

# Mobile Application Development

Higher Diploma in Science in Computer Science

---

Produced  
by

Eamonn de Leastar (edeleastar@wit.ie)

Department of Computing, Maths & Physics  
Waterford Institute of Technology

<http://www.wit.ie>

<http://elearning.wit.ie>



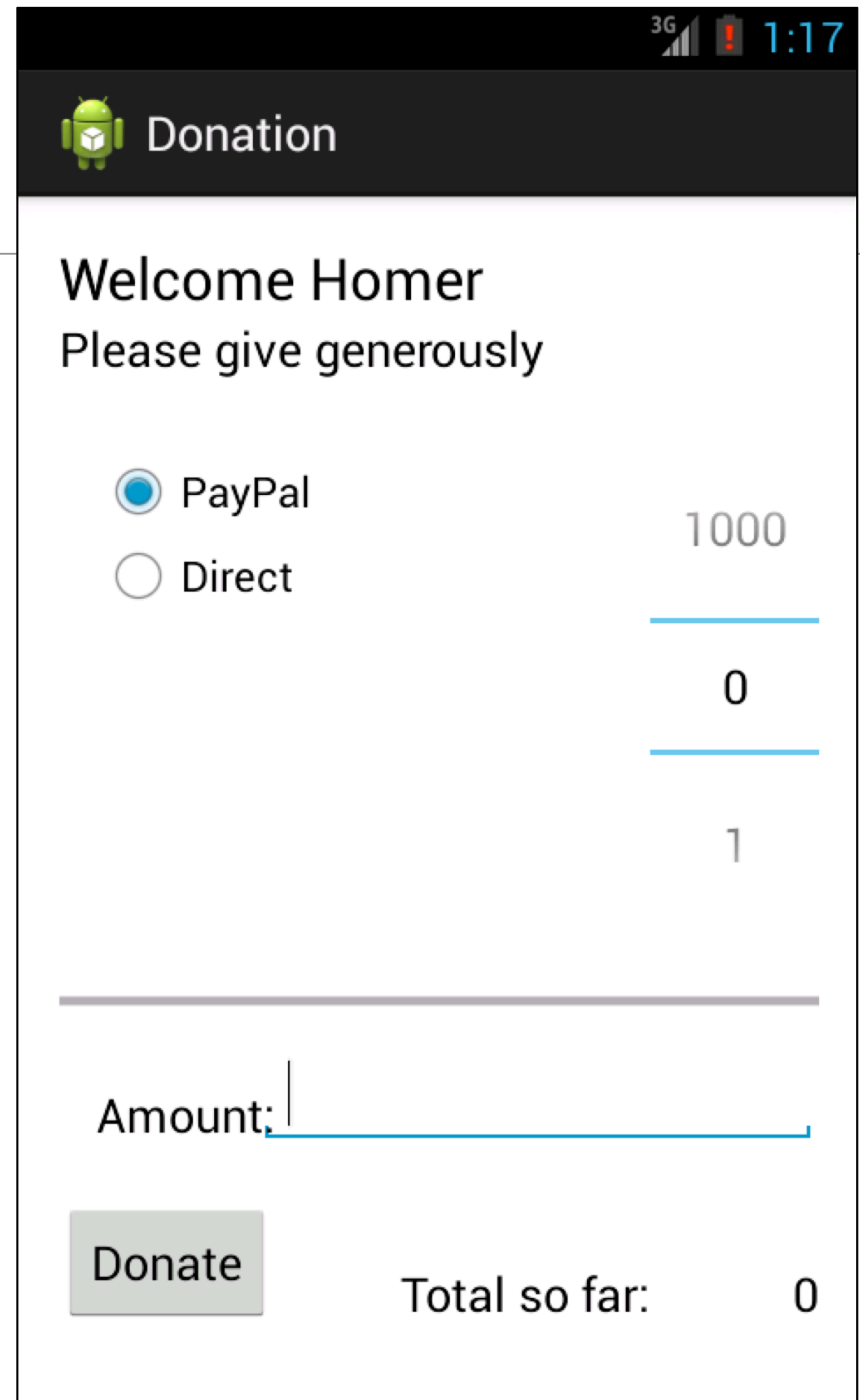
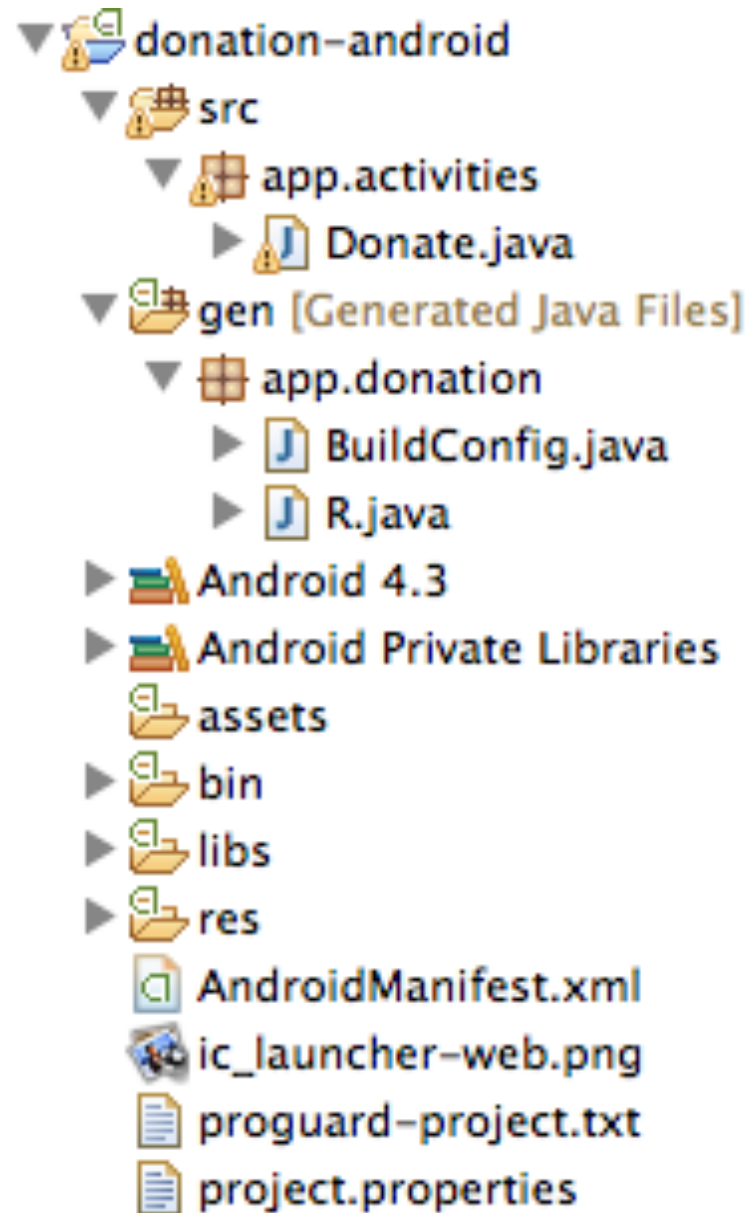
Waterford Institute of Technology  
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

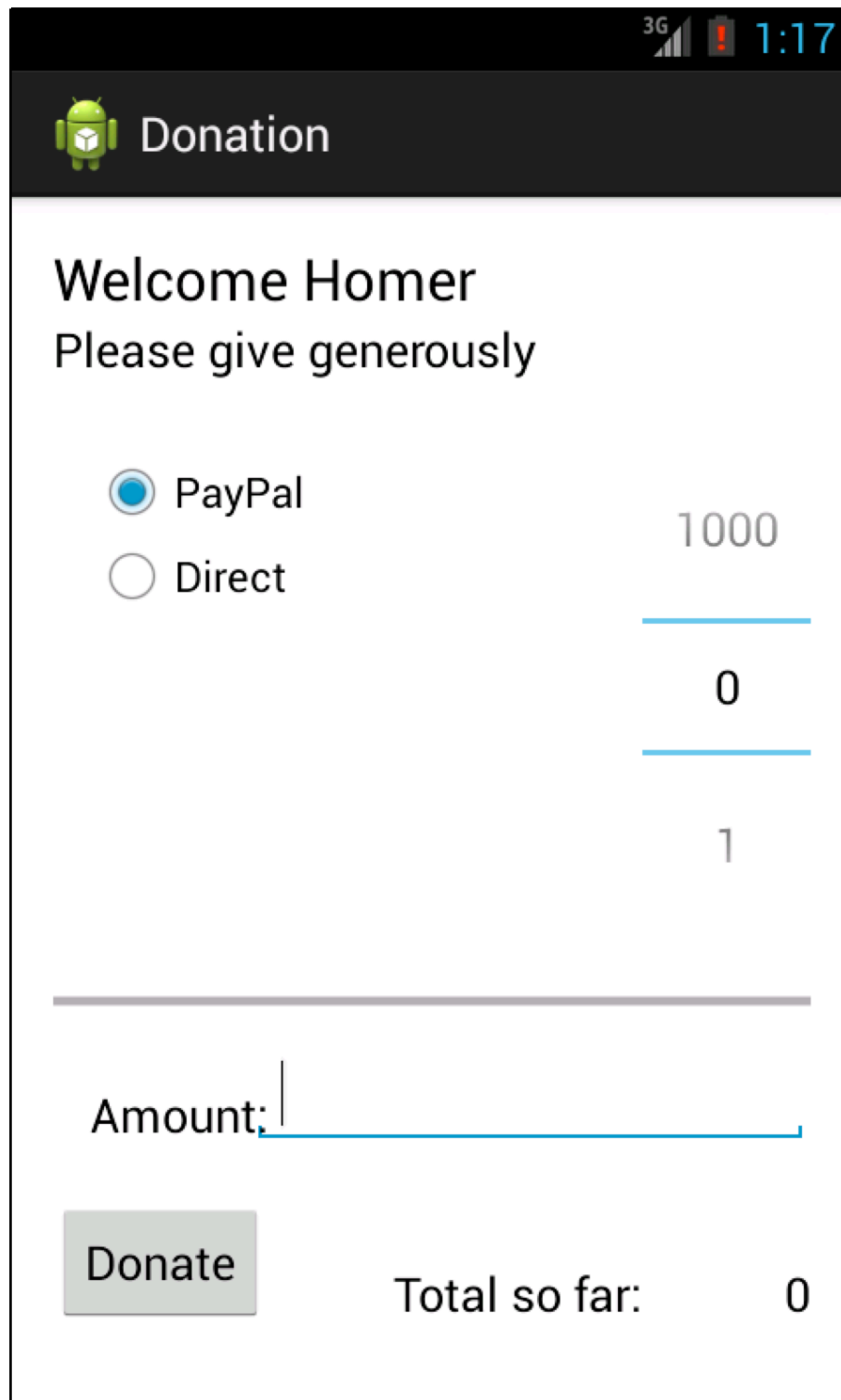


# donation-android case study - v2

---

# donation v1





```
public class Donate extends Activity
{
    private int          totalDonated = 0;
    private int          target = 10000;

    private RadioGroup    paymentMethod;
    private ProgressBar   progressBar;
    private NumberPicker  amountPicker;
    private TextView      amountText;
    private TextView      amountTotal;

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

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

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

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

# donate button event handler

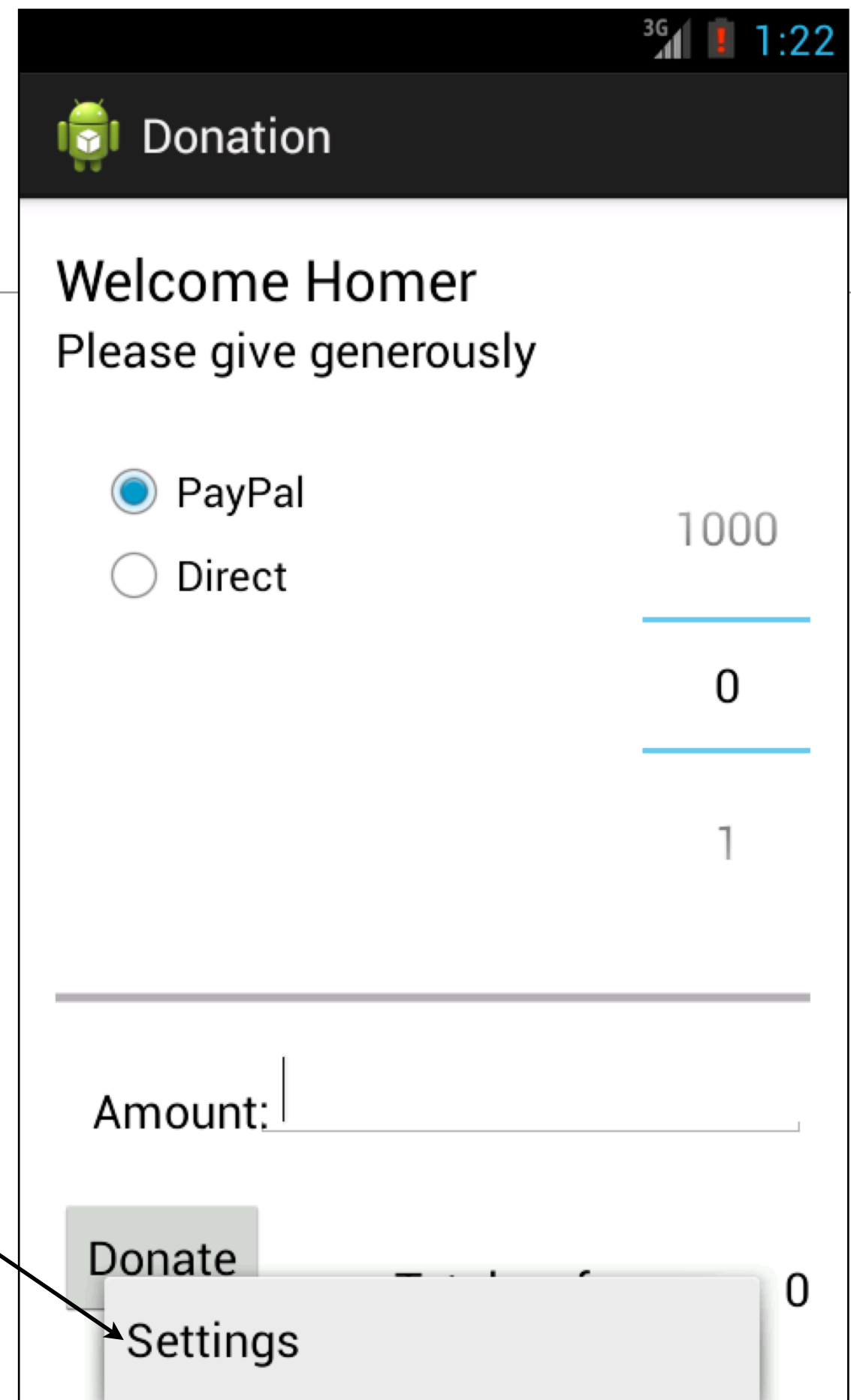
---

```
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 (totalDonated > target)
    {
        Toast toast = Toast.makeText(this, "Target Exceeded!", Toast.LENGTH_SHORT);
        toast.show();
        Log.v("Donate", "Target Exceeded: " + totalDonated);
    }
    else
    {
        totalDonated = totalDonated + donatedAmount;
        progressBar.setProgress(totalDonated);
        Log.v("Donate", amountPicker.getValue() + " donated by " + method + "\nCurrent total " + totalDonated);
    }
    String totalDonatedStr = "$" + totalDonated;
    amountTotal.setText(totalDonatedStr);
}
```

# Menus

- Pressing the “Menu” button on emulator brings up a menu with single entry:



# Menu Load

```
public class Donate extends Activity
{
    //...

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

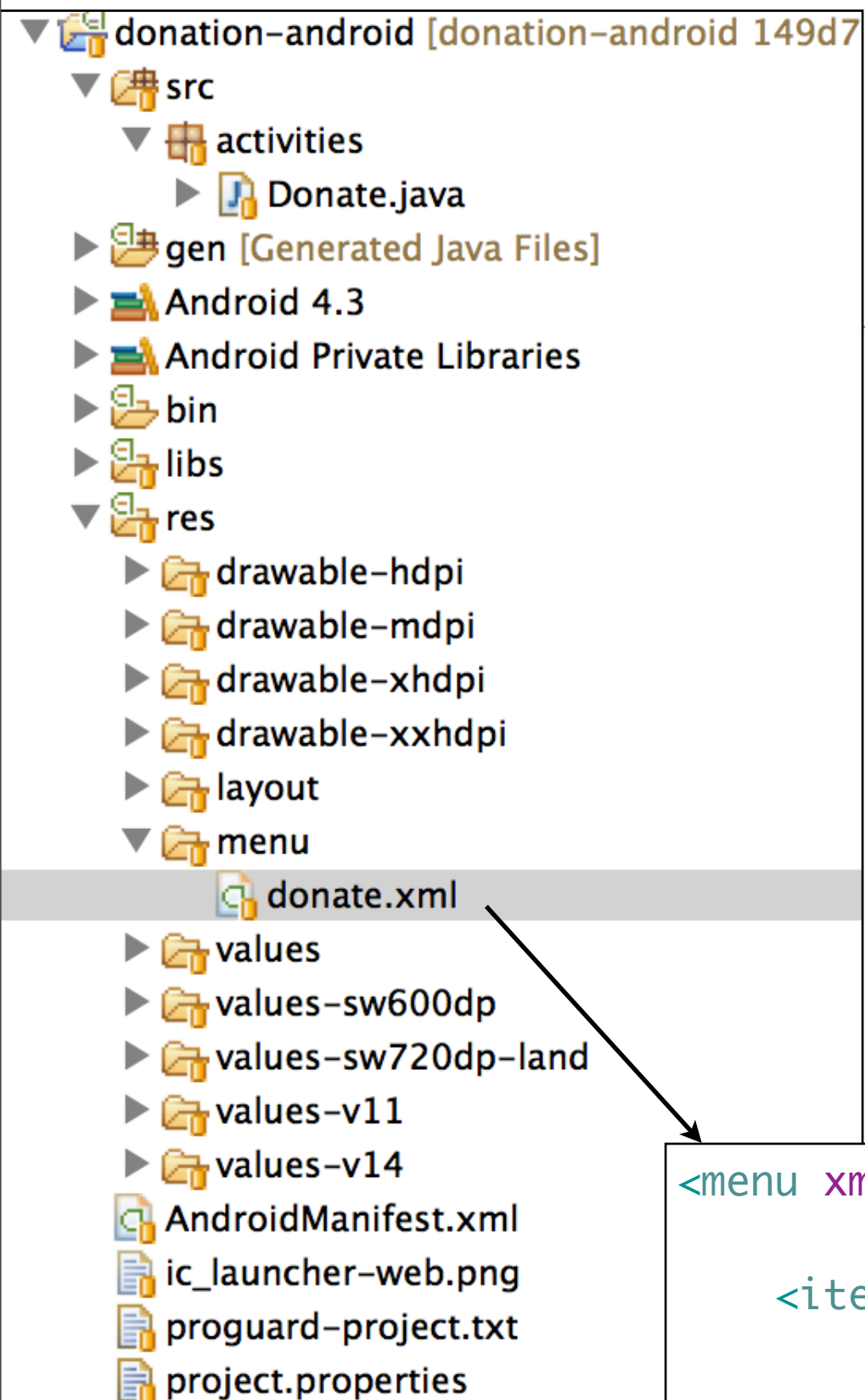
    //...
}
```

# Menu Specification

```
<menu xmlns:android="http://schemas.android.com/apk/res/android" >


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

</menu>
```



# Menu Event Handler

- Display 'Toast' for a few seconds

 Donation

Welcome Homer  
Please give generously

☒ PayPal

1000

☐ Direct

0

1

Amount: 4

Settings Selected

Donate

Total so far: 0

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



# New Menu Item

```
<menu xmlns:android="http://schemas.android.com/apk/res/android" >

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

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

</menu>
```

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

Amount:

Settings

Report

0

# Design new Report activity

The screenshot shows the Android Studio interface for designing a new Report activity. The main canvas displays a mobile app layout with a dark header bar containing an Android icon and the word "Donation". Below the header is a white container labeled "Report" which contains a list of six items, each with a title and a subtitle. The items are: "Item 1" (Sub Item 1), "Item 2" (Sub Item 2), "Item 3" (Sub Item 3), "Item 4" (Sub Item 4), "Item 5" (Sub Item 5), and "Item 6" (Sub Item 6). Blue dimension lines and green arrows indicate the layout constraints for the "Report" container.

On the right side, the "Outline" pane shows the hierarchy: "RelativeLayout" containing "Ab reportTitle (TextView) - \"Report\"" and "reportList (ListView)". The "Properties" pane at the bottom right shows the "Id" and "Content Descri..." properties for the selected element.

# activity\_report.xml

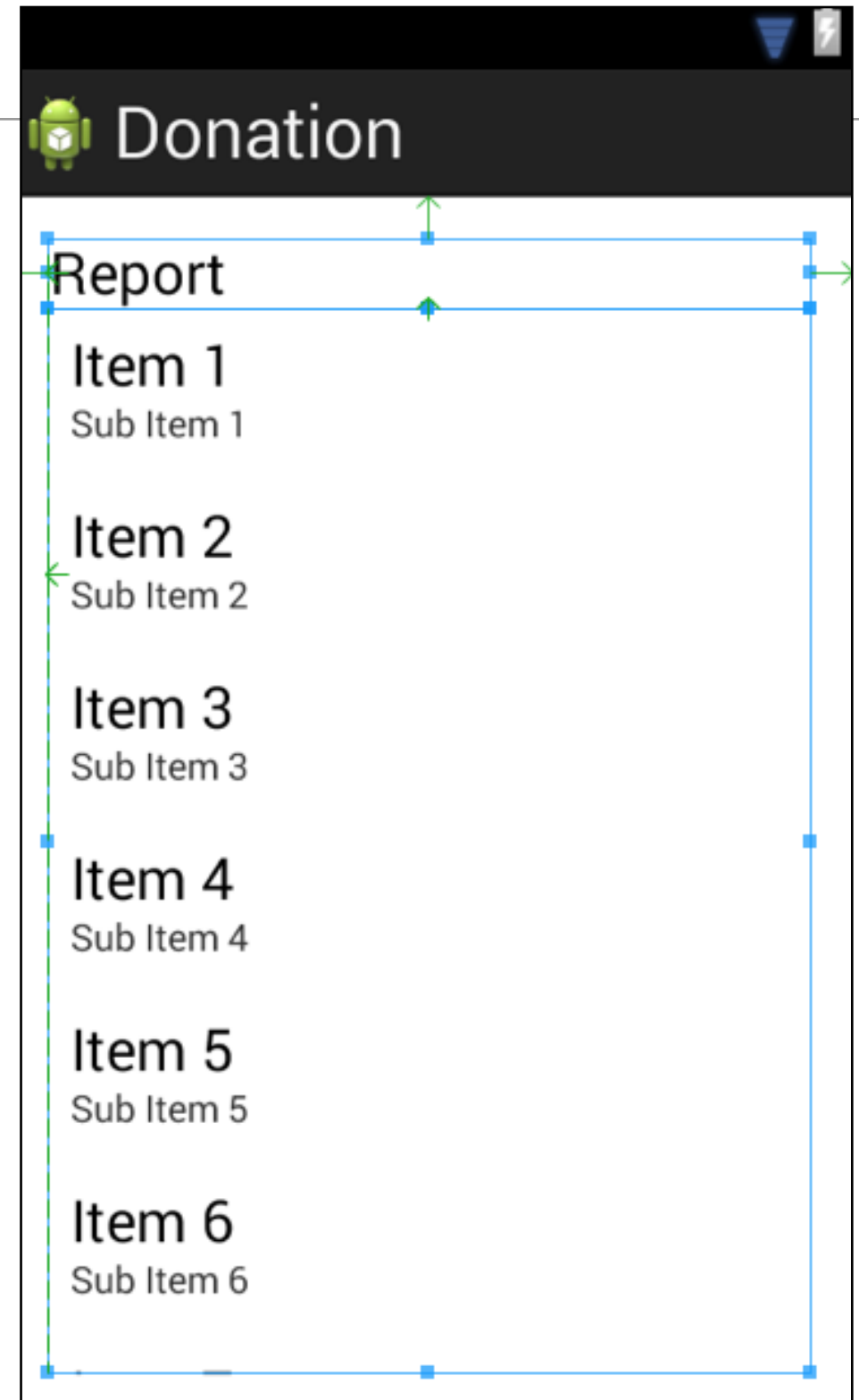
```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".Test" >

    <TextView
        android:id="@+id/reportTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:text="@string/reportTitle"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <ListView
        android:id="@+id/reportList"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/reportTitle"
        android:layout_below="@+id/reportTitle" >

    </ListView>

</RelativeLayout>
```



# ActivityReport

```
public class Report extends Activity
{
    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);
    }
}
```



Welcome Homer

## Report

Amount	Pay method
10	Direct
100	PayPal
1000	Direct
100	PayPal
5000	PayPal

# ActivityReport

```
public class Report extends Activity  
{  
    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);
```

```
}
```

```
}
```



Welcome Homer

## Report

Amount	Pay method
10	Direct
100	PayPal
1000	Direct
100	PayPal
5000	PayPal

# Application Object

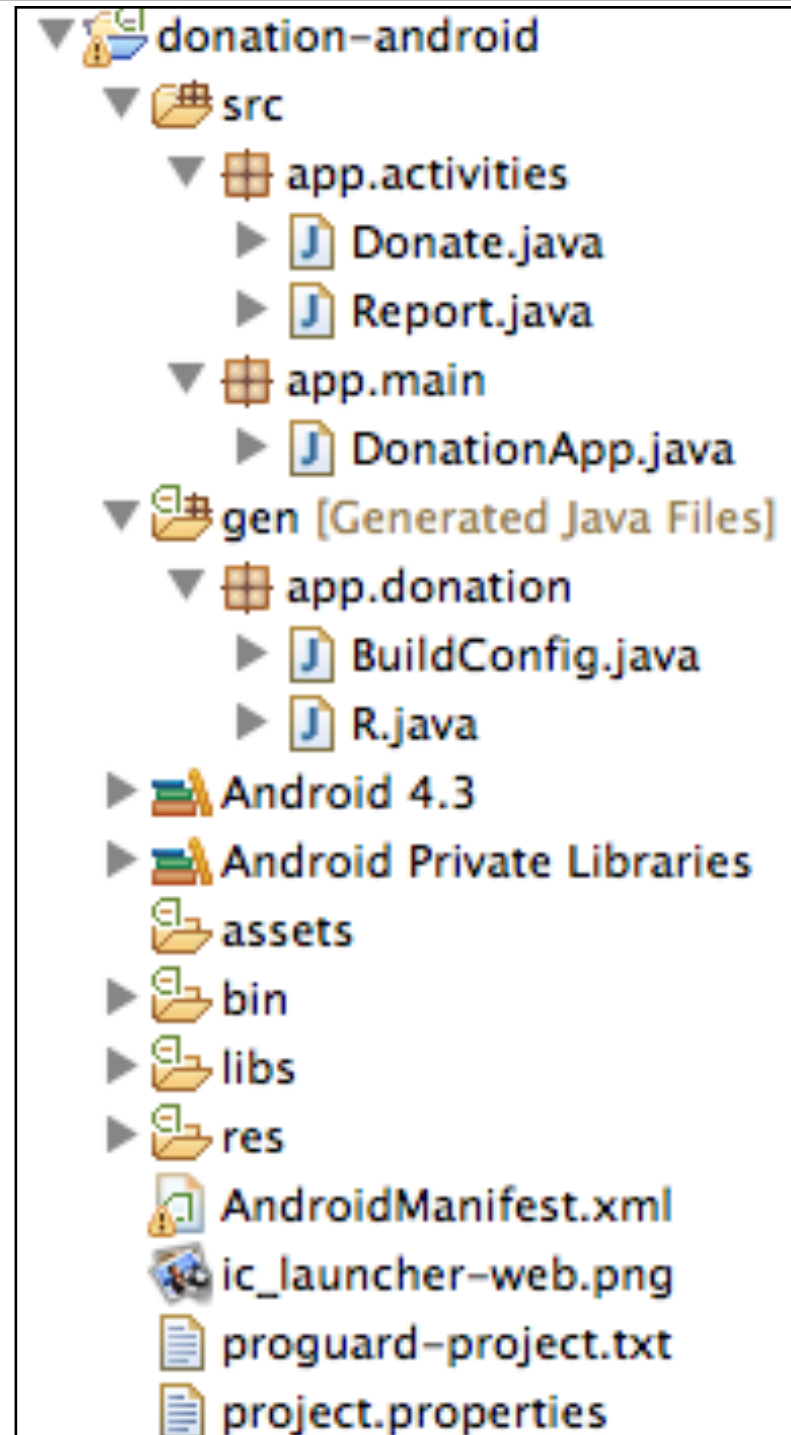
```
<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name"
    android:theme="@style/AppTheme"
    android:name="app.main.DonationApp">
```

```
package app.main;

import android.app.Application;
import android.util.Log;

public class DonationApp extends Application
{
    @Override
    public void onCreate()
    {
        super.onCreate();
        Log.v("Donation", "Donation App Started");
    }
}
```

- Activities come and go based on user interaction
- Application objects can be a useful 'anchor' for an android app
- Use it to hold information shared by all activities

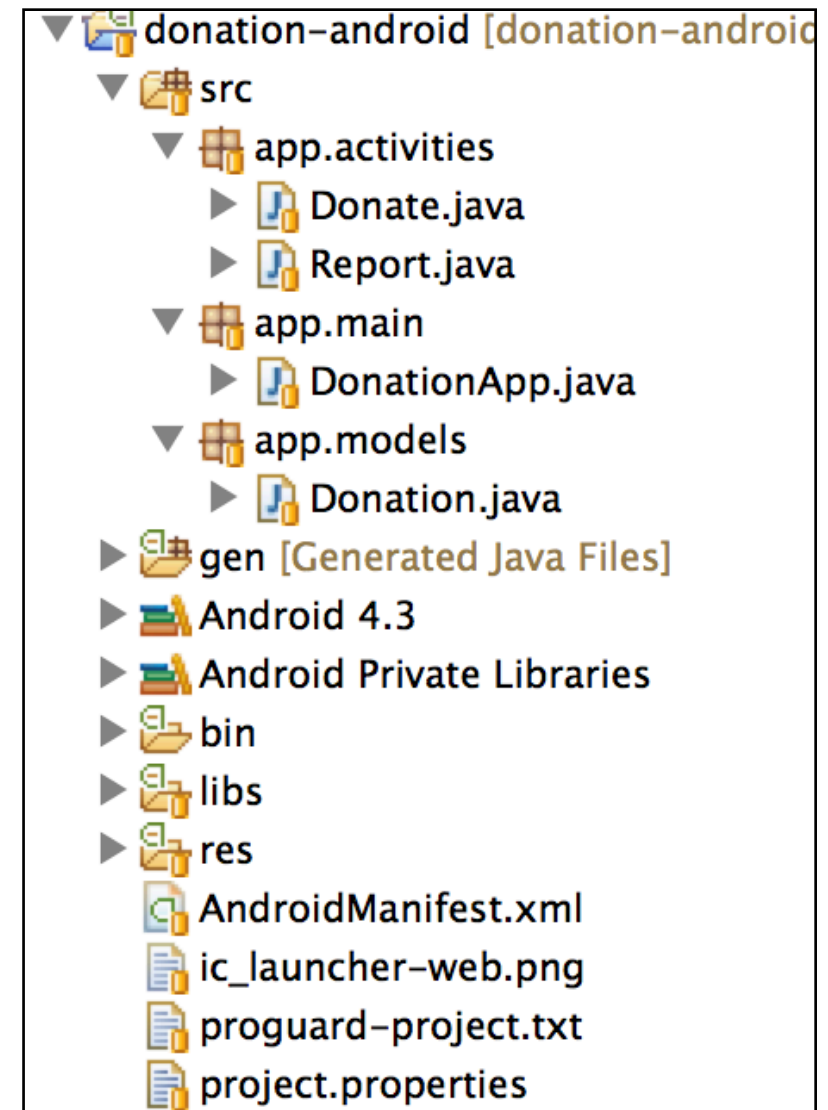


# Model Package

- Introduce a 'models' package - similar to play framework models package
- Model key application domain - candidates for objects to be stored in a database:
  - locally (sql\_lite)
  - remove (via API)

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

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



# Revised DonationApp

```
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 toast = Toast.makeText(this, "Target Exceeded!", Toast.LENGTH_SHORT);
            toast.show();
        }
        return targetAchieved;
    }

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

- Maintain list of donations
- Main current total
- Allow donations to me made (via 'newDonation')
- Track if total exceeded or not



# Donate Activity

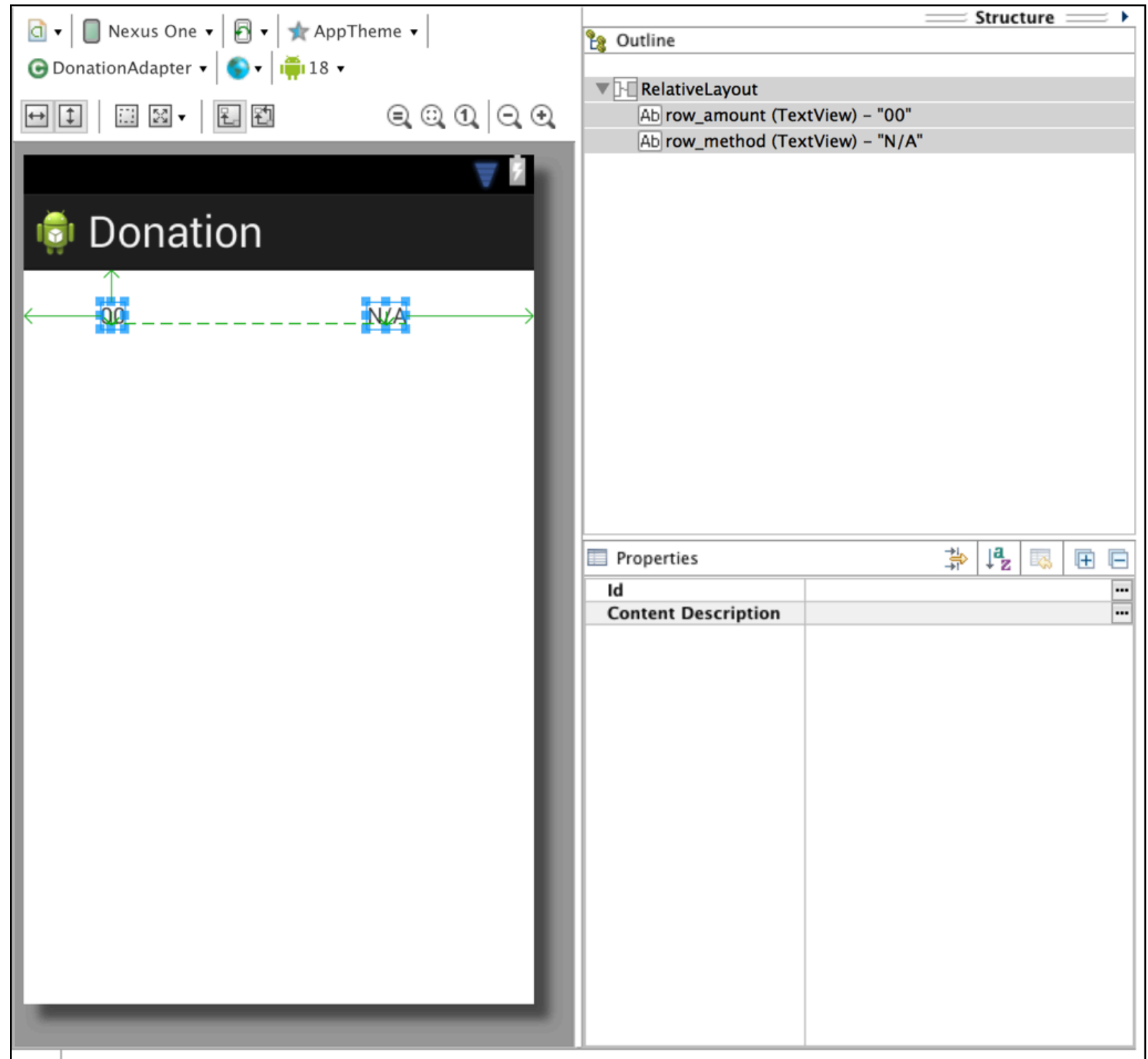
```
public class Donate extends Activity
{
    //...

    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));
            progressBar.setProgress(app.totalDonated);
            String totalDonatedStr = "$" + app.totalDonated;
            amountTotal.setText(totalDonatedStr);
        }
        amountText.setText("");
        amountPicker.setValue(0);
    }
    //..
}
```

- Use the Application Object to store donations

# row\_donate.xml

- Not all layouts need to be full screen activities
- A layout xml file is just a description of a set of UI elements.
- It can be a full activity, or loaded as a part of some other activity



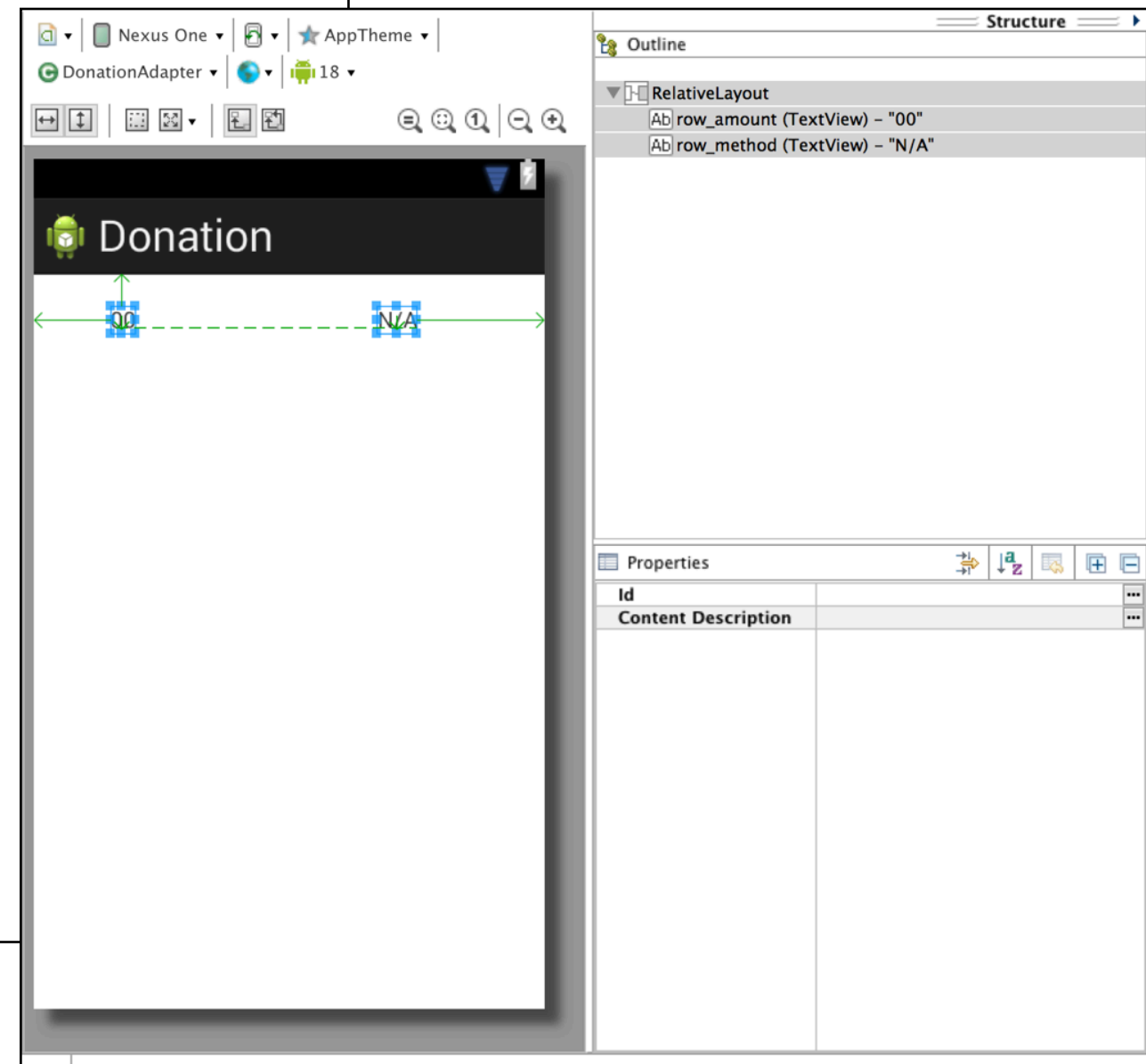
# row\_donate.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    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_alignBaseline="@+id/row_amount"
        android:layout_alignBottom="@+id/row_amount"
        android:layout_alignParentRight="true"
        android:layout_marginRight="79dp"
        android:text="@string/defaultMethod" />

</RelativeLayout>
```



# Revised Report Activity

```
public class Report extends Activity
{
    private ListView    listView;
    private DonationApp app;

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

        app = (DonationApp) getApplication();

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

- Remove hard coded list of donations
- Fetch current donations list from Application Object
- Pass this list to a 'DonationAdapter' - and give the adapter to the list view.

# DonationAdapter

- ‘Adapt’ a list of Donation objects for display in a ListView
- Report the size of the list when asked (getCount())
- Given a specific position - create a ‘View’ representing a row when asked
- This row is created using the row\_donate.xml layout we have just designed.

```
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)
    {
        LayoutInflater inflater
            = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);

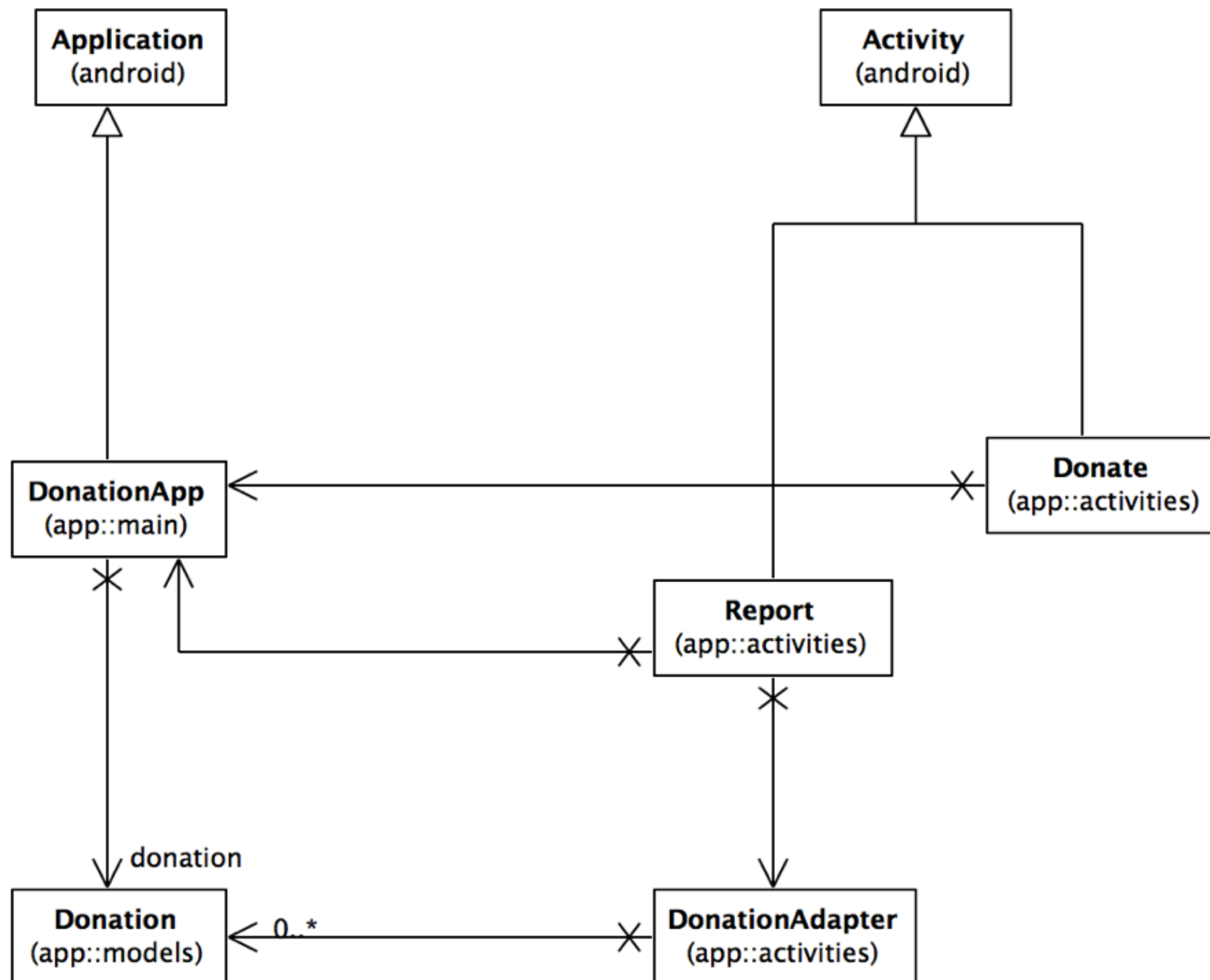
        View view = inflater.inflate(R.layout.row_donate, parent, false);
        Donation donation = donations.get(position);
        TextView amountView = (TextView) view.findViewById(R.id.row_amount);
        TextView methodView = (TextView) view.findViewById(R.id.row_method);

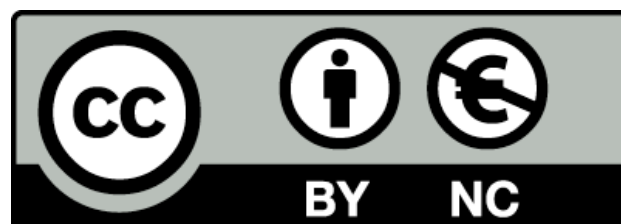
        amountView.setText("" + donation.amount);
        methodView.setText(donation.method);

        return view;
    }

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

# Donation v2 UML Model





Except where otherwise noted, this content is licensed under a Creative Commons Attribution-NonCommercial 3.0 License.

For more information, please see <http://creativecommons.org/licenses/by-nc/3.0/>



Waterford Institute of Technology  
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

