Nguyễn Trường Hưng – 1150070013 – TMĐT

Bài tập lý thuyết buổi 7

**Bước 1: Tạo project mới**

**Bước 2: Thiết kế giao diện Tạo bố cục cơ bản với EditText nhập link và nút tải xuống.**

<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">  
  
 <EditText  
 android:id="@+id/edtLink"  
 android:hint="Nhập link cần tải xuống"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"/>  
  
 <Button  
 android:id="@+id/btnDownload"  
 android:text="Download"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content"/>  
</LinearLayout>

**Bước 3: Tạo lớp Service (DownloadService.java)**

public class DownloadService extends Service {  
 private boolean isPaused = false, isCancelled = false;  
 private NotificationManager notificationManager;  
 private static final String *CHANNEL\_ID* = "download\_channel";  
  
 @Override  
 public int onStartCommand(Intent intent, int flags, int startId) {  
 String url = intent.getStringExtra("url");  
 new Thread(() -> downloadFile(url)).start();  
 return START\_STICKY;  
 }  
  
 private void downloadFile(String urlStr) {  
 try {  
 URL url = new URL(urlStr);  
 HttpURLConnection conn = (HttpURLConnection) url.openConnection();  
 conn.connect();  
  
 int length = conn.getContentLength();  
 InputStream in = new BufferedInputStream(url.openStream());  
 FileOutputStream out = openFileOutput("downloaded\_file", MODE\_PRIVATE);  
  
 byte[] buffer = new byte[1024];  
 int total = 0, count;  
  
 while ((count = in.read(buffer)) != -1) {  
 if (isCancelled) break;  
 if (isPaused) continue;  
  
 total += count;  
 out.write(buffer, 0, count);  
 int progress = (int) (total \* 100 / length);  
 updateNotification(progress);  
 }  
  
 out.flush();  
 out.close();  
 in.close();  
 showCompletedNotification();  
  
 } catch (Exception e) {  
 e.printStackTrace();  
 }  
 }  
  
 private void updateNotification(int progress) {  
 NotificationCompat.Builder builder = new NotificationCompat.Builder(this, *CHANNEL\_ID*)  
 .setContentTitle("Đang tải...")  
 .setSmallIcon(android.R.drawable.*stat\_sys\_download*)  
 .setProgress(100, progress, false)  
 .setOngoing(true);  
 notificationManager.notify(1, builder.build());  
 }  
  
 private void showCompletedNotification() {  
 NotificationCompat.Builder builder = new NotificationCompat.Builder(this, *CHANNEL\_ID*)  
 .setContentTitle("Tải hoàn tất")  
 .setSmallIcon(android.R.drawable.*stat\_sys\_download\_done*);  
 notificationManager.notify(1, builder.build());  
 }  
  
 @Override  
 public IBinder onBind(Intent intent) { return null; }  
}

**Bước 4: Tạo lớp BroadcastReceiver (DownloadReceiver.java)**

package com.example.nguyentruonghungbtlt7;  
  
import android.content.BroadcastReceiver;  
import android.content.Context;  
import android.content.Intent;  
  
*/\*\*  
 \* Nhận các action từ Notification và forward sang DownloadService  
 \* để Service đổi trạng thái (pause/resume/cancel) mà không tạo lần tải mới.  
 \*/*public class DownloadReceiver extends BroadcastReceiver {  
 @Override  
 public void onReceive(Context context, Intent intent) {  
 if (intent == null || intent.getAction() == null) return;  
  
 String action = intent.getAction();  
 Intent svc = new Intent(context, DownloadService.class);  
 svc.setAction(action);  
 *// Android 8+ dùng startForegroundService* context.startForegroundService(svc);  
 }  
}

Khai báo trong AndroidManifest.xml: <receiver android:name=".DownloadReceiver" />

**Bước 5: Tạo Notification có nút điều khiển**

package com.example.nguyentruonghungbtlt7;  
  
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 androidx.annotation.Nullable;  
import androidx.core.app.NotificationCompat;  
  
import java.io.BufferedInputStream;  
import java.io.FileOutputStream;  
import java.io.InputStream;  
import java.net.HttpURLConnection;  
import java.net.URL;  
  
public class DownloadService extends Service {  
  
 public static final String *ACTION\_PAUSE* = "PAUSE";  
 public static final String *ACTION\_RESUME* = "RESUME";  
 public static final String *ACTION\_CANCEL* = "CANCEL";  
  
 private static final String *CHANNEL\_ID* = "download\_channel";  
 private static final int *NOTI\_ID* = 1;  
  
 private volatile boolean isPaused = false;  
 private volatile boolean isCancelled = false;  
  
 private NotificationManager notificationManager;  
 private PendingIntent pausePI, resumePI, cancelPI;  
  
 @Override  
 public void onCreate() {  
 super.onCreate();  
 notificationManager = (NotificationManager) getSystemService(*NOTIFICATION\_SERVICE*);  
 createNotificationChannel();  
 createActionIntents();  
 *// khởi tạo foreground sớm để tránh ANR trên Android 8+* startForeground(*NOTI\_ID*, buildProgressNotification(0, true).build());  
 }  
  
 @Override  
 public int onStartCommand(@Nullable Intent intent, int flags, int startId) {  
 if (intent != null && intent.getAction() != null) {  
 switch (intent.getAction()) {  
 case *ACTION\_PAUSE*: isPaused = true; updateNotiPaused(); break;  
 case *ACTION\_RESUME*: isPaused = false; updateNotiResumed(); break;  
 case *ACTION\_CANCEL*: isCancelled = true; break;  
 }  
 return *START\_STICKY*;  
 }  
  
 final String url = intent != null ? intent.getStringExtra("url") : null;  
 new Thread(() -> downloadFile(url)).start();  
 return *START\_STICKY*;  
 }  
  
 private void downloadFile(String urlStr) {  
 if (urlStr == null || urlStr.isEmpty()) {  
 stopSelf();  
 return;  
 }  
  
 HttpURLConnection conn = null;  
 try {  
 URL url = new URL(urlStr);  
 conn = (HttpURLConnection) url.openConnection();  
 conn.setConnectTimeout(10000);  
 conn.setReadTimeout(15000);  
 conn.connect();  
  
 int length = conn.getContentLength(); *// có thể = -1* InputStream in = new BufferedInputStream(conn.getInputStream());  
 *// lưu nội bộ: data/data/<pkg>/files/downloaded\_file* FileOutputStream out = openFileOutput("downloaded\_file", *MODE\_PRIVATE*);  
  
 byte[] buffer = new byte[8 \* 1024];  
 long total = 0;  
 int count;  
  
 *// cập nhật noti 0% khi bắt đầu* notificationManager.notify(*NOTI\_ID*, buildProgressNotification(0, length <= 0).build());  
  
 while ((count = in.read(buffer)) != -1) {  
 if (isCancelled) {  
 in.close();  
 out.close();  
 stopForeground(true);  
 stopSelf();  
 return;  
 }  
  
 *// pause "mềm": đợi 200ms rồi kiểm tra lại* if (isPaused) {  
 Thread.*sleep*(200);  
 continue;  
 }  
  
 out.write(buffer, 0, count);  
 total += count;  
  
 if (length > 0) {  
 int progress = (int) (total \* 100 / length);  
 notificationManager.notify(*NOTI\_ID*, buildProgressNotification(progress, false).build());  
 } else {  
 *// indeterminate* notificationManager.notify(*NOTI\_ID*, buildProgressNotification(0, true).build());  
 }  
 }  
  
 out.flush();  
 out.close();  
 in.close();  
  
 showCompletedNotification();  
  
 } catch (Exception e) {  
 e.printStackTrace();  
 *// báo lỗi ngắn gọn trên notification* NotificationCompat.Builder err = new NotificationCompat.Builder(this, *CHANNEL\_ID*)  
 .setSmallIcon(android.R.drawable.*stat\_notify\_error*)  
 .setContentTitle("Tải thất bại")  
 .setContentText(e.getMessage())  
 .setOngoing(false);  
 notificationManager.notify(*NOTI\_ID*, err.build());  
 } finally {  
 if (conn != null) conn.disconnect();  
 stopForeground(false);  
 stopSelf();  
 }  
 }  
  
 */\* ============= Notification helpers ============= \*/* private void createActionIntents() {  
 Intent pause = new Intent(this, DownloadReceiver.class).setAction(*ACTION\_PAUSE*);  
 Intent resume = new Intent(this, DownloadReceiver.class).setAction(*ACTION\_RESUME*);  
 Intent cancel = new Intent(this, DownloadReceiver.class).setAction(*ACTION\_CANCEL*);  
  
 int flags = PendingIntent.*FLAG\_IMMUTABLE*;  
 pausePI = PendingIntent.*getBroadcast*(this, 0, pause, flags);  
 resumePI = PendingIntent.*getBroadcast*(this, 1, resume, flags);  
 cancelPI = PendingIntent.*getBroadcast*(this, 2, cancel, flags);  
 }  
  
 private NotificationCompat.Builder buildProgressNotification(int progress, boolean indeterminate) {  
 NotificationCompat.Builder b = new NotificationCompat.Builder(this, *CHANNEL\_ID*)  
 .setSmallIcon(android.R.drawable.*stat\_sys\_download*)  
 .setContentTitle("Đang tải file…")  
 .setOnlyAlertOnce(true)  
 .setOngoing(true)  
 .addAction(android.R.drawable.*ic\_media\_pause*, "Pause", pausePI)  
 .addAction(android.R.drawable.*ic\_media\_play*, "Resume", resumePI)  
 .addAction(android.R.drawable.*ic\_menu\_close\_clear\_cancel*, "Cancel", cancelPI);  
  
 if (indeterminate) {  
 b.setProgress(0, 0, true).setContentText("Đang tải…");  
 } else {  
 b.setProgress(100, progress, false).setContentText(progress + "% hoàn thành");  
 }  
 return b;  
 }  
  
 private void updateNotiPaused() {  
 NotificationCompat.Builder b = new NotificationCompat.Builder(this, *CHANNEL\_ID*)  
 .setSmallIcon(android.R.drawable.*stat\_sys\_download*)  
 .setContentTitle("Đã tạm dừng")  
 .setContentText("Nhấn Resume để tiếp tục")  
 .setOnlyAlertOnce(true)  
 .setOngoing(true)  
 .addAction(android.R.drawable.*ic\_media\_play*, "Resume", resumePI)  
 .addAction(android.R.drawable.*ic\_menu\_close\_clear\_cancel*, "Cancel", cancelPI);  
 notificationManager.notify(*NOTI\_ID*, b.build());  
 }  
  
 private void updateNotiResumed() {  
 *// khi resume, để progress indeterminate 1 nhịp; vòng lặp sẽ cập nhật % thật* notificationManager.notify(*NOTI\_ID*, buildProgressNotification(0, true).build());  
 }  
  
 private void showCompletedNotification() {  
 NotificationCompat.Builder builder = new NotificationCompat.Builder(this, *CHANNEL\_ID*)  
 .setSmallIcon(android.R.drawable.*stat\_sys\_download\_done*)  
 .setContentTitle("Tải hoàn tất")  
 .setContentText("File đã lưu trong bộ nhớ trong của ứng dụng")  
 .setOngoing(false);  
 notificationManager.notify(*NOTI\_ID*, builder.build());  
 }  
  
 private void createNotificationChannel() {  
 if (Build.VERSION.*SDK\_INT* >= Build.VERSION\_CODES.*O*) {  
 NotificationChannel ch = new NotificationChannel(  
 *CHANNEL\_ID*, "Download Channel", NotificationManager.*IMPORTANCE\_LOW*);  
 ch.setDescription("Hiển thị tiến trình tải xuống");  
 notificationManager.createNotificationChannel(ch);  
 }  
 }@Nullable  
 @Override  
 public IBinder onBind(Intent intent) {  
 return null;  
 }  
}

Bước 6: Kết nối giao diện với Service (MainActivity.java)

btnDownload.setOnClickListener(v -> {  
 String url = edtLink.getText().toString().trim();  
 if (TextUtils.*isEmpty*(url)) {  
 Toast.*makeText*(this, "Vui lòng nhập link tải xuống", Toast.*LENGTH\_SHORT*).show();  
 return;  
 }  
 startDownloadService(url);  
 });  
}

**Bước 7: Cấp quyền truy cập Internet**

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

Kiểm tra:

Nhập link test download

A screenshot of a phone

AI-generated content may be incorrect.

A close-up of a computer code

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.