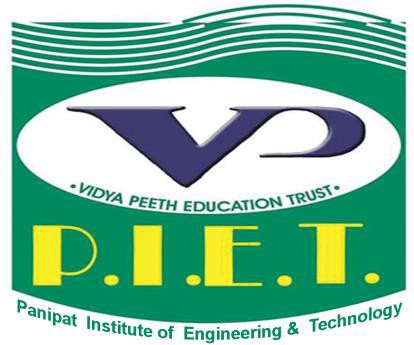
**Panipat Institute of Engineering & Technology, Samalkha, Panipat**



**Computer Science & Engineering Department**

Practical File of Mobile Apps Development Sub Code: CSE- 406N

Submitted To: Submitted By:

## Mr. Shiraz Khurana Deepak

Assistant Professor 2817205

(CSE Department)

**Affiliated to**

**Kurukshetra University Kurukshetra, India**

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. No.** | **Title of the Practical** | **Date** | **Page No.** | **Remarks** |
| 1. | Develop an application that uses GUI components, Font and colors. |  | 1-4 |  |
| 2. | Develop an application that uses Layout Managers and event listeners. |  | 5-10 |  |
| 3. | Develop a native calculator application. |  | 11-15 |  |
| 4. | Write an application that draws basic graphical primitives on the screen. |  | 16-18 |  |
| 5. | Develop a real-life application that makes use of databases. |  | 19-24 |  |
| 6. | Implement an application that implements multithreading. |  | 25-27 |  |
| 7. | Develop a native application that uses GPS location information. |  | 28-33 |  |
| 8. | Implement an application that creates an alert upon receiving a message. |  | 34-37 |  |
| 9. | Write a mobile application that creates alarm clock. |  | 38-42 |  |
| 10. | Implement an application that writes data to the SD card. |  | 43-46 |  |

# Practical No.-1

### Aim: Develop an application that uses GUI components, Fonts and colors.

**Hardware Requirements & Software Requirements:**

* Android Studio 4.2.1
* Windows 7/8/10 OS
* RAM 4 GB
* Hard disk 10 GB,
* Emulator/mobile phones etc

### Procedure:

* 1. Open android studio and select new android project.
  2. Give project name and select next.
  3. Choose the minimum target API version and select next.
  4. Enter the package name, and its domain. Package name must be of two words separated by comma and then click on finish.
  5. Go to package explorer in the left-hand side and select your project.
  6. Go to res folder and select layout. Double click the main.xml file.
  7. Now you can see the Graphics layout window.

## Source code:

### Front-end:

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

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

    android:orientation="vertical"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent">

    <TextView

        android:id="@+id/textView"

        android:layout\_width="match\_parent"

        android:layout\_height="wrap\_content"

        android:layout\_margin="30dp"

        android:gravity="center"

        android:text="Hello World!"

        android:textSize="25sp"

        android:textStyle="bold" />

    <Button

        android:id="@+id/button1"

        android:layout\_width="match\_parent"

        android:layout\_height="wrap\_content"

        android:layout\_margin="20dp"

        android:gravity="center"

        android:text="Change font size"

        android:textSize="25sp" />

    <Button

        android:id="@+id/button2"

        android:layout\_width="match\_parent"

        android:layout\_height="wrap\_content"

        android:layout\_margin="20dp"

        android:gravity="center"

        android:text="Change color"

        android:textSize="25sp" />

</LinearLayout>

### Back-end:

package com.example.exno1;

 import android.graphics.Color;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;

public class MainActivity extends AppCompatActivity

{

    int ch=1;

    float font=30;

    @Override

    protected void onCreate(Bundle savedInstanceState)

    {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_main);

        final TextView t= (TextView) findViewById(R.id.textView);

        Button b1= (Button) findViewById(R.id.button1);

        b1.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View v) {

                t.setTextSize(font);

                font = font + 5;

                if (font == 50)

                    font = 30;

            }

        });

        Button b2= (Button) findViewById(R.id.button2);

        b2.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View v) {

                switch (ch) {

                    case 1:

                        t.setTextColor(Color.RED);

                        break;

                    case 2:

                        t.setTextColor(Color.GREEN);

                        break;

                    case 3:

                        t.setTextColor(Color.BLUE);

                        break;

                    case 4:

                        t.setTextColor(Color.CYAN);

                        break;

                    case 5:

                        t.setTextColor(Color.YELLOW);

                        break;

                    case 6:

                        t.setTextColor(Color.MAGENTA);

                        break;

                }

                ch++;

                if (ch == 7)

                    ch = 1;

            }

        });

    }

}

**Output:**

# https://codingconnect.net/wp-content/uploads/2016/02/ex1-s4-e1456596139830.png

# Practical No.-2

### Aim: Develop an application that uses layout managers and event listeners.

**Hardware Requirements & Software Requirements:**

* Android Studio 4.2.1
* Windows 7/8/10 OS
* RAM 4 GB
* Hard disk 10 GB,
* Emulator/mobile phones etc;

### Procedure:

1. Open android studio and select new android project.
2. Give project name and select next.
3. Choose the minimum target API version and select next.
4. Enter the package name, and its domain. Package name must be of two words separated by comma and then click on finish.
5. Go to package explorer in the left-hand side and select your project.
6. Go to res folder and select layout. Double click the main.xml file.
7. Now you can see the Graphics layout window.

**Source code:**

### Front-end:

### activity\_main.xml

encoding="utf-8"?>

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

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="100dp">

<TextView

android:id="@+id/textView"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="30dp"

android:textColor="#99ffbb"

android:text="User Form"

android:textSize="35sp"

android:gravity="center"/>

</LinearLayout>

<GridLayout

android:id="@+id/gridLayout"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_marginTop="100dp"

android:layout\_marginBottom="200dp"

android:columnCount="2"

android:rowCount="3">

<TextView

android:id="@+id/textView1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_margin="10dp"

android:layout\_row="0"

android:layout\_column="0"

android:text="Name"

android:textColor="#ff9999"

android:textSize="20sp"

android:gravity="center"/>

<EditText

android:id="@+id/editText"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_margin="10dp"

android:layout\_row="0"

android:layout\_column="1"

android:ems="10"/>

<TextView

android:id="@+id/textView2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_margin="10dp"

android:layout\_row="1"

android:layout\_column="0"

android:text="Id No."

android:textColor="#ff9999"

android:textSize="20sp"

android:gravity="center"/>

<EditText

android:id="@+id/editText2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_margin="10dp"

android:layout\_row="1"

android:layout\_column="1"

android:inputType="number"

android:ems="10"/>

<TextView

android:id="@+id/textView3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_margin="10dp"

android:layout\_row="2"

android:layout\_column="0"

android:text="Department"

android:textColor="#ff9999"

android:textSize="20sp"

android:gravity="center"/>

<Spinner

android:id="@+id/spinner"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_margin="10dp"

android:layout\_row="2"

android:layout\_column="1"

android:textColor="#99ffbb"

android:spinnerMode="dropdown"/>

</GridLayout>

<Button

android:id="@+id/button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentBottom="true"

android:layout\_centerInParent="true"

android:layout\_marginBottom="250dp"

android:text="Submit"/>

</RelativeLayout>

**activity\_second.xml**

encoding="utf-8"?>

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

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context="com.example.devang.exno2.SecondActivity"

android:orientation="vertical"

android:gravity="center">

<TextView

android:id="@+id/textView1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_margin="20dp"

android:text="New Text"

android:textColor="#ff9999"

android:textSize="30sp"/>

<TextView

android:id="@+id/textView2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_margin="20dp"

android:text="New Text"

android:textColor="#ff9999"

android:textSize="30sp"/>

<TextView

android:id="@+id/textView3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_margin="20dp"

android:text="New Text"

android:textColor="#ff9999"

android:textSize="30sp"/>

</LinearLayout>

### Back end:

### MainActivity.java

package com.example.layouteventlistnerapp;

import android.content.Intent;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.view.View;

import android.widget.ArrayAdapter;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Spinner;

public class MainActivity extends AppCompatActivity {

//Defining the Views

EditText e1,e2;

Button bt;

Spinner s;

//Data for populating in Spinner

String [] dept\_array={"Sales","HR","IT","Testing","Developer"};

String name,reg,dept;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

//Referring the Views

e1= (EditText) findViewById(R.id.editText);

e2= (EditText) findViewById(R.id.editText2);

bt= (Button) findViewById(R.id.button);

s= (Spinner) findViewById(R.id.spinner);

//Creating Adapter for Spinner for adapting the data from array to Spinner

ArrayAdapter adapter= new ArrayAdapter(MainActivity.this,android.R.layout.simple\_spinner\_item,dept\_array);

s.setAdapter(adapter);

//Creating Listener for Button

bt.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

//Getting the Values from Views(Edittext & Spinner)

name=e1.getText().toString();

reg=e2.getText().toString();

dept=s.getSelectedItem().toString();

//Intent For Navigating to Second Activity

Intent i = new Intent(MainActivity.this,SecondActivity.class);

//For Passing the Values to Second Activity

i.putExtra("name\_key", name);

i.putExtra("reg\_key",reg);

i.putExtra("dept\_key", dept);

startActivity(i);

}

});

}

}

**secondActivity.java**

package com.example.layouteventlistnerapp;

import android.content.Intent;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.widget.TextView;

public class SecondActivity extends AppCompatActivity {

TextView t1,t2,t3;

String name,reg,dept;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_second);

t1= (TextView) findViewById(R.id.textView1);

t2= (TextView) findViewById(R.id.textView2);

t3= (TextView) findViewById(R.id.textView3);

//Getting the Intent

Intent i = getIntent();

//Getting the Values from First Activity using the Intent received

name=i.getStringExtra("name\_key");

reg=i.getStringExtra("reg\_key");

dept=i.getStringExtra("dept\_key");

//Setting the Values to Intent

t1.setText(name);

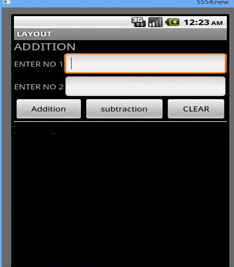
t2.setText(reg);

t3.setText(dept);

}

}

**Output:**



# Practical No.-3

### Aim: Develop a native calculator application.

**Hardware Requirements & Software Requirements:**

* Android Studio 4.2.1
* Windows 7/8/10 OS
* RAM 4 GB
* Hard disk 10 GB,
* Emulator/mobile phones etc;

### Procedure:

1. Open android studio and select new android project.
2. Give project name and select next.
3. Choose the minimum target API version and select next.
4. Enter the package name, and its domain. Package name must be of two words separated by comma and then click on finish.
5. Go to package explorer in the left-hand side and select your project.
6. Go to res folder and select layout. Double click the main.xml file.
7. Now you can see the Graphics layout window.

## Source code:

### Front-end:

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

<LinearLayout

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

    android:orientation="vertical"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    android:layout\_margin="20dp">

    <LinearLayout

        android:id="@+id/linearLayout1"

        android:layout\_width="match\_parent"

        android:layout\_height="wrap\_content"

        android:layout\_margin="20dp">

        <EditText

            android:id="@+id/editText1"

            android:layout\_width="match\_parent"

            android:layout\_height="wrap\_content"

            android:layout\_weight="1"

            android:inputType="numberDecimal"

            android:textSize="20sp" />

        <EditText

            android:id="@+id/editText2"

            android:layout\_width="match\_parent"

            android:layout\_height="wrap\_content"

            android:layout\_weight="1"

            android:inputType="numberDecimal"

            android:textSize="20sp" />

    </LinearLayout>

    <LinearLayout

        android:id="@+id/linearLayout2"

        android:layout\_width="match\_parent"

        android:layout\_height="wrap\_content"

        android:layout\_margin="20dp">

        <Button

            android:id="@+id/Add"

            android:layout\_width="match\_parent"

            android:layout\_height="wrap\_content"

            android:layout\_weight="1"

            android:text="+"

            android:textSize="30sp"/>

        <Button

            android:id="@+id/Sub"

            android:layout\_width="match\_parent"

            android:layout\_height="wrap\_content"

            android:layout\_weight="1"

            android:text="-"

            android:textSize="30sp"/>

        <Button

            android:id="@+id/Mul"

            android:layout\_width="match\_parent"

            android:layout\_height="wrap\_content"

            android:layout\_weight="1"

            android:text="\*"

            android:textSize="30sp"/>

        <Button

            android:id="@+id/Div"

            android:layout\_width="match\_parent"

            android:layout\_height="wrap\_content"

            android:layout\_weight="1"

            android:text="/"

            android:textSize="30sp"/>

    </LinearLayout>

    <TextView

        android:id="@+id/textView"

        android:layout\_width="match\_parent"

        android:layout\_height="wrap\_content"

        android:layout\_marginTop="50dp"

        android:text="Answer is"

        android:textSize="30sp"

        android:gravity="center"/>

</LinearLayout>

### Back end:

package com.example.exno3;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.text.TextUtils;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

public class MainActivity extends AppCompatActivity implements OnClickListener

{

    //Defining the Views

    EditText Num1;

    EditText Num2;

    Button Add;

    Button Sub;

    Button Mul;

    Button Div;

    TextView Result;

    @Override

    public void onCreate(Bundle savedInstanceState)

    {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_main);

        //Referring the Views

        Num1 = (EditText) findViewById(R.id.editText1);

        Num2 = (EditText) findViewById(R.id.editText2);

        Add = (Button) findViewById(R.id.Add);

        Sub = (Button) findViewById(R.id.Sub);

        Mul = (Button) findViewById(R.id.Mul);

        Div = (Button) findViewById(R.id.Div);

        Result = (TextView) findViewById(R.id.textView);

        // set a listener

        Add.setOnClickListener(this);

        Sub.setOnClickListener(this);

        Mul.setOnClickListener(this);

        Div.setOnClickListener(this);

    }

    @Override

    public void onClick (View v)

    {

        float num1 = 0;

        float num2 = 0;

        float result = 0;

        String oper = "";

        // check if the fields are empty

        if (TextUtils.isEmpty(Num1.getText().toString()) || TextUtils.isEmpty(Num2.getText().toString()))

                return;

        // read EditText and fill variables with numbers

        num1 = Float.parseFloat(Num1.getText().toString());

        num2 = Float.parseFloat(Num2.getText().toString());

        // defines the button that has been clicked and performs the corresponding operation

        // write operation into oper, we will use it later for output

        switch (v.getId())

        {

            case R.id.Add:

                oper = "+";

                result = num1 + num2;

                break;

            case R.id.Sub:

                oper = "-";

                result = num1 - num2;

                break;

            case R.id.Mul:

                oper = "\*";

                result = num1 \* num2;

                break;

            case R.id.Div:

                oper = "/";

                result = num1 / num2;

                break;

            default:

                break;

        }

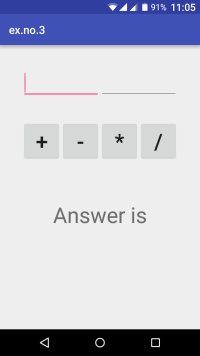
        // form the output line

        Result.setText(num1 + " " + oper + " " + num2 + " = " + result);

    }

}

**Output:-**



# Practical No.-4

### Aim: Write an application that draws graphical primitives on the screen.

**Hardware Requirements & Software Requirements:**

* Android Studio 4.2.1
* Windows 7/8/10 OS
* RAM 4 GB
* Hard disk 10 GB,
* Emulator/mobile phones etc;

### Procedure:

1. Open android studio and select new android project.
2. Give project name and select next.
3. Choose the minimum target API version and select next.
4. Enter the package name, and its domain. Package name must be of two words separated by comma and then click on finish.
5. Go to package explorer in the left-hand side and select your project.
6. Go to res folder and select layout. Double click the main.xml file.
7. Now you can see the Graphics layout window.

## Source code:

### Front-end:

<?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">

<ImageView

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:id="@+id/imageView" />

</RelativeLayout>

### Back end:

package com.example.graphicalprimitives;

import android.app.Activity;

import android.graphics.Bitmap;

import android.graphics.Canvas;

import android.graphics.Color;

import android.graphics.Paint;

import android.graphics.drawable.BitmapDrawable;

import android.os.Bundle;

import android.widget.ImageView;

public class MainActivity extends Activity

{

@Override

public void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

//Creating a Bitmap

Bitmap bg = Bitmap.createBitmap(720, 1280, Bitmap.Config.ARGB\_8888);

//Setting the Bitmap as background for the ImageView

ImageView i = (ImageView) findViewById(R.id.imageView);

i.setBackgroundDrawable(new BitmapDrawable(bg));

//Creating the Canvas Object

Canvas canvas = new Canvas(bg);

//Creating the Paint Object and set its color & TextSize

Paint paint = new Paint();

paint.setColor(Color.RED);

paint.setTextSize(50);

//To draw a Rectangle

canvas.drawText("Rectangle", 420, 150, paint);

canvas.drawRect(400, 200, 650, 700, paint);

Paint paint1 = new Paint();

paint1.setColor(Color.BLACK);

paint1.setTextSize(50);

//To draw a Circle

canvas.drawText("Circle", 120, 150, paint1);

canvas.drawCircle(200, 350, 150, paint1);

Paint paint2 = new Paint();

paint2.setColor(Color.LTGRAY);

paint2.setTextSize(50);

//To draw a Square

canvas.drawText("Square", 120, 800, paint2);

canvas.drawRect(50, 850, 350, 1150, paint2);

Paint paint3 = new Paint();

paint3.setColor(Color.CYAN);

paint3.setTextSize(50);

//To draw a Line

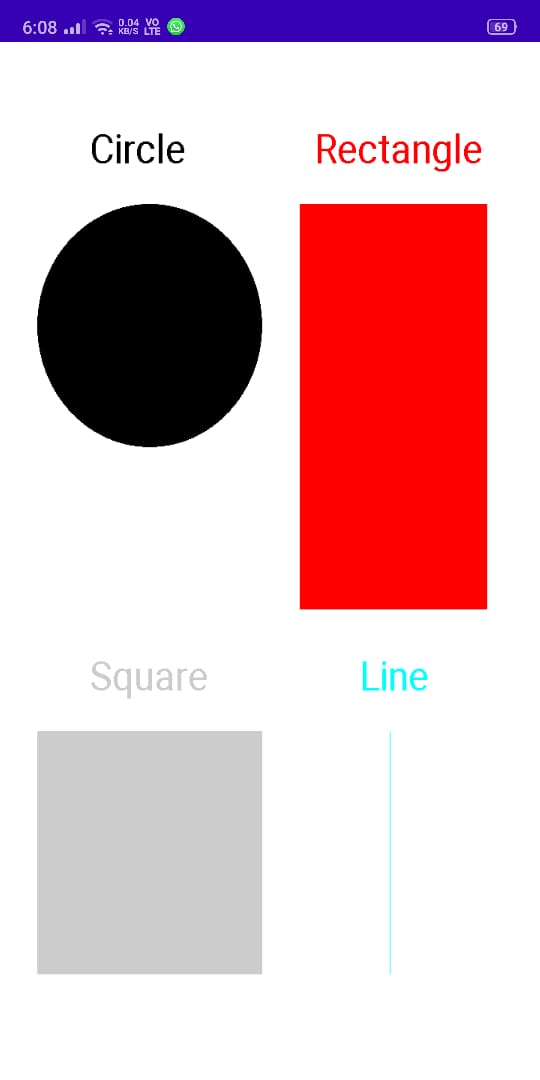
canvas.drawText("Line", 480, 800, paint3);

canvas.drawLine(520, 850, 520, 1150, paint3);

}

}

**Output:**

****

# Practical No.-5

### Aim: Develop a real-life application that makes use of database.

**Hardware Requirements & Software Requirements:**

* Android Studio 4.2.1
* Windows 7/8/10 OS
* RAM 4 GB
* Hard disk 10 GB,
* Emulator/mobile phones etc;

### Procedure:

1. Open android studio and select new android project.
2. Give project name and select next.
3. Choose the minimum target API version and select next.
4. Enter the package name, and its domain. Package name must be of two words separated by comma and then click on finish.
5. Go to package explorer in the left-hand side and select your project.
6. Go to res folder and select layout. Double click the main.xml file.
7. Now you can see the Graphics layout window.

## Source code:

### Front-end:

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

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_x="50dp"

android:layout\_y="30dp"

android:text="DataBase Application"

android:textColor="#000"

android:textSize="30sp" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_x="20dp"

android:layout\_y="110dp"

android:text="Enter Id:"

android:textColor="#ff3333"

android:textSize="20sp" />

<EditText

android:id="@+id/Id"

android:layout\_width="150dp"

android:layout\_height="wrap\_content"

android:layout\_x="175dp"

android:layout\_y="100dp"

android:inputType="number"

android:textSize="20sp" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_x="20dp"

android:layout\_y="160dp"

android:text="Enter Name:"

android:textColor="#ff3333"

android:textSize="20sp" />

<EditText

android:id="@+id/Name"

android:layout\_width="150dp"

android:layout\_height="wrap\_content"

android:layout\_x="175dp"

android:layout\_y="150dp"

android:inputType="text"

android:textSize="20sp" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_x="20dp"

android:layout\_y="210dp"

android:text="Phone Number:"

android:textColor="#ff3333"

android:textSize="20sp" />

<EditText

android:id="@+id/Number"

android:layout\_width="150dp"

android:layout\_height="wrap\_content"

android:layout\_x="175dp"

android:layout\_y="200dp"

android:inputType="number"

android:textSize="20sp" />

<Button

android:id="@+id/Insert"

android:layout\_width="150dp"

android:layout\_height="wrap\_content"

android:layout\_x="25dp"

android:layout\_y="300dp"

android:text="Insert"

android:textSize="25dp" />

<Button

android:id="@+id/Delete"

android:layout\_width="150dp"

android:layout\_height="wrap\_content"

android:layout\_x="200dp"

android:layout\_y="300dp"

android:text="Delete"

android:textSize="25dp" />

<Button

android:id="@+id/Update"

android:layout\_width="150dp"

android:layout\_height="wrap\_content"

android:layout\_x="25dp"

android:layout\_y="400dp"

android:text="Update"

android:textSize="25dp" />

<Button

android:id="@+id/View"

android:layout\_width="150dp"

android:layout\_height="wrap\_content"

android:layout\_x="200dp"

android:layout\_y="400dp"

android:text="View"

android:textSize="25dp" />

</AbsoluteLayout>

### Back end:

package com.example.databaseapplication;

import android.app.Activity;

import android.app.AlertDialog.Builder;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.os.Bundle;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.EditText;

public class MainActivity extends Activity implements OnClickListener

{

EditText Id,Name,Number;

Button Insert,Delete,Update,View;

SQLiteDatabase db;

/\*\* Called when the activity is first created. \*/

@Override

public void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

Id=(EditText)findViewById(R.id.Id);

Name=(EditText)findViewById(R.id.Name);

Number=(EditText)findViewById(R.id.Number);

Insert=(Button)findViewById(R.id.Insert);

Delete=(Button)findViewById(R.id.Delete);

Update=(Button)findViewById(R.id.Update);

View=(Button)findViewById(R.id.View);

Insert.setOnClickListener(this);

Delete.setOnClickListener(this);

Update.setOnClickListener(this);

View.setOnClickListener(this);

// Creating database and table

db=openOrCreateDatabase("StudentDB", Context.MODE\_PRIVATE, null);

db.execSQL("CREATE TABLE IF NOT EXISTS student(rollno VARCHAR,name VARCHAR,marks VARCHAR);");

}

public void onClick(View view)

{

// Inserting a record to the Student table

if(view==Insert)

{

// Checking for empty fields

if(Id.getText().toString().trim().length()==0||

Name.getText().toString().trim().length()==0||

Number.getText().toString().trim().length()==0)

{

showMessage("Error", "Please enter all values");

return;

}

db.execSQL("INSERT INTO student VALUES('"+Id.getText()+"','"+Name.getText()+

"','"+Number.getText()+"');");

showMessage("Success", "Record added");

clearText();

}

// Deleting a record from the Student table

if(view==Delete)

{

// Checking for empty roll number

if(Id.getText().toString().trim().length()==0)

{

showMessage("Error", "Please enter Rollno");

return;

}

Cursor c=db.rawQuery("SELECT \* FROM student WHERE rollno='"+Id.getText()+"'", null);

if(c.moveToFirst())

{

db.execSQL("DELETE FROM student WHERE rollno='"+Id.getText()+"'");

showMessage("Success", "Record Deleted");

}

else

{

showMessage("Error", "Invalid Rollno");

}

clearText();

}

// Updating a record in the Student table

if(view==Update)

{

// Checking for empty roll number

if(Id.getText().toString().trim().length()==0)

{

showMessage("Error", "Please enter Rollno");

return;

}

Cursor c=db.rawQuery("SELECT \* FROM student WHERE rollno='"+Id.getText()+"'", null);

if(c.moveToFirst()) {

db.execSQL("UPDATE student SET name='" + Name.getText() + "',marks='" + Number.getText() +

"' WHERE rollno='"+Id.getText()+"'");

showMessage("Success", "Record Modified");

}

else {

showMessage("Error", "Invalid Rollno");

}

clearText();

}

// Display a record from the Student table

if(view==View)

{

// Checking for empty roll number

if(Id.getText().toString().trim().length()==0)

{

showMessage("Error", "Please enter Rollno");

return;

}

Cursor c=db.rawQuery("SELECT \* FROM student WHERE rollno='"+Id.getText()+"'", null);

if(c.moveToFirst())

{

Name.setText(c.getString(1));

Number.setText(c.getString(2));

}

else

{

showMessage("Error", "Invalid Rollno");

clearText();

}

}

// Displaying all the records

}

public void showMessage(String title,String message)

{

Builder builder=new Builder(this);

builder.setCancelable(true);

builder.setTitle(title);

builder.setMessage(message);

builder.show();

}

public void clearText()

{

Id.setText("");

Name.setText("");

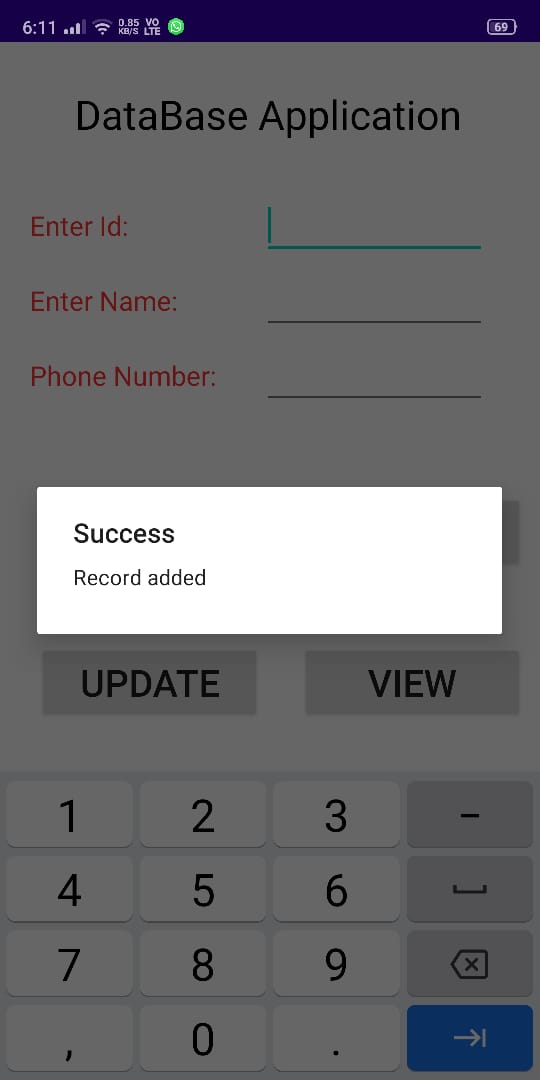
Number.setText("");

Id.requestFocus();

}

}

**Output**:



# Practical No.-6

### Aim: Implement an application that implements Multithreading.

**Hardware Requirements & Software Requirements:**

* Android Studio 4.2.1
* Windows 7/8/10 OS
* RAM 4 GB
* Hard disk 10 GB,
* Emulator/mobile phones etc;

### Procedure:

1. Open android studio and select new android project.
2. Give project name and select next.
3. Choose the minimum target API version and select next.
4. Enter the package name, and its domain. Package name must be of two words separated by comma and then click on finish.
5. Go to package explorer in the left-hand side and select your project.
6. Go to res folder and select layout. Double click the main.xml file.
7. Now you can see the Graphics layout window.

## Source code:

### Front-end:

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

<RelativeLayout 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">

<ProgressBar

android:id="@+id/ProgressBar1"

style="?android:attr/progressBarStyleHorizontal"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:layout\_alignParentRight="true"

android:layout\_alignParentTop="true" />

<TextView

android:id="@+id/TextView1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@id/ProgressBar1"

android:layout\_centerHorizontal="true"

android:text="" />

<Button

android:id="@+id/Button1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@id/TextView1"

android:layout\_centerHorizontal="true"

android:text="Start Progress" />

</RelativeLayout>

### Backend:

package com.example.multithreading;

import androidx.appcompat.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 {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

final ProgressBar p = (ProgressBar) findViewById(R.id.ProgressBar1);

final TextView t = (TextView) findViewById(R.id.TextView1);

Button b = (Button) findViewById(R.id.Button1);

b.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

Runnable r = new Runnable() {

@Override

public void run() {

for (int i=0; i<= 100; i++)

{

final int temp = i;

try {

Thread.sleep(500);

}

catch(InterruptedException e){

e.printStackTrace();

}

p.post(new Runnable() {

@Override

public void run() {

p.setProgress(temp);

t.setText(temp + "%");

}

});

}

}

};

new Thread(r).start();

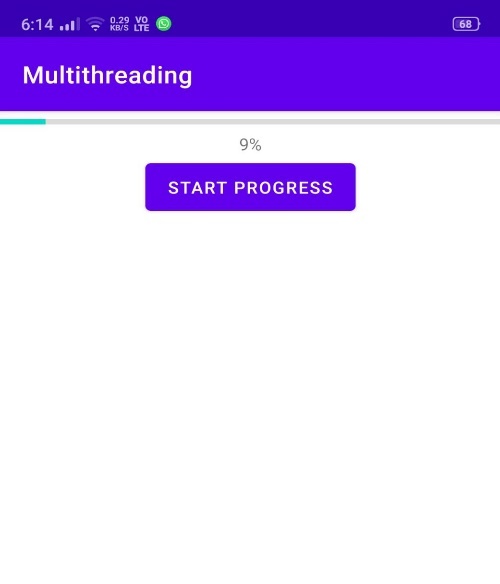
}

});

}

}

**Output:**

****

# Practical No.-7

### Aim: Develop a native application that uses GPS location information.

**Hardware Requirements & Software Requirements:**

* Android Studio 4.2.1
* Windows 7/8/10 OS
* RAM 4 GB
* Hard disk 10 GB,
* Emulator/mobile phones

### Procedure:

1. Open android studio and select new android project.
2. Give project name and select next.
3. Choose the minimum target API version and select next.
4. Enter the package name, and its domain. Package name must be of two words separated by comma and then click on finish.
5. Go to package explorer in the left-hand side and select your project.
6. Go to res folder and select layout. Double click the main.xml file.
7. Now you can see the Graphics layout window.

## Source code:

### Front-end:

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

<RelativeLayout

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

android:id="@+id/relativeLayout1"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent" >

<Button

android:id="@+id/show\_Location"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Show\_Location"

android:layout\_centerVertical="true"

android:layout\_centerHorizontal="true" />

</RelativeLayout>

### Backend:

### GPStrace.java

package com.example.locationapp;

import android.app.AlertDialog;

import android.app.Service;

import android.content.Context;

import android.content.DialogInterface;

import android.content.Intent;

import android.location.Location;

import android.location.LocationListener;

import android.location.LocationManager;31

import android.os.Bundle;

import android.os.IBinder;

import android.provider.Settings;

public class GPStrace extends Service implements

LocationListener{

private final Context context;

boolean isGPSEnabled=false;

boolean canGetLocation=false;

boolean isNetworkEnabled=false;

Location location;

double latitude;

double longtitude;

private static final long MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES=10;

private static final long MIN\_TIME\_BW\_UPDATES=1000\*60\*1;

protected LocationManager locationManager;

public GPStrace(Context context)

{

this.context=context;

getLocation();

}

public Location getLocation()

{

try{

locationManager=(LocationManager)

context.getSystemService(LOCATION\_SERVICE);

isGPSEnabled=locationManager.isProviderEnabled(LocationManager.G

PS\_PROVIDER);

isNetworkEnabled=locationManager.isProviderEnabled(LocationManag

er.NETWORK\_PROVI

DER);

if(!isGPSEnabled && !isNetworkEnabled){

}else{

this.canGetLocation=true;

if(isNetworkEnabled){

locationManager.requestLocationUpdates(

LocationManager.NETWORK\_PROVIDER,

MIN\_TIME\_BW\_UPDATES,

MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES,this);

}

if(locationManager!=null){

location=locationManager.getLastKnownLocation(LocationManager.NE

TWORK\_PROVIDER)

;

if(location !=null){

latitude=location.getLatitude();

longtitude=location.getLongitude();

}}}32

if(isGPSEnabled){

if(location==null){

locationManager.requestLocationUpdates(LocationManager.GPS\_PROVI

DER,MIN\_TIME\_B

W\_UPDATES, MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES, this);

if(locationManager!=null){

location=locationManager.getLastKnownLocation(LocationManager.GP

S\_PROVIDER);

if(location!=null){

latitude=location.getLatitude();

longtitude=location.getLongitude();

}}}}}

catch(Exception e)

{

e.printStackTrace();

}

return location;

}

public void stopUsingGPS(){

if(locationManager!=null){

locationManager.removeUpdates(GPStrace.this);

}}

public double getLatitude(){

if(location!=null){

latitude=location.getLatitude();

}

return latitude;

}

public double getLongtiude(){

if(location!=null){

longtitude=location.getLatitude();

}

return longtitude;

}

public boolean canGetLocation(){

return this.canGetLocation;

}

public void showSettingAlert(){

AlertDialog.Builder alertDialog=new

AlertDialog.Builder(context);

alertDialog.setTitle("GPS is settings");

alertDialog.setMessage("GPS is not enabled.Do you want to go to

setting menu?");

alertDialog.setPositiveButton("settings", new

DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog,int which){33

Intent intent=new

Intent(Settings.ACTION\_LOCATION\_SOURCE\_SETTINGS);

context.startActivity(intent);

}});

alertDialog.setNegativeButton("cancel", new

DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

// TODO Auto-generated method stub

dialog.cancel();

}

});

alertDialog.show();

}

@Override

public void onLocationChanged(Location location) {

// TODO Auto-generated method stub

}

@Override

public void onProviderDisabled(String provider) {

// TODO Auto-generated method stub

}

@Override

public void onProviderEnabled(String provider) {

// TODO Auto-generated method stub

}

@Override

public void onStatusChanged(String provider, int status, Bundle

extras) {

// TODO Auto-generated method stub

}

@Override

public IBinder onBind(Intent intent) {

// TODO Auto-generated method stub

return null;

}}

### ActivityMain.java

package com.example.locationapp;

//import android.R;

import android.app.Activity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.Toast;

public class GPSlocationActivity extends Activity {

/\*\* Called when the activity is first created. \*/

Button btnShowLocation;

GPStrace gps;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

btnShowLocation=(Button)findViewById(R.id.show\_Location);

btnShowLocation.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

gps=new GPStrace(GPSlocationActivity.this);

if(gps.canGetLocation()){

double latitude=gps.getLatitude();

double longitude=gps.getLongtiude();

Toast.makeText(getApplicationContext(),"Your Location is

\nLat:"+latitude+"\nLong:"+longitude, Toast.LENGTH\_LONG).show();

}

else

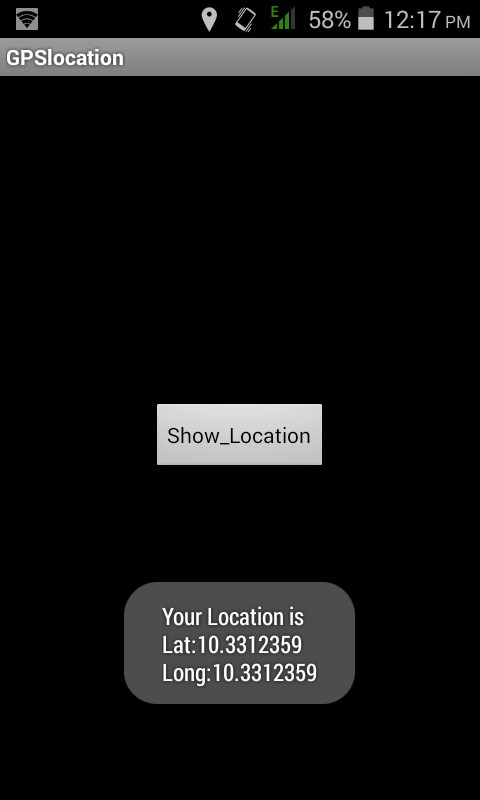
{

gps.showSettingAlert();

}

} }); } })

* + Now go to main.xml and right click.select run as option and select run configuration
  + Android output is present in the android emulator as shown in below.

**Output:**

# Practical No.-8

### Aim: Implement an application that creates an alert upon receiving a message.

**Hardware Requirements & Software Requirements:**

* Android Studio 4.2.1
* Windows 7/8/10 OS
* RAM 4 GB
* Hard disk 10 GB,
* Emulator/mobile phones etc;

### Procedure:

1. Open android studio and select new android project.
2. Give project name and select next.
3. Choose the minimum target API version and select next.
4. Enter the package name, and its domain. Package name must be of two words separated by comma and then click on finish.
5. Go to package explorer in the left-hand side and select your project.
6. Go to res folder and select layout. Double click the main.xml file.
7. Now you can see the Graphics layout window.

## Source code:

### Front-end:

### activity\_main.xml

<?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:layout\_margin="10dp"

android:orientation="vertical">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Message"

android:textSize="30sp" />

<EditText

android:id="@+id/editText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:singleLine="true"

android:textSize="30sp" />

<Button

android:id="@+id/button"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_margin="30dp"

android:layout\_gravity="center"

android:text="Notify"

android:textSize="30sp"/>

</LinearLayout>

**activity\_second.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=".SecondActivity">

</androidx.constraintlayout.widget.ConstraintLayout>

**Backend:**

**MainActvity.java**

package com.example.alertapp;

import android.app.Notification;

import android.app.NotificationManager;

import android.app.PendingIntent;

import android.content.Intent;

import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

public class MainActivity extends AppCompatActivity

{

Button notify;

EditText e;

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

notify= (Button) findViewById(R.id.button);

e= (EditText) findViewById(R.id.editText);

notify.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

Intent intent = new Intent(MainActivity.this, SecondActivity.class);

PendingIntent pending = PendingIntent.getActivity(MainActivity.this, 0, intent, 0);

Notification noti = new Notification.Builder(MainActivity.this).setContentTitle("New Message").setContentText(e.getText().toString()).setSmallIcon(R.mipmap.ic\_launcher).setContentIntent(pending).build();

NotificationManager manager = (NotificationManager) getSystemService(NOTIFICATION\_SERVICE);

noti.flags |= Notification.FLAG\_AUTO\_CANCEL;

manager.notify(0, noti);

}

});

}

}

package com.example.alertapp;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class SecondActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

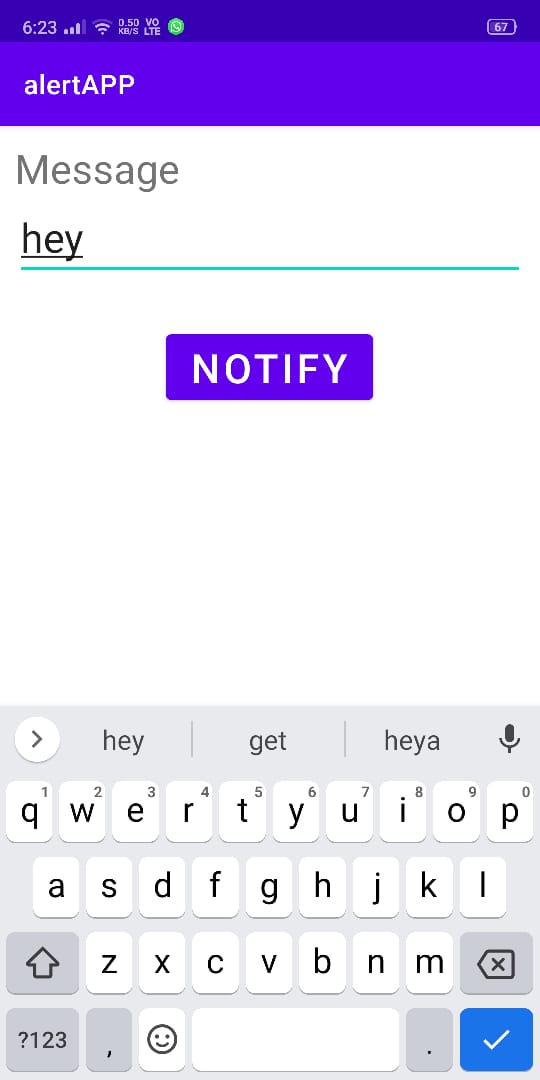
super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_second);

}

}

**Output:**

****

# Practical No.-9

### Aim: Write a mobile application that creates alarm clock.

**Hardware Requirements & Software Requirements:**

* Android Studio 4.2.1
* Windows 7/8/10 OS
* RAM 4 GB
* Hard disk 10 GB,
* Emulator/mobile phones etc

### Procedure:

1. Open android studio and select new android project.
2. Give project name and select next.
3. Choose the minimum target API version and select next.
4. Enter the package name, and its domain. Package name must be of two words separated by comma and then click on finish.
5. Go to package explorer in the left-hand side and select your project.
6. Go to res folder and select layout. Double click the main.xml file.
7. Now you can see the Graphics layout window.

## Source code:

### Front-end:

### activity\_main.xml

<?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:orientation="vertical">

<ToggleButton

android:id="@+id/toggleButton"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:layout\_margin="20dp"

android:checked="false"

android:onClick="OnToggleClicked" />

<TimePicker

android:id="@+id/timePicker"

android:layout\_width="wrap\_content"

android:amPmBackgroundColor="#4dff88"

android:numbersBackgroundColor="#ffbf80"

android:layout\_height="wrap\_content"

android:layout\_gravity="center" />

</LinearLayout>

**activity\_receiver.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=".AlarmReceiver">

</androidx.constraintlayout.widget.ConstraintLayout>

## Backend:

## MainActivity.java

package com.example.alarmclock;

import android.app.AlarmManager;

import android.app.PendingIntent;

import android.content.Intent;

import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;

import android.view.View;

import android.widget.TimePicker;

import android.widget.Toast;

import android.widget.ToggleButton;

import java.util.Calendar;

public class MainActivity extends AppCompatActivity

{

TimePicker alarmTimePicker;

PendingIntent pendingIntent;

AlarmManager alarmManager;

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

alarmTimePicker = (TimePicker) findViewById(R.id.timePicker);

alarmManager = (AlarmManager) getSystemService(ALARM\_SERVICE);

}

public void OnToggleClicked(View view)

{

long time;

if (((ToggleButton) view).isChecked())

{

Toast.makeText(MainActivity.this, "ALARM ON", Toast.LENGTH\_SHORT).show();

Calendar calendar = Calendar.getInstance();

calendar.set(Calendar.HOUR\_OF\_DAY, alarmTimePicker.getCurrentHour());

calendar.set(Calendar.MINUTE, alarmTimePicker.getCurrentMinute());

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

pendingIntent = PendingIntent.getBroadcast(this, 0, intent, 0);

time=(calendar.getTimeInMillis()-(calendar.getTimeInMillis()%60000));

if(System.currentTimeMillis()>time)

{

if (calendar.AM\_PM == 0)

time = time + (1000\*60\*60\*12);

else

time = time + (1000\*60\*60\*24);

}

alarmManager.setRepeating(AlarmManager.RTC\_WAKEUP, time, 10000, pendingIntent);

}

else

{

alarmManager.cancel(pendingIntent);

Toast.makeText(MainActivity.this, "ALARM OFF", Toast.LENGTH\_SHORT).show();

}

}

}

**AlarmReceiver.java**

package com.example.alarmclock;

import android.content.BroadcastReceiver;

import android.content.Context;

import android.content.Intent;

import android.media.Ringtone;

import android.media.RingtoneManager;

import android.net.Uri;

import android.widget.Toast;

public class AlarmReceiver extends BroadcastReceiver

{

@Override

public void onReceive(Context context, Intent intent)

{

Toast.makeText(context, "Alarm! Wake up! Wake up!", Toast.LENGTH\_LONG).show();

Uri alarmUri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE\_ALARM);

if (alarmUri == null)

{

alarmUri = RingtoneManager.getDefaultUri(RingtoneManager.TYPE\_NOTIFICATION);

}

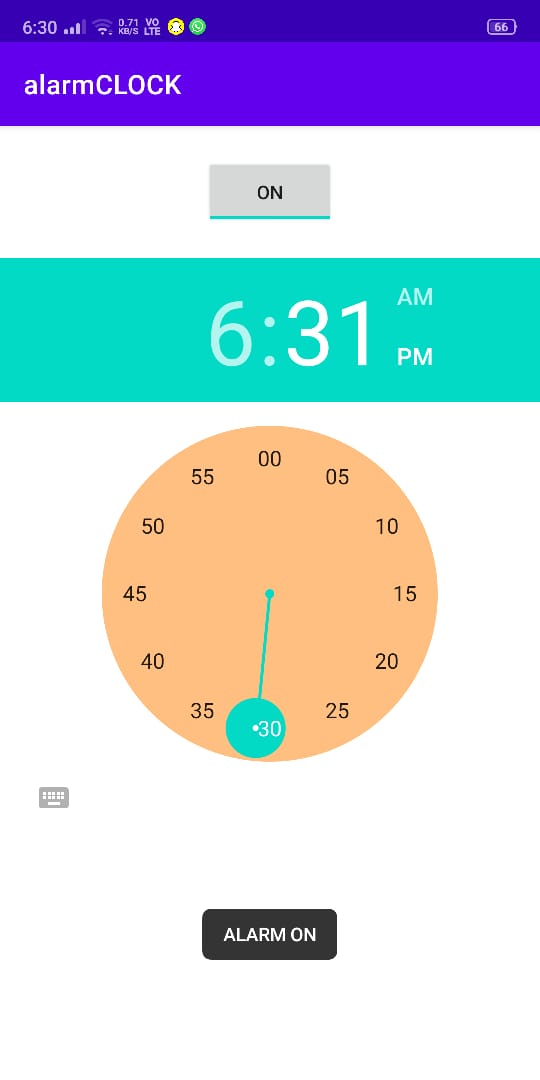
Ringtone ringtone = RingtoneManager.getRingtone(context, alarmUri);

ringtone.play();

}

}

**Output:**



# Practical No.-10

### Aim: Implement an application that writes data to the SD card.

**Hardware Requirements & Software Requirements:**

* Android Studio 4.2.1
* Windows 7/8/10 OS
* RAM 4 GB
* Hard disk 10 GB,
* Emulator/mobile phones etc;

### Procedure:

1. Open android studio and select new android project.
2. Give project name and select next.
3. Choose the minimum target API version and select next.
4. Enter the package name, and its domain. Package name must be of two words separated by comma and then click on finish.
5. Go to package explorer in the left-hand side and select your project.
6. Go to res folder and select layout. Double click the main.xml file.
7. Now you can see the Graphics layout window.

## Source code:

### Front-end:

<?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:layout\_margin="20dp"

android:orientation="vertical">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="SD Card Writer"

android:textColor="#000"

android:textSize="30sp" />

<EditText

android:id="@+id/editText"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:singleLine="true"

android:textSize="30dp" />

<Button

android:id="@+id/button"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="10dp"

android:text="Write Data"

android:textSize="30dp" />

<Button

android:id="@+id/button2"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="10dp"

android:text="Read data"

android:textSize="30dp" />

<Button

android:id="@+id/button3"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="10dp"

android:text="Clear"

android:textSize="30dp" />

</LinearLayout>

## Backend:

package com.example.sdcard;

import android.os.Bundle;

import androidx.appcompat.app.AppCompatActivity;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.Toast;

import java.io.BufferedReader;

import java.io.File;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.io.InputStreamReader;

public class MainActivity extends AppCompatActivity

{

EditText e1;

Button write,read,clear;

@Override

protected void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

e1= (EditText) findViewById(R.id.editText);

write= (Button) findViewById(R.id.button);

read= (Button) findViewById(R.id.button2);

clear= (Button) findViewById(R.id.button3);

write.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

String message=e1.getText().toString();

try

{

File f=new File("/sdcard/myfile.txt");

f.createNewFile();

FileOutputStream fout=new FileOutputStream(f);

fout.write(message.getBytes());

fout.close();

Toast.makeText(getBaseContext(),"Data Written in SDCARD",Toast.LENGTH\_LONG).show();

}

catch (Exception e)

{

Toast.makeText(getBaseContext(),e.getMessage(),Toast.LENGTH\_LONG).show();

}

}

});

read.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

String message;

String buf = "";

try

{

File f = new File("/sdcard/myfile.txt");

FileInputStream fin = new FileInputStream(f);

BufferedReader br = new BufferedReader(new InputStreamReader(fin));

while ((message = br.readLine()) != null)

{

buf += message;

}

e1.setText(buf);

br.close();

fin.close();

Toast.makeText(getBaseContext(),"Data Recived from SDCARD",Toast.LENGTH\_LONG).show();

}

catch (Exception e)

{

Toast.makeText(getBaseContext(), e.getMessage(), Toast.LENGTH\_LONG).show();

}

}

});

clear.setOnClickListener(new View.OnClickListener()

{

@Override

public void onClick(View v)

{

e1.setText("");

}

});

}

}

**Output:**



**Github Link for Experiments :**

https://github.com/Anuj\_khokhar/MAD\_Practicals.git