**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 4](#_Toc190855147)

[Bài 1) Tạo ứng dụng đầu tiên 4](#_Toc190855148)

[1.1) Android Studio và Hello World 4](#_Toc190855149)

[1.2) Giao diện người dùng tương tác đầu tiên 5](#_Toc190855150)

[1.3) Trình chỉnh sửa bố cục 5](#_Toc190855151)

[1.4) Văn bản và các chế độ cuộn 5](#_Toc190855152)

[1.5) Tài nguyên có sẵn 5](#_Toc190855153)

[Bài 2) Activities 5](#_Toc190855154)

[2.1) Activity và Intent 5](#_Toc190855155)

[2.2) Vòng đời của Activity và trạng thái 5](#_Toc190855156)

[2.3) Intent ngầm định 5](#_Toc190855157)

[Bài 3) Kiểm thử, gỡ lỗi và sử dụng thư viện hỗ trợ 5](#_Toc190855158)

[3.1) Trình gỡ lỗi 5](#_Toc190855159)

[3.2) Kiểm thử đơn vị 5](#_Toc190855160)

[3.3) Thư viện hỗ trợ 5](#_Toc190855161)

[CHƯƠNG 2. Trải nghiệm người dùng 6](#_Toc190855162)

[Bài 1) Tương tác người dùng 6](#_Toc190855163)

[1.1) Hình ảnh có thể chọn 6](#_Toc190855164)

[1.2) Các điều khiển nhập liệu 6](#_Toc190855165)

[1.3) Menu và bộ chọn 6](#_Toc190855166)

[1.4) Điều hướng người dùng 6](#_Toc190855167)

[1.5) RecycleView 6](#_Toc190855168)

[Bài 2) Trải nghiệm người dùng thú vị 6](#_Toc190855169)

[2.1) Hình vẽ, định kiểu và chủ đề 6](#_Toc190855170)

[2.2) Thẻ và màu sắc 6](#_Toc190855171)

[2.3) Bố cục thích ứng 6](#_Toc190855172)

[Bài 3) Kiểm thử giao diện người dùng 6](#_Toc190855173)

[3.1) Espresso cho việc kiểm tra UI 6](#_Toc190855174)

[CHƯƠNG 3. Làm việc trong nền 6](#_Toc190855175)

[Bài 1) Các tác vụ nền 6](#_Toc190855176)

[1.1) AsyncTask 6](#_Toc190855177)

[1.2) AsyncTask và AsyncTaskLoader 6](#_Toc190855178)

[1.3) Broadcast receivers 6](#_Toc190855179)

[Bài 2) Kích hoạt, lập lịch và tối ưu hóa nhiệm vụ nền 6](#_Toc190855180)

[2.1) Thông báo 6](#_Toc190855181)

[2.2) Trình quản lý cảnh báo 6](#_Toc190855182)

[2.3) JobScheduler 6](#_Toc190855183)

[CHƯƠNG 4. Lưu dữ liệu người dùng 7](#_Toc190855184)

[Bài 1) Tùy chọn và cài đặt 7](#_Toc190855185)

[1.1) Shared preferences 7](#_Toc190855186)

[1.2) Cài đặt ứng dụng 7](#_Toc190855187)

[Bài 2) Lưu trữ dữ liệu với Room 7](#_Toc190855188)

[2.1) Room, LiveData và ViewModel 7](#_Toc190855189)

[2.2) Room, LiveData và ViewModel 7](#_Toc190855190)

3.1) Trinfh gowx loi ……………………………………………………………………...

# LÀM QUEN

## Tạo ứng dụng đầu tiên

### 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.

A screenshot of a computer

AI-generated content may be incorrect.

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.

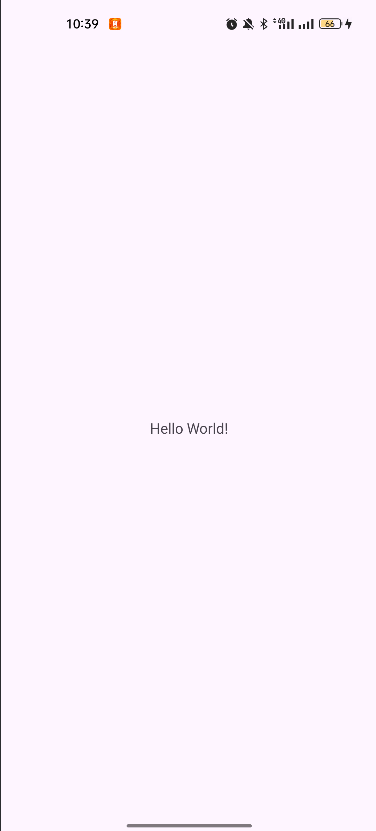
Tạo và chạy ứng dụng **Hello World** trên các thiết bị ảo và vật lý.

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 cài đặt thành công Android Studio, bạn sẽ tạo một dự án mới từ mẫu cho ứng dụng Hello World. Ứng dụng đơn giản này hiển thị chuỗi chữ **"Hello World"** trên màn hình của thiết bị Android, cả ảo và thực tế.



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

Android Studio cung cấp một môi trường phát triển tích hợp hoàn chỉnh (IDE) bao gồm một trình soạn thảo mã nâng cao và một tập hợp các mẫu ứng dụng. Ngoài ra, nó còn có các công cụ phát triển, gỡ lỗi, kiểm thử và đo hiệu năng, giúp bạn phát triển ứng dụng nhanh hơn và dễ dàng hơn. Bạn có thể kiểm thử ứng dụng trên nhiều trình giả lập được cấu hình sẵn hoặc thiết bị di động cá nhân, xây dựng ứng dụng hoàn chỉnh và xuất bản lên Google Play Store.

*Lưu ý:* Android Studio liên tục được cải thiện. Để có thông tin mới nhất về yêu cầu hệ thống và hướng dẫn cài đặt, hãy tham khảo Android Studio.

Android Studio khả dụng cho máy tính chạy Windows, Linux hoặc macOS. Phiên bản OpenJDK (Java Development Kit) mới nhất đã được đi kèm trong Android Studio.

**Các bước cài đặt Android Studio:**

Truy cập vào trang Android Developer và làm theo hướng dẫn để tải xuống và cài đặt Android Studio.

Chấp nhận cấu hình mặc định cho tất cả các bước và đảm bảo rằng tất cả các thành phần cần thiết đã được chọn để cài đặt.

Sau khi hoàn thành cài đặt, Setup Wizard sẽ tải xuống và cài đặt thêm một số thành phần, bao gồm Android SDK. Điều này có thể mất thời gian tùy thuộc vào tốc độ Internet của bạn.

Khi quá trình tải xuống hoàn tất, Android Studio sẽ khởi động và bạn đã sẵn sàng để tạo dự án đầu tiên.

**Xử lý sự cố:** Nếu gặp vấn đề khi cài đặt, hãy kiểm tra Android Studio release notes hoặc nhờ sự giúp đỡ từ giảng viên của bạn.

**Nhiệm vụ 2: Tạo ứng dụng Hello World**

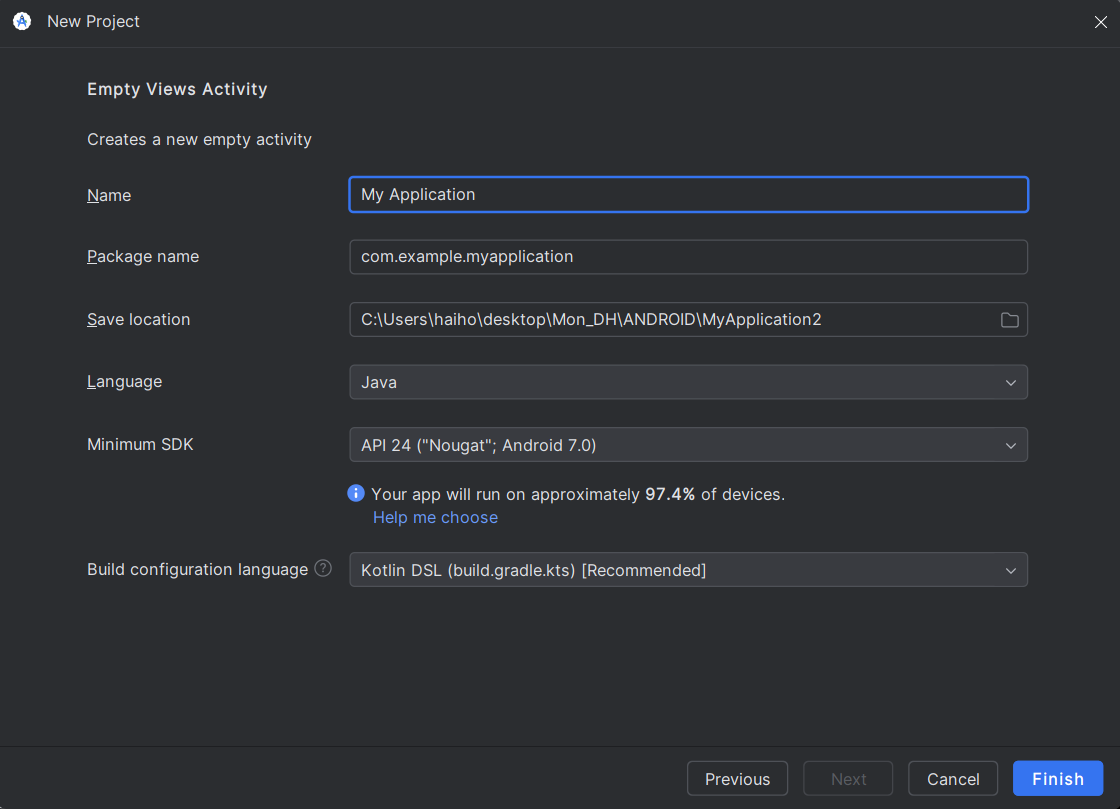
Trong nhiệm vụ này, bạn sẽ tạo một ứng dụng hiển thị "Hello World" để kiểm tra xem Android Studio đã được cài đặt chính xác chưa và học các bước cơ bản khi làm việc với Android Studio.

**Bước 2.1: Tạo dự án ứng dụng**

Mở Android Studio nếu chưa mở.

Trong cửa sổ **Welcome to Android Studio**, nhấp vào **Start a new Android Studio project**.

Trong cửa sổ **Create Android Project**, nhập **Hello World** vào mục **Application name**.



Kiểm tra xem **Project location (Vị trí dự án mặc định)** có đúng là nơi bạn muốn lưu trữ ứng dụng Hello World và các dự án Android Studio khác hay không. Nếu không, thay đổi nó thành thư mục bạn mong muốn.

Chấp nhận mặc định **android.example.com** cho mục **Company Domain (Tên miền công ty)** hoặc tạo một tên miền công ty độc đáo.

Nếu bạn không dự định xuất bản ứng dụng, bạn có thể chấp nhận mặc định. Hãy nhớ rằng việc thay đổi tên gói của ứng dụng sau này sẽ đòi hỏi thêm công việc.

Bỏ chọn các tùy chọn **Include C++ support (Bao gồm hỗ trợ C++)** và **Include Kotlin support (Bao gồm hỗ trợ Kotlin)**, sau đó nhấp vào **Next**.

Trong màn hình **Target Android Devices (Chọn thiết bị mục tiêu Android)**, chọn **Phone and Tablet (Điện thoại và Máy tính bảng)**.

Đảm bảo rằng **API 15: Android 4.0.3 IceCreamSandwich** được đặt làm **Minimum SDK**. Nếu không, sử dụng menu thả xuống để chọn.

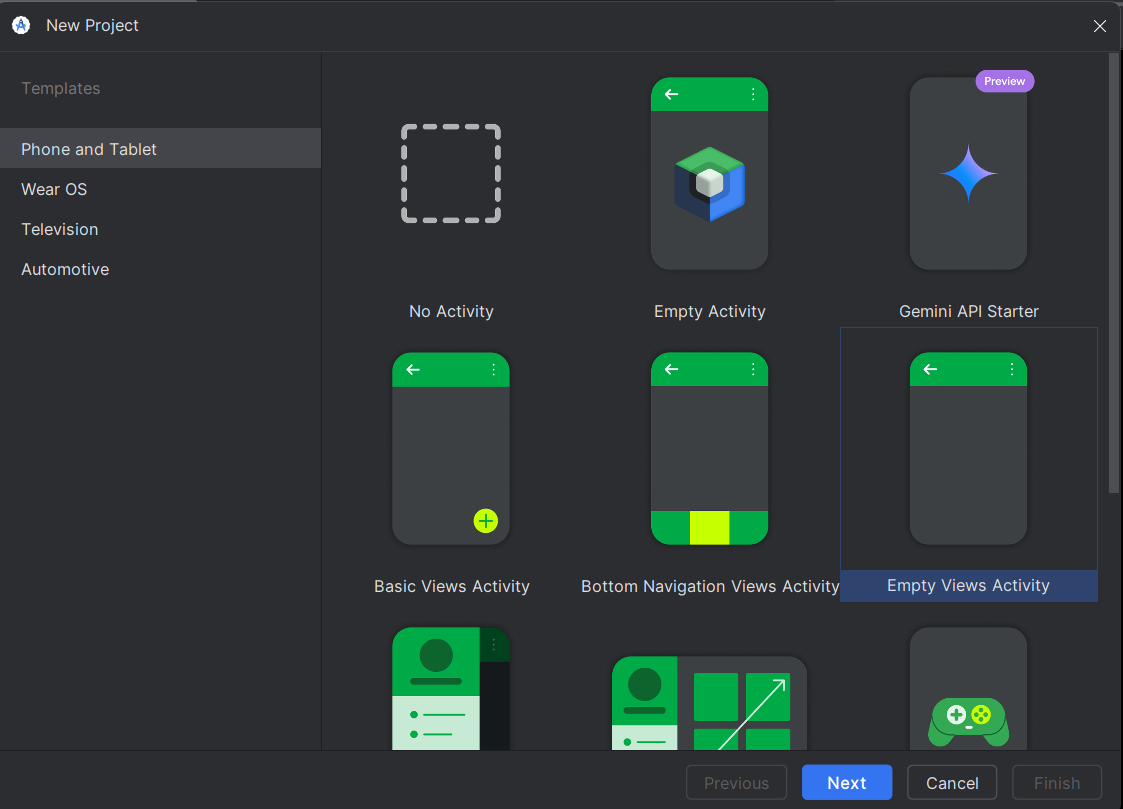
Để trống tùy chọn **Include Instant App support (Hỗ trợ Ứng dụng tức thời)** và tất cả các tùy chọn khác. Sau đó, nhấp vào **Next**.

Nếu dự án của bạn yêu cầu các thành phần bổ sung cho SDK mục tiêu đã chọn, Android Studio sẽ tự động cài đặt chúng.

Cửa sổ **Add an Activity (Thêm một Hoạt động)** sẽ xuất hiện.

**Activity** là một thành phần quan trọng của bất kỳ ứng dụng Android nào, đại diện cho một hành động tập trung mà người dùng có thể thực hiện. Mỗi **Activity** thường được liên kết với một bố cục, định nghĩa cách các phần tử giao diện người dùng (UI) 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 (Hoạt động trống)** như hình dưới và nhấp vào **Next**.



10. Màn hình **Configure Activity (Cấu hình Hoạt động)** sẽ xuất hiện (nội dung hiển thị sẽ khác nhau tùy theo mẫu bạn đã chọn ở bước trước).

Theo mặc định, **Activity (Hoạt động)** trống do mẫu cung cấp được đặt tên là **MainActivity**.

Bạn có thể thay đổi tên này nếu muốn, nhưng bài học này sử dụng **MainActivity**

11. Đảm bảo rằng tùy chọn **Generate Layout file (Tạo tệp bố cục)** đã được chọn. Tên bố cục mặc định là **activity\_main**. Bạn có thể thay đổi tên này nếu muốn, nhưng bài học này sử dụng **activity\_main**.

12. Đảm bảo rằng tùy chọn **Backwards Compatibility (App Compat) (Tương thích ngược)** đã được chọn. Điều này đảm bảo ứng dụng của bạn sẽ tương thích với các phiên bản Android trước đây.

13. Nhấp vào **Finish (Hoàn tất)**

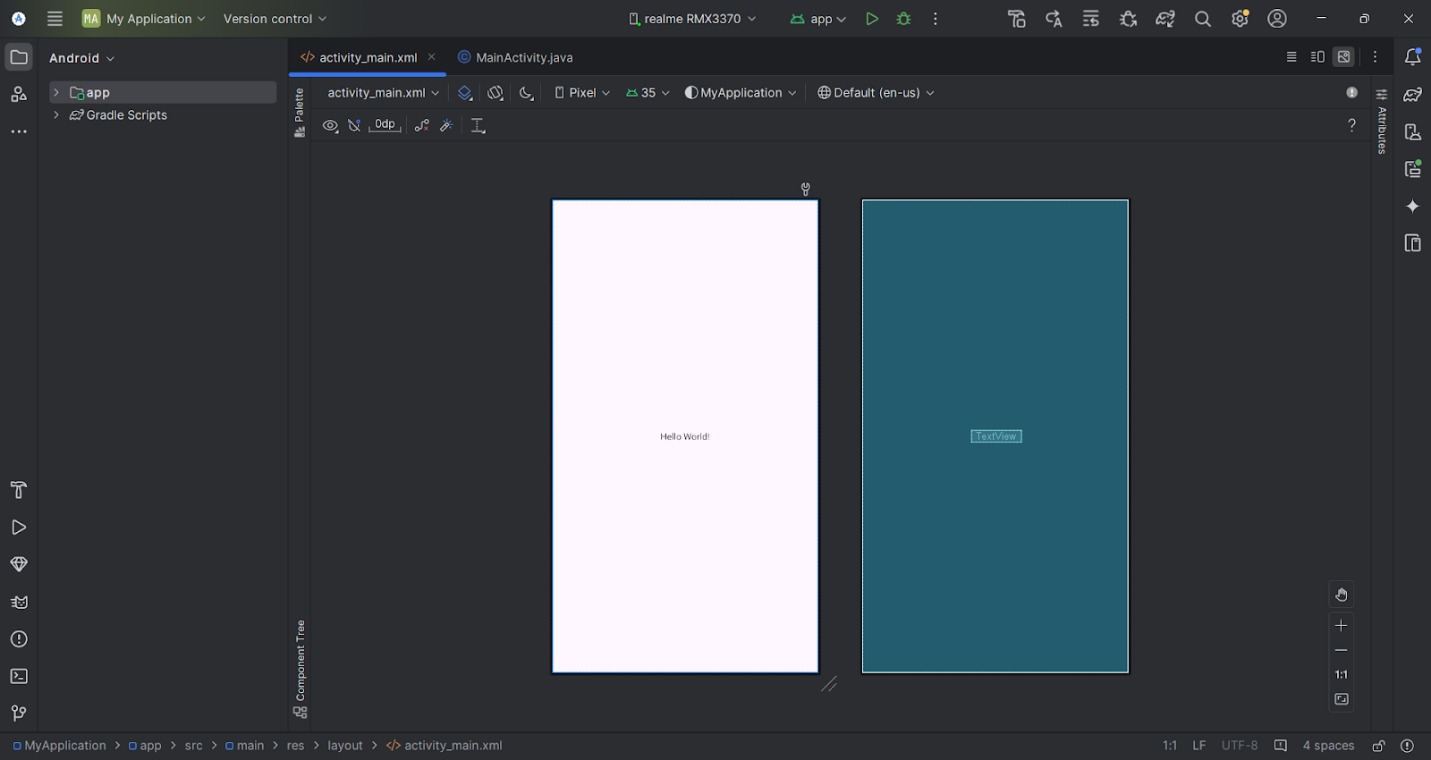
Android Studio tạo một thư mục cho các dự án của bạn và xây dựng dự án bằng Gradle (quá trình này có thể mất một vài phút).

**Mẹo:** Xem trang dành cho nhà phát triển Cấu hình bản dựng để biết thông tin chi tiết.  
Bạn cũng có thể thấy thông báo "Mẹo trong ngày" với các phím tắt và các mẹo hữu ích khác. Nhấn **Close** để đóng thông báo.

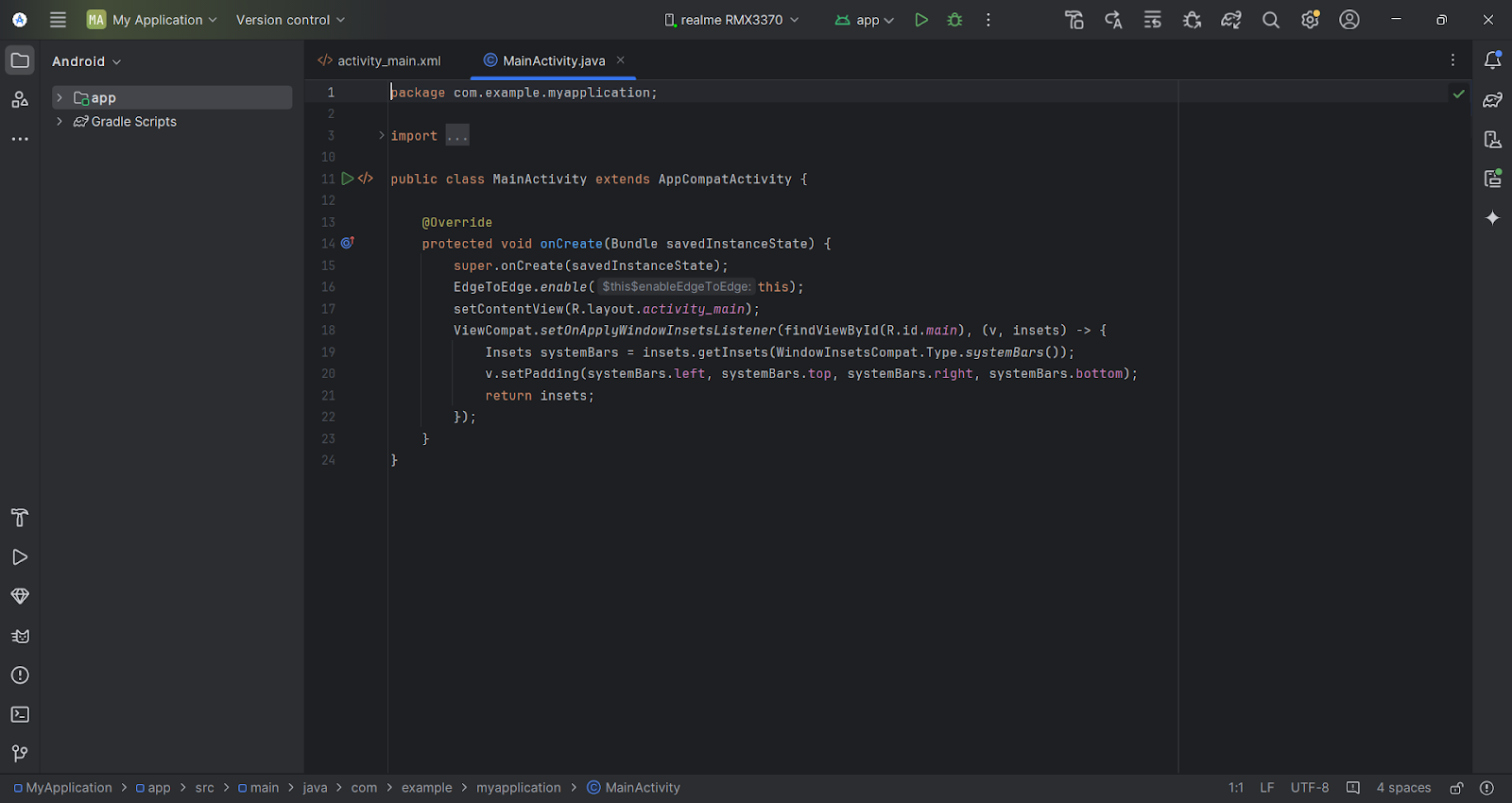
Trình chỉnh sửa Android Studio sẽ xuất hiện. Hãy làm theo các bước sau:

Nhấp vào tab **activity\_main.xml** để mở trình chỉnh sửa giao diện.

Nhấp vào tab **Design** trong trình chỉnh sửa giao diện (nếu chưa được chọn) để hiển thị bản xem trước đồ họa của bố cục như hình dưới đây.



3. Nhấp vào tab **MainActivity.java** để mở trình chỉnh sửa mã như hình dưới đây.



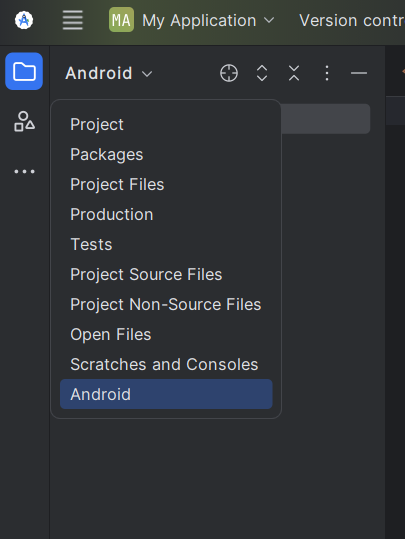
**2.2 Khám phá khung Project > Android**

Trong thực tế này, bạn sẽ khám phá cách tổ chức dự án trong Android Studio.

Nếu chưa chọn, hãy nhấp vào tab **Project** trong cột tab dọc ở phía bên trái của

Cửa sổ Android Studio. Ngăn Dự án xuất hiện.

2. Để xem dự án trong hệ thống phân cấp dự án Android tiêu chuẩn, hãy chọn **Android** từ menu bật lên ở đầu ngăn Dự án, như hình dưới đây.

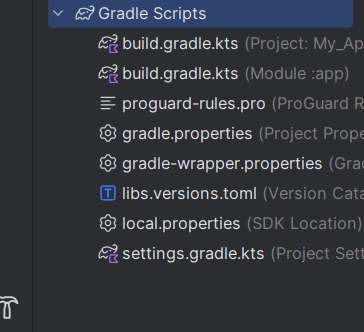


**Lưu ý:** Chương này và các chương khác đề cập đến ngăn Dự án, khi được đặt thành **Android**, làngăn **Dự án > Android**.

**2.3 Khám phá thư mục Tập lệnh Gradle**

Hệ thống bản dựng Gradle trong Android Studio giúp bạn dễ dàng đưa các tệp nhị phân bên ngoài hoặc các mô-đun thư viện khác vào bản dựng dưới dạng phần phụ thuộc.

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



Làm theo các bước sau để khám phá hệ thống Gradle:

Nếu thư mục **Gradle Scripts** không được mở rộng, hãy nhấp vào hình tam giác để mở rộng nó. Thư mục này chứa tất cả các tệp cần thiết cho hệ thống xây dựng.

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

Đây là nơi bạn sẽ tìm thấy các tùy chọn cấu hình chung cho tất cả các mô-đun tạo nên dự án của bạn. Mỗi dự án Android Studio đều chứa một tệp bản dựng Gradle cấp cao nhất. Hầu hết thời gian, bạn sẽ không cần thực hiện bất kỳ thay đổi nào đối với tệp này, nhưng vẫn hữu ích khi hiểu nội dung của nó.

Theo mặc định, tệp bản dựng cấp cao nhất sử dụng khối buildscript để xác định kho lưu trữ Gradle và các phần phụ thuộc chung cho tất cả các mô-đun trong dự án. Khi phần phụ thuộc của bạn không phải là thư viện cục bộ hoặc cây tệp, Gradle sẽ tìm kiếm các tệp trong bất kỳ kho lưu trữ trực tuyến nào được chỉ định trong khối kho lưu trữ của tệp này. Theo mặc định, các dự án Android Studio mới khai báo JCenter và Google (bao gồm kho lưu trữ Google Maven) là vị trí kho lưu trữ:

allprojects { kho lưu trữ {

google()

jcenter()

}

}

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

Ngoài tệp build.gradle cấp dự án, mỗi mô-đun có một tệp build.gradle riêng cho phép bạn định cấu hình cài đặt bản dựng cho từng mô-đun cụ thể (ứng dụng HelloWorld chỉ có một mô-đun). Việc định cấu hình các tùy chọn cài đặt bản dựng này cho phép bạn cung cấp các tùy chọn đóng gói tùy chỉnh, chẳng hạn như các loại bản dựng bổ sung và hương vị sản phẩm. Bạn cũng có thể ghi đè các tùy chọn cài đặt trong tệp AndroidManifest.xml hoặc tệp build.gradle cấp cao nhất.

Tệp này thường là tệp cần chỉnh sửa khi thay đổi cấu hình cấp ứng dụng, chẳng hạn như khai báo các phần phụ thuộc trong phần dependencies. Bạn có thể khai báo phần phụ thuộc thư viện bằng cách sử dụng một trong một số cấu hình phần phụ thuộc khác nhau. Mỗi cấu hình phần phụ thuộc cung cấp cho Gradle các hướng dẫn khác nhau về cách sử dụng thư viện. Ví dụ: câu lệnh implementation fileTree(dir: 'libs', include: ['\*.jar']) thêm một

sự phụ thuộc của tất cả các tệp ".jar" bên trong thư mục LIBS .

Sau đây là tệp **build.gradle(Module:app)** cho ứng dụng HelloWorld:

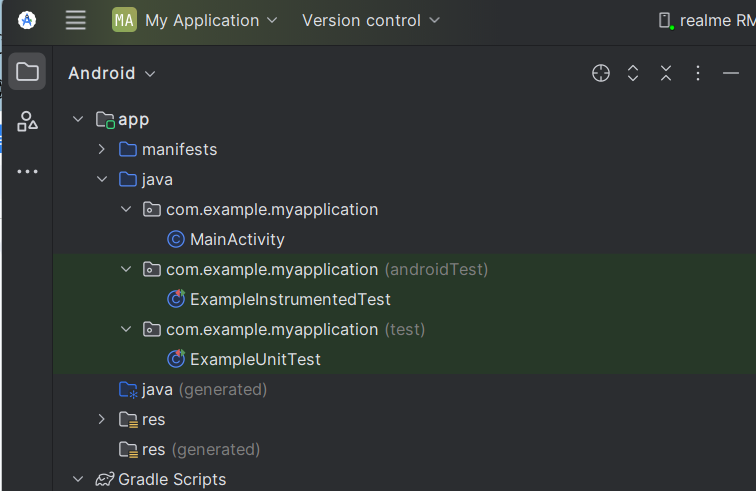
|  |
| --- |
| plugins **{**  alias(*libs*.*plugins*.*android*.*application*)  **}**  *android* **{**  namespace = "com.example.myapplication"  compileSdk = 35  defaultConfig **{**  applicationId = "com.example.myapplication"  minSdk = 24  targetSdk = 35  versionCode = 1  versionName = "1.0"  testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"  **}**  buildTypes **{**  *release* **{**  isMinifyEnabled = false  proguardFiles(  getDefaultProguardFile("proguard-android-optimize.txt"),  "proguard-rules.pro"  )  **}**  **}**  compileOptions **{**  sourceCompatibility = JavaVersion.*VERSION\_11*  targetCompatibility = JavaVersion.*VERSION\_11*  **}**  **}**  *dependencies* **{**  *implementation*(*libs*.*appcompat*)  *implementation*(*libs*.*material*)  *implementation*(*libs*.*activity*)  *implementation*(*libs*.*constraintlayout*)  *testImplementation*(*libs*.*junit*)  *androidTestImplementation*(*libs*.*ext*.*junit*)  *androidTestImplementation*(*libs*.*espresso*.*core*)  **}** |

4. Nhấp vào hình tam giác để đóng **Gradle Scripts**.

**2.4 Khám phá ứng dụng và thư mục res**

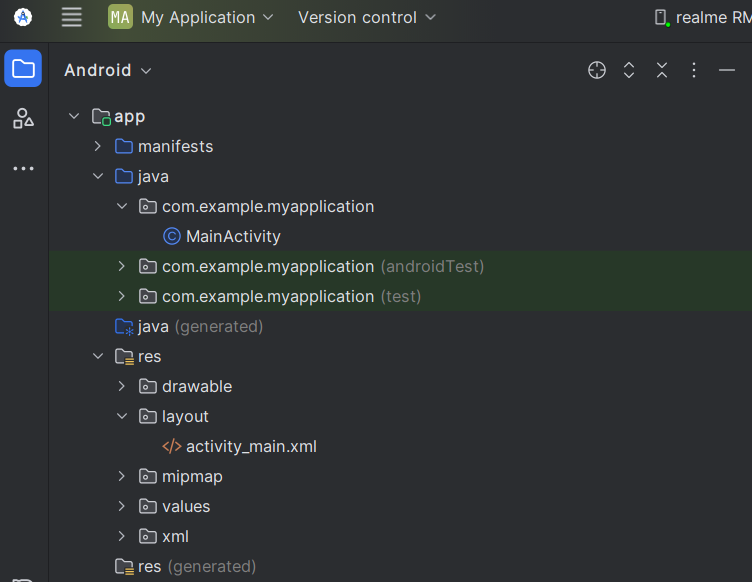
Tất cả mã và tài nguyên cho ứng dụng nằm trong thư mục app và res .

Mở rộng thư mục **app**, thư mục **java** và thư mục **com.example.android.helloworld** để xem tệp java **MainActivity**. Nhấp đúp vào tệp sẽ mở tệp đó trong trình soạn thảo mã.



Thư mục **java** bao gồm các tệp lớp Java trong ba thư mục con, như trong hình trên. Thư mục **c** **om.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 và nó chứa MainActivity.java. Tên của màn hình A (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 bị bỏ qua trong ngăn **Project > Android**).

2. Mở rộng thư mục **res** và thư mục **layout**, đồng thời nhấp đúp vào tệp **activitymain.xml** để mở nó trong trình chỉnh sửa bố cục.



Thư mục **res** chứa các tài nguyên, chẳng hạn như bố cục, chuỗi và hình ảnh. Activity thường được liên kết với bố cục của các chế độ xem giao diện người dùng được định nghĩa dưới dạng tệp XML. Tệp này thường được đặt tên theo Hoạt động của nó.

**2.5 Khám phá thư mục tệp kê khai**

Thư mục manifests chứa các tệp cung cấp thông tin cần thiết về ứng dụng của bạn cho

Hệ thống Android, hệ thống phải có trước khi có thể chạy bất kỳ mã nào của ứng dụng.

Mở rộng thư mục **manifests**.

Mở **tệp AndroidManifest.xml**.

Tệp AndroidManifest.xml mô tả tất cả các thành phần của ứng dụng Android. Tất cả các thành phần cho một ứng dụng, chẳng hạn như mỗi thành phần A , 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 bản kê khai pp.

**Nhiệm vụ 3: Sử dụng thiết bị ảo (trình giả lập)**

Trong tác vụ này, bạn sẽ sử dụng trình quản lý Thiết bị ảo (AVD) để tạo một thiết bị ảo (còn 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. Xin lưu ý rằng Trình mô phỏng Android có các yêu cầu riêng ngoài các yêu cầu hệ thống cơ bản cho Android Studio.

Khi sử dụng Trình quản lý AVD, bạn xác định các đặc điểm phần cứng của thiết bị, cấp độ API, bộ nhớ, giao diện và các thuộc tính khác và lưu thiết bị đó dưới dạng thiết bị ảo. Với thiết bị ảo, bạn có thể kiểm tra ứng dụng trên các 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 thiết bị thực.

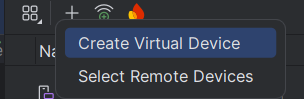
**3.1 Tạo thiết bị ảo Android (AVD)**

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

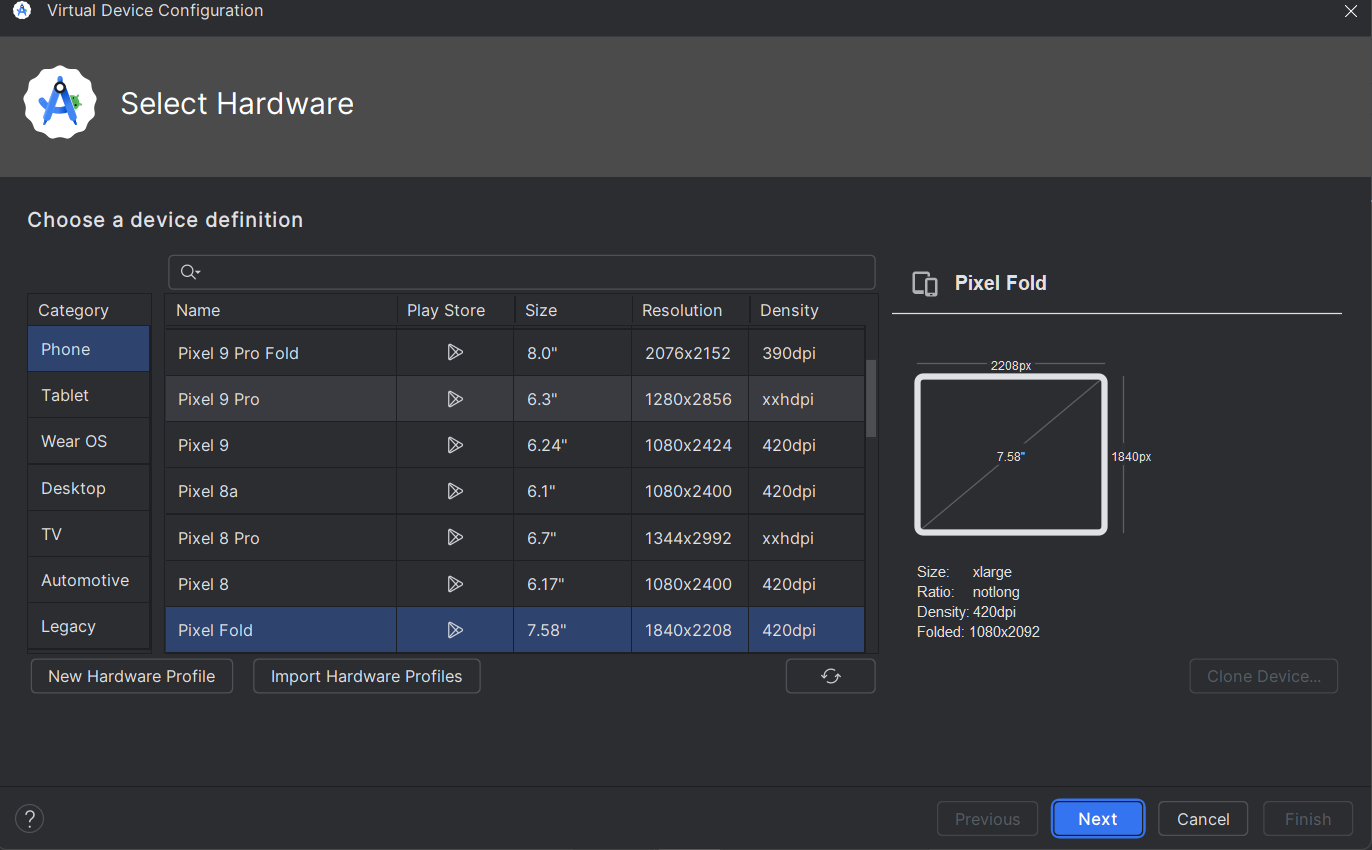
Trong Android Studio, chọn **Tools** **> Trình quản lý Android > AVD** hoặcnhấp vào biểu tượng Trình quản lý AVD

 trên thanh công cụ. Thiết **bị ảo của chúng tôi** xuấthiện. 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.

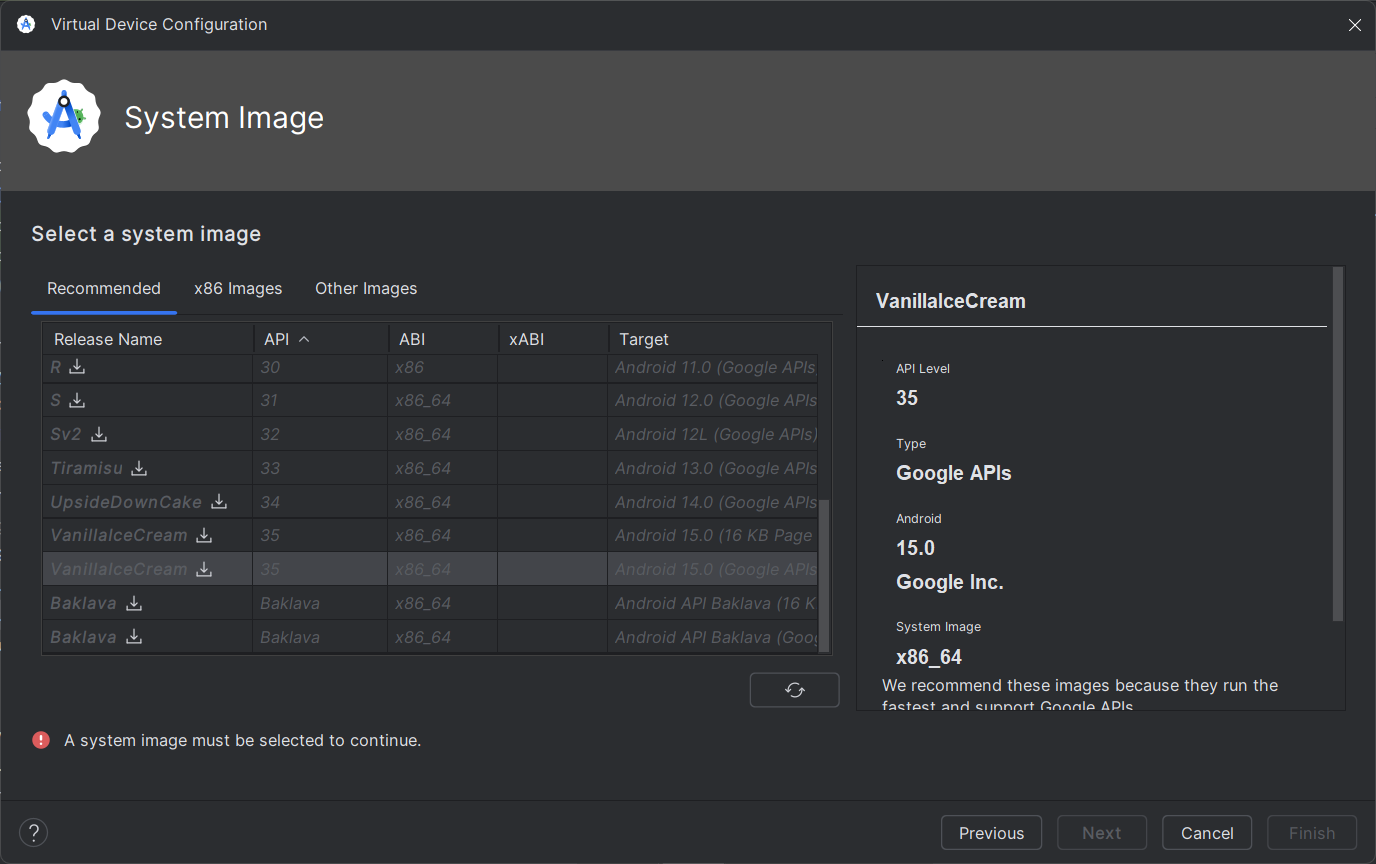


2. Nhấp vào **+** **Tạo thiết bị ảo**.Cửa sổ **Select Hardware** xuất hiện hiển thị danh sách các thiết bị phần cứng được định cấu hình sẵn. Đối với mỗi thiết bị, bảng cung cấp một cột cho kích thước hiển thị đường chéo (**Size**),độ phân giải màn hình tính bằng pixel (**Resolution**)và mật độ pixel (**Density**).



3. Chọn một thiết bị như **Nexus 5x** hoặc **Pixel XL**,và nhấp vào **Next**. **System Image** screen 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**).



Có nhiều phiên bản có sẵn hơn so với hiển thị trong tab **R** **khuyến nghị**. Nhìn vào các tab **x86 Images** và **Other** **Images** để xem chúng.

Nếu liên kết **Download** hiển thị bên cạnh hình ảnh hệ thống bạn muốn sử dụng, nó vẫn chưa được cài đặt. Nhấp vào liên kết để bắt đầu tải xuống và nhấp vào **F** **inish** khi hoàn tất.

5. Sau khi chọn hình ảnh hệ thống, nhấp vào **Next**.Cửa sổ **Thiết** **bị ảo Android (AVD)** xuất hiện. Bạn cũng có thể thay đổi tên của AVD. Kiểm tra cấu hình của bạn và nhấp vào **Finish**.

### Giao diện người dùng tương tác đầu tiên

### Trình chỉnh sửa bố cục

### Văn bản và các chế độ cuộn

### Tài nguyên có sẵn

## Activities

### Activity và Intent

### Vòng đời của Activity và trạng thái

### Intent ngầm định

## Kiểm thử, gỡ lỗi và sử dụng thư viện hỗ trợ

### Trình gỡ lỗi

### Kiểm thử đơn vị

### Thư viện hỗ trợ

# TRẢI NGHIỆM NGƯỜI DÙNG

## Tương tác người dùng

### Hình ảnh có thể chọn

### Các điều khiển nhập liệu

### Menu và bộ chọn

### Điều hướng người dùng

### RecycleView

## Trải nghiệm người dùng thú vị

### Hình vẽ, định kiểu và chủ đề

### Thẻ và màu sắc

### Bố cục thích ứng

## Kiểm thử giao diện người dùng

### Espresso cho việc kiểm tra UI

# LÀM VIỆC TRONG NỀN

## Các tác vụ nền

### AsyncTask

### AsyncTask và AsyncTaskLoader

### Broadcast receivers

## Kích hoạt, lập lịch và tối ưu hóa nhiệm vụ nền

### Thông báo

### Trình quản lý cảnh báo

### JobScheduler

# LƯU DỮ LIỆU NGƯỜI DÙNG

## Tùy chọn và cài đặt

### Shared preferences

### Cài đặt ứng dụng

## Lưu trữ dữ liệu với Room

### Room, LiveData và ViewModel

### Room, LiveData và ViewModel