Mobile Application Development

Produced by

Eamonn de Leastar (edeleastar@wit.ie)
Dr. Siobhán Drohan (sdrohan@wit.ie)

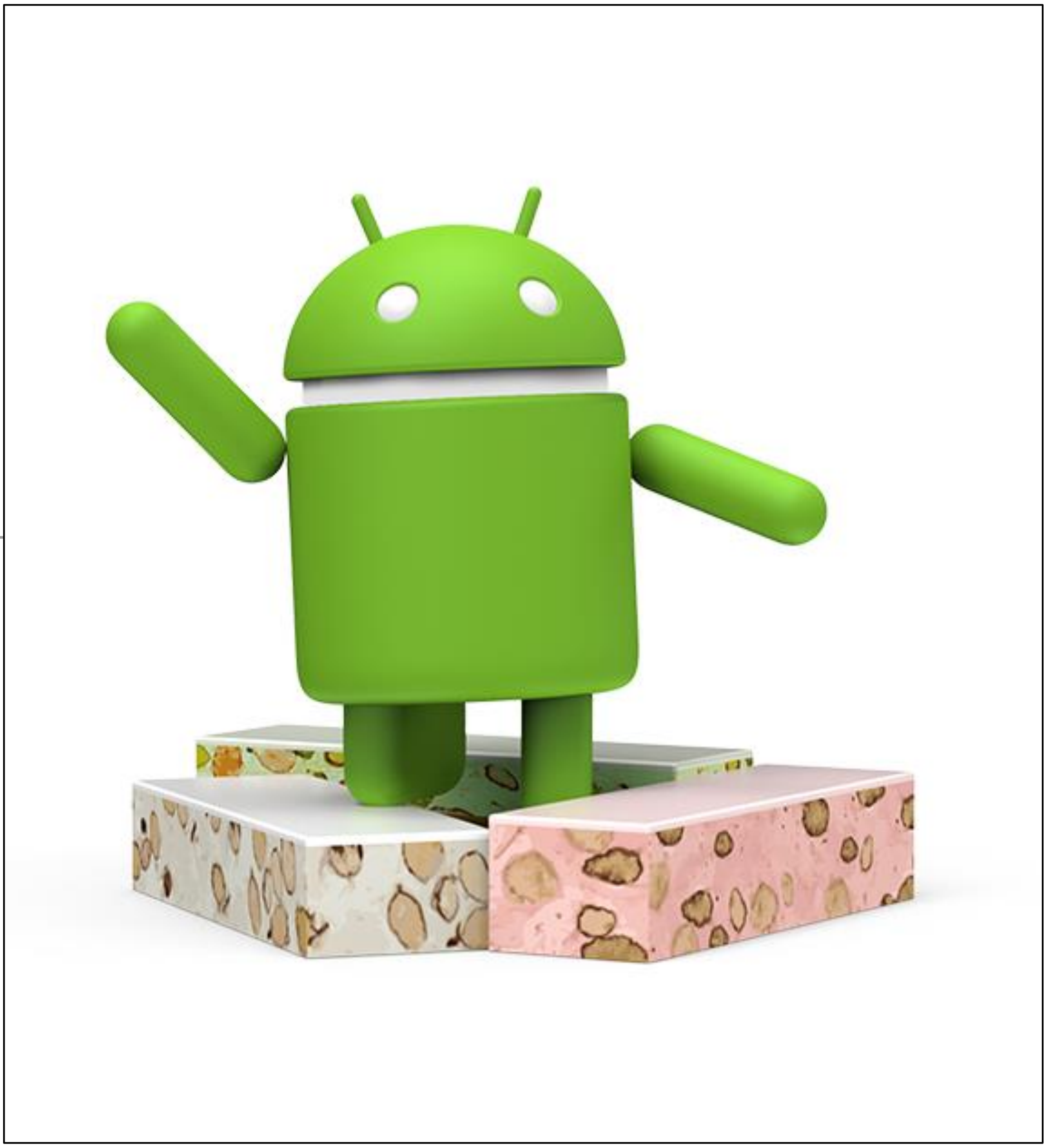
Department of Computing, Maths & Physics
Waterford Institute of Technology

http://www.wit.ie

http://elearning.wit.ie

Second Android Application

MyRent V02

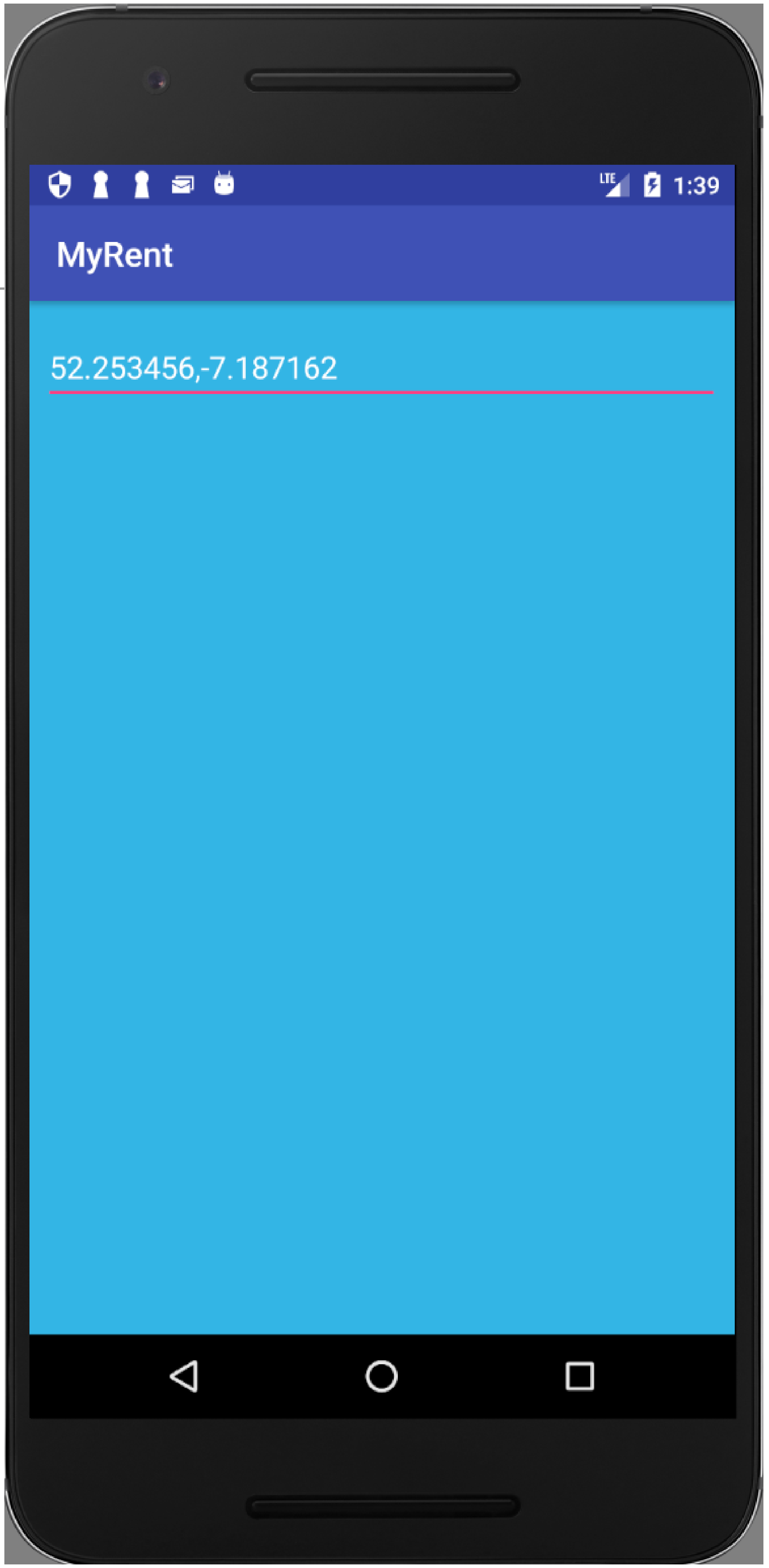


MyRent Versions (week 3 – 5 inclusive)

~~MyRent V00~~	One Activity with a simple TextField where user enters two coordinates, separated by comma.
~~MyRent V01~~	One Activity with multiple widgets to display Location, Status and Rented?
MyRent V02	**Two Activities utilising a '*Master-Detail*' pattern. *Master* holds a list of rented residences whereas the *Detail* is the Activity from V01.**
MyRent V03	Significant update to include an Action Bar, allowing new residence creation and navigation to existing ones. Also includes a Date Picker Dialog that can add a Registration Date for the residence.
MyRent V04	Allows Residences to save to and load from a file. Contents are loaded on launch and saved automatically as Residences are added / updated.
MyRent V05	Evolution of the App Navigation to provide navigation from the Activity back to the List of Activities.
MyRent V06	Enable app to select a contact from the phone's contact list and send an email to the selected user. Requesting permissions is included here.
MyRent V07	Use of Fragments to create a multi-pane screen, which can later support landscape orientations of our app.

MyRent V00

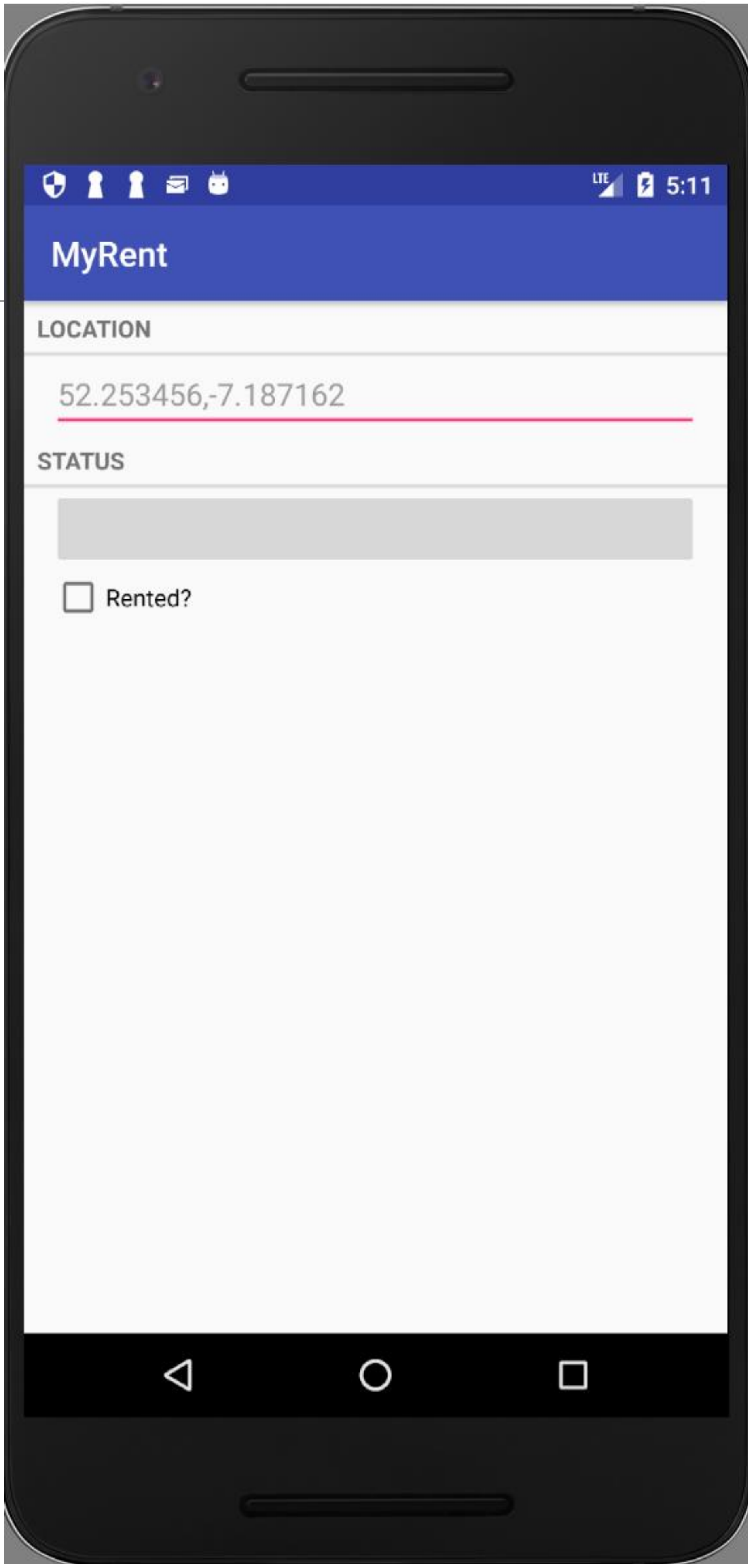
- Simple TextField
- User enters two coordinates, separated by comma



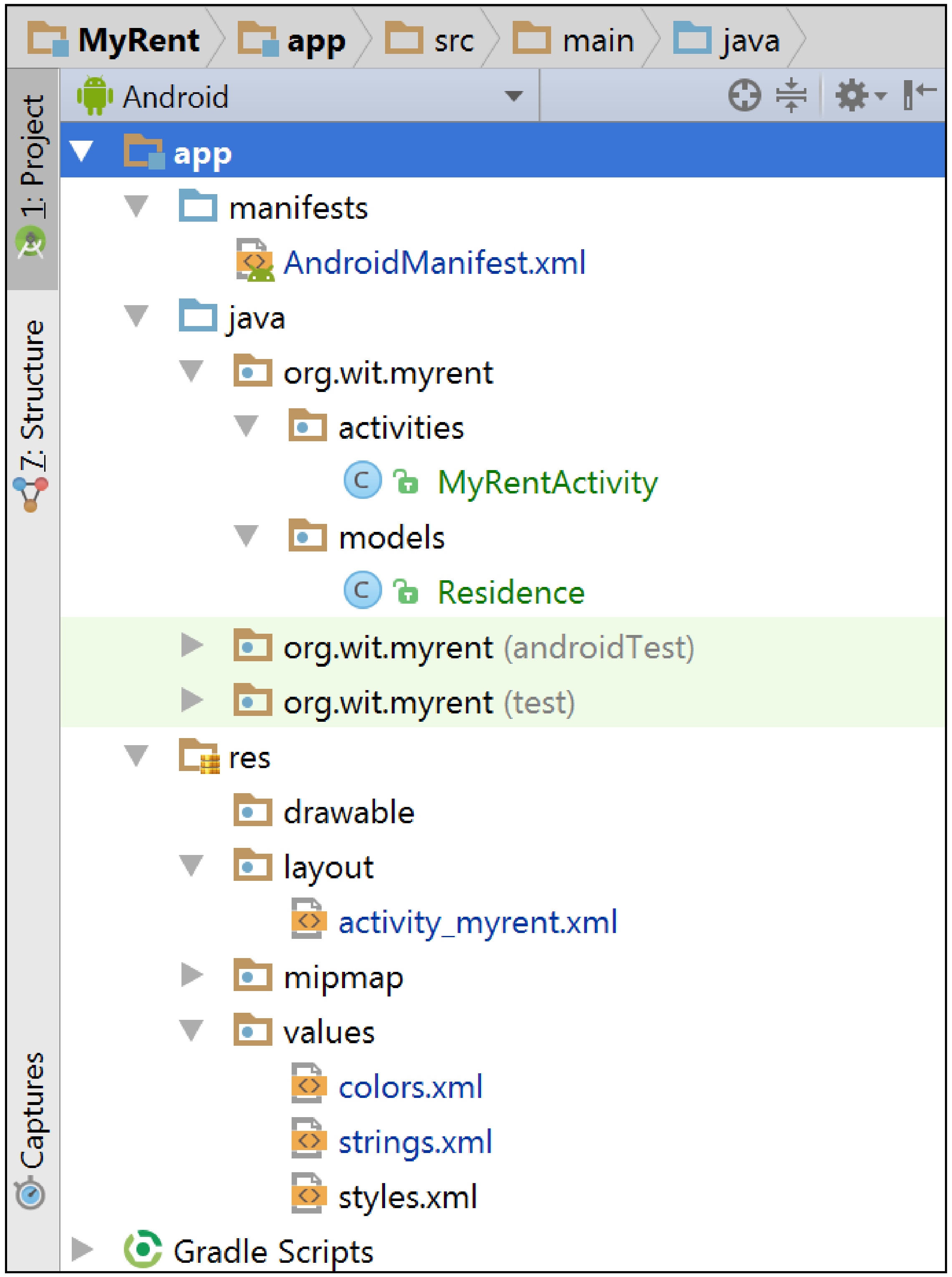
52.253456,-7.187162

MyRent V01

- One Activity with multiple widgets to display:
  - Location (with hint text)
  - Status
  - Rented?



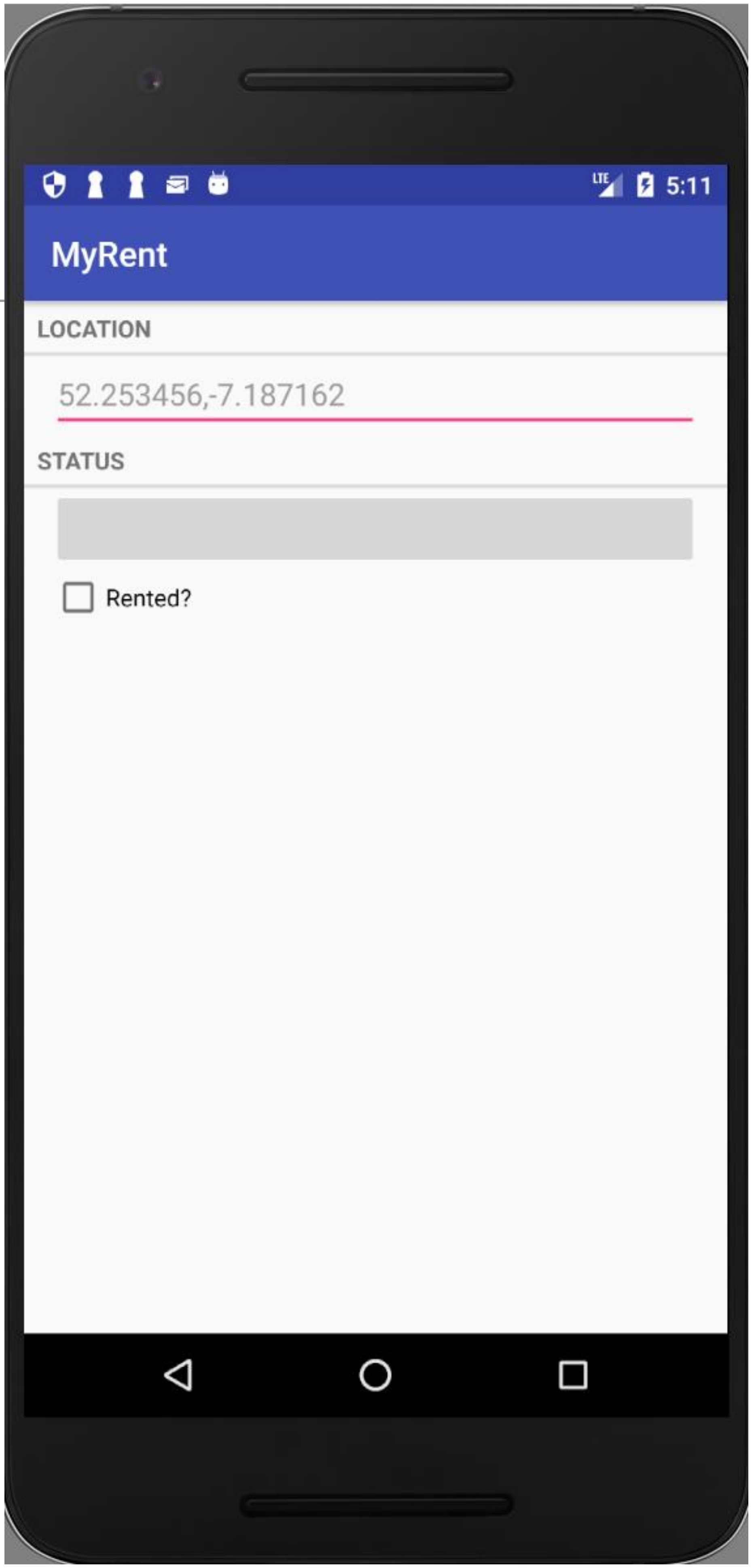
MyRent V01



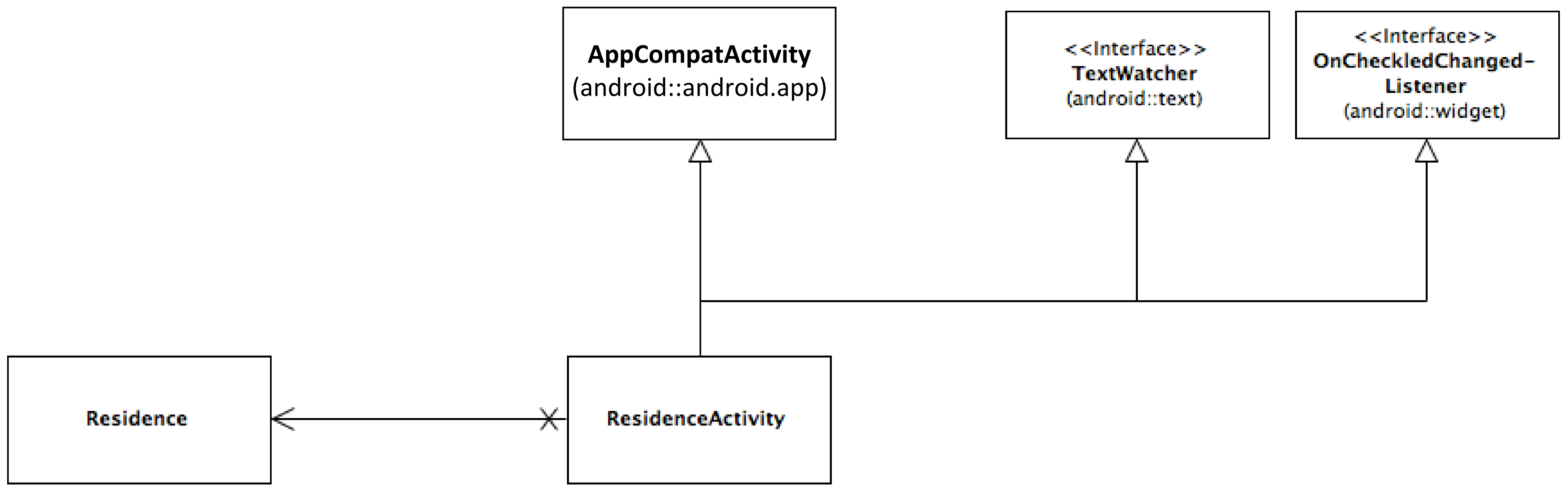
MyRent > app > src > main > java

Android

- app
  - manifests
    - AndroidManifest.xml
  - java
    - org.wit.myrent
      - activities
        - MyRentActivity
      - models
        - Residence
    - org.wit.myrent (androidTest)
    - org.wit.myrent (test)
  - res
    - drawable
    - layout
      - activity_myrent.xml
    - mipmap
    - values
      - colors.xml
      - strings.xml
      - styles.xml
- Gradle Scripts

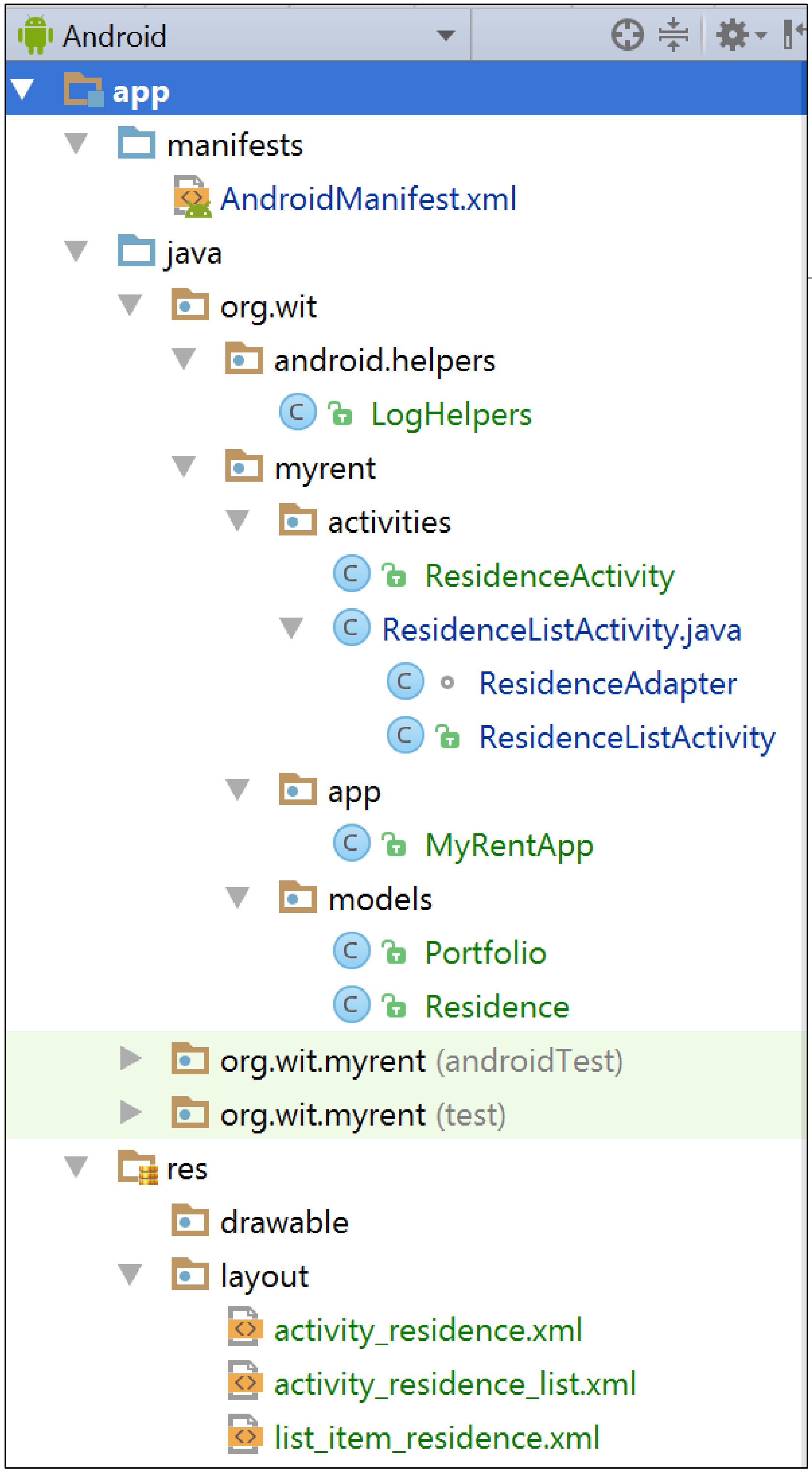


MyRent V01 UML

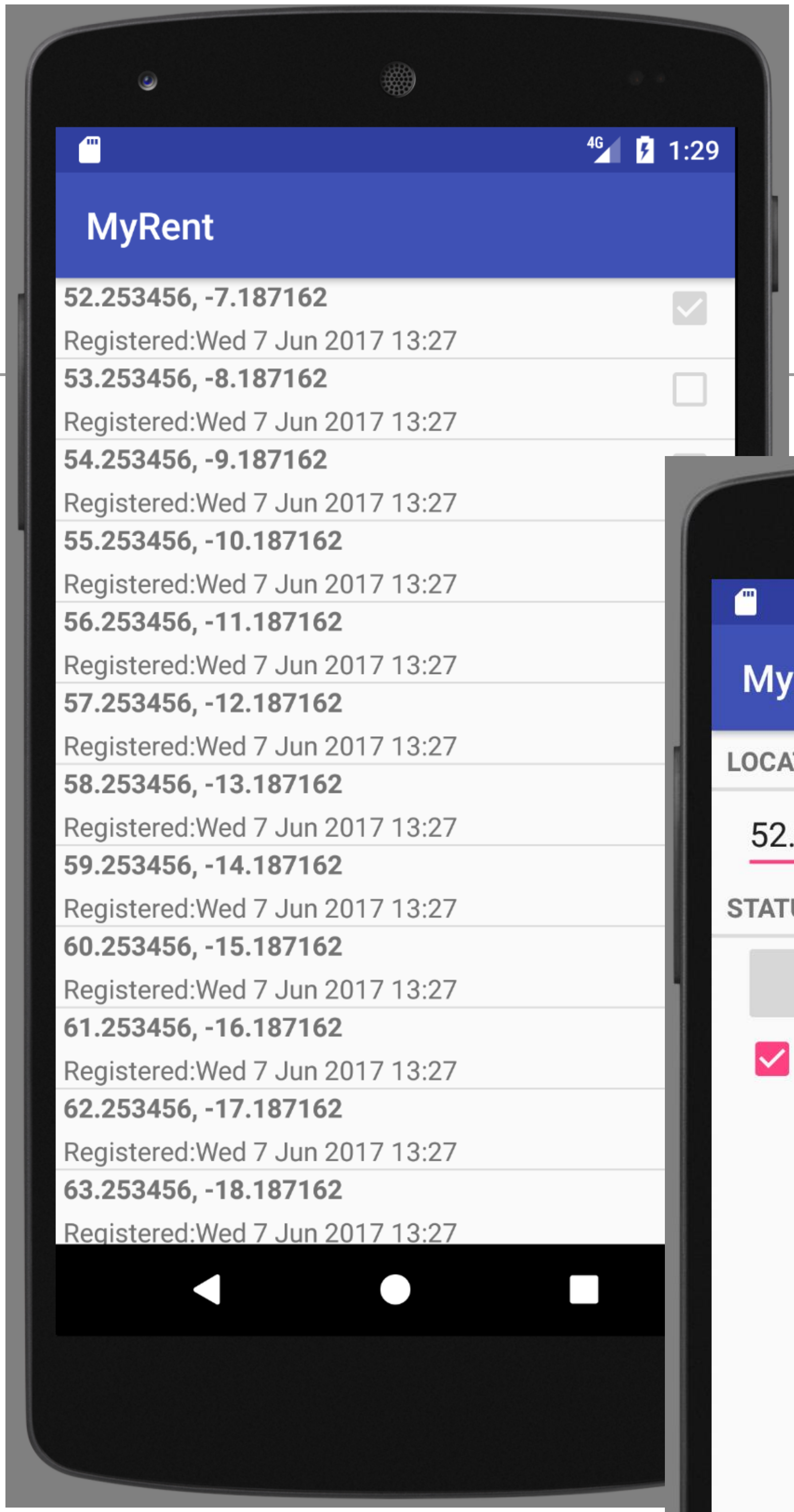


MyRent V02

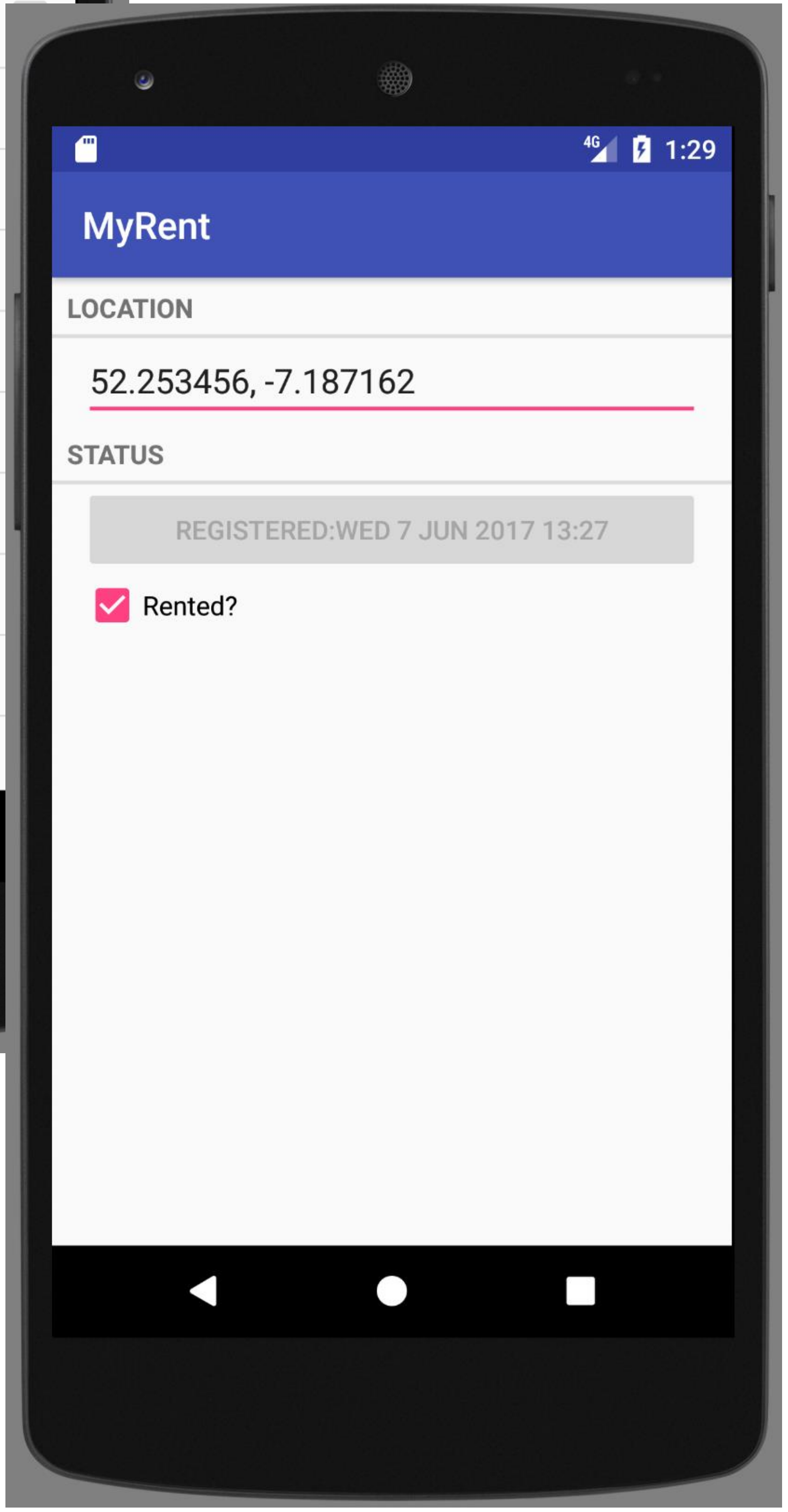
MyRent V02



```
Android
▼ app
  ▼ manifests
      AndroidManifest.xml
  ▼ java
    ▼ org.wit
      ▼ android.helpers
          LogHelpers
      ▼ myrent
        ▼ activities
            ResidenceActivity
          ▼ ResidenceListActivity.java
              ResidenceAdapter
              ResidenceListActivity
        ▼ app
            MyRentApp
        ▼ models
            Portfolio
            Residence
    ▶ org.wit.myrent (androidTest)
    ▶ org.wit.myrent (test)
  ▼ res
      drawable
    ▼ layout
        activity_residence.xml
        activity_residence_list.xml
        list_item_residence.xml
```

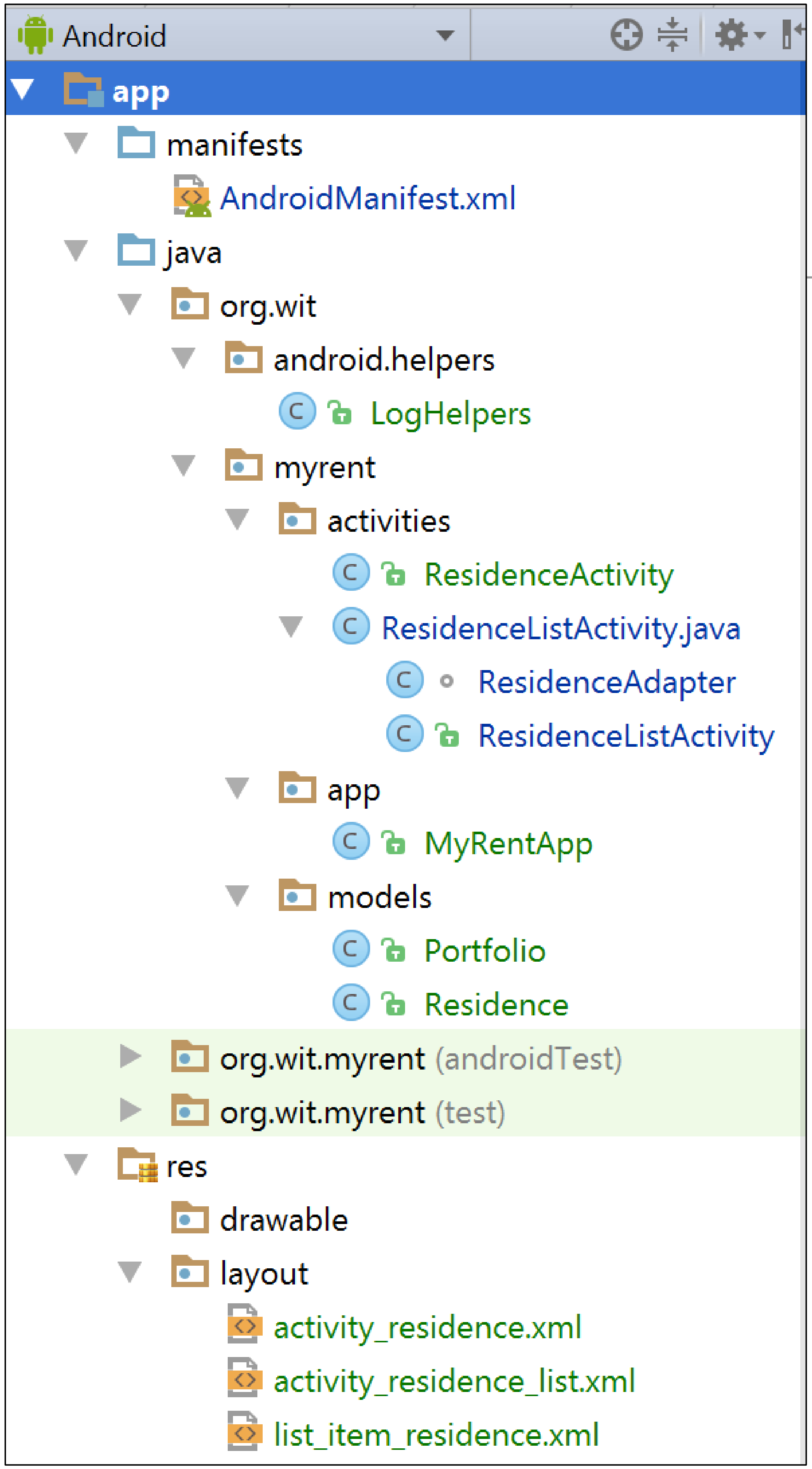


```
1:29
MyRent
52.253456, -7.187162
Registered:Wed 7 Jun 2017 13:27
53.253456, -8.187162
Registered:Wed 7 Jun 2017 13:27
54.253456, -9.187162
Registered:Wed 7 Jun 2017 13:27
55.253456, -10.187162
Registered:Wed 7 Jun 2017 13:27
56.253456, -11.187162
Registered:Wed 7 Jun 2017 13:27
57.253456, -12.187162
Registered:Wed 7 Jun 2017 13:27
58.253456, -13.187162
Registered:Wed 7 Jun 2017 13:27
59.253456, -14.187162
Registered:Wed 7 Jun 2017 13:27
60.253456, -15.187162
Registered:Wed 7 Jun 2017 13:27
61.253456, -16.187162
Registered:Wed 7 Jun 2017 13:27
62.253456, -17.187162
Registered:Wed 7 Jun 2017 13:27
63.253456, -18.187162
Registered:Wed 7 Jun 2017 13:27
```



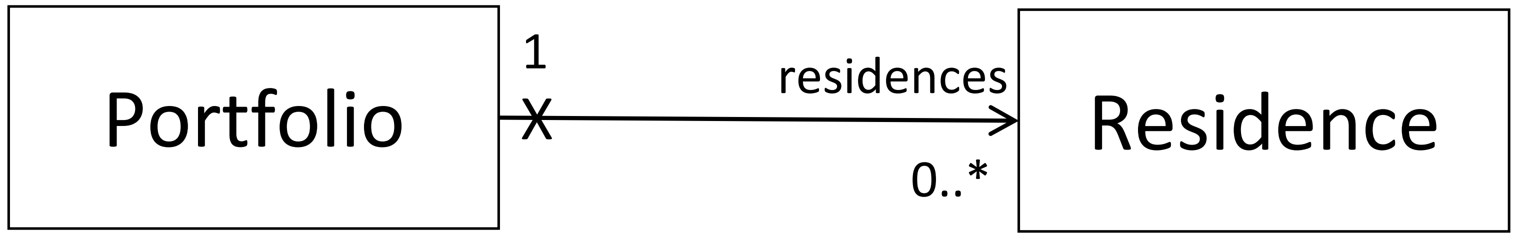
```
1:29
MyRent
LOCATION
52.253456, -7.187162
STATUS
REGISTERED:WED 7 JUN 2017 13:27
☑ Rented?
```

Model classes



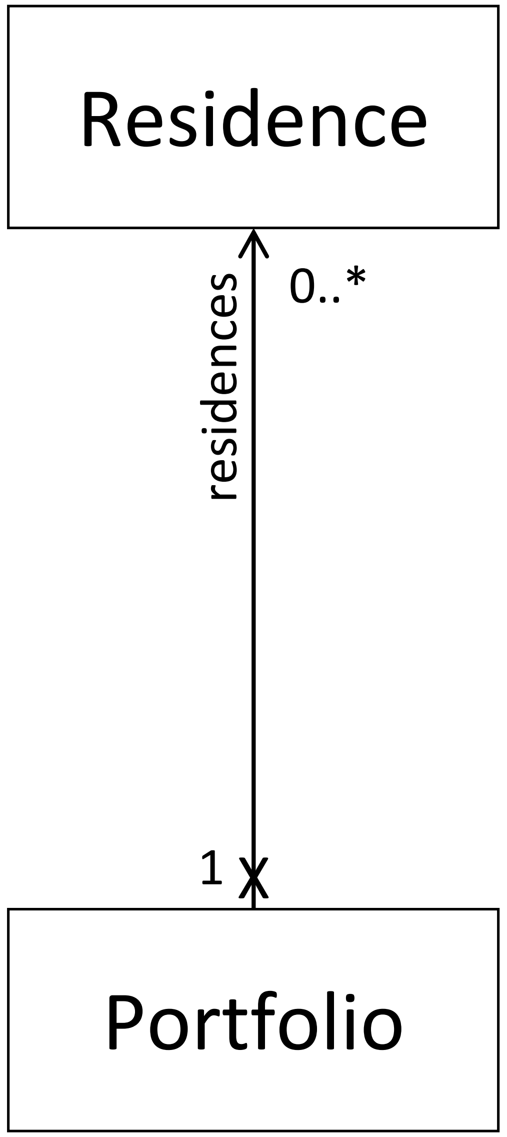
Android

- app
  - manifests
    - AndroidManifest.xml
  - java
    - org.wit
      - android.helpers
        - LogHelpers
      - myrent
        - activities
          - ResidenceActivity
          - ResidenceListActivity.java
            - ResidenceAdapter
            - ResidenceListActivity
        - app
          - MyRentApp
        - models
          - Portfolio
          - Residence
    - org.wit.myrent (androidTest)
    - org.wit.myrent (test)
  - res
    - drawable
    - layout
      - activity_residence.xml
      - activity_residence_list.xml
      - list_item_residence.xml



Residence Model

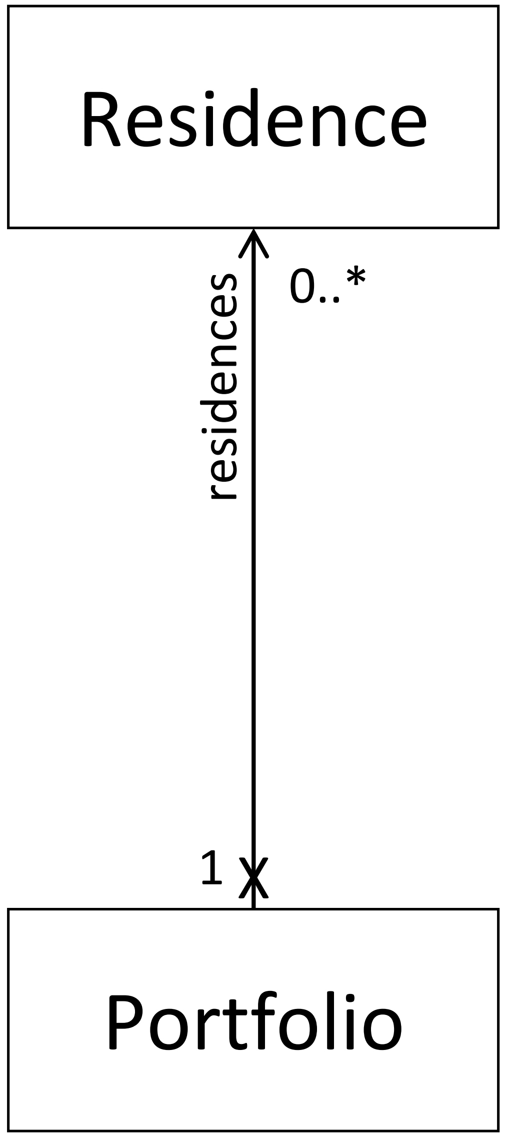
```
public class Residence
{
    public Long id;
    public Long date;

    public String geolocation;
    public boolean rented;

    public Residence(){
        id = unsignedLong();
        date = new Date().getTime();
    }

    private Long unsignedLong() {
        long rndVal = 0;
        do {
            rndVal = new Random().nextLong();
        } while (rndVal <= 0);
        return rndVal;
    }

    public void setGeolocation(String geolocation){
        this.geolocation = geolocation;
    }

    public String getGeolocation(){
        return geolocation;
    }

    public String getDateString() {
        return "Registered:" + dateString();
    }

    private String dateString() {
        String dateFormat = "EEE d MMM yyyy H:mm";
        return android.text.format.DateFormat.format(dateFormat, date).toString();
    }
}
```



Portfolio Model

```java
import java.util.ArrayList;
import android.util.Log;

public class Portfolio
{
    public ArrayList<Residence> residences;

    public Portfolio() {
        residences = new ArrayList<Residence>();
        this.generateTestData();
    }

    public void addResidence(Residence residence) {
        residences.add(residence);
    }

    public Residence getResidence(Long id) {
        Log.i(this.getClass().getSimpleName(), "Long parameter id: " + id);

        for (Residence res : residences) {
            if (id.equals(res.id)) {
                return res;
            }
        }
        return null;
    }

    private void generateTestData() {
        for (int i = 0; i < 100; i += 1) {
            Residence r = new Residence();
            r.geolocation = (52.253456 + i) % 90 + ", " + (-7.187162 - i) % 180 + "";
            if (i % 2 == 0) {
                r.rented = true;
            }
            else {
                r.rented = false;
            }
            residences.add(r);
        }
    }
}
```

Generate random Residence objects to exercise UI



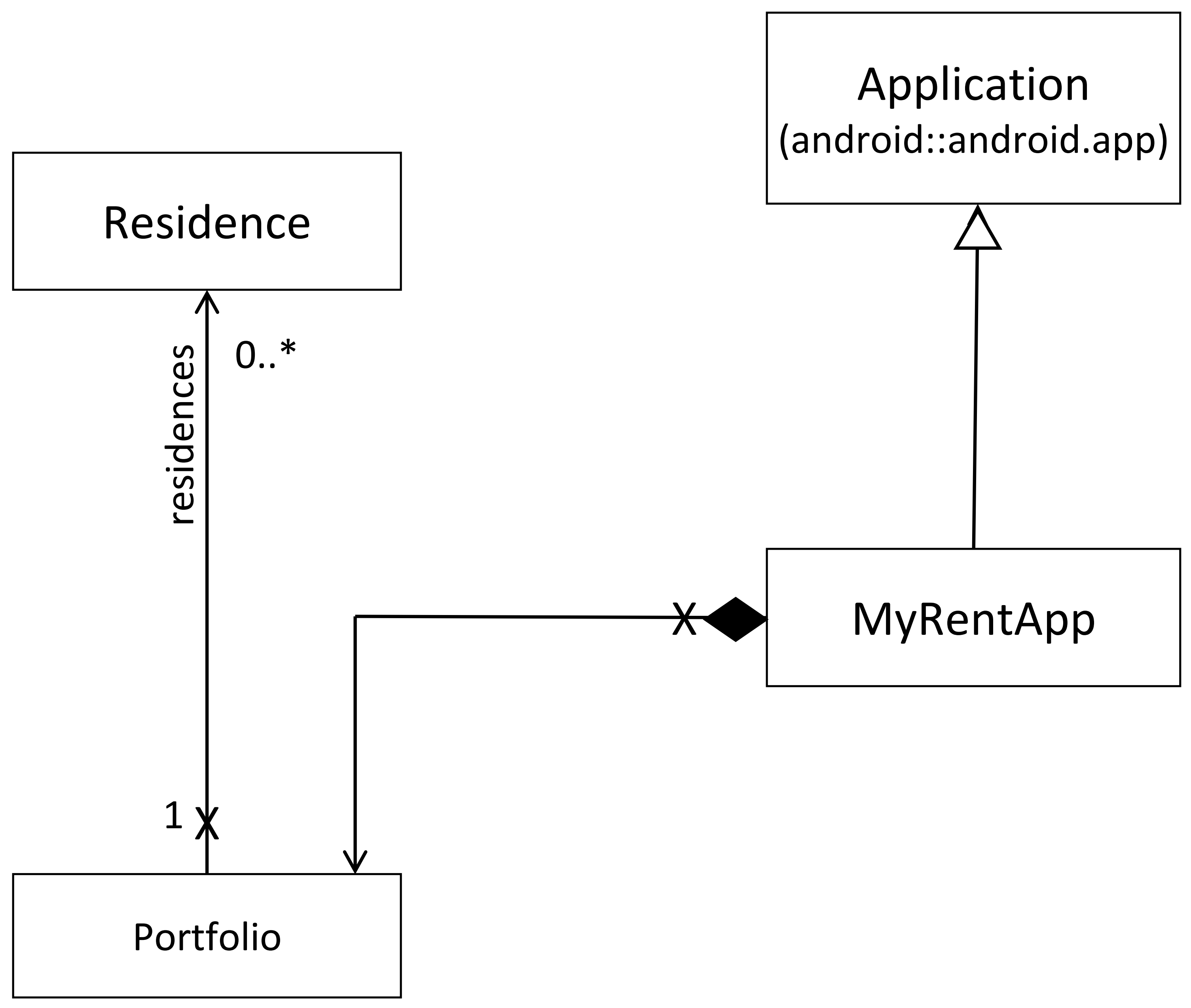
Application Object

Recall that the Application object is created when the app is launched; we are guaranteed there will only ever be one of them!

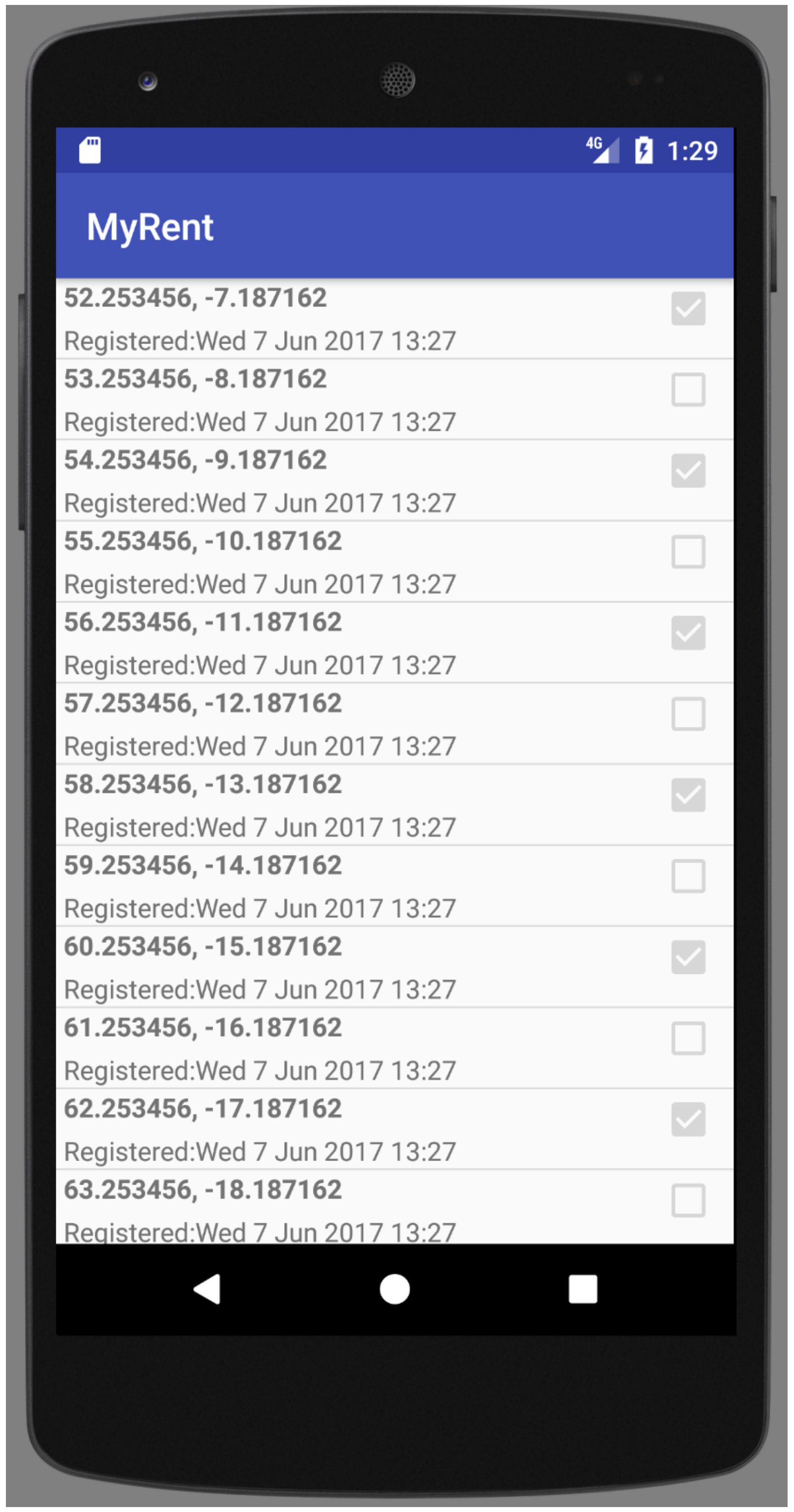
```java
package org.wit.myrent.app;

import org.wit.myrent.models.Portfolio;
import android.app.Application;
import static org.wit.android.helpers.LogHelpers.info;

public class MyRentApp extends Application
{
    public Portfolio portfolio;

    @Override
    public void onCreate()
    {
        super.onCreate();
        portfolio = new Portfolio();

        info(this, "MyRent app launched");
    }
}
```

We use the application object to manage the list of residences i.e. an object of Portfolio.

V2.0, UML so far…

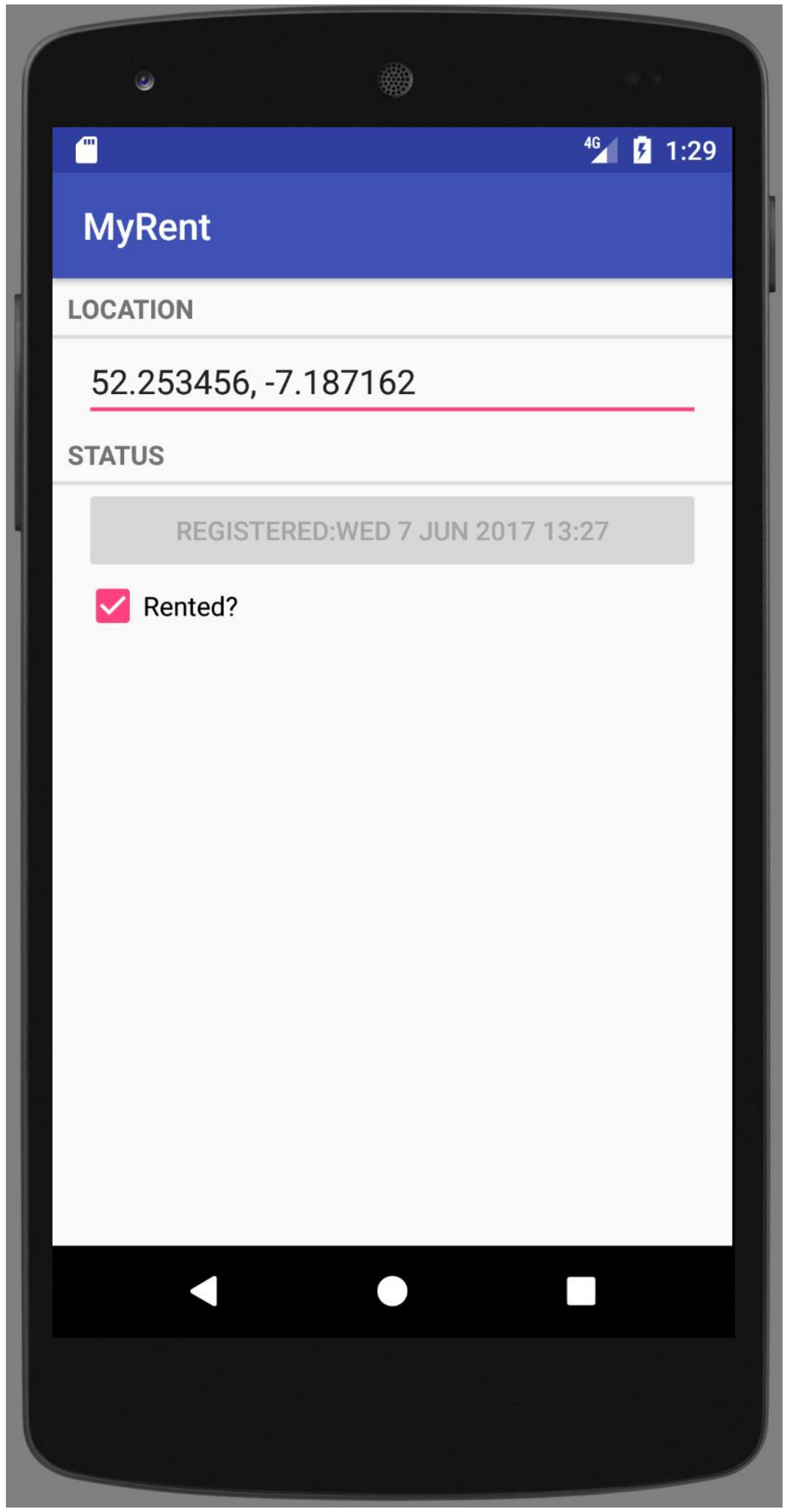


Adding a new new list activity for Residences….


1:29

MyRent

52.253456, -7.187162 ✓
Registered:Wed 7 Jun 2017 13:27
53.253456, -8.187162 ☐
Registered:Wed 7 Jun 2017 13:27
54.253456, -9.187162 ✓
Registered:Wed 7 Jun 2017 13:27
55.253456, -10.187162 ☐
Registered:Wed 7 Jun 2017 13:27
56.253456, -11.187162 ✓
Registered:Wed 7 Jun 2017 13:27
57.253456, -12.187162 ☐
Registered:Wed 7 Jun 2017 13:27
58.253456, -13.187162 ✓
Registered:Wed 7 Jun 2017 13:27
59.253456, -14.187162 ☐
Registered:Wed 7 Jun 2017 13:27
60.253456, -15.187162 ✓
Registered:Wed 7 Jun 2017 13:27
61.253456, -16.187162 ☐
Registered:Wed 7 Jun 2017 13:27
62.253456, -17.187162 ✓
Registered:Wed 7 Jun 2017 13:27
63.253456, -18.187162 ☐
Registered:Wed 7 Jun 2017 13:27


1:29

MyRent

LOCATION

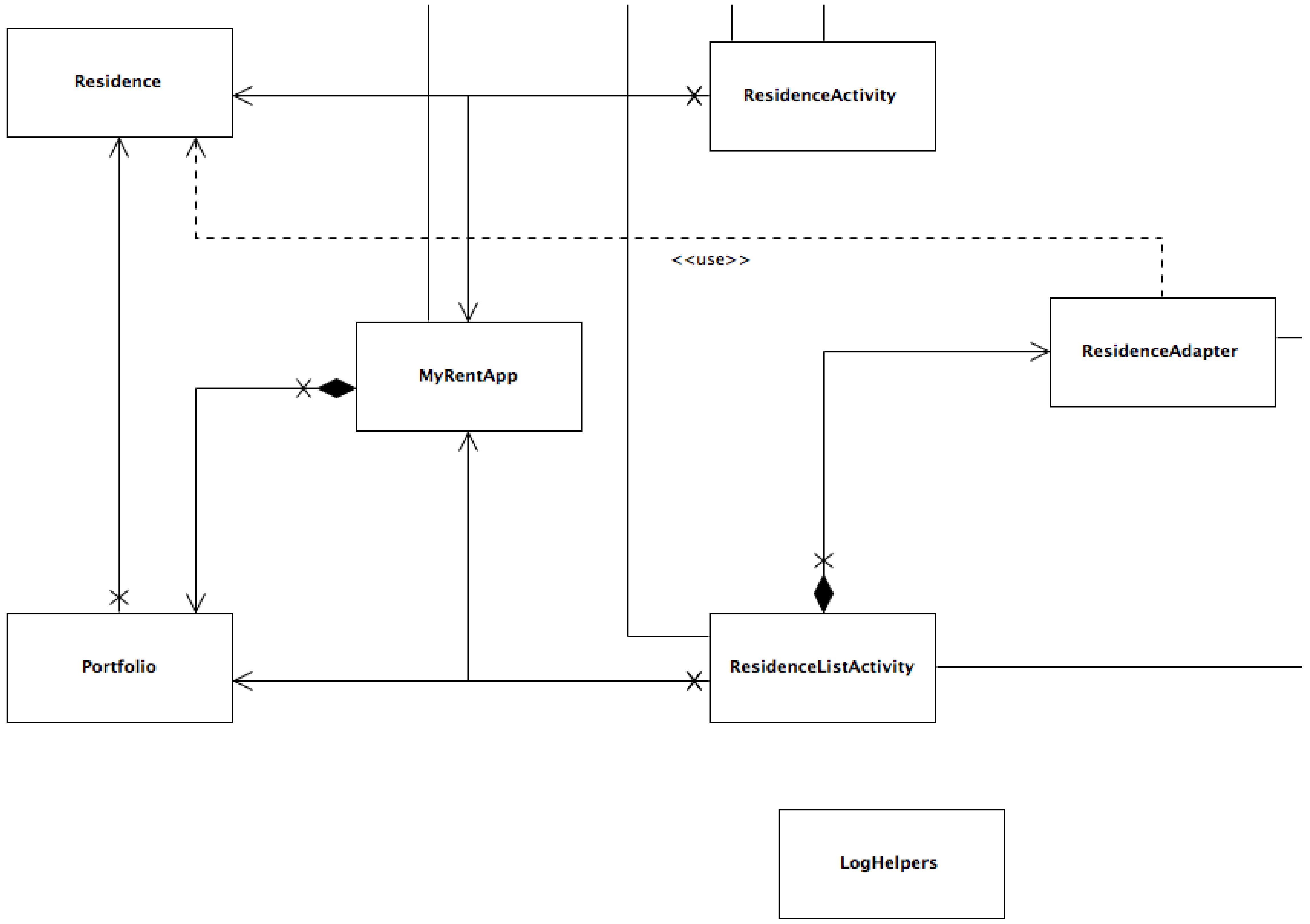
52.253456, -7.187162

STATUS

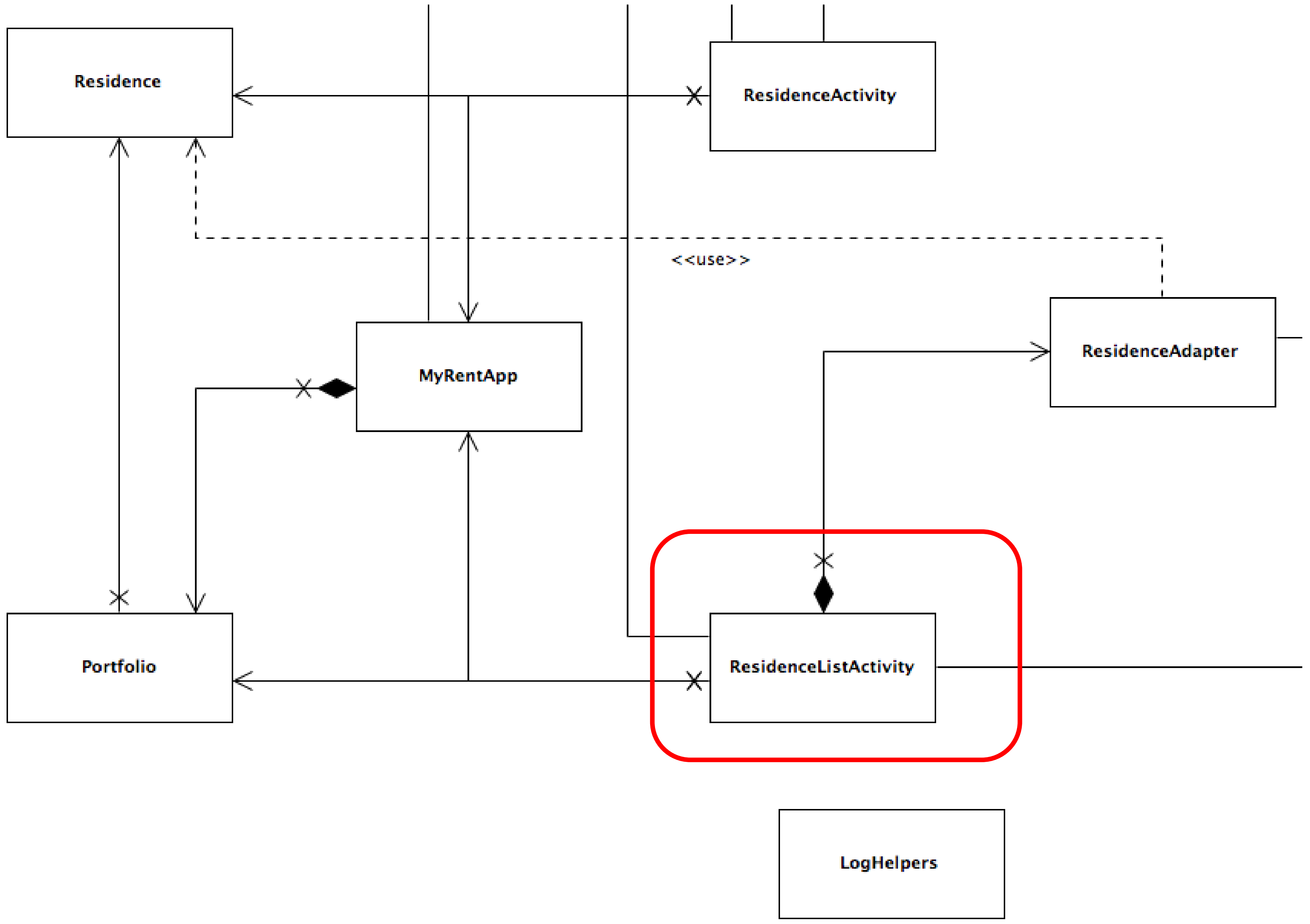
REGISTERED:WED 7 JUN 2017 13:27

✓ Rented?

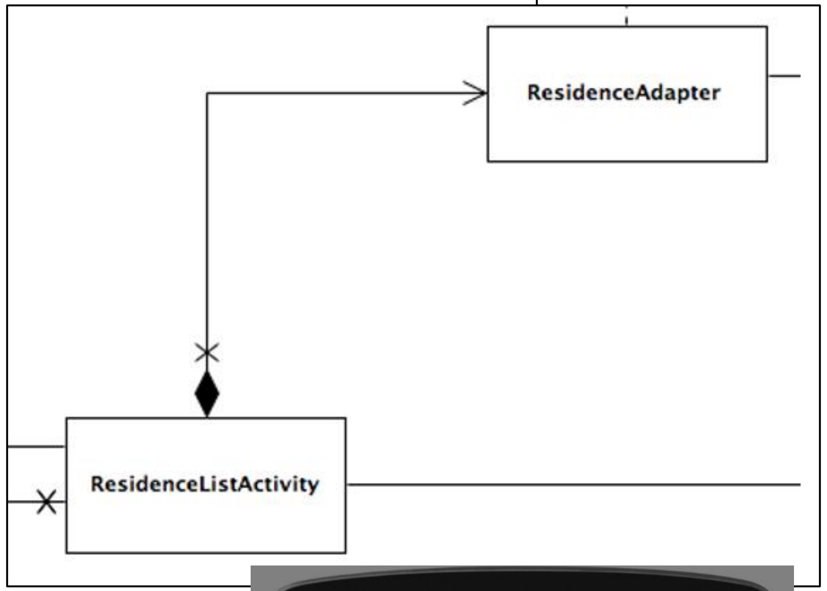
V2.0 – UML with ResidenceListActivity (and the associated adapter)

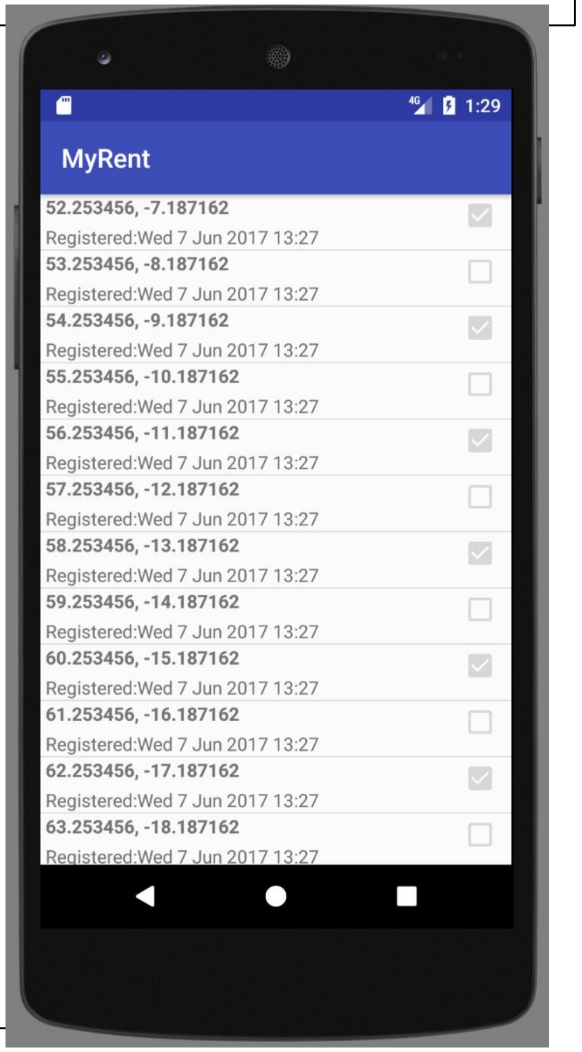


V2.0 – UML with ResidenceListActivity (and the associated adapter)



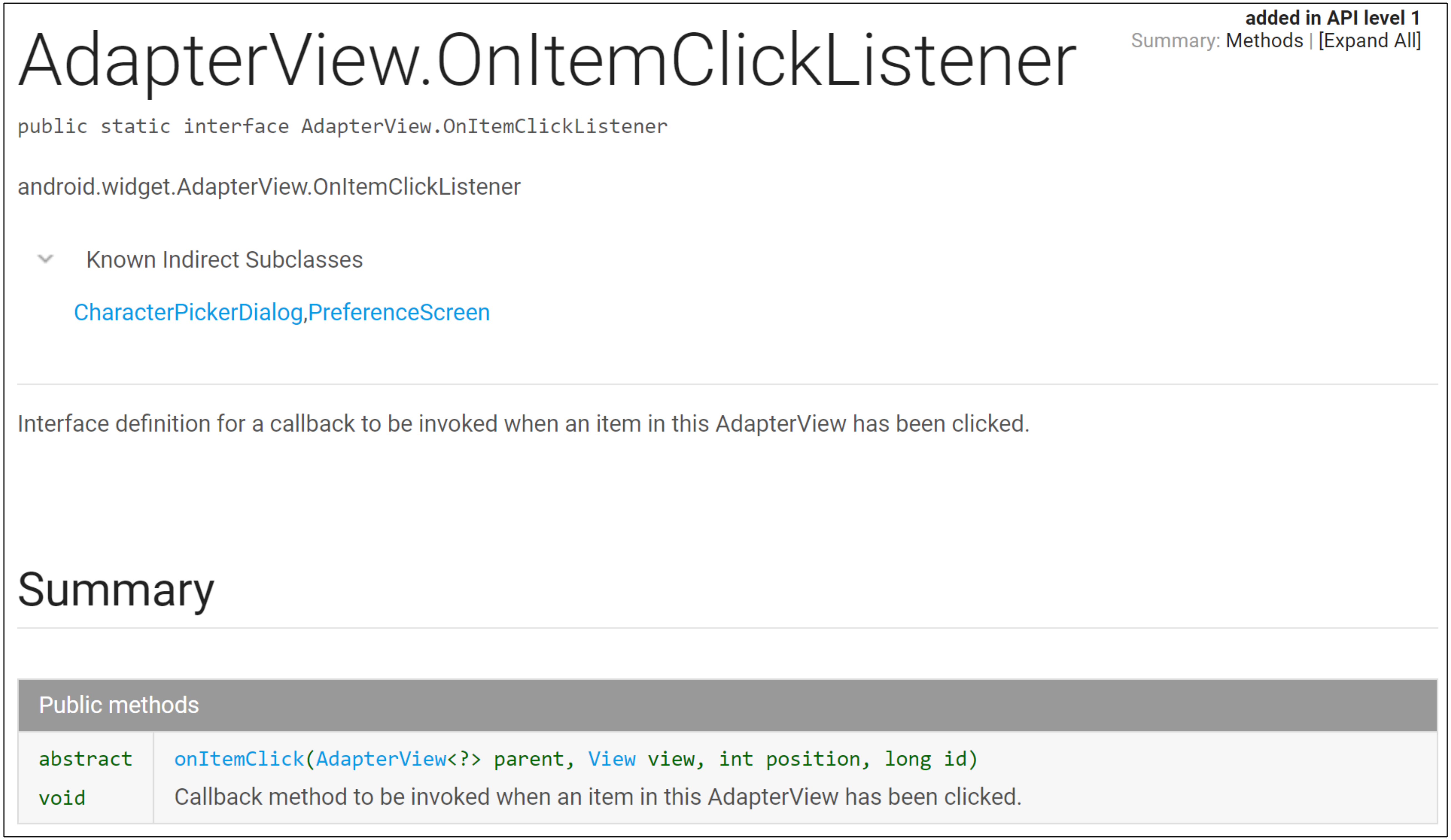
```java
public class ResidenceListActivity extends AppCompatActivity implements AdapterView.OnItemClickListener
{
    private ListView listView;
    private Portfolio portfolio;
    private ResidenceAdapter adapter;

    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setTitle(R.string.app_name);
        setContentView(R.layout.activity_residence_list);

        listView = (ListView) findViewById(R.id.residenceList);
        MyRentApp app = (MyRentApp) getApplication();
        portfolio = app.portfolio;

        adapter = new ResidenceAdapter(this, portfolio.residences);
        listView.setAdapter(adapter);
        listView.setOnItemClickListener(this);
    }

    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        Residence residence = adapter.getItem(position);
        Intent intent = new Intent(this, ResidenceActivity.class);
        intent.putExtra("RESIDENCE_ID", residence.id);
        startActivity(intent);
    }

    @Override
    public void onResume()
    {
        super.onResume();
        adapter.notifyDataSetChanged();
    }
}
```





MyRent

52.253456, -7.187162
Registered:Wed 7 Jun 2017 13:27
53.253456, -8.187162
Registered:Wed 7 Jun 2017 13:27
54.253456, -9.187162
Registered:Wed 7 Jun 2017 13:27
55.253456, -10.187162
Registered:Wed 7 Jun 2017 13:27
56.253456, -11.187162
Registered:Wed 7 Jun 2017 13:27
57.253456, -12.187162
Registered:Wed 7 Jun 2017 13:27
58.253456, -13.187162
Registered:Wed 7 Jun 2017 13:27
59.253456, -14.187162
Registered:Wed 7 Jun 2017 13:27
60.253456, -15.187162
Registered:Wed 7 Jun 2017 13:27
61.253456, -16.187162
Registered:Wed 7 Jun 2017 13:27
62.253456, -17.187162
Registered:Wed 7 Jun 2017 13:27
63.253456, -18.187162
Registered:Wed 7 Jun 2017 13:27


added in API level 1
Summary: Methods | [Expand All]

AdapterView.OnItemClickListener

```
public static interface AdapterView.OnItemClickListener
```

android.widget.AdapterView.OnItemClickListener

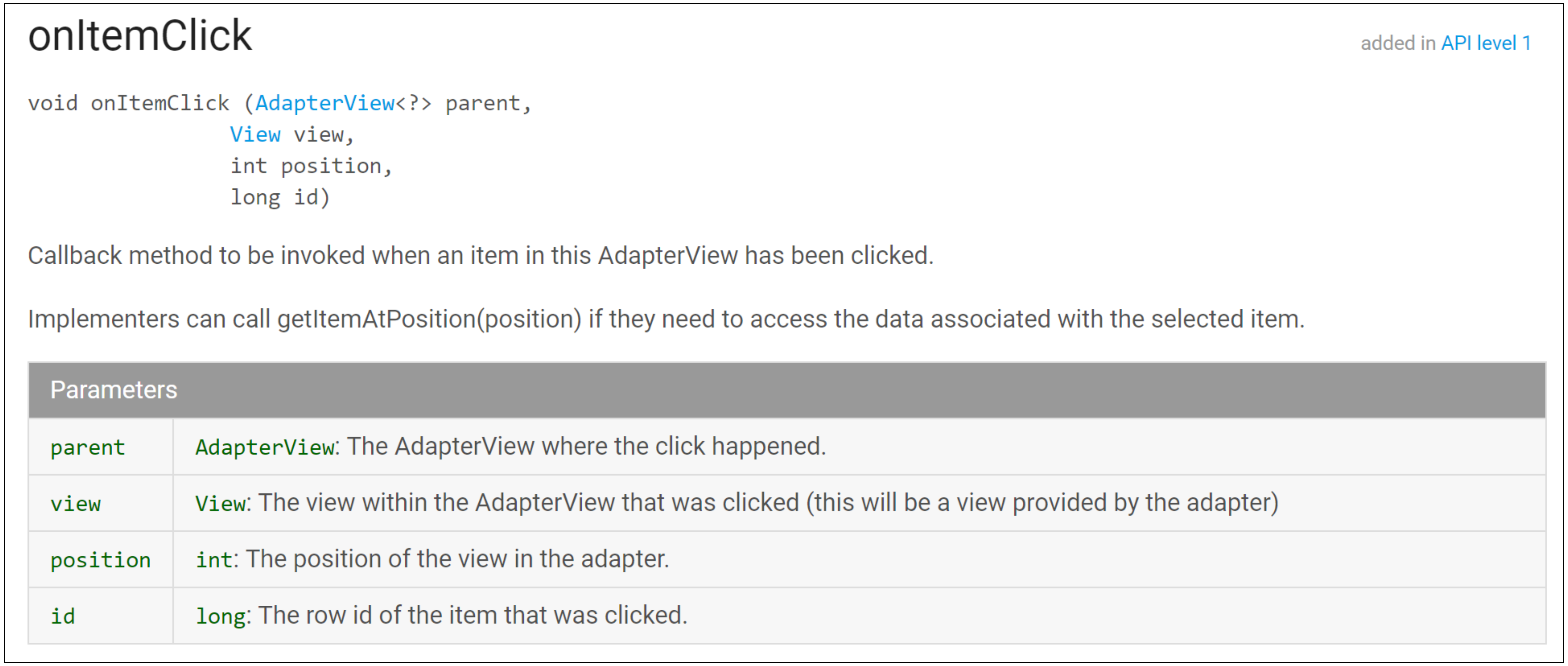
Known Indirect Subclasses

CharacterPickerDialog,PreferenceScreen

Interface definition for a callback to be invoked when an item in this AdapterView has been clicked.

Summary

Public methods	
`abstract void`	`onItemClick(AdapterView<?> parent, View view, int position, long id)` Callback method to be invoked when an item in this AdapterView has been clicked.



onItemClick

added in API level 1

```
void onItemClick (AdapterView<?> parent,
                View view,
                int position,
                long id)
```

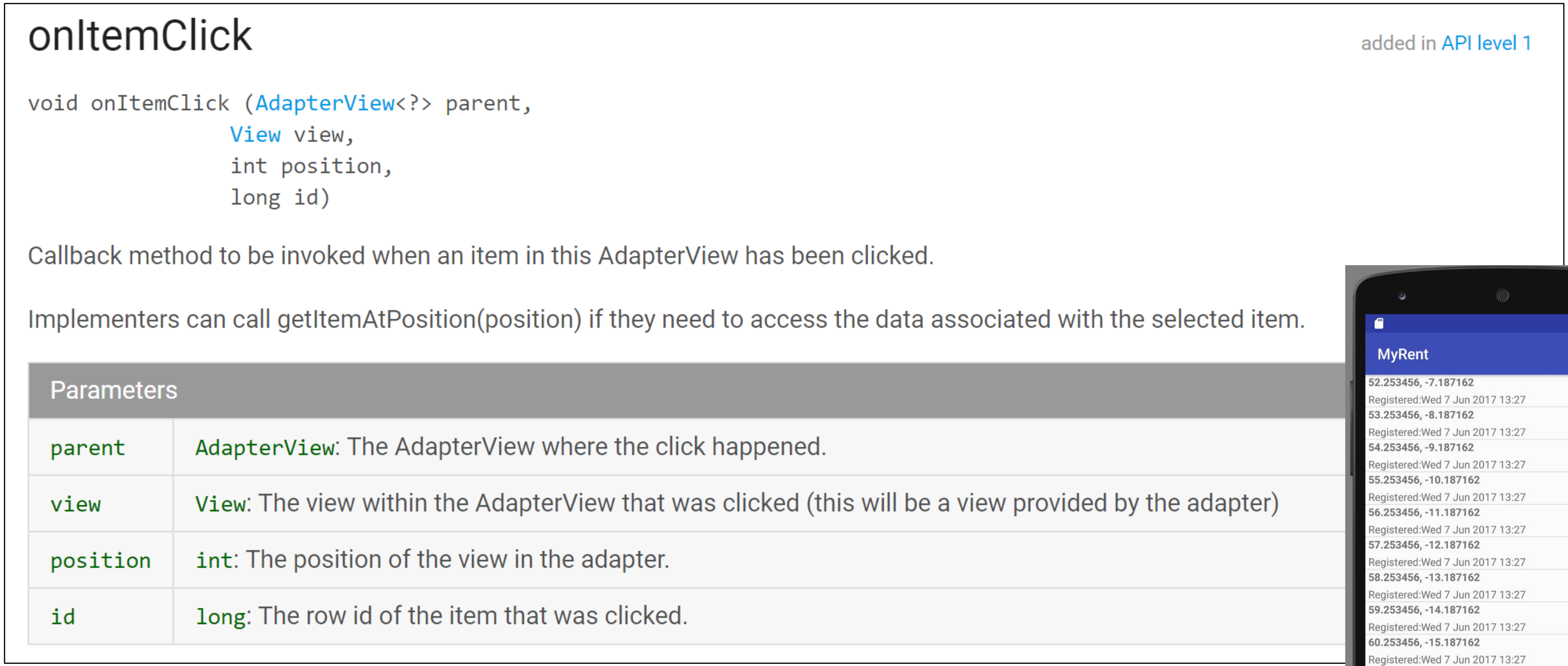
Callback method to be invoked when an item in this AdapterView has been clicked.

Implementers can call getItemAtPosition(position) if they need to access the data associated with the selected item.

Parameters	
`parent`	`AdapterView`: The AdapterView where the click happened.
`view`	`View`: The view within the AdapterView that was clicked (this will be a view provided by the adapter)
`position`	`int`: The position of the view in the adapter.
`id`	`long`: The row id of the item that was clicked.

ResidenceListActivity

```java
@Override
public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
  Residence residence = adapter.getItem(position);
  Intent intent = new Intent(this, ResidenceActivity.class);
  intent.putExtra("RESIDENCE_ID", residence.id);
  startActivity(intent);
}
```



onItemClick

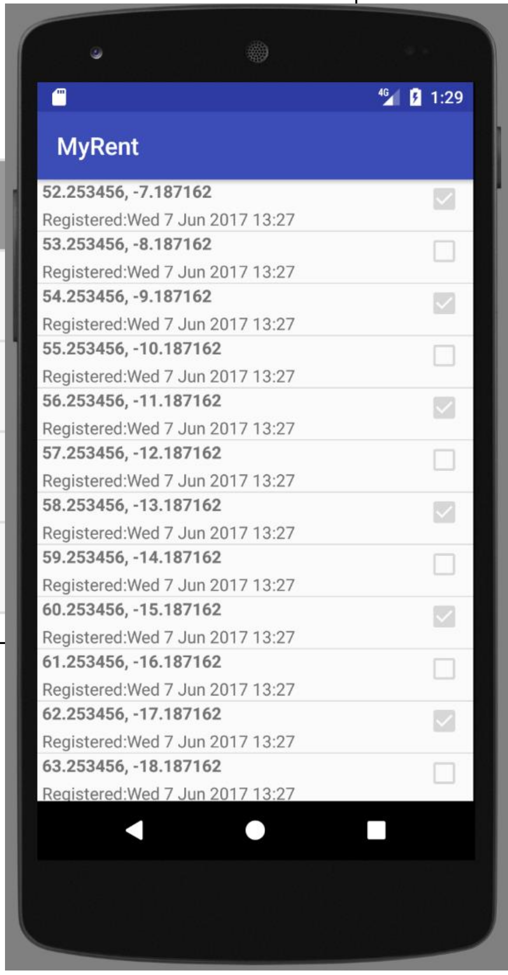
added in API level 1

```
void onItemClick (AdapterView<?> parent,
                View view,
                int position,
                long id)
```

Callback method to be invoked when an item in this AdapterView has been clicked.

Implementers can call getItemAtPosition(position) if they need to access the data associated with the selected item.

Parameters	
`parent`	`AdapterView`: The AdapterView where the click happened.
`view`	`View`: The view within the AdapterView that was clicked (this will be a view provided by the adapter)
`position`	`int`: The position of the view in the adapter.
`id`	`long`: The row id of the item that was clicked.



52.253456, -7.187162 Registered:Wed 7 Jun 2017 13:27	✓
53.253456, -8.187162 Registered:Wed 7 Jun 2017 13:27	☐
54.253456, -9.187162 Registered:Wed 7 Jun 2017 13:27	✓
55.253456, -10.187162 Registered:Wed 7 Jun 2017 13:27	☐
56.253456, -11.187162 Registered:Wed 7 Jun 2017 13:27	✓
57.253456, -12.187162 Registered:Wed 7 Jun 2017 13:27	☐
58.253456, -13.187162 Registered:Wed 7 Jun 2017 13:27	✓
59.253456, -14.187162 Registered:Wed 7 Jun 2017 13:27	☐
60.253456, -15.187162 Registered:Wed 7 Jun 2017 13:27	✓
61.253456, -16.187162 Registered:Wed 7 Jun 2017 13:27	☐
62.253456, -17.187162 Registered:Wed 7 Jun 2017 13:27	✓
63.253456, -18.187162 Registered:Wed 7 Jun 2017 13:27	☐

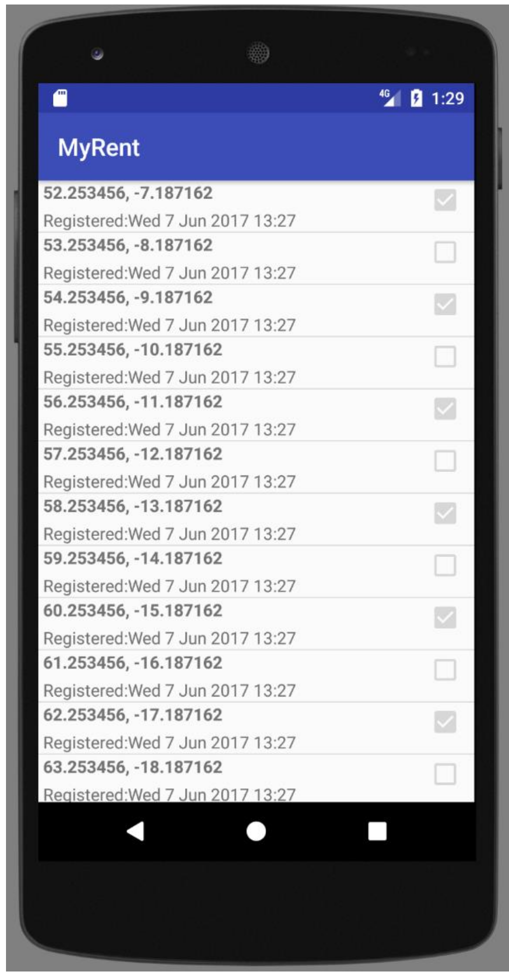
ResidenceListActivity

```java
@Override
public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
  Residence residence = adapter.getItem(position);
  Intent intent = new Intent(this, ResidenceActivity.class);
  intent.putExtra("RESIDENCE_ID", residence.id);
  startActivity(intent);
}
```

1. Retrieve the Residence object by its position in the list

2. Create a new Intent to start ResidenceActivity class.

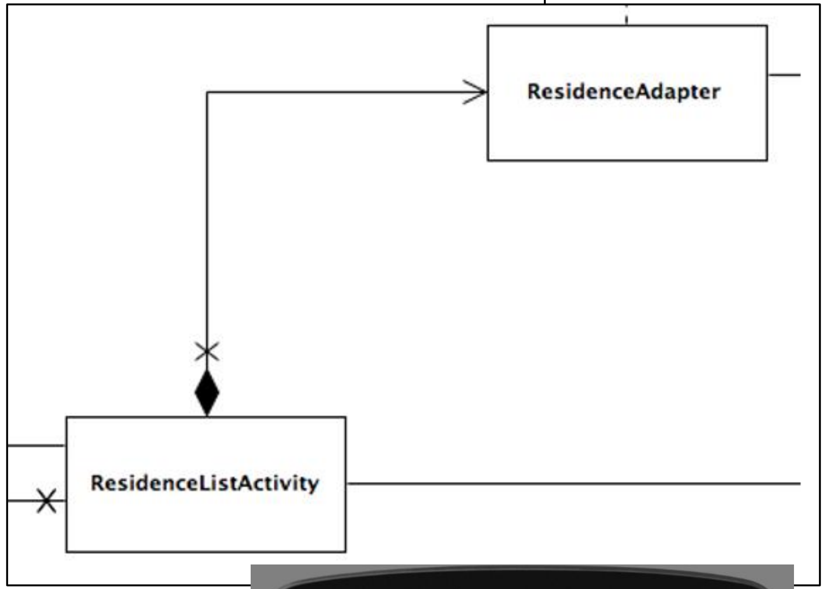
   - Before starting it, put the ID of the object we retrieved into the 'extra' information passed to the intent.

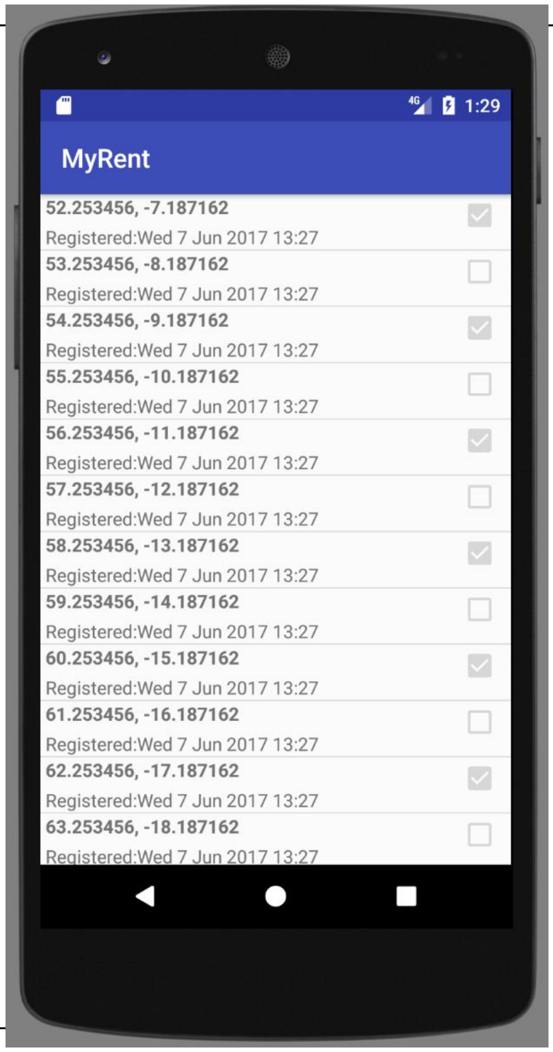


52.253456, -7.187162 ✓
Registered:Wed 7 Jun 2017 13:27
53.253456, -8.187162 ☐
Registered:Wed 7 Jun 2017 13:27
54.253456, -9.187162 ✓
Registered:Wed 7 Jun 2017 13:27
55.253456, -10.187162 ☐
Registered:Wed 7 Jun 2017 13:27
56.253456, -11.187162 ✓
Registered:Wed 7 Jun 2017 13:27
57.253456, -12.187162 ☐
Registered:Wed 7 Jun 2017 13:27
58.253456, -13.187162 ✓
Registered:Wed 7 Jun 2017 13:27
59.253456, -14.187162 ☐
Registered:Wed 7 Jun 2017 13:27
60.253456, -15.187162 ✓
Registered:Wed 7 Jun 2017 13:27
61.253456, -16.187162 ☐
Registered:Wed 7 Jun 2017 13:27
62.253456, -17.187162 ✓
Registered:Wed 7 Jun 2017 13:27
63.253456, -18.187162 ☐
Registered:Wed 7 Jun 2017 13:27

Note: An Intent is a messaging object you can use to request an action from another app component.

```java
public class ResidenceListActivity extends AppCompatActivity implements AdapterView.OnItemClickListener
{
    private ListView listView;
    private Portfolio portfolio;
    private ResidenceAdapter adapter;

    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setTitle(R.string.app_name);
        setContentView(R.layout.activity_residence_list);

        listView = (ListView) findViewById(R.id.residenceList);
        MyRentApp app = (MyRentApp) getApplication();
        portfolio = app.portfolio;

        adapter = new ResidenceAdapter(this, portfolio.residences);
        listView.setAdapter(adapter);
        listView.setOnItemClickListener(this);
    }

    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long id) {
        Residence residence = adapter.getItem(position);
        Intent intent = new Intent(this, ResidenceActivity.class);
        intent.putExtra("RESIDENCE_ID", residence.id);
        startActivity(intent);
    }

    @Override
    public void onResume()
    {
        super.onResume();
        adapter.notifyDataSetChanged();
    }
}
```

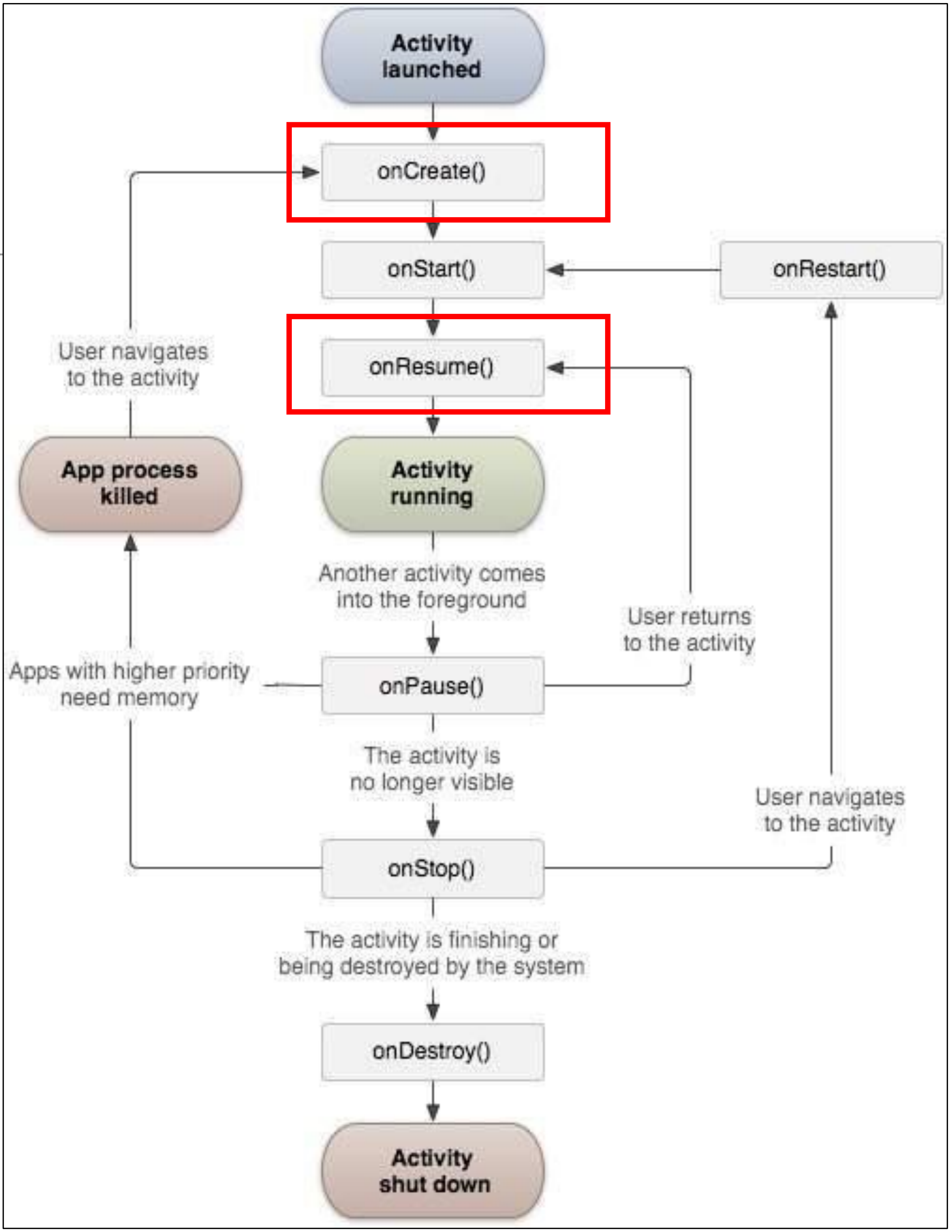




52.253456, -7.187162
Registered:Wed 7 Jun 2017 13:27
53.253456, -8.187162
Registered:Wed 7 Jun 2017 13:27
54.253456, -9.187162
Registered:Wed 7 Jun 2017 13:27
55.253456, -10.187162
Registered:Wed 7 Jun 2017 13:27
56.253456, -11.187162
Registered:Wed 7 Jun 2017 13:27
57.253456, -12.187162
Registered:Wed 7 Jun 2017 13:27
58.253456, -13.187162
Registered:Wed 7 Jun 2017 13:27
59.253456, -14.187162
Registered:Wed 7 Jun 2017 13:27
60.253456, -15.187162
Registered:Wed 7 Jun 2017 13:27
61.253456, -16.187162
Registered:Wed 7 Jun 2017 13:27
62.253456, -17.187162
Registered:Wed 7 Jun 2017 13:27
63.253456, -18.187162
Registered:Wed 7 Jun 2017 13:27

Activity Life Cycle

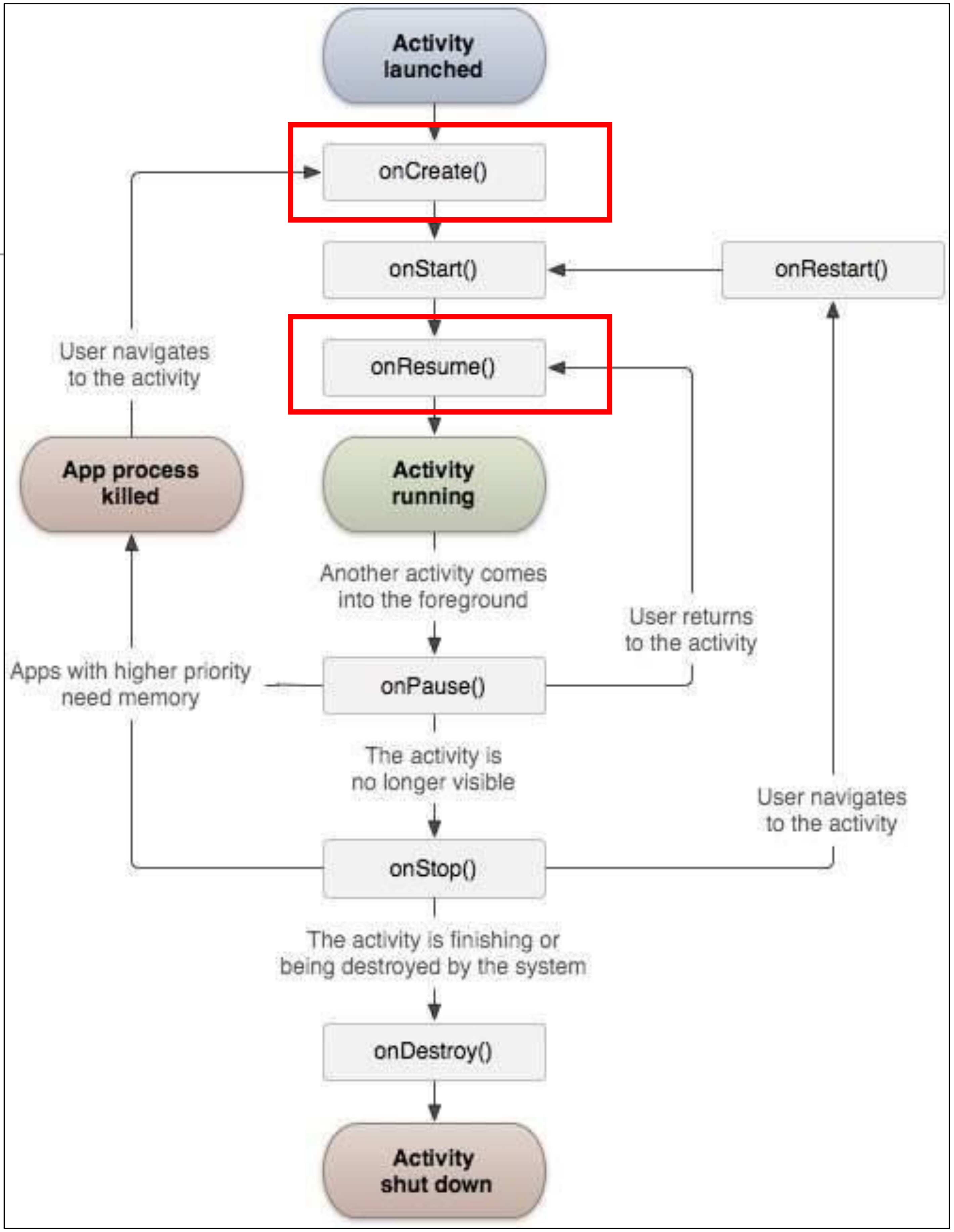
- Recall that an Activity has many callback methods.
  - Callback methods are triggered when an action to which it is attached is executed.
  - An activity doesn't need to implement all the callback methods.


Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

https://developer.android.com/guide/components/activities/activity-lifecycle.html#onresume

Activity Life Cycle

- *onCreate()* method is called when an instance of the Activity subclass is created.
- When the activity enters the Resumed state from the Paused state, it comes to the foreground, and then the system invokes the *onResume()* method.

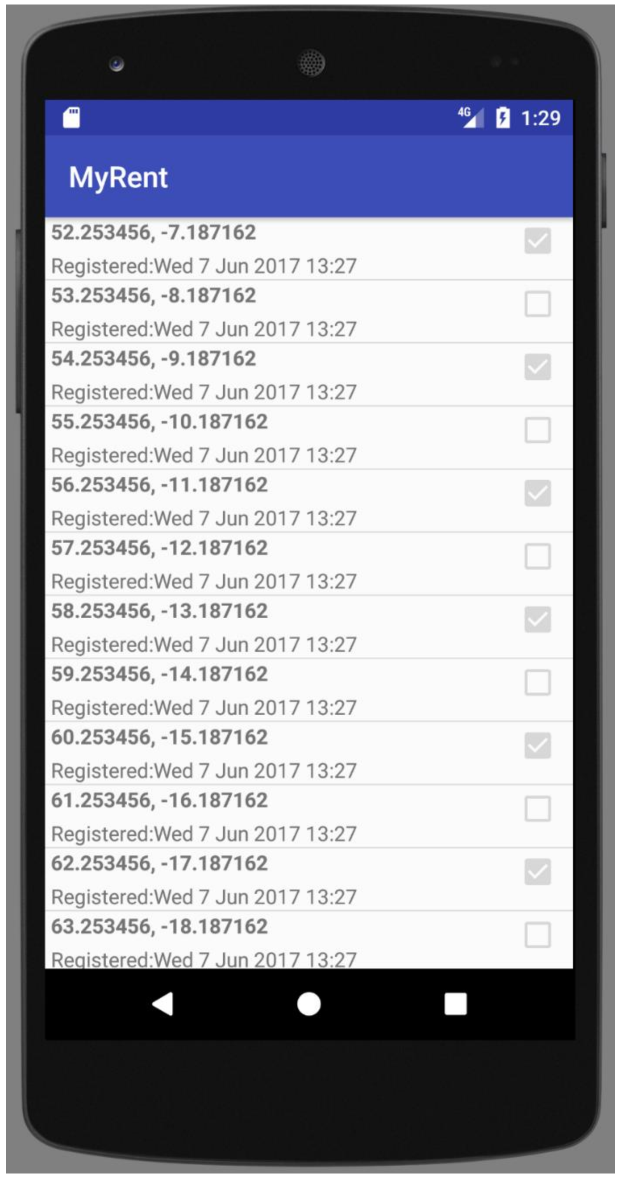

Activity launched
onCreate()
onStart()
onRestart()
User navigates to the activity
onResume()
App process killed
Activity running
Another activity comes into the foreground
User returns to the activity
Apps with higher priority need memory
onPause()
The activity is no longer visible
User navigates to the activity
onStop()
The activity is finishing or being destroyed by the system
onDestroy()
Activity shut down

https://developer.android.com/guide/components/activities/activity-lifecycle.html#onresume

ResidenceListActivity

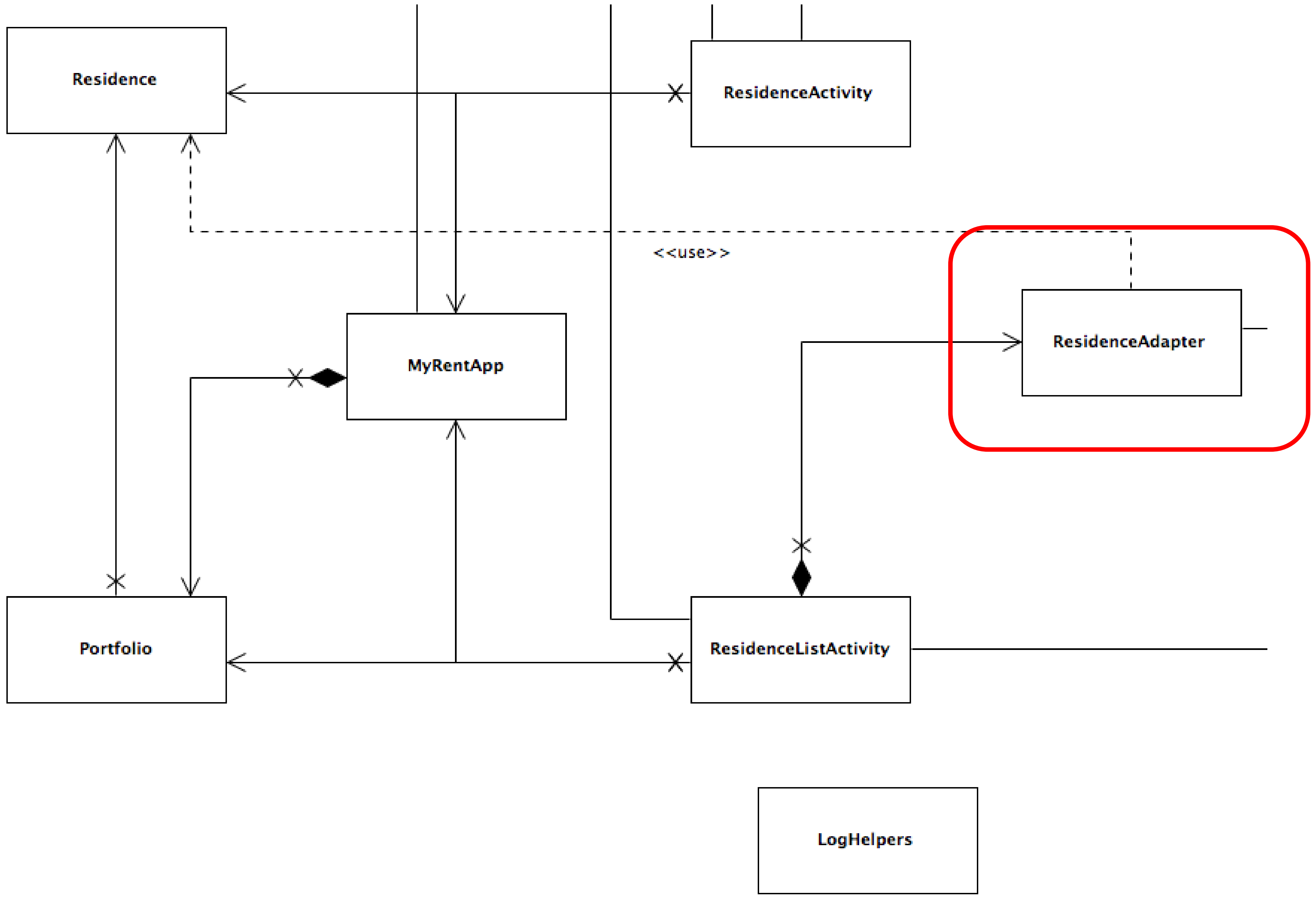
```
    @Override
    public void onResume()
    {
        super.onResume();
        adapter.notifyDataSetChanged();
    }
```

We will cover this in a few slides time…


MyRent

52.253456, -7.187162
Registered:Wed 7 Jun 2017 13:27
53.253456, -8.187162
Registered:Wed 7 Jun 2017 13:27
54.253456, -9.187162
Registered:Wed 7 Jun 2017 13:27
55.253456, -10.187162
Registered:Wed 7 Jun 2017 13:27
56.253456, -11.187162
Registered:Wed 7 Jun 2017 13:27
57.253456, -12.187162
Registered:Wed 7 Jun 2017 13:27
58.253456, -13.187162
Registered:Wed 7 Jun 2017 13:27
59.253456, -14.187162
Registered:Wed 7 Jun 2017 13:27
60.253456, -15.187162
Registered:Wed 7 Jun 2017 13:27
61.253456, -16.187162
Registered:Wed 7 Jun 2017 13:27
62.253456, -17.187162
Registered:Wed 7 Jun 2017 13:27
63.253456, -18.187162
Registered:Wed 7 Jun 2017 13:27

V2.0 – UML with ResidenceListActivity (and the associated adapter)

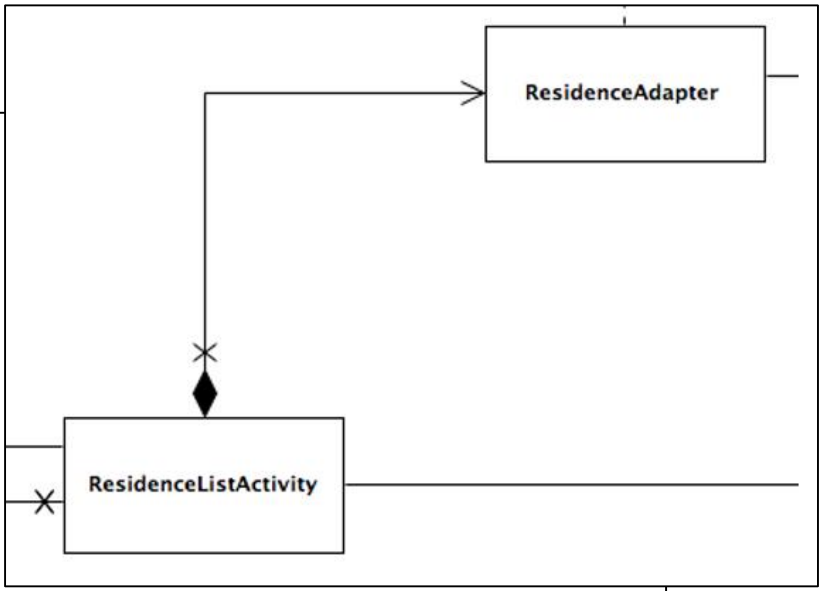


Recap of ArrayAdapter

*An **adapter** is the bridge between a UI component and its data source.*

*An **ArrayAdapter** is commonly used in Android. It returns a view for each object in a collection of data objects you provide, and can be used with list-based user interface widgets such as **ListView** or **Spinner**.*

```
class ResidenceAdapter extends ArrayAdapter<Residence>
{
    private Context context;

    public ResidenceAdapter(Context context, ArrayList<Residence> residences)
    {
        super(context, 0, residences);
        this.context = context;
    }

    @Override
    public View getView(int position, View convertView, ViewGroup parent)
    {
        LayoutInflater inflater = (LayoutInflater) context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        if (convertView == null)
        {
            convertView = inflater.inflate(R.layout.list_item_residence, null);
        }
        Residence res = getItem(position);

        TextView geolocation = (TextView) convertView.findViewById(R.id.residence_list_item_geolocation);
        geolocation.setText(res.geolocation);

        TextView dateTextView = (TextView) convertView.findViewById(R.id.residence_list_item_dateTextView);
        dateTextView.setText(res.getDateString());

        CheckBox rentedCheckBox = (CheckBox) convertView.findViewById(R.id.residence_list_item_isrented);
        rentedCheckBox.setChecked(res.rented);

        return convertView;
    }
}
```

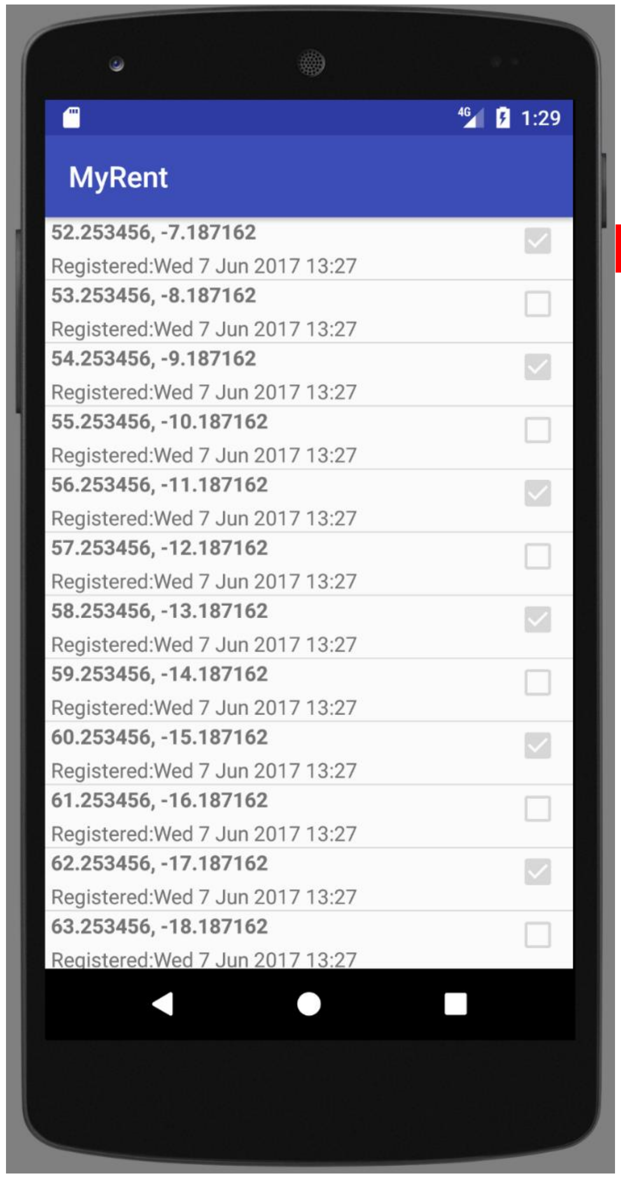


getView is called for each position being displayed → expensive!

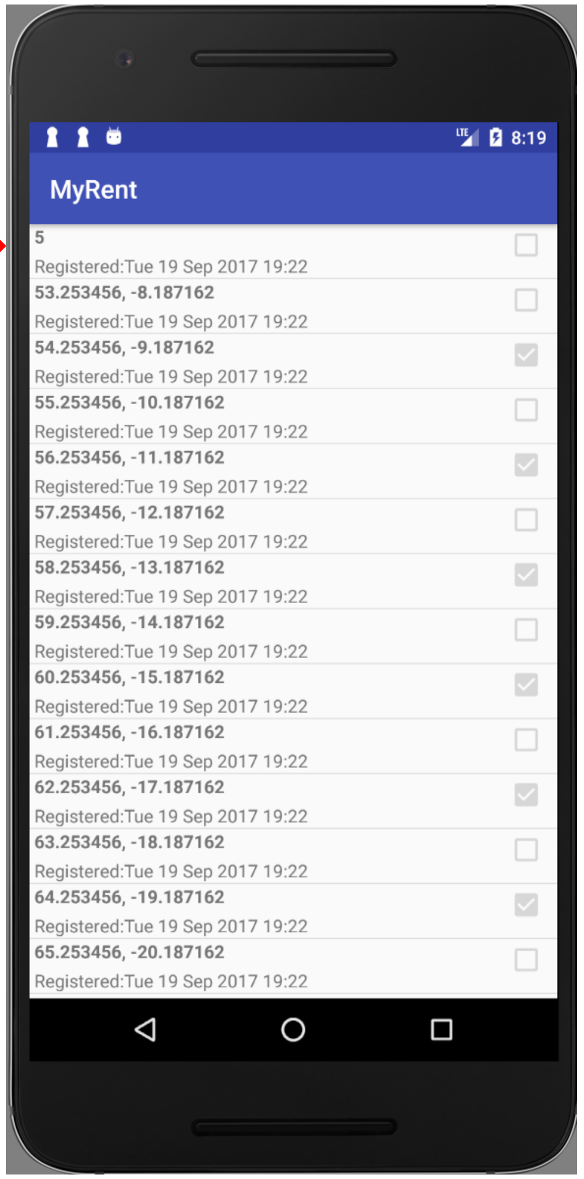
ResidenceListActivity

```
@Override
public void onResume()
{
    super.onResume();
    adapter.notifyDataSetChanged();
}
```

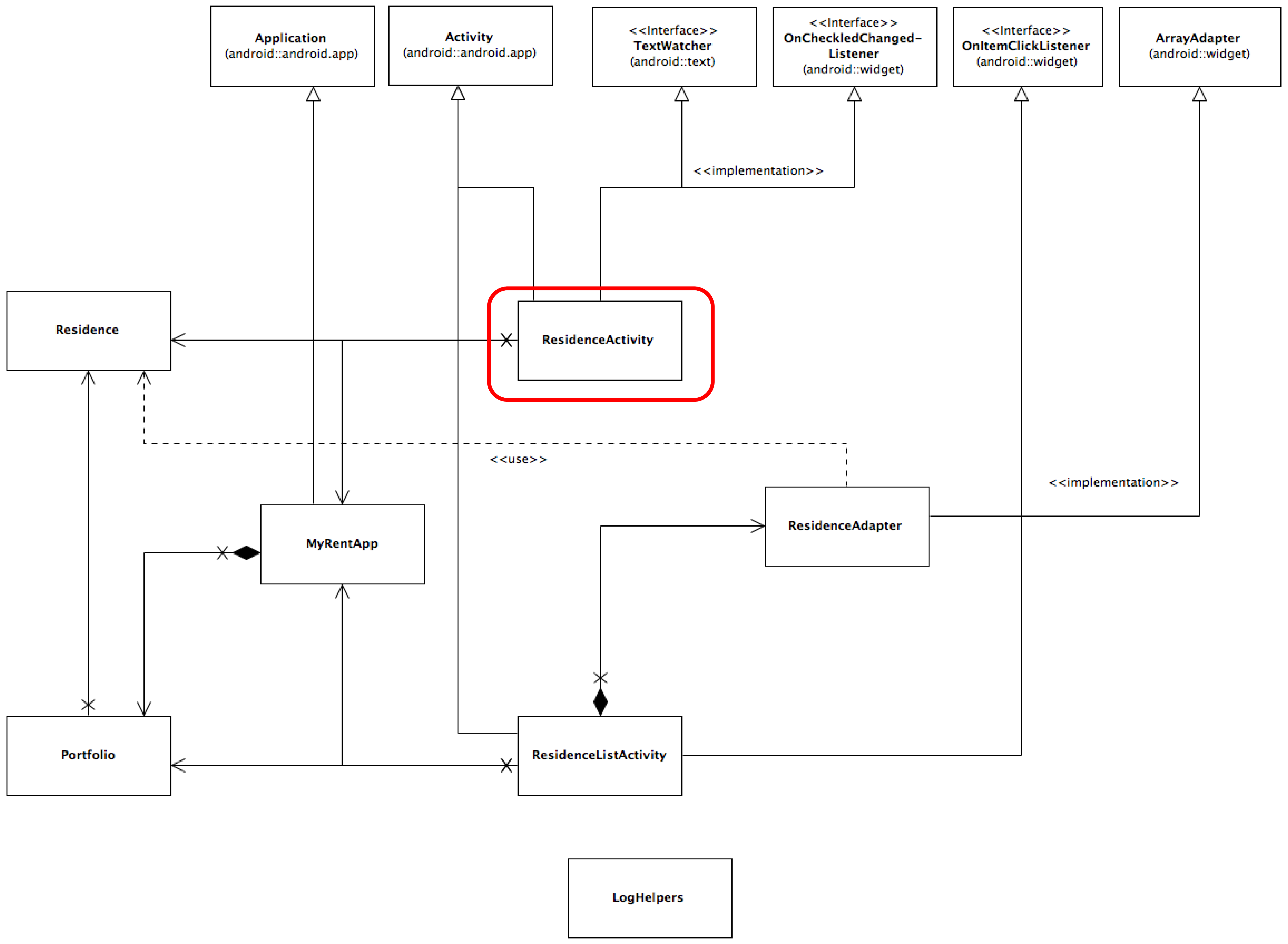
Notify the adapter that the underlying data has been changed and any View reflecting the data set should refresh itself.



```
1:29
MyRent
52.253456, -7.187162
Registered:Wed 7 Jun 2017 13:27
53.253456, -8.187162
Registered:Wed 7 Jun 2017 13:27
54.253456, -9.187162
Registered:Wed 7 Jun 2017 13:27
55.253456, -10.187162
Registered:Wed 7 Jun 2017 13:27
56.253456, -11.187162
Registered:Wed 7 Jun 2017 13:27
57.253456, -12.187162
Registered:Wed 7 Jun 2017 13:27
58.253456, -13.187162
Registered:Wed 7 Jun 2017 13:27
59.253456, -14.187162
Registered:Wed 7 Jun 2017 13:27
60.253456, -15.187162
Registered:Wed 7 Jun 2017 13:27
61.253456, -16.187162
Registered:Wed 7 Jun 2017 13:27
62.253456, -17.187162
Registered:Wed 7 Jun 2017 13:27
63.253456, -18.187162
Registered:Wed 7 Jun 2017 13:27
```

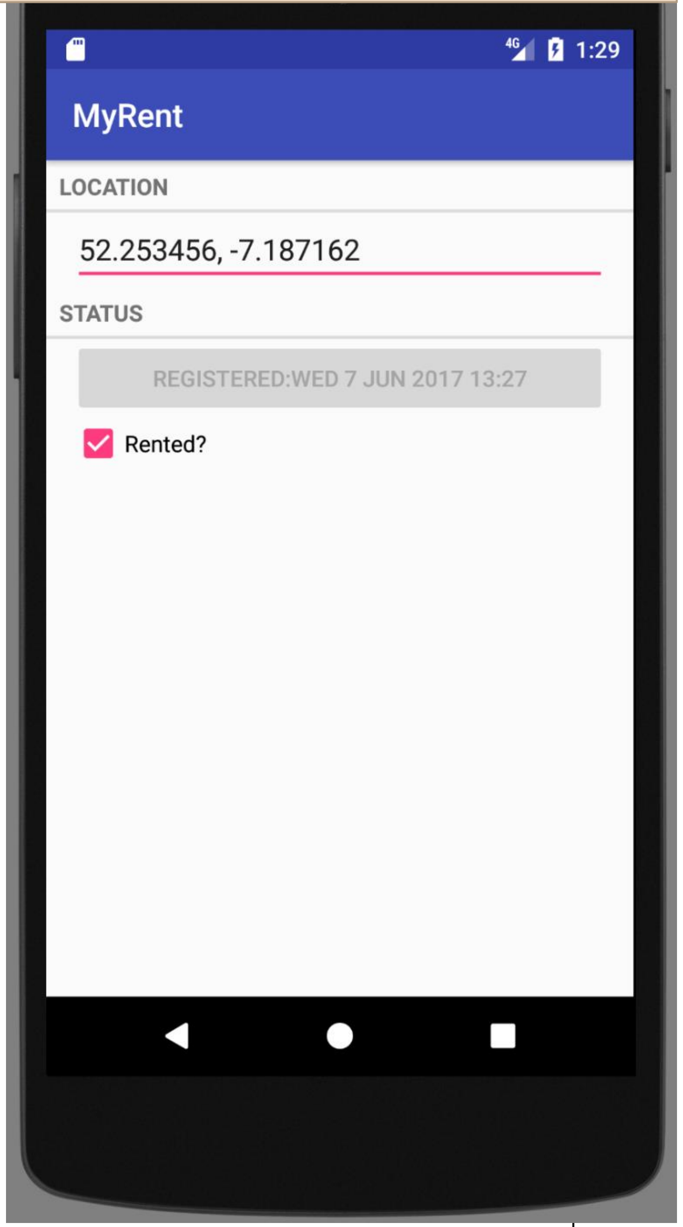


```
8:19
MyRent
5
Registered:Tue 19 Sep 2017 19:22
53.253456, -8.187162
Registered:Tue 19 Sep 2017 19:22
54.253456, -9.187162
Registered:Tue 19 Sep 2017 19:22
55.253456, -10.187162
Registered:Tue 19 Sep 2017 19:22
56.253456, -11.187162
Registered:Tue 19 Sep 2017 19:22
57.253456, -12.187162
Registered:Tue 19 Sep 2017 19:22
58.253456, -13.187162
Registered:Tue 19 Sep 2017 19:22
59.253456, -14.187162
Registered:Tue 19 Sep 2017 19:22
60.253456, -15.187162
Registered:Tue 19 Sep 2017 19:22
61.253456, -16.187162
Registered:Tue 19 Sep 2017 19:22
62.253456, -17.187162
Registered:Tue 19 Sep 2017 19:22
63.253456, -18.187162
Registered:Tue 19 Sep 2017 19:22
64.253456, -19.187162
Registered:Tue 19 Sep 2017 19:22
65.253456, -20.187162
Registered:Tue 19 Sep 2017 19:22
```


Application
(android::android.app)
Activity
(android::android.app)
<<Interface>>
TextWatcher
(android::text)
<<Interface>>
OnCheckledChanged-
Listener
(android::widget)
<<Interface>>
OnItemClickListener
(android::widget)
ArrayAdapter
(android::widget)
<<implementation>>
Residence
ResidenceActivity
<<use>>
MyRentApp
ResidenceAdapter
<<implementation>>
Portfolio
ResidenceListActivity
LogHelpers

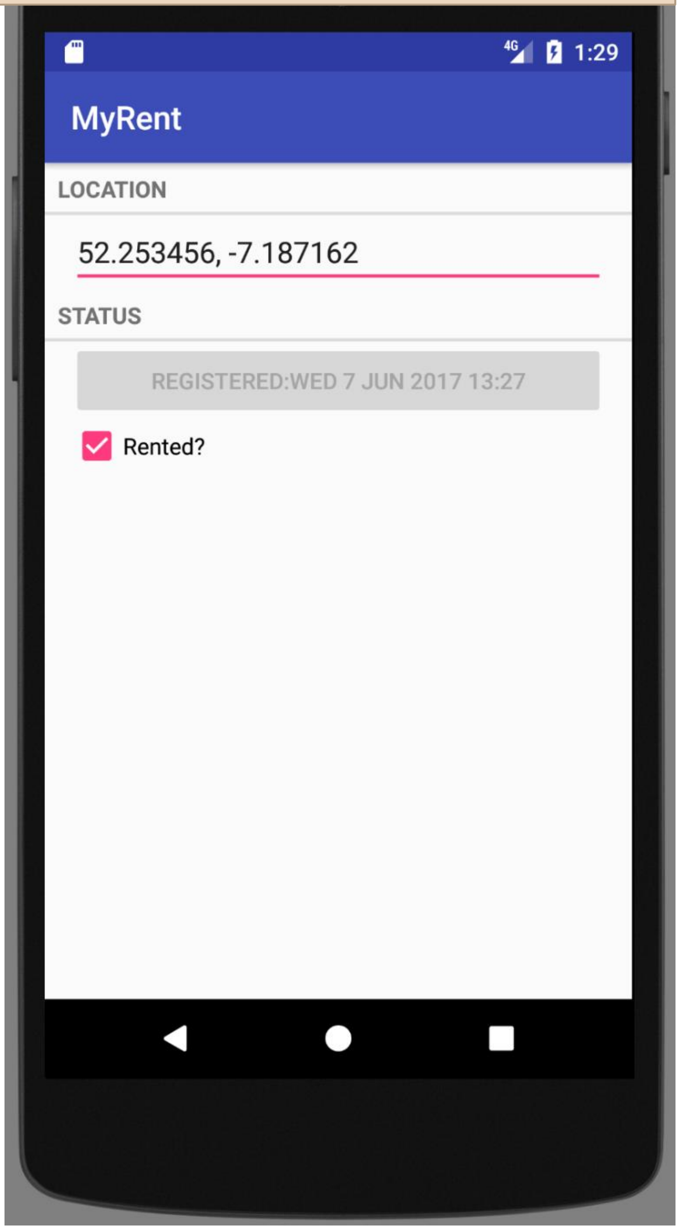
```java
public class ResidenceActivity extends AppCompatActivity implements TextWatcher, OnCheckedChangeListener{

    private EditText geolocation;
    private Residence residence;
    private CheckBox rented;
    private Button dateButton;
    private Portfolio portfolio;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        //omitted code
    }

    public void updateControls(Residence residence){
        //omitted code
    }

    @Override
    public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2)

    }

    @Override
    public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {

    }

    @Override
    public void afterTextChanged(Editable editable) {
        //omitted code
    }

    @Override
    public void onCheckedChanged(CompoundButton compoundButton, boolean isChecked)
        //omitted code
    }
}
```

ResidenceActivity



```
public class ResidenceActivity extends AppCompatActivity implements TextWatcher, OnCheckedChangeListener{

    private EditText geolocation;
    private Residence residence;
    private CheckBox rented;
    private Button dateButton;
    private Portfolio portfolio;

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

        geolocation = (EditText) findViewById(R.id.geolocation);
        residence   = new Residence();
        geolocation.addTextChangedListener(this);

        dateButton  = (Button)   findViewById(R.id.registration_date);
        dateButton.setEnabled(false);

        rented = (CheckBox) findViewById(R.id.isrented);
        rented.setOnCheckedChangeListener(this);

        MyRentApp app = (MyRentApp) getApplication();
        portfolio = app.portfolio;

        Long resId = (Long) getIntent().getExtras()
                                       .getSerializable("RESIDENCE_ID");
        residence = portfolio.getResidence(resId);
        if (residence != null)
        {
            updateControls(residence);
        }
    }

    //omitted code
}
```

ResidenceActivity


MyRent

LOCATION

52.253456, -7.187162

STATUS

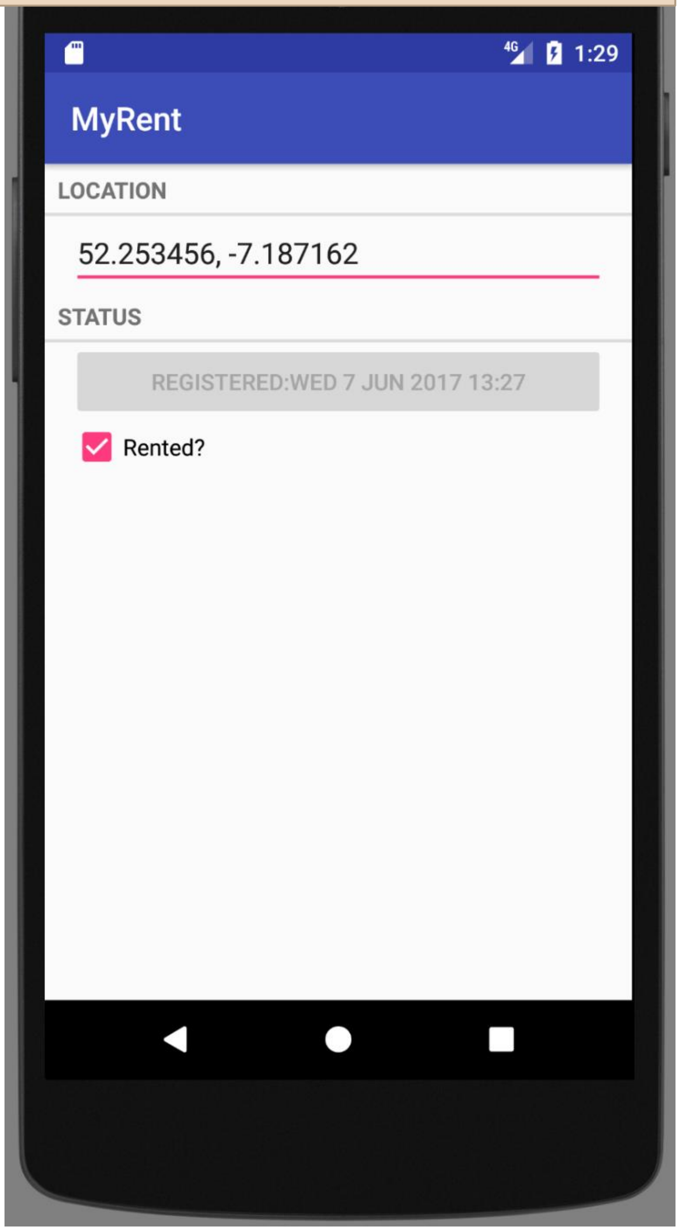
REGISTERED:WED 7 JUN 2017 13:27

☑ Rented?

Retrieve the ID from the ‘Extra’ information

```
public class ResidenceActivity extends AppCompatActivity implements TextWatcher, OnCheckedChangeListener{

    private EditText geolocation;
    private Residence residence;
    private CheckBox rented;
    private Button dateButton;
    private Portfolio portfolio;

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

        geolocation = (EditText) findViewById(R.id.geolocation);
        residence   = new Residence();
        geolocation.addTextChangedListener(this);

        dateButton  = (Button)   findViewById(R.id.registration_date);
        dateButton.setEnabled(false);

        rented = (CheckBox) findViewById(R.id.isrented);
        rented.setOnCheckedChangeListener(this);

        MyRentApp app = (MyRentApp) getApplication();
        portfolio = app.portfolio;

        Long resId = (Long) getIntent().getExtras()
                                        .getSerializable("RESIDENCE_ID");
        residence = portfolio.getResidence(resId);
        if (residence != null)
        {
            updateControls(residence);
        }
    }

    //omitted code
}
```

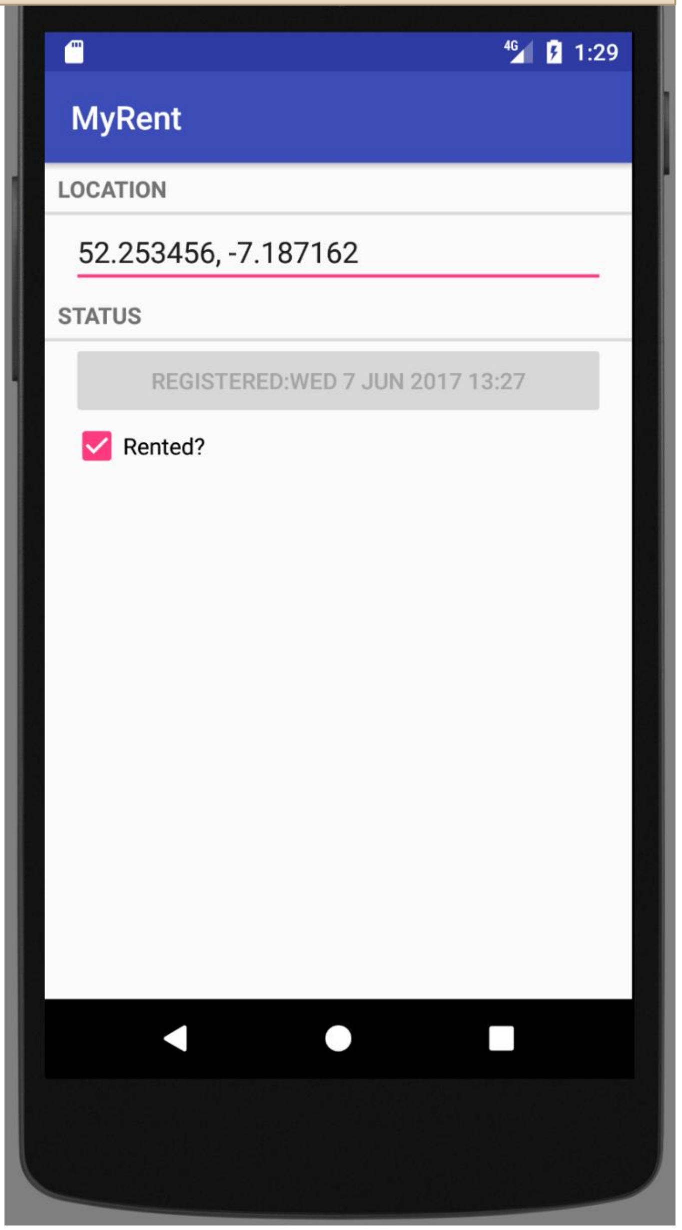
ResidenceActivity



Use the ID to recover the actual Residence object from the portfolio.

```
public class ResidenceActivity extends AppCompatActivity implements TextWatcher, OnCheckedChangeListener{

    private EditText geolocation;
    private Residence residence;
    private CheckBox rented;
    private Button dateButton;
    private Portfolio portfolio;

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

        geolocation = (EditText) findViewById(R.id.geolocation);
        residence   = new Residence();
        geolocation.addTextChangedListener(this);

        dateButton  = (Button)   findViewById(R.id.registration_date);
        dateButton.setEnabled(false);

        rented = (CheckBox) findViewById(R.id.isrented);
        rented.setOnCheckedChangeListener(this);

        MyRentApp app = (MyRentApp) getApplication();
        portfolio = app.portfolio;

        Long resId = (Long) getIntent().getExtras()
                                      .getSerializable("RESIDENCE_ID");
        residence = portfolio.getResidence(resId);
        if (residence != null)
        {
            updateControls(residence);
        }
    }

    //omitted code
}
```

ResidenceActivity

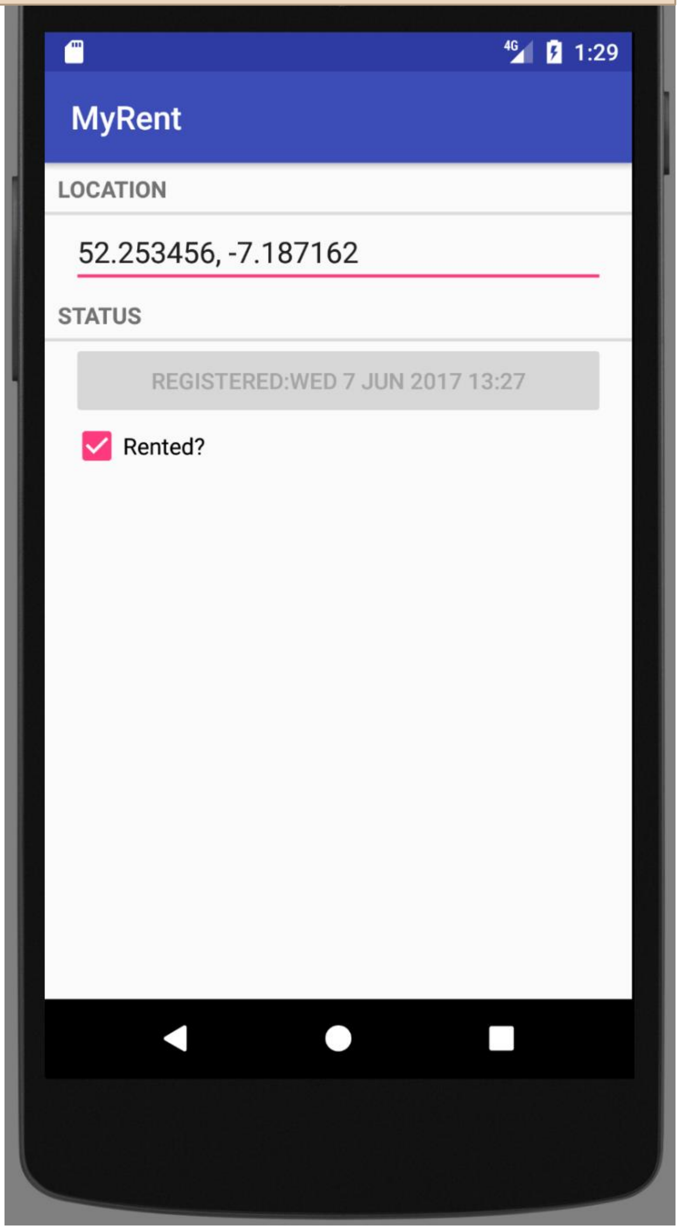


Send this residence information to the controls on the layout.

ResidenceActivity

```
    //omitted code

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

        geolocation = (EditText) findViewById(R.id.geolocation);
        residence   = new Residence();
        geolocation.addTextChangedListener(this);

        dateButton  = (Button)   findViewById(R.id.registration_date);
        dateButton.setEnabled(false);

        rented = (CheckBox) findViewById(R.id.isrented);
        rented.setOnCheckedChangeListener(this);

        MyRentApp app = (MyRentApp) getApplication();
        portfolio = app.portfolio;

        Long resId = (Long) getIntent().getExtras()
                                       .getSerializable("RESIDENCE_ID");
        residence = portfolio.getResidence(resId);
        if (residence != null)
        {
            updateControls(residence);
        }

    }

    public void updateControls(Residence residence)
    {
        geolocation.setText(residence.geolocation);
        rented.setChecked(residence.rented);
        dateButton.setText(residence.getDateString());
    }
  //omitted code
```

Send this residence information to the controls on the layout.

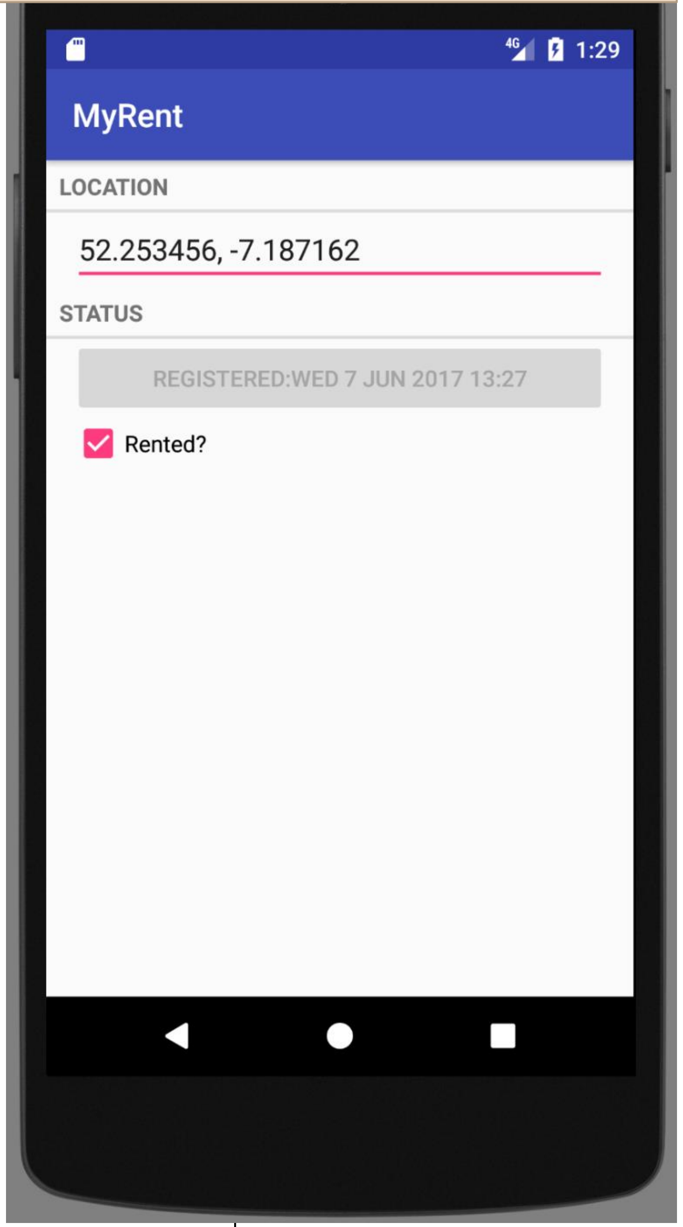


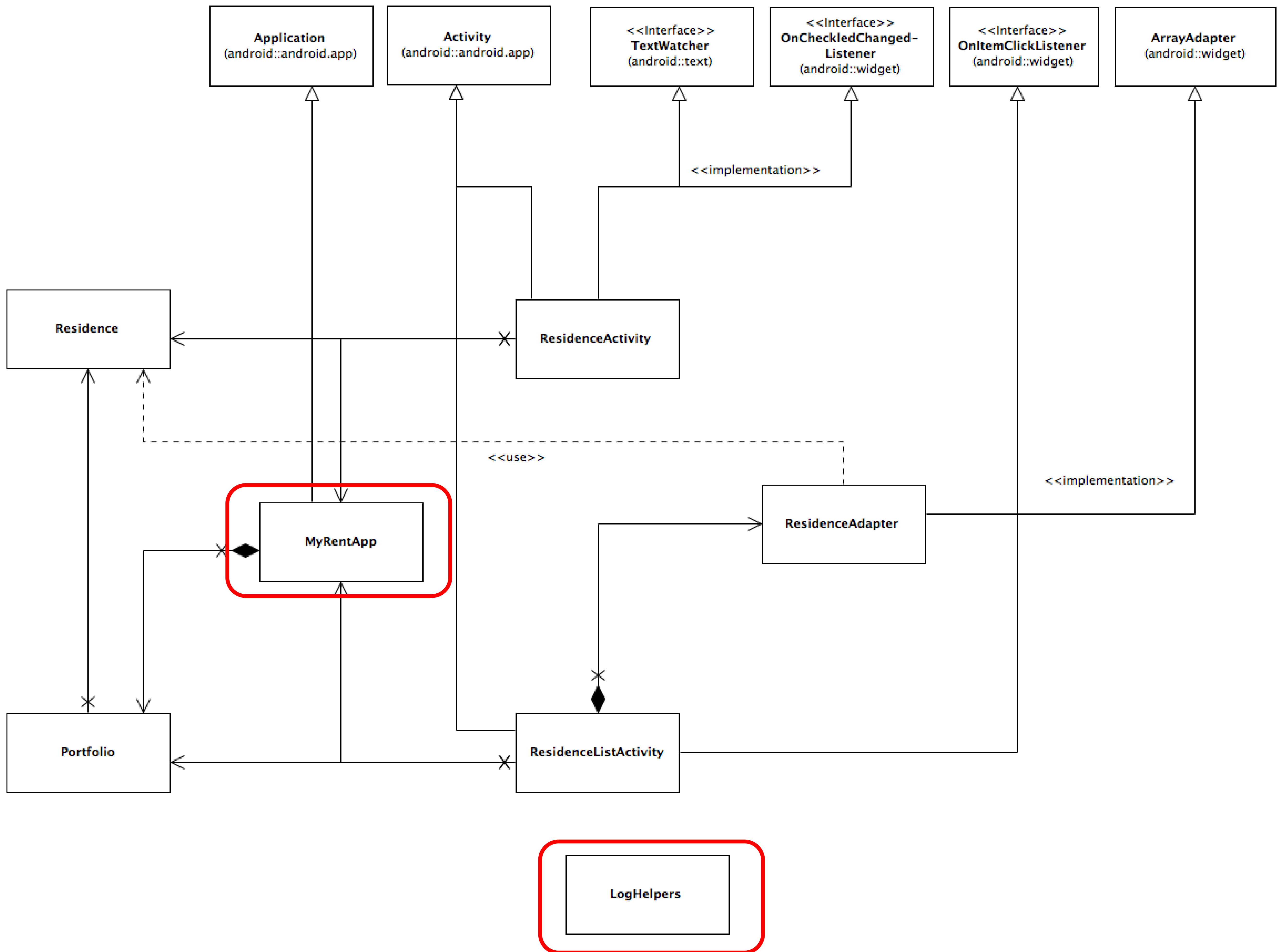
ResidenceActivity

```
public class ResidenceActivity extends AppCompatActivity implements TextWatcher, OnCheckedChangeListener{

    //omitted code
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_residence);

        geolocation = (EditText) findViewById(R.id.geolocation);
        residence   = new Residence();
        geolocation.addTextChangedListener(this);

        dateButton  = (Button)   findViewById(R.id.registration_date);
        dateButton.setEnabled(false);

        rented = (CheckBox) findViewById(R.id.isrented);
        rented.setOnCheckedChangeListener(this);

        MyRentApp app = (MyRentApp) getApplication();
        portfolio = app.portfolio;

        Long resId = (Long) getIntent().getExtras().getSerializable("RESIDENCE_ID");
        residence = portfolio.getResidence(resId);
        if (residence != null){
            updateControls(residence);
        }
    }

    @Override
    public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {
    }

    @Override
    public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {
    }

    @Override
    public void afterTextChanged(Editable editable) {
        residence.setGeolocation(editable.toString());
    }

    @Override
    public void onCheckedChanged(CompoundButton compoundButton, boolean isChecked) {
        Log.i(this.getClass().getSimpleName(), "rented Checked");
        residence.rented = isChecked;
    }

    //omitted code
}
```

Registering the handlers

Event handler methods




Application
(android::android.app)
Activity
(android::android.app)
<<Interface>>
TextWatcher
(android::text)
<<Interface>>
OnCheckledChanged-
Listener
(android::widget)
<<Interface>>
OnItemClickListener
(android::widget)
ArrayAdapter
(android::widget)
<<implementation>>
Residence
ResidenceActivity
<<use>>
<<implementation>>
MyRentApp
ResidenceAdapter
Portfolio
ResidenceListActivity
LogHelpers

Application and LogHelper

```java
package org.wit.myrent.app;

import org.wit.myrent.models.Portfolio;
import android.app.Application;
import static org.wit.android.helpers.LogHelpers.info;

public class MyRentApp extends Application
{
    public Portfolio portfolio;

    @Override
    public void onCreate()
    {
        super.onCreate();
        portfolio = new Portfolio();

        info(this, "MyRent app launched");
    }
}
```

Notice how the LogHelpers class is used here to simplify our “logging” code

→ Easier to read.

```java
public class LogHelpers
{
  public static void info(Object parent, String message){
    Log.i(parent.getClass().getSimpleName(), message);
  }
}
```

AndroidManifest

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

    <application
        android:name=".app.MyRentApp"
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".activities.ResidenceActivity">
        </activity>
        <activity android:name=".activities.ResidenceListActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

Questions?

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning support unit