

**TRƯỜNG ĐẠI HỌC THỦY LỢI**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**GIÁO TRÌNH**  
**THỰC HÀNH PHÁT TRIỂN ỨNG DỤNG CHO THIẾT BỊ DI**  
**ĐỘNG**

Hà Nội, 2.2025

# MỤC LỤC

CHƯƠNG 1.	Làm quen .....	3
Bài 1)	Tạo ứng dụng đầu tiên.....	3
1.1)	Android Studio và Hello World .....	3
1.2)	Giao diện người dùng tương tác đầu tiên.....	18
1.3)	Trình chỉnh sửa bố cục .....	18
1.4)	Văn bản và các chế độ cuộn.....	18
1.5)	Tài nguyên có sẵn .....	18
Bài 2)	Activities .....	18
2.1)	Activity và Intent .....	18
2.2)	Vòng đời của Activity và trạng thái .....	18
2.3)	Intent ngầm định .....	18
Bài 3)	Kiểm thử, gỡ lỗi và sử dụng thư viện hỗ trợ.....	18
3.1)	Trình gỡ lỗi.....	18
3.2)	Kiểm thử đơn vị .....	18
3.3)	Thư viện hỗ trợ.....	18
CHƯƠNG 2.	Trải nghiệm người dùng .....	19
Bài 1)	Tương tác người dùng.....	19
1.1)	Hình ảnh có thể chọn .....	19
1.2)	Các điều khiển nhập liệu.....	19
1.3)	Menu và bộ chọn.....	19
1.4)	Điều hướng người dùng .....	19
1.5)	RecyclerView .....	19
Bài 2)	Trải nghiệm người dùng thú vị .....	19
2.1)	Hình vẽ, định kiểu và chủ đề.....	19
2.2)	Thẻ và màu sắc.....	19
2.3)	Bố cục thích ứng .....	19
Bài 3)	Kiểm thử giao diện người dùng .....	19

3.1) Espresso cho việc kiểm tra UI.....	19
CHƯƠNG 3. Làm việc trong nền .....	19
Bài 1) Các tác vụ nền .....	19
1.1) AsyncTask .....	19
1.2) AsyncTask và AsyncTaskLoader .....	19
1.3) Broadcast receivers .....	19
Bài 2) Kích hoạt, lập lịch và tối ưu hóa nhiệm vụ nền .....	19
2.1) Thông báo.....	19
2.2) Trình quản lý cảnh báo.....	19
2.3) JobScheduler .....	19
CHƯƠNG 4. Lưu dữ liệu người dùng .....	20
Bài 1) Tùy chọn và cài đặt .....	20
1.1) Shared preferences .....	20
1.2) Cài đặt ứng dụng .....	20
Bài 2) Lưu trữ dữ liệu với Room .....	20
2.1) Room, LiveData và ViewModel .....	20
2.2) Room, LiveData và ViewModel .....	20
3.1) Trình gowx lỗi .....	

## CHƯƠNG 1. LÀM QUEN

### Bài 1) Tạo ứng dụng đầu tiên

#### 1.1) Android Studio và Hello World

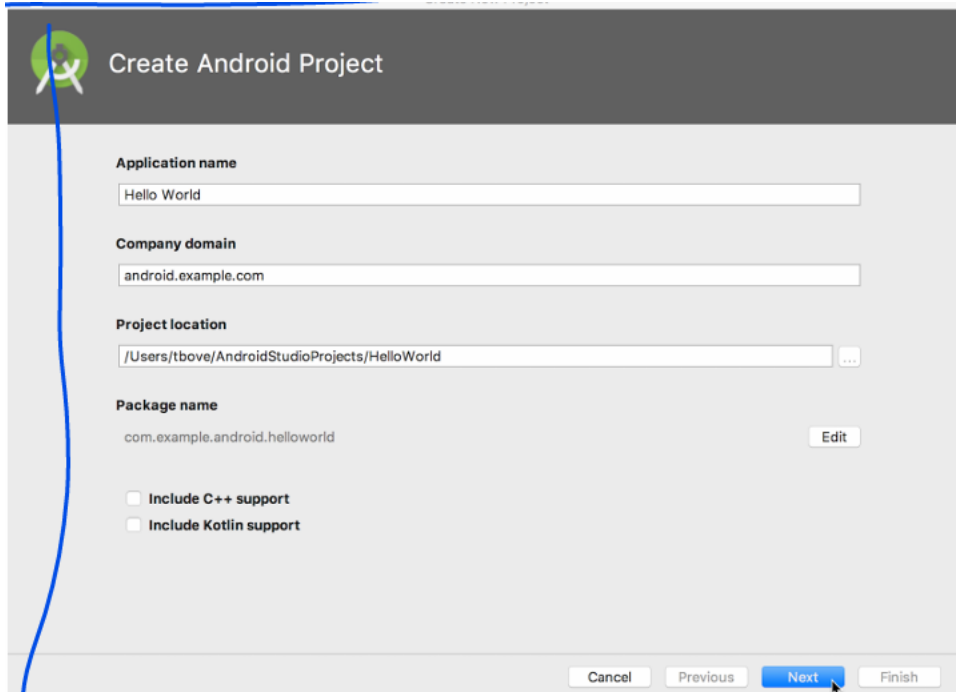
### Giới thiệu

Trong bài thực hành này, bạn sẽ tìm hiểu cách cài đặt Android Studio, môi trường phát triển Android. Bạn cũng sẽ tạo và chạy ứng dụng Android đầu tiên của mình, Hello World, trên một trình giả lập và trên một thiết bị vật lý.

### Những gì Bạn nên biết

Bạn nên có khả năng:

- Hiểu quy trình phát triển phần mềm tổng quát cho các ứng dụng lập trình hướng đối tượng sử dụng một IDE (môi trường phát triển tích hợp) như Android Studio.
- Chứng minh rằng bạn có ít nhất 1-3 năm kinh nghiệm trong lập trình hướng đối tượng, với một phần trong số đó tập trung vào ngôn ngữ lập trình Java. (Các bài thực hành này sẽ không giải thích về lập trình hướng đối tượng hoặc ngôn ngữ Java.



### Những gì Bạn sẽ cần:

- Một máy tính chạy Windows hoặc Linux, hoặc một Mac chạy macOS. Xem trang tải xuống Android Studio để biết yêu cầu hệ thống cập nhật.
- Truy cập Internet hoặc một phương pháp thay thế để tải các cài đặt mới nhất của Android Studio và Java lên máy tính của bạn.

### Những gì bạn sẽ học

- Cách cài đặt và sử dụng IDE Android Studio.
- Cách sử dụng quy trình phát triển để xây dựng ứng dụng Android.
- Cách tạo một dự án Android từ một mẫu.
- Cách thêm thông điệp ghi lại vào ứng dụng của bạn để phục vụ mục đích gỡ lỗi.

### Những gì bạn sẽ làm

- Cài đặt môi trường phát triển **Android Studio**.

- Tạo một trình giả lập (thiết bị ảo) để chạy ứng dụng của bạn trên máy tính.
- Tạo và chạy ứng dụng **Hello World** trên các thiết bị ảo và vật lý.
- Khám phá cấu trúc dự án.
- Tạo và xem các thông điệp ghi lại từ ứng dụng của bạn.
- Khám phá tệp **AndroidManifest.xml**

## Tổng quan về ứng dụng

Sau khi bạn cài đặt thành công Android Studio, bạn sẽ bắt đầu tạo một dự án mới cho ứng dụng "Hello World" dựa trên một mẫu có sẵn. Đây là một ứng dụng đơn giản, chỉ hiển thị dòng chữ “Hello World” trên màn hình của thiết bị Android, có thể là thiết bị ảo hoặc thật.

Dưới đây là hình ảnh minh họa ứng dụng khi hoàn thiện:



## Nhiệm vụ 1: Cài đặt Android Studio

Android Studio là một môi trường phát triển tích hợp (IDE) đầy đủ, cung cấp một trình chỉnh sửa mã nguồn tiên tiến cùng với bộ sưu tập các mẫu ứng dụng sẵn có. Ngoài ra, nó còn đi kèm các công cụ hỗ trợ lập trình, sửa lỗi, kiểm thử và tối ưu hiệu suất, giúp việc phát triển ứng dụng trở nên nhanh chóng và dễ dàng hơn. Bạn có thể thử nghiệm ứng dụng trên nhiều trình giả lập được cấu hình sẵn hoặc ngay

trên thiết bị di động của mình, sau đó xây dựng ứng dụng hoàn chỉnh và đưa lên Google Play Store.

**Lưu ý:** Android Studio thường xuyên được cập nhật và cải thiện. Để biết thông tin mới nhất về yêu cầu hệ thống và hướng dẫn cài đặt, hãy truy cập trang chính thức của Android Studio.

Để cài đặt Android Studio, bạn chỉ cần làm theo các bước đơn giản sau:

1. **Kiểm tra yêu cầu hệ thống:** Đảm bảo máy tính của bạn đáp ứng được yêu cầu cấu hình trước khi cài đặt. Android Studio hỗ trợ Windows, Linux và macOS.
2. **Tải và cài đặt:** Truy cập trang web chính thức của Android Developers, tải Android Studio và tiến hành cài đặt theo hướng dẫn.
3. **Chọn cài đặt mặc định:** Trong quá trình cài đặt, bạn chỉ cần giữ nguyên các tùy chọn mặc định và đảm bảo tất cả các thành phần cần thiết đều được chọn.
4. **Hoàn tất cài đặt và chờ tải thêm dữ liệu:** Sau khi cài đặt xong, Setup Wizard sẽ tiếp tục tải và cài đặt các thành phần bổ sung như Android SDK. Quá trình này có thể mất một chút thời gian tùy vào tốc độ mạng của bạn.
5. **Bắt đầu sử dụng:** Khi quá trình tải hoàn tất, Android Studio sẽ tự động khởi động, và bạn có thể bắt đầu tạo dự án đầu tiên của mình.

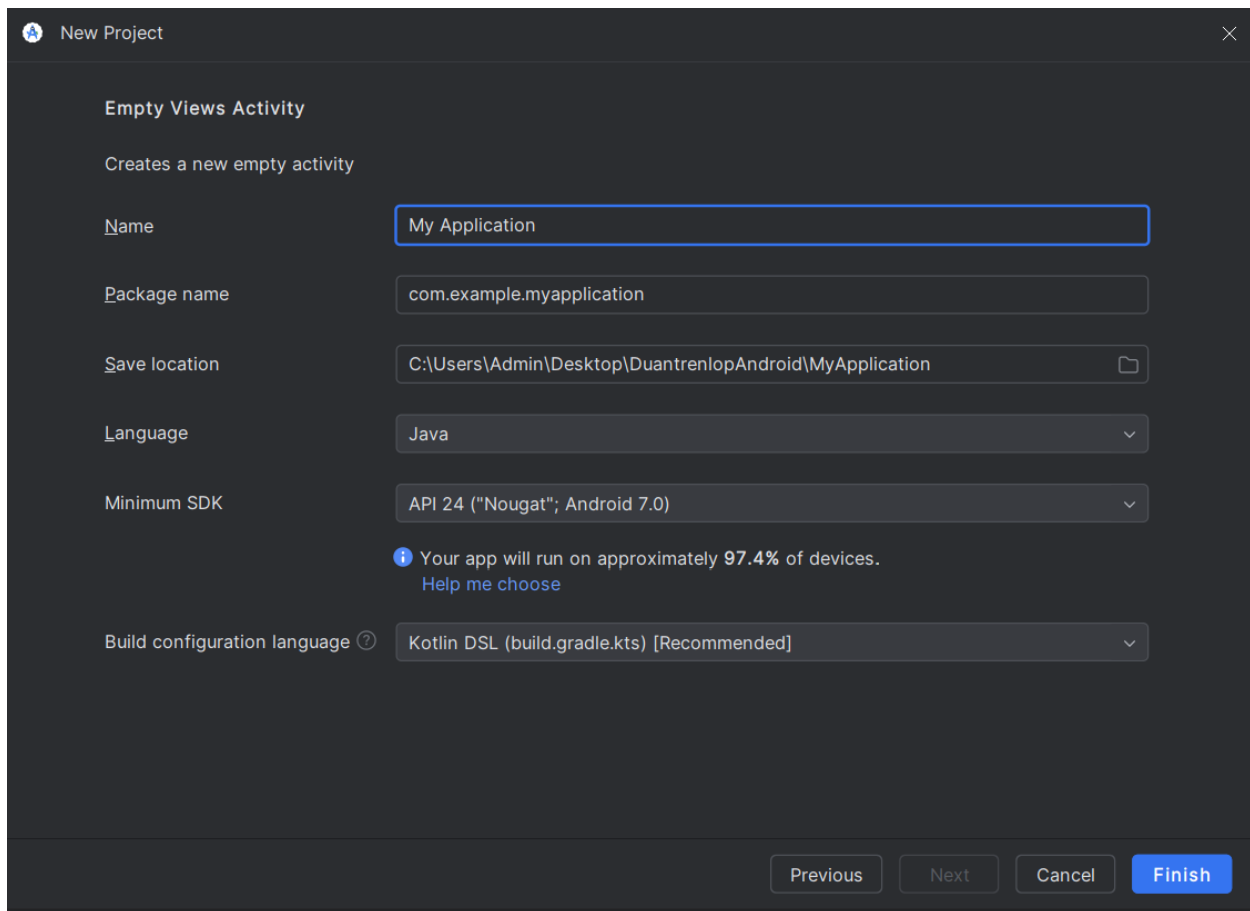
**Lưu ý:** Nếu gặp sự cố trong quá trình cài đặt, hãy kiểm tra [ghi chú phát hành của Android Studio](#) hoặc tìm sự trợ giúp từ giảng viên hoặc cộng đồng hỗ trợ.

## Nhiệm vụ 2: Tạo ứng dụng Hello World

Trong phần này, bạn sẽ tạo một ứng dụng đơn giản hiển thị dòng chữ "Hello World" để kiểm tra xem Android Studio đã được cài đặt đúng hay chưa, đồng thời làm quen với những bước cơ bản khi phát triển ứng dụng bằng Android Studio.

### 2.1 Tạo dự án ứng dụng

1. Mở Android Studio nếu bạn chưa mở sẵn.
2. Tại màn hình chính "Welcome to Android Studio", chọn "Start a new Android Studio project" để bắt đầu một dự án mới.
3. Trong cửa sổ "Create Android Project", đặt tên ứng dụng là "Hello World".

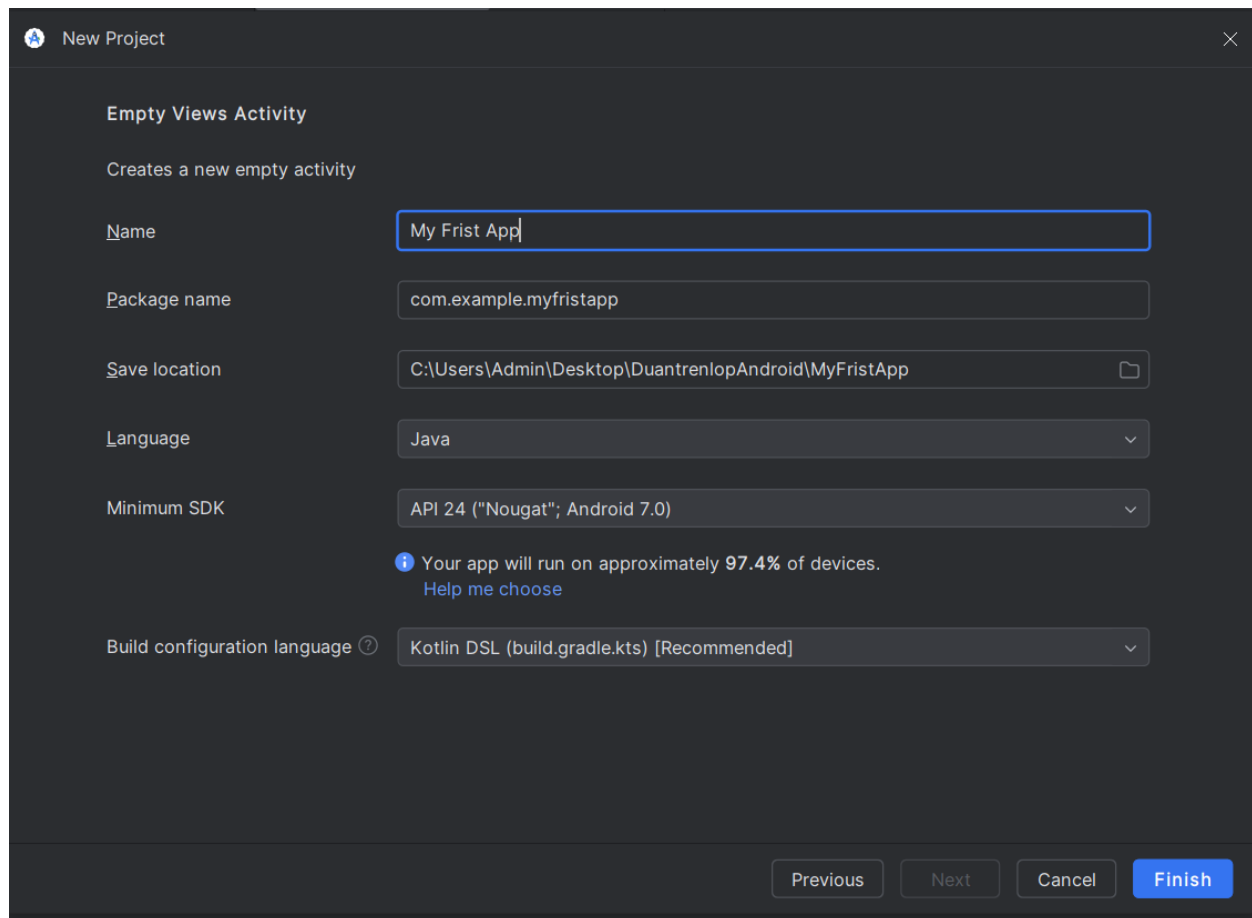


4. Kiểm tra xem thư mục mặc định trong phần "Project location" có phải là nơi bạn muốn lưu ứng dụng Hello World và các dự án Android Studio khác không. Nếu không, bạn có thể đổi sang thư mục mình thích.

5. Giữ nguyên "android.example.com" làm "Company Domain" mặc định, hoặc tự tạo một tên miền công ty độc đáo nếu muốn. Nếu bạn không có ý định phát hành ứng dụng, cứ để mặc định là được. Nhưng lưu ý rằng việc đổi tên gói (package name) sau này sẽ hơi mất công.

6. Bỏ chọn các tùy chọn "Include C++ support" và "Include Kotlin support", rồi nhấn "Next".

7. Ở màn hình "Target Android Devices", chọn "Phone and Tablet". Đảm bảo rằng "Minimum SDK" được đặt là "API 15: Android 4.0.3 IceCreamSandwich". Nếu chưa đúng, dùng menu thả xuống để chỉnh lại.

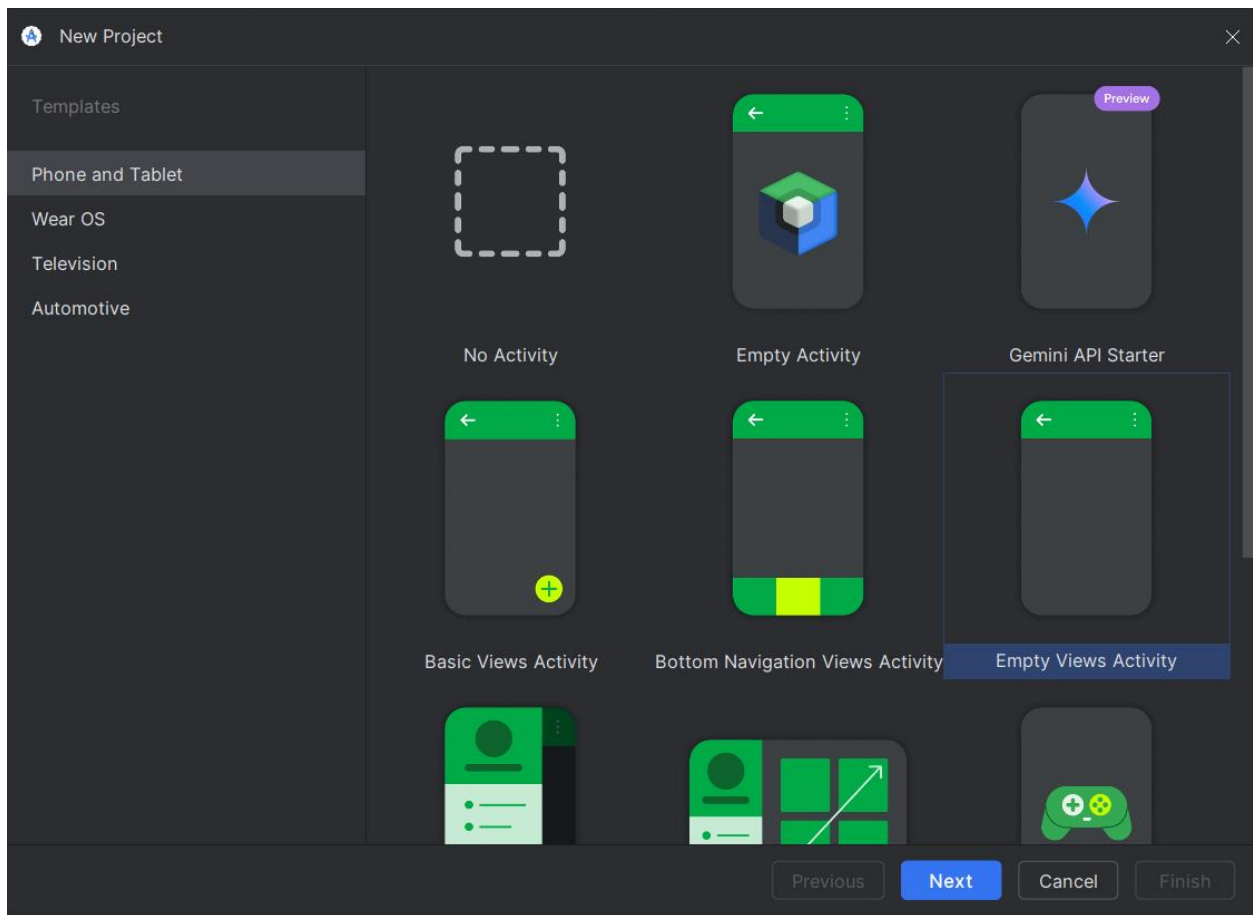


Những cài đặt này được sử dụng trong các ví dụ của khóa học này. Tính đến thời điểm hiện tại, với những cài đặt trên, ứng dụng Hello World của bạn sẽ tương thích với 97% thiết bị Android đang hoạt động trên Google Play Store.

8. Giữ nguyên, không chọn "Include Instant App support" hay bất kỳ tùy chọn nào khác, rồi nhấn "Next". Nếu dự án của bạn cần thêm thành phần nào cho SDK bạn đã chọn, Android Studio sẽ tự động cài đặt chúng.
9. Cửa sổ "Add an Activity" sẽ hiện ra. "Activity" là một chức năng cụ thể mà người dùng có thể thực hiện, và nó là một phần quan trọng của bất kỳ ứng dụng Android nào. Mỗi Activity thường đi kèm một bố cục (layout) để quyết định cách các yếu tố giao diện hiển thị trên màn hình. Android Studio cung cấp các mẫu Activity để bạn bắt đầu dễ dàng hơn. Đối với dự án Hello World, chọn "Empty Activity" như hình minh họa dưới đây, rồi nhấn "Next".



A



10. Màn hình "**Configure Activity**" sẽ xuất hiện (nội dung có thể khác nhau tùy vào mẫu bạn chọn ở bước trước). Theo mặc định, Activity trống từ mẫu sẽ được đặt tên là "MainActivity". Bạn có thể đổi tên nếu muốn, nhưng trong bài học này, chúng ta sẽ giữ nguyên tên "MainActivity".

11. Đảm bảo rằng tùy chọn "Generate Layout file" được chọn. Tên bố cục mặc định là "activity\_main". Bạn có thể đổi tên nếu thích, nhưng bài học này sẽ dùng "activity\_main".

12. Kiểm tra xem tùy chọn "Backwards Compatibility (App Compat)" đã được bật chưa. Điều này giúp ứng dụng của bạn tương thích ngược với các phiên bản Android cũ hơn.

13. Nhấn "Finish".

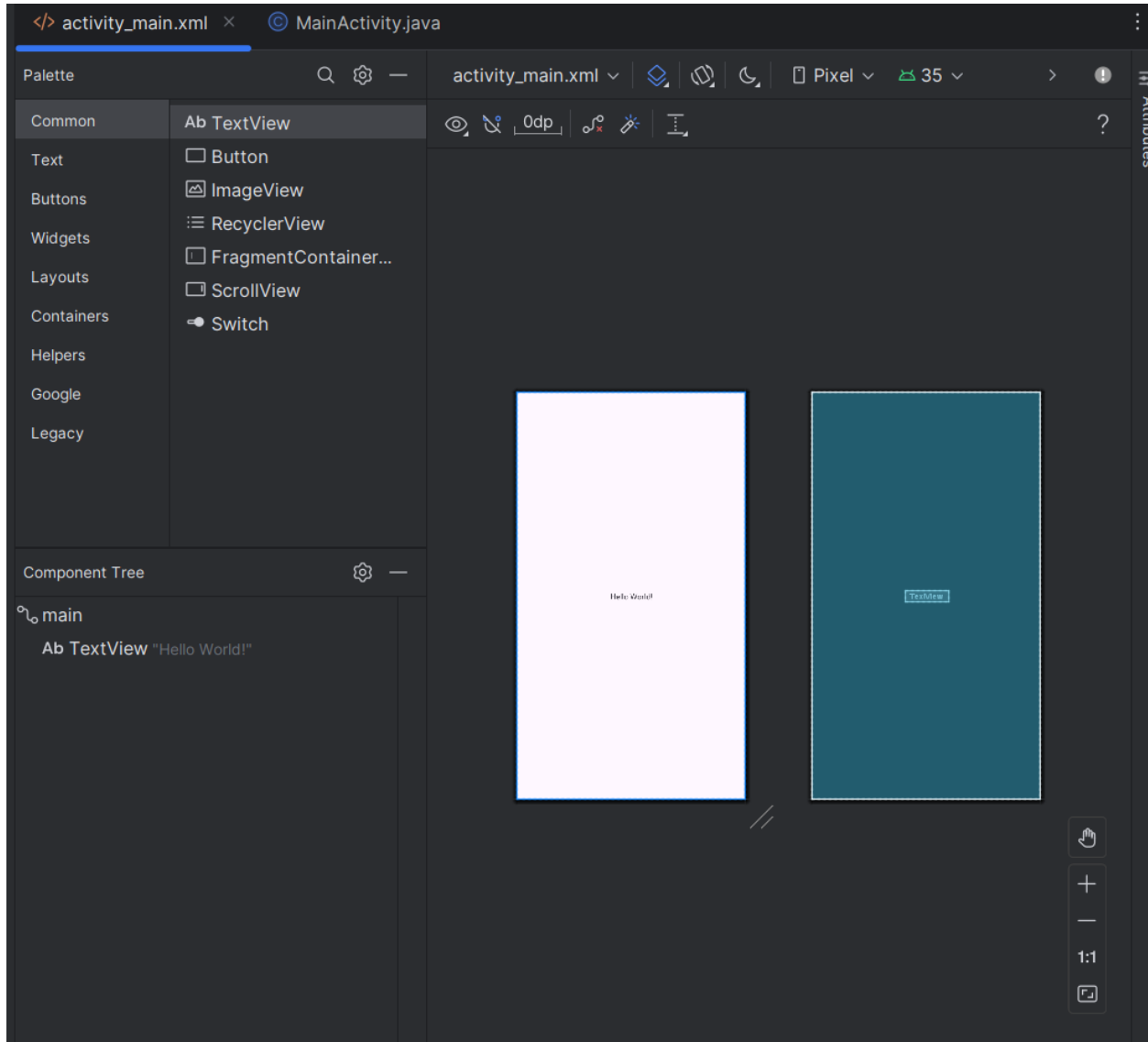
Android Studio sẽ tạo một thư mục cho dự án của bạn và xây dựng dự án bằng Gradle (có thể mất vài phút).

**Mẹo:** Xem trang "Configure your build" trên tài liệu dành cho nhà phát triển để biết thêm chi tiết.

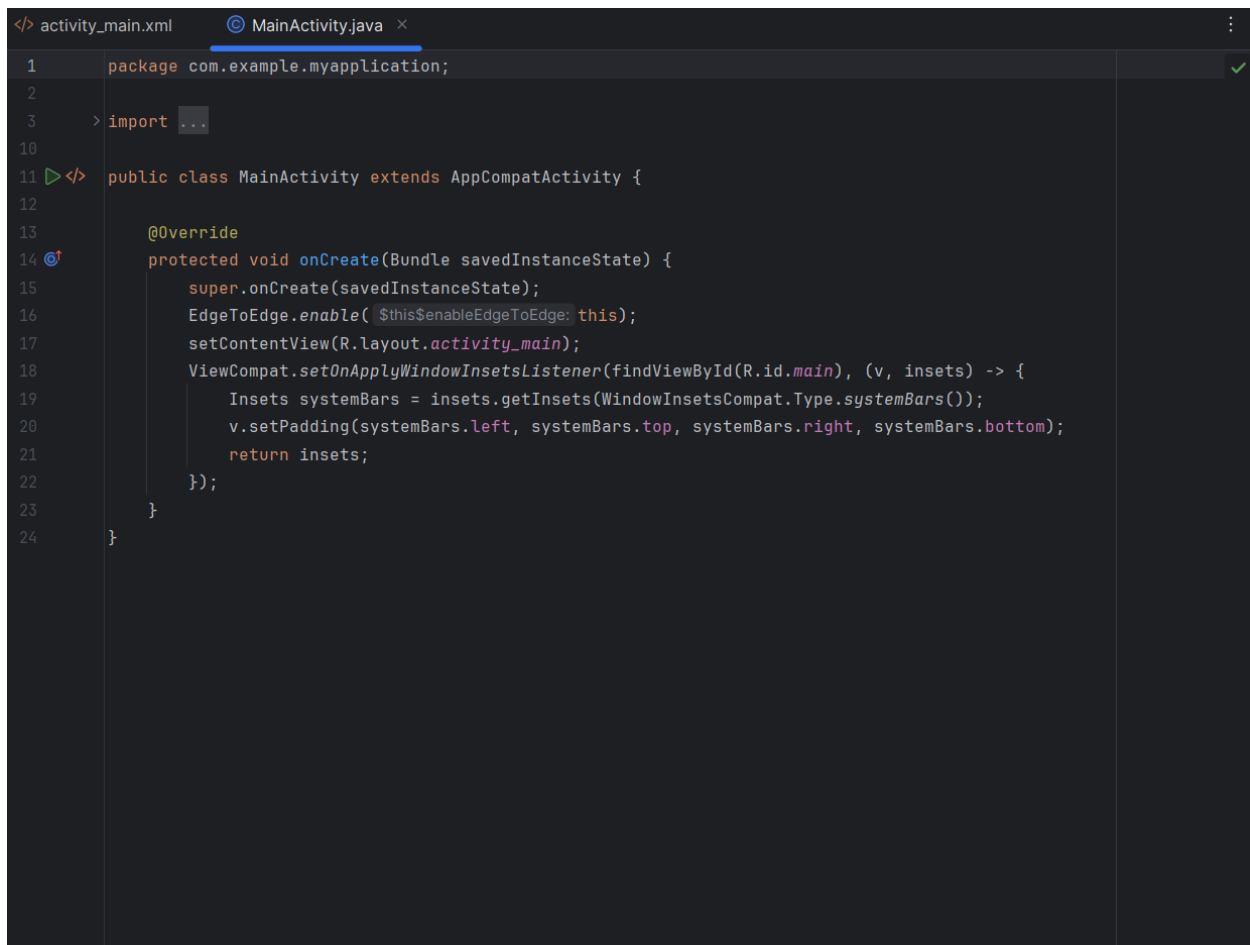
Bạn cũng có thể thấy một thông báo "Tip of the day" với các phím tắt và mẹo hữu ích. Nhấn "Close" để tắt thông báo đó.

Giao diện chỉnh sửa của Android Studio sẽ hiện ra. Làm theo các bước sau:

1. Nhấn vào tab "activity\_main.xml" để mở trình chỉnh sửa bố cục.
2. Chọn tab "Design" trong trình chỉnh sửa bố cục (nếu chưa được chọn) để xem giao diện đồ họa của bố cục như hình dưới đây.



3. Nhấn vào tab "MainActivity.java" để mở trình chỉnh sửa mã nguồn, như hình minh họa dưới đây.

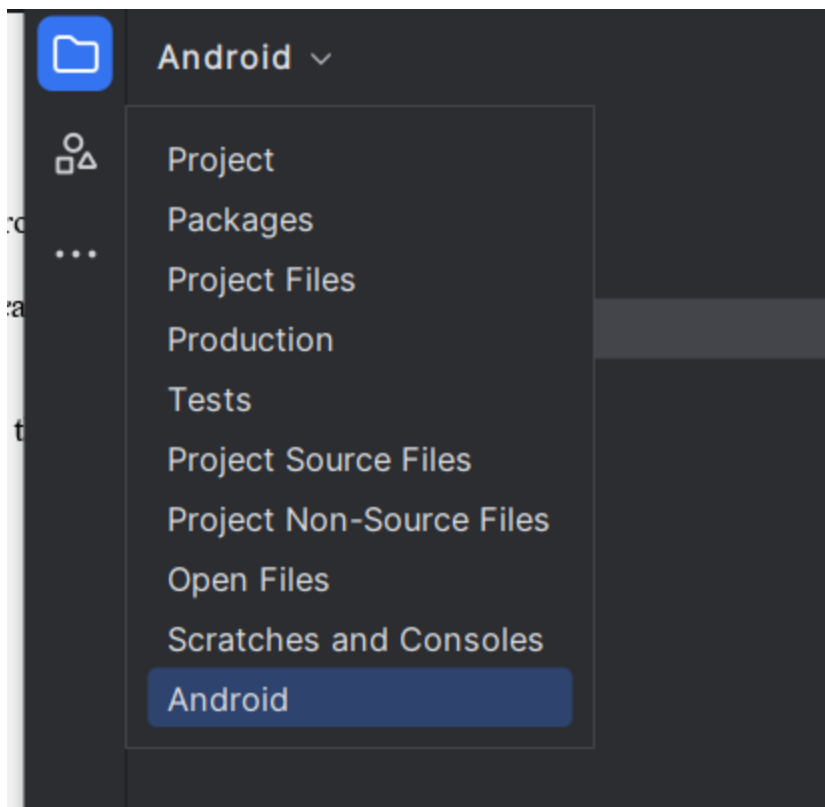


```
1 package com.example.myapplication;
2
3 > import ...
10
11 <> public class MainActivity extends AppCompatActivity {
12
13     @Override
14     protected void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         EdgeToEdge.enable(this);
17         setContentView(R.layout.activity_main);
18         ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
19             Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
20             v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
21             return insets;
22         });
23     }
24 }
```

## 2.2 Tìm hiểu ngăn Project > Android

Trong phần thực hành này, bạn sẽ khám phá cách dự án được sắp xếp trong Android Studio

1. Nếu chưa được chọn, hãy nhấp vào tab "Project" ở cột tab dọc bên trái của sổ Android Studio. Ngăn "Project" sẽ hiện ra.
2. Để xem dự án theo cấu trúc chuẩn của Android, chọn "Android" từ menu thả xuống ở đầu ngăn "Project", như hình minh họa dưới đây.

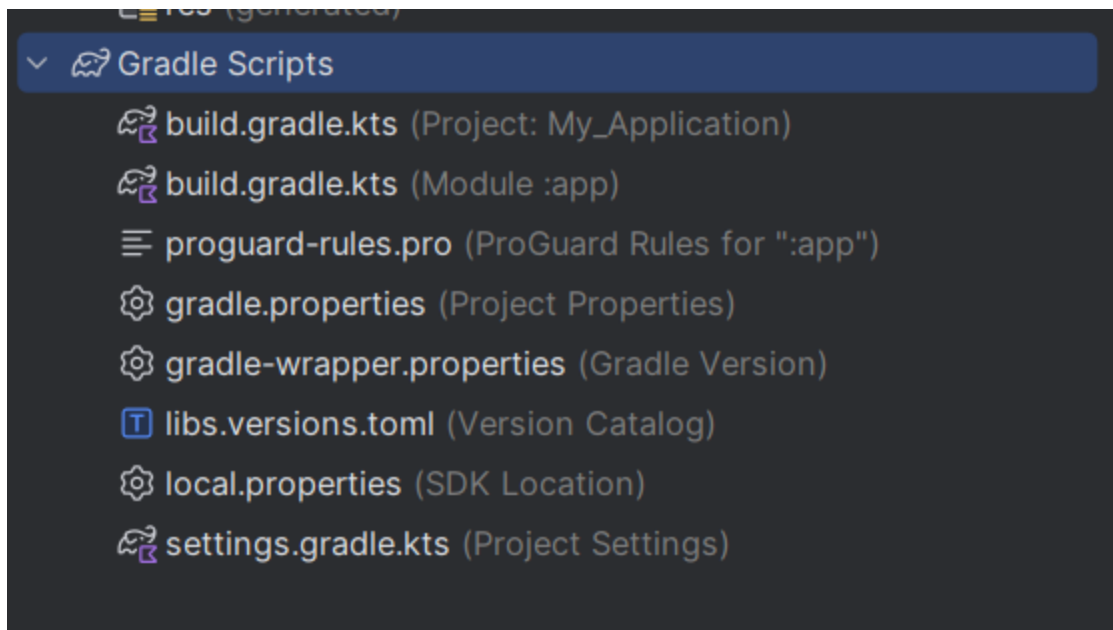


- **Lưu ý:** Chương này và các chương khác sẽ gọi ngắn "Project" khi được đặt ở chế độ "Android" là ngắn "Project > Android".

## 2.3 Khám phá thư mục Gradle Scripts

Hệ thống xây dựng Gradle trong Android Studio giúp bạn dễ dàng thêm các tệp nhị phân bên ngoài hoặc các mô-đun thư viện khác vào dự án của mình dưới dạng phụ thuộc (dependencies).

Khi bạn lần đầu tạo một dự án ứng dụng, ngắn "Project > Android" sẽ hiển thị với thư mục "Gradle Scripts" được mở rộng như hình



Làm theo các bước sau để tìm hiểu về hệ thống Gradle:

1. Nếu thư mục "Gradle Scripts" chưa được mở rộng, nhấp vào dấu tam giác để mở nó ra.

Thư mục này chứa tất cả các tệp cần thiết cho hệ thống xây dựng dự án.

2. Tìm tệp "build.gradle (Project: HelloWorld)".  
Đây là nơi bạn sẽ thấy các tùy chọn cấu hình chung cho tất cả các mô-đun trong dự án của mình. Mỗi dự án Android Studio đều có một tệp Gradle cấp cao duy nhất. Thông thường, bạn không cần chỉnh sửa tệp này, nhưng hiểu nội dung của nó vẫn rất hữu ích.  
Theo mặc định, tệp Gradle cấp cao sử dụng khối "buildscript" để xác định các kho lưu trữ (repositories) và phụ thuộc (dependencies) chung cho tất cả các mô-đun trong dự án. Khi phụ thuộc của bạn không phải là thư viện cục bộ hay cây tệp, Gradle sẽ tìm kiếm các tệp trong những kho lưu trữ trực tuyến được chỉ định trong khối "repositories" của tệp này. Mặc định, các dự án mới trong Android Studio sử dụng JCenter và Google (bao gồm kho Google Maven) làm vị trí kho lưu trữ

```
You can use the Project Structure dialog to view and edit your project configuration  Open (Ctrl+Alt+Shift+S)  Hide notification
1  // Top-level build file where you can add configuration options common to all sub-projects/modules
2  plugins {
3      alias(libs.plugins.android.application) apply false
4  }
```

### 3. Tìm tệp "build.gradle(Module.app)".

Ngoài tệp build.gradle cấp dự án, mỗi mô-đun trong dự án cũng có một tệp build.gradle riêng, cho phép bạn tùy chỉnh cài đặt xây dựng cho từng mô-đun cụ thể (ứng dụng HelloWorld chỉ có một mô-đun). Việc chỉnh cấu hình cài đặt này giúp bạn tạo ra các tùy chọn đóng gói riêng, như thêm các loại xây dựng (build types) hoặc biến thể sản phẩm (product flavors). Bạn cũng có thể ghi đè các cài đặt trong tệp AndroidManifest.xml hoặc tệp build.gradle cấp cao hơn.

Tệp này thường là nơi bạn sẽ chỉnh sửa khi cần thay đổi cấu hình cấp ứng dụng, chẳng hạn như khai báo các phụ thuộc trong phần "dependencies". Bạn có thể thêm một thư viện phụ thuộc bằng cách sử dụng một trong nhiều cấu hình phụ thuộc khác nhau. Mỗi cấu hình sẽ cho Gradle biết cách sử dụng một trong nhiều cấu hình phụ thuộc khác nhau. Mỗi cấu hình sẽ cho Gradle biết cách sử dụng thư viện đó. Ví dụ, dòng lệnh "implementation fileTree(dir: 'libs', include: ['\*.jar'])" sẽ thêm tất cả các tệp ".jar" trong thư mục "libs" làm phụ thuộc.

Dưới đây là nội dung của tệp "build.gradle (Module: app)" cho ứng dụng HelloWorld:

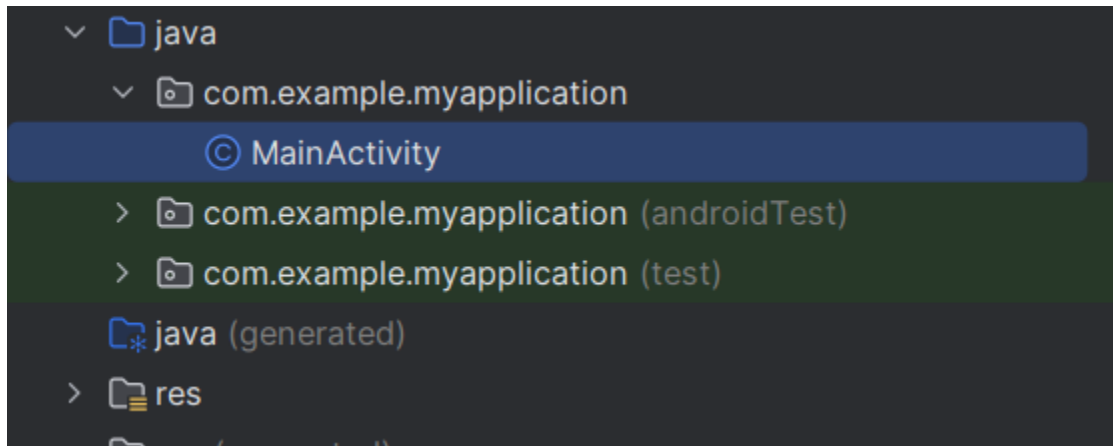
```
1 plugins {
2     alias(libs.plugins.android.application)
3 }
4
5 android {
6     namespace = "com.example.myapplication"
7     compileSdk = 35
8
9     defaultConfig {
10         applicationId = "com.example.myapplication"
11         minSdk = 24
12         targetSdk = 35
13         versionCode = 1
14         versionName = "1.0"
15
16         testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"
17     }
18
19     buildTypes {
20         release {
21             isMinifyEnabled = false
22             proguardFiles(
23                 getDefaultProguardFile("proguard-android-optimize.txt"),
24                 "proguard-rules.pro"
25             )
26         }
27     }
28     compileOptions {
29         sourceCompatibility = JavaVersion.VERSION_11
30         targetCompatibility = JavaVersion.VERSION_11
31     }
32 }
33
34 dependencies {
35     implementation(libs.appcompat)
36     implementation(libs.material)
37     implementation(libs.activity)
38     implementation(libs.constraintlayout)
39     testImplementation(libs.junit)
40     androidTestImplementation(libs.ext.junit)
41     androidTestImplementation(libs.espresso.core)
42 }
43 }
```

4. Nhấp vào tam giác để đóng các tập lệnh Gradle.

## 2.4 Tìm hiểu thư mục app và res

Tất cả mã nguồn và tài nguyên của ứng dụng đều nằm trong hai thư mục "app" và "res".

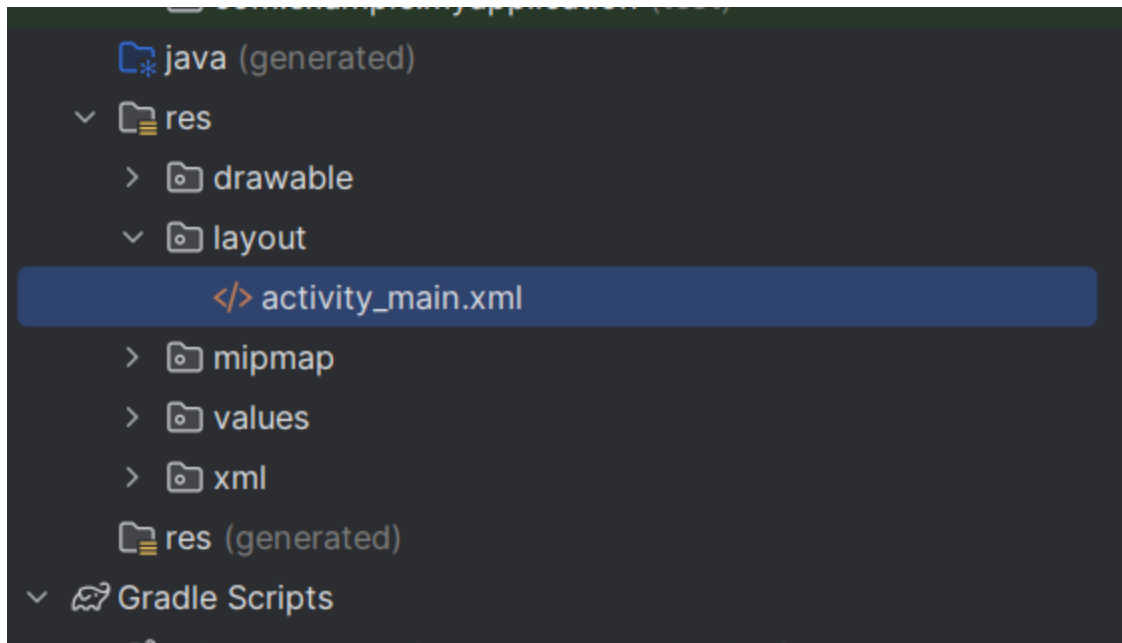
1. Mở rộng thư mục "app", sau đó đến thư mục "java", rồi tiếp tục mở thư mục "com.example.android.helloworld" để thấy tệp "MainActivity.java". Nhấp đúp vào tệp này để mở nó trong trình chỉnh sửa mã nguồn.



Thư mục "java" chứa các tệp lớp Java, được chia thành ba thư mục con, như hình minh họa phía trên. Thư mục "com.example.hello.helloworld" (hoặc tên miền bạn đã chỉ định) chứa tất cả các tệp thuộc gói ứng dụng. Hai thư mục còn lại dùng để kiểm thử và sẽ được giải thích trong bài học khác. Với ứng dụng Hello World, chỉ có một gói duy nhất, và trong đó chứa tệp "MainActivity.java". Tên của Activity đầu tiên (màn hình mà người dùng thấy khi mở ứng dụng), đồng thời khởi tạo các tài nguyên chung cho toàn ứng dụng, thường được đặt là "MainActivity" (phần mở rộng tệp không hiển thị trong ngăn "Project > Android").

2. Mở rộng thư mục "res", sau đó đến thư mục "layout", rồi nhấp đúp vào tệp "activity\_main.xml" để mở nó trong trình chỉnh sửa bố cục.





Thư mục "res" chứa các tài nguyên như bố cục, chuỗi văn bản và hình ảnh. Một Activity thường đi kèm với một bố cục giao diện người dùng được định nghĩa trong tệp XML, và tệp này thường được đặt tên theo Activity tương ứng.

## 2.5 Tìm hiểu thư mục manifests

Thư mục "manifests" chứa các tệp cung cấp thông tin quan trọng về ứng dụng của bạn cho hệ điều hành Android. Hệ thống cần những thông tin này trước khi có thể chạy bất kỳ mã nào của ứng dụng.

1. Mở rộng thư mục "manifests".
2. Mở tệp "AndroidManifest.xml".

Tệp "AndroidManifest.xml" mô tả tất cả các thành phần của ứng dụng Android. Mọi thành phần, như mỗi Activity, đều phải được khai báo trong tệp XML này. Trong các bài học khác của khóa học, bạn sẽ chỉnh sửa tệp này để thêm tính năng và quyền truy cập. Để tìm hiểu thêm, xem phần "Tổng quan về App Manifest".

**1.2) Giao diện người dùng tương tác đầu tiên**

**1.3) Trình chỉnh sửa bố cục**

**1.4) Văn bản và các chế độ cuộn**

**1.5) Tài nguyên có sẵn**

## **Bài 2) Activities**

**2.1) Activity và Intent**

**2.2) Vòng đời của Activity và trạng thái**

**2.3) Intent ngầm định**

## **Bài 3) Kiểm thử, gỡ lỗi và sử dụng thư viện hỗ trợ**

**3.1) Trình gỡ lỗi**

**3.2) Kiểm thử đơn vị**

**3.3) Thư viện hỗ trợ**

## **CHƯƠNG 2. TRẢI NGHIỆM NGƯỜI DÙNG**

### **Bài 1) Tương tác người dùng**

- 1.1) Hình ảnh có thể chọn**
- 1.2) Các điều khiển nhập liệu**
- 1.3) Menu và bộ chọn**
- 1.4) Điều hướng người dùng**
- 1.5) RecyclerView**

### **Bài 2) Trải nghiệm người dùng thú vị**

- 2.1) Hình vẽ, định kiểu và chủ đề**
- 2.2) Thẻ và màu sắc**
- 2.3) Bố cục thích ứng**

### **Bài 3) Kiểm thử giao diện người dùng**

- 3.1) Espresso cho việc kiểm tra UI**

## **CHƯƠNG 3. LÀM VIỆC TRONG NỀN**

### **Bài 1) Các tác vụ nền**

- 1.1) AsyncTask**
- 1.2) AsyncTask và AsyncTaskLoader**
- 1.3) Broadcast receivers**

### **Bài 2) Kích hoạt, lập lịch và tối ưu hóa nhiệm vụ nền**

- 2.1) Thông báo**
- 2.2) Trình quản lý cảnh báo**
- 2.3) JobScheduler**

## **CHƯƠNG 4. LƯU DỮ LIỆU NGƯỜI DÙNG**

### **Bài 1) Tùy chọn và cài đặt**

**1.1) Shared preferences**

**1.2) Cài đặt ứng dụng**

### **Bài 2) Lưu trữ dữ liệu với Room**

**2.1) Room, LiveData và ViewModel**

**2.2) Room, LiveData và ViewModel**