SRM VALLIAMMAI ENGINEERING COLLEGE

(An Autonomous Institution)

SRM Nagar, Kattankulathur-603203.

Department of Computer Science and Engineering

Lab Manual

IT3664- MOBILE APPLICATIONDEVELOPMENT LABORATORY

(VI Semester- Regulation 2023)

Prepared By
Ms.V.VijayPriya, Asst.Prof (O.G)
Ms.M.Priyadharshini, Asst.Prof (O.G)
Ms.G.Sathya, Asst.Prof (O.G)

LIST OF EXPERIMENTS

IT3664 MOBILE APPLICATION DEVELOPMENT LABORATORY

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. Develop an android application to make use of Databases
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 a native application that uses GPS location information
8. Implement an application that creates an alert upon receiving a message
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

INDEX

IT3664 MOBILE APPLICATION DEVELOPMENT LAB **L T P C**
0 0 3 1.5

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. Develop an android application to make use of Databases
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 a native application that uses GPS location information
8. Implement an application that creates an alert upon receiving a message
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

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

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

PEO1. To mould students to exhibit top performance in higher education and research and to become a State-of –the-art technocrat.

PEO2. To impart the necessary background in Computer Science and Engineering by providing solid foundation in Mathematical, Science and Engineering fundamentals.

PEO3. To equip the students with the breadth of Computer Science and Engineering innovate novel solutions for the benefit of common man.

PEO4. To groom the student to be a multifaceted entrepreneur with professional ethical attitude in a broader social perspective.

PEO5. To provide an ambience learning environment that is conducive for the growth of successful professional career of students

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 appropriateconsideration for the public health and safety, and the cultural, societal, and environmental considerations.
4. Conduct in 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 modeling 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 normsof the engineering practice.
9. Individual and team work: Function effectively as an individual, and as a member or leader indiverse 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)

PSO1: Exhibit proficiency in planning, implementing and evaluating team oriented-software programming solutions to specific business problems and society needs.

PSO2: Demonstrate professional skills in applying programming skills, competency and decision making capability through hands-on experiences.

PSO3: Apply logical thinking in analyzing complex real world problems, and use professional and ethical behaviors to provide proper solutions to those problems.

PSO4: Demonstrate the ability to work effectively as part of a team in applying technology to Business and personal situations.

COURSE OUTCOMES:
COURSE NAME: IT3664 MOBILE APPLICATION DEVELOPMENT LAB
YEAR OF STUDY: 2025-2026

IT3664.1	Design and Implement various mobile applications using emulators
IT3664.2	Deploy applications to hand-held devices
IT3664.3	Transfer current knowledge to learning of new technologies
IT3664.4	Have Skills to software development, computer programming & graphic
IT3664.5	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
IT3664.1	-	-	-	-	3	-	-	-	-	-	-	-
IT3664.2	1	-	2	-	-	-	-	-	-	-	-	-
IT3664.3	1	-	-	-	-	-	-	-	-	-	-	-
IT3664.4	-	-	2	-	-	-	-	-	-	-	-	-
IT3664.5	-	-	-	-	3	-	-	-	-	-	-	-

Justification:

Course Outcome	Program Outcome	Value	Justification
IT3664.1	PO5	3	Creating the mobile application using Eclipse/Android Studio and deploying it in emulator.
IT3664.2	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.
IT3664.3	PO1	1	Applying the knowledge of developing mobile applications in advanced platforms/tools
IT3664.4	PO3	2	Designing mobile applications for the diverse purpose in the areas of software engineering, computer programming and graphics.
IT3664.5	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
IT3664	1	-	2	-	3	-	-	-	-	-	-	-

CO-PSO Matrix:

CO	PSO 1	PSO 2	PSO 3	PSO 4
IT3664.1	-	2	-	-
IT3664.2	-	2	-	-
IT3664.3	-	3	-	-
IT3664.4	-	-	-	-
IT3664.5	-	2	-	-

Justification:

Course Outcome	PSO	Value	Justification
IT3664.1	PSO2	2	Designing and developing mobile applications and deploying it using emulators
IT3664.2	PSO2	2	Deploying the generated APK's of mobile applications in the hand-held devices.
IT3664.3	PSO2	3	Developing mobile application based on the recent technologies and needs
IT3664.5	PSO2	3	Deploying and extending the features of mobile applications for research and learning.

CO	PSO 1	PSO 2	PSO 3	PSO 4
IT3664	-	2	-	-

EVALUATION PROCEDURE FOR EACH EXPERIMENT

MARK SPLIT UP	
Aim & Pre-Lab Discussion	**05**
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	**100**

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.

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
```

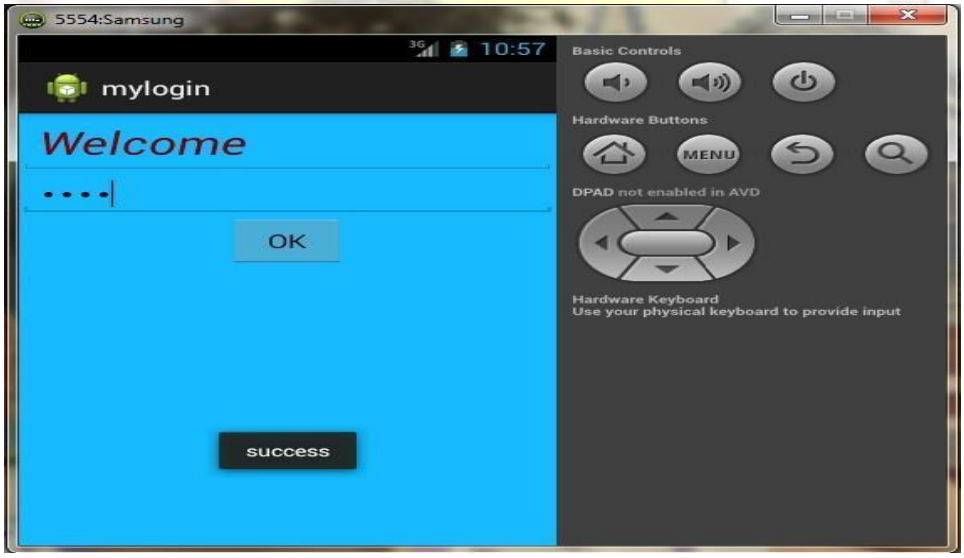
```
android:id="@+id/ed2"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:inputType="textPassword"
    />
<Button
android:id="@+id/bu"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="  OK " />
</LinearLayout>
```

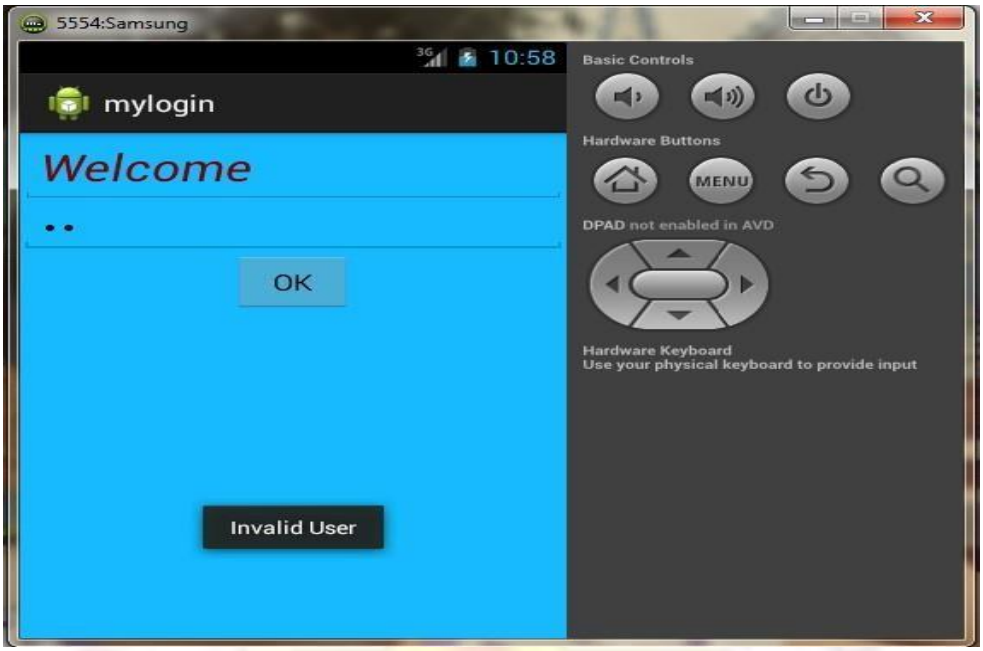
JAVA CODE

```
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





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.

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"
```

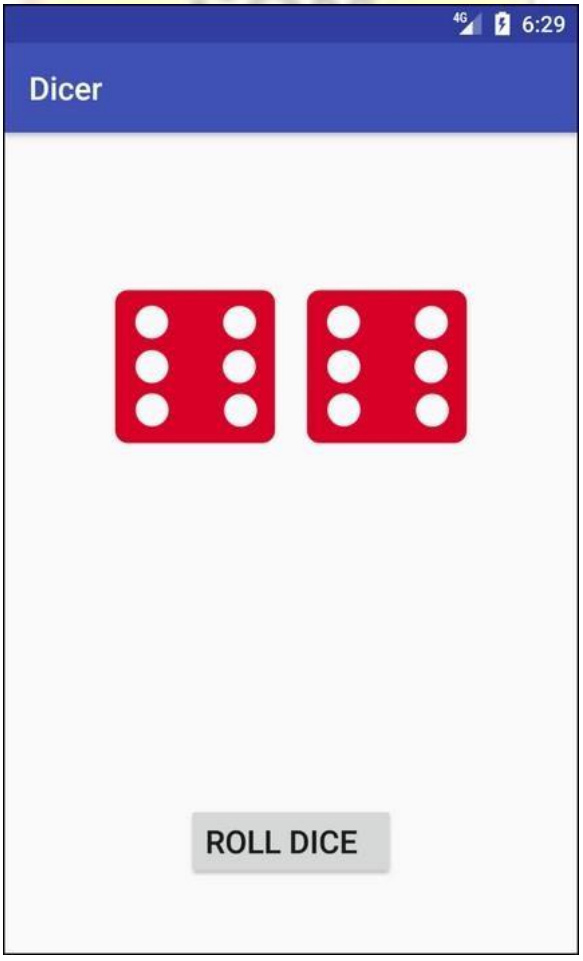
```
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:layout_marginRight="20dp"
        android:src="@drawable/dice_2"/>
<ImageView
        android:id="@+id/imageView2"
        android:layout_width="100dp"
        android:layout_height="100dp"
        android:src="@drawable/dice_4"/>
</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



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.

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);
```

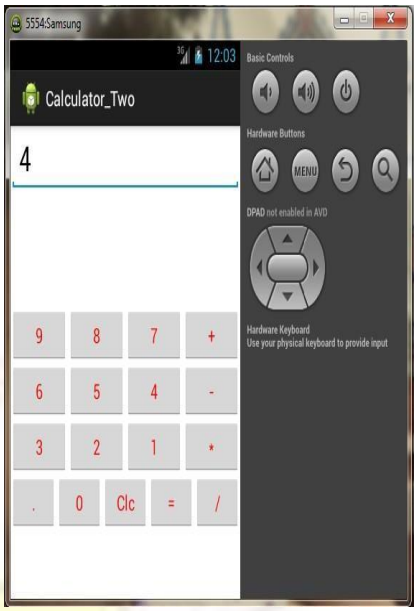
```
    thr=(Button)findViewById(R.id.b3);
    two=(Button)findViewById(R.id.b2);
    one=(Button)findViewById(R.id.b1);
    zero=(Button)findViewById(R.id.b0);
    dot=(Button)findViewById(R.id.bd);
    plus=(Button)findViewById(R.id.bpl);
    mins=(Button)findViewById(R.id.bmin);
    div=(Button)findViewById(R.id.bdiv);
    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;
        }
        et.setText(s);
        if(lO == '=')
        {
           result=0;
           lO =' ';
        }
        break;
        case R.id.bpl:
        compute();
        lO = '+';
        break;
        case R.id.bmin:
        compute();
        lO = '-';
        break;
        case R.id.bdiv:
        compute();
        lO = '/';
        break;
        case R.id.bmul:
        compute();
        lO = '*';
        break;
        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 = /

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

DEVELOP AN ANDROID APPLICATION TO MAKE USE OF DATABASES

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.

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.example.sqliteoperations.MainActivity"
    android:background="@android:color/holo_blue_dark">
    <TextView
        android:text="@string/username"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_alignParentTop="true"
        android:layout_marginTop="12dp"
        android:id="@+id/textView"
        android:textSize="18sp"
        android:textStyle="bold|italic"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
```

```
    android:gravity="center" />
<EditText
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:inputType="textPersonName"
    android:ems="10"
    android:id="@+id/editName"
    android:textStyle="bold|italic"
    android:layout_below="@+id/textView"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:hint="Enter Name"
    android:gravity="center_vertical|center" />

<TextView
    android:text="@string/password"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="13dp"
    android:id="@+id/textView2"
    android:textStyle="bold|italic"
    android:textSize="18sp"
    android:layout_below="@+id/editName"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true"
    android:gravity="center"
    android:hint="Enter Password" />
<Button
    android:text="@string/view_data"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/button2"
    android:textSize="18sp"
    android:onClick="viewdata"
    android:textStyle="bold|italic"
    android:layout_alignBaseline="@+id/button"
    android:layout_alignBottom="@+id/button"
    android:layout_alignRight="@+id/button4"
    android:layout_alignEnd="@+id/button4" />
<Button
    android:text="@string/add_user"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/button"
    android:textStyle="bold|italic"
    android:textSize="18sp"
```

```
    android:onClick="addUser"
    android:layout_marginLeft="28dp"
    android:layout_marginStart="28dp"
    android:layout_below="@+id/editPass"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_marginTop="23dp" />
<Button
    android:text="@string/update"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/button3"
    android:onClick="update"
    android:textStyle="normal|bold"
    android:layout_below="@+id/editText3"
    android:layout_alignLeft="@+id/button4"
    android:layout_alignStart="@+id/button4"
    android:layout_marginTop="13dp" />
<EditText
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:inputType="textPersonName"
    android:ems="10"
    android:id="@+id/editText6"
    android:layout_alignTop="@+id/button4"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:freezesText="false"
    android:hint="Enter Name to Delete Data"
    android:layout_toLeftOf="@+id/button2"
    android:layout_toStartOf="@+id/button2" />
<Button
    android:text="@string/delete"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_marginRight="21dp"
    android:layout_marginEnd="21dp"
    android:id="@+id/button4"
    android:onClick="delete"
    android:textStyle="normal|bold"
    tools:ignore="RelativeOverlap"
    android:layout_marginBottom="41dp"
    android:layout_alignParentBottom="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true" />
<EditText
```

```
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="textPersonName"
        android:ems="10"
        android:layout_marginTop="47dp"
        android:id="@+id/editText3"
        android:textStyle="bold|italic"
        android:textSize="14sp"
        android:layout_below="@+id/button"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_marginLeft="7dp"
        android:layout_marginStart="7dp"
        android:hint="Current Name" />
    <EditText
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:inputType="textPassword"
        android:ems="10"
        android:layout_marginTop="11dp"
        android:id="@+id/editPass"
        android:hint="Enter Password"
        android:gravity="center_vertical|center"
        android:textSize="18sp"
        android:layout_below="@+id/textView2"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:textAllCaps="false"
        android:textStyle="normal|bold" />
    <EditText
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:inputType="textPersonName"
        android:ems="10"
        android:id="@+id/editText5"
        android:textStyle="bold|italic"
        android:textSize="14sp"
        android:hint="New Name"
        android:layout_alignTop="@+id/button3"
        android:layout_alignLeft="@+id/editText3"
        android:layout_alignStart="@+id/editText3"
        android:layout_marginTop="32dp" />
</RelativeLayout>
```

JAVA CODE

MainActivity.java

```
package com.example.sqliteoperations;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.EditText;

public class MainActivity extends AppCompatActivity {
   EditText Name, Pass , updateold, updatenew, delete;
   myDbAdapter helper;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
      Name= (EditText) findViewById(R.id.editName);
      Pass= (EditText) findViewById(R.id.editPass);
      updateold= (EditText) findViewById(R.id.editText3);
      updatenew= (EditText) findViewById(R.id.editText5);
      delete = (EditText) findViewById(R.id.editText6);

      helper = new myDbAdapter(this);
   }
   public void addUser(View view)
   {
      String t1 = Name.getText().toString();
      String t2 = Pass.getText().toString();
      if(t1.isEmpty() || t2.isEmpty())
      {
         Message.message(getApplicationContext(),"Enter Both Name and Password");
      }
      else
      {
         long id = helper.insertData(t1,t2);
         if(id<=0)
         {
            Message.message(getApplicationContext(),"Insertion Unsuccessful");
            Name.setText("");
            Pass.setText("");
         } else
         {
            Message.message(getApplicationContext(),"Insertion Successful");
            Name.setText("");
            Pass.setText("");
         }
```

```
    }
}

public void viewdata(View view)
{
    String data = helper.getData();
    Message.message(this,data);
}

public void update( View view)
{
    String u1 = updateold.getText().toString();
    String u2 = updatenew.getText().toString();
    if(u1.isEmpty() || u2.isEmpty())
    {
        Message.message(getApplicationContext(),"Enter Data");
    }
    else
    {
        int a= helper.updateName( u1, u2);
        if(a<=0)
        {
            Message.message(getApplicationContext(),"Unsuccessful");
            updateold.setText("");
            updatenew.setText("");
        } else {
            Message.message(getApplicationContext(),"Updated");
            updateold.setText("");
            updatenew.setText("");
        }
    }

}
public void delete( View view)
{
    String uname = delete.getText().toString();
    if(uname.isEmpty())
    {
        Message.message(getApplicationContext(),"Enter Data");
    }
    else{
        int a= helper.delete(uname);
        if(a<=0)
        {
            Message.message(getApplicationContext(),"Unsuccessful");
            delete.setText("");
```

```
            }
            else
            {
               Message.message(this, "DELETED");
               delete.setText("");
            }
         }
      }
   }
```

myDbAdapter. Java

```
package com.example.sqliteoperations;
import android.content.ContentValues;
import  android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
public class myDbAdapter {
   myDbHelper myhelper;
   public myDbAdapter(Context context)
   {
      myhelper = new myDbHelper(context);
   }

   public long insertData(String name, String pass)
   {
      SQLiteDatabase dbb = myhelper.getWritableDatabase();
      ContentValues contentValues = new ContentValues();
      contentValues.put(myDbHelper.NAME, name);
      contentValues.put(myDbHelper.MyPASSWORD, pass);
      long id = dbb.insert(myDbHelper.TABLE_NAME, null , contentValues);
      return id;
   }

   public String getData()
   {
      SQLiteDatabase db = myhelper.getWritableDatabase();
      String[] columns =
{myDbHelper.UID,myDbHelper.NAME,myDbHelper.MyPASSWORD};
      Cursor cursor =db.query(myDbHelper.TABLE_NAME,columns,null,null,null,null,null);
      StringBuffer buffer= new StringBuffer();
      while (cursor.moveToNext())
      {
         int cid =cursor.getInt(cursor.getColumnIndex(myDbHelper.UID));
         String name =cursor.getString(cursor.getColumnIndex(myDbHelper.NAME));
         String  password
=cursor.getString(cursor.getColumnIndex(myDbHelper.MyPASSWORD));
```

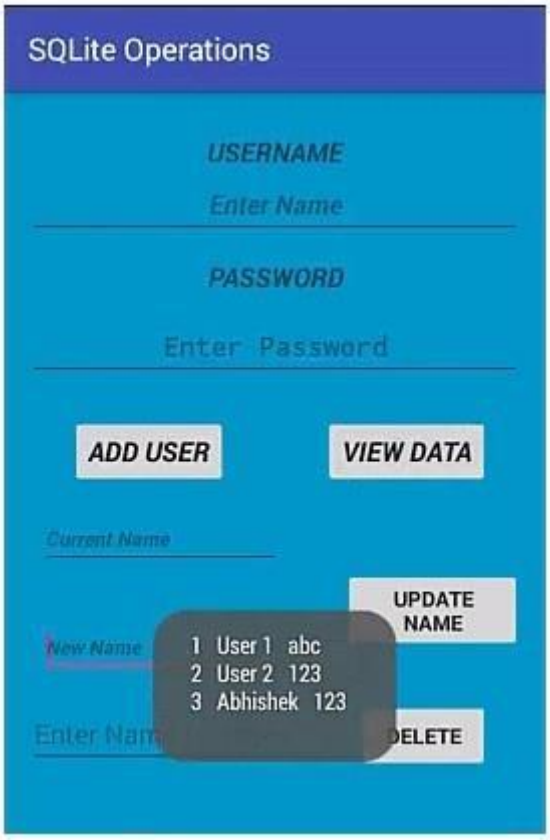
```
            buffer.append(cid+ "   " + name + "    " + password +" \n");
        }
        return buffer.toString();
    }

    public  int delete(String uname)
    {
        SQLiteDatabase db = myhelper.getWritableDatabase();
        String[] whereArgs ={uname};

        int count =db.delete(myDbHelper.TABLE_NAME ,myDbHelper.NAME+" =
?",whereArgs);
        return  count;
    }

    public int updateName(String oldName , String newName)
    {
        SQLiteDatabase db = myhelper.getWritableDatabase();
        ContentValues contentValues = new ContentValues();
        contentValues.put(myDbHelper.NAME,newName);
        String[] whereArgs= {oldName};
        int count =db.update(myDbHelper.TABLE_NAME,contentValues, myDbHelper.NAME+"
= ?",whereArgs );
        return count;
    }

    static class myDbHelper extends SQLiteOpenHelper
    {
        private static final String DATABASE_NAME = "myDatabase";    // Database Name
        private static final String TABLE_NAME = "myTable";  // Table Name
        private static final int DATABASE_Version = 1;.    // Database Version
        private static final String UID="_id";     // Column I (Primary Key)
        private static final String NAME = "Name";    //Column II
        private static final String MyPASSWORD= "Password";    // Column III
        private static final String CREATE_TABLE = "CREATE TABLE "+TABLE_NAME+
                " ("+UID+" INTEGER PRIMARY KEY AUTOINCREMENT, "+NAME+"
VARCHAR(255) ,"+ MyPASSWORD+" VARCHAR(225));";
        private static final String DROP_TABLE ="DROP TABLE IF EXISTS "+TABLE_NAME;
        private Context context;

        public myDbHelper(Context context) {
            super(context, DATABASE_NAME, null, DATABASE_Version);
            this.context=context;
        }

        public void onCreate(SQLiteDatabase db) {
```

```
        try {
            db.execSQL(CREATE_TABLE);
        } catch (Exception e) {
            Message.message(context,""+e);
        }
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        try {
            Message.message(context,"OnUpgrade");
            db.execSQL(DROP_TABLE);
            onCreate(db);
        }catch (Exception e) {
            Message.message(context,""+e);
        }
    }
  }
}
```

Message.class

```
package com.example.sqliteoperations;
import  android.content.Context;
import android.widget.Toast;
public class Message {
   public static void message(Context context, String message) {
      Toast.makeText(context, message, Toast.LENGTH_LONG).show();
   }
}
```

OUTPUT



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: 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.

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"
```

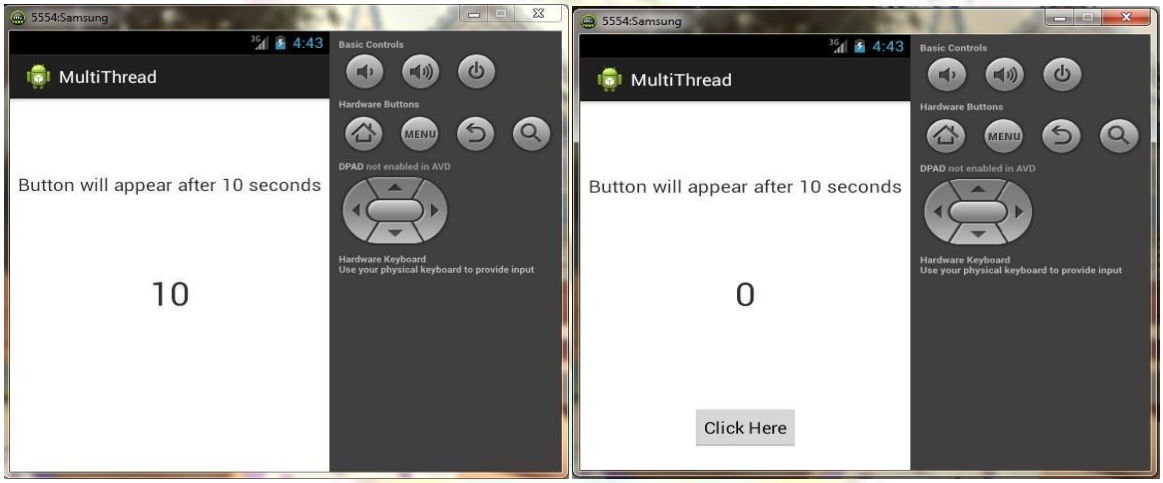
```
        android:layout_below="@+id/firstlayout"
android:gravity="center">
<TextView
        android:id="@+id/timer"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:gravity="center_horizontal"
        android:text="@string/timer"
android:layout_marginTop="80dp"
android:textSize="36sp"/>
</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 = newRunnable() {
                @Override
                publicvoid 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



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.

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)

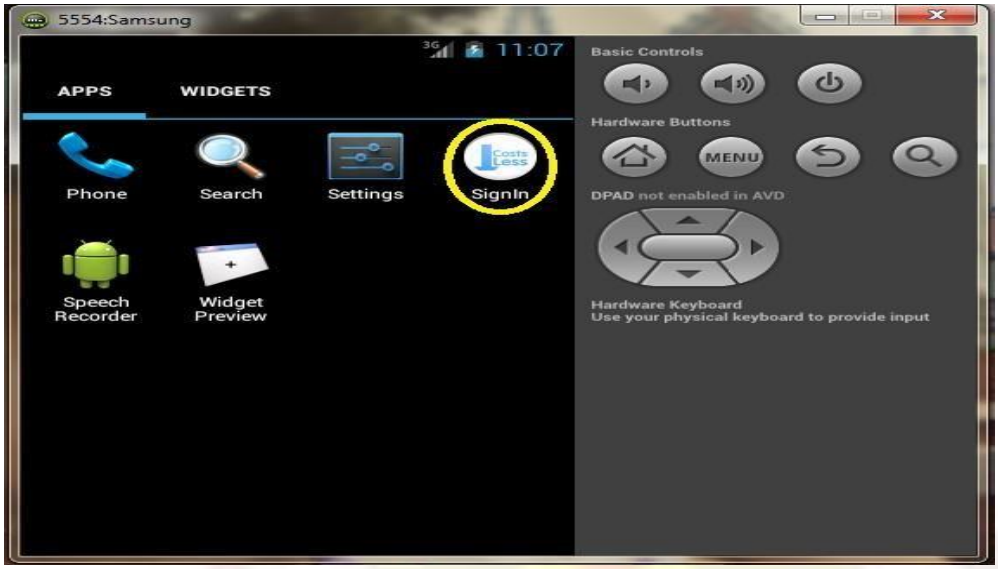
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="vertical" >
<TextView
      android:id="@+id/textView6"
android:text="Login Failed"
android:layout_width="wrap_content"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

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 A NATIVE APPLICATION THAT USES GPS LOCATION INFORMATION

AIM

To develop a native application that uses GPS location information.

DESCRIPTION

Step 1: File→ New → Android Project Application
Step 2: Specify Application Name
Step 3: Design the layout using buttons.
Step 4: Update the locations of all devices in Location Control with its latitudes and longitudes.
Step 5: Run the application using AVD Emulator.
Step 6: Now, Track the Location

SOURCE CODE

UseGps.java

```
package com.emergency; import
android.app.Activity; import
android.content.Context;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.widget.Button; import
android.widget.EditText; import
android.widget.Toast;
public class UseGps extends Activity
{
Button buttonSend; EditTexttextSMS;
EditTexttextlon;
public void onCreate(Bundle
savedInstanceState)
{
super.onCreate(savedInstanceState);
setContentView(R.layout.main);
buttonSend = (Button)
findViewById(R.id.buttonSend); textSMS =
(EditText) findViewById(R.id.editTextSMS);
textlon = (EditText)
findViewById(R.id.textlon);
LocationManagermlocManager =
(LocationManager)getSystemService(Context.
LOCATION_SERVICE);
LocationListenermlocListener = new
MyLocationListener();
mlocManager.requestLocationUpdates(
LocationManager.GPS_PROVIDER, 0, 0,
mlocListener);
}
```

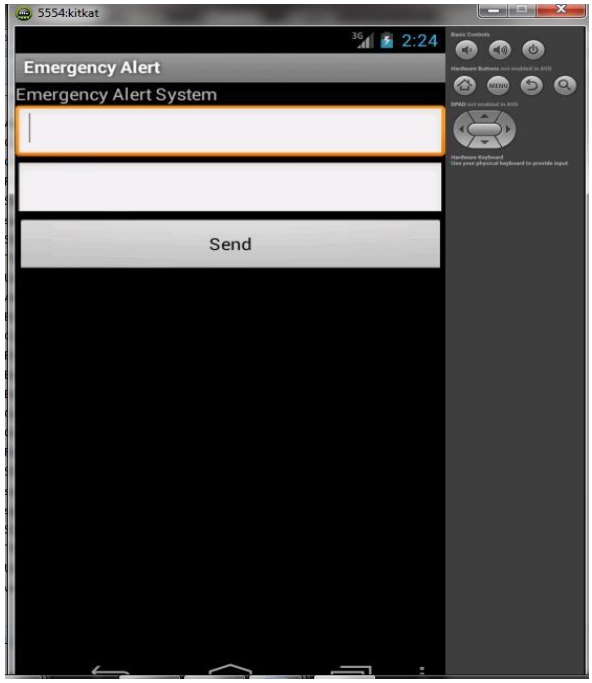
```
public class MyLocationListener implements
LocationListener
{
public void onLocationChanged(Location loc)
{
loc.getLatitude(); loc.getLongitude();
Double lat=loc.getLatitude(); Double
lon=loc.getLongitude();
textSMS.setText(lat.toString());
textlon.setText(lon.toString());
}
public void onProviderDisabled(String
provider)
{
Toast.makeText( getApplicationContext(),"Gps
Disabled",Toast.LENGTH_SHORT ).show();
}
public void onProviderEnabled(String
provider)
{
Toast.makeText( getApplicationContext(),
"Gps Enabled",
Toast.LENGTH_SHORT).show();
}
}
}
```

main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayoutxmlns:android="http://schemas.
android.com/apk/res/android"
android:orientation="vertical"
android:layout_width="fill_parent"
android:layout_height="fill_parent"
>
<TextView android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:text="Emergency Alert System"
/>

<EditText android:id="@+id/editTextSMS"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:gravity="top" />

<EditText android:id="@+id/textlon"
android:layout_width="fill_parent"
android:layout_height="wrap_content"
android:gravity="top" />

<Button android:id="@+id/buttonSend"
android:layout_width="fill_parent"
```

```
android:layout_height="wrap_content"
android:text="Send" />

</LinearLayout>
```

OUTPUT



VIVA QUESTIONS

1. What is GPS and how is it used in an Android application?
2. Which permissions are required to access location services in Android?
3. What is the role of LocationManager **or** FusedLocationProviderClient?
4. What is the difference between GPS provider and Network provider?
5. How do you handle runtime permission for location access in Android?

RESULT

Thus, the program for android application that makes use of GPS information was executed successfully.

Ex No: 8

IMPLEMENT AN APPLICATION THAT CREATES AN ALERT UPON RECEIVING A MESSAGE

AIM

To develop an android application that creates an alert upon receiving a message.

DESCRIPTION

1. Create a New AVD (Android Virtual Device):
   - click Android Virtual Device Manager from the toolbar.
   - In the Android Virtual Device Manager panel, click New.
   - Fill in the details for the AVD. Give it a name, a platform target, an SD card size, and a skin (HVGA is default).
   - Click Create AVD and Select the new AVD from the Android Virtual Device Manager and click Start.
2. Design the layout by adding a text box and a command button.
3. Run the application.
4. If the entered E-mail doesn't match the given E-mail id, then an alert will be displayed.
5. If the entered E-mail id matches with the provided mail-id then login is successful.
6. Close the Android project.

SOURCE CODE

MainActivity.java

```
package com.pa.Alert; import
android.os.Bundle;
import android.app.Activity; import
android.content.Intent; import
android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText; import
android.widget.Toast;
class MainActivity extends Activity { private
Button BTN; private EditText email; protected
void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main); BTN
= (Button) findViewById(R.id.btn); email =
(EditText) findViewById(R.id.emailInput);
BTN.setOnClickListener(new
OnClickListener() { public void onClick(View
v) { String val = email.getText().toString(); if
(val == null || val.length() <= 0) {
Toast.makeText(getApplicationContext(),
```

```
"Please Enter the email",
Toast.LENGTH_LONG).show(); } else if
(val.equals("enpboss@gmail.com")) { Intent
intent = new Intent(getApplicationContext(),
SecondActivity.class); startActivity(intent);
Toast.makeText(getApplicationContext(),
"Login Success",
Toast.LENGTH_LONG).show(); } else {
Toast.makeText(getApplicationContext(),

"Please Enter valid email",
Toast.LENGTH_LONG) .show(); } } }); } }
SecondActivity.java package com.pa.Alert;

import android.app.Activity; import
android.os.Bundle; public class SecondActivity

extends Activity { @Override protected void
onCreate(Bundle savedInstanceState) { //
TODO Auto-generated method stub
super.onCreate(savedInstanceState);
setContentView(R.layout.second_activity); } }
```

Main_activity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayoutxmlns: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:orientation="vertical" > <EditText
android:id="@+id/emailInput"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:ems="10" /> <Button
android:id="@+id/btn"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_margin="20sp"
android:gravity="center" android:text="Login"
/> </LinearLayout>
```

AndroidMainfest.Xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/ap
k/res/android"
package="com.example.admin.myapplication"
> <application android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
```

```
android:label="@string/app_name"
android:roundIcon="@mipmap/ic_launcher_ro
und" 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.LAUN
CHER" /> </intent-filter> </activity> <activity
android:name=".SecondActivity"> <intent-
filter> <action
android:name="android.intent.action.MAIN" />
<category
android:name="android.intent.category.LAUN
CHER" /> </intent-filter> </activity>
</application> </manifest>
```

OUTPUT


3:25
LOGIN
3:26
thirdyear@gmail.com
LOGIN
Please Enter valid email
3:31
LOGIN
Login Success

VIVA QUESTIONS

1. What is a BroadcastReceiver and why is it used for receiving SMS?
2. Which permissions are required to read and receive SMS in Android?
3. How does an Android application detect an incoming SMS?
4. What is the role of SmsManager in Android?
5. How do you generate an alert or notification when a message is received?

RESULT

Thus, the program for android application that creates an alert upon receiving a message was executed successfully.

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.

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

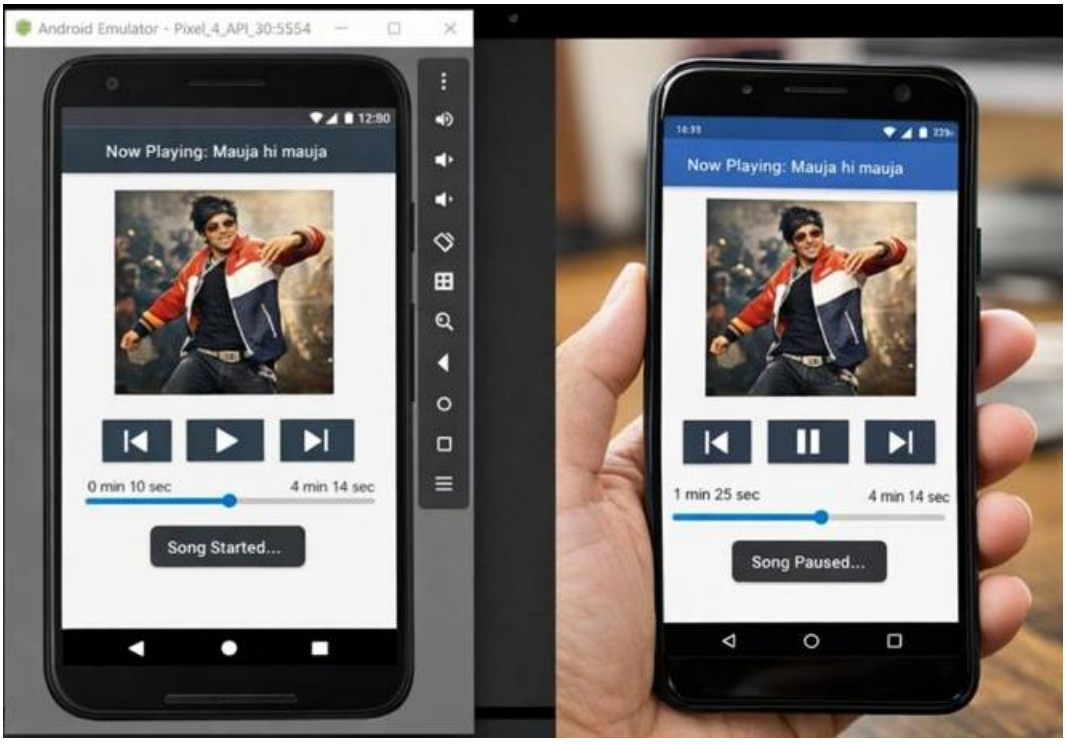
```
package com.DataFlair.mediaplayer;
import android.media.MediaPlayer;
import android.os.Bundle;
import android.os.Handler;
import android.view.View;
import android.widget.ImageButton;
import android.widget.SeekBar;
import android.widget.TextView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
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



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.

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

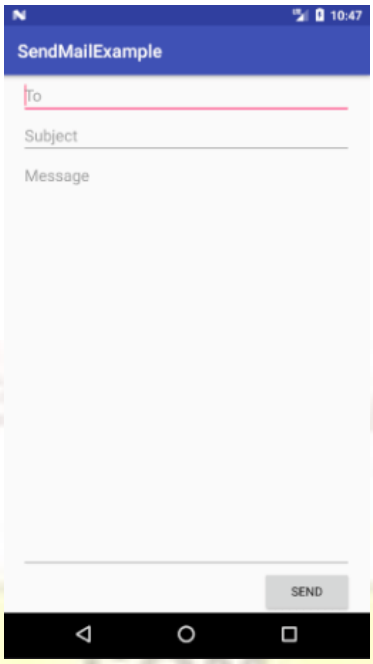
```
package com.example.sendmailexample;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
public class MainActivity extends AppCompatActivity {
   private EditText 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;
```

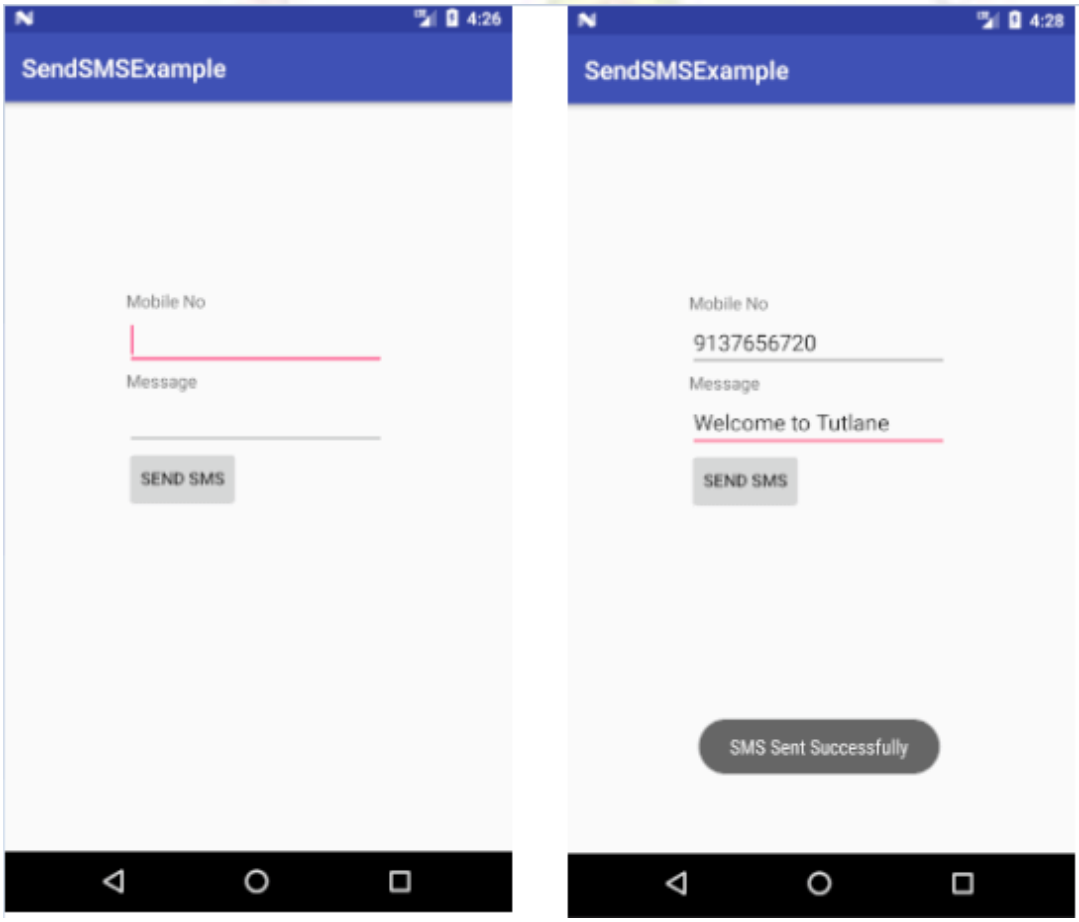
```
public class MainActivity extends AppCompatActivity {

    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



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
```

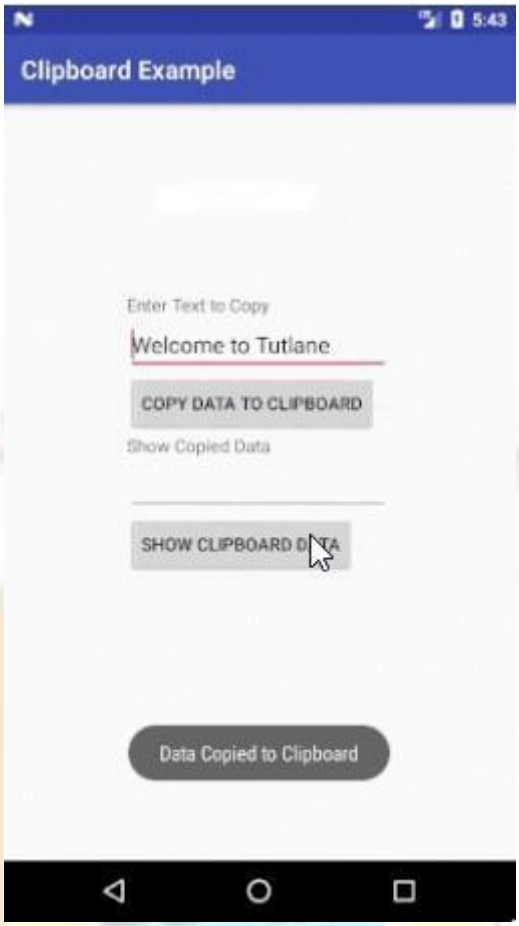
```
        android:id="@+id/btnCopy"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_marginLeft="100dp"
        android:text="Copy Data to Clipboard" />
    <TextView
        android:id="@+id/secTxt"
        android:layout_width="wrap_content"
        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.