

Mobile Application Development

Produced
by

Eamonn de Leastar (edelestar@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>

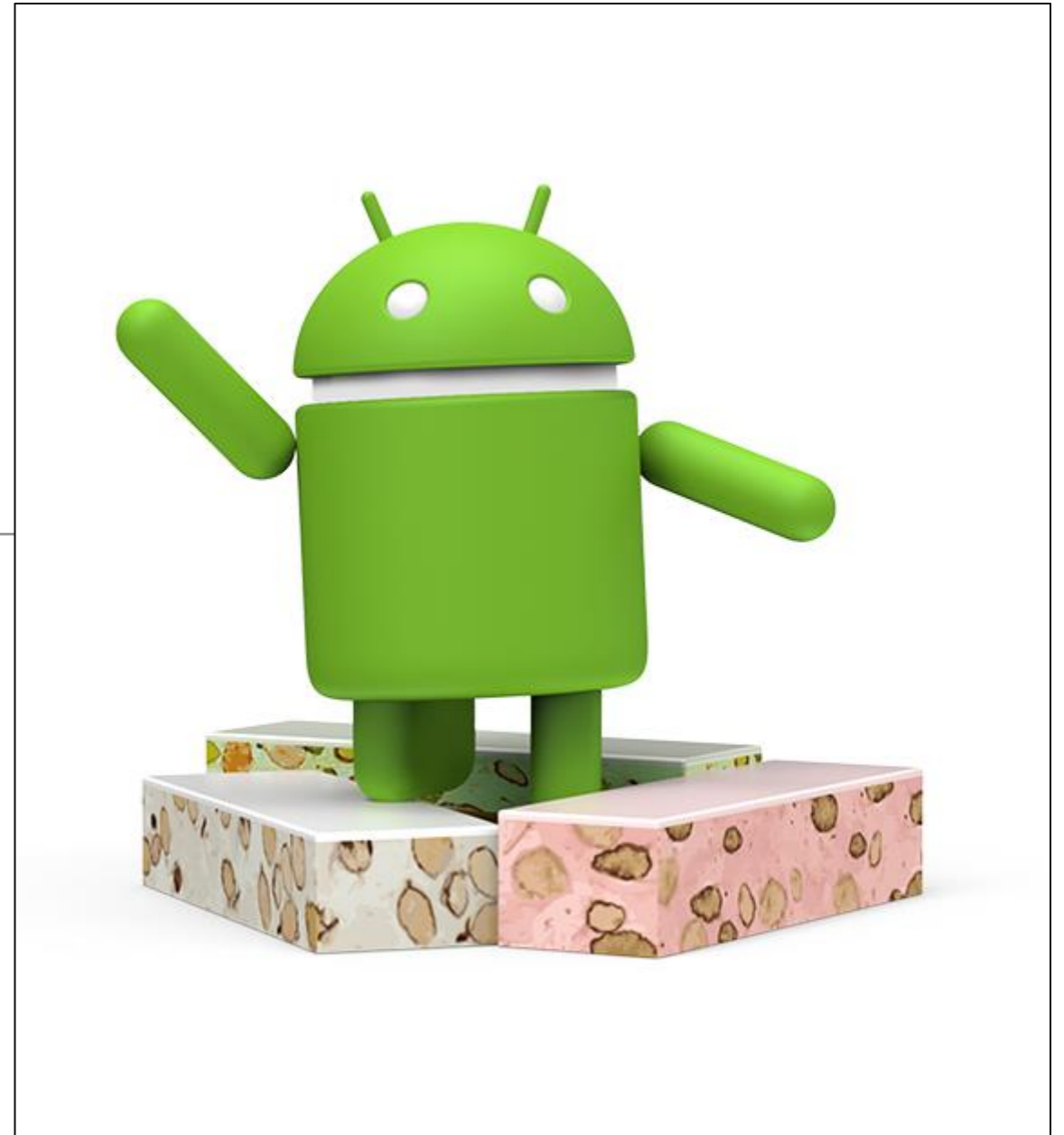


Waterford Institute of Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRCE



Second Android Application

MyRent V01



MyRent

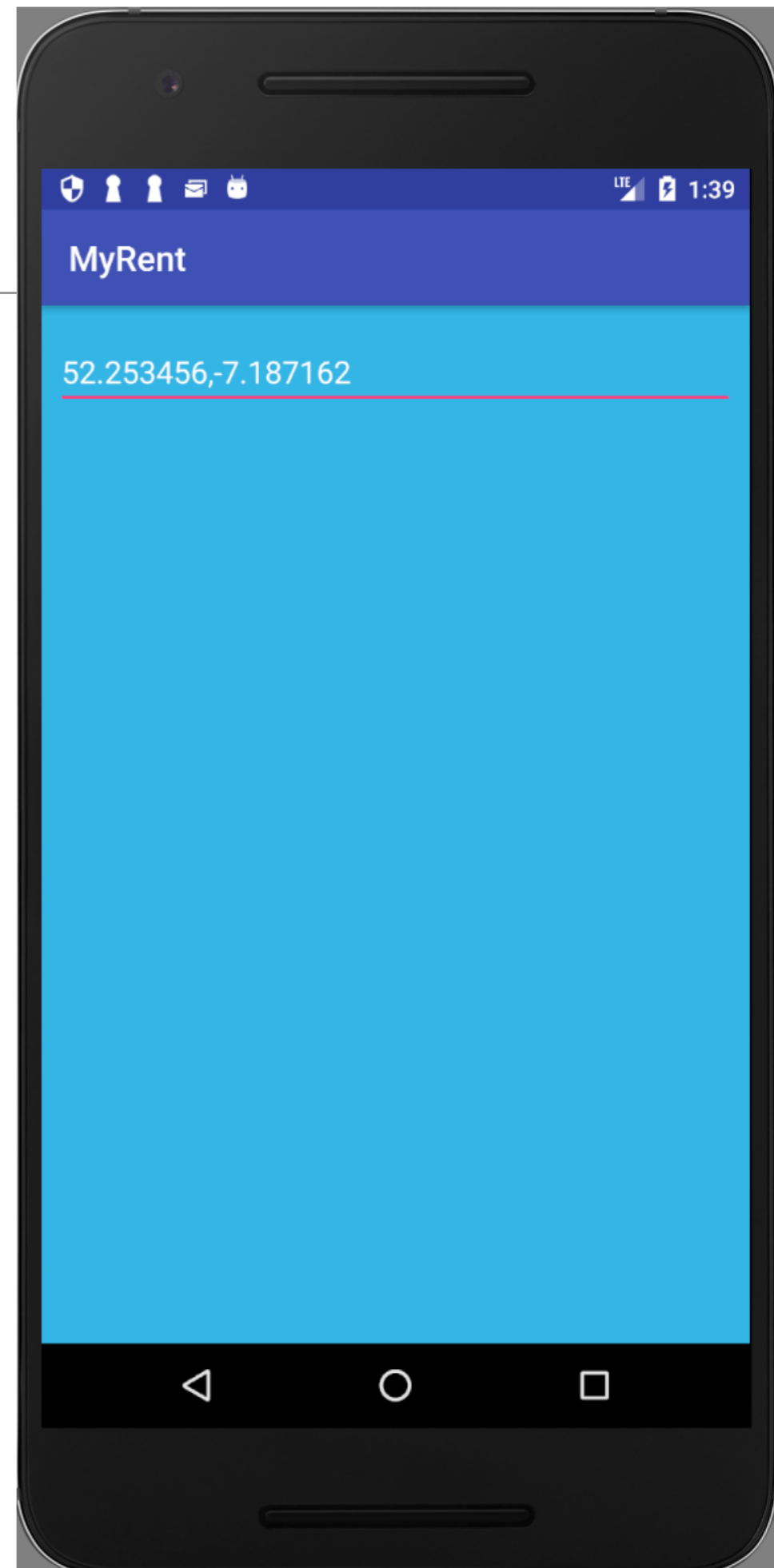
- An application to track location, rental status and other information about properties.
- Develop in several versions to include:
 - multiple properties
 - location map
 - images
 - contact information
 - etc...

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 ' <i>Master-Detail</i> ' pattern. <i>Master</i> holds a list of rented residences whereas the <i>Detail</i> 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.

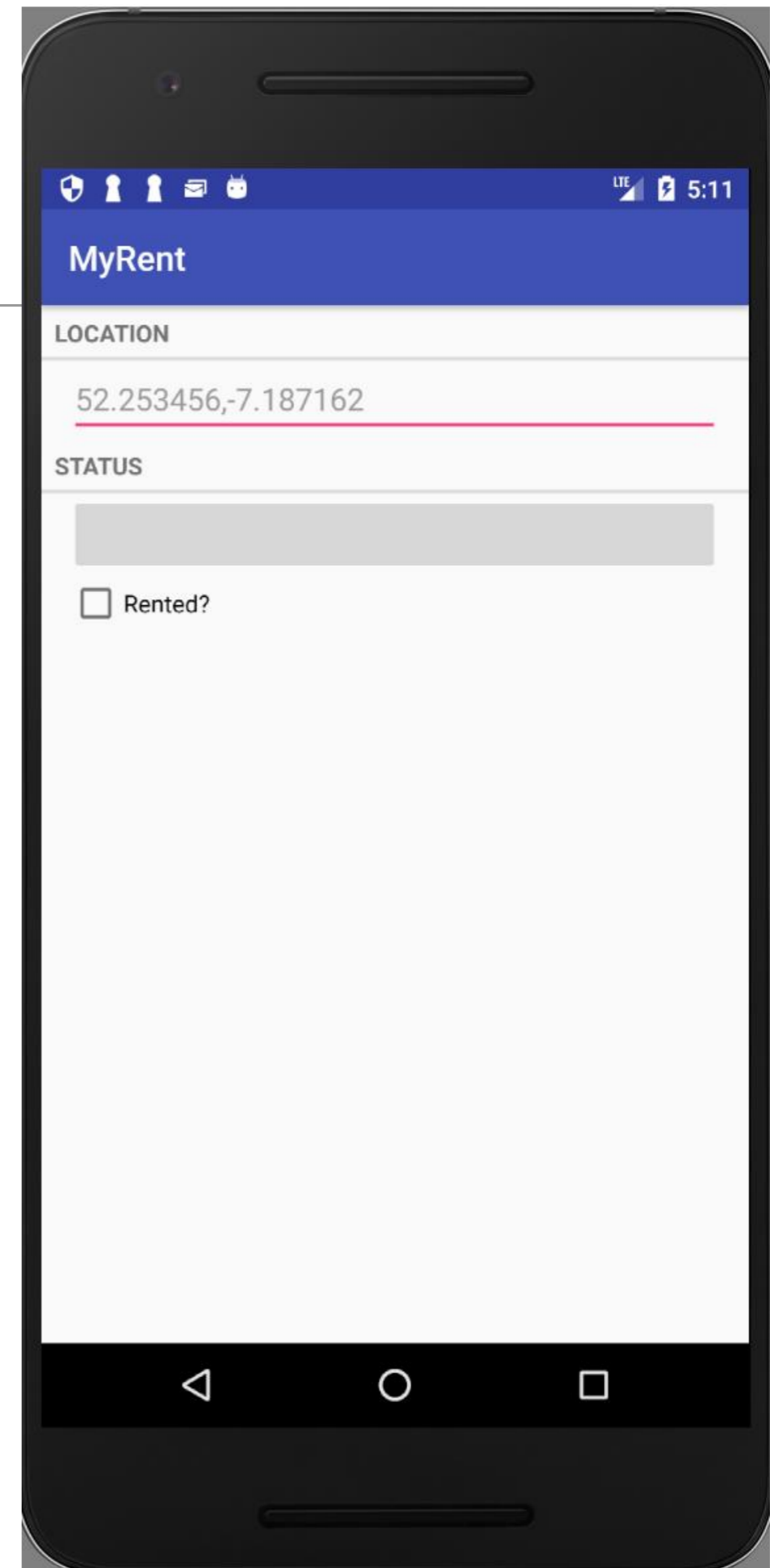
MyRent V00

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

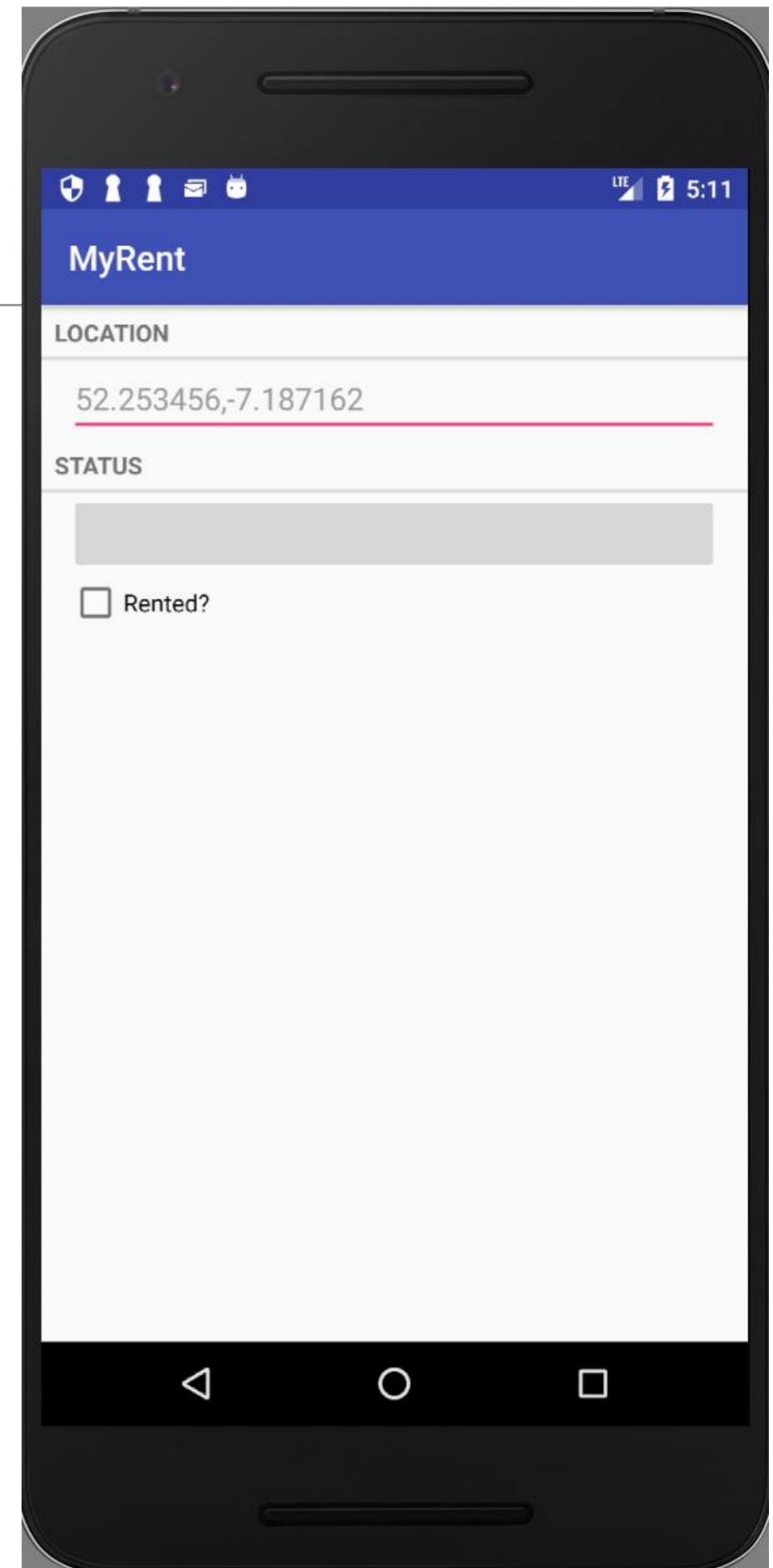
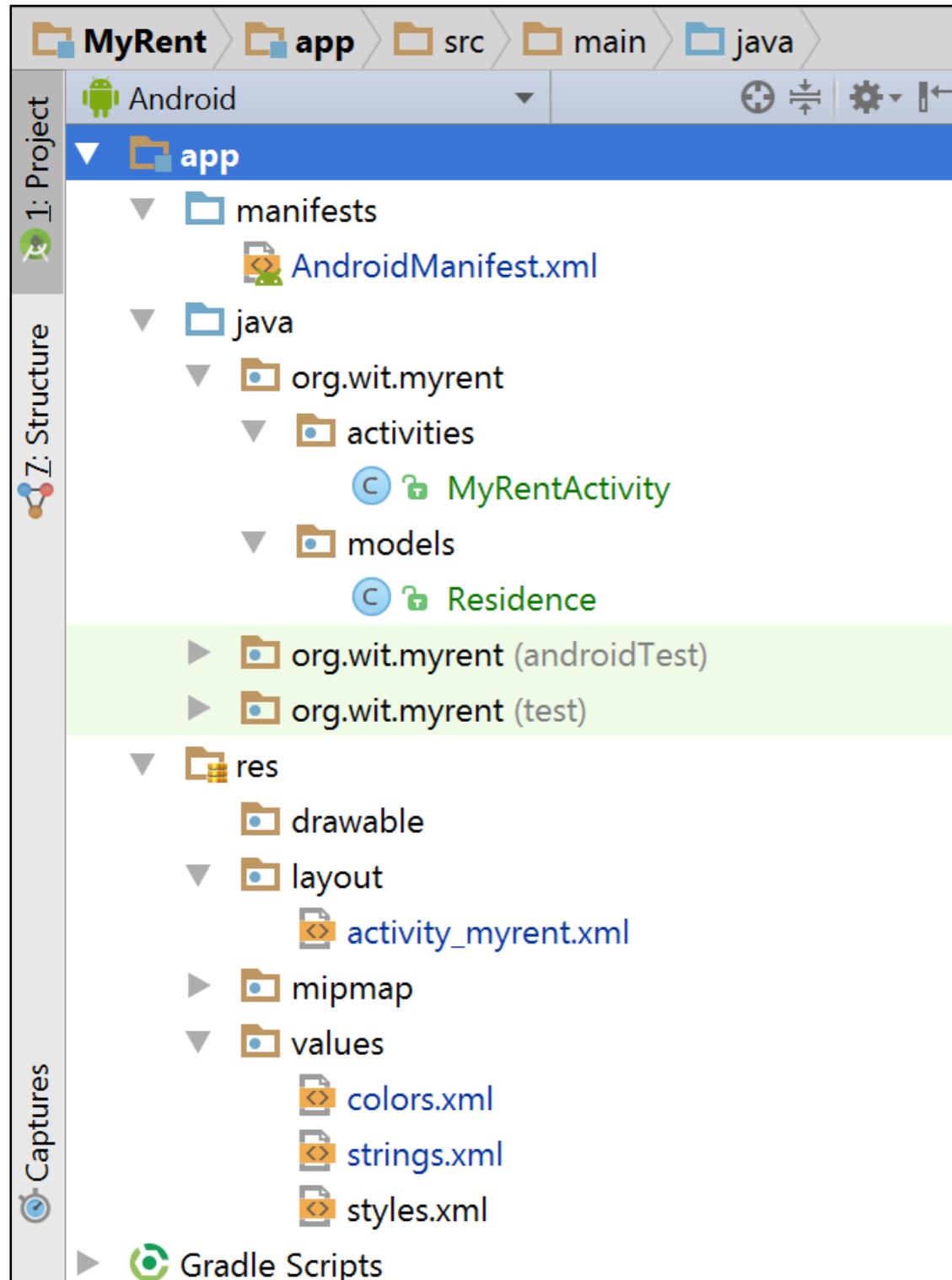


MyRent V01

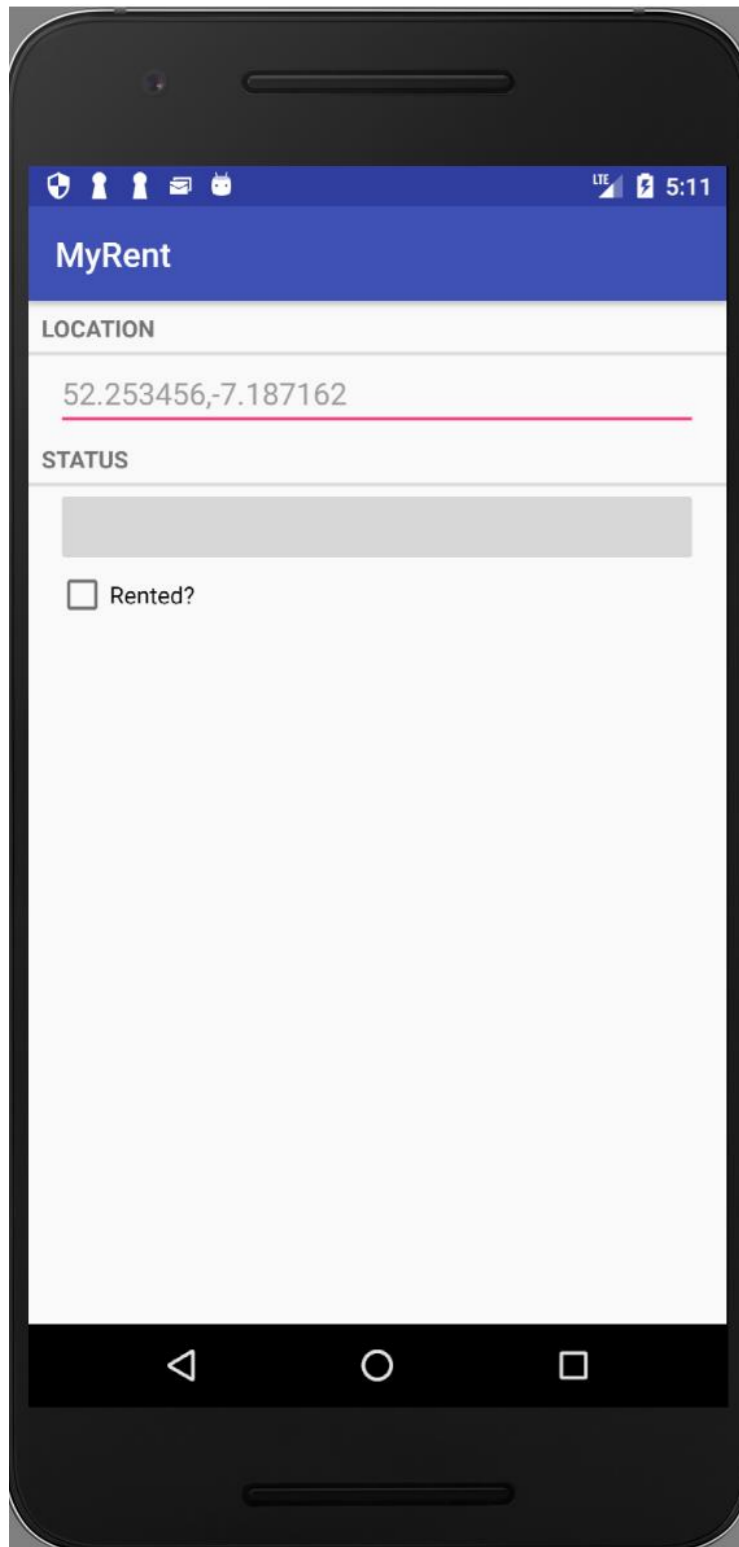
- One Activity with multiple widgets to display:
 - Location (with hint text)
 - Status
 - Rented?



MyRent V01



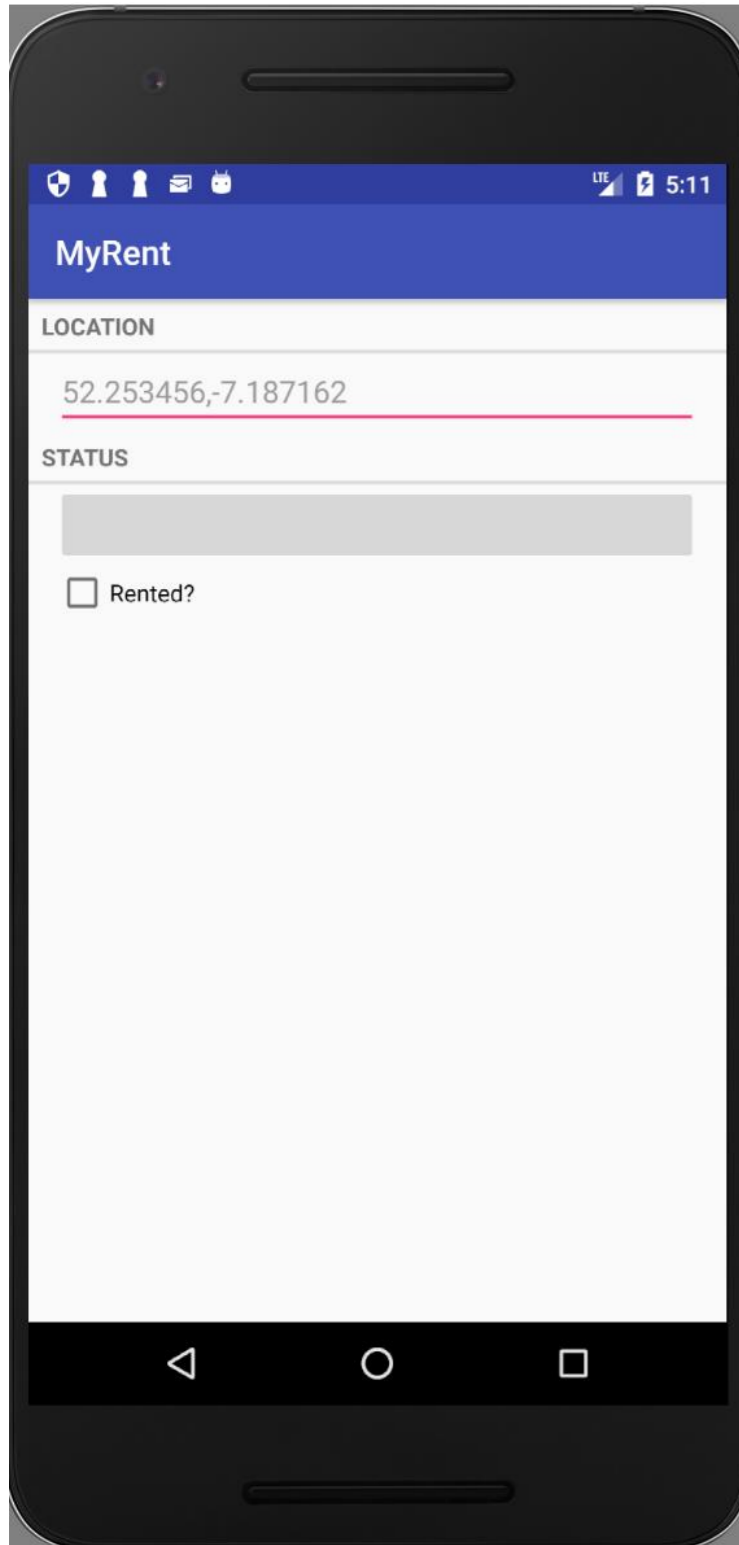
Layout Structure



LinearLayout

- a layout that arranges other views either horizontally in a single column or vertically in a single row.
- set `android:orientation` to specify whether child views are displayed in a row or column.

Layout Structure



```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
```

```
<!-- LOCATION SEPARATOR -->
<TextView
    android:id="@+id/locationLabel"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/location"
    style="?android:listSeparatorTextViewStyle"/>
```

```
<!-- Geolocation (GPS Coords) -->
<EditText
    android:id="@+id/geolocation"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="16dp"
    android:layout_marginRight="16dp"
    android:hint="@string/geolocation_hint"
    <requestFocus/>
</EditText>
```

```
<!-- STATUS SEPARATOR-->
<TextView
    android:id="@+id/statusLabel"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:text="@string/status"
    style="?android:listSeparatorTextViewStyle"/>
```

```
<Button
    android:id="@+id/registration_date"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginLeft="16dp"
    android:layout_marginRight="16dp"/>
```

```
<CheckBox
    android:id="@+id/isrented"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="@string/isrented"
    android:layout_marginLeft="16dp"
    android:layout_marginRight="16dp"
    android:checked="false"/>
```

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

    private String geolocation;
    public boolean rented;

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

    // Generate a long greater than zero
    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();
    }
}
```

Residence.java

Date and rented fields introduced.

Also, fields made public for convenience and speed.

ResidenceActivity - Initialisation

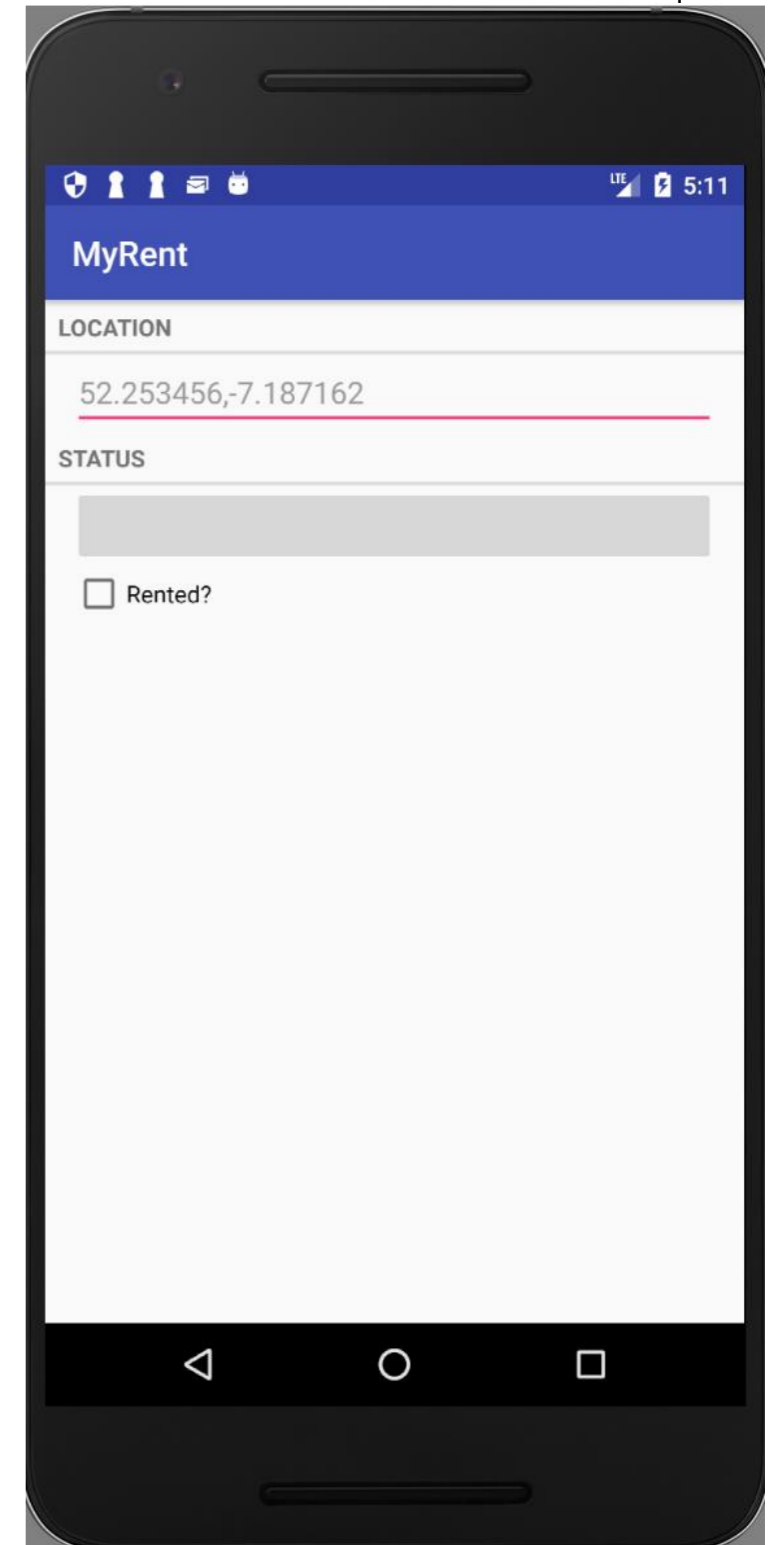
```
public class ResidenceActivity extends AppCompatActivity implements TextWatcher, OnCheckedChangeListener
{
    private EditText geolocation;
    private CheckBox rented;
    private Button dateButton;
    private Residence residence;

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

        geolocation = (EditText) findViewById(R.id.geolocation);
        dateButton = (Button) findViewById(R.id.registration_date);
        rented = (CheckBox) findViewById(R.id.isrented);

        residence = new Residence();

        geolocation.addTextChangedListener(this);
        geolocation.setText(residence.geolocation);
        dateButton.setEnabled(false);
        rented.setChecked(residence.rented);
        rented.setOnCheckedChangeListener(this);
    }
    ...
}
```



ResidenceActivity – Event Handlers

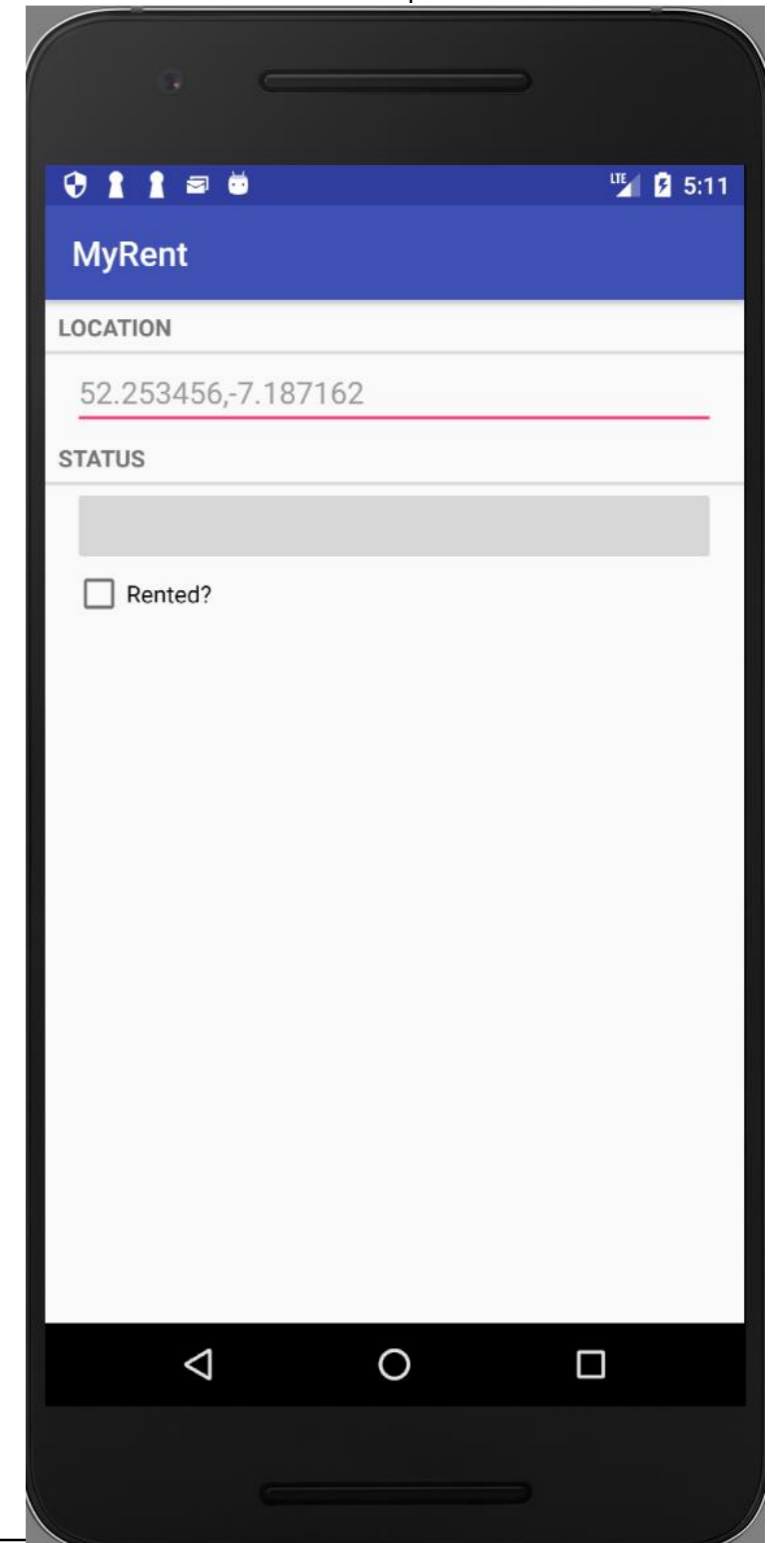
```
public class ResidenceActivity extends AppCompatActivity implements TextWatcher, OnCheckedChangeListener
{
    private EditText geolocation;
    private CheckBox rented;
    //...
    @Override
    public void onCreate(Bundle savedInstanceState){
        //..
        geolocation.addTextChangedListener(this);
        rented .setOnCheckedChangeListener(this);
    }

    @Override
    public void onCheckedChanged(CompoundButton arg0, boolean isChecked){
        Log.i(this.getClass().getSimpleName(), "rented Checked");
        residence.rented = isChecked;
    }

    @Override
    public void afterTextChanged(Editable c){
        Log.i(this.getClass().getSimpleName(), "geolocation " + c.toString());
        residence.geolocation = c.toString();
    }

    @Override
    public void beforeTextChanged(CharSequence arg0, int arg1, int arg2, int arg3){
    }

    @Override
    public void onTextChanged(CharSequence arg0, int arg1, int arg2, int arg3){
    }
}
```



ResidenceActivity – Event Handlers

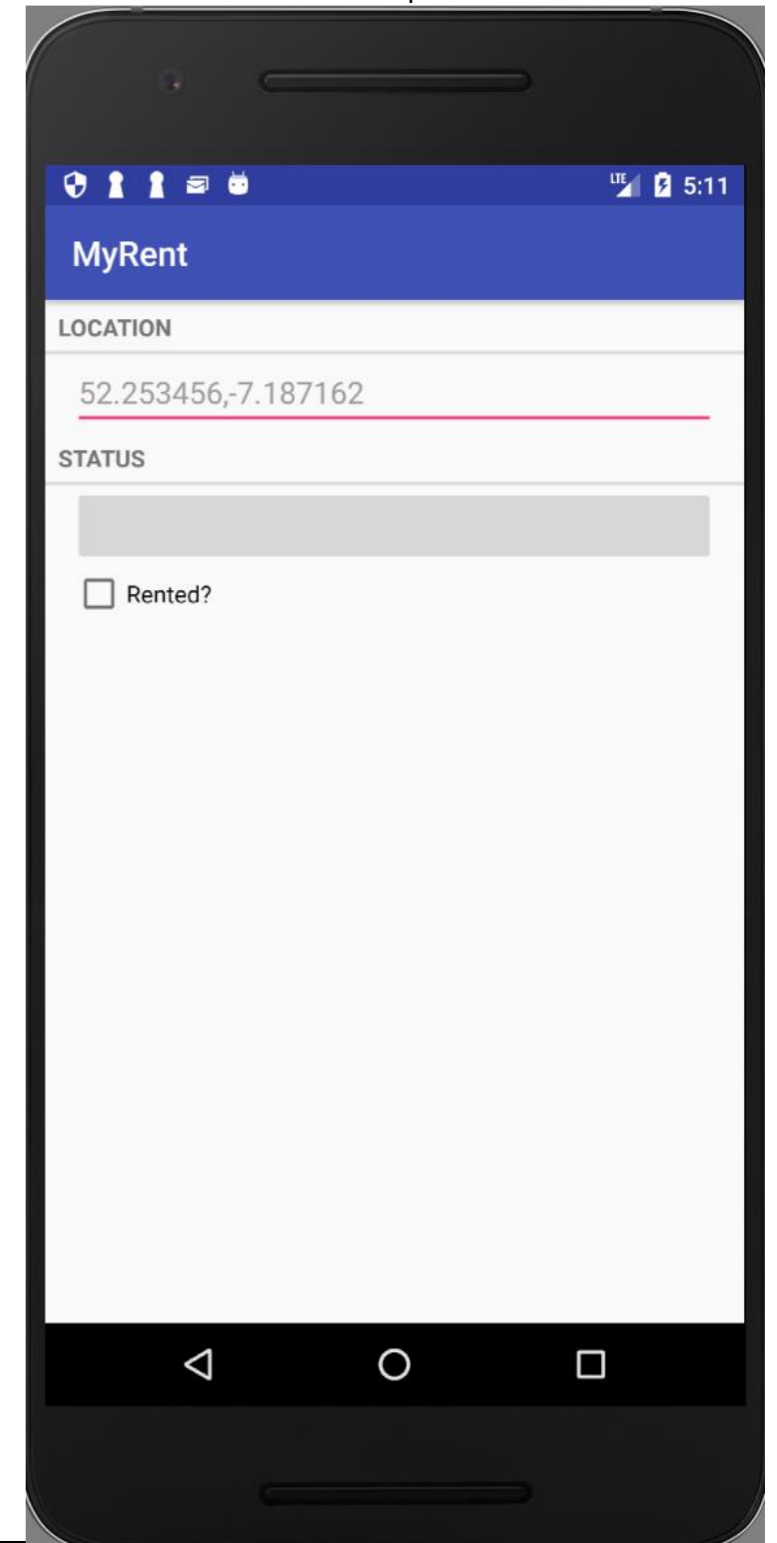
```
public class ResidenceActivity extends AppCompatActivity implements TextWatcher, OnCheckedChangeListener
{
    private EditText geolocation;
    private CheckBox rented;
    //...
    @Override
    public void onCreate(Bundle savedInstanceState){
        //..
        geolocation.addTextChangedListener(this);
        rented .setOnCheckedChangeListener(this);
    }

    @Override
    public void onCheckedChanged(CompoundButton arg0, boolean isChecked){
        Log.i(this.getClass().getSimpleName(), "rented Checked");
        residence.rented = isChecked;
    }

    @Override
    public void afterTextChanged(Editable c){
        Log.i(this.getClass().getSimpleName(), "geolocation " + c.toString());
        residence.geolocation = c.toString();
    }

    @Override
    public void beforeTextChanged(CharSequence arg0, int arg1, int arg2, int arg3){
    }

    @Override
    public void onTextChanged(CharSequence arg0, int arg1, int arg2, int arg3){
    }
}
```



OnCheckedChangeListener Interface

Interface definition for a callback method to be invoked when a button is checked/unchecked.

Public methods

<code>abstract</code>	<code>onCheckedChanged(CompoundButton buttonView, boolean isChecked)</code>
<code>void</code>	Called when the checked state of a compound button has changed.

ResidenceActivity – Event Handlers

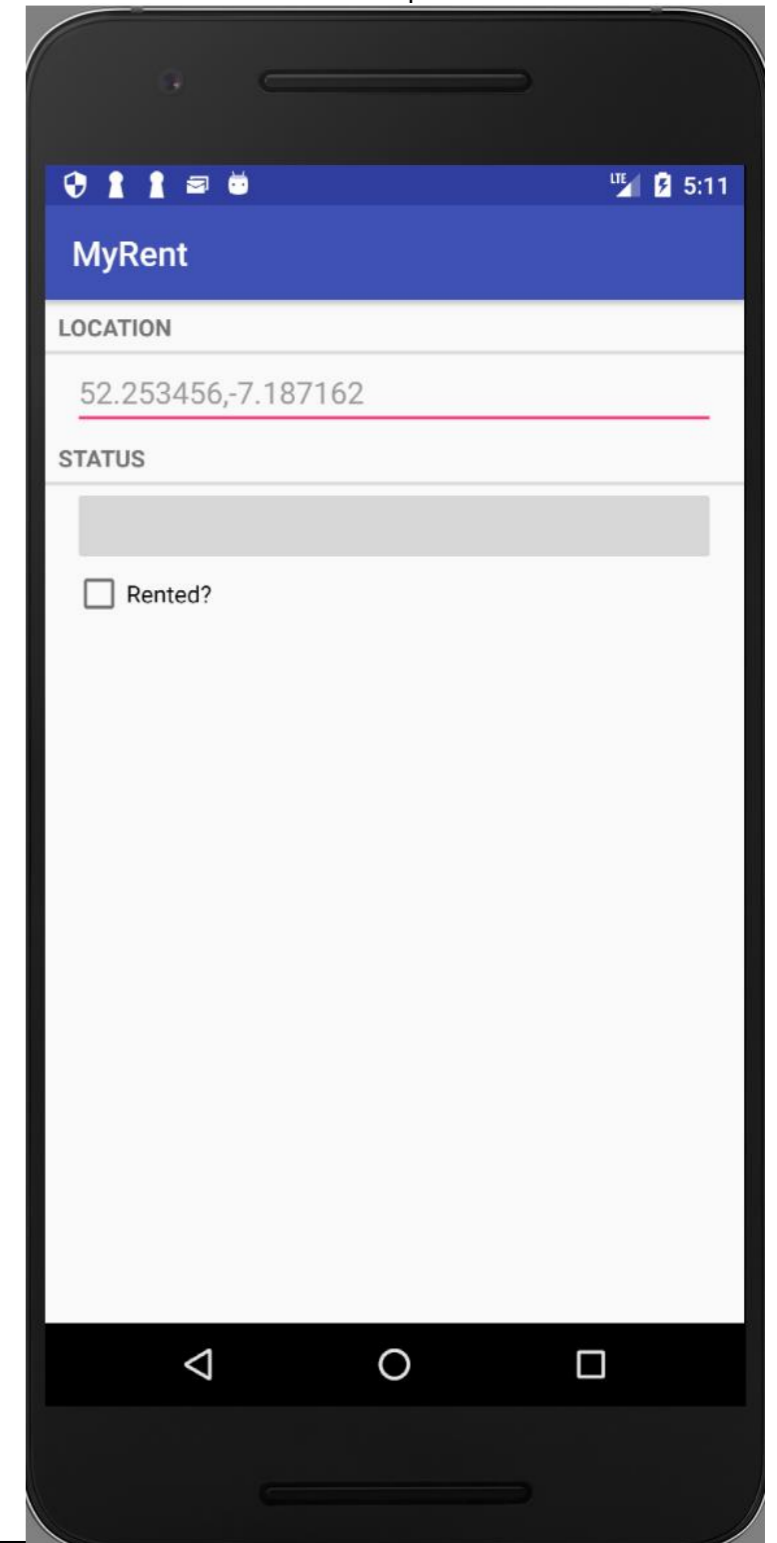
```
public class ResidenceActivity extends AppCompatActivity implements TextWatcher, OnCheckedChangeListener
{
    private EditText geolocation;
    private CheckBox rented;
    //...
    @Override
    public void onCreate(Bundle savedInstanceState){
        //..
        geolocation.addTextChangedListener(this);
        rented.setOnCheckedChangeListener(this);
    }
}
```

```
@Override
public void onCheckedChanged(CompoundButton arg0, boolean isChecked){
    Log.i(this.getClass().getSimpleName(), "rented Checked");
    residence.rented = isChecked;
}
```

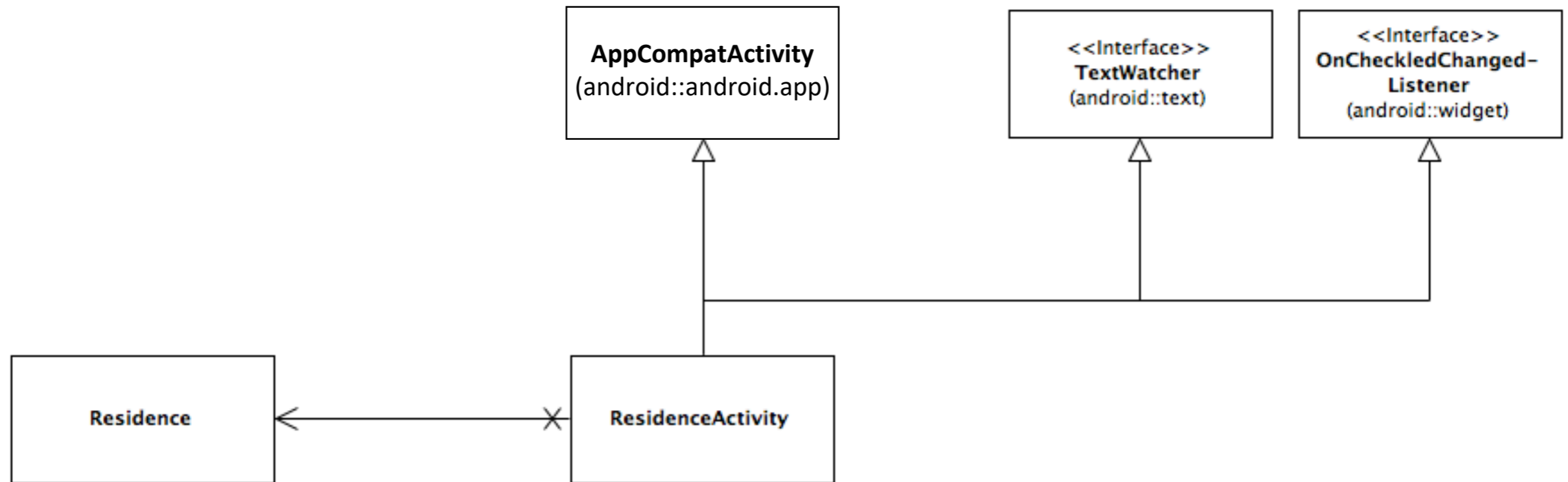
```
@Override
public void afterTextChanged(Editable c){
    Log.i(this.getClass().getSimpleName(), "geolocation " + c.toString());
    residence.geolocation = c.toString();
}
```

```
@Override
public void beforeTextChanged(CharSequence arg0, int arg1, int arg2, int arg3){
}
```

```
@Override
public void onTextChanged(CharSequence arg0, int arg1, int arg2, int arg3){
}
```

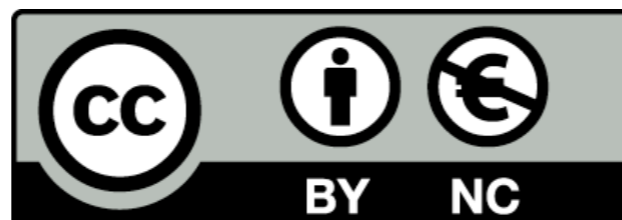


MyRent V01 UML



Questions?





Except where otherwise noted, this content is licensed under a [Creative Commons Attribution-NonCommercial 3.0 License](http://creativecommons.org/licenses/by-nc/3.0/).

For more information, please see <http://creativecommons.org/licenses/by-nc/3.0/>

