

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 - v3

Donation Android v1

 Donation

Welcome Homer
Please give generously

☒ PayPal
☐ Direct


+

0

-

Button

Donation Android v2

 Donation

Donation App

Please give generously

☒ PayPal

☐ Direct

1000


0

1

Amount:

Donate

Total so far: 0

 Donation App

Report


3	PayPal
5	PayPal
10	PayPal
100	PayPal

Donation Android v2 - Exercises

- 9 Exercises
- Allocate 2 days during midterm?
- Worked example of each exercise solution to be presented after mid-term

Exercises 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.

 Donation App	
Report	
3	PayPal
5	PayPal
10	PayPal
100	PayPal

Exercise 1 Solution

```
<?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>
```

Exercise 2

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

 Donation

Donation App

Please give generously

☒ PayPal

1000

☐ Direct

0

1

Amount:

Donate

Total so far: 0


Exercise 2 Solution

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

```
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.

 Donation

Donation App

Please give generously

☒ PayPal

☐ Direct

1000


0

1

Amount:

Donate

Total so far: 0

 Donation App

Report

3	PayPal
5	PayPal
10	PayPal
100	PayPal

Settings

Donate

Exercise 3 Solution

- Introduce a new string constant into strings.xml:
- Then a new menu - report.xml

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

```
<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/menuDonate"
        android:orderInCategory="100"
        android:showAsAction="never"
        android:title="@string/menuDonate"/>

</menu>
```

Exercise 3 Solution

- The report activity will then need two new methods:

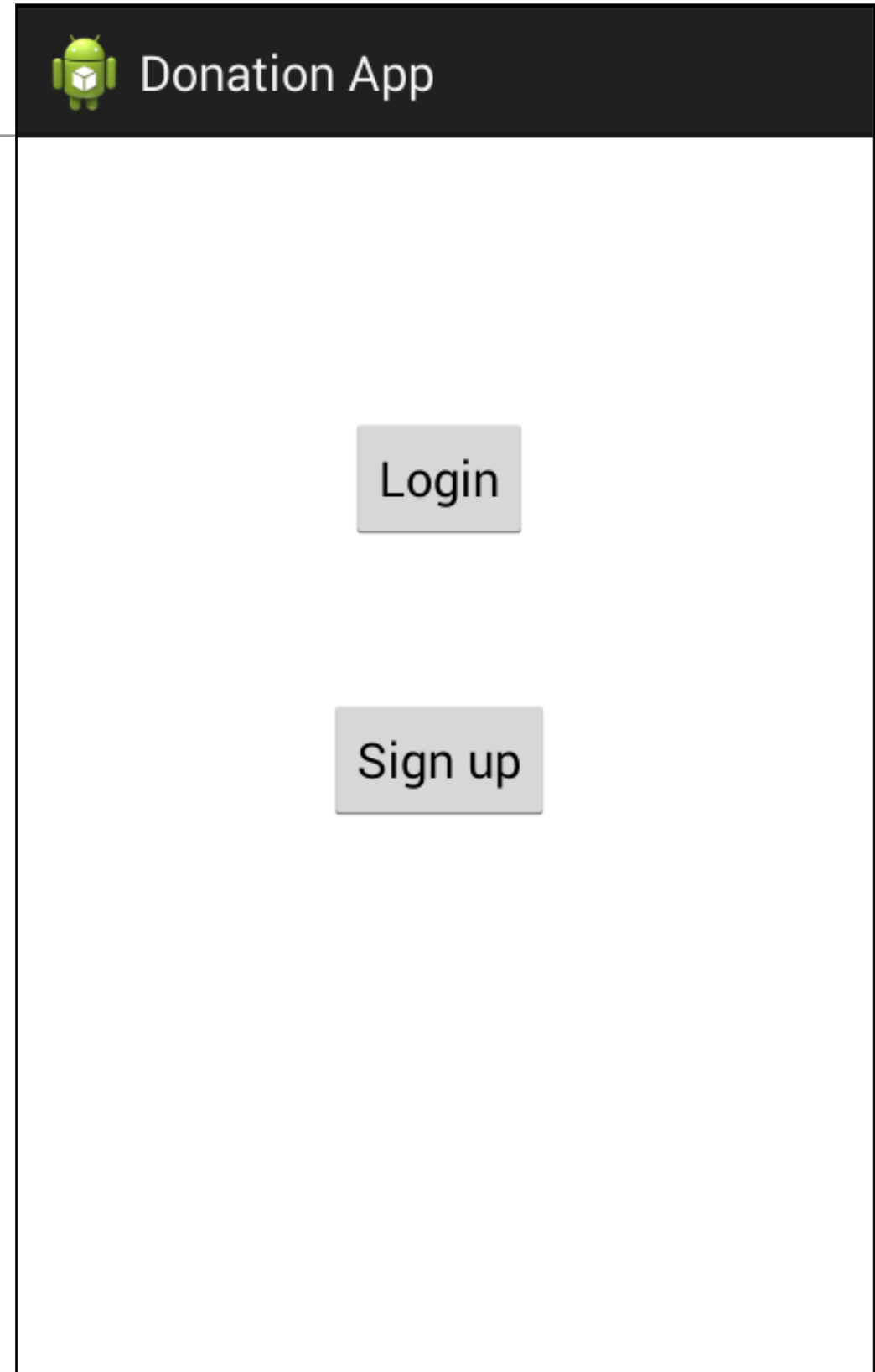
```
public class Report extends Activity
{
    private ListView    listView;
    private DonationApp app;

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

Exercise 4

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



- strings.xml

Exercise 4 Solution

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

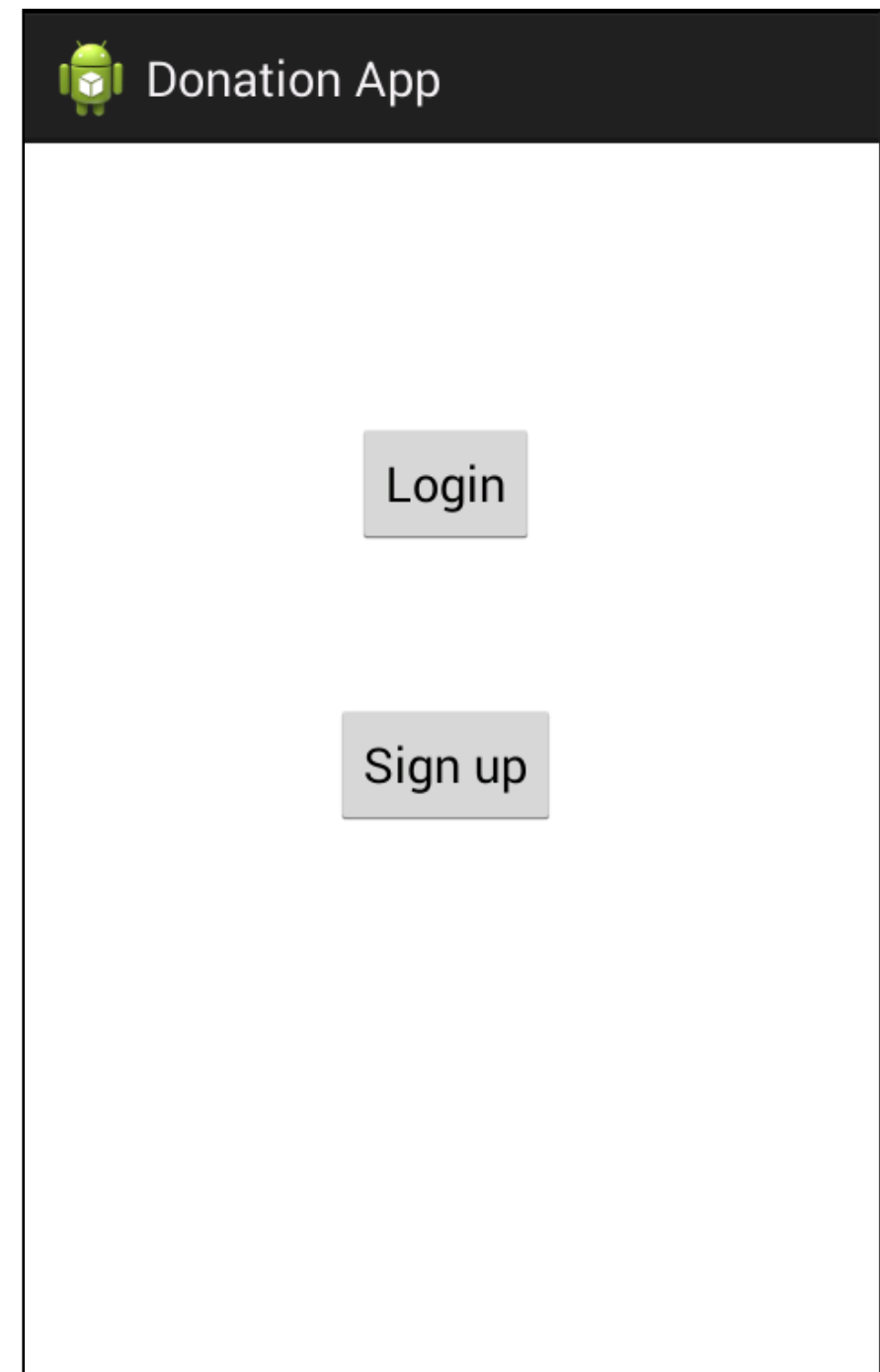
- welcome_layout.xml

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

    <Button
        android:id="@+id/welcomeLogin"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="102dp"
        android:text="@string/welcomeLogin" />

    <Button
        android:id="@+id/welcomeSignup"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_centerVertical="true"
        android:text="@string/welcomeSignup" />

</RelativeLayout>
```



AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="app.donation"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="17"
        android:targetSdkVersion="17" />

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

        <activity
            android:name="app.activities.Welcome"
            android:label="@string/donateTitle" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity
            android:name="app.activities.Donate"
            android:label="@string/app_name" >
        </activity>
        <activity
            android:name="app.activities.Report"
            android:label="@string/donateTitle" >
        </activity>
    </application>

</manifest>
```

Exercise 4 Solution

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

```

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

    <Button
        android:id="@+id/welcomeLogin"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="102dp"
        android:onClick="loginPressed"
        android:text="@string/welcomeLogin" />

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

</RelativeLayout>

```

```

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

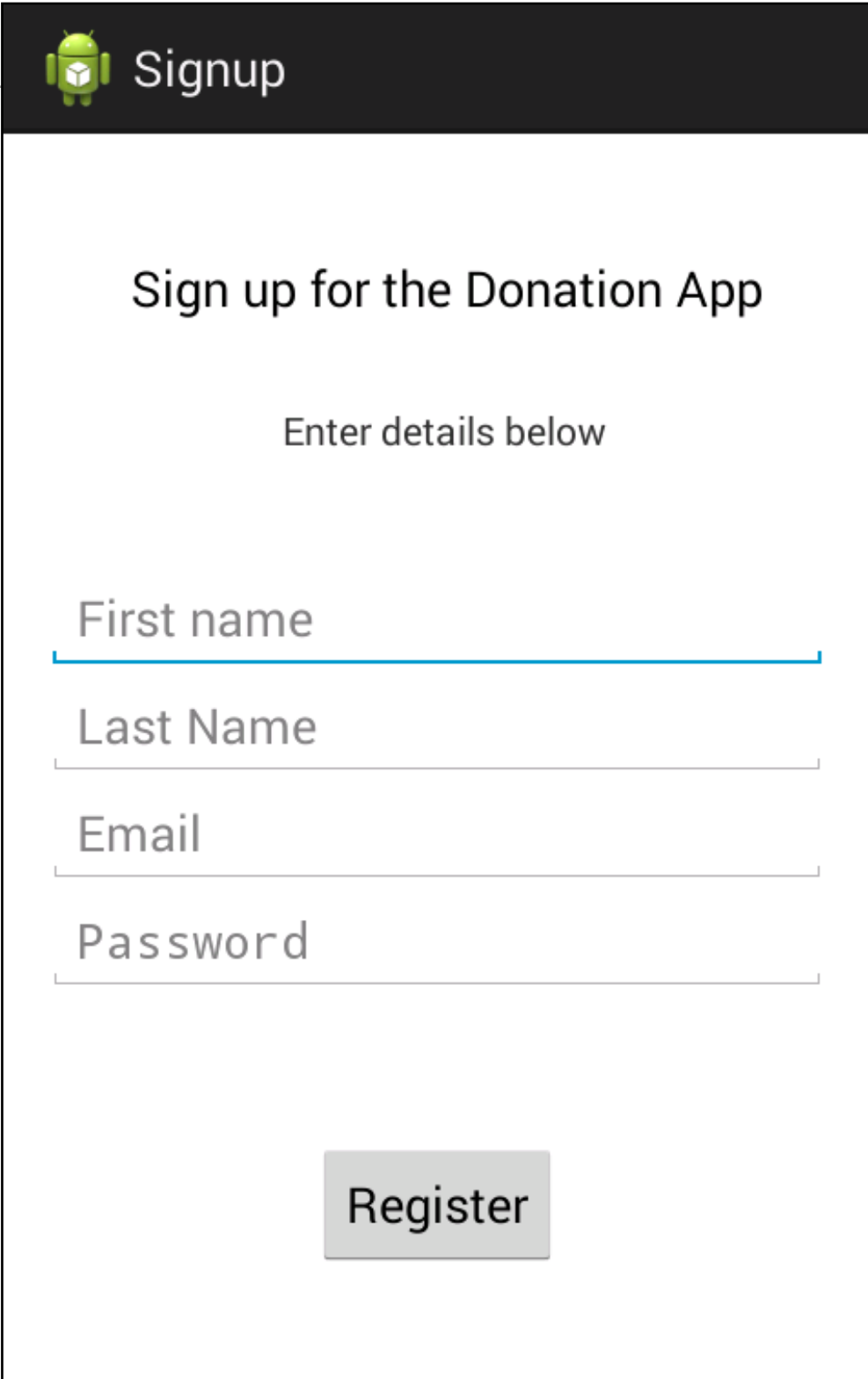
    public void loginPressed (View view)
    {
        startActivity (new Intent(this, Login.class));
    }

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

```

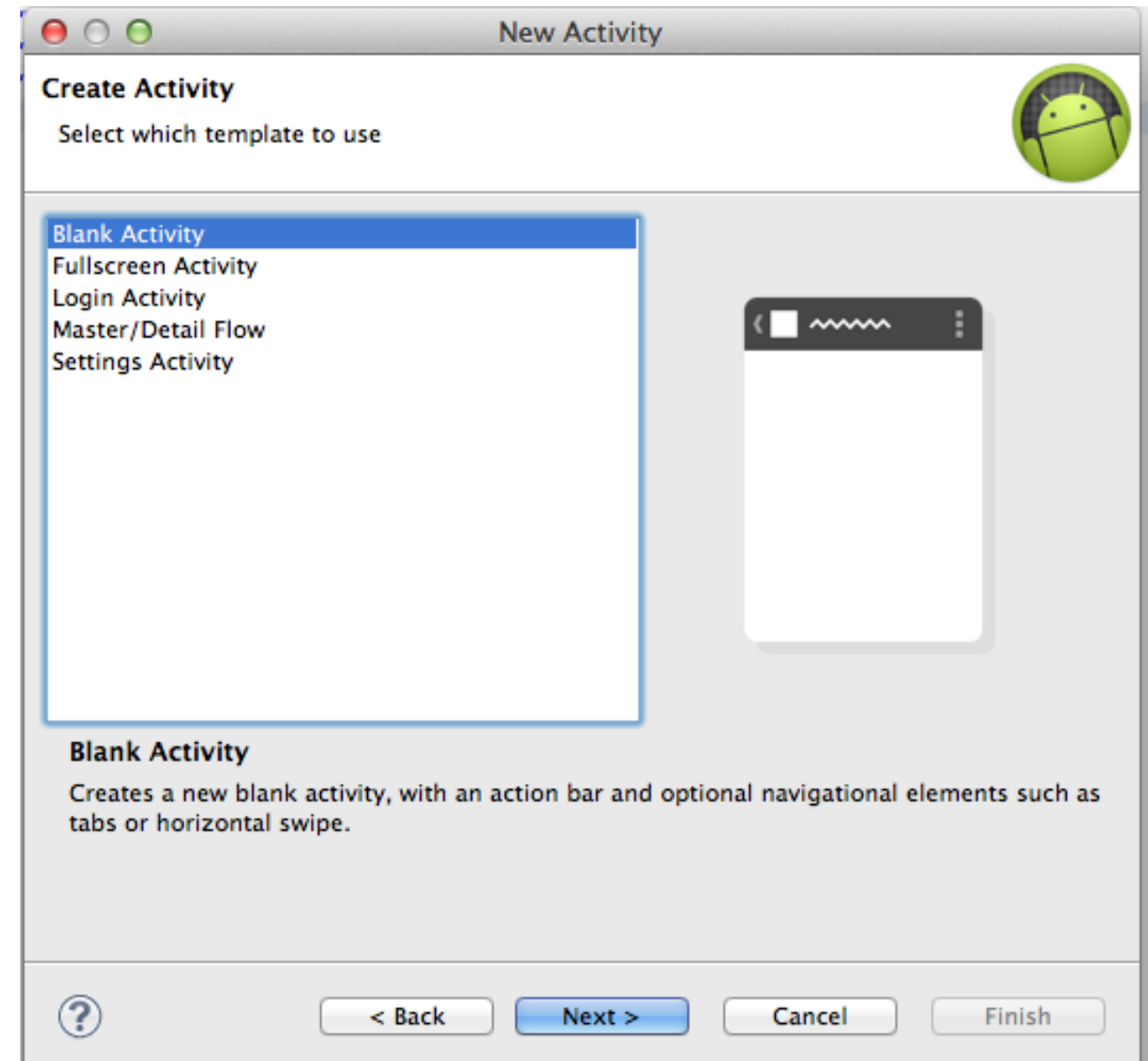
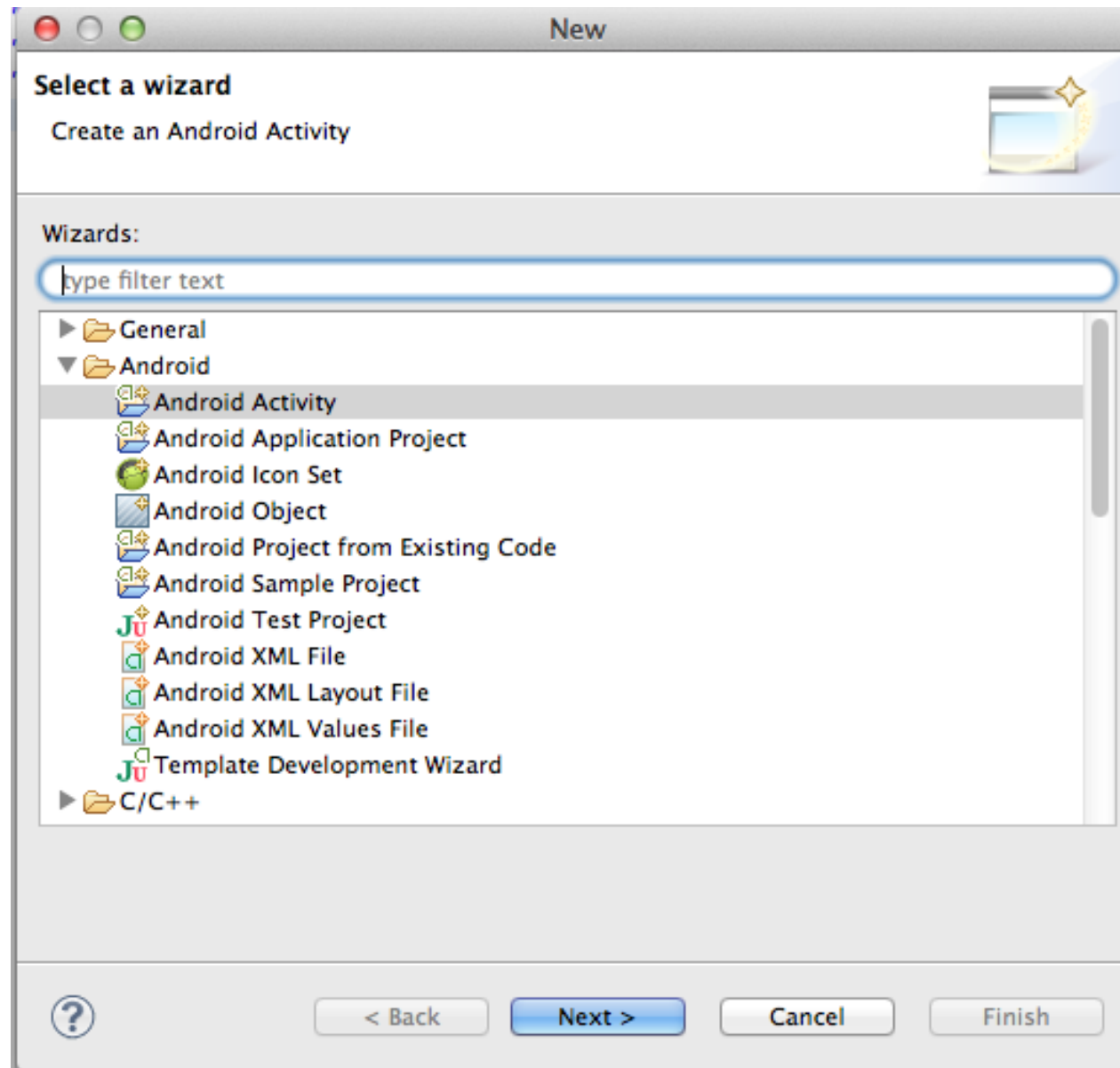

Exercise 5

- 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



The screenshot shows a mobile application interface for a 'Signup' activity. At the top, there is a dark header bar with a green Android robot icon and the text 'Signup'. Below the header, the main content area is white. It features the title 'Sign up for the Donation App' in a bold, black font, followed by the instruction 'Enter details below' in a smaller, gray font. There are four input fields stacked vertically, each with a light gray border and a placeholder text: 'First name', 'Last Name', 'Email', and 'Password'. The 'First name' field is highlighted with a blue underline. At the bottom of the form, there is a gray rectangular button with the text 'Register' in black.

Exercise 5 Solution



New Activity

Blank Activity

Creates a new blank activity, with an action bar and optional navigational elements such as tabs or horizontal swipe.

Project: donation-android

Activity Name *i* Signup

Layout Name *i* activity_signup

Title *i* Signup

i ☐ Launcher Activity

Hierarchical Parent *i* Optional

Navigation Type *i* None

? The name of the activity class to create

? < Back Next

New Activity

Preview

Optionally review pending changes

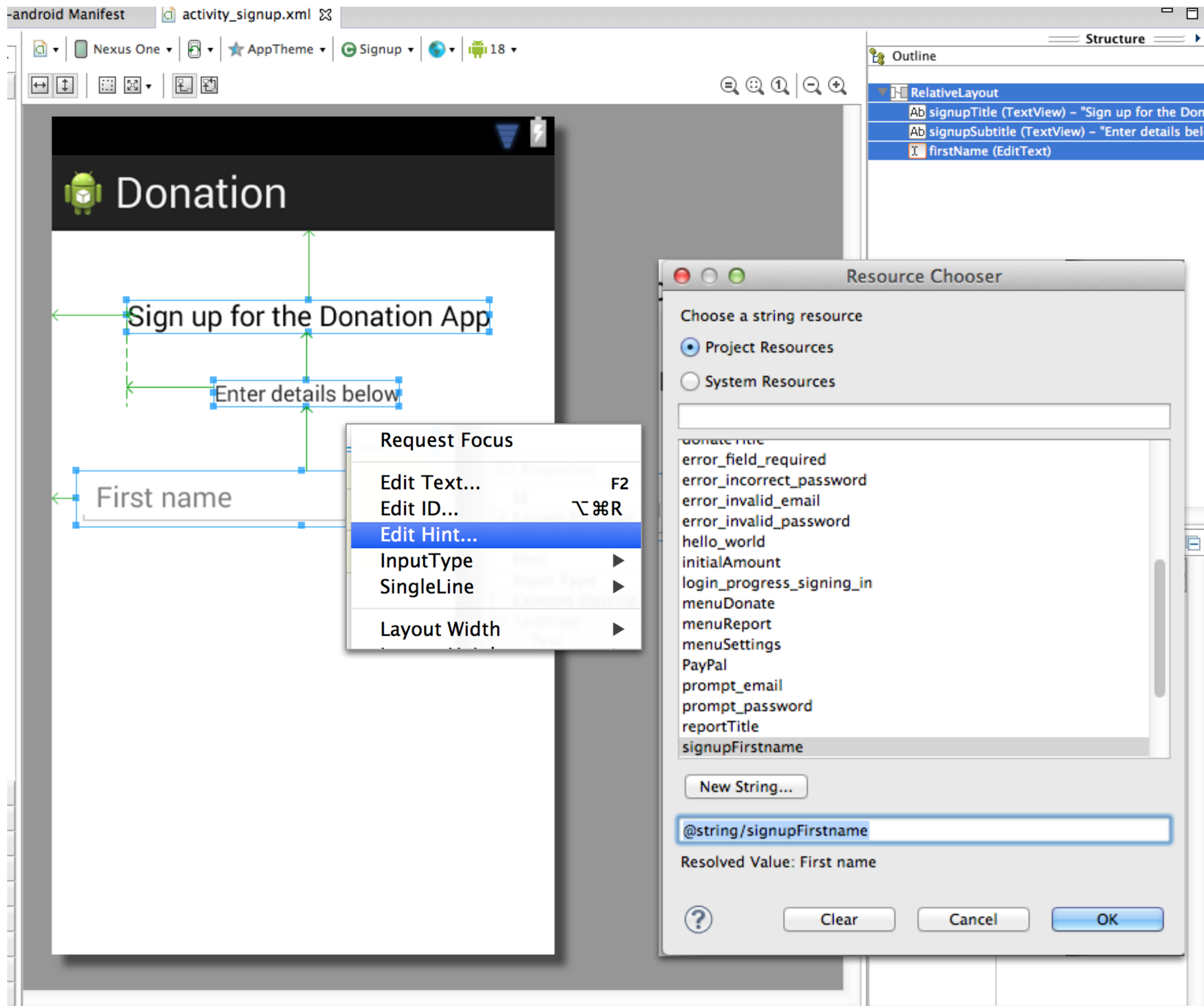
Changes to be performed

- ☒ AndroidManifest.xml - donation-android
- ☒ strings.xml - donation-android/res/values
- ☒ signup.xml - donation-android/res/menu
- ☒ activity_signup.xml - donation-android/res/layout
- ☒ Signup.java - donation-android/src/app/donation
- ☐ Not overwriting android-support-v4.jar because the files are identical

AndroidManifest.xml

Original Source	Refactored Source
<pre><category android:name="android.intent.category.LAUNCHER" /> </intent-filter> </activity> <activity android:name="app.activities.MainActivity" android:label="@string/app_name" android:icon="@mipmap/ic_launcher" android:theme="@style/AppTheme" /> </activity> <activity android:name="app.activities.LoginActivity" android:label="@string/donate" android:parentActivityName="app.activities.MainActivity" android:theme="@style/AppTheme.NoActionBar" /> </activity> </application> </manifest></pre>	<pre> android:name="app.activities.MainActivity" android:label="@string/app_name" android:icon="@mipmap/ic_launcher" android:theme="@style/AppTheme" /> </activity> <activity android:name="app.activities.LoginActivity" android:label="@string/donate" android:parentActivityName="app.activities.MainActivity" android:theme="@style/AppTheme.NoActionBar" /> </activity> <activity android:name="app.donation.SignupActivity" android:label="@string/title_activity_signup" android:parentActivityName="app.activities.MainActivity" android:theme="@style/AppTheme.NoActionBar" /> </activity> </application> </manifest></pre>

? < Back Next > Cancel Finish



Exercise 5 Solution

strings.xml

```
<string name="title_activity_signup">Signup</string>
<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 Activity
{
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_signup);
    }

    public void registerPressed (View view)
    {
        startActivity (new Intent(this, Welcome.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" />
```

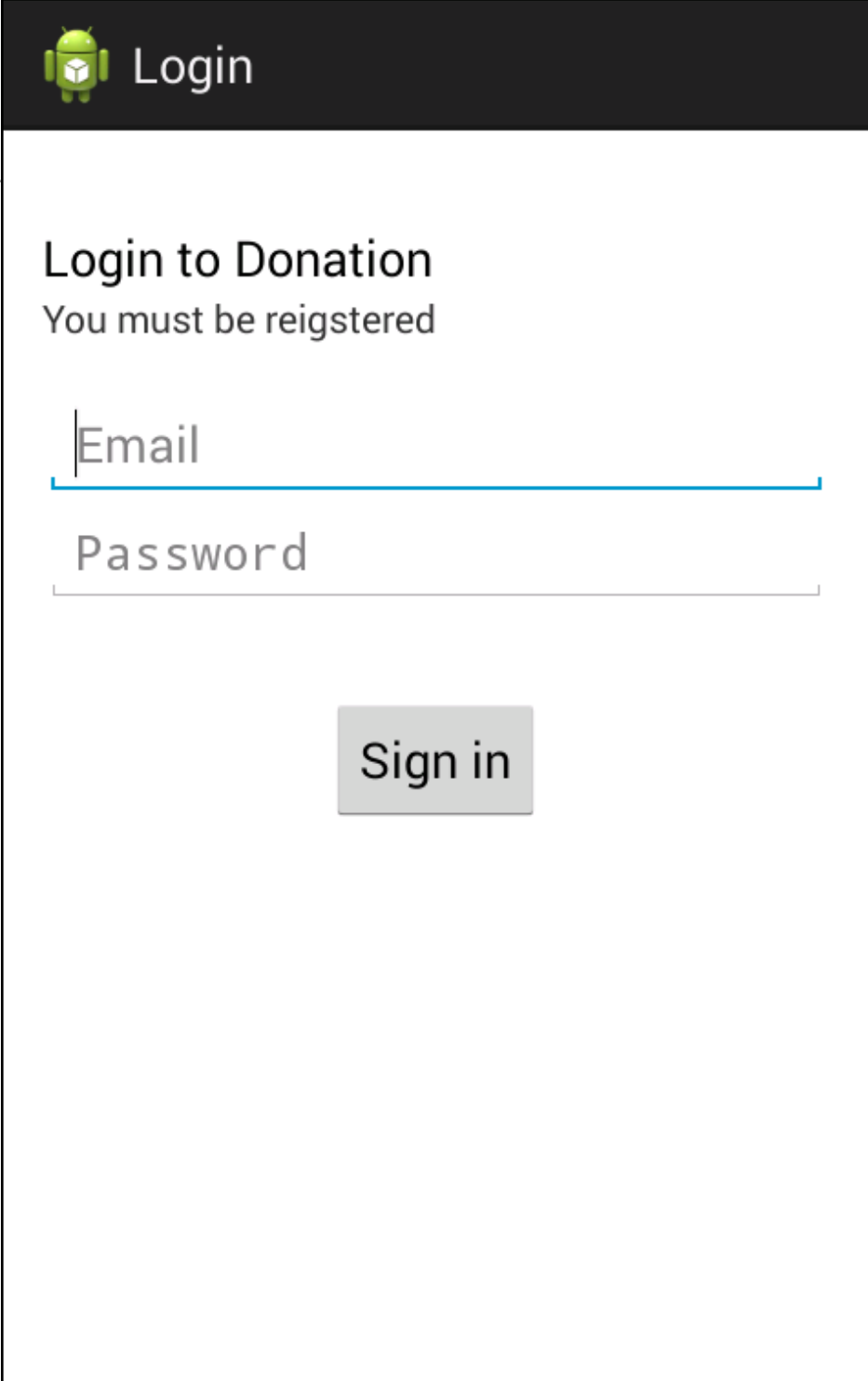
```
<EditText
    android:id="@+id/Password"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/Email"
    android:layout_alignParentRight="true"
    android:layout_below="@+id/Email"
    android:ems="10"
    android:hint="@string/signupPassword"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

```
<Button
    android:id="@+id/register"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignLeft="@+id/Password"
    android:layout_alignParentRight="true"
    android:layout_below="@+id/Password"
    android:text="@string/signupRegister" />
```

activity_signup.xml

Exercise 6

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



The image shows a mobile app interface for a login screen. At the top, there is a dark header bar with a green Android robot icon and the word "Login" in white. Below the header, the main content area has a title "Login to Donation" and a subtitle "You must be reigstered". There are two input fields: "Email" with a blue underline and "Password" with a grey underline. A grey "Sign in" button is positioned below the password field.

Exercise 6 Solution

```
<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" />

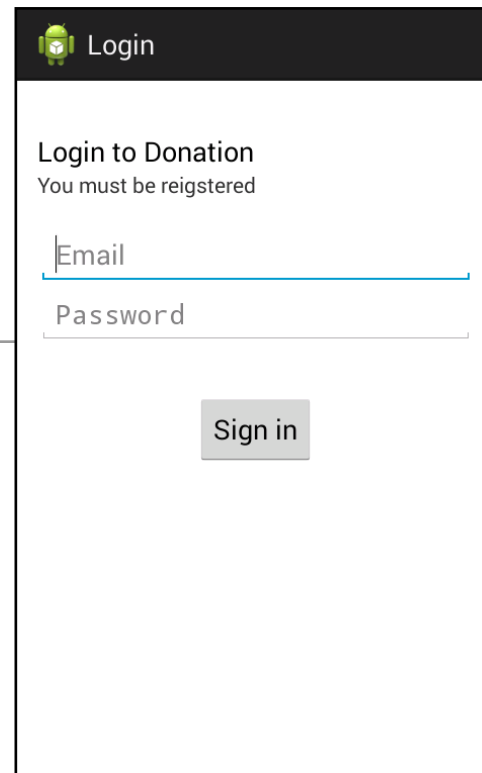
    <EditText
        android:id="@+id/loginEmail"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/loginSubtitle"
        android:layout_alignRight="@+id/loginSubtitle"
        android:layout_below="@+id/loginSubtitle"
        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: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" />

</RelativeLayout>
```



```
public class Login extends Activity
{
    @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));
    }
}
```



Donation App

Please give generously

☒ PayPal

☐ Direct

1000

0

1

Amount:

Donate

Total so far:

0

Settings

Report

Logout

Exercise 7

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



Login

Sign up



Report

3

PayPal

5

PayPal

10

PayPal

100

PayPal

Settings

Donate

Logout

Exercise 7 Solution

strings.xml

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

donate.xml

report.xml

```
<item  
    android:id="@+id/menuLogout"  
    android:orderInCategory="100"  
    android:showAsAction="never"  
    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 Activity
{
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_signup);
    }

    public void registerPressed (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));
    }
}
```

 Signup

Sign up for the Donation App

Enter details below

First name

Last Name

Email

Password

Register

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

    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.validateUser(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();
        }
    }
}
```

 Login

Login to Donation
You must be registered

Email

Password

Sign in





















Lab 08 Exercise 1

- Using the Repository lecture and lab as a guide:
- Commit your donation project to a bitbucket repository.

Lecture

- Review the purpose and primary features of:
 - git
 - egit
 - sourcetree
 - bitbucket
- Walk through the Repositories lab in depth





















→ ⚙️ 🔍 ⌕ ✖️ 👤

 a-repositories-1	        	<input checked="" type="checkbox"/>
 b-repositories-2	        	<input checked="" type="checkbox"/>

Lab

- Explore the primary features of a git repository
- Set up a repository for a Play project in eclipse
- Progressively introduce and commit new features
- Review the repository in Sourcetree
- Push the repository to bitbucket - a cloud provider of git services.

→ ⚙️ 🔍 ⌕ ✖️ 👤

 lab	        	<input checked="" type="checkbox"/>
 (offline archive)	        	<input checked="" type="checkbox"/>

Exercise 2 Database Version of Donation

- Re work the app to use a database instead of the in-memory data structures.
- To get you started, the following class is a 'helper' class, designed to manage donation and user objects in a database in Android:
 - **DbDonationHelper.java** (in lab)
- This class relies on slightly modified Model classes, which implement getters and setters:
 - **User.java** (in lab)
 - **Donation.java** (in lab)
- You should just need to rewrite DonationApp class to used this helper instead of the current approach (in memory data structures)




UML Model for donation-android

- Using the Modelling lecture and lab as a guide:
- Create a model of the current version of Donation-android.
- (sample model file in lab)

Lecture

- Introduce UML
- Explore the fundamentals of the 'one to many' relationship including:
 - How it is modelled in UML
 - How it is implemented in JPA
- Furthermore, explore how to model and implementation a bidirectional variant of this relationship



→ ⚙️ 🗑️ 🔍 👤

 a-uml-intro	🔍 ⬅️ ➡️ ⚙️ 🗑️ 🔍 👤	<input checked="" type="checkbox"/>
 b-modeling	🔍 ⬅️ ➡️ ⚙️ 🗑️ 🔍 👤	<input checked="" type="checkbox"/>
 c-relationships	🔍 ⬅️ ➡️ ⚙️ 🗑️ 🔍 👤	<input checked="" type="checkbox"/>

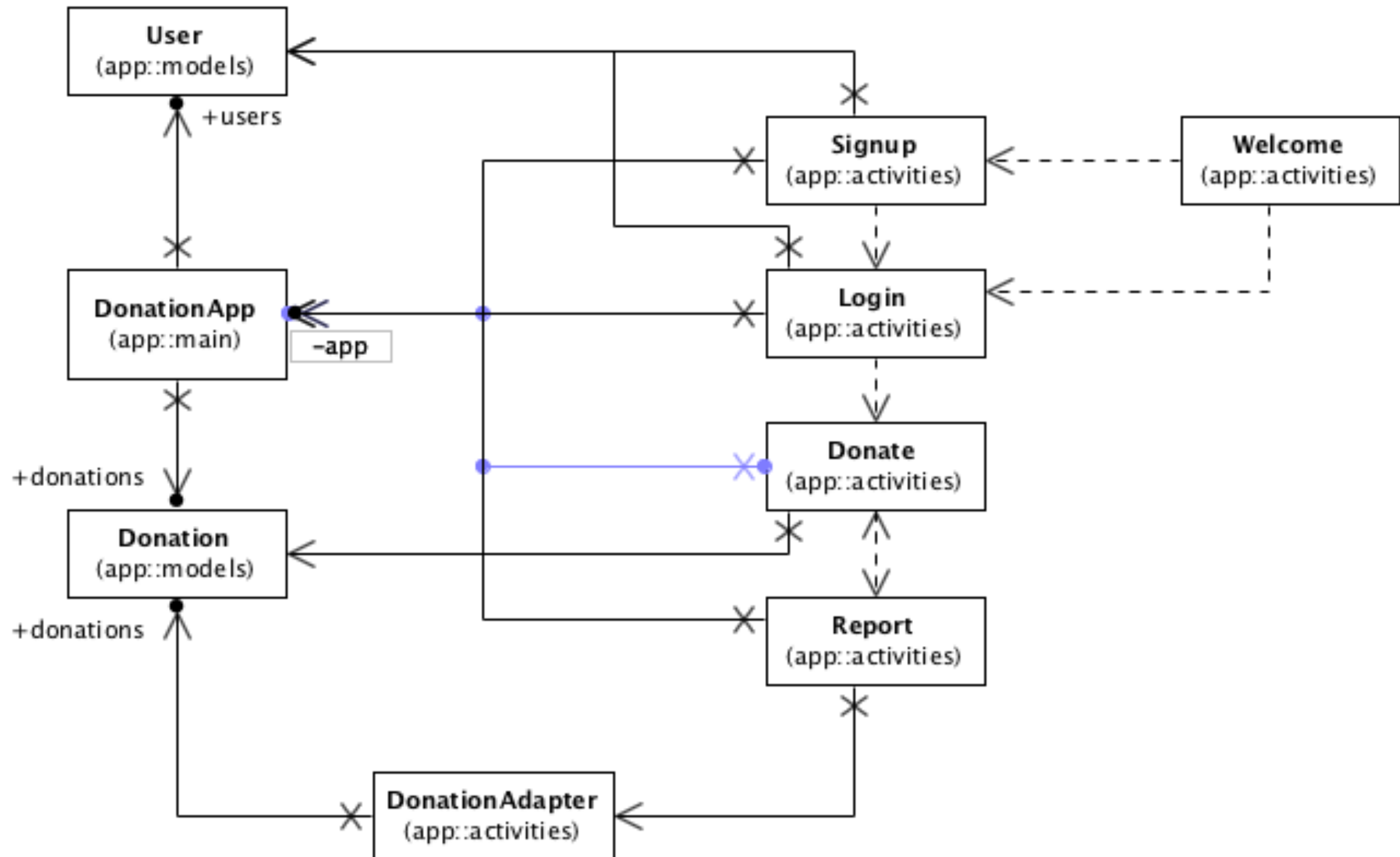
Lab

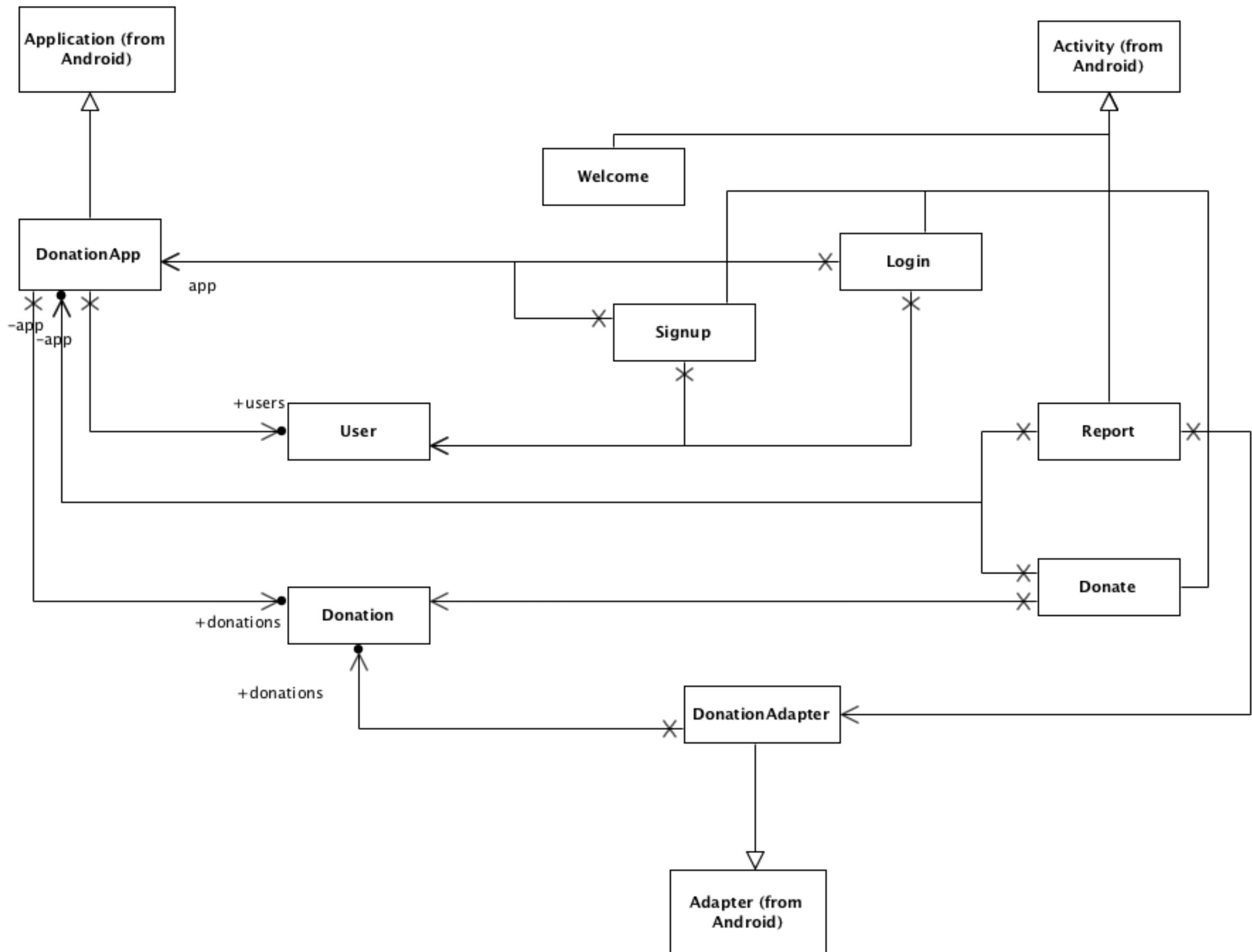
- Introduce UML Class Diagram modeling using Visual Paradigm
- Define a simple model and implement it in Play
- Write comprehensive unit tests to exercise the model

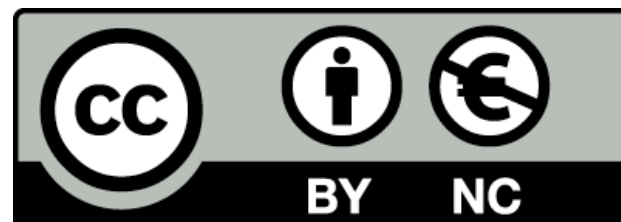
→ ⚙️ 🗑️ 🔍 👤

 lab	🔍 ⬅️ ➡️ ⚙️ 🗑️ 🔍 👤	<input checked="" type="checkbox"/>
 (offline archive)	🔍 ⬅️ ➡️ ⚙️ 🗑️ 🔍 👤	<input checked="" type="checkbox"/>

Models







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

