**[2CEIT5PE5: MOBILE APPLICATION DEVELOPMENT]**

Practical: 7

AIM: What is BroadcastReceiver? Difference between service and BroadcastReceiver. Create an Alarm application by using service & BroadcastReceiver.



Submitted By: Kanzariya Dhavanik

Enrollment number: 21012022022



**Department of Computer Engineering/Information Technology**

**What is BroadcastReceiver?**

In Android, broadcasts are the system-wide events that occur due to changes in system state, like LOW\_BATTERY, incoming or outgoing calls, network connectivity change, device reboot, etc. These are some of the system events that require us to perform some action. Android’s system sends broadcasts when various system events occur.

|  |  |
| --- | --- |
| **BroadcastReceiver** | **Service** |
| Broadcast intents are usually delivered to all BroadcastReceivers registered for that intent. | Services are only sent to one matching Service instance. |
| BroadcastReceiver is used when you want to fire some stuff or code during some event. | Service is used when you want to do something in background, any long running process can be done using Service in Background. |
| Broadcast receivers are limited by maximum amount of time. | Service are run in background long period of time. |
| Usually system will send some info which can be received by your app. | If you want to do something in background. this will be running always in background even if the application closed. |

**activity\_main.xml**

***<?*xml version="1.0" encoding="utf-8"*?>*<androidx.constraintlayout.widget.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=".MainActivity">  
  
 <com.google.android.material.appbar.AppBarLayout  
 android:id="@+id/appbar"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 app:layout\_constraintTop\_toTopOf="parent">  
  
 <androidx.appcompat.widget.Toolbar  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Alarm Application"  
 android:textSize="18dp"  
 android:textStyle="bold"/>  
  
 </androidx.appcompat.widget.Toolbar>  
  
 </com.google.android.material.appbar.AppBarLayout>**

**<androidx.core.widget.NestedScrollView  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 app:layout\_constraintTop\_toBottomOf="@id/appbar"  
 app:layout\_constraintStart\_toStartOf="parent"  
 android:layout\_marginTop="60dp"  
 app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"  
 >  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
  
 <com.google.android.material.card.MaterialCardView  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginTop="30dp"  
 android:layout\_marginEnd="20dp"  
 app:cardCornerRadius="15dp"  
 app:cardElevation="10dp"  
 android:layout\_marginBottom="40dp">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
  
 <ImageView  
 android:layout\_width="match\_parent"  
 android:layout\_height="200dp"  
 android:src="@drawable/alarm"  
 android:scaleType="fitXY"/>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:backgroundTint="#EAEFEA"  
 android:orientation="vertical"  
 android:padding="20dp">  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Create Alarm Time"  
 android:textStyle="bold"  
 android:textSize="15dp"/>  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 android:layout\_marginStart="10dp"  
 android:text=**

**"By pressing buttons, Alarm can be created and cancelled."/>  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="10dp"  
 android:text=**

**"Check Current time by looking below real digital clock."  
 android:layout\_marginTop="10dp"/>  
  
 <TextClock  
 android:id="@+id/show\_time\_tv"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="center"  
 android:text="Show Time Here"  
 android:textStyle="bold"  
 android:layout\_marginTop="10dp"  
 android:textSize="15dp"/>  
  
 <com.google.android.material.button.MaterialButton  
 android:id="@+id/btn\_set\_alarm"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 app:cornerRadius="25dp"  
 android:layout\_marginTop="10dp"  
 app:icon="@drawable/ic\_baseline\_add\_alarm\_24"  
 android:text="Create Alarm"  
 android:textStyle="bold"  
 android:textAllCaps="false"/>  
  
 </LinearLayout>  
  
 </LinearLayout>  
  
 </com.google.android.material.card.MaterialCardView>  
  
 <com.google.android.material.card.MaterialCardView  
 android:id="@+id/cancel\_alarm\_card\_view"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginStart="20dp"  
 android:layout\_marginEnd="20dp"  
 app:cardCornerRadius="15dp"  
 app:cardElevation="10dp">  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical">  
  
 <ImageView  
 android:layout\_width="match\_parent"  
 android:layout\_height="200dp"  
 android:scaleType="fitXY"  
 android:src="@drawable/alarm" />  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:backgroundTint="#EAEFEA"  
 android:orientation="vertical"  
 android:padding="20dp">**

**<TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Set Alarm Time"  
 android:textStyle="bold"  
 android:textSize="15dp"/>  
  
 <TextView  
 android:id="@+id/set\_alarm\_time"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="10dp"  
 android:text="Time"  
 android:textStyle="bold"  
 android:textSize="15dp"/>**

**<com.google.android.material.button.MaterialButton  
 android:id="@+id/btn\_cancel\_alarm"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 app:cornerRadius="25dp"  
 android:layout\_marginTop="20dp"  
 app:icon="@drawable/ic\_baseline\_alarm\_off\_24"  
 android:text="Cancel Alarm"  
 android:textStyle="bold"  
 android:textAllCaps="false"/>  
  
 </LinearLayout>  
  
 </LinearLayout>  
  
 </com.google.android.material.card.MaterialCardView>  
 </LinearLayout>  
  
 </androidx.core.widget.NestedScrollView>  
  
</androidx.constraintlayout.widget.ConstraintLayout>**

M**ainActivity.kt**

**package com.example.madpractical7\_21012022022  
  
import android.app.AlarmManager  
import android.app.PendingIntent  
import android.app.TimePickerDialog  
import android.content.Intent  
import androidx.appcompat.app.AppCompatActivity  
import android.os.Bundle  
import android.view.View  
import android.widget.TextClock  
import android.widget.TextView  
import com.google.android.material.button.MaterialButton  
import com.google.android.material.card.MaterialCardView  
import java.util.\*  
  
class MainActivity : AppCompatActivity() {  
 var mili:Long=0  
  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 setContentView(R.layout.*activity\_main*)  
  
 val CancleAlarmCardView = findViewById<MaterialCardView>**

**(R.id.*cancel\_alarm\_card\_view*)  
 val BtnCreateAlarm = findViewById<MaterialButton>(R.id.*btn\_set\_alarm*)  
 val SetAlarmTime = findViewById<TextView>(R.id.*set\_alarm\_time*)  
 val BtnCancelAlarm = findViewById<MaterialButton>(R.id.*btn\_cancel\_alarm*)  
 val clockTC = findViewById<TextClock>(R.id.*show\_time\_tv*)  
  
 clockTC.*format12Hour* = "hh:mm:ss a"  
  
 CancleAlarmCardView.*visibility*=View.*GONE* BtnCreateAlarm.setOnClickListener {  
 var cal:Calendar = Calendar.getInstance()  
 var hour = cal.get(Calendar.*HOUR\_OF\_DAY*)  
 var min = cal.get(Calendar.*MINUTE*)  
 val tpd = TimePickerDialog(this,TimePickerDialog.OnTimeSetListener(function = {**

**view, h, m ->  
 mili=getMillis(h,m)  
 setAlarm(getMillis(h,m),"Start")  
 CancleAlarmCardView.*visibility*=View.*VISIBLE* SetAlarmTime.*text*=h.toString()+":"+m.toString()  
 }),hour,min,false)  
 tpd.show()  
 }  
  
 BtnCancelAlarm.setOnClickListener{  
 setAlarm(mili,"Stop")  
 CancleAlarmCardView.*visibility*=View.*GONE* }  
 }  
 fun setAlarm(millisTime: Long, str: String)  
 {  
 val intent = Intent(this, AlarmBroadcastReceiver::class.*java*)  
 intent.putExtra("Service1", str)  
 val pendingIntent =**

**PendingIntent.getBroadcast(*applicationContext*, 234324243, intent, 0)  
 val alarmManager = getSystemService(*ALARM\_SERVICE*) as AlarmManager  
 if(str == "Start") {  
 alarmManager.setExact(  
 AlarmManager.*RTC\_WAKEUP*,  
 millisTime,  
 pendingIntent  
 )  
 }else if(str == "Stop")  
 {  
 alarmManager.cancel(pendingIntent)  
 sendBroadcast(intent)  
 }  
 }  
  
 fun getMillis(hour:Int,min:Int):Long  
 {  
 val setcalendar = Calendar.getInstance()  
 setcalendar[Calendar.*HOUR\_OF\_DAY*] = hour  
 setcalendar[Calendar.*MINUTE*] = min  
 setcalendar[Calendar.*SECOND*] = 0  
 return setcalendar.*timeInMillis* }  
}**

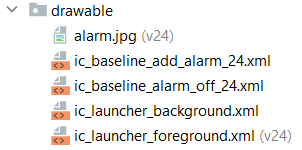
**AlarmBroadcastReceiver.kt**

**package com.example.madpractical7\_21012022022  
  
import android.content.BroadcastReceiver  
import android.content.Context  
import android.content.Intent  
  
class AlarmBroadcastReceiver : BroadcastReceiver() {  
  
 override fun onReceive(context: Context, intent: Intent) {  
 val str1 = intent.getStringExtra("Service1")  
 if(str1 == "Start" || str1 == "Stop")  
 {  
 val intentService = Intent(context, AlarmService::class.*java*)  
 intentService.putExtra("Service1", intent.getStringExtra("Service1"))  
 if(str1 == "Start")  
 {  
 context.startService(intentService)  
 }  
 else if(str1=="Stop") {  
 context.stopService(intentService)  
 }  
 }  
 }  
}**

**AlarmService.kt**

**package com.example.madpractical7\_21012022022  
  
import android.app.Service  
import android.content.Intent  
import android.media.MediaPlayer  
import android.os.IBinder  
  
class AlarmService : Service() {  
 var mp: MediaPlayer? = null  
 override fun onBind(intent: Intent): IBinder {  
 TODO("Return the communication channel to the service.")  
 }  
 override fun onStartCommand(intent: Intent?, flags: Int, startId: Int): Int {  
 if(intent != null) {  
 mp = MediaPlayer.create(this, R.raw.alarm);  
 mp?.start()  
 }  
 return START\_STICKY  
 }  
 override fun onDestroy() {  
 mp?.stop()  
 super.onDestroy()  
 }  
}**

**Resource:**

** **

**AndroidManifest.xml**

**<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 package="com.example.madpractical7\_21012022022">  
 <uses-permission android:name="android.permission.SCHEDULE\_EXACT\_ALARM"/>  
 <application  
 android:allowBackup="true"  
 android:dataExtractionRules="@xml/data\_extraction\_rules"  
 android:fullBackupContent="@xml/backup\_rules"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:roundIcon="@mipmap/ic\_launcher\_round"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.MADPractical7\_21012022022"  
 tools:targetApi="31">  
 <service  
 android:name=".AlarmService"  
 android:enabled="true"  
 android:exported="true"/>  
  
 <receiver  
 android:name=".AlarmBroadcastReceiver"  
 android:directBootAware="true"  
 android:enabled="true"  
 android:exported="true" >  
 <intent-filter>  
 <action android:name="android.intent.action.BOOT\_COMPLETED"/>  
 <action android:name="android.intent.action.LOCKED\_BOOT\_COMPLETED"/>  
 </intent-filter>  
 </receiver>  
  
 <activity  
 android:name=".MainActivity"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="android.intent.action.MAIN" />  
  
 <category android:name="android.intent.category.LAUNCHER" />  
 </intent-filter>  
 </activity>  
 </application>  
</manifest>**

**Output:**

