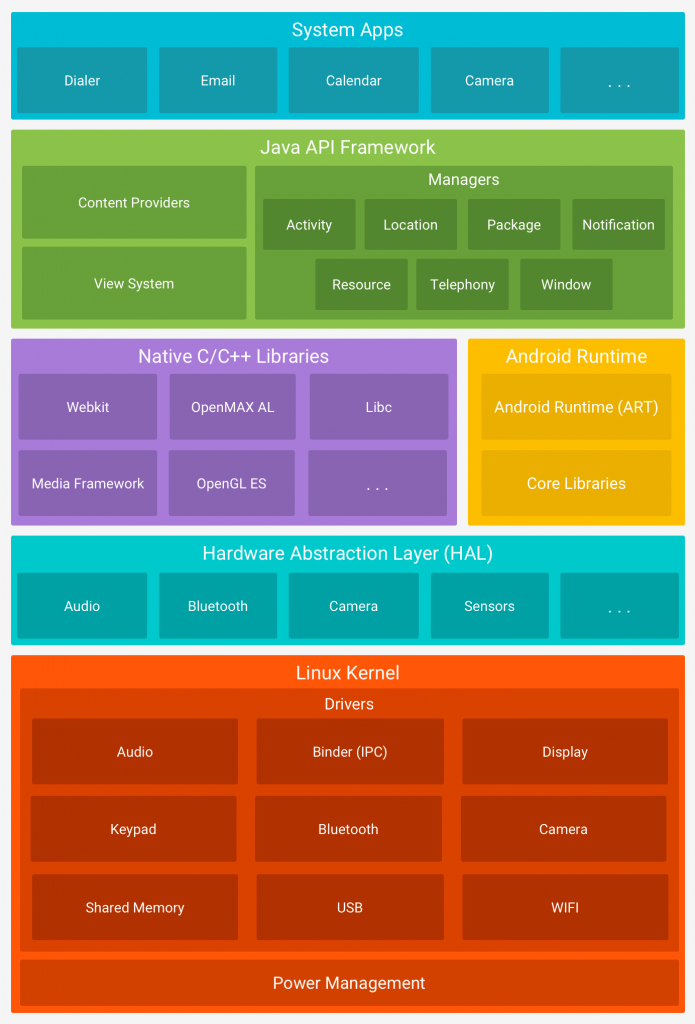
Android là một hệ điều hành mã nguồn mở dựa trên nền tảng Linux gồm có 6 tầng là Linux Kernel, Hardware Abstraction Layer - HAL, Android Runtime, Native C/C++ Libraries, Java API Framework, System Apps.



**A. Android project structure**

**1. manifests: AndroidManifest.xml**

Ứng dụng của bạn khai báo tất cả thành phần trong tệp này, đây là thư mục gốc của thư mục dự án ứng dụng.

- Xác định mọi quyền của người dùng mà ứng dụng yêu cầu, chẳng hạn như quyền truy cập Internet hoặc quyền đọc danh bạ của người dùng.

- Khai báo mức tối thiểu Cấp độ API mà ứng dụng yêu cầu, dựa trên API mà ứng dụng dùng.

- Khai báo các tính năng phần cứng và phần mềm mà ứng dụng sử dụng hoặc yêu cầu, chẳng hạn như camera, Dịch vụ Bluetooth hoặc màn hình cảm ứng đa điểm.

- Khai báo các thư viện API mà ứng dụng cần được liên kết.

**2. java: Code back-end, có thể là Kotlin**

- Main activity: khi chạy ứng dụng thì main activity sẽ được gọi.

**3. res: Chứa các tài nguyên ứng dụng, chẳng hạn như tệp XML và hình ảnh icon, layout, values tham chiếu, v.v...**

- drawable: chứa Bitmap files (PNG, .9.png, JPG, or GIF)

- layout: Các tệp XML xác định bố cục giao diện người dùng.

- mipmap: Các icon được sử dụng cho các độ phân giải khác nhau.

- values: Tệp XML chứa các giá trị đơn giản như chuỗi, số nguyên và màu sắc để tham chiếu đến.

- xml: Các tệp XML tùy ý có thể được đọc khi chạy bằng cách gọi Resources.getXML().

- ngoài ra còn các thư mục như animator, anim, color, menu, raw, font

3. Gradle Scripts:

- Hai tệp build.gradle

+ Mức project: config cho tất cả project

+ Mức module: config cho các module trong project. Mỗi module có một tệp build.gradle

- produard-rules.pro: Xác định các quy tắc để kiểm soát việc thu nhỏ, tối ưu hóa và làm tối nghĩa ứng dụng.

***B. View - ViewGroups***

- Tất cả các thành phần trong android đều được gọi là view.

- xml là công cụ hỗ trợ để tạo ra view, định hình cấu trúc của giao diện. java mới là ngôn ngữ lập trình vẽ ra giao diện cho chúng ta nhìn thấy

- **Density – independent pixels (dp):** dùng để định kích thước của view

- **Scale-independent Pixels (sp):** dùng để định kích thước cho text

**I. Layout**

1. LinearLayout

- orientation: thiết lập chiều sắp xếp của các view dọc (hoặc ngang)

- weight: tỷ trọng của các view chiếm bao nhiêu %

- gravity: căn trái, phải, trên, dưới, giữa

2. Relavtie Layout

- alignParent: căn theo lề

- below: nằm dưới 1 object nào đó theo id

- above: nằm trên 1 object nào đó theo id

- margin: cách so với bên ngoài 1 đơn vị (dp)

- padding: cách so với bên trong 1 đơn vị (dp)

- centerHorizontal: giữa màn hình theo chiều ngang

- centerVertical: giữa màn hình theo chiều dọc

- centerInParent = centerHorizontal + centerVertical: giữa màn hình

3. Frame Layout

- measureAllChildren: hiển thị tất cả thành phần

4. Table Layout

- bên trong là <TableRow></TableRow>

- layout\_span : gộp thêm n cột

- stretchColumns="\*": giãn đều các cột

5. ConstraintLayout