Mobile Applications 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

eLearning support unit

Donate Android v4

Further donation-android / donation-service integration

- The v3 application uses two aspects of the donation play service:
  - Let users log in who are already 'registered' online
  - Display a list of donations downloaded form the web service
- Incorporate the following features:
  - when users sign up, their details are send to the donation-service play app
  - when a donation is made, it is sent to the donation-service play app
- The facilities are already in the play app for this, so you will only be making changes to the android application.

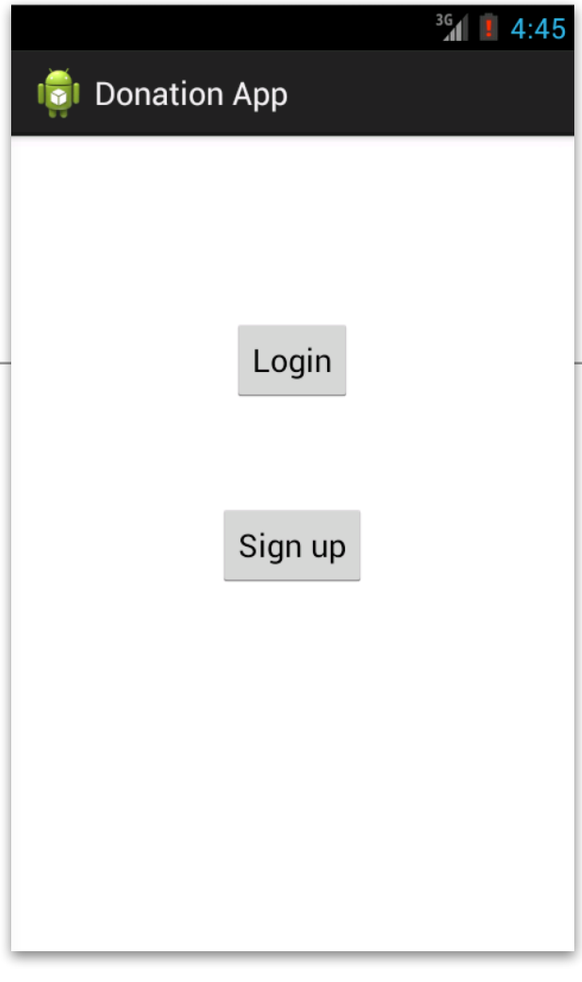
donationServiceAvailable

- Maintain a flag to indicate if the donation service is available or not
- Update this when we attempt to access the service

```
public class DonationApp extends Application
{
  public boolean donationServiceAvailable = false;
  //…


}
```

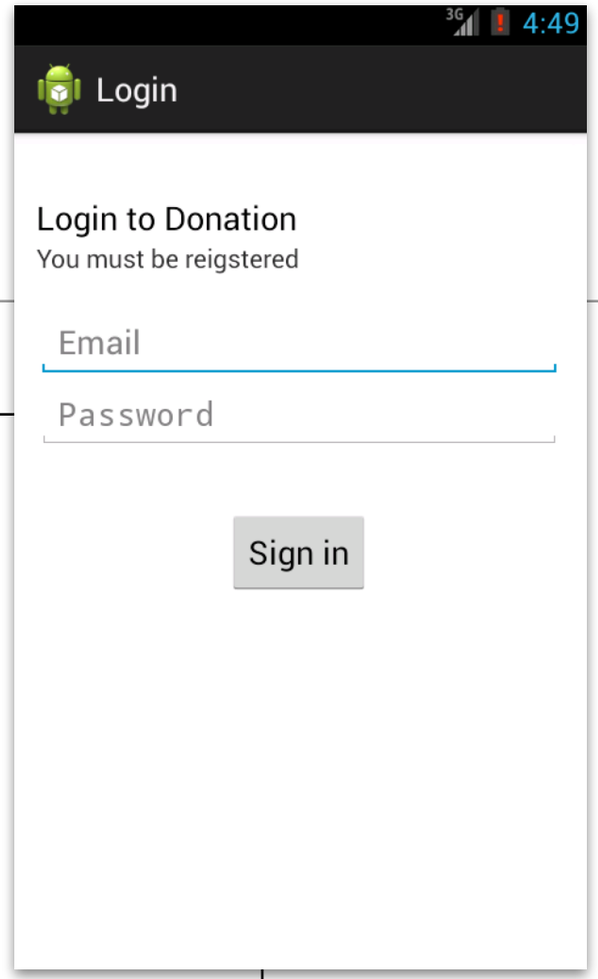
Welcome

```
public class Welcome extends Activity implements Response<User>
{
  DonationApp app;

  @Override
  public void onCreate(Bundle savedInstanceState)
  {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_welcome);
    app = (DonationApp) getApplication();

    DonationServiceAPI.getUsers(this, this, "Retrieving list of users");
  }

  void serviceUnavailableMessage()
  {
    Toast toast = Toast.makeText(this,
                                 "Donation Service Unavailable. Try again later”,
                                 Toast.LENGTH_LONG);
    toast.show();
  }

  public void loginPressed (View view)
  {
    if (app.donationServiceAvailable)
    {
      startActivity (new Intent(this, Login.class));
    }
    else
    {
      serviceUnavailableMessage();
    }
  }

  public void signupPressed (View view)
  {
    if (app.donationServiceAvailable)
    {
      startActivity (new Intent(this, Signup.class));
    }
    else
    {
      serviceUnavailableMessage();
    }
  }
```



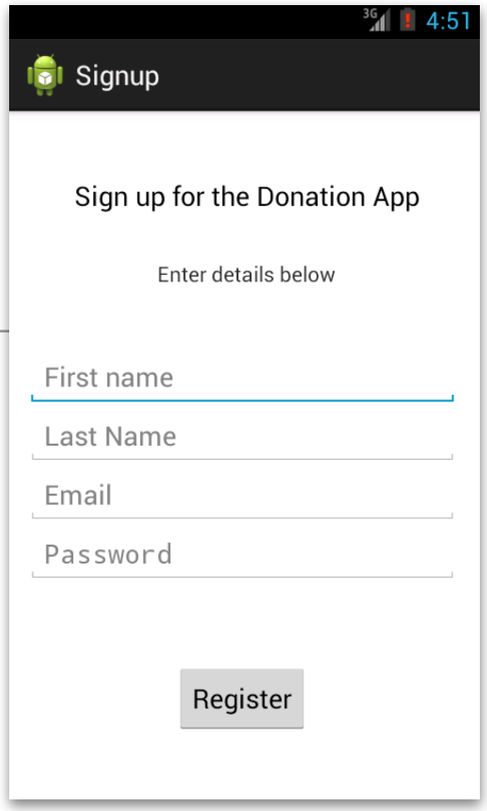

```
  @Override
  public void setResponse(List<User> aList)
  {
    app.users = aList;
    app.donationServiceAvailable = true;
  }

  @Override
  public void errorOccurred(Exception e)
  {
    app.donationServiceAvailable = false;
    serviceUnavailableMessage();
  }

  @Override
  public void setResponse(User anObject)
  {}
}
```

Login




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

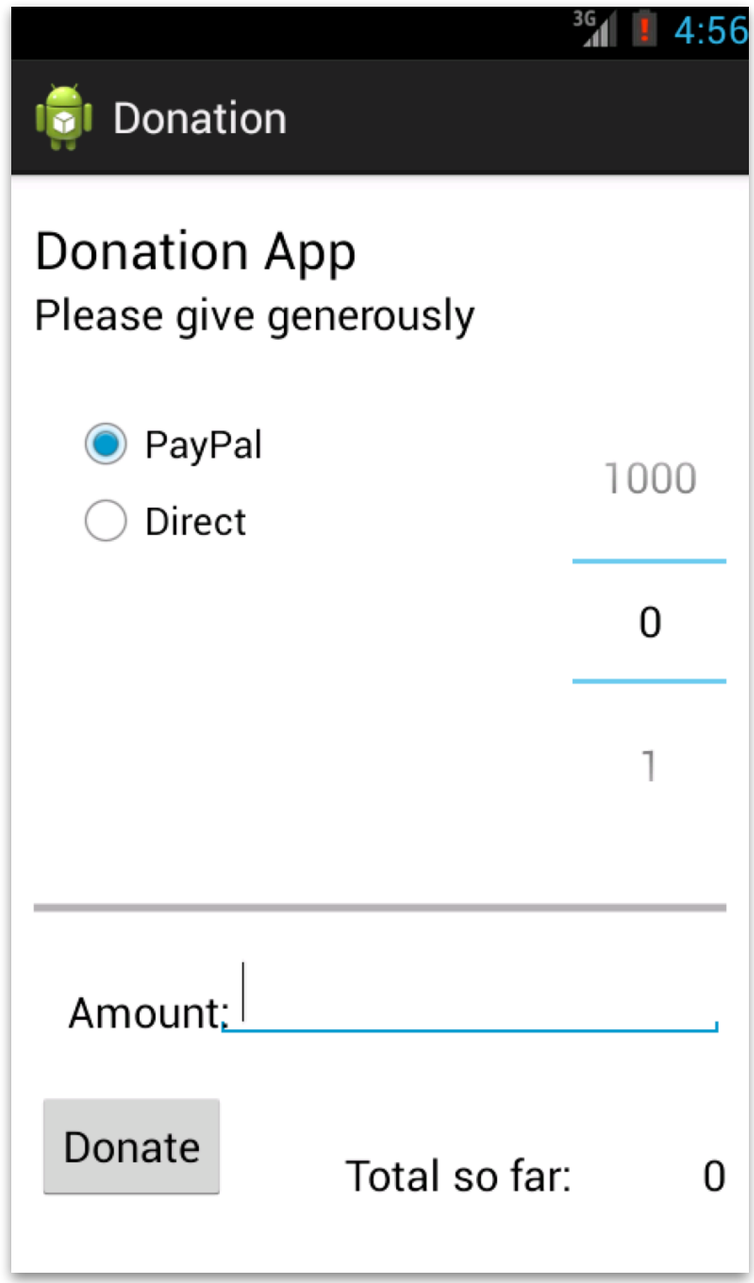
```java
public class Signup extends Activity implements Response<User>
{
  private DonationApp app;

  @Override
  protected void onCreate(Bundle savedInstanceState)
  {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_signup);
    app = (DonationApp) getApplication();
  }

  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());

    DonationServiceAPI.createUser(this, this, "Registering new user", user);
  }
```




```java
  @Override
  public void setResponse(User user)
  {
    app.users.add(user);
    startActivity (new Intent(this, Welcome.class));
  }

  @Override
  public void errorOccurred(Exception e)
  {
    app.donationServiceAvailable = false;
    Toast toast = Toast.makeText(this,
                                 "Donation Service Unavailable. Try again later",
                                  Toast.LENGTH_LONG);
    toast.show();
    startActivity (new Intent(this, Welcome.class));
  }
}
```

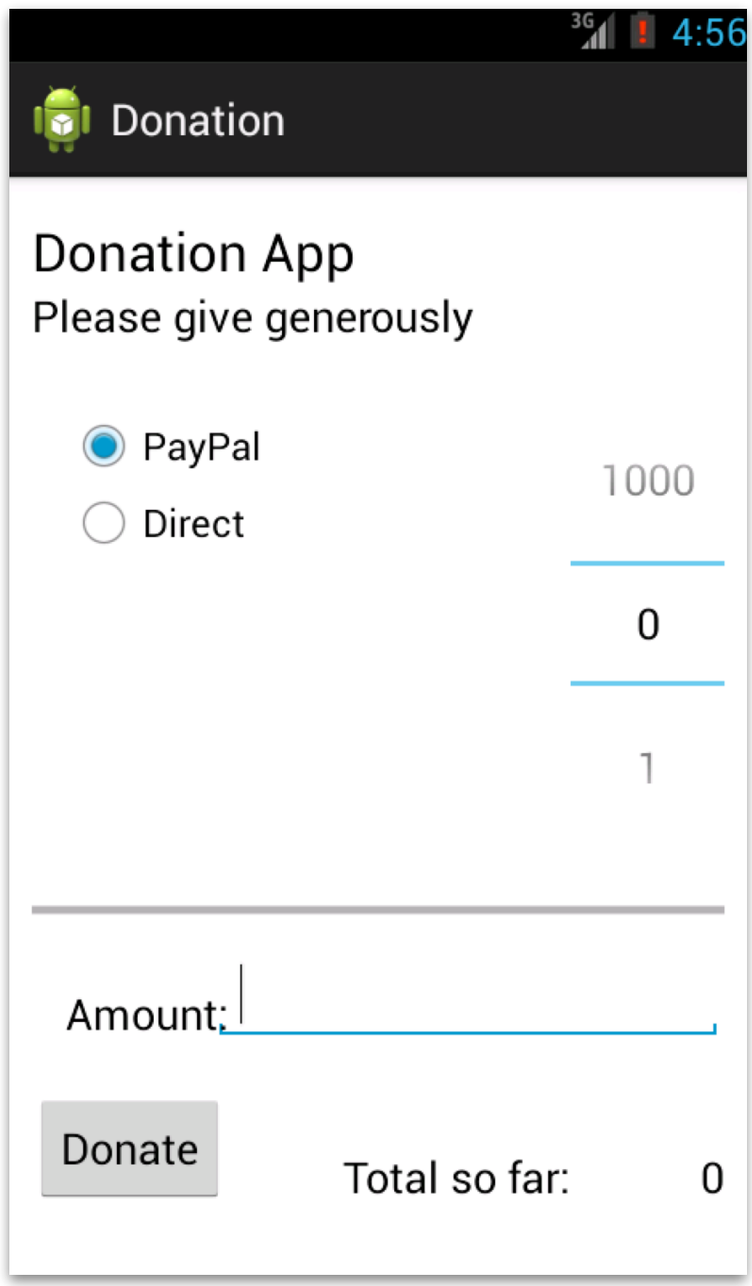
```
public class Donate extends Activity implements Response<Donation>
{
  private RadioGroup   paymentMethod;
  private ProgressBar  progressBar;
  private NumberPicker amountPicker;
  private TextView     amountText;
  private TextView     amountTotal;
  private DonationApp  app;

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

    app = (DonationApp) getApplication();

    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(app.target);
  }

  @Override
  public boolean onCreateOptionsMenu(Menu menu)
  {
    getMenuInflater().inflate(R.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;
      case R.id.menuLogout : startActivity (new Intent(this, Welcome.class));
                             break;
    }
    return true;
  }
```

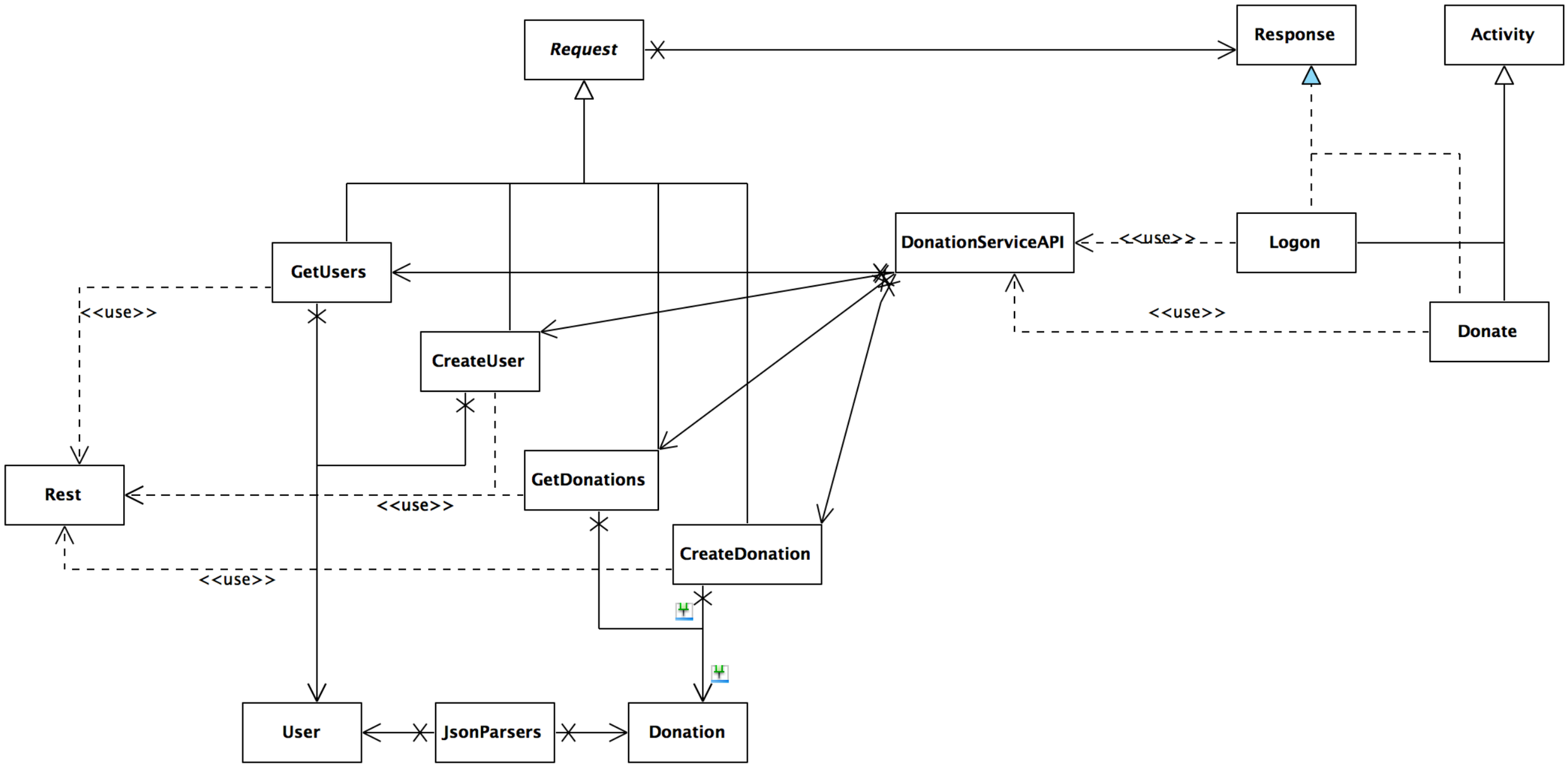
Donate

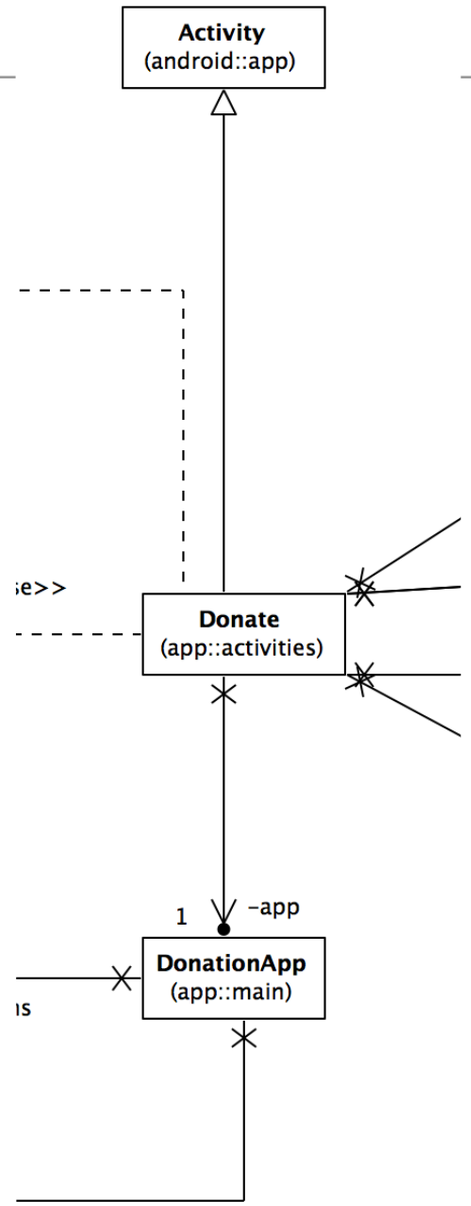


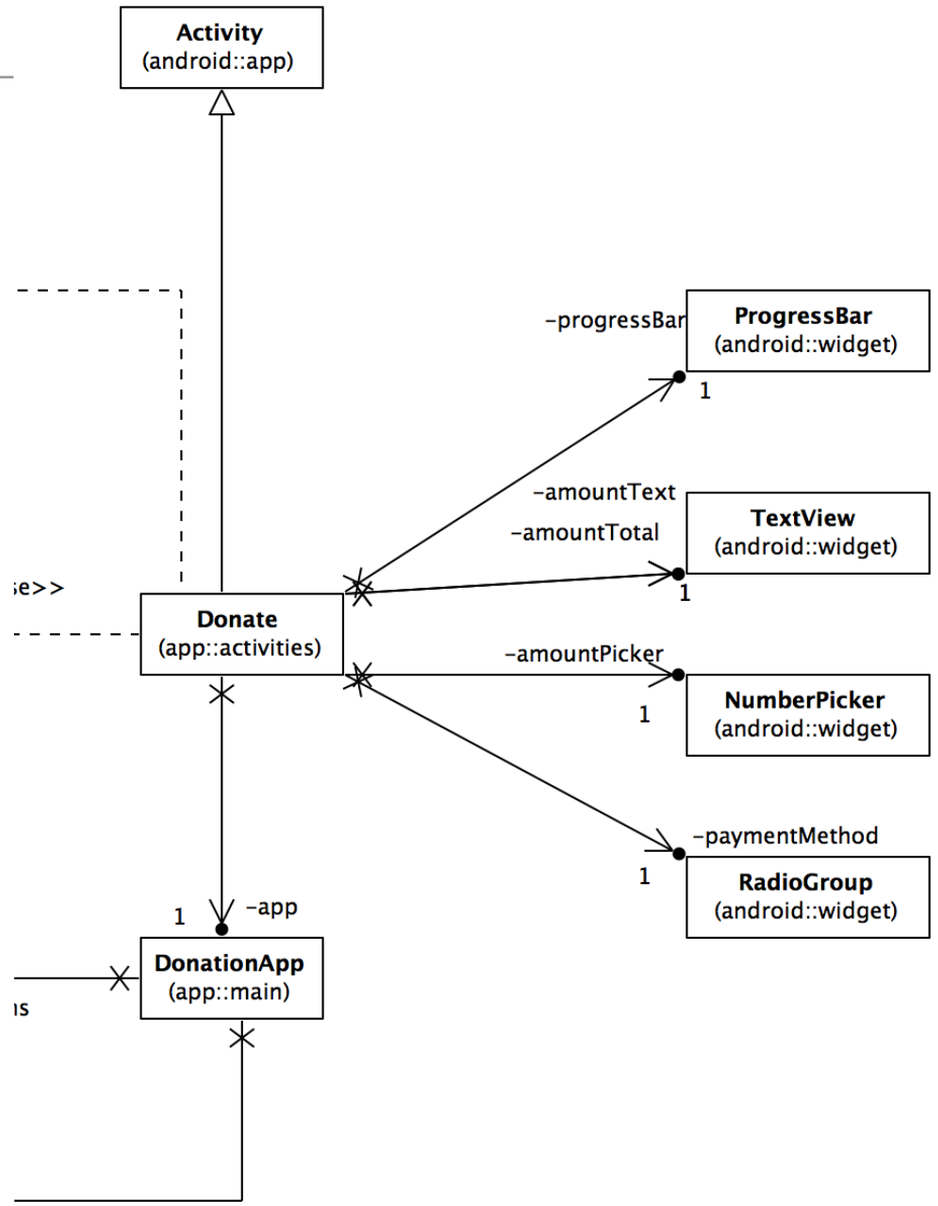
```
public class Donate extends Activity implements Response<Donation>
{
  public void donateButtonPressed (View view)
  {
    String method = paymentMethod.getCheckedRadioButtonId() == R.id.PayPal
                    ? "PayPal" : "Direct";
    int donatedAmount =  amountPicker.getValue();
    if (donatedAmount == 0)
    {
      String text = amountText.getText().toString();
      if (!text.equals(""))
        donatedAmount = Integer.parseInt(text);
    }
    if (donatedAmount > 0)
    {
      DonationServiceAPI.createDonation(this, this,
                                        "Registering new donation...",
                                        new Donation(donatedAmount, method));
    }
   }

  @Override
  public void setResponse(Donation acceptedDonation)
  {
    Toast toast = Toast.makeText(this, "Donation Accepteed", Toast.LENGTH_SHORT);
    toast.show();
    app.newDonation(acceptedDonation);
    progressBar.setProgress(app.totalDonated);
    String totalDonatedStr = "$" + app.totalDonated;
    amountTotal.setText(totalDonatedStr);
    amountText.setText("");
    amountPicker.setValue(0);
  }

  @Override
  public void errorOccurred(Exception e)
  {
    Toast toast = Toast.makeText(this,
                                 "Donation Service Unavailable. Try again later",
                                 Toast.LENGTH_LONG);
    toast.show();
  }

  @Override
  public void setResponse(List<Donation> aList)
  {}
}
```

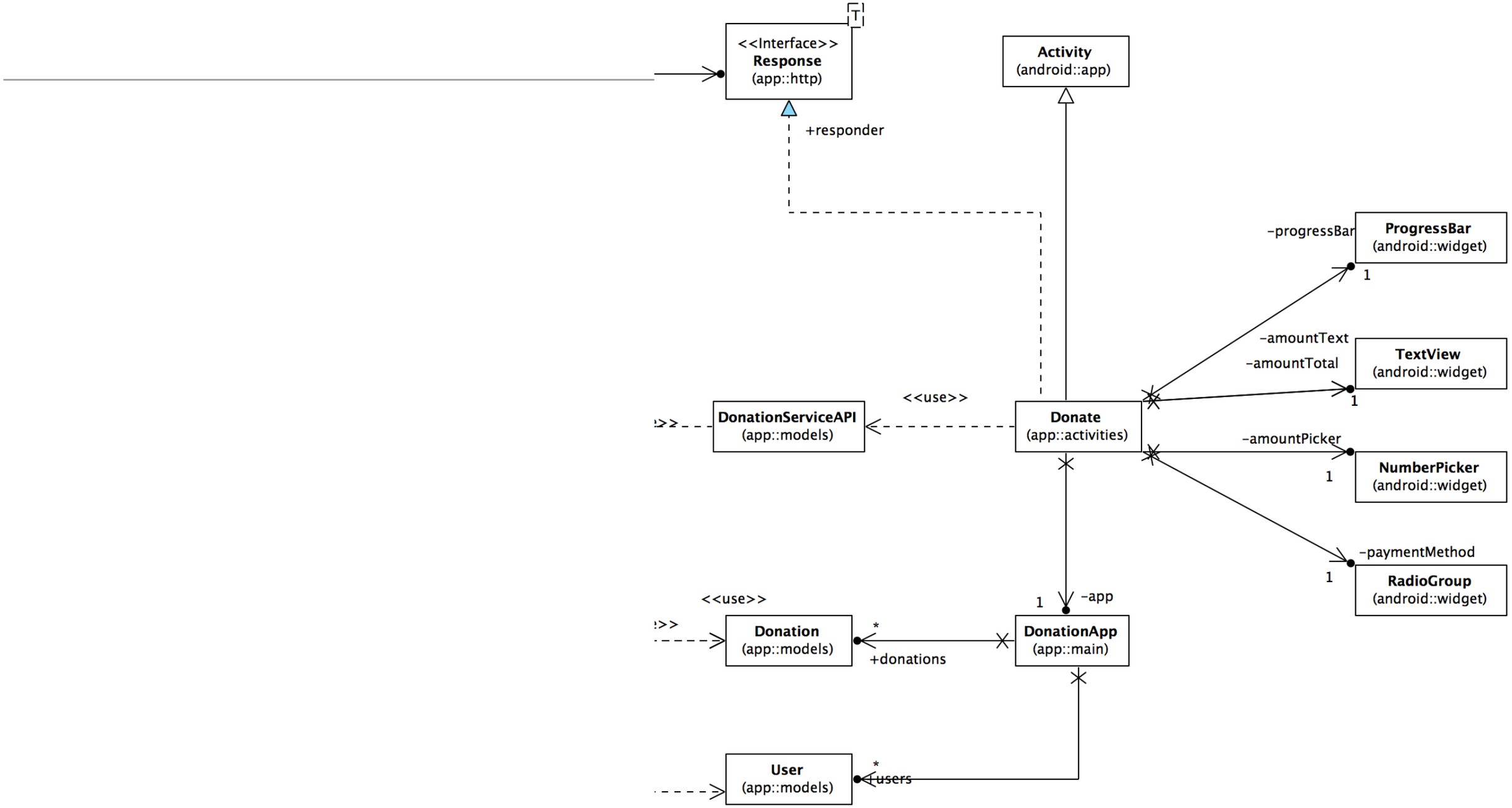
Donate

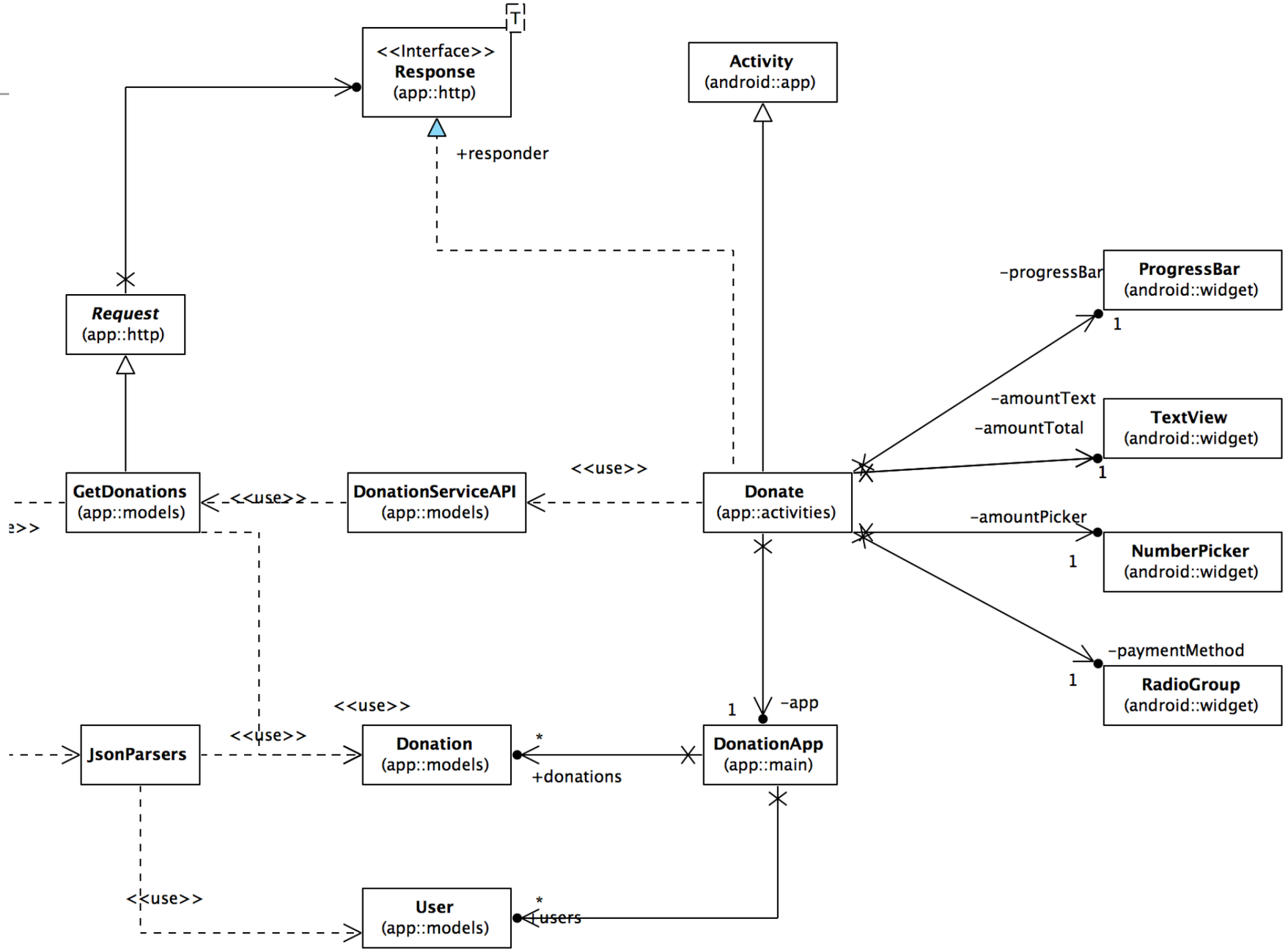


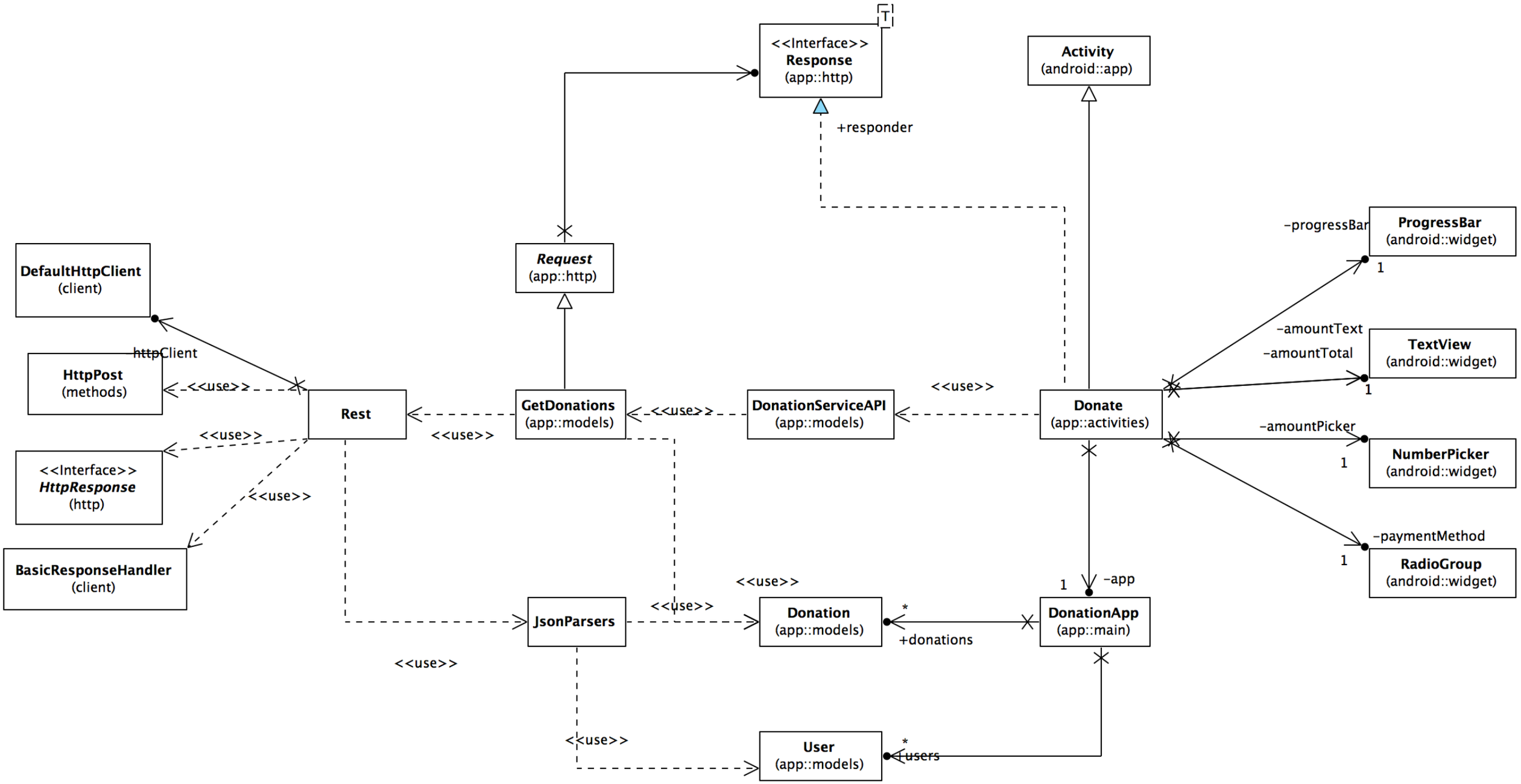

Request
Response
Activity
GetUsers
DonationServiceAPI
Logon
<<use>>
<<use>>
Donate
CreateUser
Rest
GetDonations
<<use>>
CreateDonation
<<use>>
User
JsonParsers
Donation


Activity
(android::app)
Donate
(app::activities)
1
-app
DonationApp
(app::main)


Activity
(android::app)
-progressBar
ProgressBar
(android::widget)
1
-amountText
-amountTotal
TextView
(android::widget)
1
Donate
(app::activities)
-amountPicker
1
NumberPicker
(android::widget)
-paymentMethod
1
RadioGroup
(android::widget)
1
-app
DonationApp
(app::main)

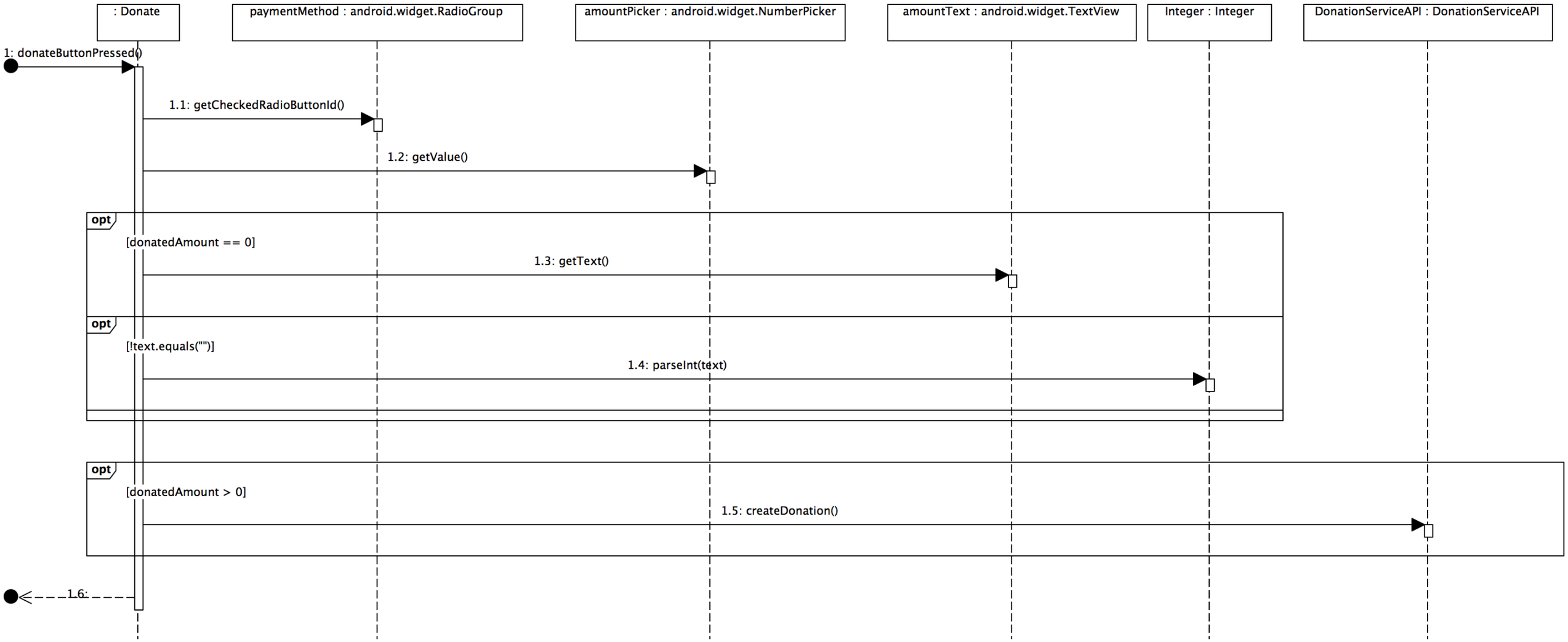

T
<<Interface>>
Response
(app::http)
Activity
(android::app)
+responder
-progressBar
ProgressBar
(android::widget)
1
-amountText
-amountTotal
TextView
(android::widget)
1
<<use>>
DonationServiceAPI
(app::models)
Donate
(app::activities)
-amountPicker
1
NumberPicker
(android::widget)
-paymentMethod
1
RadioGroup
(android::widget)
<<use>>
1
-app
Donation
(app::models)
*
+donations
DonationApp
(app::main)
User
(app::models)
*
users


T
<<Interface>>
Response
(app::http)
Activity
(android::app)
+responder
Request
(app::http)
-progressBar
ProgressBar
(android::widget)
1
-amountText
-amountTotal
TextView
(android::widget)
1
<<use>>
GetDonations
(app::models)
<<use>>
DonationServiceAPI
(app::models)
Donate
(app::activities)
-amountPicker
1
NumberPicker
(android::widget)
-paymentMethod
1
RadioGroup
(android::widget)
<<use>>
1
-app
JsonParsers
<<use>>
<<use>>
Donation
(app::models)
*
+donations
DonationApp
(app::main)
<<use>>
User
(app::models)
*
+users


T
<<Interface>>
Response
(app::http)
Activity
(android::app)
+responder
DefaultHttpClient
(client)
Request
(app::http)
-progressBar
ProgressBar
(android::widget)
1
-amountText
-amountTotal
TextView
(android::widget)
1
httpClient
HttpPost
(methods)
<<use>>
Rest
GetDonations
(app::models)
<<use>>
DonationServiceAPI
(app::models)
<<use>>
Donate
(app::activities)
-amountPicker
1
NumberPicker
(android::widget)
<<use>>
<<Interface>>
HttpResponse
(http)
<<use>>
-paymentMethod
1
RadioGroup
(android::widget)
BasicResponseHandler
(client)
<<use>>
1
-app
<<use>>
JsonParsers
<<use>>
Donation
(app::models)
*
+donations
DonationApp
(app::main)
<<use>>
<<use>>
User
(app::models)
*
-users

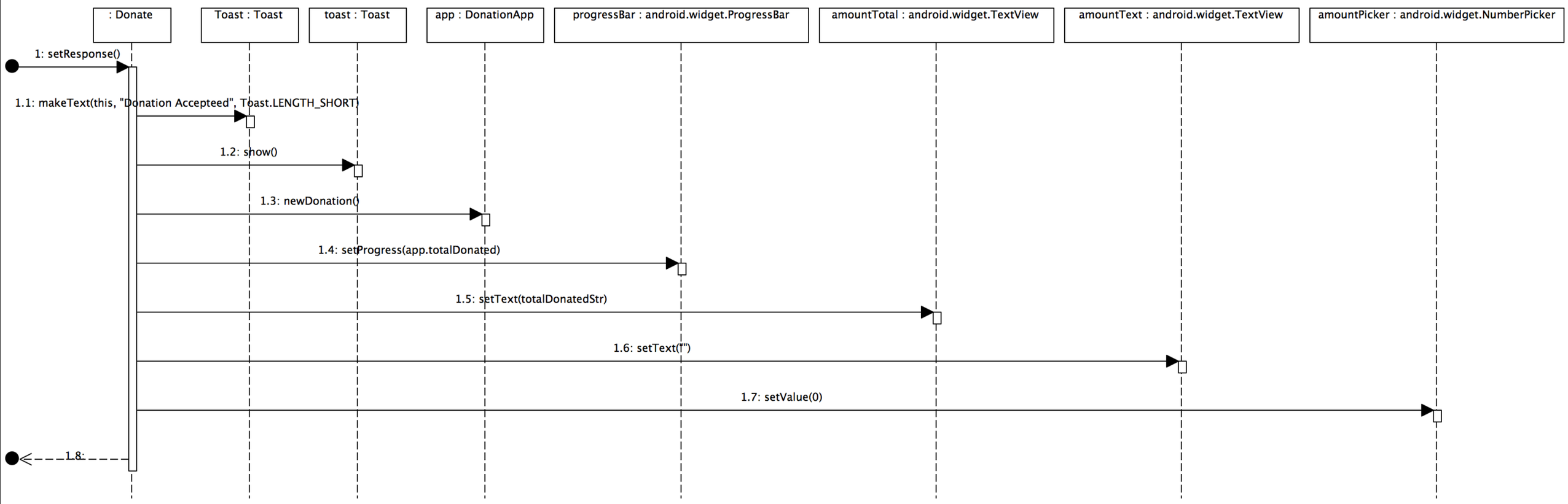
donateButtonPressed

```
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)
   {
     DonationServiceAPI.createDonation(this, this,
                                       "Registering new donation...",
                                       new Donation(donatedAmount, method));
   }
  }
```

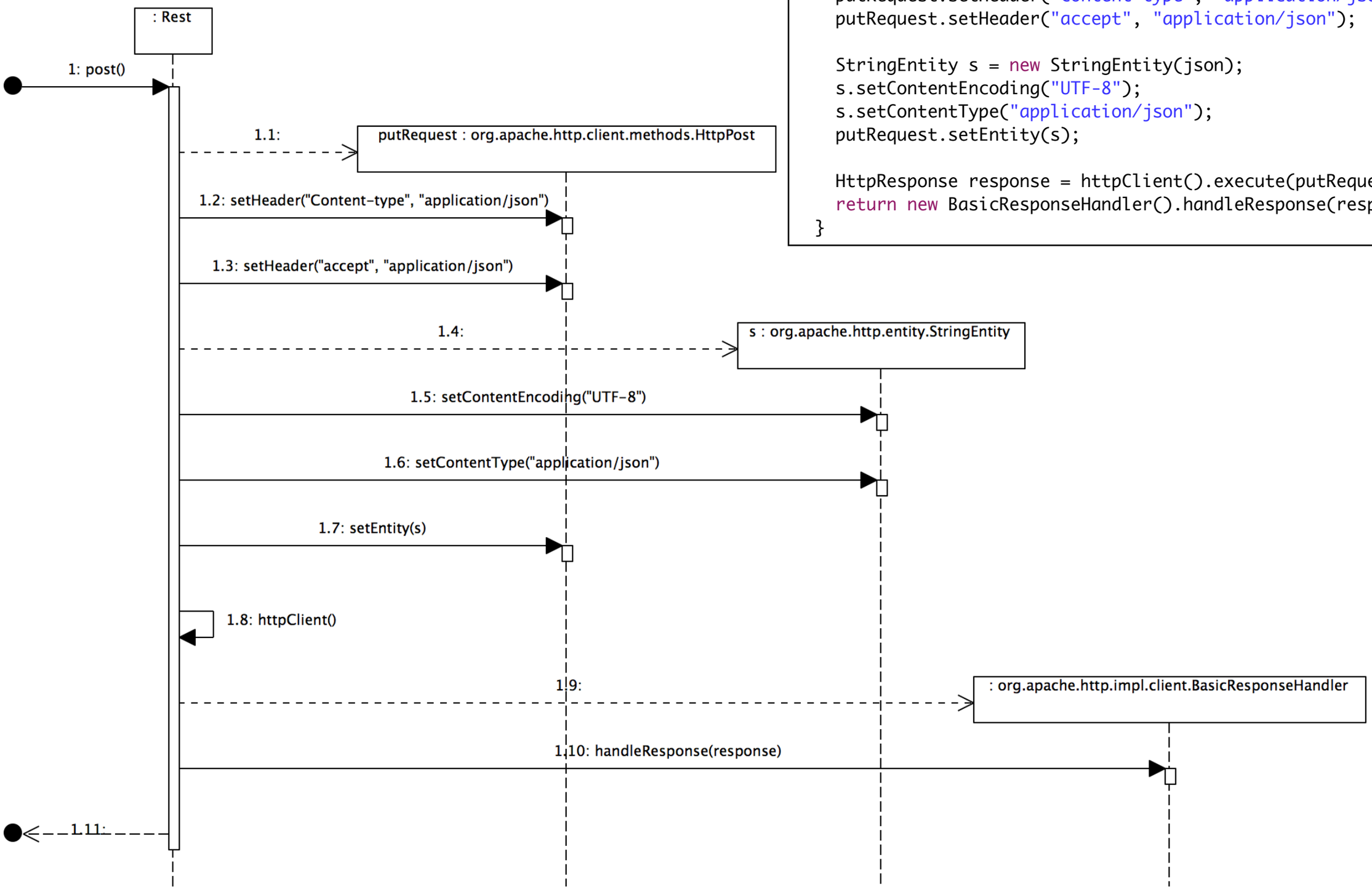


setResponse

```
@Override
 public void setResponse(Donation acceptedDonation)
 {
   Toast toast = Toast.makeText(this, "Donation Accepteed", Toast.LENGTH_SHORT);
   toast.show();
   app.newDonation(acceptedDonation);
   progressBar.setProgress(app.totalDonated);
   String totalDonatedStr = "$" + app.totalDonated;
   amountTotal.setText(totalDonatedStr);
   amountText.setText("");
   amountPicker.setValue(0);
 }
```



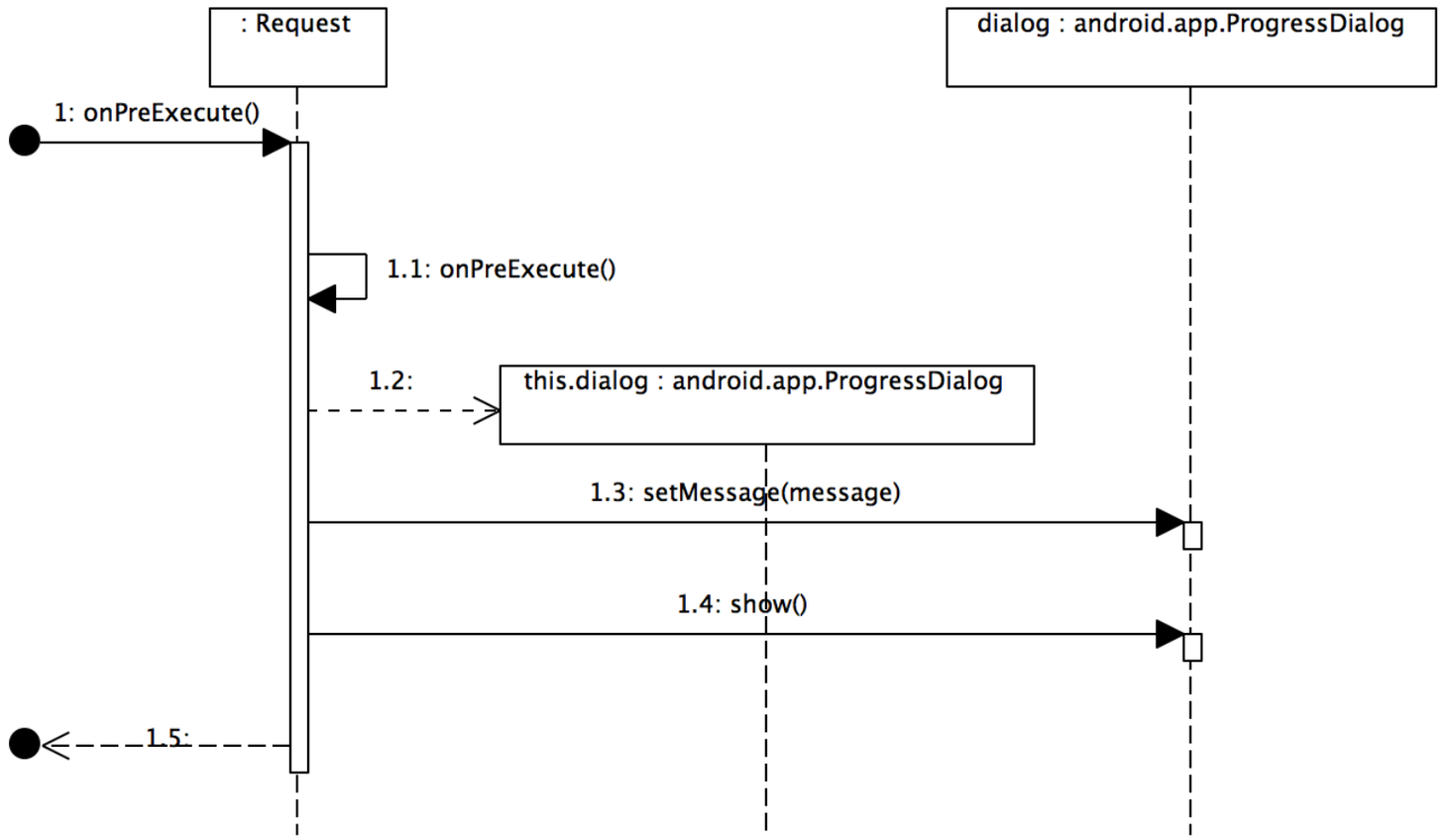
Rest.Post




```
public static String post(String path, String json) throws Exception
{
  HttpPost putRequest = new HttpPost(URL + path);
  putRequest.setHeader("Content-type", "application/json");
  putRequest.setHeader("accept", "application/json");

  StringEntity s = new StringEntity(json);
  s.setContentEncoding("UTF-8");
  s.setContentType("application/json");
  putRequest.setEntity(s);

  HttpResponse response = httpClient().execute(putRequest);
  return new BasicResponseHandler().handleResponse(response);
}
```

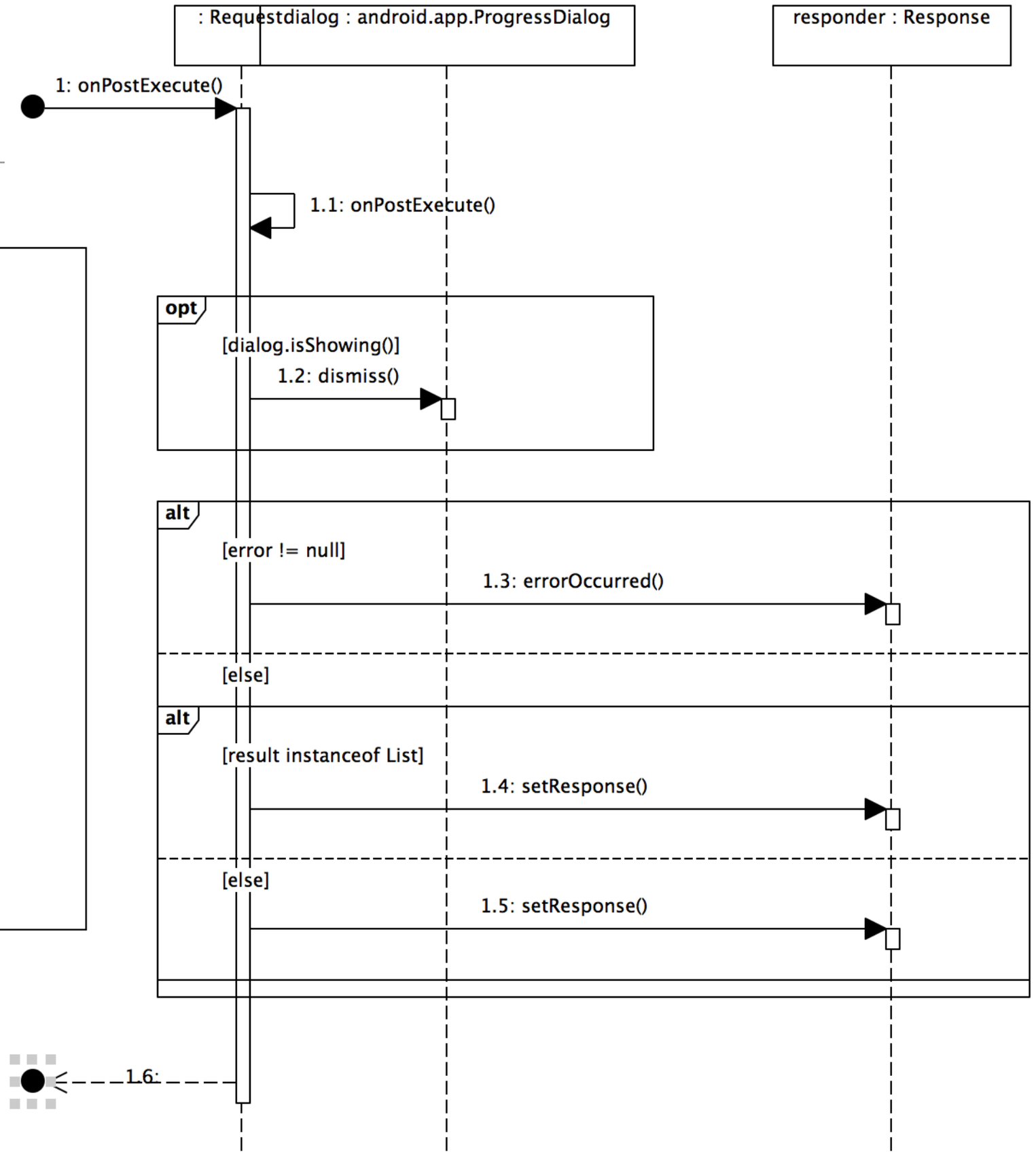
Request.onPreExecute()

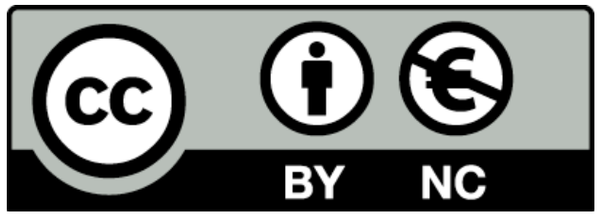
```
public abstract class Request extends AsyncTask<Object, Void, Object>
{
  public Response          responder;
  public ProgressDialog    dialog;
  public Context           context;
  public String            message;
  public Exception         error;

  public Request(Context context, Response responder, String message)
  {
    this.responder = responder;
    this.context = context;
    this.message = message;
  }

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



Request.onPostExecute

```
protected void onPostExecute(Object result)
 {
   super.onPostExecute(result);
   if (dialog.isShowing())
   {
     dialog.dismiss();
   }
   if (error != null)
   {
     responder.errorOccurred(error);
   }
   else
   {
     if (result instanceof List)
     {
       responder.setResponse((List)result);
     }
     else
     {
       responder.setResponse(result);
     }
   }
 }
```





Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning support unit