**LAB 7 – SERVICE & NOTIFICATION (DOWNLOAD MANAGER)**

Tên sinh viên: Phạm Thị Minh Phi\_1150070033— Lớp/MSSV: 11\_ĐH\_TMĐT

# I. Mục tiêu & Tổng quan

• Xây dựng Download Manager dùng Foreground Service để tải file.

• Hiển thị tiến trình trên Notification tùy biến; nút Pause/Resume/Cancel qua BroadcastReceiver.

• Lưu file vào Internal Storage; không cần quyền ghi ngoài.

# II. Chuẩn bị dự án

1) Tạo 3 class: DownloadActivity.java, DownloadService.java, DownloadActionReceiver.java.

2) Tạo 2 layout: res/layout/activity\_download.xml, res/layout/notification\_download.xml.

3) Thêm quyền & khai báo trong AndroidManifest.xml:

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android">  
 <uses-permission android:name="android.permission.INTERNET" />  
 <uses-permission android:name="android.permission.FOREGROUND\_SERVICE" />  
 <uses-permission android:name="android.permission.FOREGROUND\_SERVICE\_DATA\_SYNC" />  
 <uses-permission android:name="android.permission.POST\_NOTIFICATIONS" />  
 <application ...>  
 <activity android:name=".DownloadActivity" android:exported="false" />  
 <service  
 android:name=".DownloadService"  
 android:exported="false"  
 android:foregroundServiceType="dataSync" />  
 <receiver android:name=".DownloadActionReceiver" android:exported="false" />  
 </application>  
</manifest>

# III. Giao diện

## 1) activity\_download.xml

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical" android:padding="12dp"  
 android:layout\_width="match\_parent" android:layout\_height="match\_parent">  
  
 <EditText  
 android:id="@+id/et\_url"  
 android:hint="Link download (.jpg/.png/.zip/...)"  
 android:inputType="textUri"  
 android:layout\_width="match\_parent" android:layout\_height="wrap\_content"/>  
  
 <Button  
 android:id="@+id/btn\_download"  
 android:text="Download"  
 android:layout\_gravity="end"  
 android:layout\_marginTop="12dp"  
 android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"/>  
</LinearLayout>

## 2) notification\_download.xml

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical" android:padding="8dp"  
 android:layout\_width="match\_parent" android:layout\_height="wrap\_content">  
  
 <TextView android:id="@+id/tv\_title" android:text="Download manager"  
 android:textStyle="bold" android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"/>  
  
 <TextView android:id="@+id/tv\_link" android:text="Link:"  
 android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"/>  
  
 <TextView android:id="@+id/tv\_progress" android:text="Complete: 0%"  
 android:layout\_width="wrap\_content" android:layout\_height="wrap\_content"/>  
  
 <LinearLayout android:orientation="horizontal"  
 android:layout\_width="match\_parent" android:layout\_height="wrap\_content" android:layout\_marginTop="6dp">  
  
 <Button android:id="@+id/btn\_left" android:text="Pause"  
 android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1"/>  
 <Button android:id="@+id/btn\_right" android:text="Cancel"  
 android:layout\_width="0dp" android:layout\_height="wrap\_content" android:layout\_weight="1"/>  
 </LinearLayout>  
</LinearLayout>

# IV. Mã nguồn chính

## 1) DownloadActivity.java

package com.example.lab05\_multithreading;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.text.TextUtils;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
  
public class DownloadActivity extends AppCompatActivity {  
 private EditText etUrl;  
 private Button btnDownload;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_download);  
  
 etUrl = findViewById(R.id.et\_url);  
 btnDownload = findViewById(R.id.btn\_download);  
  
 btnDownload.setOnClickListener(v -> {  
 String url = etUrl.getText().toString().trim();  
 if (TextUtils.isEmpty(url)) {  
 Toast.makeText(this, "Nhập link cần download", Toast.LENGTH\_SHORT).show();  
 return;  
 }  
 Intent i = new Intent(this, DownloadService.class);  
 i.setAction(DownloadService.ACTION\_START);  
 i.putExtra(DownloadService.EXTRA\_URL, url);  
 startService(i);  
 Toast.makeText(this, "Bắt đầu download…", Toast.LENGTH\_SHORT).show();  
 });  
 }  
}

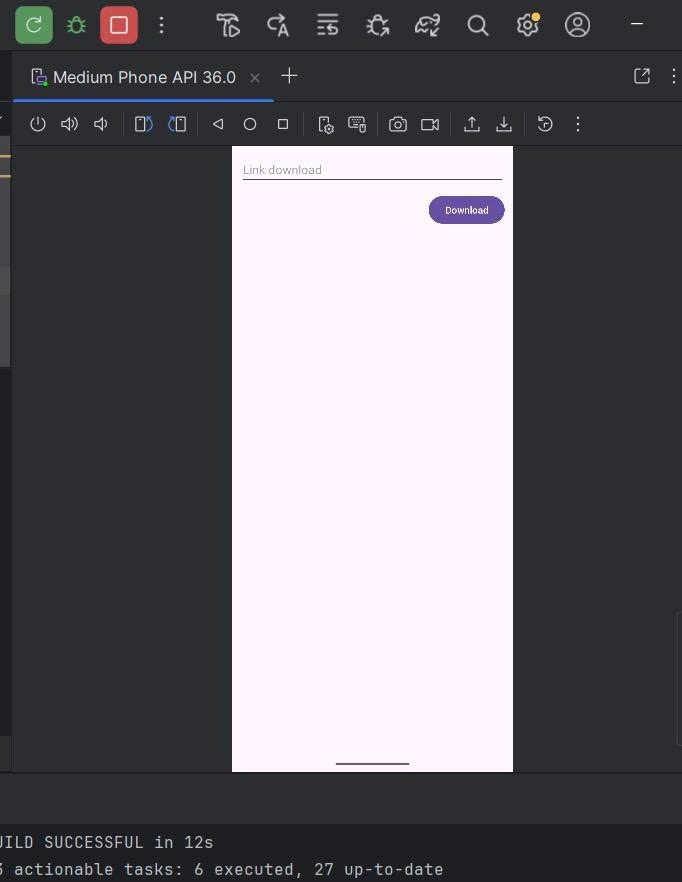
## 2) DownloadActionReceiver.java

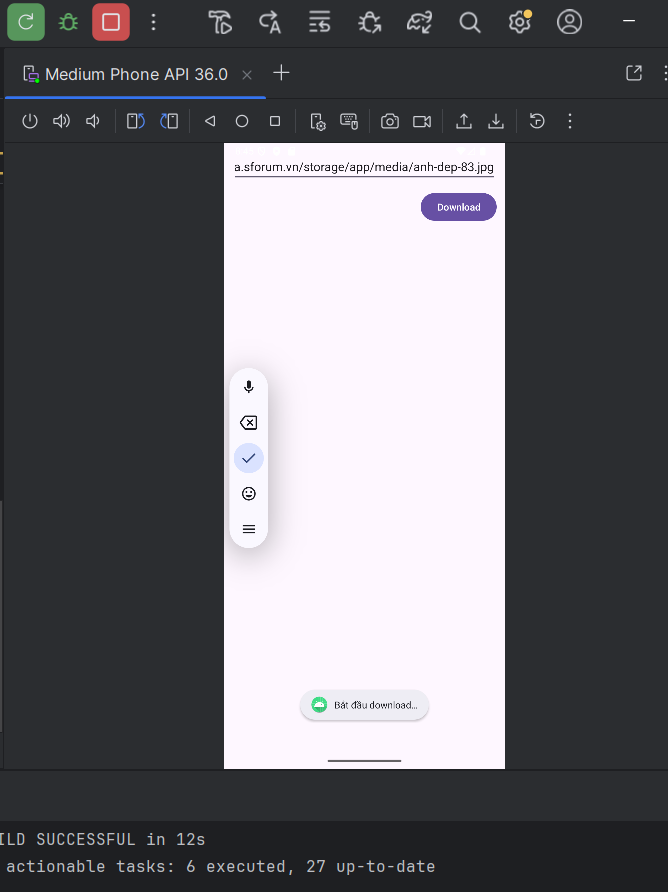
package com.example.lab05\_multithreading;  
  
import android.content.BroadcastReceiver;  
import android.content.Context;  
import android.content.Intent;  
  
public class DownloadActionReceiver extends BroadcastReceiver {  
 @Override  
 public void onReceive(Context context, Intent intent) {  
 if (intent == null || intent.getAction() == null) return;  
 Intent svc = new Intent(context, DownloadService.class);  
 svc.setAction(intent.getAction());  
 context.startService(svc);  
 }  
}

## 3) DownloadService.java

package com.example.lab05\_multithreading;  
  
import android.app.Notification;  
import android.app.NotificationChannel;  
import android.app.NotificationManager;  
import android.app.PendingIntent;  
import android.app.Service;  
import android.content.Context;  
import android.content.Intent;  
import android.os.Build;  
import android.os.IBinder;  
import android.widget.RemoteViews;  
import androidx.annotation.Nullable;  
import androidx.core.app.NotificationCompat;  
import java.io.BufferedInputStream;  
import java.io.FileOutputStream;  
import java.io.InputStream;  
import java.net.HttpURLConnection;  
import java.net.URL;  
  
public class DownloadService extends Service {  
  
 public static final String ACTION\_START = "com.example.lab05.START";  
 public static final String ACTION\_PAUSE = "com.example.lab05.PAUSE";  
 public static final String ACTION\_RESUME = "com.example.lab05.RESUME";  
 public static final String ACTION\_CANCEL = "com.example.lab05.CANCEL";  
 public static final String EXTRA\_URL = "extra\_url";  
  
 private static final String CHANNEL\_ID = "download\_channel";  
 private static final int NOTI\_ID = 17;  
  
 private volatile boolean isPaused = false;  
 private volatile boolean isCancelled = false;  
 private String currentUrl;  
 private NotificationManager nm;  
  
 @Override public void onCreate() {  
 super.onCreate();  
 nm = (NotificationManager) getSystemService(Context.NOTIFICATION\_SERVICE);  
 if (Build.VERSION.SDK\_INT >= Build.VERSION\_CODES.O) {  
 NotificationChannel ch = new NotificationChannel(CHANNEL\_ID, "Download", NotificationManager.IMPORTANCE\_LOW);  
 nm.createNotificationChannel(ch);  
 }  
 }  
  
 @Override public int onStartCommand(Intent intent, int flags, int startId) {  
 if (intent == null || intent.getAction() == null) return START\_NOT\_STICKY;  
 String action = intent.getAction();  
 if (ACTION\_START.equals(action)) {  
 currentUrl = intent.getStringExtra(EXTRA\_URL);  
 isPaused = false; isCancelled = false;  
 startForeground(NOTI\_ID, buildNotification(0, "Pause", ACTION\_PAUSE));  
 new Thread(this::downloadFile).start();  
 } else if (ACTION\_PAUSE.equals(action)) {  
 isPaused = true;  
 nm.notify(NOTI\_ID, buildNotification(0, "Resume", ACTION\_RESUME));  
 } else if (ACTION\_RESUME.equals(action)) {  
 isPaused = false;  
 nm.notify(NOTI\_ID, buildNotification(0, "Pause", ACTION\_PAUSE));  
 synchronized (this) { notifyAll(); }  
 } else if (ACTION\_CANCEL.equals(action)) {  
 isCancelled = true;  
 stopSelf();  
 }  
 return START\_NOT\_STICKY;  
 }  
  
 private void downloadFile() {  
 HttpURLConnection conn = null; FileOutputStream fos = null; InputStream is = null;  
 try {  
 URL url = new URL(currentUrl);  
 conn = (HttpURLConnection) url.openConnection();  
 conn.connect();  
 int length = conn.getContentLength();  
 is = new BufferedInputStream(conn.getInputStream());  
 fos = openFileOutput("download\_" + System.currentTimeMillis(), MODE\_PRIVATE);  
  
 byte[] buf = new byte[8192]; long total = 0; int read, last = 0;  
 while (!isCancelled && (read = is.read(buf)) != -1) {  
 synchronized (this) { while (isPaused && !isCancelled) wait(); }  
 fos.write(buf, 0, read);  
 total += read;  
 if (length > 0) {  
 int p = (int) (total \* 100 / length);  
 if (p != last) {  
 last = p;  
 nm.notify(NOTI\_ID, buildNotification(p, isPaused ? "Resume" : "Pause",  
 isPaused ? ACTION\_RESUME : ACTION\_PAUSE));  
 }  
 }  
 }  
 if (!isCancelled) nm.notify(NOTI\_ID, buildNotification(100, "Done", ACTION\_CANCEL));  
 } catch (Exception ignored) {  
 } finally {  
 try { if (fos != null) fos.close(); } catch (Exception ignore) {}  
 try { if (is != null) is.close(); } catch (Exception ignore) {}  
 if (conn != null) conn.disconnect();  
 stopForeground(false);  
 stopSelf();  
 }  
 }  
  
 private Notification buildNotification(int percent, String leftText, String leftAction) {  
 RemoteViews content = new RemoteViews(getPackageName(), R.layout.notification\_download);  
 content.setTextViewText(R.id.tv\_link, "Link: " + (currentUrl == null ? "" : currentUrl));  
 content.setTextViewText(R.id.tv\_progress, "Complete: " + percent + "%");  
 content.setTextViewText(R.id.btn\_left, leftText);  
 content.setTextViewText(R.id.btn\_right, "Cancel");  
  
 int flags = PendingIntent.FLAG\_UPDATE\_CURRENT | (Build.VERSION.SDK\_INT >= 31 ? PendingIntent.FLAG\_MUTABLE : 0);  
 PendingIntent pLeft = PendingIntent.getBroadcast(this, 1, new Intent(this, DownloadActionReceiver.class).setAction(leftAction), flags);  
 PendingIntent pRight = PendingIntent.getBroadcast(this, 2, new Intent(this, DownloadActionReceiver.class).setAction(ACTION\_CANCEL), flags);  
  
 content.setOnClickPendingIntent(R.id.btn\_left, pLeft);  
 content.setOnClickPendingIntent(R.id.btn\_right, pRight);  
  
 return new NotificationCompat.Builder(this, CHANNEL\_ID)  
 .setSmallIcon(android.R.drawable.stat\_sys\_download)  
 .setOnlyAlertOnce(true)  
 .setOngoing(percent < 100)  
 .setStyle(new NotificationCompat.DecoratedCustomViewStyle())  
 .setCustomContentView(content)  
 .setPriority(NotificationCompat.PRIORITY\_LOW)  
 .build();  
 }  
  
 @Nullable @Override public IBinder onBind(Intent intent) { return null; }  
}

# V. Kết quả chạy





# VI. Cách kiểm thử

1) Happy path: nhập URL trực tiếp (ảnh .jpg…) → Download → notification hiển thị, % tăng đến 100%, service dừng.

2) Nút Pause/Resume: dừng và tiếp tục đúng; Cancel: dừng và tắt foreground.

3) Đưa app ra nền: tiến trình vẫn chạy, quay lại không crash.

4) Trường hợp xấu: URL rỗng/sai → không crash; mất mạng giữa chừng → có thể Cancel rồi thử lại.

5) Android 13+/14: đã cấp POST\_NOTIFICATIONS; có FOREGROUND\_SERVICE\_DATA\_SYNC cho dataSync.

# VII. Ghi chú & Mẹo

• Chọn URL trực tiếp kết thúc bằng .jpg/.png/.mp3/.zip; tránh trang HTML/redirect.

• Nếu % không tăng, có thể server không gửi Content-Length; thử URL khác (Wikimedia, GitHub Raw).

• Muốn thấy file tải xong trong Device File Explorer: log ra getFilesDir() hoặc đổi sang getExternalFilesDir(null).

• Có thể thêm PendingIntent vào notification để bấm mở lại DownloadActivity.