

ĐẠI HỌC QUỐC GIA TP.HCM
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH



LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC

Xây dựng hệ thống đăng tin môi giới bất động sản

HỘI ĐỒNG: HỘI ĐỒNG 1 - HỆ THỐNG THÔNG TIN

GVHD: Trần Minh Quang

GVPB: Trần Thị Quế Nguyệt

---o0o---

SVTH: Nguyễn Ngọc Duy (51300595)

TP. HỒ CHÍ MINH, THÁNG 06 NĂM 2018

Lời cam đoan

Em xin cam đoan đây là công trình nghiên cứu và thực hiện bởi riêng em dưới sự hướng dẫn của giảng viên hướng dẫn. Nếu không đúng như đã nêu trên, em xin hoàn toàn chịu trách nhiệm về đề tài của mình.

Lời cảm ơn

Em xin chân thành cảm ơn các thầy cô trường Đại học Bách Khoa TP.HCM nói chung, cũng như các thầy cô khoa Khoa học và Kỹ thuật Máy tính nói riêng, với những sự hỗ trợ nhiệt tình cũng như những bài học bổ ích về cả kiến thức chuyên ngành lẫn cho cuộc sống của em.

Trong quá trình thực hiện đề tài, do kỹ năng kinh nghiệm thực tế chưa nhiều nên không thể tránh khỏi những thiếu sót, em mong được quý thầy cô góp ý cho em để em có thể học tập được thêm nhiều kiến thức hay.

Cuối cùng, em xin kính chúc quý thầy cô dồi dào sức khỏe và ngày một thành công trong công việc, giúp cho các thế hệ tiếp theo được dìu dắt để có thể đóng góp được nhiều hơn cho gia đình và xã hội.

Tóm tắt luận văn

Đề tài yêu cầu xây dựng một hệ thống đăng tin môi giới bất động sản, đáp ứng được các chức năng cần có của một hệ thống đăng tin bất môi giới động sản, đồng thời cung cấp các chức năng mới, hữu ích hơn, giúp người sử dụng được tiện lợi hơn.

Đề tài được thực hiện xây dựng giải pháp cho một hệ thống đăng tin môi giới bất động sản, hoạt động trên đa nền tảng, cung cấp các chức năng cơ bản của một hệ thống đăng tin môi giới bất động sản, cũng như cung cấp chức năng tìm kiếm thông minh qua hình ảnh, giúp người tìm tin đăng được chủ động được với những thông tin cần tìm kiếm. Giải pháp được xây dựng hướng đến mục đích hoạt động với hiệu năng tốt hơn các hệ thống đã có sẵn, dễ dàng mở rộng, cũng như bám sát với nghiệp vụ nhà đất của thị trường.

Nội dung của luận văn được tóm tắt trong các chương sau:

Chương 1: Phân tích đề tài

Chương 2: Phân tích thị trường

Chương 3. Giải pháp thực hiện

Chương 4. Công nghệ sử dụng

Chương 5. Quy trình thực hiện

Chương 6. Kết luận chung về giải pháp

Mục lục

Lời cam đoan.....	2
Lời cảm ơn.....	2
Tóm tắt luận văn.....	3
Danh mục hình.....	5
Danh mục bảng.....	5
Chương 1. Phân tích đề tài	6
Chương 2. Phân tích thị trường	7
Chương 3. Giải pháp thực hiện	8
Chương 4. Công nghệ sử dụng	10
4.1. Nền tảng phần mềm thiết bị di động	10
4.2. Trung tâm lưu trữ và xử lý dữ liệu (Server).....	11
4.3. Cơ sở dữ liệu	15
4.4. Phương thức truyền nhận dữ liệu	16
4.5. Ứng dụng trí tuệ nhân tạo	19
Chương 5. Quy trình thực hiện	21
5.1. Thiết kế cơ sở dữ liệu.....	21
5.1.1. Phân tích rõ yêu cầu	21
5.1.2. Phân tích cơ sở dữ liệu	22
5.1.3. Thiết kế chi tiết cơ sở dữ liệu	24
5.2. Thiết kế giao diện.....	28
5.3. Viết chương trình lưu trữ và xử lý thông tin.....	33
5.3.1. Thiết lập môi trường.....	33
5.3.2. Tạo MVC framework	34
5.3.3. Tạo Parser để nhận thông tin truy vấn.....	35
5.4. Viết chương trình di động	38
5.5. Thiết kế và áp dụng Tensorflow vào nhận dạng nhà trên thiết bị di động.....	39
Chương 6. Kết luận chung về giải pháp	41
6.1. Đề tài và nhiệm vụ đã hoàn thành.....	41
6.1.1. Đề tài luận văn.....	41
6.1.2. Nhiệm vụ hoàn thành.....	41
Danh mục tài liệu tham khảo	42

Danh mục hình

Hình 1	Mô hình quy trình nghiệp vụ hệ thống môi giới bất động sản	7
Hình 2	Minh họa một số phần mềm môi giới bất động sản hiện có	8
Hình 3	Tương tác giữa các thành phần thuộc hệ thống	9
Hình 4	Luồng xử lý dữ liệu tại server	10
Hình 5	Mô tả ngắn gọn về React Native	11
Hình 6	So sánh tốc độ làm việc của PHP và NodeJS	14
Hình 7	So sánh điểm xếp hạng giữa MySQL và MongoDB	16
Hình 8	So sánh cấu trúc của XML và JSON	18
Hình 9	So sánh thời gian xử lý trên thiết bị di động giữa XML và JSON	19
Hình 10	Thiết kế chi tiết cơ sở dữ liệu	24
Hình 11	Sơ đồ khối mô tả thuật toán phân quyền thành viên trên loại bài viết	25
Hình 12	Sơ đồ khối mô tả thuật toán đăng ký và xác thực	26
Hình 13	Sơ đồ khối mô tả thuật toán đăng nhập và gửi yêu cầu cần quyền thành viên	26
Hình 14	Ví dụ về bài toán phân quyền	27
Hình 15	Sơ đồ khối mô tả thuật toán mua gói ưu tiên bài viết	27
Hình 16	Sơ đồ khối mô tả thuật toán tính số tiền phải trả khi mua gói mới	28
Hình 17	Thiết kế giao diện – Màn hình chính	29
Hình 18	Thiết kế giao diện – Màn hình tìm kiếm	29
Hình 19	Thiết kế giao diện – Màn hình danh sách bài viết	30
Hình 20	Thiết kế giao diện – Màn hình danh sách bài viết – Mục Sắp xếp	30
Hình 21	Thiết kế giao diện – Màn hình thông tin chi tiết bài viết	31
Hình 22	Thiết kế giao diện – Màn hình bản đồ	31
Hình 23	Thiết kế giao diện – Màn hình đăng tin	32
Hình 24	Thiết kế giao diện – Màn hình danh sách môi giới	32
Hình 25	Thiết kế giao diện – Màn hình danh sách môi giới – Mục Sắp xếp	33
Hình 26	Tổ chức cấu trúc cho mô hình MVC	34
Hình 27	Khởi tạo các trình tải config, library, view và model	34
Hình 28	Mô tả cấu trúc và quy trình xử lý gói tin truyền nhận	35
Hình 29	Mô tả đoạn mã xử lý yêu cầu nhận được tại server	37
Hình 30	Mô tả đoạn mã xử lý dữ liệu của từng loại tin yêu cầu tại server	38
Hình 31	Mô tả tập dữ liệu dùng để training	40
Hình 32	Mô tả sắp xếp thư mục phục vụ training	40
Hình 33	Đoạn mã sử dụng để nhận dạng nhà bằng Tensorflow trên React Native	41

Danh mục bảng

Bảng 1	So sánh công năng các hệ thống đăng tin môi giới bất động sản nổi tiếng	8
Bảng 2	So sánh giữa PHP và NodeJS	14

Chương 1. Phân tích đề tài

Từ yêu cầu của đề tài, chương trình cần đáp ứng được các yêu cầu có thể được liệt kê chi tiết ra như sau:

1. Người môi giới:

a. Môi giới cá nhân:

- i. Đăng thông tin BĐS muốn bán/cho thuê/chia sẻ.
- ii. Liên hệ được với những người có nhu cầu (email, điện thoại, hệ thống nhắn tin được cung cấp).
- iii. Chọn mức độ ưu tiên bài viết, dựa trên khoản chi phí bỏ ra được cho bài viết (hoặc chọn miễn phí cho mức ưu tiên thấp nhất).

Chi phí: Chi trả cho hệ thống để bài viết được ưu tiên.

Thu nhập: Được chi trả trực tiếp từ người mua (hoặc người gửi bán) khi giao dịch thành công.

b. Môi giới doanh nghiệp:

- i. Đăng thông tin BĐS muốn bán/cho thuê/chia sẻ.
- ii. Đăng thông tin dự án, thông tin tham quan dự án.
- iii. Liên hệ được với những người có nhu cầu (email, điện thoại, hệ thống nhắn tin được cung cấp).
- iv. Chọn mức độ ưu tiên bài viết, dựa trên khoản chi phí bỏ ra được cho bài viết (hoặc chọn miễn phí cho mức ưu tiên thấp nhất).

Chi phí: Chi trả cho hệ thống để bài viết được ưu tiên.

Thu nhập: Được chi trả trực tiếp từ người mua (hoặc người gửi bán) khi giao dịch thành công.

2. Khách:

- i. Xem/Tìm kiếm các BĐS, dự án, người môi giới mà họ quan tâm.
- ii. Tìm kiếm theo loại BĐS, vị trí hành chính, vị trí thực địa (map location), khoảng giá, khoảng diện tích, số tầng, hướng, chiều rộng đường, dự án.
- iii. Liên lạc với người môi giới theo dự án, theo danh sách bạn bè đã liên kết sẵn qua điện thoại, email, hoặc công cụ hỗ trợ của hệ thống.

Chi phí: Chi phí mua BĐS, đưa cho bên bán.

Thu nhập: Không.

3. Điều hành viên:

- i. Quản lý các thành viên trong hệ thống.
- ii. Quản lý (Xem, duyệt, gia hạn, chỉnh sửa, xóa) các bài đăng.
- iii. Quản lý (Xem, tìm kiếm, tạo, sửa, xóa) các giao dịch nạp tiền vào hệ thống.

iv. Trả lời các liên hệ của thành viên thông qua công cụ do hệ thống cung cấp.

Chi phí: Không có.

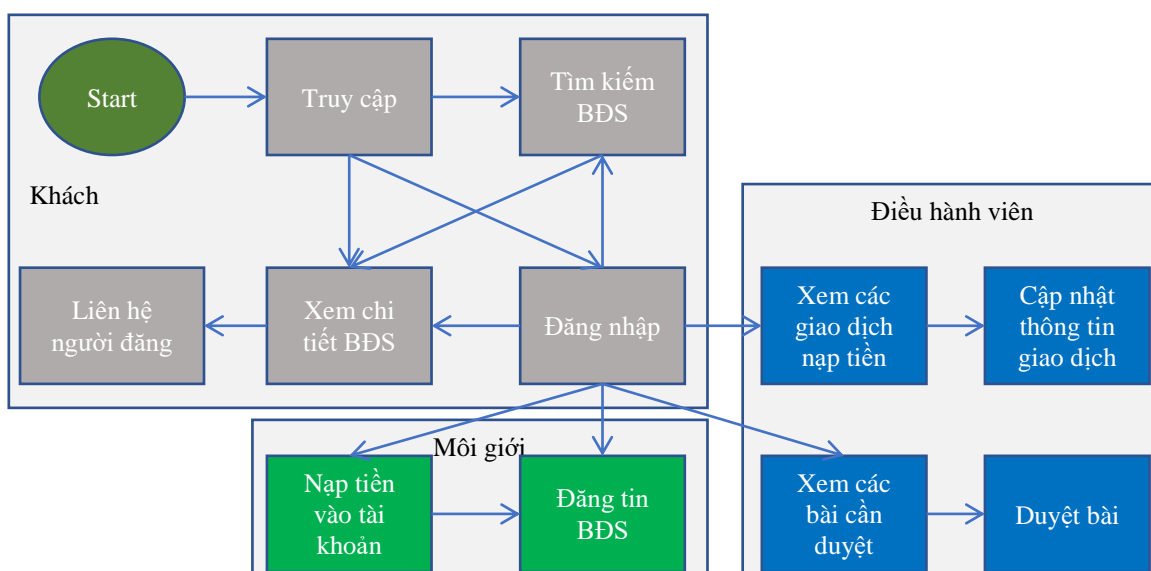
Thu nhập: Tiền lương từ công ty quản lý công cụ.

4. Công ty phát hành, quản lý công cụ:

Chi phí: Lương cho nhân viên điều hành, chi phí marketing.

Thu nhập: Tiền nạp vào từ các thành viên, thu nhập từ quảng cáo.




Quy trình nghiệp vụ của hệ thống cần được đáp ứng có thể được biểu diễn như sau:



Hình 1 Mô hình quy trình nghiệp vụ hệ thống môi giới bất động sản

Chương 2. Phân tích thị trường

Qua nghiên cứu, tìm hiểu, trên thị trường hiện đang có các hệ thống đăng tin môi giới bất động sản lớn đang hoạt động. Trong đó có ba trang thông tin / hệ thống đăng tin môi giới bất động sản tiêu biểu như sau, đi kèm với các thông tin chi tiết của từng hệ thống:

#	Tên chức năng			
1	Các tính năng cơ bản (Xem, Tìm kiếm, Lọc tin, Đăng Tin,...)	Có	Có	Có
2	Hỗ trợ đăng tin thường (Miễn phí)	Không	Có	Có
3	Hỗ trợ đăng tin VIP (Nạp tiền)	Có	Có	Có
4	Có bản đồ trong tin đăng	Có	Có	Có

5	Tin đăng có hình ảnh	Có	Có	Có
6	Tin đăng có hình ảnh 3D	Có	Không	Không
7	Hỗ trợ chụp hình ảnh 3D	Không	Không	Không
8	Hỗ trợ đăng tin dự án	Có	Có	Không
9	Hỗ trợ đăng tin phòng trọ	Có	Có	Không
10	Hỗ trợ liên hệ trợ giúp trực tuyến	Có	Không	Không
11	Chia sẻ thông tin lên mạng xã hội	Có	Không	Có
12	Nhiều/Mập mờ thông tin	Có	Có	Có
13	Phương pháp thanh toán linh hoạt (Thẻ cào, Chuyển khoản, Tại chi nhánh,...)	Có	Có	Có
14	Đội ngũ nhân viên hỗ trợ sẵn sàng liên lạc qua điện thoại	Có	Không	Không
15	Tổng đài hỗ trợ khách hàng	Có	Có	Có
16	Cho phép bình luận tại tin đăng	Không	Không	Không
17	Cho phép phản ánh tin đăng	Có	Có	Không
18	Gợi ý các tin liên quan	Có	Có	Có
19	Tư vấn nhà/đất theo phong thủy	Không	Không	Có
20	Hỗ trợ tìm kiếm nhà môi giới	Có	Có	Không

Bảng 1 So sánh công năng các hệ thống đăng tin môi giới bất động sản nổi tiếng



Hình 2 Minh họa một số phần mềm môi giới bất động sản hiện có

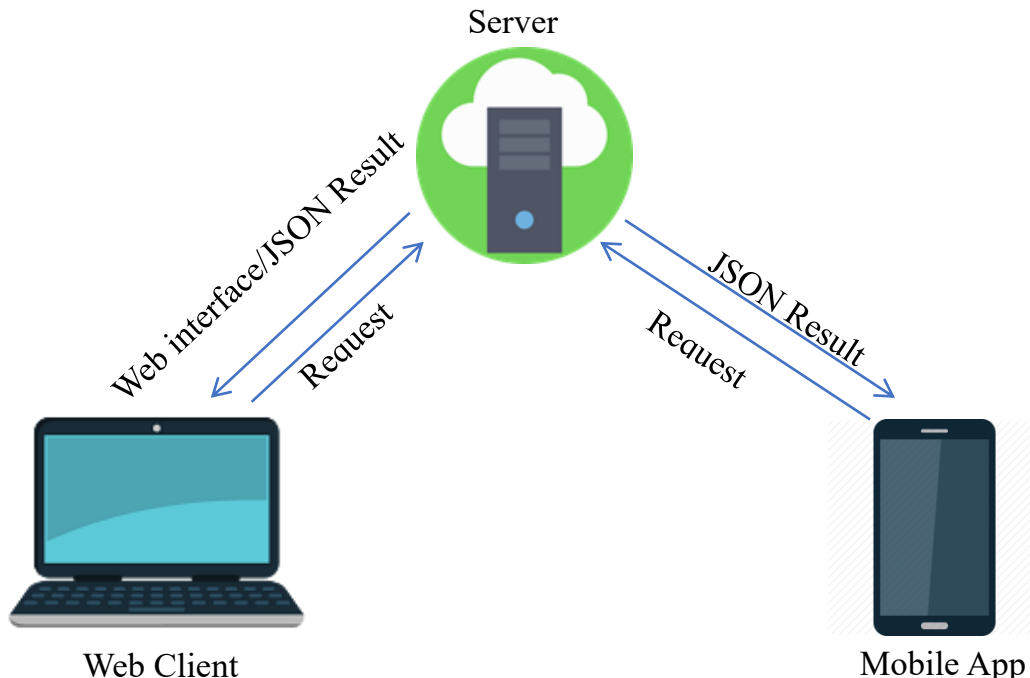
Chương 3. Giải pháp thực hiện

Với yêu cầu được phân tích tại Chương 1 và Chương 2, hệ thống cần được thực hiện bao gồm các thành phần sau:

1. Trung tâm lưu trữ - truy xuất thông tin: Là một máy chủ có nhiệm vụ lưu trữ và xử lý các yêu cầu từ các thiết bị, nền tảng khác nhau.

2. Trang web quản lý: Là công cụ dành cho nghiệp vụ quản lý, phục vụ các thao tác quản lý thành viên, bài viết, tài khoản, cũng như hệ thống.
3. Phần mềm di động: Là công cụ dành cho người dùng, phục vụ các thao tác xem, tìm kiếm và đăng các thông tin về môi giới bất động sản. Ngoài ra người dùng cũng có quyền (có giới hạn) đối với thông tin cá nhân tài khoản.

Mô hình liên kết giữa các thành phần này được biểu diễn như sau:



Hình 3 Tương tác giữa các thành phần thuộc hệ thống

Kiểu dữ liệu được lựa chọn để truyền tải giữa các thành phần thuộc hệ thống là dưới dạng JSON, với phân tích như sau:

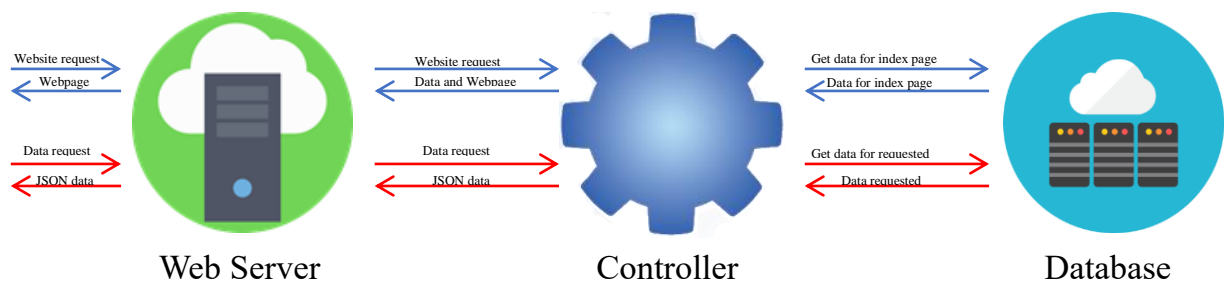
Ưu điểm:

1. Lưu trữ tạm data (JSON) trên local giúp phần mềm hoạt động được ngay cả khi có sự cố về mạng
2. Giúp đồng bộ phương thức trao đổi giữa phần mềm di động và trang web với máy server
3. Thời gian/Dung lượng tải nội dung cập nhật được giảm bớt so với việc tải lại toàn bộ nội dung website (Đối với website)

Nhược điểm:

1. Giải pháp có thể khiến thiết bị quá tải khi phải xử lý cục bộ lượng lớn dữ liệu
2. Khi có sự cập nhật về cấu trúc dữ liệu, chương trình di động cần phải được cập nhật lại

Luồng xử lý dữ liệu được thực hiện theo mô hình sau:



Hình 4 Luồng xử lý dữ liệu tại server

Chương 4. Công nghệ sử dụng

4.1. Nền tảng phần mềm thiết bị di động

React native là một framework cho phép các lập trình viên xây dựng các ứng dụng native mà chỉ sử dụng ngôn ngữ lập trình javascript. React native cho phép xây dựng các ứng dụng trên android và ios chỉ với một ngôn ngữ thống nhất là javascript nhưng mang lại trải nghiệm native app thực sự. Không như các framework hybrid khác (viết một lần triển khai nhiều nơi), React native tập trung vào việc một lập trình viên làm việc hiệu quả trên môi trường đa nền tảng như thế nào.

So với sử dụng Android SDKs mặc định, React Native có những điểm như sau:

a. Điểm mạnh:

- Dễ dàng thiết kế: Sử dụng React Native cho phép dễ dàng tạo các màn hình và tùy chỉnh chúng, vì đơn thuần như làm việc với HTML và CSS. Ngoài ra, các màn hình sẽ được cập nhật tự động ngay lập tức khi sử dụng tính năng “hot-loading”, do đó người thiết kế có thể thấy được kết quả ngay lập tức.
- Lập trình đa nền tảng: Lập trình viên có thể chạy chương trình trên cả Android, iOS, hay thậm chí là cả Windows Phone với cùng một đoạn chương trình, chỉ với một vài thay đổi cần thiết trên các nền tảng này.
- Dễ sử dụng API: API dùng trên React Native dễ sử dụng hơn rất nhiều so với trên Android, và ít phải đối mặt với các giá trị trả về là null.
- Dễ dàng tạo các component tùy biến: React Native cho phép dễ dàng tùy biến, tạo các component mới trong trường hợp các component có sẵn không đúng ý của người dùng.

b. Điểm yếu:

- Cài đặt chuyên trang (Navigation) khó khăn: Có nhiều thư viện và component hỗ trợ giải quyết vấn đề này, tuy nhiên vẫn chưa có giải pháp nào thực sự hiệu quả.

- Hiệu năng xử lý UI không cao: Người dùng cần phải sử dụng công nghệ tương tự trên nền tảng web, và việc làm cho các thao tác đạt được 60 fps là thật sự khó khăn. Các Thread nền khó có thể được tăng hiệu năng, và các Thread chạy thời gian dài có chạy tốt hay không phụ thuộc vào thư viện cung cấp công cụ quản lý.
- Việc quản lý thư viện không tốt: Đa phần các thư viện được xài đều yêu cầu cần phải chỉnh sửa ít nhiều mã lệnh khi chạy trên Android hoặc iOS. Do đó, người dùng vẫn cần phải biết lập trình iOS hoặc Android.



Hình 5 Mô tả ngắn gọn về React Native

Xét với nhu cầu thực tiễn của môi giới bất động sản, sẽ có những yêu cầu sau:

1. Lượng môi giới bất động sản sử dụng iOS là khá cao, nên cần một môi trường đa nền tảng để tối ưu hóa thời gian phát triển phần mềm.
2. Linh hoạt trong tùy biến để đáp ứng các nhu cầu phát sinh của thị trường.

So sánh giữa ưu, nhược điểm và nhu cầu thực tế, React Native là một môi trường làm việc phù hợp để phát triển một ứng dụng phục vụ cho môi giới bất động sản.

4.2. Trung tâm lưu trữ và xử lý dữ liệu (Server)

Có hai công nghệ có thể được nhắc ngay đến khi lựa chọn để thực hiện các ứng dụng server-side, đó là PHP và NodeJS.

- PHP (viết tắt "PHP: Hypertext Preprocessor") là một ngôn ngữ lập trình kịch bản hay một loại mã lệnh chủ yếu được dùng để phát triển các ứng dụng viết cho máy chủ, mã nguồn mở, dùng cho mục đích tổng quát. Nó rất thích hợp với

web và có thể dễ dàng nhúng vào trang HTML. Do được tối ưu hóa cho các ứng dụng web, tốc độ nhanh, nhỏ gọn, cú pháp giống C và Java, dễ học và thời gian xây dựng sản phẩm tương đối ngắn hơn so với các ngôn ngữ khác nên PHP đã nhanh chóng trở thành một ngôn ngữ lập trình web phổ biến nhất thế giới.

(Trích: Wikipedia^[2])

- NodeJS là một mã nguồn được xây dựng dựa trên nền tảng Javascript V8 Engine, nó được sử dụng để xây dựng các ứng dụng web như các trang video clip, các forum và đặc biệt là trang mạng xã hội phạm vi hẹp. NodeJS là một mã nguồn mở được sử dụng rộng bởi hàng ngàn lập trình viên trên toàn thế giới. NodeJS có thể chạy trên nhiều nền tảng hệ điều hành khác nhau từ Window cho tới Linux, OS X nên đó cũng là một lợi thế. NodeJS cung cấp các thư viện phong phú ở dạng Javascript Module khác nhau giúp đơn giản hóa việc lập trình và giảm thời gian ở mức thấp nhất.

(Trích: Freetuts^[8])

Ta có thể so sánh PHP và NodeJS thông qua một số tiêu chí như sau:

Tiêu chí	PHP	NodeJS
Độ phức tạp của các khái niệm và triển khai	<p>Phần code PHP có thể được đặt ở bất kỳ tập tin nào được thông dịch bởi engine PHP - thường với phần mở rộng .php. Phần code đó sẽ chỉ chạy thông qua một máy chủ web với PHP đã được cài đặt.</p> <p>Hầu hết các HĐH đều cung cấp các phần mềm máy chủ như IIS trên Windows hoặc Apache trên máy Mac và Linux, mặc dù chúng cần phải được kích hoạt và cấu hình. Nếu muốn đơn giản hơn bạn có thể sử dụng một pre-built set-up như XAMPP hoặc một virtual OS image (như Vagrant). Thậm chí dễ dàng hơn: chỉ việc tải các tập tin của bạn lên bất kỳ máy chủ web nào cũng được.</p> <p>Không giống như một số ngôn ngữ và các framework khác, PHP không bắt buộc phải làm việc một cách chính xác theo hướng nào đó. Lập trình viên có thể bắt đầu với một chương</p>	<p>Việc cài đặt Node.js rất là dễ dàng, có thể thực hiện bằng cách tải về bản cài đặt hoặc sử dụng một trình quản lý gói.</p> <p>Tuy nhiên, các câu lệnh cần để chạy một chương trình đơn giản, lại phải viết rất phức tạp với các cấu trúc đảm bảo cả việc giao tiếp với các request được gửi đến, port,... khiến các khái niệm trở nên khó hiểu hơn với những người mới bắt đầu.</p> <p>JavaScript tương đối súc tích hơn, với một vài chục câu lệnh cốt lõi. Điều đó nói rằng, cú pháp của nó thu hút các nhà phát triển, bởi vì mô hình đối tượng nguyên mẫu của nó có vẻ quen thuộc nhưng không hẳn là như vậy.</p>

	<p>trình bằng vài dòng code, bổ sung thêm các function, viết code theo lối đơn giản như các đối tượng trong PHP4 và cuối cùng viết các ứng dụng đẹp mắt theo hướng đối tượng MVC trong các phiên bản PHP5+.</p> <p>Cú pháp PHP có thể thay đổi giữa các phiên bản, nhưng tính tương thích ngược nói chung là rất tốt. Thật không may, điều này dẫn đến một vấn đề: PHP sẽ trở thành một mớ hỗn độn</p>	
Cộng đồng hỗ trợ	<p>PHP có bề dày phát triển trên 20 năm, và đã xây dựng được một cộng đồng sử dụng rộng rãi, khiến mọi vấn đề trong lập trình hầu như đều đã từng có người gặp phải và đã tìm ra được giải pháp xử lý.</p>	<p>Node.js có tài liệu rất tốt nhưng vì nó còn khá mới và có ít sự trợ giúp. JavaScript cũng có tuổi đời như PHP, nhưng phần lớn các trợ giúp chỉ liên quan đến phát triển trong trình duyệt. Điều đó không giúp ích được gì nhiều.</p>
Môi trường	<p>Có nhiều cách để sử dụng PHP trong việc phát triển các ứng dụng desktop và command-line. Nhưng chắc chắn đó không phải là một lựa chọn tốt. Bởi vì, PHP là một công nghệ phát triển phía server. Nó là tốt nhất khi làm công việc đó và hiếm khi vượt ra ngoài những ranh giới này.</p>	<p>JavaScript thường bị coi là nhiều hạn chế hơn. Có một vài công nghệ khác nhưng chức năng chính của nó là chạy trong các trình duyệt. Node.js đã thay đổi nhận thức đó và đã có một sự bùng nổ các dự án JavaScript. JavaScript có thể được sử dụng ở khắp mọi nơi - trong trình duyệt, trên máy chủ, thiết bị đầu cuối, desktop và thậm chí là cả các hệ thống nhúng. Node.js đã khiến cho JavaScript trở nên phổ biến.</p>
Tích hợp	<p>Các công nghệ phát triển sẽ bị hạn chế trừ khi chúng có thể tích hợp với các cơ sở dữ liệu và driver. PHP rất mạnh trong lĩnh vực này. Nó đã được phát triển trong nhiều năm và các hệ thống mở rộng của nó cho phép tương tác trực tiếp với một loạt các API từ phổ biến</p>	<p>Node.js đang phát triển rất nhanh, nhưng người dùng có thể phải rất vất vả để tìm thấy các thành phần tích hợp đủ độ chín cho các công nghệ cũ và ít phổ biến hơn.</p>

	đến ít người biết đến.	
Triển khai	Việc triển khai một ứng dụng mới lên một máy chủ web thì rõ ràng kẻ chiến thắng là PHP. Phần lớn các công ty web hosting đều hỗ trợ PHP. Người dùng sẽ có thể được khuyến mãi thêm MySQL với một mức giá rẻ hơn. PHP được xem là dễ dàng hơn để thử nghiệm và có thể giúp giảm thiểu được nhiều rủi ro.	Node.js có cách tiếp cận hoàn toàn khác và các ứng dụng server-side có thể hoạt động một cách lâu dài. Người dùng sẽ cần một real/virtual/cloud hoặc môi trường máy chủ chuyên biệt, lý tưởng với truy cập SSH root. Đó là một yêu cầu quá cao cho một số nhà cung cấp, đặc biệt là trên shared hosting. Node.js hosting sẽ trở nên đơn giản hơn, nhưng nó sẽ chẳng bao giờ được dễ dàng như việc dùng FTP tải lên một vài tập tin PHP.
Hiệu suất	PHP không quá nặng nề và có những dự án và các tùy chọn làm cho nó nhanh hơn. Ngay cả các nhà phát triển PHP đòi hỏi khắt khe nhất cũng hiếm khi lo lắng về tốc độ.	Hiệu suất của Node.js nói chung là tốt hơn PHP. Tất nhiên, hiệu suất phần lớn là kết quả của kinh nghiệm và sự chăm sóc được thực hiện bởi đội ngũ phát triển.

Bảng 2 So sánh giữa PHP và NodeJS



Hình 6 So sánh tốc độ làm việc của PHP và NodeJS

(Trích: <http://maxprog.net.pl/node-js/node-js-vs-php/>)

Từ những so sánh trên, có thể rút ra rằng NodeJS là một môi trường tốt để tạo nên một server. Tuy nhiên với đặc thù về môi trường và triển khai, đặc biệt phổ biến là hosting, đi kèm với cộng đồng hỗ trợ rất lớn, PHP trở thành một lựa chọn tốt hơn để phát triển

các dự án không cần quá nhiều tài nguyên và không yêu cầu chạy real-time như ứng dụng đề tài đang xây dựng.

4.3. Cơ sở dữ liệu

Hiện nay có rất nhiều hệ quản trị cơ sở dữ liệu trên thị trường, cung cấp các tính năng khác nhau từ cơ bản đến nâng cao, và ngày càng có hiệu suất cao, hỗ trợ mở rộng và có cộng đồng hỗ trợ tốt. Có hai ứng cử viên có thể được nhắc đến là MySQL đại diện cho công truyền thống theo hướng SQL và MongoDB đại diện cho các công nghệ mới hơn và là con đường của NoSQL.

MySQL là một hệ quản lý cơ sở dữ liệu quan hệ mã nguồn mở phổ biến (RDBMS) được phát triển, phân phối và hỗ trợ bởi Oracle Corporation. Giống như các hệ thống quan hệ khác, MySQL lưu trữ dữ liệu trong bảng và sử dụng ngôn ngữ truy vấn có cấu trúc (SQL) để truy cập cơ sở dữ liệu. Trong MySQL, người dùng định nghĩa sơ đồ cơ sở dữ liệu dựa trên yêu cầu và thiết lập các quy tắc để điều chỉnh mối quan hệ giữa các trường trong bảng. Bất kỳ thay đổi nào trong lược đồ đòi hỏi một thủ tục chuyển đổi có thể làm cho cơ sở dữ liệu ngoại tuyến hoặc giảm đáng kể hiệu suất ứng dụng.

(Trích: Viblo^[9])

MongoDB là một cơ sở dữ liệu nguồn mở, không quan hệ được phát triển bởi MongoDB, Inc MongoDB lưu trữ dữ liệu dưới dạng các tài liệu trong một biểu diễn nhị phân được gọi là BSON (Binary JSON). Thông tin liên quan được lưu trữ cùng nhau để truy cập truy vấn nhanh thông qua ngôn ngữ truy vấn MongoDB. Các trường có thể khác nhau từ tài liệu này sang tài liệu khác; không cần khai báo cấu trúc của tài liệu vào hệ thống - các tài liệu tự mô tả. Nếu một lĩnh vực mới cần phải được thêm vào một tài liệu, sau đó lĩnh vực này có thể được tạo ra mà không ảnh hưởng đến tất cả các tài liệu khác trong collection, mà không cần cập nhật một danh mục hệ thống trung ương, và không cần dùng hệ thống ngoại tuyến. Theo tùy chọn, tính hợp lệ của lược đồ có thể được sử dụng để thực thi kiểm soát quản trị dữ liệu đối với mỗi collection.

(Trích: Viblo^[9])

So sánh hiệu suất của MongoDB vs MySQL là rất khó, vì cả hai hệ thống quản lý đều cực kỳ hữu ích và sự khác biệt cốt lõi là cơ sở của hoạt động cơ bản và cách tiếp cận ban đầu. Cả hai đều là mã nguồn mở và dễ dàng có sẵn, cũng như cả hai hệ thống cung cấp các phiên bản thương mại với rất nhiều các tính năng bổ sung.

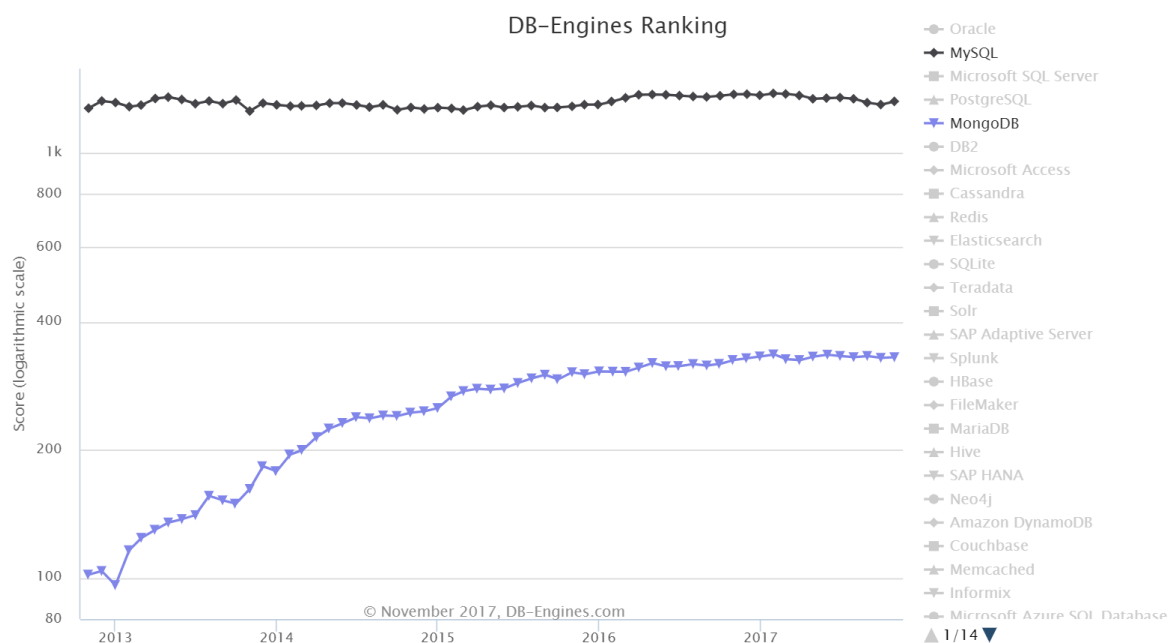
Một trong những điều khác biệt của MongoDB chính là nó không có sự ràng buộc trên thiết kế cơ sở dữ liệu. Người dùng có thể đưa vào các bản ghi khác nhau mà không cần phải quan tâm đến quan hệ giữa các bản ghi này. Ràng buộc duy nhất chính là cấu trúc dữ liệu được hỗ trợ.

Trong khi đó MySQL yêu cầu các định nghĩa rõ ràng đối với các bảng và cột, và trong mỗi hàng đều lưu trữ dữ liệu đã được quy định tại các cột.

Việc này có thể được đánh giá vừa là ưu điểm vừa là khuyết điểm của MongoDB. Ưu điểm được thể hiện khi người dùng không cần quan tâm đến các ràng buộc và tự do lưu trữ. Tuy nhiên, nhược điểm được thể hiện khi các truy vấn sẽ khó khăn hơn trong việc quản lý khi không có các mối tương quan.

MongoDB dùng ngôn ngữ phi cấu trúc để thực hiện câu lệnh. Để tạo một câu lệnh JSON cho MongoDB, người dùng cần thiết phải định nghĩa thông tin với các mục cần có trong kết quả trả về mà người dùng muốn.

MySQL dùng ngôn ngữ cấu trúc SQL để giao tiếp với cơ sở dữ liệu. Và việc này khá đơn giản, vì ngôn ngữ này thực sự mạnh, và bao gồm hai phần: ngôn ngữ định nghĩa dữ liệu (DDL) và ngôn ngữ thao tác dữ liệu (DML).



Hình 7 So sánh điểm xếp hạng giữa MySQL và MongoDB

(Trích: <https://dzone.com/articles/comparing-mongodb-amp-mysql>)

Điều giúp cho MySQL trở nên có ưu thế hơn chính là nằm ở cộng đồng và sự tương thích rộng rãi. Đặc biệt với khuôn khổ của đề tài, MySQL trở nên phù hợp hơn vì tương tác tốt với PHP và cũng khá mạnh mẽ với các kiểu dữ liệu truyền nhận (JSON).

4.4. Phương thức truyền nhận dữ liệu

Với kiến trúc sử dụng PHP Web-server, thì việc dùng POST hoặc GET là sự lựa chọn không thể tranh cãi. Tuy nhiên cần làm rõ ưu nhược điểm của hai hình thức này để có sự lựa chọn phù hợp.

GET và POST là hai phương thức của giao thức HTTP, đều là gửi dữ liệu về server xử lý sau khi người dùng nhập thông tin vào form và thực hiện submit. Trước khi gửi thông tin, nó sẽ được mã hóa bằng cách sử dụng một giản đồ gọi là url encoding. Giản

đồ này là các cặp name/value được kết hợp với các kí hiệu = và các kí hiệu khác nhau được ngăn cách bởi dấu &.

> name=value1&name1=value2&name2=value3

Phương thức GET gửi thông tin người dùng đã được mã hóa thêm vào trên yêu cầu trang:

> http://www.example.com/index.htm?name=value1&name1=value1

Chúng ta thấy rằng GET lộ thông tin trên đường dẫn URL. Bằng thông của nó chỉ khoảng 1024 kí tự vì vậy GET hạn chế về số kí tự được gửi đi. GET không thể gửi dữ liệu nhị phân, hình ảnh ... Có thể cached và được bookmark (đánh dấu trên trình duyệt). Lưu trong browser history.

Phương thức POST truyền thông tin thông qua HTTP header, thông tin này được mã hóa như phương thức GET. Dữ liệu được gửi bởi phương thức POST rất bảo mật vì dữ liệu được gửi ngầm, không đưa lên URL, bằng việc sử dụng Secure HTTP, bạn có thể chắc chắn rằng thông tin của mình là an toàn. Parameters được truyền trong request body nên có thể truyền dữ liệu lớn, hạn chế tùy thuộc vào cấu hình của Server. Không cache và bookmark được cũng như không được lưu lại trong browser history. POST không có bất kì hạn chế nào về kích thước dữ liệu sẽ gửi, có thể gửi dữ liệu nhị phân, hình ảnh.

Lưu trữ (cache) dữ liệu gửi bằng phương thức GET sẽ được lưu trữ lại trong query string và có thể được xem trong lịch sử trình duyệt. Ngược lại thì dữ liệu và địa chỉ URL của các request gửi bằng POST không được trình duyệt lưu lại. Tốc độ GET nhanh hơn rất nhiều so với POST về quá trình thực thi vì dữ liệu gửi đi luôn được webbrowser cached lại, khi dùng phương thức POST thì server luôn thực thi và trả kết quả cho client, còn dùng GET thì web browser cached sẽ kiểm tra có kết quả tương ứng đó trong cached chưa, nếu có thì trả về ngay mà không cần đưa tới server. Đánh dấu (bookmark) đối với request gửi bằng phương thức GET người dùng có thể bookmark lại được trên trình duyệt. Ngược lại các request gửi bằng POST sẽ không thể bookmark được. Gửi lại form với form gửi đi bằng phương thức GET người dùng có thể gửi lại bằng cách bấm phím F5 hoặc Ctrl + R. Tuy nhiên với phương thức POST, nếu người dùng muốn thực hiện việc gửi lại dữ liệu của form thì trình duyệt sẽ hiển thị một hộp thoại cảnh báo. Trở lại trang trước trong trường hợp người dùng đã gửi form dữ liệu đi rồi sau đó bấm phím Backspace để quay lại trang trước thì với phương thức GET người dùng sẽ vẫn được cùng một nội dung (chứa form). Ngược lại với POST thì sẽ thấy một trang trống. Bảo mật Phương thức POST bảo mật hơn GET vì dữ liệu được gửi ngầm, không xuất hiện trên URL, dữ liệu cũng không được lưu lại trong khi đó với GET thì có thể hiển thị lại được các dữ liệu này. Dữ liệu Phương thức POST không giới hạn dung lượng dữ liệu gửi đi cũng như loại dữ liệu (văn bản thông thường hay file nhị phân như upload tập tin hay hình ảnh, video...). Ngược lại, với phương thức GET dữ liệu gửi đi bị giới hạn sử dụng các ký tự chữ có trong bộ ký tự

ASCII. Đồng thời dữ liệu của GET được gửi trong URL thông qua query string nên sẽ bị giới hạn bởi số lượng ký tự tối đa cho phép trong URL.

Với các dữ liệu được gửi nhận bao gồm các thông tin đăng nhập, dữ liệu cá nhân, hay thông tin thanh toán, việc dùng phương thức POST trở nên tốt hơn, ít nhất là vì dữ liệu không thể hiện quá rõ trong địa chỉ truy cập khi có truy vấn từ phần mềm.

Với sự lựa chọn PHP tương tác với React Native như đã phân tích, dữ liệu truyền nhận có thể được sử dụng dưới cấu trúc XML hoặc JSON.

XML

```
<empinfo>
  <employees>
    <employee>
      <name>James Kirk</name>
      <age>40</age>
    </employee>
    <employee>
      <name>Jean-Luc Picard</name>
      <age>45</age>
    </employee>
    <employee>
      <name>Wesley Crusher</name>
      <age>27</age>
    </employee>
  </employees>
</empinfo>
```

JSON

```
{ "empinfo" :
  {
    "employees" : [
      {
        "name" : "James Kirk",
        "age" : 40,
      },
      {
        "name" : "Jean-Luc Picard",
        "age" : 45,
      },
      {
        "name" : "Wesley Crusher",
        "age" : 27,
      }
    ]
  }
}
```

Hình 8 So sánh cấu trúc của XML và JSON

XML là viết tắt của từ eXtensible Markup Language, hay còn gọi là ngôn ngữ đánh dấu mở rộng do W3C đề nghị với mục đích tạo ra các ngôn ngữ đánh dấu khác. Đây là một tập hợp con đơn giản có thể mô tả nhiều loại dữ liệu khác nhau nên rất hữu ích trong việc chia sẻ dữ liệu giữa các hệ thống. Ví dụ khi bạn xây dựng một ứng dụng bằng C# và một ứng dụng bằng PHP thì hai ngôn ngữ này không thể hiểu nhau, vì vậy ta sẽ sử dụng XML để trao đổi dữ liệu.

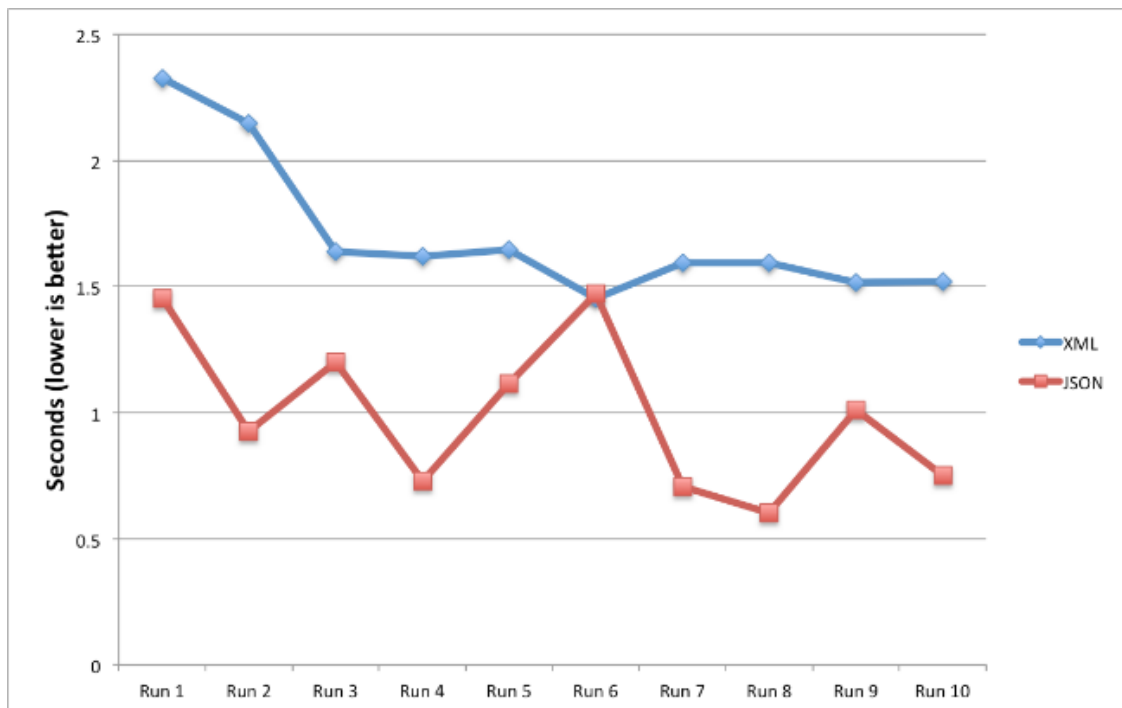
Tất cả những đặc tả dữ liệu XML đều phải tuân theo quy luật và cú pháp của nó nên hầu như các file XML đều rất nghiêm khắc trong việc biên dịch. Tuy nhiên công nghệ này cần phải được xem xét bởi vì trong quá trình thao tác và truyền dữ liệu nó có tỉ lệ sai sót lên tới 5% - 7%. Con số này không cao nhưng cũng rất đáng để cân nhắc khi sử dụng.

JSON - Viết tắt của JavaScript Object Notation nó là một chuẩn để định dạng dữ liệu, về mặt này, có thể so sánh JSON với XML YAML... Nhưng khi JSON đi với JavaScript hoặc ActionScript thì nó có tính ưu việt hơn hẳn.

Tại sao JSON có liên quan đến JavaScript, ActionScript. Đơn giản là vì dữ liệu được định dạng JSON chính là cách biểu diễn một đối tượng trong các Scripting Language này.

Khi sử dụng JSON với JavaScript hay ActionScript, không cần phải có các bước phân tích phức tạp như đối với XML mà có thể truy vấn trực tiếp giá trị theo tên (khóa) được định nghĩa trong JSON.

JSON cũng như XML, nó chỉ mô tả dữ liệu, không có khả năng trình bày, không có khả năng tự chạy, nói chung là thụ động. Ta chỉ có thể nhận JSON rồi làm gì đó, hoặc ta có gì đó rồi chuyển thành JSON để gửi đi, chứ không thể chạy JSON.



Hình 9 So sánh thời gian xử lý trên thiết bị di động giữa XML và JSON

(Trích: <https://www.infragistics.com/community/blogs/b/torrey-betts/posts/mobile-performance-testing-json-vs-xml>)

Với phân tích trên, JSON là lựa chọn thích hợp để mã hóa dữ liệu truyền nhận với hệ thống đã được xây dựng.

4.5. Ứng dụng trí tuệ nhân tạo

Hiện nay đang có một sự bùng nổ về các ứng dụng phát triển trí tuệ nhân tạo, và không thể không kể đến các framework hỗ trợ cho máy học (Machine learning), hay học sâu (Deep learning).

Tensorflow là một thư viện mã nguồn mở cung cấp khả năng xử lý tính toán số học dựa trên biểu đồ mô tả sự thay đổi của dữ liệu. Tensor được sử dụng khi bạn cần giải quyết các bài toán supervised learning.

Khi thực hành với Tensorflow, sẽ có rất nhiều khái niệm phức tạp. Tuy nhiên chỉ ở mức cơ bản, thì khái niệm quan trọng nhất trong Tensorflow là Tensor.

Yêu cầu

- Phần cứng: RAM ít nhất 4GB (cho laptop), bình thường 8-16GB là ổn. Model càng lớn thì càng cần nhiều RAM
- GPU rời nếu có NVIDIA 7xx trở lên
- OS: Ubuntu 14.04 64 bit (Không nên sử dụng 32 bit vì nhiều module chạy yêu cầu RAM >4 GB)
- Python: 2.7 hoặc 3.4 hoặc 3.5. Trong hướng dẫn này mình sẽ sử dụng python 2.7.
- Package Manager: pip (cho python)
- Git (tùy chọn nhưng nên cài)

Cài đặt

```
sudo apt-get install python-pip python-dev
```

```
sudo pip install https://storage.googleapis.com/tensorflow/linux/cpu/tensorflow-0.9.0-cp27-none-linux_x86_64.whl
```

Test

```
python
import tensorflow as tf
a=tf.constant(10)
b=tf.constant(32)
sess=tf.Session()
print(sess.run(a+b))
```

Vì Tensorflow mô tả lại dòng chảy của dữ liệu thông qua graph nên mỗi một điểm giao cắt trong graph thì được gọi là Node. Điều này quan trọng là vì các Node chính là điểm đại diện cho việc thay đổi của dữ liệu nên việc lưu trữ lại tham chiếu của các Node này là rất quan trọng.

Tensorflow cung cấp một loại dữ liệu mới được gọi là Tensor. Trong thế giới của Tensorflow, mọi kiểu dữ liệu đều được quy về một mỗi được gọi là Tensor hay trong Tensorflow, tất cả các loại dữ liệu đều là Tensor. Vậy nên có thể hiểu được phần nào cái tên Tensorflow là một thư viện mô tả, điều chỉnh dòng chảy của các Tensor.

Với một framework mới mẻ như React native, thì việc hỗ trợ các chức năng trí tuệ nhân tạo sẽ là một điểm mạnh. Và đúng như thế, React Native cho phép người dùng tích hợp Tensorflow như là một plugin trong gói chương trình, cho phép tải (load) các dữ liệu đã được train trên máy tính và chạy kết quả này với dữ liệu trên thiết bị di động.

Thư viện có tên “react-native-tensorflow” có sẵn trong gói npm, cho phép việc cài đặt chỉ cần được thực hiện chỉ qua một câu lệnh. Người dùng sử dụng các lệnh được cung

cấp để tải các tệp tin model (.pb) và tệp tin nhãn (label) (.txt) và có thể thực hiện ngay câu lệnh recognize trên thiết bị di động.

Cài đặt bằng câu lệnh: `npm install --save react-native-tensorflow`

Chương 5. Quy trình thực hiện

5.1. Thiết kế cơ sở dữ liệu

5.1.1. Phân tích rõ yêu cầu

Khách truy cập vào phần mềm sẽ tra cứu được các nhà cho thuê, nhà bán, đất cho thuê, đất bán, phòng cho thuê.

Các mục trên bao gồm các thông tin: mã tin, tên tiêu đề, nội dung tin, địa chỉ (Tỉnh/TP, Quận/Huyện, Phường/Xã), một hoặc nhiều hình ảnh mô tả, diện tích, giá, độ ưu tiên, tình trạng duyệt tin, ngày đăng.

Ngoài các thông tin cơ bản trên, mỗi loại tin sẽ bao gồm các thông tin khác như sau:

- Nhà cho thuê bao gồm: Số tầng, số phòng ngủ, số WC, số phòng khách, hướng nhà, chiều ngang nhà, chiều rộng đường.
- Nhà bán bao gồm: Số tầng, số phòng ngủ, số WC, số phòng khách, hướng nhà, chiều ngang nhà, chiều rộng đường.
- Đất cho thuê bao gồm: Loại đất (Đất ở tại đô thị, đất ở tại nông thôn, đất cây lâu năm, đất nông nghiệp, đất trồng các cây hàng năm khác, đất trồng các cây lâu năm khác, đất chăn nuôi), chiều rộng đường.
- Đất bán bao gồm: Loại đất (Đất ở tại đô thị, đất ở tại nông thôn, đất cây lâu năm, đất nông nghiệp, đất trồng các cây hàng năm khác, đất trồng các cây lâu năm khác, đất chăn nuôi), chiều rộng đường.
- Phòng cho thuê bao gồm: phòng ở tầng mấy, nhà vệ sinh riêng hay chung, phòng mặt tiền hay không, giờ đóng cửa, giá điện, giá nước, giá giữ xe, phí vệ sinh, tổng các giá khác, số người tối đa ở được.

Để được đăng thông tin bài viết lên hệ thống, khách cần đăng ký làm thành viên. Khi đăng ký, cần khai báo các thông tin bao gồm Tên hiển thị, số điện

thoại, địa chỉ email, mật khẩu, hình đại diện, là môi giới hay cá nhân, địa chỉ. Mỗi thành viên có một tài khoản nội bộ để nâng cấp độ ưu tiên của bài viết.

Ngoài việc được đăng các thông tin đã nêu trên, thành viên môi giới còn được đăng thông tin dự án để quảng bá cho sản phẩm của công ty. Các thông tin về dự án cần khai báo bao gồm Tiêu đề dự án, hình đại diện cho dự án, nội dung chi tiết dự án, các hình ảnh của dự án, địa chỉ dự án, thời gian mở bán, mức giá dự kiến thời gian mở bán. Các thông tin dự án này sẽ được điều hành viên duyệt và sẽ được hiển thị nếu hợp lệ và không trùng lặp.

Điều hành viên là nhân viên tham gia vận hành hệ thống, có nhiệm vụ duyệt, sửa, xóa thông tin được đăng lên, cộng tiền vào tài khoản của thành viên khi xác nhận được thao tác nạp tiền.

5.1.2. Phân tích cơ sở dữ liệu

Các thông tin về Khách không cần được lưu trữ. Còn lại các thông tin về thành viên và điều hành viên đều thuộc về quản lý thành viên. Thành viên được quản lý bao gồm các thông tin: Mã thành viên (Để quản lý thành viên), Tên hiển thị, địa chỉ email, số điện thoại, loại thành viên (quy bằng mã: 0 là cá nhân, 1 là môi giới, 2 là điều hành viên), hình đại diện, mật khẩu.

Các thông tin về BĐS được lưu trữ chung với nhau, gọi là bài đăng, bao gồm các thông tin: Mã tin, tên tiêu đề, nội dung tin, địa chỉ (Tỉnh/TP, Quận/Huyện, Phường/Xã), một hoặc nhiều hình ảnh mô tả, diện tích, giá, độ ưu tiên, tình trạng duyệt tin, ngày đăng. Vì có thể có nhiều hình ảnh của cùng một tin đăng, cần có một bảng riêng để lưu trữ thông tin về hình ảnh này, gọi là bảng thông tin hình ảnh, bao gồm các thông tin: Mã hình, mã tin, địa chỉ hình ảnh.

Để đảm bảo tính chất mở rộng được, loại bài đăng cần được lưu trữ vào cơ sở dữ liệu chứ không nên tạo thành các bảng riêng. Như thế cần phải có một bảng lưu trữ loại bài đăng, gọi là loại bài đăng, bao gồm các thông tin: Mã loại tin, tên loại tin.

Vì mỗi loại bài đăng có các thuộc tính cần lưu trữ khác nhau, nên các thuộc tính này cần được lưu trữ vào một bảng gọi là thuộc tính, bao gồm các thông tin: Mã thuộc tính, mã loại tin, tên thuộc tính.

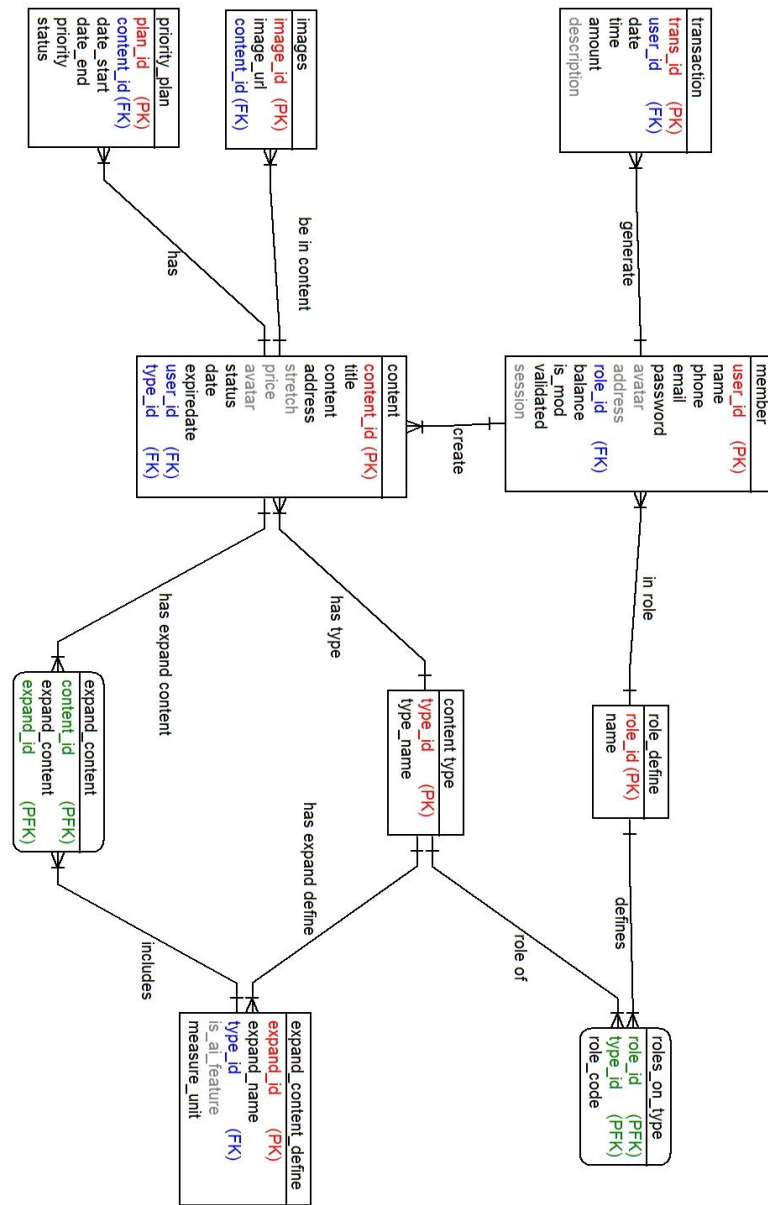
Như thế, cần có một bảng lưu các thông tin riêng biệt của các bài đăng, của các loại bài đăng khác nhau, gọi là thông tin bổ sung bài đăng, bao gồm các thông tin: Mã bổ sung, mã tin, mã thuộc tính, nội dung.

Như thế, ta có 4 bảng để lưu trữ thông tin về bài đăng.

Ngoài ra với thông tin dự án, cũng sẽ cần một bảng thông tin dự án, lưu trữ các thông tin: Mã tin, tên tiêu đề, hình đại diện cho dự án, nội dung chi tiết dự án, các hình ảnh của dự án, địa chỉ dự án, thời gian mở bán, mức giá dự kiến thời gian mở bán, tình trạng duyệt tin.

Về thông tin tài khoản nội bộ (tiền), cần có một bảng gọi là tài khoản tiền, bao gồm các thông tin: Mã giao dịch, mã thành viên, số tiền (dương hoặc âm), nội dung.

5.1.3. Thiết kế chi tiết cơ sở dữ liệu

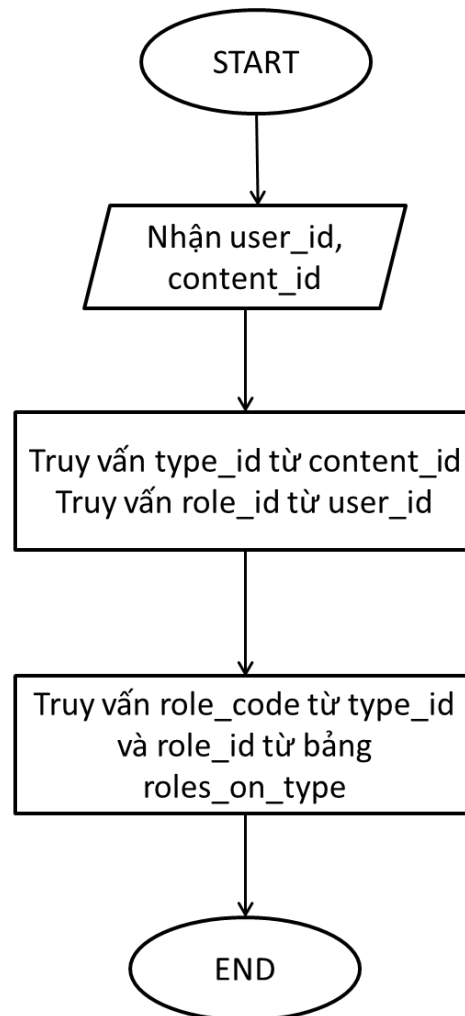


Hình 10 Thiết kế chi tiết cơ sở dữ liệu

- a. Phân tích các bài toán trên lược đồ quan hệ giúp cho bài toán:
- Bài toán phân quyền:

Quyền trên hệ thống được phân ra hai loại, là quyền trên loại bài viết và quyền trên thông tin thành viên. Việc quản lý quyền trên thông tin thành viên có thể được thực hiện dễ dàng qua một cột quy định việc có hay không quyền này của từng thành viên.

Đối với việc quản lý quyền trên loại bài viết, thiết kế trên đảm bảo việc phân quyền diễn ra một cách chủ động, và được thiết kế để có thể mở rộng một cách dễ dàng do người dùng chủ động thực hiện, không cần có sự tham gia của lập trình viên. Role_Code được quy định trên các con số 1, 2, 3, tương ứng với quyền được xem, được sửa, hay được vừa xem vừa sửa thông tin này.



Hình 11 Sơ đồ khối mô tả thuật toán phân quyền thành viên trên loại bài viết

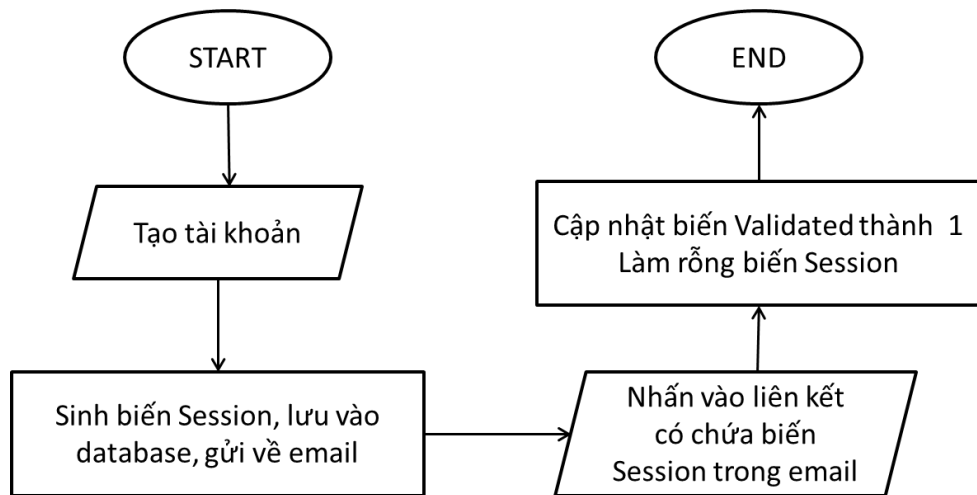
- Bài toán xác thực:

Session là một biến thuộc bảng member, và có nội dung thay đổi thường xuyên qua hoạt động của thành viên, và nội dung được tạo ngẫu nhiên bao gồm 7 ký tự chữ và số, viết hoa và viết thường.

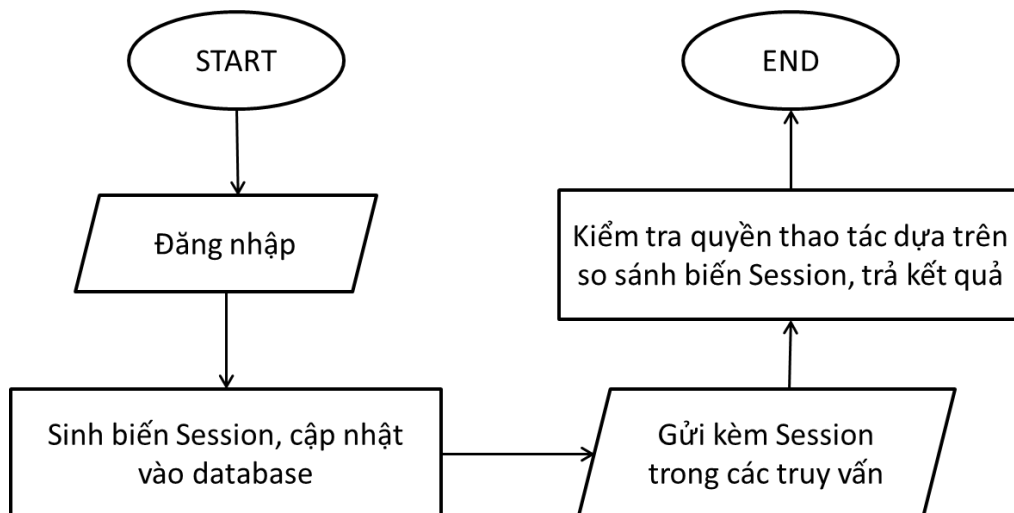
Biến session được sử dụng trong nhiều công đoạn khác nhau trong quá trình sử dụng của người dùng. Giá trị của biến session lần đầu được tạo là khi

người dùng tạo tài khoản, biến session được tạo giá trị ngẫu nhiên, và được dùng làm mã xác thực email thành viên đăng ký.

Kể từ khi tài khoản được xác thực, mỗi lần đăng nhập dựa trên email và mật khẩu thành công, biến session sẽ được sinh lại để đảm bảo truy cập của người dùng hiện tại là duy nhất và không thể bị giả mạo. Khi đó các thao tác cùng tài khoản tại các thiết bị khác không thể được thực hiện khi mã xác thực trong gói tin không trùng với giá trị session hiện tại.



Hình 12 Sơ đồ khối mô tả thuật toán đăng ký và xác thực



Hình 13 Sơ đồ khối mô tả thuật toán đăng nhập và gửi yêu cầu cần quyền thành viên

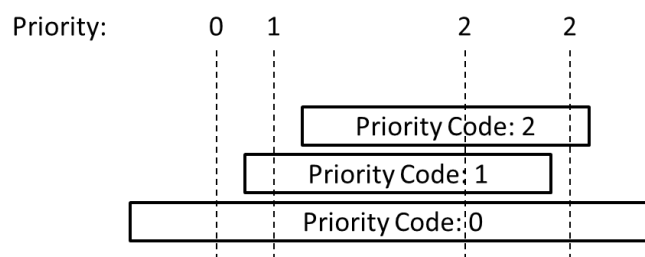
- Bài toán thanh toán và ưu tiên bài viết:

Mỗi bài viết có các độ ưu tiên khác nhau để dành lấy vị trí tốt khi có ưu cầu tìm kiếm bài viết của những người dùng. Ngoài ra, độ ưu tiên này có thể

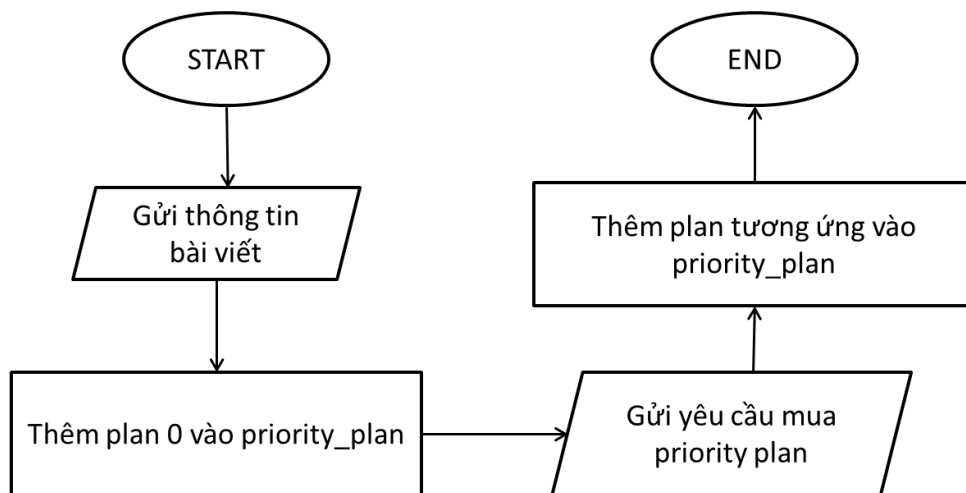
thay đổi theo thời gian, dựa vào thao tác nâng cấp bài viết của thành viên, thông qua thao tác thanh toán các gói (plan) này.

Bảng priority_plan sẽ giải quyết vấn đề này khi lưu giữ các plan đã được thanh toán của từng bài viết, và từ đó hệ thống có thể đưa ra được trạng thái ưu tiên hiện tại của bài viết.

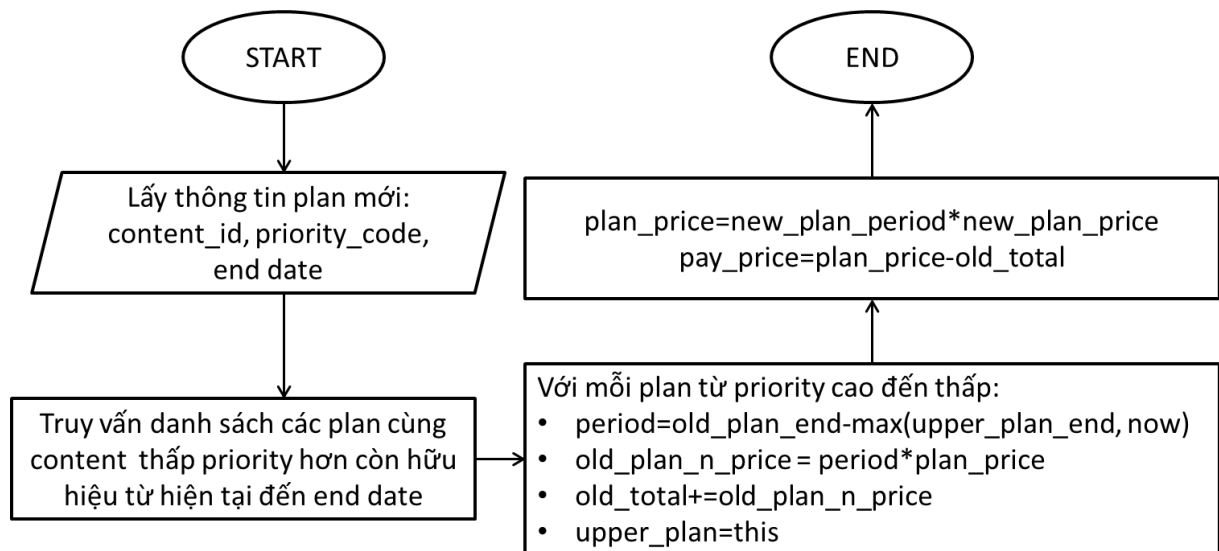
Với một câu lệnh ta có thể truy vấn được thông tin này bằng cách lọc ra các plan còn hạn trong ngày hiện tại với bài viết cần thông tin, và chọn ra mã priority cao nhất. Nếu kết quả trả về là rỗng, ta có được mã priority của bài viết là 0.



Hình 14 Ví dụ về bài toán phân quyền



Hình 15 Sơ đồ khối mô tả thuật toán mua gói ưu tiên bài viết



Hình 16 Sơ đồ khối mô tả thuật toán tính số tiền phải trả khi mua gói mới

- Bài toán quản lý bài viết:

Bài viết được phân ra nằm trong các kiểu bài viết, và tùy theo kiểu bài viết, có thể định nghĩa được các nội dung mở rộng cho kiểu bài viết đó, và các nội dung này được lưu trữ tại một bảng và có thể truy vấn để cung cấp thông tin cho bài viết. Ngoài ra cũng có mục để quy định những thông tin được lưu trữ để sử dụng vào việc nhận dạng trí tuệ nhân tạo, và được xem là kiểu mở rộng của thông tin bài viết.

Ví dụ, với kiểu các bài viết Nhà bán, ngoài các thông tin cơ bản cần phải có, có thể có thêm các thông tin về màu sắc nhà, số tầng, là các thông tin không được hiển thị trong bài viết, nhưng là mục để quy định sử dụng vào việc tìm kiếm dựa trên hình ảnh.

5.2. Thiết kế giao diện

Ngoài các thông tin về thành viên, đăng nhập, thì các thông tin về bài viết đều phụ thuộc vào kiểu bài viết, vì thế, các nội dung này sẽ được hiển thị tại màn hình đầu tiên của ứng dụng, để người dùng có thể quyết định thao tác trên mục phù hợp.



Hình 17 Thiết kế giao diện – Màn hình chính

Với mỗi kiểu bài viết, người dùng sẽ tìm kiếm với các nội dung khác nhau tại màn hình tiếp theo khi chọn vào kiểu bài viết mong muốn.



Hình 18 Thiết kế giao diện – Màn hình tìm kiếm

Danh sách các bài viết phù hợp sẽ được hiển thị theo thứ tự nhất định (dựa trên ưu tiên của bài viết) trên màn hình tiếp theo sau khi người dùng chọn Tìm kiếm.



Hình 19 Thiết kế giao diện – Màn hình danh sách bài viết

Người dùng có thể tùy chọn để sắp xếp lại các bài viết theo mức giá hay diện tích bất động sản họ quan tâm, qua việc chọn nút tương ứng trong mục Sắp xếp.



Hình 20 Thiết kế giao diện – Màn hình danh sách bài viết – Mục Sắp xếp

Nội dung chi tiết bài viết sẽ được hiển thị đầy đủ khi người dùng chọn bài viết trong danh sách được hiển thị.



Hình 21 Thiết kế giao diện – Màn hình thông tin chi tiết bài viết

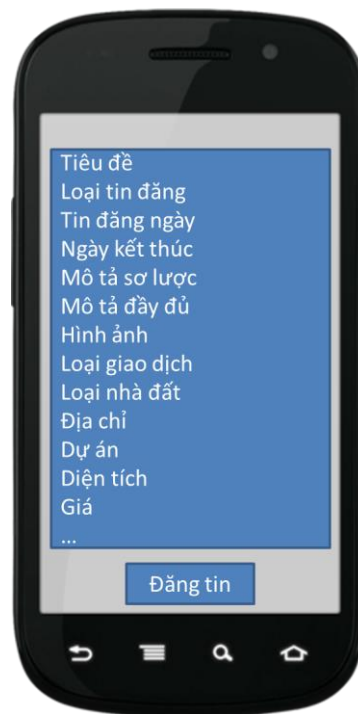
Người dùng có thể liên lạc trực tiếp với người môi giới khi bấm vào nút Gọi, lệnh gọi sẽ được chuyển tiếp qua cho điện thoại xử lý.

Để xem vị trí bất động sản người dùng quan tâm, nhấn vào nút Bản đồ.



Hình 22 Thiết kế giao diện – Màn hình bản đồ

Ngoài ra, để phục vụ việc đăng tin, người dùng cần được cung cấp màn hình đăng tin khi chọn vào nút Đăng tin tại màn hình chính.



Hình 23 Thiết kế giao diện – Màn hình đăng tin

Hay khi người dùng cảm thấy “tin tưởng” một môi giới nào đó khi đã làm việc thành công với họ, người dùng có thể duyệt qua danh sách các môi giới và tìm ra người mà họ biết và liên lạc trực tiếp với họ.



Hình 24 Thiết kế giao diện – Màn hình danh sách môi giới

Người dùng có thể sắp xếp thông tin này theo số tin bài đã đăng thông qua chức năng Sắp xếp.



Hình 25 Thiết kế giao diện – Màn hình danh sách môi giới – Mục Sắp xếp

5.3. Viết chương trình lưu trữ và xử lý thông tin

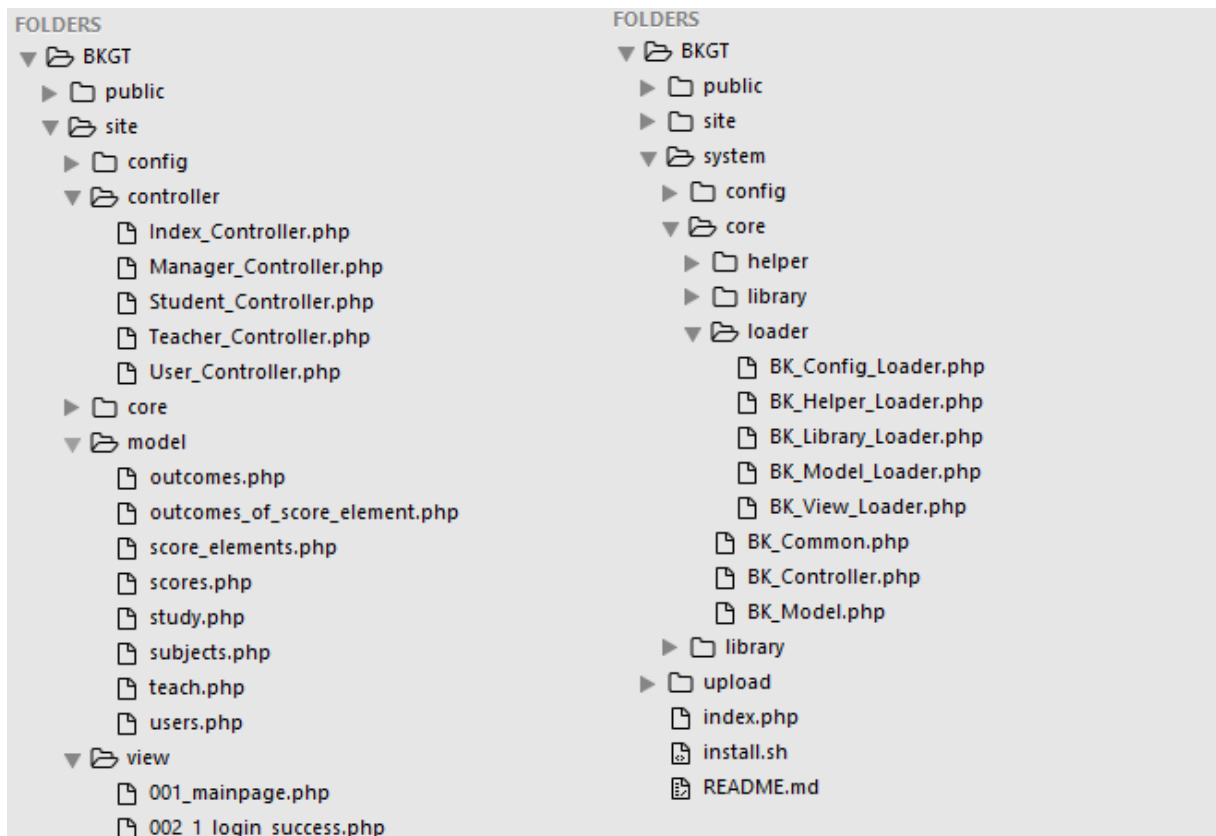
5.3.1. Thiết lập môi trường

Để thiết lập môi trường phát triển chương trình, ta cần một môi trường hỗ trợ ngôn ngữ PHP, một hệ quản trị cơ sở dữ liệu và một chương trình lập trình tốt.

XAMPP là một trong những công cụ mạnh mẽ cho phép thiết lập một môi trường web, hỗ trợ cả Apache, PHP, MySQL cũng như các hệ quản trị khác.

Để thực hiện việc viết chương trình, Sublime Text được chọn vì hỗ trợ tốt các thao tác lập trình từ cơ bản đến nâng cao; hỗ trợ nhận diện rất nhiều ngôn ngữ lập trình; thậm chí cho phép tạo chỉ mục cho phép truy vết (trace) nhanh chóng.

Mô hình MVC được thiết lập như sau:



Hình 26 Tổ chức cấu trúc cho mô hình MVC

5.3.2. Tạo MVC framework

Đoạn mã khởi động sẽ khởi tạo các trình tải config, library, view và model mà chương trình sẽ sử dụng:

```

25      * Hàm khởi tạo
26      *
27      * @desc    Load các thư viện cần thiết
28      */
29      public function __construct()
30      {
31          // Loader cho config
32          require_once PATH_SYSTEM . '/core/loader/BK_Config_Loader.php';
33          $this->config = new BK_Config_Loader();
34          $this->config->load('config');
35
36          // Loader Library
37          require_once PATH_SYSTEM . '/core/loader/BK_Library_Loader.php';
38          $this->library = new BK_Library_Loader();
39
40          // Load View
41          require_once PATH_SYSTEM . '/core/loader/BK_View_Loader.php';
42          $this->view = new BK_View_Loader();
43
44          // Load Model
45          require_once PATH_SYSTEM . '/core/loader/BK_Model_Loader.php';
46          $this->model = new BK_Model_Loader();
47      }
48  }

```

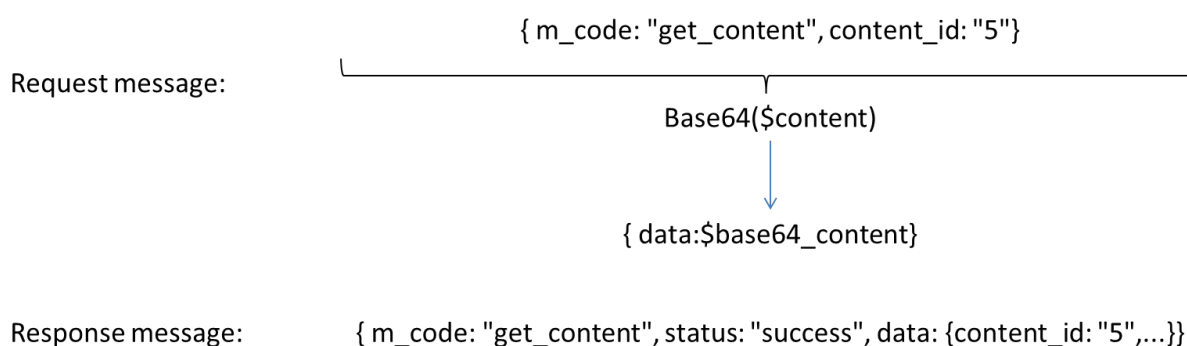
Hình 27 Khởi tạo các trình tải config, library, view và model

5.3.3. Tạo Parser để nhận thông tin truy vấn

Các gói tin được quy định về hình thức chặt chẽ để hạn chế các nguy cơ về an ninh, cụ thể thiết kế được thực hiện như sau:

- Nội dung chính bao gồm các trường dữ liệu: mcode (Message code – Mã của loại gói tin được gửi/nhận), status (Trạng thái trả về: success hoặc error), và data (Dữ liệu chi tiết của các gói tin này).
- Để bảo toàn dữ liệu gửi/nhận tránh được những tác động thay đổi qua các bước encode mặc định khi truyền gói tin, nội dung của gói tin được mã hóa thành dạng Base64, và đặt nội dung này trong một biến data duy nhất gửi về.

Ta có thể biểu diễn qua mô hình sau để có cái nhìn tổng quan về gói tin được truyền/nhận:



Hình 28 Mô tả cấu trúc và quy trình xử lý gói tin truyền nhận

Dựa vào các thiết kế về Cơ sở dữ liệu và giao diện, các gói tin được sử dụng trong hệ thống có thể được liệt kê chi tiết tại bảng sau:

#	Message name	Sender	Receiver	Information						
1	login	Phone	Server	Email	Password					
2	login	Server	Phone	Message status	User ID	Email	Phone	Avatar	Address	...
3	check_login	Phone	Server	User ID	Session Code					
4	check_login	Server	Phone	Message status						
5	logout	Phone	Server	User ID	Session Code					
6	logout	Server	Phone	Message status						
7	register	Phone	Server	Full name	Phone	Email	Password	Avatar	Address	...
8	register	Server	Phone	Message status						
9	validate	Phone	Server	Email	Session Code					
10	validate	Server	Phone	Message status						
11	check_moderator	Phone	Server	Session Code						
12	check_moderator	Server	Phone	Message status						
13	add_moderator	Web	Server	Session	User ID					

		Client		Code						
14	add_moderator	Server	Web Client	Message status						
15	add_content_type	Web Client	Server	Session Code	Type name					
16	add_content_type	Server	Web Client	Message status						
17	get_content_type_list	Phone	Server							
18	get_content_type_list	Server	Phone	Message status	Content type list					
19	add_expand_content_define	Web Client	Server	Session Code	Type ID	Expand Name				
20	add_expand_content_define	Server	Web Client	Message status						
21	get_expand_content_define	Phone	Server	Type ID						
22	get_expand_content_define	Server	Phone	Message status	Expand content define list					
23	add_content	Phone	Server	Session Code	Title	Content	Address	Stretch	Price	Expand Data
24	add_content	Server	Phone	Message status						
25	get_content_list	Phone	Server	Type ID						
26	get_content_list	Server	Phone	Message status	Content list					
27	approve_content	Web Client	Server	Session Code	Content ID					
28	approve_content	Server	Web Client	Message status						
29	add_role_define	Web Client	Server	Session Code	Role name					
30	add_role_define	Server	Web Client	Message status						
31	add_role	Web Client	Server	Session Code	Role ID	Type ID	Role Code			
32	add_role	Server	Web Client	Message status						
33	add_transaction	Web Client	Server	Session Code	User ID	Amount	Description			
34	add_transaction	Server	Web Client	Message status						
35	get_content_list_by_user	Phone	Server	User ID						
36	get_content_list_by_user	Server	Phone	Message status	Content list					
37	get_content_list_owner	Phone	Server	Session Code						
38	get_content_list_owner	Server	Phone	Message status	Content list					
39	get_content	Phone	Server	Session Code (?)	Content ID					
40	get_content	Server	Phone	Message status	Title	Content	Address	Stretch	Price	Expand Data
41	add_priority_plan	Phone	Server	Session Code	Content ID	Expire date	Priority Code			
42	add_priority_plan	Server	Phone	Message status	Plan ID					
43	approve_priority_plan	Web Client	Server	Session Code	Plan ID					
44	approve_priority_plan	Server	Web Client	Message status						

45	get_priority_plan	Phone	Server	Session Code	Plan ID					
46	get_priority_plan	Server	Phone	Message status	Priority Code	Start date	Expire date			

Khi đã xác định đầy đủ các gói tin truyền/nhận, ta thực hiện chương trình đọc dữ liệu gửi về và phân loại gói tin dựa vào Message Code nhận được:

```

public function indexAction()
{
    if ($_POST['data'])
    {
        $encoded_json = $_POST['data'];

        $received_json = base64_decode ($encoded_json);

        header("Content-Type: application/json; charset=UTF-8");

        $received_data = json_decode ($received_json, true);

        $output_data = $this->ProcessInput ($received_data);

        if (DEBUG)
            echo "{ \"input\": \".$received_json.\", \"output\": \"";
        echo json_encode($output_data);

        if (DEBUG)
            echo "}";
    }
}

function ProcessInput($raw_data)
{
    if ($raw_data["mcode"])
    {
        if (isset($raw_data["session"]))
        {
            if ($raw_data["session"] == "")
                return array ( "mcode" => $raw_data["mcode"],
                               "status" => "error",
                               "data" => "session_empty"
                             );
        }

        switch ($raw_data["mcode"]) {
            case "login":
                $email = $raw_data["email"];
                $pass = $raw_data["password"]; // MD5 encoded
                return $this->Login ($email, $pass);
                break;
        }
    }
}

```

Hình 29 Mô tả đoạn mã xử lý yêu cầu nhận được tại server

Tại các hàm xử lý, dữ liệu trả về cần được chuẩn hóa theo như cấu trúc gói tin tại Hình 22.

```

function ApproveContent($session, $content_id)
{
    $user_id = $this->check_valid_session($session);
    if ($user_id == null)
    {
    }
    else
    {
        if ($this->check_moderator($user_id))
        {
            $update_data = array (
                'status' => "1"
            );

            $this->model->update_manual('content', $update_data, 'content_id='.$content_id);

            return array (
                "mcode" => "approve_content",
                "status" => "success"
            );
        }

        return array (
            "mcode" => "approve_content",
            "status" => "permission_denied"
        );
    }

    return array (
        "mcode" => "approve_content",
        "status" => "error",
        "data" => "fail"
    );
}

```

Hình 30 Mô tả đoạn mã xử lý dữ liệu của từng loại tin yêu cầu tại server

5.4. Viết chương trình di động

Dựa trên thiết kế đã được nêu trên, có thể tìm thấy được một vài dự án nhỏ đã được thực hiện trước đó, và do đó, đề tài sẽ phát triển dựa trên một mã nguồn mở React Native với license MIT tại Github. Dự án được phát triển bởi Christine Abernathy, và có bài viết gốc tại địa chỉ:

> <https://www.raywenderlich.com/165140/react-native-tutorial-building-ios-android-apps-javascript>

Các thành phần có thể kế thừa từ dự án này:

- Có các màn hình tìm kiếm, danh sách bài viết, nội dung thông tin bài viết, bản đồ.
- JSON Parser trên React native để nhận dữ liệu sau khi request.

Những việc cần phải thực hiện thêm:

- Bổ sung slide-show cho hình ảnh của bài viết.
- Bổ sung màn hình chính.
- Thiết kế lại cấu trúc JSON Parser để hoạt động với thiết kế của đề tài.
- Bổ sung tính năng trí tuệ nhân tạo trên dự án.

- Chuyển đổi phương thức GET có sẵn thành phương thức POST. Trong quá trình thực hiện đã thể hiện ra khác biệt giữa việc sử dụng các API trong React Native, tạo ra sự mất đồng bộ thông tin khi gửi truy vấn lên máy chủ (fetch).
- Bổ sung tính năng tìm quanh đây cho dự án.

Để cài các gói bổ sung cần thiết trong dự án, ta chạy câu lệnh: `npm install --save`

Cờ `--save` có mục đích lưu lại các gói này vào trong thư mục dự án, để thời gian xử lý khi build và chạy dự án được tối ưu.

Tiếp theo ta có thể bắt đầu dùng Javascript để viết chương trình cho thiết bị di động.

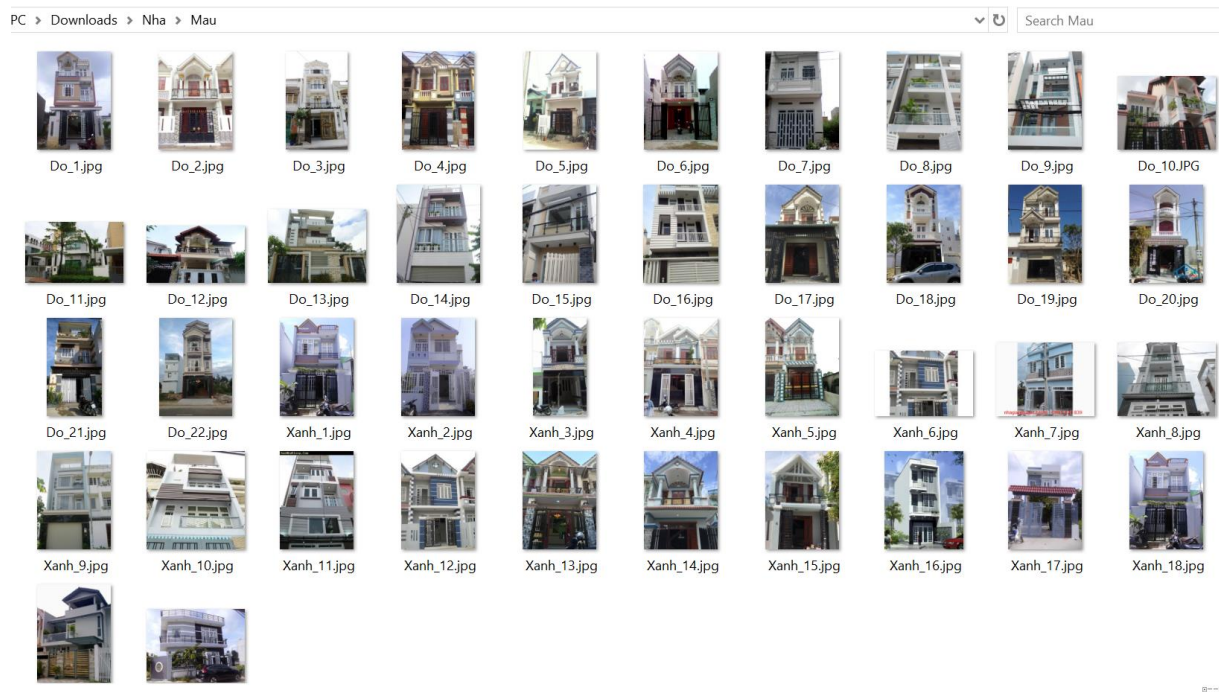
5.5. Thiết kế và áp dụng Tensorflow vào nhận dạng nhà trên thiết bị di động

Để xác định được rõ bài toán cần xử lý, đề tài giới hạn chức năng nhận dạng chỉ trên kiểu bài viết Nhà bán, vì kiểu bài viết này thường xác định được các nội dung cần nhận dạng dễ dàng, và cũng dễ kiểm tra, thử nghiệm nhất.

Đối với một đối tượng nhà, ta có thể phân biệt được dựa vào màu sắc chủ đạo, số tầng nhà, kiểu dáng nhà, ... Tuy nhiên trong giới hạn của bài toán, đề tài xác định hai nhân tố cơ bản và dễ kiểm nghiệm nhất là màu sắc nhà và số tầng của căn nhà, và cũng được giới hạn lại gồm hai loại màu là “Lạnh” hoặc “Ấm”, hai loại tầng là “Trên 1 tầng” và “Từ 1 tầng trở xuống”.

Sau khi thực hiện quá trình nhận dạng tại thiết bị di động, nội dung này sẽ được gửi về server để thực hiện thao tác tìm kiếm theo nhận dạng đã tìm ra, và nội dung tìm kiếm là các nội dung chi tiết của các mở rộng màu sắc và số tầng có cờ `is_ai_feature` được bật lên.

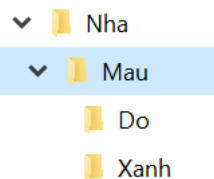
Như thế, ta cần chuẩn bị các tập dữ liệu căn nhà được phân loại theo từng hạng mục, và dùng Tensorflow để training tạo ra tập tin model đã train.



Hình 31 Mô tả tập dữ liệu dùng để training

Người dùng aymericdamien tại Github đã viết một đoạn mã hỗ trợ cho quá trình training từ tập các ảnh, đi kèm đó là hướng dẫn các tùy chỉnh cho phù hợp với tập dữ liệu đã có:

https://github.com/aymericdamien/TensorFlow-Examples/blob/master/examples/5_DataManagement/build_an_image_dataset.py



Hình 32 Mô tả sắp xếp thư mục phục vụ training

Sau khi thực hiện training xong, ta thu được các file checkpoint và file model. Tuy nhiên, để có thể sử dụng được trên thiết bị di động, ta sử dụng công cụ freeze_graph đi kèm Tensorflow để tạo ra graph node (file .pb).

```
bazel build tensorflow/tools:freeze_graph
bazel-bin/tensorflow/tools/freeze_graph \
--input_graph=/tmp/model/my_graph.pb \
--input_checkpoint=/tmp/model/model.ckpt-1000 \
--output_graph=/tmp/frozen_graph.pb \
--output_node_names=output_node \
```

Chuyển qua làm việc với React native, ta cung cấp tệp tin .pb đã tạo ra ở trên cho plugin có tên “react-native-tensorflow” để thực hiện nhận dạng.


```
import { TfImageRecognition } from 'react-native-tensorflow';

const tfImageRecognition = new TfImageRecognition({
  model: require('./assets/tensorflow_inception_graph.pb'),
  labels: require('./assets/tensorflow_labels.txt'),
  imageMean: 117, // Optional, defaults to 117
  imageStd: 1 // Optional, defaults to 1
})

const results = await tfImageRecognition.recognize({
  image: require('./assets/apple.jpg'),
  inputName: "input", //Optional, defaults to "input"
  inputSize: 224, //Optional, defaults to 224
  outputName: "output", //Optional, defaults to "output"
  maxResults: 3, //Optional, defaults to 3
  threshold: 0.1, //Optional, defaults to 0.1
})

results.forEach(result =>
  console.log(
    result.id, // Id of the result
    result.name, // Name of the result
    result.confidence // Confidence value between 0 - 1
  )
)

await tfImageRecognition.close() // Necessary in order to release objects on native side
```

Hình 33 Đoạn mã sử dụng để nhận dạng nhà bằng Tensorflow trên React Native

Khi đã có kết quả từ việc recognize, kết quả được gửi về server để thực hiện tìm kiếm và trả về danh sách kết quả như đã thiết kế. Như thế ta hoàn thành chức năng nhận dạng qua hình ảnh cho ứng dụng di động.

Chương 6. Kết luận chung về giải pháp

6.1. Đề tài và nhiệm vụ đã hoàn thành

6.1.1. Đề tài luận văn

Xây dựng hệ thống đăng tin cho môi giới bất động sản.

6.1.2. Nhiệm vụ hoàn thành

Phân tích nghiệp vụ, các sản phẩm đã có trên thị trường và yêu cầu cho hệ thống cần thực hiện.

Thiết kế giải pháp hệ thống đăng tin môi giới bất động sản đảm bảo mở rộng quy mô khi cần thiết.

Lập thiết kế chi tiết về cơ sở dữ liệu, thiết kế màn hình, thiết kế gói tin của hệ thống.

Thực hiện server và ứng dụng di động theo giải pháp đã lên.

Danh mục tài liệu tham khảo

1. Freetuts.net. Học MVC PHP nâng cao <<https://freetuts.net/hoc-php/hoc-mvc-php-nang-cao>> [Ngày truy cập: 11 tháng 11 năm 2017].
2. Wikipedia. PHP <<https://vi.wikipedia.org/wiki/PHP>> [Ngày truy cập: 18 tháng 12 năm 2017].
3. Wikipedia. SQL <<https://vi.wikipedia.org/wiki/SQL>> [Ngày truy cập: 18 tháng 12 năm 2017].
4. ReactNative.VN. Lập trình mobile với React Native <<http://reactnative.vn/lap-trinh-mobile-app-voi-react-native/>> [Ngày truy cập: 09 tháng 06 năm 2018].
5. Hackermoon. 9 best javascript charting libraries <<https://hackernoon.com/9-best-javascript-charting-libraries-46e7f4dc34e6>> [Ngày truy cập: 10 tháng 12 năm 2017].
6. ASPSnippets. Read, Parse and display CSV (Text) file using JavaScript, jQuery and HTML5 <<https://www.aspsnippets.com/Articles/Read-Parse-and-display-CSV-Text-file-using-JavaScript-jQuery-and-HTML5.aspx>> [Ngày truy cập: 12 tháng 12 năm 2017].
7. Quora. How does React native compare with Google's Java Android development software? <<https://www.quora.com/How-does-React-native-compare-with-Google's-Java-Android-development-software>> [Ngày truy cập: 09 tháng 06 năm 2018].
8. Freetuts. NodeJS là gì <<https://freetuts.net/nodejs-la-gi-584.html>> [Ngày truy cập: 10 tháng 06 năm 2018].
9. Viblo. So sánh Mongo mà MySQL <<https://viblo.asia/p/so-sanh-mongodb-va-mysql-Qbq5QWWEZD8>> [Ngày truy cập: 10 tháng 06 năm 2018].
10. Tensorflow. Preparing models for mobile deployment <https://www.tensorflow.org/mobile/prepare_models> [Ngày truy cập: 11 tháng 06 năm 2018].