Họ tên: Hồ Sỹ Thịnh

MSSV:1150070039

# BÁO CÁO LAB 7 – SERVICE, NOTIFICATION, BROADCAST RECEIVER

Đề tài: Ứng dụng Download Manager (LAB7\_DownloadManager\_Project\_v4)

Ngôn ngữ: Kotlin – Android Studio

## 1. Mục tiêu bài Lab

Xây dựng ứng dụng Android cho phép tải file từ Internet bằng Foreground Service,  
hiển thị tiến trình tải qua Notification tùy biến (có ProgressBar, hiển thị link, phần trăm hoàn thành và các nút Pause/Resume/Cancel),  
và quản lý tác vụ thông qua BroadcastReceiver.

## 2. Kiến trúc và thành phần chính

Cấu trúc thư mục ứng dụng:  
app/  
 ├─ java/com/example/downloader/  
 │ ├─ MainActivity.kt  
 │ ├─ DownloadService.kt  
 │ ├─ DownloadActionReceiver.kt  
 │ ├─ NotificationHelper.kt  
 └─ res/  
 ├─ layout/activity\_main.xml  
 ├─ layout/notification\_download.xml  
 ├─ drawable/  
 ├─ values/

## 3. Chức năng chính

• Nhập URL file cần tải từ EditText trong MainActivity.  
• Khi nhấn “Download”, ứng dụng khởi động Foreground Service.  
• Service tải dữ liệu qua HttpURLConnection, lưu file vào Internal Storage.  
• Notification hiển thị tiến độ, link và phần trăm hoàn thành.  
• Người dùng có thể Pause, Resume hoặc Cancel quá trình tải thông qua BroadcastReceiver.  
• Khi tải xong, Notification hiển thị trạng thái “Hoàn tất”.

## 4. Các file và vai trò

4.1. MainActivity.kt – Giao diện chính và khởi chạy Service

- Cho phép người dùng nhập đường dẫn (URL) cần tải.  
- Gọi DownloadService.start(context, url) khi nhấn nút Download.  
- Yêu cầu quyền POST\_NOTIFICATIONS trên Android 13 trở lên.

4.2. DownloadService.kt – Foreground Service xử lý tải file

- Sử dụng Coroutine để tải dữ liệu trên luồng nền.  
- Hỗ trợ Resume với HTTP Range header.  
- Cập nhật Notification định kỳ theo tiến độ tải.  
- Cho phép Pause/Resume/Cancel và dừng Foreground Service khi hoàn tất.

4.3. NotificationHelper.kt – Tạo Notification tùy biến

- Tạo Notification Channel (API 26+).  
- Xây dựng giao diện RemoteViews (custom layout) gồm ProgressBar, Link, % hoàn thành, các nút điều khiển.  
- Cung cấp các PendingIntent để gửi Broadcast khi nhấn Pause, Resume, Cancel.

4.4. DownloadActionReceiver.kt – Nhận và xử lý hành động người dùng

- Lắng nghe các Intent ACTION\_PAUSE, ACTION\_RESUME, ACTION\_CANCEL từ Notification.  
- Gọi DownloadService.sendAction(context, action) để chuyển hành động về Service.

## 5. Quy trình hoạt động

1. Người dùng mở app và nhập URL tải file.  
2. Nhấn nút Download → MainActivity gọi startForegroundService.  
3. DownloadService khởi tạo Notification, bắt đầu tải dữ liệu.  
4. Trong quá trình tải, Notification hiển thị tiến độ (% và link).  
5. Nếu nhấn Pause → Service dừng đọc dữ liệu (paused = true).  
6. Nếu nhấn Resume → Service tiếp tục tải với Range header.  
7. Nếu nhấn Cancel → Service hủy Job, xóa file, dừng Service.  
8. Khi hoàn tất → Notification cập nhật “Hoàn tất: <tên file>”.

## 6. Kết quả kiểm thử

- Ứng dụng hoạt động ổn định trên Android 10–14.  
- Test với các URL:  
 • https://speed.hetzner.de/10MB.bin  
 • https://ipv4.download.thinkbroadband.com/100MB.zip  
 • https://www.w3.org/WAI/ER/tests/xhtml/testfiles/resources/pdf/dummy.pdf  
- Chức năng Pause/Resume hoạt động tốt nếu server hỗ trợ HTTP 206 (Partial Content).  
- Notification hiển thị rõ ràng tiến độ, link, phần trăm, và các nút điều khiển.

## 7. Một số lỗi thường gặp và cách khắc phục

• App crash khi bấm Download: dùng ContextCompat.startForegroundService() thay vì startService().  
• Notification không hiển thị: cần xin quyền POST\_NOTIFICATIONS (Android 13+).  
• Lỗi Resource Linking: kiểm tra file themes.xml, ic\_launcher.xml, và layout XML.

## 8. Kết luận

Bài Lab 7 giúp sinh viên hiểu rõ cơ chế Foreground Service, Notification tùy biến và BroadcastReceiver.  
Thông qua bài tập, người học nắm được cách quản lý tiến trình chạy nền trong Android,  
xử lý thread an toàn bằng Coroutine và đảm bảo trải nghiệm người dùng với notification tương tác.

## 9. Hướng phát triển

• Bổ sung tính năng hiển thị tốc độ tải (KB/s) và thời gian ước lượng còn lại.  
• Cho phép chọn thư mục lưu thông qua Storage Access Framework (SAF).  
• Lưu danh sách file đã tải và cho phép mở/xóa sau khi tải xong.  
• Kết hợp WorkManager để tự động tiếp tục tải khi app bị đóng.