Họ và tên: Cao Xuân Mai

MSSV: 1150070028

**LAB 7: Service, Notification và BroadCast Receiver**

CÁC BƯỚC THỰC HIỆN

**B1 Tạo Project & Khai báo quyền/Thành phần**

Mục tiêu: Khởi tạo project, thêm quyền INTERNET, khai báo Service & Receiver.

Thao tác:

* File → New → New Project → Empty Activity → đặt tên → Finish.
* Mở AndroidManifest.xml → thêm INTERNET, khai báo Service và Receiver như dưới.
* Sync/Build để chắc không lỗi.
* Mã nguồn – 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">  
  
 *<!-- Quyền -->* <uses-permission android:name="android.permission.INTERNET"/>  
 <uses-permission android:name="android.permission.POST\_NOTIFICATIONS" />  
 <uses-permission android:name="android.permission.FOREGROUND\_SERVICE" />  
 *<!-- Android 14+ nêu rõ loại foreground service cho tác vụ tải/đồng bộ -->* <uses-permission android:name="android.permission.FOREGROUND\_SERVICE\_DATA\_SYNC" />  
  
 <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.CaoXuanMaiBT7"  
  
 <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>  
  
 *<!-- Foreground Download Service -->* <service  
 android:name=".DownloadService"  
 android:exported="false"  
 android:foregroundServiceType="dataSync" />  
  
 *<!-- Receiver nhận các action từ notification -->* <receiver  
 android:name=".DownloadReceiver"  
 android:exported="true">  
 <intent-filter>  
 <action android:name="com.example.app.ACTION\_PAUSE" />  
 <action android:name="com.example.app.ACTION\_RESUME" />  
 <action android:name="com.example.app.ACTION\_CANCEL" />  
 </intent-filter>  
 </receiver>  
  
</application>  
 </manifest>

Kết quả (ảnh chụp màn hình):



**B2 Giao diện nhập URL & nút tải (MainActivity)**

Mục tiêu: Tạo UI để nhập URL và gọi Service.

Thao tác:

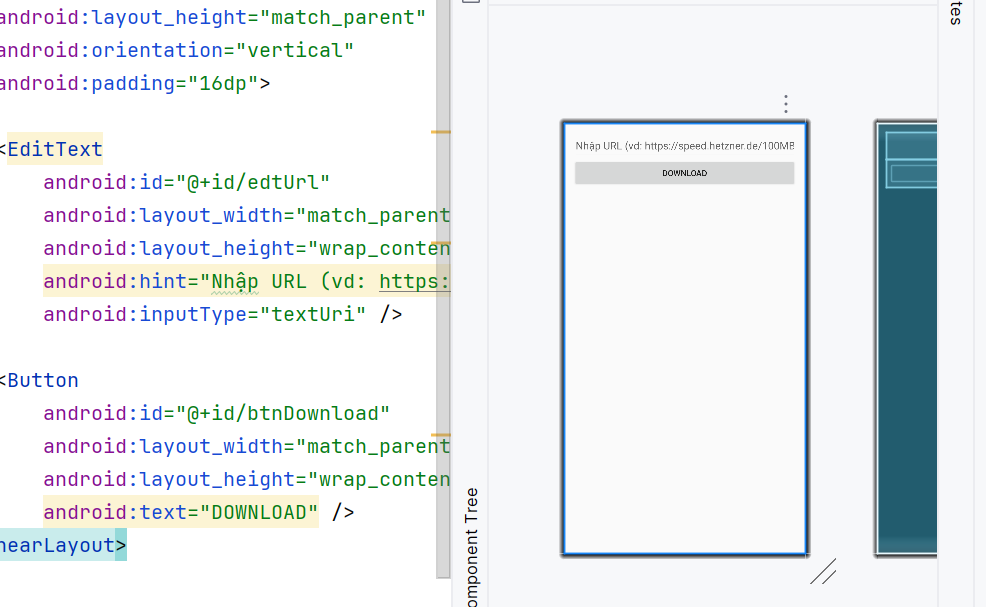
* Tạo layout res/layout/activity\_main.xml: EditText + Button.
* Viết MainActivity.java xử lý onClick → startService() với ACTION\_START.
* Run thử với URL hợp lệ.
* Mã nguồn – res/layout/activity\_main.xml:

*<?*xml version="1.0" encoding="utf-8"*?>*<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="16dp">  
  
 <EditText  
 android:id="@+id/edtUrl"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Nhập URL (vd: https://speed.hetzner.de/100MB.bin)"  
 android:inputType="textUri" />  
  
 <Button  
 android:id="@+id/btnDownload"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="DOWNLOAD" />  
</LinearLayout>

Mã nguồn – MainActivity.java:

package com.example.caoxuanmaibt7;  
  
import android.Manifest;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.os.Build;  
import android.os.Bundle;  
import android.webkit.URLUtil;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.content.ContextCompat;  
  
public class MainActivity extends AppCompatActivity {  
 EditText edtUrl;  
 Button btnDownload;  
  
 @Override protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
 edtUrl = findViewById(R.id.*edtUrl*);  
 btnDownload = findViewById(R.id.*btnDownload*);  
  
 askPostNotificationsIfNeeded();  
  
 btnDownload.setOnClickListener(v -> {  
 String url = edtUrl.getText().toString().trim();  
 if (url.isEmpty()) {  
 Toast.*makeText*(this, "Nhập URL cần tải", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
 String fileName = URLUtil.*guessFileName*(url, null, null);  
  
 Intent i = new Intent(this, DownloadService.class);  
 i.setAction(DownloadService.ACTION\_START);  
 i.putExtra(DownloadService.EXTRA\_URL, url);  
 i.putExtra(DownloadService.EXTRA\_FILE\_NAME, fileName);  
  
 *// Android 8+ dùng startForegroundService* ContextCompat.*startForegroundService*(this, i);  
 Toast.*makeText*(this, "Bắt đầu tải: " + fileName, Toast.*LENGTH\_SHORT*).show();  
 });  
 }  
  
 private void askPostNotificationsIfNeeded() {  
 if (Build.VERSION.*SDK\_INT* >= 33) {  
 if (checkSelfPermission(Manifest.permission.*POST\_NOTIFICATIONS*)  
 != PackageManager.*PERMISSION\_GRANTED*) {  
 requestPermissions(new String[]{Manifest.permission.*POST\_NOTIFICATIONS*}, 100);  
 }  
 }  
 }  
}

Kết quả:



**B3. Tạo Notification tùy biến (RemoteViews) + tiện ích Channel**

Mục tiêu: Hiển thị tiến độ & nút Pause/Resume/Cancel.

Thao tác:

* Tạo res/layout/notification\_download.xml.
* Viết NotificationUtils.java để tạo channel & build notification.
* Service gọi NotificationUtils.build() cập nhật tiến độ.
* Mã nguồn – res/layout/notification\_download.xml:

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:padding="12dp">

<TextView

android:id="@+id/txtTitle"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Đang tải..."

android:textStyle="bold"

android:textSize="14sp"/>

<ProgressBar

android:id="@+id/prog"

style="?android:attr/progressBarStyleHorizontal"

android:layout\_width="match\_parent"

android:layout\_height="8dp"

android:layout\_below="@id/txtTitle"

android:layout\_marginTop="8dp"

android:max="100"

android:progress="0"/>

<LinearLayout

android:id="@+id/actions"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@id/prog"

android:layout\_marginTop="8dp"

android:orientation="horizontal">

<ImageButton

android:id="@+id/btnPause"

android:layout\_width="36dp"

android:layout\_height="36dp"

android:contentDescription="Pause"

android:src="@android:drawable/ic\_media\_pause"/>

<ImageButton

android:id="@+id/btnResume"

android:layout\_width="36dp"

android:layout\_height="36dp"

android:contentDescription="Resume"

android:src="@android:drawable/ic\_media\_play"/>

<ImageButton

android:id="@+id/btnCancel"

android:layout\_width="36dp"

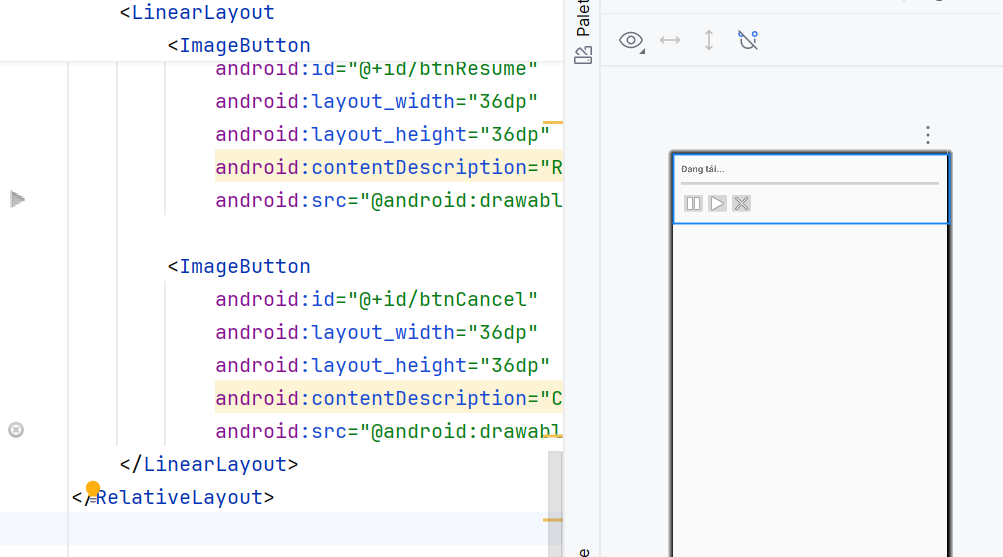
android:layout\_height="36dp"

android:contentDescription="Cancel"

android:src="@android:drawable/ic\_menu\_close\_clear\_cancel"/>

</LinearLayout>

</RelativeLayout>



Mã nguồn – NotificationUtils.java:

package com.example.caoxuanmaibt7;  
  
import android.app.Notification;  
import android.app.NotificationChannel;  
import android.app.NotificationManager;  
import android.app.PendingIntent;  
import android.content.Context;  
import android.content.Intent;  
import android.os.Build;  
import android.widget.RemoteViews;  
  
import androidx.core.app.NotificationCompat;  
  
public class NotificationUtils {  
 public static final String *CHANNEL\_ID* = "download\_channel";  
 public static final int *NOTIF\_ID* = 1001;  
  
 public static void createChannel(Context ctx) {  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*O*) {  
 NotificationChannel ch = new NotificationChannel(  
 *CHANNEL\_ID*, "Downloads", NotificationManager.*IMPORTANCE\_LOW*);  
 ch.setDescription("Foreground download");  
 NotificationManager nm = ctx.getSystemService(NotificationManager.class);  
 if (nm != null) nm.createNotificationChannel(ch);  
 }  
 }  
  
 public static Notification build(Context ctx, String title, int progress, boolean indeterminate) {  
 RemoteViews rv = new RemoteViews(ctx.getPackageName(), R.layout.notification\_download);  
 rv.setTextViewText(R.id.txtTitle, title);  
 rv.setProgressBar(R.id.prog, 100, progress, indeterminate);  
  
 *// PendingIntent cho 3 nút điều khiển* Intent pause = new Intent("com.example.app.ACTION\_PAUSE");  
 PendingIntent pPause = PendingIntent.*getBroadcast*(ctx, 1, pause,  
 PendingIntent.*FLAG\_UPDATE\_CURRENT* | PendingIntent.*FLAG\_IMMUTABLE*);  
 rv.setOnClickPendingIntent(R.id.btnPause, pPause);  
  
 Intent resume = new Intent("com.example.app.ACTION\_RESUME");  
 PendingIntent pResume = PendingIntent.*getBroadcast*(ctx, 2, resume,  
 PendingIntent.*FLAG\_UPDATE\_CURRENT* | PendingIntent.*FLAG\_IMMUTABLE*);  
 rv.setOnClickPendingIntent(R.id.btnResume, pResume);  
  
 Intent cancel = new Intent("com.example.app.ACTION\_CANCEL");  
 PendingIntent pCancel = PendingIntent.*getBroadcast*(ctx, 3, cancel,  
 PendingIntent.*FLAG\_UPDATE\_CURRENT* | PendingIntent.*FLAG\_IMMUTABLE*);  
 rv.setOnClickPendingIntent(R.id.btnCancel, pCancel);  
  
 return new NotificationCompat.Builder(ctx, *CHANNEL\_ID*)  
 .setSmallIcon(android.R.drawable.*stat\_sys\_download*)  
 .setCustomContentView(rv)  
 .setOngoing(true)  
 .setOnlyAlertOnce(true)  
 .setPriority(NotificationCompat.*PRIORITY\_LOW*)  
 .build();  
 }  
}

Kết quả:

Chèn ảnh notification với progress + 3 nút.

**B4. BroadcastReceiver cho Pause/Resume/Cancel**

Mục tiêu: Nhận sự kiện từ notification, chuyển tiếp action tới Service.

Thao tác: Tạo lớp DownloadReceiver và onReceive() map action → startService().

Mã nguồn – DownloadReceiver.java:

package com.example.caoxuanmaibt7;  
  
import android.content.BroadcastReceiver;  
import android.content.Context;  
import android.content.Intent;  
  
public class DownloadReceiver extends BroadcastReceiver {  
 @Override  
 public void onReceive(Context context, Intent intent) {  
 String action = intent.getAction();  
 Intent svc = new Intent(context, DownloadService.class);  
 if ("com.example.app.ACTION\_PAUSE".equals(action)) {  
 svc.setAction(DownloadService.*ACTION\_PAUSE*);  
 } else if ("com.example.app.ACTION\_RESUME".equals(action)) {  
 svc.setAction(DownloadService.*ACTION\_RESUME*);  
 } else if ("com.example.app.ACTION\_CANCEL".equals(action)) {  
 svc.setAction(DownloadService.*ACTION\_CANCEL*);  
 } else {  
 return;  
 }  
 context.startService(svc);  
 }  
}

**B.5. DownloadService (Foreground) – tải file + Pause/Resume/Cancel**

Mục tiêu: Viết Service foreground tải file bằng HttpURLConnection, Range cho resume, cập nhật notification.

Thao tác:

* Định nghĩa ACTION\_START/PAUSE/RESUME/CANCEL.
* Dùng Thread + AtomicBoolean cho trạng thái.
* Lưu file vào getExternalFilesDir(Downloads).
* Mã nguồn – DownloadService.java:

package com.example.caoxuanmaibt7;  
  
import android.app.Notification;  
import android.app.Service;  
import android.content.Intent;  
import android.os.Binder;  
import android.os.IBinder;  
import android.util.Log;  
  
import androidx.annotation.Nullable;  
  
import java.io.BufferedInputStream;  
import java.io.File;  
import java.io.InputStream;  
import java.io.RandomAccessFile;  
import java.net.HttpURLConnection;  
import java.net.URL;  
import java.util.concurrent.atomic.AtomicBoolean;  
  
public class DownloadService extends Service {  
 public static final String *ACTION\_START* = "ACTION\_START";  
 public static final String *ACTION\_PAUSE* = "ACTION\_PAUSE";  
 public static final String *ACTION\_RESUME* = "ACTION\_RESUME";  
 public static final String *ACTION\_CANCEL* = "ACTION\_CANCEL";  
  
 public static final String *EXTRA\_URL* = "EXTRA\_URL";  
 public static final String *EXTRA\_FILE\_NAME* = "EXTRA\_FILE\_NAME";  
  
 private final AtomicBoolean isPaused = new AtomicBoolean(false);  
 private final AtomicBoolean isCanceled = new AtomicBoolean(false);  
  
 private String urlStr;  
 private String fileName;  
 private File outFile;  
 private long downloadedBytes = 0;  
 private long totalBytes = -1;  
  
 private Thread worker;  
  
 @Nullable @Override public IBinder onBind(Intent intent) { return new Binder(); }  
  
 @Override public int onStartCommand(Intent intent, int flags, int startId) {  
 if (intent == null || intent.getAction() == null) return *START\_NOT\_STICKY*;  
 String action = intent.getAction();  
  
 switch (action) {  
 case *ACTION\_START*:  
 urlStr = intent.getStringExtra(*EXTRA\_URL*);  
 fileName = intent.getStringExtra(*EXTRA\_FILE\_NAME*);  
 isPaused.set(false);  
 isCanceled.set(false);  
 startForegroundFlow();  
 startDownload(false);  
 break;  
  
 case *ACTION\_RESUME*:  
 isPaused.set(false);  
 startDownload(true);  
 break;  
  
 case *ACTION\_PAUSE*:  
 isPaused.set(true);  
 break;  
  
 case *ACTION\_CANCEL*:  
 isCanceled.set(true);  
 stopSelf();  
 break;  
 }  
 return *START\_NOT\_STICKY*;  
 }  
  
 private void startForegroundFlow() {  
 NotificationUtils.*createChannel*(this);  
 Notification n = NotificationUtils.*build*(this, "Đang tải: " + fileName, 0, true);  
 startForeground(NotificationUtils.*NOTIF\_ID*, n); *// gọi sớm (≤5s)* }  
  
 private void updateNotification(int progress, boolean indeterminate) {  
 Notification n = NotificationUtils.*build*(this, "Đang tải: " + fileName, progress, indeterminate);  
 startForeground(NotificationUtils.*NOTIF\_ID*, n); *// cập nhật ổn định trên nhiều API* }  
  
 private void startDownload(boolean resume) {  
 if (worker != null && worker.isAlive()) return;  
  
 worker = new Thread(() -> {  
 try {  
 *// Thư mục internal storage: .../files/downloads* if (outFile == null) {  
 File dir = new File(getFilesDir(), "downloads");  
 if (!dir.exists()) dir.mkdirs();  
 outFile = new File(dir, fileName);  
 }  
 if (!resume) downloadedBytes = 0;  
  
 RandomAccessFile raf = new RandomAccessFile(outFile, "rw");  
 if (resume) {  
 downloadedBytes = outFile.length();  
 raf.seek(downloadedBytes);  
 }  
  
 URL url = new URL(urlStr);  
 HttpURLConnection conn = (HttpURLConnection) url.openConnection();  
 conn.setConnectTimeout(15000);  
 conn.setReadTimeout(20000);  
 if (downloadedBytes > 0) {  
 conn.setRequestProperty("Range", "bytes=" + downloadedBytes + "-");  
 }  
 conn.connect();  
  
 int code = conn.getResponseCode();  
 if (code == HttpURLConnection.*HTTP\_OK* || code == HttpURLConnection.*HTTP\_PARTIAL*) {  
 long contentLen = conn.getContentLengthLong();  
 if (code == HttpURLConnection.*HTTP\_OK*) {  
 totalBytes = contentLen;  
 } else if (code == HttpURLConnection.*HTTP\_PARTIAL*) {  
 String cr = conn.getHeaderField("Content-Range"); *// "bytes X-Y/TOTAL"* long total = -1;  
 if (cr != null && cr.contains("/")) {  
 try { total = Long.*parseLong*(cr.substring(cr.lastIndexOf('/') + 1)); }  
 catch (Exception ignored) {}  
 }  
 totalBytes = (total > 0) ? total : downloadedBytes + contentLen;  
 }  
  
 InputStream in = new BufferedInputStream(conn.getInputStream());  
 byte[] buf = new byte[8192];  
 long lastUi = System.*currentTimeMillis*();  
 int len;  
  
 while ((len = in.read(buf)) != -1) {  
 if (isCanceled.get()) {  
 in.close();  
 raf.close();  
 outFile.delete();  
 stopSelf();  
 return;  
 }  
 while (isPaused.get()) {  
 try { Thread.*sleep*(200); } catch (InterruptedException ignored) {}  
 }  
 raf.write(buf, 0, len);  
 downloadedBytes += len;  
  
 long now = System.*currentTimeMillis*();  
 if (now - lastUi > 500) {  
 int progress = (totalBytes > 0) ? (int)((downloadedBytes \* 100) / totalBytes) : 0;  
 updateNotification(progress, totalBytes <= 0);  
 lastUi = now;  
 }  
 }  
 in.close();  
 raf.close();  
  
 updateNotification(100, false);  
 stopForeground(true);  
 stopSelf();  
 } else {  
 stopForeground(true);  
 stopSelf();  
 }  
 } catch (Exception e) {  
 Log.*e*("DownloadService", "Lỗi: " + e.getMessage(), e);  
 stopForeground(true);  
 stopSelf();  
 }  
 });  
 worker.start();  
 }  
  
 @Override public void onDestroy() {  
 super.onDestroy();  
 isCanceled.set(true);  
 }  
}

Kết quả:

Chèn ảnh notification đang chạy, lúc pause, lúc resume, lúc cancel.

**Liên kết & Kiểm thử chức năng**

Mục tiêu: Chạy thử toàn bộ flow.

Thao tác kiểm thử:

Mở app → dán URL: https://speed.hetzner.de/100MB.bin (hoặc file khác).

Nhấn BẮT ĐẦU TẢI → notification hiển thị tiến độ.

Thử Pause → Resume → Cancel.

Kiểm tra file tại: Android/data/<package>/files/Download.

Nếu lỗi, mở Logcat xem stacktrace của DownloadService.

Kết quả: