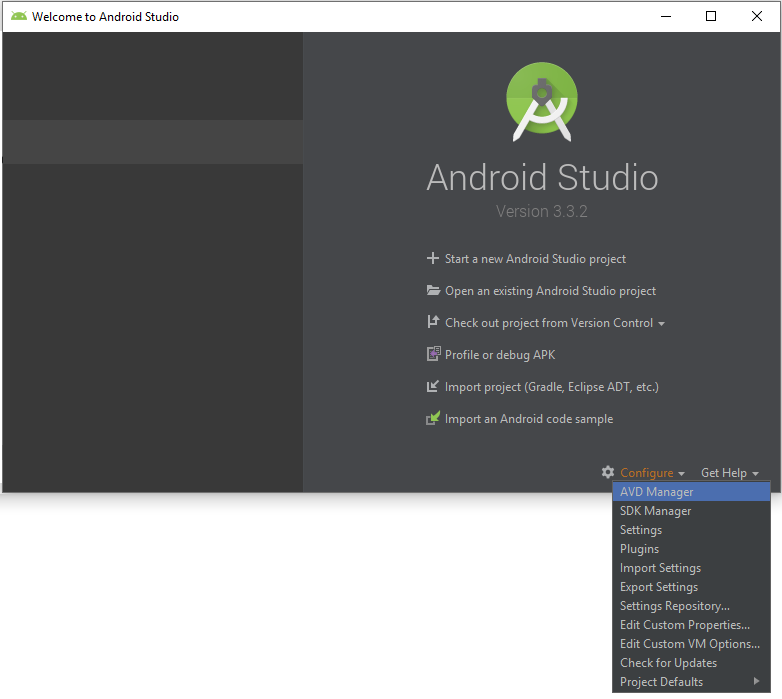
1. **Cài đặt môi trường tích hợp để lập trình Android**
2. **Hoạt động team**

* **Đặt tên cho team : Done (HSTC)**
* **Tạo tài khoản skype : Done ( Xem tại file HSTC\_Report\_Task1.xls)**
* **Tạo group add giảng viên : Done**

1. **Cài đặt tool**

* **JDK 1.8 : Done**
* **Android Studio : Done**
* **SublimeText 3 : Done**
* **Gennymotion : Done**
* **Git : Done**
* **Tạo tài khoản bitbucket : Done**

1. **Tìm hiểu Android Studio**



**Android SDK** (Software Development Kit) – hay còn gọi tắt là devkit, là tập hợp các công cụ phát triển phần mềm. SDK cho phép tạo ra các ứng dụng cho một gói phần mềm, framework, nền tảng phần cứng, hệ thống máy tính, thiết bị video game, hệ điều hành, hay nền tảng phát triển nào đó tương tự.

Install các version Android gồm: 5.1 -> 9.0 : Done

Install SupportRepository : Done

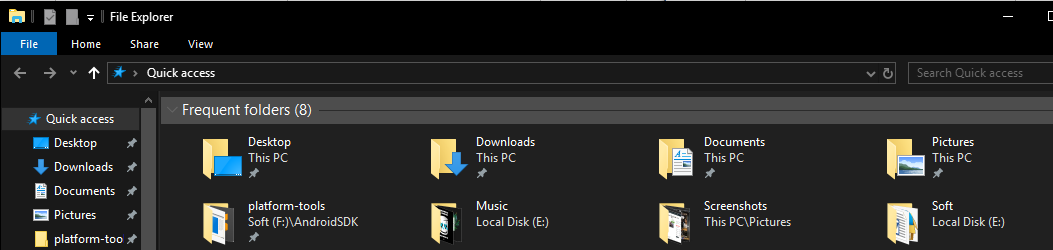
Install Android Emulator : Done

Install Android SDK Platform-Tools: Done

Install Android SDK Tools : Done

Install Android SDK Build-Tools: Done

Ghim folder platform-tools vào lối tắt : Done



**AVD** (Android Virtual Devices) : AVD có thể xem như là một thiết bị ảo mà Google cung cấp cho lập trình viên để thử nghiệm chương trình trên HĐH Android. Tùy thuộc vào loại thiết bị, kích cỡ màn hình, độ phân giải, … mà chúng ta có thể có các AVD khác nhau

Tạo AVD : Done

1. **Tạo project Android theo yêu cầu : Done**
2. **Cấu hình lại Android Studio : Done**

**LogCat** : Tương tự vs Terminal của các IDE khác như Eclipse, Inteliji…dùng để hiển thị các dòng thông tin của hệ thống

1. **Tìm hiểu các gói trong android studio :**

**Thư mục java :** Lưu các file .java

**Thư mục build:** Chứa thư mục trong đó lưu trữ tệp tin .apk. Lưu trữ module chính của ứng dụng.

**Thư mục libs:** Chứa các thư việc chỉ được dùng trong ứng dụng, lưu trữ module chính của ứng dụng.

**Thư mục res (resource) :** Chứa các thư mục **drawable, layout, mipmap, values**

**drawable**: Đây chính là thư mục chứa các file hình ảnh, config xml… trong dự án android.

**layout**: Đây chính là thư mục lưu các file xml về giao diện của các màn hình ứng dụng.

**mipmap**: Đây là thư mục thường chứa ảnh logo ứng dụng, các file hình ảnh sẽ được chứa trong thư mục drawable nhưng ngoại lệ thằng ảnh logo thì chúng ta chứa trong thư mục **mipmap** này cho chuẩn.

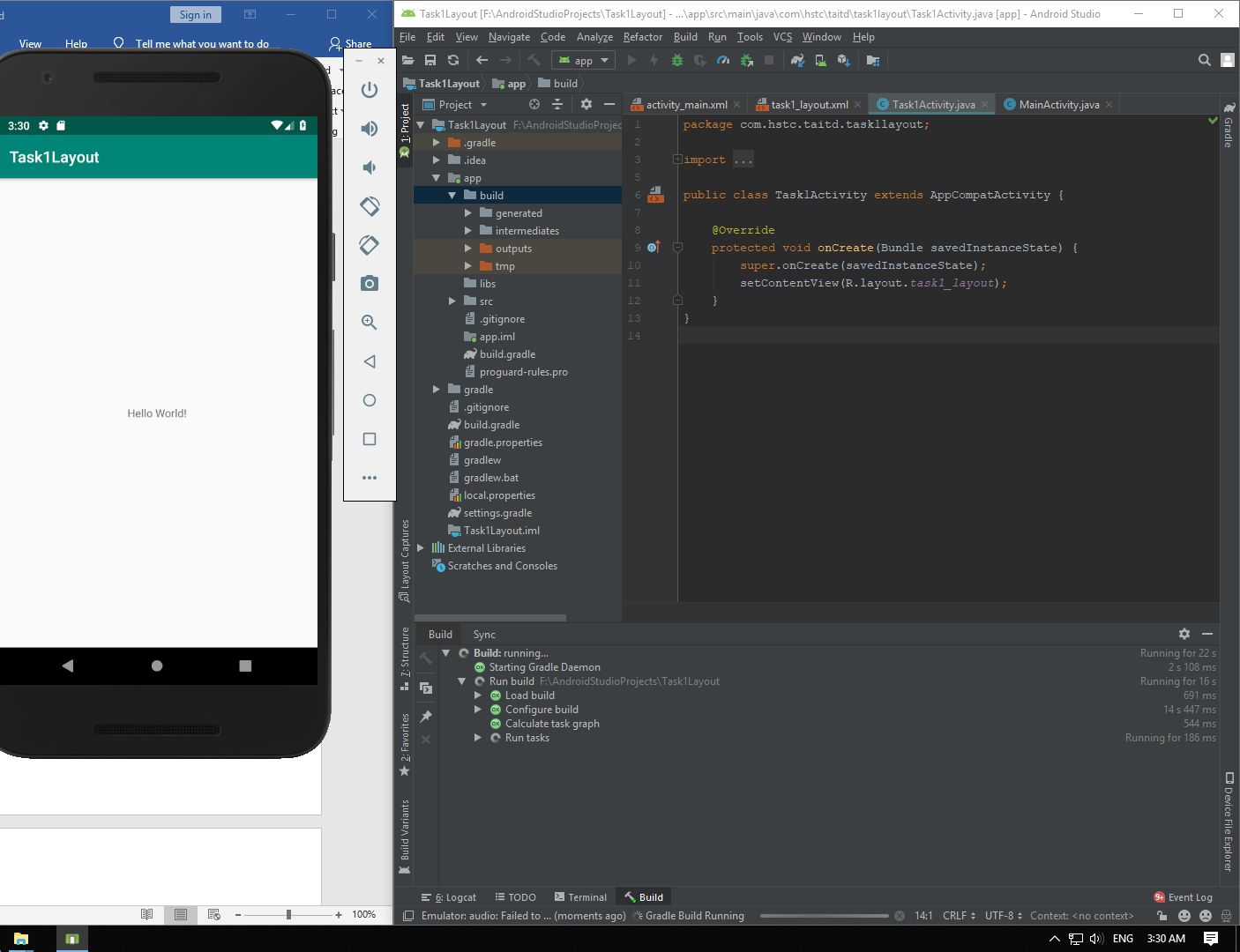
**values**: Định nghĩa các giá trị hằng trong dự án để sử dụng đồng nhất trong dự án

* **color.xml**: đây là file định nghĩa các mã màu trong dự án android, khi sử dụng màu nào bạn chỉ cần gọi tên mã màu đã được định nghĩa trong file này ra.
* **dimens.xml**: đây là file mà bạn sẽ định nghĩa ra các kích thước như cỡ chữ, chiều cao, chiều rộng các view.
* **strings.xml**: đây là file định nghĩa các đoạn văn bản trong ứng dụng Android
* **styles.xml**: đây chính là nơi định nghĩa các giao diện của các file layout trong thư mục layout.

Ví dụ: Muốn chỉnh một nút Button chiều cao 10dp, chiều rộng 10dp, màu xanh… và cần sử dụng button này ở 5 màn hình khác nhau. Chúng ta không thể mỗi màn hình lại định nghĩa lại như thế sẽ làm duplicate code và sẽ không tối ưu.Thay vào đó chỉ cần định nghĩa một file giao diện như trên và ở mỗi màn hình sẽ gọi đến nó là xong

**build.gradle :**  file cấu hình các thuộc tính cho dự án Android như phiên bản Android SDK, version ứng dụng, cấu hình thư viện nào cần dùng trong Project.

1. **Chạy thử project lên máy ảo : Done**



1. **Tìm hiểu về Layout trong Android**

Là thành phần định nghĩa cấu trúc giao diện người dùng hay nói cách khác là thành phần quyết định đến giao diện của một màn hình trong ứng dụng Android.

Layout gồm nhiều ViewGroup, View…. kết hợp với nhau để tạo nên một giao diện hoàn chỉnh. Một file layour được lưu dưới dạng tệp tin xml và lưu trong thư mục **layout.**

Khi nhìn vào một giao diện ứng dụng chúng ta nhìn thấy các button, hình ảnh, văn bản, textbox…… thì đó chính là các View, View là thành phần chính để tạo nên một file Layout, bên cạnh đó còn có ViewGroup.

##### **ViewGroup**

Được kết thừa từ lớp **android.view.ViewGroup**, ViewGroup chính là view cha, có thể bao bọc các View/ViewGroup khác ở bên trong.Mục đích của ViewGroup thì định dạng những thuộc tính mà các view chứa trong nó phải dựa vào đó mà tuỳ chỉnh theo và sẽ chịu ảnh hưởng theo.

### **LinearLayout**

Đây là ViewGroup sẽ giúp các bạn sắp xếp các view con chứa bên trong theo dạng hàng ngang hoặc hàng dọc với nhau.

### **RelativeLayout**

Đây là một ViewGroup giúp bạn có thể sắp xếp các view con ở bất kì một vị trí nào mà mình thích mà có những lúc LinearLayout làm không được hoặc rất khó để làm

Thông thường ,khi nhìn vào một giao diện, khi thấy các view xếp theo chiều dọc hoặc chiều ngang thì thường là sẽ được tạo ra từ LinearLayout, còn các view xếp lộn xộn không theo thứ tự ví dụ như nằm ở giữa, góc trái, nằm dưới cùng, trên cùng…. Thì dùng RelativeLayout sẽ dễ dàng hơn.

##### **View**

Nó  kế thừa từ lớp **android.view.View**, 1 View có thể nằm trong các ViewGroup tuy nhiên không thể chứa được các View khác giống như ViewGroup.Giao diện của ứng dụng được xây dựng chủ yếu bằng các View này.

Những View cơ bản mà thường hay sử dụng trong dự án đó là:

* Button
* ListView
* TextView
* EditText
* ImageView
* ImageButton
* CheckBox
* RadioButton
* Spinner