package com.example.downloadmanagerapp;  
  
import android.app.Notification;  
import android.app.NotificationChannel;  
import android.app.NotificationManager;  
import android.app.PendingIntent;  
import android.app.Service;  
import android.content.Intent;  
import android.os.Build;  
import android.os.Environment;  
import android.os.IBinder;  
import android.util.Log;  
  
import androidx.annotation.Nullable;  
import androidx.core.app.NotificationCompat;  
import androidx.core.app.NotificationManagerCompat;  
  
import java.io.File;  
import java.io.FileOutputStream;  
import java.io.InputStream;  
import java.net.HttpURLConnection;  
import java.net.URL;  
  
public class DownloadService extends Service {  
  
 private static final String *CHANNEL\_ID* = "download\_channel";  
 private static final int *NOTIFICATION\_ID* = 1;  
 private boolean isCanceled = false;  
 private boolean isPaused = false;  
  
 @Nullable  
 @Override  
 public IBinder onBind(Intent intent) {  
 return null;  
 }  
  
 @Override  
 public int onStartCommand(Intent intent, int flags, int startId) {  
 String url = intent.getStringExtra("url");  
 if (url == null) {  
 stopSelf();  
 return *START\_NOT\_STICKY*;  
 }  
  
 createNotificationChannel();  
  
 // Notification khởi tạo (hiển thị ngay lập tức)  
 Notification notification = buildNotification("Đang chuẩn bị tải xuống...", 0);  
 startForeground(*NOTIFICATION\_ID*, notification);  
  
 new Thread(() -> {  
 try {  
 downloadFile(url);  
 } catch (Exception e) {  
 Log.*e*("DownloadService", "Lỗi tải file", e);  
 updateNotification("Lỗi tải xuống", 0);  
 } finally {  
 stopForeground(true);  
 stopSelf();  
 }  
 }).start();  
  
 return *START\_STICKY*;  
 }  
  
 private void downloadFile(String urlStr) throws Exception {  
 Log.*d*("DownloadService", "Bắt đầu tải: " + urlStr);  
 URL url = new URL(urlStr);  
 HttpURLConnection conn = (HttpURLConnection) url.openConnection();  
 conn.connect();  
  
 InputStream in = conn.getInputStream();  
  
 File outFile = new File(  
 Environment.*getExternalStoragePublicDirectory*(Environment.*DIRECTORY\_DOWNLOADS*),  
 "downloaded\_file.zip"  
 );  
 FileOutputStream out = new FileOutputStream(outFile);  
  
 int totalSize = conn.getContentLength();  
 int downloaded = 0;  
 byte[] buffer = new byte[1024];  
 int count;  
  
 while ((count = in.read(buffer)) != -1) {  
 if (isCanceled) break;  
 out.write(buffer, 0, count);  
 downloaded += count;  
 int progress = (int) ((downloaded \* 100L) / totalSize);  
 updateNotification("Đang tải...", progress);  
 }  
  
 out.close();  
 in.close();  
  
 Log.*d*("DownloadService", "Tải xong file: " + outFile.getAbsolutePath());  
 updateNotification("Hoàn tất tải xuống", 100);  
 }  
  
 private void createNotificationChannel() {  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*O*) {  
 NotificationChannel channel = new NotificationChannel(  
 *CHANNEL\_ID*,  
 "Download Manager Channel",  
 NotificationManager.*IMPORTANCE\_LOW* );  
 channel.setDescription("Hiển thị tiến trình tải xuống");  
 getSystemService(NotificationManager.class).createNotificationChannel(channel);  
 }  
 }  
  
 private Notification buildNotification(String title, int progress) {  
 // Intent mở app khi nhấn vào notification  
 Intent intent = new Intent(this, MainActivity.class);  
 PendingIntent pendingIntent = PendingIntent.*getActivity*(  
 this, 0, intent, PendingIntent.*FLAG\_IMMUTABLE* );  
  
 // Nút Cancel (hủy tải)  
 Intent cancelIntent = new Intent(this, DownloadService.class);  
 cancelIntent.putExtra("cancel", true);  
 PendingIntent cancelPendingIntent = PendingIntent.*getService*(  
 this, 1, cancelIntent, PendingIntent.*FLAG\_IMMUTABLE* );  
  
 NotificationCompat.Builder builder = new NotificationCompat.Builder(this, *CHANNEL\_ID*)  
 .setSmallIcon(android.R.drawable.*stat\_sys\_download*)  
 .setContentTitle("Đang tải xuống")  
 .setContentText(title + " (" + progress + "%)")  
 .setContentIntent(pendingIntent)  
 .setOnlyAlertOnce(true)  
 .setOngoing(true)  
 .setPriority(NotificationCompat.*PRIORITY\_LOW*)  
 .addAction(android.R.drawable.*ic\_menu\_close\_clear\_cancel*, "Hủy", cancelPendingIntent)  
 .setProgress(100, progress, false);  
  
 return builder.build();  
 }  
  
 private void updateNotification(String title, int progress) {  
 Notification notification = buildNotification(title, progress);  
 NotificationManagerCompat.*from*(this).notify(*NOTIFICATION\_ID*, notification);  
 }  
}

A screen shot of a device

AI-generated content may be incorrect.

