

Mobile Application Development

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

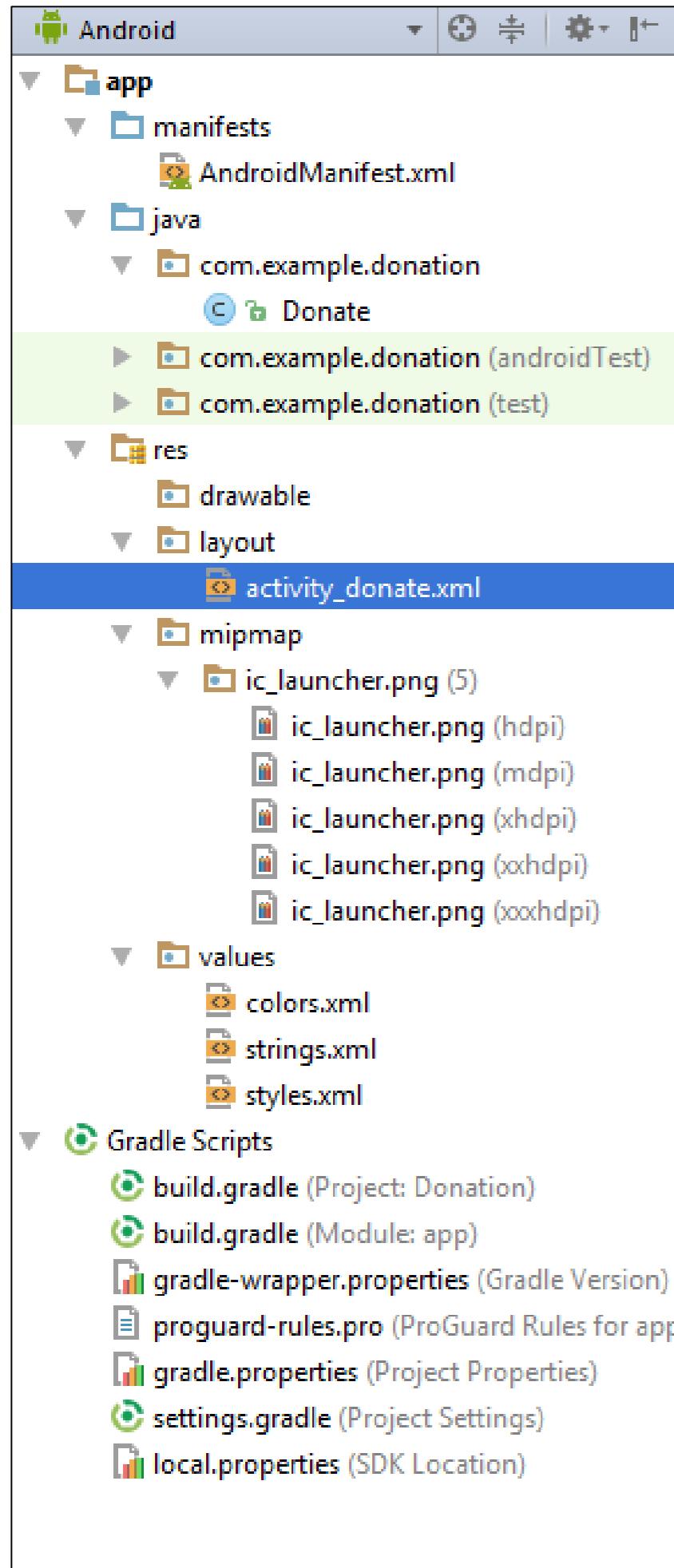


A First Android Application

Donation 3.0

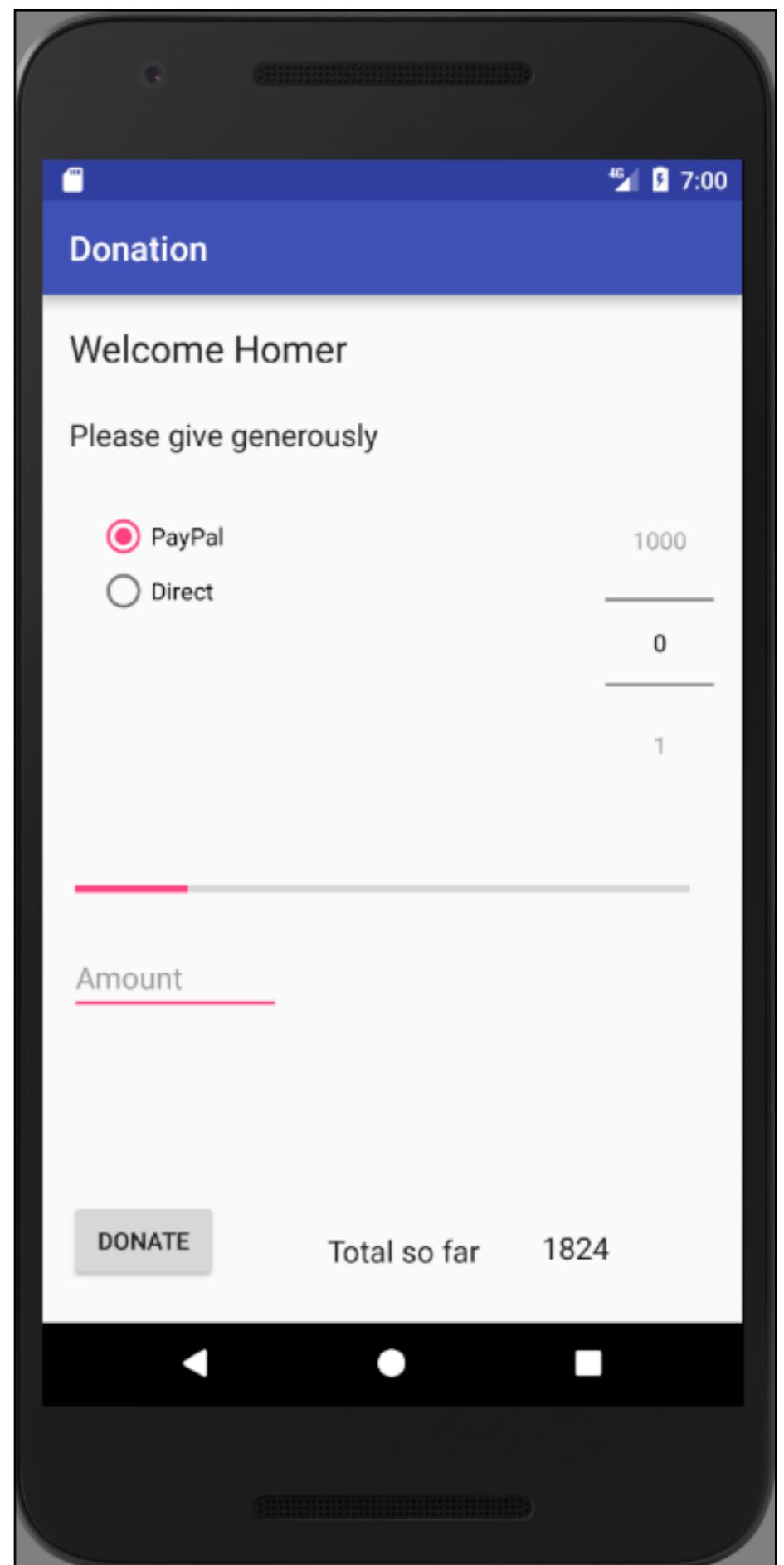
Lab 02 Exercises - SOLUTIONS

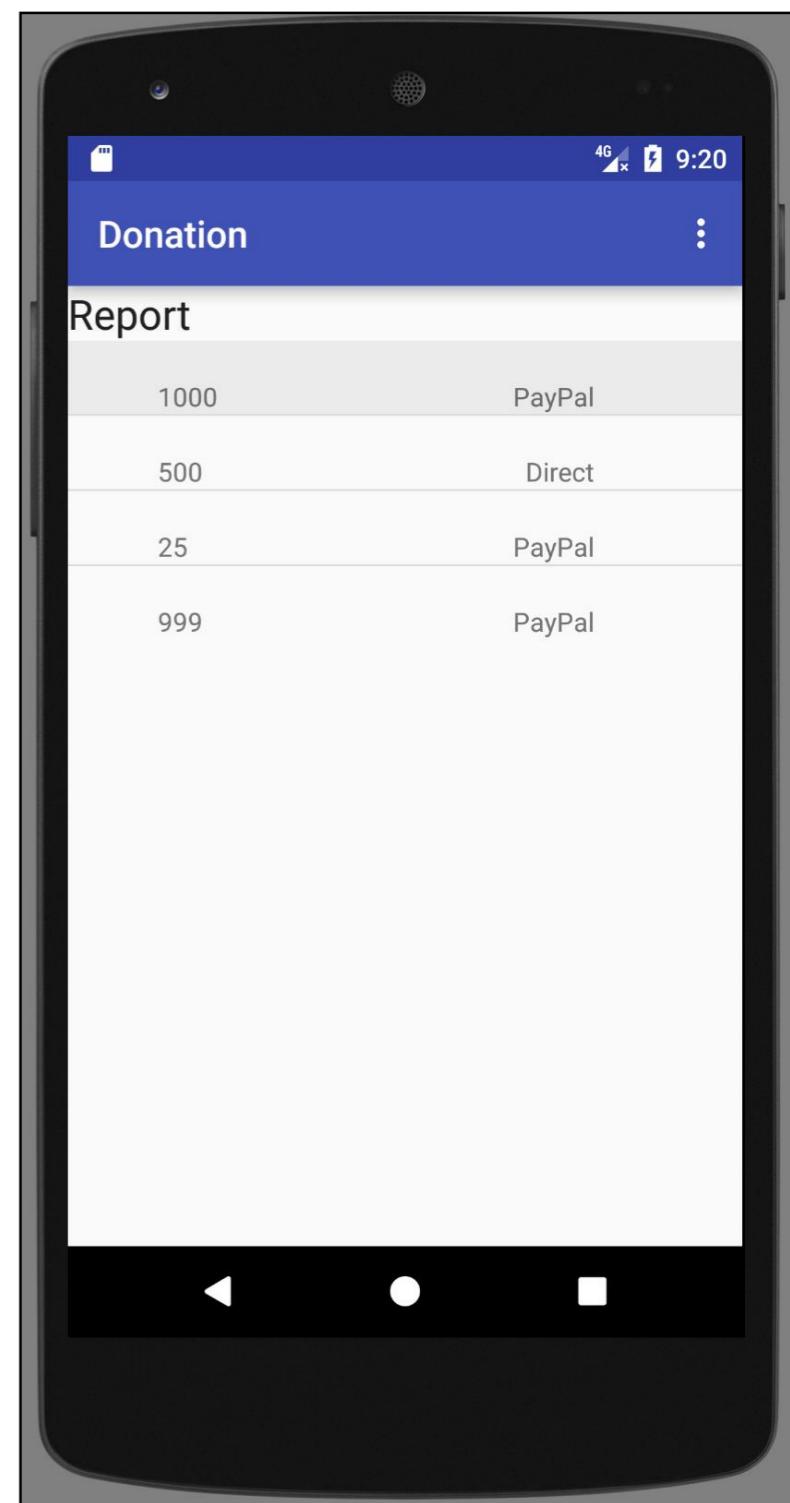
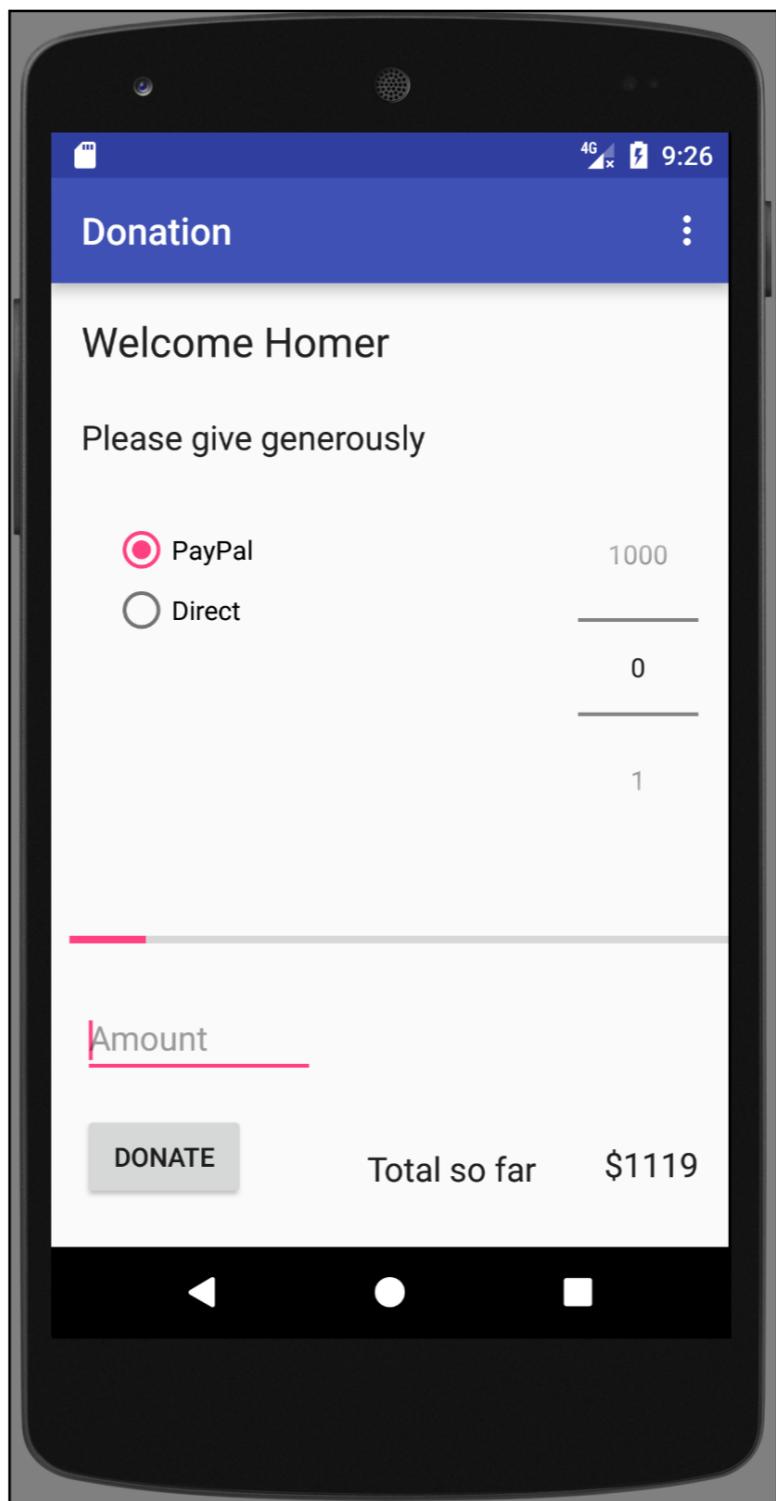
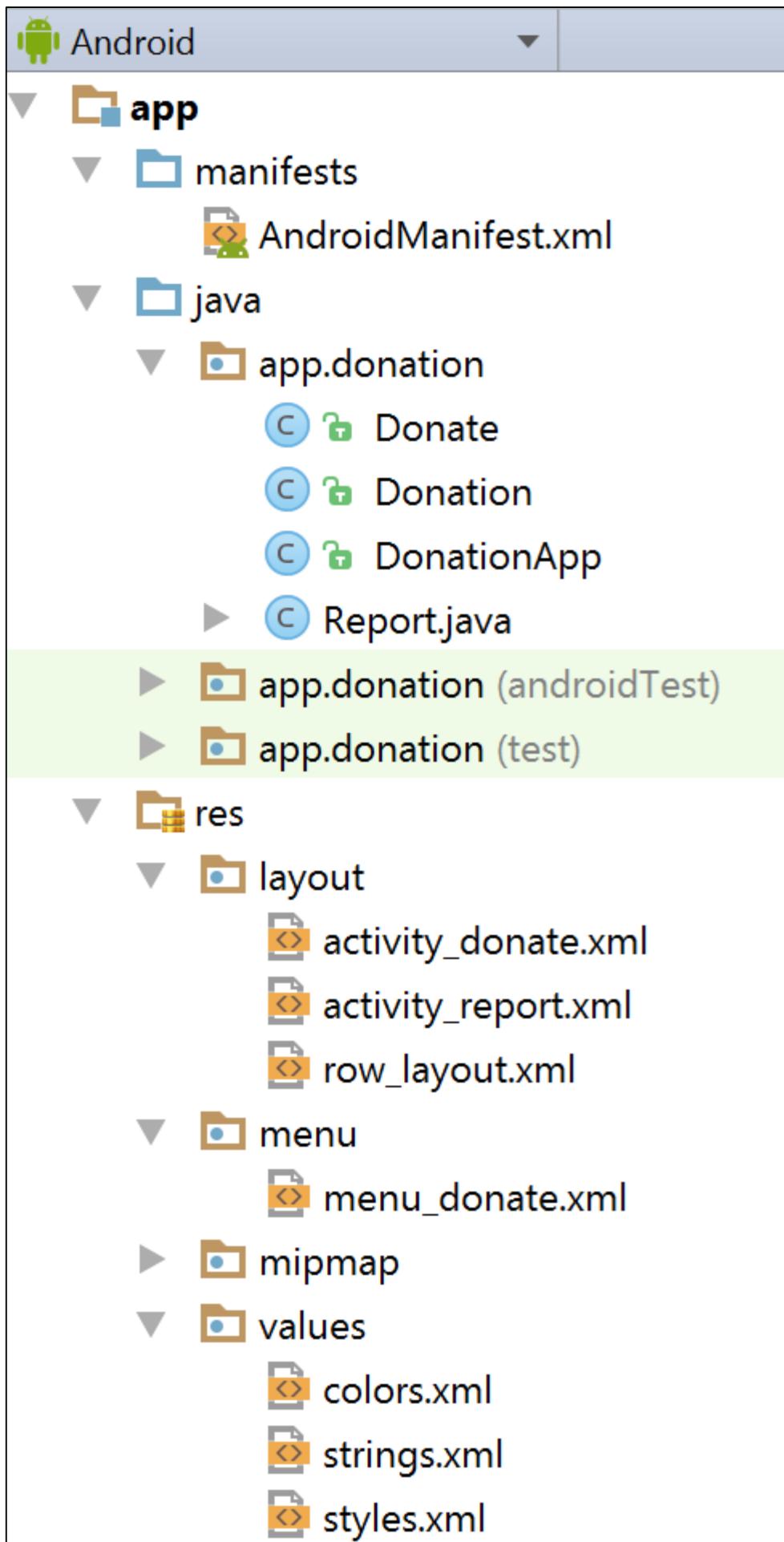




Donation V1

– single activity, no model

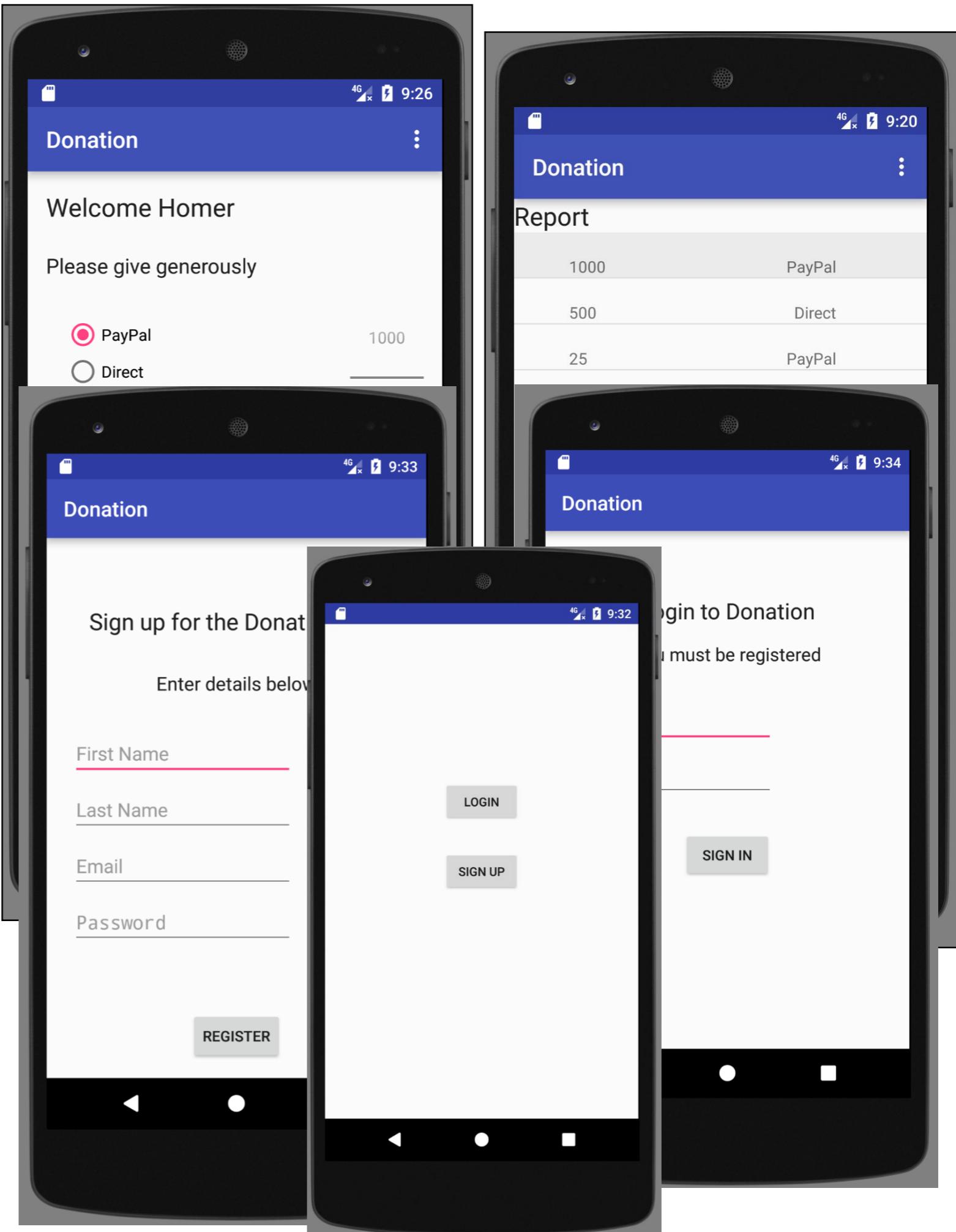
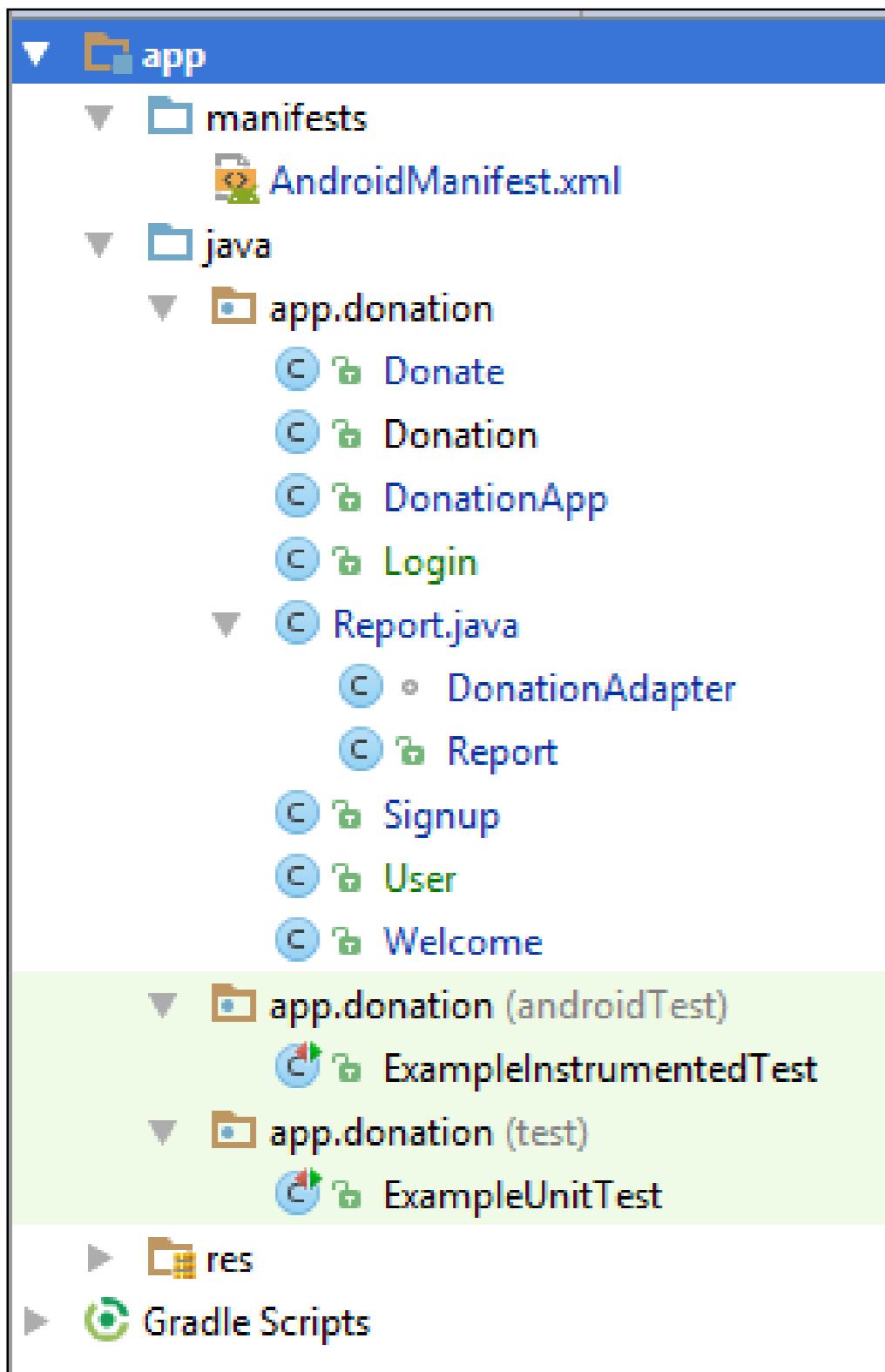




Donation V2 – two activities, app, donation model.

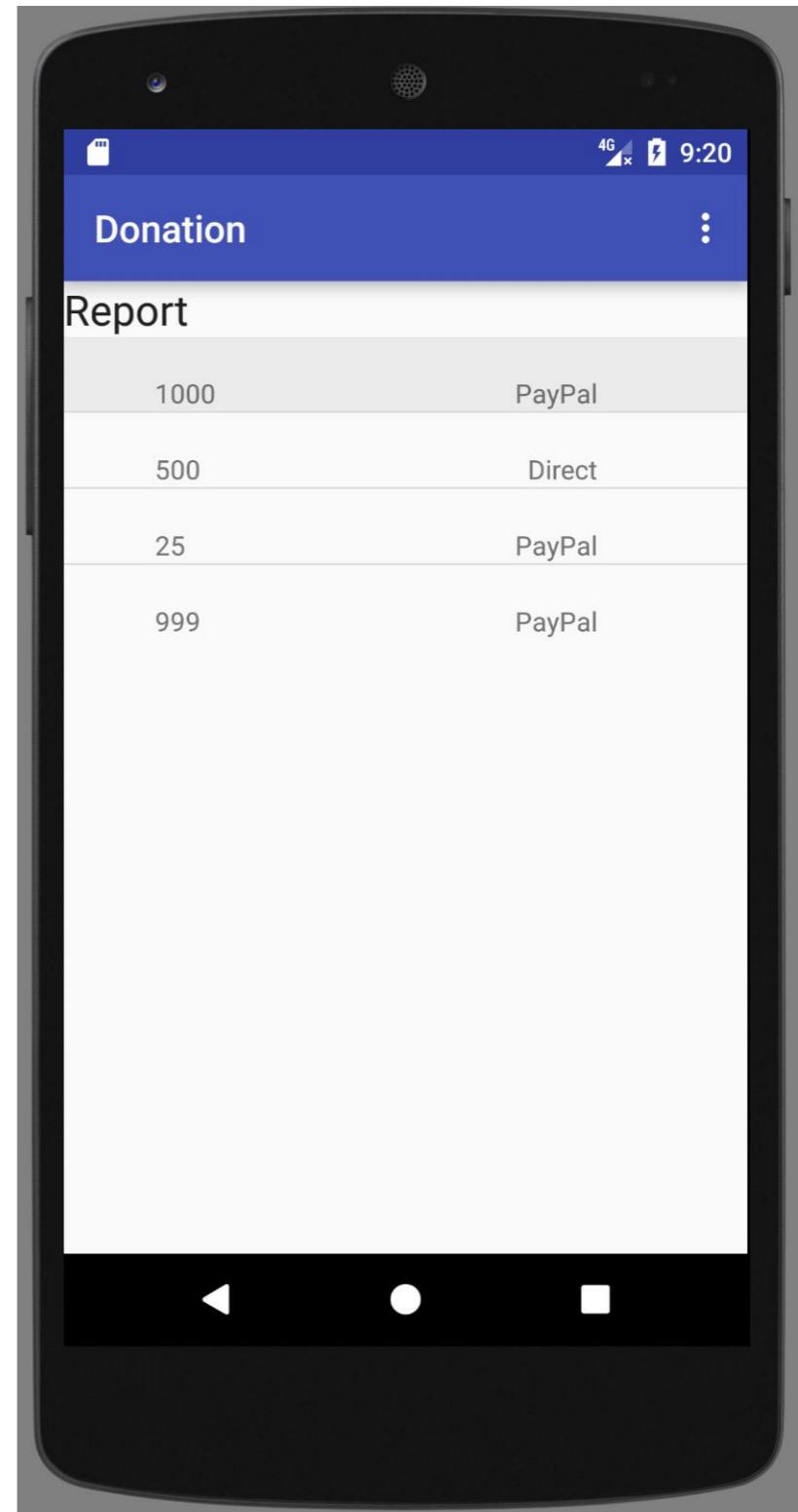
Building Donation V3.0

Donation V3



Exercise 1

- Run the app and insert amounts of varying lengths (1, 222, 23, 2323). Note that the second column - payment method - may be displayed at different positions. If this happens, fix it.
- Hint: each row is laid out by a `row_donate.xml` layout. The easiest way to fix this would be to experiment with the layout, and have the text fields aligned with the edges and not with each other.



Exercise 1 Solution

row_layout.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">

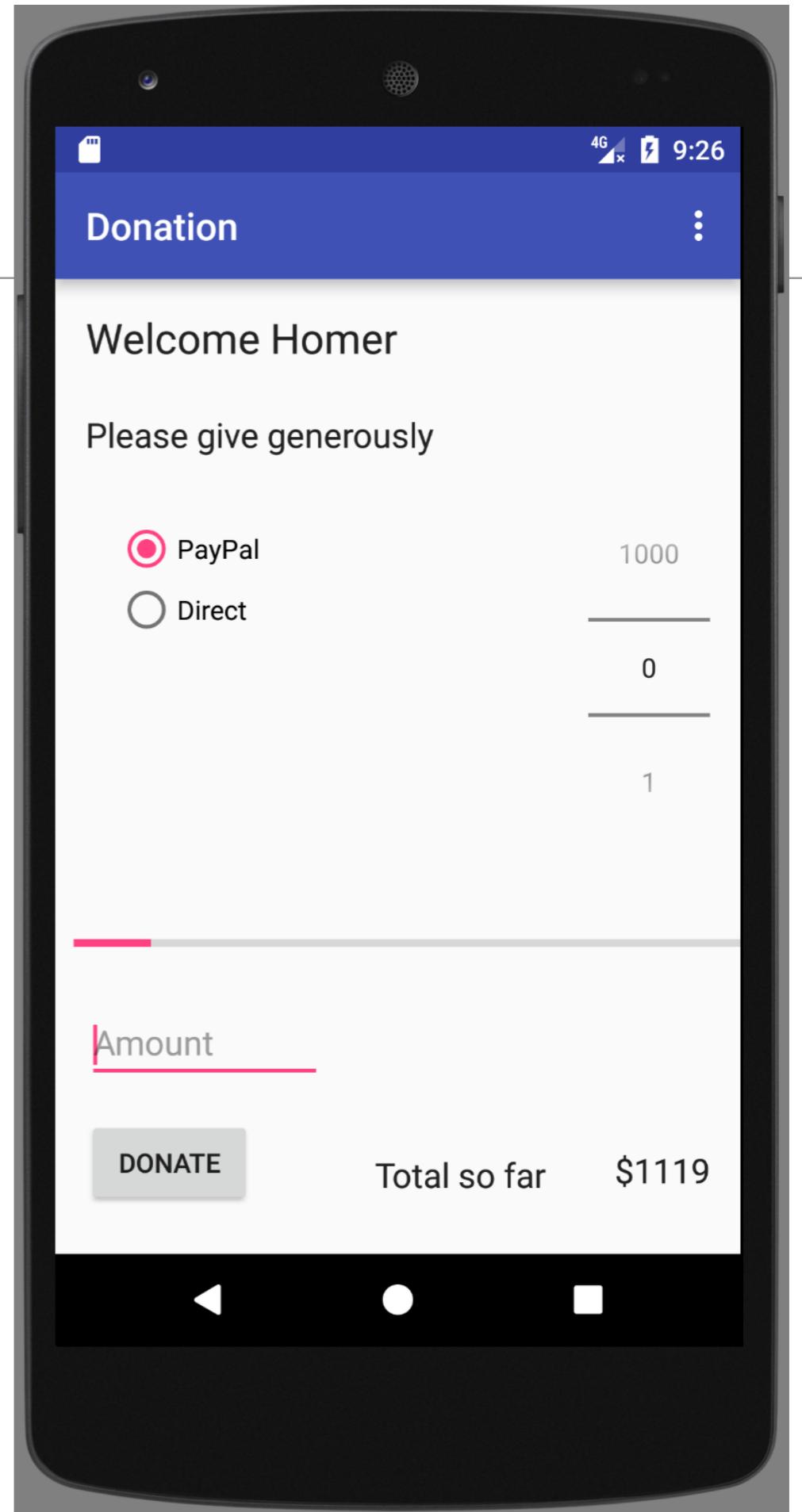
    <TextView
        android:id="@+id/row_amount"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_marginStart="48dp"
        android:layout_marginTop="20dp"
        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_alignParentEnd="true"
        android:layout_marginEnd="79dp"
        android:text="@string/defaultMethod" />

</RelativeLayout>
```

Exercise 2

- When a donation is accepted, set the amount on screen to 0 (in both picker and text field).



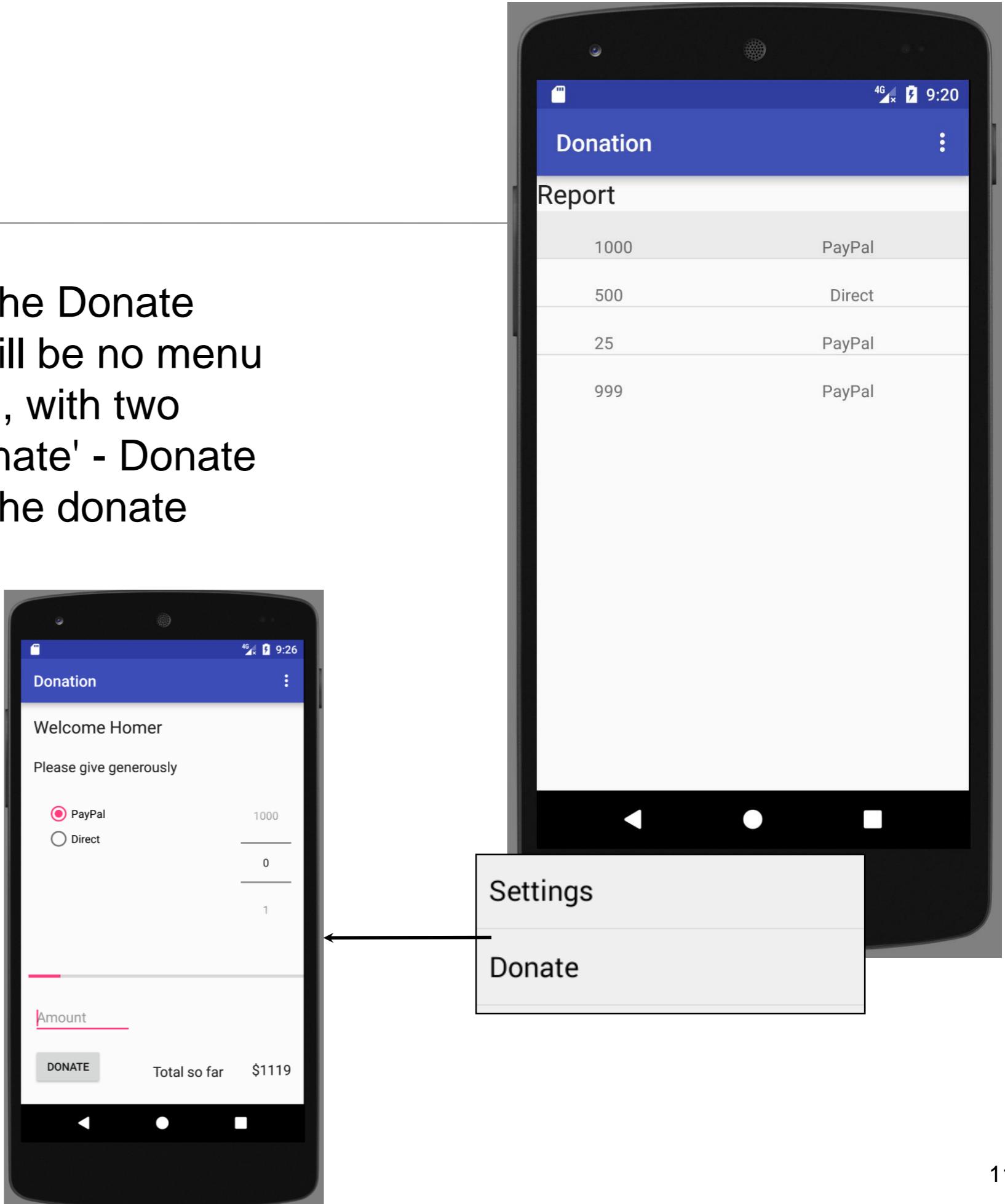
Exercise 2 Solution

- Add these two lines at the end of
Donate.donateButtonPressed()

```
amountText.setText("");  
amountPicker.setValue(0);
```

Exercise 3

- When you navigate from the Donate activity to reports, there will be no menu available. Bring in a menu, with two options 'Settings' and 'Donate' - Donate should bring you back to the donate screen.



Exercise 3 Solution

- Introduce a new string constant into strings.xml:
- A new menu in /res called menu_report.xml

```
<string name="menuDonate">Donate</string>
```

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

    <item
        android:id="@+id/menuSettings"
        android:title="@string/menuSettings"
        android:orderInCategory="100"/>
    <item
        android:id="@+id/menuDonate"
        android:title="@string/menuDonate"
        android:orderInCategory="100"/>

</menu>
```

Exercise 3 Solution

- Introduce a new string constant into strings.xml:
- A new menu in /res called menu_report.xml

orderInCategory:

Dictates the order of the items in the menu (ascending integer order). The lower the number, the higher the priority!

```
<string name="menuDonate">Donate</string>
```

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

    <item
        android:id="@+id/menuSettings"
        android:title="@string/menuSettings"
        android:orderInCategory="100"/>

    <item
        android:id="@+id/menuDonate"
        android:title="@string/menuDonate"
        android:orderInCategory="100"/>

</menu>
```

Exercise 3 Solution

```
public class Report extends AppCompatActivity {  
    private ListView listView;  
    private DonationApp app;  
  
    @Override  
    public void onCreate(Bundle savedInstanceState){  
        ...  
    }  
  
    @Override  
    public boolean onCreateOptionsMenu(Menu menu){  
        getMenuInflater().inflate(R.menu.menu_report, menu);  
        return true;  
    }  
  
    @Override  
    public boolean onOptionsItemSelected(MenuItem item){  
        switch (item.getItemId()){  
            case R.id.menuDonate : startActivity (new Intent(this, Donate.class));  
                break;  
            case R.id.menuSettings : Toast.makeText(this, "Settings Selected", Toast.LENGTH_SHORT).show();  
                break;  
        }  
        return true;  
    }  
}
```

The report activity will then need two new methods...

Exercise 4

- Introduce a new welcome screen - which should display a greeting + give the user 2 options (as simple buttons)
 - Login
 - Signup
- When either button is pressed, the app should take you directly to the Donate activity (for the moment).



strings.xml

```
<string name="welcomeLogin">Login</string>
<string name="welcomeSignup">Sign up</string>
```

Exercise 4 Solution

activity_welcome.xml

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

    <Button
        android:id="@+id/welcomeLogin"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="180dp"
        android:text="@string/welcomeLogin"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/welcomeSignup"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="32dp"
        android:text="@string/welcomeSignup"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/welcomeLogin" />
</android.support.constraint.ConstraintLayout>
```



Exercise 4 Solution

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

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

        <activity android:name="app.donation.Welcome"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity android:name="app.donation.Report"
            android:label="@string/app_name">
        </activity>

        <activity android:name="app.donation.Donate"
            android:label="@string/app_name">
        </activity>
    </application>
</manifest>
```

AndroidManifest.xml

```
public class Welcome extends AppCompatActivity
{
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_welcome);
    }
}
```

Exercise 4 Solution

activity_welcome.xml

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

    <Button
        android:id="@+id/welcomeLogin"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="180dp"
        android:text="@string/welcomeLogin"
        android:onClick="loginPressed"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />

    <Button
        android:id="@+id/welcomeSignup"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginTop="32dp"
        android:text="@string/welcomeSignup"
        android:onClick="signupPressed"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/welcomeLogin" />
</android.support.constraint.ConstraintLayout>
```

```
public class Welcome extends Activity {

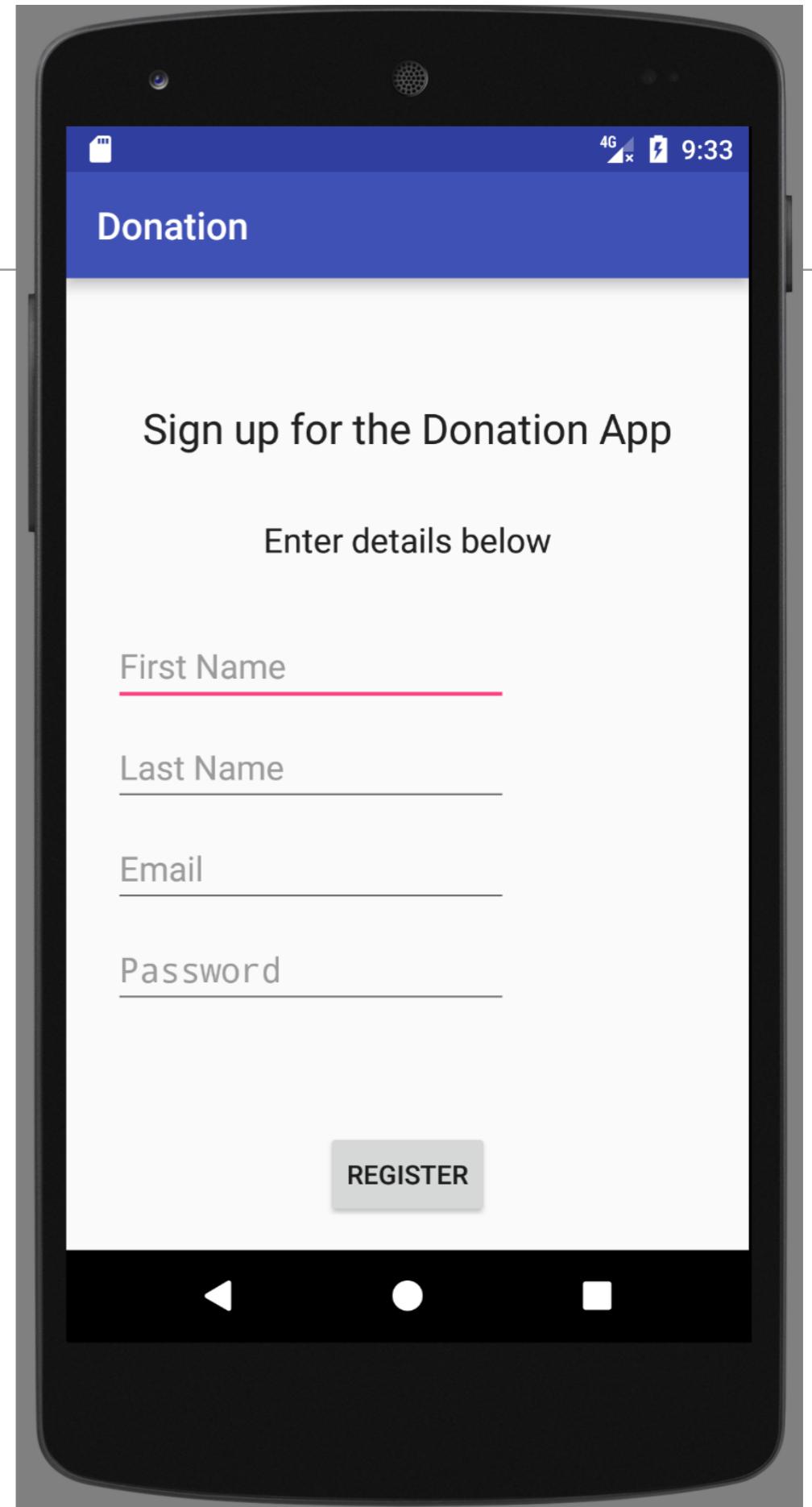
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_welcome);
    }

    public void loginPressed(View view) {
        startActivity(new Intent(this, Donate.class));
    }

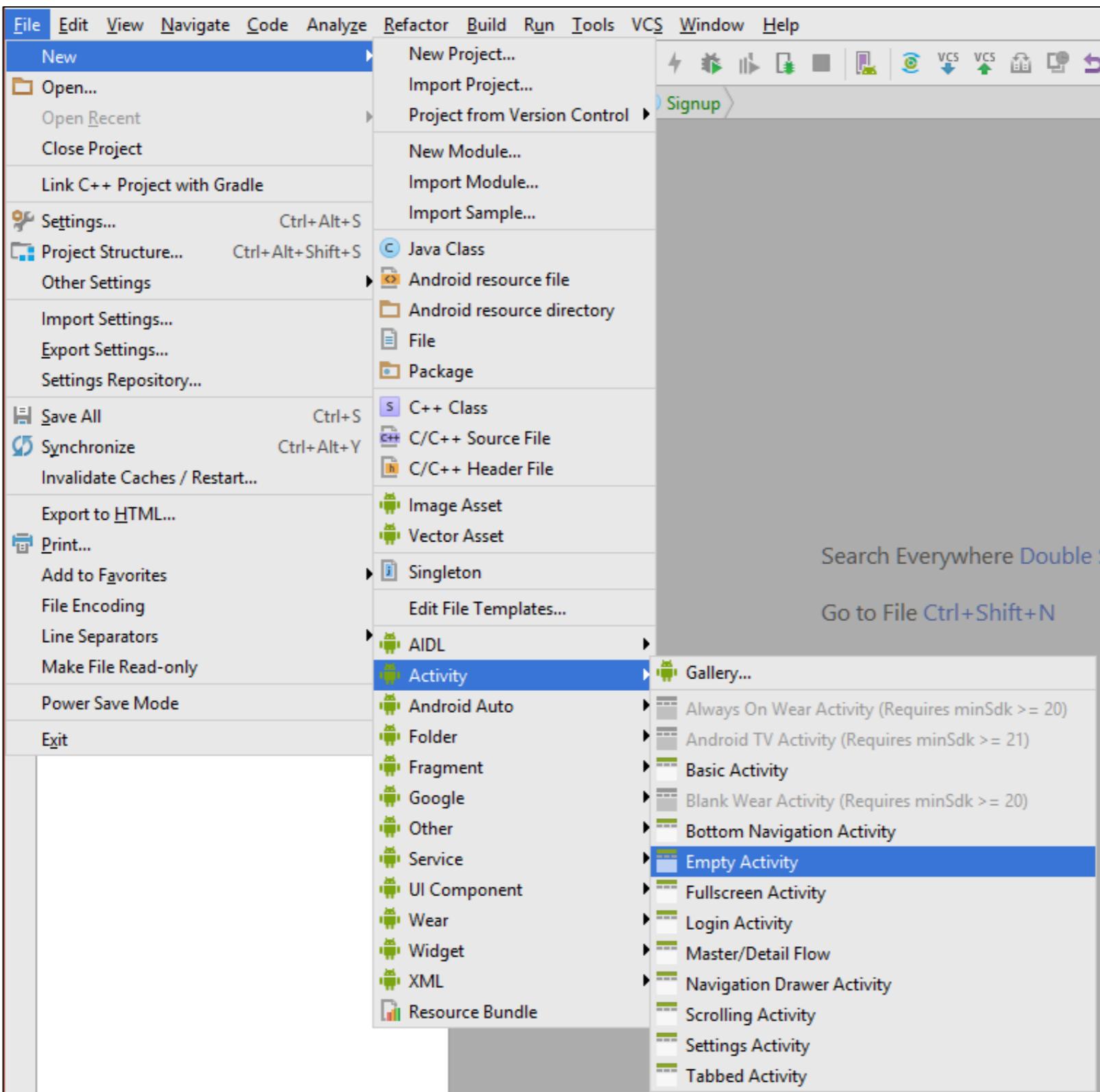
    public void signupPressed(View view) {
        startActivity(new Intent(this, Donate.class));
    }
}
```

Exercise 5

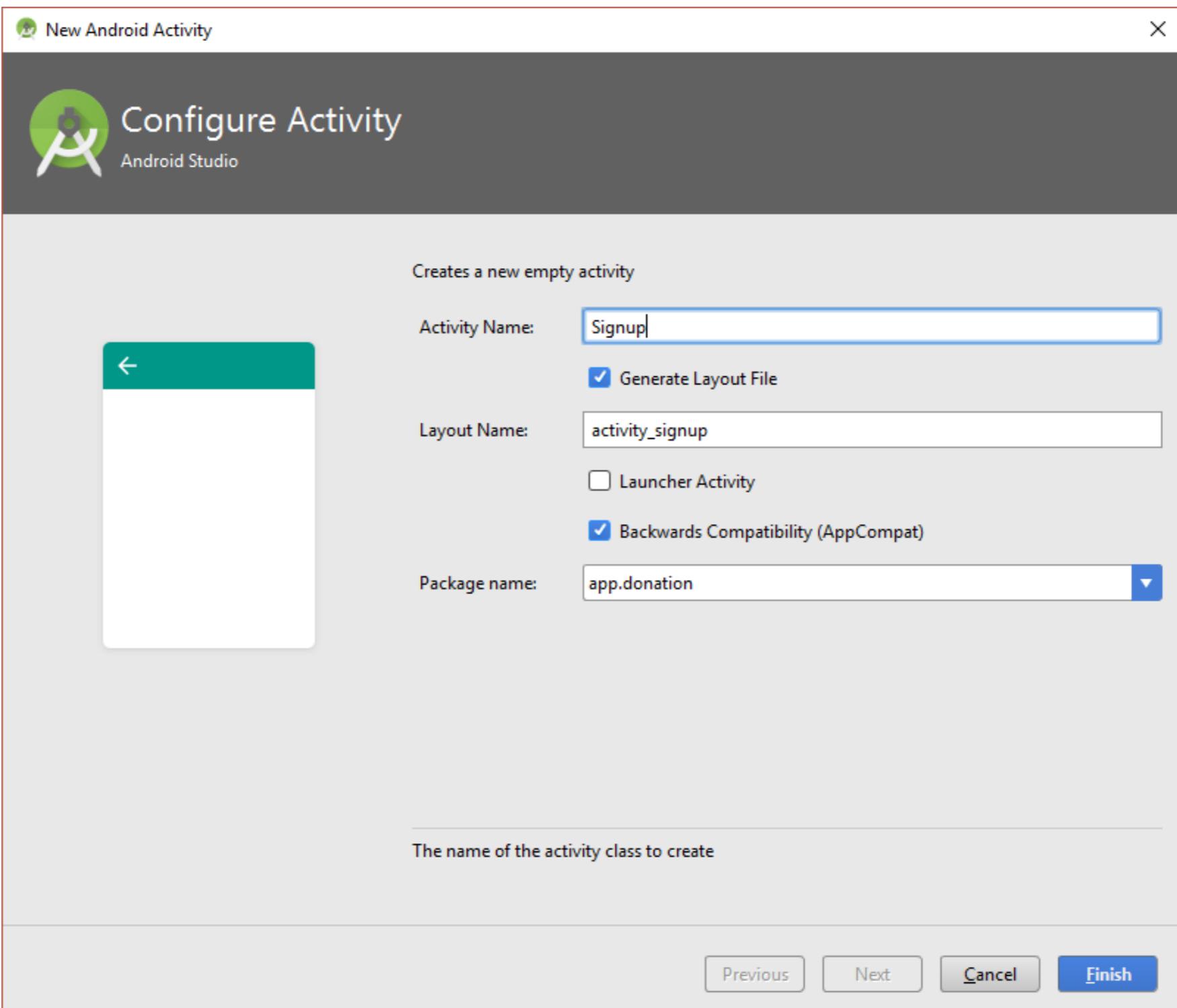
- Introduce a Signup Activity, which should present the user with:
 - First Name
 - Last Name
 - Email
 - Password
 - + 'Register' button.
- Pressing Register should take you directly to "Donate" activity



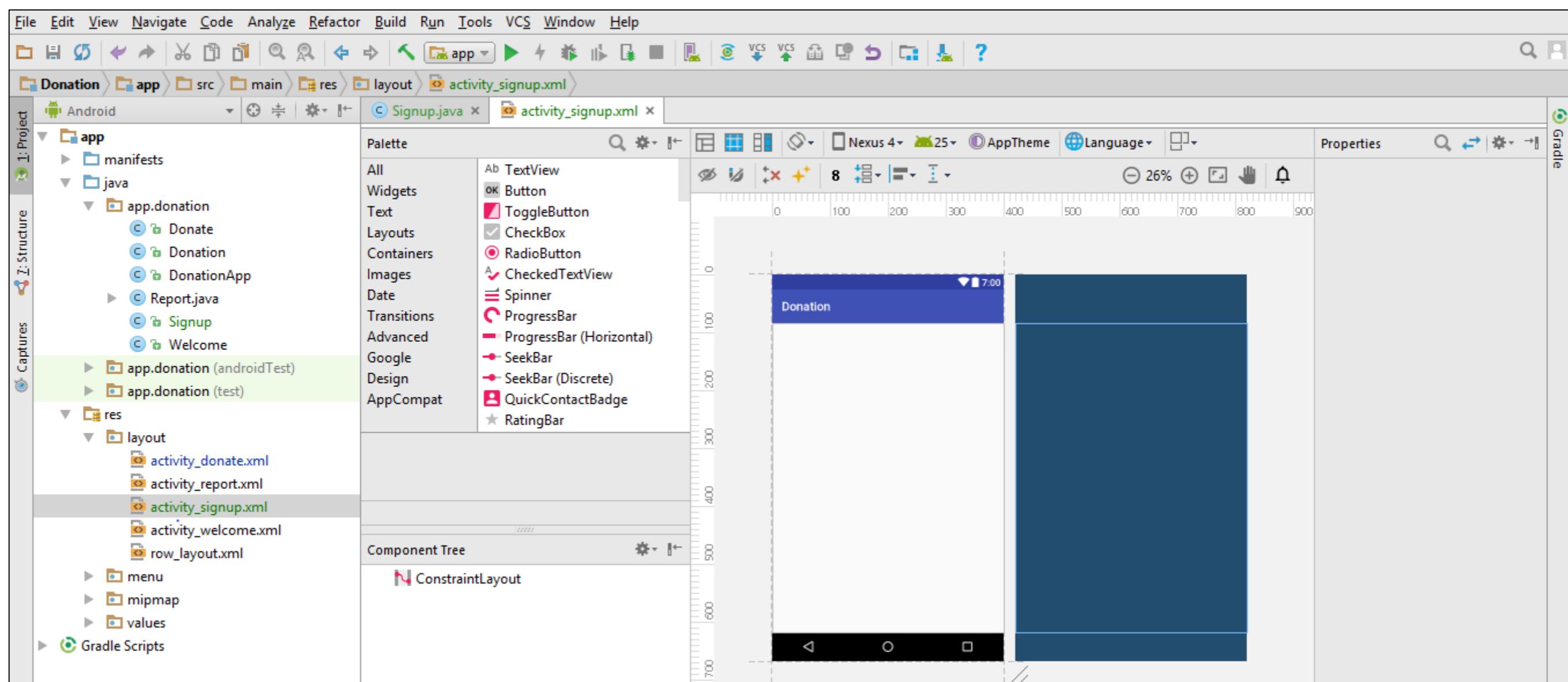
Exercise 5 Solution



Exercise 5 Solution

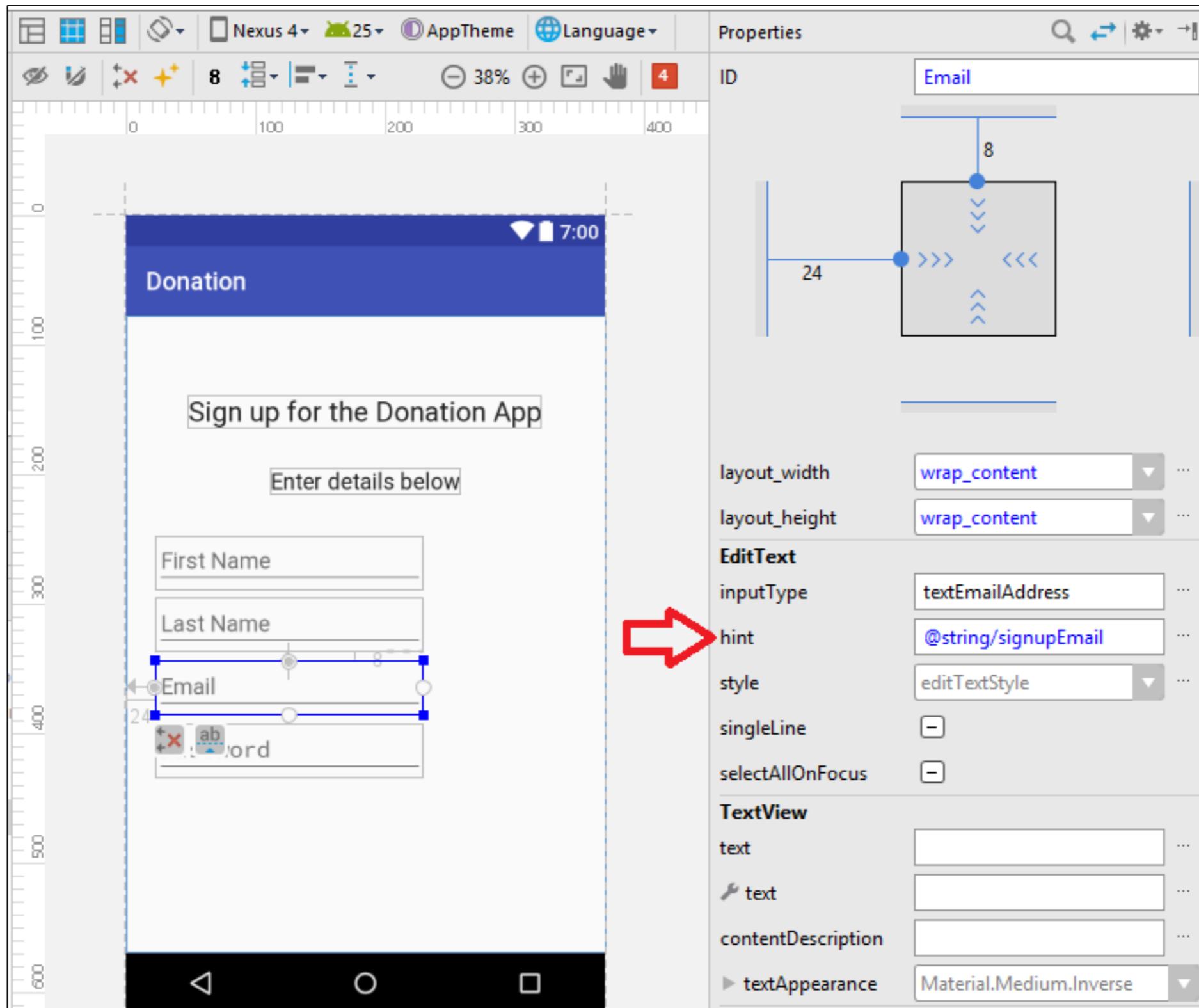


Exercise 5 Solution



Exercise 5 Solution

Add a hint (placeholder)



Exercise 5 Solution

strings.xml

```
<string name="signupTitle">Sign up for the Donation App</string>
<string name="signupSubtitle">Enter details below</string>
<string name="signupFirstname">First name</string>
<string name="signupLastName">Last Name</string>
<string name="signupEmail">Email</string>
<string name="signupPassword">Password</string>
<string name="signupRegister">Register</string>
```

```
public class Signup extends AppCompatActivity
{
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_signup);
    }

    public void signupPressed (View view)
    {
        startActivity (new Intent(this, Donate.class));
    }
}
```

```
<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=".Signup" >
```

```
<TextView
    android:id="@+id/signupTitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"
    android:layout_marginLeft="32dp"
    android:layout_marginTop="28dp"
    android:text="@string/signupTitle"
    android:textAppearance="?android:attr/textAppearanceMedium" />

<TextView
    android:id="@+id/signupSubtitle"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/signupTitle"
    android:layout_below="@+id/signupTitle"
    android:layout_marginLeft="55dp"
    android:layout_marginTop="30dp"
    android:text="@string/signupSubtitle"
    android:textAppearance="?android:attr/textAppearanceSmall" />
```

```
<EditText
    android:id="@+id/firstName"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignParentRight="true"
    android:layout_below="@+id/signupSubtitle"
    android:layout_marginTop="40dp"
    android:ems="10"
    android:hint="@string/signupFirstname"
    android:inputType="textPersonName" />
```

```
<requestFocus />
```

```
<EditText
    android:id="@+id/lastName"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/firstName"
    android:layout_alignParentRight="true"
    android:layout_below="@+id/firstName"
    android:ems="10"
    android:hint="@string/signupLastName"
    android:inputType="textPersonName" />
```

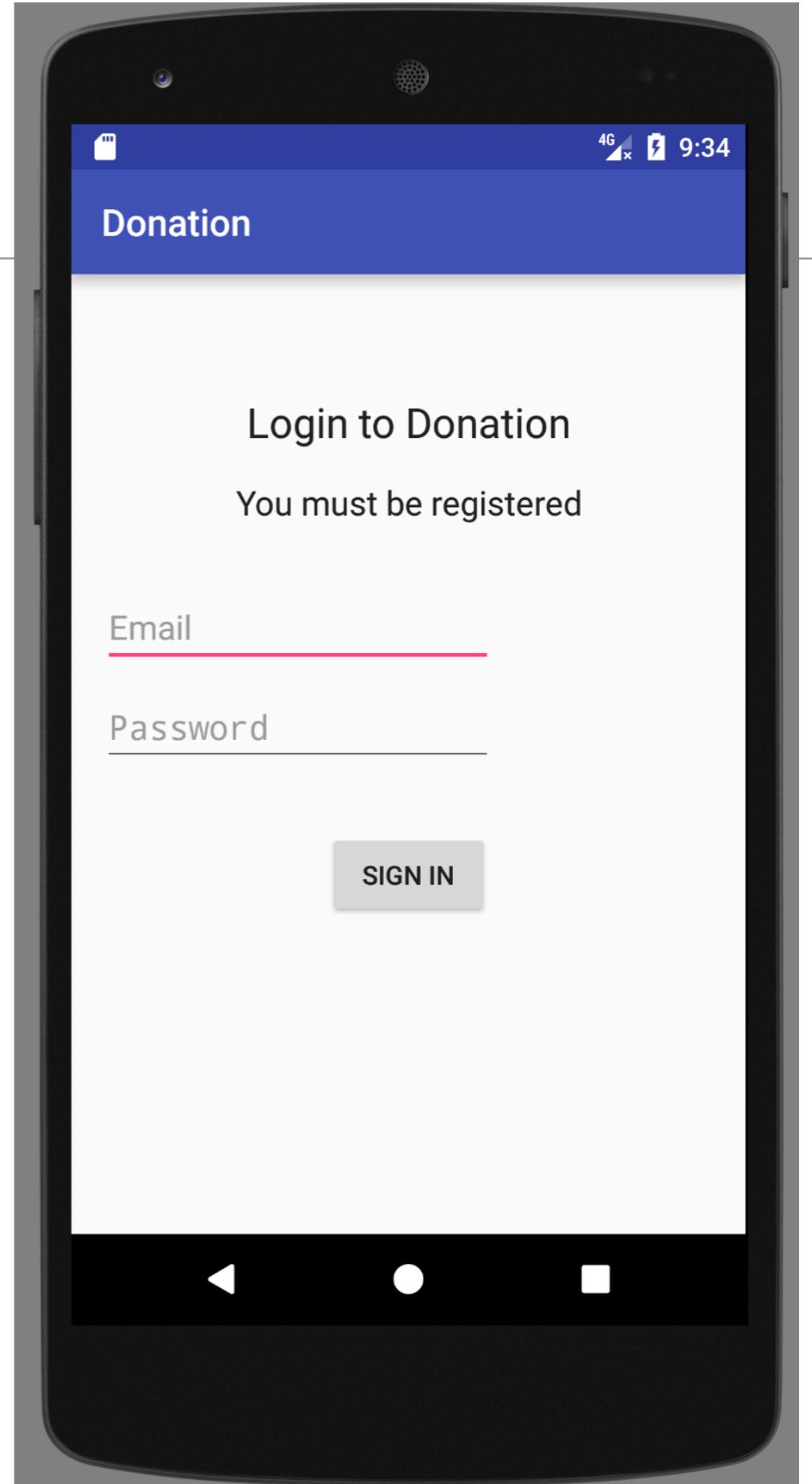
```
</EditText>
```

```
<EditText
    android:id="@+id>Email"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/lastName"
    android:layout_alignParentRight="true"
    android:layout_below="@+id/lastName"
    android:ems="10"
    android:hint="@string/signupEmail"
    android:inputType="textEmailAddress" />
```

activity_signup.xml

Exercise 6

- Introduce a Login activity, which should just look for
 - email
 - password
 - + a 'Sign in' button
- Pressing Sign in should take you directly to "Donate" activity.



```

<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=".Login" >

    <TextView
        android:id="@+id/loginTitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="18dp"
        android:text="@string/loginTitle"
        android:textAppearance="?android:attr/textAppearanceMedium" />

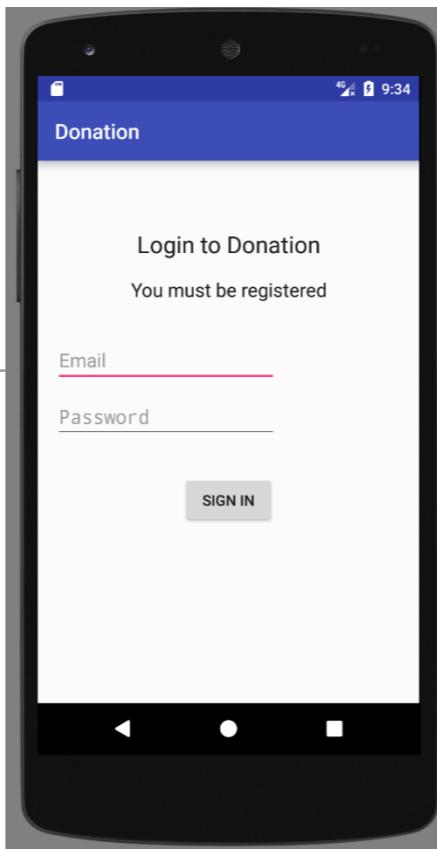
    <TextView
        android:id="@+id/loginSubtitle"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/loginTitle"
        android:layout_alignParentRight="true"
        android:layout_below="@+id/loginTitle"
        android:text="@string/loginSubtitle"
        android:textAppearance="?android:attr/textAppearanceSmall" />

    <EditText
        android:id="@+id/loginEmail"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/loginSubtitle"
        android:layout_alignRight="@+id/loginSubtitle"
        android:layout_below="@+id/loginSubtitle"
        android:layout_marginTop="17dp"
        android:ems="10"
        android:hint="@string/loginEmail"
        android:inputType="textEmailAddress" >
        <requestFocus />
    </EditText>

    <EditText
        android:id="@+id/loginPassword"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/loginEmail"
        android:layout_alignRight="@+id/loginEmail"
        android:layout_below="@+id/loginEmail"
        android:layout_centerHorizontal="true"
        android:ems="10"
        android:hint="@string/loginPassword"
        android:inputType="textPassword" />

    <Button
        android:id="@+id/login"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:onClick="signinPressed"
        android:text="@string/loginSignin" />

```



Exercise 6 Solution

```

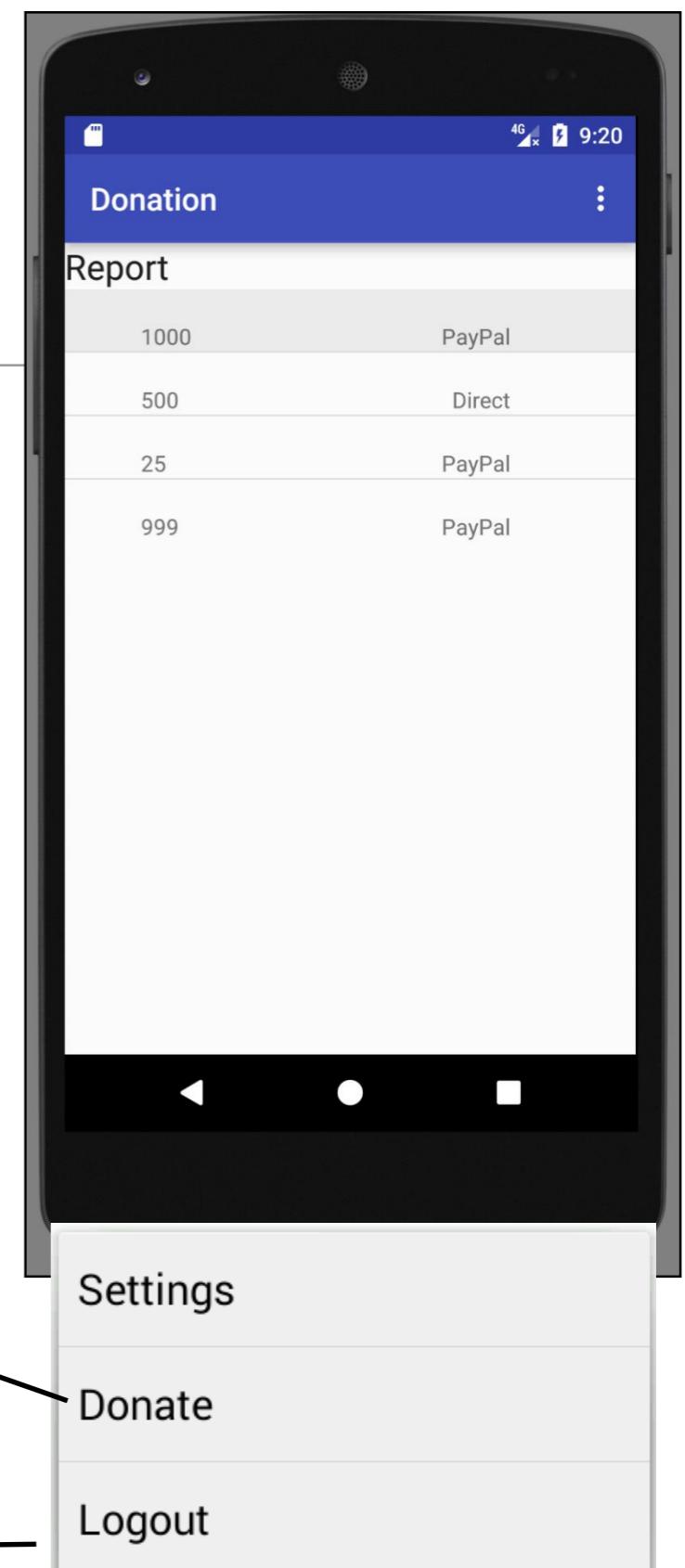
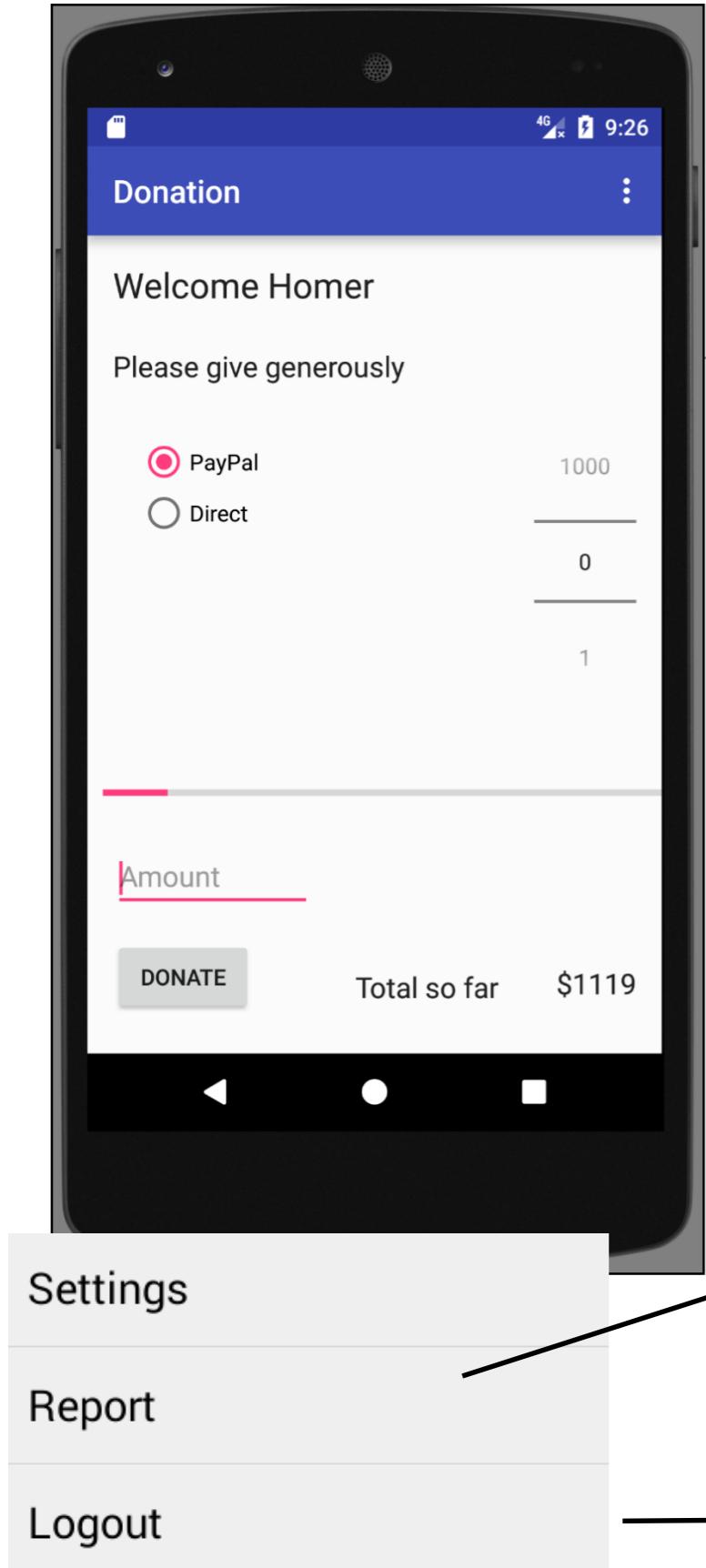
public class Login extends AppCompatActivity
{
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
    }

    public void signinPressed (View view)
    {
        startActivity (new Intent(this, Donate.class));
    }
}

```

Exercise 7

- Bring in a new menu option - 'logout'. It should take you to the welcome screen.



Exercise 7 Solution

strings.xml

```
<string name="menuLogout">Logout</string>
```

donate.xml

&

report.xml

```
<item  
    android:id="@+id/menuLogout"  
    android:orderInCategory="100"  
    android:title="@string/menuLogout"/>
```

Donate

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

Report

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

Exercise 8

- Introduce a 'User' into the models package to represent the user in the usual way.
- Maintain a list of Users in the DonationApp object.
- Whenever anyone registers, then create a new User object in this list.

Exercise 8 Solution

```
public class User
{
    public String firstName;
    public String lastName;
    public String email;
    public String password;

    public User(String firstName, String lastName, String email, String password)
    {
        this.firstName = firstName;
        this.lastName = lastName;
        this.email = email;
        this.password = password;
    }
}
```

Exercise 8 Solution

```
public class DonationApp extends Application
{
    public final int    target    = 10000;
    public int          totalDonated = 0;

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

    public void newUser(User user)
    {
        users.add(user);
    }

    //...
}
```

Exercise 8 Solution

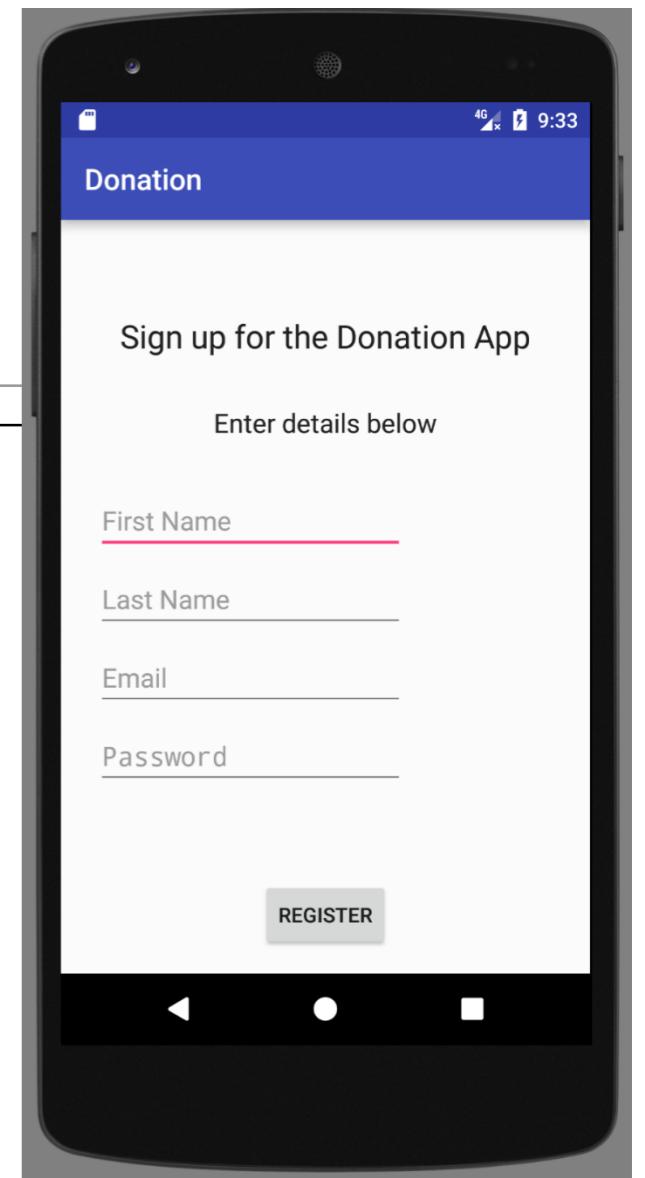
```
public class Signup extends AppCompatActivity
{
//.....

    public void signupPressed (View view)
    {
        TextView firstName = (TextView) findViewById(R.id.firstName);
        TextView lastName = (TextView) findViewById(R.id.lastName);
        TextView email   = (TextView) findViewById(R.id.Email);
        TextView password = (TextView) findViewById(R.id.Password);

        User user = new User (firstName.getText().toString(), lastName.getText().toString(),
                           email.getText().toString(), password.getText().toString());

        DonationApp app = (DonationApp) getApplication();
        app.newUser(user);

        startActivity (new Intent(this, Welcome.class));
    }
}
```



Exercise 9

- Implement the Login activity, to now only let users in to Donate if they are registered (i.e. a matching email + password in the list of users maintained by DonationApp)

Exercise 9 Solution

```
public class DonationApp extends Application
{
    public final int target = 10000;
    public int totalDonated = 0;

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

    public void newUser(User user){
        users.add(user);
    }

    public boolean validUser (String email, String password){
        for (User user : users){
            if (user.email.equals(email) && user.password.equals(password))
            {
                return true;
            }
        }
        return false;
    }

    //...
}
```

Exercise 9

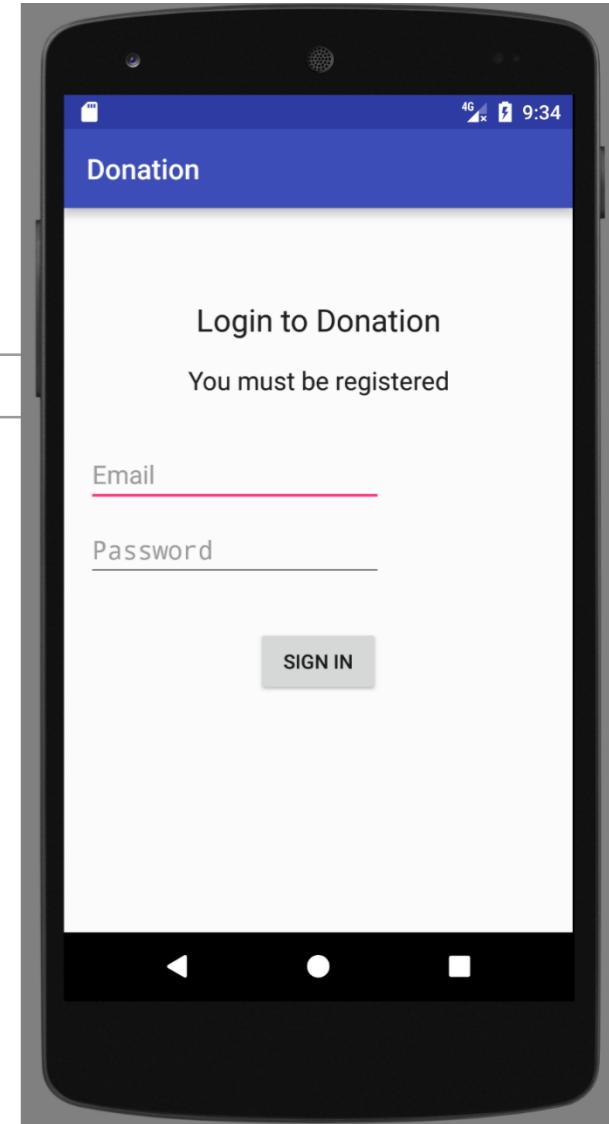
Solution

```
public class Login extends AppCompatActivity
{
    //...

    public void signinPressed (View view) {
        DonationApp app = (DonationApp) getApplication();

        TextView email    = (TextView) findViewById(R.id.loginEmail);
        TextView password = (TextView) findViewById(R.id.loginPassword);

        if (app.validUser(email.getText().toString(), password.getText().toString())){
            startActivity (new Intent(this, Donate.class));
        }
        else{
            Toast toast = Toast.makeText(this, "Invalid Credentials", Toast.LENGTH_SHORT);
            toast.show();
        }
    }
}
```

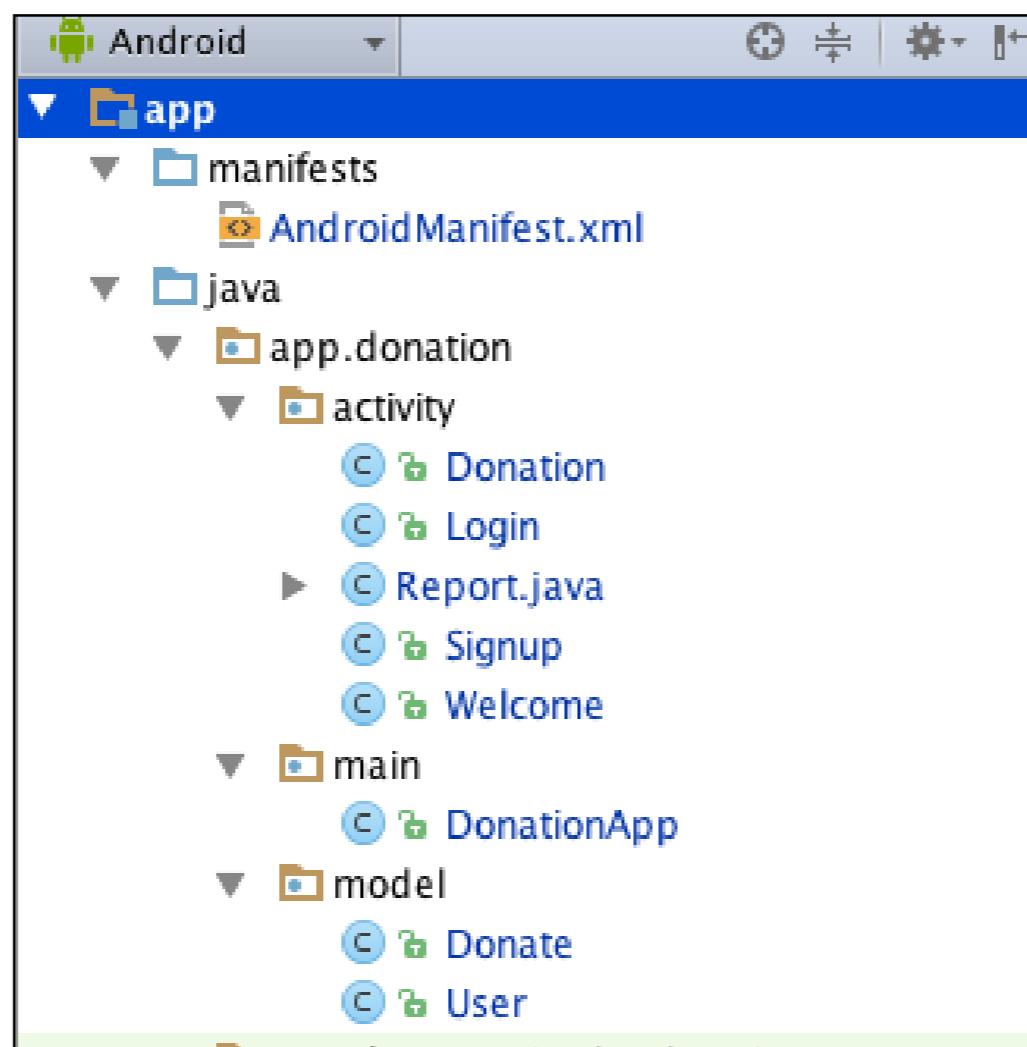
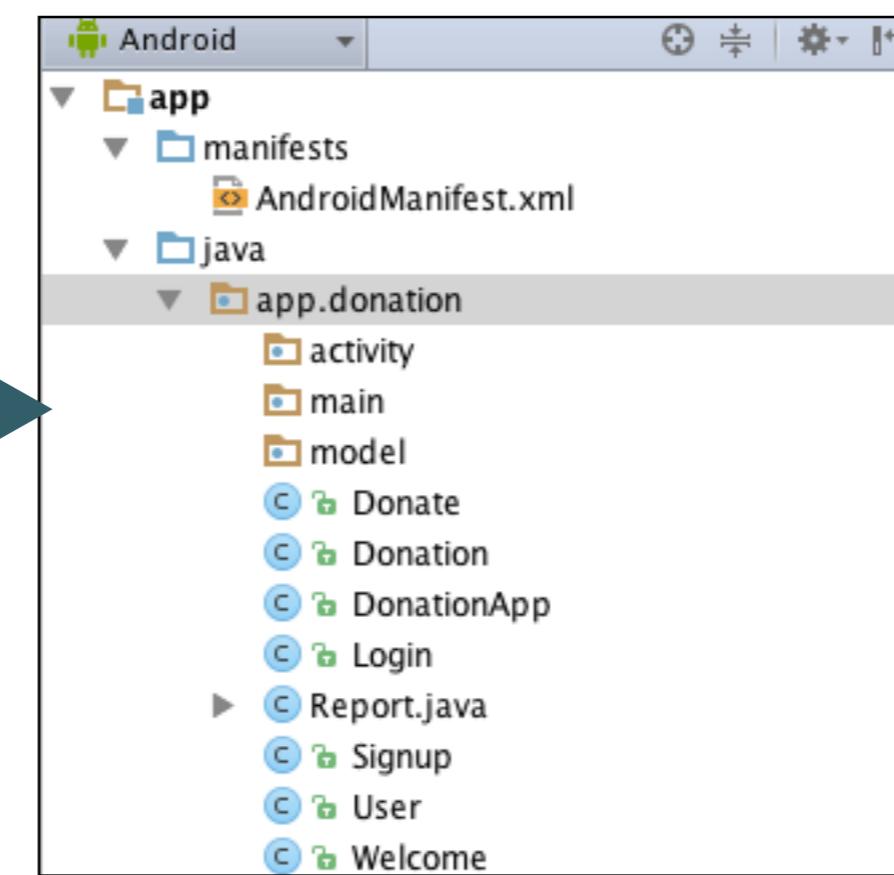
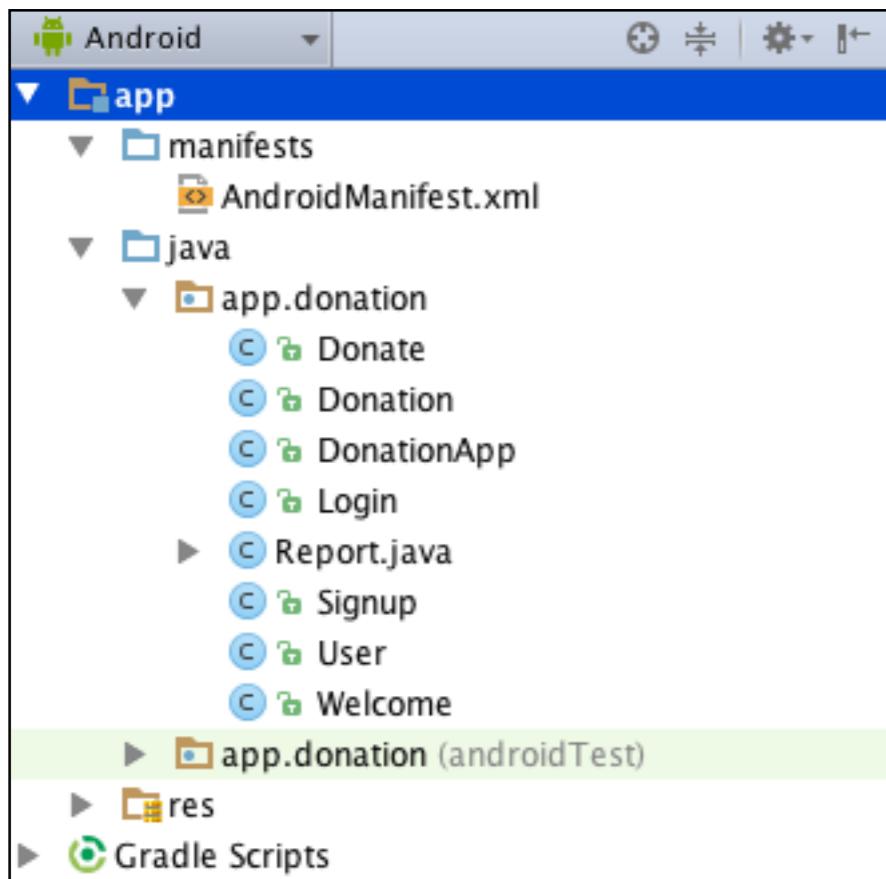


Exercise 10

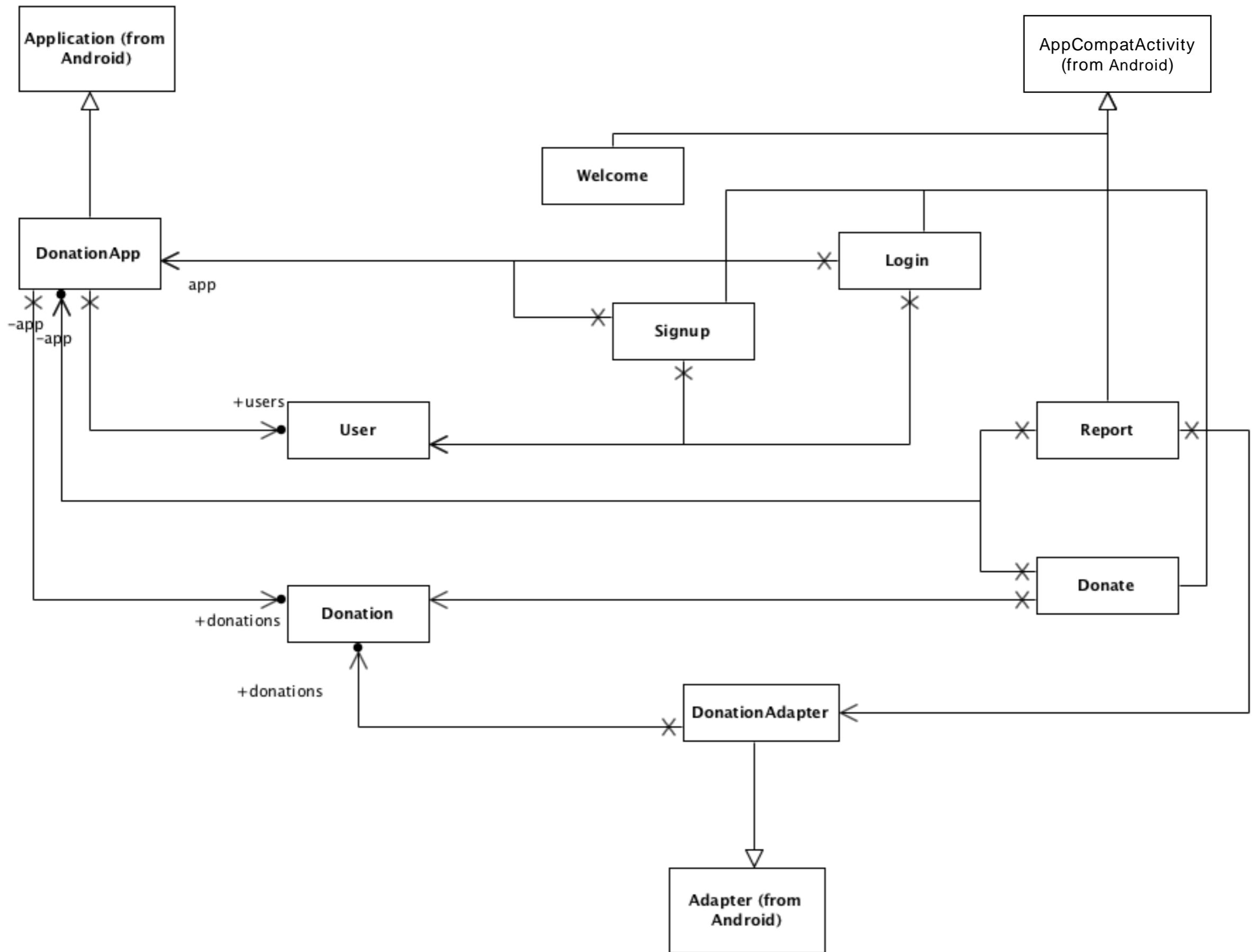
- If you have any "RelativeLayout" layouts, convert them to "ConstraintLayout". There is a conversion function in Android Studio to do this for you.

Exercise 10 - Solution

- To check if your layouts use RelativeLayout, open the xml and check the layout either in Text view or Design view (in the Component Tree panel).
- If your layout is Relative, in Design view, right click on the RelativeLayout in your Component Tree panel and select the option to Convert to ConstraintLayout. When the window appears, accept the defaults.
- Your layout will now be a ConstraintLayout. View your design in the BluePrint and make any changes to the layout you see fit (you might not need to make any).
- Run your app to make sure all is ok.

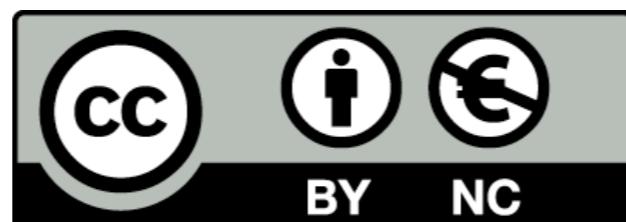


Refactor Package
Structure



Questions?





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

 eLearning
support unit