Experiment – 3.2

Student Name: Abhinav Verma   **UID:** 20BCS9258
Branch: CSE   **Section/Group:** 20BCS_KRG_DM-1A
Semester: 06   **Date of Performance:** 05/05/2023
Subject Name: Mobile Application Development Lab   **Subject Code:** 20CSP-356

1. Aim/Overview of the practical:

Implement building blocks for Android Application using different layouts (such as linear, relative and absolute).

2. System Requirements:

The required tools to develop Android applications are open source and can be downloaded from the Web. Following is the list of software's you will need before you start your Android application programming.

- ➢ Android Studio .exe file
- ➢ Java JDK5 or later version
- ➢ Java Runtime Environment (JRE) 6 Android Studio

3. Steps of Experiment:

In this experiment, we are implementing building blocks for Android Application using different layouts (such as linear, relative and absolute). And by using toast, we display whether our device is connected to power or not.

Step 01: Now, start Android Studio app and start new project as "Experiment08" and select activity as "Empty Activity" as shown below and enter activity details such as activity name, package name, save location, language as Java, minimum SDK and then select Finish.

Step 02: New window will open with MyActivity.java file, our new project is started as shown below:

Code:

MainActivity.java:

```java
package com.example.experiment08;

import androidx.appcompat.app.AppCompatActivity;
import android.content.BroadcastReceiver;
```

```java
import android.content.Context;
import android.content.Intent;
import android.content.IntentFilter;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    Button sendBtn,gotoNext;
    EditText name;
    MyBroadcastReceiver receiver = new MyBroadcastReceiver();
    IntentFilter inf = new IntentFilter();
    protected void onPause() {
        super.onPause();
        unregisterReceiver(receiver);
    }
    protected void onResume() {
        super.onResume();
        registerReceiver(receiver, inf);
    }
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        sendBtn=findViewById(R.id.submitBtn);
        gotoNext=findViewById(R.id.gotoNext);
        name=findViewById(R.id.etName);

        inf.addAction(Intent.ACTION_POWER_CONNECTED);
        inf.addAction(Intent.ACTION_POWER_DISCONNECTED);

        sendBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String n= name.getText().toString();
                Intent i=new
Intent(getApplicationContext(),MainActivity2.class);
                i.putExtra("Name",n);
                startActivity(i);
            }
        });

        gotoNext.setOnClickListener(new View.OnClickListener(){
            @Override
            public void onClick(View view){
                Intent i=new
Intent(getApplicationContext(),MainActivity3.class);
                startActivity(i);
            }
        });
```

```
        }
}
```

activity_main.xml:

```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"
    android:paddingBottom="200dp">

    <RelativeLayout
        android:id="@+id/relativeLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        tools:ignore="MissingConstraints">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginBottom="20dp"
            android:gravity="center"
            android:text="Relative Layout"
            android:textSize="18dp" />
        <EditText
            android:id="@+id/etName"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="20dp"
            android:hint="Enter Name"
            android:textSize="18dp" />
        <Button
            android:id="@+id/submitBtn"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_below="@id/etName"
            android:layout_centerInParent="true"
            android:text="Send to Activity 2" />
    </RelativeLayout>

    <LinearLayout
        android:id="@+id/linearLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        app:layout_constraintTop_toBottomOf="@+id/relativeLayout"
        android:orientation="vertical">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center"
```

```xml
            android:text="Linear Layout"
            android:textSize="18dp"/>
        <Button
            android:id="@+id/gotoNext"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Go To Activity 3"
            android:layout_gravity="center"/>
    </LinearLayout>

    <FrameLayout
        android:id="@+id/frameLayout"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        app:layout_constraintBottom_toBottomOf="parent">
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:text="Frame Layout"
            android:textSize="18dp" />
        <TextView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:gravity="center"
            android:paddingTop="20dp"
            android:text="Abhinav Verma 20BCS9258"
            android:textColor="@color/black"
            android:textSize="18dp" />
    </FrameLayout>
</androidx.constraintlayout.widget.ConstraintLayout>
```

MyBroadcastReceiver.java:

```java
package com.example.experiment08;

import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.widget.Toast;

public class MyBroadcastReceiver extends BroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        // constants to receive message
        if (intent.getAction().equals(Intent.ACTION_POWER_CONNECTED)) {
            Toast.makeText(context, "Great! The Power is Connected! :)
", Toast.LENGTH_LONG).show();
        } else if
(intent.getAction().equals(Intent.ACTION_POWER_DISCONNECTED)) {
            Toast.makeText(context, "Oops! The Power got Dis-connected!
:( ", Toast.LENGTH_LONG).show();
```
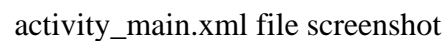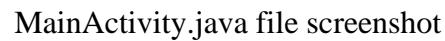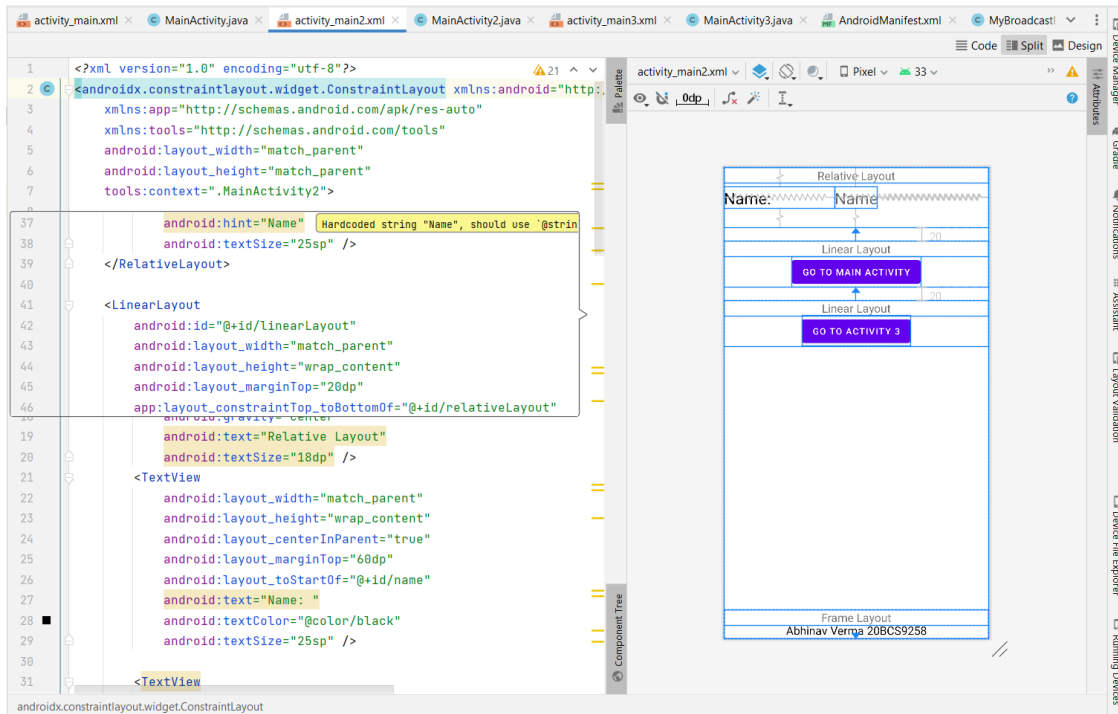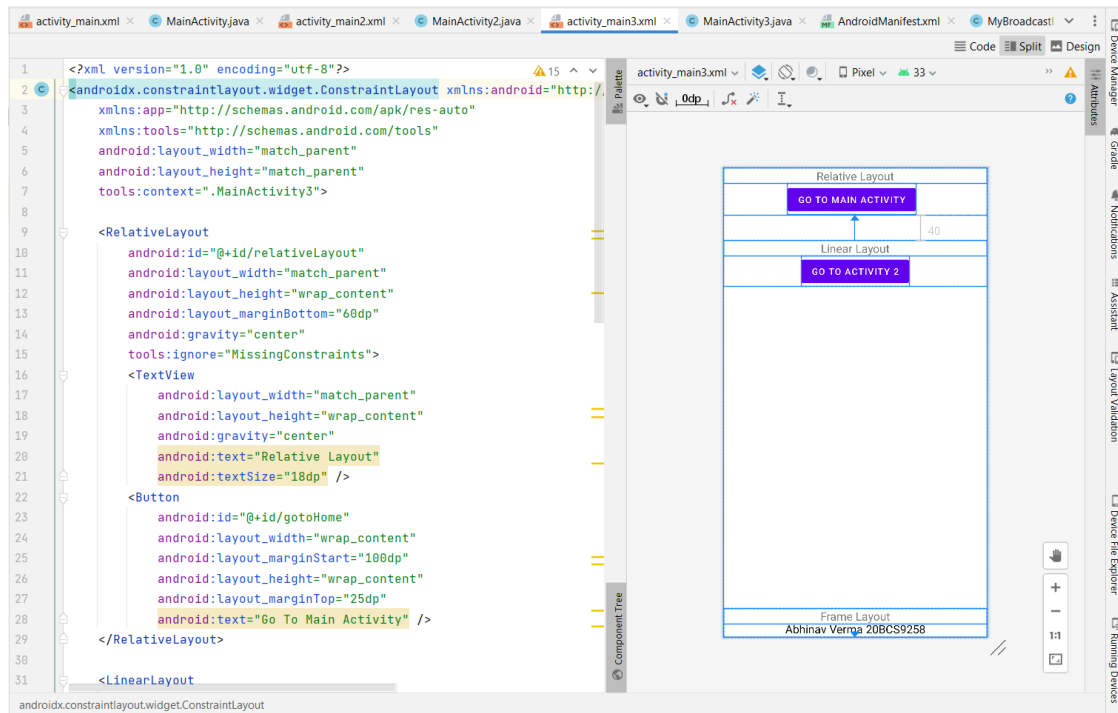
```
            }
        }
    }
```

AndroidManifest.xml:

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/Theme.Experiment08"
        tools:targetApi="31">
        <activity
            android:name=".MainActivity3"
            android:exported="false" />
        <activity
            android:name=".MainActivity2"
            android:exported="false" />
        <activity
            android:name=".MainActivity"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category
android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <!--permission for power connection status-->
        <receiver
            android:name=".MyBroadcastReceiver"
            android:exported="true">
            <intent-filter>
                <action
android:name="android.intent.action.ACTION_POWER_CONNECTED" />
                <action
android:name="android.intent.action.ACTION_POWER_DISCONNECTED" />
            </intent-filter>
        </receiver>
    </application>

</manifest>
```

MainActivity.java file screenshot

activity_main.xml file screenshot

activity_main2.xml file screenshot

activity_main3.xml file screenshot

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="Experiment08"
        android:supportsRtl="true"
        android:theme="@style/Theme.Experiment08"
        tools:targetApi="31">
        <activity...>
        <activity...>
        <activity...>

        <!--permission for power connection status-->
        <receiver
            android:name=".MyBroadcastReceiver"
            android:exported="true">
            <intent-filter>
                <action android:name="android.intent.action.ACTION_POWER_CONNECTED" />
                <action android:name="android.intent.action.ACTION_POWER_DISCONNECTED" />
            </intent-filter>
        </receiver>
    </application>

</manifest>
```
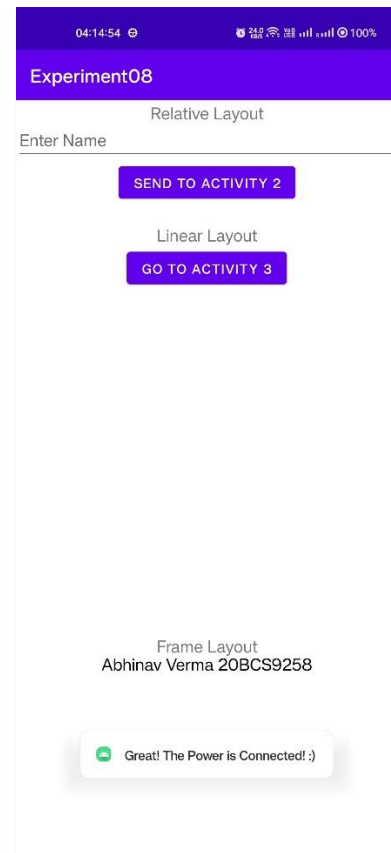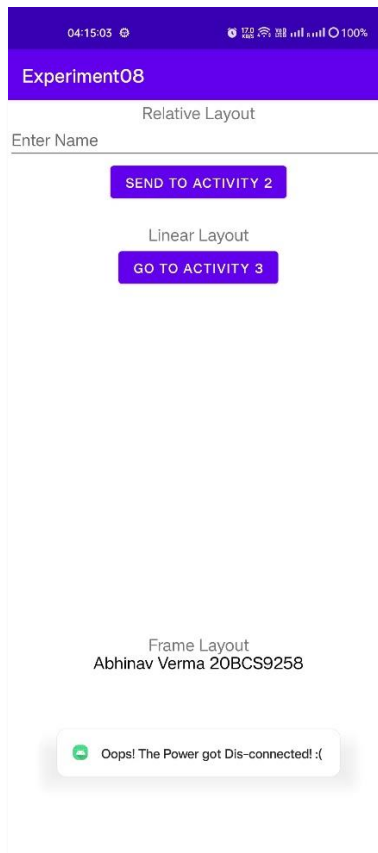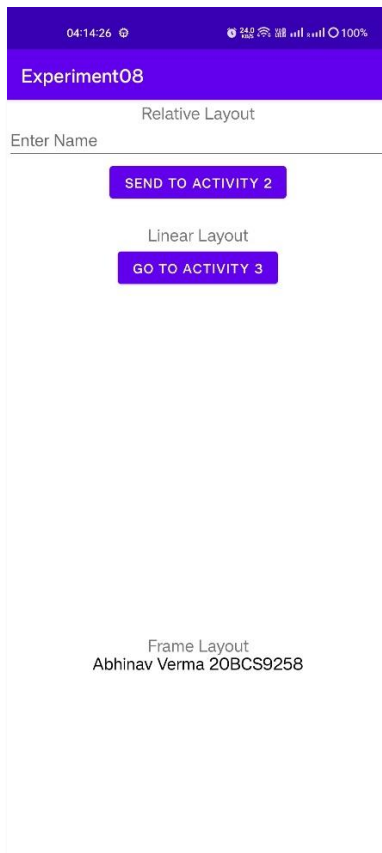
manifest > application
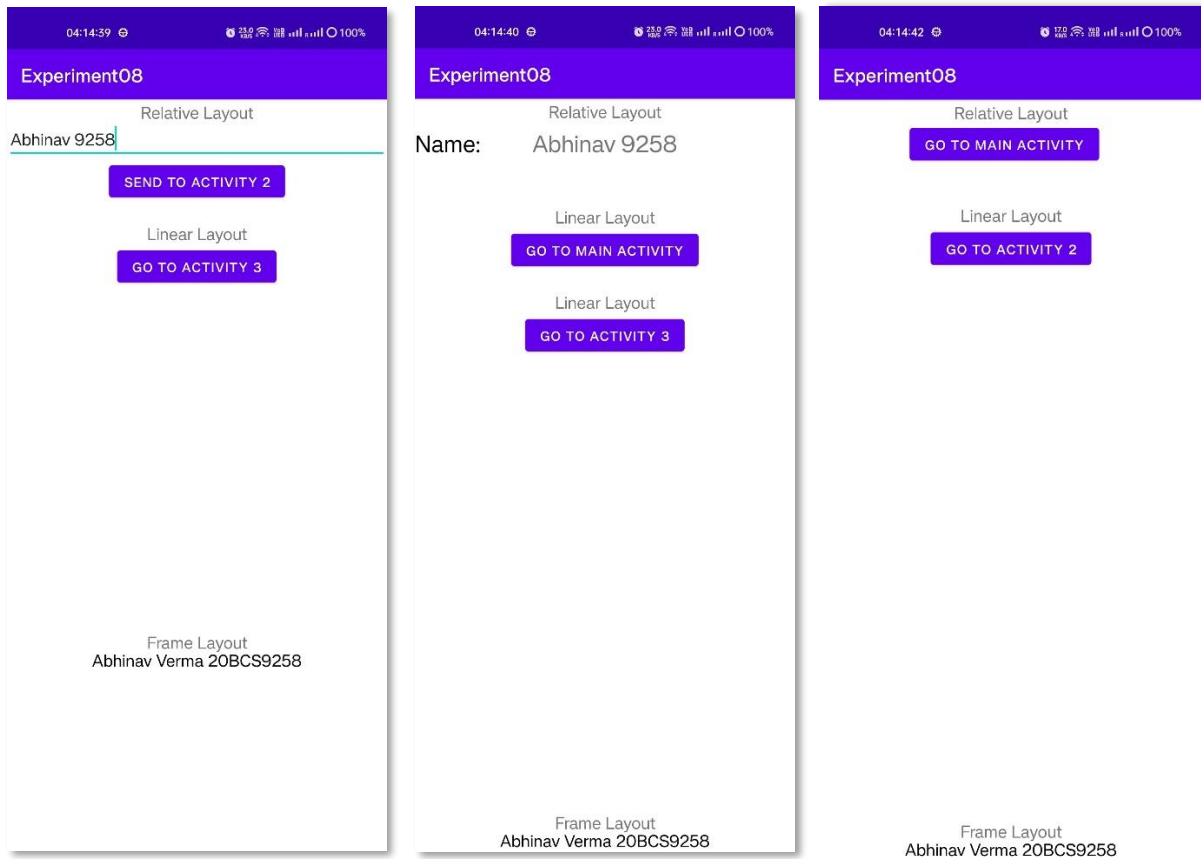
AndroidManifest.xml

Step 3: Output:

From above screen shots, we can see, we have implemented various different types of layouts and from first three snaps we can clearly see toast, displaying status of power supply, whether our device is connected to power or not.

Learning Outcomes:

1. Learned how to add toast, passing values via intent, textView, editText, layouts etc. in our application.
2. Learned about implementation of multiple layouts, android manifest, etc.
3. Learned to add buttons and textview, edittext, layouts.
4. Learned more about Android Studio and its components.