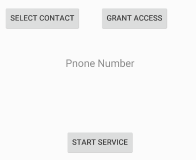
Caracteristicas

* Correr servicio al momento se encender el aparato **class** BootUpReceiver Instala servicio llamado Busqueda
* Al iniciar la aplicación verifica si servicio esta corriendo, sino lo activa (no es necesario)
* Seleccionar Contacto y traer numero y nombre MainActivity
* Envio de mensaje para solicitar coordenadas MainActivity
* El servicio verifica que solicita coordenadas ServiceCommunicator BroadcastReceiver
* Obtiene las coordenadas
* El servicio verifica que recibe coordenadas y presenta mapa ServiceCommunicator BroadcastReceiver

**

*<?***xml version="1.0" encoding="utf-8"***?>*<**RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 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"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"  
 tools:context="com.example.tdal6392.busqueda.MainActivity"  
 tools:showIn="@layout/activity\_main"**>  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text=""  
 android:id="@+id/textView"  
 android:layout\_centerHorizontal="true"** />  
  
 <**Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Select Contact"  
 android:id="@+id/btnContact"  
 android:layout\_alignTop="@+id/btnAllow"  
 android:layout\_alignParentStart="true"** />  
  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="19sp"  
 android:text="Pnone Number"  
 android:id="@+id/NumTel"  
 android:layout\_marginTop="47dp"  
 android:layout\_below="@+id/btnContact"  
 android:layout\_centerHorizontal="true"** />  
  
 <**Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Start Service"  
 android:id="@+id/btnStat"  
 android:layout\_centerVertical="true"  
 android:layout\_centerHorizontal="true"** />  
  
 <**Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Grant Access"  
 android:id="@+id/btnAllow"  
 android:layout\_below="@+id/textView"  
 android:layout\_toEndOf="@+id/textView"** />

<**ProgressBar  
 style="?android:attr/progressBarStyleHorizontal"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/progressBar"  
 android:layout\_alignTop="@+id/btnStat"  
 android:layout\_centerHorizontal="true"** />

</**RelativeLayout**>

Al momento de iniciar cellular, inicie servicio

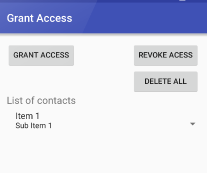
**package** com.example.tdal6392.busqueda;  
  
**import** android.content.BroadcastReceiver;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.widget.Toast;  
  
**public class** BootUpReceiver **extends** BroadcastReceiver {  
  
 @Override  
 **public void** onReceive(Context context, Intent intent) {  
  
**try** {  
 Class aClass = Class.*forName*(**"com.example.tdal6392.busqueda.ServiceCommunicator"**);  
 Intent serviceIntent = **new** Intent(context, aClass);  
 *//serviceIntent.putExtra("interval", 1 \* 1000L);//every second* **context.startService(serviceIntent);**  
 } **catch** (ClassNotFoundException e) {  
 Toast.*makeText*(context, e.toString(), Toast.***LENGTH\_LONG***).show();  
 *//e.printStackTrace();* }  
  
 ***// START ACTIVITY ON BOOT*** *//Intent i = new Intent(context, MainActivity.class);  
 //i.addFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK);  
 //i.putExtra("BOOT",1);  
 //context.startActivity(i);* }  
}

**Main activity**

**import** android.accounts.Account;  
**import** android.accounts.AccountManager;  
**import** android.app.ActivityManager;  
**import** android.content.BroadcastReceiver;  
**import** android.content.ComponentName;  
**import** android.content.ContentResolver;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.content.IntentFilter;  
**import** android.content.ServiceConnection;  
**import** android.database.ContentObserver;  
**import** android.database.Cursor;  
**import** android.net.Uri;  
**import** android.os.Bundle;  
**import** android.os.CountDownTimer;  
**import** android.os.IBinder;  
**import** android.provider.ContactsContract;  
**import** android.support.v7.app.AlertDialog;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.telephony.SmsManager;  
**import** android.util.Log;  
**import** android.view.View;  
**import** android.widget.Button;  
**import** android.widget.ProgressBar;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
**import** java.util.Calendar;  
**import** java.util.logging.Handler;

**public class** MainActivity **extends** AppCompatActivity {  
  
 **static final int *PICK\_CONTACT\_REQUEST*** = 1;  
 **static final** Boolean ***PRUEBA*** = **false**;  
 **private** ProgressBar **ProgBar**;  
 **private static final** String ***LOG\_TAG*** = **"debugger"**;  
  
 ServiceCommunicator **mService**;  
 **boolean mBound** = **false**;  
  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
  
 startService(**new** Intent(**this**, NotificationService.**class**));  
  
 **this**.setTitle(**"Locate Device"**);  
 registerReceiver(**FxShowMsg**, **new** IntentFilter(**"MENSAJE"**));  
  
 **ProgBar**= (ProgressBar) findViewById(R.id.***progressBar***);  
 **ProgBar**.setVisibility(ProgressBar.***INVISIBLE***);  
  
 **int** year = Calendar.*getInstance*().get(Calendar.***YEAR***);  
 **if** (year >2016) **return**;  
 CheckIfServiceRunning();  
  
 HandleClick handleClick = **new** HandleClick();  
 findViewById(R.id.***btnContact***).setOnClickListener(handleClick);  
 findViewById(R.id.***btnStat***).setOnClickListener(handleClick);  
 findViewById(R.id.***btnAllow***).setOnClickListener(handleClick);  
 }  
  
  
  
 **public** String GetAccount()  
 {  
 String cuenta = **""**;  
 AccountManager mgr = AccountManager.*get*(getApplicationContext());  
 Account[] accts = mgr.getAccounts();  
 **final int** count = accts.**length**;  
 Account acct = **null**;  
  
 **for** (**int** i = 0; i < count; i++) {  
 acct = accts[i];  
 **if** (acct.**type**.equals(**"com.google"**)) {  
 cuenta = acct.**name**;  
 **break**;  
 }  
 }  
 **return** cuenta;  
 }  
*// int conteo = 0;  
  
 // WHEN UPDATING MESSAGE FROM SERVICE* **private** BroadcastReceiver **FxShowMsg** = **new** BroadcastReceiver() {  
  
 @Override  
 **public void** onReceive(Context context, Intent intent) {  
 TextView msg = (TextView) findViewById(R.id.***NumTel***);  
 msg.setText(intent.getExtras().getString(**"<KEY>"**));  
 }  
 };  
  
  
 @Override  
 **protected void** onStart() {  
 **super**.onStart();  
  
 *// Bind to LocalService* Intent intent = **new** Intent(**this**, ServiceCommunicator.**class**);  
 bindService(intent, **mConnection**, Context.***BIND\_AUTO\_CREATE***);  
 }  
  
 **public void** CheckIfServiceRunning(){  
  
 TextView chequeo = (TextView) findViewById(R.id.***NumTel***);  
 Boolean hallado = **false**;  
 ActivityManager manager = (ActivityManager) getSystemService(***ACTIVITY\_SERVICE***);  
  
 **for** (ActivityManager.RunningServiceInfo service : manager.getRunningServices(Integer.***MAX\_VALUE***))  
 {  
 **if** (**"com.example.tdal6392.busqueda.ServiceCommunicator"**.equals(service.**service**.getClassName()))  
 {  
 chequeo.setText(**""**);  
 Button btnStart = (Button) findViewById(R.id.***btnStat***);  
 *//btnStart.setVisibility(View.INVISIBLE);* hallado = **true**;  
 btnStart.setText(**"CHECK YOUR LOCATION"**);  
 }  
 }  
  
 **if** (!hallado)  
 {  
 chequeo.setText(**"Service not Running\nRestart device"**);  
 Button btn1 = (Button) findViewById(R.id.***btnContact***);  
 btn1.setEnabled(**false**);  
 *// StartService();* }  
 }  
  
 **private class** HandleClick **implements** View.OnClickListener {  
  
 **public void** onClick(View view) {  
 **switch**(view.getId()) {  
  
 **case** R.id.***btnAllow***:  
 Intent intent = **new** Intent(MainActivity.**this**, GrantAccess.**class**);  
 MainActivity.**this**.startActivity(intent);  
 **break**;  
  
 **case** R.id.***btnStat***:  
 Button btnStart = (Button) findViewById(R.id.***btnStat***);  
 String opcion = btnStart.getText().toString().toUpperCase();  
  
 **if** (opcion == **"CHECK YOUR LOCATION"**) {  
 *// getLocation();* TextView chequeo = (TextView) findViewById(R.id.***NumTel***);  
 String Coordenadas = **mService**.GetCoordenada();  
  
 chequeo.setText(Coordenadas);  
  
 String localidad = **"http://maps.google.com/maps?q=loc:"** + Coordenadas;  
 *//Toast.makeText(ServiceCommunicator.this, localidad, Toast.LENGTH\_LONG).show();* Uri uri = Uri.*parse*(localidad);  
 Intent intent2 = **new** Intent(Intent.***ACTION\_VIEW***, uri);  
 intent2.setFlags(Intent.***FLAG\_ACTIVITY\_NEW\_TASK***);  
 startActivity(intent2);  
 }  
 **else** StartService();  
 **break**;  
  
 **case** R.id.***btnContact***:  
 **if** (***PRUEBA***) {  
 EnviarSMS(**"70020015"**);  
 }  
 **else** {  
 Intent pickContactIntent = **new** Intent(Intent.***ACTION\_PICK***, Uri.*parse*(**"content://contacts"**));  
 pickContactIntent.setType(ContactsContract.CommonDataKinds.Phone.***CONTENT\_TYPE***);  
 *// Show user only contacts w/ phone numbers* startActivityForResult(pickContactIntent, ***PICK\_CONTACT\_REQUEST***);  
 }  
 **break**;  
 }  
 }  
 }  
  
 **private void** StartService(){  
  
 String Mensaje = **""**;  
 TextView chequeo = (TextView) findViewById(R.id.***NumTel***);  
 chequeo.setText(**""**);  
  
 startService(**new** Intent(MainActivity.**this**, ServiceCommunicator.**class**));  
  
 chequeo.setText(**"Service Started"**);  
 Button btnStart = (Button) findViewById(R.id.***btnStat***);  
 *//btnStart.setVisibility(View.INVISIBLE);* btnStart.setText(**"CHECK YOUR LOCATION"**);  
  
 Button btnContacto = (Button) findViewById(R.id.***btnContact***);  
 btnContacto.setEnabled(**true**);  
 }  
  
  
 **private void** EnviarSMS(String Telefono) {  
 **try** {  
 SmsManager smsManager = SmsManager.*getDefault*();  
 smsManager.sendTextMessage(Telefono, **null**, **"GPS"**, **null**, **null**);  
 Mensaje(**"Sent"**);  
 }  
 **catch** (Exception e) {  
 Mensaje(**"Error, please try again"**);  
 e.printStackTrace();  
 }  
 }  
  
 **private void** Mensaje(String msg)  
 {  
 Toast.*makeText*(getApplicationContext(), msg, Toast.***LENGTH\_LONG***).show();  
 }  
  
 @Override  
 **protected void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  
  
 **if** (requestCode == ***PICK\_CONTACT\_REQUEST***) {  
 **if** (resultCode == ***RESULT\_OK***) {  
  
 *// Get the URI that points to the selected contact* Uri contactUri = data.getData();  
 String[] projection = {ContactsContract.CommonDataKinds.Phone.***NUMBER***};  
  
 Button btnStart = (Button) findViewById(R.id.***btnStat***);  
 Button btnContact = (Button) findViewById(R.id.***btnContact***);  
 Button btnContact2 = (Button) findViewById(R.id.***btnAllow***);  
  
 btnStart.setVisibility(View.***INVISIBLE***);  
 btnContact.setEnabled(**false**);  
 btnContact2.setEnabled(**false**);  
  
 **try** {  
 Cursor cursor = getContentResolver().query(contactUri, projection, **null**, **null**, **null**);  
 cursor.moveToFirst();  
  
 **int** phoneIndex = cursor.getColumnIndex(ContactsContract.CommonDataKinds.Phone.***NUMBER***);  
 String numTele = cursor.getString(phoneIndex);  
 String nombre = *getContactName*(getApplicationContext(), numTele);  
  
 *//numTele = numTele.replace("+503","");* TextView nombre2 = (TextView)findViewById(R.id.***NumTel***);  
 nombre2.setText(**"Requesting Location to \n"** +numTele+**" - "**+nombre+**"\n\nPlease Wait..."**);  
  
 EnviarSMS(numTele);  
  
 **ProgBar**.setVisibility(View.***VISIBLE***);  
 **ProgBar**.setMax(100);  
 **ProgBar**.setProgress(0);  
  
 *//final CountDownTimer timer =* **new** CountDownTimer(150000, 5000) {  
  
 **int conteo** = 0;  
  
 **public void** onTick(**long** millisUntilFinished) {  
  
 **conteo** = **conteo** + 5; *// each 2,000 is 2,000/ 40,000 = 1/15 = 7%* **if** (**conteo**>100) **conteo** = 0;  
 **ProgBar**.setProgress(**conteo**);  
  
 **if** (**conteo** > 20) {  
 TextView tv = (TextView) findViewById(R.id.***NumTel***);  
 String msg1 = tv.getText().toString();  
 **if** (msg1.indexOf(**"not Authorized"**) > 0) {  
 **ProgBar**.setVisibility(View.***INVISIBLE***);  
 String mensaje = **"Request your contact to allow you to view the location"**;  
 **new** AlertDialog.Builder(getApplicationContext()).setTitle(**"Not Authorized"**).setMessage(mensaje).setNeutralButton(**"Close"**, **null**).show();  
 *// Count.cancel();* }  
 }  
 *// mTextField.setText("seconds remaining: " + millisUntilFinished / 1000);* }  
  
 **public void** onFinish() {  
 **ProgBar**.setVisibility(View.***INVISIBLE***);  
  
 TextView nombre2 = (TextView)findViewById(R.id.***NumTel***);  
 nombre2.setText(**" "**);  
  
 Button btnStart = (Button) findViewById(R.id.***btnStat***);  
 Button btnContact = (Button) findViewById(R.id.***btnContact***);  
 Button btnContact2 = (Button) findViewById(R.id.***btnAllow***);  
  
 btnStart.setVisibility(View.***VISIBLE***);  
 btnContact.setEnabled(**true**);  
 btnContact2.setEnabled(**true**);  
 }  
 }.start();  
 }  
 **catch** (Exception e) {  
 Toast.*makeText*(getApplicationContext(), e.toString(),  
 Toast.***LENGTH\_LONG***).show();  
 }  
 }  
 }  
 }  
  
  
 **public static** String getContactName(Context context, String phoneNumber) {  
 ContentResolver cr = context.getContentResolver();  
 Uri uri = Uri.*withAppendedPath*(ContactsContract.PhoneLookup.***CONTENT\_FILTER\_URI***, Uri.*encode*(phoneNumber));  
 Cursor cursor = cr.query(uri, **new** String[]{ContactsContract.PhoneLookup.***DISPLAY\_NAME***}, **null**, **null**, **null**);  
 **if** (cursor == **null**) {  
 **return null**;  
 }  
 String contactName = **null**;  
 **if**(cursor.moveToFirst()) {  
 contactName = cursor.getString(cursor.getColumnIndex(ContactsContract.PhoneLookup.***DISPLAY\_NAME***));  
 }  
  
 **if**(cursor != **null** && !cursor.isClosed()) {  
 cursor.close();  
 }  
  
 **return** contactName;  
 }  
  
 @Override  
 **protected void** onDestroy() {  
 **super**.onDestroy();  
 unregisterReceiver(**FxShowMsg**);  
 }  
  
}

**Grant Access**

****

*<?***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: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.tdal6392.busqueda.GrantAccess"**>  
  
  
  
 <**Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Grant Access"  
 android:id="@+id/btnADD"  
 android:layout\_alignParentStart="true"** />  
  
 <**Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Revoke Acess"  
 android:id="@+id/btnRevoke"  
 android:layout\_alignBottom="@+id/btnADD"  
 android:layout\_alignParentEnd="true"** />  
  
 <**Button  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Delete All"  
 android:id="@+id/btnDelete"  
 android:layout\_below="@+id/btnRevoke"  
 android:layout\_alignStart="@+id/btnRevoke"  
 android:layout\_alignEnd="@+id/btnRevoke"** />  
  
 <**TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:textSize="18sp"  
 android:text="List of contacts"  
 android:id="@+id/msg2"  
 android:layout\_below="@+id/btnDelete"  
 android:layout\_alignParentStart="true"** />  
  
 <**Spinner  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:id="@+id/spinner"  
 android:layout\_below="@+id/msg2"  
 android:layout\_alignParentEnd="true"** />  
  
</**RelativeLayout**>

**import** android.content.ContentResolver;  
**import** android.content.Context;  
**import** android.content.DialogInterface;  
**import** android.content.Intent;  
**import** android.database.Cursor;  
**import** android.net.Uri;  
**import** android.provider.ContactsContract;  
**import** android.support.v7.app.AlertDialog;  
**import** android.support.v7.app.AppCompatActivity;  
**import** android.os.Bundle;  
**import** android.telephony.TelephonyManager;  
**import** android.view.View;  
**import** android.widget.ArrayAdapter;  
**import** android.widget.Spinner;  
**import** android.widget.Toast;  
**import** java.io.BufferedReader;  
**import** java.io.FileOutputStream;  
**import** java.io.IOException;  
**import** java.io.InputStreamReader;  
**import** java.util.ArrayList;  
**import** java.util.Locale;  
  
**public class** GrantAccess **extends** AppCompatActivity {  
  
 **static final int *PICK\_CONTACT\_REQUEST*** = 1;  
 String **FILE\_NAME** = **"ALLOW.txt"**;  
  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_grant\_access***);  
 LoadSpinner(**""**,**""**,0);  
  
 HandleClick handleClick = **new** HandleClick();  
 findViewById(R.id.***btnADD***).setOnClickListener(handleClick);  
 findViewById(R.id.***btnRevoke***).setOnClickListener(handleClick);  
 findViewById(R.id.***btnDelete***).setOnClickListener(handleClick);  
 }  
  
  
 **private class** HandleClick **implements** View.OnClickListener {  
  
 **public void** onClick(View view) {  
 **switch** (view.getId()) {  
  
 **case** R.id.***btnDelete***:  
  
 **new** AlertDialog.Builder(GrantAccess.**this**)  
 .setMessage(**"Are you Sure to Delete Allt?"**)  
 .setCancelable(**false**)  
 .setPositiveButton(**"Yes"**, **new** DialogInterface.OnClickListener() {  
 **public void** onClick(DialogInterface dialog, **int** id) {  
  
 Spinner spin1 = (Spinner) findViewById(R.id.***spinner***);  
 spin1.setAdapter(**null**);  
 WriteFile(**""**);  
 }  
 })  
 .setNegativeButton(**"No"**, **null**)  
 .show();  
 **break**;  
  
 **case** R.id.***btnRevoke***:  
  
 Spinner spinner = (Spinner)findViewById(R.id.***spinner***);  
 **final** String text = spinner.getSelectedItem().toString();  
  
 **new** AlertDialog.Builder(GrantAccess.**this**)  
 .setMessage(**"Are you Sure to Delete "**+text+**"?"**)  
 .setCancelable(**false**)  
 .setPositiveButton(**"Yes"**, **new** DialogInterface.OnClickListener() {  
 **public void** onClick(DialogInterface dialog, **int** id) {  
  
 **int** pos1 = text.indexOf(**" "**);  
 String text3 = text.substring(0,pos1);  
 text3 = text3.replace(**" "**,**""**);  
 *//Toast.makeText(getApplicationContext(), "Revoking "+ text, Toast.LENGTH\_LONG).show();* LoadSpinner(text3,text,2);  
  
 }  
 })  
 .setNegativeButton(**"No"**, **null**)  
 .show();  
  
 **break**;  
  
 **case** R.id.***btnADD***:  
 Intent pickContactIntent = **new** Intent(Intent.***ACTION\_PICK***, Uri.*parse*(**"content://contacts"**));  
 pickContactIntent.setType(ContactsContract.CommonDataKinds.Phone.***CONTENT\_TYPE***);  
 *// Show user only contacts w/ phone numbers* startActivityForResult(pickContactIntent, ***PICK\_CONTACT\_REQUEST***);  
 **break**;  
 }  
 }  
 }  
  
  
 @Override  
 **protected void** onActivityResult(**int** requestCode, **int** resultCode, Intent data) {  
  
 **if** (requestCode == ***PICK\_CONTACT\_REQUEST***) {  
 **if** (resultCode == ***RESULT\_OK***) {  
  
 *// Get the URI that points to the selected contact* Uri contactUri = data.getData();  
 String[] projection = {ContactsContract.CommonDataKinds.Phone.***NUMBER***};  
  
 **try** {  
 Cursor cursor = getContentResolver().query(contactUri, projection, **null**, **null**, **null**);  
 cursor.moveToFirst();  
  
 **int** phoneIndex = cursor.getColumnIndex(ContactsContract.CommonDataKinds.Phone.***NUMBER***);  
  
 String numTele = cursor.getString(phoneIndex);  
 **if** (numTele!=**""** && numTele!=**null**) {  
  
 *//numTele = numTele.replace(" ", "");  
 //numTele = numTele.replace("+503", "");* String numTele2 = **" "**+ numTele;  
 **if** (numTele2.indexOf(**"+"**)<1) {  
 numTele = **"+"**+GetCountryZipCode()+numTele;  
 }  
 numTele = numTele.replace(**" "**, **""**);  
 numTele = numTele.replace(**" "**, **""**);  
 *//Toast.makeText(getApplicationContext(), numTele, Toast.LENGTH\_LONG).show();* LoadSpinner(numTele, **""**,1);  
 }  
  
 } **catch** (Exception e) {  
 Toast.*makeText*(getApplicationContext(), e.toString(),  
 Toast.***LENGTH\_LONG***).show();  
 }  
 }  
 }  
 }  
  
  
 **public** String GetCountryZipCode(){  
 String CountryID=**""**;  
 String CountryZipCode=**""**;  
  
 TelephonyManager manager = (TelephonyManager) **this**.getSystemService(Context.***TELEPHONY\_SERVICE***);  
 *//getNetworkCountryIso* CountryID= manager.getSimCountryIso().toUpperCase();  
 String[] rl=**this**.getResources().getStringArray(R.array.***CountryCodes***);  
 **for**(**int** i=0;i<rl.**length**;i++){  
 String[] g=rl[i].split(**","**);  
 **if**(g[1].trim().equals(CountryID.trim())){  
 CountryZipCode=g[0];  
 **break**;  
 }  
 }  
 **return** CountryZipCode;  
 }  
  
 *// GET CONTACT NAME  
 //\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\** **public static** String getContactName(Context context, String phoneNumber) {  
 ContentResolver cr = context.getContentResolver();  
 Uri uri = Uri.*withAppendedPath*(ContactsContract.PhoneLookup.***CONTENT\_FILTER\_URI***, Uri.*encode*(phoneNumber));  
 Cursor cursor = cr.query(uri, **new** String[]{ContactsContract.PhoneLookup.***DISPLAY\_NAME***}, **null**, **null**, **null**);  
 **if** (cursor == **null**) {  
 **return ""**;  
 }  
 String contactName = **null**;  
 **if** (cursor.moveToFirst()) {  
 contactName = cursor.getString(cursor.getColumnIndex(ContactsContract.PhoneLookup.***DISPLAY\_NAME***));  
 }  
  
 **if** (cursor != **null** && !cursor.isClosed()) {  
 cursor.close();  
 }  
  
 **if** (contactName == **null**) contactName = **""**;  
 **return** contactName;  
 }  
  
  
 *// READ FILE  
 //\*\*\*\*\*\*\*\*\*\*\** **public void** LoadSpinner(String NumTel, String Contacto, **int** opcion) {  
  
 *// 0 - READ  
 // 1 - ADD  
 // 2 - REVOKE* ArrayList<String> telefonos = **new** ArrayList<String>();  
 String Lista = **""**;  
 String Lista2 = **""**;  
  
 **try** {  
 BufferedReader bReader = **new** BufferedReader(**new** InputStreamReader(openFileInput(**FILE\_NAME**)));  
 String line;  
 StringBuffer text = **new** StringBuffer();  
 **while** ((line = bReader.readLine()) != **null**) {  
  
 Lista2 = Lista2+line+**"-"**;  
  
 Boolean OK = **true**;  
 **if** (line.equals(NumTel)) {  
  
 *// REVOKE ONE* **if** (opcion ==2)  
 {  
 Toast.*makeText*(getApplicationContext(), **"Revoking "**+Contacto, Toast.***LENGTH\_LONG***).show();  
 OK = **false**;  
 }  
  
 *// ADD NEW ONE* **if** (opcion ==1) {  
 Toast.*makeText*(getApplicationContext(), **"Contact is Already in List"**, Toast.***LENGTH\_LONG***).show();  
 opcion = 0; *// don't insert a new one, just load again spinner* }  
 }  
  
 **if** (OK)  
 {  
 Lista = Lista + line + **"\n"**;  
 String Referencia1 = line + **' '** + *getContactName*(getApplicationContext(), line);  
 telefonos.add(Referencia1.toString());  
 }  
 }  
  
 **if** (opcion == 1) {  
 Lista = Lista + NumTel+**"\n"**;  
 String Referencia1 = NumTel + **' '** + *getContactName*(getApplicationContext(), NumTel);  
 telefonos.add(Referencia1.toString());  
 Toast.*makeText*(getApplicationContext(), **"Added "**+ Referencia1, Toast.***LENGTH\_LONG***).show();  
 }  
  
 } **catch** (IOException e) {  
 e.printStackTrace();  
 *// FILE DOESNT EXIST  
 //Toast.makeText(getBaseContext(), e.toString(), Toast.LENGTH\_LONG).show();* }  
  
 **if** (Lista != **""**) {  
 *//Toast.makeText(getBaseContext(), Lista2.toString(), Toast.LENGTH\_LONG).show();* ArrayAdapter<String> tel2 =  
 **new** ArrayAdapter<String>(**this**, android.R.layout.***simple\_spinner\_item***, telefonos);  
  
 tel2.notifyDataSetChanged();  
  
 Spinner spin1 = (Spinner) findViewById(R.id.***spinner***);  
 spin1.setAdapter(tel2);  
 }  
 **if** (opcion !=0) WriteFile(Lista);  
 }  
  
  
 *// WRITE FILE  
 //\*\*\*\*\*\*\*\*\*\*\** **public void** WriteFile(String Lista) {  
 **try** {  
 FileOutputStream fos = openFileOutput(**FILE\_NAME**, Context.***MODE\_PRIVATE***);  
 fos.write(Lista.toString().getBytes());  
 fos.close();  
 } **catch** (Exception e) {  
 e.printStackTrace();  
 }  
 }  
}

**Servicio**

**package** com.example.tdal6392.busqueda;

**import** android.Manifest;  
**import** android.app.Activity;  
**import** android.app.PendingIntent;  
**import** android.app.Service;  
**import** android.content.BroadcastReceiver;  
**import** android.content.Context;  
**import** android.content.Intent;  
**import** android.content.IntentFilter;  
**import** android.content.pm.PackageManager;  
**import** android.media.AudioManager;  
**import** android.media.ToneGenerator;  
**import** android.net.Uri;  
**import** android.os.Binder;  
**import** android.os.IBinder;  
**import** android.provider.Telephony;  
**import** android.support.annotation.Nullable;  
**import** android.support.v4.app.ActivityCompat;  
**import** android.support.v7.app.AlertDialog;  
**import** android.telephony.SmsManager;  
**import** android.telephony.SmsMessage;  
**import** android.location.Location;  
**import** android.location.LocationManager;  
**import** android.widget.TextView;  
**import** android.widget.Toast;  
  
**import** java.io.BufferedReader;  
**import** java.io.InputStreamReader;

**public class** ServiceCommunicator **extends** Service {  
  
 **private final** IBinder **mBinder** = **new** LocalBinder();  
 **public class** LocalBinder **extends** Binder {  
  
 ServiceCommunicator getService() {  
 *// Return this instance of LocalService so clients can call public methods* **return** ServiceCommunicator.**this**;  
 }  
 }  
  
  
 **private final** BroadcastReceiver **receiver** = **new** BroadcastReceiver() {  
  
 @Override  
 **public void** onReceive(Context context, Intent intent) {  
 String action = intent.getAction();  
  
 **if** (!action.equals(**"android.provider.Telephony.SMS\_RECEIVED"**)) {  
 **return**;  
 }  
  
 **for** (SmsMessage sms1 : Telephony.Sms.Intents.*getMessagesFromIntent*(intent)) {  
  
 String NumFrom = sms1.getOriginatingAddress();  
 String msgBody = sms1.getMessageBody();  
  
 String ss = sms1.getDisplayOriginatingAddress();  
 String ff =sms1.getDisplayMessageBody();  
  
 *// Toast.makeText(getApplicationContext(),ss + " "+ ff+ " "+NumFrom,Toast.LENGTH\_LONG).show();* **if** (msgBody.contains(**"BEEP"**))  
 {  
 ToneGenerator toneGen1 = **new** ToneGenerator(AudioManager.***STREAM\_MUSIC***, 100);  
 toneGen1.startTone(ToneGenerator.***TONE\_CDMA\_PIP***,2000); *// duration in ms* **return**;  
 }  
  
 **if**(!msgBody.contains(**"GPS"**)){  
 **return**;  
 }  
  
 String evaluar = msgBody;  
 **int** largo = msgBody.length();  
  
 **if** (largo>3)  
 {  
 evaluar = evaluar.substring(0,4);  
 }  
  
 **if** (evaluar.equals(**"GPS3"**)) {  
 **try** {  
  
 String menssaje = **"Need authorization to view coordenates from "**+ NumFrom;  
 **new** AlertDialog.Builder(getApplicationContext()).setTitle(**"Not Authorized"**).setMessage(menssaje).setNeutralButton(**"Close"**, **null**).show();  
  
 Intent i = **new** Intent(**"MENSAJE"**);  
 i.putExtra(**"<Key>"**,**"Not Authorized to view Coordenates"**);  
 sendBroadcast(i);  
  
 *//Toast.makeText(ServiceCommunicator.this, "Without Authorization to View Coordinates ", Toast.LENGTH\_LONG).show();* }  
 **catch** (Exception e)  
 {  
 Toast.*makeText*(ServiceCommunicator.**this**, **"Error "**+ e.toString(), Toast.***LENGTH\_LONG***).show();  
 }  
  
 **return**;  
 }  
  
 *// SHOW MAP* **if** (evaluar.equals(**"GPS2"**)) {  
  
 *//String lugar = msgBody.replace("GPS2 ","");* String localidad = **"http://maps.google.com/maps?q=loc:"** + msgBody.substring(5);  
 *//Toast.makeText(ServiceCommunicator.this, localidad, Toast.LENGTH\_LONG).show();* Uri uri = Uri.*parse*(localidad);  
 Intent intent2 = **new** Intent(Intent.***ACTION\_VIEW***, uri);  
 intent2.setFlags(Intent.***FLAG\_ACTIVITY\_NEW\_TASK***);  
 startActivity(intent2);  
 **return**;  
 }  
  
 SmsManager sms = SmsManager.*getDefault*();  
 String Mensaje = **""**;  
 String lista = Accesos();  
  
 **if** (lista.indexOf(NumFrom) < 1)  
 {  
 *// no tiene acceso* Toast.*makeText*(ServiceCommunicator.**this**, NumFrom + **" not Authorized"**, Toast.***LENGTH\_LONG***).show();  
  
 sms.sendTextMessage(NumFrom, **null**, **"GPS3"**, **null**, **null**);  
 **return**;  
 }  
  
 String Coordenadas = **"GPS2 "**+ GetCoordenada();  
  
 Toast.*makeText*(getApplicationContext(),NumFrom+ **" "**+ Coordenadas, Toast.***LENGTH\_LONG***).show();  
 sms.sendTextMessage(NumFrom, **null**, Coordenadas, **null**, **null**);  
 }  
 *//}* }  
 };  
  
  
  
 **public** String GetCoordenada()  
 {  
 LocationManager locMgt = (LocationManager) getSystemService(Context.***LOCATION\_SERVICE***);  
 **boolean** activo = locMgt.isProviderEnabled(LocationManager.***GPS\_PROVIDER***);  
 **if** (!activo)  
 {  
 Toast.*makeText*(getApplicationContext(),**"ERROR: GPS NOT AVAILABLE"**,Toast.***LENGTH\_LONG***).show();  
 **return "1"**;  
 }  
  
 **if** (ActivityCompat.*checkSelfPermission*(ServiceCommunicator.**this**, Manifest.permission.***ACCESS\_FINE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED*** && ActivityCompat.*checkSelfPermission*(ServiceCommunicator.**this**, Manifest.permission.***ACCESS\_COARSE\_LOCATION***) != PackageManager.***PERMISSION\_GRANTED***)  
 **return ""**;  
  
 *//locMgt.requestLocationUpdates(* Location loc = locMgt.getLastKnownLocation(LocationManager.***NETWORK\_PROVIDER***);  
 *//locMgt.requestLocationUpdates(locMgt, this);* **if** (loc == **null**) {  
 Toast.*makeText*(getApplicationContext(),**"ERROR: NO COORDENATES"**,Toast.***LENGTH\_LONG***).show();  
 **return "2"**;  
 }  
  
 **double** latitud = loc.getLatitude();  
 **double** longitud = loc.getLongitude();  
  
 String lat2 = String.*valueOf*(latitud);  
 lat2 = lat2.substring(0,10);  
  
 String long2 = String.*valueOf*(longitud);  
 long2 = long2.substring(0,10);  
  
 **return** lat2 +**","**+ long2;  
 }  
  
  
 **private** String Accesos()  
 {  
 String Lista = **"-00-+50370020015-"**;  
 String line;  
 **try** {  
 BufferedReader bReader = **new** BufferedReader(**new** InputStreamReader(openFileInput(**"ALLOW.txt"**)));  
 **while** ((line = bReader.readLine()) != **null**) {  
 Lista = Lista + line + **"-"**;  
 }  
 }  
 **catch** (Exception e)  
 {  
  
 }  
 **return** Lista;  
 }  
  
 **private void** Mostrar(Context context, String mensaje)  
 {  
 **try** {  
 context = getApplicationContext();  
 Activity a = (Activity) context;  
 *// TextView lugar = (TextView) a.findViewById(R.id.textView1);  
 // lugar.setText(mensaje);* }  
 **catch** (Exception e)  
 {  
 Toast.*makeText*(ServiceCommunicator.**this**, **"Error "**+ e.toString(), Toast.***LENGTH\_LONG***).show();  
 }  
 }  
  
 @Override  
 **public void** onCreate()  
 {  
 **super**.onCreate();  
 Toast.*makeText*(getApplicationContext(), **"Starting Search Service"**, Toast.***LENGTH\_LONG***).show();  
  
 IntentFilter filter = **new** IntentFilter();  
 filter.addAction(**"android.provider.Telephony.SMS\_RECEIVED"**);  
 filter.addAction(android.telephony.TelephonyManager.***ACTION\_PHONE\_STATE\_CHANGED***);  
 registerReceiver(**receiver**, filter);  
 }  
  
 @Override  
 **public void** onDestroy()  
 {  
 **super**.onDestroy();  
 *// unregisterReceiver(receiver);* }  
  
 @Nullable  
 @Override  
 **public** IBinder onBind(Intent intent) {  
 **return mBinder**;  
 }  
  
  
}

**Manifest.xml**

Permisos para servicio

Permisos para escoger contactos

Permisos para ejecutar al inicio

*<?***xml version="1.0" encoding="utf-8"***?>*<**manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.tdal6392.busqueda"**>  
  
<**uses-permission android:name="android.permission.ACCESS\_COARSE\_LOCATION"** />  
<**uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION"** />  
<**uses-permission android:name="android.permission.INTERNET"** />  
  
<**uses-permission android:name="android.permission.RECEIVE\_SMS"** />  
<**uses-permission android:name="android.permission.READ\_SMS"** />  
<**uses-permission android:name="android.permission.SEND\_SMS"** />  
  
<**uses-permission android:name="android.permission.READ\_CONTACTS"** />  
  
<**uses-permission android:name="android.permission.RECEIVE\_BOOT\_COMPLETED"** />  
  
<**uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"** />  
<**uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"** />  
  
  
 <**application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:supportsRtl="true"  
 android:theme="@style/AppTheme"**>

<**activity  
 android:name=".MainActivity"  
 android:screenOrientation="portrait"**

**android:theme="@style/AppTheme.NoActionBar"**>  
 <**intent-filter**>  
 <**action android:name="android.intent.action.MAIN"** />  
  
 <**category android:name="android.intent.category.LAUNCHER"** />  
 </**intent-filter**>  
 </**activity**>

<**activity  
 android:name=".GrantAccess"  
 android:label="Grant Access"  
 android:screenOrientation="portrait"**>  
 <**intent-filter**>  
 <**action android:name="android.intent.action.MAIN"** />  
 <**category android:name="android.intent.category.DEFAULT"** />  
 </**intent-filter**>  
 </**activity**>

<**service android:name=".ServiceCommunicator"** />  
  
 <**receiver android:enabled="true" android:name=".BootUpReceiver"  
 android:permission="android.permission.RECEIVE\_BOOT\_COMPLETED"**>  
  
 <**intent-filter**>  
 <**action android:name="android.intent.action.BOOT\_COMPLETED"** />  
 <**category android:name="android.intent.category.DEFAULT"** />  
 </**intent-filter**>  
 </**receiver**>  
  
 </**application**>  
  
</**manifest**>