**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">

<ToggleButton

android:id="@+id/alarmToggle"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:textOff="@string/alarm\_off\_toast"

android:textOn="@string/alarm\_on\_toast"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

</androidx.constraintlayout.widget.ConstraintLayout>

**Mainactivity.java**

package com.example.standup;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.app.NotificationCompat;

import android.app.AlarmManager;

import android.app.NotificationChannel;

import android.app.NotificationManager;

import android.app.PendingIntent;

import android.content.Context;

import android.content.Intent;

import android.graphics.Color;

import android.os.Bundle;

import android.os.SystemClock;

import android.widget.CompoundButton;

import android.widget.Toast;

import android.widget.ToggleButton;

public class MainActivity extends AppCompatActivity {

// Notification ID.

private static final int NOTIFICATION\_ID = 0;

// Notification channel ID.

private static final String PRIMARY\_CHANNEL\_ID =

"primary\_notification\_channel";

private NotificationManager mNotificationManager;

/\*\*

\* Initializes the activity.

\*

\* @param savedInstanceState The current state data.

\*/

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

mNotificationManager = (NotificationManager)

getSystemService(NOTIFICATION\_SERVICE);

ToggleButton alarmToggle = findViewById(R.id.alarmToggle);

// Set up the Notification Broadcast Intent.

Intent notifyIntent = new Intent(this, AlarmReceiver.class);

boolean alarmUp = (PendingIntent.getBroadcast(this, NOTIFICATION\_ID,

notifyIntent, PendingIntent.FLAG\_NO\_CREATE) != null);

alarmToggle.setChecked(alarmUp);

final PendingIntent notifyPendingIntent = PendingIntent.getBroadcast

(this, NOTIFICATION\_ID, notifyIntent,

PendingIntent.FLAG\_UPDATE\_CURRENT);

final AlarmManager alarmManager = (AlarmManager) getSystemService

(ALARM\_SERVICE);

// Set the click listener for the toggle button.

alarmToggle.setOnCheckedChangeListener

(new CompoundButton.OnCheckedChangeListener() {

@Override

public void onCheckedChanged

(CompoundButton buttonView, boolean isChecked) {

String toastMessage;

if (isChecked) {

long repeatInterval = AlarmManager.INTERVAL\_FIFTEEN\_MINUTES/15;

long triggerTime = SystemClock.elapsedRealtime()

+ repeatInterval;

// If the Toggle is turned on, set the repeating alarm with

// a 15 minute interval.

if (alarmManager != null) {

alarmManager.setInexactRepeating

(AlarmManager.ELAPSED\_REALTIME\_WAKEUP,

triggerTime, repeatInterval,

notifyPendingIntent);

}

// Set the toast message for the "on" case.

toastMessage = getString(R.string.alarm\_on\_toast);

} else {

// Cancel notification if the alarm is turned off.

mNotificationManager.cancelAll();

if (alarmManager != null) {

alarmManager.cancel(notifyPendingIntent);

}

// Set the toast message for the "off" case.

toastMessage = getString(R.string.alarm\_off\_toast);

}

// Show a toast to say the alarm is turned on or off.

Toast.makeText(MainActivity.this, toastMessage,

Toast.LENGTH\_SHORT).show();

}

});

// Create the notification channel.

createNotificationChannel();

}

/\*\*

\* Creates a Notification channel, for OREO and higher.

\*/

public void createNotificationChannel() {

// Create a notification manager object.

mNotificationManager =

(NotificationManager) getSystemService(NOTIFICATION\_SERVICE);

// Notification channels are only available in OREO and higher.

// So, add a check on SDK version.

if (android.os.Build.VERSION.SDK\_INT >=

android.os.Build.VERSION\_CODES.O) {

// Create the NotificationChannel with all the parameters.

NotificationChannel notificationChannel = new NotificationChannel

(PRIMARY\_CHANNEL\_ID,

"Stand up notification",

NotificationManager.IMPORTANCE\_HIGH);

notificationChannel.enableLights(true);

notificationChannel.setLightColor(Color.RED);

notificationChannel.enableVibration(true);

notificationChannel.setDescription("Notifies every 15 minutes to " +

"stand up and walk");

mNotificationManager.createNotificationChannel(notificationChannel);

}

}

}

**AlarmReciver.java**

package com.example.standup;

import android.app.NotificationManager;

import android.app.PendingIntent;

import android.content.BroadcastReceiver;

import android.content.Context;

import android.content.Intent;

import androidx.core.app.NotificationCompat;

public class AlarmReceiver extends BroadcastReceiver {

private NotificationManager mNotificationManager;

private static final int NOTIFICATION\_ID = 0;

// Notification channel ID.

private static final String PRIMARY\_CHANNEL\_ID =

"primary\_notification\_channel";

@Override

public void onReceive(Context context, Intent intent) {

mNotificationManager = (NotificationManager)

context.getSystemService(Context.NOTIFICATION\_SERVICE);

deliverNotification(context);

}

private void deliverNotification(Context context) {

Intent contentIntent = new Intent(context, MainActivity.class);

PendingIntent contentPendingIntent = PendingIntent.getActivity

(context, NOTIFICATION\_ID, contentIntent, PendingIntent.FLAG\_UPDATE\_CURRENT);

NotificationCompat.Builder builder = new NotificationCompat.Builder(context, PRIMARY\_CHANNEL\_ID)

.setSmallIcon(R.drawable.ic\_launcher\_foreground)

.setContentTitle("Stand Up Alert")

.setContentText("You should stand up and walk around now!")

.setContentIntent(contentPendingIntent)

.setPriority(NotificationCompat.PRIORITY\_HIGH)

.setAutoCancel(true)

.setDefaults(NotificationCompat.DEFAULT\_ALL);

mNotificationManager.notify(NOTIFICATION\_ID, builder.build());

}

}



