**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**



**ĐỒ ÁN CHUYÊN NGÀNH**

**TÌM HIỂU REACT NATIVE VÀ**

**XÂY DỰNG ỨNG DỤNG PHIMNOW**

**Mã lớp :** SE346.J11

**Giảng viên hướng dẫn :** Huỳnh Tuấn Anh

**Sinh viên thực hiện :** 14520683 Nguyễn Hoài Phú

13521007 Phan Văn Tùng

**LỜI CẢM ƠN**

Để hoàn thành được đồ án này, trước tiên, nhóm chúng em xin gửi lời cảm ơn chân thành đến toàn thể giảng viên Trường Đại học Công nghệ thông tin – ĐHQG TP.HCM nói chung, các thầy cô trong Khoa Công nghệ Phần mềm nói riêng đã tận tâm truyền đạt kiến thức, kinh nghiệm, chỉ dạy tận tình cho nhóm chúng em từ những ngày đầu bước chân vào trường, vào chuyên ngành.

Đặc biệt, lời cảm ơn sâu sắc nhóm chúng em xin gửi tới Thầy Huỳnh Tuấn Anh, người đã đồng hành cùng nhóm chúng em trong suốt học kỳ qua. Được sự hướng dẫn trực tiếp của cô cùng với sự tận tâm chỉ bảo, luôn theo sát và hỗ trợ hết lòng, gắn bó trong từng chặng đường nhỏ trong khoảng thời gian này để giúp cho nhóm chúng em hoàn thành tốt đồ án lần này.

Cuối cùng, nhóm chúng em cũng gửi lời cảm ơn tới các anh chị, bạn bè trong Khoa Công nghệ Phần mềm đã nhiệt tình hướng dẫn, góp ý về những thiếu sót cũng như chia sẻ những kinh nghiệm, tài liệu trong quá trình làm đồ án này cũng như trong suốt thời gian em học tập tại trường.

Mặc dù nhóm chúng em đã cố gắng, nỗ lực rất nhiều để học hỏi, tìm hiểu, nghiên cứu và thu thập được nhiều thông tin, dữ liệu cùng với sự giúp đỡ của mọi người, nhưng do kiến thức còn nhiều hạn chế nên đồ án này không tránh khỏi những thiếu sót. Hơn ai hết, nhóm chúng em mong nhận được sự góp ý từ thầy cô để đồ án của nhóm chúng em được hoàn thiện hơn.

Xin chân thành cảm ơn !

**MỤC LỤC**

[1. Giới thiệu 4](#_Toc516051525)

[1.1. Đặt vấn đề 4](#_Toc516051526)

[1.2. Đối tượng người dùng 5](#_Toc516051527)

[2. Về đồ án 5](#_Toc516051528)

[2.1. Nền tảng React Native 5](#_Toc516051529)

[2.2. Công cụ lập trình 11](#_Toc516051530)

[2.3. Cài đặt React Native 12](#_Toc516051531)

[3. Một số Components trong React Native 16](#_Toc516051532)

[4. Mô hình nghiệp vụ 26](#_Toc516051533)

[4.1. Sơ đồ phân rã chức năng 26](#_Toc516051534)

[4.2. Sơ đồ Usecase 27](#_Toc516051535)

[5. Cơ sở dữ liệu 28](#_Toc516051536)

[5.1. API PhimNOW 28](#_Toc516051537)

[6. Thư viện sử dụng 30](#_Toc516051539)

[7. Giao diện thiết kế 38](#_Toc516051540)

[7.1. Danh sách màn hình 38](#_Toc516051541)

[7.2. Chi tiết thiết kế 38](#_Toc516051542)

[8. Tài liệu tham khảo 49](#_Toc516051543)

[9. Phân công nhiệm vụ 51](#_Toc516051544)

[10. Tổng kết – Đánh giá 52](#_Toc516051545)

**1. Giới thiệu**

**1.1. Đặt vấn đề**

Với xu hướng công nghệ hiện nay thì ngành công nghệ thông tin đã có những bước phát triển vượt bậc, điển hình là ở Việt Nam. Dân số Việt Nam hơn 90 triệu người thì có đến 39,8 triệu người sử dụng Internet (chiếm 44%), và tỉ lệ người sử dụng smartphone ngày càng gia tăng nhanh chóng, chiếm 84% so với người dùng điện thoại phổ thông, tăng 6% so với năm 2016.

Cuộc sống ngày càng hiện đại, nhu cầu giải trí càng cao. Có rất nhiều hoạt động giải trí mà người dân ở các thành phố lớn, điển hình là TP.HCM quan tâm như đi công viên, bảo tàng, mua sắm, xem hòa nhạc hoặc xem phim tại các rạp cinema,...

Đi xem phim cuối tuần nay đã trở thành một thói quen của nhiều người, không chỉ tại Việt Nam mà còn trên toàn thế giới. Đặc biệt là vào dịp lễ hoặc các kỳ nghỉ, vé xem phim được bán dễ dàng hơn. Việc tìm kiếm một vé xem phim không còn khó khăn nữa. Bằng cách sử dụng smartphone, bạn có thể đặt một vé phim bất cứ lúc nào bạn muốn. Rõ ràng là công nghệ đang dần thay đổi cuộc sống của bạn – làm cho cuộc sống của bạn trở nên thuận tiện hơn, bạn có thể kiểm soát thế giới bằng cách chạm vào màn hình.

Với mong muốn giúp đỡ những người yêu thích phim ảnh, đặc biệt là phim chiếu rạp, không bỏ lỡ bất kỳ bộ phim yêu thích nào cũng như tạo ra một kênh tương tác giữa ekip sản xuất phim và khán giả, nhóm của chúng em đã phát triển một ứng dụng **PhimNOW**  cung cấp thông tin về lịch chiếu của các bộ phim mới nhất tại các rạp phim lớn trong thành phố, dựa trên nền tảng công nghệ React Native, vừa thỏa mãn nhu cầu giải trí của mọi người lại vừa tiết kiệm được thời gian và công sức, chỉ với một cái chạm để cài đặt trên smartphone là người dùng đã hoàn toàn nắm rõ và đầy đủ lịch chiếu phim hàng ngày trên hệ thông các rạp lớn nhỏ trên toàn quốc, không bị bỏ lỡ những bộ phim mới nhất, những giây phút ý nghĩa bên cạnh người thân hay bạn bè hoặc người mình yêu thương... Với từng chuyên mục phim cập nhật hàng ngày trên hệ thống, có trailer preview và có đánh giá của khán giả, đảm bảo người dùng sẽ tiết kiệm thời gian và thoải mái hơn với những giây phút thư giãn tuyệt vời. Nhờ vào ứng dụng này, nhóm chúng em tin rằng những phản hồi, nhận xét và đánh giá, góp ý từ người xem sẽ là tín hiệu tốt cho những bộ phim sau này.

**1.2. Đối tượng người dùng**

Tất cả mọi người yêu thích xem phim chiếu rạp, muốn biết chính xác thời gian và rạp nào hiện đang chiếu những bộ phim mà họ yêu thích.

# **2. Về đồ án**

## **2.1. Nền tảng React Native**

Để hiểu rõ hơn về React Native là gì, trước tiên chúng ta cần phân biệt sự khác nhau giữa ứng dụng Hybrid và ứng dụng Native

* 1. *Ứng dụng Hybrid*

Là chương trình phần mềm trên điện thoại di động được viết dựa trên nền tảng Web (HTML5, CSS3, JavaScript), bản chất hoàn toàn là ứng dụng Web nhưng có thêm được các tính năng thao tác phần hệ điều hành như tập tin, truy cập máy ảnh,… Toàn bộ những thứ này đều được bao bọc bởi một lớp ứng dụng Native mà nổi bật là Phonegap/Cordova. Phần Web được xử lý hiển thị bởi Webview, phần tính năng truy cập hệ thống được cung cấp bởi các hàm API, gọi hàm bằng JavaScript thông qua API thì chương trình bao bọc sẽ gọi trực tiếp Native xuống hệ điều hành. Bằng cách này, ứng dụng Web có thêm những tính năng cao cấp của ứng dụng Native, và do được tính hợp mã nguồn sẵn nên tốc độ của ứng dụng Hybrid nhanh hơn ứng dụng Web.

* 1. *Ứng dụng Native (iOS, Android)*

Là ứng dụng được phát triển trực tiếp bằng ngôn ngữ của hệ điều hành đó cung cấp. Ví dụ với iOS là Objective-C, Swift và Android là Java. Các ứng dụng viết bằng ngôn ngữ này được biên dịch ra ngôn ngữ máy trên điện thoại và có toàn bộ tính năng mà hệ điều hành đó cung cấp. Do là ngôn ngữ trực tiếp cũng như không phải thông qua ứng dụng nào khác nên tốc độ là nhanh nhất.

Điểm hạn chế khi viết ứng dụng Native đó là tốn thời gian re-compile. Mỗi lần sửa đổi một chức năng bạn cần re-build. Với iOS khi sử dụng Swift cho Codebase lớn, bạn sẽ thấy tốc độ re-complie khá là chậm.

Nhưng tại sao ứng dụng Native vẫn được ưu chuộng và phát triển hơn ứng dụng Web hay ứng dụng Hybrid. Lý do rất đơn giản, các ứng dụng Native mang lại trải nghiệm tốt hơn cho người dùng.

c. React Native là gì ?

React Native là một framework giúp chúng ta viết các ứng dụng Native chỉ bằng JavaScript do Facebook phát hành. Nó sử dụng cùng thiết kế như React. Hiểu đơn giản bạn viết code JavaScript để có thể sử dụng cho cả thiết bị iOS và Android. Do đó React Native sẽ rất dễ dàng với người đã nắm vững JavaScript.

Sự ra đời của React Native để khắc phục các điểm yếu của ứng dụng Web và Hybrid. Với React Native, bạn không phải đang xây dựng một “Mobile Web App”, một “HTML5 App”, hay một “Hybrid App”. Bạn đang build một “Real Mobile App” cho tất cả các hệ điều hành mobile nhưng không phân biệt bạn đang dùng Objective-C, Swift hay Java. React Native xây dựng giao diện cơ bản giống như các ứng dụng iOS và Android thông thường. Bạn chỉ cần đặt các Component lại với nhau bằng JavaScript và React.

Sử dụng React Native sẽ giảm thời gian re-compiling, cho bạn cảm giác như real-time compiling. Thay vì phải mất thời gian re-compiling, React Native hỗ trợ Hot Reloading, đơn giản sử dụng tổ hợp phím Command⌘ + R trên iOS simulator.

React Native hỗ trợ “drop down to native code” (Objective-C, Swift hay Java) nếu bạn cần tùy chỉnh hoặc tối ưu một số trường hợp đặc biệt cho ứng dụng của bạn. Bạn sẽ dễ dàng xây dựng một phần/chức năng trong ứng dụng của bạn bằng React Native, và ngược lại. Đó cũng là cách mà các ứng dụng của Facebook đang sử dụng.

d. Ưu điểm & nhược điểm của React Native

* **Ưu điểm** **:**
* Mang lại trải nghiệm Native
* Lập trình bằng một ngôn ngữ JavaScript duy nhất
* Tốc độ build nhanh
* Đa nền tảng : Phát triển được trên cả Android & iOS
* Chức năng “Hot reloading”, không cần phải buil lại tất cả mã nguồn
* Phù hợp với nhóm phát triển nhỏ
* **Nhược điểm** **:**
* Vẫn còn thiếu các View Components cho Android như Map, Modal, Slider, Spinner hoặc các module như Camera Roll, Media, PushNotificationIOS.
* Không hỗ trợ Window Phone : Đây sẽ là điểm trừ lớn nếu so với ứng dụng Hybrid. Tuy nhiên, nếu bạn chỉ tập trung cho iOS và Android thì tất nhiên đây không phải là vấn đề lớn.
* Không build được ứng dụng iOS trên Window và Linux : Tất nhiên đây không phải là điểm yếu kém của bản thân React Native, mà đó là do yêu cầu từ Apple.
* React Native không thể build được ứng dụng quá phức tạp nếu bạn không biết Swift/Objecive-C, Java. Tính phức tạp ở đây là ứng dụng của bạn cần phải chỉnh sửa các Components. Bạn nên nhớ là để viết được một ứng dụng Native bằng JavaScript luôn luôn có sẵn các Components đã được viết từ Swift/Objective-C (iOS) và Java (Android) với tính chất 1 – 1. Trường hợp bạn muốn chỉnh sửa một Components nào đó như thay đổi thành phần hoặc thêm API thì bạn phải tự viết bằng chính ngôn ngữ tương ứng của iOS hoặc Android. Tin vui cho bạn là cũng nhiều lập trình viên khác đã viết nhiều Components cần thiết cho hầu hết ứng dụng (đây cũng là lý do vì sao Facebook biến React Native thành mã nguồn mở).
* Không nên dùng để viết Game có tính đồ họa và cách chơi phức tạp.
* Dùng ES2015 / ES6 : Đây là cấu trúc mới cho JavaScript từ 2015, vì khá là mới nên những cấu trúc của nó có thể bạn chưa quen, dẫn tới việc khó khăn trong việc tiếp cận.
* Bạn có thể phải mất nhiều thời gian để theo hết hệ sinh thái của React, tuy nhiên tất cả là để làm cho đời lập trình viên fullstack đẹp đẽ và những sản phẩm chất lượng hơn, bao gồm: ReactJS, React Native, Flux, Relay, GraphQL. Bạn không cần phải quan tâm những thứ này chỉ để viết ứng dụng với React native

*e. Cách hoạt động của React Native*

Ứng dụng viết bằng React Native được chia làm 2 phần : Phần hiển thị và phần xử lý.

Phần hiển thị được biên dịch từ JavaScript sẽ map với những Components của hệ thống. Ví dụ : Điều hướng, tab, touch… Phần hiển thị này được lấy cảm hứng từ Virtual DOM của React JS, mọi xử lý hiển thị được thực hiện trên một cây DOM ảo, sau đó được React Native render lại bằng Native View.

Phần xử lý vẫn được thực hiện trực tiếp bằng ngôn ngữ JavaScript. Ví dụ “1 + 1 = 2”, biểu thức này được xử lý dưới bộ core thực thi JavaScript, không cần phải thông dịch qua Java hay Swift/Objective-C rồi mới làm phép tính.

f. Một số ứng dụng nổi tiếng viết bằng React Native

Mạng xã hội cực kỳ mạnh mẽ trong cuộc sống của con người hiện nay và nó trở thành nhu cầu gần như thiết yếu trong cuộc sống mỗi ngày của mỗi người.

Sau đây là 3 trong số những ứng dụng vô cùng tuyệt vời và nổi tiếng ứng dụng React Native :

* **FACEBOOK**

React Native bắt đầu khi dự án hackathon của Facebook phát triển để đáp ứng nhu cầu của công ty. Facebook muốn mang lại tất cả các lợi ích của phát triển web – chẳng hạn như lặp đi lặp lại nhanh và có một đội xây dựng toàn bộ sản phẩm – di động. Đó là cách React Native đã được đưa vào cuộc sống và tận dụng trong phát triển ứng dụng dành cho thiết bị di động cho cả ứng dụng iOS và Android.

A hand holding a cell phone

Description generated with very high confidence

Nhóm lập trình đã chuyển đổi tính năng Events Dashboard trong ứng dụng Facebook cho iOS sang React Native để kiểm tra hiệu suất ứng dụng, chẳng hạn như thời gian khởi động, điều này rất quan trọng trong loại hình ứng dụng này.

Đây là một phần quan trọng trong ấn tượng ban đầu của người dùng về ứng dụng và xác định liệu chúng có tồn tại hay ra đi không. Những gì họ đạt được là cắt giảm thời gian tìm hiểu thị trường một nửa.

* **SKYPE**

Phiên bản mới của Skype hoàn toàn được cải tiến bắt đầu từ các icon cho toàn bộ bố cục, thêm một vài tính năng gọn gàng nữa. Như Microsoft cũng đã thông báo, Skype Preview cho iOS đã có sẵn với số lượng giới hạn thông qua TestFlight của Apple dành cho Skype Insiders.



Không chỉ Skype cho iOS mà còn là một phiên bản dành cho máy tính để bàn, đây sẽ là dự án React Native lớn đầu tiên của Windows. Đây là một dấu hiệu tốt cho nền tảng này. Chúng ta sẽ có thể thấy React Native hoạt động khá tốt trên Windows.

* **INSTAGRAM**

Instagram đã vượt qua thử thách để tích hợp React Native vào ứng dụng gốc hiện tại của họ, bắt đầu từ chế độ xem đơn giản nhất mà bạn có thể tưởng tượng : chế độ Push Notifications đã được triển khai thực hiện dưới dạng WebView. Nó không yêu cầu xây dựng cơ sở hạ tầng navigation, vì UI khá đơn giản.

A close up of a device

Description generated with high confidence

Đội ngũ lập trình tại Instagram đã phải đối mặt với một số vấn đề trong quá trình, nhưng họ đã cải thiện đáng kể tốc độ phát triển. 85% đến 99% code được chia sẻ giữa các ứng dụng Android và iOS, tùy thuộc vào sản phẩm, do đó nhóm đã có thể cung cấp ứng dụng nhanh hơn nhiều so với các giải pháp thông thường.

* Trên đây là 3 trong số rất nhiều ứng dụng viết bằng React Native để phát triển và đạt được thành công vô cùng rực rỡ, tiện ích trong cuộc sống của con người và sử dụng rộng rãi trong xã hội.
* Với tốc độ phát triển vượt bậc của công nghệ thông tin và mạng xã hội cũng như smarphone hiện nay thì tương lai của React Native vô cùng tươi sáng và định hướng sẽ tiếp tục phát triển lâu dài.

## **2.2. Công cụ lập trình**

Mặc dù chưa có IDE độc quyền nhưng ứng dụng React Native vẫn có thể được phát triển thông qua các trình biên tập văn bản như Sublime Text, Atom, Notepad++ hay Visual Studio Code, Xcode, Expo,…

Nhóm chúng em sử dụng Sublime Text 3 để biên tập mã nguồn bởi vì tính đơn giản, gọn nhẹ, nhanh chóng, dễ cài đặt và sử dụng.

## **2.3. Cài đặt React Native**

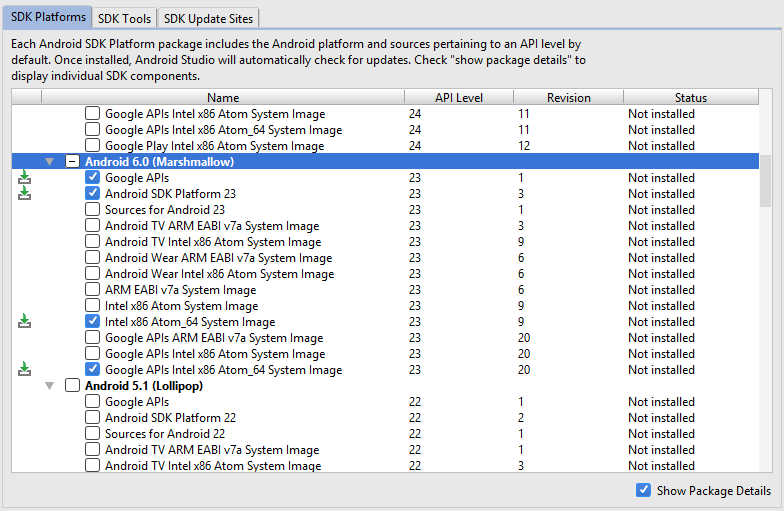
Sau đây là hướng dẫn cài đặt React Native để lập trình ứng dụng Android trên môi trường Windows. Với các hệ điều hành khác, bạn có thể tham khảo hướng dẫn cài đặt tại : [facebook.github.io/react-native/docs/getting-started.html](https://facebook.github.io/react-native/docs/getting-started.html)

* **Bước 1** : Cài đặt **Chocolatey** tại chocolatey.org/install. Mở cmd.exe với quyền quản trị và chạy đoạn lệnh sau :

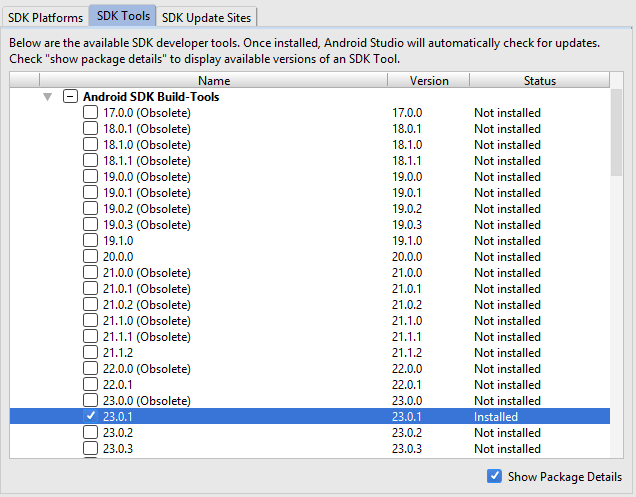
Ảnh có chứa ảnh chụp màn hình

Mô tả được tạo với mức tin cậy rất cao

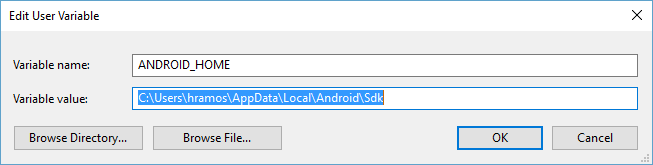
* **Bước 2** : Cài đặt **Node**, **Python 2**, **JDK**. Sau khi cài đặt **Chocolatey**, chạy đoạn lệnh sau : choco install -y nodejs.install python2 jdk8
* **Bước 3** : Cài đặt **React Native CLI**. Mở **Command Prompt**, chạy đoạn lệnh sau : npm install -g react-native-cli
* **Bước 4 :** Cài đặt **Android Studio** tại developer.android.com/studio/index.html. Sau khi cài đặt **Android Studio**, mở **SDK Manager**, chọn & cài đặt như hình sau :



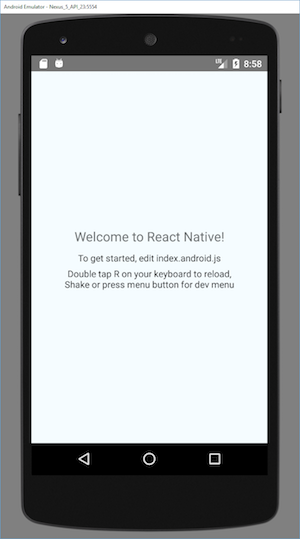
Tiếp tục cài đặt **Android SDK Build-Tools 23.0.1**



Cấu hình biến môi trường **ANDROID\_HOME** : Mở **Control Panel** → **System and Security** → **Change settings**... Chọn tab **Advanced** → **Environment Variables**... → **New**. Tìm đến thư mục cài đặt Android SDK



* **Bước 5** : Tạo ứng dụng đầu tiên
* Mở Command Line tại vị trí muốn khởi tạo project. Gõ lệnh sau : react-native init HelloWorld
* Khởi động AVD hoặc kết nối với điện thoại Android
* Chạy 2 dòng lệnh sau : cd HelloWorld → react-native run-android



# **3. Một số Components trong React Native**

**3.1. View**

* Chức năng : Là container chứa các Components khác

A screenshot of a cell phone

Description generated with high confidence

* Thuộc tính :

+ hitSlop

+ pointerEvents

+ removeClippedSubviews

+ collapsable

+ needsOffscreenAlphaCompositing

+ renderToHardwareTextureAndroid

* Phương thức :

+ onLayout()

+ onMagicTap()

+ nativeEvent() : *changedTouches, identifier, locationX, locationY, pageX, pageY, target, timestamp*

**3.2. ScrollView**

* Chức năng : Là container chứa các Components hiển thị nội dung kiểu danh sách, ví dụ như **FlatList**

A screenshot of a cell phone

Description generated with very high confidence

* Thuộc tính :

+ alwaysBounceVertical

+ pagingEnabled

+ removeClippedSubviews

+ scrollEnabled

+ showsHorizontalScrollIndicator

+ showsVerticalScrollIndicator

+ overScrollMode

* Phương thức :

+ scrollTo()

+ scrollToEnd()

+ scrollWithoutAnimationTo()

+ flashScrollIndicators()

**3.3. FlatList**

* Chức năng : Hiển thị nội dung kiểu danh sách

A screenshot of a cell phone

Description generated with very high confidence

* Thuộc tính :

+ data

+ horizontal

+ initialNumToRender

+ initialScrollIndex

+ numColumns

+ refreshing

* Phương thức :

+ renderItem()

+ getItemLayout()

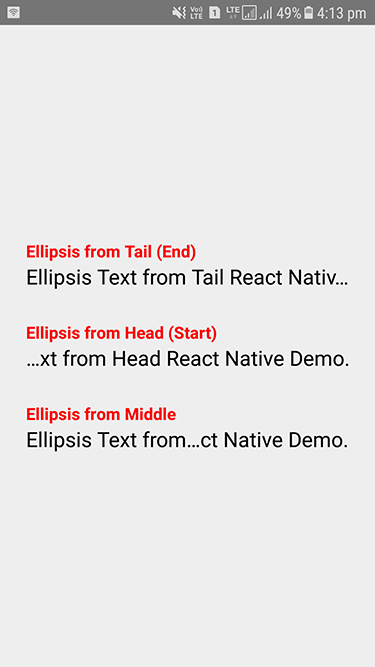
+ keyExtractor()

+ onEndReached()

+ onRefresh()

**3.4. Text**

* Chức năng : Render nội dung văn bản



* Thuộc tính :

+ selectable

+ ellipsizeMode : *head, middle, tail, clip*

+ numberOfLines

+ allowFontScaling

+ disabled

+ selectionColor

+ adjustsFontSizeToFit

+ minimumFontScale

+ suppressHighlighting

* Phương thức :

+ onLayout()

+ onLongPress()

+ onPress()

**3.5. TouchableOpacity**

* Chức năng : **TouchableOpacity** tương tự như **Button** nhưng khi chạm vào thì độ trong suốt giảm dần

A screenshot of a cell phone

Description generated with high confidence

* Thuộc tính :

+ activeOpacity

+ tvParallaxProperties

+ hasTVPreferredFocus

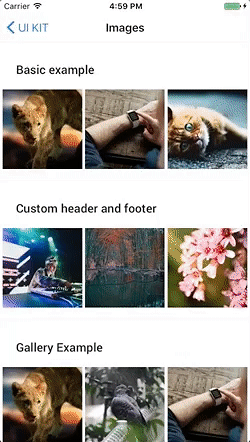
* Phương thức :

+ setOpacityTo()

+ onPress()

**3.6. Image**

* Chức năng : Dùng để render hình ảnh



* Thuộc tính :

+ blurRadius

+ resizeMode : *cover, contain, stretch, repeat, center*

+ source

+ loadingIndicatorSource

+ resizeMethod : *auto, resize, scale*

* Phương thức :

+ onLayout()

+ onLoad()

+ onLoadEnd()

+ onLoadStart()

+ onProgress()

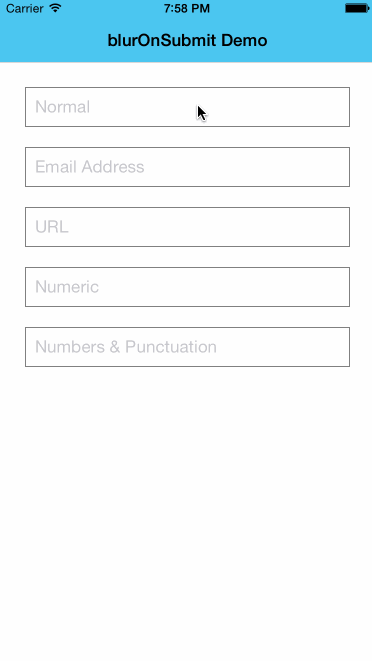
+ getSize()

+ prefetch()

+ queryCache()

**3.7. TextInput**

* Chức năng : Dùng để nhập dữ liệu dạng văn bản



* Thuộc tính :

+ allowFontScaling

+ autoCapitalize : none, sentences, words, characters

+ autoCorrect

+ autoFocus

+ blurOnSubmit

+ caretHidden

+ clearButtonMode : *never, while-editing, unless-editing, always*

+ clearTextOnFocus

+ contextMenuHidden

+ keyboardType : *default, numeric, email-address, phone-pad*

+ maxLength

+ multiline

+ numberOfLines

+ placeholder

+ value

* Phương thức :

+ onBlur()

+ onChange()

+ onChangeText()

+ onContentSizeChange()

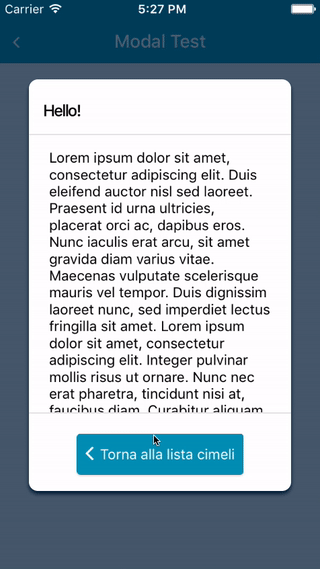
+ onEndEditing()

+ onFocus()

+ onKeyPress()

**3.8. Modal**

* Chức năng : Dùng để hiển thị nội dung nổi bật lên trên cùng so với các Components khác



* Thuộc tính :

+ visible

+ supportedOrientations : *portrait, portrait-upside-down, landscape, landscape-left, landscape-right*

+ transparent

+ animationType : *none, slide, fade*

+ presentationStyle : *fullScreen, pageSheet, formSheet, overFullScreen*

* Phương thức :

+ onRequestClose()

+ onShow()

+ onDismiss()

+ onOrientationChange()

**3.9. Picker**

* Chức năng : Dùng để tùy chọn nội dung, tương tự ComboBox trong C#

A screenshot of a cell phone

Description generated with very high confidence

* Thuộc tính :

+ selectedValue

+ enabled

+ mode : *dialog, dropdown*

+ prompt

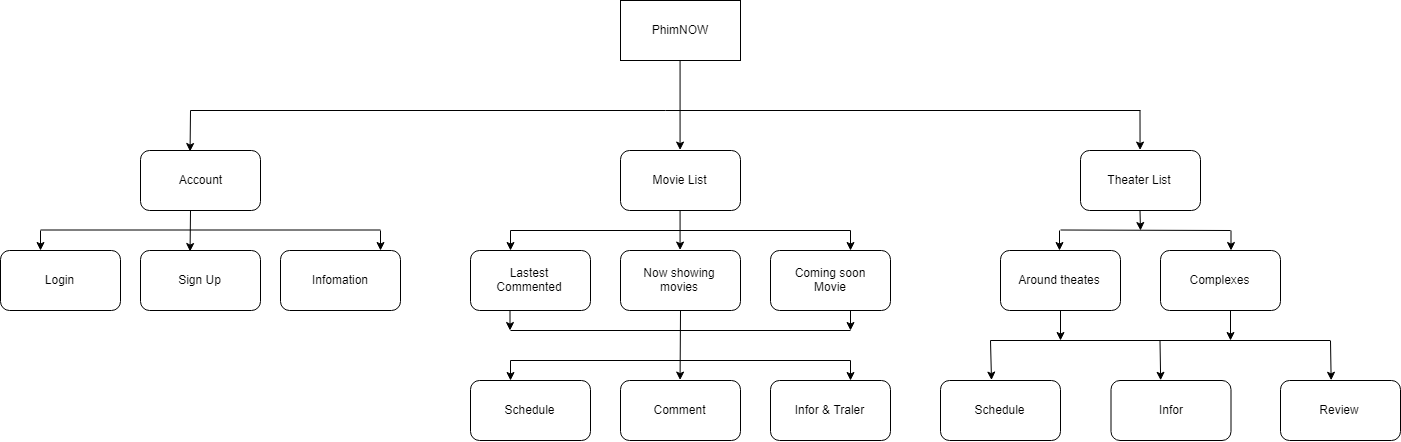
+ itemStyle

* Phương thức :

+ onValueChange()

**4. Mô hình nghiệp vụ**

**4.1. Sơ đồ phân rã chức năng**

****

**5. Cơ sở dữ liệu**

Để lấy dữ liệu phim mới nhất cho ứng dụng, chúng em sử dụng API phimmoi. Bởi vì dữ liệu từ website này đáng tin cậy và luôn được cập nhật liên tục.

**5.1. API 123 Phim**

API này được xây dựng bởi VNG Corporation, cung cấp thông tin của rất nhiều bộ phim, rạp phim, tin tức điện ảnh, kèm theo lịch chiếu phim mới nhất.

# **6. Thư viện sử dụng**

**6.1. React Native Navigation**

* Cài đặt : npm install --save react-navigation hoặc yarn add react-navigation

*a. StackNavigator*

* Chức năng : Là một stack chứa danh sách các màn hình. Trong đó, màn hình mới hơn xếp chồng lên màn hình cũ. StackNavigator giúp chuyển đổi từ màn hình hiện tại sang màn hình mới.
* Thuộc tính :

+ initialRouteName : Tên màn hình mặc định

+ initialRouteParams : Các tham số của màn hình mặc định

+ navigationOptions : Tùy chọn điều hướng mặc định cho các màn hình như title, header, gesturesEnabled,…

* Phương thức :

+ this.props.navigation.navigate( route , { params } ) : Chuyển sang màn hình với tên khai báo, đồng thời truyền các tham số sang màn hình đó

+ this.props.navigation.state.params.param : Gọi đến tham số do màn hình cũ truyền sang

*b. DrawerNavigator*

* Chức năng : Dùng để xây dựng giao diện menu bên góc trái màn hình

A screenshot of a cell phone

Description generated with very high confidence

* Thuộc tính :

+ contentOptions : Tùy chỉnh bố cục của menu : activeTintColor, activeBackgroundColor, onItemPress(route), itemsContainerStyle, itemStyle, labelStyle,…

+ drawerWidth : Độ rộng của Drawer

+ drawerPosition : Vị trí đặt Drawer (left / right)

+ drawerBackgroundColor : Màu nền của Drawer

* Phương thức :

+ this.props.navigation.navigate( route , { params } ) : Chuyển sang màn hình với tên khai báo, đồng thời truyền các tham số sang màn hình đó

+ this.props.navigation.state.params.param : Gọi đến tham số do màn hình cũ truyền sang

*c.* *TabNavigator*

* Chức năng : Dùng để xây dựng TabBar điều hướng giũa các màn hình

A screenshot of a cell phone

Description generated with very high confidence

* Thuộc tính :

+ tabBarPosition : Vị trí TabBar (top / bottom)

+ swipeEnabled : Cho phép vuốt để chuyển Tab

+ animationEnabled : Cho phép hiệu ứng chuyển Tab

+ initialLayout : Tùy chỉnh Width và Height cho TabBar

+ tabBarOptions : Tùy chỉnh nội dung Tab : activeTintColor, activeBackgroundColor, showLabel, tabStyle, allowFontScaling,…

+ screenProps : Lưu trữ tham số dùng chung giữa các Tab

* Phương thức :

+ previousScene(route, index) : Màn hình trước khi chuyển Tab

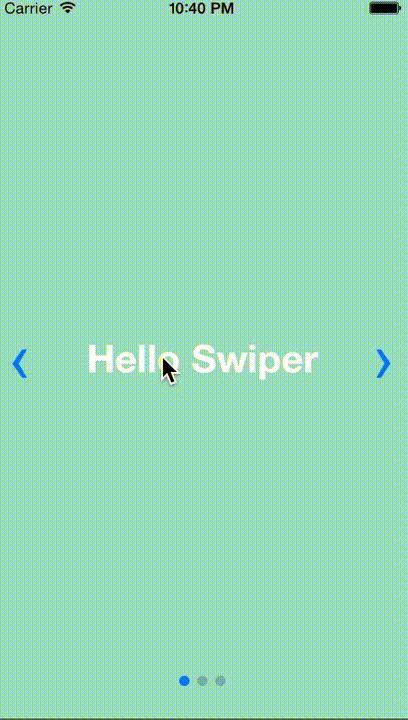
+ scene(route, index) : Màn hình khi được tap vào

+ jumpToIndex(index) : Chuyển đến Tab khác dựa vào index

+ this.props.screenProps.param : Gọi đến tham số dùng chung giữa các Tab

**6.2. React Native Swiper**

* Cài đặt : npm i react-native-swiper –save
* Chức năng : Xây dựng slider để hiển thị các banner nổi bật trong trang chủ



* Thuộc tính :

+ horizontal : Hiển thị danh mục theo chiều ngang (true)

+ loop : Chế độ lặp lại slideshow (true)

+ index : Index của slide khởi tạo (0)

+ showsButtons : Hiển thị các button điều khiển previous / next (true)

+ autoplay : Tự động chuyển slide (true)

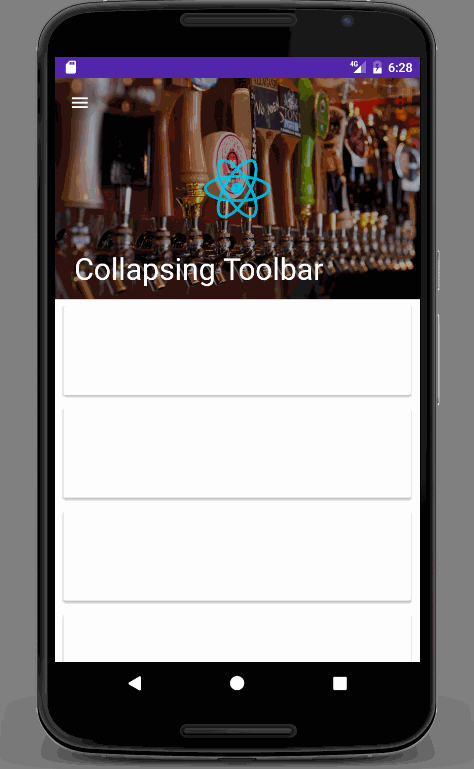
* Phương thức :

+ onIndexChanged(index) : Sự kiện khi index thay đổi

+ scrollBy(index, animated) : Sự kiện scroll

**6.3. React Native Collapsing Toolbar**

* Cài đặt : npm install react-native-collapsing-toolbar –save kèm theo npm install react-native-nested-scroll-view –save
* Chức năng : Hỗ trợ xây dựng giao diện Collapsing với Flatlist và tích hợp hiệu ứng Animation đẹp mắt



* Thuộc tính của CollapsingToolbarLayout :

+ title : Tiêu đề ToolBar

+ height : Chiều cao ToolBar khi mở rộng

+ scrimAnimationDuration : Thời gian chuyển đổi animation

+ contentScrimColor : Màu của ToolBar khi thu gọn

+ collapsedTitleTextColor : Màu tiêu đề của ToolBar khi thu gọn

+ collapsedTitleGravity : Căn lề cho tiêu đề ToolBar ("CENTER", "CENTER\_VERTICAL", "TOP", "LEFT", "RIGHT", "BOTTOM", "START", "END")

+ collapsedTitleTypeface : Font chữ tiêu đề khi thu gọn

+ expandedTitleColor : Màu chữ tiêu đề khi mở rộng

+ expandedTitleMargin : Khoảng cách giữa tiêu đề với đường biên ToolBar

+ expandedTitleGravity : Căn lề cho tiêu đề khi mở rộng ("CENTER", "CENTER\_VERTICAL", "TOP", "LEFT", "RIGHT", "BOTTOM", "START", "END")

+ expandedTitleTypeface : Font chữ tiêu đề khi mở rộng

* Phương thức của AppBarLayout :

+ show() : Mở rộng ToolBar

+ hide() : Thu gọn ToolBar

+ redraw() : Vẽ lại ToolBar

+ onOffsetChanged() : Sự kiện scroll khi thu gọn ToolBar

**6.4. React Native Modal Photo View**

* Cài đặt : npm install react-native-modal-photo-view –save
* Chức năng : Hiển thị danh sách hình ảnh dưới dạng Modal khi click vào

A screenshot of a cell phone

Description generated with very high confidence

* Thuộc tính :

+ mediaList : Mảng hình ảnh truyền vào

+ initialIndex : Index của hình ảnh hiển thị đầu tiên

+ displayCloseBtn : Hiển thị button Close

+ displayBottomBar : Hiển thị text chú thích dưới hình ảnh

+ visible : Hiển thị Modal hình ảnh

+ imageErrorTitle : Tiêu đề thông báo lỗi hiển thị hình ảnh

* Phương thức :

+ onClose() : Sự kiện khi đóng Modal

+ onPhotoLongPress() : Sự kiện long click vào hình ảnh

+ delayPhotoLongPress() : Sự kiện long click vào hình ảnh trong vòng \_ ms

**6.5. React Native Render HTML**

* Cài đặt : npm install react-native-render-html --save hoặc yarn add react-native-render-html
* Chức năng : Dùng để render HTML thành giao diện native



* Thuộc tính :

+ html : Chuỗi HTML truyền vào

+ uri : Đường dẫn của website muốn render

+ imagesMaxWidth : Độ rộng tối đa khi render hình ảnh

+ tagsStyles : Định nghĩa lại cách thức render cụ thể cho mỗi tag xác định

+ containerStyle : Style chứa nội dung render

+ baseFontStyle : Font chữ của nội dung render

* Phương thức :

+ onLinkPress() : Sự kiện click đường dẫn **href** của thẻ **a**

+ onParsed() : Sự kiện sau khi render hoàn tất

+ alterNode() : Xác định các node cụ thể và thay đổi chúng

**6.6. React Native Open Maps**

* Cài đặt : npm install --save react-native-open-maps
* Chức năng : Mở ứng dụng Maps mặc định của thiết bị để hiển thị vị trí với tọa độ đã xác định

A picture containing text, map

Description generated with very high confidence

* Thuộc tính :

+ latitude : Vĩ độ

+ longitude : Kinh độ

+ zoom : Độ phóng đại

+ provider : Google / Apple

+ name : Tên marker

* Phương thức :

+ open(options) : Mở ứng dụng Maps

+ createOpenLink(options) : Tạo đường dẫn mở ứng dụng Maps

+ createMapLink(options) : Tạo đường dẫn Map

**6.7. React Native Vector Icons**

* Cài đặt : npm install react-native-vector-icons –save
* Chức năng : Sử dụng các Icon có sẵn do FontAwesome, MaterialIcons, Ionicons, Entypo,… cung cấp

A screenshot of a cell phone

Description generated with very high confidence

* Thuộc tính :

+ size : Kích thước Icon (12)

+ name : Tên Icon

+ color : Màu sắc Icon

+ iconStyle : Style của Icon

+ backgroundColor : Màu nền của Icon

* Phương thức :

+ onPress() : Sự kiện click vào Icon

**6.8. React Native Orientation**

* Cài đặt : npm install react-native-orientation –save
* Chức năng : Lắng nghe sự thay đổi hướng của thiết bị và thiết lập hướng mong muốn

**A close up of a sign

Description generated with high confidence**

* Phương thức :

+ getInitialOrientation() : Lấy hướng hiện tại của thiết bị

+ lockToPortrait() : Khóa dọc

+ lockToLandscape() : Khóa ngang

+ addOrientationListener() : Lắng nghe & xử lý sự kiện thay đổi hướng

**7. Giao diện thiết kế**

**7.1. Danh sách màn hình**

* Màn hình Đăng ký
* Màn hình Đăng nhập
* Màn hình Đăng nhập thành công
* Màn hình Chọn thể loại
* Màn hình Thông tin phim
* Màn hình Lịch chiếu phim
* Màn hình Phim hôm nay

**7.2. Chi tiết thiết kế**

* 1. Màn hình Đăng ký



6

5

4

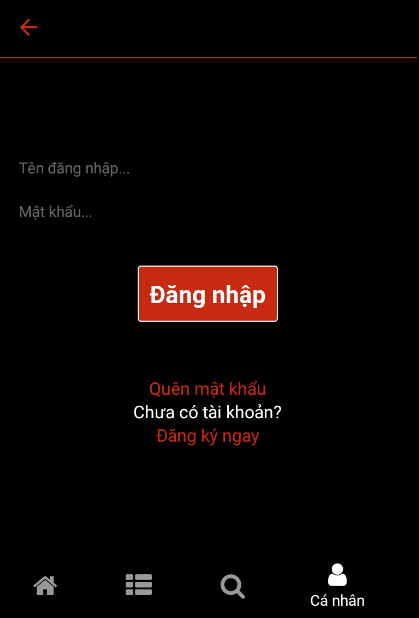
3

2

1

|  |  |  |
| --- | --- | --- |
| STT | Thành phần | Xử lý |
| 1 | Đăng ký tài khoản | Click hoặc chạm để chuyển về mục đăng ký |
| 2 | Đăng nhập | Chạm để chuyển về màn hình đăng nhập |
| 3 | Họ tên | Điền thông tin họ tên |
| 4 | Tên đăng nhập | Điền thông tin tên đăng nhập |
| 5 | Mật khẩu | Điền thông tin mật khẩu |
| 6 | Xác nhận mật khẩu | Điền thông tin xác nhận mật khẩu |

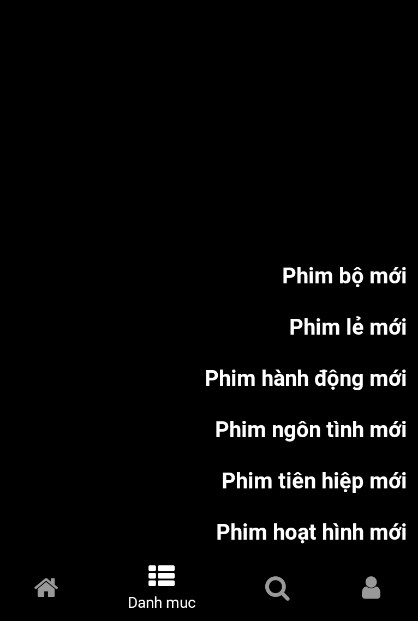
* 1. Màn hình Đăng nhập



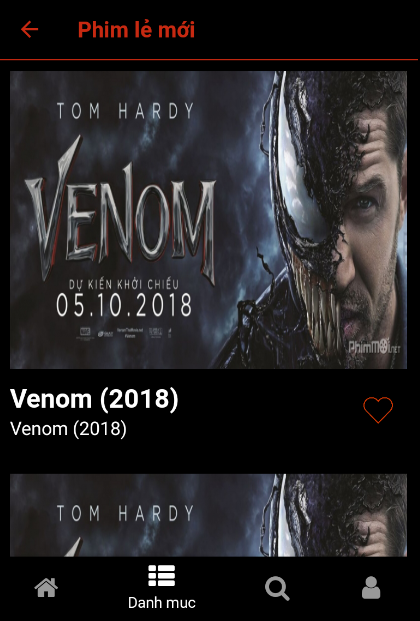
* 1. Màn hình Đăng nhập thành công



* 1. Màn hình Chọn thể loại

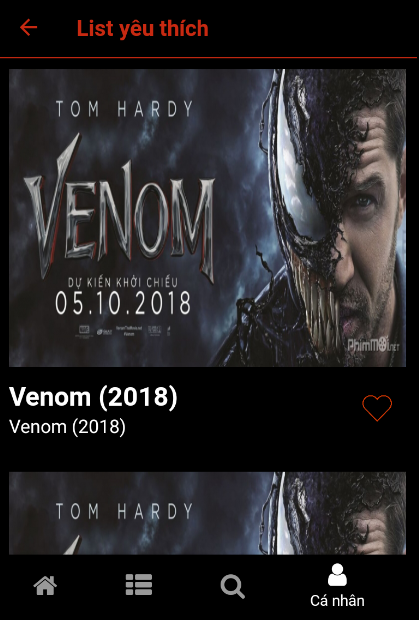


* 1. Màn hình Thông tin phim

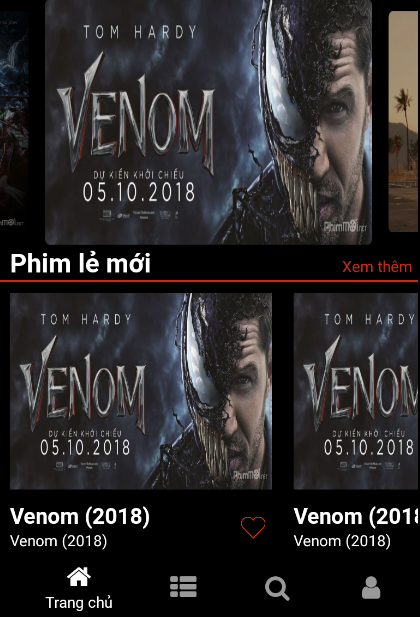




* 1. Màn hình Lịch chiếu phim



* 1. Màn hình Phim hôm nay
  2. Màn hình Phim hôm nay



2

1

|  |  |  |
| --- | --- | --- |
| STT | Thành phần | Xử lý |
| 1 | Benner | Slide chạy thông tin phim |
| 2 | Phim mới hôm nay | Thông tin phim mới hôm nay |

**8. Tài liệu tham khảo**

* API PhimNOW : www.phimmoi.net
* Trang chủ React Native : facebook.github.io/react-native/docs/tutorial.html
* React Native Navigation : [reactnavigation.org](https://reactnavigation.org)
* React Native Swiper : github.com/leecade/react-native-swiper
* React Native Collapsing Toolbar : github.com/cesardeazevedo/react-native-collapsing-toolbar
* React Native Modal Photo View : github.com/guopeng1994/react-native-modal-photo-view
* React Native Render HTML : github.com/archriss/react-native-render-html
* React Native Open Maps : github.com/brh55/react-native-open-maps
* React Native Vector Icons : github.com/oblador/react-native-vector-icons
* React Native Orientation : github.com/yamill/react-native-orientation

**9. Phân công nhiệm vụ**

|  |  |  |
| --- | --- | --- |
| **MSSV** | **Họ tên** | **Nhiệm vụ** |
| 14520683 | Nguyễn Hoài Phú | * Code * Thảo luận hỗ trợ nhóm |
| 13521007 | Phan Văn Tùng | * Viết báo cáo * Làm bài tập * Thảo luận hỗ trợ nhóm |

**10. Tổng kết – Đánh giá**

React Native là một công nghệ tương lai có thể thay thế nền tảng Native như Android và iOS để xây dựng ứng dụng hỗ trợ đa nền tảng. Nhóm nghiên cứu của chúng em thấy nó thực sự thú vị và đáng để nghiên cứu nhiều hơn.

Chúng em xin cảm ơn thầy Huỳnh Tuấn Anh đã giới thiệu cho chúng em một số kiến thức hữu ích về lập trình ứng dụng React Native để hoàn thành tốt đồ án này.

Mặc dù gặp nhiều khó khăn khi tiếp cận một framework mới như React Native, nhưng chúng em đã biết cách khắc phục nhờ việc tổ chức làm việc nhóm hiệu quả. Môn học thực sự mang lại nhiều cảm hứng và kinh nghiệm mới mẻ để chúng em tiếp tục theo đuổi con đường phát triển ứng dụng React Native trong tương lai.

**--- HẾT ---**