Andhra Pradesh State Skill Development Corporation

ANDROID
APPLICATION DEVELOPMENT

ALARM MANAGER

AlarmManager

Android AlarmManager allows you to access system alarm.

By the help of Android AlarmManager in android, you can schedule your application to run at a specific time in the future. It works whether your phone is running or not.

The Android AlarmManager holds a CPU wake lock that provides guarantee not to sleep the phone until broadcast is handled.

next →← prev Android AlarmManager Android AlarmManager allows you to access system alarm.

By the help of Android AlarmManager in android, you can schedule your application to run at a specific time in the future. It works whether your phone is running or not.

The Android AlarmManager holds a CPU wake lock that provides guarantee not to sleep the phone until broadcast is handled.

Android AlarmManager Example

Let's see a simple AlarmManager example that runs after a specific time provided by user.
activity_main.xml

You need to drag only a edittext and a button as given below.

File: activity_main.xml

```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"
    tools:context="example.javatpoint.com.alarmmanager.MainActivity">

    <Button
        android:id="@+id/button"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Start"
        android:layout_alignParentBottom="true"
        android:layout_centerHorizontal="true"
        android:layout_marginBottom="103dp" />

    <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:ems="10" />
</RelativeLayout>
```

Activity class

The activity class starts the alarm service when user clicks on the button.

File: MainActivity.java

```java
import android.app.AlarmManager;
import android.app.PendingIntent;
import android.content.Intent;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;

public class MainActivity extends AppCompatActivity {
    Button start;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        start= findViewById(R.id.button);

        start.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                startAlert();
            }
        });
    }

    public void startAlert(){
        EditText text = findViewById(R.id.time);
        int i = Integer.parseInt(text.getText().toString());
        Intent intent = new Intent(this, MyBroadcastReceiver.class);
        PendingIntent pendingIntent = PendingIntent.getBroadcast(
                this.getApplicationContext(), 234324243, intent, 0);
        AlarmManager alarmManager = (AlarmManager) getSystemService(ALARM_SERVICE);
        alarmManager.set(AlarmManager.RTC_WAKEUP, System.currentTimeMillis()
                + (i * 1000), pendingIntent);
        Toast.makeText(this, "Alarm set in " + i + " seconds",Toast.LENGTH_LONG).show();
    }
}
```

Let's create BroadcastReceiver class that starts alarm.

File: MyBroadcastReceiver.java

```java
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.media.MediaPlayer;
import android.widget.Toast;

public class MyBroadcastReceiver extends BroadcastReceiver {
    MediaPlayer mp;
    @Override
    public void onReceive(Context context, Intent intent) {
        mp=MediaPlayer.create(context, R.raw.alarm);
        mp.start();
        Toast.makeText(context, "Alarm....", Toast.LENGTH_LONG).show();
    }
}
```

File: AndroidManifest.xml
You need to provide a receiver entry in AndroidManifest.xml file.

```xml
<receiver android:name="MyBroadcastReceiver" >
</receiver>
```

Let's see the full code of AndroidManifest.xml file.

```xml
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.alarmmanager">

    <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>
        <receiver android:name="MyBroadcastReceiver" >
        </receiver>
    </application>

</manifest>
```
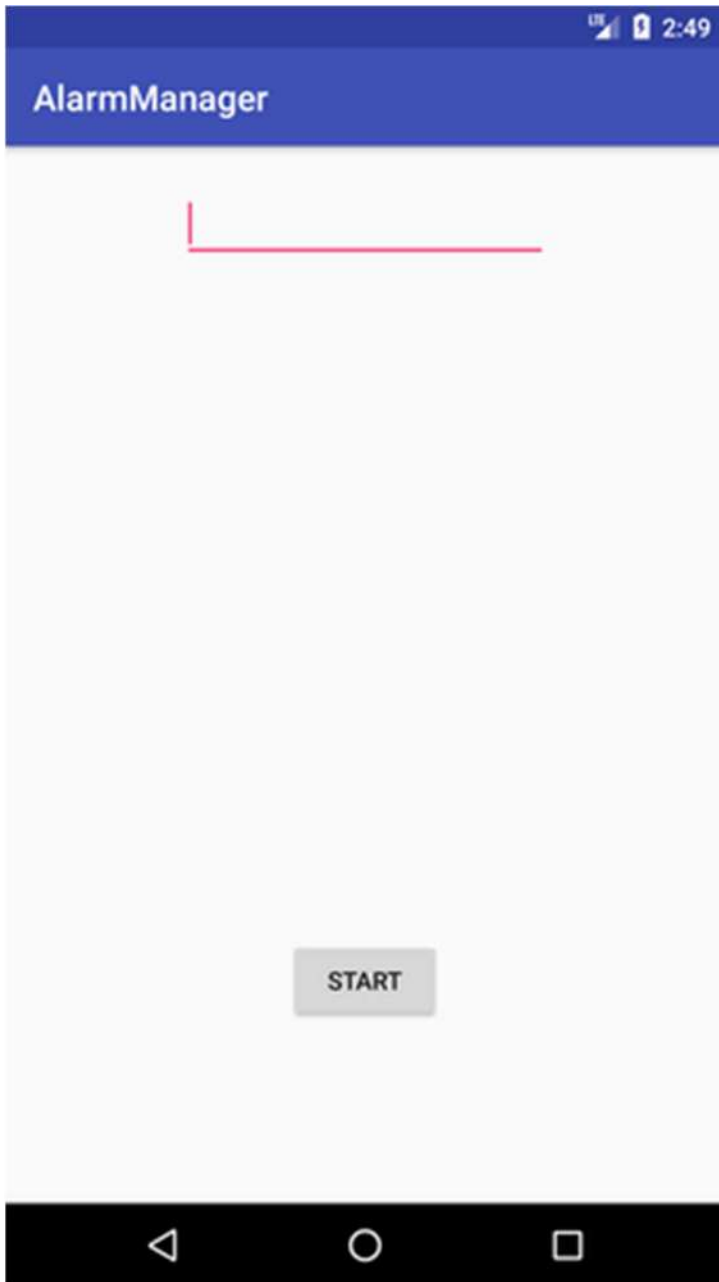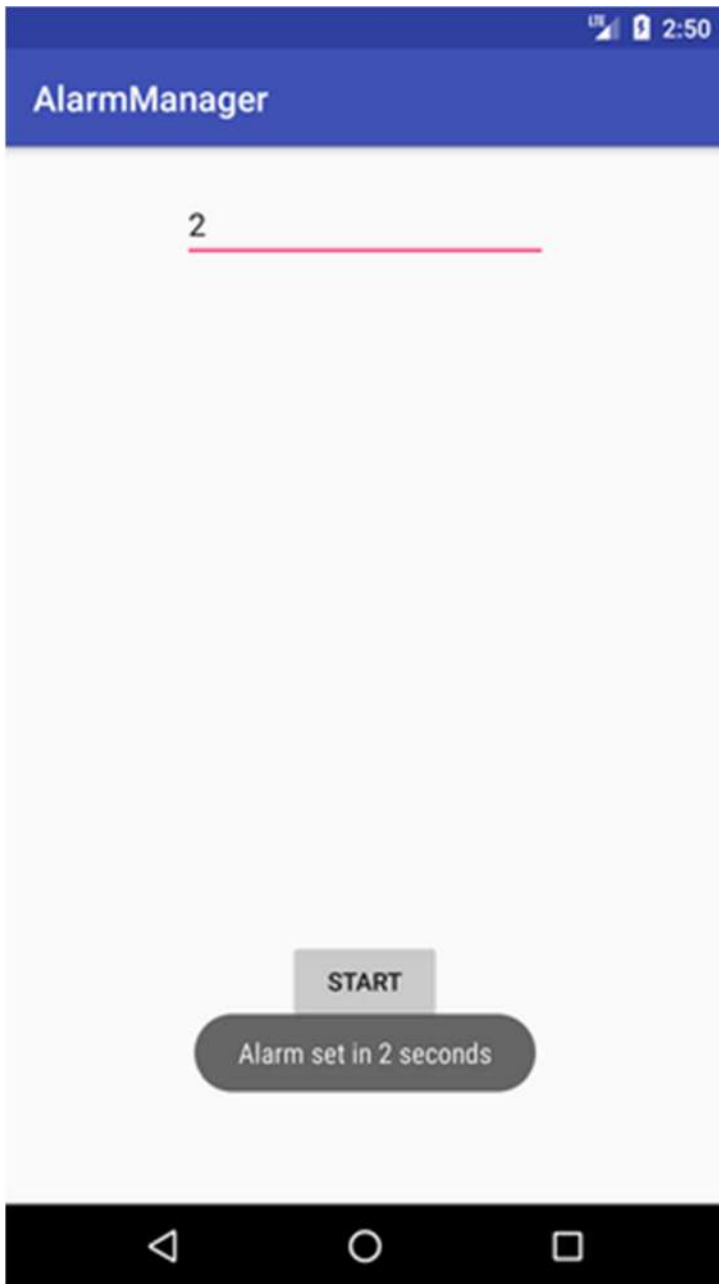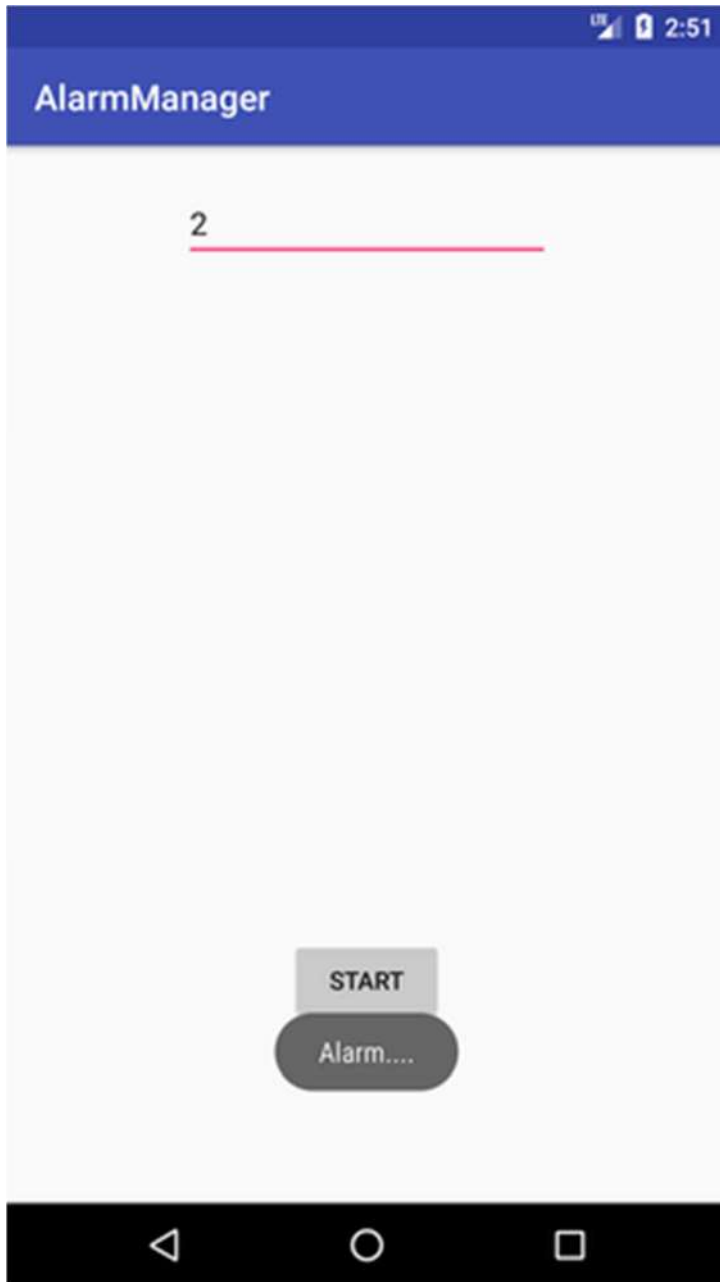
Output:

a1

a2

a3