

# Android Application that creates Alarm Clock

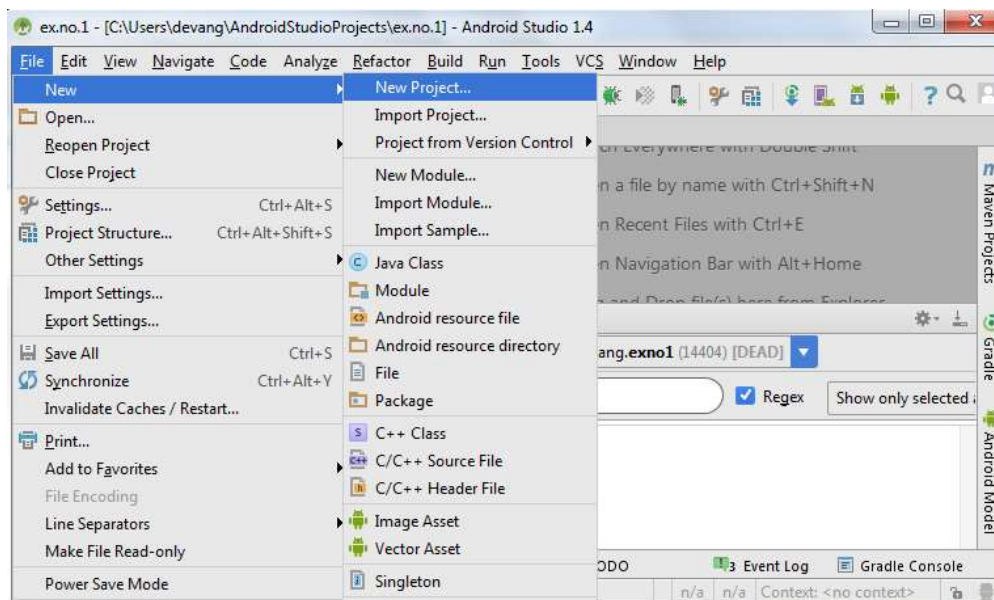
## Aim:

To develop a Android Application that creates Alarm Clock.

## Procedure:

### Creating a New project:

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



- Then type the Application name as “**ex.no.11**” and click **Next**.

Create New Project

## New Project

Android Studio

### Configure your new project

Application name:

Company Domain:

Package name:  [Edit](#)

Project location:

The application name for most apps begins with an uppercase letter.

[Previous](#) [Next](#) [Cancel](#) [Finish](#)

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

Create New Project

## Target Android Devices

### Select the form factors your app will run on

Different platforms may require separate SDKs.

☒ Phone and Tablet  
Minimum SDK:

☐ Wear  
Minimum SDK:

☐ TV  
Minimum SDK:

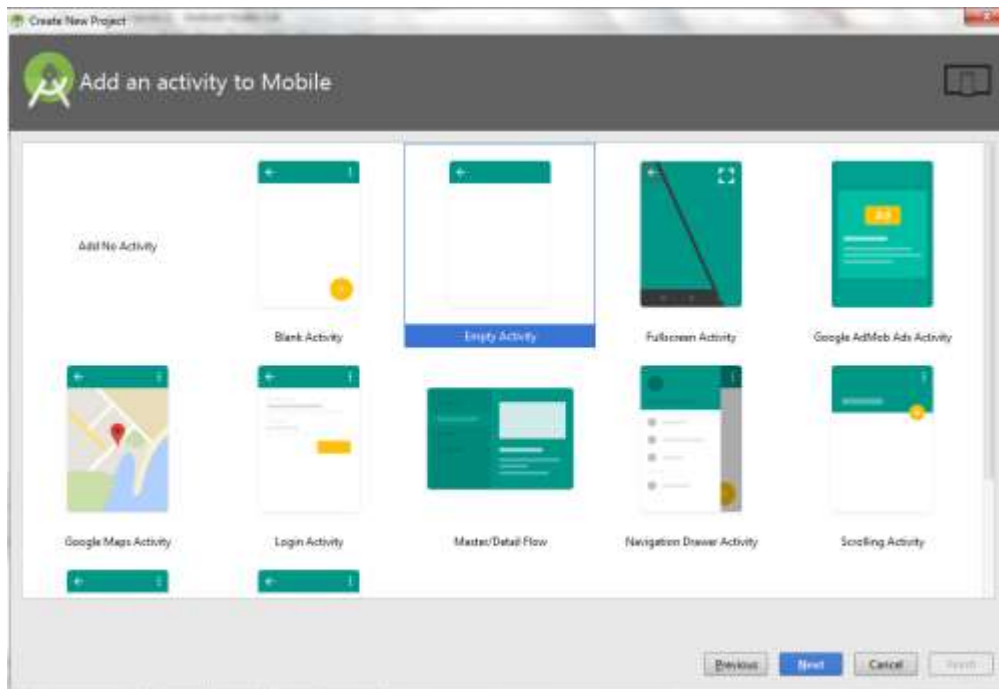
☐ Android Auto

☐ Glass (Not Supported)  
Minimum SDK:

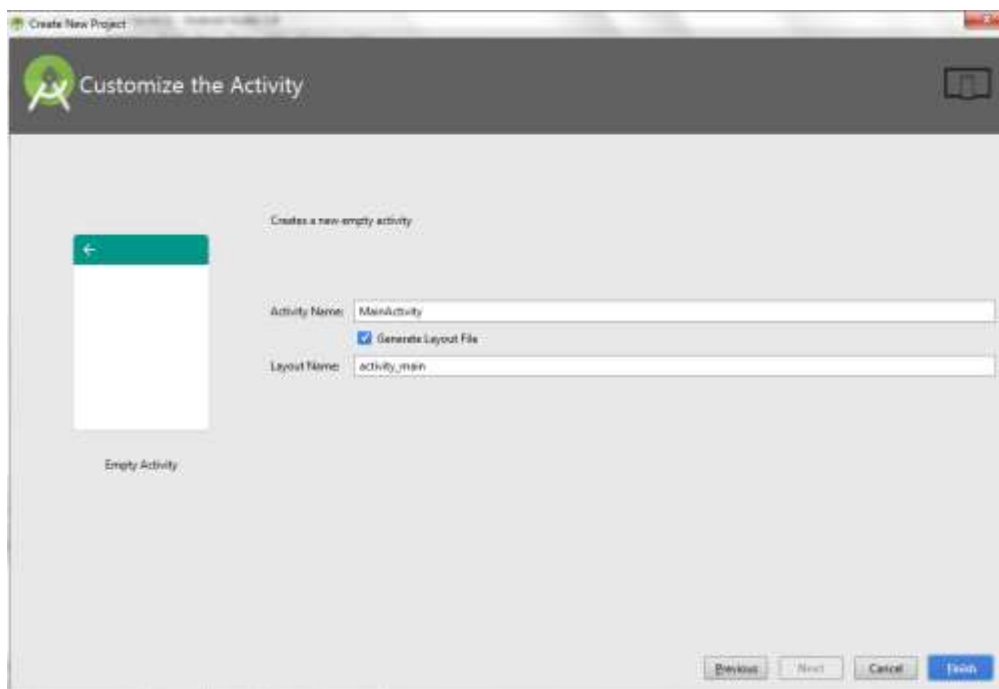
Lower API levels target more devices, but have fewer features available. By targeting API 15 and later, your app will run on approximately 94.0% of the devices that are active on the Google Play Store.  
[Help me choose](#)

[Previous](#) [Next](#) [Cancel](#) [Finish](#)

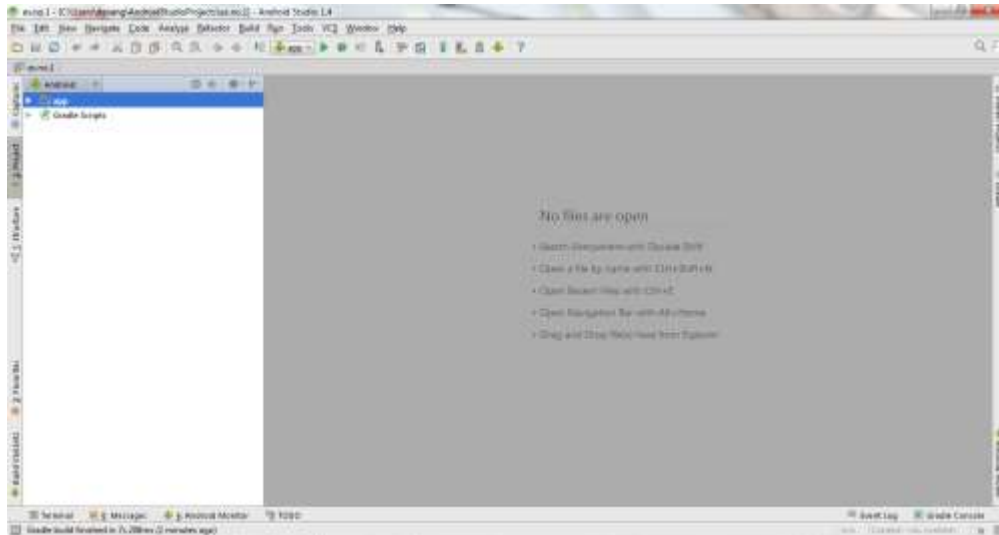
- Then select the **Empty Activity** and click **Next**.



- Finally click **Finish**.

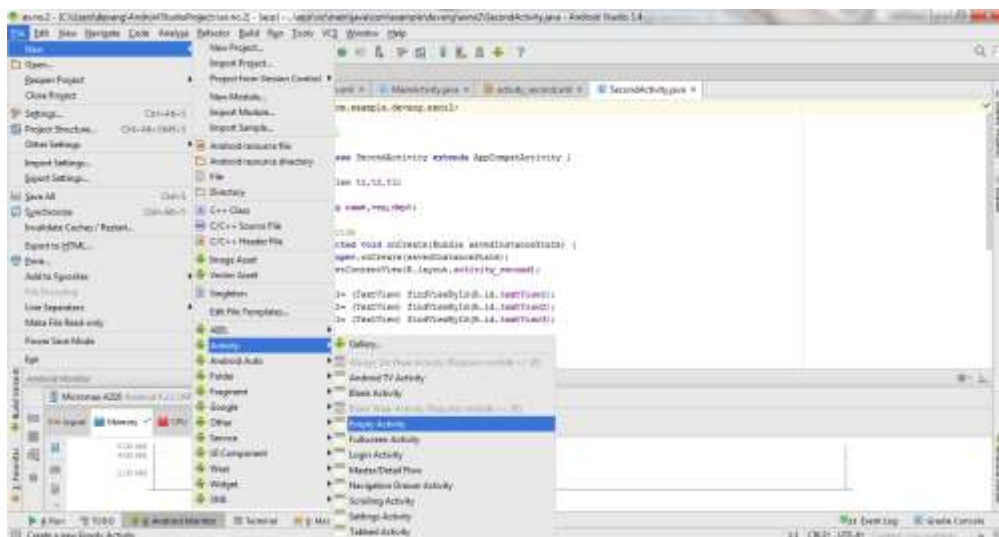


- It will take some time to build and load the project.
- After completion it will look as given below.

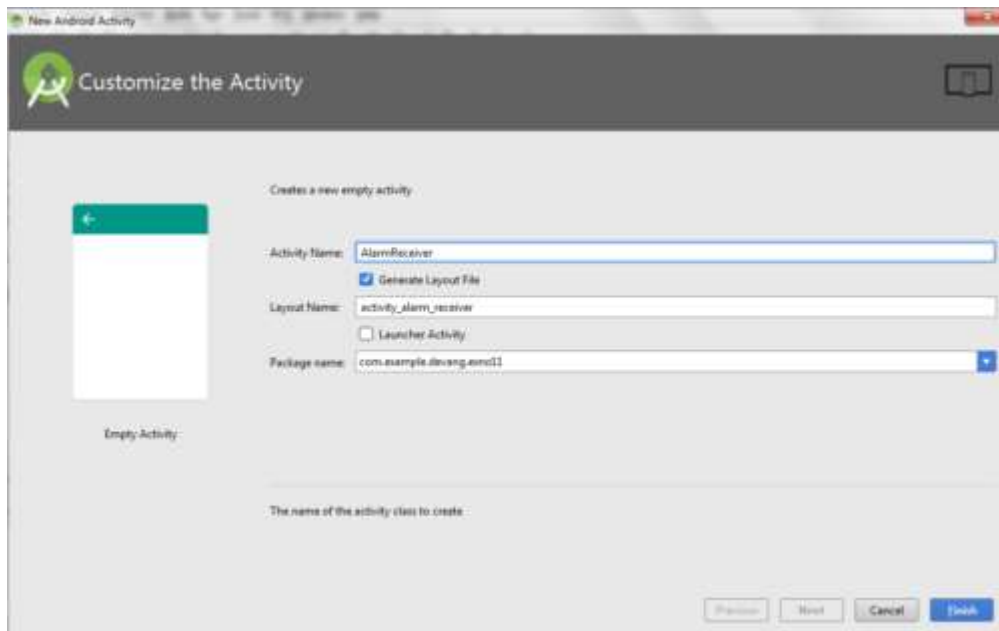


## Creating Second Activity for the Android Application:

- Click on **File -> New -> Activity -> Empty Activity**.



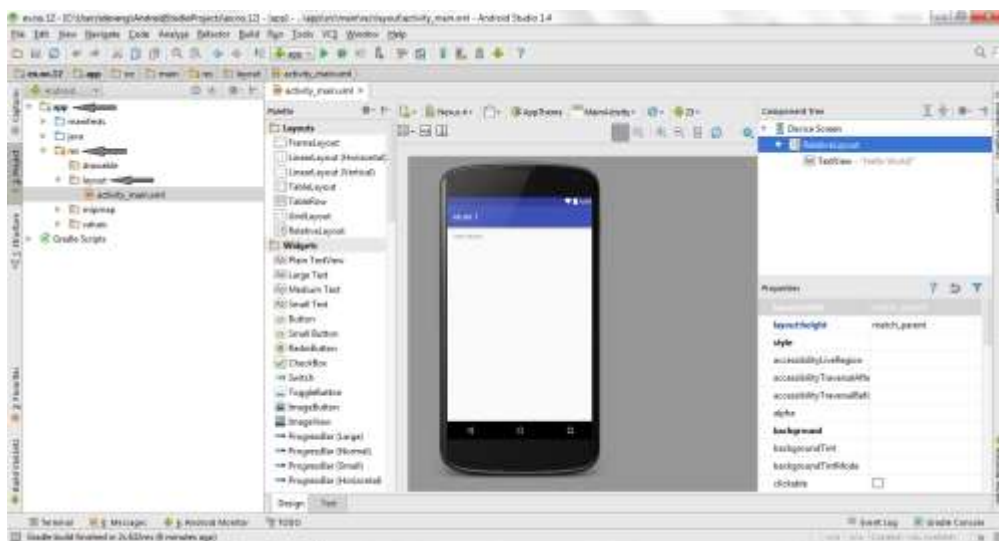
- Type the Activity Name as **AlarmReceiver** and click **Finish** button.



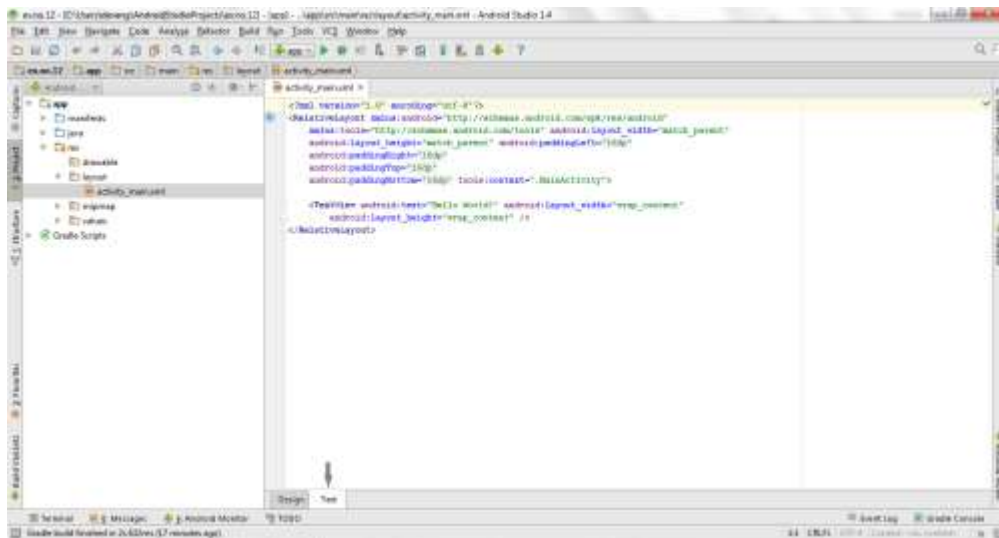
- Thus Second Activity For the application is created.

## Designing layout for the Android Application:

- Click on **app -> res -> layout -> activity\_main.xml**.



- Now click on **Text** as shown below.

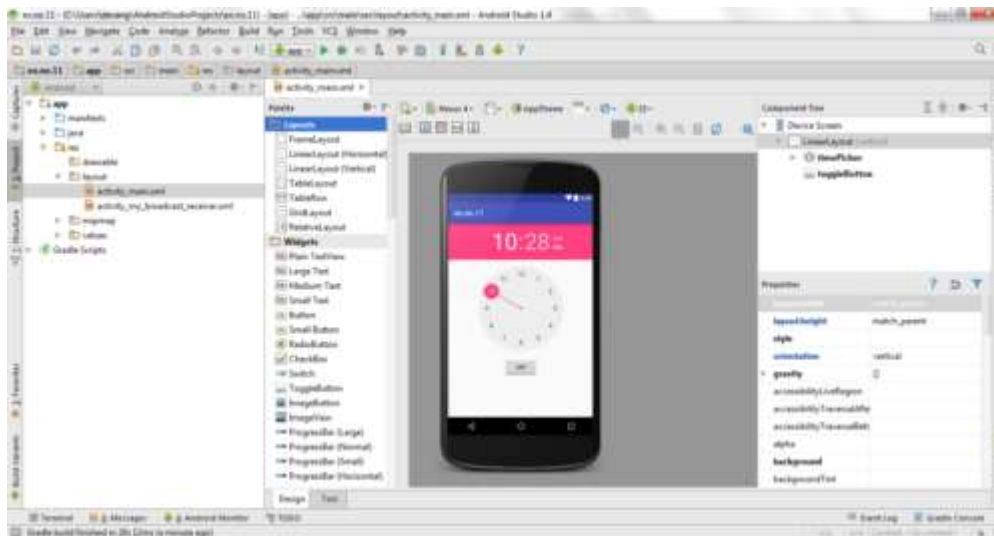


- Then delete the code which is there and type the code as given below.

### Code for Activity\_main.xml:

```
1
2
3 <?xml version="1.0" encoding="utf-8"?>
4 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
5     android:layout_width="match_parent"
6     android:layout_height="match_parent"
7     android:orientation="vertical">
8
9     <TimePicker
10         android:id="@+id/timePicker"
11         android:layout_width="wrap_content"
12         android:layout_height="wrap_content"
13         android:layout_gravity="center" />
14
15     <ToggleButton
16         android:id="@+id/toggleButton"
17         android:layout_width="wrap_content"
18         android:layout_height="wrap_content"
19         android:layout_gravity="center"
20         android:layout_margin="20dp"
21         android:checked="false"
22         android:onClick="OnToggleClicked" />
```

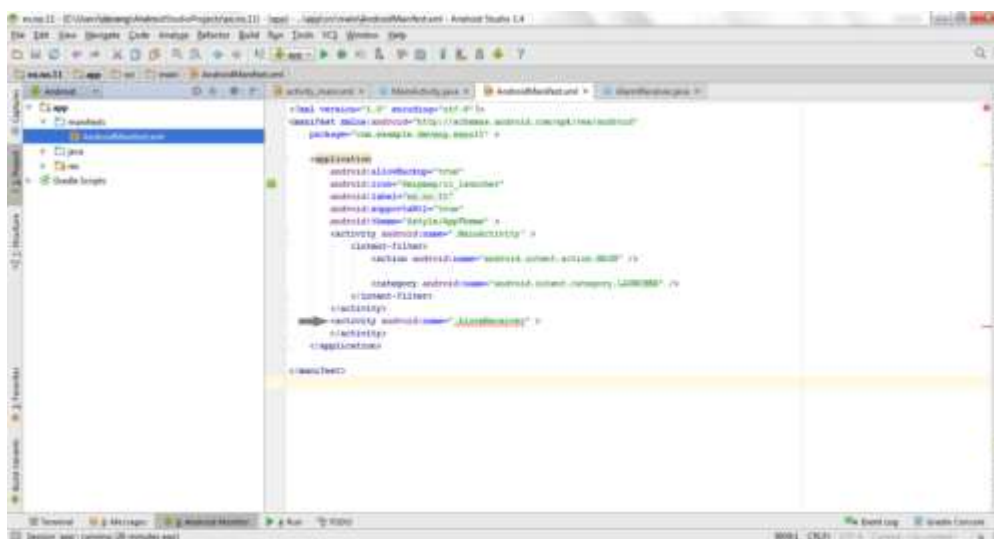
- Now click on **Design** and your application will look as given below.



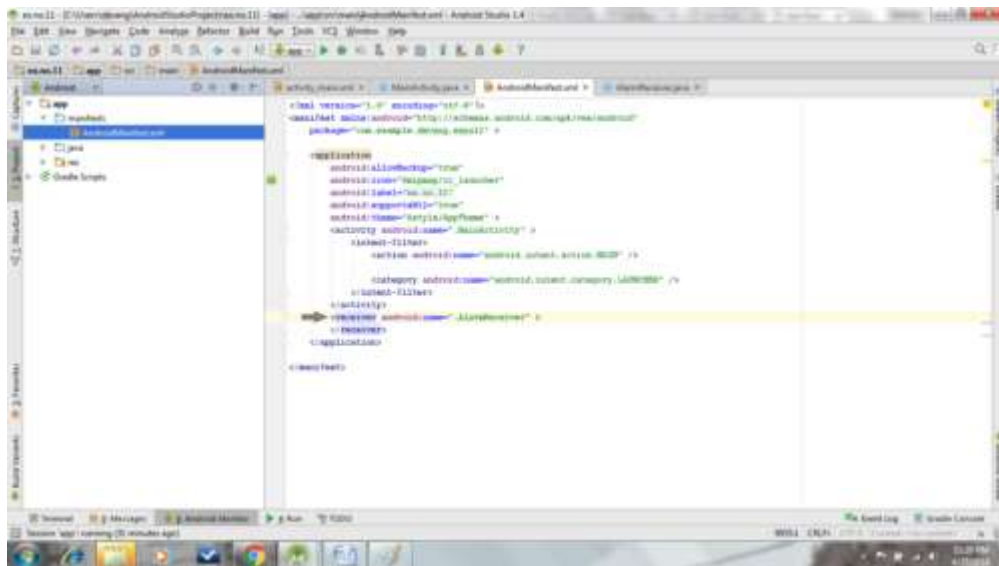
- So now the designing part is completed.

## Changes in Manifest for the Android Application:

- Click on **app** -> **manifests** -> **AndroidManifest.xml**



- Now change the **activity** tag to **receiver** tag in the AndroidManifest.xml file as shown below



## Code for AndroidManifest.xml:

```

1
2 <?xml version="1.0" encoding="utf-8"?>
3 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
4     package="com.example.exno11" >
5
6     <application
7         android:allowBackup="true"
8         android:icon="@mipmap/ic_launcher"
9         android:label="@string/app_name"
10        android:supportRtl="true"
11        android:theme="@style/AppTheme" >
12        <activity android:name=".MainActivity" >
13            <intent-filter>
14                <action android:name="android.intent.action.MAIN" />
15            </intent-filter>
16            <receiver android:name=".AlarmReceiver" >
17            </receiver>
18        </application>
19
20
21</manifest>
22

```

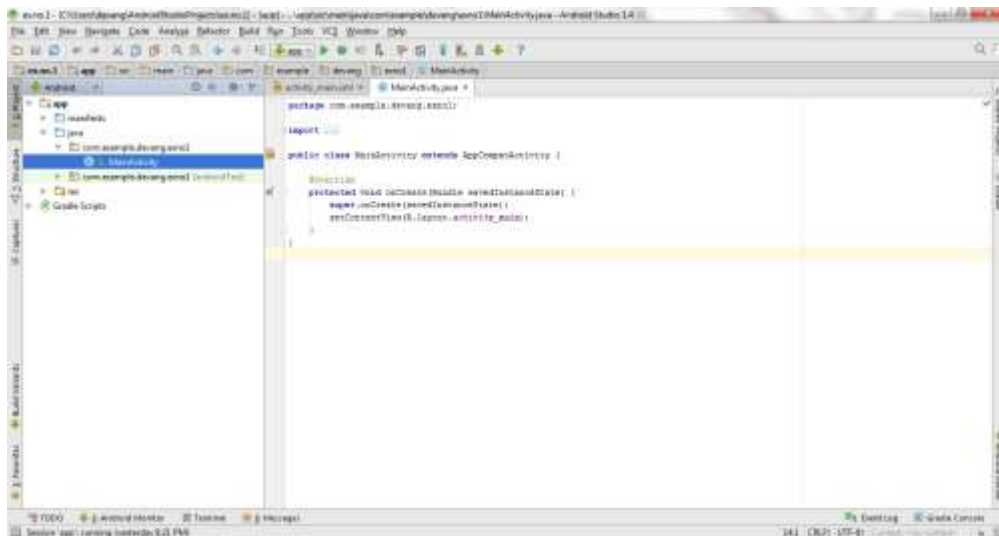
- So now the changes are done in the Manifest.

## Java Coding for the Android Application:

### Java Coding for Main Activity:

- Click on **app** -> **java** -> **com.example.exno11** -> **MainActivity**.





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

### Code for MainActivity.java:

```

1 package com.example.exno11;
2
3 import android.app.AlarmManager;
4 import android.app.PendingIntent;
5 import android.content.Intent;
6 import android.os.Bundle;
7 import android.support.v7.app.AppCompatActivity;
8 import android.view.View;
9 import android.widget.TimePicker;
10 import android.widget.Toast;
11 import android.widget.ToggleButton;
12
13 import java.util.Calendar;
14
15 public class MainActivity extends AppCompatActivity
16 {
17     TimePicker alarmTimePicker;
18     PendingIntent pendingIntent;
19     AlarmManager alarmManager;
20
21     @Override
22     protected void onCreate(Bundle savedInstanceState)
23     {
24         super.onCreate(savedInstanceState);
25         setContentView(R.layout.activity_main);
26         alarmTimePicker = (TimePicker) findViewById(R.id.timePicker);
27         alarmManager = (AlarmManager) getSystemService(ALARM_SERVICE);
28     }
29     public void OnToggleClicked(View view)
30     {
31         long time;
32         if (((ToggleButton) view).isChecked())
33         {
34             Toast.makeText(MainActivity.this, "ALARM ON",
35                 Toast.LENGTH_SHORT).show();
36         }
37     }
38 }

```

```

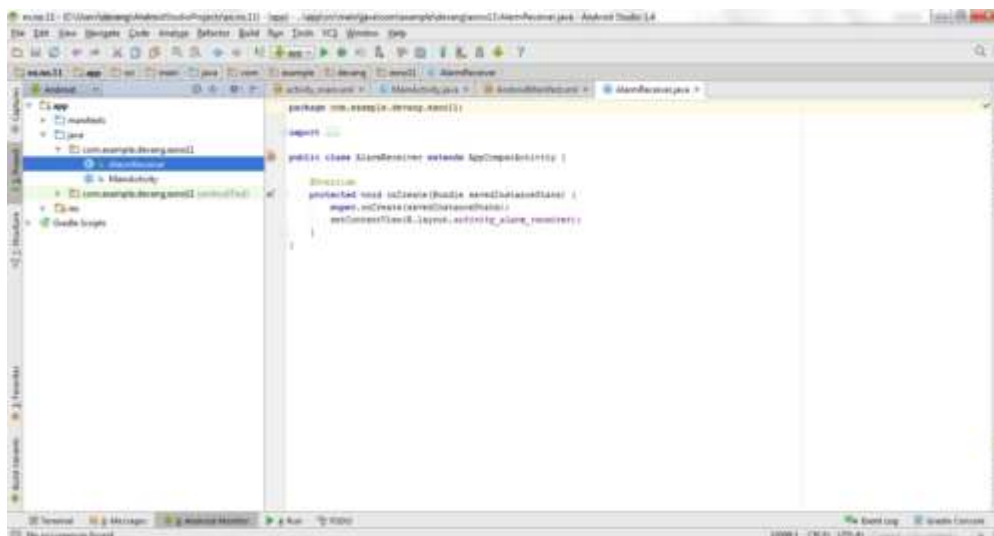
31         Calendar calendar = Calendar.getInstance();
32         calendar.set(Calendar.HOUR_OF_DAY,
33 alarmTimePicker.getCurrentHour());
34         calendar.set(Calendar.MINUTE,
35 alarmTimePicker.getCurrentMinute());
36         Intent intent = new Intent(this, AlarmReceiver.class);
37         pendingIntent = PendingIntent.getBroadcast(this, 0, intent,
38 PendingIntent.FLAG_UPDATE_CURRENT);
39         time=(calendar.getTimeInMillis()-
40 (calendar.getTimeInMillis()%60000));
41         if(System.currentTimeMillis()>time)
42         {
43             if (calendar.AM_PM == 0)
44                 time = time + (1000*60*60*12);
45             else
46                 time = time + (1000*60*60*24);
47         }
48         alarmManager.setRepeating(AlarmManager.RTC_WAKEUP, time,
49 10000, pendingIntent);
50     }
51     else
52     {
53         alarmManager.cancel(pendingIntent);
54         Toast.makeText(MainActivity.this, "ALARM OFF",
55 Toast.LENGTH_SHORT).show();
56     }
57 }
58 }
59 }

```

- So now the Coding part of Main Activity is completed.

### Java Coding for Alarm Receiver:

- Click on **app -> java -> com.example.exno11 -> AlarmReceiver.**



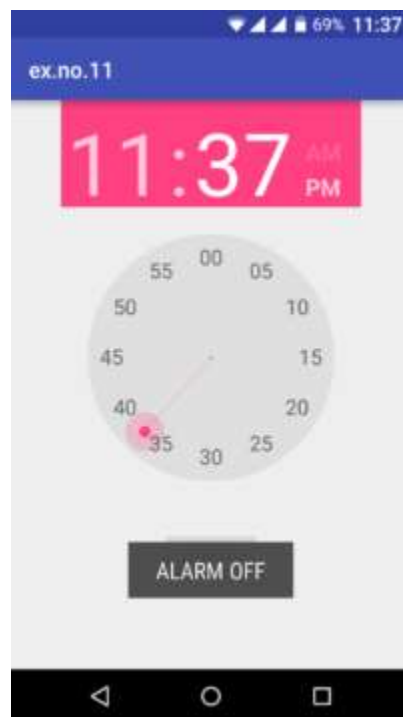
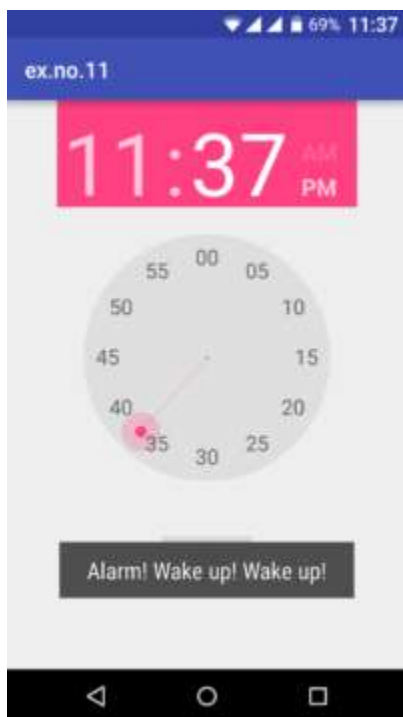
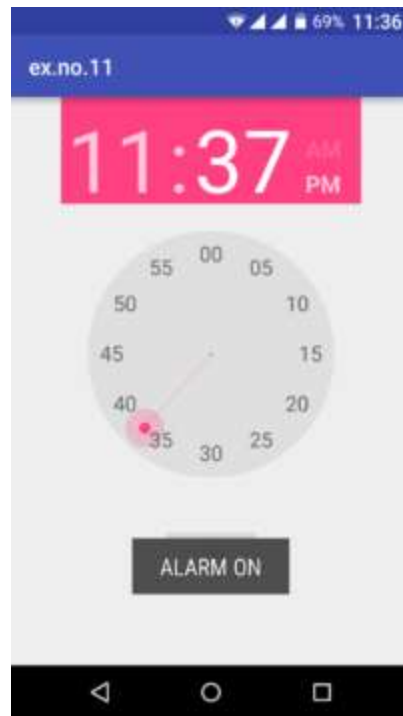
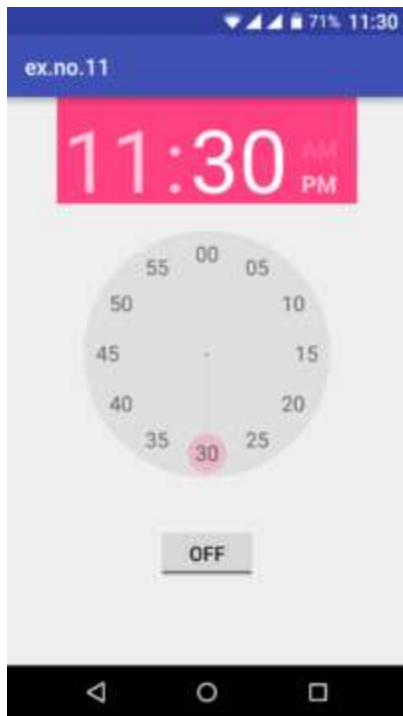
- Then delete the code which is there and type the code as given below.

### Code for AlarmReceiver.java:

```
?  
1 package com.example.exno11;  
2  
3 import android.content.BroadcastReceiver;  
4 import android.content.Context;  
5 import android.content.Intent;  
6 import android.media.Ringtone;  
7 import android.media.RingtoneManager;  
8 import android.net.Uri;  
9 import android.widget.Toast;  
10  
11 public class AlarmReceiver extends BroadcastReceiver  
12 {  
13     @Override  
14     public void onReceive(Context context, Intent intent)  
15     {  
16         Toast.makeText(context, "Alarm! Wake up! Wake up!",  
17             Toast.LENGTH_LONG).show();  
18         Uri alarmUri =  
19             RingtoneManager.getDefaultUri(RingtoneManager.TYPE_ALARM);  
20         if (alarmUri == null)  
21         {  
22             alarmUri =  
23                 RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);  
24         }  
25         Ringtone ringtone = RingtoneManager.getRingtone(context,  
26             alarmUri);  
27         ringtone.play();  
28     }  
29 }
```

- So now the Coding part of Alarm Receiver is also completed.
- Now run the application to see the output.

### Output:



## Result:

Thus Android Application that creates Alarm Clock is developed and executed successfully.