**Tên: Lê Thị Mỹ Hằng  
Lớp: 11\_ĐH\_TMĐT  
MSSV: 1150070011**

**Bài tập lý thuyết 6**

**Bài 1: Viết một ứng dụng MultiThread sử dụng Message.**->Tạo thread nền sinh số ngẫu nhiên và tăng biến đếm; gửi Message về UI qua Handler để cập nhật TextView & ProgressBar.

**Bước 1:** Xây dựng string + dimens

res/values/strings.xml

<resources>

<string name="app\_name">BTLythuyet6</string>

<string name="start">Start</string>

<string name="working">Working...</string>

<string name="done\_background\_thread\_has\_been\_stopped">Done\nBackground thread has been stopped</string>

<string name="returned\_by\_bg\_thread">Returned by background thread:\n\n</string>

<string name="global\_value\_seen">\nGlobal value seen by all thread</string>

<string name="next">Next</string>

</resources>  
res/values/ dimens.xml   
<resources>

<dimen name="margin\_base">5dp</dimen>

<dimen name="text\_medium\_large">18sp</dimen>

<dimen name="text\_medium">16sp</dimen>

</resources>

**Bước 2:** Tạo layout res/layout/activity\_main.xml

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

android:orientation="vertical"

android:padding="10dp"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<TextView

android:id="@+id/tvMsgWorking"

android:textSize="18sp"

android:textStyle="bold"

android:text="Status"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"/>

<ProgressBar

android:id="@+id/pbFirst"

style="?android:attr/progressBarStyleHorizontal"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:max="100"

android:layout\_marginTop="8dp"/>

<ProgressBar

android:id="@+id/pbSecond"

style="?android:attr/progressBarStyleLarge"

android:layout\_gravity="center"

android:layout\_marginTop="8dp"

android:visibility="gone"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

<TextView

android:id="@+id/tvMsgReturned"

android:textSize="16sp"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="8dp"/>

<Button

android:id="@+id/btnStart"

android:text="Start"

android:layout\_gravity="center"

android:layout\_marginTop="12dp"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

</LinearLayout>  
  
**Bước 3**: Code MainActivity.java

package com.example.btlythuyet6;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.os.Handler;

import android.os.Message;

import android.widget.Button;

import android.widget.ProgressBar;

import android.widget.TextView;

import java.util.Random;

public class MainActivity extends AppCompatActivity {

private ProgressBar pbFirst, pbSecond;

private TextView tvMsgWorking, tvMsgReturned;

private Button btnStart;

private boolean isRunning;

private int MAX\_SEC = 10;

private int intTest;

private Thread bgThread;

private Handler handler;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

findViewByIds();

initVariables();

}

private void findViewByIds() {

pbFirst = findViewById(R.id.pbFirst);

pbSecond = findViewById(R.id.pbSecond);

tvMsgWorking = findViewById(R.id.tvMsgWorking);

tvMsgReturned = findViewById(R.id.tvMsgReturned);

btnStart = findViewById(R.id.btnStart);

}

private void initVariables() {

isRunning = false;

intTest = 0;

handler = new Handler(msg -> {

String messageText = (String) msg.obj;

tvMsgReturned.setText(messageText);

pbFirst.incrementProgressBy(1);

if (pbFirst.getProgress() >= MAX\_SEC) {

tvMsgWorking.setText("Done\nBackground thread has been stopped");

pbSecond.setVisibility(ProgressBar.GONE);

isRunning = false;

}

return true;

});

btnStart.setOnClickListener(v -> {

pbFirst.setProgress(0);

pbSecond.setVisibility(ProgressBar.VISIBLE);

tvMsgWorking.setText("Working...");

isRunning = true;

initBgThread();

});

}

private void initBgThread() {

bgThread = new Thread(() -> {

try {

Random r = new Random();

for (int i = 0; i < MAX\_SEC && isRunning; i++) {

Thread.sleep(1000);

intTest++;

String msgData = "Returned by background thread:\n\n"

+ r.nextInt(100)

+ "\nGlobal value seen by all thread: "

+ intTest;

Message msg = handler.obtainMessage(1, msgData);

handler.sendMessage(msg);

}

} catch (InterruptedException ignored) { }

});

bgThread.start();

}

@Override

protected void onStart() {

super.onStart();

// Khởi động thread khi Activity start

// Nếu muốn tự start: initBgThread();

}

@Override

protected void onStop() {

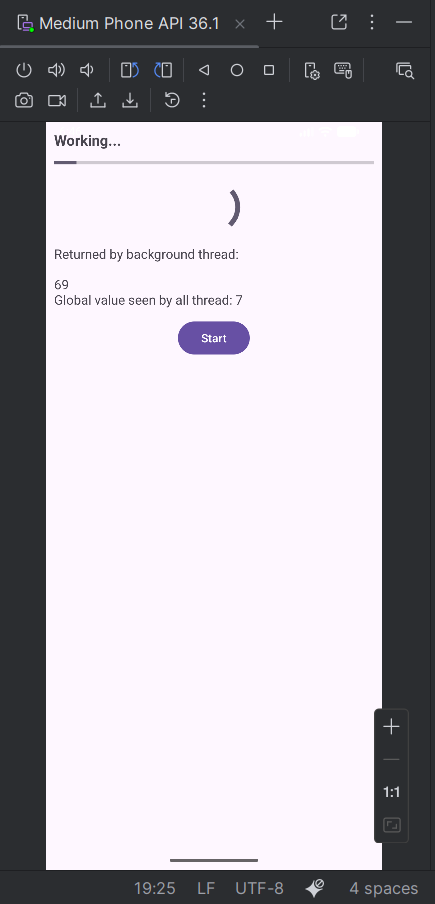
super.onStop();

isRunning = false;

}

}

* Kết quả:



**Bài 2: Handler.post(Runnable)**

Tạo 1 luồng nền (testThread) chạy bgRunnable, mỗi giây tăng giá trị toàn cục (globalValue), và dùng handler.post(fgRunnable) để cập nhật UI an toàn.  
Ngoài ra, có nút Execute để nhập text và hiện Toast bằng handler.post().

**Bước 1**: Tạo layout res/layout/activity\_post.xml

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

android:orientation="vertical"

android:padding="10dp"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<TextView

android:id="@+id/tvTopCaption"

android:text="Global Value: 0 | Accum: 0"

android:textSize="16sp"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"/>

<ProgressBar

android:id="@+id/pbWaiting"

style="?android:attr/progressBarStyleHorizontal"

android:max="20"

android:layout\_marginTop="8dp"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"/>

<EditText

android:id="@+id/etInput"

android:hint="Enter some data here"

android:textSize="16sp"

android:layout\_marginTop="12dp"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"/>

<Button

android:id="@+id/btnExecute"

android:text="Execute"

android:layout\_gravity="center"

android:layout\_marginTop="12dp"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"/>

</LinearLayout>

**Bước 2**: Code PostActivity.java

package com.example.btlythuyet6;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

import android.os.Handler;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ProgressBar;

import android.widget.TextView;

import android.widget.Toast;

public class PostActivity extends AppCompatActivity {

private ProgressBar pbWaiting;

private TextView tvTopCaption;

private EditText etInput;

private Button btnExecute;

private int globalValue = 0;

private int accum = 0;

private Handler handler;

private Runnable fgRunnable, bgRunnable;

private Thread testThread;

private volatile boolean isRunning = false;

private static final int MAX\_PROGRESS = 20;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_post);

findViews();

initVariables();

btnExecute.setOnClickListener(v -> {

String input = etInput.getText().toString().trim();

String finalInput = input.isEmpty() ? "(empty)" : input;

handler.post(() -> Toast.makeText(PostActivity.this,

"You said: " + finalInput, Toast.LENGTH\_SHORT).show());

});

}

private void findViews() {

tvTopCaption = findViewById(R.id.tvTopCaption);

pbWaiting = findViewById(R.id.pbWaiting);

etInput = findViewById(R.id.etInput);

btnExecute = findViewById(R.id.btnExecute);

}

private void initVariables() {

handler = new Handler();

pbWaiting.setMax(MAX\_PROGRESS);

// Runnable chạy trên UI thread (foreground)

fgRunnable = () -> {

synchronized (PostActivity.this) {

globalValue += 100;

accum++;

}

tvTopCaption.setText("Global Value: " + globalValue + " | Accum: " + accum);

pbWaiting.setProgress(accum);

};

// Runnable chạy trong background thread

bgRunnable = () -> {

try {

for (int i = 0; i < MAX\_PROGRESS && isRunning; i++) {

Thread.sleep(1000); // giả lập công việc

synchronized (PostActivity.this) {

globalValue += 1;

}

// post lên main thread để cập nhật giao diện

handler.post(fgRunnable);

}

} catch (InterruptedException ignored) { }

};

}

@Override

protected void onStart() {

super.onStart();

if (!isRunning) {

isRunning = true;

testThread = new Thread(bgRunnable);

testThread.start();

}

}

@Override

protected void onStop() {

super.onStop();

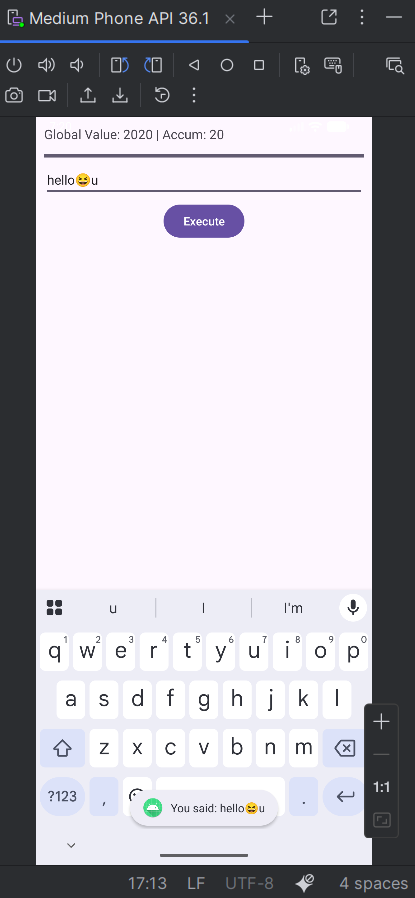
isRunning = false;

if (testThread != null) testThread.interrupt();

}

}

* Kết quả:



**Bài 3 : AsyncTask (Quick Job & Slow Job)**

**->** Tạo 2 công việc**:**

* Quick Job → chạy nhanh (hiện Toast)
* Slow Job → chạy lâu (mô phỏng tiến trình, có ProgressDialog)
* Dùng AsyncTask để xử lý nền và cập nhật tiến độ (publishProgress, onProgressUpdate).

**Bước 1**: Xây dựng string + dimens

**res/values/dimens.xml**

<resources>

<dimen name="margin\_base">5dp</dimen>

<dimen name="text\_small">14sp</dimen>

<dimen name="text\_medium">16sp</dimen>

<dimen name="text\_medium\_large">18sp</dimen>

<dimen name="text\_large">20sp</dimen>

</resources>

**res/values/strings.xml**

<resources>

<string name="app\_name">BTLythuyet6</string>

.....

<--Thêm -->

<string name="quick\_job">Quick Job</string>

<string name="slow\_job">Slow Job</string>

<string name="please\_wait">Some SLOW job is being done. Please wait...</string>

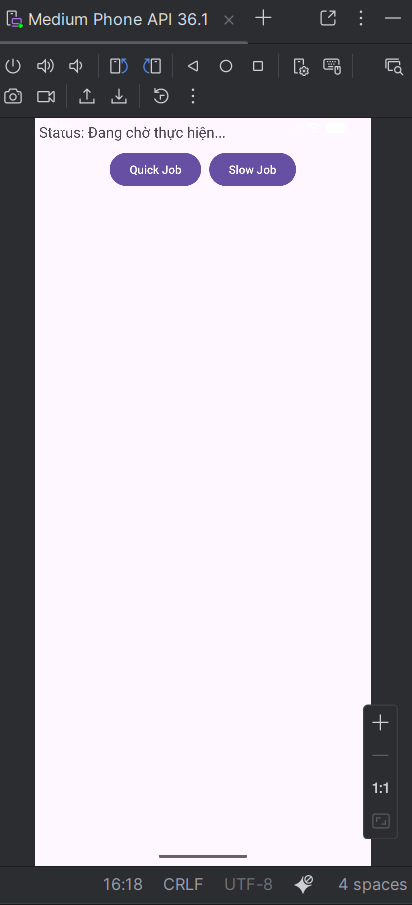
</resources>

**Bước 2**: Tạo Layout – activity\_async.xml

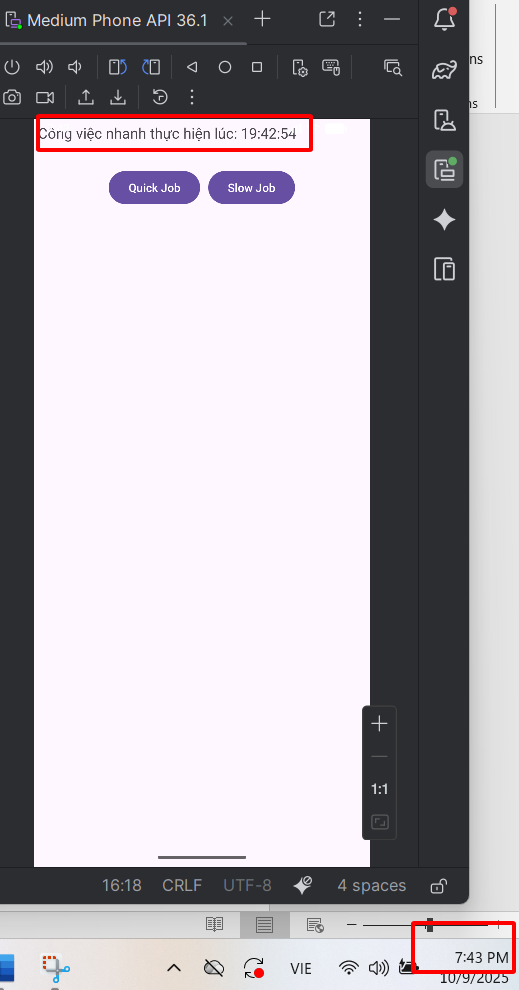
<?xml version="1.0" encoding="utf-8"?>  
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"  
 android:orientation="vertical"  
 android:padding="@dimen/margin\_base"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent">  
  
 <TextView  
 android:id="@+id/tvResult"  
 android:text="Result will appear here"  
 android:textSize="@dimen/text\_medium\_large"  
 android:layout\_width="match\_parent"  
 android:layout\_height="wrap\_content" />  
  
 <Button  
 android:id="@+id/btnQuickJob"  
 android:text="@string/quick\_job"  
 android:layout\_marginTop="@dimen/margin\_base"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content" />  
  
 <Button  
 android:id="@+id/btnSlowJob"  
 android:text="@string/slow\_job"  
 android:layout\_marginTop="@dimen/margin\_base"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content" />  
  
</LinearLayout>

**Bước 3**: Code – AsyncActivity.java

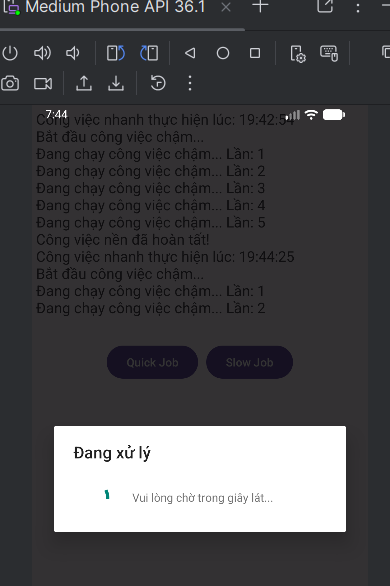
package com.example.btlythuyet6;  
  
import androidx.appcompat.app.AppCompatActivity;  
import android.os.AsyncTask;  
import android.os.Bundle;  
import android.app.ProgressDialog;  
import android.widget.Button;  
import android.widget.TextView;  
import android.widget.Toast;  
  
public class AsyncActivity extends AppCompatActivity {  
  
 private TextView tvResult;  
 private Button btnQuickJob, btnSlowJob;  
 private ProgressDialog progressDialog;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_async*);  
  
 findViews();  
  
 btnQuickJob.setOnClickListener(v -> {  
 Toast.*makeText*(this, "Quick Job Done!", Toast.*LENGTH\_SHORT*).show();  
 });  
  
 btnSlowJob.setOnClickListener(v -> {  
 new SlowTask().execute(10); // chạy 10 lần lặp  
 });  
 }  
  
 private void findViews() {  
 tvResult = findViewById(R.id.*tvResult*);  
 btnQuickJob = findViewById(R.id.*btnQuickJob*);  
 btnSlowJob = findViewById(R.id.*btnSlowJob*);  
 }  
  
 // ----------------------------  
 // ASYNCTASK LỚP CON  
 // ----------------------------  
 private class SlowTask extends AsyncTask<Integer, Integer, String> {  
  
 @Override  
 protected void onPreExecute() {  
 super.onPreExecute();  
 progressDialog = new ProgressDialog(AsyncActivity.this);  
 progressDialog.setTitle("Processing");  
 progressDialog.setMessage("Please wait...");  
 progressDialog.setCancelable(false);  
 progressDialog.show();  
 }  
  
 @Override  
 protected String doInBackground(Integer... params) {  
 int count = params[0];  
 for (int i = 1; i <= count; i++) {  
 try {  
 Thread.*sleep*(500); // mô phỏng công việc nặng  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 publishProgress(i \* 100 / count); // báo tiến độ  
 }  
 return "Background Work is Done!";  
 }  
  
 @Override  
 protected void onProgressUpdate(Integer... values) {  
 super.onProgressUpdate(values);  
 progressDialog.setMessage("Working... " + values[0] + "%");  
 }  
  
 @Override  
 protected void onPostExecute(String result) {  
 super.onPostExecute(result);  
 if (progressDialog != null && progressDialog.isShowing()) {  
 progressDialog.dismiss();  
 }  
 tvResult.setText(result);  
 Toast.*makeText*(AsyncActivity.this, result, Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
}

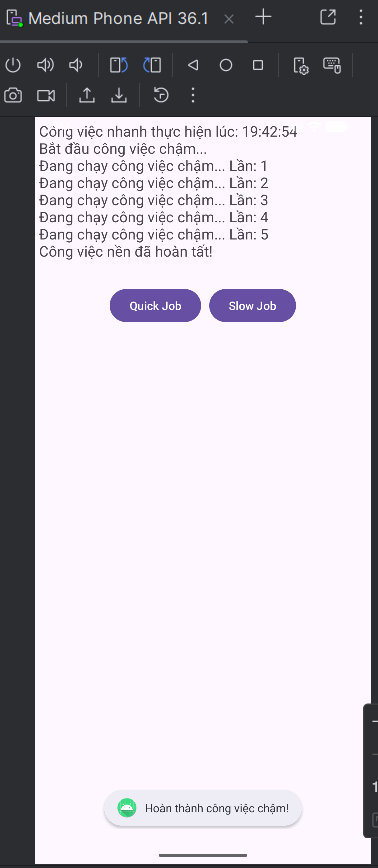


**Quickjob:**



**Slowjob:**





**Câu 4:** **Tự viết 1 ứng dụng sử dụng Asynctask để phát nhạc**

* Dùng AsyncTask để điều khiển MediaPlayer phát file nhạc.
* Có các nút Play – Pause – Stop.
* Cập nhật TextView hiển thị thời gian phát hiện tại bằng publishProgress()

**Bước 1:** **Tạo thư mục và file nhạc (VD: music.mp3):**

**->** app/src/main/res/raw/

**->** Tải 1 file nhạc mp3 về máy, copy file đó vào file raw

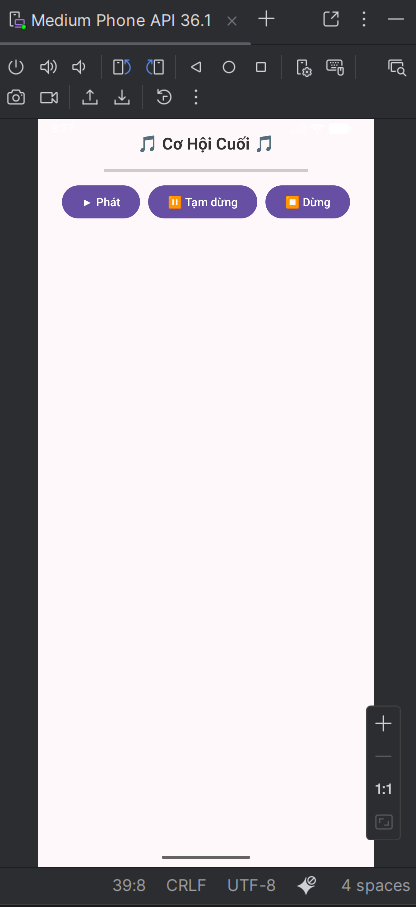
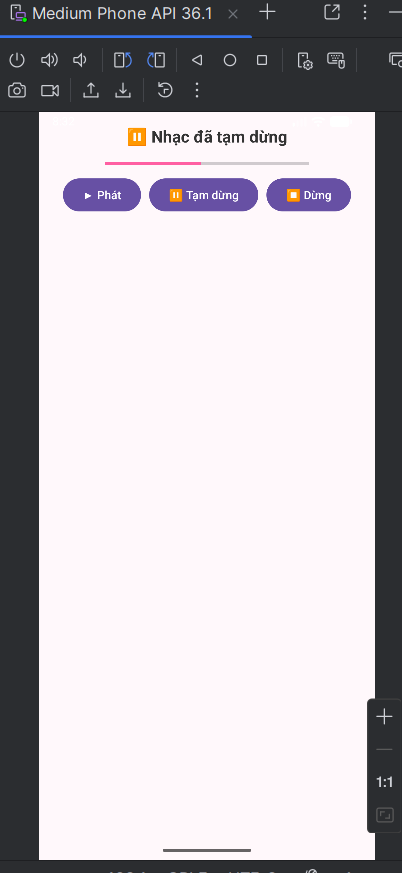
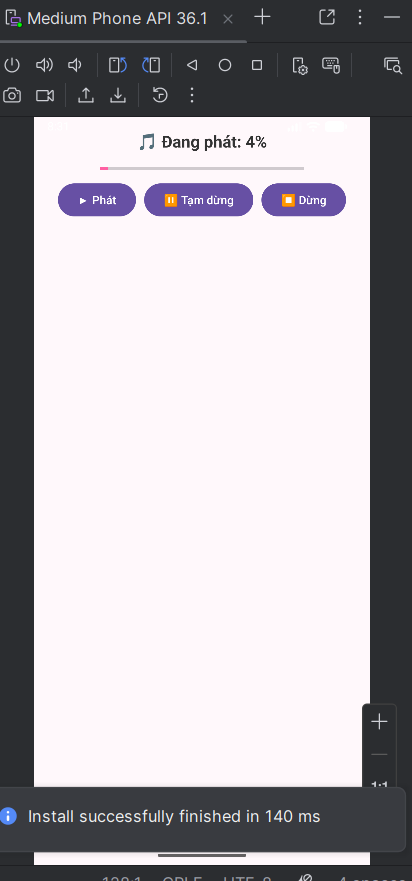
**Bước 2: Tạo Layout activity\_music.xml**

<?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:orientation="vertical"  
 android:gravity="center\_horizontal"  
 android:padding="16dp"  
 android:layout\_width="match\_parent"  
 android:layout\_height="match\_parent"  
 android:background="#FFF8FB"  
 tools:context=".MusicActivity">  
  
 <TextView  
 android:id="@+id/tvMusicStatus"  
 android:text="🎵 Cơ Hội Cuối 🎵"  
 android:textColor="#333"  
 android:textStyle="bold"  
 android:textSize="20sp"  
 android:layout\_marginBottom="12dp"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content" />  
  
 <!-- Thanh tiến trình phát nhạc -->  
 <ProgressBar  
 android:id="@+id/progressBar"  
 style="?android:attr/progressBarStyleHorizontal"  
 android:max="100"  
 android:layout\_width="250dp"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginBottom="6dp"  
 android:progressTint="#FF5FA2"  
 android:indeterminate="false" />  
  
  
 <!-- Các nút điều khiển -->  
 <LinearLayout  
 android:orientation="horizontal"  
 android:gravity="center"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content">  
  
 <Button  
 android:id="@+id/btnPlay"  
 android:text="▶ Phát"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="10dp" />  
  
 <Button  
 android:id="@+id/btnPause"  
 android:text="⏸ Tạm dừng"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content"  
 android:layout\_marginEnd="10dp" />  
  
 <Button  
 android:id="@+id/btnStop"  
 android:text="⏹ Dừng"  
 android:layout\_width="wrap\_content"  
 android:layout\_height="wrap\_content" />  
 </LinearLayout>  
</LinearLayout>

**Bước 3: Code MusicActivity**

package com.example.btlythuyet6;  
  
import androidx.appcompat.app.AppCompatActivity;  
import android.os.AsyncTask;  
import android.os.Bundle;  
import android.media.MediaPlayer;  
import android.widget.Button;  
import android.widget.ProgressBar;  
import android.widget.TextView;  
import android.widget.Toast;  
  
public class MusicActivity extends AppCompatActivity {  
  
 private TextView tvMusicStatus;  
 private Button btnPlay, btnPause, btnStop;  
 private ProgressBar progressBar;  
  
 private MediaPlayer mediaPlayer;  
 private MusicTask musicTask;  
 private boolean isPlaying = false;  
 private boolean isStopped = false;  
  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.*activity\_music*);  
  
 tvMusicStatus = findViewById(R.id.*tvMusicStatus*);  
 btnPlay = findViewById(R.id.*btnPlay*);  
 btnPause = findViewById(R.id.*btnPause*);  
 btnStop = findViewById(R.id.*btnStop*);  
 progressBar = findViewById(R.id.*progressBar*);  
  
 btnPlay.setOnClickListener(v -> startMusic());  
 btnPause.setOnClickListener(v -> pauseMusic());  
 btnStop.setOnClickListener(v -> stopMusic());  
 }  
  
 private void startMusic() {  
 if (musicTask == null || isStopped) {  
 musicTask = new MusicTask();  
 musicTask.execute();  
 isStopped = false;  
 } else if (!isPlaying && mediaPlayer != null) {  
 mediaPlayer.start();  
 isPlaying = true;  
 tvMusicStatus.setText("🎶 Tiếp tục phát nhạc...");  
 } else {  
 Toast.*makeText*(this, "Nhạc đang phát rồi nè 🎧", Toast.*LENGTH\_SHORT*).show();  
 }  
 }  
  
 private void pauseMusic() {  
 if (isPlaying && mediaPlayer != null) {  
 mediaPlayer.pause();  
 isPlaying = false;  
 tvMusicStatus.setText("⏸ Nhạc đã tạm dừng");  
 }  
 }  
  
 private void stopMusic() {  
 if (mediaPlayer != null) {  
 mediaPlayer.stop();  
 mediaPlayer.release();  
 mediaPlayer = null;  
 isPlaying = false;  
 isStopped = true;  
 tvMusicStatus.setText("⏹ Nhạc đã dừng");  
 }  
 }  
  
 private class MusicTask extends AsyncTask<Void, Integer, Void> {  
  
 @Override  
 protected void onPreExecute() {  
 super.onPreExecute();  
 tvMusicStatus.setText("🎧 Chuẩn bị phát nhạc...");  
 mediaPlayer = MediaPlayer.*create*(MusicActivity.this, R.raw.*cohoicuoi*);  
 mediaPlayer.setVolume(1.0f, 1.0f); // tăng âm lượng max  
  
 progressBar.setProgress(0);  
  
 mediaPlayer.setOnCompletionListener(mp -> {  
 tvMusicStatus.setText("✨ Bài hát đã phát xong!");  
 isPlaying = false;  
 });  
 }  
  
 @Override  
 protected Void doInBackground(Void... voids) {  
 mediaPlayer.start();  
 isPlaying = true;  
  
 int duration = mediaPlayer.getDuration() / 1000; // tổng thời gian (giây)  
 int current;  
  
 while (isPlaying && mediaPlayer != null && !isStopped) {  
 try {  
 Thread.*sleep*(1000);  
 current = mediaPlayer.getCurrentPosition() / 1000;  
 int progress = (int) ((current / (float) duration) \* 100);  
 publishProgress(progress);  
 } catch (InterruptedException e) {  
 e.printStackTrace();  
 }  
 }  
 return null;  
 }  
  
 @Override  
 protected void onProgressUpdate(Integer... values) {  
 progressBar.setProgress(values[0]);  
 tvMusicStatus.setText("🎵 Đang phát: " + values[0] + "%");  
 }  
  
 @Override  
 protected void onPostExecute(Void unused) {  
 tvMusicStatus.setText("🎶 Phát nhạc hoàn tất!");  
 }  
 }  
  
 @Override  
 protected void onDestroy() {  
 super.onDestroy();  
 if (mediaPlayer != null) mediaPlayer.release();  
 }  
}

* **Kết quả:**

**Câu 5:** Viết lại ứng dụng bài 4 sử dụng RxJava, Rx Android.

**Bước 1: Thêm thư viện** RxJava, Rx Android vào trong file **build.gradle (Module: app)**  
dependencies {  
  
 implementation libs.appcompat  
 implementation libs.material  
 implementation libs.activity  
 implementation libs.constraintlayout  
 testImplementation libs.junit  
 androidTestImplementation libs.ext.junit  
 androidTestImplementation libs.espresso.core  
 implementation 'io.reactivex.rxjava3:rxjava:3.1.8'  
 implementation 'io.reactivex.rxjava3:rxandroid:3.0.2'  
  
}  
- Rồi bấm sync now để studio tải về thư viện.

**Bước 2: Giữ lại layout cũ của câu 4  
Bước 3: Code Java – MusicRxActivity.java**

package com.example.btlythuyet6;

import androidx.appcompat.app.AppCompatActivity;

import android.media.MediaPlayer;

import android.os.Bundle;

import android.widget.Button;

import android.widget.ProgressBar;

import android.widget.TextView;

import android.widget.Toast;

import java.util.concurrent.TimeUnit;

import io.reactivex.rxjava3.android.schedulers.AndroidSchedulers;

import io.reactivex.rxjava3.core.Observable;

import io.reactivex.rxjava3.disposables.Disposable;

import io.reactivex.rxjava3.schedulers.Schedulers;

public class MusicRxActivity extends AppCompatActivity {

private TextView tvMusicStatus;

private Button btnPlay, btnPause, btnStop;

private ProgressBar progressBar;

private MediaPlayer mediaPlayer;

private Disposable disposable;

private boolean isPlaying = false;

private boolean isStopped = false;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_music);

tvMusicStatus = findViewById(R.id.tvMusicStatus);

btnPlay = findViewById(R.id.btnPlay);

btnPause = findViewById(R.id.btnPause);

btnStop = findViewById(R.id.btnStop);

progressBar = findViewById(R.id.progressBar);

btnPlay.setOnClickListener(v -> startMusic());

btnPause.setOnClickListener(v -> pauseMusic());

btnStop.setOnClickListener(v -> stopMusic());

}

private void startMusic() {

if (mediaPlayer == null || isStopped) {

tvMusicStatus.setText("🎧 Chuẩn bị phát nhạc...");

mediaPlayer = MediaPlayer.create(this, R.raw.cohoicuoi);

mediaPlayer.setVolume(1.0f, 1.0f);

progressBar.setProgress(0);

mediaPlayer.setOnCompletionListener(mp -> {

tvMusicStatus.setText("✨ Bài hát đã phát xong!");

stopProgress();

isPlaying = false;

});

mediaPlayer.start();

isPlaying = true;

isStopped = false;

startProgress();

} else if (!isPlaying) {

mediaPlayer.start();

isPlaying = true;

tvMusicStatus.setText("🎵 Tiếp tục phát nhạc...");

} else {

Toast.makeText(this, "Nhạc đang phát rồi nè 🎶", Toast.LENGTH\_SHORT).show();

}

}

private void pauseMusic() {

if (isPlaying && mediaPlayer != null) {

mediaPlayer.pause();

isPlaying = false;

tvMusicStatus.setText("⏸ Nhạc đã tạm dừng");

}

}

private void stopMusic() {

if (mediaPlayer != null) {

mediaPlayer.stop();

mediaPlayer.release();

mediaPlayer = null;

isPlaying = false;

isStopped = true;

tvMusicStatus.setText("⏹ Nhạc đã dừng");

stopProgress();

}

}

// --- RxJava: cập nhật tiến trình ---

private void startProgress() {

int duration = mediaPlayer.getDuration() / 1000; // tổng thời gian (giây)

disposable = Observable.interval(1, TimeUnit.SECONDS)

.take(duration + 1)

.subscribeOn(Schedulers.io())

.observeOn(AndroidSchedulers.mainThread())

.subscribe(

tick -> {

if (mediaPlayer != null && isPlaying) {

int current = mediaPlayer.getCurrentPosition() / 1000;

int progress = (int) ((current / (float) duration) \* 100);

progressBar.setProgress(progress);

tvMusicStatus.setText("🎵 Đang phát: " + current + "s / " + duration + "s");

}

},

error -> tvMusicStatus.setText("⚠️ Lỗi phát nhạc!"),

() -> tvMusicStatus.setText("🎶 Phát nhạc hoàn tất!")

);

}

private void stopProgress() {

if (disposable != null && !disposable.isDisposed()) {

disposable.dispose();

}

}

@Override

protected void onDestroy() {

super.onDestroy();

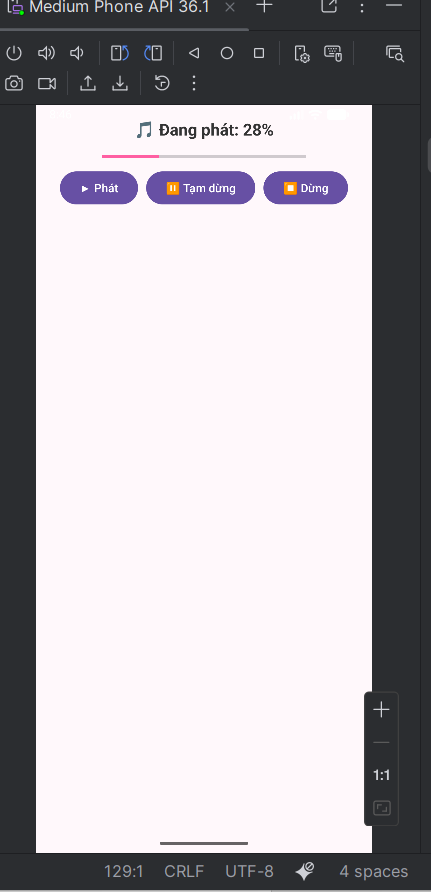
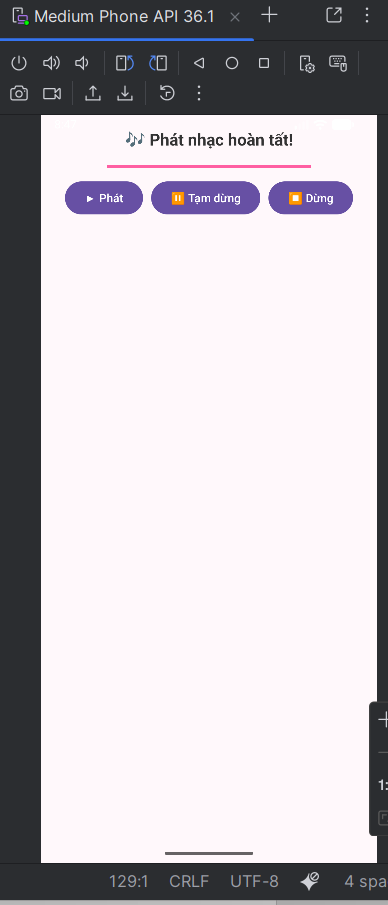
stopProgress();

if (mediaPlayer != null) mediaPlayer.release();

}

}

* **Kết quả**:

 ****

* Ở AsyncTask, ta phải tự tạo class con, chạy vòng lặp while trong doInBackground(), rồi gọi publishProgress() để cập nhật UI qua onProgressUpdate(). Mọi thứ rườm rà, bị giới hạn và dễ lỗi nếu Activity bị hủy giữa chừng.
* Còn RxJava, không cần AsyncTask nữa. Dùng Observable.interval() để phát tín hiệu mỗi giây. Dòng .subscribeOn(Schedulers.io()) cho chạy nền, còn .observeOn(AndroidSchedulers.mainThread()) cập nhật trực tiếp UI ở main thread. Dễ đọc hơn, ngắn gọn hơn, hiện đại hơn và tránh leak bộ nhớ.