Mobile Application Development Lab /EC354R19

III/IV B.TECH – I SEM-R19

LABORATORY MANUAL

PREPARED BY
MRS.B.MAHA LAKSHMI
ASSISTANT PROFESSOR
DEPARTMENT OF E.C.E

DEPARTMENT OF
ELECTRONICS AND COMMUNICATION ENGINEERING

BAPATLA WOMEN'S ENGINEERING COLLEGE::BAPATLA

(AFFILIATED TO ACHARYA NAGARJUNA UNIVERSITY)

BAPATLA,BAPATLA DISTRICT,ANDHRA PRADESH,INDIA PIN:522101

INSTITUTE VISION AND MISSION

VISION

Bapatla Women's Engineering College has been established with forethought to organize the growing needs of higher technical education for WOMEN, in order to provide them a competetive edge and a unique niche in the society.

MISSION

- Our mission is to impart the quality education at par with global standards to the students from all over India and in particular those from the local and rural areas.
- We continuously try to maintain high standards so as to make them technologically competent and ethically strong individuals who shall be able to improve the quality of life and economy of our country.

DEPARTMENT OF ELECTRONICS AND COMMUNICATION ENGINEERING

VISION

To produce highly professional and socially responsible women engineers through world class quality technical education and research to meet industry and Society needs in the field of Electronics and Communication Engineering.

MISSION

M1- To inculcate quality education for women through innovative learning practices with the use of modern tools.
M2- To impart professional ethics and moral values through co-curricular and extra-curricular activities as individual and team member.
M3- Empowering graduates through industrial trainings/on-job training in collaboration with industries with-a-view to life-long learning.

Programme Educational Objectives (PEOs)

PEO1: To develop and implementation of Electronics and Communication systems by imparting technical knowledge, professional leadership and communication skills as an individual and team in a multi-disciplinary stream.

PEO2: To train Graduates in solving complex problems, analyzing, designing and imparting research-based knowledge by using modern scientific tools to develop least cost-effective and innovative products.

PEO3: To produce women graduates through life-long learning, who ensure ethical and moral behavior.

Programme Specific Outcomes(PSO'S)

PSO1: To understand and analyze basic concepts of electronics and communication in various areas like signal processing, VLSI, embedded systems, communications, digital and analog devices.

PSO2: Ability to design and develop computing systems using concepts of mathematics, electronics engineering and other related inter disciplines to asses and resolve societal, environmental, health safety and cultural issues with professional ethics.

Programme Outcomes (POs)

- **PO1: Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- **PO2: Problem analysis:** Identify, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
- **PO3: Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
- **PO4: Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- **PO5: Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools prediction and modelling to complex engineering activities with an understanding of the limitations.
- **PO6: The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice
- **PO7: Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and the need for sustainable development.
- **PO8: Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering prac tice.
- **PO9: Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- **PO10: Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and receive clear instructions.
- **PO11: Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in team, to manage projects and in multidisciplinary environments.

- **PO12: Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

EC354R19 –Mobile Application Development Laboratory

SYLLABUS

EC354R19 MOBILE APPLICATION DEVELOPMENT LABORATORY L T P C
0 0 3 1.5

OBJECTIVES:

- ✓ To understand the components and structure of mobile application development frameworks for Android and windows OS based mobiles.
- ✓ To understand how to work with various mobile application development frameworks.
- ✓ To learn the basic and important design concepts and issues of development of mobile applications.
- ✓ To understand the capabilities and limitations of mobile devices.

LIST OF EXPERIMENTS

1. Develop an application that uses GUI components, Font and Colours
2. Develop an application that uses Layout Managers and event listeners.
3. Develop an application that makes use of databases.
4. Develop an application that makes use of Notification Manager
5. Develop a native application that uses GPS location information.
6. Implement an application that for basic Calculator.
7. Implement an application that creates an alert upon receiving a message
8. Write a mobile application that makes use of RSS feed
9. Develop a mobile application to send an email.
10. Develop a Mobile application for simple needs (Mini Project)

OUTCOMES:

Upon Completion of the course, the students will be able to:

- ✓ Develop mobile applications using GUI and Layouts.
- ✓ Develop mobile applications using Event Listener.
- ✓ Develop mobile applications using RSS Feed, Internal/External Storage, SMS and GPS.
- ✓ Analyze and discover own mobile app for simple needs.

REFERENCES:

1. Build Your Own Security Lab, Michael Gregg, Wiley India

EC354R19 –Mobile Application Development Laboratory

LIST OF EXPERIMENTS

1. Develop an application that uses GUI components, Font and Colours
2. Develop an application that uses Layout Managers and event listeners.
3. Develop an application that makes use of databases.
4.Develop an application that makes use of Notification Manager
5Develop anative application that uses GPS location information.
6. Implement an application that for basic Calculator.
7. Implement an application that creates an alert upon receiving a message
8.Write a mobile application that makes use of RSS feed
9. Develop a mobile application to send an email.
10. Develop a Mobile application for simple needs (Mini Project)

Course Outcomes (COs):

Course No.	Course Outcome (Student should be able to)	Bloom's Taxonomy Level
C354.1	Build various mobile applications related to GUI components, Font and Colours using Layout managers and Event Listeners	Apply
C354.2	Make use of database concepts for data related mobile applications	Apply
C354.3	Develop mobile applications using Internal & External Storage, SMS Notification, Multithreading, and GPS.	Apply
C354.4	Model the mobile application using RSS Feed, Alert upon receiving a message and send an E-Mail	Apply
C354.5	Develop Mobile App for simple real time needs	Apply

Mapping of Course Outcomes with PSOs and POs:

S.No.	Course out come	Program Specific Outcomes (PSOs)		Program Outcomes(POs)											
		PSO 1	PSO 2	PO 1	PO 2	PO 3	PO 4	PO 5	PO6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
1	C354.1	l	l	1	-	1	-	2	-	-	-	1	-	-	1
2	C354.2	l	l	1	-	1	-	2	-	-	-	1	-	-	1
3	C354.3	l	l	1	-	1	-	2	-	-	-	1	-	-	1
4	C354.4	l	l	1	-	1	-	2	-	-	-	1	-	-	1
5	C354.5	2	2	2	2	2	1	2	2	-	-	2	1	-	2
	C354	1.2	1.2	1 .2	2	1.2	1	2	2	-	-	1.2	1	-	1.2

Correlation Levels:---1: Low 2: Medium 3: Strong, '-' No correlation

EC354R19 –Mobile Application Development Laboratory 2021-2022 (ODD)

INDEX

Pgm.No.	Name of the Experiment	Page No.
I	Introduction to Android OS & Android Studio	8
1	Develop an application that uses GUI components, Font and Colours	15
2	Develop an application that uses Layout Managers and event listeners.	20
3	Develop an application that makes use of databases.	27
4	Develop an application that makes use of Notification Manager	35
5	Develop a native application that uses GPS location information.	46
6	Implement an application that for basic Calculator.	52
7	Implement an application that creates an alert upon receiving a message	62
8	Write a mobile application that makes use of RSS feed	68
9	Develop a mobile application to send an email.	75
10	Develop a Mobile application for simple needs (Mini Project)	82

INTRODUCTION TO ANDROID OS

INTRODUCTION :

Android is a Linux based operating system it is designed primarily for touch screen mobile devices such as smart phones and tablet computers. The android is a powerful operating system and it supports large number of applications in Smartphones. These applications are more comfortable and advanced for the users. The hardware that supports android software is based on ARM architecture platform.

The android is an open source operating system. The android has got millions of apps available that can help you managing your life one or other way and it is available low cost in market at that reasons android is very popular.

The android development supports with the full java programming language. Even other packages that are API and JSE are not supported. The first version 1.0 of android development kit (SDK) was released in 2008 .

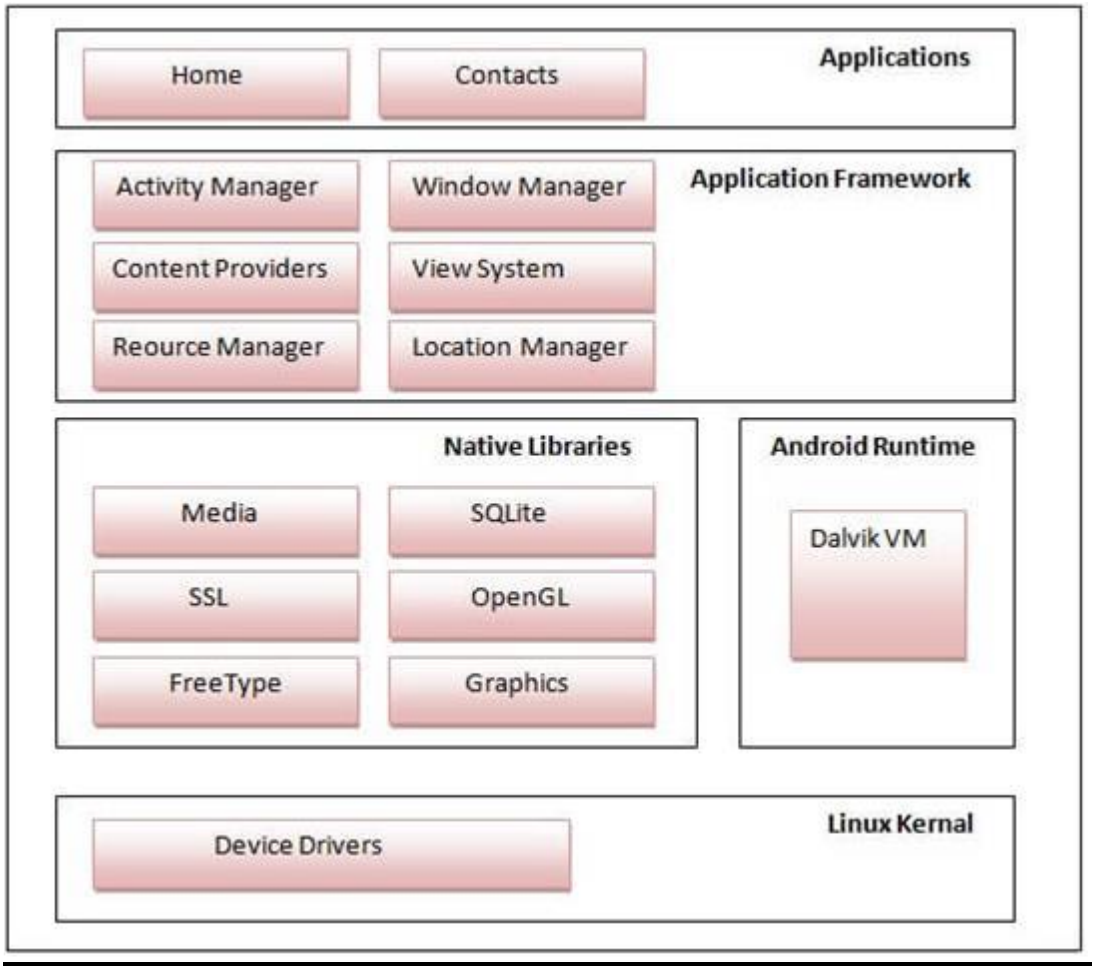
ANDROID ARCHITECTURE :

The android is an operating system and is a stack of software components which is divided into five sections and four main layers that is

- Linux kernel
- Libraries
- Android runtime

LINUX KERNEL :

The android uses the powerful Linux kernel and it supports wide range of hardware drivers. The kernel is the heart of the operating system that manages input and output requests from software. This provides basic system functionalities like process management, memory management; device management like camera, keypad, and display etc., the kernel handles all the things. The Linux is really good at networking and it is not necessary to interface it to the peripheral hardware. The kernel itself does not interact directly with the user but rather interacts with the shell and other programs as well as with the hardware devices on the system.



LIBRARIES:

On top of a Linux kennel there is a set of libraries including open source web browser such as webkit, library libc. These libraries are used to play and record audio and video. The SQLite is a data base which is useful for storage and sharing of application data. The SSL libraries are responsible for internet security etc.

ANDROID RUNTIME :

The android runtime provides a key component called **Dalvik Virtual Machine** which is a kind of java virtual machine. It is specially designed and optimized for android. The Dalvik VM is the process virtual machine in the android operating system. It is software that runs apps on android devices.

The Dalvik VM makes use of Linux core features like memory management and multithreading which is in a java language. The Dalvik VM enables every android application to run its own process. The Dalvik VM executes the files in the .dex format.

APPLICATION FRAME WORK ;

The application frame work layer provides many higher level services to applications such as windows manager, view system, package manager, resource manager etc. The application developers are allowed to make use of these services in their application.

APPLICATIONS AND FEATURES:

The android applications are at the top layer and we will write our application and install on this layer. Example of such applications is contacts, books, browsers, services etc.

FEATURES:

- Head set layout
- Storage
- Connectivity: GSM/EDGE, IDEN, CDMA, Bluetooth, WI-FI, EDGE,3G,NFC, LTE,GPS.
- Messaging: SMS, MMS, C2DM (could to device messaging), GCM (Google could messaging)
- Multilanguage support
- Multi touch
- Video calling
- Screen capture
- External storage
- Streaming media support
- Optimized graphics

ANDROID VERSIONS:

- Android 1.5: Android Cupcake
- Android 1.6: Android Donut
- Android 2.0: Android Eclair
- Android 2.2: Android Froyo
- Android 2.3: Android Gingerbread
- Android 3.0: Android Honeycomb
- Android 4.0: Android Ice Cream Sandwich
- Android 4.1 to 4.3.1: Android Jelly Bean
- Android 4.4 to 4.4.4: Android KitKat
- Android 5.0 to 5.1.1: Android Lollipop
- Android 6.0 to 6.0.1: Android Marshmallow
- Android 7.0 to 7.1: Android Nougat
- Android 8.0 to Android 8.1: Android Oreo
- Android 9.0: Android Pie

INTRODUCTION TO ANDROID STUDIO

Android Studio is the official integrated development environment (IDE) for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development. It is available for download on Windows, macOS and Linux based operating systems. It is a replacement for the Eclipse Android Development Tools (ADT) as the primary IDE for native Android application development.

Android Studio was announced on May 16, 2013 at the Google I/O conference. It was in early access preview stage starting from version 0.1 in May 2013, then entered beta stage starting from version 0.8 which was released in June 2014. The first stable build was released in December 2014, starting from version 1.0. Since May 7, 2019, Kotlin is Google's preferred language for Android app development. Still, other programming languages are supported by Android Studio, such as Java and C++.

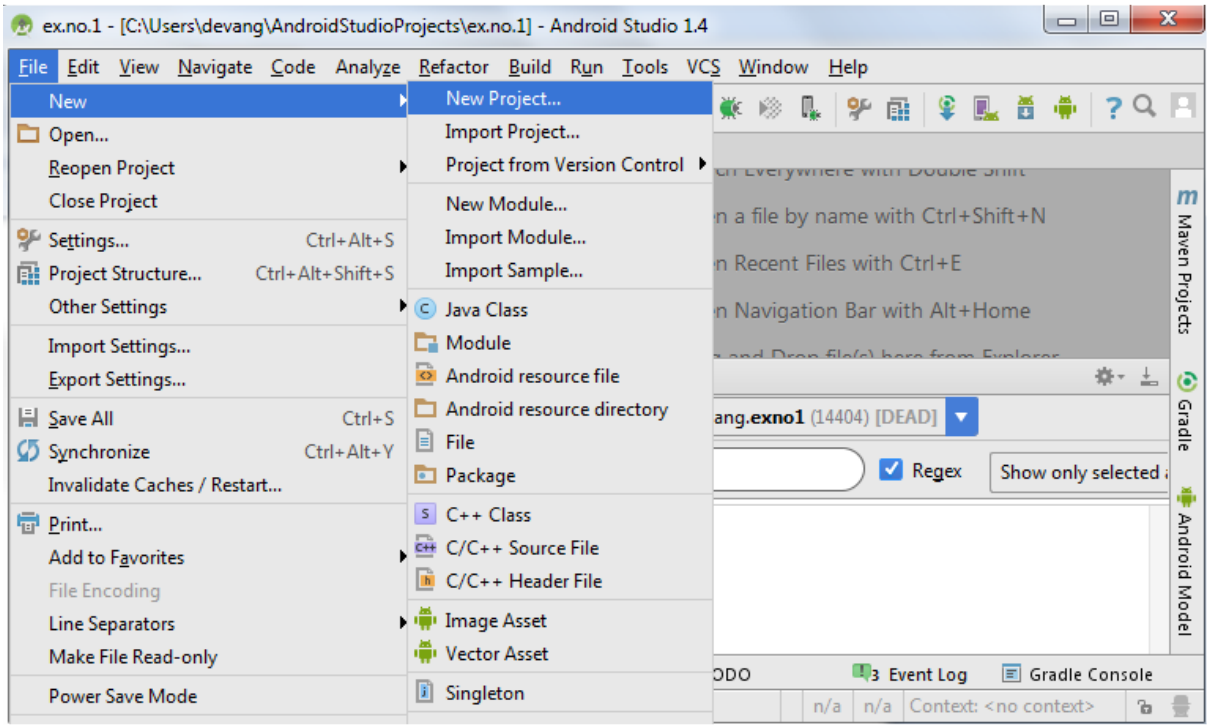
FEATURES:

- ✓ Gradle-based build support
- ✓ Android-specific refactoring and quick fixes
- ✓ Lint tools to catch performance, usability, version compatibility and other problems
- ✓ ProGuard integration and app-signing capabilities
- ✓ Template-based wizards to create common Android designs and components
- ✓ A rich layout editor that allows users to drag-and-drop UI components, option to preview layouts on multiple screen configurations
- ✓ Support for building Android Wear apps
- ✓ Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging (Earlier 'Google Cloud Messaging') and Google App Engine
- ✓ Android Virtual Device (Emulator) to run and debug apps in the Android studio.

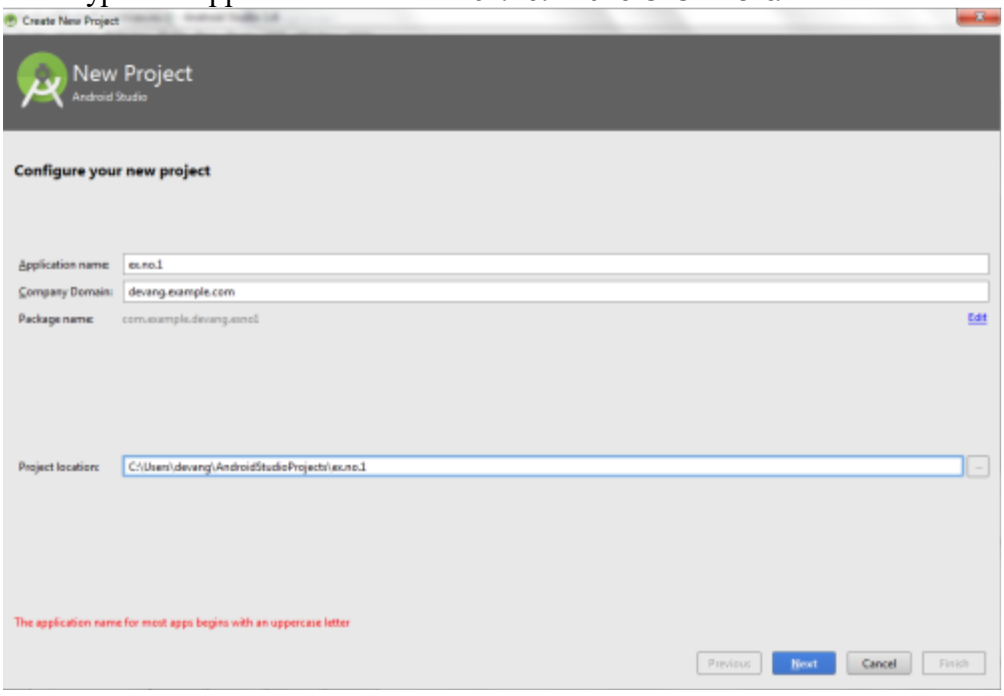
PROCEDURE:

Creating a New Project

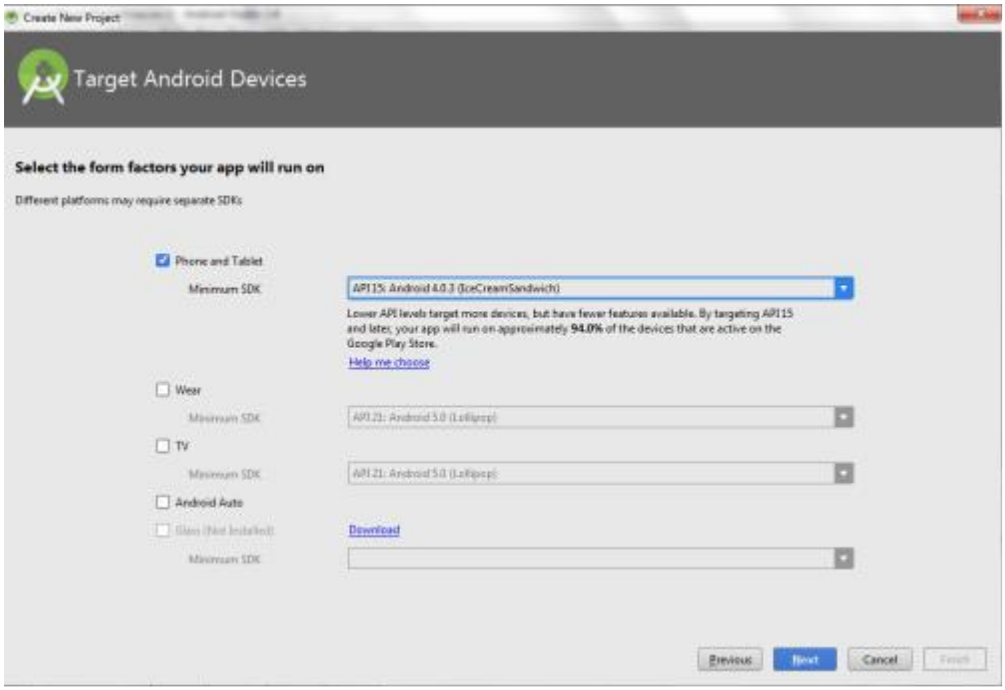
- Open Android Stdio and then click on **File -> New ->New project.**



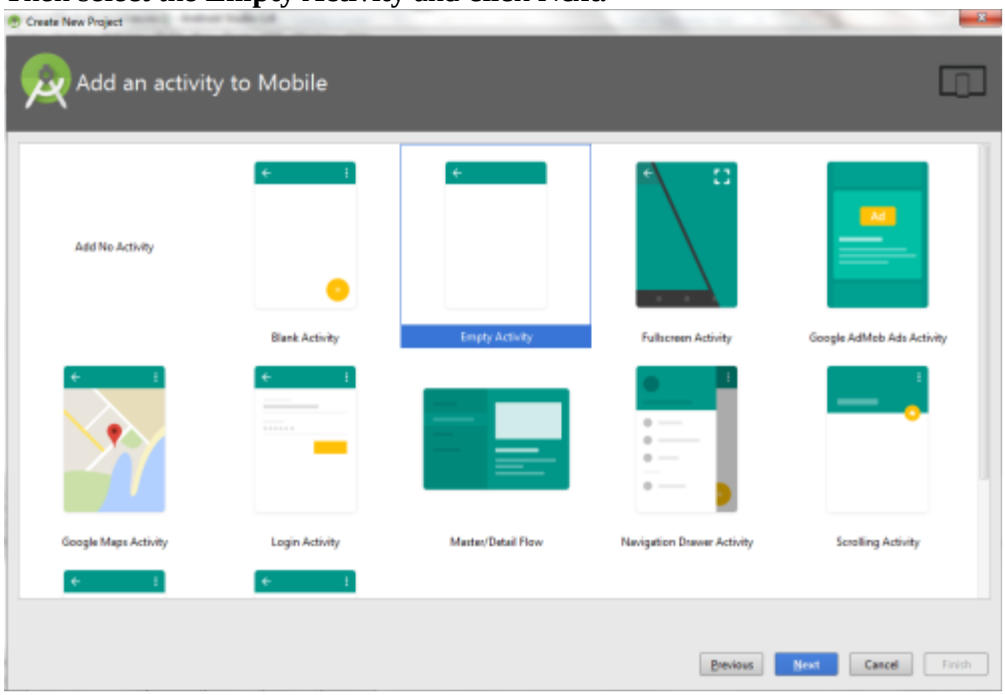
✓ Then type the Application name as "**ex.no.1"** and click **Next.**



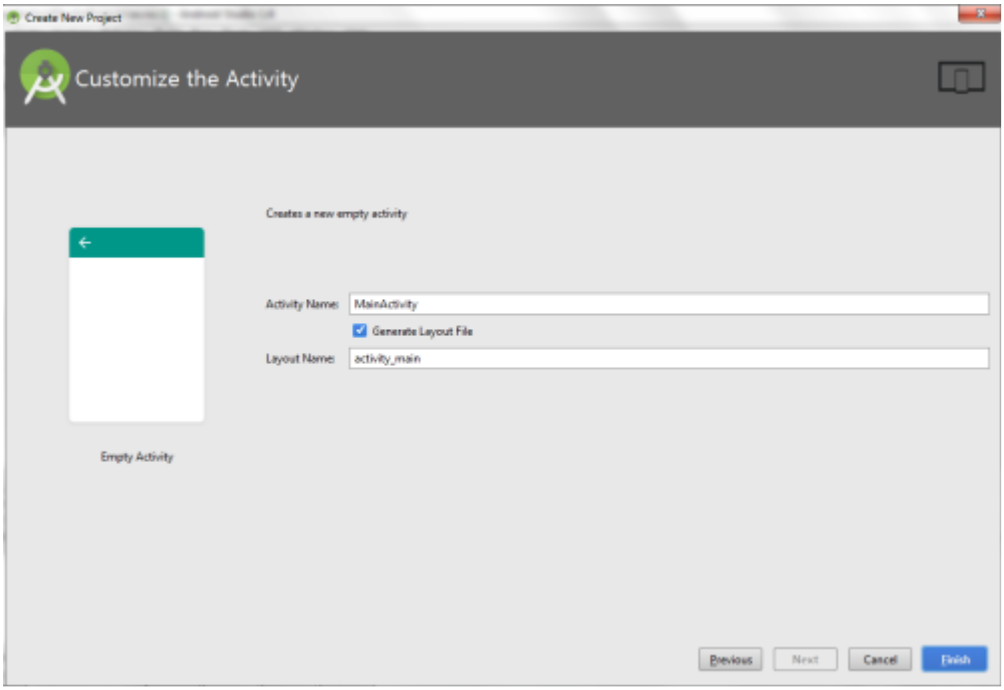
✓ Then select the **Minimum SDK** as shown below and click **Next**



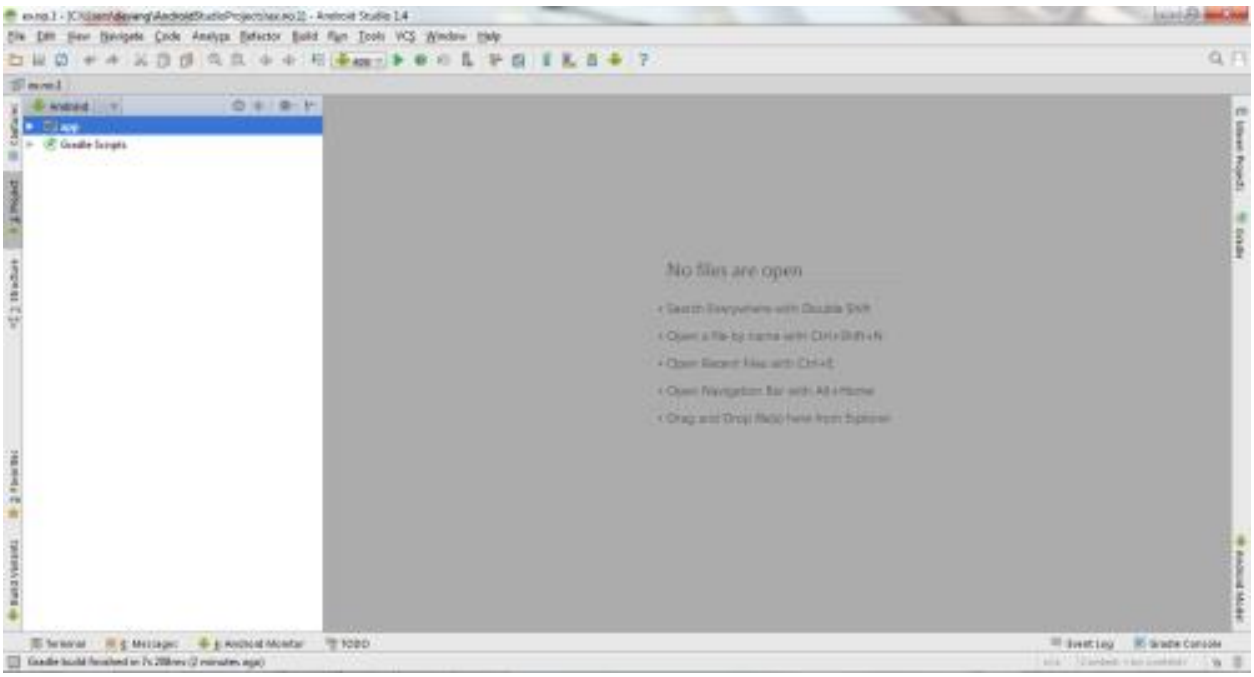
✓ Then select the **Empty Activity** and click **Next.**



✓ Finally click F**inish.**



- ✓ It will take some time to build and load the project.
- ✓ After completion it will look as given below.



PGM:01	GUI Components, Font and Colours

AIM :

To develop a Simple Android Application that uses GUI components, Font and Colours.

PROCEDURE :

Step 1: Create an android project using android studio.

Step 2: Design

2.1. Open the actual Project folder (app) in Android Studio IDE.

2.2. Click res directory -> layout -> activity_main.xml -> Design.

2.3. Insert the GUI components to **Design view** in activity_main.xml.

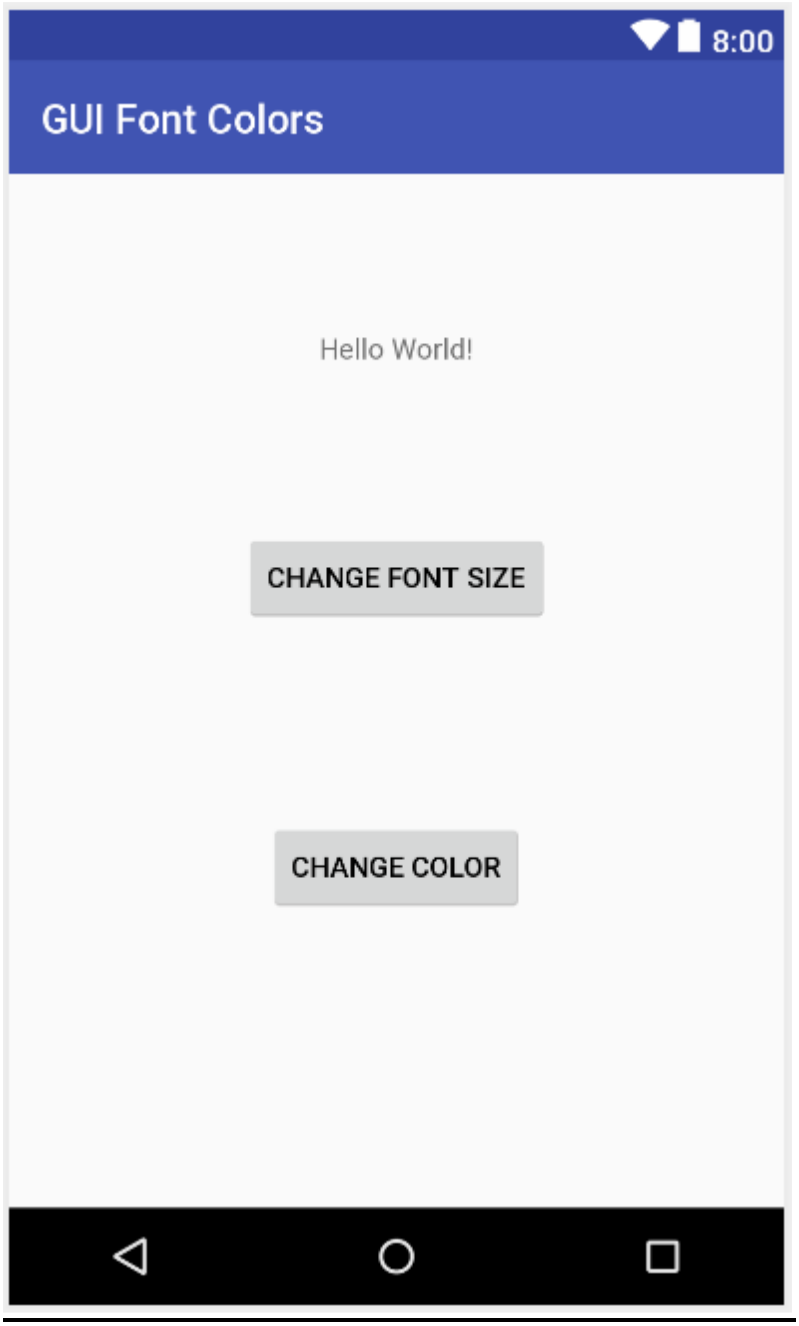
2.4. Enter the id for each component.

Step 3: Open res directory -> layout ->**activity_main.xml** and add the code.

Step 4: Open java ->**MainActivity.java** and add the code.

Step 5: Run the project and the output will be displayed in Emulator.

LAYOUT DESIGN:



XML File :-

```
<?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" />
```

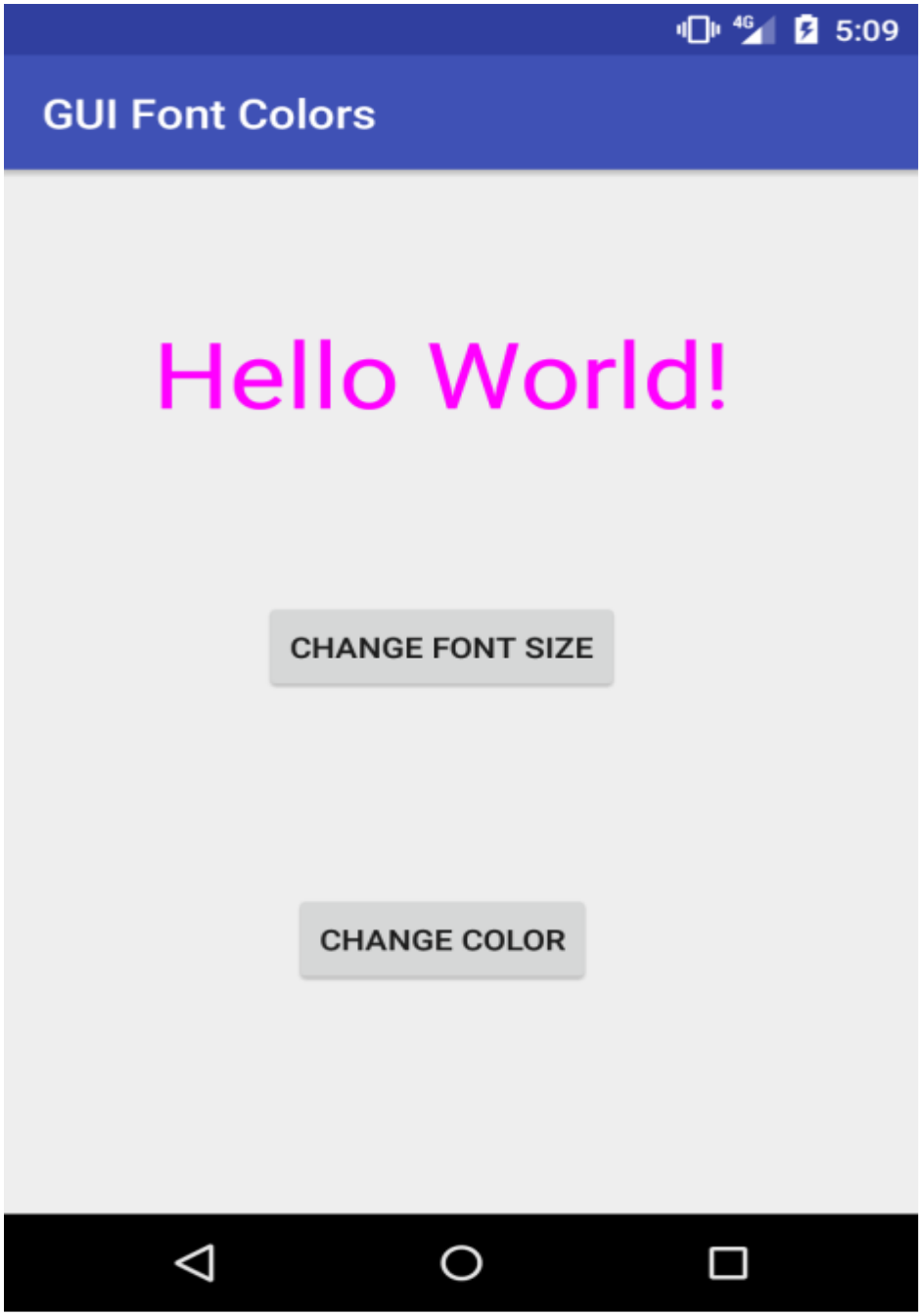
```
<Button android:id="@+id/button1" android:layout_width="match_parent"
android:layout_height="wrap_content" android:layout_margin="20dp"
android:gravity="center" android:text="Change font size" android:textSize="25sp" />

<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>
```

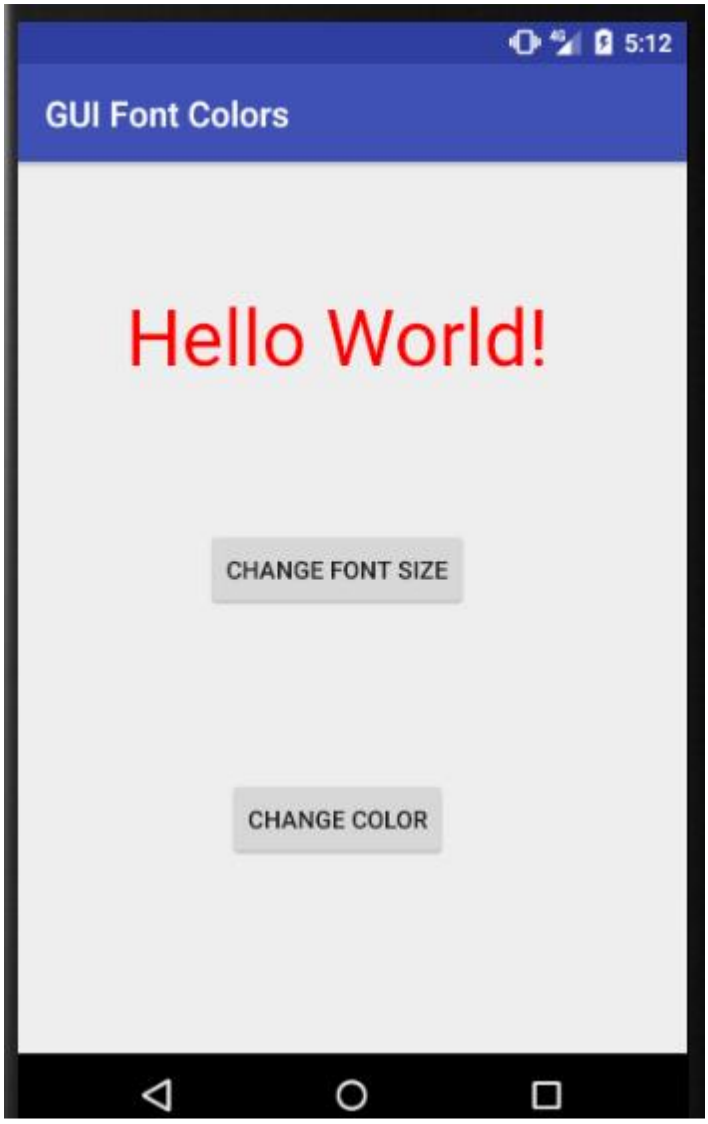
JAVA File :-

```
package com.example.exno1;

import android.graphics.Color;
import android.os.Bundle;
//import android.support.v7.app.AppCompatActivity;
import androidx.appcompat.app.AppCompatActivity;
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:-


5:09
GUI Font Colors
Hello World!
CHANGE FONT SIZE
CHANGE COLOR



Viva-Voce Questions:-

1) What is meant by class and method?
2) What is the full form of GUI?
3) What is meant by sp and dp in layout file?
4) What is the package used in this program?
5) What is the differences between xml file and java file?

PGM:02	An application that uses layout managers and event listeners

AIM :

To develop a Simple Android Application that uses Layout Managers and Event Listeners.

PROCEDURE:

Step 1: Create an android project using android studio.

Step 2: Design

2.1. Open the actual Project folder (app) in Android Studio IDE.

2.2. Click res directory -> layout -> activity_main.xml -> Design.

2.3. Insert the GUI components to **Design view** in activity_main.xml.

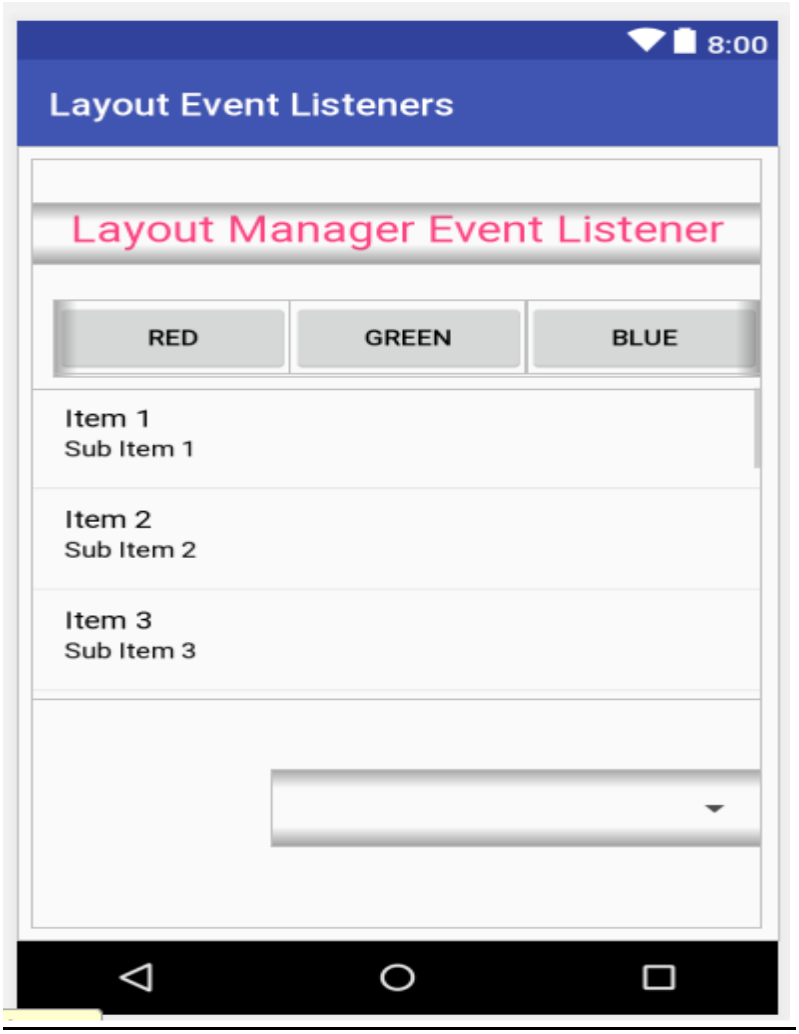
2.4. Enter the id for each component.

Step 3: Open res directory -> layout ->**activity_main.xml** and add the code.

Step 4: Open java ->**MainActivity.java** and add the code.

Step 5: Run the project and the output will be displayed in Emulator.

LAYOUT DESIGN:



XML FILE:-

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:id="@+id/relativeLayout1" android:layout_width="fill_parent"
android:layout_height="fill_parent" >
<LinearLayout
android:id="@+id/linearLayout1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_alignParentRight="true"
android:layout_alignParentTop="true" >
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="center"
android:text="ADDITION"
android:textSize="20dp" >
</TextView>
</LinearLayout>
```

```
<LinearLayout
android:id="@+id/linearLayout2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_alignParentRight="true"
android:layout_below="@+id/linearLayout1" >
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="ENTER NO 1" >
</TextView>
<EditText
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_weight="0.20"
android:id="@+id/edittext1"
android:inputType="number">
</EditText>
</LinearLayout>

<LinearLayout
android:id="@+id/linearLayout3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_alignParentRight="true"
android:layout_below="@+id/linearLayout2" >
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="ENTER NO 2" >
</TextView>
<EditText
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_weight="0.20"
android:id="@+id/edittext2"
android:inputType="number">
</EditText>
</LinearLayout>

<LinearLayout
android:id="@+id/linearLayout4"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_alignParentRight="true"
android:layout_below="@+id/linearLayout3" >
```

```
<Button
android:layout_width="wrap_content"
android:id="@+id/button1"
android:layout_height="wrap_content"
android:text="Addition"
android:layout_weight="0.50" />

<Button
android:layout_width="wrap_content"
android:id="@+id/button3"
android:layout_height="wrap_content"
android:text="subtraction"
android:layout_weight="0.50" />

<Button
android:layout_width="wrap_content"
android:id="@+id/button2"
android:layout_height="wrap_content"
android:text="CLEAR"
android:layout_weight="0.50" />

</LinearLayout>

<View
android:layout_height="2px"
android:layout_width="fill_parent"
android:layout_below="@+id/linearLayout4"
android:background="#DDFFDD"/>

</RelativeLayout>
```

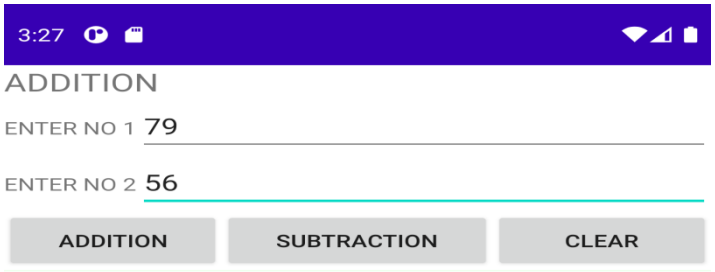
JAVA File :-

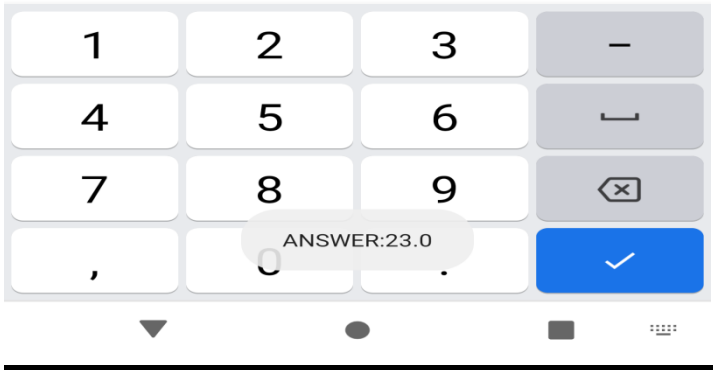
```
//package layout.ne;
package com.example.exno2;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends Activity
{
/** Called when the activity is first created. */ EditText
txtData1,txtData2; float num1,num2,result1,result2;
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
```

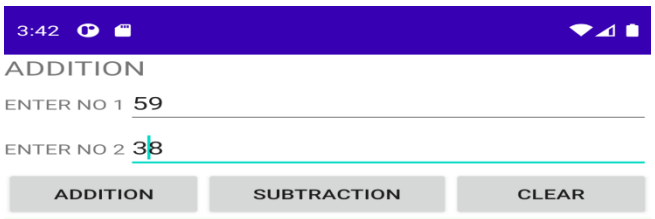
```
//setContentView(R.layout.main);
setContentView(R.layout.activity_main);
        Button add = (Button)
              findViewById(R.id.button1);
        add.setOnClickListener(new OnClickListener()
        { public void onClick(View v) {
try
{
txtData1 = (EditText) findViewById(R.id.edittext1);
txtData2 = (EditText) findViewById(R.id.edittext2);
num1 = Float.parseFloat(txtData1.getText().toString()); num2 =
Float.parseFloat(txtData2.getText().toString());
result1=num1+num2;

Toast.makeText(getBaseContext(),"ANSWER:"+result1,Toast.LENGTH_SHORT).show();
            }
catch(Exception e)
            {
               Toast.makeText(getBaseContext(), e.getMessage(),
                    Toast.LENGTH_SHORT).show();
            }
        }
        });
        Button sub = (Button) findViewById(R.id.button3);
        sub.setOnClickListener(new OnClickListener() {
public void onClick(View v) {
try
{
txtData1 = (EditText) findViewById(R.id.edittext1);
txtData2 = (EditText) findViewById(R.id.edittext2);
num1 = Float.parseFloat(txtData1.getText().toString());
num2 = Float.parseFloat(txtData2.getText().toString());
result2=num1-num2;

Toast.makeText(getBaseContext(),"ANSWER:"+result2,Toast.LENGTH_SHORT).show();
               }
catch(Exception e)
               {
                  Toast.makeText(getBaseContext(), e.getMessage(),
                       Toast.LENGTH_SHORT).show();
               }
            }
        });

        Button clear = (Button) findViewById(R.id.button2);
        clear.setOnClickListener(new OnClickListener() {
public void onClick(View v) {
try
{
txtData1.setText("");
```

```
txtData2.setText("");
            }
catch(Exception e)
            {
               Toast.makeText(getBaseContext(), e.getMessage(),
                    Toast.LENGTH_SHORT).show();
            }
         }
      });
   }
}
```

OUTPUT:-







Viva-Voce Questions:-

1) What is meant by layout managers and event listeners?
2) Write the syntax for button.
3) List out the different types of layouts?
4) What is the purpose of"Toast" command?
5) What is the purpose of "Float" command ?

PGM:03 An application that makes use of databases

AIM :-

To develop a Simple Android Application that makes use of Database.

PROCEDURE :-

Step 1: Create an android project using android studio.

Step 2: Create two resource files (*.xml) and two activity files (*.java) named activity_ main .xml & activity_main2.xml and MainActivity.java & Main2Activity.java.

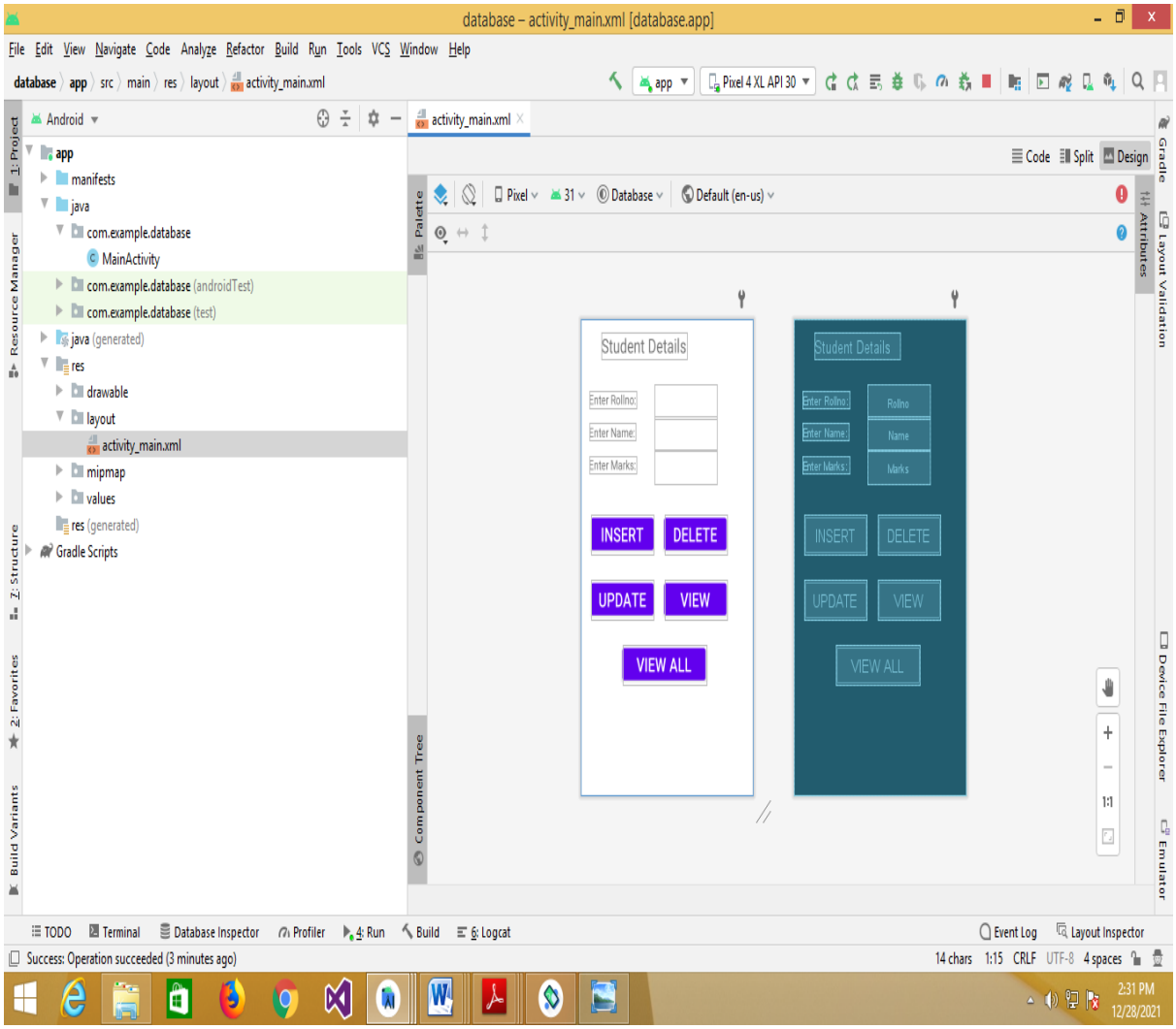
Step 3: Open res directory -> layout -> activity_main.xml -> Click -> Design button at bottom of the Android Studio. Put the necessary components for both resource files (activity_main.xml, activity_main2.xml).

Step 4: Design (After the design part, the xml code will be generated automatically in the layout file).

Step 5: Open Java ->**MainActivity.java & MainActivity2.java** and add the code.

Step 6: Run the project and the output will be displayed in Emulator.

LAYOUT DESIGN:-



XML FILE:-

```
<?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>
```

JAVA FILE:-

```
package com.example.prgm3;

import android.app.Activity;
import android.app.AlertDialog.Builder;
```

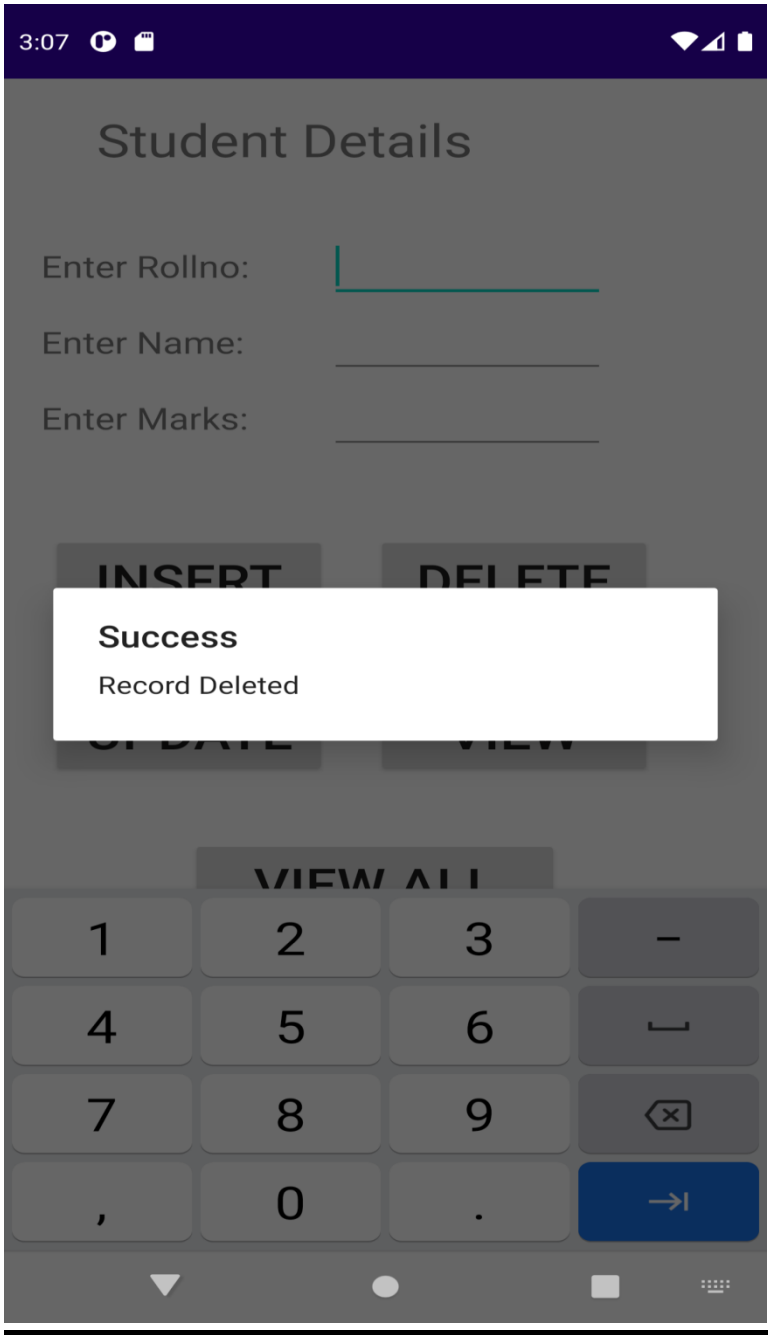
```
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;

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:-



Viva-Voce Questions:-

1) What is the database package imported in this program?
2) What is the size of text used in this program?
3) How many text fields are recorded on the database?
4) What are the different buttons assigned on database data?
5) What is the purpose of "builder " function?

PGM:04	Develop an application that makes use of Notification Manager

AIM :

To develop a Simple Android Application that makes use of Notification Manager.

PROCEDURE :

Step 1: Create an android project using android studio.

Step 2: Design

2.1. Open the actual Project folder (app) in Android Studio IDE.

2.2. Click res directory -> layout ->**activity_main.xml** -> Design.

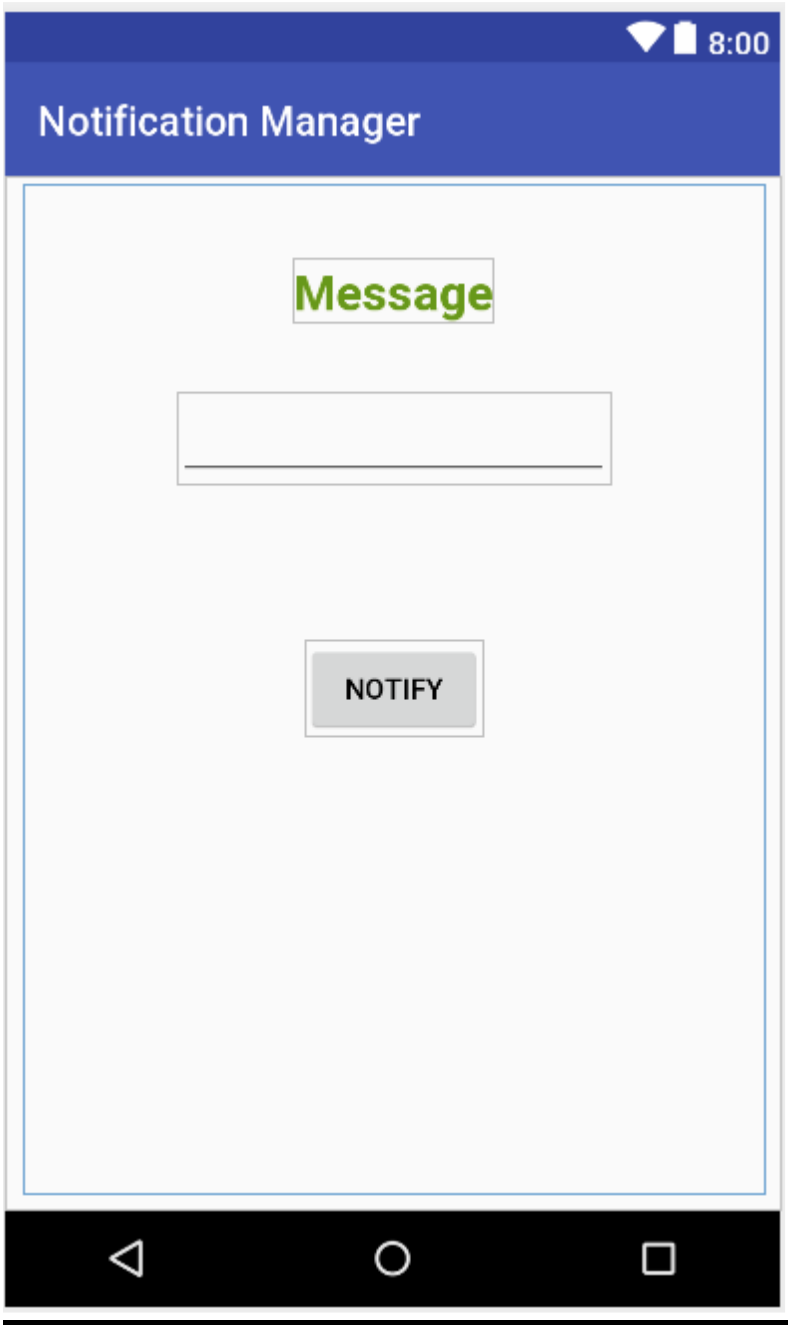
2.3. Insert the GUI components to **Design view** in activity_main.xml.

2.4. Enter the id for each component.

Step 3: Open Java ->**MainActivity.java** and add the code.

Step 4: Run the project and the output will be displayed in Emulator.

LAYOUT DESIGN:



MAIN ACTIVITYXML FILE:-

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

<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
xmlns:app="http://schemas.android.com/apk/res-auto"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity">
```

```
<Button
android:text="Notify"
android:textAllCaps="false"
android:layout_width="200dp"
android:layout_height="wrap_content"

android:id="@+id/btnNotify"
android:textColor="@android:color/white"
android:textSize="18dp"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintBottom_toBottomOf="parent"/>

</android.support.constraint.ConstraintLayout>
```

SECOND ACTIVITY XML FILE:-

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

<android.support.constraint.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="com.spaceo.notificationalertdemo.MainActivity">

<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Hello World!"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintLeft_toLeftOf="parent"
app:layout_constraintRight_toRightOf="parent"
app:layout_constraintTop_toTopOf="parent"/>

</android.support.constraint.ConstraintLayout>
```

MAIN ACTIVITY JAVA FILE:-

```
package com.example.notimgr;

//package com.spaceo.notificationalertdemo
import android.app.Notification;
import android.app.NotificationChannel;
import android.app.NotificationManager;
import android.app.PendingIntent;
```

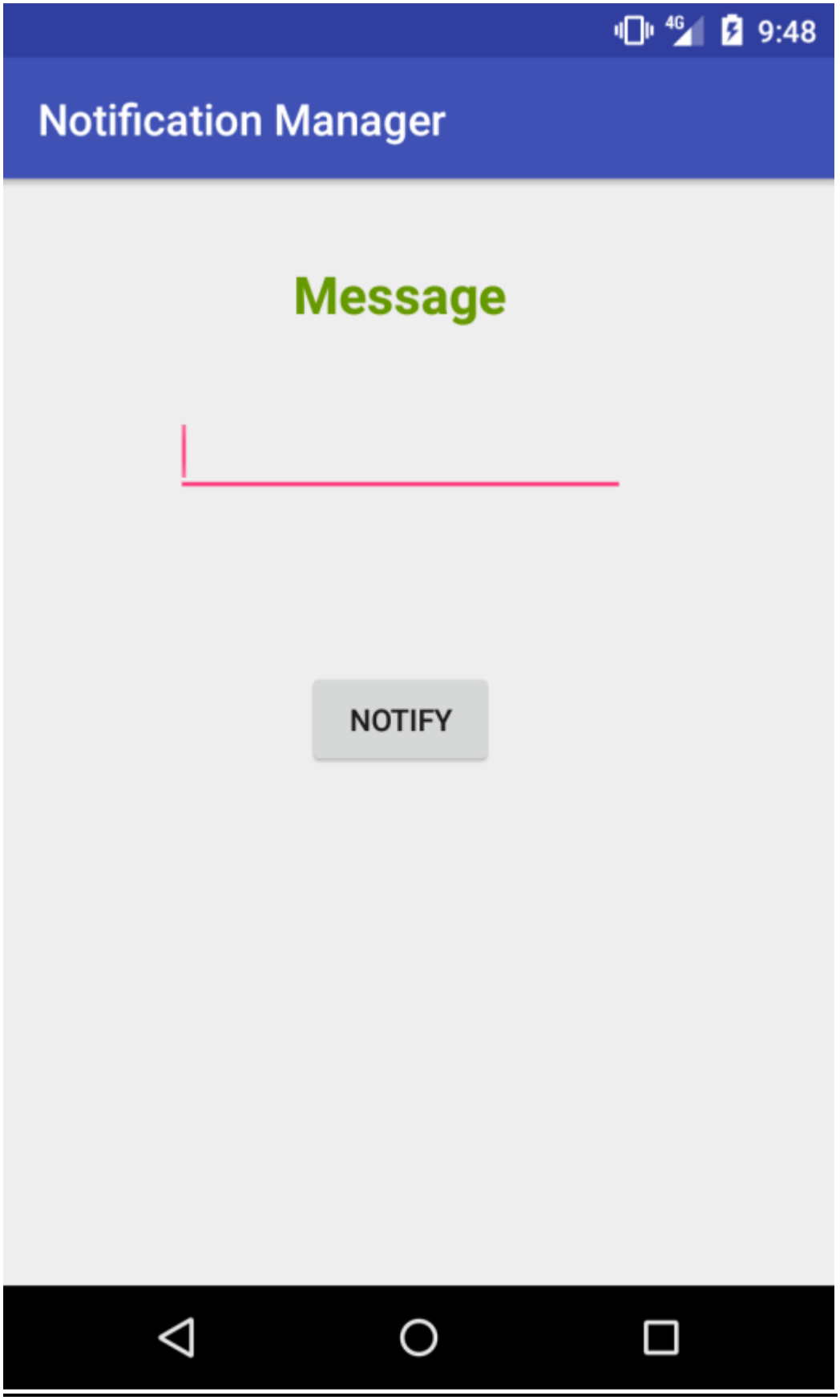
```
import android.content.Context;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.support.v4.app.NotificationCompat;
import android.support.v7.app.AppCompatActivity;
import kotlinx.android.synthetic.main.activity_main.*;
public class MainActivity extends AppCompatActivity
{
private val NOTIFY_ME_ID = 1337;
private val NOTIFICATION_CHANNEL_ID = "My Channel";

@Override   funonCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState);

      setContentView(R.layout.activity_main);

//After Click On Notification Jump To NotifyMessage

val notificationIntent = Intent(this, NotifyMessage::class.java)

//PendingIntent Create

val pendingIntent = PendingIntent.getActivity(this, 0, notificationIntent, 0);

//Notification Builder Create

val builder = NotificationCompat.Builder(this@MainActivity,
NOTIFICATION_CHANNEL_ID);
//Custom Notification Create
builder.setSmallIcon(R.drawable.notification)
      .setContentTitle(resources.getString(R.string.tvTitleofnotification))
      .setContentText(resources.getString(R.string.tvBodyofNotification))
      .setContentIntent(pendingIntent)
      builder.notification.flags = builder.notification.flags or
Notification.FLAG_AUTO_CANCEL
      builder.setAutoCancel(true)
//NotificationManager Create
val notificationManager = getSystemService(Context.NOTIFICATION_SERVICE) as
NotificationManager
//If API level 26 or higher than create channel for Notification
if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
//channel Create
val notificationChannel = NotificationChannel(NOTIFICATION_CHANNEL_ID,
      resources.getString(R.string.app_name),
NotificationManager.IMPORTANCE_DEFAULT)
      notificationManager.createNotificationChannel(notificationChannel)
      }
//Button Click After Generate Notification
btnNotify.setOnClickListener {
```

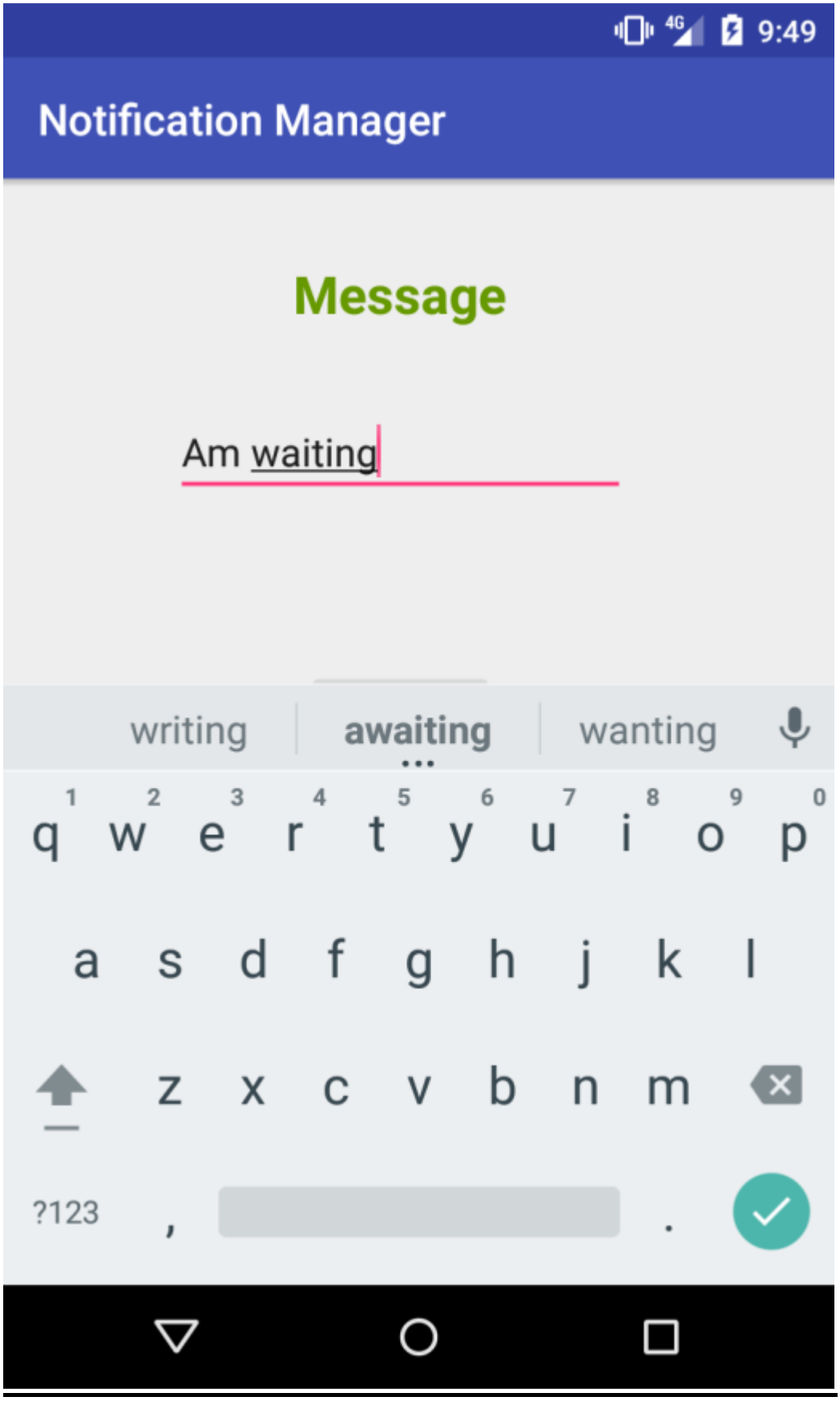
```
        notificationManager.notify(R.drawable.notification, builder.build())
        }
        }

private void setContentView(int activity_main) {
    }
}
```

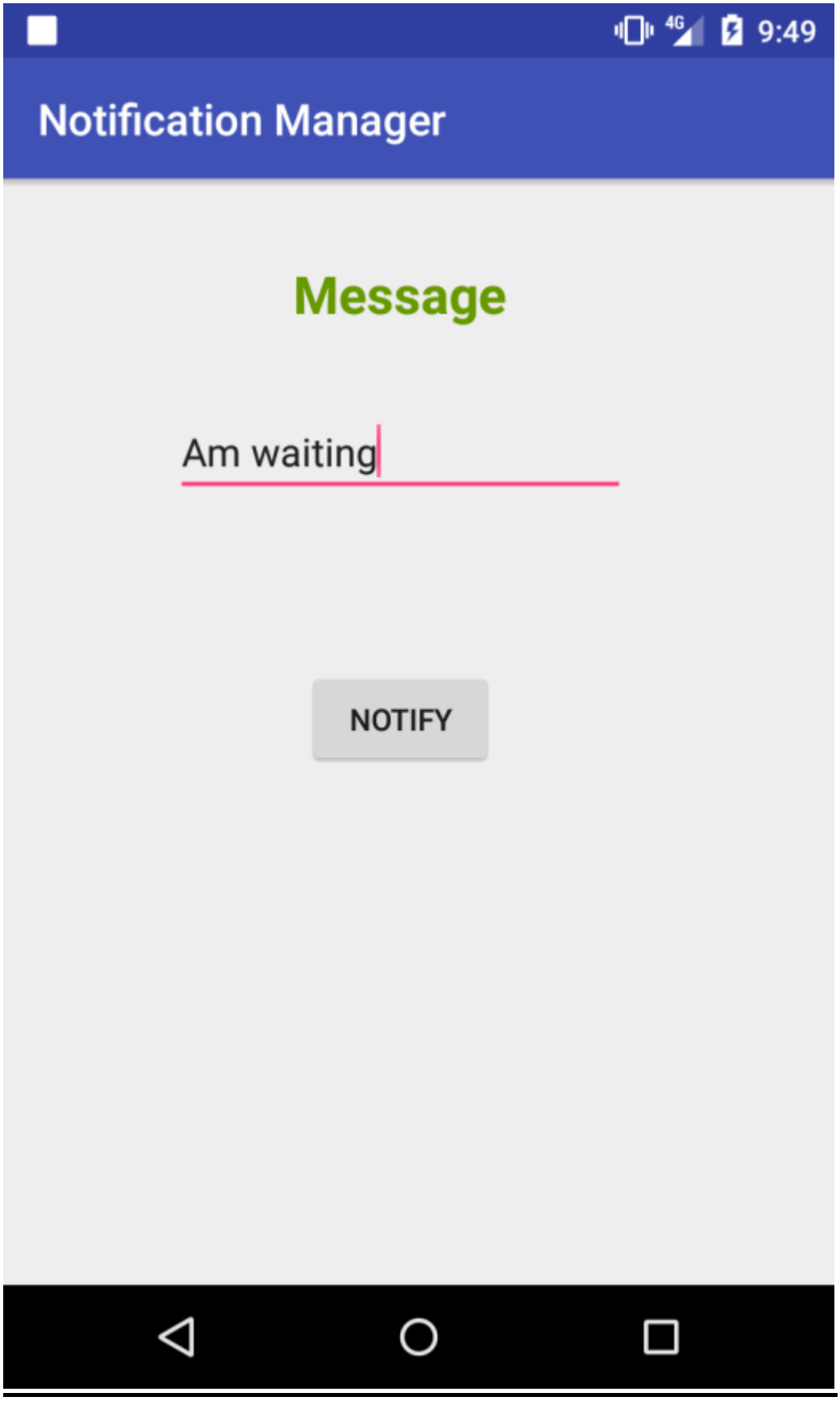
SECOND ACTIVITY JAVA FILE:-

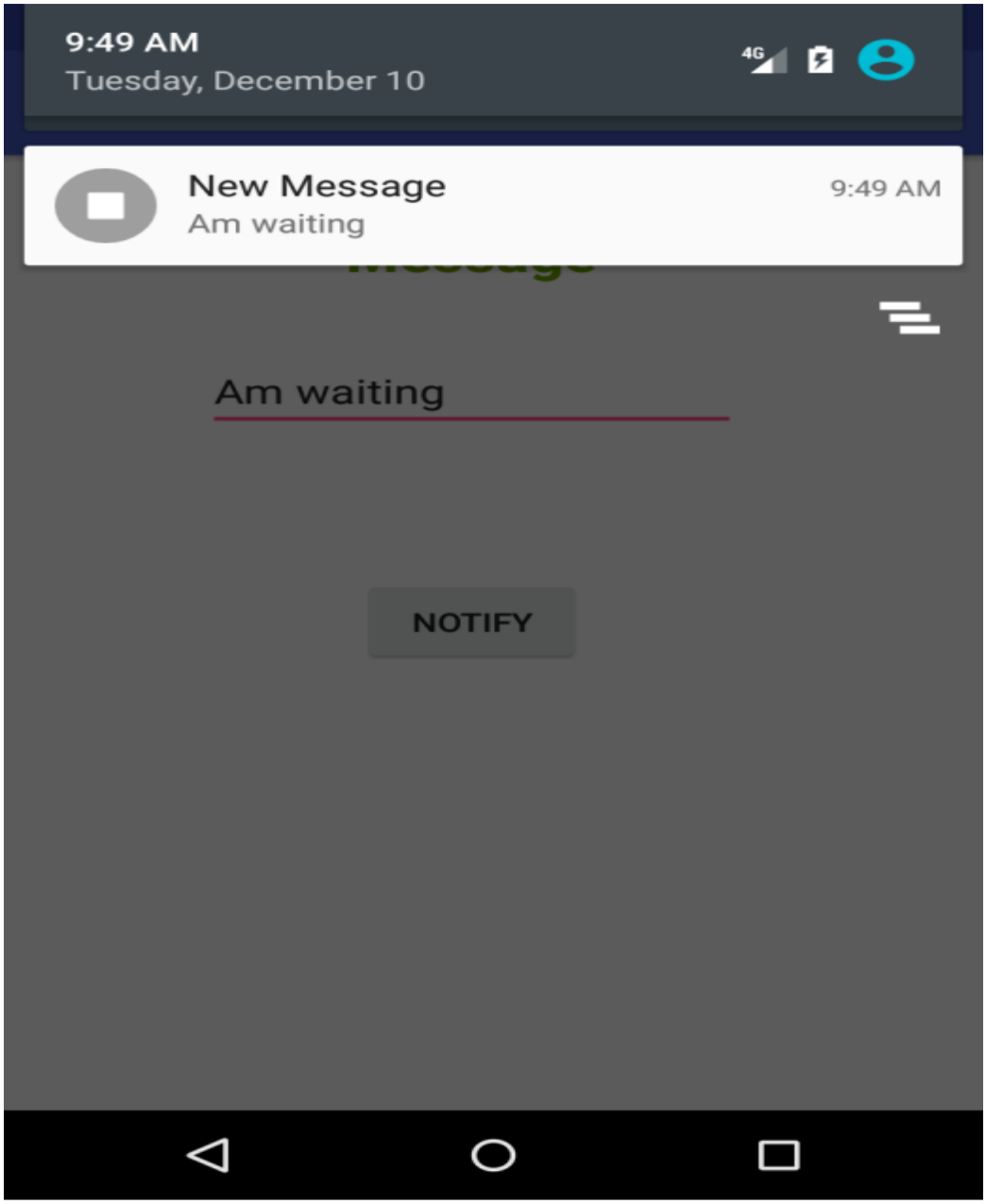
```
packagecom.example.notimgr;

//package com.spaceo.notificationalertdemo
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
class NotifyMessage : AppCompatActivity() {
        override fun onCreate(savedInstanceState: Bundle?) {
super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_notifymessage)
        }
        }
```

SAMPLE OUTPUT:-


9:48
Notification Manager
Message
NOTIFY


9:49
Notification Manager
Message
Am waiting
writing
awaiting
wanting
q w e r t y u i o p
a s d f g h j k l
z x c v b n m
?123


9:49
Notification Manager
Message
Am waiting
NOTIFY



Viva-Voce Questions:-

1)What is meant by notification manager?

2) What is the package imported for this program?

3) How many activities are initiated to do notification manager?

4) What is the differences between public void and private void?

5) What is the super class will do?

PGM:05	Develop a native application that uses GPS location information

.

AIM:-

To Develop a native application that uses GPS location information

PROCEDURE:-

1) Open eclipse or android studio and select new android project

2) Give project name and select next

3) Choose the android version. Choose the lowest android version (Android 2.2) and select next

4) Enter the package name. package name must be two word separated by comma and click finish

5) Go to package explorer in the left hand side. select our project.

6) Go to res folder and select layout. Double click the main.xml file. Add the code given.

7) Now select mainactivity.java file and type the following code. In my coding man activity name is GPS location Activity. Package gps.location;

8) Go to src folder and Right Click on your package folder and choose new class and give the class names as GPS trace

9) Select the GPStrace.java file and paste the given code.

10)Go to manifest.xml file and add the code given

11) Now go to main.xml and right click .select run as option and select run configuration

12) Android output is present in the android emulator as shown in below.

main.xml file:-

```
<?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_cont
ent"
```

```
android:layout_height="wrap_cont
ent
android:text="Show_Location"
android:layout_centerVertical="tr
ue"
android:layout_centerHorizontal="true"
/>
</RelativeLayout>
```

mainactivity.java file :-

```
//import android.R;
import
android.app.Activity;
import
android.os.Bundle;
import
android.view.View;
import
android.widget.Butto
n; 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();
}
}
});
```

```
}
}
```

GPStrace.java file :-

```
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.LocationManage
r; 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=conte
xt;
getLocation();
public Location getLocation()
{
try{
locationManager=(LocationManager) context.getSystemService(LOCATION_SERVICE);
isGPSEnabled=locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER)
;
isNetworkEnabled=locationManager.isProviderEnabled(LocationManager.NETWORK_PR
OVI DER);
if(!isGPSEnabled && !isNetworkEnabled){
}else{
this.canGetLocation=true;
if(isNetworkEnabled){
locationManager.requestLocationUpdates(
LocationManager.NETWORK_PROVIDER,
```

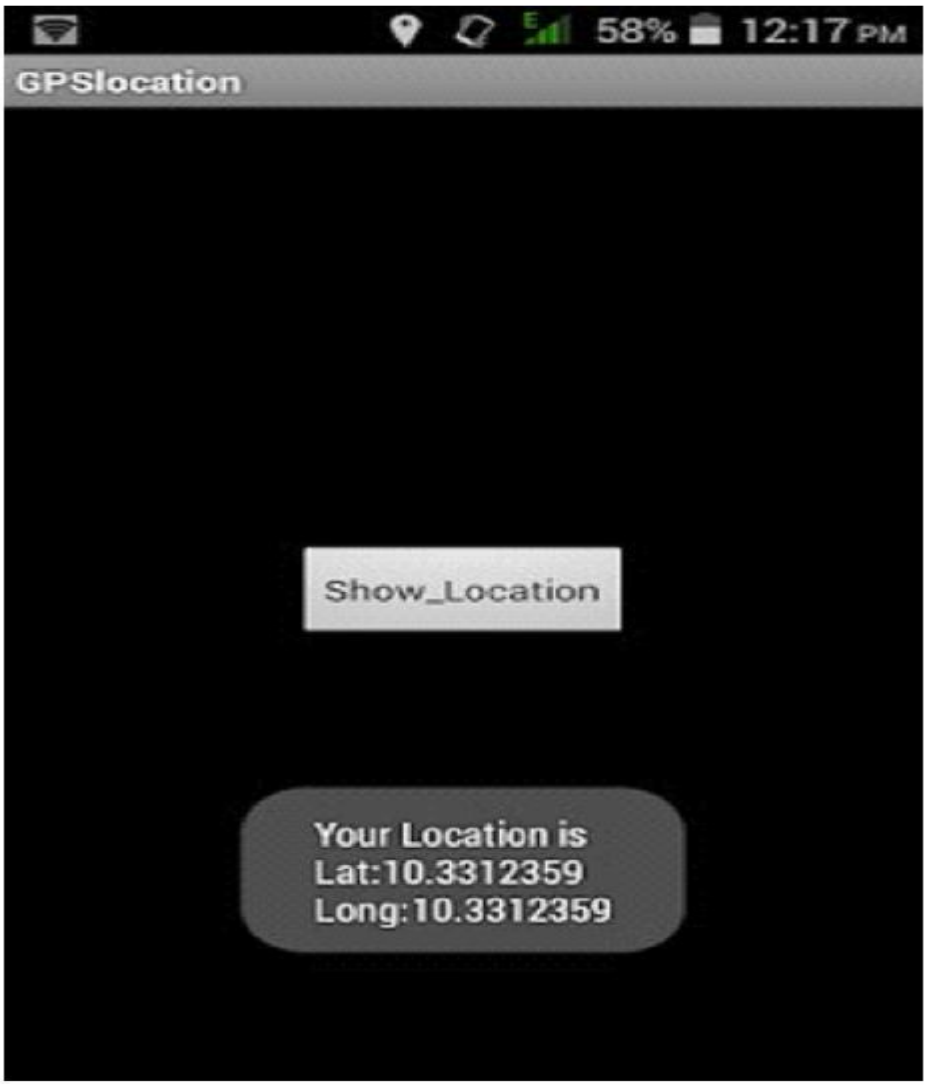
```
MIN_TIME_BW_UPDATES,
MIN_DISTANCE_CHANGE_FOR_UPDATES,this);
}
if(locationManager!=null){
location=locationManager.getLastKnownLocation(LocationManager.NETWORK_PROVI
DER)
;
if(location !=null){
latitude=location.getLatitude();
longtitude=location.getLongitude();
}
}
}
if(isGPSEnabled){
if(location==null){
locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,MIN_TIM
E_B
W_UPDATES, MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
if(locationManager!=null){
location=locationManager.getLastKnownLocation(LocationManager.GPS_PROV
IDER); 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"/>
```

OUTPUT:-



Viva-Voce Questions:-

1) What is the full form of GPS?
2) Which type of layout is used here?
3) Write the syntax of layout?
4) Which type of package is used in this type of application?
5) What is the baud rate used in this GPS location tracer?

PGM:06	Implement an application that for basicCalculator.

AIM:

To Implement an application that for basic Calculator.

PROCEDURE:-

Step 1: Firstly get the android studio downloaded in your system, then open it.

Step 2: Create a new project and name it Calculator.

Step 3: Open res -> layout -> activity_main.xml (or) main.xml. Here we are going to create the application interface like add layouts, Button , TextView and EditText.

i of Step 3 – Create a Linearlayout vertical, add a textview followed by two textfields Number(decimal) for writing numbers in it. Starting code of activity_main.xml

ii of Step 3 – Then before closing the above layout define another layout as Linearlayout horizontal, add five button (+, -, *, / and Clear) define their properties like id , width, height etc. in it and close the linearlayout

iii of Step 3 – Further in continuation with previous linearlayout add a textview, textfield(Number) for displaying result which makes the interface complete.

Step 4: Open src -> package -> MainActivity.java. The interface part of the application is over, let's focus on adding functionality to the application. This calculator app basically perform five operations i.e addition, subtraction, multiplication, division and reset. So for that we need to define these operation over button click. For that we use setOnClickListener() function.parseDouble() is used to convert String value to double. By default the value is String and we need to convert it into Double to perform operation over it.

If person doesn't enter the value and directly click on the any button then a Toast message will appear on the screen telling user to enter the required numbers.

MAIN XML FILE:-

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.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="com.example.a84.calculator.MainActivity">

<RelativeLayout
android:layout_width="400dp"
android:layout_height="495dp"
android:layout_marginBottom="8dp"
android:layout_marginEnd="8dp"
android:layout_marginTop="8dp"
app:layout_constraintBottom_toBottomOf="parent"
```

```
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintTop_toTopOf="parent">

<Button
android:id="@+id/btn_1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true"
android:layout_below="@+id/edText1"
android:layout_marginTop="60dp"
android:onClick="PressOne"
android:text="1"
android:textSize="18sp" />

<Button
android:id="@+id/btn_0"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@+id/btn_8"
android:layout_toEndOf="@+id/btn_7"
android:layout_toRightOf="@+id/btn_7"
android:text="0"
android:textSize="18sp" />

<Button
android:id="@+id/btn_9"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@+id/btn_6"
android:layout_toEndOf="@+id/btn_5"
android:layout_toRightOf="@+id/btn_5"
android:text="9"
android:textSize="18sp" />

<Button
android:id="@+id/btn_8"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@+id/btn_5"
android:layout_toEndOf="@+id/btn_7"
android:layout_toRightOf="@+id/btn_7"
android:text="8"
android:textSize="18sp" />

<Button
android:id="@+id/btn_7"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/btn_4"
```

```
android:layout_alignStart="@+id/btn_4"
android:layout_below="@+id/btn_4"
android:text="7"
android:textSize="18sp" />

<Button
android:id="@+id/btn_6"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/btn_5"
android:layout_alignBottom="@+id/btn_5"
android:layout_toEndOf="@+id/btn_5"
android:layout_toRightOf="@+id/btn_5"
android:text="6"
android:textSize="18sp" />

<Button
android:id="@+id/btn_5"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@+id/btn_2"
android:layout_toEndOf="@+id/btn_4"
android:layout_toRightOf="@+id/btn_4"
android:text="5"
android:textSize="18sp" />

<Button
android:id="@+id/btn_4"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/btn_1"
android:layout_alignStart="@+id/btn_1"
android:layout_below="@+id/btn_1"
android:text="4"
android:textSize="18sp" />

<Button
android:id="@+id/btn_3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/btn_2"
android:layout_alignBottom="@+id/btn_2"
android:layout_toEndOf="@+id/btn_2"
android:layout_toRightOf="@+id/btn_2"
android:text="3"
android:textSize="18sp" />

<Button
android:id="@+id/btn_2"
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/btn_1"
android:layout_alignBottom="@+id/btn_1"
android:layout_toEndOf="@+id/btn_1"
android:layout_toRightOf="@+id/btn_1"
android:text="2"
android:textSize="18sp" />

<Button
android:id="@+id/btn_Add"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_above="@+id/btn_6"
android:layout_alignParentEnd="true"
android:layout_alignParentRight="true"
android:backgroundTint="@android:color/darker_gray"
android:text="+"
android:textColor="@android:color/background_light"
android:textSize="18sp" />

<Button
android:id="@+id/btn_Sub"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/btn_Add"
android:layout_alignStart="@+id/btn_Add"
android:layout_below="@+id/btn_Add"
android:backgroundTint="@android:color/darker_gray"
android:text="-"
android:textColor="@android:color/background_light"
android:textSize="18sp" />

<Button
android:id="@+id/btn_Mul"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/btn_Sub"
android:layout_alignStart="@+id/btn_Sub"
android:layout_below="@+id/btn_6"
android:backgroundTint="@android:color/darker_gray"
android:text="*"
android:textColor="@android:color/background_light"
android:textSize="18sp" />

<Button
android:id="@+id/btn_Div"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/btn_Mul"
android:layout_alignStart="@+id/btn_Mul"
```

```
android:layout_below="@+id/btn_9"
android:backgroundTint="@android:color/darker_gray"
android:text="/"
android:textColor="@android:color/background_light"
android:textSize="18sp" />

<EditText
android:id="@+id/edText1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentEnd="true"
android:layout_alignParentLeft="true"
android:layout_alignParentRight="true"
android:layout_alignParentStart="true"
android:layout_alignParentTop="true"
android:layout_marginTop="80dp"
android:ems="10"
android:inputType="textPersonName"
android:textAlignment="textEnd"
android:textSize="24sp" />

<Button
android:id="@+id/btn_calc"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@+id/btn_0"
android:layout_toEndOf="@+id/btn_0"
android:layout_toRightOf="@+id/btn_0"
android:backgroundTint="@android:color/holo_green_light"
android:text="="
android:textColor="@android:color/background_light"
android:textSize="18sp" />

<Button
android:id="@+id/btn_dec"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@+id/btn_7"
android:layout_toLeftOf="@+id/btn_8"
android:layout_toStartOf="@+id/btn_8"
android:text="."
android:textSize="18sp" />

<Button
android:id="@+id/btn_clear"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentEnd="true"
android:layout_alignParentRight="true"
android:layout_below="@+id/btn_Div"
```

```
android:backgroundTint="@android:color/holo_blue_dark"
android:text="clear"
android:textColor="@android:color/background_light"
android:textSize="18sp" />

</RelativeLayout>
</android.support.constraint.ConstraintLayout>
```

Java file :-

```
package com.example.p4;

//import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.app.Activity;

public class MainActivity extends Activity {
   Button
btn_1,btn_2,btn_3,btn_4,btn_5,btn_6,btn_7,btn_8,btn_9,btn_0,btn_Add,btn_Sub,btn_
Mul,btn_Div,btn_calc,btn_dec,
btn_clear;
   EditText ed1;
float Value1, Value2;
boolean mAddition, mSubtract, mMultiplication, mDivision ;

@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);

btn_0 = (Button) findViewById(R.id.btn_0);
btn_1 = (Button) findViewById(R.id.btn_1);
btn_2 = (Button) findViewById(R.id.btn_2);
btn_3 = (Button) findViewById(R.id.btn_3);
btn_4 = (Button) findViewById(R.id.btn_4);
btn_5 = (Button) findViewById(R.id.btn_5);
btn_6 = (Button) findViewById(R.id.btn_6);
btn_7 = (Button) findViewById(R.id.btn_7);
btn_8 = (Button) findViewById(R.id.btn_8);
btn_9 = (Button) findViewById(R.id.btn_9);
btn_Add = (Button) findViewById(R.id.btn_Add);
btn_Div = (Button) findViewById(R.id.btn_Div);
btn_Sub = (Button) findViewById(R.id.btn_Sub);
btn_Mul = (Button) findViewById(R.id.btn_Mul);
```

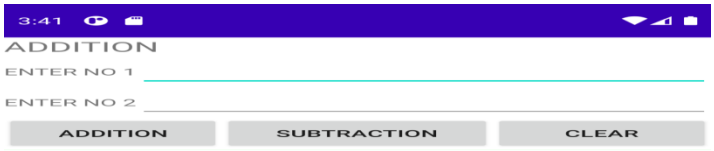
```
btn_calc = (Button) findViewById(R.id.btn_calc);
btn_dec = (Button) findViewById(R.id.btn_dec);
btn_clear = (Button) findViewById(R.id.btn_clear);
ed1 = (EditText) findViewById(R.id.edText1);

btn_0.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
ed1.setText(ed1.getText()+"0");
        }
    });

btn_1.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
ed1.setText(ed1.getText()+"1");
        }
    });

btn_2.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
ed1.setText(ed1.getText()+"2");
        }
    });

btn_3.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
ed1.setText(ed1.getText()+"3");
        }
    });

btn_4.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
ed1.setText(ed1.getText()+"4");
        }
    });

btn_5.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
ed1.setText(ed1.getText()+"5");
        }
    });

btn_6.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
```

```
ed1.setText(ed1.getText()+"6");
        }
    });

btn_7.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
ed1.setText(ed1.getText()+"7");
        }
    });

btn_8.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
ed1.setText(ed1.getText()+"8");
        }
    });

btn_9.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
ed1.setText(ed1.getText()+"9");
        }
    });

btn_dec.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
ed1.setText(ed1.getText()+".");
        }
    });

btn_Add.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {

if (ed1 == null){
ed1.setText("");
            }else {
Value1 = Float.parseFloat(ed1.getText() + "");
mAddition = true;
ed1.setText(null);
            }
        }
    });

btn_Sub.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
Value1 = Float.parseFloat(ed1.getText() + "");
```

```
mSubtract = true ;
ed1.setText(null);
        }
    });

btn_Mul.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
Value1 = Float.parseFloat(ed1.getText() + "");
mMultiplication = true ;
ed1.setText(null);
        }
    });

btn_Div.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
Value1 = Float.parseFloat(ed1.getText()+"");
mDivision = true ;
ed1.setText(null);
        }
    });

btn_calc.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
Value2 = Float.parseFloat(ed1.getText() + "");

if (mAddition == true){

ed1.setText(Value1 + Value2 +"");
mAddition=false;
            }


if (mSubtract == true){
ed1.setText(Value1 - Value2 +"");
mSubtract=false;
            }

if (mMultiplication == true){
ed1.setText(Value1 * Value2 + "");
mMultiplication=false;
            }

if (mDivision == true){
ed1.setText(Value1 / Value2+"");
mDivision=false;
            }
        }
```

```
        });

btn_clear.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
ed1.setText("");
            }
        });
    }

}
```

OUTPUT:-



Viva-Voce Questions:-

1) Which type of data is entered in the calculator?
2) How many text fields are created to enter data?
3) Which type of layouts are used here?
4) Write the syntax for Relative layout?
5) Is this calculator is useful for fractional and exponent numbers operations?

PGM:07	Implement an application that creates an alert upon receiving a message

AIM:-

T0 Implement an application that creates an alert upon receiving a message

PROCEDURE:-

1) Open eclipse or android studio and select new android project

2) Give project name and select next

3) Choose the android version. Choose the lowest android version(Android 2.2) and select next

4) Enter the package name. package name must be two word separated by comma and click finish

5) Go to package explorer in the left hand side. select our project.

6) Go to res folder and select layout. Double click the main.xml file. Add the code given.

7) Now select mainactivity.java file and type the following code. In my coding man activity name is Alert1Activity.

8) Now go to main.xml and right click .select run as option and select run configuration

MAIN XML FILE:-

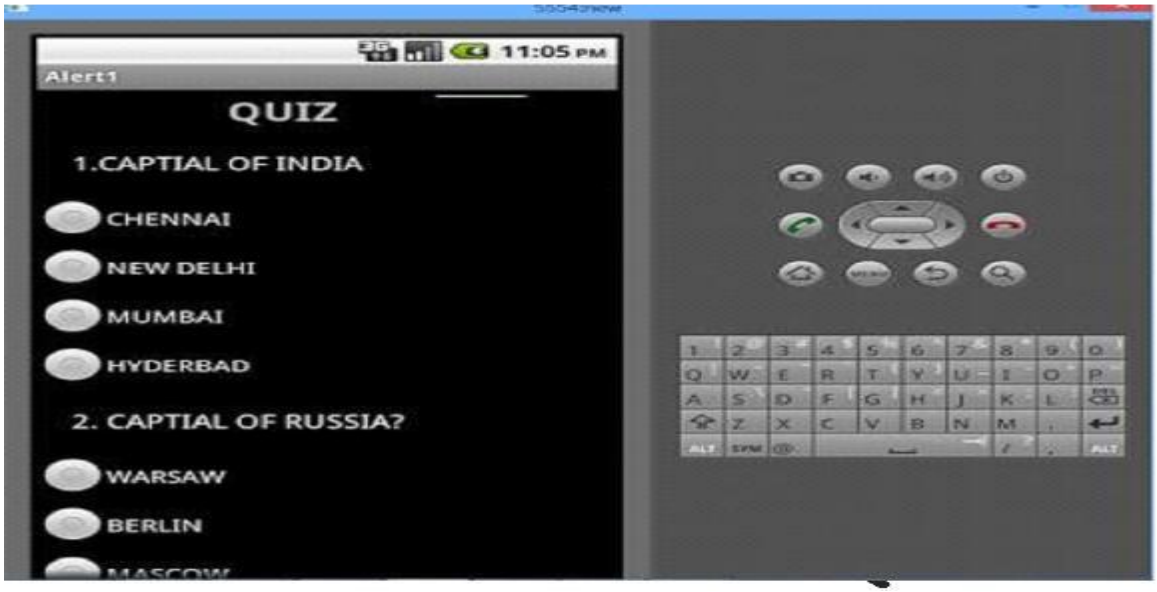
```
<ScrollView
xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:scrollbars="vertical" >
<TableLayout
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:shrinkColumns="*"
android:stretchColumns="*"
android:background="#000000">
<TableRow
android:layout_height="wrap_content"
android:layout_width="match_parent"
android:gravity="center_horizontal">
<TextView
android:id="@+id/Title"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:layout_margin="5px"
android:focusable="false"
android:focusableInTouchMode="false"
```

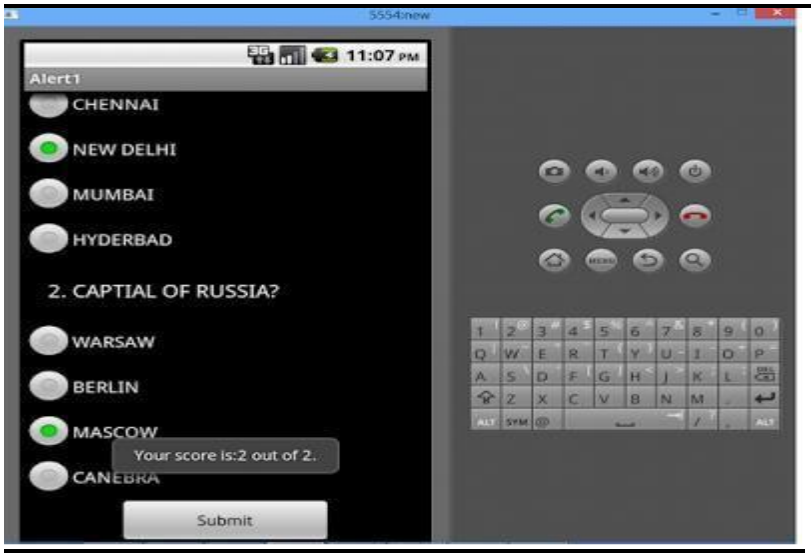
```
android:gravity="center_vertical|center_hori
zontal" android:text="QUIZ"
android:textSize="25sp"
android:textStyle="bold" />
<View
android:layout_height="2px"
android:layout_marginTop="5dip"
android:layout_marginBottom="5dip"
android:background="#DDFFDD"/>
</TableRow>
<TableRow
android:layout_height="wrap_content"
android:layout_width="match_parent"
android:gravity="center_horizontal">
<TextView
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:textSize="18sp"
android:text="1.CAPTIAL OF INDIA"
android:layout_span="4"
android:padding="18dip"
android:textColor="#ffffff"/>
</TableRow><TableRow
android:id="@+id/tableRow1"
android:layout_height="wrap_content"
android:layout_width="match_parent">
<RadioGroup
android:id="@+id/answer1"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="0.4" >
<RadioButton
android:id="@+id/answer1A"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:textColor="#ffffff"
android:text="CHENNAI" />
<RadioButton
android:id="@+id/answer1B"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:textColor="#ffffff"
android:text="NEW DELHI" />
<RadioButton
android:id="@+id/answer1C"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:textColor="#ffffff"
android:text="MUMBAI" />
<RadioButton
```

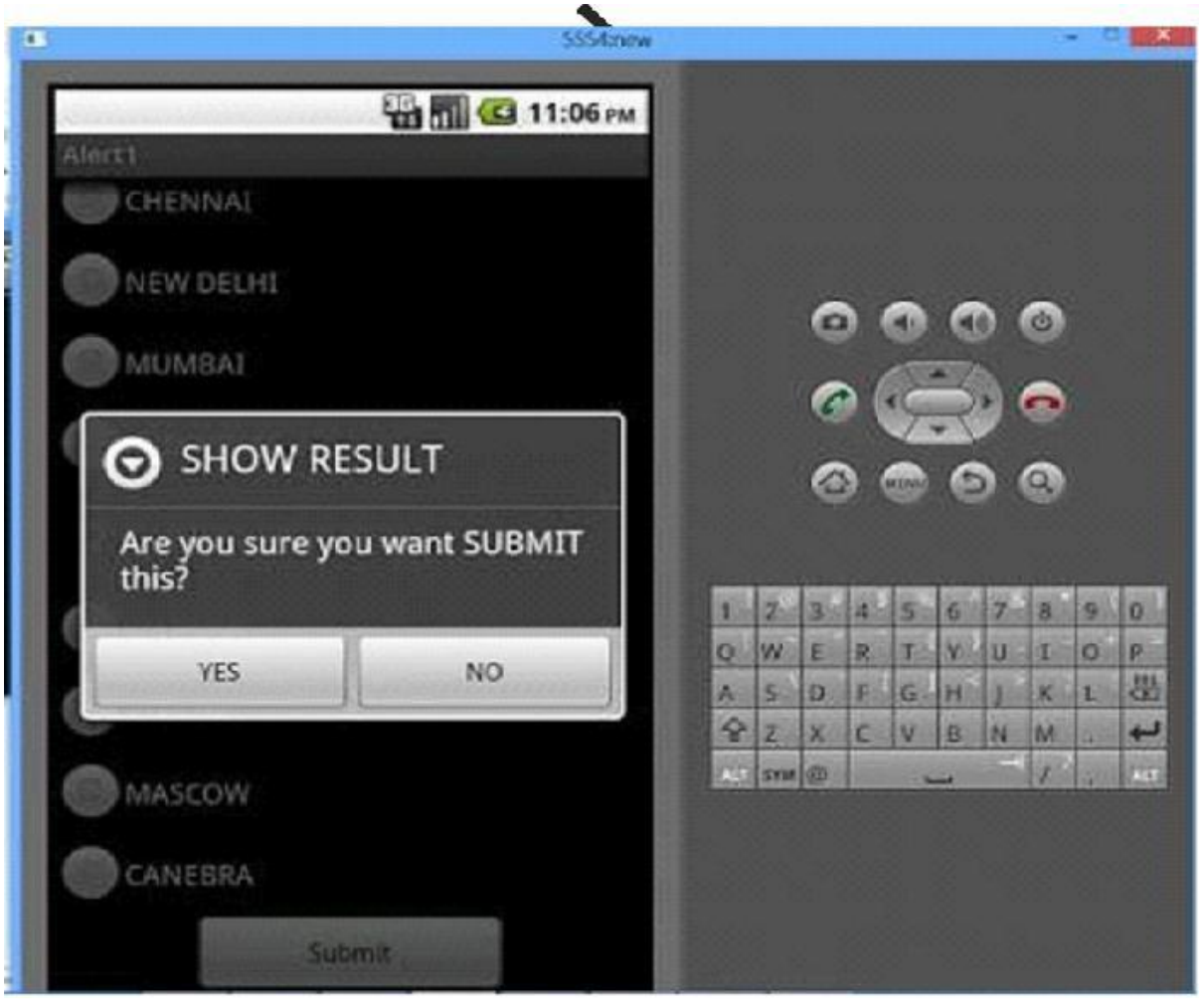
```
android:id="@+id/answer1D"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:textColor="#ffffff"
android:text="HYDERBAD" />
</RadioGroup>
</TableRow><TableRow
android:layout_height="wrap_content"
android:layout_width="match_parent"
android:gravity="center_horizontal">
<TextView
android:layout_width="match_parent" android:layout_height="wrap_content"
android:textSize="18sp" android:text="2. CAPTIAL OF
RUSSIA?" android:layout_span="4"
android:padding="18dip" android:textColor="#ffffff"/>
</TableRow>
<TableRow
android:id="@+id/tableRow2"
android:layout_height="wrap_content"
android:layout_width="match_parent">
<RadioGroup
android:id="@+id/answer2"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_weight="0.4" >
<RadioButton
android:id="@+id/answer2A"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:textColor="#ffffff"
android:text="WARSAW " />
<RadioButton
android:id="@+id/answer2B"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:textColor="#ffffff"
android:text="BERLIN" />
<RadioButton
android:id="@+id/answer2C"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:textColor="#ffffff"
android:text="MASCOW " />
<RadioButton
android:id="@+id/answer2D"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:textColor="#ffffff"
android:text="CANEBRA " />
</RadioGroup>
```

```
</TableRow><TableRow
android:layout_height="wrap_content"
android:layout_width="match_parent"
android:gravity="center_horizontal">
<Button
android:id="@+id/submit"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:gravity="center"
android:text="Submit" />
</TableRow>
</TableLayout></ScrollView>
```

OUTPUT:-







Viva-Voce Questions:-

1) What is the layout used in this application program.
2) Which package is used in this?
3) Write the syntax for Radio button?
4) What is the purpose of scroll view in this program?
5) Explain the logic behind of this application development.

PGM:08	Write a mobile application that makes use of RSS feed

AIM :-

To develop an Android Application that makes use of RSS Feed.

PROCEDURE:-

Step 1: Create an android project using android studio.

Step 2: Design

2.1. Open the actual Project folder (app) in Android Studio IDE.

2.2. Click res directory -> layout ->**activity_main.xml** -> Design.

2.3. Insert the GUI components to **Design view** in activity_main.xml.

2.4. Enter the id for each component.

Step 3: Open **AndroidManifest.xml** and add the code to provide access permission.

Step 4: Open Java ->**MainActivity.java** and add the code.

Step 5: Run the project and the output will be displayed in Emulator.

LAYOUT DESIGN:-

MAIN XML FILE:-

```
<?xmlversion="1.0"encoding="utf-8"?>
<LinearLayoutxmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical">

    <ListView
        android:id="@+id/listView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"/>

</LinearLayout>
```

Adding permissions in Manifest for the Android Application:

- Click on **app -> manifests -> AndroidManifest.xml**
- Now include the **INTERNET** permissions in the AndroidManifest.xml file as shown below
-

AndroidManifest.xml:

```
<?xmlversion="1.0"encoding="utf-8"?>
<manifestxmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.exno6">

    <uses-permissionandroid:name="android.permission.INTERNET"/>

    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activityandroid:name=".MainActivity">
            <intent-filter>
                <actionandroid:name="android.intent.action.MAIN"/>

                <categoryandroid:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
        </activity>
    </application>

</manifest>
```

- So now the Permissions are added in the Manifest.

Java Coding for the Android Application:

- Click on **app -> java -> com.example.** RSSFEED1 **-> MainActivity.**
- Then delete the code which is there and type the code as given below.
-
- ***Code for MainActivity.java:***

```
packagecom.example.RSSFEED1;

importandroid.app.ListActivity;
importandroid.content.Intent;
importandroid.net.Uri;
importandroid.os.AsyncTask;
importandroid.os.Bundle;
importandroid.view.View;
importandroid.widget.ArrayAdapter;
importandroid.widget.ListView;
importorg.xmlpull.v1.XmlPullParser;
importorg.xmlpull.v1.XmlPullParserException;
importorg.xmlpull.v1.XmlPullParserFactory;
importjava.io.IOException;
importjava.io.InputStream;
importjava.net.MalformedURLException;
importjava.net.URL;
importjava.util.ArrayList;
importjava.util.List;

publicclassMainActivity extendsListActivity
{
   List headlines;
   List links;

   @Override
   protectedvoidonCreate(Bundle savedInstanceState)
   {
      super.onCreate(savedInstanceState);
      newMyAsyncTask().execute();
   }

   classMyAsyncTask extendsAsyncTask<Object,Void,ArrayAdapter>
   {
      @Override
      protectedArrayAdapter doInBackground(Object[] params)
```

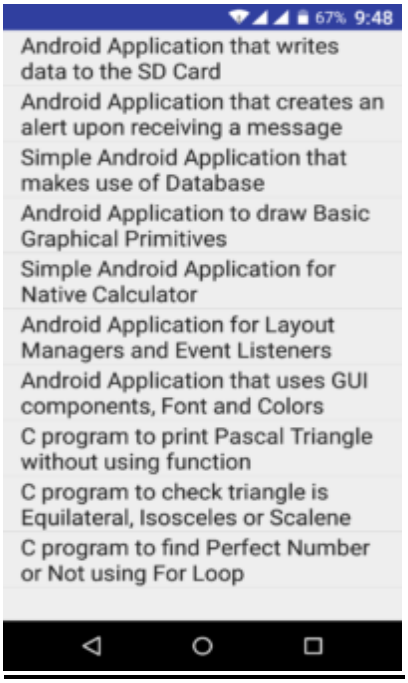
```
    {
        headlines = newArrayList();
        links = newArrayList();
        try
        {
            URL url = newURL("https://codingconnect.net/feed");
            XmlPullParserFactory factory = XmlPullParserFactory.newInstance();
            factory.setNamespaceAware(false);
            XmlPullParser xpp = factory.newPullParser();

            // We will get the XML from an input stream
            xpp.setInput(getInputStream(url), "UTF_8");
            booleaninsideItem = false;

            // Returns the type of current event: START_TAG, END_TAG, etc..
            inteventType = xpp.getEventType();
            while(eventType != XmlPullParser.END_DOCUMENT)
            {
                if(eventType == XmlPullParser.START_TAG)
                {
                    if(xpp.getName().equalsIgnoreCase("item"))
                    {
                        insideItem = true;
                    }
                    elseif(xpp.getName().equalsIgnoreCase("title"))
                    {
                        if(insideItem)
                            headlines.add(xpp.nextText()); //extract the headline
                    }
                    elseif(xpp.getName().equalsIgnoreCase("link"))
                    {
                        if(insideItem)
                            links.add(xpp.nextText()); //extract the link of article
                    }
                }
                elseif(eventType==XmlPullParser.END_TAG && xpp.getName().
equalsIgnoreCase("item"))
                {
                    insideItem=false;
                }
                eventType = xpp.next(); //move to next element
            }
```

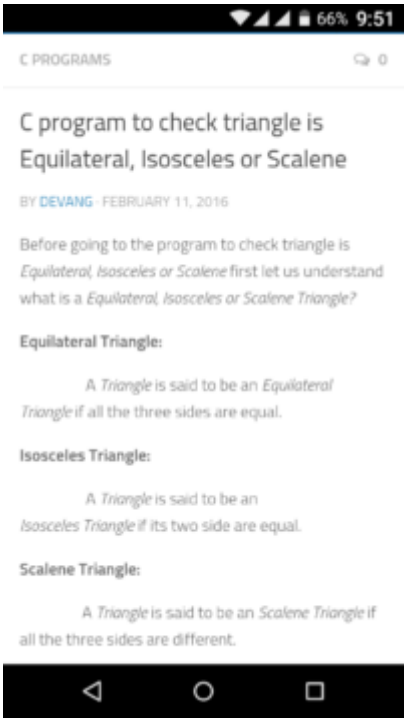
```
            }
            catch(MalformedURLException e)
            {
                e.printStackTrace();
            }
            catch(XmlPullParserException e)
            {
                e.printStackTrace();
            }
            catch(IOException e)
            {
                e.printStackTrace();
            }
            returnnull;
        }
        protectedvoidonPostExecute(ArrayAdapter adapter)
  adapter = newArrayAdapter(MainActivity.this, android.R.layout.simple_list_item_1,
headlines);
            setListAdapter(adapter);
        }
    }

    @Override
    protectedvoidonListItemClick(ListView l, View v, intposition, longid)
    {
        Uri uri = Uri.parse((links.get(position)).toString());
        Intent intent = newIntent(Intent.ACTION_VIEW, uri);
        startActivity(intent);
    }

    publicInputStream getInputStream(URL url)
    {
        try
        {
            returnurl.openConnection().getInputStream();
        }
        catch(IOException e)
        {
            returnnull;
        }
    }
```

```
}
```

- So now the Coding part is also completed.
- Now run the application to see the output.

OUTPUT:-





Viva-Voce Questions:-

1) Where the permissionsare added for the Android Application?
2) In which file, including the **INTERNET** permissions?
3) What is the full form of RSS feed?
4) Specify the purpose of RSS feed.
5) Which type of communication is possible with RSS feed connection?

PGM:09 | Develop a mobile application to send an email.

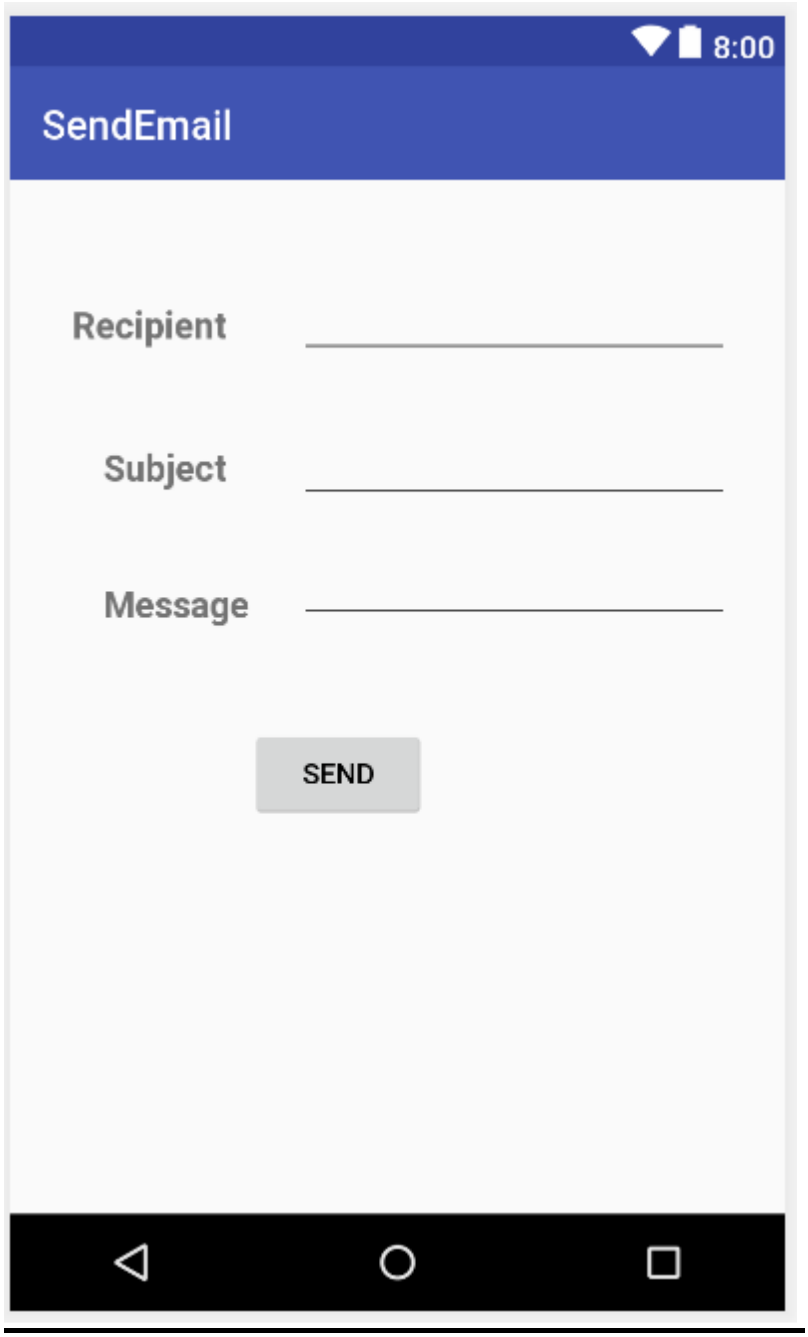
AIM:-

To develop an Android Application to send an Email.

PROCEDURE :-

Step 1: Create an android project using android studio.
Step 2: Design
2.1. Open the actual Project folder (app) in Android Studio IDE.
2.2. Click res directory -> layout ->**activity_main.xml** -> Design.
2.3. Insert the GUI components to **Design view** in activity_main.xml.
2.4. Enter the id for each component.
Step 3: Open **AndroidManifest.xml** and add the code to provide access permission.
Step 4: Open **res/values/styles.xml** and add the code.
Step 5: Open Java ->**MainActivity.java** and add the code.
Step 6: Run the project and the output will be displayed in Emulator.

LAYOUT DESIGN:-



***MAIN XML FILE*:-**

```
<?xmlversion="1.0" encoding="utf-8"?>
<!--Relative Layout-->
<RelativeLayout
xmlns:androclass="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">
<!--Edit text for email id-->
<EditText
android:id="@+id/editText1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentRight="true"
android:layout_alignParentTop="true"
android:layout_marginRight="22dp"
android:layout_marginTop="18dp"
/>
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@+id/editText1"
android:layout_alignLeft="@+id/editText1"
android:layout_marginTop="20dp"
/>
<!--Edit text for email body-->
<EditText
android:id="@+id/editText3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="30dp"
android:layout_below="@+id/editText2"
android:layout_alignLeft="@+id/editText2"
/>
<!--text Views for label-->
<TextView
android:id="@+id/textView1"
android:textColor="#0F9D58"
android:text="Send To:"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/editText1"
android:layout_alignBottom="@+id/editText1"
android:layout_alignParentLeft="true"
/>
<TextView
android:id="@+id/textView2"
android:textColor="#0F9D58"
android:text="Email Subject:"
android:layout_width="wrap_content"
```

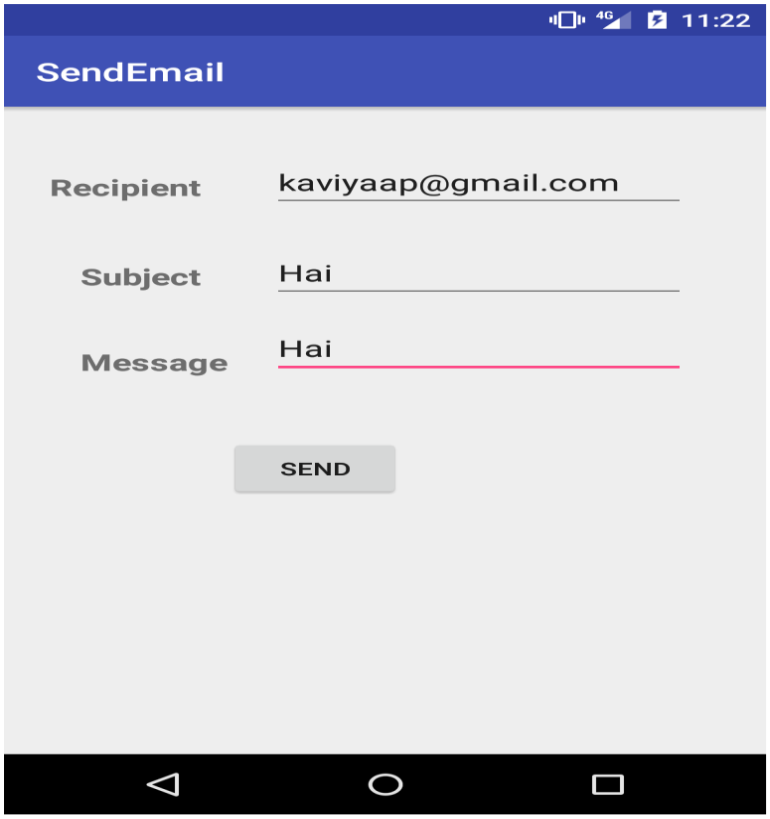
```
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/editText2"
android:layout_alignBottom="@+id/editText2"
android:layout_alignParentLeft="true"
/>
<TextView
android:id="@+id/textView3"
android:textColor="#0F9D58"
android:text="Email Body:"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/editText3"
android:layout_alignBottom="@+id/editText3"
/>
android:id="@+id/button"
android:text="Send email!!"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/editText3"
android:layout_below="@+id/editText3"
android:layout_marginLeft="76dp"
android:layout_marginTop="20dp"
/>
</RelativeLayout>
```

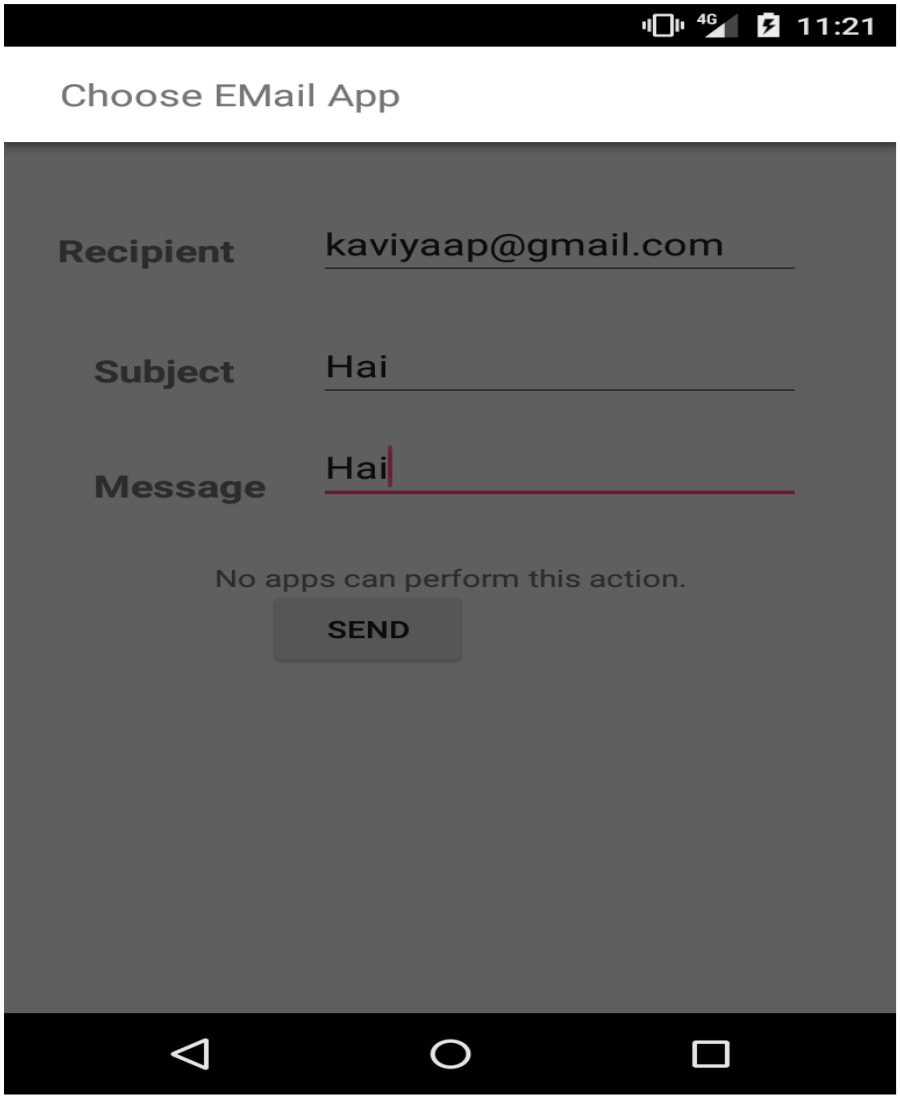
MAIN JAVA FILE:-

```
package com.geeksforgeeks.phonecall;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.content.Intent;
import android.widget.EditText;
import android.view.View;
import android.view.View.OnClickListener;
import android.net.Uri;
import android.widget.Button;
public class MainActivity extends AppCompatActivity {
// define objects for edit text and button
Button button;
EditText sendto, subject, body;

@Override
protected void onCreate(Bundle savedInstanceState)
{
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
// Getting instance of edittext and button
sendto = findViewById(R.id.editText1);
subject = findViewById(R.id.editText2);
body = findViewById(R.id.editText3);
```

```
button = findViewById(R.id.button);
// attach setOnClickListener to button
// with Intent object define in it
button.setOnClickListener(new OnClickListener() {
@Override
public void onClick(View view)
{
String emailsend = sendto.getText().toString();
String emailsubject = subject.getText().toString();
String emailbody = body.getText().toString();
// define Intent object
// with action attribute as ACTION_SEND
Intent intent = new Intent(Intent.ACTION_SEND);
// add three fiels to intent using putExtra function
intent.putExtra(Intent.EXTRA_EMAIL,
newString[] { emailsend });
intent.putExtra(Intent.EXTRA_SUBJECT, emailsubject);
intent.putExtra(Intent.EXTRA_TEXT, emailbody);
// set type of intent
intent.setType("message/rfc822");
// startActivity with intent with chooser
// as Email client using createChooser function
startActivity(
Intent
.createChooser(intent,
"Choose an Email client :"));
}
});
}
}
```

OUTPUT:-


11:22
SendEmail
Recipient
kaviyaap@gmail.com
Subject
Hai
Message
Hai
SEND



Viva-Voce Questions:-

1) What is the layout used in this?
2) What is the logic behind in development of application for an email ?
3) List different fields specified in development of an email.
4) Specify the different methods used in this.
5) List out the packages used in development of application for an email ?

PGM:10	**WRITE A MOBILE APPLICATION THAT CREATES ALARM CLOCK** (Mini Project)

AIM:-

To write a mobile application that creates alarmclock

PROCEDURE:-

1. Android Manifest--AndroidManifest.xml

We need to give uses-permission for WAKE_LOCK, other than that the AndroidManifest.xml is pretty standard one. Just need to include the service and receiver.

2. Android Activity--activity_my.xml

The Android Activity is designed to be simple. We have a Time Picker component followed by a Toggle Button. That's it. Choose the time to set the alarm and toggle the switch to on. The alarm will work.

3. Alarm Receiver--AlarmReceiver.java

this is the one that receives the alarm trigger on set time. From here we initiate different actions to notify the user as per our choice. I have given three type of notifications, first show a message to user in the activity UI, second play the alarm ringtone and third send an Android notification message. So this is the place to add enhancement for different types of user notifications.

4. Alarm Notification Message --AlarmService.java

The receiver will start the following Intent Service to send a standard notification to the user.

AndroidManifest.xml:-

```
<?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>
```

activity_my.xml FILE:-

```
<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>
```

AlarmReceiver.java FILE:-

```
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);
}
}
```

```
package com.javapapers.androidalarmclock;
import android.app.Activity; import
android.content.ComponentName; import
android.content.Context; import
android.content.Intent; import
android.media.Ringtone; import
android.media.RingtoneManager; import
android.net.Uri;
import android.support.v4.content.WakefulBroadcastReceiver; public class
AlarmReceiver extends WakefulBroadcastReceiver {
@Override
public void onReceive(final Context context, Intent intent) {
//this will update the UI with message AlarmActivity inst =
```

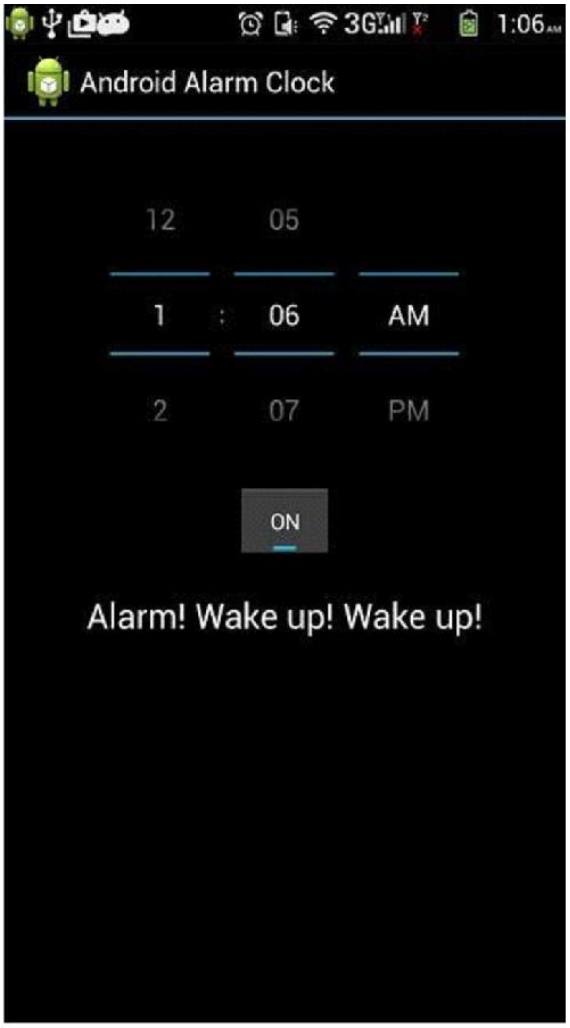
```
AlarmActivity.instance(); inst.setAlarmText("Alarm! Wake up!
Wake up!");
//this will sound the alarm tone
//this will sound the alarm once, if you wish to
//raise alarm in loop continuously then use MediaPlayer and setLooping(true)
Uri alarmUri =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_ALARM
); if (alarmUri == null) { alarmUri =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
}
Ringtone ringtone = RingtoneManager.getRingtone(context, alarmUri); ringtone.play();
//this will send a notification message
ComponentName comp = new ComponentName(context.getPackageName(),
AlarmService.class.getName());
startWakefulService(context, (intent.setComponent(comp)));
setResultCode(Activity.RESULT_OK);
}
}
```

AlarmService.java FILE:-

```
package com.javapapers.androidalarmclock;
import android.app.IntentService; import
android.app.NotificationManager; import
android.app.PendingIntent; import
android.content.Context; import
android.content.Intent;
import android.support.v4.app.NotificationCompat; import
android.util.Log;
public class AlarmService extends IntentService { private
NotificationManager alarmNotificationManager;
public AlarmService() {
super("AlarmService"); }
@Override
public void onHandleIntent(Intent intent) { sendNotification("Wake Up! Wake
Up!");
}
private void sendNotification(String msg) {
Log.d("AlarmService", "Preparing to send notification...: " + msg);
alarmNotificationManager = (NotificationManager) this
.getSystemService(Context.NOTIFICATION_SERVICE);
PendingIntent contentIntent = PendingIntent.getActivity(this, 0, new Intent(this,
AlarmActivity.class), 0);
NotificationCompat.Builder alamNotificationBuilder = new
NotificationCompat.Builder(this).setContentTitle("Alarm").setSmallIcon(R.drawable.ic_laun
cher)
.setStyle(new NotificationCompat.BigTextStyle().bigText(msg))
.setContentText(msg);
alamNotificationBuilder.setContentIntent(contentIntent);
alarmNotificationManager.notify(1, alamNotificationBuilder.build());
```

```
Log.d("AlarmService", "Notification sent."); }
}
```

OUTPUT:-



Viva-Voce Questions:-

1) List out the different activities in the application development of alaram .
2) Which type of time format is used in clock?
3) Which type of clock is generated (12Hrs/24Hrs)
4) List out the classes and methods used?
5) List out the different types of packages used?
6) Write the syntax of package.