Experiment No. : 4	CUSTOM VIEW	Date: 21/4/22

AIM: To design a Custom View in Android Application.

Theory:

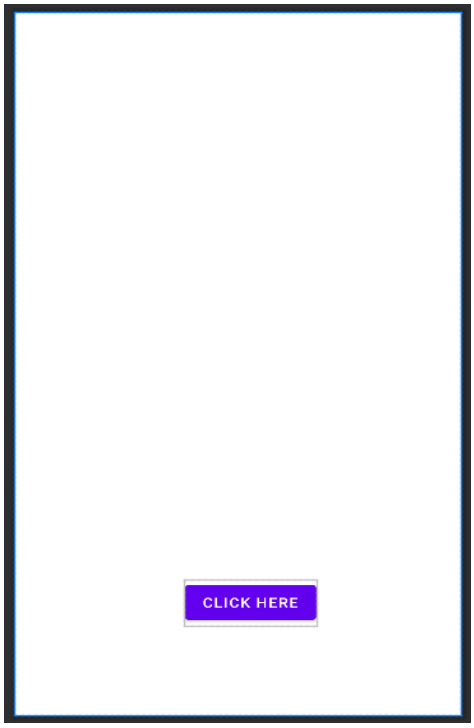
What is a custom View?
Sometimes you want to show a certain type of data and there is already a suitable view in the basic widget set. But if you want UI customization or a different user interaction, you may need to extend a widget.
Suppose that there were no Button widget in the basic widget set in the Android SDK and you want to make one. You would extend the TextView class to get all the capabilities related to the text like setting text, text color, text size, text style and so on. Then you will start your customization work, to give your new widget the look and feel of a button. this is what happens in the Android SDK the Button class extends the TextView class.
Or you could in theory extend the View class to start from scratch.

When an Android activity comes up into the foreground, Android asks it for its root view. The root view is the top parent of the layout hierarchy. Android then starts drawing the whole view hierarchy. Android draws the hierarchy starting from the top parent, then its children, and if one of the children is also a ViewGroup, Android will draw its children before drawing the second child. So it's a depth-first traversal.
Android draws the children of a ViewGroup according to the index of the child (its position in the XML file), so the view which you added first will be drawn first.
Android draws the layout hierarchy in three stages:
1. Measuring stage: each view must measure itself.
2. Layout stage: each ViewGroup finds the right position for its children on the screen by using the child size and also by following the layout rules.
3. Drawing stage: after measuring and positioning all of the views, each view happily draws itself.

Main_Activity :

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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">

   <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      app:layout_constraintBottom_toBottomOf="parent"
      app:layout_constraintHorizontal_bias="0.538"
      app:layout_constraintLeft_toLeftOf="parent"
      app:layout_constraintRight_toRightOf="parent"
      app:layout_constraintTop_toTopOf="parent"
      android:text="Click Here"
      android:id="@+id/button"
      app:layout_constraintVertical_bias="0.863" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Layout Design:



Custom View Layout:

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@color/teal_700"
    android:id="@+id/tost_root">

    <ImageView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:src="@drawable/ic_chat"
        android:layout_marginEnd="8dp"
        android:id="@+id/toast_img"
        />

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="This is a Custom Toast Message"
        android:textColor="@color/white"
        android:textSize="18sp"
        android:id="@+id/toast_txt"/>

</LinearLayout>
```

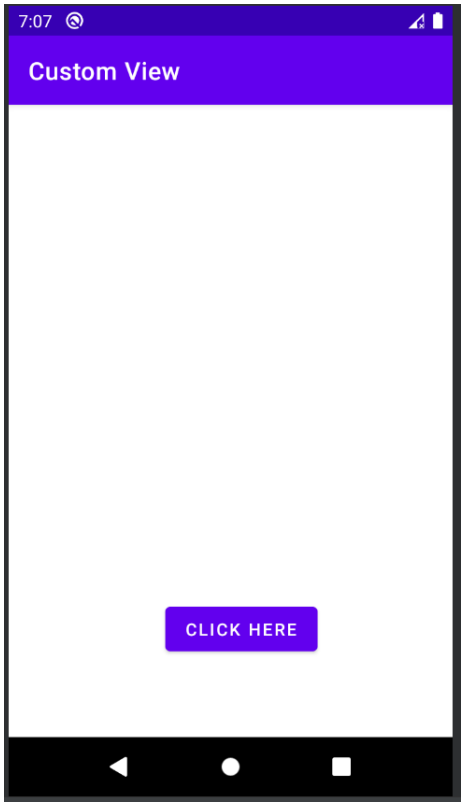
CustomView Layout Design:

MainActivity Java Code:

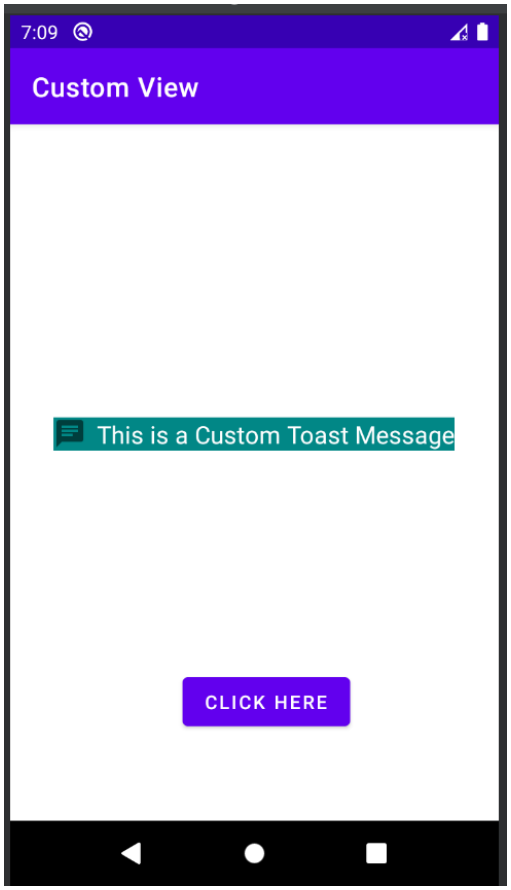
```
package com.ramanand.customview;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.Gravity;
import android.view.LayoutInflater;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
   private Button btn;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);

      btn = (Button) findViewById(R.id.button);
      btn.setOnClickListener(new View.OnClickListener() {
         @Override
         public void onClick(View view) {
```

```
            showToast();
        }
    });
}

public void showToast(){
    LayoutInflater inflater = getLayoutInflater();
    View layout = inflater.inflate((R.layout.toast_layout), findViewById(R.id.tost_root));
    Toast toast = new Toast(getApplicationContext());
    toast.setGravity(Gravity.CENTER, 0, 0);
    toast.setDuration(Toast.LENGTH_LONG);
    toast.setView(layout);

    toast.show();
}
}
```

Initial Screenshot:



Screenshot After clicking CLICK ME Button:



Conclusion: Custom Views in Android was successfully studied and implemented in an Android Application.

Agnel Institute of Technology & Design Assagao, Goa. Department of ECE		
Programming	(2)	
Execution & Result	(3)	
Neatness	(2)	
Viva	(3)	
Total	(10)	
Faculty In-charge sign		