

# THỰC HÀNH BUỔI 3

## SỬ DỤNG LISTVIEW VÀ INTENT

### I. NỘI DUNG CẦN ÔN TẬP

- Các thành phần trong ứng dụng Android.
- Giao diện người dùng mức cao và mức thấp

#### 3.1. ListView

Chương 2 đã giới thiệu các control cơ bản, bài thực hành này chúng ta sẽ được làm quen với control nâng cao, cụ thể là ListView. Trong các ứng dụng cần lưu trữ và hiển thị *danh sách các thông tin* đa phần chúng ta dùng ListView. Trước tiên, ta sẽ sử dụng ListView có sẵn của Android. Sau đó, chúng ta sẽ thiết kế Custom Layout cho ListView (tự làm mới ListView theo ý mình). Căn bản, có 5 cách thực hiện ListView như sau:

Cách 1: Sử dụng ListView control với mảng dữ liệu định sẵn

Cách 2: Sử dụng ListView với mảng dữ liệu được lưu trong Xml

Cách 3: Sử dụng ArrayList và Listview control

Cách 4: Sử dụng ArrayList và ListView nhưng từng phần tử trong ArrayList là các Object bất kỳ

Cách 5: Sử dụng ListView dưới dạng ListActivity

Cách lập trình sẽ được hướng dẫn cụ thể trong các bài thực hành.

#### 3.2. Intent

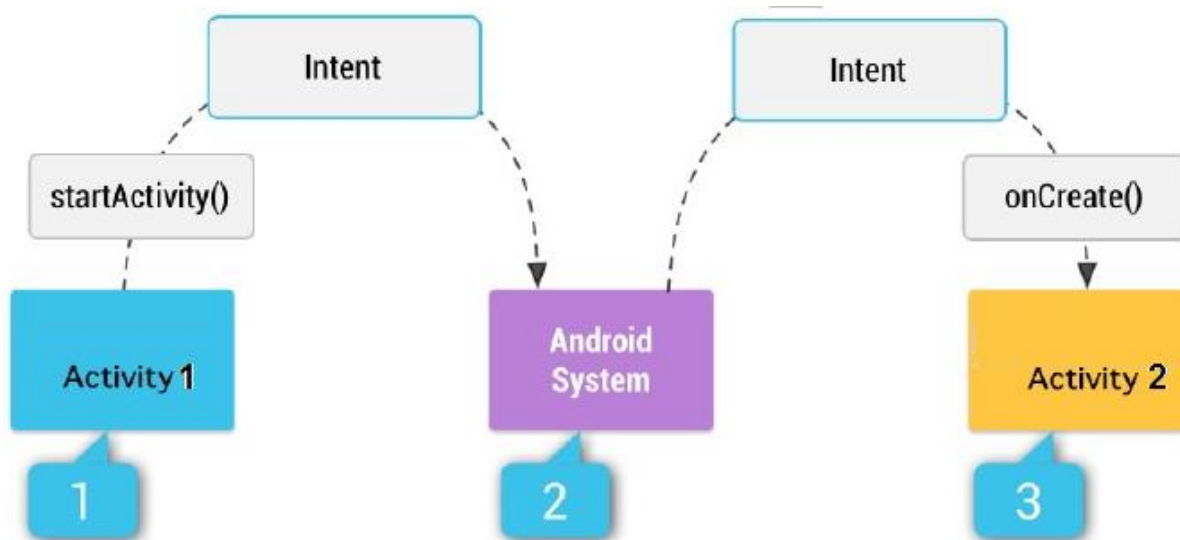
Như đã đề cập trong Chương 2, Intent là thành phần rất quan trọng trong các ứng dụng Android, chỉ sau Activity. Intent là một cấu trúc dữ liệu mô tả một thao tác sẽ được thực thi khi nó được gửi đến hệ thống. Intent thường được dùng để ***thực thi một Activity trong cùng một ứng dụng hoặc từ một ứng dụng khác, khởi động một dịch vụ (service), hoặc gửi một thông điệp (message) từ activity này sang activity khác, từ ứng dụng này sang ứng dụng khác.***

Intent là một đối tượng có kiểu là **android.content.Intent** bao gồm hai thành phần chính:

- **Action:** là một chuỗi dùng để mô tả thao tác cần thực hiện, ví dụ: ACTION\_VIEW, ACTION\_DIAL,...
- **Data:** là dữ liệu cần để thực thi thao tác, ví dụ: địa chỉ URL của trang web cần mở cho thao tác ACTION\_VIEW

Ví dụ, từ Activity1 chúng ta gọi thực thi Activity2 với đoạn code như sau:

```
Intent i = new Intent(this, Activity2.class);
startActivity(i);
```



**Hình 3.2:** Sơ đồ thực thi Activity2 từ Activity1

Hoặc dùng Intent để mở một trang web bằng trình duyệt mặc định của Android như sau:

```
String url = "http://www.cit.ctu.edu.vn";
Intent i = new Intent(Intent.ACTION_VIEW);
i.setData(Uri.parse(url));
startActivity(i);
```

Có hai loại Intent: Intent tường minh (explicit intent) và Intent bất tường minh (implicit intent).

- **Intent tường minh** là Intent mà trong đó chúng ta phải chỉ định cụ thể thành phần hoặc class nào sẽ thực thi Intent đó. Loại intent này chủ yếu dùng để thực thi một activity nào đó trong ứng dụng. Ví dụ, để thực thi activity Activity2 từ Activity1, ta tạo một intent và chỉ định class Activity2.class như sau:

```
Intent i = new Intent(this, Activity2.class);
startActivity(i);
```

- **Intent bất tường minh** là intent mà trong đó chúng ta không chỉ định thành phần hay class cụ thể nào để thực thi. Thay vào đó chúng ta phải cung cấp những thông tin cần thiết khác để hệ điều hành lựa chọn thành phần hoặc class phù hợp nhất để thực thi Intent. Ví dụ, để thực hiện gọi một số điện thoại từ ứng dụng, chúng ta có thể tạo intent như sau:

```
Intent intent = new Intent(Intent.ACTION_DIAL,
Uri.parse("tel:12345789"));
```

Chúng ta chỉ cần chỉ định action là ACTION\_DIAL và truyền thông tin bổ sung là số điện thoại theo định dạng URL. Android sẽ tìm activity tương ứng để thực hiện cuộc gọi.

### 3.2.2 Intent filters

Như đã nói ở trên Intent được dùng để mô tả thông tin thao tác sẽ được thực thi. Tuy nhiên khi intent được gửi đến hệ điều hành, làm sao để hệ điều hành biết được sẽ dùng ứng dụng hoặc dịch vụ nào để thực thi thao tác đó.

Intent Filter có thể khai báo trong tập tin AndroidManifest.xml của ứng dụng. Ví dụ, ứng dụng HelloWorld có khai báo hai Intent filter cho HelloWorldActivity như sau:

```
<activity
    android:name="edu.ctu.cit.helloworld.HelloWorldActivity"
    android:label="@string/app_name" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category
            android:name="android.intent.category.LAUNCHER" />
        </intent-filter>
    </activity>
```

- **Action Filter:** `android.intent.action.MAIN` dùng để đăng ký với hệ điều hành là HelloWorldActivity là activity chính của ứng dụng. Khi người dùng nhấn vào icon của ứng dụng trên launcher thì activity này sẽ được thực thi đầu tiên.
- **Category Filter:** `android.intent.category.LAUNCHER` dùng để báo cho hệ điều hành biết activity này sẽ được thực thi bởi launcher và launcher sẽ tạo một icon của activity này trên màn hình của nó.

Android cung cấp nhiều loại Action filter và Intent filter cho từng thao tác, chúng ta có thể tham khảo tại địa chỉ:

<http://developer.android.com/reference/android/content/Intent.html>

## II. THỰC HÀNH

### **BÀI 4: DÙNG INTENT ĐỂ GỌI ACTIVITY KHÁC TRONG CÙNG MỘT ỨNG DỤNG**

Mở project mới tên là B4\_Tên SV\_MSSV với giao diện chính gồm 1 TextView và 6 button như hình 3.1 (các text có thể không bỏ dấu tiếng Việt, nếu không thể). Tập tin Activity chính có tên là **MainActivity.java**, tập tin Layout là **activity\_main.xml**.



Hình 3.1

Khi ấn chọn button “**ListView control với mảng dữ liệu định sẵn**” thì sẽ gọi **Activity1** thực hiện một ListView control với mảng dữ liệu định sẵn.

Khi ấn chọn button “**ListView với mảng dữ liệu được lưu trong Xml**” thì sẽ gọi **Activity2** thực hiện một ListView với mảng dữ liệu được lưu trong Xml.

Khi ấn chọn button “**ArrayList và Listview control**” thì sẽ gọi **Activity3** thực hiện một Listview control với Arraylist.

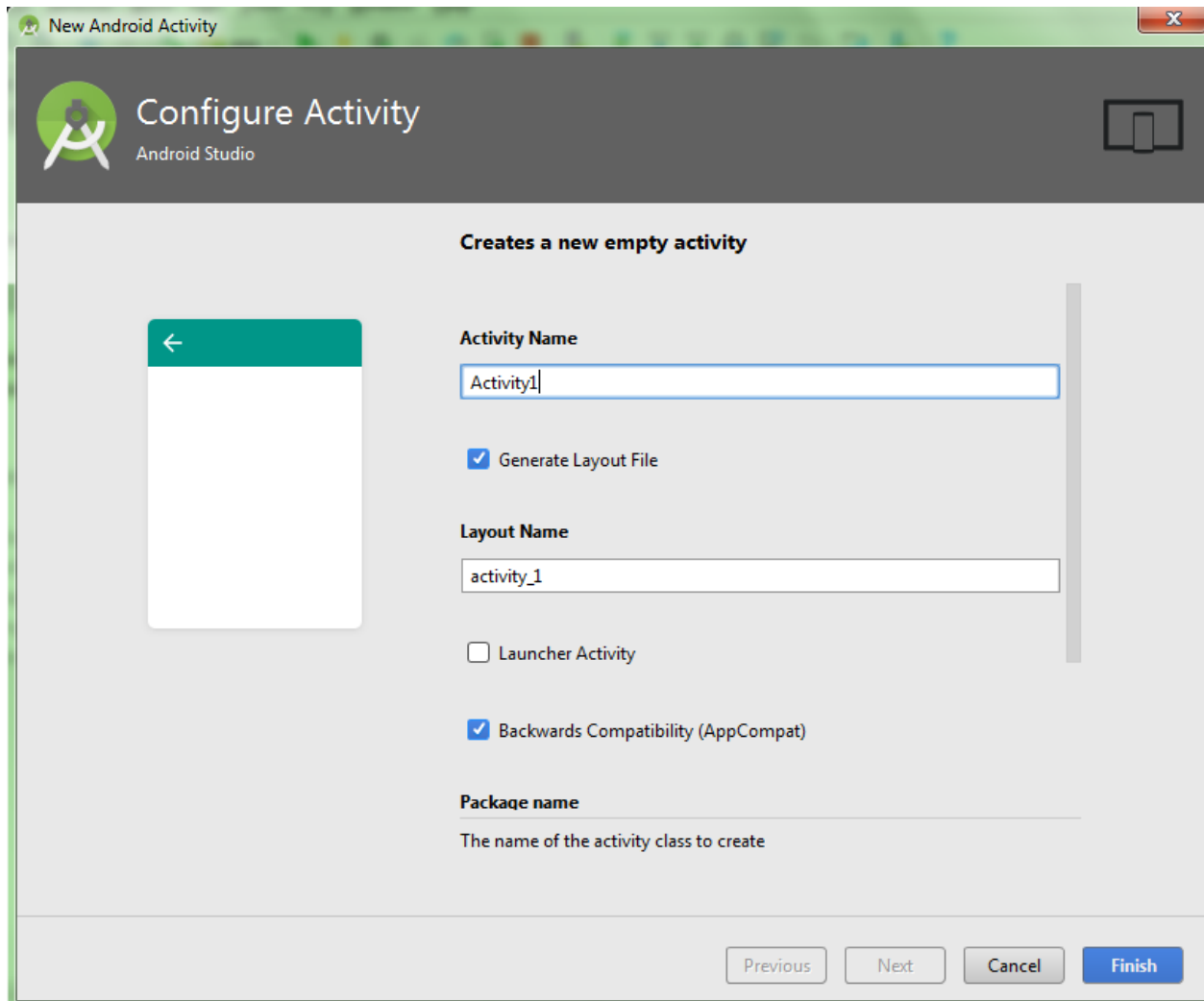
Khi ấn chọn button “**ArrayList là các Object bất kỳ**” thì sẽ gọi **Activity4** thực hiện một ListView sử dụng ArrayList nhưng từng phần tử trong ArrayList là các Object bất kỳ.

Khi ấn chọn button “**ListView dưới dạng ListActivity**” thì sẽ gọi **Activity5** thực hiện một ListView dưới dạng ListActivity.

Khi ấn chọn button “**THOÁT**” thì sẽ kết thúc ứng dụng.

## HƯỚNG DẪN:

1. Sinh viên tự **lập trình giao diện chính** (không cho code mẫu).
2. **Tạo ra Activity1:** Vào File → Chọn New → Chọn Activity → Chọn Empty Activity, giao diện New Android Activity xuất hiện (Hình 3.2). Đặt tên Activity là Activity1 và chọn Finish.



Hình 3.2

Lập trình xây dựng giao diện cho Activity1 với tập tin layout là activity\_1. Giao diện này bao gồm 1 TextView để hiển thị vị trí và giá trị của phần tử trong ListView khi được chọn, 1 ListView trình bày danh sách tên sinh viên (vị trí là số thứ tự và giá trị là tên). ListView được thành lập trước trong tập tin layout xml là listView trống (chưa có các phần tử để hiển thị), các phần tử của ListView sẽ được tạo ra trong mã nguồn Java với tập tin Activity1.java

## Mã nguồn của tập tin layout như sau:

```
<?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="com.example.doanhoaminh.buoi3_bai1.Activity1">

    <TextView
        android:id="@+id/svchon"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#ddffff"
        android:textSize="25sp"
        android:hint="Sinh vien duoc chon" />

    <ListView
        android:id="@+id/dssv"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >
    </ListView>

</LinearLayout>
```

## Mã nguồn Java:

```
package com.example.doanhoaminh.buoi3_bai1;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;

public class Activity1 extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_1);
        //1. Khởi tạo dữ liệu cho mảng arr (còn gọi là data source)
        final String arr[]={"Nguyen Van Manh", "Tran Van Giau", "Le Van Sung", "Pham van Tien"};
        //2. Lấy đối tượng ListView dựa vào id
        ListView lv=(ListView) findViewById(R.id.dssv);
        //3. Gán Data source vào ArrayAdapter
        ArrayAdapter<String>adapter=new ArrayAdapter<String>
            (this, android.R.layout.simple_list_item_1, arr);
        //4. Đưa Data source vào ListView
        lv.setAdapter(adapter);
        //5. Khai báo TextView
        final TextView txt=(TextView) findViewById(R.id.svchon);
        //6. Thiết lập sự kiện cho ListView, khi chọn phần tử nào thì hiển thị lên TextView
        lv.setOnItemClickListener(
            new AdapterView.OnItemClickListener() {
                public void onItemClick(AdapterView<?> arg0,
```

```

        View arg1,
        int arg2,
        long arg3) {
    //đổi số arg2 là vị trí phần tử trong Data Source (arr)
    txt.setText("position :"+arg2+" ; value =" +arr[arg2]);
    }
    });
}
}
}

```

## ➤ Giải thích thêm các lệnh ở bên trong lớp Activity1

[1] Khởi tạo mảng arr (còn gọi là data source) với dòng lệnh:

```

final String arr[]={ "Nguyen Van Manh","Tran Van Giau","Le Van Sung","Pham van Tien"};

```

[2] Khởi tạo thành phần ArrayAdapter là cơ sở dữ liệu để tạo ra các phần tử cho ListView, xem câu lệnh sau:

```

ArrayAdapter<String>adapter=new ArrayAdapter<String>(this,
android.R.layout.simple_list_item_1, arr);

```

Trong đó, dữ liệu từ data source (arr) sẽ được gắn vào ArrayAdapter, ArrayAdapter sẽ được gắn vào ListView.

Đối số đầu tiên của constructor ArrayAdapter : **this**, chính là biến context của Activity hiện tại, ta cũng có thể viết **MainActivity.this** (nếu viết như thế này thì ở bất kỳ vị trí nào nó cũng hiểu là context của MainActivity, nên viết như thế này để có thể copy paste nó tới bất kỳ Activity nào trong ứng dụng).

Đối số thứ 2 **android.R.layout.simple\_list\_item\_1** : là layout của ListView đã được Android xây dựng sẵn (các bài tập kế tiếp ta sẽ tự xây dựng mà không sử dụng cái có sẵn này). Tập tin **simple\_list\_item\_1** được lưu trong SDK/platforms/android-api(x)/data/res/layout/**simple\_list\_item\_1.xml**. Ta có thể xem nội dung và vị trí của layout này một cách nhanh chóng bằng đề phím Ctrl + click chuột vào dòng lệnh này, Android Studio sẽ mở cho xem tập tin này, như sau:

```

?xml version="1.0" encoding="utf-8"?>
<!-- Copyright (C) 2006 The Android Open Source Project

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

    http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software

```

*distributed under the License is distributed on an "AS IS" BASIS,  
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.  
See the License for the specific language governing permissions and  
limitations under the License.*

-->

```
<TextView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@android:id/text1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceListItemSmall"
    android:gravity="center_vertical"
    android:paddingStart="?android:attr/listPreferredItemPaddingStart"
    android:paddingEnd="?android:attr/listPreferredItemPaddingEnd"
    android:minHeight="?android:attr/listPreferredItemHeightSmall" />
```

Đối số thứ 3: chính là arr (data source), bạn có thể truyền vào ArrayList.

### [3] Thiết lập sự kiện cho ListView, khi chọn phần tử nào thì hiển thị phần tử đó lên TextView, xem dòng lệnh

```
lv.setOnItemClickListener(
    new AdapterView.OnItemClickListener() {
        public void onItemClick(AdapterView<?> arg0,
            View arg1,
            int arg2,
            long arg3) {
            //đối số arg2 là vị trí phần tử trong Data Source (arr)
            txt.setText("position :"+arg2+" ; value =" +arr[arg2]);
        }
    })
```

(chỉ cần gõ một vài ký tự đầu rồi nhấn Ctrl+ Space Bar thì các lệnh phía sau sẽ tự động xuất hiện).

Ta có interface **AdapterView.OnItemClickListener**, nó dùng để thiết lập sự kiện cho ListView, interface này có 1 phương thức trừu tượng là **onItemClick** nên ta override nội dung cần xử lý trong này, chỗ này không cần nhập từ bàn phím mà chỉ cần nhấn tổ hợp phím **Ctrl + 1** chọn add unimplement method là nó tự xuất hiện. Ngoài ra nó còn nhiều sự kiện khác SV tự tìm hiểu thêm.

### 3. Bổ sung thành phần Intent trong MainActivity như sau, phần mã nguồn chữ đậm được highlight nền vàng:

```
package com.example.doanhoaminh.buoi3_bai1;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
```



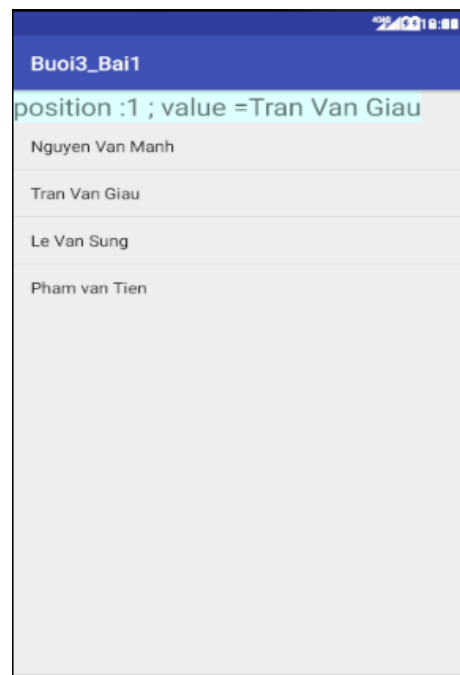
```

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button nut1=(Button)this.findViewById(R.id.bt1);
        nut1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                Intent i1 = new Intent(MainActivity.this,Activity1.class);
                startActivity(i1);
            }
        });
        Button nut2=(Button)this.findViewById(R.id.bt2);
        nut2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                Intent i2 = new Intent(MainActivity.this,Activity2.class);
                startActivity(i2);
            }
        });
    }
}

```

Chạy thử, kết quả như hình 3.3.a và 3.3.b



**Hình 3.3.a:** Chưa có phần tử nào được chọn **Hình 3.3.b:** Phần tử thứ 2 được chọn

4. **Activity2**, tương tự ta lần lượt tạo ra các Activity2, Activity3, Activity4 và Activity5.

Sau đây là hướng dẫn thực hiện Activity2.

**Activity2** là trường hợp sử dụng **ListView** với mảng dữ liệu được lưu trong **Xml** (Tương ứng với **Button** thứ 2)

Giao diện và xử lý sự kiện trong trường hợp này cũng giống hệt như trong Activity1. Chỉ khác ở chỗ là dữ liệu sẽ được load từ XML, nên trong phần này chỉ hướng dẫn **cách tạo String – Array** trong XML và **cách load String-Array** trong mã nguồn Java.

**Để tạo String – Array trong XML** ta thực hiện các bước như sau:

**Bước 1:** Bấm chuột phải vào thư mục **values** của Project → chọn New → chọn XML → Chọn Values XML files :

**Bước 2:** Màn hình New Android Component hiển thị lên, đặt tên tập tin là **mystrings.xml** rồi nhấn nút **Finish**. Tập tin mystrings.xml được sinh ra với nội dung ban đầu như sau:

```
<?xml version="1.0" encoding="utf-8"?>
<resources></resources>
```

**Bước 3:** Mở tập tin mystrings.xml, tạo một tài nguyên kiểu array có tên là **myarray**, như sau (số phần tử nhiều hay ít tùy ý):

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string-array name="myarray">
    <item>Trần Văn Thành</item>
    <item>Nguyễn Thị Phở</item>
    <item>Hồ Văn Cần</item>
    <item>Lê Văn Thơ</item>
    <item>Lê Văn Sáng</item>
    <item>Lâm Văn Xanh</item>
    <item>Trần Thị Sạch</item>
    <item>Hứa Thị Đẹp</item>
  </string-array>
</resources>
```

Như vậy ta đã biết cách tạo 1 tập tin XML và biết cách tạo String Array cũng như tạo các phần tử nằm bên trong nó. Chú ý là phần tử myarray cũng được tạo ra trong tập tin R.java:

#### Bước 4: Mở Activity2.java:

Mã nguồn Activity2 giống như Activity1, chỉ có khác ở dòng lệnh tạo ra mảng Arr , thay vì phải khai báo trực tiếp trong mã nguồn Java, ta đã khai báo trong tập tin mystrings.xml rồi. Do đó ta thay dòng lệnh:

```
final String arr[]={ "Nguyen Van Manh", "Tran Van Giau", "Le Van Sung", "Pham van Tien" };
```

trong Activity1 bằng dòng lệnh sau:

```
final String arr[]=getResources().getStringArray(R.array.myarray);
```

Ta thấy, cách này thuận lợi hơn, vì khi muốn thay đổi dữ liệu, ta chỉ cần sửa lại tập tin mystrings.xml mà không cần can thiệp vào mã nguồn Java.

#### Bước 5: Mở MainActivity.java lên và thêm đoạn lệnh xử lý sự kiện tương tác lên các button :

```
package com.example.doanhoaminh.buoi3_bai1;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Button nut1=(Button) this.findViewById(R.id.bt1);
        nut1.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                // TODO Auto-generated method stub
                Intent i1 = new Intent(MainActivity.this, Activity1.class);
                startActivity(i1);
            }
        });
        Button nut2=(Button) this.findViewById(R.id.bt2);
        nut2.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
```

```

        // TODO Auto-generated method stub
        Intent i2 = new Intent(MainActivity.this, Activity2.class);
        startActivity(i2);
    }
});

Button nut3=(Button) this.findViewById(R.id.bt3);
nut3.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        Intent i3 = new Intent(MainActivity.this, Activity3.class);
        startActivity(i3);
    }
});

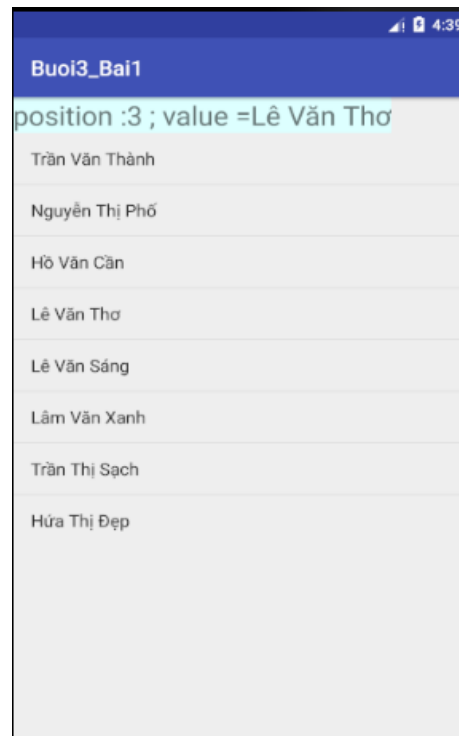
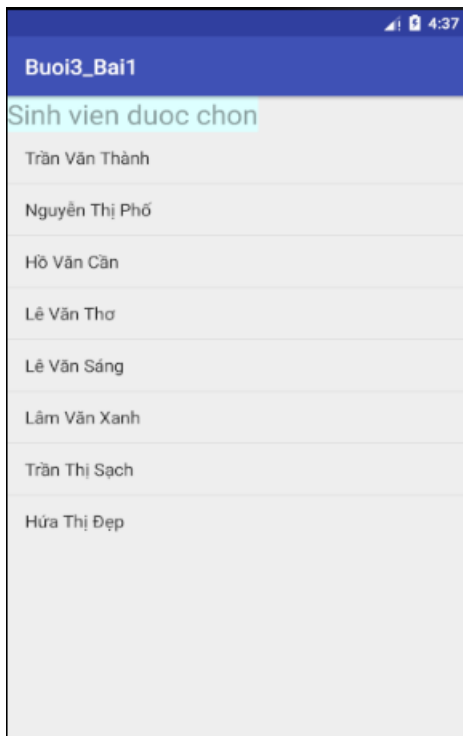
Button nut4=(Button) this.findViewById(R.id.bt4);
nut4.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        Intent i4 = new Intent(MainActivity.this, Activity4.class);
        startActivity(i4);
    }
});

Button nut5=(Button) this.findViewById(R.id.bt5);
nut5.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        Intent i5 = new Intent(MainActivity.this, Activity5.class);
        startActivity(i5);
    }
});

Button nut6=(Button) this.findViewById(R.id.bt6);
nut6.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // TODO Auto-generated method stub
        System.exit(0);
    }
});
}
}

```

Chạy thử ứng dụng và ấn chọn button thứ 2, ta sẽ được kết quả như hình 3.4:



**Hình 3.4.a:** Chưa có phần tử nào được chọn      **Hình 3.4.b:** Phần tử thứ 2 được chọn

5. **Activity3.** Sau đây là hướng dẫn thực hiện Activity3.

**Activity3** là trường hợp Sử dụng **ArrayList** và **Listview control** (Tương ứng với **Button thứ 3**). Trong phần này, ta xây dựng giao diện cho phép người dùng nhập trực tiếp các phần tử vào ListView (không cần phải nhập từ tập tin mã nguồn Java hoặc XML). Từ giao diện người dùng có thể:

- Nhập dữ liệu vào EditText và nhấn button “Nhập” thì sẽ đưa phần tử từ EditText vào ArrayList và hiển thị lên ListView.
- Nhấn vào phần tử nào thì hiển thị vị trí và giá trị của phần tử đó lên TextView.
- Nhấn thật lâu (long click ) vào phần tử nào đó trên ListView thì sẽ xóa phần tử đó.

Hướng dẫn:

**Bước 1:** Lập trình layout cho Activity3, tập tin mã nguồn XML như sau:

```
<?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="com.example.doanhoaminh.buoi3_bai1.Activity3">
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:background="#deb887"
        android:text="Nhập tên:" />
    <EditText
        android:id="@+id/txtTen"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:inputType="text"
        android:ems="10" />
    <Button
        android:id="@+id/btnNhap"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAlignment="center"
        android:layout_gravity="end"
        android:text="Nhập" />

    <TextView
        android:id="@+id/txtselection"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#007380" />

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

</LinearLayout>

```

**Bước 2:** Lập trình Activity3.java, mã nguồn như sau:

```

package com.example.doanhoaminh.buoi3_bai1;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.app.Activity;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.TextView;
import java.util.ArrayList;

```

```

public class Activity3 extends AppCompatActivity {
    EditText txtten;
    TextView txtchon;
    Button btn;
    ListView lv;
    ArrayList<String> arrList=null;
    ArrayAdapter<String> adapter=null;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_3);
        txtten=(EditText) findViewById(R.id.txtTen);
        txtchon=(TextView) findViewById(R.id.txtselection);

        lv=(ListView) findViewById(R.id.lvperson);
        //1. Tạo ArrayList object
        arrList=new ArrayList<String>();
        //2. Gán Data Source (ArrayList object) vào ArrayAdapter
        adapter=new ArrayAdapter<String>
            (this,
             android.R.layout.simple_list_item_1,
             arrList);
        //3. gán Adapter vào ListView
        lv.setAdapter(adapter);

        btn=(Button) findViewById(R.id.btnNhap);
        //4. Xử lý sự kiện nhấn nút Nhập
        btn.setOnClickListener(new View.OnClickListener() {
            public void onClick(View arg0) {
                arrList.add(txtten.getText()+"");
                adapter.notifyDataSetChanged();
            }
        });
        //5. Xử lý sự kiện chọn một phần tử trong ListView
        lv.setOnItemClickListener(new AdapterView
            .OnItemClickListener() {
            public void onItemClick(
                AdapterView<?> arg0,View arg1,
                int arg2,long arg3) {

                txtchon.setText("position : "+ arg2+
                    "; value =" +arrList.get(arg2));
            }
        });
        //6. xử lý sự kiện Long click
        lv.setOnItemLongClickListener(new AdapterView
            .OnItemLongClickListener() {
            @Override
            public boolean onItemLongClick(AdapterView<?> arg0, View arg1,
                int arg2, long arg3) {
                arrList.remove(arg2); //xóa phần tử thứ arg2
                adapter.notifyDataSetChanged();
                return false;
            }
        });
    }
}

```

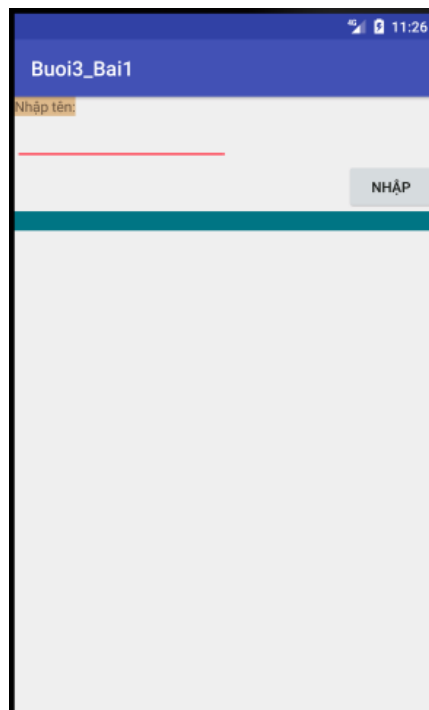
```
}
}
```

Giải thích thêm:

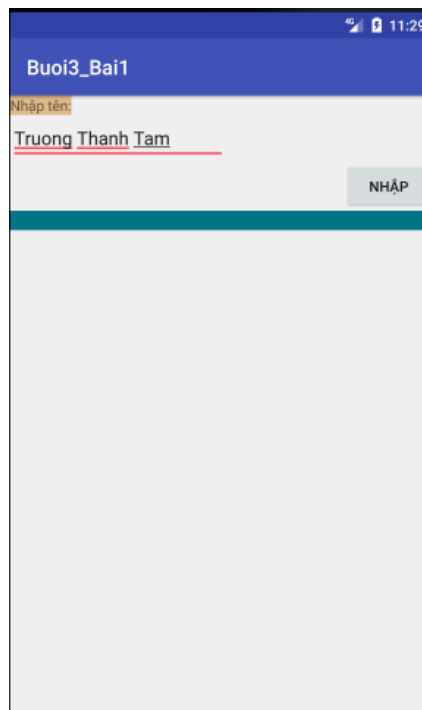
ArrayList được gán vào adapter nên mọi sự thay đổi trong ArrayList thì adapter đều nhận biết được. Khi có sự thay đổi trong ArrayList ta chỉ cần gọi hàm **notifyDataSetChanged** thì ListView sẽ được cập nhật (bởi vì adapter được gán vào ListView).

Sự kiện **setOnItemLongClickListener**, được gán cho ListView Item, khi nhấn lâu từ 2.5 tới 3 giây thì sự kiện này sẽ xảy ra. Tương tự như **setOnItemClickListener**, đối số có tên **arg2** được dùng để xác định được vị trí của phần tử nằm trong ArrayList.

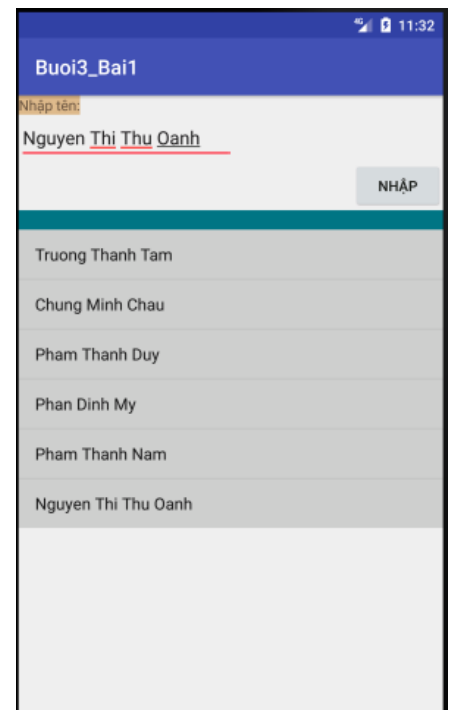
Chạy thử ứng dụng và ấn chọn button thứ 3, ta sẽ được kết quả như hình 3.5, khi ấn giữ lâu khoảng 2 giây vào phần tử nào thì phần tử đó sẽ bị xoá:



Hình 3.5.a: Chưa nhập tên



Hình 3.5.b: Nhập tên



Hình 3.5.a: Nhập tên và ấn button NHẬP

6. **Activity4**, tương tự ta tạo ra Activity4. Sau đây là hướng dẫn thực hiện Activity4.

**Activity4** là trường hợp sử dụng **ArrayList** và **ListView** nhưng **từng phần tử trong ArrayList là các Object bất kỳ (Tương ứng với Button thứ 4)**. Trong phần này, ta tạo ra ListView với nhiều phần tử khác nhau chia thành 2 nhóm thực hành.



Có 2 nhóm sinh viên : Sinh viên thực hành ngày thứ 3 và Sinh viên thực hành ngày thứ 4.  
Sinh viên thực hành ngày thứ 4.

Mỗi sinh viên có phương thức toString để xuất thông tin, nội dung xuất khác nhau. Thêm Nhóm đằng sau Id và Name.

Xem giao diện chương trình (Tập tin activity\_4.xml) hình 3.6.a, kết quả đã nhập vào 4 SV như Hình 3.6.b:

The screenshot shows the Buoi3\_Bai1 app interface. At the top is a blue header with the title "Buoi3\_Bai1". Below it is a green bar with the text "DANH SÁCH NHÓM THỰC HÀNH". The main area contains a form with the following fields: "MSSV:" with an empty text input field, "Tên SV:" with an empty text input field, and "NHÓM THỰC HÀNH:" with two radio buttons: "CHIỀU THỨ BA" (selected) and "CHIỀU THỨ TƯ". Below these fields is a button labeled "NHẬP VÀO DANH SÁCH".

Hình 3.6.a

The screenshot shows the Buoi3\_Bai1 app interface after data entry. The top part is the same as in Hình 3.6.a. The "MSSV:" field now contains "B1400592" and the "Tên SV:" field contains "Tran Nhat Quyen". The "NHÓM THỰC HÀNH:" radio buttons are now "CHIỀU THỨ BA" (unselected) and "CHIỀU THỨ TƯ" (selected). Below the form is a list of four entries, each on a separate line: "B1408584 - Ngo Tan Phuc - Nhom 1 - Thứ3.0", "B140614 - Nguyen Quang Nam - Nhom 1 - Thứ3.0", "B1400557 - Pham To Hoai - Nhom 2 - Thứ4.0", and "B1400592 - Tran Nhat Quyen - Nhom 2 - Thứ4.0".

Hình 3.6.b

Sinh viên tham khảo, phân tích các mã nguồn sau đây

#### Tập tin layout activity\_4.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="com.example.doanhoaminh.buoi3_bai1.Activity4">
<TextView
    android:id="@+id/textView1"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#008000"
    android:gravity="center"
    android:text="DANH SÁCH NHÓM THỰC HÀNH"
    android:textColor="#FFFFFF"
    android:textSize="20sp" />

<TableLayout
    android:layout_width="match_parent"
    android:stretchColumns="*"
    android:layout_height="wrap_content" >

    <TableRow
        android:id="@+id/tableRow1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >

        <TextView
            android:id="@+id/textView2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="MSSV:" />

        <EditText
            android:id="@+id/editMa"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_span="2"
            android:ems="10" >

            <requestFocus />
        </EditText>

    </TableRow>

    <TableRow
        android:id="@+id/tableRow2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" >

        <TextView
            android:id="@+id/textView3"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Tên SV:" />

        <EditText
            android:id="@+id/editTen"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_span="2"

```

```

        android:ems="10" />

</TableRow>

<TableRow
    android:id="@+id/tableRow3"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" >

    <TextView
        android:id="@+id/textView4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="NHÓM THỰC HÀNH:" />

    <RadioGroup
        android:id="@+id/radiogroud1"
        android:orientation="horizontal" >

        <RadioButton
            android:id="@+id/radNhom1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:checked="true"
            android:text="CHIỀU THỨ BA" />

        <RadioButton
            android:id="@+id/radNhom2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="CHIỀU THỨ TƯ" />
    </RadioGroup>

</TableRow>

<TableRow
    android:id="@+id/tableRow4"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" >

    <Button
        android:id="@+id/btnnhap"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_column="1"
        android:text="NHẬP VÀO DANH SÁCH" />

</TableRow>
</TableLayout>

<TextView
    android:id="@+id/textView5"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:background="#008000" />

<ListView

```

```

        android:id="@+id/lvsinhvien"
        android:layout_width="match_parent"
        android:layout_height="wrap_content" >
    </ListView>

</LinearLayout>

```

## Tập tin mã nguồn Java Activity4.java

```

package com.example.doanhoaminh.buoi3_bail;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import java.util.ArrayList;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.ListView;
import android.widget.RadioGroup;

public class Activity4 extends AppCompatActivity {

    EditText editId,editName;
    Button btnNhap;
    RadioGroup radgroup;
    ListView lvSinhvien;
    ArrayList<Nhom>arrSinhvien=new ArrayList<Nhom>();
    ArrayAdapter<Nhom>adapter=null;
    //Khai báo 1 sinhvien object
    Nhom sinhvien=null;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_4);
        editId=(EditText) findViewById(R.id.editMa);
        editName=(EditText) findViewById(R.id.editTen);
        btnNhap=(Button) findViewById(R.id.btnnhap);
        radgroup=(RadioGroup) findViewById(R.id.radiogroud1);
        lvSinhvien=(ListView) findViewById(R.id.lvsinhvien);
        //đưa Data Source là các sinhvien vào Adapter
        adapter=new ArrayAdapter<Nhom>(this,
            android.R.layout.simple_list_item_1,
            arrSinhvien);
        //đưa adapter vào ListView
        lvSinhvien.setAdapter(adapter);

        btnNhap.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View arg0) {
                // TODO Auto-generated method stub
                processNhap();
            }
        });
    }
}

```

```

    }
    });
}
//Xử lý sự kiện nhập
public void processNhap()
{
    //Lấy ra đúng id của Radio Button được checked
    int radId=radgroup.getCheckedRadioButtonId();
    String id=editId.getText()+"";
    String name=editName.getText()+"";
    if(radId==R.id.radNhom1)
    {
        //tạo instance là Nhom1
        sinhvien=new Nhom_1();
    }
    else
    {
        //Tạo instance là Nhom2
        sinhvien=new Nhom_2();
    }
    //Nhom1 hay Nhom2 thì cũng là Sinhvien
    //nên có các hàm này là hiển nhiên
    sinhvien.setId(id);
    sinhvien.setName(name);
    //Đưa sinhvien vào ArrayList
    arrSinhvien.add(sinhvien);
    //Cập nhập giao diện
    adapter.notifyDataSetChanged();
}

//Tạo các lớp chức năng để sử dụng trong Activity4
public abstract class Nhom {
    private String id;
    private String name;

    public String getId() {
        return id;
    }

    public void setId(String id) {
        this.id = id;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public abstract double TinhThu();
    @Override
    public String toString() {
        // TODO Auto-generated method stub
        return this.id+" - "+this.name;
    }
}

```

```

    }

    //
    public class Nhom_1 extends Nhom {

        @Override
        public double TinhThu() {
            return 3;
        }
        @Override
        public String toString() {
            // TODO Auto-generated method stub
            return super.toString() + " - Nhom 1 - Thứ"+TinhThu();
        }
    }

    //
    public class Nhom_2 extends Nhom {

        @Override
        public double TinhThu() {
            // TODO Auto-generated method stub
            return 4;
        }
        @Override
        public String toString() {
            // TODO Auto-generated method stub
            return super.toString() + " - Nhom 2 - Thứ"+TinhThu();
        }
    }
}

```

**7. Activity5.** Sau đây là hướng dẫn thực hiện Activity5.

- Thay vì kế thừa từ lớp **Activity**, ta sẽ cho kế thừa từ **ListActivity**.
- Và dĩ nhiên cách đặt Id cho ListView cũng có sự khác biệt.
- Tham khảo tập tin mã nguồn layout (**activity\_5.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="com.example.doanhoaminh.buoi3_bai1.Activity5">
    <TextView
        android:id="@+id/selection"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:background="#008000"

```

```

        android:textColor="#FFFFFF"
        android:textSize="20sp" />

<ListView
    android:id="@android:id/list"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >
</ListView>

<TextView
    android:id="@android:id/empty"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Không có gì cả" />

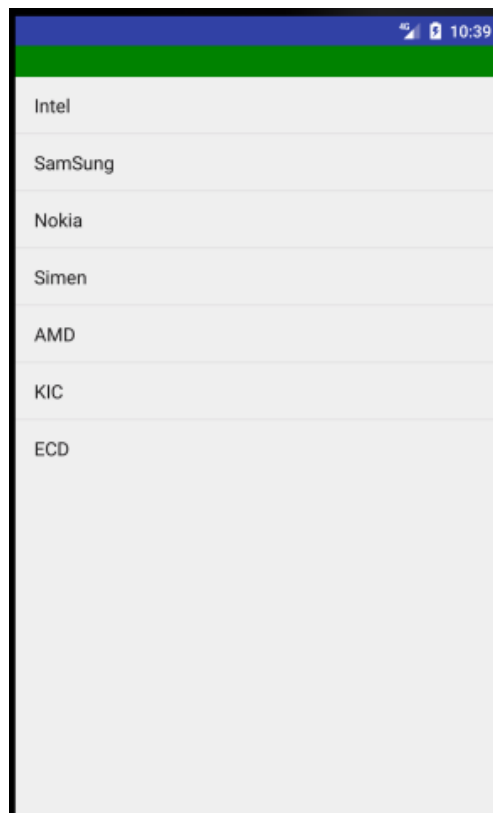
</LinearLayout>

```

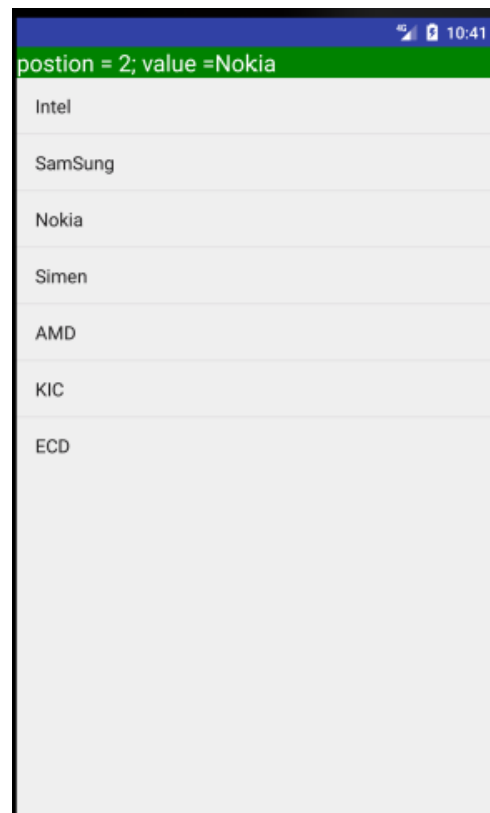
Lưu ý id của ListView

**android:id="@android:id/list"**, **phải đúng như thế này**, bởi vì nó là id được định nghĩa sẵn bên trong Android.

Giao diện như Hình 3.7.a



**Hình 3.7.a:** Chưa ấn chọn phần tử



**Hình 3.7.b:** Ấn chọn phần tử Nokia

Sinh viên tham khảo và phân tích mã nguồn tập tin **Activity5.java** sau đây:

```
package com.example.doanhoaminh.buoi3_bai1;

import android.os.Bundle;
import android.app.ListActivity;
import android.view.View;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.TextView;

public class Activity5 extends ListActivity {
    TextView selection;
    String arr[]={ "Intel", "SamSung",
                   "Nokia", "Simen", "AMD",
                   "KIC", "ECD" };
    ArrayAdapter<String >adapter=null;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_5);
        //Thiết lập Data Source cho Adapter
        adapter=new ArrayAdapter<String>
            (this, android.R.layout.simple_list_item_1, arr);
        //Gán Adapter vào ListView
        //Nhớ là phải đặt id cho ListView theo đúng quy tắc
        setListAdapter(adapter);

        selection=(TextView) findViewById(R.id.selection);
    }
    @Override
    protected void onListItemClick(ListView l, View v, int position, long id)
    {
        // TODO Auto-generated method stub
        super.onListItemClick(l, v, position, id);
        String txt="position = "+position +"; value =" +arr[position];
        selection.setText(txt);
    }
}
```

Chạy thử ứng dụng, bấm chọn button thứ 5, ta được kết quả như hình 3.7.

**Lưu ý:**

Activity5 được kế thừa từ **ListActivity** chứ không phải **Activity**. (Android đã viết class ListActivity kế thừa từ Activity rồi). Tức là ListActivity cũng chính là Activity.

- Xem dòng lệnh:

**setListAdapter(adapter);**



Ở đây ta hoàn toàn không sử dụng control ListView . Mà ta chỉ cần gọi hàm **setListAdapter** thì ListView cũng tự động cập nhập dữ liệu. Ta phải làm đúng 2 điều sau đây thì mới được như vậy:

1. Kết thừa từ **ListActivity**,

2. đặt id cho ListView theo đúng quy tắc **android:id="@android:id/list"**

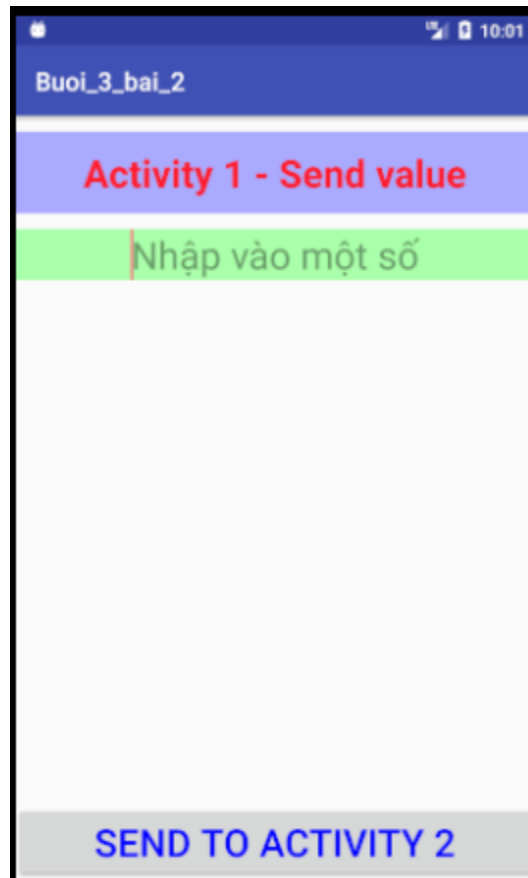
## **BÀI 5: DÙNG INTENT ĐỂ GỌI ACTIVITY KHÁC TRONG CÙNG MỘT ỨNG DỤNG CÓ TRUYỀN DỮ LIỆU KÈM THEO**

**Yêu cầu:** Xây dựng ứng dụng gồm 2 Activity. Activity thứ nhất tên là MainActivity là Activity chạy đầu tiên lúc khởi động ứng dụng, có giao diện như hình 3.8, cho phép nhập vào EditText 1 giá trị (số), sau đó chọn button SEND thì Activity2 được gọi và gửi theo dữ liệu đã nhập vào EditText. Activity2 khởi chạy, nhận và hiển thị giá trị, rồi lại gửi giá trị này tới 1 BroadcastReceiver. Cơ chế gửi và khởi chạy Activity2 sử dụng thông qua Intent.

**Bước 1:** Khởi tạo project: Vào File → Chọn New → Chọn Android Project → Đặt Project name là **Buoi\_3\_bai\_2** → Chọn Phone/Tablet, API16 - Android 4.1.

Activity chính ( được gọi khi khởi chạy ứng dụng) là MainActivity và layout tương ứng là activity\_main.xml

**Bước 2:** Lập trình giao diện cho MainActivity như hình 3.8



**Hình 3.8**

Sau đây là mã nguồn của layout activity\_main.xml , sinh viên tham khảo, làm quen các câu lệnh để xây dựng các layout tùy chọn sau này.

```
<?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="com.example.doanhoaminh.b3_2.MainActivity">

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="12dp"
        android:background="#aaaaff"
        android:gravity="center_horizontal"
        android:paddingTop="12dp"
        android:paddingBottom="12dp"
        android:text="Activity 1 - Send value"
        android:textColor="#ff1b2a"
        android:textSize="25sp"
        android:textStyle="bold">
```

```

        android:typeface="normal"
        tools:paddingBottom="12dp" />

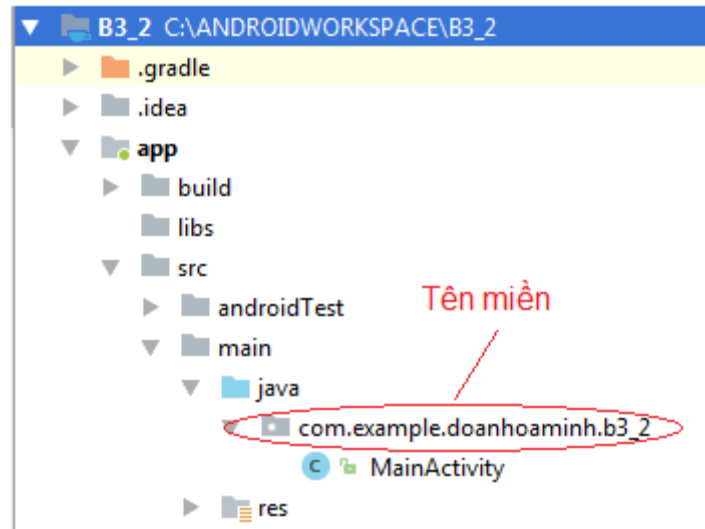
<EditText
    android:id="@+id/value_edit"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="12dp"
    android:background="#aaffaa"
    android:gravity="center"
    android:hint="Nhập vào một số"
    android:inputType="numberDecimal"
    android:lines="1"
    android:textSize="25sp" />
<RelativeLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent">
    <Button
        android:id="@+id/send_button"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textColor="#0000ff"
        android:textSize="30sp"
        android:text="Send to Activity 2"
        android:layout_alignParentBottom="true"
        />
    </RelativeLayout>
</LinearLayout>

```

Layout cho MainActivity bao gồm 1 LinearLayout chứa 1 TextView, 1 EditText để nhập giá trị (đã giới hạn kiểu nhập là number), và 1 RelativeLayout có 1 Button để khởi chạy Activity2. Ở đây, một layout con kiểu RelativeLayout được dùng để có thể xếp Button này xuống phía cuối của màn hình.

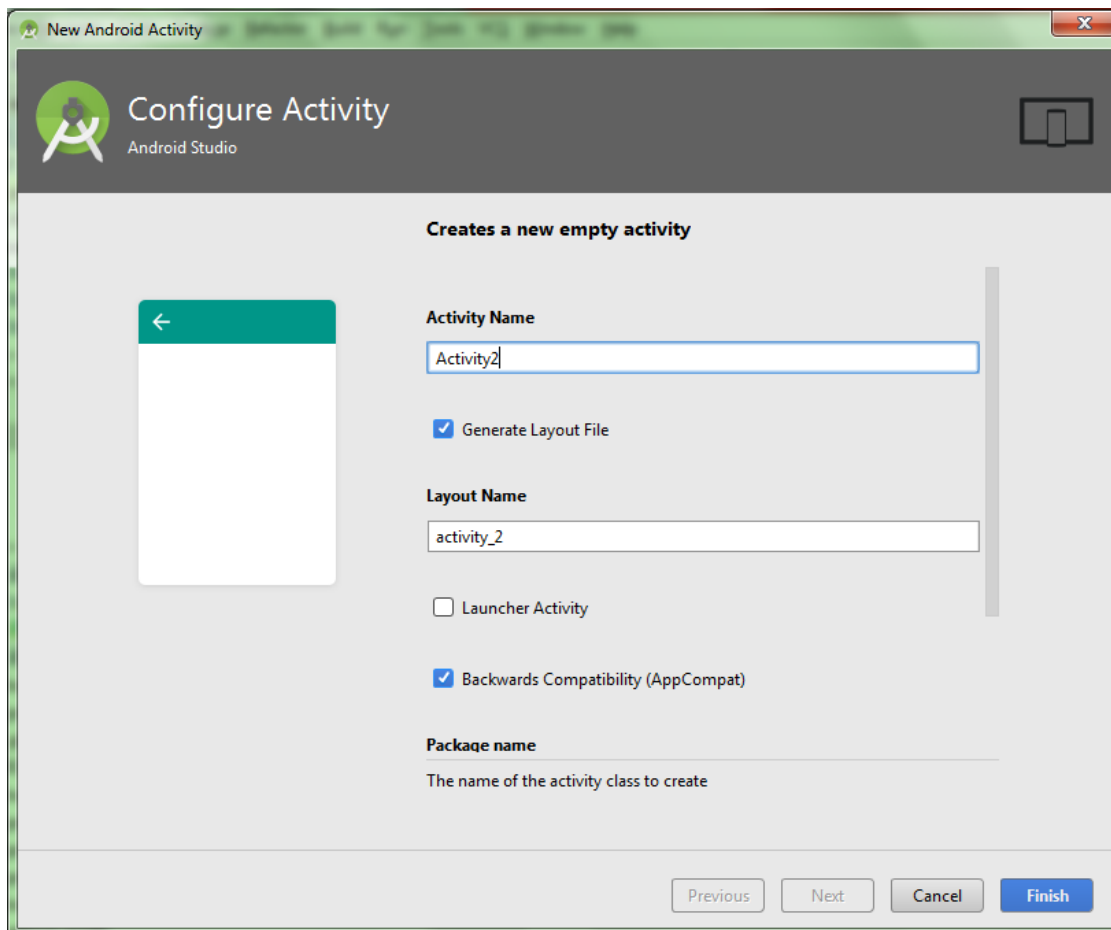
**Bước 3:** Tạo mới Activity2 và lập trình xây dựng layout cho nó.

Cách làm: Nhấp chuột phải vào tên miền của ứng dụng (hình 3.9)



**Hình 3.9**

Chọn New → Chọn Activity → Chọn Empty Activity (Hình 3.10). Cửa sổ New Android Activity xuất hiện → Đặt tên cho Activity là Activity2 → Chọn Finish.



**Hình 3.10**

Lập trình layout activity\_2.xml để có giao diện như hình 3.11. Mã nguồn của layout activity\_2.xml như sau:

```
<?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="com.example.doanhoaminh.b3_2.Activity2">
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="12dp"
        android:background="#aaaaff"
        android:gravity="center_horizontal"
        android:paddingTop="12dp"
        android:text="Activity 2 - Nhận value"
        android:textColor="#ff1b2a"
        android:textSize="25sp"
        android:textStyle="bold"
        android:typeface="normal"
        tools:paddingBottom="12dp" />
    <EditText
        android:id="@+id/value_edit"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="12dp"
        android:background="#aaffaa"
        android:gravity="center"
        android:hint="Hiển thị giá trị nhận được"
        android:inputType="numberDecimal"
        android:lines="1"
        android:textSize="25sp"
        android:enabled="false"
    />
    <RelativeLayout
        android:layout_width="fill_parent"
        android:layout_height="fill_parent">
        <Button
            android:id="@+id/call_button"
            android:layout_width="fill_parent"
            android:layout_height="wrap_content"
            android:text="Gọi Broadcast Receiver"
            android:textSize="25sp"
            android:layout_alignParentBottom="true"
        />
    </RelativeLayout>
```

</LinearLayout>

Layout của Activity2 tương tự như Activity1, nhưng Button bây giờ là để gọi Broadcast Receiver. Ngoài ra, ví dụ này đã dùng EditText để hiển thị value nhận được (do nó có cái đường bao ngoài đẹp hơn TextView) nên đã dùng thuộc tính `android:enabled="false"` để không cho phép nhập giá trị vào EditText này.

**B4:** Sửa lại nội dung của MainActivity.java, lập trình tương tác lên Button “SEND TO ACTIVITY2”, dùng **Intent tường minh** để gọi Activity2 và dùng **hàm putExtras()** để gửi dữ liệu kèm theo.

Sinh viên tham khảo mã nguồn sau đây, phân tích, hiểu được công dụng, cú pháp để áp dụng sáng tạo sau này.

```
package com.example.doanhoaminh.b3_2;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        //Khai báo EditText và Button
        final EditText editValue = (EditText) findViewById(R.id.value_edit);
        final Button sendButton = (Button) findViewById(R.id.send_button);

        // Đáp ứng tương tác người dùng
        sendButton.setOnClickListener(new OnClickListener() {
            public void onClick(View v) {
                //Lấy giá trị từ EditText và chuyển thành String tên là valueString.
                String valueString = editValue.getText().toString();
                long value;
                //Nếu valueString khác trống thì đổi thành số Long và gán cho value, ngược
```

```

//lại thì gán value=0.
    if (!valueString.equals("")) {
        value = Long.parseLong(valueString);
    }
    else {
        value = 0;
    }

    //Tạo 1 đối tượng Bundle để gửi đi cùng Intent
    Bundle sendBundle = new Bundle();
    sendBundle.putLong("value", value);

    //Tạo Intent để khởi chạy Activity2 và gán đối tượng sendBundle vào Intent
    Intent i = new Intent(MainActivity.this, Activity2.class);
    i.putExtras(sendBundle); // Dùng hàm putExtras
    startActivity(i);

    //Giải phóng Activity1 khỏi Activity Stack vì ta sẽ không quay lại nó nữa
    finish();
}
});
}
}

```

## B5: Lập trình Activity2.java

Mã:

```

package com.example.doanhoaminh.b3_2;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

public class Activity2 extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {

```

```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_2);
final EditText receiveValueEdit = (EditText) findViewById(R.id.value_receive);
final Button callReceiverButton = (Button) findViewById(R.id.call_button);

//Lấy về Bundle được gửi kèm Intent rồi lấy ra giá trị
Bundle receiveBundle = this getIntent().getExtras();
final long receiveValue = receiveBundle.getLong("value");
//Hiển thị giá trị lấy về trên EditText
receiveValueEdit.setText(String.valueOf(receiveValue));
// Gửi giá trị đến BroadcastReceiver
callReceiverButton.setOnClickListener(new OnClickListener() {
    public void onClick(View v) {
        //Khởi tạo 1 Intent để gửi tới Broadcast Receiver
        //Gắn giá trị vào Intent, lần này không cần Bundle nữa (dùng hàm putExtra
        Intent i = new Intent(Activity2.this, Receiver.class);
        i.putExtra("new value", receiveValue - 10);
        sendBroadcast(i);
    }
});
}
}

```

**B6:** Tạo Broadcast Receiver để nhận Intent mà Activity2 gửi tới. Đây là 1 Java Class. D9w63 tạo ra lớp này, ta ấn chuột phải vào tên miền → Chọn New → Chọn Java Class → Đặt tên là Receiver → Chọn OK. Lập trình cho nội dung của lớp này như sau:

```

package com.example.doanhoaminh.buoi_3_bai_2;
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.widget.Toast;

public class Receiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        long value = intent.getLongExtra("new value", -10) + 10;
        Toast toast = Toast.makeText(context, "Broadcast Receiver catch an Intent"
            + " \n" + "The value is stored in the Intent is "
            + String.valueOf(value), Toast.LENGTH_LONG);
        toast.show();
    }
}

```



Sinh viên đọc và tìm hiểu mã nguồn. Lưu ý, ta đã dùng Toast, là lớp để hiển thị một thông báo đơn giản trong 1 khoảng thời gian cố định, ta không thể thay đổi thời gian này mà chỉ có thể chọn giữa LENGTH\_SHORT với LENGTH\_LONG.

**B7:** Bổ sung thêm thông tin thành phần (component) mới vào tập tin AndroidManifest.xml đó là Receiver (đây không phải là 1 activity nên Android SDK không tự động nhập vào):

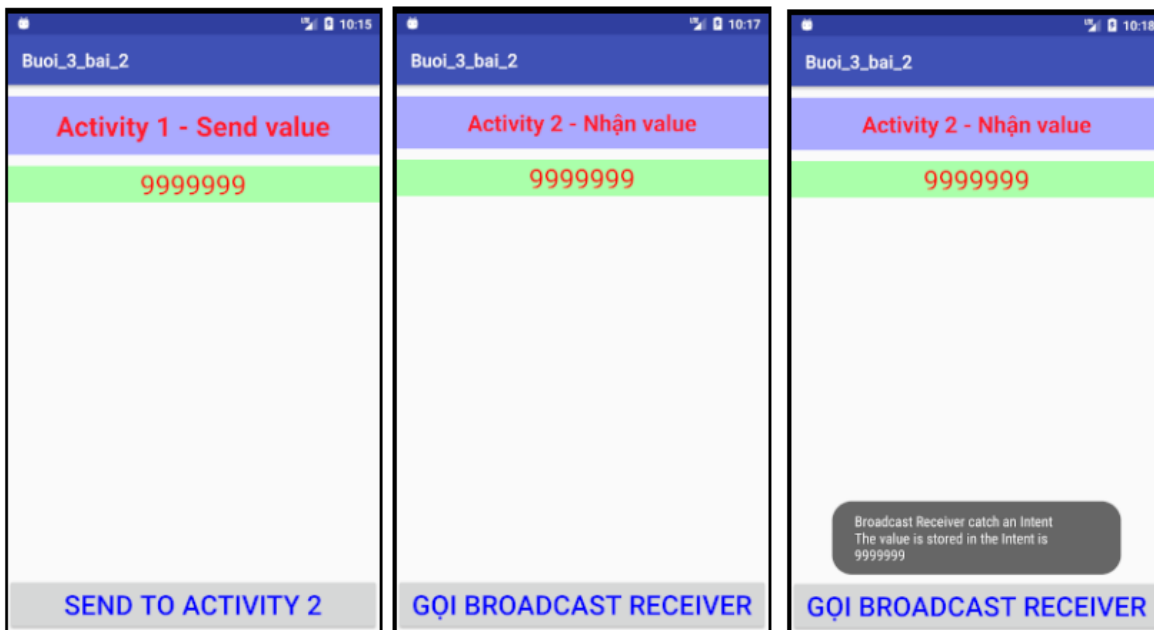
```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.doanhoaminh.buoi_3_bai_2">

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="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>
        <activity android:name=".Activity_2"></activity>
        <receiver android:name=".Receiver"></receiver>
    </application>

</manifest>
```

**B8:** Chạy thử, kết quả như hình 3.12



**Hình 3.12.a:**  
Nhập 1 số vào EditText  
trong giao diện chính

**Hình 3.12.b:**  
Button SEND TO ACTIVITY 2  
được chọn để chuyển đến Activity 2

**Hình 3.12.c:**  
Button BROADCASTRECEIVER được chọn  
để gọi giá trị đến BroadcastReceiver

## **BÀI 6:** DÙNG INTENT ĐỂ GỌI ACTIVITY KHÁC TRONG MỘT ỨNG DỤNG KHÁC

Chúng ta đã từng sử dụng lớp Intent để di chuyển qua lại giữa các Activity, làm như thế là chúng ta đã gọi Intent một cách tường minh, tức là khai báo rõ ràng tên lớp Activity sẽ được mở bởi Intent. Còn đôi khi chúng ta muốn mở một ứng dụng khác thì lúc đó chúng ta sử dụng Intent ngầm (không tường minh - Implicit Intent).

Các đối tượng Intent ngầm sẽ không nhận tên của một lớp Activity nào cả, mà thay vào đó nó mô tả một hành động (Action) nào đó và cung cấp một số dữ liệu để có thể thực hiện được hành động đó, chẳng hạn như mở ứng dụng bản đồ, mở trình duyệt, mở ứng dụng gửi mail, chụp hình, quay phim,...

Khi ta gọi phương thức `startActivity()` hay `startActivityForResult()` và truyền một Implicit Intent, hệ thống sẽ liên kết Intent này với ứng dụng mà có thể xử lý hành động đó và chạy Activity tương ứng. Nếu có nhiều hơn một ứng dụng có thể xử lý hành động này, thì hệ thống sẽ hiển thị một hộp thoại để người dùng có thể chọn lựa.

**Lưu ý:** Nếu không có ứng dụng nào trong thiết bị có thể nhận Implicit Intent, thì ứng dụng của ta sẽ bị lỗi khi thực hiện lệnh `startActivity()`. Để có thể kiểm chứng trước xem có tồn tại một ứng dụng nào đó có thể nhận Intent này, ta gọi hàm `resolveActivity()` trên đối tượng Intent đó.

- Nếu kết quả trả về khác null, đồng nghĩa với việc có ít nhất một ứng dụng có thể xử lý Intent và lúc này thì ta có thể gọi `startActivity()` một cách an toàn.

- Nếu kết quả trả về là null, thì không nên sử dụng Intent này. Nếu có thể thì nên thông báo cho người dùng biết, và chỉ ra hướng xử lý, đồng thời ta nên tắt các tính năng liên quan đến Intent.

Tham khảo thêm về cách tạo Intent hay Intent Filter với các link sau:

<https://developer.android.com/guide/components/intents-filters.html>

Bảng dưới đây liệt kê một số định dạng dữ liệu và action tương ứng đã được định nghĩa bởi Android:

Định dạng	Action	Mô tả
tel:phone_number	ACTION_VIEW	Mở một Dial form
tel:phone_number	ACTION_CALL	Thực hiện gọi tới số
http://web_address https://web_address	ACTION_VIEW	Mở trình duyệt web theo địa chỉ
http://web_address https://web_address	ACTION_WEB_SEARCH	Thực hiện tìm kiếm
sms://	ACTION_SENDTO	Gửi tin nhắn
geo:latitude,longitude geo:latitude,longitude?z=zoom geo:0,0?q=my+street+address geo:0,0?q=business+near+city	ACTION_VIEW	Mở ứng dụng Bản đồ và trở tới vị trí được xác định

Tham khảo thêm các Action của Implicit Intent tại url:

<https://developer.android.com/reference/android/content/Intent.html>

**Các bước thực hiện một intent ngắ:**

- (1) Tại dữ liệu kèm theo Intent dưới dạng uri.
- (2) Khai báo một intent với một hành động cụ thể.
- (3) Gọi hàm `startActivity()` hay `startActivityForResult()` để khởi động Intent.

Ví dụ:

```
Uri uri = Uri.parse("tel:" + "900");
Intent intent = new Intent(Intent.ACTION_CALL, uri);
startActivity(intent);
```

**Lưu ý:** Nếu không có ứng dụng nào trong thiết bị có thể nhận Implicit Intent, thì ứng dụng của ta sẽ bị lỗi khi thực hiện lệnh `startActivity()`. Để có thể kiểm chứng trước xem có tồn tại một ứng dụng nào đó có thể nhận Intent này, ta gọi hàm `resolveActivity()` trên đối tượng Intent đó.

- Nếu kết quả trả về khác null, đồng nghĩa với việc có ít nhất một ứng dụng có thể xử lý Intent và lúc này thì ta có thể gọi `startActivity()` một cách an toàn.

Ví dụ:

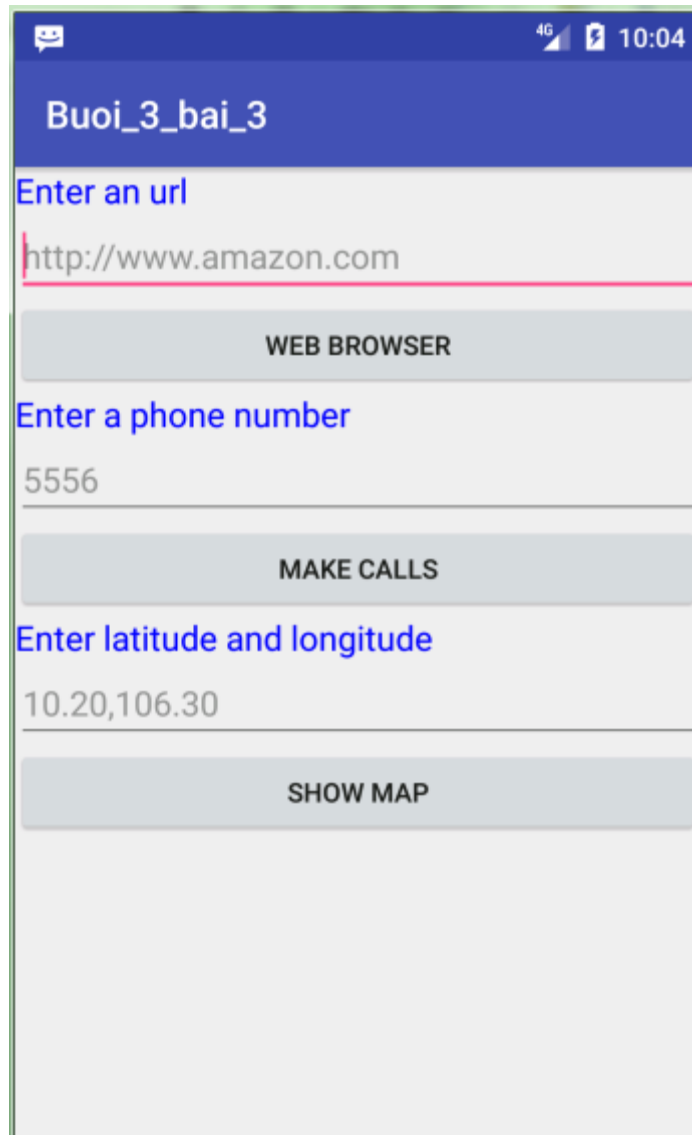
```
if (intent.resolveActivity(getPackageManager()) != null) {  
    startActivity(intent);  
}
```

- Nếu kết quả trả về là null, thì không nên sử dụng Intent này. Nếu có thể thì nên thông báo cho người dùng biết, và chỉ ra hướng xử lý, đồng thời ta nên tắt các tính năng liên quan đến Intent.

## Bài thực hành:

Mở một project mới và xây dựng một ứng dụng dùng Intent ngầm để gọi một số ứng dụng sẵn của thiết bị: khởi động một trình duyệt để duyệt một trang web; sử dụng gọi điện thoại; sử dụng Google map để tải về bản đồ,... các bước thực hiện như sau:

**B1:** Mở project mới có tên là Buoi\_3\_bai 3, Activity chính là MainActivity, có giao diện gồm 3 TextView, 3 EditText và 3 Button, như hình 3.13. Khi nhập URL vào EditText thứ nhất vào và ấn chọn nút WEB BROWSER thì ứng dụng sẽ gọi 1 trình duyệt mặc định của thiết bị để truy cập đến 1 trang web. Khi nhập số điện thoại vào EditText thứ hai và ấn chọn nút MAK CALLS thì ứng dụng sẽ khởi động chức năng gọi điện thoại của thiết bị để gọi đến số đã nhập vào. Khi nhập số vĩ độ và kinh độ vào EditText thứ ba và ấn chọn nút SHOW MAP thì ứng dụng Maps của thiết bị được khởi động để hiển thị bản đồ (Trong ví dụ này đã nhập lần lượt vĩ độ và kinh độ của 1 khu vực TP. Hồ Chí Minh: 10<sup>0</sup>20 và 106<sup>0</sup>30).



**Hình 3.13**

Sinh viên tham khảo mã nguồn layout chính sau đây:

```
<?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:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="#0000ff"
```

```
        android:textSize="18sp"
        android:text="Enter an url"/>
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/nhapurl"
        android:hint="http://www.amazon.com"/>
    <Button
        android:id="@+id/btn_webbrowser"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Web Browser"
        android:onClick="onClickWebBrowser" />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="#0000ff"
        android:textSize="18sp"
        android:text="Enter a phone number"/>
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/nhapnumberphone"
        android:inputType="numberDecimal"
        android:hint="5556"/>
    <Button
        android:id="@+id/btn_makecalls"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Make Calls"
        android:onClick="onClickMakeCalls" />
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textColor="#0000ff"
        android:textSize="18sp"
        android:text="Enter meridian and latitude"/>
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:id="@+id/nhapgeo"
        android:hint="37.827500,-122.481670"/>
    <Button
        android:id="@+id/btn_showMap"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Show Map"
        android:onClick="onClickShowMap" />

</LinearLayout>
```

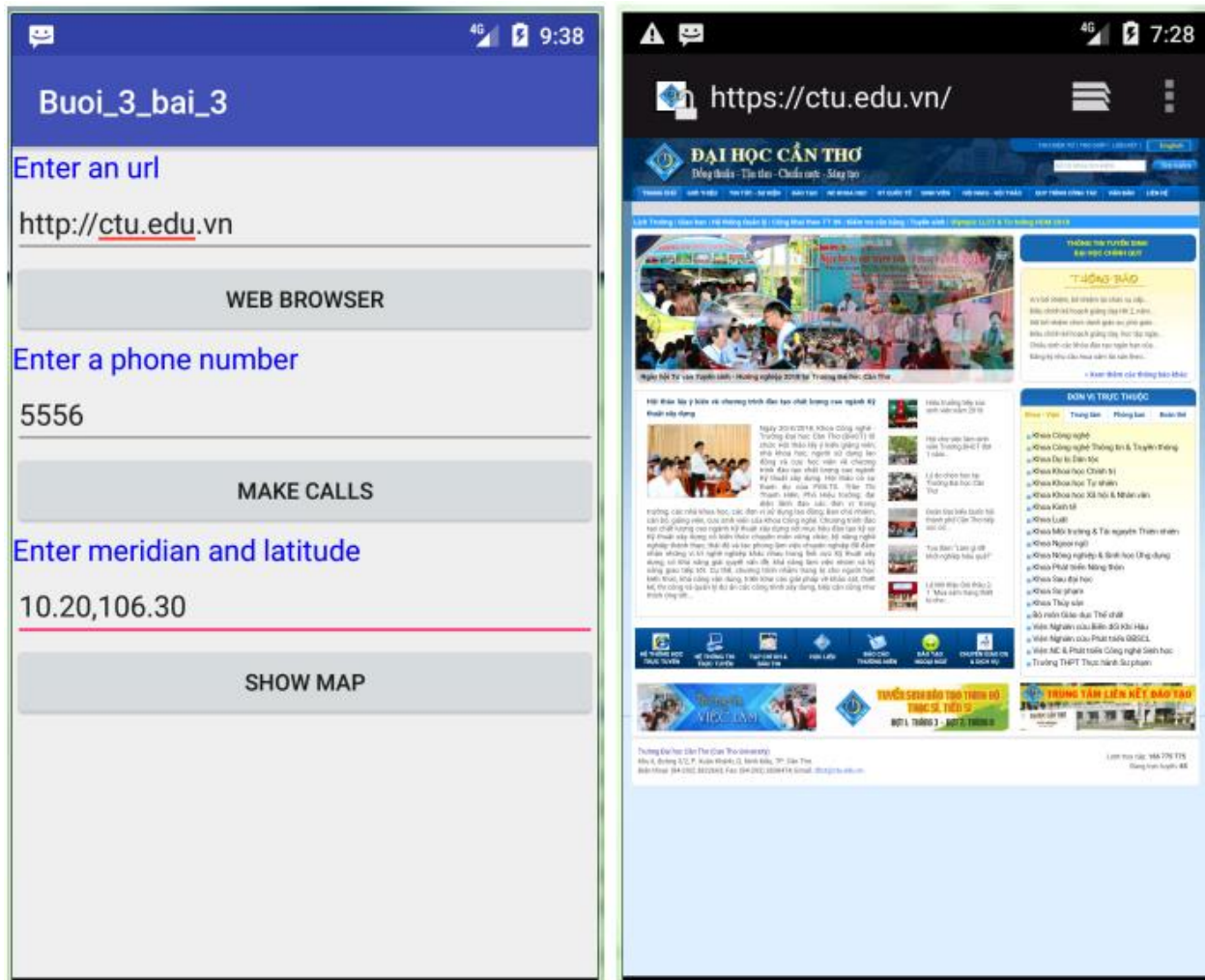
Sinh viên tham khảo mã nguồn tập tin MainActivity như sau (mã nguồn này cũng dễ hiểu nên không cần giải thích thêm):

```
package com.example.doanhoaminh.buoi_3_bai_3;

import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.content.Intent;
import android.net.Uri;
import android.view.View;
import android.widget.EditText;

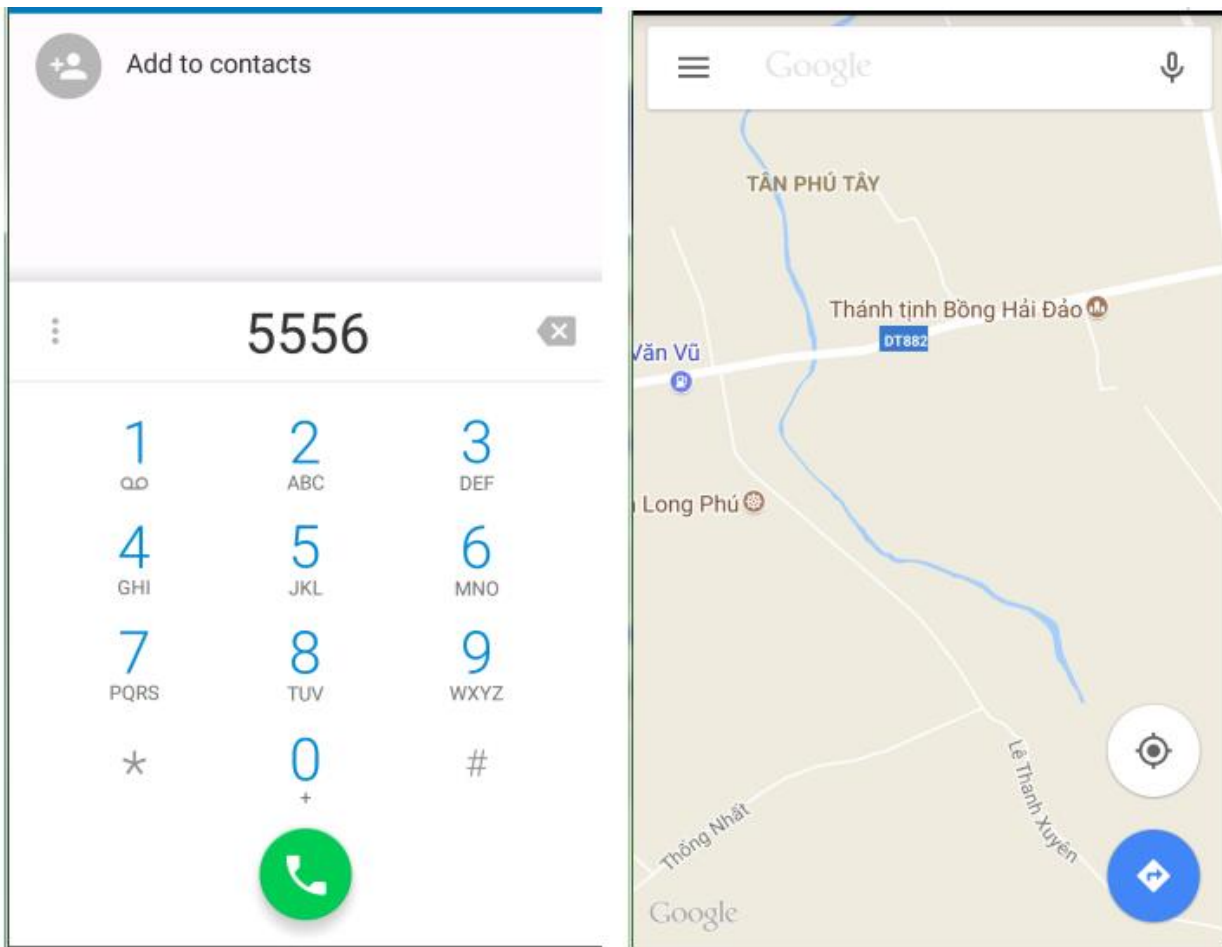
public class MainActivity extends AppCompatActivity {
    int request_Code = 1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    public void onClickWebBrowser(View view) {
        EditText txt_url = (EditText) findViewById(R.id.nhapurl);
        String url = txt_url.getText().toString();
        Intent i = new Intent(android.content.Intent.ACTION_VIEW,
            Uri.parse(url));
        startActivity(i);
    }
    public void onClickMakeCalls(View view) {
        EditText phone_number = (EditText)
findViewById(R.id.nhapnumberphone);
        String pn = phone_number.getText().toString();
        Intent i = new Intent(android.content.Intent.ACTION_DIAL,
            Uri.parse("tel:"+pn));
        startActivity(i);
    }
    public void onClickShowMap(View view) {
        EditText position = (EditText) findViewById(R.id.nhapgeo);
        String ps = position.getText().toString();
        Intent i = new Intent(android.content.Intent.ACTION_VIEW,
            Uri.parse("geo:"+ps));
        startActivity(i);
    }
}
```

Chạy thử với máy ảo Android, kết quả như các hình sau:



Hình 3.14





**Hình 3.15**