

Manisha Vyas

Question:

1. Contact Database

- This Android app is combination of three Activities

A. AndContactsTester

- Has two buttons

- **Go to AndContact**

Clicking this button will explicitly start for result to the subactivity "AndContacts"

- ○ `Intent intent = new Intent(AndContactsTester.this,`
○ `AndContacts.class);`
○ `startActivityForResult(intent, SHOW_SUB_ACTIVITY);`

- **Go to ContactPicker**

Clicking this button will implicitly start for result to the subactivity "ContactPicker"

- ○ `Intent intent1 = new Intent (Intent.ACTION_PICK,`
○
○ `Uri.parse("content://contacts/people"));`
○ `startActivityForResult(intent1, PICK_CONTACT);`

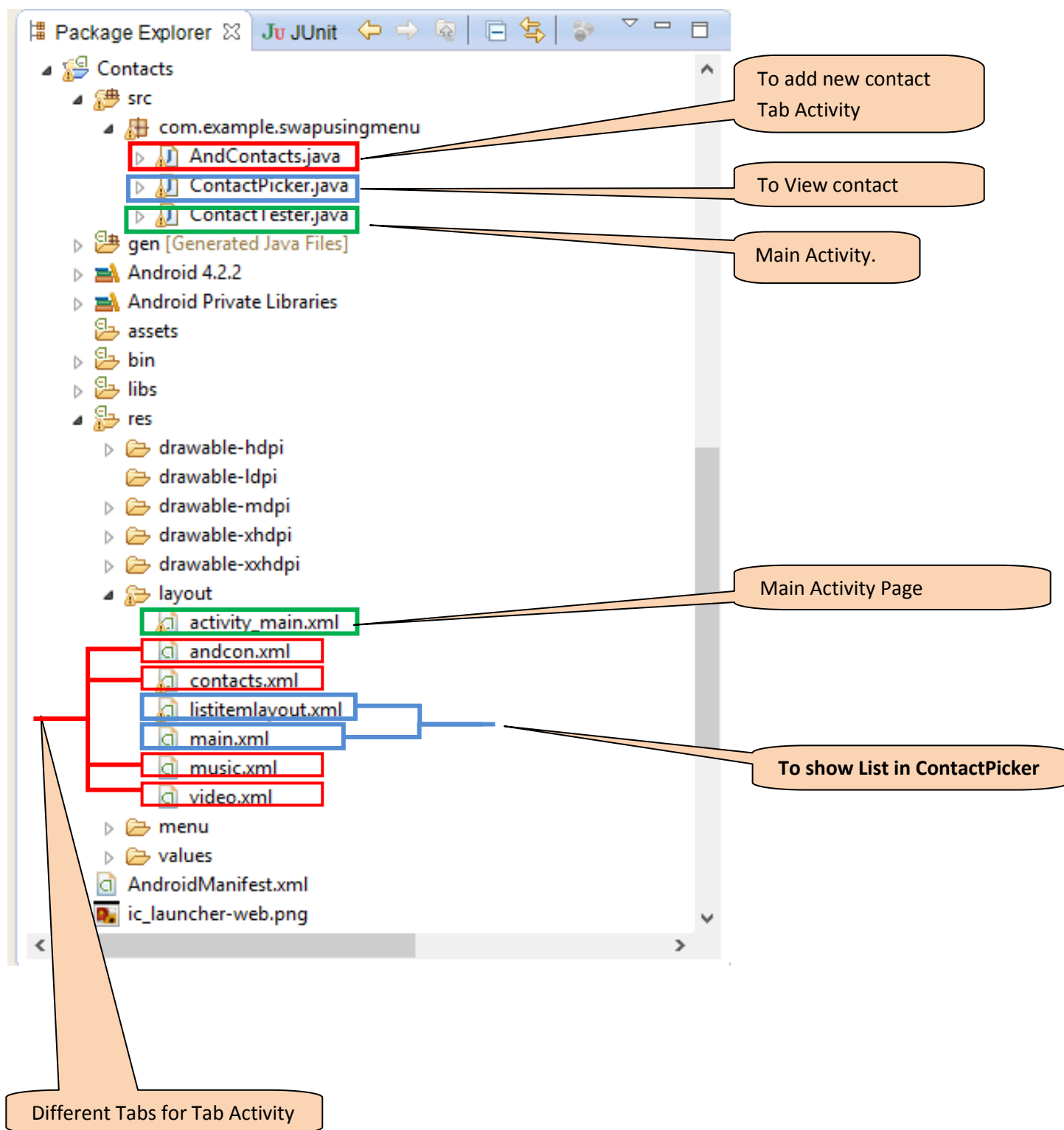
B. AndContacts

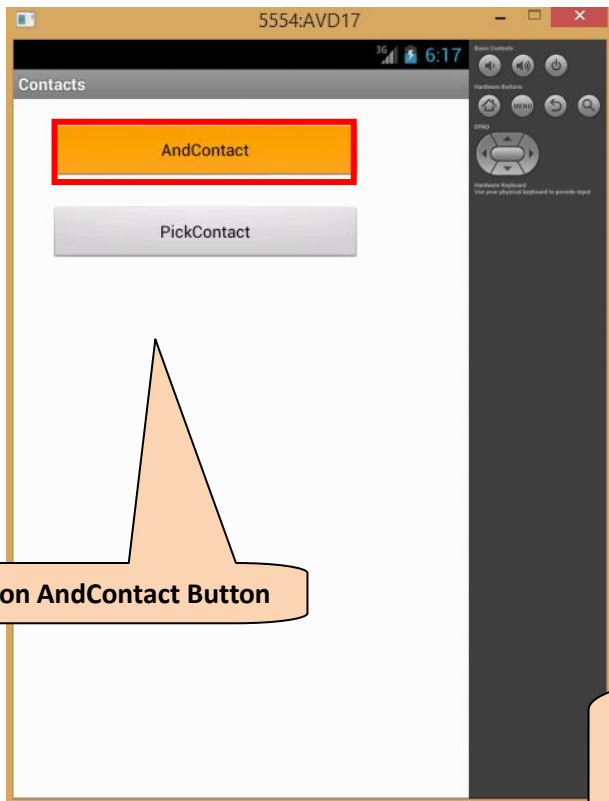
- This Activity inherits from TabActivity
- When the name entered in the AndContacts screen, it is **saved to the Contact database**.
- This is a subactivity which returns name to "AndContactsTester". The name is obtained from the TabActivity

C. ContactPicker

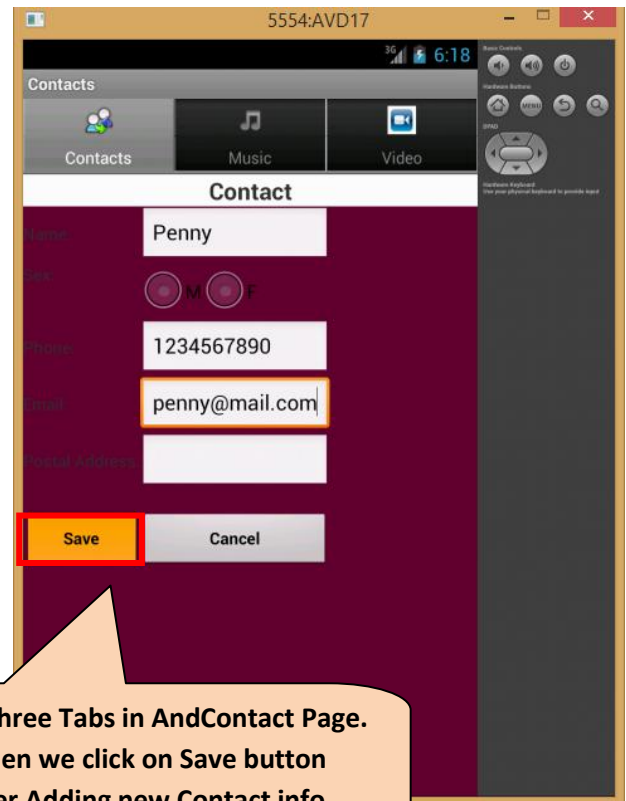
- This is a subactivity which returns a selected contact name to "AndContactsTester".

2. Modify the **AndContacts** class of the **Tab Activity** Project so that when the name entered in the AndContacts screen is saved to the Contact database.
3. Create a class **AndContactsTester** class which implicitly starts a new sub-Activity by specifying the ACTION_PICK and the contact database URI (content://contacts/):

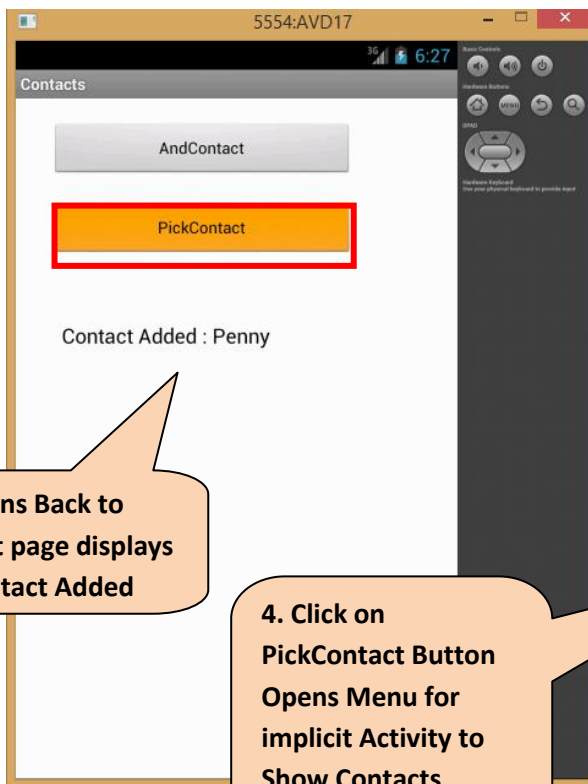




1. Click on AndContact Button

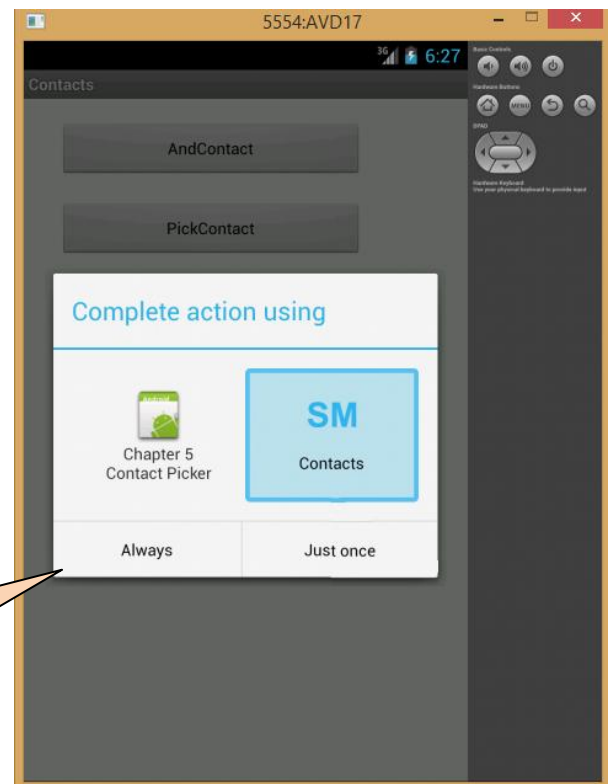


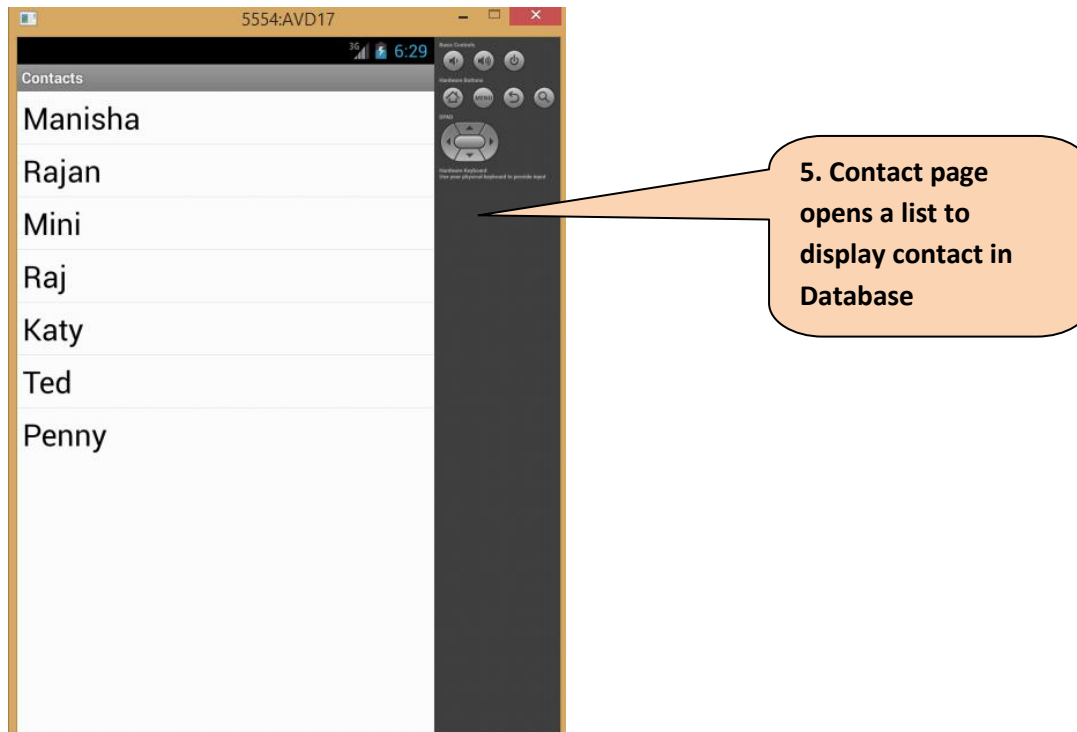
2. Three Tabs in AndContact Page. When we click on Save button after Adding new Contact info



3. Returns Back to Contact page displays the contact Added

4. Click on PickContact Button Opens Menu for implicit Activity to Show Contacts





AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.swapusingmenu"
    android:versionCode="1"
    android:versionName="1.0" >

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >

        <activity
            android:name="com.example.swapusingmenu.ContactTester"
            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=".AndContacts"
            android:label="@string/app_name">
        </activity>
    </application>
</manifest>
```

```

        <activity android:name=".ContactPicker"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.PICK"></action>
                <category android:name="android.intent.category.DEFAULT"></category>
                <data android:path="contacts" android:scheme="content"></data>
            </intent-filter>
        </activity>
    </application>
    <uses-sdk android:minSdkVersion="7" />

    <uses-permission android:name="android.permission.READ_CONTACTS"/>
    <uses-permission android:name="android.permission.WRITE_CONTACTS" />

</manifest>

```

Here to use Contact Database we are required to add these two lines in our AndroidManifest.xml

//ContactTester.java

```

package com.example.swapusingmenu;

import android.app.Activity;
import android.content.Intent;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.Contacts.People;
import android.view.*;
import android.widget.AdapterView;
import android.widget.AutoCompleteTextView;
import android.widget.Button;
import android.widget.TextView;

public class ContactTester extends Activity {
    private static final int AND_CON_ACTIVITY = 1;
    private static final int PICK_CON_ACTIVITY = 2;

    Button and_conb, pick_conb;
    TextView result_con;

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

        and_conb = (Button) findViewById(R.id.and_conb);
        result_con = (TextView) findViewById(R.id.result_text);
        pick_conb = (Button) findViewById(R.id.pick_conb);

        and_conb.setOnClickListener(new View.OnClickListener() {
            @Override

```

```

        public void onClick(View v) {
            Intent intent = new Intent
                (ContactTester.this, AndContacts.class);
            startActivityForResult(intent, AND_CON_ACTIVITY);
        }

    });
    pick_conb.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            Intent intent = new Intent
                (Intent.ACTION_PICK, Uri.parse("content://contacts/"));
            startActivityForResult(intent, PICK_CON_ACTIVITY);
        }
    });
}

@Override
public void onActivityResult(int requestCode, int resultCode, Intent data)
{
    super.onActivityResult(requestCode, resultCode, data);
    switch (requestCode)
    {
        case AND_CON_ACTIVITY:
        {
            if(resultCode == Activity.RESULT_OK)
            {
                String new_con = data.getStringExtra("Name");
                result_con.setText("Contact Added : "+new_con);
            }
            break;
        }
        case PICK_CON_ACTIVITY :
        {
            if (resultCode == Activity.RESULT_OK)
            {
                Uri contactData = data.getData();
                Cursor c = managedQuery(contactData, null, null, null, null);
                c.moveToFirst();
                String new_con =
                    c.getString(c.getColumnIndexOrThrow(Contacts.People.NAME));
                result_con.setText("You have selected : "+new_con);
            }
            break;
        }
    }
}

@Override //Option menu/Action menu
public boolean onCreateOptionsMenu(Menu menu)
{
    MenuInflater menuInflater = getMenuInflater();
    menuInflater.inflate(R.menu.main, menu);
    return true;
}
}

```

//ContactPicker.java

```
package com.example.swapusingmenu;

import android.app.Activity;
import android.content.Intent;
import android.database.Cursor;
import android.net.Uri;
import android.os.Bundle;
import android.provider.Contacts.People;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.SimpleCursorAdapter;

public class ContactPicker extends Activity{
    @Override
    public void onCreate(Bundle icle) {
        super.onCreate(icle);
        setContentView(R.layout.main);

        Intent intent = getIntent();
        String dataPath = intent.getData().toString();

        final Uri data = Uri.parse(dataPath + "people/");
        final Cursor c = managedQuery(data, null, null, null, null);

        String[] from = new String[] {People.NAME};
        int[] to = new int[] { R.id.itemTextView };

        SimpleCursorAdapter adapter = new
            SimpleCursorAdapter(this,R.layout.listitemlayout,c,from,to);
        ListView lv = (ListView)findViewById(R.id.contactListView);
        lv.setAdapter(adapter);

        lv.setOnItemClickListener(new OnItemClickListener() {
            public void onItemClick(AdapterView<?> parent, View view, int pos, long id)
            {
                // Move the cursor to the selected item
                c.moveToPosition(pos);
                // Extract the row id.
                int rowId = c.getInt(c.getColumnIndexOrThrow("_id"));
                // Construct the result URI.
                Uri outURI = Uri.parse(data.toString() + rowId);
                Intent outData = new Intent();
                outData.setData(outURI);
                setResult(Activity.RESULT_OK, outData);
                finish();
            }
        });
    }
}
```

```
//AndContacts.java
```

```
package com.example.swapusingmenu;
```

```
import java.util.ArrayList;
```

```
import android.app.Activity;
```

```
import android.app.TabActivity;
```

```
import android.content.ContentProviderOperation;
```

```
import android.content.Intent;
```

```
import android.content.OperationApplicationException;
```

```
import android.os.Bundle;
```

```
import android.os.RemoteException;
```

```
import android.provider.ContactsContract;
```

```
import android.provider.ContactsContract.CommonDataKinds.*;
```

```
import android.provider.ContactsContract.*;
```

```
import android.view.View;
```

```
import android.widget.Button;
```

```
import android.widget.EditText;
```

```
import android.widget.TabHost;
```

```
public class AndContacts extends TabActivity {
```

```
    Button saveb, cancelb;
```

```
    EditText txtname, txtemail, txtphone, txtpostaladd;
```

```
    TabHost mTabHost = null;
```

```
    @Override
```

```
    public void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.andcon);
```

```
        mTabHost = getTabHost();
```

```
        mTabHost.addTab(mTabHost.newTabSpec("tab_test1").setIndicator("Contacts",  
getResources().getDrawable(R.drawable.contact)).setContent(R.id.contactsLayout));
```

```
        mTabHost.addTab(mTabHost.newTabSpec("tab_test2").setIndicator("Music",  
getResources().getDrawable(R.drawable.music)).setContent(R.id.musicLayout));
```

```
        mTabHost.addTab(mTabHost.newTabSpec("tab_test3").setIndicator("Video",  
getResources().getDrawable(R.drawable.video)).setContent(R.id.videoLayout));
```

```
        mTabHost.setCurrentTab(0);
```

```
        saveb = (Button) findViewById(R.id.buttonsave);
```

```
        cancelb = (Button) findViewById(R.id.buttoncancel);
```

```
        txtname = (EditText) findViewById(R.id.txtname);
```

```
        txtemail = (EditText) findViewById(R.id.txtemail);
```

```
        txtphone = (EditText) findViewById(R.id.txtphone);
```

```
        txtpostaladd = (EditText) findViewById(R.id.txtpostaladdress);
```

```
        saveb.setOnClickListener(new View.OnClickListener() {
```

```
            @Override
```

```
            public void onClick(View v) {
```

```
                Intent result = new Intent();
```

```
                result.putExtra("Name", txtname.getText().toString());
```



```

        addcontact();
        setResult(RESULT_OK, result);

        finish();
    }
});

cancelb.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {

        setResult(RESULT_CANCELED, null);
        finish();
    }
});
}

protected void addcontact() {
    ArrayList ops = new
        ArrayList();
    int rawContactInsertIndex = ops.size();

    ops.add(ContentProviderOperation.newInsert(RawContacts.CONTENT_URI)
        .withValue(RawContacts.ACCOUNT_TYPE, null)
        .withValue(RawContacts.ACCOUNT_NAME, null )
        .build());

    //phone field
    ops.add(ContentProviderOperation.newInsert(ContactsContract.Data.CONTENT_URI)
        .withValueBackReference(ContactsContract.Data.RAW_CONTACT_ID,
            rawContactInsertIndex)
        .withValue(Data.MIMETYPE, Phone.CONTENT_ITEM_TYPE)
        .withValue(Phone.NUMBER, txtphone.getText().toString())
        .build());

    //name field
    ops.add(ContentProviderOperation.newInsert(Data.CONTENT_URI)
        .withValueBackReference(Data.RAW_CONTACT_ID, rawContactInsertIndex)
        .withValue(Data.MIMETYPE, StructuredName.CONTENT_ITEM_TYPE)
        .withValue(StructuredName.DISPLAY_NAME,
            txtname.getText().toString())
        .build());

    //email field
    ops.add(ContentProviderOperation.newInsert(ContactsContract.Data.CONTENT_URI)
        .withValueBackReference(Data.RAW_CONTACT_ID, 0)

.withValue(ContactsContract.Data.MIMETYPE, ContactsContract.CommonDataKinds.Email.CONTENT
_ITEM_TYPE)
        .withValue(ContactsContract.CommonDataKinds.Email.DATA,
txtemail.getText().toString())
        .withValue(ContactsContract.CommonDataKinds.Email.TYPE,
ContactsContract.CommonDataKinds.Email.TYPE_HOME)

```

```
.build());

try {
    //update data to the database, operations are run in batchmode
    getContentResolver().applyBatch(ContactsContract.AUTHORITY, ops);
} catch (RemoteException e) {
    e.printStackTrace();
} catch (OperationApplicationException e) {
    e.printStackTrace();
}

}
```