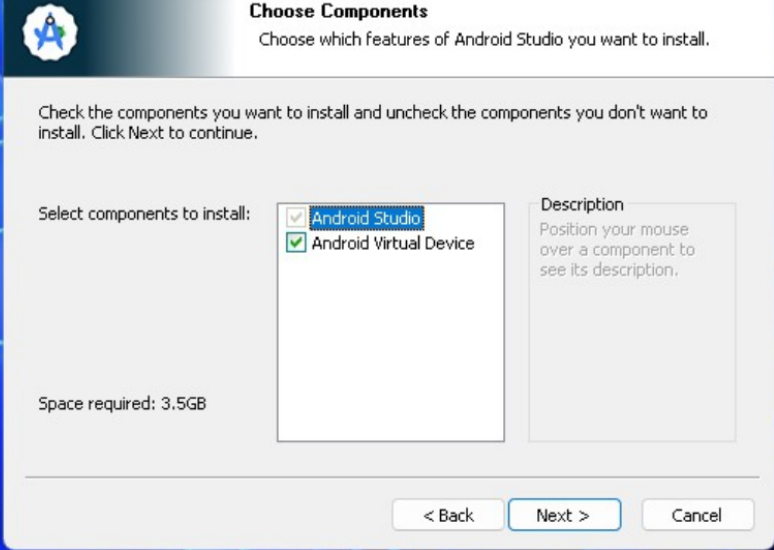
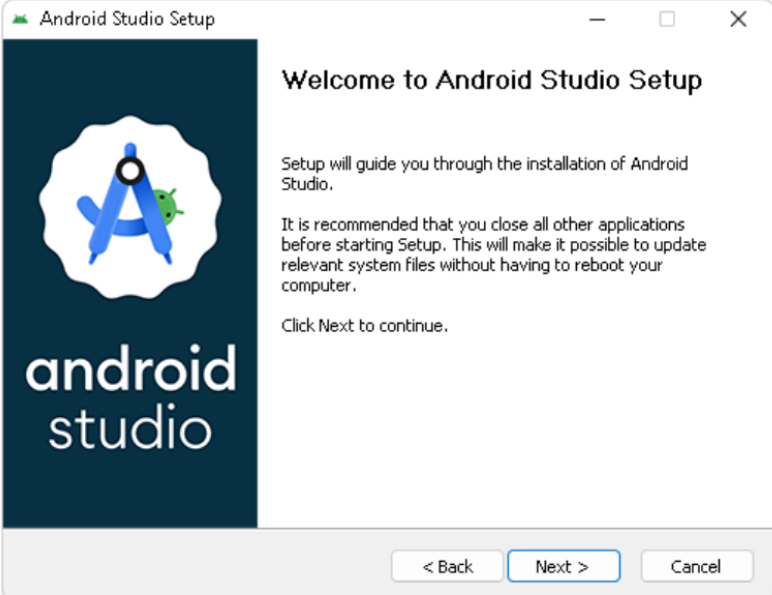
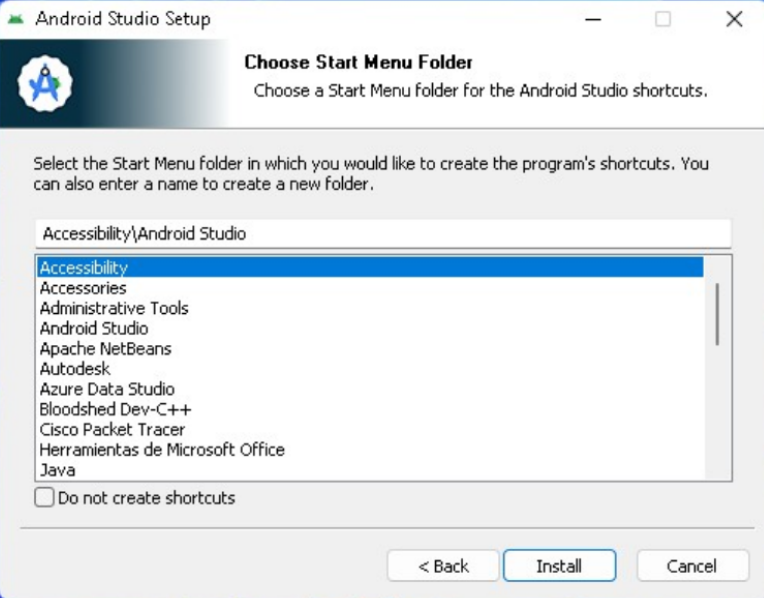
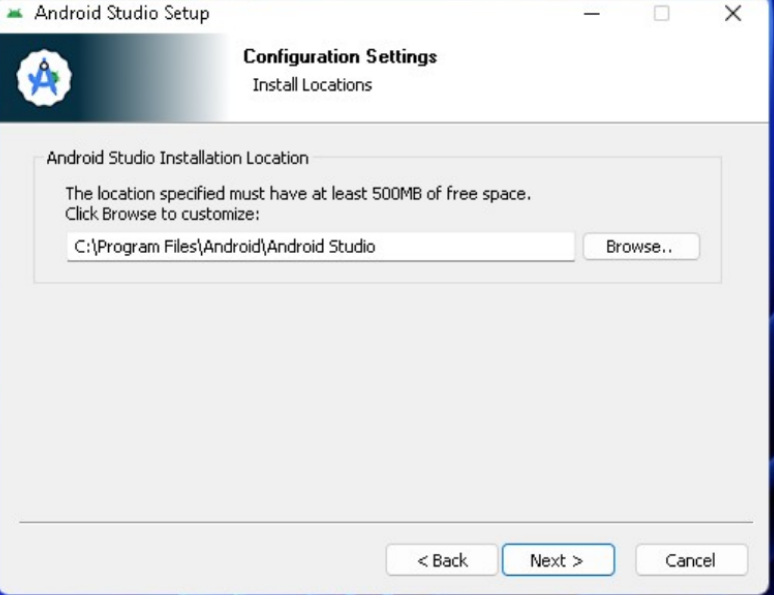
**Tài liệu liên quan đến lập trình android phát triển máy quét qr code**

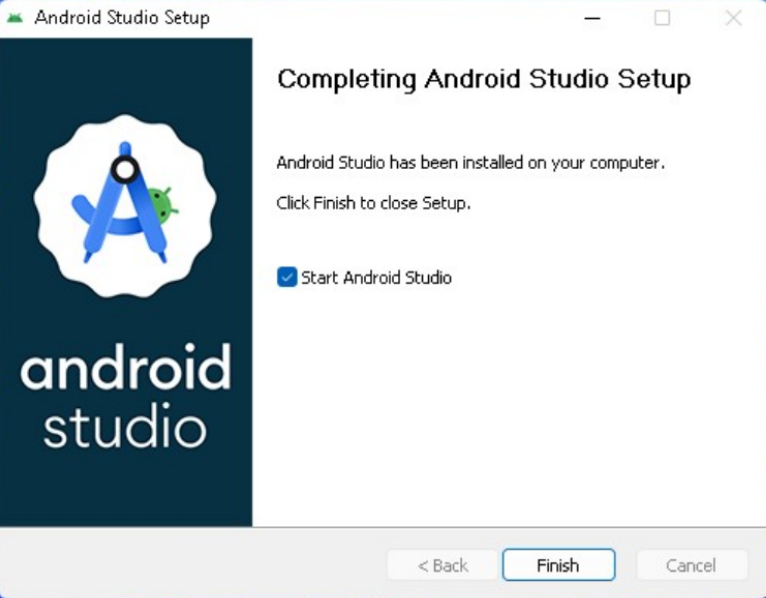
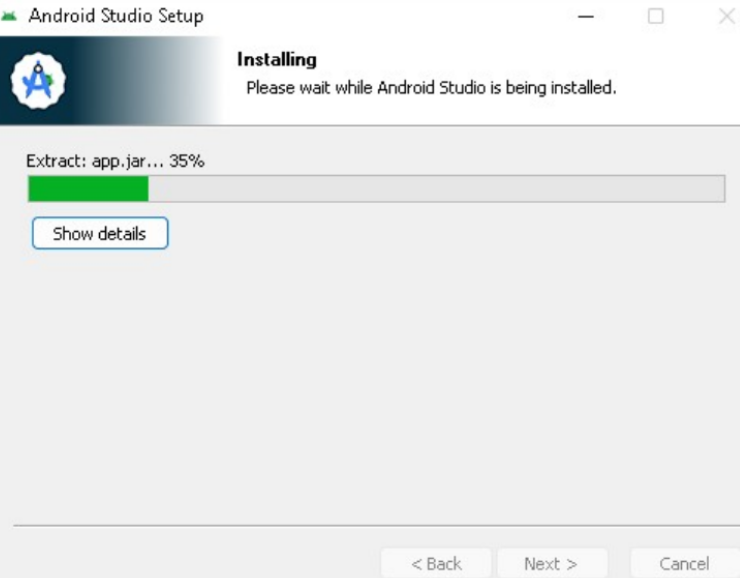
1. **Setup môi trường và tạo project**

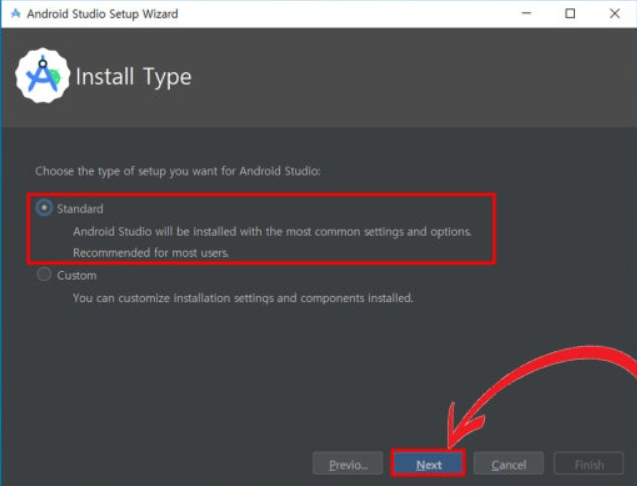
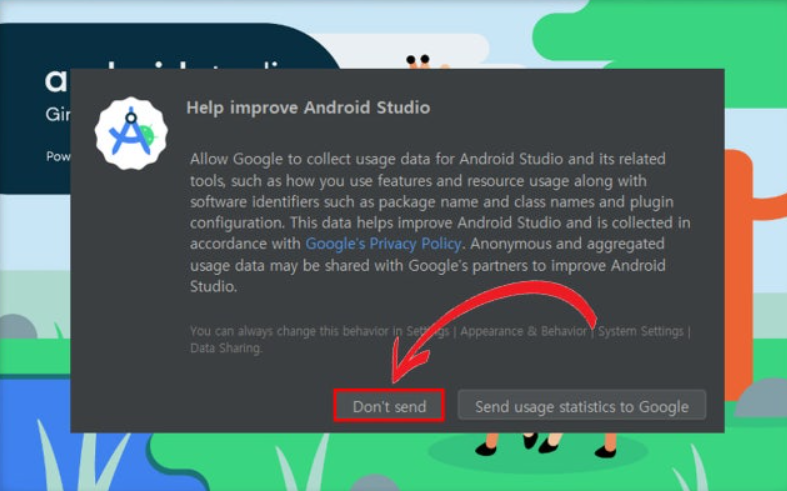
Link download: <https://developer.android.com/studio?hl=vi>

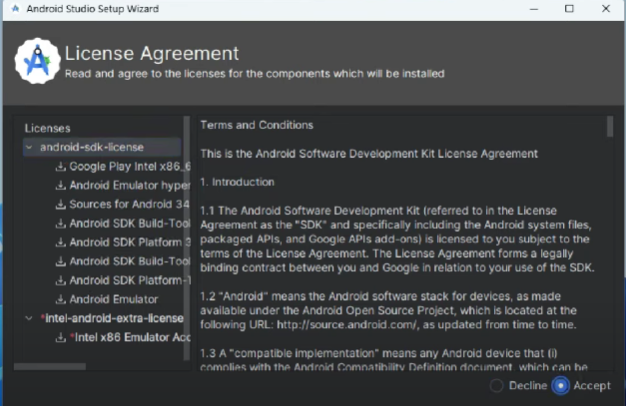
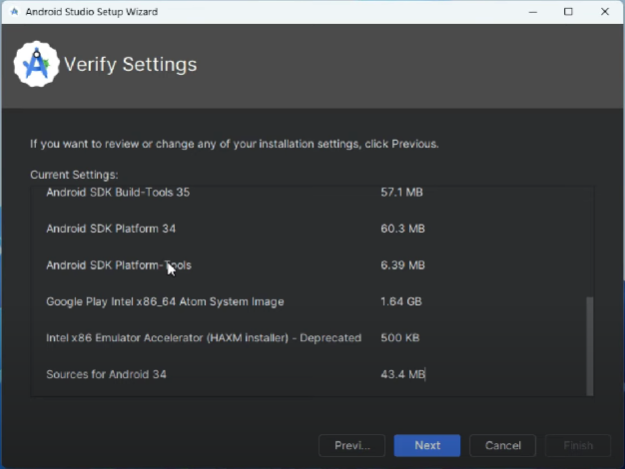
* Sau khi tải và install file zip , thì ứng dụng sẽ chạy :



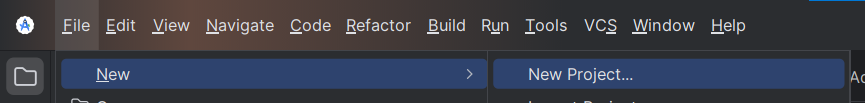


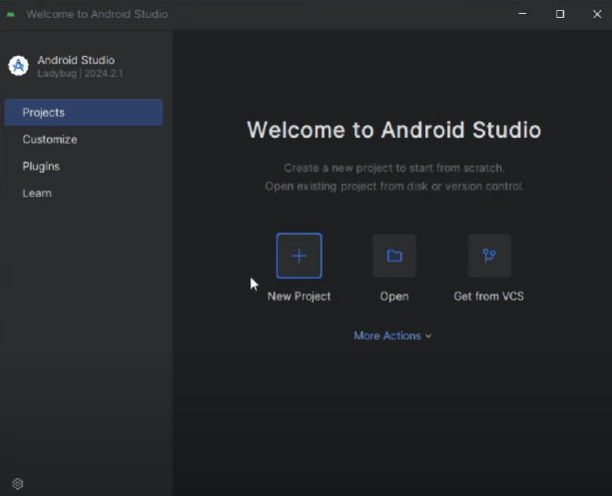




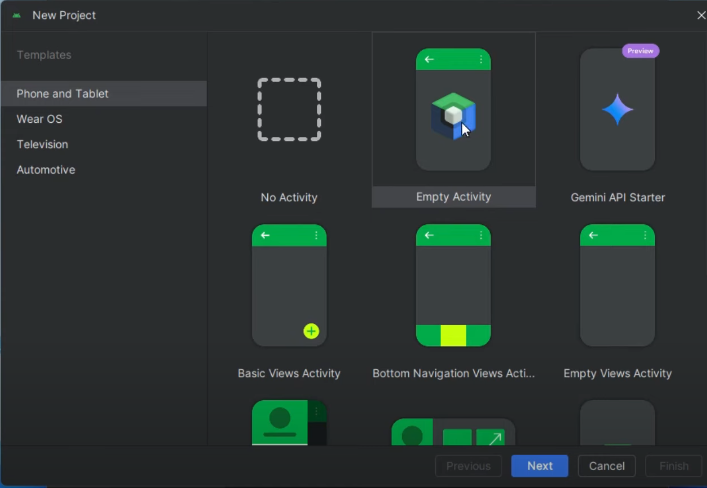


Tạo dự án mới : File → New → New Project . Còn nếu tải lần đầu thì chỉ cần bấm vào New Project trên màn hình



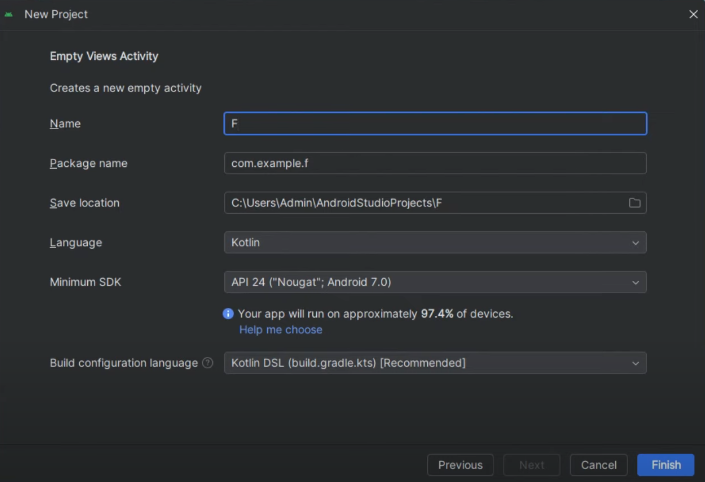


-Chọn loại dự án phù hợp:

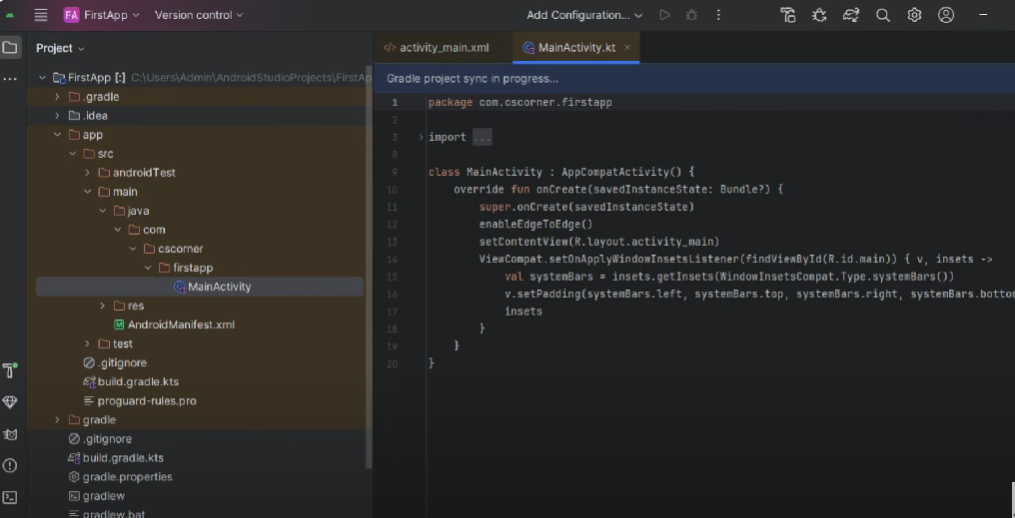


-Sau đó đặt tên và lưu dự án ở ổ đĩa bạn mong muốn, có thể chọn ngôn ngữ để code(Kotlin, java), và chọn Minimum SDK tương ứng với máy andoird của bạn

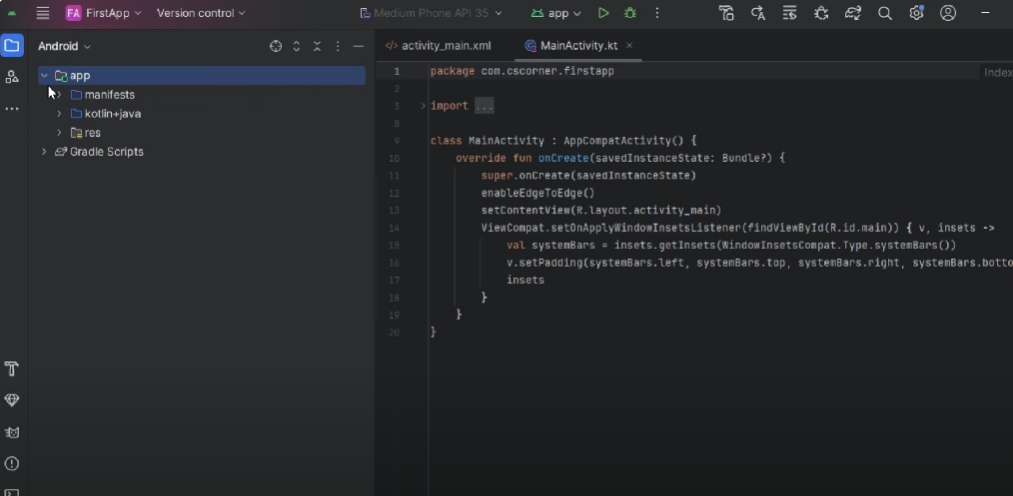
Điều này có nghĩa là ứng dụng chỉ có thể chạy trên các thiết bị có phiên bản hệ điều hành Android bằng hoặc cao hơn phiên bản được chỉ định trong trường **minSdkVersion** trong file cấu hình build.gradle của dự án Android.



-Sau khi hoàn thành xong nó sẽ tự động tạo project cho bạn, đầu tiên nó sẽ hiển thị ra với project như sau:



Và đợi để dự án build grade import các thư viện xong thì sẽ ra màn hình này:



2.Chi tiết cấu trúc của 1 project android

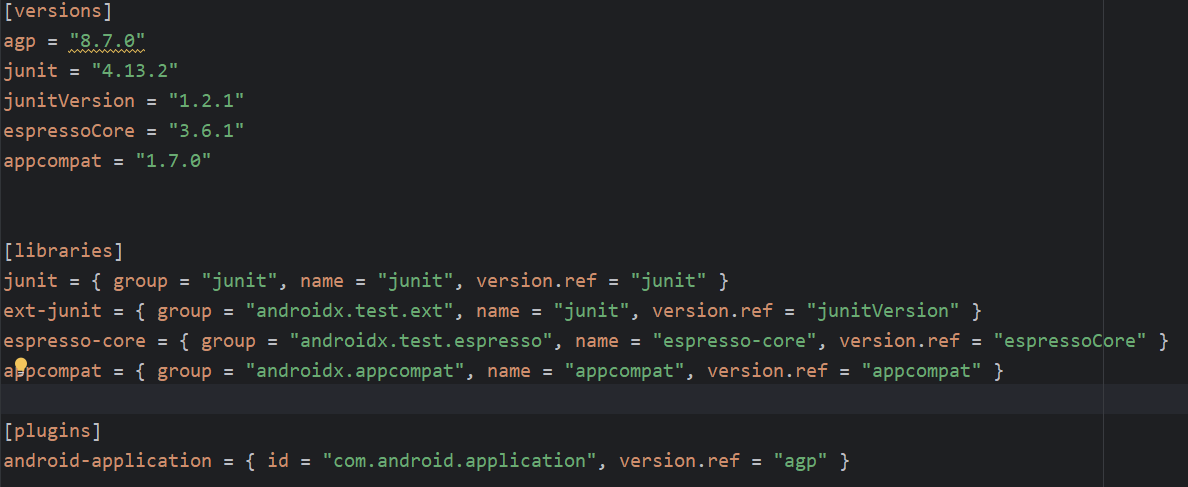
1. **Thư mục app/**:

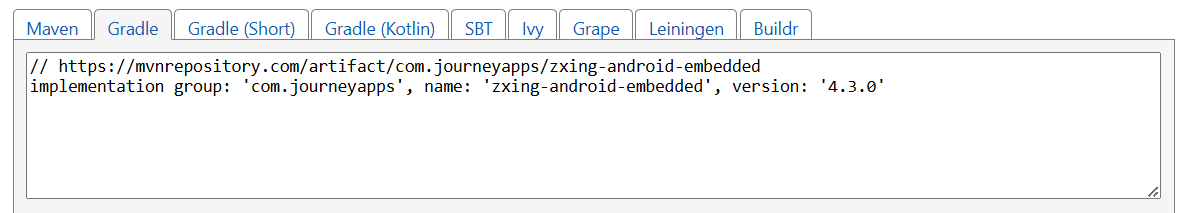
**src/**: Thư mục này chứa mã nguồn của ứng dụng, bao gồm các thư mục con:

* **main/**: Đây là thư mục chứa mã nguồn chính của ứng dụng.
  + **manifests / AndroidManifest.xml**: Đây là tệp cấu hình chính của ứng dụng Android. Nó định nghĩa các thành phần ứng dụng (Activities, Services, Receivers), quyền truy cập (permissions), và các thuộc tính khác của ứng dụng.
  + **java/**: Chứa mã nguồn Java hoặc Kotlin (tùy vào ngôn ngữ bạn chọn). Các lớp, hoạt động (Activities), dịch vụ (Services), trình phát (Fragments), và mã logic của ứng dụng đều được đặt ở đây.
    - **test/**: Chứa các bài kiểm tra đơn vị (unit tests) và các bài kiểm tra khác liên quan đến logic của ứng dụng.
    - **androidTest/**: Chứa các bài kiểm tra UI (instrumentation tests) để kiểm tra sự tương tác của người dùng với ứng dụng trên các thiết bị Android.
  + **res/**: Chứa tất cả các tài nguyên của ứng dụng như hình ảnh, layout (giao diện người dùng), màu sắc, và chuỗi văn bản.
    - **drawable/**: Chứa các tệp hình ảnh, icon.
    - **layout/**: Chứa các tệp XML dùng để định nghĩa giao diện người dùng (UI).
    - **values/**: Chứa các tệp XML định nghĩa các giá trị như chuỗi văn bản (strings.xml), màu sắc (colors.xml), kích thước (dimens.xml), kiểu chữ (styles.xml).
    - **mipmap/**: Thư mục chứa các biểu tượng ứng dụng ở các độ phân giải khác nhau. (Nó như là biểu tượng hình của app)

2. **Tệp build.gradle**:

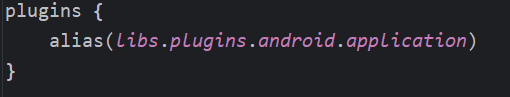
* + **libs.versions.toml:**
* ( tệp **libs.versions.toml** trong Android là một tệp cấu hình được sử dụng để quản lý và định nghĩa các phiên bản của các thư viện (dependencies) trong một dự án Android)
* **Cấu trúc của libs.versions.toml**: (Bạn có thể import thêm các phiên bản khác để phục vụ cho project của bạn cần). Bạn có thể tìm một số thư viện trong trang Maven repository





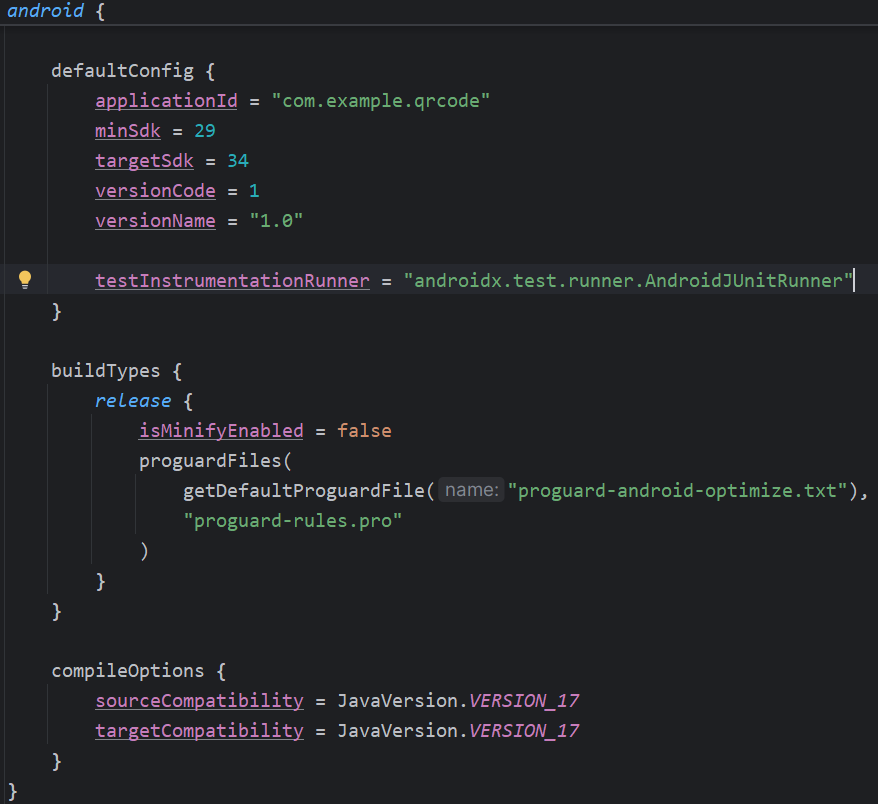
* + **build.gradle.kts (Module app) :**
* là tệp cấu hình chính cho module ứng dụng trong dự án Android

1. **Cấu hình Plugin**:



* **Plugin Android Application**: Tệp này khai báo plugin **android.application**, cho biết rằng module này là một ứng dụng Android, không phải thư viện hay một mô-đun khác.
* **alias(libs.plugins.android.application)**: Tham chiếu đến plugin từ tệp **libs.versions.toml** thay vì ghi trực tiếp phiên bản của plugin.

**2. Cấu hình android (Đối với java)**



**Chức năng**: Phần android cấu hình các thuộc tính quan trọng của ứng dụng Android, bao gồm:

* **namespace**: Định nghĩa không gian tên cho ứng dụng, tương đương với package name trong Java/Kotlin.
* **compileSdk**: Chỉ định phiên bản SDK mà ứng dụng sẽ biên dịch. Trong ví dụ này là **SDK 34**.
* **defaultConfig**: Cấu hình các thông tin mặc định của ứng dụng:
  + **applicationId**: ID ứng dụng (tên gói) duy nhất của ứng dụng.
  + **minSdk**: Phiên bản SDK tối thiểu mà ứng dụng hỗ trợ. Trong trường hợp này là **API 29 (Android 10)**.
  + **targetSdk**: Phiên bản SDK mà ứng dụng được tối ưu hóa để chạy. Ở đây là **SDK 34**.
  + **versionCode**: Mã phiên bản, giúp Google Play phân biệt các bản cập nhật.
  + **versionName**: Tên phiên bản ứng dụng mà người dùng sẽ thấy (ví dụ: "1.0").
  + **testInstrumentationRunner**: Trình chạy thử nghiệm (test runner) để thực hiện các bài kiểm tra tự động.
* **buildTypes**: Cấu hình các loại build, ví dụ như release:
  + **isMinifyEnabled**: Tùy chọn bật/tắt quá trình làm nhỏ mã nguồn (minify). Đây là cách giảm kích thước APK và bảo vệ mã nguồn.
  + **proguardFiles**: Định nghĩa các tệp cấu hình ProGuard để tối ưu hóa và bảo vệ mã nguồn khi đóng gói bản phát hành.
* **compileOptions**: Chỉ định các phiên bản Java được sử dụng để biên dịch mã nguồn. Trong trường hợp này là **Java 17**.

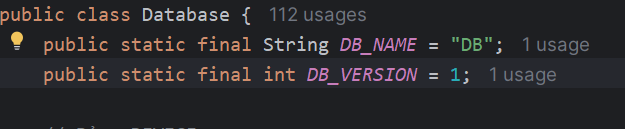
**3. Cấu hình dependencies**

****

* Đoạn mã này khai báo tất cả các thư viện và phụ thuộc mà dự án cần để xây dựng và chạy ứng dụng. Các thư viện này sẽ được tải và biên dịch vào trong ứng dụng Android.

1. **Nội dung chi tiết project**
2. **Database**

- SQLite là một cơ sở dữ liệu SQL mã nguồn mở, nó lưu trữ dữ liệu vào một tập tin văn bản trên một thiết bị. Nó mặc định đã được tích hợp trên thiết bị Android. Để truy cập dữ liệu này, bạn không cần phải thiết lập bất kỳ loại kết nối nào cho nó như JDBC, ODBC, …



* **DB\_NAME**: Tên của cơ sở dữ liệu.
* **DB\_VERSION**: Phiên bản của cơ sở dữ liệu. Khi thay đổi cấu trúc của cơ sở dữ liệu (ví dụ như thay đổi các bảng, cột, kiểu dữ liệu), bạn sẽ phải tăng giá trị này.

#### **Bảng DEVICE**

Các hằng số tiếp theo liên quan đến bảng DEVICE, nơi lưu trữ thông tin của các thiết bị. Các cột của bảng này bao gồm:

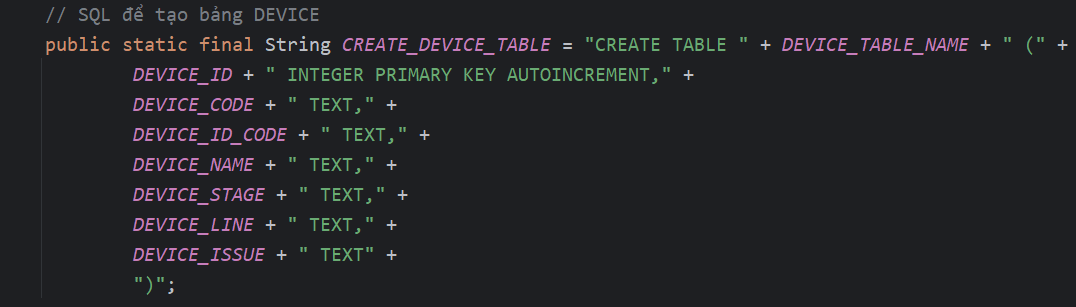
* **DEVICE\_ID**: ID của thiết bị (khóa chính).
* **DEVICE\_ID\_CODE, DEVICE\_CODE, DEVICE\_NAME, DEVICE\_STAGE, DEVICE\_LINE, DEVICE\_ISSUE**: Các cột chứa thông tin chi tiết về thiết bị.

#### **Bảng DEVICE\_HISTORY**

Các hằng số liên quan đến bảng DEVICE\_HISTORY, nơi lưu trữ lịch sử của các thiết bị:

* **HISTORY\_ID, HISTORY\_ID\_CODE, HISTORY\_CODE, HISTORY\_NAME, HISTORY\_STAGE, HISTORY\_LINE**: Các cột lưu thông tin về lịch sử thiết bị.
* **HISTORY\_ERROR\_COUNT, HISTORY\_STARTTIME, HISTORY\_ENDTIME, HISTORY\_TOTALTIME, HISTORY\_TYPE\_NAME**: Các cột lưu các thông tin về lỗi, thời gian, và loại thiết bị.

#### **Lệnh SQL để tạo bảng**

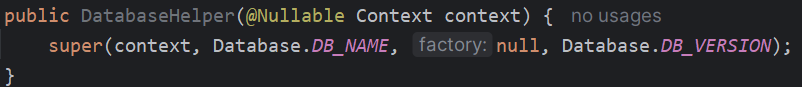


* **CREATE\_DEVICE\_TABLE**: Lệnh SQL để tạo bảng DEVICE.
* **CREATE\_DEVICE\_HISTORY\_TABLE**: Lệnh SQL để tạo bảng DEVICE\_HISTORY.

**b**. **Lớp DatabaseHelper**

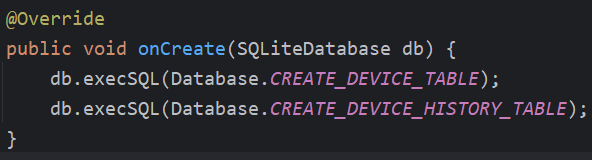
* Lớp này extend từ SQLiteOpenHelper và giúp quản lý csdl.

**DatabaseHelper(Context context)**:



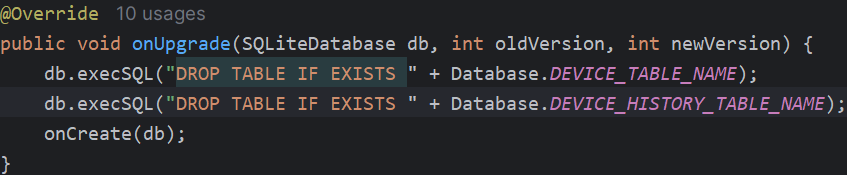
- Hàm tạo này gọi constructor của SQLiteOpenHelper để tạo cơ sở dữ liệu với tên và phiên bản đã định sẵn trong lớp Database.

#### **Hàm onCreate(SQLiteDatabase db)**



#### - Hàm này được gọi khi cơ sở dữ liệu lần đầu được tạo. Nó sẽ tạo các bảng DEVICE và DEVICE\_HISTORY bằng cách chạy các lệnh SQL đã được định nghĩa trong lớp Database.

#### **Hàm onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion)**



#### **-** Khi nâng cấp phiên bản của csdl (tăng DB\_VERSION), hàm này sẽ được gọi. Nó xóa các bảng cũ và tạo lại các bảng mới.

#### - Trong trường hợp này, hàm này sẽ xóa bảng DEVICE và DEVICE\_HISTORY nếu tồn tại, sau đó gọi lại onCreate() để tạo lại các bảng.

* Lưu ý:
* **onCreate() và onUpgrade()**: Là hai hàm quan trọng khi làm việc với csdl trong Android. onCreate() dùng để tạo cơ sở dữ liệu và các bảng lần đầu, còn onUpgrade() giúp xử lý các thay đổi giữa các phiên bản cơ sở dữ liệu.
* **SQLiteDatabase**: Là lớp đại diện cho cơ sở dữ liệu thực tế mà bạn sẽ thao tác trên đó, bao gồm các thao tác như chèn, sửa, xóa, truy vấn dữ liệu.
* **ContentValues**: Sử dụng để lưu trữ các giá trị dữ liệu theo dạng key-value trước khi chèn vào cơ sở dữ liệu.
* **Cursor**: Khi truy vấn cơ sở dữ liệu, kết quả trả về là một đối tượng Cursor, giúp duyệt qua từng dòng dữ liệu.
* Cursor bao gồm một số hàm để điều hướng kết quả truy vấn: moveToFirst : di chuyển con trỏ đến hàng đầu tiên trong kết quả truy vấn.
* moveToNext : di chuyển con trỏ đến hàng tiếp theo.
* **Mở cơ sở dữ liệu**: Sử dụng getWritableDatabase() hoặc getReadableDatabase() của SQLiteOpenHelper.
* **Chèn dữ liệu**: Sử dụng insert() của SQLiteDatabase để chèn dữ liệu vào các bảng.
* **Cập nhật dữ liệu**: Sử dụng update() để cập nhật các bản ghi trong bảng.
* **Xóa dữ liệu**: Dùng delete() để xóa các bản ghi.
* **Truy vấn dữ liệu**: Sử dụng query() để truy vấn các bản ghi và trả về một Cursor để duyệt qua kết quả.