SRM VALLIAMMAI ENGINEERINGCOLLEGE

(An Autonomous Institution)

SRM Nagar, Kattankulathur-603203.

DEPARTMENT OF COMPUTER APPLICATIONS

Regulation 2024 - Lab Manual *for* II Semester

MC4267 MOBILE APPLICATION DEVELOPMENT LABORATORY

Academic Year 2024-2025 (EVEN Semester)

Prepared by

Dr. R. Thenmozhi, Associate Professor/ AI-DS

Dr. D. Sridevi, Associate Professor / IT

Mr. K. Maniraj, Assistant Professor/ MCA

Mr. M. Nagarajan, Assistant Professor/ MCA

MC4267	MOBILE APPLICATION DEVELOPMENT LABORATORY	L	T	P	C
		0	0	4	2

COURSE OBJECTIVES:

Students will able to know the

- Installation and configuration of Android sdk
- Application development using Android –layout and event handling
- Application using SQL Lite and Fire base database
- Knowledge about Google play services and geolocation using GPS tracking

LIST OF EXPERIMENTS:

1. Install and configure Java Development Kit (JDK), android studio and android SDK.
2. Develop an application that uses GUI components, fonts and colours.
3. Design an application that uses Layout Managers, Event listeners, Event handling and push notification in Android.
4. Build a simple native calculator application to do simple arithmetic operations.
5. Create animations and graphical primitives in Android environment.
6. Develop an application that makes use of SQL Lite mobile database.
7. Develop an application that makes use of internet for communication using Firebase to send SMS and E-Mail services.
8. Implement an android application that writes data into the SD card and makes use of Notification Manager.
9. Develop a native application that uses Location based services such as GPS tracking, Geo fencing and activity recognition using Google play services.
10. Implement simple gaming application using open-source tools like flutter or Unity.

TOTAL: 60 PERIODS

COURSE OUTCOMES (COs):

At the end of the course, students will be able to

CO1: Design the right user interface for mobile application.

CO2: Implement mobile application using UI toolkits and frameworks.

CO3: Design mobile applications that are aware of the resource constraints of mobile devices.

CO4: Develop web based mobile application that accesses internet and location data.

CO5: Implement android application with multimedia support.

CO's- PO's MAPPING

Course Outcomes	PROGRAM OUTCOMES					
	1	**2**	**3**	**4**	**5**	**6**
CO1	3		3	2	2	2
CO2	3		3	3	2	2
CO3	3		3	3	2	2
CO4	3	1	3	3	2	2
CO5	3		3	3	2	2
AVG	3	0.2	3	2.8	2	2

Software Requirements:

Standalone desktops with Windows or Android or iOS or Equivalent Mobile Application Development Tools with appropriate emulators and debuggers (Android Studio / Eclipse)

PROGRAM EDUCATIONAL OBJECTIVES (PEOs)

1.To prepare students with breadth of knowledge to comprehend, analyze, design and create computing solutions to real-life problems and to excel in industry/ technical profession.

2.To provide students with solid foundation in mathematical and computing fundamentals and techniques required to solve technology related problems and to pursue higher studies and research.

3.To inculcate a professional and ethical attitude in students, to enable them to work towards a broad social context.

4.To empower students with skills required to work as member and leader in multidisciplinary teams and with continuous learning ability on technology and trends needed for a successful career.

PROGRAM OUTCOMES (POs)

After going through the two years of study, our master's in computer applications Graduates will exhibit ability to:

PO#	Programme Outcomes
1.	An ability to independently carry out research/investigation and development work to solve practical problems.
2.	An ability to write and present a substantial technical report/document.
3.	An ability to demonstrate a degree of mastery over design and development of computer applications.
4.	An ability to create, select, adapt and apply appropriate innovative techniques, resources, and modern computing tools to complex computing activities with an understanding of the limitations.
5.	An ability to recognize the need and to engage in independent learning for continual development as a computing professional.
6.	An ability to function effectively as an individual and as a member/leader of a team in various technical environments.

COURSE OUTCOMES:

Course Name: MC4267 MOBILE APPLICATION DEVELOPMENT LABORATORY

Year of study: 2024 –2025

MC4267.1	Design the right user interface for mobile application
MC4267.2	Implement mobile application using UI toolkits and frameworks.
MC4267.3	Design mobile applications that are aware of the resource constraints of mobile devices.
MC4267.4	Develop web based mobile application that accesses internet and location data.
MC4267.5	Implement android application with multimedia support.
MC4267.6	Configure open-source tools like Flutter or Unity.

CO-PO Matrix:

1	3		3	2	2	2
2	3		3	3	2	2
3	3		3	2	2	2
4	3	1	3	2	2	2
5	3		3	3	2	2
Avg	3	1	3	3	2	2

Justification:

Course Outcome	Program Outcome	Value	Justification
MC4267.1	PO1	3	Students will be able to write, debug, and execute programs in a chosen programming language
	PO5	2	Students will develop a strong understanding of core data structures such as arrays, linked lists, stacks, queues, trees, graphs, and hash tables.
MC4267.2	PO1	2	Students will be able to implement and analyze fundamental algorithms for searching, sorting, and traversal.
	PO2	2	Students will enhance their problem-solving skills by applying appropriate data structures and algorithms to solve computational problems.
MC4267.3	PO1	2	Students will gain practical experience by working on projects that require the application of programming and data structure concepts.
	PO2	2	Students will develop collaboration skills by working in teams to complete lab assignments and projects.
MC4267.4	PO1	3	Students will learn effective debugging techniques and how to write test cases to ensure their code is robust and error-free.

MC4267.5	PO1	3	Students will be able to apply theoretical concepts learned in lectures to practical scenarios in the laboratory.
	PO2	3	Students will learn how to optimize code for performance, including memory management and efficient algorithm design.
	PO3	3	Develop the ability to analyze problems, design algorithms, and implement solutions using appropriate data structures and programming techniques.
	PO5	2	Apply data structures to solve real-world problems and understand their practical applications.

CO-PO Average:

CO	PO1	PO2	PO3	PO4	PO5	PO6
MC4267	3	1	3	3	2	2

ASSESSMENT METHOD

EVALUATION PROCEDURE FOR EACH EXPERIMENTS

S.No	Description	Mark
1.	Aim & Pre-Lab discussion	20
2.	Observation	20
3.	Conduction and Execution	30
4.	Output & Result	10
5.	Viva	20
	Total	100

INTERNAL ASSESSMENT FOR LABORATORY

S.No	Description	Mark
1.	Conduction & Execution of Experiment	30
2.	Record	10
3.	Model Test	20
	Total	**60**

TABLE OF CONTENTS

Ex.No.1	Install and configure Java Development Kit (JDK), android studio and android SDK.

Aim:

To Install and configure Java Development Kit (JDK), android studio and android SDK

Procedure:

1. Installing the Java Development Kit

The Android SDK was developed using the Java programming language. Similarly, Android applications are also developed using Java. As a result, the Java Development Kit (JDK) is the first component that must be installed. Android development requires the installation of either version 6 or 7 of the Standard Edition of the Java Platform Development Kit. Java is provided in both development (JDK) and runtime (JRE) packages. For the purposes of Android development, the JDK must be installed.

2. Downloading the Android Studio Package

Most of the work involved in developing applications for Android will be performed using the Android Studio environment. Android Studio may be downloaded from the following web page:

http://developer.android.com/sdk/index.html

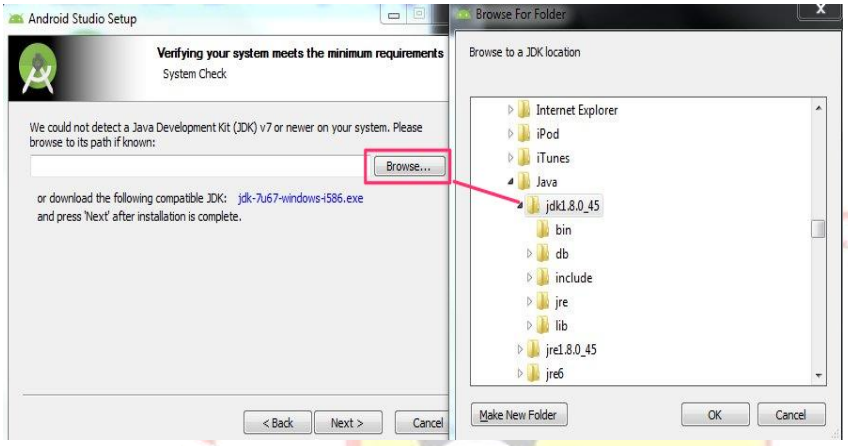
From this page, either click on the download button if it lists the correct platform (for example on a Windows based web browser the button will read "Download Android Studio for Windows"), or select the "Other Download Options" link to manually select the appropriate package for your platform and operating system. On the subsequent screen, accept the terms and conditions to initiate the download.

3. Installing Android Studio

Locate the downloaded Android Studio installation executable file (named *android-studio-bundle-<version>*.exe) in a Windows Explorer window and double click on it to start the installation process, clicking the *Yes* button in the User Account Control dialog if it appears.

Once the Android Studio setup wizard appears, work through the various screens to configure the installation to meet your requirements in terms of the file system location into which Android Studio should be installed and whether or not it should be made available to other users of the system. Although there are no strict rules on where Android Studio should be installed on the system, the remainder of this book will assume that the installation was performed into a sub-folder of the user's home directory named *android-studio*. Once the options have been configured, click on the *Install* button to begin the installation process.


Android Studio Setup
Welcome to Android Studio Setup
Setup will guide you through the installation of Android Studio.
It is recommended that you close all other applications before starting Setup. This will make it possible to update relevant system files without having to reboot your computer.
Click Next to continue.
Android Studio
< Back
Next >
Cancel

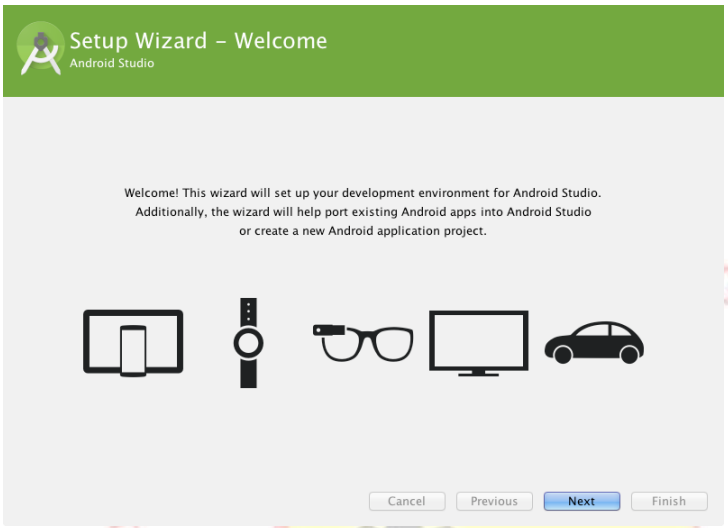

Android Studio Setup
Verifying your system meets the minimum requirements
System Check
We could not detect a Java Development Kit (JDK) v7 or newer on your system. Please browse to its path if known:
Browse...
or download the following compatible JDK: jdk-7u67-windows-i586.exe
and press 'Next' after installation is complete.
< Back
Next >
Cancel
Browse For Folder
Browse to a JDK location
Internet Explorer
iPod
iTunes
Java
jdk1.8.0_45
bin
db
include
jre
lib
jre1.8.0_45
jre6
Make New Folder
OK
Cancel


Android Studio Setup
Completing the Android Studio Setup
Android Studio has been installed on your computer.
Click Finish to close Setup.
Start Android Studio
Android Studio
< Back
Finish
Cancel

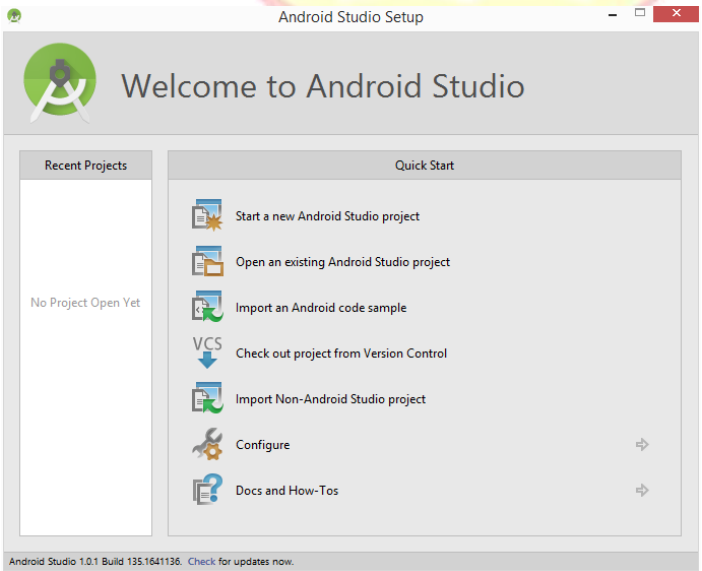
4. The Android Studio Setup Wizard

The first time that Android Studio is launched after being installed, a dialog will appear providing the option to import settings from a previous Android Studio version. If you have settings from a previous version and would like to import them into the latest installation, select the appropriate option and location. Alternatively, indicate that you do not need to import any previous settings and click on the OK button to proceed.

After Android Studio has finished loading, the setup wizard will appear as shown



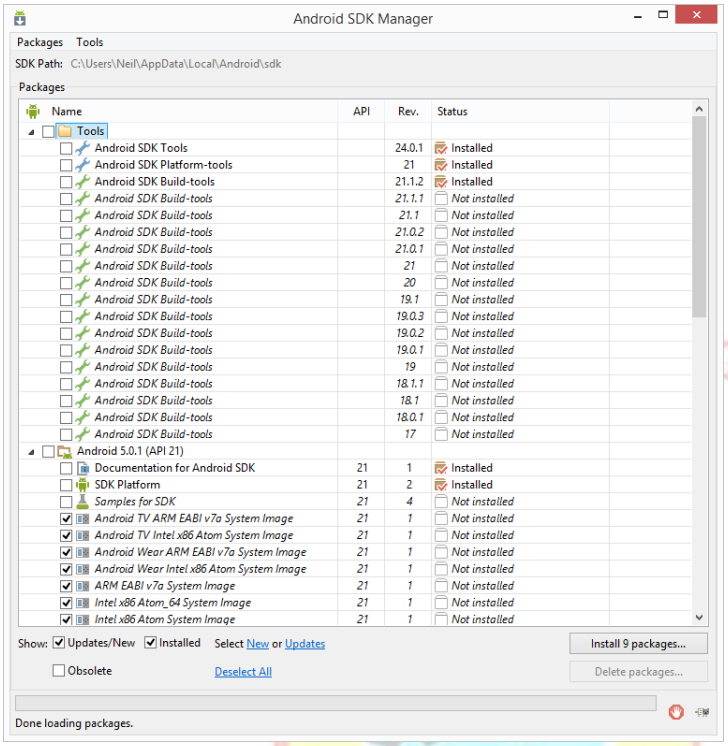
Click on the Next button, choose the Standard installation option and click on Next once again. On the license agreement screen, select and accept each of the licenses listed before clicking on Finish to complete the setup process. The Welcome to Android Studio screen should then appear:



5. Installing the Latest Android SDK Packages

The steps performed so far have installed Java, the Android Studio IDE and the current set of default Android SDK packages. Before proceeding, it is worth taking some time to verify which packages are installed and to install any missing packages.

This task can be performed using the *Android SDK Manager*, which may be launched from within the Android Studio tool by selecting the *Configure -> SDK Manager* option from within the Android Studio welcome dialog. Once invoked, the SDK Manager tool will appear as illustrated in Figure



Within the Android SDK Manager, make sure that the following packages are listed as *Installed* in the Status column:

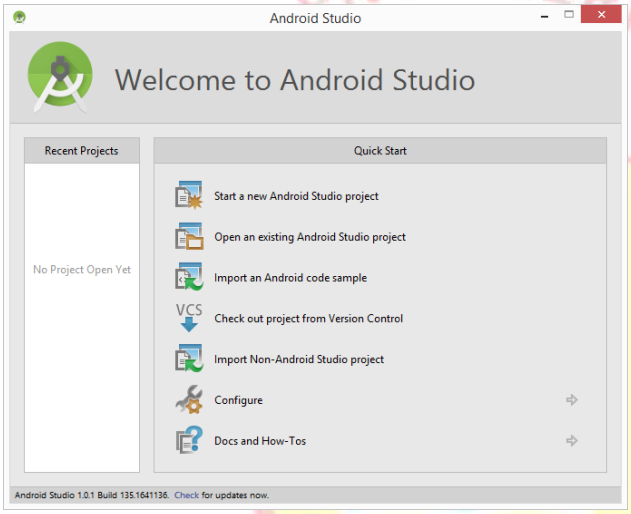
- Tools > Android SDK Tools
- Tools > Android SDK Platform-tools
- Tools > Android SDK Build-tools
- SDK Platform (most recent version) > SDK Platform
- SDK Platform (most recent version) > ARM EABI v7a System Image
- Extras > Android Support Repository
- Extras > Android Support Library
- Extras > Google Repository
- Extras > Google USB Driver (Required on Windows systems only)
- Extras > Intel x86 Emulator Accelerator (HAXM installer)

In the event that any of the above packages are listed as *Not Installed*, simply select the checkboxes next to those packages and click on the *Install packages* button to initiate the installation process. In the resulting dialog, accept the license agreements before clicking on the *Install* button. The SDK Manager will then begin to download and install the designated packages. As the installation proceeds, a progress bar will appear at the bottom of the manager window indicating the status of the installation.

Once the installation is complete, review the package list and make sure that the selected packages are now listed as *Installed*in the *Status* column. If any are listed as *Not installed,* make sure they are selected and click on the *Install packages…* button again.

6. Creating a New Android Project

The first step in the application development process is to create a new project within the Android Studio environment. Begin, therefore, by launching Android Studio so that the "Welcome to Android Studio" screen appears as illustrated in Figure

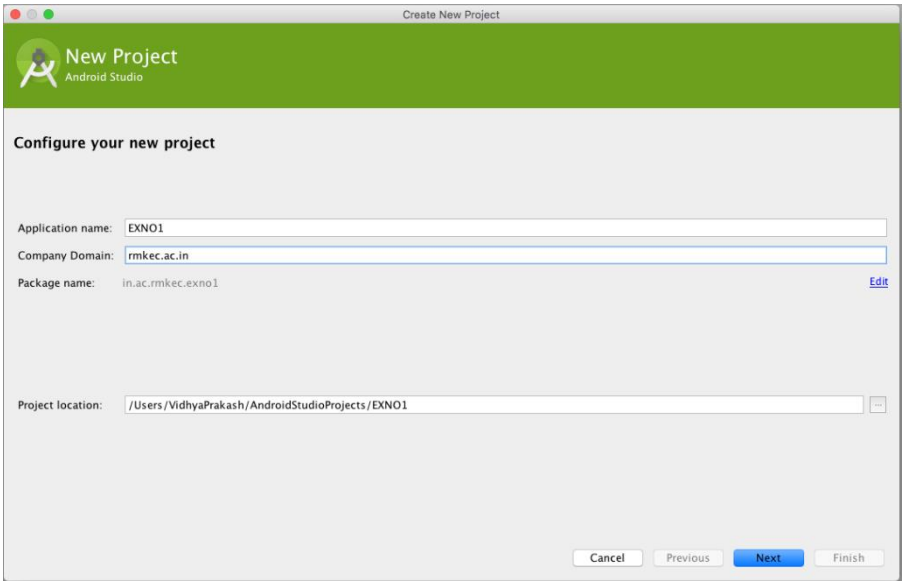


Once this window appears, Android Studio is ready for a new project to be created. To create the new project, simply click on the *Start a new Android Studio project* option to display the first screen of the *New Project* wizard as shown in Figure

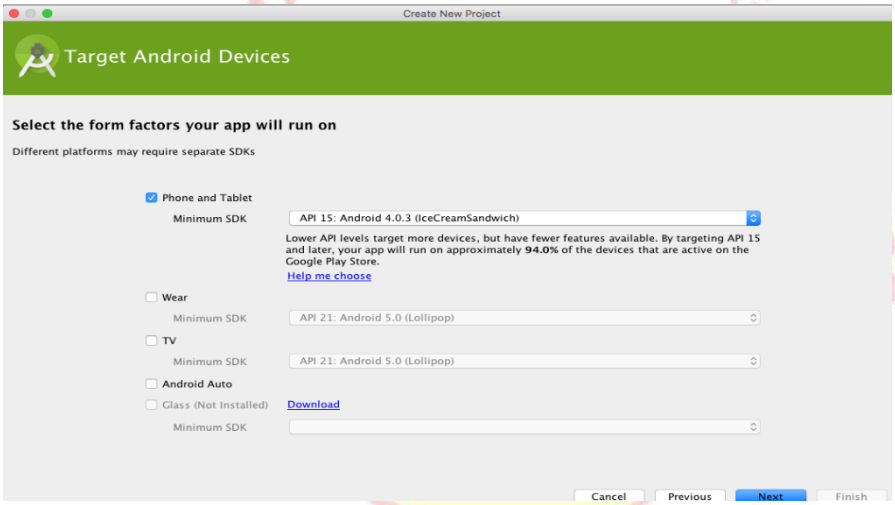
7. Defining the Project and SDK Settings

In the *New Project* window, set the *Application name* field to *EXNO1*. The application name is the name by which the application will be referenced and identified within Android Studio and is also the name that will be used when the completed application goes on sale in the Google Play store.

The *Package Name* is used to uniquely identify the application within the Android application ecosystem. It should be based on the reversed URL of your domain name followed by the name of the application. For example, if your domain is *rmkec.ac.in*, and the application has been named *EXNO1*, then the package name might be specified as follows:



Next select Android package SDK we need to build



Result:

The Java Development Kit (JDK), android studio and android SDK are installed.

Ex.No.2 **Develop an application that uses GUI components, Font and Colours**

Aim:

To develop an application that uses GUI components,Fonts and Colors.

Procedure:

Step 1: File→ New → Android Project Application

Step 2: Give the Project Name → Next

Step 3: Go to Res Folder and Select Layout Click the Main.xml

Step 4: Design the Graphical Layout

Step 5: Write code the in MainActivity.java

Step 6: Run the code using AVD Emulator

ActivityMain.Java

```
packagecom.example.text;
importandroid.app.Activity;
importandroid.graphics.Color;
importandroid.os.Bundle;
importandroid.view.Menu;
importandroid.view.View;
importandroid.view.View.OnClickListener;
importandroid.widget.Button;
importandroid.widget.TextView;

public class MainActivity extends Activity {
        Button b1,b2;
        TextView t1;
        float font;
        inti=1;


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

```
        b1=(Button)findViewById(R.id.button1);
        t1=(TextView)findViewById(R.id.textView1);
    b1.setOnClickListener(new OnClickListener() {

                        @Override
                        public void onClick(View arg0) {
                                // TODO Auto-generated method stub
                                t1.setTextSize(font);
                                font=font+4;
                                if(font==40)
                                        font=20;

                        }
                });

        b2=(Button)findViewById(R.id.button2);
b2.setOnClickListener(new OnClickListener() {

                        @Override
                        public void onClick(View v) {
                                // TODO Auto-generated method stub
                                switch (i) {
                                case 1:
                                        t1.setTextColor(Color.parseColor("#0000FF"));
                                        break;

                                case 2:
                                        t1.setTextColor(Color.parseColor("#00FF00"));
                                        break;

                                case 3:
                                        t1.setTextColor(Color.parseColor("#FF0000"));
                                        break;
                                default:
                                        t1.setTextColor(Color.parseColor("#800000"));
                                        break;

                                }
```

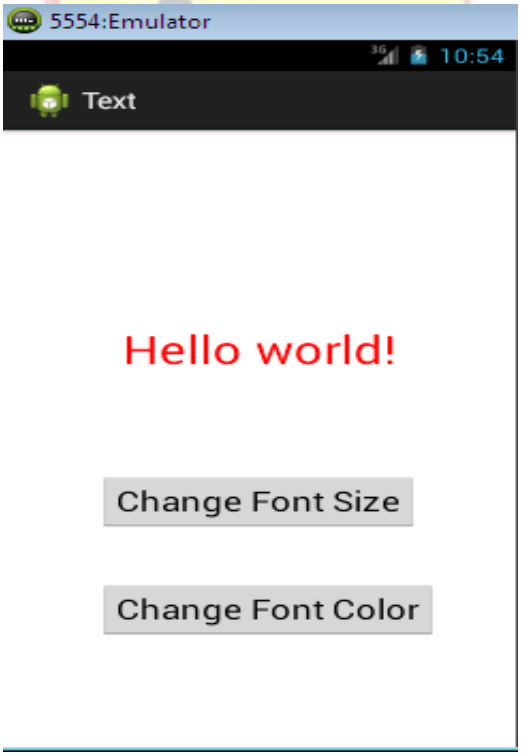
```
                    i++;
                    if(i==5)
                        i=1;
                    }
            });

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

}
```

activity_main.xml

```
<RelativeLayoutxmlns: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="140dp"
android:text="@string/hello_world" />

<Button
android:id="@+id/button1"
```

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@+id/textView1"
android:layout_centerHorizontal="true"
android:layout_marginTop="78dp"
android:text="Change Font Size"
android:textSize="23sp"/>

<Button
android:id="@+id/button2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/button1"
android:layout_below="@+id/button1"
android:layout_marginTop="40dp"
android:text="Change Font Color"
android:textSize="23sp" />
</RelativeLayout>
```

Output



Result:

Thus the program to develop an application that uses GUI components, Fonts and Colors is executed successfully.

Ex.No.3	Design an application that uses Layout Managers, Event listeners, Event handling and push notification in Android

Aim:

To develop an application that uses Layout Managers and Event listeners.

Procedure:

Step 1: File→ New → Android Project Application

Step 2: Specify Application Name

Step 3: Design the Acitivity_main.xml with EditText and Command Button

Step 4: In AcitivityMain.java declare the necessary variable and specify the name to the controls.

Step 5: Get the Input and Convert it to Value

Step6 : Print the result using Toast Class

activity_main.xml

```
<RelativeLayoutxmlns: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">

<EditText
android:id="@+id/editText1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_alignParentTop="true"
android:layout_marginLeft="56dp"
android:layout_marginTop="48dp"
android:ems="10"
android:text="No 1">
```

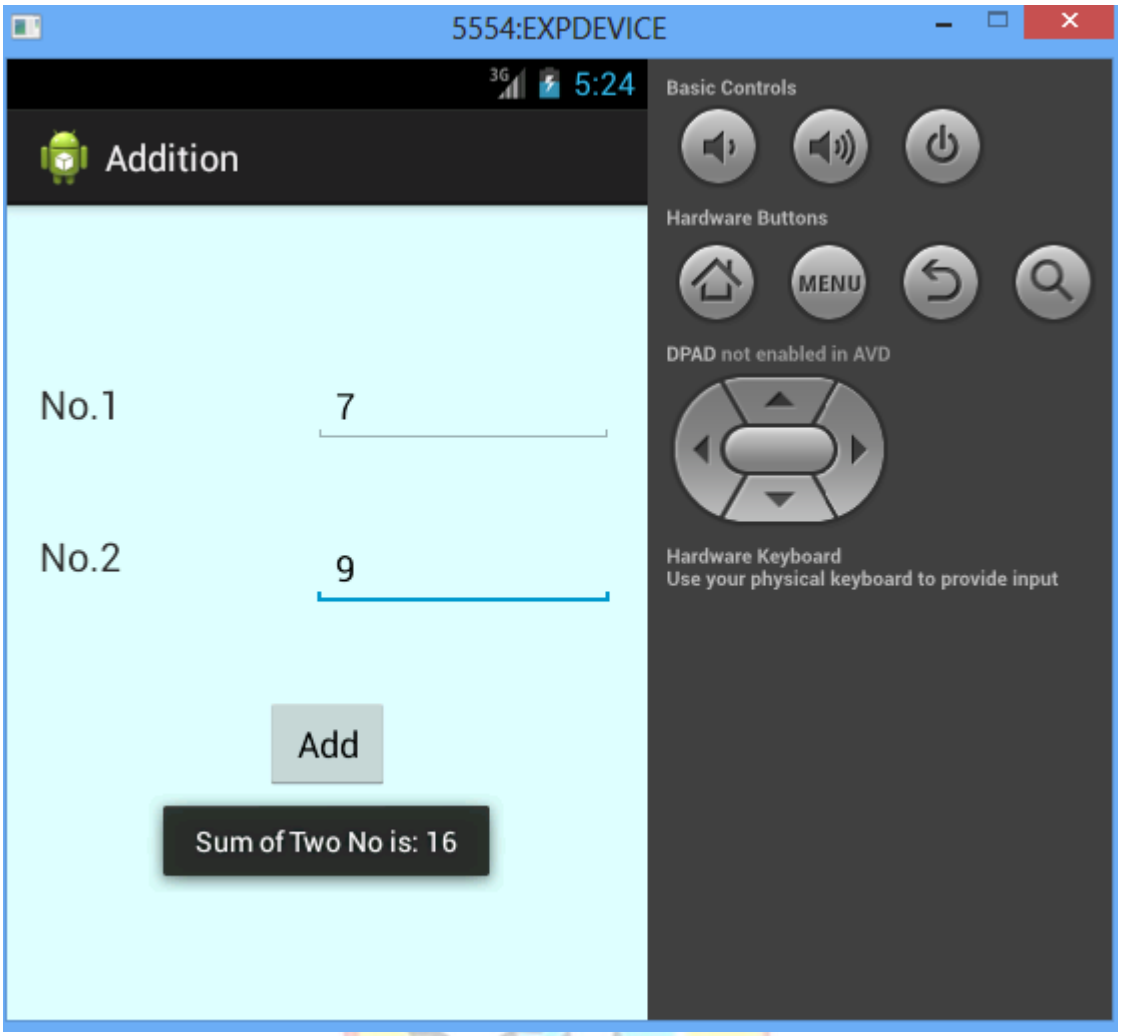
```
<requestFocus/>
</EditText>

<EditText
android:id="@+id/editText2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/editText1"
android:layout_below="@+id/editText1"
android:layout_marginTop="41dp"
android:ems="10"
android:text="No 2"/>

<Button
android:id="@+id/button1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@+id/editText2"
android:layout_centerHorizontal="true"
android:layout_marginTop="44dp"
android:text="Add"/>
/RelativeLayout>
```

Activity Main.java

```
packagecom.example.addition;
importandroid.R.integer;
importandroid.R.string;
importandroid.app.Activity;
importandroid.os.Bundle;
importandroid.view.Menu;
importandroid.view.View;
importandroid.view.View.OnClickListener;
importandroid.widget.Button;
importandroid.widget.EditText;
importandroid.widget.Toast;
```

```
publicclassMainActivityextends Activity {
        EditTextt1,t2;
        Button b1;

        @Override
        protectedvoidonCreate(Bundle savedInstanceState) {
                super.onCreate(savedInstanceState);
                setContentView(R.layout.activity_main);
                t1=(EditText)findViewById(R.id.editText1);
                t2=(EditText)findViewById(R.id.editText2);
                b1=(Button)findViewById(R.id.button1);
                b1.setOnClickListener(newOnClickListener() {

                        @Override
                        publicvoidonClick(View arg0) {
                                // TODO Auto-generated method stub
                                String v1=t1.getText().toString();
                                String v2=t2.getText().toString();

                                int a = Integer.parseInt(v1);
                                int b=Integer.parseInt(v2);
                                int sum=a+b;
        Toast.makeText(MainActivity.this, "Sum of Two No"+sum, 5000).show();
                                        }

                });
        }

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

}
```

Output



Result:

Thus the program to develop an application that uses Layout Managers and Event listeners is executed successfully.

Ex.No.4	Build a simple native calculator application to do simple arithmetic operations.

Aim:

To develop a native calculator application.

Procedure:

Step 1: File→ New → Android Project Application

Step 2: Specify Application Name

Step 3: Design the Acitivity_main.xml with EditText and Command Button

Step 4: In AcitivityMain.java declare the necessary variable and specify the name to the controls.

Step 5: Get the Input and Convert it to Value

Step6 : Print the result using Toast Class

/* Native Calculator Program-MainActivity.xml */

```
<RelativeLayoutxmlns: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:text="Native Calculator Apps"/>

<EditText
android:id="@+id/editText1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/textView1"
android:layout_below="@+id/textView1"
```

```
android:layout_marginTop="32dp"
android:ems="10">
<requestFocus/>
</EditText>

<Button
android:id="@+id/button1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignLeft="@+id/editText1"
android:layout_below="@+id/editText1"
android:layout_marginTop="15dp"
android:text="1"/>

<Button
android:id="@+id/button2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/button1"
android:layout_alignBottom="@+id/button1"
android:layout_toRightOf="@+id/button1"
android:text="2"/>

<Button
android:id="@+id/button3"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/button2"
android:layout_alignBottom="@+id/button2"
android:layout_toRightOf="@+id/button2"
android:text="3"/>

<Button
android:id="@+id/button4"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_below="@+id/button1"
android:layout_toLeftOf="@+id/button2"
android:text="4"/>
```

```
<Button
android:id="@+id/button5"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/button4"
android:layout_alignBottom="@+id/button4"
android:layout_toRightOf="@+id/button4"
android:text="5"/>

<Button
android:id="@+id/button6"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/button5"
android:layout_alignBottom="@+id/button5"
android:layout_alignLeft="@+id/button3"
android:text="6"/>

<Button
android:id="@+id/button7"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_below="@+id/button4"
android:text="7"/>

<Button
android:id="@+id/button8"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/button7"
android:layout_alignBottom="@+id/button7"
android:layout_toRightOf="@+id/button7"
android:text="8"/>

<Button
android:id="@+id/button9"
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/button8"
android:layout_alignBottom="@+id/button8"
android:layout_toRightOf="@+id/button8"
android:text="9"/>

<Button
android:id="@+id/button10"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignParentLeft="true"
android:layout_below="@+id/button7"
android:text="0"/>

<Button
android:id="@+id/button11"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/button10"
android:layout_alignBottom="@+id/button10"
android:layout_alignLeft="@+id/button8"
android:text="."/>

<Button
android:id="@+id/button12"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/button11"
android:layout_alignBottom="@+id/button11"
android:layout_toRightOf="@+id/button11"
android:text="="/>

<Button
android:id="@+id/button13"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/button3"
android:layout_alignBottom="@+id/button3"
android:layout_toRightOf="@+id/button3"
```

```
android:text="+"/>

<Button
android:id="@+id/button14"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/button6"
android:layout_alignBottom="@+id/button6"
android:layout_alignLeft="@+id/button13"
android:text="-"/>

<Button
android:id="@+id/button15"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/button9"
android:layout_alignBottom="@+id/button9"
android:layout_toRightOf="@+id/button9"
android:text="*"/>

<Button
android:id="@+id/button16"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignBaseline="@+id/button12"
android:layout_alignBottom="@+id/button12"
android:layout_alignLeft="@+id/button15"
android:text="/"/>
<Button
android:id="@+id/button17"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_alignTop="@+id/editText1"
android:layout_toRightOf="@+id/editText1"
android:text="C"/>
</RelativeLayout>
```

/* Native Calculator Program-MainActivity.java*/

```
packagecom.example.calculator;
importandroid.os.Bundle;
importandroid.app.Activity;
importandroid.view.Menu;
importandroid.view.View;
importandroid.widget.Button;
importandroid.widget.EditText;

public class MainActivity extends Activity {

Button b1,b2,b3,b4,b5,b6,b7,b8,b9,b0,bdot,badd,bsub,bmul,bdiv,beq, bC;
EditTextet;
int val1,val2;
booleanadd,sub,div,mul;

   @Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
      b1=(Button) findViewById(R.id.button1);
      b2=(Button) findViewById(R.id.button2);
      b3=(Button) findViewById(R.id.button3);
      b4=(Button) findViewById(R.id.button4);
      b5=(Button) findViewById(R.id.button5);
      b6=(Button) findViewById(R.id.button6);
      b7=(Button) findViewById(R.id.button7);
      b8=(Button) findViewById(R.id.button8);
      b9=(Button) findViewById(R.id.button9);
      b0=(Button) findViewById(R.id.button10);
bdot=(Button) findViewById(R.id.button11);
beq=(Button) findViewById(R.id.button12);
badd=(Button) findViewById(R.id.button13);
bsub=(Button) findViewById(R.id.button14);
bmul=(Button) findViewById(R.id.button15);
bdiv=(Button) findViewById(R.id.button16);
bC=(Button)findViewById(R.id.button17);
et=(EditText) findViewById(R.id.editText1);
```

```
b1.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View arg0) {
                        // TODO Auto-generated method stub
                        et.setText(et.getText()+"1");
                }
        });
b2.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View arg0) {
                        // TODO Auto-generated method stub
                        et.setText(et.getText()+"2");
                }
        });
b3.setOnClickListener(new View.OnClickListener() {
            @Override
                public void onClick(View arg0) {
                        // TODO Auto-generated method stub
                        et.setText(et.getText()+"3");
                }
        });
b4.setOnClickListener(new View.OnClickListener() {

                @Override
                public void onClick(View arg0) {
                        // TODO Auto-generated method stub
                        et.setText(et.getText()+"4");
                }
        });
b5.setOnClickListener(new View.OnClickListener() {

                @Override
                public void onClick(View arg0) {
                        // TODO Auto-generated method stub
                        et.setText(et.getText()+"5");
                }
        });
b6.setOnClickListener(new View.OnClickListener() {
```

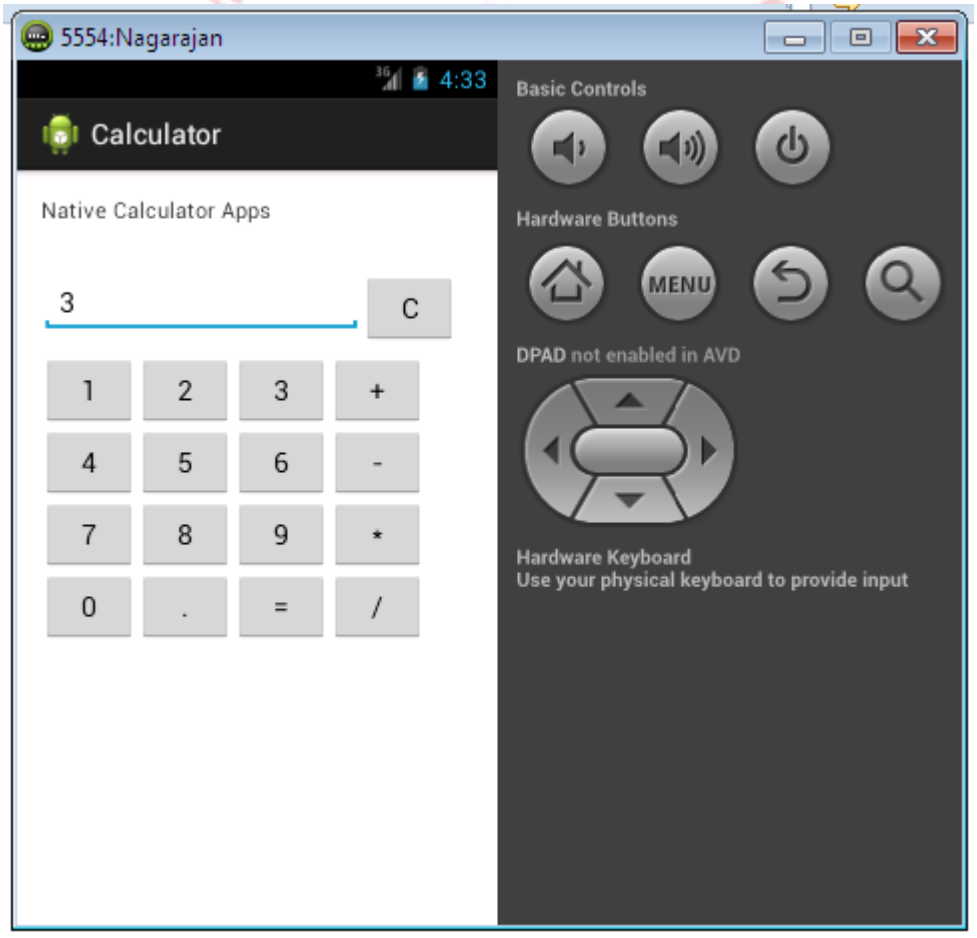
```
                @Override
                public void onClick(View arg0) {
                        // TODO Auto-generated method stub
                        et.setText(et.getText()+"6");
                }
        });
b7.setOnClickListener(new View.OnClickListener() {

                @Override
                public void onClick(View arg0) {
                        // TODO Auto-generated method stub
                        et.setText(et.getText()+"7");
                }
        });
b8.setOnClickListener(new View.OnClickListener() {

                @Override
                public void onClick(View arg0) {
                        // TODO Auto-generated method stub
                        et.setText(et.getText()+"8");
                }
        });
b9.setOnClickListener(new View.OnClickListener() {

                @Override
                public void onClick(View arg0) {
                        // TODO Auto-generated method stub
                        et.setText(et.getText()+"9");
                }
        });
b0.setOnClickListener(new View.OnClickListener() {

                @Override
                public void onClick(View arg0) {
                        // TODO Auto-generated method stub
                        et.setText(et.getText()+"10");
                }
        });
bdot.setOnClickListener(new View.OnClickListener() {
```

```
                @Override
                public void onClick(View arg0) {
                        // TODO Auto-generated method stub
                        et.setText(et.getText()+".");
                }
        });
badd.setOnClickListener(new View.OnClickListener() {

                @Override
                public void onClick(View arg0) {
                        // TODO Auto-generated method stub
                        val1=Integer.parseInt(et.getText()+"");
                        add=true;
                        et.setText(null);
                }
        });
bsub.setOnClickListener(new View.OnClickListener() {

                @Override
                public void onClick(View arg0) {
                        // TODO Auto-generated method stub
                        val1=Integer.parseInt(et.getText()+"");
                        sub=true;
                        et.setText(null);

                         }
        });
bmul.setOnClickListener(new View.OnClickListener() {

                @Override
                public void onClick(View arg0) {
                        // TODO Auto-generated method stub
                        val1=Integer.parseInt(et.getText()+"");
                        mul=true;
                        et.setText(null);

                         }
        });
```

```
bdiv.setOnClickListener(new View.OnClickListener() {

		@Override
		public void onClick(View arg0) {
			// TODO Auto-generated method stub
			val1=Integer.parseInt(et.getText()+"");
			div=true;
			et.setText(null);

			 }
	});
beq.setOnClickListener(new View.OnClickListener() {

		@Override
		public void onClick(View arg0) {
			// TODO Auto-generated method stub
			val2=Integer.parseInt(et.getText()+"");

			if (add==true) {
			et.setText(val1+val2+"");
			add=false;
			 }
			if (sub==true) {
			et.setText(val1-val2+"");
			sub=false;
			 }
			if (mul==true) {
			et.setText(val1*val2+"");
			mul=false;
			 }
			if (div==true) {
			et.setText(val1/val2+"");
			div=false;
			 }
			 }
		});
bC.setOnClickListener(new View.OnClickListener() {
		@Override
		public void onClick(View arg0) {
```

```
                // TODO Auto-generated method stub
                et.setText("");
            }
        });

}
    @Override
publicbooleanonCreateOptionsMenu(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



Result:

Thus the program to develop a native calculator application is executed successfully.

Ex. No.5	Create animations and graphical primitives in Android environment

5. (a) Draw a Rectangle

Aim:

To develop an application to draw basic graphical primitives like rectangle on the screen.

Procedure:

Step 1: File→ New → Android Project Application

Step 2: Specify Application Name

Step 3: use paint class

Step 4: Specify the stroke size & color

Step 5: Specify Rectangle object size

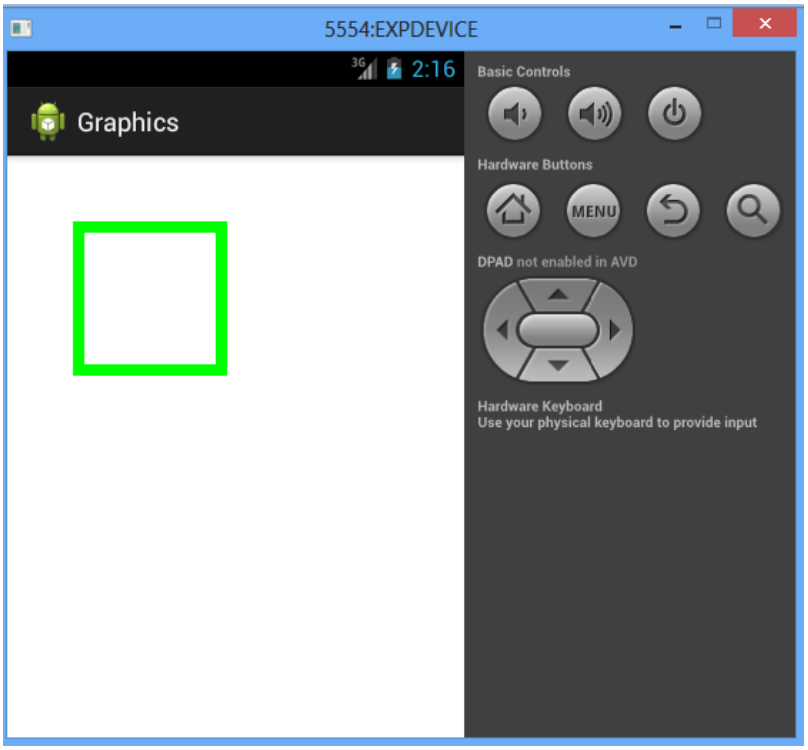
Step6 : Run the code using AVD Emulator

/* Program to Draw a Rectangle Using the Paint Class in Android */

```
package com.example.graphics;
import android.app.Activity;
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.os.Bundle;
import android.view.View;

public class MainActivity  extends Activity {

 @Override
 public void onCreate(Bundle savedInstanceState) {
       super.onCreate(savedInstanceState);
       setContentView(new myView(this));
 }

 private class myView extends View{

          public myView(Context context) {
    super(context);
```

```
}

    @Override
    protected void onDraw(Canvas canvas) {

        Paint myPaint = new Paint();
        myPaint.setColor(Color.GREEN);
        myPaint.setStyle(Paint.Style.STROKE);
        myPaint.setStrokeWidth(3);
        canvas.drawRect(20, 20, 100, 100, myPaint);
          } } }
```

Output:



Result:

Thus the program to develop an application to draw basic graphical primitives like rectangle on the screen is executed successfully

5. (b) Draw a Circle

Aim:

To develop an application to draw basic graphical primitives like circle on the screen.

Procedure:

Step 1: File→ New → Android Project Application

Step 2: Specify Application Name

Step 3: use paint class and Specify the style and color

Step 4: Specify the Circle Height, Width and Radios

Step 5: Specify Rectangle object size

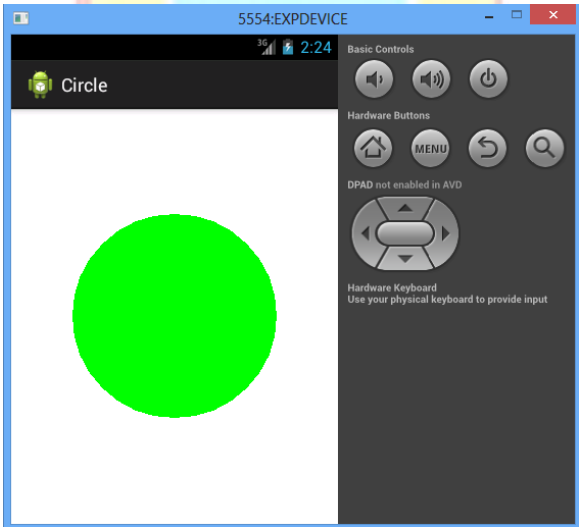
Step6 : Run the code using AVD Emulator

/* Program to Draw a Circle Using the Paint Class in Android */

```
package com.example.circle;

import android.app.Activity;
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.os.Bundle;
import android.view.View;


public class MainActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(new MyView(this));
    }
    public class MyView extends View {
        public MyView(Context context) {
            super(context);
            // TODO Auto-generated constructor stub
        }
```

```
        @Override
        protected void onDraw(Canvas canvas) {
          // TODO Auto-generated method stub
          super.onDraw(canvas);
          int x = getWidth();
          int y = getHeight();
          int radius;
          radius = 100;
          Paint paint = new Paint();
          paint.setStyle(Paint.Style.FILL);
          paint.setColor(Color.WHITE);
          canvas.drawPaint(paint);
          // Use Color.parseColor to define HTML colors
          paint.setColor(Color.parseColor("#00FF00"));
          canvas.drawCircle(x / 2, y / 2, radius, paint);
        }
    }
}
```

Output:



Result:

Thus the program to develop an application to draw basic graphical primitives like circle on the screen is executed successfully.

Ex. No. 6	Develop an application that makes use of SQL Lite mobile database.

Aim:

To develop an application that makes use of database.

Procedure:

Step 1: File→ New → Android Project Application

Step 2: Specify Application Name

Step 3: Create a layout using the widgets i.e. use Buttons and also textviews.

Step 4: Write the code to make use of database.

Step 5: Write the code to perform insertion, deletion, updation, viewing files and to view all files.

Step 6: Run the code as Android Application

Step 7: Check the actions in the database.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<AbsoluteLayout xmlns:android="http://schemas.android.com/apk/res/android"
   android:layout_width="match_parent"
   android:layout_height="match_parent">
   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_x="50dp"
      android:layout_y="20dp"
      android:text="Student Details"
      android:textSize="30sp" />

   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:layout_x="20dp"
```

```
    android:layout_y="110dp"
    android:text="Enter Rollno:"
    android:textSize="20sp" />

<EditText
    android:id="@+id/Rollno"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="175dp"
    android:layout_y="100dp"
    android:inputType="number"
    android:textSize="20sp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="20dp"
    android:layout_y="160dp"
    android:text="Enter Name:"
    android:textSize="20sp" />

<EditText
    android:id="@+id/Name"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="175dp"
    android:layout_y="150dp"
    android:inputType="text"
    android:textSize="20sp" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_x="20dp"
    android:layout_y="210dp"
    android:text="Enter Marks:"
```

```
    android:textSize="20sp" />

<EditText
    android:id="@+id/Marks"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="175dp"
    android:layout_y="200dp"
    android:inputType="number"
    android:textSize="20sp" />

<Button
    android:id="@+id/Insert"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="25dp"
    android:layout_y="300dp"
    android:text="Insert"
    android:textSize="30dp" />

<Button
    android:id="@+id/Delete"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="200dp"
    android:layout_y="300dp"
    android:text="Delete"
    android:textSize="30dp" />

<Button
    android:id="@+id/Update"
    android:layout_width="150dp"
    android:layout_height="wrap_content"
    android:layout_x="25dp"
    android:layout_y="400dp"
    android:text="Update"
```

```
        android:textSize="30dp" />

    <Button
        android:id="@+id/View"
        android:layout_width="150dp"
        android:layout_height="wrap_content"
        android:layout_x="200dp"
        android:layout_y="400dp"
        android:text="View"
        android:textSize="30dp" />

    <Button
        android:id="@+id/ViewAll"
        android:layout_width="200dp"
        android:layout_height="wrap_content"
        android:layout_x="100dp"
        android:layout_y="500dp"
        android:text="View All"
        android:textSize="30dp" />

</AbsoluteLayout>
```

Main Activity .java

```
package com.example.database;
import android.app.Activity;
import android.app.AlertDialog.Builder;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
```

```
public class MainActivity extends Activity implements OnClickListener
{
    EditText Rollno,Name,Marks;
    Button Insert,Delete,Update,View,ViewAll;
    SQLiteDatabase db;
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        Rollno=(EditText)findViewById(R.id.Rollno);
        Name=(EditText)findViewById(R.id.Name);
        Marks=(EditText)findViewById(R.id.Marks);
        Insert=(Button)findViewById(R.id.Insert);
        Delete=(Button)findViewById(R.id.Delete);
        Update=(Button)findViewById(R.id.Update);
        View=(Button)findViewById(R.id.View);
        ViewAll=(Button)findViewById(R.id.ViewAll);

        Insert.setOnClickListener(this);
        Delete.setOnClickListener(this);
        Update.setOnClickListener(this);
        View.setOnClickListener(this);
        ViewAll.setOnClickListener(this);

        // Creating database and table
        db=openOrCreateDatabase("StudentDB", Context.MODE_PRIVATE, null);
        db.execSQL("CREATE TABLE IF NOT EXISTS student(rollno VARCHAR,name
VARCHAR,marks VARCHAR);");
    }
    public void onClick(View view)
    {
        // Inserting a record to the Student table
```

```
        if(view==Insert)
        {
            // Checking for empty fields
            if(Rollno.getText().toString().trim().length()==0||
                    Name.getText().toString().trim().length()==0||
                    Marks.getText().toString().trim().length()==0)
            {
                showMessage("Error", "Please enter all values");
                return;
            }
            db.execSQL("INSERT INTO student VALUES('"+Rollno.getText()+"','"+Name.getText()+
                    "','"+Marks.getText()+"');");
            showMessage("Success", "Record added");
            clearText();
        }
        // Deleting a record from the Student table
        if(view==Delete)
        {
            // Checking for empty roll number
            if(Rollno.getText().toString().trim().length()==0)
            {
                showMessage("Error", "Please enter Rollno");
                return;
            }
            Cursor c=db.rawQuery("SELECT * FROM student WHERE
rollno='"+Rollno.getText()+"'", null);
            if(c.moveToFirst())
            {
                db.execSQL("DELETE FROM student WHERE rollno='"+Rollno.getText()+"'");
                showMessage("Success", "Record Deleted");
            }
            else
            {
                showMessage("Error", "Invalid Rollno");
            }
            clearText();
```

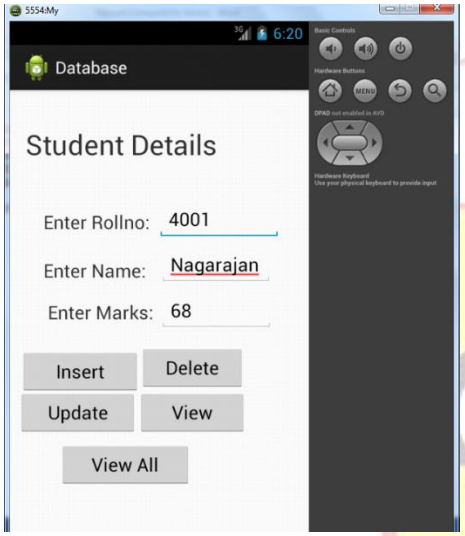
```
        }
        // Updating a record in the Student table
        if(view==Update)
        {
            // Checking for empty roll number
            if(Rollno.getText().toString().trim().length()==0)
            {
                showMessage("Error", "Please enter Rollno");
                return;
            }
            Cursor c=db.rawQuery("SELECT * FROM student WHERE
rollno='"+Rollno.getText()+"'", null);
            if(c.moveToFirst()) {
                db.execSQL("UPDATE student SET name='" + Name.getText() + "',marks='" +
Marks.getText() +
                        "' WHERE rollno='"+Rollno.getText()+"'");
                showMessage("Success", "Record Modified");
            }
            else {
                showMessage("Error", "Invalid Rollno");
            }
            clearText();
        }
        // Display a record from the Student table
        if(view==View)
        {
            // Checking for empty roll number
            if(Rollno.getText().toString().trim().length()==0)
            {
                showMessage("Error", "Please enter Rollno");
                return;
            }
            Cursor c=db.rawQuery("SELECT * FROM student WHERE
rollno='"+Rollno.getText()+"'", null);
            if(c.moveToFirst())
            {
```

```
            Name.setText(c.getString(1));
            Marks.setText(c.getString(2));
        }
        else
        {
            showMessage("Error", "Invalid Rollno");
            clearText();
        }
    }
    // Displaying all the records
    if(view==ViewAll)
    {
        Cursor c=db.rawQuery("SELECT * FROM student", null);
        if(c.getCount()==0)
        {
            showMessage("Error", "No records found");
            return;
        }
        StringBuffer buffer=new StringBuffer();
        while(c.moveToNext())
        {
            buffer.append("Rollno: "+c.getString(0)+"\n");
            buffer.append("Name: "+c.getString(1)+"\n");
            buffer.append("Marks: "+c.getString(2)+"\n\n");
        }
        showMessage("Student Details", buffer.toString());
    }
}
public void showMessage(String title,String message)
{
    Builder builder=new Builder(this);
    builder.setCancelable(true);
    builder.setTitle(title);
    builder.setMessage(message);
    builder.show();
}
```

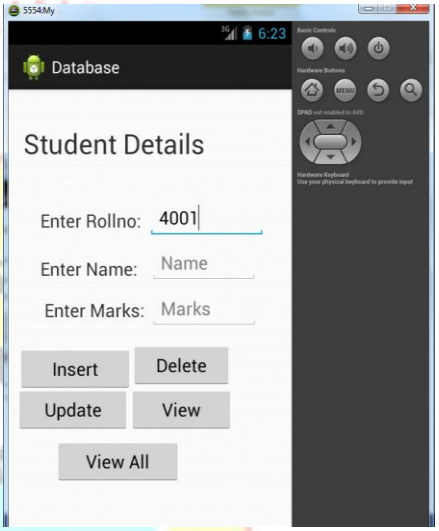
```
    public void clearText()
    {
        Rollno.setText("");
        Name.setText("");
        Marks.setText("");
        Rollno.requestFocus();
    }
}
```

Output

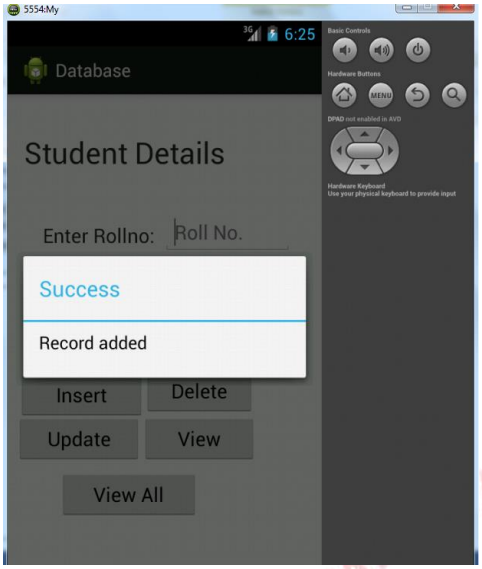
(i) Insert a Record



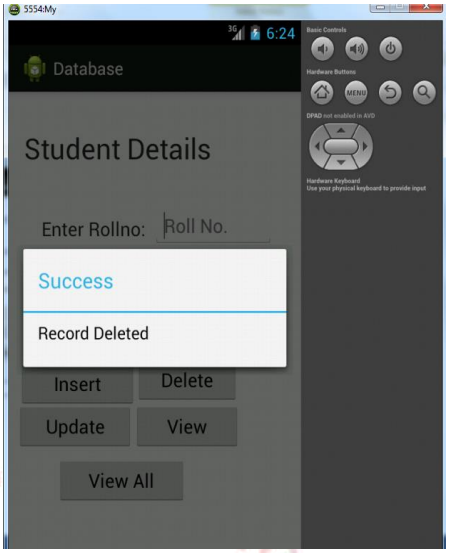
(iii) Delete a Record



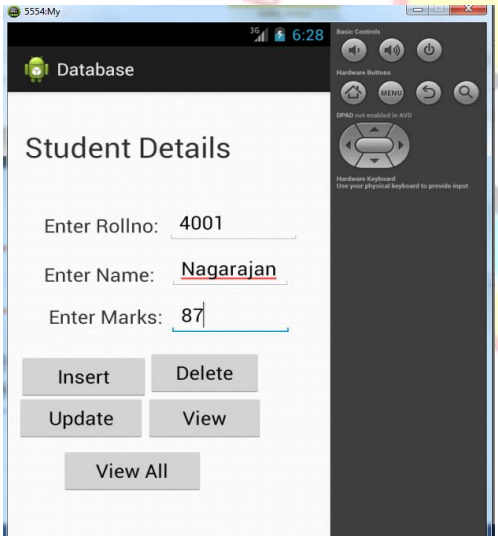
(ii) After Insertion



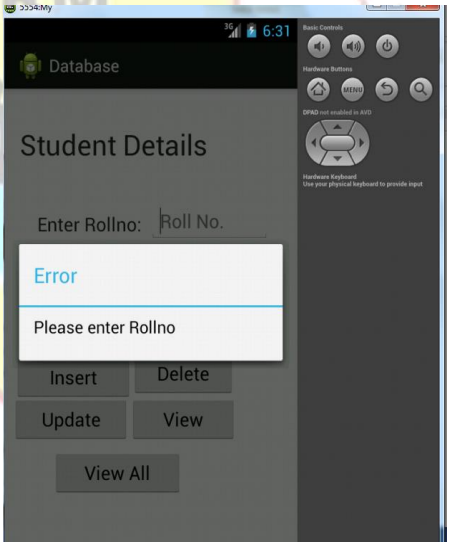
(iv) After Delete



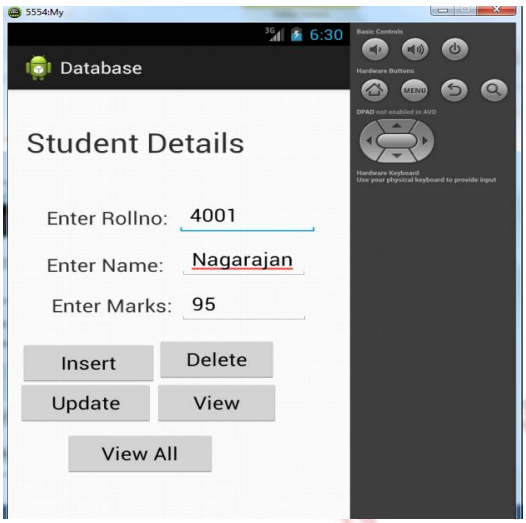
(v) Before Update



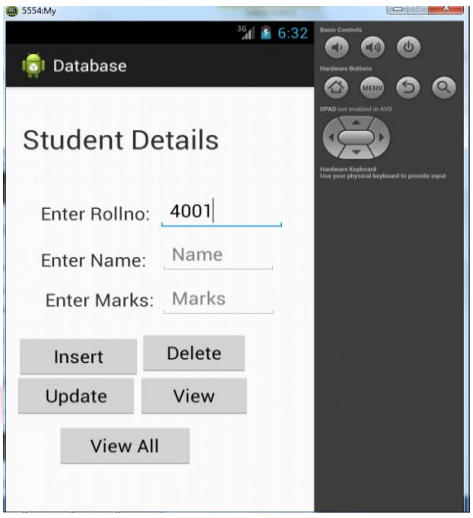
(vii) View



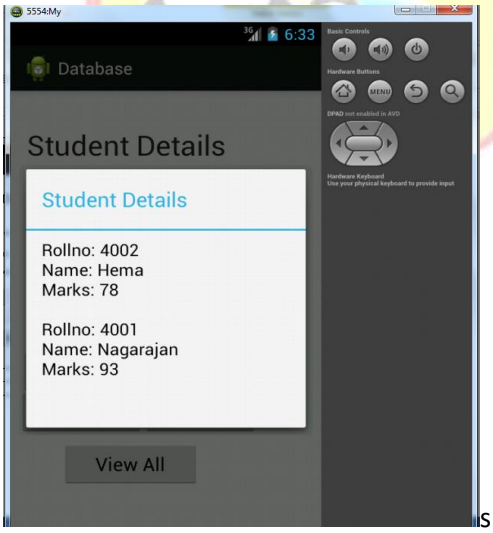
(vi) After update



(viii) View



(ix)View All



s

Result:

Thus the program to develop an application that makes use of database was executed successfully.

Ex. No. 7	Develop an application that makes use of internet for communication using Firebase to send SMS and E-Mail services..

Aim:

To develop a chat application using Firebase.

Procedure:

Step 1: Create three directories

"https://<your-firebase>/currentUsers/"
"https://<your-firebase>/Users/"
"https://<your-firebase>/Messages/"

Step 2: Main screen for choosing the entry option (Login or Register) and showing the number of currently logged in users

Step 3: Login for the actual user authentication

Step 4: Registration screen, in which we create new Users

Step 5 Chat screen (Which can show a Chat fragment or a ListOfUsers fragment)

Step 6: Run the code as Android Application

Step 7: Check the actions.

The Main Screen

MainActivity:

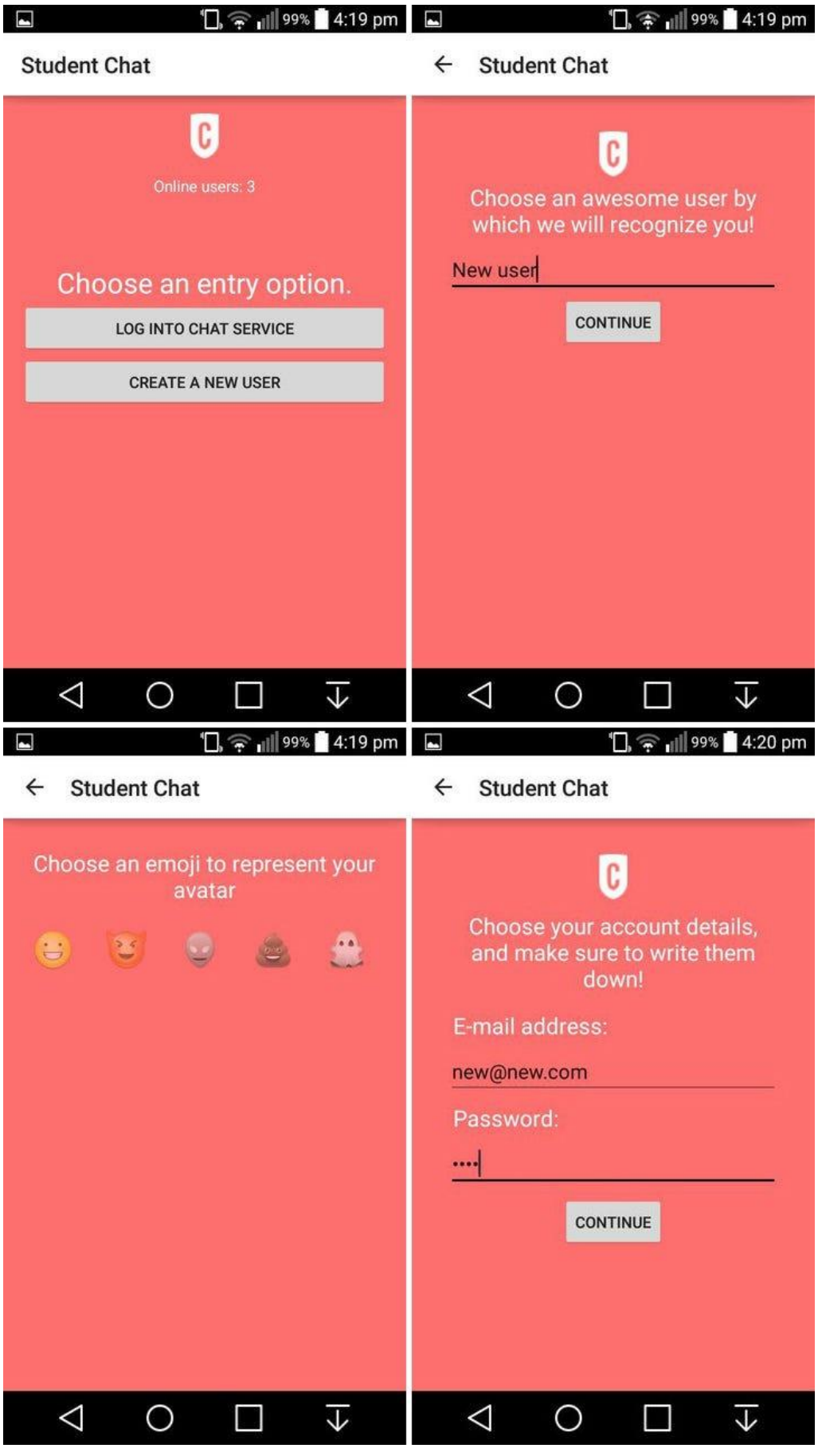
```
public class MainActivityPresenterImpl implements MainPresenter {
private final MainView mainView;
private final MainInteractor interactor;

public MainActivityPresenterImpl(MainView view) {
this.mainView = view;
interactor = new MainInteractor(this);
}


@Override
public void receiveRequest() {
```

```
interactor.receiveRequest();
}

@Override
public String getNumberOfUsers(long numberOfUsers) {
return "Online users: " + String.valueOf(numberOfUsers);
}

@Override
public void sendNumberOfChildren(long number) {
mainView.setNumberOfUsersTextView(getNumberOfUsers(number));
}
```

The MainInteractor:

```
public class MainInteractor implements MInteractor {
private final Firebase mainRef = new Firebase("https://<your-firebase>/currentUsers");
private final MainPresenter presenter;

public MainInteractor(MainPresenter pre) {
this.presenter = pre;
}

@Override
public void receiveRequest() {
mainRef.addListenerForSingleValueEvent(new ValueEventListener() {
@Override
public void onDataChange(DataSnapshot dataSnapshot) {
presenter.sendNumberOfChildren(dataSnapshot.getChildrenCount());
});
}
}
```

The Registration Screen Output:



public class FirebaseUserRegisterPresenterImpl implements FirebaseUserRegisterPresenter {
private final RegisterView registerView;

```
private final RegisterInteractor interactor;

public FirebaseUserRegisterPresenterImpl(RegisterView view) {
this.registerView = view;
this.interactor = new RegisterInteractor(this);
}


@Override
public void receiveRegisterRequest(String username, String email, String password, String emoji) {
interactor.receiveRegisterRequest(username, email, password, emoji);
registerView.spinProgressBar();
}

@Override
public void onFailure() {
registerView.onFailure();
registerView.stopProgressBar();
}

@Override
public void onSuccess() {
registerView.onSuccess();
registerView.stopProgressBar();
}
}
```

The Interactor:

```
public class RegisterInteractor implements RInteractor {
private Firebase userRef = new Firebase("https://<your-firebase>/Users/");
private final FirebaseUserRegisterPresenter presenter;

public RegisterInteractor(FirebaseUserRegisterPresenter pre) {
this.presenter = pre;
}

@Override
public void receiveRegisterRequest(final String username, String email, String password, final String
emoji) {
userRef.createUser(email, password, new Firebase.ValueResultHandler<Map<String, Object>>() {
@Override
public void onSuccess(Map<String, Object> stringObjectMap) {
String uid = stringObjectMap.get("uid").toString();
userRef = new Firebase("https://<your-firebase>/Users/" + uid);
userRef.setValue(createUser(username, emoji));
presenter.onSuccess();
}

@Override
public void onError(FirebaseError firebaseError) {
presenter.onFailure();
```

```
}
});
}

@Override
public Map<String, Object> createUser(String username, String emoji) {
Map<String, Object> user = new HashMap<>();
user.put("username", username);
user.put("emoji", emoji);
return user;
}
}
```

The Login Presenter:

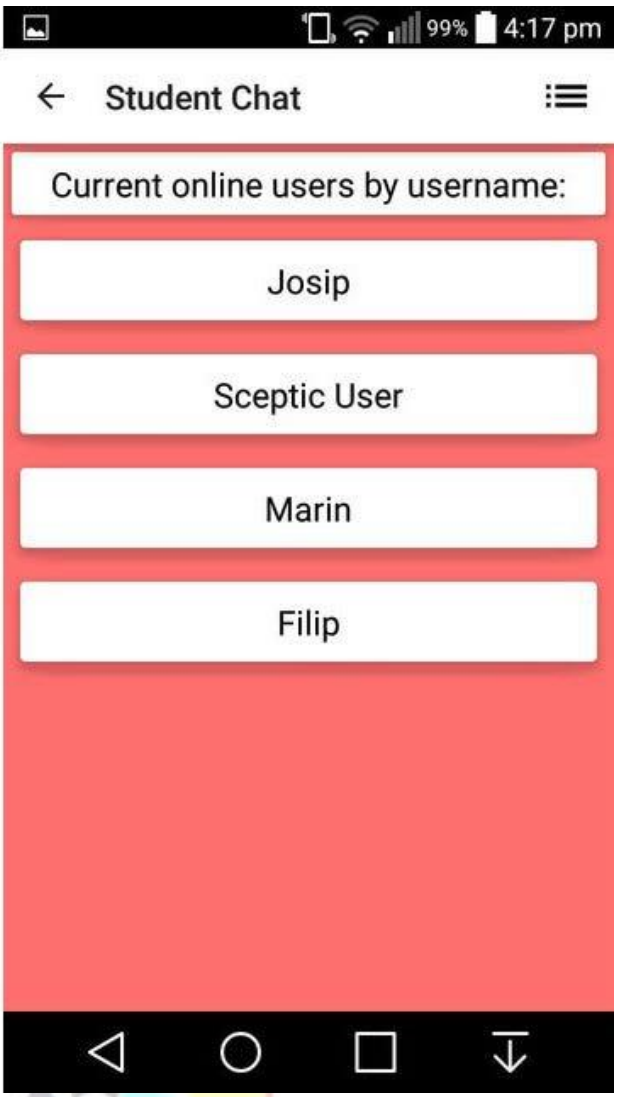
```
public class FirebaseLoginPresenterImpl implements FirebaseLoginPresenter {
private final LoginView loginView;
private final LoginInteractor interactor;


public FirebaseLoginPresenterImpl(LoginView view) {
this.loginView = view;
interactor = new LoginInteractor(this);
}

@Override
public void receiveUserLogin(String email, String password) {
loginView.spinProgressBar();
interactor.attemptToLogIn(email, password);

}

@Override
public void onFailure() {
loginView.stopProgressBar();
loginView.onFailure();
}

@Override
public void onSuccess(String user, String uid) {
loginView.stopProgressBar();
loginView.logTheUserIn(user, uid);
}
}
```

The Interactor:

```
public class LoginInteractor implements LInteractor {
private Firebase userRef = new Firebase("https://<your-firebase>/Users/");
private final FirebaseLoginPresenter presenter;

public LoginInteractor(FirebaseLoginPresenter pre) {
this.presenter = pre;
}

@Override
public void attemptToLogIn(String email, String password) {
userRef.authWithPassword(email, password, new Firebase.AuthResultHandler() {
@Override
public void onAuthenticated(final AuthData authData) {
userRef = new Firebase("https://<your-firebase>/Users/" + authData.getUid()); //retrieve the user data
userRef.addListenerForSingleValueEvent(new ValueEventListener() {
@Override
public void onDataChange(DataSnapshot dataSnapshot) {
User user = dataSnapshot.getValue(User.class);
Firebase loggedUser = new Firebase("https://<your-firebase>/currentUsers/" + authData.getUid());
//helps us log the user out later on
loggedUser.setValue(createUser(user.getUsername(), user.getEmoji()));
presenter.onSuccess(user.getUsername(), authData.getUid(), user.getEmoji());
}

@Override
public void onCancelled(FirebaseError firebaseError) {
}
});
}

@Override
public void onAuthenticationError(FirebaseError firebaseError) {
presenter.onFailure();
}
});
}

@Override
public Map<String, Object> createUser(String user, String emoji) {
Map<String, Object> userToCreate = new HashMap<>();
userToCreate.put("username", user);
userToCreate.put("emoji", emoji);
return userToCreate;
}
}
```

The Chat Screen Output:


99% 4:15 pm
Student Chat
Sceptic User
Hey, this really works!
You
Ofc it works, I wrote it!!
Josip
Don't be so sceptic all the time...
Marin
This design is awesome!
Enter a message.


99% 4:17 pm
Student Chat
Current online users by username:
Josip
Sceptic User
Marin
Filip

Ex. No. 8	Implement an android application that writes data into the SD card and makes use of NotificationManager.

Aim:

To implement an application that writes data to the SD card.

Procedure:

Step 1: File→ New → Android Project Application

Step 2: Specify Application Name

Step 3: Create the layout using the icons present in the widgets.

Step 4: Use Buttons and Text view to design the layout.

Step 5: Write the code to save data to SD card.

Step6 : Perform the actions.

Program:

```
package com.example.file;


import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.widget.*;
import android.view.View;
import android.view.View.OnClickListener;
import java.io.*;


import android.content.Context;


public class MainActivity extends Activity {


    Button b1,b2;
    EditText e;
```

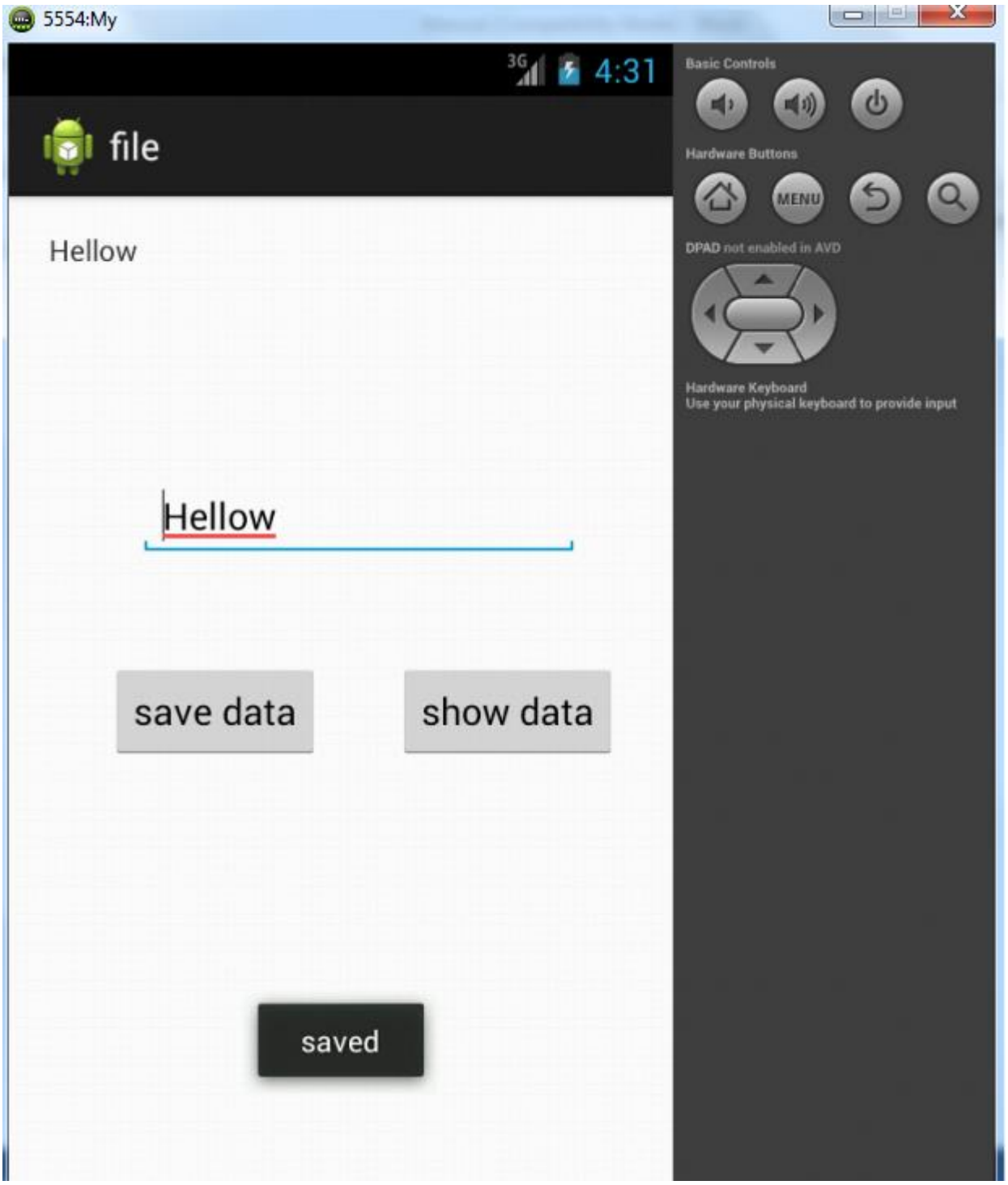
```
TextView tv;
String s;
protected FileOutputStream fout;
protected FileInputStream fin;

@Override
protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
tv=(TextView) findViewById(R.id.textView1);
b1=(Button)findViewById(R.id.button1);
b2=(Button)findViewById(R.id.button2);
e=(EditText)findViewById(R.id.editText1);
b1.setOnClickListener(new OnClickListener(){
        public void onClick(View arg0){
                s=e.getText().toString();
                try{
                        fout=openFileOutput("file 1",Context.MODE_APPEND);
                        fout.write(s.getBytes());
                        fout.close();
                        Toast.makeText(getBaseContext(), "saved",          Toast.LENGTH_LONG).show();

                }
                catch(Exception e){
                        e.printStackTrace();
                }
        }
});
b2.setOnClickListener(new OnClickListener(){
```

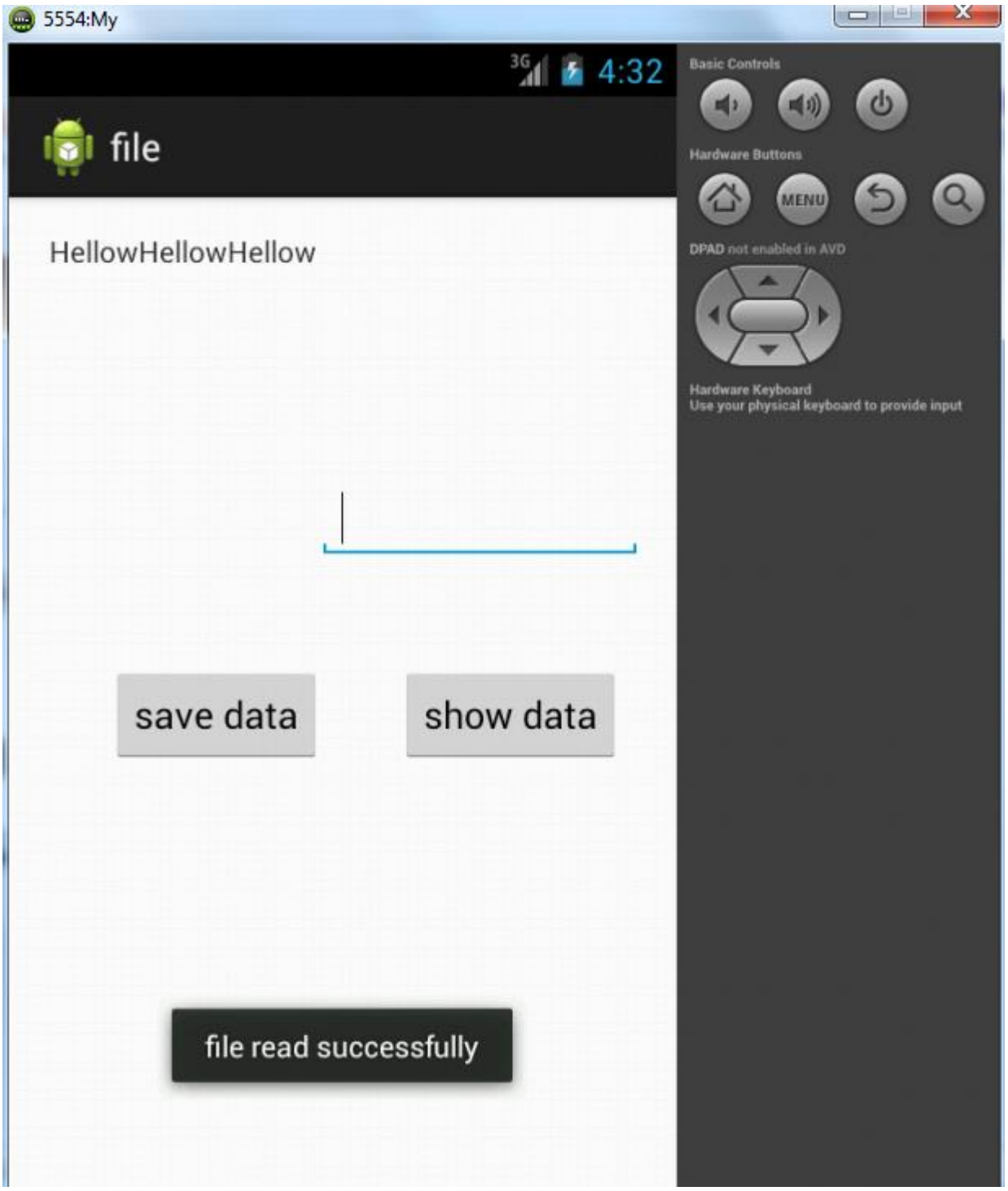
```
                public void onClick(View v){
                        s=e.getText().toString();
                        try{
                                fin=openFileInput("file 1");
                                int c;
                                String temp=" ";
                                while((c=fin.read())!=-1){
                                        temp=temp+Character.toString((char)c);
                                }
                                tv.setText(temp);
                                Toast.makeText(getBaseContext(), "file read successfully",
Toast.LENGTH_LONG).show();
                        }
                        catch(Exception e){
                                e.printStackTrace();
                        }
                }
        });
                        }

        @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;
        }

}
```

XML File:

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android: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:text="vec" />

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignLeft="@+id/textView1"
        android:layout_centerVertical="true"
        android:layout_marginLeft="32dp"
        android:text="save data" />

    <Button
        android:id="@+id/button2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_centerVertical="true"
        android:layout_marginLeft="36dp"
        android:layout_toRightOf="@+id/button1"
        android:text="show data" />

    <EditText
        android:id="@+id/editText1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_above="@+id/button1"
        android:layout_marginBottom="50dp"
        android:layout_toRightOf="@+id/textView1"
        android:ems="10" >

        <requestFocus />
    </EditText>

</RelativeLayout>
```

Output:


5554:My
4:31
file
Hellow
Hellow
save data
show data
saved
Basic Controls
Hardware Buttons
MENU
DPAD not enabled in AVD
Hardware Keyboard
Use your physical keyboard to provide input



Result:

Thus the application that writes data to the SD card is executed successfully

Ex. No. 9	Develop a native application that uses Location based services such as GPS tracking, Geo fencing,and activity recognition using Google play services

Aim:

To develop a native application *that uses GPS location Information.*

Procedure:

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.

AndroidGPSTrackingActivity.java

```
package com.example.gpstracking;

import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.Toast;

public class AndroidGPSTrackingActivity extends Activity {

	Button btnShowLocation;

	// GPSTracker class
	GPSTracker gps;

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

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

        // show location button click event
        btnShowLocation.setOnClickListener(new View.OnClickListener() {

                    @Override
                    public void onClick(View arg0) {
                          // create class object
                      gps = new GPSTracker(AndroidGPSTrackingActivity.this);

                          // check if GPS enabled
                      if(gps.canGetLocation()){
```

```
                        double latitude = gps.getLatitude();
                        double longitude = gps.getLongitude();

                        // \n is for new line
                        Toast.makeText(getApplicationContext(), "Your Location is - \nLat:
" + latitude + "\nLong: " + longitude, Toast.LENGTH_LONG).show();
                    }else{
                        // can't get location
                        // GPS or Network is not enabled
                        // Ask user to enable GPS/network in settings
                        gps.showSettingsAlert();
                    }

                }
            });
    }

}
```

GPSTracker.java

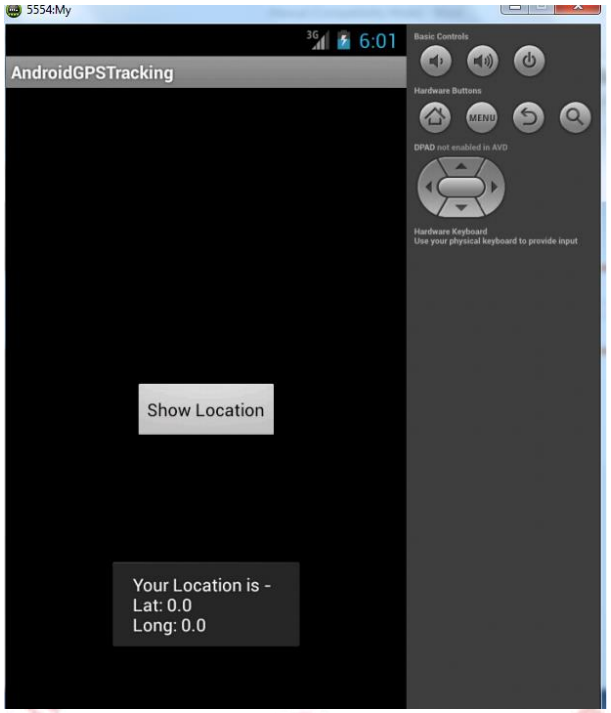
```
package com.example.gpstracking;

import android.app.AlertDialog;
import android.app.Service;
import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.os.IBinder;
import android.provider.Settings;
import android.util.Log;

public class GPSTracker extends Service implements LocationListener {

    private final Context mContext;

    // flag for GPS status
    boolean isGPSEnabled = false;

    // flag for network status
    boolean isNetworkEnabled = false;

    // flag for GPS status
    boolean canGetLocation = false;

    Location location; // location
    double latitude; // latitude
    double longitude; // longitude

    // The minimum distance to change Updates in meters
    private static final long MIN_DISTANCE_CHANGE_FOR_UPDATES = 10; // 10 meters

    // The minimum time between updates in milliseconds
    private static final long MIN_TIME_BW_UPDATES = 1000 * 60 * 1; // 1 minute

    // Declaring a Location Manager
    protected LocationManager locationManager;
```

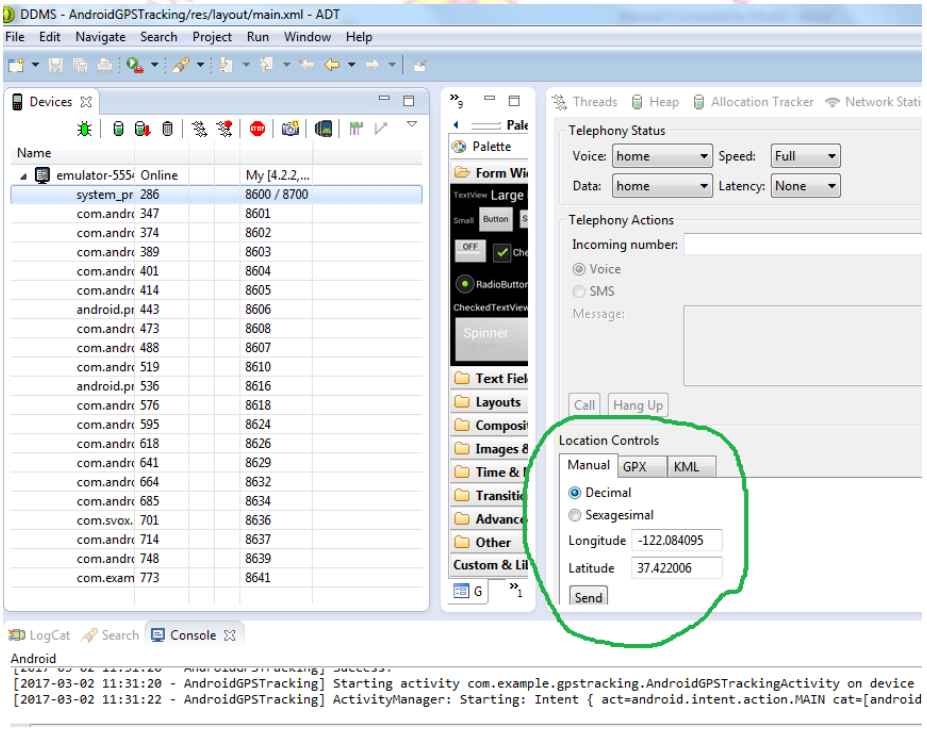
```
public GPSTracker(Context context) {
	this.mContext = context;
	getLocation();
}

public Location getLocation() {
	try {
		locationManager = (LocationManager) mContext
				.getSystemService(LOCATION_SERVICE);

		// getting GPS status
		isGPSEnabled = locationManager
				.isProviderEnabled(LocationManager.GPS_PROVIDER);

		// getting network status
		isNetworkEnabled = locationManager
				.isProviderEnabled(LocationManager.NETWORK_PROVIDER);

		if (!isGPSEnabled && !isNetworkEnabled) {
			// no network provider is enabled
		} else {
			this.canGetLocation = true;
			if (isNetworkEnabled) {
				locationManager.requestLocationUpdates(
						LocationManager.NETWORK_PROVIDER,
						MIN_TIME_BW_UPDATES,
						MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
				Log.d("Network", "Network");
				if (locationManager != null) {
					location = locationManager

.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);
					if (location != null) {
						latitude = location.getLatitude();
						longitude = location.getLongitude();
					}
				}
			}
			// if GPS Enabled get lat/long using GPS Services
			if (isGPSEnabled) {
				if (location == null) {
					locationManager.requestLocationUpdates(
							LocationManager.GPS_PROVIDER,
							MIN_TIME_BW_UPDATES,
							MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
					Log.d("GPS Enabled", "GPS Enabled");
					if (locationManager != null) {
						location = locationManager

.getLastKnownLocation(LocationManager.GPS_PROVIDER);
						if (location != null) {
							latitude = location.getLatitude();
							longitude = location.getLongitude();
						}
					}
				}
			}
		}

	} catch (Exception e) {
		e.printStackTrace();
	}
```

```
        return location;
    }

    /**
     * Stop using GPS listener
     * Calling this function will stop using GPS in your app
     * */
    public void stopUsingGPS(){
        if(locationManager != null){
            locationManager.removeUpdates(GPSTracker.this);
        }
    }

    /**
     * Function to get latitude
     * */
    public double getLatitude(){
        if(location != null){
            latitude = location.getLatitude();
        }

        // return latitude
        return latitude;
    }

    /**
     * Function to get longitude
     * */
    public double getLongitude(){
        if(location != null){
            longitude = location.getLongitude();
        }

        // return longitude
        return longitude;
    }

    /**
     * Function to check GPS/wifi enabled
     * @return boolean
     * */
    public boolean canGetLocation() {
        return this.canGetLocation;
    }

    /**
     * Function to show settings alert dialog
     * On pressing Settings button will lauch Settings Options
     * */
    public void showSettingsAlert(){
        AlertDialog.Builder alertDialog = new AlertDialog.Builder(mContext);

      // Setting Dialog Title
      alertDialog.setTitle("GPS is settings");

      // Setting Dialog Message
      alertDialog.setMessage("GPS is not enabled. Do you want to go to settings menu?");

      // On pressing Settings button
      alertDialog.setPositiveButton("Settings", new DialogInterface.OnClickListener() {
          public void onClick(DialogInterface dialog,int which) {
            Intent intent = new Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
```

```
                mContext.startActivity(intent);
            }
        });

        // on pressing cancel button
        alertDialog.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {
            public void onClick(DialogInterface dialog, int which) {
            dialog.cancel();
            }
        });

        // Showing Alert Message
        alertDialog.show();
    }

    @Override
    public void onLocationChanged(Location location) {
    }

    @Override
    public void onProviderDisabled(String provider) {
    }

    @Override
    public void onProviderEnabled(String provider) {
    }

    @Override
    public void onStatusChanged(String provider, int status, Bundle extras) {
    }

    @Override
    public IBinder onBind(Intent arg0) {
           return null;
    }

}
```

Output:


5554:My
6:01
AndroidGPSTracking
Show Location
Your Location is -
Lat: 0.0
Long: 0.0


DDMS - AndroidGPSTracking/res/layout/main.xml - ADT
File Edit Navigate Search Project Run Window Help
Devices
Name
Telephony Status
Voice: home
Speed: Full
Data: home
Latency: None
Telephony Actions
Incoming number:
Voice
SMS
Message:
Call
Hang Up
Location Controls
Manual
GPX
KML
Decimal
Sexagesimal
Longitude -122.084095
Latitude 37.422006
Send
LogCat
Search
Console
Android
[2017-03-02 11:31:20 - AndroidGPSTracking] Starting activity com.example.gpstracking.AndroidGPSTrackingActivity on device
[2017-03-02 11:31:22 - AndroidGPSTracking] ActivityManager: Starting: Intent { act=android.intent.action.MAIN cat=[android

Result:

Thus the program to track the location using GPS is executed successfully.

Ex. No. 10	Implement simple gaming application using open-source tools like flutter or Unity.

Aim:

To develop a Flutter, Drag and Drop game

.

Procedure:

Step 1: File→ New → Android Project Application

Step 2: Specify Application Name

Step 3: Design the layout and Import images.

Step 4: Write code for Main Activity.java.

Step 5: Run the application using AVD Emulator.

Mainactivity.Java

```
package com.example.dnd;


import android.os.Bundle;
import io.flutter.app.FlutterActivity;
import io.flutter.plugins.GeneratedPluginRegistrant;


public class MainActivity extends FlutterActivity {
  @Override
  protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    GeneratedPluginRegistrant.registerWith(this);
  }
}
```

```
import 'package:flutter/material.dart';
import 'dart:math';
import 'package:audioplayers/audio_cache.dart';

void main() => runApp(MyApp());

class MyApp extends StatelessWidget {
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      theme: ThemeData(
        fontFamily: 'PressStart',
      ),
      home: ColorGame(),
    );
  }
}

class ColorGame extends StatefulWidget {
  ColorGame({Key key}) : super(key: key);

  createState() => ColorGameState();
}

class ColorGameState extends State<ColorGame> {
```

```
/// Map to keep track of score
final Map<String, bool> score = {};

/// Choices for game
final Map choices = {
  '🍏': Colors.green,
  '🍋': Colors.yellow,
  '🍅': Colors.red,
  '🍇': Colors.purple,
  '🥥': Colors.brown,
  '🥕': Colors.orange
};

// Random seed to shuffle order of items.
int seed = 0;

@override
Widget build(BuildContext context) {
  return Scaffold(
    appBar: AppBar(
        title: Text('Score ${score.length} / 6'),
        backgroundColor: Colors.pink),
    floatingActionButton: FloatingActionButton(
      child: Icon(Icons.refresh),
      onPressed: () {
        setState(() {
```

```
        score.clear();
        seed++;
      });
    },
  ),
  body: Row(
    mainAxisAlignment: MainAxisAlignment.spaceEvenly,
    children: [
      Column(
        mainAxisAlignment: MainAxisAlignment.spaceAround,
        crossAxisAlignment: CrossAxisAlignment.end,
        children: choices.keys.map((emoji) {
          return Draggable<String>(
            data: emoji,
            child: Emoji(emoji: score[emoji] == true ? '✅' : emoji),
            feedback: Emoji(emoji: emoji),
            childWhenDragging: Emoji(emoji: '🌱'),
          );
        }).toList()),
      Column(
        mainAxisAlignment: MainAxisAlignment.spaceAround,
        crossAxisAlignment: CrossAxisAlignment.start,
        children:
          choices.keys.map((emoji) => _buildDragTarget(emoji)).toList()
            ..shuffle(Random(seed)),
      )
```

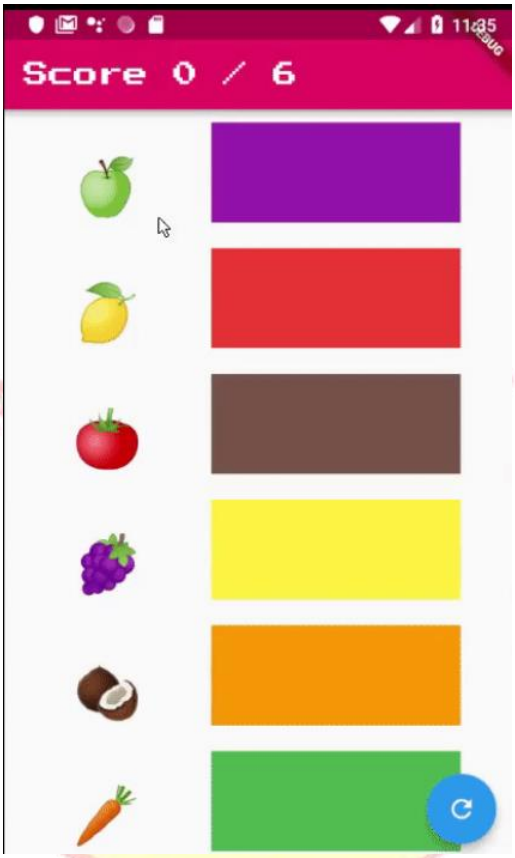
```
      ],
    ),
  );
}

Widget _buildDragTarget(emoji) {
  return DragTarget<String>(
    builder: (BuildContext context, List<String> incoming, List rejected) {
      if (score[emoji] == true) {
        return Container(
          color: Colors.white,
          child: Text('Correct!'),
          alignment: Alignment.center,
          height: 80,
          width: 200,
        );
      } else {
        return Container(color: choices[emoji], height: 80, width: 200);
      }
    },
    onWillAccept: (data) => data == emoji,
    onAccept: (data) {
      setState(() {
        score[emoji] = true;
        plyr.play('success.mp3');
      });
```

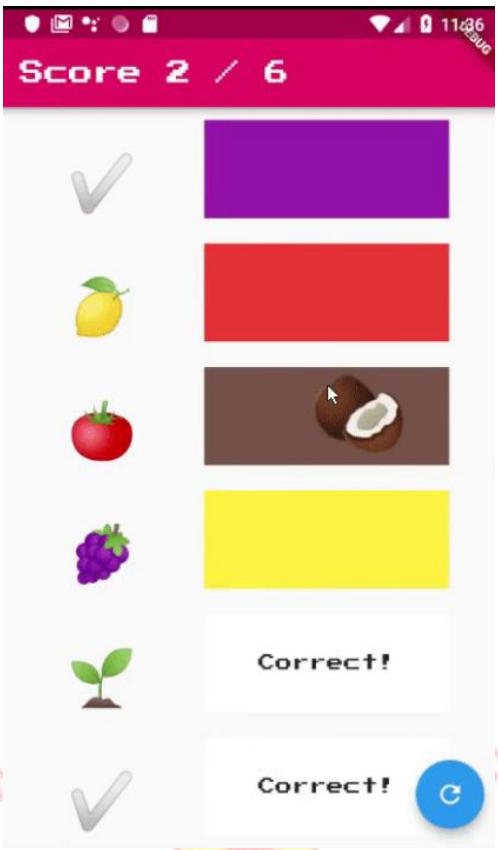
```
      },
      onLeave: (data) {},
    );
  }
}

class Emoji extends StatelessWidget {
  Emoji({Key key, this.emoji}) : super(key: key);

  final String emoji;

  @override
  Widget build(BuildContext context) {
    return Material(
      color: Colors.transparent,
      child: Container(
        alignment: Alignment.center,
        height: 50,
        padding: EdgeInsets.all(10),
        child: Text(
          emoji,
          style: TextStyle(color: Colors.black, fontSize: 50),
        ),
      ),
    );
  }
```

}

AudioCache plyr = AudioCache();

Output:





Content Beyond Syllabus

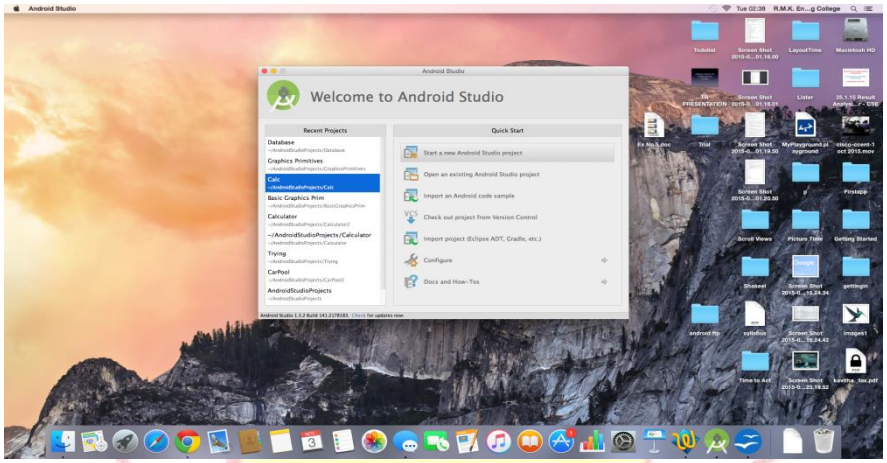
Ex No: 11 Write a mobile application that creates alarm clock

Aim:

To develop an application that creates alarm clock

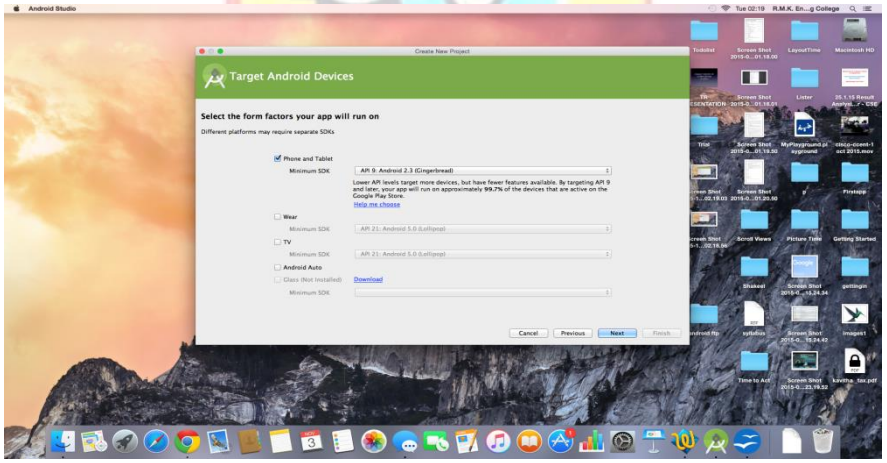
Procedure:

Open the Android Studio and Click Start New Android Project

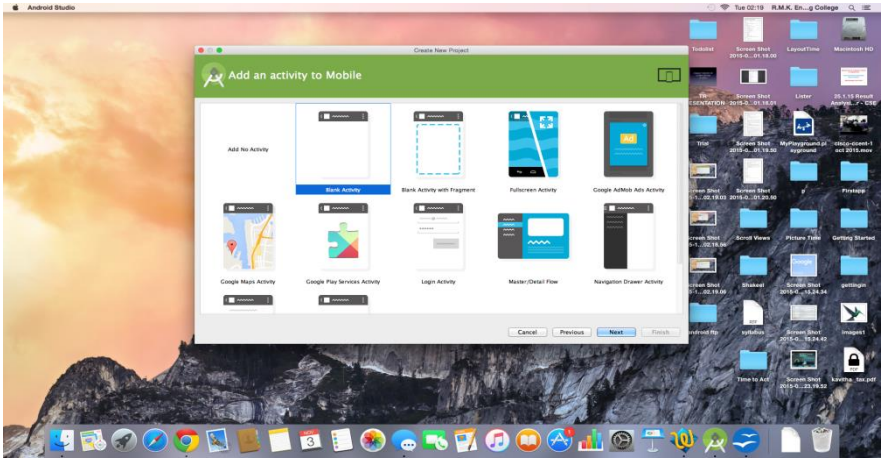


Configure the Project

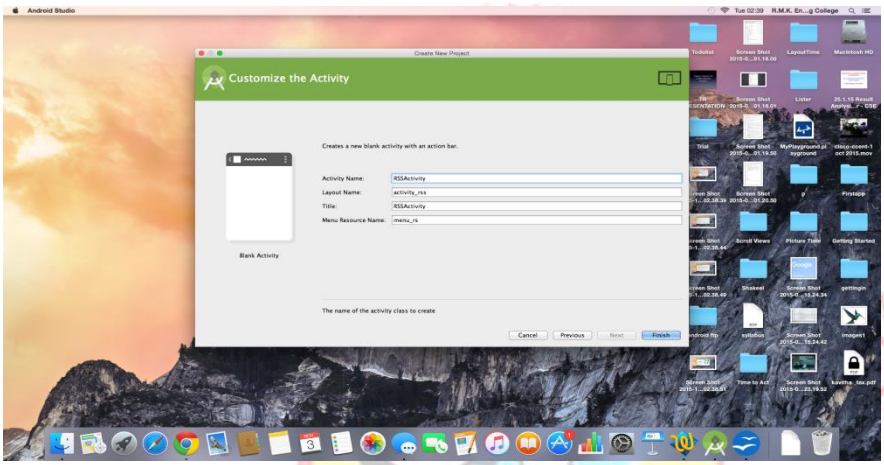
Selecting the form factors for running the application (minimal requirement)



Select Blank Activity and give the activity names.



Place the editor and button on the relative layout and the activity_main.xml code is shown below



Code:

Add required permissions to AndroidManifest file.

The application will make use of Internet, so this should be specified in the AndroidManifest file, otherwise an exception will be thrown. Just after the <application> tag, (as a child of <manifest> tag), add the following line:

```
<uses-permission android:name="android.permission.WAKE_LOCK" />
```

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

   <TextView android:text="@string/hello_world" android:layout_width="wrap_content"
      android:layout_height="wrap_content" />
   <TimePicker android:layout_width="wrap_content" android:layout_height="wrap_content"
android:id="@+id/alarmTimePicker" android:layout_alignParentTop="true"
```

```
android:layout_centerHorizontal="true" />
   <ToggleButton android:layout_width="wrap_content" android:layout_height="wrap_content"
android:text="Alarm On/Off" android:id="@+id/alarmToggle"
      android:onClick="onToggleClicked"
      android:layout_below="@+id/alarmTimePicker"
      android:layout_centerHorizontal="true"
      android:layout_marginTop="40dp" />
   <TextView android:layout_width="wrap_content" android:layout_height="wrap_content"
android:textAppearance="?android:attr/textAppearanceLarge" android:text=""
      android:id="@+id/alarmText"
      android:layout_alignParentBottom="true" android:layout_centerHorizontal="true"
android:layout_marginTop="20dp" android:layout_below="@+id/alarmToggle" />
</RelativeLayout>
```

AlarmActivity.java

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

```
package com.javapapers.androidalarmclock;
import android.app.Activity; import android.app.AlarmManager; import android.app.PendingIntent;
import android.content.Intent; import android.os.Bundle;
import android.util.Log;
import android.view.View;
import android.widget.TextView;
import android.widget.TimePicker;
import android.widget.ToggleButton;

import java.util.Calendar;
public class AlarmActivity extends Activity { AlarmManager alarmManager;
private PendingIntent pendingIntent;
private TimePicker alarmTimePicker;
private static AlarmActivity inst;
private TextView alarmTextView;
public static AlarmActivity instance() {
return inst;
}
@Override
public void onStart() {
super.onStart();
inst = this;
}
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_my);
alarmTimePicker = (TimePicker) findViewById(R.id.alarmTimePicker);
alarmTextView = (TextView) findViewById(R.id.alarmText); ToggleButton alarmToggle =
(ToggleButton)
findViewById(R.id.alarmToggle);
alarmManager = (AlarmManager) getSystemService(ALARM_SERVICE);
}
public void onToggleClicked(View view) {
if (((ToggleButton) view).isChecked()) { Log.d("MyActivity", "Alarm On");
Calendar calendar = Calendar.getInstance();
calendar.set(Calendar.HOUR_OF_DAY, alarmTimePicker.getCurrentHour());
calendar.set(Calendar.MINUTE, alarmTimePicker.getCurrentMinute());
```

```
Intent myIntent = new Intent(AlarmActivity.this, AlarmReceiver.class);
pendingIntent = PendingIntent.getBroadcast(AlarmActivity.this, 0,
myIntent, 0);
alarmManager.set(AlarmManager.RTC, calendar.getTimeInMillis(),
pendingIntent);
} else {
alarmManager.cancel(pendingIntent);
setAlarmText(""); Log.d("MyActivity", "Alarm Off");
}
}
public void setAlarmText(String alarmText) {
alarmTextView.setText(alarmText);
}
}
```

AlarmReceiver.java

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

```
package com.javapapers.androidalarmclock;
import android.app.Activity;
import android.content.ComponentName;
import android.content.Context;
import android.content.Intent;
import android.media.Ringtone;
import android.media.RingtoneManager;
import android.net.Uri;
import android.support.v4.content.WakefulBroadcastReceiver;
public class AlarmReceiver extends WakefulBroadcastReceiver {
@Override
public void onReceive(final Context context, Intent intent) {
//this will update the UI with message
AlarmActivity inst = AlarmActivity.instance();
inst.setAlarmText("Alarm! Wake up! Wake up!");
//this will sound the alarm tone
//this will sound the alarm once, if you wish to
//raise alarm in loop continuously then use MediaPlayer and setLooping(true)
Uri alarmUri =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_ALARM);
if (alarmUri == null) {
alarmUri =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
}
Ringtone ringtone = RingtoneManager.getRingtone(context, alarmUri);
ringtone.play();
//this will send a notification message
ComponentName comp = new ComponentName(context.getPackageName(),
```

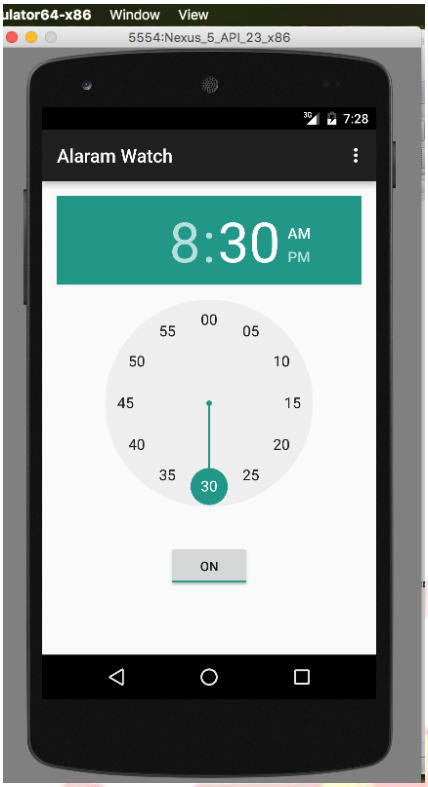
```
AlarmService.class.getName());
startWakefulService(context, (intent.setComponent(comp)));
setResultCode(Activity.RESULT_OK);
}
}
```

AlarmService.java

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

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

Output:



Result:

Mobile application created with alarm clock.

Vivia Question

1. Install and configure Java Development Kit (JDK), Android Studio, and Android SDK

1. Q: What is JDK?
   A: The Java Development Kit is a software development environment used for building Java applications.
2. Q: Why is Android Studio used?
   A: It is the official IDE for Android development, providing tools for coding, debugging, and testing.
3. Q: What is an Android SDK?
   A: It is a collection of libraries and tools required for Android app development.
4. Q: What is Gradle?
   A: A build automation tool used in Android Studio for compiling and managing dependencies.
5. Q: How do you set up an emulator in Android Studio?
   A: Use the AVD (Android Virtual Device) Manager to create and configure virtual devices.

2. Develop an application that uses GUI components, fonts, and colours

6. Q: What is a GUI component?
   A: Graphical User Interface components like buttons, text views, and sliders used to build app interfaces.
7. Q: How do you set custom fonts in Android?
   A: By placing the font file in the res/fonts folder and referencing it in XML or code.
8. Q: How can you change the background colour of an activity?
   A: By using the android:background attribute in XML or setting it programmatically.
9. Q: What is a TextView in Android?
   A: A UI element used to display text to the user.
10. Q: How do you handle button clicks in Android?
    A: By setting an OnClickListener in Java or Kotlin.

3. Design an application with Layout Managers, Event listeners, and push notifications

11. Q: What are Layout Managers?
    A: They define the arrangement of UI components, such as LinearLayout, RelativeLayout, and ConstraintLayout.

12. Q: What is an Event Listener?
    A: A mechanism to handle user interactions like clicks or swipes.

13. Q: How do you implement push notifications in Android?
    A: By using Firebase Cloud Messaging (FCM).

14. Q: What is the role of onClick() in event handling?
    A: It is a callback method triggered when a button is clicked.

15. Q: What is the ConstraintLayout?
    A: A flexible layout manager that allows you to position UI components relative to each other or the parent.

4. Build a simple native calculator application

16. Q: What is the use of EditText in Android?
    A: It allows users to enter and edit text in the app.

17. Q: How do you perform arithmetic operations in Android?
    A: By capturing inputs, performing operations in Java/Kotlin code, and displaying the result in a TextView.

18. Q: What is the role of setText() in Android?
    A: It sets the text of a TextView or EditText dynamically.

19. Q: What is the difference between int and double in calculations?
    A: int is for whole numbers, while double supports decimal numbers.

20. Q: How do you clear the input fields in a calculator app?
    A: By setting the text of EditText fields to an empty string.

5. Create animations and graphical primitives in Android

21. Q: What is a Drawable in Android?
    A: A resource that can be drawn on the screen, such as shapes or images.

22. Q: How do you create an animation in Android?
    A: By using the Animator or Animation classes.

23. Q: What is the role of Canvas in graphics?
    A: It provides methods to draw shapes, text, and images.

24. Q: What are the types of animations in Android?
    A: Property animations, view animations, and drawable animations.

25. Q: What is a VectorDrawable?
    A: A type of drawable that uses XML to define vector graphics.

6. Develop an application with SQLite mobile database

26. Q: What is SQLite?
    A: A lightweight relational database built into Android for local data storage.

27. Q: How do you create a database in Android?
    A: By extending the SQLiteOpenHelper class.

28. Q: What is a Content Provider?
    A: A component that facilitates data sharing between apps.

29. Q: What are CRUD operations?
    A: Create, Read, Update, and Delete operations in a database.

30. Q: What is the use of Cursor in SQLite?
    A: It retrieves query results from the database.

7. Develop an application using Firebase for SMS and email services

31. Q: What is Firebase?
    A: A platform by Google for building and scaling apps with tools like authentication, database, and messaging.

32. Q: How do you send an email using Firebase?
    A: By integrating a third-party email service like SendGrid.

33. Q: What is Firebase Authentication?
    A: A service for user authentication using email, phone, or social providers.

34. Q: How do you send SMS using Firebase?
    A: By using Firebase's phone authentication or integrating with an SMS gateway.

35. Q: What is the Firebase Realtime Database?
    A: A cloud-hosted NoSQL database for syncing data in real time.

8. Write data to SD card and use Notification Manager

36. Q: How do you write data to an SD card in Android?
    A: By using the File and FileOutputStream classes.

37. Q: What is a Notification Manager?
    A: A service to manage and display app notifications.

38. Q: What is the difference between internal and external storage?
    A: Internal storage is private to the app; external storage can be accessed by other apps.

39. Q: What is the importance of runtime permissions for SD card access?
    A: Starting from Android 6.0, explicit permissions are required to access external storage.

40. Q: How do you create a notification in Android?
    A: By using the NotificationCompat.Builder class.

9. Location-based services (GPS, Geo-fencing)

41. Q: What is GPS?
    A: Global Positioning System for determining a device's location.

42. Q: What is Geo-fencing?
    A: Setting up virtual boundaries to trigger actions when a user enters or exits them.

43. Q: What is the Fused Location Provider API?
    A: A high-level API for location tracking in Android.

44. Q: How do you track a device’s location in Android?
    A: By using the LocationManager or Google Play Services.

45.Q: What is activity recognition?
A: A feature to detect physical activities like walking or driving.

10. Implement a simple gaming application using Flutter or Unity

46.Q: What is Flutter?
A: A cross-platform UI toolkit for building apps using Dart.

47.Q: What is Unity?
A: A game development platform for creating 2D and 3D games.

48.Q: How do you handle user input in Flutter?
A: By using widgets like GestureDetector or event listeners.

49.Q: What is the role of a Game Engine in Unity?
A: It provides tools for physics, rendering, and scripting.

50.Q: What is a widget tree in Flutter?
A: The hierarchy of widgets that make up a Flutter app.