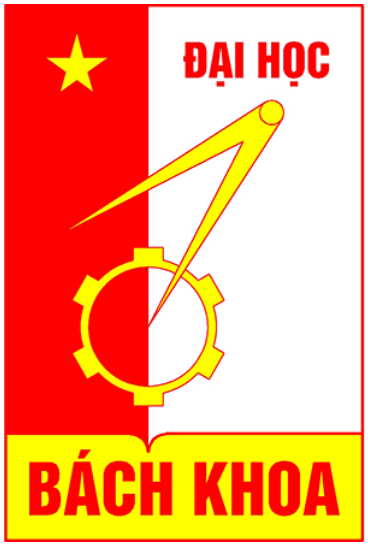
**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN TOÁN ỨNG DỤNG VÀ TIN HỌC**

——————o0o——————




# LẬP TRÌNH ANDROID

## ĐỒ ÁN II

**Chuyên ngành: TOÁN TIN**

**Chuyên sâu: Lập trình Android với React Native**

Giảng viên hướng dẫn : **TS. Nguyễn Huy Trường**

Nhóm sinh viên thực hiện : **Nguyễn Bùi Nam Trường**

MSSV : **20185418**

**HÀ NỘI, tháng ../2022**

## NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

**1. Mục đích và nội dung của đồ án:**

-

-

-

**2. Kết quả đạt được**

-

-

-

**3. Ý thức làm việc của sinh viên:**

-

-

-

*Hà Nội, ngày .. tháng .. năm 2022*

Giảng viên

**TS. Nguyễn Huy Trường**

# Lời nói đầu

Hà Nội, ngày .. tháng .. năm 2022

Sinh viên thực hiện

Nguyễn Bùi Nam Trường

# Mục lục

# 1 Giới thiệu về Android

Như chúng ta biết, hiện tại đã có hơn nửa nhân loại sử dụng máy di động để thoại và giao tiếp qua các mạng không dây. Con số 3 tỉ người này sẽ còn tăng lên và máy di động càng ngày càng "thông minh" với nhiều chức năng và dịch vụ rất hấp dẫn, cho nên thị trường máy di động thông minh sẽ vượt xa máy vi tính trong một tương lai rất gần... Vì thế việc lập trình trên thiết bị di động ngày càng phổ biến và phát triển rất mạnh mẽ. Từ nền tảng mã nguồn mở, Google đã cho ra mắt Android chạy trên các thiết bị di động. Android có rất nhiều công cụ và dụng cụ miễn phí để nghiên cứu và phát triển phần mềm trên nền tảng của nó. Trong đồ án này, ta sẽ tìm hiểu về Android và cách viết một ứng dụng trên nền tảng này.

## 1.1 Khái niệm về Android

Trước hết, Hệ điều hành (Operating System - OS) là phần mềm hệ thống quản lý phần cứng máy tính, phần mềm và cung cấp các dịch vụ chung cho các chương trình máy tính. Android là hệ điều hành dựa trên mã nguồn mở Linux OS cho các thiết bị di động và các thiết bị thông minh như máy tính bảng, laptop, netbook, smartbook, TV thông minh(Google TV),... Hiện nay, một số hệ điều hành cho thiết bị di động: Android, OS X (iPhone), Windows Mobile, Symbian,... Tuy nhiên, hệ điều hành được sử dụng nhiều nhất là Android và OS X.

Ngay từ khi ra mắt, Android đã gây ấn tượng mạnh khi đây là đứa con của Goodle sử dụng giấy phép mã nguồn mở. Android là sản phẩm kết tinh từ ý tưởng của Khối Liên minh thiết bị cầm tay mở do Google dẫn đầu, gồm 34 thành viên với các công ty hàng đầu về công nghệ và di động toàn cầu như Qualcomm, Intel, Motorola, Texas Instruments và LG Electronics, các nhà mạng như T-Mobile, Sprint Nextel, NTT DoCoMo và China Mobile.

Với đặc tính "mở", Android cho phép các nhà phát triển có thể sử dụng miễn phí bộ Kit Android Software Development để xây dựng các ứng dụng của mình. Ví dụ, một ứng dụng có thể gọi bất kì chức năng lõi của điện thoại như thực hiện gọi, gửi tin nhắn, sử dụng máy ảnh,... Điều này thực sự thu hút các nhà phát triển sáng tạo ra các phần mềm hấp dẫn, từ đó tạo ra một cộng đồng phát triển lớn - là nguyên nhân đến sự phổ biến của Android như hiện nay.

## 1.2 Kiến trúc của Android

Để bắt đầu lập trình với Android, chúng ta cần nghiên cứu qua bản thân HĐH Android, chúng ta không cần hiểu quá chi tiết, nhưng trước hết là cần có cái nhìn chung và toàn diện nhất về Android.
Android Platform bao gồm đầy đủ các tính năng HĐH Android, các ứng dụng và các tâng trung gian để nhà phá triển có thể mở rộng, tùy chỉnh thêm theo nhu cầu của họ. Có 4 tầng cơ bản trong HĐH Android: Linux Kernel, Native Libraries, Android Runtime, Application Framework,... Các tầng làm việc có sự liên kết với nhau.

1. *Tầng Linux Kernel:* Đây là tầng nhân của HĐH Android, Linux Kernel giúp hệ điều hành có thể giao tiếp với phần cứng của thiết bị như: camera, USB, Wifi, Bluetooth, Display, Power Management,... Linux Kernel chịu trách nhiệm cho các trình điều khiển thiết bị, quản lý nguồn điện, quản lý bộ nhớ, quản lý thiết bị và truy cập tài nguyên. Kernel hoạt động như một lớp trừu tượng giữa phần cứng và phần mềm còn lại của hệ thống.

2. *Native Libraries:* Native Libraries là tập hợp của nhiều thư viện như WebKit, OpenGL, FreeType, SQLite, Media, SSL,...

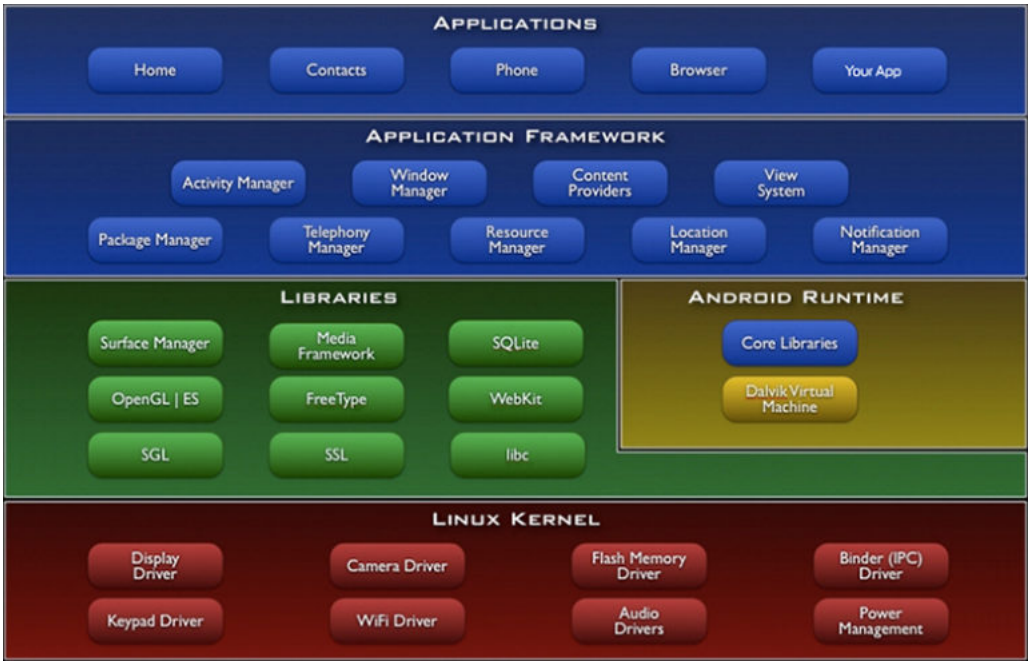
3. *Tầng Android Runtime:* Android Runtime cung cấp một thành phần quan trọng được gọi là DVM (Dalvik Virtual Machine - máy ảo) có trách nhiệm chạy ứng

dụng android. DVM thực thi các file có định dạng .dex (Dalvik Excutable), định dạng này là định dạng được tối ưu hóa về bộ nhớ.

DVM có phần tương tự như JVM (Java Virtual Machine) nhưng được tối ưu hóa cho các thiết bị di động như tiêu thụ ít bộ nhớ hơn và tăng hiệu suất hoạt động tốt hơn.

4. *Tầng Application Framework:* Application Framework bao gồm tập hợp những API cho phép các nhà phát triển ứng dụng được phép sử dụng các dịch vụ này trong các ứng dụng của họ.

Ở góc độ người dùng, ta có thêm tầng Application - là các ứng dụng do nhà phát triển ứng dụng viết. Dưới đây là sơ đồ tương quan liên kết giữa các tầng:




Hình 1: Sơ đồ tương quan liên kết giữa các tầng

# 2 Giới thiệu về React Native

## 2.1 Giới thiệu

React Native là một Framework, do công ty Công nghệ Meta (trước đây là Facebook) phát triển, nhằm giải quyết vấn đề về hiệu năng và việc phải sử dụng nhiều ngôn ngữ native trên nền tảng di động.

React Native cho phép xây dựng và phát triển ứng dụng native đa nền tảng một cách dễ dàng, khác với HTML5 App, Mobile Web App, Hybrid App. Mục đích khi tạo ra React Native là khắc phục các điểm yếu của ứng dụng web, giúp cho nhà lập trình tiết kiệm thời gian, công sức bởi sự hỗ trợ đắc lực từ JavaScript.

React Native là một trong những framework sử dụng cấu hình thiết kế tương tự như React. Có thể nói, biết về React là biết được 80% về React Native. Tuy nhiên, đồ án này sẽ trình bày đủ các thông tin cả về React.

## 2.2 Lịch sử phát triển

Năm 2012 Mark Zuckerberg đã phát biểu: "Sai lầm lớn nhất của chúng tôi khi làm công ty là dựa trên quá nhiều HTML hơn là môi trường phát triển gốc". Ông hứa rằng Facebook sẽ sớm cung cấp trải nghiệm di động tốt hơn.

Kỹ sư Jordan Walke tại Facebook đã tìm ra cách xây dựng các thành phần UI cho iOS bằng một luồng JavaScript. Họ quyết định tổ chức cuộc thi Hackathon để hoàn thiện nguyên mẫu hệ thống để có thể xây dựng các ứng dụng di động gốc (native app) bằng công nghệ này.

Sau nhiều tháng phát triển, Facebook đã phát hành phiên bản đầu tiên cho React Native vào năm 2015. Trong một cuộc hội thảo công nghệ, Christopher Chedeau cho biết Facebook đã sử dụng React Native trong phát triển ứng dụng nhóm và ứng dụng

quản lí quảng cáo của họ.

Với cộng đồng phát triển ứng dụng lớn, ngày nay React Native trở thành framework được ưa chuông trong việc xây dựng ứng dụng native.

## 2.3 Lý do React Native được ưa chuộng

Để biết tại sao React Native lại được ưa chuộng, ta đề cập đến các ứng dụng phổ biến trước đây là Hybrid Apps. Hybrid App được hiểu là ứng dụng được xây dựng dựa trên các công nghệ web phổ biến là CSS, Javascript, HTML. Như vậy, ứng dụng được xây dựng lớn, cần phát triển lâu dài thì sẽ không đảm bảo được hiệu năng.

Trong khi đó, React Native lại có những ưu điểm sau:

1. Tiết kiệm thời gian học: Việc học từng loại ngôn ngữ cho từng nền tảng thường rất khó và mất nhiều thời gian. Tuy nhiên với React Native, lập trình viên chỉ cần học duy nhất một bộ công cụ.

2. Tái sử dụng code: Trong lập trình phần mềm, React Native là công cụ tái sử dụng code hiệu quả nhất mang lại các lợi thế như duy trì ít code, tận dụng tốt nguồn nhân lực,...

3. Hot reloading: Khi phát triển ứng dụng, nhà phát triển không tốn quá nhiều thời gian để tổng hợp app mỗi khi có sự thay đổi mà chỉ cần làm mới app trong thiết bị hoặc giả lập

Chính vì lý do trên mà hiện nay, React Native đang dần trở thành lựa chọn số một cho công việc xây dựng app của hầu hết các công ty lớn. Cũng từ việc được ưa chuộng khiến cho cộng đồng phát triển lớn, từ đó tạo thành vòng lặp khiến các nhà phát triển mới tiếp cận đều chọn React Native.

## 2.4 Nguyên lý hoạt động

Về cơ bản, React Native hoạt động bằng cách tích hợp cho ứng dụng đi động 2 thread là JS thread và Main thread.

1. *Main thread*: giữ vai trò cập nhật giao diện người dùng UI và xử lý các tương tác của người dùng ngay sau đó.

2. *JS thread*: thực thi và xử lý các code JavaScript.

Hai thread này hoạt động độc lập với nhau và giao tiếp qua một cầu nối trung gian.

## 2.5 Một số ứng dụng sử dụng react native

Với việc được ưa chuộng và có cộng đồng phát triển lớn, thế giới càng ngày càng có nhiều ứng dụng sử dụng React Native ra đời. Một số ứng dụng có thể kế đến như sau:

1. *Facebook:* là công ty phát triển React Native, sau khi phát triển xong framework này, Meta đã chuyển đổi tính năng Event Dashboard cho iOS sang React Native để kiểm tra hiệu suất ứng dụng, từ đó cắt giảm thời gian tìm hiểu thị trường đi một nửa

2. *Facebook Ads:* Đến thời điểm hiện tại, tất cả ứng dụng quảng cáo trên Facebook đều được sử dụng React Native.

3. *Instagram:* Sau khi được Meta mua lại, ứng dụng này cũng được chuyển đổi sử dụng React Native, cụ thể chế độ Push Notifications đã được triển khai dưới dạng WebView và không yêu cầu xây dựng cơ sở hạ tầng Navigation vì UI khá đơn giản.

# 3 Xây dựng ứng dụng Android với React Native

## 3.1 Thiết lập môi trường

Để bắt đầu xây dựng một ứng React Native, ta cần cài đặt môi trường lập trình trên thiết bị của mình. Cụ thể cần: NodeJS, JDK, Android Studio.

### 3.1.1 NodeJS

*a. Tổng quan*

NodeJS là một nền tảng (flatform) cung cấp môi trường runtime chạy JavaScript, được sử dụng để chạy các ứng dụng web bên ngoài client. Nền tảng này được phát triển bởi Ryan Dahl vào năm 2009, được xem là một giải pháp hoàn hảo cho các ứng dụng sử dụng nhiều dữ liệu nhờ vào mô hình hướng sự kiện (event-driven) không đồng bộ. NodeJS cung cấp các package để xây dựng ứng dụng React Native.

*b. Cài đặt*

Link cài đặt: `https://nodejs.org/en/download/`

Có thể cài bằng các dòng lệnh thông qua terminal (phụ thuộc vào hệ điều hành của thiết bị):

- Windows: choco install -y nodejs-lts
- MacOS: brew install node | brew install watchman
- Linux: apk add nodejs npm

Sau khi cài đặt xong, có thể kiểm tra lại bằng cách chạy dòng lệnh trong terminal:


Hình 2: Cài đặt NodeJs

### 3.1.2 JDK

*a. Tổng quan*

JDK (Java Development Kit) là một trong ba gói công nghệ cốt lõi được sử dụng trong lập trình Java: JVM (Máy ảo Java - Java Virtual Machine), JRE (Java Runtime Enviroment - Môi trường Java Runtime), JDK.

Có thể định nghĩa JDK theo 2 cách sau:

- *Định nghĩa chuyển ngành*: JDK là một hệ tiêu chuẩn trong việc triển khai nền tảng Java, bao gồm các trình thông dịch và thư viện lớp.
- *Định nghĩa thông thường*: JDK là gói phần mềm bạn tải xuống để tạo các ứng dụng dựa trên Java.

Như đã trình bày ở phần kiến trúc Android, tầng Android Runtime cần trình biên dịch Java để compile ra các lớp.

*b. Cài đặt*

Link cài đặt: `https://www.oracle.com/eg/java/technologies/downloads/`

Có thể cài bằng các dòng lệnh thông qua terminal (phụ thuộc vào hệ điều hành của thiết bị):

- Windows: choco install -y microsoft-openjdk11
- MacOS: brew install java

- Linux: sudo apt install default-jdk

Sau khi cài đặt xong, có thể kiểm tra lại bằng cách chạy dòng lệnh trong terminal:

```
C:\Users\admin>java --version
openjdk 11.0.17 2022-10-18
OpenJDK Runtime Environment OpenLogic-OpenJDK (build 11.0.17+8-adhoc..jdk11u)
OpenJDK 64-Bit Server VM OpenLogic-OpenJDK (build 11.0.17+8-adhoc..jdk11u, mixed mode)
```

Hình 3: Cài đặt JDK

### 3.1.3 Android Studio

*a. Tổng quan*

Android Studio là Môi trường phát triển tích hợp (IDE) chính thức để phát triển ứng dụng Android.

Nhờ các công cụ cho nhà phát triển tích hợp trình soạn thảo mạnh mẽ của IntelliJ IDEA, Android Studio cung cấp các tính năng giúp hỗ trợ và nâng cao năng suất trong quá trình phát triển ứng dụng Android:

- Hệ thống xây dựng linh hoạt dựa trên Gradle.
- Trình mô phỏng nhanh và có nhiều tính năng.
- Môi trường hợp nhất để phát triển các thiết bị Android.
- Tự cập nhật thay đổi ứng dụng mà đang chạy mà không cần khởi động lại.
- Hỗ trợ tự động chạy code template và code snippet.
- Tích hợp GitHub để tạo môi trường làm việc hiệu quả.

*b. Cài đặt*

Link cài đặt: `https://developer.android.com/studio`

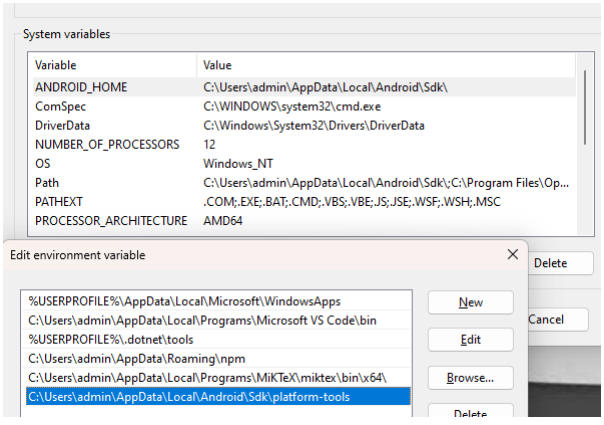
Trong quá trình cài đặt Android Studio cần đảm bảo có các mục được chọn:

- Android SDK.
- Android SDK Platform.
- Android Virtual Device.

Sau khi cài đặt Android Studio, cần cấu hình Android SDK phù hợp để chạy React Native (theo yêu cầu được cung cấp trong tài liệu). Cụ thể, trong Android Studio, chọn "Appearance &Behavior" -> "System setting" -> "Android SDK". Khi đó cần cài các gói:

- SDK Flatform: Android SDK Platform 33, Intel x86 Atom_64 System Image.
- SDK Tools: Android SDK Build-Tools 33.0.0.

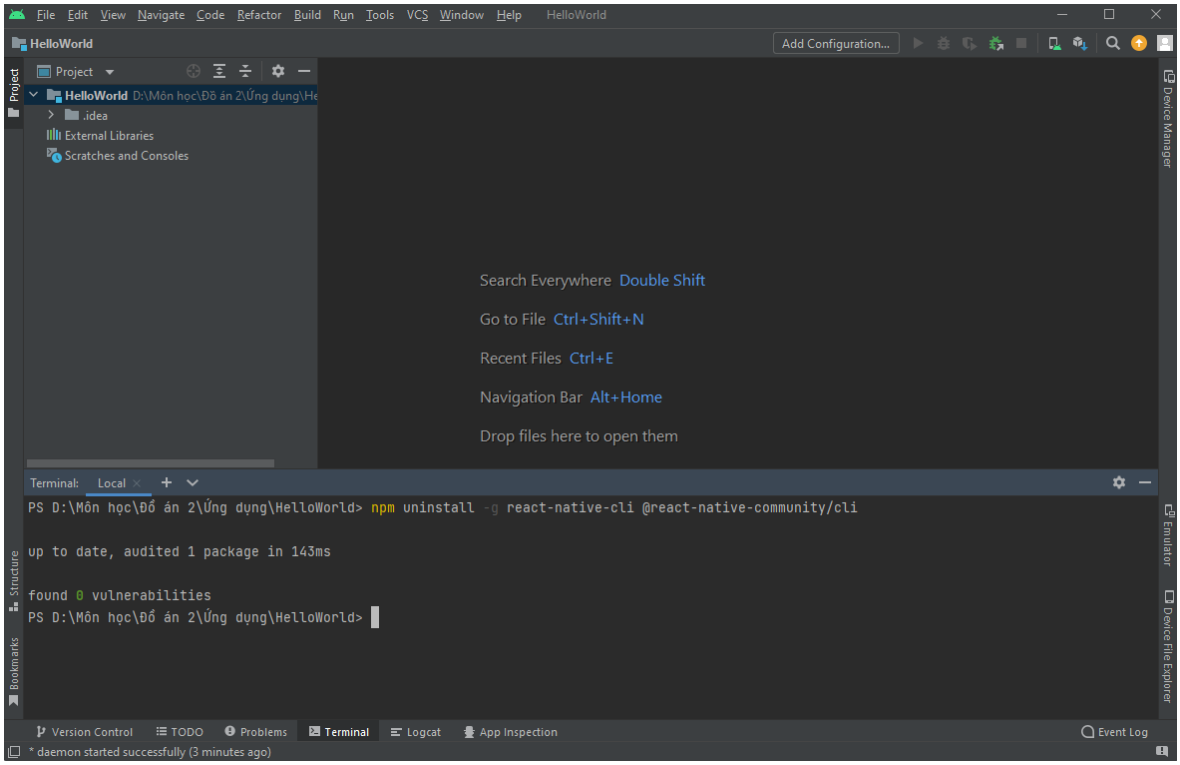
Một yêu cầu khác, các công cụ React Native yêu cầu biến môi trường được thiết lập để code. Cụ thể, cần cấu hình biến ANDROID_HOME cho đường dẫn SDK Android và thêm đường dẫn SDK Android Flatform vừa cài đặt trên (cụ thể phụ thuộc vào hệ điều hành của thiết bị).




Hình 4: Cài đặt biến môi trường ANDROID_HOME

## 3.2 Xây dựng chương trình đầu tiên

Nếu trước đó thiết bị đã cài đặt gói react-native-cli toàn cầu, chúng ta cần xóa gói này vì gói đó có thể gây ra các sự cố không mong muốn:

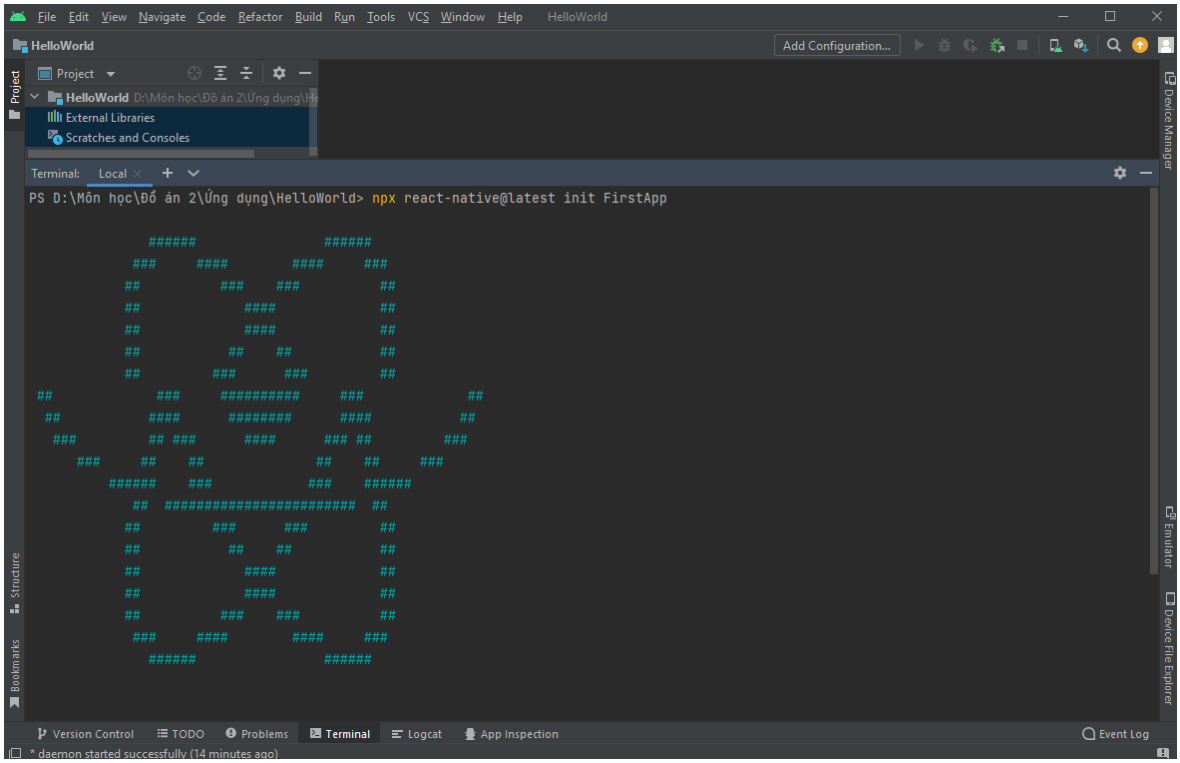



Hình 5: Xoá gói React Native CLI

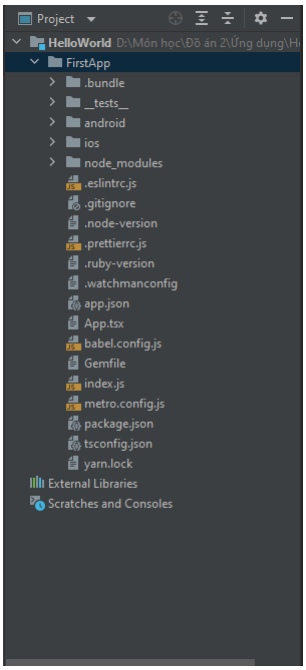
Để bắt đầu xây dựng chương trình, ta tạo thư mục cho project này. Cụ thể, tôi đặt "HelloWorld". Sau đó mở thư mục đó bằng Android Studio, chạy dòng lệnh sau trên terminal:

Cụ thể, "npx" là công cụ CLI do NodeJs cung cấp để cài đặt và quản lý các thành phần phụ thuộc npm. Ta sử dụng gói react-native để khởi tạo ứng dụng FirstApp. Sau khi chạy dòng lệnh trên, quan sát project, ta thấy thư mục FirstApp và các file trong đó:

Công việc tiếp theo là chạy ứng dụng FirstApp đã khởi tạo trên. Có 2 cách để build và chạy ứng dụng react native trong quá trình xây dựng:




Hình 6: Khởi tạo ứng dụng React Native




Hình 7: Dữ liệu FirstApp

- Chạy trên thiết bị vật lý
- Chạy trên máy ảo

### 3.2.1 Chạy trên thiết bị vật lý

*a. Bật gỡ lỗi trên thiết bị vậy lý*

Để bật gỡ lỗi trên thiết bị Android của bạn, ta cần thực hiện các bước:

- Bật chế độ nhà phát triển
- Bật gỡ lỗi USB

Cụ thể cách bật chế độ nhà phát triển sẽ khác nhay tùy thuộc vào thiết bị, có thể tìm kiếm cách bật trên trình duyệt. Tuy nhiên để bật chế độ này có thể tổng quát: **Cài đặt** → **Giới thiệu về điện thoại** → **Thông tin phần mềm** rồi nhấn Build number bảy lần vào hàng ở dưới cùng; sau đó, quay lại **Cài đặt** → **Tùy chọn nhà phát triển** để bật "Gỡ lỗi USB".

*b. Kết nối với thiết bị*

Tiếp theo cần kết nối thiết bị Android với thiết bị phát triển ứng dụng qua USB. Sau đó, kiểm tra lại thiết bị đã bật gỡ lỗi USB chưa bằng cách chạy dòng lệnh "adb devices".

```
PS D:\Môn học\Đồ án 2\Ứng dụng\HelloWorld> adb devices
List of devices attached
VNBKGMF1900028  device
```

Hình 8: Kiểm tra kết nối thiết bị

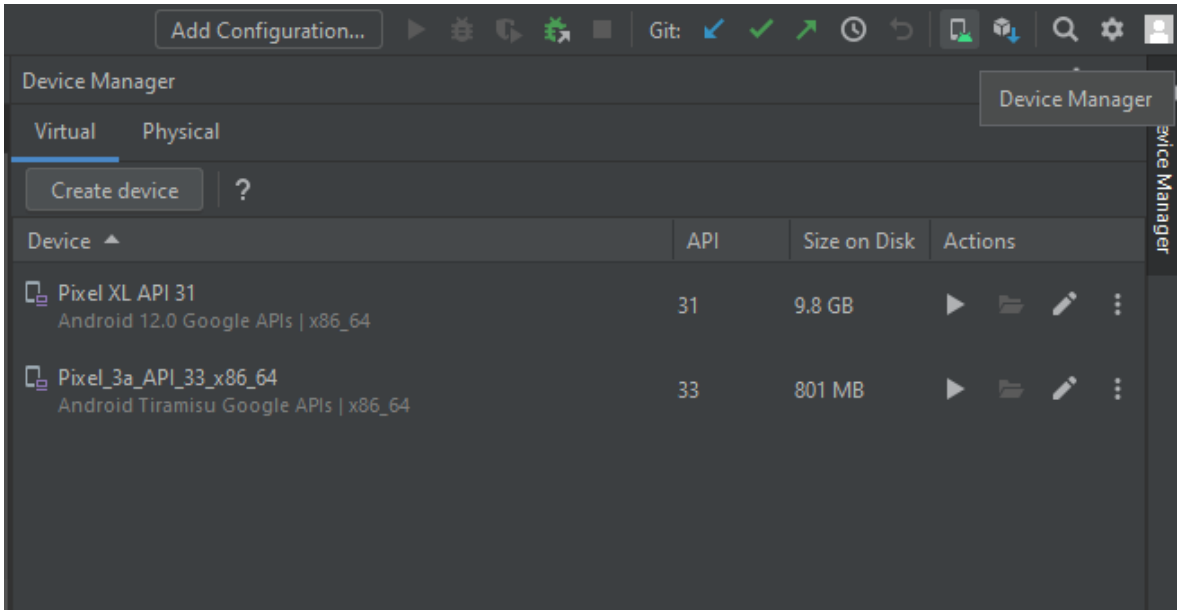
*c. Chạy ứng dụng trên thiết bị*

Để chạy ứng dụng, ta di chuyển đến FirstApp và chạy dòng lệnh sau: npx react-native run-android.

### 3.2.2 Chạy trên máy ảo

*a. Tạo AVD mới*

Trước hết ta cần tạo AVD mới bằng Android Studio. Ta vào mục **Device Manager** → **Create device** Sau đó, Android Studio hiển thị giao diện cấu hình thiết




Hình 9: Tạo thiết bị ảo trên Android Studio

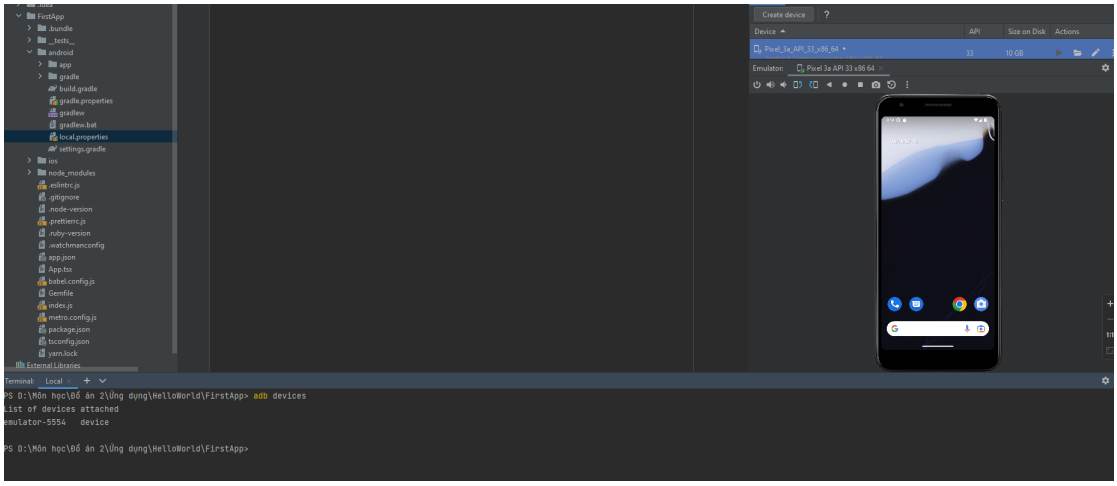
bị, ta chọn bất kỳ Điện thoại nào từ danh sách và nhấp vào **Next**, sau đó chọn hình ảnh **Tiramisu** API Cấp 33. Tiếp theo chọn **Finish** để tạo thiết bị ảo.

*b. Kiểm tra kết nối*

Sau khi khởi tạo, ta mở thiết bị ảo trên Android Studio và kiểm tra (giống chạy trên thiết bị vật lý):

*c. Chạy ứng dụng trên AVD*

Thực hiện chạy ứng dụng trên thiết bị ảo giống trên thiết bị vật lý.

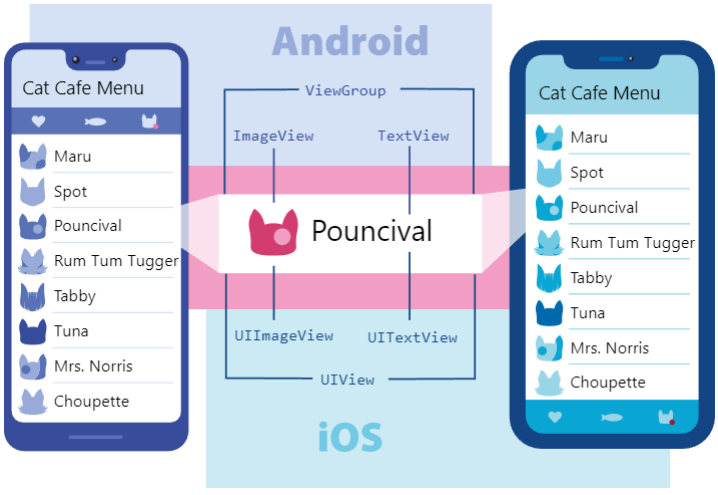


Hình 10: Kiểm tra kết nối thiết bị ảo trên Android Studio

## 3.3 Các khái niệm, thành phần

### 3.3.1 Components

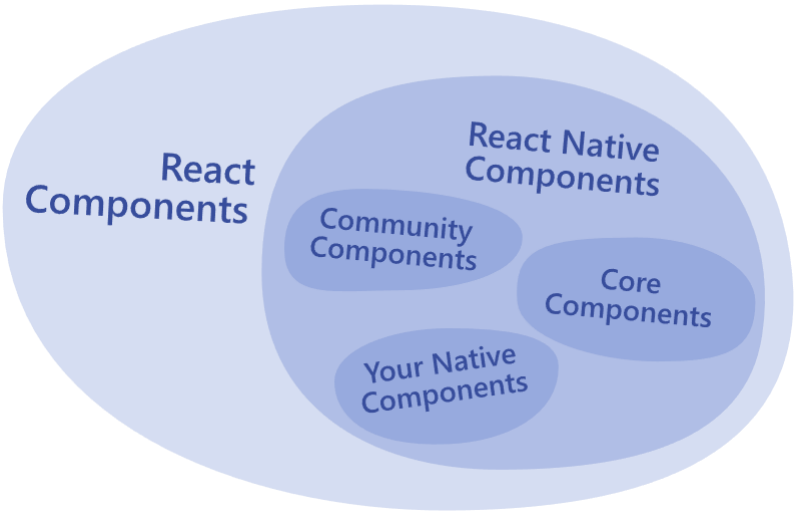
Trong quá trình phát triển Android, người ta thường phân giao diện thành các khối hình chữ nhật để thực hiện các chức năng riêng như: hiển thị văn bản, hình ảnh, phản hồi của người dùng,... Mỗi một khối như vậy được gọi là View. Đối với React Native,




Hình 11: View trong giao diện Native

các View trên được gọi là *Component*. Một cách dễ hiểu, để xây dựng ứng dụng React

Native, ta cần tập chung mô tả giao diện mong muốn thành các component như trên. Ta có thể phân loại các components như hình vẽ sau (chứa cả React Component vì React Native được xây dựng dựa trên ReactJS)




Hình 12: Phân loại các component

*a. Core Components*

Core Components là các component cốt lõi để xây dựng ra các component khác. Các component này điều khiển các hành động cơ bản về giao diện như: hiển thị văn bản, hình ảnh, cho phép người dùng nhập, ấn,...

Có thể phân loại Core Component như sau:

- *Basic Components*: bao gồm các component trong bảng 13.
- *User Interface*: Gồm các component điều khiển các hành động của người dùng:
  + Button: Xử lý thao tác chạm của người dùng.
  + Switch: Xử lý bật/tắt cho một chức năng.
- *List View*: Không giống như ScrolView, các component này hiển thị nội

| REACTNATIVE COMPONENT | ANDROID VIEW | IOS VIEW | WEB VIEW TƯƠNG ỨNG | MÔ TẢ |
|---|---|---|---|---|
| `<View>` | `<ViewGroup>` | `<UIView>` | Không cuộn `<div>` | Vùng chứa hỗ trợ bố cục với flexbox, kiểu, một số xử lý cảm ứng và điều khiển trợ năng |
| `<Text>` | `<TextView>` | `<UITextView>` | `<p>` | Hiển thị, tạo kiểu và lồng các chuỗi văn bản và thậm chí xử lý các sự kiện chạm |
| `<Image>` | `<ImageView>` | `<UIImageView>` | `<img>` | Hiển thị các loại hình ảnh khác nhau |
| `<ScrollView>` | `<ScrollView>` | `<UIScrollView>` | `<div>` | Một vùng chứa cuộn chung có thể chứa nhiều thành phần và dạng xem |
| `<TextInput>` | `<EditText>` | `<UITextField>` | `<input type="text">` | Cho phép người dùng nhập văn bản |

Hình 13: Mô tả một số Core Components

dung dưới dạng các phần tử. Điều này làm cho chúng trở thành lựa chọn hiệu quả để hiển thị danh sách dữ liệu dài. Bao gồm:

+ FlatList: Hiển thị một danh sách có thể cuộn.
+ SectionList: Giống FlatList nhưng đối với danh sách được phân đoạn.

- *Android-specific*: là các component cung cấp các dịch vụ đặc trưng của Android. Bao gồm:
  + BackHandler: Phát hiện các nút phần cứng điều hướng quay lại. (Ví dụ: nút back ảo, vuốt cạnh bên,...)
  + DrawerLayoutAndroid: Tạo một giao diện dạng ngăn kéo trên Android.
  + PermissionsAndroid: Cung cấp quyền truy cập đối với Android
  + ToastAndroid: Tạo một cảnh báo trên Android
- *iOS-specific*: là các component cung cấp các dịch vụ đặc trưng của UIKit. (Đồ án này chỉ tìm hiểu về react native và android nên không đi sâu vào IOS, chỉ nêu ra ở đây)

- *Other*: các component thường dùng khác
  + ActivityIndicator: Hiển thi giao diện loading xoay tròn
  + Alert: Hiển thị hộp thoại thông báo với tiêu đề và nội dung
  + Animated: Một thư viện để tạo các hoạt ảnh linh hoạt, mạnh mẽ, dễ xây dựng và bảo trì.
  + Dimensions: Cung cấp giao diện để nhận kích thước thiết bị.
  + KeyboardAvoidingView: Cung cấp giao diện điều khiển hiển thị bàn phím.
  + Linking: Xử lý hành động tương tác với liên kết.
  + Modal: Một giao diện nổi phía trên giao diện gốc.
  + PixelRatio: Cung cấp quyền truy cập vào mật độ pixel của thiết bị.
  + RefreshControl: Thành phần này được sử dụng bên trong ScrollView chức năng thêm kéo để làm mới.
  + StatusBar: Thành phần để kiểm soát thanh trạng thái ứng dụng.

Các core component được cung cấp trong package "react-native", vì thế để sử dụng chúng, ta chỉ cần import từ package đó:

*b. Community Components*

Như đã trình bày, cộng đồng phát triển React Native rất lớn, vì thế để phục vụ cho phát triển ứng dụng nhanh chóng, một số nhóm, công ty xây dựng ra các gói, thư viện chứa các component xây dựng sẵn như: ant-design, axios, calendar, react-native-webview,...Cụ thể các thư viện này được trình bày tại mục "Một số thư viện phổ biến".

Để sử dụng được các component này, ta cần cài đặt các thư viện bằng dòng lệnh: npm install <package-name> Sau khi cài đặt gói thành công, các component này sử dụng như core component, chỉ cần import và sử dụng.

```
2   import { View, Text } from 'react-native';
3
4   const  MyComponent = () => {
5     return (
6       <View>
7         <Text>my component</Text>
8       </View>
9     )
10  }
11
12  export default MyComponent;
```

Hình 14: Ví dụ sử dụng Core Component

```
PS D:\Môn học\Đồ án 2\Ứng dụng\HelloWorld\FirstApp> npm install @ant-design/react-native --save
npm WARN deprecated core-js@2.6.12: core-js@<3.23.3 is no longer maintained and not recommended for usage due to the number of issues. Because of the V8 engine whims, feature detection in old core-js versions could cause a slowdown
up to 100x even if nothing is polyfilled. Some versions have web compatibility issues. Please, upgrade your dependencies to the actual version of core-js.

added 39 packages, changed 50 packages, and audited 1003 packages in 20s

121 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

Hình 15: Cài đặt package - AntDesign

c. *Your Native Components*

Đây là các component do chính người phát triển ứng dụng xây dựng.

Việc phân chia, xây dựng các component một các chính xác giúp cho ứng dụng được tối ưu, tái sử dụng code một cách hiệu quả

### 3.3.2 JSX

### 3.3.3 Props

### 3.3.4 State

## 3.4 Một số thư viện phổ biến

## 3.5 Phát hành ứng dụng