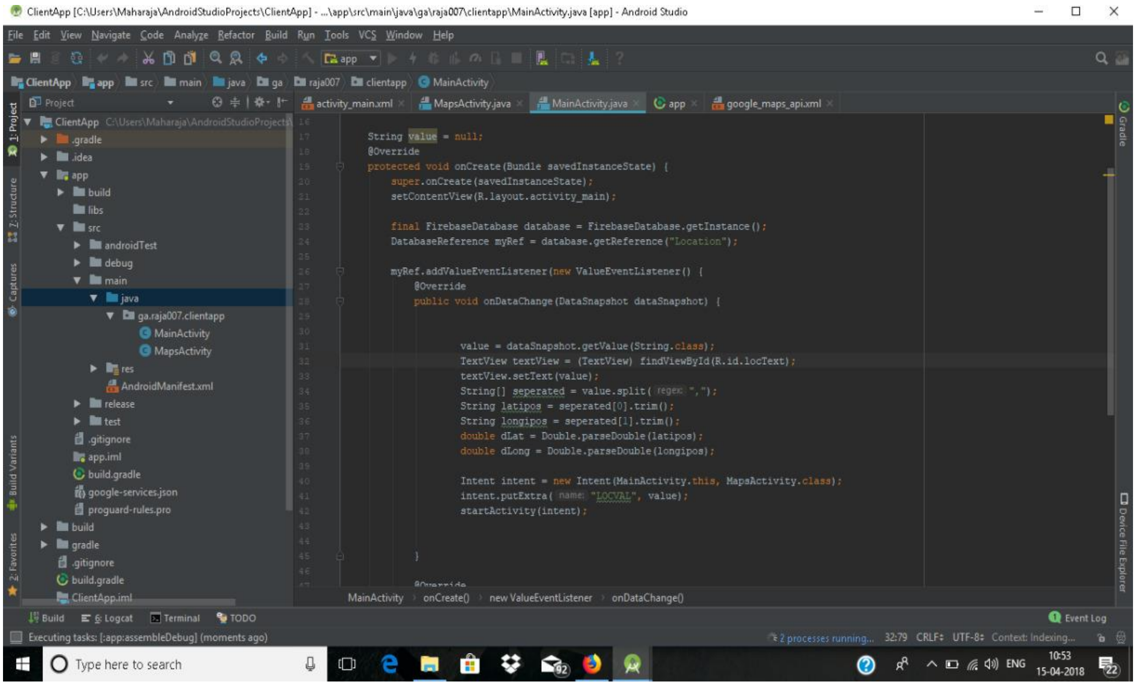


BusTrackSystem [C:\Users\Maharaja\AndroidStudioProjects\BusTrackSystem] - ...\app\src\main\java\ga\raja007\bustracksystem\GPSTracker.java [app] - Android Studio

```java
public Location getLocation()
{
    try
    {
        locationManager = (LocationManager) mContext.getSystemService(LOCATION_SERVICE);

        isGPSEnabled = locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);

        isNetworkEnabled = locationManager.isProviderEnabled(LocationManager.NETWORK_PROVIDER);

        if(!isGPSEnabled && !isNetworkEnabled)
        {

        }else
        {
            this.canGetLocation = true;
            if (isNetworkEnabled)
            {
                if (ActivityCompat.checkSelfPermission((Activity)mContext, Manifest.permission.ACCESS_FINE_LOCATION) !=
                        PackageManager.PERMISSION_GRANTED && ActivityCompat.checkSelfPermission
                        ((Activity)mContext, Manifest.permission.ACCESS_COARSE_LOCATION) != PackageManager.PERMISSION_GRANTED)
                {
                    return null;
                }
                locationManager.requestLocationUpdates(LocationManager.NETWORK_PROVIDER, MIN_TIME_BW_UPDATES,MIN_DISTANCE_FOR_UPDATES, listener: this);

                if (locationManager != null)
                {
                    location = locationManager.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
                    if (location!=null)
                    {
```

```
Executing tasks: [:app:generateDebugSources] (a minute ago)    Gradle Build Running    36:56  CRLF  UTF-8  Context: <no context>
```

Fig 3.1: Code implementation of geo location

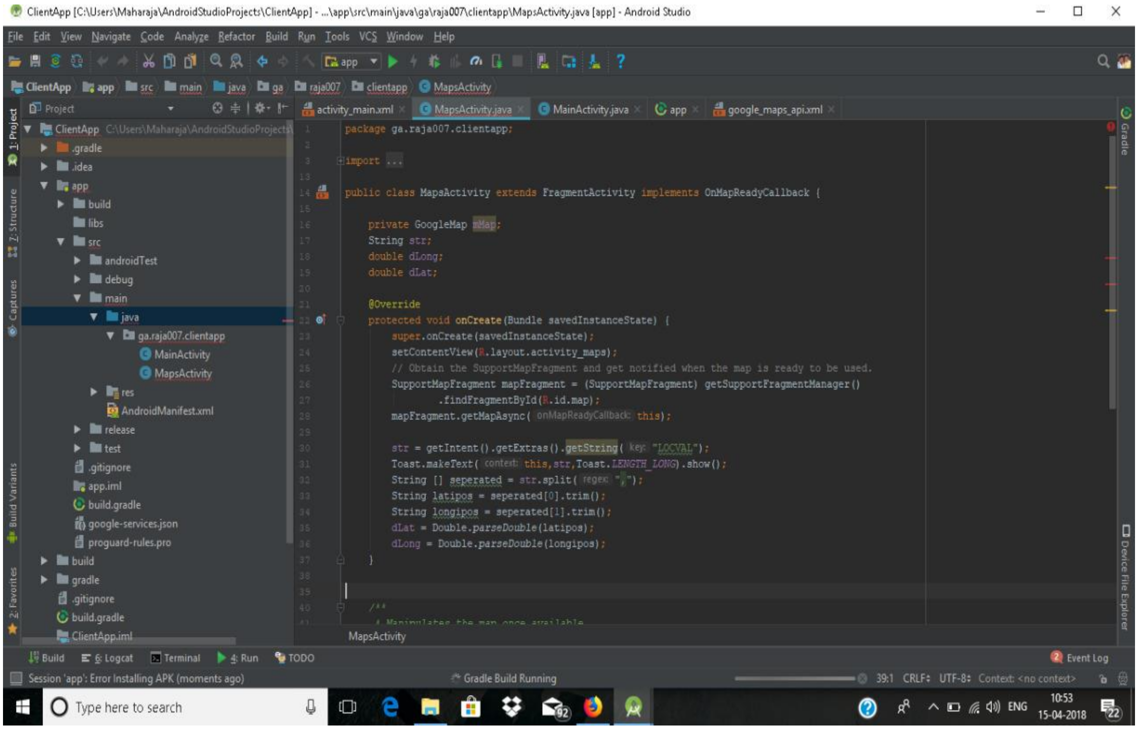


ClientApp [C:\Users\Maharaja\AndroidStudioProjects\ClientApp] - ...\app\src\main\java\ga\raja007\clientapp\MainActivity.java [app] - Android Studio

```java
String value = null;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    final FirebaseDatabase database = FirebaseDatabase.getInstance();
    DatabaseReference myRef = database.getReference("Location");

    myRef.addValueEventListener(new ValueEventListener() {
        @Override
        public void onDataChange(DataSnapshot dataSnapshot) {


                value = dataSnapshot.getValue(String.class);
                TextView textView = (TextView) findViewById(R.id.locText);
                textView.setText(value);
                String[] seperated = value.split( regex: ",");
                String latipos = seperated[0].trim();
                String longipos = seperated[1].trim();
                double dLat = Double.parseDouble(latipos);
                double dLong = Double.parseDouble(longipos);

                Intent intent = new Intent(MainActivity.this, MapsActivity.class);
                intent.putExtra( name: "LOCVAL", value);
                startActivity(intent);


        }
```

```
Executing tasks: [:app:assembleDebug] (moments ago)    2 processes running...    32:79  CRLF  UTF-8  Context: Indexing...
```

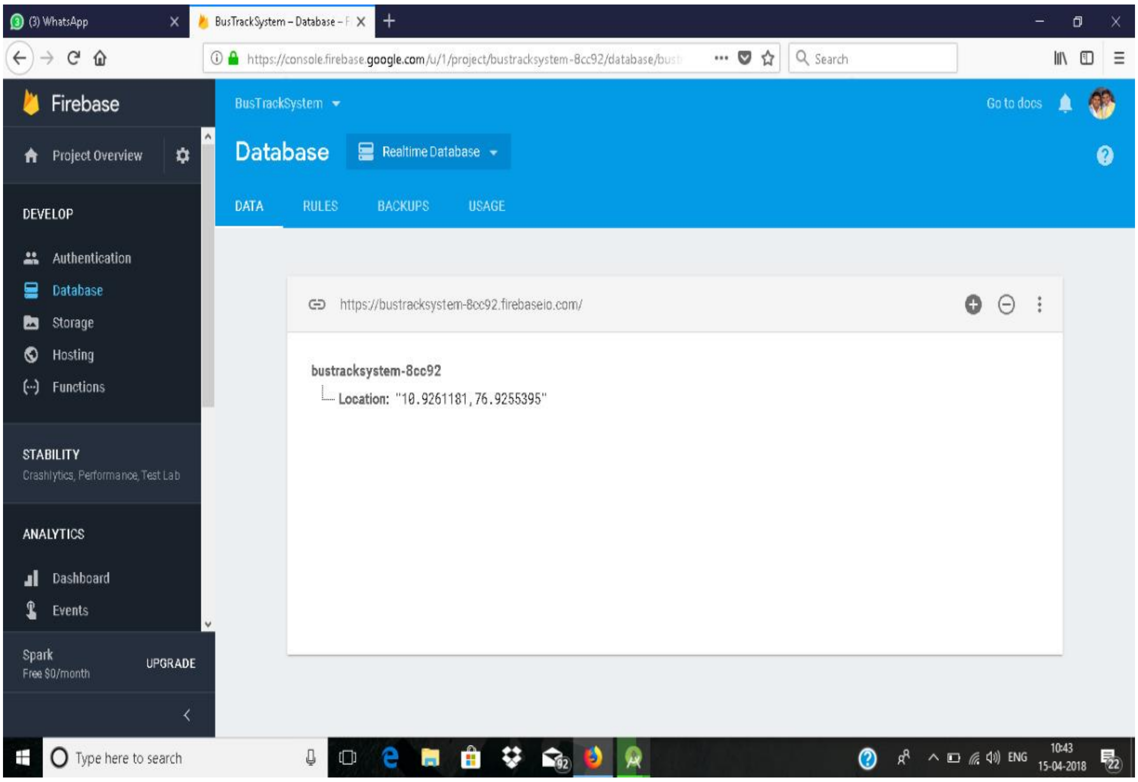
Fig 3.2: Code Implementation of Firebase integration


ClientApp [C:\Users\Maharaja\AndroidStudioProjects\ClientApp] - ...\app\src\main\java\ga\raja007\clientapp\MapsActivity.java [app] - Android Studio
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
ClientApp app src main java ga raja007 clientapp MapsActivity
activity_main.xml MapsActivity.java MainActivity.java app google_maps_api.xml
Project
ClientApp C:\Users\Maharaja\AndroidStudioProjects\
.gradle
.idea
app
build
libs
src
androidTest
debug
main
java
ga.raja007.clientapp
MainActivity
MapsActivity
res
AndroidManifest.xml
release
test
.gitignore
app.iml
build.gradle
google-services.json
proguard-rules.pro
build
gradle
.gitignore
build.gradle
ClientApp.iml
MapsActivity
Build 6: Logcat Terminal 4: Run TODO
Event Log
Session 'app': Error Installing APK (moments ago)
Gradle Build Running
39:1 CRLF: UTF-8: Context: <no context>
Type here to search
10:53
15-04-2018

```
package ga.raja007.clientapp;

import ...

public class MapsActivity extends FragmentActivity implements OnMapReadyCallback {

    private GoogleMap mMap;
    String str;
    double dLong;
    double dLat;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps);
        // Obtain the SupportMapFragment and get notified when the map is ready to be used.
        SupportMapFragment mapFragment = (SupportMapFragment) getSupportFragmentManager()
                .findFragmentById(R.id.map);
        mapFragment.getMapAsync( onMapReadyCallback: this);

        str = getIntent().getExtras().getString( key: "LOCVAL");
        Toast.makeText( context: this,str,Toast.LENGTH_LONG).show();
        String [] seperated = str.split( regex: ",");
        String latipos = seperated[0].trim();
        String longipos = seperated[1].trim();
        dLat = Double.parseDouble(latipos);
        dLong = Double.parseDouble(longipos);
    }


    /**
     * Manipulates the map once available
```

Fig 3.3: Retrival of info from Firebase

3.3.2 Configuring Google Firebase



https://bustracksystem-8cc92.firebaseio.com/

bustracksystem-8cc92
Location: "10.9261181,76.9255395"

Fig 3.4: Firebase data structure

3.3.3 Hardware and Software Requirements

(i)HARDWARE COMPONENTS

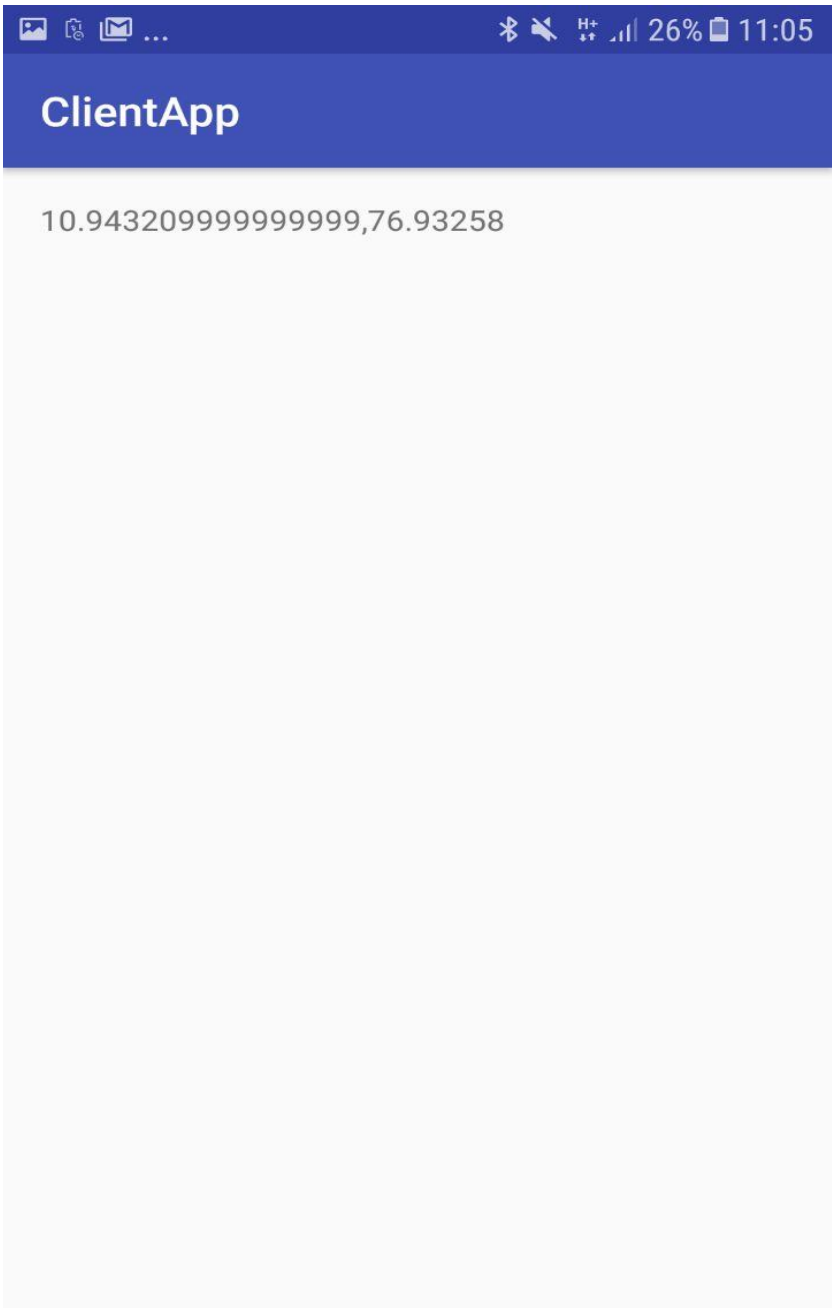
- Mobile phones to download application

(ii) SOFTWARE COMPONENTS

- GOOGLE FIREBASE
- ANDROID STUDIO
- JAVA SE 7
- XML

4. IMPLEMENTATION AND RESULT

The implementation of the project has been successful and the resultant location of the user has been successfully stored and retrieved using firebase cloud service.



10.943209999999999,76.93258

Fig 4.1: Current location of the user

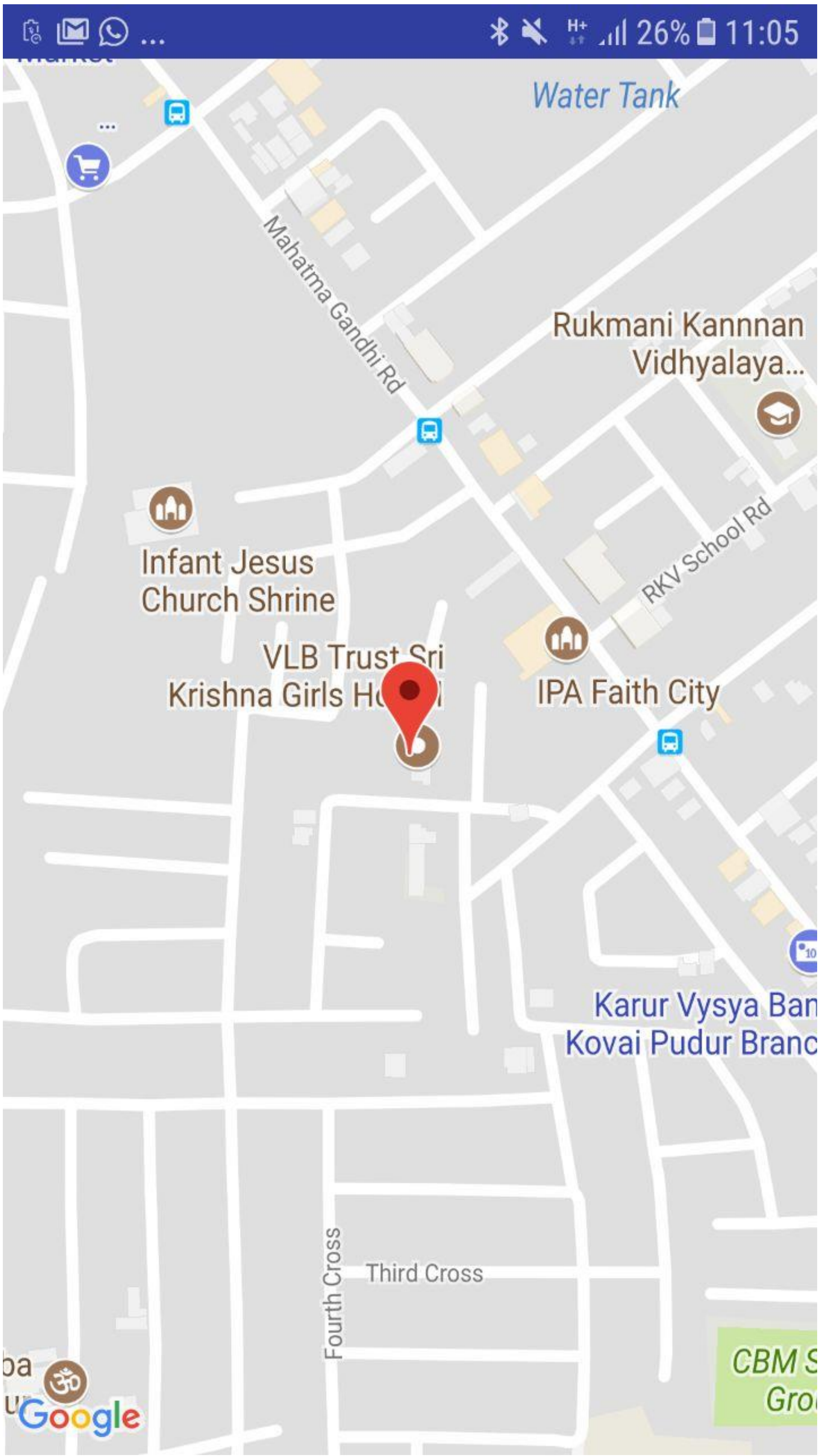

26% 11:05
Water Tank
Mahatma Gandhi Rd
Rukmani Kannnan Vidhyalaya...
Infant Jesus Church Shrine
RKV School Rd
VLB Trust Sri Krishna Girls H
IPA Faith City
Karur Vysya Ban Kovai Pudur Branc
Fourth Cross
Third Cross
CBM S Gro
Google

Fig 4.2: Real time scenario

5. CONCLUSION AND FUTURE ENHANCEMENT

Through the back track application lost mobile could be found faster and easily. There is no need to depend on any other agencies to find our mobile phones.

5.1 CONCLUSION :

Thus this mini project has successfully completed its objective and we believe that it contributes to welfare and security of society.

5.2 FUTURE ENHANCEMENT:

The application can be made for all mobile operating system and can be made available in offline mode.

6. REFERENCES

1. S. Y. Fiawoo dan R. A. Sowah, "Design and Development of an Android Application to Process and Display Summarised Corporate Data," dalam 2012 IEEE 4th International Conference on Adaptive Science and Technology (ICAST), Kumasi, 2012.

2. Z. Chen dan S. Zhu, "The Research of Mobile Application User Experience and Assessment Model," dalam International Conference on Computer Science and Network Technology, Hangzhou, 2011.

3. R. B'Far, Mobile Computing Principles: Designing and Developing Mobile Applications with UML and XML, Cambridge: Cambridge University Press, 2005.

4. S. Anjaneyulu Pasala, N. S, R. P. Gorthi dan K. B. Gadde, "Context-Aware Mobile Assistant Agents in Software Project Management," dalam TENCON 2008 - 2008 IEEE Region 10 Conference, Bangalore, 2008.