



BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ TP. HCM

ĐỒ ÁN TỔNG HỢP NGÀNH

SMARTH DOCTOR

Ngành: CÔNG NGHỆ THÔNG TIN

Chuyên ngành: CÔNG NGHỆ PHẦN MỀM

Giảng viên hướng dẫn :NGUYỄN MẠNH HÙNG

Sinh viên thực hiện :NGUYỄN TRƯỜNG THANH NHÃ

MSSV: 1411060908 Lớp: 15DTH04

TP. Hồ Chí Minh, <2020>

LỜI CẢM ƠN

Đầu tiên em xin gửi lời cảm ơn đến thầy Nguyễn Mạnh Hùng, giảng viên phụ trách hướng dẫn đồ án tổng hợp của em trong học kỳ vừa qua, trong thời gian làm đồ án tổng hợp vừa qua thầy đã góp ý, chỉ bảo tận tình và giúp đỡ chúng em hoàn thành đồ án tổng hợp này.

Em xin chân thành cảm ơn các thầy khoa Công Nghệ Thông Tin, trường Đại Học Công Nghệ thành phố Hồ Chí Minh đã tạo điều kiện, trang bị kiến thức cũng như truyền đạt cho chúng em các kinh nghiệm quý báu. Điều đó góp một phần không nhỏ giúp chúng em có thể hoàn thành đồ án và có kiến thức để làm việc sau này.

Tuy nhiên, do còn hạn chế về kiến thức cũng như kinh nghiệm, bài báo cáo này không thể tránh được những thiếu sót. Em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các quý thầy cô để em có điều kiện bổ sung thêm được nhiều kinh nghiệm và sẽ hoàn thành tốt hơn trong công việc thực tiễn sau này.

Tp. Hồ Chí Minh, ngày ...tháng ... năm 2020

Sinh viên thực hiện

Nguyễn Trương Thanh Nhã

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN



.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

MỤC LỤC

CHƯƠNG 1. TỔNG QUAN.....	9
1.1 Phát biểu đề tài.....	9
1.1.1 Giới thiệu đề tài:	9
1.2 Yêu cầu chức năng.....	9
1.2.1 Đăng kí.....	9
1.2.2 Đăng nhập.....	10
1.2.3 Tin tức.....	11
1.2.4 Tra cứu bác sĩ.....	11
1.2.5 Tra cứu bệnh viện.....	12
1.2.6 Đặt lịch khám/xét nghiệm.....	13
1.2.7 Đặt câu hỏi, chat với bác sĩ.....	14
CHƯƠNG 2 GIỚI THIỆU CÔNG NGHỆ.....	16
2.1 Giới thiệu về REACT NATIVE	16
2.1.1 REACT NATIVE là gì?	16
2.1.2 Cách hoạt động của REACT NATIVE.....	16
2.1.3 Ưu điểm của REACT NATIVE.....	17
2.1.4 Nhược điểm của REACT NATIVE.....	19
2.2 Giới thiệu về MySQL	20
2.2.1 MySQL là gì?	20
2.2.2 Làm việc với MySQL.....	21
2.3 Firebase	21
2.3.1 Firebase là gì ?	21
2.3.2 Firebase realtime	22
2.3.3 Firebase Authentication	22
2.3.4 Firebase Query.....	23
2.3.5 Firestore.....	24
2.4 Giới thiệu về WEB SERVICES.....	25
2.4.1 WEB SERVICES là gì?.....	25
2.4.2 Đặc điểm của WEB SERVICES.....	26
2.4.3 Kiến trúc của WEB SERVICES.....	27

2.4.4 Ưu điểm và nhược điểm của WEB SERVICES.	28
2.5 Giới thiệu về JavaScript.	29
2.5.1 JavaScript là gì?	29
2.5.2 Điểm khác biệt của JavaScript với các ngôn ngữ khác.....	30
2.5.3 Các thư viện JavaScript đình đám hiện nay.	31
2.5.4 Điều gì khiến JavaScript trở nên vĩ đại.....	31
2.5.5 Khuyết điểm của JavaScript.....	32
CHƯƠNG 3 PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....	33
3.1 Sơ đồ Use Case của ứng dụng	33
3.1.1 Sơ đồ Use Case tổng quát.	33
3.1.2 Sơ đồ Use Case người dùng.	34
3.1.3 Sơ đồ Use Case bác sĩ.	34
3.1.4 Sơ đồ Use Case tìm bác sĩ.	35
3.1.5 Sơ đồ Use Case xem tin tức.	35
3.1.6 Sơ đồ Use Case đặt câu hỏi.....	35
3.1.7 Sơ đồ Use Case đặt lịch khám.	36
3.1.8 Sơ đồ Use Case tra cứu dịch vụ.	36
3.2 Sơ đồ Use Case quản trị viên.	37
3.2.1 Sơ đồ Use Case tổng quát	37
3.2.2 Sơ đồ Use Case chức năng tin tức.....	38
3.2.3 Sơ đồ Use Case chức năng đặt câu hỏi.....	38
3.2.4 Sơ đồ Use Case chức năng đặt lịch khám.	39
3.2.5 Sơ đồ Use Case chức năng dịch vụ.	40
3.2.6 Sơ đồ Use Case chức năng bác sĩ.	41
3.3 Sơ đồ ERD.....	42
3.4 Sơ đồ Activity Diagram.....	42
3.5 Thiết kế cơ sở dữ liệu	43
3.5.1 Bảng dữ liệu Tin Tức.....	43
3.5.2 Bảng dữ liệu Chức Vụ.	43
3.5.3 Bảng dữ liệu Người Dùng.	43
3.5.5 Bảng dữ liệu đặt lịch.	44
3.5.6 Bảng dữ liệu Chat.	45
3.5.8 Bảng dữ liệu đặt lịch.	45

3.5.9 Bảng dữ liệu Xác Nhận Đặt Lịch.	46
3.5.11 Bảng dữ liệu bệnh viện.....	47
3.5.12 Bảng dữ liệu dịch vụ.....	47
3.5.13 Bảng dữ liệu danh mục dịch vụ.	48
CHƯƠNG 4: GIAO DIỆN ỨNG DỤNG.....	49
4.1 Giao diện màn hình khởi động ứng dụng.....	49
4.2 Giao diện màn hình chính.	50
4.3 Giao diện bác sĩ.....	51
4.3.1 Giao diện danh sách bác sĩ.	51
4.3.1 Giao diện thông tin chi tiết bác sĩ.	52
4.4Giao diện tin tức.....	53
4.4.1 Giao diện danh sách tin tức.	53
4.4.2Giao diện chi tiết tin tức.	54
4.5 Giao diện bệnh viện.....	55
4.5.1Giao diện danh sách bệnh viện.	55
4.5.2Giao diện chi tiết bệnh viện.	56
4.6 Giao diện chat.....	57
4.7 Giao diện đặt lịch.....	58
4.8 Giao diện đăng kí, đăng nhập.....	59
4.8.1Giao diện đăng nhập.	60
4.8.2 Giao diện đăng kí.....	61
4.9 Giao diện dịch vụ.....	62
4.9.1Giao diện danh sách dịch vụ.....	62
4.9.2Giao diện chi tiết dịch vụ.....	63
CHƯƠNG 5 TỔNG KẾT	64
5.1 Những điểm còn hạn chế	64
5.2 Những điểm đã đạt được.	64
5.3 Hướng phát triển.....	64
5.4 Tài liệu tham khảo	65

Muc Luc Hình Ảnh

Hình 1 Firebase	21
Hình 2 Authentication	23
Hình 3 Truy vấn Firebase.....	24
Hình 4 Database Firebase	24
Hình 5 Web-Services	25
Hình 6 XML Web Services	26
Hình 7 Usecase tổng.....	33
Hình 8 use case người dùng	34
Hình 9 use case Bác sĩ.....	34
Hình 10 Use case tìm bác sĩ	35
Hình 11 use case tin tức	35
Hình 12 use case đặt câu hỏi	35
Hình 13 use case đặt lịch khám.....	36
Hình 14 use case tra cứu dịch vụ.....	36
Hình 15 use case quản trị viên	37
Hình 16 Use case chức năng tin tức	38
Hình 17 Use case đặt câu hỏi	38
Hình 18 Use case quản lý khám.....	39
Hình 19 Use case chức năng dịch vụ	40
Hình 20 Use case chức năng bác sĩ.....	41
Hình 21 Sơ đồ ERD	42
Hình 22 Activity Diagram.....	42
Hình 23 màn hình khởi động ứng dụng	49
Hình 24 màn hình chính.....	50
Hình 25 Danh sách bác sĩ.....	51
Hình 26 thông tin chi tiết bác sĩ	52
Hình 27 danh sách tin tức.....	53
Hình 28 chi tiết tin tức.....	54
Hình 29 Danh sách bệnh viện	55
Hình 30 Chi tiết bệnh viện	56
Hình 31 Giao diện chat	57
Hình 32 Giao diện đặt lịch	58
Hình 33 đăng kí, đăng nhập.	59
Hình 34 đăng kí, đăng nhập.	60
Hình 35 đăng kí, đăng ký.	61
Hình 36 Giao diện danh sách dịch vụ.....	62
Hình 37 Giao diện danh sách dịch vụ.....	63

Mục Lục Bảng Biểu

Bảng 1 ngôn ngữ lập trình phổ biến	30
Bảng 2 Dữ liệu tin tức.....	43
Bảng 3 Dữ Liệu Chức Vụ	43
Bảng 4 dữ liệu người dùng	43
Bảng 5 dữ liệu câu hỏi	44
Bảng 6 Dữ liệu đặt lịch	44
Bảng 7 Dữ liệu Chat	45
Bảng 8 Dữ liệu chuyên khoa.....	45
Bảng 9 dữ liệu đặt lịch.....	45
Bảng 10 dữ liệu xác nhận đặt lịch.....	46
Bảng 11 dữ liệu bác sĩ.....	46
Bảng 12 Dữ liệu bệnh viện	47
Bảng 13 Dữ liệu dịch vụ	48
Bảng 14 Danh mục dịch vụ	48

CHƯƠNG 1. TỔNG QUAN

1.1 Phát biểu đề tài.

1.1.1 Giới thiệu đề tài:

Sức khỏe là vốn quý nhất của con người, là một trong những điều cơ bản để con người sống hạnh phúc, là nhân tố quan trọng trong việc phát triển kinh tế, văn hoá, xã hội. Thế nhưng, vấn đề chăm sóc sức khỏe cộng đồng của nước ta hiện nay lại đang đối mặt với không ít những khó khăn. Ngoài ra, ngày nay với sự phát triển của ngành Công Nghệ Thông Tin, là một trong những lực vực đóng góp thực tế nhất. Công Nghệ Thông Tin góp mặt trong tất cả lĩnh vực của cuộc sống. Đặc biệt nhất là công cụ hỗ trợ đắc lực nhất trong việc công tác quản lý.

Nhận thấy điều đó, S-Doctor đã ra đời với mục đích giúp đỡ mọi người giảm thiểu những lo lắng về sức khỏe, S-Doctor sẽ hỗ trợ bạn tiếp cận với các chuyên gia y tế nhanh chóng. Tận dụng sức mạnh của công nghệ, S-Doctor sẽ giúp bạn nhận được những lời khuyên y tế đáng tin cậy một cách dễ dàng và tiện lợi mọi lúc, mọi nơi S-Doctor là một phương thức mới giúp mọi người xác định, hiểu rõ vấn đề sức khỏe của mình nhanh chóng, tiện lợi.

1.2 Yêu cầu chức năng.

1.2.1 Đăng kí.

Mô tả:

Nếu khách hàng muốn sử dụng các chức năng của ứng dụng như: (Đặt câu hỏi, tìm kiếm thông tin bác sĩ...) thì khách hàng phải đăng ký thành viên tại giao diện đăng kí.

Quy trình:

Khi người dùng nhấn vào chức năng đăng ký, hệ thống sẽ hiển thị giao diện đăng kí tài khoản và yêu cầu người dùng nhập các thông tin:

- Email
- Mật khẩu
- Số điện thoại.
- Địa chỉ liên hệ.

Tiếp theo người dùng cần nhấn vào button đăng kí để hoàn thành việc đăng kí tài khoản, tại đây xảy ra hai trường hợp:

- Trong trường hợp người dùng nhập đúng và đủ các thông tin đầu vào cho việc đăng kí tài khoản thì hệ thống sẽ lưu thông tin vừa nhập và đồng thời chuyển trang sang giao diện đăng nhập.
- Ngược lại khi người dùng nhập thiếu thông tin hoặc sai định dạng thì sẽ thông báo yêu cầu người dùng nhập lại.

1.2.2 Đăng nhập.

Mô tả:

Khi người dùng muốn sử dụng các dịch vụ của ứng dụng thì người dùng cần đăng nhập tài khoản tại giao diện đăng nhập.

Quy trình:

Để đăng nhập, người dùng cần trước tiên cần có tài khoản, nếu chưa có thì cần phải đăng kí tại giao diện đăng kí tài khoản.

Tại giao diện đăng nhập, hệ thống sẽ yêu cầu người dùng nhập các thông tin sau:

- Email
- Password

Trong trường hợp người dùng nhập đúng và đầy đủ, hệ thống sẽ chuyển qua giao diện chính của ứng dụng.

Nếu người dùng nhập sai hoặc thiếu thì hệ thống sẽ thông báo và yêu cầu người dùng nhập lại.

1.2.3 Tin tức.

Mô tả:

Khi người dùng muốn cập nhật thêm các tin tức về sức khỏe cũng như các chương trình khuyến mãi, thì có thể lựa chọn chức năng này.

Quy trình:

Để có thể sử dụng chức năng tin tức của ứng dụng, người dùng cần đăng nhập bằng tài khoản của mình.

Sau khi đăng nhập, hệ thống sẽ hiển thị danh sách tin tức ở mục tin tức, tin tức hiển thị bao gồm:

- Tiêu đề.
- Hình ảnh minh họa.
- Nội dung ngắn.

Khi nhấn vào tin tức mà người dùng muốn đọc, hệ thống sẽ chuyển qua giao diện chi tiết tin tức và chi tiết tin tức sẽ hiển thị.

1.2.4 Tra cứu bác sĩ.

Mô tả:

Chức năng này giúp người dùng tìm kiếm thông tin bác sĩ để kết nối trao đổi tư vấn đến các vấn đề liên quan đến tình trạng sức khỏe người dùng cần tư vấn.

Quy trình:

Để có thể tìm hiểu về thông tin bác sĩ, người dùng cần đăng nhập để sử dụng được chức năng này.

Sau khi đăng nhập người dùng cần chọn chức năng bác sĩ ở giao diện chính, hệ thống sẽ hiển thị danh sách bác sĩ, người dùng có thể hạn chế số lượng bác sĩ hiển thị bằng cách nhập thông tin bác sĩ mà mình muốn tìm vào ô tìm kiếm. Hệ thống sẽ hiển thị các bác sĩ có tên như thông tin đã nhập.

Trong trường hợp tên bác sĩ không có trong hệ thống, danh sách bác sĩ sẽ trống. Ngược lại sẽ hiển thị danh sách bác sĩ có tên vừa tìm.

Các thông tin sẽ hiển thị trong danh sách gồm:

- Tên bác sĩ.
- Chuyên khoa.
- Hình ảnh.
- Số bệnh nhân đã giúp.

Khi người dùng chọn một bác sĩ trong danh sách thì hệ thống sẽ chuyển sang giao diện chi tiết bác sĩ, các thông tin của bác sĩ đó sẽ được hiển thị là:

- Tên bác sĩ.
- Hình ảnh.
- Chuyên khoa.
- Nơi làm việc.
- Số ca tư vấn.
- Số năm kinh nghiệm.
- Thông tin về bác sĩ.
- Trình độ học vấn.
- Bằng cấp chứng chỉ.

1.2.5 Tra cứu bệnh viện.

Mô tả:

Chức năng này giúp người dùng tìm kiếm thông tin bệnh viện để kết nối trao đổi tư vấn đến các vấn đề liên quan đến tình trạng sức khỏe người dùng cần tư vấn.

Quy trình:

Để có thể tìm hiểu về thông tin bệnh viện, người dùng cần đăng nhập để sử dụng được chức năng này.

Sau khi đăng nhập người dùng cần chọn chức năng đặt lịch ở giao diện chính, hệ thống sẽ hiển thị danh sách các bệnh viện, người dùng có thể hạn chế số lượng bệnh viện hiển thị bằng cách nhập thông tin bệnh viện mà mình muốn tìm vào ô tìm kiếm. Hệ thống sẽ hiển thị các bệnh viện có tên như thông tin đã nhập.

Trong trường hợp tên bệnh viện không có trong hệ thống, danh sách bác sĩ sẽ trống. Ngược lại sẽ hiển thị bệnh viện có tên vừa tìm.

Các thông tin sẽ hiển thị trong danh sách gồm:

- Hình ảnh.
- Tên bệnh viện.

Khi người dùng chọn button chi tiết của một bệnh viện trong danh sách thì hệ thống sẽ chuyển sang giao diện chi tiết bệnh viện, các thông tin của bệnh viện đó sẽ được hiển thị là:

- Hình ảnh
- Tên bệnh viện
- Địa chỉ
- Giới thiệu
- Điện thoại

1.2.6 Đặt lịch khám/xét nghiệm.

Mô tả:

Để tiết kiệm thời gian và chi phí trong việc khám bệnh hay xét nghiệm thì người dùng có thể sử dụng tính năng đặt lịch xét nghiệm /khám bệnh.

Quy trình:

Bước 1: Để tiến hành việc đặt lịch khám, thì người dùng phải tiến hành đăng nhập tài khoản, trong trường hợp chưa có tài khoản thì người dùng cần tiến hành đăng ký. Sau khi đăng ký thành công người dùng quay lại đăng nhập để sử dụng tính năng đặt lịch.

Bước 2: Tại đây người dùng cần chọn nơi khám/ xét nghiệm.

Bước 3: Sau khi chọn phòng khám, hệ thống sẽ cung cấp những dịch vụ xét nghiệm/khám bệnh. Tại đây người dùng cần chọn khám/xét nghiệm muốn làm. Nếu người dùng cần biết thông tin về xét nghiệm, người dùng chỉ cần nhấn chọn vào xét nghiệm hệ thống sẽ hiển thị thông tin chi tiết về xét nghiệm đó.

Bước 4: Tiếp theo người dùng cần chọn thời gian để làm các xét nghiệm, khám bệnh. Sau đó hệ thống sẽ thông báo đến người dùng số tiền thanh toán về các xét nghiệm.

Bước 5 Các thông tin trên sẽ được gửi về phòng khám đã được chọn. Sau đó phòng khám liên hệ xác nhận lại thời gian người dùng cần làm các xét nghiệm, khám bệnh.

Bước 6: Người dùng xem lại lịch đã đặt tại phần lịch sử đặt lịch khám.

1.2.7 Đặt câu hỏi, chat với bác sĩ

Mô tả:

Nếu người dùng có các vấn đề liên quan tới sức khỏe cần được giải đáp. Người dùng có thể đăng nhập vào ứng dụng và chọn chức năng đặt câu hỏi với chuyên gia.

Quy trình:

Để sử dụng được tính năng này đầu tiên người dùng cần đăng nhập vào ứng dụng. Trong lúc đăng nhập, hệ thống sẽ xét vai trò của người dùng.

Nếu vai trò của người dùng là 'user' thì sẽ hiện tên người chat là admin.

Nếu vai trò của người dùng là 'admin' thì sẽ hiện danh sách chat tất cả người dùng.

CHƯƠNG 2 GIỚI THIỆU CÔNG NGHỆ

2.1 Giới thiệu về REACT NATIVE

2.1.1 REACT NATIVE là gì?

React Native là framework giúp lập trình viên viết ứng dụng Native chỉ bằng Javascript. Đúng vậy, chỉ đơn giản là Javascript.

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.

2.1.2 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: ví dụ “1+1=2” :)), 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 đâu nhé.

2.1.3 Ưu điểm của REACT NATIVE

HOT RELOADING! KHÔNG CẦN TỐN THỜI GIAN ĐỂ RECOMPILING

Đó là một trong những tính năng mình thích nhất khi phát triển những ứng dụng React Native. Những ai từng lập trình Java, điều này sẽ cực kì có ý nghĩa.

Thử tưởng tượng khi bạn chỉ thay đổi mỗi String mà ứng dụng lại phải compile lại từ đầu và rồi bạn muốn kiểm tra thì lại phải mở ứng dụng từ màn hình chính và “chọt chọt” để đến được màn hình muốn kiểm tra. mất thời gian phải không? Với React Native thì điều đó sẽ không còn nữa. Nó sẽ update luôn giá trị trên chính màn hình mà bạn thay đổi code, không hề compile lại ứng dụng.

Với Hot Reloading, trạng thái của ứng dụng được duy trì, và nó sẽ reload lại chính màn hình đang sửa code, tất cả ngay trước mắt bạn trong chỉ ít hơn vài giây thôi.

Để bật tính năng này trên emulator thì bạn làm như sau:

Cmd+D- chọn ENABLE DEVICE

THỰC SỰ NATIVE

Không giống những framework khác như Cordova cái mà thường chỉ là một Webview. React Native được sử dụng để xây dựng một ứng dụng Native thật sự. Những Webview không thể cung cấp trải nghiệm chân thật cho người dùng – điều mà chỉ React Native có thể làm được.

Với React Native, những component cơ bản (View, Image, Button, Checkbox) đều là những component native. Do đó ứng dụng sẽ mang một trải nghiệm liền mạch với từng

hệ điều hành. Điều đó thực sự làm lên một khác biệt lớn so với các framework lập trình cross-platform khác.

Nó thực sự ấn tượng hơn nữa khi bạn lại code bằng ngôn ngữ javascript và render ra các component native từ OS.

Mình lấy một ví dụ đơn giản: Đó chính là DatePicker widget. Bạn sẽ thấy ứng dụng của bạn với widget này sẽ hiển thị khác nhau trên Android và IOS.

Đây chính là một đặc điểm tạo nên sự khác biệt: Không chỉ là tốc độ vì nó gần native nhất, mà nó còn làm cho cuộc sống của lập trình viên trở nên dễ dàng hơn. Họ không cần phải code lại một logic cho các nền tảng khác nhau mà vẫn có ứng dụng native cho từng nền tảng đó.

CODE MỘT LẦN CHẠY NHIỀU CHỖ

Expo start

Ví dụ sau đây sẽ đem đến lí do quan trọng tiếp theo rằng tại sao mình thích làm với React Native- Đó là khả năng đa hệ điều hành.

Bạn không cần biết Objective-C, Swift hay Java. Với Javascript và JSX, bạn có thể xây dựng một ứng dụng hoạt động đa hệ điều hành rất tốt.

Từ kinh nghiệm, mình thấy rằng gần 95% code được chia sẻ giữa iOS và Android, và chỉ cần tinh chỉnh nhỏ(tweak) để hoàn thiện sản phẩm cuối cùng cho từng hệ điều hành. Đó có phải là điều bạn hướng tới không? Bạn không cần phải có nhiều team và codebase để hỗ trợ các hệ điều hành khác nhau

Thay vào đó, bạn chỉ cần một nhóm và một codebase làm việc trên ứng dụng mà support cả iOS và Android. Điều này sẽ giúp tiết kiệm rất lớn về thời gian và tiền bạc cho công ty bạn đấy!

Mặc dù mình chưa thử, nhưng React Native cũng hoạt động tốt trên cả hệ điều hành Windows. Nếu bạn đã thử thì comment bên dưới cho mọi người biết nhé

Rõ ràng là càng nhiều càng tốt đúng không?

MỘT CỘNG ĐỒNG HỖ TRỢ RẤT LỚN

Nhiều năm trở lại đây, React Native đang trở lên rất phổ biến, nhiều nhà lập trình đang đóng góp để làm React Native tốt hơn mỗi ngày. Đặc biệt là nó được chống lưng bởi tập đoàn Facebook.

React Native Github repro là một nguồn mở và có hàng nghìn cộng tác viên hoạt động rất năng nổ.

Có một diễn đàn thảo luận mới về React Native và bạn cũng có thể là một phần trong đó nếu muốn.

Stack Overflow là một địa chỉ nữa nơi mà có nhiều tài liệu và câu hỏi đã được trả lời về React Native.

Cộng đồng rất lớn và đang dần phát triển hơn nữa. Nhiều vấn đề đã và đang được giải quyết và bạn sẽ không cần phải tốn thời gian để nghiên cứu lại trong suốt quá trình phát triển của bản thân.

Nói chung, mình nghĩ React Native sẽ trở thành một phần không thể thiếu và chắc chắn có một tương lai rực rỡ trong việc phát triển đa hệ điều hành. Mình hi vọng bạn cũng sẽ xây dựng được một ứng dụng React Native đầu tiên sớm thôi.

Các bạn có thể tham gia khóa học miễn phí về React native mà mình nghĩ các bạn nên tham gia. Hẹn gặp lại ở những bài viết sau và đừng quên để lại comment ý kiến của bạn nhé.

2.1.4 Nhược 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/Objective-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

Dùng ES6: đây là cấu trúc mới cho Javascript từ 2020, 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

2.2 Giới thiệu về MySQL

2.2.1 MySQL là gì?

MySQL là một chương trình dùng để quản lý hệ thống cơ sở dữ liệu (CSDL), CSDL là một hệ thống lưu trữ thông tin! Giống như một file text chứ gì? Không, hoàn toàn không! Điều quan trọng ở đây là CSDL là một hệ thống được sắp xếp, phân lớp rõ ràng ngăn nắp những thông tin mà mình lưu trữ.

Do đó bạn có thể truy cập lấy dữ liệu một cách dễ dàng, MySQL hỗ trợ hầu hết các ngôn ngữ lập trình PHP cũng không phải là ngoại lệ, và tôi cũng có thể nói nó là một cặp bài

trùng trong lập trình WEB:) chính vì lẽ đó mà mã nguồn mở phổ biến nhất hiện nay là WordPress lại sử dụng MySQL làm cơ sở dữ liệu mặc định

2.2.2 Làm việc với MySQL.

Bởi vì truy vấn MySQL dựa trên SQL nên bạn cần phải hiểu SQL là gì?

SQL là gì?

SQL là viết tắt của Structured Query Language, là ngôn ngữ truy vấn mang tính cấu trúc. Nó được thiết kế để quản lý dữ liệu trong một hệ thống quản lý cơ sở dữ liệu quan hệ (RDBMS). SQL là ngôn ngữ cơ sở dữ liệu, được sử dụng để tạo, xóa trong cơ sở dữ liệu, lấy các hàng và sửa đổi các hàng

Chúng tôi giả sử bạn đã cài đặt MySQL và kết nối vào phần điều khiển của nó

2.3 Firebase

2.3.1 Firebase là gì ?

Firebase là nền tảng phát triển ứng dụng dành cho thiết bị di động và web, cung cấp cho nhà phát triển nhiều công cụ và dịch vụ giúp họ phát triển các ứng dụng chất lượng cao, phát triển user base và tìm kiếm thêm lợi nhuận



Hình 1 Firebase

2.3.2 Firebase realtime

Firebase Realtime là một dạng cloud-hosted database cho phép người dùng lưu trữ và đồng bộ trong thời gian thực

Đồng bộ hóa trong thời gian thực giúp người dùng dễ dàng truy cập dữ liệu của họ từ bất kỳ thiết bị nào, có thể là web hoặc thiết bị di động. Cơ sở dữ liệu thời gian thực cũng giúp người dùng có khả năng cộng tác với nhau.

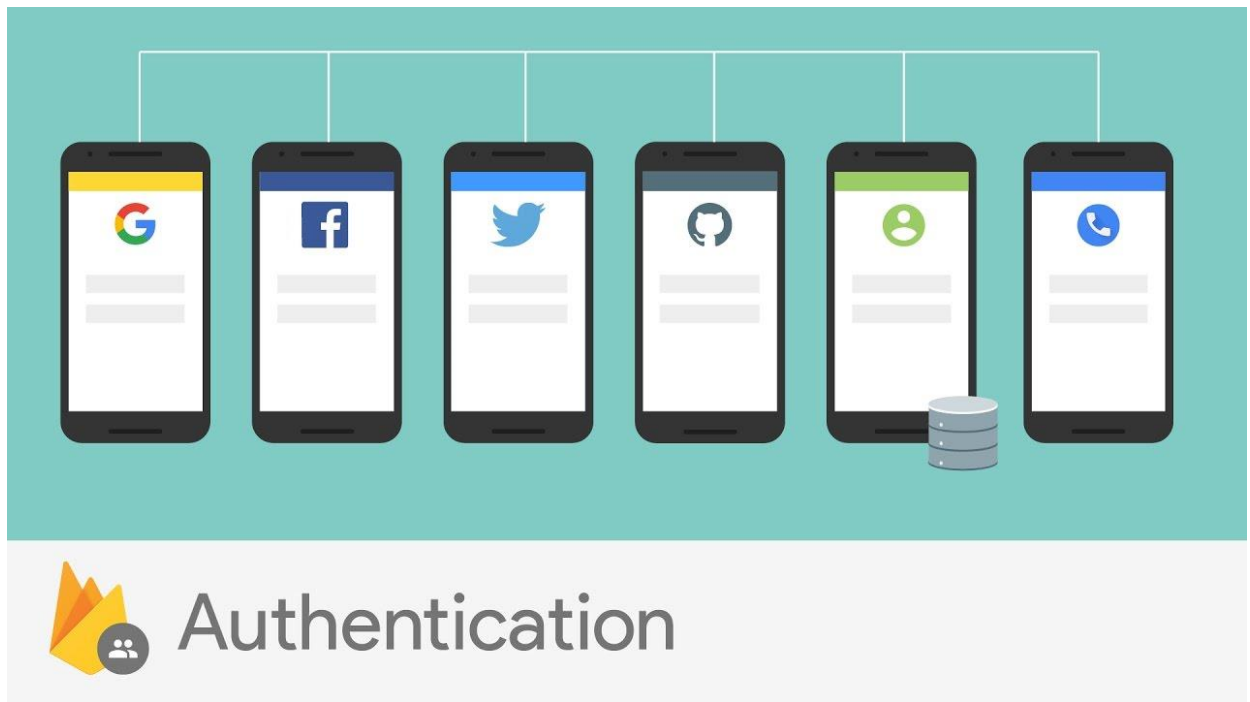
Một lợi ích tuyệt vời khác của Cơ sở dữ liệu thời gian thực là nó chứa mobile và web SDKs, cho phép bạn tạo ứng dụng mà không cần máy chủ.

Khi người dùng của bạn ngoại tuyến, Realtime Database SDKs sử dụng bộ nhớ cache cục bộ trên thiết bị để phân phát và lưu trữ thay đổi. Khi thiết bị trực tuyến, dữ liệu cục bộ sẽ tự động được đồng bộ hóa.

Realtime Database cũng có thể tích hợp với Firebase Authentication để cung cấp quy trình.

2.3.3 Firebase Authentication

- Firebase Authentication cung cấp backend services, easy-to-use SDKs, và ready-made UI libraries để xác thực người dùng cho ứng dụng.
- User có thể xác thực ứng dụng qua các phương pháp sau:
- Email và mật khẩu.
- Số Điện Thoại.
- Facebook.
- Github.
- Microsoft.



Hình 2 Authentication

Xác thực firebase giúp việc xây dựng hệ thống dễ dàng hơn.

2.3.4 *Firebase Query*

Firebase đơn giản hóa quá trình truy xuất dữ liệu. Firebase có nhiều kiểu truy vấn như:

- `StartAt('value')`.
- `endAt('value')`.
- `equalTo('child_key')`.
- `limitToFirst(10)`.
- `limitToLast(10)`.

```
const db = firebase.database();

const firebaseRef = db.child(`child`);

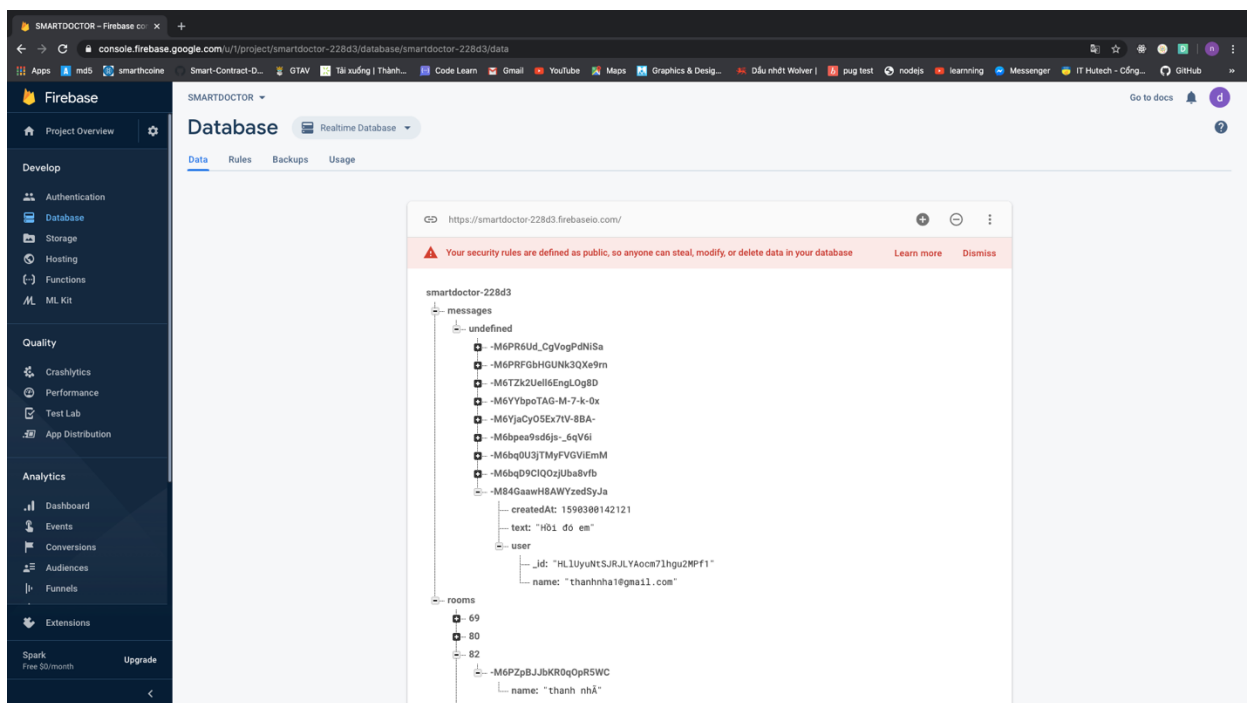
firebaseRef.orderByChild("user").equalTo("GeekyAnts").on("child_added",
  function(snapshot) {
    console.log(snapshot.key);
  }
);
```

Hình 3 Truy vấn Firebase

Firebase truy vấn gồm 2 bước. Đầu tiên tạo một tham chiếu đến khóa cha và sau đó sử dụng hàm tương tự. Cũng có thể nối thêm 1 hàm truy vấn để nâng cao hơn.

2.3.5 Firestore.

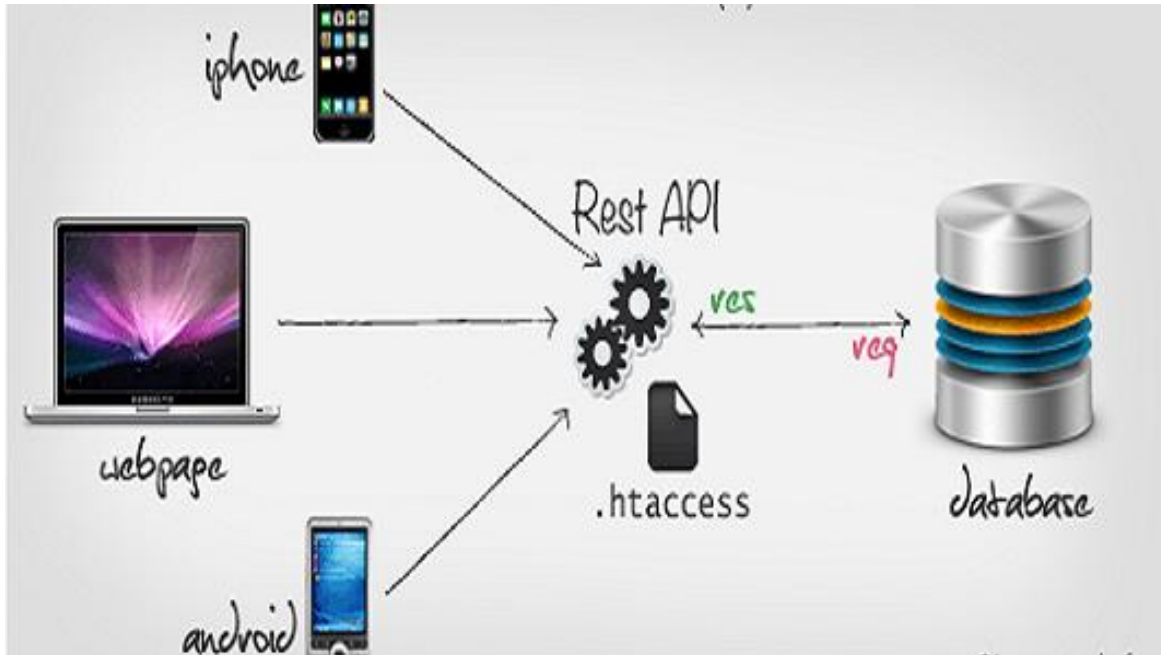
Firestore là một NoSQL document database cho phép bạn dễ dàng lưu trữ, đồng bộ hóa và truy vấn dữ liệu cho các ứng dụng di động và web



Hình 4 Database Firebase

2.4 Giới thiệu về WEB SERVICES.

2.4.1 WEB SERVICES là gì?



Hình 5 Web-Services

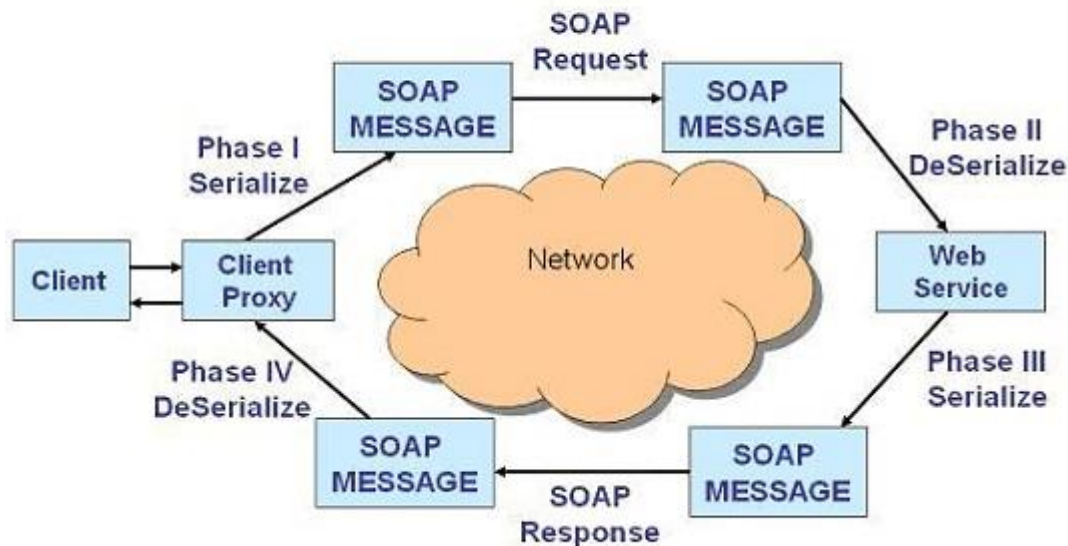
Web services là một phần mềm được xây dựng để hỗ trợ các ứng dụng trên máy tính khác nhau. Chúng được kết nối thông qua mạng internet trên một giao diện chung và được mô tả bằng XML.

Để xác định chính xác Web services bạn thông qua các URL đồng thời thực hiện các chức năng và cung cấp thông tin cần thiết cho người sử dụng.

Kết quả của Web services được thể hiện dưới dạng json hoặc xml bao gồm các Module độc lập của người sử dụng và bản thân nó được ghi trên server. Do đó chúng được sử dụng với mục đích tạo nên các ứng dụng phân tán.

2.4.2 Đặc điểm của WEB SERVICES.

XML Web Services Architecture



Hình 6 XML Web Services

- Đặc điểm nổi bật nhất của Web services là không phụ thuộc vào ngôn ngữ lập trình.
- Khi sử dụng Web services bạn có thể truy cập vào bất kỳ ứng dụng nào và khả năng tích hợp rất lớn. Ví dụ như: dịch vụ chọn lọc và phân loại các tin tức, ứng dụng trong các lĩnh vực du lịch (giá vé, địa điểm, thông tin...), thông tin về giá cả, tỷ giá hối đoái hay các dịch vụ online khác như đặt vé máy bay, xe ô tô..
- Với Web services client và server có thể tương tác với nhau ngay cả trong các môi trường khác nhau. Ví dụ khi bạn cài đặt Web services trên hệ điều hành Linux thì máy chủ nền tảng là Windows vẫn có thể hoạt động dễ dàng, các ứng dụng vẫn có thể chạy và xử lý mà không cần bất cứ yêu cầu đặc biệt nào khác.

- Kỹ thuật của Web service được thiết kế dựa trên mã nguồn mở và phát triển theo các tiêu chuẩn đã được công nhận.
- Web service bao gồm nhiều Module độc lập và được sử dụng rộng rãi trên Internet.
- Web service hoạt động trên nguyên tắc kết hợp từng thành phần trong các lĩnh vực cụ thể, cung cấp lợi ích cao nhất cho các doanh nghiệp, khách hàng thông qua Internet.
- Các ứng dụng trong Web service sẽ hoạt động theo mô hình client-server với các phần mềm ứng dụng phía server như PHP, Microsoft.Net hay Oracle Application server.
- Chi phí để phát triển Web service rất thấp nên phù hợp với tất cả các doanh nghiệp, cá nhân vừa và nhỏ.
- Với Web service việc bảo trì rất dễ dàng và nhanh chóng.

2.4.3 Kiến trúc của WEB SERVICES.

Trong Web service có 3 tầng cơ bản như sau:

- Tầng vận chuyển(Service Transport) với các giao thức chuẩn là HTTP , SMTP và JMS, FTP và gần đây nhất là giao thức BEEP. Chức năng của tầng này là truyền thông tin giữa các ứng dụng mạng.
- Tầng giao thức với giao thức chuẩn là SOAP. Chức năng của tầng này là giải mã các thông tin theo định dạng XML.
- Tầng mô tả dịch vụ (Web Service Description Language) với các giao thức chuẩn là XML và WSDL. Chức năng của tầng này là mô tả giao diện chung của Web Service
- Tầng dịch vụ: Tầng này có các chức năng của service
- Tầng đăng ký dịch vụ: Với các công nghệ chuẩn là UDDI .UDDI. Chức năng của tầng này là giúp người sử dụng có thể đăng ký các dịch vụ từ xa qua mạng.

2.4.4 Ưu điểm và nhược điểm của WEB SERVICES.

Ưu điểm của Web service

- Web service có ưu điểm nổi bật so với các phần mềm khác là làm hệ thống đơn giản, khả năng tích hợp cao hơn. Khi sử dụng Web service chi phí phát triển web sẽ giảm, tốc độ ứng dụng nhanh, khả năng tương tác giữa các doanh nghiệp cao hơn.
- Với việc sử dụng Web service trong lập trình web thì khả năng ứng dụng rất lớn có thể chạy trên nhiều hệ điều hành khác nhau.
- Web service cung cấp khả năng hoạt động rộng lớn với các ứng dụng phần mềm khác nhau chạy trên những nền tảng khác nhau.
- Web service được thiết kế mở và sử dụng các giao thức. Với các giao thức và định dạng dữ liệu trên các văn bản nên các lập trình viên có thể sử dụng dễ dàng.
- Web service giúp nâng cao khả năng tái sử dụng.
- Khi sử dụng Web service các thành phần trong hệ thống sẽ tương tác dễ dàng hơn rất nhiều. Giúp quá trình phát triển các ứng dụng phân tán diễn ra nhanh hơn.

Nhược điểm của Web service

- Vấn đề bảo mật vẫn còn hạn chế, các nhà phát triển Web service đang cố gắng khắc phục hạn chế này.
- Ưu điểm sử dụng các giao thức và chuẩn mở của Web service cũng là hạn chế của nó. Vì có quá nhiều chuẩn mở nên người dùng khó nắm bắt về nó.
- Khi Web service nếu ngừng hoạt động trong một khoảng thời gian nào đó thì giao diện web sẽ không thay đổi, các giao thức để vận hành thiếu trầm trọng và

hệ thống của bạn sẽ bị lỗi nếu không được nâng cấp kịp thời gây ra nhiều thiệt hại lớn.

2.5 Giới thiệu về JavaScript.

2.5.1 JavaScript là gì?

JavaScript là ngôn ngữ lập trình mang đến sự sinh động của website. Nó khác với HTML (thường chuyên cho nội dung) và CSS (thường chuyên dùng cho phong cách), và khác hẳn với PHP (chạy trên server chứ không chạy dưới máy client).

Bạn cần biết gì:

- JavaScript là ngôn ngữ dễ học;
- Nó được phát triển bởi Netscape, và đang được dùng trên 92% webstie;
- JS có thể được gắn vào một element của trang web hoặc sự kiện của trang web như cú click chuột;
- Hoạt động trên đa trình duyệt và đa thiết bị;
- Nhanh và nhẹ hơn các ngôn ngữ lập trình khác;
- Có thể ít an toàn hơn vì độ phổ biến của nó;
- Bạn có thể thêm JavaScript trực tiếp vào HTML hoặc bạn có thể lưu nó trên files riêng biệt và gọi lên khi cần.

Cuối cùng, có một lý do vì sao JavaScript phổ biến đến vậy là: nó làm tốt những gì nó có thể làm. Vậy bạn không cần phải đắn đo nữa khi tìm hiểu về nó đâu! Chúng tôi hy vọng bài viết này đã giúp ích cho bạn và giúp bạn biết JavaScript là gì, cũng như cách thức hoạt động của nó.

2.5.2 Điểm khác biệt của JavaScript với các ngôn ngữ khác.

Lý do vì sao JavaScript là một trong các ngôn ngữ lập trình phổ biến nhất là nó rất linh hoạt. Trên thực tế, có nhiều lập trình viên chọn nó làm ngôn ngữ chính và chỉ sử dụng các ngôn ngữ khác trong danh sách bên dưới nếu nó họ cần dùng điều gì đó đặc biệt.

Hãy xem qua các ngôn ngữ lập trình phổ biến nhất bên dưới:

JavaScript	JavaScript hoặc JS sẽ giúp tăng tính tương tác trên website. Script này chạy trên các trình duyệt của người dùng thay vì trên server và thường sử dụng thư viện của bên thứ 3 nên có thể tăng thêm chức năng cho website mà không phải code từ đầu.
HTML	Viết tắt của “Hypertext Markup Language”, HTML là một trong số các ngôn ngữ lập trình phổ biến nhất trên web và xây dựng nên các khối chính của một trang web. Ví dụ về HTML tags là <p> cho đoạn văn và cho hình ảnh.
PHP	PHP là ngôn ngữ phía server, khác với JavaScript chạy trên máy client. Nó thường được sử dụng trong các hệ quản trị nội dung nền PHP như WordPress, nhưng cũng thường được dùng với lập trình back-end và có thể tạo ra kênh truyền thông tin hiệu quả nhất tới và từ database.
CSS	CSS viết tắt của “Cascading Style Sheets”, nó giúp webmaster xác định styles và định nghĩa nhiều loại nội dung. Bạn có thể làm vậy thủ công với mọi yếu tố trong HTML, nhưng nếu vậy bạn sẽ cứ lặp đi lặp lại thành phần đó mà bạn dùng ở nhiều nơi khác nhau.

Bảng 1 ngôn ngữ lập trình phổ biến

2.5.3 Các thư viện JavaScript đình đám hiện nay.

Những ứng dụng to lớn của Javascript khiến người ta không thể quên nó được. Hiện nay có rất nhiều libraries và framework được viết từ Javascript như:

- AngularJS: Một thư viện dùng để xây dựng ứng dụng Single Page
- NodeJS: Một thư viện được phát triển phía Server dùng để xây dựng ứng dụng realtime
- Sencha Touch: Một Framework dùng để xây dựng ứng dụng Mobile
- ExtJS: Một Framework dùng xây dựng ứng dụng quản lý (Web Applications)
- jQuery: Một thư viện rất mạnh về hiệu ứng
- ReactJS: Một thư viện viết ứng dụng mobie
- Và còn nhiều thư viện khác

2.5.4 Điều gì khiến JavaScript trở nên vĩ đại.

JavaScript có rất nhiều ưu điểm khiến nó vượt trội hơn so với các đối thủ, đặc biệt trong các trường hợp thực tế. Sau đây chỉ là một số lợi ích của JavaScript:

- Bạn không cần một compiler vì web browser có thể biên dịch nó bằng HTML;
- Nó dễ học hơn các ngôn ngữ lập trình khác;
- Lỗi dễ phát hiện hơn và vì vậy dễ sửa hơn;
- Nó có thể được gắn trên một số element của trang web hoặc event của trang web như là thông qua click chuột hoặc di chuột tới;
- JS hoạt động trên nhiều trình duyệt, nền tảng, vâng vâng;
- Bạn có thể sử dụng JavaScript để kiểm tra input và giảm thiểu việc kiểm tra thủ công khi truy xuất qua database;
- Nó giúp website tương tác tốt hơn với khách truy cập;
- Nó nhanh hơn và nhẹ hơn các ngôn ngữ lập trình khác.

2.5.5 Khuyết điểm của JavaScript.

Mọi ngôn ngữ lập trình đều có các khuyết điểm. Một phần là vì ngôn ngữ đó khi phát triển đến một mức độ như JavaScript, nó cũng sẽ thu hút lượng lớn hacker, scammer, và những người có ác tâm luôn tìm kiếm những lỗ hổng và các lỗi bảo mật để lợi dụng nó. Một số khuyết điểm có thể kể đến là:

- Dễ bị khai thác;
- Có thể được dùng để thực thi mã độc trên máy tính của người dùng;
- Nhiều khi không được hỗ trợ trên mọi trình duyệt;
- JS code snippets lớn;
- Có thể bị triển khai khác nhau tùy từng thiết bị dẫn đến việc không đồng nhất.

Cách hoạt động của JavaScript trên trang web là gì?

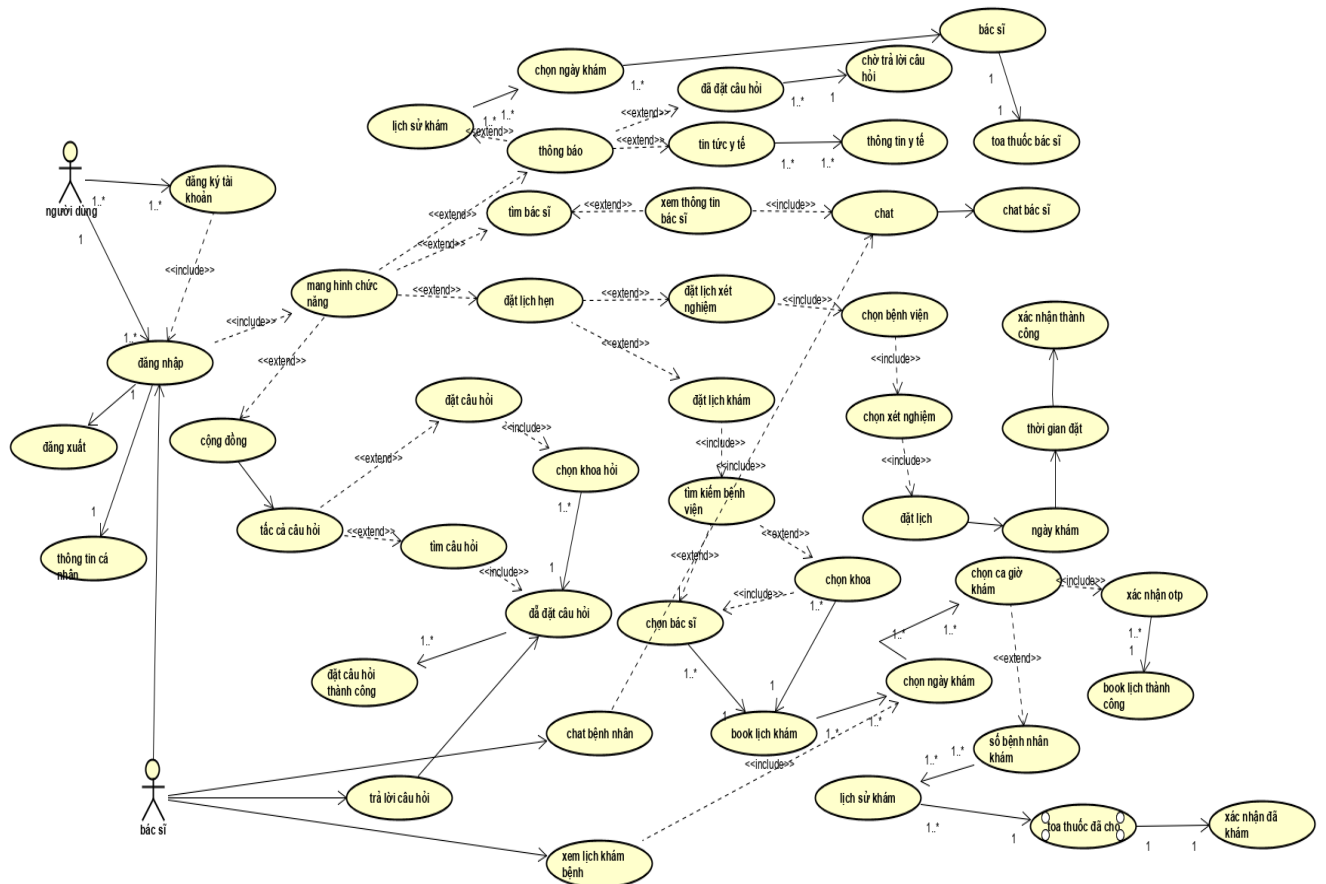
JavaScript thường được nhúng trực tiếp vào một trang web hoặc được tham chiếu qua file .js riêng. Nó là ngôn ngữ phía client, tức là script được tải về máy của khách truy cập và được xử lý tại đó thay vì phía server là xử lý trên server rồi mới đưa kết quả tới khách truy cập.

Hãy lưu ý là các trình duyệt web phổ biến cũng hỗ trợ việc người dùng có muốn tắt JavaScript hay không. Đó là lý do bạn nên biết trang web sẽ hoạt động như thế nào trong trường hợp không có JavaScript.

CHƯƠNG 3 PHÂN TÍCH THIẾT KẾ HỆ THỐNG

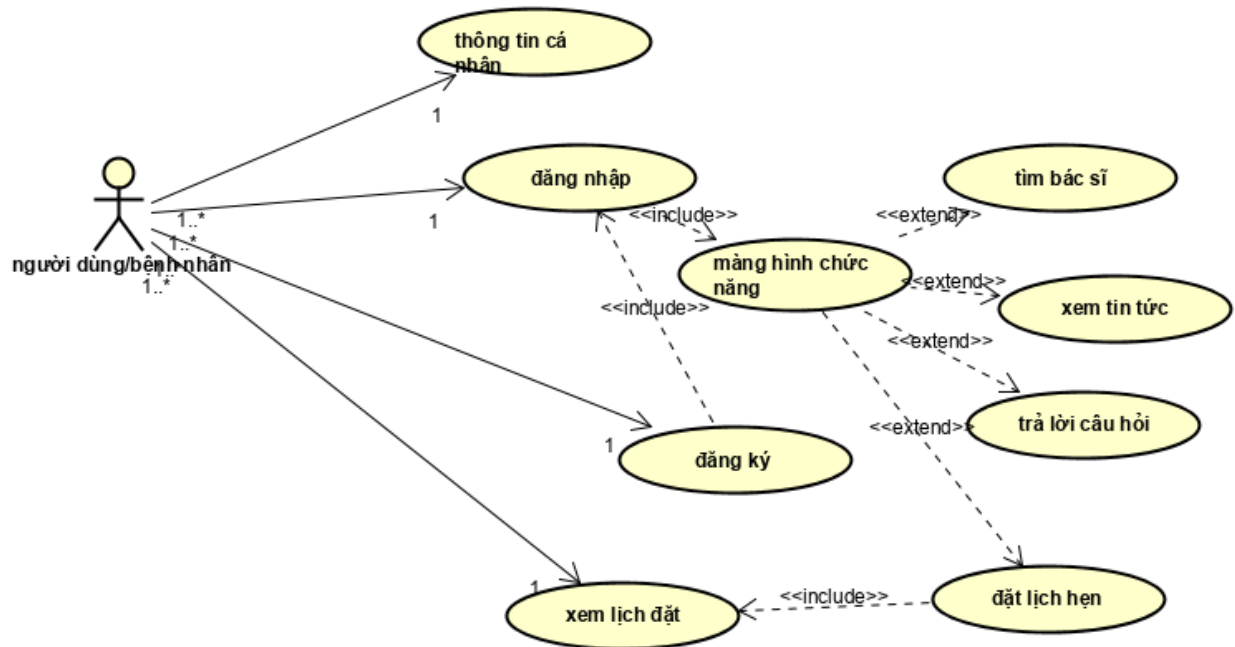
3.1 Sơ đồ Use Case của ứng dụng

3.1.1 Sơ đồ Use Case tổng quát.



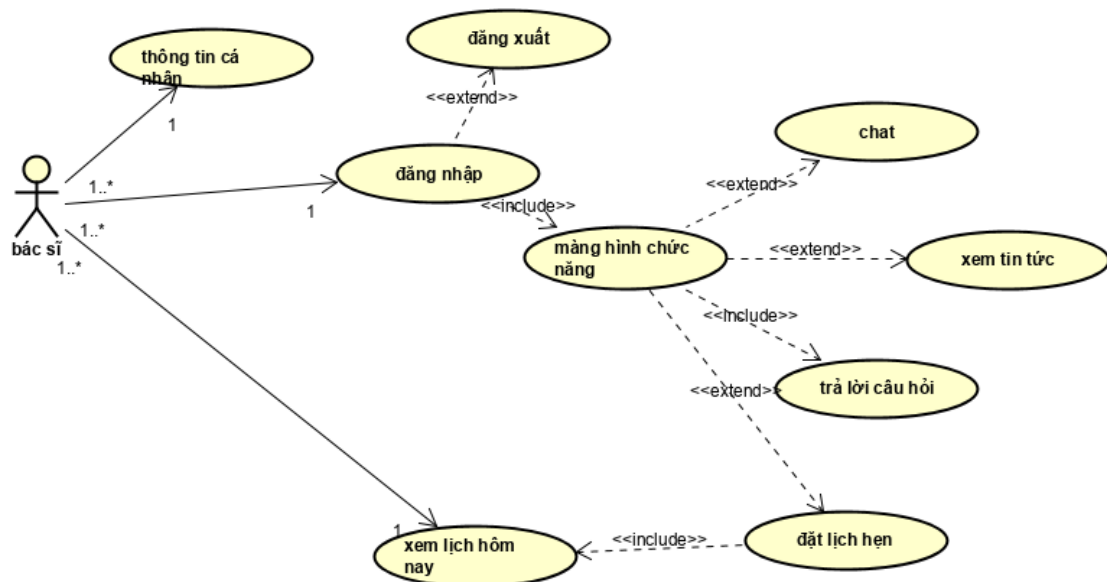
Hình 7 Usecase tổng

3.1.2 Sơ đồ Use Case người dùng.



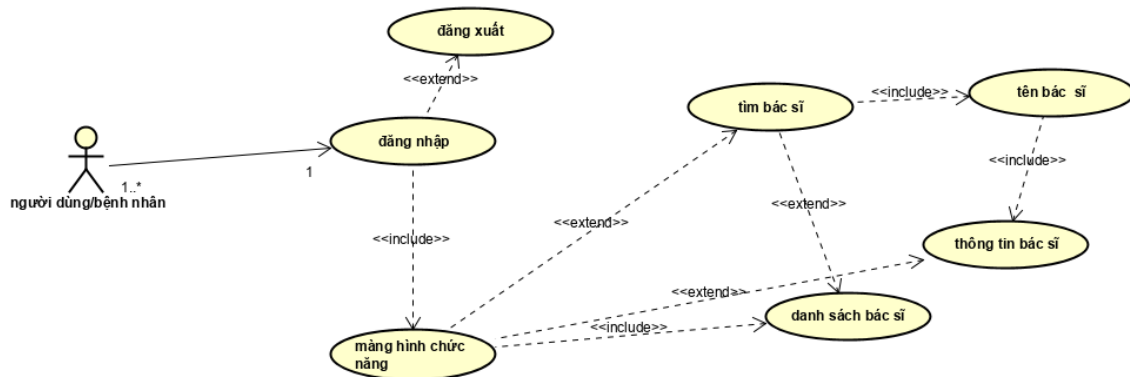
Hình 8 use case người dùng

3.1.3 Sơ đồ Use Case bác sĩ.



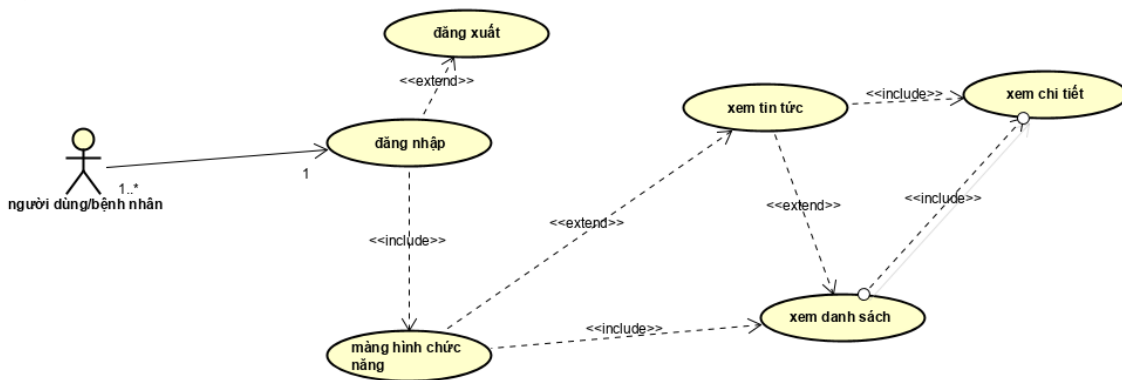
Hình 9 use case Bác sĩ

3.1.4 Sơ đồ Use Case tìm bác sĩ.



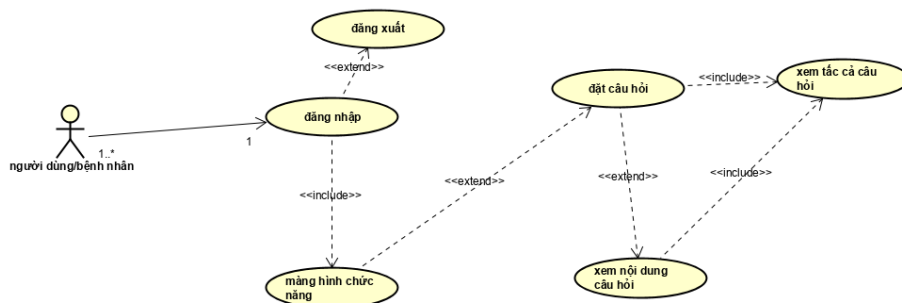
Hình 10 Use case tìm bác sĩ

3.1.5 Sơ đồ Use Case xem tin tức.



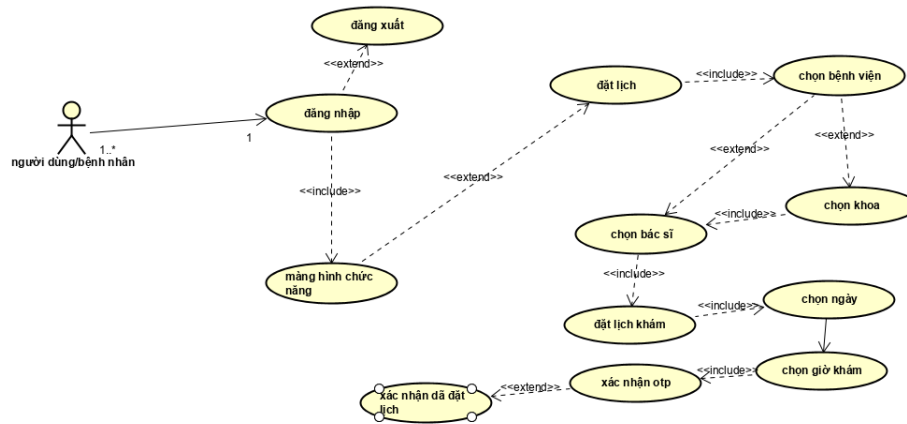
Hình 11 use case tin tức

3.1.6 Sơ đồ Use Case đặt câu hỏi.



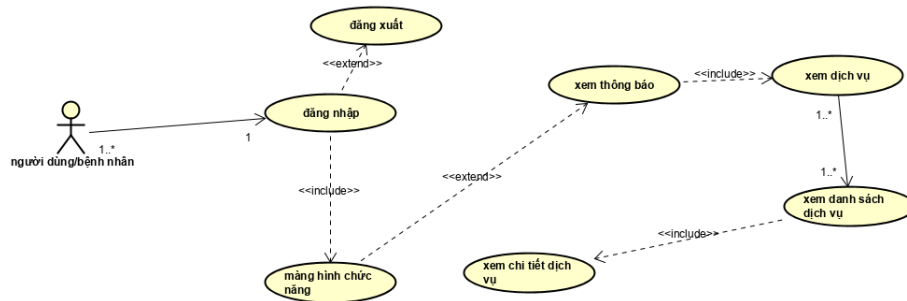
Hình 12 use case đặt câu hỏi

3.1.7 Sơ đồ Use Case đặt lịch khám.



Hình 13 use case đặt lịch khám

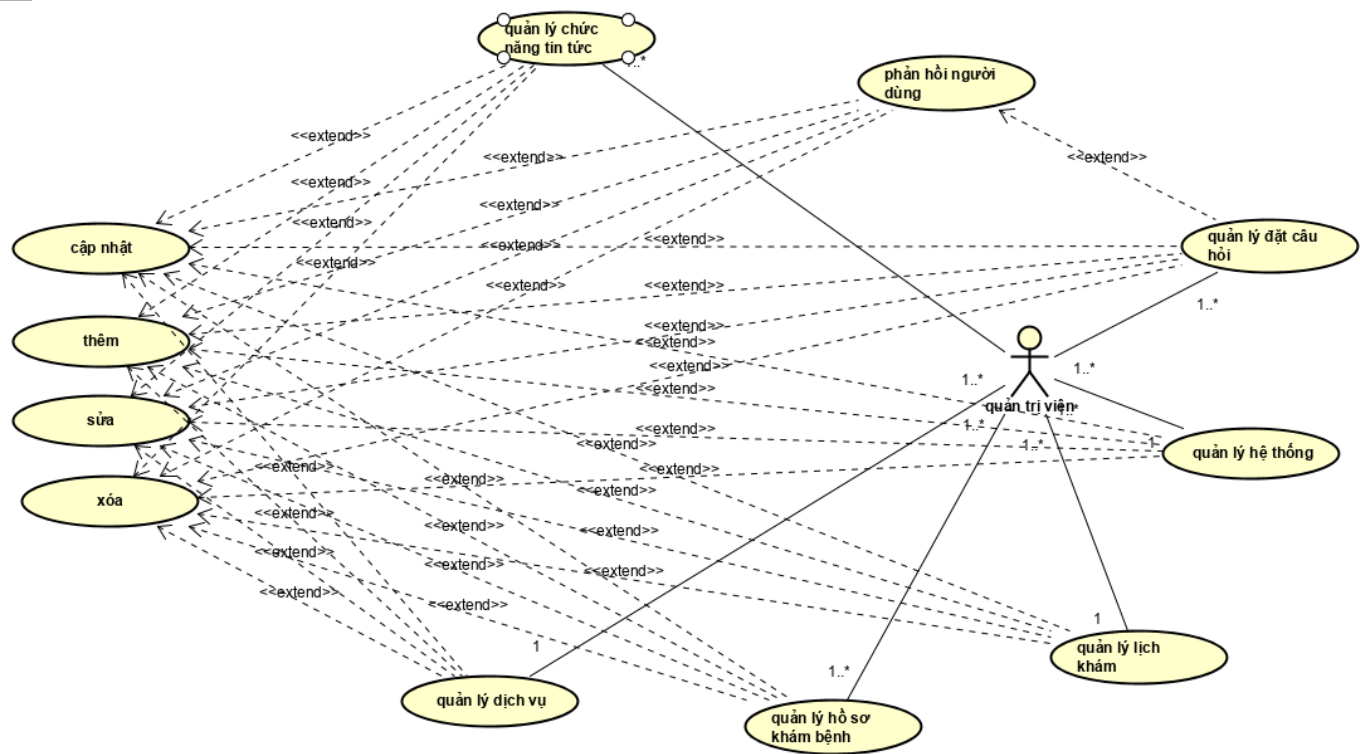
3.1.8 Sơ đồ Use Case tra cứu dịch vụ.



Hình 14 use case tra cứu dịch vụ

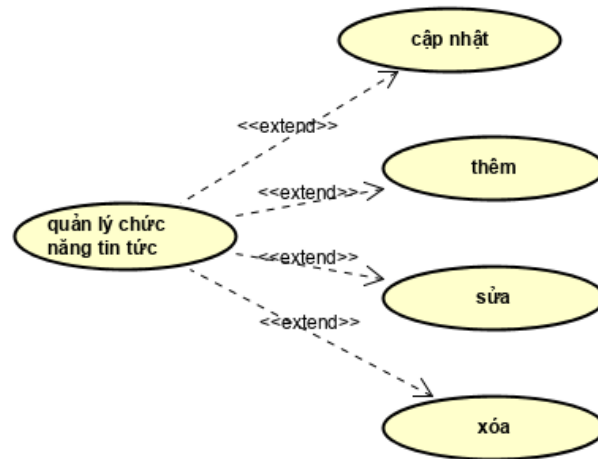
3.2 Sơ đồ Use Case quản trị viên.

3.2.1 Sơ đồ Use Case tổng quát



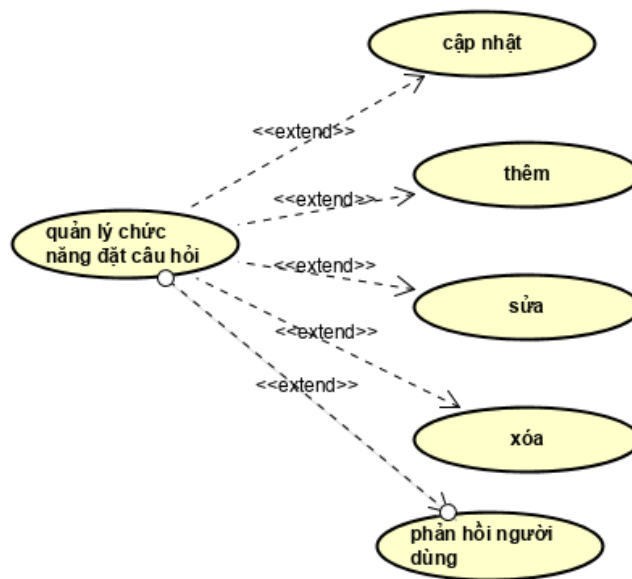
Hình 15 use case quản trị viên

3.2.2 Sơ đồ Use Case chức năng tin tức.



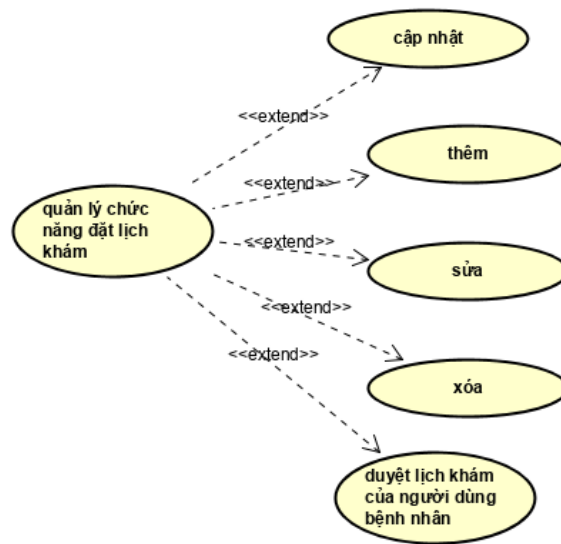
Hình 16 Use case chức năng tin tức

3.2.3 Sơ đồ Use Case chức năng đặt câu hỏi.



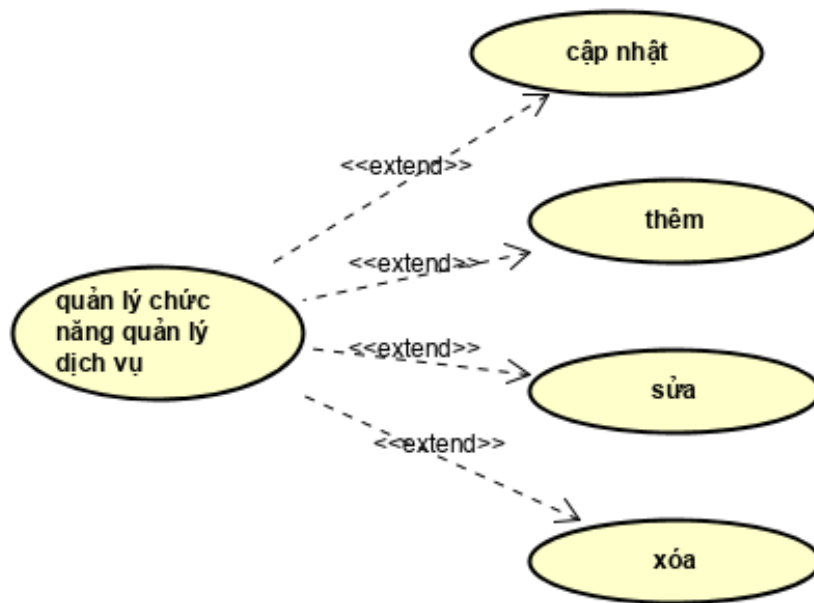
Hình 17 Use case đặt câu hỏi

3.2.4 Sơ đồ Use Case chức năng đặt lịch khám.



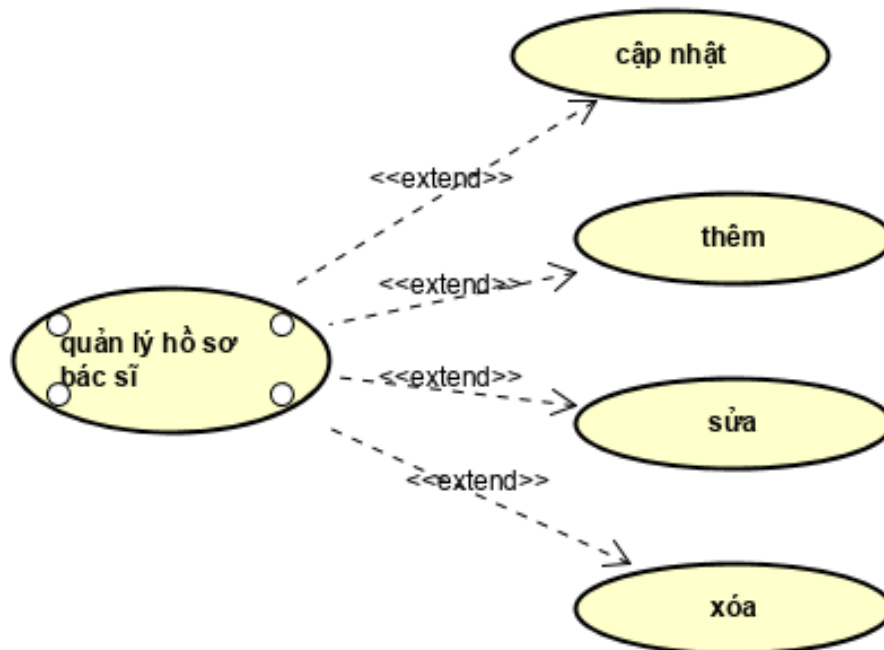
Hình 18 Use case quản lý khám

3.2.5 Sơ đồ Use Case chức năng dịch vụ.



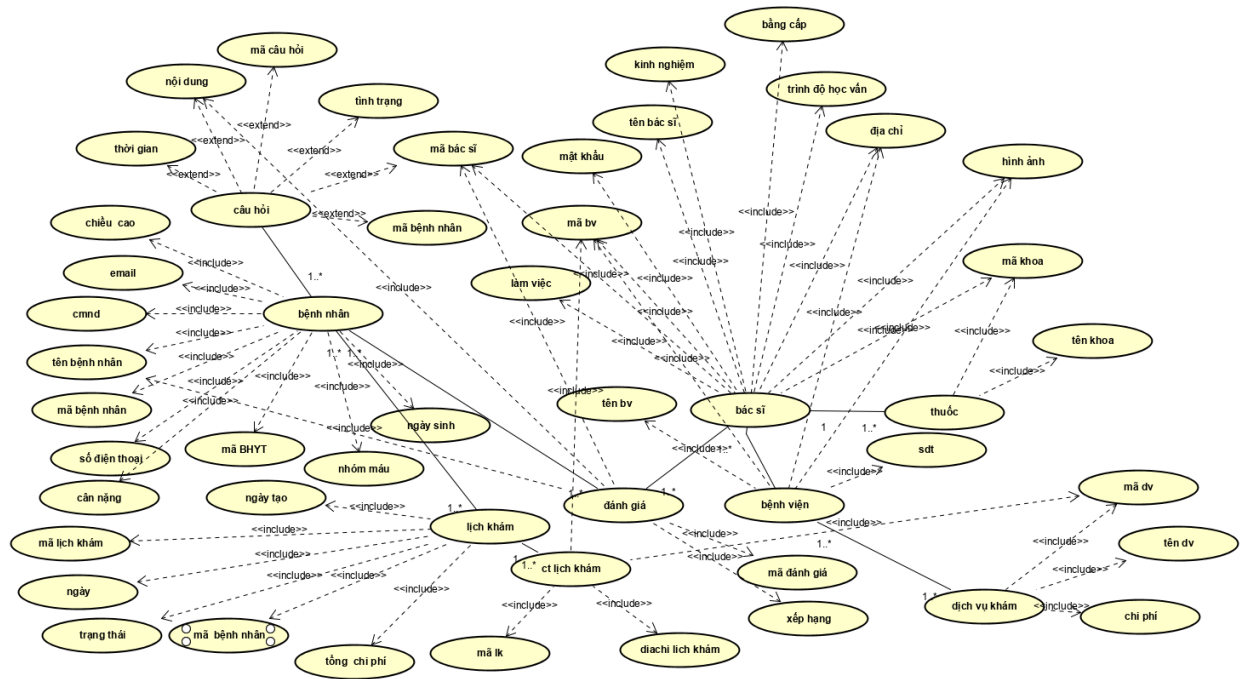
Hình 19 Use case chức năng dịch vụ

3.2.6 Sơ đồ Use Case chức năng bác sĩ.



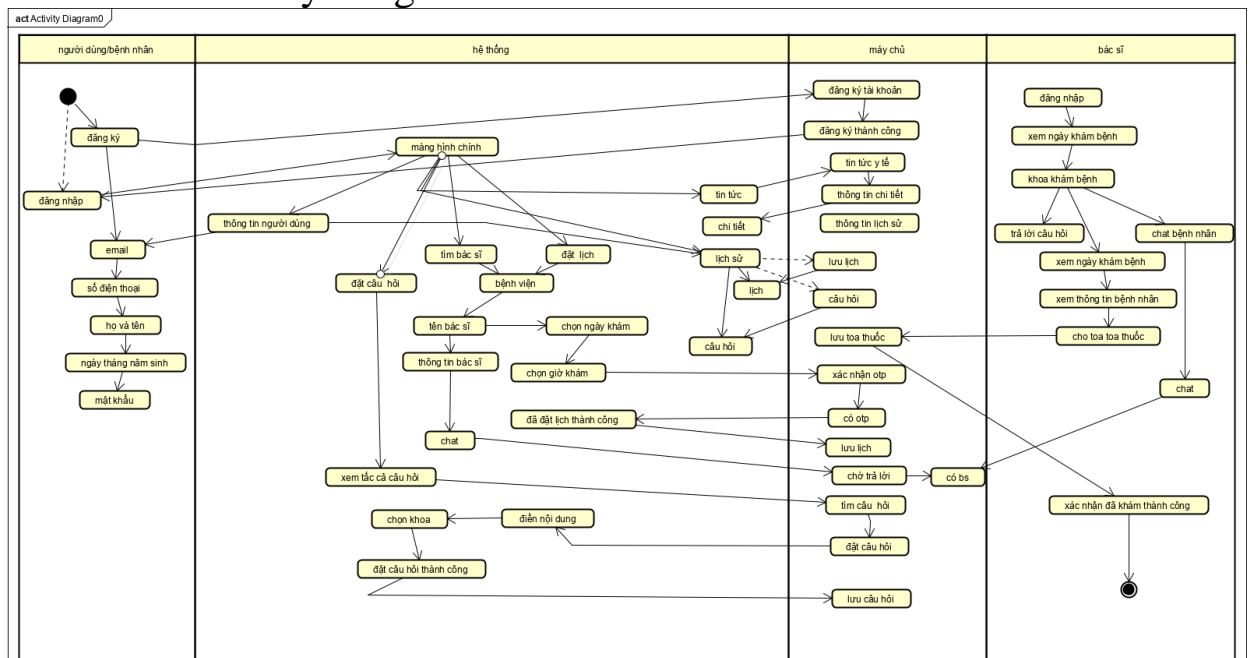
Hình 20 Use case chức năng bác sĩ

3.3 Sơ đồ ERD



Hình 21 Sơ đồ ERD

3.4 Sơ đồ Activity Diagram



Hình 22 Activity Diagram

3.5 Thiết kế cơ sở dữ liệu

3.5.1 Bảng dữ liệu Tin Tức.

TIN TỨC	
ID	INT (KHÓA CHÍNH)
TIEUDE	TEXT
NOIDUNG	TEXT
THOIGIAN	DATETIME
HINHANH	TEXT

Bảng 2 Dữ liệu tin tức

3.5.2 Bảng dữ liệu Chức Vụ.

CHỨC VỤ	
MACV	INT (AUTO_INCREMENT)
TENCV	VARCHAR(255)

Bảng 3 Dữ Liệu Chức Vụ

3.5.3 Bảng dữ liệu Người Dùng.

NGƯỜI DÙNG	
IDUSER	INT (AUTO_INCREMENT)
TEN	VARCHAR(255)
EMAIL	VARCHAR(255)
PASSWORD	VARCHAR(255)
SDT	AUTO_INCREMENT
DIACHI	VARCHAR(255)
MACV	INT
ROOM	TEXT
GIOI TINH	TEXT

Bảng 4 dữ liệu người dùng

3.5.4 Bảng dữ liệu Câu Hỏi.

CÂU HỎI	
ID	INT (AUTO_INCREMENT)
NOIDUNG	VARCHAR(255)
CAUTRALOI	VARCHAR(255)
THOIGIANDAT	DATETIME
THOIGIANTRALOI	DATETIME
IDUSER	INT

Bảng 5 dữ liệu câu hỏi

3.5.5 Bảng dữ liệu đặt lịch.

ĐẶT LỊCH	
IDDATLICH	INT (AUTO_INCREMENT)
IDDICHVU	INT
IDBACSI	INT
IDUSER	INT
IDNGAY	INT
IDCA	INT

Bảng 6 Dữ liệu đặt lịch

3.5.6 Bảng dữ liệu Chat.

CHAT	
ID	INT (AUTO_INCREMENT)
NOIDUNG	VARCHAR(255)
THOIGIAN	DATETIME
IDUSER	INT
ID	Khóa Chính
IDUSER	Khóa Ngoại

Bảng 7 Dữ liệu Chat

3.5.7 Bảng dữ liệu Chuyên Khoa.

CHUYÊN KHOA	
ID	INT (AUTO_INCREMENT)
TENCK	VARCHAR(255)
MOTA	TEXT

Bảng 8 Dữ liệu chuyên khoa

3.5.8 Bảng dữ liệu đặt lịch.

NGÀY ĐẶT LỊCH	
IDNGAY	INT (AUTO_INCREMENT)
IDDICHVU	INT
NGAY	DATETIME

Bảng 9 dữ liệu đặt lịch

3.5.9 Bảng dữ liệu Xác Nhận Đặt Lịch.

CA ĐẶT LỊCH	
IDCA	INT (AUTO_INCREMENT)
IDNGAY	INT
THOIGIAN	VARCHAR(255)
SOLUONG	VARCHAR(255)

Bảng 10 dữ liệu xác nhận đặt lịch

3.5.10 Bảng dữ liệu bác sĩ.

BÁC SĨ	
IDBACSI	INT (AUTO_INCREMENT)
IDDICHVU	INT
IDCHUYENKHOA	INT
IDBENHVIEN	INT
TENBACSI	TEXT
SOBENHNHANDAGIUP	INT
HINHANH	TEXT
CATUVAN	INT
SONAMKINHNGHIEM	INT
TRINHDOHOCVAN	TEXT
BANGCAPCHUNGCHI	TEXT

Bảng 11 dữ liệu bác sĩ

3.5.11 Bảng dữ liệu bệnh viện

BỆNH VIỆN	
IDBENHVIEN	INT (AUTO_INCREMENT)
TENBENHVIEN	INT
DIACHI	TEXT
SDT	NVARCHAR(10)
HINHANH	TEXT
GIOITHIEU	TEXT

Bảng 12 Dữ liệu bệnh viện

3.5.12 Bảng dữ liệu dịch vụ.

DỊCH VỤ	
IDDICHVU	INT (AUTO_INCREMENT)
IDBENHVIEN	INT
IDDANHMUC	INT
HINHANH	TEXT
TENDICHVU	TEXT
MOTA	TEXT
GIATIEN	TEXT
TENKHAC	TEXT
MUCTIEU	TEXT
DOITUONG	TEXT
HUONGDAN	TEXT

CHITIET	TEXT
---------	------

Bảng 13 Dữ liệu dịch vụ

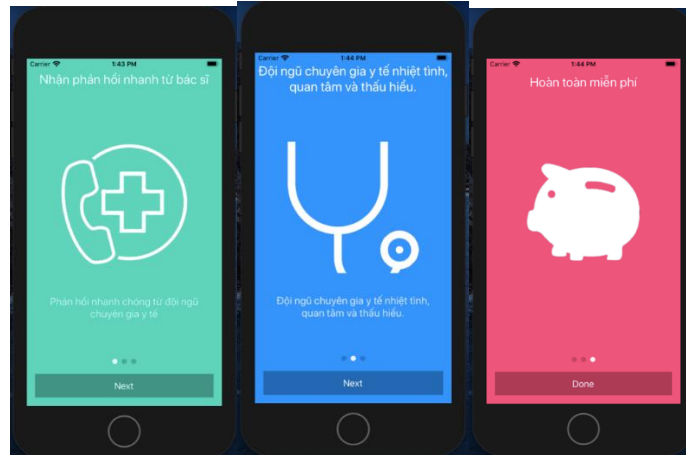
3.5.13 Bảng dữ liệu danh mục dịch vụ.

DANH MỤC DỊCH VỤ	
IDDANHMUC	INT (AUTO_INCREMENT)
TENDANHMUC	TEXT

Bảng 14 Danh mục dịch vụ

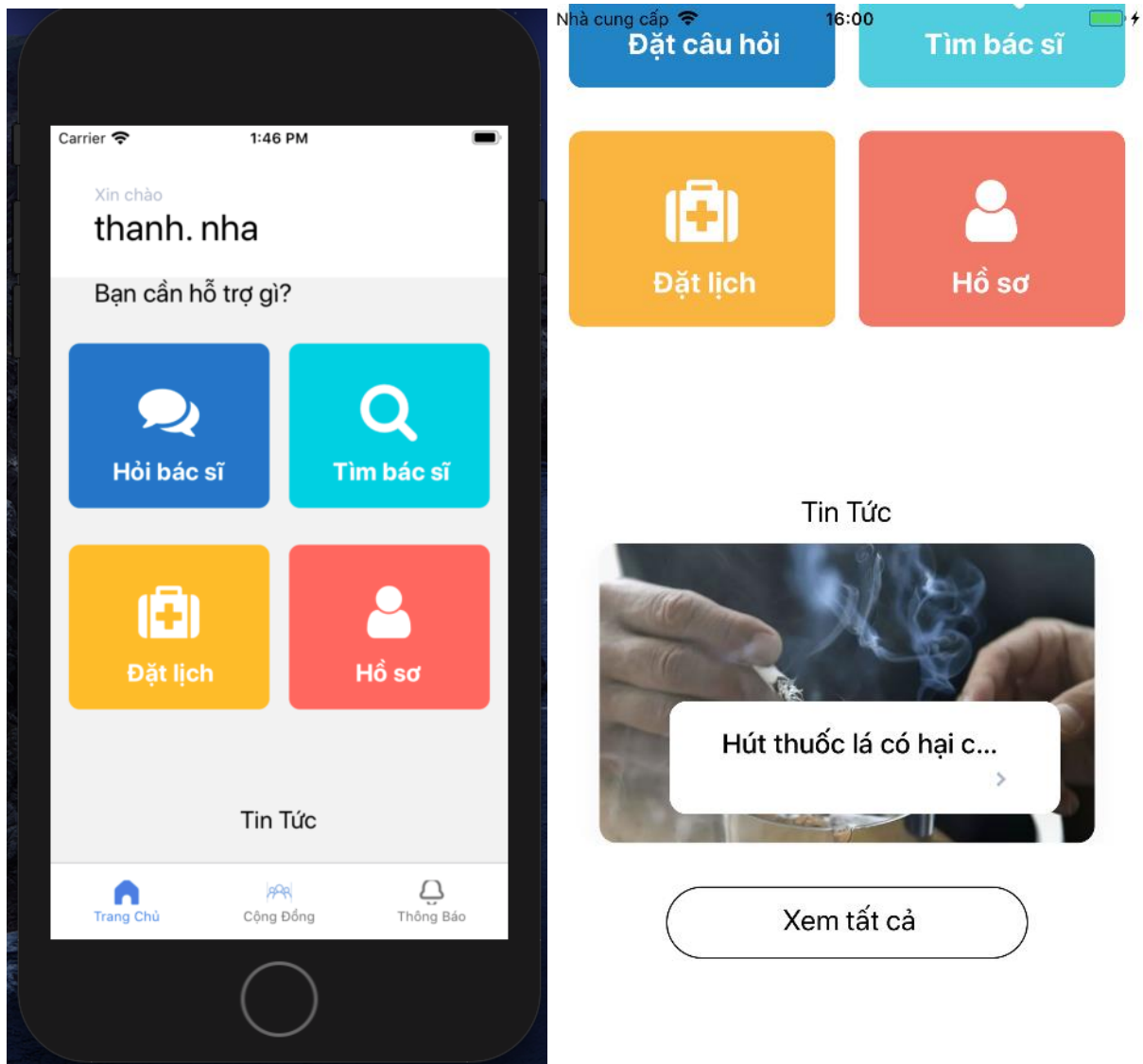
CHƯƠNG 4: GIAO DIỆN ỨNG DỤNG

4.1 Giao diện màn hình khởi động ứng dụng.



Hình 23 màn hình khởi động ứng dụng

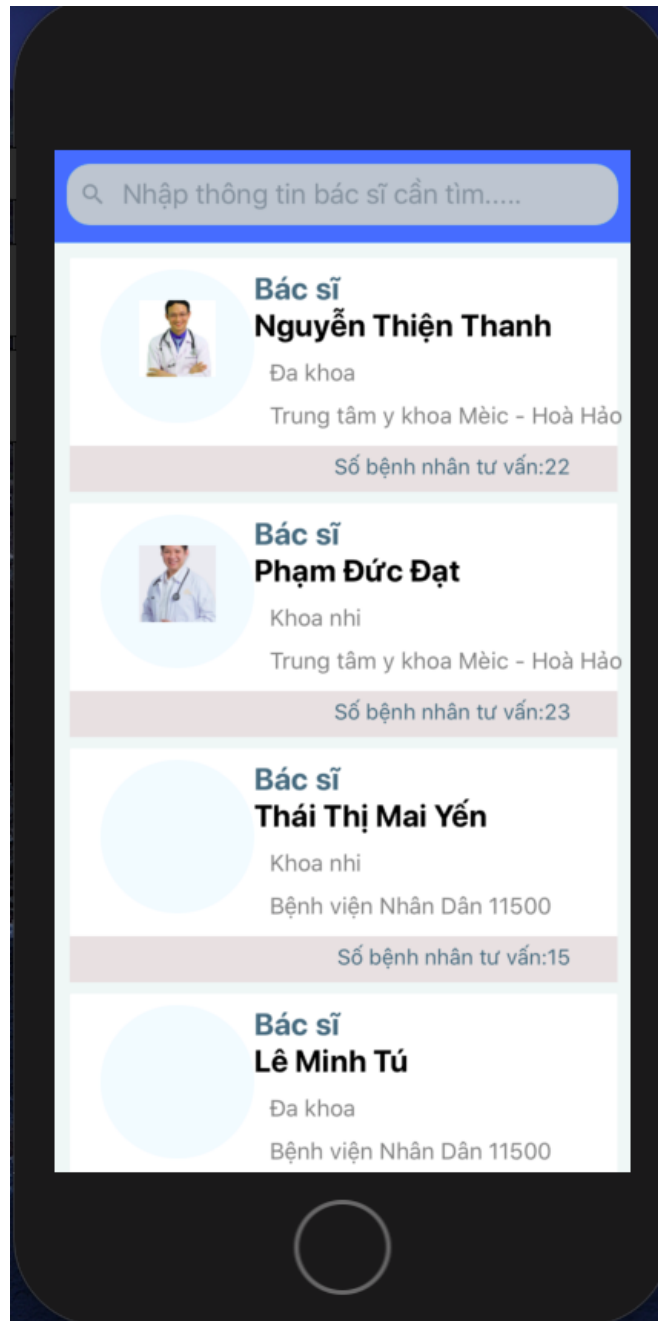
4.2 Giao diện màn hình chính.



Hình 24 màn hình chính

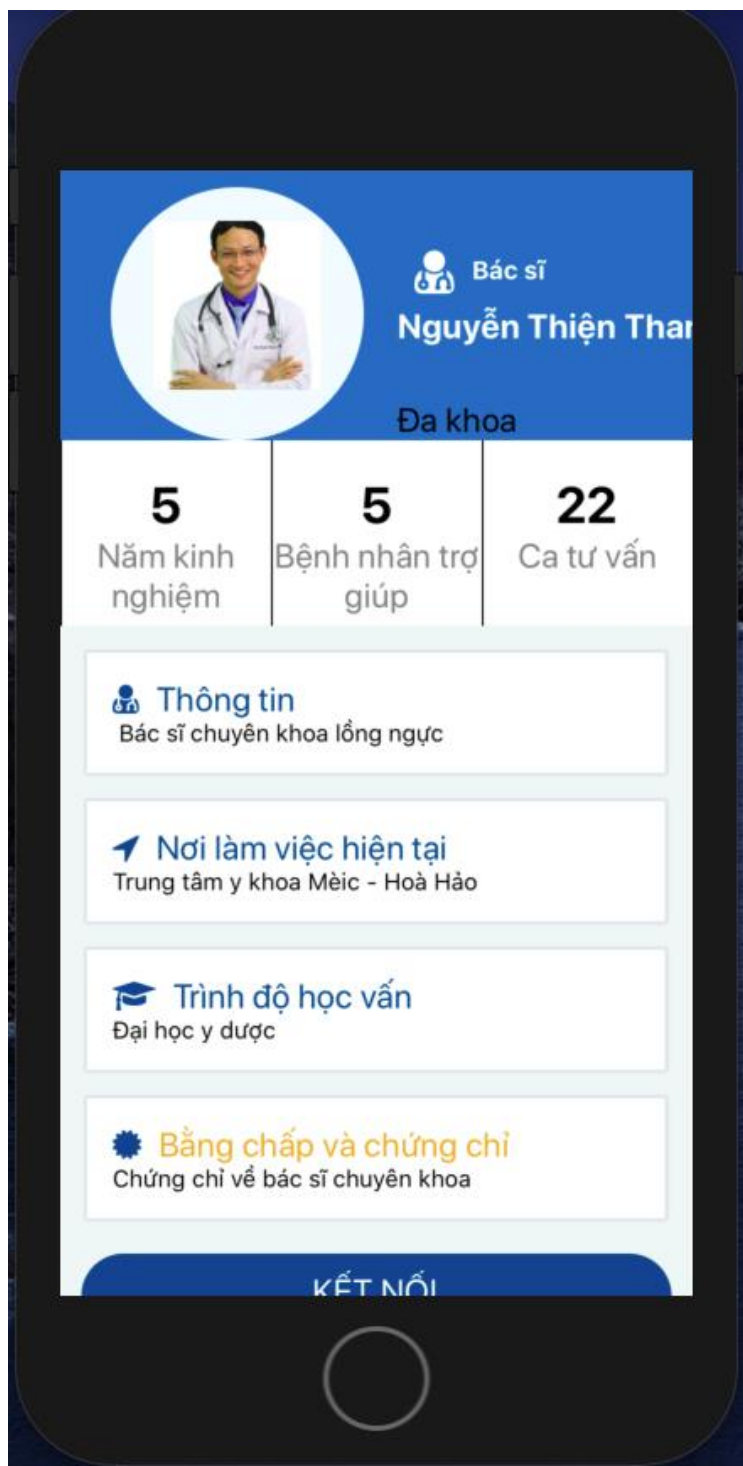
4.3 Giao diện bác sĩ.

4.3.1 Giao diện danh sách bác sĩ.



Hình 25 Danh sách bác sĩ

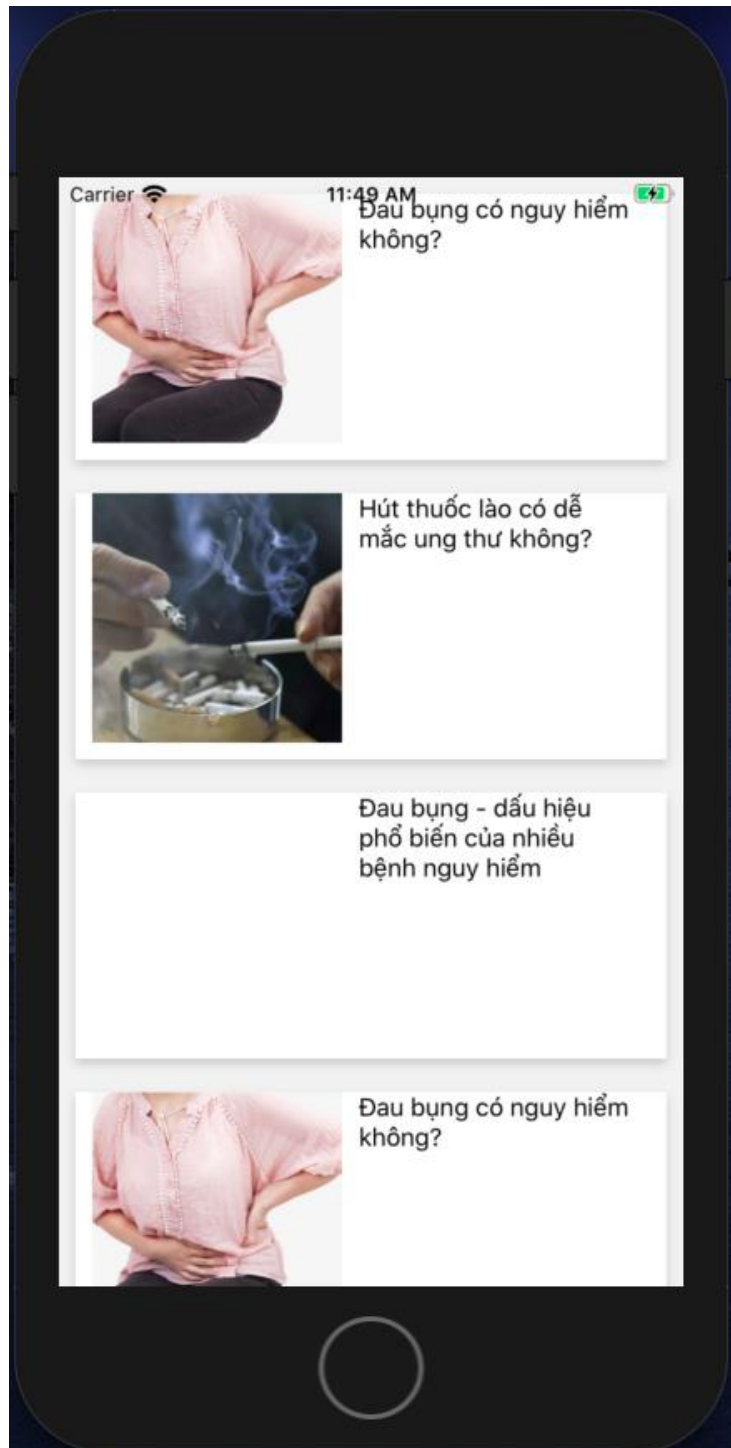
4.3.1 Giao diện thông tin chi tiết bác sĩ.



Hình 26 thông tin chi tiết bác sĩ

4.4 Giao diện tin tức.

4.4.1 Giao diện danh sách tin tức.



Hình 27 danh sách tin tức

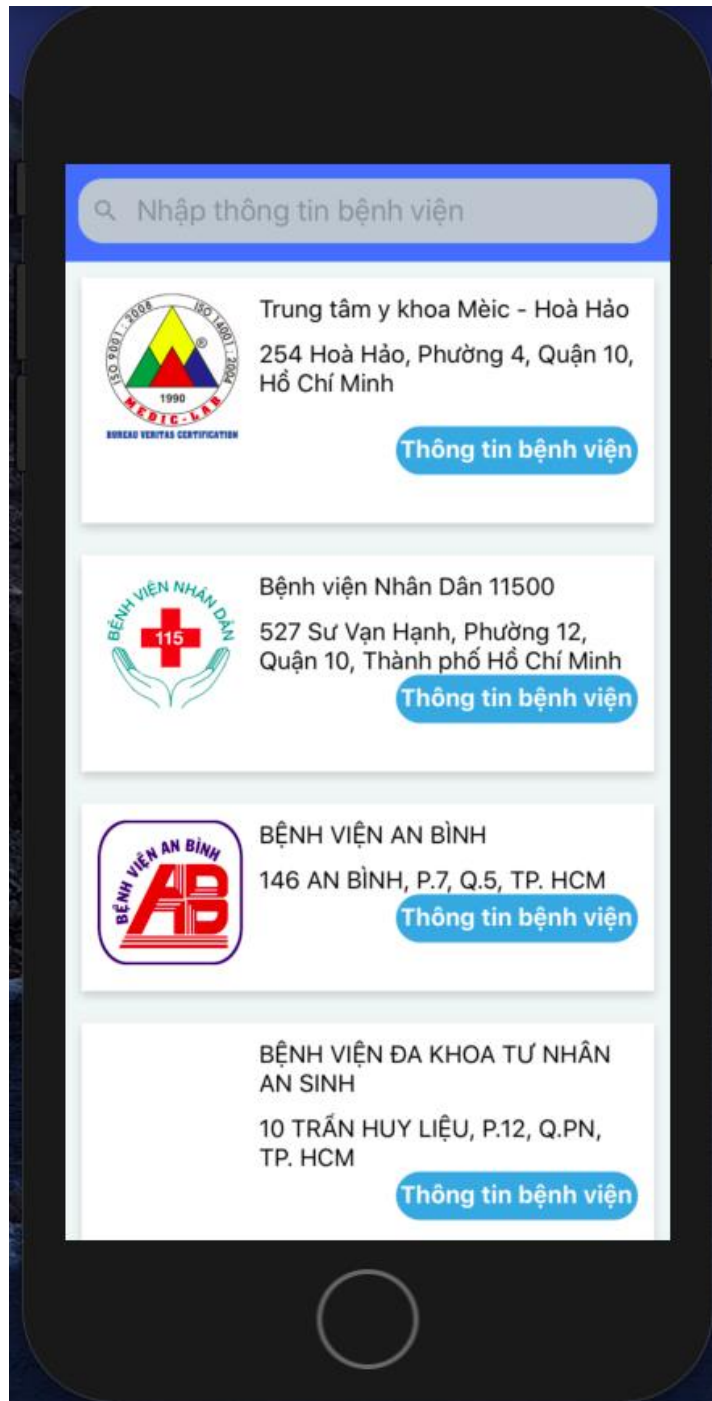
4.4.2 Giao diện chi tiết tin tức.



Hình 28 chi tiết tin tức.

4.5 Giao diện bệnh viện.

4.5.1 Giao diện danh sách bệnh viện.



Hình 29 Danh sách bệnh viện

4.5.2 Giao diện chi tiết bệnh viện.



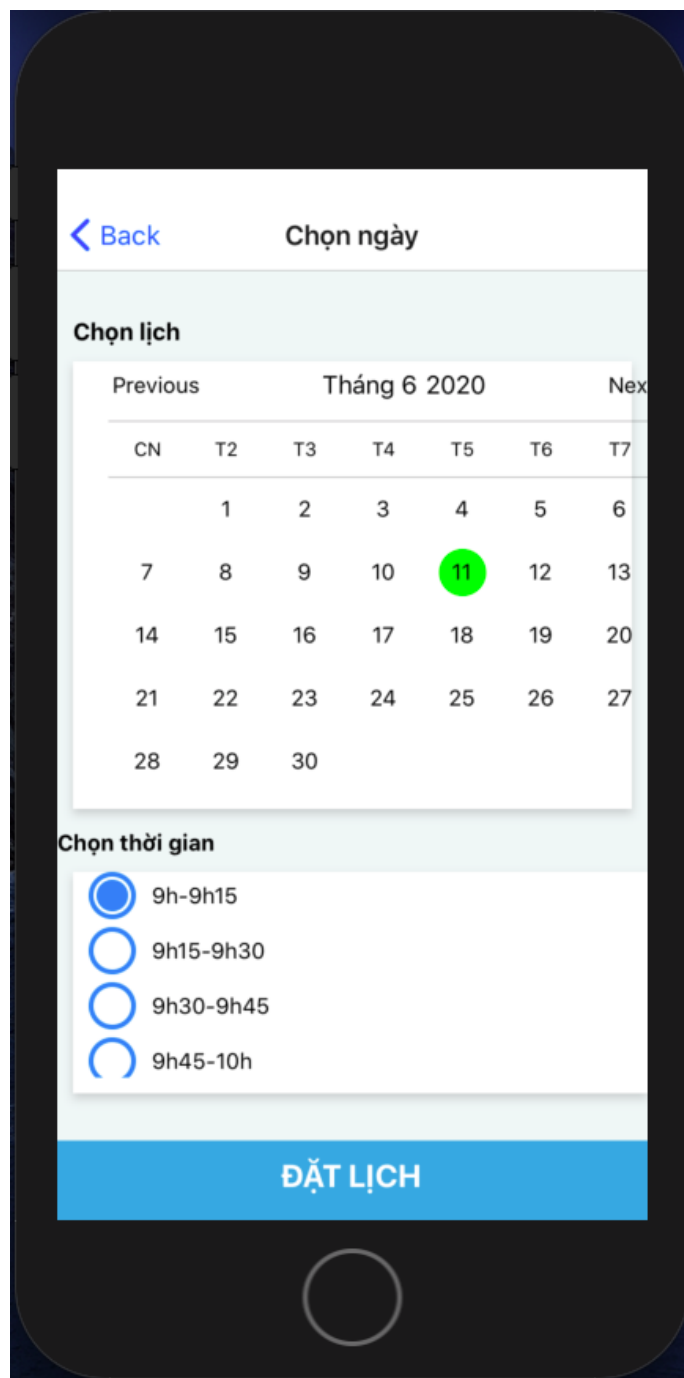
Hình 30 Chi tiết bệnh viện

4.6 Giao diện chat.



Hình 31 Giao diện chat

4.7 Giao diện đặt lịch.



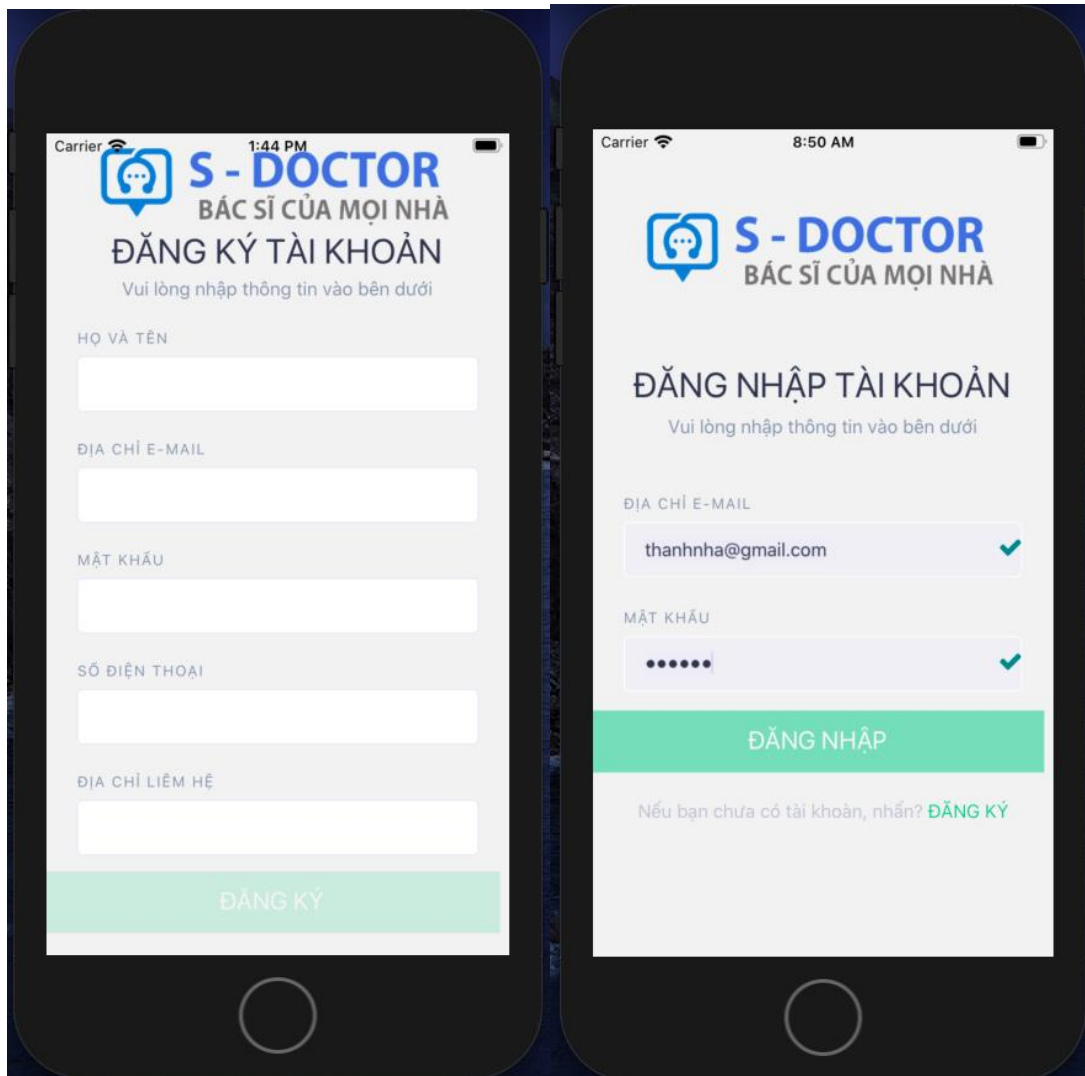
Hình 32 Giao diện đặt lịch

4.8 Giao diện đăng kí, đăng nhập.



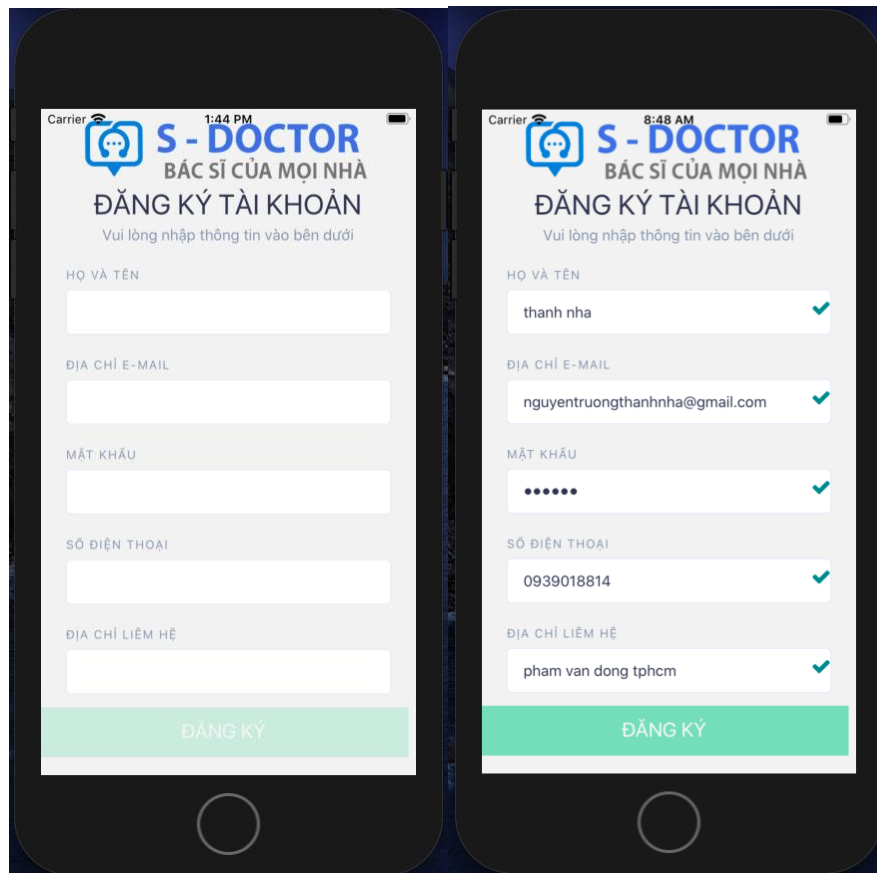
Hình 33 đăng kí, đăng nhập.

4.8.1 Giao diện đăng nhập.



Hình 34 đăng kí, đăng nhập.

4.8.2 Giao diện đăng kí



Hình 35 đăng kí, đăng ký.

4.9 Giao diện dịch vụ

4.9.1 Giao diện danh sách dịch vụ



Hình 36 Giao diện danh sách dịch vụ

4.9.2 Giao diện chi tiết dịch vụ



Hình 37 Giao diện danh sách dịch vụ

CHƯƠNG 5 TỔNG KẾT

5.1 Những điểm còn hạn chế

- Do thời gian nghiên cứu còn hạn chế và khả năng còn có hạn nên một số vấn đề còn trong ý tưởng, chưa được hoàn thành.
- Tính bảo mật còn thấp.
- Chức năng chat với bác sĩ chưa tối ưu.
- Giao diện chưa thật sự đẹp, còn những điểm hạn chế.

5.2 Những điểm đã đạt được.

- Xây dựng được Web Service để lưu trữ dữ liệu, quản lý dữ liệu và cập nhật dữ liệu một cách dễ dàng.
- Đã hoàn thành các chức năng sau:
 - Tra cứu bác sĩ, xem chi tiết bác sĩ.
 - Đăng nhập.
 - Đăng kí.
 - Tin tức.
 - Chat.
 - Đặt lịch.
 - Tra cứu bệnh viện.
 - Xem thông tin chi tiết dịch vụ.
- Giao diện thân thiện với người dùng.
- Chạy ứng dụng được trên cả hai nền tảng Android và IOS.

5.3 Hướng phát triển

- Sẽ hoàn thiện các chức năng chưa hoàn thành.
- Hoàn thiện thêm giao diện để người dùng dễ sử dụng.
- Nâng cao tính bảo mật của ứng dụng.
- Phát triển thêm website dùng chung cơ sở dữ liệu của ứng dụng, nhằm mở rộng người dùng tiềm năng.

5.4 Tài liệu tham khảo

- [1] https://react-native-training.github.io/react-native-elements/docs/0.19.1/getting_started.html
- [2] <https://facebook.github.io/react-native/>
- [3] <https://www.npmjs.com/package/react-native>
- [4] <https://docs.expo.io/guides/routing-and-navigation/?redirected>
- [5] <http://www.reactnative.com/courses/>
- [6] https://www.youtube.com/watch?v=kNHuLOXR5T0&list=PLWBrqglNjNI31S91FFE63DtuRc9v9LSGl&ab_channel=NguyenDucHoang
- [7] <https://react-native-elements.github.io/react-native-elements/>
- [8] <https://reactnative.dev/docs/intro-react-native-components>
- [9] <https://expo.io/@monte9/react-native-elements-app>
- [10] <https://wcandillon.github.io/react-native-elements/BuildingASeedProject.html>
- [11] <https://blog.bitsrc.io/11-react-native-component-libraries-you-should-know-in-2018-71d2a8e33312>