Lab 6
Working with UI Controls - 2

1. ProgressBar

In android, **ProgressBar** is a user interface control that is used to indicate the progress of an operation. For example, downloading a file, uploading a file.

Following is the pictorial representation of using a different type of progress bars in android applications.

By default the ProgressBar will be displayed as a spinning wheel, in case if we want to show it like a horizontal bar then we need to change the style property to horizontal like style="?android:attr/progressBarStyleHorizontal".

Android ProgressBar Example

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent" android:layout_height="match_parent">
   <ProgressBar
      android:id="@+id/pBar"
      style="?android:attr/progressBarStyleHorizontal"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginLeft="100dp"
      android:layout_marginTop="200dp"
      android:minHeight="50dp"
      android:minWidth="200dp"
      android:max="100"
      android:indeterminate="false"
```

```
        android:progress="0" />
    <TextView
        android:id="@+id/tView"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/pBar"
        android:layout_below="@+id/pBar" />
    <Button
        android:id="@+id/btnShow"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="130dp"
        android:layout_marginTop="20dp"
        android:text="Start Progress"
        android:layout_below="@+id/tView"/>
</RelativeLayout>
```

MainActivity.java

```
package com.tutlane.progressbarexample;
import android.os.Handler;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.ProgressBar;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    private ProgressBar pgsBar;
    private int i = 0;
    private TextView txtView;

    //Creating a Handler

   *A Handler allows you to send and process Message and Runnable objects associated with a thread's MessageQueue */

    private Handler hdlr = new Handler();


   @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        pgsBar = (ProgressBar) findViewById(R.id.pBar);
        txtView = (TextView) findViewById(R.id.tView);
        Button btn = (Button)findViewById(R.id.btnShow);
```

```
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                i = pgsBar.getProgress();
                new Thread(new Runnable() {
                    public void run() {
                        while (i < 100) {
                            i += 1;
                            // Update the progress bar and display the current value in text view
                            hdlr.post(new Runnable() {
                                public void run() {
                                    pgsBar.setProgress(i);
                                    txtView.setText(i+"/"+pgsBar.getMax());
                                }
                            });
                            try {
                                // Sleep for 100 milliseconds to show the progress slowly.
                                Thread.sleep(100);
                            } catch (InterruptedException e) {
                                e.printStackTrace();
                            }
                        }
                    }
                }).start();
            }
        });
    }
}
```

2. Spinner

In android, **Spinner** is a view that allows a user to select one value from the list of values. The spinner in android will behave same as a dropdown list in other programming languages.

Generally, the android spinners will provide a quick way to select one item from the list of values and it will show a dropdown menu with a list of all values when we click or tap on it.

By default, the android spinner will show its currently selected value and by using **Adapter** we can bind the items to spinner objects.

Following is the pictorial representation of using **spinner** in android applications.

We can populate our **Spinner** control with list of choices by defining an **ArrayAdapter** in our Activity file.

Generally, the **Adapter** pulls data from sources such as an array or database and converts each item into a result view and that's placed into the list.

Android Spinner Example

activity_main.xml

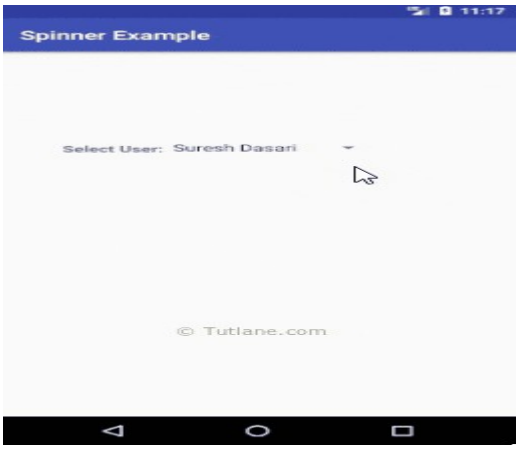
```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent" android:layout_height="match_parent">
   <TextView
      android:id="@+id/txtVw"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginLeft="50dp"
      android:layout_marginTop="150dp"
      android:text="Select User:"
      android:textStyle="bold"
      android:textSize="15dp" />
```

```
    <Spinner
        android:id="@+id/spinner1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignBottom="@+id/txtVw"
        android:layout_toRightOf="@+id/txtVw" />
</RelativeLayout>
```

MainActivity.java

```
package com.tutlane.spinnerexample;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Spinner;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity implements
AdapterView.OnItemSelectedListener {
String[] users = { "Suresh Dasari", "Trishika Dasari", "Rohini Alavala", "Praveen Kumar", "Madhav
Sai" };
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Spinner spin = (Spinner) findViewById(R.id.spinner1);
        ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,
android.R.layout.simple_spinner_item, users);
        adapter.setDropDownViewResource(android.R.layout.simple_spinner_dropdown_item);
        spin.setAdapter(adapter);
        spin.setOnItemSelectedListener(this);
    }
    @Override
    public void onItemSelected(AdapterView<?> arg0, View arg1, int position,long id) {
        Toast.makeText(getApplicationContext(), "Selected User:
"+users[position] ,Toast.LENGTH_SHORT).show();
    }
    @Override
    public void onNothingSelected(AdapterView<?> arg0) {
        // TODO - Custom Code
    }
}
```

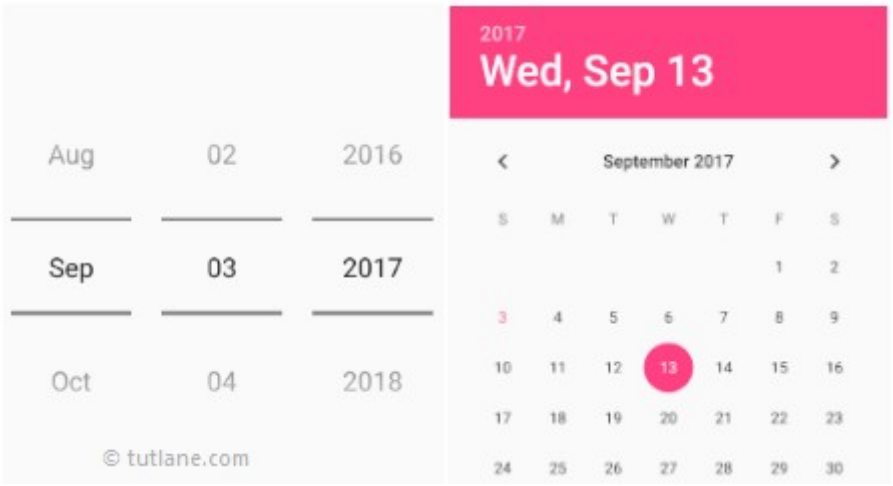


3. DatePicker

In android, **DatePicker** is a control that will allow users to select the date by a day, month and year in our application user interface.

If we use **DatePicker** in our application, it will ensure that the users will select a valid date.

Following is the pictorial representation of using a datepicker control in android applications.



Generally, in android DatePicker available in two modes, one is to show the complete calendar and another one is to show the dates in spinner view.

Android DatePicker Example

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent" android:layout_height="match_parent">
   <DatePicker
```

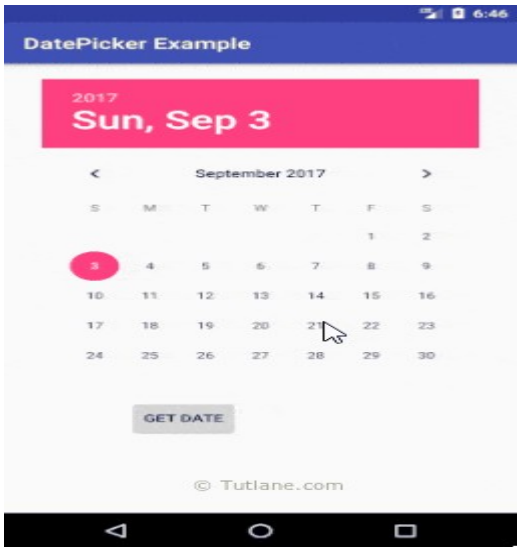
```
        android:id="@+id/datePicker1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerHorizontal="true"
        android:layout_marginTop="20dp" />
    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/datePicker1"
        android:layout_marginLeft="100dp"
        android:text="Get Date" />
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/button1"
        android:layout_marginLeft="100dp"
        android:layout_marginTop="10dp"
        android:textStyle="bold"
        android:textSize="18dp"/>
</RelativeLayout>
```

MainActivity.java

```
package com.tutlane.datepickerexample;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.DatePicker;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    DatePicker picker;
    Button btnGet;
    TextView tvw;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        tvw=(TextView)findViewById(R.id.textView1);
        picker=(DatePicker)findViewById(R.id.datePicker1);
        btnGet=(Button)findViewById(R.id.button1);
        btnGet.setOnClickListener(new View.OnClickListener() {
            @Override
```

```
        public void onClick(View v) {
            tvw.setText("Selected Date: "+ picker.getDayOfMonth()+"/"+ (picker.getMonth() +
1)+"/"+picker.getYear());
        }
    });
  }
}
```



4. TimePicker

In android, **TimePicker** is a widget for selecting the time of day, in either 24-hour or AM/PM mode.

If we use **TimePicker** in our application, it will ensure that the users will select a valid time for the day.

Following is the pictorial representation of using a timepicker control in android applications.

Generally, in android TimePicker available in two modes, one is to show the time in clock mode and another one is to show the time in spinner mode.

Android TimePicker Example

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent" android:layout_height="match_parent">
   <TimePicker
      android:id="@+id/timePicker1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_centerHorizontal="true"
      android:layout_marginTop="20dp" />
   <Button
      android:id="@+id/button1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_below="@+id/timePicker1"
      android:layout_marginTop="10dp"
      android:layout_marginLeft="160dp"
      android:text="Get Date" />
   <TextView
      android:id="@+id/textView1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_below="@+id/button1"
      android:layout_marginLeft="120dp"
      android:layout_marginTop="10dp"
      android:textStyle="bold"
      android:textSize="18dp"/>
</RelativeLayout>
```

MainActivity.java

```
package com.tutlane.timepickerexample;
import android.os.Build;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
import android.widget.TimePicker;

public class MainActivity extends AppCompatActivity {
   TimePicker picker;
   Button btnGet;
   TextView tvw;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      tvw=(TextView)findViewById(R.id.textView1);
      picker=(TimePicker)findViewById(R.id.timePicker1);
      picker.setIs24HourView(true);
      btnGet=(Button)findViewById(R.id.button1);
      btnGet.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View v) {
            int hour, minute;
            String am_pm;
            if (Build.VERSION.SDK_INT >= 23 ){
               hour = picker.getHour();
               minute = picker.getMinute();
            }
            else{
               hour = picker.getCurrentHour();
               minute = picker.getCurrentMinute();
            }
            if(hour > 12) {
               am_pm = "PM";
               hour = hour - 12;
            }
            else
            {
               am_pm="AM";
            }
            tvw.setText("Selected Date: "+ hour +":"+ minute+" "+am_pm);
         }
```

```
        });
    }
}
```

5. RatingBar

In android, **RatingBar** is a UI control that is used to get the rating from the user. The **RatingBar** shows a rating in stars and it allows users to set the rating value by touch or click on the stars.

The android **RatingBar** will always return a rating value as a floating-point number such as 1.0, 2.0, 2.5, 3.0, 3.5, etc.

Following is the pictorial representation of using a RatingBar in android applications.

In android, by using **android:numStars** attribute we can define the number of stars to display in **RatingBar**. An example of using RatingBar is in movie sites or product sites to collect the user rating about the movies or products, etc.

Get Android RatingBar Value

In android, by using **RatingBar** methods (**getNumStars()**, **getRating()**) we can get the number of stars and the rating value which was selected.

Following is the code snippet to get the rating details from RatingBar in android applications.

```
int noofstars = rBar.getNumStars();
float getrating = rBar.getRating();
tView.setText("Rating: "+getrating+"/"+noofstars);
```

This is how we can get the number of stars in RatingBar control and the selected rating value from RatingBar control in android applications.

Android RatingBar Control Example

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent" android:layout_height="match_parent">
   <RatingBar
      android:id="@+id/ratingBar1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginLeft="80dp"
      android:layout_marginTop="200dp"
      android:numStars="5"
      android:rating="3.5"/>
   <Button
      android:id="@+id/btnGet"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignLeft="@+id/ratingBar1"
      android:layout_below="@+id/ratingBar1"
      android:layout_marginTop="30dp"
      android:layout_marginLeft="60dp"
      android:text="Get Rating"/>
   <TextView
      android:id="@+id/textview1"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_alignLeft="@+id/btnGet"
      android:layout_below="@+id/btnGet"
      android:layout_marginTop="20dp"
      android:textSize="20dp"
      android:textStyle="bold"/>
</RelativeLayout>
```

MainActivity.java

```
package com.tutlane.ratingbarexample;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import android.widget.RatingBar;
import android.widget.TextView;

public class MainActivity extends AppCompatActivity {
    private RatingBar rBar;
    private TextView tView;
    private Button btn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        rBar = (RatingBar) findViewById(R.id.ratingBar1);
        tView = (TextView) findViewById(R.id.textview1);
        btn = (Button)findViewById(R.id.btnGet);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                int noofstars = rBar.getNumStars();
                float getrating = rBar.getRating();
                tView.setText("Rating: "+getrating+"/"+noofstars);
            }
        });
    }
}
```