e-ISSN: 2582-5208
International Research Journal of Modernization in Engineering Technology and Science
Volume:03/Issue:01/January-2021 Impact Factor- 5.354 www.irjmets.com

APPROACH ON BESAFE ANDROID APPLICATION

Prof. Bijender Tyagi[*1], **Piyush Jain**[*2], **Avi Gupta**[*3], **Nitya Gupta**[*4], **Nishchint Gauniyal**[*5]

[*1,2,3,4,5]Department of Information Technology, Dr. Akhilesh Das Gupta Institute of Technology and Management , Delhi.

ABSTRACT

In today's world, smartphones have become a necessity and their usage is increasing rapidly. There may be a situation when a user is not around his phone and it's important to make sure that everything should be secure and properly managed. Also, places where getting good internet connectivity is difficult, there's always a need for an application which doesn't require any internet for above mentioned purposes.

With an application like "BeSafe'', during an emergency, you can keep your data secure and still give access to others to manage a few features on your smartphones such as screen lock, wifi/data on/off, profile management(silent/DND/ringer) or location details. This proposed application is totally based around SMS where one user can access others phone through SMS but only after a user sends a specific code through SMS to the phone it has access to, other users will automatically do the assigned task respectively to that SMS code.

Its interface is user friendly and easily accessible. Also, internal communication is done through SMS which makes it quite simple.

Keywords - Intent, Android, Framework, Smartphone, geolocation.

I. INTRODUCTION

With the flourishing recognition of the internet, this application focuses on communicating only through SMS which makes it easy and simple to use. SMS is the short message service used for exchanging short text messages and supported by many operating systems.

BeSafe is a user-friendly android application that comes in handy if a user wants to perform some actions like location on/off, Wi-Fi on/off, change profile from general to DND or vice versa, and screen locking but not able to find his phone. This application allows the user to give his smartphone access to another person.

To perform these actions, a user has to send a specific code through SMS to the phone it has access to, the user will automatically send/do the assigned task/information respectively to that SMS code. It is secure and doesn't let anyone have your personal information.

Product feature it offers is:

Where are you? – This feature allows the user to get the latest location with the exact latitude and longitude details for friends and family if needed through SMS.

SCREAM: This feature allows to change mobile's profile from silent/DND/vibrate to full ringer mode.

SCREEN LOCK: Mobile screen can be locked if left unlocked just by SMS.WIFI ON/OFF: Mobile data/Wi-Fi can be enabled or disabled with the help of this feature.

II. MOTIVATION

We often lose our phones or forget where we have put them in a house or anywhere for that matter but, we can find it if it is not on silent mode or if internet connectivity is on. However, there are times when the phone is on silent mode and there is no internet connection on the device, in those conditions it becomes very hectic to search the phone. Hence, we came up with this application based on SMS.

III. PROBLEM STATEMENT

Nowadays the internet is very important in our lives and since it has many benefits, it is being used by everyone. However, a strong network is required for the internet to work but, there are places where there is no strong network- lift is one such example where there is no stable network hence the internet connectivity- whereas short messaging services i.e. SMS don't need as strong a network as the internet needs to work, SMS can be sent over a low network.

IV. PROPOSED SOLUTION

To solve the problem stated above we have worked on this "BeSafe" application which is completely based on SMS. In this, the user will input a phone number in the given field which then will give permission to that number to access various features such as locking the phone, disable or on wifi, set phone to highest ringing mode, or to get the location of the person who has authorized you to access his phone all this just by sending an SMS, when this section is triggered user's phone will automatically send an SMS to the same number it was triggered from this SMS will contain the user's coordinates in the form of longitude and latitude and will be visible to the user using google maps.

A. Algorithm:

1. First, we store the phone number of a user (+91XXXXXXXXXX) whom the user wants to give access to.
2. If the user received a message(SMS) then the application will check if that message is from a registered user or not. (SmsReceiver.java)
3. Suppose the message received is from a registered user then the application will check the content of the message, if the content is equal to any of the keywords(There are some keywords and each keyword is assigned a task) then it will perform a task assigned to that keyword.
4. Some of the assigned tasks are screen lock, GPS locations, change to full ringer mode, etc.

B. FlowChart for Proposed System

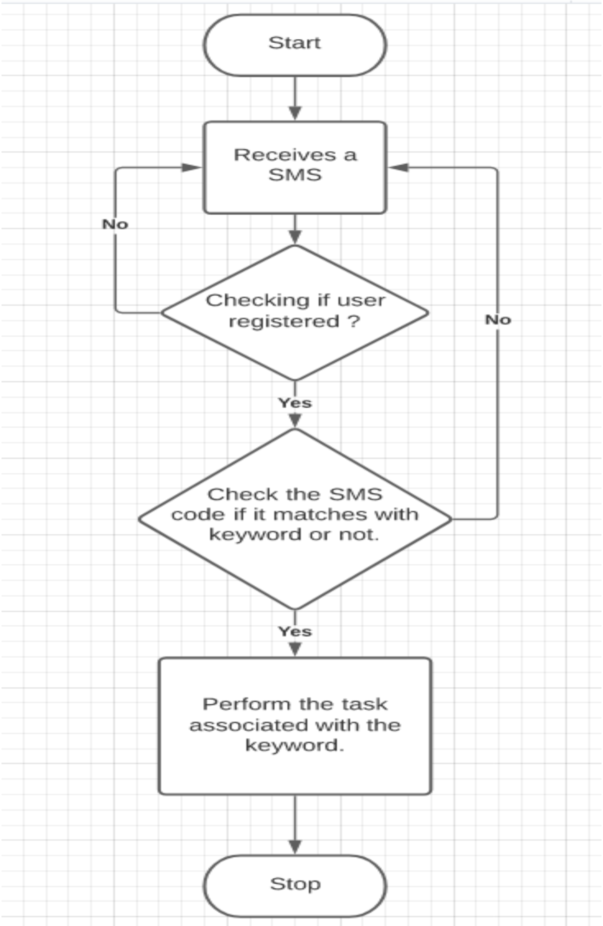

Start
Receives a SMS
No
Checking if user registered ?
No
Yes
Check the SMS code if it matches with keyword or not.
Yes
Perform the task associated with the keyword.
Stop

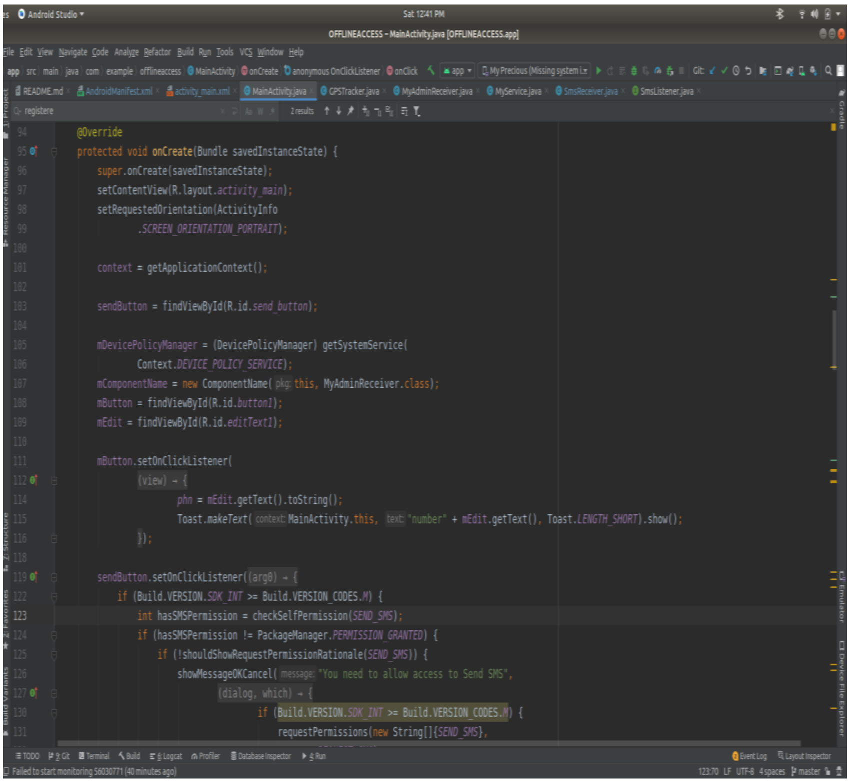
Fig.-1: Flow Chart

V. CLASSES, LIBRARIES, AND THEIR FUNCTIONALITIES

Mainactivity.java

An Android Application usually consists of multiple screens that are loosely bound to each other where each screen represents an activity. Typically, one activity in an application is specified as the "main" activity (MainActivity.java), which is presented to the user when the application is launched. The main activity can then start other activities to perform different actions.

Each time a new activity starts, the previous activity is stopped, but the system preserves the activity in a stack (the "back stack"). When a new activity starts, that new activity is pushed onto the back stack and takes user focus. The back stack follows basic "last in, first out" stack logic. When the user is done with the current activity and presses the Back button, that activity is popped from the stack and destroyed, and the previous activity resumes.



Android Studio
Sat 12:41 PM
OFFLINEACCESS – MainActivity.java [OFFLINEACCESS.app]
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
app src main java com example offlineaccess MainActivity onCreate anonymous OnClickListener onClick
app
My Precious (Missing system i...
Git:
README.md AndroidManifest.xml activity_main.xml MainActivity.java GPSTracker.java MyAdminReceiver.java MyService.java SmsReceiver.java SmsListener.java
registere
2 results
1: Project
Resource Manager
Z: Structure
2: Favorites
Build Variants
Gradle
Emulator
Device File Explorer
TODO 9: Git Terminal Build 6: Logcat Profiler Database Inspector 4: Run
Event Log Layout Inspector
Failed to start monitoring 56030771 (40 minutes ago)
123:70 LF UTF-8 4 spaces master

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    setRequestedOrientation(ActivityInfo
            .SCREEN_ORIENTATION_PORTRAIT);

    context = getApplicationContext();

    sendButton = findViewById(R.id.send_button);

    mDevicePolicyManager = (DevicePolicyManager) getSystemService(
            Context.DEVICE_POLICY_SERVICE);
    mComponentName = new ComponentName( pkg: this, MyAdminReceiver.class);
    mButton = findViewById(R.id.button1);
    mEdit = findViewById(R.id.editText1);

    mButton.setOnClickListener(
            (view) → {
                phn = mEdit.getText().toString();
                Toast.makeText( context: MainActivity.this,  text: "number" + mEdit.getText(), Toast.LENGTH_SHORT).show();
            });

    sendButton.setOnClickListener((arg0) → {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
            int hasSMSPermission = checkSelfPermission(SEND_SMS);
            if (hasSMSPermission != PackageManager.PERMISSION_GRANTED) {
                if (!shouldShowRequestPermissionRationale(SEND_SMS)) {
                    showMessageOKCancel( message: "You need to allow access to Send SMS",
                            (dialog, which) → {
                                if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {
                                    requestPermissions(new String[]{SEND_SMS},
```

Fig.-2: MainActivity.java file

An activity is started or activated with an intent. An Intent is an asynchronous message that you can use in your activity to request an action from another activity, or from some other application component. You use an intent to start one activity from another activity, and to pass data between activities.

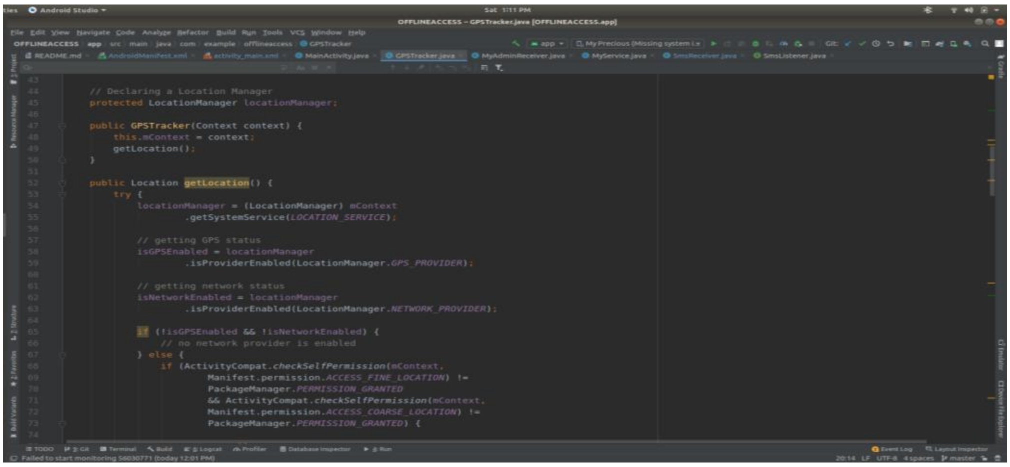
We have multiple functions in our MainActivity.java such as :

1.SendMySMS : Which will send one user's location to a registered user who would have asked for location.

2.SmsReceiver.bindListerner: This function will check the received messages and check whether the code of the message is related to the keyword of services that we are providing and after checking that it will check whether that number is registered or not. If the number is registered it will perform the function that is requested. and the rest of the functions are generic functions like onCreate, on Activity Result, etc.

GPSTracker.java

This class will help to extract the location of the device based on latitude and longitude and we will be able to send that location “latitude and longitude” via sms to the device whom we have given the permission to access the device and via that message user can track another user via google maps.



```java
    // Declaring a Location Manager
    protected LocationManager locationManager;

    public GPSTracker(Context context) {
        this.mContext = context;
        getLocation();
    }

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

            // getting GPS status
            isGPSEnabled = locationManager
                    .isProviderEnabled(LocationManager.GPS_PROVIDER);

            // getting network status
            isNetworkEnabled = locationManager
                    .isProviderEnabled(LocationManager.NETWORK_PROVIDER);

            if (!isGPSEnabled && !isNetworkEnabled) {
                // no network provider is enabled
            } else {
                if (ActivityCompat.checkSelfPermission(mContext,
                        Manifest.permission.ACCESS_FINE_LOCATION) !=
                        PackageManager.PERMISSION_GRANTED
                        && ActivityCompat.checkSelfPermission(mContext,
                        Manifest.permission.ACCESS_COARSE_LOCATION) !=
                        PackageManager.PERMISSION_GRANTED) {
```

Failed to start monitoring 56030771 (today 12:01 PM)

Fig.-3: GPSTracker.java file

Inside this class there is a reference location Manager,

Location Manager : This class provides access to the system location services. These services allow applications to obtain periodic updates of the device's geographical location, or to be notified when the device enters the proximity of a given geographical location.

This class provides several function, functions that are used in this project is listed below:

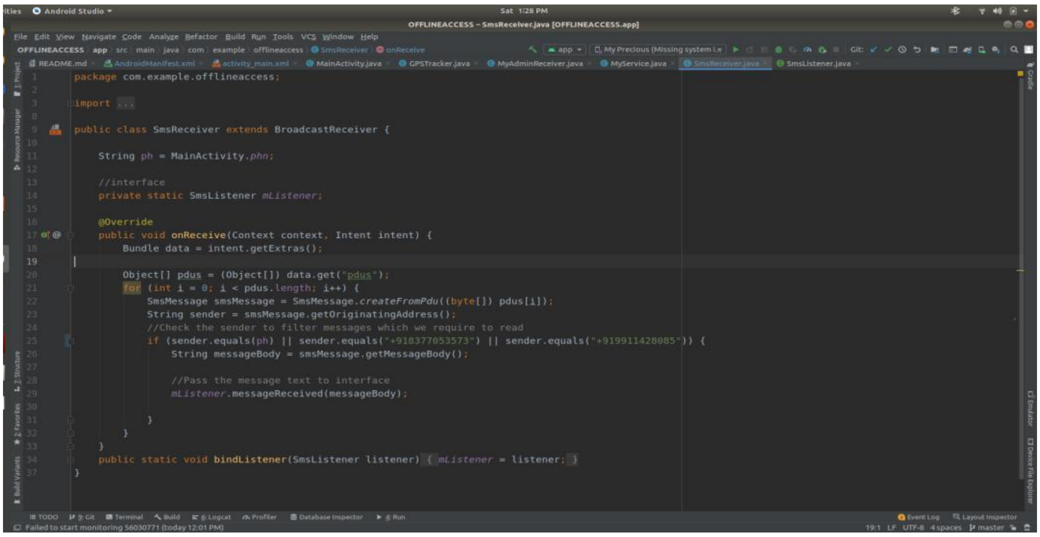
Is Provider Enabled(String provider):

Returns the current enabled/disabled status of the given provider.

Get Last Known Location(String provider):Gets the last known location from the given provider, or null if there is no last known location.

Remove Updates(Location Listener listener):Removes location updates for the specified Location Listener.

SmsReceiver.java



```java
package com.example.offlineaccess;

import ...

public class SmsReceiver extends BroadcastReceiver {

    String ph = MainActivity.phn;

    //interface
    private static SmsListener mListener;

    @Override
    public void onReceive(Context context, Intent intent) {
        Bundle data = intent.getExtras();

        Object[] pdus = (Object[]) data.get("pdus");
        for (int i = 0; i < pdus.length; i++) {
            SmsMessage smsMessage = SmsMessage.createFromPdu((byte[]) pdus[i]);
            String sender = smsMessage.getOriginatingAddress();
            //Check the sender to filter messages which we require to read
            if (sender.equals(ph) || sender.equals("+918377053573") || sender.equals("+919911428085")) {
                String messageBody = smsMessage.getMessageBody();

                //Pass the message text to interface
                mListener.messageReceived(messageBody);

            }
        }
    }
    public static void bindListener(SmsListener listener) { mListener = listener; }
}
```

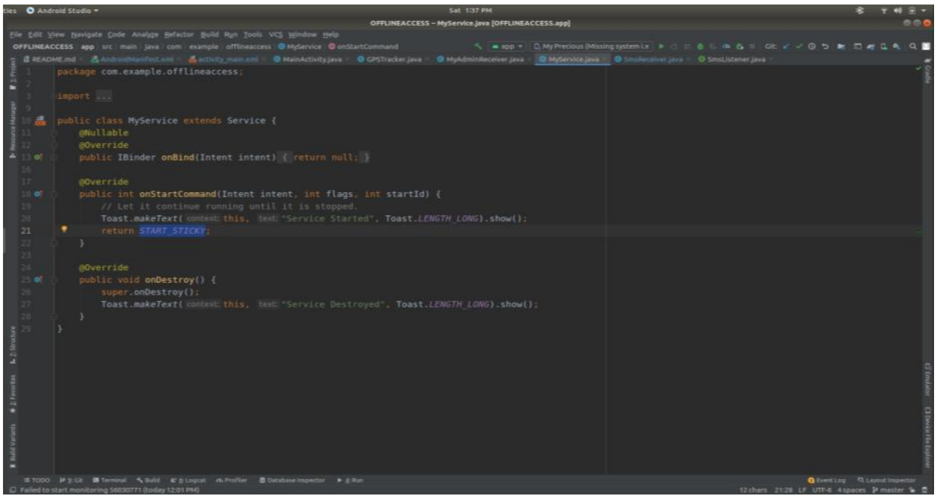
Failed to start monitoring 56030771 (today 12:01 PM)

Fig.-4: SmsReceiver.java file

This class will help to check whether the user(phone-number) is registered or not if the number is registered then only the tasks will take place for this we are using java bean i.e mlistner this will help us to read the SMS and scan whether there is a keyword that we should check or not.

Myservice.java

This class will receive a signal from MainActivity.java that whether we should start the service or stop in case we have started the service it will show a toast on the application that service has started and we would be able to allow smsListener to check for the Keywords.



```java
package com.example.offlineaccess;

import ...

public class MyService extends Service {
    @Nullable
    @Override
    public IBinder onBind(Intent intent) { return null; }

    @Override
    public int onStartCommand(Intent intent, int flags, int startId) {
        // Let it continue running until it is stopped.
        Toast.makeText( this, "Service Started", Toast.LENGTH_LONG).show();
        return START_STICKY;
    }

    @Override
    public void onDestroy() {
        super.onDestroy();
        Toast.makeText( this, "Service Destroyed", Toast.LENGTH_LONG).show();
    }
}
```

Fig.-5: Myservice.java file

VI. TOOL USED

Android Studio

Android Studio is an Integrated Development Environment (IDE) based on Intellij IDEA to develop Android Applications. Its purpose is to build high-quality Applications and to accelerate your development period. There are various approaches to Android Development but Android Studio by far is the most powerful and efficient way to develop Android Applications.

To support application development within the Android operating system, Android Studio uses a Gradle-based build system, emulator, code templates, and GitHub integration. Every project in Android Studio has one or more modalities with source code and resource files. These modalities include Android application modules, Library modules, and Google App Engine modules.

Android Studio uses an Instant Push feature to push code and resource changes to a running application. A code editor assists the developer with writing code and offering code completion, refraction, and analysis. Applications built-in Android Studio are then compiled into the APK format for submission to the Google Play Store.

What is Android

Android is an open-source Linux-based operating system that was primarily built for small portable Touchscreen devices such as smartphones, tablets, etc. Android was first founded in Palo Alto in California in 2003.

The android architecture is divided into five sections and four main layers that are.

- Linux kernel- The android is built on the most powerful Linux kernel. This layer stands at the bottom of the android software stack and provides basic functionalities such as Process Management, Device Management, Memory Management, etc. Among all the other components this layer provides the main functionalities to the device and it also provides a platform to run all the android applications that are needed.

- Android runtime- The android runtime provides a runtime environment called Dalvik Virtual Machine. This runtime environment makes use of android features such as multithreading, memory management, etc so that every android application can run its own process.

- Libraries- There is a set of Java-based libraries to handle certain tasks as File Manipulation, SSL libraries, WebKit, library libc, etc that are provided by the Android runtime that are specific to android development.

- Application Framework- The application framework layer provides accessibility to higher-level android services or features that use java API. This layer allows android applications to securely access and utilizes android device resources. The services this layer provides are the Activity Manager, Window Manager, Location Manager, Telephony Manager, resource manager, view system, Notification Manager, package manager, and Content Providers.

VII. RESULTS AND DISCUSSION

As long as the application is in service and other user's mobile number is registered, we are able to perform some tasks such as:

1.) Extract the location of the device from another device.

2.) Change the mobile profile from silent or vibration to complete ringer.

3.) Enable/Disable wifi.

4.) Apply lock to an unlocked device.

All of the above functionalities are totally based on SMSs no wifi or mobile data is required.

VIII. CONCLUSION

Summing it all up, the BeSafe Android application, as the name suggests is the application that makes sure that your phone is in safe hands. It allows a user to give access to another person through SMS so that another user can perform some assigned tasks. Features it performs are location on/off, WiFi on/off, screen lock, and profile (general to DND or vice versa)

Future Scope

The project can be made better by

1. Firstly by making the UI more attractive and more user-friendly.
2. Secondly by adding more features like erasing all data of the device, switching off the device, etc.

IX. REFERENCES

[1] ADMOB. AdMob Android SDK: Installation Instructions. http://www.admob.com/docs/AdMob_Android_SDK_Instructions.pdf. Accessed November 2010

[2] ONGTANG, M.BUTLER, K., AND MCDANIEL, P. Porscha: Policy Oriented Secure Content Handling in Android. In Proc. of the Annual Computer Security Applications Conference (2010)

[3] OL. Google Android Developers, Android Developer Guide. http://developer.android.com/guide/topics/fundamentals.html

[4] J.F. DiMarzio "Android a programmer's guide".

[5] William Enck, Damien Octeau, Patrick McDaniel, and Swarat Chaudhuri "A Study of Android Application Security".

[6] KRALEVICH, N. Best Practices for Handling Android User Data. http://android-developers.blogspot.com/2010/ 08/best-practices-for-handling-android.html, 2010.

[7] Manav Singhal, Anupam Shukla "Implementation of Location based Services in Android using GPS and Web Services".