

## BÀI THỰC HÀNH BUỔI 4

### Gồm 3 bài: Bài 7, Bài 8 và Bài 9

## I. LÝ THUYẾT CẦN ÔN TẬP

- Lưu dữ liệu với SQLite.
  1. Tạo 1 CSDL
  2. Tạo bảng
  3. Thêm dữ liệu vào bảng
  4. Truy xuất dữ liệu
  5. Cập nhập dữ liệu
  6. Xóa dữ liệu
  7. Xóa bảng
  8. Xóa CS dữ liệu
- Lưu dữ liệu với Content Provider.
  1. Tạo ra một Content Provider.
  2. Sử dụng một content provider dựng sẵn.

## II. THỰC HÀNH

### BÀI 7: LẬP TRÌNH LƯU TRỮ DỮ LIỆU VỚI SQLITE

Tạo chương trình tạo cơ sở dữ liệu SQLite quản lý các Sinh viên (SV), mỗi SV gồm các thông tin: *họ và tên*, *MSSV*, *số điện thoại*, *địa chỉ email*, có giao diện chính như hình 4.1, bao gồm:



Hình 4.1: Giao diện chính

- 1 TextView hiển thị 1 text “CSDL QUẢN LÝ SINH VIÊN DÙNG SQLITE”,
- 6 Button để thực hiện tương tác của người dùng
- 1 FrameLayout (Layout con) chứa 1 ImageView và 1 ListView (để hiển thị danh sách SV.

với các chức năng như sau:

- **Thêm SV** (Khi button “THÊM SINH VIÊN” được chọn),
- **Xem tất cả SV** (Khi button “XEM TẤT CẢ SINH VIÊN” được chọn),
- **Xem một SV** (Khi button “XEM MỘT SINH VIÊN” được chọn),
- **Sửa SV** (Khi button “SỬA MỘT SINH VIÊN” được chọn),
- **Xóa SV** (Khi button “XÓA MỘT SINH VIÊN” được chọn),
- **Trở về giao diện lúc khởi chạy** (Khi button “TRỞ VỀ MÀN HÌNH CHÍNH” được chọn).

Trong ứng dụng này, ngoài Activity chính, chúng ta xây dựng thêm **1 Java class** và **4 Activity** phụ có chức năng:

- **Java class**, trong hướng dẫn này nó được đặt tên là **DBAdapter**, dùng để: định nghĩa các hằng, các biến và các đối tượng để được dùng trong ứng dụng này; tạo ra CSDL và xây dựng các phương thức để sử dụng trong các activity với các chức năng như: *đóng/mở CSDL, thêm 1 sinh viên, xem các sinh viên (tất cả hoặc một), sửa 1 sinh viên, xoá 1 sinh viên,...*
- 1 Activity dùng để thêm sinh viên, trong hướng dẫn này nó được đặt tên là **ThemStudent**.
- 1 Activity dùng để xem 1 sinh viên, trong hướng dẫn này nó được đặt tên là **XemStudent**.
- 1 Activity dùng để sửa 1 sinh viên, trong hướng dẫn này nó được đặt tên là **SuaStudent**.
- 1 Activity dùng để xoá 1 sinh viên, trong hướng dẫn này nó được đặt tên là **XoaStudent**.

Các bước thực hiện như sau:

**Bước 1:** Mở project mới tên là **B7\_TenSV\_MSSV**, với Activity chính tên là **MainActivity**

**Bước 2:** Lập trình Layout cho Activity chính với giao diện như hình 4.1 và chạy thử với máy ảo (tất nhiên là chưa có đáp ứng khi tương tác lên các button).

Sau đây là mã nguồn Layout của mainActivity, sinh viên cần tham khảo và phân tích để có thể vận dụng sáng tạo trong các ứng dụng khác.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:orientation="vertical"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/txtthemsv"
        android:layout_width="match_parent"
        android:layout_height="45dp"
        android:background="#bbbbff"
        android:gravity="center_vertical|center_horizontal"
        android:text="CSDL QUẢN LÝ SINH VIÊN DÙNG SQLITE"
        android:textAllCaps="true"
        android:textColor="#ff0000"
        android:textSize="20sp"
        android:textStyle="bold" />
```

```
<Button
    android:id="@+id/btnthemsv"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#bbffbb"
    android:textSize="18sp"
    android:text="Thêm student" />

<Button
    android:id="@+id/btnxemtatca"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#ffff00"
    android:gravity="center"
    android:textSize="18sp"
    android:text="Xem tất cả student" />

<Button
    android:id="@+id/btnxemmotstv"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:background="#ffaa66"
    android:textSize="18sp"
    android:text="Xem một student" />

<Button
    android:id="@+id/btncapnhatsv"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:background="#ddddff"
    android:textSize="18sp"
    android:text="Sửa một student" />

<Button
    android:id="@+id/btnxoasv"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:background="#ff99ff"
    android:textSize="18sp"
    android:text="Xóa một student" />

<Button
    android:id="@+id/btntrove"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:background="#ADABAD"
    android:textSize="18sp"
    android:text="TRỞ VỀ MÀN HÌNH CHÍNH" />

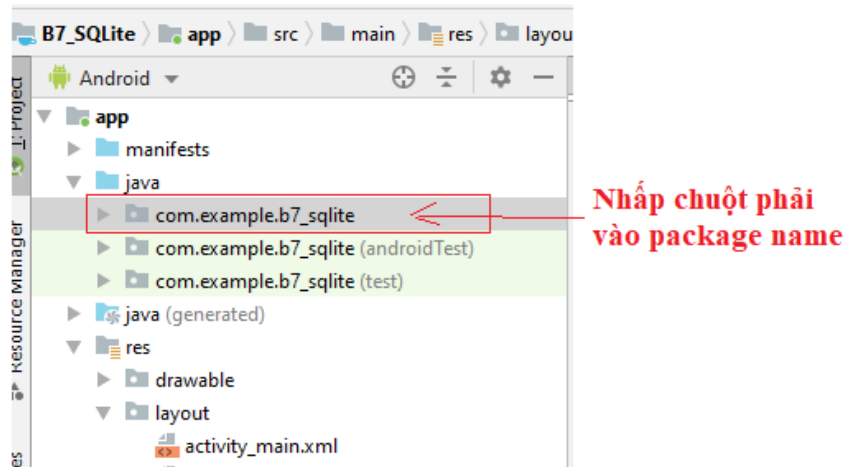
<FrameLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent" >
    <ImageView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:src="@drawable/dhcantho" />
    <ListView
        android:id="@+id/lvperson"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#cccccc">
    </ListView>

</FrameLayout>

</LinearLayout>
```

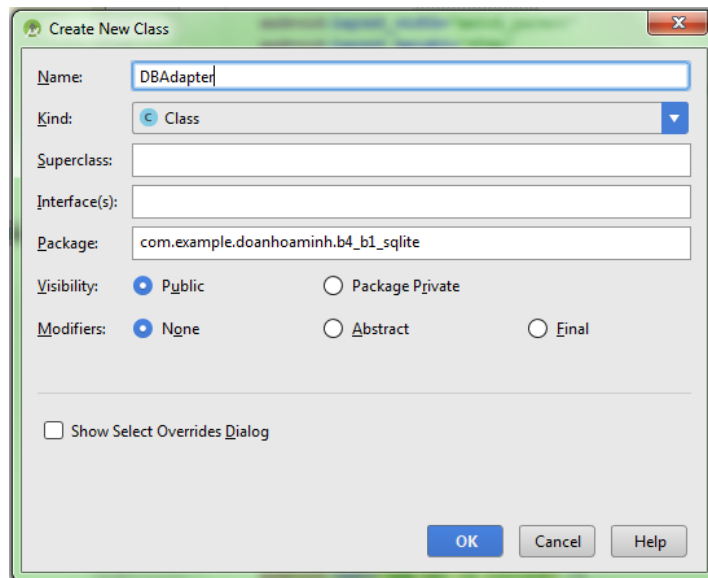
**Bước 3:** Tạo mới 1 **Java class** có chức năng tạo CSDL, trong hướng dẫn này lớp có tên là **DBAdapter**. Lớp này có chức năng là: *định nghĩa các hằng, các biến, các đối tượng mà chúng sẽ được sử dụng, tạo ra CSDL, xây dựng một số phương thức ( mở CSDL, đóng CSDL, thêm SV vào CSDL, truy xuất tất cả SV, truy xuất 1 SV, thêm 1 SV, xóa SV) sẽ được sử dụng trong các activity hoặc các lớp khác.*

Cách làm: Nhấp chuột phải vào tên miền của ứng dụng (Hình 4.2) → Chọn New → Chọn **Java class**



Hình 4.2

Cửa sổ “Create New Class” xuất hiện, nhập tên class (trong ví dụ này là DBAdapter) vào trường Name và chọn OK



Hình 4.3

Sau đây là mã nguồn của lớp DBAdapter, bao gồm các chú thích, sinh viên tham khảo, phân tích, đối chiếu với lý thuyết đã học.

**Lưu ý:** Trong các mã nguồn mẫu sau đây chỉ tạo ra CSDL với bảng chỉ gồm 3 cột: Số thứ tự (ID), mã số SV, họ và tên, địa chỉ email, sinh viên phải tự nghiêm cứu thêm vào một thông tin nữa là số điện thoại. SV nên nhập từng lệnh để nhớ, hiểu và có thể vận dụng trong các ứng dụng khác.

```
package com.example.b7_sqlite;
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.SQLiteOpenHelper;
import android.util.Log;

/**
 * Created by Doan Hoa Minh on 12-Mar-18.
 */

public class DBAdapter {
    /* Định nghĩa các hằng số để chứa các trường khác nhau cho các bảng mà ta sẽ tạo ra trong CSDL.*/
    static final String KEY_ROWID = "_id";
    static final String KEY_MSSV = "mssv";
    static final String KEY_NAME = "name";
    static final String KEY_EMAIL = "email";
    static final String TAG = "DBAdapter";
    static final String DATABASE_NAME = "MyDB";
    static final String DATABASE_TABLE = "students";
    static final int DATABASE_VERSION = 1;
    final Context context;
    DatabaseHelper DBHelper;
    SQLiteDatabase db;
    //Định nghĩa chuỗi sinh ra bảng sinh viên.
    static final String DATABASE_CREATE =
        "create table students (_id integer primary key autoincrement, "
        + "mssv text not null, name text not null, email text not null);";

    //Hàm tạo (constructor)
    public DBAdapter(Context ctx)
    {
        this.context = ctx;
        DBHelper = new DatabaseHelper(context);
    }

    //Tạo đối tượng SQLiteOpenHelper tên là DatabaseHelper

    private static class DatabaseHelper extends SQLiteOpenHelper
    {
        //Hàm tạo
        DatabaseHelper(Context context)
        {
            super(context, DATABASE_NAME, null, DATABASE_VERSION);
        }

        //Sinh ra CSDL SQLite tên db và tạo bảng sinh viên.
        @Override
        public void onCreate(SQLiteDatabase db)
        {
            try {
```

```

        db.execSQL(DATABASE_CREATE);
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

// Tạo hàm cập nhật CSDL
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)
{
    //Tạo thông báo trên màn hình console của IDE
    Log.w(TAG, "Upgrading database from version " + oldVersion + " to "
        + newVersion + ", which will destroy all old data");
    // Dùng hàm execSQL để xóa bảng cũ.
    db.execSQL("DROP TABLE IF EXISTS students");
    // Tạo CSDL cho phiên bản mới
    onCreate(db);
}

}

//---Tạo hàm mở CSDL---
public DBAdapter open() throws SQLException
{
    db = DBHelper.getWritableDatabase();
    return this;
}

//--- Hàm đóng CSDL---
public void close()
{
    DBHelper.close();
}

//---Hàm chèn một nội dung (sinh viên) vào CSDL
public long insertStudent(String name, String mssv, String email)
{
    ContentValues initialValues = new ContentValues();
    initialValues.put(KEY_NAME, name);
    initialValues.put(KEY_MSSV, mssv);
    initialValues.put(KEY_EMAIL, email);
    return db.insert(DATABASE_TABLE, null, initialValues);
}

//---Hàm xóa nội dung (SV) xác định---
public boolean deleteStudent(long rowId)
{
    return db.delete(DATABASE_TABLE, KEY_ROWID + "=" + rowId, null) > 0;
}

//---Hàm lấy tất cả các nội dung (Danh sách SV)---
public Cursor getAllStudent()
{
    return db.query(DATABASE_TABLE, new String[] {KEY_ROWID, KEY_MSSV, KEY_NAME,
        KEY_EMAIL}, null, null, null, null, null);
}

//--- Hàm lấy một nội dung xác định (1 SV)---
public Cursor getStudent(int chiso) throws SQLException
{
    Cursor mCursor =
        db.query(true, DATABASE_TABLE, new String[] {KEY_ROWID, KEY_MSSV,
            KEY_NAME, KEY_EMAIL}, KEY_ROWID + "=" + chiso, null,
            null, null, null, null);
    if (mCursor != null) {

```

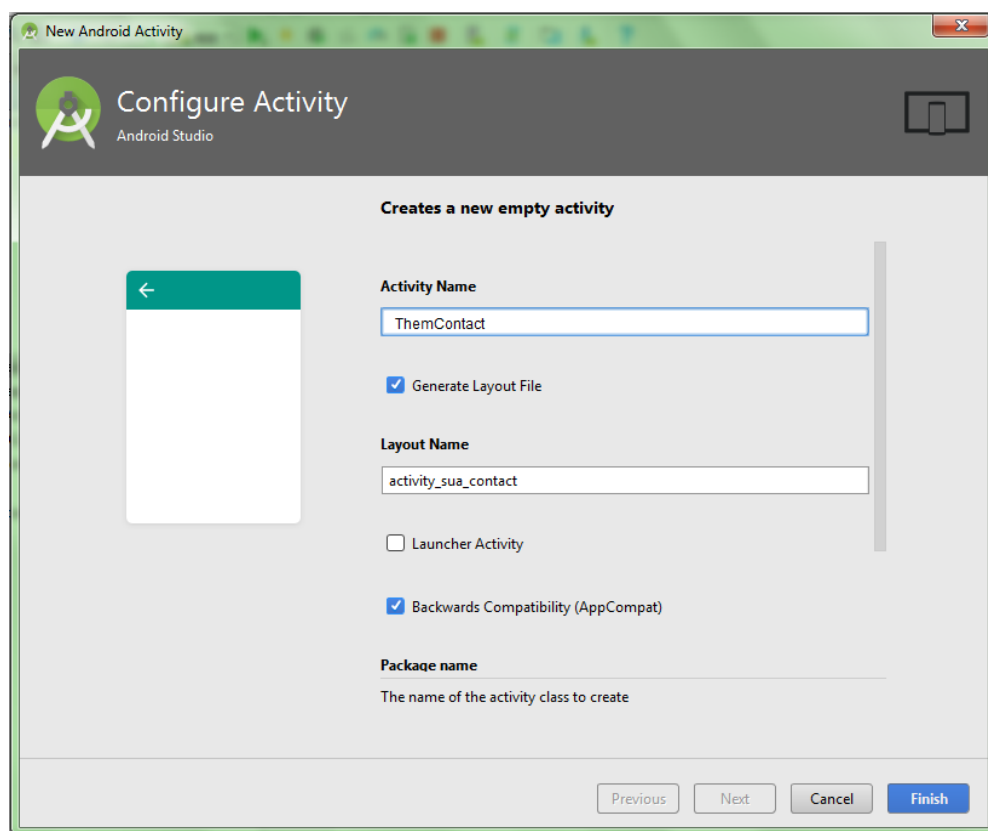
## Lập trình cho thiết bị di động

```
        mCursor.moveToFirst();
    }
    return mCursor;
}
//--- Hàm cập nhật một nội dung (sinh viên)---
public boolean updateStudent(long rowId, String mssv, String name, String email)
{
    ContentValues args = new ContentValues();
    args.put(KEY_NAME, mssv);
    args.put(KEY_NAME, name);
    args.put(KEY_EMAIL, email);
    return db.update(DATABASE_TABLE, args, KEY_ROWID+"="+rowId, null)>0;
}
}
```

**Bước 4:** Tạo 1 **Android Activity** có chức năng thêm sinh viên cho danh sách, có tên là **ThemSinh viên**.

**Cách làm:** Ấn chuột phải vào tên miền của ứng dụng → Chọn New → Chọn Activity → Chọn Empty Activity.

Cửa sổ “New Android Activity” xuất hiện (hình 4.5), nhập vào tên Activity là ThemSinh viên → Chọn OK



**Hình 4.5**

Lập trình cho thiết bị di động

Lập trình xây dựng Layout (*Tạm thời chưa cần lập trình Java cho activity này*). Giao diện như hình 4.4:



Hình 4.4

Sau đây là mã nguồn giao diện của Activity ThemSinh viên, sinh viên tham khảo để thực hiện giao diện cho activity này và các activity khác (không cung cấp mã nguồn layout nữa). Lưu ý: Trong tập tin xml này chưa khai báo EditText để nhập vào số điện thoại.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:orientation="vertical"
    tools:context=".ThemStudent">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:gravity="center_vertical|center_horizontal"
        android:padding="12dp"
        android:text="Hãy nhập vào thông tin để thêm một sinh viên"
        android:textColor="#00f"
        android:textSize="20sp" />
```



```
<EditText
    android:id="@+id/hoten"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Nhập vào họ và tên"
    android:textSize="18sp" />

<EditText
    android:id="@+id/mssv"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Nhập vào mã số SV"
    android:textSize="18sp" />

<EditText
    android:id="@+id/email"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Nhập vào địa chỉ email"
    android:textSize="18sp" />
<TableLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <TableRow>
        <Button
            android:id="@+id/btntrove"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:background="#D5D0D0"
            android:gravity="center_vertical|center_horizontal|end"
            android:text="TRỞ VỀ"
            android:textAlignment="center"
            android:textSize="18sp"
            android:textStyle="bold"
            android:visibility="visible" />
        <TextView
            android:layout_width="0dp"
            android:layout_weight="1"
            android:layout_height="wrap_content"/>
        <Button
            android:id="@+id/btnthem"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:background="#DBD1D1"
            android:gravity="center_vertical|center_horizontal|end"
            android:text="THÊM"
            android:textAlignment="center"
            android:textSize="18sp"
            android:textStyle="bold"
            android:visibility="visible" />
    </TableRow>
</TableLayout>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="Sinh viên đã được thêm vào"
```

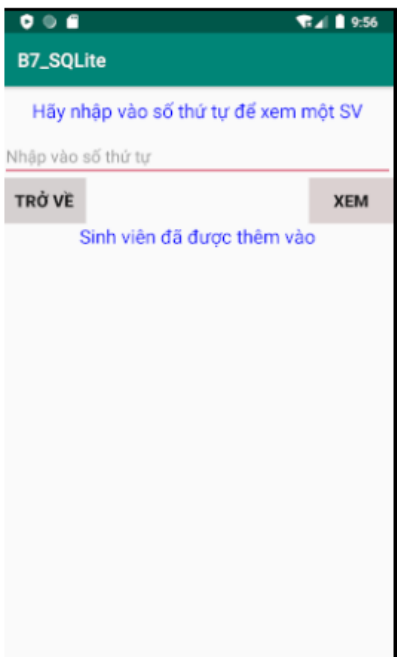
Lập trình cho thiết bị di động

```
        android:textColor="#f0f"
        android:textSize="20sp" />

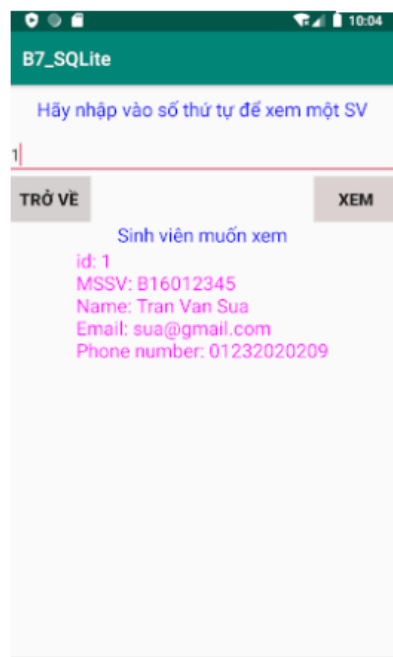
<TextView
    android:id="@+id/studentduocthem"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="start"
    android:text=""
    android:textColor="#0000ff"
    android:textSize="18sp" />

</LinearLayout>
```

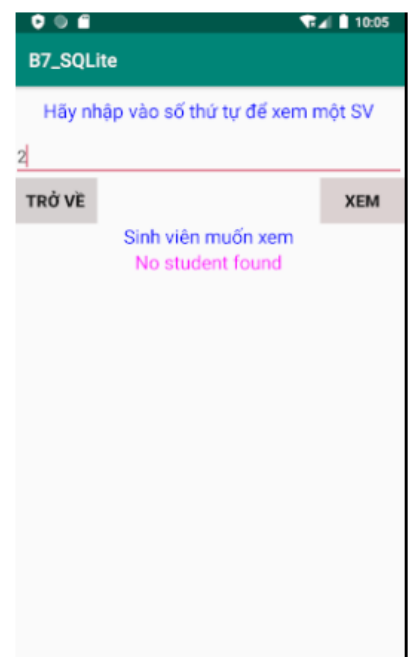
**Bước 5:** Tương tự, tạo 1 **Android Activity** có chức năng **xem 1 sinh viên** (với số thứ tự xác định) trong danh bạ, có tên là **Xem1Student**. (Trường hợp “Xem tất cả các sinh viên” được lập trình trực tiếp trong MainActivity).



**Hình 4.6.a.** Giao diện ban đầu



**Hình 4.6.b.** Nhấp buutton XEM và tìm thấy



**Hình 4.6.b.** Nhấp buutton XEM và không tìm thấy

## Hình 4.6

➔ Lập trình xây dựng Layout (Tạm thời chưa cần lập trình Java cho activity này).

Mã nguồn XML tham khảo (Các activity sau sẽ không cung cấp mã nguồn SV tham khảo các mã nguồn đã có để xây dựng).

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:orientation="vertical"
tools:context=".Xem1student">
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:gravity="center_vertical|center_horizontal"
    android:padding="12dp"
    android:text="Hãy nhập vào số thứ tự để xem một SV"
    android:textColor="@color/colorPrimary"
    android:textSize="20sp" />

<EditText
    android:id="@+id/stt"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:ems="10"
    android:hint="Nhập vào số thứ tự"
    android:textSize="18sp" />

<TableLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <TableRow>
        <Button
            android:id="@+id/btntrove"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:background="#D5D0D0"
            android:gravity="center_vertical|center_horizontal|end"
            android:text="TRỞ VỀ"
            android:textAlignment="center"
            android:paddingEnd="30dp"
            android:textSize="18sp"
            android:textStyle="bold"
            android:visibility="visible" />
        <TextView
            android:layout_width="0dp"
            android:layout_weight="1"
            android:layout_height="wrap_content"/>
        <Button
            android:id="@+id/btnxemmot"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:background="#DBD1D1"
            android:gravity="center_vertical|center_horizontal|end"
            android:text="XEM"
            android:textAlignment="center"
            android:paddingStart="30dp"
            android:textSize="18sp"
            android:textStyle="bold"
            android:visibility="visible" />
    </TableRow>
</TableLayout>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="Sinh viên muốn xem"

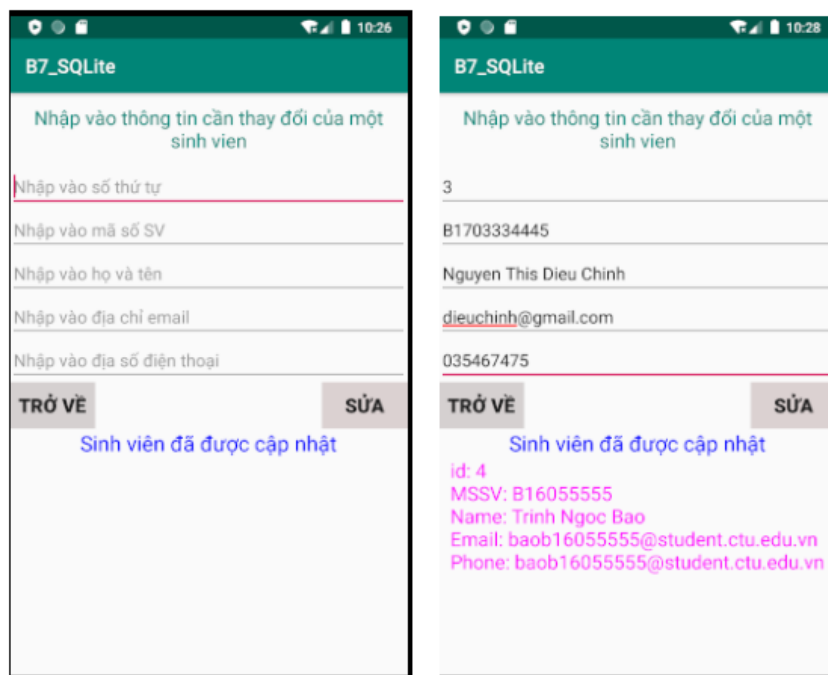
```

Lập trình cho thiết bị di động

```
        android:textColor="#00f"
        android:textSize="20sp" />
<TextView
    android:id="@+id/hienthi"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text=" "
    android:textColor="#f0f"
    android:textSize="20sp" />

</LinearLayout>
```

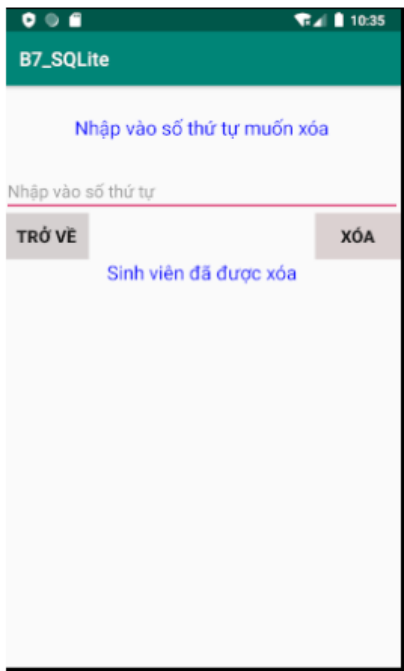
**Bước 6:** Tương tự, tạo 1 **Android Activity** có chức năng sửa 1 sinh viên (với số thứ tự xác định) trong danh bạ, có tên là **SuaStudent**. Giao diện ban đầu như hình 4.7.a:



Hình 4.7.a. Gia diện ban đầu      Hình 4.7.b. Sau khi nhập thông tin  
và nhấn Button SỬA  
Khi button TRỞ VỀ được nhấn thì trở về giao diện chính

Hình 4.7

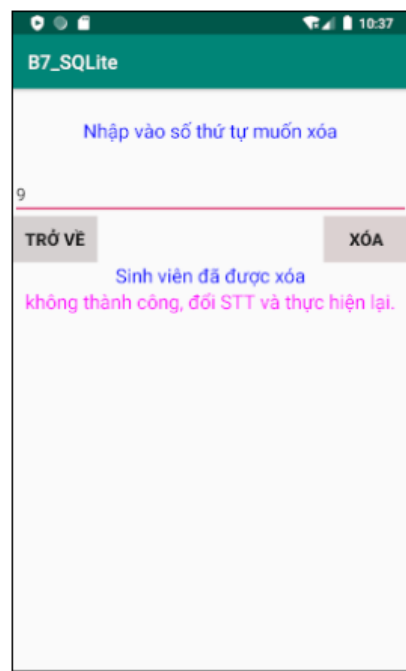
**Bước 7:** Tương tự, tạo 1 **Android Activity** có chức năng xóa 1 sinh viên (với số thứ tự xác định) trong danh bạ, có tên là **XoaStudent**. Giao diện như sau:



Hình 4.8.a. Giao diện ban đầu



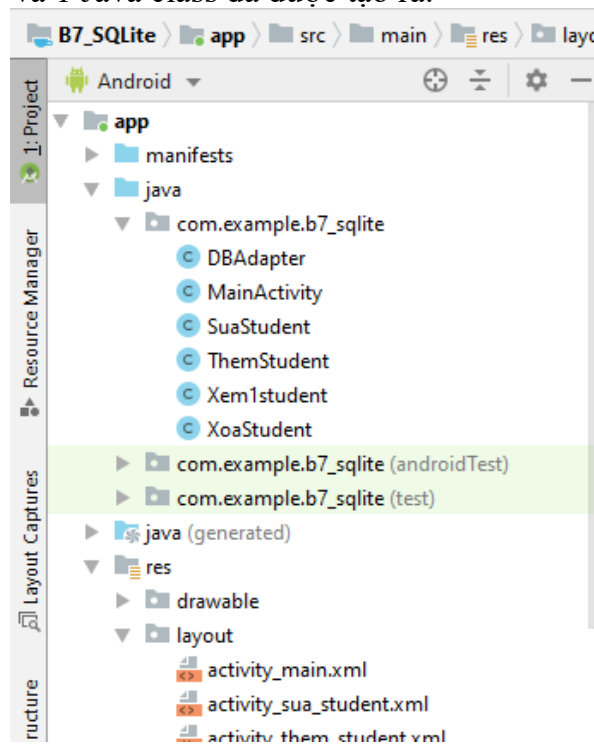
Hình 4.8.b. Nhập STT và xóa thành công



Hình 4.8.b. Nhập STT và xóa không thành công

## Hình 4.8

Cuối cùng, có 5 activity và 1 Java class đã được tạo ra.



## Hình 4.9

**Bước 7:** Lập trình đáp ứng tương tác người dùng cho **Activity chính**. Dùng Intent để gọi các Activity chức năng tương ứng với các button (Riêng button “XEM TẤT CẢ SINH VIÊN” được lập trình tương tác trực tiếp trên MainActivity). Chạy thử, lúc này khi một button được ấn chọn thì giao diện của nó sẽ xuất hiện. Tuy nhiên, các activity chức năng chưa được lập trình, nên chưa có đáp ứng.

Sau đây là mã nguồn của **Activity chính**, *sinh viên tham khảo, phân tích, hiểu các câu lệnh để có thể vận dụng sáng tạo*.

*Ghi chú: Trong các mã nguồn sau đây chưa có mã xử lý số điện thoại.*

```
package com.example.b7_sqlite;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.content.Intent;
import android.database.Cursor;
import android.view.Gravity;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import java.util.ArrayList;

public class MainActivity extends AppCompatActivity {
    DBAdapter db = new DBAdapter(this);
    ListView lv;
    ArrayList<String> arrList=null;
    ArrayAdapter<String> adapter=null;
    String msg;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button themsv = (Button) this.findViewById(R.id.btnthemsv);
        Button xemtatcasv = (Button) this.findViewById(R.id.btnxemtatca);
        Button xemmotstv = (Button) this.findViewById(R.id.btnxemmotstv);
        Button suamotstv = (Button) this.findViewById(R.id.btncapnhatsv);
        Button xoamotstv = (Button) this.findViewById(R.id.btnxoasv);
        Button back = (Button) this.findViewById(R.id.bnttrove);
        //---Thêm một sinh viên---
        themsv.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                //TODO Auto-generated method stub
                Intent it = new Intent(MainActivity.this, ThemStudent.class);
                startActivity(it);
            }
        });
        //---get all sinh viêns---
        lv=(ListView) findViewById(R.id.lvperson);
        //Tạo ArrayList object
        arrList=new ArrayList<String>();
        //Gán Data Source (ArrayList object) vào ArrayAdapter
        adapter=new ArrayAdapter<String>
            (this, android.R.layout.simple_list_item_1,arrList);
        //gán Adapter vào ListView
        lv.setAdapter(adapter);

        xemtatcasv.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
```

## Lập trình cho thiết bị di động

```
//TODO Auto-generated mehtod stub
db.open();
Cursor c = db.getAllStudent();
if (c.moveToFirst()) {
    do {
        String msg = "id: " + c.getString(0) + "\n" +
            "MSSV: " + c.getString(1) + "\n" +
            "Name: " + c.getString(2) + "\n" +
            "Email: " + c.getString(3);
        arrList.add(msg);
        adapter.notifyDataSetChanged();
        c.moveToNext();
    }
    //while (c.moveToNext());
    while (c.isAfterLast() == false);
}
db.close();
}
});
//---get a sinh vien---
xemmotstv.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        //TODO Auto-generated mehtod stub
        Intent it = new Intent(MainActivity.this, Xem1student.class);
        startActivity(it);
    }
});

// Sửa 1 sinh viên
suamotstv.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        //TODO Auto-generated mehtod stub
        Intent it = new Intent(MainActivity.this, SuaStudent.class);
        startActivity(it);
    }
});

// Xoá 1 sinh viên
xoamotstv.setOnClickListener(new View.OnClickListener() {
    public void onClick(View v) {
        //TODO Auto-generated mehtod stub
        Intent it = new Intent(MainActivity.this, XoaStudent.class);
        startActivity(it);
    }
});

//Trở về
back.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        //finish();
        Intent it = new Intent(MainActivity.this, MainActivity.class);
        startActivity(it);
    }
});
});
}
```

**Bước 8:**Lập trình cho Activity **ThemStudent**. Sau đây là mã nguồn của Activity này, sinh viên tham khảo, phân tích để có thể vận dụng sáng tạo.

```
package com.example.b7_sqlite;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.database.Cursor;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class ThemStudent extends AppCompatActivity {
    DBAdapter db = new DBAdapter(this);
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_them_student);
        //Thêm contact
        final EditText masosv=(EditText) this.findViewById(R.id.mssv); //Thêm
        final EditText ten=(EditText) this.findViewById(R.id.hoten);
        final EditText email=(EditText) this.findViewById(R.id.email);
        final TextView ct=(TextView) this.findViewById(R.id.studentduocthem);
        Button themcontact=(Button) this.findViewById(R.id.bntnthem);
        Button trove=(Button) this.findViewById(R.id.bntntrove);

        themcontact.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                db.open();
                String masosinhvien=masosv.getText().toString(); //Thêm
                String hoten=ten.getText().toString();
                String e_mail=email.getText().toString();
                long id = db.insertStudent(hoten,masosinhvien, e_mail); //Sửa
                // Hiện thị contact được thêm
                Cursor c = db.getAllStudent();
                c.moveToLast();
                //Có thêm
                String tx="id: " + c.getString(0) + "\n" +
                    "MSSV: " + c.getString(1) + "\n" +
                    "Name: " + c.getString(2) + "\n" +
                    "Email: " + c.getString(3);
                ct.setText(tx);
                db.close();
            }
        });
        trove.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new Intent(ThemStudent.this,MainActivity.class);
                startActivity(intent);
            }
        });
    }
}
```

**Bước 9:**Lập trình cho Activity **Xem1Student**. Sau đây là mã nguồn của Activity này, sinh viên tham khảo, phân tích để có thể vận dụng sáng tạo.



## Lập trình cho thiết bị di động

```
package com.example.b7_sqlite;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.database.Cursor;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class Xem1student extends AppCompatActivity {
    DBAdapter db = new DBAdapter(this);
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_xem1student);
        final EditText so=(EditText) this.findViewById(R.id.stt);
        final TextView ct=(TextView) this.findViewById(R.id.hienthi);
        Button xemsv=(Button) this.findViewById(R.id.btnxemsv);
        Button trove=(Button) this.findViewById(R.id.btntrove);
        xemsv.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                //TODO Auto-generated method stub
                db.open();
                int chiso=Integer.parseInt(so.getText().toString());
                Cursor cc = db.getStudent(chiso);
                //Có thêm
                if (cc.moveToFirst()) {
                    String tx="id: " + cc.getString(0) + "\n" +
                        "MSSV: " + cc.getString(1) + "\n" +
                        "Name: " + cc.getString(2) + "\n" +
                        "Email: " + cc.getString(3);
                    ct.setText(tx);
                }
                else {
                    String tx2="No student found";
                    ct.setText(tx2);
                    db.close();
                }
            }
        });
        trove.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent=new Intent(Xem1student.this,MainActivity.class);
                startActivity(intent);
            }
        });
    }
}
```

**Bước 10:**Lập trình cho Activity **SuaStudent**. Sau đây là mã nguồn của Activity này, sinh viên tham khảo, phân tích, hiểu các câu lệnh để có thể vận dụng sáng tạo.

```
package com.example.b7_sqlite;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
```

## Lập trình cho thiết bị di động

```
import android.database.Cursor;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class SuaStudent extends AppCompatActivity {
    DBAdapter db = new DBAdapter(this);
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sua_student);
        final EditText sott=(EditText) this.findViewById(R.id.stt); //Thêm
        final EditText masosv=(EditText) this.findViewById(R.id.mssv); //Thêm
        final EditText ten=(EditText) this.findViewById(R.id.hoten);
        final EditText email=(EditText) this.findViewById(R.id.email);
        final TextView ct=(TextView) this.findViewById(R.id.studentduocthem);
        Button suacontact=(Button) this.findViewById(R.id.btnsua);
        Button trove=(Button) this.findViewById(R.id.btntrove);
        suacontact.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                db.open();
                String masosinhvien=masosv.getText().toString();
                String tensv=ten.getText().toString();
                String e_mail=email.getText().toString();
                int chiso=Integer.parseInt(sott.getText().toString());
                //---update contact---
                if (db.updateStudent(chiso,masosinhvien, tensv,e_mail,sodienthoai))
                    ct.setText("thành công.");
                else
                    ct.setText("không thành công, đây là giá trị cũ, vui lòng thực hiện
lại lần nữa");

                // Hiển thị contact được sửa
                Cursor c = db.getAllStudent();
                c.moveToPosition(chiso-1);
                String tx="id: " + c.getString(0) + "\n" +
                    "MSSV: " + c.getString(1) + "\n" +
                    "Name: " + c.getString(2) + "\n" +
                    "Email: " + c.getString(3);
                ct.setText(tx);
                db.close();
            }
        });

        trove.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent3=new Intent(SuaStudent.this,MainActivity.class);
                startActivity(intent3);
            }
        });
    }
}
```

Chạy thử, nhập vào số sinh viên và các thông tin muốn sửa và ấn button SỬA. Hình 4.12 là kết quả hiển thị sinh viên đã sửa thành công (nếu thất bại thì có thông báo sửa thất bại).

**Bước 11:** Lập trình cho Activity **XoaStudent**. Sau đây là mã nguồn của Activity này, sinh viên tham khảo, phân tích, hiểu các câu lệnh để có thể vận dụng sáng tạo.

## Lập trình cho thiết bị di động

```
package com.example.b7_sqlite;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;

public class XoaStudent extends AppCompatActivity {
    DBAdapter db = new DBAdapter(this);
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_xoa_student);
        final EditText so=(EditText) this.findViewById(R.id.sttxoa);
        final TextView ct=(TextView) this.findViewById(R.id.hienthi);
        Button xoacontact=(Button) this.findViewById(R.id.btnxoasv);
        Button btntrove=(Button) this.findViewById(R.id.btntrove);
        xoacontact.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                db.open();
                int chiso=Integer.parseInt(so.getText().toString());
                //---delete a contact---
                if (db.deleteStudent(chiso))
                    ct.setText("thành công.");
                else
                    ct.setText("không thành công, đổi STT và thực hiện lại.");
                db.close();
            }
        });
        btntrove.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent intent4=new Intent(XoaStudent.this,MainActivity.class);
                startActivity(intent4);
            }
        });
    }
}
```

Chạy thử, nhập vào số sinh viên muốn xóa và ấn button XOÁ, kết quả hiển thị sinh viên đã xóa thành công (nếu thất bại thì có thông báo xóa thất bại).

## BÀI 8: TẠO RA MỘT CONTENT PROVIDER

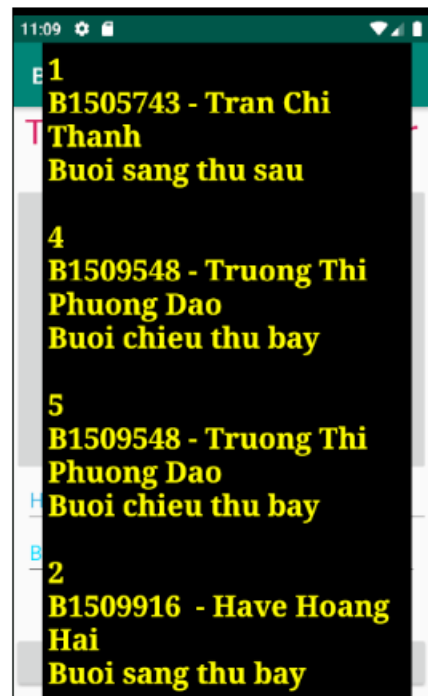
Chúng ta sẽ tạo một lớp Content Provider trong ứng dụng dùng cơ sở dữ liệu SQLite để lưu trữ dữ liệu là Danh sách sinh viên các buổi thực hành (Tuy rằng có thể dùng bất kỳ cách lưu trữ nào khác mà ta muốn). CSDL này có thể dùng chung cho các ứng dụng khác. Các bước thực hành như sau:

Bước	Miêu tả
1	Bạn sử dụng Android Studio IDE để tạo một ứng dụng Android có

	tên <i>B4_2_MyContentProvider</i> với Activity chính là <i>MainActivity.java</i> , layout chính là <i>activity_main.xml</i> .
2	Tạo một lớp ContentProvider mới có tên là <i>StudentsProvider.java</i> dưới Package name của project và định nghĩa các phương thức được gắn kết. <b>Định nghĩa Uri bằng cách lấy Package name thay tên project bằng tên provider.</b> Ví dụ: Package name là: “ <b>com.vn.doanhoaminh.b4_2_mycontentprovider_2</b> ” Thì Uri là: “ <b>com.vn.doanhoaminh.provider.College</b> ”
3	Sửa đổi nội dung mặc định của <i>res/layout/activity_main.xml</i> để tạo một GUI cho phép thêm vào bản ghi vào <b>tên SV</b> và <b>buổi thực hành</b> . (Hình 4.14)
4	Sửa đổi <i>MainActivity.java</i> để thêm hai phương thức mới <i>onClickAddName()</i> (thêm SV vào danh sách thực hành) và <i>onClickRetrieveStudents()</i> (xem danh sách).
5	<b>Đăng ký</b> Content Provider trong tập tin <i>AndroidManifest.xml</i>
6	Không cần thiết sửa đổi string.xml file
7	Chạy thử ứng dụng với Emulator: Nhập tên SV và Buổi thực hành vào EditText tương ứng, chọn button THÊM và kiểm tra kết quả bằng cách chọn button XEM (Hình 4.10 và Hình 4.11)



**Hình 10 .** Giao diện chính



**Hình 11.** Sau khi thêm tên và chọn XEM

Sau đây là mã nguồn của các tập tin java và xml. Sinh viên tham khảo, phân tích, hiểu ý nghĩa và cách dùng để có thể vận dụng sáng tạo vào các ứng dụng khác và bài thi.

- 1) Lớp tạo Content Provider “**StudentsProvider**” kế thừa lớp **ContentProvider** và ghi đề lên các phương thức như sau.

```
package com.example.b8_mycontentprovider;

import android.content.ContentProvider;
import android.net.Uri;

import android.content.ContentProvider;
import java.util.HashMap;
import android.content.ContentUris;
import android.content.ContentValues;
import android.content.Context;
import android.content.UriMatcher;
import android.database.Cursor;
import android.database.SQLException;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.database.sqlite.SQLiteQueryBuilder;

import android.net.Uri;
import android.text.TextUtils;

public class StudentsProvider extends ContentProvider {
    static final String PROVIDER_NAME = "com.vn.doanhoaminh.provider.College";
    static final String URL = "content://" + PROVIDER_NAME + "/students";
    static final Uri CONTENT_URI = Uri.parse(URL);

    static final String _ID = "_id";
    static final String NAME = "ten";
    static final String BUOI = "buoi";

    // Khai báo 1 bảng băm
    private static HashMap<String, String> STUDENTS_PROJECTION_MAP;

    static final int STUDENTS = 1;
    static final int STUDENT_ID = 2;

    // Tạo 1 đối tượng tiện ích để hỗ trợ kết hợp các URI trong các Content provider
    // SV tìm hiểu thêm về HashMap và UriMatcher.
    static final UriMatcher uriMatcher;
    static{
        uriMatcher = new UriMatcher(UriMatcher.NO_MATCH);
        uriMatcher.addURI(PROVIDER_NAME, "students", STUDENTS);
        uriMatcher.addURI(PROVIDER_NAME, "students/#", STUDENT_ID);
    }

    /**
     * Khai báo các hằng đặc tả CSDL
     */
    private SQLiteDatabase db;
    static final String DATABASE_NAME = "College";
    static final String STUDENTS_TABLE_NAME = "students";
    static final int DATABASE_VERSION = 1;
    static final String CREATE_DB_TABLE =
        " CREATE TABLE " + STUDENTS_TABLE_NAME +
        " ( _id INTEGER PRIMARY KEY AUTOINCREMENT, " +
        " ten TEXT NOT NULL, " +
        " buoi TEXT NOT NULL );";
}
```

```

/**
 * Tạo đối tượng SQLiteOpenHelper để tạo ra và quản lý CSDL của Provider.
 */
private static class DatabaseHelper extends SQLiteOpenHelper {
    DatabaseHelper(Context context){
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }

    @Override
    public void onCreate(SQLiteDatabase db)
    {
        db.execSQL(CREATE_DB_TABLE);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS " + STUDENTS_TABLE_NAME);
        onCreate(db);
    }
}

@Override
public boolean onCreate() {
    Context context = getContext();
    DatabaseHelper dbHelper = new DatabaseHelper(context);

    /**
     * Create a write able database which will trigger its
     * creation if it doesn't already exist.
     */
    db = dbHelper.getWritableDatabase();
    return (db == null)? false:true;
}

@Override
public Uri insert(Uri uri, ContentValues values) {
    /**
     * Add a new student record
     */
    long rowID = db.insert(STUDENTS_TABLE_NAME, "", values);

    /**
     * If record is added successfully
     */

    if (rowID > 0)
    {
        Uri _uri = ContentUris.withAppendedId(CONTENT_URI, rowID);
        getContext().getContentResolver().notifyChange(_uri, null);
        return _uri;
    }
    throw new SQLException("Failed to add a record into " + uri);
}

@Override
public Cursor query(Uri uri, String[] projection, String selection, String[]
selectionArgs, String sortOrder) {
    SQLiteQueryBuilder qb = new SQLiteQueryBuilder();
    qb.setTables(STUDENTS_TABLE_NAME);

    switch (uriMatcher.match(uri)) {
        case STUDENTS:
            qb.setProjectionMap(STUDENTS_PROJECTION_MAP);
            break;
    }
}

```

```

        case STUDENT_ID:
            qb.appendWhere( _ID + "=" + uri.getPathSegments().get(1));
            break;

        default:
            throw new IllegalArgumentException("Unknown URI " + uri);
    }

    if (sortOrder == null || sortOrder == ""){
        /**
         * By default sort on student names
         */
        sortOrder = NAME;
    }
    Cursor c = qb.query(db, projection, selection,
        selectionArgs,null, null, sortOrder);

    /**
     * register to watch a content URI for changes
     */
    c.setNotificationUri(getContext().getContentResolver(), uri);
    return c;
}

@Override
public int delete(Uri uri, String selection, String[] selectionArgs) {
    int count = 0;

    switch (uriMatcher.match(uri)){
        case STUDENTS:
            count = db.delete(STUDENTS_TABLE_NAME, selection, selectionArgs);
            break;

        case STUDENT_ID:
            String id = uri.getPathSegments().get(1);
            count = db.delete( STUDENTS_TABLE_NAME, _ID + " = " + id +
                (!TextUtils.isEmpty(selection) ? " AND (" + selection + ')' : ""),
                selectionArgs);
            break;

        default:
            throw new IllegalArgumentException("Unknown URI " + uri);
    }

    getContext().getContentResolver().notifyChange(uri, null);
    return count;
}

@Override
public int update(Uri uri, ContentValues values, String selection, String[]
selectionArgs) {
    int count = 0;

    switch (uriMatcher.match(uri)){
        case STUDENTS:
            count = db.update(STUDENTS_TABLE_NAME, values, selection,
selectionArgs);
            break;

        case STUDENT_ID:
            count = db.update(STUDENTS_TABLE_NAME, values, _ID + " = " +
uri.getPathSegments().get(1) +
                (!TextUtils.isEmpty(selection) ? " AND (" +selection + ')' :
""), selectionArgs);

```

```
        break;

        default:
            throw new IllegalArgumentException("Unknown URI " + uri );
    }
    getContext().getContentResolver().notifyChange(uri, null);
    return count;
}

@Override
public String getType(Uri uri) {
    switch (uriMatcher.match(uri)) {
        /**
         * Get all student records
         */
        case STUDENTS:
            return "vnd.android.cursor.dir/vnd.example.students";

        /**
         * Get a particular student
         */
        case STUDENT_ID:
            return "vnd.android.cursor.item/vnd.example.students";
        default:
            throw new IllegalArgumentException("Unsupported URI: " + uri);
    }
}
}
```

## 2) Layout XML

```
<?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"
    tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Tạo một Content provider"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:textColor="@color/colorAccent"
        android:textSize="30dp" />

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="DANH SÁCH SV THỰC HÀNH"
        android:textColor="#0000ff"
        android:textSize="20dp"
        android:layout_below="@+id/textView1"
        android:layout_centerHorizontal="true" />

    <ImageButton
        android:layout_width="match_parent"
```



```
        android:layout_height="wrap_content"
        android:id="@+id/imageButton"
        android:src="@drawable/hoso"
        android:layout_below="@+id/textView2"
        android:layout_centerHorizontal="true" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/editText"
    android:layout_below="@+id/imageButton"
    android:layout_alignRight="@+id/imageButton"
    android:layout_alignEnd="@+id/imageButton" />

<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/editText2"
    android:layout_alignTop="@+id/editText"
    android:layout_alignLeft="@+id/textView1"
    android:layout_alignStart="@+id/textView1"
    android:layout_alignRight="@+id/textView1"
    android:layout_alignEnd="@+id/textView1"
    android:hint="Họ và tên"
    android:textColorHint="@android:color/holo_blue_light" />

<EditText
    android:layout_width="300dp"
    android:layout_height="wrap_content"
    android:id="@+id/editText3"
    android:layout_below="@+id/editText"
    android:layout_alignLeft="@+id/editText2"
    android:layout_alignStart="@+id/editText2"
    android:layout_alignRight="@+id/editText2"
    android:layout_alignEnd="@+id/editText2"
    android:hint="Buổi thực hành"
    android:textColorHint="@android:color/holo_blue_bright" />

<Button
    android:layout_width="300dp"
    android:layout_height="wrap_content"
    android:id="@+id/button2"
    android:text="THÊM TÊN"
    android:layout_below="@+id/editText3"
    android:layout_alignRight="@+id/textView2"
    android:layout_alignEnd="@+id/textView2"
    android:layout_alignLeft="@+id/textView2"
    android:layout_alignStart="@+id/textView2"
    android:onClick="onClickAddName" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="XEM"
    android:id="@+id/button"
    android:layout_below="@+id/button2"
    android:layout_alignParentLeft="true"
    android:layout_alignParentRight="true"
    android:onClick="onClickRetrieveStudents" />

</RelativeLayout>
```

### 3) Tập tin MainActivity.java

## Lập trình cho thiết bị di động

```
package com.example.b8_mycontentprovider;

import androidx.appcompat.app.AppCompatActivity;

import android.graphics.Color;
import android.graphics.Typeface;
import android.os.Bundle;
import android.net.Uri;
import android.content.ContentValues;
import android.content.CursorLoader;
import android.database.Cursor;
import android.view.Gravity;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

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

    public void onClickAddName(View view) {
        // Add a new student record
        ContentValues values = new ContentValues();

        values.put (StudentsProvider.NAME, ((EditText) findViewById(R.id.editText2)).getText().toString());

        values.put (StudentsProvider.BUOI, ((EditText) findViewById(R.id.editText3)).getText().toString());

        Uri uri = getContentResolver().insert(StudentsProvider.CONTENT_URI, values);

        Toast.makeText(getApplicationContext(),uri.toString(), Toast.LENGTH_LONG).show();
    }

    public void onClickRetrieveStudents(View view) {
        Uri students = StudentsProvider.CONTENT_URI;
        String[] projection = {StudentsProvider._ID, StudentsProvider.NAME, StudentsProvider.BUOI};
        CursorLoader loader = new CursorLoader(this,students, projection,null,null,null);
        Cursor c = loader.loadInBackground();
        c.moveToFirst();
        String s = "";
        while (!c.isAfterLast()){
            for (int i=0; i<c.getColumnCount();i++){
                s+= c.getString(i)+"\n";
            }
            s+="\n";
            c.moveToNext();
        }
        displayToast(s);
        c.close();
    }

    private void displayToast(String msg) {
        Toast toast = new Toast(this);
        TextView tv4 = new TextView(this);
        tv4.setBackgroundColor(Color.BLACK);
        tv4.setTextColor(Color.YELLOW);
    }
}
```

```
tv4.setTextSize(25);
Typeface t4 = Typeface.create("serif", Typeface.BOLD);
tv4.setTypeface(t4);
tv4.setPadding(10, 10, 10, 10);
tv4.setText(msg);
toast.setGravity(Gravity.CENTER, 10, 100);
toast.setView(tv4);
toast.setDuration	Toast.LENGTH_LONG;
toast.show();
}
}
```

#### 4) Thêm vào Provider trong tập tin AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.b8_mycontentprovider">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

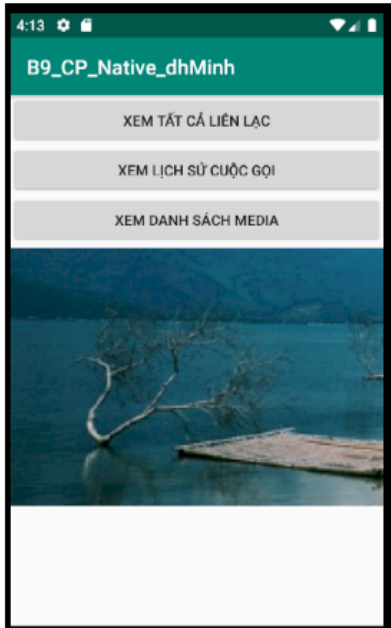
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <provider
            android:authorities="com.vn.doanhoaminh.provider.College"
            android:name="StudentsProvider">
        </provider>

    </application>

</manifest>
```

## BÀI 9: TRUY XUẤT DỮ LIỆU VỚI CONTENT PROVIDERS DỰNG SẴN

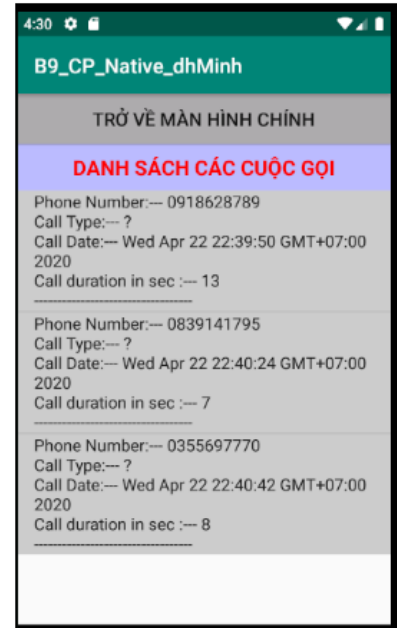
Xây dựng một ứng dụng hiển Contact, Call log và Media trên một điện thoại di động. Giao diện gồm 3 button như hình 4.12, khi người dùng muốn xem nội dung nào thì chọn button đó, chẳng hạn như khi chọn button “XEM TẤT CẢ LIÊN LẠC” thì sẽ mở một giao diện mới hiển thị danh sách các Contact được lưu trên danh bạ điện thoại như hình 4.13, trên màn hình này, nút “TRỞ VỀ MÀN HÌNH CHÍNH” được nhấp để trở về màn hình chính hình 4.12. Nếu nhấp nút “XEM LỊCH SỬ CUỘC GỌI” thì mở ra giao diện mới để hiển thị lịch sử cuộc gọi như Hình 4.14, sau đó nhấp nút “TRỞ VỀ MÀN HÌNH CHÍNH” để chọn “XEM DANH SÁCH MEDIA” (*Phần xem danh sách media, gồm music, video, picture được lưu trong thẻ nhớ không bắt buộc, vì với mã nguồn như đã lập trình học kỳ rồi, bây giờ trong phiên bản mới nó không lấy được danh sách, thầy chưa sửa được*).



Hình 4.13



Hình 4.13



Hình 4.14

### Các bước thực hiện:

1. Hãy mở project mới (trong ví dụ này có tên là B9\_MSSV\_Tên SV), với activity chính là **MainAcvtivity.java** và layout là **activity\_main.xml**.
2. Sửa lại tập tin **activity\_main.xml** để có giao diện như Hình 4.13. (ImageView hiển thị hình do SV tùy chọn, thầy không cho code xml).
3. Phải thêm các permission: **READ\_CONTACTS**, **READ\_CALL\_LOG**, **READ\_EXTERNAL\_STORAGE**, **WRITE\_EXTERNAL\_STORAGE** vào tập tin AndroidManifest.xml (các lệnh được highlight):

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.b9_cp_native_dhminh">

    <uses-permission android:name="android.permission.READ_CONTACTS" />
    <uses-permission android:name="android.permission.READ_CALL_LOG" />
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".DisplayMediaList"></activity>
        <activity android:name=".DisplayAllContact" />
        <activity android:name=".DisplayAllCallLog" />
        <activity android:name=".MainActivity"></activity>
    </application>
</manifest>
```

```
<intent-filter>
    <action android:name="android.intent.action.MAIN" />

    <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
</application>

</manifest>
```

4. Lần lượt tạo 3 activity mới để thực hiện các chức năng khi các button “XEM TẤT CẢ LIÊN LẠC”, “ XEM LỊCH SỬ CUỘC GỌI” và “ XEM DANH SÁCH MEDIA” được nhấp chọn. Trong ví dụ này, các activity lần lượt được đặt tên là: **DisplayAllContact**, **DisplayAllCallLog** và **DisplayMediaList** (Nhấp chuột phải vào thư mục tên miền, chọn New → Chọn Activity → Chọn Empty Activity → Nhập tên Activity → Chọn Finish. Xem hình 4.15).



Hình 4.15

5. Lập trình layout cho 3 activity vừa tạo ra. Sau đây là tập tin xml mẫu, layout của activity DisplayAllContact, tương tự cho các layout khác.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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:orientation="vertical"
    tools:context=".DisplayAllContact">
    <Button
        android:id="@+id/btntrove"
        android:layout_width="match_parent"
        android:layout_height="wrap_content">
```

```
        android:gravity="center"
        android:background="#ADABAD"
        android:textSize="18sp"
        android:text="TRỞ VỀ MÀN HÌNH CHÍNH" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="45dp"
    android:background="#bbbbbf"
    android:gravity="center_vertical|center_horizontal"
    android:text="DANH SÁCH CÁC LIÊN LẠC"
    android:textAllCaps="true"
    android:textColor="#ff0000"
    android:textSize="20sp"
    android:textStyle="bold" />

<ListView
    android:id="@+id/lvcontact"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#cccccc">
</ListView>

</LinearLayout>
```

## 6. Lập trình MainActivity.java và chạy thử các layout. Sau đây là mã nguồn tham khảo.

```
package com.example.b9_cp_native_dhminh;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.widget.Button;
import android.content.Intent;
import android.view.View;

public class MainActivity extends AppCompatActivity {
    private Button btgetAllCallLogs, btgetAllContacts, btgetAllMusic;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // link button den layout de dieu khien
        btgetAllCallLogs = (Button) findViewById(R.id.btgetAllCallLogs);
        btgetAllContacts = (Button) findViewById(R.id.btgetAllContacts);
        btgetAllMusic = (Button) findViewById(R.id.btgetAllMusic);

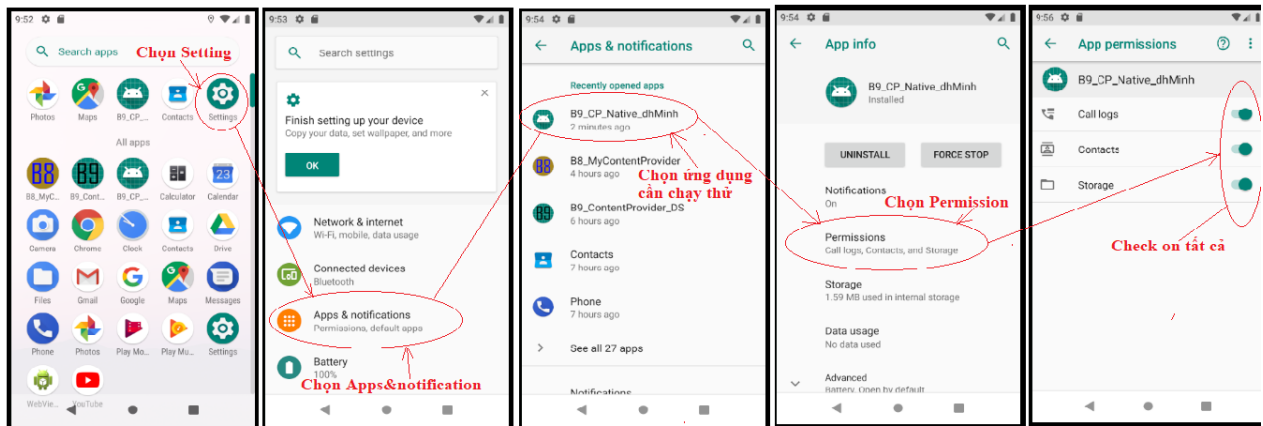
        // dang ky lang nghe su kien onclick vao cac button
        btgetAllCallLogs.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                //TODO Auto-generated method stub
                Intent intent1 = new Intent(MainActivity.this,
                    DisplayAllCallLog.class);
                startActivity(intent1);
            }
        });
        btgetAllContacts.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                //TODO Auto-generated method stub
                Intent intent2 = new Intent(MainActivity.this,
                    DisplayAllContact.class);
                startActivity(intent2);
            }
        });
    }
}
```

## Lập trình cho thiết bị di động

```
    }  
    });  
    btgetAllMusic.setOnClickListener(new View.OnClickListener() {  
        public void onClick(View v) {  
            //TODO Auto-generated method stub  
            Intent intent3 = new Intent(MainActivity.this,  
                DisplayMediaList.class);  
            startActivity(intent3);  
        }  
    });  
}  
}
```

7. Lập trình activity **DisplayAllContact.java**. Sau đây là tập tin mã nguồn mẫu. SV tham khảo, tìm hiểu chức năng của các câu lệnh, chạy thử trên thiết bị giả lập.

Để chạy thử ứng dụng này ta phải *thực hiện permission trên máy ảo*, các bước như Hình 4.16, và *phải thêm vào danh bạ một số liên lạc* (Trên máy ảo, chọn Contacts và thêm các số liên lạc vào, họ tên và điện thoại).



Hình 4.16. Các bước thực hiện check permission trên emulator

Layout XML:

```
<?xml version="1.0" encoding="utf-8" ?>  
<LinearLayout  
    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:orientation="vertical"  
    tools:context=".DisplayAllCallLog">  
    <Button  
        android:id="@+id/btntrove2"  
        android:layout_width="match_parent"  
        android:layout_height="wrap_content"  
        android:gravity="center"  
        android:background="#ADABAD"
```



```

        android:textSize="18sp"
        android:text="TRỞ VỀ MÀN HÌNH CHÍNH" />

<TextView
    android:layout_width="match_parent"
    android:layout_height="45dp"
    android:background="#bbbbff"
    android:gravity="center_vertical|center_horizontal"
    android:text="DANH SÁCH CÁC CUỘC GỌI"
    android:textAllCaps="true"
    android:textColor="#ff0000"
    android:textSize="20sp"
    android:textStyle="bold" />

<ListView
    android:id="@+id/lvcallLog"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#cccccc">
</ListView>

</LinearLayout>

```

### DisplayAllContact.java

```

package com.example.b9_cp_native_dhminh;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;
import android.content.ContentResolver;
import android.database.Cursor;
import android.provider.ContactsContract;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import java.util.ArrayList;

public class DisplayAllContact extends AppCompatActivity {
    Button back;
    ListView lv;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_display_all_contact);
        back = (Button) findViewById(R.id.btntrove);
        lv = (ListView) findViewById(R.id.lvcontact);
        ArrayList<String> list=new ArrayList<String>();
        ContentResolver cr = getContentResolver();
        ArrayAdapter<String> adapter=new ArrayAdapter<String>(this,

```



```

        android.R.layout.simple_list_item_1, list);
lv.setAdapter(adapter);
// lay toan bo danh ba
    Cursor cur = cr.query(ContactsContract.Contacts.CONTENT_URI,
null,
        null, null, null);
    if (cur.getCount() > 0) {
        while (cur.moveToNext()) {
            // get id va name cua tung nguoi trong danh ba
            String id = cur.getString(cur
                .getColumnIndex(ContactsContract.Contacts._ID));
            String name = cur
                .getString(cur
                    .getColumnIndex(ContactsContract.Contacts.DISPLAY_NAME));
            if (Integer
                .parseInt(cur.getString(cur
                    .getColumnIndex(ContactsContract.Contacts.HAS_PHONE_NUMBER))) > 0) {
                // lay tat ca cac sdt cua ng co _id = id
                Cursor pCur = cr.query(
                    ContactsContract.CommonDataKinds.Phone.CONTENT_URI,
                        null,
                            ContactsContract.CommonDataKinds.Phone.CONTACT_ID
                                + " = ?", new String[] { id },
                                null);
                while (pCur.moveToNext()) {
                    String phoneNo = pCur
                        .getString(pCur
                            .getColumnIndex(ContactsContract.CommonDataKinds.Phone.NUMBER));
                    // log toan bo so dien thoai ra logcat
                    String contact = "Name: " + name
                        + "|| Phone No: " + phoneNo
                        + "\n-----";
                    list.add(contact);
                    adapter.notifyDataSetChanged();
                }
                pCur.close();
            }
        }
    }
    cur.close();

    back.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            finish();
        }
    });

```

```
}
}
```

8. Lập trình activity **DisplayAllCallLog.java**. Sau đây là tập tin mã nguồn mẫu. SV tham khảo, tìm hiểu chức năng của các câu lệnh, chạy thử. **Trước khi chạy thử ta cần thực hiện một số cuộc gọi trên máy ảo.**

```
package com.example.b9_cp_native_dhminh;

import androidx.appcompat.app.AppCompatActivity;

import android.Manifest;
import android.os.Bundle;
import android.annotation.SuppressLint;
import android.content.ContentResolver;
import android.content.pm.PackageManager;
import android.database.Cursor;
import android.provider.CallLog;
import android.view.View;
import android.widget.AdapterView;
import android.widget.Button;
import android.widget.ListView;
import java.util.ArrayList;
import java.util.Date;

public class DisplayAllCallLog extends AppCompatActivity {
    Button back2;
    ListView lvcallLog;

    @SuppressWarnings("NewApi")
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_display_all_call_log);
        back2 = (Button) findViewById(R.id.btntrove2);
        lvcallLog = (ListView) findViewById(R.id.lvcallLog);

        ArrayList<String> list2 = new ArrayList<String>();
        ContentResolver cr2 = getContentResolver();
        ArrayAdapter<String> adapter2 = new ArrayAdapter<String>(this,
            android.R.layout.simple_list_item_1, list2);
        lvcallLog.setAdapter(adapter2);

        // Lấy lịch sử cuộc gọi
        String[] projection = new String[]{CallLog.Calls.DATE, CallLog.Calls.NUMBER,
            CallLog.Calls.DURATION};

        if (checkSelfPermission(Manifest.permission.READ_CALL_LOG) !=
            PackageManager.PERMISSION_GRANTED) {
            // TODO: Consider calling
            //    Activity#requestPermissions
            // here to request the missing permissions, and then overriding
            // public void onRequestPermissionsResult(int requestCode, String[]
permissions,
            //                                     int[] grantResults)
            // to handle the case where the user grants the permission. See the
documentation
            // for Activity#requestPermissions for more details.
            return;
        }
        Cursor managedCursor = getContentResolver().query(CallLog.Calls.CONTENT_URI,
```

## Lập trình cho thiết bị di động

```
projection,
    CallLog.Calls.DURATION + "<?", new String[]{"30"},
    CallLog.Calls.DATE + " Asc"); // Asc: sắp theo thứ tự tăng dần

int number = managedCursor.getColumnIndex(CallLog.Calls.NUMBER);
int type = managedCursor.getColumnIndex(CallLog.Calls.TYPE);
int date = managedCursor.getColumnIndex(CallLog.Calls.DATE);
int duration = managedCursor.getColumnIndex(CallLog.Calls.DURATION);

while (managedCursor.moveToNext()) {
    // lấy các cột trong cursor
    String phNumber = managedCursor.getString(number);
    String callDate = managedCursor.getString(date);
    Date callDayTime = new Date(Long.valueOf(callDate));
    String callDuration = managedCursor.getString(duration);

    // hiển thị toàn bộ lịch sử cuộc gọi
    String cuocgoi = "Phone Number:--- " + phNumber
        + " \nCall Type:--- " + "?" + " \nCall Date:--- "
        + callDayTime + " \nCall duration in sec :--- "
        + callDuration + "\n-----";

    list2.add(cuocgoi);
    adapter2.notifyDataSetChanged();
}
managedCursor.close();

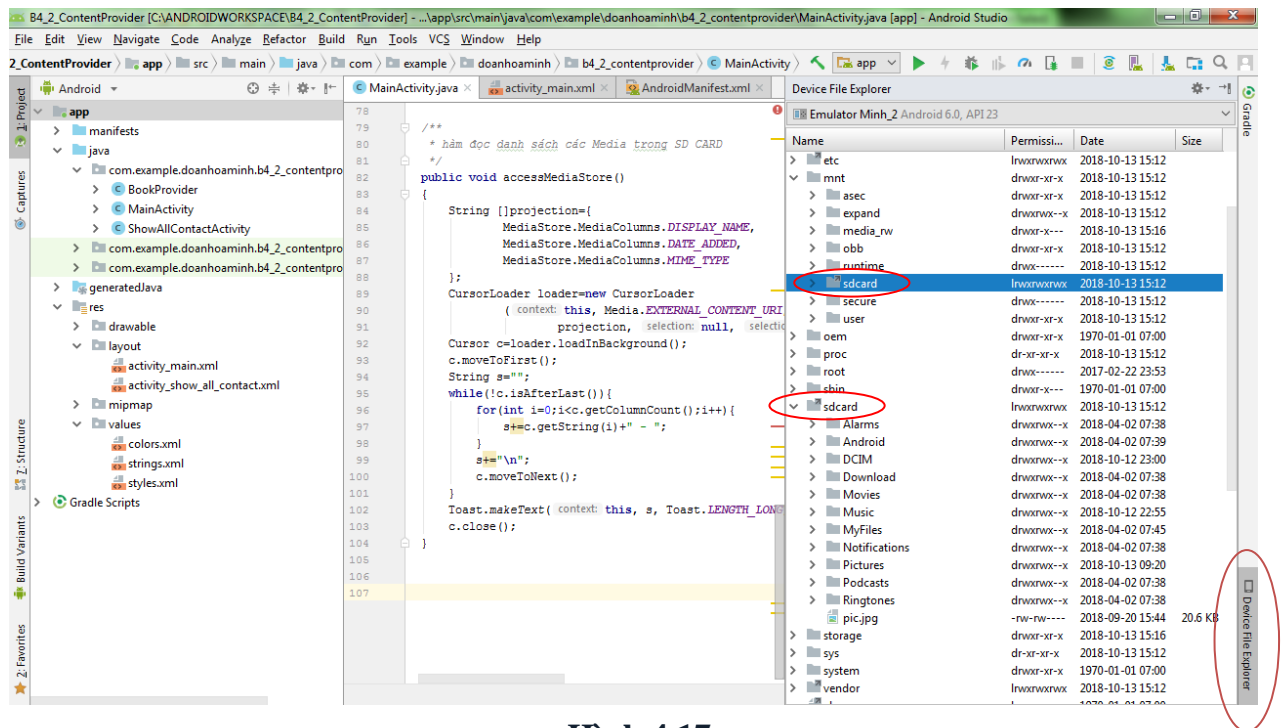
back2.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        finish();
    }
});
}
```

9. Lập trình activity **DisplayMediaList.java**. Sau đây là các tập tin mã nguồn mẫu. SV tham khảo, tìm hiểu chức năng của các câu lệnh, chạy thử trên thiết bị giả lập.

Trước khi chạy thử ứng dụng, chúng ta phải nhập một số tập tin hình ảnh/âm nhạc vào các thư mục tương ứng trên SDCard của máy ảo, để chuẩn bị danh sách hiển thị. Các bước thực hiện:

- ➔ Trên Android studio, nhấp chọn “Device File Explorer” ở góc phải phía dưới của màn hình, cửa sổ “Device File Explorer” hiện ra (Hình 4.17). Ta thấy thư mục **SDCard** nằm ở 2 vị trí, trong thư mục **mnt** và ở ngang hàng với **mnt**. Trong đó có nhiều thư mục như Alarm, Android, DCIM, Download, Movies, Notifications, Pictures, Podcasts Ringtones,...
- ➔ Chúng ta muốn thêm thư mục con, tập tin vào thư mục nào thì click chuột phải vào thư mục đó và chọn mục thích hợp,
- ➔ Muốn chép âm nhạc, hình ảnh, video từ máy tính PC hoặc tải về trên mạng thì chọn Upload và chọn đường dẫn đến tập tin cần chép vào.

## Lập trình cho thiết bị di động



### Hình 4.17

**Lưu ý:** Với phiên bản cao hơn API 22, Android 5.0, Android không còn hỗ trợ Browser BookMark, lớp **BookmarkColummn**s bị từ chối (deprecate). Muốn truy xuất dữ liệu BookMark ta phải lập trình tạo ra dữ liệu này. Chúng ta không thực hành Browser BookMark.

Tham Khảo: <https://developer.android.com/guide/topics/providers/content-provider-basics>