SRM VALLIAMMAI ENGINEERING COLLEGE

(An Autonomous Institution)

SRM Nagar, Kattankulathur-603203.

Department of Information Technology

Lab Manual

1904610- MOBILE APPLICATION DEVELOPMENT LABORATORY

(VI Semester- Regulation 2019)

Prepared By

Dr. K. Revathi, Associate Professor / IT

Ms. S. Jonisha, Assistant Professor (O.G) / IT

Vision of the Institute

"Educate to Excel in Social Transformation"

To accomplish and maintain international eminence and become a model institution for higher learning through dedicated development of minds, advancement of knowledge and professional application of skills to meet the global demands.

Mission of the Institute

- To contribute to the development of human resources in the form of professional engineers and managers of international excellence and competence with high motivation and dynamism, who besides serving as ideal citizen of our country will contribute substantially to the economic development and advancement in their chosen areas of specialization.
- To build the institution with international repute in education in several areas at several levels with specific emphasis to promote higher education and research through strong institute-industry interaction and consultancy.

VISION OF THE DEPARTMENT

To become a model for higher learning through development to prepare self-disciplined, creative culturally competent and dynamic Information Technocrats while remaining sensitive to ethical, societal and environmental issues.

MISSION OF THE DEPARTMENT

M1: To mould the students as innovative and high quality IT professionals to meet the global challenges and Entrepreneurs of international excellence as global leaders capable of contributing towards technological innovations learning process, participation citizenship in their neighborhood and economic growth.

M2: To impart value-based IT education to the students and enrich their knowledge and to achieve effective interaction between industry and institution for mutual benefits.

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

1. To ensure graduates will be proficient in utilizing the fundamental knowledge of basic sciences, mathematics and Information Technology for the applications relevant to various streams of Engineering and Technology.
2. To enrich graduates with the core competencies necessary for applying knowledge of computers and telecommunications equipment to store, retrieve, transmit, manipulate and analyze data in the context of business enterprise.
3. To enable graduates to think logically, pursue lifelong learning and will have the capacity to understand technical issues related to computing systems and to design optimal solutions.
4. To enable graduates to develop hardware and software systems by understanding the importance of social, business and environmental needs in the human context.
5. To enable graduates to gain employment in organizations and establish themselves as professionals by applying their technical skills to solve real world problems and meet the diversified needs of industry, academia and research.

PROGRAM OUTCOMES (POs) ENGINEERING GRADUATES WILL BE ABLE TO:

1. **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
2. **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
3. **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.
4. **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.
5. **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.

6. **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.
7. **Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
8. **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
9. **Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
10. **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 give and receive clear instructions.
11. **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 a team, to manage projects and in multidisciplinary environments.
12. **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.

PROGRAM SPECIFIC OBJECTIVES (PSOs)

1. Design secured database applications involving planning, development and maintenance using state of the art methodologies based on ethical values.
2. Design and develop solutions for modern business environments coherent with the advanced technologies and tools.
3. Design, plan and setting up the network that is helpful for contemporary business environments using latest hardware components.
4. Planning and defining test activities by preparing test cases that can predict and correct errors ensuring a socially transformed product catering all technological needs.

1904610 MOBILE APPLICATION DEVELOPMENT LAB L T P C

0 0 4 2

OBJECTIVES:

- Know the components and structure of mobile application development frameworks for Android and windows OS based mobiles.
- Understand how to work with various mobile application development frameworks.
- Learn the basic and important design concepts and issues of development of mobile applications.
- Understand the capabilities and limitations of mobile devices.
- To use engineering, physics & mathematical concepts critical to mobile application development

LIST OF EXPERIMENTS:

1. Develop an android application to work on GUI components
2. Write an android application to develop dice roller
3. Write an android application to develop Native Calculator
4. Write an android application to develop a Reminder App.
5. Develop an android application to implement Multithreading
6. Write an android application to implement all the UI design (Widgets, Layouts, UI Events and Event Listeners).
7. Develop an android application to make use of Database Concepts
8. Implement an android application to use 2D graphics
9. Develop an android application to implement multimedia (Audio playback and Media Player).
10. Develop an android application to make use of Networking Concept.
    I. Making Phone call.
    II. Sending Emails.
    III. Sending SMS

TOTAL: 60 PERIODS

OUTCOMES:

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

- Design and Implement various mobile applications using emulators
- Deploy applications to hand-held devices
- Transfer current knowledge to learning of new technologies
- Have Skills to software development, computer programming & graphic
- Design using appropriate and accessible digital tools for research and learning.

SOFTWARE REQUIREMENTS:

Eclipse, Android Studio

HARDWARE REQUIREMENTS:

Standalone desktops - 30 Nos. (or) Server supporting 30 terminals or more

COURSE NAME: 1904610 MOBILE APPLICATION DEVELOPMENT

LAB YEAR OF STUDY: 2024-2025

COURSE OUTCOMES:

1904610.1	Design and Implement various mobile applications using emulators
1904610.2	Deploy applications to hand-held devices
1904610.3	Transfer current knowledge to learning of new technologies
1904610.4	Have Skills to software development, computer programming & graphic
1904610.5	Design using appropriate and accessible digital tools for research and learning

CO-PO Matrix:

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
1904610.1	-	-	-	-	3	-	-	-	-	-	-	-
1904610.2	1	-	2	-	-	-	-	-	-	-	-	-
1904610.3	1	-	-	-	-	-	-	-	-	-	-	-
1904610.4	-	-	2	-	-	-	-	-	-	-	-	-
1904610.5	-	-	-	-	3	-	-	-	-	-	-	-

Justification:

Course Outcome	Program Outcome	Value	Justification
1904610.1	PO5	3	Creating the mobile application using Eclipse/Android Studio and deploying it in emulator.
1904610.2	PO1	1	Applying the basic knowledge for the inclusion of mobile application features in the hand-held devices.
	PO3	2	Designing the mobile applications for the hand-held devices which serve the needs of the users.
1904610.3	PO1	1	Applying the knowledge of developing mobile applications in advanced platforms/tools
1904610.4	PO3	2	Designing mobile applications for the diverse purpose in the areas of software engineering, computer programming and graphics.
1904610.5	PO5	3	Developing the mobile application that incorporates digital tools that are used for the purpose of research and learning.

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
1904610	1	-	2	-	3	-	-	-	-	-	-	-

CO-PSO Mapping:

CO	PSO 1	PSO 2	PSO 3	PSO 4
1904610.1	-	2	-	-
1904610.2	-	2	-	-
1904610.3	-	3	-	-
1904610.4	-	-	-	-
1904610.5	-	2	-	-

Justification:

Course Outcome	PSO	Value	Justification
1904610.1	PSO2	2	Designing and developing mobile applications and deploying it using emulators
1904610.2	PSO2	2	Deploying the generated APK's of mobile applications in the hand-held devices.
1904610.3	PSO2	3	Developing mobile application based on the recent technologies and needs
1904610.5	PSO2	3	Deploying and extending the features of mobile applications for research and learning.

CO	PSO 1	PSO 2	PSO 3	PSO 4
1904610	-	2	-	-

EVALUATION PROCEDURE FOR EACH EXPERIMENT

MARK SPLIT UP	
Aim & Pre-Lab Discussion	**20**
Observation	**30**
Conduction and Execution	**30**
Output & Result	**10**
Viva	**10**
TOTAL	**100**

INTERNAL ASSESSMENT

Description	Marks
Conduction and Evaluation	**25**
Record	**10**
Model Test	**15**
TOTAL	**50**

INDEX

* Beyond the Syllabus

A. INTRODUCTION

CONFIGURING MOBILE APPLICATION DEVELOPMENT IN ANDROID ENVIRONMENT

Required Software to Install Android?

1. JDK 1.7
2. ADT (Android Development Kit) Bundle
3. Android SDK for Windows (if you have Windows OS)
4. Eclipse launcher.

What is Android SDK?

- The Android SDK includes a complete set of development tools. It includes a debugger, libraries, a handset emulator.
- Software written in Java can be compiled to be executed in the Dalvik virtual machine, which is a specialized VM implementation designed for mobile device use.

Steps for Software Installation

1. Download the latest version of jdk 7 and follow the installation procedure.
2. Download Android SDK from http://developer.android.com/sdk/index.html and the sdk can be installed.
3. The eclipse launcher is available in ADT bundle/android/eclipse.exe. Double clicking on the exe file will launch the eclipse in which android application is developed.

What is AVD? How to Configure AVD?

- AVD, stands Android Virtual Device, AVD's are used to start emulator. When we launch the emulator, we should specify the AVD configuration that we want to load.
- Before Configuring AVD, first confirm SDK tools and API levels are installed or not!
- Then select AVD Manager from **Window -> AVD Manager**.

- Click **New...** button, name your new AVD, select API target, SD Card size, then click **Create AVD** button.
- Click **Start...** button to start your new AVD. It will takes a few minutes for creating new AVD.

What is Emulator? How to Start?

- The emulator lets you prototype, develop and test Android applications without using a physical device.
- To use the emulator, first we must create AVD configuration, Remember, If project is used with Android API15 level , then we should use AVD API level also 15.
- Every Emulator needs one AVD support, because the AVD configuration provides both software and hardware support for Emulator, like it can use the services of the Android platform to invoke other applications, access the network, play audio and video, store and retrieve data, notify the user, and render graphical transitions and themes.
- The emulator also includes a variety of debug capabilities.
- The emulator provides dynamic binary translation of device machine code to the OS and processor architecture of your development machine.
- The Android emulator contains all of the hardware and software features of a real mobile device, except that it cannot place actual phone calls, can’t carry.
- Emulator provides a screen like as real device to display output of our testing application, using this we can easily test our applications.

COMMON PROCEDURE FOR ANDROID APPLICATION DEVELOPMENT (for all exercises)

1. Launch the Eclipse in android bundle.
2. After launching the eclipse, select File -> New -> Android -> Android Application Project.
3. In that name the application suitably as per the requirement.
4. Click Next. Default workspace will be displayed then click next.
5. The launcher icon of the application can be chosen preferably. Click next and select blank activity.
6. The activity and layout can be named or the default can be chosen and then click Finish.
7. Now the Android Application Project has been created.
8. The coding part of the application will be carried out in
   a. Project Name -> src -> MainActivity.java
   b. Project Name -> res -> layout-> activity_main.xml, where in the designing of the application will be carried in.
9. In activity_main.xml, the design for the application is created with linear layout, textview and edittext for the sign in application with background color and font styles.
10. In MainActivity.java, the sign in validation is performed. If the username and password matches it will toast a success message, if not log in failed will be toasted on the screen.
11. Right click on the activity.xml file -> run as-> android application. The application will be launched in the emulator.
12. The .apk file for the application developed will be available in Project -> bin folder.
13. The .apk file can be copied to an android device and the application can be installed and verify the output.

Ex No: 1

DEVELOP AN ANDROID APPLICATION TO WORK ON GUI COMPONENTS

AIM:

To develop a mobile application that uses GUI components.

DESCRIPTION:

In android UI or input controls are the interactive or View components that are used to design the user interface of an application. In android we have a wide variety of UI or input controls available, those are TextView, EditText, Buttons, Checkbox, Progressbar, Spinners, etc. The View is a base class for all UI components in android and it is used to create interactive UI components such as TextView, EditText, Checkbox, Radio Button, etc. and it is responsible for event handling and drawing. The ViewGroup is a subclass of View and it will act as a base class for layouts and layout parameters. The ViewGroup will provide invisible containers to hold other Views or ViewGroups and to define the layout properties.

PRE LAB VIVA QUESTIONS:

1. What is android?
2. What are the advantages of android?
3. Explain the architecture of android.
4. What is an activity?
5. What is a service?

SOURCE CODE:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
android:orientation="vertical"
android:gravity="center_horizontal"
android:background="#1EBAFF">
<EditText
      android:id="@+id/ed1"
android:layout_width="fill_parent"
android:layout_height="wrap_content
" android:textColor="#800000"
android:textSize="30dp"
android:textStyle="italic"/>
<EditText
```

```
android:id="@+id/ed2"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:inputType="textPassword"/>
<Button
android:id="@+id/bu"
android:layout_width="wrap_content"
android:layout_height="wrap_content
" android:text=" OK " />
</LinearLayout>
```

JAVA CODE

```
package com.example.mylogin;
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
public class MainActivity extends Activity
        { EditText ed1,ed2;
        Button bu;
        @Override
        protected void onCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.activity_main);
                ed1=(EditText)findViewById(R.id.ed1);
                ed2=(EditText)findViewById(R.id.ed2);
                bu=(Button)findViewById(R.id.bu);
                bu.setOnClickListener(new View.OnClickListener() {
                @Override
                        public void onClick(View v) {
                                // TODO Auto-generated method stub
                                String a = ed1.getText().toString();
                                String b = ed2.getText().toString();
                                if(a.equals("Welcome")&& b.equals("abcd")){
                                        Toast.makeText(getApplicationContext(), "success",
Toast.LENGTH_LONG).show();
                                }
                                else{
                                                Toast.makeText(getApplicationContext(), "Invalid
                                                User",
Toast.LENGTH_LONG).show();

                                }
```

```
            }
            });
    }

    @Override
    protected void onResume() {
            // TODO Auto-generated method stub
            super.onResume();

        }
        @Override
        public boolean onCreateOptionsMenu(Menu menu) {
                        // Inflate the menu; this adds items to the action bar if it
                is present. getMenuInflater().inflate(R.menu.main, menu);
                    return true;

}
```

OUTPUT:

POST LAB VIVA QUESTIONS:

1. Describe the APK format.
2. What is .apk extension?
3. What is .dex extension?
4. Describe a real time scenario where android can be used.
5. What language does Android support for application development?

RESULT:

Thus the mobile application using GUI components have been developed and deployed using the emulator.

Ex No: 2

WRITE AN ANDROID APPLICATION TO DEVELOP DICE ROLLER

AIM:

To develop an android application for dice roller.

DESCRIPTION:

A Roll Dice Game is a simple game in which you have to roll some dice and trying to make the highest score possible by adding the values of the two dice.

PRE LAB VIVA QUESTIONS:

1. Which components are necessary for a New Android project?
2. Provide the important core components of Android.
3. What is an Intent?
4. What is Orientation?
5. Explain the AndroidManifest.xml file and why do you need this?

SOURCE CODE:

DESIGN CODE – activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
                xmlns:tools="http://schemas.android.com/tools"
                android:id="@+id/activity_main"
                android:layout_width="match_parent"
                android:layout_height="match_parent"
                android:paddingBottom="@dimen/activity_vertical_margin"
                android:paddingLeft="@dimen/activity_horizontal_margin"
                android:paddingRight="@dimen/activity_horizontal_margin"
                android:paddingTop="@dimen/activity_vertical_margin"
                tools:context="com.ssaurel.dicer.MainActivity">
<LinearLayout
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginTop="80dp"
    android:orientation="horizontal"
    android:layout_centerHorizontal="true">
<ImageView
      android:id="@+id/imageView1"
```

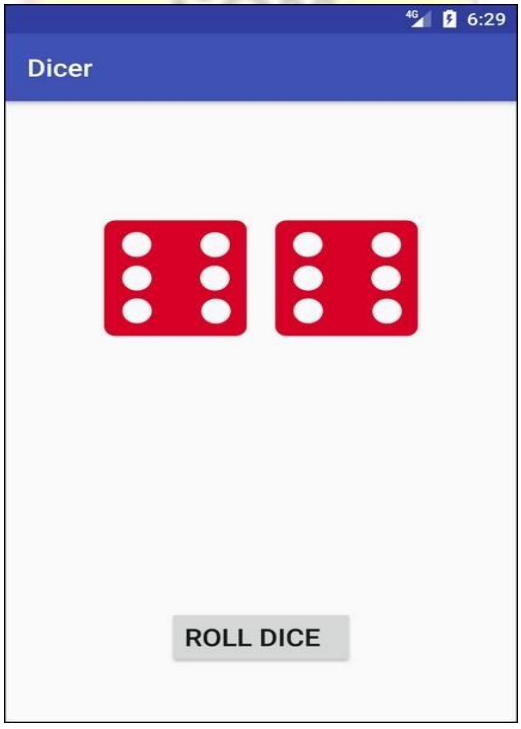
```
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_marginRight="20dp"
        android:src="@drawable/dice_2"/>
<ImageView
        android:id="@+id/imageView2"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:src="@drawable/dice_4"/>
</LinearLayout>
<Button
     android:id="@+id/rollDices"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Roll Dices"
      android:textSize="20sp"
      android:layout_alignParentBottom="true
      " android:layout_marginBottom="30dp"
      android:layout_centerHorizontal="true"/
      >
</RelativeLayout>
```

JAVA CODE:

```
package
com.example.dicer; import
android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.Button;
import
android.widget.ImageView;
import java.util.Random;
public class MainActivity extends AppCompatActivity {
   public static final Random RANDOM = new
   Random(); private Button rollDices;
   private ImageView imageView1,
   imageView2; @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      rollDices = (Button) findViewById(R.id.rollDices);
      imageView1 = (ImageView)
      findViewById(R.id.imageView1); imageView2 =
```

```
        (ImageView) findViewById(R.id.imageView2);
        rollDices.setOnClickListener(new View.OnClickListener() {
            @Override

            public void onClick(View view) {
                int value1 =
                randomDiceValue(); int value2
                = randomDiceValue();
                int res1 = getResources().getIdentifier("dice_" + value1, "drawable",
"com.example.dicer");
                int res2 = getResources().getIdentifier("dice_" + value2, "drawable",
"com.example.dicer");
                imageView1.setImageResource(res1)
                ;
                imageView2.setImageResource(res2)
                ;
            }
        });
    }
    public static int randomDiceValue()
        { return RANDOM.nextInt(6) + 1;
    }
}
```

OUTPUT:



POST LAB VIVA QUESTIONS:

1. Which tools are used for debugging on the Android platform?
2. Which scenario can test only on real devices but not on an emulator?
3. Name the mobile automation tools that are available in the market.
4. How do you troubleshoot the android application which is crashing frequently?
5. How do you find memory leaks in the mobile app on the Android platform?

RESULT:

Thus the mobile application for dice roller have been developed and deployed using the emulator.

Ex No: 3

WRITE AN ANDROID APPLICATION TO DEVELOP NATIVE CALCULATOR

AIM:

To develop a mobile application that serves a simple calculator.

DESCRIPTION:

The android calculator app will help us to do various arithmetic calculations. This android application will have a user interface with numbers and arithmetic operations.

PRE LAB VIVA QUESTIONS:

1. What is DDMS?
2. Which kernel is used in android?
3. What is fragment in android?
4. How to launch an activity in android?
5. How is the use of web view in Android?

SOURCE CODE:

DESIGN CODE – activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical" >
   <EditText
   android:layout_width="match_parent"
   android:layout_height="wrap_content
   " android:hint="enter no. here"
   android:id="@+id/tv"
   android:textSize="30dp"/>"

   <LinearLayout
   android:layout_width="match_parent"
   android:layout_height="wrap_content
   " android:orientation="horizontal"
   android:layout_marginTop="100dp"
   android:weightSum="4">

      <Button
      android:layout_width="match_parent"
      android:layout_height="wrap_content
```

```
" android:id="@+id/b9"

android:layout_weight="1"
android:text="9"
android:textColor="#ff0000"
/>"
<Button
android:layout_width="match_parent"
android:layout_height="wrap_content
" android:id="@+id/b8"
android:layout_weight="1"
android:text="8"
android:textColor="#ff0000"
/>
<Button
android:layout_width="match_parent"
android:layout_height="wrap_content
" android:id="@+id/b7"
android:layout_weight="1"
android:text="7"
android:textColor="#ff0000"
/>
<Button
android:layout_width="match_parent"
android:layout_height="wrap_content
" android:id="@+id/bpl"
android:layout_weight="1"
android:text="+"
android:textColor="#ff0000"
/>

</LinearLayout>
<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content
" android:orientation="horizontal"
android:weightSum="4">
<Button
android:layout_width="match_parent"
android:layout_height="wrap_content
" android:id="@+id/b6"
android:layout_weight="1"
android:text="6"
android:textColor="#ff0000"
/>
<Button
android:layout_width="match_parent"
android:layout_height="wrap_content
```

```
    "

    android:id="@+id/b5"
    android:layout_weight="1"
    android:text="5"
    android:textColor="#ff0000"
    />
    <Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content
    " android:id="@+id/b4"
    android:layout_weight="1"
    android:text="4"
    android:textColor="#ff0000"
    />
    <Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content
    " android:id="@+id/bmin"
    android:layout_weight="1"
    android:text="-"
    android:textColor="#ff0000"
    />
</LinearLayout>

<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content
" android:orientation="horizontal"
android:weightSum="4">
    <Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content
    " android:id="@+id/b3"
    android:layout_weight="1"
    android:text="3"
    android:textColor="#ff0000"
    />
    <Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content
    " android:id="@+id/b2"
    android:layout_weight="1"
    android:text="2"
    android:textColor="#ff0000"
    />
    <Button
    android:layout_width="match_parent"
```

```
    android:layout_height="wrap_content
    " android:id="@+id/b1"
    android:layout_weight="1"
    android:text="1"
    android:textColor="#ff0000"
    />
    <Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content
    " android:id="@+id/bmul"
    android:layout_weight="1"
    android:text="*"
    android:textColor="#ff0000"
    />
</LinearLayout>

<LinearLayout
android:layout_width="match_parent"
android:layout_height="wrap_content
" android:orientation="horizontal"
android:weightSum="5">
    <Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content
    " android:id="@+id/bd"
    android:layout_weight="1"
    android:text="."
    android:textColor="#ff0000"
    />
    <Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content
    " android:id="@+id/b0"
    android:layout_weight="1"
    android:text="0"
    android:textColor="#ff0000"
    />
    <Button
    android:layout_width="match_parent"
    android:layout_height="wrap_content
    " android:id="@+id/bcl"
    android:layout_weight="1"
    android:text="Clc"
    android:textColor="#ff0000"
    />
    <Button
```

```
        android:layout_width="match_parent"
        android:layout_height="wrap_content
        " android:id="@+id/beq"
        android:layout_weight="1"
        android:text="="
        android:textColor="#ff0000"
        />
        <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content
        " android:id="@+id/bdiv"
        android:layout_weight="1"
        android:text="/"
        android:textColor="#ff0000"
        />
    </LinearLayout>
</LinearLayout>
```

JAVA CODE:

```
package com.example.calculator_two;
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;

public class MainActivity extends Activity implements OnClickListener
{
    Button
    nine,eig,sev,six,fiv,four,thr,two,one,zero,dot,plus,mins,div,mul,eq,cl;
    EditText et;
    String s =
    "0"; int result
    = 0; char lO
    = ' ';
    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        // TODO Auto-generated method stub
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main)
        ; nine=(Button)findViewById(R.id.b9);
        eig=(Button)findViewById(R.id.b8);
        sev=(Button)findViewById(R.id.b7);
```

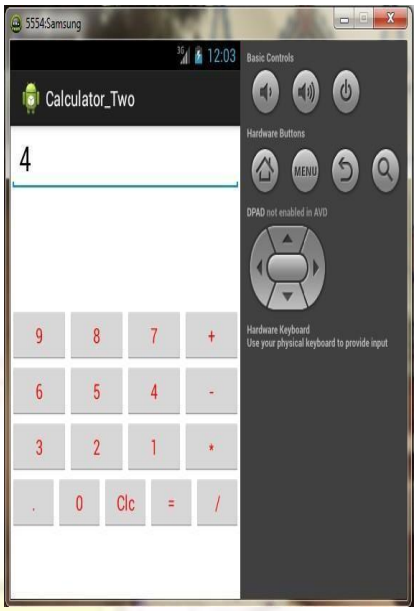
```
    six=(Button)findViewById(R.id.b6);
    fiv=(Button)findViewById(R.id.b5);
    four=(Button)findViewById(R.id.b4);
    thr=(Button)findViewById(R.id.b3);
    two=(Button)findViewById(R.id.b2);
    one=(Button)findViewById(R.id.b1);
    zero=(Button)findViewById(R.id.b0);
    dot=(Button)findViewById(R.id.bd);
    plus=(Button)findViewById(R.id.bpl);
    mins=(Button)findViewById(R.id.bmin)
    ; div=(Button)findViewById(R.id.bdiv);
    mul=(Button)findViewById(R.id.bmul);
    eq=(Button)findViewById(R.id.beq);
    cl=(Button)findViewById(R.id.bcl);
    et=(EditText)findViewById(R.id.tv);
    nine.setOnClickListener(this);
    eig.setOnClickListener(this);
    sev.setOnClickListener(this);
    six.setOnClickListener(this);
    fiv.setOnClickListener(this);
    four.setOnClickListener(this);
    thr.setOnClickListener(this);
    two.setOnClickListener(this);
    one.setOnClickListener(this);
    dot.setOnClickListener(this);
    plus.setOnClickListener(this);
    mins.setOnClickListener(this);
    div.setOnClickListener(this);
    mul.setOnClickListener(this);
    eq.setOnClickListener(this);
    cl.setOnClickListener(this);
    et.setOnClickListener(this);
}

@Override
public void onClick(View v)
{
    // TODO Auto-generated method stub
    switch(v.getId())
    {
        case R.id.b0:
        case R.id.b1:
        case R.id.b2:
        case R.id.b3:
        case R.id.b4:
        case R.id.b5:
        case R.id.b6:
        case R.id.b7:
```

```
        case R.id.b8:
        case R.id.b9:

        String inDigit = ((Button) v).getText().toString();
        if (s.equals("0"))
        {
           s= inDigit;
        }
        else
        {
           s+=inDigit;
        }

        Case R.id.beq:
        compute();
        lO = '=';
        break;
        case
        R.id.bcl:
        result = 0;
        s = "0";
        lO = ' ';
        et.setText("0");
        break;
    }
}
```

```
    private void compute()
    {
        // TODO Auto-generated method stub
        int inNum =
        Integer.parseInt(s); s = "0";
        if (lO == ' ')
        {
            result = inNum;
        }
        else if (lO == '+')
        {
            result += inNum;
        }
        else if (lO == '-')
        {
            result -= inNum;
        }
        else if (lO == '*')
        {
            result *= inNum;
        }
        else if (lO == '/')
        {
            result /= inNum;
        }
        else if (lO == '=')
        {
            // Keep the result for the next operation
        }
        et.setText(String.valueOf(result));
    }
}
```

OUTPUT:


5554:Samsung
12:02
Calculator_Two
5
Basic Controls
Hardware Buttons
MENU
DPAD not enabled in AVD
Hardware Keyboard
Use your physical keyboard to provide input
9 8 7 +
6 5 4 -
3 2 1 *
. 0 Clc = /


5554:Samsung
12:03
Calculator_Two
4
Basic Controls
Hardware Buttons
MENU
DPAD not enabled in AVD
Hardware Keyboard
Use your physical keyboard to provide input
9 8 7 +
6 5 4 -
3 2 1 *
. 0 Clc = /


5554:Samsung
12:03
Calculator_Two
9
Basic Controls
Hardware Buttons
MENU
DPAD not enabled in AVD
Hardware Keyboard
Use your physical keyboard to provide input
9 8 7 +
6 5 4 -
3 2 1 *
. 0 Clc = /

POST LAB VIVA QUESTIONS:

1. How to change application name after its deployment?
2. What is a Sticky Intent in android?
3. Where layouts are placed in android?
4. What are the exceptions available in android?
5. What is the order of dialog-box in android?

RESULT:

Thus the mobile application for simple calculator have been developed using the emulator.

Ex No: 4

WRITE AN ANDROID APPLICATION TO DEVELOP A REMINDER APP

AIM:
To develop a mobile application that serves as a reminder app.

DESCRIPTION:
A simple mobile application that serves as a reminder app that can keep track of all your necessary tasks. It will also notify you about the job through notifications.

PRE LAB VIVA QUESTIONS:

1. How is simulator different from emulator?
2. What are the reasons for which android is based on linux?
3. What are application widgets in android?
4. What is drawable folder in android?
5. How do you find any view element into your program?

SOURCE CODE:
DESIGN CODE – activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
  xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:background="@color/lightblue"
  android:layout_height="match_parent"
  tools:context=".MainActivity">
   <androidx.recyclerview.widget.RecyclerView
     android:id="@+id/recyclerView"
     android:layout_width="match_parent"
     android:layout_height="match_parent" />
   <com.google.android.material.floatingactionbutton.FloatingActionButton
     android:id="@+id/create_reminder"
     android:layout_width="wrap_content"
     android:layout_height="match_parent"
     android:layout_alignParentEnd="true"
     android:layout_alignParentBottom="true"
     android:layout_gravity="bottom"
     android:layout_marginEnd="30dp"
     android:layout_marginBottom="50dp"
     android:src="@drawable/plus_icon"
     />

 </RelativeLayout>
```

JAVA CODE:

```
package com.example.reminderapp;
import androidx.appcompat.app.AppCompatActivity;
import
androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;
import android.content.Intent;
import
android.database.Cursor;
import android.os.Bundle;
import android.view.View;
import com.google.android.material.floatingactionbutton.FloatingActionButton;
import java.util.ArrayList;
public class MainActivity extends AppCompatActivity {
FloatingActionButton mCreateRem;
RecyclerView mRecyclerview;
ArrayList<Model> dataholder = new ArrayList<Model>(); //Array list to add reminders and
display in recyclerview
myAdapter adapter;
@Override
protected void onCreate(Bundle savedInstanceState)
{ super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
mRecyclerview = (RecyclerView) findViewById(R.id.recyclerView);
mRecyclerview.setLayoutManager(new LinearLayoutManager(getApplicationContext()));
mCreateRem = (FloatingActionButton) findViewById(R.id.create_reminder); //Floating action
button to change activity
mCreateRem.setOnClickListener(new View.OnClickListener()
{ @Override
public void onClick(View view) {
Intent intent = new Intent(getApplicationContext(), ReminderActivity.class);
startActivity(intent); //Starts the new activity to add Reminders
}
});
Cursor cursor = new dbManager(getApplicationContext()).readallreminders(); //Cursor To
Load data From the database
while (cursor.moveToNext()) {
Model model = new Model(cursor.getString(1), cursor.getString(2), cursor.getString(3));
dataholder.add(model);
}
```

```
adapter = new myAdapter(dataholder);
mRecyclerview.setAdapter(adapter); //Binds the adapter with recyclerview
}
@Override
public void onBackPressed() {
finish(); //Makes the user to exit from the app
super.onBackPressed();
}
}
```

OUTPUT:

POST LAB VIVA QUESTIONS:

1. Explain any two android layouts.
2. List some of the best reminder apps.
3. Is there an app for random reminders?
4. How do you set reminders on an Android?
5. How do Google Assistant reminders work?

RESULT:

Thus the mobile application for creating reminder app has been developed and the output is displayed in emulator and an android device.

Ex No: 5

DEVELOP AN ANDROID APPLICATION TO IMPLEMENT MULTITHREADING

AIM:

To develop a mobile application that implements Multi-Threading.

DESCRIPTION:

An application is created using the multithreading concept. This can display any information by performing several actions at a time.

PRE LAB VIVA QUESTIONS:

1. List some benefits of using React Native for building mobile apps?
2. What are the advantages of hybrid apps over native apps?
3. What are the top Image Loading Libraries in Android?
4. How Garbage Collector Works in Android?
5. How to Reduce APK size in android?

SOURCE CODE:

DESIGN CODE – activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
>
<RelativeLayout
      android:id="@+id/firstlayout
      "
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:gravity="center"
android:layout_marginTop="80dp">
<TextView
         android:id="@+id/display"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
         android:text="@string/hello_world"
android:textSize="19sp" />
</RelativeLayout>
<RelativeLayout
      android:id="@+id/secondlayout
      "
android:layout_width="fill_parent"
android:layout_height="wrap_content"
```

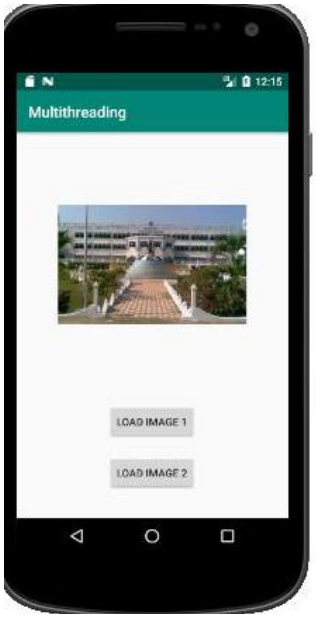
```
    android:layout_below="@+id/firstlayout
" android:gravity="center">
<TextView
        android:id="@+id/timer"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:gravity="center_horizontal"
        android:text="@string/timer"
android:layout_marginTop="80dp"
android:textSize="36sp"/>
</RelativeLayout>
<RelativeLayout
    android:id="@+id/thirdlayout
    "
android:layout_width="fill_parent"
android:layout_height="wrap_content"
    android:layout_below="@+id/secondlayout
" android:gravity="center">
<Button
        android:id="@+id/clickme"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
        android:text="@string/button"
android:visibility="invisible"
android:layout_marginTop="100dp"/>
</RelativeLayout>
</RelativeLayout>
```

JAVA CODE:

```
package
com.example.multithread; import
android.app.Activity; import
android.os.Bundle;
import android.os.Handler;
import android.widget.Button;
import
android.widget.TextView;
public class MainActivity extends Activity {
                Handler hand =
                newHandler(); Button
                clickme;
                TextView timer;
                @Override
                public void onCreate(Bundle savedInstanceState) {
                        super.onCreate(savedInstanceState);
                        setContentView(R.layout.activity_main); timer
```

```
                = (TextView) findViewById(R.id.timer);
                clickme = (Button) findViewById(R.id.clickme);
                hand.postDelayed(run, 1000);
        }

        Runnable run = new Runnable()
                { @Override
                public void run() {
                        updateTime();
                }
        };
        public void updateTime() {
                timer.setText("" + (Integer.parseInt(timer.getText().toString()) - 1));
                if (Integer.parseInt(timer.getText().toString()) == 0)
                        { clickme.setVisibility(0);
                                } else {
                                                                hand.postDelayed(run, 1000);
                                }
        }
}
```

OUTPUT:



POST LAB VIVA QUESTIONS:

1. What is ADB?
2. What are the four essential states of an activity?
3. Why can't you run java byte code on android?
4. What are the different storages available in android?
5. How many dialog boxes do support in android?

RESULT:

Thus the mobile application using multithreading have been developed and the output is displayed in emulator and an android device.

Ex No: 6

ANDROID APPLICATION TO IMPLEMENT ALL THE UI DESIGN (WIDGETS, LAYOUTS, UI EVENTS AND EVENT LISTENERS)

AIM:
To develop a mobile application that uses layout managers and event listeners.

DESCRIPTION:
When we design the user interface of our application, we decide what components we will use and how we will organise those components in the application. To organise our components, we use specialised non visible objects called layout managers.

PRE LAB VIVA QUESTIONS:

1. Give the use of Layout Managers in Android.
2. List out the various Layout Managers.
3. Explain about Event Listeners in Android.
4. What are the three concepts related to Android Event Management?
5. What is Touch Mode?

SOURCE CODE:
DESIGN CODE – activity_main.xml

```
<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:paddingBottom="@dimen/activity_vertical_margin"
   android:paddingLeft="@dimen/activity_horizontal_margin"
   android:paddingRight="@dimen/activity_horizontal_margin"
   android:paddingTop="@dimen/activity_vertical_margin"
tools:context=".MainActivity">

<TextView
      android:id="@+id/textView1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentTop="true"
android:layout_centerHorizontal="true"
android:layout_marginTop="42dp"
android:textAppearance="?android:attr/textAppearanceMedium" />

<TextView

      android:id="@+id/textView2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
```

```
        android:layout_alignTop="@+id/textView1"
android:layout_centerHorizontal="true"
android:text="User Name"
android:textAppearance="?android:attr/textAppearanceMedium" />

<EditText
        android:id="@+id/editText1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
        android:layout_below="@+id/textView1"
android:layout_centerHorizontal="true"
android:layout_marginTop="16dp"
android:ems="10">
<requestFocus />
</EditText>

<TextView
        android:id="@+id/textView3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_centerHorizontal="true"
android:layout_centerVertical="true"
android:text="Password"
android:textAppearance="?android:attr/textAppearanceMedium" />

<EditText
        android:id="@+id/editText2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
        android:layout_below="@+id/textView3"
android:layout_centerHorizontal="true"
android:layout_marginTop="30dp"
android:ems="10"
android:inputType="textPassword" />

<Button
        android:id="@+id/button1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
        android:layout_alignRight="@+id/textView2"
        android:layout_below="@+id/editText2"
android:layout_marginTop="51dp"

android:text="Button" />

</RelativeLayout>
```

second.xml (For Successful Login)

```
<?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:id="@+id/textView5"
android:text="Successful"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentTop="true"
android:layout_centerHorizontal="true"
android:layout_marginTop="42dp"
android:textAppearance="?android:attr/textAppearanceMedium" />
</AbsoluteLayout>
```

third.xml (For Login Failed)

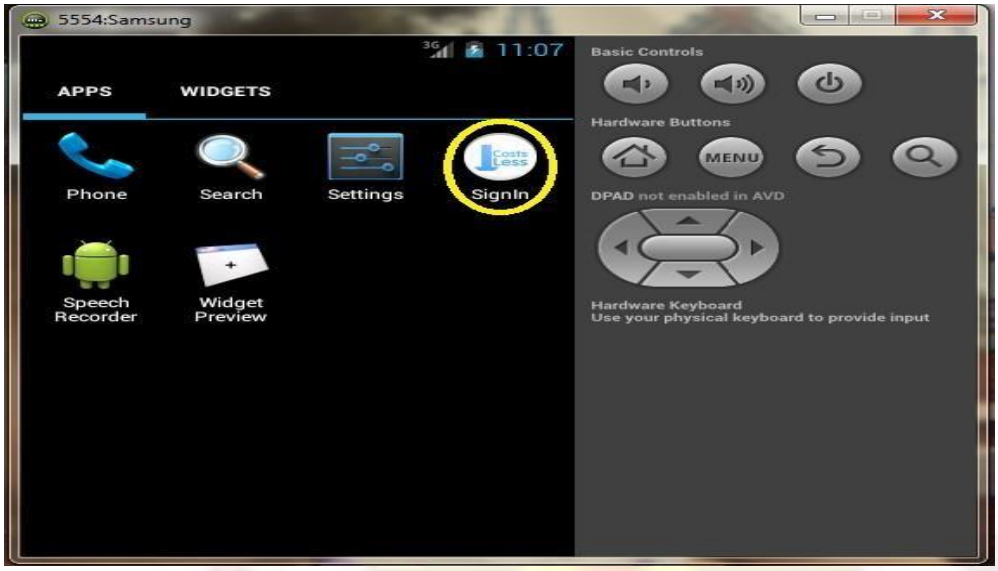
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical" >
<TextView
      android:id="@+id/textView6"
android:text="Login Failed"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentTop="true"
android:layout_centerHorizontal="true"
android:layout_marginTop="42dp"
android:textAppearance="?android:attr/textAppearanceMedium" />
</LinearLayout>
```

JAVA CODE:

```
package
com.example.signin; import
android.os.Bundle; import
android.app.Activity; import
android.view.Menu; import
android.view.View;
import android.view.View.OnClickListener;
```

```
import android.widget.Button;
import
android.widget.EditText;

publicclass MainActivity extends Activity {

                    EditText A,B,G;
                    Button C;
                    String E,F;

                    @Override
                    protectedvoid onCreate(Bundle savedInstanceState) {
                        super.onCreate(savedInstanceState);
                        setContentView(R.layout.activity_main);
                        A= (EditText)findViewById(R.id.editText1);
                        B= (EditText)findViewById(R.id.editText2);
                        C= (Button)findViewById(R.id.button1);
                        C.setOnClickListener(new OnClickListener()
                        {

                            @Override
                            publicvoid onClick(View v) {
                                // TODO Auto-generated method stub
                                E=A.getText().toString();
                                F=B.getText().toString();
                                if(E.equals("Admin")&&
                                F.equals("admin")){
                                setContentView(R.layout.second);
                                }
                                else {
                                    setContentView(R.layout.third);
                                }
                            }
                        });
                    }

                    @Override
                    publicboolean onCreateOptionsMenu(Menu menu) {
                        // Inflate the menu; this adds items to the action bar if it is present.
                        getMenuInflater().inflate(R.menu.main, menu);
                        returntrue;
                    }

}
```

OUTPUT:


5554:Samsung
11:07
APPS
WIDGETS
Phone
Search
Settings
SignIn
Speech Recorder
Widget Preview
Basic Controls
Hardware Buttons
MENU
DPAD not enabled in AVD
Hardware Keyboard
Use your physical keyboard to provide input


5554:Samsung
11:10
SignIn
User Name
Admin
Password
Button
Basic Controls
Hardware Buttons
MENU
DPAD not enabled in AVD
Hardware Keyboard
Use your physical keyboard to provide input


5554:Samsung
11:10
SignIn
Successful
Basic Controls
Hardware Buttons
MENU
DPAD not enabled in AVD
Hardware Keyboard
Use your physical keyboard to provide input

POST LAB VIVA QUESTIONS:

1. How to add second activity?
2. What is Linear Layout?
3. What is Frame Layout?
4. What is Grid Layout?
5. List out the various Event handlers.

RESULT:

Thus the mobile application using layout managers and event listeners have been developed using the emulator and an android device.

Ex No: 7

DEVELOP AN ANDROID APPLICATION TO MAKE USE OF DATABASE CONCEPTS

AIM:
To develop a mobile application that uses SQLite database.

DESCRIPTION:
An application is created in which the data is stored in the database and are retrieved. The database created can be for student information, employee information etc.

PRE LAB VIVA QUESTIONS:

1. How can the ANR be prevented?
2. What role does Dalvik play in Android development?
3. What is the AndroidManifest.xml?
4. What is the proper way of setting up an Android-powered device for app development?
5. Enumerate the steps in creating a bounded service through AIDL.

SOURCE CODE:

DESIGN CODE – activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayoutxmlns: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"
22
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"
23
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 CODE:

MainActivity.java

```
package com.example.admin.myapplication;
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 MainActivityextends Activity implements OnClickListener
{
EditTextRollno,Name,Marks;
19
 Button Insert,Delete,Update,View,ViewAll;
SQLiteDatabasedb;
/** 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);
```

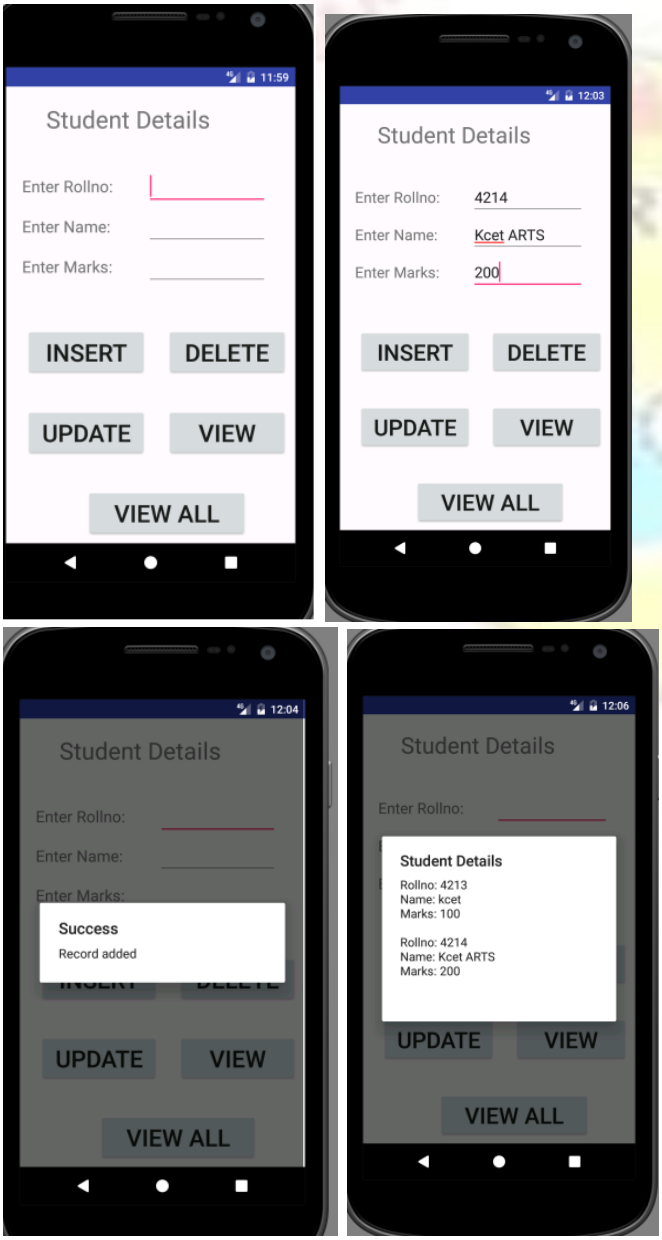
```
Insert=(Button)findViewById(R.id.Insert);
Delete=(Button)findViewById(R.id.Delete);
Update=(Button)findViewById(R.id.Update);
View=(Button)findViewById(R.id.View);
ViewAll=(Button)findViewById(R.id.ViewAll);
Insert.setOnClickListener(this);
Delete.setOnClickListener(this);
Update.setOnClickListener(this);
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(rollnoVARCHAR,nameVARCHAR,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;
20
```

```
 }
 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");
21
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:



POST LAB VIVA QUESTIONS:

1. What is the importance of Default Resources?
2. What are the different data storage options available on the Android platform?
3. What is *Android Data Binding*?
4. What is a Toast? Write its syntax.
5. What database is used in Android? How it is different from client-server database management systems?

RESULT:

Thus the mobile application for sign up and sign in is carried out using SQLLite Database and the output is displayed in emulator and an android device.

Ex No: 8

IMPLEMENT AN ANDROID APPLICATION TO USE 2D GRAPHICS

AIM:

To develop a mobile application that draws 2D graphics on the screen.

DESCRIPTION:

An application is created where the basic graphic primitives like line in different planes.

PRE LAB VIVA QUESTIONS:

1. What is the use of Image View?
2. What are Graphics Primitives?
3. What is ANR?
4. Which elements can occur only once and must be present?
5. How is escape characters used as attribute?
6. What is the importance of settings permissions in app development?

SOURCE CODE:

DESIGN CODE – activity_main.xml

```
<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">
<ImageView
      android:id="@+id/imageView1"
android:layout_width="wrap_content"
android:layout_height="match_parent"
android:layout_alignParentLeft="true"
android:layout_alignParentRight="true"
android:layout_alignParentTop="true"
      android:src="@drawable/ic_launcher" />
</RelativeLayout>
```

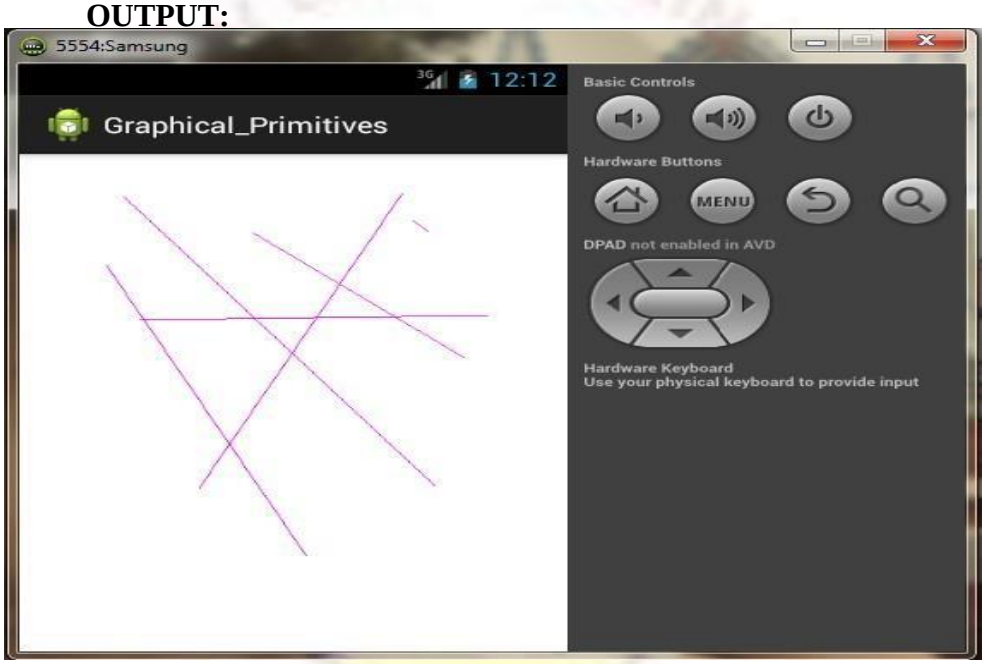
JAVA CODE:

```
package com.example.graphical_primitives;
import android.app.Activity;
import android.graphics.Bitmap;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
```

```
import android.os.Bundle;
import android.view.Display;
import
android.view.MotionEvent;
import android.view.View;
import android.view.View.OnTouchListener;
import android.widget.ImageView;

publicclass MainActivity extends Activity implements OnTouchListener {
  ImageView imageView;
  Bitmap bitmap;
  Canvas canvas;
  Paint paint;
float downx = 0, downy = 0, upx = 0, upy =
  0; @Override
publicvoid onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
imageView = (ImageView) this.findViewById(R.id.imageView1);
    Display currentDisplay = getWindowManager().getDefaultDisplay();
float dw = currentDisplay.getWidth();
float dh = currentDisplay.getHeight();

bitmap = Bitmap.createBitmap((int) dw, (int) dh,
      Bitmap.Config.ARGB_8888);
canvas = new Canvas(bitmap);
paint = new Paint();
paint.setColor(Color.MAGENTA)
;
imageView.setImageBitmap(bitmap);

imageView.setOnTouchListener(this);
  }

publicboolean onTouch(View v, MotionEvent event) {
int action = event.getAction();
switch (action) {
case
MotionEvent.ACTION_DOWN:
downx = event.getX();
downy = event.getY();
        break;
case MotionEvent.ACTION_MOVE:
        break;
case MotionEvent.ACTION_UP:
upx = event.getX();
```

```
upy = event.getY();

canvas.drawLine(downx, downy, upx, upy,
paint); imageView.invalidate();
        break;
case MotionEvent.ACTION_CANCEL:
break;
default:
break;
   }
        return true;
  }
}
```

OUTPUT:



POST LAB VIVA QUESTIONS:

1. What is the function of an intent filter?
2. Enumerate the three key loops when monitoring an activity.
3. When is the onStop() method invoked?
4. Is there a case wherein other qualifiers in multiple resources take precedence over locale?
5. What are the different states wherein a process is based?

RESULT:

Thus the mobile application to draw 2D graphics has been developed and the output is displayed in the emulator and an android device.

Ex No: 9

DEVELOP AN ANDROID APPLICATION TO IMPLEMENT MULTIMEDIA (AUDIO PLAYBACK AND MEDIA PLAYER)

AIM:
To develop a mobile application that implements media player.

DESCRIPTION:
Android framework provides support for a variety of media types as well. This enables us to integrate these audios, videos or images into our applications. For this, we can add the required files in the resource folder, in the raw file (res -> raw). Once we add them, we can open and play these using Media Player API.

PRE LAB VIVA QUESTIONS:

1. What is Android media player?
2. Can you deploy executable JARs on Android? Which package does it support?
3. What is the importance of having an Emulator?
4. Tell the importance of setting permissions in Android?
5. What is the significance of the .dex file?

SOURCE CODE:

DESIGN CODE – activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent
" android:paddingLeft="10dp"
android:paddingRight="10dp">
<TextView
android:id="@+id/txtSname"
android:layout_width="wrap_content"
android:layout_height="wrap_content
" android:layout_marginTop="30dp"
android:text="Now Playing: "
android:textAppearance="?android:attr/textAppearanceMedium" />
<ImageView
android:id="@+id/imgLogo
"
```

```
android:layout_width="402dp"
android:layout_height="419dp"
android:layout_marginTop="67dp"
android:src="@drawable/jwm"
tools:layout_marginLeft="40dp"
/>
<ImageButton
android:id="@+id/Backwardbtn"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentBottom="true
" android:layout_marginLeft="20dp"
android:layout_marginBottom="44dp"
android:src="@android:drawable/ic_media_rew" />
<ImageButton
android:id="@+id/Playbtn"
android:layout_width="wrap_content"
android:layout_height="wrap_content
"
android:layout_alignTop="@+id/Backwardbtn"
android:layout_marginLeft="20dp"
android:layout_toRightOf="@+id/Backwardbtn"
android:src="@android:drawable/ic_media_play" />
<ImageButton
android:id="@+id/Pausebtn"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignTop="@+id/Playbtn"
android:layout_marginLeft="20dp"
android:layout_toRightOf="@+id/Playbtn
"
android:src="@android:drawable/ic_media_pause" />
<ImageButton
android:id="@+id/Forwardbtn"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignTop="@+id/Pausebtn"
android:layout_marginLeft="20dp"
android:layout_toRightOf="@+id/Pausebtn
"
```

```
android:contentDescription="@+id/imageButton3"
android:src="@android:drawable/ic_media_ff" />
<TextView
android:id="@+id/StartTimebtn"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignTop="@+id/seek_Bar"
android:text="0 min 0 sec" />
<SeekBar
android:id="@+id/seek_Bar"
android:layout_width="match_parent"
android:layout_height="wrap_content
"
android:layout_above="@+id/Backwardbtn"
android:layout_toLeftOf="@+id/Song_t_txtview"
android:layout_toRightOf="@+id/StartTimebtn" />
<TextView
android:id="@+id/Song_t_txtview"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignTop="@+id/seek_Bar"
android:layout_toRightOf="@+id/Forwardbtn
" android:text="0 min 0 sec" />
</RelativeLayout>
```

JAVA CODE:

```
package
com.DataFlair.mediaplayer; import
android.media.MediaPlayer; import
android.os.Bundle;
import android.os.Handler;
import android.view.View;
import
android.widget.ImageButton;
import android.widget.SeekBar;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import java.util.concurrent.TimeUnit;
public class MainActivity extends AppCompatActivity {
private static int o_Time = 0, s_Time = 0, e_Time = 0, f_Time = 5000, b_Time = 5000;
```

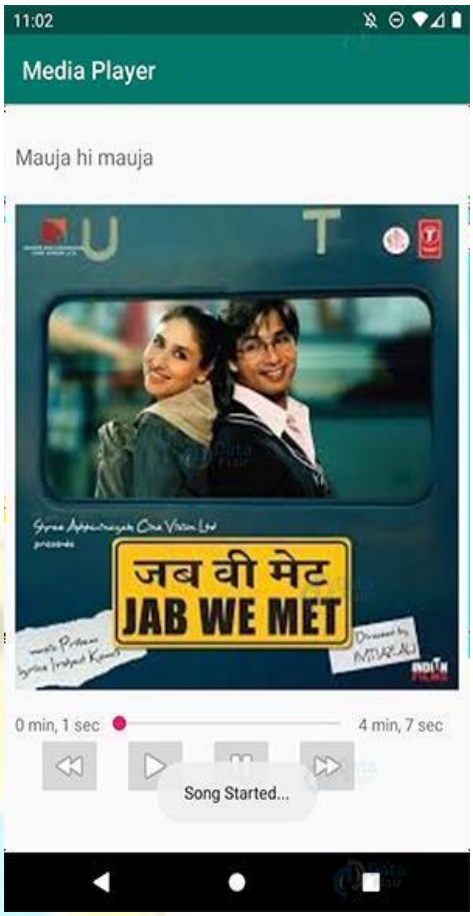
```
private ImageButton forward, backward, pause, play;
private Handler handle = new
Handler(); private MediaPlayer
mPlayer;
private TextView song, start,
song_time; private SeekBar Progress;

private Runnable UpdateSongTime = new Runnable() {
@Override
public void run() {
s_Time = mPlayer.getCurrentPosition();
start.setText(String.format("%d min, %d sec", TimeUnit.MILLISECONDS.toMinutes(s_Time),
TimeUnit.MILLISECONDS.toSeconds(s_Time) -
TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes(s_Time))));
Progress.setProgress(s_Time);
handle.postDelayed(this, 100);
}
};
@Override
protected void onCreate(Bundle savedInstanceState)
{ super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
backward =
findViewById(R.id.Backwardbtn); forward =
findViewById(R.id.Forwardbtn); play =
findViewById(R.id.Playbtn);
pause = findViewById(R.id.Pausebtn);
song = findViewById(R.id.txtSname);
start =
findViewById(R.id.StartTimebtn);
song_time = findViewById(R.id.Song_t_txtview);
song.setText("Mauja hi mauja");
mPlayer = MediaPlayer.create(this, R.raw.maujahimauja);
Progress = findViewById(R.id.seek_Bar);
Progress.setClickable(false);
pause.setEnabled(false);
play.setOnClickListener(new View.OnClickListener()
{ @Override
public void onClick(View v) {
Toast.makeText(MainActivity.this, "Song Started...", Toast.LENGTH_SHORT).show();
mPlayer.start();
e_Time = mPlayer.getDuration();
```

```
s_Time =
mPlayer.getCurrentPosition(); if
(o_Time == 0) {
Progress.setMax(e_Time);
o_Time = 1;
}
song_time.setText(String.format("%d min, %d
sec",
TimeUnit.MILLISECONDS.toMinutes(e_Time),
TimeUnit.MILLISECONDS.toSeconds(e_Time) -
TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes(e_Time))));
start.setText(String.format("%d min, %d sec", TimeUnit.MILLISECONDS.toMinutes(s_Time),
TimeUnit.MILLISECONDS.toSeconds(s_Time) -
TimeUnit.MINUTES.toSeconds(TimeUnit.MILLISECONDS.toMinutes(s_Time))));
Progress.setProgress(s_Time);
handle.postDelayed(UpdateSongTime,
100); pause.setEnabled(true);
play.setEnabled(false);
}
});
pause.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
mPlayer.pause();
pause.setEnabled(false);
play.setEnabled(true);
Toast.makeText(getApplicationContext(), "Song Paused...", Toast.LENGTH_LONG).show();
}
});
forward.setOnClickListener(new View.OnClickListener()
{ @Override
public void onClick(View v) {
if ((s_Time + f_Time) <= e_Time)
{ s_Time = s_Time + f_Time;
mPlayer.seekTo(s_Time);
} else {
Toast.makeText(getApplicationContext(), "Song Over", Toast.LENGTH_LONG).show();
}
if (!play.isEnabled())
{
play.setEnabled(true);
}
```

```
}
});
backward.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
if ((s_Time - b_Time) > 0) {
s_Time = s_Time - b_Time;
mPlayer.seekTo(s_Time);
} else {
Toast.makeText(getApplicationContext(), "Song not started yet...",
Toast.LENGTH_LONG).show();
}
if (!play.isEnabled())
{
play.setEnabled(true);
}
}
});
}
}
```

OUTPUT:


11:02
Media Player
Mauja hi mauja
जब वी मेट
JAB WE MET
0 min 0 sec
0 min 0 sec


11:02
Media Player
Mauja hi mauja
जब वी मेट
JAB WE MET
0 min, 1 sec
4 min, 7 sec
Song Started...

POST LAB VIVA QUESTIONS:

1. What is JNI in android?
2. What is sandbox in android?
3. What is runnable in android?
4. Which are the parts of "Application" layer of Android Architecture?
5. Who developed DVM?

RESULT:

Thus the mobile application to implement media player has been developed and the output is displayed in the emulator and an android device.

Ex No: 10

DEVELOP AN ANDROID APPLICATION TO MAKE USE OF NETWORKING CONCEPT

I. Making Phone call

AIM:

To develop an android application that makes phone call**.**

DESCRIPTION:

In android, we can easily make a phone call from our android applications by invoking built-in phone calls app using Intents action (ACTION_CALL). Generally, the Intent object in android with proper action (ACTION_CALL) and data will help us to launch a built-in phone calls app to make a phone calls in our application. In android, Intent is a messaging object which is used to request an action from another app component such as activities, services, broadcast receivers, and content providers.

PRE LAB VIVA QUESTIONS:

1. Which manifest file permission is required for an application to read the device's address book?
2. Which one is not a nickname of a version of android?
3. Which of the following has not included support for Adobe Flash or Java on its platform?
4. Which of the following is Apple's cloud service for hosted email service?
5. Which of the following is a photo and video sharing service by MobileMe?

SOURCE CODE:

DESIGN CODE – activity_main.xml

```
<?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/fstTxt"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content
      " android:layout_marginLeft="100dp"
      android:layout_marginTop="150dp"
      android:text="Mobile No"
      />
   <EditText
```

```
        android:id="@+id/mblTxt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content
        " android:layout_marginLeft="100dp"
        android:ems="10">
    </EditText>
    <Button
        android:id="@+id/btnCall"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content
        " android:layout_marginLeft="100dp"
        android:text="Call" />
</LinearLayout>
```

JAVA CODE:

```
package com.example.phonecallexample;
import android.Manifest;
import android.content.Intent;
import
android.content.pm.PackageManager;
import android.net.Uri;
import android.os.Build;
import android.support.v4.app.ActivityCompat;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {
    private EditText txtPhone;
    private Button
    btn; @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        txtPhone = (EditText)findViewById(R.id.mblTxt);
        btn = (Button)findViewById(R.id.btnCall);

        btn.setOnClickListener(new View.OnClickListener()
            { @Override
```

```
        public void onClick(View v)
            { callPhoneNumber();
            }
        });
    }
    @Override
    public void onRequestPermissionsResult(int requestCode, String[] permissions, int[]
grantResults)
    {
        if(requestCode == 101)
        {
            if(grantResults[0] == PackageManager.PERMISSION_GRANTED)
            {
                callPhoneNumber();
            }
        }
    }

    public void callPhoneNumber()
    {
        try
        {
            if(Build.VERSION.SDK_INT > 22)
            {
                if (ActivityCompat.checkSelfPermission(this, Manifest.permission.CALL_PHONE) !=
PackageManager.PERMISSION_GRANTED) {

ActivityCompat.requestPermissions(MainActivity.this, new String[]{Manifest.permission.CALL
_PHONE}, 101);
                    return;
                }

                Intent callIntent = new Intent(Intent.ACTION_CALL);
                callIntent.setData(Uri.parse("tel:" + txtPhone.getText().toString()));
                startActivity(callIntent);

            }
            else {
                Intent callIntent = new Intent(Intent.ACTION_CALL);
                callIntent.setData(Uri.parse("tel:" + txtPhone.getText().toString()));
```

```
            startActivity(callIntent);
          }
        }
        catch (Exception ex)
        {
          ex.printStackTrace();
        }
    }
}
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.phonecallexample">
    <uses-permission android:name="android.permission.CALL_PHONE" />
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher
        " android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

OUTPUT:

II. Sending Emails.

AIM:

To develop an android application that makes phone call.

DESCRIPTION:

In android, we can easily send an email from our android application using existing email clients such as GMAIL, Outlook, etc. instead of building an email client from scratch. Generally, the Intent object in android with proper action (ACTION_SEND) and data will help us to launch the available email clients to send an email in our application. In android, Intent is a messaging object which is used to request an action from another app component such as activities, services, broadcast receivers, and content providers.

SOURCE CODE:

DESIGN CODE – activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
```

```
    android:paddingLeft="20dp" android:paddingRight="20dp"
    android:orientation="vertical" >
   <EditText
       android:id="@+id/txtTo"
       android:layout_width="match_parent"
       android:layout_height="wrap_content
       " android:hint="To"/>
   <EditText
       android:id="@+id/txtSub"
       android:layout_width="match_parent"
       android:layout_height="wrap_content
       " android:hint="Subject"/>
   <EditText
       android:id="@+id/txtMsg"
       android:layout_width="match_parent
       " android:layout_height="0dp"
       android:layout_weight="1"
       android:gravity="top"
       android:hint="Message"/>
   <Button
       android:layout_width="100dp"
       android:layout_height="wrap_content
       " android:layout_gravity="right"
       android:text="Send"
     android:id="@+id/btnSend"/>
</LinearLayout>
```

JAVA CODE:

```
package com.example.sendmailexample;
import android.content.Intent;
import
android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import
android.widget.EditText;
public class MainActivity extends AppCompatActivity {
    private EditText eTo;
    private EditText
    eSubject; private
    EditText eMsg; private
    Button btn; @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
```

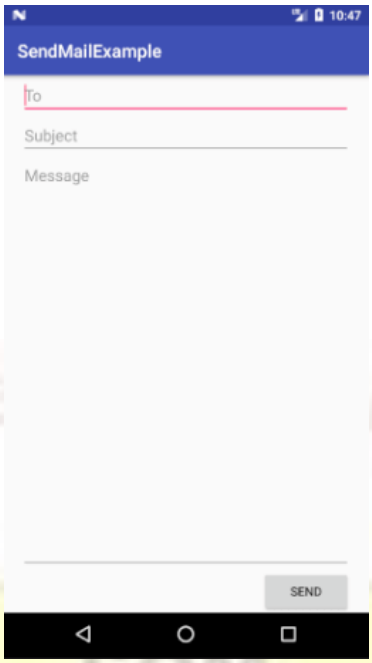
```
        setContentView(R.layout.activity_main);
        eTo = (EditText)findViewById(R.id.txtTo);

        eSubject =
        (EditText)findViewById(R.id.txtSub); eMsg =
        (EditText)findViewById(R.id.txtMsg); btn =
        (Button)findViewById(R.id.btnSend);
        btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent it = new Intent(Intent.ACTION_SEND);
                it.putExtra(Intent.EXTRA_EMAIL, new
                String[]{eTo.getText().toString()});
                it.putExtra(Intent.EXTRA_SUBJECT,eSubject.getText().toString());
                it.putExtra(Intent.EXTRA_TEXT,eMsg.getText());
                it.setType("message/rfc822");
                startActivity(Intent.createChooser(it,"Choose Mail App"));
            }
        });
    }
}
```

AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.example
.sendmailexample">
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher
        " android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
                <action android:name="android.intent.action.SEND"/>
                <category android:name="android.intent.category.DEFAULT"/>
                <data android:mimeType="message/rfc822"/>
            </intent-filter>
        </activity>
```

```
    </application>
</manifest>
```

OUTPUT:



III. Sending SMS

AIM:

To develop an android application that enables users to send SMS

DESCRIPTION:

In android, we can send SMS from our android application in two ways either by using SMSManager API or Intents based on our requirements. If we use SMSManager API, it will directly send SMS from our application. In case if we use Intent with proper action (ACTION_VIEW), it will invoke a built-in SMS app to send SMS from our application.

```
<?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/fstTxt"
        android:layout_width="wrap_content" android:layout_height="wrap_content"
        android:layout_marginLeft="100dp" android:layout_marginTop="150dp"

        android:text="Mobile No" />
    <EditText
        android:id="@+id/mblTxt
        "
        android:layout_width="wrap_content"
        android:layout_height="wrap_content
        " android:layout_marginLeft="100dp"
        android:ems="10"/>

    <TextView
        android:id="@+id/secTxt
        "
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Message"
        android:layout_marginLeft="100dp" />
    <EditText
        android:id="@+id/msgTxt"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content
        " android:layout_marginLeft="100dp"
        android:ems="10" />
    <Button
        android:id="@+id/btnSend"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content
        " android:layout_marginLeft="100dp"
        android:text="Send SMS" />
</LinearLayout>
```

JAVA CODE:

```
package com.example.sendsmsexample;
import android.content.Intent;
import android.net.Uri;
import android.provider.Telephony;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
```

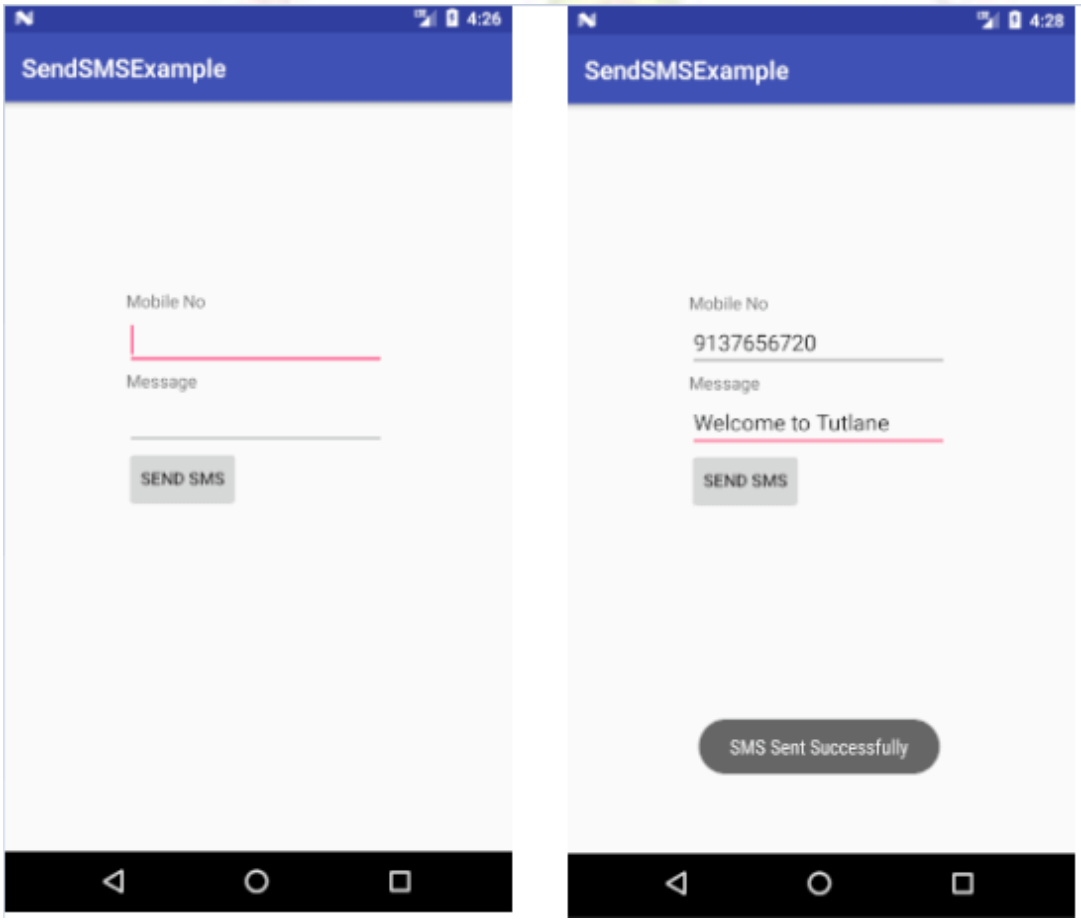
```
import android.telephony.SmsManager;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

   private EditText txtMobile;
   private EditText
   txtMessage; private Button
   btnSms; @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      txtMobile = (EditText)findViewById(R.id.mblTxt);
      txtMessage = (EditText)findViewById(R.id.msgTxt);
      btnSms = (Button)findViewById(R.id.btnSend);
      btnSms.setOnClickListener(new View.OnClickListener()
      {
         @Override
         public void onClick(View v)
            { try{
               SmsManager smgr = SmsManager.getDefault();

smgr.sendTextMessage(txtMobile.getText().toString(),null,txtMessage.getText().toString(),null,
null);
               Toast.makeText(MainActivity.this, "SMS Sent Successfully",
Toast.LENGTH_SHORT).show();
            }
            catch (Exception e){
               Toast.makeText(MainActivity.this, "SMS Failed to Send, Please try again",
Toast.LENGTH_SHORT).show();
            }
         }
      });
   }
}
```

AndroidManifest.xml

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

```
    package="com.example.sendsmsexample">
    <uses-permission android:name="android.permission.SEND_SMS"/>
    <application
        android:allowBackup="true"
        android:icon="@mipmap/ic_launcher
        "

        android:label="@string/app_name"
        android:roundIcon="@mipmap/ic_launcher_round"
        android:supportsRtl="true"
        android:theme="@style/AppTheme">
        <activity android:name=".MainActivity">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

OUTPUT:



POST LAB VIVA QUESTIONS:

1. Which is Apple's instant messaging service that interoperates with AIM?
2. What is a visible activity?
3. How do you remove icons and widgets from the main screen of the Android device?
4. What is portable wi-fi hotspot?
5. What is an action?

RESULT:

Thus the mobile application for phone calls, sending e-mails and SMS has been developed and the output is displayed in the emulator and an android device.

Ex No: 11

DEVELOP AN ANDROID APPLICATION TO CREATE CLIPBOARD

AIM:

To develop an android application that makes phone call**.**

DESCRIPTION:

In android, Clipboard is a framework that is useful for copying and pasting the different types of data such as text strings, images, binary stream data, and other complex data types. Generally, the android Clipboard framework will store the simple text data directly in the clipboard and the complex data is stored as a reference that the pasting application resolves with a content provider. In android, the clipboard copying and pasting works within an application and between the applications that implement the framework. To use the android clipboard framework, we need to put the data into a clip object, and then put the clip object on the system-wide clipboard.

SOURCE CODE:

DESIGN CODE – activity_main.xml

```
<?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/fstTxt"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content
      " android:layout_marginLeft="100dp"
      android:layout_marginTop="150dp"
      android:text="Enter Text to Copy"
      />
   <EditText
      android:id="@+id/txtCopy"
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_marginLeft="100dp"
      android:text="Welcome to Android Clipboard"
      android:ems="10">
   </EditText>
   <Button
```

```
    android:id="@+id/btnCopy"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="100dp"
    android:text="Copy Data to Clipboard" />
  <TextView
    android:id="@+id/secTxt
    "
    android:layout_width="wrap_content"
    android:layout_height="wrap_content
    " android:text="Show Copied Data"
    android:layout_marginLeft="100dp"
    />
  <EditText
    android:id="@+id/txtShow"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content
    " android:layout_marginLeft="100dp"
    android:ems="10" />
  <Button
    android:id="@+id/btnShow"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginLeft="100dp"
    android:text="Show Clipboard Data" />
</LinearLayout>
```

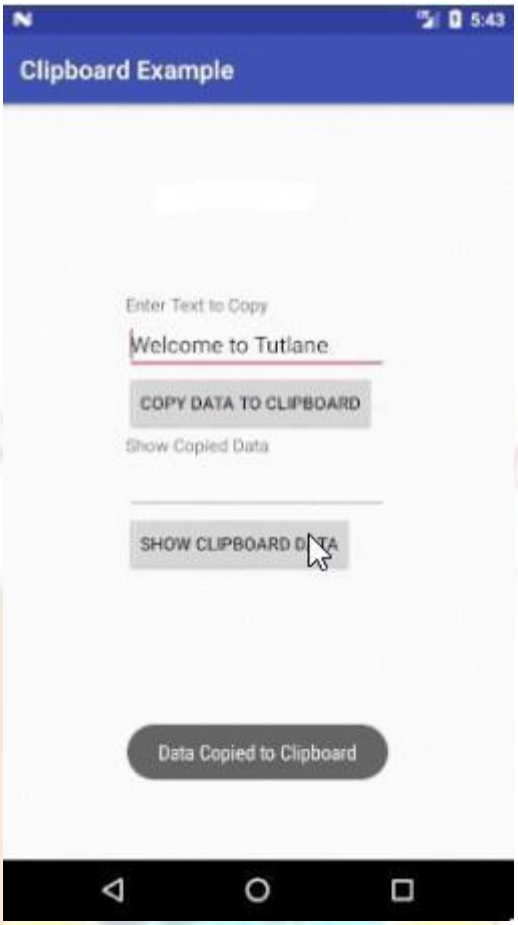
JAVA CODE:

```
package com.example.clipboardexample;
import android.content.ClipData;
import
android.content.ClipboardManager;
import android.content.Context;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
```

```
    private EditText ctxt;

    private EditText ptxt;
    private Button btncpy;
    private Button btnpst;
    private ClipboardManager
    clipboardManager; private ClipData
    clipData;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        ctxt =(EditText)findViewById(R.id.txtCopy);
        ptxt = (EditText)findViewById(R.id.txtShow);
        btncpy = (Button)findViewById(R.id.btnCopy);
        btnpst = (Button)findViewById(R.id.btnShow);
        clipboardManager =
(ClipboardManager)getSystemService(Context.CLIPBOARD_SERVICE);
        btncpy.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                String txtcopy = ctxt.getText().toString();
                clipData = ClipData.newPlainText("text",txtcopy);
                clipboardManager.setPrimaryClip(clipData);
                Toast.makeText(getApplicationContext(),"Data Copied to Clipboard",
Toast.LENGTH_SHORT).show();
            }
        });
        btnpst.setOnClickListener(new View.OnClickListener()
            { @Override
            public void onClick(View v) {
                ClipData pData =
                clipboardManager.getPrimaryClip(); ClipData.Item
                item = pData.getItemAt(0);
                String txtpaste = item.getText().toString();
                ptxt.setText(txtpaste);
                Toast.makeText(getApplicationContext(),"Data Pasted
                from
Clipboard",Toast.LENGTH_SHORT).show();
            }
        });
```

```
    }
}
```

OUTPUT:



RESULT:

Thus the mobile application to implement android clipboard has been developed and the output is displayed in the emulator and an android device.

Ex. No. 12

Designing BMI Calculator Application

AIM:

To develop a Simple Android Application to design a Simple BMI Calculator

DESCRIPTION:

Creating a New project:

• Open Android Studio and then click on File -> New -> New project.

• Then type the Application name as "BMICalc", change the project location and click Next.

• Then select the Minimum SDK as shown below and click Next.

• Then select the Empty Activity and click Next.

• Finally click Finish.

Designing layout for the Android Application:

• Click on app -> res -> layout -> activity_main.xml.

• Now click on Text shown below.

• Then delete the code which is there and type the code as given below.

SOURCE CODE:

```
<!-- Linear layout start here -->
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
 xmlns:tools="http://schemas.android.com/tools"
 android:layout_width="fill_parent"
 android:layout_height="fill_parent"
 android:background="@drawable/images1"
 android:fadingEdge="horizontal"
 android:orientation="vertical" >
 <!-- Text view for BMI Text -->
 <TextView
 android:id="@+id/tv1"
 android:layout_width="124dp"
 android:layout_height="wrap_content"
 android:layout_gravity="center"
 android:paddingLeft="15dp"
 android:paddingTop="40dp"
 android:shadowColor="@android:color/black"
 android:shadowDx="4"
 android:shadowDy="4"
 android:text="BMI"
 android:textAppearance="?android:attr/textAppearanceLarge"
 android:textColor="@android:color/white"
 android:textSize="50sp"
 android:typeface="serif" />
 <!-- Textview for calculator text -->
```

```
<TextView
android:id="@+id/tv2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="center"
android:text="Calculator"
android:textColor="@android:color/white"
android:textSize="20dp"
android:textStyle="bold" />
<!-- Textview for WEIGHT(KG) text -->
<TextView
android:id="@+id/tv3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="center"
android:paddingTop="30dp"
android:text="WEIGHT (KG)"
android:textAppearance="?android:attr/textAppearanceMedium"
android:textColor="@android:color/white"
android:textStyle="bold|italic"
android:typeface="serif" />
<!-- Edit text for entering weight with hint in kgs -->
<EditText
android:id="@+id/et1"
android:layout_width="96dp"
android:layout_height="wrap_content"
android:layout_gravity="center"
android:hint="IN KGs"
android:ems="10"
android:fadingEdgeLength="10dp"
android:inputType="numberDecimal"
android:textAlignment="center" >
<requestFocus />
</EditText>
<!-- Text view for HEIGHT(CM)text -->
<TextView
android:id="@+id/tv4"
android:layout_width="151dp"
android:layout_height="wrap_content"
android:layout_gravity="center"
android:foregroundGravity="center_horizontal"
android:gravity="center_horizontal"
```

```
 android:paddingTop="30dp"
 android:text="HEIGHT (CM)"
 android:textAppearance="?android:attr/textAppearanceMedium"
 android:textColor="@android:color/white"
 android:textStyle="bold|italic"
 android:typeface="serif" />
 <!-- Edit text for entering height with hint in cm -->
 <EditText
 android:id="@+id/et2"
 android:layout_width="96dp"
 android:layout_height="wrap_content"
 android:layout_gravity="center"
 android:hint="IN CMs"
 android:ems="10"
 android:inputType="numberDecimal" >
 </EditText>
<!--Button for calculating the formula, when pressed, with calculate written over it-->
 <Button
 android:id="@+id/ib1"
 android:layout_width="158dp"
 android:layout_height="51dp"
 android:layout_gravity="center"
 android:layout_marginTop="20dp"
 android:fadingEdge="vertical"
 android:longClickable="true"
 android:nextFocusRight="@android:color/holo_orange_dark"
 android:text="Calculate"
 android:visibility="visible" />
 <!-- Text view for showing result -->
 <TextView
 android:id="@+id/tv5"
 android:layout_width="278dp"
 android:layout_height="wrap_content"
 android:layout_gravity="center"
android:gravity="center"
 android:paddingTop="20dp"
 android:text=""
 android:textColor="@android:color/holo_orange_dark"
 android:textSize="20dp"
 android:textStyle="bold" />
</LinearLayout>
 <!-- Linear layout ends here -->
```

Now click on the Design tab and now the application will look as given below.

Java Coding for the Android Application:

• Click on app -> java -> com.example. calcapp -> MainActivity.

• Then delete the code which is there and type the code as given below.

```
package akn.bmicalc;
//Import necessary package and file
import android.os.Bundle;
import android.app.Activity;
import android.text.TextUtils;
import android.view.Menu;
import android.view.View;
import android.widget.EditText;
import android.widget.TextView;
//Main activity class start here
public class MainActivity extends Activity {
 //Define layout
 @Override
 protected void onCreate(Bundle savedInstanceState) {
 super.onCreate(savedInstanceState);
 setContentView(R.layout.activity_main);
// Get the references to the widgets
 final EditText e1 = (EditText) findViewById(R.id.et1);
 final EditText e2 = (EditText) findViewById(R.id.et2);
 final TextView tv5 = (TextView) findViewById(R.id.tv5);
 findViewById(R.id.ib1).setOnClickListener(new View.OnClickListener() {
 // Logic for validation, input can't be empty
 @Override
 public void onClick(View v) {
 String str1 = e1.getText().toString();
 String str2 = e2.getText().toString();
 if(TextUtils.isEmpty(str1)){
 e1.setError("Please enter your weight");
 e1.requestFocus();
 return;
 }
 if(TextUtils.isEmpty(str2)){
 e2.setError("Please enter your height");
 e2.requestFocus();
 return;
 }
//Get the user values from the widget reference
 float weight = Float.parseFloat(str1);
```

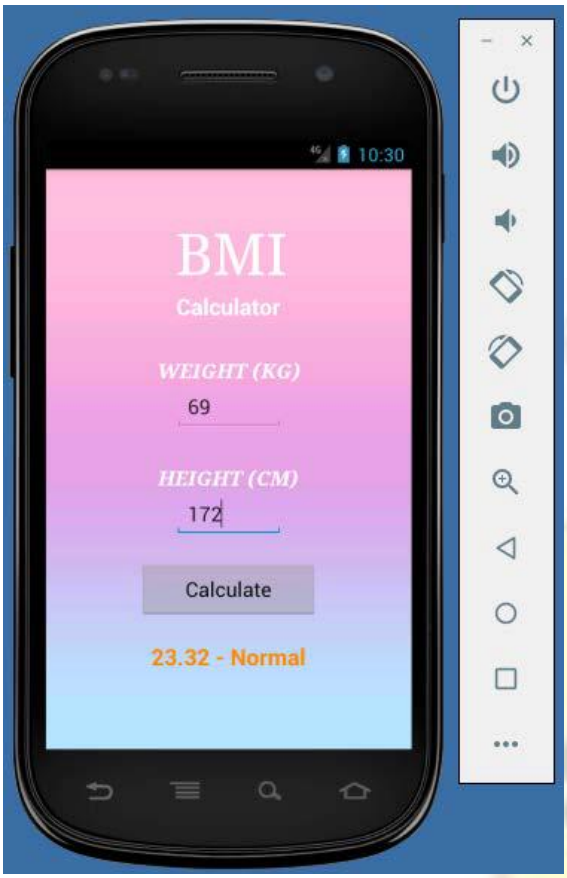
```
 float height = Float.parseFloat(str2)/100;
//Calculate BMI value
 float bmiValue = calculateBMI(weight, height);
//Define the meaning of the bmi value
 String bmiInterpretation = interpretBMI(bmiValue);
 tv5.setText(String.valueOf(bmiValue + " - " + bmiInterpretation));
 }
 });
 }
 //Calculate BMI
 private float calculateBMI (float weight, float height) {
 return (float) Math.round((weight / (height * height))*100)/100;
 }
 // Interpret what BMI means
 private String interpretBMI(float bmiValue) {
 if (bmiValue < 16) {
 return "Severely underweight";
 } else if (bmiValue < 18.5) {
 return "Underweight";
 } else if (bmiValue < 25) {
 return "Normal";
 } else if (bmiValue < 30) {
 return "Overweight";
 } else {
 return "Obese";
 }
 }
}
```

Click Play icon or press Shift+F10 and select any of the Available Virtual devices from the list or create a New AVD.

Emulator will be loaded and displays the output of the app developed as shown below.

(Select Build APK(s) from Build Menu and build the .apk file for this application. Locate the apk file created and copy it to the mobile Phone to install it and run it from it for verification.)

OUTPUT:



RESULT:

Thus a Simple Android Application to design a Simple BMI Calculator is developed and executed successfully in emulator and Mobile device.