

Design Patterns

MSc 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>



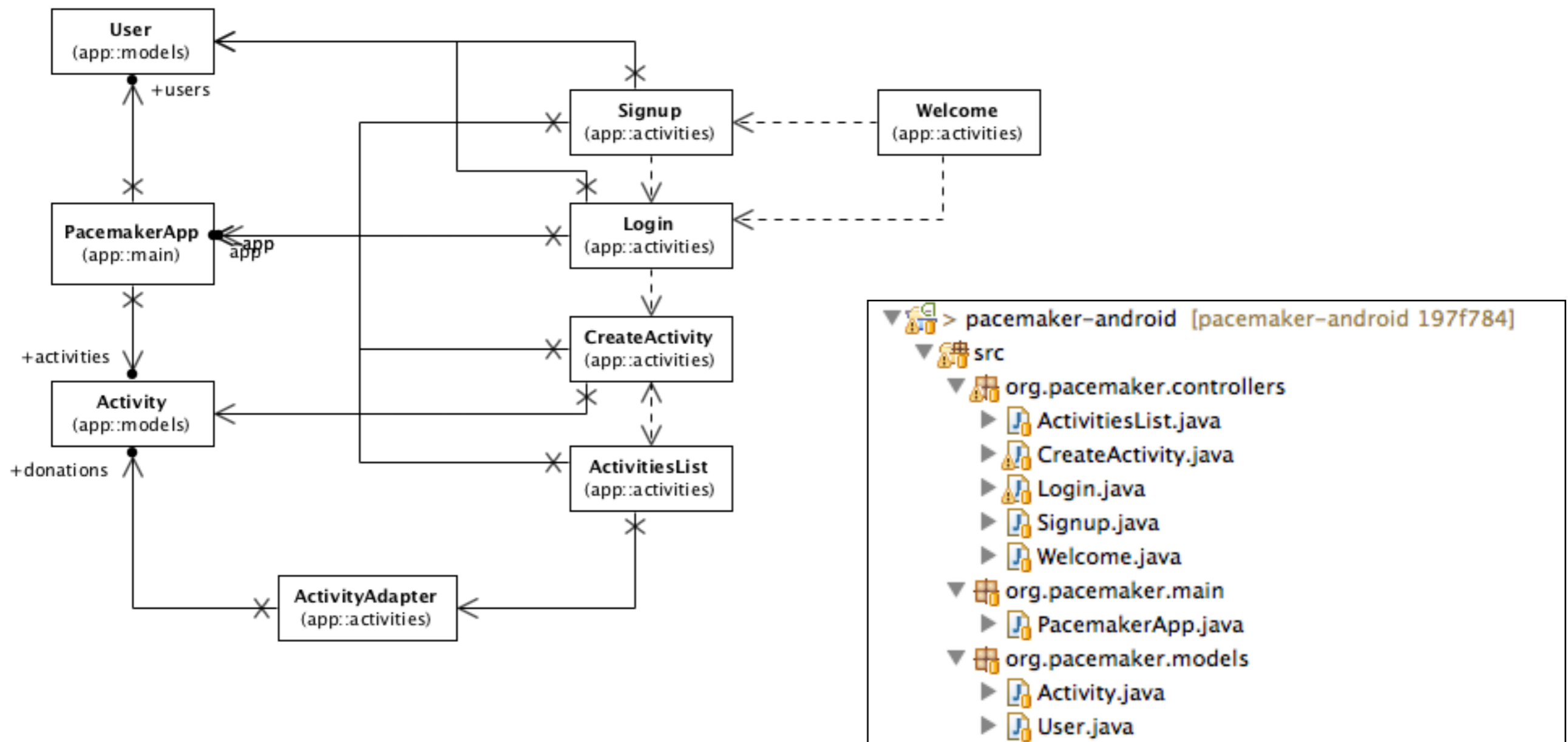
Waterford Institute of Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRCE



Pacemaker V3 - Lab-08

Pacemaker V3 - Lab 08

- Standalone version (no service access)



Model

```
public class User
{
    public Long id;
    public String firstname;
    public String lastname;
    public String email;
    public String password;

    public User()
    {}

    public User(String firstname, String lastname, String email, String password)
    {
        this.firstname = firstname;
        this.lastname = lastname;
        this.email = email;
        this.password = password;
    }
}
```

```
public class Activity
{
    public Long id;
    public String type;
    public String location;
    public double distance;

    public Activity()
    {}

    public Activity(String type, String location, double distance)
    {
        this.type = type;
        this.location = location;
        this.distance = distance;
    }
}
```

App

```
public class PacemakerApp extends Application
{
    private Map<String, User>          users          = new HashMap<String, User>();
    private Map<String, List<Activity>> activities    = new HashMap<String, List<Activity>>();
    private User                      loggedInUser;

    public void registerUser(User user)
    {
        users.put(user.email, user);
        activities.put(user.email, new ArrayList<Activity>());
    }
    public boolean loginUser(String email, String password)
    {
        loggedInUser = users.get(email);
        if (loggedInUser != null && !loggedInUser.password.equals(password))
        {
            loggedInUser = null;
        }
        return loggedInUser != null;
    }
    public void logout()
    {
        loggedInUser = null;
    }
    public void createActivity (Activity activity)
    {
        if (loggedInUser != null)
        {
            List<Activity> usersActivities = activities.get(loggedInUser.email);
            activities.put(loggedInUser.email, usersActivities);
            usersActivities.add(activity);
        }
    }
    public List<Activity> getActivities()
    {
        List<Activity> usersActivities = null;
        if (loggedInUser != null)
        {
            usersActivities = activities.get(loggedInUser.email);
        }
        return usersActivities;
    }
    @Override
    public void onCreate()
    {
        super.onCreate();
        Log.v("Pacemaker", "Pacemaker App Started");
    }
}
```



Pacemaker

Login

Sign up

```
public class Welcome extends Activity
{
    PacemakerApp app;

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

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

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



Signup

Sign up for the Pacemaker

Enter details below

```
public class Signup extends Activity
{
    private PacemakerApp app;

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

        app.registerUser(user);

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



Login to Donation

You must be registered

```
public class Login extends Activity
{
    PacemakerApp app;

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

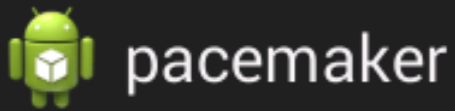
    public void signinPressed (View view)
    {
        app = (PacemakerApp) getApplication();

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

        boolean loggedIn = app.loginUser(email.getText().toString(),
                                         password.getText().toString());

        if (loggedIn)
        {
            startActivity (new Intent(this, CreateActivity.class));
        }
        else
        {
            Toast toast = Toast.makeText(this, "Invalid Credentials",
                                         Toast.LENGTH_SHORT);

            toast.show();
        }
    }
}
```

Enter Activity Details

Enter Activity type...

Enter Location...

20

Distance

0

1

Create Activity

3G 5:53

```
public class CreateActivity extends android.app.Activity
{
    private PacemakerApp    app;
    private Button          createActivityButton;
    private TextView        activityType;
    private TextView        activityLocation;
    private NumberPicker    distancePicker;

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_create);
        app = (PacemakerApp) getApplication();
        createActivityButton = (Button) findViewById(R.id.createActivityButton);
        activityType         = (TextView) findViewById(R.id.activityType);
        activityLocation     = (TextView) findViewById(R.id.activityLocation);
        distancePicker       = (NumberPicker) findViewById(R.id.distancePicker);

        distancePicker.setMinValue(0);
        distancePicker.setMaxValue(20);
    }

    public void createActivityButtonPressed (View view)
    {
        double distance = distancePicker.getValue();
        Activity activity = new Activity (activityType.getText().toString(), activityLocation.getText().toString(), distance);
        app.createActivity(activity);
    }

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

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

Activities

cycle	fenor	19.0
-------	-------	------

```
public class ActivitiesList extends android.app.Activity
{
    private PacemakerApp app;
    private ListView activitiesListView;

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

        app = (PacemakerApp) getApplication();

        activitiesListView = (ListView) findViewById(R.id.activitiesListView);

        List<Activity> activities = app.getActivities();

        ActivityAdapter activitiesAdapter = new ActivityAdapter(this, activities);
        activitiesListView.setAdapter(activitiesAdapter);
        activitiesAdapter.notifyDataSetChanged();
    }

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

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



ActivitiesList

Activities

cycle	fenor	19.0
-------	-------	------

```
class ActivityAdapter extends ArrayAdapter<Activity>
{
    private Context context;
    public List<Activity> activities;

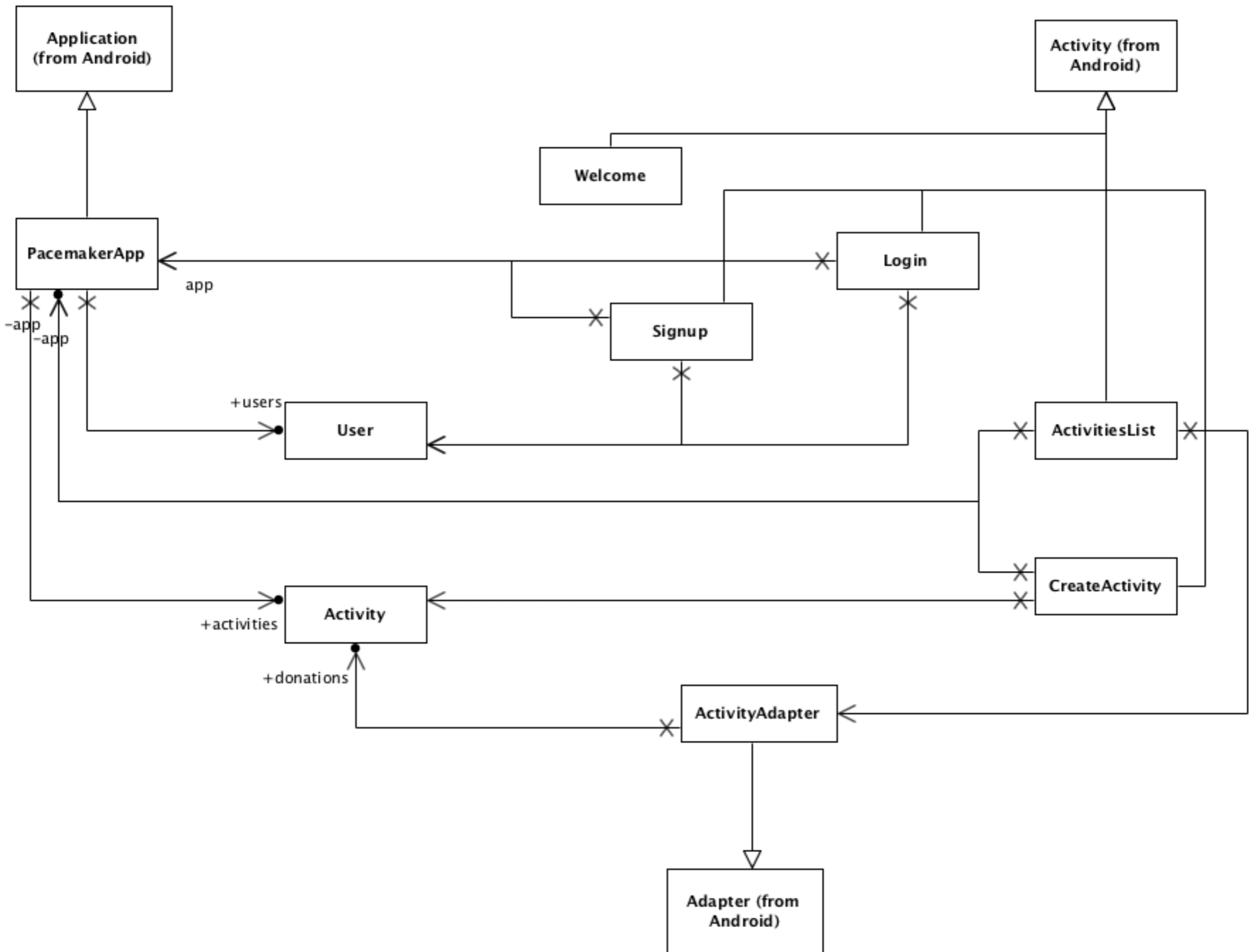
    public ActivityAdapter(Context context, List<Activity> activities)
    {
        super(context, R.layout.activity_row_layout, activities);
        this.context = context;
        this.activities = activities;
    }

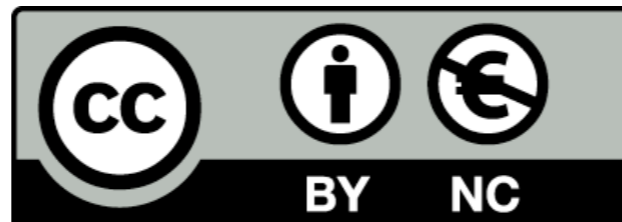
    @Override
    public View getView(int position, View convertView, ViewGroup parent)
    {
        LayoutInflater inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);

        View view = inflater.inflate(R.layout.activity_row_layout, parent, false);
        Activity activity = activities.get(position);
        TextView type = (TextView) view.findViewById(R.id.type);
        TextView location = (TextView) view.findViewById(R.id.location);
        TextView distance = (TextView) view.findViewById(R.id.distance);

        type.setText(activity.type);
        location.setText(activity.location);
        distance.setText(" " + activity.distance);
        return view;
    }

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





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/>

