

TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM
ĐẠI HỌC TÔN ĐỨC THẮNG
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO GIỮA KÌ
MÔN PHÁT TRIỂN ỨNG DỤNG DI ĐỘNG
TÌM HIỂU VỀ REACT NATIVE XÂY
DỰNG ỨNG DỤNG MINH HỌA

Người hướng dẫn: ThS HÀ LÊ HOÀI TRUNG

Người thực hiện: PHẠM ANH DUY - 51702088

TRẦN NGỌC BẢO DUY - 51702091

TRƯƠNG THỊ HUỲNH NHƯ - 51702154

Khóa: : 21

TP.Hồ Chí Minh, 2020

LỜI NÓI ĐẦU

Trong thời gian làm đề án này, chúng em đã nhận được nhiều sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của cô và bạn bè.

Nhóm em xin gửi lời cảm ơn chân thành đến thầy Hà Lê Hoài Trung, người đã tận tình hướng dẫn, chỉ bảo nhóm em trong suốt quá trình làm đề án.

Nhóm em cũng xin chân thành cảm ơn các thầy cô giáo trong trường ĐH Tôn Đức Thắng nói chung, các thầy cô trong khoa CNTT nói riêng đã dạy dỗ cho em kiến thức về các môn đại cương cũng như các môn chuyên ngành, giúp chúng em có được cơ sở lý thuyết vững vàng và tạo điều kiện giúp đỡ nhóm em trong suốt quá trình học tập.

Cuối cùng, nhóm em xin chân thành cảm ơn gia đình và bạn bè, đã luôn tạo điều kiện, quan tâm, giúp đỡ, động viên em trong suốt quá trình học tập và hoàn thành bài đề án này.

ĐỒ ÁN NÀY ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG

Tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của thầy Hà Lê Hoài Trung. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong bài đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình. Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

TP.Hồ Chí Minh Ngày 2 tháng 4 năm 2020

(Ký tên và ghi rõ họ tên)

PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN

Phần xác nhận của GV hướng dẫn

This image shows a single sheet of white paper with horizontal blue or grey ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

TP.Hồ Chí Minh, 2019

(Ký tên và ghi rõ họ tên)

This image shows a blank sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

(Ký tên và ghi rõ họ tên)

TÓM TẮT

Ngày nay, công nghệ thông tin đã và đang đóng vai trò quan trọng trong xã hội. Nó giúp con người làm việc với mức độ chính xác cao, quản lý và tổ chức công việc đạt hiệu quả cùng với đó thông tin cũng được phổ biến và cập nhật nhanh chóng. Làm thế nào có thể dễ dàng theo dõi phối hợp công việc hiệu quả giúp cho mọi người trong team chỉ cần nhìn qua là biết được có những đầu việc nào, ai đang làm gì, và làm đến giai đoạn nào rồi.

Vì vậy để đáp ứng được nhu cầu trên thì nhóm chúng em xin chọn đề tài phát triển hệ thống ứng dụng Co-Work. Do hạn chế về kiến thức về kỹ năng nên không tránh khỏi sai sót mong thầy thông cảm.

Mục lục

LỜI NÓI ĐẦU	i
TÓM TẮT	v
Danh sách hình vẽ	2
Danh sách bảng	3
1 MỞ ĐẦU	4
1.1 Giới thiệu đề tài	4
1.2 Phương pháp nghiên cứu	5
1.3 Ý nghĩa thực tiễn	5
2 GIỚI THIỆU VỀ NATIVE APP VÀ HYBRID APP	6
2.1 Native App	6
2.1.1 Khái niệm	6
2.1.2 Ưu điểm và nhược điểm	7
2.2 Mobile Web App	8
2.2.1 Khái niệm	8
2.2.2 Ưu điểm và nhược điểm	8
2.3 Hybrid App	10

2.3.1	Khái niệm	10
2.3.2	Ưu điểm và nhược điểm	10
3	TÌM HIỂU REACT NATIVE	12
3.1	Nguồn gốc	12
3.2	Ưu điểm	13
3.2.1	Tuy một mà hai, tuy hai mà một – Hai hệ sinh thái trong một	13
3.2.2	Khả năng reuse - sử dụng lại, viết một được hai	14
3.2.3	Hot reloading – Mã nguồn mở	14
3.2.4	Tổng kết ưu điểm và nhược điểm	15
4	MINH HỌA ỨNG DỤNG	17
4.1	Cài đặt môi trường	17
4.2	Chạy ứng dụng minh họa	18
5	ĐÁNH GIÁ VÀ HƯỚNG PHÁT TRIỂN	30
5.1	Kết quả đạt được	30
5.2	Hạn chế	30
5.3	Hướng phát triển	31
	Tài liệu tham khảo	32

Danh sách hình vẽ

4.1	Chạy server Node.js	18
4.2	Chạy server Python	19
4.3	QR-Expo	20
4.4	Expo load dữ liệu trước khi vào ứng dụng	21
4.5	Giao diện ứng dụng (1)	22
4.6	Giao diện ứng dụng (2)	23
4.7	Giao diện ứng dụng (3)	24
4.8	Giao diện ứng dụng (4)	25
4.9	Giao diện ứng dụng (5)	26
4.10	Giao diện ứng dụng (6)	27
4.11	Giao diện ứng dụng (7)	28
4.12	Giao diện ứng dụng (8)	29

Danh sách bảng

2.1	Ưu và nhược điểm của Native App	8
2.2	Ưu và nhược điểm của Mobile Web App	10
3.1	Ưu và nhược điểm của Web Mobile App	16

Chương 1

MỞ ĐẦU

1.1 Giới thiệu đề tài

Sự bùng nổ về công nghệ thông tin đã dẫn đến việc sử dụng smart phone đã và đang trở thành một thói quen của mọi người. Những công nghệ tân tiến ra đời kèm theo đó cũng không ít những nâng cấp từ các chương trình, bởi lẽ ai cũng biết công nghệ càng hiện đại thì ứng dụng càng phải có khả năng đáp ứng nhu cầu của người dùng càng cao. Từ lúc mới ra đời cho đến nay smartphone đã có những bước tiến mạnh mẽ vì vậy mà những công nghệ kèm theo cũng đòi hỏi những nhà phát triển phần mềm viết ra nhiều phần mềm hơn để có thể sử dụng được trên smartphone chứ không còn đơn thuần là những ứng dụng nhắn tin gọi điện thông thường.

Việc xây dựng được một ứng dụng tốt hay xấu, một ứng dụng có nhiều tiện ích cũng như khả năng đáp ứng, giải quyết nhu cầu của người sử dụng nhanh hay chậm cũng là nhờ cái cốt lõi của nó. Cái mà chúng em đang đề cập đến đó chính là những ngôn ngữ đã viết lên những ứng dụng, thật sự trên thế giới hiện tại đã hỗ trợ

khá nhiều ngôn ngữ để đáp ứng nhu cầu viết ứng dụng trên cả 2 hệ điều hành chính của smartphone đó chính là Android và iOS. Trong bài báo cáo này chúng em muốn báo cáo về việc tìm hiểu về một framework hỗ trợ viết ứng dụng cho cả Android và iOS đơn giản mà không cần dùng thêm bất cứ thêm một ứng dụng hỗ trợ nào khác, đó chính là React Native.

1.2 Phương pháp nghiên cứu

Phương pháp thu thập tài liệu: thu thập tài liệu từ sách, giáo trình, những bài blog về tin học, công nghệ, các video chia sẻ, hướng dẫn trên Youtube.

Phương pháp phân tích và tổng hợp tài liệu: từ những tài liệu đã thu thập, tiến hành tìm hiểu, phân tích và tổng hợp kiến thức.

Phương pháp chuyên gia: trong quá trình nghiên cứu, thực hiện đề tài có sự góp ý của giảng viên bộ môn.

1.3 Ý nghĩa thực tiễn

Thực hiện đề tài là cơ hội để sinh viên áp dụng vào thực tế những kiến thức đã học được từ môn Phát triển ứng dụng di động, hiểu biết sâu sắc hơn những kiến thức, kỹ năng cần thiết để thiết kế và xây dựng nên một ứng dụng di động.

Đem đến cho người đọc biết thêm những lợi ích mà React Native đã và đang mang lại cho ngành lập trình ứng dụng di động, những công cụ không thể thiếu khi nói về ứng dụng di động. Cũng như biết rõ vai trò quan trọng của smartphone trong nền công nghiệp 4.0 hiện tại.

Ngoài ra, đây cũng là cơ hội để chúng em tự trang bị thêm cho mình những kỹ năng làm việc với phần mềm lập trình ứng dụng di động Android Studio, React Native, công cụ quản lý mã nguồn Github.

Chương 2

GIỚI THIỆU VỀ NATIVE APP VÀ HYBRID APP

2.1 Native App

2.1.1 Khái niệm

Native app hay còn gọi là ứng dụng gốc được viết cho một loại nền tảng như iOS, Android, Windows Phone bằng các ngôn ngữ tương ứng của mỗi nền tảng đó ví dụ Java trên Android, Object C trên iOS, C# trên Windows Phone. Mỗi Native App chỉ chạy được trên một nền tảng và không thể mang sang các nền tảng khác. Ví dụ game cho iOS sẽ không thể chạy được trên các máy Android. Đa phần các game mobile hiện nay là Native App.

2.1.2 Ưu điểm và nhược điểm

Ưu điểm	Nhược điểm
Performance: native app chạy nhanh hơn mobile app trong hầu hết trường hợp, sự chênh lệch hiệu suất khá nhỏ khiến người dùng không thể nhận ra. Chính vì vậy nó cũng là một trong ưu tiên hàng đầu cho các ứng dụng game di động.	Không thể cross platform: một native app chỉ có thể chạy trên một hệ điều hành nhất định.
	Tính nhất quán: với mỗi HDH thì người lập trình phải viết riêng native code cho nó, dẫn đến không nhất quán giữa các phiên bản ứng dụng, chi phí phát triển cao và đòi hỏi khả năng thành thạo nhiều ngôn ngữ ở lập trình viên.
Feature: native app có khả năng truy cập các phần cứng (như camera, GPS, thiết bị thu âm...) và có thể truy cập đến các quyền local như : sao chép, tạo, ghi, đọc tập tin (trên bộ nhớ của mobile), các thông tin danh bạ, thông tin cá nhân, thực hiện cuộc gọi, nhắn tin...	Tính bảo trì hay nâng cấp sẽ làm mất nhiều thời gian. Do phải sửa chữa từng app trên từng HDH, thủ tục đưa lên chợ ứng dụng và chờ đợi approval (đôi khi bị reject vì vi phạm một số điều lệ của chợ ứng dụng như app store, CH Play), nên bản cập nhật không đến tay người dùng ngay lập tức, hơn nữa họ phải tự update khi vào đó.

Offline-mode: native app sẽ dùng những dữ liệu đã cache trước đó ở lần cuối cùng mà người dùng truy cập online.	Tính đa dạng phiên bản của HDH làm cản trở việc phổ dụng các native app. Các thiết bị chạy phiên bản HDH cũ hơn sẽ không cài đặt được.
---	--

Bảng 2.1: Ưu và nhược điểm của Native App

2.2 Mobile Web App

2.2.1 Khái niệm

Là ứng dụng chạy trên nền web, được viết bằng các ngôn ngữ web như HTML5, CSS, javascript hoặc jQuery. Về cơ bản là website với giao diện giống như app và nội dung được load từ web. Ứng dụng web dễ hình dung nhất là [Facebook](#).

2.2.2 Ưu điểm và nhược điểm

Ưu điểm	Nhược điểm
Cross platform: Có thể chạy trên tất cả trình duyệt của mobile hỗ trợ phiên bản HTML và javascript. Không cần cài đặt trên máy, gần như là sử dụng mọi thiết bị nào có trình duyệt web.	Phụ thuộc vào trình duyệt, nền tảng, loại máy. Đặc biệt khi mức độ phân mảnh thị trường của các thiết bị di động lớn như hiện nay lại càng làm các nhà lập trình khó khăn trong việc lựa chọn đúng phần nào của HTML5 có thể sử dụng.

<p>Thuận lợi cho các nhà phát triển:</p> <ul style="list-style-type: none"> • Viết 1 sử dụng cho tất cả: nên giảm chi phí và thời gian cho phát triển, bảo trì, cũng như nâng cấp sau này. • Ngôn ngữ lập trình phổ dụng là HTML, CSS và Javascript mà hầu hết các lập trình viên đều biết. • Không giới hạn môi trường lập trình, lập trình viên có thể làm việc trên hầu hết các HĐH. • Ứng dụng không cần phải được build lại qua SDK hoặc một công cụ phát triển độc quyền khác như xCode (chỉ cần deploy lên server và chạy). • Việc cập nhật phiên bản sẽ đến ngay lập tức và trong suốt với người dùng (ngay sau khi deploy) mà không cần thông qua thủ tục và chờ đợi approval như các chợ ứng dụng. 	<p>Performance: web-based app không chạy nhanh như native app, và không mang lại trải nghiệm tốt.</p>
---	---

Có thể được đánh dấu bởi search engines (tốt cho SEO). Dễ dàng quảng bá: người dùng có thể tìm thấy thông qua search engine, share link từ email, từ social networks hoặc từ các dịch vụ quảng cáo trực tuyến như Google Ads.	Luôn phải chạy online: Khi mạng chập chờn thì ứng dụng web cũng ngay lập tức thể hiện điểm yếu của mình.
---	--

Bảng 2.2: Ưu và nhược điểm của Mobile Web App

2.3 Hybrid App

2.3.1 Khái niệm

Hybrid App là ứng dụng kết hợp những ưu điểm của cả Mobile Web App và Native App. Tức là nó vừa có thể cài đặt lên điện thoại người dùng giống như là những ứng dụng Native bình thường và cũng có thể tìm thấy chúng trên chợ ứng dụng (App store) nhưng khó có thể nhận ra đâu là ứng dụng Native và đâu là ứng dụng Hybrid thường thì người dùng sẽ không để ý quá nhiều. Đồng thời như ứng dụng Web chúng được xây dựng dựa trên HTML + CSS + JavaScript.

2.3.2 Ưu điểm và nhược điểm

Hybrid có thể kết hợp điểm mạnh của cả native app và Mobile Web App và khai thác được trải nghiệm cao nhất của nền tảng. Cho nên nhược điểm của hybrid thường ít và thay vào đó là những ưu điểm của native app và web-base app.

Ví dụ như: Hiển thị nội dung trên tất cả các thiết bị di động, tận dụng tối đa các tính năng của thiết bị di động như GPS, Camera, cảm biến gia tốc...

điều này không thể làm được trên mobile web, sử dụng ứng dụng ngay cả khi không có kết nối internet. Những ứng dụng mobile của một số trang tin tức hiện nay như VnExpress, Dantri v.v.. đều là những ứng dụng lai.

Chương 3

TÌM HIỂU REACT NATIVE

3.1 Nguồn gốc

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. Cách vận hành của React Native là wrapper của code native, có rất nhiều thành tố được tạo nên khi đang wrapping – “bọc” chức năng native của iOS hoặc Android.

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. Một điểm hay ho nữa mà đượ đề cập là giảm chi phí recompile của Native bằng cách sử dụng Hot-Loading tức là không cần phải build lại ứng dụng từ đầu nên việc chỉnh sửa diễn ra rất nhanh chóng. Giúp cho lập trình viên có thể thấy được những chỉnh sửa của họ một cách nhanh chóng trực quan, không còn phải bỏ

quá nhiều thời gian trong việc build và run ứng dụng nữa.

3.2 Ưu điểm

3.2.1 Tuy một mà hai, tuy hai mà một – Hai hệ sinh thái trong một

Như đã đề cập ở bên trên, khi build một ứng dụng to lớn, thông thường lập trình viên hoặc người phân tích yêu cầu sẽ yêu cầu dùng ứng dụng native bởi vì hiệu suất và lợi ích mà nó mang lại, nhưng bên cạnh đó người dùng cũng cần chạy song song cả hai hệ điều hành IOS và Android.

Một lý do lập trình mobile app rất khó và tốn thời gian là vì thực tế cần tìm hiểu 2 hệ sinh thái hoàn toàn khác biệt.

- Nếu muốn lập trình app iOS thì phải học Swift hoặc Objective-C.
- Nếu muốn lập trình app Android thì phải cần học Java và Android SDK.

Tuy nhiên, sẽ ngắn khá nhiều thời gian cho việc học từng ngôn ngữ. Và tất nhiên, mỗi frameworks luôn có 1 gói các công cụ như công cụ testing, các libs, packages... và việc các lập trình phải cập nhật các tính năng mới nhất của mỗi hệ sinh thái là điều không thể bàn cãi. Mặc khác, nếu chọn lập trình trên React Native, phần lớn thời gian sẽ chỉ cần học 1 bộ công cụ. Có rất nhiều thứ như: JavaScript, Node, React Native... nhưng chỉ có 1 công cụ duy nhất để học và hỗ trợ cả hai hệ sinh thái, vì thế lợi ích nó mang lại là không thể chối cãi.

3.2.2 Khả năng reuse - sử dụng lại, viết một được hai

Khả năng sử dụng lại đóng vai trò quan trọng trong lập trình phần mềm và nó cũng là một trong những lợi ích mà react native mang lại. Viết một nhưng dùng nhiều, React Native không phải chỉ viết 1 lần mà nó chạy platform ở mọi nơi. Người sử dụng thường phân nó thành các Component để tiện quản lý và sử dụng lại.

Khi xây dựng một ứng dụng UI hoặc back-end sẽ luôn luôn có những đoạn code giống nhau về mặt công dụng, hoặc về mặt giao diện, lúc đó khả năng reuse của React Native sẽ phát huy tối đa tác dụng. Vì thế sẽ có rất nhiều lợi điểm như: tận dụng nguồn nhân lực tốt hơn, duy trì ít code hơn, ít bugs hơn, các tính năng trong cả 2 platforms cũng tương tự nhau, giảm thiểu cho người lập trình phải cover quá nhiều việc.

Khả năng viết ở 1 platform nhưng có thể dùng ở cả 2 platforms chính là ưu điểm của react native, một ưu điểm tạo cho người lập trình cảm thấy thoải mái, vì không cần phải ở viết lại cho platform này hay platform kia.

3.2.3 Hot reloading – Mã nguồn mở

Thông thường khi build một ứng dụng IOS hoặc là Android thì phải build trên một thiết bị thực sự hoặc simulator để hiện thực hoá kết quả ngay lập tức, đó là một cách trải nghiệm ứng dụng thực sự khi viết chương trình. Vì vậy khi build chương trình sẽ cần khá nhiều thời gian để ứng dụng khởi tạo trên simulator hoặc thiết bị, điều này làm cho người lập trình mất thời gian, cũng như cảm giác chán nản khi build chương trình.

Với React Native, phần lớn thời gian, không cần phải tổng hợp lại ứng dụng mỗi lần có thay đổi. Chúng ta chỉ cần làm mới ứng dụng trong simulator hoặc thiết bị. Thậm chí còn có 1 tính năng là Live Reload để tự động refresh app mỗi lần phát

hiện 1 thay đổi trong code.

Vì là mã nguồn mở nên React Native được 1 cộng đồng lớn các developer hỗ trợ thực hiện nhiều tính năng hơn, fix bugs... Được nhiều người sử dụng đồng nghĩa với việc cách giải quyết vấn đề khi gặp một bugs sẽ nhanh hơn rất nhiều, người sử dụng có thể tập trung làm những việc quan trọng hơn.

3.2.4 Tổng kết ưu điểm và nhược điểm

Ưu điểm	Nhược điểm
<ul style="list-style-type: none"> • Hiệu quả về mặt thời gian khi mà muốn phát triển một ứng dụng nhanh chóng. • Hiệu năng tương đối ổn định. • Cộng đồng phát triển mạnh. • Tiết kiệm tiền. • Team phát triển nhỏ. • Ứng dụng tin cậy và ổn định. • Xây dựng cho nhiều hệ điều hành khác nhau với ít native code nhất. • Trải nghiệm người dùng tốt hơn là hybrid app. 	<ul style="list-style-type: none"> • Vẫn đòi hỏi native code. • Hiệu năng sẽ thấp hơn với app thuần native code. • Bảo mật không cao do dựa trên JS. • Quản lý bộ nhớ và khả năng tùy biến cũng không thực sự tốt đối với một vài module.

Bảng 3.1: Ưu và nhược điểm của Web Mobile App

Chương 4

MINH HỌA ỨNG DỤNG

4.1 Cài đặt môi trường

Bước 1 Cài [Node.js](#)

Bước 2 Cài đặt create-react-native-app

```
1 npm install -g create-react-native-app
2
```

Listing 4.1: Cài đặt create-react-native-app

Bước 3 Tạo project và run project mẫu

```
1 create-react-native-app AwesomeProject
2
3 cd AwesomeProject
4 npm start
5
```

Listing 4.2: Tạo project và run project mẫu

Bước 4 Khởi chạy project trên điện thoại

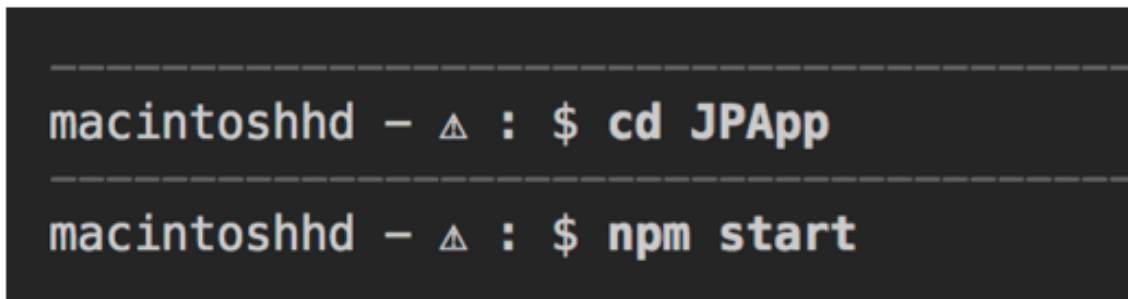
- Đầu tiên cài đặt Expo client trên điện thoại Android, iOS.
- Kết nối cùng mạng wifi với máy tính mà chúng ta run project.
- Sau đó mở ứng dụng Expo Client lên và quét mã QR trên terminal của run project.
- Khi đó thay đổi gì trong App.js thì ứng dụng sẽ tự động thay đổi tự động.

4.2 Chạy ứng dụng minh họa

Bước 1 cd vào thư mục chứa ứng dụng và chạy npm start để start UI, và python run.py để chạy server lấy data.

```
1 cd JPAApp
2
```

Listing 4.3: cd JPAApp



Hình 4.1: Chạy server Node.js

Start server python

```
1 python run.py
```

Listing 4.4: cd JApp

```
(jpp)
macintoshhd - Δ : $ python run.py
* Serving Flask app "JAppServer" (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: on
* Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
* Restarting with stat
* Debugger is active!
* Debugger PIN: 131-215-928
127.0.0.1 - - [16/Mar/2020 00:02:58] "GET / HTTP/1.1" 404 -
127.0.0.1 - - [16/Mar/2020 00:02:58] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [16/Mar/2020 00:02:58] "GET / HTTP/1.1" 404 -
127.0.0.1 - - [16/Mar/2020 00:04:35] "GET / HTTP/1.1" 404 -
127.0.0.1 - - [16/Mar/2020 00:04:36] "GET /favicon.ico HTTP/1.1" 404 -
```

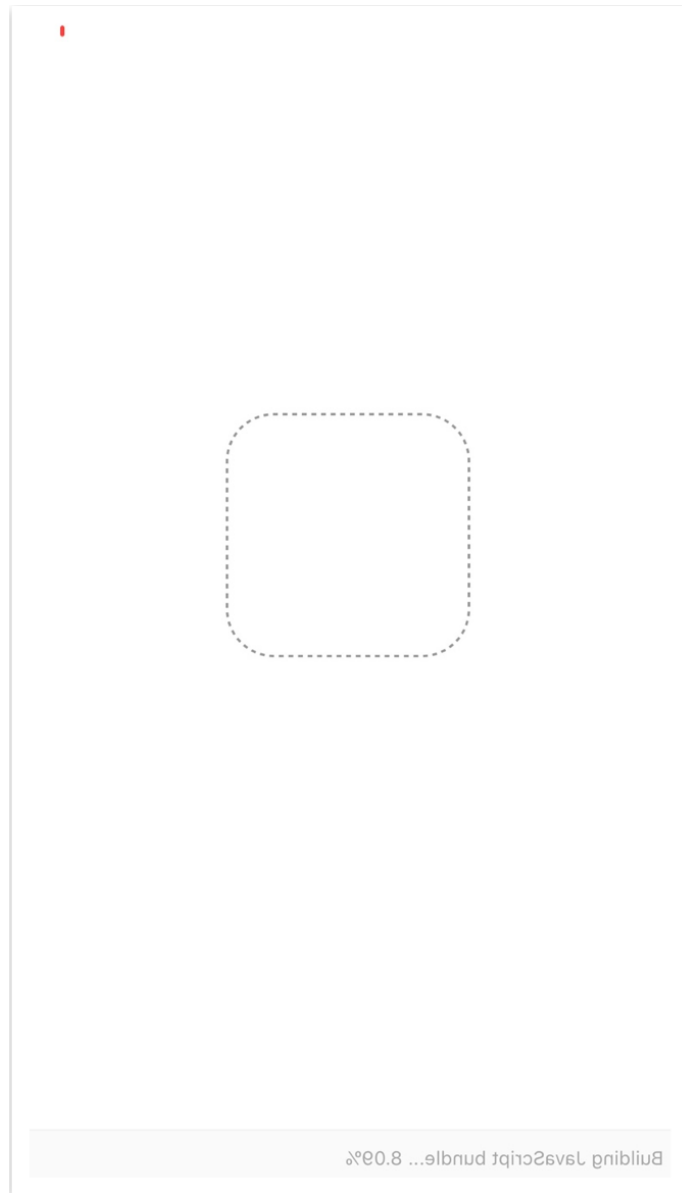
Hình 4.2: Chạy server Python

Bước 2 Mở ứng dụng expo trong điện thoại của mình và quét mã QR trong terminal.



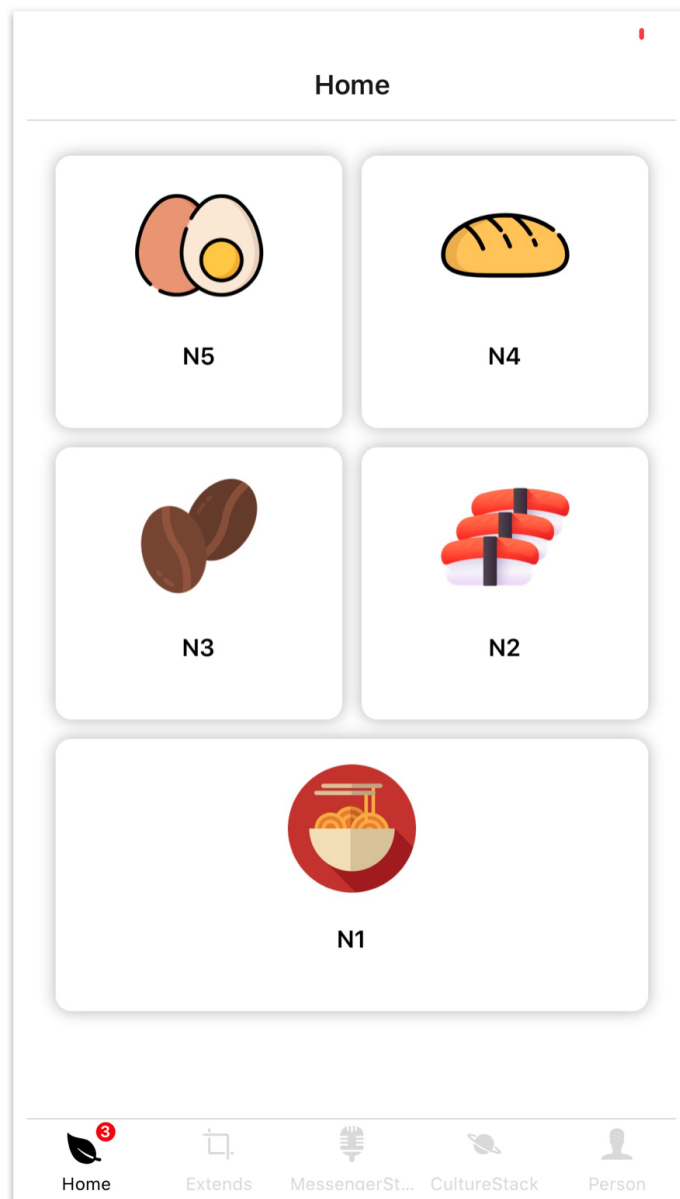
Hình 4.3: QR-Expo

Bước 3 Màn hình load chương trình

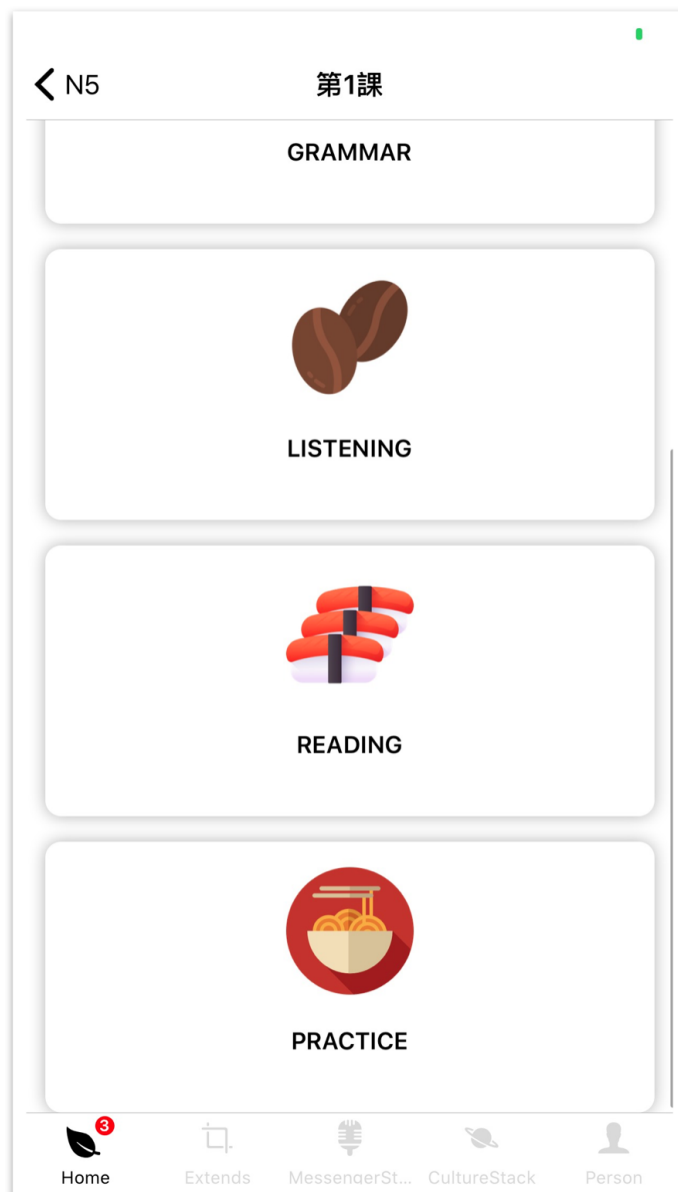


Hình 4.4: Expo load dữ liệu trước khi vào ứng dụng

Bước 4



Hình 4.5: Giao diện ứng dụng (1)



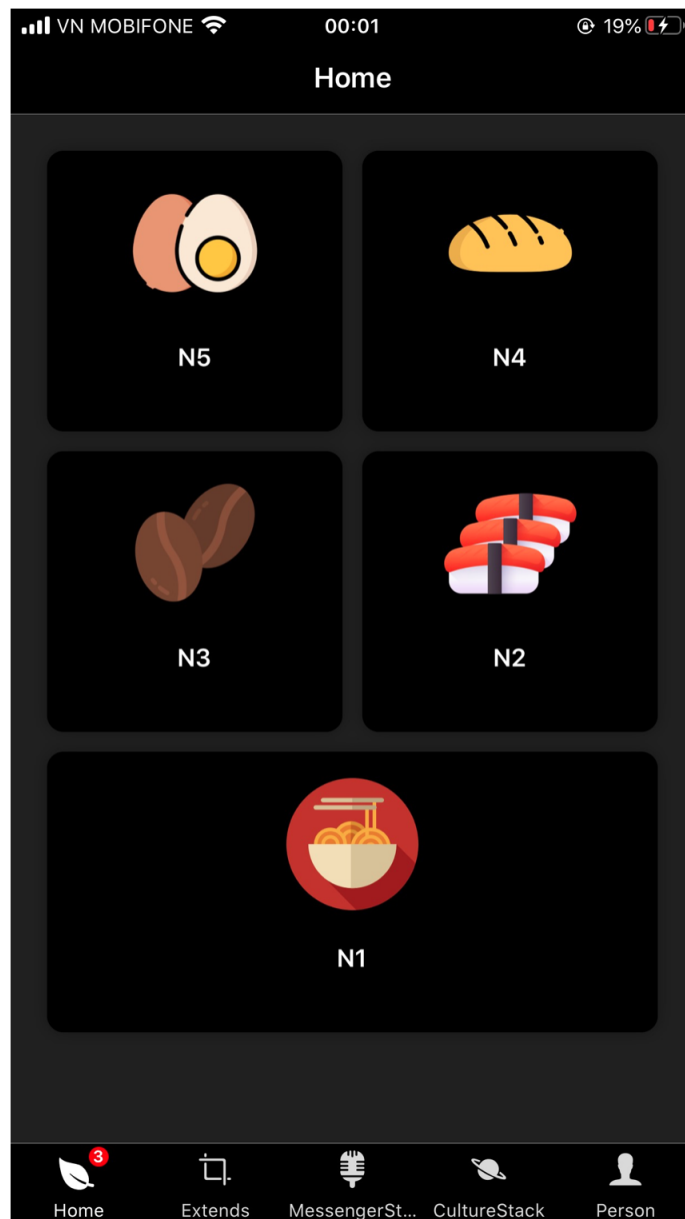
Hình 4.6: Giao diện ứng dụng (2)



Hình 4.7: Giao diện ứng dụng (3)



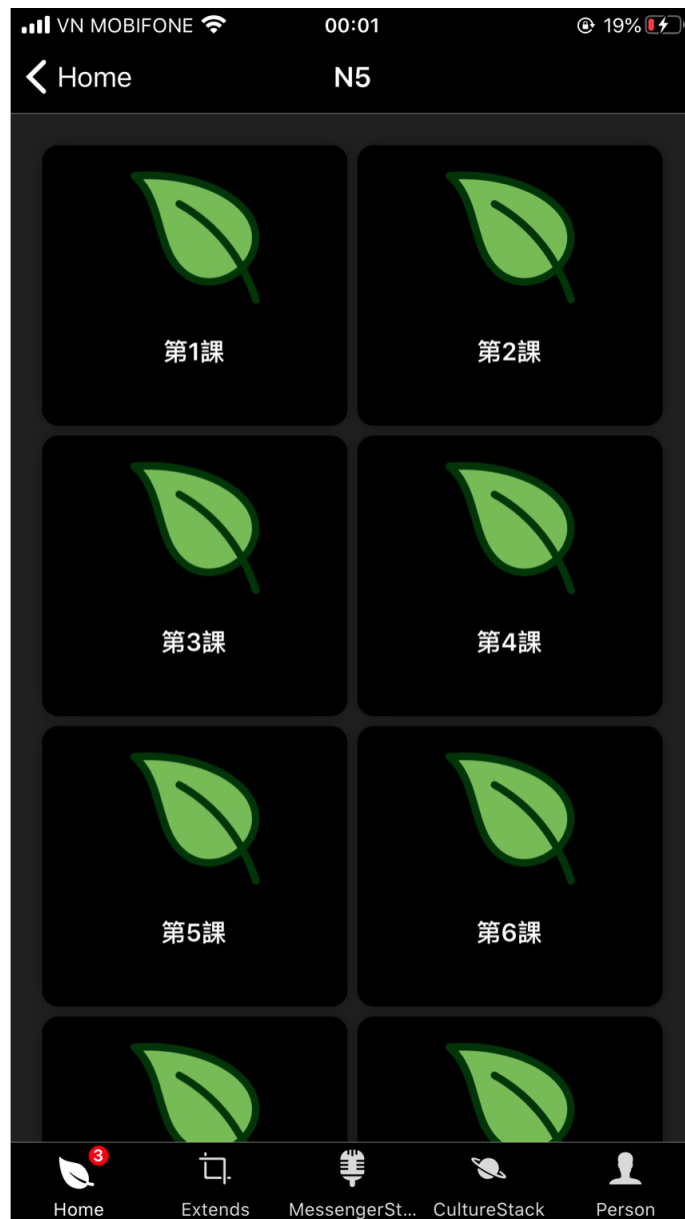
Hình 4.8: Giao diện ứng dụng (4)



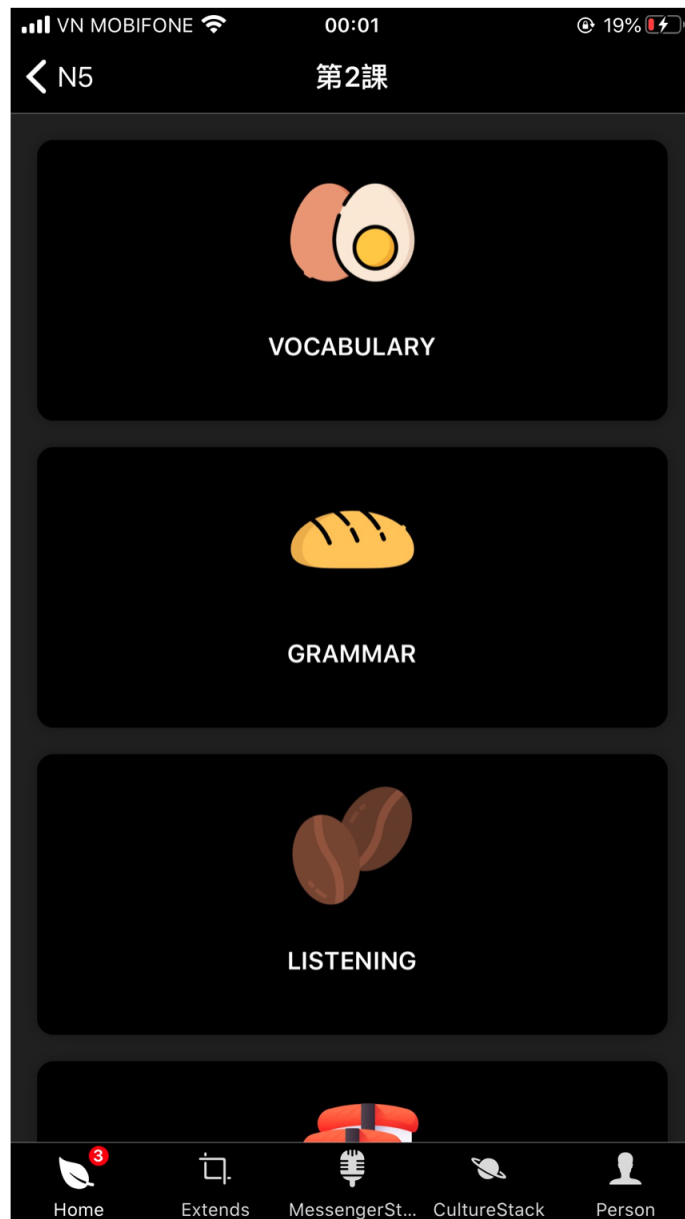
Hình 4.9: Giao diện ứng dụng (5)



Hình 4.10: Giao diện ứng dụng (6)



Hình 4.11: Giao diện ứng dụng (7)



Hình 4.12: Giao diện ứng dụng (8)

Chương 5

ĐÁNH GIÁ VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết quả đạt được

Qua bài báo cáo “Tìm hiểu về React Native xây dựng ứng dụng minh họa” em đã biết về nguồn gốc, những ưu nhược điểm của React Native, đồng thời chúng em cũng đã xây dựng một ứng dụng bằng React Native. Giúp chúng em phân biệt được Native App, Web Mobile App, Hybrid App và nắm được điểm mạnh cũng như hạn chế của từng loại App, biết trường hợp nào nên sử dụng công nghệ nào cho phù hợp và tiết kiệm nhất có thể.

5.2 Hạn chế

- Không thể build được các ứng dụng quá phức tạp.
- Còn thiếu nhiều component view (có thể đợi cộng đồng phát triển lên).

- Chỉ hỗ trợ 2 nền tảng iOS và Android.
- Khả năng cập nhật không được nhanh được như native.
- Có một số thư viện cần phải trả phí.

5.3 Hướng phát triển

Trong tương lai, em sẽ sử dụng React Native để xây dựng cho riêng mình những app Android cũng như IOS có tích hợp Artificial Intelligence hoặc những ứng dụng Recommendation vào hệ thống, bởi vì tính tiện dụng, có thể tái sử dụng component của nó nên cũng không khó để thay đổi giao diện, mục đích khi đã nắm rõ cách hoạt động của React Native. Có thể áp dụng thêm Redux để tăng khả năng lưu trữ, tính mượt mà cho app.

Tài liệu tham khảo

[1] TECH Talk,

<https://techtalk.vn/su-khac-nhau-giua-native-app-mobile-web-app-va-hybrid-app.html>

[2] TOP Dev,

<https://topdev.vn/blog/lap-trinh-app-su-dung-react-native-so-voi-android-ios/>

[3] NORDIC CODER,

<https://nordiccoder.com/blog/tong-quan-ve-react-native/>