

**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 .....	28
1.3) Trình chỉnh sửa bố cục .....	42
1.4) Văn bản và các chế độ cuộn .....	42
1.5) Tài nguyên có sẵn.....	42
Bài 2) Activities .....	42
2.1) Activity và Intent .....	42
2.2) Vòng đời của Activity và trạng thái .....	42
2.3) Intent ngầm định.....	42
Bài 3) Kiểm thử, gỡ lỗi và sử dụng thư viện hỗ trợ .....	42
3.1) Trình gỡ lỗi .....	42
3.2) Kiểm thử đơn vị .....	42
3.3) Thư viện hỗ trợ .....	42
CHƯƠNG 2. Trải nghiệm người dùng .....	43
Bài 1) Tương tác người dùng .....	43
1.1) Hình ảnh có thể chọn .....	43
1.2) Các điều khiển nhập liệu .....	43
1.3) Menu và bộ chọn .....	43
1.4) Điều hướng người dùng .....	43
1.5) RecyclerView .....	43
Bài 2) Trải nghiệm người dùng thú vị.....	43
2.1) Hình vẽ, định kiểu và chủ đề .....	43
2.2) Thẻ và màu sắc .....	43

2.3)	Bố cục thích ứng.....	43
Bài 3)	Kiểm thử giao diện người dùng.....	43
3.1)	Espresso cho việc kiểm tra UI .....	43
CHƯƠNG 3. Làm việc trong nền .....		43
Bài 1)	Các tác vụ nền.....	43
1.1)	AsyncTask .....	43
1.2)	AsyncTask và AsyncTaskLoader .....	43
1.3)	Broadcast receivers .....	43
Bài 2)	Kích hoạt, lập lịch và tối ưu hóa nhiệm vụ nền.....	43
2.1)	Thông báo .....	43
2.2)	Trình quản lý cảnh báo .....	43
2.3)	JobScheduler.....	43
CHƯƠNG 4. Lưu dữ liệu người dùng .....		44
Bài 1)	Tùy chọn và cài đặt.....	44
1.1)	Shared preferences.....	44
1.2)	Cài đặt ứng dụng.....	44
Bài 2)	Lưu trữ dữ liệu với Room .....	44
2.1)	Room, LiveData và ViewModel.....	44
2.2)	Room, LiveData và ViewModel.....	44
3.1)	Trình gowx loi .....	

## CHƯƠNG 1. LÀM QUEN

### Bài 1) Tạo ứng dụng đầu tiên

#### 1.1) Android Studio và Hello World

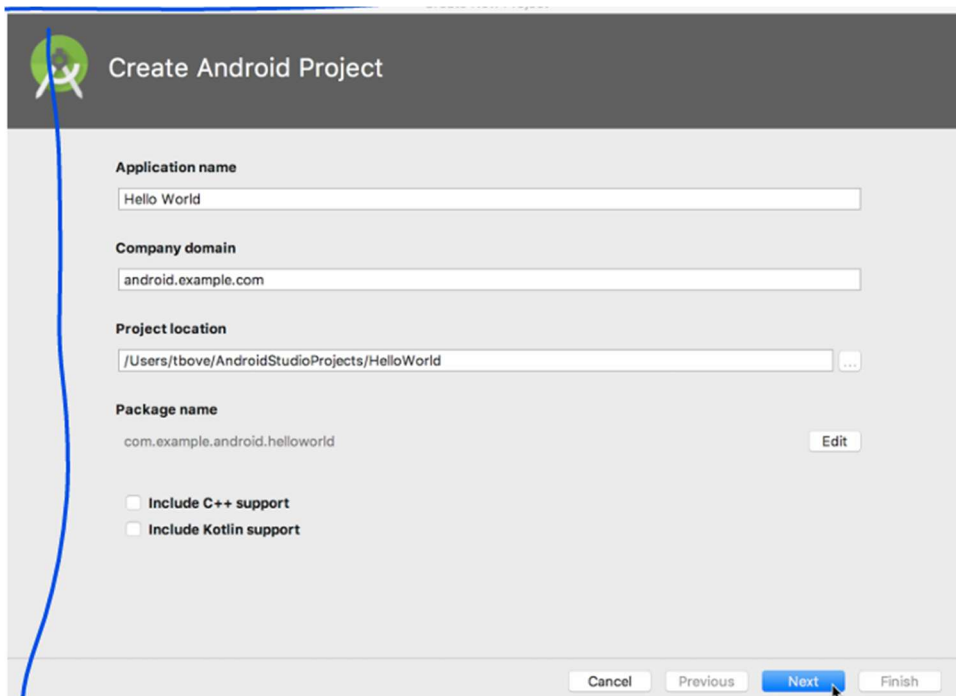
### Giới thiệu

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.

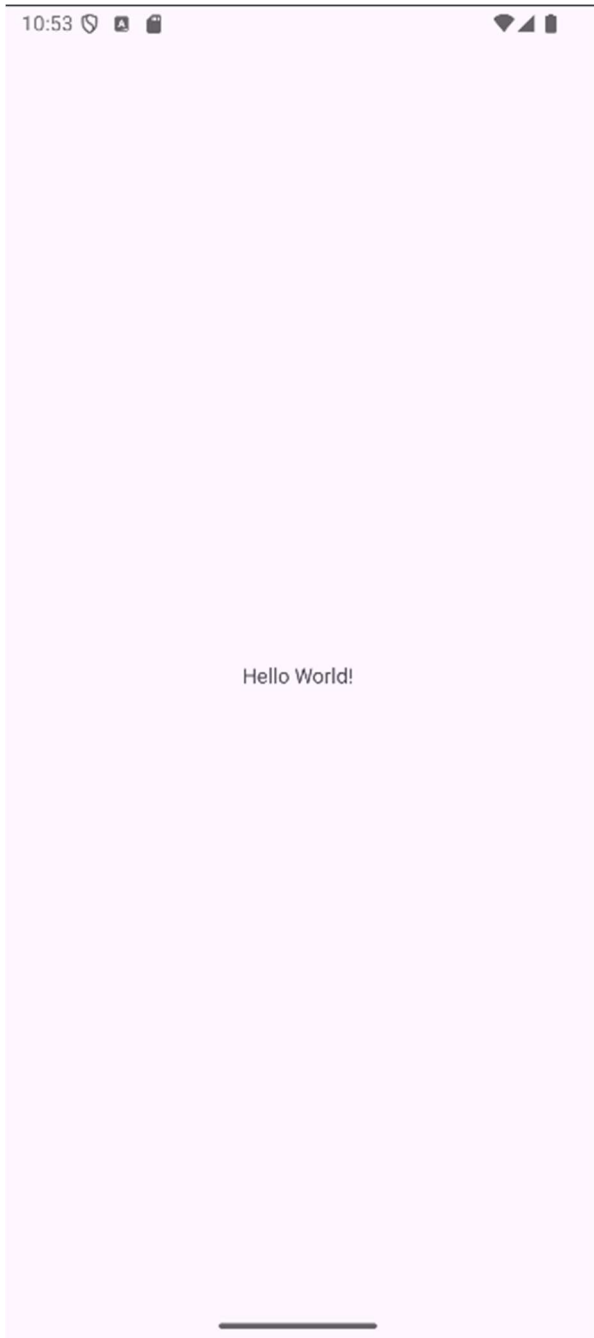
## **Những gì bạn sẽ làm**

- 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 tạo một dự án mới cho ứng dụng Hello World từ một mẫu có sẵn. Ứng dụng đơn giản này hiển thị chuỗi “Hello World” trên màn hình của thiết bị Android ảo hoặc thật

Ở đây là giao diện của ứng dụng hoàn thiện:



### **Nhiệm vụ 1: Cài đặt Android Studio**

Android Studio cung cấp một môi trường phát triển tích hợp (IDE) hoàn chỉnh, bao gồm trình soạn thảo mã tiên tiến và một bộ mẫu ứng dụng. Ngoài ra, nó còn chứa các công cụ để phát triển, gỡ lỗi, kiểm thử và tối ưu hiệu suất, giúp việc phát triển ứng dụng nhanh chóng và dễ dàng hơn. Bạn có thể kiểm tra ứng dụng của mình bằng một loạt các trình giả lập được cấu hình sẵn hoặc trên thiết bị di động của riêng bạn, xây dựng ứng dụng sản xuất và xuất bản lên cửa hàng Google Play.

**Lưu ý:** Android Studio liên tục được cải tiến. Để biết thông tin mới nhất về yêu cầu hệ thống và hướng dẫn cài đặt, hãy xem tại Android Studio

Để bắt đầu sử dụng Android Studio, trước tiên hãy kiểm tra yêu cầu hệ thống để đảm bảo rằng hệ thống của bạn đáp ứng chúng. Quá trình cài đặt tương tự cho tất cả các nền tảng. Bất kỳ khác biệt nào sẽ được ghi chú dưới đây.

1. Truy cập trang web dành cho nhà phát triển Android và làm theo hướng dẫn để tải xuống và cài đặt Android Studio.
2. Chấp nhận các cấu hình mặc định cho tất cả các bước và đảm bảo rằng tất cả các thành phần được chọn để cài đặt.
3. Sau khi hoàn tất cài đặt, Trình hướng dẫn Thiết lập sẽ tải xuống và cài đặt một số thành phần bổ sung, bao gồm Android SDK. Hãy kiên nhẫn, quá trình này có thể mất một chút thời gian tùy thuộc vào tốc độ Internet của bạn, và một số bước có thể có vẻ dư thừa.
4. Khi quá trình tải xuống hoàn tất, Android Studio sẽ khởi động, và bạn đã sẵn sàng để tạo dự án đầu tiên của mình.

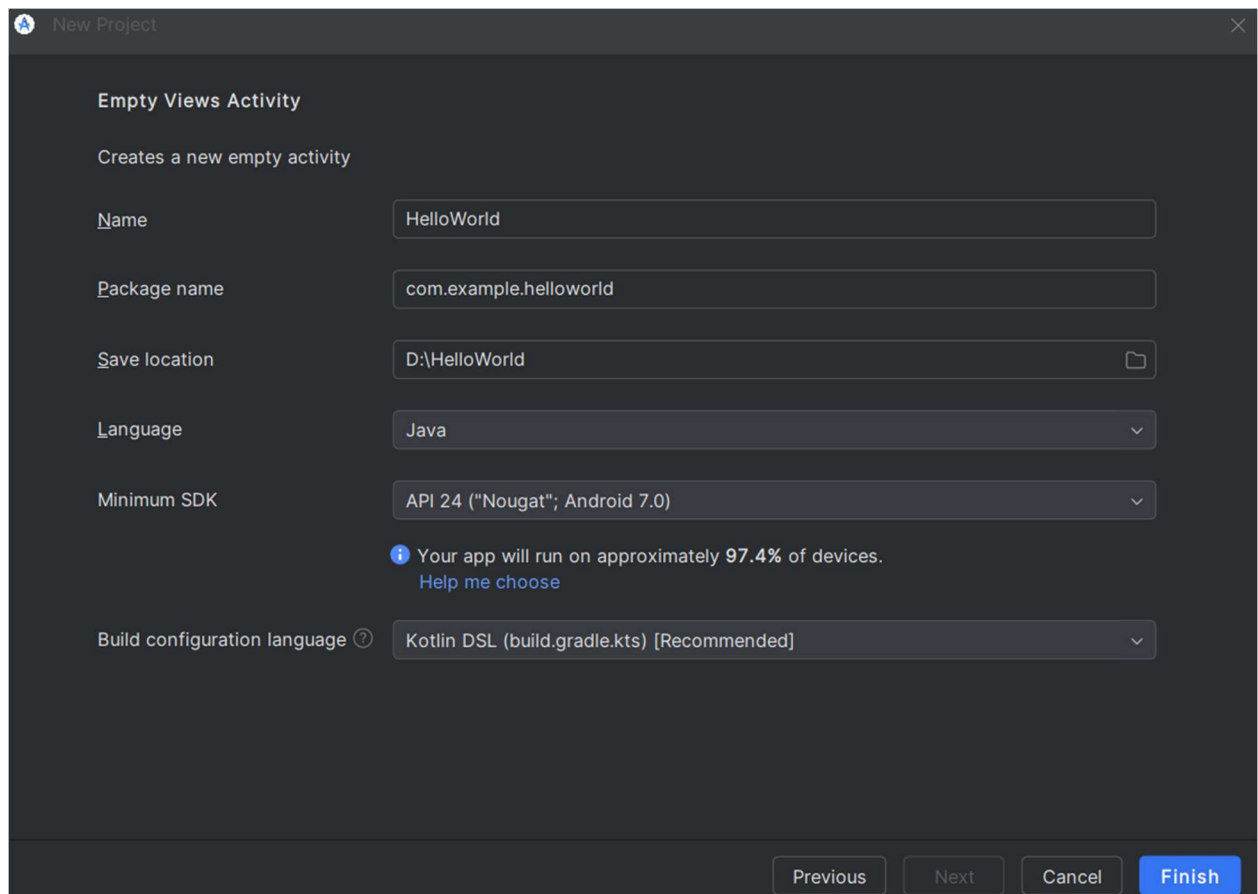
## **Task 2 : tạo HelloWorld app**

Trong nhiệm vụ này, bạn sẽ tạo một ứng dụng hiển thị "Hello World" để xác minh rằng Android Studio đã được cài đặt đúng cách và để tìm hiểu những điều cơ bản về phát triển với Android Studio.

### **2.1 tạo 1app project:**

1. Mở Android Studio nếu nó chưa được mở.
2. Trong cửa sổ chính "Chào mừng đến với Android Studio", nhấp vào "Bắt đầu một dự án Android Studio mới".

3. Trong cửa sổ "Tạo Dự án Android", nhập "Hello World" vào phần Tên ứng dụng (Application name).



4. Xác minh rằng vị trí dự án mặc định là nơi bạn muốn lưu trữ ứng dụng Hello World và các dự án Android Studio khác, hoặc thay đổi nó thành thư mục bạn muốn.

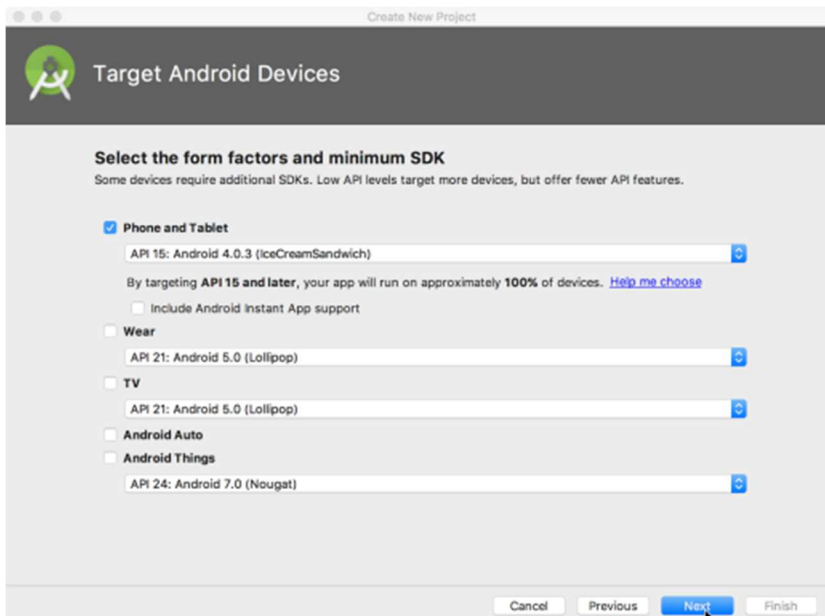
5. Chấp nhận mặc định "android.example.com" cho Tên miền công ty (Company Domain), hoặc tạo một tên miền công ty độc đáo. Nếu bạn không có kế hoạch xuất bản ứng dụng, bạn có thể giữ nguyên mặc định. Lưu ý rằng việc thay đổi tên gói (package name) của ứng dụng sau này sẽ đòi hỏi thêm công sức.

6. Để trống các tùy chọn "Hỗ trợ C++" (Include C++ support) và "Hỗ trợ Kotlin" (Include Kotlin support), sau đó nhấp vào "Tiếp theo" (Next).

7. Trên màn hình "Thiết bị Android mục tiêu" (Target Android Devices), mục "Điện thoại và Máy tính bảng" (Phone and Tablet) nên được chọn. Đảm bảo rằng API 15:



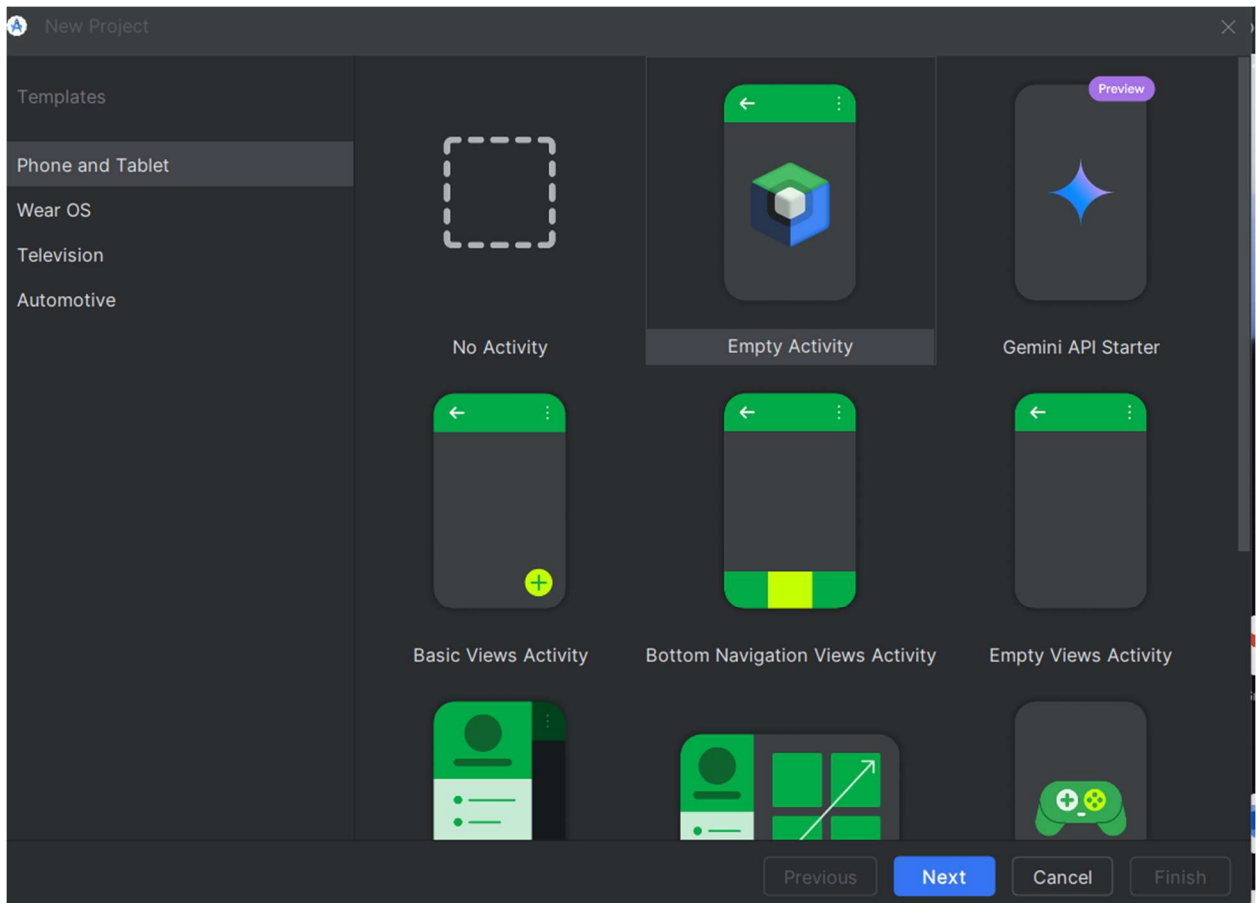
Android 4.0.3 IceCreamSandwich được đặt làm SDK tối thiểu (Minimum SDK); nếu không, hãy sử dụng menu thả xuống để thiết lập.



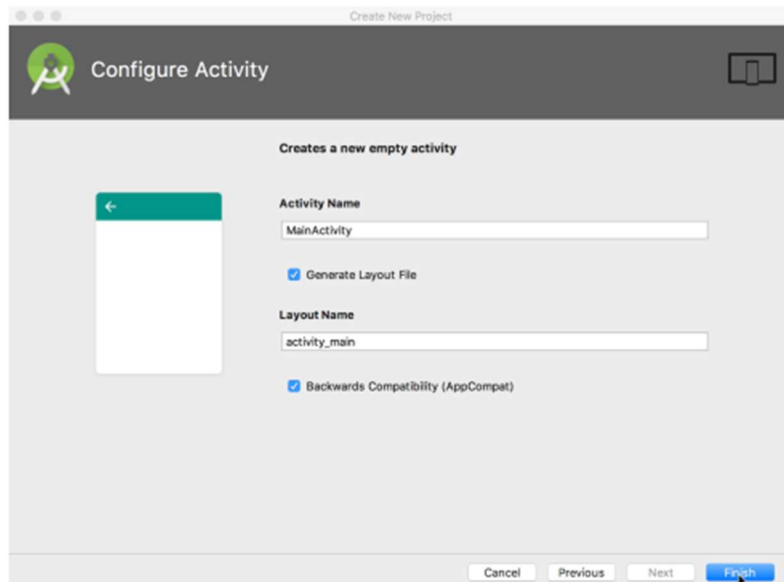
Đây là các cài đặt được sử dụng trong các ví dụ trong các bài học cho khóa học này. Đến thời điểm viết này, các cài đặt này làm cho ứng dụng Hello World của bạn tương thích với 97% thiết bị Android đang hoạt động trên Cốc Cốc Play Store.

8. Bỏ chọn Include Instant App support và tắt cả các tùy chọn khác. Sau đó nhấn Next. Nếu dự án của bạn cần các thành phần bổ sung cho SDK mục tiêu bạn đã chọn, Android Studio sẽ tự động cài đặt chúng.

9. Cửa sổ Add an Activity xuất hiện. Một Activity là một hoạt động tập trung mà người dùng có thể thực hiện. Đó là một thành phần quan trọng của bất kỳ ứng dụng Android nào. Một Activity thường có một bố cục đi kèm xác định cách các phần tử giao diện người dùng xuất hiện trên màn hình. Android Studio cung cấp các mẫu Activity để giúp bạn bắt đầu. Đối với dự án Hello World, hãy chọn Empty Activity như hình dưới đây, và nhấn Next.



10. Màn hình Cấu hình Hoạt động (Configure Activity) xuất hiện (giao diện sẽ khác nhau tùy thuộc vào mẫu bạn đã chọn ở bước trước). Theo mặc định, Hoạt động trống (Empty Activity) do mẫu cung cấp được đặt tên là **MainActivity**. Bạn có thể thay đổi nếu muốn, nhưng bài học này sẽ sử dụng **MainActivity**.



11. Đảm bảo rằng tùy chọn Generate Layout file được chọn. Theo mặc định, tên tệp giao diện là activity\_main. Bạn có thể thay đổi nếu muốn, nhưng bài học này sẽ sử dụng activity\_main.

12. Đảm bảo rằng tùy chọn Backwards Compatibility (App Compat) được chọn. Điều này giúp ứng dụng của bạn tương thích ngược với các phiên bản Android trước đó.

13. Nhấn Finish.

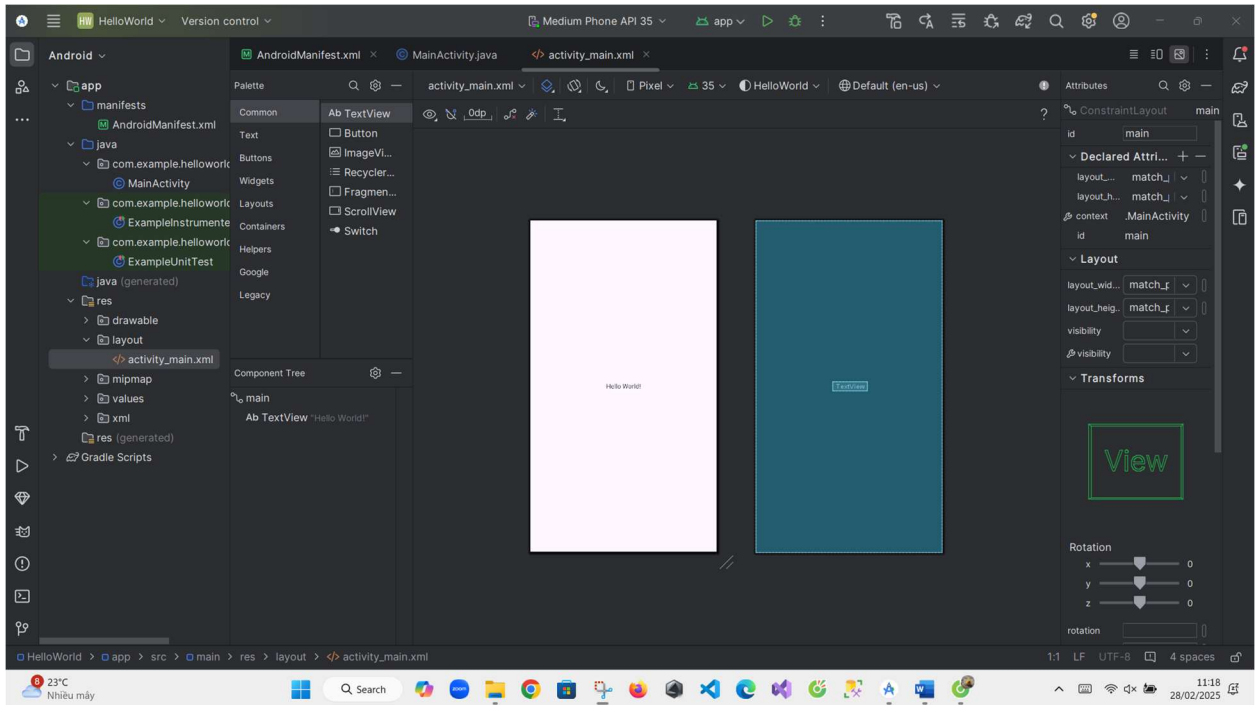
Android Studio tạo một thư mục cho các dự án của bạn và xây dựng dự án bằng Gradle (quá trình này có thể mất một lúc).

**Mẹo:** Xem trang **Configure your build** dành cho nhà phát triển để biết thông tin chi tiết.

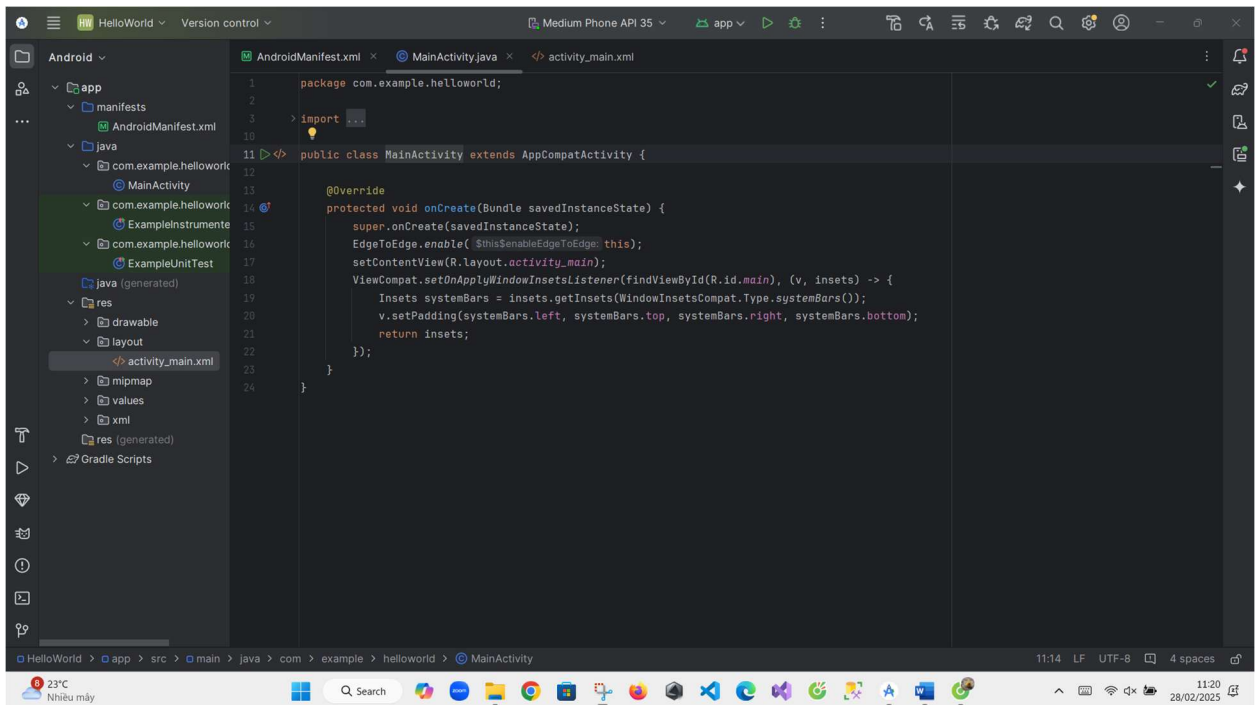
Bạn cũng có thể thấy thông báo **"Tip of the day"** với các phím tắt và mẹo hữu ích khác. Nhấn **Close** để đóng thông báo.

Trình chỉnh sửa của Android Studio xuất hiện. Thực hiện các bước sau:

1. Nhấp vào tab **activity\_main.xml** để mở trình chỉnh sửa giao diện.
2. Nhấp vào tab **Design** của trình chỉnh sửa giao diện (nếu chưa được chọn) để xem bản xem trước đồ họa của giao diện như hình bên dưới.



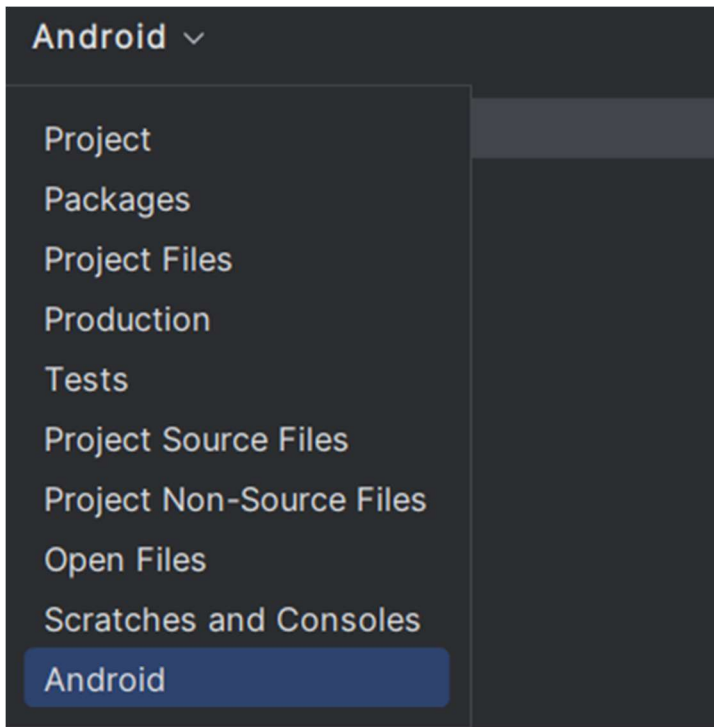
3. Ấn vào MainActivity.java để xem code như dưới đây :



## 2.2 Khám phá Project > Android

1. Nếu chưa được chọn, nhấp vào tab **Project** trong cột tab dọc bên trái cửa sổ Android Studio. Ngăn **Project** sẽ xuất hiện.

2. Để xem dự án theo cấu trúc chuẩn của Android, chọn **Android** từ menu thả xuống ở đầu ngăn **Project**, như hình bên dưới.

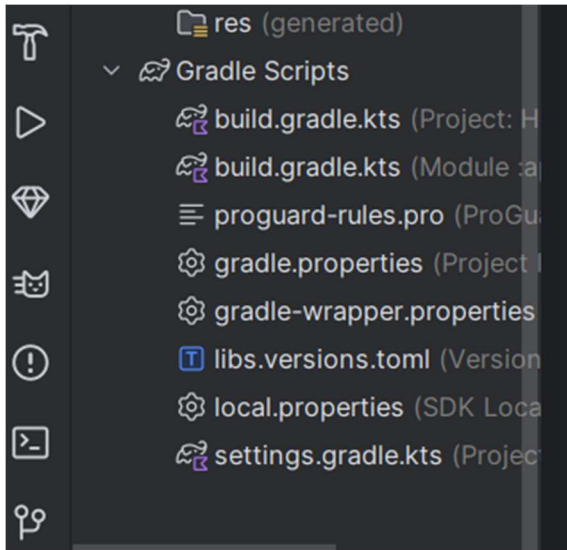


Lưu ý: Chương này và các chương khác gọi ngắn Project khi được đặt ở chế độ Android là Project > Android.

### 2.3 Khám phá thư mục Gradle Scripts:

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

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



### Các bước khám phá hệ thống Gradle:

1. Nếu thư mục Gradle Scripts chưa được mở rộng, nhấp vào biểu tượng tam giác để mở nó.

Thư mục này chứa tất cả các tệp cần thiết cho hệ thống build.

2. Tìm tệp build.gradle (Project: HelloWorld).

Đây là nơi chứa các tùy chọn cấu hình chung cho tất cả các mô-đun trong dự án của bạn. Mỗi dự án Android Studio đều có một tệp Gradle build cấp cao nhất. Thông thường, bạn không cần chỉnh sửa tệp này, nhưng hiểu nội dung của nó vẫn rất hữu ích.

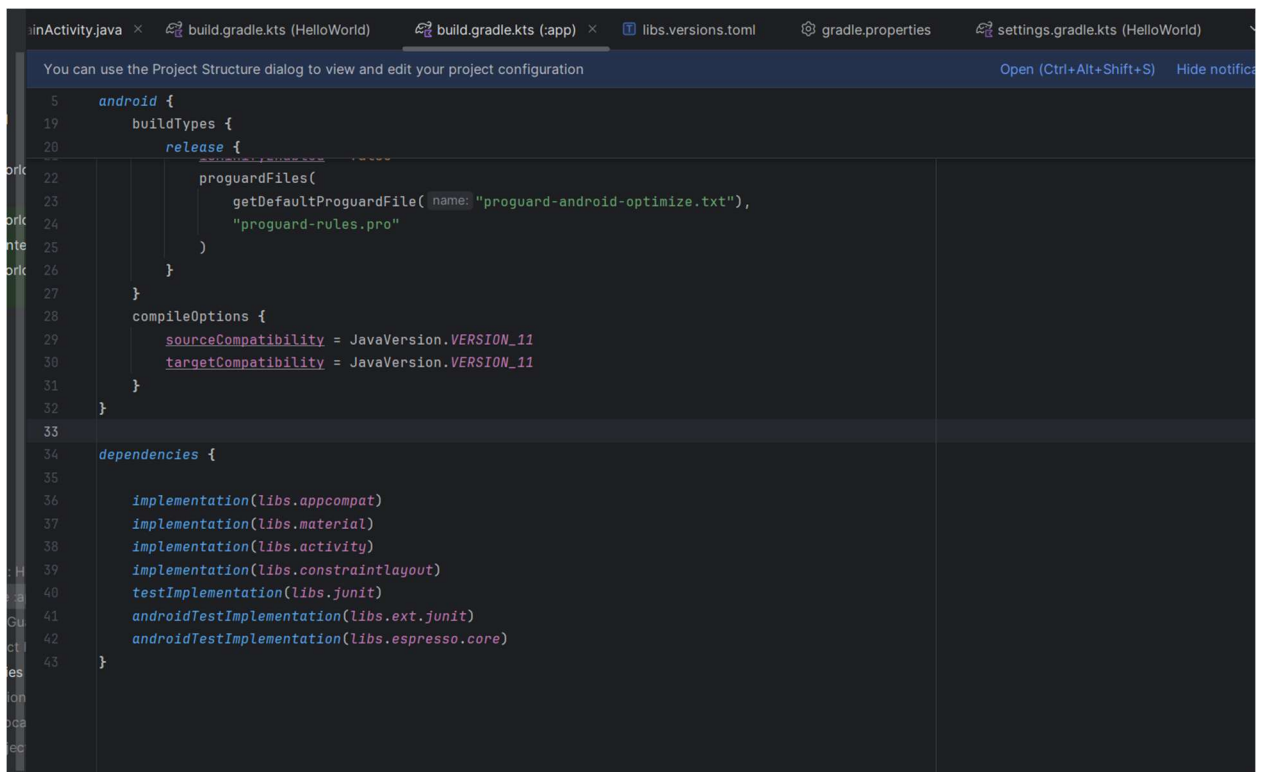
Theo mặc định, tệp build cấp cao nhất sử dụng khối buildscript để xác định kho lưu trữ (repositories) và các phụ thuộc (dependencies) chung cho tất cả mô-đun trong dự án. Khi bạn khai báo một dependency không phải thư viện cục bộ hoặc tập tin nội bộ, Gradle sẽ tìm kiếm tệp trong các kho trực tuyến được chỉ định trong khối repositories. Mặc định, dự án Android Studio mới sử dụng JCenter và Google (bao gồm kho Maven của Google) làm kho lưu trữ.

```
dependencyResolutionManagement {  
    repositoriesMode.set(RepositoriesMode.FAIL_ON_PROJECT_REPOS)  
    repositories {  
        google()  
        mavenCentral()  
    }  
}
```

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

Ngoài tệp build.gradle ở cấp dự án, mỗi mô-đun cũng có tệp build.gradle riêng. Tệp này cho phép bạn cấu hình các thiết lập build cho từng mô-đun cụ thể (trong ứng dụng HelloWorld, chỉ có một mô-đun duy nhất). Việc cấu hình các thiết lập build giúp bạn tùy chỉnh các tùy chọn đóng gói, chẳng hạn như build types và product flavors. Bạn cũng có thể ghi đè các thiết lập trong AndroidManifest.xml hoặc tệp build.gradle cấp dự án.

Tệp này thường là tệp cần chỉnh sửa khi thay đổi cấu hình cấp ứng dụng, chẳng hạn như khai báo dependencies trong phần dependencies. Bạn có thể khai báo một library dependency bằng nhiều cách khác nhau. Mỗi cấu hình dependency sẽ cung cấp cho Gradle hướng dẫn khác nhau về cách sử dụng thư viện. Ví dụ `e(dir: 'libs', include: ['*.jar'])` adds a dependency of all “\*.jar”



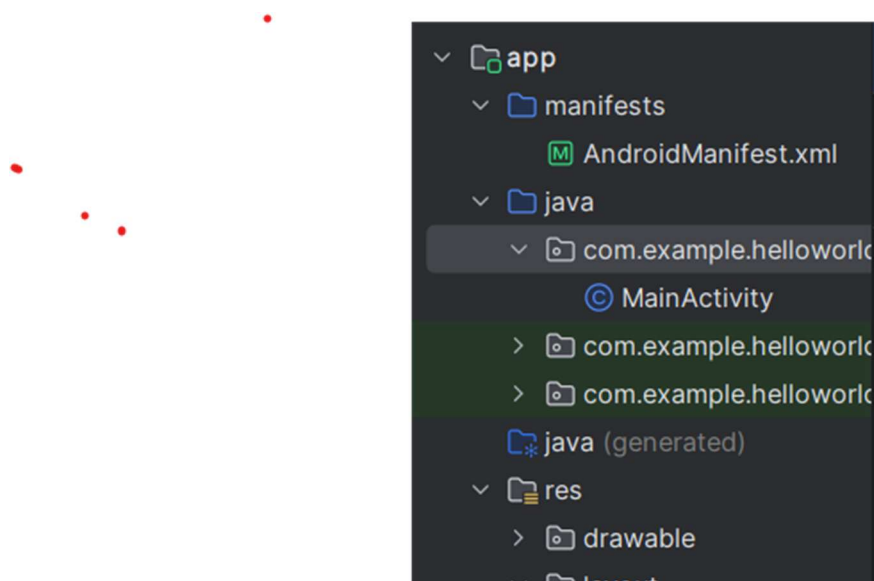
```
5  android {
19      buildTypes {
20          release {
21              // minifyEnabled = false
22              proguardFiles(
23                  getDefaultProguardFile( name: "proguard-android-optimize.txt"),
24                  "proguard-rules.pro"
25              )
26          }
27      }
28      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. Ấn vào hình tam giác để đóng Grandle Scripts.

## 2.4 Khám phá ứng và thư mục res:

Tất cả code và tài nguyên của ứng dụng nằm trong các thư mục app và res.

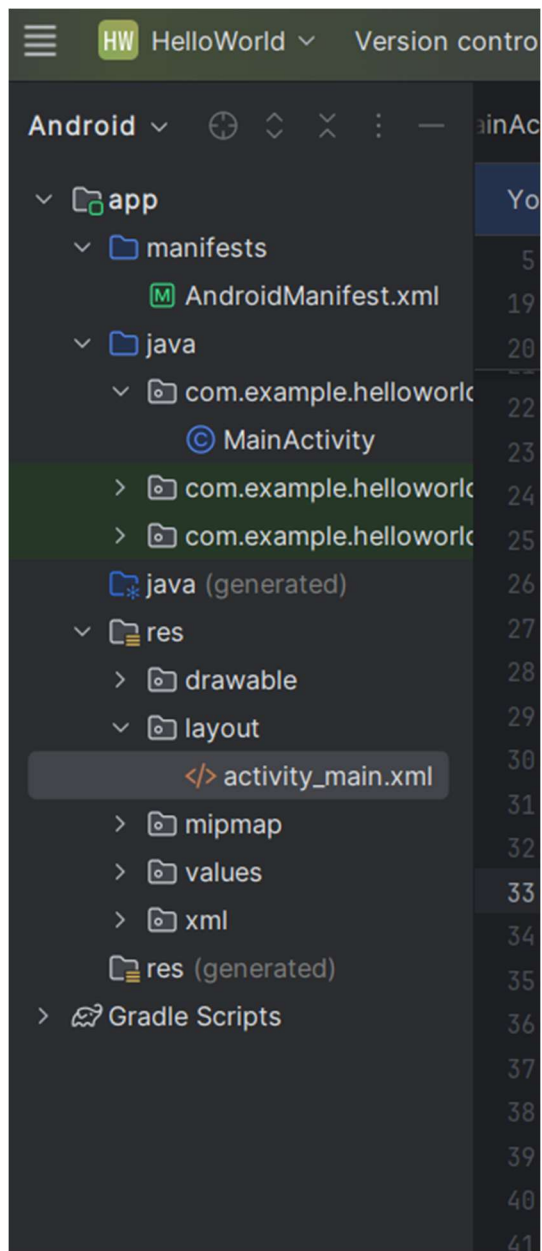
Mở rộng thư mục ứng dụng, thư mục java và thư mục com.example.android.helloworld thành xem tệp java MainActivity. Bấm đúp vào tệp sẽ mở nó trong trình chỉnh sửa mã.





Thư mục java bao gồm các tệp lớp Java trong ba thư mục con, như thể hiện trong hình trên. Thư mục com.example.hello.helloworld (hoặc tên miền bạn đã chỉ định) chứa tất cả các tệp cho một gói ứng dụng. Hai thư mục còn lại được sử dụng để kiểm tra và được mô tả trong một bài học khác. Đối với ứng dụng Hello World, chỉ có một gói duy nhất và nó chứa MainActivity.java. Tên của Hoạt động (màn hình) đầu tiên mà người dùng nhìn thấy cũng khởi tạo tài nguyên trên toàn ứng dụng, thường được gọi là MainActivity (phần mở rộng tệp là bị bỏ qua trong khung Project > Android).

2. Mở rộng thư mục res và thư mục bố cục, rồi bấm 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 và hình ảnh. Một hoạt động thường là được liên kết với bố cục của các chế độ xem giao diện người dùng được xác định dưới dạng tệp XML. Tập tin này thường được đặt tên theo tên của nó Activity.

## 2.5 Khám phá thư mục tệp kê khai

Thư mục bảng kê khai chứa các tệp cung cấp thông tin cần thiết về ứng dụng của bạn cho Hệ thống Android mà hệ thống phải có trước khi có thể chạy bất kỳ mã nào của ứng dụng.

1. Mở rộng thư mục bảng kê khai.

2. Mở tệp AndroidManifest.xml.

Tệp AndroidManifest.xml mô tả tất cả các thành phần của ứng dụng Android. Tất cả các thành phần của một ứng dụng, chẳng hạn như mỗi Hoạt động, phải được khai báo trong tệp XML này. Trong các bài học khóa học khác, bạn sẽ sửa đổi tệp này để thêm các tính năng và quyền tính năng. Để biết phần giới thiệu, hãy xem Tổng quan về tệp kê khai ứng dụng.

### **Task 3 : Sử dụng thiết bị ảo ( Giả lập)**

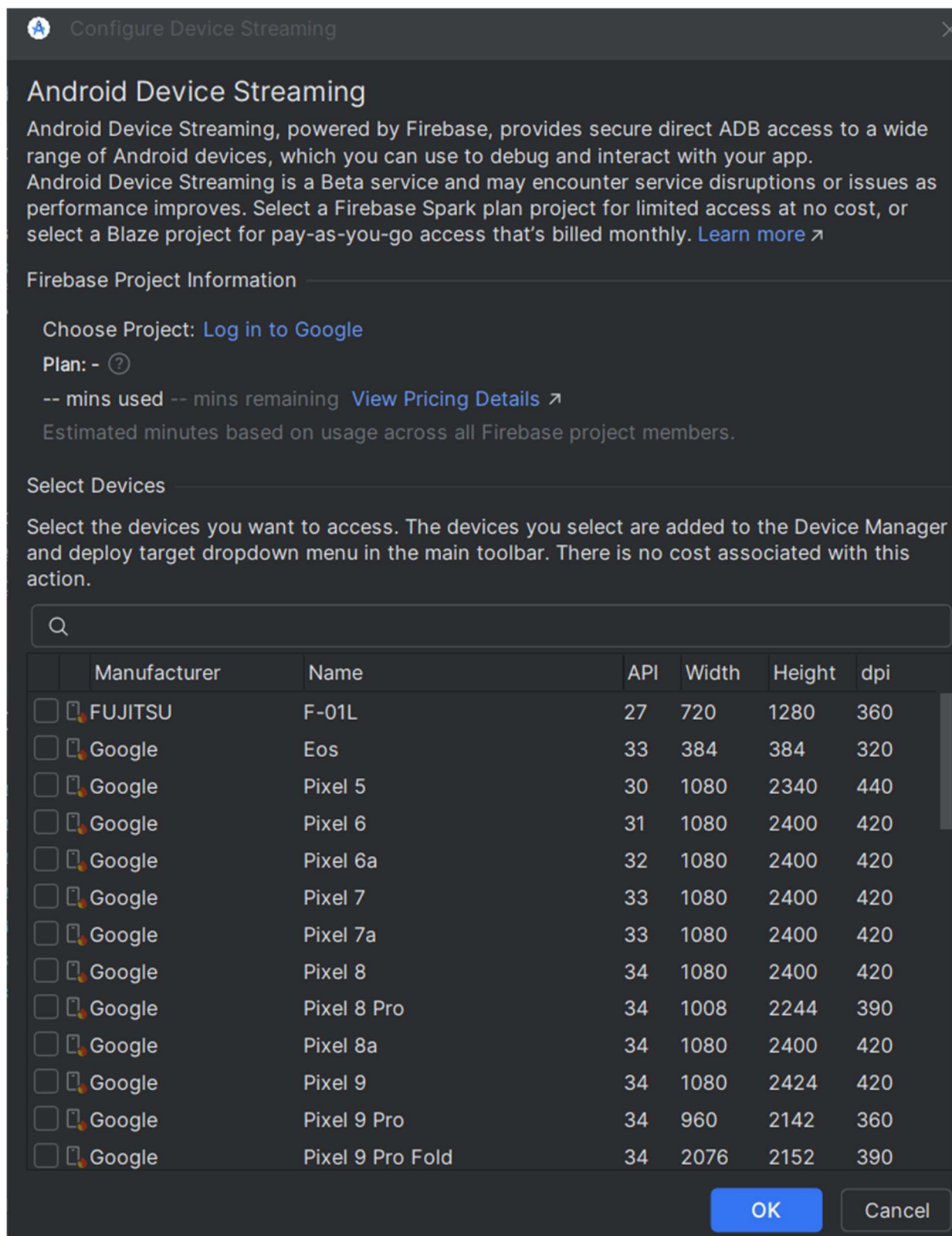
Trong nhiệm vụ này, bạn sẽ sử dụng trình quản lý Thiết bị ảo Android (AVD) để tạo một thiết bị ảo (cũng được gọi là trình mô phỏng) mô phỏng cấu hình cho một loại thiết bị Android cụ thể và sử dụng thiết bị ảo đó để chạy ứng dụng. Lưu ý rằng Trình mô phỏng Android có các yêu cầu bổ sung ngoài các yêu cầu hệ thống cơ bản dành cho Android Studio.

Sử dụng Trình quản lý AVD, bạn xác định các đặc điểm phần cứng của thiết bị, cấp độ API, bộ lưu trữ, skin và các thuộc tính khác và lưu nó dưới dạng thiết bị ảo. Với thiết bị ảo, bạn có thể thử nghiệm ứng dụng trên cấu hình thiết bị khác nhau (chẳng hạn như máy tính bảng và điện thoại) với các cấp độ API khác nhau mà không cần phải sử dụng các thiết bị vật lý.

#### **3.1 Tạo thiết bị ảo Android (AVD)**

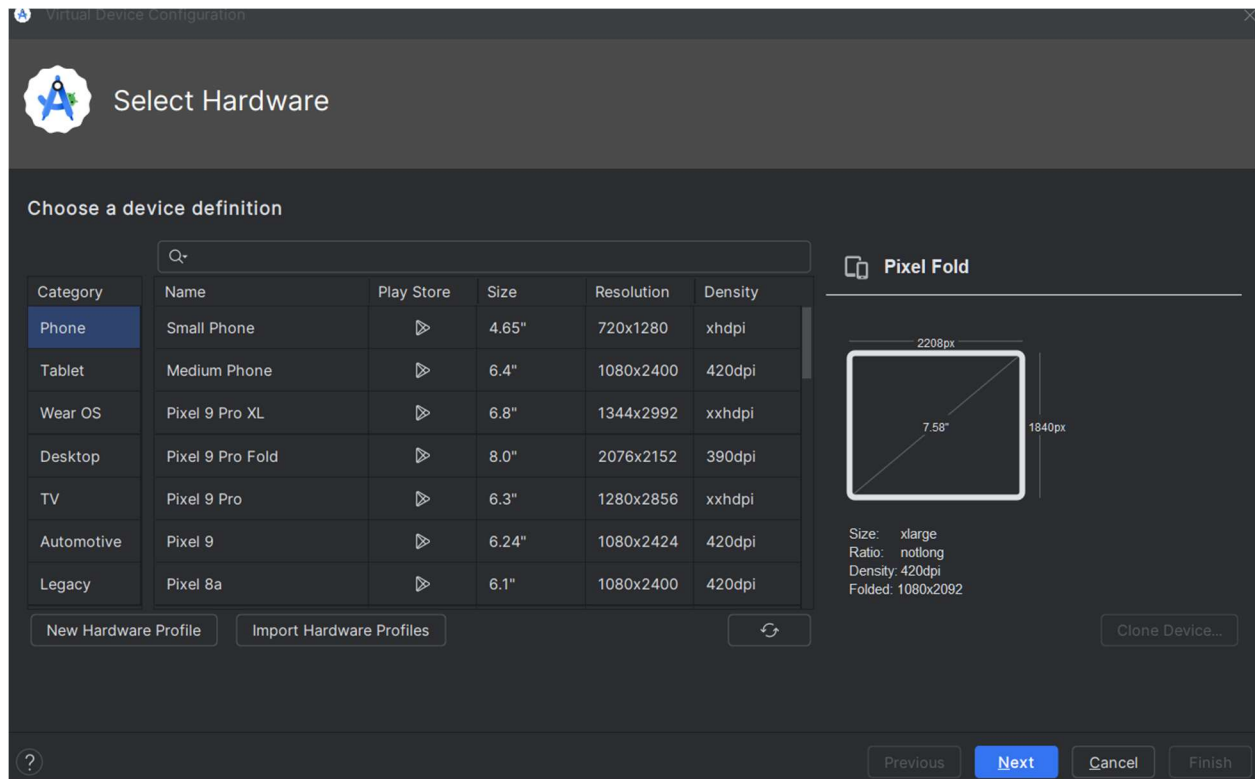
Để chạy trình mô phỏng trên máy tính, bạn phải tạo cấu hình mô tả thiết bị ảo.

1. Trong Android Studio, chọn Tools > Device Manager, hoặc nhấp vào biểu tượng Trình quản lý AVD trên thanh công cụ. Màn hình Thiết bị ảo của bạn xuất hiện. Nếu bạn đã tạo thiết bị ảo, màn hình hiển thị chúng (như trong hình bên dưới); nếu không bạn sẽ thấy một danh sách trống.



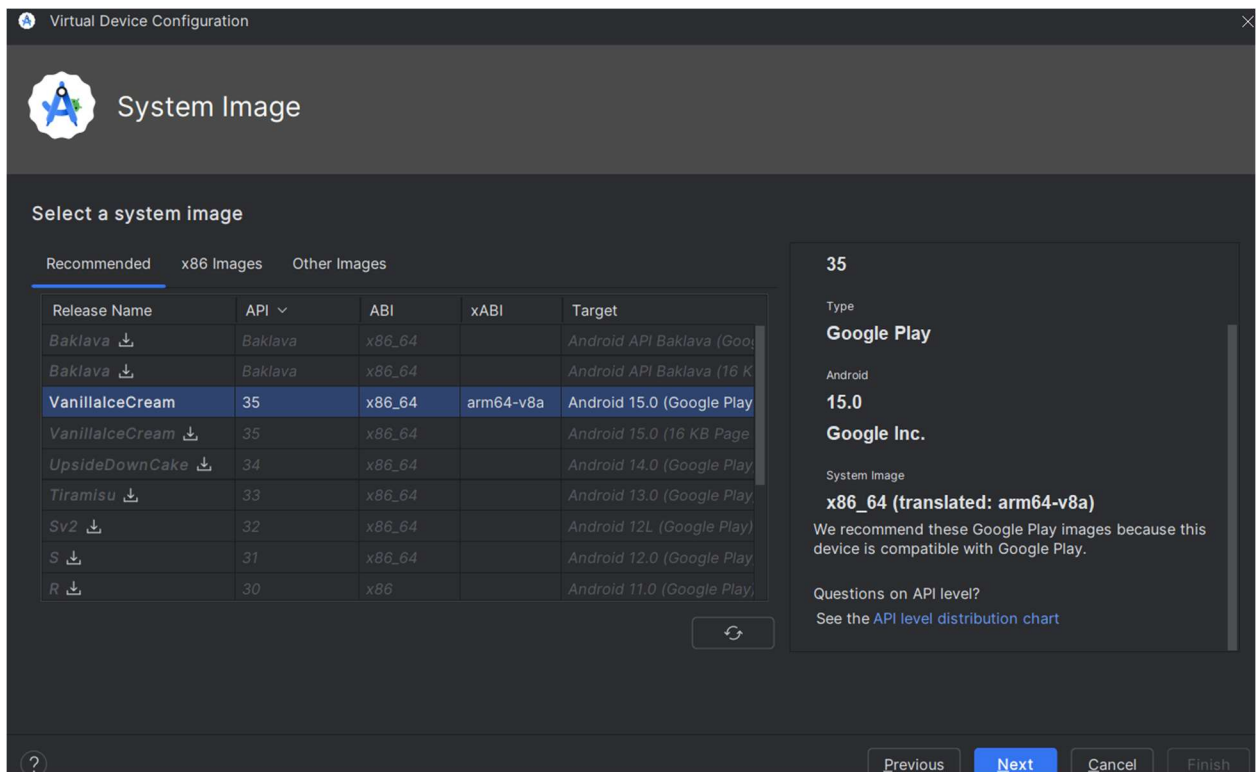
2. Nhấp vào +Tạo thiết bị ảo. Cửa sổ Chọn phần cứng xuất hiện hiển thị danh sách thiết bị phần cứng được cấu hình sẵn. Đối với mỗi thiết bị, bảng cung cấp một cột cho kích

thước hiển thị đường chéo (Size), độ phân giải màn hình tính bằng pixel (Resolution) và mật độ pixel (Tỉ trọng).



3. Chọn một thiết bị như Nexus 5x hoặc Pixel XL và nhấp vào Tiếp theo. Màn hình System image xuất hiện.

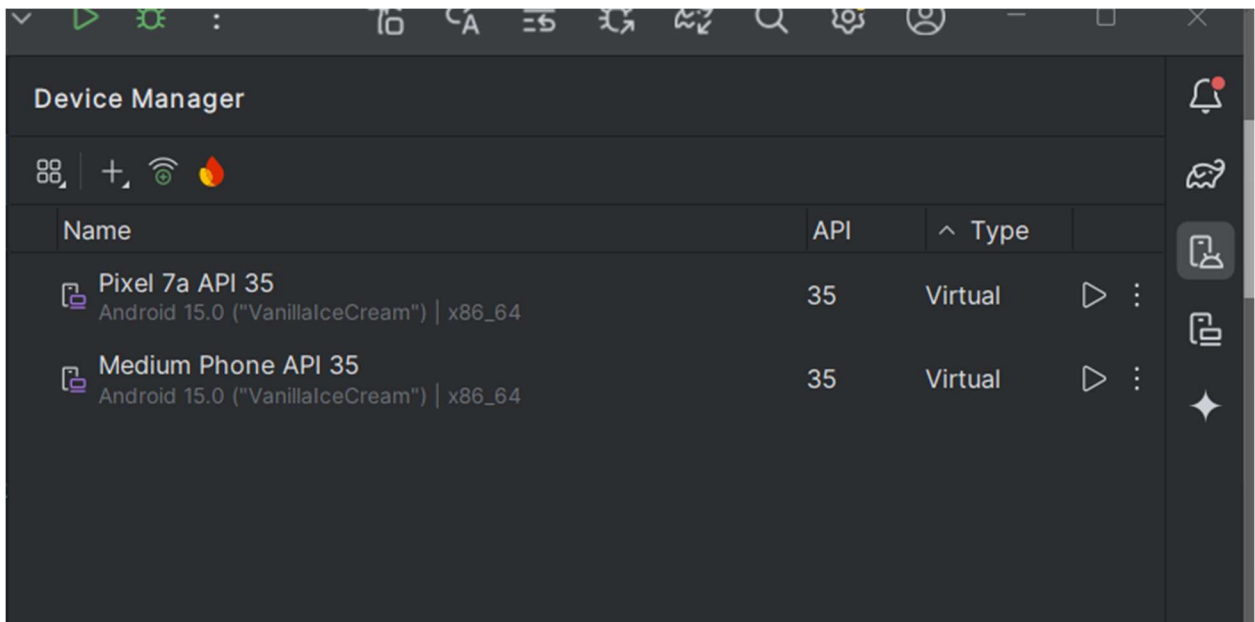
4. Nhấp vào tab Đề xuất nếu nó chưa được chọn và chọn phiên bản hệ thống Android để chạy trên thiết bị ảo (chẳng hạn như Oreo).



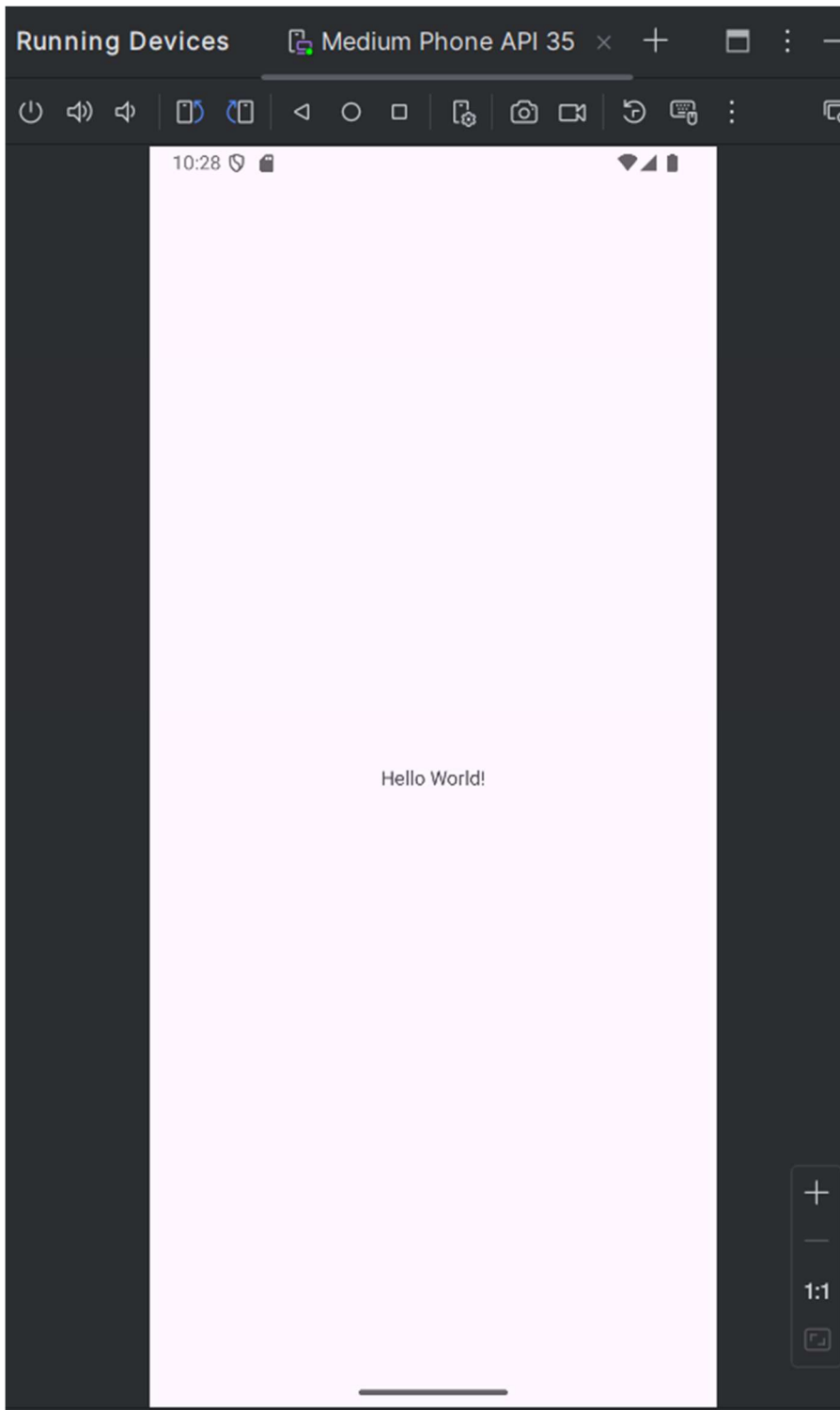
- Sau khi chọn system image, nhấn **Next**, kiểm tra cấu hình trong cửa sổ AVD, rồi nhấn **Finish** để hoàn tất.

### 3.2 Run the app on the virtual device

- vào android studio, chọn chạy hoặc ấn vào kí hiệu
- Trong cửa sổ Select Deployment Target, chọn thiết bị ảo vừa tạo trong Available Virtual Devices, rồi nhấn OK.



Trình giả lập khởi động như một thiết bị thật và có thể mất một lúc tùy vào tốc độ máy tính. Khi trình giả lập sẵn sàng, Android Studio sẽ tải ứng dụng lên và chạy nó, hiển thị ứng dụng Hello World.



tip: Khi kiểm thử trên thiết bị ảo, hãy khởi động nó ngay từ đầu và giữ mở cho đến khi hoàn tất để tránh mất thời gian khởi động lại. Để đóng, nhấn X, chọn Quit, hoặc dùng Control-Q (Windows) / Command-Q (macOS).

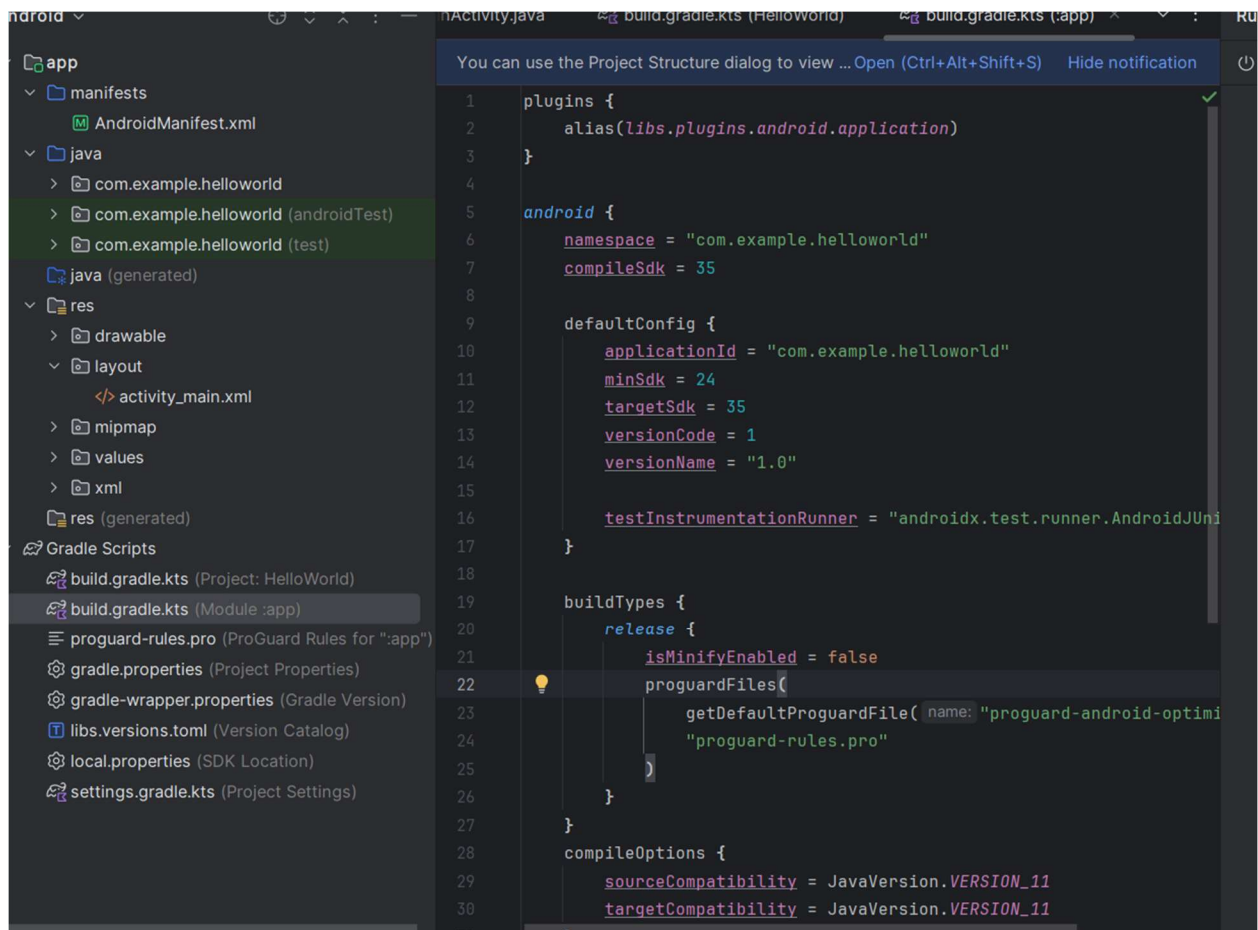


TASK 4: sử dụng thiết bị thật

TASK 5: thay đổi cấu hình Gradle của ứng dụng

### 5.1 thay đổi phiên bản sdk tối thiểu cho ứng dụng

1. Mở rộng thư mục **Gradle Scripts** (nếu chưa mở) và nhấp đúp vào tệp **build.gradle (Module: app)** để mở trong trình chỉnh sửa mã.
2. Trong khối **defaultConfig**, thay đổi giá trị **minSdkVersion** thành **17** (ban đầu là 15).



Trình chỉnh sửa mã hiển thị một thanh thông báo ở trên cùng với liên kết **Sync Now**.

### 5.2 đồng bộ hoá cấu hình Gradle mới

Khi thay đổi tệp cấu hình build, Android Studio yêu cầu **đồng bộ dự án** để nhập các thay đổi và kiểm tra lỗi.

Nhấp **Sync Now** trong thanh thông báo hoặc biểu tượng **Sync Project with Gradle Files** trên thanh công cụ.

Khi đồng bộ hoàn tất, thông báo **Gradle build finished** sẽ xuất hiện ở góc dưới bên trái.

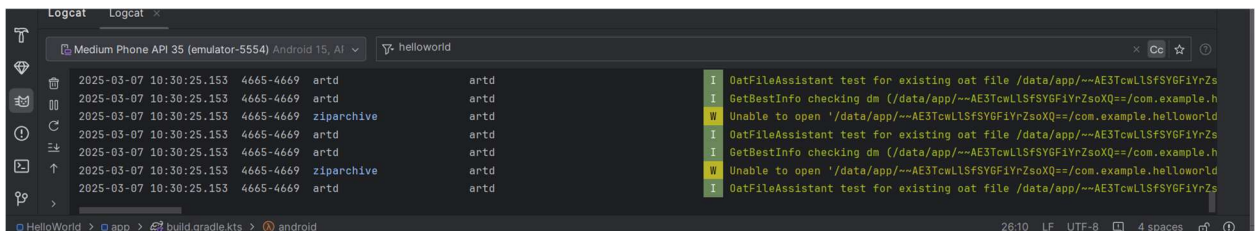
Để tìm hiểu thêm, hãy xem tài liệu **Build System Overview** và **Configuring Gradle Builds**.

TASK6 : thêm báo cáo vào ứng dụng của bạn

Trong nhiệm vụ này, bạn sẽ thêm **Log statements** vào ứng dụng để hiển thị thông báo trong ngăn **Logcat**.

**Log messages** là công cụ gỡ lỗi mạnh mẽ giúp kiểm tra giá trị, luồng thực thi và báo cáo lỗi.

## 6.1 xem khung Logcat



2. **Tab Logcat:** Dùng để mở/đóng ngăn **Logcat**, hiển thị thông tin về ứng dụng khi đang chạy. Nếu bạn thêm **Log statements**, các thông báo sẽ xuất hiện tại đây.

1. **Menu Log level:** Mặc định đặt ở **Verbose** (hiển thị tất cả Log messages). Các tùy chọn khác gồm **Debug**, **Error**, **Info**, **Warn**.

Câu lệnh nhật ký trong mã ứng dụng của bạn sẽ hiển thị thông báo trong ngăn Logcat.

Ví dụ:

```
Log.d("MainActivity", "Hello World");
```

Các phần của thông điệp là: • **Log** : Lớp Log để gửi thông báo nhật ký đến ngăn Logcat. • **d** : Cài đặt mức nhật ký Debug để lọc thông báo nhật ký hiển thị trong ngăn Logcat. Các cấp độ nhật ký khác là E cho Error , W cho Warn và I cho Info . • **"MainActivity"** : Đối số đầu tiên là một thẻ có thể được sử dụng để lọc tin nhắn trong ngăn Logcat. Đây thường là tên của Hoạt động mà thông điệp bắt đầu. Tuy

nhiên, bạn có thể làm cho điều này bất cứ thứ gì hữu ích cho bạn để gỡ lỗi. Theo quy ước, thẻ nhật ký được định nghĩa là hằng số cho Hoạt động:

```
private static final String LOG_TAG = MainActivity.class.getSimpleName();
```

- "Hello world" : Đối số thứ hai là thông điệp thực tế. Làm theo các bước sau:

1. Mở ứng dụng Hello World của bạn trong Android studio và mở MainActivity .
2. Để tự động thêm các mục nhập rõ ràng vào dự án của bạn (chẳng hạn như android.util.Log cần thiết để sử dụng Log ), hãy chọn File > Settings trong Windows hoặc Tùy chọn Android Studio > Preferences trong macOS.
3. Chọn Editor > General >Auto Import. Chọn tất cả các hộp kiểm và Insert imports on paste to All.
4. Nhấp vào Apply và sau đó nhấp vào OK 5. Trong phương thức onCreate() của MainActivity , thêm câu lệnh sau

```
Log.v("main", "Hello World");
```

Phương thức onCreate() bây giờ sẽ giống như mã sau:

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    EdgeToEdge.enable( $this$enableEdgeToEdge: this);
    setContentView(R.layout.activity_main);
    Log.v( tag: "main", msg: "Hello World");
    ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets) -> {
        Insets systemBars = insets.getInsets(WindowInsetsCompat.Type.systemBars());
        v.setPadding(systemBars.left, systemBars.top, systemBars.right, systemBars.bottom);
        return insets;
    });
}
```

6. Nếu ngăn Logcat chưa mở, hãy nhấp vào tab Logcat ở cuối Android Studio để mở.
7. Kiểm tra xem tên của mục tiêu và tên gói của ứng dụng có chính xác không.
8. Thay đổi mức nhật ký trong ngăn Logcat thành Debug (hoặc để nguyên ở chế độ Verbose vì có rất ít thông báo nhật ký).
9. Chạy ứng dụng của bạn.

```
2025-02-28 07:53:27.297 26270-26270 main android.example.com.helloworld V Hello World
```

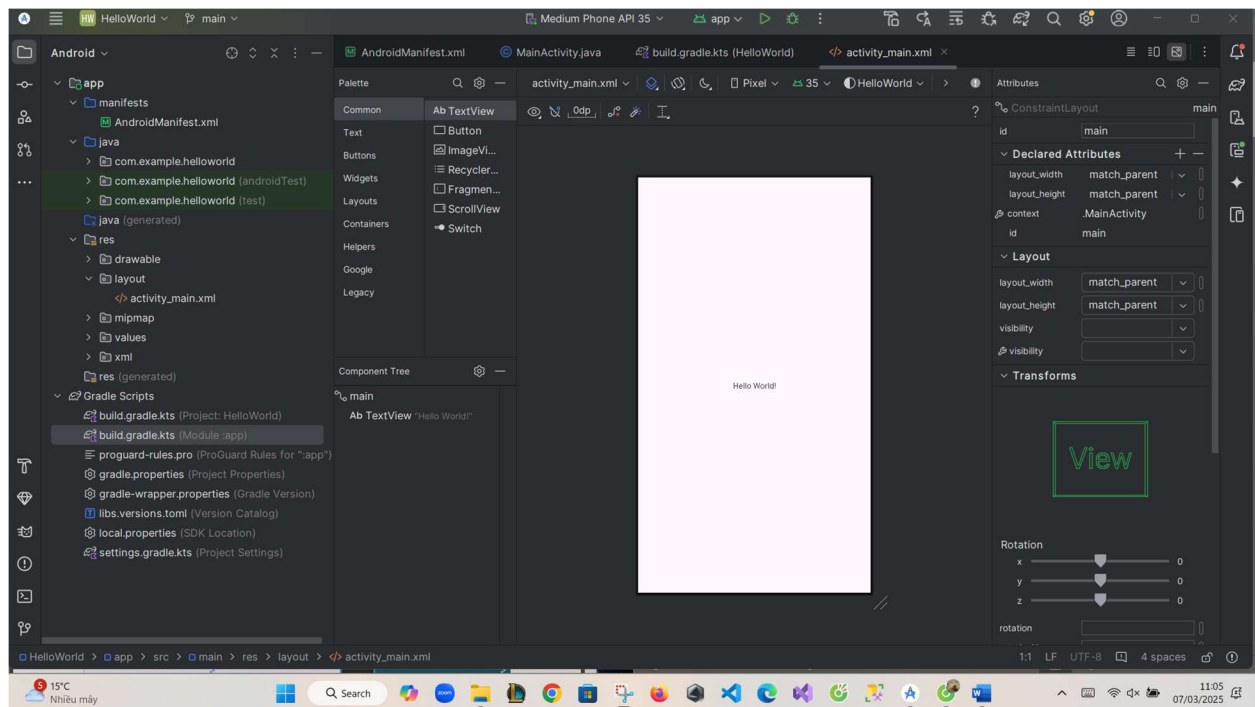
## 1.2) Giao diện người dùng tương tác đầu tiên

Giới thiệu Giao diện người dùng (UI) xuất hiện trên màn hình của thiết bị Android bao gồm một hệ thống phân cấp các đối tượng được gọi là views — mọi phần tử của màn hình là một View . Lớp View đại diện cho khối xây dựng cơ bản cho tất cả các thành phần giao UI và lớp cơ sở cho các lớp cung cấp các thành phần UI tương tác như nút, hộp kiểm và trường nhập văn bản. Các lớp con View thường được sử dụng được mô tả trong một số bài học bao gồm:

- TextView để hiển thị văn bản.
- EditText để cho phép người dùng nhập và chỉnh sửa văn bản.
- Nút và các yếu tố có thể nhấp khác (chẳng hạn như RadioButton, CheckBox và Spinner ) để cung cấp hành vi tương tác.
- ScrollView và RecyclerView để hiển thị các mục có thể cuộn.
- ImageView để hiển thị hình ảnh.
- ConstraintLayout và LinearLayout để chứa các phần tử View khác và định vị chúng

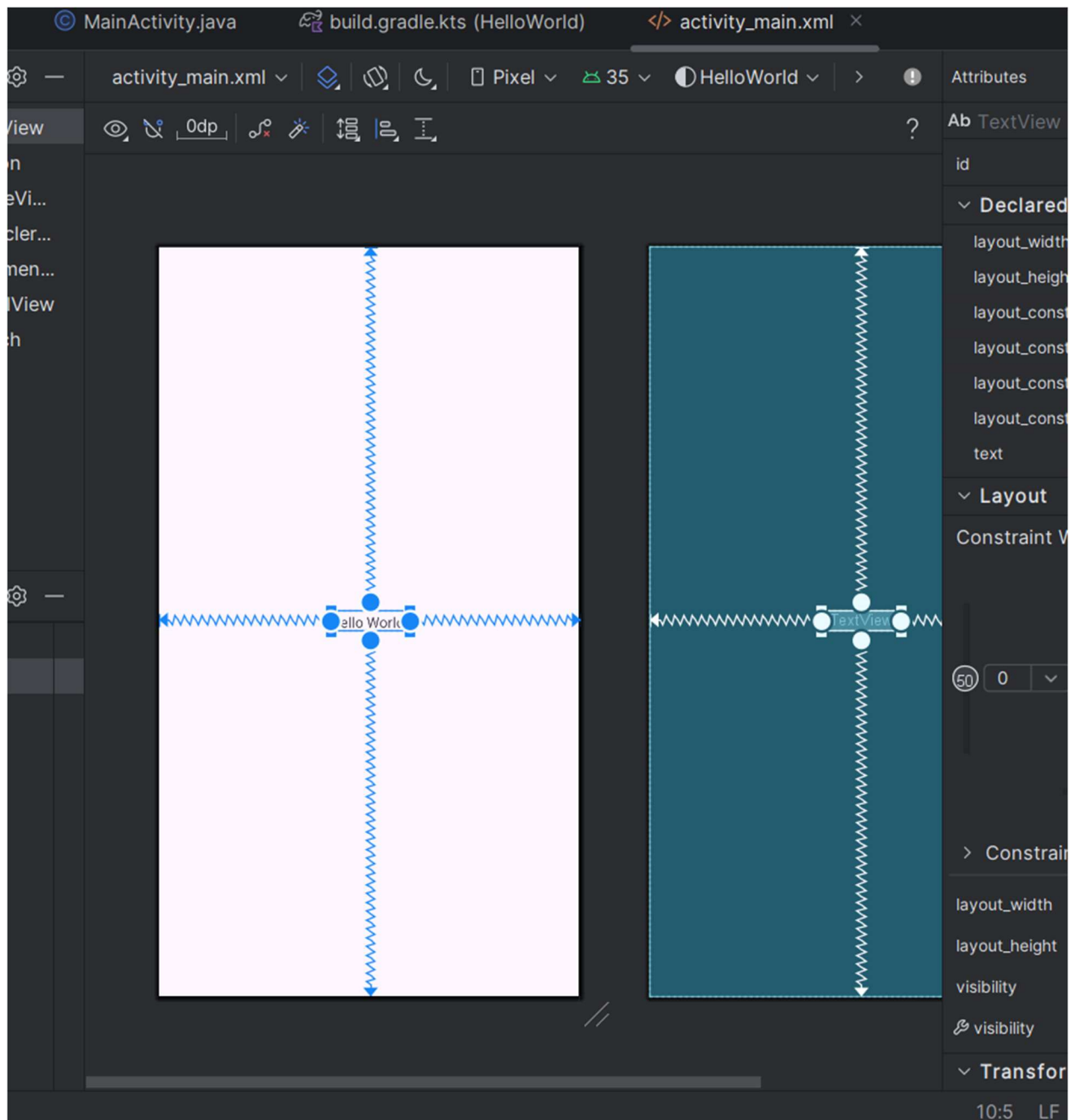
a. Tạo và khám phá một dự án mới

Trình chỉnh sửa bố cục



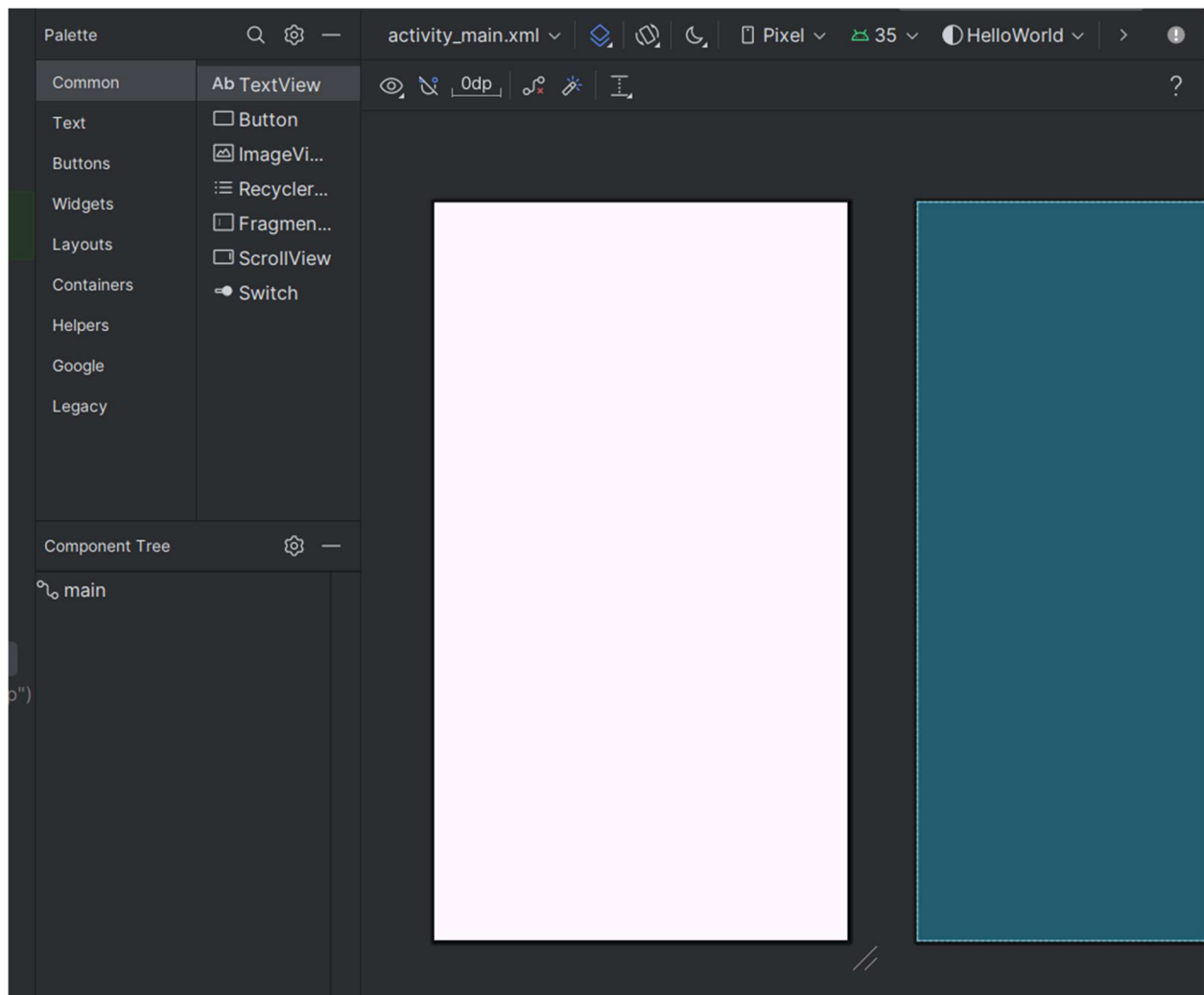
b. Thêm các phần tử View vào trình chỉnh sửa bố cục

Kiểm tra rằng buộc phần tử



Thêm một nút vào bố cục:

Bố cục trông

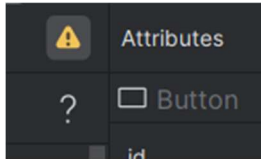
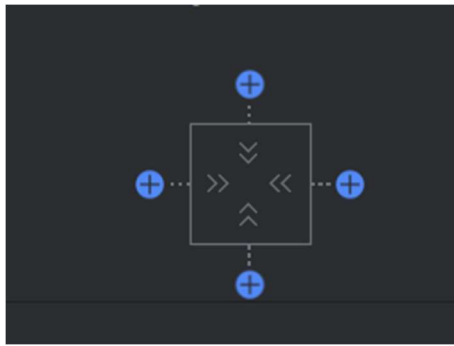


Thêm nút bấm

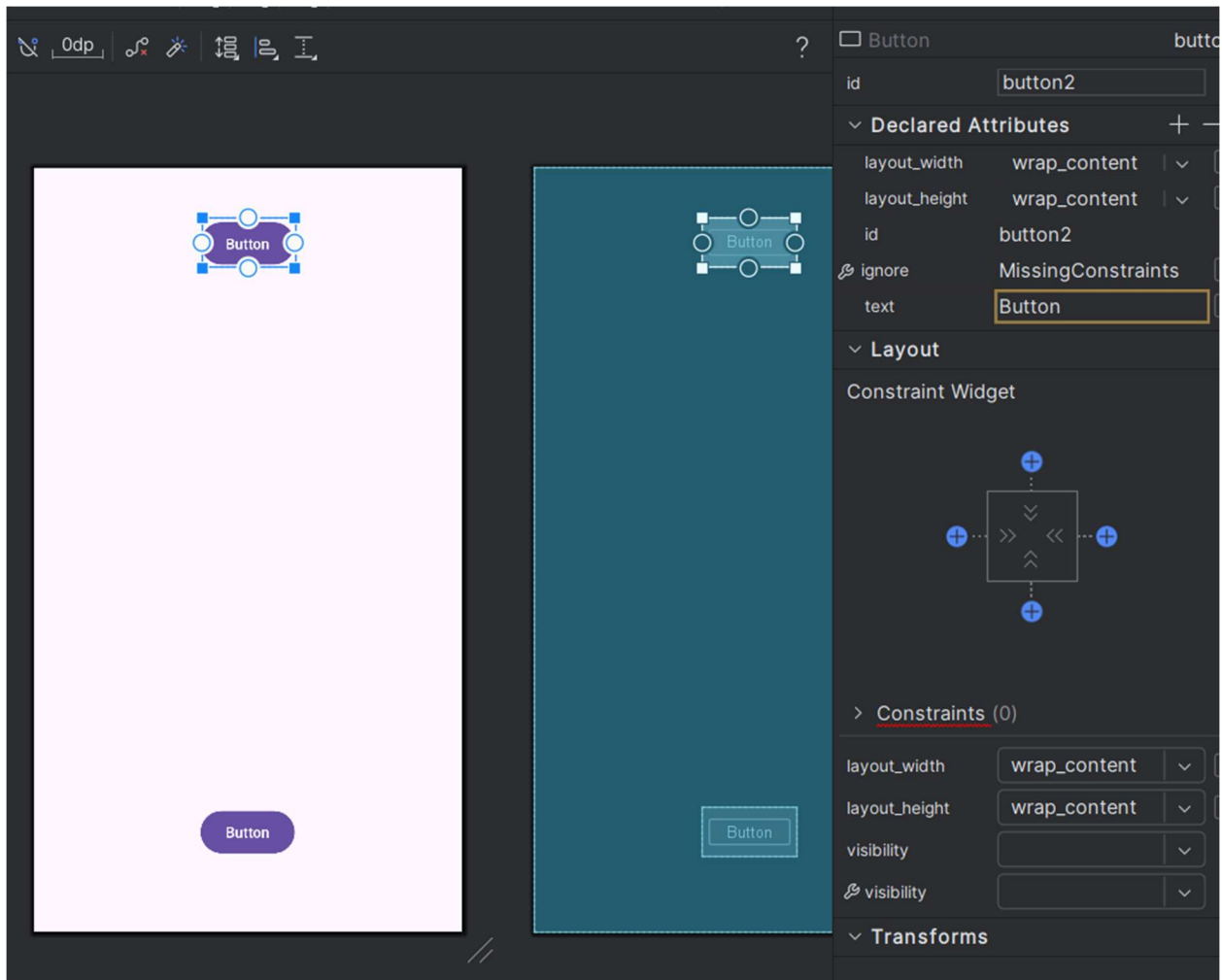


c. Thay đổi thuộc tính của phần tử giao diện người dung

Thay đổi kích cỡ nút



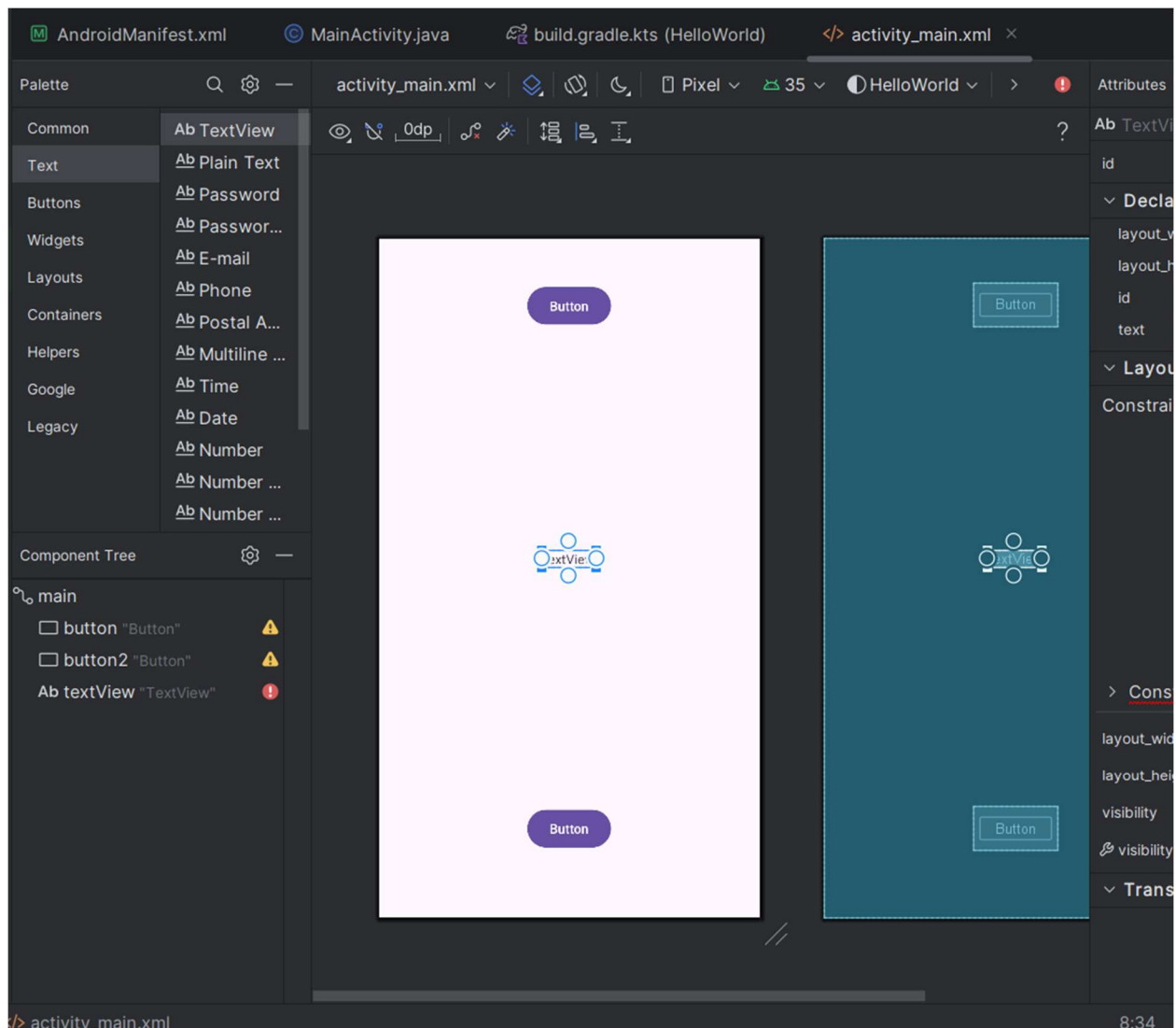
Thêm 2 nút bấm và thay đổi kích thước



d. Thêm một TextEdit và đặt thuộc tính cho nó



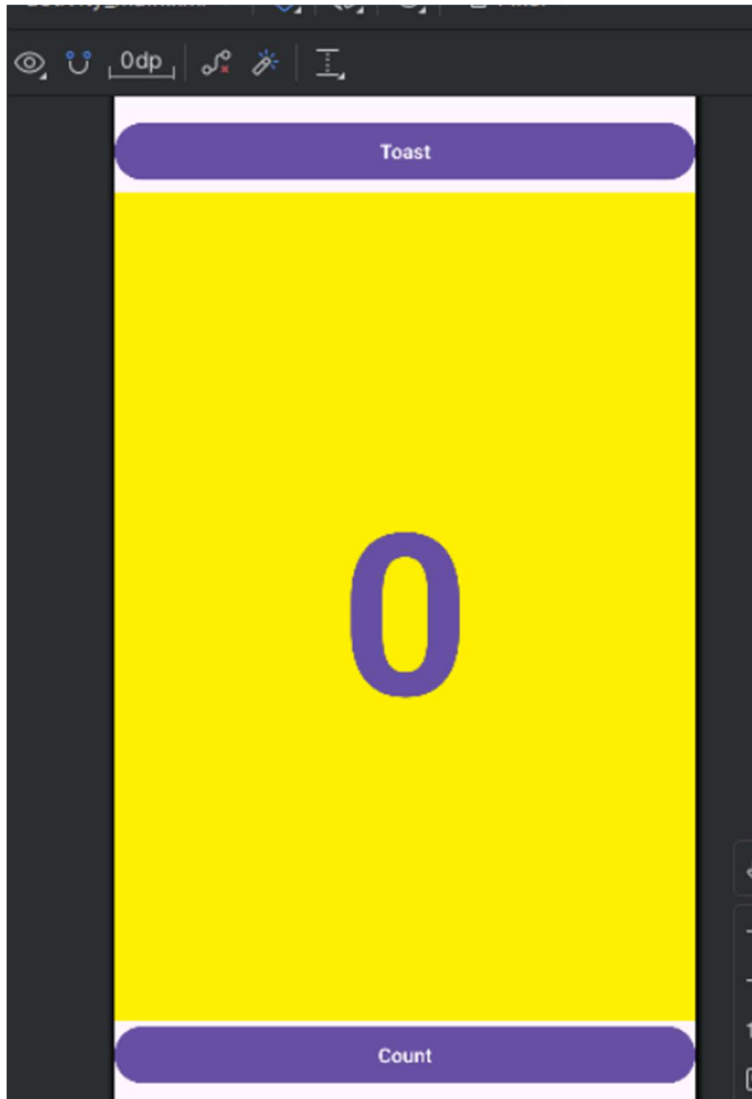
Thêm TextView vào màn hình và kết nối bên trên của nó với bên dưới của nút Toast, bên dưới của nó với bên trên của nút Count



1. Đặt ID thành show\_count .
2. Đặt văn bản thành 0 .
3. Đặt textSize thành 160sp .
4. Đặt textStyle thành B (in đậm) và textAlignment thành ALIGNCENTER (căn giữa đoạn văn).
5. Thay đổi các điều khiển kích thước chế độ xem ngang và dọc ( layout\_width và layout\_height ) thành match\_constraint .

6. Đặt textColor thành @color/colorPrimary .

7. Cuộn xuống gần và nhấp vào View all attributes, cuộn xuống trang thứ hai của các thuộc tính sang nền , sau đó nhập #FFF00 cho màu vàng. 8. Cuộn xuống trọng lực , mở rộng trọng lực và chọn center\_ver (đối với trung tâm dọc).



- textSize : Kích thước văn bản của TextView . Đối với bài học này, kích thước được đặt thành 160sp . sp là viết tắt của pixel-independent scale, và giống như dp, là một đơn vị chia tỷ lệ theo mật độ màn hình và tùy chọn kích thước phông chữ của người dùng. Sử dụng đơn vị dp khi bạn chỉ định kích thước phông chữ để kích thước được điều chỉnh cho cả mật độ màn hình và sở thích của người dùng.

- `textStyle` và `textAlignment` : Kiểu văn bản, được đặt thành B (in đậm) trong bài học này và căn chỉnh văn bản, được đặt thành `ALIGNCENTER` (căn giữa đoạn văn).

- trọng lực : Thuộc tính trọng lực chỉ định cách một Chế độ xem được căn chỉnh trong Chế độ xem hoặc Nhóm chế độ xem cha của nó. Trong bước này, bạn căn giữa `TextView` theo chiều dọc trong `ConstraintLayout`. Bạn có thể nhận thấy rằng thuộc tính nền nằm trên trang đầu tiên của ngăn Thuộc tính cho Nút, nhưng trên trang thứ hai của ngăn Thuộc tính cho `TextView` . Ngăn Thuộc tính thay đổi cho từng loại Dạng xem: Các thuộc tính phổ biến nhất cho loại Chế độ xem xuất hiện trên trang đầu tiên và phần còn lại được liệt kê trên trang thứ hai. Để quay lại trang đầu tiên của ngăn Thuộc tính, hãy bấm vào biểu tượng trên thanh công cụ ở đầu ngăn.

e. Thiết kế bố cục trong XML Bố cục ứng dụng Hello Toast gần như đã hoàn thành! Tuy nhiên, một dấu chấm than xuất hiện bên cạnh mỗi phần tử giao diện người dùng trong Cây thành phần. Di con trỏ của bạn qua các dấu chấm than này để xem thông báo cảnh báo, như được hiển thị bên dưới. Cảnh báo tương tự xuất hiện cho cả ba phần tử: chuỗi được mã hóa cứng nên sử dụng tài nguyên.

Cách dễ nhất để khắc phục sự cố bố cục là chỉnh sửa bố cục trong XML. Mặc dù trình soạn thảo bố cục là một công cụ mạnh mẽ, nhưng một số thay đổi dễ dàng thực hiện trực tiếp trong mã nguồn XML.

Mở mã XML cho bố cục Đối với tác vụ này, hãy mở tệp `activity_main.xml` nếu nó chưa được mở và nhấp vào tab Văn bản ở góc trên bên phải trình chỉnh sửa bố cục.

Trình soạn thảo XML xuất hiện, thay thế các ngăn thiết kế và bản thiết kế. Như bạn có thể thấy trong hình bên dưới, cho thấy một phần của mã XML cho bố cục, các cảnh báo được đánh dấu — các chuỗi được mã hóa cứng "Toast" và "Count" . (Mã hóa cứng "0" cũng được đánh dấu nhưng không được hiển thị trong hình.) Di con trỏ của bạn lên chuỗi được mã hóa cứng "Toast" để xem thông báo cảnh báo

```

<Button
    android:layout_width="0dp"
    android:layout_height="wrap_content"
    android:layout_marginTop="28dp"
    android:text="Toast"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="0.498"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent" />

<TextView
    android:id="@+id/show_count"
    android:layout_width="0dp"
    android:layout_height="0dp"
    android:layout_marginTop="5dp"
    android:background="#FFF000"
    android:gravity="center_horizontal"
    android:text="0"
    android:textAlignment="center"
    android:textColor="?attr/colorPrimary"
    android:textSize="16sp"
    android:textStyle="bold"
    app:layout_constraintBottom_toTopOf="@id/btn_count"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@id/btn_toast" />

```

### Trích xuất tài nguyên chuỗi

Thay vì mã hóa phần cứng các chuỗi, một thực tiễn tốt nhất là sử dụng tài nguyên chuỗi, đại diện cho các chuỗi đó. Việc lưu trữ các chuỗi trong một tệp riêng biệt giúp quản lý dễ dàng hơn, đặc biệt nếu bạn sử dụng chúng nhiều lần. Ngoài ra, chuỗi tài nguyên bị bắt buộc phải dịch và bản địa hóa ứng dụng của bạn, vì bạn cần tạo một chuỗi tài nguyên tệp

1. Nhấp một lần vào từ "Toast" (cảnh báo được đánh dấu đầu tiên).
2. Nhấn Alt-Enter trong Windows hoặc Option-Enter trong macOS và Extract string resource từ menu bật lên.
3. Nhập button\_label\_toast cho Resource name.
4. Nhấp vào OK . Tài nguyên chuỗi được tạo trong tệp values/res/string.xml và chuỗi trong mã của bạn được thay thế bằng tham chiếu đến tài nguyên: @string/button\_label\_toast
5. Trích xuất các chuỗi còn lại: button\_label\_count cho "Count" và count\_initial\_value cho "0" .

6. Trong ngăn Project > Android, hãy mở rộng các value trong res , sau đó nhấp đúp vào strings.xml để xem tài nguyên chuỗi của bạn trong tệp strings.xml:

```
<resources>
    <string name="app_name">Hello World</string>
    <string name="button_label_toast">Toast</string>
    <string name="button_label_count">Count</string>
    <string name="count_initial_value">0</string>
</resources>
```

7. Bạn cần một chuỗi khác để sử dụng trong tác vụ tiếp theo hiển thị thông báo. Thêm vào tệp strings.xml một tài nguyên chuỗi khác có tên toast\_message cho cụm từ "Hello Toast!":

```
<resources>
    <string name="app_name">Hello World</string>
    <string name="button_label_toast">Toast</string>
    <string name="button_label_count">Count</string>
    <string name="count_initial_value">0</string>
    <string name="toast_message">Hello Toast!</string>
</resources>
```

f. Thêm bộ điều khiển onClick cho nút Trong tác vụ này, bạn thêm một phương thức Java cho mỗi Button trong MainActivity thực thi khi người dùng nhấn vào Button

Thêm thuộc tính onClick và bộ điều khiển cho mỗi nút Trình xử lý nhấp chuột là một phương thức được gọi khi người dùng nhấp hoặc nhấn vào phần tử giao diện người dùng có thể nhấp vào. Trong Android Studio, bạn có thể chỉ định tên của phương thức trong trường onClick trong ngăn Attributes của tab Design. Bạn cũng có thể chỉ định tên của phương thức xử lý trong trình soạn thảo XML bằng cách thêm thuộc tính android:onClick vào nút . Bạn sẽ sử dụng phương thức thứ hai vì bạn chưa tạo các phương thức xử lý và trình soạn thảo XML cung cấp một cách tự động để tạo các phương thức đó

1. Với trình chỉnh sửa XML đang mở (tab Văn bản), hãy tìm Nút với android:id được đặt thành button\_toast

2. Thêm thuộc tính android:onClick vào cuối phần tử button\_toast sau thuộc tính cuối cùng và trước chỉ báo kết thúc />:

3. Nhấp vào biểu tượng bóng đèn màu đỏ xuất hiện bên cạnh thuộc tính. Chọn Tạo trình xử lý nhấp chuột , chọn MainActivity và nhấp vào OK . Nếu biểu tượng bóng đèn màu đỏ không xuất hiện, hãy nhấp vào tên phương thức ( "showToast" ). Nhấn Alt-Enter (Option-Enter trên máy Mac), chọn Tạo 'showToast(view)' trong MainActivity và

nhấp vào OK . Hành động này tạo sơ khai phương thức giữ chỗ cho phương thức showToast() trong MainActivity , như được hiển thị ở cuối các bước này.

4. Lặp lại hai bước cuối cùng với nút button\_count : Thêm thuộc tính android:onClick vào cuối và thêm trình xử lý nhấp chuột:

```
android:onClick="showToast" />
```

```
android:onClick="countUp"/>
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.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:id="@+id/main"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".MainActivity" >

    <Button
        android:id="@+id/btn_count"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginBottom="16dp"
        android:text="@string/button_label_count"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        android:onClick="countUp"/>

    <Button
        android:id="@+id/btn_toast"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginTop="28dp"
        android:text="@string/button_label_toast"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintHorizontal_bias="0.498"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toTopOf="parent"
        android:onClick="showToast" />

    <TextView
```

```
        android:id="@+id/show_count"
        android:layout_width="0dp"
        android:layout_height="0dp"
        android:layout_marginTop="5dp"
        android:background="#FFF000"
        android:gravity="center_horizontal"
        android:text="@string/count_initial_value"
        android:textAlignment="center"
        android:textColor="?attr/colorPrimary"
        android:textSize="160sp"
        android:textStyle="bold"
        app:layout_constraintBottom_toTopOf="@id/btn_count"
        app:layout_constraintEnd_toEndOf="parent"
        app:layout_constraintStart_toStartOf="parent"
        app:layout_constraintTop_toBottomOf="@id/btn_toast" />
    </androidx.constraintlayout.widget.ConstraintLayout>
```

Mở tệp MainActivity.java

5. Nếu MainActivity.java chưa mở, hãy mở rộng java trong chế độ xem Project > Android, mở rộng com.example.android.hellotoast, sau đó nhấp đúp vào MainActivity. Trình chỉnh sửa mã xuất hiện cùng với mã trong MainActivity:

```
package android.example.com.helloworld;

import android.os.Bundle;
import android.util.Log;
import android.view.View;

import androidx.activity.EdgeToEdge;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.graphics.Insets;
import androidx.core.view.ViewCompat;
import androidx.core.view.WindowInsetsCompat;

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        EdgeToEdge.enable(this);
        setContentView(R.layout.activity_main);
        Log.v("main", "Hello World");
        ViewCompat.setOnApplyWindowInsetsListener(findViewById(R.id.main), (v, insets)
-> {
            Insets systemBars =
insets.getInsets(WindowInsetsCompat.Type.systemBars());
            v.setPadding(systemBars.left, systemBars.top, systemBars.right,
systemBars.bottom);
            return insets;
        }));

        public void showToast(View view) {
        }

        public void countUp(View view) {
        }
    }
}
```

Thiết kế bộ điều khiển cho Toast Button Bây giờ bạn sẽ chỉnh sửa phương thức showToast() — trình xử lý nhấp chuột Nút Toast trong MainActivity — để nó hiển thị một thông báo. Toast cung cấp một cách để hiển thị một thông báo đơn giản trong một cửa sổ bật lên nhỏ. Nó chỉ lấp đầy dung lượng cần thiết cho tin nhắn. Hoạt động hiện tại vẫn hiển thị và tương tác. Toast có thể hữu ích để kiểm tra tính tương tác trong ứng dụng của bạn—thêm thông báo Toast để hiển thị kết quả nhấn vào Nút hoặc thực hiện một hành động.

Làm theo các bước sau để chỉnh sửa trình xử lý nhấp chuột Nút Toast:

1. Xác định vị trí phương thức showToast() mới tạo.

2. Để tạo một thực thể của Toast, hãy gọi phương thức factory `makeText()` trên lớp `Toast`

3. Cung cấp ngữ cảnh của ứng dụng Hoạt động . Vì Toast hiển thị trên đầu giao diện người dùng Hoạt động, hệ thống cần thông tin về Hoạt động hiện tại. Khi bạn đã ở trong ngữ cảnh của Hoạt động mà bạn cần, hãy sử dụng nó như một phím tắt.

4. Cung cấp thông báo để hiển thị, chẳng hạn như tài nguyên chuỗi (`toast_message` bạn đã tạo ở bước trước). Tài nguyên chuỗi `toast_message` được xác định bởi `R.string`

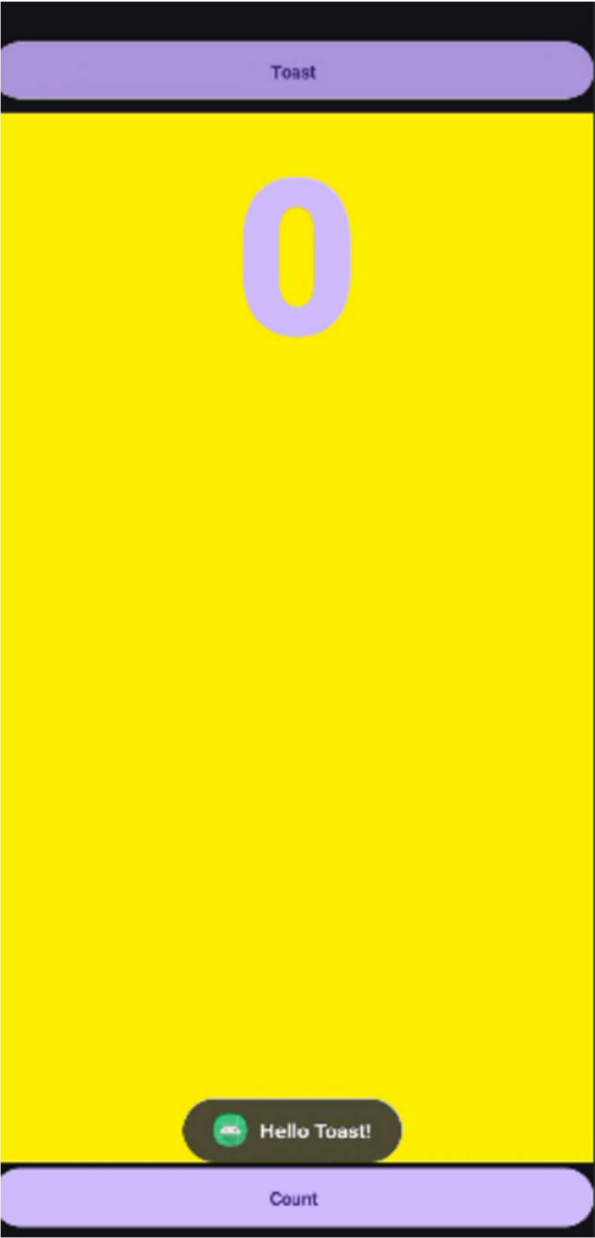
5. Cung cấp thời lượng cho màn hình. Ví dụ: `Toast.LENGTH_SHORT` hiển thị bánh mì nướng trong một thời gian tương đối ngắn Thời lượng của màn hình Toast có thể là `Toast.LENGTH_LONG` hoặc `Toast.LENGTH_SHORT` . Độ dài thực tế là khoảng 3.5 giây đối với Bánh mì nướng dài và 2 giây đối với Bánh mì nướng ngắn.

6. Hiển thị Toast bằng cách gọi `show()` . Sau đây là toàn bộ phương thức `showToast()`:

```
public void showToast(View view) {  
    Toast toast = Toast.makeText(this, R.string.toast_message, Toast.LENGTH_SHORT);  
    toast.show();  
}
```

Chạy chương trình





**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**