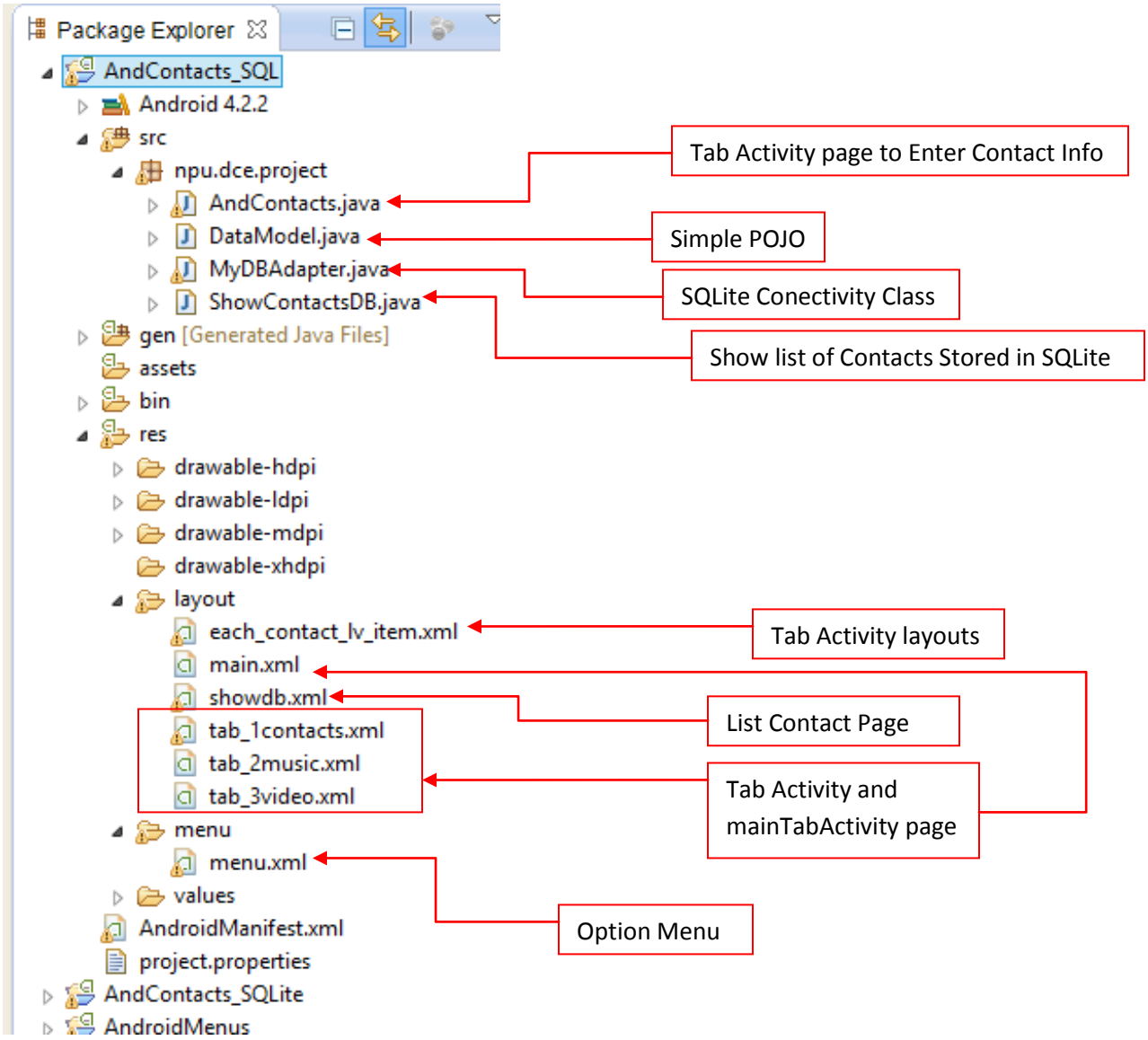
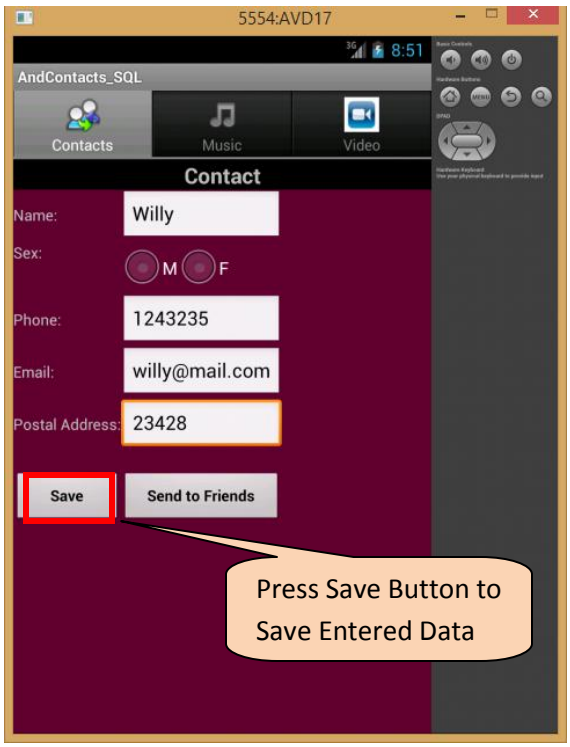
Manisha Vyas

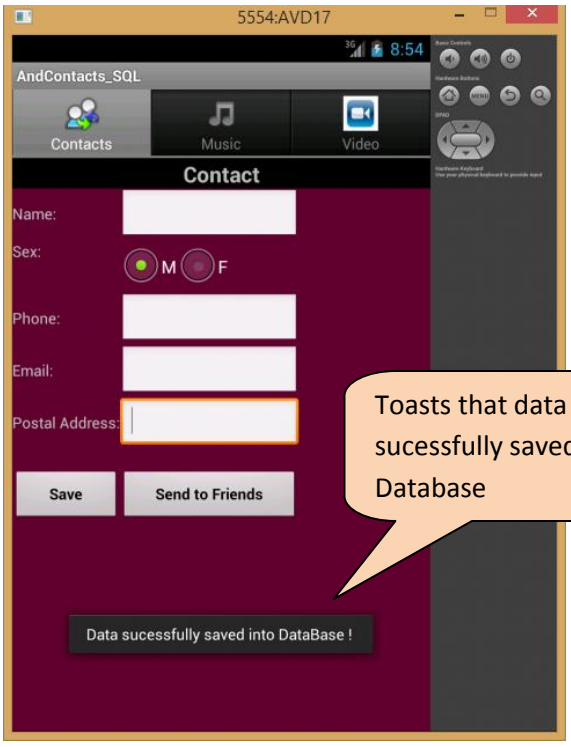
Question:

- Continue the Tab Activity so that the data entered by the users will be saved in a data base.
- You need to demonstrate database operation that you can show at least more than 1 recorded data (a list of contacts), as well as it should have more than 1 field (column in database table) being retrieved (such that name, phone, email, and address).

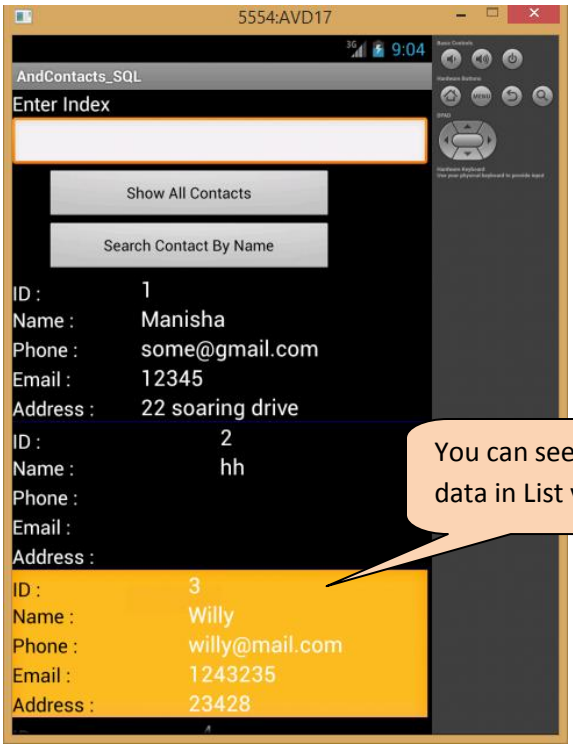
We will be working maily with

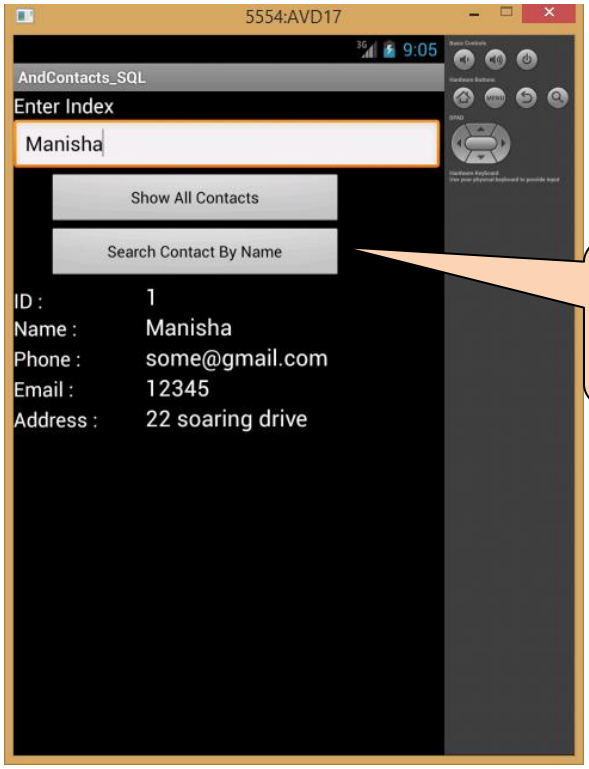



5554:AVD17
8:51
AndContacts_SQL
Contacts
Music
Video
Contact
Name:
Willy
Sex:
M
F
Phone:
1243235
Email:
willy@mail.com
Postal Address:
23428
Save
Send to Friends
Press Save Button to Save Entered Data


5554:AVD17
8:54
AndContacts_SQL
Contacts
Music
Video
Contact
Name:
Sex:
M
F
Phone:
Email:
Postal Address:
Save
Send to Friends
Toasts that data are sucessfully saved in Database
Data sucessfully saved into DataBase !


5554:AVD17
8:55
AndContacts_SQL
Contacts
Music
Video
Contact
Name:
Sex:
M
F
Phone:
Email:
Postal Address:
Save
Send to Friends
Once the data stored you can press Menu to go ContectList view using OptionMenu
Search Item in DataBase


5554:AVD17
9:04
AndContacts_SQL
Enter Index
Show All Contacts
Search Contact By Name
ID : 1
Name : Manisha
Phone : some@gmail.com
Email : 12345
Address : 22 soaring drive
ID : 2
Name : hh
Phone :
Email :
Address :
ID : 3
Name : Willy
Phone : willy@mail.com
Email : 1243235
Address : 23428
You can see data in List view



//AndContacts.java

```
package npu.dce.project;

import android.os.Bundle;
import android.app.TabActivity;
import android.content.Intent;
import android.view.Menu;
import android.view.MenuInflater;
import android.view.MenuItem;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TabHost;
import android.widget.Toast;

public class AndContacts extends TabActivity {

	private static final int SHOW_CONTACTS = 0;
	private Button saveb,cancelb;
	private EditText txtname,txtemail,txtphone,txtpostaladd;
	private String strName,strEmail,strPhone,strPostalAdd;

	private MyDBAdapter myDBAdapter;

	TabHost mTabHost = null;


	@Override
```

```
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);

        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);

       myDBAdapter = new MyDBAdapter(this);
       myDBAdapter.open();

       //myDBAdapter.deleteAllEntries();
       saveb.setOnClickListener(new View.OnClickListener() {
                    @Override
                    public void onClick(View v) {

                           strName = txtname.getText().toString();
                    strEmail = txtemail.getText().toString();
                    strPhone = txtphone.getText().toString();
                    strPostalAdd = txtpostaladd.getText().toString();

                           DataModel newContact = new
DataModel(4,strName,strEmail,strPhone,strPostalAdd);
                   myDBAdapter.insertEntry(newContact);
                   //updateArray();

                   txtname.setText("");
                   txtemail.setText("");
                   txtphone.setText("");
                   txtpostaladd.setText("");

                   Toast.makeText(AndContacts.this,"Data sucessfully saved into DataBase
! ",Toast.LENGTH_LONG).show();
                    }
               });


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

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

        public boolean onCreateOptionsMenu(Menu menu)
    {
         MenuInflater menuInflater = getMenuInflater();
         menuInflater.inflate(R.menu.menu, menu);
         return true;
    }
        public boolean onOptionsItemSelected(MenuItem item) {
            super.onOptionsItemSelected(item);

            switch (item.getItemId())
            {
            case R.id.filter_name:
                Intent i = new Intent(this, ShowContactsDB.class);
                startActivityForResult(i, SHOW_CONTACTS);
                return true;

            default:
                return super.onOptionsItemSelected(item);
            }
         }
}
```

//DataModel.java A simple POJO

```
package npu.dce.project;

public class DataModel {

        private int theId;
        private String name, phone, email, postaladdr;

        public DataModel(int theId,String name, String phone, String email,String
postaladdr) {
                super();
                this.theId = theId;
                this.name = name;
                //this.gender = gender;
                this.phone = phone;
                this.email = email;
                this.postaladdr = postaladdr;
        }

        public void setTheId(int theId) {
                this.theId = theId;
        }

        public void setName(String name) {
```

```
		this.name = name;
	}


	public void setPhone(String phone) {
		this.phone = phone;
	}

	public void setEmail(String email) {
		this.email = email;
	}

	public void setPostaladdr(String postaladdr) {
		this.postaladdr = postaladdr;
	}

	public int getTheId() {
		return theId;
	}

	public String getName() {
		return name;
	}

	public String getPhone() {
		return phone;
	}

	public String getEmail() {
		return email;
	}

	public String getPostaladdr() {
		return postaladdr;
	}
}
```

//ShowContactsDB.java

```
//  This class saves is Data model class to store info between Resume() and Pause()
package npu.dce.project;

import android.annotation.SuppressLint;
import android.app.Activity;
import android.database.Cursor;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.SimpleCursorAdapter;

public class ShowContactsDB extends Activity
{
```

```
    private Button showall,showbyid;
    private EditText txt_searchbox;
    private MyDBAdapter myDBAdapter;
    private ListView lv;
    private Cursor cursor;
    private SimpleCursorAdapter dataAdapter;

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

    showbyid = (Button) findViewById(R.id.btn_showbyid);
    showall = (Button) findViewById(R.id.btn_showall);

    lv = (ListView)findViewById(R.id.lv_contacts);

    myDBAdapter = new MyDBAdapter(this);
    myDBAdapter.open();

          displayListView("ALL");

     showbyid.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {
                      txt_searchbox = (EditText) findViewById(R.id.txt_searchbox);
                      displayListView(txt_searchbox.getText().toString());
                }
          });

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

                      displayListView("ALL");
                }
          });
  }
    @SuppressLint("NewApi")
    private void displayListView(String whichView)
    {
          if (whichView.equals("ALL"))//which view -> all Contacts
          {
                cursor = myDBAdapter.getAllEntries();
          }
          else //which view -> by Contact Name
          {
                cursor = myDBAdapter.getEntry(whichView);
          }

            // The desired columns to be bound
            String[] columns = new String[] {
              MyDBAdapter.KEY_ID,
              MyDBAdapter.KEY_NAME,
              MyDBAdapter.KEY_PHONE,
```

```
                MyDBAdapter.KEY_EMAIL,
                MyDBAdapter.KEY_POSTALADDR
              };

              // the XML defined views which the data will be bound to
              int[] to = new int[] {
                R.id.lv_id,
                R.id.lv_name,
                R.id.lv_phone,
                R.id.lv_email,
                R.id.lv_add
              };

        dataAdapter = new SimpleCursorAdapter(this,
R.layout.each_contact_lv_item,cursor, columns, to,0);
        lv.setAdapter(dataAdapter);

       }
}
```

//MyDBAdapter.java

```
package npu.dce.project;

import java.util.ArrayList;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteDatabase.CursorFactory;
import android.database.sqlite.SQLiteException;
import android.database.sqlite.SQLiteOpenHelper;
import android.util.Log;
import android.widget.Toast;

public class MyDBAdapter {

      private static final String DATABASE_NAME = "conDatabase.db";
    private static final String DATABASE_TABLE = "contactsTable";
    private static final int DATABASE_VERSION = 2;

    //EACH COLUMN IN DATABASE TABLE
    public static final String KEY_ID = "_id"; //primary key, CursorAdapter will use
this
    public static final String KEY_NAME = "NAME";
    public static final String KEY_PHONE = "PHONE";
    public static final String KEY_EMAIL = "EMAIL";
    public static final String KEY_POSTALADDR = "POSTALADDR";

    private SQLiteDatabase db;
    private final Context context;
```

```
    private myDatabaseOpenHelper dbHelper;

    private String tmpID,tempName,tempPhone,tempEmail,temAdd;


    public MyDBAdapter(Context _context)
    {
        this.context = _context;
        dbHelper = new myDatabaseOpenHelper(context, DATABASE_NAME, null,
DATABASE_VERSION);
    }

    public void open() throws SQLiteException
    {
        try
        {
            db = dbHelper.getWritableDatabase();
        }
        catch (SQLiteException ex)
        {
            db = dbHelper.getReadableDatabase();
        }
    }

    //wrapper method, release database object
    public void close()
    {
        db.close();
    }

    //Insert a new entry (consists a set of rows) into the table
    public long insertEntry(DataModel dataModel)
    {
        ContentValues rows = new ContentValues();

        rows.put(KEY_NAME, dataModel.getName());
        rows.put(KEY_PHONE, dataModel.getPhone());
        rows.put(KEY_EMAIL, dataModel.getEmail());
        rows.put(KEY_POSTALADDR, dataModel.getPostaladdr());

        return db.insert(DATABASE_TABLE, null, rows);
    }
    //return a single DataModel object based on what name to search
    public Cursor getAllEntries() throws SQLException {

        Cursor cursor = db.query(DATABASE_TABLE,
                    new String[] {KEY_ID, KEY_NAME, KEY_PHONE, KEY_EMAIL,
KEY_POSTALADDR},null, null, null, null,null, null);

        if (cursor != null)
        {
             cursor.moveToFirst();
        }
        return cursor;
    }
```

```
    public Cursor getEntry(String searchname) throws SQLException {

       Cursor cursor = db.query(DATABASE_TABLE,
                    new String[] {KEY_ID, KEY_NAME, KEY_PHONE, KEY_EMAIL,
KEY_POSTALADDR},
                    KEY_NAME + "=" + "'" + searchname.trim() + "'", null, null,
null,null, null);

        return cursor;
    }

    public void deleteAllEntries() {
        db.execSQL("DELETE FROM " + DATABASE_TABLE);
    }

    public boolean removeEntry(long _rowIndex) {
        return db.delete(DATABASE_TABLE, KEY_ID + "=" + _rowIndex, null) > 0;
    }

    private static class myDatabaseOpenHelper extends SQLiteOpenHelper {

        public myDatabaseOpenHelper(Context context, String name,
                        CursorFactory factory, int version) {
                    super(context, name, factory, version);
                    // TODO Auto-generated constructor stub
             }

        private static final String CREATE_TABLE =
                "create table " + DATABASE_TABLE + " (" +
                        KEY_ID + " integer primary key autoincrement, " +
                        KEY_NAME + " text not null, " +
                        KEY_PHONE + " text, " +
                        KEY_EMAIL + " text, " +
                        KEY_POSTALADDR + " text);";

              @Override
              public void onCreate(SQLiteDatabase _db) {
               _db.execSQL(CREATE_TABLE);
            }

              @Override
            public void onUpgrade(SQLiteDatabase _db,
                      int _oldVersion, int _newVersion) {
              Log.w("TaskDBAdapter", "Upgrading from version " +
                                 _oldVersion + " to " +
                                 _newVersion
                      + ", which will destroy all old data");

              _db.execSQL("DROP TABLE IF EXISTS " + DATABASE_TABLE);

              onCreate(_db);
            }
    }
}
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="npu.dce.project"
      android:versionCode="1"
      android:versionName="1.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".AndContacts"
                  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=".ShowContactsDB"
             android:label="@string/app_name">
        </activity>

    </application>


    <uses-sdk android:minSdkVersion="7" />
    <uses-permission android:name="android.permission.READ_CONTACTS"/>
      <uses-permission android:name="android.permission.WRITE_CONTACTS" />

</manifest>
```