**ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**



**BÁO CÁO**

**ĐỒ ÁN MÃ NGUỒN MỞ**

*Đề tài: Xây dựng ứng dụng đọc tin tức với React Native*

GVHD: **Ths Nguyễn Thị Thanh Trúc**

Sinh viên thực hiện:

Lê Việt Hoàng 16520434

Nguyễn Đăng Quang 165121560

Huỳnh Hạ Vy 16521470

TP. Hồ Chí Minh, ngày 25 tháng 12 năm 2019

# **MỤC LỤC**

[MỤC LỤC 1](#_Toc28330249)

[Lời Cảm Ơn 3](#_Toc28330250)

[NHẬN XÉT CỦA GIẢNG VIÊN 4](#_Toc28330251)

[CHƯƠNG 1: GIỚI THIỆU VỀ REACT NATIVE 5](#_Toc28330252)

[**1.1.** **Giới thiệu tổng quan** 5](#_Toc28330253)

[**1.2.** **Lịch sử ra đời** 5](#_Toc28330254)

[**1.2.1.** **Bắt đầu với React** 5](#_Toc28330255)

[**1.2.2.** **Khó khăn phát triển React App** 5](#_Toc28330256)

[**1.2.3.** **Sự cần thiết của React** 6](#_Toc28330257)

[**1.2.4.** **React Native ra đời** 7](#_Toc28330258)

[1.3. **Tìm hiểu về React Native** 7](#_Toc28330259)

[**1.3.1.** **Native** 7](#_Toc28330260)

[**1.3.2.** **Hybird** 7](#_Toc28330261)

[**1.3.3.** **React Native** 7](#_Toc28330262)

[**1.3.4.** **Cách hoạt động của React Native** 8](#_Toc28330263)

[**1.3.5.** **Ưu điểm của React Native** 8](#_Toc28330264)

[**1.3.6.** **Khuyết điểm của React Native** 8](#_Toc28330265)

[CHƯƠNG 2: CHỨC NĂNG CHÍNH 10](#_Toc28330266)

[**2.1 Cập nhật tin tức mỗi ngày, bao gồm đầy đủ các lĩnh vực** 10](#_Toc28330267)

[**2.2 Cập nhật tin tức mỗi ngày, bao gồm đầy đủ các lĩnh vực** 10](#_Toc28330268)

[**2.2 Đề xuất tin tức cho người đọc** 13](#_Toc28330269)

[**2.2 Xem tin sơ lược - xem chi tiết** 14](#_Toc28330270)

[CHƯƠNG 3: THIẾT KẾ 15](#_Toc28330271)

[**3.1 Kiến trúc chương trình** 15](#_Toc28330272)

[**3.2 Thiết kế giao diện màn hình** 16](#_Toc28330273)

[***3.3.1 Giao diện màn hình chính – Tất cả tin tức*** 16](#_Toc28330274)

[***3.3.2 Giao diện màn hình tin tức*** 17](#_Toc28330275)

[CHƯƠNG 4: CÀI ĐẶT VÀ KIỂM THỬ 18](#_Toc28330276)

[**4.1 Công nghệ sử dụng** 18](#_Toc28330277)

[**4.2 Kiểm thử** 18](#_Toc28330278)

[CHƯƠNG 5: KẾT LUẬN 19](#_Toc28330279)

[**5.1 Đánh giá chung** 19](#_Toc28330280)

[**5.2 Hướng phát triển trong tương lai** 19](#_Toc28330281)

[CHƯƠNG 6: TÀI LIỆU THAM KHẢO 20](#_Toc28330282)

# **Lời Cảm Ơn**

Đầu tiên, nhóm chúng em xin gởi lời cảm ơn chân thành đến tập thể quý Thầy Cô Trường Đại học Công nghệ thông tin – Đại học Quốc gia TP.HCM và quý Thầy Cô khoa Công Nghệ Phần Mềm đã giúp cho nhóm chúng em có những kiến thức cơ bản làm nền tảng để thực hiện đề tài này.

Trong thời gian một học kỳ thực hiện đề tài, nhóm chúng em đã vận dụng những kiến thức nền tảng đã tích lũy đồng thời kết hợp với việc học hỏi và nghiên cứu những kiến thức mới. Từ đó, nhóm chúng em vận dụng tối đa những gì đã thu thập được để hoàn thành một báo cáo đồ án tốt nhất. Tuy nhiên, trong quá trình thực hiện, nhóm chúng em không tránh khỏi những thiếu sót. Chính vì vậy, nhóm chúng em rất mong nhận được những sự góp ý từ phía các Thầy Cô nhằm hoàn thiện những kiến thức mà nhóm chúng em đã học tập và là hành trang để nhóm chúng em thực hiện tiếp các đề tài khác trong tương lai.

Nhóm em xin chân thành cảm ơn quý Thầy Cô!

# **NHẬN XÉT CỦA GIẢNG VIÊN**

………………………………………………………………………………………………..........................................................................................................................................................................................

………………………………………………………………………………………………..........................................................................................................................................................................................

………………………………………………………………………………………………..........................................................................................................................................................................................

………………………………………………………………………………………………..........................................................................................................................................................................................

………………………………………………………………………………………………..........................................................................................................................................................................................

………………………………………………………………………………………………..........................................................................................................................................................................................

………………………………………………………………………………………………..........................................................................................................................................................................................

………………………………………………………………………………………………..........................................................................................................................................................................................

………………………………………………………………………………………………..........................................................................................................................................................................................

# **CHƯƠNG 1: GIỚI THIỆU VỀ REACT NATIVE**

## **Giới thiệu tổng quan**

React Native là một framework do công ty công nghệ nổi tiếng Facebook phát triển nhằm mục đích giải quyết bài toán hiệu năng của Hybrid và bài toán chi phí khi mà phải viết nhiều loại ngôn ngữ native cho từng nền tảng di động.

Chúng ta sẽ build được ứng dụng Native, và chúng ta cũng có thể build ứng dụng đó một cách đa nền tảng (multi-platform) chứ không phải là một “mobile web app”, không phải là “HTML5 app”, và cũng không phải là một “hybrid app” hay cũng không chỉ build trên iOS hay Android mà chúng ta build và chạy được cả hai hệ sinh thái.

React Native cho phép bạn tạo ứng dụng dành cho thiết bị di động chỉ sử dụng JavaScript. Với React Native, bạn không cần phải build như mobile web app, HTML5 app hay hybrid app, mà được build trực tiếp giống như các ứng dụng được viết bằng objective-c hay java.

## **Lịch sử ra đời**

### **Bắt đầu với React**

React được facebook giới thiệu tới cộng đồng lập trình viên, từ thời điểm đó cho đến nay, React đã phát triển một cách nhanh chóng không chỉ trong nội bộ facebook mà còn cả cộng đồng bên ngoài. Hiện nay, rất nhiều dự án đã được xây dựng dựa trên React, tỷ lệ các lập trình viên chọn và sử dụng React ngày càng nhiều vì nó giúp tiết kiệm thời gian dành cho framework và tập trung hơn vào sản phẩm của mình hơn.

React tập trung vào việc chia nhỏ ứng dụng ra thành nhiều thành phần nhỏ, mỗi thành phần thể hiện một view tương ứng. Những thành phần này có thể dễ dàng được sử dụng lại trong hệ thống, khi có yêu cầu thay đổi chúng ta không cần phải thay đổi toàn bộ hệ thống nữa mà thay vào đó chỉ cần thay đổi một phần nhỏ trong đó. Quan trọng hơn React đóng gói các thẻ DOM, API chỉ cần khai báo một lần, giúp tăng tính trừu tượng và đơn giản hóa mô hình lập trình. Xây dựng app với react giúp code của chúng ta sẽ dễ hình dung hơn, dễ đoán hơn. Tính dễ đoán này giúp chúng ta có thể tự tin sử dụng lại và ứng dụng đáng tin cậy hơn. Ngoài ra, xây dựng ứng dụng với react không những giúp ứng dụng có khả năng mở rộng mà còn giúp chúng ta dễ dàng mở rộng qui mô của nhóm hơn.

Với công nghệ web và react chúng ta có thể xây dựng rất nhiều ứng dụng tuyệt vời, Ví dụ như Facebook. Tuy nhiên web chỉ là một phần của câu chuyện dài. Với những hạn chế hiện tại của web, các ứng dụng android và ios đang dần dần lên ngôi, tuy nhiên không phải ai cũng có thể xây dựng được một ứng dụng android hay ios một cách dễ dàng.

### **Khó khăn phát triển React App**

Có rất nhiều lý do khiến cho việc xây dựng một ứng dụng native trên thiết bị di động lại khó khăn hơn trên web. Khó khăn đầu tiên đó là việc sắp xếp các thành phần trên màn hình, chúng ta luôn luôn phải tính toán kích thước, vị trí của tất cả các view. Một phần khó khăn nhất là giai đoạn chuyển đổi sang mobile, thử nghĩ mà xem, tốc độ phát triển ứng dụng của lập trình viên sẽ giảm xuống như thế nào?

Để xây dựng website, khi thay đổi một đoạn code nào đó, chúng ta chỉ cần lưu file và reload lại trang web để xem kết quả. Tuy nhiên với native, chúng ta cần phải recompile sau mỗi lần thay đổi, ngay cả khi chúng ta chỉ muốn dịch text lên một vài pixel. Do đó, tốc độ hoàn thành công việc của các lập trình viên sẽ bị chậm lại, đặc biệt nếu codebase lớn, sẽ mất thời gian hơn khi re-complie. Xây dựng native app cũng làm cho việc kiểm thử các chức năng mới trở nên khó khăn hơn. Tại Facebook, khi deliver một version mới của website, chỉ mất có hai ngày để testing team có thể test và feedback lại kết quả ngay sau đó. Với mobile, thường phải mất tới một tuần hoặc một tháng để test và nhận feedback từ testing team. Như vậy ở facebook, thời gian phát triển một chức năng trên web luôn luôn nhanh hơn thời gian phát triển cùng chức năng đó trên mobile, vậy tại sao họ lại muốn chuyển đổi từ web sang app cho mất thời gian ?

### **Sự cần thiết của React**

Mặc dù phát triển ứng dụng native tốn nhiều thời gian hơn, có rất nhiều lý do lý giải tại sao chúng ta có thể mang lại cho người dùng trải nghiệm trên mobile tốt hơn trên web. Điều đầu tiên, chúng ta có thể sử dụng các thành phần đặc biệt của platform như map, date pickers, switches và navigation stacks. Chúng ta cũng có thể phát triển lại các thành phần này trên web nhưng các nổ lực xây dựng lại của chúng ta không bao giờ có thể mang lại trải nghiệm native thực sự như trên mobile được, và chúng cũng không bao giờ được cập nhật một cách tự động với những thay đổi từ platform khi cần. Chúng ta cũng không có bất cứ thứ gì hoạt động y chang như native app được công nhận trên web, và chúng ta chưa có công cụ hoặc các phương pháp nào chính xác cần thiết để xây dựng một hệ thống chuẩn xác.

Trên web, chúng ta cũng không có một mô hình thread hoàn chỉnh, vì vậy chúng ta không thể làm việc song song trên nhiều thread. Chúng ta có thể cố gắng sử dụng web workers để thực thi một vài logic ứng dụng trên background nhưng chúng ta chưa thể thực thi tính toán phức tạp như image decoding hay đo lường văn bản hiệu quả trên thread chính của browser. Đây chính là thách thức lớn nhất đổi với việc xây dựng một ứng dụng web apps có performance tốt và responsive.

Những gì Facebook thực sư muốn đạt được đó là trải nghiệm của người dùng trên ứng dụng native, kết hợp với kinh nghiệm có sẵn của các lập trình viên khi xây dựng ứng dụng web bằng react. Họ đã thử một vài cách để đạt được điều này:

* Sử dụng webviews :

Sử dụng WebViews để load ứng dựng. Facebook đã thử cách này vài năm trước, và lúc đầu họ thực sự nghĩ đó là một ý kiến tuyệt vời. Nhưng quá trình thực hiện đã không mang lại hiệu suất tốt và tính co giãn mà họ muốn. Cách này có ưu điểm là tận dụng lại được các lập trình viên có kinh nghiệm trên web để tiếp tục phát triển các ứng dụng của facebook trên android và ios. Nhưng không may là việc render đang sử dụng công nghệ web nên không thể mang lại trải nghiệm native thực sự cho người dùng.

* Scripting native
* Chuyển react sang native :

Chuyển đổi react sang native là một ý tưởng tuyệt vời của facebook và sự thực họ đã xây dựng thành công trên ios. Họ tạo ra một component gọi là ComponenKit, với nó họ đã tận dụng lại được các lợi ích của react, đồng thời sử dụng được sức mạnh của môi trường native.

Tuy nhiên có một số hạn chế với cách tiếp cận này. Đầu tiên, chỉ hỗ trợ cho IOS, vì vậy nếu muốn xây dựng ứng dụng trên android, họ phải tạo ra bản build riêng và train cho các lập trình viên cách sử dụng component đó. Ngoài ra, họ không thể sử dụng lại bất kì các stuff nào mà họ đã xây dựng cho web bằng react như relay. Quan trọng nhất vẫn là nền tảng này chưa giúp họ giảm được thời gian phát triển ứng dụng, vẫn phải recompile sau mỗi lần thay đổi.

### **React Native ra đời**

Cũng sử dụng javascript để phát triển nhưng sự khác biệt duy nhất trên môi trường mobile là thay vì react sử dụng trên browser và render ra divs và spans, trên react native, javascript được nhúng vào một instance của javascript core bên trong app và render tới platform cụ thể ở mức độ cao hơn

## **Tìm hiểu về React Native**

### **Native**

Ứng dụng native 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, Window phone là C++ hoặc C#. 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.

Tốc độ ở đây bao gồm tốc độ hiển thị (độ mượt), và tốc độ xử lý. Nếu để ý kỹ bạn sẽ thấy các thao tác của ứng dụng native như chuyển trang, điều hướng, cuộn trang đều rất mượt. Với việc hơn hẳn về tốc độ xử lý, nên ứng dụng hybrid khó có thể làm được ứng dụng nặng đồ hoạ như game, mặc dù vẫn có thể làm được nhưng không mượt mà nên hầu như hiếm ai dùng hybrid cho lập trình game cho điện thoại di động (ngoại trừ một số game đơn giản như xếp hình, giải câu đố, 2048…)

### **Hybird**

Ứng dung 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, GPS hoặc các cảm biến như con quay hồi chuyển, gia tốc kế…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. Ứng dụng gọi hàm bằng Javascript thông qua API thì chương trình Phonegap/Cordova 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.

### **React Native**

React Native là framework giúp lập trình viên viết ứng dụng Native chỉ bằng Javascript. Đúng vậy, chỉ đơn giản là Javascript, React Native phủ nhận định nghĩa về ứng dụng native ở đoạn trên.

React Native giúp cho lập trình viên web có thể viết ứng dụng native để khắc phục các điểm yếu của ứng dụng web và hybrid. Chỉ với một lập trình viên thành thạo javascript, bạn có thể chiến đấu trên mọi mặt trận web, desktop, server và bây giờ là mobile. Điều này không những có lợi cho lập trình viên web mà nó giúp cho các doanh nghiệp phát triển sản phẩm đầu cuối với ít nhân lực hơn.

Khi xây dựng React Native, điều tuyệt vời là đã được tích hợp tính năng Live Reload – tương tự tính năng Hot Replacement Module trong Webpack. Tính năng Live Reload khác tính năng Reload, trong khi Live Reload chỉ tải lại chức năng/tập tin nào thay đổi, thì Reload sẽ tải lại toàn bộ mã nguồn. Ngoài ra, bạn cũng dễ dàng debug javascript trong Chrome và Safari. Đối với những lỗi thuộc Native thì phải cần đến XCode cho iOS hoặc Android Studio cho Android.

### **Cách hoạt động của React Native**

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

Phần hiển thị được biên dịch từ javascript sẽ map với những component của hệ thống ví dụ: điều hướng, tab, touch…Phần view này được lấy cảm hứng từ Virtual DOM của React JS, mọi xử lý view 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, biểu thức này được xử lý dưới bộ core thực thi Javascript, không phải thông dịch qua Java hay Swift/Objective-C rồi mới làm phép tính.

### **Ưu điểm của React Native**

React Native là framework giúp lập trình viên viết ứng dụng Native chỉ bằng Javascript. Đúng vậy, chỉ đơn giản là Javascript, React Native phủ nhận định nghĩa về ứng dụng native ở đoạn trên.

React Native giúp cho lập trình viên web có thể viết ứng dụng native để khắc phục các điểm yếu của ứng dụng web và hybrid. Chỉ với một lập trình viên thành thạo javascript, bạn có thể chiến đấu trên mọi mặt trận web, desktop, server và bây giờ là mobile. Điều này không những có lợi cho lập trình viên web mà nó giúp cho các doanh nghiệp phát triển sản phẩm đầu cuối với ít nhân lực hơn.

Khi xây dựng React Native, điều tuyệt vời là đã được tích hợp tính năng Live Reload, Hot Reload – tương tự tính năng Hot Replacement Module trong Webpack. Tính năng Live Reload khác tính năng Reload, trong khi Live Reload chỉ tải lại chức năng/tập tin nào thay đổi, thì Reload sẽ tải lại toàn bộ mã nguồn. Ngoài ra, bạn cũng dễ dàng debug javascript trong Chrome và Safari. Đối với những lỗi thuộc Native thì phải cần đến XCode cho iOS hoặc Android Studio cho Android.

React Native có thể được sử dụng cho việc phát triển ứng dụng trên Windows với React Native Windows được hỗ trợ chính thức bởi Microsoft. Bạn có thể sử dụng React Native Windows để phát triển cho toàn bộ hệ sinh thái của Microsoft: desktop, tablet, xbox

### **Khuyết điểm của React Native**

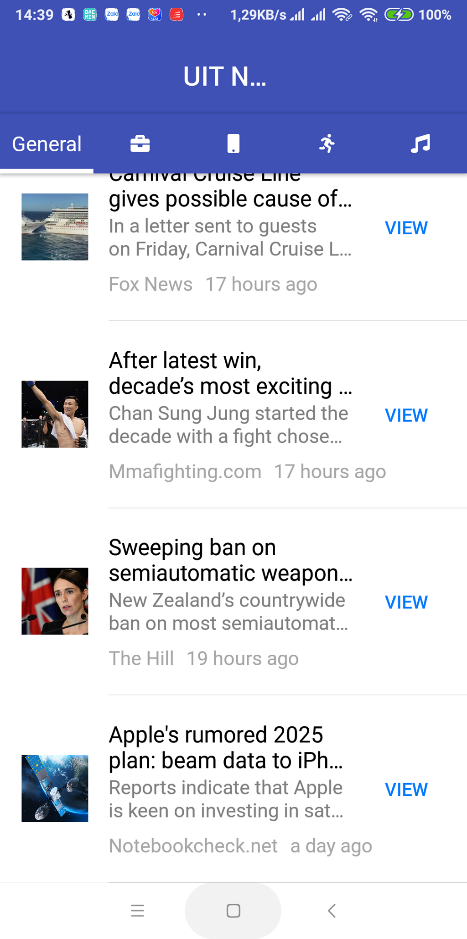
React Native là một giải pháp tuyệt vời cho phát triển ứng dụng trên điện thoại di động, tuy nhiên đến thời điểm hiện tại, vẫn còn tồn tại một số khuyết điểm:

* Vẫn còn thiếu các component view cho Android: 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 component. Bạn nên nhớ là để viết được 1 ứng dụng native bằng javascript “luôn luôn” có sẵn các component đã được viết từ Swift/Objective-C (iOS) và Java (Android) với tính chất 1-1. Trường hợp bạn muốn chỉnh sửa 1 component nào đó: 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 component 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 (Xem thêm [Flutter là gì](https://fullstackstation.com/flutter-la-gi/) nếu muốn lập trình game)
* Dùng ES2015/ES6 (bạn đọc thêm ở bài [ES2015 là gì)](https://www.businesscard.vn/blog/es6-es2015-la-gi-va-tim-hieu-nhung-tinh-nang-moi/): 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: React Js, React Native, Flux, Relay, GraphQL. Bạn không cần phải quan tâm những thứ này chỉ để viết ứng dụng với React native

# **CHƯƠNG 2: CHỨC NĂNG CHÍNH**

## **2.1 Cập nhật tin tức mỗi ngày, bao gồm đầy đủ các lĩnh vực**

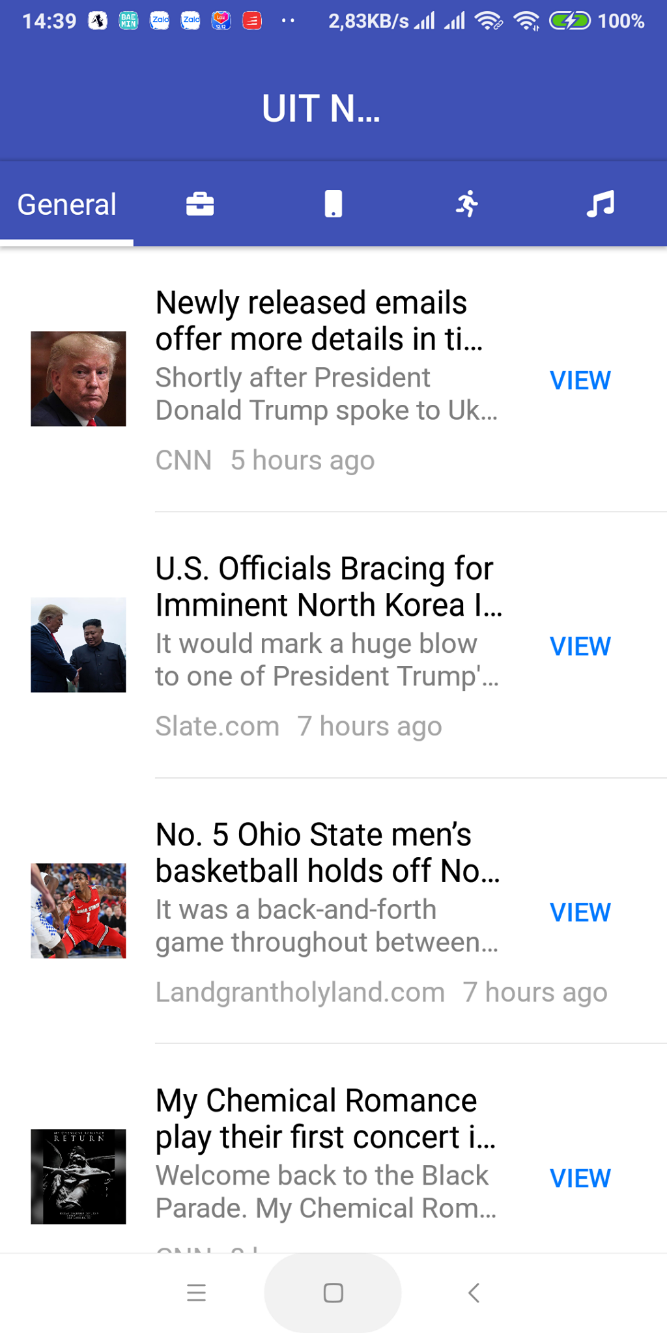
Nhóm sử dụng api từ trang web newsapi.org để lấy thông tin dưới dạng json. Ứng dụng mỗi khi khởi động sẽ gửi yêu cầu để lấy về tin tức mới nhất.



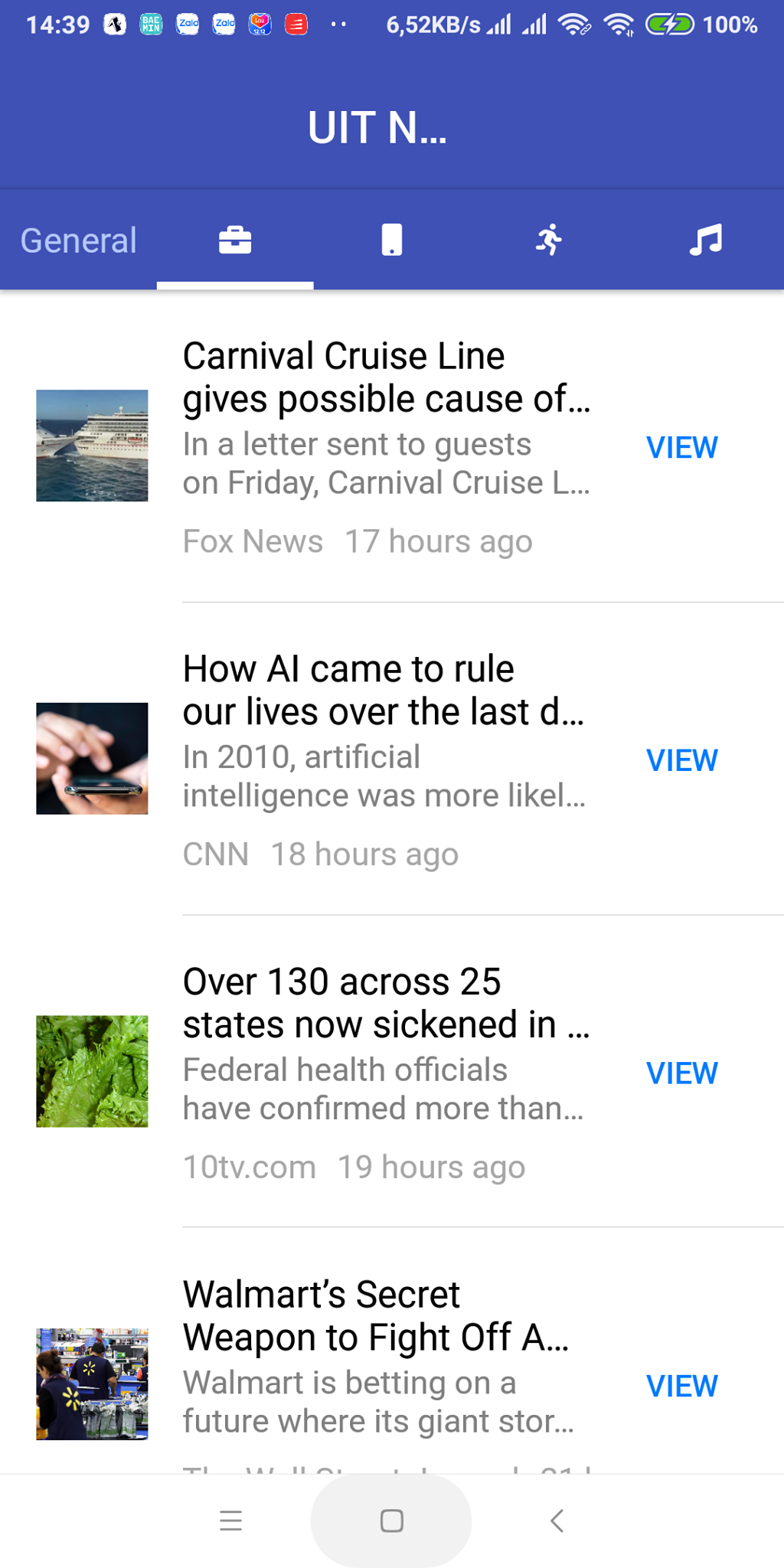
## **2.2 Phân chia tin tức cập nhật theo chuyên mục**

Ứng dụng phân chia tin tức thành từng tab theo những lĩnh vực phổ biến, giúp người đọc dễ dàng tiếp cận được với thông tin cần thiết một cách nhanh chóng.

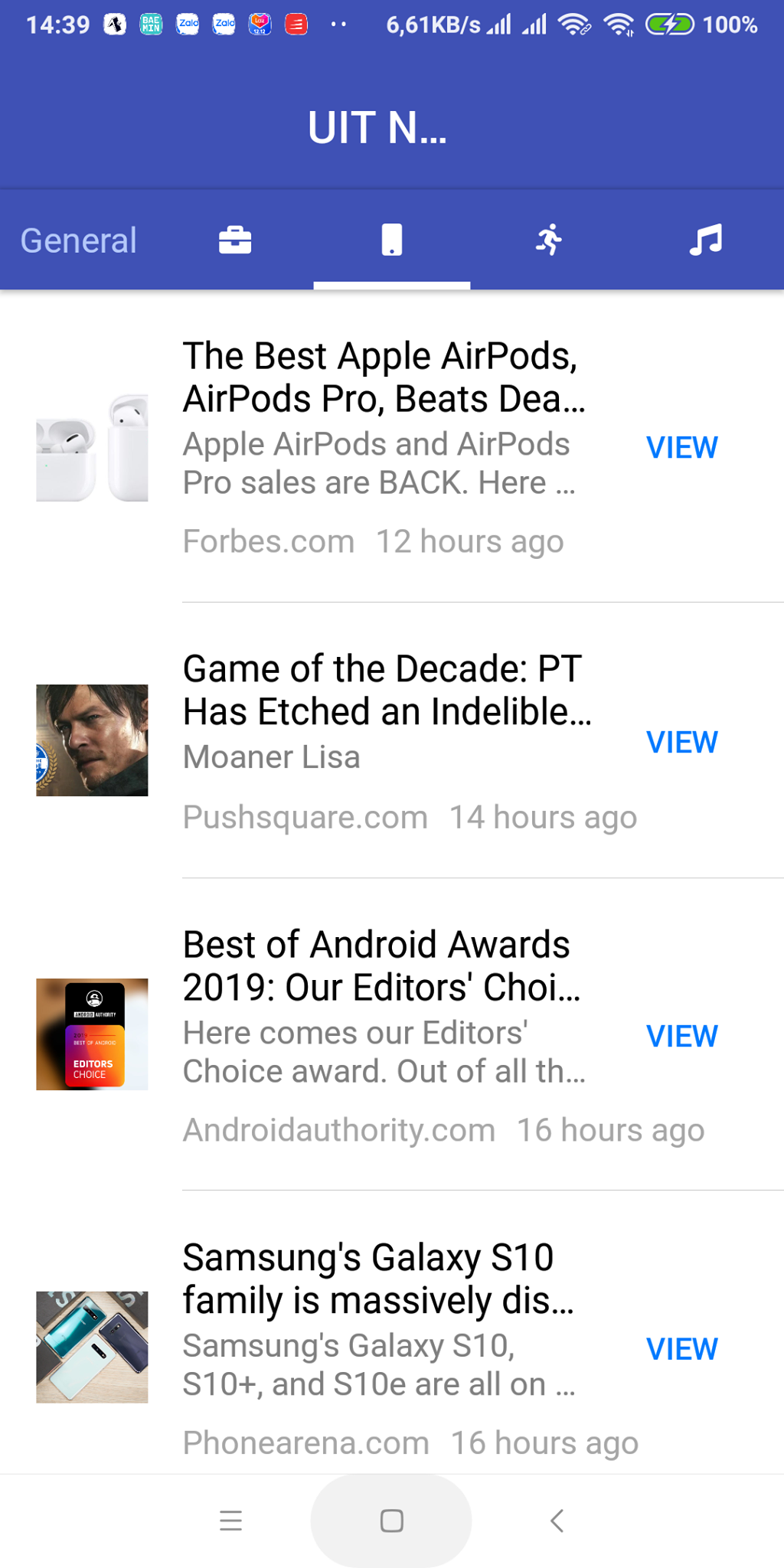
* Chuyên mục chung:



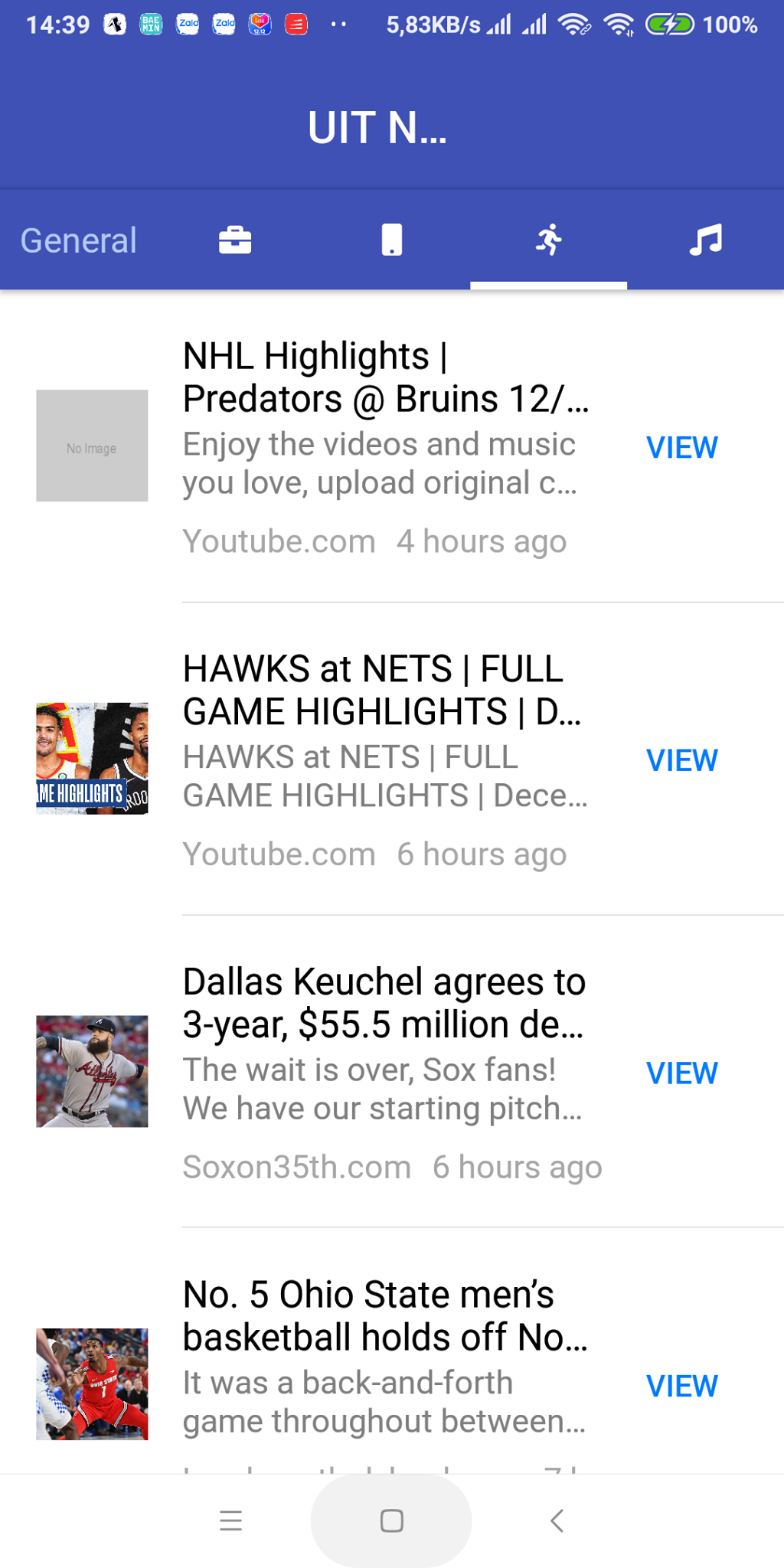
* Chuyên mục kinh doanh:



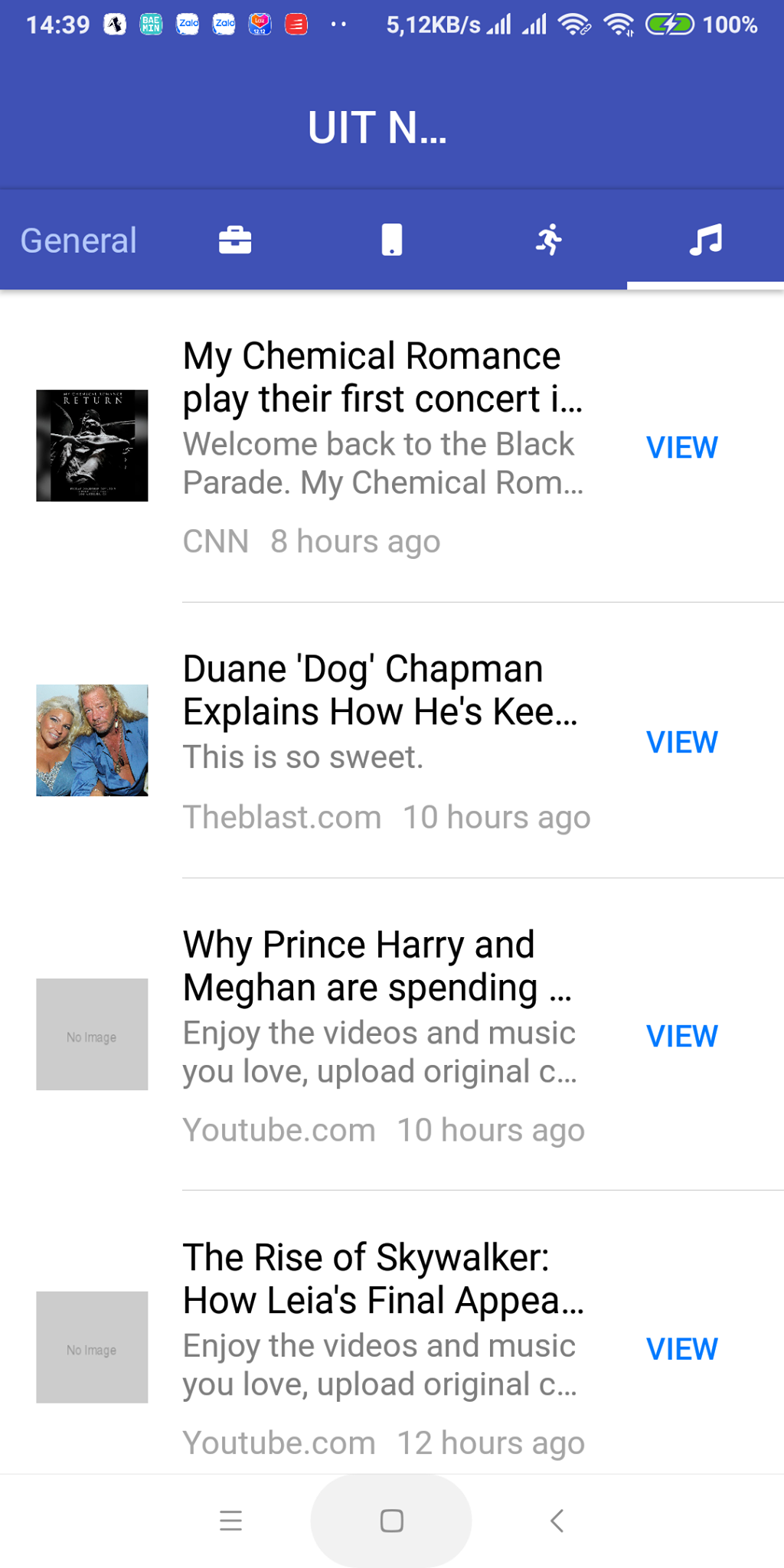
* Chuyên mục công nghệ:



* Chuyên mục thể thao:

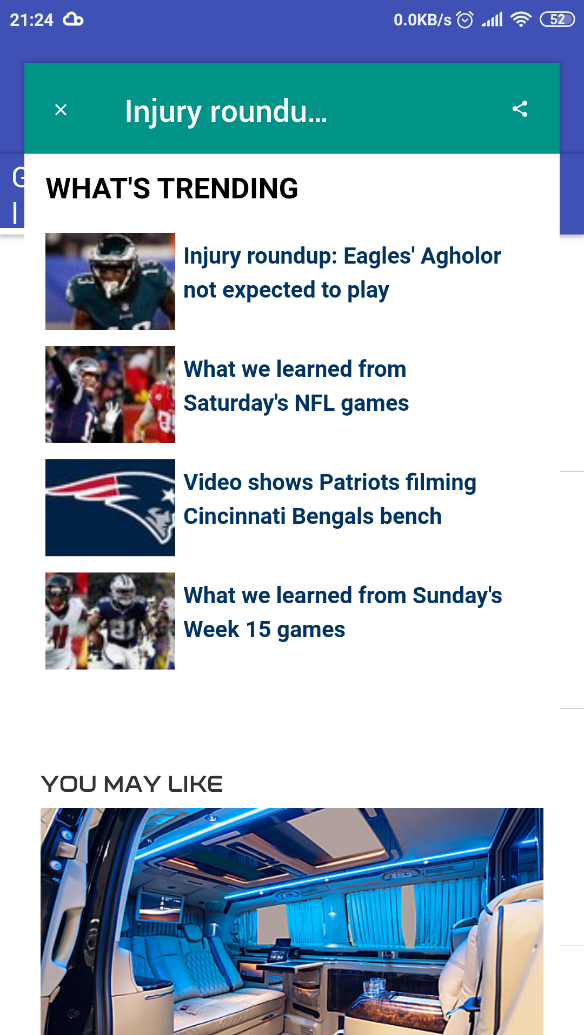


* Chuyên mục âm nhạc:



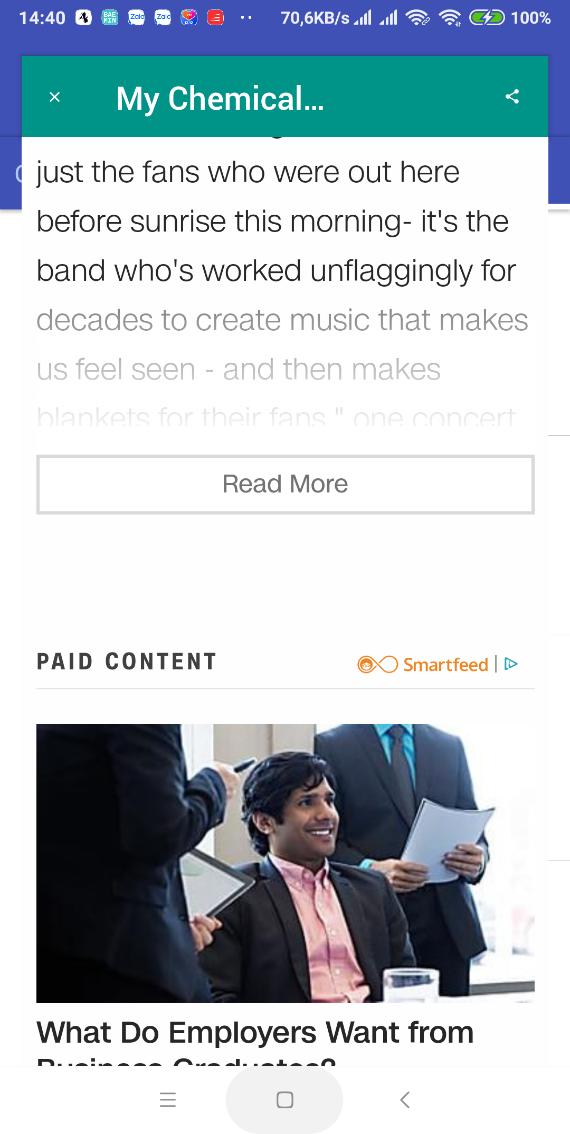
## **2.2 Đề xuất tin tức cho người đọc**

Khi đọc đến cuối trang của một tin tức nào đó, ứng dụng sẽ tự động đề xuất các tin liên quan mà người dùng có thể quan tâm, giúp người dùng tiết kiệm thời gian tìm kiếm, mang lại trải nghiệm đọc báo thuận tiện, đồng thời giữ chân người đọc ở lại ứng dụng lâu hơn.



## **2.2 Xem tin sơ lược - xem chi tiết**

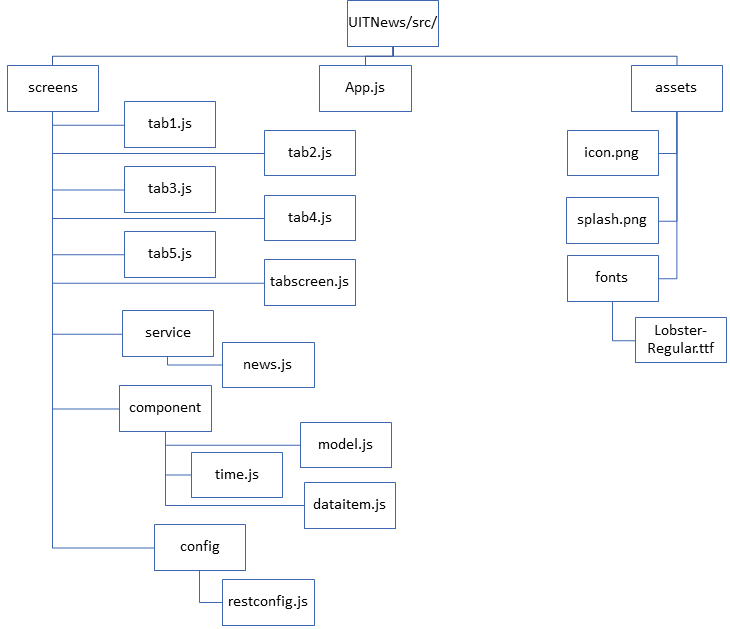
Người dùng có thể xem sơ lược tin tức khi lướt trên điện thoại. Nếu người dùng muốn xem toàn bộ mẩu tin tức đó, người dùng nhấn vào nút “Read more”, tin tức sẽ được chuyển sang trang mới với toàn bộ nội dung và các tính năng phụ khác.



# **CHƯƠNG 3: THIẾT KẾ**

## **3.1 Kiến trúc chương trình**

Danh sách các component:



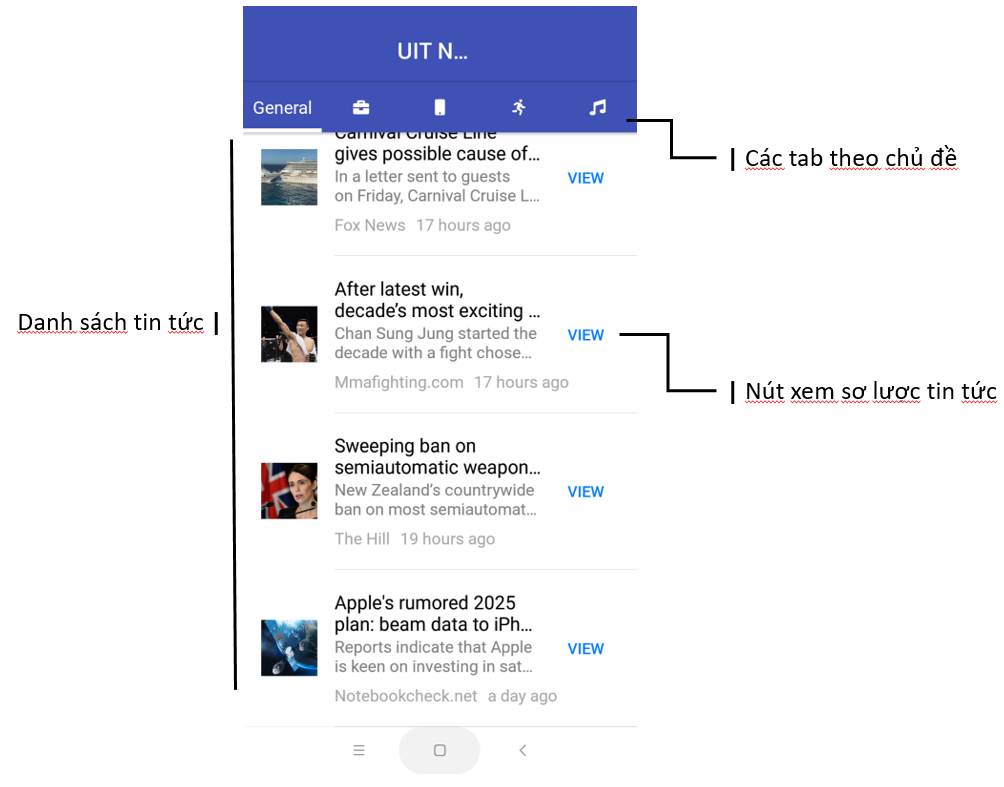
- dataitem.js: hiển thị danh sách tin tức

- model.js: khung thông tin pop-up hiển thị nội dung chính của mẩu tin để người dùng năm được thông tin sơ lược.

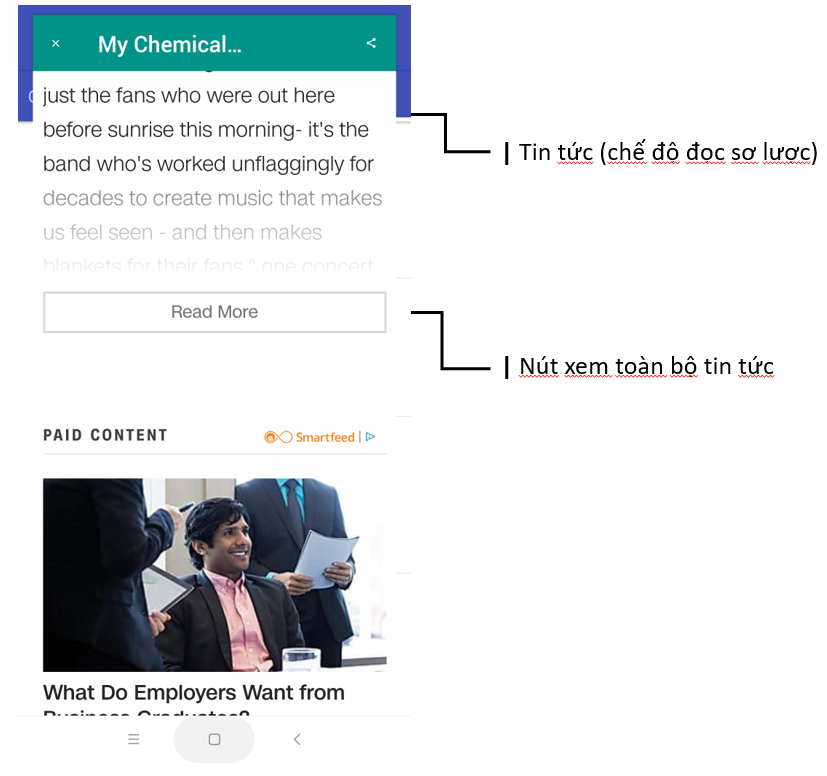
- time.js: hiển thị thời gian mẩu tin được đăng lên lần đầu tiên.

## **3.2 Thiết kế giao diện màn hình**

### ***3.3.1 Giao diện màn hình chính – Tất cả tin tức***



### ***3.3.2 Giao diện màn hình tin tức***



# **CHƯƠNG 4: CÀI ĐẶT VÀ KIỂM THỬ**

## **4.1 Công nghệ sử dụng**

* Môi trường lập trình: Visual Studio Code
* Ngôn ngữ lập trình: Javascript
* Thư viện: React Native
* Yêu cầu hệ thống để cài đặt phần mềm:
* Phần cứng: Thiết bị di động có bộ nhớ trong còn trống tối thiểu là 100 MB, RAM tối thiểu là 512 MB
* Phần mềm: Thiết bị di động chạy trên nền hệ điều hành Android 5.0 hoặc mới hơn.

## **4.2 Kiểm thử**

|  |  |  |
| --- | --- | --- |
| **STT** | **Vấn đề** | **Khắc phục/ hoàn thành** |
| 1 | Tốc độ tải về tin tức chậm | 100% |
| 2 | Danh sách các tin được đề xuất không liên quan với chủ đề của tin đang được đọc | 100% |
| 3 | Có mục bình luận dành cho người dùng | Chưa hoàn thành |
| 4 | Chức năng chia sẻ thông tin lên Facebook không hoạt động | 100% |
| 5 | Nút bấm để xem tin tức không có hiệu ứng được nhấn vào | Chưa hoàn thành |

# **CHƯƠNG 5: KẾT LUẬN**

## **5.1 Đánh giá chung**

Vì tập trung tối đa vào trải nghiệm đọc báo của người dùng nên phần mềm chỉ bao gồm những tính năng thiết yếu cho việc đọc báo, không có các tính năng nâng cao hoặc mở rộng.

Tuy nhiên, do thời gian có hạn và kĩ năng sử dụng thư viện React Native của các thành viên trong nhóm chưa thành thạo nên ứng dụng vẫn còn thiếu một số chức năng cần thiết như tìm kiếm tin theo từ khóa, cho phép người dùng bình luận tại mỗi bài báo, …

Nhìn chung, ứng dụng có thể đáp ứng cơ bản nhu cầu cập nhật tin tức của người dùng, giao diện thiết kế đơn giản, dễ dùng nhưng vẫn hiện đại và đẹp mắt là những điểm cộng cho UITNews.

## **5.2 Hướng phát triển trong tương lai**

Nhóm sẽ tiếp thu ý kiến đóng góp từ người dùng để tiếp tục hoàn thiện các chức năng hiện có của ứng dụng và bổ sung những chức năng cần thiết còn thiếu.

Bên cạnh đó, giao diện ứng dụng cũng sẽ được thiết kế lại nhằm tuân thủ đúng triết lí thiết kế Material Design và cho phép người dùng tùy biến giao diện theo sở thích cá nhân.

Cuối cùng, sau khi đã hoàn thiện, nhóm sẽ đưa ứng dụng lên kho ứng dụng CH Play của Android để mọi người có thể tải về và sử dụng rộng rãi.

# **CHƯƠNG 6: TÀI LIỆU THAM KHẢO**

1. <https://github.com/FredJul/Flym>
2. <https://newsapi.org/>
3. <https://code.tutsplus.com/tutorials/how-to-create-a-news-reader-with-react-native-setup-and-news-item-component--cms-25935>
4. <https://github.com/jondot/awesome-react-native>
5. <https://code.tutsplus.com/tutorials/beautiful-material-design-apps-with-the-antiqueruby-react-native-template--cms-33796>
6. <https://facebook.github.io/react-native>
7. <https://viblo.asia/p/bat-dau-voi-react-native-maGK7jQD5j2>
8. <https://viblo.asia/tags/react-native>