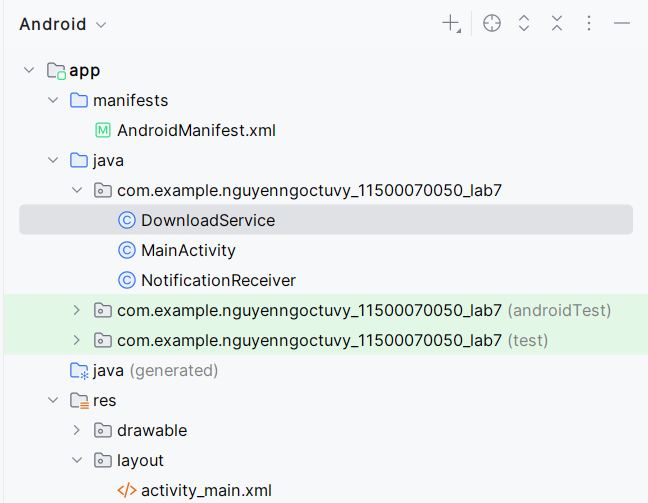
**MSSV: 1150070050  
Họ và tên: Nguyễn Ngọc Tú Vy**

**Lab7 Service Notification và BroadCast Receiver**

**Cấu trúc file thực hiện bài**

****

**Khai báo các quyền cần có trong bài**

*<!-- Quyền truy cập internet và ghi bộ nhớ -->*

<uses-permission android:name="android.permission.INTERNET"/>

<uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE"/>

<uses-permission android:name="android.permission.READ\_EXTERNAL\_STORAGE"/>

<uses-permission android:name="android.permission.POST\_NOTIFICATIONS"/>

<uses-permission android:name="android.permission.FOREGROUND\_SERVICE" />

### **Luồng hoạt động**

* Người dùng nhập link và nhấn Download.
* MainActivity gửi Intent đến DownloadService.
* DownloadService chạy Foreground Service, tạo Notification hiển thị tiến trình.
* Khi người dùng bấm Pause/Resume/Cancel, NotificationReceiver xử lý và điều khiển Service.
* File tải về được lưu trong bộ nhớ.

## **Các thành phần chính**

* Service:
  + Extend từ Service.
  + Dùng HttpURLConnection để kết nối và tải file.
  + Chạy dưới dạng Foreground Service (có foregroundServiceType).
* Notification:
  + Dùng NotificationCompat để tạo Notification tùy chỉnh.
  + Hiển thị tiến trình tải (%).
  + Thêm Action button (Pause, Resume, Cancel).

Cấu trúc code

@Override

public void onReceive(Context context, Intent intent) {

String action = intent.getAction();

DownloadService service = new DownloadService();

if ("PAUSE".equals(action)) {

service.pauseDownload();

} else if ("RESUME".equals(action)) {

service.resumeDownload();

} else if ("CANCEL".equals(action)) {

service.cancelDownload();

}

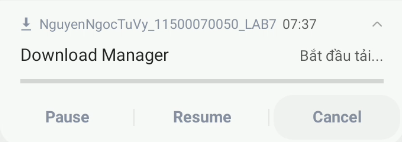
}

Khai báo thông báo

<receiver

android:name=".NotificationReceiver"

android:exported="false"/>



* BroadcastReceiver:
  + Nhận các action từ Notification.
  + Gọi method tương ứng trong DownloadService để dừng, tiếp tục hoặc hủy tải.

Code download  
@Override

public int onStartCommand(Intent intent, int flags, int startId) {

String fileUrl = intent.getStringExtra("url");

createNotificationChannel();

startForeground(*NOTIFY\_ID*, buildNotification("Bắt đầu tải..."));

new Thread(() -> downloadFile(fileUrl)).start();

return *START\_NOT\_STICKY*;

}

private void downloadFile(String fileUrl) {

try {

URL url = new URL(fileUrl);

HttpURLConnection conn = (HttpURLConnection) url.openConnection();

conn.connect();

int length = conn.getContentLength();

InputStream input = new BufferedInputStream(conn.getInputStream());

String fileName = "downloaded\_file\_" + System.*currentTimeMillis*() + ".bin";

FileOutputStream output = new FileOutputStream(

getExternalFilesDir(Environment.*DIRECTORY\_DOWNLOADS*) + "/" + fileName);

byte[] data = new byte[1024];

int total = 0;

int count;

while ((count = input.read(data)) != -1) {

if (isCanceled) {

stopSelf();

return;

}

while (isPaused) {

Thread.*sleep*(500);

}

total += count;

output.write(data, 0, count);

int progress = (int) ((total \* 100L) / length);

updateNotification("Đang tải: " + progress + "%", progress);

}

output.flush();

output.close();

input.close();

updateNotification("Hoàn tất!", 100);

} catch (Exception e) {

e.printStackTrace();

}

}

private Notification buildNotification(String text) {

Intent pauseIntent = new Intent(this, NotificationReceiver.class);

pauseIntent.setAction("PAUSE");

Intent resumeIntent = new Intent(this, NotificationReceiver.class);

resumeIntent.setAction("RESUME");

Intent cancelIntent = new Intent(this, NotificationReceiver.class);

cancelIntent.setAction("CANCEL");

return new NotificationCompat.Builder(this, *CHANNEL\_ID*)

.setContentTitle("Download Manager")

.setContentText(text)

.setSmallIcon(android.R.drawable.*stat\_sys\_download*)

.addAction(android.R.drawable.*ic\_media\_pause*, "Pause",

PendingIntent.*getBroadcast*(this, 0, pauseIntent, PendingIntent.*FLAG\_UPDATE\_CURRENT*))

.addAction(android.R.drawable.*ic\_media\_play*, "Resume",

PendingIntent.*getBroadcast*(this, 1, resumeIntent, PendingIntent.*FLAG\_UPDATE\_CURRENT*))

.addAction(android.R.drawable.*ic\_menu\_close\_clear\_cancel*, "Cancel",

PendingIntent.*getBroadcast*(this, 2, cancelIntent, PendingIntent.*FLAG\_UPDATE\_CURRENT*))

.setOnlyAlertOnce(true)

.setProgress(100, 0, false)

.build();

}

private void updateNotification(String text, int progress) {

NotificationManager manager = (NotificationManager) getSystemService(*NOTIFICATION\_SERVICE*);

Notification notification = new NotificationCompat.Builder(this, *CHANNEL\_ID*)

.setContentTitle("Download Manager")

.setContentText(text)

.setSmallIcon(android.R.drawable.*stat\_sys\_download\_done*)

.setProgress(100, progress, false)

.build();

manager.notify(*NOTIFY\_ID*, notification);

}

private void createNotificationChannel() {

if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*O*) {

NotificationChannel channel = new NotificationChannel(

*CHANNEL\_ID*, "Download Service", NotificationManager.*IMPORTANCE\_LOW*);

NotificationManager manager = getSystemService(NotificationManager.class);

manager.createNotificationChannel(channel);

}

}

**Layout giao diện**

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

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

android:orientation="vertical"

android:padding="16dp"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<EditText

android:id="@+id/edtUrl"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Nhập link tải file"

android:minHeight="48dp" />

<Button

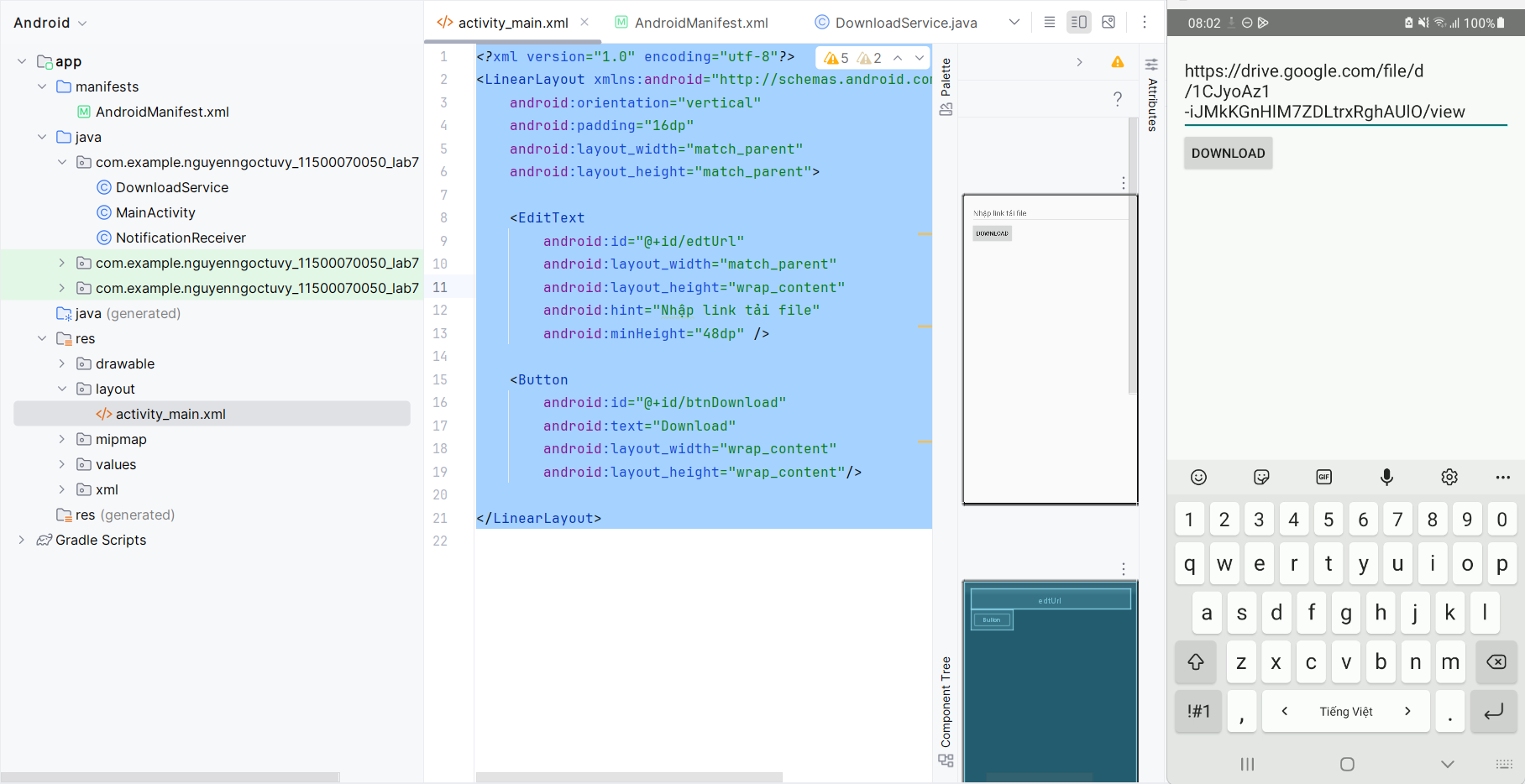
android:id="@+id/btnDownload"

android:text="Download"

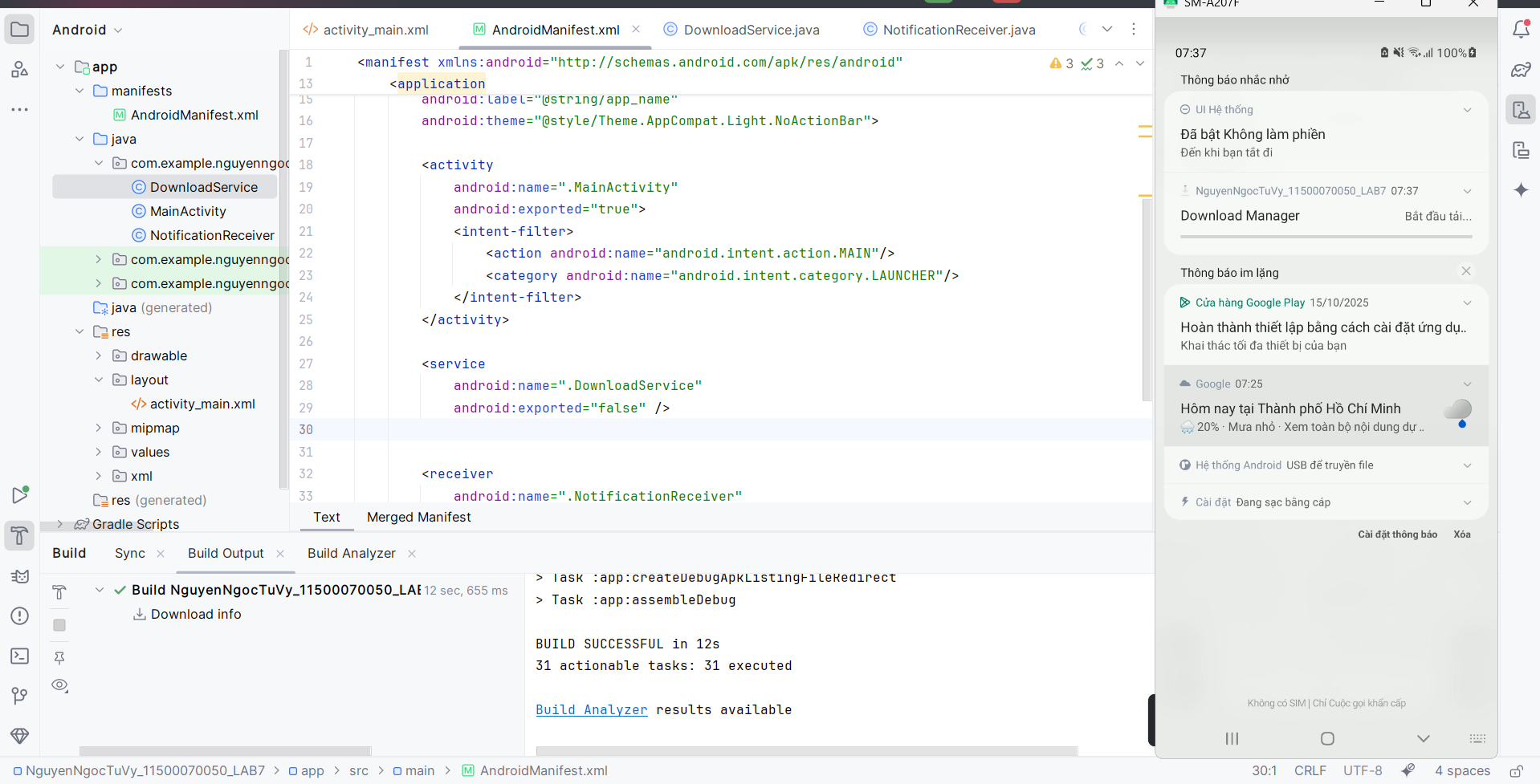
android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

</LinearLayout>

****

**Kết quả**

****