Bước 1. Xây dựng 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"  
 android:gravity="center">  
  
 <TextView  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Download Manager"  
 android:textSize="24sp"  
 android:textStyle="bold"  
 android:layout\_marginBottom="32dp"/>  
  
 <EditText  
 android:id="@+id/etDownloadUrl"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:hint="Enter download URL"  
 android:inputType="textUri"  
 android:padding="12dp"  
 android:layout\_marginBottom="16dp"/>  
  
 <Button  
 android:id="@+id/btnDownload"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:text="Download"  
 android:textSize="16sp"  
 android:padding="12dp"/>  
  
</LinearLayout>

A screenshot of a computer

AI-generated content may be incorrect.

Bước 2. Xây dựng notification\_download.xml tronglayout *<?*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="wrap\_content"  
 android:orientation="vertical"  
 android:padding="16dp">  
  
 <TextView  
 android:id="@+id/tvNotificationTitle"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Downloading..."  
 android:textSize="16sp"  
 android:textStyle="bold"  
 android:layout\_marginBottom="8dp"/>  
  
 <ProgressBar  
 android:id="@+id/progressBar"  
 style="?android:attr/progressBarStyleHorizontal"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:max="100"  
 android:progress="0"  
 android:layout\_marginBottom="8dp"/>  
  
 <TextView  
 android:id="@+id/tvNotificationProgress"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="0%"  
 android:textSize="14sp"  
 android:layout\_marginBottom="12dp"/>  
  
 <LinearLayout  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"  
 android:orientation="horizontal"  
 android:gravity="end">  
  
 <Button  
 android:id="@+id/btnPause"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Pause"  
 android:layout\_marginEnd="8dp"  
 style="?android:attr/buttonBarButtonStyle"/>  
  
 <Button  
 android:id="@+id/btnCancel"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:text="Cancel"  
 style="?android:attr/buttonBarButtonStyle"/>  
 </LinearLayout>  
  
</LinearLayout>

A screenshot of a computer

AI-generated content may be incorrect.

Bước 3.Xây dựng DownloadReceiver.java - BroadcastReceiver xử lý các action từ notification

package com.example.lab7\_duy;  
  
import android.content.BroadcastReceiver;  
import android.content.Context;  
import android.content.Intent;  
  
public class DownloadReceiver extends BroadcastReceiver {  
  
 public static final String *ACTION\_PAUSE* = "com.example.lab7\_duy.ACTION\_PAUSE";  
 public static final String *ACTION\_RESUME* = "com.example.lab7\_duy.ACTION\_RESUME";  
 public static final String *ACTION\_CANCEL* = "com.example.lab7\_duy.ACTION\_CANCEL";  
  
 @Override  
 public void onReceive(Context context, Intent intent) {  
 String action = intent.getAction();  
  
 if (action == null) return;  
  
 Intent serviceIntent = new Intent(context, DownloadService.class);  
 serviceIntent.setAction(action);  
  
 switch (action) {  
 case *ACTION\_PAUSE*:  
 context.startService(serviceIntent);  
 break;  
  
 case *ACTION\_RESUME*:  
 context.startService(serviceIntent);  
 break;  
  
 case *ACTION\_CANCEL*:  
 context.stopService(serviceIntent);  
 break;  
 }  
 }  
}

Bước 4. Xây dựng DownloadService.java

package com.example.lab7\_duy;  
  
import android.app.Notification;  
import android.app.NotificationChannel;  
import android.app.NotificationManager;  
import android.app.PendingIntent;  
import android.app.Service;  
import android.content.Intent;  
import android.os.Build;  
import android.os.IBinder;  
import android.widget.RemoteViews;  
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 {  
  
 private static final String *CHANNEL\_ID* = "DownloadChannel";  
 private static final int *NOTIFICATION\_ID* = 1;  
  
 private String downloadUrl;  
 private boolean isPaused = false;  
 private boolean isCancelled = false;  
 private Thread downloadThread;  
  
 @Override  
 public void onCreate() {  
 super.onCreate();  
 createNotificationChannel();  
 }  
  
 @Override  
 public int onStartCommand(Intent intent, int flags, int startId) {  
 if (intent != null) {  
 String action = intent.getAction();  
  
 if (action == null) {  
 *// Start new download* downloadUrl = intent.getStringExtra("download\_url");  
 startDownload();  
 } else {  
 handleAction(action);  
 }  
 }  
  
 return *START\_STICKY*;  
 }  
  
 private void handleAction(String action) {  
 switch (action) {  
 case DownloadReceiver.*ACTION\_PAUSE*:  
 isPaused = true;  
 updateNotification("Download Paused", 0, true);  
 break;  
  
 case DownloadReceiver.*ACTION\_RESUME*:  
 isPaused = false;  
 synchronized (this) {  
 notifyAll();  
 }  
 break;  
  
 case DownloadReceiver.*ACTION\_CANCEL*:  
 isCancelled = true;  
 isPaused = false;  
 if (downloadThread != null) {  
 downloadThread.interrupt();  
 }  
 stopForeground(true);  
 stopSelf();  
 break;  
 }  
 }  
  
 private void startDownload() {  
 startForeground(*NOTIFICATION\_ID*, createNotification("Starting download...", 0, false));  
  
 downloadThread = new Thread(() -> {  
 try {  
 URL url = new URL(downloadUrl);  
 HttpURLConnection connection = (HttpURLConnection) url.openConnection();  
 connection.connect();  
  
 int fileLength = connection.getContentLength();  
 String fileName = getFileNameFromUrl(downloadUrl);  
  
 File file = new File(getFilesDir(), fileName);  
 InputStream input = connection.getInputStream();  
 FileOutputStream output = new FileOutputStream(file);  
  
 byte[] buffer = new byte[4096];  
 long total = 0;  
 int count;  
  
 while ((count = input.read(buffer)) != -1) {  
 if (isCancelled) {  
 input.close();  
 output.close();  
 file.delete();  
 return;  
 }  
  
 while (isPaused && !isCancelled) {  
 synchronized (this) {  
 try {  
 wait();  
 } catch (InterruptedException e) {  
 return;  
 }  
 }  
 }  
  
 total += count;  
 output.write(buffer, 0, count);  
  
 int progress = (int) ((total \* 100) / fileLength);  
 updateNotification("Downloading...", progress, false);  
 }  
  
 output.flush();  
 output.close();  
 input.close();  
  
 updateNotification("Download Complete!", 100, false);  
  
 Thread.*sleep*(2000);  
 stopForeground(true);  
 stopSelf();  
  
 } catch (Exception e) {  
 e.printStackTrace();  
 updateNotification("Download Failed", 0, false);  
 }  
 });  
  
 downloadThread.start();  
 }  
  
 private String getFileNameFromUrl(String url) {  
 String fileName = url.substring(url.lastIndexOf('/') + 1);  
 if (fileName.contains("?")) {  
 fileName = fileName.substring(0, fileName.indexOf("?"));  
 }  
 if (fileName.isEmpty()) {  
 fileName = "download\_" + System.*currentTimeMillis*();  
 }  
 return fileName;  
 }  
  
 private void createNotificationChannel() {  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*O*) {  
 NotificationChannel channel = new NotificationChannel(  
 *CHANNEL\_ID*,  
 "Download Notifications",  
 NotificationManager.*IMPORTANCE\_LOW* );  
 NotificationManager manager = getSystemService(NotificationManager.class);  
 if (manager != null) {  
 manager.createNotificationChannel(channel);  
 }  
 }  
 }  
  
 private Notification createNotification(String title, int progress, boolean isPaused) {  
 RemoteViews notificationLayout = new RemoteViews(getPackageName(), R.layout.*notification\_download*);  
  
 notificationLayout.setTextViewText(R.id.*tvNotificationTitle*, title);  
 notificationLayout.setProgressBar(R.id.*progressBar*, 100, progress, false);  
 notificationLayout.setTextViewText(R.id.*tvNotificationProgress*, progress + "%");  
  
 *// Pause/Resume button* Intent pauseIntent = new Intent(this, DownloadReceiver.class);  
 if (isPaused) {  
 pauseIntent.setAction(DownloadReceiver.*ACTION\_RESUME*);  
 notificationLayout.setTextViewText(R.id.*btnPause*, "Resume");  
 } else {  
 pauseIntent.setAction(DownloadReceiver.*ACTION\_PAUSE*);  
 notificationLayout.setTextViewText(R.id.*btnPause*, "Pause");  
 }  
 PendingIntent pausePendingIntent = PendingIntent.*getBroadcast*(  
 this, 0, pauseIntent, PendingIntent.*FLAG\_UPDATE\_CURRENT* | PendingIntent.*FLAG\_IMMUTABLE* );  
 notificationLayout.setOnClickPendingIntent(R.id.*btnPause*, pausePendingIntent);  
  
 *// Cancel button* Intent cancelIntent = new Intent(this, DownloadReceiver.class);  
 cancelIntent.setAction(DownloadReceiver.*ACTION\_CANCEL*);  
 PendingIntent cancelPendingIntent = PendingIntent.*getBroadcast*(  
 this, 1, cancelIntent, PendingIntent.*FLAG\_UPDATE\_CURRENT* | PendingIntent.*FLAG\_IMMUTABLE* );  
 notificationLayout.setOnClickPendingIntent(R.id.*btnCancel*, cancelPendingIntent);  
  
 return new NotificationCompat.Builder(this, *CHANNEL\_ID*)  
 .setSmallIcon(android.R.drawable.*stat\_sys\_download*)  
 .setStyle(new NotificationCompat.DecoratedCustomViewStyle())  
 .setCustomContentView(notificationLayout)  
 .setPriority(NotificationCompat.*PRIORITY\_LOW*)  
 .setOngoing(true)  
 .build();  
 }  
  
 private void updateNotification(String title, int progress, boolean isPaused) {  
 NotificationManager notificationManager = (NotificationManager) getSystemService(*NOTIFICATION\_SERVICE*);  
 if (notificationManager != null) {  
 notificationManager.notify(*NOTIFICATION\_ID*, createNotification(title, progress, isPaused));  
 }  
 }  
  
 @Override  
 public IBinder onBind(Intent intent) {  
 return null;  
 }  
  
 @Override  
 public void onDestroy() {  
 super.onDestroy();  
 isCancelled = true;  
 if (downloadThread != null) {  
 downloadThread.interrupt();  
 }  
 }  
}

bước 5. Xây dựng Mainactivity.java

package com.example.lab7\_duy;  
  
import android.Manifest;  
import android.content.Intent;  
import android.content.pm.PackageManager;  
import android.os.Build;  
import android.os.Bundle;  
import android.widget.Button;  
import android.widget.EditText;  
import android.widget.Toast;  
import androidx.appcompat.app.AppCompatActivity;  
import androidx.core.app.ActivityCompat;  
import androidx.core.content.ContextCompat;  
  
public class MainActivity extends AppCompatActivity {  
  
 private static final int *PERMISSION\_REQUEST\_CODE* = 100;  
 private EditText etDownloadUrl;  
 private Button btnDownload;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_main*);  
  
 etDownloadUrl = findViewById(R.id.*etDownloadUrl*);  
 btnDownload = findViewById(R.id.*btnDownload*);  
  
 *// Request permissions* requestPermissions();  
  
 btnDownload.setOnClickListener(v -> {  
 String url = etDownloadUrl.getText().toString().trim();  
  
 if (url.isEmpty()) {  
 Toast.*makeText*(this, "Please enter a download URL", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
  
 if (!url.startsWith("http://") && !url.startsWith("https://")) {  
 Toast.*makeText*(this, "Please enter a valid URL", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
  
 startDownloadService(url);  
 });  
 }  
  
 private void requestPermissions() {  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*TIRAMISU*) {  
 *// Android 13+* if (ContextCompat.*checkSelfPermission*(this, Manifest.permission.*POST\_NOTIFICATIONS*)  
 != PackageManager.*PERMISSION\_GRANTED*) {  
 ActivityCompat.*requestPermissions*(this,  
 new String[]{Manifest.permission.*POST\_NOTIFICATIONS*},  
 *PERMISSION\_REQUEST\_CODE*);  
 }  
 }  
  
 *// Request INTERNET permission (should be in manifest)* if (ContextCompat.*checkSelfPermission*(this, Manifest.permission.*INTERNET*)  
 != PackageManager.*PERMISSION\_GRANTED*) {  
 ActivityCompat.*requestPermissions*(this,  
 new String[]{Manifest.permission.*INTERNET*},  
 *PERMISSION\_REQUEST\_CODE*);  
 }  
 }  
  
 private void startDownloadService(String url) {  
 Intent intent = new Intent(this, DownloadService.class);  
 intent.putExtra("download\_url", url);  
  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*O*) {  
 startForegroundService(intent);  
 } else {  
 startService(intent);  
 }  
  
 Toast.*makeText*(this, "Download started", Toast.*LENGTH\_SHORT*).show();  
 }  
}

bước 6. Bổ sung trong AndroidManifest.xml

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

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

package="com.example.lab7\_duy">

<!-- Permissions -->

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

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

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

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

<uses-permission android:name="android.permission.ACCESS\_NETWORK\_STATE" />

<application

android:allowBackup="true"

android:icon="@mipmap/ic\_launcher"

android:label="@string/app\_name"

android:roundIcon="@mipmap/ic\_launcher\_round"

android:supportsRtl="true"

android:theme="@style/Theme.AppCompat.Light.DarkActionBar"

android:usesCleartextTraffic="true">

<!-- Main Activity -->

<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>

<!-- Download Service -->

<service

android:name=".DownloadService"

android:enabled="true"

android:exported="false"

android:foregroundServiceType="dataSync" />

<!-- Broadcast Receiver -->

<receiver

android:name=".DownloadReceiver"

android:enabled="true"

android:exported="false">

<intent-filter>

<action android:name="com.example.lab7\_duy.ACTION\_PAUSE" />

<action android:name="com.example.lab7\_duy.ACTION\_RESUME" />

<action android:name="com.example.lab7\_duy.ACTION\_CANCEL" />

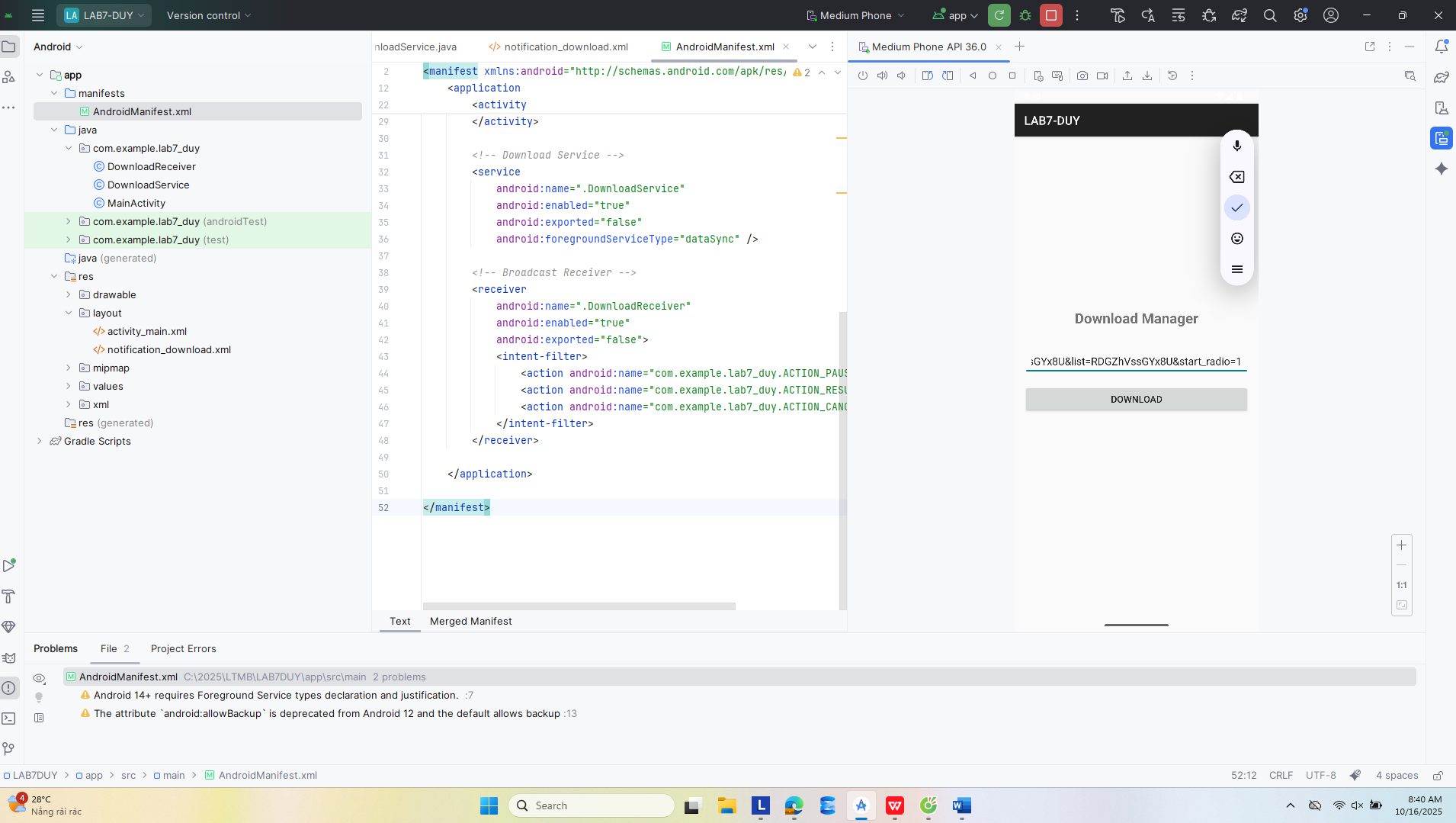
</intent-filter>

</receiver>

</application>

</manifest>

Kết quả:



A screenshot of a computer

AI-generated content may be incorrect.