LAB: 04

EXPLORE ANDROID WIDGET OR CONTROLS TOOLBOX

View

Views are the base class for all visual interface elements (commonly known as controls or widgets). All UI controls, including the layout classes, are derived from View. Android provides several common UI controls, widgets, and Layout Managers.

Android supplies a toolbox of standard Views to help you create your UIs. By using these controls, you can simplify your development and provide consistency between applications.

- **TextView** A standard read-only text label that supports multiline display, string formatting, and automatic word wrapping.
- **EditText** An editable text entry box that accepts multiline entry, word-wrapping, and hint text.
- **ListView** A View Group that creates and manages a vertical list of Views, displaying them as rows within the list. The simplest List View displays the toString value of each object in an array, using a Text View for each item.
- **Button** A standard push button.
- CheckBox A two-state button represented by a checked or unchecked box
- **RadioButton** A two-state grouped button. A group of these presents the user with a number of possible options, of which only one can be enabled at a time.
- **ToggleButton** A two-state button that can be used as an alternative to a check box. It's particularly appropriate where pressing the button will initiate an action as well as changing a state (such as when turning something on or off)
- **VideoView** Handles all state management and display Surface configuration for playing videos more simply from within your Activity.

Android also supports several more advanced View implementations, including date-time pickers, auto-complete input boxes, maps, galleries, and tab sheets. For a more comprehensive list of the available widgets, head to

Attributes:

Let's go over some of the attributes that you have used to configure your widgets.

• android:layout_width / android:layout_height match_parent view will be as big as its parent. wrap_content view will be as big as its contents require.

android:orientation

Vertical – column representation. Horizontal – row representation.

· android:text · android:id

each control contains unique id with prefix "@+id"

android:padding/android:margin

Margin attributes are layout parameters. They determine the distance between widget Padding, on the other hand, is not a layout parameter.

The **android:padding** attribute tells the widget how much bigger than its contents it should draw itself.

Let's try:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout width="match parent"
android:layout_height="match_parent"
                                        tools:context=".MainActivity">
  <TextView
    android:id="@+id/txtsubject"
android:layout width="wrap content"
android:layout height="wrap content"
android:layout marginTop="60dp"
                                      android:text="Date
of Birth"
    app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/txtcity"/>
  <TextView
    android:id="@+id/txtname"
android:layout width="wrap content"
android:layout height="wrap content"
android:layout marginStart="8dp"
                                      android:text="Name"
    app:layout_constraintBaseline_toBaselineOf="@+id/edtName"
app:layout_constraintStart_toStartOf="parent"/>
```

```
<TextView
    android:id="@+id/txtemail"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="4dp"
                                      android:text="Email"
    app:layout constraintBaseline toBaselineOf="@+id/edtEmail"
app:layout_constraintStart_toStartOf="parent" />
  <TextView
    android:id="@+id/txtgender"
android:layout_width="wrap_content"
android:layout height="wrap content"
android:layout_marginStart="8dp"
                                      android:text="Gender"
android:layout_marginTop="52dp"
    app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/edtEmail"/>
  <TextView
    android:id="@+id/txtcity"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginStart="16dp"
android:layout marginTop="104dp"
    android:text="City"
    app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/txtgender" />
  <EditText
android:id="@+id/edtName"
android:layout width="wrap content"
android:layout_height="wrap_content"
android:layout marginStart="23dp"
android:layout marginTop="57dp"
android:ems="10"
    android:inputType="textPersonName"
app:layout_constraintStart_toEndOf="@+id/txtname"
    app:layout constraintTop toTopOf="parent" />
  <EditText
    android:id="@+id/edtEmail"
android:layout width="wrap content"
```

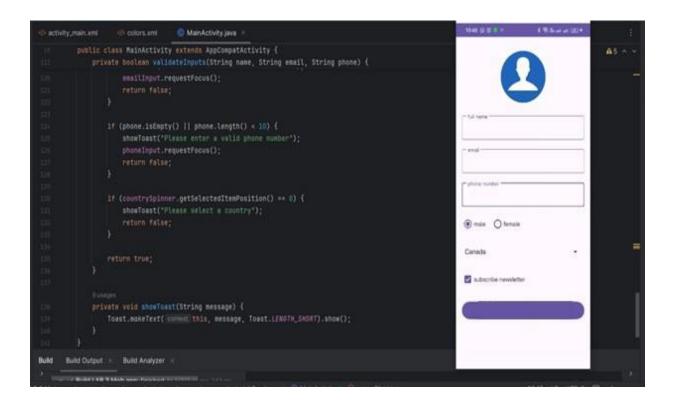
```
android:layout_height="wrap_content"
android:layout marginTop="15dp"
                                      android:ems="10"
    android:inputType="textEmailAddress"
app:layout constraintStart toStartOf="@+id/edtName"
    app:layout constraintTop toBottomOf="@+id/edtName" />
  < Radio Group
    android:id="@+id/radioGroup"
android:layout_width="163dp"
android:layout_height="wrap_content"
android:layout_marginTop="88dp"
app:lavout constraintStart_toStartOf="@+id/edtEmail"
    app:layout_constraintTop_toTopOf="@+id/edtEmail">
    < Radio Button
android:id="@+id/radioButton2"
android:layout width="wrap content"
android:layout height="wrap content"
android:text="Male"
tools:layout editor absoluteX="83dp"
      tools:layout_editor_absoluteY="205dp"/>
    < Radio Button
android:id="@+id/radioButton3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Female"
tools:layout_editor_absoluteX="166dp"
tools:layout_editor_absoluteY="205dp"/>
  </RadioGroup>
  <Spinner
               android:id="@+id/cityspinner"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout marginStart="36dp"
android:layout_marginTop="60dp"
android:entries="@array/city"
app:layout_constraintStart_toEndOf="@+id/txtcity"
app:layout constraintTop toBottomOf="@+id/radioGroup"
    tools:ignore="MissingConstraints" />
```

```
<DatePicker
    android:id="@+id/dtpicker"
android:layout_width="wrap_content"
android:layout height="210dp"
android:layout_marginStart="104dp"
android:layout_marginTop="328dp"
android:calendarViewShown="false"
android:datePickerMode="spinner"
android:minDate="01/01/2000"
                                   android:spinnersShown="true"
app:layout constraintStart toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />
  <Button
    android:id="@+id/button"
android:layout_width="match_parent"
android:layout_height="47dp" android:layout_marginStart="16dp"
android:onClick="submit" android:text="Submit"
app:layout constraintBottom toBottomOf="parent"
app:layout_constraintStart_toStartOf="parent" />
</android.support.constraint.ConstraintLayout>
//MainActivity.Java
package com.example.controls project;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle; import
android.view.View; import
android.widget.CheckBox; import
android.widget.DatePicker; import
android.widget.EditText; import
android.widget.RadioButton; import
android.widget.RadioGroup; import
android.widget.Spinner;
import android.widget.Toast;
import java.util.Date;
public class MainActivity extends AppCompatActivity {
  EditText name, email;
```

```
RadioGroup gender;
 Spinner city;
 DatePicker dob;
  @Override
  protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    name=(EditText) findViewById(R.id.edtName);
email=(EditText) findViewById(R.id.edtEmail);
gender=(RadioGroup) findViewById(R.id.radioGroup);
city=(Spinner) findViewById(R.id.cityspinner);
dob=(DatePicker)findViewById(R.id.dtpicker);
  public void submit(View v)
String username=name.getText().toString();
String useremail=email.getText().toString();
int g=gender.getCheckedRadioButtonId();
RadioButton rd=(RadioButton) findViewById(g);
String usercity=city.getSelectedItem().toString();
String dt=dob.getDayOfMonth()+"-"+(dob.getMonth()+1)+"-"+dob.getYear();
    Toast.makeText(this,"Name: "+username+" "+" Email: "+useremail+" City:
"+usercity+" Gender: "+rd.getText()+" Date_of_Birth:
"+dt,Toast.LENGTH_LONG).show();
  }
}
```

Lab Task

1. Build interactive user details page and display information in Toast.



2. Build customer feedback activity, necessary control should be added in activity.

