**Module 4:**

Android UI components: Text Controls, Buttons, Widgets, Layouts, Containers

Task: Explore all the UI Controls and design a Student Registration Activity

Android UI Controls

There are number of UI controls provided by Android that allow you to build the graphical user interface for your app.

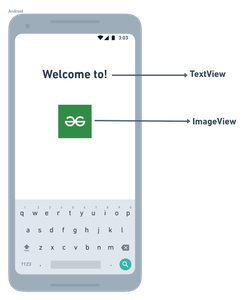
|  |  |
| --- | --- |
| **Sr.No.** | **UI Control & Description** |
| 1 | [**TextView**](https://www.tutorialspoint.com/android/android_textview_control.htm)  This control is used to display text to the user. |
| 2 | [**EditText**](https://www.tutorialspoint.com/android/android_edittext_control.htm)  EditText is a predefined subclass of TextView that includes rich editing capabilities. |
| 3 | [**AutoCompleteTextView**](https://www.tutorialspoint.com/android/android_autocompletetextview_control.htm)  The AutoCompleteTextView is a view that is similar to EditText, except that it shows a list of completion suggestions automatically while the user is typing. |
| 4 | [**Button**](https://www.tutorialspoint.com/android/android_button_control.htm)  A push-button that can be pressed, or clicked, by the user to perform an action. |
| 5 | [**ImageButton**](https://www.tutorialspoint.com/android/android_imagebutton_control.htm)  An ImageButton is an AbsoluteLayout which enables you to specify the exact location of its children. This shows a button with an image (instead of text) that can be pressed or clicked by the user. |
| 6 | [**CheckBox**](https://www.tutorialspoint.com/android/android_checkbox_control.htm)  An on/off switch that can be toggled by the user. You should use check box when presenting users with a group of selectable options that are not mutually exclusive. |
| 7 | [**ToggleButton**](https://www.tutorialspoint.com/android/android_togglebutton_control.htm)  An on/off button with a light indicator. |
| 8 | [**RadioButton**](https://www.tutorialspoint.com/android/android_radiobutton_control.htm)  The RadioButton has two states: either checked or unchecked. |
| 9 | [**RadioGroup**](https://www.tutorialspoint.com/android/android_radiogroup_control.htm)  A RadioGroup is used to group together one or more RadioButtons. |
| 10 | [**ProgressBar**](https://www.tutorialspoint.com/android/android_progressbar.htm)  The ProgressBar view provides visual feedback about some ongoing tasks, such as when you are performing a task in the background. |
| 11 | [**Spinner**](https://www.tutorialspoint.com/android/android_spinner_control.htm)  A drop-down list that allows users to select one value from a set. |
| 12 | [**TimePicker**](https://www.tutorialspoint.com/android/android_timepicker_control.htm)  The TimePicker view enables users to select a time of the day, in either 24-hour mode or AM/PM mode. |
| 13 | [**DatePicker**](https://www.tutorialspoint.com/android/android_datepicker_control.htm)  The DatePicker view enables users to select a date of the day. |

### **Android UI Layouts**

Android **Layout** is used to define the user interface that holds the UI controls or widgets that will appear on the screen of an android application or activity screen. Generally, every application is a combination of View and ViewGroup. As we know, an android application contains a large number of activities and we can say each activity is one page of the application. So, each activity contains multiple user interface components and those components are the instances of the View and ViewGroup. All the elements in a layout are built using a hierarchy of **View**and **ViewGroup**objects.

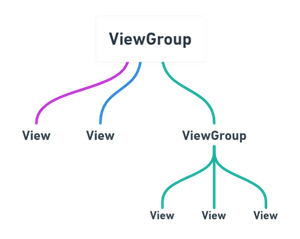
### **View**

A **View** is defined as the user interface which is used to create interactive UI components such as [TextView](https://www.geeksforgeeks.org/textview-widget-in-android-using-java-with-examples/), [ImageView](https://www.geeksforgeeks.org/imageview-in-android-with-example/), [EditText](https://www.geeksforgeeks.org/edittext-widget-in-android-using-java-with-examples/), [RadioButton](https://www.geeksforgeeks.org/radiobutton-in-kotlin/), etc., and is responsible for event handling and drawing. They are Generally Called Widgets.

[](https://whimsical.com/91txrXHfW3NywpD1qBXxhD)

*View*

A **ViewGroup** act as a base class for layouts and layouts parameters that hold other Views or ViewGroups and to define the layout properties. They are Generally Called layouts.

[](https://whimsical.com/viewgroup-8fCbKjTrG2rG76qpCiANX3)

*ViewGroup*

The Android framework will allow us to use UI elements or widgets in two ways:

* Use UI elements in the XML file
* Create elements in the Kotlin file dynamically

### **Types of Android Layout**

* **Android Linear Layout:**LinearLayout is a ViewGroup subclass, used to provide child View elements one by one either in a particular direction either horizontally or vertically based on the orientation property.
* **Android Relative Layout:**RelativeLayout is a ViewGroup subclass, used to specify the position of child View elements relative to each other like (A to the right of B) or relative to the parent (fix to the top of the parent).
* **Android Constraint Layout:**ConstraintLayout is a ViewGroup subclass, used to specify the position of layout constraints for every child View relative to other views present. A ConstraintLayout is similar to a RelativeLayout, but having more power.
* **Android Frame Layout:**FrameLayout is a ViewGroup subclass, used to specify the position of View elements it contains on the top of each other to display only a single View inside the FrameLayout.
* **Android Table Layout:**TableLayout is a ViewGroup subclass, used to display the child View elements in rows and columns.
* **Android Web View:**WebView is a browser that is used to display the web pages in our activity layout.
* **Android ListView:**ListView is a ViewGroup, used to display scrollable lists of items in a single column.
* **Android Grid View:**GridView is a ViewGroup that is used to display a scrollable list of items in a grid view of rows and columns.

# **Android Login and Registration Screen Design**

Create a new android application using android studio and give names as **LoginExample**. In case if you are not aware of creating an app in android studio check this article [Android Hello World App](https://www.tutlane.com/tutorial/android/android-hello-world-app-example).

Once we create an application, open **activity\_main.xml** file from **\res\layout** folder path and write the code like as shown below.

## activity\_main.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:orientation="vertical" android:layout\_width="match\_parent"  
    android:layout\_height="match\_parent">  
    <TextView  
        android:id="@+id/loginscrn"  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:layout\_marginTop="80dp"  
        android:text="Login"  
        android:textSize="25dp"  
        android:textStyle="bold"  
        android:layout\_gravity="center"/>  
    <TextView  
        android:id="@+id/fstTxt"  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:layout\_marginLeft="100dp"  
        android:layout\_marginTop="20dp"  
        android:text="Email"/>  
    <EditText  
        android:id="@+id/txtEmail"  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:layout\_marginLeft="100dp"  
        android:ems="10"/>  
    <TextView  
        android:id="@+id/secTxt"  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:text="Password"  
        android:layout\_marginLeft="100dp" />  
    <EditText  
        android:id="@+id/txtPwd"  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:layout\_marginLeft="100dp"  
        android:inputType="textPassword"  
        android:ems="10" />  
    <Button  
        android:id="@+id/btnLogin"  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:layout\_marginLeft="100dp"  
        android:text="Login" />  
    <TextView android:id="@+id/lnkRegister"  
        android:layout\_width="match\_parent"  
        android:layout\_height="wrap\_content"  
        android:layout\_marginTop="40dp"  
        android:text="New to Tutlane? Register here"  
        android:gravity="center"  
        android:textSize="20dp"  
        android:textColor="#3F51B5"/>  
</LinearLayout>

Now we will create another layout resource file **registration.xml** in **\res\layout** path to allow new users to register in our application for that right click on your **layout** folder à Go to **New** à select **Layout Resource File** and give name as **registration.xml**.

Once we create a new layout resource file **registration.xml**, open it and write the code like as shown below

## registration.xml

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
    android:orientation="vertical" android:layout\_width="match\_parent"  
    android:layout\_height="match\_parent">  
    <TextView  
        android:id="@+id/loginscrn"  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:layout\_marginTop="80dp"  
        android:text="Registration"  
        android:textSize="25dp"  
        android:textStyle="bold"  
        android:layout\_gravity="center"/>  
    <TextView  
        android:id="@+id/fstTxt"  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:layout\_marginLeft="100dp"  
        android:layout\_marginTop="20dp"  
        android:text="Full Name"/>  
    <EditText  
        android:id="@+id/txtName"  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:layout\_marginLeft="100dp"  
        android:ems="10"/>  
    <TextView  
        android:id="@+id/secTxt"  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:text="Email"  
        android:layout\_marginLeft="100dp" />  
    <EditText  
        android:id="@+id/txtEmail"  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:layout\_marginLeft="100dp"  
        android:ems="10" />  
    <TextView  
        android:id="@+id/thirdTxt"  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:text="Password"  
        android:layout\_marginLeft="100dp" />  
    <EditText  
        android:id="@+id/txtPwd"  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:layout\_marginLeft="100dp"  
        android:inputType="textPassword"  
        android:ems="10" />  
    <Button  
        android:id="@+id/btnLogin"  
        android:layout\_width="wrap\_content"  
        android:layout\_height="wrap\_content"  
        android:layout\_marginLeft="100dp"  
        android:text="Login" />  
    <TextView android:id="@+id/lnkLogin"  
        android:layout\_width="match\_parent"  
        android:layout\_height="wrap\_content"  
        android:layout\_marginTop="40dp"  
        android:text="Already Registered? Login here"  
        android:gravity="center"  
        android:textSize="20dp"  
        android:textColor="#3F51B5"  
        android:onClick="test"/>  
</LinearLayout>

Now open your main activity file **MainActivity.java** from **\java\com.tutlane.loginexample** path and write the code like as shown below

## MainActivity.java

package com.tutlane.loginexample;  
import android.content.Intent;  
import android.support.v7.app.AppCompatActivity;  
import android.os.Bundle;  
import android.text.method.LinkMovementMethod;  
import android.view.View;  
import android.widget.TextView;  
  
public class MainActivity extends AppCompatActivity {  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity\_main);  
        TextView register = (TextView)findViewById(R.id.lnkRegister);  
        register.setMovementMethod(LinkMovementMethod.getInstance());  
        register.setOnClickListener(new View.OnClickListener() {  
            @Override  
            public void onClick(View v) {  
                Intent intent = new Intent(MainActivity.this, RegistrationActivity.class);  
                startActivity(intent);  
            }  
        });  
    }  
}

If you observe the above code, whenever the user click on **register** link, we are redirecting the user from login screen to registration screen using “**RegistrationActivity**” for that create another [activity](https://www.tutlane.com/tutorial/android/android-activity-lifecycle) file **RegistrationActivity.java** in **\java\com.tutlane.loginexample** path.

## Output :

When we run the above example in the android emulator we will get a result like as shown below.

