**Lab 23: Android AlarmManager**

# **Introduction**

You may get access to the system alarm by using the Android AlarmManager. You can use it to schedule your programme to run at a certain future time. Whether or not your phone is in use, it still functions. The AlarmManager object will maintain a CPU "wake lock" that ensures the phone won't go to sleep until the broadcast's (alarm's) handling.

**Let’s get Started:**

In this experiment we will develop an Android App to demonstrate the use of Android AlarmManager.

**Step 1: Create a New Project in Android Studio as shown below**

Graphical user interface, text, application

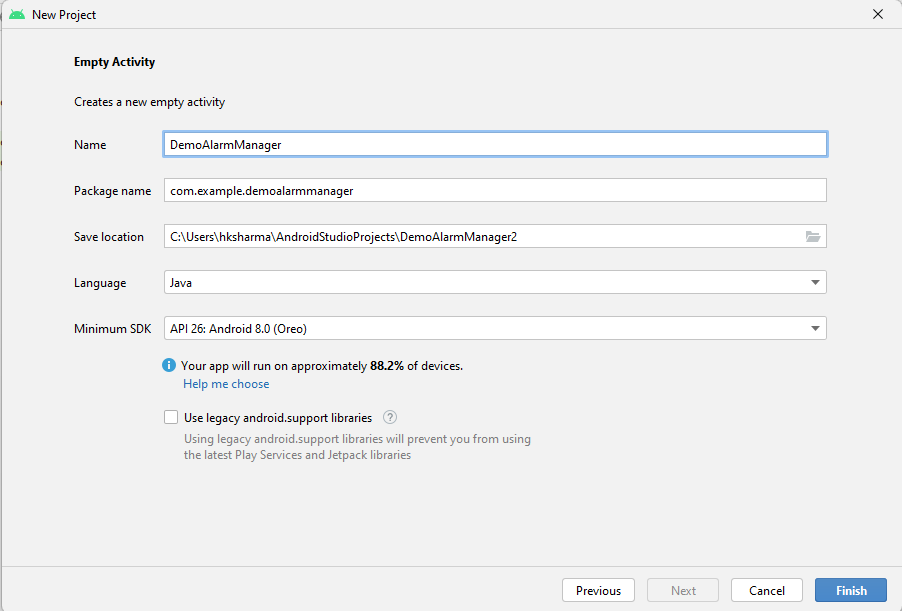
Description automatically generated

**Step 2: Select Empty Activity as shown below**

Graphical user interface, application, shape

Description automatically generated

**Step 3: Provide a Project Name as shown below**

****

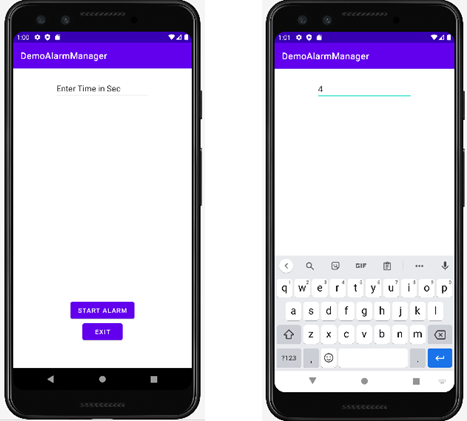
**Step 4: Update MainActivity.java as per the code given below**

**package** com.example. demoalarmmanager;  
**import** androidx.appcompat.app.AppCompatActivity;  
**import** android.os.Bundle;  
**public class** MainActivity **extends** AppCompatActivity {  
 @Override  
 **protected void** onCreate(Bundle savedInstanceState) {  
 **super**.onCreate(savedInstanceState);  
 setContentView(R.layout.***activity\_main***);  
 }  
}

**Step 5: Update activity\_main.xml for Vertical Orientation as per the code given below**

*<?***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=""**>  
  
 <**Button  
 android:id="@+id/button"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Start Alarm"  
 android:layout\_alignParentBottom="true"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginBottom="103dp"** />  
  
 <**Button  
 android:id="@+id/button2"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentBottom="true"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginBottom="56dp"  
 android:text="EXIT"** />  
  
 <**EditText  
 android:id="@+id/time"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_alignParentTop="true"  
 android:layout\_centerHorizontal="true"  
 android:layout\_marginTop="22dp"  
 android:text="Enter Time in Sec"  
 android:ems="10"** />  
</**RelativeLayout**>

**Step 6: Check Output on Android Emulator and it should look like as given below**

****

**Voila!!** We have successfully completed this lab.