

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	22
1.3) Trình chỉnh sửa bố cục	23
1.4) Văn bản và các chế độ cuộn	23
1.5) Tài nguyên có sẵn.....	23
Bài 2) Activities	23
2.1) Activity và Intent	23
2.2) Vòng đời của Activity và trạng thái	23
2.3) Intent ngầm định.....	23
Bài 3) Kiểm thử, gỡ lỗi và sử dụng thư viện hỗ trợ	23
3.1) Trình gỡ lỗi	23
3.2) Kiểm thử đơn vị	23
3.3) Thư viện hỗ trợ	23
CHƯƠNG 2. Trải nghiệm người dùng	24
Bài 1) Tương tác người dùng	24
1.1) Hình ảnh có thể chọn	24
1.2) Các điều khiển nhập liệu	24
1.3) Menu và bộ chọn	24
1.4) Điều hướng người dùng	24
1.5) RecyclerView	24
Bài 2) Trải nghiệm người dùng thú vị.....	24
2.1) Hình vẽ, định kiểu và chủ đề	24
2.2) Thẻ và màu sắc	24

2.3)	Bố cục thích ứng.....	24
Bài 3)	Kiểm thử giao diện người dùng.....	24
3.1)	Espresso cho việc kiểm tra UI	24
CHƯƠNG 3. Làm việc trong nền		24
Bài 1)	Các tác vụ nền.....	24
1.1)	AsyncTask	24
1.2)	AsyncTask và AsyncTaskLoader	24
1.3)	Broadcast receivers	24
Bài 2)	Kích hoạt, lập lịch và tối ưu hóa nhiệm vụ nền.....	24
2.1)	Thông báo	24
2.2)	Trình quản lý cảnh báo	24
2.3)	JobScheduler.....	24
CHƯƠNG 4. Lưu dữ liệu người dùng		25
Bài 1)	Tùy chọn và cài đặt.....	25
1.1)	Shared preferences.....	25
1.2)	Cài đặt ứng dụng.....	25
Bài 2)	Lưu trữ dữ liệu với Room	25
2.1)	Room, LiveData và ViewModel.....	25
2.2)	Room, LiveData và ViewModel.....	25
3.1)	Trình gỡ lỗi	

CHƯƠNG 1. LÀM QUEN

Bài 1) Tạo ứng dụng đầu tiên

1.1) Android Studio và Hello World

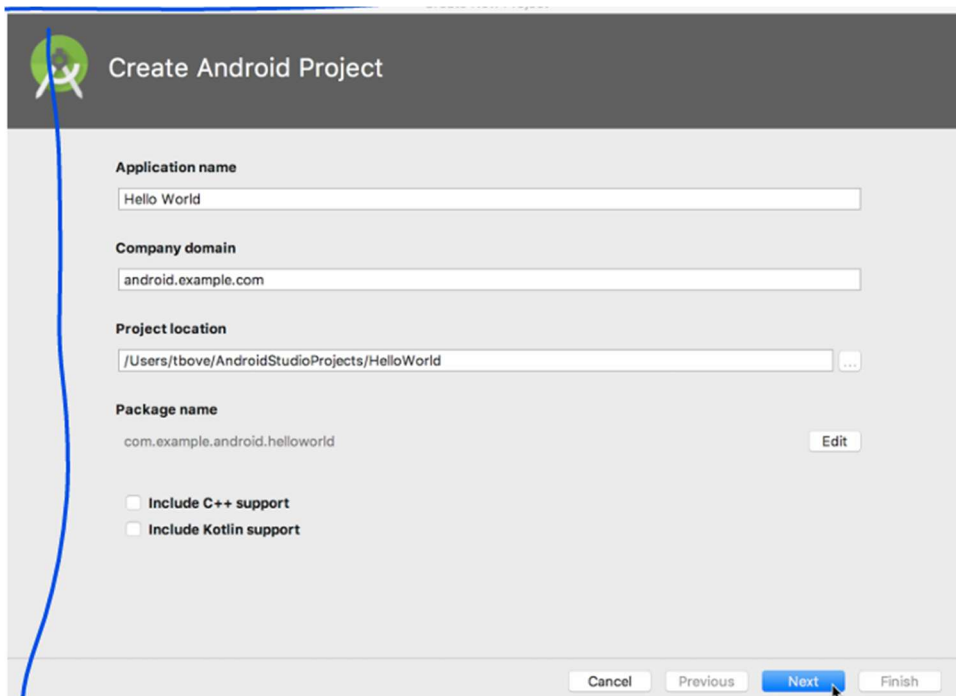
Giới thiệu

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.

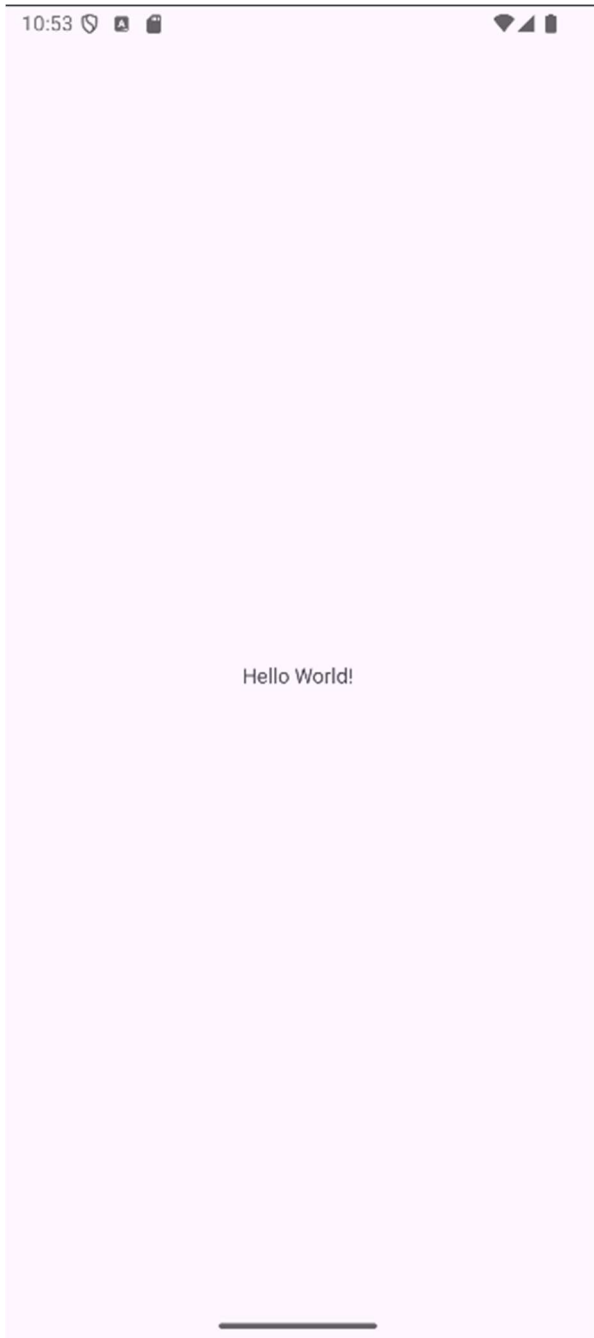
Những gì bạn sẽ làm

- 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 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.

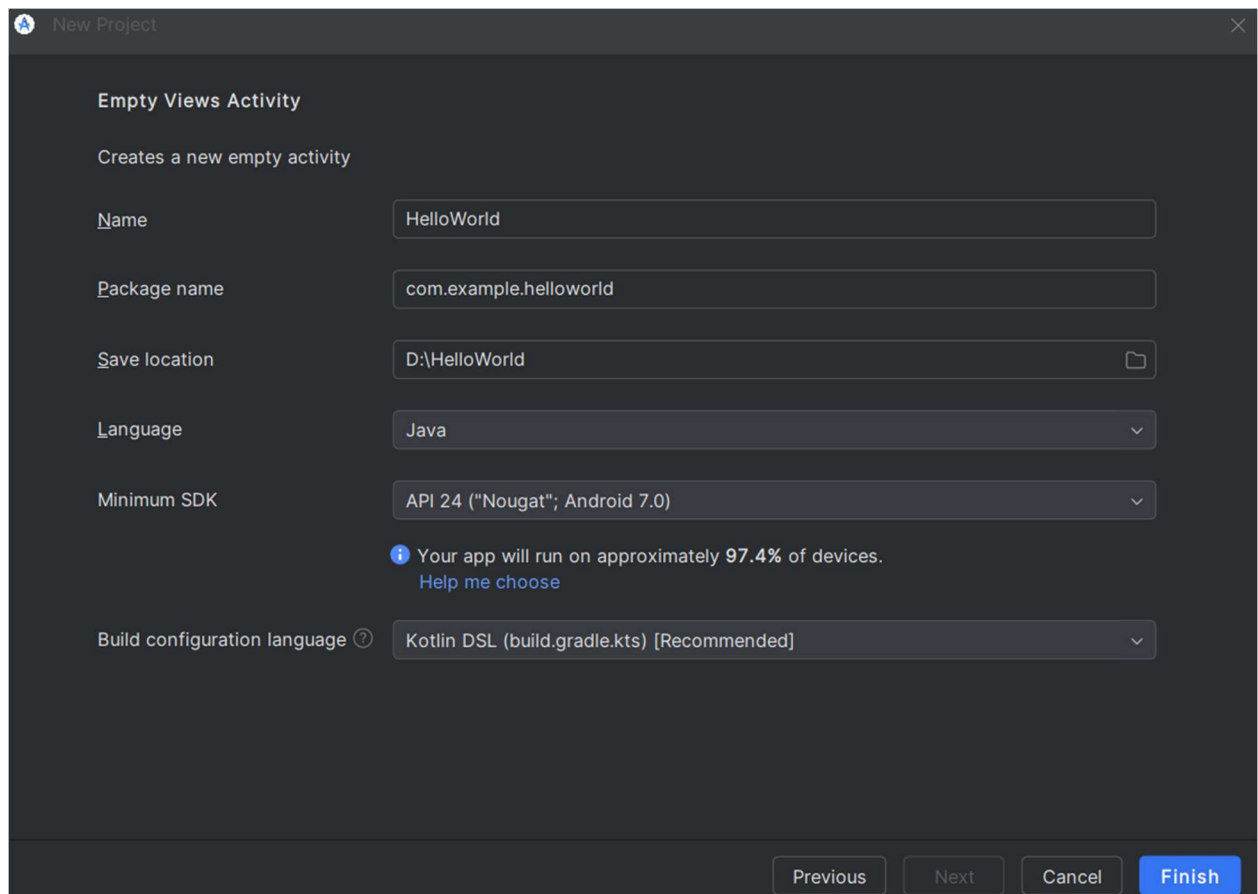
Task 2 : tạo HelloWorld app

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).



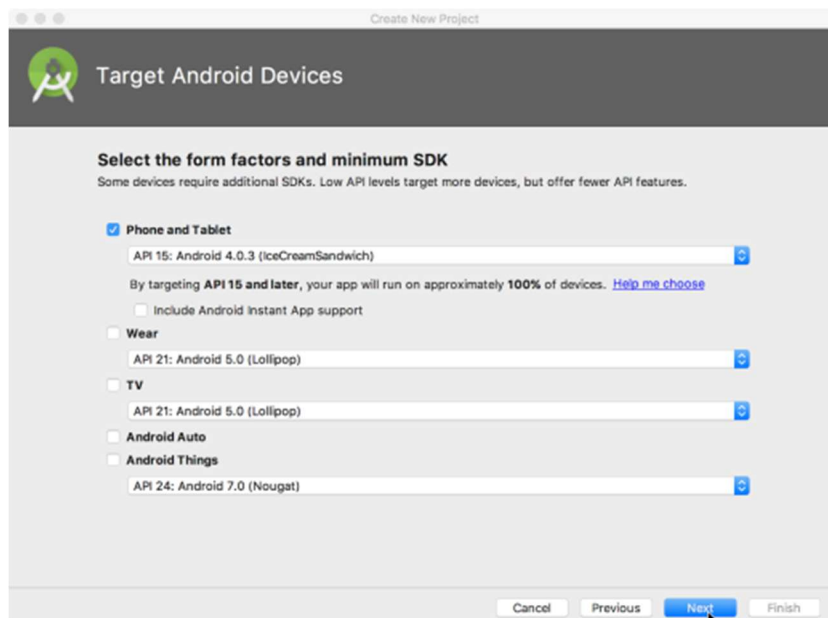
4. Xác minh rằng vị trí dự án mặc định là nơi bạn muốn lưu trữ ứng dụng Hello World và các dự án Android Studio khác, hoặc thay đổi nó thành thư mục bạn muốn.

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:

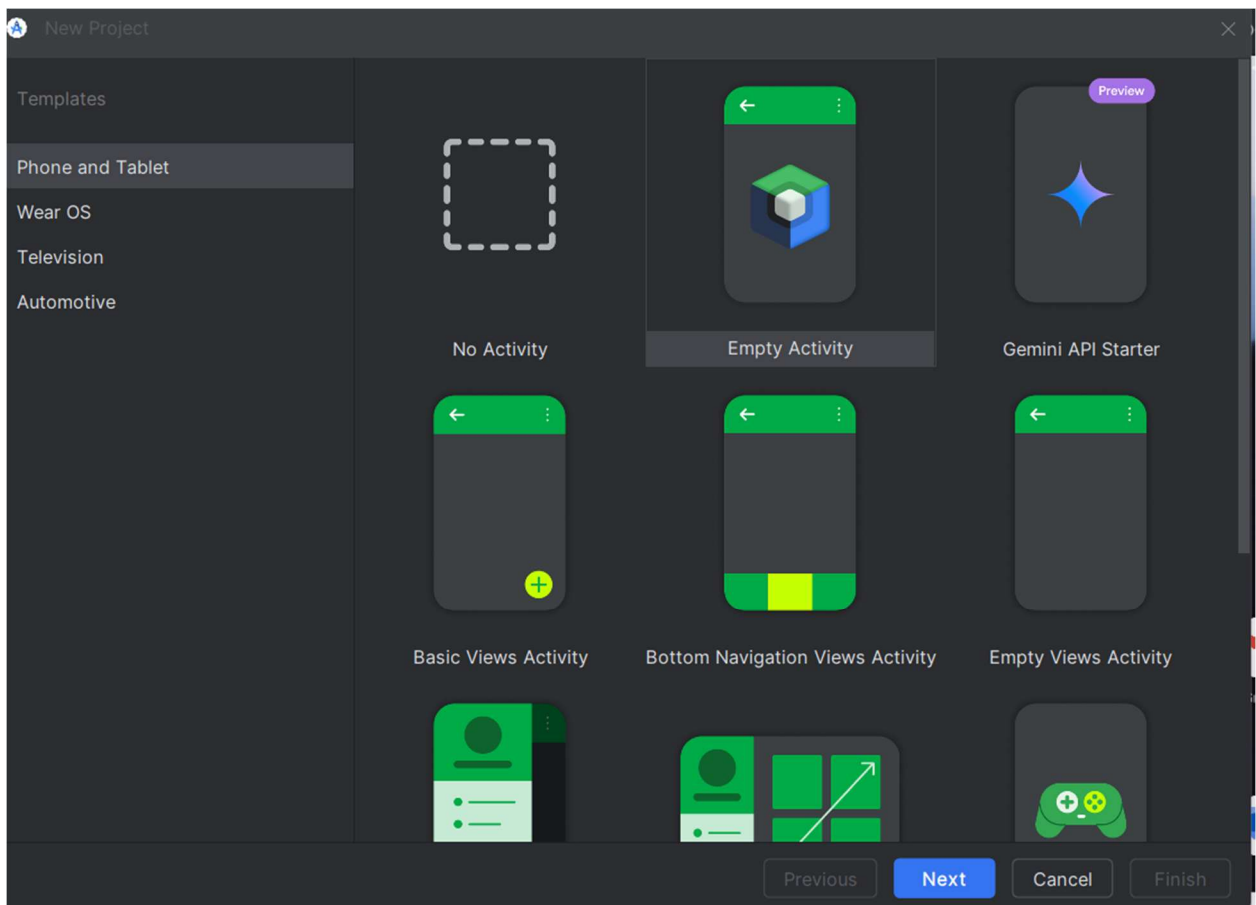
Android 4.0.3 IceCreamSandwich được đặt làm SDK tối thiểu (Minimum SDK); nếu không, hãy sử dụng menu thả xuống để thiết lập.



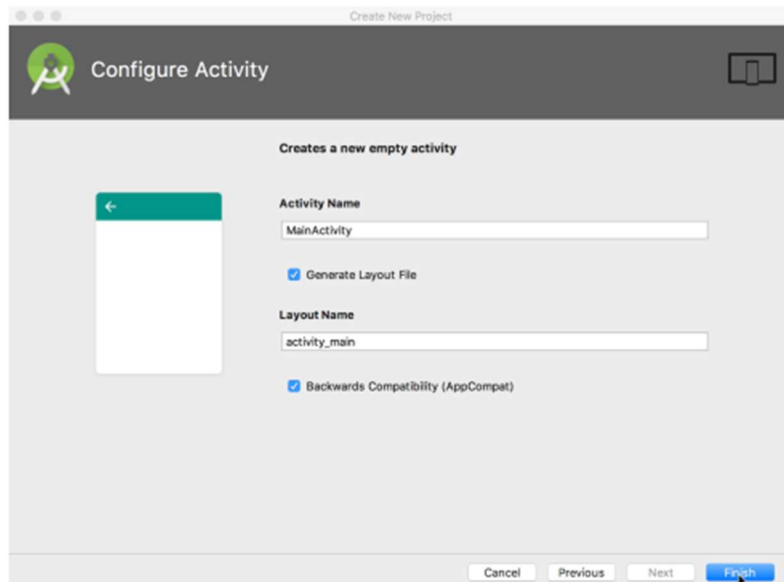
Đây là các cài đặt được sử dụng trong các ví dụ trong các bài học cho khóa học này. Đến thời điểm viết này, các cài đặt này làm cho ứng dụng Hello World của bạn tương thích với 97% thiết bị Android đang hoạt động trên Cốc Cốc Play Store.

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.

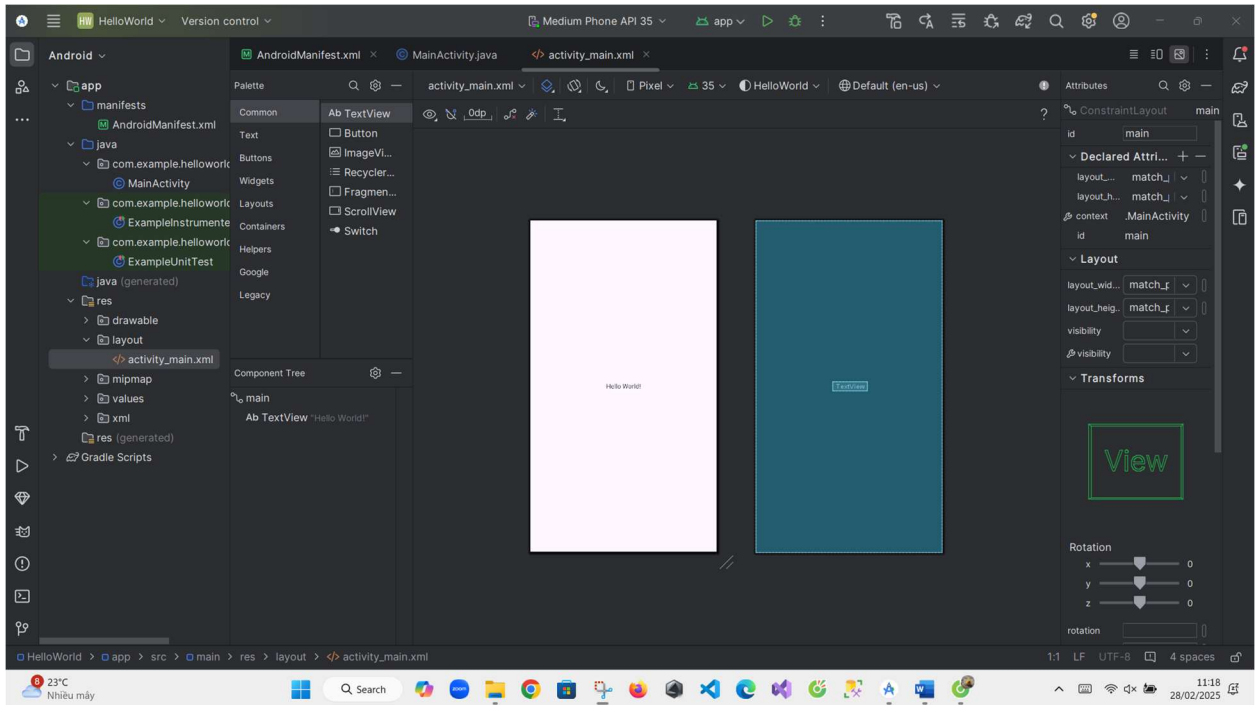
Android Studio tạo một thư mục cho các dự án của bạn và xây dựng dự án bằng Gradle (quá trình này có thể mất một lúc).

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

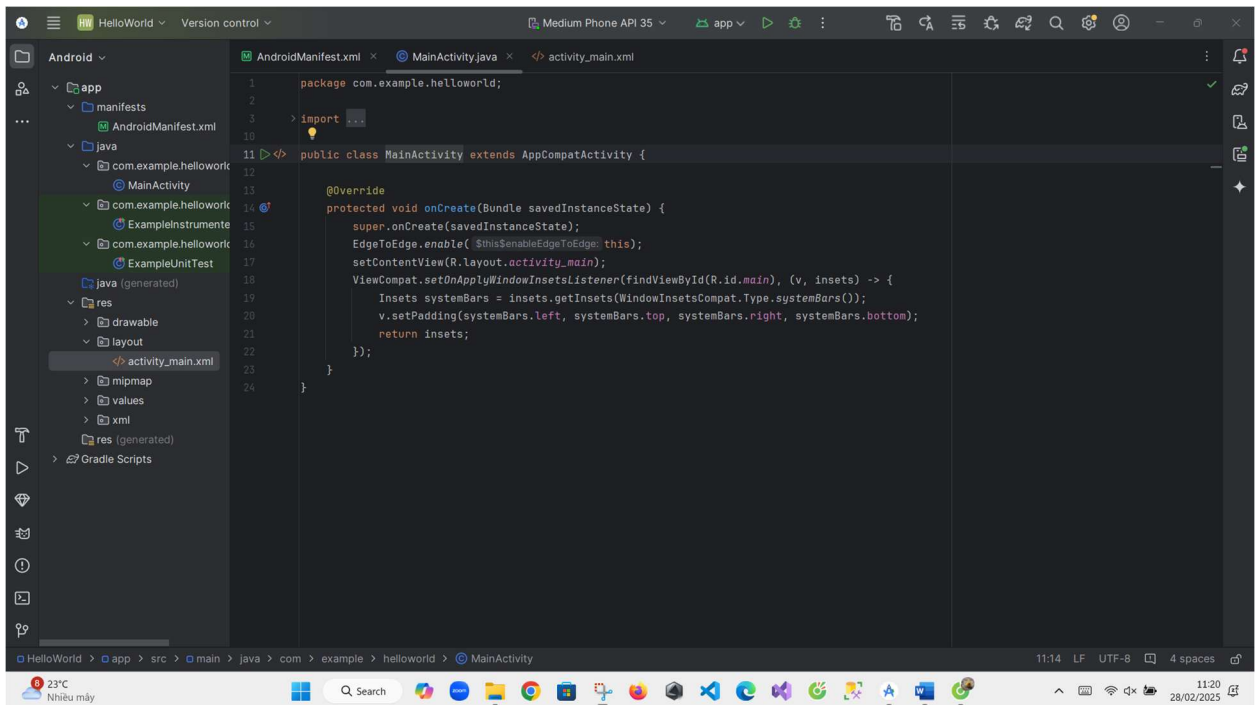
Bạn cũng có thể thấy thông báo **"Tip of the day"** với các phím tắt và mẹo hữu ích khác. Nhấn **Close** để đóng thông báo.

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.



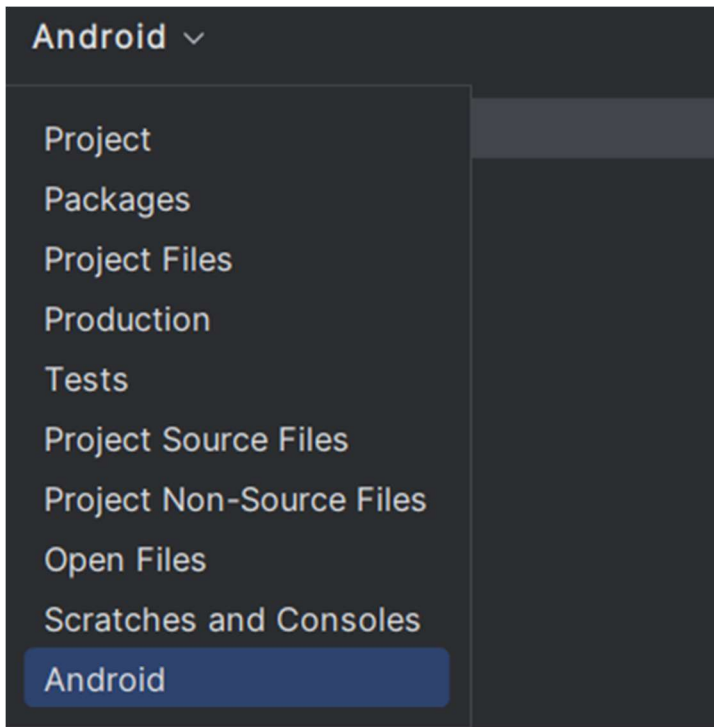
3. Ấn vào MainActivity.java để xem code như dưới đây :



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.

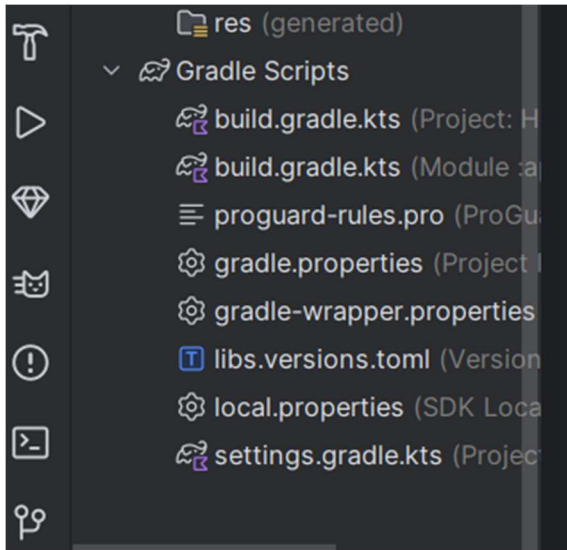


Lưu ý: Chương này và các chương khác gọi ngắn Project khi được đặt ở chế độ Android là Project > Android.

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.

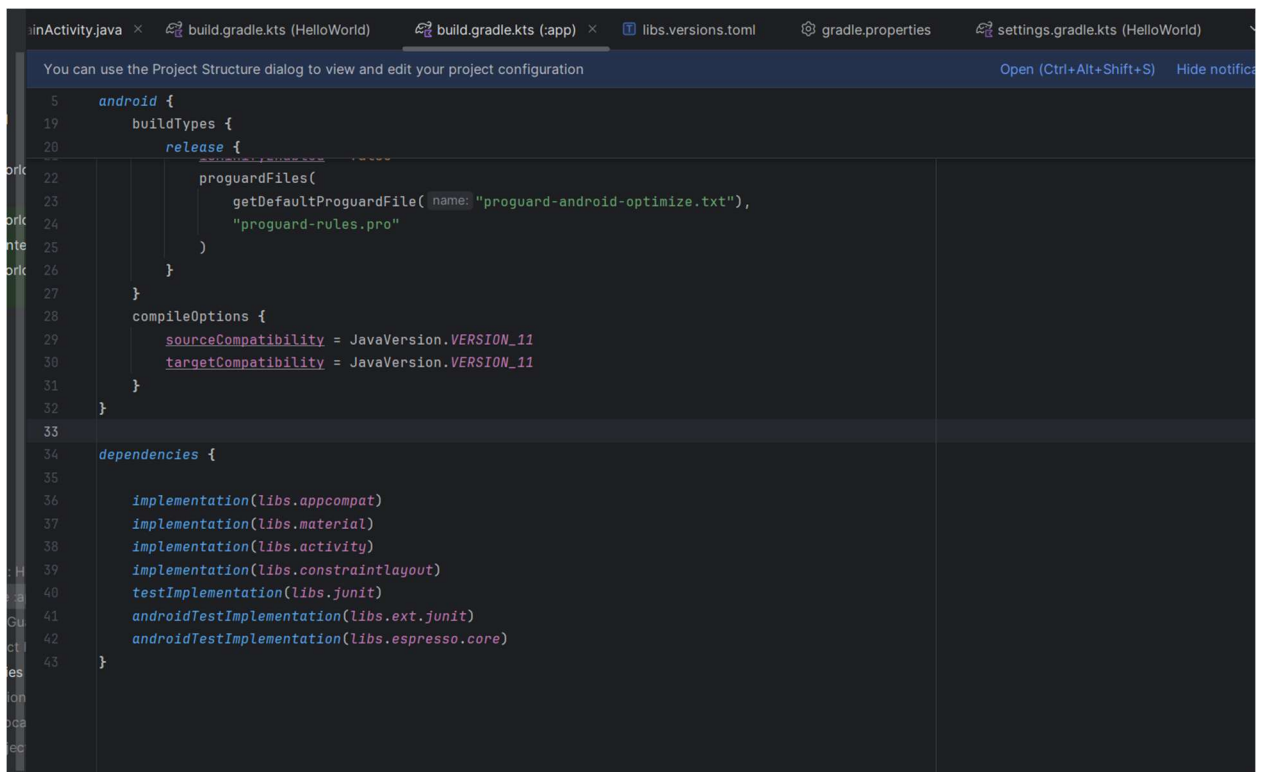
Theo mặc định, tệp build cấp cao nhất sử dụng khối buildscript để xác định kho lưu trữ (repositories) và các phụ thuộc (dependencies) chung cho tất cả mô-đun trong dự án. Khi bạn khai báo một dependency không phải thư viện cục bộ hoặc tập tin nội bộ, Gradle sẽ tìm kiếm tệp trong các kho trực tuyến được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



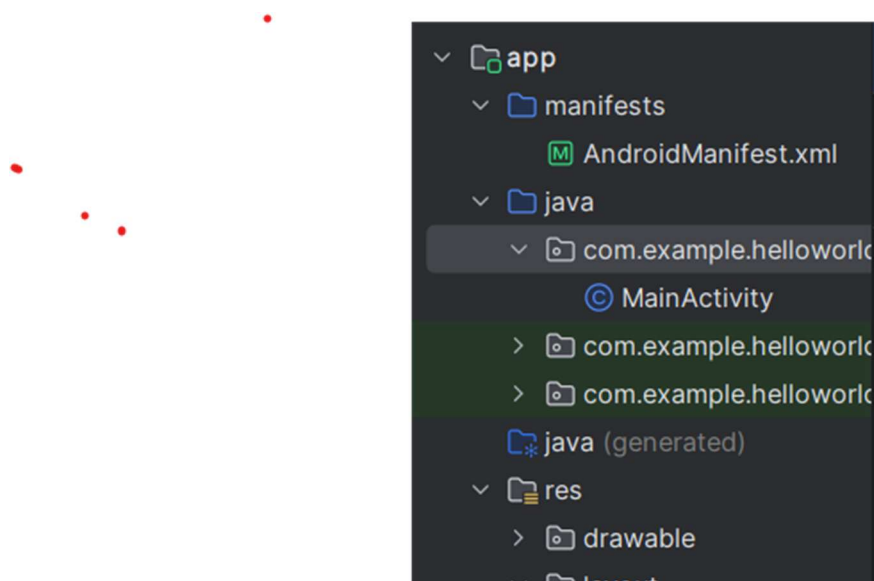
```
5  android {
19      buildTypes {
20          release {
21              // TODO: Add your own signing config for the release build.
22              proguardFiles(
23                  getDefaultProguardFile("proguard-android-optimize.txt"),
24                  "proguard-rules.pro"
25              )
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. Ấn vào hình tam giác để đóng Grandle Scripts.

2.4 Khám phá ứng và thư mục res:

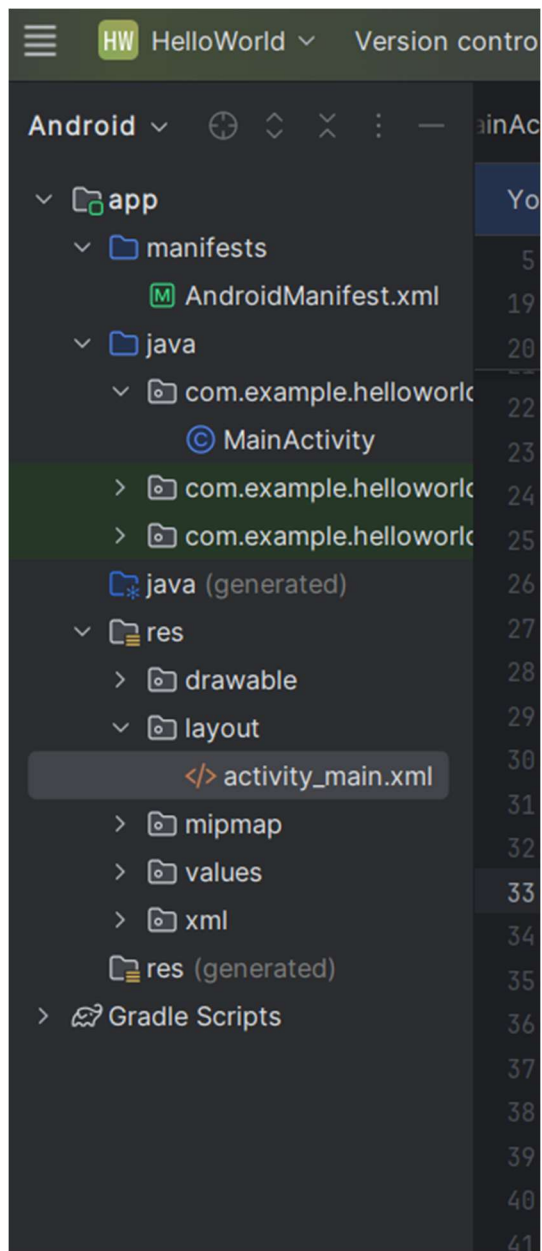
Tất cả code và tài nguyên của ứng dụng nằm trong các thư mục app và res.

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)

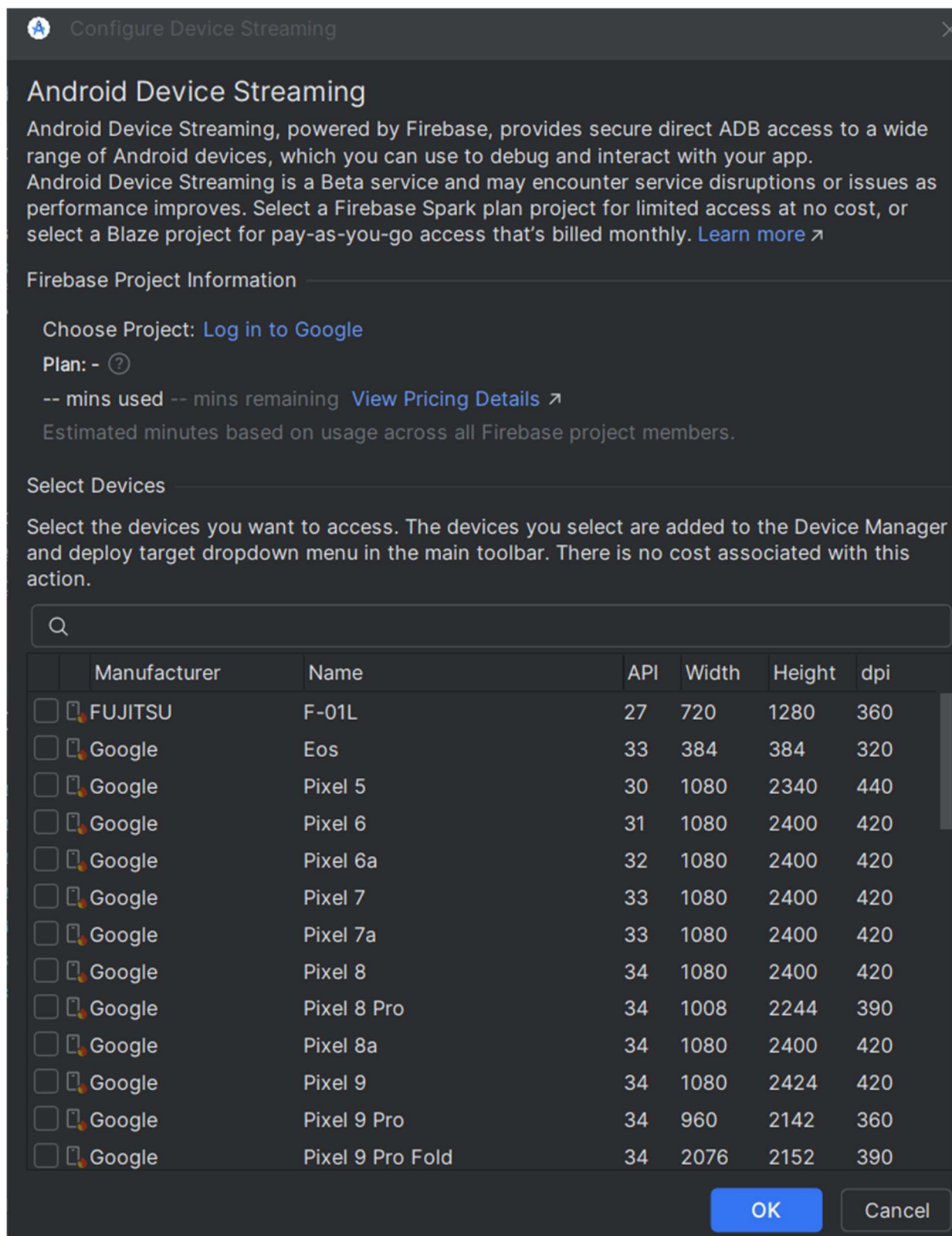
Trong nhiệm vụ này, bạn sẽ sử dụng trình quản lý Thiết bị ảo Android (AVD) để tạo một thiết bị ảo (cũng được gọi là trình mô phỏng) mô phỏng cấu hình cho một loại thiết bị Android cụ thể và sử dụng thiết bị ảo đó để chạy ứng dụng. Lưu ý rằng Trình mô phỏng Android có các yêu cầu bổ sung ngoài các yêu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

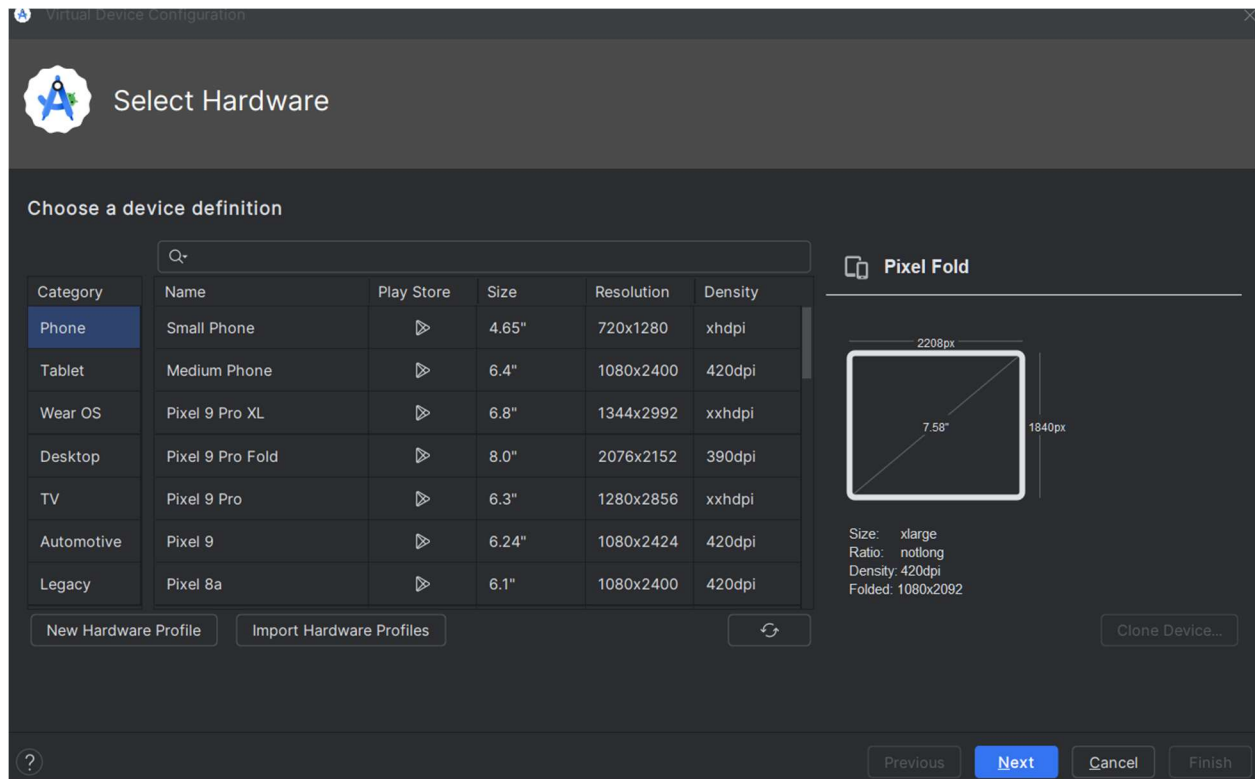
Để chạy trình mô phỏng trên máy tính, bạn phải tạo cấu hình mô tả thiết bị ảo.

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.



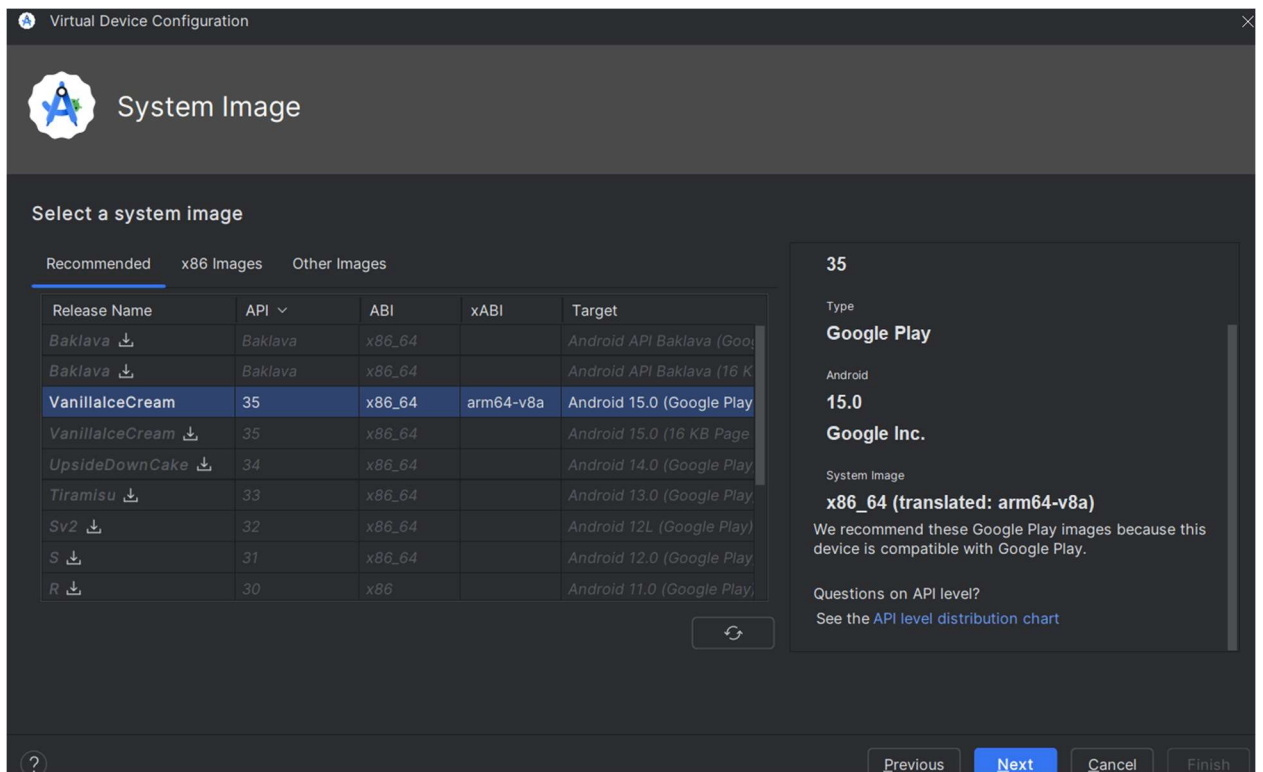
2. Nhấp vào +Tạo thiết bị ảo. Cửa sổ Chọn phần cứng xuất hiện hiển thị danh sách thiết bị phần cứng được cấu hình sẵn. Đối với mỗi thiết bị, bảng cung cấp một cột cho kích

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).



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