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

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**



**--------------------**

**BÁO CÁO   
LẬP TRÌNH TRÊN THIẾT BỊ DI ĐỘNG**

**Đề tài: TÌM HIỂU VỀ REACT NATIVE**

**VÀ XÂY DỰNG ỨNG DỤNG MINH HỌA**

**Giảng viên hướng dẫn:**

**Ths. HUỲNH TUẤN ANH**

**Sinh viên thực hiện:**

**NGUYỄN TUẤN HUY 16520526**

**TRẦN GIA BẢO 16520088**

**TRẦN HOÀNG KHA 16520549**

***TPHCM, 20/01/2021***

***Năm học 2007 - 2008***

# Nhận xét của giảng viên hướng dẫn

…………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

# Lời cảm ơn

*Trước tiên em xin gửi lời cảm ơn đến thầy Huỳnh Tuấn Anh. Cảm ơn thầy đã nhiệt tình hỗ trợ những thông tin cần thiết và giải đáp những thắc mắc cho nhóm và các bạn trong suốt quá trình thực hiện đề tài.*

*Đồng thời em cũng muốn cảm ơn các anh chị khóa trên, đặc biệt là anh chị trong khoa đã chia sẽ kinh nghiệm quý báu về môn học cũng như những kiến thức liên quan. Cũng xin cảm ơn bạn bè đã tạo điều kiện thuận, mọi người đã đưa ra nhận xét và góp ý chân thành, vô cùng quý giá. Những người đã động viên, hỗ trợ nhóm hoàn thành đề tài.*

***Nhóm thực hiện***

*Thủ Đức, 20 tháng 1 năm 2021*

**MỤC LỤC**

[Nhận xét của giảng viên hướng dẫn 2](#_Toc62248996)

[Lời cảm ơn 3](#_Toc62248997)

[1 Giới thiệu tổng quan 1](#_Toc62248998)

[1.1 Thông tin, đánh giá nhóm 1](#_Toc62248999)

[1.2 Tổng quan đề tài 1](#_Toc62249000)

[1.2.1. Bài toán 1](#_Toc62249004)

[1.2.2. Phạm vi nghiên cứu 1](#_Toc62249005)

[1.2.3. Nội dung nghiên cứu 2](#_Toc62249006)

[1.2.4. Kết quả hướng tới 2](#_Toc62249007)

[2 Tìm hiểu về React Native 3](#_Toc62249008)

[2.1 Giới thiệu: 3](#_Toc62249009)

[2.2 Lịch sử phát triển: 4](#_Toc62249010)

[2.3. Lý do React Native được ưa chuộng sử dụng: 5](#_Toc62249014)

[2.4. Cách thức hoạt động của React Native: 5](#_Toc62249015)

[2.5. Các ứng dụng được viết bằng React Native: 6](#_Toc62249016)

[2.6. Các phiên bản của React Native 13](#_Toc62249017)

[2.6.1. Phiên bản 0.60 13](#_Toc62249018)

[2.6.2. Phiên bản 0.61 16](#_Toc62249019)

[2.6.3. Phiên bản 0.62 17](#_Toc62249020)

[2.6.4. Phiên bản 0.63 21](#_Toc62249021)

[2.7. Các ưu và nhược điểm của React Native 23](#_Toc62249022)

[2.7.1. Ưu điểm của React Native 23](#_Toc62249023)

[2.7.2. Nhược điểm của React Native 24](#_Toc62249024)

[3 Xây dụng ứng dụng di động sử dụng framework React Native: 25](#_Toc62249025)

[3.1 Khảo sát các ứng dụng liên quan 25](#_Toc62249026)

[3.2 Quy trình phát triển ứng dụng: 28](#_Toc62249027)

[3.2.1. Mô tả phần mềm 28](#_Toc62249031)

[3.2.2. Phân tích yêu cầu 29](#_Toc62249032)

[3.2.3. Phân tích thiết kế hệ thống 29](#_Toc62249033)

[3.2.4. Thiết kế giao diện hệ thống 45](#_Toc62249034)

[3.3. Lợi ích của framework React Native đối với ứng dụng 51](#_Toc62249043)

[3.4. Đóng góp của nhóm 52](#_Toc62249044)

[4 Thử nghiệm và đánh giá: 52](#_Toc62249045)

[4.1 Môi trường thử nghiệm: 52](#_Toc62249046)

[4.2 Kết quả thử nghiệm: 52](#_Toc62249047)

[4.3 Đánh giá kết quả 53](#_Toc62249048)

[5 Kết quả đạt được và hướng phát triển: 53](#_Toc62249049)

[5.1 Kết quả đạt được 53](#_Toc62249050)

[5.2 Hướng phát triển 53](#_Toc62249051)

[6 Tài liệu tham khảo 53](#_Toc62249052)

# Giới thiệu tổng quan

## Thông tin, đánh giá nhóm

|  |  |  |  |
| --- | --- | --- | --- |
| MSSV | Họ tên | Email | Vai trò |
| 16520526 | Nguyễn Tuấn Huy | 16520526@gmail.com | Trưởng nhóm |
| 16520088 | Trần Gia Bảo | 16520088@gmail.com | Thành viên |
| 16520549 | Trần Hoàng Kha | [16520549@gmail.com](mailto:16520549@gmail.com) | Thành viên |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| MSSV | Họ tên | Tỷ lệ % công việc được giao | Tỷ lệ % công việc hoàn thành tốt | Nhận xét |
| 16520526 | Nguyễn Tuấn Huy | 40% | 100% |  |
| 16520088 | Trần Gia Bảo | 40% | 100% |  |
| 16520549 | Trần Hoàng Kha | 30% | 100% |  |

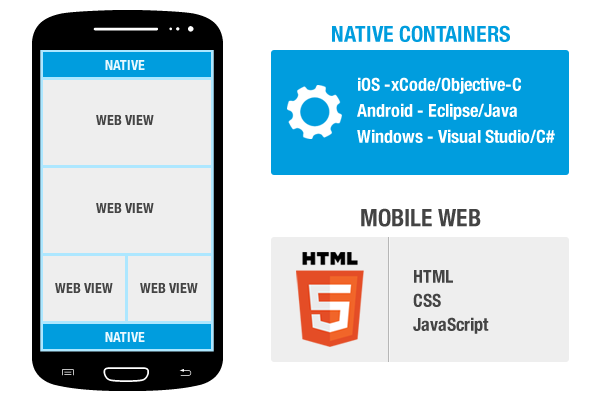
## Tổng quan đề tài



### Bài toán

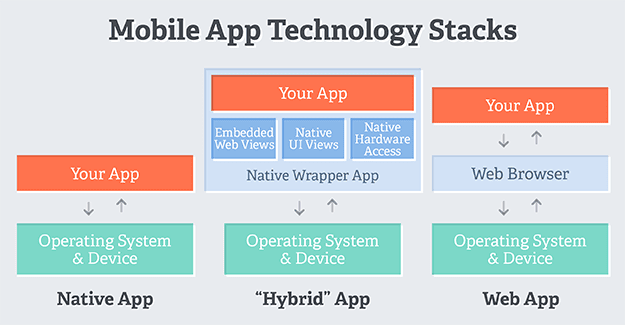
Hiện nay theo sự phát triển của xã hội, nhu cầu tạo ra một ứng dụng mobiles và ứng dụng web đồng nhất và nhanh nhất ngày càng tăng. Tuy nhiên do rào cản nền tảng và mong muốn đó chưa được thực hiện.

Thời gian gần đây xuất hiện nhiều framework cho phép sử dụng các ngôn ngữ chạy trên web để phát triển ứng dụng mobile hay còn gọi là **cross-platform development** như PhoneGap, Ionic, Cordova,.. tuy nhiên ứng dụng tạo ra chỉ đạt đến mức **hybrid-app**, sử dụng các Webviews để hiển thị giao diện lên chứ **không thực sự là native**.



Việc sử dụng các ứng dụng hybrid làm **giảm hiệu năng** và **không tối ưu** trong việc xử lý.

Google cũng công bố cho mình một hướng tiếp cận khác, **hoàn toàn bỏ đi native app** mà sử dụng Progressive Web App (**PWA**), tuy hiệu năng khá tốt, nhưng vẫn không thể nào so sánh được với native app khi ứng dụng trở nên lớn hơn, mặt khác PWA được cho là**khó giữ lại người dùng** hơn là native app.



React Native cung cấp cho bạn một thứ hoàn toàn khác biệt, ứng dụng được tạo ra sẽ **hoàn toàn native** với hiệu năng cao, chứ không phải là “mobile web app”, “HTML5 app” hay “hybrid app”. Bạn có thể build một ứng dụng mobile mà khó có thể phân biệt được nó với các native app khác tạo hoàn toàn bằng Java hay Swift/Objective-C. React Native sử dụng các UI hoàn toàn giống với iOS hay Android app thông thường, điều đặc biệt là nó được **xây dựng dễ dàng** hơn thông qua Javascript và React.

### Phạm vi nghiên cứu

Trong đồ án lần này, nhóm chủ yếu tập trung vào nghiên cứu tổng quan về lịch sử hình thành và cách thức hoạt động của framework React Native vào việc phát triển ứng đa nền tảng (IOS và Android). Với các phiên bản thử nghiệm trên các thiết bị ảo (trong phạm vi nghiên cứu các thức lưu trữ trên thiết bị) và các thiết bị thật đã đi đến kết luận thành công trong việc ứng dụng framework React Native này vào môi trường thực tế.

Thông tin các thiết bị được thử nghiệm trong đề tài lần này: bao gồm các thiết bị:

* Sony XZ1
* Pixel 3a XL API 29
* Iphone Xr IOS 14.3

### Nội dung nghiên cứu

Nghiên cứu chi tiết về quá trình hình thành, cách thức hoạt động và tác dụng của framework React Native đối với việc phát triển các ứng dụng trên thiết bị dị động. Và xây dựng ứng dụng framework React Native.

### Kết quả hướng tới

Với đề tài này, nhóm chúng có 2 mục tiêu chính:

* Đối với các lập trình viên: mang đến sự hứng thú trong trong việc lập trình trên các thiết bị di động, khuyến khích cộng đồng React Native phát triển các ứng dụng thông minh hơn và ít tốn chi phí hơn, qua đó thúc đẩy sự phát triển của công nghệ.
* Đối với người dùng: mang đến sự tiện lợi từ việc đồng nhất trãi nghiệm sản phẩm trên cả hai nên tảng là IOS và Android. Qua đó giúp gia tăng sự hài lòng của người dùng đến với sản phẩm phần mềm.

# Tìm hiểu về React Native

## Giới thiệu:

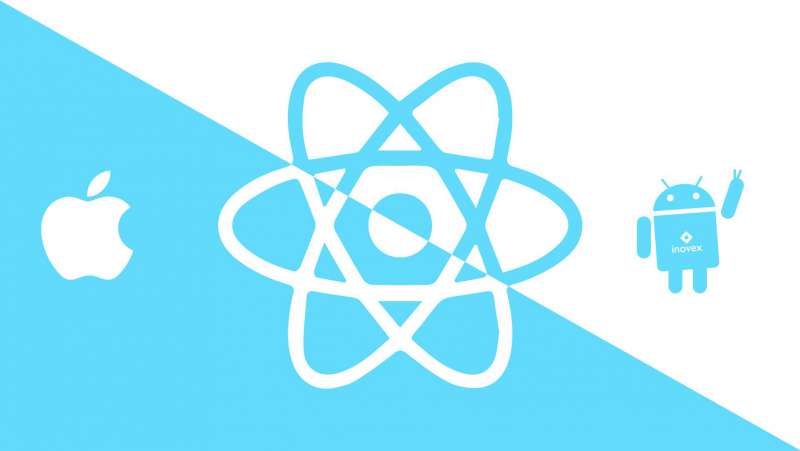
React Native là framework, do công ty Công nghệ sở hữu mạng xã hội được nhiều người sử dụng nhất hiện nay Facebook phát triển, nhằm mục đích giải quyết bài toán hiệu năng của Hybrid và chi phí khi phải viết nhiều loại ngôn ngữ native trên nền tảng di động. Nếu bạn đang sử dụng React Native để xây dựng các ứng dụng trên IOS hay Android thì chỉ có thể dùng ngôn ngữ JavaScript để viết chương trình.

React Native cho phép xây dựng và phát triển ứng dụng Native đa nền tảng một cách dễ dàng, khác với Mobile Web App, HTML5 App và Hybrid App.

Mục đích khi tạo ra ngôn ngữ này là giúp khắc phục được các điểm yếu của các ứng dụng web. Các nhà lập trình tiết kiệm được khá nhiều thời gian cũng như công sức của mình nhờ sự hỗ trợ đắc lực từ JavaScript mà họ có thể hoạt động được trên mọi mặt trận web.

Đây là một trong những framework sử dụng cấu hình thiết kế tương tự như React.

Hầu hết các giao diện và chức năng của nó đều được cấu thành từ rất nhiều thành phần con. React Native được sử dụng để phát triển cho rất nhiều các ứng dụng di động khác như:  Android, iOS, Web, UWP.



## 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](https://vi.wikipedia.org/wiki/Giao_di%E1%BB%87n_ng%C6%B0%E1%BB%9Di_d%C3%B9ng) cho iOS bằng một luồng [JavaScript](https://vi.wikipedia.org/wiki/JavaScript). Họ quyết định tổ chức cuộc thi [Hackathon](https://vi.wikipedia.org/wiki/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](https://vi.wikipedia.org/wiki/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](https://vi.wikipedia.org/wiki/Facebook) đã sử dụng React Native trong phát triển ứng dụng nhóm và ứng dụng quản lí quảng cáo của họ.



## Lý do React Native được ưa chuộng sử dụng:

Hybrid apps thường sẽ được sử dụng cho bất kỳ các nền tảng nào dựa vào một code base duy nhất chạy trên nhiều nền tảng, nên chúng sẽ có giá rẻ hơn so với native apps.

Tuy nhiên, theo các chuyên gia thì Hybrid apps chỉ là một lựa chọn hợp lý cho những doanh nghiệp có nhu cầu phát triển các sản phẩm có ngân sách nhỏ và thời gian hạn hẹp. Với các dự án lớn muốn xây dựng lâu dài thì các hiệu năng đòi hỏi thường sẽ rất cao để có thể gia tăng được trải nghiệm cho khách hàng.

Trong khi đó, Hybrid apps thường có hiệu năng thấp và không thể tương tác được cùng với các tài nguyên hệ thống phần cứng và phần mềm. Chính vì vậy, trong trường hợp này thì Native sẽ là lựa chọn phù hợp hơn cả.

Native app có thể nâng cao tương tác nhanh hơn do chúng được xây dựng với framework có nguồn gốc phát triển từ platform. Ngoài ra, chúng còn có khả năng hoạt động với nhiều chế độ ngoại tuyến, nên nó có thể tiếp cận được cả những khách hàng không có mạng internet.

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.

* 1. Cách thức hoạt động của React Native:

React Native hoạt động chủ yếu nhờ vào sự tích hợp của 2 thread là Main Thread và JS Thread cho các ứng dụng mobile.

* Main Thread sẽ đảm nhận các vai trò trong việc cập nhật các giao diện người dùng UI, rồi xử lý các tương tác của người dùng.
* JS Thread có nhiệm vụ thực thi và tiến hành xử lý code Javascript.

Đây là 2 luồng hoạt động hoàn toàn đối lập nhau trong React. Chính vì vậy, để hai Thread có thể tương tác được với nhau thì nó sẽ sử dụng một cầu nối Bridge. Chúng sẽ cho phép chúng có thể giao tiếp mà không quá phụ thuộc vào nhau cũng như các chuyển đổi từ thread này sang thread khác. Các dữ liệu từ hai Thread này đều sẽ được vận hành ngay khi tiếp nối các dữ liệu cho nhau.

* 1. Các ứng dụng được viết bằng React Native:

Mặc dù, ngôn ngữ này chưa được hoàn thiện nhưng các lập trình viên có thể sử dụng React Native để đẩy nhanh quá trình xây dựng các ứng dụng trên nhiều nền tảng khác nhau.

**Facebook**

Các nhà phát triển ứng dụng này đã chuyển đổi tính năng Event Dashboard cho iOS sang React Native để kiểm tra hiệu suất ứng dụng. Mục đích để cắt giảm thời gian tìm hiểu thị trường đi một nửa.



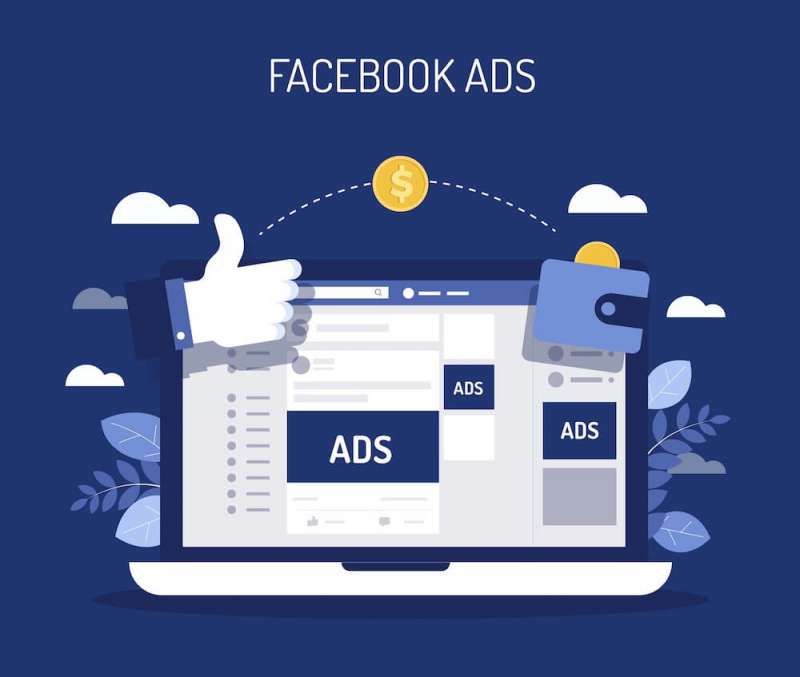
**Skype**

Họ đang thử nghiệm một ứng dụng Android mới bằng React Native. Có thể là tin vui cho dành cho những người thường xuyên sử dụng Skype để trao đổi công việc. Phiên bản Skype mới được phát triển bằng ngôn ngữ React Native sẽ thực sự được cải tiến với nhiều tính năng nổi bật nhưng lại rất gọn gàng.



**Facebook Ads**

Đến thời điểm hiện tại, những ứng dụng quảng cáo trên Facebook đều được sử dụng bằng ngôn ngữ React Native. Framework này đã hoàn toàn chiếm được niềm tin của nhiều nhà kinh doanh, giúp việc xử lý các định dạng quảng cáo như múi giờ, ngày, tiền tệ, được trở nên thuận tiện và dễ dàng hơn.



**Instagram**

Mọi ứng dụng gốc của Instagram đã vượt qua thử thách để tích hợp React Native. 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.



**Walmart**

Trước đây, Walmart đã chứng minh sự táo bạo của mình khi giới thiệu NodeJS vào stack của ứng dụng nhưng vài năm sau họ đã viết lại tất cả ứng dụng trên thiết bị di động bằng React Native. Đây cũng là lý do làm cho hiệu suất ứng dụng trên cả hai hệ điều hành IOS và Android được cải tiến thông qua việc sử dụng ít tài nguyên hơn trong khoảng thời gian ngắn.



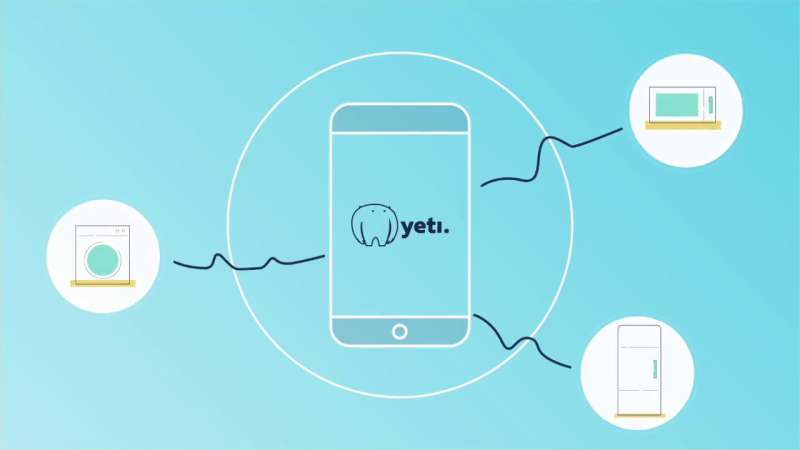
**Uber Eats**

Uber Eats đang tận dụng React Native trong kỹ thuật của ứng dụng nhằm phân phối thực phẩm của họ.



**Yeti Smart Home**

Mặc dù cũng gặp những vấn đề tương tự khi tích hợp React Native vào ứng dụng SoundCloud Pulse. Tuy nhiên, sau một thời gian được sự hỗ trợ và đóng góp của cộng đồng thì ứng dụng đã có thể vượt khỏi màn hình di động mà sử dụng được trên nhiều thiết bị khác nhau.



* 1. Các phiên bản của React Native

Phiên bản đầu tiên được Facebook công bố lần đầu tiên vào tháng 3 năm 2016 là 0.5. Sau đó, React Native  dần được phát triển thường xuyên với các bản như: bản 0.61.5 được phát hành vào tháng 11 năm 2019. Hiện nay, Facebook đã cho ra đời phiên bản mới nhất là 0.63.

Các phiên bản nổi bật:

* + 1. Phiên bản 0.60

**Những điểm nổi bật:**

* **React Native** là một framework ứng dụng di động mã nguồn mở, được phát triển bởi **Facebook**. Đây là một trong những nguồn phổ biến nhất để phát triển ứng dụng Đa nền tảng cho Android, iOS, Windows, v.v.
* Với công nghệ vượt trội, [**React Native**](https://cafedev.vn/gioi-thieu-chi-tiet-va-day-du-ve-react-native-demo/#Tong_ket) tiết kiệm thời gian trong việc phát triển ứng dụng so với các framework khác. Nó giảm chi phí phát triển ứng dụng từ **40% đến 50%.**
* Các bản cập nhật mới bao gồm các **cải tiến về khả năng truy cập**, **làm mới màn hình wellcome**, **CocoaPods cho iOS và hơn thế nữa.**

Mới đây, React Native, do Facebook tạo ra đã phát hành phiên bản mới 0.60. Và được hưởng ứng từ nhiều developer với khá nhiều sự phấn khích và tò mò khi thấy tất cả các tính năng tuyệt vời mà [**React Native**](https://cafedev.vn/gioi-thieu-chi-tiet-va-day-du-ve-react-native-demo/#Tong_ket) cung cấp. Trước khi đi sâu vào các tính năng mới, chúng ta hãy làm quen ngắn gọn với framework này.

[React Native](https://cafedev.vn/gioi-thieu-chi-tiet-va-day-du-ve-react-native-demo/#Tong_ket) là một framework nguồn mở, về cơ bản khác với phiên bản React chính. Nó hoạt động trong với các tiến trình chạy ngầm và dùng ngôn ngữ JavaScript để dev. Cùng với sự xuất hiện của các bản cập nhật mới trong phiên bản mới nhất 0.60, giờ đây người dùng có thể tối ưu hóa các tính năng hiện có của [React Native](https://cafedev.vn/gioi-thieu-chi-tiet-va-day-du-ve-react-native-demo/#Tong_ket) 0.59. Hơn nữa, nhiều vấn đề với phiên bản trước cũng đã được sửa chữa để mang lại trải nghiệm thoãi mái cho người dùng. Dưới đây là một số cập nhật mới trong React Native 0.60.

#### **Cải tiến trong khả năng truy cập**

Bản phát hành 0.60 mang đến nhiều cải tiến của API để tăng khả năng truy cập cho cả iOS và Android. Danh sách các cải tiến như [announceForAccessibility](https://github.com/facebook/react-native/commit/cfe0032) và các tiến bộ trong hỗ trợ [hành động](https://github.com/facebook/react-native/commit/14b4668), [roles](https://github.com/facebook/react-native/commit/1aeac1c), [flags](https://github.com/facebook/react-native/commit/0090ab3) và hơn thế nữa. Nhóm React Native hy vọng rằng với những thay đổi này sẽ giúp các app của chúng ta dễ dàng truy cập vào iOS và Androd như ứng dụng bản địa.

#### **Màn hình mới**

Với bản phát hành mới này, đi kèm với màn hình ứng dụng mới – Màn hình “Wellcome to React”. Nó được khẳng định là thân thiện với người dùng hơn, vì nó hiển thị các hướng dẫn hữu ích như chỉnh sửa App.js, Documents links, cách bắt đầu với debug menu và cuối cùng là làm thế nào để phù hợp với việc thiết kế lại trang web sắp tới. UI mới “Hello world” sẽ giới thiệu cho hệ sinh thái người dùng theo cách tốt hơn.

#### **Hỗ trợ cho AndroidX**

Với việc migration React Native với AndroidX (Android Extension Library), các code và thư viện gốc sẽ cần được migration. Do sự thay đổi này, tất cả các ứng dụng React Native sẽ phải sử dụng AndroidX. Tuy nhiên, nhóm React Native đã phát triển một giải pháp tạm thời cho vấn đề này. Nó được gọi là “[Jetifier](https://stackoverflow.com/questions/51680671/what-is-jetifier)”, đây là một công cụ chuyển tiếp AndroidX với kiểu dáng tương thích với React Native.

Lưu ý: Một app chỉ có thể chạy được Andoird X or chạy trên hệ điều hành khác. Hiện tại bên React Native không hỗ trợ chạy app trên cùng nhiều hệ điều hành khác vs thèn Android X.

#### [**Auto-linking**](https://github.com/react-native-community/cli/blob/master/docs/autolinking.md)**for OS**

Các thư viện React Native thường bao gồm code thuẩn của nó(Javascript) và một số code của từng nền tảng cụ thể. Cơ chế mới này cho phép dự án của bạn tự link và sau đó sử dụng code đó. Và hầu hết các thư viện,… sẽ không yêu cầu sử dụng react-native link nữa. Nhóm React Native CLI đã giới thiệu những tiến bộ lớn như thế này trong [auto-linking](https://github.com/react-native-community/cli/blob/master/docs/autolinking.md).

#### **CocoaPods là một phần của iOS**

CocoaPods là một trình quản lý các thư viện nào đó cần được dùng cho ứng dụng của Swift và Objective-C. **Theo mặc *định***, [React Native đi kèm với CocoaPods trong iOS.](https://github.com/react-native-community/discussions-and-proposals/blob/master/proposals/0004-cocoapods-support-improvements.md)

#### **Loại bỏ Lean Core**

Những thay đổi mới bao gồm việc đặt [Webview](https://github.com/react-native-community/react-native-webview) và [NetInfo](https://github.com/react-native-community/react-native-netinfo) vào các repositories riêng biệt. Họ đã được xóa nó thành công khỏi repositories của React Native. [Geolocation](https://github.com/react-native-community/react-native-geolocation) cũng đã bị xóa, dựa trên phản hồi của cộng đồng.

Mark Zuckerberg, đã nhận xét vào năm 2012 và tôi xin trích dẫn “Lỗi sai lớn nhất mà chúng tôi đã làm trong công ty là đặt cược quá nhiều vào HTML, trái ngược với nền tảng bản địa(iOS, Android)”.

Trước tuyên bố trên, chúng ta có thể kết luận rằng framework React Native về cơ bản tốt hơn các lựa chọn khác. Bạn không cần phải tin lời chúng tôi vì bạn có thể tự mình nhìn nhận nó thông qua các sử dụng thực tế của bạn thân.

* + 1. Phiên bản 0.61

#### React Native nâng cấp việc hỗ trợ CocoaPods

Trong 0.60, một số cập nhật đã được thực hiện để tích hợp **CocoaPods**. Tuy nhiên, điều này tạo ra các vấn đề trong use\_frameworks!. Vấn đề này đã được giải quyết và hiện đã được khắc phục trong 0.61, giúp tích hợp React Native đơn giản và dễ dàng hơn vào các dự án iOS cần dùng dynamic frameworks.

#### **Sử dụng mới WindowDimensions Hook**

Tính năng mới này có tên là **Hook,** Nó tự cung cấp và đăng ký cập nhật kích thước một cách tự động. Nó cũng có thể được sử dụng thay cho API Dimensions trong hầu hết các trường hợp.

#### React hiện được nâng cấp lên 16.9

Phiên bản mới bỏ các tên cũ cho các lifecycle methods UNSAFE\_, bao gồm các cải tiến để hành động, v.v.

#### **Fast Refresh**

Tính năng mới này là sự pha trộn giữa “live reloading” and “hot reloading”.

Đây là thời điểm cao để nhóm React Native giải quyết các vấn đề của “hot reload”. Và bây giờ họ có, với **Fast Refresh**. Đây là những gì Fast Refresh cung cấp cho các developer.

* Tính năng **Fast Refresh** hoàn toàn hỗ trợ React hiện đại cùng với các thành phần chức năng và **Hook**.
* Nó khéo léo phục hồi từ các lỗi như lỗi chính tả, v.v. Hơn nữa, Fast Refresh cũng quay lại và tải lại đầy đủ khi cần. **Fast Refresh** không thực hành chuyển đổi code, Nó đáng tin cậy, vì đáng tin cậy nên được bật theo cài đặt mặc định.

Các developer đã bắt đầu đồng ý sau một số thử nghiệm rằng Fast Refresh có độ tin cậy cao cùng với tính linh hoạt. Thật không dễ dàng và đơn giản để đặt React ở vị trí mà bạn cần tải lại thủ công. Fast Refresh hoạt động hoàn hảo cho cả các thành phần lớp và các thành phần chức năng và một số thậm chí tin rằng nó có thể cập nhật nhanh hơn.

**Nó thực sự giống như “Hot Reload” – sự khác biệt duy nhất là lần này, nó được thực hiện một cách xuất sắc.**

* + 1. Phiên bản 0.62

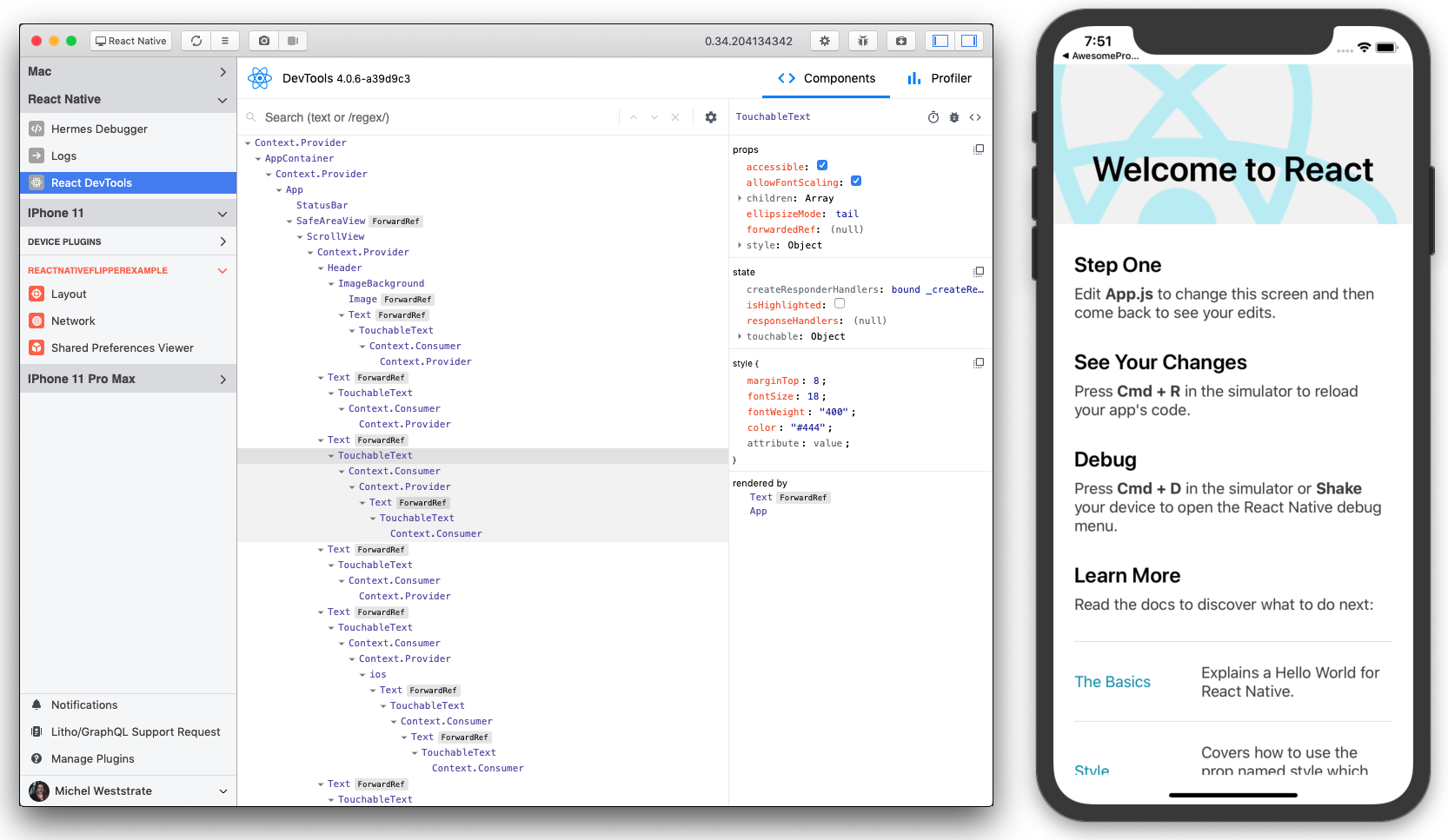
Vào ngày 26 tháng 3 năm 2020, phía Facebook đã phát hành phiên bản React Native 0.62 với một số thay đổi, trong đó bao gồm hỗ trợ **Flipper** như một tool debugging mặc định.

Phiên bản này đi kèm vs đại dịch Covid toàn cầu hiện tại. Phía Facebook cho biết họ phát hành phiên bản này nhằm thể hiện tôn trọng đối với sự hỗ trợ của hàng trăm contributors trên github, bên cạnh đó cũng ngăn chặn việc những bản release kế tiếp quá rời xa với phiên bản chính thức trên nhánh master. Lưu ý rằng việc giảm bớt các contributors nhằm giải quyết các vấn đề hiện tại và trì hoãn nâng cấp nếu cần thiết.

**Đặt Flipper làm trình debugging mặc định**

Flipper là một trình debugging cho các ứng dụng di động. Nó khá nổi tiếng trong cộng đồng lập trình viên Android và iOS.

Trong bản release này, Facebook đã đem chúng thành trình debugging mặc định cho các dự án React Native mới hoặc đang có sẵn.



**Flipper cung cấp các tính năng mới mẻ sau đây:**

* **Metro Actions**: cho phép tải lại ứng dụng và kích hoạt Dev Menu ngay trên thanh công cụ.
* **Crash Reporter**: xem báo cáo sự cố trực tiếp trên thiết bị Android hoặc iOS.
* **React DevTools**: sử dụng phiên bản mới nhất của React DevTools ngay bên cạnh các công cụ khác.
* **Network Inspector**: kiểm tra tất cả các kết nối mạng (network requests) được thực hiện bởi thiết bị.
* **Metro and Device Logs**: xem, tìm kiếm và lọc nhật ký trên cả Metro và thiết bị.
* **Native Layout Inspector**: xem và chỉnh sửa các thành phần Native layout được xuất ra bởi React Native Renderer.
* **Database and Preference Inspectors**: xem và chỉnh sửa cơ sở dữ liệu cũng như các tùy chỉnh của thiết bị.

Ngoài ra, do Flipper là một nền tảng có thể mở rộng, nó cung cấp một Marketplace lấy các plugin từ NPM giúp chúng ta dễ dàng cài đặt và sử dụng (thậm chí là tự bạn có thể publish và cài đặt các plugin của chính mình cho phù hợp quy trình công việc).

**Kiểm tra Dark Mode**

Module Appearance vừa được thêm vào nhằm cung cấp truy cập tới các tùy chọn **Appearance preferences** của người dùng, chẳng hạn như bảng màu ưa thích của họ (light hoặc dark).

const colorScheme = Appearance.getColorScheme();

if (colorScheme === 'dark') {

// Use dark color scheme

}

Ngoài ra một hook mới đã được thêm vào để đăng ký cập nhật trạng thái tới **Users preferences**.

import { Text, useColorScheme } from 'react-native';

const MyComponent = () => {

const colorScheme = useColorScheme();

return useColorScheme(): {colorScheme};

};

**Chuyển Apple TV sang react-native-tvos**

Là một phần trong nỗ lực đơn giản hóa Core React Native của Facebook và đưa Apple TV phù hợp hóa với các nền tảng khác như React Native Windows và React Native macOS, họ đã bắt đầu loại bỏ các code cụ thể cho Apple TV ra khỏi React Native Core.

Trong tương lai, phiên bản hỗ trợ Apple TV của React Native sẽ được duy trì trên react-native-community/react-native-tvos cùng với package NPM react-native-tvos tương ứng.

Các phiên bản của **react-native-tvos** sẽ dựa trên bản phát hành công khai của React Native. Với phiên bản **0.62** hiện tại của react-native, bạn cần cập nhật lại package.json của mình, sử dụng react-native-tvos phiên bản **0.62**.

**Hỗ trợ nâng cấp**

Khi version 0.61 được phát hành, cộng đồng đã giới thiệu một [công cụ hỗ trợ nâng cấp](https://react-native-community.github.io/upgrade-helper/) mới nhằm giúp các nhà phát triển nâng cấp ứng dụng hiện tại lên các phiên bản mới hơn của React Native. Công cụ trợ giúp này cung cấp một loạt các thay đổi từ phiên bản hiện tại của bạn tới phiên bản bạn muốn nâng cấp, cho phép bạn xem các thay đổi cụ thể cần thực hiện trong quá trình nâng cấp.

Tuy nhiên, ngay cả có sự hỗ trợ từ công cụ này, các vấn đề vẫn luôn luôn có khả năng phát sinh. Do đó, Facebook giới thiệu hỗ trợ nâng cấp chuyên dụng hơn bằng thông báo [Upgrade-Support](https://github.com/react-native-community/upgrade-support). Upgrade-Support là một trình **Github Issue Tracker** nơi người dùng có thể submit các vấn đề xảy ra trong quá trình nâng cấp dự án của họ và nhận được sự hỗ trợ từ cộng đồng.

Facebook cho biết họ luôn nỗ lực để cải thiện trải nghiệm trong quá trình nâng cấp ứng dụng. Họ cũng hi vọng rằng công cụ này có thể cung cấp sự hỗ trợ khi cần cho các nhà phát triển cho các trường hợp Facebook chưa hỗ trợ.

**Những cải tiến khác**

* **LogBox**: cải thiện trải nghiệm cho các lỗi và cảnh báo dưới dạng tùy chọn. Để kích hoạt nó, bạn cần thêm require('react-native').unstable\_enableLogBox() vào index.js.
* **React DevTools v4**: thay đổi này bao gồm việc nâng cấp lên React Devtools mới nhất mang lại hiệu suất tăng đáng kể, trải nghiệm điều hướng được cải thiện và hỗ trợ đầy đủ cho React Hook.
* **Khả năng truy cập**: cải thiện khả năng truy cập bằng cách thêm các **accessibilityValue**, các **props** bị thiếu trên Touchables, sự kiện onSlidingComplete, thay đổi role mặc định của Switch từ "**button**" sang "**switch**".

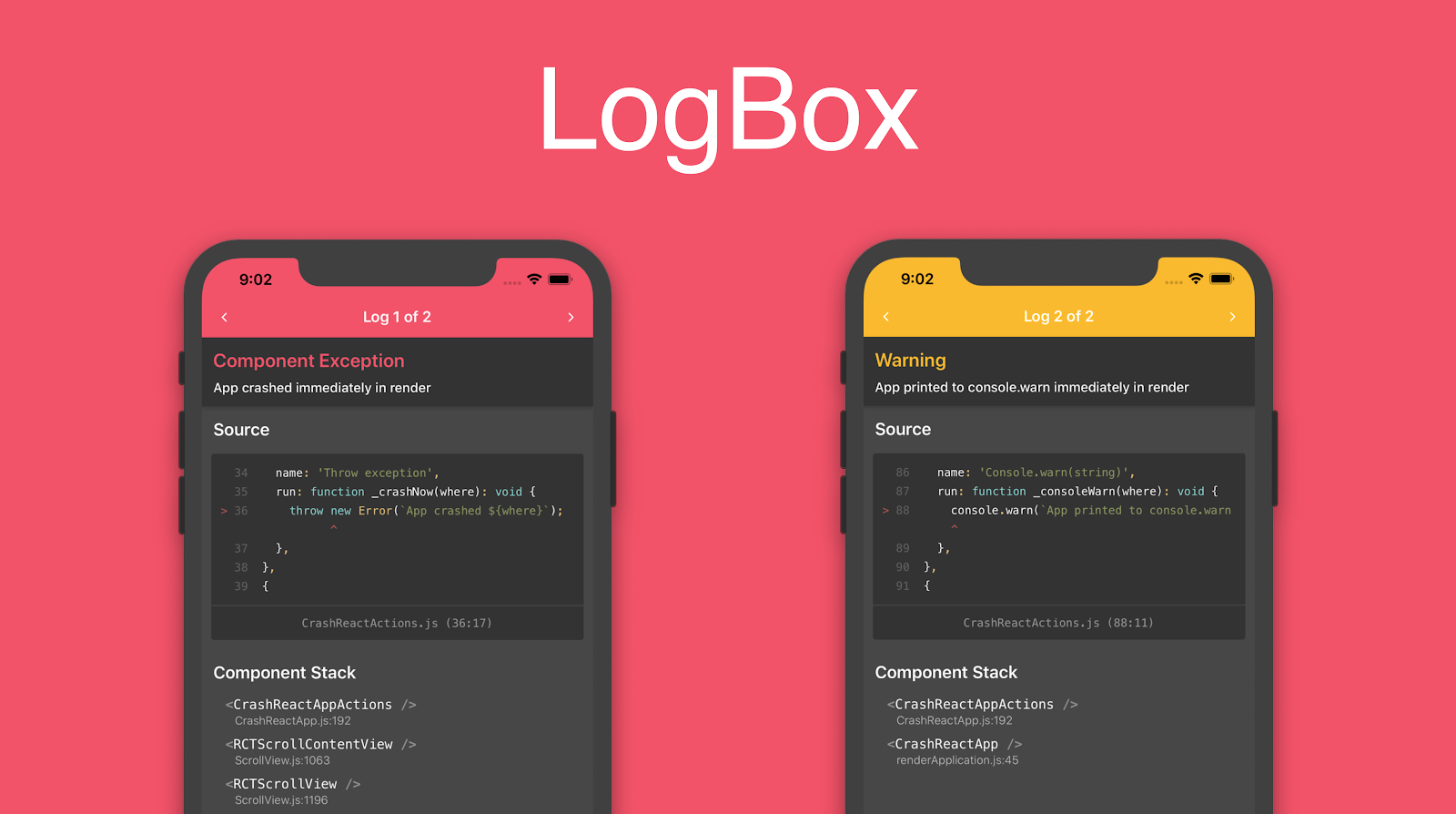
**Các Breaking Changes**

* Xóa **PropTypes**: propTypes đã bị loại bỏ ra khỏi React Native Core nhằm giảm kích thước của phần core, các hệ thống **static type systems** sẽ ưu tiên kiểm tra tại thời điểm compile ứng dụng thay vì runtime.
* Xóa accessibilityStates: thuộc tính đã bị deprecated accessibilityStates và thay bằng thuộc tính accessibilityState mới - mà theo Facebook, nó sẽ phong phú hơn về mặt ngữ nghĩa để các **component** mô tả thông tin **state** của chúng với các dịch vụ trợ năng (**accessibility services**).
* Thay đổi TextInput: prop onTextInput trên component TextInput đã bị xóa bởi chúng không được sử dụng phổ biến, không tuân thủ chuẩn **W3C** cũng như khó triển khai trên **Fabric**. Họ cũng xóa bỏ một số prop chưa được thể hiện trên tài liệu chính thức như inputView hay selectionState.

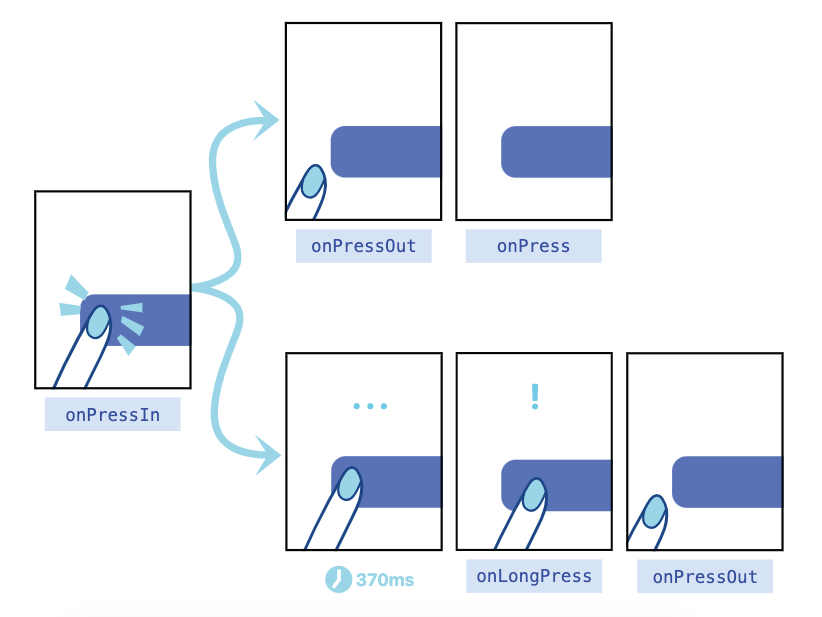
**Các phần bị deprecated**

* AccessibilityInfo.fetch đã bị deprecated, tuy nhiên ở phiên bản này họ chỉ thêm một cảnh báo.
* Thiết lập useNativeDriver hiện giờ là cần thiết để hỗ trợ chuyển đổi mặc định trong tương lai.
* ref của component Animated đã trở thành thành phần nội bộ và getNode đã bị deprecated.
  + 1. Phiên bản 0.63

Mặc dù Logbox đã ra mắt ở phiên bản 0.62 nhưng đến 0.63 nó đã chính thức trở thành trình báo lỗi mặc định của React Native. Logbox đã được thiết kế để đọc lỗi dễ hơn. Chức năng này có giao diện trực quan để thông báo những lỗi cú pháp.



Điểm nổi bật tiếp theo của version này là sự ra đời của Pressable component cho phép chúng ta dễ dàng tùy chỉnh phản hồi chạm theo từng giai đoạn để tối đa hóa trải nghiệm người dùng ở các nền tảng/thiết bị khác nhau.



Version này cũng ra mắt API PlatformColor và DynamicColorIOS để dễ dàng truy cập vào hệ thống màu của thiết bị và lấy ra giá trị có độ chính xác cao hơn đó, đừng quên sử dụng nó cho chế độ dark mode.

* 1. Các ưu và nhược điểm của React Native
     1. Ưu điểm của React Native
* Có khả năng tối ưu thời gian hiệu quả.
* Có hiệu năng ổn định.
* Có khả năng tiết kiệm chi phí.
* React Native sở hữu đội ngũ phát triển ứng dụng không quá lớn.
* Ứng dụng được xây dựng bởi React Native đều ổn định và đáng tin cậy.
* Việc xây dựng các ứng dụng ít Native Code nhất cho nhiều hệ điều hành khác nhau.
* Hầu hết, các trải nghiệm của người dùng đều sẽ tốt hơn nếu như được so sánh với các ứng dụng Hybrid.
  + 1. Nhược điểm của React Native
* Nó yêu cầu Native code.
* Hiệu năng làm việc kém hơn so với Native App.
* Khả năng bảo mật của app chưa tốt do sử dụng JS.
* Khả năng quản lý bộ nhớ chưa tốt.
* Khả năng tùy biến chưa thực sự là quá tốt với một số module.

# Xây dụng ứng dụng di động sử dụng framework React Native:

## Khảo sát các ứng dụng liên quan

Ý tưởng tạo ứng dụng chỉ sử dụng một kiểu mẫu cho tất cả các nền tảng có vẻ không thực tế chút nào. Tuy nhiên, React Native, mặc dù chưa thật sự hoàn thiện, nhưng cho phép đẩy nhanh quá trình xây dựng các ứng dụng trên các nền tảng khác nhau, nhờ khả năng sử dụng lại hầu hết các code giữa chúng. Không có gì ngạc nhiên khi nhiều công ty sử dụng React Native cho các ứng dụng trên thiết bị di động của họ. Dưới đây là danh sách những công ty đã tận dụng framework này trong sản xuất.



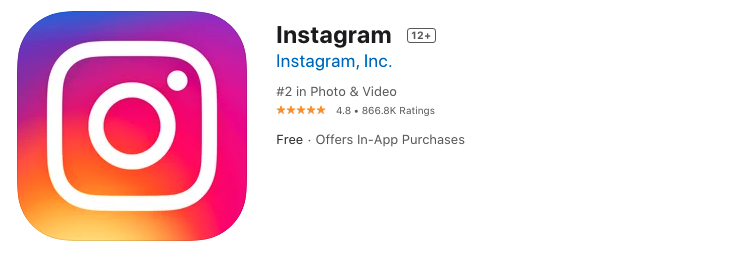
**Mạng xã hội Facebook**

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.



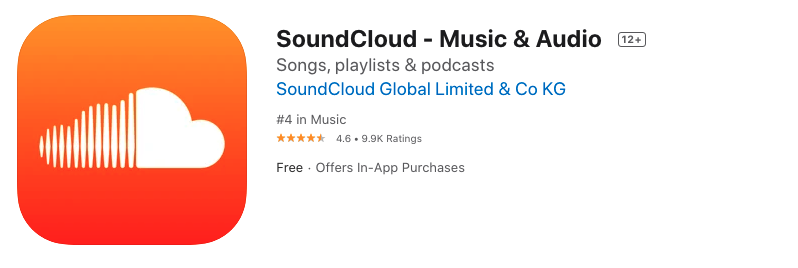
**Skype**

Skype gần đây đã thông báo rằng họ đang thử nghiệm một ứng dụng Android mới được viết bằng React Native. Đó là một thông tin khá tốt cho tất cả người dùng, vì ứng dụng hiện tại gặp nhiều vấn đề. Phiên bản mới hoàn toàn được cải tiến bắt đầu từ các icon cho toàn bộ bố cục, thêm một vài tính năng gọn gàng nữa



**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.



**SoundCloud**

SoundCloud là một ứng dụng dành cho người sáng tạo giúp họ quản lý tài khoản và giữ cho cộng đồng của họ sôi động. Khi công ty bắt đầu thiết kế bộ ứng dụng gốc thứ hai, họ phải đối mặt với một vài trở ngại. Không thể tìm thấy các lập trình viên iOS và họ không muốn có một khoảng cách lớn giữa bản phát hành iOS và Android. Do đó, một nhóm nghiên cứu độc lập đã bắt đầu chạy các phiên thử nghiệm người dùng bằng các nguyên mẫu dựa trên React Native.

## Quy trình phát triển ứng dụng:



### Mô tả phần mềm

Đến với đề tài này, nhóm chúng em vận dụng framework React Native để tạo ra một ứng dụng “Covid Info” ***có chức năng cập nhật thông tin về tình hình đại dịch COVID*,** bao gồm thông tin về số ca mắc bệnh, số ca phục hồi, số ca tử vong ở những khu vực và quốc gia khác nhau.

Bên cạnh đó, ứng dụng còn hổ trợ:

* Cập nhật thông tin về phòng chống, làm sao tự bảo vệ bản thân tránh lây nhiễm conronavirus.
* Cập nhật thông tin về số ca lây nhiễm, hồi phục, chết trên toàn thế giới tính đến thời điểm hiện tại.
* Tìm kiếm thông tin ca bệnh tại những khu vực khác nhau.
* Cập nhật tin tức mới nhất về tình hình dịch bệnh. Dữ liệu được cập nhật liên tục từ các nguồn dữ liệu tin cậy.



### Phân tích thiết kế hệ thống

#### Sơ đồ Use Case

Diagram, schematic

Description automatically generated

#### Danh sách các Actor

|  |  |  |
| --- | --- | --- |
| STT | Tên | Mô tả |
| 1 | Người dùng | Là người thao tác trực tiếp với thiết bị, chủ thiết bị di động |

#### Danh sách các Use Case

|  |  |  |
| --- | --- | --- |
| STT | Tên | Mô tả |
| 1 | Lastest News | Hiển thị các tin tức mới nhất về COVID 19 |
| 2 | Daywise Cases (VN) | Hiển thị các trường hợp:  - Đang điều trị - Tổng số ca nhiễm - Phục hồi - Tử vong |
| 3 | Safety Tips | Gợi ý các tips về bảo vệ bản thân, cách phòng tránh COVID 19 hiệu quả |
| 4 | Important Links | Các đường dẫn dẫn tới các website quan trọng về COVID 19 |
| 5 | Self Check-Ups | Tự kiểm tra sức khỏe (Dẫn tới website của WHO về cách tự kiểm tra sức khỏe) |
| 6 | World Covid Situation | Hiển thị tình hình COVID 19 trên Thế giới |

### Thiết kế giao diện hệ thống



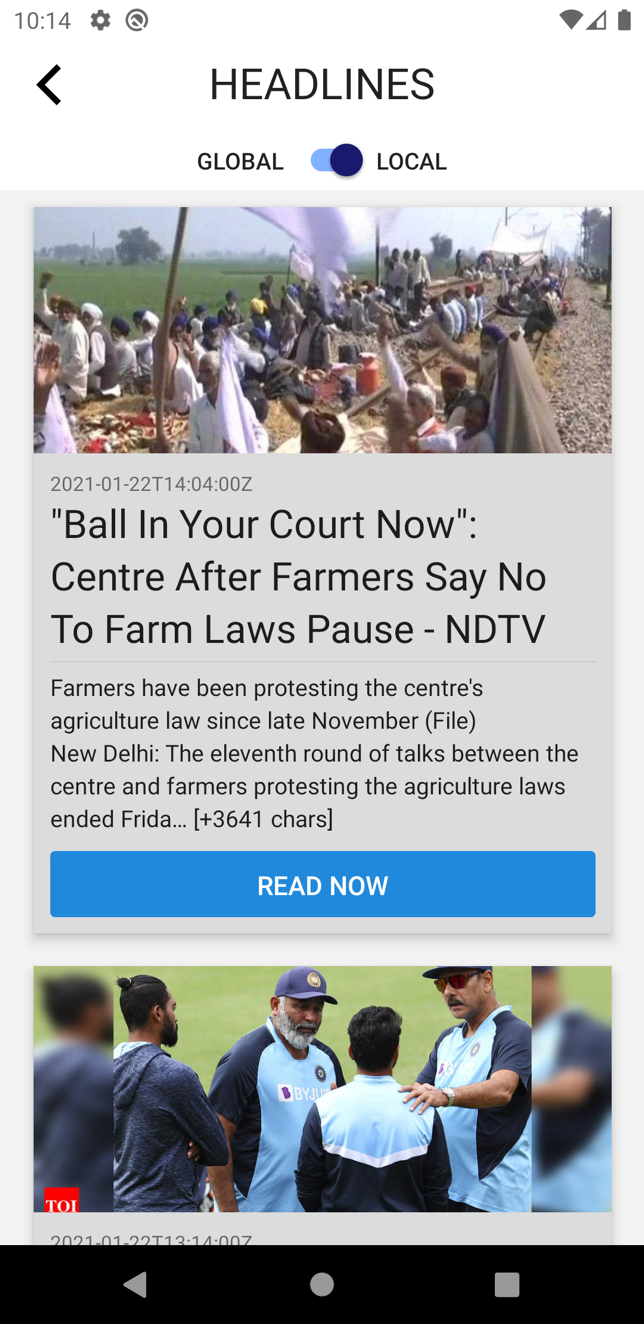
#### Giao diện màn hình HOME SCREEN



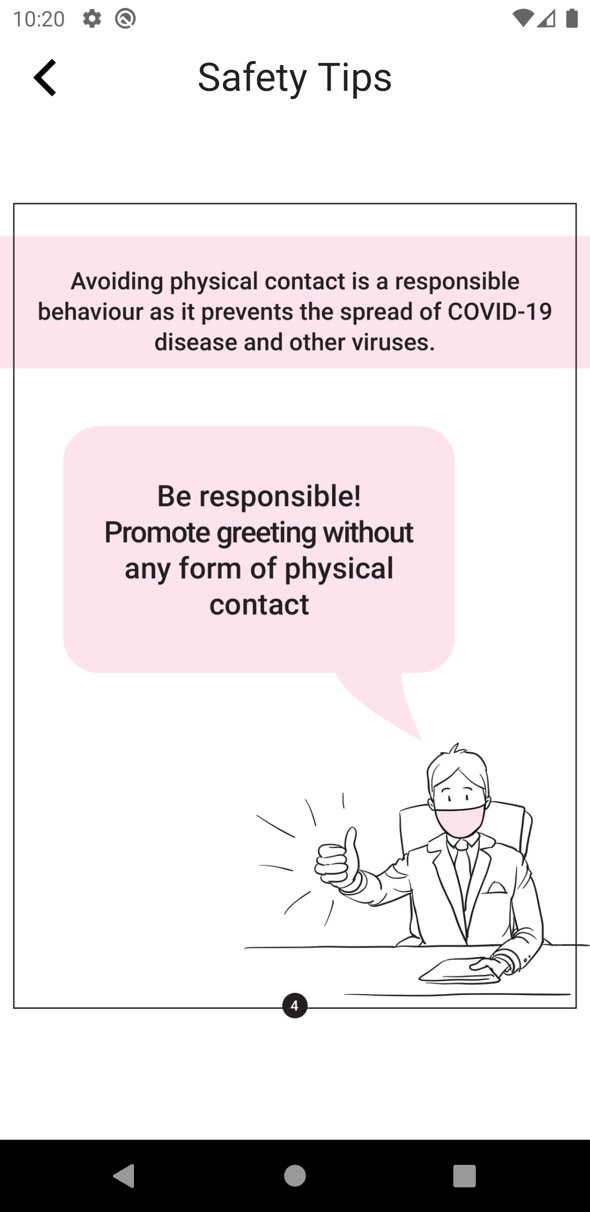
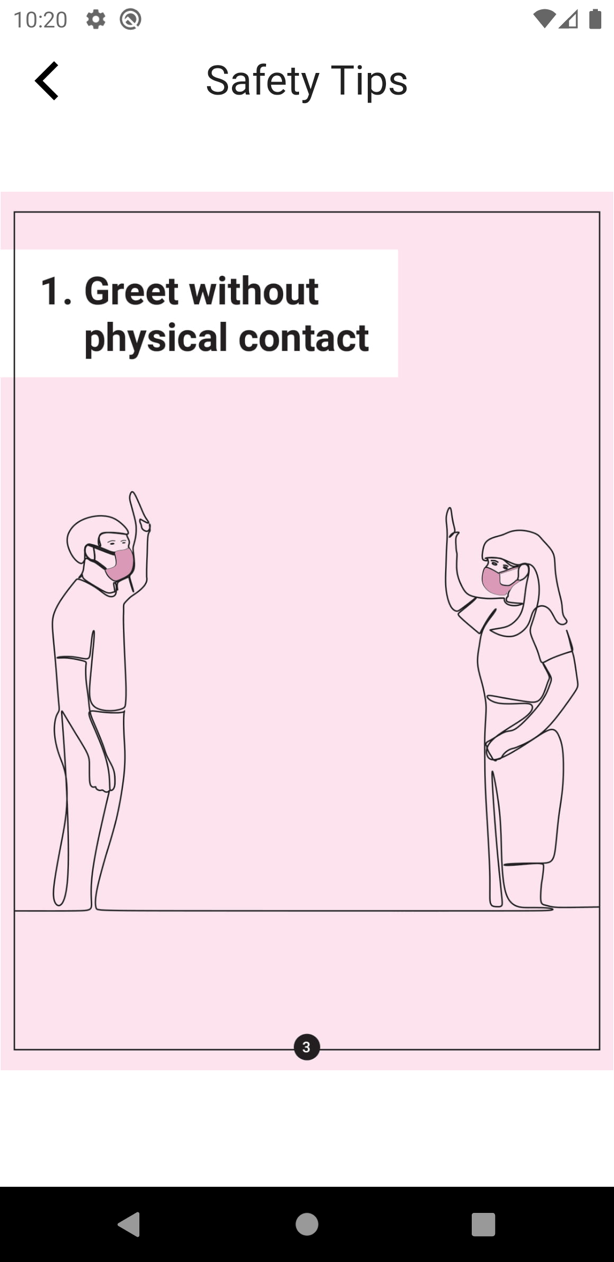
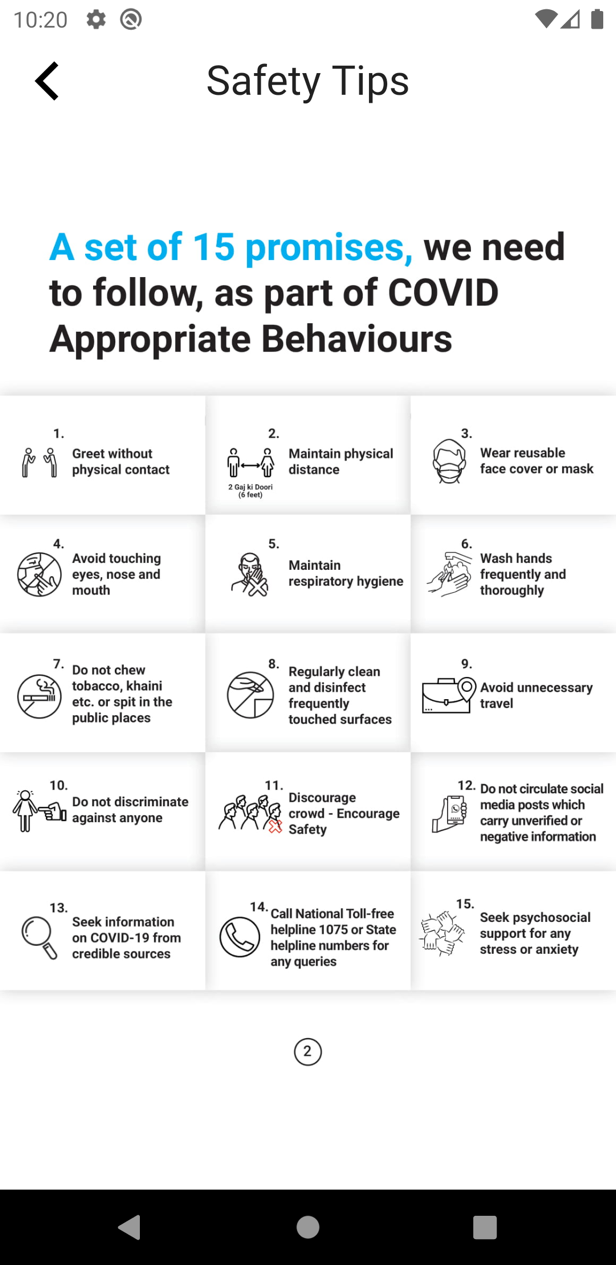
Đây là giao diện home, với các chức năng người dùng có thể tùy chọn

* “Latest News” – Xem tin tức mưới nhất về tình hình dịch covid.
* “Safety Tips” – Xem các mẹo về an toàn phòng chống Covid.
* “Daywise Cases” – Thông số ca bệnh cập nhật theo ngày.
* “Important Links” – Hiển thị các liên kết quan trọng của bộ y tế giúp cập nhật thông tin về dịch covid.
* Hiển thị số ca mắc bệnh, ca tử vong và số đã hồi phục trên toàn cầu.

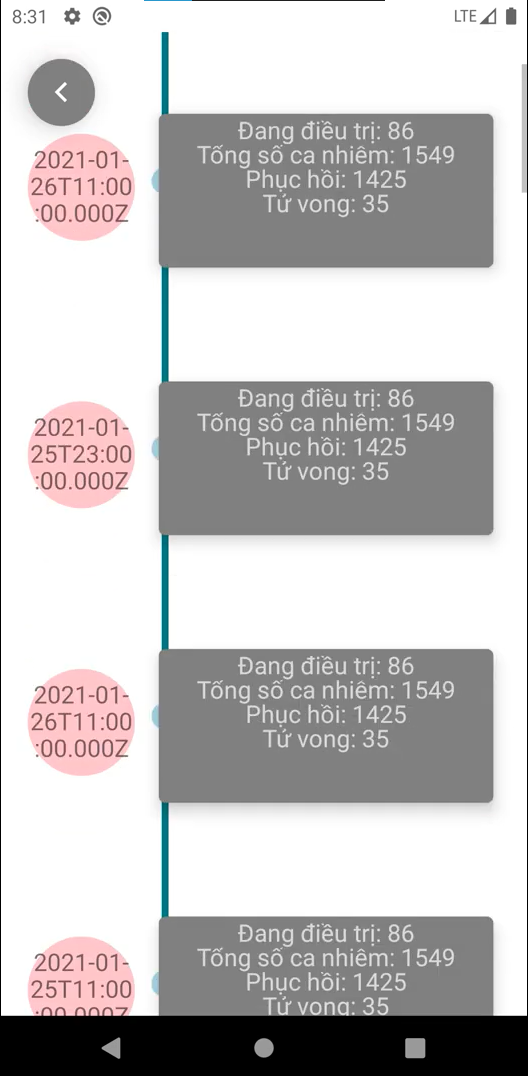
#### Giao chức năng “Latest News” – Xem tin tức mới nhất về Covid



#### Giao diện chức năng ‘’Safety Tips’’ – Những biện pháp phòng tránh Covid



#### Giao diện chức năng “Daywise Cases” – Thông số ca bệnh theo ngày



#### Giao diện chức năng “Important Links” – Các liên kết quan trọng





## Lợi ích của framework React Native đối với ứng dụng

Với việc sử dụng framework React Native trong việc phát triển ứng dụng Covid Info đã giúp tạo ra một mobile apps thực sự và thú vị với sự trợ giúp của JavaScript hỗ trợ cho cả nền tảng Android và iOS. Chỉ cần code một lần, các REACT Native apps luôn có sẵn cho cả nền tảng iOS và Android, giúp các nhà phát triển tiết kiệm thời gian tối đa. Được chống lưng bởi Facebook, REACT Native sở hữu một cộng đồng hỗ trợ rất lớn hiện nay. React Native được xây dựng dựa trên ReactJS đã mang đến một sự cạnh tranh rất lớn đó là AngularJS.

# Thử nghiệm và đánh giá:

## Môi trường thử nghiệm:

Nhóm thử nghiệm sản phẩm này trên hệ điều hành Android và IOS cho các thiết bị di động. Danh sách các bản thử nghiệm như sau:

|  |  |  |
| --- | --- | --- |
| STT | Tên thiết bị | Phiên bản |
| 2 | Iphone Xr | IOS 14.3 |
| 3 | Sonny Xperia XZ1 | Android 8.0 |
| 4 | Google Pixel 3A XL | Android 9.0 |

## Kết quả thử nghiệm:

* Ứng dụng tương thích tốt với các thiết bị Ios và android phiên bản từ 7.0 trở lên và có hiện tượng giật lag nếu ở máy có cấu hình và phiên bản android quá thấp
* Đây là bản thử nghiệm chạy trên thiết bị, phong sử dụng điện toán đám mây
* Ứng dụng xảy ra hiện tượng tràn nếu một số máy có tỉ lệ màn hình quá cũ hoặc quá mới

## Đánh giá kết quả

Về cơ bản, nhóm em tự đánh giá đã vận dụng được các kiến thức về React Native vào ứng dụng, giúp hổ trợ tốt hơn cho người sử dụng trong các thao tác hằng ngày một cánh tự động và nhanh chóng, qua đó thúc đẩy nền công nghệ phát triển và nâng cao trải nghiệm người dùng.

Tuy nhiên, do React Native vẫn còn là một nền tảng mới và đang trên đà phát triển, do đó một số tính năng hổ trợ cho ra kết quả không như mong đợi. Đó là một điều hạn chế khiến ứng dụng còn chưa đạt như mong đợi

# Kết quả đạt được và hướng phát triển:

## Kết quả đạt được

Nhóm đã ứng dụng thành công kiến thức của React Native và xây dựng thành công 1 sản phẩm riêng cho mình. Sản phẩm của nhóm về cơ bản cung cấp các tính năng tự động, cho phép người dùng cập nhật được thông tin, tiết kiệm thời gian và công sức trong xã hội hiện đại gấp rút này.

Về phía cộng đồng developer, nhóm cũng truyền tải được sự hứng thú trong việc áp dụng React Native vào dự án phát triể ứng dụng mobile, thúc đẩy quá trình hiện đại hóa của thời đại công nghệ đang phát triển. Góp phần vào sự xây dựng nền công nghiệp 4.0 và đang đi lên của nền văn minh nhân loại.

.

## Hướng phát triển

Trong tương lai, nhóm sẽ tập trung ứng dụng thêm các tính năng khác, có thể như sau:

* Cập nhật thông tin sức khoẻ của bản thân đến và vị trí của bản thân cho bộ y tế
* Phát hiện những ca mắc bệnh gần bạn
* Cảnh báo những khu vực nguy hiểm
* Và nâng cao trải nghiệm người dùng một cách thoải mái nhất và khoa học nhất

# Tài liệu tham khảo

Chúng em thao khảo tài liệu trên một số trang web sau đây:

<https://reactnative.dev/>

<https://stackoverflow.com/>

https://codelearn.io/

Xin chân thành cám ơn các tác giả đã giúp nhóm em hoàn thành đề tài này!

**----------------------- HẾT -----------------------**