NORTH EASTERN HILL UNIVERSITY

A PROJECT REPORT
ON

ANDROID SMS APP

SUBMITTED BY:

ENRICO MARVLE THABAH

BCA 3rd yr

ROLL -NO:P1400010

REG-NO-14525 of 2013-2014

DEPARTMENT OF COMPUTER SCIENCE & APPLICATIONS
SHILLONG COLLEGE
SHILLONG 793003
MEGHALAYA.
April, 2017

CERTIFICATE

This is to certify that ENRICO MARVLE THABAH of BCA 3rd year has successfully completed the project work on Android Sms App for BCA 3rd year final examination 2017 of North Eastern Hill University in the year 2017. It is Further Certified that the project is individual work of the candidate.

Shri Nicholas P.Jyrwa
Guidance
Dept. Of Comp. Sci.&App.
Shillong College

17/4/17

Smt. Aiom Mitri
HOD
Dept. Of Comp. Sci.&App.
Shillong College
Shillong

External examiner

ACKNOWLEDGEMENT

I have taken effort in this project. However, it would not be possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them. I would like to offer my special thanks to Sir Nicholas P.Jyrwa for his guidance and for providing necessary information regarding the project. I also would like to express my gratitude to Smt Aiom Mitri, HOD and other teachers of the Department Of Computer Science And Application for their kind co-operation and encouragement which help me in complete this project.

I also thank Dr. K.D Ramsej, Principal, Shillong College for giving
me the opportunity to work on this project.

Thanking You

Enrico Marvle Thabah

CONTENTS

Abstract

"Android Sms App" is the very simple form of application which is used pervasively around the globe. The service allows for sending and receiving messages from one cell phone to another.

It is the application which shows up when we press the sms icon app which is shown below. It is a very small size app that it doesn't consume your memory.

A lot of devices have their own default sms application for sending and receiving messages. But here I have created my own simple messaging app which is extremely easy and fast to use.

Purpose of Using this App

The purpose of using this app is to send, receive and read text messages. Most devices have their own default messaging app but here I have created a very fast, simple and easy to use messaging app by simply writing the message and the number to whom we want to send and press the button SEND IT to send the message.

It has got a nice feature that we commonly need as a messaging app with simple and nice interface.

Requirements

We need to have the following installed and configured on our development machine:

❖ Mobile Application:

- Android version Lollipop 5.0 and above.

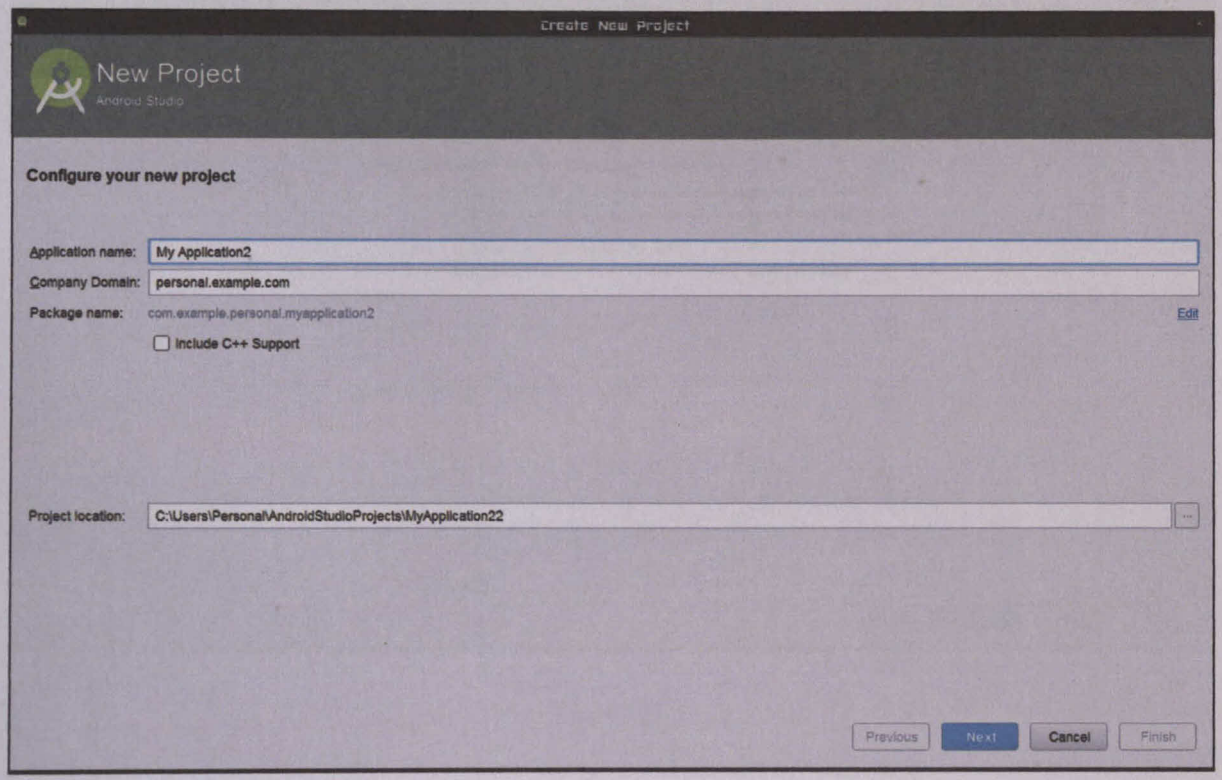
❖ PC Applications:

- Android SDK and platform tools.
- ANDROID STUDIO 2.2.3.
- An Emulator or Android device running Android 2.3 or higher.

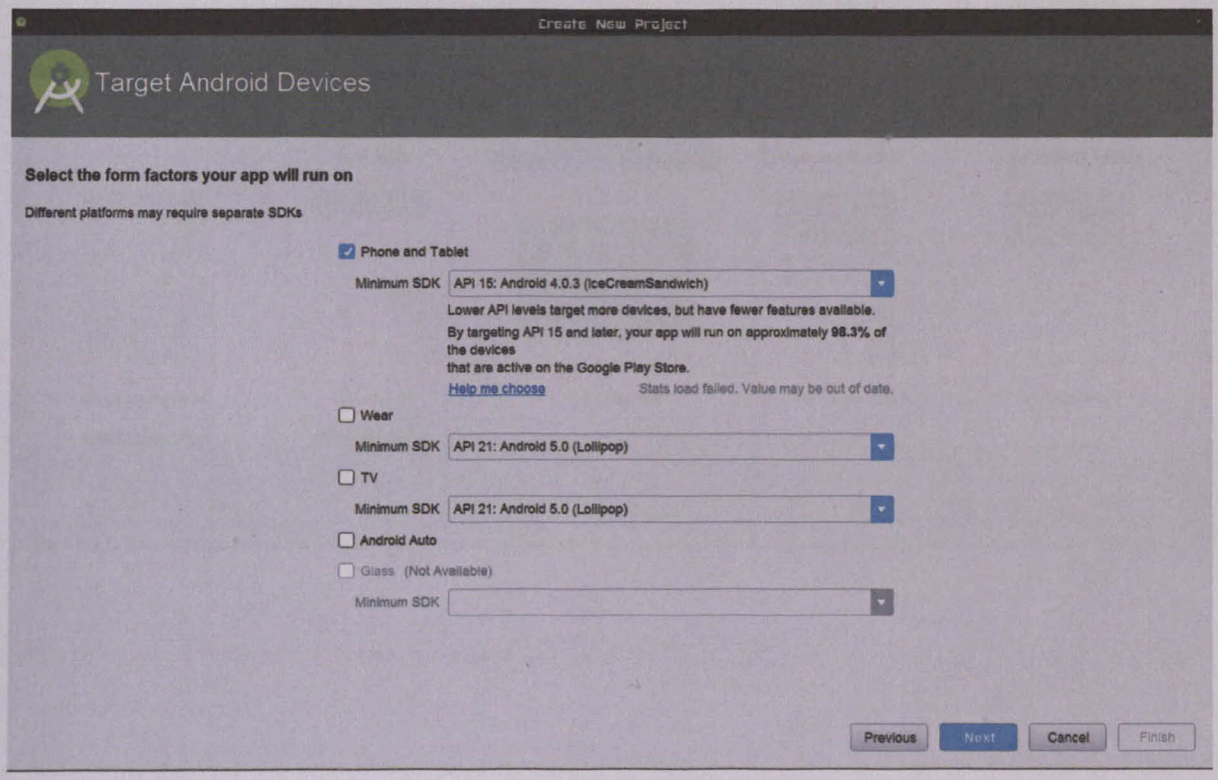
Templates

Create a new Android project

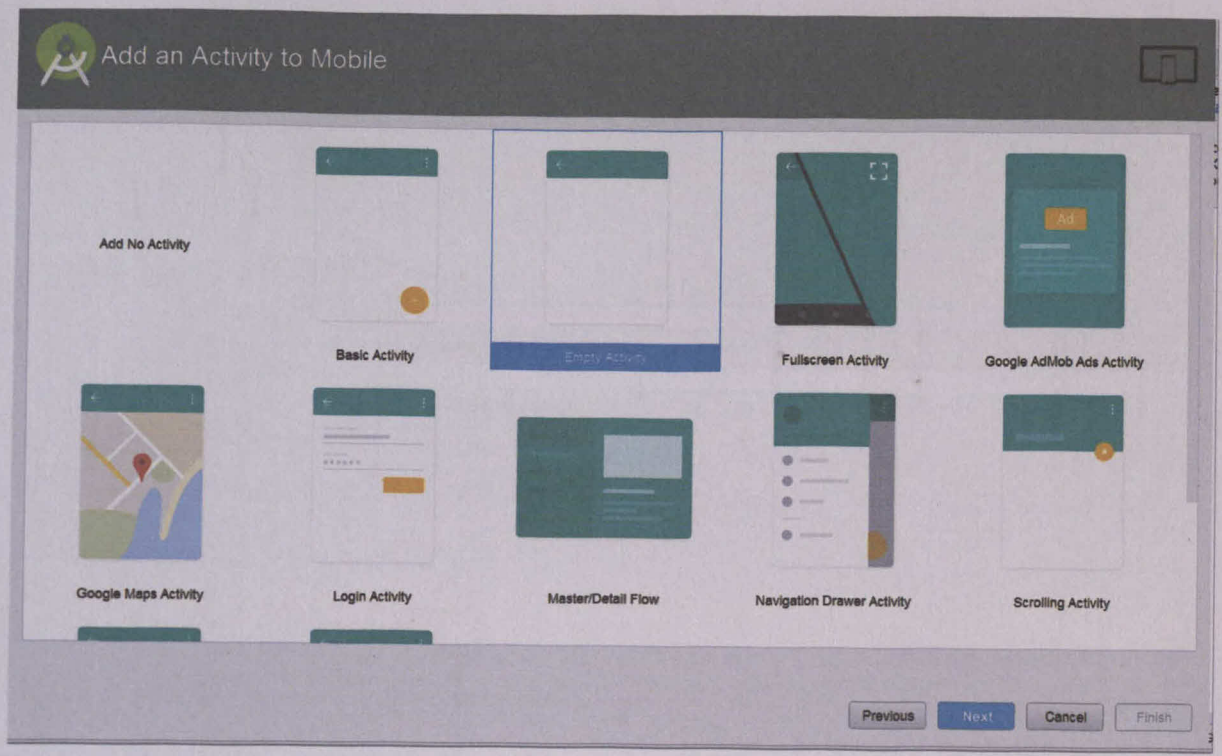
In Android Studio, go to File > New > Project and in the New Project dialog, expand Android folder to select Android Project. And here I m naming it as SmsApp.



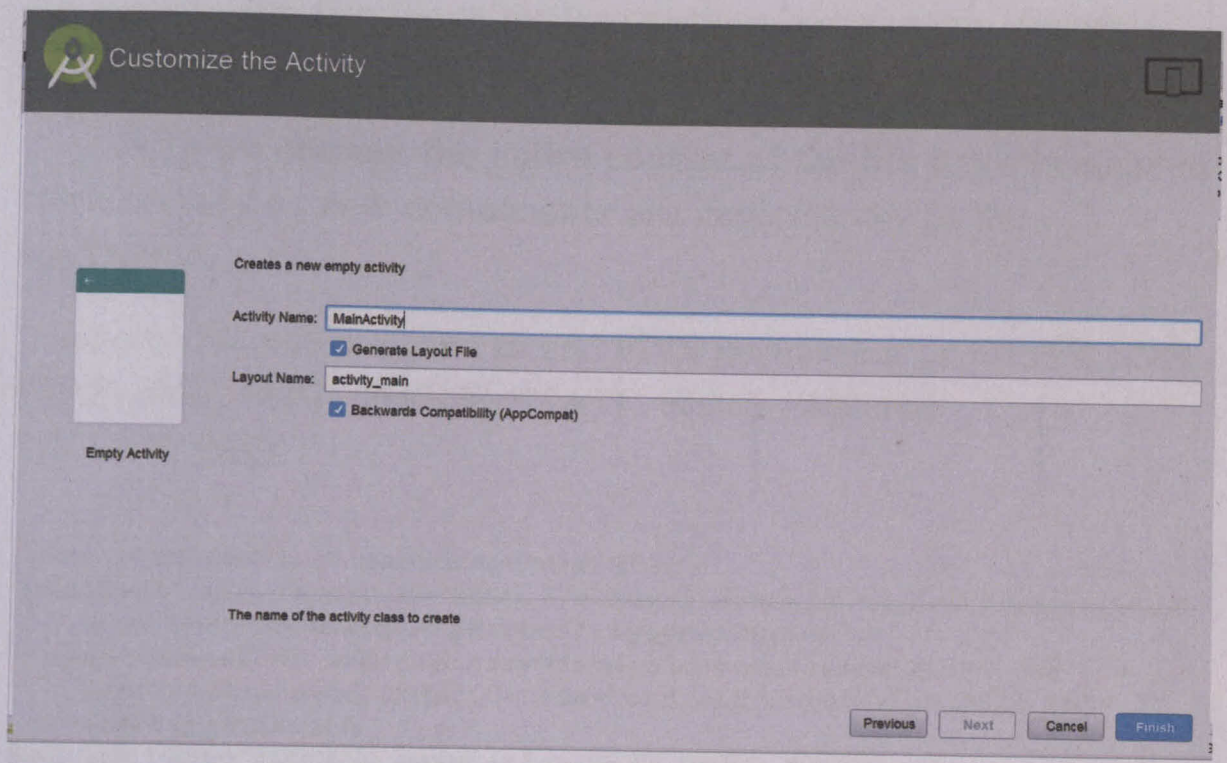
Now We Choose API and SDK



After that Choose Empty Activity



And Then We Click Finish



Android Manifest

The AndroidManifest.xml describes the Android application. We need to edit this file to declare various components, features, permissions, etc. used by the application.

Here we present the entire content of the file but it is updated incrementally as new components are implemented in the application.

Under this app we have to add FIVE permission i.e WRITE_SMS, READ_SMS, READ_PHONE_STATE, SEND_SMS and RECEIVE_SMS.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.personal.myapplication">
 <uses-permission android:name="android.permission.WRITE_SMS" />
    <uses-permission android:name="android.permission.READ_SMS" />
    <uses-permission
android:name="android.permission.READ_PHONE_STATE" />
    <uses-permission android:name="android.permission.RECEIVE_SMS"
/>
    <uses-permission android:name="android.permission.SEND_SMS" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme">
        <activity
            android:name=".MainActivity"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name=".ReceiveSmsActivity"
            android:label="@string/app_name"></activity>
        <activity
            android:name=".SendSmsActivity"
            android:label="@string/app_name"></activity>
        <receiver
```

```
android:name=".SmsBroadcastReceiver"

android:exported="true">
            <intent-filter android:priority="999">
                <action
android:name="android.provider.Telephony.SMS_RECEIVED" />
            </intent-filter>
        </receiver>
    </application>

</manifest>
```

As you can see all the components used in the application are Activities. We have declared Activity for each screen in the application.

ACTIVITY LAYOUTS

Here the layout defines the visual structure for a user interface, such as the UI for an activity or app widget.

Following are the activities created for different layout:

(a). Main Activity

Create an XML file for the MainActivity class in the project's res/layout folder and name it activity_main.xml

```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/abc">


    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Compose"
        android:textSize="20sp"
        android:id="@+id/btnCompose"
        android:textColor="@color/colorAccent"
        android:background="@color/colorPrimary"

        android:onClick="goToCompose"
        android:layout_below="@+id/btnInbox"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginTop="19dp"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="@color/colorAccent"
        android:text="Inbox"
        android:id="@+id/btnInbox"
        android:textSize="20sp"
        android:background="@color/colorPrimary"
        android:onClick="goToInbox"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
```

```
android:layout_alignParentStart="true"
android:layout_marginTop="19dp"
android:layout_alignParentRight="true"
android:layout_alignParentEnd="true" />


</RelativeLayout>
```

(b). Received SMS Activity

Then, create an XML file for the activity_receive_sms class in the project's res/layout folder and name it as activity_receive_sms.xml

```
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/abc">


    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Compose"
        android:textSize="20sp"
        android:id="@+id/btnCompose"
        android:textColor="@color/colorAccent"
        android:background="@color/colorPrimary"

        android:onClick="goToCompose"
        android:layout_below="@+id/btnInbox"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginTop="19dp"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="@color/colorAccent"
        android:text="Inbox"
```

```
        android:id="@+id/btnInbox"
         android:textSize="20sp"
        android:background="@color/colorPrimary"


        android:onClick="goToInbox"
        android:layout_alignParentTop="true"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginTop="19dp"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true" />

</RelativeLayout>
```

(c). Send SMS Activity

Again, create an XML file for the activity_send_sms class in the project's res/layout folder and name it as activity_send_sms.xml

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:background="@drawable/abc"
    android:weightSum="1">


    <TextView
        android:id="@+id/textViewPhoneNo"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Please Type the number"
        android:layout_marginTop="20dp"
        android:textColor="@color/colorAccent"
        android:layout_weight="0.18" />

    <EditText
        android:id="@+id/editTextPhoneNo"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="phone"
        android:textColor="@color/colorAccent"/>

    <TextView
        android:id="@+id/textViewMessage"
        android:layout_width="wrap_content"
```

```
        android:layout_height="wrap_content"
        android:text="Please type your message"
        android:layout_marginTop="20dp"
        android:layout_weight="0.18"
        android:textColor="@color/colorAccent"/>

    <EditText
        android:id="@+id/editTextSMS"
        android:layout_width="match_parent"
        android:layout_height="280dp"
        android:ems="80"
        android:inputType="textMultiLine"
        android:textColor="@color/colorAccent"
        android:layout_weight="0.25" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="@color/colorAccent"
        android:ems="500"

        android:background="@color/colorPrimary"
        android:id="@+id/btnSendSMS"
        android:layout_marginBottom="20dp"
        android:text="Send it" />


</LinearLayout>
```

(d). Broadcast Receiver Activity

Lastly,we create an XML file for the activity_sms_broadcast_receiver class in the project's res/layout folder and name it as activity_sms_broadcast_receiver.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:id="@+id/activity_sms_broadcast_receiver"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"

tools:context="com.example.personal.myapplication.SmsBroadcastReceiv
er">

</RelativeLayout>
```

Implementing the Activity classes

An activity represents a single screen with a user interface just like a window or a frame of java. It is the subclass of ContextThemeWrapper class.

Following are the different Activity Classes:

(a) MainActivity.java

```
package com.example.personal.myapplication;

import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.ActionBarActivity;
import android.view.View;

public class MainActivity extends ActionBarActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    public void goToInbox(View view) {
        Intent intent = new Intent(MainActivity.this,
ReceiveSmsActivity.class);
        startActivity(intent);
    }

    public void goToCompose(View view) {
        Intent intent = new Intent(MainActivity.this,
SendSmsActivity.class);
        startActivity(intent);
    }

}
```

Now we have to create a class for Receiver sms activity

(b) Receiver Sms activity

```
package com.example.personal.myapplication;


import android.app.Activity;
import android.content.ContentResolver;
import android.content.Intent;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;

import java.text.SimpleDateFormat;
import java.util.ArrayList;
import java.util.Date;

public class ReceiveSmsActivity extends Activity implements
OnItemClickListener {

    private static ReceiveSmsActivity inst;
    ArrayList<String> smsMessagesList = new ArrayList<String>();
    ListView smsListView;
    ArrayAdapter arrayAdapter;

    public static ReceiveSmsActivity instance() {
        return inst;
    }

    @Override
    public void onStart() {
        super.onStart();
        inst = this;
    }

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_receive_sms);
        smsListView = (ListView) findViewById(R.id.SMSList);
        arrayAdapter = new ArrayAdapter<String>(this,
       android.R.layout.simple_list_item_1, smsMessagesList);
```

```
        smsListView.setAdapter(arrayAdapter);
        smsListView.setOnItemClickListener(this);

        refreshSmsInbox();
    }

    public void refreshSmsInbox() {
        ContentResolver contentResolver = getContentResolver();
        Cursor smsInboxCursor =
contentResolver.query(Uri.parse("content://sms/inbox"), null, null,
null, null);
        int indexBody = smsInboxCursor.getColumnIndex("body");
        int indexAddress = smsInboxCursor.getColumnIndex("address");
        long timeMillis = smsInboxCursor.getColumnIndex("date");
        Date date = new Date(timeMillis);
        SimpleDateFormat format = new SimpleDateFormat("dd/MM/yy");
        String dateText = format.format(date);

        if (indexBody < 0 || !smsInboxCursor.moveToFirst()) return;
        arrayAdapter.clear();
        do {
            String str = smsInboxCursor.getString(indexAddress) +"
at "+
                    "\n" + smsInboxCursor.getString(indexBody)
+dateText+ "\n";
            arrayAdapter.add(str);
        } while (smsInboxCursor.moveToNext());
    }

    public void updateList(final String smsMessage) {
        arrayAdapter.insert(smsMessage, 0);
        arrayAdapter.notifyDataSetChanged();
    }

    public void onItemClick(AdapterView<?> parent, View view, int
pos, long id) {
        try {
            String[] smsMessages =
smsMessagesList.get(pos).split("\n");
            String address = smsMessages[0];
            String smsMessage = "";
            for (int i = 1; i < smsMessages.length; ++i) {
                smsMessage += smsMessages[i];
            }

            String smsMessageStr = address + "\n";
            smsMessageStr += smsMessage;
            Toast.makeText(this, smsMessageStr,
Toast.LENGTH_SHORT).show();
        } catch (Exception e) {
            e.printStackTrace();
```

```
}
    }

    public void goToCompose(View view) {
        Intent intent = new Intent(ReceiveSmsActivity.this,
SendSmsActivity.class);
        startActivity(intent);
    }
}
```

Again, we create a class activity for

(c) SendSmsActivity

```
package com.example.personal.myapplication;

import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.telephony.SmsManager;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class SendSmsActivity extends Activity {

    Button sendSMSBtn;
    EditText toPhoneNumberET;
    EditText smsMessageET;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_send_sms);
        sendSMSBtn = (Button) findViewById(R.id.btnSendSMS);
        toPhoneNumberET = (EditText)
findViewById(R.id.editTextPhoneNo);
        smsMessageET = (EditText) findViewById(R.id.editTextSMS);
        sendSMSBtn.setOnClickListener(new View.OnClickListener() {
            public void onClick(View view) {
                sendSMS();
            }
        });
    }

    protected void sendSMS() {
        String toPhoneNumber = toPhoneNumberET.getText().toString();
        String smsMessage = smsMessageET.getText().toString();
        try {
```

```
SmsManager smsManager = SmsManager.getDefault();
            smsManager.sendTextMessage(toPhoneNumber, null,
smsMessage, null, null);
            Toast.makeText(getApplicationContext(), "SMS sent.",
                    Toast.LENGTH_LONG).show();
        } catch (Exception e) {
            Toast.makeText(getApplicationContext(),
                    "Sending SMS failed.",
                    Toast.LENGTH_LONG).show();
            e.printStackTrace();
        }
    }

    public void goToInbox(View view) {
        Intent intent = new Intent(SendSmsActivity.this,
ReceiveSmsActivity.class);
        startActivity(intent);
    }
}
```

Lastly, we create a class for

(d) SmsBroadcastReceiver

```
package com.example.personal.myapplication;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.telephony.SmsMessage;
import android.widget.Toast;
import java.text.SimpleDateFormat;
import java.util.Date;

public class SmsBroadcastReceiver extends BroadcastReceiver {

    public static final String SMS_BUNDLE = "pdus";

    public void onReceive(Context context, Intent intent) {
        Bundle intentExtras = intent.getExtras();
        if (intentExtras != null) {
            Object[] sms = (Object[]) intentExtras.get(SMS_BUNDLE);
            String smsMessageStr = "";
            for (int i = 0; i < sms.length; ++i) {
                SmsMessage smsMessage =
SmsMessage.createFromPdu((byte[]) sms[i]);
```

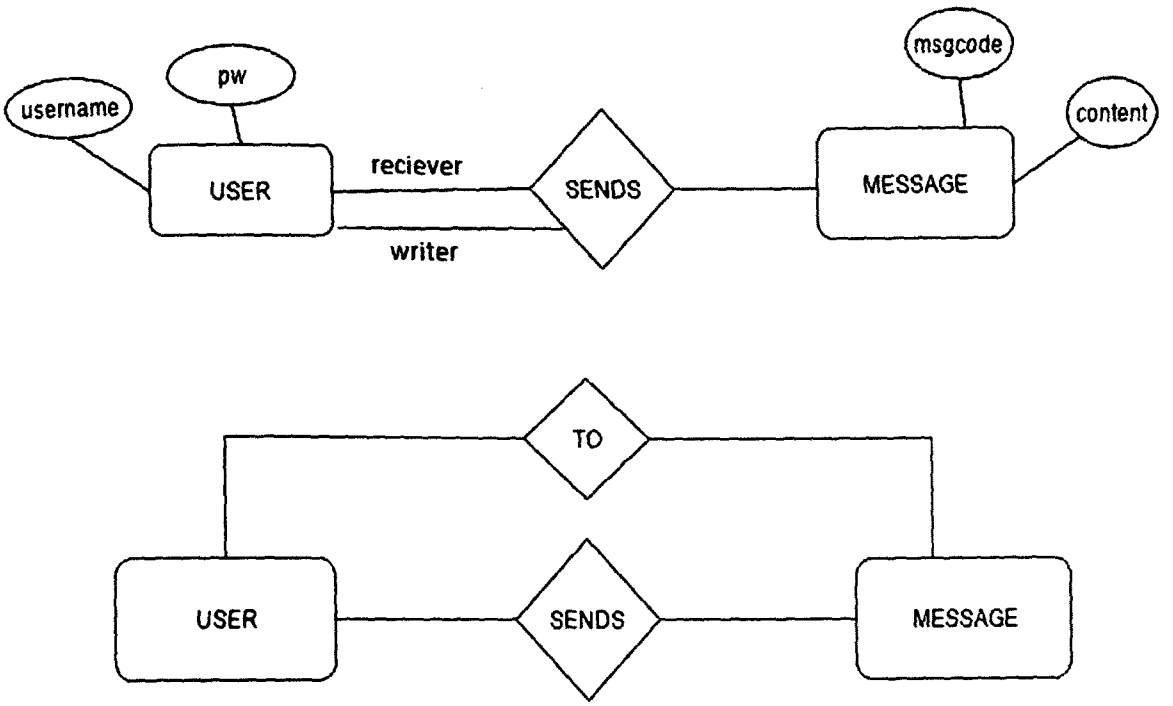
```
                String smsBody =
smsMessage.getMessageBody().toString();
                String address = smsMessage.getOriginatingAddress();
                long timeMillis = smsMessage.getTimestampMillis();

                Date date = new Date(timeMillis);
                SimpleDateFormat format = new
SimpleDateFormat("dd/MM/yy");
                String dateText = format.format(date);

                smsMessageStr += address +" at "+"\t"+ dateText +
"\n";
                smsMessageStr += smsBody + "\n";
            }
            Toast.makeText(context, smsMessageStr,
Toast.LENGTH_SHORT).show();

            //this will update the UI with message
            ReceiveSmsActivity inst = ReceiveSmsActivity.instance();
            inst.updateList(smsMessageStr);
        }
    }
}
```

ER DIAGRAM


msgcode
pw
content
username
USER
reciever
SENDS
MESSAGE
writer
TO
USER
SENDS
MESSAGE

Executing the application

So now its the time to execute the application, we will open the App.... and the Screenshot of our "**SMS APP**" is shown Below:

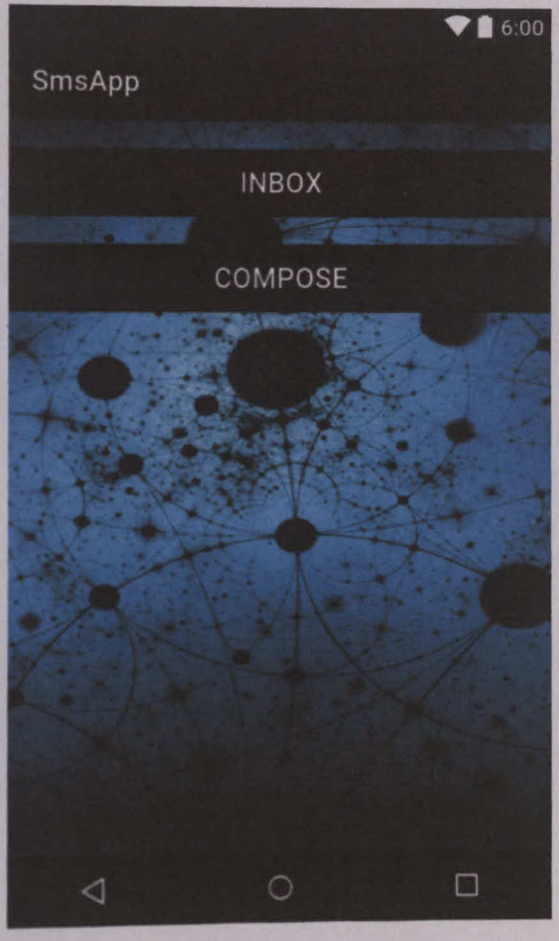
The most interesting feature of this custom made app is that we can not only send messages but as well as receive the message.

Here, the interface has two parts that is –

1.Inbox.

2.Compose

Below is the layout of the first interface:



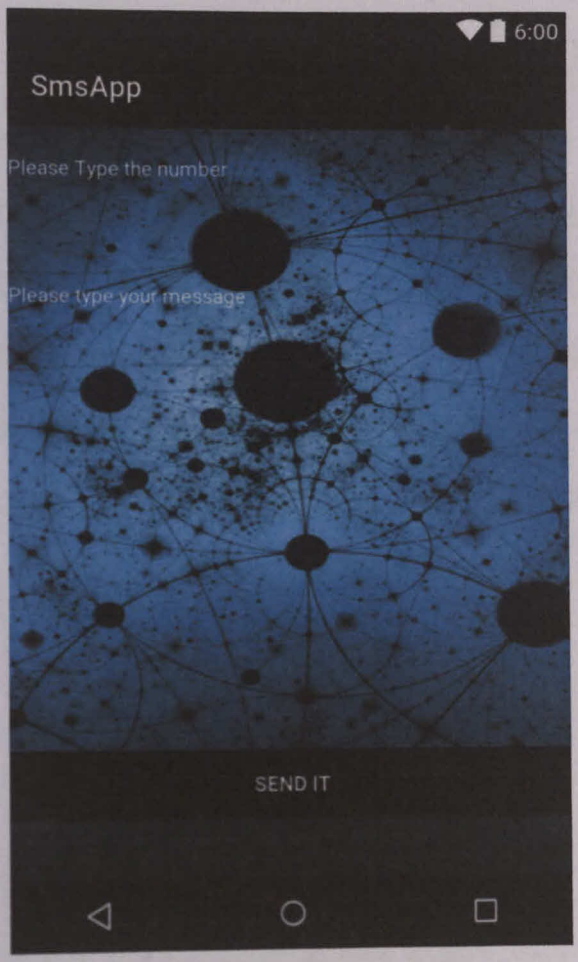
1.INBOX

In this part of the layout we can read the recieved messages.

2.COMPOSE

Now if we want to compose a message, we have to click the "**Compose**" button which is in the first interface.

The layout of the compose message is shown below, where the user can enter the recipient number and type the message below and click on "**Send It**" button.



CONCLUSION

As we have seen in today's world Android is the most advance operating system that most of the people used. So, even i learned that android is one of the best OS with so many application that had made for it along with various features.

So,for this reason i've choose to make this app which is a very simple and cool application. The main objective of this project is to develop a simple, nice and a fast messaging app with this we can even have our own Sms Application and we can even customize and add some more features as much as we can.