VIDYA JYOTHI INSTITUTE OF TECHNOLOGY

(Autonomous)

Aziz Nagar Gate, C.B. Post, Hyderabad-75

Department of Information Technology

Year : IV B.Tech **Semester : I**

MOBILE APPLICATION DEVELOPMENT

(R19)

LAB MANUAL

Faculty Coordinator : G Indira Priyadarshini **HOD**

List of Experiments

Week 1	Installation and configuration of Android Studio	3-14
	Develop an application that uses GUI components to display a "Hello World" message and change its color and font size.	
Week 2	Develop an application that receives user's name, contact and city and displays the same using Layout Managers and Event Listeners.	15-19
Week 3	Create a native calculator application.	20-23
Week 4	Develop a Registration and Login application that makes use of database.	24-30
Week 5	Develop a native application that uses GPS location information.	31-35
Week 6	Develop an application that creates notification upon receiving a message.	36-38
Week 7	Create an alarm clock mobile application.	39-43

Week 1

- **Installation and configuration of Android Studio**

Aim: Installation and configuration of Android Studio

Step 1 - System Requirements

The required tools to develop Android applications are open source and can be downloaded from the Web. Following is the list of software's you will need before you start your Android application programming.
Java JDK5 or later version
Java Runtime Environment (JRE)6 AndroidStudio

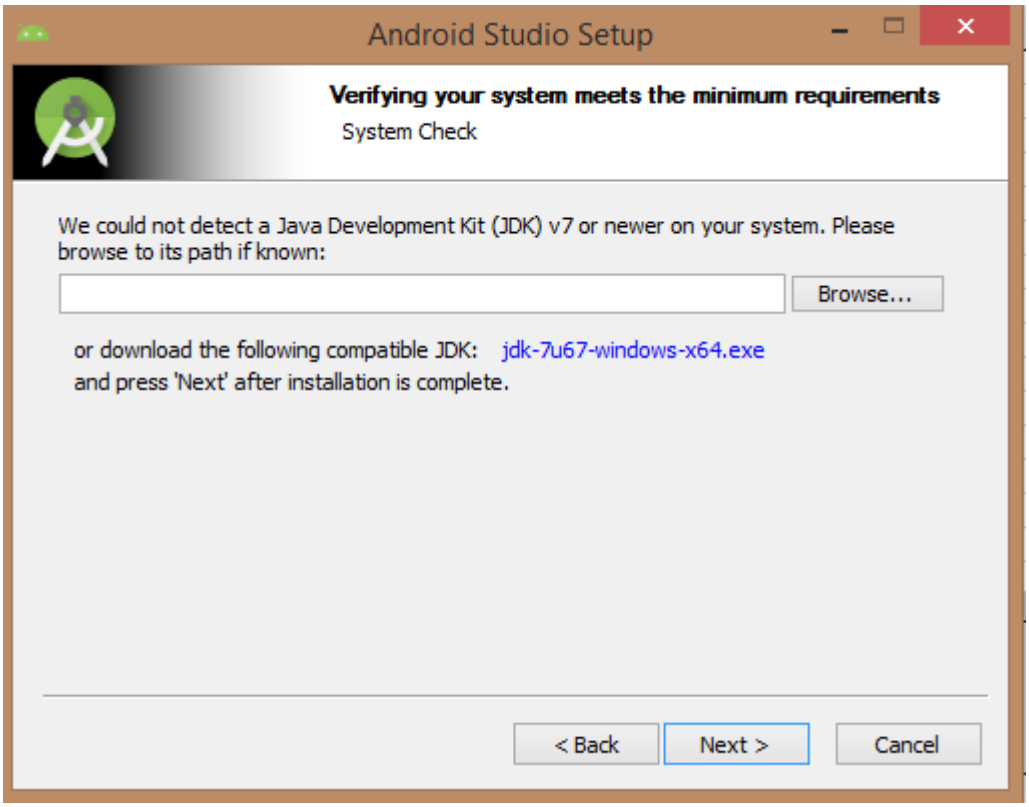
Step 2 - Setup Android Studio

Android Studio is the official IDE for android application development.It works based on IntelliJ IDEA, You can download the latest version of android studio from Android Studio 2.2 Download, If you are new to installing Android Studio on windows,you will find a file, which is named as android-studio-bundle-143.3101438-windows.exe.So just download and run on windows machine according to android studio wizardguideline.

If you are installing Android Studio on Mac or Linux, You can download the latest version from Android Studio Mac Download,or Android Studio Linux Download, check the instructions provided along with the downloaded file for Mac OS and Linux. This tutorial will consider that you are going to setup your environment on Windows machine having Windows 8.1 operating system.
Installation

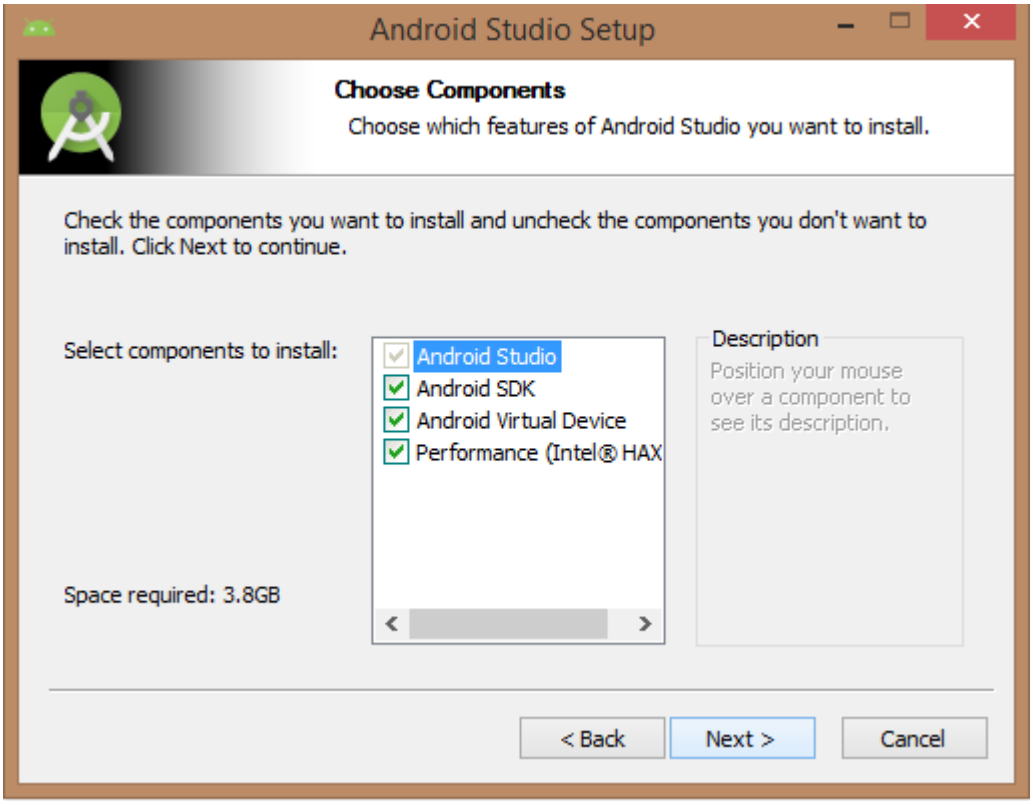
So let's launch Android Studio.exe,Make sure before launch Android Studio, Our Machine should required installed Java JDK. To install Java JDK,take a references of Android environment setup

Once you launched Android Studio, its time to mention JDK7 path or later version in android studio installer.

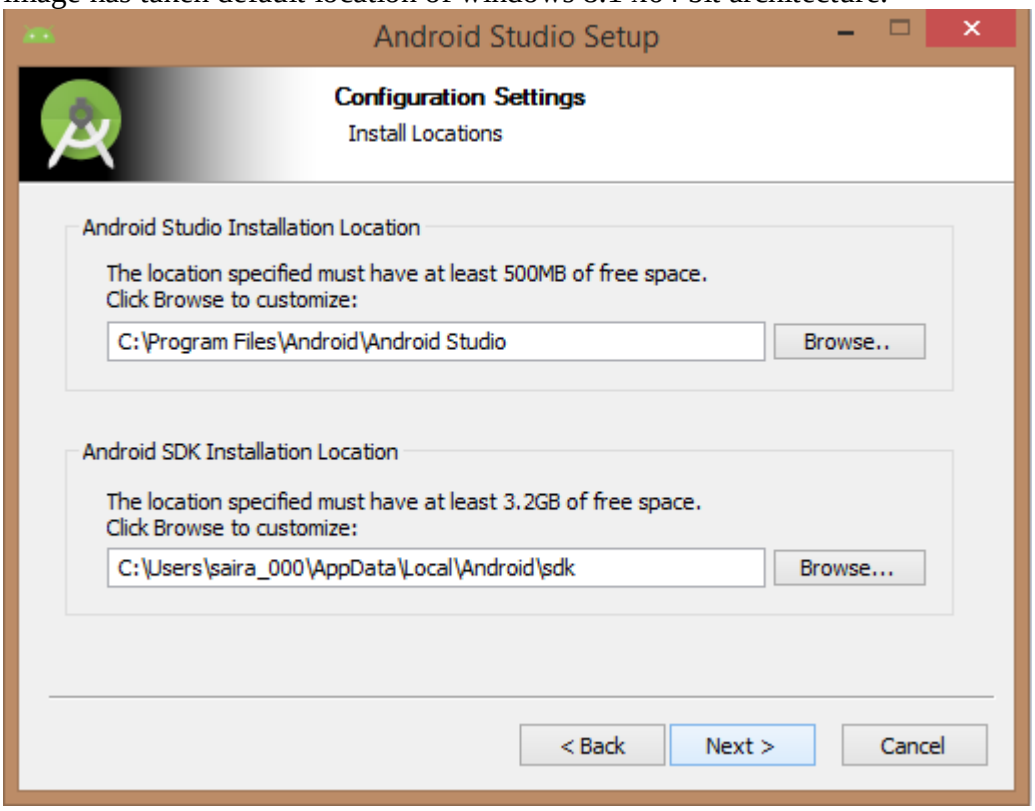


Below is the image initiating JDK to android SDK

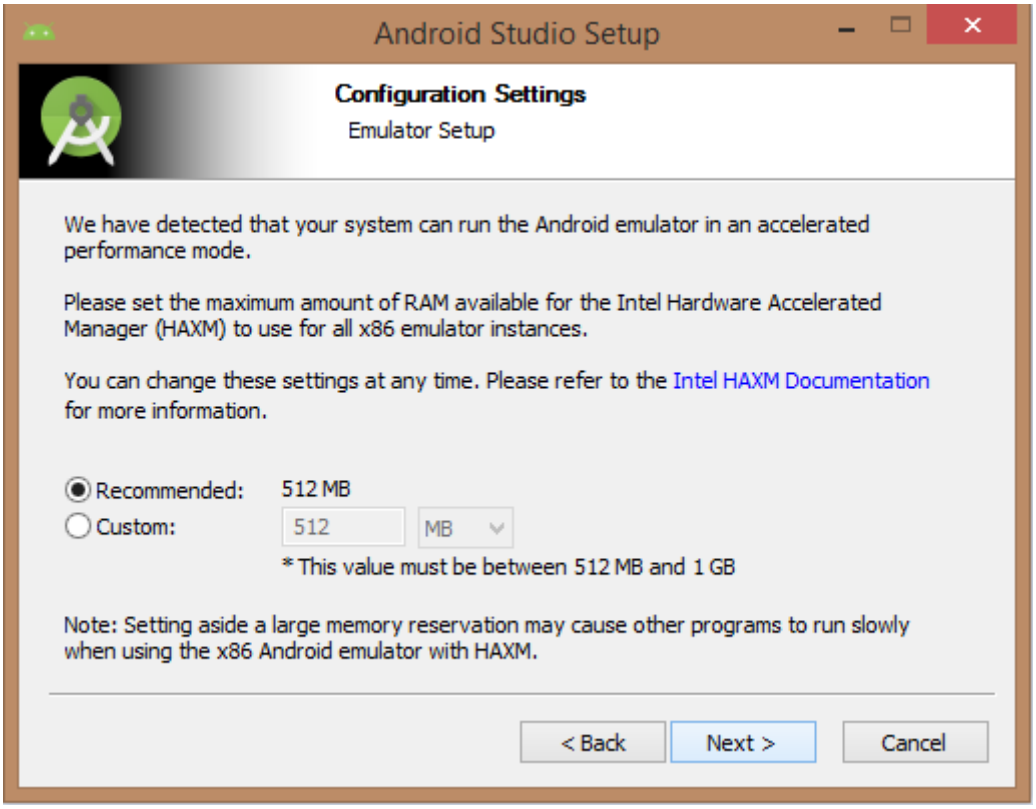
Need to check the components, which are required to create applications, below the image has selected Android Studio, Android SDK, Android Virtual Machine and performance(Intel chip).



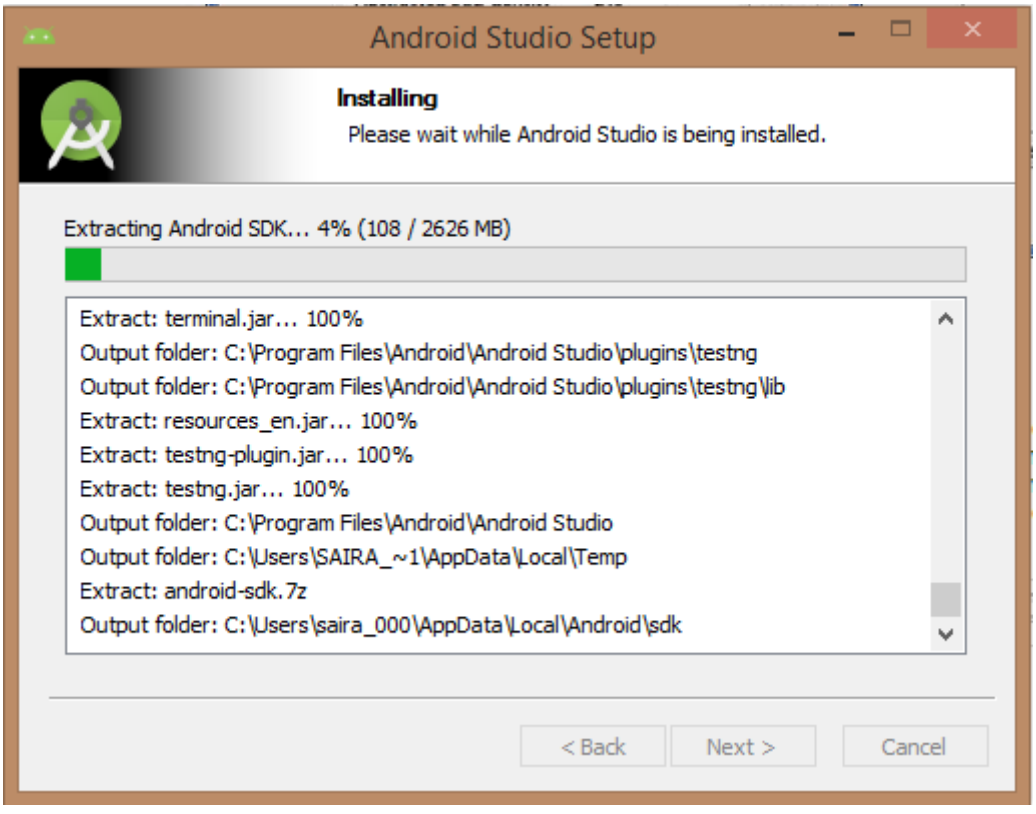
Need to specify the location of local machine path for Android studio and Android SDK, below the image has taken default location of windows 8.1 x64 bit architecture.



Need to specify the ram space for Android emulator by default it would take 512MB of local machine RAM.

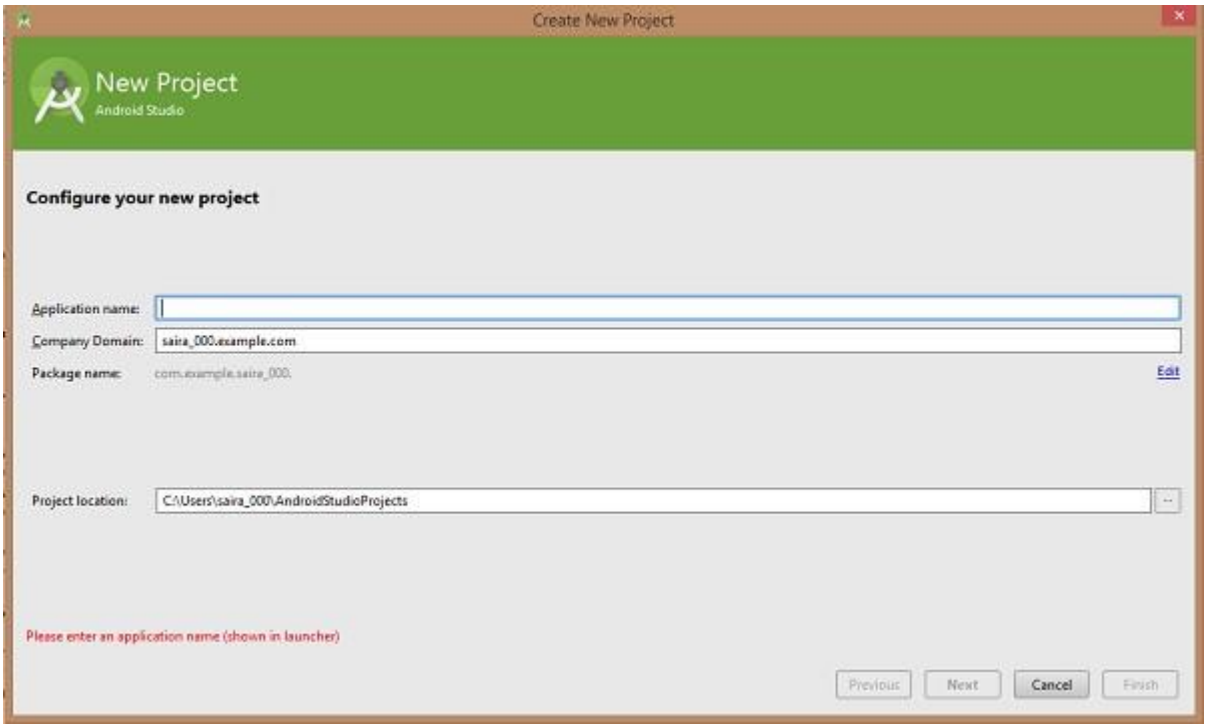


At final stage, it would extract SDK packages into our local machine, it would take a while time to finish the task and would take 2626MB of Hard disk space.

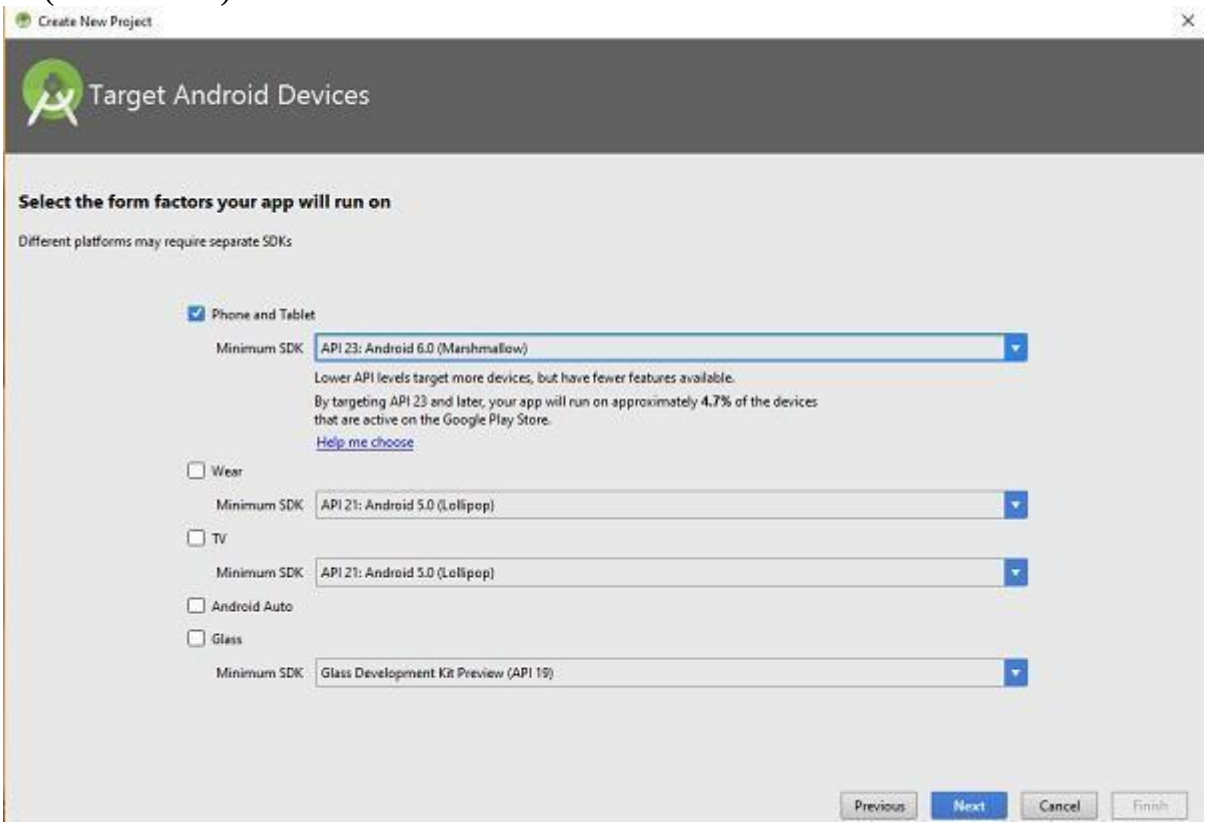


After done all above steps perfectly, you must get finish button and it gonna be open android studio project with Welcome to android studio message as shown below

You can start your application development by calling start a new android studio project. in a new installation frame should ask Application name, package information and location of the project.

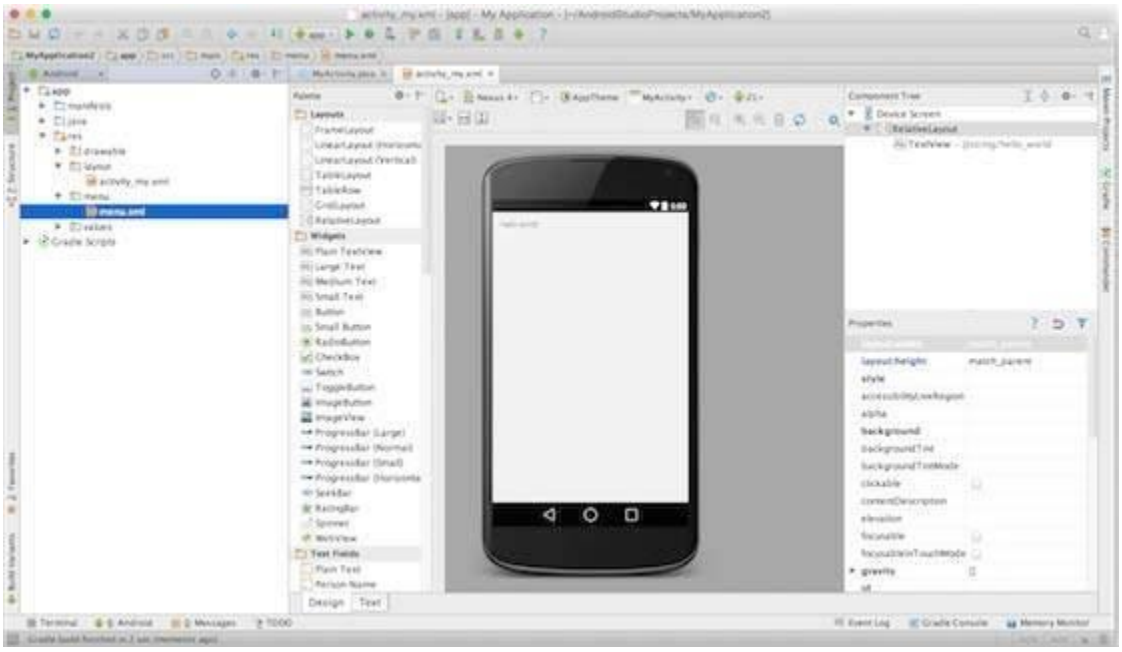


After entered application name, it going to be called select the form factors your application runs on, here need to specify Minimum SDK, in our tutorial, I have declared as API23: Android 6.0(Mashmallow)



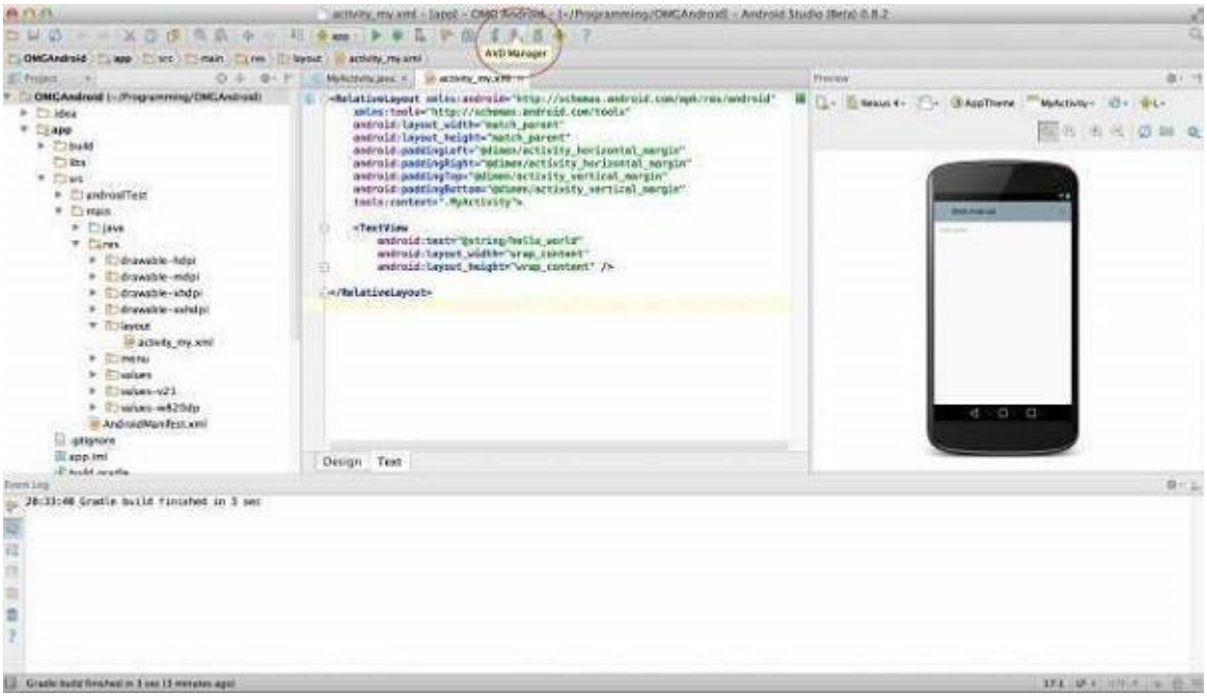
The next level of installation should contain selecting the activity to mobile, it specifies the default layout for Applications

At the final stage it going to be open development tool to write the application code.



Step 3 - Create Android Virtual Device

To test your Android applications, you will need a virtual Android device. So before we start writing our code, let us create an Android virtual device. Launch Android AVD Manager Clicking AVD_Manager icon as shown below



After Click on a virtual device icon, it going to be shown by default virtual devices which are present on your SDK, or else need to create a virtual device by clicking Create new Virtual device button

If your AVD is created successfully it means your environment is ready for Android application development. If you like, you can close this window using top-right cross button. Better you re- start your machine and once you are done with this last step, you are ready to proceed for your first Android example but before that we will see few more important concepts related to Android Application Development.

- **Aim: Develop an application that uses GUI components to display a "Hello World" message and change its color and font size.**

Procedure:

1) Open eclipse or android studio and select new android project
2) Give project name and select next
3) Choose the android version.Choose the lowest android version(Android 2.2)and selectnext
4) Enter the package name.package name must be two word seprated bycomma and clickfinish
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.xmlfile
7) Now you can see the Graphics layoutwindow.
8) Click the main.xml file and type the codebelow

```
<?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/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>
```

9) Again click the graphics layout tab and screen layout is look likebelow

10) Go to project explorer and select src folder.Now select mainactivity.java fileand typethe following code

PROGRAM

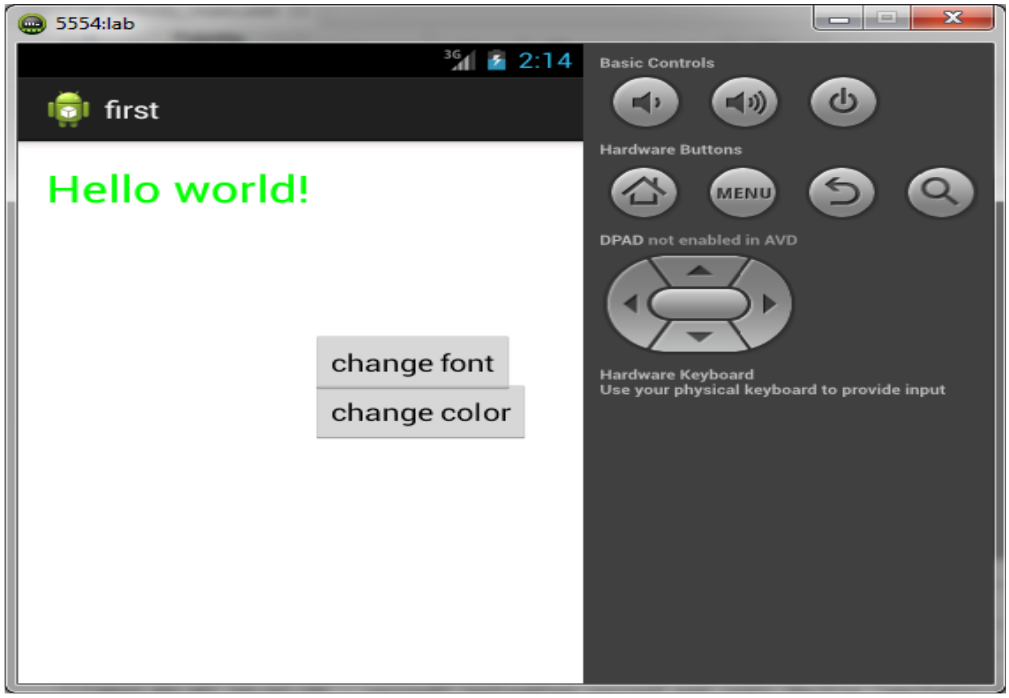
```
//import android.R;
import android.app.Activity;
import android.graphics.Color;
import android.graphics.Typeface;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.TextView;
public class AndroidActivity extends Activity { float font =24;
int i=1; @Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.main);
final TextView 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
;
}
});
```

}
}

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

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

Output:



Week 2

Aim: Develop an application that receives user's name, contact and city and displays the same using 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) and selectnext
4) Enter the package name.package name must be two word seprated bycomma and clickfinish
5) Go to package explorer in the left hand side.select ourproject.
6) Go to res folder and select layout.Double click the main.xml file.Add the code below

XML FILE:

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent" android:layout_height="match_parent"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/TV"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentTop="true"
        android:layout_alignParentEnd="true"
        android:layout_centerHorizontal="true"
        android:layout_marginStart="103dp"
        android:layout_marginTop="85dp"
        android:layout_marginEnd="98dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:minHeight="48dp"
        android:text="Name"
        tools:visibility="visible" />

    <EditText
```

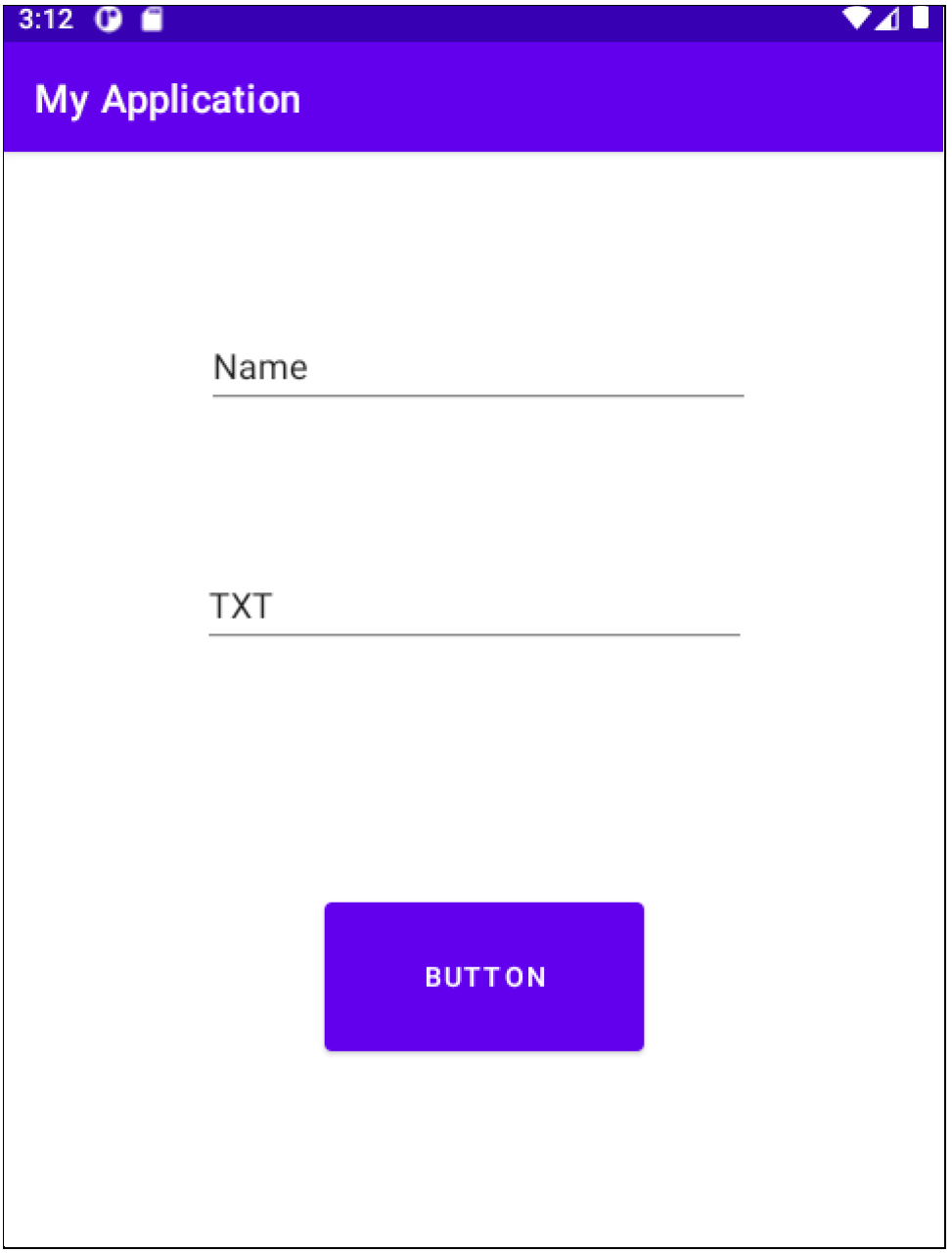
```
        android:id="@+id/N"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/TV"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true"
        android:layout_marginStart="101dp"
        android:layout_marginTop="74dp"
        android:layout_marginEnd="100dp"
        android:ems="10"
        android:inputType="textPersonName"
        android:minHeight="48dp"
        android:text="TXT"

tools:visibility="visible"
/>

    <Button
        android:id="@+id/B1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_below="@+id/N"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginStart="164dp"
        android:layout_marginTop="122dp"
        android:layout_marginEnd="153dp"
        android:layout_marginBottom="309dp"
        android:text="Button" />

    <EditText
        android:id="@+id/OUT"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentStart="true"
        android:layout_alignParentEnd="true"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginStart="105dp"
        android:layout_marginEnd="96dp"
        android:layout_marginBottom="187dp"
        android:ems="10"
        android:inputType="textPersonName"
```

```
        android:text="Name"
        tools:visibility="invisible" />

 </RelativeLayout>
```

JAVA:

```
 package com.hello;
 import android.graphics.Color;
 import androidx.appcompat.app.AppCompatActivity;
 import android.widget.*;
 import android.view.View;
 import android.os.Bundle;

 public class MainActivity extends AppCompatActivity {
    EditText name, contact;
    TextView out;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
       name = findViewById(R.id.TV);
       contact = (EditText)findViewById(R.id.N);
        Button B =(Button) findViewById(R.id.B1) ;
        out= findViewById(R.id.OUT);
        B.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v)
        {
            if(!(name.getText().toString().isEmpty()||contact.getText().toString().isEmpty()))
{
        out.setText("hello"+name.getText().toString()+""+contact.getText().toString());
              }
        }});
    }}
```

OUTPUT:



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

4) Android output is present in the android emulator as shown inbelow.

Week 3

Aim: Create a native calculator application.

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)and selectnext

4) Enter the package name.package name must be two word seprated bycomma and clickfinish

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

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

Main.xml coding

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
android:orientation="vertical" android:layout_width="fill_parent"
android:layout_height="fill_parent">
<LinearLayout android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/linearLayout1"
android:layout_marginLeft="10pt"
android:layout_marginRight="10pt"
android:layout_marginTop="3pt">
<EditText android:layout_weight="1"
 android:layout_height="wrap_content" android:layout_marginRight="5pt" android:id="@+id/etNum1"
 android:layout_width="match_parent" android:inputType="numberDecimal">
</EditText>
<EditText android:layout_height="wrap_content"
android:layout_weight="1"
android:layout_marginLeft="5pt"
android:id="@+id/etNum2"
android:layout_width="match_parent"
android:inputType="numberDecimal">
</EditText>
</LinearLayout>
<LinearLayout android:layout_width="match_parent"
android:layout_height="wrap_content"
android:id="@+id/linearLayout2"
android:layout_marginTop="3pt"
android:layout_marginLeft="5pt"
android:layout_marginRight="5pt">
<Button android:layout_height="wrap_content"
```

```
android:layout_width="match_parent"
android:layout_weight="1" android:text="+"
android:textSize="15pt" android:id="@+id/btnAdd">
</Button>
<Button android:layout_height="wrap_content"
android:layout_width="match_parent"
android:layout_weight="1" android:text="-"
android:textSize="15pt"
android:id="@+id/btnSub">
</Button>
<Button android:layout_height="wrap_content"
android:layout_width="match_parent"
android:layout_weight="1" android:text="*"
android:textSize="15pt"
android:id="@+id/btnMult">
</Button>
<Button android:layout_height="wrap_content"
android:layout_width="match_parent"
android:layout_weight="1" android:text="/"
android:textSize="15pt" android:id="@+id/btnDiv">
</Button>
</LinearLayout>
<TextView android:layout_height="wrap_content"
android:layout_width="match_parent"
android:layout_marginLeft="5pt"
android:layout_marginRight="5pt"
android:textSize="12pt"
android:layout_marginTop="3pt"
android:id="@+id/tvResult"
android:gravity="center_horizontal">

</TextView>
</LinearLayout>
```

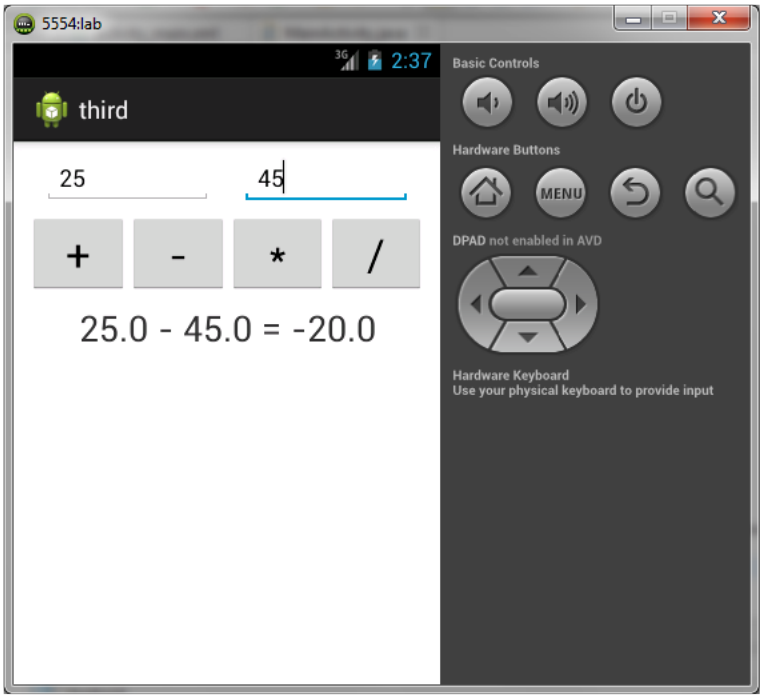
MainActivity.java coding

```
package CALCU.CALU;
import android.app.Activity; import
android.os.Bundle; import
android.text.TextUtils; import
android.view.View;
import android.view.View.OnClickListener; import
android.widget.Button;
import android.widget.EditText; import
```

```
android.widget.TextView;
public class CALCULATORActivity extends Activity implements OnClickListener
{
EditText input1; EditText
input2; Button addition;
Button subtraction; Button
multiplication; Button
division; TextView tvResult;
String oper = "";
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState); setContentView(R.layout.main);
input1 = (EditText) findViewById(R.id.etNum1);
input2 = (EditText) findViewById(R.id.etNum2);
addition = (Button) findViewById(R.id.btnAdd);
subtraction = (Button) findViewById(R.id.btnSub);
 multiplication = (Button) findViewById(R.id.btnMult);
division = (Button) findViewById(R.id.btnDiv);

tvResult = (TextView) findViewById(R.id.tvResult);
// set a listener addition.setOnClickListener(this);
subtraction.setOnClickListener(this);
multiplication.setOnClickListener(this);
division.setOnClickListener(this);
}
@Override
public void onClick(View v) {
float num1 =0;
float num2 = 0;
float result =0;
// check if the fields are empty
if (TextUtils.isEmpty(input1.getText().toString())
|| TextUtils.isEmpty(input2.getText().toString()))
{ return;
}
// read EditText and fill variables with numbers num1
=Float.parseFloat(input1.getText().toString()); num2
=Float.parseFloat(input2.getText().toString());
// defines the button that has been clicked and performs the corresponding operation
// write operation into oper, we will use it later for output switch (v.getId()) {
case R.id.btnAdd: oper =
"+";
```

```
result = num1 + num2;
break;
case R.id.btnSub:
oper = "-";
result = num1 - num2; break;

case R.id.btnMult:
oper = "*";
result = num1 * num2; break;
case R.id.btnDiv:
oper = "/";
result = num1 / num2; break;
default:
break;
}
// form the output line
tvResult.setText(num1 + " " + oper + " " + num2 + " = " + result);
}
}
```

OUTPUT:



Week 4

Aim:Develop a Registration and Login application that makes use of database.

Procedure:

Week 6: Develop a Registration and Login application that makes use of database.

Download the DB Browser.

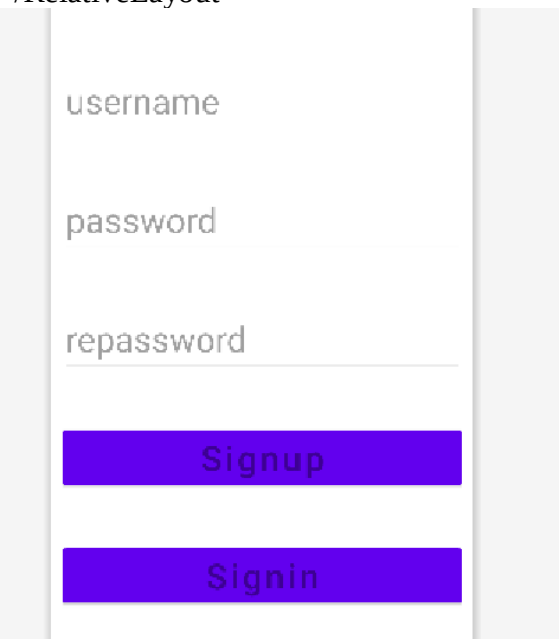
DB.Browser.for.SQLite-3.12.2-win64

Note: In this code Registration form contains name ,password, repassword fields only First practice this code then add email, phone number ,gender to this layout file and functionality with validation for email and phonenumber.

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="10dp"
    tools:context=".MainActivity">

    <EditText
        android:id="@+id/username"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="50dp"
        android:hint="username"
        android:textSize="34sp"
        />

    <EditText
        android:id="@+id/password"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
```

```
    android:layout_below="@+id/username"
    android:layout_marginTop="50dp"
    android:hint="password"
    android:textSize="34sp" />
  <EditText
    android:id="@+id/repassword"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="repassword"
    android:layout_marginTop="50dp"
    android:layout_below="@+id/password"
    android:textSize="34sp"
    />
  <Button
    android:id="@+id/signup"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Signup"
    android:layout_marginTop="50dp"
    android:layout_below="@+id/repassword"
    android:textSize="34sp"
    />
  <Button
    android:id="@+id/signin"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="Signin"
    android:layout_marginTop="50dp"
    android:textSize="34sp"
    android:layout_below="@+id/signup"
    />

</RelativeLayout>
```



MainActivity.java

```
package com.example.sqldbex;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.database.sqlite.SQLiteDatabase;
import android.os.Bundle;
import android.os.strictmode.SqliteObjectLeakedViolation;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    EditText username,password,repass;
    Button signin,signup;
    MyHelper helper;
    Boolean checkUsername;
    Boolean insert;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        username=(EditText) findViewById(R.id.username);
        password=(EditText) findViewById(R.id.password);
        repass=(EditText) findViewById(R.id.repassword);
        signin=(Button)findViewById(R.id.signin);
        signup=(Button)findViewById(R.id.signup);
        helper=new MyHelper(this);
        signup.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String user = username.getText().toString();
            String pass = password.getText().toString();
            String rpwd = repass.getText().toString();
            if (user.equals("") || pass.equals("") || rpwd.equals("")) {
                Toast.makeText(MainActivity.this, "please enter all the fields", Toast.LENGTH_SHORT).show();
            } else {
                if (pass.equals(rpwd)) {
                    checkUsername = helper.checkusername(user);
                    if (checkUsername == false) {
                        insert = helper.insertData(user, pass);
                        if (insert == true) {
                            Toast.makeText(MainActivity.this, "Registered Successfully", Toast.LENGTH_SHORT).show();
                            Intent intent = new Intent(getApplicationContext(), LoginActivity.class);
                            startActivity(intent);
                        } else
                            Toast.makeText(MainActivity.this, "Registered failed", Toast.LENGTH_SHORT).show();
```

```
                } else
                    Toast.makeText(MainActivity.this, "User Already exists", Toast.LENGTH_SHORT).show();
            } else
                Toast.makeText(MainActivity.this, "password not matching ", Toast.LENGTH_SHORT).show();
        }
    }
});
signin.setOnClickListener(new View.OnClickListener() {
@Override
public void onClick(View v) {
    Intent intent = new Intent(MainActivity.this, LoginActivity.class);
    startActivity(intent);
}
});
}
}
```

MyHelper.java

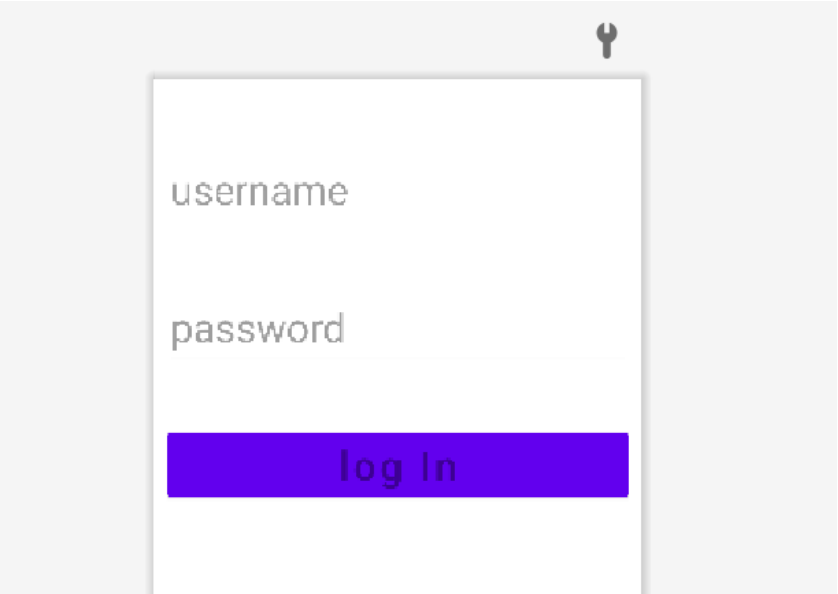
```
package com.example.sqldbex;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;

public class MyHelper extends SQLiteOpenHelper {
    private static final String dbname="mydb.db";
    private static final int version=1;
    public MyHelper(Context context)
    {
        super(context,dbname,null,version);
    }

    @Override
    public void onCreate(SQLiteDatabase db) {
    String sql="CREATE TABLE users (username TEXT, password TEXT)";
    db.execSQL(sql);
    //insert

    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    db.execSQL("drop table if exists users");
    }

    public Boolean insertData(String username, String password)
```

```
    {
        SQLiteDatabase db=this.getWritableDatabase();
        ContentValues contentValues=new ContentValues();
        contentValues.put("username",username);
        contentValues.put("password",password);
        long result=db.insert("users",null,contentValues);
        if(result==-1)
            return false;
        else
            return true;
    }
    public boolean checkusername(String username){
        SQLiteDatabase db=this.getWritableDatabase();
        Cursor cursor=db.rawQuery("select * from users where username=?",new String[]{username});
        if(cursor.getCount()>0)
            return true;
        else
            return false;
    }
    public boolean checkUsernamePass(String username,String password){
        SQLiteDatabase Mydb=this.getWritableDatabase();
        Cursor cursor=Mydb.rawQuery("select * from users where username=? and password =?",new
String[]{username,password});
        if(cursor.getCount()>0)
            return true;
        else
            return false;

    }
}
```

activity_login.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="10dp"
    tools:context=".LoginActivity">

    <EditText
        android:id="@+id/username1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="username"
        android:layout_marginTop="50dp"
        android:textSize="34sp"
        />
```

```
    <EditText
        android:id="@+id/password1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/username1"
        android:layout_marginTop="50dp"
        android:hint="password"
        android:textSize="34sp" />

    <Button
        android:id="@+id/login"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/password1"
        android:layout_marginTop="50dp"
        android:hint="log In"
        android:textSize="34sp" />

            <TextView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:id="@+id/text"
    android:textSize="34dp"
    android:textColor="@color/purple_700"
    android:layout_below="@+id/login"
    />

</RelativeLayout>
```



LoginActivity.java

```
package com.example.sqldbex;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class LoginActivity extends AppCompatActivity {
   EditText username,password;
   Button login;
   MyHelper helper;
   Boolean checkUsernamePass;
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_login);
   username=(EditText) findViewById(R.id.username1);
   password=(EditText) findViewById(R.id.password1);
   login=(Button)findViewById(R.id.login);
   helper= new MyHelper(this);
   login.setOnClickListener(new View.OnClickListener() {
      @Override
      public void onClick(View v) {
         String user= username.getText().toString();
         String pass=password.getText().toString();
         if(user.equals("")|| pass.equals(""))
         {
            Toast.makeText(LoginActivity.this,"please enter all the fields",Toast.LENGTH_SHORT).show();
         }
         else {
            checkUsernamePass = helper.checkUsernamePass(user, pass);
            if (checkUsernamePass == true) {
               Toast.makeText(LoginActivity.this, "sign in successful", Toast.LENGTH_SHORT).show();
              // Intent intent = new Intent(getApplicationContext(), MainActivity.class);
               //startActivity(intent);
TextView txt=(TextView) findViewById(R.id.text);
txt.setText("welcome"+user);

            } else {
               Toast.makeText(LoginActivity.this, "invalid Credentials", Toast.LENGTH_SHORT).show();
            }
         }

      }
   });
   }
}
```

Week 5

Aim:Develop a native application that uses GPS location information.

Procedure:

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

2)Give project name and select next

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

4) Enter the package name.package name must be two word seprated bycomma and clickfinish

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

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

```
<?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>
```

7) Now select mainactivity.java file and type the following code.In mycoding maniactivityname

is GPSlocationActivity.

```
package gps.location;
//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();

}
}
});
}
}
```

8) Go to src folder and Right Click on your package folder and choose new classand givethe class nams as GPStrace

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

```
packagegps.location;
import android.app.AlertDialog; import
android.app.Service; import
android.content.Context;
import android.content.DialogInterface; import
android.content.Intent;
import android.location.Location;
import android.location.LocationListener;
importandroid.location.LocationManager;
importandroid.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;
booleanisNetworkEnabled=false;
Locationlocation;
double latitude;
```

```
doublelongtitude;
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_PROV IDER);
isNetworkEnabled=locationManager.isProviderEnabled(LocationManager.NETWO RK_PROVI
DER);
if(!isGPSEnabled && !isNetworkEnabled){
}else{ this.canGetLocation=true;
if(isNetworkEnabled){
locationManager.requestLocationUpdates(
LocationManager.NETWORK_PROVIDER, MIN_TIME_BW_UPDATES,
MIN_DISTANCE_CHANGE_FOR_UPDATES,this);
}
if(locationManager!=null){ location=locationManager.getLastKnownLocation(LocationManager.NETWORK_
PROVIDER)
;
if(location !=null){ latitude=location.getLatitude();
longtitude=location.getLongitude();
}
}
}
if(isGPSEnabled){
if(location==null){
locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,MI N_TIME_B
W_UPDATES, MIN_DISTANCE_CHANGE_FOR_UPDATES, this);
if(locationManager!=null){ location=locationManager.getLastKnownLocation(LocationManager.GPS_PROVI DER);
if(location!=null){ latitude=location.getLatitude();
longtitude=location.getLongitude();
}
}
}
}
```

```
}
catch(Exception e)
{
e.printStackTrace();
}
return location;
}
public void stopUsingGPS(){ if(locationManager!=null){
locationManager.removeUpdates(GPStrace.this);
}
}
public double getLatitude(){
if(location!=null){ latitude=location.getLatitude();
}
return latitude;
}
public double getLongtiude(){ if(location!=null){
longtitude=location.getLatitude();
}
return longtitude;
}
public boolean canGetLocation(){ return
this.canGetLocation;
}
public void showSettingAlert(){
AlertDialog.Builder alertDialog=new AlertDialog.Builder(context); alertDialog.setTitle("GPS is settings");
alertDialog.setMessage("GPS is not enabled.Do you want to go to setting menu?");
alertDialog.setPositiveButton("settings", new DialogInterface.OnClickListener() { @Override
public void onClick(DialogInterface dialog,int which){
Intent intent=new Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS); context.startActivity(intent);
}
});
alertDialog.setNegativeButton("cancel", new DialogInterface.OnClickListener() { @Override
public void onClick(DialogInterface dialog, int which) {
// TODO Auto-generated method stub
dialog.cancel();
}
});
alertDialog.show();

}
@Override
public void onLocationChanged(Location location) {
```

```
// TODO Auto-generated method stub
}
@Override
public void onProviderDisabled(String provider) {
// TODO Auto-generated method stub
}
@Override
public void onProviderEnabled(String provider) {
// TODO Auto-generated method stub
}
@Override
public void onStatusChanged(String provider, int status, Bundle extras) {
// TODO Auto-generated method stub
}
@Override
public IBinder onBind(Intent intent) {
// TODO Auto-generated method stub return null;
}
}
```

10) Go to manifest.xml file and add the codebelow

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

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

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

Week 6

Aim:Develop an application that creates notification upon receiving a message.

Procedure:

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

2)Give project name and select next

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

4) Enter the package name.package name must be two word seprated bycomma and clickfinish

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

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

```
<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>
```

7) Now select mainactivity.java file and type the following code.In mycoding maniactivity name is Alert1Activity.

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

9) Android output is present in the android emulator as shown inbelow.

Week 7

Aim:Create an alarm clock mobile application.

AndroidManift

AndroidManifest.xml

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

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.javapapers.androidalarmclock">
<uses-permission android:name="android.permission.WAKE_LOCK" />
<application android:allowBackup="true"
android:icon="@drawable/ic_launcher"
android:label="@string/app_name"
android:theme="@style/AppTheme">
<activity android:name=".AlarmActivity"
android:label="@string/app_name">
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
<service android:name=".AlarmService"
android:enabled="true" />
<receiver android:name=".AlarmReceiver" />
</application></manifest>
```

Android Activity

activity_my.xml

The Android Activity is designed to be simple. We have a TimePickercomponent followed bya ToggleButton. That's it. Choose the time to set the alarm and toggle the switchto on. Thealarm will work.

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
android:layout_height="match_parent" android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
android:paddingBottom="@dimen/activity_vertical_margin" tools:context=".MyActivity">
<TimePicker android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/alarmTimePicker"
```

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

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()
{ returninst;
}
@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);
}
}
```

Alarm Receiver

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;
importandroid.content.ComponentName;
importandroid.content.Context;
```

```
import android.content.Intent;
importandroid.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);
}
}
```

Alarm Notification Message

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.");
}
}
```