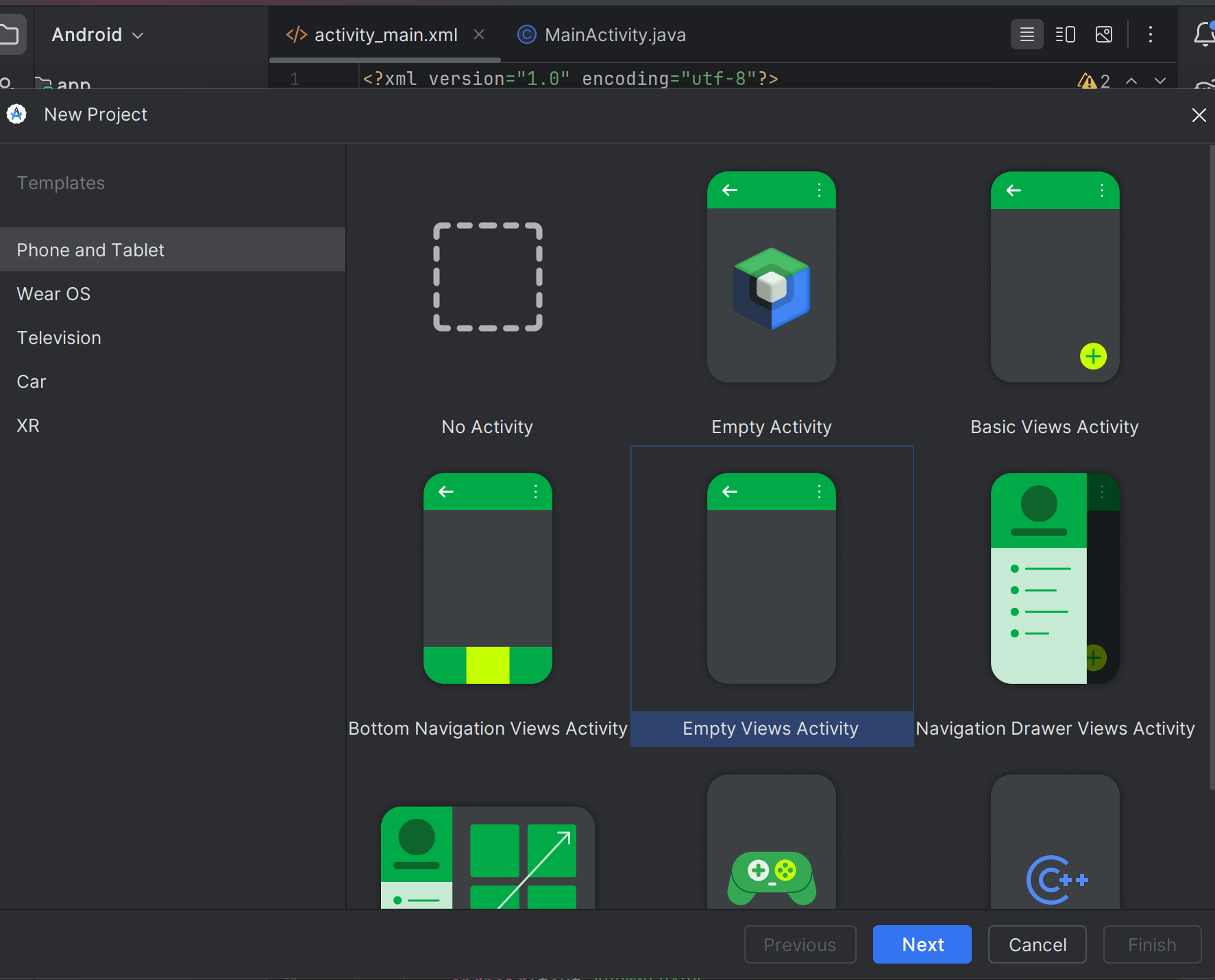
**BÀI TẬP LÝ THUYẾT LAB 7**

**Bước 1:** Tạo **project** mới – đặt tên là **Lab7**

****

**Bước 2:** Ở mục **activity\_main.xml** viết code như sau:

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:app="http://schemas.android.com/apk/res-auto"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:orientation="vertical"  
 android:padding="16dp"  
 tools:context=".MainActivity">  
  
 <com.google.android.material.textfield.TextInputLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox">  
  
 <com.google.android.material.textfield.TextInputEditText  
 android:id="@+id/etLinkDownload"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Link download"  
 tools:ignore="HardcodedText" />  
  
 </com.google.android.material.textfield.TextInputLayout>  
  
 <Button  
 android:id="@+id/btnDownload"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_gravity="end"  
 android:layout\_marginTop="8dp"  
 android:text="DOWNLOAD"  
 tools:ignore="HardcodedText" />  
  
</LinearLayout>

**Bước 3:** Ở file **AndroidManifest.xml** thêm các khai báo

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

**Bước 4:** Ở **com.example.lab7 –** chọn **New – Service – Service-** đặt tên **DownloadService – Finish**

**Ảnh có chứa văn bản, ảnh chụp màn hình, phần mềm, Phần mềm đa phương tiện

Nội dung do AI tạo ra có thể không chính xác.**

**Bước 5:** Ở file **AndroidManifest.xml** code như sau

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools">  
  
 <uses-permission android:name="android.permission.INTERNET" />  
 <uses-permission android:name="android.permission.FOREGROUND\_SERVICE"  
 tools:ignore="ForegroundServicesPolicy" />  
 <uses-permission android:name="android.permission.POST\_NOTIFICATIONS" />  
  
 <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.Lab7">  
 <service  
 android:name=".DownloadService"  
 android:exported="false" />  
  
 <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>  
 </application>  
  
</manifest>

**Bước 6:** Ở **MainActivity.java** code như sau

package com.example.lab7;  
  
import android.Manifest;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.os.Build;  
import android.os.Bundle;  
import android.util.Patterns;  
import android.view.View;  
import android.widget.Button;  
import android.widget.Toast;  
  
import androidx.activity.result.ActivityResultLauncher;  
import androidx.activity.result.contract.ActivityResultContracts;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.content.ContextCompat;  
  
import com.google.android.material.textfield.TextInputEditText;  
  
public class MainActivity extends AppCompatActivity {  
  
 private TextInputEditText etLinkDownload;  
 private Button btnDownload;  
  
 // Trình xử lý yêu cầu quyền (permission handler)  
 private final ActivityResultLauncher<String> requestPermissionLauncher =  
 registerForActivityResult(new ActivityResultContracts.RequestPermission(), isGranted -> {  
 if (isGranted) {  
 // Quyền đã được cấp, bắt đầu download  
 startDownload();  
 } else {  
 // Người dùng từ chối cấp quyền  
 Toast.*makeText*(this, "Permission denied to post notifications", Toast.*LENGTH\_SHORT*).show();  
 }  
 });  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 // Ánh xạ view từ layout  
 etLinkDownload = findViewById(R.id.*etLinkDownload*);  
 btnDownload = findViewById(R.id.*btnDownload*);  
  
 // Gán sự kiện click cho nút download  
 btnDownload.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View v) {  
 askNotificationPermission();  
 }  
 });  
 }  
  
 private void askNotificationPermission() {  
 // Chỉ cần yêu cầu quyền trên Android 13 (API 33) trở lên  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*TIRAMISU*) {  
 if (ContextCompat.*checkSelfPermission*(this, Manifest.permission.*POST\_NOTIFICATIONS*) ==  
 PackageManager.*PERMISSION\_GRANTED*) {  
 // Quyền đã có, bắt đầu download  
 startDownload();  
 } else {  
 // Yêu cầu quyền  
 requestPermissionLauncher.launch(Manifest.permission.*POST\_NOTIFICATIONS*);  
 }  
 } else {  
 // Không cần quyền cho các phiên bản thấp hơn, bắt đầu download luôn  
 startDownload();  
 }  
 }  
  
 private void startDownload() {  
 String url = "";  
 if (etLinkDownload.getText() != null) {  
 url = etLinkDownload.getText().toString().trim();  
 }  
  
 if (!url.isEmpty() && Patterns.*WEB\_URL*.matcher(url).matches()) {  
 // Tạo Intent để khởi chạy Service  
 Intent serviceIntent = new Intent(this, DownloadService.class);  
 // Gửi link download qua cho Service  
 serviceIntent.putExtra("URL", url);  
  
 // Bắt đầu Service  
 startService(serviceIntent);  
 Toast.*makeText*(this, "Download started...", Toast.*LENGTH\_SHORT*).show();  
 } else {  
 Toast.*makeText*(this, "Please enter a valid download link", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
}

**Bước 7:** Ở **layout –** tạo file mới **–** đặt tên **notification\_layout.xml**

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Nội dung do AI tạo ra có thể không chính xác.**

**\* viết code như sau:**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 xmlns:tools="http://schemas.android.com/tools"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="vertical"  
 android:padding="8dp">  
  
 <TextView  
 android:id="@+id/tvLink"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Link: ..."  
 android:ellipsize="end"  
 android:maxLines="1"  
 android:textColor="?android:attr/textColorPrimary"  
 tools:ignore="HardcodedText" />  
  
 <TextView  
 android:id="@+id/tvProgress"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="4dp"  
 android:text="Complete: 0%"  
 android:textColor="?android:attr/textColorSecondary"  
 tools:ignore="HardcodedText" />  
  
 <ProgressBar  
 android:id="@+id/progressBar"  
 style="?android:attr/progressBarStyleHorizontal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="8dp"  
 android:max="100"  
 android:progress="0" />  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginTop="8dp"  
 android:orientation="horizontal">  
  
 <Button  
 android:id="@+id/btnPauseResume"  
 style="@style/Widget.AppCompat.Button.Borderless"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="Pause"  
 tools:ignore="HardcodedText" />  
  
 <Button  
 android:id="@+id/btnCancel"  
 style="@style/Widget.AppCompat.Button.Borderless"  
 android:layout\_width="0dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_weight="1"  
 android:text="Cancel"  
 tools:ignore="HardcodedText" />  
 </LinearLayout>  
</LinearLayout>

**Bước 8:** Ở **com.example.lab7 –** chọn **New – Other - Broadcast Receiver –** đặt tên **NotificationActionReceiver – Finish**

**Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Nội dung do AI tạo ra có thể không chính xác.**

**Bước 9:** Ở file **DownloadService.java viết code như sau**

package com.example.lab7;  
  
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.File;  
import java.io.FileOutputStream;  
import java.io.InputStream;  
import java.net.HttpURLConnection;  
import java.net.URL;  
  
public class DownloadService extends Service {  
  
 public static DownloadService *instance*;  
  
 private static final int *NOTIFICATION\_ID* = 1;  
 private static final String *CHANNEL\_ID* = "DownloadChannel";  
  
 private NotificationManager notificationManager;  
 private NotificationCompat.Builder notificationBuilder;  
 private RemoteViews remoteViews;  
  
 private Thread downloadThread;  
 private volatile boolean isPaused = false;  
 private volatile boolean isCancelled = false;  
  
 @Override  
 public void onCreate() {  
 super.onCreate();  
 *instance* = this;  
 notificationManager = (NotificationManager) getSystemService(Context.*NOTIFICATION\_SERVICE*);  
 createNotificationChannel();  
 }  
  
 @Override  
 public int onStartCommand(Intent intent, int flags, int startId) {  
 final String urlString = intent.getStringExtra("URL");  
 if (urlString == null) {  
 stopSelf();  
 return *START\_NOT\_STICKY*;  
 }  
  
 isPaused = false;  
 isCancelled = false;  
  
 startForeground(*NOTIFICATION\_ID*, createNotification(urlString, 0));  
  
 // Bắt đầu download trên một thread riêng để không chặn UI  
 downloadThread = new Thread(() -> downloadFile(urlString));  
 downloadThread.start();  
  
 return *START\_NOT\_STICKY*;  
 }  
  
 private void downloadFile(String urlString) {  
 HttpURLConnection connection = null;  
 InputStream inputStream = null;  
 FileOutputStream outputStream = null;  
  
 try {  
 URL url = new URL(urlString);  
 connection = (HttpURLConnection) url.openConnection();  
 connection.connect();  
  
 if (connection.getResponseCode() != HttpURLConnection.*HTTP\_OK*) {  
 stopSelf();  
 return;  
 }  
  
 int fileLength = connection.getContentLength();  
 String fileName = urlString.substring(urlString.lastIndexOf('/') + 1);  
 File file = new File(getFilesDir(), fileName);  
  
 inputStream = connection.getInputStream();  
 outputStream = new FileOutputStream(file);  
  
 byte[] data = new byte[4096];  
 long total = 0;  
 int count;  
  
 while ((count = inputStream.read(data)) != -1) {  
 // Vòng lặp kiểm tra tạm dừng  
 while (isPaused && !isCancelled) {  
 try {  
 Thread.*sleep*(1000); // Ngủ 1 giây nếu bị tạm dừng  
 } catch (InterruptedException e) {  
 Thread.*currentThread*().interrupt();  
 }  
 }  
  
 if (isCancelled) {  
 break;  
 }  
  
 total += count;  
 outputStream.write(data, 0, count);  
  
 if (fileLength > 0) {  
 int progress = (int) (total \* 100 / fileLength);  
 updateNotification(progress);  
 }  
 }  
 } catch (Exception e) {  
 e.printStackTrace();  
 } finally {  
 try {  
 if (outputStream != null) outputStream.close();  
 if (inputStream != null) inputStream.close();  
 if (connection != null) connection.disconnect();  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
  
 if (isCancelled) {  
 String fileName = urlString.substring(urlString.lastIndexOf('/') + 1);  
 new File(getFilesDir(), fileName).delete();  
 }  
  
 stopForeground(true);  
 stopSelf();  
 }  
 }  
  
 private Notification createNotification(String link, int progress) {  
 remoteViews = new RemoteViews(getPackageName(), R.layout.*notification\_layout*);  
 remoteViews.setTextViewText(R.id.*tvLink*, "Link: " + link);  
 remoteViews.setTextViewText(R.id.*tvProgress*, "Complete: " + progress + "%");  
 remoteViews.setProgressBar(R.id.*progressBar*, 100, progress, false);  
  
 // Intent cho nút Pause/Resume  
 Intent pauseIntent = new Intent(this, NotificationActionReceiver.class);  
 pauseIntent.setAction("PAUSE\_RESUME");  
 PendingIntent pausePendingIntent = PendingIntent.*getBroadcast*(this, 0, pauseIntent, PendingIntent.*FLAG\_UPDATE\_CURRENT* | PendingIntent.*FLAG\_IMMUTABLE*);  
 remoteViews.setOnClickPendingIntent(R.id.*btnPauseResume*, pausePendingIntent);  
  
 // Intent cho nút Cancel  
 Intent cancelIntent = new Intent(this, NotificationActionReceiver.class);  
 cancelIntent.setAction("CANCEL");  
 PendingIntent cancelPendingIntent = PendingIntent.*getBroadcast*(this, 1, cancelIntent, PendingIntent.*FLAG\_UPDATE\_CURRENT* | PendingIntent.*FLAG\_IMMUTABLE*);  
 remoteViews.setOnClickPendingIntent(R.id.*btnCancel*, cancelPendingIntent);  
  
 notificationBuilder = new NotificationCompat.Builder(this, *CHANNEL\_ID*)  
 .setSmallIcon(R.mipmap.*ic\_launcher*) // Thay icon của bạn ở đây  
 .setCustomContentView(remoteViews)  
 .setOnlyAlertOnce(true);  
  
 return notificationBuilder.build();  
 }  
  
 private void updateNotification(int progress) {  
 remoteViews.setTextViewText(R.id.*tvProgress*, "Complete: " + progress + "%");  
 remoteViews.setProgressBar(R.id.*progressBar*, 100, progress, false);  
 notificationManager.notify(*NOTIFICATION\_ID*, notificationBuilder.build());  
 }  
  
 private void createNotificationChannel() {  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*O*) {  
 NotificationChannel channel = new NotificationChannel(  
 *CHANNEL\_ID*,  
 "Download Channel",  
 NotificationManager.*IMPORTANCE\_LOW* );  
 notificationManager.createNotificationChannel(channel);  
 }  
 }  
  
 // Các hàm công khai để BroadcastReceiver gọi  
 public void pauseOrResumeDownload() {  
 isPaused = !isPaused;  
 String buttonText = isPaused ? "Resume" : "Pause";  
 remoteViews.setTextViewText(R.id.*btnPauseResume*, buttonText);  
 notificationManager.notify(*NOTIFICATION\_ID*, notificationBuilder.build());  
 }  
  
 public void cancelDownload() {  
 isCancelled = true;  
 if (downloadThread != null) {  
 downloadThread.interrupt();  
 }  
 }  
  
 @Override  
 public void onDestroy() {  
 super.onDestroy();  
 *instance* = null;  
 if (downloadThread != null) {  
 downloadThread.interrupt();  
 }  
 }  
  
 @Nullable  
 @Override  
 public IBinder onBind(Intent intent) {  
 return null;  
 }  
}

**Bước 10:** Ở file **NotificationActionReceiver.java** viết code như sau

package com.example.lab7;  
  
import android.content.BroadcastReceiver;  
import android.content.Context;  
import android.content.Intent;  
  
public class NotificationActionReceiver extends BroadcastReceiver {  
  
 @Override  
 public void onReceive(Context context, Intent intent) {  
 if (intent != null && intent.getAction() != null) {  
 DownloadService serviceInstance = DownloadService.*instance*;  
 if (serviceInstance == null) return;  
  
 switch (intent.getAction()) {  
 case "PAUSE\_RESUME":  
 serviceInstance.pauseOrResumeDownload();  
 break;  
 case "CANCEL":  
 serviceInstance.cancelDownload();  
 break;  
 }  
 }  
 }  
}