

TRƯỜNG CAO ĐẲNG KTCN VIỆT NAM - HÀN QUỐC
KHOA CÔNG NGHỆ THÔNG TIN

GIÁO TRÌNH
LẬP TRÌNH ANDROID

Tác giả: Phan Thị Thùy Nhung

Năm 2021

MỤC LỤC

LỜI NÓI ĐẦU	5
NỘI DUNG TỔNG QUÁT VÀ PHÂN BỐ THỜI GIAN CỦA MÔ-ĐUN	7
MỤC TIÊU MÔ-ĐUN	7
NỘI DUNG TỔNG QUÁT VÀ PHÂN BỐ THỜI GIAN	7
BÀI 1: TỔNG QUAN VỀ LẬP TRÌNH TRÊN NỀN TẢNG ANDROID	9
1. MỤC TIÊU	9
2. NỘI DUNG	9
2.1. <i>Nhập môn lập trình Android</i>	9
2.2. <i>Tạo ứng dụng android đầu tiên</i>	20
BÀI TẬP	27
3. TÓM TẮT TRÌNH TỰ THỰC HIỆN HOẶC QUY TRÌNH CÔNG NGHỆ	32
BÀI 2: SỬ DỤNG NGÔN NGỮ JAVA	33
1. MỤC TIÊU	33
2. NỘI DUNG	33
2.1. <i>Lý thuyết liên quan</i>	33
2.2. <i>Trình tự thao tác</i>	35
BÀI TẬP:	37
3. TÓM TẮT TRÌNH TỰ THỰC HIỆN HOẶC QUY TRÌNH CÔNG NGHỆ	37
BÀI 3: SỬ DỤNG LAYOUT	38
1. MỤC TIÊU	38
2. NỘI DUNG	38
2.1. <i>Lý thuyết liên quan</i>	38
2.2. <i>Trình tự thao tác</i>	42
BÀI TẬP	52
3. TÓM TẮT TRÌNH TỰ THỰC HIỆN HOẶC QUY TRÌNH CÔNG NGHỆ	52
BÀI 4: THIẾT KẾ GIAO DIỆN NGƯỜI DÙNG VỚI CÁC ĐIỀU KHIỂN CƠ BẢN	53
1. MỤC TIÊU	53
2. NỘI DUNG	53

2.1. Lý thuyết liên quan	53
2.2 Trình tự thao tác	61
BÀI TẬP	74
3. TÓM TẮT TRÌNH TỰ THỰC HIỆN HOẶC QUY TRÌNH CÔNG NGHỆ	89
BÀI 5: TRUYỀN THÔNG GIỮA CÁC ACTIVITY	90
1. MỤC TIÊU	90
2. NỘI DUNG	90
2.1. Lý thuyết liên quan	90
2.2	93
BÀI TẬP	97
3. TÓM TẮT TRÌNH TỰ THỰC HIỆN HOẶC QUY TRÌNH CÔNG NGHỆ	105
BÀI 6: SỬ DỤNG MENU	106
1. MỤC TIÊU	106
2. NỘI DUNG	106
2.1. Lý thuyết liên quan	106
2.1. Trình tự thao tác	107
BÀI TẬP	112
3. TÓM TẮT TRÌNH TỰ THỰC HIỆN HOẶC QUY TRÌNH CÔNG NGHỆ	113
BÀI 7: LƯU TRỮ DỮ LIỆU	115
1. MỤC TIÊU	115
2. NỘI DUNG	115
2.1. Lý thuyết liên quan	115
2.2. Trình tự thao tác	116
BÀI TẬP	129
3. TÓM TẮT TRÌNH TỰ THỰC HIỆN HOẶC QUY TRÌNH CÔNG NGHỆ	138
BÀI 8: ĐƯA ỨNG DỤNG LÊN THIẾT BỊ ANDOIRD VÀ UPLOAD LÊN ANDROID MARKET	139
1. MỤC TIÊU	139
2. NỘI DUNG	139
2.1. Lý thuyết liên quan	139

2.2 <i>Trình tự thao tác</i>	139
3. TÓM TẮT TRÌNH TỰ THỰC HIỆN HOẶC QUY TRÌNH CÔNG NGHỆ	160
VI. HƯỚNG DẪN THỰC HIỆN MÔ ĐUN:	161
1. PHẠM VI ÁP DỤNG CHƯƠNG TRÌNH MÔ ĐUN:	161
2. HƯỚNG DẪN MỘT SỐ ĐIỂM CHÍNH VỀ PHƯƠNG PHÁP GIẢNG DẠY MÔ ĐUN ĐÀO TẠO:	
161	
3. NHỮNG TRỌNG TÂM CHƯƠNG TRÌNH MÔ ĐUN CẦN CHÚ Ý:	161
4. TÀI LIỆU CẦN THAM KHẢO:	161

LỜI NÓI ĐẦU

Trong sự phát triển của kinh tế tri thức hiện nay, không thể phủ nhận vai trò của công nghệ thông tin. Trong lĩnh vực quản trị cơ sở dữ liệu, lập trình Android đặc biệt đã được sử dụng phổ biến trong các hệ quản trị cơ sở dữ liệu thương mại và có vai trò quan trọng trong những hệ thống này.

Xuất phát từ thực tế nhu cầu học tập của sinh viên chuyên ngành CNTT nói chung và sinh viên trường Cao đẳng KTCN Việt Nam – Hàn Quốc nói riêng. Giáo trình bám sát theo đề cương môn học của trường, để phục vụ cho sinh viên năm thứ hai theo học môn “Lập trình Android”.

Giáo trình được chia thành tám bài với nội dung như sau:

Bài 1: Tổng quan về lập trình trên nền tảng Android

Bài 2: Sử dụng ngôn ngữ Java

Bài 3: Sử dụng Layout

Bài 4: Thiết kế giao diện người dùng với các điều khiển cơ bản

Bài 5: Truyền thông giữa các Activity

Bài 6: Sử dụng Menu

Bài 7: Lưu trữ dữ liệu

Bài 8: Đưa ứng dụng lên thiết bị Android và upload lên Android market.

Bài 1 giới thiệu tổng quan về lập trình Android và hướng dẫn cách cài đặt và cấu hình Java và Android Studio. Bài 2 trình bày một số khái niệm và câu lệnh cơ bản của ngôn ngữ sẽ dùng để lập trình trên Android là ngôn ngữ Java. Bài 3, bài 4 và bài 5 hướng dẫn tổng quan về lập trình trên nền tảng Android, thiết kế giao diện người dùng với các điều khiển cơ bản và Layout. Bài 5 và bài 6 được dành để bàn luận đến truyền thông giữa các Activity và sử dụng Menu. Bài 7, 8 trình bày một số cách lưu trữ dữ liệu và cách đưa ứng dụng lên thiết bị Android và upload lên Android market.

Giáo trình được trình bày vấn đề này một cách rõ ràng và đơn giản. Cách tiếp cận định lượng được sử dụng cùng với nhiều những ví dụ chi tiết. Có rất nhiều bài tập trong mỗi chương, giúp tăng cường khả năng của sinh viên để áp dụng những khái niệm vào giải quyết các vấn đề thực tế.

Trong quá trình biên soạn giáo trình này, mặc dù đã cố gắng hết sức nhưng không tránh khỏi những sai sót. Rất mong nhận được ý kiến đóng góp của độc giả để cuốn sách hoàn thiện hơn.

TÁC GIẢ

Phan Thị Thùy Nhung

NỘI DUNG TỔNG QUÁT VÀ PHÂN BỐ THỜI GIAN CỦA MÔ-ĐUN

Mục tiêu Mô-đun

Sau khi học xong bài này, người học có khả năng:

- Về kiến thức: Trình bày được các kiến thức cơ bản về tổng quan lập trình trên môi trường Android Studio và ngôn ngữ Java. Mô tả được cách thiết kế giao diện người dùng trên màn hình Android với các điều khiển cơ bản, sử dụng luân phiên được các Layout của Android. Thiết lập được cách truyền thông tin giữa các Activity và tạo được Menu cho giao diện Android. Tổng hợp được các cách lưu trữ dữ liệu và xây dựng được ứng dụng truy xuất cơ sở dữ liệu SQLite và đưa được ứng dụng lên thiết bị Android và lên Android market.
- Về kỹ năng: Khai thác và sử dụng thành thạo các đối tượng, công cụ để thiết kế giao diện chạy trên hệ điều hành Android. Sử dụng được các cách thức lưu trữ dữ liệu và xây dựng được ứng dụng truy xuất cơ sở dữ liệu SQLite. Hoàn thành các bài tập ứng dụng đáp ứng yêu cầu. Phân tích, thiết kế và triển khai các ứng dụng thực tế và Upload lên Android market.
- Về thái độ: Hình thành được đức tính cẩn thận, có ý thức trách nhiệm trong công việc. Nhận thức được vai trò của ứng dụng trên thiết bị di động trong cuộc sống. Chịu trách nhiệm đối với kết quả mình làm ra. Rèn luyện kỹ năng và phát triển tính sáng tạo.

Nội dung tổng quát và phân bố thời gian

TT	Tên các bài trong Mô-đun	Thời gian			
		Tổng số	Lý thuyết	Thực hành	Kiểm tra
1	Bài 1: Tổng quan về lập trình trên nền tảng Android	8	2	6	
2	Bài 2: Sử dụng ngôn ngữ Java	4	2	2	

3	Bài 3: Sử dụng Layout	8	2	6	
4	Bài 4: Thiết kế giao diện người dùng với các điều khiển cơ bản	16	2	14	
5	Bài 5: Truyền thông giữa các Activity	12	2	8	2
6	Bài 6: Sử dụng Menu	8	2	6	
7	Bài 7: Lưu trữ dữ liệu	16	2	14	
8	Bài 8: Đưa ứng dụng lên thiết bị Andoird và upload lên Android market.	8	2	4	2
	Cộng	80	16	60	4

BÀI 1: TỔNG QUAN VỀ LẬP TRÌNH TRÊN NỀN TẢNG ANDROID

1. MỤC TIÊU

Sau khi học xong bài này, người học có khả năng:

- Về kiến thức: Trình bày được các kiến thức cơ bản về tổng quan lập trình trên môi trường Android Studio. Tổng hợp được cách cài đặt môi trường lập trình Android, tạo được ứng dụng đơn giản nhất. Trình bày được cấu trúc của một dự án Android.
- Về kỹ năng: Sử dụng được kiến thức để cài đặt được Android Studio. Khai thác và sử dụng thành thạo các chức năng cơ bản của ứng dụng Android Studio. Hoàn thành các bài tập ứng dụng đáp ứng yêu cầu.
- Về thái độ: Nhận thức được vai trò của ứng dụng trên thiết bị di động trong cuộc sống. Chịu trách nhiệm đối với kết quả mình làm ra. Rèn luyện kỹ năng và phát triển tính sáng tạo.

2. NỘI DUNG

2.1. Nhập môn lập trình Android

2.1.1 Lý thuyết liên quan

Android là hệ điều hành trên điện thoại di động (và hiện nay là cả trên một số đầu phát HD, HD Player, TV) phát triển bởi Google và dựa trên nền tảng Linux.

Một số hệ điều hành cùng với ngôn ngữ lập trình và IDE phổ biến nhất của nó.

HĐH	Ngôn ngữ lập trình	IDE
Android	Java	IBM's Eclipse với Google's ADT plugins
iOS	Objective-C	Apple' Xcode
Windows Phone	C#	Microsoft's Visual Studio cho Windows phone

Giao diện: Giao diện người dùng của Android dựa trên nguyên tắc tác động trực tiếp, sử dụng cảm ứng chạm tương tự như những động tác ngoài đời thực như vuốt, chạm,

kéo dẫn và thu lại để xử lý các đối tượng trên màn hình. Ở phía trên cùng màn hình là thanh trạng thái, hiển thị thông tin về thiết bị và tình trạng kết nối. Một điểm mạnh của hệ điều hành Android là cho phép người dùng chạy nhiều ứng dụng cùng lúc (đa nhiệm - multi-tasking).

Ứng dụng: Android có lượng ứng dụng của bên thứ ba được chọn lọc và đặt trên một cửa hàng ứng dụng như Google Play để người dùng lấy về. Các ứng dụng cho Android được phát triển bằng ngôn ngữ Java sử dụng bộ phát triển phần mềm Android.

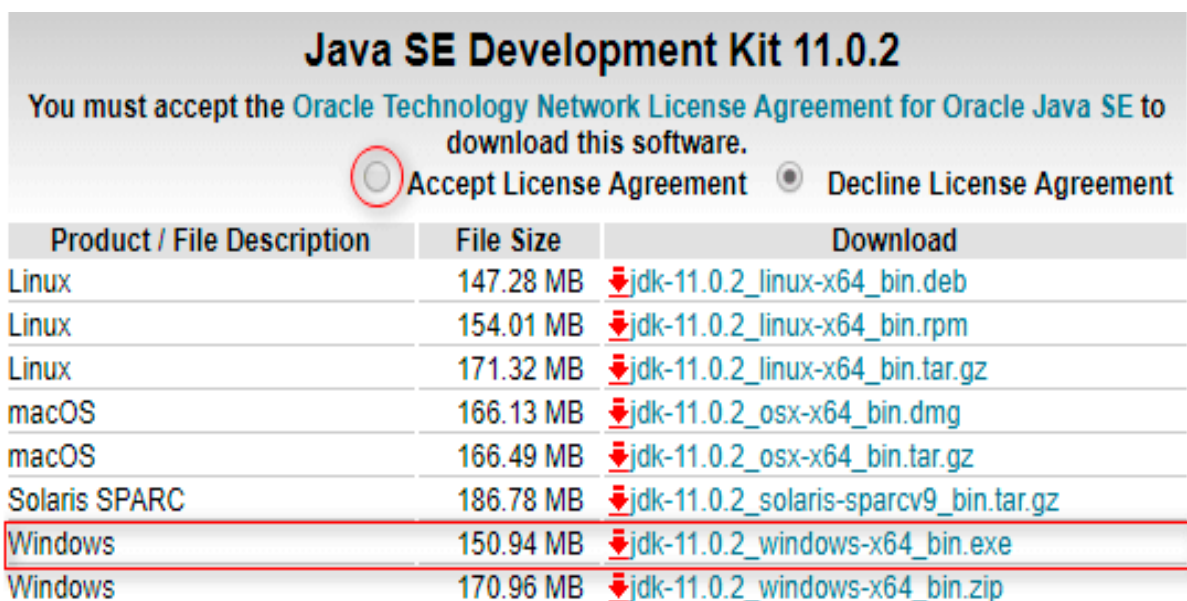
Quản lý bộ nhớ: Vì các thiết bị Android chủ yếu chạy bằng pin, nên Android được thiết kế để quản lý bộ nhớ (RAM) nhằm giảm tối đa tiêu thụ điện năng, trái với hệ điều hành máy tính để bàn luôn có nguồn điện không giới hạn. Khi một ứng dụng Android không còn được sử dụng, hệ thống sẽ tự động ngưng nó trong bộ nhớ, trong khi ứng dụng về mặt kỹ thuật vẫn "mở", những ứng dụng này sẽ không tiêu thụ bất cứ tài nguyên nào và nằm đó cho đến khi nó được cần đến.

2.1.2 Trình tự thao tác

a. Cài đặt JDK

Bước 1: Download JDK

Vào trang tải Java JDK của Oracle để tải phiên bản phù hợp nhất, link download <http://www.oracle.com/technetwork/java/javase/downloads/index.html>



Java SE Development Kit 11.0.2

You must accept the [Oracle Technology Network License Agreement for Oracle Java SE](#) to download this software.

Accept License Agreement Decline License Agreement

Product / File Description	File Size	Download
Linux	147.28 MB	jdk-11.0.2_linux-x64_bin.deb
Linux	154.01 MB	jdk-11.0.2_linux-x64_bin.rpm
Linux	171.32 MB	jdk-11.0.2_linux-x64_bin.tar.gz
macOS	166.13 MB	jdk-11.0.2_osx-x64_bin.dmg
macOS	166.49 MB	jdk-11.0.2_osx-x64_bin.tar.gz
Solaris SPARC	186.78 MB	jdk-11.0.2_solaris-sparcv9_bin.tar.gz
Windows	150.94 MB	jdk-11.0.2_windows-x64_bin.exe
Windows	170.96 MB	jdk-11.0.2_windows-x64_bin.zip

Nên kiểm tra xem hệ điều hành của mình là 32bit hay 64bit để download bộ cài **JDK** phù hợp.

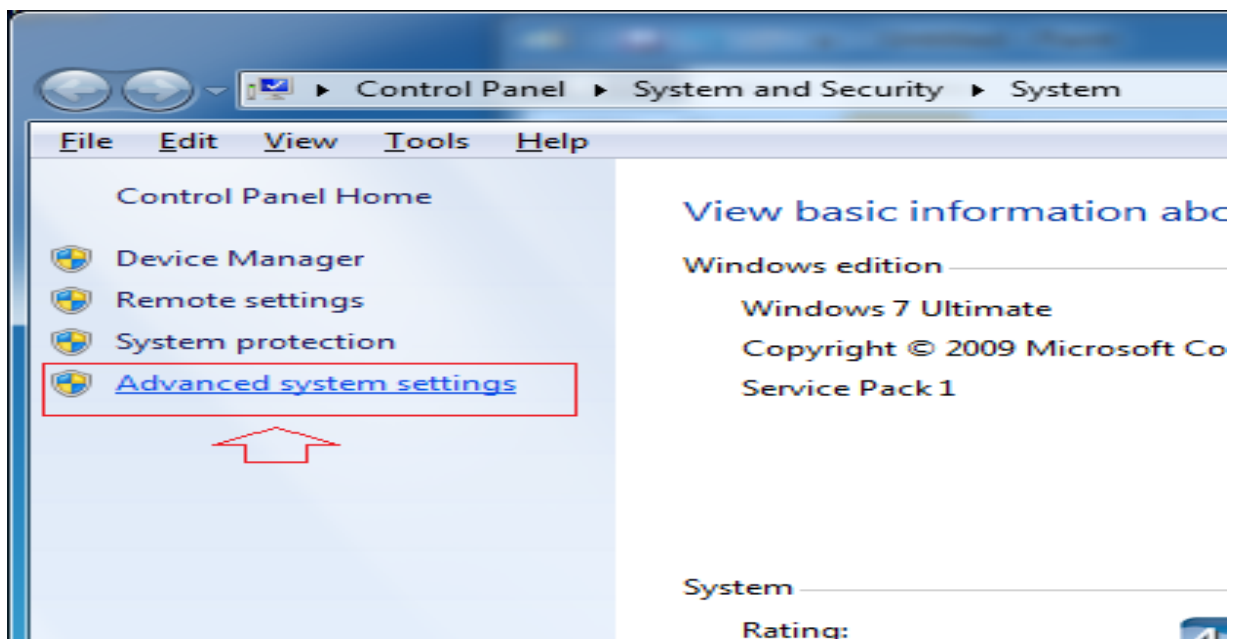
Bước 2: Cài đặt Java

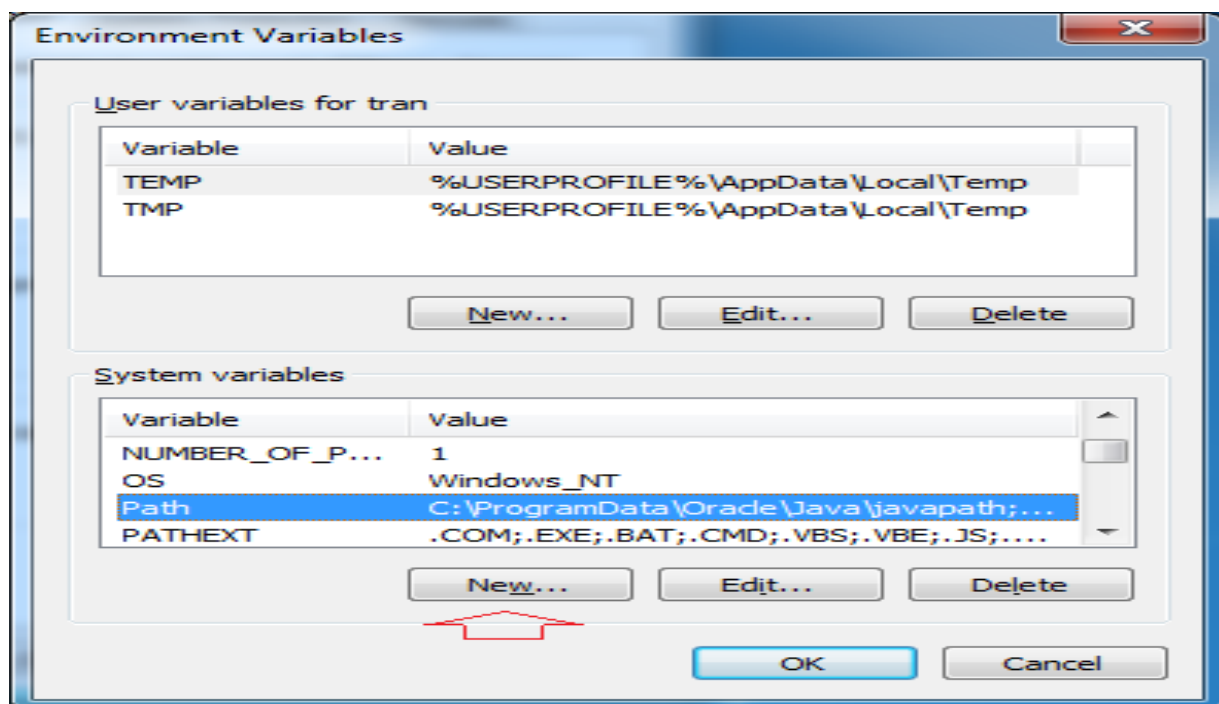
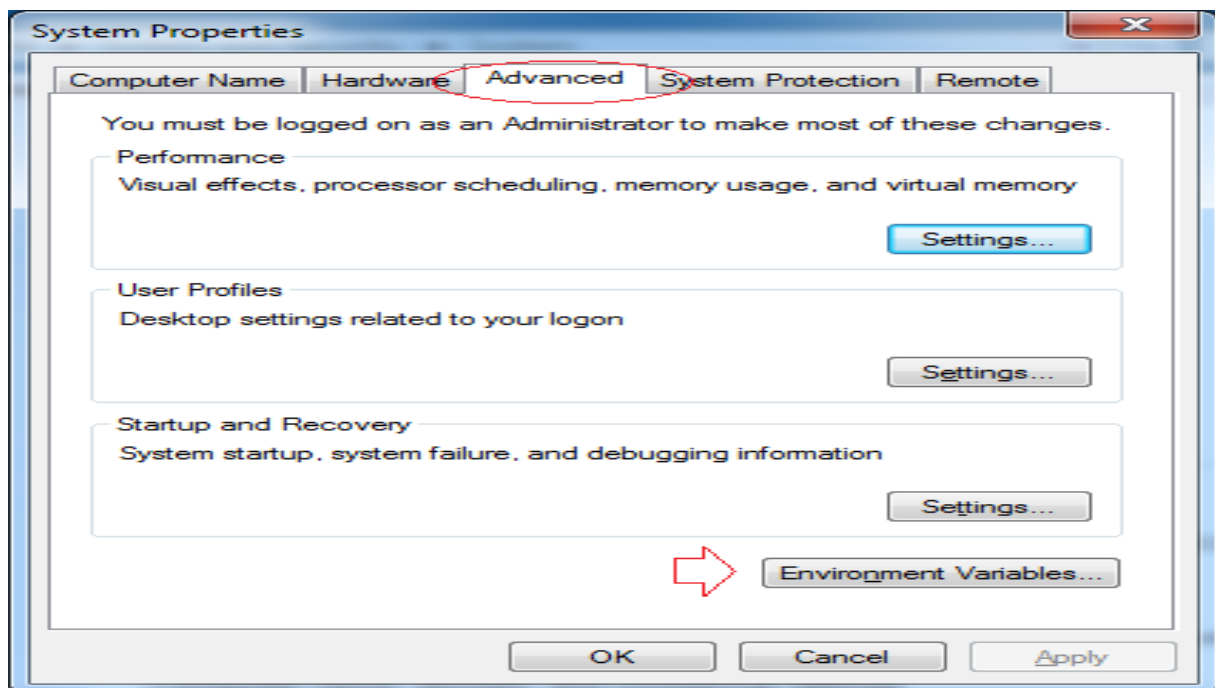
Khi quá trình tải xuống hoàn tất, hãy chạy file “.exe” để cài đặt Java JDK. Nhấn Next. Sau khi cài đặt hoàn tất, nhấn Close.

Bước 3: Cấu hình biến môi trường cho Java

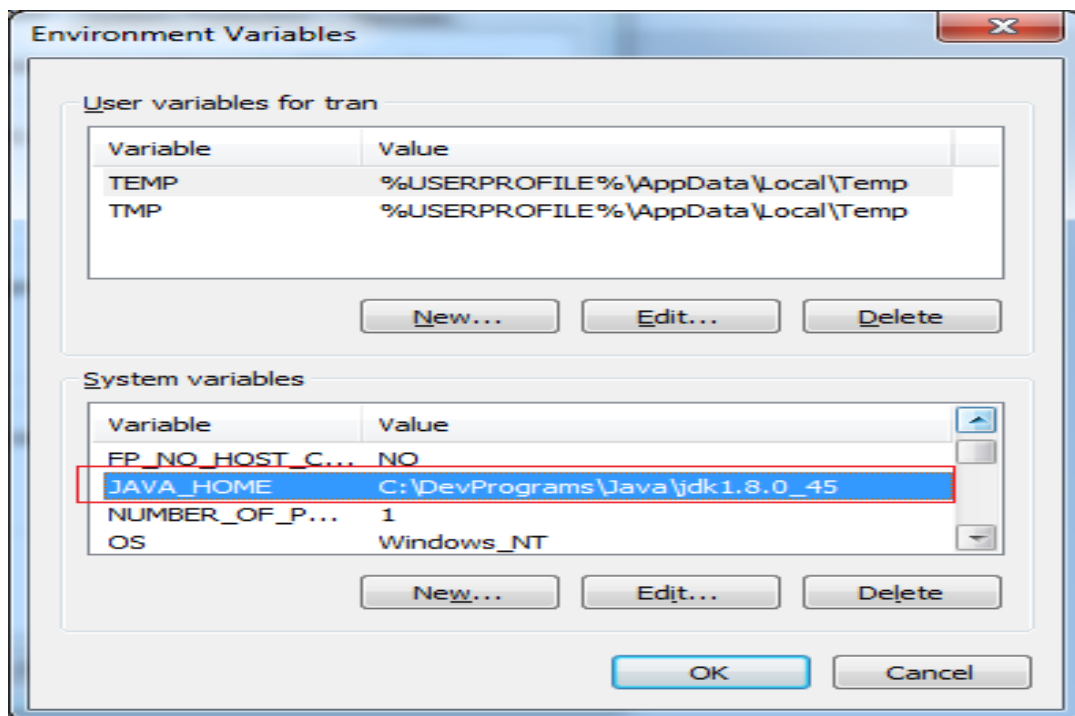
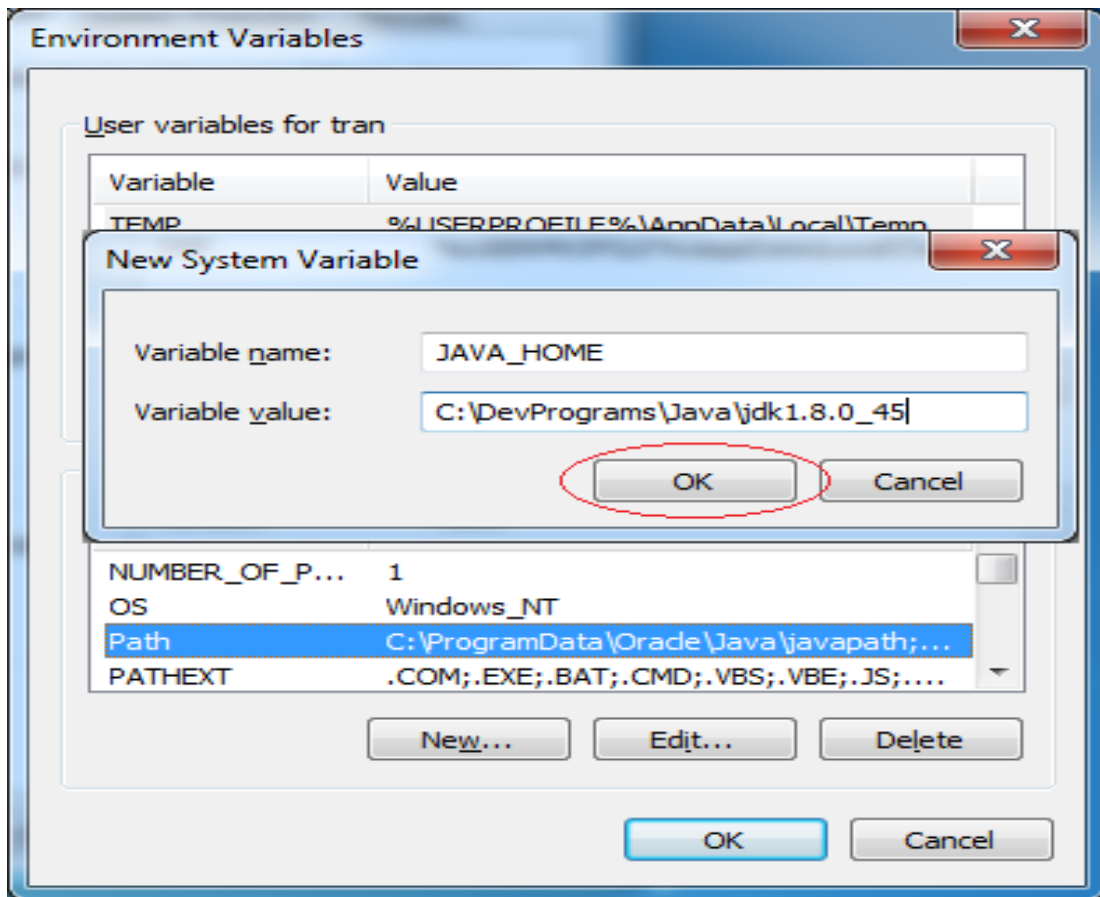
Việc cấu hình biến môi trường cho Java là không bắt buộc, nhưng nếu trên máy tính cài đặt nhiều phiên bản Java, thì việc cấu hình là cần thiết để xác định phiên bản java nào mặc định được sử dụng.

Nhấn phải chuột vào Computer, chọn **Properties** > Click vào **Advanced system settings** > Click vào **Environment Variables** > Tại Khung làm việc **User variables for Admin** > Chọn **New**.

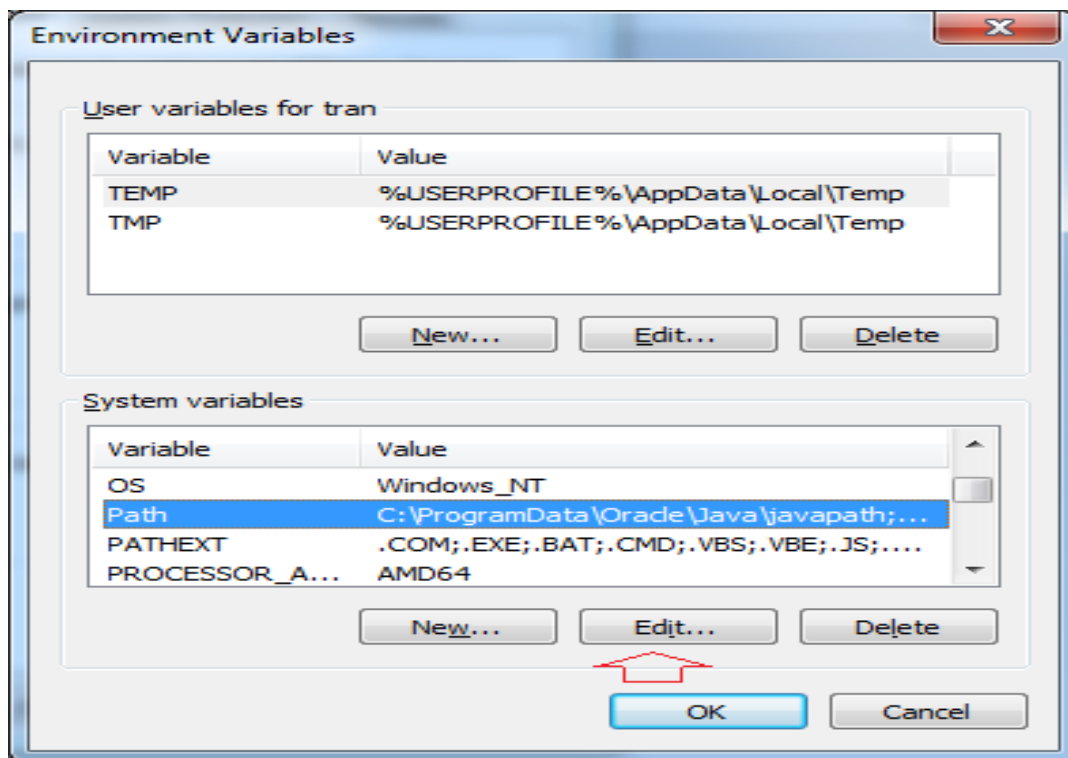




Nhập **PATH** trong mục **Variable name**, sao chép đường dẫn của thư mục bin được cài đặt trong thư mục JDK.

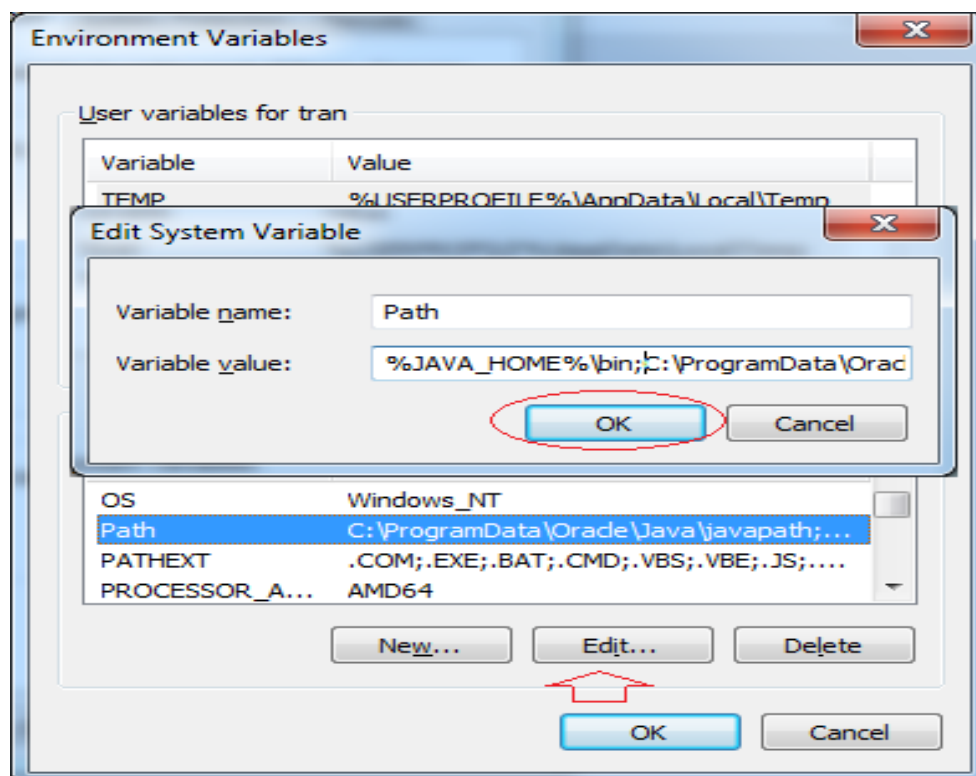


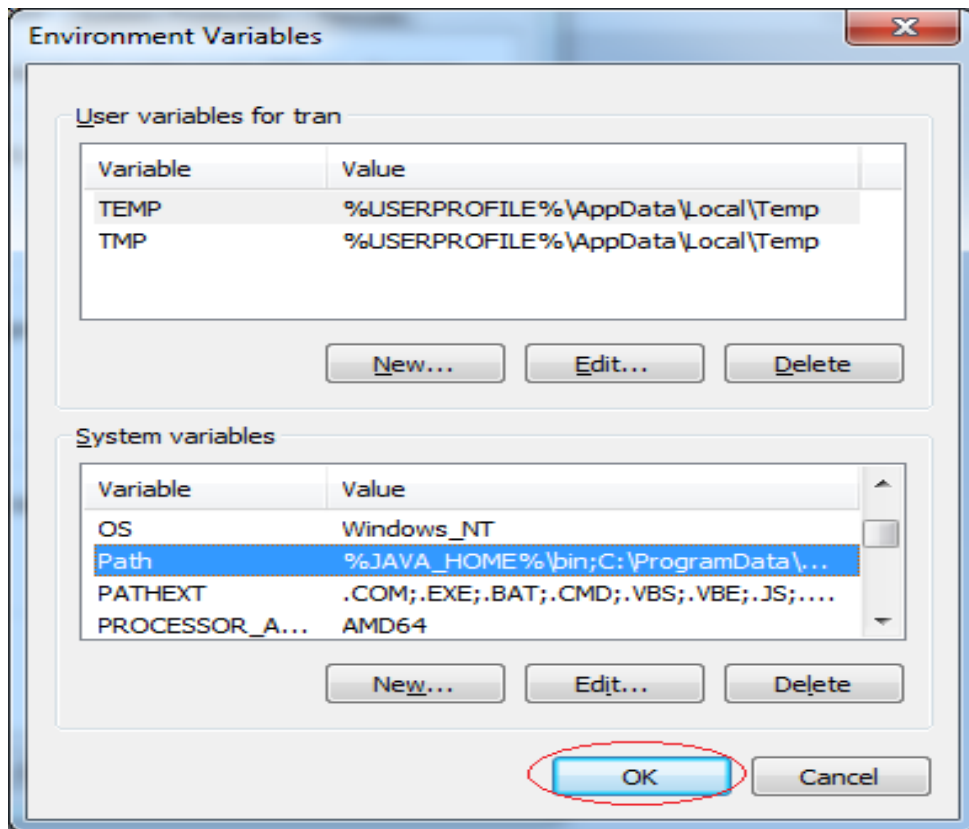
Sửa đổi biến môi trường *path*:



Thêm vào phía trước giá trị của biến môi trường *path*:

%JAVA_HOME%\bin;





Bước 4: Kiểm tra kết quả cài đặt

Gõ phím “Windows + R” > gõ “cmd” > gõ câu lệnh “java -version”. Nếu màn hình hiển thị version của Java là cài đặt thành công.

```
cmd Command Prompt
Microsoft Windows [Version 10.0.18363.959]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\tc>java -version
java version "1.8.0_191"
Java(TM) SE Runtime Environment (build 1.8.0_191-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.191-b12, mixed mode)

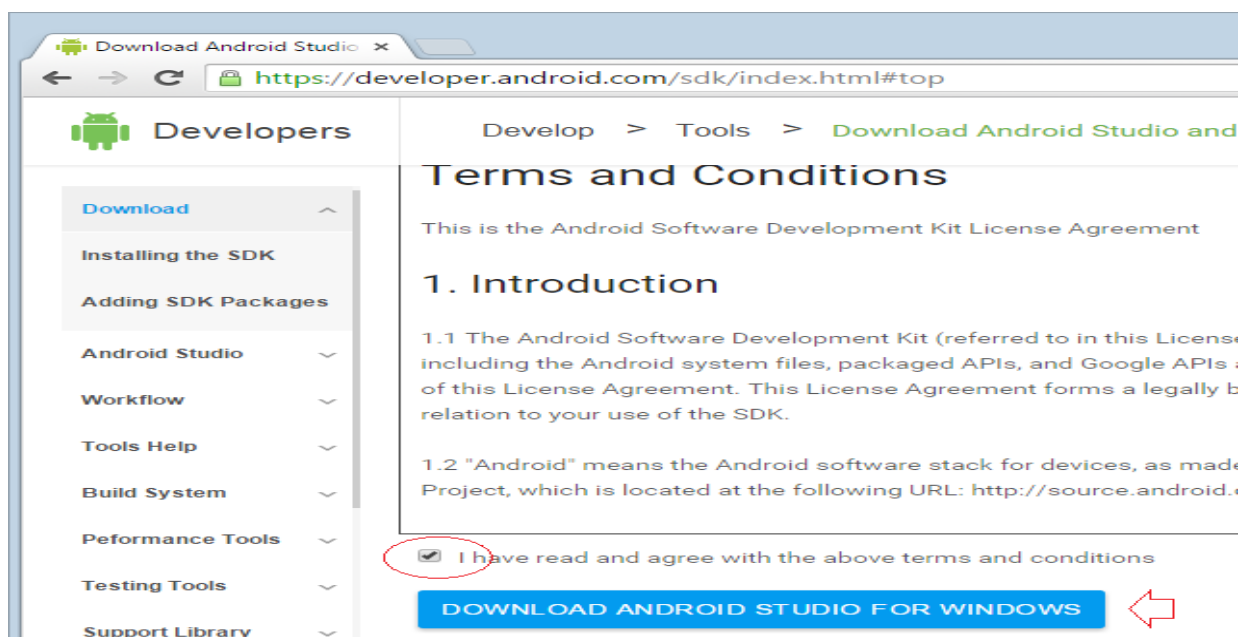
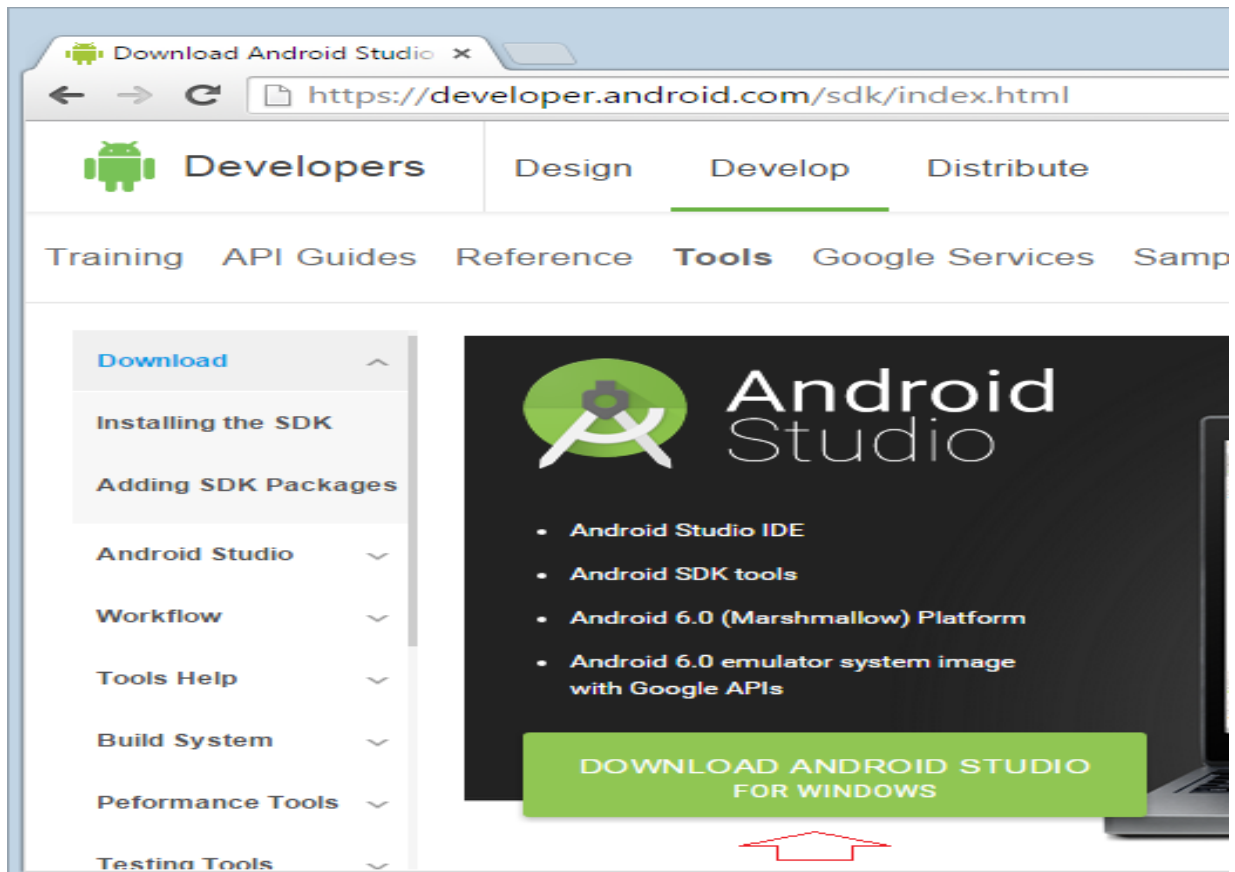
C:\Users\tc>javac -version
javac 1.8.0_191

C:\Users\tc>
```

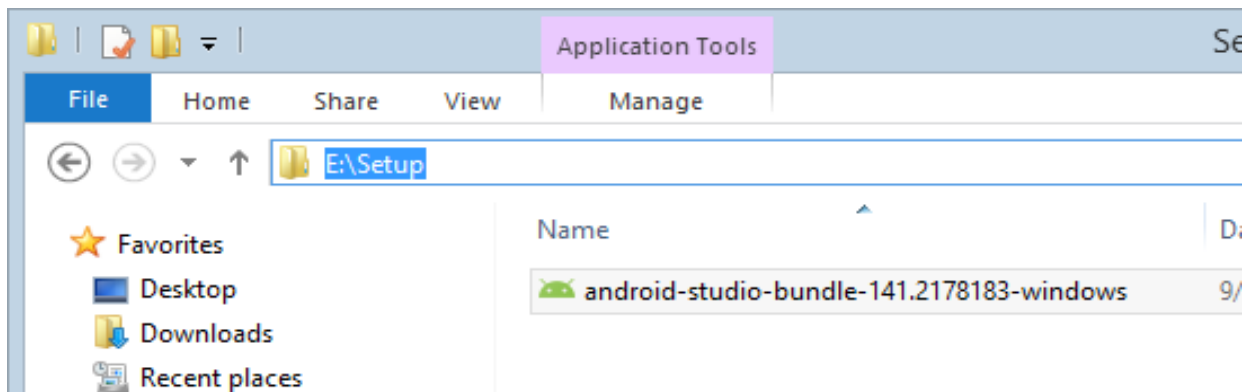
b. Cài đặt Android Studio

Bước 1: Download Android Studio tại link:

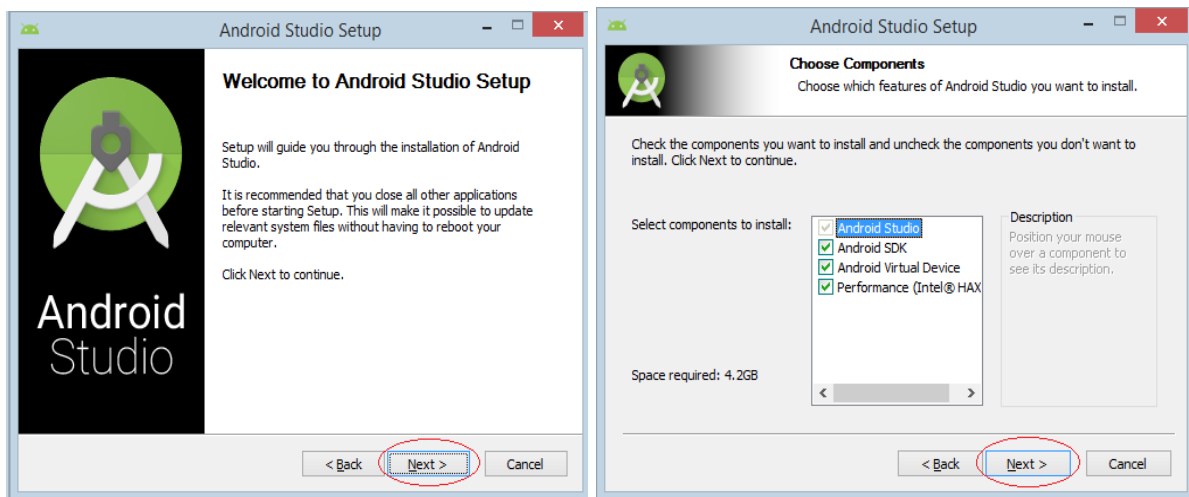
<https://developer.android.com/sdk/index.html>



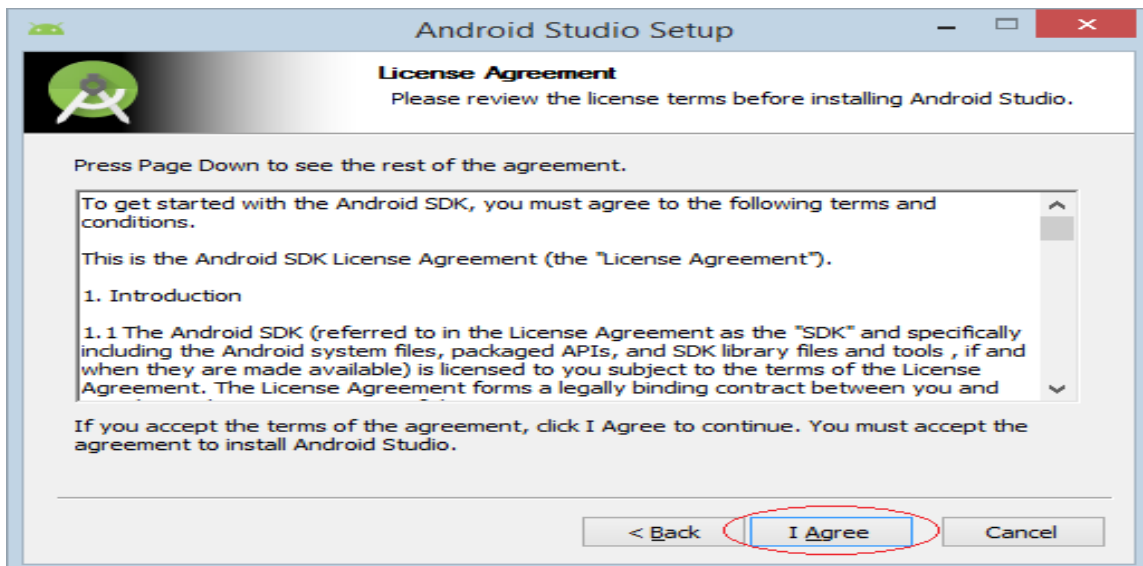
Kết quả download được:



Bước 2: Cài đặt Android Studio



Lựa chọn tất cả các tùy chọn (options).



Chọn thư mục cài đặt:

Android Studio Installation location: C:\DevPrograms\Android\Android Studio

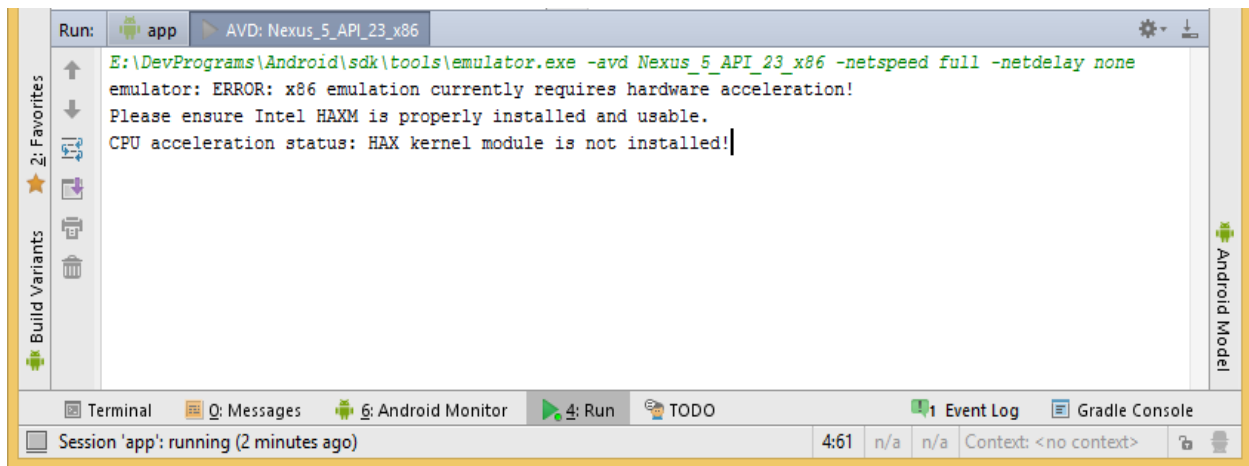
Android SDK Install Location: C:\DevPrograms\Android\sdk

Bước 3: Khởi động Android Studio:

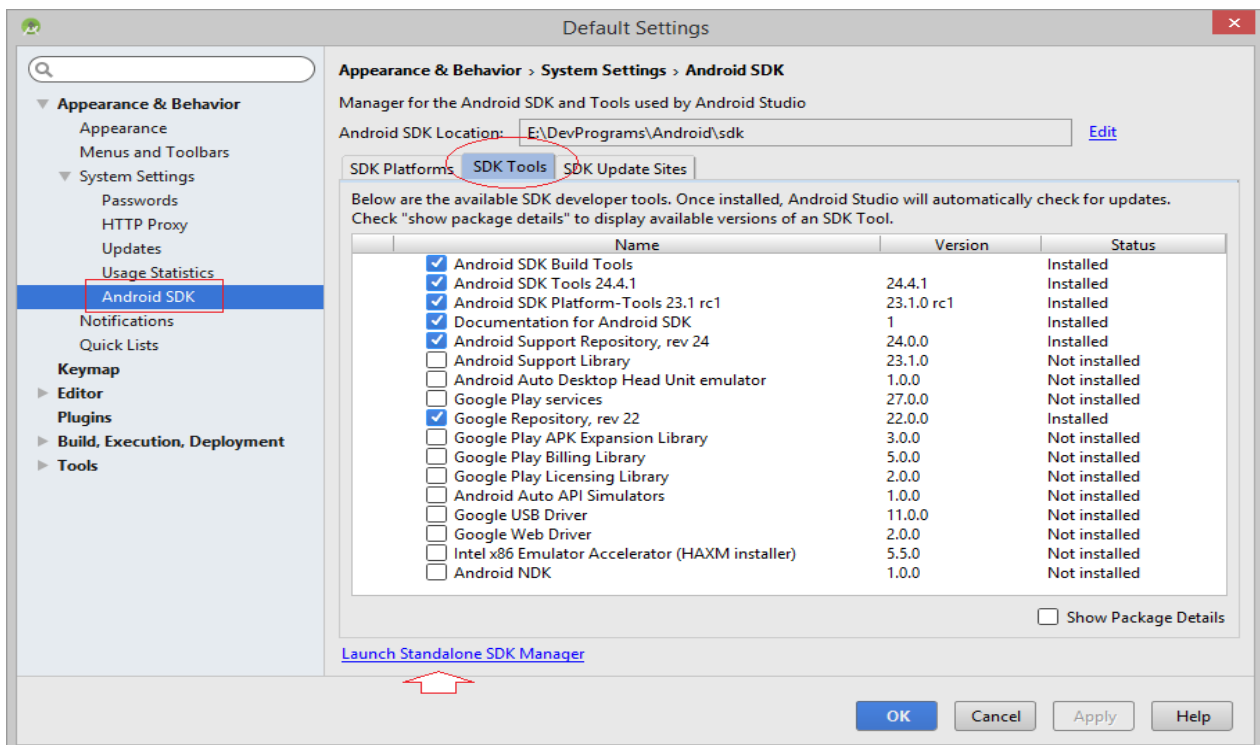
Android Studio hỏi có nhập khẩu các cài đặt từ phiên bản **Android Studio** mà có thể đã cài đặt trước đó hay không, có thể chọn **No** > Lựa chọn một **Theme** > chờ đợi cho khi tiến trình download và cài đặt hoàn thành.

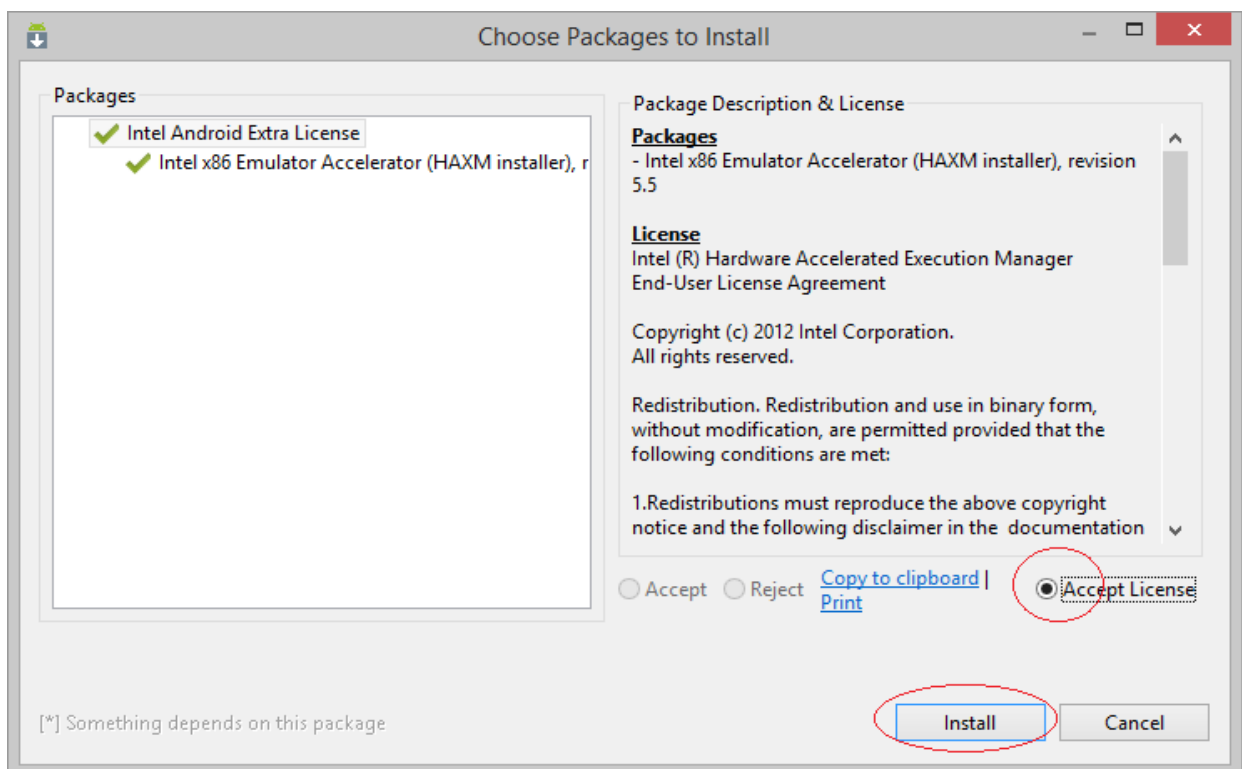
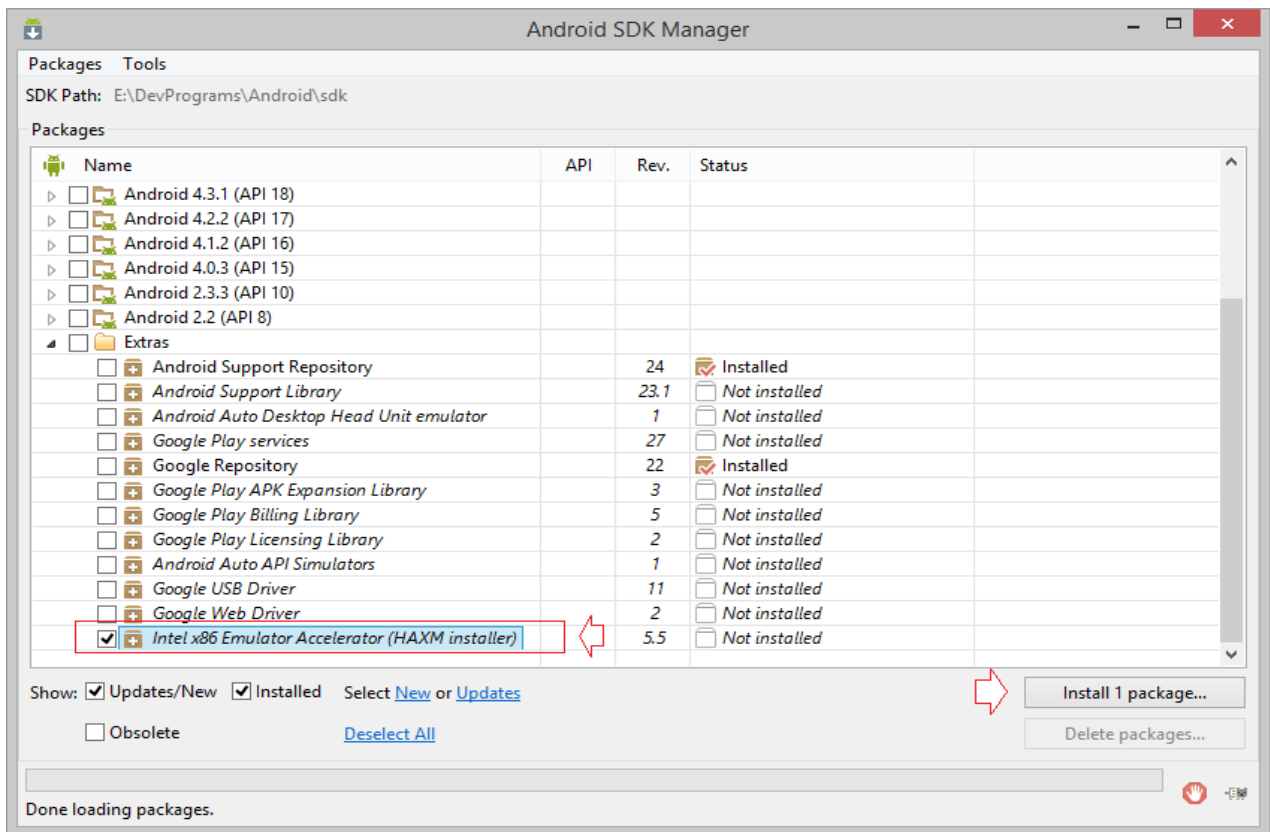
Bước 4: Cài đặt Intel® HAXM (Phần mềm quản lý tăng tốc phần cứng).

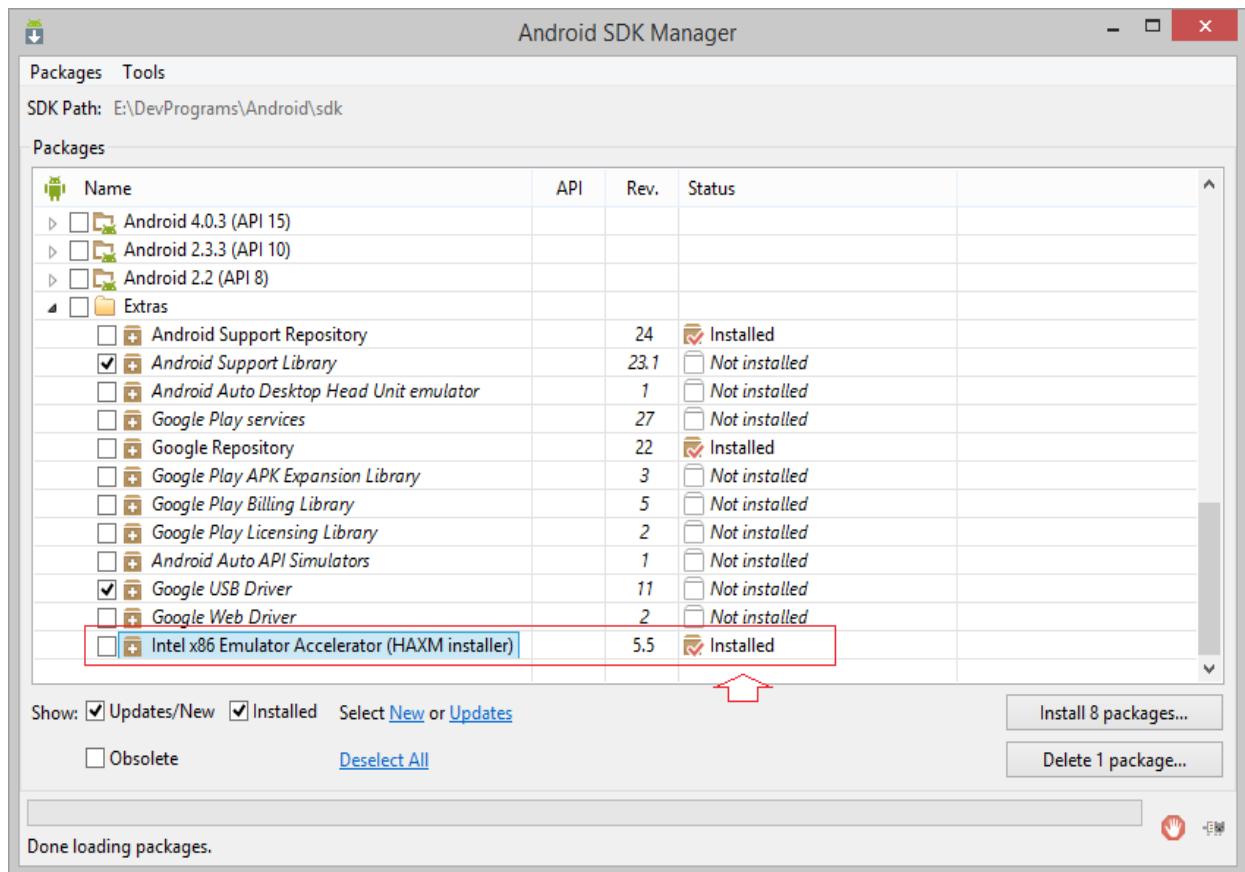
Nếu thấy thông báo dưới đây khi chạy ứng dụng **Android**, nghĩa là chưa cài đặt Intel® HAXM, cần phải cài đặt nó.



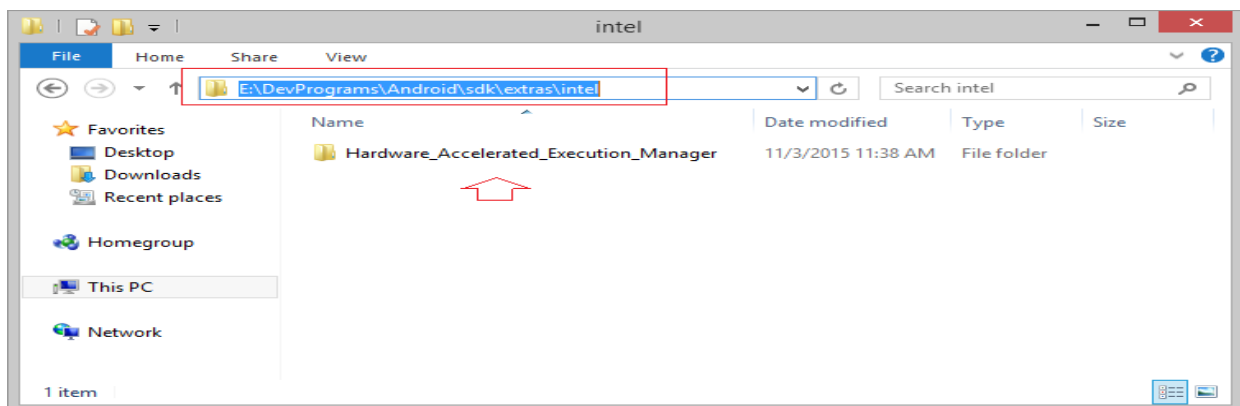
Trên Android Studio chọn: **Tools** > **Android** > **SDK Manager** rồi làm theo hướng dẫn ở dưới để tải về.

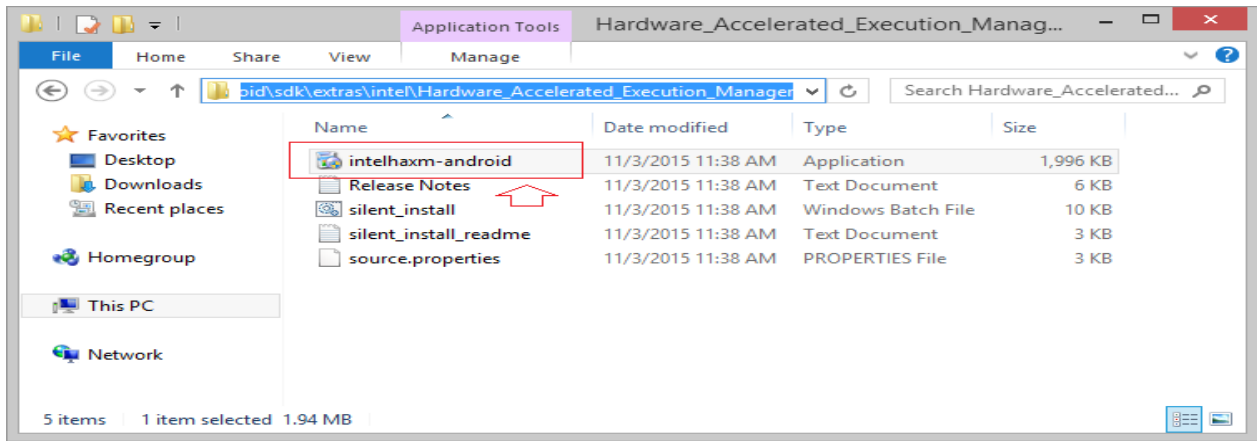






Chạy trình cài đặt từ thư mục **"extras"** để hoàn thành việc cài đặt.

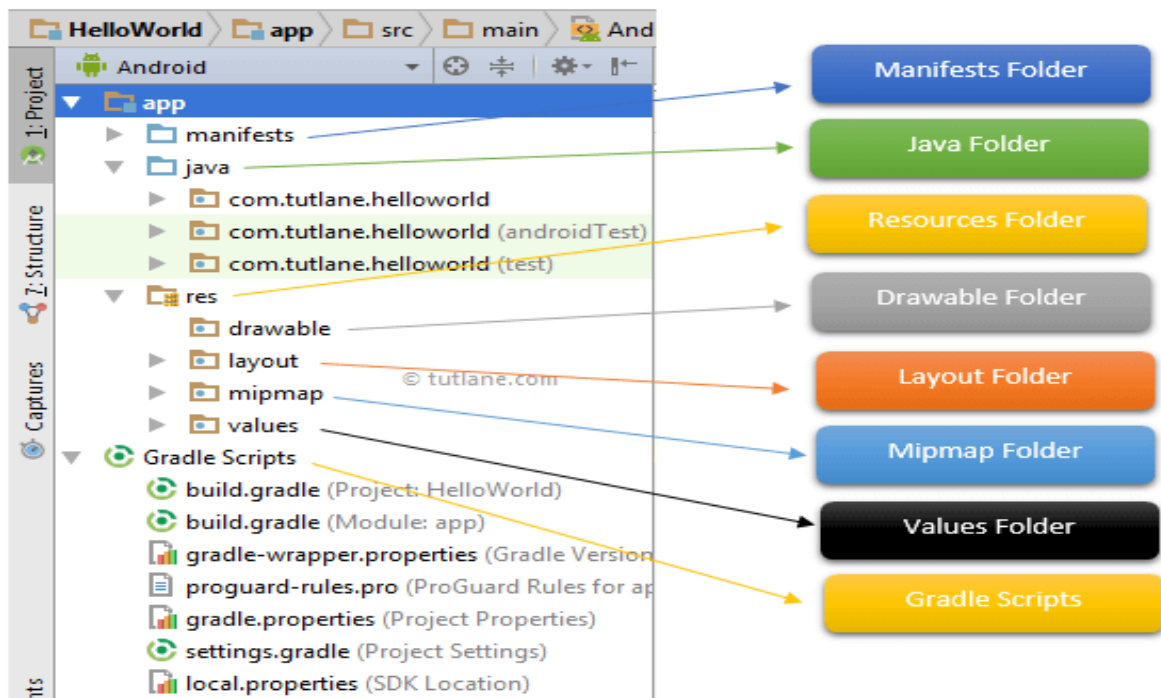




2.2. Tạo ứng dụng android đầu tiên

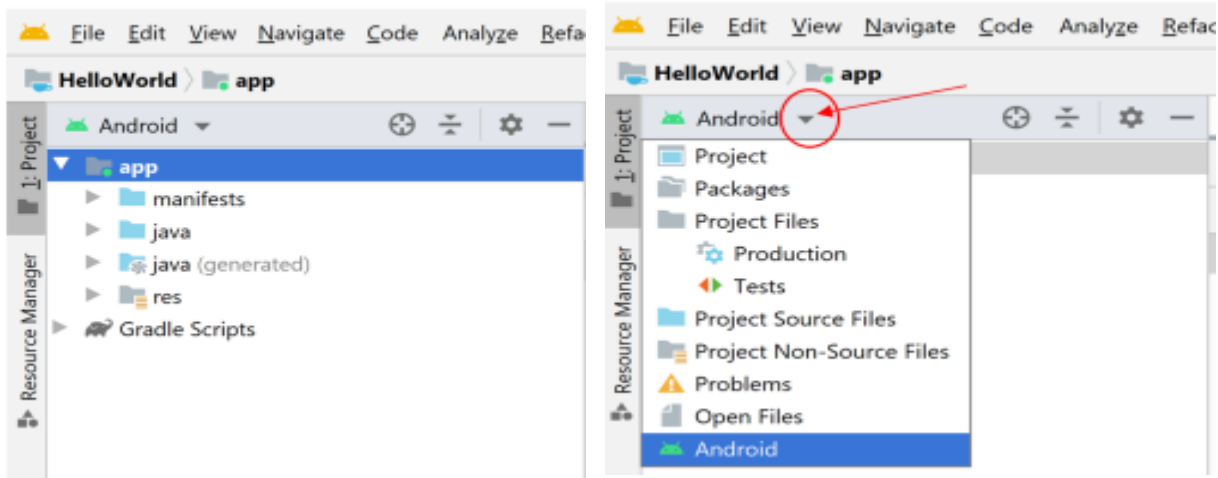
2.1.1 Lý thuyết liên quan

a. Cấu trúc của một dự án Android:



Cấu trúc project có thể khác với minh họa tùy thuộc vào mục chọn. Để xem cấu trúc file thực tế của project, chọn Project từ menu dropdown nằm ở góc trên bên trái (mặc định chọn Android).

Trong một dự án Android có rất nhiều chế độ quan sát cấu trúc, để thay đổi ta nhấn vào mục khoan tròn rồi lựa chọn các kiểu quan sát. Tuy nhiên thường chọn loại hiển thị là “**Android**”.



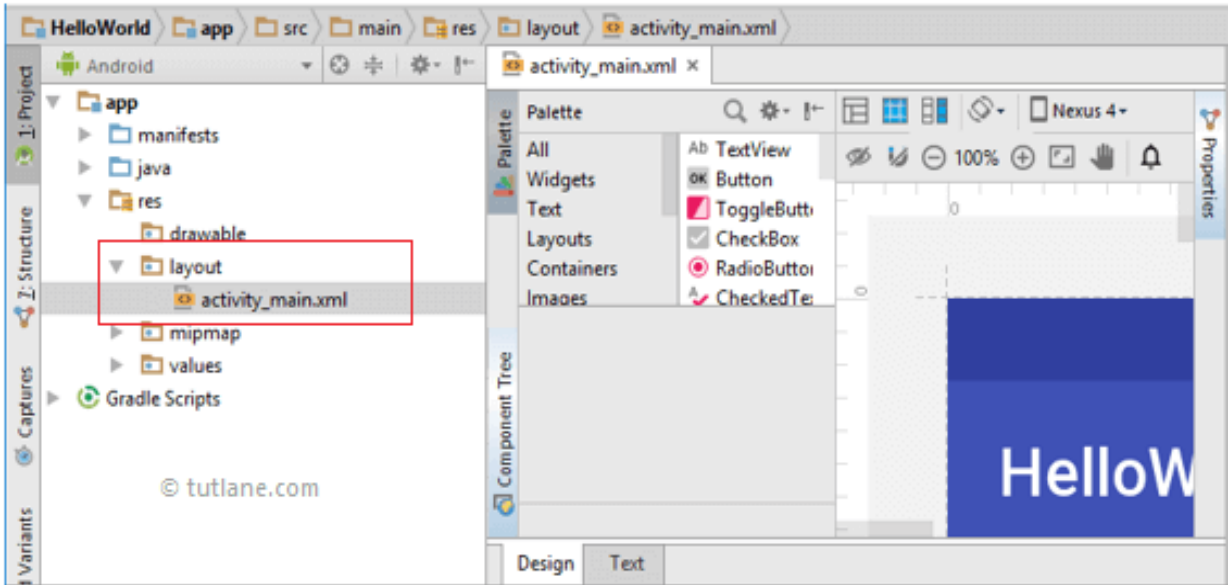
File hoặc thư mục	Mô tả
AndroidManifest.xml	Đây là file manifest mô tả các đặc điểm cơ bản của ứng dụng và xác định từng thành phần của nó.
java	Thư mục này có chứa các file nguồn java cho dự án. Theo mặc định, nó bao gồm một tập tin nguồn MainActivity.java và một lớp hoạt động (activity) chạy khi ứng dụng của được khởi động.
res/drawable	Các phiên bản Android trước đây sử dụng thư mục này để chứa ảnh, các phiên bản hiện tại sử dụng thư mục mipmap thay thế làm nơi chứa ảnh. Thư mục này gần như không còn sử dụng.
res/layout	Thư mục này chứa các file định nghĩa giao diện người dùng.
res/menu	Thư mục này chứa các file xml, định nghĩa các menu sẽ hiển thị trên Action Bar.
res/mipmap	Chứa các ảnh 'mipmap'.

res/values	<p>Đây là một thư mục cho các tập tin XML khác nhau có chứa một tập hợp các nguồn, chẳng hạn như các chuỗi (String) và các định nghĩa màu sắc.</p>
------------	--

b. Các file quan trọng cần để triển khai một ứng dụng trong android studio.

- **File Android Layout (activity_main.xml)**

File activity_main.xml nằm trong thư mục layout (res/layout).

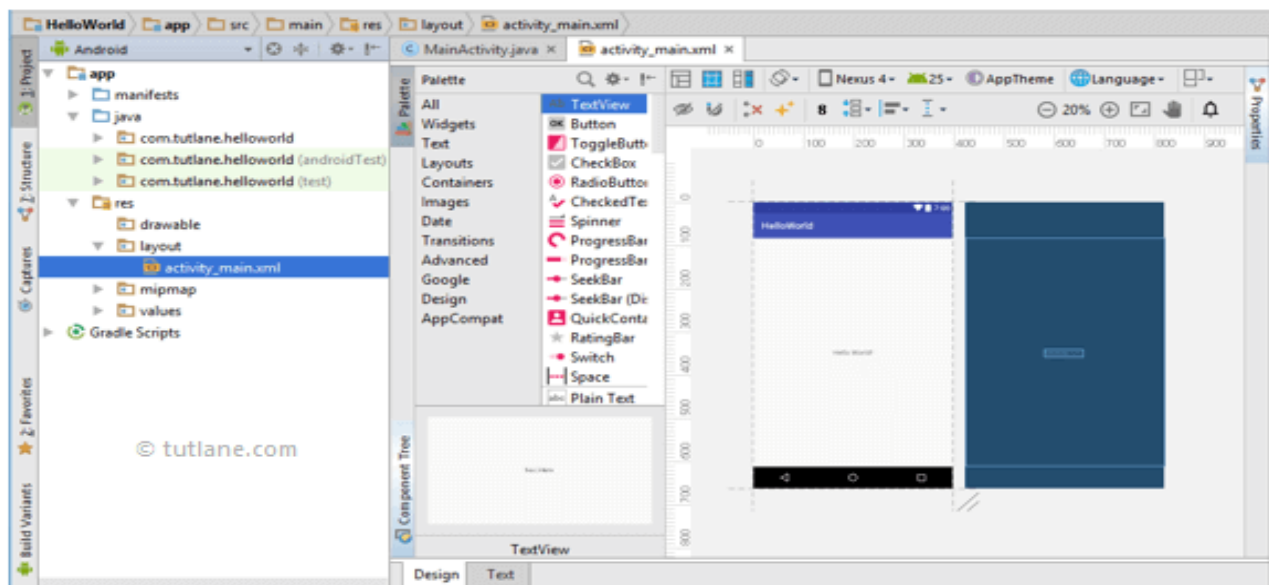


Giao diện trong file activity_main.xml bao gồm:

- Thanh Palette: kéo thả các control vào màn hình thiết kế của điện thoại.
- Thanh Component Tree giúp quan sát được cấu trúc của các control kéo thả trên giao diện.
- Thanh Attributes: thiết lập các trạng thái cho control trên giao diện.

Giao diện người dùng của ứng dụng sẽ được thiết kế trong file này và nó sẽ có ba chế độ: **Code** (thiết kế bằng mã XML), **Split** (vừa nhìn thấy code XML vừa nhìn thấy giao diện chạy theo), **Design** (thiết kế giao diện bằng kéo thả).

Cấu trúc của file **activity_main.xml** trong chế độ **Design** như hình dưới.



Cấu trúc của file `activity_main.xml` trong chế độ Code.

Ví dụ:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout xmlns:android="
http://schemas.android.com/apk/res/android
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context="com.tutlane.helloworld.MainActivity">
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Hello World!"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
</android.support.constraint.ConstraintLayout>
```

- **File Android Main Activity (MainActivity.java)**

File **MainActivity.java** nằm trong thư mục **java**, chứa mã java để xử lý tất cả các activity liên quan đến ứng dụng .

Ví dụ mã mặc định file MainActivity.java của ứng dụng HelloWorld:

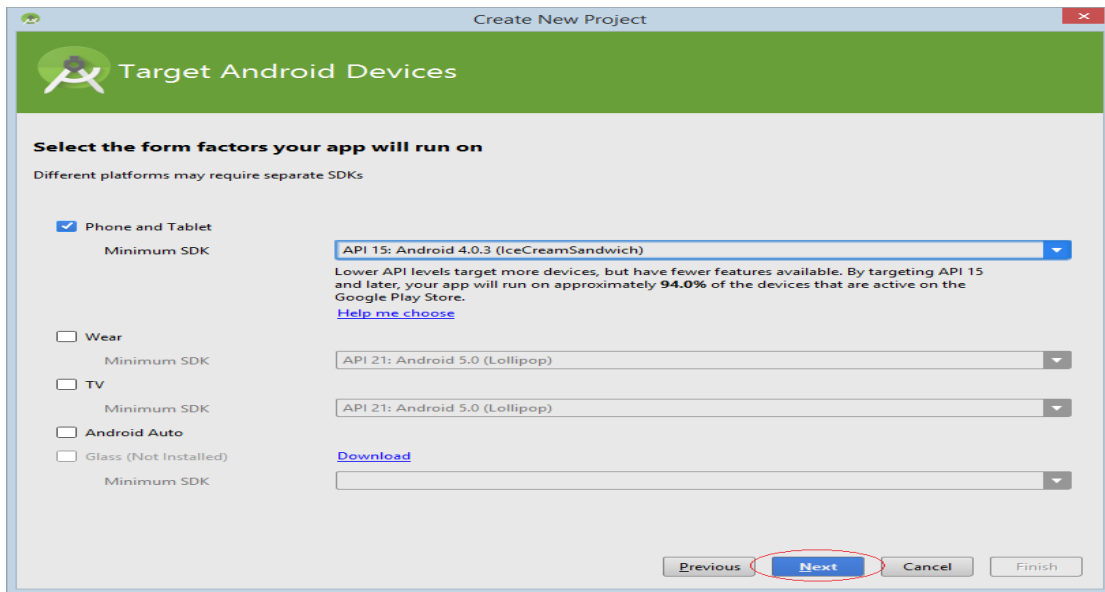
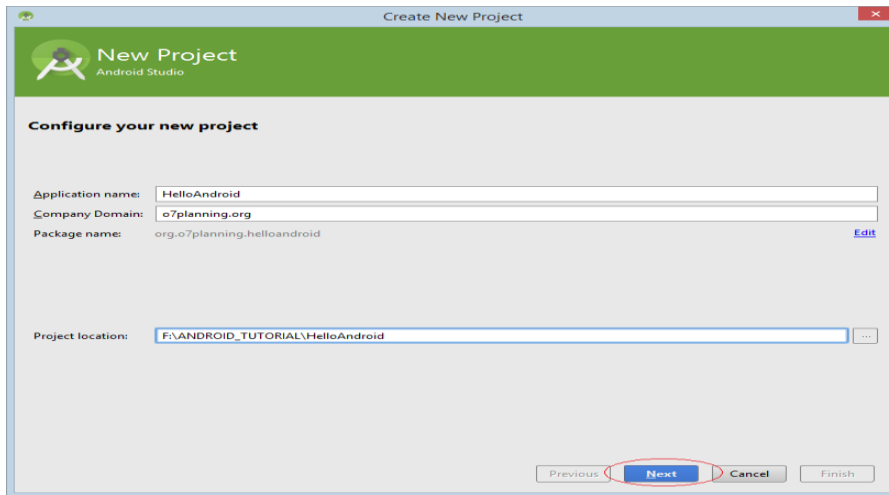
```
package com.tutlane.helloworld;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
public class MainActivity extends AppCompatActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

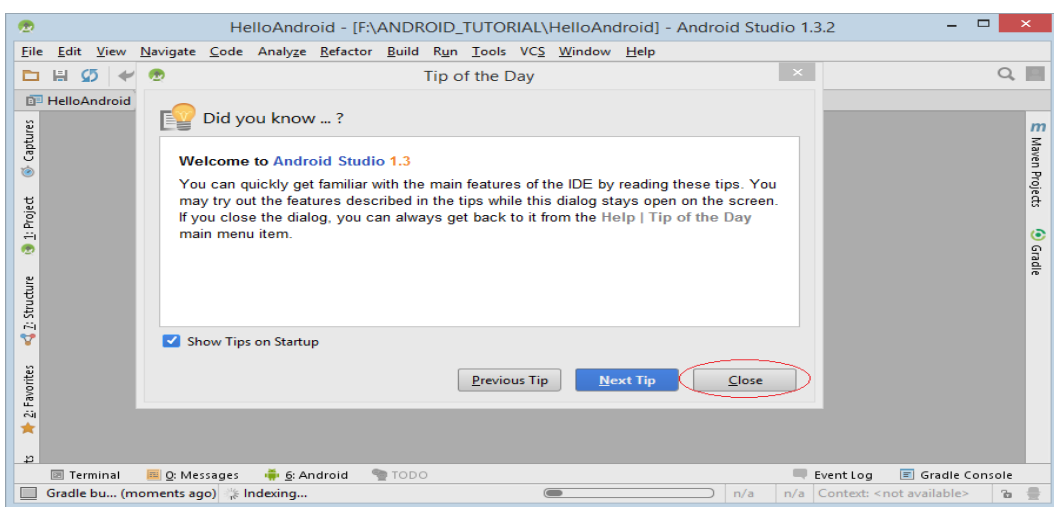
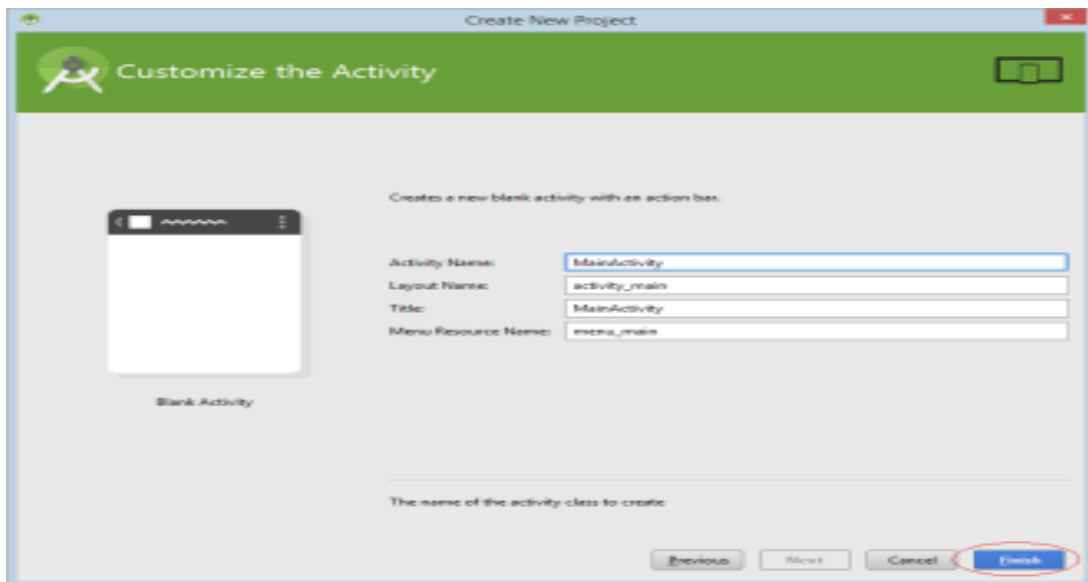
2.1.2 Trình tự thao tác

Bước 1: Khởi động Android Studio > chọn **File** > **New** > **New Project** > điền các thông tin về dự án > **Next**.

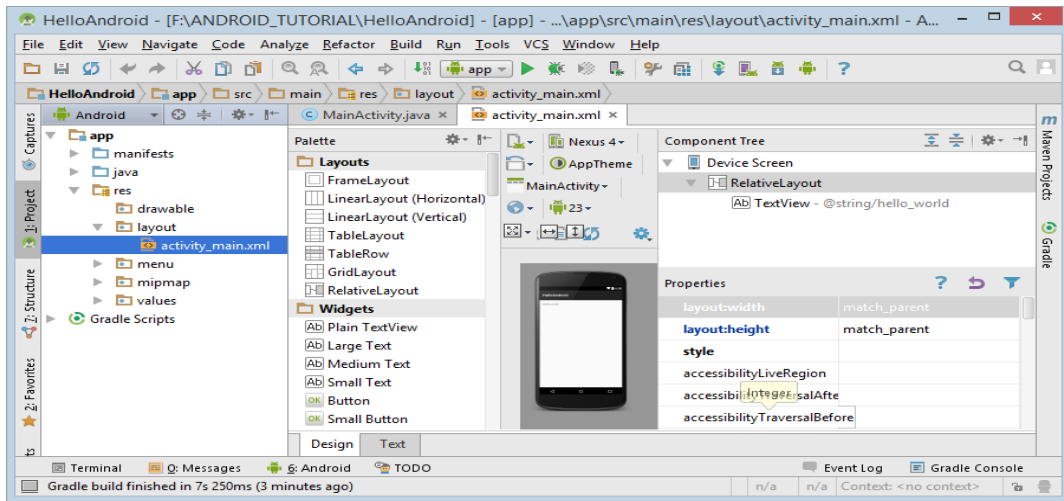
- *Application name:* Khai báo tên của ứng dụng muốn tạo, mặc định khi đặt tên của ứng dụng thì đó cũng sẽ là tên Project.
- *Project location:* Khai báo đường dẫn lưu trữ ứng dụng.
- *Project name:* Khai báo tên package ứng dụng, mỗi ứng dụng khi đưa lên Google Play thì tên package phải là duy nhất.

Chọn mục **Phone and Tablet** > chọn phiên bản bản SDK thấp nhất > **Next**.

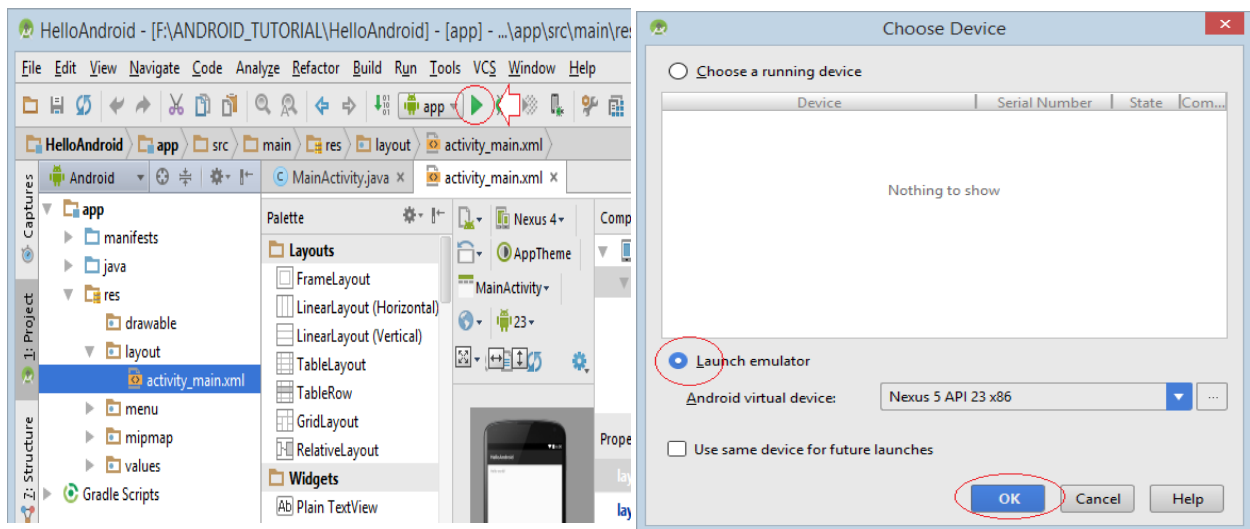




Project của đã được tạo ra.



Bước 2: Nhấn nút Run để chạy thử ứng dụng



Màn hình mô phỏng đã hiển thị một chiếc điện thoại, và ghi ra dòng chữ **"Hello World"**.

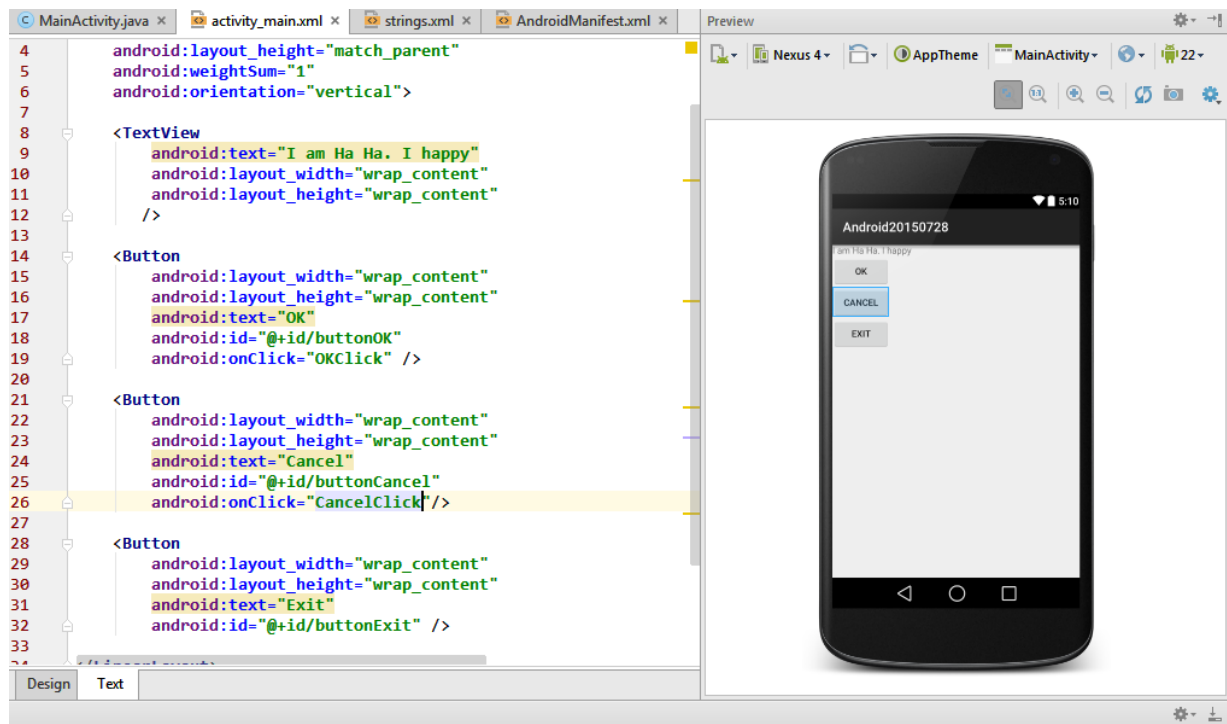


BÀI TẬP

Bài 1: Làm theo hướng dẫn: Tạo giao diện như hình, click và nút xuất hiện thông báo.



Gợi ý: Mở file activity_main.xml Tạo giao diện bằng Code or Design



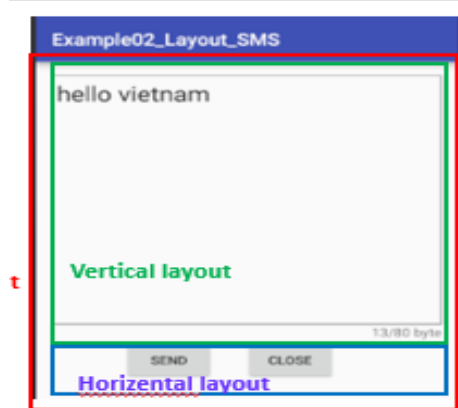
- Coding cho 3 nút

```

13
14
15 @Override
16 protected void onCreate(Bundle savedInstanceState) {
17     super.onCreate(savedInstanceState);
18     setContentView(R.layout.activity_main);
19
20     Button buttonOK1 = (Button) findViewById(R.id.buttonExit);
21     buttonOK1.setOnClickListener(new View.OnClickListener() {
22         @Override
23         public void onClick(View v) {
24             Toast.makeText(getApplicationContext(), "exit click ", Toast.LENGTH_LONG).show();
25         }
26     });
27
28     public void OKClick(View v){
29         Toast.makeText(getApplicationContext(), "ok click ", Toast.LENGTH_LONG).show();
30     }
31     public void CancelClick(View v){
32         Toast.makeText(getApplicationContext(), "cancel click ", Toast.LENGTH_LONG).show();
33     }
34
35 @Override
36 public boolean onCreateOptionsMenu(Menu menu) {
37     // Inflate the menu; this adds items to the action bar if it is present.
38     getMenuInflater().inflate(R.menu.menu_main, menu);
39     return true;
40 }
41
42 @Override
43 public boolean onOptionsItemSelected(MenuItem item) {
44     // Handle action bar item clicks here. The action bar will
45     // automatically handle clicks on the Home/Up button, so Long

```

Bài 2: Làm theo hướng dẫn: Tạo giao diện như hình, click và nút xuất hiện thông báo.



File activity_main.xml:

```

<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context="com.example.admin.myapplication.MainActivity">
<LinearLayout

```

```

android:layout_width="368dp"
android:layout_height="495dp"
android:orientation="vertical"
tools:layout_editor_absoluteY="0dp"
tools:layout_editor_absoluteX="0dp">
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="300dp"
    android:orientation="vertical">
    <EditText
        android:id="@+id/edit_send"
        android:layout_width="match_parent"
        android:layout_height="300dp"
        android:gravity="start"/>
    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="100dp"
        android:layout_weight="1"
        android:layout_gravity="right"
        android:text="TextViewfhgfh" />
</LinearLayout>
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:orientation="horizontal"
    tools:layout_editor_absoluteY="0dp"
    tools:layout_editor_absoluteX="8dp">
    <Button
        android:layout_width="wrap_content"

```

```

        android:layout_height="wrap_content"
        android:text="Send"
        android:layout_marginLeft="70dp"
        android:layout_marginRight="30dp"
        android:id="@+id/send"/>
    <Button
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Close"
        android:id="@+id/close"/>
</LinearLayout>
</LinearLayout>
</android.support.constraint.ConstraintLayout>

```

- File ActivityMain.java

```

public class MainActivity extends AppCompatActivity {
    EditText editSend;
    TextView textView;
    Button send;
    Button close;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        editSend = (EditText)findViewById(R.id.edit_send);
        textView = (TextView)findViewById(R.id.textView2);
        send = (Button)findViewById(R.id.send);
        close = (Button)findViewById(R.id.close);
        //Lấy độ dài chuỗi gõ trên TextEdit
        TextWatcher textWatcher = new TextWatcher() {
            @Override

```

```

public void beforeTextChanged(CharSequence s, int start, int count, int after) {
}
@Override
public void onTextChanged(CharSequence s, int start, int before, int count) {
}
@Override
public void afterTextChanged(Editable s) {
    String a = editSend.toString();
    textView.setText(editSend.length()+"/80 byte");
}
};
editSend.addTextChangedListener(textWatcher);
//Nút SEND
send.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        String inputText = editSend.getText().toString();
        Toast.makeText(getApplicationContext(),inputText, Toast.LENGTH_SHORT).
            show();    }
});
//Nút close
close.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        finish();
    }
});
}
}

```

3. TÓM TẮT TRÌNH TỰ THỰC HIỆN HOẶC QUY TRÌNH CÔNG NGHỆ

TT	Tên các bước công việc	Dụng cụ, thiết bị, vật tư	Yêu cầu kỹ thuật	Các chú ý về an toàn lao động
1	Cài đặt JDK	Phần mềm Java Máy tính.	Cài đặt chính xác phần mềm.	Đảm bảo an toàn cho người và thiết bị.
2	Cài đặt Android Studio	Phần mềm Android Studio Máy tính.	Thực hiện đúng các bước.	Đảm bảo an toàn cho người và thiết bị.
3	Khởi động Android Studio	Phần mềm Android Studio	Thực hiện đúng các bước.	Đảm bảo an toàn cho người và thiết bị.
4	Thực hành thiết kế và code cho các nút lệnh	Máy tính.	Thực hiện đúng các bước.	Đảm bảo an toàn cho người và thiết bị.
5	Kiểm tra kết quả	Phần mềm Android Studio	Thực hiện đúng các bước.	Đảm bảo an toàn cho người và thiết bị.

BÀI 2: SỬ DỤNG NGÔN NGỮ JAVA

1. MỤC TIÊU

Trình bày được kiến thức cơ bản về ngôn ngữ Java, sử dụng được các câu lệnh Java trong lập trình. Trình bày được cú pháp các câu lệnh cơ bản trong java. Trình bày được kiến thức về đối tượng, lớp, kế thừa và đa hình.

Nghiêm túc, tự giác trong học tập. Đảm bảo an toàn cho người và thiết bị.

2. NỘI DUNG

2.1. Lý thuyết liên quan

Java là ngôn ngữ lập trình hướng đối tượng (tựa C++) do Sun Microsystem đưa ra vào giữa thập niên 90. Chương trình viết bằng ngôn ngữ lập trình java có thể chạy trên bất kỳ hệ thống nào có cài máy ảo java (Java Virtual machine).

Các kiểu chương trình Java

- Java và ứng dụng Console: là ứng dụng nhập xuất ở chế độ văn bản.
- Java và ứng dụng Applet: Java Applet là loại ứng dụng có thể nhúng và chạy trong trang web của một trình duyệt web.
- Java và phát triển ứng dụng Desktop dùng AWT và JFC.
- Java và phát triển ứng dụng Web thông qua công nghệ J2EE (Java 2 Enterprise Edition).
- Java và phát triển các ứng dụng nhúng cung cấp một môi trường có thể chạy được trên các thiết bị như: điện thoại di động, máy tính bỏ túi PDA hay Palm, cũng như các thiết bị nhúng khác.

Máy ảo Java

Một chương trình chạy được trên Windows muốn chạy được trên hệ điều hành khác như Linux chẳng hạn thì phải chỉnh sửa và biên dịch lại. Ngôn ngữ lập trình Java ra đời, nhờ vào máy ảo Java mà khó khăn nêu trên đã được khắc phục. Một chương trình viết bằng ngôn ngữ lập trình Java sẽ được biên dịch ra mã của máy ảo java (mã java bytecode). Sau đó máy ảo Java chịu trách nhiệm chuyển mã java bytecode thành mã máy tương ứng.

Java Development Kit (Bộ công cụ phát triển Java - viết tắt là JDK) cho phép lập trình viên viết code và chạy các chương trình lập trình Java.

a. Khai báo biến và các kiểu dữ liệu.

- Cú pháp khai báo biến: **Kiểu dữ liệu** **Tên biến**;

```
public static float PI = 3.14f; // Đây là biến static
```

```
int n; // Đây là biến instance kiểu nguyên
```

```
String c = "Hello"; // Đây là biến instance kiểu chuỗi
```

```
String[] mảng; // Đây là kiểu mảng
```

- Có nhiều kiểu dữ liệu trong java.

Ví dụ: byte, byte, boolean, int, float, Array, class,...

b. Các câu lệnh

- Câu lệnh if else

```
if (<điều kiện>) {
```

```
    // khối lệnh này được thực thi
```

```
    // nếu điều kiện đúng
```

```
} else {
```

```
    // khối lệnh này được thực thi
```

```
    // nếu điều kiện sai
```

```
}
```

- Câu lệnh lặp for

```
for (<biến khởi tạo>; <biểu thức điều kiện>; <tăng/giảm biến>) {
```

```
    // các lệnh;
```

```
}
```

- Câu lệnh lặp while

```
while (<điều kiện>){
```

```
    // các lệnh;
```

```
}
```

- Câu lệnh switch case

```
switch ( <Biến kiểm tra> ) {
```

```
    case value1:
```

```
        // các lệnh;
```

```
        break;
```

```

case value2:
    // các lệnh;
    break;
default:
    // các lệnh;
}

```

c. Lớp, đối tượng.

- Đối tượng ở đây ta thể hiểu như khái niệm bên ngoài: Con người, Xe máy, Nhà cửa...

Trong một đối tượng sẽ bao gồm 2 thông tin: thuộc tính và phương thức.

- Lớp chính là định nghĩa của đối tượng, ta sẽ xây dựng lớp để tạo ra những đối tượng khác nhau.

d. Kế thừa, đa hình

- Kế thừa trong java là sự liên quan giữa hai class với nhau, trong đó có class cha (superclass) và class con (subclass).

- Đa hình trong java (Polymorphism) là một khái niệm mà chúng ta có thể thực hiện một hành động bằng nhiều cách khác nhau như là nạp chồng phương thức và ghi đè phương thức.

2.2. Trình tự thao tác

Bước 1: Tạo ra một lớp Student có hai thành viên dữ liệu *id* và *name* theo mẫu dưới. Tạo ra các đối tượng của lớp *Student* bởi từ khóa *new* và in giá trị của các đối tượng.

```

public class Student {
    int id; // thành viên dữ liệu
    String name; // thành viên dữ liệu
    public static void main(String args[]) {
        Student student1 = new Student(); // tạo một đối tượng student1
        System.out.println(student1.id);
        System.out.println(student1.name);
    }
}

```

Bước 2: Tạo lớp Programmer Kế thừa lớp Employee theo mẫu dưới. In các dữ liệu lên màn hình.

```
class Employee {
    float salary = 1000;
}
class Programmer extends Employee {
    int bonus = 150;
}
public class InheritanceSample1 {
    public static void main(String args[]) {
        Programmer p = new Programmer();
        System.out.println("Programmer salary is: " + p.salary);
        System.out.println("Bonus of Programmer is: " + p.bonus);
    }
}
```

Bước 3: Đa hình: Tạo hai lớp Bike và Splendar theo mẫu dưới. Lớp Splendar kế thừa lớp Bike và ghi đè phương thức run() của nó. Chúng ta gọi phương thức run bởi biến tham chiếu của lớp cha. Khi nó tham chiếu tới đối tượng của lớp con và phương thức lớp con ghi đè phương thức của lớp cha, phương thức lớp con được gọi lúc runtime.

```
class Bike {
    void run() {
        System.out.println("running");
    }
}
public class Splender extends Bike {
    void run() {
        System.out.println("running safely with 60km");
    }
    public static void main(String args[]) {
        Bike b = new Splender();// upcasting
    }
}
```

```

        b.run();
    }
}

```

BÀI TẬP:

1. Hãy tải và cài đặt Java bản mới nhất. Cấu hình và kiểm tra kết quả cài đặt
2. Trình bày một số câu lệnh cơ bản trong Java
3. Tạo lớp và lớp kế thừa.

3. TÓM TẮT TRÌNH TỰ THỰC HIỆN HOẶC QUY TRÌNH CÔNG NGHỆ

<i>TT</i>	<i>Tên các bước công việc</i>	<i>Dụng cụ, thiết bị, vật tư</i>	<i>Yêu cầu kỹ thuật</i>	<i>Các chú ý về an toàn lao động</i>
1	Cấu hình phần mềm	Phần mềm Java Máy tính.	Thực hiện đúng các bước.	Đảm bảo an toàn cho người và thiết bị.
2	Kiểm tra kết quả.	Phần mềm Java Máy tính.	Thực hiện đúng các bước.	Đảm bảo an toàn cho người và thiết bị.
3	Thực hiện các ví dụ tạo lớp, kế thừa và đa hình	Phần mềm Java Máy tính.	Thực hiện đúng các bước.	Đảm bảo an toàn cho người và thiết bị.

BÀI 3: SỬ DỤNG LAYOUT

1. MỤC TIÊU

Trình bày được các khái niệm Layout. Tạo được các Layout thường dùng. Sử dụng thành thạo Studio Android để xây dựng các layout.

Nghiêm túc, tự giác trong học tập. Đảm bảo an toàn cho người và thiết bị.

2. NỘI DUNG

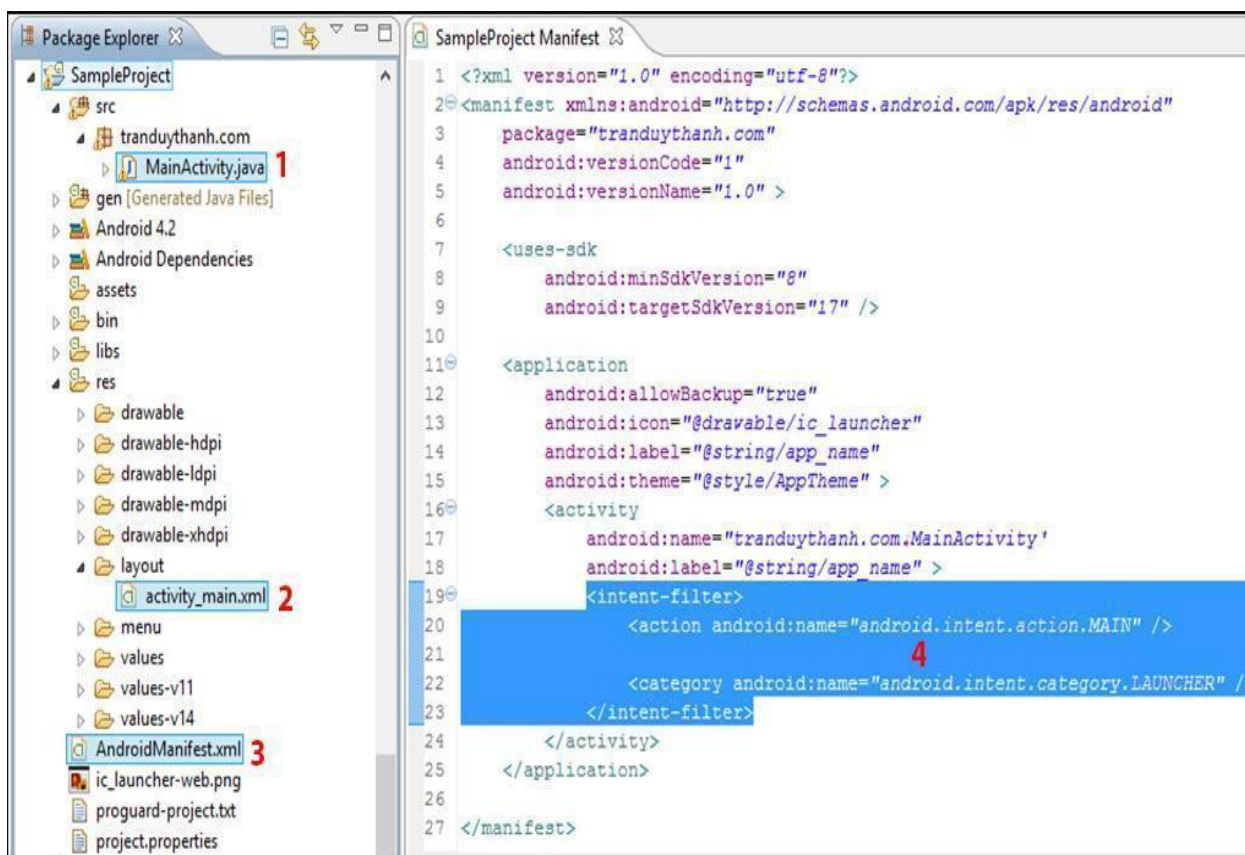
2.1. Lý thuyết liên quan

2.1.1. Các khái niệm về Layout

Layout là thành phần định nghĩa cấu trúc giao diện người dùng hay nói cách khác là thành phần quyết định đến giao diện của một màn hình trong ứng dụng Android. Một layout hỗ trợ việc căn chỉnh các widget (Ví dụ: TextView, Button, hay EditText...) như chúng ta thấy trong các ứng dụng Android.

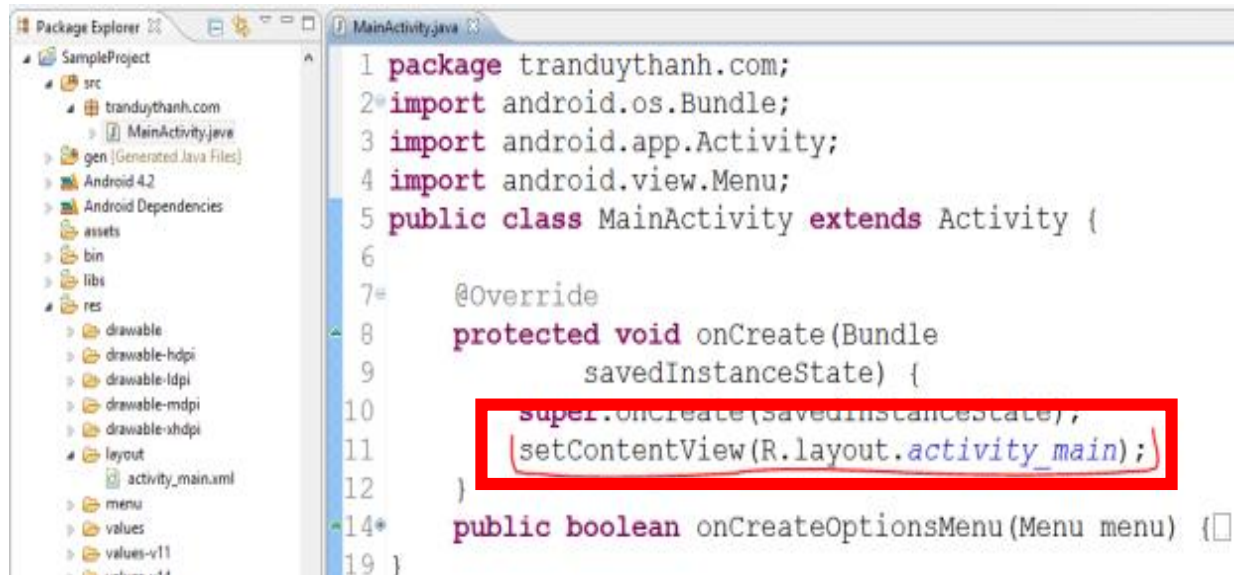
Toàn bộ cấu trúc giao diện mà chúng ta thấy trong ứng dụng Android đều được thiết kế trong một layout. Mỗi layout được định nghĩa là một file xml và được đặt trong App > res > Layout trong Android Studio.

Layout mặc định đầu tiên khi tạo một Project là activity_main.xml



Ứng với MainActivity.java (số 1) sẽ có layout đi kèm là activity_main.xml (số 2). Mọi activity muốn được triệu gọi thành công thì phải được khai báo trong AndroidManifest.xml (số 3). dòng lệnh xanh được khai báo trong vùng số 4.

- Trong tập tin **MainActivity.java**:



```
1 package tranduythanh.com;
2 import android.os.Bundle;
3 import android.app.Activity;
4 import android.view.Menu;
5 public class MainActivity extends Activity {
6
7     @Override
8     protected void onCreate(Bundle
9         savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12     }
13
14     public boolean onCreateOptionsMenu(Menu menu) {
15
16     }
17
18
19 }
```

dòng lệnh: **setContentView (R.layout.activity_main)**; chính là dòng lệnh dùng để kết nối Layout vào Activity.

2.1.2. Một số thuộc tính cơ bản của các Layout thường dùng

Layout_width, layout_height: chiều rộng của view

Orientation: hướng các view con ngang hoặc dọc (horizontal/vertical)

Gravity: các view con đặt theo vị trí nào so với layout (trung tâm, trái, phải, trên dưới...)

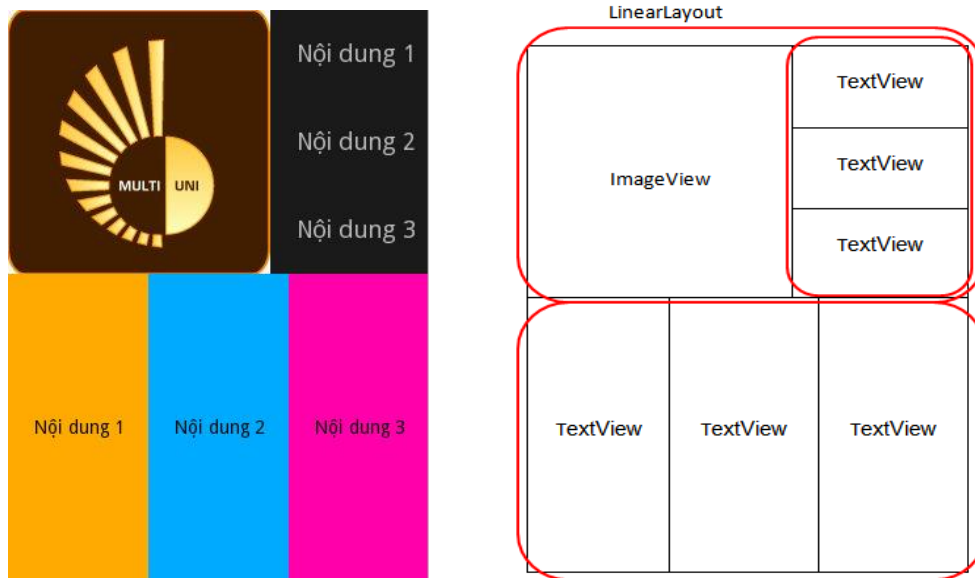
Weight: để các view phân chia tỉ lệ diện tích hiển thị trên màn hình (tỉ lệ tính theo width của từng view trên tổng số weight, các view không khai báo weight thì sẽ xem qua width và height). Gán một giá trị (1, 2, 3, ...) để quy định tỷ lệ không gian dành cho widget đó.

Lưu ý khi thiết kế giao diện: Hạn chế độ sâu của cây. Với các Layout phức tạp, đừng dùng RelativeLayout mà nên dùng ConstraintLayout. Nên chèn vào dữ liệu tạm để xem trước layout hiển thị ra sao.

2.1.3. *LinearLayout*

Layout này cho phép sắp xếp các control theo 2 hướng trên giao diện: Hướng từ trái qua phải và hướng từ trên xuống dưới.

Ví dụ: Màn hình như hình dưới gồm một *LinearLayout* dọc lớn bao bên ngoài, có hai hàng. Hàng trên là một *LinearLayout* ngang, bên trái là *ImageView* (vì chỉ có một view nên không cần bỏ vào trong *LinearLayout*), bên phải là *LinearLayout* dọc chứa 3 *TextView*. Hàng dưới là một *LinearLayout* ngang.



2.1.4. *RelativeLayout*

RelativeLayout: sắp xếp các view con ở bất kì một vị trí nào. Một số thuộc tính:

- *android:layout_alignParentTop*: Nếu “true”, cạnh trên của View sẽ trùng với cạnh trên của thành phần cha.
- *android:layout_centerVertical*: Nếu “true”, View này sẽ được xếp vào giữa theo chiều dọc của thành phần cha.
- *android:layout_below*: View sẽ nằm dưới một view khác, và xác định view đó qua id.
- *android:layout_toRightOf*: View sẽ nằm bên phải so với một View khác.

2.1.5. *TableLayout*

Các View con bên trong sẽ sắp xếp thành dạng bảng. Mỗi hàng là một đối tượng view **TableRow** bên trong **TableLayout** chứa các View con, mỗi View con này nằm ở vị trí một ô bảng (cell). Cột / hàng trong bảng bắt đầu từ số 0.

2.1.6. *FrameLayout*

Các view bên trong được qui định vị trí bằng khoảng cách so với biên trái và trên so với layout, các view có thể đè lên nhau. *FrameLayout* là Layout đơn giản nhất, không nên sử dụng khi có nhiều View con.

2.1.7. *ConstraintLayout*

ConstraintLayout là một layout mạnh, khuyến khích dùng nếu có thể vì nó giúp tạo ra các giao diện phức tạp, mềm dẻo (hạn chế tối đa sử dụng các layout lồng nhau). Nó giúp định vị, sắp xếp các View con dựa trên sự ràng buộc liên hệ của các View con với View cha và sự liên hệ ràng buộc giữa các View con với nhau, với cơ chế tạo xích các View, gán trọng số hay sử dụng trợ giúp giao diện với *Guideline*.

Ràng buộc	Ý nghĩa ràng buộc
<code>layout_constraintLeft_toLeftOf</code>	Ràng buộc cạnh trái của phần tử tới phần tử chỉ ra trong giá trị (gán ID)
<code>layout_constraintLeft_toRightOf</code>	Bên trái với bên phải của phần tử chỉ ra
<code>layout_constraintRight_toLeftOf</code>	Bên phải với bên trái
<code>layout_constraintRight_toRightOf</code>	Phải với phải
<code>layout_constraintTop_toTopOf</code>	Cạnh trên với cạnh trên
<code>layout_constraintTop_toBottomOf</code>	Cạnh trên nối với cạnh dưới
<code>layout_constraintBottom_toTopOf</code>	Dưới với trên
<code>layout_constraintBottom_toBottomOf</code>	Dưới với dưới
<code>layout_constraintBaseline_toBaselineOf</code>	Trùng Baseline
<code>layout_constraintStart_toEndOf</code>	Bắt đầu - Kết thúc
<code>layout_constraintStart_toStartOf</code>	Bắt đầu - Bắt đầu
<code>layout_constraintEnd_toStartOf</code>	Cuối với bắt đầu
<code>layout_constraintEnd_toEndOf</code>	Cuối với cuối

- Thuộc tính Margin trong các phần tử con: Cạnh nào của View con có ràng buộc thì có thể thiết lập thêm thuộc tính Margin để điều chỉnh thêm khoảng cách các cạnh tới điểm nối ràng buộc.

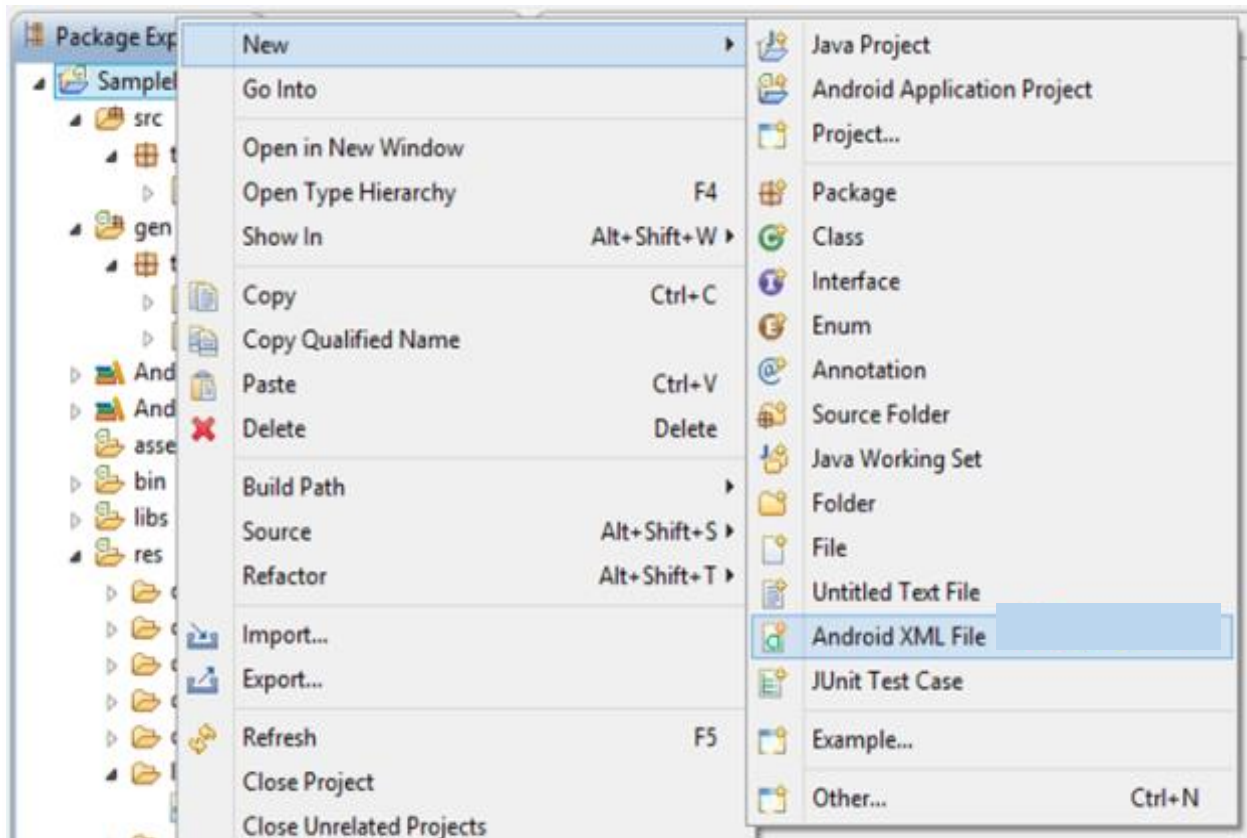
Các thuộc tính margin theo các cạnh: *android:layout_marginStart*,
android:layout_marginEnd, *android:layout_marginLeft*, *android:layout_marginTop*,
android:layout_marginRight, *android:layout_marginBottom*

Tương tự có *layout_goneMarginStart*, *layout_goneMarginEnd*, *layout_goneMarginLeft*,
layout_goneMarginTop, *layout_goneMarginRight*, *layout_goneMarginBottom* có hiệu lực khi đối tượng ràng buộc đến gone

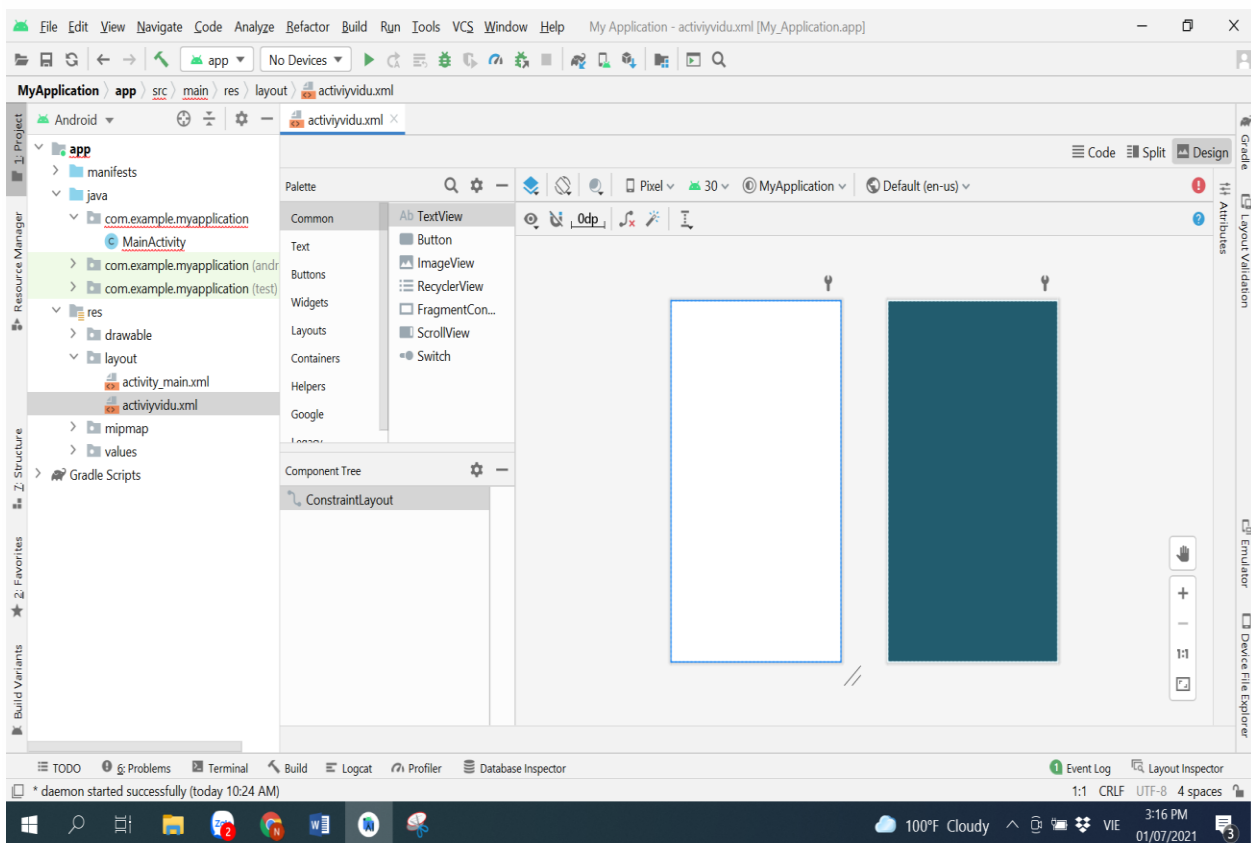
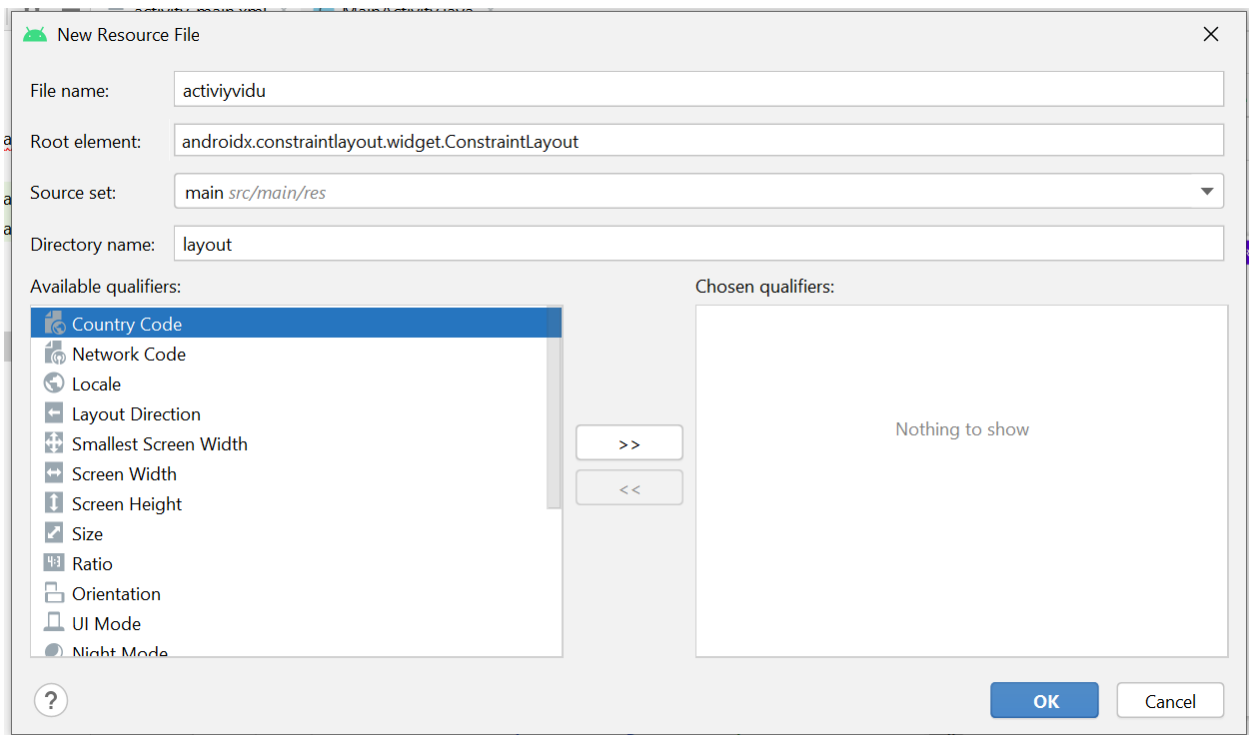
2.2 Trình tự thao tác

2.2.1. Tạo Layout mới

–Tạo một Layout mới: **File/** chọn **New/** chọn **Android XML File:**



–Đặt tên cho Layout trong mục File (ví dụ: **activityvidu**), chọn kiểu Layout rồi nhấn nút **OK**:



- Khi double - click vào layout activity vidu vừa tạo, sẽ xuất hiện màn hình để có thể kéo thả giao diện cho layout hoặc code cho layout.

2.2.2. Tạo *LinearLayout*

Ví dụ: Tạo giao diện như hình:

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

```
    xmlns:tools="http://schemas.android.com/tools"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
```

```
    android:orientation="vertical">
```

```
    <LinearLayout
```

```
        android:layout_width="match_parent"
```

```
        android:layout_height="wrap_content"
```

```
        android:orientation="horizontal">
```

```
        <ImageView
```

```
            android:layout_width="wrap_content"
```

```
            android:layout_height="wrap_content"
```

```
            android:src="@mipmap/ic_launcher" />
```

```
        <TextView
```

```
            android:layout_width="wrap_content"
```

```
            android:layout_height="wrap_content"
```

```
            android:layout_gravity="center"
```

```
            android:layout_marginLeft="20dp"
```

```
            android:text="Xin Chào! Tôi là ngôn ngữ lập trình Android" />
```

```
    </LinearLayout>
```

```
    <Button
```

```
        android:id="@+id/btnClick"
```

```
        android:layout_width="wrap_content"
```

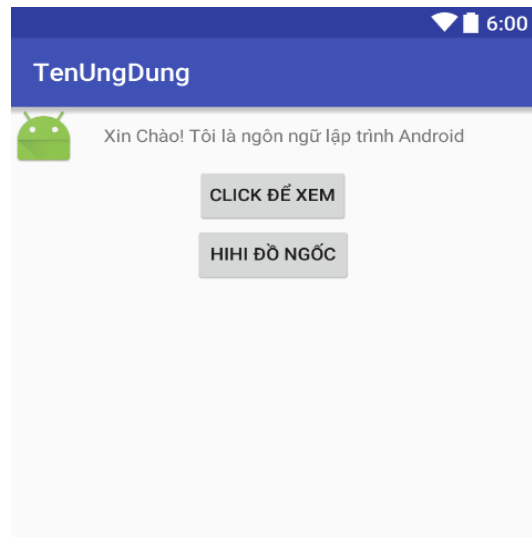
```
        android:layout_height="wrap_content"
```

```
        android:layout_gravity="center"
```

```
        android:padding="10dp"
```

```
        android:text="Click để xem" />
```

```
    <Button
```



```

android:id="@+id/btnHihi"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_gravity="center"
android:text="Hihi Đồ Ngốc" />

```

</LinearLayout>

2.2.3. Tạo RelativeLayout

Ví dụ: Tạo màn hình đăng nhập

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

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

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
```

```
    android:layout_margin="8dp"
```

```
    android:gravity="center"
```

```
    android:background="#e2e3eb">
```

```
<TextView
```

```
    android:id="@+id/textview"
```

```
    android:text="Đăng nhập"
```

```
    android:layout_centerHorizontal="true"
```

```
    android:textSize="20sp"
```

```
    android:textAllCaps="true"
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content" />
```

```
<EditText
```

```
    android:id="@+id/username"
```

```
    android:hint="Username"
```

```
    android:layout_below="@id/textview"
```

```
    android:layout_marginTop="40dp"
```

```
    android:layout_width="match_parent"
```



```

    android:layout_height="wrap_content" />
<EditText
    android:id="@+id/password"
    android:hint="passwords"
    android:inputType="textPassword"
    android:layout_below="@+id/username"
    android:layout_marginTop="8dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
<Button
    android:id="@+id/btnLogin"
    android:layout_below="@id/password"
    android:layout_toLeftOf="@id/recoverypassword"
    android:text="Đăng nhập"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
<Button
    android:id="@+id/recoverypassword"
    android:text="Quên MK"
    android:layout_centerHorizontal="true"
    android:layout_below="@id/password"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
<Button
    android:id="@+id/help"
    android:text="Trợ giúp"
    android:layout_below="@id/password"
    android:layout_toRightOf="@id/recoverypassword"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

```

</RelativeLayout>

2.2.4. Tạo *TableLayout*

Ví dụ: Tạo giao diện như mẫu

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

```
<TableLayout android:layout_height="match_parent"
```

```
    android:layout_width="match_parent"
```

```
    android:stretchColumns="1"
```

```
    android:layout_margin="8dp"
```

```
xmlns:android="http://schemas.android.com/apk/res/android">
```

```
<TableRow>
```

```
    <TextView
```

```
        android:text="ĐĂNG NHẬP"
```

```
        android:layout_span="2"
```

```
        android:textStyle="bold"
```

```
        android:textSize="20sp"
```

```
        android:gravity="center"/>
```

```
</TableRow>
```

```
<TableRow>
```

```
    <TextView
```

```
        android:text="Username"
```

```
        android:textStyle="bold" />
```

```
    <EditText android:hint="nhập id" />
```

```
</TableRow>
```

```
<TableRow>
```

```
    <TextView android:text="Password" android:textStyle="bold" />
```

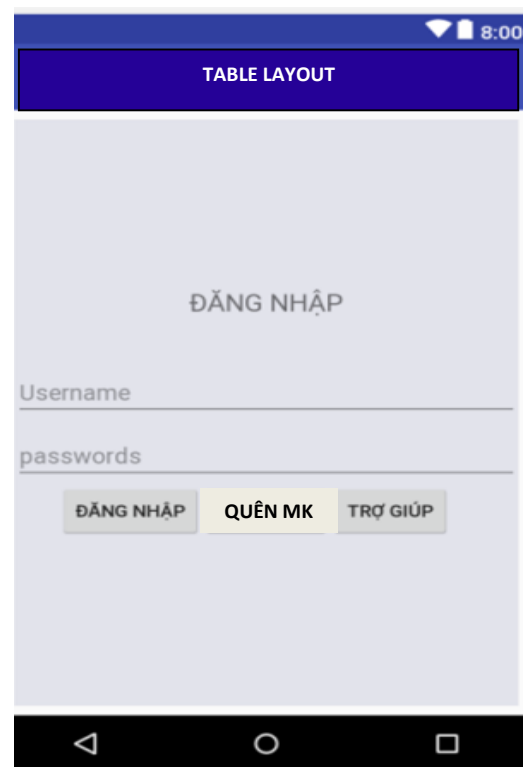
```
    <EditText android:inputType="textPassword"
```

```
        android:hint="nhập mật khẩu" />
```

```
</TableRow>
```

```
<LinearLayout
```

```
    android:gravity="center"
```



```

android:orientation="horizontal">
<Button
    style="@style/Widget.AppCompat.Button.Colored"
    android:text="Ok"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
<Button
    style="@style/Widget.AppCompat.Button.Colored"
    android:backgroundTint="#291010"
    android:text="Hủy"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
<Button
    android:text="Quên MK"
    style="@style/Widget.AppCompat.Button.Colored"
    android:backgroundTint="#291010"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
</LinearLayout>
</TableLayout>

```

2.2.5. Tạo *FrameLayout*

Ví dụ: Tạo layout theo mẫu

```

<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:padding="10dp"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
<!--Ảnh kín FrameLayout-->
<ImageView
    android:src="@drawable/butterfly"

```

```

    android:scaleType="centerCrop"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />

```

```
<Button
```

```

    android:layout_gravity="center|right"
    android:layout_marginRight="20dp"
    android:text="Button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

```

```
<ImageView
```

```

    android:layout_gravity="center|bottom"
    android:src="@drawable/xuanthulab"
    android:adjustViewBounds="true"
    android:layout_width="150dp"
    android:layout_height="wrap_content" />

```

```
<TextView
```

```

    android:layout_gravity="center|top"
    android:layout_marginTop="10dp"
    android:text="Đây là một TextView"
    android:layout_width="100dp"
    android:gravity="center"
    android:layout_height="wrap_content" />

```

```
</FrameLayout>
```

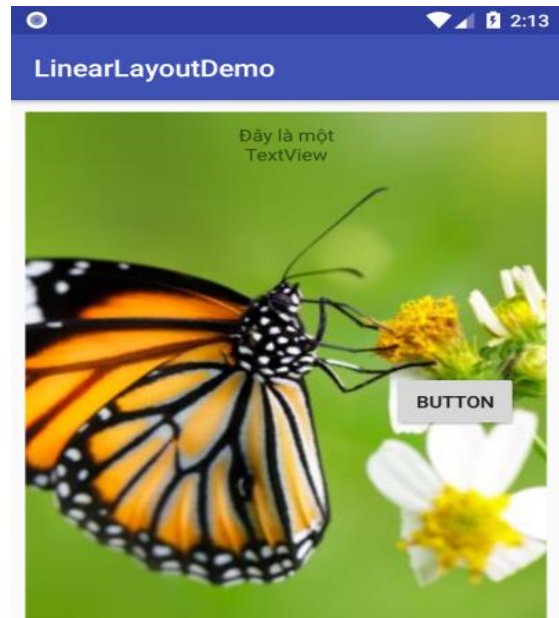
2.2.6. Tạo ConstraintLayout

Ví dụ: Tạo layout theo mẫu

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"

```



```

android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity"
tools:layout_editor_absoluteY="81dp">
<Button
    android:id="@+id/b1"
    android:layout_width="wrap_content"
    android:layout_height="0dp"
    android:textColor="#1eb729"
    android:text="B1 - HEAD"
    app:layout_constraintBottom_toTopOf="@id/b5"
    app:layout_constraintHorizontal_chainStyle="spread_inside"
    app:layout_constraintRight_toLeftOf="@id/b2"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintHorizontal_weight="1"
    app:layout_constraintVertical_weight="2"
    app:layout_constraintTop_toBottomOf="@id/b4" />

```

```

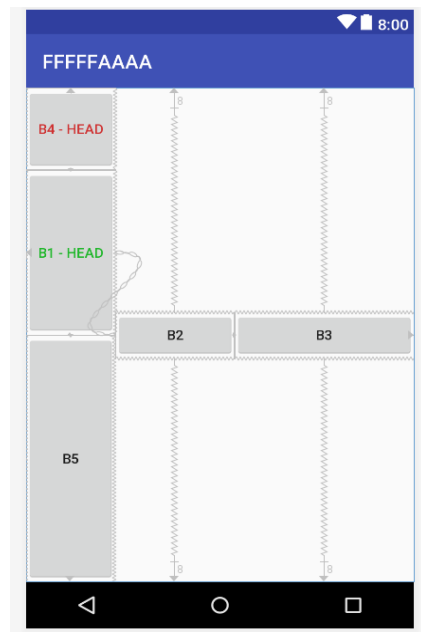
<Button
    android:id="@+id/b2"
    android:text="B2"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginTop="8dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toRightOf="@id/b1"
    app:layout_constraintRight_toLeftOf="@id/b3"
    app:layout_constraintHorizontal_weight="2"
    app:layout_constraintTop_toTopOf="parent" />

```

```

<Button

```



```
android:id="@+id/b3"
android:text="B3"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginBottom="8dp"
android:layout_marginTop="8dp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_weight="3"
app:layout_constraintLeft_toRightOf="@id/b2"
app:layout_constraintTop_toTopOf="parent" />
```

<Button

```
android:id="@+id/b4"
android:text="B4 - HEAD"
app:layout_constraintVertical_chainStyle="spread_inside"
android:textColor="#cf3131"
app:layout_constraintBottom_toTopOf="@id/b1"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintLeft_toLeftOf="@id/b1"
app:layout_constraintVertical_weight="1"
android:layout_width="wrap_content"
android:layout_height="0dp" />
```

<Button

```
android:id="@+id/b5"
android:text="B5"
app:layout_constraintTop_toBottomOf="@id/b1"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintLeft_toLeftOf="@id/b1"
android:layout_width="wrap_content"
app:layout_constraintVertical_weight="3"
```

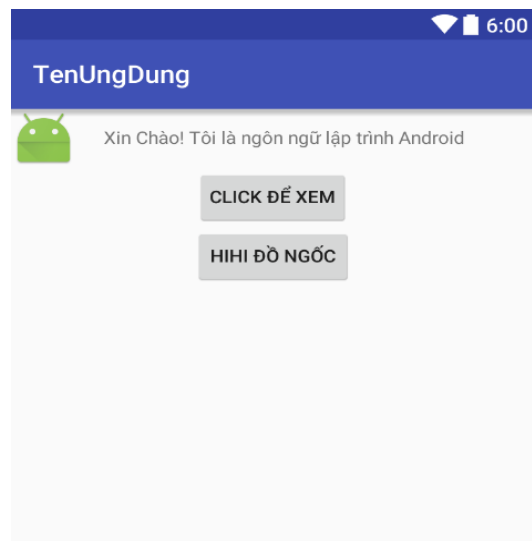
```
        android:layout_height="0px" />
</android.support.constraint.ConstraintLayout>
```

BÀI TẬP

Bài 1. Tạo giao diện như hình :

Gợi ý:

```
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="horizontal">
        <ImageView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:src="@mipmap/ic_launcher" />
        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_gravity="center"
            android:layout_marginLeft="20dp"
            android:text="Xin Chào! Tôi là ngôn ngữ lập trình Android" />
    </LinearLayout>
    <Button
        android:id="@+id/btnClick"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```



```

android:layout_gravity="center"
android:padding="10dp"
android:text="Click để xem" />

```

```

<Button
    android:id="@+id/btnHihi"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:text="Hihi Đồ Ngốc" />

```

```
</LinearLayout>
```

Bài 2. Tạo màn hình đăng nhập như hình :

Gợi ý dùng Relative Layout:

```

<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:layout_margin="8dp"
    android:gravity="center"
    android:background="#e2e3eb">
    <TextView
        android:id="@+id/textview"
        android:text="Đăng nhập"
        android:layout_centerHorizontal="true"
        android:textSize="20sp"
        android:textAllCaps="true"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <EditText
        android:id="@+id/username"
        android:hint="Username"

```



```

    android:layout_below="@id/textview"
    android:layout_marginTop="40dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
<EditText
    android:id="@+id/password"
    android:hint="passwords"
    android:inputType="textPassword"
    android:layout_below="@+id/username"
    android:layout_marginTop="8dp"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" />
<Button
    android:id="@+id/btnLogin"
    android:layout_below="@id/password"
    android:layout_toLeftOf="@id/recoverypassword"
    android:text="Đăng nhập"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
<Button
    android:id="@+id/recoverypassword"
    android:text="Quên MK"
    android:layout_centerHorizontal="true"
    android:layout_below="@id/password"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
<Button
    android:id="@+id/help"
    android:text="Trợ giúp"
    android:layout_below="@id/password"

```

```
    android:layout_toRightOf="@id/recoverypassword"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />
```

```
</RelativeLayout>
```

- Gợi ý dùng Relative Layout:

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

```
<TableLayout android:layout_height="match_parent"
```

```
    android:layout_width="match_parent"
```

```
    android:stretchColumns="1"
```

```
    android:layout_margin="8dp"
```

```
xmlns:android="http://schemas.android.com/apk/res/android">
```

```
    <TableRow>
```

```
        <TextView
```

```
            android:text="ĐĂNG NHẬP"
```

```
            android:layout_span="2"
```

```
            android:textStyle="bold"
```

```
            android:textSize="20sp"
```

```
            android:gravity="center"/>
```

```
    </TableRow>
```

```
    <TableRow>
```

```
        <TextView
```

```
            android:text="Username"
```

```
            android:textStyle="bold" />
```

```
        <EditText android:hint="nhập id" />
```

```
    </TableRow>
```

```
    <TableRow>
```

```
        <TextView android:text="Password" android:textStyle="bold" />
```

```
        <EditText android:inputType="textPassword"
```

```
            android:hint="nhập mật khẩu" />
```

```
    </TableRow>
```

```

<LinearLayout
    android:gravity="center"
    android:orientation="horizontal">
    <Button
        style="@style/Widget.AppCompat.Button.Colored"
        android:text="Ok"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <Button
        style="@style/Widget.AppCompat.Button.Colored"
        android:backgroundTint="#291010"
        android:text="Hủy"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
    <Button
        android:text="Quên MK"
        style="@style/Widget.AppCompat.Button.Colored"
        android:backgroundTint="#291010"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content" />
</LinearLayout>

```

```
</TableLayout>
```

Bài 3. Tạo giao diện như hình :

Gợi ý:

```

<?xml version="1.0" encoding="utf-8"?>
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:padding="10dp"
    android:layout_width="match_parent"
    android:layout_height="match_parent">
<!--Ảnh kín FrameLayout-->

```

```

<ImageView
    android:src="@drawable/butterfly"
    android:scaleType="centerCrop"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />

```

```

<Button
    android:layout_gravity="center|right"
    android:layout_marginRight="20dp"
    android:text="Button"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content" />

```

```

<ImageView
    android:layout_gravity="center|bottom"
    android:src="@drawable/xuanthulab"
    android:adjustViewBounds="true"
    android:layout_width="150dp"
    android:layout_height="wrap_content" />

```

```

<TextView
    android:layout_gravity="center|top"
    android:layout_marginTop="10dp"
    android:text="Đây là một TextView"
    android:layout_width="100dp"
    android:gravity="center"
    android:layout_height="wrap_content" />

```

```

</FrameLayout>

```

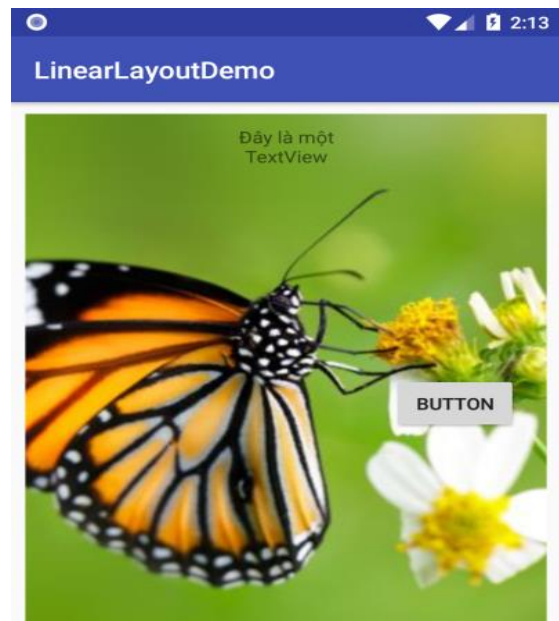
Bài 4. Tạo giao diện như hình :

Gợi ý:

```

<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"

```



```

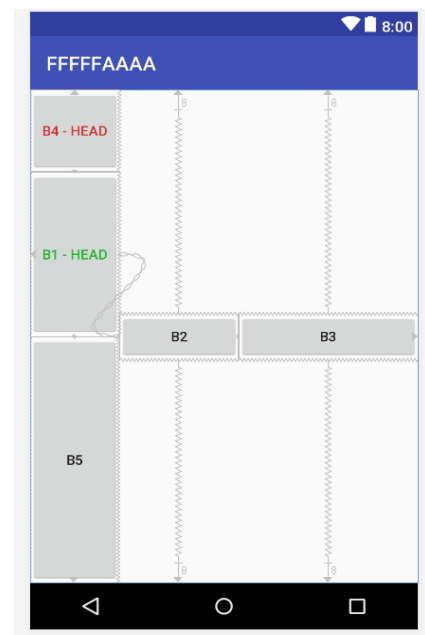
xmlns:app="http://schemas.android.com/apk/res-auto"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
tools:context=".MainActivity"
tools:layout_editor_absoluteY="81dp">
<Button
    android:id="@+id/b1"
    android:layout_width="wrap_content"
    android:layout_height="0dp"
    android:textColor="#1eb729"
    android:text="B1 - HEAD"
    app:layout_constraintBottom_toTopOf="@id/b5"
    app:layout_constraintHorizontal_chainStyle="spread_inside"
    app:layout_constraintRight_toLeftOf="@id/b2"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintHorizontal_weight="1"
    app:layout_constraintVertical_weight="2"
    app:layout_constraintTop_toBottomOf="@id/b4" />

```

```

<Button
    android:id="@+id/b2"
    android:text="B2"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginTop="8dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toRightOf="@id/b1"
    app:layout_constraintRight_toLeftOf="@id/b3"
    app:layout_constraintHorizontal_weight="2"

```



```

    app:layout_constraintTop_toTopOf="parent" />
<Button
    android:id="@+id/b3"
    android:text="B3"
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginBottom="8dp"
    android:layout_marginTop="8dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_weight="3"
    app:layout_constraintLeft_toRightOf="@id/b2"
    app:layout_constraintTop_toTopOf="parent" />
<Button
    android:id="@+id/b4"
    android:text="B4 - HEAD"
    app:layout_constraintVertical_chainStyle="spread_inside"
    android:textColor="#cf3131"
    app:layout_constraintBottom_toTopOf="@id/b1"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintLeft_toLeftOf="@id/b1"
    app:layout_constraintVertical_weight="1"
    android:layout_width="wrap_content"
    android:layout_height="0dp" />
<Button
    android:id="@+id/b5"
    android:text="B5"
    app:layout_constraintTop_toBottomOf="@id/b1"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintLeft_toLeftOf="@id/b1"

```

```

android:layout_width="wrap_content"
app:layout_constraintVertical_weight="3"
android:layout_height="0px" />
</android.support.constraint.ConstraintLayout>

```

3. TÓM TẮT TRÌNH TỰ THỰC HIỆN HOẶC QUY TRÌNH CÔNG NGHỆ

<i>TT</i>	<i>Tên các bước công việc</i>	<i>Dụng cụ, thiết bị, vật tư</i>	<i>Yêu cầu kỹ thuật</i>	<i>Các chú ý về an toàn lao động</i>
1	Khởi động Android Studio	Phần mềm Android Studio Máy tính.	Cài đặt chính xác phần mềm.	Đảm bảo an toàn cho người và thiết bị.
2	Thực hành thiết kế và code cho các nút lệnh	Phần mềm Android Studio Máy tính.	Cài đặt chính xác phần mềm.	Đảm bảo an toàn cho người và thiết bị.
3	Kiểm tra kết quả	Phần mềm Android Studio Máy tính.	Cài đặt chính xác phần mềm.	Đảm bảo an toàn cho người và thiết bị.

BÀI 4: THIẾT KẾ GIAO DIỆN NGƯỜI DÙNG VỚI CÁC ĐIỀU KHIỂN CƠ BẢN

1. MỤC TIÊU

Thiết kế được giao diện người dùng với các điều khiển cơ bản. Sử dụng được thành thạo Visual Android để xây dựng ứng dụng.

Nghiêm túc, tự giác trong học tập. Đảm bảo an toàn cho người và thiết bị.

2. NỘI DUNG

2.1. Lý thuyết liên quan

2.1.1. Các thuộc tính và sự kiện chung

- Một số thuộc tính chung thường dùng:

`android:id="@+id/tvHello"` - tên để quản lý, trong quá trình lập trình muốn lấy giá trị đó dùng cú pháp ***R.id.tên_id***

`android:layout_width="wrap_content"` – độ rộng

`android:layout_height="wrap_content"` – chiều cao

`android:textSize="28sp"` – cỡ chữ

`android:textColor="#f00"` – màu chữ

`android:background="#000"` – màu nền

`android:textStyle="bold"` – style chữ

`android:padding="10dp"`

`android:layout_margin="10dp"`

`android:text="Hello iNET" />` – nội dung

Tất cả các widget phải đặt giá trị `android:layout_width` and `android:layout_height`. Các giá trị tùy chọn cho height và width có thể là một trong ba trường hợp sau:

- kích thước cụ thể: ví dụ 125dip.
 - `wrap_content`: co vừa với nội dung phía trong.
 - `fill_parent`: bằng không gian của view chứa nó.
- Sự kiện thường dùng: Tất cả các View đều có khả năng lắng nghe các sự kiện chạm/ vuốt trên màn hình bằng sự kiện Click thông qua phương thức `setOnClickListener()` và xử lý các công việc trong phương thức `onClick()`.

```
Button btnClick = (Button)findViewById(R.id.btn_click);
```

```

btnClick.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
    }
});

```

2.1.2. *TextView*

- Chức năng: hiển thị thông tin mà không cho phép người dùng chỉnh sửa.

- Một số câu lệnh java thường dùng:

+ Đặt nội dung lên textView: **Tên_biến_textView.setText("nội dung hiển thị");**

Ví dụ:

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:layout_marginTop="20dp"
    android:text="XIN CHÀO CÁC BẠN"
    android:textSize="30dp" />

```

2.1.3. *EditText*

- Chức năng: hiển thị thông tin cho phép người dùng chỉnh sửa.

- Một số câu lệnh java thường dùng:

+ Đặt nội dung lên EditText: **Tên_biến_EditText.setText("nội dung hiển thị");**

+ Lấy nội dung từ EditText: **Tên_biến_EditText.getText();**

Ví dụ:

```

<EditText
    android:id="@+id/editTextTextPersonName3"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="120dp"

```

```

android:ems="10"
android:hint="Gõ ID"
android:inputType="textPersonName"
android:text="Name" />

```

2.1.4. Button

- Chức năng: thiết lập sự kiện khi người dùng chọn lựa.
- Sự kiện thường dùng: `setOnClickListener()`

```

Button btnClick = (Button)findViewById(R.id.btn_click);
btnClick.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
    }
});

```

2.1.5. CheckBox, RadioButton

- Thuộc tính thường dùng:

android:checked="true": thiết lập ở trạng thái checked mặc định.

2.1.6. Image view: hiển thị ảnh.

- Thuộc tính thường:

android:src - Lấy đường dẫn ảnh: copy ảnh vào thư mục *mipmap* hoặc *drawable* trong android > Kéo thả view *ImageView* vào activity > Chọn *src* của *ImageView* > chọn ảnh.

android:background - đặt màu nền hoặc ảnh nền cho view.

android:scaleTyle - Cho biết tỉ lệ co giãn hoặc vị trí của ảnh so với khung của view

Giá trị	Sử dụng
center	<code>ImageView.ScaleType.CENTER</code> : Đặt ảnh vào giữa <i>ImageView</i> , không có thay đổi tỷ lệ ảnh.
centerCrop	<code>ImageView.ScaleType.CENTER_CROP</code> : Đặt ảnh vào giữa <i>ImageView</i> , có thu phóng ảnh (nhưng giữ nguyên tỉ lệ cao / rộng) sao cho ảnh phủ kín hết cả <i>ImageView</i> (phần thừa bị cắt)

centerInside	ImageView.ScaleType.CENTER_INSIDE: Đặt ảnh vào giữa ImageView, có thu phóng ảnh (nhưng giữ nguyên tỉ lệ cao / rộng) sao cho toàn bộ các phần của ảnh hiển thị trên ImageView.
fitCenter	ImageView.ScaleType.FIT_CENTER: Đặt ảnh vào giữa ImageView, có thu phóng ảnh (nhưng giữ nguyên tỉ lệ cao / rộng) sao cho toàn bộ các phần của ảnh hiển thị trên ImageView.
fitEnd fitStart	ImageView.ScaleType.FIT_END, ImageView.ScaleType.FIT_START: Co ảnh vừa View, vị trí ảnh ở cuối (ở đầu) ImageView
fitXY	ImageView.ScaleType.FIT_XY: Co ảnh vừa khít cả chiều rộng và cao.

- Câu lệnh java thường dùng:

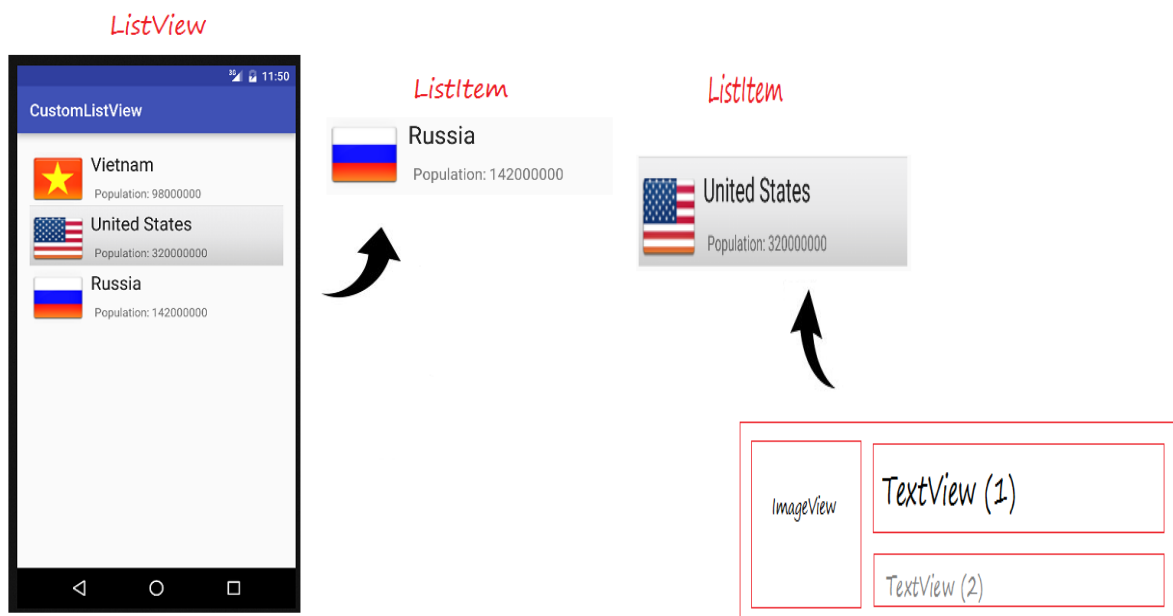
- Đặt ảnh vào ImageView

```
this.imageView.setImageResource (R.drawable.triceratops_1);
```

- Xóa ảnh: imageView.setImageResource(0);

2.1.7. ListView:

Một ListView được tạo từ một danh sách các ListItem. ListItem là một dòng riêng lẻ trong listview nơi mà dữ liệu sẽ được hiển thị, một ListItem được tạo bởi một số View.



Android Adapter (bộ tiếp nối): quản lý dữ liệu và ghép nối **ListView** với các dòng riêng lẻ (**ListItem**) của nó thông qua phương thức **setAdapter**.

2.2 Trình tự thao tác

2.2.1. *TextView, Button*

Ví dụ: Tạo giao diện gồm **TextView** hiển thị một đoạn văn bản, **Button Edit**: thông báo người dùng vừa bấm vào nút lệnh theo mẫu.

Bước 1- Khởi động Android Studio, tạo project mới.

Bước 2 – Tạo file activity_main.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<android.support.constraint.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    xmlns:app="http://schemas.android.com/apk/res-auto">
    <TextView
        android:id="@+id/txtnoidung"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginBottom="8dp"
        android:layout_marginTop="8dp"
        android:background="#2aab1194"
        android:padding="5dp"
        android:text="Đây là một đoạn văn trong View của
layout: lesson2_activiy. Lập trình Android bài 2."
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintTop_toTopOf="parent" />
    <Button
        android:id="@+id/editbutton"
        android:text="Edit"
```



```

    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    app:layout_constraintTop_toBottomOf="@+id/textView"
    app:layout_constraintLeft_toLeftOf="parent"
    app:layout_constraintRight_toRightOf="parent" />
</android.support.constraint.ConstraintLayout>

```

Bước 3 - Tạo file java: MainActivity.java

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    TextView textView = findViewById(R.id.txtnoidung); //Lấy view txtnoidung
    Button edibutton = findViewById(R.id.editbutton); // Lấy view editbutton
    //Tạo listener cho sự kiện Click
    edibutton.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Toast.makeText(MainActivity.this, " Vừa bấm vào Edit ", Toast.LENGTH_SHORT);
        }
    });
    //Gán Listener cho editbutton
}

```

2.2.2. CheckBox, RadioButton

Ví dụ: Tạo giao diện Đăng kí tìm bạn bốn phương như mẫu gồm: Button OK: Kiểm tra và thông báo khi người dùng chưa nhập đủ thông tin. Button Clear: xóa các thông tin cũ trên giao diện

Bước 1- Khởi động Android Studio, tạo project mới.

Bước 2- Tạo file activity_main.xml theo giao diện mẫu.

Bước 3- Tạo file File MainActivity.java:

```

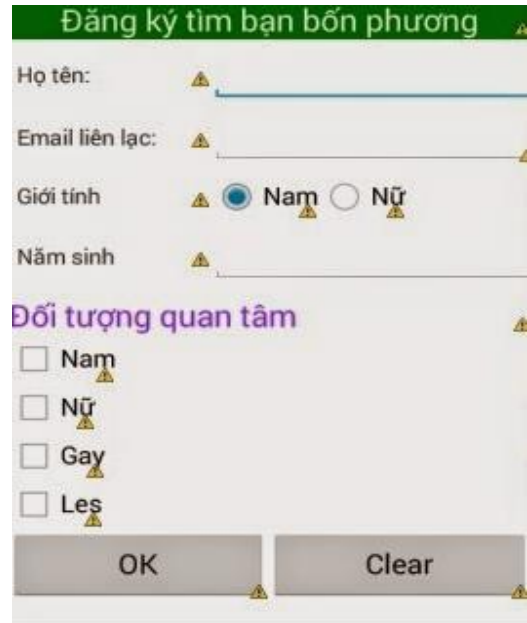
public class MainActivity extends ActionBarActivity {
    Button btnOK, btnClear;
    EditText edtHoten, edtEmail, edtNamSinh;
}

```

```

RadioGroup rgGioiTinh;
CheckBox cbNam, cbNu, cbGay, cbLes;
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    getControlView();
    btnOK.setOnClickListener(new
OnClickListener() {
        @Override
        public void onClick(View v) {
            xuLyNhap();
        }
    });
    btnClear.setOnClickListener(new
OnClickListener() {
        @Override
        public void onClick(View v) {
            xuLyXoa();
        }
    });
private void getControlView() {
    btnOK = (Button)findViewById(R.id.btnOK);
    btnClear = (Button)findViewById(R.id.btnClear);
    edtHoten = (EditText)findViewById(R.id.edtHoTen);
    edtEmail = (EditText)findViewById(R.id.edtEmail);
    edtNamSinh = (EditText)findViewById(R.id.edtNamSinh);
    rgGioiTinh = (RadioGroup)findViewById(R.id.rgGioiTinh);
    cbNam = (CheckBox)findViewById(R.id.cbNam);
    cbNu = (CheckBox)findViewById(R.id.cbNu);
    cbLes = (CheckBox)findViewById(R.id.cbLes);
    cbGay = (CheckBox)findViewById(R.id.cbGay);

```



```

}
protected void xuLyNhap() {
    String hoTen = edtHoten.getText().toString();
    String email = edtEmail.getText().toString();
    String namsinh = edtNamSinh.getText().toString();
    if(hoTen==null || hoTen.equalsIgnoreCase("")){
        Toast.makeText(getBaseContext(), "Bạn phải nhập họ tên",
            Toast.LENGTH_LONG).show();
        return;
    }
    if(email==null || email.equalsIgnoreCase("")){
        Toast.makeText(getBaseContext(), "Bạn phải nhập email",
            Toast.LENGTH_LONG).show();
        return;
    }
    if(namsinh==null || namsinh.equalsIgnoreCase("")){
        Toast.makeText(getBaseContext(), "Bạn phải nhập năm sinh",
            Toast.LENGTH_LONG).show();
        return;
    }
    String gioiTinh;
    if(rgGioiTinh.getCheckedRadioButtonId()==R.id.rbNam){
        gioiTinh = "Nam";
    }else{
        gioiTinh = "Nữ";
    }
    String doiTuongQuanTam = "";
    if(cbNam.isChecked()){
        doiTuongQuanTam += "Nam"+" ";
    }
}

```

```

if(cbNu.isChecked()){
    doiTuongQuanTam += "Nữ"+" ";
}
if(cbLes.isChecked()){
    doiTuongQuanTam += "Les"+" ";
}
if(cbGay.isChecked()){
    doiTuongQuanTam += "Gay"+" ";
}
if(doiTuongQuanTam == null || doiTuongQuanTam.equalsIgnoreCase("")){
    Toast.makeText(getBaseContext(), "Bạn cần chọn đối tượng quan tâm",
        Toast.LENGTH_LONG).show();
    return;
}
AlertDialog.Builder ad = new AlertDialog.Builder(this);
ad.setTitle("Thông tin cá nhân");
ad.setPositiveButton("Close", new DialogInterface.OnClickListener() {
    @Override
    public void onClick(DialogInterface dialog, int which) {
        dialog.cancel();
    }
});
String message = "";
message += "Họ tên: "+hoTen + "\n";
message += "Email: "+email + "\n";
message += "Giới tính: "+gioiTinh + "\n";
message += "Năm sinh: "+namsinh + "\n";
message += "Đối tượng quan tâm : "+doiTuongQuanTam + "\n";
ad.setMessage(message);
ad.create().show();
}

```

```

protected void xuLyXoa() {
    edtHoten.setText("");
    edtNamSinh.setText("");
    edtEmail.setText("");
    cbNam.setChecked(false);
    cbNu.setChecked(false);
    cbLes.setChecked(false);
    cbGay.setChecked(false);
}
}

```

2.2.3. *ImageView*

Ví dụ: Tạo giao diện như hình. Nhấn vào 2 Button sẽ hiển thị ảnh tương ứng.

Bước 1- Khởi động Android Studio, tạo project mới.

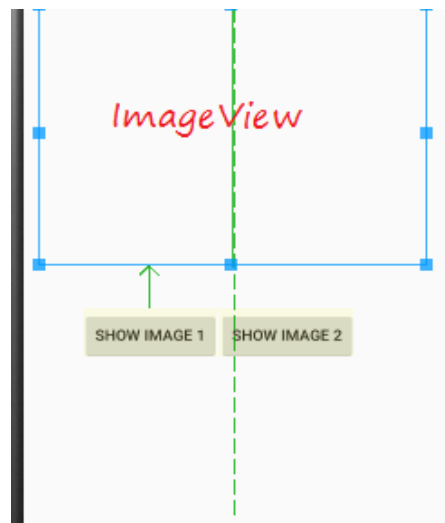
Bước 2- Tạo file activity_main.xml theo giao diện mẫu.

Bước 3- Tạo file MainActivity.java:

```

private ImageView imageView;
private Button button1;
private Button button2;
@Override
protected void onCreate(Bundle savedInstanceState)
{
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    this.imageView = (ImageView)
this.findViewById(R.id.imageView);
    this.button1 = (Button) this.findViewById(R.id.button1);
    this.button2 = (Button) this.findViewById(R.id.button2);
    this.button1.setOnClickListener(new Button.OnClickListener() {
@Override
public void onClick(View v) {
    showImage1();
}
});

```



```

this.button2.setOnClickListener(new Button.OnClickListener() {
    @Override
    public void onClick(View v) {
        showImage2();
    } });
private void showImage1() {
    this.imageView.setImageResource(R.drawable.triceratops_1);
}
private void showImage2() {
    this.imageView.setImageResource(R.drawable.triceratops_2);
}

```

2.2.4. *ListView*

Ví dụ 1: Tạo ListView hiển thị dữ liệu là một danh sách gồm ID, tên sản phẩm, giá:

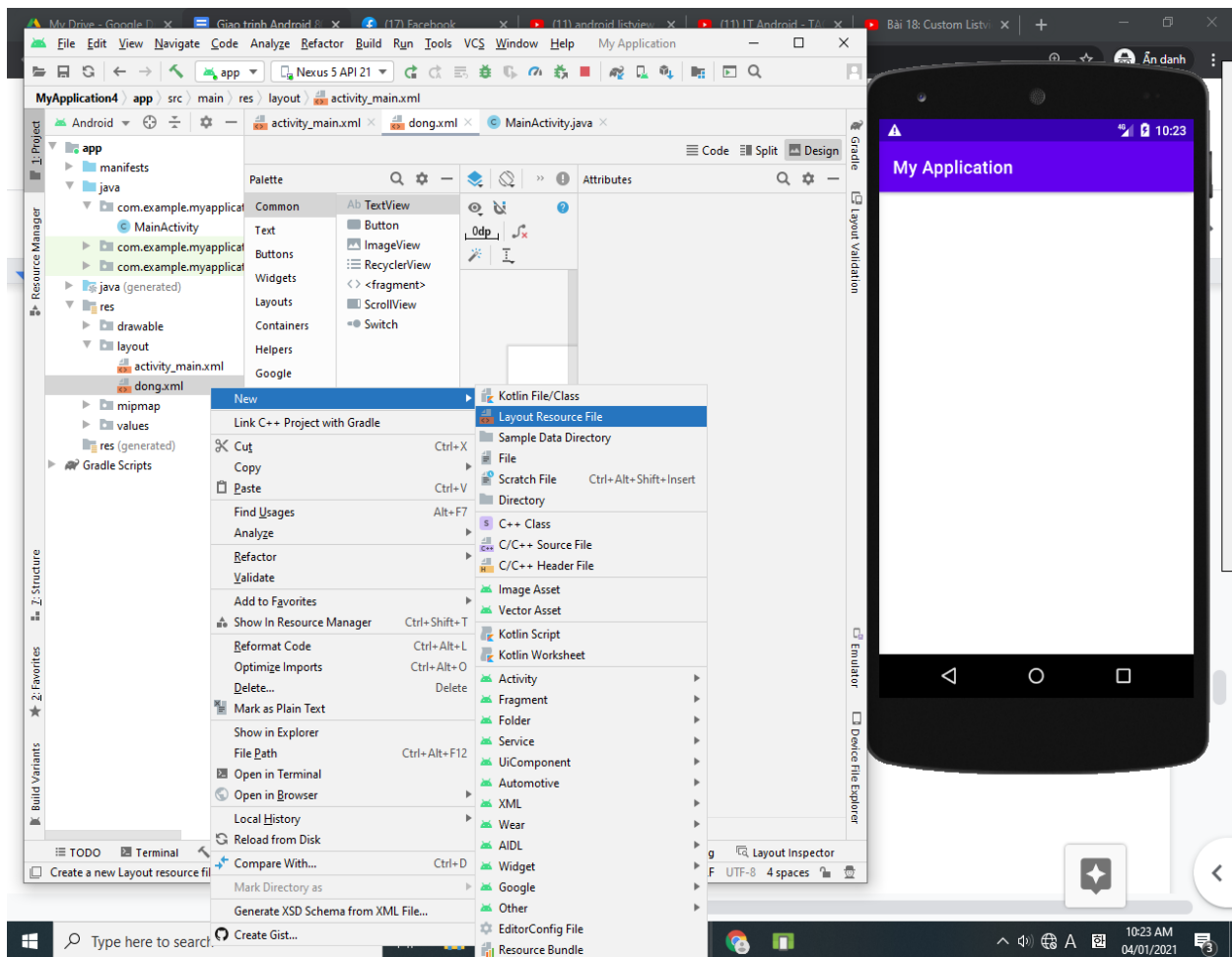
Bước 1: Tạo ListView trong file **activity.xml**

```

<ListView
    android:id="@+id/listproduct"
    android:layout_width="match_parent"
    android:layout_height="match_parent" />

```

Bước 2: Tạo file xml mới (file có tên **dongsanpham.xml**): Chọn **File > New > Layout Resource File > đặt tên **dongsanpham.xml** > tạo các view chứa dòng dữ liệu trong Listview**



<LinearLayout

```

android:layout_width="match_parent"
android:layout_height="match_parent"
android:orientation="horizontal">

```

<TextView

```

    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/txtID" />

```

<TextView

```

    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:id="@+id/txtTen" />

```

<TextView

```

    android:layout_width="wrap_content"

```

```
        android:layout_height="wrap_content"
        android:id="@+id/txtDonGia"    />
</LinearLayout>
```

Bước 3. Tạo file **sanpham.class** khai báo dòng sản phẩm gồm: **ID, tên sản phẩm, giá:**

Chọn File > New > chọn Java Class > đặt tên class là **sanpham.class**.

```
class sanpham{
    String name;
    int price;
    int productID;
    public Product(int productID, String name, int price) {
        this.name = name;
        this.price = price;
        this.productID = productID;
    }
}
```

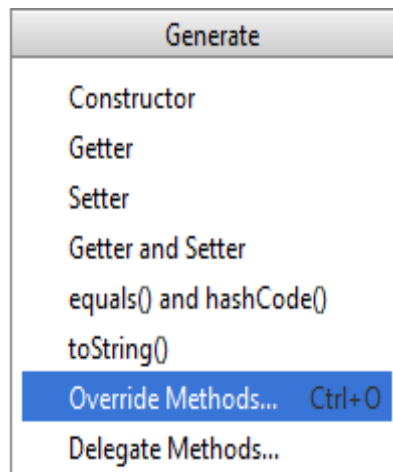
Bước 4: Tạo file **listViewAdapter.class**: Chọn File > New > chọn Java Class > đặt tên class là **listViewAdapter** > Bổ sung thêm **extends BaseAdapter** như phía dưới:

```
class listViewAdapter extends BaseAdapter {
}
```

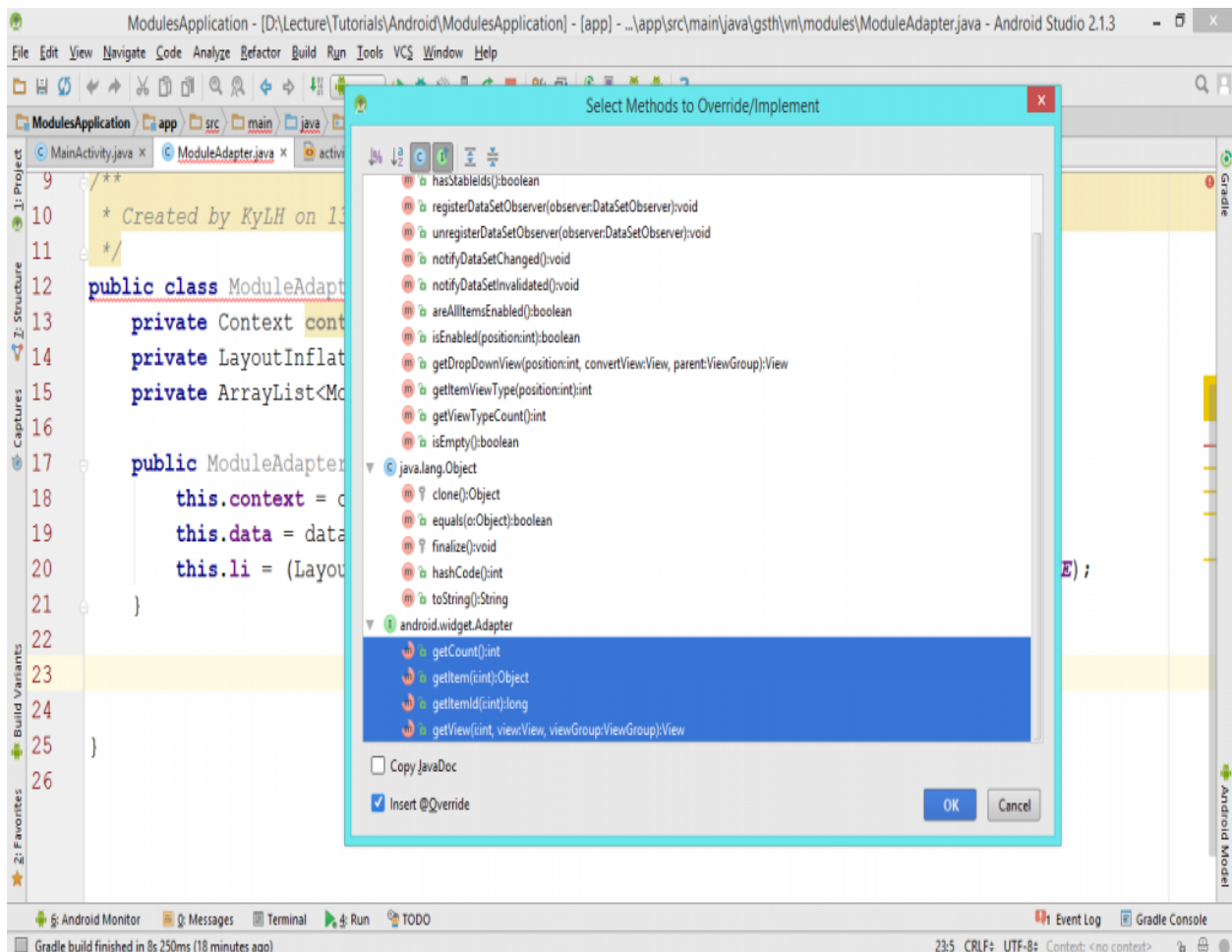
- Thêm đoạn code sau vào lớp **listViewAdapter**:

```
public class listViewAdapter extends BaseAdapter {
    Context context;
    int mylayout;
    List<sanpham> sanphamList;
    public listViewAdapter(Context context, int mylayout, List<sanpham> sanphamList) {
        this.context = context;
        this.mylayout = mylayout;
        this.sanphamList = sanphamList;
    }
}
```

- Thêm tiếp các phương thức **getCount()**, **getItem()**, **getItemId** và **getView()** bằng cách nhấn tổ hợp phím **Alt + Ins** -> chọn **Override Methods...**



Tại màn hình **Select Methods to Override/Implement** -> chọn các phương thức **getCount()**, **getItem()**, **getItemId** và **getView()**



Lúc này file có các câu lệnh như sau:

```

class listViewAdapter extends BaseAdapter {
    @Override
    public int getCount() {
        //Cần trả về số phần tử mà ListView hiển thị
        return 0;
    }
    @Override
    public Object getItem(int position) {
        //Cần trả về đối tượng dữ liệu phần tử ở vị trí position
        return null;
    }
    @Override
    public long getItemId(int position) {
        //Trả về một ID liên quan đến phần tử ở vị trí position
        return 0;
    }
    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        return null;
    }
}

```

- Đối với phương thức **getCount()** chúng ta thay đổi lại như sau:

```

@Override
public int getCount() {
    return data.size();
}

```

- **Sửa nội dung theo bài mẫu sau:**

```

public class listViewAdapter extends BaseAdapter {
    Context context;
    int mylayout;
    List<sanpham> sanphamList;
}

```

```

public listViewAdapter(Context context, int mylayout, List<sanpham> sanphamList) {
    this.context = context;
    this.mylayout = mylayout;
    this.sanphamList = sanphamList;
}
@Override
public int getCount() {
    return sanphamList.size();
}
@Override
public Object getItem(int i) {
    return null;
}
@Override
public long getItemId(int i) {
    return 0;
}
@Override
public View getView(int i, View view, ViewGroup viewGroup) {
    LayoutInflater inflater = (LayoutInflater)
        context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    view = inflater.inflate(mylayout, null);
    TextView txtTen = (TextView) view.findViewById(R.id.txtTen);
    txtTen.setText(sanphamList.get(i).name);
    TextView txtGia = (TextView) view.findViewById(R.id.txtDonGia);
    txtGia.setText(sanphamList.get(i).price + " ");
    TextView txtID = (TextView) view.findViewById(R.id.txtID);
    txtGia.setText(sanphamList.get(i).productID + " ");
    return view;
}

```

```
}
```

Bước 5: Mở file **MainActivity**, trong hàm **onCreate**, gán **listViewAdapter** đã tạo ở trên là nguồn cấp dữ liệu cho nó bằng câu lệnh:

```
listViewProduct = findViewById(R.id.listproduct);  
listViewProduct.setAdapter(listViewAdapter );
```

- Sửa MainActivity như sau:

```
public class MainActivity extends AppCompatActivity {  
    ArrayList<sanpham> listProduct; //Mảng dữ liệu sản phẩm  
    ListView listView;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        //Khởi tạo ListProduct, tự sinh một số dữ liệu  
        listProduct = new ArrayList<sanpham>();  
        listProduct.add(new sanpham(1, "Iphone 6", 500));  
        listProduct.add(new sanpham(2, "Iphone 7", 700));  
        listProduct.add(new sanpham(3, "Sony Abc", 800));  
        listProduct.add(new sanpham(4, "Samsung XYZ", 900));  
        listProduct.add(new sanpham(5, "SP 5", 500));  
        listProduct.add(new sanpham(6, "SP 6", 700));  
        listProduct.add(new sanpham(7, "SP 7", 800));  
        listProduct.add(new sanpham(8, "SP 8", 900));  
        listView = findViewById(R.id.listView);  
        listViewAdapter listViewAdapter= new listViewAdapter  
        (MainActivity.this,R.layout.dongsanpham,listProduct);  
        listView.setAdapter(listViewAdapter);  
    }  
}
```

Bước 6: Chạy chương trình có giao diện như hình.

My Application
Iphone 61
Iphone 72
Sony Abc3
Samsung XYZ4
SP 55
SP 66
SP 77
SP 88

BÀI TẬP

Bài 1. Tạo giao diện theo mẫu, nhấn nút OK thì hiển thị dòng chữ XINCHAO lên màn hình.

Gợi ý:

Bước 1: Tạo giao diện, button đặt ID là btnOK

Bước 2: Mở file MainActivity, khai báo button, viết code cho sự kiện click

```
public class MainActivity extends AppCompatActivity {
```

```
    Button btnOK;
```

```
    @Override
```

```
    protected void onCreate(Bundle savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
        btnOK = findViewById(R.id.btnOK);
```

```
        btnOK.setOnClickListener(new
```

```
View.OnClickListener() {
```

```
            @Override
```

```
            public void onClick(View view) {
```

```
                Toast.makeText(MainActivity.this,
```

```
"XINCHAO", Toast.LENGTH_SHORT);
```

```
            }    });    }
```



Bài 2: Tạo giao diện theo mẫu. Button Send: Hiển thị dữ liệu được lấy trong EditText. Button Close: thoát khỏi ứng dụng.

Gợi ý:

-File activitymain.xml

```

<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical">
    <EditText
        android:id="@+id/editTextTextPersonName"
        android:layout_width="match_parent"
        android:layout_height="339dp"
        android:ems="80"
        android:hint="Hello Viet Nam"
        android:gravity="start"
        android:inputType="textPersonName" />
    <TextView
        android:id="@+id/textView"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:gravity="right"
        android:text="13/80byte" />
    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="horizontal">
        <Button
            android:id="@+id/btnSend"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_weight="1"
            android:text="Send" />
        <Button
            android:id="@+id/btnClose"

```

```

        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="Close" />
    </LinearLayout>
</LinearLayout>

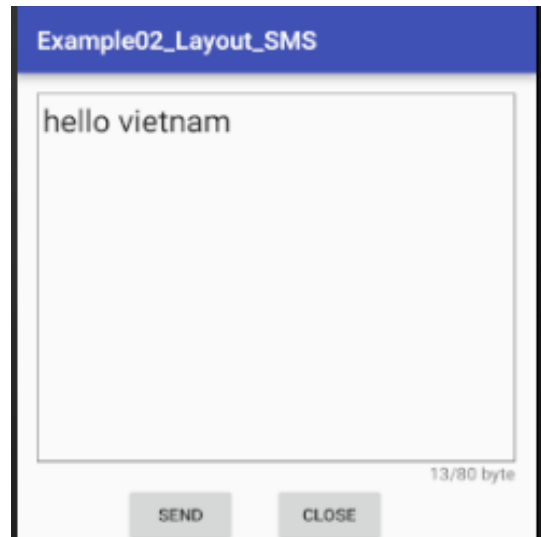
```

- File MainActivity:

```

public class MainActivity extends
AppCompatActivity {
    EditText editText;
    TextView textView;
    Button send, close;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        editText = (EditText)findViewById(R.id.textview1);
        textView = (TextView)findViewById(R.id.textView2);
        send = (Button)findViewById(R.id.send);
        close= (Button)findViewById(R.id.close);
        TextWatcher textWatcher = new TextWatcher(){
            @Override
            public void beforeTextChanged(CharSequence charSequence, int i, int i1, int i2) {
            }
            @Override
            public void onTextChanged(CharSequence charSequence, int i, int i1, int i2) {
            }
            @Override
            public void afterTextChanged(Editable editable) {
                String s = editText.toString();
            }
        };
    }
}

```



```

        textView.setText(editText.length()+"/80byte");
    }
};
editText.addTextChangedListener(textWatcher);
send.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        String inputText = editText.getText().toString();
        Toast.makeText(getApplicationContext(),inputText,Toast.LENGTH_SHORT).show();
    }
});
close.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        finish();
    }
});

```

Bài 3: Tạo giao diện theo mẫu: Nhấn button “Kiểm tra chuỗi”: kiểm tra dữ liệu gõ vào edittext phía dưới, nếu đã có dữ liệu thì hiển thị “Bạn đã gõ đúng”, nếu chưa có dữ liệu thì hiển thị “Chưa có dữ liệu, mời nhập lại”.

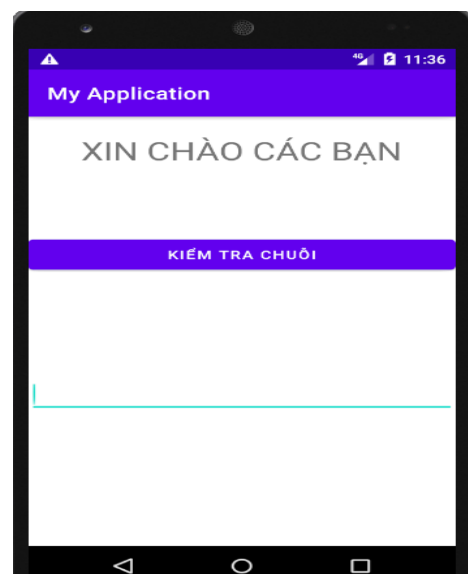
Gợi ý:

- File MainActivity

```

public class MainActivity extends
AppCompatActivity {
    Button btnOK;
    EditText edHoTen;
    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

```



```

setContentView(R.layout.activity_main);
edHoTen = findViewById(R.id.edHoTen);
btnOK = findViewById(R.id.btnOK);
btnOK.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        kiemTraView();
    }
});
void kiemTraView(){
    if(edHoTen.getText().toString().equals("")) {
        Toast.makeText(MainActivity.this , "Chưa có dữ liệu, mời nhập lại",
Toast.LENGTH_SHORT).show();
    }
    else {
        Toast.makeText(MainActivity.this, " Bạn đã gõ đúng ",
Toast.LENGTH_SHORT).show();
    }
}

```

Bài 4: Tạo giao diện form đăng nhập gồm EditText User và Password. Button Login: Nếu tài khoản đúng là “Admin”, pass là ”123” thì thông báo “Đã đăng nhập”, nếu sai thì thông báo “Đăng nhập lại”. Button Clear: Xóa dữ liệu. Button Finish: Thoát khỏi giao diện.

Gợi ý:

- File MainActivity: Xử lý form đăng nhập

```

public class MainActivity extends AppCompatActivity {
    Button btnLogin, btnClear, btnExit;
    EditText edUser, edPass;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnlogin = (Button) findViewById(R.id. btnLogin);

```

```

        btnClear = (Button) findViewById(R.id. btnClear);
        btnExit = (Button) findViewById(R.id. btnExit);
        edUser = (EditText) findViewById(R.id. edUser);
        edPass = (EditText) findViewById(R.id. edPass);
btnLogin.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if(edUser.getText().toString().equals("Admin") &&
            edPass.getText().toString().equals("123")) {
            Toast.makeText(getApplicationContext(),
                "Password đúng, đang đăng nhập...",Toast.LENGTH_SHORT).show();
        }
        else{
            Toast.makeText(getApplicationContext(), "Password sai, vui lòng nhập lại",
                Toast.LENGTH_SHORT).show();
        }
    }
});
btnClear.setOnClickListener(new View.OnClickListener(){
    @Override
    public void onClick(View v){
        edUser.setText("");
        edPass.setText("");
    }
});
btnExit.setOnClickListener(new View.OnClickListener(){
    public void onClick(View v) {
        finish();
    }
}); } }

```

Bài 5. Tạo listview và giao diện theo mẫu. Button Add: Thêm dữ liệu từ các edit Name và Population vào Listview. Button Delete: Xóa dòng dữ liệu trên ListView. Button Edit: sửa dòng dữ liệu trên ListView.

Gợi ý:

- File MainActivity

```
public class MainActivity extends
```

```
AppCompatActivity {
```

```
    EditText edName, edPopulation;
```

```
    ListView listView;
```

```
    Adapter adapter;
```

```
    ArrayList<dongDulieu> arrayList;
```

```
    Button btnAdd, btnDelete, btnEdit;
```

```
    int row;
```

```
    @Override
```

```
    protected void onCreate(Bundle
```

```
savedInstanceState) {
```

```
        super.onCreate(savedInstanceState);
```

```
        setContentView(R.layout.activity_main);
```

```
        edName = findViewById(R.id.edName);
```

```
        edPopulation = findViewById(R.id.edPopulation);
```

```
        listView = findViewById(R.id.listview);
```

```
//Tạo dữ liệu trong listview
```

```
        arrayList = new ArrayList<>();
```

```
        arrayList.add(new dongDulieu(R.mipmap.aaa,"Viet Nam", "200"));
```

```
        arrayList.add(new dongDulieu(R.mipmap.images,"English", "100"));
```

```
        arrayList.add(new dongDulieu(R.mipmap.aaa,"france", "300"));
```

```
        adapter = new Adapter(MainActivity.this, R.layout.dongdulieu,arrayList);
```

```
        listView.setAdapter(adapter);
```

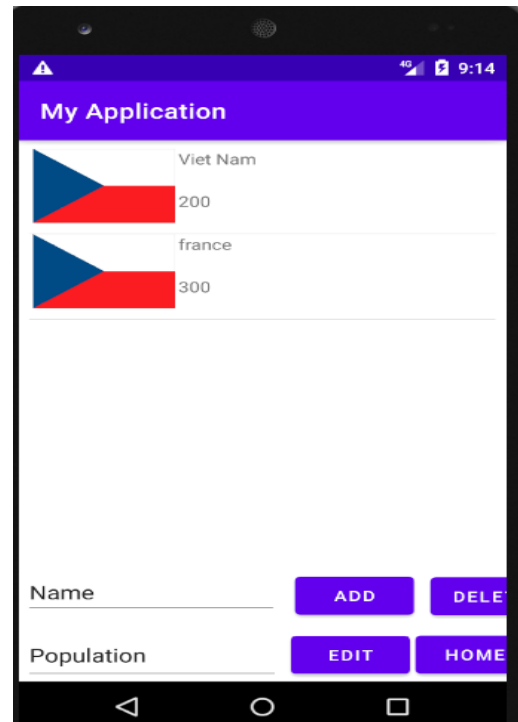
```
//sự kiện kích một phần tử danh sách trong listview
```

```
listView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
```

```
    @Override
```

```
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
```

```
        row = i;
```



```

Toast.makeText(MainActivity.this, "Vị trí: " + i+ " , nước : " + ((dongDulieu)
    adapter.getItem(i)).name + " , dân số: " + ((dongDulieu)
    adapter.getItem(i)).population, Toast.LENGTH_LONG).show();
edName.setText(((dongDulieu) adapter.getItem(i)).name);
edPopulation.setText(((dongDulieu) adapter.getItem(i)).population);
}
});
//thêm dòng dữ liệu vào listview
btnAdd = findViewById(R.id.btnADD);
btnAdd.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        arrayList.add(new dongDulieu(R.mipmap.aaa,"france","400"));
        adapter.notifyDataSetChanged();
    }
});
//sự kiện nhấn giữ lâu sẽ xuất hiện một hộp thoại hỏi có xóa sản phẩm này không
listView.setOnItemLongClickListener(new AdapterView.OnItemLongClickListener()
{
    @Override
    public boolean onItemLongClick(AdapterView<?> adapterView, View view, int i,
long l) {
        AlertDialog.Builder alertDialogBuilder = new
            AlertDialog.Builder(MainActivity.this);
        alertDialogBuilder.setMessage("Bạn có muốn xóa sản phẩm này!");
        alertDialogBuilder.setPositiveButton("Có", new
            DialogInterface.OnClickListener() {
                @Override
                public void onClick(DialogInterface dialog, int which) {
                    // xóa sản phẩm đang nhấn giữ

```

```

        arrayList.remove(i);
        //cập nhật lại listview
        adapter.notifyDataSetChanged();
    }    });
    alertDialogBuilder.setNegativeButton("Không", new
        DialogInterface.OnClickListener() {
        @Override
        public void onClick(DialogInterface dialog, int which) {
        }    });
    alertDialogBuilder.show();
    return true;
}    });
//sự kiện nhấn nút xóa dòng trong listview
btnDelete = findViewById(R.id.btnDelete);
btnDelete.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        if(!arrayList.isEmpty() && (arrayList != null)) {
            arrayList.remove(row);
            adapter.notifyDataSetChanged();
        }    else {
            Toast.makeText(MainActivity.this, "Listview trống",
                Toast.LENGTH_SHORT).show();
        }
    }    });
//sự kiện nhấn nút sửa trong listview
btnEdit = findViewById(R.id.btnEdit);
btnEdit.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

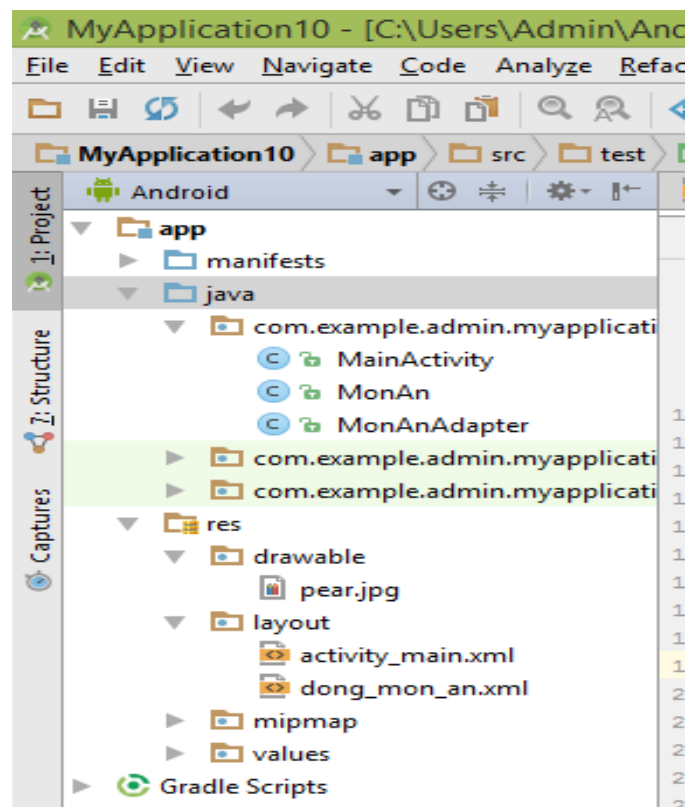
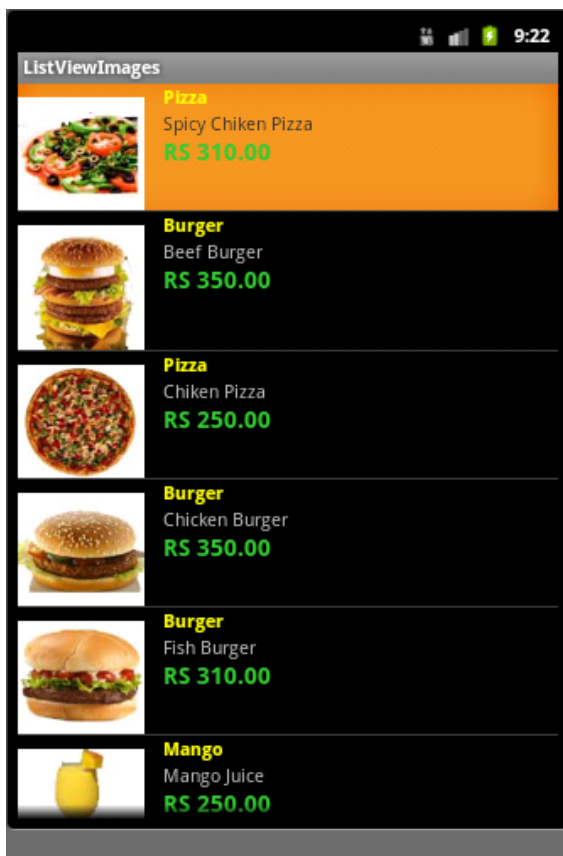
```

```

String name = edName.getText().toString();
String population = edPopulation.getText().toString();
arrayList.set(row, new dongDulieu(R.mipmap.aaa, name, population));
adapter.notifyDataSetChanged();
}
}); } }

```

Bài 6: Tạo ứng dụng có listView hiển thị thông tin của các món ăn gồm Tên, Giá và Hình ảnh theo mẫu.



Gợi ý:

- File MonAn.java:

```

public class MonAn {
    public String Ten;
    public Integer Gia;
    public Integer HinhAnh;
    public MonAn(String ten, Integer gia, Integer hinhAnh) {

```

```

    Ten = ten;
    Gia = gia;
    HinhAnh = hinhAnh;
} }

```

- File MonanAdapter.java

```

public class MonAnAdapter extends BaseAdapter {
    Context context;
    int mylayout;
    List<MonAn> monAnList;
    public MonAnAdapter(Context context, int mylayout, List<MonAn> monAnList) {
        this.context = context;
        this.mylayout = mylayout;
        this.monAnList = monAnList;
    }
    @Override
    public int getCount() {
        return monAnList.size(); }
    @Override
    public Object getItem(int position) {
        return null; }
    @Override
    public long getItemId(int position) {
        return 0; }
    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        LayoutInflater inflater = (LayoutInflater)
            context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        convertView = inflater.inflate(mylayout,null);
        TextView txtTen = (TextView) convertView.findViewById(R.id.txtTen);
        txtTen.setText(monAnList.get(position).Ten);
    }
}

```

```

TextView txtGia = (TextView) convertView.findViewById(R.id.txtGia);
txtGia.setText(monAnList.get(position).Gia + " ");
ImageView imgHinh =(ImageView) convertView.findViewById(R.id.imgHinh);
imgHinh.setImageResource(monAnList.get(position).HinhAnh);
return convertView;
}}

```

- File MainActivity.java

```

public class MainActivity extends AppCompatActivity {
    ListView lvMonAn;
    ArrayList<MonAn> mangMonAn;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        lvMonAn = (ListView) findViewById(R.id.lvMonAn);
        mangMonAn = new ArrayList<MonAn>();
        mangMonAn.add(new MonAn("CHao", 100, R.drawable.pear));
        mangMonAn.add(new MonAn("CHao", 100, R.drawable.pear));
        mangMonAn.add(new MonAn("CHao", 100, R.drawable.pear));
        MonAnAdapter anAdapter = new MonAnAdapter(
            MainActivity.this, R.layout.dong_mon_an, mangMonAn );
        lvMonAn.setAdapter(anAdapter); } }

```

Bài 7. Tạo ứng dụng dùng camera đọc mã QR Code gồm 1 Button Scan, 2 EditText hiển thị thông tin tên và địa chỉ tương ứng với mã QR sau khi bấm nút Scan .

3. TÓM TẮT TRÌNH TỰ THỰC HIỆN HOẶC QUY TRÌNH CÔNG NGHỆ

TT	Tên các bước công việc	Dụng cụ. thiết bị, vật tư	Yêu cầu kỹ thuật	Các chú ý về an toàn lao động

1	Khởi động Android Studio	Phần mềm Android Studio Máy tính.	Cài đặt chính xác phần mềm.	Đảm bảo an toàn cho người và thiết bị.
2	Thực hành thiết kế và code cho các nút lệnh	Phần mềm Android Studio Máy tính.	Cài đặt chính xác phần mềm.	Đảm bảo an toàn cho người và thiết bị.
3	Kiểm tra kết quả	Phần mềm Android Studio Máy tính.	Cài đặt chính xác phần mềm.	Đảm bảo an toàn cho người và thiết bị.

BÀI 5: TRUYỀN THÔNG GIỮA CÁC ACTIVITY

1. MỤC TIÊU

Trình bày được khái niệm Activity, Intent. Sử dụng được Intent truyền dữ liệu giữa các Activity.

Nghiêm túc, tự giác trong học tập. Đảm bảo an toàn cho người và thiết bị.

2. NỘI DUNG

2.1. Lý thuyết liên quan

2.1.1. Activity

Activity là một trong 4 thành phần chính của một ứng dụng Android. Activity được dùng để hiển thị một màn hình.

2.1.2. Intent và việc tương tác giữa các Activity

Mục đích của Intent: bắt đầu 1 activity, bắt đầu một dịch vụ khác và truyền dữ liệu giữa các activity

Có 2 loại Intent:

- 1) Dùng **Intent** để mở một Activity khác nhưng không kiểm soát kết quả trả về.
- 2) Dùng **Intent** để mở một Activity khác và có kiểm soát kết quả trả về.

2.2. Trình tự thao tác

2.2.1. Tạo Activity

- Tạo activity mới: Kích chuột phải vào **java/ New / Activity / EmptyActivity** / Đặt tên / **Finish**.

- Khởi động Activity tên là **TargetActivity**

```
Intent intent = new Intent(getApplicationContext(), TargetActivity.class);
startActivity(intent);
```

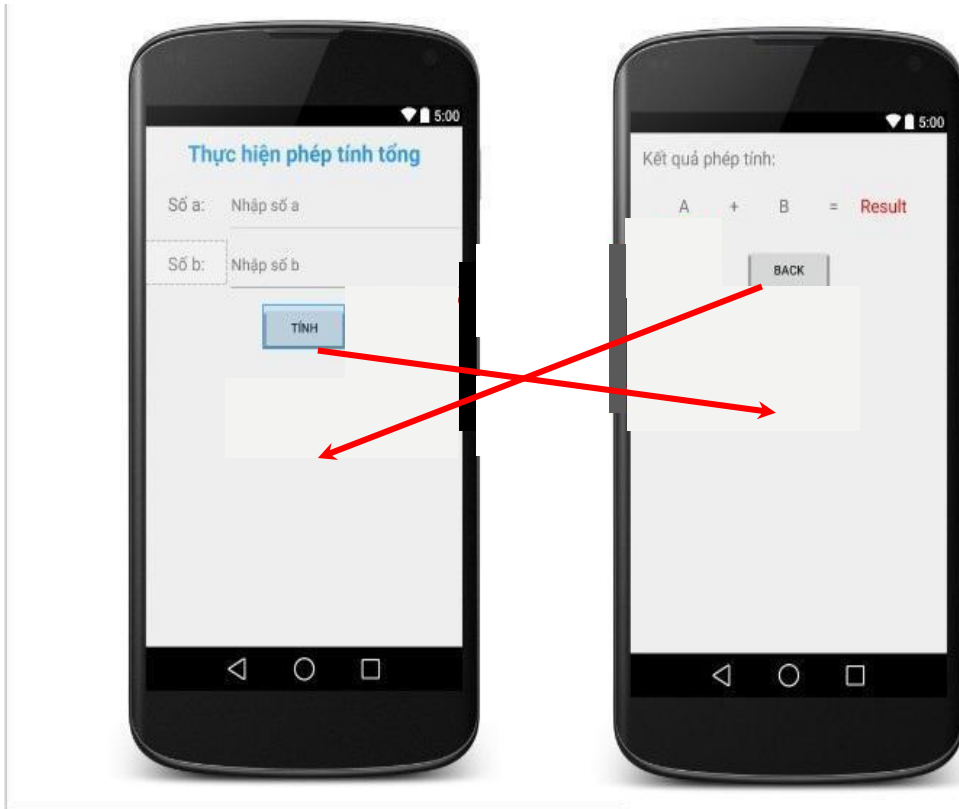
- Liên lạc giữa 2 activity: Khi khởi động một activity, câu lệnh gửi kèm dữ liệu trong intent như ví dụ sau:

```
intent.putExtra("value1", new String("Hello"));
intent.putExtra("value2", new Long(100));
```

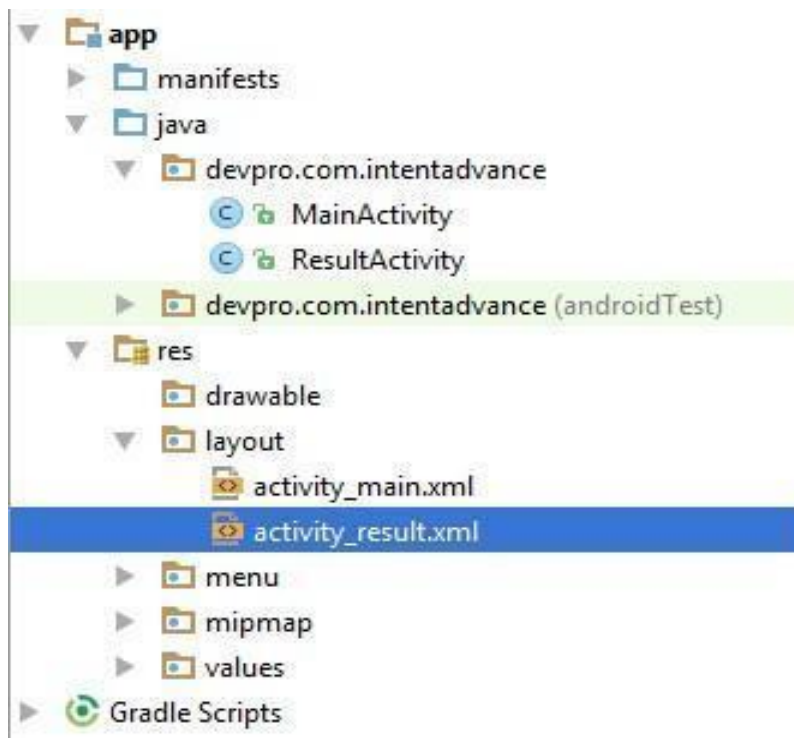
Bên phía activity được khởi động, lấy dữ liệu đã gửi như sau:

```
getIntent().getExtras().getString("value1");
getIntent().getExtras().getLong("value2");
```

2.2.2. Sử dụng Intent để mở một Activity khác nhưng không kiểm soát kết quả trả về



Bước 1. Tạo project mới và activity mới với cấu trúc như sau.



Bước 2. Xây dựng file **activity_main** có giao diện như sau:



- File MainActivity

```
public class MainActivity extends AppCompatActivity {  
    EditText editNumberA, editNumberB;  
    Button btnCalculate;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
        editNumberA = findViewById(R.id. editNumberA);  
        editNumberB = findViewById(R.id. editNumberB);  
        btnCalculate = findViewById(R.id. btnCalculate);  
    }  
}
```

```
btnCalculate.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        Intent intent= new Intent(MainActivity.this,ResultActivity.class);  
        //Khai báo bundle  
        Bundle bundle= new Bundle();  
        int a = Integer.parseInt(editNumberA.getText().toString());  
        int b = Integer.parseInt(editNumberB.getText().toString());  
        //đưa dữ liệu riêng lẻ vào Bundle  
        bundle.putInt("NUMBERA",a);  
        bundle.putInt("NUMBERB",b);  
        //Đưa Bundle vào Intent  
        intent.putExtra("CACULATION",bundle);  
        //Mở activity ResultActivity  
        startActivity(intent);  
    }  
}
```

```
});
}
```

Bước 3. File `activity_result`

Bước 4. File `ResultActivity`

```
public class ResultActivity extends Activity {TextView txtNumberA,
txtNumberB, txtResult;
    Button btnBack;
```

```
    @Override
    protected void onCreate(Bundle
savedInstanceState)
    { super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_result);
    txtNumberA = (TextView)
findViewById(R.id.txtNumberA);
    txtNumberB = (TextView)
findViewById(R.id.txtNumberB);
    txtResult = (TextView)
findViewById(R.id.txtResult);
    btnBack = (Button)
findViewById(R.id.btnBack);
    //lấy intent gọi Activity này
    Intent callerIntent = getIntent();
    //có intent rồi thì lấy Bundle dựa vào key
    Calculation
    Bundle packageFromCaller=
callerIntent.getBundleExtra( "Calculation");
    //Có Bundle rồi thì lấy các thông số dựa
vào key NUMBERA và NUMBERB
    int a =
packageFromCaller.getInt("NUMBERA");
```



```

int b = packageFromCaller.getInt(" NUMBERB");
int result = a +b ;
txtNumberA.setText(""+a);
txtNumberB.setText(+b);
txtResult.setText(""+ (a+b));

btnBack.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick (View view){
        finish();
    }
});
}
}
}

```

Bên ResultActivity chúng ta tiến hành lấy bundle dựa vào key từ bên MainActivity truyền sang, sau đó từ bundle chúng ta lấy ra 2 số a và b, Thực hiện tính tổng và gán vào các textview

2.2.3. Lấy kết quả trả về từ Activity thông qua Intent

- File MainActivity

```

public class MainActivity extends AppCompatActivity {
    Button btnA;
    EditText edtA, edtBgui;
    public static final int GUI = 1;
    public static final int NHAN = 2;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnA = (Button) findViewById(R.id.btnA);
        edtA = (EditText) findViewById(R.id.edtA);
        edtBgui = (EditText) findViewById(R.id.edtBgui);
        btnA.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                chuyen();
            }
        });
    }
}

```

```

    }
    });
}
public void chuyen() {
    Intent intent = new Intent(getApplicationContext(), ActivityB.class);
    // đưa các dữ liệu vào Bundle
    Bundle bundle = new Bundle();
    bundle.putString("A", edtA.getText().toString());
    // Đưa Bundle cho GoiTinA của intent
    intent.putExtra("GoiTinA", bundle);
    startActivityForResult(intent, GUI);
}
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (data == null) {
        return;
    }
    if (requestCode == GUI) {
        if(resultCode == NHAN) {
            Bundle bundle = data.getBundleExtra("GoiTinB");
            edtBgui.setText("B gửi trả lại là: " + bundle.getString("B"));
        }
    }
} }

```

- File ActivityB.java

```

public class ActivityB extends AppCompatActivity {
    Button btnB;
    EditText edtB,edtAgui;
    @Override

```

```

protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_b);
    btnB = (Button) findViewById(R.id.btnB);
    edtB = (EditText) findViewById(R.id.edtB);
    edtAgui = (EditText) findViewById(R.id.edtAgui);
    nhantuMainActivity();
    btnB.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            GuiSangMainActivity(edtB.getText().toString(), MainActivity.NHAN);
        }
    });
}

public void nhantuMainActivity(){
    Intent intent = getIntent();
    // lấy Bundle tên là "GoiTinA"
    Bundle bundle = intent.getBundleExtra("GoiTinA");
    // lấy giá trị kiểu nguyên tên là A
    edtAgui.setText("A chuyển sang: "+ bundle.getString("A"));
}

public void GuiSangMainActivity(String giatri, int requestsultcode){
    Intent intent = getIntent();
    Bundle bundle = new Bundle();
    bundle.putString("B",giatri );
    intent.putExtra("GoiTinB",bundle);
    setResult(requestsultcode,intent);
    finish();
}
}

```

BÀI TẬP

Bài 1. Tạo 2 activity theo hướng dẫn, trong MainActivity có 1 Button OK để mở Activity2. Trong Activity2 có 1 Button btnQuayLai để trở về MainActivity.

Bước 1: Trong activity_main: tạo nút btnOK để mở activity2

Bước 2: Tạo Activity2

Bước 3: Mở MainActivity, code cho nút btnOK

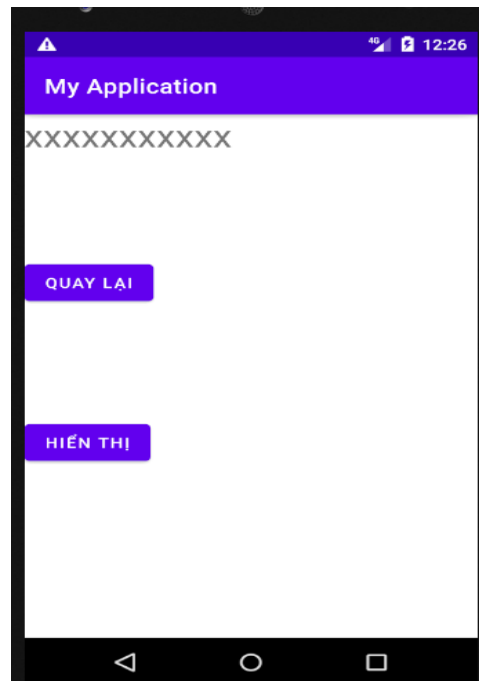
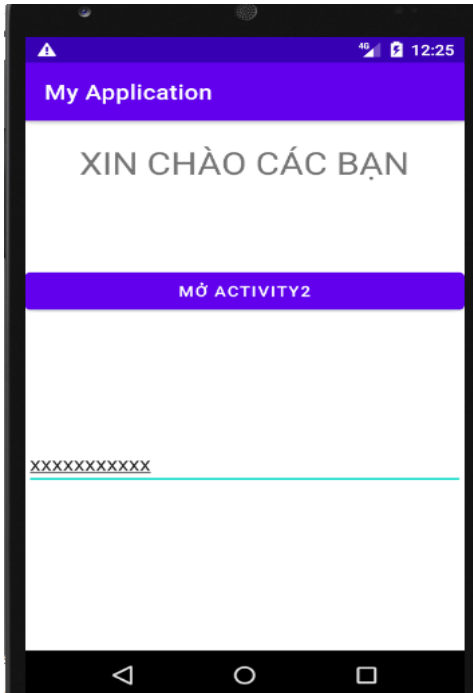
```
btnOK.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        Intent intent = new Intent(MainActivity.this,MainActivity2.class);  
        startActivity(intent);  
    }  
});
```

Bước 4: Trong activity_main2: tạo nút btnQuayLai để đóng activity2

Bước 5: Mở MainActivity2, code cho nút btnQuayLai

```
btnOK.setOnClickListener(new View.OnClickListener() {  
    @Override  
    public void onClick(View view) {  
        finish();  
    }  
});
```

- Bài 2: Tạo 2 activity có giao diện như mẫu. Trong MainActivity1, tạo nút Mở Activity 2. Trong MainActivity2, tạo nút Quay Lại để mở lại MainActivity1. Nút Hiển thị để lấy dữ liệu trên MainActivity2 gửi sang và hiển thị trên MainActivity1.



- File MainActivity2, code cho nút

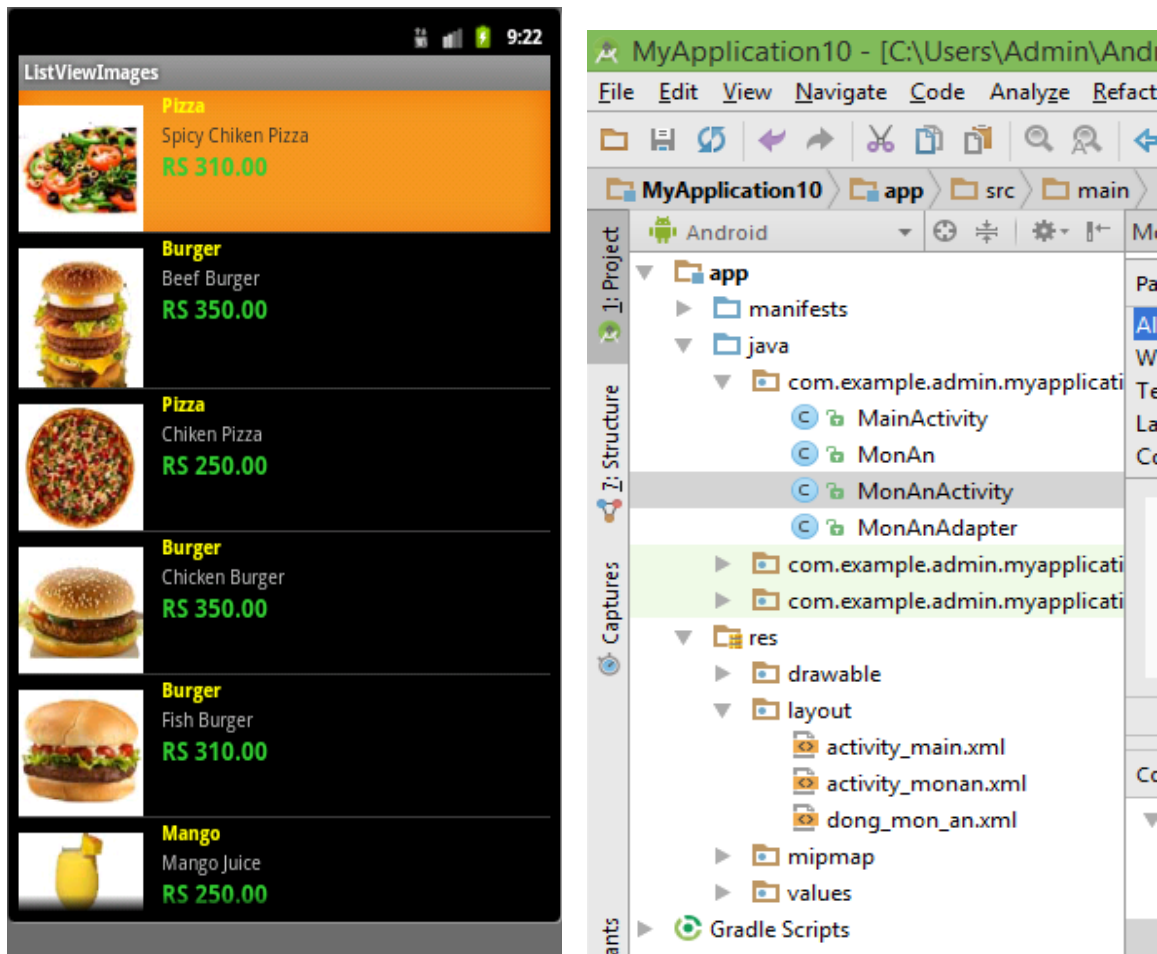
btnQuayLai để mở lại MainActivity1. Nút Hiển thị để lấy dữ liệu trên MainActivity2 gửi sang và hiển thị trên MainActivity1

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main2);
    btnQuayLai = findViewById(R.id.btnQuayLai);
    btnHienThi = findViewById(R.id.btnHienThi);
    textView = findViewById(R.id.textView);
    btnHienThi.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View view) {
            Intent intent= getIntent();
            Bundle bundle= intent.getBundleExtra("MoAc2");
            String test = bundle.getString("TEST");
            textView.setText(test);
        }
    });
    btnQuayLai.setOnClickListener(new View.OnClickListener() {
```

@Override

```
public void onClick(View view) {  
    finish();  
} });
```

Bài 3. Tạo giao diện có ListView Chọn món ăn như mẫu, khi kích chọn listItem trên ListView món ăn thì thông báo bạn đã đặt món.



- File MainActivity

```
public class MainActivity extends AppCompatActivity {  
    ListView lvMonAn;  
    ArrayList<MonAn> mangMonAn;  
    public static final int GUI = 1;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {
```

```

super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
lvMonAn = (ListView) findViewById(R.id.lvMonAn);
mangMonAn = new ArrayList<MonAn>();
mangMonAn.add(new MonAn("CHao", 100, R.drawable.pear));
mangMonAn.add(new MonAn("PHO", 100, R.drawable.pear));
mangMonAn.add(new MonAn("THIT", 100, R.drawable.pear));
MonAnAdapter anAdapter = new MonAnAdapter(
    MainActivity.this,
    R.layout.dong_mon_an,
    mangMonAn    );
lvMonAn.setAdapter(anAdapter);
//click vào từng item thì chuyển sang layout khác
lvMonAn.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> parent, View view, int position, long
id) {
        Intent intent = new Intent(getApplicationContext(), MonAnActivity.class);
        // đưa các dữ liệu vào "thùng chứa" Bundle
        Bundle bundle = new Bundle();
        bundle.putString("A", mangMonAn.get(position).Ten.toString());
        // Đưa thùng chứa Bundle cho người đưa thư Intent, trong đó GoiTin là tên của
        thùng chứa
        intent.putExtra("GoiTinA", bundle);
        startActivityForResult(intent, GUI);
    }    }); }}

```

- File MonanActivity.java

```

public class MonAnActivity extends AppCompatActivity {
    TextView txtMonAn ;
    Button btnBack;

```

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_monan);
    txtMonAn = (TextView)findViewById(R.id.txtTenMonAn);
    btnBack = (Button) findViewById(R.id.btnBack);
    nhantuMainActivity();
    btnBack.setOnClickListener(new View.OnClickListener() {
        @Override
        public void onClick(View v) {
            finish();
        }
    });
}

public void nhantuMainActivity(){
    Intent intent = getIntent();
    // lấy thùng chứa Bundle với tên giao dịch là "GoiTinA"
    Bundle bundle = intent.getBundleExtra("GoiTinA");
    // lấy giá trị kiểu nguyên với tên giao dịch là A
    txtMonAn.setText("A chuyển sang: "+ bundle.getString("A"));
}
}

```

Bài 4. Tạo giao diện đăng nhập. nếu gõ đúng ID và Pass thì mở activity2, hiển thị thông tin ID và Pass. Nếu sai thì mở activity3, hiển thị thông tin ID, Pass và yêu cầu quay lại nhập lại ở activitymain.

- **File MainActivity:**

```

public class MainActivity extends AppCompatActivity {
    Button btnDangNhap;
    EditText edID, edPass;
    String edID1,edPass1;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

```

```

setContentView(R.layout.activity_main);
khaiBaoView();
btnDangNhap.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        dangNhap();
    } });
}
void khaiBaoView(){
    edID = findViewById(R.id.edID);
    edPass = findViewById(R.id.edpass);
    btnDangNhap = findViewById(R.id.btnDangNhap);
}
Intent intent;
void dangNhap(){
    edID1 = edID.getText().toString();
    edPass1 = edPass.getText().toString();
    if(edID1.equals("")) {
        Toast.makeText(getBaseContext(), "Phải Nhập ID",
            Toast.LENGTH_SHORT).show();
        return;
    }
    if(edPass1.equals("")) {
        Toast.makeText(getBaseContext(), "Phải Nhập pass",
            Toast.LENGTH_SHORT).show();
        return;
    }
    if(edID1.equals("a")&&(edPass1.equals("a"))) {
        intent = new Intent(MainActivity.this,MainActivity2.class);
        layDuLieu();
        startActivity(intent);
    }
}

```

```

    }else    {
        intent = new Intent(MainActivity.this,MainActivity3.class);
        layDuLieu();
        startActivity(intent);
    } }
void layDuLieu(){
    Bundle bundle = new Bundle();
    bundle.putString("ID", edID1);
    bundle.putString("Pass", edPass1);
    intent.putExtra("DangNhap", bundle);
}
}

```

- **File MainActivity2:**

```

public class MainActivity2 extends AppCompatActivity {
    Button btnHome;
    EditText editID, editPass;
    TextView txtThongBao;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main2);
        khaiBaoView();
        Intent intent=getIntent();
        Bundle bundle=intent.getBundleExtra("DangNhap");
        editID.setText(bundle.getString("ID"));
        editPass.setText(bundle.getString("Pass"));
        String str ="Chúc mừng bạn đăng nhập đúng tài khoản";
        txtThongBao.setText(str);
        btnHome.setOnClickListener(new View.OnClickListener() {
            @Override

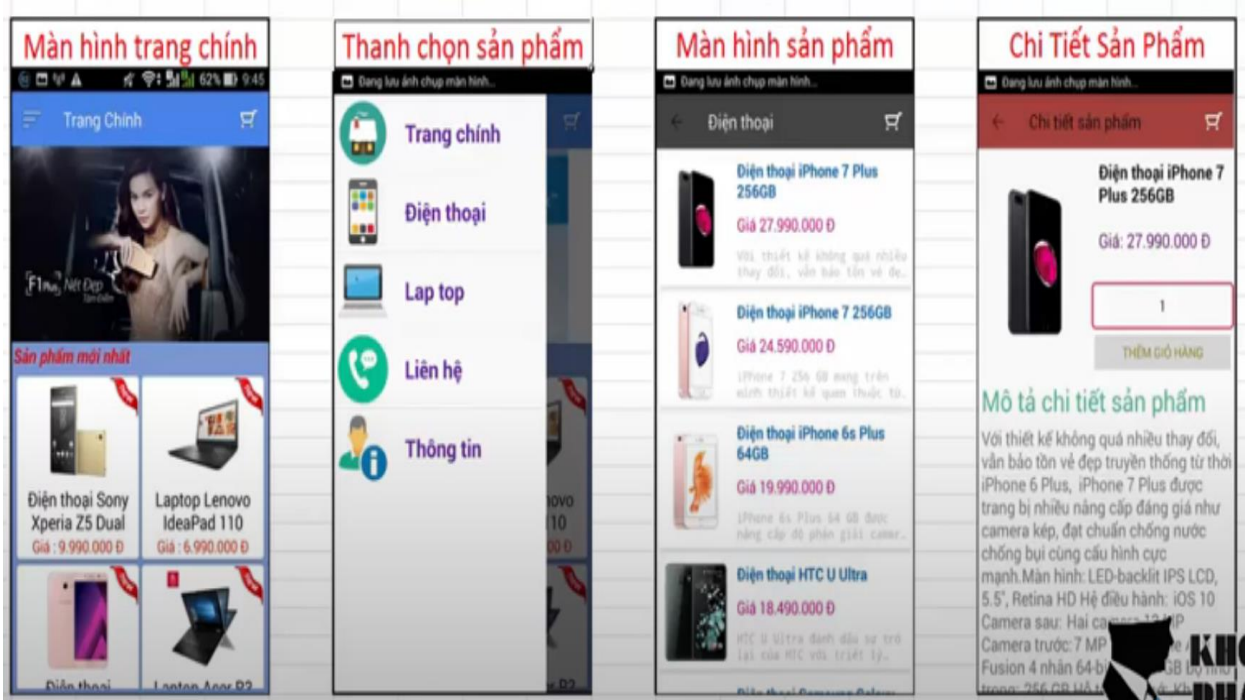
```

```
public void onClick(View view) {
    finish();
} }); }
```

```
void khaiBaoView(){
    editID = findViewById(R.id.editID);
    editPass = findViewById(R.id.editPass);
    txtThongBao = findViewById(R.id.txtThongBao);
    btnHome = findViewById(R.id.btnQuayLai);
}
}}
```

- **File MainActivity3:** Tương tự MainActivity2.

Bài 5: Tạo giao diện theo mẫu sau:



3. TÓM TẮT TRÌNH TỰ THỰC HIỆN HOẶC QUY TRÌNH CÔNG NGHỆ

TT	Tên các bước công việc	Dụng cụ, thiết bị, vật tư	Yêu cầu kỹ thuật	Các chú ý về an toàn lao động

1	Khởi động Android Studio	Phần mềm Android Studio Máy tính.	Cài đặt chính xác phần mềm.	Đảm bảo an toàn cho người và thiết bị.
2	Thực hành thiết kế và code cho các nút lệnh	Phần mềm Android Studio Máy tính.	Cài đặt chính xác phần mềm.	Đảm bảo an toàn cho người và thiết bị.
3	Kiểm tra kết quả	Phần mềm Android Studio Máy tính.	Cài đặt chính xác phần mềm.	Đảm bảo an toàn cho người và thiết bị.

BÀI 6: SỬ DỤNG MENU

1. MỤC TIÊU

Sử dụng được trình đơn (Menu) và thanh tác vụ (Action Bar). Sử dụng thành thạo Android Studio để xây dựng ứng dụng.

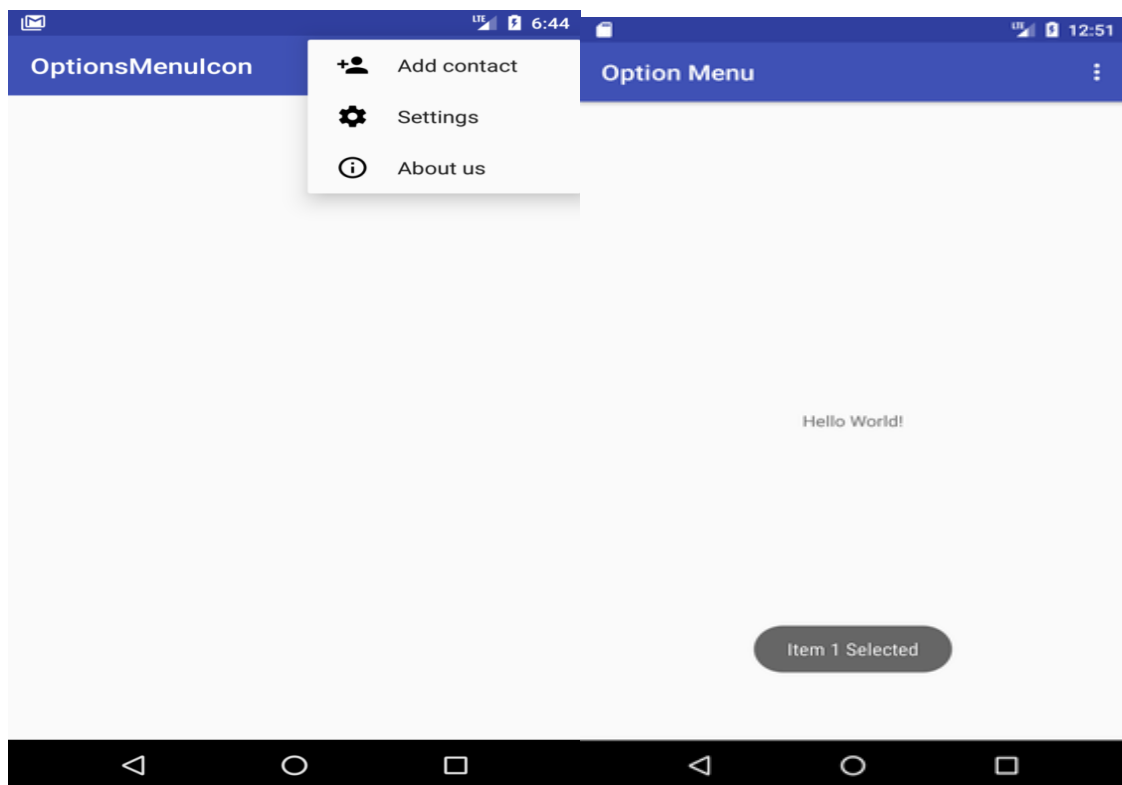
Nghiêm túc, tự giác trong học tập. Đảm bảo an toàn cho người và thiết bị.

2. NỘI DUNG

2.1. Lý thuyết liên quan

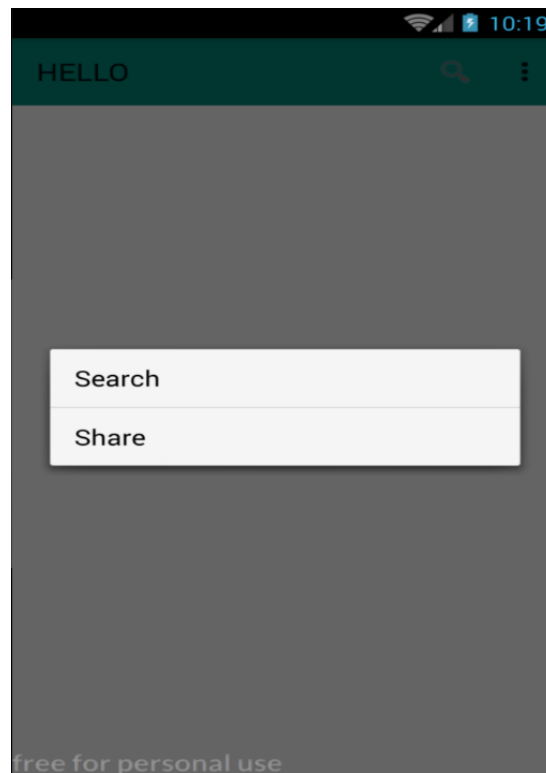
2.1.1 Options menu

Là loại menu nằm trên ActionBar hoặc Toolbar ở phía bên phải. Menu hiển thị dưới dạng text, text+icon hoặc icon “ba chấm”, menu cũng hỗ trợ tùy biến giao diện, có dạng như sau.



2.1.2. Context Menu

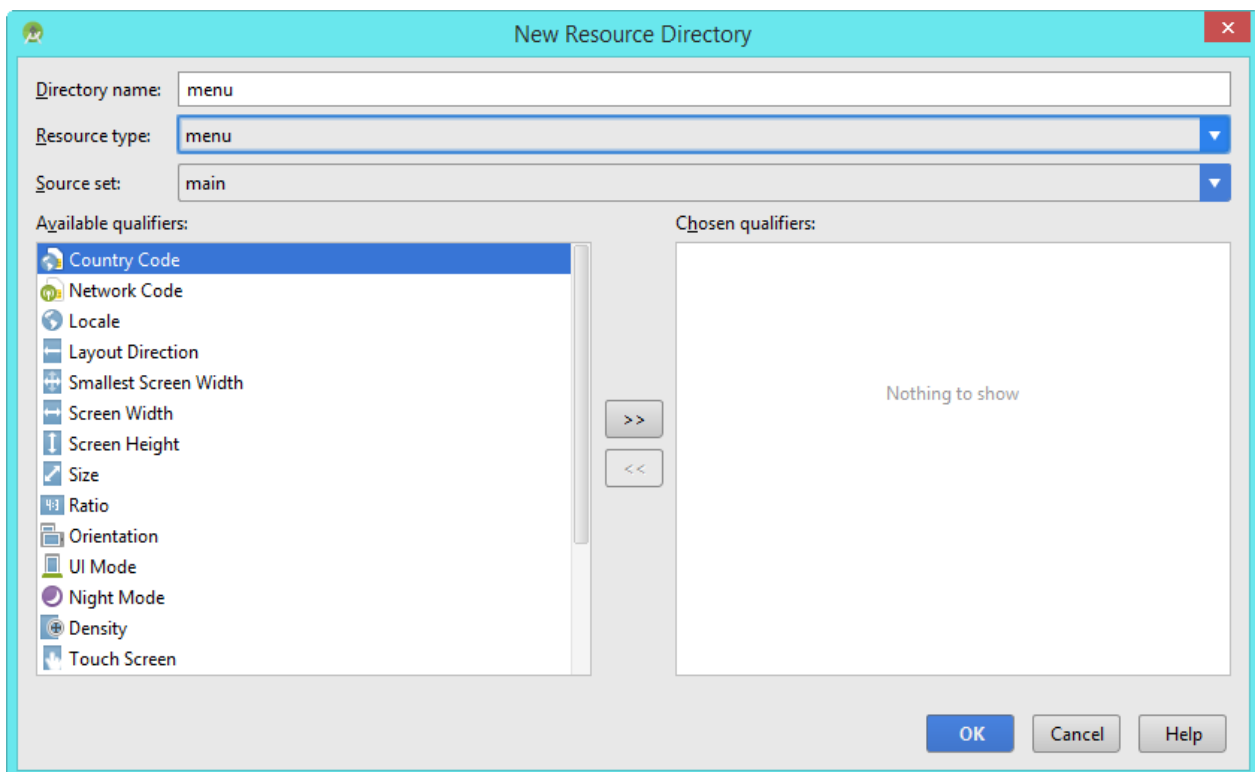
Là menu có dạng như sau

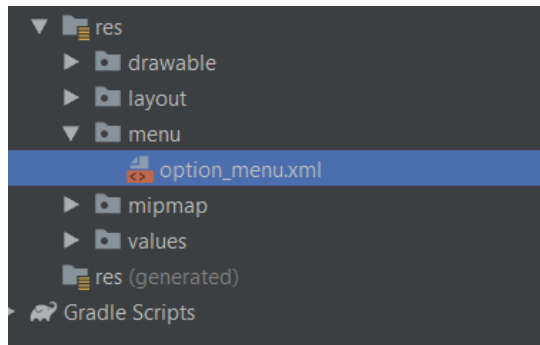


2.1. Trình tự thao tác

2.2.1. Options menu

Tạo menu: click chuột phải vào mục **res** > **New** > **Android Resource Directory** để tạo một thư mục chứa menu (lưu ý: phải đặt chính xác tên thư mục là **menu**) > OK.





Sau khi hoàn thành ta có thư mục menu chứa file **option_menu.xml**

- Để tạo các ra **menu** ta dùng các thẻ <item> trong file **option_menu.xml**

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

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

```
  xmlns:app="http://schemas.android.com/apk/res-auto">
```

```
  <item android:id="@+id/item1"
```

```
    android:title="Item1"
```

```
    android:icon="@drawable/ic_search_black_24dp"
```

```
    android:orderInCategory="100"
```

```
    app:showAsAction="ifRoom|collapseActionView"
```

```
    app:actionViewClass="android.widget.SearchView" />
```

```
  <item android:id="@+id/item2"
```

```
    android:title="Item2"
```

```
    android:icon="@drawable/ic_people_black_24dp"
```

```
    android:orderInCategory="100"
```

```
    app:showAsAction="withText" />
```

```
  <item android:id="@+id/item3"
```

```
    android:title="Item3"
```

```
    android:icon="@drawable/ic_settings_black_24dp"
```

```
    android:orderInCategory="100"
```

```
    app:showAsAction="withText" />
```

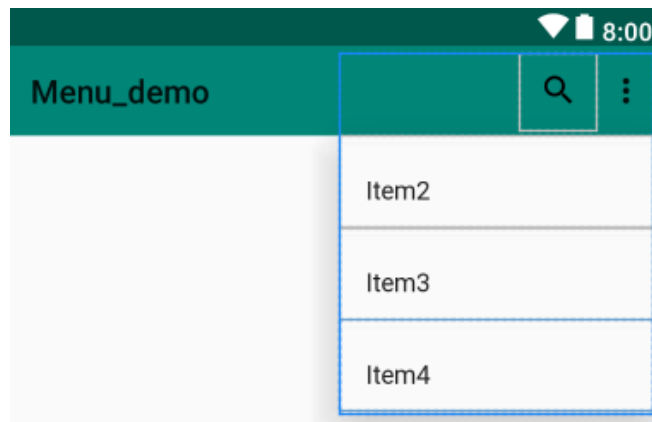
```
  <item android:id="@+id/item4"
```

```
    android:title="Item4"
```

```
    android:icon="@drawable/ic_photo_camera_black_24dp"
```

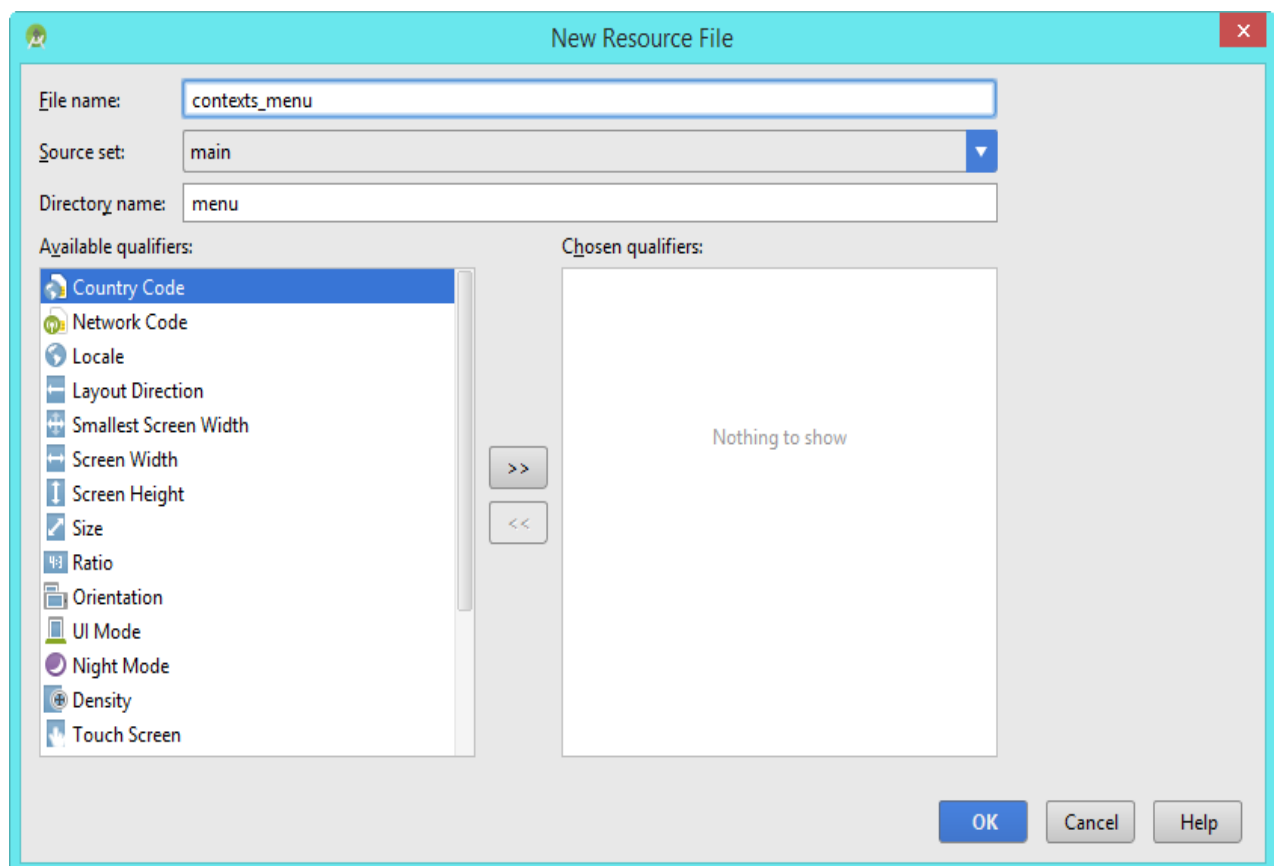
```
android:orderInCategory="100"  
app:showAsAction="withText" />
```

```
</menu>
```



2.2.2. Context Menu

Bước 1: Tạo context menu: Chuột phải menu -> chọn New -> chọn menu resource file -> Nhập tên là contexts_menu.xml -> OK



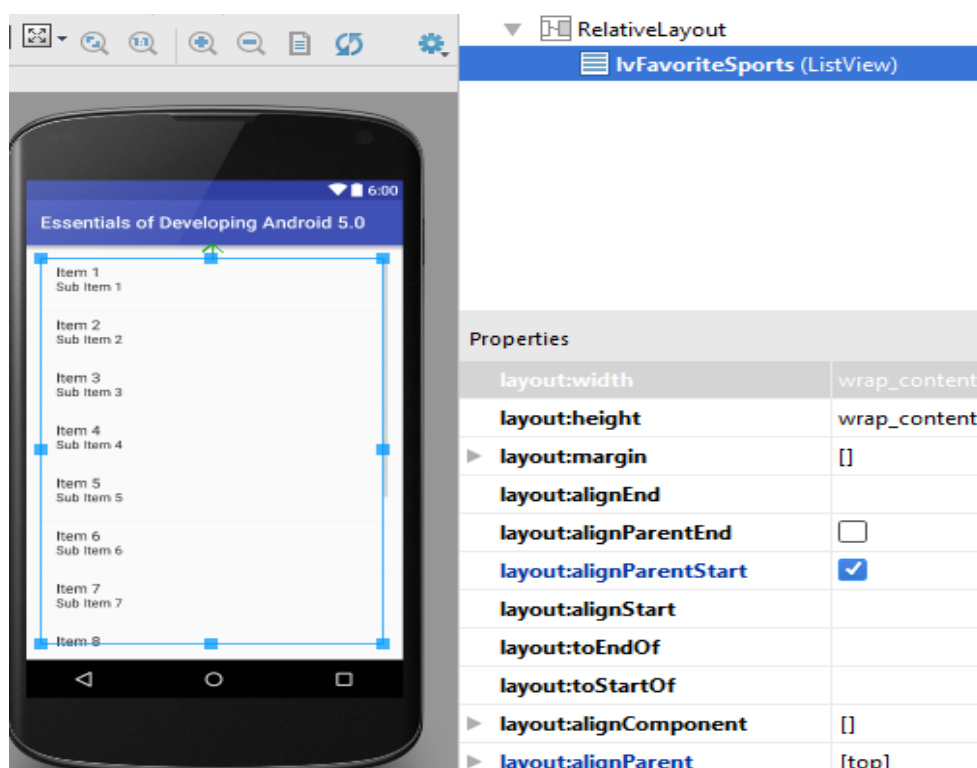
- **File contexts_menu.xml**

```

<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item
    android:id="@+id/edit"
    android:title="Edit"/>
  <item
    android:id="@+id/delete"
    android:title="Delete"/>
</menu>

```

Bước 2: Thiết kế giao diện cho MainActivity



Bước 3: Đăng ký layout cho option menu

Nhấn phím Alt và phím Insert -> chọn Override Method -> chọn onCreateOptionsMenu

```

public boolean onCreateOptionsMenu(Menu menu) {
    getMenuInflater().inflate(R.menu.options_menu, menu);
    return super.onCreateOptionsMenu(menu);
}

```

Bước 4: Xử lý sự kiện cho option menu

Nhấn phím Alt và phím Insert -> chọn Override Method -> chọn onOptionsItemSelected

```
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    switch (id) {
        case R.id.add:
            // Gọi màn hình AdditionSportActivity
        case R.id.exit:
    }
    return super.onOptionsItemSelected(item);
}
```

Bước 5: Đăng ký layout cho context menu

Nhấn phím Alt và phím Insert -> chọn Override Method -> chọn onCreateContextMenu

```
public void onCreateContextMenu(ContextMenu menu, View v,
ContextMenu.ContextMenuInfo menuInfo) {
    super.onCreateContextMenu(menu, v, menuInfo);
    getMenuInflater().inflate(R.menu.contexts_menu, menu);
}
```

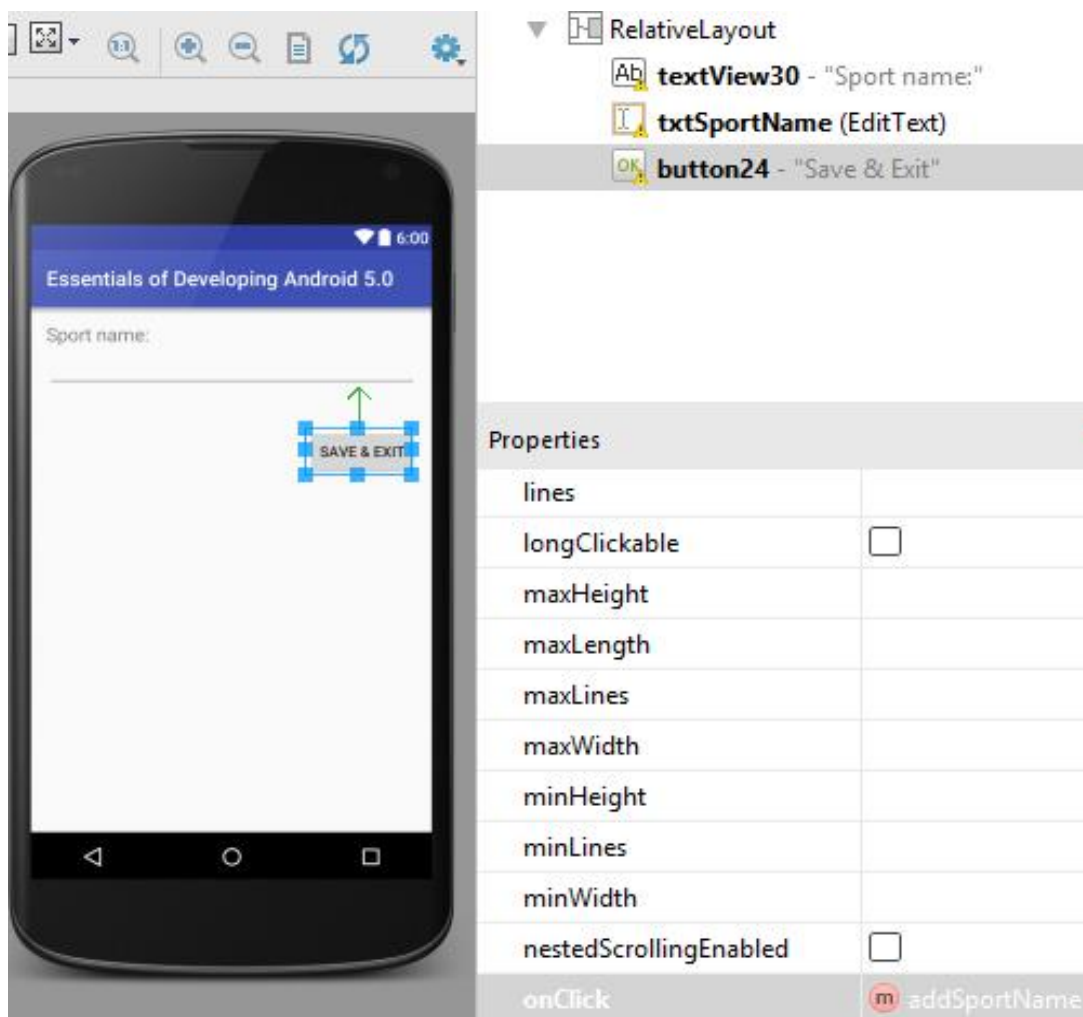
Bước 6: Xử lý sự kiện cho context menu

Nhấn phím Alt và phím Insert -> chọn Override Method -> chọn onOptionsItemSelected

```
public boolean onOptionsItemSelected(MenuItem item) {
    int id = item.getItemId();
    switch (id) {
        case R.id.edit:
            // Gọi màn hình AdditionSportActivity
            break;
        case R.id.delete:
            break;
    }
    return super.onOptionsItemSelected(item);
}
```

Bước 6: Tạo Activity tên AdditionSportActivity

Thiết kế giao diện



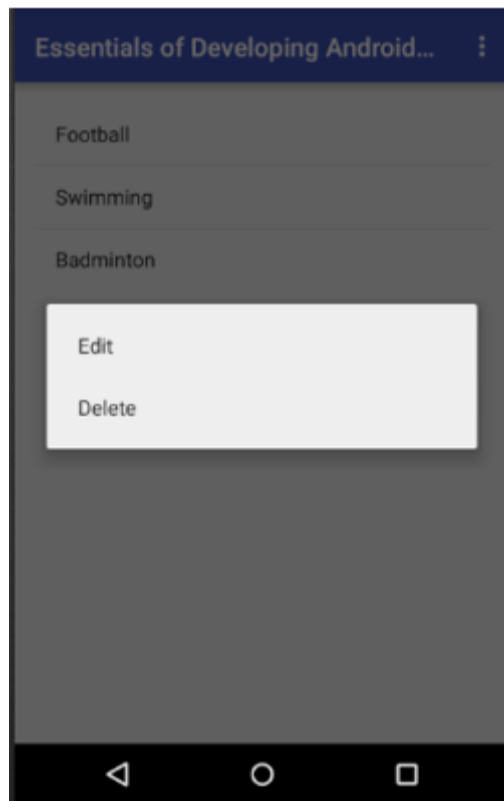
Viết xử lý cho nút “Save & Exit” và “Save”

```
public void addSportName(View v) {  
    Intent i = getIntent();  
}
```

Bước 7: Build và run

BÀI TẬP

Bài 1. Tạo contextmenu cho listview theo mẫu



Chú ý: Nếu tạo contextmenu cho listview, nhấn giữ sẽ hiển thị menu thì phải đăng kí ngay sau lệnh tạo listview:

Ví dụ:

```
theLoaiAdapterListview = new theLoaiAdapterListview(this, listTheLoai);
listView.setAdapter(theLoaiAdapterListview);
registerForContextMenu(listView);
```

3. TÓM TẮT TRÌNH TỰ THỰC HIỆN HOẶC QUY TRÌNH CÔNG NGHỆ

TT	Tên các bước công việc	Dụng cụ, thiết bị, vật tư	Yêu cầu kỹ thuật	Các chú ý về an toàn lao động
1	Khởi động Android Studio	Phần mềm Android Studio Máy tính.	Cài đặt chính xác phần mềm.	Đảm bảo an toàn cho người và thiết bị.

2	Thực hành thiết kế và code cho các menu lệnh	Phần mềm Android Studio Máy tính.	Cài đặt chính xác phần mềm.	Đảm bảo an toàn cho người và thiết bị.
3	Kiểm tra kết quả	Phần mềm Android Studio Máy tính.	Cài đặt chính xác phần mềm.	Đảm bảo an toàn cho người và thiết bị.

BÀI 7: LƯU TRỮ DỮ LIỆU

1. MỤC TIÊU

Trình bày được các khái niệm lưu trữ dữ liệu trong Android. Sử dụng được các cách lưu trữ dữ liệu để tạo ứng dụng

Nghiêm túc, tự giác trong học tập. Đảm bảo an toàn cho người và thiết bị.

2. NỘI DUNG

2.1. Lý thuyết liên quan

Hệ điều hành Android có hỗ trợ khá khác các phương thức lưu trữ dữ liệu. Việc bạn chọn phương thức nào tùy thuộc vào nhu cầu và tính chất của nhu cầu đó, cũng như dung lượng cần lưu trữ là bao nhiêu.

Các phương thức nói trên bao gồm:

Internal file storage: Lưu trữ các file riêng tư của ứng dụng (app-private files) trong hệ thống file của thiết bị.

External file storage: Lưu trữ file trên hệ thống file chia sẻ ra được bên ngoài. Thường dùng cho các file mà người dùng được chia sẻ, chẳng hạn như hình ảnh, video.

Shared Preferences: Lưu trữ dữ liệu nguyên thủy bởi các cặp key-value.

Database: Lưu trữ các dữ liệu có cấu trúc trong cơ sở dữ liệu private.

Trong giới hạn giáo trình này, chúng ta sẽ đi sâu vào lưu trữ dữ liệu có cấu trúc bằng hệ quản trị SQLite. SQLite là hệ quản trị cơ sở dữ liệu quan hệ giống các hệ khác như SQL Server, MySQL, Oracle, v.v. Sử dụng SQLite, ta có thể tạo độc lập, cơ sở dữ liệu quan hệ cho mỗi ứng dụng. Sử dụng chúng để lưu trữ và quản lý cấu trúc dữ liệu phức tạp của ứng dụng. Tất cả cơ sở dữ liệu của Android được lưu trữ trong thư mục /data/data/<tên package>/databases trong thiết bị hoặc máy ảo. Mặc định tất cả cơ sở dữ liệu là riêng tư chỉ có ứng dụng tạo ra chúng mới có quyền truy cập. Tuy nhiên, để chia sẻ cơ sở dữ liệu của ứng dụng ra ngoài chúng ta có thể dùng Content Providers.

[Phân biệt 2 cách lưu trữ CSDL trên SQLite và Firebase ??????](#)

Cơ sở dữ liệu SQLite có thể được truy cập bằng cách dùng ngôn ngữ truy vấn SQL (Structured Query Language).

2.2. Trình tự thao tác

2.2.1. Tải phần mềm DB Browse để tạo CSDL

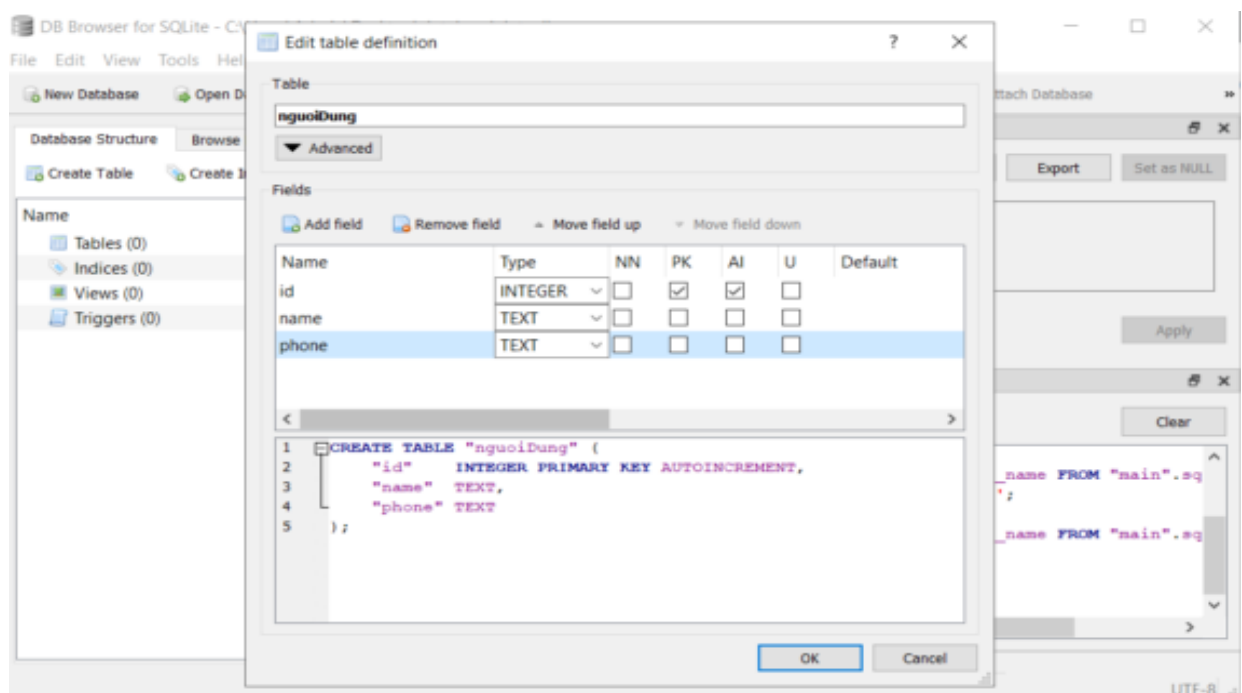
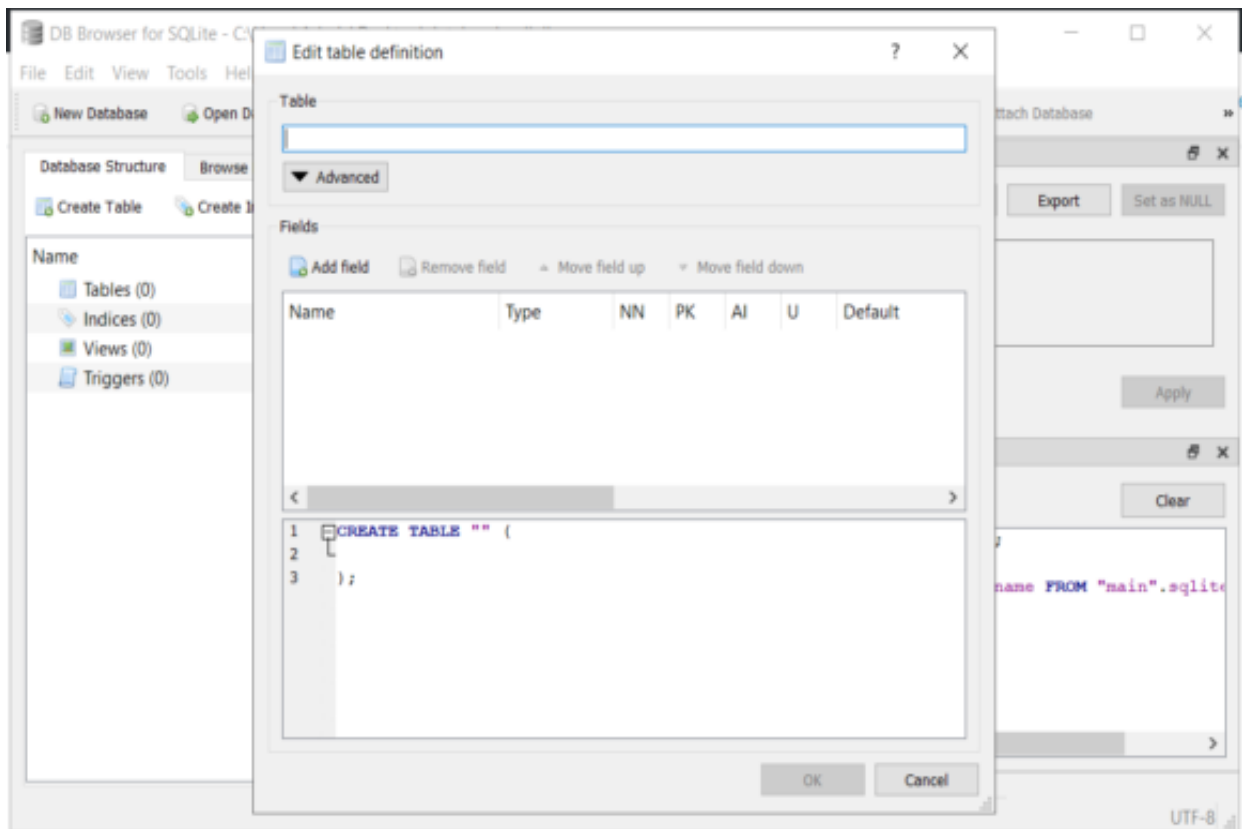
Link tải: <https://sqlitebrowser.org/dl/>

Ví dụ: Tạo CSDL dethi.db có bảng TheLoai(idTheLoai, TenTheLoai).

Tên cột	Kiểu dữ liệu	Ràng buộc
idTheLoai	int	Primary Key
TenTheLoai	text	

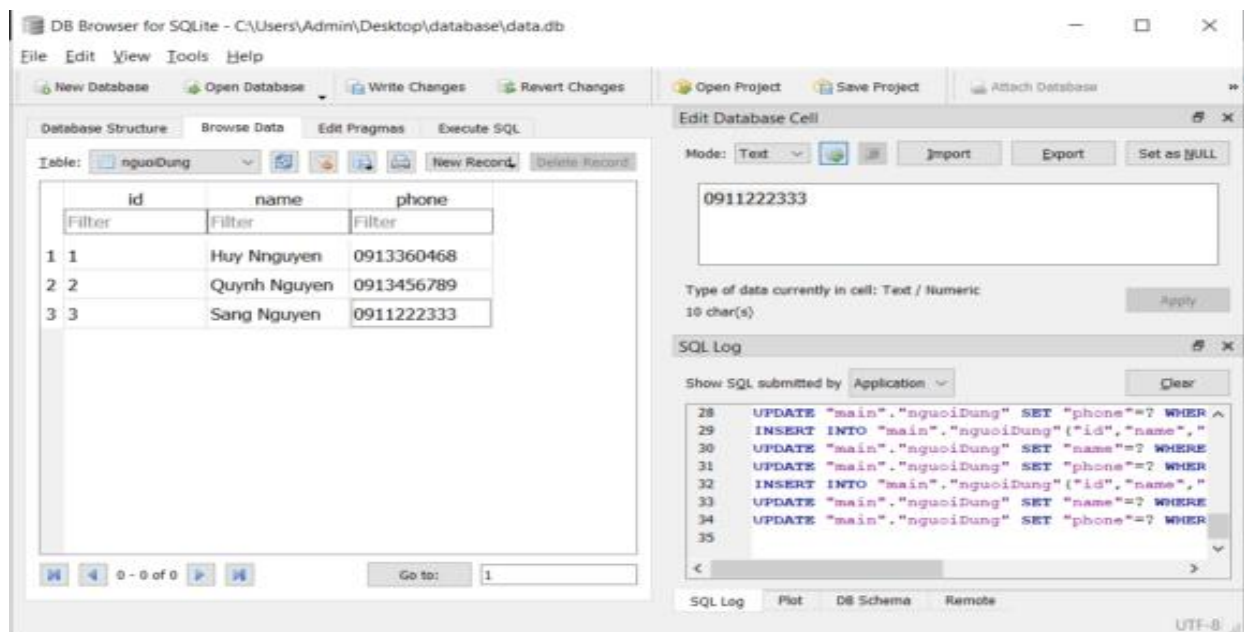
Bước 1: Nhấn “New Database” và đặt tên cho file CSDL.

Bước 2: Khai báo tên bảng và các cột bằng cách nhấn nút “Add field”



PK là khóa chính và AI là tự động tăng. Bấm OK để kết thúc tạo bảng

Bước 3: Nhấn tab Browse Data > New Record để thêm dòng dữ liệu mẫu



Bước 4: Nhấn **Save Project** để lưu dữ liệu vào file **dethi.db** tạo lúc ban đầu

2.2.2. Tạo project/Tạo thư mục Asset và copy CSDL vào project

Bước 1: Tạo thư mục assets: Vào File > New > Folder > Assets Folder > Xuất hiện thư mục Assets trong Project.

Bước 2: Trong giao diện Android Studio > Copy file **dethi.db** vào thư mục Assets.

2.2.3. Xử lý code

Ví dụ: Tạo giao diện chứa listview để hiển thị thông tin về thẻ loại, editText để người dùng gõ tên thẻ loại, button Search để tìm thẻ loại trong cơ sở dữ liệu và Refresh để làm mới dữ liệu trên editText.

- File activity.xml

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal"
    tools:layout_editor_absoluteX="8dp"
    tools:layout_editor_absoluteY="370dp">
    <EditText
        android:id="@+id/edIDTheLoai"
        android:layout_width="36dp"
        android:layout_height="wrap_content"
```

```
    android:layout_weight="1"
    android:ems="10"
    android:hint="tên thể loại"
    android:inputType="textPersonName"
    android:text="1" />
```

```
<Button
```

```
    android:id="@+id/btnSearch"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="Tìm" />
```

```
<Button
```

```
    android:id="@+id/btnRefresh"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_weight="1"
    android:text="Refresh" />
```

```
</LinearLayout>
```

```
<ListView
```

```
    android:id="@+id/listView"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginTop="8dp"
    android:layout_marginEnd="8dp"
    app:layout_constraintBottom_toBottomOf="parent"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">
```

```
</ListView>
```

- **File dongtheloai.xml: chứa các dòng the loại trên listView**

```
<LinearLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="8dp"
    android:layout_marginLeft="8dp"
    android:layout_marginTop="8dp"
    android:clickable="false"
    android:orientation="horizontal"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent">
    <TextView
        android:id="@+id/txtTenTheLoai"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="TextView" />
    <TextView
        android:id="@+id/txtIdTheLoai"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_weight="1"
        android:text="TextView" />
</LinearLayout>
```

- **File java dongTheLoai.java**

```
public class dongTheLoai {
    public int idTheLoai;
    public String TenTheLoai;
    public dongTheLoai(int idTheLoai, String tenTheLoai) {
        this.idTheLoai = idTheLoai;
```

```

        TenTheLoai = tenTheLoai;
    }
    public int getIdTheLoai() {
        return idTheLoai;
    }
    public void setIdTheLoai(int idTheLoai) {
        this.idTheLoai = idTheLoai;
    }
    public String getTenTheLoai() {
        return TenTheLoai;
    }
    public void setTenTheLoai(String tenTheLoai) {
        TenTheLoai = tenTheLoai;
    }
    @Override
    public String toString() {
        return "dongTheLoai{" +
            "idTheLoai=" + idTheLoai +
            ", TenTheLoai=" + TenTheLoai + "\" +
            '}; }}

```

- File DBHelper.java để kết nối tới CSDL

```

public class DBHeper extends SQLiteOpenHelper {
    String DB_PATH = null;
    private static final int DATABASE_VERSION = 1;
    private static String DB_NAME = "dethi.db";
    private SQLiteDatabase myDataBase;
    private final Context myContext;

    public DBHeper(Context context) {
        super(context, DB_NAME, null, DATABASE_VERSION);
    }

```

```

this.myContext = context;
this.DB_PATH = "/data/data/" + context.getPackageName() + "/" + "databases/";
Log.e("Path 1", DB_PATH);
try {
    createDataBase();
} catch (IOException ioe) {
    throw new Error("Unable to create database");
} try {
    openDataBase();
} catch (SQLException sqle) {
    throw sqle;
} }

public DBHeper(@Nullable Context context, @Nullable String name, @Nullable
SQLiteDatabase.CursorFactory factory, int version, Context myContext) {
    super(context, name, factory, version);
    this.myContext = myContext;
}

public void createDataBase() throws IOException {
    boolean dbExist = checkDataBase();
    if (dbExist) {
    } else {
        this.getReadableDatabase();
        try {
            copyDataBase();
        } catch (IOException e) {
            throw new Error("Error copying database");
        } } }

private boolean checkDataBase() {
    SQLiteDatabase checkDB = null;

```

```

try {
    String myPath = DB_PATH + DB_NAME;
    checkDB = SQLiteDatabase.openDatabase(myPath, null,
SQLiteDatabase.OPEN_READONLY);
} catch (SQLException e) {
}
if (checkDB != null) {
    checkDB.close();
}
return checkDB != null ? true : false;
}

private void copyDataBase() throws IOException {
    InputStream myInput = myContext.getAssets().open(DB_NAME);
    String outFileName = DB_PATH + DB_NAME;
    OutputStream myOutput = new FileOutputStream(outFileName);
    byte[] buffer = new byte[10];
    int length;
    while ((length = myInput.read(buffer)) > 0) {
        myOutput.write(buffer, 0, length);
    }
    myOutput.flush();
    myOutput.close();
    myInput.close(); }

public void openDataBase() throws SQLException {
    String myPath = DB_PATH + DB_NAME;
    myDataBase = SQLiteDatabase.openDatabase(myPath, null,
SQLiteDatabase.OPEN_READONLY);

}

@Override
public synchronized void close() {

```

```

    if (myDataBase != null)
        myDataBase.close();
    super.close();
}
@Override
public void onCreate(SQLiteDatabase db) { }
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    if (newVersion > oldVersion)
        try {
            copyDataBase();
        } catch (IOException e) {
            e.printStackTrace(); } }}

```

- File TheLoaiDbHelper.java: kết nối và thêm xóa sửa bảng TheLoai trong CSDL.

```

public class TheLoaiDbHelper {
    private SQLiteDatabase myDataBase;
    private DBHeper dbHeper;
    private static String DB_NAME = "dethi.db";
    private static String TABLE_NAME = "theLoai";
    private static String ID_COLUMN = "idTheLoai";
    private static String COLUMN1 = "TenTheLoai";
    public TheLoaiDbHelper(Context context) {
        dbHeper = new DBHeper(context);
    }
    //Lấy toàn bộ dữ liệu trả về cursor
    public Cursor query(String table, String[] columns, String selection, String[]
selectionArgs, String groupBy, String having, String orderBy) {
    //    myDataBase = this.getReadableDatabase();
        return myDataBase.query(table, null, null, null, null, null, null);
    }
}

```

//Lấy dòng dữ liệu theo điều kiện

```
public dongTheLoai getdongTheLoai(int id) {
    myDataBase = dbHeper.getReadableDatabase();
    dongTheLoai dongTheLoai = null;
    String[] columns = new String[]{ID_COLUMN, COLUMN1};
    String selection = ID_COLUMN + " = ?";
    String[] selectionArgs = new String[]{String.valueOf(id)};
    Cursor cursor = myDataBase.query(TABLE_NAME, columns, selection,
selectionArgs, null, null, null);
    if (cursor != null && cursor.moveToFirst()) {
        dongTheLoai = new dongTheLoai(cursor.getInt(0), cursor.getString(1));
        cursor.close();
    }
    myDataBase.close();
    return dongTheLoai;
}
```

//Lấy toàn bộ dữ liệu không điều kiện

```
public List<dongTheLoai> getAlltheLoai() {
    SQLiteDatabase db = dbHeper.getReadableDatabase();
    List<dongTheLoai> listTheLoai = new ArrayList<>();
    String sql = "SELECT * FROM " + TABLE_NAME;
    Cursor cursor = db.rawQuery(sql, null);
    if (cursor != null && cursor.moveToFirst()) {
        do {
            listTheLoai.add(new dongTheLoai(cursor.getInt(0), cursor.getString(1)));
        } while (cursor.moveToNext());
        cursor.close();
    }
    db.close();
    return listTheLoai;
}
```

```

}
//Lấy dữ liệu theo điều kiện
public List<dongTheLoai> getdongtheLoai1(int idtheLoai) {
    SQLiteDatabase db = dbHeper.getReadableDatabase();
    List<dongTheLoai> listTheLoai = new ArrayList<>();
    String sql = "SELECT * FROM " + TABLE_NAME + " where idTheLoai = ?";
    String[] selectionArgs = new String[]{String.valueOf(idtheLoai)};
    Cursor cursor = db.rawQuery(sql, selectionArgs);
    if (cursor != null && cursor.moveToFirst()) {
        do {
            listTheLoai.add(new dongTheLoai(cursor.getInt(0), cursor.getString(1)));
        } while (cursor.moveToNext());
        cursor.close();
    }
    db.close();
    return listTheLoai;
}

public void updateTheLoai(dongTheLoai dongTheLoai) {
    SQLiteDatabase db = dbHeper.getWritableDatabase();
    db.execSQL("UPDATE theLoai SET tenTheLoai=? where idTheLoai = ?",
        new String[]{dongTheLoai.TenTheLoai + "", dongTheLoai.idTheLoai + ""});
}

public void insertTheLoai(dongTheLoai dongTheLoai) {
    SQLiteDatabase db = dbHeper.getWritableDatabase();
    db.execSQL("INSERT INTO theLoai (tenTheLoai ) VALUES (?)",
        new String[]{dongTheLoai.TenTheLoai + ""});
}

public void deletedongTheLoai(final int idTheLoai) {
    SQLiteDatabase db = dbHeper.getWritableDatabase();

```

```

        db.execSQL("DELETE FROM theLoai where idTheLoai = ?", new
String[] {String.valueOf(idTheLoai)});
    }}

```

- File AdapterListview.java

```

public class AdapterListevew extends BaseAdapter {
    Activity context;
    ArrayList< dongTheLoai > list;
    public AdapterListevew (Activity context, ArrayList< dongTheLoai > list) {
        this.context = context;
        this.list = list;
    }
    @Override
    public int getCount() {
        return list.size();
    }
    @Override
    public Object getItem(int position) {
        return null;
    }
    @Override
    public long getItemId(int position) {
        return 0; }
    @Override
    public View getView(int position, View convertView, ViewGroup parent) {
        LayoutInflater inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
        View row = inflater.inflate(R.layout.nguoidung, null);
        TextView txtId = (TextView) row.findViewById(R.id.txtTheLoai);
        TextView txtName = (TextView) row.findViewById(R.id.txtTenTheLoai);
        final dongTheLoai dongTheLoai = list.get(position);

```

```

txtId.setText(dongTheLoai. TheLoai + "");
txtName.setText(dongTheLoai. TenTheLoai);
return row;
}}

```

- File MainActivity.java

```

public class MainActivity extends AppCompatActivity {
    Cursor cursor = null;
    ListView listView;
    ArrayList<dongTheLoai> listTheLoai;
    AdapterListview adapter;
    TheLoaiDbHelper myDbHelper;
    EditText edIDTheLoai;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        myDbHelper = new TheLoaiDbHelper (MainActivity.this);
        addControls();
        readData();    searchData();
    }
    private void searchData () {
        ((Button) findViewById(R.id.btnSearch)).setOnClickListener(new
View.OnClickListener() {
        @Override
        public void onClick(View v) {
            String[] columns = {"idTheLoai", "tenTheLoai"};
            String selection = "idTheLoai = ? ";
            String[] selectionArgs = {edIDTheLoai.getText().toString()};
            cursor = myDbHelper.query("theLoai", null, null, null, null, null, null);
            for (int i = 0; i < cursor.getCount(); i++) {

```

```

        cursor.moveToPosition(i);
        int idTheLoai = cursor.getInt(0);
        String tenTheLoai = cursor.getString(1);
        listTheLoai.clear();
        listTheLoai.add(new dongTheLoai(idTheLoai, tenTheLoai ));
    }
    adapter.notifyDataSetChanged();
}    });    }

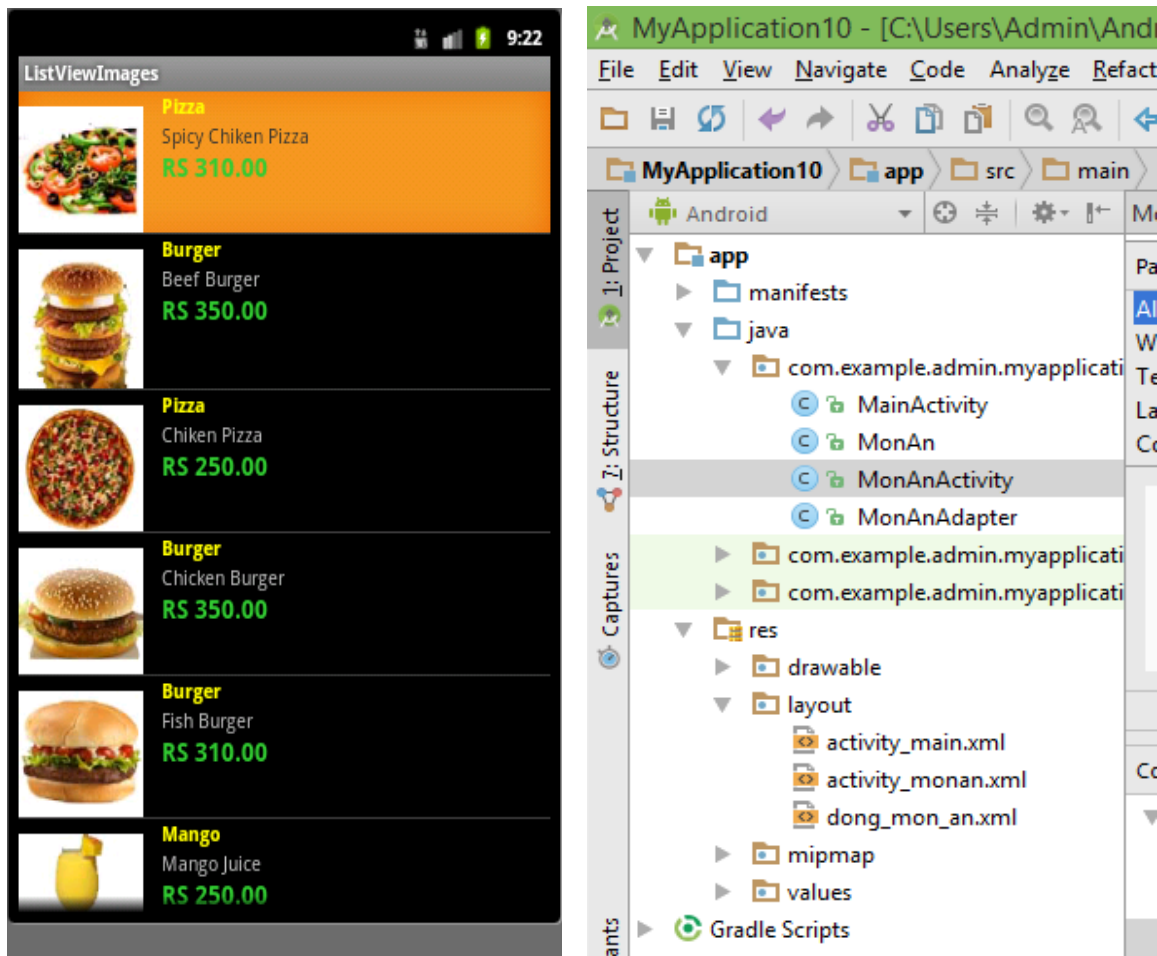
private void addControls () {
    edIDTheLoai = findViewById(R.id.edIDTheLoai);
    listView = (ListView) findViewById(R.id.listView);
    listTheLoai = new ArrayList<>();
    adapter = new AdapterListview(this, listTheLoai);
    listView.setAdapter(adapter);
}

private void readData () {
    cursor = myDbHelper.query("theLoai", null, null, null, null, null, null);
    for (int i = 0; i < cursor.getCount(); i++) {
        cursor.moveToPosition(i);
        int id = cursor.getInt(0);
        String tenTheLoai = cursor.getString(1);
        listTheLoai.add(new dongTheLoai(id, tenTheLoai));
    }
    adapter.notifyDataSetChanged();
}
}

```

BÀI TẬP

Bài 1: Tạo ứng dụng gọi món ăn.



Bước 1: Tạo listview trong file activitymain.xml

Bước 2: Tạo mới layout dong_mon_an.xml hiển thị từng dòng dữ liệu trên listview

<ImageView

```
android:layout_width="70dp"
android:layout_height="70dp"
android:id="@+id/invMonAn"
android:layout_alignParentTop="true"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true" />
```

<TextView

```
android:layout_width="match_parent"
android:layout_height="35dp"
```

```

        android:text="New Text"
        android:id="@+id/tvGiaMonAn"
        android:layout_alignBottom="@+id/invMonAn"
        android:layout_toRightOf="@+id/invMonAn"
        android:layout_toEndOf="@+id/invMonAn" />
<TextView
    android:layout_width="match_parent"
    android:layout_height="35dp"
    android:text="New Text"
    android:id="@+id/tvTenMonAn"
    android:layout_above="@+id/tvGiaMonAn"
    android:layout_toRightOf="@+id/invMonAn"
    android:layout_toEndOf="@+id/invMonAn" />
</RelativeLayout>

```

Bước 3: Tạo class MonAn thiết lập các giá trị cho món ăn

```

public class MonAn {
    public int id;
    public String Ten;
    public int Gia;
    public int HinhAnh;
    public MonAn(){ }
    public MonAn(String ten, int gia, int hinhAnh) {
        Ten = ten;
        Gia = gia;
        HinhAnh = hinhAnh;
    }
    public MonAn(int id, String ten, int gia, int hinhAnh) {
        this.id = id;
        Ten = ten;
        Gia = gia;
    }
}

```

```

        HinhAnh = hinhAnh;
    }
    public int getId() {
        return id;
    }
    public String getTen() {
        return Ten;
    }
    public int getGia() {
        return Gia;
    }
    public int getHinhAnh() {
        return HinhAnh;
    }
    public void setId(int id) {
        this.id = id;
    }
    public void setTen(String ten) {
        Ten = ten;
    }
    public void setGia(int gia) {
        Gia = gia;
    }
    public void setHinhAnh(int hinhAnh) {
        HinhAnh = hinhAnh;
    }
    @Override
    public String toString() {
        return "MonAn{" +
            ", Ten=" + Ten + "\" +

```

```

        ", Giá=" + Gia + "\" +
        ", Hình Ảnh=" + HìnhAnh + "\" +
        '}';
    }
}

```

Bước 4: Tạo class MonAnAdapter để đặt dữ liệu lên listview

```

public class MonAnAdapter extends BaseAdapter {
    Context context;
    int mylayout;
    ArrayList<MonAn> monAnList;
    public MonAnAdapter(Context context, int mylayout, ArrayList<MonAn>
monAnList) {
        this.context = context;
        this.mylayout = mylayout;
        this.monAnList = monAnList;
    }
    @Override
    public int getCount() {
        return monAnList.size();
    }
    @Override
    public Object getItem(int position) {
        return null;
    }
    @Override
    public long getItemId(int position) {
        return 0;
    }
    @Override
    public View getView(int position, View convertView, ViewGroup parent) {

```

```

    LayoutInflater inflater = (LayoutInflater)
context.getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    convertView = inflater.inflate(mylayout,null);
    //Ánh xạ và gán giá trị
    TextView txtTen = (TextView) convertView.findViewById(R.id.tvTenMonAn);
    txtTen.setText(monAnList.get(position).Ten);
    TextView txtGia = (TextView) convertView.findViewById(R.id.tvGiaMonAn);
    txtGia.setText(monAnList.get(position).Gia + " ");
    ImageView imgHinh =(ImageView) convertView.findViewById(R.id.imvMonAn);
    imgHinh.setImageResource(monAnList.get(position).HinhAnh);
    return convertView;
}
}

```

Bước 5: Tạo lớp DatabaseHeper kết nối và làm việc với CSDL

```

public class DatabaseHeper extends SQLiteOpenHelper {
    private static final String TAG = "SQLite";
    private static final int DATABASE_VERSION = 1;
    // Tên cơ sở dữ liệu.
    private static final String DATABASE_NAME = "MonAnDatabase1";
    // Tên bảng: MonAn.
    private static final String TABLE = "MonAn";
    private static final String COLUMN_ID = "Id";
    private static final String COLUMN_TenMonAn = "TenMonAn";
    private static final String COLUMN_Gia = "Gia";
    private static final String COLUMN_HinhAnh = "HinhAnh";
    String script = "CREATE TABLE " + TABLE + "("
        + COLUMN_ID + " INTEGER PRIMARY KEY AUTOINCREMENT Not
NULL ," + COLUMN_TenMonAn + " TEXT,"
        + COLUMN_Gia + " INTEGER," + COLUMN_HinhAnh + " TEXT" + ")";
    private int MonAnsCount;

```

```

public DatabaseHelper(Context context) {
    super(context, DATABASE_NAME, null, DATABASE_VERSION);
}
@Override
public void onCreate(SQLiteDatabase db) {
    Log.i(TAG, "MyDatabaseHelper.onCreate ... ");
    db.execSQL(script);
}
@Override
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
    Log.i(TAG, "MyDatabaseHelper.onUpgrade ... ");
    // Hủy (drop) bảng cũ nếu nó đã tồn tại.
    db.execSQL("DROP TABLE IF EXISTS " + TABLE);
    // Và tạo lại.
    onCreate(db);
}
public void createDefaultMonAnsIfNeed() {
    Log.i(TAG, "MyDatabaseHelper is createDefaultMonAnsIfNeed ");
    int count = this.getMonAnsCount();
    if (count == 0) {
        MonAn monAn1 = new MonAn("Cháo",100,R.drawable.face1);
        MonAn monAn2 = new MonAn("Phở",100,R.drawable.face2);
        this.addMonAn(monAn1);
        this.addMonAn(monAn2);
    }
}
private void addMonAn(MonAn monAn) {
    Log.i(TAG, "MyDatabaseHelper.addMonAN ... " + monAn.getTen());
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();

```

```

values.put(COLUMN_TenMonAn, monAn.getTen());
values.put(COLUMN_Gia, monAn.getGia());
values.put(COLUMN_HinhAnh, monAn.getHinhAnh());
db.insert(TABLE, null, values);
db.close(); }

public MonAn getMonAn(int id) {
    Log.i(TAG, "MyDatabaseHelper.getMonAn ... " + id);
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor cursor = db.query(TABLE, new String[] { COLUMN_ID,
        COLUMN_TenMonAn, COLUMN_Gia,COLUMN_HinhAnh },
COLUMN_ID + "=?",
        new String[] { String.valueOf(id) }, null, null, null, null);
    if (cursor != null)
        cursor.moveToFirst();
    MonAn monAn = new MonAn(Integer.parseInt(cursor.getString(0)),
        cursor.getString(1),Integer.parseInt(cursor.getString(2)),
Integer.parseInt(cursor.getString(3)));
    // return MonAn
    return monAn;
}

public ArrayList<MonAn> getAllMonAn() {
    Log.i(TAG, "MyDatabaseHelper.getAllMonAns ... ");
    ArrayList<MonAn> MonAnList = new ArrayList<MonAn>();
    // Select All Query
    String selectQuery = "SELECT * FROM " + TABLE;
    SQLiteDatabase db = this.getWritableDatabase();
    Cursor cursor = db.rawQuery(selectQuery, null);
    if (cursor.moveToFirst()) {
        do {
            MonAn monAn = new MonAn();

```

```

        monAn.setId(Integer.parseInt(cursor.getString(0)));
        monAn.setTen(cursor.getString(1));
        monAn.setGia(Integer.parseInt(cursor.getString(2)));
monAn.setHinhAnh(Integer.parseInt(cursor.getString(3)));
        MonAnList.add(monAn);
    } while (cursor.moveToNext());
}
return MonAnList;
}

public int getMonAnsCount() {
    Log.i(TAG, "MyDatabaseHelper.getMonAnsCount ... ");
    String countQuery = "SELECT * FROM " + TABLE;
    SQLiteDatabase db = this.getReadableDatabase();
    Cursor cursor = db.rawQuery(countQuery, null);
    int MonAnsCount = cursor.getCount();
    cursor.close();
    return MonAnsCount;
}

public int updateMonAn(MonAn monAn) {
    Log.i(TAG, "MyDatabaseHelper.updateMonAn ... " + monAn.getTen());
    SQLiteDatabase db = this.getWritableDatabase();
    ContentValues values = new ContentValues();
    values.put(COLUMN_TenMonAn, monAn.getTen());
    values.put(COLUMN_Gia, monAn.getGia());
    values.put(COLUMN_HinhAnh, monAn.getHinhAnh());
    return db.update(TABLE, values, COLUMN_ID + " = ?",
        new String[]{String.valueOf(monAn.getId())});
}

public void deleteMonAn(MonAn monAn) {
    Log.i(TAG, "MyDatabaseHelper.updateMonAn ... " + monAn.getId() );
}

```

```

        SQLiteDatabase db = this.getWritableDatabase();
        db.delete(TABLE, COLUMN_ID + " = ?",
            new String[] { String.valueOf(monAn.getId()) });
        db.close();
    }
}

```

- **Gọi hàm trong lớp DatabaseHeper để xử lý dữ liệu trong MainActivity**

```

public class MainActivity extends AppCompatActivity {
    private static final String TAG = "MonAn";
    Button btnAdd,btnDelete;
    TextView txtTenMonAn;
    ListView listView;
    final int currentDBversion = 1;
    final String BDVERSION = "DBVERSION_KEY";
    private final ArrayList<MonAn> MonAnList = new ArrayList<MonAn>();
    private ArrayAdapter<MonAn> listViewAdapter;
    MonAnAdapter anAdapter;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        Log.i(TAG, "MyDatabaseHelper bắt đầu ... ");
        listView = (ListView) findViewById(R.id.lvMonAn);
        DatabaseHeper db = new DatabaseHeper(this);
        db.createDefaultMonAnsIfNeed();
        ArrayList<MonAn> list= db.getAllMonAn();
        this.MonAnList.addAll(list);
        anAdapter = new
MonAnAdapter(MainActivity.this,R.layout.dong_mon_an,this.MonAnList );
        this.listView.setAdapter(this.anAdapter);
    }
}

```

}
}

3. TÓM TẮT TRÌNH TỰ THỰC HIỆN HOẶC QUY TRÌNH CÔNG NGHỆ

TT	Tên các bước công việc	Dụng cụ, thiết bị, vật tư	Yêu cầu kỹ thuật	Các chú ý về an toàn lao động
1	Khởi động Android Studio	Phần mềm Android Studio Máy tính.	Cài đặt chính xác phần mềm.	Đảm bảo an toàn cho người và thiết bị.
2	Thực hành thiết kế và code cho các nút lệnh	Phần mềm Android Studio Máy tính.	Cài đặt chính xác phần mềm.	Đảm bảo an toàn cho người và thiết bị.
3	Kiểm tra kết quả	Phần mềm Android Studio Máy tính.	Cài đặt chính xác phần mềm.	Đảm bảo an toàn cho người và thiết bị.

BÀI 8: ĐƯA ỨNG DỤNG LÊN THIẾT BỊ ANDROID VÀ UPLOAD LÊN ANDROID MARKET

1. MỤC TIÊU

Trình bày được cách upload ứng dụng lên điện thoại Android và upload lên Android market. Upload được ứng dụng lên điện thoại Android và upload lên Android market

Nghiêm túc, tự giác trong học tập. Đảm bảo an toàn cho người và thiết bị.

2. NỘI DUNG

2.1. Lý thuyết liên quan

Chuẩn bị ứng dụng trước khi upload

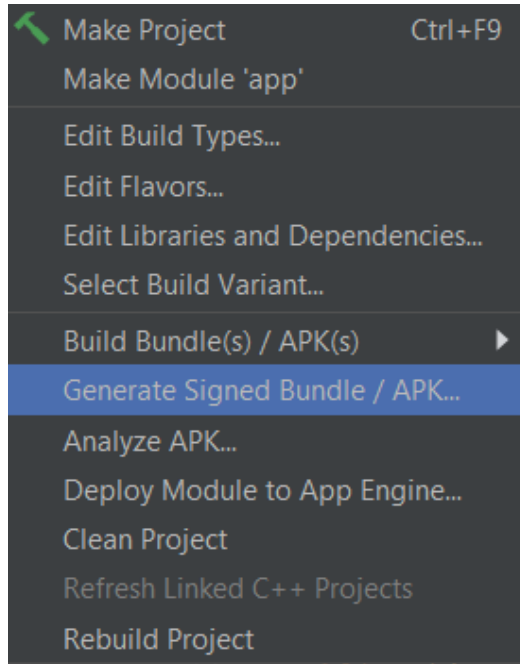
- Tạo file keystore (.jks)
- Nhúng keystore vào file apk

- Chuyển file apk từ debug -> release
- Submit Google Play (Cần tài khoản 25\$)

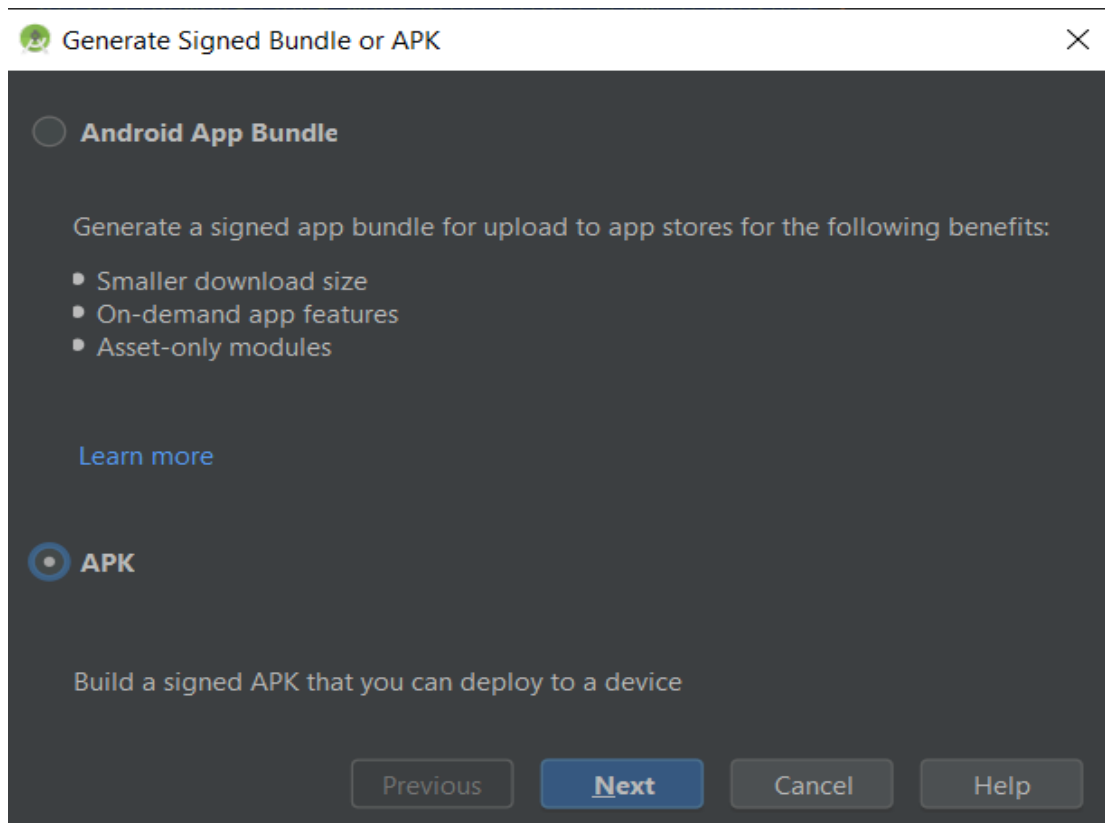
2.2 Trình tự thao tác

2.2.1. Upload ứng dụng lên thiết bị Andoird

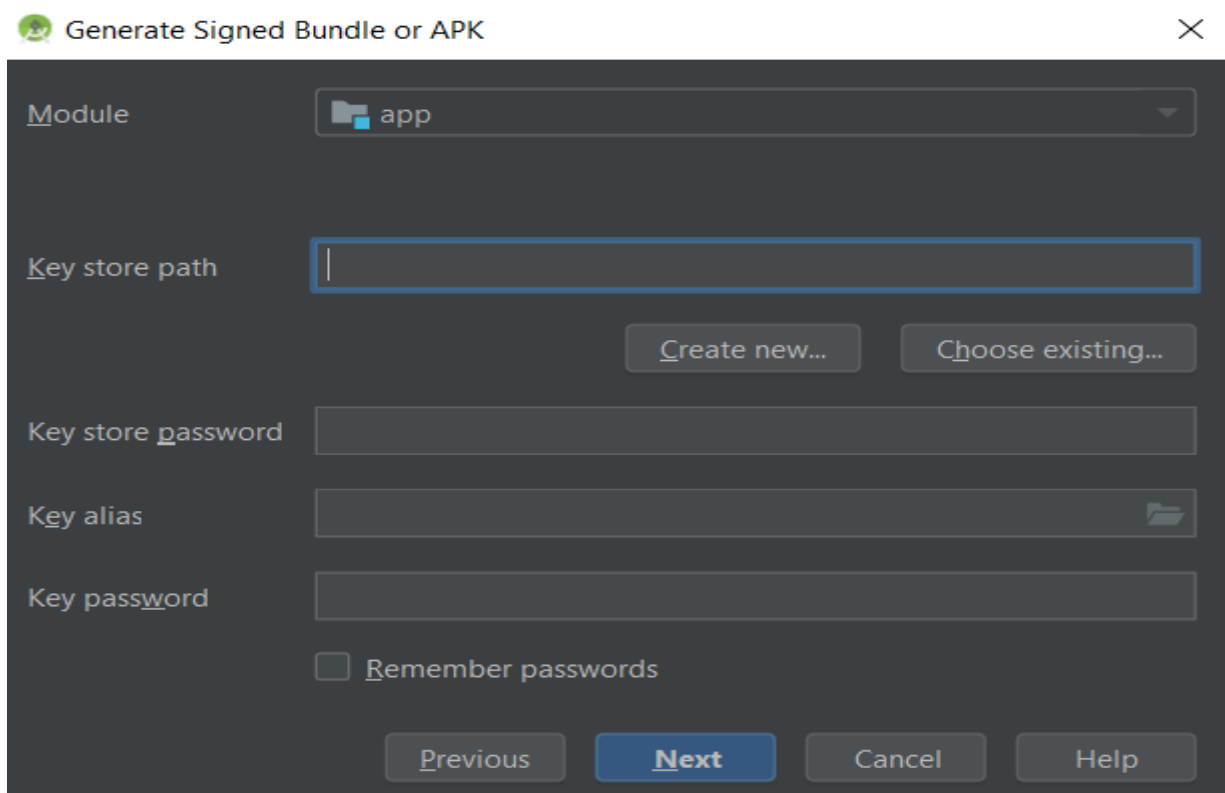
Bước 1: Tạo file keystore : Vào menu Build/ chọn Generate Signed APK > chọn mobile.



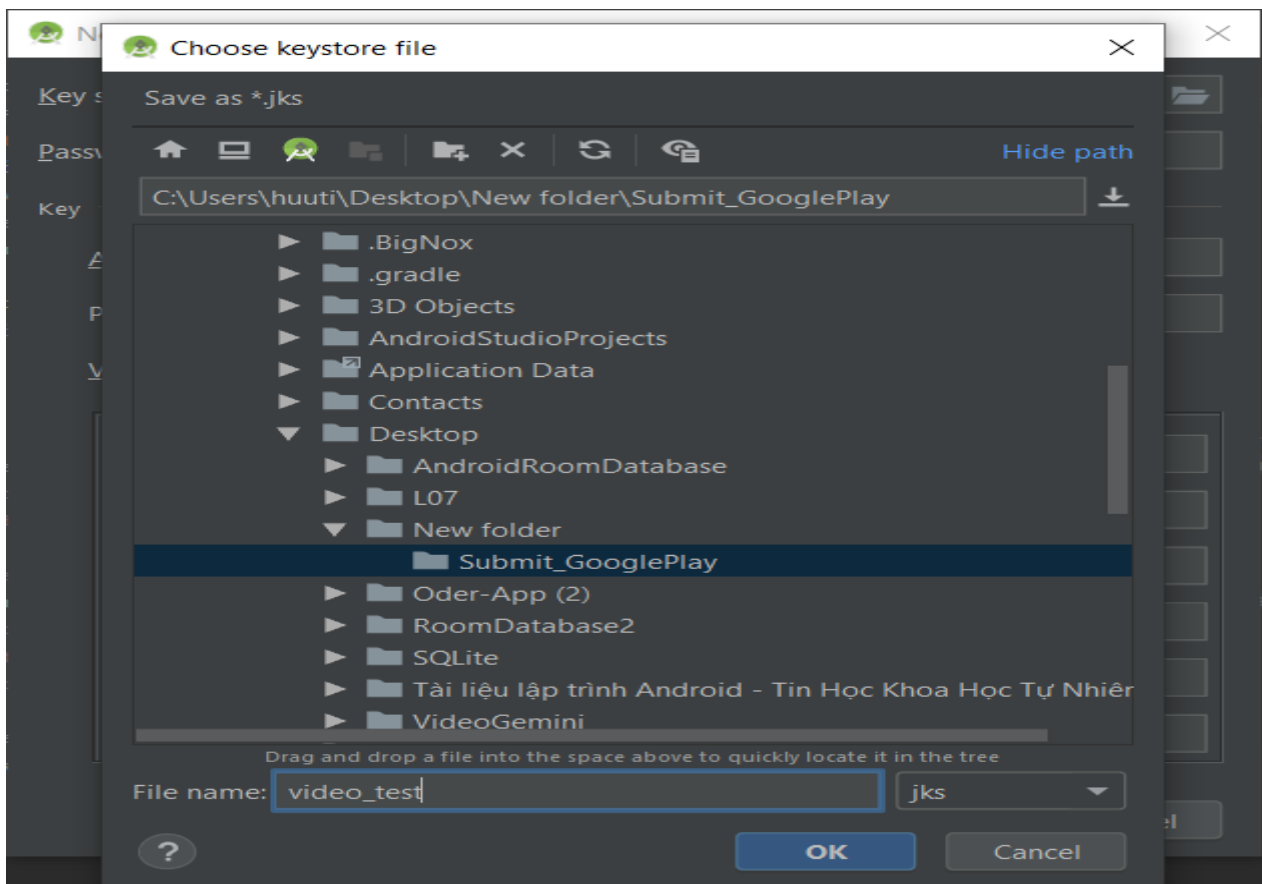
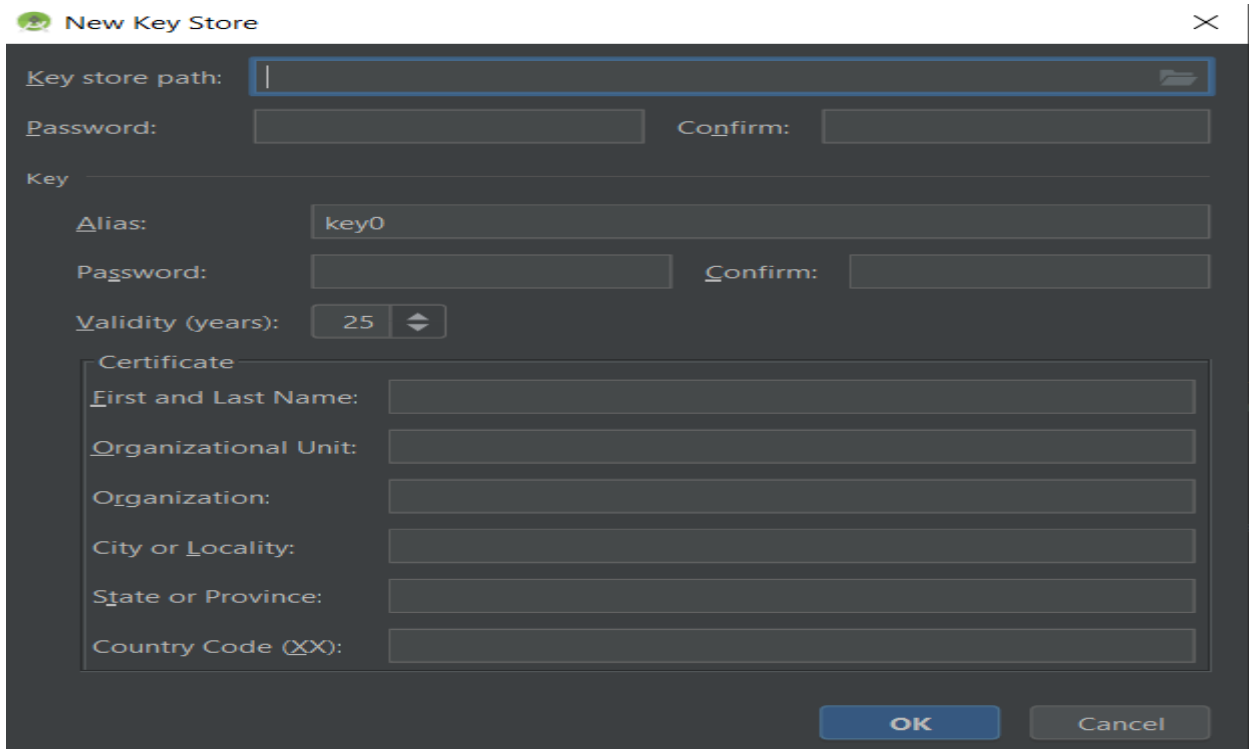
Sau khi chọn Generate Signed APK >



Bấm Next để tiếp tục:

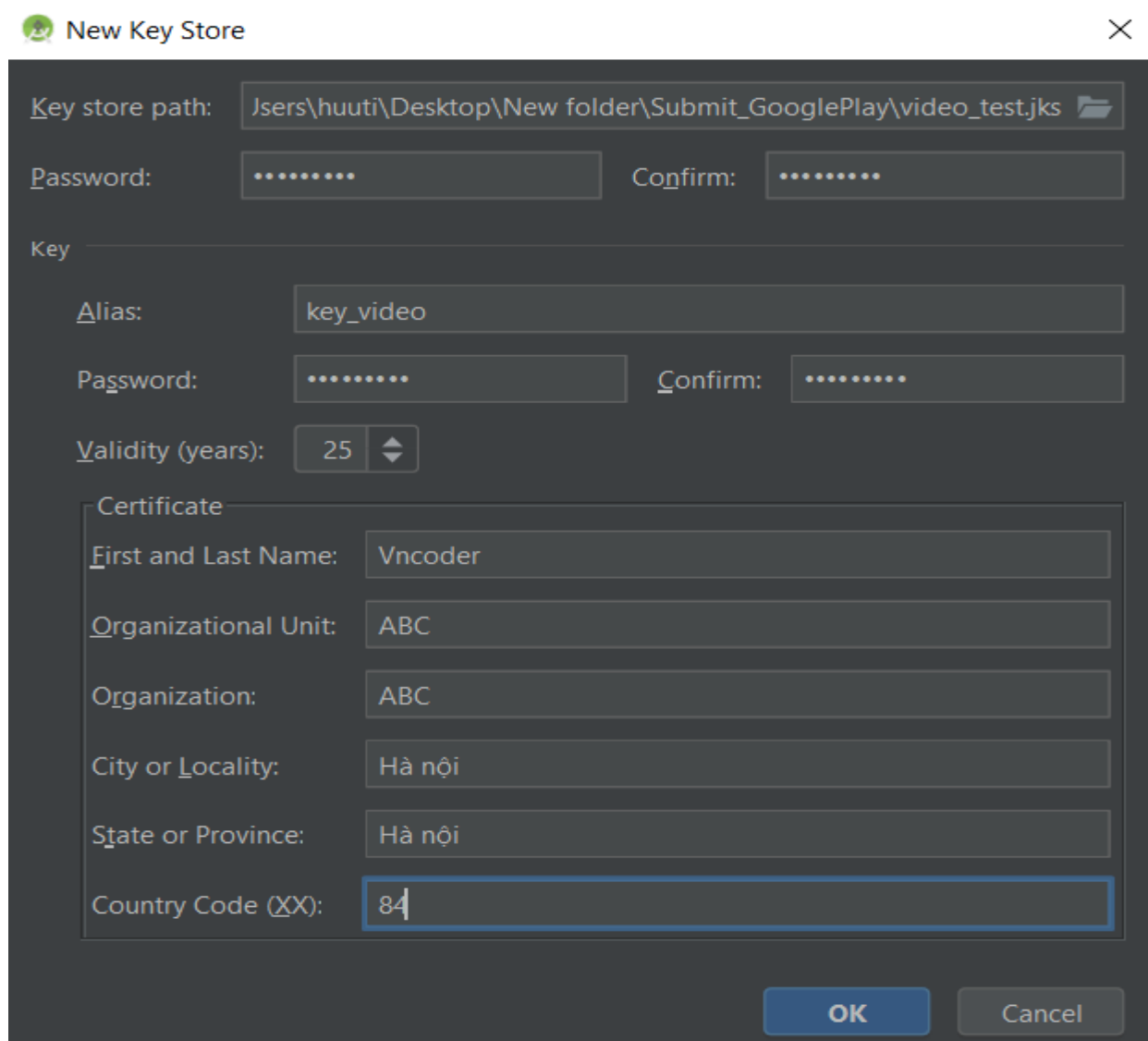


Mục Key store path: Là nơi lưu trữ key, nhấn vào nút “Create New...”:



Bước 2. Nhúng Keystore vào file apk

Sau khi đặt tên file rồi nhấn OK:



New Key Store [X]

Key store path: J:\sers\huuti\Desktop\New folder\Submit_GooglePlay\video_test.jks

Password: Confirm:

Key

Alias: key_video

Password: Confirm:

Validity (years): 25

Certificate

First and Last Name: Vncoder

Organizational Unit: ABC

Organization: ABC

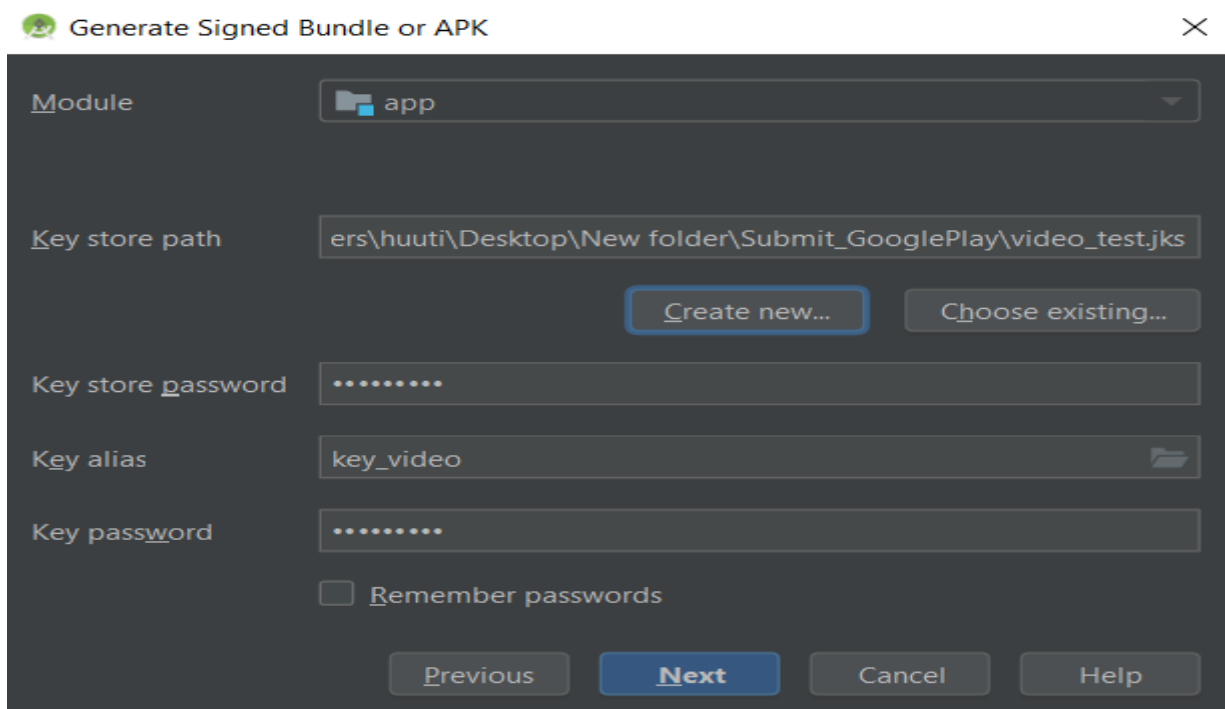
City or Locality: Hà nội

State or Province: Hà nội

Country Code (XX): 84

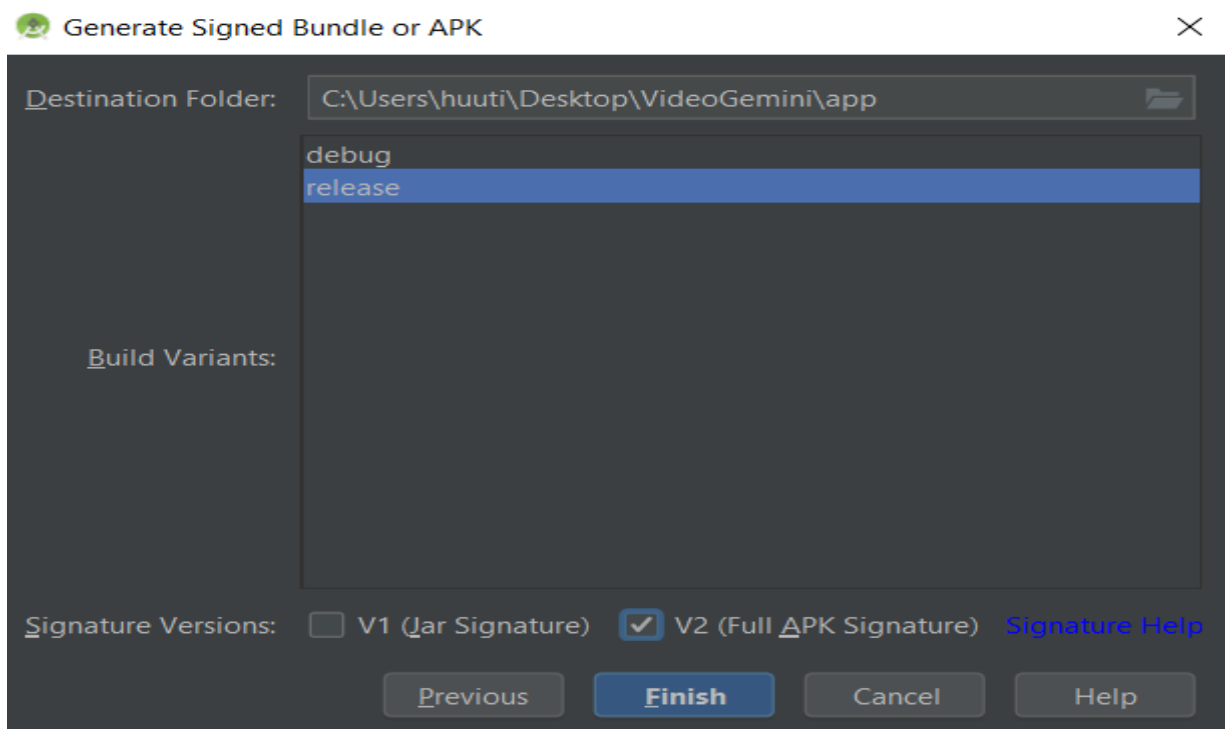
OK Cancel

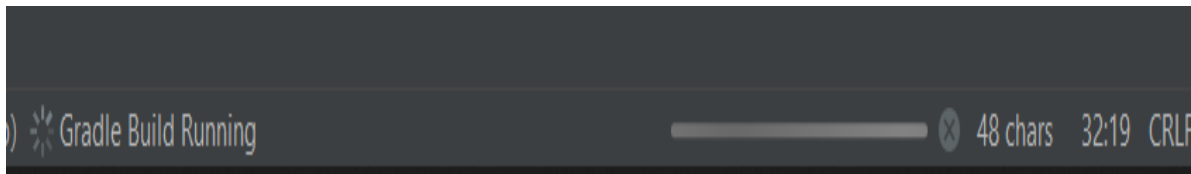
Ở màn hình trên tiến hành đặt mật khẩu cho Key Store, nhập thông tin cho Certificate. Chú ý là thông tin này vô cùng quan trọng, nếu như làm mất nó thì về sau muốn cập nhật phiên bản sẽ không bao giờ được, đồng nghĩa với việc phải viết lại ứng dụng (đặt package khác vì google chỉ cho tồn tại 1 package không trùng nhau). Do đó lưu ở đâu thì phải nhớ kỹ. Sau khi nhấn OK ở màn hình trên thì thông tin sẽ được hiển thị lại màn hình như sau:



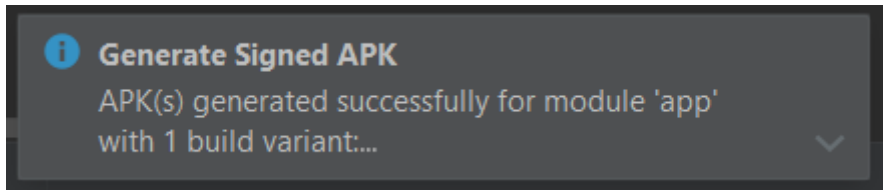
Bước 3. Chuyển file apk từ debug -> release

Tiếp tục bấm Next > Chọn Release rồi bấm Finish, sau đó chờ để Android studio tạo ra file xuất bản phần mềm, khi build xong thì nó hiển thị “Show in explorer” > bấm vào “Show in explorer” để lấy file xuất bản:



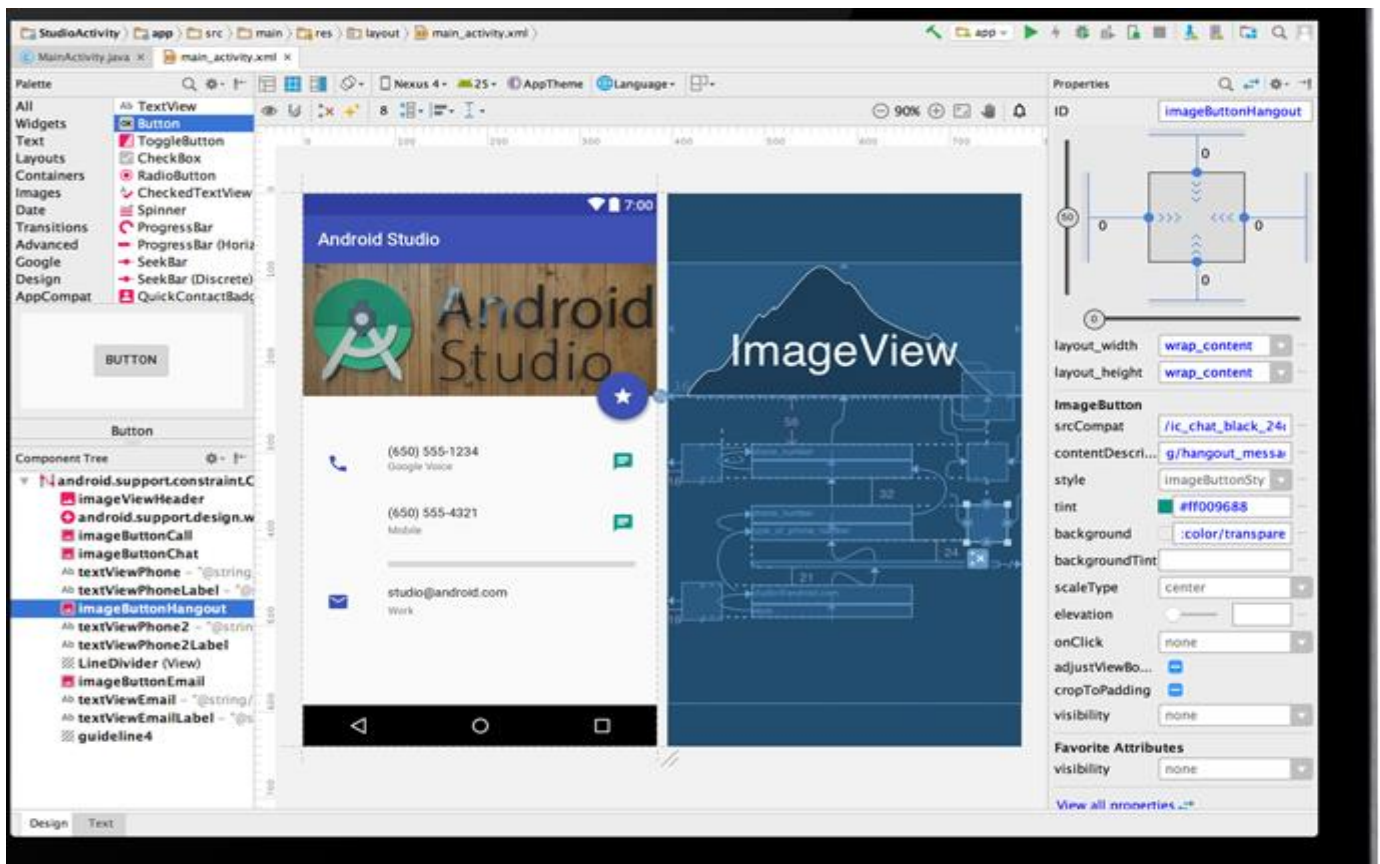


Xem thư mục chứa file xuất bản:



app.aab	3/18/2020 1:20 PM	AAB File	6,722 KB
output.json	3/18/2020 1:26 PM	JSON File	1 KB
video_test	3/18/2020 1:26 PM	Nox.apk	7,103 KB

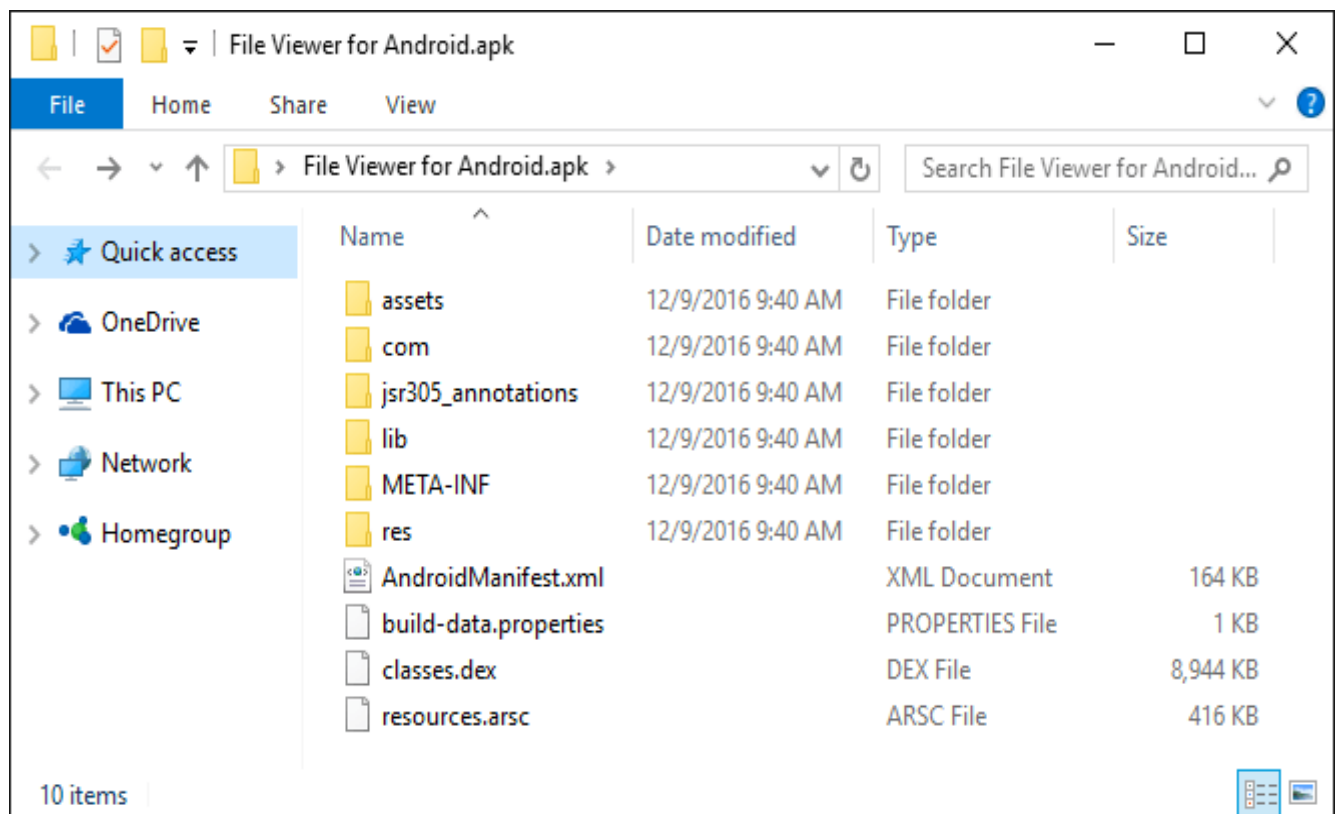
File APK là gì?



File APK là một định dạng tập tin ứng dụng được tạo cho Android, hệ điều hành di động của Google. Một số ứng dụng được cài đặt sẵn trên điện thoại Android, trong khi các ứng dụng khác có thể được tải xuống từ Google Play. Các ứng dụng được tải xuống từ Google Play được tự động cài đặt trên thiết bị của bạn, trong khi các ứng dụng được tải xuống từ các nguồn khác phải được cài đặt theo cách thủ công.

Thông thường, người dùng không bao giờ thấy File APK vì Android xử lý cài đặt ứng dụng trong nền thông qua Google Play hoặc nền tảng phân phối ứng dụng khác. Tuy nhiên, có rất nhiều trang web cung cấp tải File APK trực tiếp cho người dùng Android tự cài đặt ứng dụng. Trong trường hợp này, bạn nên cẩn thận rằng bạn tin cậy nguồn của File APK, bởi vì phần mềm độc hại có thể được phân phối trong các File APK, cũng như nó có thể trong các File Windows và .EXE.

File APK chứa gì?



File APK gói gọn tất cả các File cần thiết của một ứng dụng Android. Dưới đây là danh sách mô tả các File và thư mục nổi bật nhất:

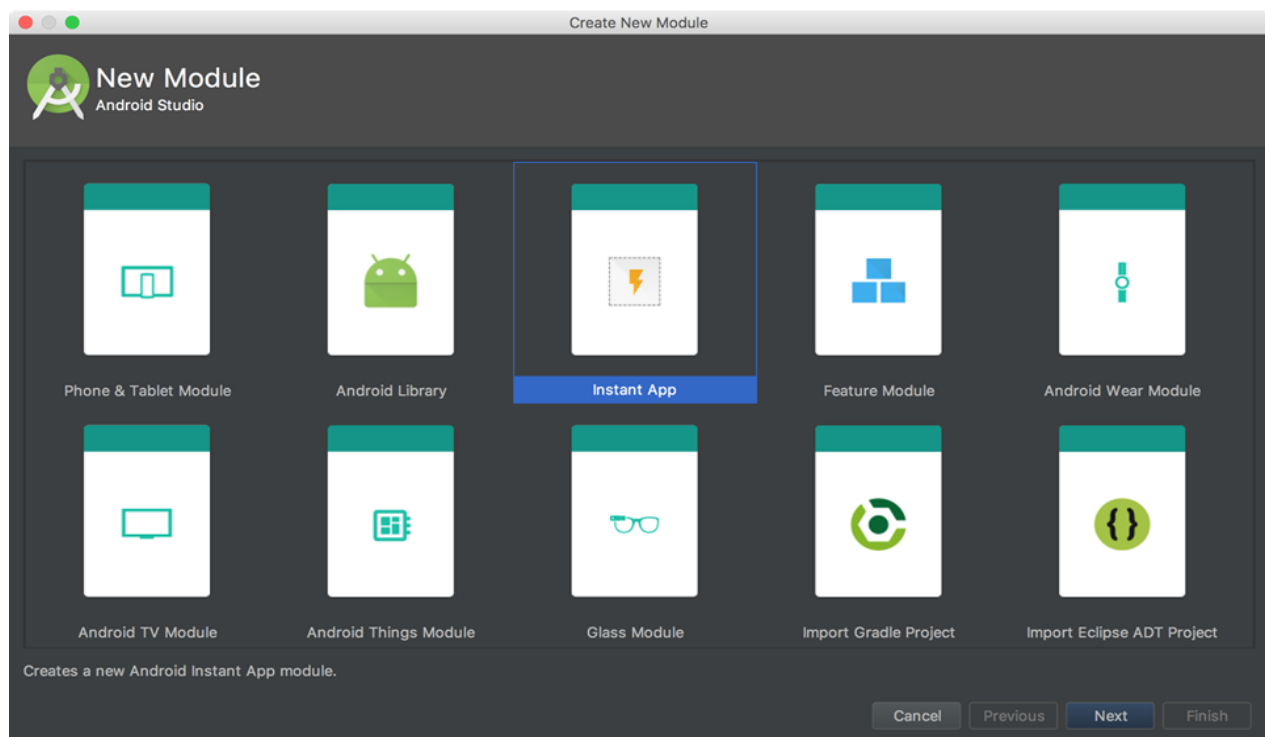
- META-INF /: Bao gồm File kê khai, chữ ký và danh sách các tài nguyên trong kho lưu trữ
- Lib /: Các thư viện gốc chạy trên kiến trúc thiết bị cụ thể (armeabi-v7a, x86, v.v.)
- Res /: Tài nguyên, chẳng hạn như hình ảnh, không được biên dịch thành resources.arsc
- AndroidManifest.xml: Mô tả tên, phiên bản và nội dung của File APK
- Classes.dex: Các lớp Java được biên dịch sẽ chạy trên thiết bị (File .EXE)
- Resources.arsc: Các tài nguyên đã biên dịch, chẳng hạn như chuỗi, được ứng dụng sử dụng

Làm cách nào để bung File APK?

Các File APK được lưu ở định dạng nén .ZIP và có thể được mở bằng bất kỳ công cụ giải nén Zip nào. Do đó, nếu bạn muốn bung File APK, bạn có thể đổi tên phần mở rộng của File thành ".zip" và mở File bằng các ứng dụng giải nén.

Tuy nhiên, để xem được nội dung bung File APK, các bạn sẽ phải dùng một trong số những phần mềm biên dịch sau. Đây đều là những công cụ viết phần mềm Android được các lập trình sử dụng. Do đó, nó sẽ có thể bung File APK và xem được nội dung cho bạn.

Để xem nội dung bung File APK, các bạn có thể sử dụng Android Studio - bộ công cụ lập trình Android được phát hành chính thức bởi Google. Bạn tải về và cài Android Studio lên máy sau đó khởi chạy lên, bấm Open để mở bung File APK bạn muốn xem.



Lúc này File APK sẽ hiển thị như là một ứng dụng Android thực thụ bên trong nền của chương trình Android Studio. Từ đây các bạn có thể khám phá thêm nhiều nội dung của ứng dụng. Mặt khác, bạn cũng có thể xem được những dòng code lập trình bung File APK với Android Studio.

2.2.2. Upload ứng dụng lên Android market.

Muốn xuất bản ứng dụng Android lên Play Store cần chuẩn bị:

- Một ứng dụng Android do cá nhân tự phát triển.
- Một thẻ thanh toán quốc tế (VISA hoặc MasterCard) để có thể đăng ký tài khoản Google Developers.
- Nhập đầy đủ thông tin được yêu cầu, sau đó chờ Google duyệt thì ứng dụng của sẽ được xuất bản ứng dụng Android lên Play Store.

a. Đăng kí tài khoản Google Developer

Link đăng kí trở thành Publisher: [Google Play Console and log in.](#)

Đăng nhập bằng tài khoản Google > Chọn “Developer Agreement” để chuyển sang màn hình thanh toán.



Google Play Console



Sign-in with your Google account



Accept Developer Agreement



Pay Registration Fee



Complete your Account details

You are signed in as...



dev@example.com

This is the Google account that will be associated with your Developer Console.

If you would like to use a different account, you can choose from the following options below. If you are an organization, consider registering a new Google account rather than using a personal account.

[SIGN IN WITH A DIFFERENT ACCOUNT](#) [CREATE A NEW GOOGLE ACCOUNT](#)

Before you continue...



Accept developer agreement

Read and agree to the [Google Play Developer distribution agreement](#).

I agree and I am willing to associate my account registration with the [Google Play Developer distribution agreement](#).



Review distribution countries

Review the distribution countries where you can distribute and sell applications. [Learn more](#)

If you are planning to sell apps or in-app products, check if you can have a merchant account in your country. [Learn more](#)



Credit card

Make sure you have your credit card handy to pay the \$25 registration fee in the next step.

[CONTINUE TO PAYMENT](#)

Sau khi thanh toán bằng thẻ VISA/ MasterCard xong > điền các thông tin cần thiết cho nhà phát triển như developer name, email address, website, phone number > Chọn “COMPLETE REGISTRATION.”

Sign-in with your Google account Accept Developer Agreement Pay Registration Fee Complete your Account details

You are almost done...

Just complete the following details. You can change this information later in your account settings if you need to.

Developer Profile Fields marked with * need to be filled before saving.

Developer name * 37/50

The developer name will appear to users under the name of your application.

Email address *

Website

Phone Number *

Include plus sign, country code and area code. For example, +1-800-555-0199.
[Why do we ask for your phone number?](#)

Email preferences

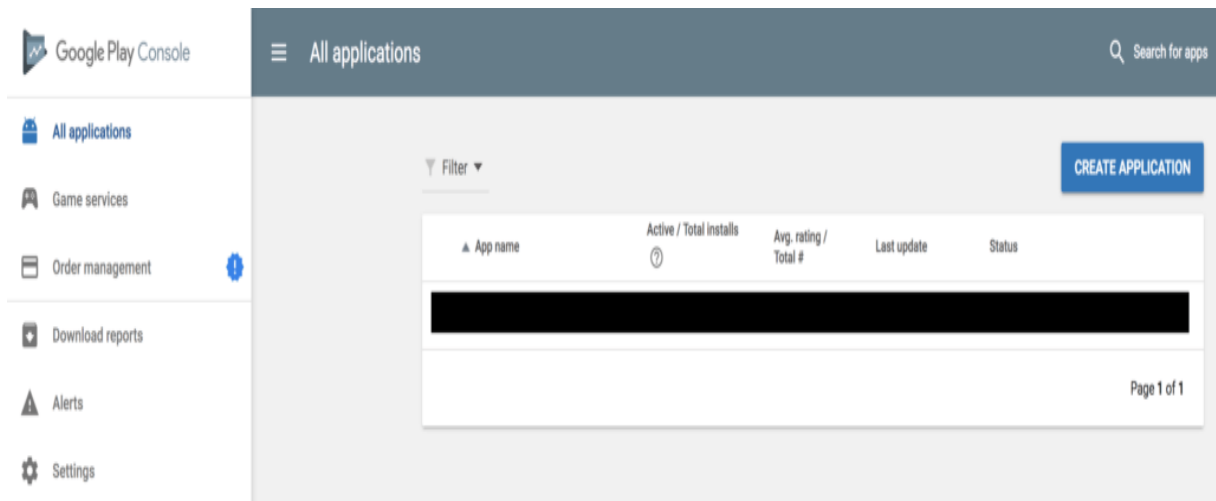
I'd like to get new feature announcements and tips to help improve my apps.

I'd like to give feedback to help improve the Google Play Developer Console.

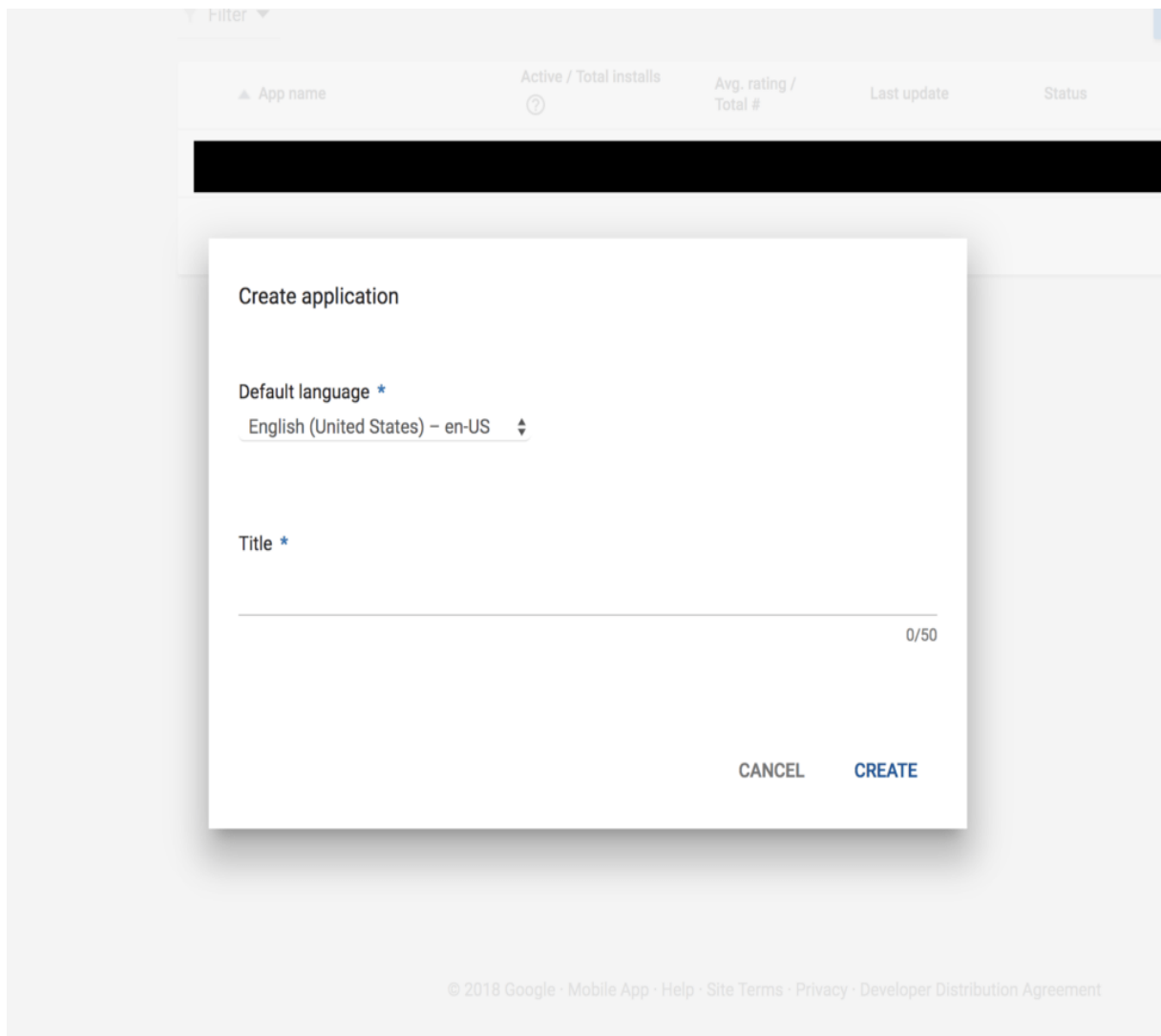
By accessing the Play Console, in accordance with the [Google Privacy Policy](#) and [Developer Distribution Agreement](#), you agree to share information with Google about your use of the Play Console. This data will be used for feature development and Console personalization. You can disable sharing usage data from within Settings > Preferences. [Learn more](#)

COMPLETE REGISTRATION

b. Tạo ứng dụng mới và điền thông tin mô tả ứng dụng
Chọn “CREATE APPLICATION”

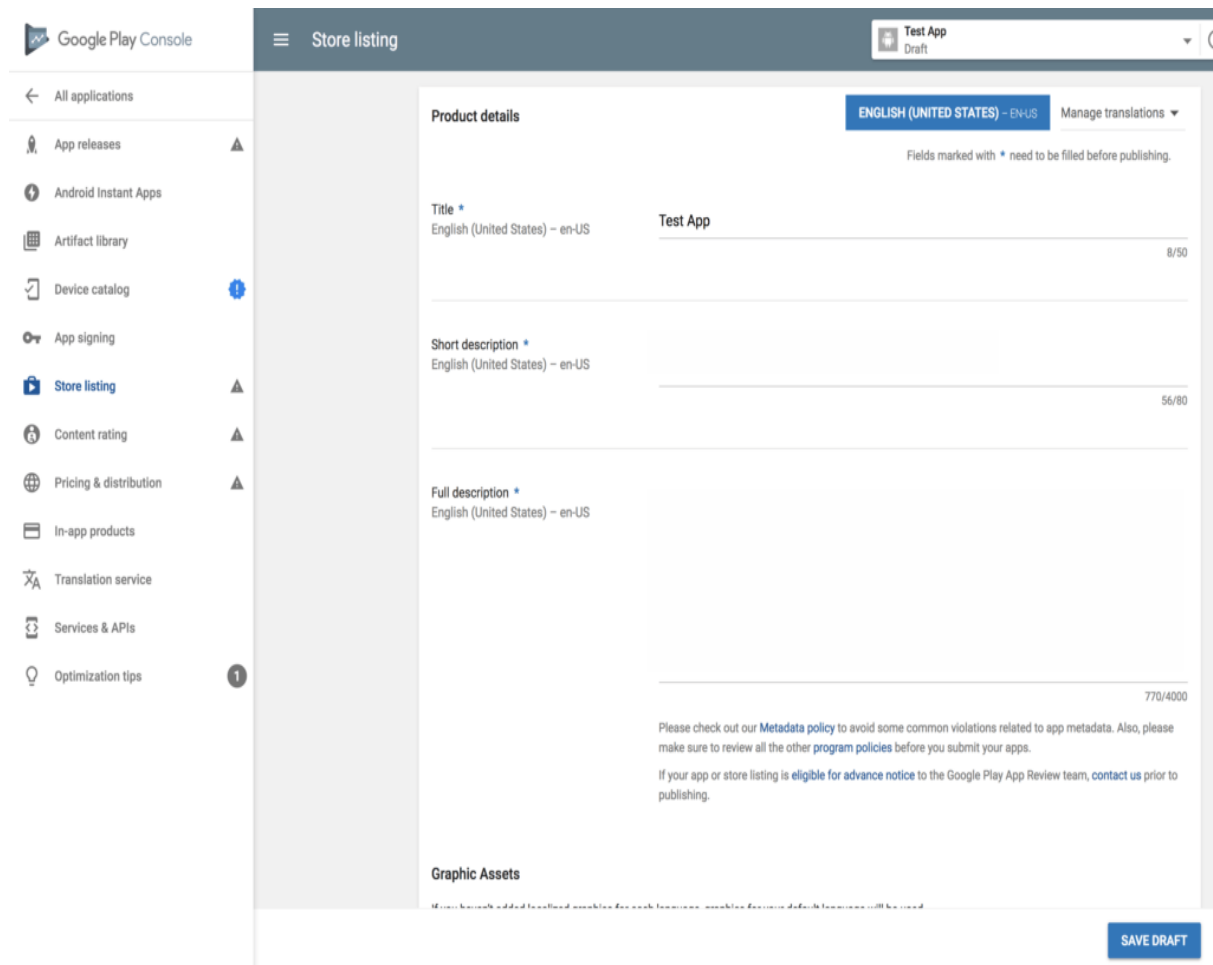


Lưu ý: Tiêu đề là phần hiển thị trên Google Play chứ không phải tên ứng dụng khi cài vào điện thoại, nên đưa từ khóa vào tiêu đề để tối ưu ASO.



Điền thông tin mô tả ứng dụng vào Short description và full description > Chọn “SAVE DRAFT.”

c. Upload logo và screenshot ứng dụng

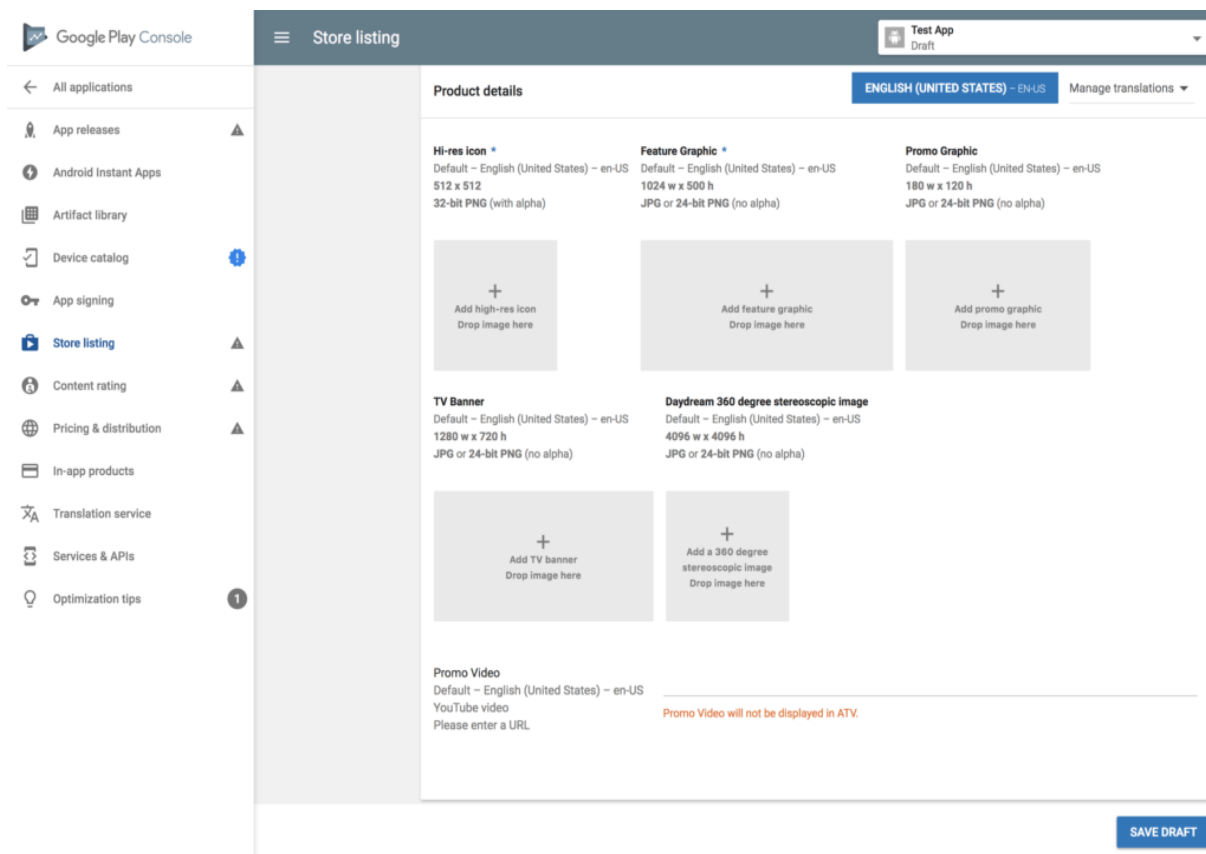
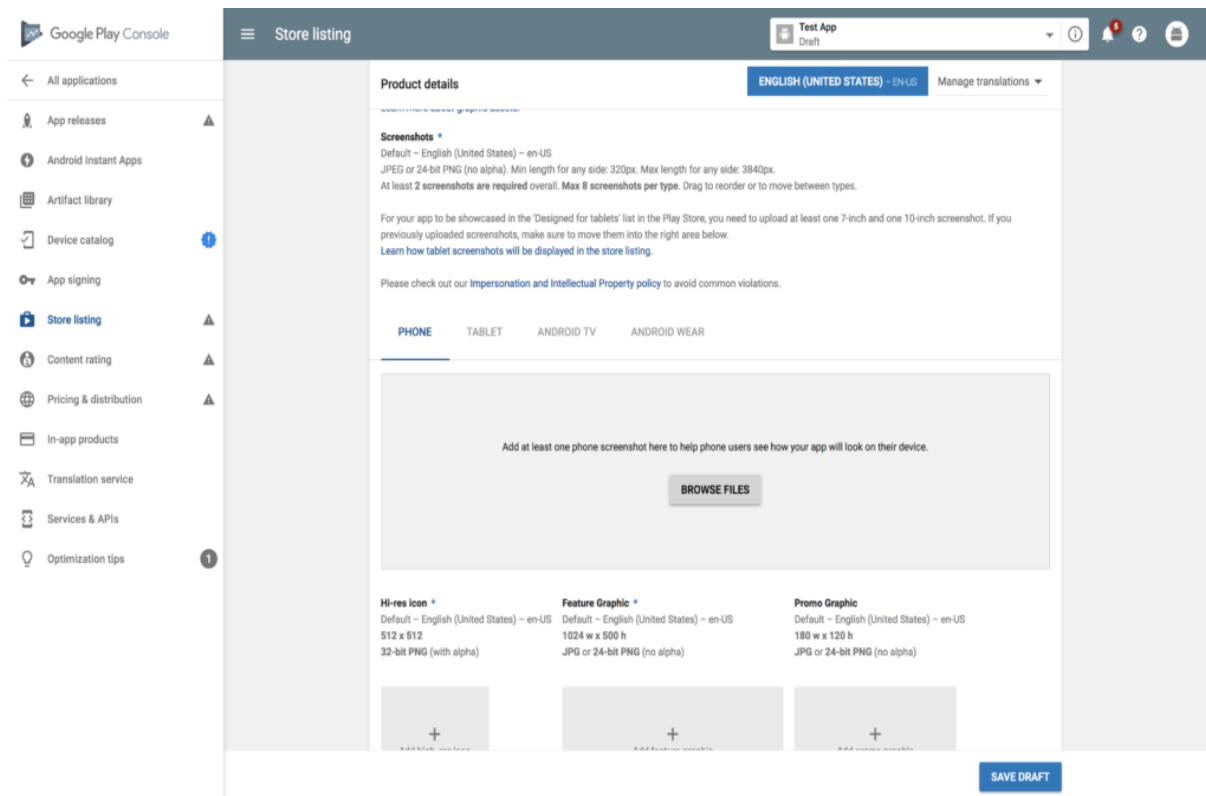


Phần này chúng ta sẽ cần upload ảnh logo, screenshot của ứng dụng.

Chọn “Add high-res icon”: tải logo lên (Lưu ý: kích thước logo bắt buộc là 512x512px)

Chọn “Add feature graphic”: tải ảnh promo được hiển thị trên đỉnh của trang ứng dụng trên Google play.

Chọn “SAVE DRAFT”



Chọn kiểu ứng dụng (App hay Game) > Chọn Category phù hợp với ứng dụng (Tool, Productivity, Entertainment...)

Google Play Console

Store listing

Test App Draft

← All applications

- App releases ▲
- Android Instant Apps
- Artifact library
- Device catalog ⚙️
- App signing
- Store listing** ▲
- Content rating ▲
- Pricing & distribution ▲
- In-app products
- Translation service
- Services & APIs
- Optimization tips 1

Categorization

Application type *
 Applications
 Games

Category *
 Select a category

Content rating *
 You need to fill a rating questionnaire and apply a content rating.

Contact details

Website
 https://www.example.com

Email *
 contact@example.com
 Please provide an email address where you may be contacted. This address will be publicly displayed with your app.

Phone
 +14165555555

SAVE DRAFT

Điều URL tới file privacy policy.

Google Play Console

Store listing

Test App Draft

← All applications

- App releases ▲
- Android Instant Apps
- Artifact library
- Device catalog ⚙️
- App signing
- Store listing** ▲
- Content rating ▲
- Pricing & distribution ▲
- In-app products
- Translation service
- Services & APIs
- Optimization tips 1

Contact details

Website
 http://www.example.com

Email *
 contact@example.com
 Please provide an email address where you may be contacted. This address will be publicly displayed with your app.

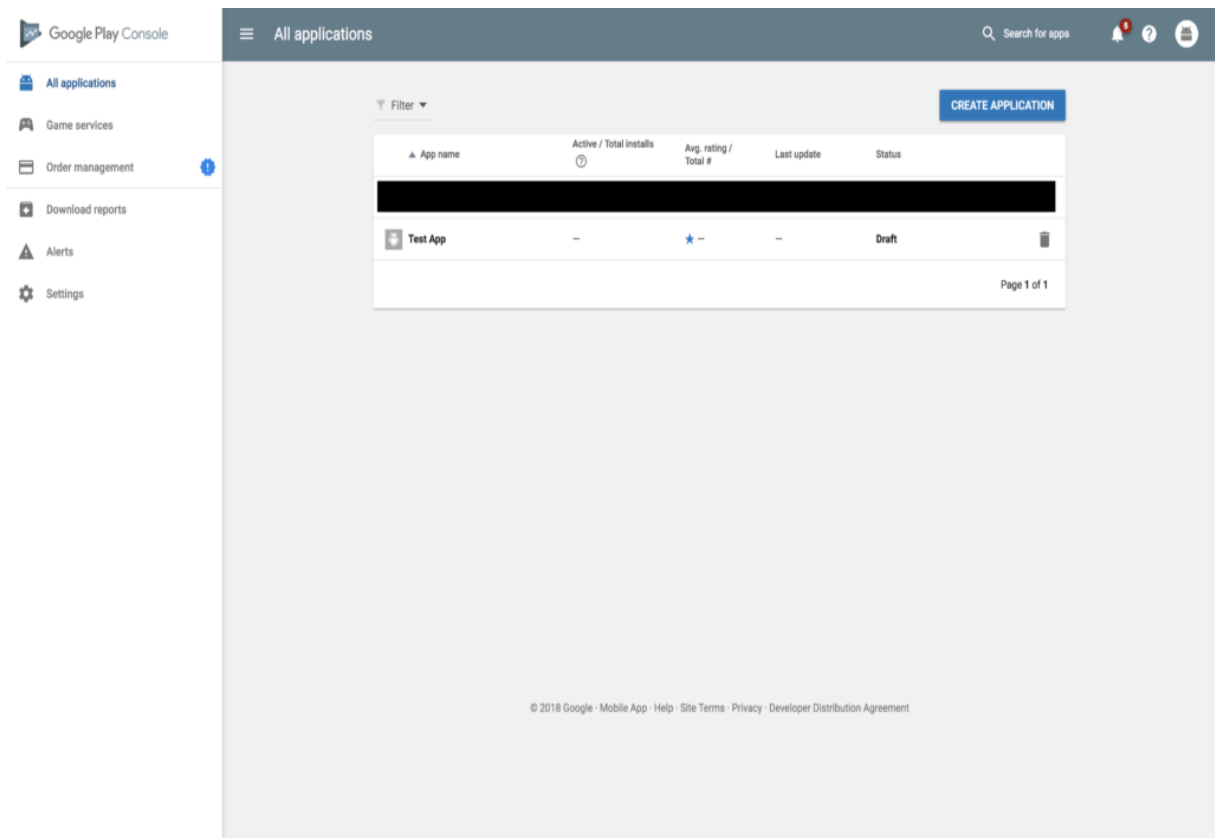
Phone
 555-555-5555

Privacy Policy *

If you wish to provide a privacy policy URL for this application, please enter it below. Also, please check out our [User Data policy](#) to avoid common violations.

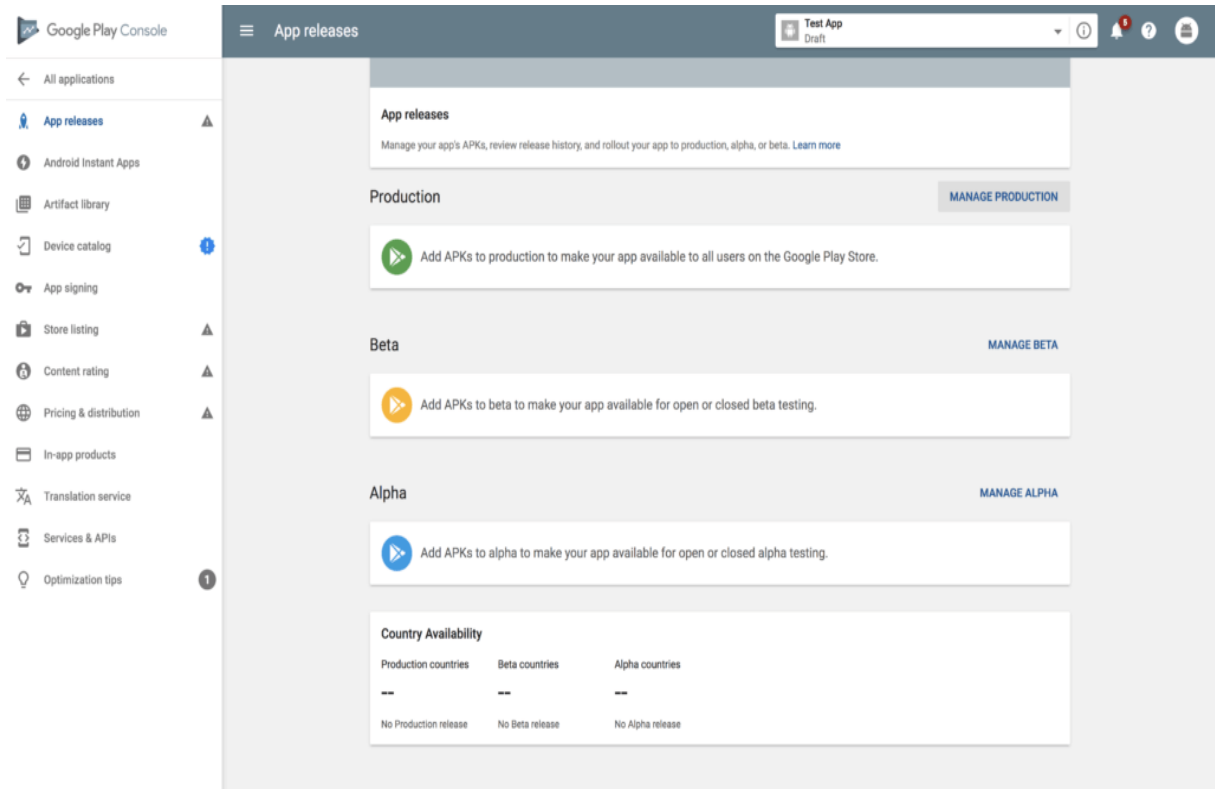
Privacy Policy
 http://www.example.com/privacy
 Not submitting a privacy policy URL at this time. [Learn more](#)

SAVE DRAFT

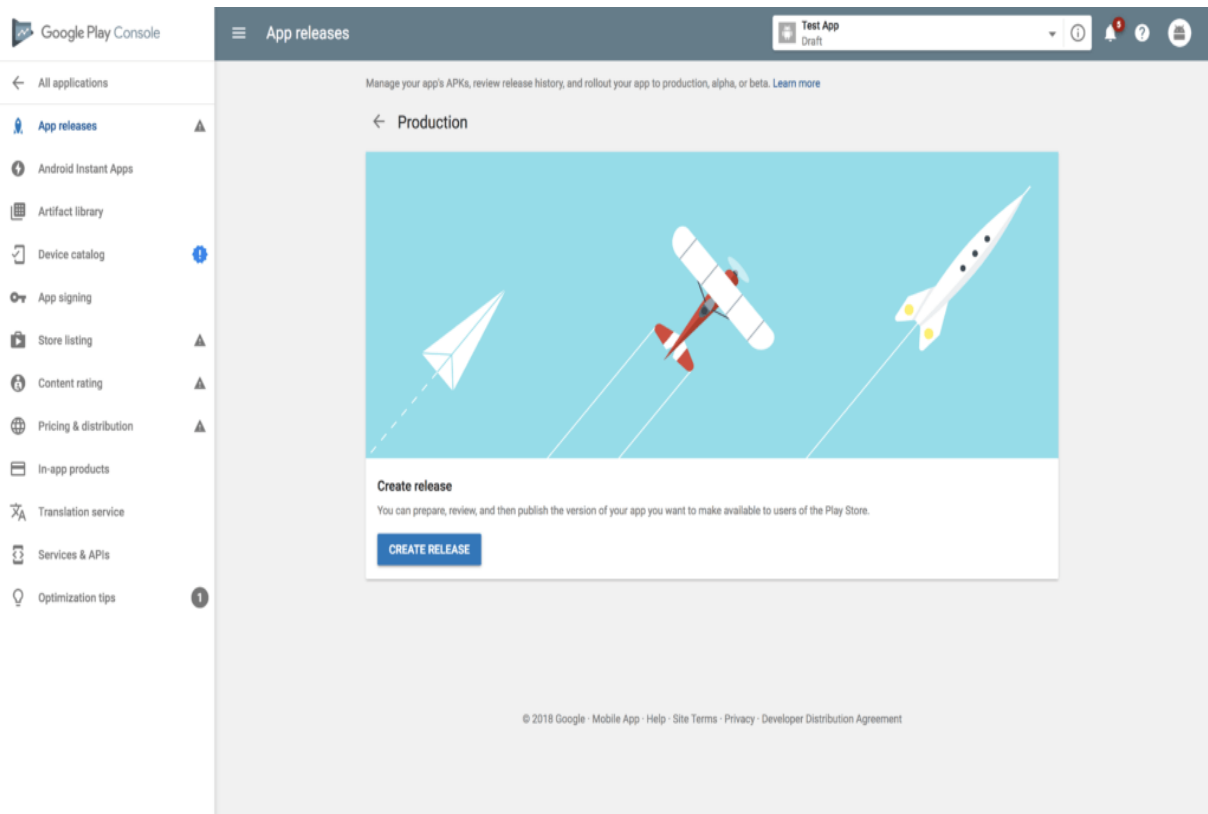


d. Tiến hành upload APK lên Store

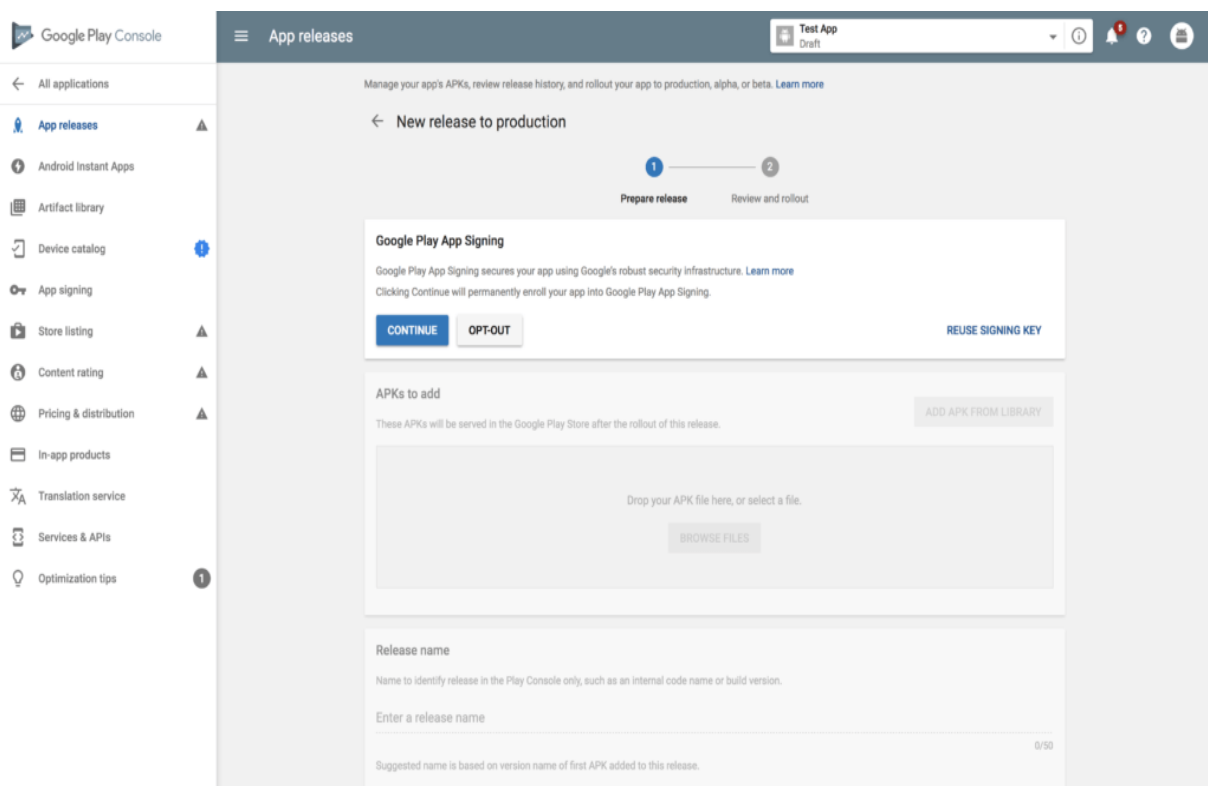
Chọn “MANAGE PRODUCTION”



Nhấn nút “CREATE RELEASE.”



Nếu đã tạo APK có sẵn sign key rồi thì chọn “opt out ”



Chọn “browse files” để upload APK từ máy tính lên

Google Play Console

App releases

Test App Draft

Manage your app's APKs, review release history, and rollout your app to production, alpha, or beta. [Learn more](#)

← New release to production

1 — 2
Prepare release — Review and rollout

APKs to add

These APKs will be served in the Google Play Store after the rollout of this release.

ADD APK FROM LIBRARY

Drop your APK file here, or select a file.

BROWSE FILES

Release name

Name to identify release in the Play Console only, such as an internal code name or build version.

Enter a release name

0/50

Suggested name is based on version name of first APK added to this release.

What's new in this release?

Google Play Console

App releases

Test App Draft

Manage your app's APKs, review release history, and rollout your app to production, alpha, or beta. [Learn more](#)

← New release to production

1 — 2
Prepare release — Review and rollout

APKs to add

These APKs will be served in the Google Play Store after the rollout of this release.

ADD APK FROM LIBRARY

Drop your APK file here, or select a file.

BROWSE FILES

Version code

98 REMOVE

Release name

Name to identify release in the Play Console only, such as an internal code name or build version.

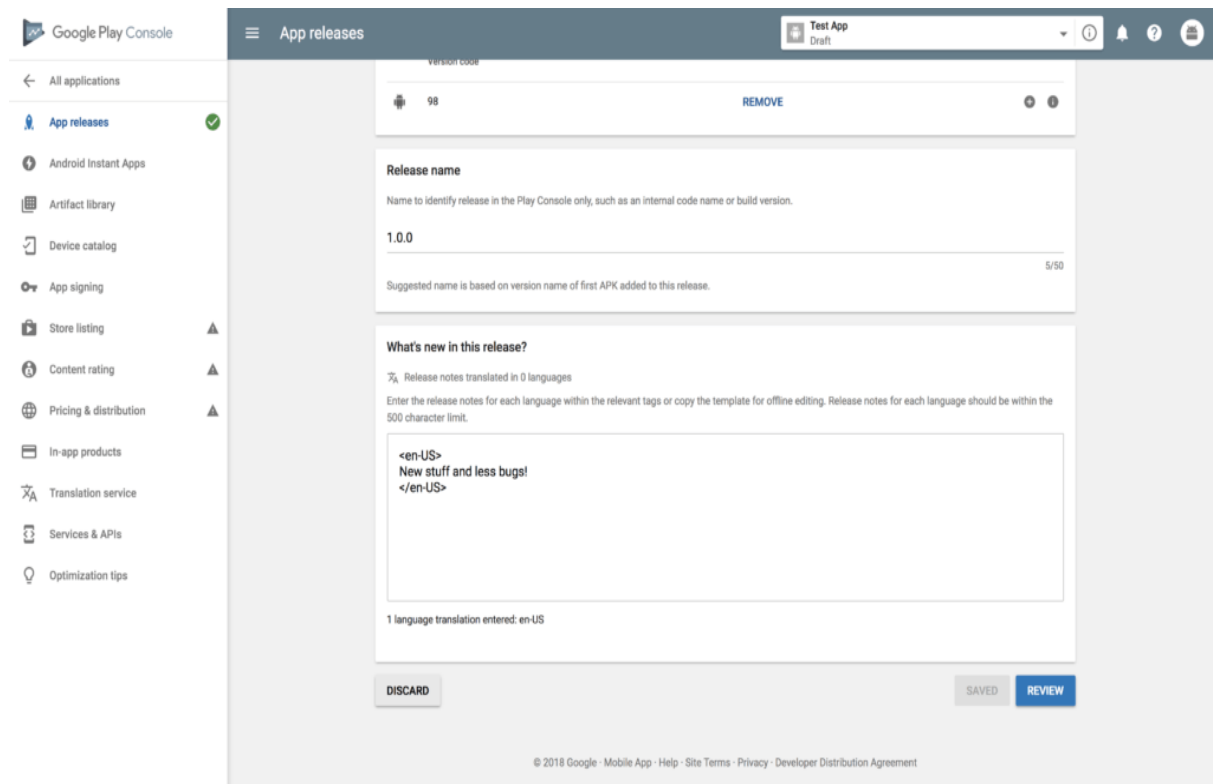
1.0.0

5/50

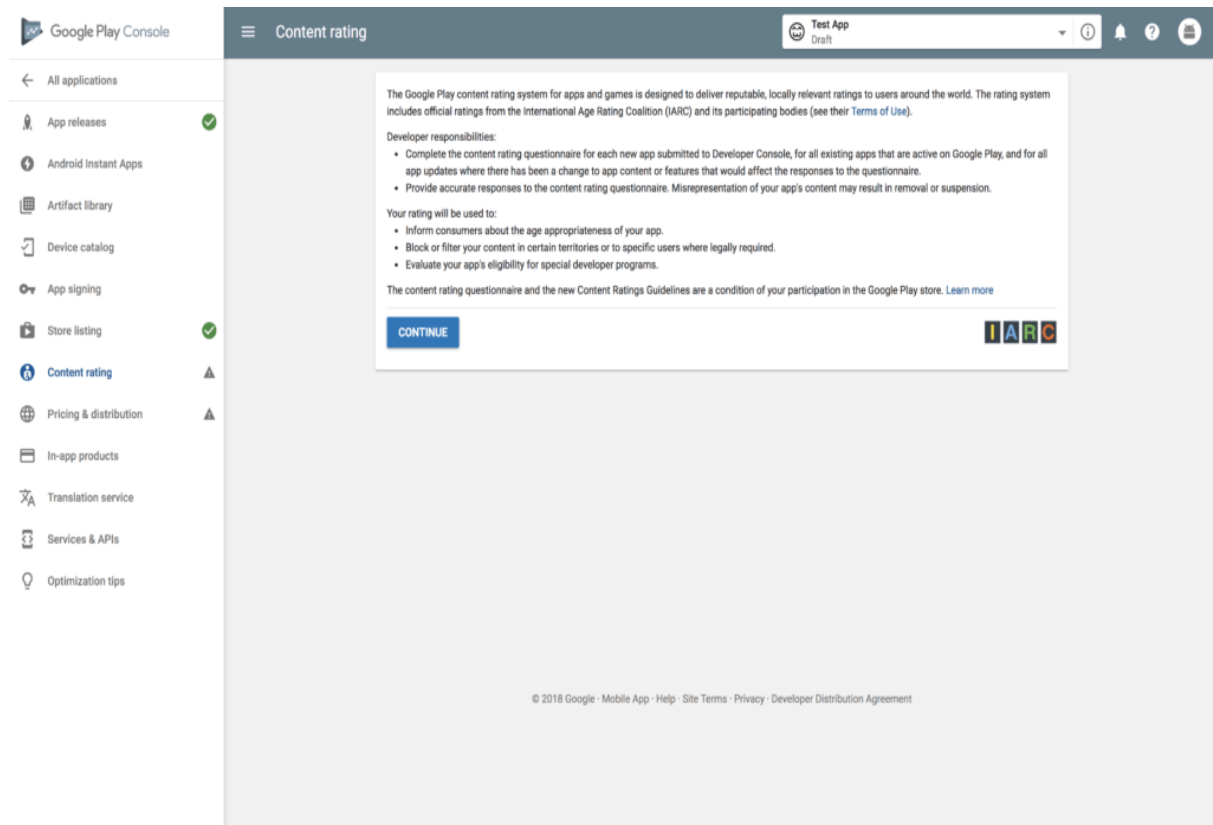
Suggested name is based on version name of first APK added to this release.

What's new in this release?

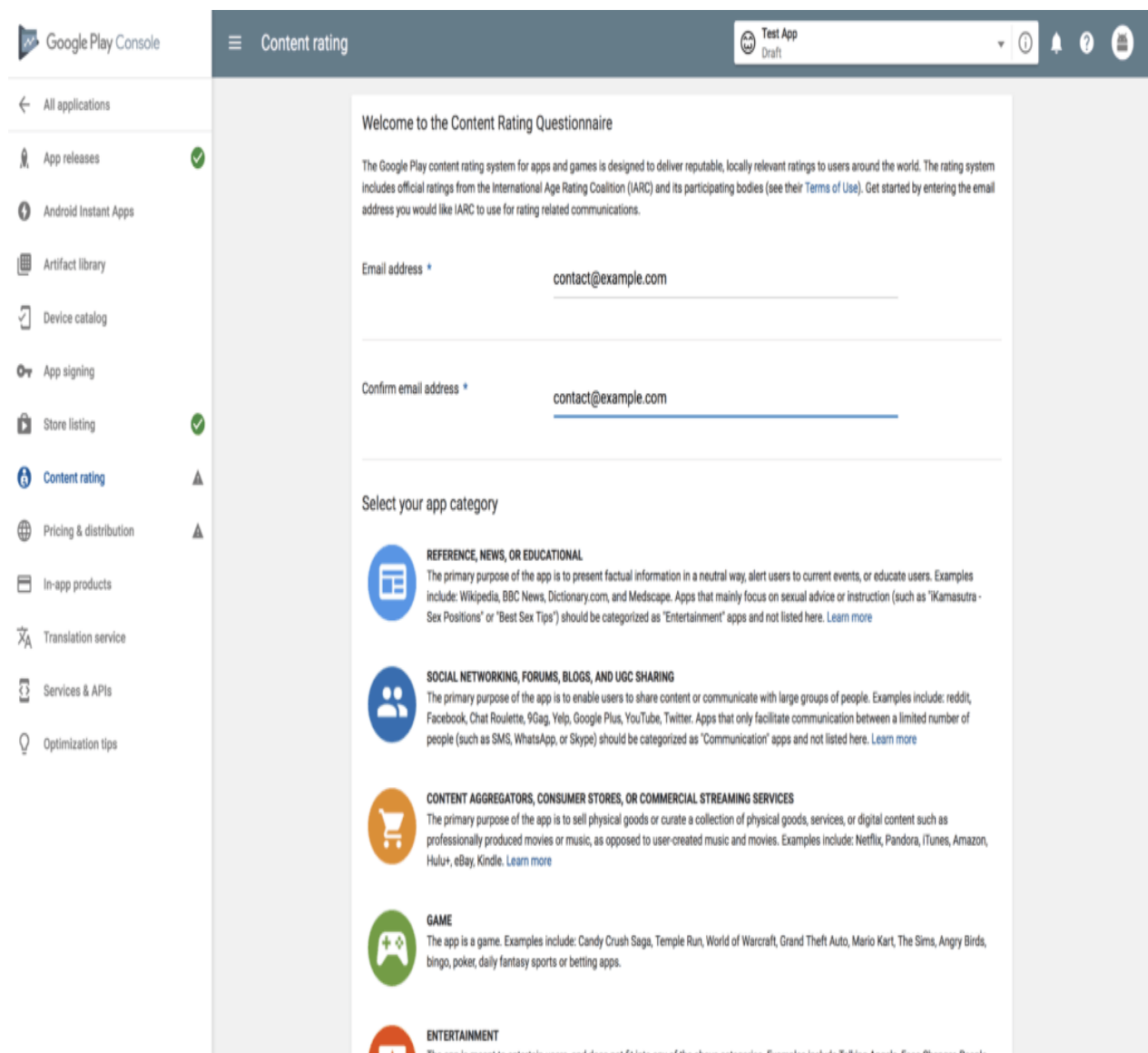
Điền các thông tin chính cho bản apk (Thường thì sẽ điền các tính năng mới mà cho lần upgrade ứng dụng) > nhấn “REVIEW.” > APK đã được upload lên.



e. Hoàn thành đánh giá Content Rating



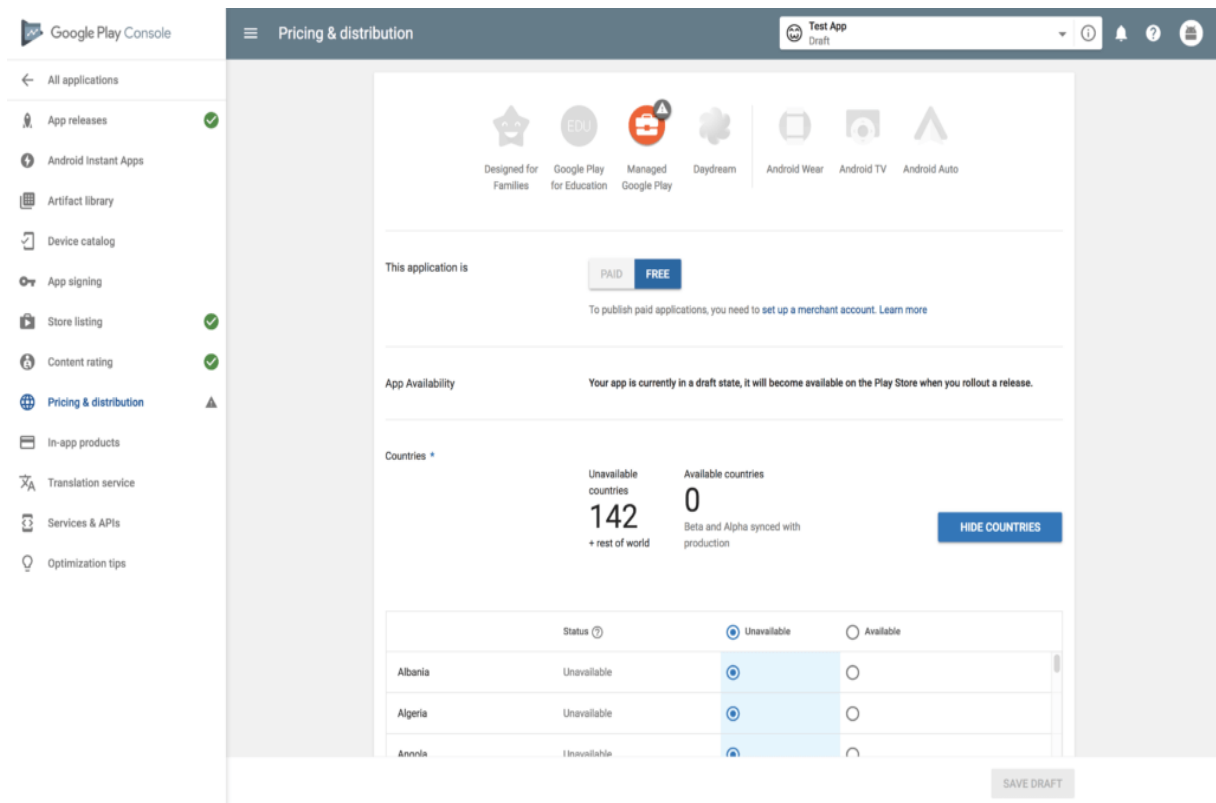
Phần content rating: trả lời các câu hỏi liên quan đến ứng dụng.



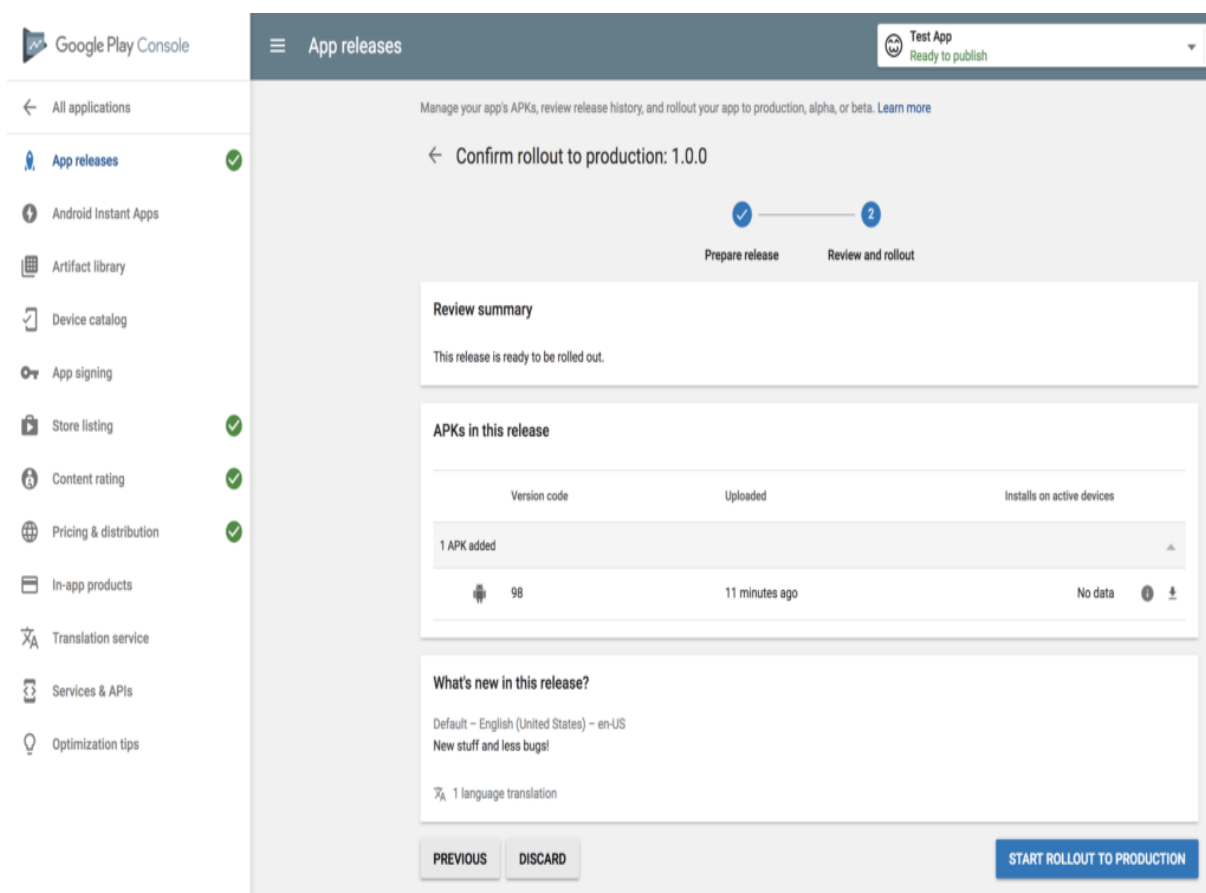
f. Đăng kí ứng dụng miễn phí hay trả phí

Chọn “Pricing & distribution” > chọn phát hành miễn phí hay có phí.

Lưu ý là: Khi đã chọn là ứng dụng miễn phí thì không thể chuyển thành ứng dụng trả phí được nữa. Nhưng ngược lại thì được.



Chọn “App releases ” > nhấn nút ” START ROLLOUT TO PRODUCTION ” chờ đợi Google review và approve cho ứng dụng.



3. TÓM TẮT TRÌNH TỰ THỰC HIỆN HOẶC QUY TRÌNH CÔNG NGHỆ

TT	Tên các bước công việc	Dụng cụ, thiết bị, vật tư	Yêu cầu kỹ thuật	Các chú ý về an toàn lao động
1	Khởi động Android Studio	Phần mềm Android Studio Máy tính.	Cài đặt chính xác phần mềm.	Đảm bảo an toàn cho người và thiết bị.
2	Thực hành thiết kế và code cho các nút lệnh	Phần mềm Android Studio Máy tính.	Cài đặt chính xác phần mềm.	Đảm bảo an toàn cho người và thiết bị.
3	Kiểm tra kết quả	Phần mềm Android Studio Máy tính.	Cài đặt chính xác phần mềm.	Đảm bảo an toàn cho người và thiết bị.

VI. HƯỚNG DẪN THỰC HIỆN MÔ ĐUN

1. Phạm vi áp dụng chương trình mô đun

- Modun được áp dụng cho sinh viên năm thứ 3 khoa Công nghệ thông tin, trường cao đẳng KTCN Việt Nam Hàn Quốc. Sau khi sinh viên đã được học các môn lập trình cơ sở, môn Cơ sở dữ liệu, môn Phân tích và thiết kế hệ thống.

2. Hướng dẫn một số điểm chính về phương pháp giảng dạy mô đun đào tạo

- Hình thành kỹ năng thực hành cho sinh viên các kiến thức cơ bản về ngôn ngữ lập trình Java và lập trình trên nền hệ điều hành Android. Sinh viên có thể vận dụng để tạo các ứng dụng đáp ứng yêu cầu thực tế chạy trên nền hệ điều hành Android.

3. Những trọng tâm chương trình mô đun cần chú ý

- Trình bày được các kiến thức cơ bản về ngôn ngữ lập trình Java và lập trình trên nền hệ điều hành Android.

- Vận dụng được để tạo các ứng dụng chạy trên thiết bị Android đáp ứng được các nhu cầu trong thực tế.

4. Tài liệu cần tham khảo

- Giáo trình lập trình Android – Lê Hoàng Sơn và Nguyễn Thọ Thông – Nhà xuất bản lao động – Xã hội

- Giáo trình Lập trình Android – Ths. Trương Thị Ngọc Phượng – Nhà xuất bản thời đại

- Giáo trình Andoid – PGS.TS Vũ Đức Thi – Nhà xuất bản thống kê