

I. MỤC TIÊU

- Tìm hiểu Thread và các tác vụ bất đồng bộ
- Làm quen với AsyncTask để xử lý tác vụ bất đồng bộ

II. TÓM TẮT LÝ THUYẾT

Mỗi ứng dụng Android khi sử dụng đều được hệ điều hành cấp cho một process, và trên đó có một thread mặc định. Đó là main UI thread. Android xử lý tất cả các sự kiện/tác vụ trên một thread duy nhất gọi là main UI thread. Main UI thread không xử lý các hoạt động đồng thời vì nó chỉ xử lý một sự kiện/task tại một thời điểm. Do vậy, nếu ta thực hiện một tác vụ gì đó mà tốn nhiều thời gian trên main UI thread sẽ gây ra hiện tượng treo ứng dụng hay còn gọi là ANR (Application Not Responding).

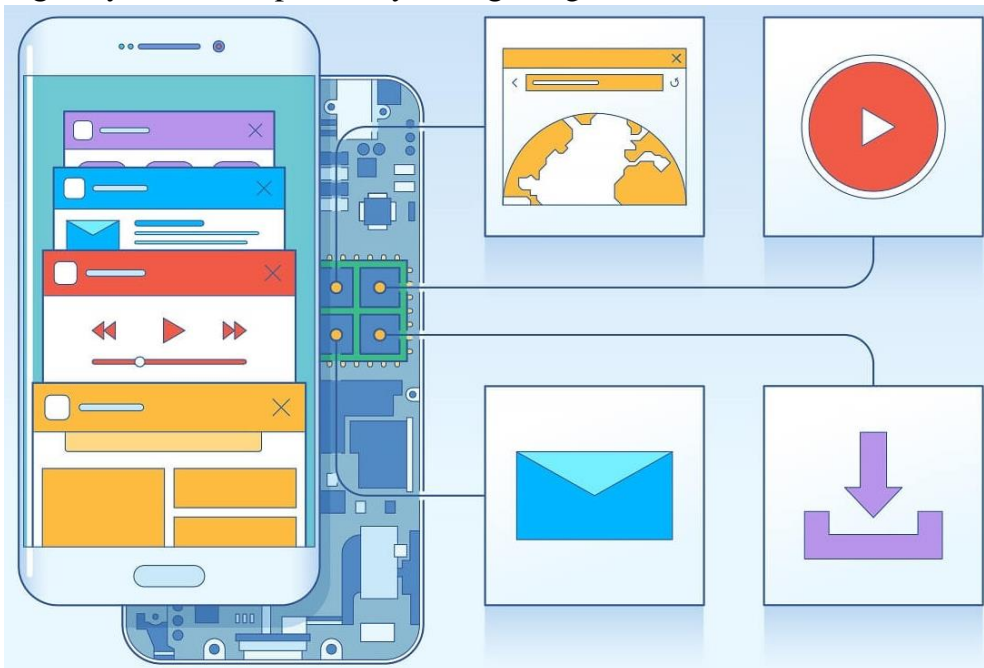
Để xử lý các tác vụ cần nhiều thời gian như: Tải file từ internet, nén hoặc giải nén... thì chúng ta phải tách tác vụ đó khỏi main UI thread(gọi là xử lý đa nhiệm). Android cung cấp một số công cụ để chúng ta có thể làm được điều đó như:

- Sử dụng Service – IntentService
- Sử dụng Thread – một khái niệm của Java
- Loader trong Android
- Hoặc sử dụng AsyncTask trong Android...

Nhưng trước hết, chúng ta cùng nhau tìm hiểu xử lý đa nhiệm trong Android là gì? Và khi nào cần phải xử lý đa nhiệm thay vì thực hiện ngay trên UI thread?

Xử lý đa nhiệm trong Android

Nếu các sự kiện hoặc một task nào đó không được xử lý đồng thời. Thì toàn bộ mã của ứng dụng Android sẽ chạy trên luồng chính và code sẽ được thực hiện tuần tự từng dòng một. Giả sử nếu bạn thực hiện một công việc/ tác vụ cần thời gian xử lý như tải nhạc từ Internet, ứng dụng sẽ hiển thị trạng thái treo cho đến khi tải xong. Để mang lại trải nghiệm người dùng tốt, tất cả tác vụ có khả năng chạy chậm đều phải chạy không đồng bộ.



Ta có thể tạm liệt kê một số tác vụ cần thời gian xử lý như:

- Truy cập tài nguyên (như MP3, JSON, Hình ảnh) từ Internet.
- Thao tác với cơ sở dữ liệu.

LAB 9: AsyncTask

- Tương tác với webService như RESTful, SOAP...
- Các Logic phức tạp mất khá nhiều thời gian như: Nén/giải nén file, sao chép/di chuyển file trong bộ nhớ...

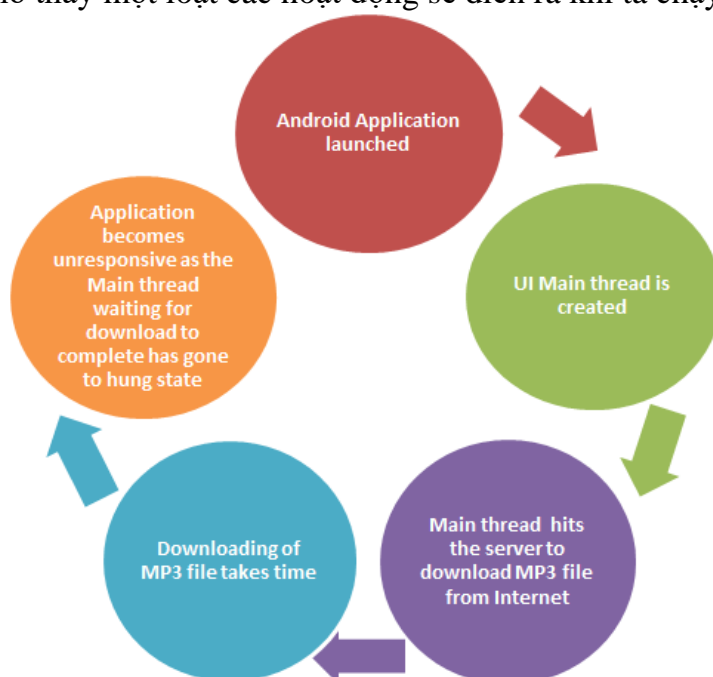
Và còn rất nhiều các trường hợp khác cần phải xử lý bất đồng bộ, đa nhiệm khác nữa. Tùy vào ứng dụng như thế nào mà ứng biến cho phù hợp. Trong Android, để xử lý các tác vụ nặng, cần nhiều thời gian, lời khuyên là ta nên sử dụng AsyncTask.

AsyncTask trong Android là gì?

AsyncTask là một abstract Android class, giúp ứng dụng Android xử lý main UI thread hiệu quả hơn. AsyncTask trong Android cho phép chúng ta thực hiện những tác vụ dài mà không ảnh hưởng đến main thread

Khi nào thì sử dụng AsyncTask?

Để dễ hình dung, giả sử ta tạo một ứng dụng Android để tải xuống tệp MP3 từ Internet. Sơ đồ trạng thái dưới đây cho thấy một loạt các hoạt động sẽ diễn ra khi ta chạy ứng dụng:



Trong khi chờ nhận file MP3 từ máy chủ, ứng dụng sẽ bị treo vì main thread vẫn đang chờ tác vụ tải xuống hoàn tất. Để khắc phục điều này, chúng ta có thể tạo thread mới và thực hiện các tác vụ trên thread mới đó. Do đó giao diện người dùng sẽ không bị ảnh hưởng và treo nữa. Nhưng việc xử lý với thread riêng biệt có thể tạo ra một số vấn đề như việc cập nhật giao diện người dùng. Bạn sẽ cần phải cập nhật trạng thái download được bao nhiêu % file đó, và khi kết thúc tải thì cũng phải cập nhật cho người dùng biết. Nếu bạn sử dụng Thread đơn giản của java thì việc cập nhật này sẽ khá phức tạp. Android đã xem xét tất cả các vấn đề này và tạo một lớp chuyên dụng có tên là AsyncTask.

Cách triển khai AsyncTask trong Android?

Tạo một class mới bên trong Activity và kế thừa từ AsyncTask như dưới đây

```
private class DownloadMp3Task extends AsyncTask<URL, Integer, Long>
{
    protected Long doInBackground(URL... urls) {
        //Yet to code
    }
    protected void onProgressUpdate(Integer... progress) {
        //Yet to code
    }
}
```

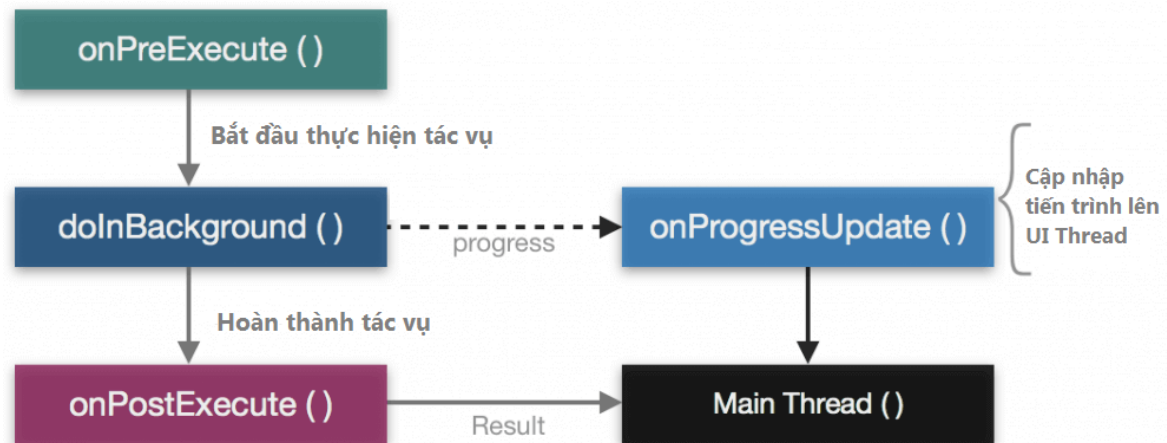
LAB 9: AsyncTask

```
}  
protected void onPostExecute(Long result) {  
    //Yet to code  
}
```

Để thực thi tác vụ, đơn giản bằng cách gọi phương thức execute

```
new DownloadMp3Task().execute(mp3URL);
```

Bản chất AsyncTask gồm có 4 bước:



Bước 1: onPreExecute()

Được thực hiện trước khi bắt đầu thực hiện tác vụ. Hàm được gọi trước phương thức `doInBackground()` và được gọi trên UI thread. Thông thường, hàm này được dùng để hiển thị thanh progressbar thông báo cho người dùng biết tác vụ bắt đầu thực hiện

Bước 2: doInBackground()

Tất cả code mà cần thời gian thực hiện sẽ được đặt trong hàm này. Vì hàm này được thực hiện ở một thread hoàn toàn riêng biệt với UI thread nên bạn không được phép cập nhật giao diện ở đây. Để có thể cập nhật giao diện khi tác vụ đang thực hiện. Ví dụ như cập nhật trạng thái % file đã download được, chúng ta sẽ phải sử dụng đến hàm bên dưới `onProgressUpdate()`

Bước 3: onProgressUpdate()

Hàm này được gọi khi trong hàm `doInBackground()` gọi đến hàm `publishProgress()`

Bước 4: onPostExecute()

Hàm này được gọi khi `doInBackground` hoàn thành công việc. Kết quả của `doInBackground()` sẽ được trả cho hàm này để hiển thị lên giao diện người dùng.

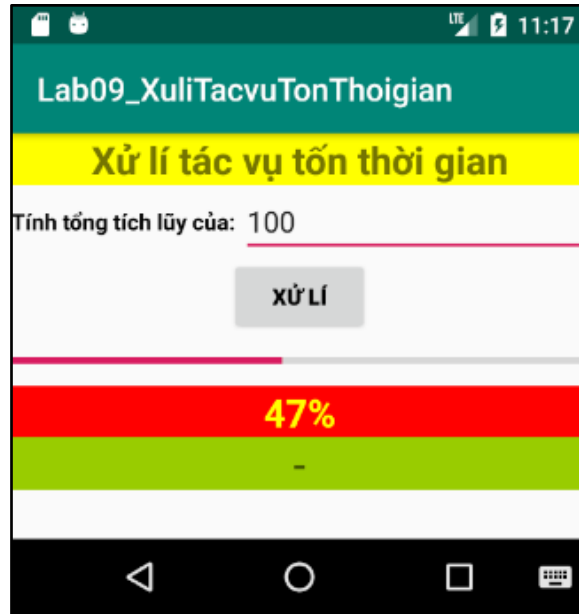
Trong quá trình AsyncTask thực hiện tác vụ, ta hoàn toàn có thể tạm dừng bất kể lúc nào mà không cần phải đợi AsyncTask làm xong. Đơn giản là ta gọi hàm `cancel(boolean)`.

Một số lưu ý về các sử dụng AsyncTask

- Lớp AsyncTask phải được thực hiện trên UI Thread
- Hàm `execute(Params...)` phải được gọi trên UI Thread
- Không nên gọi `onPreExecute()`, `onPostExecute()`, `doInBackground(Params...)`, `onProgressUpdate(Progress...)` theo cách thủ công.
- Task chỉ được thực thi một lần tại một thời điểm (Exception sẽ được throw nếu thực hiện lần thứ hai).

III. NỘI DUNG THỰC HÀNH**1. Xử lý tác vụ tốn thời gian**

Xây dựng ứng dụng mô phỏng một tác vụ cần nhiều thời gian để xử lý: nhập vào một số n , tính tổng tích lũy S của các số nguyên từ 1 đến n ($S = 1+2+3+\dots+n$). Sử dụng vòng lặp để tính S , mỗi lần lặp tạm nghỉ 100ms để mô phỏng tác vụ cần nhiều thời gian xử lý. Trong quá trình lặp, hiển thị phần trăm công việc đã làm được trên TextView và trên ProgressBar.

**activity_main.xml**

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:tools="http://schemas.android.com/tools"
4      android:layout_width="match_parent"
5      android:layout_height="match_parent"
6      android:orientation="vertical"
7      tools:context=".MainActivity">
8
9      <TextView
10         android:layout_width="match_parent"
11         android:layout_height="wrap_content"
12         android:background="#ffff00"
13         android:gravity="center"
14         android:text="Xử lý tác vụ tốn thời gian"
15         android:textSize="24sp"
16         android:textStyle="bold" />
17
18     <LinearLayout
19         android:layout_width="match_parent"
20         android:layout_height="wrap_content"
21         android:orientation="horizontal">
22
23         <TextView
24             android:layout_width="wrap_content"
25             android:layout_height="match_parent"
26             android:gravity="center_vertical"
27             android:text="Tính tổng tích lũy của: "
28             android:textColor="#000000"
29             android:textStyle="bold" />
```

```
30
31 <EditText
32     android:id="@+id/txtN"
33     android:layout_width="match_parent"
34     android:layout_height="match_parent"
35     android:inputType="number"
36     android:text="100" />
37
38 </LinearLayout>
39
40 <Button
41     android:id="@+id/btnXuli"
42     android:layout_width="wrap_content"
43     android:layout_height="wrap_content"
44     android:layout_gravity="center_horizontal"
45     android:text="Xử lý"
46     android:textColor="#000000"
47     android:textStyle="bold" />
48
49 <ProgressBar
50     android:id="@+id/progressBar"
51     style="?android:attr/progressBarStyleHorizontal"
52     android:layout_width="match_parent"
53     android:layout_height="32dp" />
54
55 <TextView
56     android:id="@+id/txtTrangthai"
57     android:layout_width="match_parent"
58     android:layout_height="wrap_content"
59     android:background="#ff0000"
60     android:gravity="center"
61     android:text="0%"
62     android:textColor="#ffff00"
63     android:textSize="24sp"
64     android:textStyle="bold" />
65
66 <TextView
67     android:id="@+id/txtKetqua"
68     android:layout_width="match_parent"
69     android:layout_height="wrap_content"
70     android:background="@android:color/holo_green_light"
71     android:gravity="center"
72     android:text="-"
73     android:textSize="24sp"
74     android:textStyle="bold" />
75 </LinearLayout>
```

MainActivity.java

```
1 package vn.edu.stu.lab09_xulitacvutonthoigian;
2
3 import android.os.Bundle;
4 import android.support.v7.app.AppCompatActivity;
5 import android.view.View;
6 import android.widget.Button;
7 import android.widget.EditText;
8 import android.widget.ProgressBar;
```

LAB 9: AsyncTask

```
9      import android.widget.TextView;
10
11      import vn.edu.stu.task.TimeConsumingTask;
12
13      public class MainActivity extends AppCompatActivity {
14          EditText txtN;
15          Button btnXuli;
16          ProgressBar progressBar;
17          TextView txtTrangthai;
18          TextView txtKetqua;
19
20
21          @Override
22          protected void onCreate(Bundle savedInstanceState) {
23              super.onCreate(savedInstanceState);
24              setContentView(R.layout.activity_main);
25              addControls();
26              addEvents();
27          }
28
29          private void addControls() {
30              txtN = findViewById(R.id.txtN);
31              btnXuli = findViewById(R.id.btnXuli);
32              progressBar = findViewById(R.id.progressBar);
33              txtTrangthai = findViewById(R.id.txtTrangthai);
34              txtKetqua = findViewById(R.id.txtKetqua);
35          }
36
37          private void addEvents() {
38              btnXuli.setOnClickListener(new View.OnClickListener() {
39                  @Override
40                  public void onClick(View v) {
41                      int n = Integer.parseInt(txtN.getText().toString());
42                      new TimeConsumingTask(
43                          txtTrangthai,
44                          progressBar,
45                          txtKetqua
46                      ).execute(n);
47                  }
48              });
49          }
50      }
```

TimeConsumingTask.java

```
1      package vn.edu.stu.task;
2
3      import android.os.AsyncTask;
4      import android.os.SystemClock;
5      import android.widget.ProgressBar;
6      import android.widget.TextView;
7
8      import java.text.DecimalFormat;
9
10     public class TimeConsumingTask extends AsyncTask<Integer, Integer, Long> {
11         TextView txtTrangthai;
12         ProgressBar progressBar;
13         TextView txtKetqua;
```



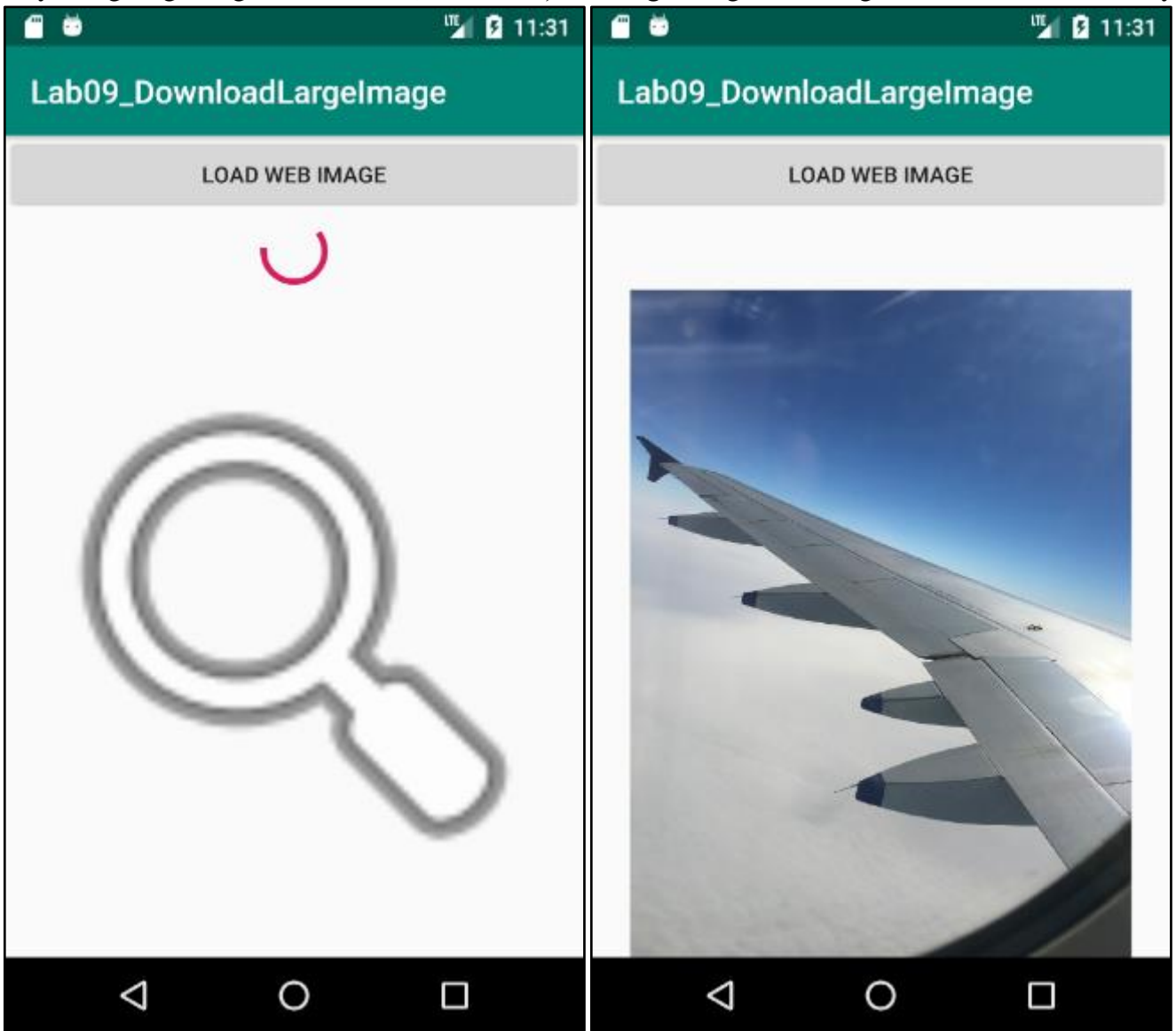
```
14 public TimeConsumingTask(TextView txtTrangthai, ProgressBar progressBar,
15                          TextView txtKetqua) {
16     this.txtTrangthai = txtTrangthai;
17     this.progressBar = progressBar;
18     this.txtKetqua = txtKetqua;
19 }
20
21 @Override
22 protected void onPreExecute() {
23     super.onPreExecute();
24     txtTrangthai.setText("0%");
25     progressBar.setProgress(0);
26 }
27
28 @Override
29 protected Long doInBackground(Integer... integers) {
30     int n = integers[0];
31     long cumulative = 0;
32     for (int i = 1; i <= n; i++) {
33         cumulative += i;
34         publishProgress(i * 100 / n);
35         SystemClock.sleep(100);
36     }
37     return cumulative;
38 }
39
40 @Override
41 protected void onProgressUpdate(Integer... values) {
42     super.onProgressUpdate(values);
43     txtTrangthai.setText(values[0] + "%");
44     progressBar.setProgress(values[0]);
45 }
46
47 @Override
48 protected void onPostExecute(Long aLong) {
49     super.onPostExecute(aLong);
50     txtTrangthai.setText("100%");
51     progressBar.setProgress(100);
52     txtKetqua.setText(new DecimalFormat("#,##0").format(aLong));
53 }
54 }
```

Yêu cầu sinh viên:

- Tiến hành cài đặt lại và chạy thử

2. Tải hình có dung lượng lớn

Xây dựng ứng dụng tải hình ảnh (từ internet) có dung lượng lớn xuống và hiển thị trên Activity

**AndroidManifest.xml**

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3          xmlns:tools="http://schemas.android.com/tools"
4          package="vn.edu.stu.lab09_downloadlargeimage">
5
6      <uses-permission android:name="android.permission.INTERNET" />
7      <application
8          android:allowBackup="true"
9          android:icon="@mipmap/ic_launcher"
10         android:label="@string/app_name"
11         android:roundIcon="@mipmap/ic_launcher_round"
12         android:supportRtl="true"
13         android:theme="@style/AppTheme"
14         tools:ignore="GoogleAppIndexingWarning">
15         <activity android:name=".MainActivity">
16             <intent-filter>
17                 <action android:name="android.intent.action.MAIN" />
18             </intent-filter>
19         </activity>
20     </application>
21 </manifest>
```


LAB 9: AsyncTask

```
19         <category android:name="android.intent.category.LAUNCHER" />
20     </intent-filter>
21 </activity>
22 </application>
23 </manifest>
```

activity_main.xml

```
1  <?xml version="1.0" encoding="utf-8"?>
2  <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3      xmlns:app="http://schemas.android.com/apk/res-auto"
4      xmlns:tools="http://schemas.android.com/tools"
5      android:layout_width="match_parent"
6      android:layout_height="match_parent"
7      android:orientation="vertical"
8      tools:context=".MainActivity">
9
10     <Button
11         android:id="@+id/btnLoad"
12         android:layout_width="match_parent"
13         android:layout_height="wrap_content"
14         android:text="Load Web Image" />
15
16     <ProgressBar
17         android:id="@+id/progressBar"
18         style="?android:attr/progressBarStyle"
19         android:layout_width="match_parent"
20         android:layout_height="wrap_content"
21         android:visibility="invisible" />
22
23     <ImageView
24         android:id="@+id/imageView"
25         android:layout_width="match_parent"
26         android:layout_height="match_parent"
27         app:srcCompat="@android:drawable/ic_menu_search" />
28 </LinearLayout>
```

MainActivity.java

```
1  package vn.edu.stu.lab09_downloadlargeimage;
2
3  import android.os.Bundle;
4  import android.support.v7.app.AppCompatActivity;
5  import android.view.View;
6  import android.widget.Button;
7  import android.widget.ImageView;
8  import android.widget.ProgressBar;
9
10 import vn.edu.stu.task.LoadWebImageTask;
11
12 public class MainActivity extends AppCompatActivity {
13     Button btnLoad;
14     ProgressBar progressBar;
15     ImageView imageView;
16
17     @Override
18     protected void onCreate(Bundle savedInstanceState) {
19         super.onCreate(savedInstanceState);
20         setContentView(R.layout.activity_main);
```

LAB 9: AsyncTask

```
21         addControls();
22         addEvents();
23     }
24
25     private void addControls() {
26         btnLoad = findViewById(R.id.btnLoad);
27         progressBar = findViewById(R.id.progressBar);
28         imageView = findViewById(R.id.imageView);
29     }
30
31     private void addEvents() {
32         btnLoad.setOnClickListener(new View.OnClickListener() {
33             @Override
34             public void onClick(View v) {
35                 // Hình ảnh được lấy từ địa chỉ:
36                 // https://sample-videos.com/download-sample-jpg-image.php
37                 String link = "https://sample-videos.com/img/Sample-jpg-image" +
38                     "-1mb.jpg";
39                 new LoadWebImageTask(
40                     progressBar,
41                     imageView
42                 ).execute(link);
43             }
44         });
45     }
46 }
```

LoadWebImageTask.java

```
1  package vn.edu.stu.task;
2
3  import android.graphics.Bitmap;
4  import android.graphics.BitmapFactory;
5  import android.os.AsyncTask;
6  import android.view.View;
7  import android.widget.ImageView;
8  import android.widget.ProgressBar;
9
10 import java.io.InputStream;
11 import java.net.HttpURLConnection;
12 import java.net.URL;
13
14 public class LoadWebImageTask extends AsyncTask<String, Void, Bitmap> {
15     ProgressBar progressBar;
16     ImageView imageView;
17
18     public LoadWebImageTask(ProgressBar progressBar, ImageView imageView) {
19         this.progressBar = progressBar;
20         this.imageView = imageView;
21     }
22
23     @Override
24     protected void onPreExecute() {
25         super.onPreExecute();
26         imageView.setImageResource(android.R.drawable.ic_menu_search);
27         progressBar.setVisibility(View.VISIBLE);
28     }
```

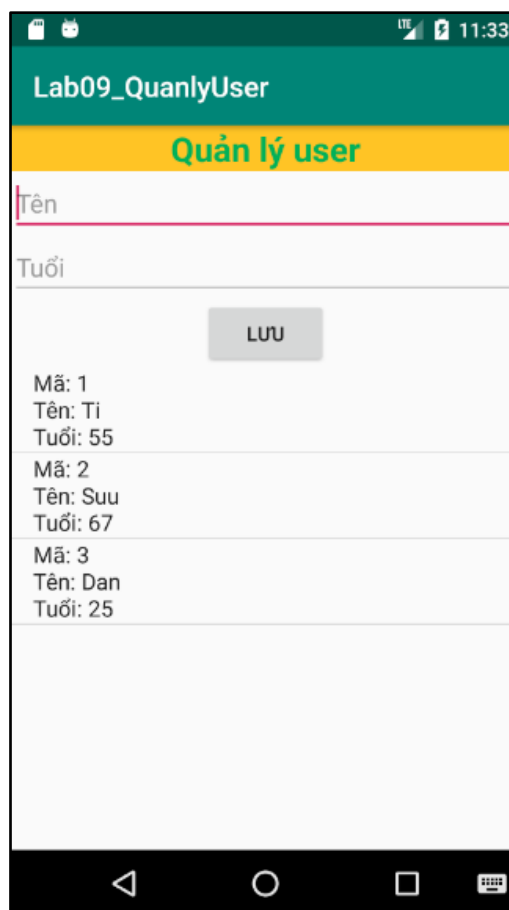
```
29
30     @Override
31     protected Bitmap doInBackground(String... strings) {
32         Bitmap bitmap = null;
33         try {
34             String link = strings[0];
35             URL url = new URL(link);
36             HttpURLConnection connection =
37                 (HttpURLConnection) url.openConnection();
38             InputStream is = connection.getInputStream();
39             bitmap = BitmapFactory.decodeStream(is);
40         } catch (Exception ex) {
41         }
42         return bitmap;
43     }
44
45     @Override
46     protected void onPostExecute(Bitmap bitmap) {
47         super.onPostExecute(bitmap);
48         imageView.setImageBitmap(bitmap);
49         progressBar.setVisibility(View.INVISIBLE);
50     }
51 }
```

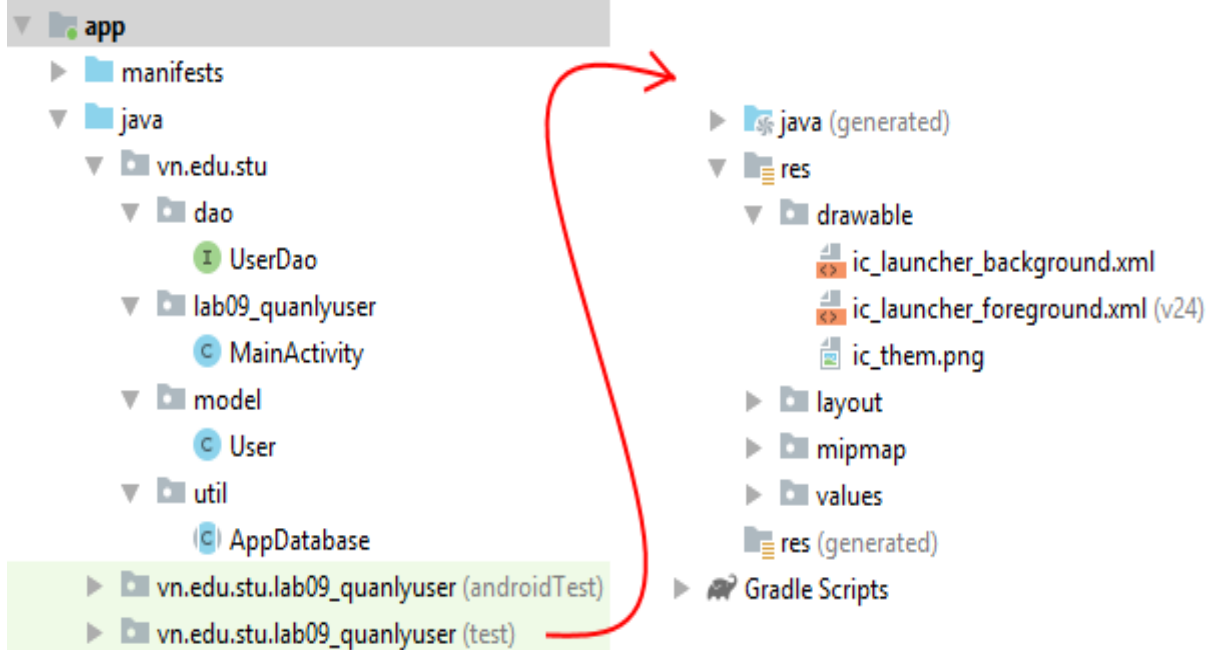
Yêu cầu sinh viên:

- Tiến hành cài đặt lại và chạy thử

3. Quản lý User

Xây dựng ứng dụng quản lý thông tin User, sử dụng SQLite database, Room và AsyncTask để truy vấn dữ liệu



Cấu trúc dự án**build.gradle (Module: app)**

```
1  apply plugin: 'com.android.application'
2
3  android {
4      compileSdkVersion 28
5      defaultConfig {
6          applicationId "vn.edu.stu.lab09_quanlyuser"
7          minSdkVersion 23
8          targetSdkVersion 28
9          versionCode 1
10         versionName "1.0"
11         testInstrumentationRunner "android.support.test.runner.AndroidJUnitRunner"
12     }
13     buildTypes {
14         release {
15             minifyEnabled false
16             proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'),
17                 'proguard-rules.pro'
18         }
19     }
20 }
21
22 dependencies {
23     implementation fileTree(dir: 'libs', include: ['*.jar'])
24     implementation 'com.android.support:appcompat-v7:28.0.0'
25     implementation 'com.android.support.constraint:constraint-layout:1.1.3'
26     testImplementation 'junit:junit:4.12'
27     androidTestImplementation 'com.android.support.test:runner:1.0.2'
28     androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
29     implementation 'com.android.support:design:28.0.0'
30     implementation 'android.arch.persistence.room:runtime:1.1.1'
31     annotationProcessor "android.arch.persistence.room:compiler:1.1.1"
32 }
```

model/User.java

```
1  package vn.edu.stu.model;
2
3  import android.arch.persistence.room.ColumnInfo;
4  import android.arch.persistence.room.Entity;
5  import android.arch.persistence.room.Ignore;
6  import android.arch.persistence.room.PrimaryKey;
7  import android.support.annotation.NonNull;
8
9  @Entity(tableName = "user")
10 public class User {
11     @PrimaryKey(autoGenerate = true)
12     private long id;
13
14     @ColumnInfo(name = "name")
15     private String name;
16
17     @ColumnInfo(name = "age")
18     private int age;
19
20     @ public User() {
21     }
22
23     @Ignore
24     @ public User(String name, int age) {
25         this.name = name;
26         this.age = age;
27     }
28
29     @Ignore
30     @ public User(long id, String name, int age) {
31         this.id = id;
32         this.name = name;
33         this.age = age;
34     }
35
36     public long getId() { return id; }
37
38     public void setId(long id) { this.id = id; }
39
40     public String getName() { return name; }
41
42     public void setName(String name) { this.name = name; }
43
44     public int getAge() { return age; }
45
46     public void setAge(int age) { this.age = age; }
47
48     @NonNull
49     @Override
50     public String toString() {
51         return "Mã: " + id
52             + "\nTên: " + name
53             + "\nTuổi: " + age;
54     }
55 }
56
57 }
```

dao/UserDao.java

```
1  package vn.edu.stu.dao;
2
3  import android.arch.persistence.room.Dao;
4  import android.arch.persistence.room.Insert;
5  import android.arch.persistence.room.Query;
6  import android.arch.persistence.room.Update;
7
8  import java.util.List;
9
10 import vn.edu.stu.model.User;
11
12 @Dao
13 public interface UserDao {
14     @Query("SELECT * FROM user")
15     List<User> getAll();
16
17     @Query("SELECT * FROM user where name LIKE :name")
18     User findByName(String name);
19
20     @Query("SELECT COUNT(*) from user")
21     int countUsers();
22
23     @Insert
24     Long insertUser(User user);
25
26     @Insert
27     List<Long> insertUsers(User... users);
28
29     @Update
30     int update(User user);
31 }
```

util/AppDatabase.java

```
1  package vn.edu.stu.util;
2
3  import android.arch.persistence.room.Database;
4  import android.arch.persistence.room.Room;
5  import android.arch.persistence.room.RoomDatabase;
6  import android.content.Context;
7
8  import vn.edu.stu.dao.UserDao;
9  import vn.edu.stu.model.User;
10
11 @Database(entities = {User.class}, version = 1, exportSchema = false)
12 public abstract class AppDatabase extends RoomDatabase {
13     private static final String DATABASE_NAME = "dbUser";
14     private static AppDatabase INSTANCE;
15
16     public abstract UserDao userDao();
17
18     public static AppDatabase getAppDatabase(Context context) {
19         if (INSTANCE == null) {
20             INSTANCE =
21                 Room.databaseBuilder(context.getApplicationContext(),
22                                     AppDatabase.class, DATABASE_NAME)
```


LAB 9: AsyncTask

```
23 // Dòng allowMainThreadQueries() cho phép truy vấn
24 // trên MainThread. Nếu dữ liệu nhỏ thì được, nhưng
25 // nếu dữ liệu lớn có thể làm cho ứng dụng bị treo
26 // lúc truy vấn. Vì vậy chỉ nên allow trong các
27 // ứng dụng demo, còn ứng dụng thực tế với
28 // dữ liệu lớn thì không nên allow, lúc này các
29 // truy vấn nên thực hiện bằng AsyncTask
30 // .allowMainThreadQueries()
31 .build();
32 }
33 return INSTANCE;
34 }
35
36 public static void destroyInstance() {
37     INSTANCE = null;
38 }
39 }
```

activity_main.xml (giảng viên cung cấp)

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <android.support.constraint.ConstraintLayout
3     xmlns:android="http://schemas.android.com/apk/res/android"
4     xmlns:app="http://schemas.android.com/apk/res-auto"
5     xmlns:tools="http://schemas.android.com/tools"
6     android:layout_width="match_parent"
7     android:layout_height="match_parent"
8     tools:context=".MainActivity">
9
10     <TextView
11         android:id="@+id/textView"
12         android:layout_width="0dp"
13         android:layout_height="wrap_content"
14         android:background="#FFC425"
15         android:gravity="center"
16         android:text="Quản lý user"
17         android:textColor="#00B159"
18         android:textSize="24sp"
19         android:textStyle="bold"
20         app:layout_constraintEnd_toEndOf="parent"
21         app:layout_constraintStart_toStartOf="parent"
22         app:layout_constraintTop_toTopOf="parent" />
23
24     <EditText
25         android:id="@+id/txtAge"
26         android:layout_width="0dp"
27         android:layout_height="wrap_content"
28         android:hint="Tuổi"
29         app:layout_constraintEnd_toEndOf="parent"
30         app:layout_constraintStart_toStartOf="parent"
31         app:layout_constraintTop_toBottomOf="@id/txtName" />
32
33     <EditText
34         android:id="@+id/txtName"
35         android:layout_width="0dp"
36         android:layout_height="wrap_content"
37         android:hint="Tên"
38         app:layout_constraintEnd_toEndOf="parent"
```

LAB 9: AsyncTask

```
39         app:layout_constraintStart_toStartOf="parent"
40         app:layout_constraintTop_toBottomOf="@id/textView" />
41
42     <Button
43         android:id="@+id/btnSave"
44         android:layout_width="wrap_content"
45         android:layout_height="48dp"
46         android:text="Luu"
47         app:layout_constraintEnd_toEndOf="parent"
48         app:layout_constraintStart_toStartOf="parent"
49         app:layout_constraintTop_toBottomOf="@id/txtAge" />
50
51     <ListView
52         android:id="@+id/lvUser"
53         android:layout_width="0dp"
54         android:layout_height="0dp"
55         app:layout_constraintBottom_toBottomOf="parent"
56         app:layout_constraintEnd_toEndOf="parent"
57         app:layout_constraintStart_toStartOf="parent"
58         app:layout_constraintTop_toBottomOf="@id/btnSave" />
59 </android.support.constraint.ConstraintLayout>
```

MainActivity.java

```
1 package vn.edu.stu.lab09_quanlyuser;
2
3 import android.os.AsyncTask;
4 import android.os.Bundle;
5 import android.support.v7.app.AppCompatActivity;
6 import android.view.View;
7 import android.widget.AdapterView;
8 import android.widget.ArrayAdapter;
9 import android.widget.Button;
10 import android.widget.EditText;
11 import android.widget.ListView;
12 import android.widget.Toast;
13
14 import java.lang.ref.WeakReference;
15 import java.util.ArrayList;
16 import java.util.List;
17
18 import vn.edu.stu.model.User;
19 import vn.edu.stu.util.AppDatabase;
20
21 public class MainActivity extends AppCompatActivity {
22     AppDatabase db;
23     EditText txtName, txtAge;
24     Button btnSave;
25     ArrayList<User> dsUser;
26     ArrayAdapter<User> adapter;
27     ListView lvUser;
28     User selectedUser;
29
30     @Override
31     protected void onCreate(Bundle savedInstanceState) {
32         super.onCreate(savedInstanceState);
33         setContentView(R.layout.activity_main);
34         addControls();
```

```
35         addEvents();
36     }
37
38     private void addControls() {
39         db = AppDatabase.getAppDatabase(this);
40         txtName = findViewById(R.id.txtName);
41         txtAge = findViewById(R.id.txtAge);
42         btnSave = findViewById(R.id.btnSave);
43         dsUser = new ArrayList<>();
44         adapter = new ArrayAdapter<>(
45             MainActivity.this,
46             android.R.layout.simple_list_item_1,
47             dsUser
48         );
49         lvUser = findViewById(R.id.lvUser);
50         lvUser.setAdapter(adapter);
51         new GetAllTask(MainActivity.this).execute();
52     }
53
54     private void addEvents() {
55         lvUser.setOnItemClickListener(new AdapterView.OnItemClickListener() {
56             @Override
57             public void onItemClick(AdapterView<?> parent, View view,
58                                     int position, long id) {
59                 selectedUser = dsUser.get(position);
60                 txtName.setText(selectedUser.getName());
61                 txtAge.setText(selectedUser.getAge() + "");
62             }
63         });
64
65         lvUser.setOnItemLongClickListener(
66             new AdapterView.OnItemLongClickListener() {
67                 @Override
68                 public boolean onItemLongClick(AdapterView<?> parent, View v
69                                                 int position, long id) {
70                     // Sinh viên tự viết code xóa User
71                     return false;
72                 }
73             });
74
75         btnSave.setOnClickListener(new View.OnClickListener() {
76             @Override
77             public void onClick(View v) {
78                 if (selectedUser == null) {
79                     String name = txtName.getText().toString();
80                     int age = Integer.parseInt(txtAge.getText().toString());
81                     User user = new User(name, age);
82                     new InsertUserTask(MainActivity.this).execute(user);
83                 } else {
84                     long id = selectedUser.getId();
85                     String name = txtName.getText().toString();
86                     int age = Integer.parseInt(txtAge.getText().toString());
87                     User user = new User(id, name, age);
88                     new UpdateUserTask(MainActivity.this).execute(user);
89                 }
90             }
91         });
92     }
```

```
92     }
93
94     private static class GetAllTask extends AsyncTask<Void, Void, List<User>> {
95         private WeakReference<MainActivity> activityReference;
96
97         // only retain a weak reference to the activity
98         GetAllTask(MainActivity context) {
99             activityReference = new WeakReference<>(context);
100         }
101
102         @Override
103         protected void onPreExecute() {
104             activityReference.get().adapter.clear();
105         }
106
107         @Override
108         protected List<User> doInBackground(Void... voids) {
109             if (activityReference.get() != null)
110                 return activityReference.get().db.userDao().getAll();
111             else
112                 return null;
113         }
114
115         @Override
116         protected void onPostExecute(List<User> users) {
117             activityReference.get().dsUser.clear();
118             if (users != null && users.size() > 0) {
119                 activityReference.get().adapter.addAll(users);
120             } else {
121                 Toast.makeText(
122                     activityReference.get(),
123                     "Có lỗi xảy ra. Không thể nạp danh sách user.",
124                     Toast.LENGTH_SHORT
125                 ).show();
126             }
127             activityReference.get().adapter.notifyDataSetChanged();
128         }
129     }
130
131     private static class InsertUserTask extends AsyncTask<User, Void, User> {
132         private WeakReference<MainActivity> activityReference;
133
134         InsertUserTask(MainActivity context) {
135             activityReference = new WeakReference<>(context);
136         }
137
138         @Override
139         protected User doInBackground(User... users) {
140             if (activityReference.get() != null) {
141                 long newID = activityReference.get().db.userDao()
142                     .insertUser(users[0]);
143                 if (newID > 0) {
144                     users[0].setId(newID);
145                     return users[0];
146                 } else {
147                     return null;
148                 }
149             }
150         }
151     }
```

```
148         }
149     } else
150         return null;
151     }
152
153     @Override
154     protected void onPostExecute(User user) {
155         if (user != null) {
156             activityReference.get().adapter.add(user);
157             activityReference.get().adapter.notifyDataSetChanged();
158             Toast.makeText(
159                 activityReference.get(),
160                 "Thêm thành công. Mã mới: " + user.getId(),
161                 Toast.LENGTH_SHORT
162             ).show();
163         } else {
164             Toast.makeText(
165                 activityReference.get(),
166                 "Có lỗi xảy ra. Không thể thêm user.",
167                 Toast.LENGTH_SHORT
168             ).show();
169         }
170     }
171 }
172
173 private static class UpdateUserTask extends AsyncTask<User, Void, User> {
174     private WeakReference<MainActivity> activityReference;
175
176     UpdateUserTask(MainActivity context) {
177         activityReference = new WeakReference<>(context);
178     }
179
180     @Override
181     protected User doInBackground(User... users) {
182         if (activityReference.get() != null) {
183             int ret = activityReference.get().db.userDao().update(users[0]);
184             if (ret > 0) {
185                 return users[0];
186             } else {
187                 return null;
188             }
189         } else
190             return null;
191     }
192
193     @Override
194     protected void onPostExecute(User user) {
195         if (user != null) {
196             activityReference.get().selectedUser.setName(user.getName());
197             activityReference.get().selectedUser.setAge(user.getAge());
198             activityReference.get().adapter.notifyDataSetChanged();
199             Toast.makeText(
200                 activityReference.get(),
201                 "Cập nhật thành công.",
202                 Toast.LENGTH_SHORT
203             ).show();
204             activityReference.get().selectedUser = null;
```

```
205         } else {
206             Toast.makeText(
207                 activityReference.get(),
208                 "Có lỗi xảy ra. Không thể cập nhật user.",
209                 Toast.LENGTH_SHORT
210             ).show();
211         }
212     }
213 }
214 }
```

Yêu cầu sinh viên:

- Tiến hành cài đặt lại và chạy thử.
- Viết code xử lý Xóa User khi nhấn giữ trên item.

IV. BÀI TẬP LÀM THÊM

1. Thực hiện lại các bài lab ở Lab 7 (SQLite), sử dụng AsyncTask.
2. Thực hiện lại các bài lab ở Lab 8 (WebService), sử dụng AsyncTask.