

COMP1661 Logbook Upload Template

1. Basic Information

1.1	Student name	Myles Findlay
1.2	Who did you work with? Note that for logbook exercises you are allowed to work with one other person as long as you give their name and login id and both contribute to the work.	Name: Login id:
1.3	Which Exercise is this? Tick as appropriate.	1 - Create a PhoneGap App utilizing the Notification API <input type="checkbox"/> 2 - Create a PhoneGap data entry screen RentalZ App <input type="checkbox"/> 3 - Create an SQLite database for RentalZ App <input type="checkbox"/> 4 - Create Android data entry screen for RentalZ App <input checked="" type="checkbox"/>
1.4	How well did you complete the exercise? Tick as appropriate.	• I tried but couldn't complete it <input type="checkbox"/> • I did it but I feel I should have done better <input type="checkbox"/> • I did everything that was asked <input type="checkbox"/> • I did more than was asked for <input checked="" type="checkbox"/>
1.5	Briefly explain your answer to question 1.4 e.g. what you could have done better?	n/a

2. Exercise answer

2.1 Screenshots demonstrating what you achieved

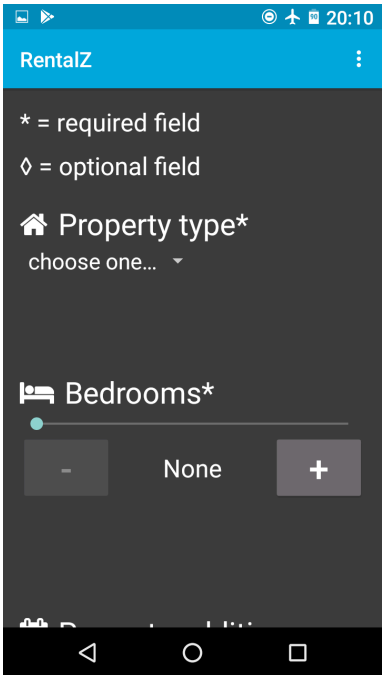


Figure 1: RentalZ form ('Property type' and 'Bedrooms' fields shown)

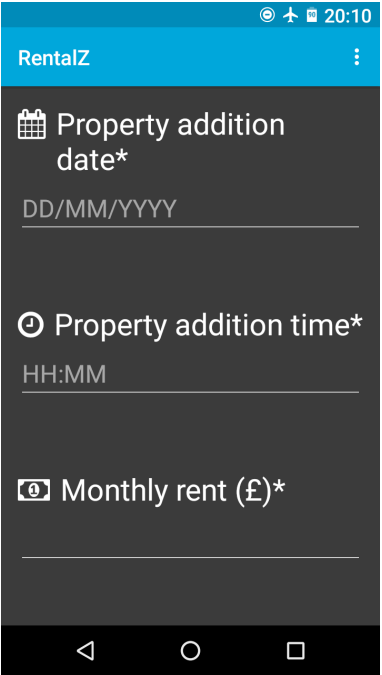


Figure 2: RentalZ form ('Property addition date', 'Property addition time' and 'Monthly rent' fields shown)

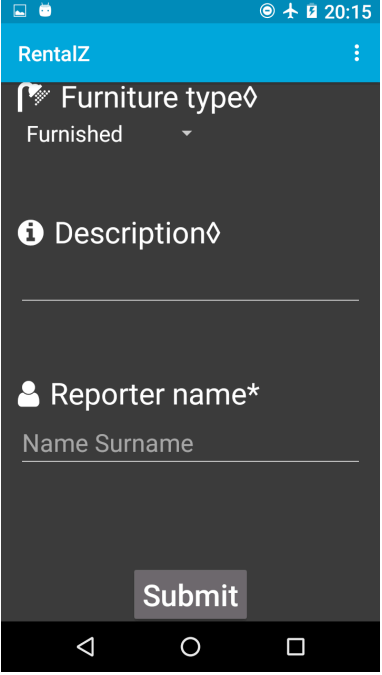
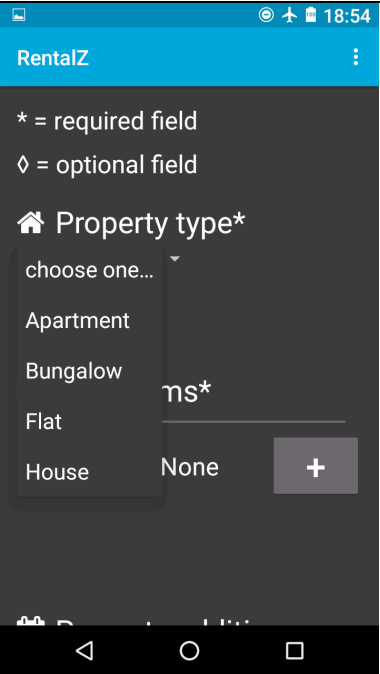
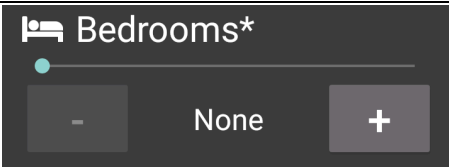
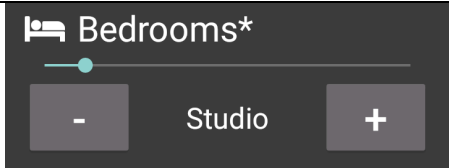
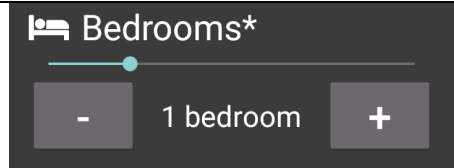
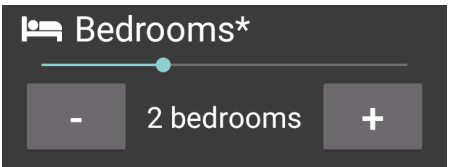
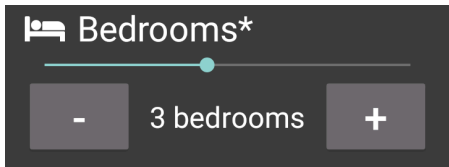
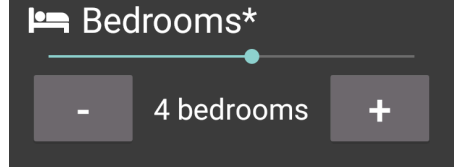
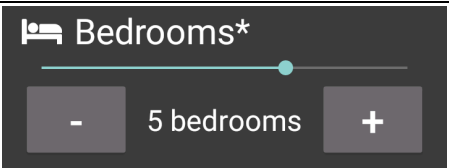
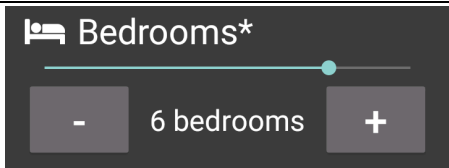
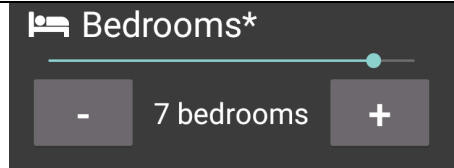
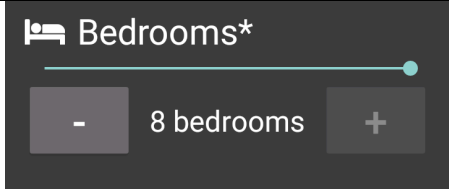


Figure 3: RentalZ form ('Furniture type', 'Description' and 'Reporter name' fields shown)

This is the first screen the user will see when the app is launched. The user will be shown a group of options that can be tweaked, incl. allowing the user to add a property, and showing all data with the database.

 <p>Figure 4: 'Property type' Spinner open</p>	<p>When the user clicks the Spinner widget for 'Property type', they will be presented with a pop-up list of different kinds of properties to choose from (one must be chosen).</p>
---	---

<p>In order to select the desired amount of 'Bedrooms', the user will be presented with a progress bar widget (which the user can adjust by dragging the circle) along with clickable 'plus' and 'minus' buttons (which increment or decrement the bedroom quantity respectively by 1). Either can be done in relation to the amount of bedrooms they want to specify for the property (a quantity must be chosen).</p>		
 <p>Figure 5: 'Bedrooms' field (no bedrooms 'None' option selected)</p>	 <p>Figure 6: 'Bedrooms' field ('Studio' option selected)</p>	 <p>Figure 7: 'Bedrooms' field ('1 bedroom' option selected)</p>
 <p>Figure 8: 'Bedrooms' field ('2 bedrooms' option selected)</p>	 <p>Figure 9: 'Bedrooms' field ('3 bedrooms' option selected)</p>	 <p>Figure 10: 'Bedrooms' field ('4 bedrooms' option selected)</p>
 <p>Figure 11: 'Bedrooms' field ('5 bedrooms' option selected)</p>	 <p>Figure 12: 'Bedrooms' field ('6 bedrooms' option selected)</p>	 <p>Figure 13: 'Bedrooms' field ('7 bedrooms' option selected)</p>
 <p>Figure 14: 'Bedrooms' field ('8 bedrooms' option selected)</p>		

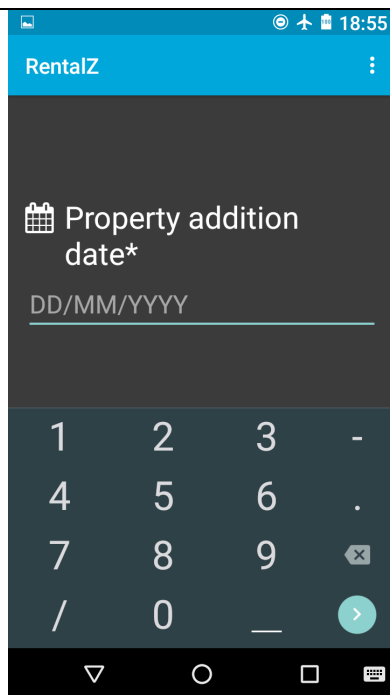


Figure 15: 'Property addition date' EditText selected - 'date' inputType keyboard shown

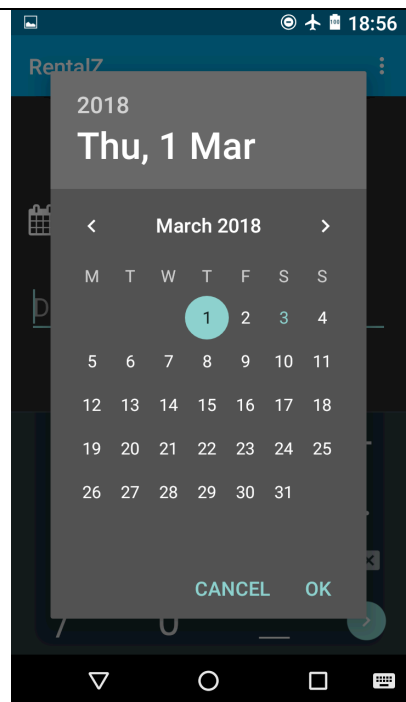


Figure 16: DatePicker widget - current date of device automatically selected

When the user clicks the input field for 'Property addition date', they will be presented with a date-related keyboard, allowing them to type in a date for the property. Alternatively, a second click of that input field will open a DatePicker widget (a date must be chosen).

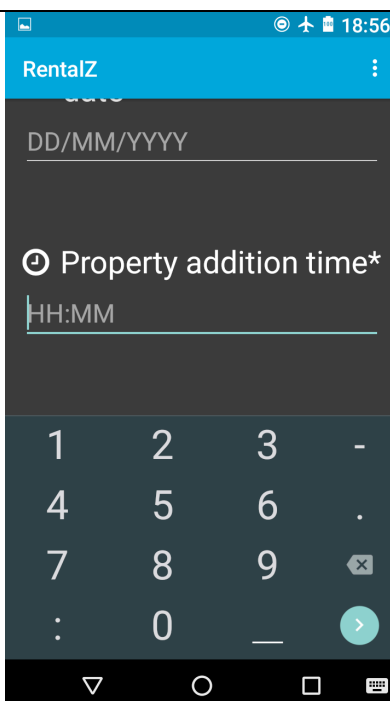


Figure 17: 'Property addition time' EditText selected - 'time' inputType keyboard shown

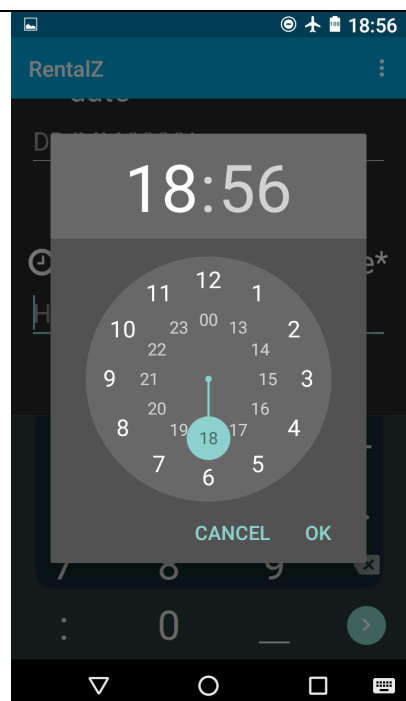


Figure 18: TimePicker widget - current time of device automatically selected

When the user clicks the input field for 'Property addition time', they will be presented with a time-related keyboard, allowing them to type in a time for the property. Alternatively, a second click of that input field will open a TimePicker widget (a time must be chosen).

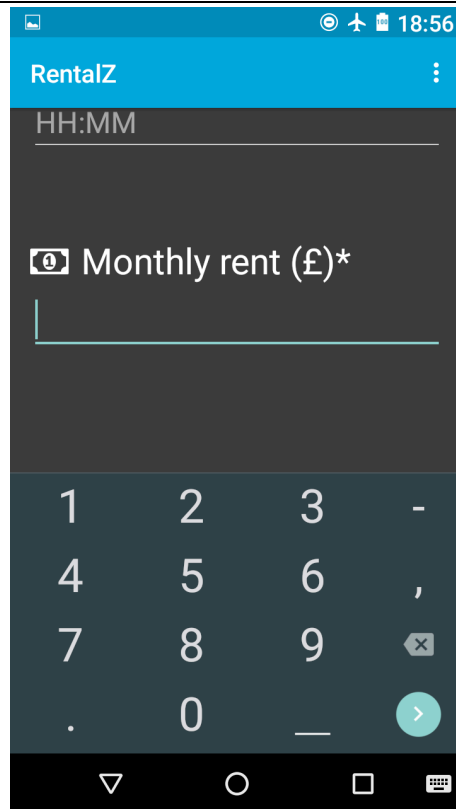


Figure 19: 'Monthly rent' input field selected with 'number' type keyboard open

When the user clicks the input field for 'Monthly rent', they will be presented with a numeric keypad, allowing them to choose a monetary value for the property (a numeric value must be specified).

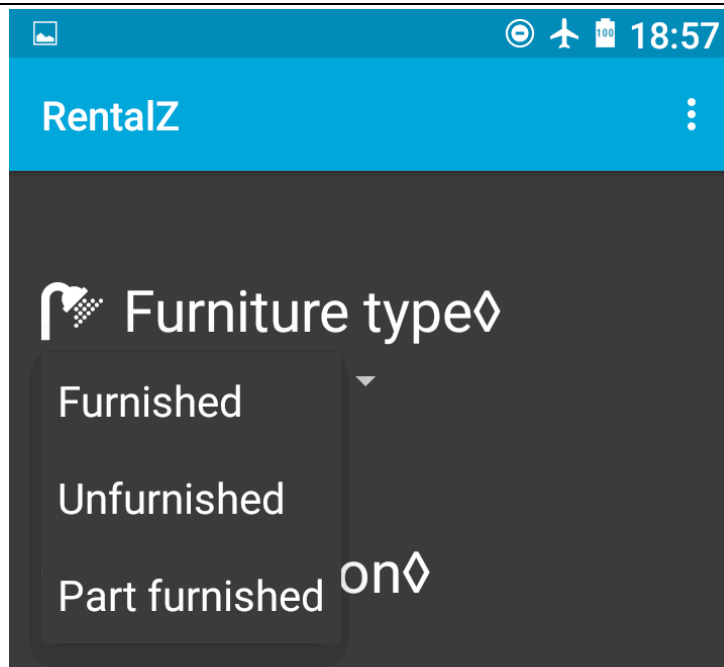


Figure 20: 'Furniture type' select list

When the user clicks the Spinner widget for 'Furniture type', they will be presented with a list of different kinds of furnishing to choose from (this is optional).

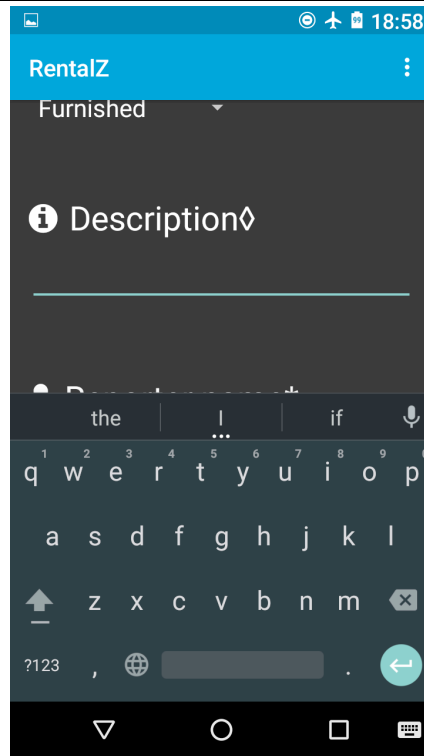


Figure 21: 'Description' input field selected

When the user clicks the input field for 'Description', they will be presented with a QWERTY keyboard, which will allow them add a description of the property (this is optional).

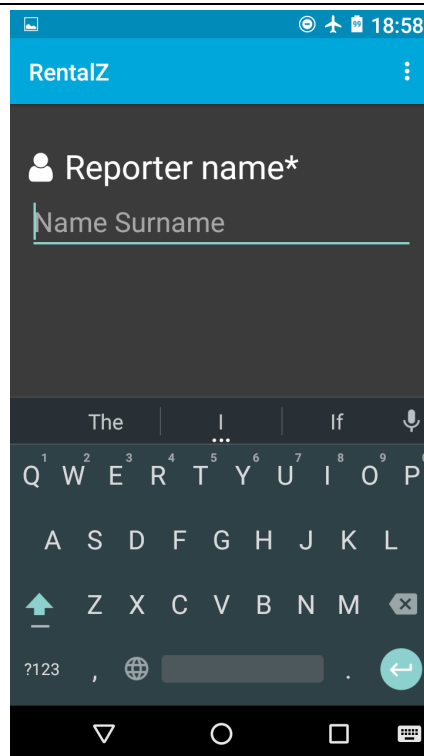


Figure 22: 'Reporter Name' input field selected with 'text' type keyboard open

When the user clicks the input field for 'Reporter Name', they will be presented with a QWERTY keyboard. In addition, this input field contains a placeholder (Name Surname) informing the user what kind of information needs to be added to that field. I.e. a reporter's name and surname (this is required).

When the user clicks the 'Submit' button, the code connected to this button will be executed to check whether or not the required fields meet the specified criteria. If any of them don't, an error message will be displayed to inform the user what must be done in order to rectify the error. NB: The 'Furniture type' and 'Description' fields are optional hence error messages for these fields are not necessary.

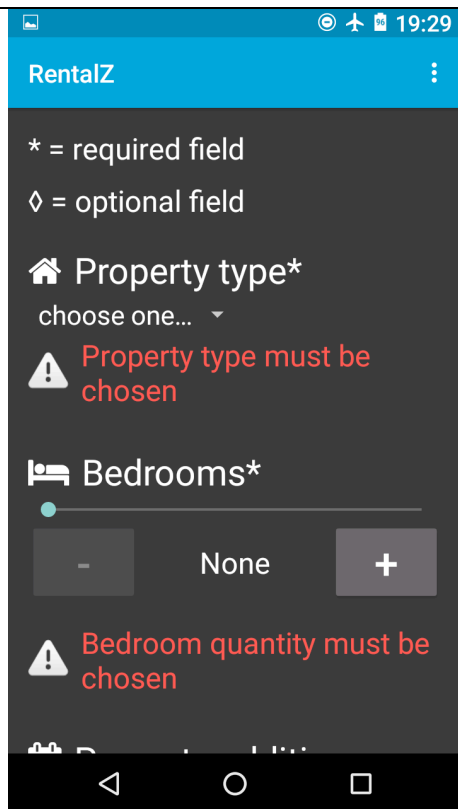


Figure 23: Error messages for 'Property type' and 'Bedrooms' fields

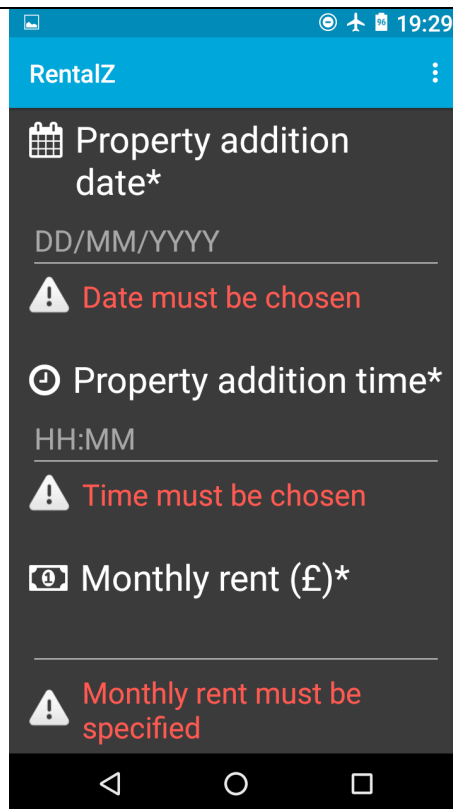


Figure 24: Error messages for 'Property addition date', 'Property addition time' and 'Monthly rent' fields

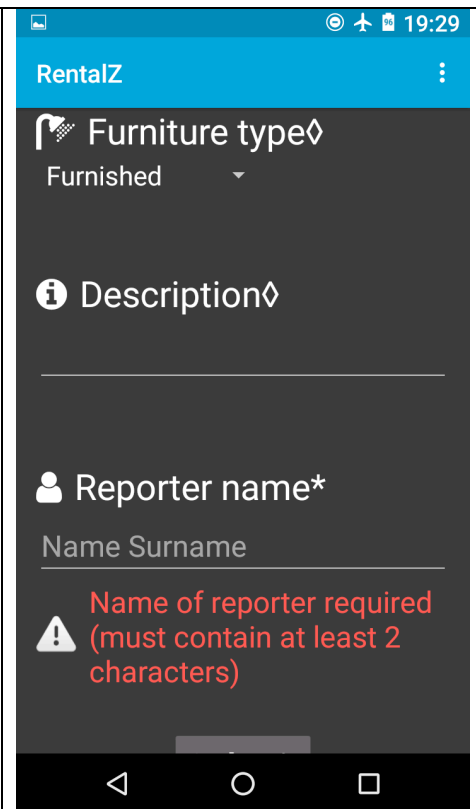


Figure 25: Error message for 'Reporter name' field

2.2 Code that you wrote

File name: layout/activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/mainScrollView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
    <!-- 'android:focusable="true"' and 'android:focusableInTouchMode="true"' stop the EditText
    from gaining focus upon Activity startup -->
    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:paddingLeft="@dimen/activity_horizontal_margin"
        android:paddingRight="@dimen/activity_horizontal_margin"
        android:paddingTop="@dimen/activity_vertical_margin"
        android:paddingBottom="@dimen/activity_vertical_margin"
        android:focusable="true"
        android:focusableInTouchMode="true"
        tools:context="com.fm404.rentalz.MainActivity">

        <!-- 'required field' section -->
        <TextView
            android:id="@+id/textView_required"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
```

```

        android:layout_marginBottom="10dp"
        style="@android:style/TextAppearance.Medium"
        android:text="@string/required_field_notice" />

<!-- 'optional field' section -->
<TextView
    android:id="@+id/textView_optional"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="20dp"
    style="@android:style/TextAppearance.Medium"
    android:text="@string/optional_field_notice" />

<!-- 'Property type' section -->
<!-- 'android:layout_height="0dp"' is used instead of
'android:layout_height="wrap_content"' for better performance -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:orientation="horizontal"
    android:layout_marginBottom="2dp"
    android:layout_weight="1">
    <TextView
        android:id="@+id/imgView_propertytype"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="10dp"
        android:gravity="center"
        style="@android:style/TextAppearance.Large"
        android:text="@string/fa_icon_home" />
    <TextView
        android:id="@+id/textView_propertytype"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:labelFor="@+id/editText_reportername"
        style="@android:style/TextAppearance.Large"
        android:text="@string/property_type" />
</LinearLayout>
<Spinner
    android:id="@+id/spinner_propertytype"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="2dp"
    android:entries="@array/array_propertytype"
    android:prompt="@string/choose_one" />
<TextView
    android:id="@+id/textViewerror_propertytype"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="30dp"
    android:textColor="@android:color/holo_red_light"
    android:drawableStart="@android:drawable/ic_dialog_alert"
    android:drawablePadding="10dp"
    style="@android:style/TextAppearance.Medium"
    android:text="@string/propertytype_required_notice" />

<!-- 'Bedrooms' section -->
<!-- 'android:layout_height="0dp"' is used instead of
'android:layout_height="wrap_content"' for better performance -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:orientation="horizontal"
    android:layout_marginBottom="2dp"
    android:layout_weight="1">
    <TextView
        android:id="@+id/imgView_bedrooms"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="10dp"

```

```

        android:gravity="center"
        style="@android:style/TextAppearance.Large"
        android:text="@string/fa_icon_bed" />
    <TextView
        android:id="@+id/textView_bedrooms"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:labelFor="@+id/spinner_bedrooms"
        style="@android:style/TextAppearance.Large"
        android:text="@string/bedrooms_title" />
</LinearLayout>

<SeekBar
    android:id="@+id/seekbar_bedrooms"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:max="9"
/>

<TableLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:gravity="center"
    android:stretchColumns="1"
    android:focusable="true"
    android:focusableInTouchMode="true">
    <TableRow
        android:id="@+id/MainActivity_tableRow0"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="15dp" >
        <Button
            android:id="@+id/btn_bedroomsminus"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_column="0"
            android:onClick="btnclick_bedroomsminus"
            android:text="-"
            android:textSize="26sp"
            />
        <TextView
            android:id="@+id/textView_seekbarbedroomcount"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_column="1"
            android:layout_gravity="center"
            style="@android:style/TextAppearance.Medium"
            android:text="@string/none" />
        <Button
            android:id="@+id/btn_bedroomsplus"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_column="2"
            android:onClick="btnclick_bedroomsplus"
            android:text="+"
            android:textSize="26sp"
            />
    </TableRow>
</TableLayout>
<!--<Spinner-->
<!--android:id="@+id/spinner_bedrooms"-->
<!--android:layout_width="wrap_content"-->
<!--android:layout_height="wrap_content"-->
<!--android:layout_marginBottom="2dp"-->
<!--android:entries="@array/array_bedroomquantity"-->
<!--android:prompt="@string/choose_bedroomquantity" />-->

<TextView
    android:id="@+id/textViewerror_bedrooms"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"

```



```

        android:layout_marginBottom="30dp"
        android:textColor="@android:color/holo_red_light"
        android:drawableStart="@android:drawable/ic_dialog_alert"
        android:drawablePadding="10dp"
        style="@android:style/TextAppearance.Medium"
        android:text="@string/bedrooms_required_notice" />

<!-- 'Property addition date' section -->
<!-- 'android:layout_height="0dp"' is used instead of
'android:layout_height="wrap_content"' for better performance -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:orientation="horizontal"
    android:layout_marginBottom="2dp"
    android:layout_weight="1">
    <TextView
        android:id="@+id/imgView_propertadditiondate"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="10dp"
        android:gravity="center"
        style="@android:style/TextAppearance.Large"
        android:text="@string/fa_icon_calendar_alt" />
    <TextView
        android:id="@+id/textView_propertyadditiondate"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:labelFor="@+id/editText_date"
        style="@android:style/TextAppearance.Large"
        android:text="@string/property_addition_date" />
</LinearLayout>
<EditText
    android:id="@+id/editText_date"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="2dp"
    android:maxLength="10"
    android:hint="@string/hint_date"
    android:inputType="date" />
<TextView
    android:id="@+id/textViewerror_date"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="30dp"
    android:textColor="@android:color/holo_red_light"
    android:drawableStart="@android:drawable/ic_dialog_alert"
    android:drawablePadding="10dp"
    style="@android:style/TextAppearance.Medium"
    android:text="@string/date_required_notice" />

<!-- 'Property addition time' section -->
<!-- 'android:layout_height="0dp"' is used instead of
'android:layout_height="wrap_content"' for better performance -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:orientation="horizontal"
    android:layout_marginBottom="2dp"
    android:layout_weight="1">
    <TextView
        android:id="@+id/imgView_propertadditiontime"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="10dp"
        android:gravity="center"
        style="@android:style/TextAppearance.Large"
        android:text="@string/fa_icon_clock_o" />
    <TextView
        android:id="@+id/textView_propertyadditiontime"

```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:labelFor="@+id/editText_time"
        style="@android:style/TextAppearance.Large"
        android:text="@string/property_addition_time" />
</LinearLayout>
<EditText
    android:id="@+id/editText_time"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="2dp"
    android:maxLength="5"
    android:hint="@string/hint_time"
    android:inputType="time" />
<TextView
    android:id="@+id/textViewerror_time"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="30dp"
    android:textColor="@android:color/holo_red_light"
    android:drawableStart="@android:drawable/ic_dialog_alert"
    android:drawablePadding="10dp"
    style="@android:style/TextAppearance.Medium"
    android:text="@string/time_required_notice" />

<!-- 'Monthly rent price' section -->
<!-- 'android:layout_height="0dp"' is used instead of
'android:layout_height="wrap_content"' for better performance -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:orientation="horizontal"
    android:layout_marginBottom="2dp"
    android:layout_weight="1">
    <TextView
        android:id="@+id/imgView_monthlyrent"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="10dp"
        android:gravity="center"
        style="@android:style/TextAppearance.Large"
        android:text="@string/fa_icon_money" />
    <TextView
        android:id="@+id/textView_monthlyrent"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:labelFor="@+id/editText_monthlyrent"
        style="@android:style/TextAppearance.Large"
        android:text="@string/monthly_rent" />
</LinearLayout>
<EditText
    android:id="@+id/editText_monthlyrent"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="2dp"
    android:inputType="numberDecimal" />
<TextView
    android:id="@+id/textViewerror_monthlyrent"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="30dp"
    android:textColor="@android:color/holo_red_light"
    android:drawableStart="@android:drawable/ic_dialog_alert"
    android:drawablePadding="10dp"
    style="@android:style/TextAppearance.Medium"
    android:text="@string/monthlyrent_required_notice" />

<!-- 'Furniture type' section -->
<!-- 'android:layout_height="0dp"' is used instead of
'android:layout_height="wrap_content"' for better performance -->

```

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:orientation="horizontal"
    android:layout_marginBottom="2dp"
    android:layout_weight="1">
    <TextView
        android:id="@+id/imgView_furnituretype"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="10dp"
        android:gravity="center"
        style="@android:style/TextAppearance.Large"
        android:text="@string/fa_icon_shower" />
    <TextView
        android:id="@+id/textView_furnituretype"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:labelFor="@+id/spinner_furnituretype"
        style="@android:style/TextAppearance.Large"
        android:text="@string/furniture_type" />
</LinearLayout>
<Spinner
    android:id="@+id/spinner_furnituretype"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginBottom="60dp"
    android:entries="@array/array_furnituretype"
    android:prompt="@string/choose_furnituretype" />

<!-- 'Description' section -->
<!-- 'android:layout_height="0dp"' is used instead of
'android:layout_height="wrap_content"' for better performance -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"
    android:orientation="horizontal"
    android:layout_marginBottom="2dp"
    android:layout_weight="1">
    <TextView
        android:id="@+id/imgView_description"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginEnd="10dp"
        android:gravity="center"
        style="@android:style/TextAppearance.Large"
        android:text="@string/fa_icon_info_circle" />
    <TextView
        android:id="@+id/textView_description"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:labelFor="@+id/editText_description"
        style="@android:style/TextAppearance.Large"
        android:text="@string/description" />
</LinearLayout>
<EditText
    android:id="@+id/editText_description"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginBottom="60dp"
    android:inputType="textMultiLine"
    android:minLines="1"
    android:maxLines="2"/>

<!-- 'Reporter name' section -->
<!-- 'android:layout_height="0dp"' is used instead of
'android:layout_height="wrap_content"' for better performance -->
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="0dp"

```

```

        android:orientation="horizontal"
        android:layout_marginBottom="2dp"
        android:layout_weight="1">
        <TextView
            android:id="@+id/imgView_reportername"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginEnd="10dp"
            android:gravity="center"
            style="@android:style/TextAppearance.Large"
            android:text="@string/fa_icon_user" />
        <TextView
            android:id="@+id/textView_reportername"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:labelFor="@+id/editText_reportername"
            style="@android:style/TextAppearance.Large"
            android:text="@string/reporter_name" />
    </LinearLayout>
    <!-- android:inputType="textPersonName" - ensures red underline does not appear under
names entered -->
    <!-- android:inputType="textCapWords" - ensures the first letter of each word is
capitalised -->
    <EditText
        android:id="@+id/editText_reportername"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="2dp"
        android:digits=""
        android:hint="@string/hint_reportername"
        android:inputType="textPersonName|textCapWords" />
    <TextView
        android:id="@+id/textViewerror_reportername"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginBottom="30dp"
        android:textColor="@android:color/holo_red_light"
        android:drawableStart="@android:drawable/ic_dialog_alert"
        android:drawablePadding="10dp"
        style="@android:style/TextAppearance.Medium"
        android:text="@string/reportername_required_notice" />

    <!-- 'Submit' button-->
    <Button
        android:id="@+id/btn_submit"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center_horizontal"
        android:layout_marginBottom="30dp"
        android:contentDescription="@string/submit"
        android:text="@string/submit"
        android:textAllCaps="false"
        android:textSize="22sp"/>
</LinearLayout>
</ScrollView>

```

File name: DatePickerFragment.java

```

public class DatePickerFragment extends DialogFragment {

    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        Calendar c = Calendar.getInstance();
        int year = c.get(Calendar.YEAR);
        int month = c.get(Calendar.MONTH);
        int day = c.get(Calendar.DAY_OF_MONTH);

        return new DatePickerDialog(getActivity(), (DatePickerDialog.OnDateSetListener)
getActivity(), year, month, day);
    }
}

```

File name: TimePickerFragment.java

```
public class TimePickerFragment extends DialogFragment {

    @Override
    public Dialog onCreateDialog(Bundle savedInstanceState) {
        Calendar c = Calendar.getInstance();
        int hour = c.get(Calendar.HOUR_OF_DAY);
        int minute = c.get(Calendar.MINUTE);

        return new TimePickerDialog(getActivity(), (TimePickerDialog.OnTimeSetListener)
getActivity(), hour, minute, DateFormat.is24HourFormat(getActivity()));
    }
}
```

File name: FontManager.java

```
public class FontManager {

    static final String ROOT = "fonts/", FONTAWESOME = ROOT + "fontawesome-webfont.ttf";

    static Typeface getTypeface(Context context, String font) {
        return Typeface.createFromAsset(context.getAssets(), font);
    }
}
```

File name: MainActivity.java

```
public class MainActivity extends Activity implements DatePickerDialog.OnDateSetListener,
TimePickerDialog.OnTimeSetListener {
    private Button btnBedroomsMinus;
    private Button btnBedroomsPlus;
    private EditText edittextPropertyAdditionDate;
    private EditText edittextPropertyAdditionTime;
    private EditText edittextMonthlyRent;
    private EditText edittextDescription;
    private EditText edittextReporterName;
    private SeekBar seekbarBedrooms;
    private Spinner spinnerPropertyType;
    private TextView textviewErrorPropertyType;
    private TextView textviewErrorBedrooms;
    private TextView textviewErrorPropertyAdditionDate;
    private TextView textviewErrorPropertyAdditionTime;
    private TextView textviewErrorMonthlyRent;
    private TextView textviewErrorReporterName;
    private TextView textviewSeekBarBedrooms;
    private int requestCode_takephoto = 1;
    private int requestCode_choosephoto = 2;

    private int getFactorColor(int color, float factor) {
        float[] hsv = new float[3];
        Color.colorToHSV(color, hsv);
        hsv[2] *= factor;
        color = Color.HSVToColor(hsv);
        return color;
    }

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

        final String toolbarColour = "#0098D4";

        //set status bar background colour
        Window window = getWindow();
        window.addFlags(WindowManager.LayoutParams.FLAG_DRAWS_SYSTEM_BAR_BACKGROUNDS);
        getWindow().setStatusBarColor(getFactorColor(Color.parseColor(toolbarColour), 0.8f));

        //set action bar background colour
        ActionBar actionBar = getActionBar();

        //Setting up Action bar color using # color code.
        actionBar.setBackgroundDrawable(new ColorDrawable(Color.parseColor(toolbarColour)));
    }
}
```

```

        Typeface iconFont = FontManager.getTypeface(getApplicationContext(),
FontManager.FONTAWESOME); //FontManager class must be accessed first before text views can be
set as image views
        TextView tvimgviewPropertyType = findViewById(R.id.imgView_propertytype); //get access
to the 'Property Type' ImageView (text view using image font)
        tvimgviewPropertyType.setTypeface(iconFont); //set typeface of textview to FontAwesome
font (to enable use of vector images in TextView)
        TextView tvimgviewBedrooms = findViewById(R.id.imgView_bedrooms); //get access to the
'Bedrooms' ImageView (text view using image font)
        tvimgviewBedrooms.setTypeface(iconFont); //set typeface of textview to FontAwesome font
(to enable use of vector images in TextView)
        TextView tvimgviewPropertyAdditionDate = findViewById(R.id.imgView_propertadditiondate);
//get access to the 'Property Addition Date' ImageView (text view using image font)
        tvimgviewPropertyAdditionDate.setTypeface(iconFont); //set typeface of textview to
FontAwesome font (to enable use of vector images in TextView)
        TextView tvimgviewPropertyAdditionTime = findViewById(R.id.imgView_propertadditiontime);
//get access to the 'Property Addition Time' ImageView (text view using image font)
        tvimgviewPropertyAdditionTime.setTypeface(iconFont); //set typeface of textview to
FontAwesome font (to enable use of vector images in TextView)
        TextView tvimgviewMonthlyRent = findViewById(R.id.imgView_monthlyrent); //get access to
the 'Monthly Rent' ImageView (text view using image font)
        tvimgviewMonthlyRent.setTypeface(iconFont); //set typeface of textview to FontAwesome
font (to enable use of vector images in TextView)
        TextView tvimgviewFurnitureType = findViewById(R.id.imgView_furnituretype); //get access
to the 'Furniture Type' ImageView (text view using image font)
        tvimgviewFurnitureType.setTypeface(iconFont); //set typeface of textview to FontAwesome
font (to enable use of vector images in TextView)
        TextView tvimgviewDescription = findViewById(R.id.imgView_description); //get access to
the 'Description' ImageView (text view using image font)
        tvimgviewDescription.setTypeface(iconFont); //set typeface of textview to FontAwesome
font (to enable use of vector images in TextView)
        TextView tvimgviewReporterName = findViewById(R.id.imgView_reportername); //get access
to the 'Reporter Name' ImageView (text view using image font)
        tvimgviewReporterName.setTypeface(iconFont); //set typeface of textview to FontAwesome
font (to enable use of vector images in TextView)

        btnBedroomsMinus = findViewById(R.id.btn_bedroomsminus); //get access to the 'Bedrooms
Minus' button
        btnBedroomsPlus = findViewById(R.id.btn_bedroomsplus); //get access to the 'Bedrooms
Plus' button
        editTextPropertyAdditionDate = findViewById(R.id.editText_date); //get access to the
'Property Addition Date' EditText
        editTextPropertyAdditionTime = findViewById(R.id.editText_time); //get access to the
'Property Addition Time' EditText
        editTextMonthlyRent = findViewById(R.id.editText_monthlyrent); //get access to the
'Monthly Rent' EditText
        editTextDescription = findViewById(R.id.editText_description); //get access to the
'Notes' EditText
        editTextReporterName = findViewById(R.id.editText_reportername); //get access to the
'Reporter Name' EditText
        seekbarBedrooms = findViewById(R.id.seekbar_bedrooms); //get access to the 'Bedrooms'
SeekBar
        spinnerPropertyType = findViewById(R.id.spinner_propertytype); //get access to the
'Property Type' Spinner
        textViewErrorPropertyType = findViewById(R.id.textViewerror_propertytype); //get access
to the 'textViewerror_propertytype' TextView
        textViewErrorBedrooms = findViewById(R.id.textViewerror_bedrooms); //get access to the
'textViewerror_bedrooms' TextView
        textViewErrorPropertyAdditionDate = findViewById(R.id.textViewerror_date); //get access
to the 'textViewerror_date' TextView
        textViewErrorPropertyAdditionTime = findViewById(R.id.textViewerror_time); //get access
to the 'textViewerror_time' TextView
        textViewErrorMonthlyRent = findViewById(R.id.textViewerror_monthlyrent); //get access to
the 'textViewerror_monthlyrent' TextView
        textViewErrorReporterName = findViewById(R.id.textViewerror_reportername); //get access
to the 'textViewerror_reportername' TextView
        textViewSeekBarBedrooms = findViewById(R.id.textView_seekbarbedroomcount); //get access
to the 'textView_seekbarbedroomcount' TextView

```

```

        textViewErrorPropertyType.setVisibility(View.INVISIBLE); //make
'textViewerror_propertytype' text view INVISIBLE (not GONE) to keep space between items
        textViewErrorBedrooms.setVisibility(View.INVISIBLE); //make 'textViewerror_bedrooms'
text view INVISIBLE (not GONE) to keep space between items
        textViewErrorPropertyAdditionDate.setVisibility(View.INVISIBLE); //make
'textViewerror_propertytype' text view INVISIBLE (not GONE) to keep space between items
        textViewErrorPropertyAdditionTime.setVisibility(View.INVISIBLE); //make
'textViewErrorPropertyAdditionTime' text view INVISIBLE (not GONE) to keep space between items
        textViewErrorMonthlyRent.setVisibility(View.INVISIBLE); //make
'textViewErrorMonthlyRent' text view INVISIBLE (not GONE) to keep space between items
        textViewErrorReporterName.setVisibility(View.INVISIBLE); //make
'textViewErrorReporterName' text view INVISIBLE (not GONE) to keep space between items

        btnBedroomsMinus.setEnabled(false); //disable 'Bedrooms Minus' button when activity is
first created

        //callback to notify the user when the progress level has been changed
        seekbarBedrooms.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
            @Override
            public void onProgressChanged(SeekBar seekBar, int progress, boolean fromUser) {
                if (progress == 2) {
                    textViewSeekBarBedrooms.setText((progress - 1) + " " +
getString(R.string.bedroom));
                } else if (progress == 1) {
                    textViewSeekBarBedrooms.setText(getString(R.string.studio));
                    btnBedroomsMinus.setEnabled(true);
                } else if (progress == 0) {
                    textViewSeekBarBedrooms.setText(getString(R.string.none));
                    btnBedroomsMinus.setEnabled(false);
                } else if (progress == 9) {
                    textViewSeekBarBedrooms.setText((progress - 1) + " " +
getString(R.string.bedrooms));
                    btnBedroomsPlus.setEnabled(false);
                } else {
                    textViewSeekBarBedrooms.setText((progress - 1) + " " +
getString(R.string.bedrooms));
                    btnBedroomsMinus.setEnabled(true);
                    btnBedroomsPlus.setEnabled(true);
                }
            }

            @Override
            public void onStartTrackingTouch(SeekBar seekBar) {

            }

            @Override
            public void onStopTrackingTouch(SeekBar seekBar) {

            }
        });

        //click event for 'submit' button
        Button button0 = this.findViewById(R.id.btn_submit);
        button0.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) { //Perform all of the following functions when the
'submit' button is clicked

                hideKeyboard(); //hide the on-screen keyboard, because the Android system
doesn't do this automatically

                //'Property type' Spinner decision making
                if
(spinnerPropertyType.getSelectedItem().toString().contains(getString(R.string.choose_one))) {
//if the selected item of 'spinnerPropertytype' spinner is 'choose bedroom quantity...',
                textViewErrorPropertyType.setVisibility(View.VISIBLE); //...make the
'textViewErrorPropertyType' text view visible
                } else {
                    textViewErrorPropertyType.setVisibility(View.INVISIBLE); //otherwise, make
the 'textViewErrorPropertyType' text view invisible

```



```

    }

    //'Bedrooms' SeekBar decision making
    if (seekbarBedrooms.getProgress() == 0) { //if the progress of 'seekbarBedrooms'
SeekBar is '0',
        textViewErrorBedrooms.setVisibility(View.VISIBLE); //make the
'textViewErrorBedrooms' text view visible
    } else {
        textViewErrorBedrooms.setVisibility(View.INVISIBLE); //otherwise, make the
'textViewErrorBedrooms' text view invisible
    }

    //'Property addition date' EditText decision making
    if (edittextPropertyAdditionDate.getText().length() == 0) { //if the
'edittextPropertyAdditionDate' EditText field is empty (contains 0 characters),
        textViewErrorPropertyAdditionDate.setVisibility(View.VISIBLE); //make the
'textViewErrorPropertyAdditionDate' text view visible
    } else {
        textViewErrorPropertyAdditionDate.setVisibility(View.INVISIBLE);
//otherwise, make the 'textViewErrorPropertyAdditionDate' text view invisible
    }

    //'Property addition time' EditText decision making
    if (edittextPropertyAdditionTime.getText().length() == 0) { //if the
'edittextPropertyAdditionTime' EditText field is empty (contains 0 characters),
        textViewErrorPropertyAdditionTime.setVisibility(View.VISIBLE); //make the
'textViewErrorPropertyAdditionTime' text view visible
    } else {
        textViewErrorPropertyAdditionTime.setVisibility(View.INVISIBLE);
//otherwise, make the 'textViewErrorPropertyAdditionTime' text view invisible
    }

    //'Monthly rent' EditText decision making
    if (edittextMonthlyRent.getText().length() == 0) { //if the
'edittextMonthlyRent' EditText field is empty (contains 0 characters),
        textViewErrorMonthlyRent.setVisibility(View.VISIBLE); //make the
'textViewErrorMonthlyRent' text view visible
    } else {
        textViewErrorMonthlyRent.setVisibility(View.INVISIBLE); //otherwise, make
the 'textViewErrorMonthlyRent' text view invisible
    }

    //'Reporter name' EditText decision making
    if (edittextReporterName.getText().length() < 2) { //if the
'edittextReporterName' EditText field has less than 2 characters,
        textViewErrorReporterName.setVisibility(View.VISIBLE); //make the
'textViewErrorReporterName' text view visible
    } else {
        textViewErrorReporterName.setVisibility(View.INVISIBLE); //otherwise, make
the 'textViewErrorReporterName' text view invisible
    }
}

});

//click event for 'edittextPropertyAdditionDate' EditText (shows DatePicker dialog)
edittextPropertyAdditionDate.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        DialogFragment datePicker = new DatePickerFragment();
        datePicker.show(getFragmentManager(), "date picker");
    }
});

//click event for 'edittextPropertyAdditionTime' EditText (shows TimePicker dialog)
edittextPropertyAdditionTime.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        DialogFragment timePicker = new TimePickerFragment();
        timePicker.show(getFragmentManager(), "time picker");
    }
});
}

```



```

@Override
public void onDateSet(DatePicker view, int year, int month, int dayOfMonth) {
    Calendar c = Calendar.getInstance(); //new Calendar instance
    c.set(Calendar.YEAR, year); //show current year of device (data passed by 'int year'
variable)
    c.set(Calendar.MONTH, month); //show current month of device (data passed by 'int month'
variable)
    c.set(Calendar.DAY_OF_MONTH, dayOfMonth); //show current day (of the month) of device
(data passed by 'int dayOfMonth' variable)
    String currentDateString =
DateFormat.getDateInstance(DateFormat.SHORT).format(c.getTime()); // use the short date format
when displaying the date

    editTextPropertyAdditionDate.setText(currentDateString); //set text of
'edittextPropertyAdditionDate' EditText to the 'currentDateString' variable
}

@Override
public void onTimeSet(TimePicker view, int hourOfDay, int minute) {
    editTextPropertyAdditionTime.setText(String.format("%02d:%02d", hourOfDay, minute));
//set text of 'edittextPropertyAdditionDate' EditText to the 'currentDateString' variable
// '%02d:%02d' prevents any zeros from being omitted upon time selection
}

private void hideKeyboard() {
    View view = getCurrentFocus();
    if (view != null) {
        ((InputMethodManager) getSystemService(Context.INPUT_METHOD_SERVICE)).
hideSoftInputFromWindow(view.getWindowToken(), InputMethodManager.HIDE_NOT_ALWAYS);
    }
}

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    //Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu_main, menu);

    //Return true so Android will know that we want to display the menu
    return true;
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    //Handle item selection
    switch (item.getItemId()) {
        case R.id.action_reset:
            //User chose the "Reset" item, show confirm dialog ...

            AlertDialog.Builder adb = new AlertDialog.Builder(this);
            adb.setTitle(getString(R.string.reset_form)); //Title of alert dialog
            adb.setIcon(android.R.drawable.ic_dialog_alert); //Icon of alert dialog (warning
sign)

            adb.setCancelable(false); //Prevent user from tapping outside dialog to cancel
            adb.setMessage(getString(R.string.reset_warning)); //Message of alert dialog

            adb.setPositiveButton(getString(R.string.ok), new
DialogInterface.OnClickListener() {
                public void onClick(DialogInterface dialog, int which) {
                    Toast.makeText(MainActivity.this, "Positive selected",
Toast.LENGTH_SHORT).show();

                    spinnerPropertyType.setSelection(0); // set 'Property Type' Spinner
position to 0
                    seekbarBedrooms.setProgress(0); // set progress of 'Bedrooms' SeekBar to
0
                    editTextPropertyAdditionDate.setText(""); //Clear text from 'Property
Addition Date' EditText field
                    editTextPropertyAdditionTime.setText(""); //Clear text from 'Property
Addition Time' EditText field
                    editTextMonthlyRent.setText(""); //Clear text from 'Monthly Rent'

```

```

EditText field
    edittextDescription.setText(""); //Clear text from 'Notes' EditText
field
    edittextReporterName.setText(""); //Clear text from 'Reporter Name'
EditText field
    textviewErrorPropertyType.setVisibility(View.INVISIBLE); //make
'textViewerror_propertytype' text view INVISIBLE (not GONE) to keep space between items
    textviewErrorBedrooms.setVisibility(View.INVISIBLE); //make
'textViewerror_bedrooms' text view INVISIBLE (not GONE) to keep space between items
    textviewErrorPropertyAdditionDate.setVisibility(View.INVISIBLE); //make
'textViewerror_propertytype' text view INVISIBLE (not GONE) to keep space between items
    textviewErrorPropertyAdditionTime.setVisibility(View.INVISIBLE); //make
'textViewErrorPropertyAdditionTime' text view INVISIBLE (not GONE) to keep space between items
    textviewErrorMonthlyRent.setVisibility(View.INVISIBLE); //make
'textViewErrorMonthlyRent' text view INVISIBLE (not GONE) to keep space between items
    textviewErrorReporterName.setVisibility(View.INVISIBLE); //make
'textViewErrorReporterName' text view INVISIBLE (not GONE) to keep space between items
    }
    });

    adb.setNegativeButton(getString(R.string.cancel), new
DialogInterface.OnClickListener() {
        public void onClick(DialogInterface dialog, int which) {
            Toast.makeText(MainActivity.this, "Negative selected",
Toast.LENGTH_SHORT).show();
            dialog.dismiss(); //Dismiss the dialog
        }
    });

    adb.show(); //Show the alert dialog

    return true;

default:
    //If we got here, the user's action was not recognized.
    //Invoke the superclass to handle it.
    return super.onOptionsItemSelected(item);
}
}

public void btnclick_bedroomsminus(View view) {
    if (seekbarBedrooms.getProgress() == 1) {
        seekbarBedrooms.setProgress(seekbarBedrooms.getProgress() - 1);
        btnBedroomsMinus.setEnabled(false);
    } else {
        btnBedroomsMinus.setEnabled(true);
        btnBedroomsPlus.setEnabled(true);
        seekbarBedrooms.setProgress(seekbarBedrooms.getProgress() - 1);
    }
}

public void btnclick_bedroomsplus(View view) {
    if (seekbarBedrooms.getProgress() == 8) {
        seekbarBedrooms.setProgress(seekbarBedrooms.getProgress() + 1);
        btnBedroomsPlus.setEnabled(false);
    } else {
        btnBedroomsMinus.setEnabled(true);
        btnBedroomsPlus.setEnabled(true);
        seekbarBedrooms.setProgress(seekbarBedrooms.getProgress() + 1);
    }
}
}
}

```