**Android Basics**

**Introduction**:

For getting a fair idea about the android studio check this link: <https://developer.android.com/studio/intro/index.html>.

**Views and Layouts (View Groups):**

A Good Mobile Application successfully interacts with its user. Hence, the quality of mobile application depends on the responsiveness of the User Interface. User Interface consists of Layouts and Layouts consists of Views. In this section, we are going to learn more about views in Android Studio. View objects are basically used for drawing contents on the UI of android. Views are created in Android Studio with the help of XML. The UI of android is coded in “activity\_main.xml“ file.

There are different types of Views but main types are TextView, ImageView, Button and Edit View.

**TextView:**

Anything that consists of words, sentences or in short text is called as TextView

Let’s see an example of TextView.

<**TextView  
 android:id="@+id/sample\_text"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Hello World!"** />

Note: In XML every tag is closed ( /> ) and the syntax is case sensitive hence remember not to forget the CamelCase i.e. the TextView cannot be written as Textview or Text View in the above syntax.

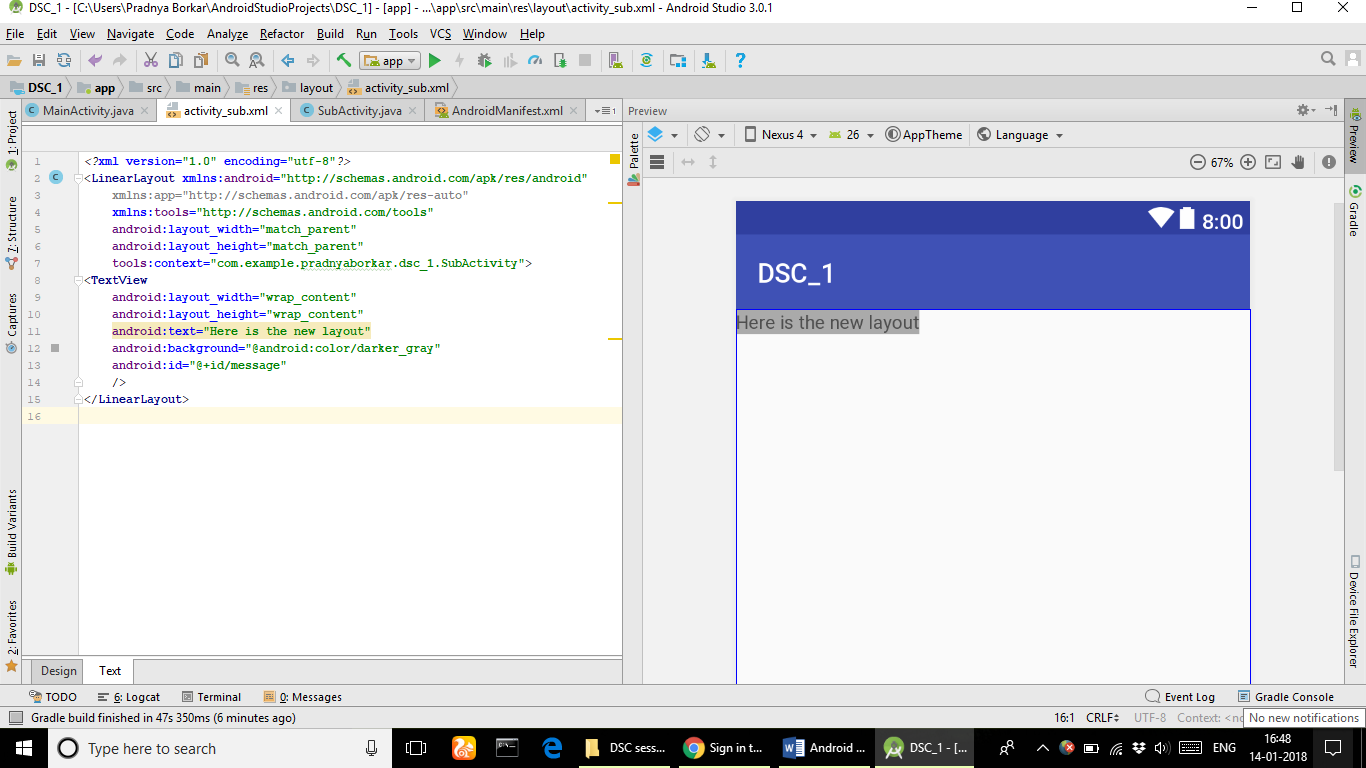
List of attributes of TextView:

1. **android:id=”@+id/id\_name”**

For a Layout with one TextView, it is easy to refer the view. But imagine a Layout with many TextView. For referring a TextView we need to have some kind of unique name. This is basically done using the id attribute. It helps to refer to a particular TextView.

1. **Layout\_width and Layout\_height**: As the name suggests, it defines the width and height of the layout respectively. This attributes can have two values “ wrap\_content “ and “ match\_parent”

In wrap\_content the Text Box wraps the complete content and expands or contracts with accordance with increase or decrease of the content respectively as shown in the below image:

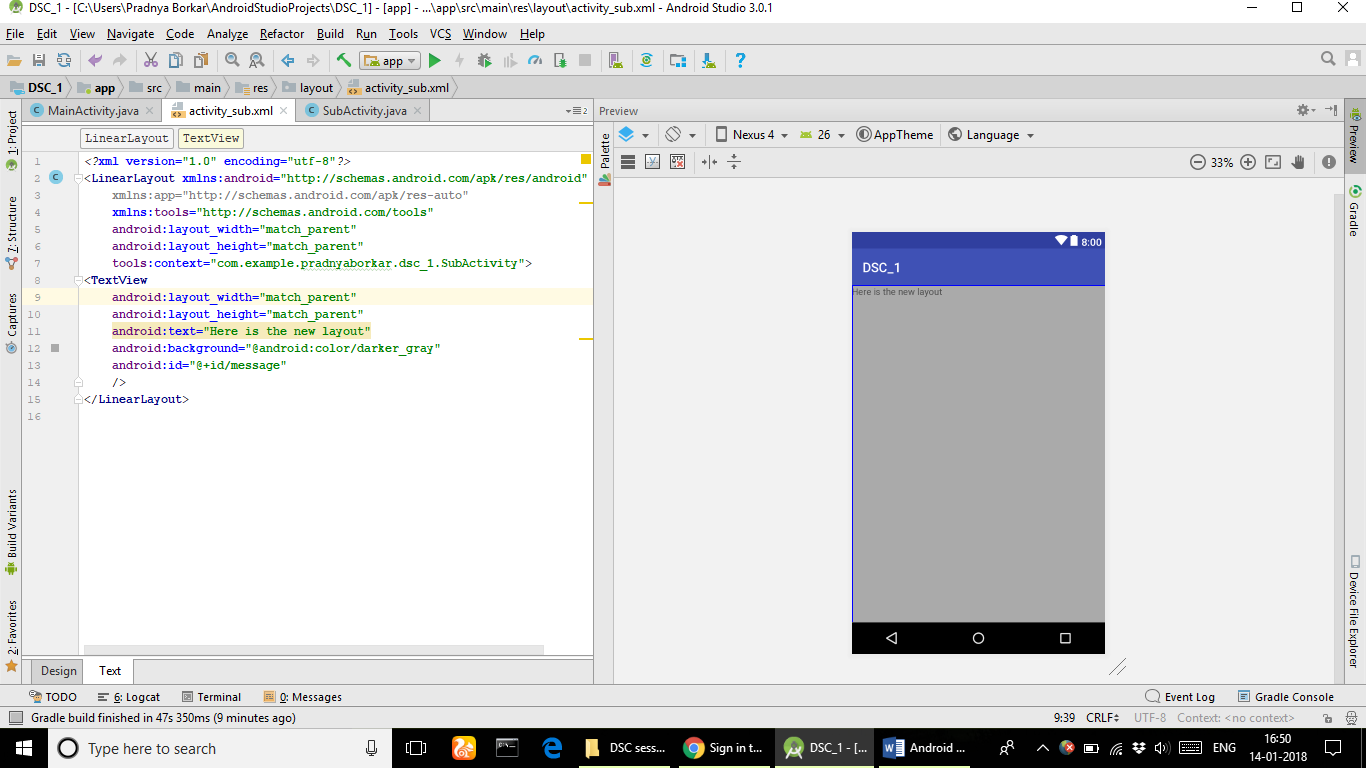


In match\_parent the Text Box matches the width/height of its parent as shown in the below image:

Note: Here Parent refers to the view Groups and views are referred as children

1. **Android:text**

It is used to define the value of the text.



**ImageView:**

It is used to display the Images on the screen of the device.

The sorce images for ImageView should be stored in the drawable folder.

**LinearLayout:**

When we arrange different views in a horizontal or vertical orientation it is called a LinearLayout.

Let’s try the following code:

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:tools="http://schemas.android.com/tools"

android:id="@+id/activity\_main"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

android:orientation="vertical"

tools:context="com.example.pradnyaborkar.test123.MainActivity">

<ImageView

android:layout\_width="match\_parent"

android:layout\_height="50dp"

android:src="@drawable/shark"

android:id="@+id/image"

android:scaleType="fitXY"

android:layout\_weight="1"

/>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="50dp"

android:text="Hello World!"

android:id="@+id/text1"

android:layout\_weight="1"

/>

<TextView

android:layout\_width="match\_parent"

android:layout\_height="50dp"

android:text="Good Morning!"

android:id="@+id/text"

android:layout\_weight="1"

/>

<Button

android:layout\_width="match\_parent"

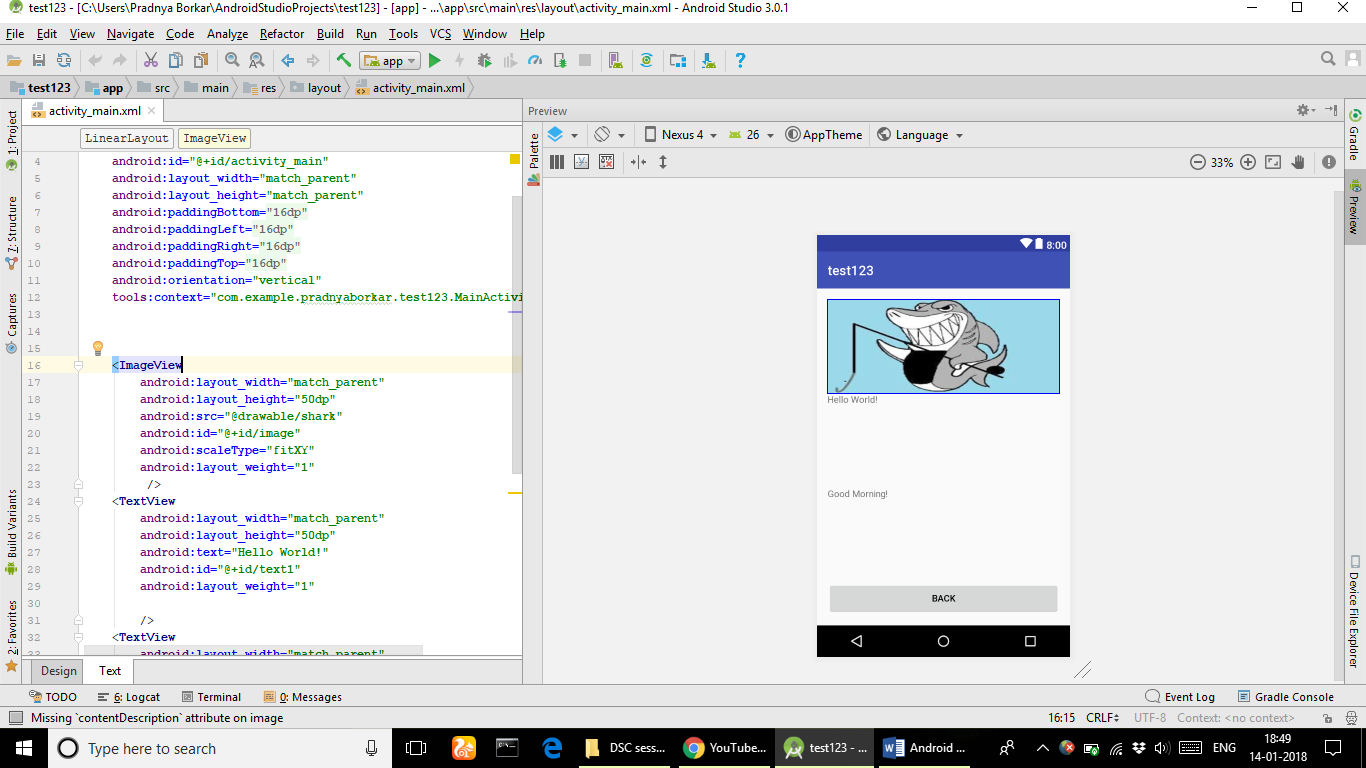
android:layout\_height="50dp"

android:text="@string/back"

/>

</LinearLayout>

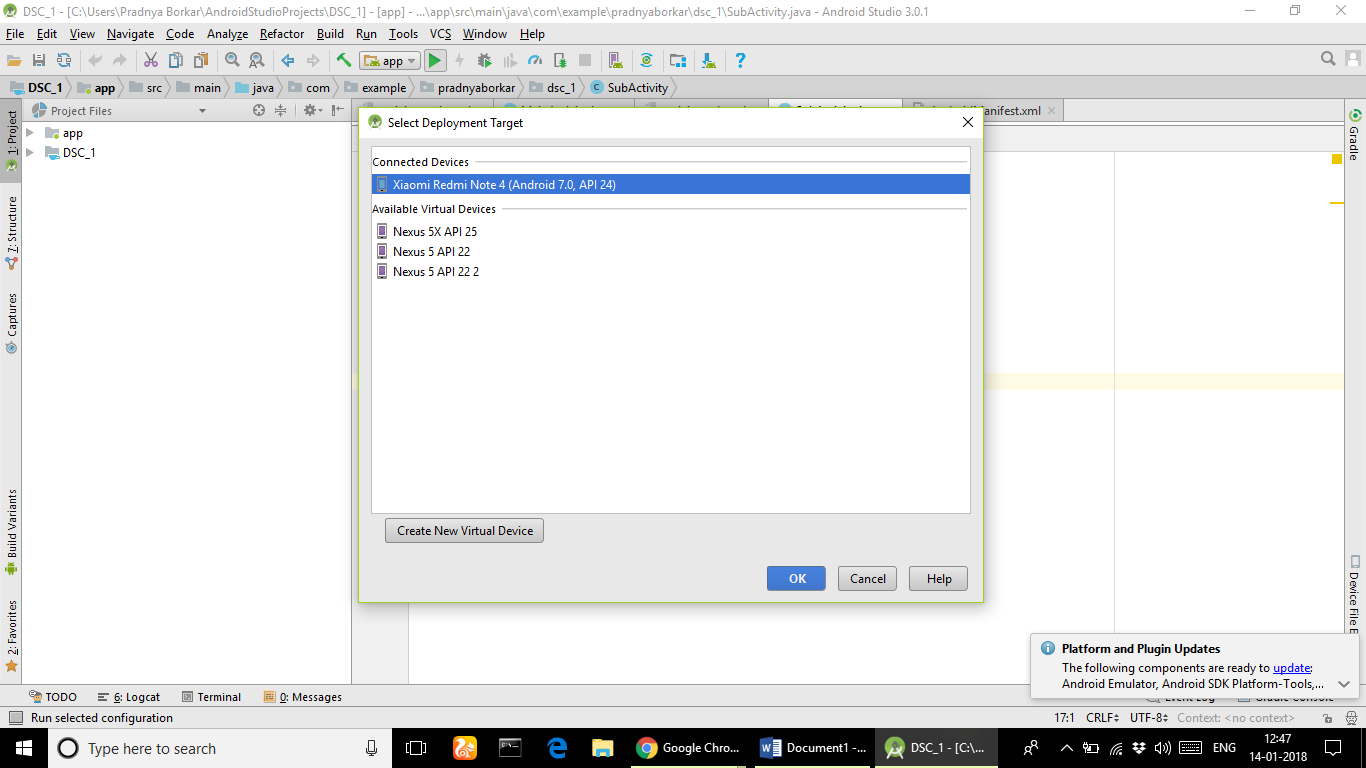
The Output looks like this:



**Building an Application on your device:**

For building an application, you need to use your mobile in developer mode that allows you to build and run your application on your device.

**Step 1**: Press Run

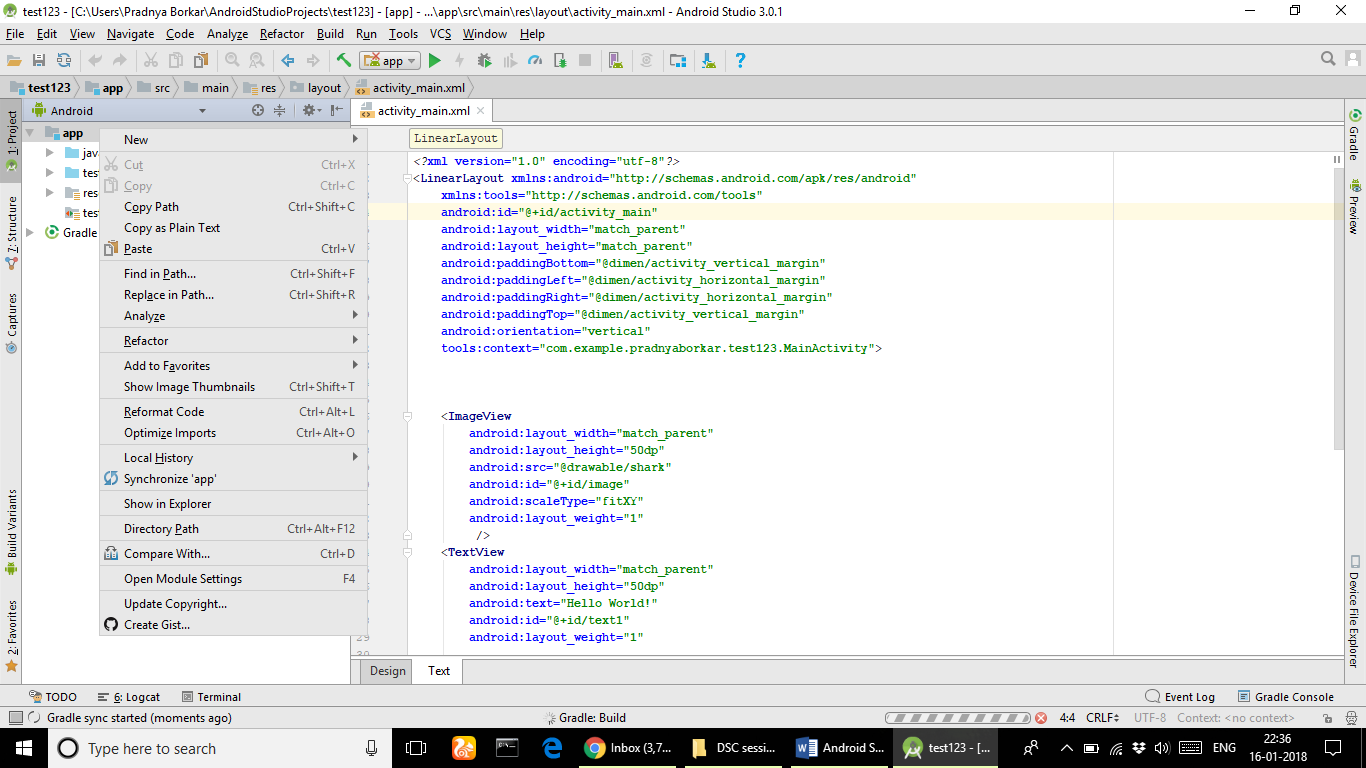


**Step 2**: Select your connected device and press ok.

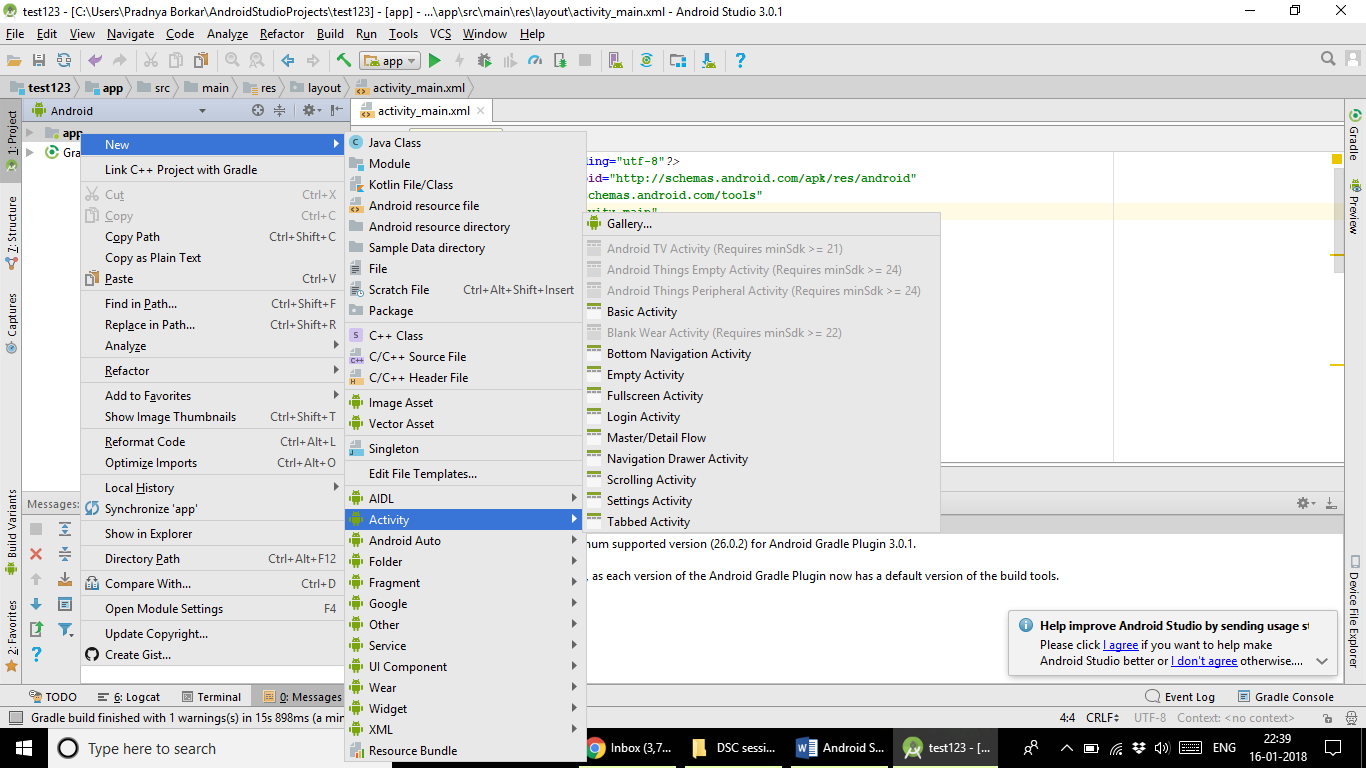
The application would be build in your mobile and would start running.

**Creating New Activities in Android:**

**Step1:** Right click on app folder as shown in figure

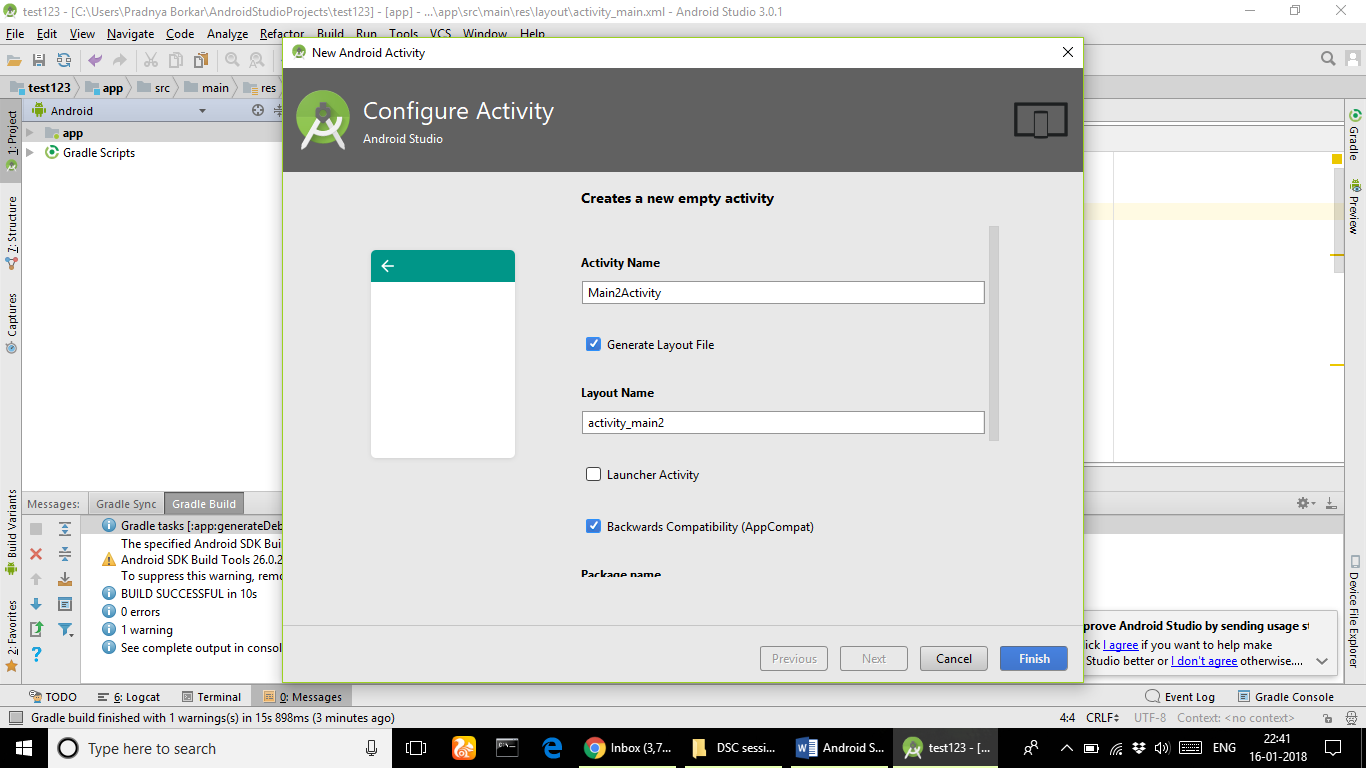


**Step 2:** Click New and go to Activity and select Empty Activity as shown



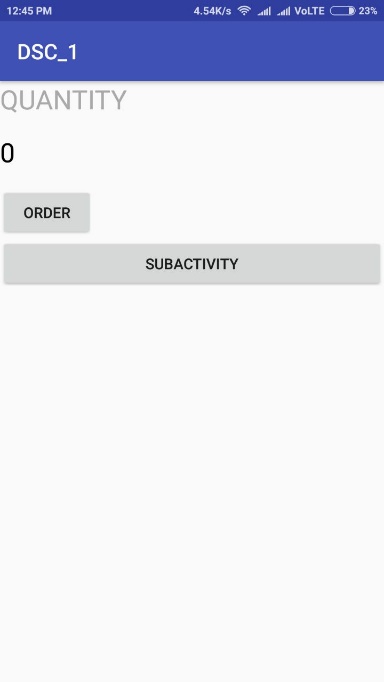
**Step 3:** Name the Activity and press Finish

The New Activity is Created.



**Making a Simple App:**

**Screen Shots of the application to be built:**

**  **

**Figure 1 Figure 2 Figure 3**

**Description of the App:**

1) This Application consists of two activities namely the Main\_Activity and Sub\_Activity.

2) The Main\_Activity.xml consists of two text views and two Button views as shown in figure 1.

3) When the ORDER button is pressed the value displayed in the TextView with id count increases by 1.

4) When the SUBACTIVITY button is pressed we move to the next screen of the application which consists of a single TextView as shown in figure 3.

**Source code:**

**Main\_Activity.java**

package com.example.pradnyaborkar.dsc\_1;

import android.content.Intent;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

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);

}

public void submitOrder(View view){

display(1);

}

private void display(int number) {

TextView quantityTextView = (TextView) findViewById(R.id.count);

String i=quantityTextView.getText().toString();

int a=Integer.parseInt(i);

quantityTextView.setText(""+(a+number));

}

public void openSubActivity(View view){

Intent intent= new Intent(this,SubActivity.class);

startActivity(intent);

}

}

**Main\_Activity.xml**

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout 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="com.example.pradnyaborkar.dsc\_1.MainActivity"

android:orientation="vertical"

>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/quantity"

android:textColor="@android:color/darker\_gray"

android:textSize="25sp"

android:paddingBottom="16dp"

android:textAllCaps="true"

/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/count"

android:textColor="@android:color/black"

android:id="@+id/count"

android:textSize="25sp"

android:paddingBottom="16dp"

/>

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Order"

android:onClick="submitOrder"

android:id="@+id/order"

/>

<Button

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="SubActivity"

android:onClick="openSubActivity"

android:id="@+id/next"

/>

</LinearLayout>

**Sub\_Activity.java**

package com.example.pradnyaborkar.dsc\_1;

import android.content.Intent;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

public class SubActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_sub);

}

}

Sub\_Activity.xml

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout 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="com.example.pradnyaborkar.dsc\_1.SubActivity">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="1"

android:text="Here is the new layout"

android:id="@+id/message"

/>

</LinearLayout>