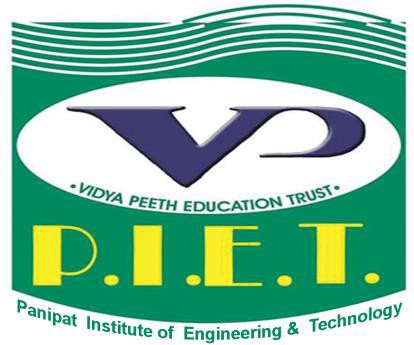
**Panipat Institute of Engineering & Technology, Samalkha, Panipat**



**Computer Science & Engineering Department**

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

Submitted To: Submitted By:

## Mr. Shiraz Khurana Bijay Mandal

Assistant Professor 2817208

(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. |  |  |  |
| 2. | Develop an application that uses Layout Managers and event listeners. |  |  |  |
| 3. | Develop a native calculator application. |  |  |  |
| 4. | Write an application that draws basic graphical primitives on the screen. |  |  |  |
| 5. | Develop a real-life application that makes use of databases. |  |  |  |
| 6. | Implement an application that implements multithreading. |  |  |  |
| 7. | Develop a native application that uses GPS location information. |  |  |  |
| 8. | Implement an application that creates an alert upon receiving a message. |  |  |  |
| 9. | Write a mobile application that creates alarm clock. |  |  |  |
| 10. | Implement an application that writes data to the SD card. |  |  |  |

# 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:**



# 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

<?xml version="1.0" 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:text="Details Form"

            android:textSize="25sp"

            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: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="Reg.No"

            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="Dept"

            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: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="150dp"

        android:text="Submit"/>

</RelativeLayout>

**Activity\_second.xml**

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

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

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

    android: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: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: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:textSize="30sp"/>

</LinearLayout>

### Back end:

### Main\_activity.java

package com.example.exno2;

import android.content.Intent;

import android.support.v7.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={"CSE","ECE","IT","Mech","Civil"};

    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.exno2;

import android.content.Intent;

import android.support.v7.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);

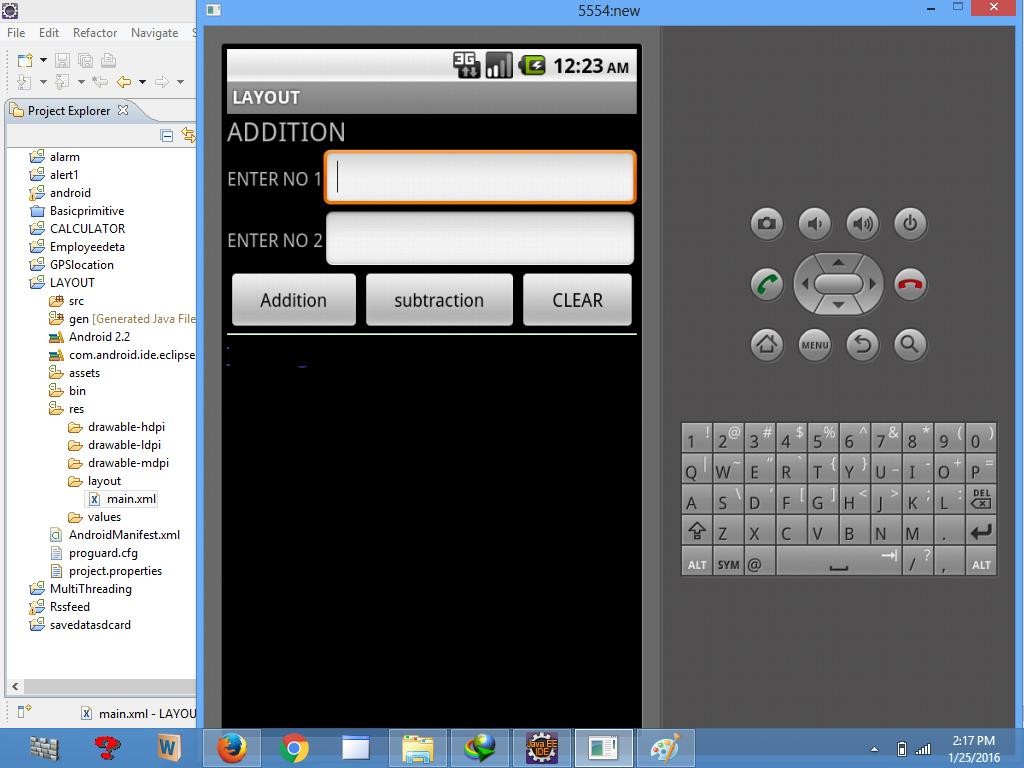
        t2.setText(reg);

        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;

        }

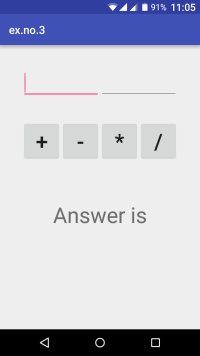
        // form the output line

        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.exno4;

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.BLUE);

        paint.setTextSize(50);

        //To draw a Rectangle

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

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

        //To draw a Circle

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

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

        //To draw a Square

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

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

        //To draw a Line

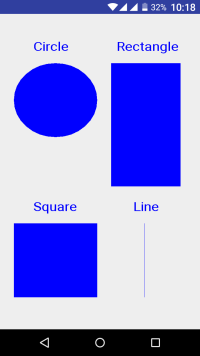
        canvas.drawText("Line", 480, 800, paint);

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

    }

}

**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="20dp"

        android:text="Student Details"

        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 Rollno:"

        android:textSize="20sp" />

    <EditText

        android:id="@+id/Rollno"

        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: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="Enter Marks:"

        android:textSize="20sp" />

    <EditText

        android:id="@+id/Marks"

        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="30dp" />

    <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="30dp" />

    <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="30dp" />

    <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="30dp" />

    <Button

        android:id="@+id/ViewAll"

        android:layout\_width="200dp"

        android:layout\_height="wrap\_content"

        android:layout\_x="100dp"

        android:layout\_y="500dp"

        android:text="View All"

        android:textSize="30dp" />

</AbsoluteLayout>

### Back end:

package com.example.exno5;

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 Rollno,Name,Marks;

    Button Insert,Delete,Update,View,ViewAll;

    SQLiteDatabase db;

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

    @Override

    public void onCreate(Bundle savedInstanceState)

    {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_main);

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

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

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

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

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

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

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

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

        Insert.setOnClickListener(this);

        Delete.setOnClickListener(this);

        Update.setOnClickListener(this);

        View.setOnClickListener(this);

        ViewAll.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(Rollno.getText().toString().trim().length()==0||

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

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

            {

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

                return;

            }

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

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

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

            clearText();

        }

        // Deleting a record from the Student table

        if(view==Delete)

        {

            // Checking for empty roll number

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

            {

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

                return;

            }

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

            if(c.moveToFirst())

            {

                db.execSQL("DELETE FROM student WHERE rollno='"+Rollno.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(Rollno.getText().toString().trim().length()==0)

            {

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

                return;

            }

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

            if(c.moveToFirst()) {

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

                        "' WHERE rollno='"+Rollno.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(Rollno.getText().toString().trim().length()==0)

            {

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

                return;

            }

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

            if(c.moveToFirst())

            {

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

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

            }

            else

            {

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

                clearText();

            }

        }

        // Displaying all the records

        if(view==ViewAll)

        {

            Cursor c=db.rawQuery("SELECT \* FROM student", null);

            if(c.getCount()==0)

            {

                showMessage("Error", "No records found");

                return;

            }

            StringBuffer buffer=new StringBuffer();

            while(c.moveToNext())

            {

                buffer.append("Rollno: "+c.getString(0)+"\n");

                buffer.append("Name: "+c.getString(1)+"\n");

                buffer.append("Marks: "+c.getString(2)+"\n\n");

            }

            showMessage("Student Details", buffer.toString());

        }

    }

    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()

    {

        Rollno.setText("");

        Name.setText("");

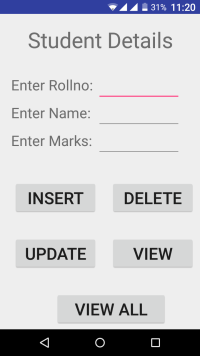
        Marks.setText("");

        Rollno.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();

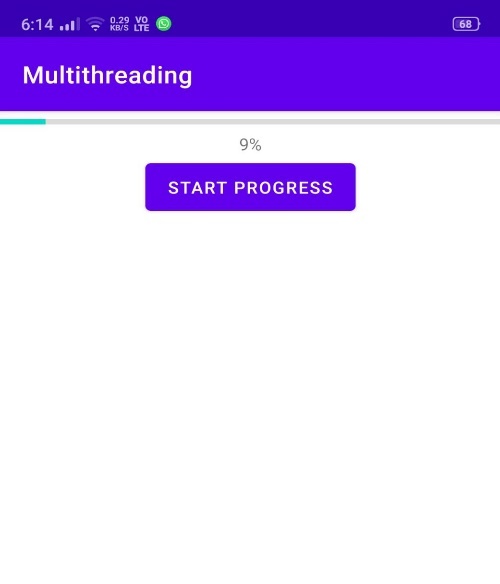
}

});

}

}

**Output:**

****

**Practical No.-7**

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

**Hardware Requirements & Software Requirements:**

• Android Studio3.0

• Windows 7/8/10 OS

• RAM4GB

• 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: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:**

package gps.location;

//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();

}

}}); }})

package gps.location;

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;

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(); }}}

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){

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) {

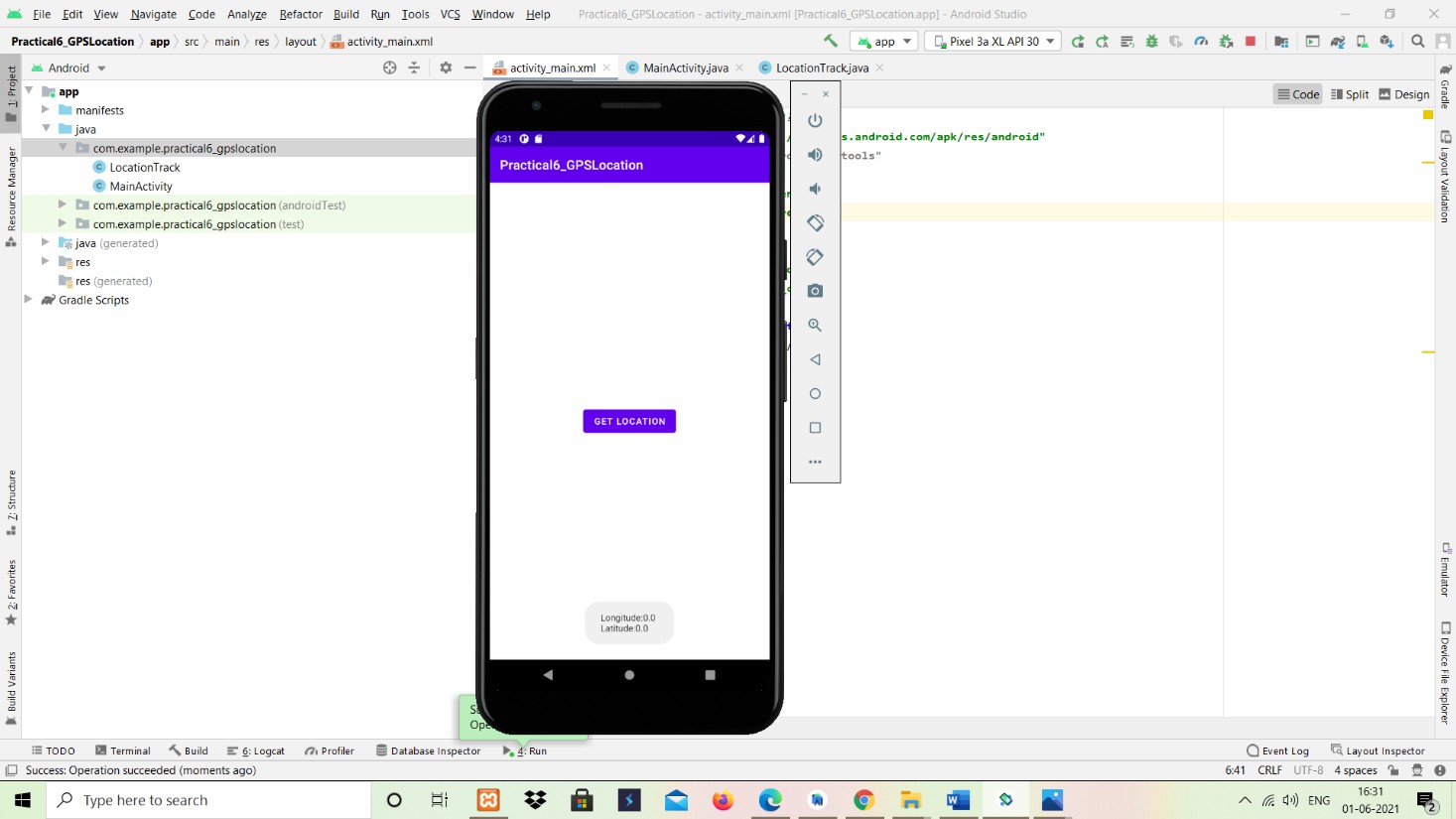
* TODO Auto-generated method stub return null;

}}

10)Go to manifest.xml file and add the code below <uses-permission android:name="android.permission.ACCESS\_FINE\_LOCATION"/> <uses-permission android:name="android.permission.INTERNET"/>

* + 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:

<?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>

**Backend:**

package com.example.exno10;

import android.app.Notification;

import android.app.NotificationManager;

import android.app.PendingIntent;

import android.content.Intent;

import android.os.Bundle;

import android.support.v7.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);

            }

        });

    }

}

**Output:**



**Practical No.-9**

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

**Hardware Requirements & Software Requirements:**

• Android Studio3.0

• Windows 7/8/10 OS

• RAM4GB

• 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"?>

<manifest

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

package="com.javapapers.androidalarmclock">

<uses-permission android:name="android.permission.WAKE\_LOCK" />

<application

android:allowBackup="true"

android:icon="@drawable/ic\_launcher"

android:label="@string/app\_name"

android:theme="@style/AppTheme">

<activity

android:name=".AlarmActivity" android:label="@string/app\_name"> <intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<service

android:name=".AlarmService"

android:enabled="true" />

<receiver android:name=".AlarmReceiver" />

</application>

</manifest>

**//**\*Android Activity\***//**

//\*activity\_my.xml**\*//**

//\*The Android Activity is designed to be simple. We have a

TimePicker component followed by a ToggleButton. That’s it.

Choose the time to set the alarm and toggle the switch to on.

The alarm will work\*//

<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"

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

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

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

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

tools:context=".MyActivity">

<TimePicker

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:id="@+id/alarmTimePicker"

android:layout\_alignParentTop="true"

android:layout\_centerHorizontal="true" />

<ToggleButton

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Alarm On/Off"

android:id="@+id/alarmToggle"

android:layout\_centerHorizontal="true"

android:layout\_below="@+id/alarmTimePicker"

android:onClick="onToggleClicked" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textAppearance="?android:attr/textAppearanceLarge"

android:text=""

android:id="@+id/alarmText"

android:layout\_alignParentBottom="true"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="20dp"

android:layout\_below="@+id/alarmToggle" />

</RelativeLayout>

**Backend:**

**//**\*AlarmActivity.java\*//

AlarmActivity uses the AlarmManager to set the alarm and send notification on alarm trigger.

package com.javapapers.androidalarmclock; import android.app.Activity;

import android.app.AlarmManager; import android.app.PendingIntent; import android.content.Intent; import android.os.Bundle; import android.util.Log;

import android.view.View; import android.widget.TextView; import android.widget.TimePicker; import android.widget.ToggleButton; import java.util.Calendar;

public class AlarmActivity extends Activity { AlarmManager alarmManager;

private PendingIntent pendingIntent; private TimePicker alarmTimePicker; private static AlarmActivity inst; private TextView alarmTextView;

public static AlarmActivity instance() {

return inst;

}

@Override

public void onStart() {

super.onStart();

inst = this;

}

@Override

protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_my);

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

alarmTextView = (TextView) findViewById(R.id.alarmText); ToggleButton alarmToggle = (ToggleButton) findViewById(R.id.alarmToggle);

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

public void onToggleClicked(View view) {

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

Log.d("MyActivity", "Alarm On");

Calendar calendar = Calendar.getInstance();

calendar.set(Calendar.HOUR\_OF\_DAY,

alarmTimePicker.getCurrentHour());

calendar.set(Calendar.MINUTE,

alarmTimePicker.getCurrentMinute());

Intent myIntent = new Intent(AlarmActivity.this, AlarmReceiver.class);

pendingIntent = PendingIntent.getBroadcast(AlarmActivity.this, 0,

myIntent, 0);

alarmManager.set(AlarmManager.RTC, calendar.getTimeInMillis(), pendingIntent);

} else { alarmManager.cancel(pendingIntent); setAlarmText(""); Log.d("MyActivity", "Alarm Off");

}

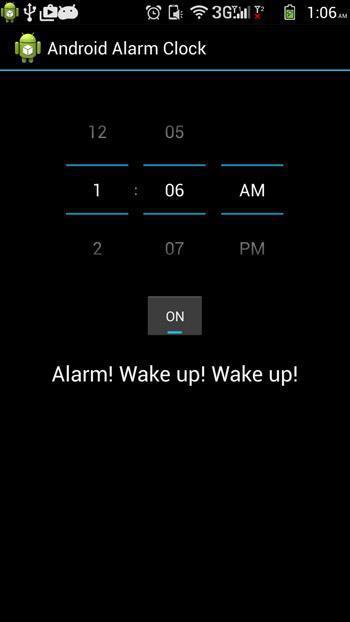
}

public void setAlarmText(String alarmText) { alarmTextView.setText(alarmText);

}

}

**Output:**



**Practical No.-10**

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

**Hardware Requirements & Software Requirements:**

• Android Studio3.0

• Windows 7/8/10 OS

• RAM4GB

• 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="fill\_parent"

android:layout\_height="fill\_parent"

android:background="#ff0000ff"

android:orientation="vertical" >

<EditText

android:id="@+id/editText1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" >

<requestFocus />

</EditText>

<Button

android:id="@+id/button1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="SAVE DATA" />

<Button

android:id="@+id/button2"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="SHOW DATA" />

<TextView

android:id="@+id/textView1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

/>

</LinearLayout>

**Backend:**

package save.sd;

import java.io.File;

import java.io.FileInputStream; import java.io.FileNotFoundException; import java.io.FileOutputStream; import java.io.IOException;

import java.io.InputStreamReader; import java.io.OutputStreamWriter; import android.app.Activity; import android.os.Bundle; import android.os.Environment;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

public class SavedatasdcardActivity extends Activity {

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

Button save,load;

EditText message;

TextView t1;

String Message1;

@Override

public void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.main);

save=(Button) findViewById(R.id.button1); load=(Button) findViewById(R.id.button2); message=(EditText) findViewById(R.id.editText1); t1=(TextView) findViewById(R.id.textView1); save.setOnClickListener(new View.OnClickListener(){ public void onClick(View v){

//Get message from user store in message1 variable Message1 =message.getText().toString(); try{

//Create a new folder called MyDirectory in SDCard

File sdcard=Environment.getExternalStorageDirectory();

File directory=new

File(sdcard.getAbsolutePath()+"/MyDirectory");

directory.mkdirs();

//Create a new file name textfile.txt inside MyDirectory File file=new File(directory,"textfile.txt"); //Create File Outputstream to read the file FileOutputStream fou=new FileOutputStream(file); OutputStreamWriter osw=new OutputStreamWriter(fou); try{

//write a user data to file

osw.append(Message1);

osw.flush();

osw.close();

Toast.makeText(getBaseContext(),"Data

Saved",Toast.LENGTH\_LONG).show();

}catch(IOException e){

e.printStackTrace();

}

}catch (FileNotFoundException e){ e.printStackTrace(); }

}

});

load.setOnClickListener(new View.OnClickListener(){ public void onClick(View v){ try{

File sdcard=Environment.getExternalStorageDirectory();

File directory=new

File(sdcard.getAbsolutePath()+"/MyDirectory"); File file=new File(directory,"textfile.txt"); FileInputStream fis=new FileInputStream(file); InputStreamReader isr=new InputStreamReader(fis); char[] data=new char[100]; String final\_data="";

int size;

try{

while((size=isr.read(data))>0)

{

//read a data from file

String read\_data=String.copyValueOf(data,0,size);

final\_data+=read\_data;

data=new char[100];

}

//display the data in output

Toast.makeText(getBaseContext(),"Message:"+final\_data,Toast.LENG

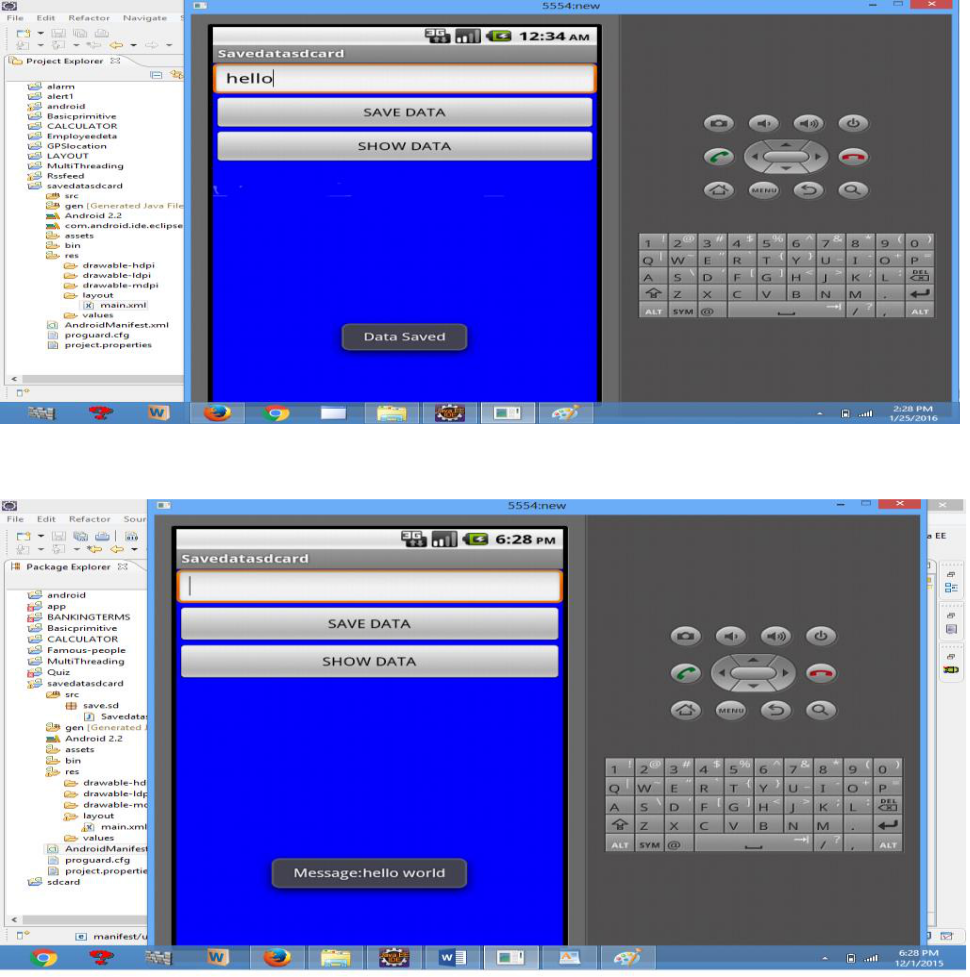
TH\_LONG).show();

}catch(IOException e){

e.printStackTrace();

}}catch (FileNotFoundException e){ e.printStackTrace(); }}});}}

**Output:**

****

**Github link :-**

**https://github.com/Sahil-934/MAD\_PRACTICALS.git**