

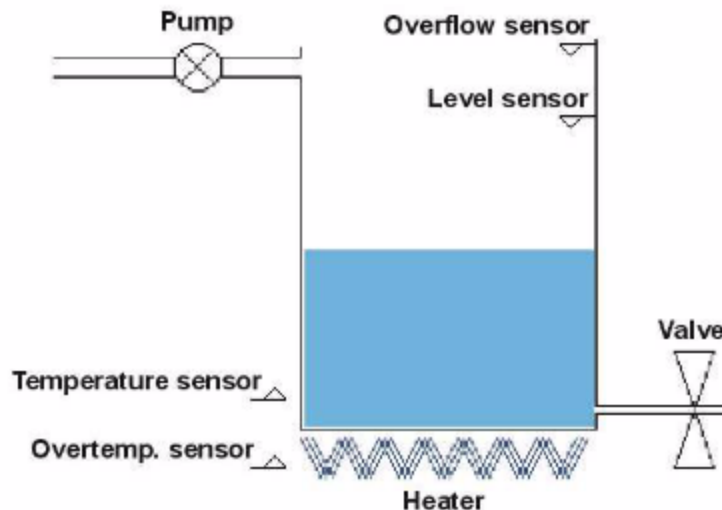
Lab 6 – GUI development and implementation I

The purpose of this lab is to get more familiar to GUI – Graphical User Interface, understand what app's user interface is about. You will get familiar with different build UI components, UI controls and UI models that help you to build a nice GUI. Some of which you already know. The Water Tank Controller is again represented to you. You should do the design of the model.

1.1 Brief introduction to GUI programming, Water Tank Controller

For the labs concerning this course you are supposed to implement a brief model of the Water Tank Controller. For that reason in this lab you will have to do more practice about GUI programming and play around with different components that will be useful to you for the design of the Water Tank Controller and their implementation in the Socket Programming lab.

The model of the Water Tank Controller that is familiar to you is shown below. You should think how to implement it in Android in terms of representing the temperature, pressure etc. Think what kind of Android components can be used?



Android provide you with different pre-build UI components which help you to build the graphical user interface. It provides you with UI controls, UI modules for special interfaces (like dialogs, notifications and menus) and structured layout objects.

More read on: <http://developer.android.com/guide/topics/ui/index.html>

When building an Android app you will spend lots of time on perfecting the UI (User interface). It is important that you will spend some time on your own and try to understand how the things work.

1.2 UI Screen Components

Layouts - A Generic ViewGroup that defines a structure for the Views it contains

- **Types of layouts:**
 - Linear Layout
 - Absolute Layout
 - Table Layout
 - Frame Layout
 - Relative Layout

LinearLayout – Child views arranged in a single horizontal or vertical row.

AbsoluteLayout - specifies the exact location of the components.

TableLayout – Child views arranged into rows and columns

FrameLayout - is a placeholder on screen that you can use to display a single view.

RelativeLayout – Child views are positioned relative to each other and to parent view

You can find more about layout on the following link:

http://www.tutorialspoint.com/android/android_ui_deign.htm

More about **Building the UI** you can find on the following link:

<http://www.i-programmer.info/programming/android/5937-android-adventures-building-the-ui.html>

Note about your future work:

Since you already know about the Water Tank Controller and you know how it looks like, your job will be to create a nice user interface which means a good looking model of the Water Tank Controller and be able to realize a connection between a server and a client. For the connection you will use Socket programming. With more information and details you will be provided later.

1.3 Some predefined views

- Button
- ToggleButton
- CheckBox
- RatingBar
- AutoCompleteTextView

- **UIButton**

```
<Button
    android:id="@+id/button"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_marginLeft="10dip"
    android:text="@string/press_me_string" >
</Button>
```

1. Get a reference to the Press Me Button
2. Set an **onClickListener** on this button. This listener is called each time the user clicks the Button
3. Count the number of times the user click the button inside the **onClickListener()** in the **onClick()** method
`Button.setText("Got pressed:" + ++count);`



- **UIToggleButton**

```
<ToggleButton
    android:id="@+id/togglebutton"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textOff="@string/start_string"
    android:textOn="@string/stop_string"
    android:textSize="24sp" />
```

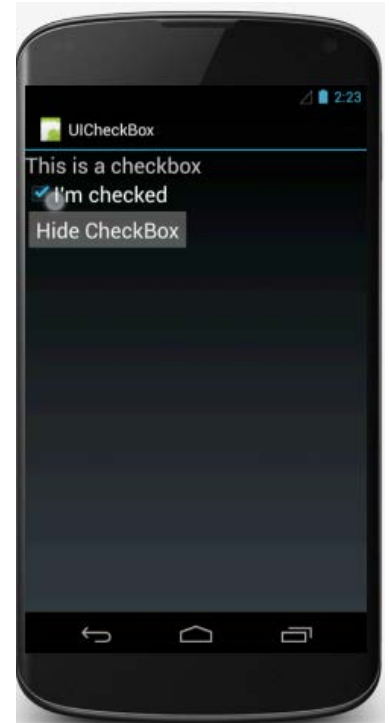
1. Get a reference to the ToggleButton
2. Set an **onClickListener()** on the ToggleButton
3. Toggle the background color between a light and a dark color in the **onClickListener()** in the **onClick()** method
`if(button.isChecked()){
 bg.setBackgroundColor(0xFF3F3F3F);
}else{
 Bg.setBackgroundColor(0xFF000000);
}`



- **UICheckBox**

```
<CheckBox  
    android:id="@+id/checkbox"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="@string/im_not_checked_string"  
    android:textSize="24sp"/>
```

1. Get a reference to the CheckBox
2. Set **onClickListener** on the CheckBox
3. In the **onClickListener** in the **onClick()** method check whether the CheckBox is currently checked. Set the text properly.



1.4 Extra

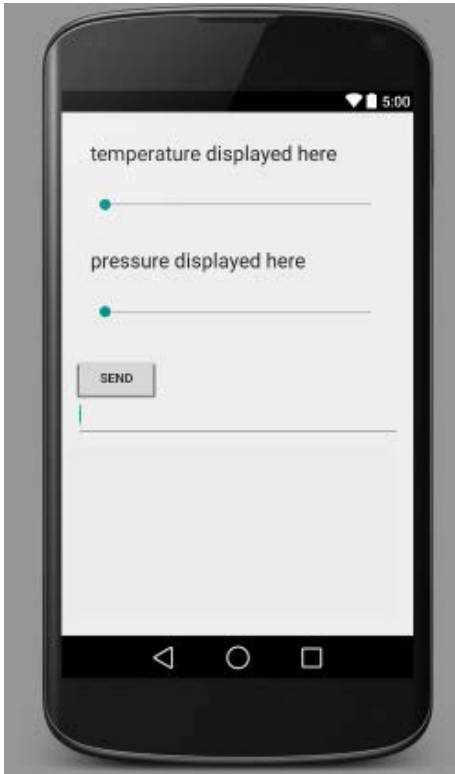
- **UIAutoCompleteTextView**

```
<AutoCompleteTextView  
    android:id="@+id/autocomplete_country"  
    android:layout_width="fill_parent"  
    android:layout_height="wrap_content"  
    android:layout_marginLeft="5dp"  
    android:textSize="24sp"/>
```

1. Get a reference to the AutoCompleteTextView
2. Create an ArrayAdapter containing country names
ArrayAdapter<String> adapter = new ArrayAdapter<String>(this, R.layout.list_item, COUNTRIES);
3. Set the adapter for the AutoCompleteTextView
textView.setAdapter(adapter);



1.5 Extension



The UI components you learned so far can be useful when modelling the Water Tank Controller.

If you still feel not comfortable with the UI you can read more about each element in the official Android documentation as well as other online resources.

On the picture is what you are going to implement. In order to represent the changing of the temperature and pressure of the Water Tank Controller, two **sliders** are used.

Another component, **EditText** is used. Here you can enter any string you want. This string later, is supposed to be sent to the client using the **Button**.

- **Task:**

Use your knowledge from the previous lab. Create a new activity where you will send the text and the value of the sliders about the Pressure and the Temperature.

1.6 More programs

On the following link you can find tutorials about simple small android apps. Make sure you can create the projects on your own. Feel free to extend them and be creative.

- <http://blog.idleworx.com/2011/06/build-simple-android-app-2-button.html>
- <http://coenraets.org/blog/androidtutorial/>
- <http://www.techillumination.in/2010/02/simple-android-application-for-adding.html>