

# AN ANALYSIS OF WORLD POPULATION LEADING TO THE NECESSITY OF LOST ALARM: AN APPLICATION TO FIND YOUR MOBILE DEVICE

SWASTI SAXENA

B.tech 2nd year CSE, SRM University, Kattankulathur

**Abstract**— Mobile phone has become a necessity of life. Or rather smart phone has become a lifeline of a modern human being. 84 percent of people surveyed in a new TIME Mobility Poll said they couldn't go a single day without their mobile device in hand and seventy-five percent of those ages 25 to 29 sleep with their mobile phones. Still many of us forget that where did we keep our phone last and we search for it and spend our time and energy in finding the device. The situation becomes more critical if the phone is in silent mode. This problem can be resolved by using Lost Alarm application.

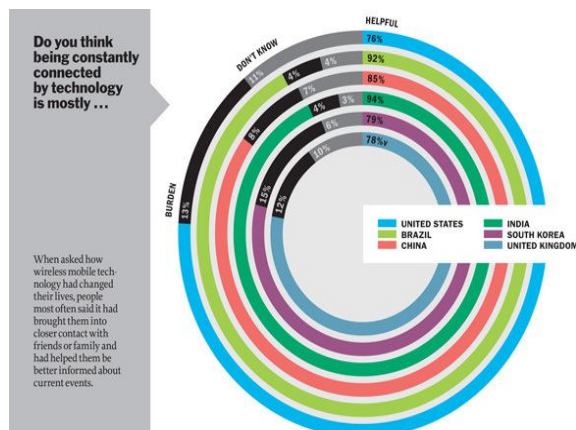
**Keywords**— Mobiles lost, Find lost Android device, Lost alarm, Voice Command find.

## I. INTRODUCTION

Mobile phones: necessity of life, yet people lose them often. One in four people check their phones every thirty minutes, while one in five check every ten. Researchers polled nearly 5,000 people in eight countries: the U.S., the UK, China, India, South Korea, South Africa, Indonesia and Brazil, according to Time.

More than 9 in 10 Brazilians and Indians agree being constantly connected is "mostly a good thing," only 76 percent of Americans felt the same way.

To better understand attitudes about mass mobility, Time, in cooperation with Qualcomm, launched the [Time Mobility Poll](#), a survey of close to 5,000 people of all age groups and income levels in eight countries: the U.S., the U.K., [China](#), [India](#), South Korea, South [Africa](#), Indonesia and Brazil. Even the best survey can be only a snapshot in time, but this is a crisp and textured one—revealing a lot about both where we are now and where the mobile wave is taking us next.



Losing things is irritating and yet we are a forgetful people. The average person misplaces up to nine items a day, and one-third of respondents in a poll said they spend an average of 15 minutes each day searching for items—cellphones, keys and paperwork

top the list, according to an online survey of 3,000 people published in 2012 by a British insurance company. Everyday forgetfulness isn't a sign of a more serious medical condition rather stress, fatigue, and multitasking can exacerbate our propensity to make such errors.

### Mobiles on silent mode

Phone silence has become ubiquitous thus according to a [Quora poster](#), all the young, hip "millennials" have switched their phones to silent mode, "even though they use them incessantly." "It spares the person from the constant beeping of their cell and also from the weird looks that one receives when one's phone keeps on beeping." This is one of the contributing reasons that why most of the people find it difficult to find their mobile phones where they last kept. As they are not even able to hear the ringtone of their mobile phone to find it where they last kept since it was on silent mode.

### Lost Alarm: An application to find your lost phone

In everyday life when we leave our phone somewhere nearby and we are unable to find it, at that time and energy is spent in finding our lost device without which an everyday life of any common individual cannot be imagined.

Till now the most common trick that anybody uses in everyday life is to dial our mobile number from someone else's phone and then try to figure out where the phone is ringing. This trick comes into play nearly every other day but the greatest issue arises when you had left your phone on silent mode maybe at work and had never switched it to normal mode.

As mentioned earlier 9 out of 10 people prefer to keep their mobiles on silent mode so that nobody else get disturbed because of their messages or caller tones. Thus in such a scenario it becomes a herculean task to find our mobile phone.

Thus to resolve this issue of our everyday life Lost Alarm is the best application to go for.

Lost alarm is an android application in which we can find our lost phone by a two way technique. First Method : If we kept our mobile in a nearby place and are unable to find it back , then we can use voice command ‘find active’ to start our mobile taking voice commands. After that we should call out our mobile number. If the number saved as my number in the contact list matches the number called then the phone’s alarm rings for next 30 seconds along with the turning on of the flash light. The alarm rings even in silent mode as set in default settings of any smartphone.

Using Google speech recognition module we can effectively let our device understand as we speak. We’ll implement the search functionality of an e-commerce application. In our application we’ll import

Import.android.speech.RecognizerIntent or we can simply click on ctrl+shift+o to include all the necessary files that are required. In java file we’ll write the code as

```
Button.setOnClickListener() //this initiates the intent and we map into the action to recognize speech
{
Public void onClick(View v)
Intent intent=new
intent(RecognizerIntent.ACTION_RECOGNIZER_S
PEECH)
Intent.putExtras(RecognizerIntent.Language_MODE
L_FREE_FORM);
Intent.putExtra(RecognizerIntent.EXTRA_MAX_RE
SULTS,1);
Intent.putExtra(RecognizerIntent.EXTRA_LANGUA
GE,Locale.English);
startActivityResult(intent,1);
}
```

Then we send some parameters along with intent that will be sent as Extras which will help in speech recognition. Through putExtras we’ll add Language model . This will help in understanding what has been spoken. We’ll take it in ‘Freeform’. Next we’ll add Extras for Google max results and set it to 1 so that google may select the first and only matched result i.e. our set command . Next we’ll add extras to add language and give English as locale language. Further for results we’ll set it as 1 so that the application understands that the speech recognition is the first activity to be performed . Thus as we say find active then the application which is kept in resume mode comes to foreground. Thus the voice command are accepted after the application activation.

For sure there are some limitations that the background should not be too noisy, the mobile should not be too far from the user so as not to be able to accept voice commands and the speech should be clear for the device to recognize. In spite of all these barriers this approach will prove to be useful in future.

Second Method: By sending a sms through any mobile device and simply typing ‘find active’, the Lost Alarm application gets activated. After that the phone rings for 30seconds along with the turning on of the flash light which would help in easily finding the mobile device even in dark. Here also the phone would ring the alarm for 30 seconds even if the mobile phone is in silent mode.

#### Using Built-in Intent to send SMS

We can use Android Intent to send SMS by calling built-in SMS functionality of the Android.

#### Intent Object - Action to send SMS

We will use ACTION\_VIEW action to launch an SMS client installed on our Android device. Following is simple syntax to create an intent with ACTION\_VIEW action

```
Intent smsIntent = new
Intent(Intent.ACTION_VIEW);
```

#### Intent Object - Data/Type to send SMS

To send an SMS you need to specify smsto: as URI using setData() method and data type will be to vnd.android-dir/mms-sms using setType() method as follows –

```
smsIntent.setData(Uri.parse("smsto:"));
smsIntent.setType("vnd.android-dir/mms-sms");
```

#### Intent Object - Extra to send SMS

Android has built-in support to add phone number and text message to send an SMS as follows –

```
smsIntent.putExtra("address" , new
String("0123456789;3393993300"));
smsIntent.putExtra("sms_body" , "Test SMS to
Angilla");
```

Here address and sms\_body are case sensitive and should be specified in small characters only. You can specify more than one number in single string but separated by semi-colon (;).

#### Example

Following example shows you in practical how to use Intent object to launch SMS client to send an SMS to the given recipients.

You can use BroadcastReceiver for reading sms. And extract that sms and save values in DataBase in android . When you call the first Activity check the particular value contains in the DataBase then only start the Activity.

```
public class ReadingMessage extends
BroadcastReceiver {
@Override
public void onReceive(Context context, Intent intent)
{
//---get the SMS message passed in---
Bundle bundle = intent.getExtras();
DBAdapter dbHelper = new DBAdapter(context);
SmsMessage[] msgs = null;
String msg=null;
String str=null;
if (bundle != null)
{
```

```

    ---retrieve the SMS message received---
    Object[] pdus = (Object[]) bundle.get("pdus");

    msgs = new SmsMessage[pdus.length];
    for (int i=0; i<msgs.length; i++){
        msgs[i] =
        SmsMessage.createFromPdu((byte[])pdus[i]);

        msg = msgs[i].getMessageBody().toString();
        str =msg.toUpperCase();

        if(str.contains("your value"))
        {
            try{
                dbHelper.open();

                dbHelper.insertinfo(msg);

                dbHelper.close();

            }
            catch(Exception e)
            {
                e.toString();
            }

        }
    }
}

```

This code for Reading SMS.

```

public class StartActivity extends Activity{

    private static final int
    ACTIVITY_REGISTRATION1=0;
    private static final int
    ACTIVITY_SENDALERT3=1;
    private static final int
    ACTIVITY_REGISTRATION2 = 2;

    Context context;
    DBAdapter dbHelper=null;
    Intent intent;
    String db_activation=null;
    Cursor cursor;

    public StartActivity()
    {
        this.context=this;
    }

    @Override

    /* Method Header
    * Method Name : onCreate
    * Input Parameter : Bundle

```

```

    * Return Value : nil
    */
    protected void onCreate(Bundle savedInstanceState)
    {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);

        dbHelper=new DBAdapter(this);

        try
        {
            dbHelper.open();

            cursor = dbHelper.getActivtaion();
            if(cursor.getCount()==0)
            {

                intent=new Intent(this,Registration.class);

                intent.addFlags(Intent.FLAG_ACTIVITY_BROUGHT_TO_FRONT);

                startActivityForResult(intent,ACTIVITY_REGISTRATION1);
            }
            else
            {
                for(int i=0;i<cursor.getCount();i++)
                {
                    cursor.moveToNext();
                    db_activation =
                    cursor.getString(cursor.getColumnIndex(DBAdapter.KEY_ACTIVATION));

                    if(db_activation.equals("1"))

                    {

                        intent=new Intent(this,SendAlert.class);

                        intent.addFlags(Intent.FLAG_ACTIVITY_BROUGHT_TO_FRONT);

                        startActivityForResult(intent,ACTIVITY_SENDALERT3);

                    }
                    else
                    {

                        intent=new Intent(this,Registration.class);

                        intent.addFlags(Intent.FLAG_ACTIVITY_BROUGHT_TO_FRONT);

```

```

startActivityForResult(intent,ACTIVITY_REGISTR
ATION2);
    }

    dbHelper.close();
}
}
}
catch(Exception e)
{
finish();
System.exit(0);
e.toString();
}
}
@Override
protected void onDestroy() {
// TODO Auto-generated method stub
super.onDestroy();
finish();
}

protected void onActivityResult(int requestCode, int
resultCode, Intent data)
{
super.onActivityResult(requestCode, resultCode,
data);
if (resultCode == Activity.RESULT_OK)
finish();
}
}
}
this code for the First Activity
public long insertTruckinfo(String db_Truckmsg)
{
ContentValues cVal=new ContentValues();

cVal.put(KEY_INFO,db_Truckmsg);

return db.insert(TRUCKINFO_TABLE,
null,cVal);
}

```

```

public Cursor getActivtaion()
{
Cursor cursor =db.query(ACTIVATION_TABLE,
new String[] {KEY_ID,KEY_ACTIVATION},
null,null, null, null, null);
return cursor;
}

public Cursor getTruckinfo()
{
Cursor cursor =db.query(TRUCKINFO_TABLE,
new String[] {KEY_ID,KEY_INFO}, null,null, null,
null, null);
return cursor;
}

```

## CONCLUSION

Misplacing of mobile phones in every day life has caused havoc on day to day basis especially when you are getting late and your mobile was on silent mode and you are unable to find it. Such a situation is faced by most of us on frequent basis. The problem can be resolved by using Lost Alarm application. This would help one find his/her phone even when the phone is on silent mode with minimum efforts.

## REFERENCES

- [1] Wilson, B. S., Finley, C. C., Lawson, D. T., Wolford, R. D., Eddington, D. K. & Rabinowitz, W. M. (1991) *Nature* **352**, 236-238. [CrossRefMedlineWeb of Science](#)
- [2] Alexandros, P. & Maragos, P. (1999) *Speech Commun.* **28**, 195-209.
- [3] Sheft, S. & Yost, W. A. (2001) *Air Force Research Laboratory Progress Report No. 1, Contract SPO700-98-D-4002* (Loyola University, Chicago).
- [4] Google Play Android Market, <https://play.google.com/store/apps>
- [5] Android (operating system), [http://en.wikipedia.org/wiki/Android\\_\(operating\\_system\)](http://en.wikipedia.org/wiki/Android_(operating_system))
- [6] Wikipedia.

★★★