

# BÀI TẬP THỰC HÀNH

PHÁT TRIỂN ỨNG DỤNG DI ĐỘNG

## **MỤC LỤC**

BÀI THỰC HÀNH 01: Giới thiệu ứng dụng Donation	3
BÀI THỰC HÀNH 02: Giới thiệu ứng dụng Donation (tiếp)	15
BÀI THỰC HÀNH 03: Giới thiệu ứng dụng Donation (tiếp)	22
BÀI THỰC HÀNH 04: Giới thiệu ứng dụng Donation (tiếp)	30
BÀI THỰC HÀNH 05: Multi Screen App & Menus	40
BÀI THỰC HÀNH 06: Multi Screen App & Menus(tiếp)	42
BÀI THỰC HÀNH 07: Donation Object Model	49
BÀI THỰC HÀNH 08: Donation Object Model (tiếp)	54
BÀI THỰC HÀNH 09: Donation 4.0 - Database/Application Support	60
BÀI THỰC HÀNH 10: Donation 4.0 - Database/Application Support (tiếp)	64
BÀI THỰC HÀNH 11: Donation 4.0 - Database/Application Support (tiếp)	69
BÀI THỰC HÀNH 12: Hỗ trợ REST/Cloud cho ứng dụng	71
BÀI THỰC HÀNH 13: Hỗ trợ REST/Cloud cho ứng dụng (tiếp)	76
BÀI THỰC HÀNH 14: Hỗ trợ REST/Cloud cho ứng dụng (tiếp)	87
BÀI THỰC HÀNH 15: Hỗ trợ REST/Cloud cho ứng dụng (tiếp)	94

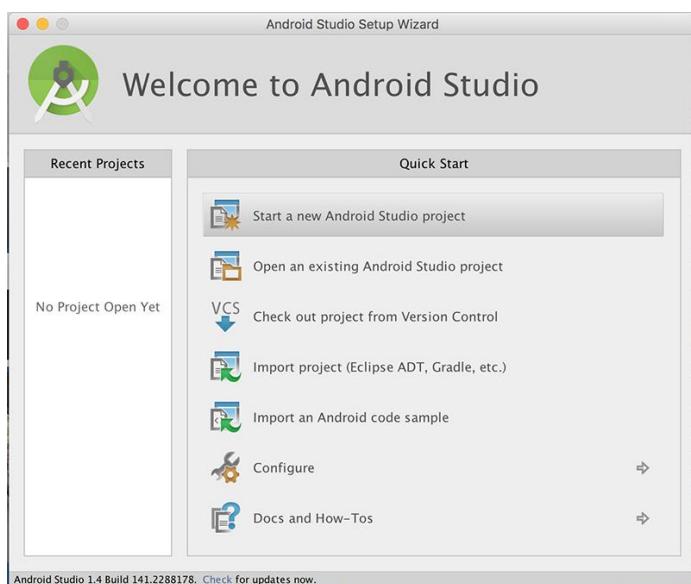
# BÀI THỰC HÀNH 01: Giới thiệu ứng dụng Donation

**Mục tiêu:** Xây dựng ứng dụng **Donation ver1.0**

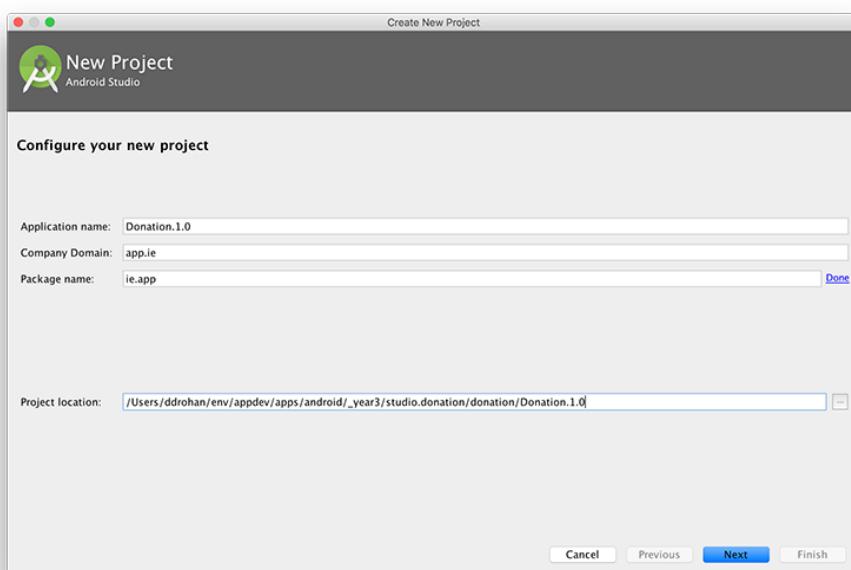
- Thiết kế layout ban đầu cho ứng dụng
- Thêm một số widgets vào giao diện
- Cài đặt một số hàm xử lý sự kiện cơ bản khi nhấn vào một nút

## Bước 1. Tạo project Donation.1.0

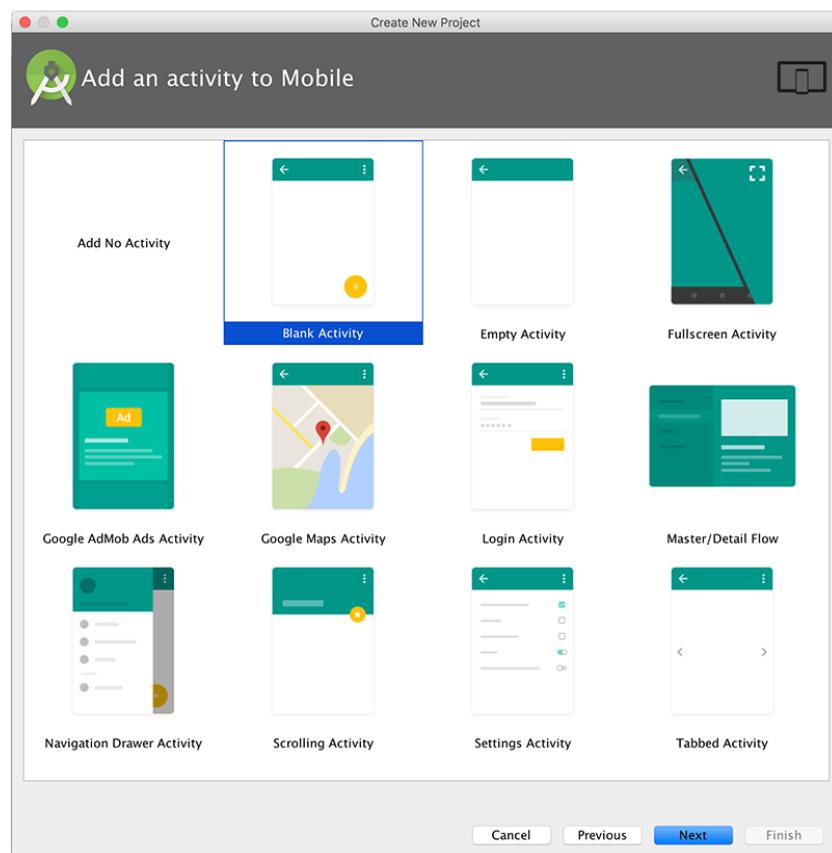
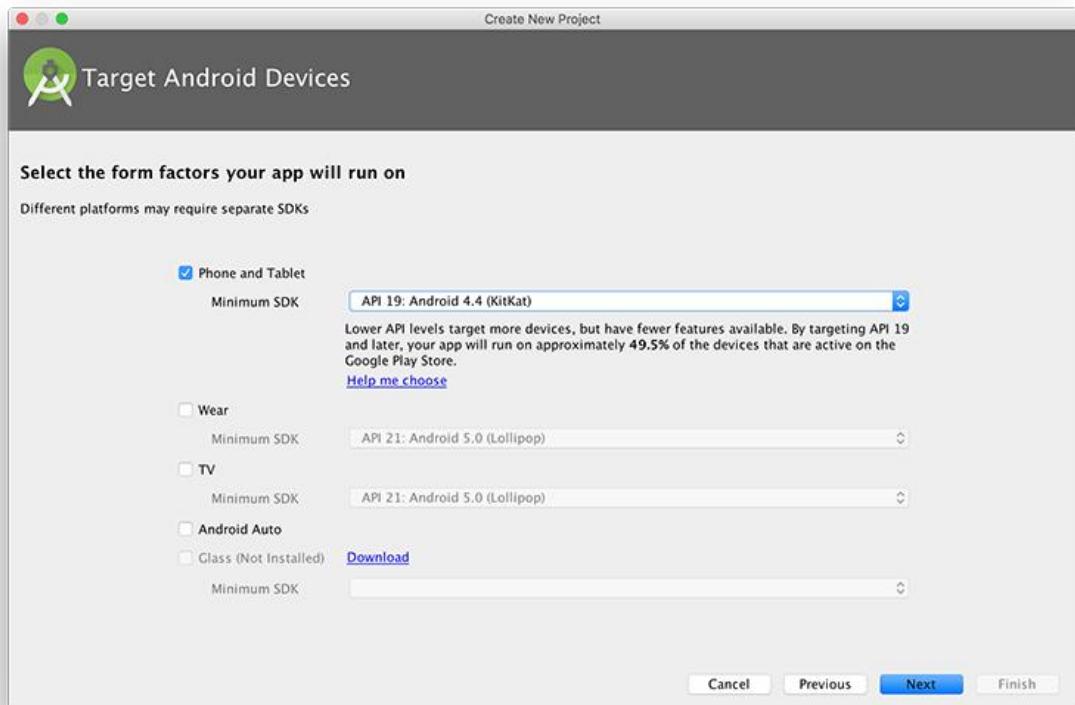
- Tạo một ứng dụng Android mới



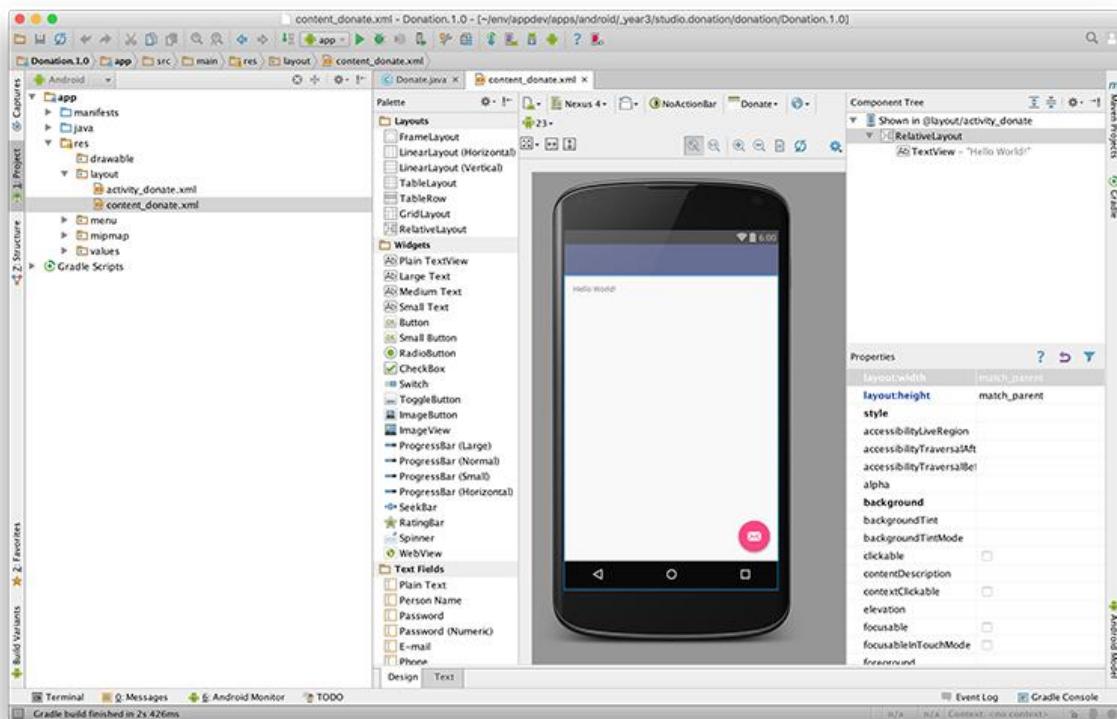
- Thiết lập cho project



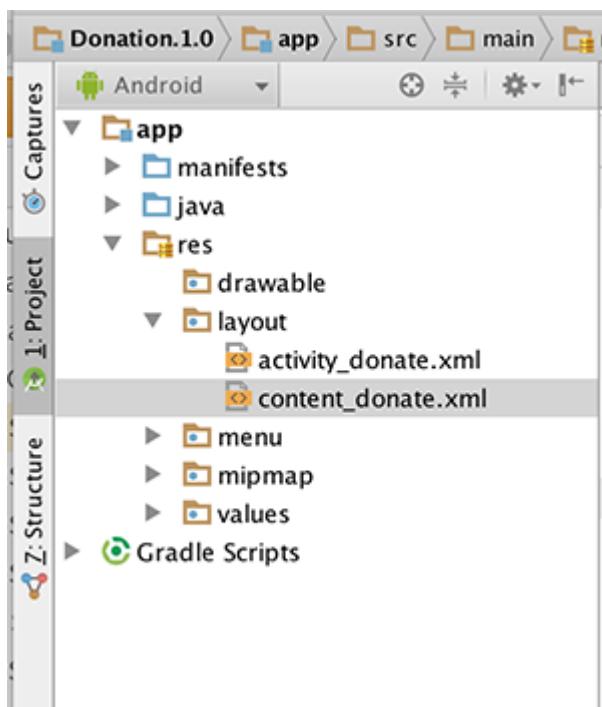
- Lựa chọn phiên bản API thấp nhất mà ứng dụng hỗ trợ; phiên bản API càng thấp thì càng nhiều thiết bị có thể chạy được, tuy nhiên các tính năng được hỗ trợ sẽ bị hạn chế



Khung thiết kế layout `content_donate` hiện ra như sau:

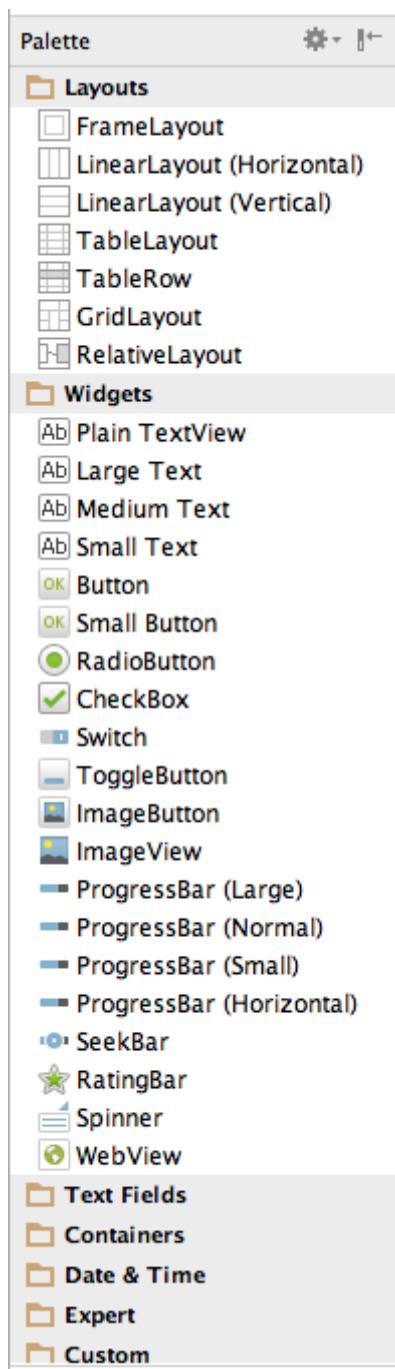


- Cấu trúc project Donation như sau

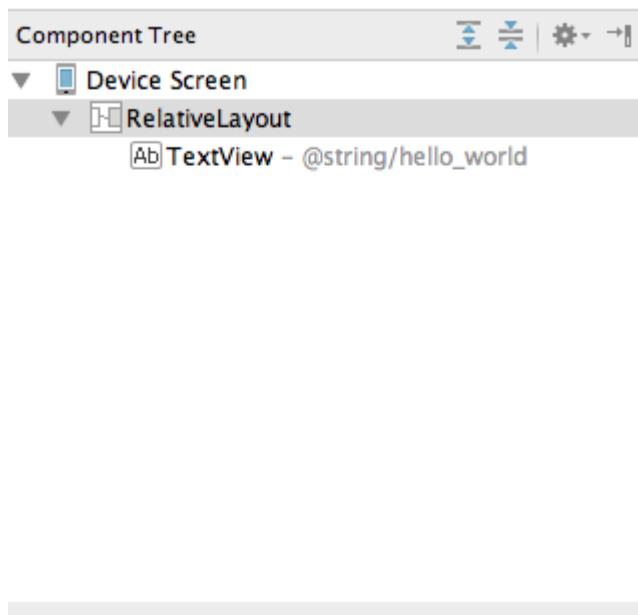


- Làm quen với cấu trúc của 3 phần tùy chỉnh giao diện ứng dụng **Donation**

## Pallette



## Outline



## Properties

Properties	
layout:width	match_parent
layout:height	match_parent
style	
accessibilityLiveRegion	
alpha	
background	
clickable	<input type="checkbox"/>
contentDescription	
focusable	<input type="checkbox"/>
focusableInTouchMode	<input type="checkbox"/>
▶ gravity	<input type="checkbox"/>
id	
ignoreGravity	
importantForAccessibility	
labelFor	
layoutMode	

- Những giao diện tùy chỉnh này có mối liên quan chặt chẽ, cần theo dõi các thông số hiển thị trong quá trình thiết kế giao diện ứng dụng
- Các thiết lập và phụ thuộc của ứng dụng được lưu trong tệp **build.gradle**

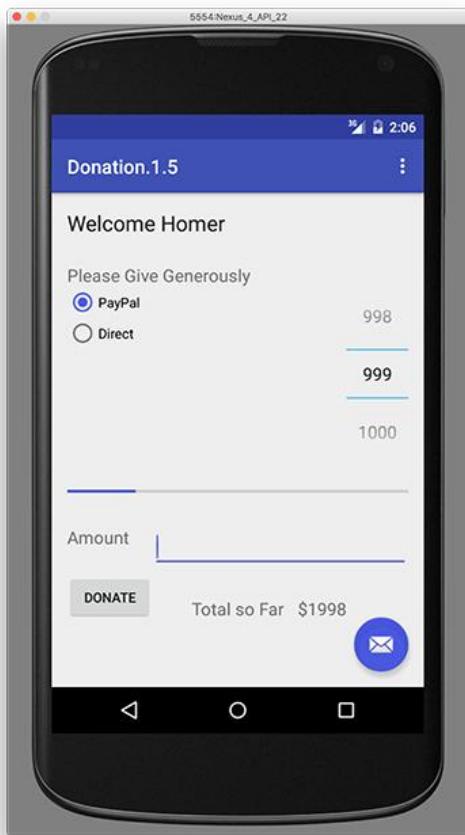
```

1 apply plugin: 'com.android.application'
2
3 android {
4     compileSdkVersion 23
5     buildToolsVersion "23.0.1"
6
7     defaultConfig {
8         applicationId "ie.app"
9         minSdkVersion 19
10        targetSdkVersion 23
11        versionCode 1
12        versionName "1.0"
13    }
14    buildTypes {
15        release {
16            minifyEnabled false
17            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
18        }
19    }
20
21    dependencies {
22        compile fileTree(dir: 'libs', include: ['*.jar'])
23        testCompile 'junit:junit:4.12'
24        compile 'com.android.support:appcompat-v7:23.0.1'
25        compile 'com.android.support:design:23.0.1'
26    }
27
28

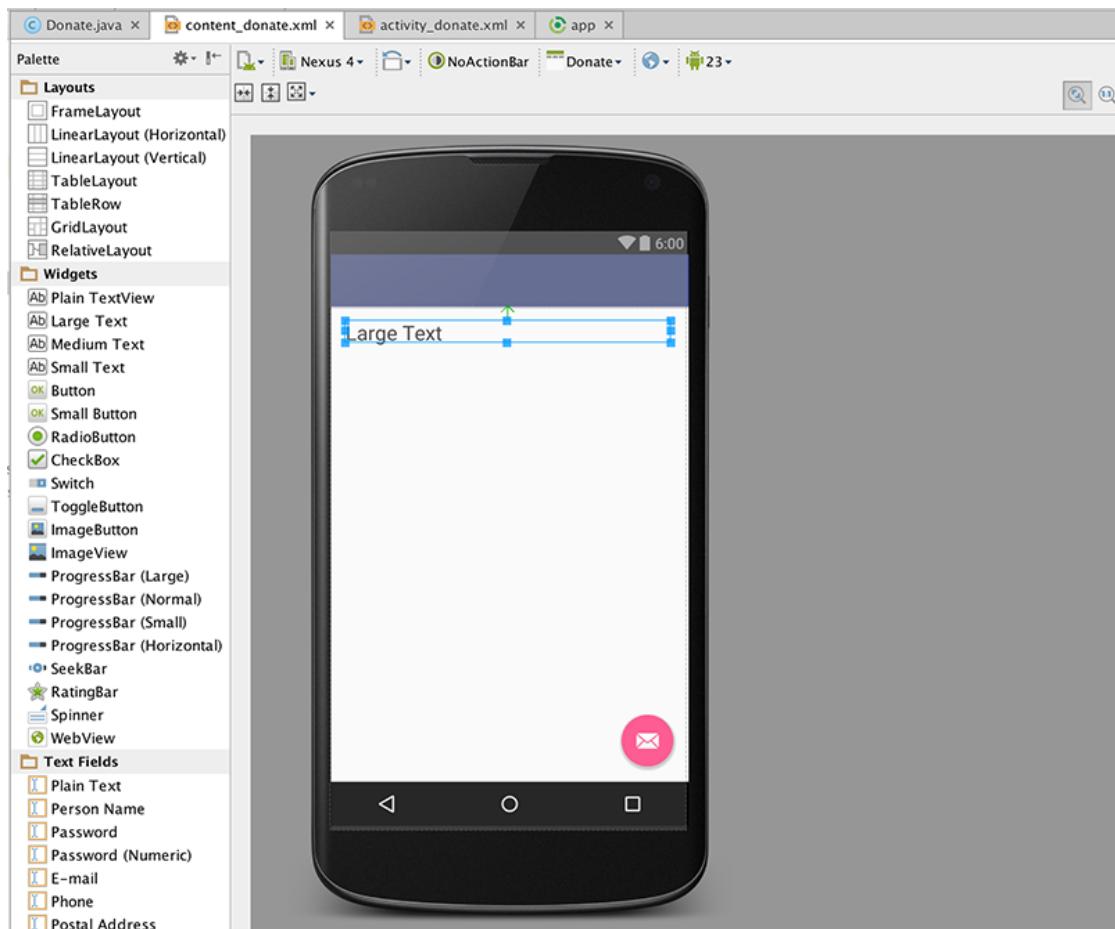
```

## Bước 2. Thiết kế giao diện Activity Donation

- Mục tiêu của bước này là thiết kế được giao diện Activity Donation giống như sau:



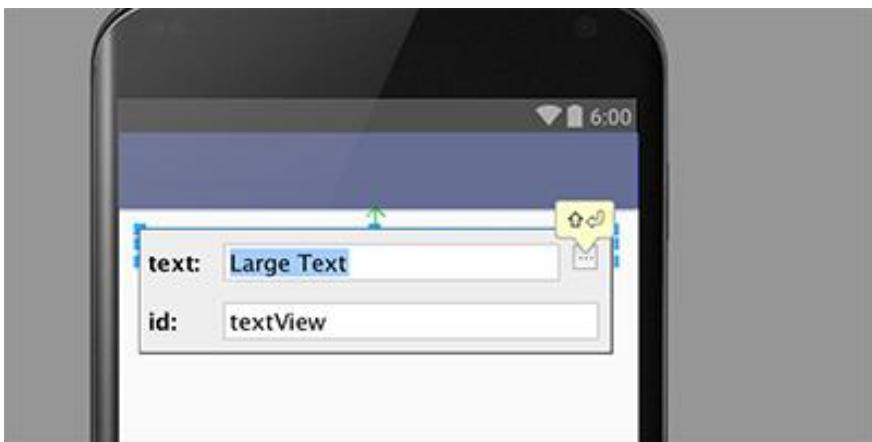
- Trong tệp **content\_donate.xml**, xóa chữ "Hello world" mặc định, kéo thả widget "LargeText" vào màn hình chính, kéo giãn widget sao cho vừa với kích thước màn hình như bên dưới.



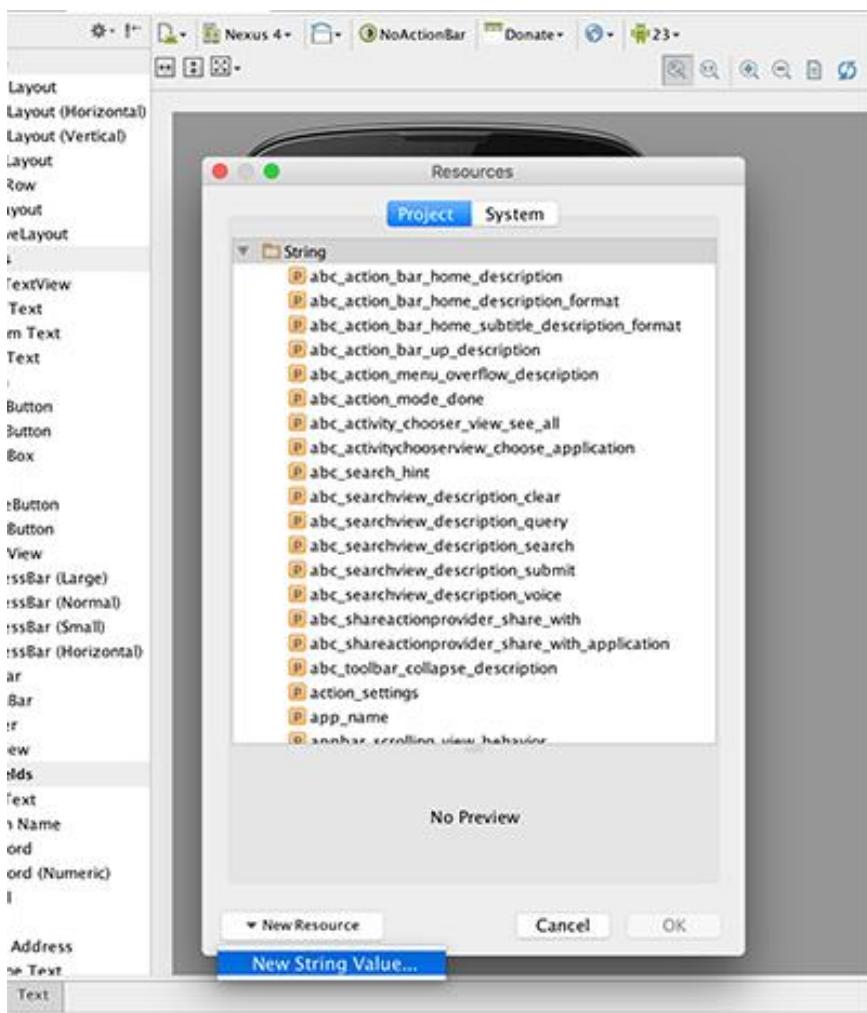
- Nháy đúp chuột vào widget, thông tin bên dưới sẽ hiện ra



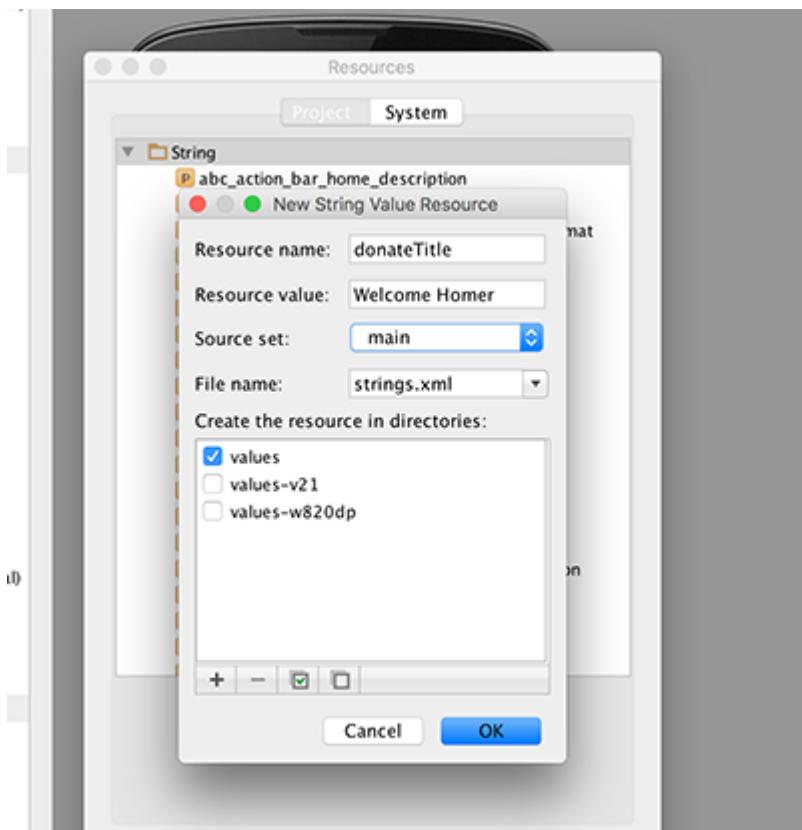
- Nhấn vào ô vuông có hình dấu ba chấm "..." phía bên tay phải



và menu Resource hiện ra



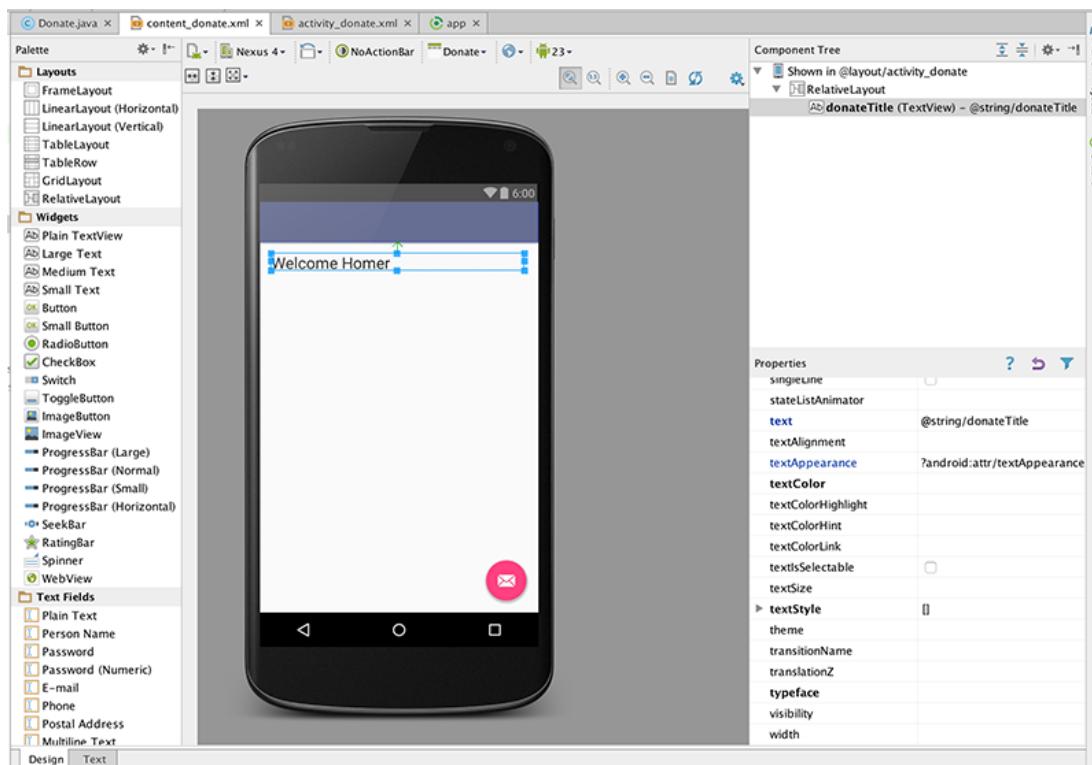
- Chọn *New Resource* -> *New String Value*, điền vào các giá trị như ảnh dưới



- Nháy đúp vào widget một lần nữa và điền "donateTitle" vào ô **id** và nhấn **Enter**



- Khi kết thúc, bạn sẽ có giao diện như sau



- Một số chú ý:
  - Trong Outline, tên của thành phần điều khiển được chuyển từ mặc định sang "donateTitle"
  - Trong Properties, kiểm tra giá trị của "text", chú ý tham chiếu của xâu ký tự được lưu trong **strings.xml**

```
Edit translations for all locales in the translations editor.

1 <resources>
2   <string name="app_name">Donation.1.0</string>
3   <string name="action_settings">Settings</string>
4   <string name="donateTitle">Welcome Homer</string>
5 </resources>
6
```

- Xác định vị trí hai tệp dưới đây và xem nội dung hai tệp

### **res/layout/content\_donate.xml**

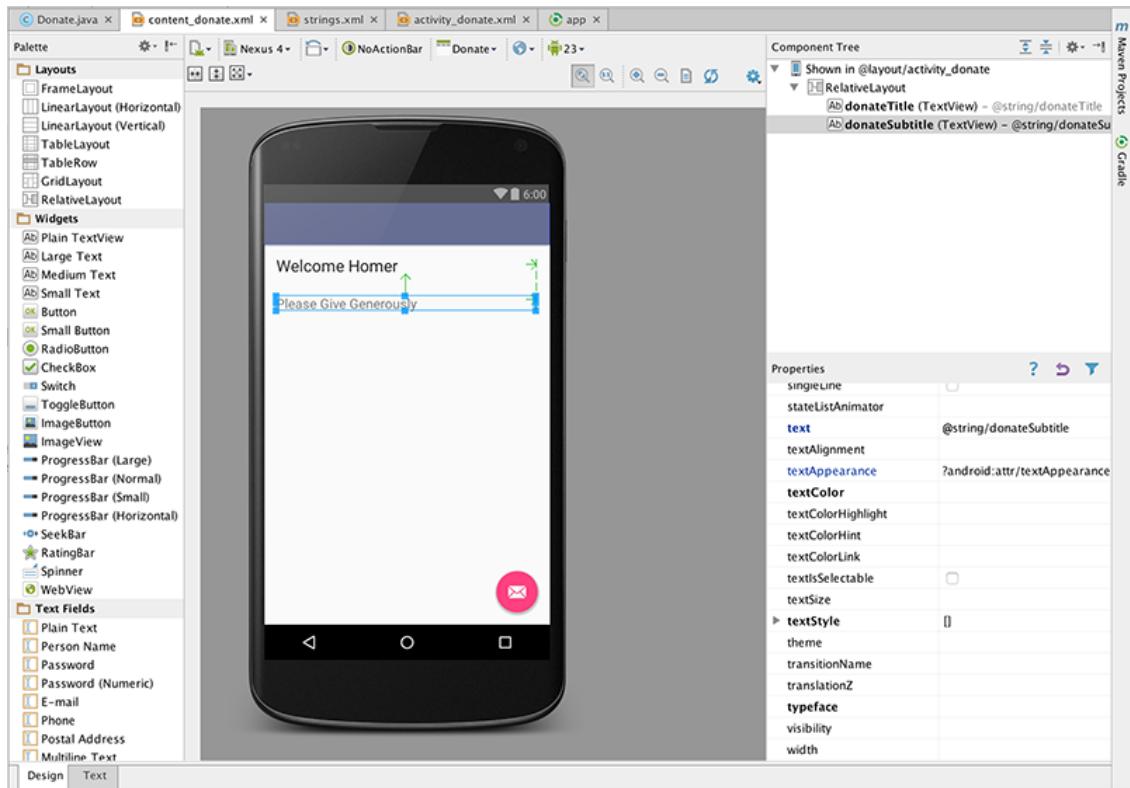
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:showIn="@layout/activity_donate" tools:context=".Donate">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/donateTitle"
        android:id="@+id/donateTitle"
        android:layout_alignParentTop="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true" />
</RelativeLayout>
```

### **res/values/strings.xml**

```
<resources>
    <string name="app_name">Donation</string>
    <string name="action_settings">Settings</string>
    <string name="donateTitle">Welcome Homer</string>
</resources>
```

- Chú ý mối quan hệ giữa "donateTitle" trong cả 2 tệp
- Đưa thêm xâu ký tự (medium text) sau vào activity Donate và làm theo các bước tương tự như trên; giao diện sẽ như sau:



và các tệp XML sẽ như sau:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:showIn="@layout/activity_donate" tools:context=".Donate">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/donateTitle"
        android:id="@+id/donateTitle"
        android:layout_alignParentTop="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/donateSubtitle" />

```

```

        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="@string/donateSubtitle"
        android:id="@+id/donateSubtitle"
        android:layout_below="@+id/donateTitle"
        android:layout_alignParentStart="true"
        android:layout_marginTop="27dp"
        android:layout_alignEnd="@+id/donateTitle" />
    </RelativeLayout>

```

tệp **strings.xml** như sau

```

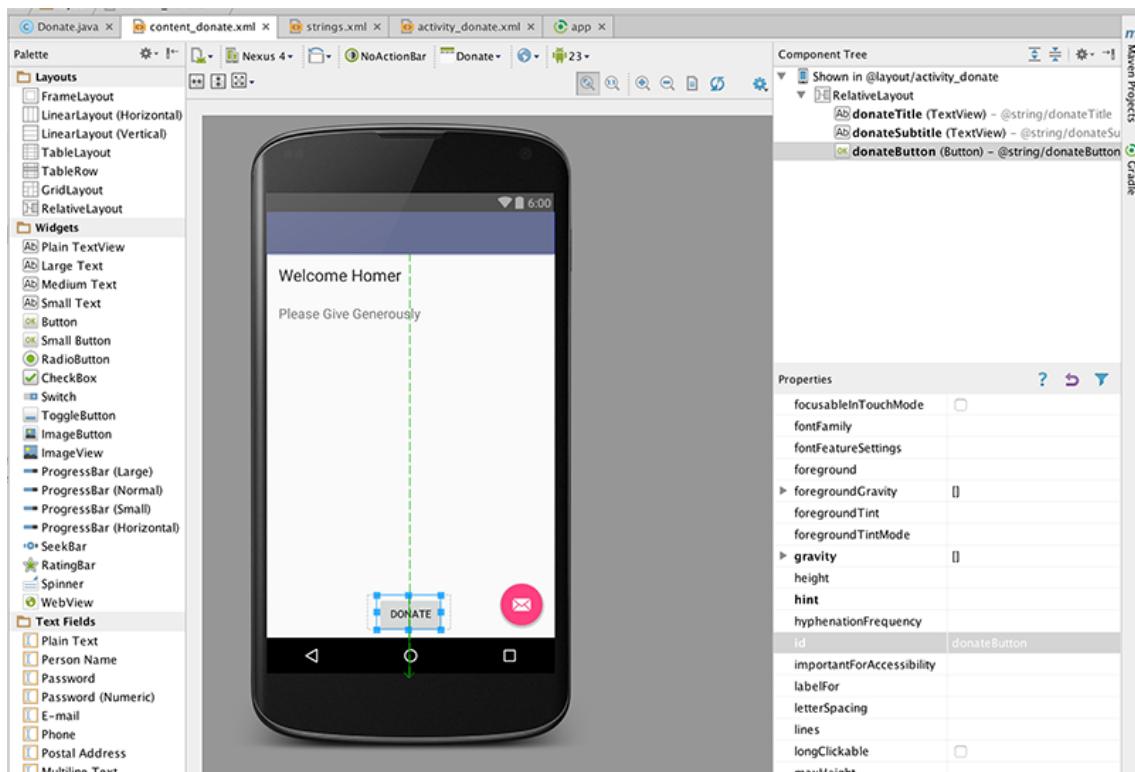
<resources>
    <string name="app_name">Donation.1.0</string>
    <string name="action_settings">Settings</string>
    <string name="donateTitle">Welcome Homer</string>
    <string name="donateSubtitle">Please Give Generously</string>
</resources>

```

## BÀI THỰC HÀNH 02: Giới thiệu ứng dụng Donation (tiếp)

### Bước 3. Thiết kế nút "Donate"

- Đặt một nút nhấn ở trên Activity, phía dưới màn hình như sau:



- Tương tự, đổi tên nút nhấn và thêm **id** "donateButton"; sau khi thành công, các tệp XML sẽ như sau:

## activity\_donate.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:showIn="@layout/activity_donate" tools:context=".Donate">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/donateTitle"
        android:id="@+id/donateTitle"
        android:layout_alignParentTop="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true" />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceMedium"
        android:text="@string/donateSubtitle"
        android:id="@+id/donateSubtitle"
        android:layout_below="@+id/donateTitle"
        android:layout_alignParentStart="true"
        android:layout_marginTop="27dp"
        android:layout_alignEnd="@+id/donateTitle" />

    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/donateButton"
        android:id="@+id/donateButton"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="47dp" />
</RelativeLayout>
```

## strings.xml

```
<resources>
    <string name="app_name">Donation.1.0</string>
    <string name="action_settings">Settings</string>
    <string name="donateTitle">Welcome Homer</string>
    <string name="donateSubtitle">Please Give Generously</string>
    <string name="donateButton">Donate</string>
</resources>
```

- Chúng ta tiếp tục với activity Donate:

```
package ie.app;
import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;

public class Donate extends AppCompatActivity {

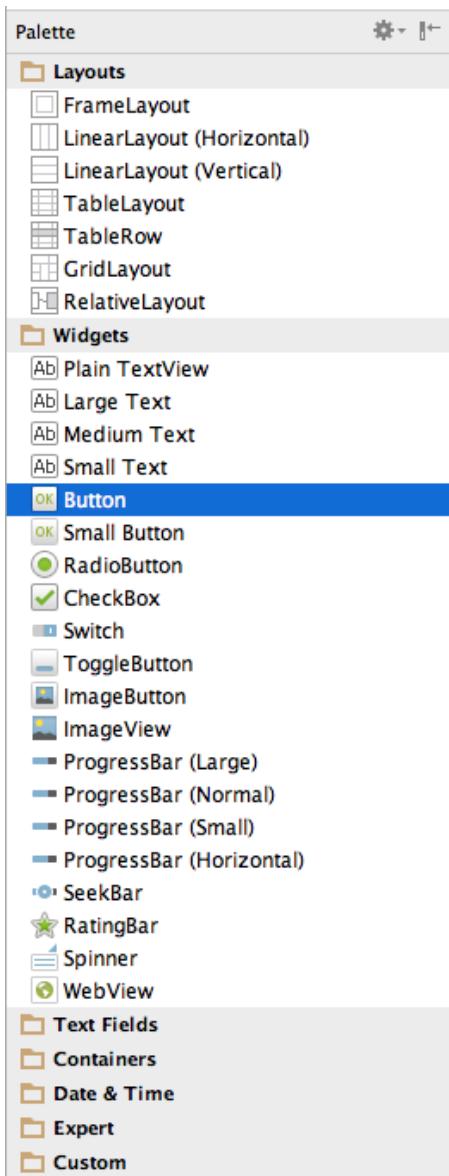
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_donate);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action",
                        Snackbar.LENGTH_LONG)
                        .setAction("Action", null).show();
            }
        });
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_donate, menu);
        return true;
    }

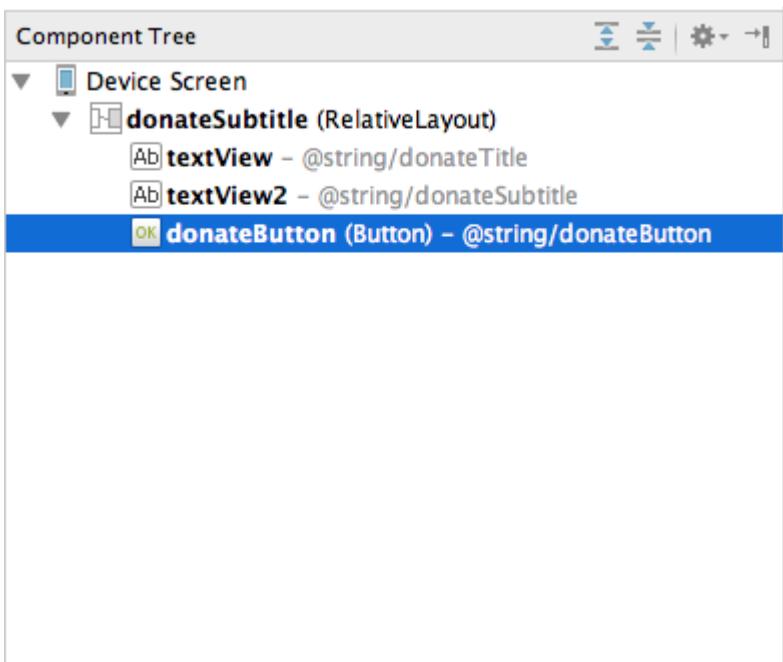
    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
```

```
// Handle action bar item clicks here. The action bar will
// automatically handle clicks on the Home/Up button, so long
// as you specify a parent activity in AndroidManifest.xml.
int id = item.getItemId();
//noinspection SimplifiableIfStatement
if (id == R.id.action_settings) {
    return true;
}
return super.onOptionsItemSelected(item);
}
```

- Mỗi phần tử có thể được tương tác bởi người dùng, thường có một lớp tương ứng được định nghĩa trong mã nguồn. Hiện tại các trường văn bản được coi như không được tương tác thường xuyên bởi người dùng.
- Thêm trường mới như sau vào class:  
`private Button donateButton;`
- Class sẽ được import vào, tên của class sẽ trùng với tên trong Palette



- Để tránh nhầm lẫn, luôn luôn gọi biến với tên được sử dụng trong Outline view:



- Trong phương thức **onCreate()**, chúng ta cần khởi tạo biến này  
donateButton = (Button) findViewById(R.id.donateButton);
- Chúng ta có thể thêm log để nhận phản hồi khi ứng dụng được chạy  

```
if (donateButton != null)
{
    Log.v("Donate", "Really got the donate button");
}
```
- Đây là class activity hoàn chỉnh

```
package ie.app;

import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Button;

public class Donate extends AppCompatActivity {

    private Button donateButton;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
```

```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_donate);
Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
setSupportActionBar(toolbar);

FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Snackbar.make(view, "Replace with your own action",
        Snackbar.LENGTH_LONG)
            .setAction("Action", null).show();
    }
});

donateButton = (Button) findViewById(R.id.donateButton);

if (donateButton != null)
{
    Log.v("Donate", "Really got the donate button");
}
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu_donate, menu);
    return true;
}

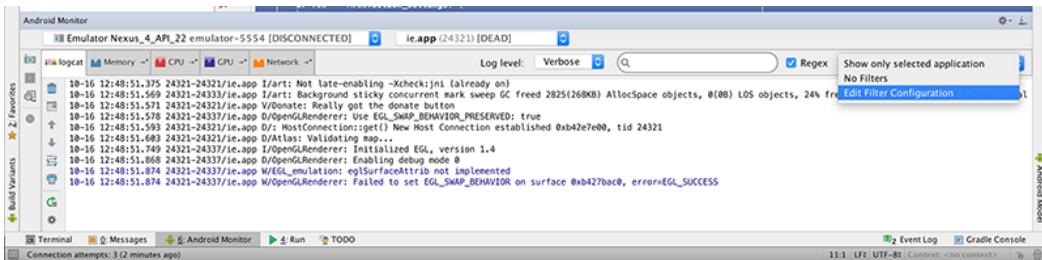
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }

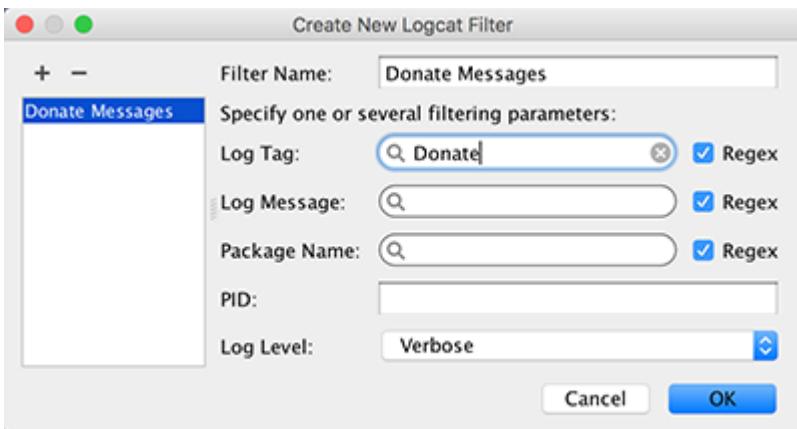
    return super.onOptionsItemSelected(item);
}
}

```

- Để tiện cho việc tìm kiếm tin nhắn trong log, tạo một Logcat filter như sau
- Chọn "Edit Filter Configuration" phía bên tay phải của LogCat View



- Nhập tên và tag như bên dưới



và nếu bạn chọn filter, tin nhắn sẽ hiển thị như sau:



- Khởi động lại ứng dụng, xác nhận tin nhắn vừa xuất hiện

## BÀI THỰC HÀNH 03: Giới thiệu ứng dụng Donation (tiếp)

### Bước 4. Sử dụng tài liệu Android

- Tài liệu Android được thiết kế rất tốt và rất hữu ích
  - <http://developer.android.com/guide/components/index.html>
  - <http://developer.android.com/reference/packages.html>
- Với class Button chúng ta vừa sử dụng, chúng ta có thể tìm thấy tài liệu tương ứng dựa vào câu lệnh import

`import android.widget.Button;`

tương ứng với ...

- <http://developer.android.com/reference/android/widget/Button.html>

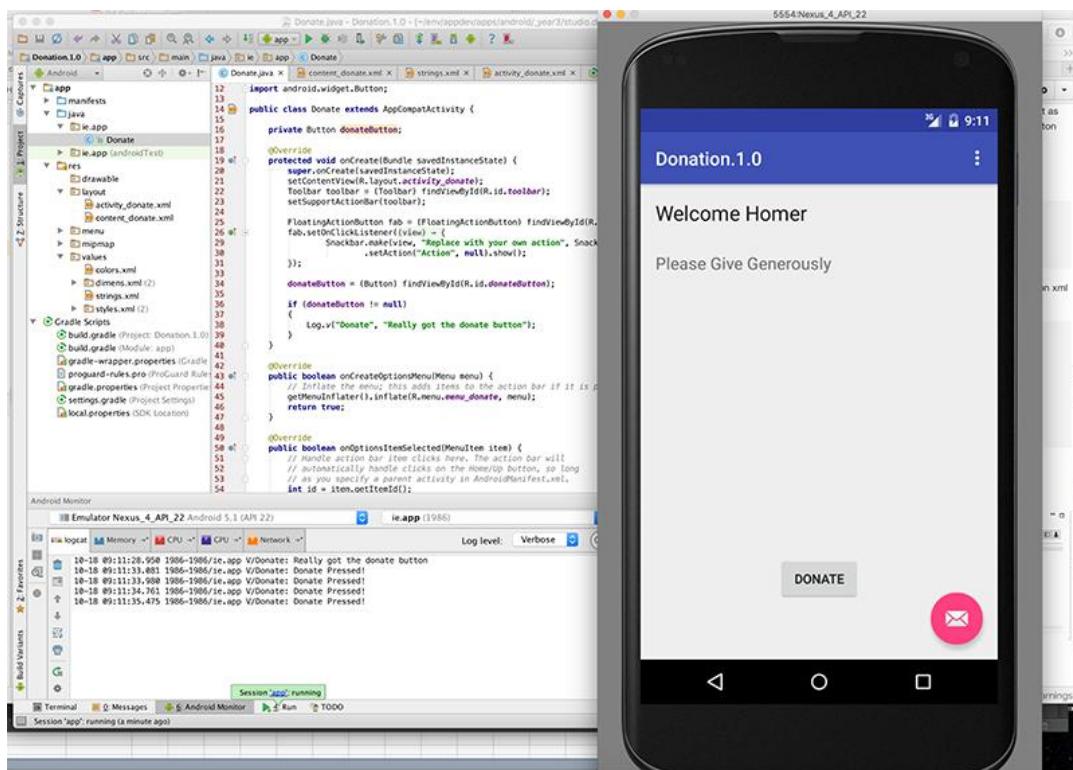
- Tạo một phương thức mới trong class Donate

```
public void donateButtonPressed (View view)
{
    Log.v("Donate", "Donate Pressed!");
}
```

- Sau đó sửa tệp **content\_donate.xml** và thêm vào một thuộc tính mới trong phần Button tên là "onclick"

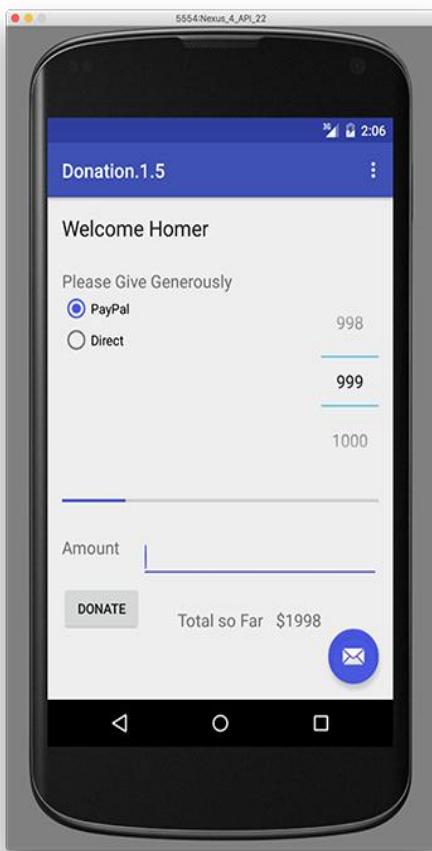
```
<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/donateButton"
    android:id="@+id/donateButton"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="47dp"
    android:onClick="donateButtonPressed"/>
```

- Lưu lại chỉnh sửa và khởi động ứng dụng, theo dõi log khi bạn nhấn nút "Donate"

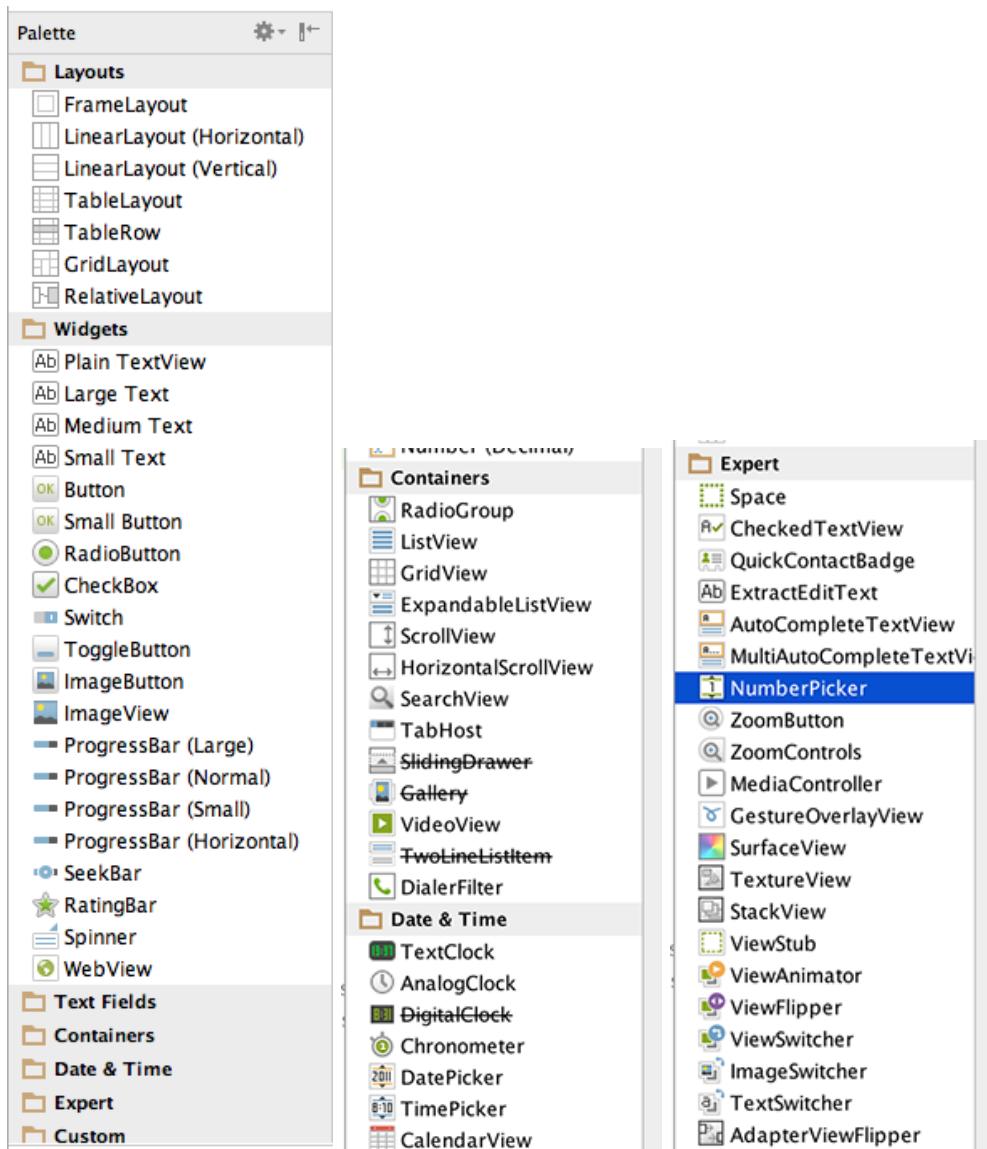


## Bước 5. Thiết kế giao diện điều khiển mới

- Giao diện ứng dụng xây dựng



- Chúng ta cần radio buttons, selection/combo box và một progress bar. Những thành phần có thể được tìm thấy trong Pallette.



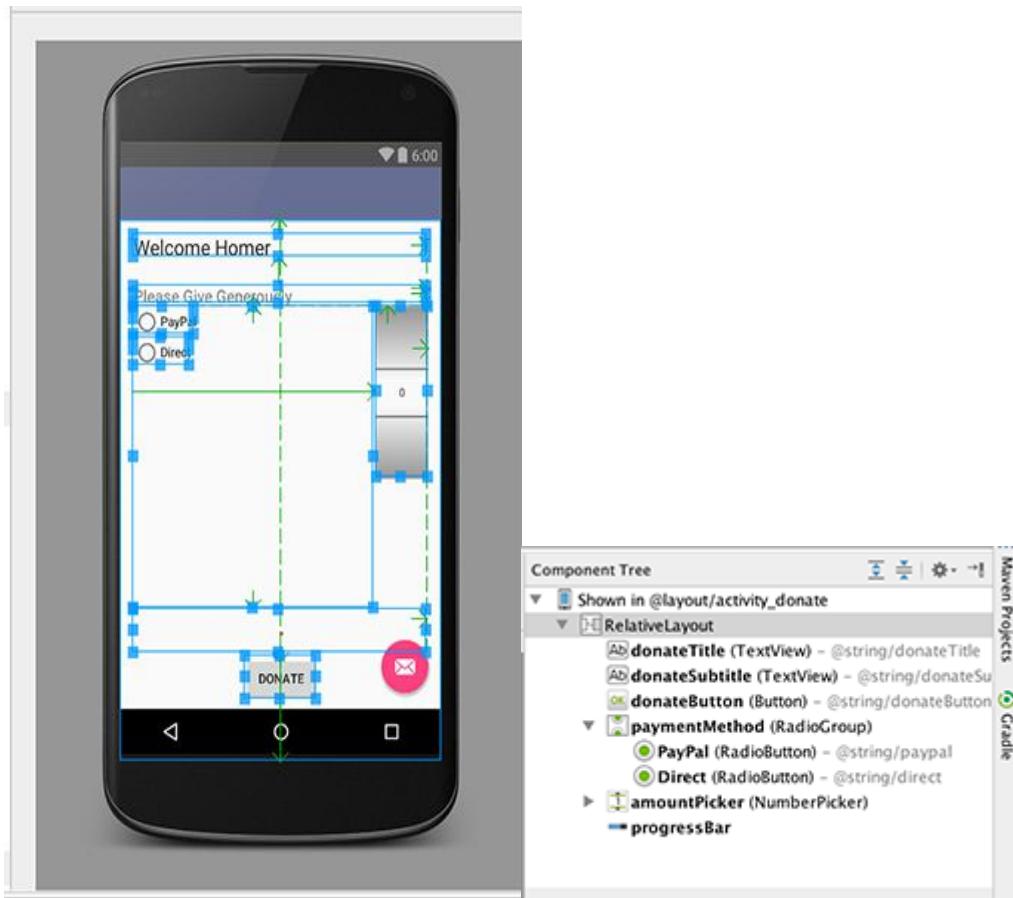
- RadioGroup, ProgressBar and NumberPicker là những đối tượng chúng ta sẽ sử dụng. Import những đối tượng này vào đầu class **Donate**

```
import android.widget.RadioGroup;
import android.widget.NumberPicker;
import android.widget.ProgressBar;
```

... chúng ta thêm các trường tương ứng vào class

```
private RadioGroup paymentMethod;
private ProgressBar progressBar;
private NumberPicker amountPicker;
```

- Tương tự, chúng ta có thể tìm thấy tài liệu Android về 3 đối tượng trên
  - <http://developer.android.com/reference/android/widget/RadioGroup.html>
  - <http://developer.android.com/reference/android/widget/ProgressBar.html>
  - <http://developer.android.com/reference/android/widget/NumberPicker.html>



- Một số tệp XML tham khảo

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    app:layout_behavior="@string/appbar_scrolling_view_behavior"
    tools:showIn="@layout/activity_donate" tools:context=".Donate">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/donateTitle"
        android:id="@+id/donateTitle"
        android:layout_alignParentTop="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true" />
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceMedium"
    android:text="@string/donateSubtitle"
    android:id="@+id/donateSubtitle"
    android:layout_below="@+id/donateTitle"
    android:layout_alignParentStart="true"
    android:layout_marginTop="27dp"
    android:layout_alignEnd="@+id/donateTitle" />

<Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/donateButton"
    android:id="@+id/donateButton"
    android:layout_alignParentBottom="true"
    android:layout_centerHorizontal="true"
    android:layout_marginBottom="47dp"
    android:onClick="donateButtonPressed"/>

<RadioGroup
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_below="@+id/donateSubtitle"
    android:layout_alignParentStart="true"
    android:id="@+id/paymentMethod"
    android:layout_toStartOf="@+id/amountPicker"
    android:layout_above="@+id/progressBar">

    <RadioButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/paypal"
        android:id="@+id/PayPal"
        android:checked="false" />

    <RadioButton
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/direct"
        android:id="@+id/Direct"
        android:checked="false" />
</RadioGroup>

<NumberPicker
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/amountPicker"
    android:layout_alignTop="@+id/paymentGroup"
    android:layout_alignEnd="@+id/donateSubtitle" />
```

```

<ProgressBar
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    style="?android:attr/progressBarStyleHorizontal"
    android:id="@+id/progressBar"
    android:layout_alignParentStart="true"
    android:layout_alignEnd="@+id/donateSubtitle"
    android:layout_above="@+id/donateButton"
    android:indeterminate="false" />

</RelativeLayout>

<resources>
    <string name="app_name">Donation.1.0</string>
    <string name="action_settings">Settings</string>
    <string name="donateTitle">Welcome Homer</string>
    <string name="donateSubtitle">Please Give Generously</string>
    <string name="donateButton">Donate</string>
    <string name="paypal">PayPal</string>
    <string name="direct">Direct</string>
</resources>

```

- Chúng ta có thể liên kết các điều khiển này trong *onCreate()*

```

        paymentMethod = (RadioGroup) findViewById(R.id.paymentMethod);
        progressBar = (ProgressBar) findViewById(R.id.progressBar);
        amountPicker = (NumberPicker) findViewById(R.id.amountPicker);

```

- Đây là class *Donate* hoàn chỉnh đến thời điểm này

```

package ie.app;

import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Button;
import android.widget.NumberPicker;
import android.widget.ProgressBar;
import android.widget.RadioGroup;

```

```

public class Donate extends AppCompatActivity {

    private Button      donateButton;
    private RadioGroup   paymentMethod;
    private ProgressBar  progressBar;

```

```
private NumberPicker amountPicker;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_donate);
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);

    FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
    fab.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Snackbar.make(view, "Replace with your own action",
                    Snackbar.LENGTH_LONG)
                    .setAction("Action", null).show();
        }
    });
}

donateButton = (Button) findViewById(R.id.donateButton);

if (donateButton != null)
{
    Log.v("Donate", "Really got the donate button");
}

paymentMethod = (RadioGroup) findViewById(R.id.paymentMethod);
progressBar = (ProgressBar) findViewById(R.id.progressBar);
amountPicker = (NumberPicker) findViewById(R.id.amountPicker);

amountPicker.setMinValue(0);
amountPicker.setMaxValue(1000);
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu_donate, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }
}
```

```

        return super.onOptionsItemSelected(item);
    }

    public void donateButtonPressed (View view)
    {
        Log.v("Donate", "Donate Pressed!");
    }
}

```

## BÀI THỰC HÀNH 04: Giới thiệu ứng dụng Donation (tiếp)

### Bước 6. Sử dụng NumberPicker

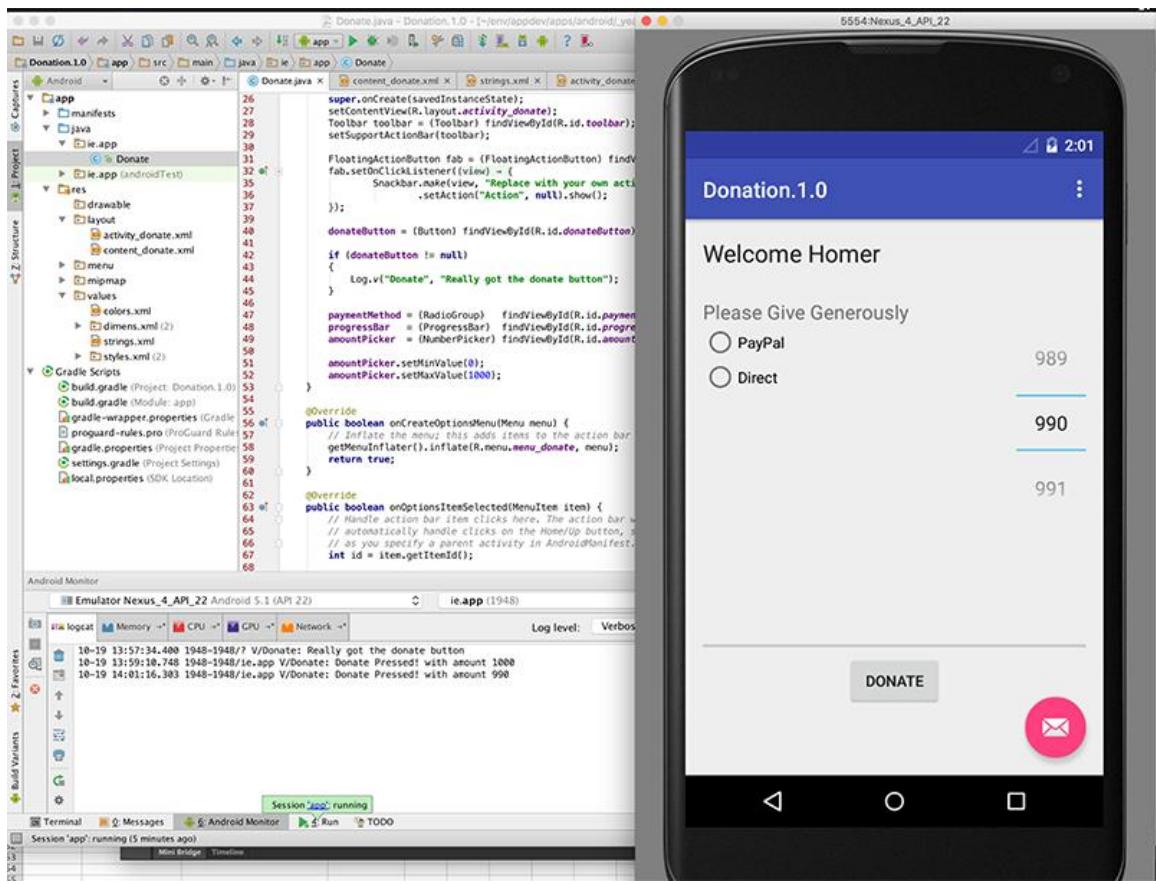
- Một số tài liệu tham khảo
  - <http://developer.android.com/reference/android/widget/NumberPicker.html>
  - <http://developer.android.com/guide/components/index.html>
  - <http://developer.android.com/guide/topics/ui/controls.html>
  - <http://developer.android.com/guide/topics/ui/pickers.html>
- Tài liệu này liên quan tới Fragments, date and time pickers. Bắt đầu với 3 phương thức sau:
  - [http://developer.android.com/reference/android/widget/NumberPicker.html#setMaxValue\(int\)](http://developer.android.com/reference/android/widget/NumberPicker.html#setMaxValue(int))
  - [http://developer.android.com/reference/android/widget/NumberPicker.html#setMinValue\(int\)](http://developer.android.com/reference/android/widget/NumberPicker.html#setMinValue(int))
  - [http://developer.android.com/reference/android/widget/NumberPicker.html#getValue\(\)](http://developer.android.com/reference/android/widget/NumberPicker.html#getValue())
- Trong phương thức onCreate(), khởi tạo các giá trị:

```
amountPicker.setMinValue(0);
amountPicker.setMaxValue(1000);
```

và trong donateButtonPressed():

```
public void donateButtonPressed (View view)
{
    int amount = amountPicker.getValue();
    Log.v("Donate", "Donate Pressed! with amount " + amount);
}
```

- Chạy ứng dụng và kiểm tra số tiền hiển thị và trong log có trùng nhau hay không



## Bước 7: Sử dụng Radio Buttons

- Trong phương thức `donateButtonPressed()`
  - <http://developer.android.com/reference/android/widget/RadioGroup.html>
- Phương thức chúng ta cần sử dụng
  - `getCheckedRadioButtonId()`
- Đây là một phiên bản sửa đổi của `donateButtonPressed()`

```
public void donateButtonPressed (View view)
{
    int amount = amountPicker.getValue();
    int radioId = paymentMethod.getCheckedRadioButtonId();
    String method = "";
    if (radioId == R.id.PayPal)
    {
        method = "PayPal";
    }
    else
    {
        method = "Direct";
    }
    Log.v("Donate", "Donate Pressed! with amount " + amount + ", method: " + method);
}
```

- Chạy ứng dụng và kiểm tra log
- Đơn giản hóa câu lệnh kiểm tra (tham khảo  
<http://marxsoftware.blogspot.ie/2010/09/how-i-learned-to-stop-worrying-and-love.html>)

```
String method = radioId == R.id.PayPal ? "PayPal" : "Direct";
```

- Đây là class **Donate** cho đến lúc này

```
package ie.app;

import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Button;
import android.widget.NumberPicker;
import android.widget.ProgressBar;
import android.widget.RadioGroup;
```

```
public class Donate extends AppCompatActivity {

    private Button      donateButton;
    private RadioGroup   paymentMethod;
    private ProgressBar  progressBar;
    private NumberPicker amountPicker;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_donate);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action",
                        Snackbar.LENGTH_LONG)
                        .setAction("Action", null).show();
            }
        });

        donateButton = (Button) findViewById(R.id.donateButton);

        if (donateButton != null)
        {
```

```

        Log.v("Donate", "Really got the donate button");
    }

    paymentMethod = (RadioGroup) findViewById(R.id.paymentMethod);
    progressBar = (ProgressBar) findViewById(R.id.progressBar);
    amountPicker = (NumberPicker) findViewById(R.id.amountPicker);

    amountPicker.setMinValue(0);
    amountPicker.setMaxValue(1000);
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu_donate, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

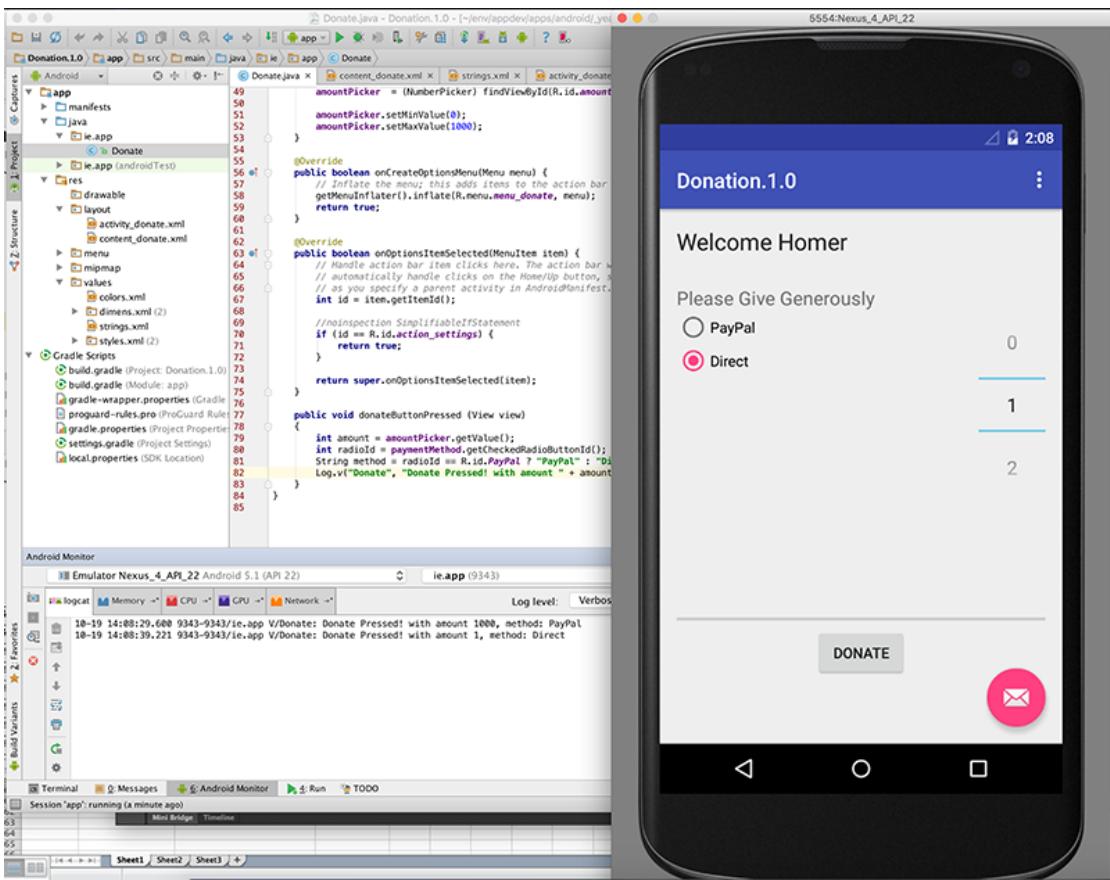
    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }

    return super.onOptionsItemSelected(item);
}

public void donateButtonPressed (View view)
{
    int amount = amountPicker.getValue();
    int radioId = paymentMethod.getCheckedRadioButtonId();
    String method = radioId == R.id.PayPal ? "PayPal" : "Direct";
    Log.v("Donate", "Donate Pressed! with amount " + amount + ", method: " +
method);
}
}

```

- Chạy ứng dụng và kiểm tra log



## Bước 8. Sử dụng Progress Bar

- Tài liệu tham khảo
  - <http://developer.android.com/reference/android/widget/ProgressBar.html>
  - [http://developer.android.com/reference/android/widget/ProgressBar.html#setMax\(int\)](http://developer.android.com/reference/android/widget/ProgressBar.html#setMax(int))
  - [http://developer.android.com/reference/android/widget/ProgressBar.html#setProgress\(int\)](http://developer.android.com/reference/android/widget/ProgressBar.html#setProgress(int))

- Đầu tiên, chúng ta cần bổ sung tính năng khi nhớ số tiền quyên góp

```
private int totalDonated = 0;
```

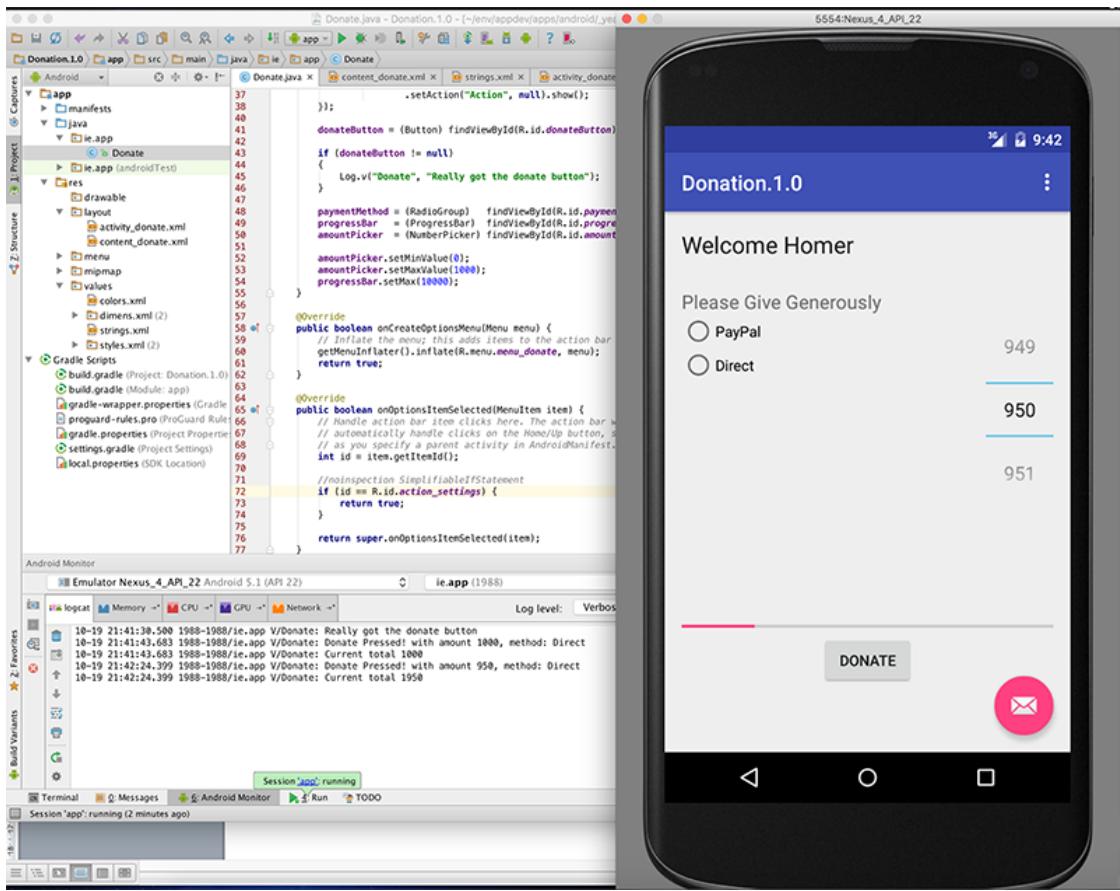
- Thiết lập giới hạn progress bar là 10000 trong phương thức `onCreate()`

```
progressBar.setMax(10000);
```

và thiết lập progress trong `donateButtonPressed()`

```
totalDonated = totalDonated + amount;
progressBar.setProgress(totalDonated);
Log.v("Donate", "Donate Pressed! with amount " + amount + ", method: " + method);
Log.v("Donate", "Current total " + totalDonated);
```

- Chạy ứng dụng, quan sát progress bar và logCat



- Đây là class **Donate** cho tới thời điểm này

```
package ie.app;

import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Button;
import android.widget.NumberPicker;
import android.widget.ProgressBar;
import android.widget.RadioGroup;

public class Donate extends AppCompatActivity {

    private Button donateButton;
    private RadioGroup paymentMethod;
    private ProgressBar progressBar;
    private NumberPicker amountPicker;
```

```
private int totalDonated = 0;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_donate);
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);

    FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
    fab.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Snackbar.make(view, "Replace with your own action",
                    Snackbar.LENGTH_LONG)
                    .setAction("Action", null).show();
        }
    });

    donateButton = (Button) findViewById(R.id.donateButton);

    if (donateButton != null)
    {
        Log.v("Donate", "Really got the donate button");
    }

    paymentMethod = (RadioGroup) findViewById(R.id.paymentMethod);
    progressBar = (ProgressBar) findViewById(R.id.progressBar);
    amountPicker = (NumberPicker) findViewById(R.id.amountPicker);

    amountPicker.setMinValue(0);
    amountPicker.setMaxValue(1000);
    progressBar.setMax(10000);
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu_donate, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle action bar item clicks here. The action bar will
    // automatically handle clicks on the Home/Up button, so long
    // as you specify a parent activity in AndroidManifest.xml.
    int id = item.getItemId();

    //noinspection SimplifiableIfStatement
    if (id == R.id.action_settings) {
        return true;
    }
}
```

```

        }

        return super.onOptionsItemSelected(item);
    }

    public void donateButtonPressed (View view)
    {
        int amount = amountPicker.getValue();
        int radioId = paymentMethod.getCheckedRadioButtonId();
        String method = radioId == R.id.PayPal ? "PayPal" : "Direct";

        totalDonated = totalDonated + amount;
        progressBar.setProgress(totalDonated);

        Log.v("Donate", "Donate Pressed! with amount " + amount + ", method: " +
method);
        Log.v("Donate", "Current total " + totalDonated);
    }
}

```

- Đây là một phiên bản khác của class **Donate**

```

package ie.app;

import android.os.Bundle;
import android.support.design.widget.FloatingActionButton;
import android.support.design.widget.Snackbar;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.View;
import android.view.Menu;
import android.view.MenuItem;
import android.widget.Button;
import android.widget.NumberPicker;
import android.widget.ProgressBar;
import android.widget.RadioGroup;

```

```

public class Donate extends AppCompatActivity {

    private Button      donateButton;
    private RadioGroup   paymentMethod;
    private ProgressBar  progressBar;
    private NumberPicker amountPicker;

    private int          totalDonated = 0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_donate);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);

```

```
        setSupportActionBar(toolbar);

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action",
                        Snackbar.LENGTH_LONG)
                        .setAction("Action", null).show();
            }
        });

        donateButton = (Button) findViewById(R.id.donateButton);

        if (donateButton != null)
        {
            Log.v("Donate", "Really got the donate button");
        }

        paymentMethod = (RadioGroup) findViewById(R.id.paymentMethod);
        progressBar = (ProgressBar) findViewById(R.id.progressBar);
        amountPicker = (NumberPicker) findViewById(R.id.amountPicker);

        amountPicker.setMinValue(0);
        amountPicker.setMaxValue(1000);
        progressBar.setMax(10000);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_donate, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

        return super.onOptionsItemSelected(item);
    }

    public void donateButtonPressed (View view)
    {
```

```
totalDonated = totalDonated + amountPicker.getValue();
String method = paymentMethod.getCheckedRadioButtonId() == R.id.PayPal ?
"PayPal" : "Direct";
progressBar.setProgress(totalDonated);

Log.v("Donate", amountPicker.getValue() + " donated by " + method + "\nCurrent
total " + totalDonated);
}
}
```

- Xem xét kỹ và đưa ra sự khác nhau giữa hai cách cài đặt trên và lý giải

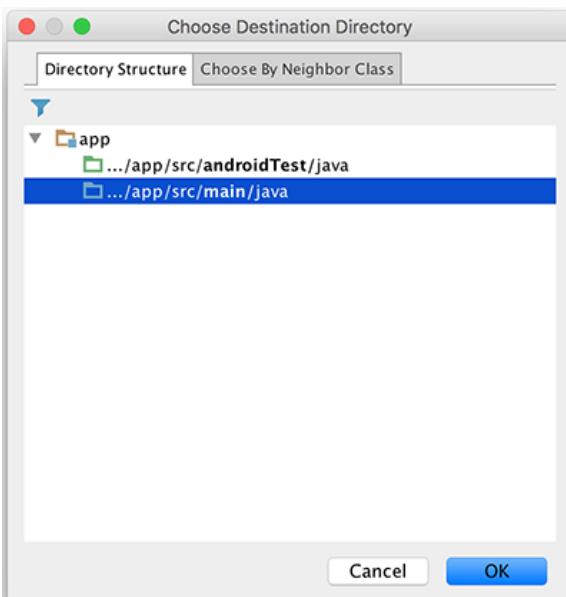
## BÀI THỰC HÀNH 05: Multi Screen App & Menus

**Mục tiêu:** Xây dựng ứng dụng **Donation ver2.0**

- Giới thiệu menu
- Hiển thị Report
- Sử dụng ArrayAdapter đơn giản

### Bước 1. Kiểm tra Package Name

- Đây là phiên bản hiện tại của ứng dụng Donation
  - [Donation.1.5.zip](#)
- Tiến hành refactor ứng dụng hiện tại thành phiên bản **Donation 2.0**
  1. Đảm bảo ứng dụng **Donation 1.5** đang không chạy trên Android Studio, nếu có thì đóng lại
  2. Di chuyển tới folder chứa **Donation.1.5** đã giải nén
  3. Đổi tên folder thành **Donation.2.0**
  4. Đổi tên tệp **Donation.1.5.iml** thành **Donation.2.0.iml**
  5. Trong tệp **Donation.2.0.iml**, đổi các tham chiếu từ **Donation.1.5** thành **Donation.2.0**
  6. Di chuyển tới thư mục **.idea**, mở tệp **.name** và đổi project name thành **Donation.2.0**
  7. Mở Android Studio và mở project **Donation.2.0**
- Đổi tên package "**ie.app**" thành "**ie.app.activities**" sau đó tạo package mới trong folder main



- Kéo tệp **Donate.java** vào package vừa tạo rồi xóa package trống đi. Làm tương tự đối với class **ApplicationTest**. Kiểm tra các câu lệnh import trong **Donate.java** như sau

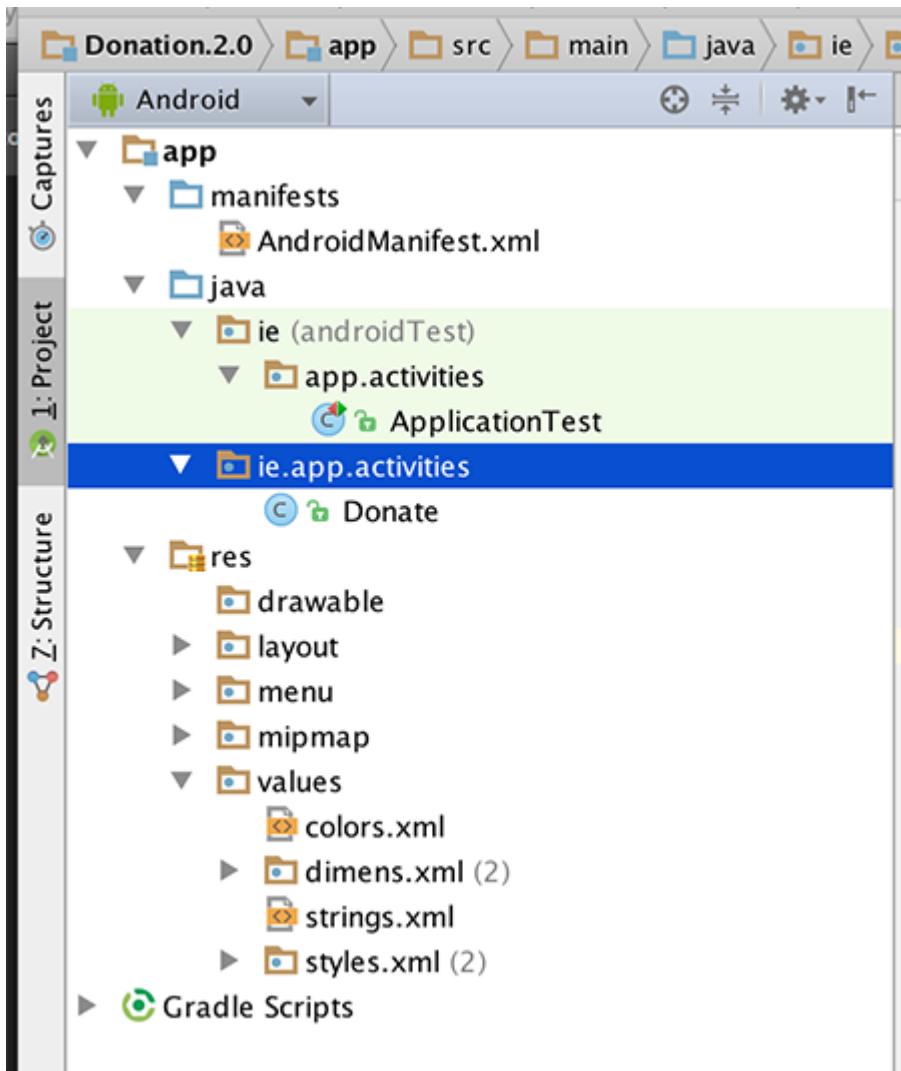
```

import android.widget.Toast;

import ie.app.R;

public class Donate extends AppCompatActivity

```



- Đảm bảo cấu trúc project giống như trên, đồng thời thay đổi **app\_name** trong tệp **strings.xml**.
- Chạy Build -> Clean Project -> Rebuild, sau đó chạy ứng dụng.

## Bước 2. Cài đặt Menu

- Mở tệp **res/values/strings.xml** và thêm vào dòng sau
 

```
<string name="menuReport">Report</string>
```
- Chúng ta có một menu trong **menu\_donate.xml** nằm trong thư mục **res/menu**, sửa tệp này và thêm vào một menu item mới như sau:

```

<item
    android:id="@+id/menuReport"
    android:orderInCategory="100"
    android:title="@string/menuReport"
    app:showAsAction="never"/>

```

- Trong **Donate.java**, sửa phương thức *onOptionsItemSelected()* như sau

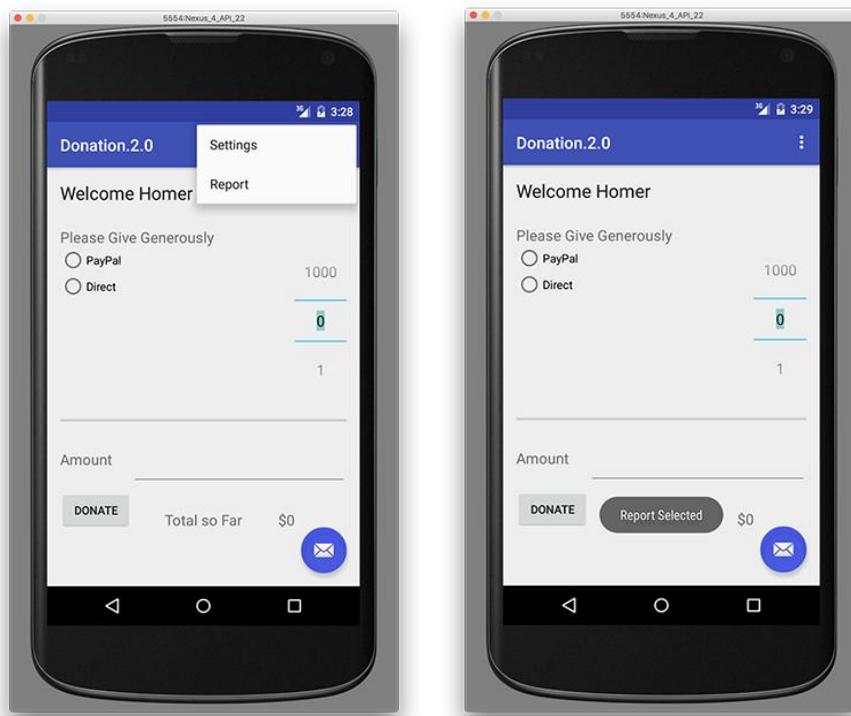
```

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.menuReport:
            Toast toast = Toast.makeText(this, "Report Selected",
                    Toast.LENGTH_SHORT);
            toast.show();
            break;
    }

    return super.onOptionsItemSelected(item);
}

```

- Chạy ứng dụng và nhấn vào menu, chọn **Report**, thông báo sẽ hiện ra



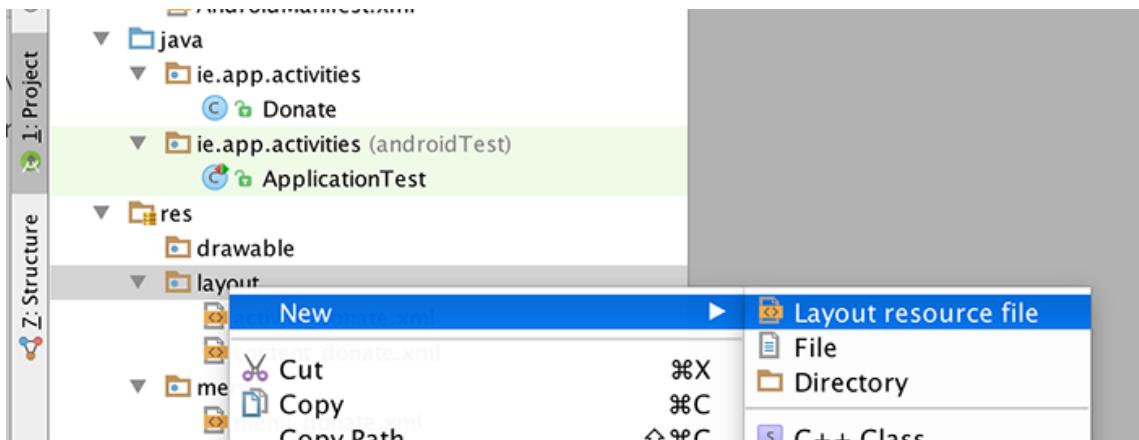
## BÀI THỰC HÀNH 06: Multi Screen App & Menus(tiếp)

### Bước 3: Tạo activity cho Reports

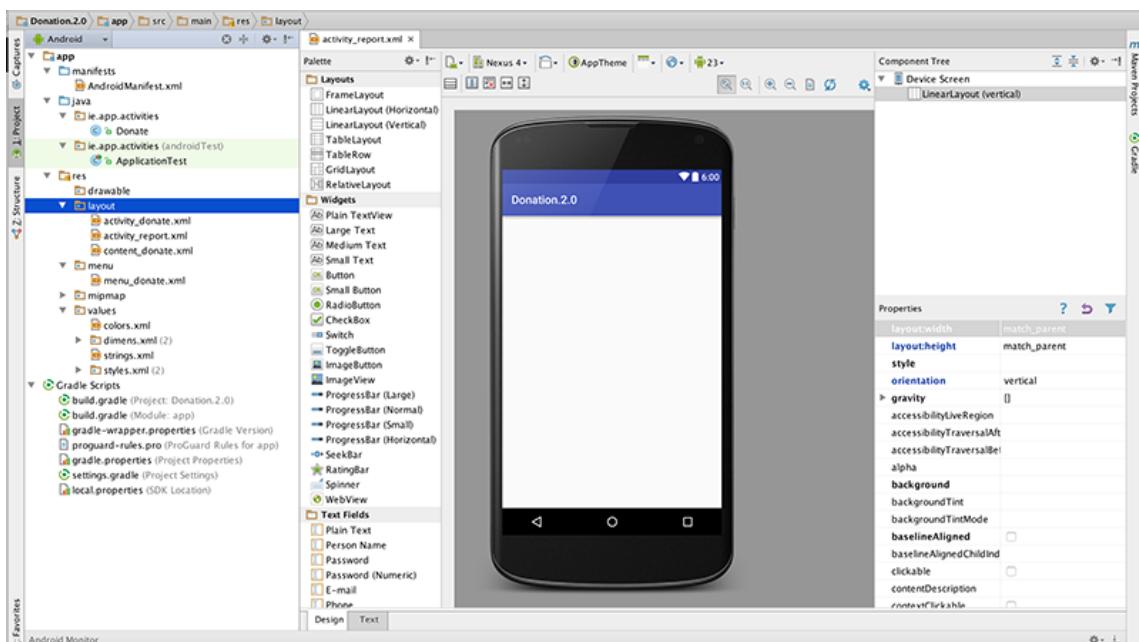
- Trước khi bắt đầu thiết kế activity mới, thêm dòng sau vào **res/values/strings.xml**

```
<string name="reportTitle">Report</string>
```

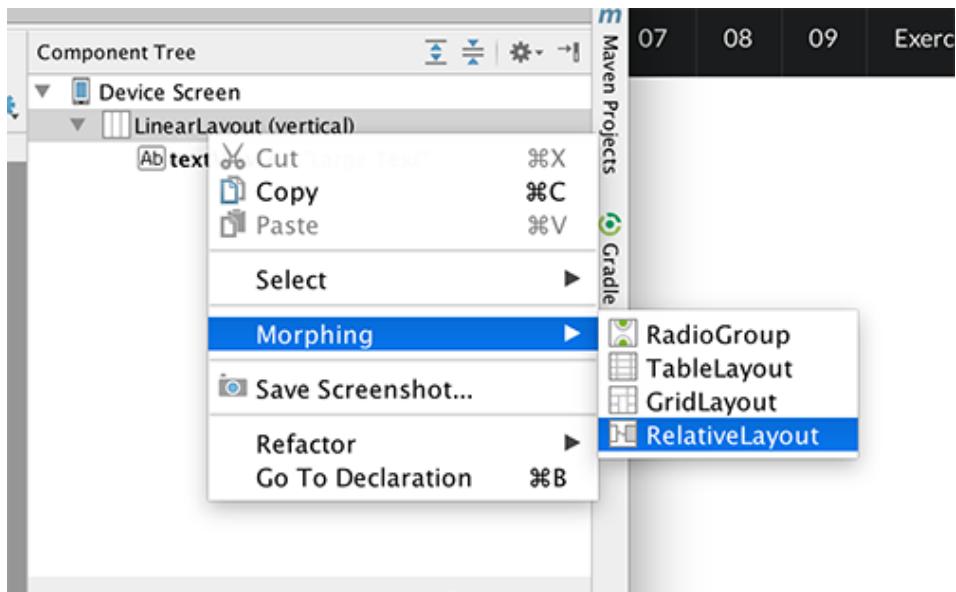
- Thiết kế layout mới **activity\_report** bằng cách xác định thư mục **res/layout** và chọn New -> Layout resource file



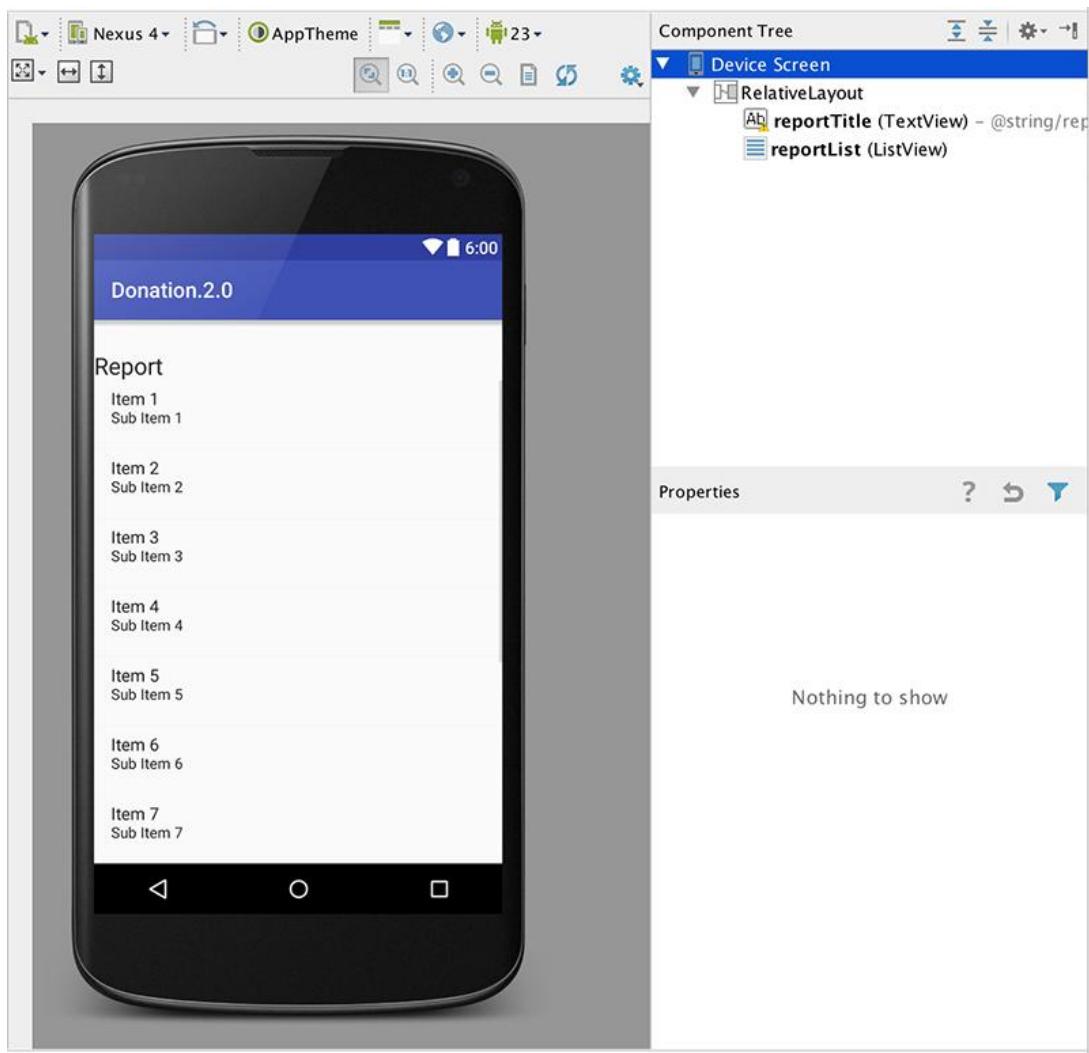
- Có thể chọn thiết kế mặc định cho layout này



- Chuyển sang **RelativeLayout** như sau



xây dựng layout giống như sau



- Nội dung tệp layout

```

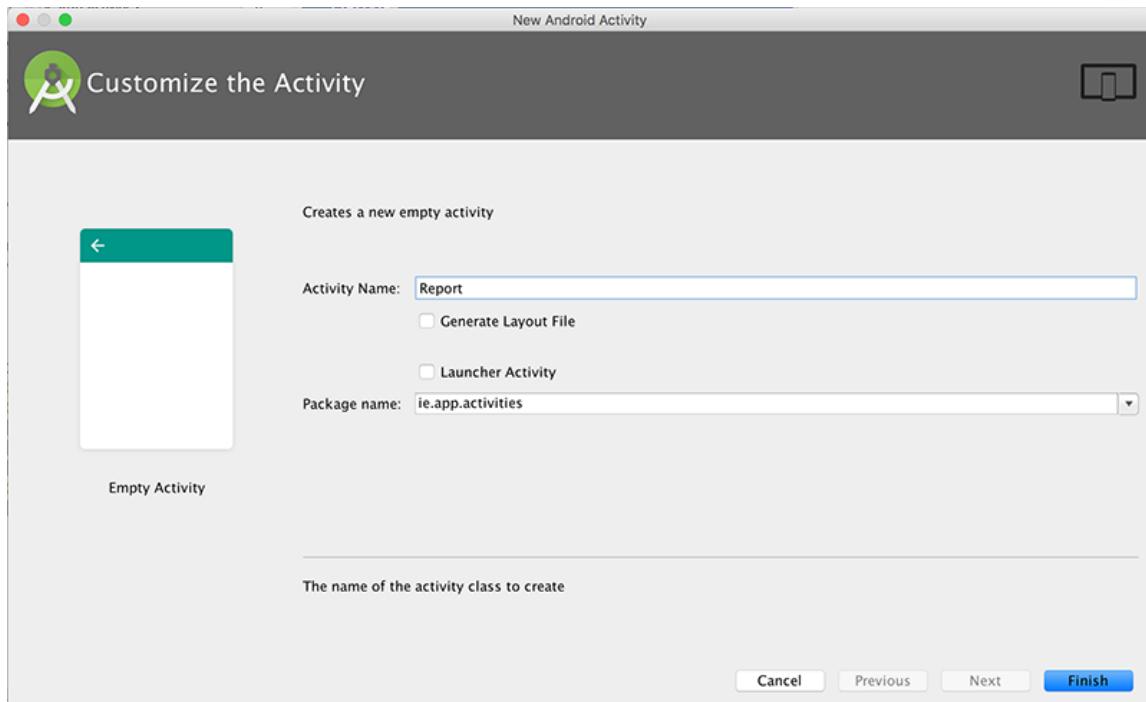
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:textAppearance="?android:attr/textAppearanceLarge"
        android:text="@string/reportTitle"
        android:id="@+id/reportTitle"
        android:layout_marginLeft="0dp"
        android:layout_marginTop="31dp"
        android:layout_alignParentTop="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true" />

    <ListView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:id="@+id/reportList"
        android:layout_below="@+id/reportTitle"
        android:layout_alignParentStart="true" />
</RelativeLayout>

```

- Tạo mới một Empty Activity trong **ie.app.activities** như sau



sau đó thay thế nội dung như sau

```

package ie.app.activities;

import android.support.v7.app.AppCompatActivity;

```

```

import android.os.Bundle;
import android.app.Activity;
import android.widget.ArrayAdapter;
import android.widget.ListView;

import ie.app.R;

public class Report extends AppCompatActivity
{
    ListView listView;

    static final String[] numbers = new String[] {
        "Amount, Pay method",
        "10, Direct",
        "100, PayPal",
        "1000, Direct",
        "10, PayPal",
        "5000, PayPal"};
}

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

    listView = (ListView) findViewById(R.id.reportList);
    ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
    android.R.layout.simple_list_item_1, numbers);
    listView.setAdapter(adapter);
}

```

- Đoạn code trên sẽ hiển thị danh sách quyên góp được viết sẵn trong code. Thay đổi **Donation** activity để tải danh sách này khi chọn **Report** trong menu

```

@Override
public boolean onOptionsItemSelected(MenuItem item)
{
    switch (item.getItemId())
    {
        case R.id.menuReport : startActivity (new Intent(this, Report.class));
        break;
    }
    return super.onOptionsItemSelected(item);
}

```

- Đảm bảo activity được khai báo trong tệp **AndroidManifest.xml**

```

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

```

```

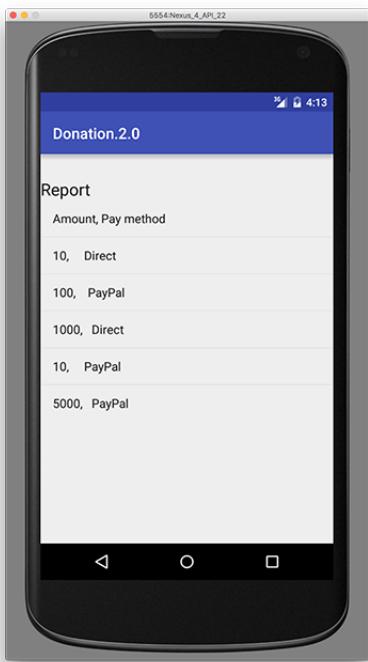
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="@string/app_name"
    android:supportsRtl="true"
    android:theme="@style/AppTheme" >
    <activity
        android:name=".activities.Donate"
        android:label="@string/app_name"
        android:theme="@style/AppTheme.NoActionBar" >
        <intent-filter>
            <action android:name="android.intent.action.MAIN" />

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

</manifest>

```

- Chạy ứng dụng, danh sách quyên góp sẽ hiển thị như bên dưới

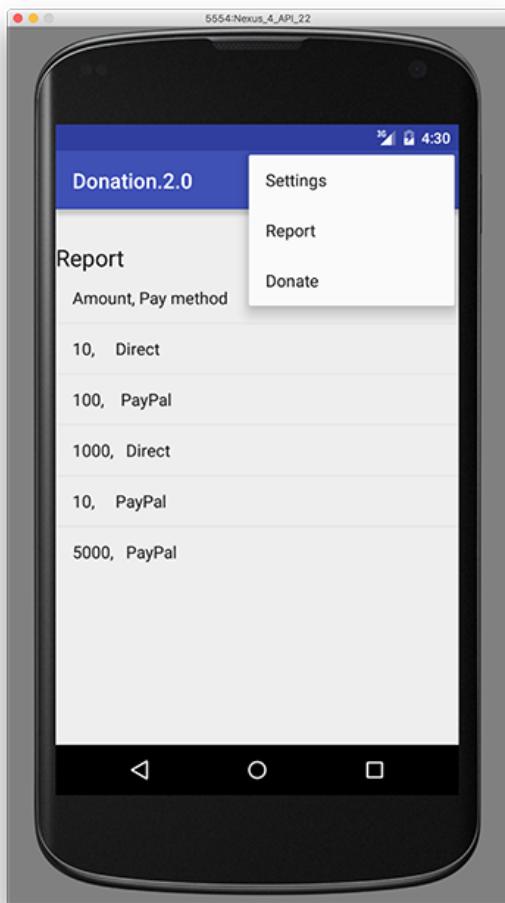


#### Bước 4. Thêm Menu trên màn hình Report

- Trong Donate activity, menu sẽ không khả dụng, cần đưa menu vào bằng cách sau

```
@Override  
public boolean onCreateOptionsMenu(Menu menu) {  
    // Inflate the menu; this adds items to the action bar if it is present.  
    getMenuInflater().inflate(R.menu.menu_donate, menu);  
    return true;  
}
```

- Tự thêm menu item mới "**Donate**" theo các bước trong bài học trước, lựa chọn này đưa người dùng trở lại màn hình **Donate**.



# BÀI THỰC HÀNH 07: Donation Object Model

**Mục tiêu:** Xây dựng ứng dụng **Donation ver3.0**

- Sử dụng Donation Object Model và Base class
- Sử dụng CustomAdapter hiển thị Report
- Thay đổi menu sử dụng event handling

## Bước 1. Sử dụng Donation Model & Base Class

- Bắt đầu với **Donation.2.0**, sử dụng Base class và Donation class để quản lý quyên góp.
- Đổi tên project thành **Donation.3.0**, tạo package **ie.app.models** trong **main**
- Tạo class **Donation** như sau

```
package ie.app.models;

public class Donation
{
    public int amount;
    public String method;

    public Donation (int amount, String method)
    {
        this.amount = amount;
        this.method = method;
    }
}
```

- Tiếp theo, tạo class **Base** trong package **ie.app.activities** như sau và sửa lỗi nếu có

```
public class Base extends AppCompatActivity
{
    public final int target = 10000;
    public int totalDonated = 0;
    public static List <Donation> donations = new ArrayList<Donation>();

    public boolean newDonation(Donation donation)
    {
        boolean targetAchieved = totalDonated > target;
        if (!targetAchieved)
        {
            donations.add(donation);
            totalDonated += donation.amount;
        }
        else
        {
            Toast toast = Toast.makeText(this, "Target Exceeded!", Toast.LENGTH_SHORT);
            toast.show();
        }
    }
}
```

```
        }

    return targetAchieved;
}

@Override
public boolean onCreateOptionsMenu(Menu menu)
{
    getMenuInflater().inflate(R.menu.donate, menu);
    return true;
}

@Override
public boolean onPrepareOptionsMenu (Menu menu){
    super.onPrepareOptionsMenu(menu);
    MenuItem report = menu.findItem(R.id.menuReport);
    MenuItem donate = menu.findItem(R.id.menuDonate);

    if(donations.isEmpty())
        report.setEnabled(false);
    else
        report.setEnabled(true);

    if(this instanceof Donate){
        donate.setVisible(false);
        if(!donations.isEmpty())
            report.setVisible(true);
    }
    else {
        report.setVisible(false);
        donate.setVisible(true);
    }

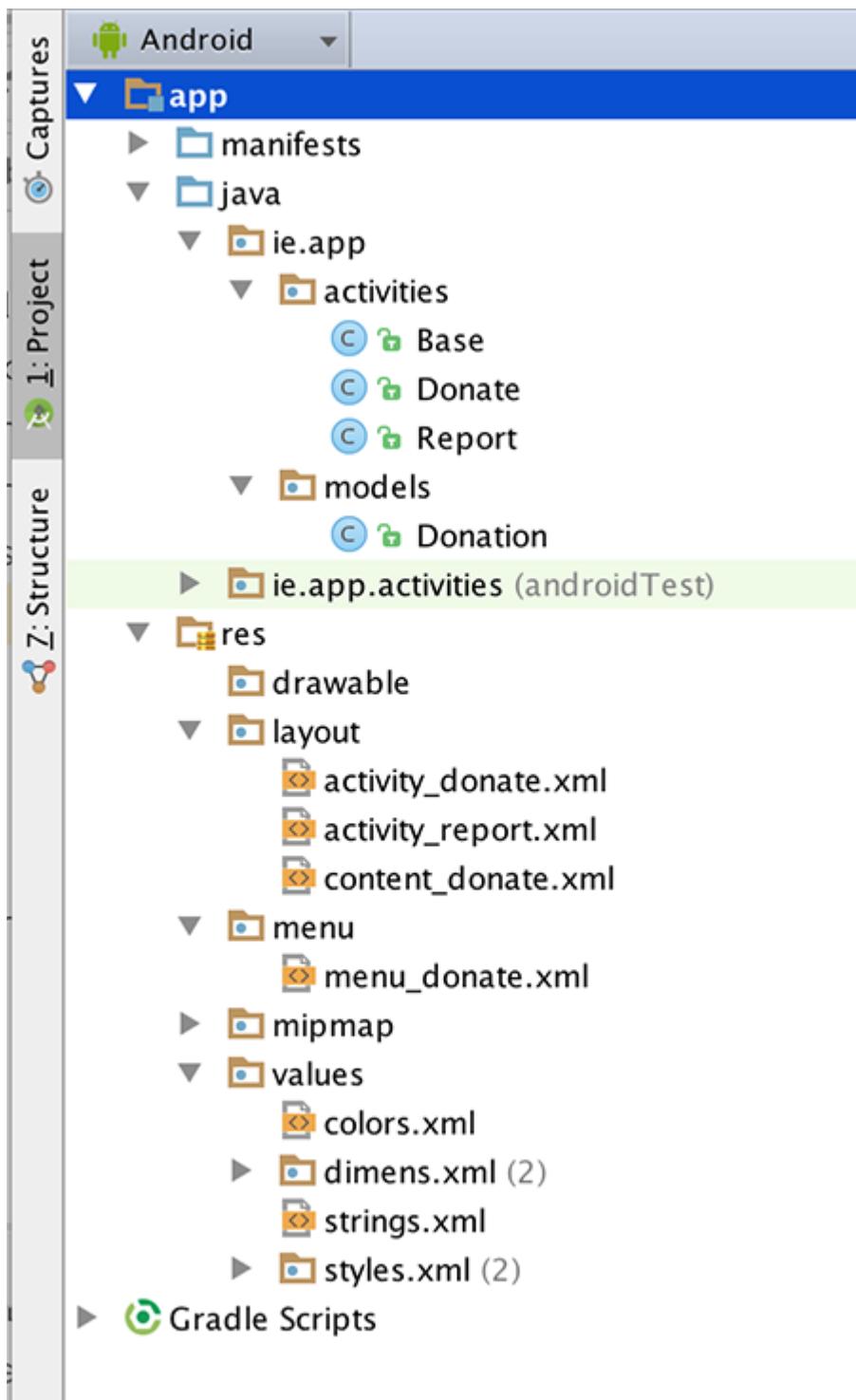
    return true;
}

public void settings(MenuItem item)
{
    Toast.makeText(this, "Settings Selected", Toast.LENGTH_SHORT).show();
}

public void report(MenuItem item)
{
    startActivity (new Intent(this, Report.class));
}

public void donate(MenuItem item)
{
    startActivity (new Intent(this, Donate.class));
}
```

- Chúng ta sẽ sử dụng List of Donations trong **Base** class để hiển thị Report. Tiếp tục sửa lỗi nếu có (gợi ý: liên quan tới thừa kế). Project sẽ trông như sau:



## Bước 2. Chỉnh sửa class Donate sử dụng class Base

- Donate** activity sử dụng **Base** class

```
public class Donate extends Base {
```

```
private Button      donateButton;
private RadioGroup paymentMethod;
private ProgressBar progressBar;
private NumberPicker amountPicker;
private EditText    amountText;
private TextView    amountTotal;

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_donate);
    Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
    setSupportActionBar(toolbar);

    FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
    fab.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Snackbar.make(view, "Replace with your own action",
                    Snackbar.LENGTH_LONG)
                    .setAction("Action", null).show();
        }
    });

    donateButton = (Button) findViewById(R.id.donateButton);

    paymentMethod = (RadioGroup) findViewById(R.id.paymentMethod);
    progressBar = (ProgressBar) findViewById(R.id.progressBar);
    amountPicker = (NumberPicker) findViewById(R.id.amountPicker);
    amountText = (EditText) findViewById(R.id.paymentAmount);
    amountTotal = (TextView) findViewById(R.id.totalSoFar);

    amountPicker.setMinValue(0);
    amountPicker.setMaxValue(1000);
    progressBar.setMax(10000);
    amountTotal.setText("$0");
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu_donate, menu);
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item)
{
    switch (item.getItemId())
    {
        case R.id.menuReport : startActivity (new Intent(this, Report.class));
        break;
    }
}
```

```

        }
        return super.onOptionsItemSelected(item);
    }

    public void donateButtonPressed (View view)
    {
        String method = paymentMethod.getCheckedRadioButtonId() == R.id.PayPal ?
"PayPal" : "Direct";
        int donatedAmount = amountPicker.getValue();
        if (donatedAmount == 0)
        {
            String text = amountText.getText().toString();
            if (!text.equals(""))
                donatedAmount = Integer.parseInt(text);
        }
        if (donatedAmount > 0)
        {
            newDonation(new Donation(donatedAmount, method));
            progressBar.setProgress(totalDonated);
            String totalDonatedStr = "$" + totalDonated;
            amountTotal.setText(totalDonatedStr);
        }
    }
}

```

- Thay thế nội dung class bằng nội dung trên và chạy thử, sửa các lỗi nếu có và xem xét thay sự khác biệt so với phiên bản trước

### Bước 3. Giao diện layout mới cho Report

- Xử lý để hiển thị danh sách quyên góp thực tế sử dụng **Base** class. Thêm hai dòng sau vào **strings.xml**

```

<string name="defaultAmount">00</string>
<string name="defaultMethod">N/A</string>

```

- Đây là layout mới "**row\_donate.xml**", đưa tệp này vào trong thư mục **layout**

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/row_amount"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginLeft="48dp"
        android:layout_marginTop="20dp"
        android:text="@string/defaultAmount" />

```

```

<TextView
    android:id="@+id/row_method"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="106dp"
    android:text="@string/defaultMethod"
    android:layout_alignTop="@+id/row_amount"
    android:layout_centerHorizontal="true" />

</RelativeLayout>

```

## BÀI THỰC HÀNH 08: Donation Object Model (tiếp)

### Bước 4. Chỉnh sửa class Report sử dụng class Base

- Sửa class **Report**, xóa bỏ các quyền góp được ghi sẵn trong code và sử dụng Adapter để hiển thị

```

public class Report extends Base
{
    ListView listView;

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

        listView = (ListView) findViewById(R.id.reportList);
        DonationAdapter adapter = new DonationAdapter(this, donations);
        listView.setAdapter(adapter);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_donate, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item)
    {
        switch (item.getItemId())
        {
            case R.id.menuDonate : startActivity (new Intent(this, Donate.class));
            break;
        }
    }
}

```

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

- Đây là **DonationAdapter**

```
class DonationAdapter extends ArrayAdapter<Donation>
{
    private Context context;
    public List<Donation> donations;

    public DonationAdapter(Context context, List<Donation> donations)
    {
        super(context, R.layout.row_donate, donations);
        this.context = context;
        this.donations = donations;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent)
    {
        LayoutInflator inflater = (LayoutInflator)
        context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);

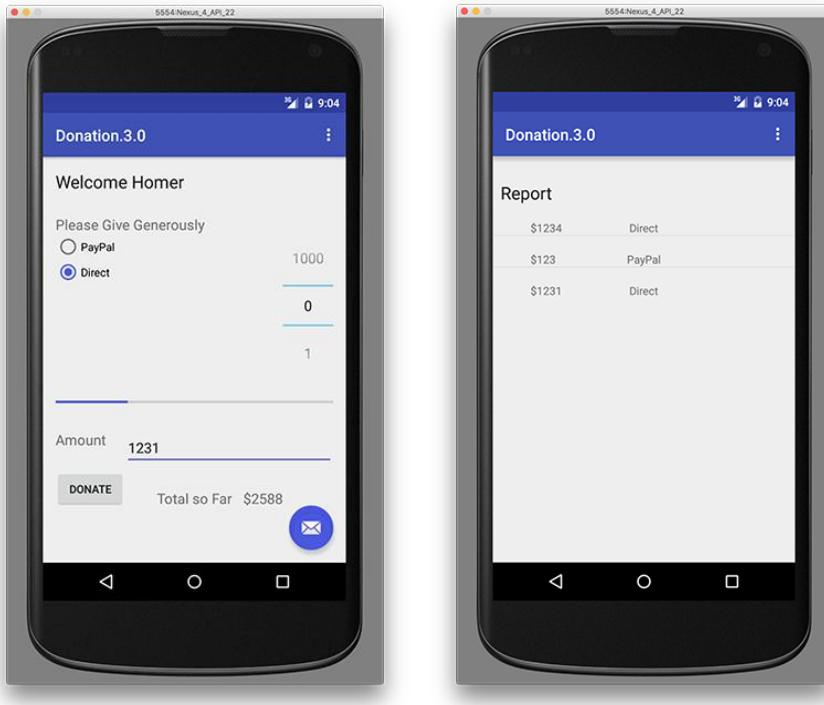
        View view = inflater.inflate(R.layout.row_donate, parent, false);
        Donation donation = donations.get(position);
        TextView amountView = (TextView) view.findViewById(R.id.row_amount);
        TextView methodView = (TextView) view.findViewById(R.id.row_method);

        amountView.setText("$" + donation.amount);
        methodView.setText(donation.method);

        return view;
    }

    @Override
    public int getCount()
    {
        return donations.size();
    }
}
```

- Sau khi hoàn thành, bạn có thể tiến hành quyên góp và nhìn thấy chúng trong danh sách Report



## Bước 5: Thêm xử lý sự kiện cho Menu

- Base class có 3 phương thức mà chúng ta chưa sử dụng

```

public void settings(MenuItem item)
{
    Toast.makeText(this, "Settings Selected", Toast.LENGTH_SHORT).show();
}

public void report(MenuItem item)
{
    startActivity (new Intent(this, Report.class));
}

public void donate(MenuItem item)
{
    startActivity (new Intent(this, Donate.class));
}

```

- Chúng ta sẽ sử dụng những phương thức này và sử dụng thuộc tính `onClick` ở trong `MenuItem`s. Chúng ta sẽ liên kết các phương thức này với các lựa chọn trong menu
- Đầu tiên, sửa đổi tệp `menu_donate.xml` và đảm bảo nó giống như sau:

```

<menu 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" tools:context=".Donate">

    <item android:id="@+id/action_settings"

```

```

        android:title="@string/action_settings"
        android:orderInCategory="100"
        app:showAsAction="never"
        android:onClick="settings"/>

    <item
        android:id="@+id/menuReport"
        android:orderInCategory="100"
        android:title="@string/menuReport"
        app:showAsAction="never"
        android:onClick="report"/>

    <item
        android:id="@+id/menuDonate"
        android:orderInCategory="100"
        android:title="@string/menuDonate"
        app:showAsAction="never"
        android:onClick="donate"/>
</menu>
```

- Mỗi thuộc tính `onCLick` cho mỗi MenuItem tương ứng với tên các phương thức trong **Base** class. Tiếp theo, chúng ta sẽ refactor class **Donate** và **Report** và bỏ menu inflation và event handling trong cả 2 class.

## Donate

```

public class Donate extends Base {

    private Button      donateButton;
    private RadioGroup  paymentMethod;
    private ProgressBar progressBar;
    private NumberPicker amountPicker;
    private EditText    amountText;
    private TextView    amountTotal;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_donate);
        Toolbar toolbar = (Toolbar) findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);

        FloatingActionButton fab = (FloatingActionButton) findViewById(R.id.fab);
        fab.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                Snackbar.make(view, "Replace with your own action",
                        Snackbar.LENGTH_LONG)
                        .setAction("Action", null).show();
            }
        });
    }
}
```

```

donateButton = (Button) findViewById(R.id.donateButton);

paymentMethod = (RadioGroup) findViewById(R.id.paymentMethod);
progressBar = (ProgressBar) findViewById(R.id.progressBar);
amountPicker = (NumberPicker) findViewById(R.id.amountPicker);
amountText = (EditText) findViewById(R.id.paymentAmount);
amountTotal = (TextView) findViewById(R.id.totalSoFar);

amountPicker.setMinValue(0);
amountPicker.setMaxValue(1000);
progressBar.setMax(10000);
amountTotal.setText("$0");
}

public void donateButtonPressed (View view)
{
    String method = paymentMethod.getCheckedRadioButtonId() == R.id.PayPal ?
"PayPal" : "Direct";
    int donatedAmount = amountPicker.getValue();
    if (donatedAmount == 0)
    {
        String text = amountText.getText().toString();
        if (!text.equals(""))
            donatedAmount = Integer.parseInt(text);
    }
    if (donatedAmount > 0)
    {
        newDonation(new Donation(donatedAmount, method));
        progressBar.setProgress(totalDonated);
        String totalDonatedStr = "$" + totalDonated;
        amountTotal.setText(totalDonatedStr);
    }
}
}
}

```

## Report

```

public class Report extends Base
{
    ListView listView;

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

        listView = (ListView) findViewById(R.id.reportList);
        DonationAdapter adapter = new DonationAdapter(this, donations);
        listView.setAdapter(adapter);
    }
}

```

}

- Chạy ứng dụng và kiểm tra các thay đổi

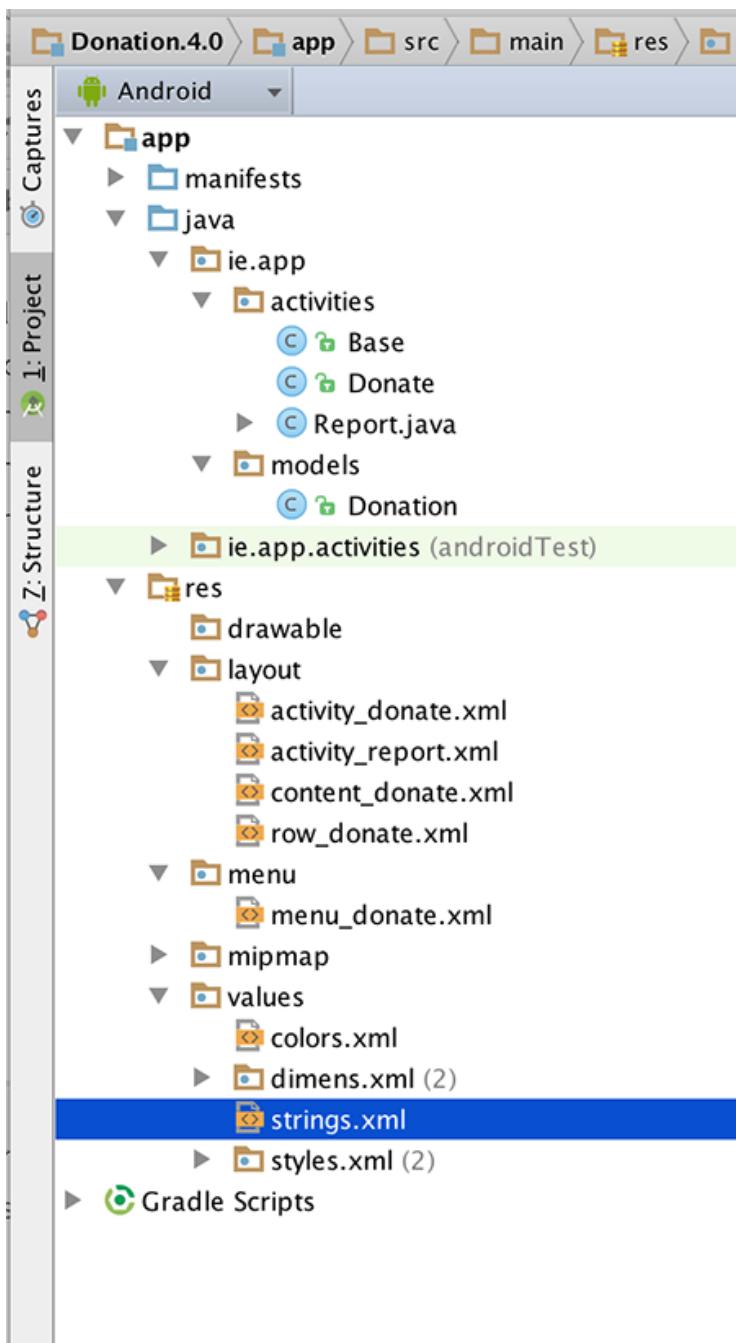
# BÀI THỰC HÀNH 09: Donation 4.0 - Database/Application Support

Mục tiêu: Xây dựng ứng dụng Donation ver4.0

- Thêm cơ sở dữ liệu cho ứng dụng
- Làm việc với một đối tượng "Application"

## Bước 1. Thiết lập sử dụng starter code

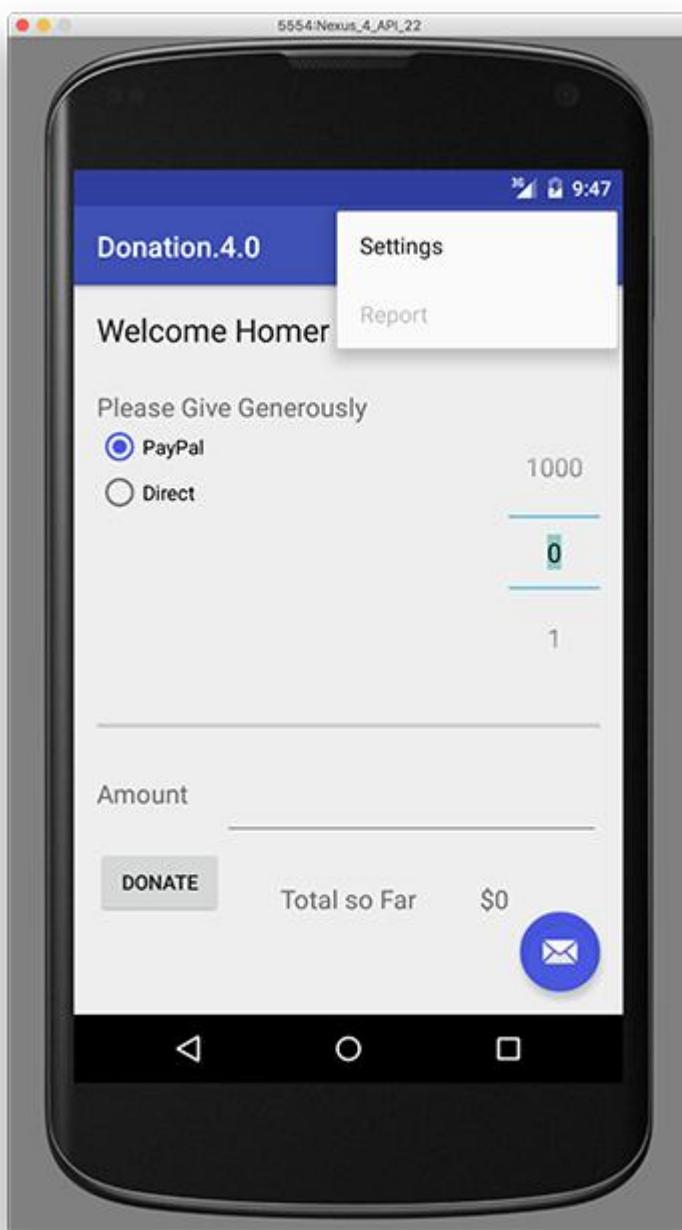
- Tiếp tục với project của mình hoặc tải starter code ở đây
  - [Donation.4.0.starter](#)



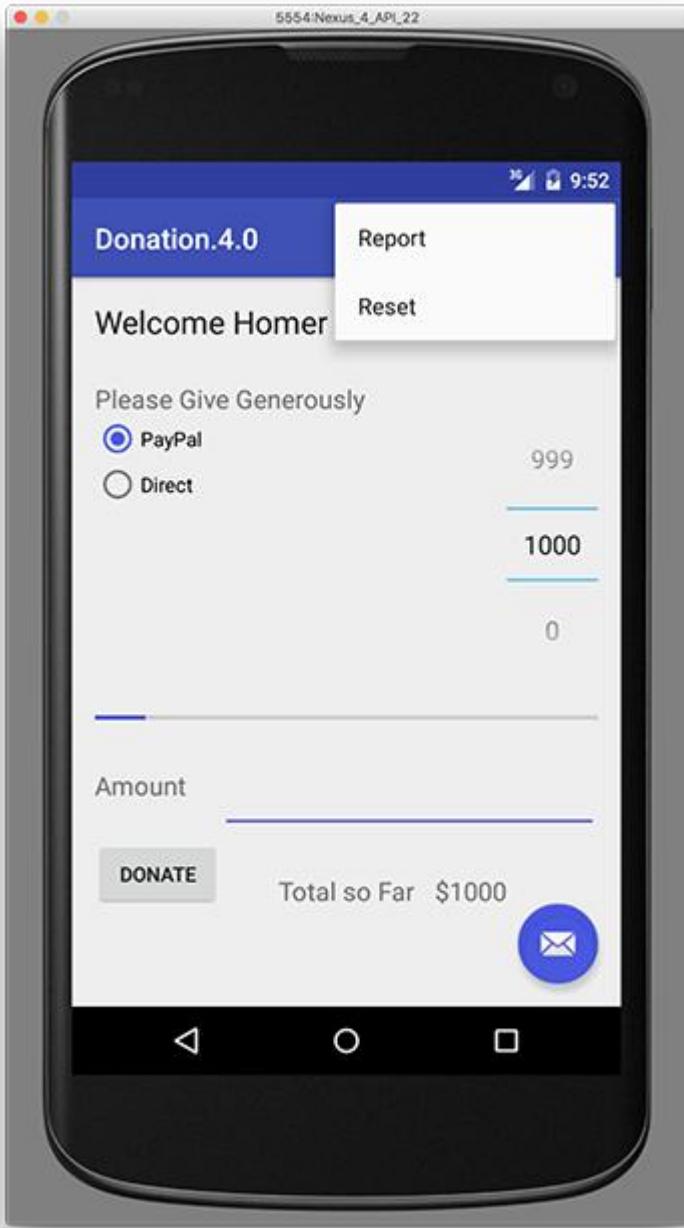
- Trong lab này, bạn sẽ làm các việc sau:
  - Thêm một lựa chọn mới "Reset" trong Menu để xóa hết các quyên góp khi đã đạt đủ
  - Thêm cơ sở dữ liệu để quản lý quyên góp
  - Refactor các class cũ để tích hợp thêm cơ sở dữ liệu

## Bước 2. Thêm lựa chọn mới trong Menu

- Bắt đầu với Menu như sau



và chúng ta muốn một Menu như sau



- Điều đầu tiên cần làm là thêm dòng sau vào **strings.xml**

```
<string name="menuReset">Reset</string>
```

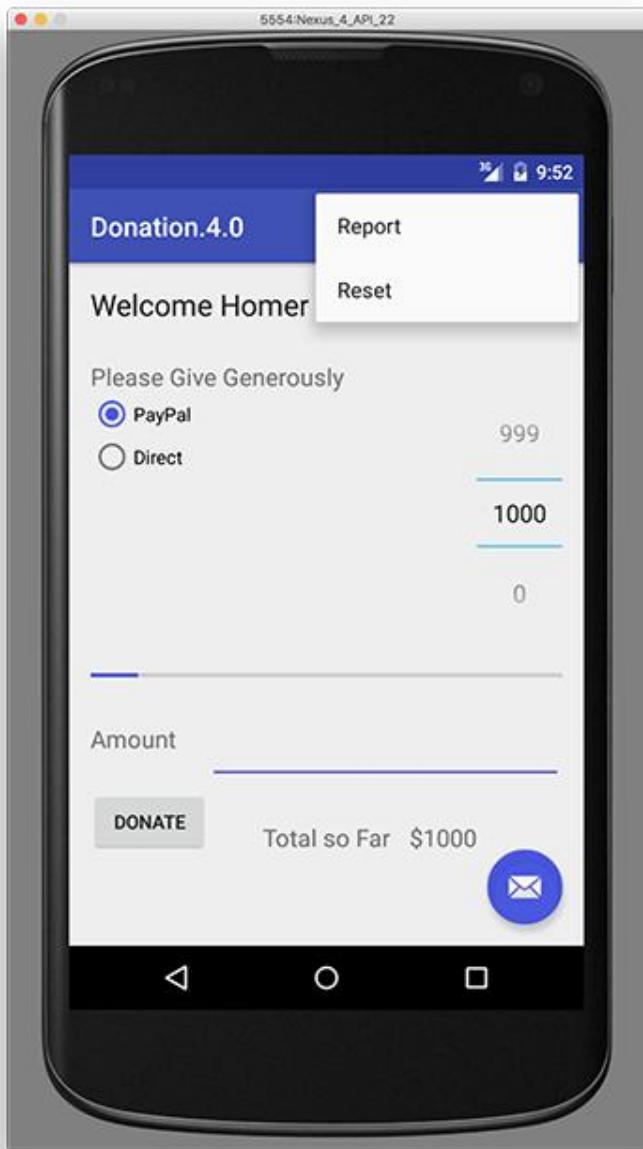
sau đó thêm vào tệp **donate.xml** đoạn sau

```
<item  
    android:id="@+id/menuReset"  
    android:orderInCategory="100"  
    android:showAsAction="never"  
    android:title="@string/menuReset"  
    android:onClick="reset"/>
```

- Có thể bỏ đi lựa chọn "Settings" và các phương thức liên quan. Tiếp theo, sửa tệp **Base.java** và thêm vào dòng sau

```
public void reset(MenuItem item) {}
```

để đảm bảo ứng dụng không bị crash khi tải menu, chạy lại ứng dụng và quan sát menu như dưới đây



- Chúng ta chưa thể cài đặt hoàn toàn lựa chọn này trong menu, vì vậy chúng ta chỉ tạm thời reset amount về 0 (zero)

## BÀI THỰC HÀNH 10: Donation 4.0 - Database/Application Support (tiếp)

### Bước 3. Resetting the Target Amount

- Sửa tệp **Donate.java** và thêm vào phương thức "reset"

```
@Override  
public void reset(MenuItem item)  
{  
    // Your implementation goes here  
}
```

- Cài đặt nội dung phương thức để khi lựa chọn **Reset**, *totalDonated* trở về 0. Đồng thời cập nhật giao diện màn hình Donate sao cho phù hợp.
- Chạy ứng dụng và kiểm tra chức năng **Reset**

### Bước 4. Sử dụng Application Object

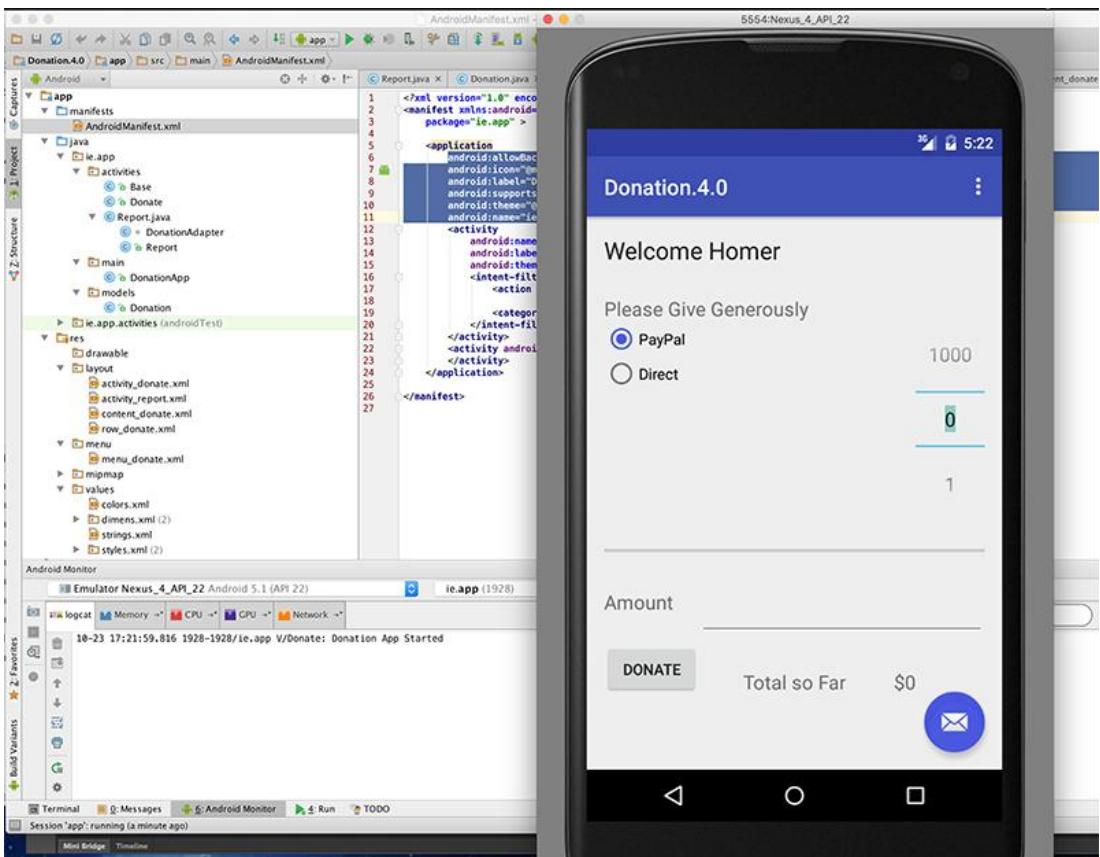
- Để thiết kế ứng dụng một cách mạch lạc, chúng ta sử dụng thêm **Application object**. Tạo một package mới "**ie.app.main**" và thêm vào class sau

```
package ie.app.main;  
  
import android.app.Application;  
import android.util.Log;  
  
public class DonationApp extends Application  
{  
    @Override  
    public void onCreate()  
    {  
        super.onCreate();  
        Log.v("Donate", "Donation App Started");  
    }  
}
```

- Application objects cần được tham chiếu trong **AndroidManifest.xml**

```
<application  
    android:allowBackup="true"  
    android:icon="@mipmap/ic_launcher"  
    android:label="@string/app_name"  
    android:supportsRtl="true"  
    android:theme="@style/AppTheme"  
    android:name="ie.app.main.DonationApp">
```

- Đảm bảo xuất hiện dòng chữ "*Donation App Started*" trong log để chắc chắn rằng ứng dụng được khởi động một cách chính xác



## Bước 5. Sử dụng Donation Model

- Chúng ta sẽ refactor **Base** class và di chuyển các thuộc tính và phương thức liên quan tới quyên góp vào trong **DonationApp** class
- Đây là nội dung của **DonationApp**, có chức năng quản lý danh sách quyên góp.

```

package ie.app.main;

import java.util.ArrayList;
import java.util.List;

import android.app.Application;
import android.util.Log;
import android.widget.Toast;
import ie.app.models.Donation;

public class DonationApp extends Application
{
    public final int      target      = 10000;
    public int            totalDonated = 0;
    public List <Donation> donations   = new ArrayList<Donation>();

    public boolean newDonation(Donation donation)
    {
        boolean targetAchieved = totalDonated > target;
        if (!targetAchieved)
    }
}

```

```

    {
        donations.add(donation);
        totalDonated += donation.amount;
    }
    else
    {
        Toast.makeText(this, "Target Exceeded!", Toast.LENGTH_SHORT).show();
    }
    return targetAchieved;
}

@Override
public void onCreate()
{
    super.onCreate();
    Log.v("Donation", "Donation App Started");
}
}

```

## Bước 6. Chỉnh sửa class Base

- **Base** activity được refactor để tận dụng **DonationApp** object, một số lỗi xuất hiện sẽ được sửa trong các bước sau

```

package ie.app.activities;

import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.view.Menu;
import android.view.MenuItem;
import android.os.Bundle;

import ie.app.R;
import ie.app.main.DonationApp;

public class Base extends AppCompatActivity {

    public DonationApp app;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        app = (DonationApp) getApplication();

        app.dbManager.open();
        app.dbManager.setTotalDonated(this);
    }

    @Override
    protected void onDestroy() {
        super.onDestroy();
        app.dbManager.close();
    }
}

```

```
}

@Override
public boolean onCreateOptionsMenu(Menu menu)
{
    getMenuInflater().inflate(R.menu.menu_donate, menu);
    return true;
}

@Override
public boolean onPrepareOptionsMenu (Menu menu){
    super.onPrepareOptionsMenu(menu);
    MenuItem report = menu.findItem(R.id.menuReport);
    MenuItem donate = menu.findItem(R.id.menuDonate);
    MenuItem reset = menu.findItem(R.id.menuReset);

    if(app.dbManager.getAll().isEmpty())
    {
        report.setEnabled(false);
        reset.setEnabled(false);
    }
    else {
        report.setEnabled(true);
        reset.setEnabled(true);
    }
    if(this instanceof Donate){
        donate.setVisible(false);
        if(!app.dbManager.getAll().isEmpty())
        {
            report.setVisible(true);
            reset.setEnabled(true);
        }
    }
    else {
        report.setVisible(false);
        donate.setVisible(true);
        reset.setVisible(false);
    }

    return true;
}

public void report(MenuItem item)
{
    startActivity (new Intent(this, Report.class));
}

public void donate(MenuItem item)
{
    startActivity (new Intent(this, Donate.class));
}
```

```
    public void reset(MenuItem item) {}
}
```

## Bước 7. Thêm hỗ trợ cơ sở dữ liệu

- Sau khi import đầy đủ các class hỗ trợ cơ sở dữ liệu, thay thế lời gọi hàm xử lý **donationList** với lời gọi tới **dbManager**
  - database classes:** Thêm các class có trong tệp này vào package mới **ie.app.database**
- Khởi tạo một đối tượng **DBManager** trong **Base.java** và đóng/mở cơ sở dữ liệu khi cần
- Cập nhật class **DonationApp** như sau

```
public class DonationApp extends Application
{
    public final int      target      = 10000;
    public int            totalDonated = 0;
    //public List <Donation> donations = new ArrayList<Donation>();
    public DBManager dbManager;

    public boolean newDonation(Donation donation)
    {
        boolean targetAchieved = totalDonated > target;
        if (!targetAchieved)
        {
            dbManager.add(donation);
            totalDonated += donation.amount;
        }
        else
        {
            Toast.makeText(this, "Target Exceeded!", Toast.LENGTH_SHORT).show();
        }
        return targetAchieved;
    }

    @Override
    public void onCreate()
    {
        super.onCreate();
        Log.v("Donate", "Donation App Started");
        dbManager = new DBManager(this);
        Log.v("Donate", "Database Created");

    }
}
```

- Thay thế class **Donation** hiện tại với đoạn code sau

```
public class Donation
{
    public int id;
```

```

public int amount;
public String method;

public Donation (int amount, String method)
{
    this.amount = amount;
    this.method = method;
}

public Donation ()
{
    this.amount = 0;
    this.method = "";
}

public String toString()
{
    return id + ", " + amount + ", " + method;
}
}

```

- Sau khi thay đổi xong, comment **donationList** và lưu lại, bạn sẽ nhìn thấy lỗi xuất hiện giúp bạn chỉ ra class nào cần được cập nhật với các lời gọi tới cơ sở dữ liệu.
- Sau khi chạy lại ứng dụng, bạn có thể quyên góp và những quyên góp này sẽ được lưu trong cơ sở dữ liệu.
- Kiểm tra lần cuối bằng cách tắt và khởi động lại ứng dụng và nhìn vào Report để kiểm tra cơ sở dữ liệu

## BÀI THỰC HÀNH 11: Donation 4.0 - Database/Application Support (tiếp)

### Bước 8. Resetting the Donations

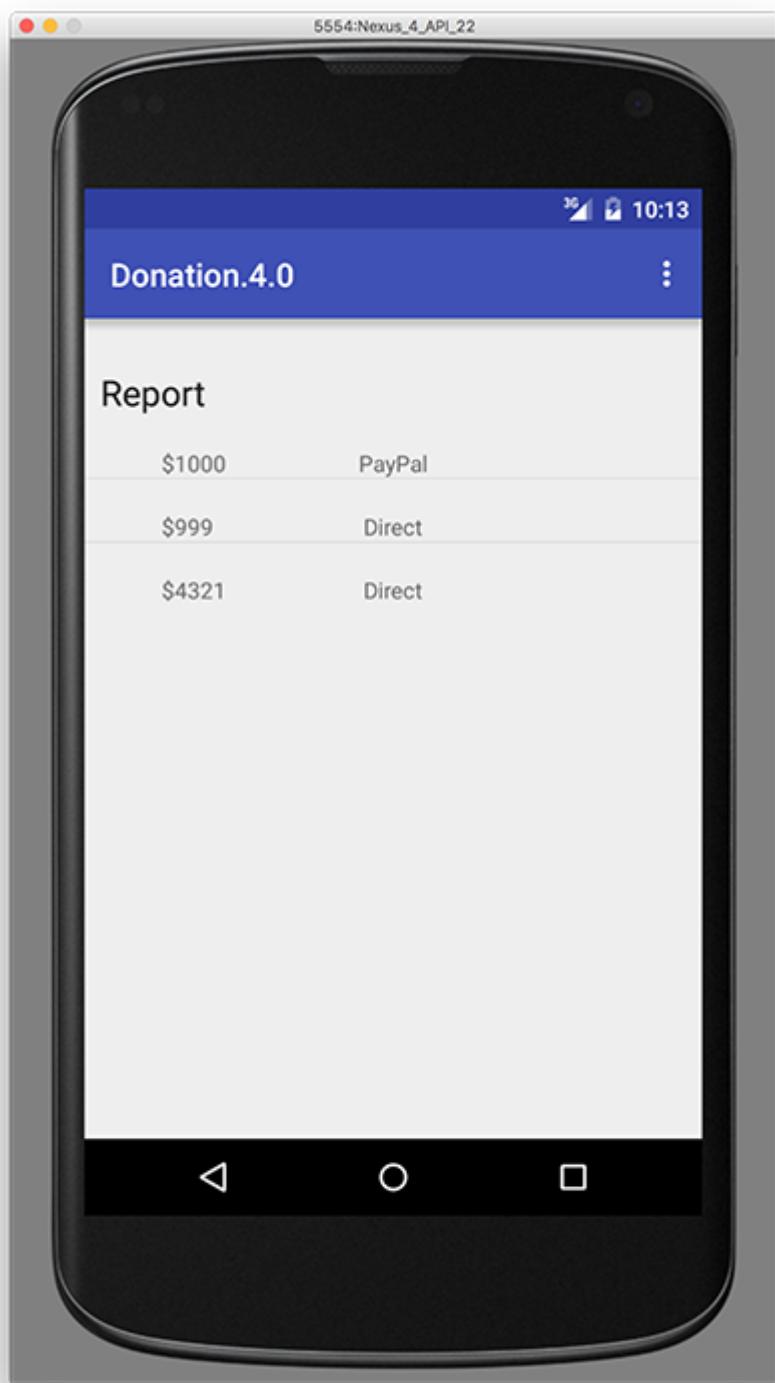
- Bước cuối cùng liên quan tới việc xóa toàn bộ cơ sở dữ liệu quyên góp khi người dùng nhấn "**Reset**"
- Thực hiện gọi phương thức **reset()** trên đối tượng cơ sở dữ liệu khi người dùng lựa chọn "**Reset**" trên menu

```

@Override
public void reset(MenuItem item)
{
    app.dbManager.reset();
    app.totalDonated = 0;
    amountTotal.setText("$" + app.totalDonated);
}

```

- Cần cập nhật phương thức `onPrepareOptionsMenu()` trong **Base** class để xử lý vô hiệu hóa hay hiển thị "**Reset**" trên menu
- Có một lỗi nhỏ trong ứng dụng liên quan tới việc khởi động lại ứng dụng, bạn hãy xác định và sửa nó.



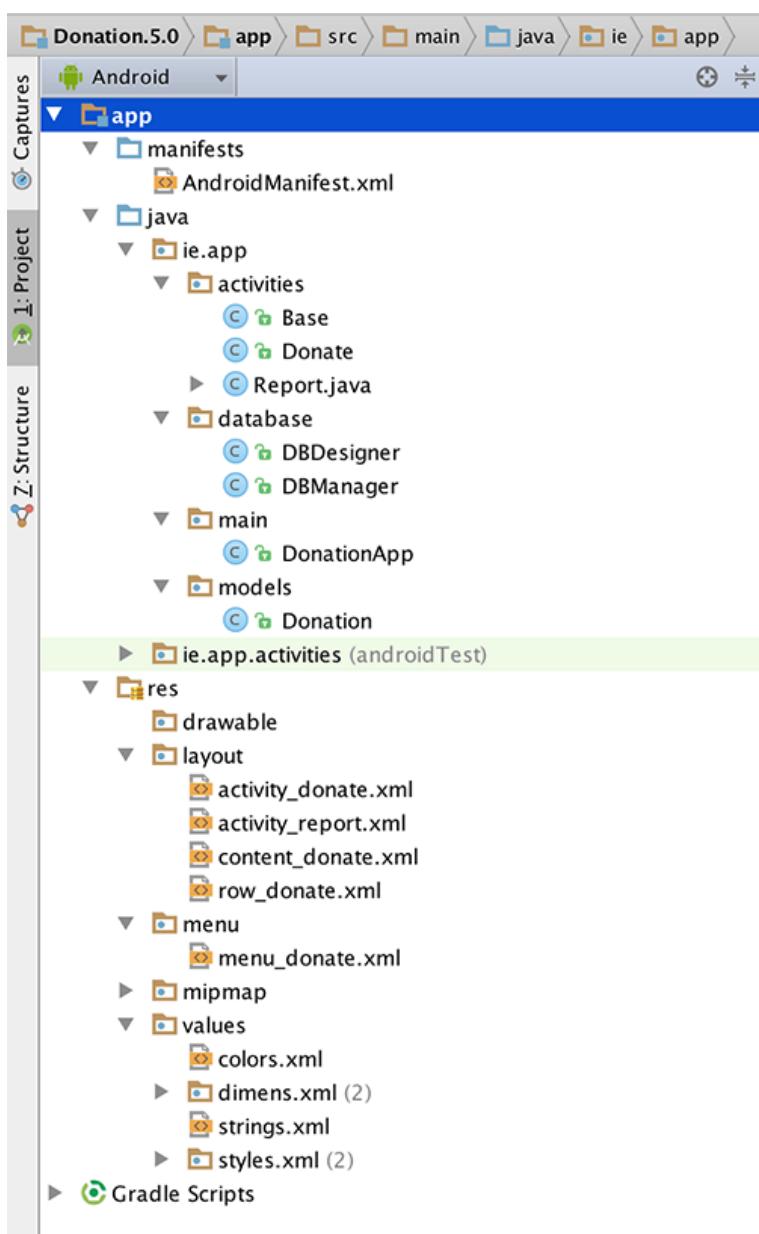
## BÀI THỰC HÀNH 12: Hỗ trợ REST/Cloud cho ứng dụng

**Mục tiêu:** Xây dựng ứng dụng **Donation ver5.0**

- Kết nối tới REST Web Service
- Kết nối tới một remote server để nhận và xóa nhiều quyên góp
- Kết nối tới một remote server để nhận một quyên góp
- Thêm xử lý sự kiện đối với ảnh

### Bước 1. Thiết lập với starter code

- Tiếp tục với project cũ hoặc tải starter code ở đây
  - [Donation.5.0.starter](#)
- Project hiện tại như sau



- Trong bài này, bạn sẽ thực hiện các việc sau:

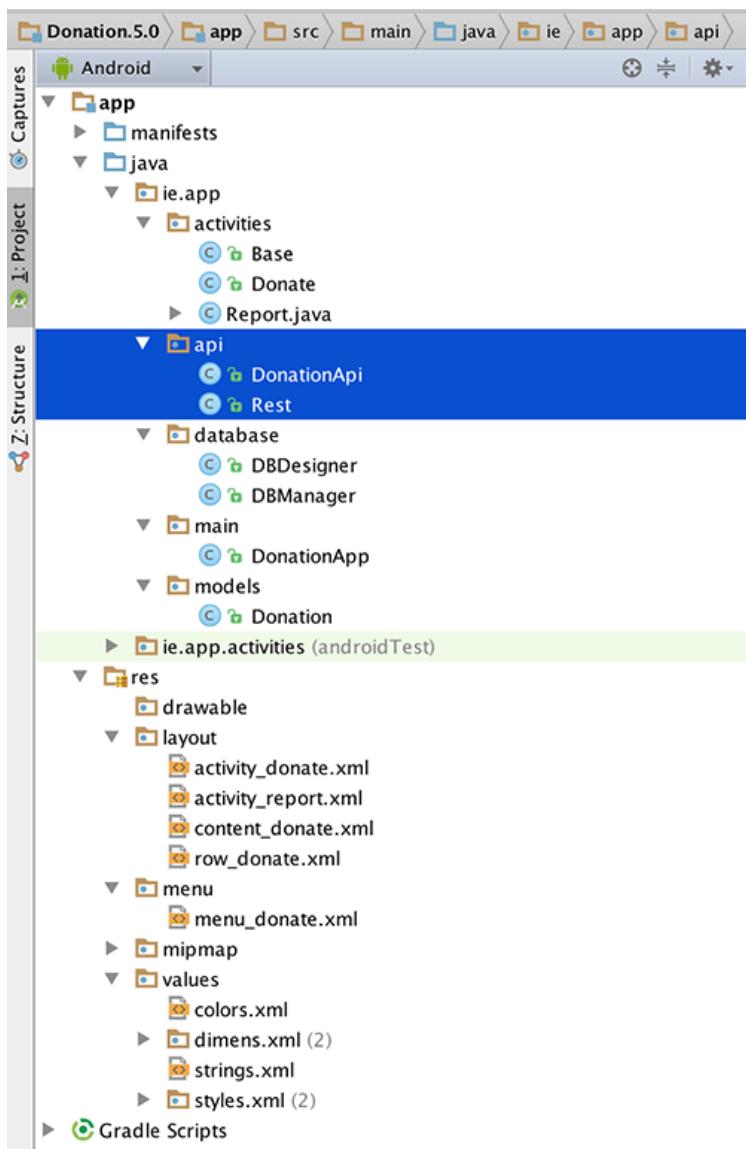
- Hỗ trợ REST thông qua [Donation sister site](#)
- Bổ hỗ trợ cơ sở dữ liệu và chuyển về danh sách Donation ban đầu
- Refactor các class hiện tại để tích hợp REST API

## Bước 2. Thêm REST APIs

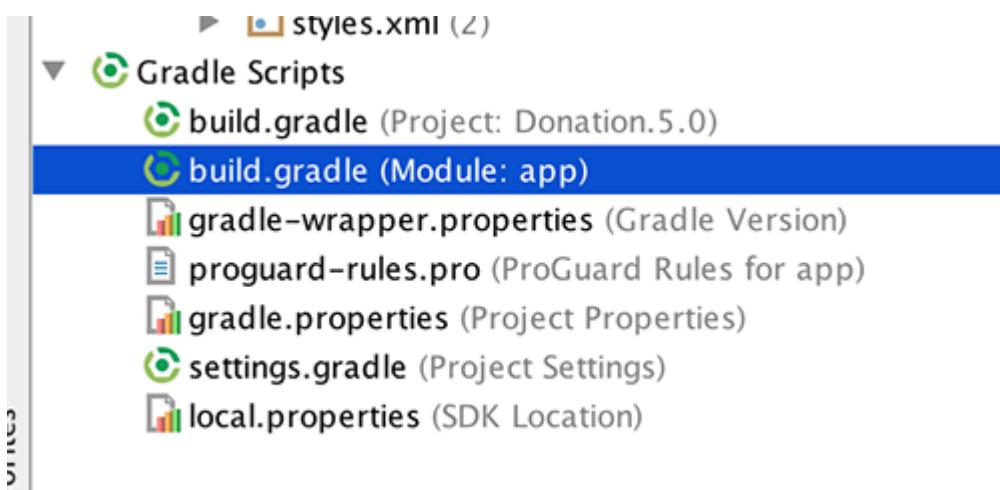
- Tải package chứa API đã được viết sẵn ([tại đây](#)) bao gồm
  - DonationApi.java
  - Rest.java



- Copy thư mục **api** vào package **ie.app**



- Rebuild project, một số lỗi liên quan tới [Googles Gson](#) xuất hiện yêu cầu bạn thêm **Googles Gson's** dependency vào project
- Mở tệp **build.gradle** của ứng dụng (không phải của project)



- Thêm Gson dependency

```
compile 'com.google.code.gson:gson:2.2.3'
```



- Rebuild project và các lỗi đã được sửa
- Xóa bỏ toàn bộ package **ie.app.database**, sử dụng lại danh sách **Donation**

```
public List <Donation> donations = new ArrayList<Donation>();
```

- Cuối cùng, xóa bỏ toàn bộ tham chiếu tới **dbManager** trong project và thay thế bằng **donations** hoặc **app.donations** (tùy ngữ cảnh)

### Bước 3. Chỉnh sửa Donate Activity cập nhật tổng số tiền

- Khi ứng dụng khởi động, tổng số tiền hiện tại cần chính xác theo web app. Chúng ta cần lấy danh sách toàn bộ lượt quyên góp từ Server để gán giá trị tổng sử dụng các class REST và AsyncTasks để thực hiện ngầm
- Trước khi bắt đầu, chúng ta cần cho phép ứng dụng truy cập Internet

**AndroidManifest.xml** file (just before the `<application>` tag)

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
```

- Tạo class mới

```
private class GetAllTask extends AsyncTask<String, Void, List<Donation>> {

    protected ProgressDialog dialog;
    protected Context context;

    public GetAllTask(Context context) {
        this.context = context;
    }

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        this.dialog = new ProgressDialog(context, 1);
        this.dialog.setMessage("Retrieving Donations List");
        this.dialog.show();
    }

    @Override
    protected List<Donation> doInBackground(String... params) {
        try {
            Log.v("donate", "Donation App Getting All Donations");
            return (List<Donation>) DonationApi.getAll((String) params[0]);
        }
        catch (Exception e) {
            Log.v("donate", "ERROR : " + e);
            e.printStackTrace();
        }
        return null;
    }

    @Override
    protected void onPostExecute(List<Donation> result) {
        super.onPostExecute(result);

        //use result to calculate the totalDonated amount here

        progressBar.setProgress(app.totalDonated);
        amountTotal.setText("$" + app.totalDonated);

        if (dialog.isShowing())
            dialog.dismiss();
    }
}
```

- Để khởi động task, sử dụng phương thức sau

```

@Override
public void onResume() {
    super.onResume();
    new GetAllTask(this).execute("/donations");
}

```

- Nếu bạn chạy ứng dụng và kết nối tới web service rồi trả lại danh sách từ website, ứng dụng sẽ crash
- Từ log, bạn hãy tìm ra nguyên nhân và xử lý

#### Bước 4. Chỉnh sửa Donation Model

- JSON được trả về ở bài trước có 4 trường, với các kiểu khác nhau và không phù hợp với **Donation** class (lý do ứng dụng bị crash)
- Thay thế Donation model hiện tại với code này

```

public class Donation
{
    public String _id;
    public int amount;
    public String paymenttype;
    public int upvotes;

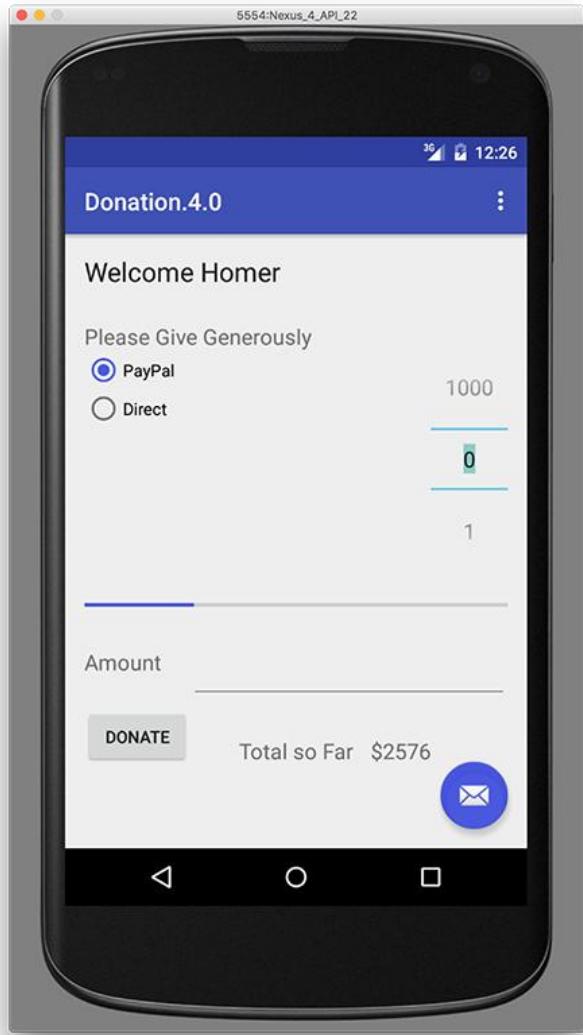
    public Donation (int amount, String method, int upvotes)
    {
        this.amount = amount;
        this.paymenttype = method;
        this.upvotes = upvotes;
    }

    public Donation ()
    {
        this.amount = 0;
        this.paymenttype = "";
        this.upvotes = 0;
    }

    public String toString()
    {
        return _id + ", " + amount + ", " + paymenttype + ", " + upvotes;
    }
}

```

- Có một vài lỗi nhỏ cần sửa, sau khi sửa xong và chạy ứng dụng sẽ được như sau



## BÀI THỰC HÀNH 13: Hỗ trợ REST/Cloud cho ứng dụng (tiếp)

### Bước 5. Thay đổi layout của Report

- Bắt đầu hiển thị tổng lên Report
  - Sửa đổi layout hiện tại
  - Sử dụng Androids [SwipeRefreshLayout](#)

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical" android:layout_width="match_parent"
    android:layout_height="match_parent">
```

```
<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
```

```
    android:text="@string/reportTitle"
    android:id="@+id/reportTitle"
    android:layout_marginLeft="0dp"
    android:layout_marginTop="31dp"
    android:layout_alignParentTop="true"
    android:layout_alignParentStart="true"
    android:layout_alignParentEnd="true"
    android:paddingLeft="10dp" />

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal" android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_below="@+id/reportTitle"
    android:layout_alignParentStart="true"
    android:id="@+id/linearLayout">

    <TextView
        android:id="@+id/row_upvotes"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginStart="15dp"
        android:layout_marginTop="20dp"
        android:text="Upvotes"
        android:layout_weight="0.5"
        android:textSize="20sp"
        android:textColor="@color/colorPrimaryDark" />

    <TextView
        android:id="@+id/row_amount"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:text="Amount"
        android:layout_weight="0.5"
        android:textSize="20sp"
        android:textColor="@color/colorPrimaryDark" />

    <TextView
        android:id="@+id/row_method"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:text="Method"
        android:layout_weight="1.5"
        android:textSize="20sp"
        android:textColor="@color/colorPrimaryDark" />

</LinearLayout>

<android.support.v4.widget.SwipeRefreshLayout
    android:id="@+id/report_swipe_refresh_layout"
    android:layout_width="match_parent"
```

```

        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_below="@+id/linearLayout">

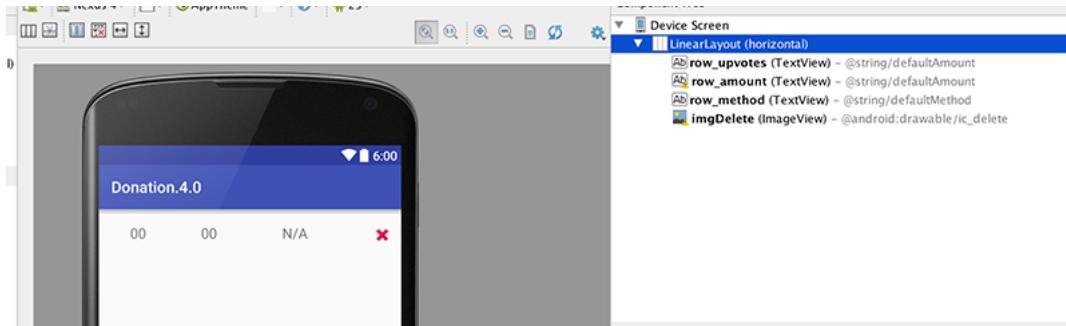
    <ListView
        android:id="@+id/reportList"
        android:layout_width="match_parent"
        android:layout_height="match_parent">
    </ListView>

</android.support.v4.widget.SwipeRefreshLayout>

</RelativeLayout>

```

- Cần sửa đổi cách hiển thị hàng để người dùng có thể nhìn thấy số "upvotes" của mỗi quyên góp
- Refactor **row\_donate.xml** với TextView mới hiển thị "upvotes"
- Thêm chức năng xóa một quyên góp thông qua nút "delete"



## Bước 6. Hiển thị danh sách quyên góp lên Report

- Đây là layout sau khi cập nhật

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="horizontal" android:layout_width="fill_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/row_upvotes"
        android:layout_width="80dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="40dp"
        android:layout_marginTop="20dp"
        android:text="@string/defaultAmount"
        android:textSize="18sp" />

```

```

<TextView
    android:id="@+id/row_amount"
    android:layout_width="80dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"
    android:layout_marginLeft="10dp"
    android:text="@string/defaultAmount"
    android:layout_weight="1"
    android:textSize="18sp" />

<TextView
    android:id="@+id/row_method"
    android:layout_width="80dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="20dp"

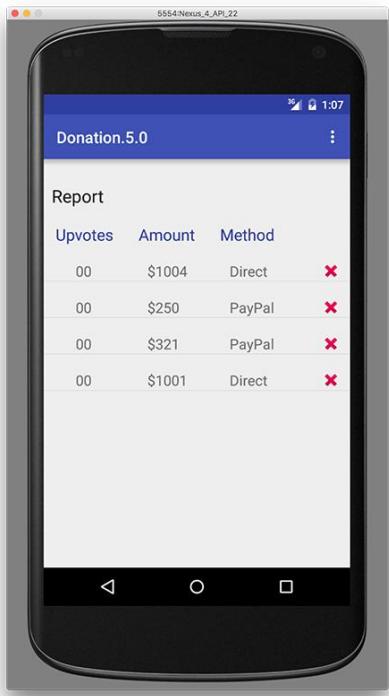
    android:text="@string/defaultMethod"
    android:layout_weight="1"
    android:textSize="18sp" />

<ImageView
    android:id="@+id/imgDelete"
    android:layout_width="25dp"
    android:layout_height="25dp"
    android:layout_marginTop="20dp"
    android:layout_weight="1"
    android:src="@android:drawable/ic_delete" />

</LinearLayout>

```

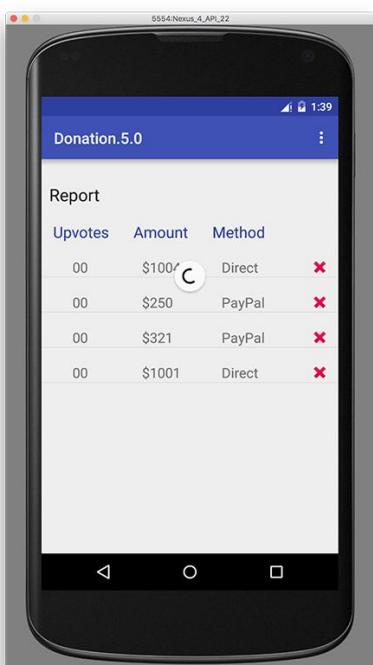
- Nếu bạn chạy ứng dụng và chọn “**Report**”, bạn sẽ nhìn thấy danh sách được lấy về từ server mà không cần viết thêm code



- Thử chức năng vuốt để làm mới danh sách

### Bước 7. Report Activity - Implementing Swipe Refresh

- Mặc dù chúng ta có thể làm mới danh sách trong report nhưng chúng ta chỉ thấy vòng tròn tiến độ xoay vô hạn mà không có cách nào dừng nó



- Lý do là chúng ta chưa cài đặt backend cho tính năng này
- Sử dụng AsyncTask để lấy dữ liệu về các lần quyên góp

```

private class GetAllTask extends AsyncTask<String, Void, List<Donation>> {

    protected ProgressDialog dialog;
    protected Context context;

    public GetAllTask(Context context) {
        this.context = context;
    }

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        this.dialog = new ProgressDialog(context, 1);
        this.dialog.setMessage("Retrieving Donations List");
        this.dialog.show();
    }

    @Override
    protected List<Donation> doInBackground(String... params) {

        try {
            return (List<Donation>) DonationApi.getAll((String) params[0]);
        } catch (Exception e) {
            Log.v("ASYNC", "ERROR : " + e);
            e.printStackTrace();
        }
        return null;
    }

    @Override
    protected void onPostExecute(List<Donation> result) {
        super.onPostExecute(result);

        app.donations = result;
        adapter = new DonationAdapter(context, app.donations);
        listView.setAdapter(adapter);
        listView.setOnItemClickListener(Report.this);
        mSwipeRefreshLayout.setRefreshing(false);
        if (dialog.isShowing())
            dialog.dismiss();
    }
}

```

- Sửa các lỗi xuất hiện và khai báo thêm như sau

SwipeRefreshLayout mSwipeRefreshLayout;

- Thay thế nội dung phương thức `onCreate()` bằng đoạn code dưới đây

```

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

    listView = (ListView) findViewById(R.id.reportList);
    mSwipeRefreshLayout = (SwipeRefreshLayout)
        findViewById(R.id.report_swipe_refresh_layout);

    new GetAllTask(this).execute("/donations");

    mSwipeRefreshLayout.setOnRefreshListener(new
        SwipeRefreshLayout.OnRefreshListener() {
            @Override
            public void onRefresh() {
                new GetAllTask(Report.this).execute("/donations");
            }
        });
}

```

- Chạy ứng dụng và kiểm tra chức năng làm mới danh sách

## Bước 8. Thêm event handling

- Thay thế nội dung class **DonationAdapter** bằng đoạn code sau, chạy và kiểm tra ứng dụng

```

class DonationAdapter extends ArrayAdapter<Donation> {
    private Context context;
    public List<Donation> donations;

    public DonationAdapter(Context context, List<Donation> donations) {
        super(context, R.layout.row_donate, donations);
        this.context = context;
        this.donations = donations;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        LayoutInflator inflater = (LayoutInflator) context
            .getSystemService(Context.LAYOUT_INFLATER_SERVICE);

        View view = inflater.inflate(R.layout.row_donate, parent, false);
        Donation donation = donations.get(position);

        TextView amountView = (TextView) view.findViewById(R.id.row_amount);
        TextView methodView = (TextView) view.findViewById(R.id.row_method);
        TextView upvotesView = (TextView) view.findViewById(R.id.row_upvotes);
    }
}

```

```

        amountView.setText("+" + donation.amount);
        methodView.setText(donation.paymenttype);
        upvotesView.setText("+" + donation.upvotes);

        view.setTag(donation._id); // setting the 'row' id to the id of the donation

        return view;
    }

    @Override
    public int getCount() {
        return donations.size();
    }
}

```

- Trong khi làm việc trên Report activity, chúng ta thêm vào một số event handling
  - hiển thị chi tiết từng lần quyên góp
  - xóa một quyên góp cụ thể thông qua nút “Delete”
- Sử dụng AsyncTask trong ReportActivity

```

private class GetTask extends AsyncTask<String, Void, Donation> {

    protected ProgressDialog dialog;
    protected Context context;

    public GetTask(Context context) {
        this.context = context;
    }

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        this.dialog = new ProgressDialog(context, 1);
        this.dialog.setMessage("Retrieving Donation Details");
        this.dialog.show();
    }

    @Override
    protected Donation doInBackground(String... params) {

        try {
            return (Donation) DonationApi.get((String) params[0], (String)
params[1]);
        } catch (Exception e) {
            Log.v("donate", "ERROR : " + e);
            e.printStackTrace();
        }
    }
}

```

```
        return null;
    }

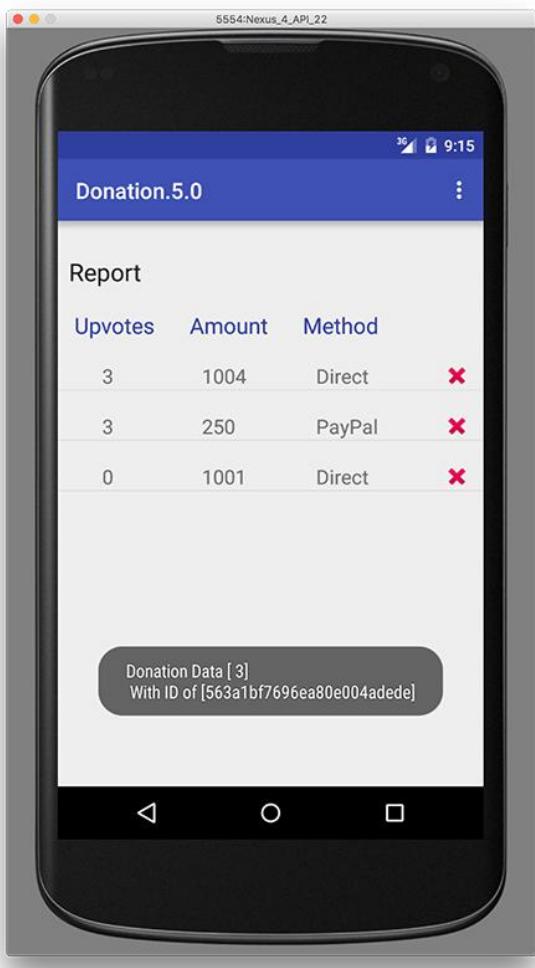
    @Override
    protected void onPostExecute(Donation result) {
        super.onPostExecute(result);

        Donation donation = result;

        Toast.makeText(Report.this, "Donation Data [ " + donation.upvotes +
                "]\n " +
                "With ID of [ " + donation._id + "]", Toast.LENGTH_LONG).show();

        if (dialog.isShowing())
            dialog.dismiss();
    }
}
```

- Sửa các lỗi xuất hiện và thử thêm *OnItemClickListener()* vào trong class Report để hiển thị Toast khi một dòng được lựa chọn bằng cách sử dụng AsyncTask phía trên



- Để cài đặt chức năng xóa, chúng ta cần cài đặt một *OnClickListener* interface

```
private class DeleteTask extends AsyncTask<String, Void, String> {

    protected ProgressDialog dialog;
    protected Context context;

    public DeleteTask(Context context) {
        this.context = context;
    }

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        this.dialog = new ProgressDialog(context, 1);
        this.dialog.setMessage("Deleting Donation");
        this.dialog.show();
    }

    @Override
    protected String doInBackground(String... params) {
```

```

        try {
            return (String) DonationApi.delete((String) params[0], (String)
params[1]);
        } catch (Exception e) {
            Log.v("donate", "ERROR : " + e);
            e.printStackTrace();
        }
        return null;
    }

    @Override
    protected void onPostExecute(String result) {
        super.onPostExecute(result);

        String s = result;
        Log.v("donate", "DELETE REQUEST : " + s);

        new GetAllTask(Report.this).execute("/donations");

        if (dialog.isShowing())
            dialog.dismiss();
    }
}

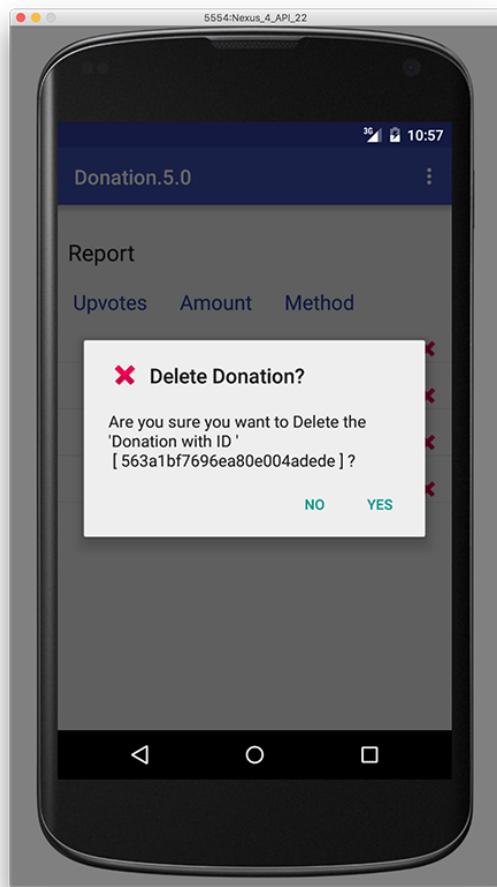
public void onDonationDelete(final Donation donation) {
    String stringId = donation._id;
    AlertDialog.Builder builder = new AlertDialog.Builder(this);
    builder.setTitle("Delete Donation?");
    builder.setIcon(android.R.drawable.ic_delete);
    builder.setMessage("Are you sure you want to Delete the \\'Donation with ID\n"
        + stringId + " ] ?");
    builder.setCancelable(false);

    builder.setPositiveButton("Yes", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int id) {

            new DeleteTask(Report.this).execute("/donations", donation._id);
        }
    }).setNegativeButton("No", new DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int id) {
            dialog.cancel();
        }
    });
    AlertDialog alert = builder.create();
    alert.show();
}

```

- Thử gọi phương thức `onDonationDelete()` trong `onClick()` và quan sát khi nhấn nút “Delete”



## BÀI THỰC HÀNH 14: Hỗ trợ REST/Cloud cho ứng dụng (tiếp)

### Bước 9. Xử lý khi thêm một quyên góp mới

- Đây là nội dung đầy đủ của Report.java

```
package ie.app.activities;

import android.app.AlertDialog;
import android.app.ProgressDialog;
import android.content.Context;
import android.content.DialogInterface;
import android.os.AsyncTask;
import android.support.v4.widget.SwipeRefreshLayout;
import android.os.Bundle;
import android.util.Log;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
```

```
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemClickListener;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.TextView;
import android.widget.Toast;

import java.util.List;

import ie.app.R;
import ie.app.api.DonationApi;
import ie.app.models.Donation;

public class Report extends Base implements OnItemClickListener, OnClickListener {
    ListView listView;
    DonationAdapter adapter;
    SwipeRefreshLayout mSwipeRefreshLayout;

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

        listView = (ListView) findViewById(R.id.reportList);
        mSwipeRefreshLayout = (SwipeRefreshLayout)
        findViewById(R.id.report_swipe_refresh_layout);

        new GetAllTask(this).execute("/donations");

        mSwipeRefreshLayout.setOnRefreshListener(new
        SwipeRefreshLayout.OnRefreshListener() {
            @Override
            public void onRefresh() {
                new GetAllTask(Report.this).execute("/donations");
            }
        });
    }

    @Override
    public void onItemClick(AdapterView<?> arg0, View row, int pos, long id) {
        new GetTask(this).execute("/donations", row.getTag().toString());
    }

    @Override
    public void onClick(View view) {
```

```

        if (view.getTag() instanceof Donation) {
            onDonationDelete((Donation) view.getTag());
        }
    }

    public void onDonationDelete(final Donation donation) {
        String stringId = donation._id;
        AlertDialog.Builder builder = new AlertDialog.Builder(this);
        builder.setTitle("Delete Donation?");
        builder.setIcon(android.R.drawable.ic_delete);
        builder.setMessage("Are you sure you want to Delete the 'Donation with ID\n"
            + stringId + " ] ?");
        builder.setCancelable(false);

        builder.setPositiveButton("Yes", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {

                new DeleteTask(Report.this).execute("/donations", donation._id);
            }
        }).setNegativeButton("No", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int id) {
                dialog.cancel();
            }
        });
        AlertDialog alert = builder.create();
        alert.show();
    }

    private class GetAllTask extends AsyncTask<String, Void, List<Donation>> {

        protected ProgressDialog dialog;
        protected Context context;

        public GetAllTask(Context context) {
            this.context = context;
        }

        @Override
        protected void onPreExecute() {
            super.onPreExecute();
            this.dialog = new ProgressDialog(context, 1);
            this.dialog.setMessage("Retrieving Donations List");
            this.dialog.show();
        }

        @Override
        protected List<Donation> doInBackground(String... params) {

            try {

```

```
        return (List<Donation>) DonationApi.getAll((String) params[0]);
    } catch (Exception e) {
        Log.v("ASYNC", "ERROR : " + e);
        e.printStackTrace();
    }
    return null;
}

@Override
protected void onPostExecute(List<Donation> result) {
    super.onPostExecute(result);

    app.donations = result;
    adapter = new DonationAdapter(context, app.donations);
    listView.setAdapter(adapter);
    listView.setOnItemClickListener(Report.this);
    mSwipeRefreshLayout.setRefreshing(false);

    if (dialog.isShowing())
        dialog.dismiss();
}
}

private class GetTask extends AsyncTask<String, Void, Donation> {

    protected ProgressDialog dialog;
    protected Context context;

    public GetTask(Context context) {
        this.context = context;
    }

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        this.dialog = new ProgressDialog(context, 1);
        this.dialog.setMessage("Retrieving Donation Details");
        this.dialog.show();
    }

    @Override
    protected Donation doInBackground(String... params) {

        try {
            return (Donation) DonationApi.get((String) params[0], (String)
params[1]);
        } catch (Exception e) {
            Log.v("donate", "ERROR : " + e);
            e.printStackTrace();
        }
    }
}
```

```
        return null;
    }

    @Override
    protected void onPostExecute(Donation result) {
        super.onPostExecute(result);

        Donation donation = result;

        Toast.makeText(Report.this, "Donation Data [ " + donation.upvotes +
" ]\n " +
                "With ID of [ " + donation._id + " ]", Toast.LENGTH_LONG).show();

        if (dialog.isShowing())
            dialog.dismiss();
    }
}

private class DeleteTask extends AsyncTask<String, Void, String> {

    protected ProgressDialog dialog;
    protected Context context;

    public DeleteTask(Context context) {
        this.context = context;
    }

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        this.dialog = new ProgressDialog(context, 1);
        this.dialog.setMessage("Deleting Donation");
        this.dialog.show();
    }

    @Override
    protected String doInBackground(String... params) {

        try {
            return (String) DonationApi.delete((String) params[0], (String)
params[1]);
        } catch (Exception e) {
            Log.v("donate", "ERROR : " + e);
            e.printStackTrace();
        }
        return null;
    }

    @Override
    protected void onPostExecute(String result) {
```

```

super.onPostExecute(result);

String s = result;
Log.v("donate", "DELETE REQUEST : " + s);

new GetAllTask(Report.this).execute("/donations");

if (dialog.isShowing())
    dialog.dismiss();
}

}

class DonationAdapter extends ArrayAdapter<Donation> {
    private Context context;
    public List<Donation> donations;

    public DonationAdapter(Context context, List<Donation> donations) {
        super(context, R.layout.row_donate, donations);
        this.context = context;
        this.donations = donations;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        LayoutInflator inflater = (LayoutInflator) context
            .getSystemService(Context.LAYOUT_INFLATER_SERVICE);

        View view = inflater.inflate(R.layout.row_donate, parent, false);
        Donation donation = donations.get(position);

        ImageView imgDelete = (ImageView) view.findViewById(R.id.imgDelete);
        imgDelete.setTag(donation);
        imgDelete.setOnClickListener(Report.this);

        TextView amountView = (TextView)
view.findViewById(R.id.row_amount);
        TextView methodView = (TextView)
view.findViewById(R.id.row_method);
        TextView upvotesView = (TextView)
view.findViewById(R.id.row_upvotes);

        amountView.setText("₹" + donation.amount);
        methodView.setText(donation.paymenttype);
        upvotesView.setText("₹" + donation.upvotes);

        view.setTag(donation._id); // setting the 'row' id to the id of the donation

        return view;
    }
}

```

```

    @Override
    public int getCount() {
        return donations.size();
    }
}

```

- Làm sao để chèn thêm một quyên góp vào danh sách lưu trên server
- Mở tệp **Donate.java** và xem cài đặt của *donateButtonPressed()* và *reset()*

```

public void donateButtonPressed (View view)
{
    String method = paymentMethod.getCheckedRadioButtonId() ==
R.id.PayPal ? "PayPal" : "Direct";
    int donatedAmount = amountPicker.getValue();
    if (donatedAmount == 0)
    {
        String text = amountText.getText().toString();
        if (!text.equals(""))
            donatedAmount = Integer.parseInt(text);
    }
    if (donatedAmount > 0)
    {
        app.newDonation(new Donation(donatedAmount, method, 0));
        progressBar.setProgress(app.totalDonated);
        String totalDonatedStr = "$" + app.totalDonated;
        amountTotal.setText(totalDonatedStr);
    }
}

@Override
public void reset(MenuItem item)
{
    app.totalDonated = 0;
    amountTotal.setText("$" + app.totalDonated);
}

```

- Sử dụng **AsyncTask** để thêm một quyên góp lưu trữ trên server, refactor phương thức *donateButtonPressed()* để thực hiện task này

```

private class InsertTask extends AsyncTask<Object, Void, String> {

    protected ProgressDialog dialog;
    protected Context context;

    public InsertTask(Context context)
    {
        this.context = context;
    }
}

```

```

    }

    @Override
    protected void onPreExecute() {
        super.onPreExecute();
        this.dialog = new ProgressDialog(context, 1);
        this.dialog.setMessage("Saving Donation....");
        this.dialog.show();
    }

    @Override
    protected String doInBackground(Object... params) {

        String res = null;
        try {
            Log.v("donate", "Donation App Inserting");

        }

        catch(Exception e)
        {
            Log.v("donate","ERROR : " + e);
            e.printStackTrace();
        }
        return res;
    }

    @Override
    protected void onPostExecute(String result) {
        super.onPostExecute(result);
    }
}

```

## BÀI THỰC HÀNH 15: Hỗ trợ REST/Cloud cho ứng dụng (tiếp)

### Bước 10. Xử lý xóa và reset

- Cài đặt chức năng **Reset** sử dụng AsyncTask

```

private class ResetTask extends AsyncTask<Object, Void, String> {

    protected ProgressDialog dialog;
    protected Context context;

    public ResetTask(Context context)
    {
        this.context = context;
    }
}

```

```

@Override
protected void onPreExecute() {
    super.onPreExecute();
    this.dialog = new ProgressDialog(context, 1);
    this.dialog.setMessage("Deleting Donations....");
    this.dialog.show();
}

@Override
protected String doInBackground(Object... params) {

    String res = null;
    try {
        res = DonationApi.deleteAll((String)params[0]);
    }

    catch(Exception e)
    {
        Log.v("donate"," RESET ERROR : " + e);
        e.printStackTrace();
    }
    return res;
}

@Override
protected void onPostExecute(String result) {
    super.onPostExecute(result);

    app.totalDonated = 0;
    progressBar.setProgress(app.totalDonated);
    amountTotal.setText("$" + app.totalDonated);

    if (dialog.isShowing())
        dialog.dismiss();
}
}

```

- Chỉ cần thêm một dòng code vào phương thức *reset()* để hoàn thành, hãy tự tìm hiểu và thêm vào
- Có một lỗi nhỏ trong ứng dụng cho phép người dùng vẫn có thể nhìn thấy màn hình Report ngay cả khi chúng ta đã “Reset”, hãy sửa lỗi này
- Thêm tính năng hỏi xác nhận của người dùng khi muốn “Reset”

