

Practical NO :- 01

Date:- 05/03/2025

Aim:- Introduction to android, introduction to android studio IDE, application fundamentals; creating a project, simple "Hello World" Program.

Introduction to Android

- 1) Android is an open source and Linux-based Operating System for mobile devices such as smartphones and tablet computers.
- 2) Android was developed by the Open Handset Alliance, led by Google, and other companies.
- 3) Android offers a unified approach to application development for mobile devices which means developers need only develop for Android, and their applications should be able to run on different devices powered by Android.
- 4) Android operating System
 - a) The main advantage of adopting Android is that it offers a unified approach to application development
- 5) Developers need only develop for Android, and their applications should be able to run on numerous different devices as long as the device devices are powered using Android.

Introduction to Android IDE

Android Studio :- Android studio is the official integrated development environment (IDE) for Android application development.

It is based on IntelliJ IDEA, a Java integrated development environment for software, and incorporates its code editing and developer tools.

To support application development within the Android operating system, Android studio uses a Gradle-based build system, ~~Android Emulator~~, code templates and GitHub integration.

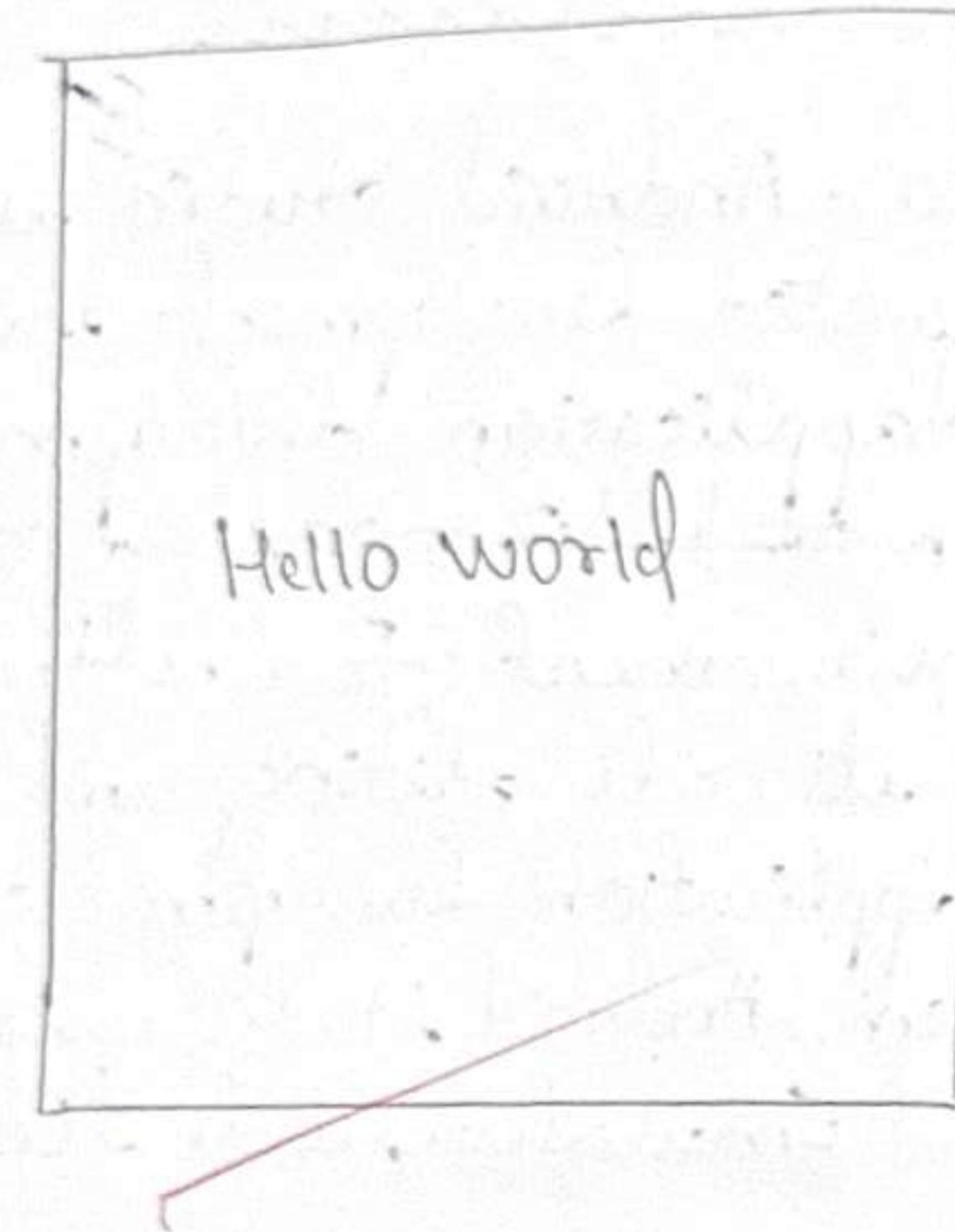
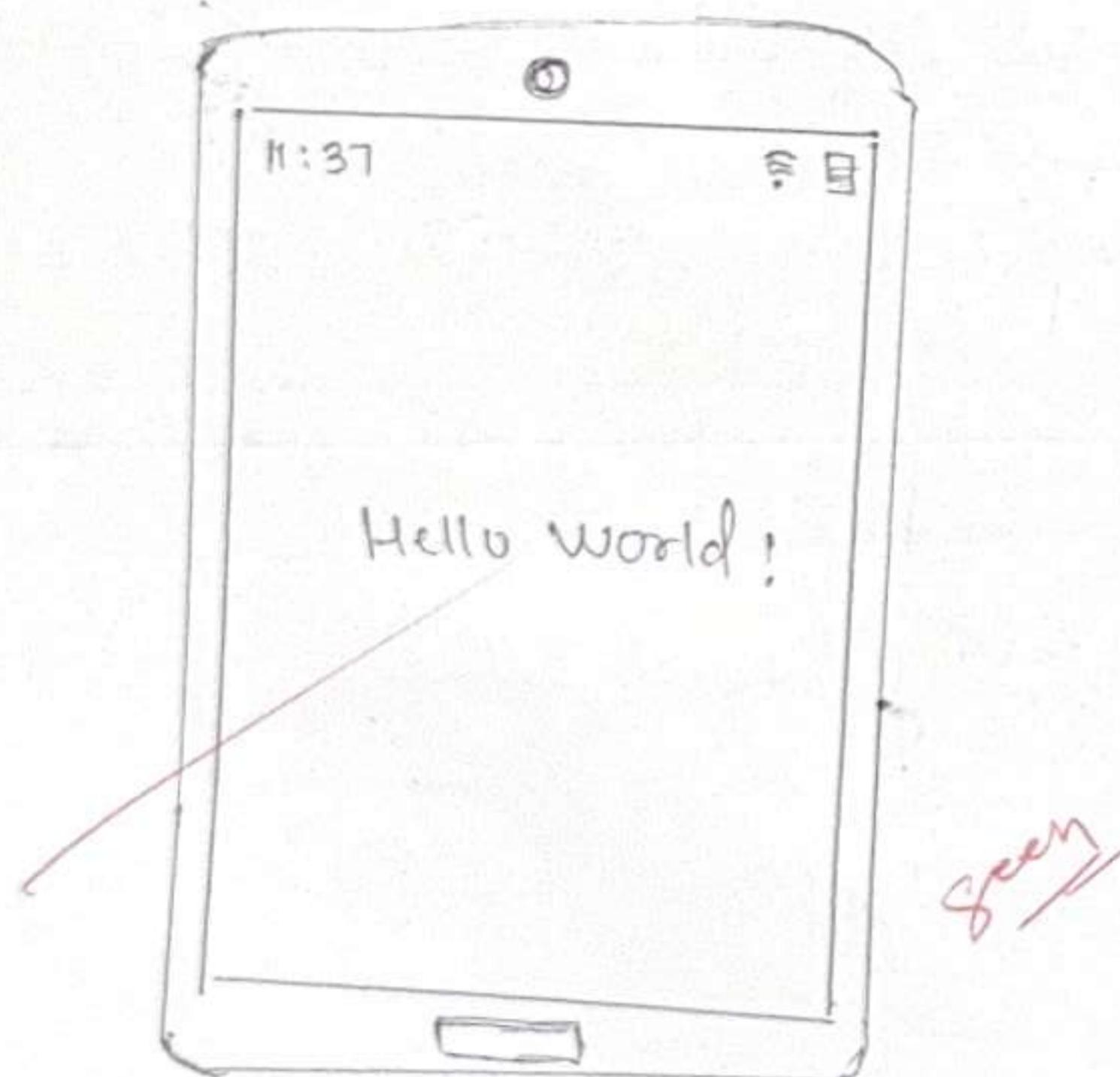
Steps:-

Open Android Studio → click New project →

Select 'Empty view Activity' → click next →

Name the project → Then select SDK 'API 21'

('Lollipop', ~~Android 5.0~~)

Design ViewOutput:activity_main.xml

```

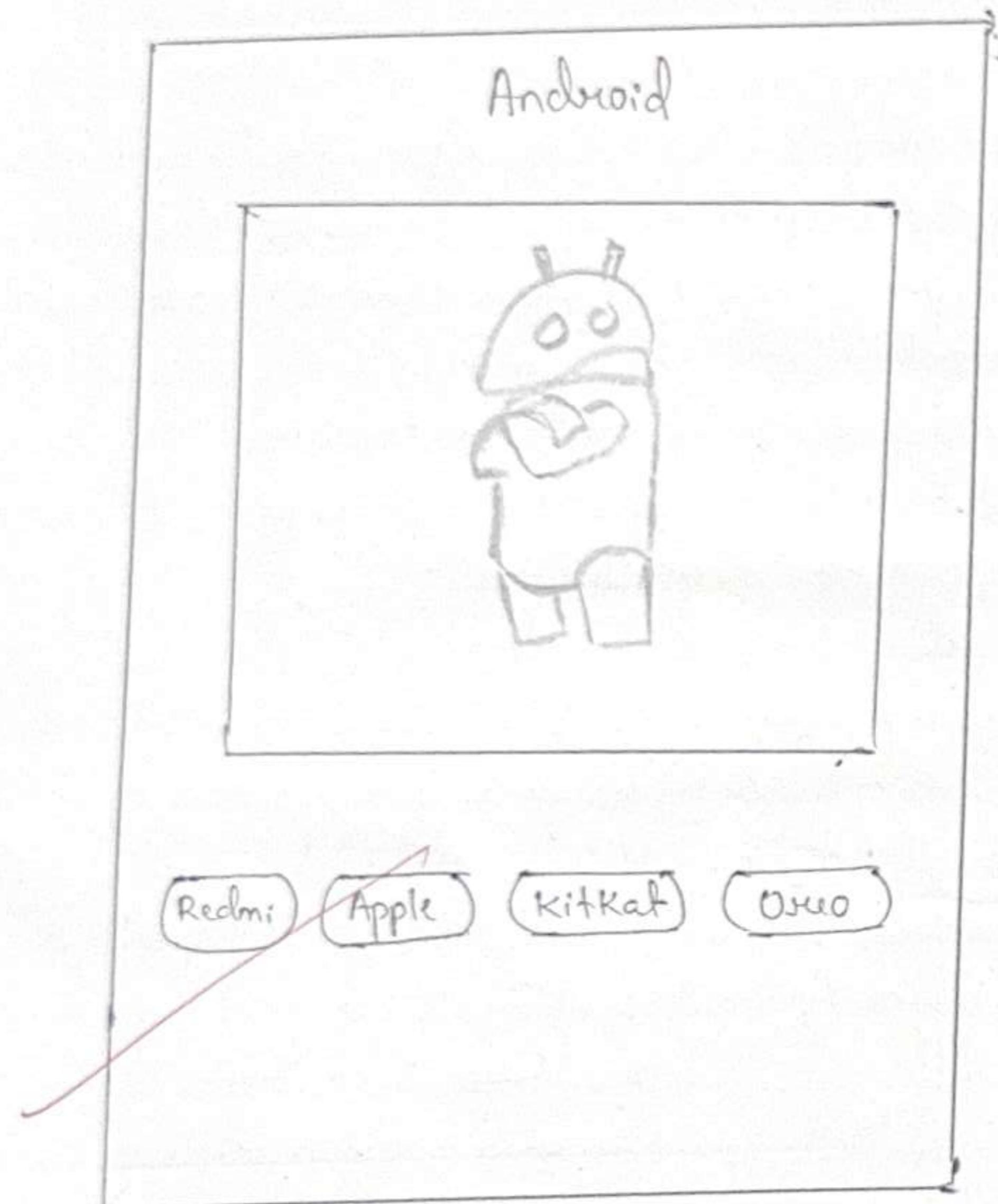
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout xmlns
    android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match-parent"
    android:layout_height="match-parent"
    tools:context=".MainActivity">

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</androidx.constraintlayout.widget.ConstraintLayout>

```

Moral
Btw if
do

Design View



A+

Practical NO :- 02

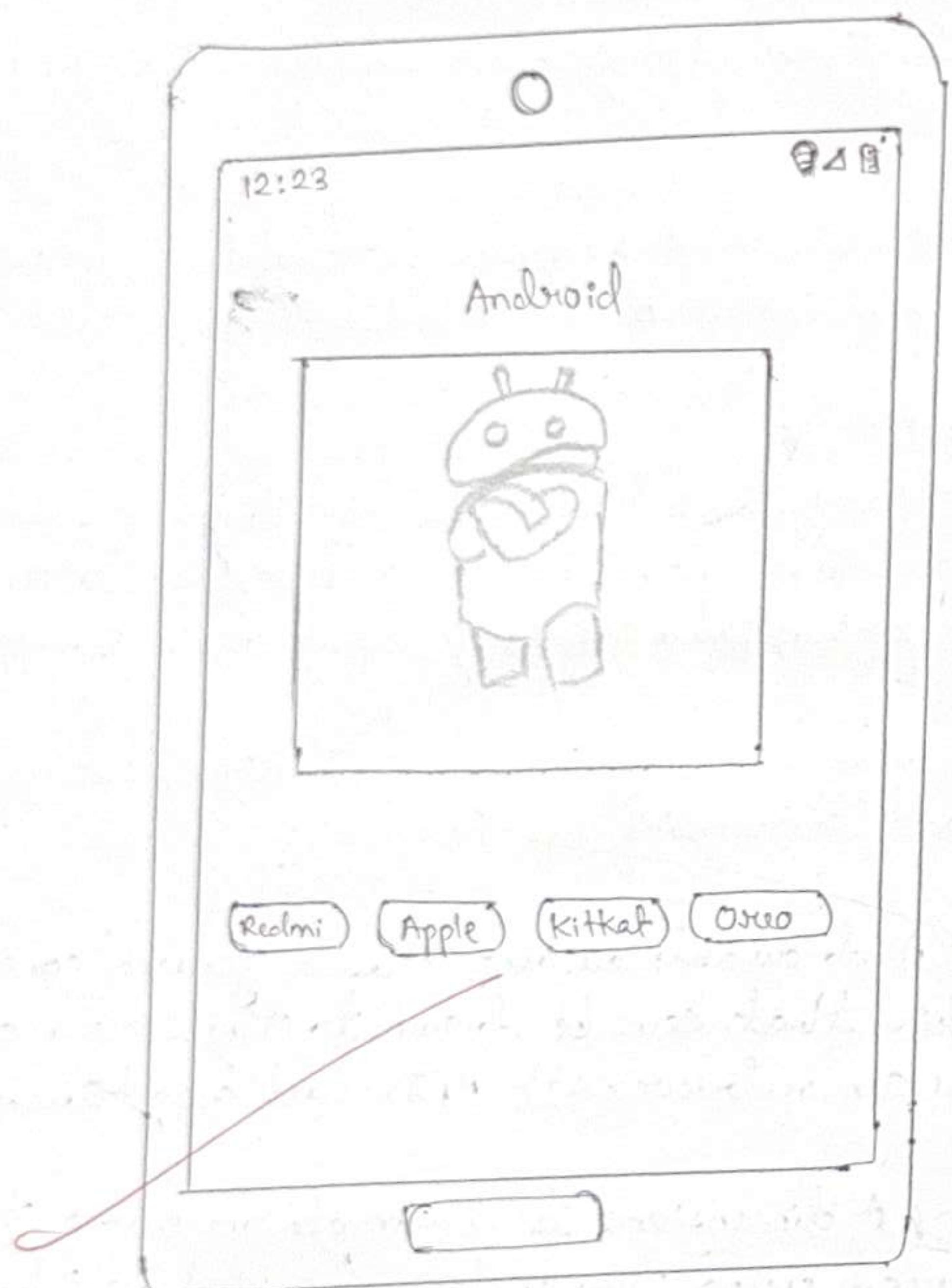
Date:- 10/03/2025

Aim:- Programming Resources Android Resources : (color, theme, string, Drawable, Dimension, Image).

Theory:-

- 1) color: A color is a simple resource that is referenced using the value provided in the name attribute, not the name of the XML file.
- 2) Theme: A theme is a set of styles or attributes such as color, type and shape, which can affect the look and feel of a user's mobile or large-screen device & app experience.
- 3) String: A string resource provides text string for your application with optional text styling and formatting.
- 4) Drawable: A drawable resource is a general concept for a graphic that can be drawn to the screen and that you can retrieve with APIs such as getDrawable.
- 5) Dimension: A dimension is a simple resource that is referenced using the value provided in the name attribute, not the name of the XML file.
- 6) Image: In android development, resources like image are stored in the res/drawable directory.

Output:



Screen

activity_main.xml

```
<?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">
```

<TextView

```
    android:id="@+id/textView"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Android"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="0.143" />
```

<ImageView

```
    android:id="@+id/imageView"
    android:layout_width="240dp"
    android:layout_height="295dp"
```

```
    app:srcCompat="@drawable/image"
    tools:layout_editor_absoluteX="94dp"
    tools:layout_editor_absoluteY="173dp" />
```

```
<LinearLayout
    android:layout_width="409dp"
    android:layout_height="33dp"
    android:layout_marginTop="528dp"
    android:orientation="horizontal"
    app:layout_constraintTop_toTopOf="parent"
    tools:layout_editor_absoluteX="16dp">
```

```
<Button
    android:id="@+id/button1"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:text="Redmi" />
```

```
<Button
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Apple" />
```

```
<Button
    android:id="@+id/button3"
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
    android:text="kitkat"/> 
```

<Button

```
    android:id="@+id/button4"
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
```

```
    android:text="Oreo"/> 
```

</LinearLayout>

<LinearLayout

```
    android:layout_width="12dp"
```

```
    android:layout_height="12dp"
```

```
    android:orientation="vertical"
```

```
    tools:layout_editor_absoluteX="178dp"
```

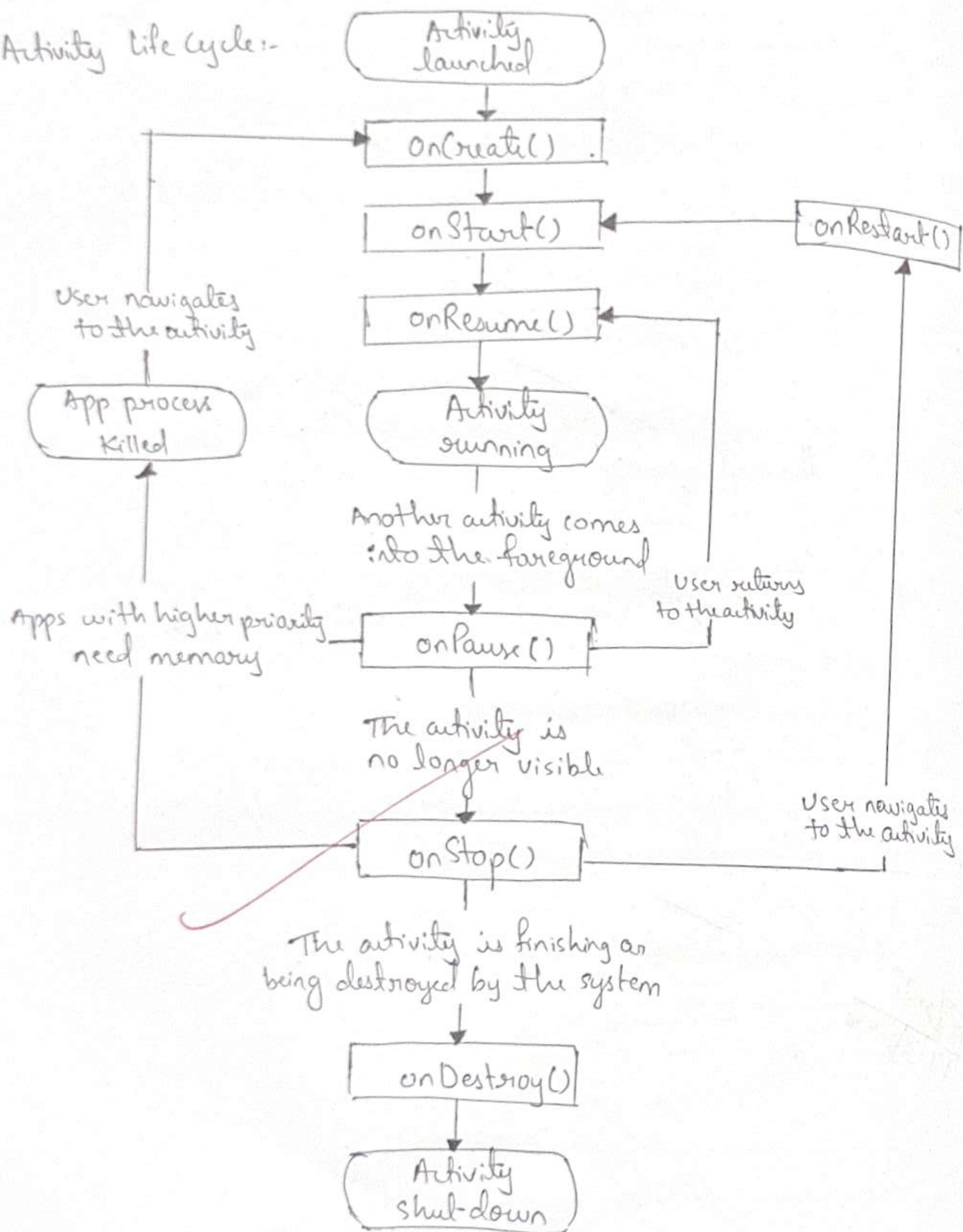
```
    tools:layout_editor_absoluteY="445dp"/>
```

</LinearLayout>

</android.constraintlayout.widget.ConstraintLayout>

Mond
17/3/2020

Activity life cycle:-



Practical No :- 03

Date:- 15/03/2025

Aim:- Programming Activities and fragments.

Activity lifecycle methods

1. onCreate(): when a user first opens an activity than the first method that gets called is called as onCreate. It acts the same as a customer constructor of a class, then when an activity is instantiated then onCreate gets called.
2. onStart():- This method is called when an activity becomes visible to the user and is called after onCreate.
3. onResume():- It is called just before the user starts interacting with the application.
4. onPause():- It is called when the app is partially visible to the user on the mobile screen.
5. onStop():- It is called when the activity is no longer visible to the user.
6. onRestart():- It is called when the activity in the stopped state is about to start again.
7. onDestroy():- It is called when the activity is cleared from

the application stack.

UseCase 1:-

when we open the activity for the first time, the sequence of state change it goes through is,
`onCreate → onStart → onResume`

UseCase 2 :-

now, let's say we are minimizing the app by pressing the home button of the phone. The state change it will go through is,
`onPause → onStop`

UseCase 3 :-

when we are moving to and from between activities, let's say Activity A and Activity B, so, we will break it down in steps.

~~First, it opened Activity A, where the following states are called initially,~~
`onCreate → onStart → onResume.`

~~Then let's say on a click of a button we opened Activity B, while opening Activity B, first, onPause will be called for Activity A and then,~~
`onCreate → onStart → onResume`

~~will be called for Activity B. Then to finish this off, onStop of Activity A will be called and finally, Activity B would be loaded.~~

override fun onStart()

{

super.onStart()

Log.i(TAG, "onStart")

}

override fun onResume()

{

super.onResume()

Log.i(TAG, "onResume")

}

override fun onPause()

{

super.onPause()

Log.i(TAG, "onPause")

}

~~override fun onStop()~~

{

super.onStop()

Log.i(TAG, "onStop")

}

override fun onRestart()

{

super.onRestart()

Log.i(TAG, "onRestart")

}

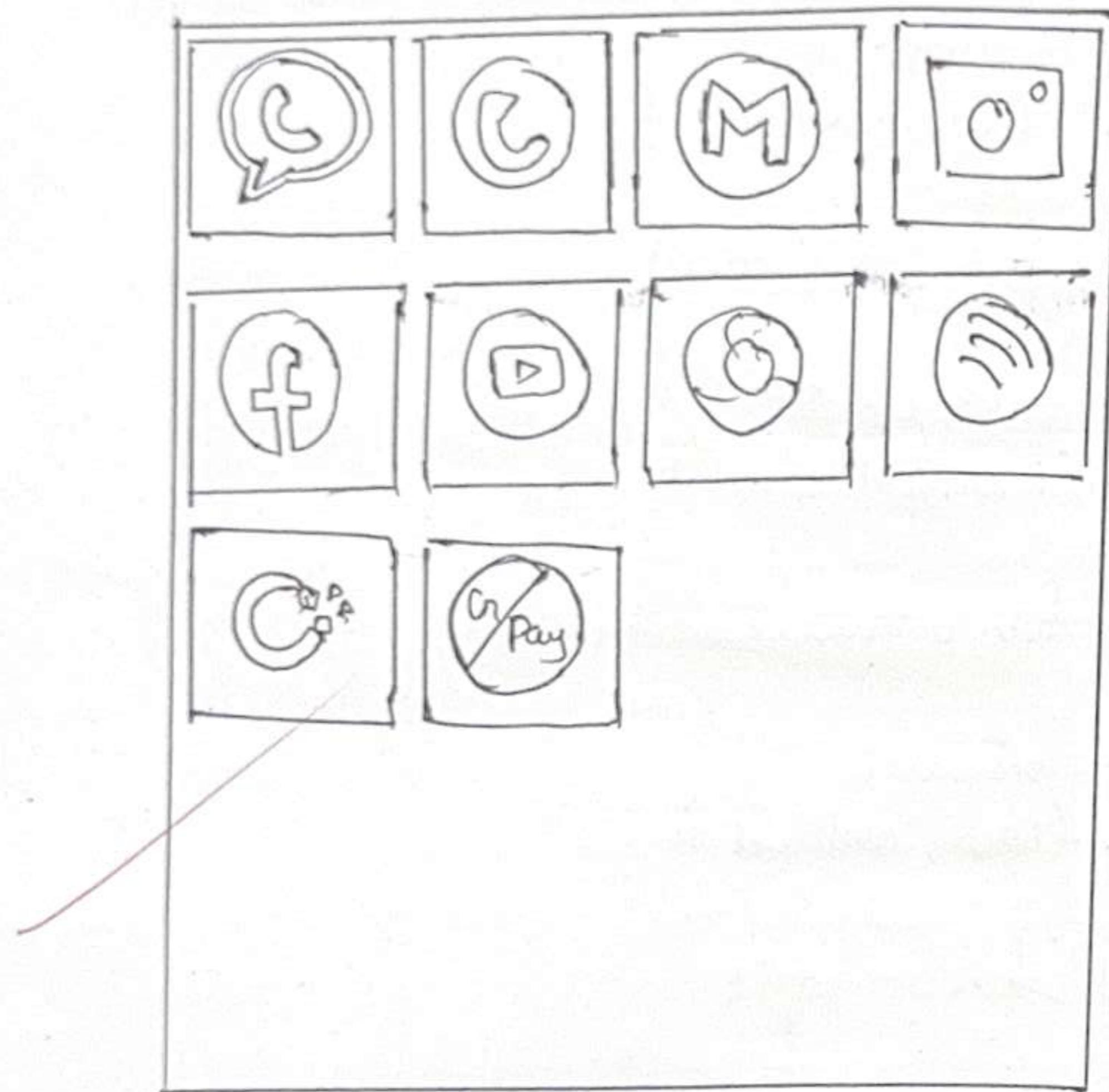
~~override fun onDestroy()~~

{

super.onDestroy()

Log.i(TAG, "onDestroy")

}

Design ViewPractical No : 04

Date: 17/03/2025

Aim:- Program related to different layouts.

- i] Grid Layout
- ii] Linear Layout
- iii] Frame Layout

i] Grid Layout..

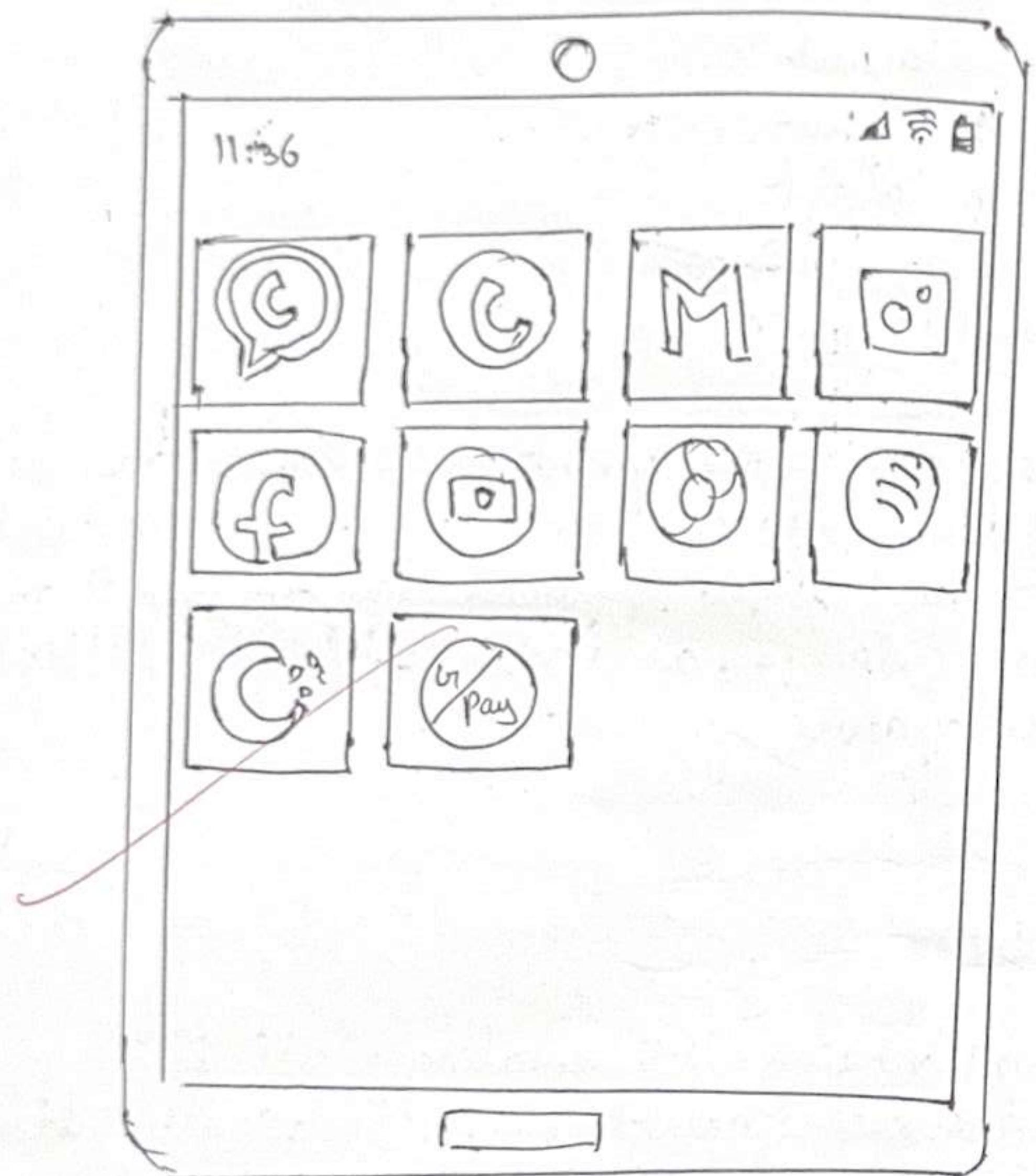
In Android UI development, a GridLayout arranges child views into a rectangular grid, using a grid of thin lines to separate the drawing area into rows, columns, and cells, allowing for flexible layout arrangements.

code:-

```

<?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">
```

Output:



```

<androidx.gridlayout.widget.GridLayout
    android:layout_width="40dp"
    android:layout_height="72dp"
    tools:layout_editor_absoluteX="1dp"
    tools:layout_editor_absoluteY="1dp">

```

```

<ImageButton
    android:id="@+id/imageButton2"
    android:layout_width="wrap-content"
    android:layout_height="wrap-content"
    app:layout_column="0"
    app:layout_row="0"
    app:srcCompat="@drawable/whatsapp1"/>

```

```

<ImageButton
    android:id="@+id/imageButton3"
    android:layout_width="wrap-content"
    android:layout_height="wrap-content"
    app:layout_column="0"
    app:layout_row="0"
    app:srcCompat="@drawable/contact1"/>

```

```

<ImageButton
    android:id="@+id/imageButton4"
    android:layout_width="wrap-content"
    android:layout_height="wrap-content"
    app:layout_column="0"
    app:layout_row="0"/>

```

```
    app:layout_row = "0"
    app:srcCompat = "@drawable/email1"/>
```

```
<ImageButton
    android:id = "@+id/imageButton5"
    android:layout_width = "wrap-content"
    android:layout_height = "wrap-content"
    app:layout_column = "03"
    app:layout_row = "0"
    app:srcCompat = "@drawable/install"/>
```

```
<ImageButton
    android:id = "@+id/imageButton6"
    android:layout_width = "wrap-content"
    android:layout_height = "wrap-content"
    app:layout_column = "04"
    app:layout_row = "0"
    app:srcCompat = "@drawable/facebook1"/>
```

```
<ImageButton
    android:id = "@+id/imageButton7"
    android:layout_width = "wrap-content"
    android:layout_height = "wrap-content"
    app:layout_column = "05"
    app:layout_row = "0"
    app:srcCompat = "@drawable/youtube1"/>
```

```

<ImageButton
    android:id="@+id/imageButton8"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_column="06"
    app:layout_row="0"
    app:srcCompat="@drawable/chrome1"/>

```

```

<ImageButton
    android:id="@+id/imageButton9"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_column="07"
    app:layout_row="0"
    app:srcCompat="@drawable/Sputify1"/>

```

```

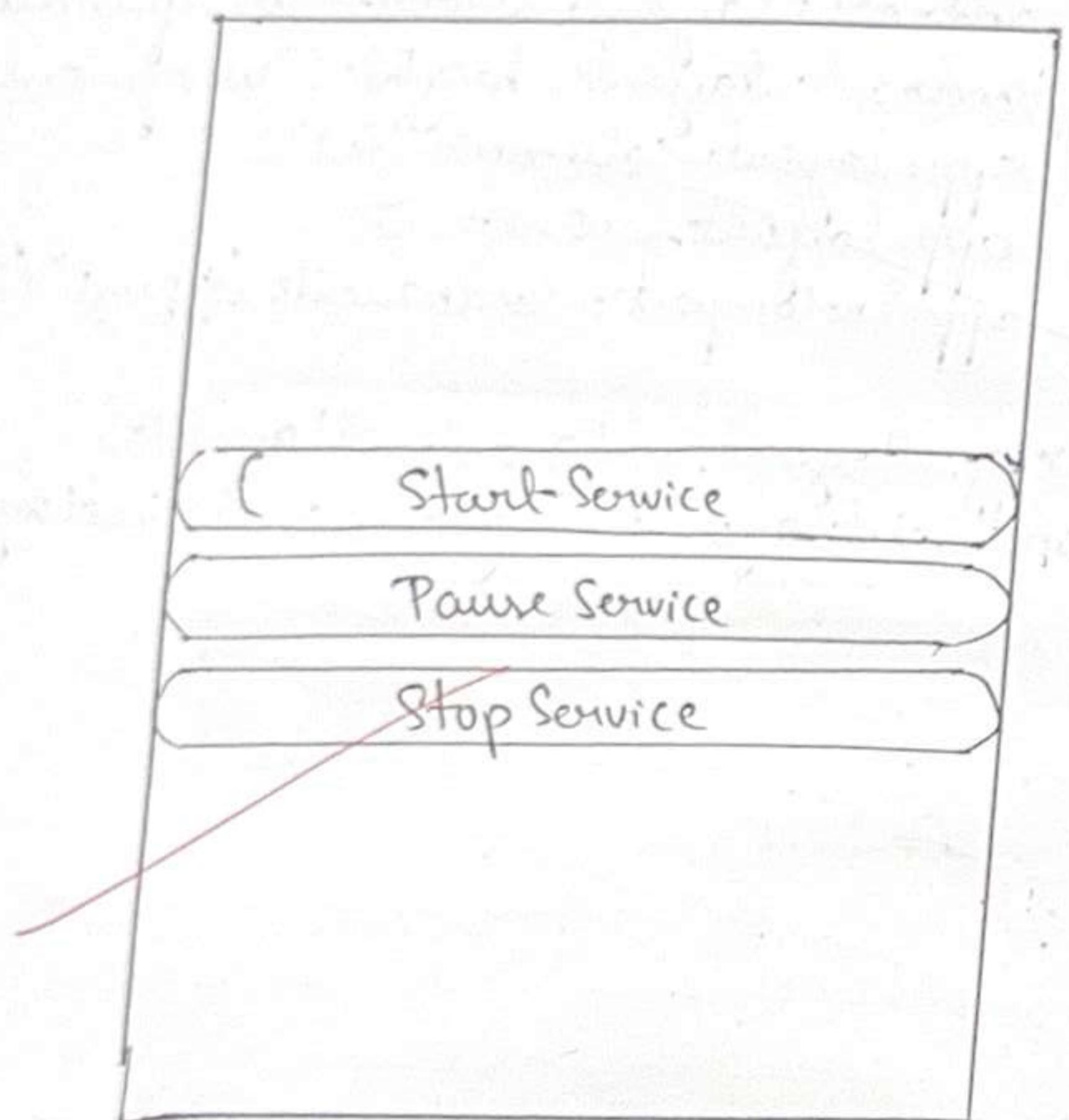
<ImageButton
    android:id="@+id/imageButton10"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_columns="08"
    app:layout_row="0"
    app:srcCompat="@drawable/Opera1"/>

```

```
<ImageButton  
    android:id="@+id/imageButton1"  
    android:layout_width="wrap-content"  
    android:layout_height="wrap-content"  
    app:layout-column="09"  
    app:layout_row="0"  
    app:srcCompat="@drawable/gpay"/>
```

```
< ?>  
</androidx.constraintlayout.widget.ConstraintLayout>
```

Design : View :-



ii) &LinearLayout:-

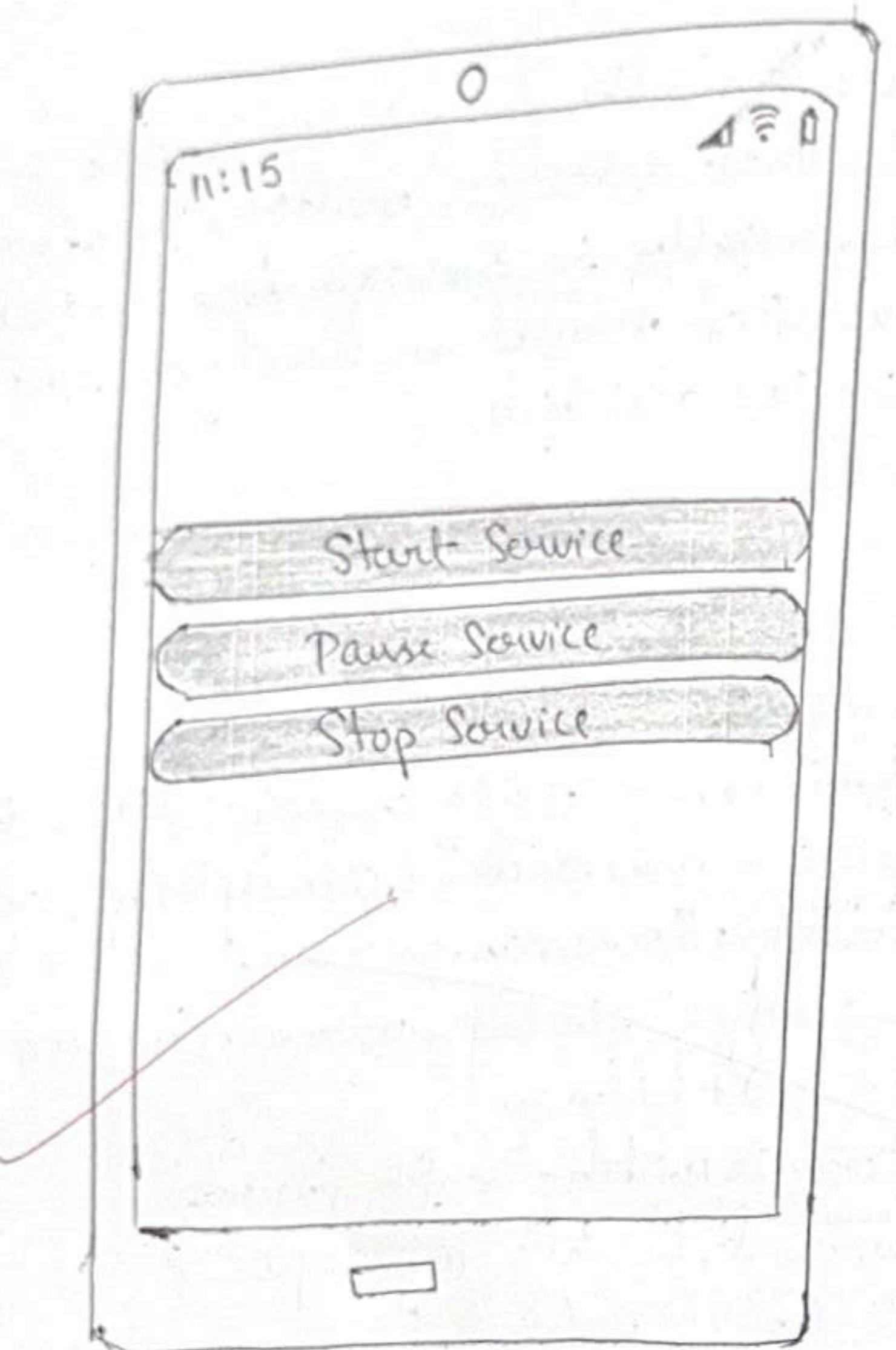
LinearLayout is one of the most basic layouts in android studio, that arranges multiple sub-views (UI elements) sequentially in a single direction i.e. horizontal or vertical manner by specifying the android:orientation attribute.

code :-

```
<?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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity">
```

<LinearLayout

```
    android:layout_width = "409dp"
    android:layout_height = "729dp"
    android:orientation = "vertical"
    tools:layout_editor_absoluteY="1dp"
    tools:layout_editor_absoluteX="1dp"
    android:gravity = "center">
```

Output:

8/22

<Button

```

    android:id="@+id/button"
    android:layout_width="match-parent"
    android:layout_height="wrap-content"
    android:backgroundTint="@color/design-
        default_color-primary"
    android:text="Start Service"
    android:textColorLink="#2135A3"/>

```

<Button

```

    android:id="@+id/button2"
    android:layout_width="match-parent"
    android:layout_height="wrap-content"
    android:backgroundTint="@color/design-
        default_color-secondary"
    android:text="Pause Service"
    android:textColorLink="#03A9F4"/>

```

<Button

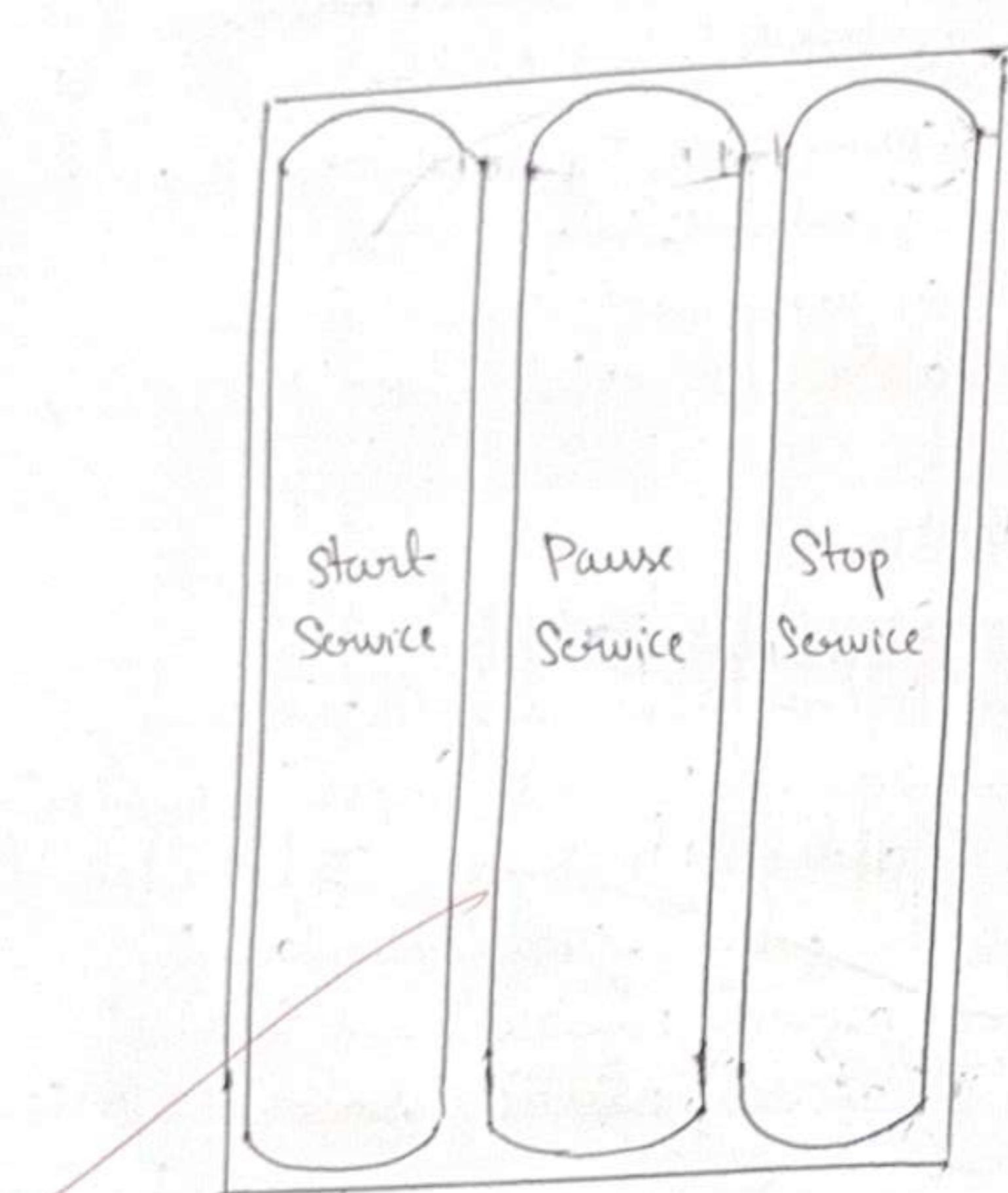
```

    android:id="@+id/button3"
    android:layout_width="match-parent"
    android:layout_height="wrap-content"
    android:backgroundTint="@color/design-
        default_color-primary"
    android:text="Stop Service"
    android:textColorLink="#FF5722"/>

```

</LinearLayout>

<androidx.constraintlayout.widget.ConstraintLayout>

Design View

Code 2:-

```
<?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:id="@+id/main"
    android:layout_width="match-parent"
    android:layout_height="match-parent"
    tools:context=".MainActivity">
```

```
<LinearLayout
    android:layout_width="match-parent"
    android:layout_height="match-parent"
    android:gravity="center"
    android:orientation="horizontal"
    tools:layout_editor_absoluteX="1dp"
    tools:layout_editor_absoluteY="1dp">
```

<Button

```
    android:id="@+id/button"
    android:layout_width="wrap-content"
    android:layout_height="match-parent"
    android:layout_margin="10dp"
    android:layout_weight="1"
    android:backgroundTint="#2196F3"
```

Output

```
    android:text="Start Service" />
```

<Button

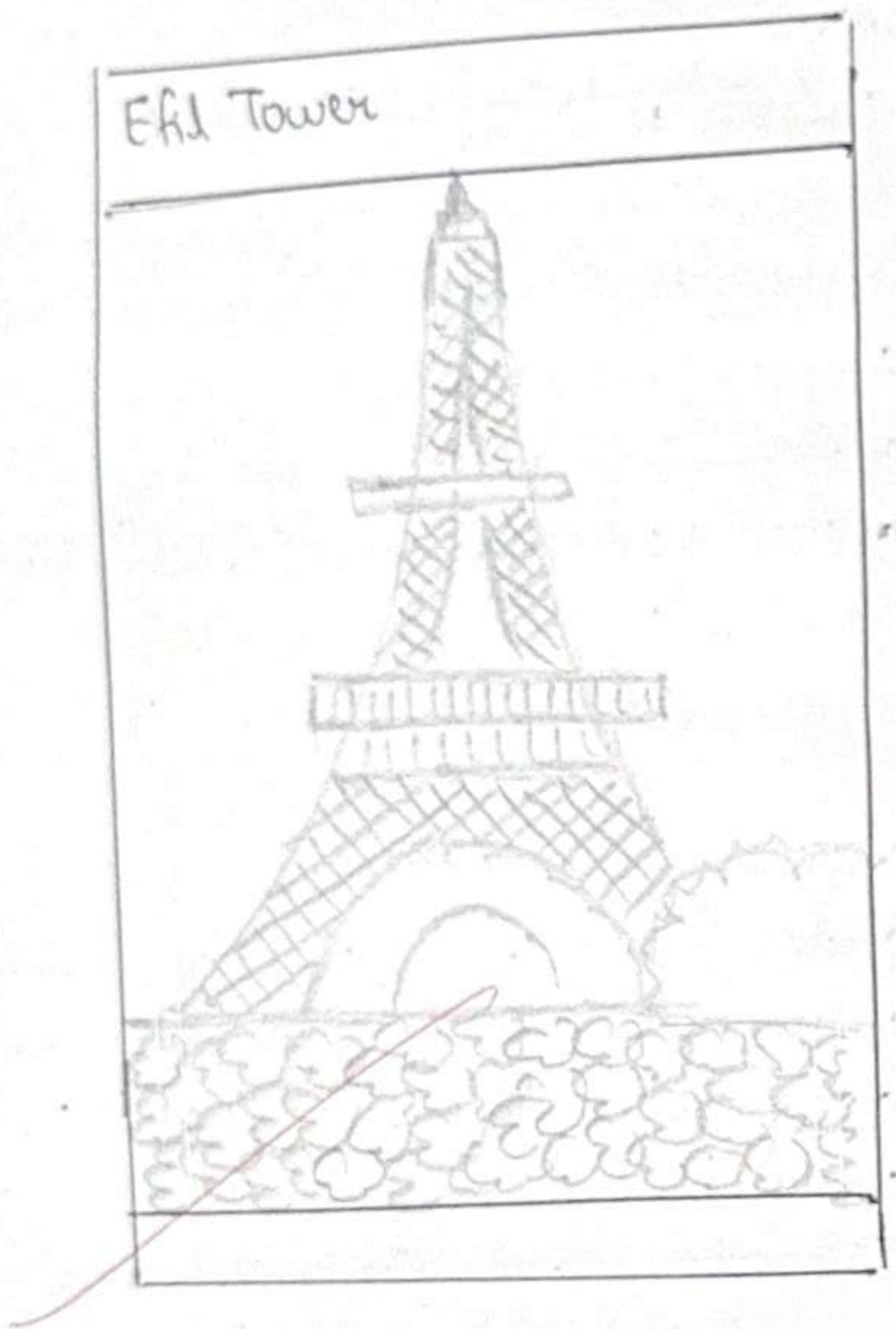
```
    android:id="@+id/button2"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_margin="10dp"
    android:layout_weight="1"
    android:backgroundTint="#FF5722"
    android:text="Pause Service" />
```

<Button

```
    android:id="@+id/button3"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_margin="10dp"
    android:layout_weight="1"
    android:backgroundTint="#9C27B0"
    android:text="Stop Service" />
```

</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

Design ViewIII) Frame Layout:-

In Android development, a FrameLayout is a layout container that displays child views stacked on top of each other, with the most recently added view appearing on top.

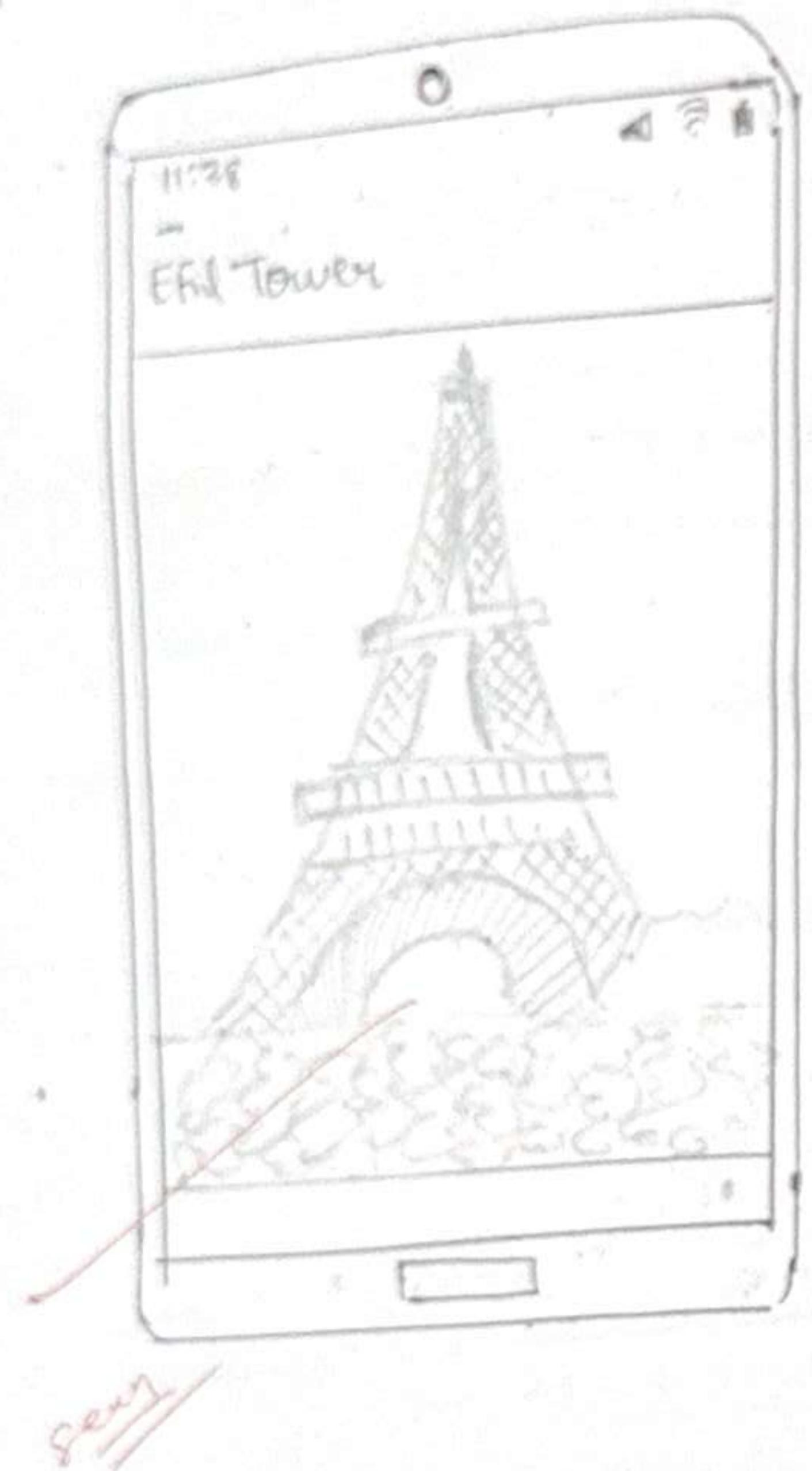
It is often used for simple layout layering and is commonly employed to hold a single child view or as a container for fragments.

Code:-

```
<?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:id = "@+id/main"
    android:layout_width = "match_parent"
    android:layout_height = "match_parent"
    tools:context = ".MainActivity">
```

<FrameLayout

```
    android:layout_width = "409dp"
    android:layout_height = "729dp"
    tools:layout_editor_absoluteX = "1dp"
    tools:layout_editor_absoluteY = "1dp">
```

Output:

<ImageView

android:id="@+id/imageView"

android:layout_width="match_parent"

android:layout_height="match_parent"

app:safeCompat="@drawable/image" />

<TextView

android:id="@+id/textView"

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:text="Eiff Tower" />

</FrameLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

Design View

STUDENT INFORMATION		
Roll No	Name	Address
1	Iram	Uran
2	Vanshika	Uran
3	Manasi	Uran

iv) Table Layout:-

TableLayout in Android is a ViewGroup subclass that is designed to align child views in rows and columns like a grid structure.

It automatically arranges all the child elements into rows and columns without displaying any border lines between cells.

The TableLayout's functionality is almost similar to an HTML table, where the number of columns in the layout is determined by the row with the most cells.

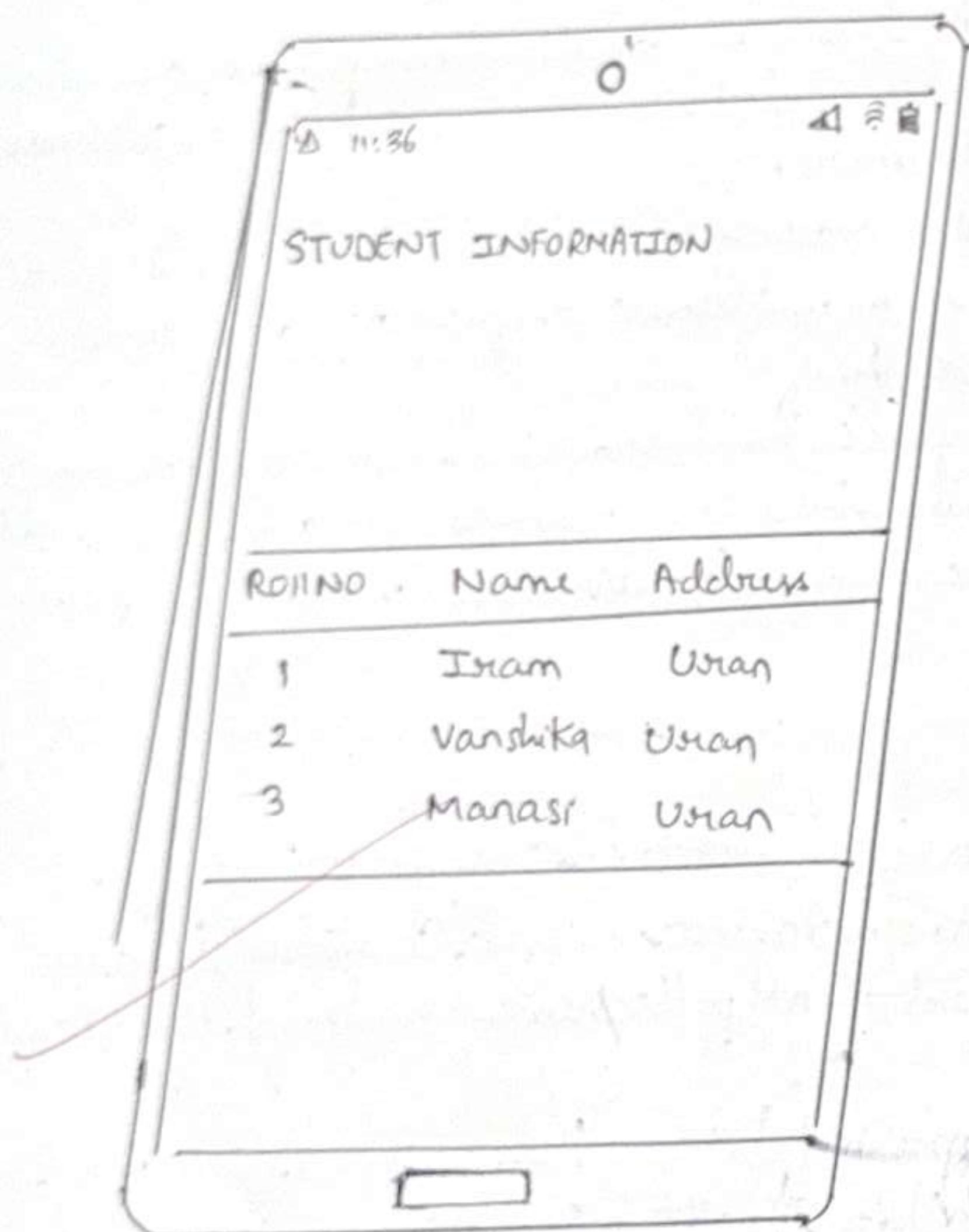
Code:-

```

<?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">
```

```

<TableLayout
    android:layout_width = "409dp"
    android:layout_height = "729dp"
```

Output:

see //

android:gravity = "center"
 tools:layout_editor_absoluteX = "1dp"
 tools:layout_editor_absoluteY = "4dp"

<TableRow

android:layout_width = "match_parent"
 android:layout_height = "match_parent"
 android:background = "#03A9F4">

<TextView

android:id = "@+id/textView1"
 android:layout_width = "111dp"
 android:layout_height = "match_parent"
~~android:backgroundTint = "@color/design-default-color-error"~~
 android:text = "ROLL NO" />

<TextView

android:id = "@+id/textView2"
 android:layout_width = "171dp"
 android:layout_height = "match_parent"
 android:text = "Name" />

<TextView

android:id = "@+id/textView3"
 android:layout_width = "171dp"
 android:layout_height = "match_parent"

```
        android:text="Address"/>  
    </TableRow>  
    <TextView  
        android:layout-width="match-parent"  
        android:layout-height="match-parent">  
  
        <TextView  
            android:id="@+id/textView4"  
            android:layout-width="wrap-content"  
            android:layout-height="wrap-content"  
            android:text="I" />  
  
        <TextView  
            android:id="@+id/textView5"  
            android:layout-width="wrap-content"  
            android:layout-height="wrap-content"  
            android:text="Item" />  
  
        <TextView  
            android:id="@+id/textView6"  
            android:layout-width="wrap-content"  
            android:layout-height="wrap-content"  
            android:text="Item" />  
    </TableRow>
```

STUDENT INFORMATION		
ROLL NO	Name	Address
1	Iram	Uran
2	Vanshika	Uran
3	Manasi	Uran

iv)

Table Layout:-

TableLayout in Android is a ViewGroup subclass that is designed to align child views in rows and columns like a grid structure.

It automatically arranges all the child elements into rows and columns without displaying any border lines between cells.

The Table layout's functionality is almost similar to an HTML table, where the number of columns in the layout is determined by the row with the most cells.

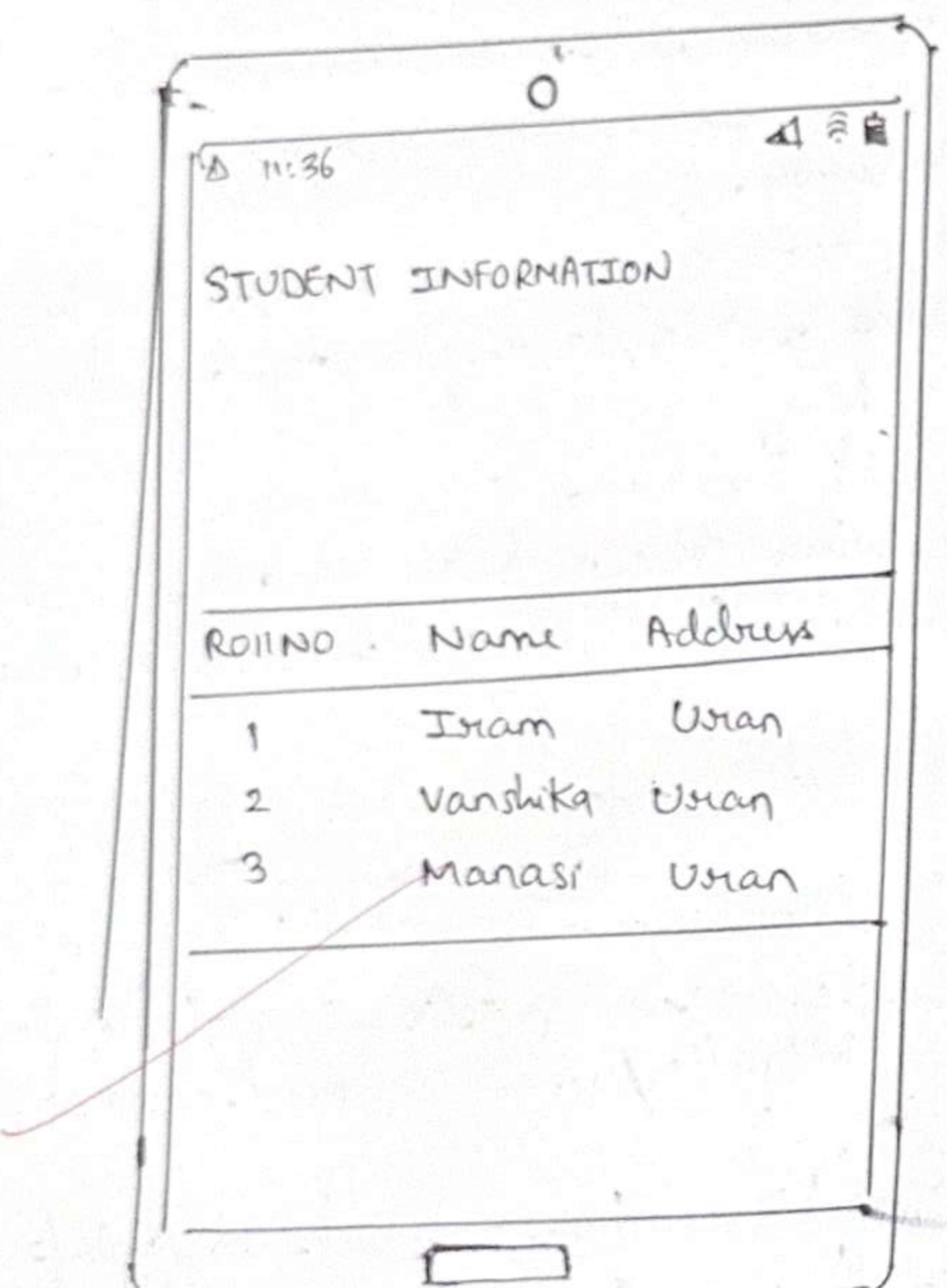
Code:-

```
<?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">
```

TableLayout

```
    android:layout_width="409dp"
    android:layout_height="729dp"
```

Output:



see //

```
    android:gravity = "center"
    tools:layout_editor_absoluteX = "1dp"
    tools:layout_editor_absoluteY = "4dp"
```

<TableRow

```
        android:layout_width = "match_parent"
        android:layout_height = "match_parent"
        android:background = "#03A9F4">
```

<TextView

```
        android:id = "@+id/textView1"
        android:layout_width = "111dp"
        android:layout_height = "match_parent"
        android:background_tint = "@color/design-
        default_color_error"
        android:text = "Roll No" />
```

<TextView

```
        android:id = "@+id/textView2"
        android:layout_width = "171dp"
        android:layout_height = "match_parent"
        android:text = "Name" />
```

<TextView

```
        android:id = "@+id/textView3"
        android:layout_width = "171dp"
        android:layout_height = "match_parent"
```

```

    android:text="Address"/>
</TableRow>
<TextView
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <TextView
        android:id="@+id/textView4"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="I" />

    <TextView
        android:id="@+id/textView5"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Tram" />

    <TextView
        android:id="@+id/textView6"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Tram" />
</TableRow>

```

<TableRow

 android:layout_width="match-parent"

 android:layout_height="match-parent"

<TextView

 android:id="@+id/textView7"

 android:layout_width="wrap-content"

 android:layout_height="wrap-content"

 android:text="2" />

<TextView

 android:id="@+id/textView8"

 android:layout_width="wrap-content"

 android:layout_height="wrap-content"

 android:text="Vanshika" />

<TextView

 android:id="@+id/textView9"

 android:layout_width="wrap-content"

 android:layout_height="wrap-content"

 android:text="Aryan" />

</TableRow>

<TextView

<TableRow>

 android:layout_width="match-parent"

 android:layout_height="match-parent"

```
<TextView  
    android:id="@+id/textView10"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="3" />
```

```
<Text View  
    android:id="@+id/textView11"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Manasi" />
```

```
<TextView  
    android:id="@+id/textView12"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Uran" />
```

</TableRow>
</TableLayout>

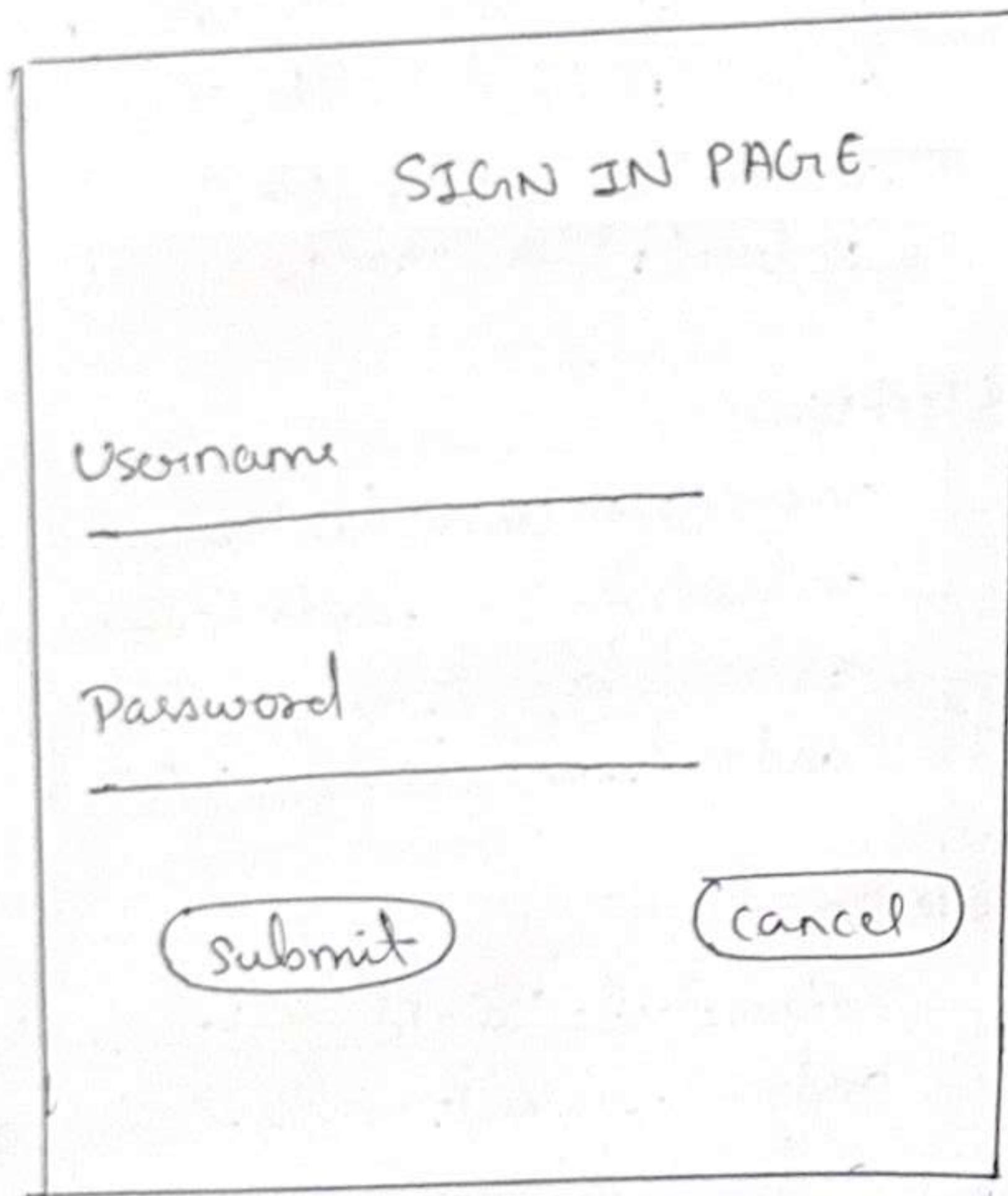
<TextView

```
    android:id="@+id/textView"  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="STUDENT INFORMATION"  
    tools:layout_editor_absoluteX="35dp"  
    tools:layout_editor_absoluteY="15dp" />
```

</androidx.constraintlayout.widget.ConstraintLayout>

Manasi
Uran
3

Design View

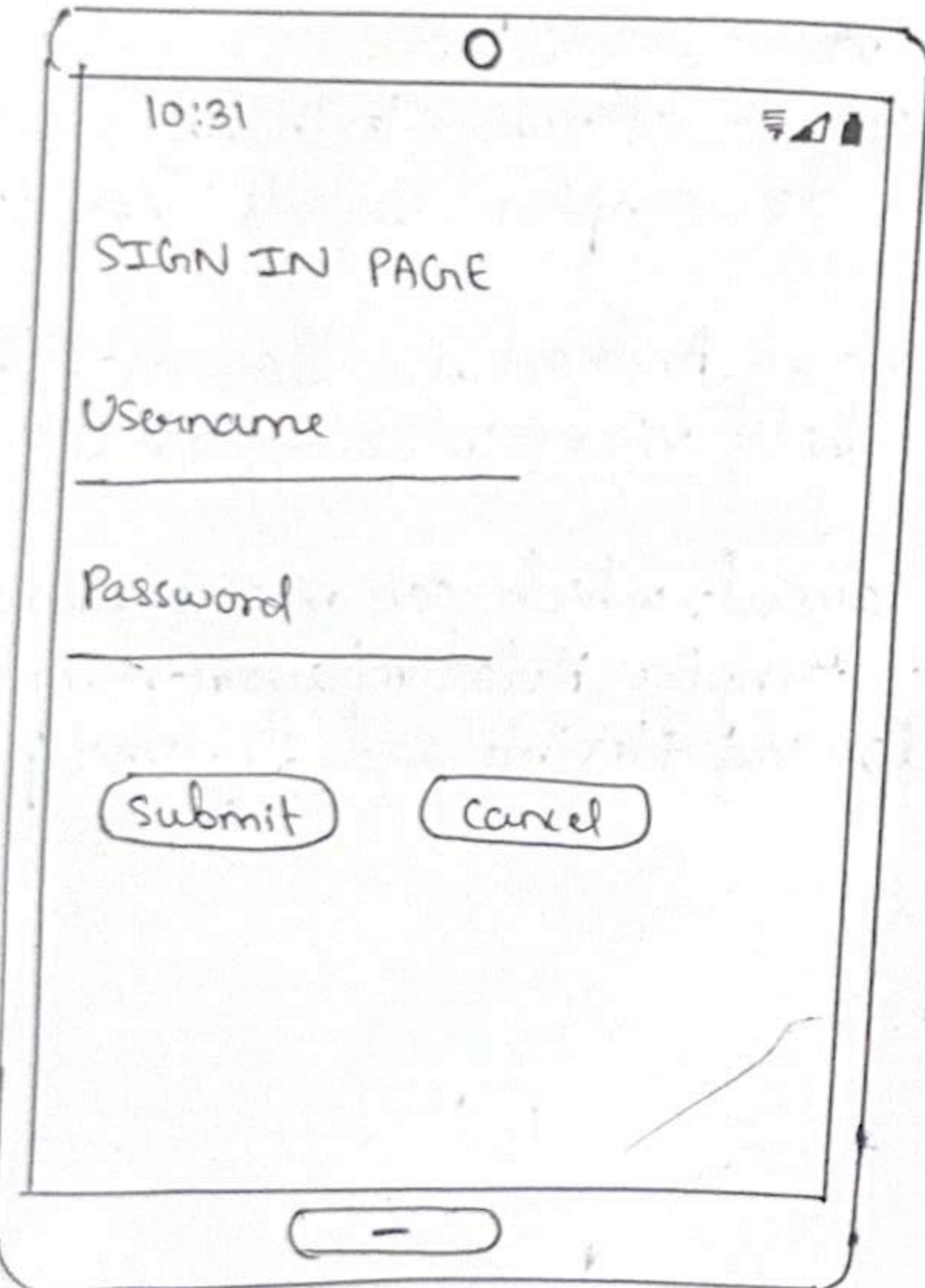


v) Relative Layout :-

RelativeLayout is a ViewGroup that allows you to position child views relative to each other or to the parent layout, offering flexibility in UI design compared to simpler layouts like Linear layout.

Relative layout in Android is a layout that arranges its child views in relation to each other.

Unlike linear layout, which can make element arrangement complex, RelativeLayout simplifies the process by allowing flexible and dynamic positioning.

Output

code:

```
<?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:id="@+id/main"
    android:layout_width="match-parent"
    android:layout_height="match-parent"
    tools:context=".MainActivity">
```

<TextView

```
    android:id="@+id/textView1"
    android:layout_width="99dp"
    android:layout_height="36dp"
    android:text="SIGN IN PAGE"
    tools:layout_editor_absoluteX="148dp"
    tools:layout_editor_absoluteY="90dp">
```

<Relativelayout

```
    android:layout_width="362dp"
    android:layout_height="276dp"
    tools:layout_editor_absoluteX="14dp"
    tools:layout_editor_absoluteY="90dp">
```

<EditText

```
    android:layout_width="240dp"
    android:layout_height="276dp"
    tools:layout_editor_absoluteX="1"
    android:layout_marginTop="24dp"
    android:id="@+id/username" />
```

<EditText

```
    android:layout_width="240dp"
    android:layout_height="57dp"
    android:layout_marginTop="97dp"
    android:id="@+id/password" />
```

<Button

```
    android:layout_width="wrap-content"
    android:id="@+id/button"
    android:layout_width="wrap-content"
    android:layout_height="wrap-content"
    android:layout_marginTop="200dp"
    android:layout_marginEnd="102dp"
    android:backgroundTint="#4CAF50"
    android:text="Submit" />
```

<Button

```
    android:id="@+id/button2"
    android:layout_width="wrap-content"
    android:layout_height="wrap-content"
```

android:layout_marginStart="162dp"
 android:layout_marginTop="200dp"
 android:layout_marginEnd="50dp"
 android:backgroundTint="#F44336"
 android:text="cancel"/>)

· </RelativeLayout>
</androidx.constraintlayout.widget.ConstraintLayout>

Practical No:- 05

Date:- 28/03/25

Aim:- Create android application to create a simple calculator.

code:

```
<?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:id="@+id/main"
    android:layout_width="match-parent"
    android:layout_height="match-parent"
    tools:context=".MainActivity">
```

```
<TextView
    android:id="@+id/textView"
    android:layout_width="390dp"
    android:layout_height="58dp"
    android:text="CALCULATOR"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent" />
```

<Button

```

        android:id="@+id/button2"
        android:layout_width="76dp"
        android:layout_height="58dp"
        android:layout_marginStart="16dp"
        android:layout_marginTop="248dp"
        android:backgroundTint="#9C27B0"
        android:text="AC"
        android:textAppearance="@style/TextAppearance
        AppCompat.Body2"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@+id/firstView"/>
    
```

<Button

```

        android:id="@+id/button3"
        android:layout_width="76dp"
        android:layout_height="58dp"
        android:layout_marginStart="16dp"
        android:layout_marginTop="240dp"
        android:backgroundTint="#9C27B0"
        android:text="X"
        android:textAppearance="@style/TextAppearance
        AppCompat.BodyDisplay1"
        app:layout_constraintStart_toEndOf="@+id/button2"
        app:layout_constraintTop_toBottomOf="@+id/firstView"/>
    
```

<Button

```

    android:id="@+id/button4"
    android:layout_width="73dp"
    android:layout_height="58dp"
    android:layout_marginStart="24dp"
    android:layout_marginTop="240dp"
    android:backgroundTint="#9C27B0"
    android:text="%"
    android:textAppearance="@style/TextAppearance.
        AppCompat.Display1"
    app:layout_constraintStart_toEndOf="@+id/button3"
    app:layout_constraintTop_toBottomOf="@+id/textView"

```

<Button

```

    android:id="@+id/button5"
    android:layout_width="64dp"
    android:layout_height="57dp"
    android:layout_marginStart="28dp"
    android:layout_marginTop="240dp"
    android:backgroundTint="#9C27B0"

```