| **EX.NO:1** | **DEVELOP AN APPLICATION THAT USES GUI COMPONENTS, FONT AND COLOURS** |
| --- | --- |
| **DATE:** |

**AIM:**

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

**PROCEDURE:**

1. Open android studio and select new android project.
2. Give project name and select next
3. Choose the android version(Android 2.2) and select next
4. Enter the package name. Package name must be two word separated by comma and click finish
5. Go to package explorer in the left hand side.
6. Go to res folder and select layout. Double click the main.xml file
7. Now you can see the Graphics layout window.
8. Click the main.xml file and type the xml code
9. Again click the graphics layout tab and screen layout is look like below.
10. Go to project explorer and select src folder. Now select mainactivity.java file and type the java code.
11. Now go to main.xml and right click .select run as option and select run configuration.
12. Android output is present in the android emulator.

**SOURCE CODE:**

**Layout.xml**

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

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

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:orientation="vertical" >

<TextView

android:id="@+id/textView1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="20sp"

android:gravity="center"

android:text="HELLO WORLD"

android:textSize="20sp"

android:textStyle="bold" />

<Button

android:id="@+id/button1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:gravity="center"

android:text="Change font size"

android:textSize="20sp" />

<Button

android:id="@+id/button2"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:gravity="center"

android:text="Change color"

android:textSize="20sp" />

<Button

android:id="@+id/button3"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:gravity="center"

android:text="Change font"

android:textSize="20sp" />

</LinearLayout

**Activity.java**

//import android.R;

importandroid.app.Activity;

importandroid.graphics.Color;

importandroid.graphics.Typeface;

importandroid.os.Bundle;

importandroid.view.View;

importandroid.widget.Button;

importandroid.widget.TextView;

public class AndroidActivity extends Activity {

float font =24;

inti=1;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

finalTextView t1=(TextView) findViewById(R.id.textView1);

Button b1 = (Button) findViewById(R.id.button1);

b1.setOnClickListener(new View.OnClickListener() {

public void onClick(View view) {

t1.setTextSize(font);

font=font+4;

if(font==40)

font=20;

}

});

Button b2 = (Button) findViewById(R.id.button2);

b2.setOnClickListener(new View.OnClickListener() {

public void onClick(View view) {

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;

case 4:

t1.setTextColor(Color.parseColor("#800000"));

break;

}

i++;

if(i==5)

i=1;

}

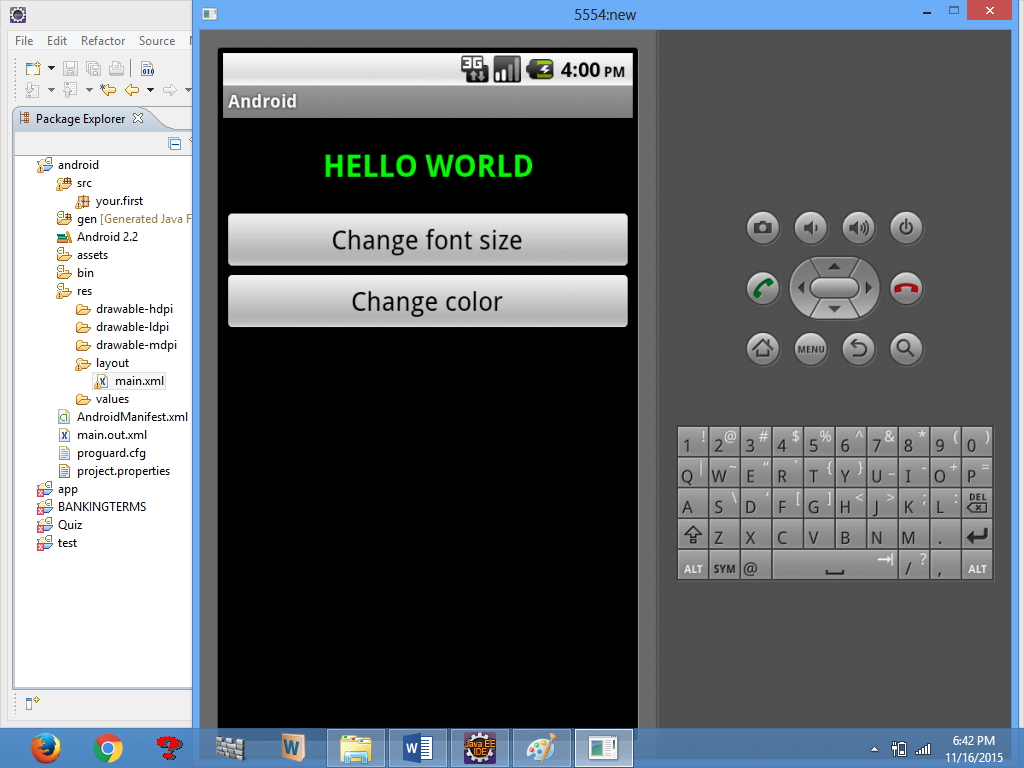
});

}

}

\

**OUTPUT:**



**RESULT:**

Thus the Project to develop An Application That Uses GUI Components, Font and colours was developed using Android Suite Software Successfully.

**OUTCOME:**

Understand the Concept of GUI Component font and colors for a mobile application.

| **EX.NO: 2** | **DEVELOP AN APPLICATION THAT USES LAYOUT MANAGERS AND EVENT LISTENERS**. |
| --- | --- |
| **DATE:** |

**AIM:**

To develop an application that uses layout managers and event listeners.

**PROCEDURE:**

1. Open android studio and select new android project.
2. Give project name and select next.
3. Choose the android version. Choose the lowest android version(Android 2.2)
4. Enter the package name. Package name must be two word separated by comma and click finish.
5. Go to package explorer in the left hand side. Select our project.
6. Go to res folder and select layout. Double click the main.xml file.
7. Now select mainactivity.java file.
8. Now go to main.xml and right click. Select run as option and select run configuration.
9. Android output is present in the android emulator.

**SOURCE CODE:**

**Layout.xml**

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

android:id="@+id/relativeLayout1"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent" >

<LinearLayout

android:id="@+id/linearLayout1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:layout\_alignParentRight="true"

android:layout\_alignParentTop="true" >

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:text="ADDITION"

android:textSize="20dp" >

</TextView>

</LinearLayout>

<LinearLayout

android:id="@+id/linearLayout2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:layout\_alignParentRight="true"

android:layout\_below="@+id/linearLayout1" >

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="ENTER NO 1" >

</TextView>

<EditText

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="0.20"

android:id="@+id/edittext1"

android:inputType="number">

</EditText>

</LinearLayout>

<LinearLayout

android:id="@+id/linearLayout3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:layout\_alignParentRight="true"

android:layout\_below="@+id/linearLayout2" >

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="ENTER NO 2" >

</TextView>

<EditText

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_weight="0.20"

android:id="@+id/edittext2"

android:inputType="number">

</EditText>

</LinearLayout>

<LinearLayout

android:id="@+id/linearLayout4"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:layout\_alignParentRight="true"

android:layout\_below="@+id/linearLayout3" >

<Button

android:layout\_width="wrap\_content"

android:id="@+id/button1"

android:layout\_height="wrap\_content"

android:text="Addition"

android:layout\_weight="0.50" />

<Button

android:layout\_width="wrap\_content"

android:id="@+id/button3"

android:layout\_height="wrap\_content"

android:text="subtraction"

android:layout\_weight="0.50" />

<Button

android:layout\_width="wrap\_content"

android:id="@+id/button2"

android:layout\_height="wrap\_content"

android:text="CLEAR"

android:layout\_weight="0.50" />

</LinearLayout>

<View

android:layout\_height="2px"

android:layout\_width="fill\_parent"

android:layout\_below="@+id/linearLayout4"

android:background="#DDFFDD"/>

</RelativeLayout>

package layout.ne;

**Activity.java**

importandroid.app.Activity;

importandroid.os.Bundle;

importandroid.view.View;

importandroid.view.View.OnClickListener;

importandroid.widget.Button;

importandroid.widget.EditText;

importandroid.widget.Toast;

public class LAYOUTActivity extends Activity {

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

EditText txtData1,txtData2;

float num1,num2,result1,result2;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

Button add = (Button) findViewById(R.id.button1);

add.setOnClickListener(new OnClickListener() {

public void onClick(View v) {

try

{

txtData1 = (EditText) findViewById(R.id.edittext1);

txtData2 = (EditText) findViewById(R.id.edittext2);

num1 = Float.parseFloat(txtData1.getText().toString());

num2 = Float.parseFloat(txtData2.getText().toString());

result1=num1+num2;

Toast.makeText(getBaseContext(),"ANSWER:"+result1,Toast.LENGTH\_SHORT).show();

}

catch(Exception e)

{

Toast.makeText(getBaseContext(), e.getMessage(),

Toast.LENGTH\_SHORT).show();

}

}

});

Button sub = (Button) findViewById(R.id.button3);

sub.setOnClickListener(new OnClickListener() {

public void onClick(View v) {

try

{

txtData1 = (EditText) findViewById(R.id.edittext1);

txtData2 = (EditText) findViewById(R.id.edittext2);

num1 = Float.parseFloat(txtData1.getText().toString());

num2 = Float.parseFloat(txtData2.getText().toString());

result2=num1-num2;

Toast.makeText(getBaseContext(),"ANSWER:"+result2,Toast.LENGTH\_SHORT).show();

}

catch(Exception e)

{

Toast.makeText(getBaseContext(), e.getMessage(),

Toast.LENGTH\_SHORT).show();

}

}

});

Button clear = (Button) findViewById(R.id.button2);

clear.setOnClickListener(new OnClickListener() {

public void onClick(View v) {

try

{

txtData1.setText("");

txtData2.setText("");

}

catch(Exception e)

{

Toast.makeText(getBaseContext(), e.getMessage(),

Toast.LENGTH\_SHORT).show();

}

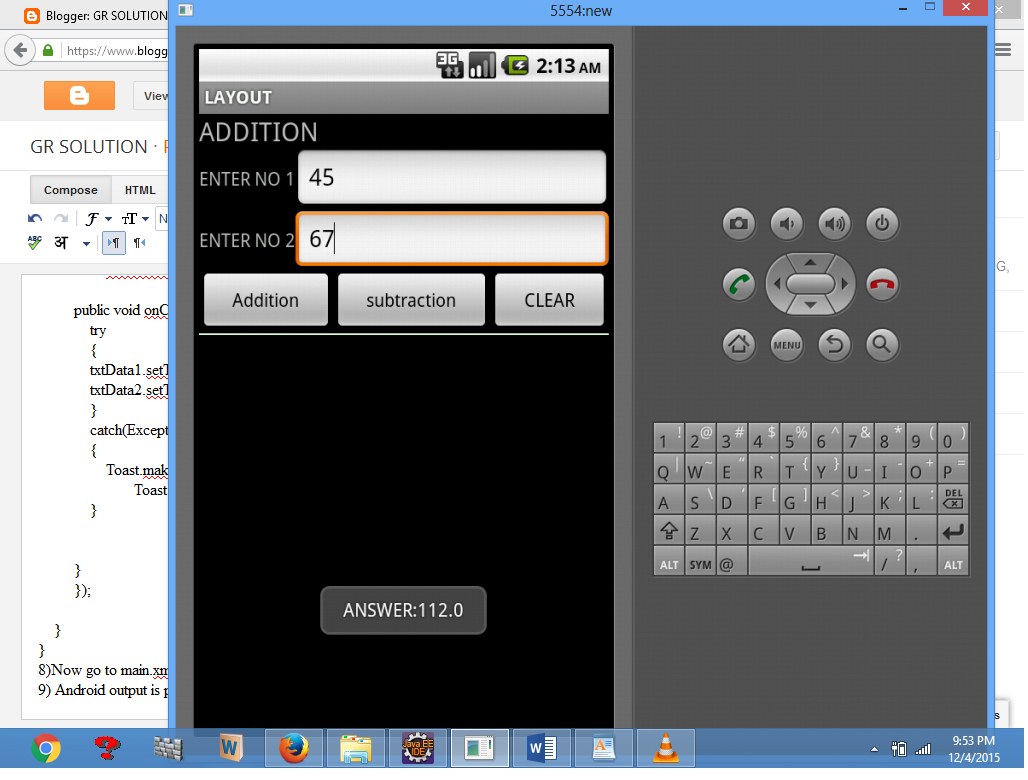
}

});

}

}

**OUTPUT:**



**RESULT:**

Thus the Project to Develop an Application that Uses Layout Managers and Event Listeners **w**as Developed Using Android Suite Software Successfully.

**OUTCOME:**

Understand the programming technique and tools (layout and Event Listener) of a mobile device.

| **EX.NO: 3** | **WRITE AN APPLICATION THAT DRAWS BASIC GRAPHICAL PRIMITIVES ON THE SCREEN** |
| --- | --- |
| **DATE:** |

**AIM:**

To develop an application that draws basic graphical primitives on the screen in android.

**PROCEDURE:**

1. Open android studio and select new android project.
2. Give project name and select next.
3. Choose the android version .Choose the lowest android version(Android 2.2)
4. Enter the package name. Package name must be two word separated by comma and click finish.
5. Go to package explorer in the left hand side. Select our project.
6. Go to res folder and select layout.Double click the main.xml file.Don’t change anything in layout.Leave as default.
7. Now select mainactivity.java file.
8. Now go to main.xml and right click. Select run as option and select run configuration.
9. Android output is present in the android emulator.

**SOURCE CODE:**

**Activity.java**

packageBasic.primitive;

importandroid.app.Activity;

importandroid.content.Context;

importandroid.graphics.Canvas;

importandroid.graphics.Color;

importandroid.graphics.Paint;

importandroid.os.Bundle;

importandroid.view.View;

public class BasicprimitiveActivity extends Activity {

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

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(new myview(this));

}

private class myview extends View {

publicmyview(Context context)

{

super(context);

}

@Override

protected void onDraw(Canvas canvas)

{

super.onDraw(canvas);

Paint paint=new Paint();

paint.setTextSize(40);

paint.setColor(Color.GREEN);

canvas.drawText(“Circle”, 55, 30, paint);

paint.setColor(Color.RED);

canvas.drawCircle(100, 150,100, paint);

paint.setColor(Color.GREEN);

canvas.drawText(“Rectangle”, 255, 30, paint);

paint.setColor(Color.YELLOW);

canvas.drawRect(250, 50,400,350, paint);

paint.setColor(Color.GREEN);

canvas.drawText(“SQUARE”, 55, 430, paint);

paint.setColor(Color.BLUE);

canvas.drawRect(50, 450,150,550, paint);

paint.setColor(Color.GREEN);

canvas.drawText(“LINE”, 255, 430, paint);

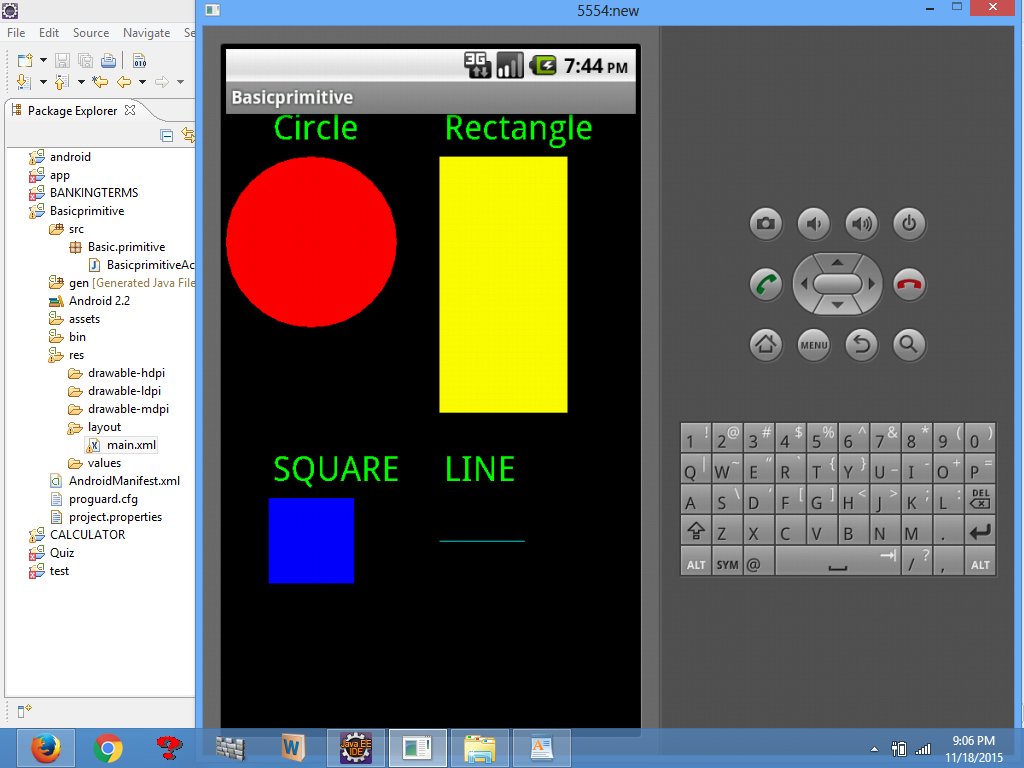
paint.setColor(Color.CYAN);

canvas.drawLine(250, 500, 350, 500, paint);

}

}

**OUTPUT:**



**RESULT:**

Thus the Project to develop an application that draws basic graphical primitives on the screen in android **w**as Developed Using Android Suite Software Successfully.

**OUTCOME:**

Apply the Knowledge to draw Graphical primitives for modern Real time Mobile Application

| **EX.NO: 4** | **DEVELOP AN APPLICATION THAT MAKES USE OF DATABASE** |
| --- | --- |
| **DATE:** |

**AIM:**

To develop an application that makes use of database.

**PROCEDURE:**

1. Open android studio and select new android project.
2. Give project name and select next.
3. Choose the android version. Choose the lowest android version(Android 2.2) .
4. Enter the package name. Package name must be two word separated by comma and click finish.
5. Go to package explorer in the left hand side.
6. Go to res folder and select layout. Double click the main.xml file.
7. Go to values folder and select string.xml file.
8. Now select mainactivity.java file and type the following code.
9. Now go to main.xml and right click .select run as option and select run configuration.
10. Android output is present in the android emulator.

**SOURCE CODE:**

**Layout.xml**

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

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

android:id="@+id/myLayout"

android:stretchColumns="0"

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent">

<TextView android:text="@string/title"

android:layout\_x="110dp"

android:layout\_y="10dp"

android:layout\_width="wrap\_content"

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

<TextView android:text="@string/empid"

android:layout\_x="30dp"

android:layout\_y="50dp"

android:layout\_width="wrap\_content"

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

<EditText android:id="@+id/editEmpid"

android:inputType="number"

android:layout\_x="150dp"

android:layout\_y="50dp"

android:layout\_width="150dp"

android:layout\_height="40dp"/>

<TextView android:text="@string/name"

android:layout\_x="30dp"

android:layout\_y="100dp"

android:layout\_width="wrap\_content"

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

<EditText android:id="@+id/editName"

android:inputType="text"

android:layout\_x="150dp"

android:layout\_y="100dp"

android:layout\_width="150dp"

android:layout\_height="40dp"/>

<TextView android:text="@string/salary"

android:layout\_x="30dp" android:layout\_y="150dp"

android:layout\_width="wrap\_content"

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

<EditText android:id="@+id/editsalary"

android:inputType="number"

android:layout\_x="150dp"

android:layout\_y="150dp"

android:layout\_width="150dp"

android:layout\_height="40dp"/>

<Button android:id="@+id/btnAdd"

android:text="@string/add"

android:layout\_x="30dp"

android:layout\_y="200dp"

android:layout\_width="130dp"

android:layout\_height="40dp"/>

<Button android:id="@+id/btnDelete"

android:text="@string/delete"

android:layout\_x="160dp"

android:layout\_y="200dp"

android:layout\_width="130dp"

android:layout\_height="40dp"/>n

<Button android:id="@+id/btnModify"

android:text="@string/modify"

android:layout\_x="30dp"

android:layout\_y="250dp"

android:layout\_width="130dp"

android:layout\_height="40dp"/>

<Button android:id="@+id/btnView"

android:text="@string/view"

android:layout\_x="160dp"

android:layout\_y="250dp"

android:layout\_width="130dp"

android:layout\_height="40dp"/>

<Button android:id="@+id/btnViewAll"

android:text="@string/view\_all"

android:layout\_x="85dp"

android:layout\_y="300dp"

android:layout\_width="150dp"

android:layout\_height="40dp"/>

</AbsoluteLayout>

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

<resources>

<string name="app\_name">Employee detail1</string>

<string name="hello">Hello World, Employee detail Activity!</string>

<string name="title">Employee Details</string>

<string name="empid">Enter Employee ID: </string>

<string name="name">Enter Name: </string>

<string name="salary">Enter salary: </string>

<string name="add">Add Employee</string>

<string name="delete">Delete Employee</string>

<string name="modify">Modify Employee</string>

<string name="view">View Employee</string>

<string name="view\_all">View All Employee</string>

</resources>

maniactivity name is EmployeedetailActivity.

package student.detail;

**Activity.java:**

//import android.R;

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 Studentdetail1Activity extends Activity implements OnClickListener {

EditText editEmpid,editName,editsalary;

Button btnAdd,btnDelete,btnModify,btnView,btnViewAll;

SQLiteDatabase db;

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

@Override

public void onCreate(Bundle savedInstanceState)

{

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

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

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

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

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

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

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

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

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

btnAdd.setOnClickListener(this);

btnDelete.setOnClickListener(this);

btnModify.setOnClickListener(this);

btnView.setOnClickListener(this);

btnViewAll.setOnClickListener(this);

db=openOrCreateDatabase("EmployeeDB", Context.MODE\_PRIVATE, null);

db.execSQL("CREATE TABLE IF NOT EXISTS employee(empid VARCHAR,name VARCHAR,salary VARCHAR);");

}

public void onClick(View view)

{

if(view==btnAdd)

{

if(editEmpid.getText().toString().trim().length()==0||

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

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

{

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

return;

}

db.execSQL("INSERT INTO employee VALUES('"+editEmpid.getText()+"','"+editName.getText()+

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

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

clearText();

}

if(view==btnDelete)

{

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

{

showMessage("Error", "Please enter Employee id");

return;

}

Cursor c=db.rawQuery("SELECT \* FROM employee WHERE empid='"+editEmpid.getText()+"'", null);

if(c.moveToFirst())

{

db.execSQL("DELETE FROM employee WHERE empid='"+editEmpid.getText()+"'");

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

}

else

{

showMessage("Error", "Invalid Employee id");

}

clearText();

}

if(view==btnModify)

{

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

{

showMessage("Error", "Please enter Employee id");

return;

}

Cursor c=db.rawQuery("SELECT \* FROM employee WHERE empid='"+editEmpid.getText()+"'", null);

if(c.moveToFirst())

{

db.execSQL("UPDATE employee SET name='"+editName.getText()+"',salary='"+editsalary.getText()+

"' WHERE empid='"+editEmpid.getText()+"'");

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

}

else

{

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

}

clearText();

}

if(view==btnView)

{

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

{

showMessage("Error", "Please enter Employee id");

return;

}

Cursor c=db.rawQuery("SELECT \* FROM employee WHERE empid='"+editEmpid.getText()+"'", null);

if(c.moveToFirst())

{

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

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

}

else

{

showMessage("Error", "Invalid Employee id");

clearText();

}

}

if(view==btnViewAll)

{

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

if(c.getCount()==0)

{

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

return;

}

StringBuffer buffer=new StringBuffer();

while(c.moveToNext())

{

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

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

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

}

showMessage("Employee details 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()

{

editEmpid.setText("");

editName.setText("");

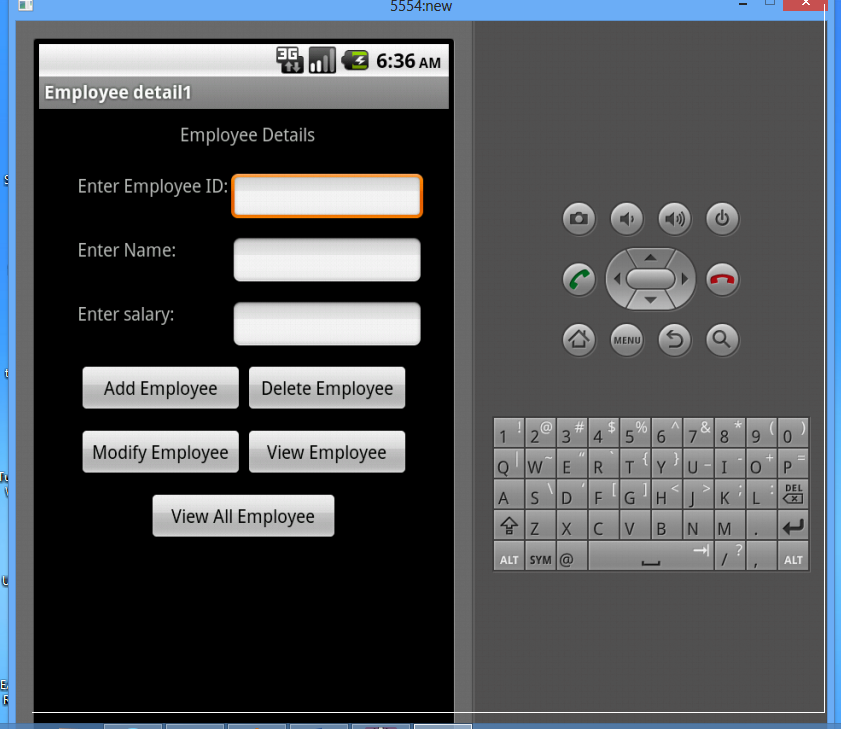
editsalary.setText("");

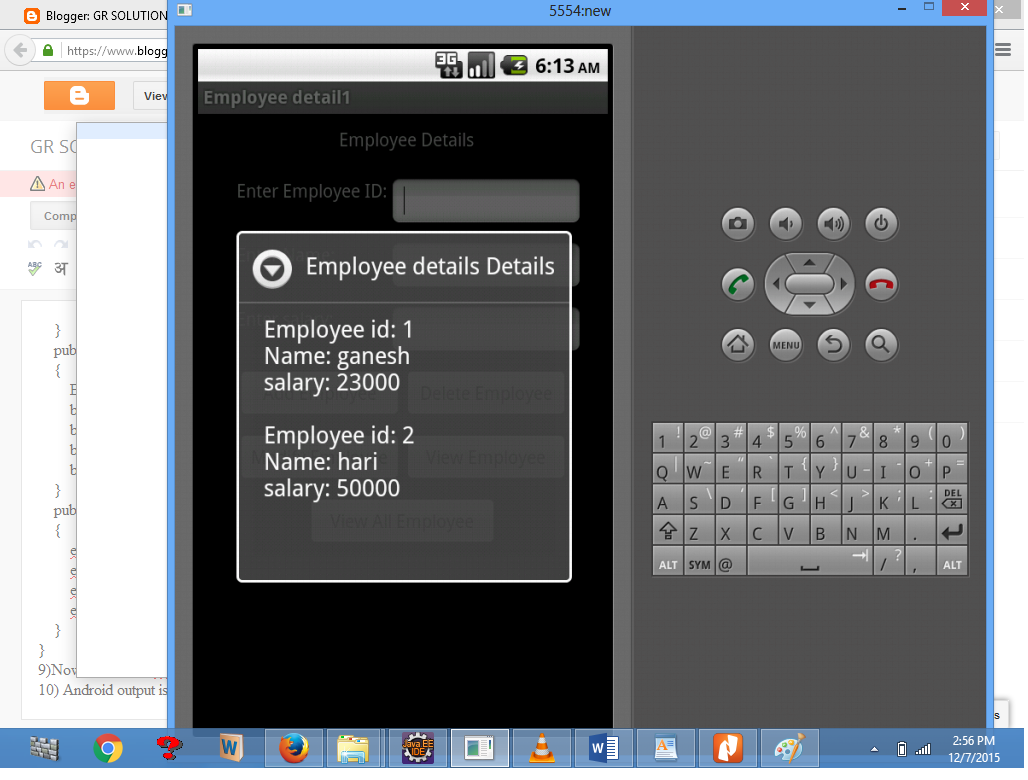
editEmpid.requestFocus();

}

}

**OUTPUT:**





**RESULT:**

Thus the Project to develop an application that makes use of database **w**as Developed Using Android Suite Software Successfully.

**OUTCOME:**

Understand the basic concept of Database using SQL Database for the mobile application.

| .EX.NO: 5 | DEVELOP AN APPLICATION THAT MAKES USE OF NOTIFICATION MANAGER |
| --- | --- |
| Date: |

# **AIM:**

To implement an application to develop An Application That Makes Use Of Notification Manager

# **PROCEDURE:**

1. Select Apps in the Settings, select the app you are interested in and unset the Show notifications checkbox to prevent the app to show notifications. Give project name and select next.
2. Notifications in Android are represented by the Notification class. To create notifications you use the NotificationManager class which can be received from the Context, e.g. an *activity* or a *service*, via the getSystemService() method Enter the

package name. Package name must be two word separated by comma and click finish.

The Notification.Builder provides an builder interface to create an Notification object. You use a PendingIntent to specify the action which should be performed once the user select the notification.

The Notification.Builder allows you to add up to three buttons with definable actions to the notification.

3). Create a new project called *de.vogella.android.notificationmanager* with the *activity* class called CreateNotificationActivity. This *activity* should use the *main.xml* layout file. Now select mainactivity.java file .

4). Create the following *result.xml* layout file. Android output is present in the android emulator.

5). Change the CreateNotificationActivity class to the following coding.

## **Activity\_main.xml**

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

<android.support.constraint.ConstraintLayout

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

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

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<Button

android:text="Notify"

android:textAllCaps="false"

android:layout\_width="200dp"

android:layout\_height="wrap\_content"

android:background="@drawable/shape\_button"

android:id="@+id/btnNotify"

android:textColor="@android:color/white"

android:textSize="18dp"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintBottom\_toBottomOf="parent"/>

</android.support.constraint.ConstraintLayout>

## activity\_notifymessage.xml

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

<android.support.constraint.ConstraintLayout

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

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

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context="com.spaceo.notificationalertdemo.MainActivity">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Hello World!"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintLeft\_toLeftOf="parent"

app:layout\_constraintRight\_toRightOf="parent"

app:layout\_constraintTop\_toTopOf="parent"/>

</android.support.constraint.ConstraintLayout>

## **MainActivity**

package com.spaceo.notificationalertdemo

import android.app.Notification

import android.app.NotificationChannel

import android.app.NotificationManager

import android.app.PendingIntent

import android.content.Context

import android.content.Intent

import android.os.Build

import android.os.Bundle

import android.support.v4.app.NotificationCompat

import android.support.v7.app.AppCompatActivity

import kotlinx.android.synthetic.main.activity\_main.\*

class MainActivity : AppCompatActivity() {

private val NOTIFY\_ME\_ID = 1337

private val NOTIFICATION\_CHANNEL\_ID = "My Channel"

override fun onCreate(savedInstanceState: Bundle?) {

super.onCreate(savedInstanceState)

setContentView(R.layout.activity\_main)

//After Click On Notification Jump To NotifyMessage

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

//PendingIntent Create

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

//Notification Builder Create

valbuilder=NotificationCompat.Builder(this@MainActivity, NOTIFICATION\_CHANNEL\_ID)

//Custom Notification Create

builder.setSmallIcon(R.drawable.notification)

.setContentTitle(resources.getString(R.string.tvTitleofnotification))

.setContentText(resources.getString(R.string.tvBodyofNotification))

.setContentIntent(pendingIntent)

builder.notification.flags=builder.notification.flagsor Notification.FLAG\_AUTO\_CANCEL builder.setAutoCancel(true)

//NotificationManager Create

val notificationManager = getSystemService(Context.NOTIFICATION\_SERVICE) as NotificationManager

//If API level 26 or higher than create channel for Notification

if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.O) {

//channel Create

val notificationChannel = NotificationChannel(NOTIFICATION\_CHANNEL\_ID,

resources.getString(R.string.app\_name), NotificationManager.IMPORTANCE\_DEFAULT)

notificationManager.createNotificationChannel(notificationChannel)

}

//Button Click After Generate Notification

btnNotify.setOnClickListener {

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

}

}

}

## AndroidManifest.xml

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

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

package="com.spaceo.notificationalertdemo">

<application

android:allowBackup="true"

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

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

android:roundIcon="@mipmap/ic\_launcher\_round"

android:supportsRtl="true"

android:theme="@style/AppTheme">

<activity android:name=".MainActivity">

<intent-filter>

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

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

</intent-filter>

</activity>

<activity android:name=".NotifyMessage"></activity>

</application>

</manifest>

## **OUTPUT:**

| **Description: Main Screen** | **Description: Notification Screen** |
| --- | --- |

**RESULT:**

Thus the Project to implement develop an application that makes use of notification manager Successfully.

# **OUTCOME:**

CO2: Develop mobile applications using Event Listener.

| **EX.NO: 6** | **IMPLEMENT AN APPLICATION THAT IMPLEMENTS MULTITHREADING** |
| --- | --- |
| **DATE:** |

**AIM:**

To implement an application that implements multithreading.

**PROCEDURE:**

1. Open android studio and select new android project.
2. Give project name and select next.
3. Choose the android version. Choose the lowest android version(Android 2.2)
4. Enter the package name. Package name must be two word separated by comma and click finish.
5. Go to package explorer in the left hand side.
6. Go to res folder and select layout. Double click the main.xml file.
7. Now select mainactivity.java file .
8. Now go to main.xml and right click .select run as option and select run configuration.
9. Android output is present in the android emulator.

**SOURCE CODE:**

**Layout.xml**

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

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:id="@+id/info" >

<Button

android:id="@+id/button1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:onClick="fetchData"

android:text="Start MULTITHREAD" />

<TextView

android:id="@+id/textView1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Main thread" />

</LinearLayout>

package multi.threading;

**Activity.java:**

//import your.first.R;

import android.app.Activity;

import android.os.Bundle;

import android.os.Handler;

import android.view.View;

import android.widget.TextView;

public class MultiThreadingActivity extends Activity {

private TextView tvOutput;

private static final int t1 = 1;

private static final int t2 = 2;

private static final int t3 = 3;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

tvOutput = (TextView) findViewById(R.id.textView1);

}

public void fetchData(View v) {

tvOutput.setText("Main thread");

thread1.start();

thread2.start();

thread3.start();

}

Thread thread1 = new Thread(new Runnable() {

@Override

public void run() {

for (int i = 0; i < 5; i++) {

try {

Thread.sleep(1000);

} catch (InterruptedException e) {

e.printStackTrace();

}

handler.sendEmptyMessage(t1);

}

}

});

Thread thread2 = new Thread(new Runnable() {

@Override

public void run() {

for (int i = 0; i < 5; i++) {

try {

Thread.sleep(1000);

} catch (InterruptedException e) {

e.printStackTrace();

}

handler.sendEmptyMessage(t2);

}

}

});

Thread thread3 = new Thread(new Runnable() {

@Override

public void run() {

for (int i = 0; i < 5; i++) {

try {

Thread.sleep(1000);

} catch (InterruptedException e) {

e.printStackTrace();

}

handler.sendEmptyMessage(t3);

}

}

});

Handler handler = new Handler() {

public void handleMessage(android.os.Message msg) {

if(msg.what == t1) {

tvOutput.append("\nIn thread 1");

}

if(msg.what == t2) {

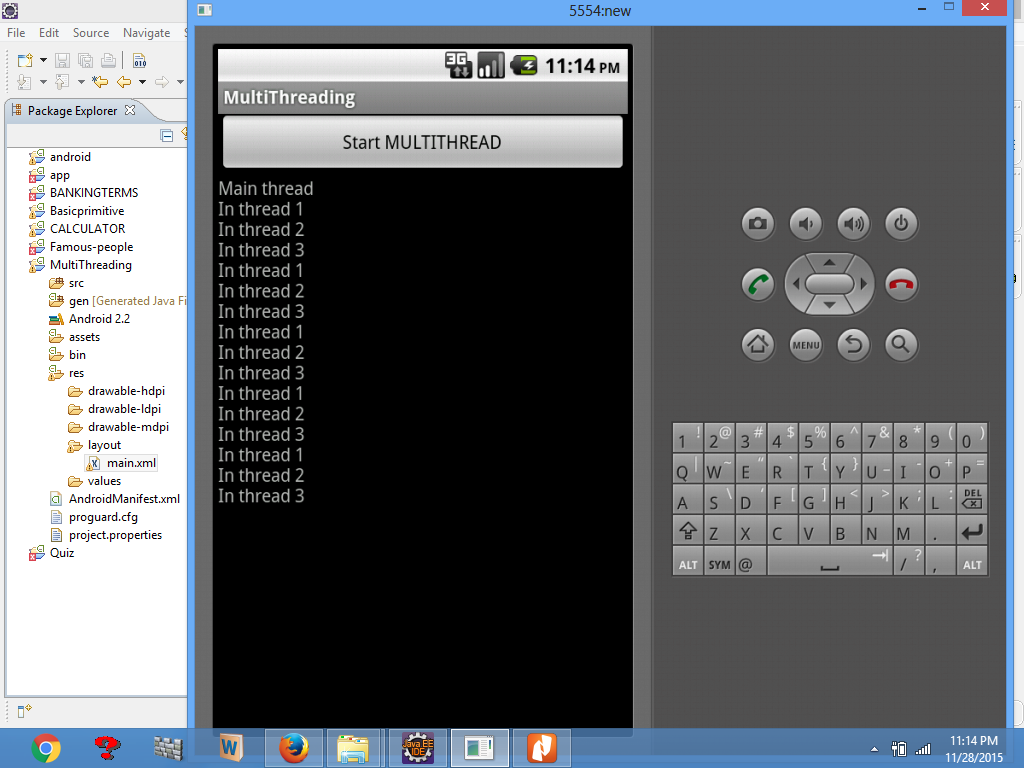
tvOutput.append("\nIn thread 2");

}

if(msg.what == t3) {

tvOutput.append("\nIn thread 3");}}};}

**OUTPUT:**



**RESULT:**

Thus the Project to implement an application that implements multithreading **w**as Developed Using Android Suite Software Successfully.

**OUTCOME:**

Understand an application implements Multithreading for a mobile application.

| **EX.NO: 7** | **DEVELOP A NATIVE APPLICATION THAT USES GPS LOCATION INFORMATION** |
| --- | --- |
| **DATE:** |

**AIM:**

To develop a native application that uses GPS location information.

**PROCEDURE:**

1. Open eclipse or android studio and select new android project
2. Give project name and select next
3. Choose the android version. Choose the lowest android version(Android 2.2)
4. Enter the package name. Package name must be two word separated by comma and click finish
5. Go to package explorer in the left hand side.
6. Go to res folder and select layout. Double click the main.xml file.
7. Now select mainactivity.java file and type the following code.
8. Go to src folder and Right Click on your package folder and choose new class.
9. Select the GPStrace.java file.
10. Now go to main.xml and right click .select run as option and select run configuration.
11. Android output is present in the android emulator as shown in below.

**SOURCE CODE:**

**Layout.xml**

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

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

android:id="@+id/relativeLayout1"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent" >

<Button

android:id="@+id/show\_Location"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Show\_Location"

android:layout\_centerVertical="true"

android:layout\_centerHorizontal="true"

/>

</RelativeLayout>

name is GPSlocationActivity.

package gps.location;

**Activity.java**

//import android.R;

import android.app.Activity;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.Toast;

public class GPSlocationActivity extends Activity {

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

Button btnShowLocation;

GPStrace gps;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

btnShowLocation=(Button)findViewById(R.id.show\_Location);

btnShowLocation.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

// TODO Auto-generated method stub

gps=new GPStrace(GPSlocationActivity.this);

if(gps.canGetLocation()){

double latitude=gps.getLatitude();

double longitude=gps.getLongtiude();

Toast.makeText(getApplicationContext(),"Your Location is \nLat:"+latitude+"\nLong:"+longitude, Toast.LENGTH\_LONG).show();

}

else

{

gps.showSettingAlert();

}

}

});

}

}

package gps.location;

**Activity.java**

import android.app.AlertDialog;

import android.app.Service;

import android.content.Context;

import android.content.DialogInterface;

import android.content.Intent;

import android.location.Location;

import android.location.LocationListener;

import android.location.LocationManager;

import android.os.Bundle;

import android.os.IBinder;

import android.provider.Settings;

public class GPStrace extends Service implements LocationListener{

private final Context context;

boolean isGPSEnabled=false;

boolean canGetLocation=false;

boolean isNetworkEnabled=false;

Location location;

double latitude;

double longtitude;

private static final long MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES=10;

private static final long MIN\_TIME\_BW\_UPDATES=1000\*60\*1;

protected LocationManager locationManager;

public GPStrace(Context context)

{

this.context=context;

getLocation();

}

public Location getLocation()

{

try{

locationManager=(LocationManager) context.getSystemService(LOCATION\_SERVICE);

isGPSEnabled=locationManager.isProviderEnabled(LocationManager.GPS\_PROVIDER); isNetworkEnabled=locationManager.isProviderEnabled(LocationManager.NETWORK\_PROVIDER);

if(!isGPSEnabled && !isNetworkEnabled){

}else{

this.canGetLocation=true;

if(isNetworkEnabled){

locationManager.requestLocationUpdates(

LocationManager.NETWORK\_PROVIDER,

MIN\_TIME\_BW\_UPDATES,

MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES,this);

}

if(locationManager!=null){ location=locationManager.getLastKnownLocation(LocationManager.NETWORK\_PROVIDER);

if(location !=null){

latitude=location.getLatitude();

longtitude=location.getLongitude();

}

}

}

if(isGPSEnabled){

if(location==null){ locationManager.requestLocationUpdates(LocationManager.GPS\_PROVIDER,MIN\_TIME\_BW\_UPDATES, MIN\_DISTANCE\_CHANGE\_FOR\_UPDATES, this);

if(locationManager!=null){

location=locationManager.getLastKnownLocation(LocationManager.GPS\_PROVIDER);

if(location!=null){

latitude=location.getLatitude();

longtitude=location.getLongitude();

}

}

}

}

}

catch(Exception e)

{

e.printStackTrace();

}

return location;

}

public void stopUsingGPS(){

if(locationManager!=null){

locationManager.removeUpdates(GPStrace.this);

}

}

public double getLatitude(){

if(location!=null){

latitude=location.getLatitude();

}

return latitude;

}

public double getLongtiude(){

if(location!=null){

longtitude=location.getLatitude();

}

return longtitude;

}

public boolean canGetLocation(){

return this.canGetLocation;

}

public void showSettingAlert(){

AlertDialog.Builder alertDialog=new AlertDialog.Builder(context);

alertDialog.setTitle("GPS is settings");

alertDialog.setMessage("GPS is not enabled.Do you want to go to setting menu?");

alertDialog.setPositiveButton("settings", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog,int which){

Intent intent=new Intent(Settings.ACTION\_LOCATION\_SOURCE\_SETTINGS);

context.startActivity(intent);

}

});

alertDialog.setNegativeButton("cancel", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialog, int which) {

// TODO Auto-generated method stub

dialog.cancel();

}

});

alertDialog.show();

}

@Override

public void onLocationChanged(Location location) {

// TODO Auto-generated method stub

}

@Override

public void onProviderDisabled(String provider) {

// TODO Auto-generated method stub

}

@Override

public void onProviderEnabled(String provider) {

// TODO Auto-generated method stub

}

@Override

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

// TODO Auto-generated method stub

}

@Override

public IBinder onBind(Intent intent) {

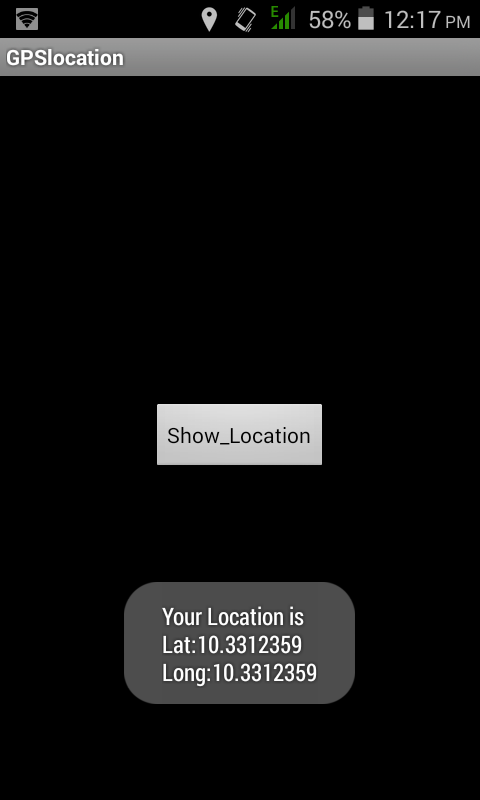
// TODO Auto-generated method stub

return null;

}

}

**OUTPUT:**



**RESULT:**

Thus the Project to develop a native application that uses GPS location information **w**as Developed Using Eclipse Suite Software Successfully.

**OUTCOME:**

Understand the capability and features of GPS location information for a mobile application.

| **EX.NO: 8** | **IMPLEMENT AN APPLICATION THAT WRITES DATA TO THE SD CARD** |
| --- | --- |
| **DATE:** |

**AIM:**

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

**PROCEDURE:**

1. Open eclipse or android studio and select new android project
2. Give project name and select next
3. Choose the android version.Choose the lowest android version(Android 2.2) .
4. Enter the package name. Package name must be two word separated by comma and click finish
5. Go to package explorer in the left hand side.
6. Go to res folder and select layout. Double click the main.xml file.
7. Now select mainactivity.java file.
8. Next step is to set permission to write data in sd card.So go to AndroidManifest.xml file.
9. Now go to main.xml and right click .select run as option and select run configuration.
10. Android output is present in the android emulator as shown in below.

**SOURCE CODE:**

**Layout.xml**

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

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

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:background="#ff0000ff"

android:orientation="vertical" >

<EditText

android:id="@+id/editText1"

android:layout\_width="match\_parent"

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

<requestFocus />

</EditText>

<Button

android:id="@+id/button1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="SAVE DATA" />

<Button

android:id="@+id/button2"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:text="SHOW DATA" />

<TextView

android:id="@+id/textView1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

/>

</LinearLayout>

package save.sd;

**Activity.java**

import java.io.File;

import java.io.FileInputStream;

import java.io.FileNotFoundException;

import java.io.FileOutputStream;

import java.io.IOException;

import java.io.InputStreamReader;

import java.io.OutputStreamWriter;

import android.app.Activity;

import android.os.Bundle;

import android.os.Environment;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

public class SavedatasdcardActivity extends Activity {

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

Button save,load;

EditText message;

TextView t1;

String Message1;

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

save=(Button) findViewById(R.id.button1);

load=(Button) findViewById(R.id.button2);

message=(EditText) findViewById(R.id.editText1);

t1=(TextView) findViewById(R.id.textView1);

save.setOnClickListener(new View.OnClickListener(){

public void onClick(View v){

//Get message from user store in message1 variable

Message1 =message.getText().toString();

try{

//Create a new folder called MyDirectory in SDCard

File sdcard=Environment.getExternalStorageDirectory();

File directory=new File(sdcard.getAbsolutePath()+"/MyDirectory");

directory.mkdirs();

//Create a new file name textfile.txt inside MyDirectory

File file=new File(directory,"textfile.txt");

//Create File Outputstream to read the file

FileOutputStream fou=new FileOutputStream(file);

OutputStreamWriter osw=new OutputStreamWriter(fou);

try{

//write a user data to file

osw.append(Message1);

osw.flush();

osw.close();

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

}catch(IOException e){

e.printStackTrace();

}

}catch (FileNotFoundException e){

e.printStackTrace();

}

}

});

load.setOnClickListener(new View.OnClickListener(){

public void onClick(View v){

try{

File sdcard=Environment.getExternalStorageDirectory();

File directory=new File(sdcard.getAbsolutePath()+"/MyDirectory");

File file=new File(directory,"textfile.txt");

FileInputStream fis=new FileInputStream(file);

InputStreamReader isr=new InputStreamReader(fis);

char[] data=new char[100];

String final\_data="";

int size;

try{

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

{

//read a data from file

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

final\_data+=read\_data;

data=new char[100];

}

//display the data in output

Toast.makeText(getBaseContext(),"Message:"+final\_data,Toast.LENGTH\_LONG).show();

}catch(IOException e){

e.printStackTrace();

}

}catch (FileNotFoundException e){

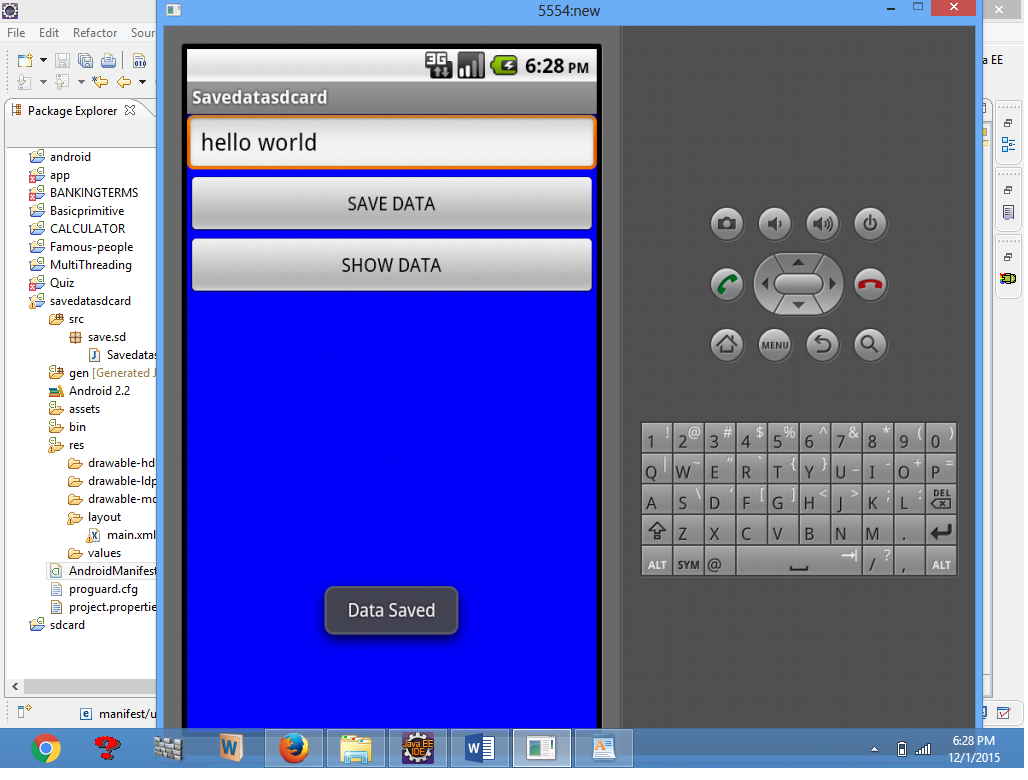
e.printStackTrace();

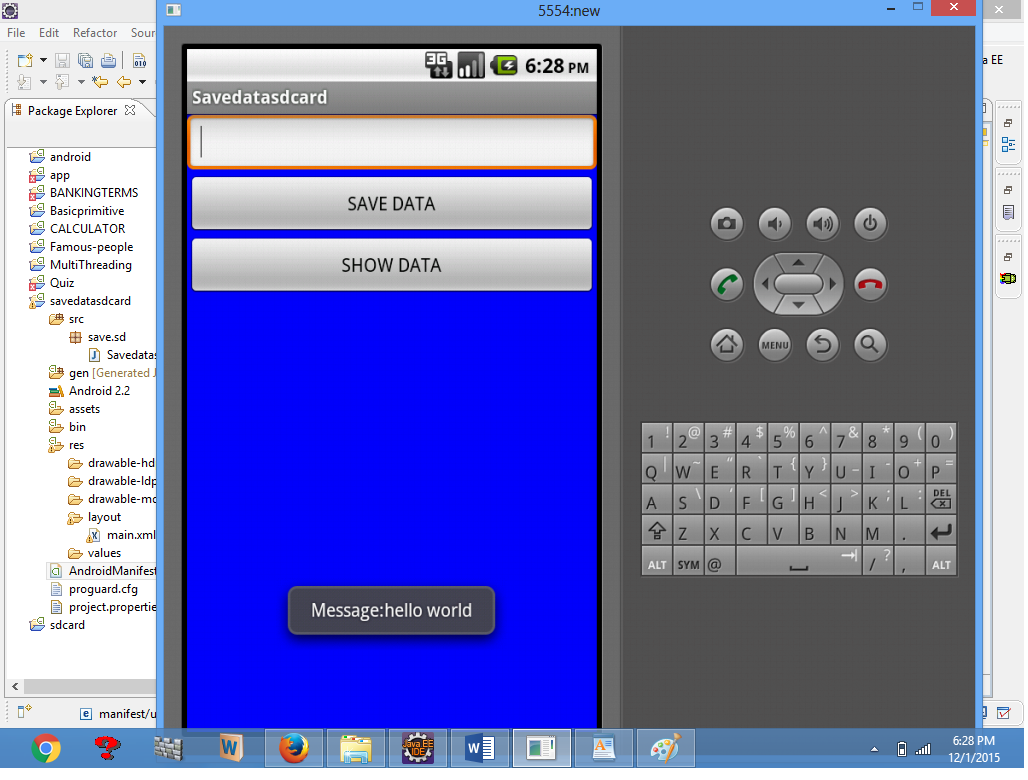
}

}

}); } }

**OUTPUT:**





**RESULT:**

Thus the Project to implement an application that writes data to the SD Card **w**as Developed Using Eclipse Suite Software Successfully.

**OUTCOME:**

Understand the concept of SD card for a mobile application with the help of emulator

| **EX.NO: 9** | **IMPLEMENT AN APPLICATION THAT CREATES AN ALERT UPON RECEIVING A MESSAGE IN ANDROID** |
| --- | --- |
| **DATE:** |

**AIM:**

To implement an application that creates an alert upon receiving a message in android

**PROCEDURE:**

1. Open eclipse or android studio and select new android project
2. Give project name and select next
3. Choose the android version. Choose the lowest android version(Android 2.2) .
4. Enter the package name. Package name must be two word separated by comma and click finish
5. Go to package explorer in the left hand side.
6. Go to res folder and select layout. Double click the main.xml file.
7. Now select mainactivity.java file and type the following code.
8. Now go to main.xml and right click .select run as option and select run configuration
9. Android output is present in the android emulator as shown in below.

**SOURCE CODE:**

**Layout.xml**

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

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:scrollbars="vertical" >

<TableLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:shrinkColumns="\*" android:stretchColumns="\*" android:background="#000000">

<TableRow

android:layout\_height="wrap\_content"

android:layout\_width="match\_parent"

android:gravity="center\_horizontal">

<TextView

android:id="@+id/Title"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="5px"

android:focusable="false"

android:focusableInTouchMode="false"

android:gravity="center\_vertical|center\_horizontal"

android:text="QUIZ"

android:textSize="25sp"

android:textStyle="bold" />

<View

android:layout\_height="2px"

android:layout\_marginTop="5dip"

android:layout\_marginBottom="5dip"

android:background="#DDFFDD"/>

</TableRow>

<TableRow

android:layout\_height="wrap\_content"

android:layout\_width="match\_parent"

android:gravity="center\_horizontal">

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textSize="18sp" android:text="1.CAPTIAL OF INDIA"

android:layout\_span="4"

android:padding="18dip"

android:textColor="#ffffff"/>

</TableRow>

<TableRow

android:id="@+id/tableRow1"

android:layout\_height="wrap\_content"

android:layout\_width="match\_parent">

<RadioGroup

android:id="@+id/answer1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_weight="0.4" >

<RadioButton

android:id="@+id/answer1A"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textColor="#ffffff"

android:text="CHENNAI" />

<RadioButton

android:id="@+id/answer1B"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textColor="#ffffff"

android:text="NEW DELHI" />

<RadioButton

android:id="@+id/answer1C"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textColor="#ffffff"

android:text="MUMBAI" />

<RadioButton

android:id="@+id/answer1D"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textColor="#ffffff"

android:text="HYDERBAD" />

</RadioGroup>

</TableRow>

<TableRow

android:layout\_height="wrap\_content"

android:layout\_width="match\_parent"

android:gravity="center\_horizontal">

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textSize="18sp"

android:text="2. CAPTIAL OF RUSSIA?"

android:layout\_span="4"

android:padding="18dip"

android:textColor="#ffffff"/>

</TableRow>

<TableRow

android:id="@+id/tableRow2"

android:layout\_height="wrap\_content"

android:layout\_width="match\_parent">

<RadioGroup

android:id="@+id/answer2"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_weight="0.4" >

<RadioButton

android:id="@+id/answer2A"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textColor="#ffffff"

android:text="WARSAW " />

<RadioButton

android:id="@+id/answer2B"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textColor="#ffffff"

android:text="BERLIN" />

<RadioButton

android:id="@+id/answer2C"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textColor="#ffffff"

android:text="MASCOW " />

<RadioButton

android:id="@+id/answer2D"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:textColor="#ffffff"

android:text="CANEBRA " />

</RadioGroup>

</TableRow>

<TableRow

android:layout\_height="wrap\_content"

android:layout\_width="match\_parent"

android:gravity="center\_horizontal">

<Button

android:id="@+id/submit"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:gravity="center"

android:text="Submit"

/>

</TableRow>

</TableLayout>

</ScrollView>

package alert1.quiz;

**Activity.java**

import android.app.Activity;

import android.app.AlertDialog;

import android.content.DialogInterface;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.RadioGroup;

import android.widget.Toast;

import android.view.View.OnClickListener;

import android.widget.RadioGroup.OnCheckedChangeListener;

public class Alert1Activity extends Activity {

private Button btnSubmitQuiz;

int score,ans1,ans2;

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

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

RadioGroup b1=(RadioGroup)findViewById(R.id.answer1);

b1.setOnCheckedChangeListener(new OnCheckedChangeListener() {

public void onCheckedChanged(RadioGroup group, int checkedId) {

// TODO Auto-generated method stub

switch(checkedId) {

case R.id.answer1A:

ans1 =1;

break;

case R.id.answer1B:

ans1 =2;

break;

case R.id.answer1C:

ans1 =3;

break;

case R.id.answer1D:

ans1=4;

break;

}

}

});

RadioGroup b2=(RadioGroup)findViewById(R.id.answer2);

b2.setOnCheckedChangeListener(new OnCheckedChangeListener() {

public void onCheckedChanged(RadioGroup group, int checkedId) {

// TODO Auto-generated method stub

switch(checkedId) {

case R.id.answer2A:

ans2 =1;

break;

case R.id.answer2B:

ans2 =2;

break;

case R.id.answer2C:

ans2 =3;

break;

case R.id.answer2D:

ans2=4;

break;

}

}

});

btnSubmitQuiz = (Button) findViewById(R.id.submit);

btnSubmitQuiz.setOnClickListener(new OnClickListener() {

public void onClick(View v) {

AlertDialog.Builder alertDialog = new AlertDialog.Builder(Alert1Activity.this);

// Setting Dialog Title

alertDialog.setTitle("SHOW RESULT");

// Setting Dialog Message

alertDialog.setMessage("Are you sure you want SUBMIT this?");

// Setting Icon to Dialog

// alertDialog.setIcon(R.drawable.tick);

// Setting Positive "Yes" Button

alertDialog.setPositiveButton("YES", new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog,int which) {

score =0;

// TODO Auto-generated method stub

if(ans1 == 2)

score++;

if(ans2 == 3)

score++;

Toast.makeText(Alert1Activity.this, "Your score is:"+score+" out of 2.", Toast.LENGTH\_LONG).show();

}

});

// Setting Negative "NO" Button

alertDialog.setNegativeButton("NO", new DialogInterface.OnClickListener() {

public void onClick(DialogInterface dialog, int which) {

// Write your code here to invoke NO event

Toast.makeText(getApplicationContext(), "You clicked NO.CHECK YOUR ANSWER", Toast.LENGTH\_SHORT).show();

dialog.cancel();

}

});

// Showing Alert Message

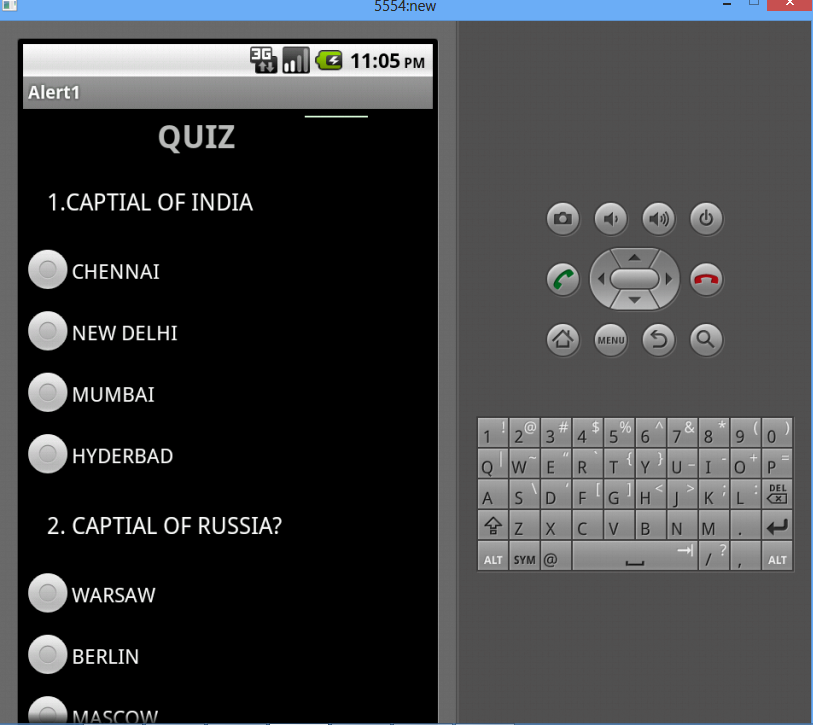
alertDialog.show();

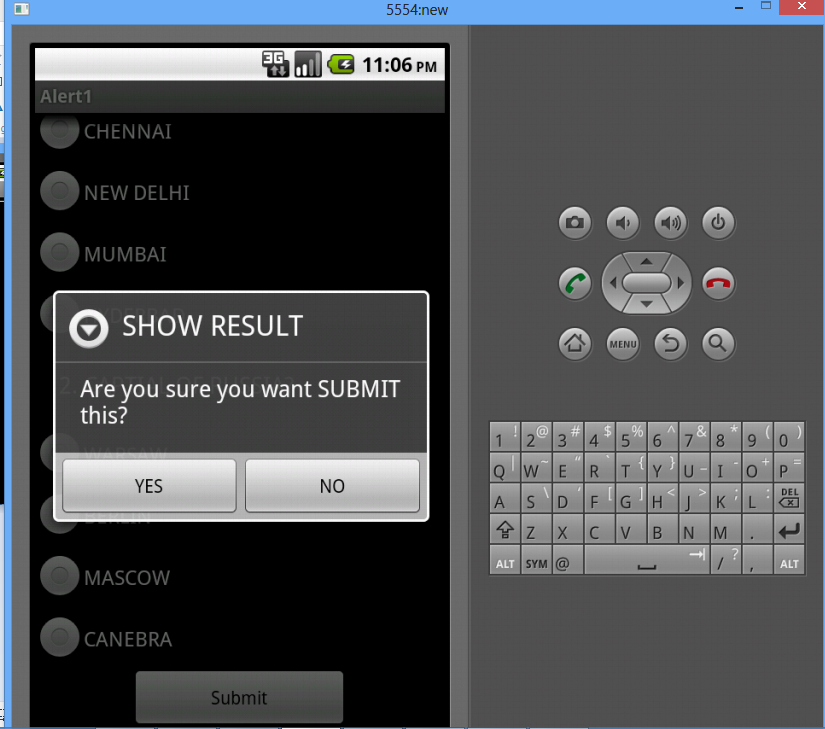
}

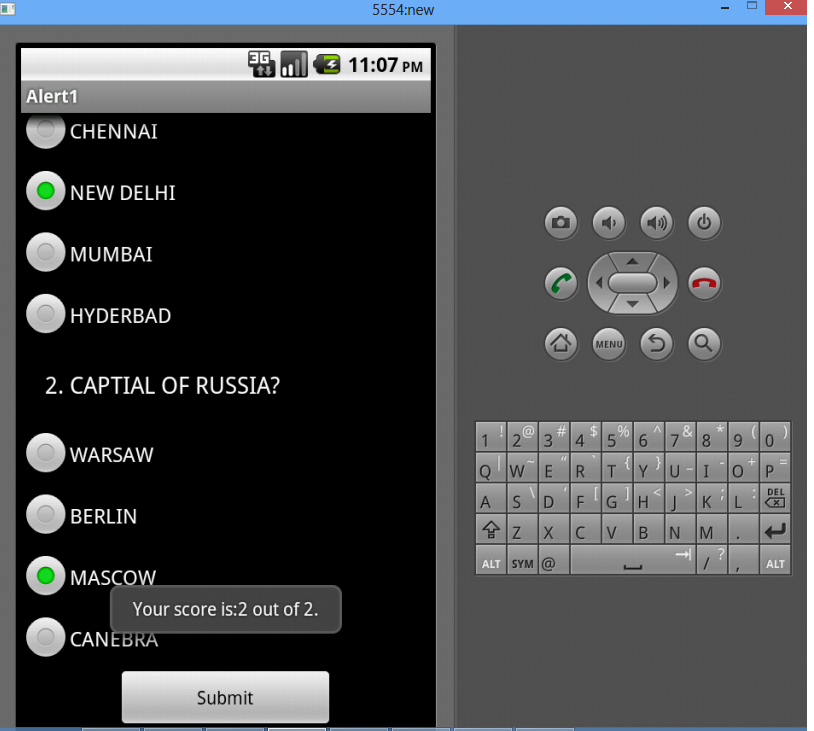
});

}}

**OUTPUT:**







**RESULT:**

Thus the project to implement an application that creates an alert upon receiving a message in android was developed using Eclipse Suite Software Successfully.

**OUTCOME:**

Development of application that creates an alert message using mobile application

| **EX.NO: 10** | **DEVELOP AN APPLICATION THAT MAKES USE OF RSS FEED** |
| --- | --- |
| **DATE:** |

**AIM:**

To develop an application that makes use of RSS Feed

**PROCEDURE:**

1. Open android studio and select new android project.
2. Give project name and select next.
3. Choose the android version. Choose the lowest android version(Android 2.2) .
4. Enter the package name. Package name must be two word separated by comma and click finish.
5. Go to package explorer in the left hand side.
6. Go to res folder and select layout. Double click the main.xml file.
7. Now select mainactivity.java.
8. Go to AndroidManifest.xml file and paste the following code before <appllication> tab

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

1. Now go to main.xml and right click .select run as option and select run configuration.
2. Android output is present in the android emulator.

**SOURCE CODE:**

**Layout.xml:**

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

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

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:orientation="vertical" >

<ListView

android:id="@+android:id/list"

android:layout\_width="fill\_parent"

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

</ListView>

</LinearLayout>

package rss.feed;

**Activity.java:**

//import android.app.Activity;

import java.io.IOException;

import java.io.InputStream;

import java.net.MalformedURLException;

import java.net.URL;

import java.util.ArrayList;

import java.util.List;

import org.xmlpull.v1.XmlPullParser;

import org.xmlpull.v1.XmlPullParserException;

import org.xmlpull.v1.XmlPullParserFactory;

import android.app.ListActivity;

import android.content.Intent;

import android.net.Uri;

import android.os.Bundle;

import android.view.View;

import android.widget.ArrayAdapter;

import android.widget.ListView;

public class RssfeedActivity extends ListActivity {

List headlines;

List links;

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

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

headlines = new ArrayList();

links = new ArrayList();

try {

URL url = new URL("http://feeds.pcworld.com/pcworld/latestnews");

XmlPullParserFactory factory = XmlPullParserFactory.newInstance();

factory.setNamespaceAware(false);

XmlPullParser xpp = factory.newPullParser();

// We will get the XML from an input stream

xpp.setInput(getInputStream(url), "UTF\_8");

/\* We will parse the XML content looking for the "<title>" tag which appears inside the "<item>" tag.

\* However, we should take in consideration that the rss feed name also is enclosed in a "<title>" tag.

\* As we know, every feed begins with these lines: "<channel><title>Feed\_Name</title>...."

\* so we should skip the "<title>" tag which is a child of "<channel>" tag,

\* and take in consideration only "<title>" tag which is a child of "<item>"

\*

\* In order to achieve this, we will make use of a boolean variable.

\*/

boolean insideItem = false;

// Returns the type of current event: START\_TAG, END\_TAG, etc..

int eventType = xpp.getEventType();

while (eventType != XmlPullParser.END\_DOCUMENT) {

if (eventType == XmlPullParser.START\_TAG) {

if (xpp.getName().equalsIgnoreCase("item")) {

insideItem = true;

} else if (xpp.getName().equalsIgnoreCase("title")) {

if (insideItem)

headlines.add(xpp.nextText()); //extract the headline

} else if (xpp.getName().equalsIgnoreCase("link")) {

if (insideItem)

links.add(xpp.nextText()); //extract the link of article

}

}else if(eventType==XmlPullParser.END\_TAG && xpp.getName().equalsIgnoreCase("item")){

insideItem=false;

}

eventType = xpp.next(); //move to next element

}

} catch (MalformedURLException e) {

e.printStackTrace();

} catch (XmlPullParserException e) {

e.printStackTrace();

} catch (IOException e) {

e.printStackTrace();

}

// Binding data

ArrayAdapter adapter = new ArrayAdapter(this,

android.R.layout.simple\_list\_item\_1, headlines);

setListAdapter(adapter);

}

public InputStream getInputStream(URL url) {

try {

return url.openConnection().getInputStream();

} catch (IOException e) {

return null;

}

}

@Override

protected void onListItemClick(ListView l, View v, int position, long id) {

Uri uri = Uri.parse((String) links.get(position));

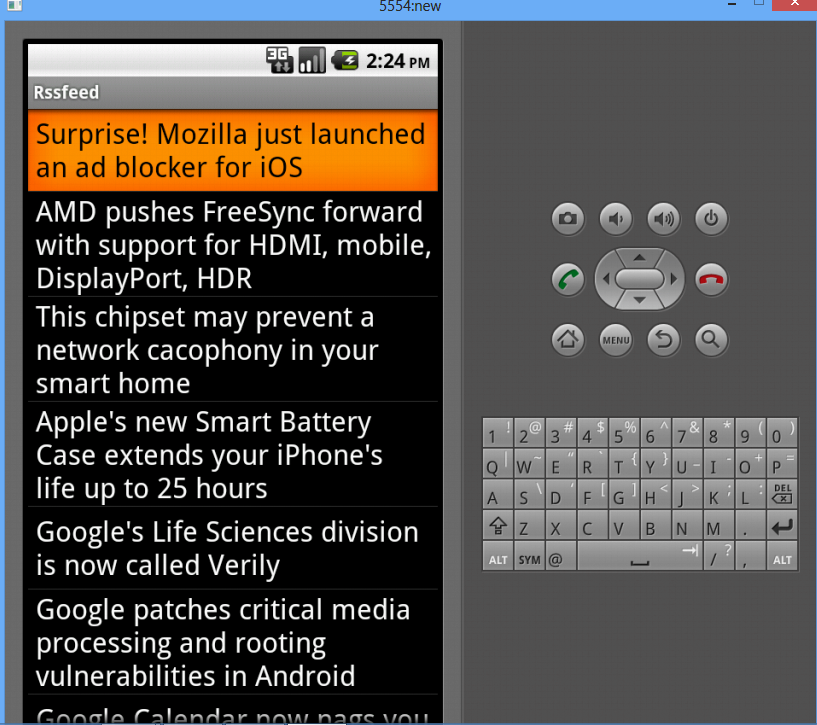
Intent intent = new Intent(Intent.ACTION\_VIEW, uri);

startActivity(intent);

}

}

**OUTPUT:**





**RESULT:**

Thus the Project to develop an application that makes use of RSS Feed **w**as Developed Using Android Suite Software Successfully.

**OUTCOME:**

Understand the Structure of OS that makes use of RSS feed for a mobile Computer System.

| **EX.NO: 11** | **WRITE A MOBILE APPLICATION THAT CREATES ALARM CLOCK IN ANDROID** |
| --- | --- |
| **DATE:** |

**AIM**:

To develop an android application that send an email.

**ALGORITHM:**

1. Create a New Android Project:
   * Click New in the toolbar.
   * In the window that appears, open the Android folder, select Android Application Project, and click next.
   * Provide the application name and the project name and then finally give the desired package name.
   * Choose a launcher icon for your application and then select Blank Activity and then click Next
   * Provide the desired Activity name for your project and then click Finish.
2. Create a New AVD (Android Virtual Device):
   * click Android Virtual Device Manager from the toolbar.
   * In the Android Virtual Device Manager panel, click New.
   * Fill in the details for the AVD. Give it a name, a platform target, an SD card size, and a skin (HVGA is default).
   * Click Create AVD and Select the new AVD from the Android Virtual Device Manager and click Start.
3. Design the graphical layout.
4. Run the application.
5. When the application starts alarm sound will be invoked.
6. Stop alarm button is clicked to stop the alarm.
7. Close the Android project.

**PROGRAM CODE:**

**MainActivity.java**

**package** com.example.admin.myapplication;

**import** android.content.Intent;

**import** android.net.Uri;

**import** android.os.Bundle;

**import** android.support.v7.app.AppCompatActivity;

**import** android.util.Log;

**import** android.view.View;

**import** android.widget.Button;

**import** android.widget.Toast;

**public class** MainActivity**extends** AppCompatActivity {

@Override

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

Button startBtn = (Button) findViewById(R.id.***sendbttn***); startBtn.setOnClickListener(**new** View.OnClickListener() { **public void** onClick(View view) {

sendEmail();

}

});

}

**protected void** sendEmail() {

Log.*i*(**"Send email"**, **""**);

String[] TO = {

# **"**[**muthuramalingam566@gmail.com**](mailto:muthuramalingam566@gmail.com)**"**

};

String[] CC = {

# **"**[**ramdurai25@gmail.com**](mailto:ramdurai25@gmail.com)**"**

};

Intent emailIntent = **new** Intent(Intent.***ACTION\_SEND***); emailIntent.setData(Uri.*parse*(**"mailto:"**)); emailIntent.setType(**"text/plain"**); emailIntent.putExtra(Intent.***EXTRA\_EMAIL***, TO); emailIntent.putExtra(Intent.***EXTRA\_CC***, CC); emailIntent.putExtra(Intent.***EXTRA\_SUBJECT***, **"Your subject"**); emailIntent.putExtra(Intent.***EXTRA\_TEXT***, **"Email message goes here"**); **try** {

startActivity(Intent.*createChooser*(emailIntent, **"Send mail..."**)); finish();

Log.*i*(**"Finished sending email..."**, **""**);

} **catch** (android.content.ActivityNotFoundException ex) { Toast.*makeText*(MainActivity.**this**, **"There is no email client installed."**, Toast.***LENGTH\_SHORT***).show();

}

}

}

**activity\_main.xml**

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

<**RelativeLayoutxmlns:android="**[**http://schemas.android.com/apk/res/android**](http://schemas.android.com/apk/res/android)**" xmlns:tools="**[**http://schemas.android.com/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="com.example.admin.myapplication.MainActivity"**>

<**EditTextandroid:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:inputType="textEmailAddress" android:ems="10"**

# **android:id="@+id/editText" android:layout\_alignParentTop="true" android:layout\_alignParentRight="true" android:layout\_alignParentEnd="true"** />

<**EditText android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:inputType="textEmailAddress" android:ems="10" android:id="@+id/editText2" android:layout\_below="@+id/editText" android:layout\_alignRight="@+id/editText" android:layout\_alignEnd="@+id/editText"** />

# <**EditText android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:inputType="textEmailAddress" android:ems="10" android:id="@+id/editText3" android:layout\_below="@+id/editText2" android:layout\_alignRight="@+id/editText2" android:layout\_alignEnd="@+id/editText2"** />

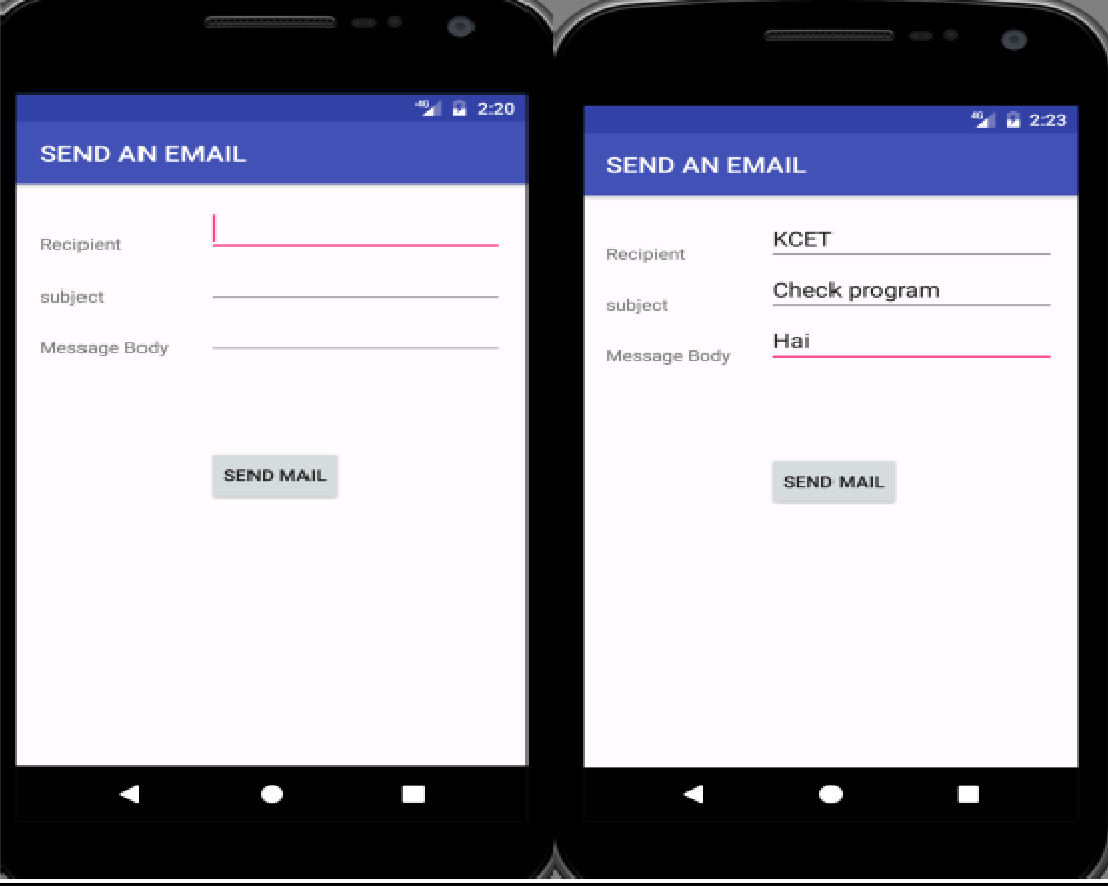
<**Button android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="SEND MAIL" android:id="@+id/sendbttn" android:layout\_centerVertical="true" android:layout\_alignLeft="@+id/editText3" android:layout\_alignStart="@+id/editText3"** />

# <**TextView android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="Recipient" android:id="@+id/textView" android:layout\_alignBottom="@+id/editText" android:layout\_alignParentLeft="true" android:layout\_alignParentStart="true"** />

<**TextView android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="subject" android:id="@+id/textView2" android:layout\_alignBottom="@+id/editText2" android:layout\_alignParentLeft="true" android:layout\_alignParentStart="true"** />

# <**TextView android:layout\_width="wrap\_content" android:layout\_height="wrap\_content" android:text="Message Body" android:id="@+id/textView3" android:layout\_alignBottom="@+id/editText3" android:layout\_alignParentLeft="true" android:layout\_alignParentStart="true"** />

</**RelativeLayout**>

**OUTPUT:**

****

**RESULT:**

Thus, the program for android application to send an email was executed successfully.

| **EX.NO: 11** | **WRITE A MOBILE APPLICATION THAT CREATES ALARM CLOCK IN ANDROID** |
| --- | --- |
| **DATE:** |

**Mini Project 1**

**AIM:**

To write a mobile application that creates alarm clock in android.

**PROCEDURE:**

1. Open eclipse or android studio and select new android project
2. Give project name and select next
3. Choose the android version.Choose the lowest android version(Android 2.2) .
4. Enter the package name.package name must be two word seprated by comma and click finish.
5. Go to package explorer in the left hand side.
6. Go to res folder and select layout.Double click the main.xml file.
7. Now select mainactivity.java file.
8. Right Click package and select new class .
9. Now go to main.xml and right click .select run as option and select run configuration
10. Android output is present in the android emulator.

**SOURCE CODE:**

**Layout.xml**

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

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

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

android:orientation="vertical" >

<TextView

android:id="@+id/Title"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="5px"

android:focusable="false"

android:focusableInTouchMode="false"

android:gravity="center\_vertical|center\_horizontal"

android:text="ALARM CLOCK"

android:textSize="20sp"

android:textStyle="bold" />

<Button

android:id="@+id/startSetDialog"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:text="Set Target Time"/>

<TextView

android:id="@+id/alarmprompt"

android:layout\_width="fill\_parent"

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

</LinearLayout>

package alarm.android

**Activity.java**

import java.util.Calendar;

import java.util.Calendar;

import android.app.Activity;

import android.app.AlarmManager;

import android.app.PendingIntent;

import android.app.TimePickerDialog;

import android.app.TimePickerDialog.OnTimeSetListener;

import android.content.Context;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.TextView;

import android.widget.TimePicker;

public class AlarmActivity extends Activity {

TimePicker myTimePicker;

Button buttonstartSetDialog;

TextView textAlarmPrompt;

TimePickerDialog timePickerDialog;

final static int RQS\_1 = 1;

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

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

textAlarmPrompt = (TextView)findViewById(R.id.alarmprompt);

buttonstartSetDialog = (Button)findViewById(R.id.startSetDialog);

buttonstartSetDialog.setOnClickListener(new OnClickListener(){

@Override

public void onClick(View v) {

textAlarmPrompt.setText("");

openTimePickerDialog(false);

}});

}

private void openTimePickerDialog(boolean is24r){

Calendar calendar = Calendar.getInstance();

timePickerDialog = new TimePickerDialog(

AlarmActivity.this,

onTimeSetListener,

calendar.get(Calendar.HOUR\_OF\_DAY),

calendar.get(Calendar.MINUTE),

is24r);

timePickerDialog.setTitle("Set Alarm Time");

timePickerDialog.show();

}

OnTimeSetListener onTimeSetListener

= new OnTimeSetListener(){

@Override

public void onTimeSet(TimePicker view, int hourOfDay, int minute) {

Calendar calNow = Calendar.getInstance();

Calendar calSet = (Calendar) calNow.clone();

calSet.set(Calendar.HOUR\_OF\_DAY, hourOfDay);

calSet.set(Calendar.MINUTE, minute);

calSet.set(Calendar.SECOND, 0);

calSet.set(Calendar.MILLISECOND, 0);

if(calSet.compareTo(calNow) <= 0){

//Today Set time passed, count to tomorrow

calSet.add(Calendar.DATE, 1);

}

setAlarm(calSet);

}};

private void setAlarm(Calendar targetCal){

textAlarmPrompt.setText(

"\n\n\*\*\*\n"

+ "Alarm is set@ " + targetCal.getTime() + "\n"

+ "\*\*\*\n");

Intent intent = new Intent(getBaseContext(), AlarmReceiver.class);

PendingIntent pendingIntent = PendingIntent.getBroadcast(getBaseContext(), RQS\_1, intent, 0);

AlarmManager alarmManager = (AlarmManager)getSystemService(Context.ALARM\_SERVICE);

alarmManager.set(AlarmManager.RTC\_WAKEUP, targetCal.getTimeInMillis(), pendingIntent);

}

}

package alarm.android;

import android.content.BroadcastReceiver;

import android.content.Context;

import android.content.Intent;

import android.widget.Toast;

public class AlarmReceiver extends BroadcastReceiver {

@Override

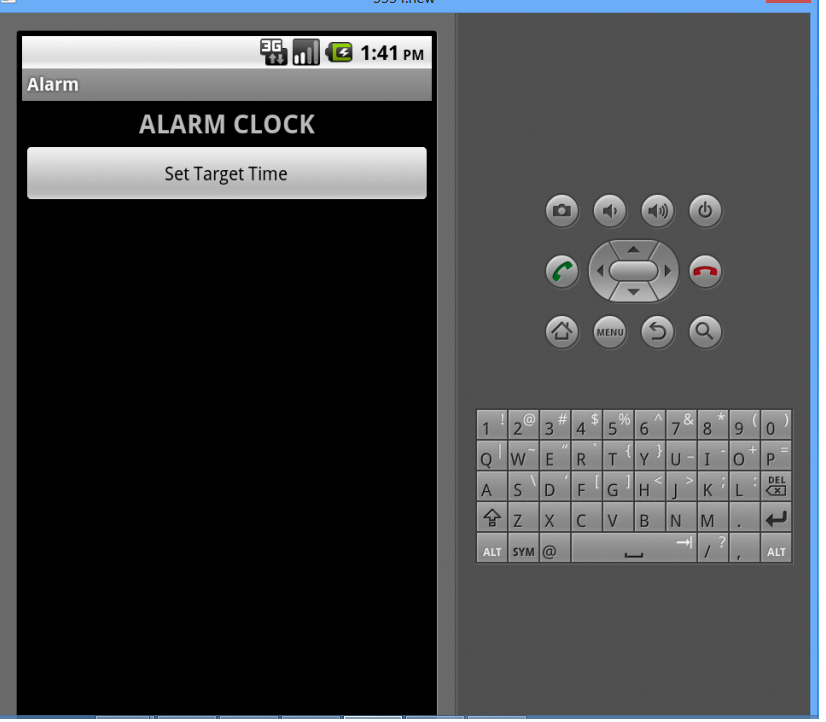
public void onReceive(Context arg0, Intent arg1) {

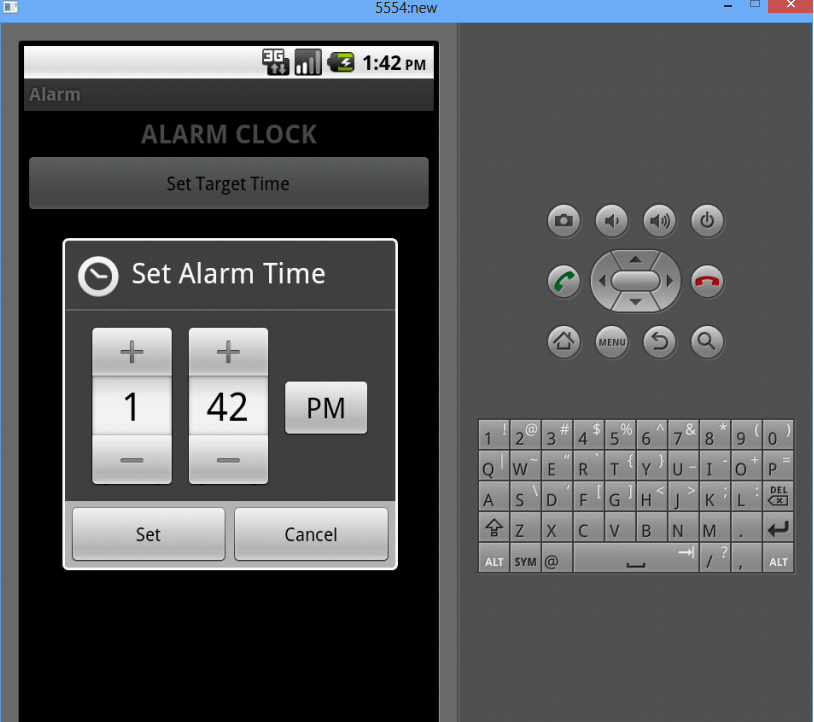
Toast.makeText(arg0, "Alarm received!", Toast.LENGTH\_LONG).show();

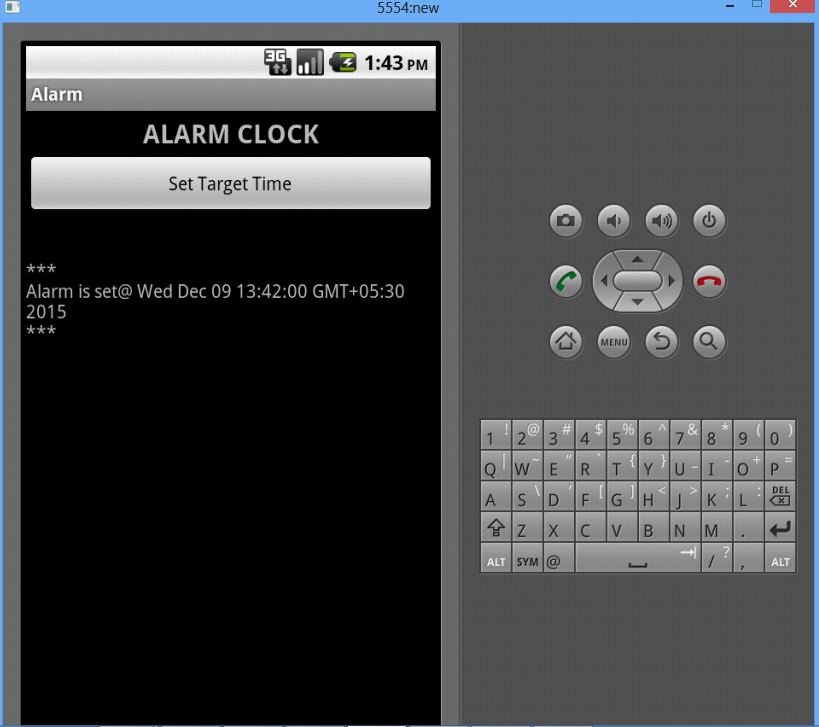
}

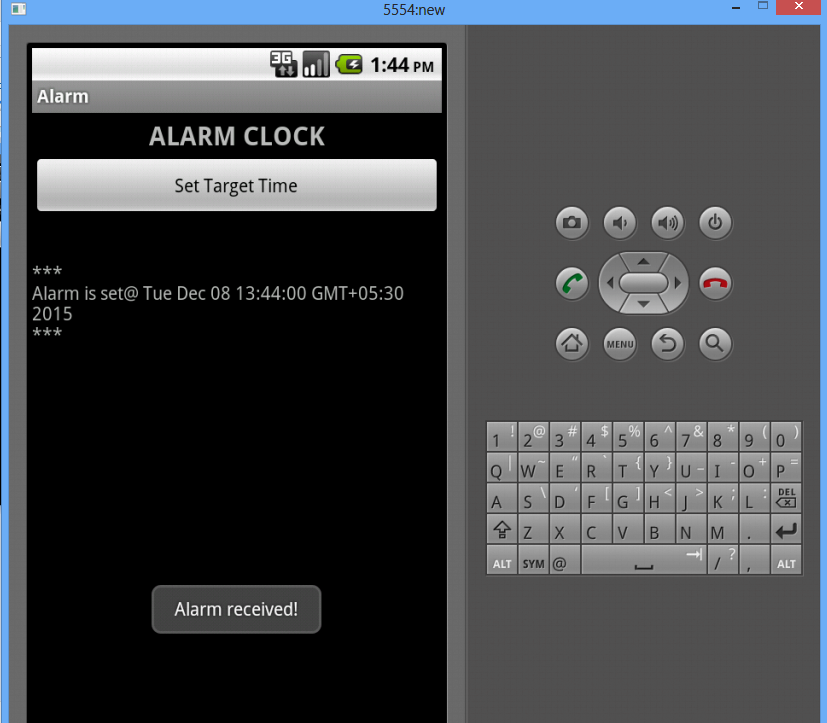
}

**OUTPUT:**









**RESULT:**

Thus the project to write a mobile application that creates alarm clock in android

Was developed using Eclipse Suite Software Successfully.

**OUTCOME:**

Developed an application that creates an alarm clock in mobile application

**EXTRA PROGRAMS**

| **EX:NO:12 CREATE A SLIDESHOW WHICH HAS THREE SLIDES** |  |  |
| --- | --- | --- |

| **AIM:**  To Create a slideshow this has three slides, which includes only text. Program should change to the new slide after 5 seconds. After the third slide program returns to the First Slide  **ALGORITHM:**   * Open eclipse or android studio and select new android project * Give project name and select next * Choose the android version. Choose the lowest android version(Android 2.2) and select next * Enter the package name. Package name must be two word separated by comma and click finish * Go to package explorer in the left hand side. Select our project. * Go to res folder and select layout. Double click the main.xml file * Now you can see the Graphics layout window. * Go to project explorer and select source folder. Now select mainactivity.java file and type the following code   **SOURCE CODE:** |  |  |
| --- | --- | --- |
| importjavax.microedition.midlet.\*; |  |
|  |  |
| importjavax.microedition.lcdui.\*; |  |  |

public class SlideShowxtndsMIDlim l m ntsCommandListener

{

publicSlideShow()Dept

public Form slide1;

public Command Exit; public Displaydisplay;

{

display= isplay.getisplay(this);

Exit=new Command("Exit",Command.EXIT,1); slide1=new Form("Slide1"); slide1.append("This is Slide number 1"); slide1.addCommand(Exit);

slide2=new Form("Slide2");

slide2.append("This is Slide number 2"); slide2.addCommand(Exit);

slide3=new Form("Slide3"); slide3.append("This is Slide number 3");

slide3.addCommand(Exit); slide1.setCommandListener(this); slide2.setCommandListener(this); slide3.setCommandListener(this);

}

public void startApp() {

Thread runner = new Thread(new ThreadRunner(display,slide1,slide2,slide3)); runner.start();

}

public void pauseApp() {

}

public void destroyApp(boolean unconditional) {

}

public void commandAction(Command command,Displayable displayable)

| { | if(displayable==slide1) | | |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |
|  | { | if(command== xit) | |  |
|  |  |  |
|  | } | notifyDestroyed(); | |  |
|  |  |  |  |
|  |  |  |  |  |
| else if(displayable==slide2) | | | |  |  |
|  | { | if(command==Exit) | |  |  |
|  |  |  |  |
|  | } | notifyDestroyed(); | |  |  |
|  |  |  |  |  |
|  | else if(displayable==slide3) | | |  |  |
|  |  |  |  |
|  | { | if(command==Exit) | |  |  |  |
|  |  |  |  |  |
|  | } | notifyDestroy d(); | |  |  |  |
| } |  |  |  |  |  |
|  |  |  |  |  |  |
| } |  |  |  |  |  |  |
| class ThreadRunnerimpl m nts Runnable { | | | |  |  |  |
| Display display; | | |  |  |  |  |
| public int c=0; | | |  |  |  |  |
| public Form slide1; | | | |  |  |  |
| public Form slide2; | | | |  |  |  |
| public Form slide3; | | | |  |  |  |
| public ThreadRunner(Display display,Form slide1,Form slide2,Form slide3) { | | | | | |  |
| this.display = display; | | | |  |  |  |
| this.slide1=slide1; | | | |  |  |  |

this.slide2=slide2;

this.slide3=slide3;

}

public void run() { while(true)

{

c++;

if(c==1)

display.setCurrent(slide1); else if(c==2)

display.setCurrent(slide2); else if(c==3)

display.setCurrent(slide3); else if(c==4)

c=0;

try

{

Thread.sleep(1500);

}

catch(Exception ex)

{

}

}

}

}

**OUTPUT:**



**RESULT:**

Thus the program was created a slideshow that has three slides, using android studio successfully.

**OUTCOME:** Applied Knowledge to develop an modern real time application for a mobile application

**EX:NO:13 WORKING ON DRAWING AND IMAGES**

| **AIM:**  To Create a slideshow consists of three slides, which include pictures at PNG format. Program should change to the new slide after 5 seconds | | | |  |  |
| --- | --- | --- | --- | --- | --- |
| **ALGORITHM:**   * Open eclipse or android studio and select new android project * Give project name and select next * Choose the android version. Choose the lowest android version(Android 2.2) and select next * Enter the package name. Package name must be two word separated by comma and click finish * Go to package explorer in the left hand side. Select our project. * Go to res folder and select layout. Double click the main.xml file * Now you can see the Graphics layout window. * Go to project explorer and select source folder. Now select mainactivity.java file and type the following code | | | | |  |
|  | | | |  |  |
| **SOURCE CODE:** |  |  | | | |
|  |  | | | |
| importjavax.microedition.midlet.\*; |  |  | | | |

public class imageSlideShowxtndsMIDlimlementsCommandListener { public Form slide1;

public Form slide2; public Form slide3; public Command Exit; public Display display; public Image image1; public Image image2; public Image image3;

publicImageItem imageitem1;

publicImageItem imageitem2;

publicImageItem imageitem3;

publicimageSlideShow()

{

display=Display.getDisplay(this); try

{

image1=Image.createImage("/1.png");

image2=Image.createImage("/2.png");

image3=Image.createImage("/3.png");

imageitem1=new ImageItem(null,image1,ImageItem.LAYOUT\_CENTER,"image1");

imageitem2=new ImageItem(null,image2,ImageItem.LAYOUT\_CENTER,"image2"); imageitem3=new ImageItem(null,image3,ImageItem.LAYOUT\_CENTER,"image3");

|  | } |  |  |  |
| --- | --- | --- | --- | --- |
|  | catch(Exception ex) |  |  |  |
|  | { |  |  |  |
|  | } |  |  |  |
|  | Exit=new Command("Exit",Command.EXIT,1); |  |  |
|  | slide1=new Form("Slide1"); |  |  |
|  | slide1.append(imageitem1); |  |  |
|  | slide2=new Form("Slide2"); |  |  |
|  | slide2.append(imageitem2); |  |  |
|  | slide2.addCommand(Exit); |  |  |
|  | slide3=new Form("Slide3"); |  |  |
|  | slide3.append(imageitem3); |  |  |
|  | slide1.setCommandListener(this);  slide2.setCommandListener(this);   | } | slide3.setCommandListener(this); |  |  |  | | --- | --- | --- | --- | --- | |  |  |  |  | | public void startApp() { | |  |  |  | |  | Thread runner = new Thread(new ThreadRunner(dis lay,slide1,slide2,slide3)); | | |  | |  |  |  |
|  |  |  |  |  |
|  | |  | runner.start(); |  |  |  | | --- | --- | --- | --- | --- | |  |  |  |  |   } |  |  |  |
|  |  |  |  |
|  |  |  |  |  |

public void pauseApp() {

}

public void destroyApp(bool an unconditional) {

}

| public void commandAction(Command command,Displayable displayable) | |  |
| --- | --- | --- |
| { | |  |
| if(displayable==slide1) | |  |
| { | if(command==Exit) |  |
|  |  |
| } | notify estroyed(); |  |
|  |  |
| else if(displayable==slide2) | |  |
| { |  |  |
|  |  |

if(command==Exit)

notifyDestroyed();

}

else if(displayable==slide3)

{

if(command==Exit)

notifyDestroyed();

}}}

classThreadRunner implements Runnable { Display display;

publicint c=0; public Form slide1; public Form slide2;

public Form slide3;

publicThreadRunner(Display display, Form slide1,Form slide2,Form slide3)

{

this.display = display;

this.slide1=slide1;

this.slide2=slide2;

this.slide3=slide3;

}

public void run() { while(true)

{

c++;

if(c==1)

display.setCurrent(slide1);

else if(c==2)

display.setCurrent(slide2); else if(c==3)

display.setCurrent(slide3); else if(c==4)

c=0; try

{

Thread.sleep(1500);

}

catch(Exception ex)

{

}

}

| **}**  **}** |  |  |
| --- | --- | --- |

| **OUTPUT:** |  | | |  |
| --- | --- | --- | --- | --- |
|  | |  | | |

**RESULT:**

Thus the program was created a new Image in the slide show, using android studio successfully.

**OUTCOME:**

Understand the capability of creating new image using slide show for a mobile application.