[2CEIT5PE18: MOBILE APPLICATION DEVELOPMENT]

Practical: 4

AIM-Create an Android Alarm application by using service & BroadcastReceiver.

.

Submitted By: KANSAGARA KRISH KALPESHBHAI
Enrollment number: 23012011026

Ganpat University
॥ विद्यया समाजोत्कर्षः ॥
U.V. Patel College of Engineering

Department of Computer Engineering/Information Technology

Practical-4

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:id="@+id/main"

    android:layout_width="match_parent"

    android:layout_height="match_parent"

    tools:context=".MainActivity">


    <androidx.core.widget.NestedScrollView

        android:layout_width="match_parent"

        android:layout_height="match_parent">


        <androidx.constraintlayout.widget.ConstraintLayout

            android:layout_width="match_parent"

            android:layout_height="wrap_content">



            <com.google.android.material.card.MaterialCardView

                android:id="@+id/card1"

                android:layout_width="match_parent"

                android:layout_height="wrap_content"
                android:layout_margin="20dp"
                app:cardElevation="5dp"
                style="?attr/materialCardViewElevatedStyle"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toTopOf="parent">

                <androidx.constraintlayout.widget.ConstraintLayout
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content">

                    <ImageView
                        android:id="@+id/alram_image"
                        android:layout_width="match_parent"
                        android:layout_height="200dp"
                        android:scaleType="centerCrop"
                        android:src="@drawable/img"
                        app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
                        app:layout_constraintTop_toTopOf="parent" />

                    <TextView
                        android:id="@+id/alram_label"
                        android:layout_width="wrap_content"

                        android:layout_height="wrap_content"

                        android:padding="10dp"

                        android:text="Create Alarm Time"

android:textAppearance="?attr/textAppearanceTitleMedium"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@id/alram_image" />


                    <TextView

                        android:id="@+id/alram_description"

                        android:layout_width="wrap_content"

                        android:layout_height="wrap_content"

                        android:padding="10dp"

                        android:text="By pressing buttons, Alarm can
be created and cancelled."

android:textAppearance="?attr/textAppearanceBodyMedium"

android:textColor="?android:attr/textColorSecondary"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@id/alram_label" />
```

```
                    <TextView
                        android:id="@+id/alram_description1"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:padding="10dp"
                        android:text="Check Current time by looking
below real digital clock."

android:textColor="?android:attr/textColorSecondary"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@id/alram_description" />


                    <TextClock
                        android:id="@+id/datetime"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:format12Hour="hh:mm:ss a MMM, dd
yyyy"
                        android:padding="10dp"

android:textAppearance="?attr/textAppearanceTitleMedium"
                        app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintStart_toStartOf="parent"
```

```
app:layout_constraintTop_toBottomOf="@id/alram_description1" />


<com.google.android.material.button.MaterialButton
                        android:id="@+id/create_alarm"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_marginStart="10dp"
                        android:layout_marginLeft="10dp"
                        android:text="Create Alarm"
                        app:icon="@drawable/baseline_add_alarm_24"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@id/datetime" />



                </androidx.constraintlayout.widget.ConstraintLayout>


            </com.google.android.material.card.MaterialCardView>


            <com.google.android.material.card.MaterialCardView
                android:id="@+id/card2"
                style="?attr/materialCardViewElevatedStyle"
                android:layout_width="match_parent"
                android:layout_height="wrap_content"
                android:layout_margin="20dp"
                app:cardElevation="5dp"
                app:layout_constraintEnd_toEndOf="parent"
                app:layout_constraintStart_toStartOf="parent"
                app:layout_constraintTop_toBottomOf="@id/card1">

                <androidx.constraintlayout.widget.ConstraintLayout
                    android:layout_width="match_parent"
                    android:layout_height="wrap_content">

                    <ImageView
                        android:id="@+id/alram_image1"
                        android:layout_width="match_parent"
                        android:layout_height="200dp"
                        android:scaleType="centerCrop"
                        android:src="@drawable/img"
                        app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
                        app:layout_constraintTop_toTopOf="parent" />

                    <TextView
```

```
                        android:id="@+id/alram_label1"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:padding="10dp"
                        android:text="Create Alarm Time"

android:textAppearance="?attr/textAppearanceTitleMedium"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@id/alram_image1" />


                    <TextView
                        android:id="@+id/alram_description2"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:padding="10dp"
                        android:text="By pressing buttons, Alarm can
be created and cancelled."

android:textAppearance="?attr/textAppearanceBodyMedium"

android:textColor="?android:attr/textColorSecondary"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@id/alram_label1" />
```

```
                    <TextView
                        android:id="@+id/alram_description3"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:padding="10dp"
                        android:text="Check Current time by looking
below real digital clock."

android:textColor="?android:attr/textColorSecondary"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@id/alram_description2" />
                    <TextClock
                        android:id="@+id/datetime1"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:format12Hour="hh:mm:ss a MMM, dd
yyyy"
                        android:padding="10dp"

android:textAppearance="?attr/textAppearanceTitleMedium"
                        app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintStart_toStartOf="parent"
```

```
app:layout_constraintTop_toBottomOf="@id/alram_description3" />


<com.google.android.material.button.MaterialButton
                        android:id="@+id/cancle_alarm"
                        android:layout_width="wrap_content"
                        android:layout_height="wrap_content"
                        android:layout_marginStart="10dp"
                        android:layout_marginLeft="10dp"
                        android:text="Cancle Alarm"
                        app:icon="@drawable/baseline_add_alarm_24"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@id/datetime1" />
                </androidx.constraintlayout.widget.ConstraintLayout>


            </com.google.android.material.card.MaterialCardView>
        </androidx.constraintlayout.widget.ConstraintLayout>
    </androidx.core.widget.NestedScrollView>
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.kt

```
package com.bytemap.mad_23012011026_practical4
```

```
import android.Manifest
import android.app.AlarmManager
import android.app.PendingIntent
import android.app.TimePickerDialog
import android.content.Intent
import android.content.pm.PackageManager
import android.os.Build
import android.os.Bundle
import android.provider.Settings
import android.view.View
import android.widget.TextView
import android.widget.Toast
import androidx.activity.enableEdgeToEdge
import androidx.appcompat.app.AppCompatActivity
import androidx.core.app.ActivityCompat
import androidx.core.content.ContextCompat
import androidx.core.view.ViewCompat
import androidx.core.view.WindowInsetsCompat
import com.google.android.material.button.MaterialButton
import com.google.android.material.card.MaterialCardView
import java.text.SimpleDateFormat
import java.util.Calendar

class MainActivity : AppCompatActivity() {

    lateinit var cardListAlarm: MaterialCardView
    lateinit var btnCreateAlarm: MaterialButton
    lateinit var btnCancleAlarm: MaterialButton
    lateinit var textAlaramTime: TextView

    private val REQUEST_NOTIFICATION_PERMISSION = 2001
    private val REQUEST_TIME_PICKER = 3001
    private val PENDING_INTENT_REQUEST_CODE = 234324243

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        enableEdgeToEdge()
        setContentView(R.layout.activity_main)
```

```
ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main))
{ v, insets ->
            val systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars())
            v.setPadding(systemBars.left, systemBars.top,
systemBars.right, systemBars.bottom)
            insets
        }

        cardListAlarm = findViewById(R.id.card2)
        btnCreateAlarm = findViewById(R.id.create_alarm)
        btnCancleAlarm = findViewById(R.id.cancle_alarm)
        textAlaramTime = findViewById(R.id.datetime)
        cardListAlarm.visibility = View.GONE

        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.TIRAMISU) {
            if (ContextCompat.checkSelfPermission(this,
Manifest.permission.POST_NOTIFICATIONS) !=
PackageManager.PERMISSION_GRANTED) {
                ActivityCompat.requestPermissions(this,
arrayOf(Manifest.permission.POST_NOTIFICATIONS),
REQUEST_NOTIFICATION_PERMISSION)
            }
        }

        btnCreateAlarm.setOnClickListener {
            showTimeDialog()
        }

        btnCancleAlarm.setOnClickListener {
            cancelAlarm()
            cardListAlarm.visibility = View.GONE
        }
    }

    private fun showTimeDialog() {
        val cldr: Calendar = Calendar.getInstance()
        val hour = cldr.get(Calendar.HOUR_OF_DAY)
        val minutes = cldr.get(Calendar.MINUTE)
        val picker = TimePickerDialog(
            this,
            { _, sHour, sMinute -> sendDialogDateToActivity(sHour,
sMinute) },
            hour,
            minutes,
            false
        )
        picker.show()
    }

    private fun sendDialogDateToActivity(hour: Int, minute: Int) {
        val alarmCalendar: Calendar = Calendar.getInstance()
        val year: Int = alarmCalendar.get(Calendar.YEAR)
        val month: Int = alarmCalendar.get(Calendar.MONTH)
        val day: Int = alarmCalendar.get(Calendar.DATE)
        alarmCalendar.set(year, month, day, hour, minute, 0)

        if (alarmCalendar.timeInMillis <= System.currentTimeMillis())
{
            alarmCalendar.add(Calendar.DAY_OF_YEAR, 1)
        }

        textAlaramTime.text = SimpleDateFormat("dd/MM/yyyy
HH:mm:ss").format(alarmCalendar.time)
        cardListAlarm.visibility = View.VISIBLE

        setAlarm(alarmCalendar.timeInMillis, "Start")
    }

    private fun setAlarm(millisTime: Long, str: String) {
        val intent = Intent(this, AlarmBroadcastReceiver::class.java)
        intent.putExtra("Service1", str)

        val pendingIntent = PendingIntent.getBroadcast(
            applicationContext,
            PENDING_INTENT_REQUEST_CODE,
            intent,
            PendingIntent.FLAG_UPDATE_CURRENT or
PendingIntent.FLAG_IMMUTABLE
        )
        val alarmManager = getSystemService(ALARM_SERVICE) as
AlarmManager

        if (str == "Start") {
            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.S) {
                if (!alarmManager.canScheduleExactAlarms()) {
                    Toast.makeText(this, "Please allow scheduling
exact alarms in settings.", Toast.LENGTH_LONG).show()

startActivity(Intent(Settings.ACTION_REQUEST_SCHEDULE_EXACT_ALARM))
                    return
                }
            }

            if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.M) {

alarmManager.setExactAndAllowWhileIdle(AlarmManager.RTC_WAKEUP,
millisTime, pendingIntent)
            } else {
                alarmManager.setExact(AlarmManager.RTC_WAKEUP,
millisTime, pendingIntent)
            }

            Toast.makeText(this, "Alarm set",
Toast.LENGTH_LONG).show()
        } else if (str == "Stop") {
            alarmManager.cancel(pendingIntent)

            val stopIntent = Intent(this,
AlarmBroadcastReceiver::class.java)
            stopIntent.putExtra("Service1", "Stop")
            sendBroadcast(stopIntent)

            Toast.makeText(this, "Stop Alarm",
Toast.LENGTH_LONG).show()
        }
    }

    private fun cancelAlarm() {
        setAlarm(System.currentTimeMillis(), "Stop")
    }
}
```

AlarmServices.kt

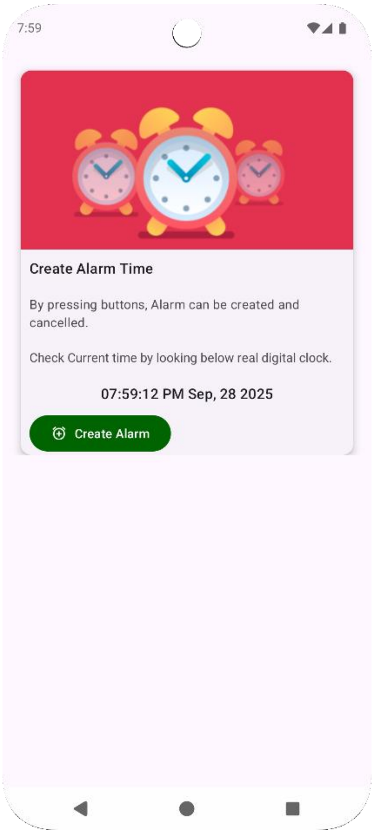
```
package com.bytemap.mad_23012011026_practical4

import android.app.Notification
import android.app.NotificationChannel
import android.app.NotificationManager
import android.app.Service
import android.content.Intent
import android.media.AudioAttributes
import android.media.MediaPlayer
import android.media.RingtoneManager
import android.os.Build
import android.os.IBinder
import android.widget.Toast
import androidx.core.app.NotificationCompat

class AlarmServices : Service() {

    private var mediaPlayer: MediaPlayer? = null
    private val CHANNEL_ID = "alarm_channel"
    private val NOTIFICATION_ID = 1001

    override fun onBind(intent: Intent?): IBinder? = null

    override fun onStartCommand(intent: Intent?, flags: Int, startId:
Int): Int {
        val action = intent?.getStringExtra("Service1")
        when (action) {
            "Start" -> {
                startAsForeground()
                startAlarmSound()
                return START_STICKY
            }
            "Stop" -> {
                stopAlarmSound()
                stopForeground(STOP_FOREGROUND_REMOVE)
                stopSelf()
                return START_NOT_STICKY
            }
            else -> {
                stopAlarmSound()
                stopForeground(STOP_FOREGROUND_REMOVE)
                stopSelf()
                return START_NOT_STICKY
            }
        }
    }

    private fun startAsForeground() {
        createNotificationChannelIfNeeded()

        val notification: Notification =
NotificationCompat.Builder(this, CHANNEL_ID)
            .setContentTitle("Alarm")
            .setContentText("Alarm is ringing")
            .setSmallIcon(android.R.drawable.ic_lock_idle_alarm)
            .setOngoing(true)
            .setPriority(NotificationCompat.PRIORITY_HIGH)
            .build()

        startForeground(NOTIFICATION_ID, notification)
    }

    private fun createNotificationChannelIfNeeded() {
        if (Build.VERSION.SDK_INT >= Build.VERSION_CODES.O) {
            val nm =
getSystemService(NotificationManager::class.java)
            val channel = NotificationChannel(
                CHANNEL_ID,
                "Alarms",
                NotificationManager.IMPORTANCE_HIGH
            )
            channel.setSound(null, null)
            nm.createNotificationChannel(channel)
        }
    }

    private fun startAlarmSound() {
        if (mediaPlayer?.isPlaying == true) return
        val alarmUri =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_ALARM)
            ?:
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION)

        try {
            mediaPlayer = MediaPlayer().apply {
                setAudioAttributes(
                    AudioAttributes.Builder()
                        .setUsage(AudioAttributes.USAGE_ALARM)
                        .setContentType(AudioAttributes.CONTENT_TYPE
_SONIFICATION)
                        .build()
                )
                setDataSource(this@AlarmServices, alarmUri)
                isLooping = true
                prepare()
                start()
            }
        } catch (e: Exception) {
            e.printStackTrace()
            Toast.makeText(this, "Failed to play alarm sound",
Toast.LENGTH_SHORT).show()
        }
    }

    private fun stopAlarmSound() {
        try {
            mediaPlayer?.let {
                if (it.isPlaying) it.stop()
                it.reset()
                it.release()
            }
        } catch (e: Exception) {
            e.printStackTrace()
        } finally {
            mediaPlayer = null
        }
        Toast.makeText(this, "Alarm Cancelled!",
Toast.LENGTH_SHORT).show()
```

```
    }

    override fun onDestroy() {
        stopAlarmSound()
        super.onDestroy()
    }
}
```

AlarmBroadcastReceiver.kt

```
package com.bytemap.mad_24172012068_practical4

import android.content.BroadcastReceiver
import android.content.Context
import android.content.Intent
import androidx.core.content.ContextCompat

class AlarmBroadcastReceiver : BroadcastReceiver() {

    override fun onReceive(context: Context, intent: Intent) {
        val str1 = intent.getStringExtra("Service1")
        if (str1 == "Start" || str1 == "Stop") {
            val intentServices = Intent(context, 
AlarmServices::class.java)
            intentServices.putExtra("Service1", str1)

            if (str1 == "Start") {
                if (android.os.Build.VERSION.SDK_INT >= 
android.os.Build.VERSION_CODES.O) {
                    ContextCompat.startForegroundService(context, 
intentServices)
                } else {
                    context.startService(intentServices)
                }
            } else {
                context.stopService(intentServices)
            }
        }
    }
}
```

Output:

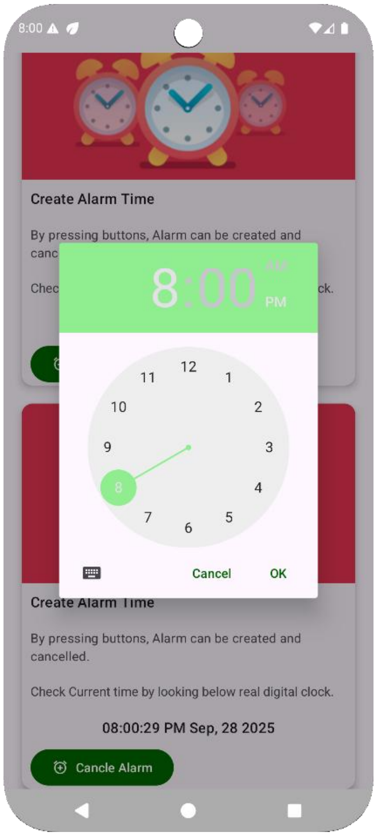


Create Alarm Time

By pressing buttons, Alarm can be created and cancelled.

Check Current time by looking below real digital clock.

07:59:12 PM Sep, 28 2025

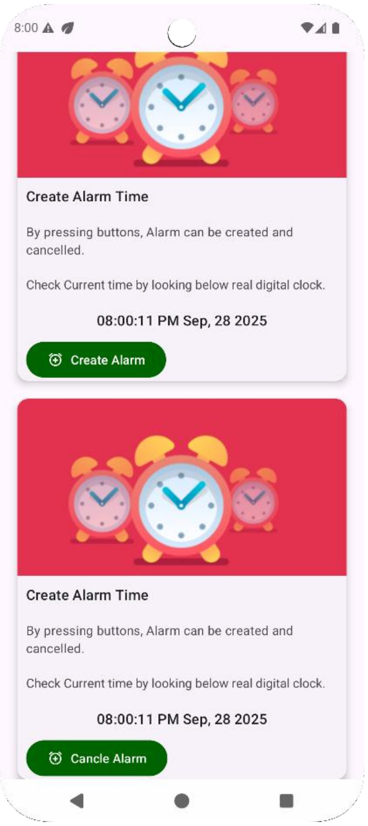


Create Alarm Time

By pressing buttons, Alarm can be created and cancelled.

Check Current time by looking below real digital clock.

08:00:29 PM Sep, 28 2025



Create Alarm Time

By pressing buttons, Alarm can be created and cancelled.

Check Current time by looking below real digital clock.

08:00:11 PM Sep, 28 2025

Create Alarm Time

By pressing buttons, Alarm can be created and cancelled.

Check Current time by looking below real digital clock.

08:00:11 PM Sep, 28 2025