Tên: LÊ THỊ MỸ HẰNG  
Lớp: 11\_ĐH\_TMDT  
MSSV: 1150070011

**LAB7**: **Service, Notification và BroadCast Receiver**

**Bước 1**: **Tạo project**

* Tạo mới project: **DownloadManagerLab7**
* Chọn ngôn ngữ: **Java**
* Chọn Minimum SDK: **API 24 (Android 7.0)**

**2.2. Giao diện người dùng (activity\_main.xml)**

<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical"  
 android:padding="24dp"  
 android:gravity="center\_horizontal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <TextView  
 android:text="Download Manager Lab 7"  
 android:textSize="20sp"  
 android:textStyle="bold"  
 android:layout\_marginBottom="20dp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content" />  
  
 <EditText  
 android:id="@+id/edtUrl"  
 android:hint="Nhập đường dẫn tải..."  
 android:inputType="textUri"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content" />  
  
 <Button  
 android:id="@+id/btnDownload"  
 android:text="Bắt đầu tải"  
 android:layout\_marginTop="16dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content" />  
</LinearLayout>

**2.3. MainActivity.java**

package com.example.downloadmanagerlab7;  
  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.content.ContextCompat;  
  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
  
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*);  
  
 btnDownload.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 String url = edtUrl.getText().toString().trim();  
  
 if (url.isEmpty()) {  
 Toast.*makeText*(MainActivity.this, "Vui lòng nhập đường dẫn tải!", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
  
 Intent intent = new Intent(MainActivity.this, DownloadService.class);  
 intent.putExtra("url", url);  
 ContextCompat.*startForegroundService*(MainActivity.this, intent);  
 }  
 });  
 }  
}

**2.4. DownloadService.java**

package com.example.downloadmanagerlab7;  
  
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 androidx.annotation.Nullable;  
import androidx.core.app.NotificationCompat;  
  
import java.io.BufferedInputStream;  
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 {  
  
 private static final String *CHANNEL\_ID* = "download\_channel";  
 private static final int *NOTIFICATION\_ID* = 1;  
  
 private boolean isPaused = false;  
 private boolean isCancelled = false;  
  
 @Nullable  
 @Override  
 public IBinder onBind(Intent intent) {  
 return null;  
 }  
  
 @Override  
 public void onCreate() {  
 super.onCreate();  
 createNotificationChannel();  
 }  
  
 @Override  
 public int onStartCommand(Intent intent, int flags, int startId) {  
 if (intent == null) return *START\_NOT\_STICKY*;  
 startForeground(*NOTIFICATION\_ID*, createNotification(0));  
 String action = intent.getAction();  
 if (action != null) {  
 switch (action) {  
 case "ACTION\_PAUSE":  
 isPaused = true;  
 updateNotificationText("Tạm dừng tải");  
 return *START\_STICKY*;  
  
 case "ACTION\_RESUME":  
 isPaused = false;  
 updateNotificationText("Tiếp tục tải");  
 return *START\_STICKY*;  
  
 case "ACTION\_CANCEL":  
 isCancelled = true;  
 updateNotificationText("Huỷ tải");  
 stopSelf();  
 return *START\_NOT\_STICKY*;  
 }  
 }  
  
 // Nếu không phải action điều khiển → bắt đầu tải mới  
 String url = intent.getStringExtra("url");  
 if (url == null || url.isEmpty()) {  
 stopSelf();  
 return *START\_NOT\_STICKY*;  
 }  
  
 startForeground(*NOTIFICATION\_ID*, createNotification(0));  
 new Thread(() -> downloadFile(url)).start();  
  
 return *START\_STICKY*;  
 }  
  
  
 private void downloadFile(String urlStr) {  
 try {  
 URL url = new URL(urlStr);  
 HttpURLConnection connection = (HttpURLConnection) url.openConnection();  
 connection.connect();  
  
 int fileLength = connection.getContentLength();  
 InputStream input = new BufferedInputStream(url.openStream());  
 File outputFile = new File(getFilesDir(), "downloaded\_file");  
 FileOutputStream output = new FileOutputStream(outputFile);  
  
 byte[] data = new byte[1024];  
 long total = 0;  
 int count;  
  
 while ((count = input.read(data)) != -1) {  
 if (isCancelled) break;  
 while (isPaused) Thread.*sleep*(300);  
  
 total += count;  
 output.write(data, 0, count);  
  
 int progress = (int) (total \* 100 / fileLength);  
 updateNotification(progress);  
 }  
  
 output.flush();  
 output.close();  
 input.close();  
  
 stopSelf();  
  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 private Notification createNotification(int progress) {  
 int flag = (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*M*)  
 ? PendingIntent.*FLAG\_IMMUTABLE* : 0;  
  
 Intent pauseIntent = new Intent(this, DownloadReceiver.class);  
 pauseIntent.setAction("ACTION\_PAUSE");  
 PendingIntent pausePending = PendingIntent.*getBroadcast*(this, 0, pauseIntent, flag);  
  
 Intent resumeIntent = new Intent(this, DownloadReceiver.class);  
 resumeIntent.setAction("ACTION\_RESUME");  
 PendingIntent resumePending = PendingIntent.*getBroadcast*(this, 1, resumeIntent, flag);  
  
 Intent cancelIntent = new Intent(this, DownloadReceiver.class);  
 cancelIntent.setAction("ACTION\_CANCEL");  
 PendingIntent cancelPending = PendingIntent.*getBroadcast*(this, 2, cancelIntent, flag);  
  
 NotificationCompat.Builder builder = new NotificationCompat.Builder(this, *CHANNEL\_ID*)  
 .setSmallIcon(android.R.drawable.*stat\_sys\_download*)  
 .setContentTitle("Đang tải xuống...")  
 .setContentText(progress + "% hoàn thành")  
 .setProgress(100, progress, false)  
 .addAction(0, "Tạm dừng", pausePending)  
 .addAction(0, "Tiếp tục", resumePending)  
 .addAction(0, "Huỷ", cancelPending)  
 .setOngoing(true)  
 .setPriority(NotificationCompat.*PRIORITY\_LOW*);  
  
 return builder.build();  
 }  
 private void updateNotificationText(String message) {  
 NotificationManager manager = (NotificationManager) getSystemService(Context.*NOTIFICATION\_SERVICE*);  
 Notification notification = new NotificationCompat.Builder(this, *CHANNEL\_ID*)  
 .setSmallIcon(android.R.drawable.*stat\_sys\_download\_done*)  
 .setContentTitle("Download Manager")  
 .setContentText(message)  
 .setPriority(NotificationCompat.*PRIORITY\_LOW*)  
 .build();  
 manager.notify(*NOTIFICATION\_ID*, notification);  
 }  
  
 private void updateNotification(int progress) {  
 NotificationManager manager =  
 (NotificationManager) getSystemService(Context.*NOTIFICATION\_SERVICE*);  
  
 // ✅ Fix cho Android 7 (API 24)  
 if (manager != null) {  
 Notification notification = createNotification(progress);  
 manager.notify(*NOTIFICATION\_ID*, notification);  
 }  
 }  
  
 private void createNotificationChannel() {  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*O*) {  
 NotificationChannel channel = new NotificationChannel(  
 *CHANNEL\_ID*,  
 "Download Channel",  
 NotificationManager.*IMPORTANCE\_LOW* );  
 NotificationManager manager = getSystemService(NotificationManager.class);  
 if (manager != null) manager.createNotificationChannel(channel);  
 }  
 }  
  
 public void pause() {  
 isPaused = true;  
 }  
  
 public void resume() {  
 isPaused = false;  
 }  
  
 public void cancel() {  
 isCancelled = true;  
 }  
}

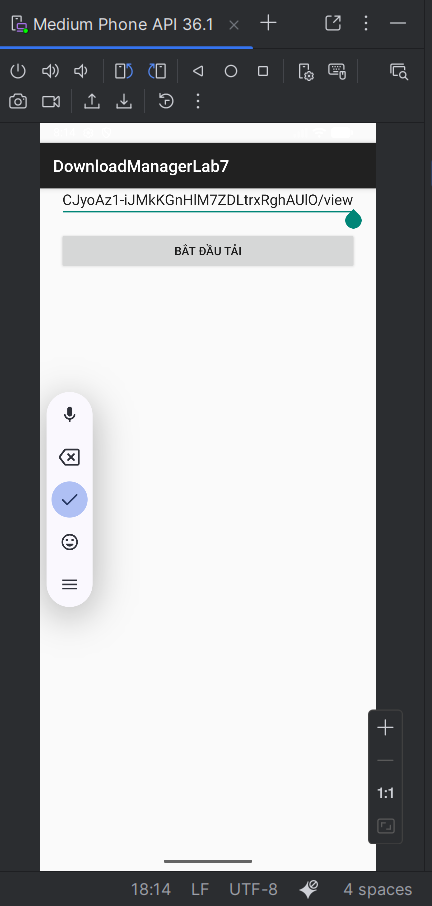
**2.5. DownloadReceiver.java**

package com.example.downloadmanagerlab7;  
  
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();  
  
 if (action != null) {  
 // Gửi lại Intent cho Service xử lý  
 Intent serviceIntent = new Intent(context, DownloadService.class);  
 serviceIntent.setAction(action);  
 context.startService(serviceIntent);  
 }  
 }  
}

**2.6. AndroidManifest.xml**

<?xml version="1.0" encoding="utf-8"?>  
<manifest xmlns:android="http://schemas.android.com/apk/res/android"  
 package="com.example.downloadmanagerlab7">  
  
 <!-- Quyền thông báo -->  
 <uses-permission android:name="android.permission.POST\_NOTIFICATIONS"/>  
  
 <application  
 android:allowBackup="true"  
 android:icon="@mipmap/ic\_launcher"  
 android:label="@string/app\_name"  
 android:supportsRtl="true"  
 android:theme="@style/Theme.AppCompat.Light.DarkActionBar">  
  
 <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>  
  
 <service  
 android:name=".DownloadService"  
 android:enabled="true"  
 android:exported="false"  
 android:foregroundServiceType="dataSync" />  
  
 <receiver  
 android:name=".DownloadReceiver"  
 android:enabled="true"  
 android:exported="true" />  
 </application>  
  
</manifest>

* **Kết quả:**

****