

TRƯỜNG ĐẠI HỌC KINH TẾ
KHOA THỐNG KÊ – TIN HỌC



BÁO CÁO TỐT NGHIỆP

NGÀNH HỆ THỐNG THÔNG TIN QUẢN LÝ
CHUYÊN NGÀNH QUẢN TRỊ HỆ THỐNG THÔNG TIN

Xây dựng ứng dụng chia sẻ và kết nối địa điểm yêu thích

Hotspotter

Sinh viên thực hiện : Nguyễn Duy Quang
Lớp : 46K21.2
Đơn vị thực tập : Wiicamp
Cán bộ hướng dẫn : Hoàng Xuân Quang
Giảng viên hướng dẫn : ThS.Cao Thị Nhâm

Đà Nẵng, 05/2024

NHẬN XÉT CỦA ĐƠN VỊ THỰC TẬP

Họ và tên sinh viên:

Lớp: Khoa: Trường:

Thực tập từ ngày: / / 2024 đến ngày: / / 2024

Tại:

Địa chỉ:

Sau quá trình thực tập tại đơn vị của sinh viên, chúng tôi có một số nhận xét, đánh giá như sau:

ST T	Mục đánh giá	Rất tệ	Tệ	Bình thườn g	Tốt	Rất tốt
1	Về thái độ, ý thức, đạo đức, kỷ luật và văn hóa công ty	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
2	Kiến thức chuyên môn	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	Khả năng hòa nhập, thích nghi và tác phong nghề nghiệp	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
4	Trách nhiệm, sáng tạo trong công việc	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
5	Sáng tạo trong công việc	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Các nhận xét khác (Khoa chúng tôi mong muốn nhận thêm những ý kiến khác từ quý doanh nghiệp nhằm nâng cao chất lượng đào tạo)

.....

.....

.....

.....

.....

Điểm:

....., ngàythángnăm 2024

Xác nhận của đơn vị thực tập

LỜI CẢM ƠN

Lời đầu tiên, em xin gửi lời cảm ơn chân thành đến cô Cao Thị Nhâm, Giảng viên hướng dẫn khoa Thống Kê – Tin học, người đã tận tình hướng dẫn, góp ý và tạo điều kiện cho em hoàn thành đề tài "Ứng dụng chia sẻ và kết nối địa điểm yêu thích". Nhờ sự dìu dắt và những lời khuyên quý báu của thầy/cô, em đã có thêm nhiều kiến thức và kinh nghiệm để hoàn thành đề tài này một cách tốt nhất.

Em cũng xin gửi lời cảm ơn đến anh Hoàng Xuân Quang, Mentor của em, người đã dành thời gian và tâm huyết để hướng dẫn, chia sẻ những kinh nghiệm thực tế và giúp đỡ em giải quyết những khó khăn trong quá trình thực hiện đề tài.

Em xin cảm ơn các đồng nghiệp đã nhiệt tình hỗ trợ, giúp đỡ và tạo điều kiện cho em hoàn thành đề tài. Em trân trọng những đóng góp ý kiến quý báu của các anh/chị, góp phần hoàn thiện đề tài của em

Em cũng xin cảm ơn gia đình đã luôn động viên, khích lệ và tạo điều kiện cho em trong suốt quá trình học tập và nghiên cứu.

Cuối cùng, em xin cảm ơn tất cả mọi người đã quan tâm và giúp đỡ em trong quá trình thực hiện đề tài. Em mong rằng đề tài này sẽ góp phần mang lại những giá trị thiết thực cho cộng đồng.

LỜI CAM ĐOAN

1. Nội dung trong đồ án này là do em thực hiện dưới sự hướng dẫn trực tiếp của cô Cao Thị Nhâm. Em đã tự mình nghiên cứu, tìm tòi và hoàn thành toàn bộ nội dung đề tài, không sao chép hay sử dụng trái phép các tài liệu khác
2. Các tham khảo dùng trong đồ án đều được trích dẫn rõ ràng tên tác giả, tên công trình, thời gian, địa điểm công bố. Em đã thực hiện việc trích dẫn theo đúng quy định của trường Đại học, đảm bảo tính trung thực và minh bạch trong quá trình nghiên cứu
3. Nếu có những sao chép không hợp lệ, vi phạm quy chế đào tạo, em xin chịu hoàn toàn trách nhiệm. Em hiểu rõ những quy định về đạo đức học thuật và cam kết tuân thủ nghiêm túc. Bất kỳ hành vi gian lận nào đều sẽ dẫn đến việc xử lý kỷ luật theo quy định của trường

Đà Nẵng, ngày 17 tháng 02 năm 2024

Sinh viên thực hiện

Nguyễn Duy Quang

MỤC LỤC

NHẬN XÉT CỦA ĐƠN VỊ THỰC TẬP.....	1
LỜI CẢM ƠN.....	i
LỜI CAM ĐOAN	ii
MỤC LỤC	iii
DANH MỤC HÌNH ẢNH.....	vi
DANH MỤC BẢNG BIỂU	viii
DANH MỤC CÁC TỪ VIẾT TẮT	x
LỜI MỞ ĐẦU.....	1
CHƯƠNG 1. TỔNG QUAN VỀ ĐƠN VỊ THỰC TẬP.....	5
1.1. Giới thiệu chung về Wiicamp	5
1.2. Tầm nhìn, sứ mệnh	5
1.3. Dịch vụ.....	5
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT	10
2.1. JavaScript	10
2.2. NodeJs.....	10
2.2.1. Giới thiệu về NodeJs.....	10
2.2.2. Ưu điểm của NodeJs	11
2.2.3. Khi nào nên sử dụng NodeJs.....	12
2.3. ExpressJs	12
2.4. REST APIs	13
2.4.1. Giới thiệu về REST APIs.....	13
2.4.2. Sự cần thiết của REST APIs trong phát triển ứng dụng web.....	14
2.4.3. Thiết kế REST APIs.....	14
2.4.4. Các quy tắc định dạng URL trong REST APIs	15
2.4.5. Phương thức yêu cầu (Request Method) trong REST APIs	16

2.4.6. Mã trạng thái phản hồi (Response Status Codes) trong REST APIs .	17
2.5. MongoDB.....	19
2.5.1. Giới thiệu về MongoDB.....	19
2.5.2. Ưu điểm của MongoDB.....	20
2.6. Đồng bộ và bất đồng bộ	20
2.6.1. Đồng bộ	20
2.6.2. Bất đồng bộ.....	20
2.6.3. Ưu và nhược điểm của bất đồng bộ	21
CHƯƠNG 3. PHÂN TÍCH HỆ THỐNG	22
3.1. Tổng quan về ứng dụng và website quản trị hotspotter.....	22
3.1.1. Giới thiệu ứng dụng và website quản trị.....	22
3.1.2. Module ứng dụng và website quản trị	23
3.2. Workflow	30
3.2.1. Workflow ứng dụng Hotspotter.....	30
3.2.2. Workflow website quản trị hotspotter	32
3.3. Sơ đồ Use Case.....	33
3.3.1. Sơ đồ Use Case tổng quát ứng dụng Hotspotter.....	33
3.3.2. Sơ đồ usecase website quản trị.....	34
3.4. Mô tả chi tiết các Usecase ứng dụng và website quản trị Hotspotter....	34
3.4.1. Mô tả chi tiết các Usecase ứng dụng Hotspotter	34
3.4.2. Mô tả chi tiết các Usecase website quản trị Hotspotter	48
CHƯƠNG 4. THIẾT KẾ HỆ THỐNG	54
4.1. Thiết kế kiến trúc hệ thống	54
4.1.1. Thiết kế kiến trúc ứng dụng Hotspotter	54
4.1.2. Thiết kế kiến trúc hệ thống website quản trị Hotspotter.....	54
4.2. Thiết kế cơ sở dữ liệu	55

4.3. Thiết kế lớp	55
4.4. Thiết kế giao diện người dùng	55
4.4.1. Giao diện website quản trị Hotspotter	55
4.5. Thiết kế API	55
4.5.1. Thiết kế API ứng dụng Hotspotter.....	55
CHƯƠNG 5. XÂY DỰNG HỆ THỐNG.....	60
5.1. Công cụ và môi trường phát triển.....	60
5.2. Những chức năng đã hoàn thành.....	65
TÀI LIỆU THAM KHẢO	66
CHECK LIST CỦA BÁO CÁO	67
PHỤ LỤC	1

DANH MỤC HÌNH ẢNH

Hình 1.1 Logo công ty Wiicamp	5
Hình 2.1 Hình ảnh về JavaScript	10
Hình 2.2 Hình ảnh về NodeJs	11
Hình 2.3 Hình ảnh về ExpressJs	13
Hình 2.4 Hình ảnh về web API.....	14
Hình 2.5 Hình ảnh về HTTP Request Methods.....	16
Hình 2.6 Hình ảnh danh mục mã trạng thái phản hồi	17
Hình 2.7 Hình ảnh về MongoDB.....	20
Hình 2.8 Hình ảnh về xử lý bất đồng bộ	21
Hình 3.1 Hình workflow ứng dụng Hotspotter.....	31
Hình 3.2 Hình workflow website quản trị Hotspotter.....	32
Hình 3.3 Hình sơ đồ usecase ứng dụng Hotspotter	33
Hình 3.4 Hình sơ đồ usecase website quản trị Hotspotter	34
Hình 4.1 Kiến trúc hệ thống ứng dụng Hotspotter	54
Hình 4.2 Kiến trúc hệ thống website quản trị Hotspotter	54
Hình 5.1 Logo Trello.....	60
Hình 5.2 Logo git	61
Hình 5.3 Logo gitlab	61
Hình 5.4 Logo fima	62
Hình 5.5 Logo clickup.....	63
Hình 5.6 Logo Visual Studio Code.....	64
Hình 5.7 Logo ExpressJs	64
Hình 5.8 Logo MongoDB.....	65

DANH MỤC BẢNG BIỂU

Bảng 3.1 Module ứng dụng Hotspotter.....	27
Bảng 3.2 Module website quản trị Hotspotter.....	30
Bảng 3.3 Mô tả chi tiết chức năng đăng kí bằng email ()	35
Bảng 3.4 Mô tả chi tiết chức năng đăng nhập bằng email (ứng dụng Hotspotter)	36
Bảng 3.5 Mô tả chi tiết chức năng Đăng ký/ Đăng nhập bằng Google/ Facebook/ AppleID (ứng dụng Hotspotter).....	37
Bảng 3.6 Mô tả chi tiết chức năng xem thông tin tài khoản (ứng dụng Hotspotter)	38
Bảng 3.7 Mô tả chi tiết chức năng đăng ảnh đại diện/ ảnh bìa (ứng dụng Hotspotter)	39
Bảng 3.8 Mô tả chi tiết chức năng thay đổi email cá nhân (ứng dụng Hotspotter)	40
Bảng 3.9 Mô tả chi tiết chức năng xem địa điểm theo dạng danh sách/ bản đồ (ứng dụng Hotspotter)	40
Bảng 3.10 Mô tả chi tiết chức năng tìm kiếm địa điểm theo từ khóa (ứng dụng Hotspotter)	41
Bảng 3.11 Mô tả chi tiết chức năng lọc địa điểm theo các tiêu chí lọc (ứng dụng Hotspotter)	42
Bảng 3.12 Mô tả chi tiết chức năng xem danh sách cá nhân (ứng dụng Hotspotter)	43
Bảng 3.13 Mô tả chi tiết chức năng xem danh sách đang theo dõi (ứng dụng Hotspotter)	44
Bảng 3.14 Mô tả chức năng xem chi tiết danh sách cá nhân (ứng dụng Hotspotter)	45
Bảng 3.15 Mô tả chi tiết chức năng tạo mới danh sách (ứng dụng Hotspotter)	46

Bảng 3.16 Mô tả chi tiết chức năng chỉnh sửa danh sách (ứng dụng Hotspotter)	47
Bảng 3.17 Mô tả chi tiết chức năng nhân bản danh sách (ứng dụng Hotspotter)	47
Bảng 3.18 Mô tả chi tiết chức năng lưu địa điểm vào danh sách cá nhân/ danh sách tham gia làm cộng tác viên (ứng dụng Hotspotter)	48
Bảng 3.19 Mô tả chi tiết chức năng đăng nhập (website quản trị Hotspotter)	49
Bảng 3.20 Mô tả chi tiết chức năng xem danh sách người dùng (website quản trị Hotspotter)	49
Bảng 3.21 Mô tả chi tiết chức năng xem chi tiết thông tin người dùng (website quản trị Hotspotter)	50
Bảng 3.22 Mô tả chi tiết chức năng vô hiệu hóa/ kích hoạt tài khoản người dùng (website quản trị Hotspotter)	50
Bảng 3.23 Mô tả chi tiết chức năng tạo tài khoản quản trị viên (website quản trị Hotspotter)	51
Bảng 3.24 Mô tả chi tiết chức năng tạo địa điểm (website quản trị Hotspotter)	52
Bảng 3.25 Mô tả chi tiết chức năng xem danh sách địa điểm (website quản trị Hotspotter)	52
Bảng 3.26 Mô tả chi tiết chức năng xem chi tiết địa điểm (website quản trị Hotspotter)	53
Bảng 3.27 Mô tả chi tiết chức năng chỉnh sửa địa điểm (website quản trị Hotspotter)	53

DANH MỤC CÁC TỪ VIẾT TẮT

AI: Artificial Intelligence

API: Application Programming Interface

URL: Uniform Resource Locator

DB: Database

UI: User Interface

UX: User Experience

HTTP: Hypertext Transfer Protocol

TCP: Transmission Control Protocol

ID: Identification

LỜI MỞ ĐẦU

1. Lý do chọn đề tài

- Đây là một ứng dụng chia sẻ địa điểm yêu thích, hướng đến giới trẻ, mang lại trải nghiệm tuyệt vời và kết nối cộng đồng. Ứng dụng này sẽ cung cấp nhiều tính năng mới và hấp dẫn, như kết bạn, theo dõi, giao diện đơn giản, hệ thống đánh giá, lọc theo tiêu chí và thông tin cập nhật địa điểm.
- Với khả năng của mình là một backend developer, em sẽ chịu trách nhiệm xây dựng và duy trì các API chính để đảm bảo tính ổn định và hiệu suất của hệ thống. Điều này sẽ đòi hỏi kiến thức vững về xử lý dữ liệu, tối ưu hóa cơ sở dữ liệu và kỹ năng tương tác với frontend để đảm bảo tính đồng bộ và hiệu quả trong trải nghiệm người dùng.
- Bằng cách tham gia vào việc phát triển phần backend của ứng dụng, em có cơ hội nâng cao kỹ năng lập trình của mình, đặc biệt là trong việc xây dựng và duy trì các API chất lượng cao. Đồng thời, dự án này cũng mang lại lợi ích xã hội, giúp giới trẻ kết nối, chia sẻ trải nghiệm và khám phá những địa điểm mới, qua sự hỗ trợ từ phía backend của ứng dụng.
- Đây là cơ hội áp dụng những kiến thức học được vào thực tế, xây dựng một ứng dụng hữu ích và đồng thời phát triển bản thân trong môi trường làm việc thực tế. Điều này không chỉ mang lại lợi ích cá nhân mà còn đóng góp vào sự phát triển và kết nối trong cộng đồng.

2. Mục tiêu của đề tài

- Nghiên cứu và triển khai các phương pháp và công nghệ hiện đại trong xây dựng backend API cho ứng dụng chia sẻ địa điểm
- Phát triển các API chính để đảm bảo tính ổn định và hiệu suất cao của hệ thống.
- Tối ưu hóa cơ sở dữ liệu để xử lý mượt mà lượng dữ liệu lớn và đảm bảo độ tin cậy.
- Tương tác hiệu quả với frontend và các anh chị trong team để đảm bảo trải nghiệm người dùng tốt nhất.
- Xây dựng hệ thống thông báo và tương tác giữa người dùng và hệ thống.

- Thực hiện kiểm thử và đảm bảo tính bảo mật của hệ thống.
- Áp dụng kiến thức học được vào việc xây dựng một ứng dụng thực tế và góp phần vào việc kết nối cộng đồng giới trẻ.

3. Phương pháp nghiên cứu

- Nghiên cứu tài liệu
 - Tham khảo tài liệu, tìm tòi kiến thức về RESTful API, NodeJs, MongoDB, mô hình MVC, kiến trúc phần mềm, bảo mật, ... từ các nguồn trên mạng.
- Phân tích và thiết kế
 - Phân tích yêu cầu của ứng dụng di động và trang web quản trị
 - Đọc hiểu tài liệu yêu cầu từ BA, lên ý tưởng thiết kế API
 - Thiết kế kiến trúc phần mềm cho cả ứng dụng di động và web quản trị
 - Thiết kế cơ sở dữ liệu với MongoDB, lựa chọn loại dữ liệu, thiết kế bảng dữ liệu.
- Lập trình và triển khai
 - Lập trình các API RESTful theo thiết kế
 - Tối ưu hóa code để đảm bảo hiệu suất cao
 - Xây dựng trang web quản trị với mô hình MVC
- Kiểm thử
 - Thực hiện kiểm thử API khi đã làm xong một chức năng
 - Sau khi tự phát hiện lỗi hoặc được Tester báo lỗi => Tiến hành khắc phục lỗi
- Bảo mật
 - Áp dụng các biện pháp bảo mật phù hợp cho API và web admin
 - Mã hóa dữ liệu nhạy cảm
- Tài liệu
 - Viết tài liệu hướng dẫn sử dụng API cho các nhà phát triển khác, đặc biệt là Frontend

4. Đối tượng và phạm vi nghiên cứu

- Đối tượng nghiên cứu
 - Hệ thống chia sẻ và kết nối địa điểm yêu thích: Hệ thống được xây dựng nhằm mục đích chia sẻ địa điểm giữa người dùng với nhau, đặc biệt là những người trẻ tuổi có đam mê và sở thích du lịch
 - Thành phần hệ thống:
 - Ứng dụng di động: Cho phép người dùng chia sẻ địa điểm hiện tại, tìm kiếm địa điểm khác, tương tác với nhau
 - Web quản trị: Cung cấp chức năng quản lý hệ thống, quản lý tài khoản, quản lý dữ liệu
- Phạm vi nghiên cứu
 - Backend API:
 - Nghiên cứu và triển khai các phương pháp và công nghệ hiện đại để xây dựng backend API cho ứng dụng chia sẻ địa điểm
 - Phát triển các API chính để đảm bảo tính ổn định và hiệu suất cao của hệ thống
 - Tối ưu hóa cơ sở dữ liệu để xử lý mượt mà lượng dữ liệu lớn và đảm bảo độ tin cậy
 - Tương tác hiệu quả với frontend và các thành viên trong nhóm để đảm bảo trải nghiệm người dùng tốt nhất
 - Sử dụng NodeJS và MongoDB để phát triển API
 - Áp dụng kiến thức về RESTful API, kiến trúc phần mềm, bảo mật để xây dựng API an toàn và hiệu quả
 - Mô hình MVC cho Web Admin
 - Xây dựng hệ thống thông báo và tương tác giữa người dùng và hệ thống
 - Sử dụng mô hình MVC để tổ chức code web admin
 - Sử dụng ExpressJS và EJS để xây dựng web admin
 - Kiểm thử và bảo mật
 - Thực hiện kiểm thử API và web admin để đảm bảo chất lượng hệ thống

- Áp dụng các biện pháp bảo mật phù hợp cho API và web admin
- Mã hóa dữ liệu nhạy cảm

5. Nội dung của đề tài

Đề tài được tổ chức gồm phần mở đầu, 4 chương nội dung và phần kết luận và hướng phát triển

- Mở đầu
- **Chương 1:** Tổng quan về đơn vị thực tập
- **Chương 2:** Cơ sở lý thuyết
- **Chương 3:** Phân tích hệ thống
- **Chương 4:** Thiết kế hệ thống
- **Chương 5:** Xây dựng hệ thống
- Kết luận và hướng phát triển

CHƯƠNG 1. TỔNG QUAN VỀ ĐƠN VỊ THỰC TẬP

1.1. Giới thiệu chung về Wiicamp

- Wiicamp là một công ty phát triển phần mềm tập trung vào việc mang lại giá trị cho khách hàng bằng cách cung cấp các giải pháp phần mềm phù hợp nhất với doanh nghiệp của bạn thông qua việc đồng hành và hiểu rõ hệ thống của bạn. Cùng với đội ngũ nhân viên trẻ tràn đầy năng lượng, sáng tạo cùng hướng tới mục tiêu làm hài lòng khách hàng
- Doanh nghiệp Wiicamp được thành lập 06/03/2020 [1]
- Địa chỉ: 51 Trần Văn Trứ, Hải Châu, Đà Nẵng, Việt Nam



Hình 1.1 Logo công ty Wiicamp

Logo Wiicamp (Hình 1.1) được thiết kế bởi công ty Wiicamp

1.2. Tầm nhìn, sứ mệnh

“Trở thành người bạn đồng hành tốt nhất với khách hàng”. Chúng tôi đồng hành cùng các doanh nghiệp khởi nghiệp tại tất cả các thời điểm của dự án để đảm bảo thành công của họ

1.3. Dịch vụ

- Phát triển website:
 - Trong kỷ nguyên công nghệ bùng nổ như hiện nay, thiết lập một hình ảnh trực tuyến ấn tượng là điều không thể bỏ qua. Wiicamp tự hào là đơn vị tiên phong trong lĩnh vực thiết kế website, nơi giao thoa hoàn

hảo giữa tính thẩm mỹ và chức năng tối ưu, giúp thương hiệu của bạn bứt phá và tạo dấu ấn mạnh mẽ

- Tại Wiicamp, sự hài lòng của khách hàng là kim chỉ nam:
 - Website phản hồi linh hoạt: Chuyển thăm liền mạch trên mọi thiết bị, từ desktop đến di động, đảm bảo trải nghiệm người dùng tuyệt vời
 - SEO đỉnh cao: Thu hút khách hàng tiềm năng qua công cụ tìm kiếm, đưa thương hiệu của bạn lên trang nhất Google
 - Giao diện trực quan, dễ sử dụng: Dẫn dắt khách hàng khám phá nội dung một cách mượt mà, thúc đẩy tương tác và chuyển đổi hiệu quả
 - Tốc độ vượt trội: Không còn lo lắng chờ đợi, website của bạn sẽ load nhanh chóng, giữ chân khách hàng ở lại
 - Giải pháp tùy biến, mở rộng: Chúng tôi thiết kế website phù hợp với nhu cầu riêng biệt của bạn, sẵn sàng đồng hành cùng sự phát triển của thương hiệu
 - Yếu tố tương tác đa dạng: Nâng cao tính hấp dẫn với hình ảnh động, video, tạo nội dung lôi cuốn và truyền tải thông tin hiệu quả
- Phát triển ứng dụng di động:
 - Nâng cao tương tác, thúc đẩy tăng trưởng với ứng dụng di động được thiết kế chuyên nghiệp
 - Tính năng và chức năng cốt lõi:
 - Thiết kế lấy người dùng làm trung tâm: Ứng dụng của chúng tôi ưu tiên trải nghiệm người dùng, tạo ra giao diện đẹp mắt và đảm bảo tương tác liền mạch
 - Công nghệ tiên tiến: Giữ vững vị thế dẫn đầu bằng cách tích hợp những công nghệ mới nhất vào ứng dụng di động của bạn
 - Tương thích đa nền tảng: Trong một bối cảnh kỹ thuật số đa dạng, việc tiếp cận người dùng trên nhiều thiết bị khác nhau là điều tối quan trọng

- Giao diện trực quan, dễ sử dụng: Giữ chân người dùng nhờ sự dễ dàng sử dụng của ứng dụng
- Khả năng mở rộng: Khi lượng người dùng của bạn tăng lên, ứng dụng của bạn cũng nên phát triển tương ứng
- Các biện pháp bảo mật dữ liệu: Bảo mật dữ liệu người dùng là ưu tiên hàng đầu của Wiicamp
- Phát triển phần mềm tùy chỉnh:
 - Đây là dịch vụ thiết kế và xây dựng phần mềm theo yêu cầu riêng của doanh nghiệp, đáp ứng những nhu cầu và mục tiêu cụ thể
 - Điểm khác biệt của Phát triển Phần mềm Tùy chỉnh:
 - Khác biệt: Khác với phần mềm bán sẵn, phần mềm được phát triển riêng biệt dựa trên nhu cầu và đặc thù của từng doanh nghiệp
 - Phù hợp: Giải quyết những vấn đề độc đáo mà phần mềm bán sẵn không đáp ứng được
 - Tối ưu hóa: Tăng hiệu quả hoạt động, tiết kiệm chi phí và nâng cao năng suất cho doanh nghiệp
 - Lợi ích của Phát triển Phần mềm Tùy chỉnh:
 - Phù hợp hoàn hảo: Đáp ứng chính xác nhu cầu của doanh nghiệp, không lãng phí tính năng không cần thiết
 - Tăng hiệu quả: Tự động hóa quy trình, tiết kiệm thời gian và chi phí vận hành
 - Lợi thế cạnh tranh: Tạo ra giải pháp độc đáo, giúp doanh nghiệp nổi bật so với đối thủ
 - Khả năng mở rộng: Dễ dàng nâng cấp và mở rộng theo nhu cầu phát triển của doanh nghiệp
 - Đến với dịch vụ phát triển phần mềm tùy chỉnh ở Wiicamp khi:
 - Doanh nghiệp cần giải pháp phần mềm đáp ứng nhu cầu đặc thù, không có phần mềm bán sẵn phù hợp
 - Doanh nghiệp muốn tối ưu hóa quy trình và nâng cao hiệu quả hoạt động

- Doanh nghiệp muốn tạo ra lợi thế cạnh tranh với giải pháp phần mềm độc đáo
- Thiết kế Web và Ứng dụng
 - Đây là lĩnh vực bao gồm việc tạo ra giao diện và trải nghiệm người dùng (UI/UX) cho các trang web và ứng dụng di động
 - Mục tiêu của Thiết kế Web & Ứng dụng:
 - Tạo ra giao diện đẹp mắt và thu hút: Thu hút sự chú ý của người dùng và tạo ấn tượng ban đầu tốt đẹp
 - Đảm bảo trải nghiệm người dùng (UX) tốt: Dễ sử dụng, điều hướng dễ dàng và đáp ứng nhu cầu của người dùng
 - Truyền tải thông tin hiệu quả: Giúp người dùng dễ dàng tìm thấy thông tin họ cần
 - Thúc đẩy hành động: Khuyến khích người dùng thực hiện các hành động mong muốn, như mua hàng, đăng ký hoặc liên hệ
 - Các yếu tố quan trọng trong Thiết kế Web & Ứng dụng:
 - Bố cục: Sắp xếp các yếu tố trên trang web hoặc ứng dụng một cách hợp lý và khoa học
 - Màu sắc: Sử dụng màu sắc phù hợp với thương hiệu và mục tiêu của trang web hoặc ứng dụng
 - Hình ảnh: Sử dụng hình ảnh chất lượng cao và phù hợp với nội dung
 - Kiểu chữ: Lựa chọn kiểu chữ dễ đọc và phù hợp với phong cách chung của trang web hoặc ứng dụng
 - Tính tương tác: Tạo ra các yếu tố tương tác để thu hút người dùng và giữ chân họ trên trang web hoặc ứng dụng
 - Lợi ích của việc đầu tư vào Thiết kế Web & Ứng dụng:
 - Tăng tỷ lệ chuyển đổi: Giao diện đẹp mắt và trải nghiệm người dùng tốt sẽ khuyến khích người dùng thực hiện hành động mong muốn
 - Nâng cao thương hiệu: Thiết kế chuyên nghiệp sẽ giúp tạo dựng hình ảnh thương hiệu uy tín và chuyên nghiệp

- Tăng khả năng cạnh tranh: Trong thời đại ngày nay, một trang web hoặc ứng dụng được thiết kế tốt là yếu tố quan trọng để cạnh tranh trong thị trường trực tuyến
- Hơn cả một nhà cung cấp dịch vụ, Wiicamp là đối tác đáng tin cậy trên hành trình số của các doanh nghiệp:
 - Đội ngũ chuyên gia: Sở hữu kiến thức sâu rộng, giàu kinh nghiệm, thấu hiểu từng nhu cầu của bạn
 - Quy trình minh bạch: Giao tiếp chặt chẽ, cập nhật tiến độ liên tục, đảm bảo sự hài lòng tuyệt đối
 - Hỗ trợ tận tâm: Đội ngũ chăm sóc khách hàng luôn sẵn sàng giải đáp mọi thắc mắc, hỗ trợ kịp thời
- Đến với Wiicamp để:
 - Nâng cao thương hiệu: Tạo website chuyên nghiệp, thu hút ánh nhìn, truyền tải giá trị cốt lõi đến khách hàng.
 - Thúc đẩy doanh thu: Tăng tương tác, gia tăng chuyển đổi, tối ưu hóa lợi nhuận
 - Tiết kiệm thời gian: Giao phó mọi công việc cho chúng tôi, tập trung vào thế mạnh của doanh nghiệp

CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1. JavaScript

JavaScript ra đời vào năm 1995 với mục đích nhúng các chương trình nhỏ vào trang web trên trình duyệt Netscape Navigator. Kể từ đó, ngôn ngữ này đã được tất cả các trình duyệt web đồ họa lớn khác chấp nhận. JavaScript chính là nền tảng giúp tạo ra các ứng dụng web hiện đại - những ứng dụng cho phép bạn tương tác trực tiếp mà không cần tải lại toàn bộ trang cho mỗi thao tác. JavaScript cũng được sử dụng trong các trang web truyền thống để cung cấp các tính năng tương tác và hiệu ứng thú vị



Hình 2.1 Hình ảnh về JavaScript

2.2. NodeJs

2.2.1. Giới thiệu về NodeJs

Ra đời vào năm 2009 bởi Ryan Dahl, Node.js xuất hiện như một giải pháp đột phá cho vấn đề tốc độ ì ạch của các web server thời bấy giờ. Hầu hết các server lúc đó đều bị "đóng băng" mỗi khi thực hiện tác vụ I/O (Input/Output) như đọc dữ liệu từ hệ thống file hoặc truy cập mạng, điều này làm giảm đáng kể hiệu suất xử lý. Chẳng hạn, khi tiếp nhận upload file, server sẽ không thể xử lý bất kỳ

yêu cầu nào khác cho đến khi việc upload hoàn tất. Lúc đó, Dahl đang chủ yếu làm việc với Ruby, và mô hình thống trị cho các ứng dụng web là sử dụng một pool các tiến trình Ruby được một web server (ví dụ như Nginx) quản lý. Nếu một tiến trình Ruby bị "dính" do việc upload, Nginx sẽ chuyển yêu cầu sang một tiến trình khác. Node.js thay đổi mô hình này bằng cách biến tất cả các tác vụ I/O thành non-blocking (không chặn) và asynchronous (không đồng bộ). Điều này cho phép các web server viết bằng Node.js xử lý hàng ngàn yêu cầu đồng thời - các yêu cầu tiếp theo không phải đợi các yêu cầu trước hoàn thành. Bản demo đầu tiên của Node.js đã tạo được tiếng vang lớn vì đây là lần đầu tiên developer có thể dễ dàng tạo ra một web server riêng với hiệu suất ấn tượng. Theo thời gian, Node.js không chỉ dừng lại ở mảng web server mà còn tỏa sáng như một ngôn ngữ lập trình server-side linh hoạt và mạnh mẽ. Nó có thể thực hiện bất kỳ tác vụ nào thông thường được thực hiện bởi Python, Ruby, Perl và PHP, nhưng nhanh hơn, tốn ít bộ nhớ hơn và trong nhiều trường hợp có các API tốt hơn cho các system call. Ví dụ, với Node.js, chúng ta có thể tạo các HTTP và TCP server chỉ với vài dòng code.



Hình 2.2 Hình ảnh về NodeJs

2.2.2. Ưu điểm của NodeJs

- Node.js ra đời để giải quyết vấn đề tốc độ của web server
- Sử dụng mô hình non-blocking và asynchronous I/O để xử lý nhiều yêu cầu cùng lúc

- Linh hoạt, mạnh mẽ và có thể thay thế các ngôn ngữ lập trình server-side truyền thống

2.2.3. Khi nào nên sử dụng NodeJs

NodeJs là lựa chọn tuyệt vời cho bất kỳ tác vụ hoặc dự án nào vốn thường sử dụng ngôn ngữ lập trình động như Python, PHP, Perl hay Ruby. Tuy nhiên, Node.js đặc biệt phát huy sức mạnh trong các lĩnh vực:

- API HTTP: Node.js được thiết kế để xây dựng web server đỉnh cao, và nó không làm chúng ta thất vọng. Các API HTTP và web server xây dựng bằng Node.js thường có hiệu năng vượt trội so với các ngôn ngữ động khác
- HTTP: Node.js được thiết kế để xây dựng web server đỉnh cao, và nó không làm chúng ta thất vọng. Các API HTTP và web server xây dựng bằng Node.js thường có hiệu năng vượt trội so với các ngôn ngữ động khác
- HTTP: Node.js được thiết kế để xây dựng web server đỉnh cao, và nó không làm chúng ta thất vọng. Các API HTTP và web server xây dựng bằng Node.js thường có hiệu năng vượt trội so với các ngôn ngữ động khác
- Ứng dụng desktop đa nền tảng: Electron cho phép xây dựng ứng dụng desktop đa nền tảng bằng JavaScript, HTML và CSS. Nó kết hợp giao diện người dùng của trình duyệt với Node.js. Nhờ Node.js, chúng ta có thể truy cập hệ thống file, mạng và các tài nguyên khác của hệ điều hành

2.3. ExpressJs

ExpressJs được mệnh danh là bộ khung web nhanh gọn, linh hoạt và tối giản dành cho Node.js, được sử dụng rộng rãi trong các môi trường sản xuất trên toàn thế giới. Express phổ biến đến mức nhiều người dùng Node.js gắn liền với nó hơn cả việc sử dụng Node.js độc lập. Có thể coi Express như người anh em thay thế tiện lợi cho module http cốt lõi của Node.js. Điểm khác biệt lớn nhất giữa Express và http là khả năng định tuyến. Không giống như http,

Express sở hữu bộ định tuyến tích hợp sẵn, giúp đơn giản hóa việc xây dựng web server.

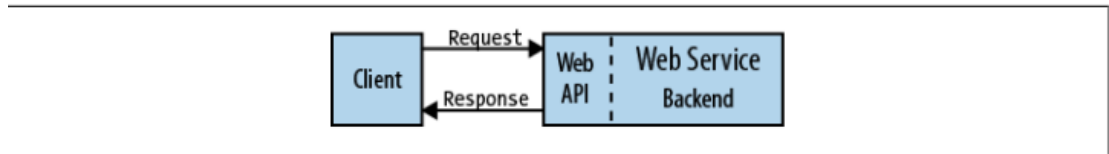


Hình 2.3 Hình ảnh về ExpressJs

2.4. REST APIs

2.4.1. Giới thiệu về REST APIs

- Dịch vụ Web hoạt động giống như những máy chủ web được xây dựng với mục đích riêng, hỗ trợ cho một website hoặc bất kỳ ứng dụng nào khác. Các chương trình máy tính (client) sử dụng các bộ giao diện lập trình ứng dụng (API) để trao đổi thông tin với Dịch vụ Web. Nói một cách đơn giản, API cung cấp một bộ dữ liệu và các chức năng giống như "ngôn ngữ chung" để các chương trình máy tính có thể tương tác và trao đổi thông tin với nhau
- Hình 2.4 mô tả API Web như "bộ mặt" của Dịch vụ Web, nơi lắng nghe và trả lời trực tiếp các yêu cầu từ các chương trình khác. Kiểu thiết kế API phổ biến hiện nay là REST, giúp xây dựng các API cho Dịch vụ Web linh hoạt và hiệu quả. Một API Web tuân theo phong cách thiết kế REST được gọi là REST APIs



Hình 2.4 Hình ảnh về web API

2.4.2. Sự cần thiết của REST APIs trong phát triển ứng dụng web

REST APIs đóng vai trò quan trọng trong việc biến Dịch vụ Web trở nên "dễ dàng sử dụng" hơn. REST APIs được xây dựng dựa trên các "nguồn lực" (resource) liên kết với nhau. Tập hợp các nguồn lực này được gọi là "mô hình nguồn lực" của REST APIs. Các REST APIs được thiết kế tốt sẽ thu hút các nhà phát triển của các chương trình khác (client) sử dụng Dịch vụ Web. Trong môi trường cạnh tranh hiện nay, một REST API được thiết kế đẹp mắt và dễ sử dụng là yếu tố then chốt để thu hút người dùng

2.4.3. Thiết kế REST APIs

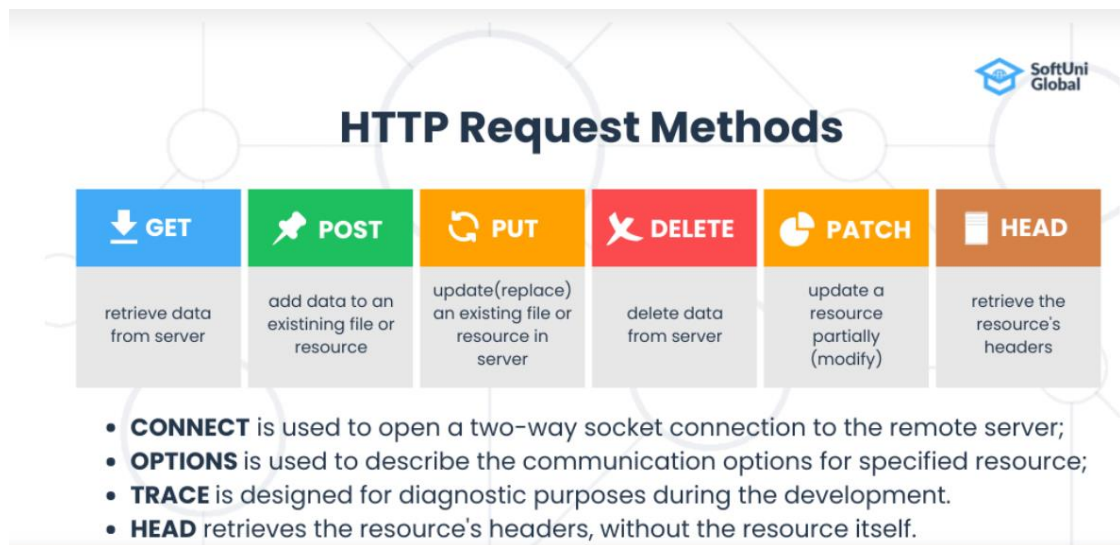
Đối với nhiều người, thiết kế REST APIs đôi khi có cảm giác giống như nghệ thuật hơn là khoa học. Một số phương pháp hay cho thiết kế REST APIs được ngầm định trong chuẩn HTTP, trong khi những cách tiếp cận bán chuẩn khác đã xuất hiện trong vài năm qua. Tuy nhiên, ngày nay chúng ta vẫn phải tiếp tục tìm kiếm câu trả lời cho một loạt các câu hỏi, chẳng hạn như:

- Khi nào các phân đoạn đường dẫn URI được đặt tên với danh từ số nhiều?
- Phương thức yêu cầu nào nên được sử dụng để cập nhật trạng thái tài nguyên?
- Làm thế nào để ánh xạ các hoạt động ngoài CRUD (Tạo, Đọc, Cập nhật, Xóa) vào URI của tôi?
- Mã trạng thái phản hồi HTTP phù hợp cho một tình huống nhất định là gì?
- Làm thế nào để tôi có thể quản lý các phiên bản của biểu diễn trạng thái tài nguyên?
- Làm thế nào để tôi cấu trúc một liên kết siêu văn bản trong JSON?

2.4.4. Các quy tắc định dạng URL trong REST APIs

- Phải sử dụng dấu gạch chéo (/): Ký tự dấu gạch chéo (/) được sử dụng trong phần path của URI để chỉ mối quan hệ theo hệ thống phân cấp giữa các tài nguyên
- Không nên bao gồm dấu gạch chéo (/) ở cuối URI: Là ký tự cuối cùng trong path của URI, dấu gạch chéo (/) không bổ sung giá trị ngữ nghĩa và có thể gây nhầm lẫn. REST APIs không nên mong đợi một dấu gạch chéo ở cuối và không nên bao gồm chúng trong các liên kết mà chúng cung cấp cho client
- Sử dụng dấu gạch ngang (-) để cải thiện khả năng đọc của URI: Để làm cho URI dễ dàng cho mọi người đọc và hiểu nên sử dụng dấu gạch ngang (-) để cải thiện khả năng đọc của tên trong các phân đoạn path dài
- Không nên sử dụng dấu gạch dưới (_) trong URI: Các ứng dụng xem văn bản (trình duyệt, trình soạn thảo, v.v.) thường gạch dưới URI để cung cấp tín hiệu trực quan cho biết chúng có thể nhấp được. Tùy thuộc vào font chữ của ứng dụng, ký tự dấu gạch dưới (_) có thể bị che khuất một phần hoặc hoàn toàn bị ẩn bởi gạch dưới này. Để tránh nhầm lẫn này, Sử dụng dấu gạch ngang (-) thay vì dấu gạch dưới
- Nên ưu tiên sử dụng chữ cái thường trong path của URI: Khi thuận tiện, nên ưu tiên sử dụng chữ cái thường trong path của URI vì chữ cái hoa đôi khi có thể gây ra sự cố. RFC 3986 định nghĩa URI phân biệt chữ hoa chữ thường ngoại trừ các thành phần scheme và host
- Không nên bao gồm phần mở rộng tệp trong URI: Trên Web, ký tự dấu chấm (.) thường được dùng để phân tách phần tên tệp và phần mở rộng của URI. Một REST API không nên bao gồm các phần mở

2.4.5. Phương thức yêu cầu (Request Method) trong REST APIs



Hình 2.5 Hình ảnh về HTTP Request Methods

- Mỗi phương thức HTTP có ý nghĩa ngữ nghĩa được xác định rõ ràng trong ngữ cảnh của mô hình tài nguyên của REST APIs:
 - GET được sử dụng để lấy một bản thể hiện trạng thái của tài nguyên
 - PUT/PATCH được sử dụng để thêm một tài nguyên mới vào kho lưu trữ hoặc cập nhật một tài nguyên
 - DELETE xóa một tài nguyên khỏi tài nguyên cha của nó
 - POST được sử dụng để tạo một tài nguyên mới trong một tập hợp và thực thi các bộ điều khiển
- Các quy tắc:
 - GET và POST không được sử dụng để thực hiện các phương thức yêu cầu khác
 - GET phải được sử dụng để truy xuất một biểu diễn của tài nguyên
 - PUT phải được sử dụng để cả chèn và cập nhật tài nguyên được lưu trữ
 - PUT phải được sử dụng để cập nhật tài nguyên có thể thay đổi
 - POST phải được sử dụng để tạo tài nguyên mới trong một tập hợp
 - POST phải được sử dụng để thực thi trình điều khiển
 - DELETE phải được sử dụng để xóa tài nguyên khỏi tài nguyên cha

2.4.6. Mã trạng thái phản hồi (*Response Status Codes*) trong *REST APIs*

- REST API sử dụng phần Status-Line (Dòng trạng thái) của tin nhắn phản hồi HTTP để thông báo cho client về kết quả tổng thể của yêu cầu của họ

Category	Description
1xx: Informational	Communicates transfer protocol-level information.
2xx: Success	Indicates that the client's request was accepted successfully.
3xx: Redirection	Indicates that the client must take some additional action in order to complete their request.
4xx: Client Error	This category of error status codes points the finger at clients.
5xx: Server Error	The server takes responsibility for these error status codes.

Hình 2.6 Hình ảnh danh mục mã trạng thái phản hồi

- Các quy tắc trả về mã
 - 200 ("OK") nên được sử dụng để biểu thị thành công không cụ thể
 - 200 ("OK") không được sử dụng để truyền đạt lỗi trong nội dung phản hồi. Mã này được dùng để báo hiệu yêu cầu của client đã được xử lý thành công
 - 201 ("Created") phải được sử dụng để chỉ việc tạo tài nguyên thành công. Mã này được dùng khi một tài nguyên mới được tạo trên server do yêu cầu của client
 - 202 ("Accepted") phải được sử dụng để chỉ việc bắt đầu thành công một hành động bất đồng bộ. Mã này được dùng để báo hiệu server đã chấp nhận yêu cầu và đang xử lý nó theo phương thức bất đồng bộ (asynchronous)
 - 204 ("No Content") nên được sử dụng khi nội dung phản hồi được cố tình để trống. Mã này được dùng khi server không có gì để gửi lại cho client ngoại trừ header của phản hồi
 - 301 ("Moved Permanently") nên được sử dụng để di chuyển tài nguyên. Mã này được dùng để thông báo cho client rằng tài nguyên được yêu cầu đã được chuyển đến một vị trí khác (URI mới)
 - 302 ("Found") không nên sử dụng. Mã này có nghĩa tương tự như 303 ("See Other") nhưng đã lỗi thời trong giao thức HTTP hiện đại

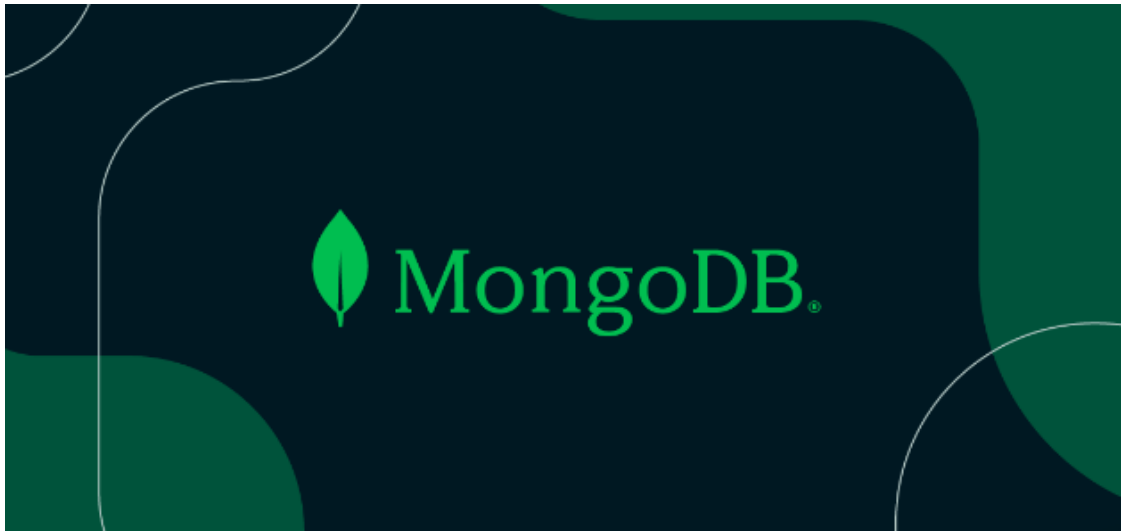
- 303 ("See Other") nên được sử dụng để tham chiếu client đến một URI khác. Mã này được dùng để hướng dẫn client đến một URI khác để hoàn thành yêu cầu
- 304 ("Not Modified") nên được sử dụng để tiết kiệm băng thông. Mã này được dùng để thông báo cho client rằng tài nguyên theo yêu cầu chưa được thay đổi kể từ lần truy cập trước đó (dựa trên header)
- 307 ("Temporary Redirect") nên được sử dụng để yêu cầu client gửi lại yêu cầu tới một URI khác. Mã này được dùng cho các tình huống tạm thời, yêu cầu client truy cập vào một URI khác để xử lý yêu cầu
- 400 ("Bad Request") có thể được sử dụng để chỉ lỗi không cụ thể. Mã này được dùng khi yêu cầu của client không hợp lệ về cú pháp hoặc ngữ nghĩa
- 401 ("Unauthorized") phải được sử dụng khi có vấn đề với thông tin xác thực của client. Mã này được dùng để báo hiệu client cần cung cấp thông tin xác thực hợp lệ (ví dụ như username và password)
- 403 ("Forbidden") nên được sử dụng để cấm truy cập bất kể trạng thái ủy quyền. Mã này được dùng để thông báo cho client rằng họ không được phép truy cập vào tài nguyên, ngay cả khi đã được xác thực
- 404 ("Not Found") phải được sử dụng khi URI của client không thể ánh xạ đến một tài nguyên. Mã này được dùng để báo hiệu rằng server không thể tìm thấy tài nguyên theo yêu cầu
- 405 ("Method Not Allowed") phải được sử dụng khi phương thức HTTP không được hỗ trợ. Mã này được dùng để thông báo cho client rằng phương thức HTTP được sử dụng trong yêu cầu không được hỗ trợ bởi tài nguyên
- 406 ("Not Acceptable") phải được sử dụng khi loại media được yêu cầu không thể được phục vụ. Mã này được dùng để báo hiệu server không thể cung cấp nội dung theo định dạng media mà client yêu cầu
- 409 ("Conflict") nên được sử dụng để chỉ vi phạm trạng thái tài nguyên. Mã này được dùng để báo hiệu rằng yêu cầu của client gây ra xung đột với trạng thái hiện tại của tài nguyên

- 412 ("Precondition Failed") nên được sử dụng để hỗ trợ các hoạt động có điều kiện. Mã này được dùng để báo hiệu một điều kiện tiên quyết (precondition) trong yêu cầu của client đã thất bại
- 415 ("Unsupported Media Type") phải được sử dụng khi loại media của payload yêu cầu không thể được xử lý. Mã này được dùng để báo hiệu server không thể xử lý định dạng media của dữ liệu được gửi kèm trong yêu cầu của client
- 500 ("Internal Server Error") nên được sử dụng để chỉ trục trặc của API. Mã này được dùng trong trường hợp server gặp phải lỗi nội bộ và không thể hoàn thành yêu cầu

2.5. MongoDB

2.5.1. Giới thiệu về MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu theo kiểu tài liệu (document-oriented), khác biệt hoàn toàn so với mô hình quan hệ (relational) truyền thống. Lý do chính cho sự thay đổi này là để việc mở rộng hệ thống (scale out) dễ dàng hơn, nhưng bên cạnh đó còn nhiều ưu điểm khác. MongoDB thay thế khái niệm "bảng dữ liệu" (row) bằng một mô hình linh hoạt hơn - "tài liệu" (document). Bằng cách cho phép nhúng các tài liệu và mảng con, MongoDB giúp biểu diễn các mối quan hệ phức tạp theo dạng phân cấp chỉ trong một bản ghi duy nhất. Cách tiếp cận này phù hợp tự nhiên với cách các lập trình viên sử dụng ngôn ngữ hướng đối tượng hiện đại để suy nghĩ về dữ liệu của họ. MongoDB không có schema (cấu trúc) được định sẵn: các khóa (key) và giá trị (value) của tài liệu không bị giới hạn về kiểu dữ liệu hoặc kích thước. Nhờ đó, việc thêm hoặc xóa các trường dữ liệu theo nhu cầu trở nên dễ dàng hơn. Điều này thường giúp đẩy nhanh quá trình phát triển vì các lập trình viên có thể nhanh chóng lặp lại và thử nghiệm nhiều mô hình dữ liệu khác nhau để chọn ra lựa chọn tối ưu



Hình 2.7 Hình ảnh về MongoDB

2.5.2. Ưu điểm của MongoDB

- Mở rộng dễ dàng: Phù hợp cho các hệ thống cần lưu trữ lượng dữ liệu lớn và có khả năng mở rộng
- Linh hoạt: Cấu trúc dữ liệu linh hoạt, dễ dàng thay đổi theo yêu cầu
- Phát triển nhanh: Thêm/xóa trường dữ liệu đơn giản, hỗ trợ các lập trình viên thử nghiệm nhiều mô hình dữ liệu

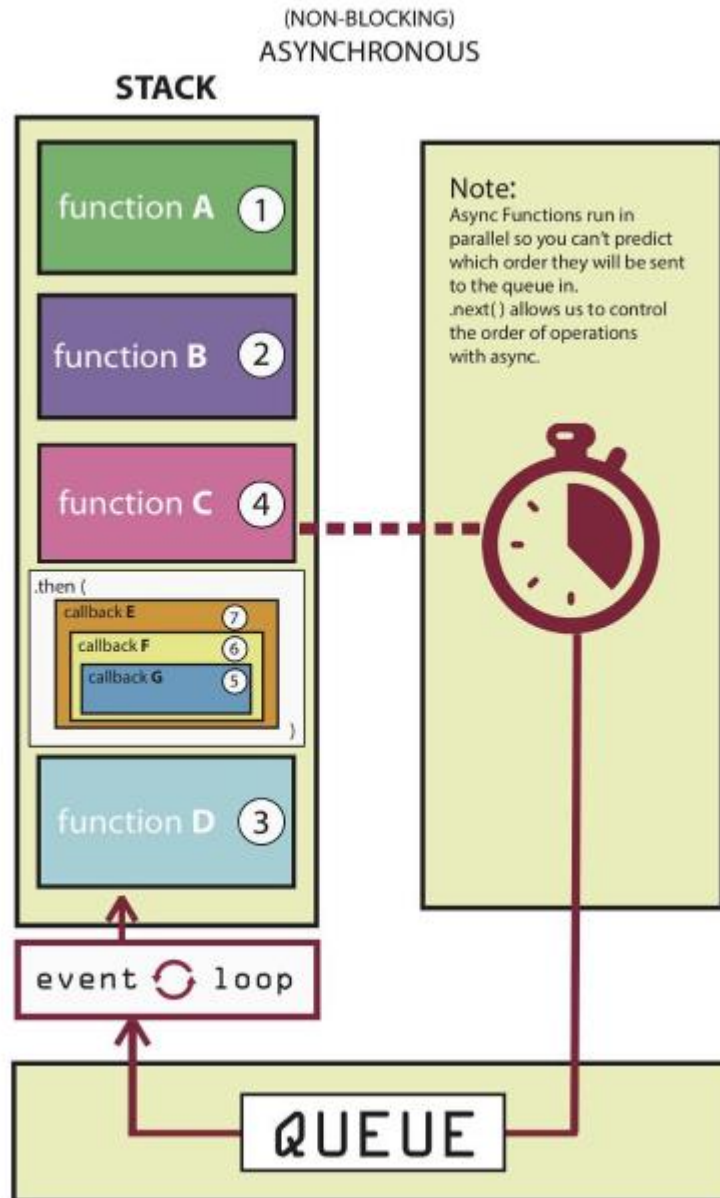
2.6. Đồng bộ và bất đồng bộ

2.6.1. Đồng bộ

Chương trình phải đợi yêu cầu thứ nhất hoàn thành, nhận kết quả rồi mới bắt đầu yêu cầu thứ hai. Tổng thời gian sẽ bằng tổng thời gian phản hồi của cả hai yêu cầu

2.6.2. Bất đồng bộ

Chương trình có thể bắt đầu cả hai yêu cầu cùng một lúc. Trong khi chờ đợi kết quả từ mạng, chương trình vẫn có thể thực hiện các tác vụ khác. Khi kết quả trả về, chương trình sẽ được thông báo và xử lý



Hình 2.8 Hình ảnh về xử lý bất đồng bộ

2.6.3. Ưu và nhược điểm của bất đồng bộ

- Ưu điểm: Cho phép tận dụng hiệu quả thời gian chờ đợi các tác vụ chậm. Chương trình phản hồi nhanh hơn, không bị dính vào các tác vụ lâu
- Nhược điểm: Logic chương trình có thể phức tạp hơn so với đồng bộ. Cần xử lý các sự kiện trả về kết quả bất đồng thời

CHƯƠNG 3. PHÂN TÍCH HỆ THỐNG

3.1. Tổng quan về ứng dụng và website quản trị hotspotter

3.1.1. Giới thiệu ứng dụng và website quản trị

a. Ứng dụng Hotspotter

Hotspotter không chỉ là ứng dụng chia sẻ địa điểm thông thường, mà còn là công kết nối cộng đồng, đặc biệt là giới trẻ, thông qua những trải nghiệm độc đáo và vị trí tuyệt vời trên khắp thế giới. Ứng dụng bao gồm những chức năng nổi bật sau:

- Người dùng đăng nhập, đăng kí bằng email, facebook, google, appleID
- Người dùng thay đổi thông tin cá nhân, ảnh đại diện, cập nhật quyền riêng tư cho tài khoản
- Xem địa điểm ở dạng bản đồ hoặc theo dạng danh sách
- Tìm kiếm địa điểm theo thẻ lọc
- Xem chi tiết thông tin địa điểm
- Người dùng tạo danh sách và lưu các địa điểm vào danh sách đã tạo
- Người dùng theo dõi trang cá nhân người dùng khác
- Hiện thị bản tin mới nhất ở trang bản tin
- Gửi thông báo cho người dùng
- Người dùng báo cáo những vấn đề của ứng dụng cho quản trị viên
- Người dùng gửi yêu cầu cần hỗ trợ cho quản trị viên

b. Website quản trị Hotspotter

Website quản trị Hotspotter là công cụ dành cho quản trị viên để quản lý mọi hoạt động của ứng dụng Hotspotter. Website bao gồm các module chính sau

- Quản trị địa điểm
- Quản lý các thẻ lọc địa điểm
- Quản lý tài khoản người dùng
- Quản lý yêu cầu giúp đỡ
- Quản lý báo cáo địa điểm

3.1.2. Module ứng dụng và website quản trị

STT	Module	Danh sách chức năng	Mô tả
1	Quản lý chứng thực	<ul style="list-style-type: none"> - Đăng nhập bằng email và mật khẩu - Đăng ký bằng email - Đăng ký, đăng nhập bằng Facebook - Đăng ký, đăng nhập bằng google - Đăng ký, đăng nhập bằng AppleID - Quên mật khẩu 	Cung cấp các chức năng liên quan đến xác thực người dùng trên hệ thống. Module này giúp người dùng đăng nhập và đăng ký tài khoản bằng nhiều phương thức khác nhau như email, Facebook, Google và AppleID. Ngoài ra, module còn cung cấp tính năng đặc biệt để khôi phục mật khẩu khi người dùng quên mật khẩu của mình
2	Xem địa điểm	<ul style="list-style-type: none"> - Xem danh sách địa điểm theo dạng bản đồ qua các biểu tượng - Xem danh sách địa điểm theo dạng danh sách - Xem chi tiết địa điểm 	Module cung cấp các chức năng liên quan đến xem thông tin về các địa điểm trên hệ thống. Module này cho phép người dùng khám phá danh sách các địa điểm thông qua giao diện bản đồ hoặc danh sách chi tiết

3	Tìm kiếm địa điểm	<ul style="list-style-type: none"> - Tìm kiếm địa điểm theo tên - Tìm kiếm địa điểm theo vị trí - Tìm kiếm địa điểm theo thẻ lọc - Tìm kiếm địa điểm theo danh sách của người dùng 	Module cung cấp các chức năng để người dùng có thể tìm kiếm các địa điểm dựa trên các tiêu chí khác nhau. Module này giúp người dùng dễ dàng tìm kiếm địa điểm theo tên, vị trí, thẻ lọc hoặc danh sách đã lưu trữ
4	Người dùng quản lý tài khoản cá nhân	<ul style="list-style-type: none"> - Cập nhật ảnh đại diện, ảnh bìa - Cập nhật tên, địa chỉ email, giới tính, tuổi - Cập nhật mật khẩu - Cài đặt tài khoản cá nhân ở chế độ riêng tư hoặc công khai - Cài đặt cho phép hoặc không cho phép các loại thông báo 	Module cung cấp các chức năng cho người dùng quản lý thông tin cá nhân của mình trên hệ thống. Module này giúp người dùng có thể cập nhật và quản lý thông tin cá nhân một cách dễ dàng và thuận tiện
5	Chia sẻ địa điểm địa điểm	<ul style="list-style-type: none"> - Chia sẻ địa điểm lên mạng xã hội 	Module này cho phép người dùng thực hiện các hoạt động tương tác với các địa điểm trên hệ thống. Giúp người dùng chia sẻ các địa điểm yêu thích lên các mạng xã hội

6	Quản lý ghi chú	<ul style="list-style-type: none"> - Xem danh sách ghi chú ở một địa điểm - Thêm, chỉnh sửa ghi chú cá nhân ở một địa điểm 	Module quản lý ghi chú cho phép người dùng tạo, xem và chỉnh sửa các ghi chú cá nhân liên quan đến các địa điểm trên hệ thống. Module này giúp người dùng ghi lại các thông tin, nhận xét hoặc kỷ niệm về một địa điểm cụ thể để dễ dàng tham khảo và quản lý sau này
7	Quản lý danh sách	<ul style="list-style-type: none"> - Tạo mới danh sách - Xem tất cả danh sách mặc định và danh sách người dùng tạo - Theo dõi danh sách của người dùng khác - Xem danh sách đã theo dõi - Lưu địa điểm vào danh sách - Xóa địa điểm ra khỏi danh sách - Chia sẻ danh sách lên các trang mạng xã hội - Nhân bản danh sách của cá nhân/ danh sách của người dùng mình đã theo dõi 	Module quản lý danh sách cho phép người dùng tạo, quản lý, và chia sẻ các danh sách trên hệ thống. Module này giúp người dùng tổ chức và lưu trữ các địa điểm yêu thích của họ theo cách thuận tiện và linh hoạt. Mỗi tài khoản khi tạo mới sẽ có 3 danh sách mặc định, người dùng có thể tạo thêm danh sách sau đó

		<ul style="list-style-type: none"> - Xóa danh sách người dùng tạo - Chỉnh sửa danh sách - Cài đặt cho phép người khác làm cộng tác viên của danh sách - Mời người dùng khác tham gia cộng tác vào danh sách mình tạo - Chấp nhận lời mời cộng tác vào danh sách của người dùng khác 	
8	Theo dõi và hủy theo dõi người dùng	<ul style="list-style-type: none"> - Xem danh sách tất cả người dùng có trên hệ thống - Theo dõi người dùng - Hủy theo dõi người dùng - Xem thông tin chi tiết người dùng - Xem hoạt động của người dùng đang theo dõi - Xem danh sách người dùng theo dõi mình/ mình đang theo dõi - Tìm kiếm người dùng có trên hệ thống 	Module Theo dõi và hủy theo dõi người dùng cung cấp các chức năng để người dùng quản lý việc theo dõi và theo dõi các người dùng khác trên hệ thống. Module này giúp người dùng theo dõi hoạt động của những người họ quan tâm và tìm kiếm thông tin về họ một cách dễ dàng

9	Bảng tin	- Hiện thị người dùng đang theo dõi lưu địa điểm vào danh sách	Module Bảng tin cung cấp một giao diện tổng quan cho người dùng xem các hoạt động mới nhất từ những người dùng mà họ đang theo dõi trên hệ thống. Module này giúp người dùng cập nhật thông tin và hoạt động từ cộng đồng của mình một cách dễ dàng
10	Thông báo	- Người dùng xem danh sách thông báo	Module thông báo cung cấp cho người dùng khả năng xem và quản lý danh sách thông báo trên hệ thống. Module này cho phép người dùng đánh dấu thông báo là đã đọc hoặc chưa đọc và tùy chỉnh cài đặt để cho phép hoặc không cho phép các loại thông báo cụ thể
11	Mua tài khoản trả phí	- Người dùng mua tài khoản trả phí	Module này cung cấp cho người dùng chức năng thanh toán để người dùng có thể tham gia trải nghiệm những chức năng cần trả phí của ứng dụng

Bảng 3.1 Module ứng dụng Hotspotter

- Những chức năng sinh viên không tham gia trực tiếp vào phát triển:

- Mua tài khoản trả phí

STT	Module	Danh sách chức năng	Mô tả
1	Quản lý chứng thực	<ul style="list-style-type: none"> - Đăng nhập bằng email và mật khẩu - Quên mật khẩu 	Module Quản lý chứng thực cung cấp các chức năng liên quan đến xác thực người dùng trên hệ thống. Nó bao gồm việc đăng nhập bằng email và mật khẩu, cũng như khả năng khôi phục mật khẩu trong trường hợp người dùng quên mật khẩu của mình
2	Quản lý người dùng	<ul style="list-style-type: none"> - Xem danh sách tất cả người dùng có trong hệ thống - Tìm kiếm người dùng - Khóa tài khoản người dùng - Xem chi tiết hồ sơ người dùng 	Module Quản lý người dùng cung cấp các chức năng quản lý và tương tác với tài khoản người dùng trên hệ thống. Nó bao gồm việc xem danh sách tất cả người dùng có trên hệ thống, tìm kiếm người dùng, khóa tài khoản người dùng, và xem chi tiết hồ sơ người dùng

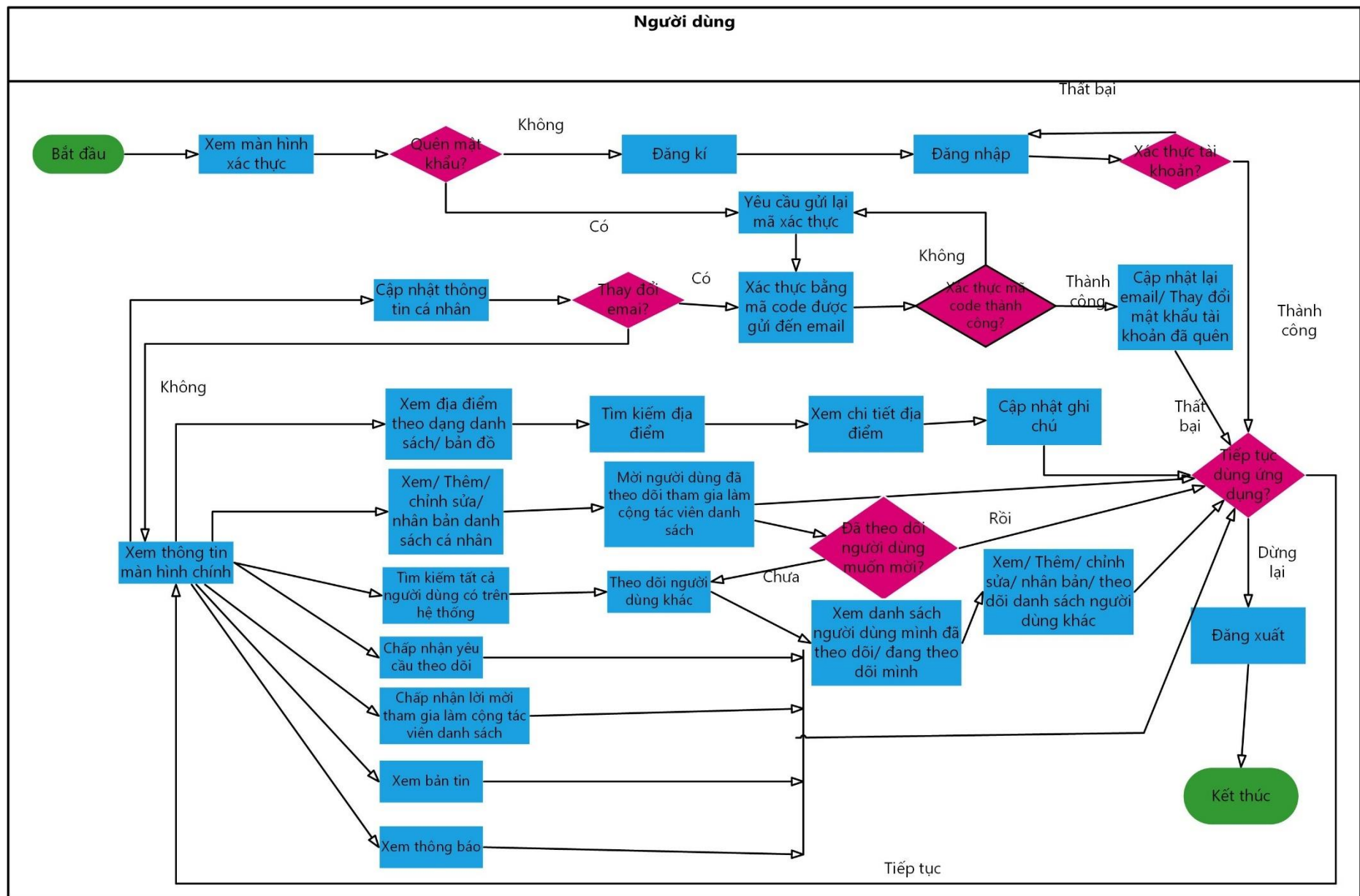
3	Quản lý địa điểm	<ul style="list-style-type: none"> - Xem, thêm, sửa, xóa địa điểm 	Module Quản lý địa điểm cung cấp các chức năng quản lý và tương tác với danh sách địa điểm trên hệ thống. Nó bao gồm việc xem danh sách địa điểm, thêm mới, sửa đổi thông tin, và xóa địa điểm từ cơ sở dữ liệu
4	Quản lý danh sách thẻ lọc	<ul style="list-style-type: none"> - Xem, thêm, sửa, xóa thẻ lọc 	Module quản lý danh sách thẻ lọc cung cấp các chức năng quan trọng như xem, thêm, sửa, và xóa thẻ lọc giúp người dùng có thể tùy chỉnh và quản lý danh sách thẻ lọc một cách linh hoạt và hiệu quả
5	Quản lý Admin	<ul style="list-style-type: none"> - Xem, thêm, sửa, xóa, chặn tài khoản admin - Thay đổi mật khẩu tài khoản admin 	Module này cho phép người dùng thực hiện các hoạt động tương tác với các địa điểm trên hệ thống. Giúp người dùng chia sẻ các địa điểm yêu thích lên các mạng xã hội
6	Quản lý ghi chú	<ul style="list-style-type: none"> - Xem danh sách ghi chú ở một địa điểm - Thêm, chỉnh sửa ghi chú cá nhân ở một địa điểm 	Module quản lý ghi chú cho phép người dùng tạo, xem và chỉnh sửa các ghi chú cá nhân liên quan đến các địa điểm trên hệ thống.

			Module này giúp người dùng ghi lại các thông tin, nhận xét hoặc kỷ niệm về một địa điểm cụ thể để dễ dàng tham khảo và quản lý sau này
--	--	--	--

Bảng 3.2 Module website quản trị Hotspotter

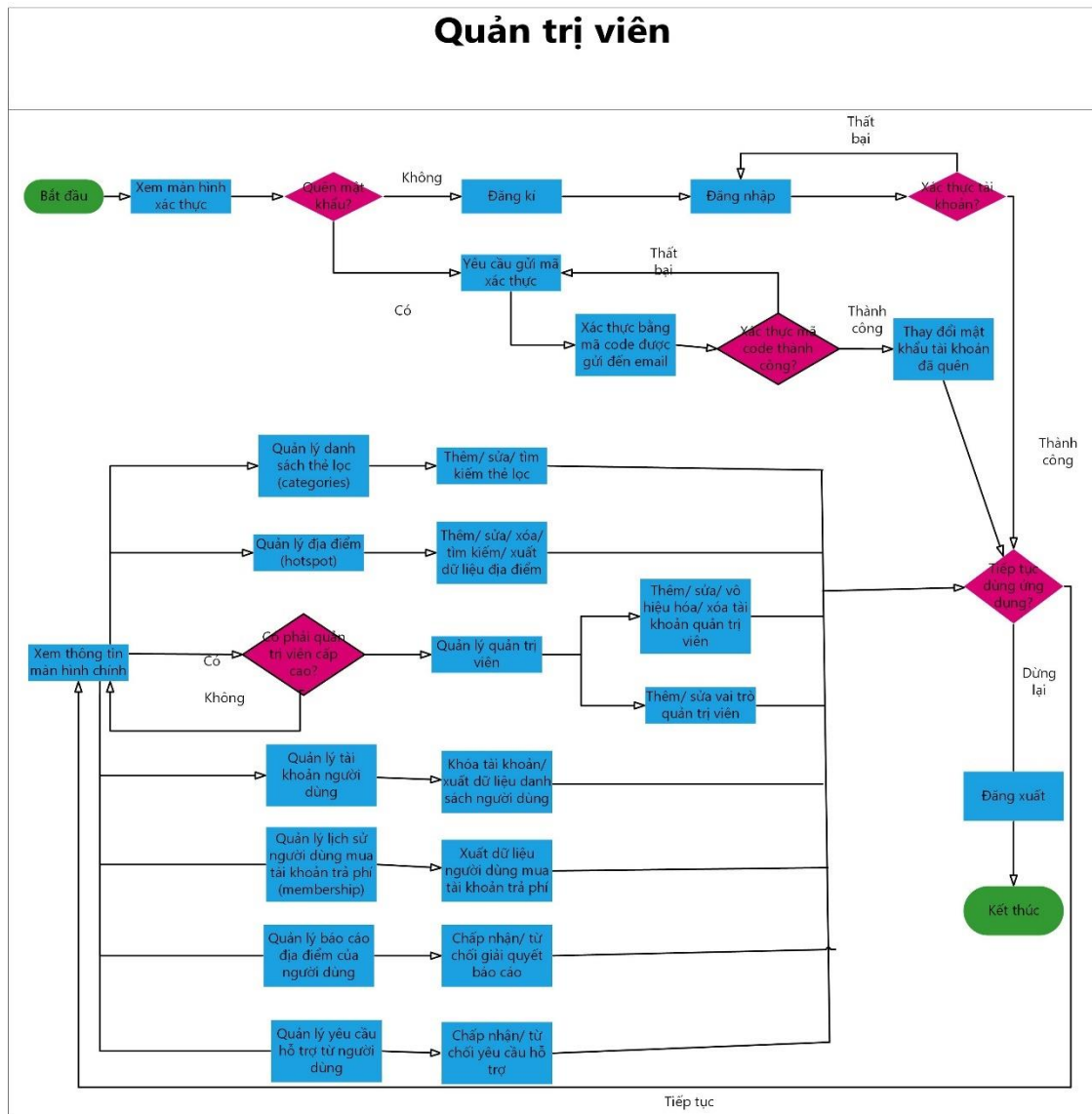
3.2. Workflow

3.2.1. *Workflow ứng dụng Hotspotter*



Hình 3.1 Hình workflow ứng dụng Hotspotter

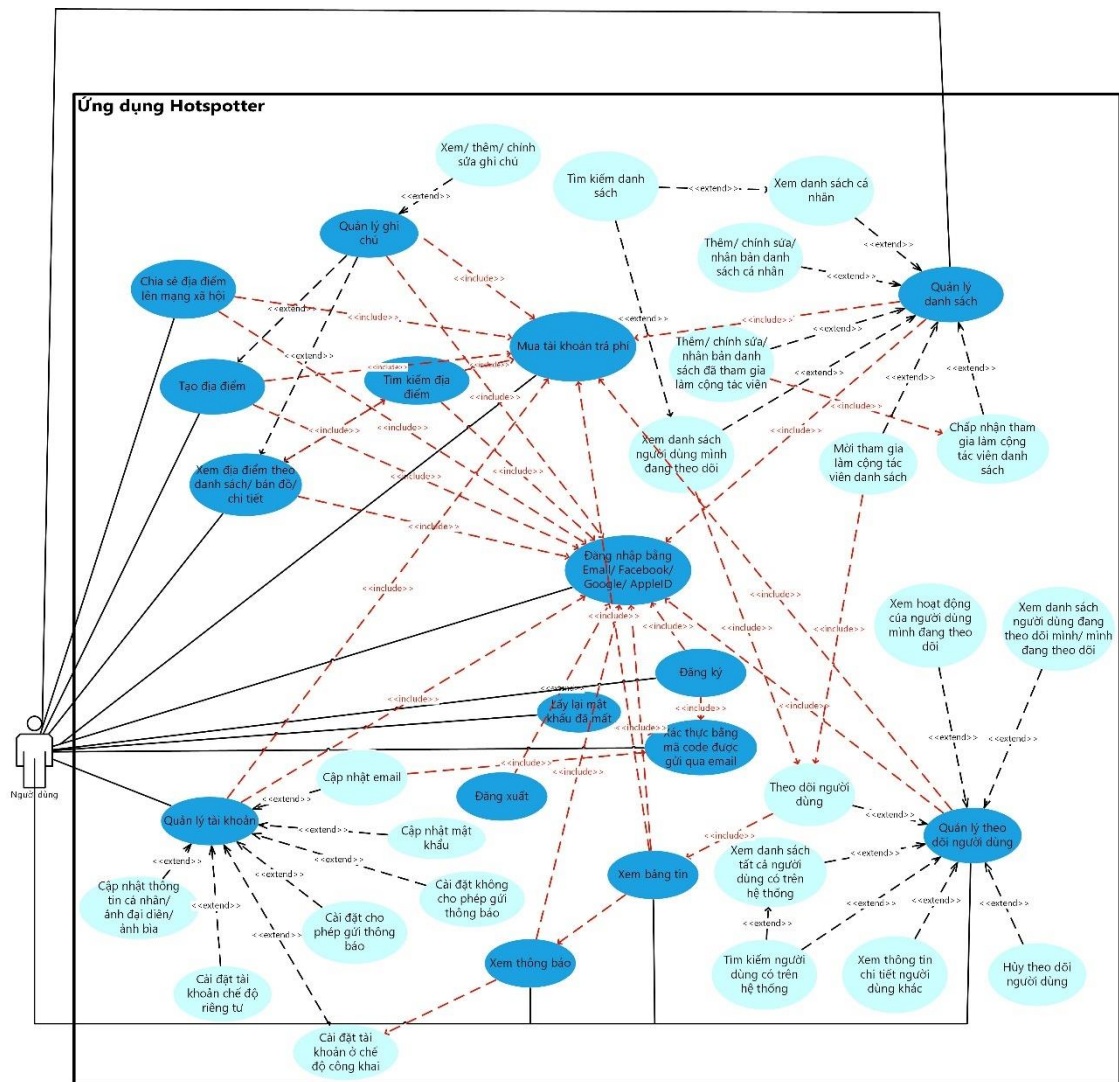
3.2.2. Workflow website quản trị hotspotter



Hình 3.2 Hình workflow website quản trị Hotspotter

3.3. Sơ đồ Use Case

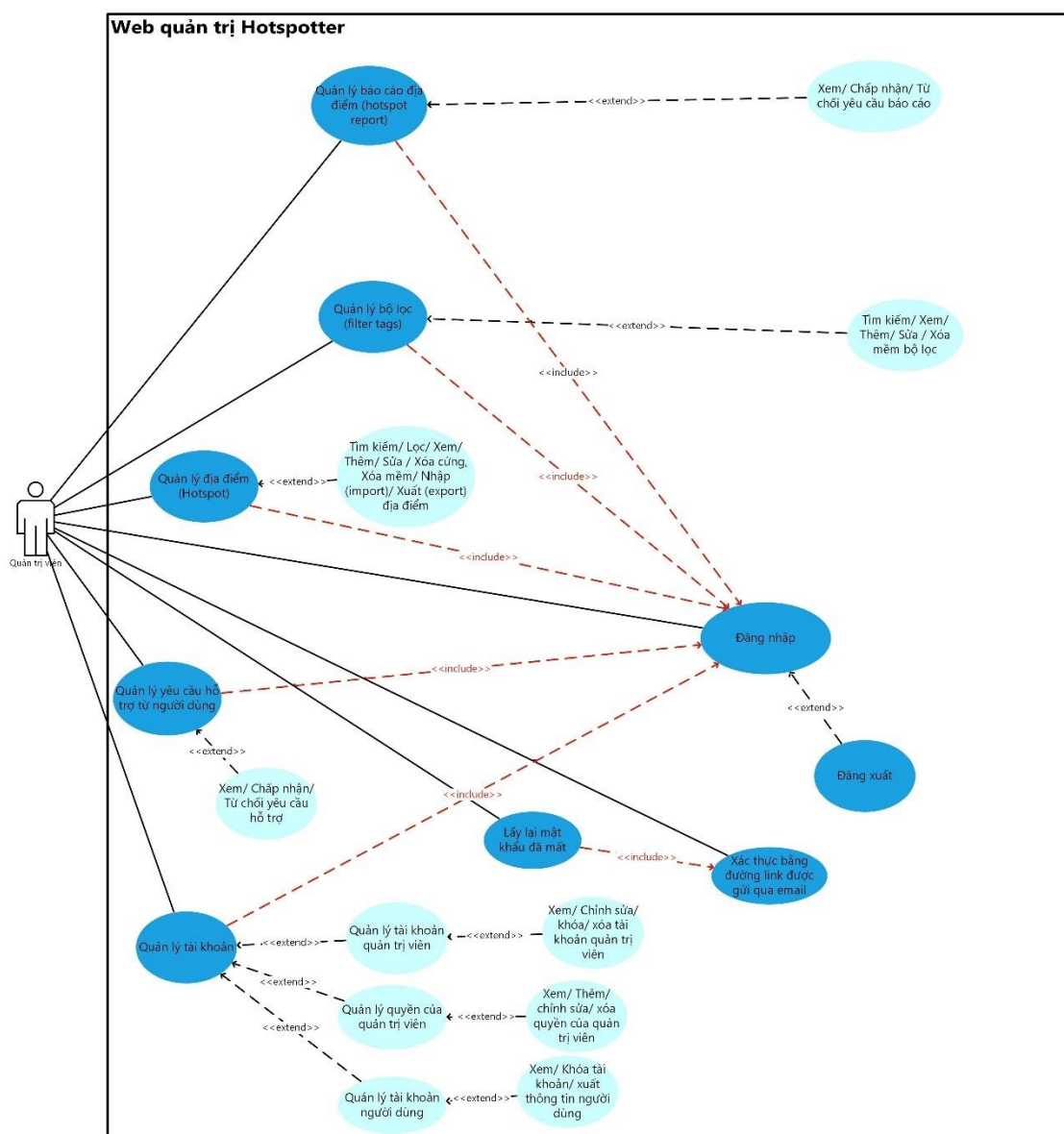
3.3.1. Sơ đồ Use Case tổng quát ứng dụng Hotspotter



Hình 3.3 Hình sơ đồ usecase ứng dụng Hotspotter

- Những usecase không thực hiện trong đề tài: Mua tài khoản trả phí, Gửi thông báo đẩy đến người dùng (push notification)

3.3.2. Sơ đồ usecase website quản trị



Hình 3.4 Hình sơ đồ usecase website quản trị Hotspotter

- Những use case không thực hiện trong đề tài: Import dữ liệu địa điểm (hotspot)

3.4. Mô tả chi tiết các Usecase ứng dụng và website quản trị Hotspotter

3.4.1. Mô tả chi tiết các Usecase ứng dụng Hotspotter

- Đăng kí bằng email

Tên UC	Đăng kí bằng email	UC01_1
--------	--------------------	--------

Điều kiện tiên quyết		
Hậu điều kiện	Hệ thống thông báo thành công và điều hướng đến màn hình HomeScreen	
Luồng xử lý chính	Người dùng	Hệ thống phản hồi
	Truy cập đến màn hình Đăng ký	Hiển thị màn hình Đăng ký
	Nhập các thông tin như email, firstname, lastname, username, password	<ul style="list-style-type: none"> - Kiểm tra thông tin hợp lệ - Kiểm tra Email và Username đã tồn tại trong hệ thống - Kiểm tra password hợp lệ
	Bấm nút đăng kí	<ul style="list-style-type: none"> - Tạo tài khoản cho USER và tự động đăng nhập - Điều hướng đến màn hình HomeScreen

Bảng 3.3 Mô tả chi tiết chức năng đăng kí bằng email ()

Mở rộng:

1. Khi đăng ký: Tại bước đăng nhập bằng Email → Hệ thống kiểm tra xem Email đã được đăng ký chưa:

- Nếu chưa: Hiển thị popup xác nhận tạo mới tài khoản → điều hướng đến màn hình đăng ký
- Nếu đã đăng ký: đăng nhập thành công điều hướng đến màn hình Homescreen

2. Một email, username chỉ tồn tại duy nhất 1 tài khoản

- Đăng nhập bằng email

Tên UC	Đăng nhập bằng email	UC02_1
Điều kiện tiên quyết		
Hậu điều kiện	Hệ thống thông báo thành công và điều hướng đến màn hình HomeScreen	
Luồng xử lý chính	Người dùng	Hệ thống phản hồi
	Truy cập đến trang Đăng nhập bằng Email	Hiển thị giao diện nhập Email, trường Email
	Nhập Email và bấm nút Next	<ul style="list-style-type: none"> - Kiểm tra Email đã tồn tại trong hệ thống - Hiển thị giao diện nhập mật khẩu
	<ul style="list-style-type: none"> - Nhập mật khẩu - Bấm nút Log in 	<ul style="list-style-type: none"> - Điều hướng đến màn hình HomeScreen

Bảng 3.4 Mô tả chi tiết chức năng đăng nhập bằng email (ứng dụng Hotspotter)

Mở rộng:

1. Nếu email chưa được đăng ký điều hướng đến luồng đăng kí
2. Nếu là lần đầu đăng nhập hiển thị màn hình Allow Location và Allow Notification

- Đăng ký/ đăng nhập bằng Google/ Facebook/ AppleId

Tên UC	Đăng kí/ Đăng nhập bằng Google/ Facebook/ AppleId	UC02_2
--------	---	--------

Điều kiện tiên quyết		
Hậu điều kiện	Hệ thống thông báo thành công và điều hướng đến màn hình HomeScreen	
Luồng xử lý chính	Người dùng	Hệ thống phản hồi
	<ul style="list-style-type: none"> - Truy cập đến trang Đăng ký/ Đăng nhập - Bấm vào icon Facebook/ Google tương ứng 	<ul style="list-style-type: none"> - Xác thực quyền truy cập vào tài khoản Google/ Facebook/ AppleId - Tạo tài khoản mới (Nếu USER chưa có tài khoản trong ứng dụng Hotsptter) - Đăng nhập và điều hướng về màn hình trước đó của USER

Bảng 3.5 Mô tả chi tiết chức năng Đăng ký/ Đăng nhập bằng Google/ Facebook/ AppleID (ứng dụng Hotspotter)

Mở rộng:

1. Nếu tài khoản Google/ Facebook/ AppleId đó đã được đăng ký trong hệ thống → Không hiện màn hình yêu cầu USER xác thực quyền truy cập
2. Khi đăng ký/ đăng nhập bằng Google → Hệ thống sẽ lưu Email, Tên, Ảnh đại diện, Google ID
3. Nếu email đăng nhập của tài khoản Google/ Facebook/ AppleId đã được lưu trong hệ thống HotSpotter trước đó → Hệ thống liên kết 2 tài khoản thành 1 → Giữ nguyên thông tin của tài khoản, không cập nhật theo tài khoản Google/ Facebook/ AppleId

- **Xem thông tin tài khoản**

Tên UC	Xem thông tin tài khoản	UC03_1
Điều kiện tiên quyết		
Hậu điều kiện	Hệ thống hiển thị thông tin tài khoản người dùng	
Luồng xử lý chính	Người dùng	Hệ thống phản hồi
	- Đi đến màn hình Profile	- Hiển thị Thông tin tài khoản của USER gồm: Cover, avatar, first name, last name, user name

Bảng 3.6 Mô tả chi tiết chức năng xem thông tin tài khoản (ứng dụng Hotspotter)

Mở rộng:

1. Trong trường hợp đăng ký email, nếu người dùng chưa có ảnh đại diện → Giao diện hiển thị ảnh đại diện mặc định của hệ thống

- **Đăng ảnh đại diện/ ảnh bìa**

Tên UC	Đăng ảnh đại diện/ ảnh bìa	UC03_2
Điều kiện tiên quyết		
Hậu điều kiện	Hệ thống lưu và hiển thị ảnh đại diện/ ảnh bìa	
Luồng xử lý chính	Người dùng	Hệ thống phản hồi

	<ul style="list-style-type: none"> - Đi đến màn hình Profile - Chọn thay đổi avatar/cover 	<ul style="list-style-type: none"> - Hiện thị popup gồm 2 lựa chọn: chụp ảnh/ chọn ảnh từ thư viện
	<ul style="list-style-type: none"> - Xác nhận ảnh muốn đổi 	<ul style="list-style-type: none"> - Cập nhật ảnh đại diện cho tài khoản tại tất cả những mục có ảnh đại diện

Bảng 3.7 Mô tả chi tiết chức năng đăng ảnh đại diện/ ảnh bìa (ứng dụng Hotspotter)

Mở rộng:

1. Nếu chưa có avatar/ cover thì hiển thị ảnh mặc định
2. Người dùng không được phép xóa avatar sau khi cập nhật
3. Ảnh upload không quá 10MB

- Thay đổi email cá nhân

Tên UC	Thay đổi email cá nhân	UC03_3
Điều kiện tiên quyết		
Hậu điều kiện	Hệ thống lưu email mới được cập nhật	
Luồng xử lý chính	Người dùng	Hệ thống phản hồi
	<ul style="list-style-type: none"> - Đi đến trang Thông tin tài khoản - Bấm nút Cập nhật tại trường Email 	<ul style="list-style-type: none"> - Hiện thị màn hình Cập nhật Email

	<ul style="list-style-type: none"> - Nhập email muốn cập nhật - Bấm nút Save 	<ul style="list-style-type: none"> - Gửi mã xác thực đến email người dùng vừa nhập - Hiển thị màn hình nhập mã xác thực
	<ul style="list-style-type: none"> - Nhập mã xác thực - Bấm nút Lưu thay đổi 	<ul style="list-style-type: none"> - Cập nhật email cho tài khoản - Quay lại trang Thông tin tài khoản

Bảng 3.8 Mô tả chi tiết chức năng thay đổi email cá nhân (ứng dụng Hotspotter)

- **Xem địa điểm theo dạng danh sách/ bản đồ (map view/ list view)**

Tên UC	Xem địa điểm theo danh sách/ bản đồ	UC04_1
Điều kiện tiên quyết		
Hậu điều kiện	Người dùng xem được các địa điểm	
Luồng xử lý chính	Người dùng	Hệ thống phản hồi
	<ul style="list-style-type: none"> - Truy cập đến màn hình danh sách/ bản đồ (Map View/ List view) 	<ul style="list-style-type: none"> - Hiển thị danh sách các icon địa điểm trên bản đồ/ các địa điểm theo dạng danh sách

Bảng 3.9 Mô tả chi tiết chức năng xem địa điểm theo dạng danh sách/ bản đồ (ứng dụng Hotspotter)

Mở rộng:

1. Trường hợp hiển thị theo dạng bản đồ (map view) mặc định hiển thị icon địa điểm trong bán kính 50km tính từ điểm trung tâm của bản đồ

- Tìm kiếm địa điểm theo từ khóa

Tên UC	- Tìm kiếm địa điểm theo từ khóa	UC04_2
Điều kiện tiên quyết		
Hậu điều kiện	Hệ thống trả về kết quả theo nội dung tìm kiếm	
Luồng xử lý chính	Người dùng	Hệ thống phản hồi
		- Hiển thị màn hình Tìm kiếm và các lịch sử tìm kiếm theo tên Hotspot
	Chọn tìm theo tên địa điểm/ theo khu vực và bấm vào kết quả được đề xuất	<ul style="list-style-type: none"> - TH1: Tìm kiếm trên bản đồ <ul style="list-style-type: none"> o Di chuyển map đến vị trí của địa điểm o Hiển thị các địa điểm trong bán kính 15km kể từ spot được chọn - TH2: Tìm kiếm theo dạng danh sách <ul style="list-style-type: none"> o Hiển thị danh sách các địa điểm tại màn hình danh sách

Bảng 3.10 Mô tả chi tiết chức năng tìm kiếm địa điểm theo từ khóa (ứng dụng Hotspotter)

Mở rộng:

1. Nếu tìm kiếm không có kết quả hiển thị thông báo We couldn't find the results you are looking for. Please try again

- Lọc các địa điểm theo các tiêu chí

Tên UC	- Lọc các địa điểm theo các tiêu chí	UC04_3
Điều kiện tiên quyết		
Hậu điều kiện	Hệ thống trả về kết quả theo các tiêu chí lọc	
Luồng xử lý chính	Người dùng	Hệ thống phản hồi
		<ul style="list-style-type: none"> - Hiển thị màn hình các tiêu chí lọc gồm các thông tin: <ul style="list-style-type: none"> o Your list (TBD) o Categories o Price o Country cuisine o Food style o Location
	Chọn tìm theo tên địa điểm/ theo khu vực và bấm vào kết quả được đề xuất	<ul style="list-style-type: none"> - Lọc và hiển thị địa điểm mới dựa theo các tiêu chí lọc đã chọn - Điều hướng về màn hình hiển thị kết quả - Hiển thị dưới dạng bản đồ

Bảng 3.11 Mô tả chi tiết chức năng lọc địa điểm theo các tiêu chí lọc (ứng dụng Hotspotter)

Mở rộng:

1. Có thể chọn nhiều item trong 1 tiêu chí lọc và chọn nhiều tiêu chí lọc cùng một lúc

- Xem danh sách cá nhân (view my list)

Tên UC	- Xem danh sách cá nhân (view my list)	UC05_1
Điều kiện tiên quyết		
Hậu điều kiện	Hệ thống trả về danh sách mặc định (default list) hoặc danh sách người dùng tạo (custom list)	
Luồng xử lý chính	Người dùng	Hệ thống phản hồi
	- Đi đến màn hình Save	- Hiện thị giao diện màn hình gồm 2 tab: - Danh sách của tôi (my lists) - Danh sách đang theo dõi (following lists)
	- Chọn tab My lists	- Hiện thị Lists của người dùng gồm: 3 default list (Favorites; Been; Want to go), các list được tạo bởi user

Bảng 3.12 Mô tả chi tiết chức năng xem danh sách cá nhân (ứng dụng Hotspotter)

Mở rộng:

1. Người dùng có sẵn 3 lists cố định khi có tài khoản trong hệ thống
2. Người dùng không được phép xóa 3 list cố định
3. Danh sách cố định hiển thị trạng thái private

- **Xem danh sách đang theo dõi (view list following)**

Tên UC	- Xem danh sách đang theo dõi (view list following)	UC05_2
Điều kiện tiên quyết		
Hậu điều kiện	Hệ thống trả về danh sách đang theo dõi	
Luồng xử lý chính	Người dùng	Hệ thống phản hồi
	- Đi đến màn hình Save	- Hiển thị giao diện màn hình gồm 2 tab: - Danh sách của tôi (my lists) - Danh sách đang theo dõi (following lists)
	- Chọn tab following	- Hiển thị lists đã theo dõi

Bảng 3.13 Mô tả chi tiết chức năng xem danh sách đang theo dõi (ứng dụng Hotspotter)

Mở rộng:

1. Sắp theo thời gian lưu cũ nhất → mới nhất
2. Bao gồm những danh sách (List) được mời làm cộng tác viên (collaborative)
3. List được mời vào collaborative hiển thị icon đóng góp ở avatar

- **Xem chi tiết danh sách (view list detail)**

Tên UC	- Xem chi tiết danh sách (view list detail)	UC05_3
Điều kiện tiên quyết		
Hậu điều kiện	Hệ thống trả về chi tiết danh sách	
Luồng xử lý chính	Người dùng	Hệ thống phản hồi
	- Chọn vào List tại My list/ Following để xem chi tiết	- Hiện thị giao diện màn hình xem chi tiết gồm các thông tin: Thông tin danh sách (list), các địa điểm (hotspot) được lưu trong danh sách (list)
	- Chọn tab following	- Hiện thị lists đã theo dõi

Bảng 3.14 Mô tả chức năng xem chi tiết danh sách cá nhân (ứng dụng Hotspotter)

Mở rộng:

1. Sắp theo thời gian lưu cũ nhất → mới nhất
2. Bao gồm những danh sách (List) được mời làm cộng tác viên (collaborative)
3. List được mời vào collaborative hiển thị icon đóng góp ở avatar

- **Tạo mới danh sách (create new list)**

Tên UC	- Tạo mới danh sách (create new list)	UC05_4
Điều kiện tiên quyết		
Hậu điều kiện	Lưu danh sách mới của người dùng vào hệ thống	
Luồng xử lý chính	Người dùng	Hệ thống phản hồi
	- Chọn nút Create new list	- Hiện thị giao diện Create new list gồm các trường: name, description, isCollaborative
	- Nhập các vào các trường và chọn nút Create	- Điều hướng đến màn hình List details khi tạo mới thành công

Bảng 3.15 Mô tả chi tiết chức năng tạo mới danh sách (ứng dụng Hotspotter)

- **Chỉnh sửa danh sách (edit list)**

Tên UC	- Chỉnh sửa danh sách (edit list)	UC05_5
Điều kiện tiên quyết		
Hậu điều kiện	Lưu danh sách đã được chỉnh sửa vào hệ thống	
Luồng xử lý chính	Người dùng	Hệ thống phản hồi
	- Chọn icon xem thêm tại List details	- Hiện thị popup các Custom lists gồm: Edit list, duplicate list,

		select hotspots, delete list
	<ul style="list-style-type: none"> - Chọn edit list - Điền các thông tin cần chỉnh sửa - Bấm lưu 	<ul style="list-style-type: none"> - Lưu danh sách đã chỉnh sửa vào hệ thống

Bảng 3.16 Mô tả chi tiết chức năng chỉnh sửa danh sách (ứng dụng Hotspotter)

- **Nhân bản danh sách (duplicate list)**

Tên UC	- Nhân bản danh sách (duplicate list)	UC05_6
Điều kiện tiên quyết		
Hậu điều kiện	Lưu danh sách đã được chỉnh sửa vào hệ thống	
Luồng xử lý chính	Người dùng	Hệ thống phản hồi
	<ul style="list-style-type: none"> - Chọn icon xem thêm tại List details 	<ul style="list-style-type: none"> - Hiện thị popup các Custom lists gồm: Edit list, duplicate list, select hotspots, delete list
	<ul style="list-style-type: none"> - Chọn edit list - Điền các thông tin cần chỉnh sửa - Bấm lưu 	<ul style="list-style-type: none"> - Lưu danh sách đã chỉnh sửa vào hệ thống

Bảng 3.17 Mô tả chi tiết chức năng nhân bản danh sách (ứng dụng Hotspotter)

- **Lưu địa điểm vào danh sách (save hotspot to list)**

Tên UC	- Lưu địa điểm vào danh sách (save hotspot to list)	UC05_7
Điều kiện tiên quyết		
Hậu điều kiện	Lưu địa điểm vào danh sách được chọn	
Luồng xử lý chính	Người dùng	Hệ thống phản hồi
	- Tại màn hình Hotspot detail, chọn nút Save	- Hiện thị popup save to my lists, gồm các thông tin: title, description, note, danh sách my list dưới dạng checkbox, nút done
	- Nhập kí tự vào trường note	- Xác thực dữ liệu
	- Chọn checkbox muốn lưu - Bấm nút done	- Hệ thống lưu hotspot vào các lists được chọn - Quay về lại màn hình trước đó

Bảng 3.18 Mô tả chi tiết chức năng lưu địa điểm vào danh sách cá nhân/ danh sách tham gia làm cộng tác viên (ứng dụng Hotspotter)

Mở rộng

1. Danh sách list hiển thị bao gồm:

- List được tạo bởi người dùng hoặc list người dùng được mời làm cộng tác viên

3.4.2. Mô tả chi tiết các Usecase website quản trị Hotspotter

- **Đăng nhập**

Tên UC	Đăng nhập	UC01_1
Điều kiện tiên quyết		
Hậu điều kiện	Hệ thống thông báo đăng nhập thành công và điều hướng đến màn hình trang chủ	
Luồng xử lý chính	Quản trị viên	Hệ thống phản hồi
	Truy cập đến trang Đăng nhập	Hiển thị giao diện nhập gồm các trường Email, Password
	<ul style="list-style-type: none"> - Nhập email, password - Bấm nút Login 	<ul style="list-style-type: none"> - Điều hướng đến màn hình trang chủ

Bảng 3.19 Mô tả chi tiết chức năng đăng nhập (website quản trị Hotspotter)

- Xem danh sách người dùng

Tên UC	Xem danh sách người dùng	UC01_1
Điều kiện tiên quyết		
Hậu điều kiện	Hệ thống hiển thị danh sách người dùng đăng kí sử dụng ứng dụng	
Luồng xử lý chính	Quản trị viên	Hệ thống phản hồi
	Chọn Account Manage -> chọn User	Hiển thị danh sách người dùng

Bảng 3.20 Mô tả chi tiết chức năng xem danh sách người dùng (website quản trị Hotspotter)

- **Xem chi tiết thông tin người dùng**

Tên UC	Xem chi tiết thông tin người dùng	UC01_1
Điều kiện tiên quyết		
Hậu điều kiện	Hệ thống hiển thị thông tin người dùng	
Luồng xử lý chính	Quản trị viên	Hệ thống phản hồi
	Đến danh sách người dùng -> Chọn vào người dùng để xem chi tiết	Hiển thị thông tin người dùng

Bảng 3.21 Mô tả chi tiết chức năng xem chi tiết thông tin người dùng (website quản trị Hotspotter)

- **Vô hiệu hóa/ kích hoạt tài khoản người dùng**

Tên UC	Vô hiệu hóa/ kích hoạt tài khoản người dùng	UC02_1
Điều kiện tiên quyết		
Hậu điều kiện	Hệ thống vô hiệu hóa hoặc kích hoạt tài khoản người dùng trở lại	
Luồng xử lý chính	Quản trị viên	Hệ thống phản hồi
	Ở màn hình xem chi tiết người dùng -> Bấm vào nút Block/ Active	Vô hiệu hóa/ kích hoạt tài khoản người dùng

Bảng 3.22 Mô tả chi tiết chức năng vô hiệu hóa/ kích hoạt tài khoản người dùng (website quản trị Hotspotter)

- Tạo tài khoản quản trị viên

Tên UC	Tạo tài khoản quản trị viên	UC02_2
Điều kiện tiên quyết		
Hậu điều kiện	Hệ thống tạo mới tài khoản quản trị viên	
Luồng xử lý chính	Quản trị viên	Hệ thống phản hồi
	Chọn Account Manage -> Chọn Administrators -> Chọn button Add admin	Hiển thị form gồm các trường fullname, email, password, confirm password, role
	- Điền các thông tin vào các ô input được hiển thị - Bấm nút save	Lưu thông tin tài khoản admin vừa tạo vào hệ thống

Bảng 3.23 Mô tả chi tiết chức năng tạo tài khoản quản trị viên (website quản trị Hotspotter)

- Tạo mới địa điểm (Hotspot)

Tên UC	Tạo mới địa điểm (Hotspot)	UC03_1
Điều kiện tiên quyết		
Hậu điều kiện	Hệ thống tạo địa điểm	
Luồng xử lý chính	Quản trị viên	Hệ thống phản hồi

	Chọn Hotspots Manage -> Chọn Hotspots -> Chọn button Add hotspot	Hiển thị form các trường thông tin hotspot cần nhập
	<ul style="list-style-type: none"> - Điền các thông tin vào các ô input được hiển thị - Bấm nút save 	Lưu thông tin hotspot vừa tạo vào hệ thống

Bảng 3.24 Mô tả chi tiết chức năng tạo địa điểm (website quản trị Hotspotter)

- Xem danh sách địa điểm (Hotspot)

Tên UC	Xem danh sách địa điểm (Hotspot)	UC03_2
Điều kiện tiên quyết		
Hậu điều kiện	Hệ thống hiển thị danh sách địa điểm	
Luồng xử lý chính	Quản trị viên	Hệ thống phản hồi
	Chọn Hotspots Manage -> chọn Hotspots	Hiển thị danh sách địa điểm được phân trang -> 20 địa điểm trên 1 trang
	<ul style="list-style-type: none"> - Điền các thông tin vào các ô input được hiển thị - Bấm nút save 	Lưu thông tin hotspot vừa tạo vào hệ thống

Bảng 3.25 Mô tả chi tiết chức năng xem danh sách địa điểm (website quản trị Hotspotter)

- Xem chi tiết địa điểm (Hotspot)

Tên UC	Xem chi tiết địa điểm (Hotspot)	UC03_3
Điều kiện tiên quyết		
Hậu điều kiện	Hệ thống hiển thị thông tin chi tiết địa điểm	
Luồng xử lý chính	Quản trị viên	Hệ thống phản hồi
	Chọn Hotspots Manage -> chọn Hotspots -> Chọn địa điểm muốn xem thông tin	Hiển thị thông tin chi tiết địa điểm

Bảng 3.26 Mô tả chi tiết chức năng xem chi tiết địa điểm (website quản trị Hotspotter)

- Chỉnh sửa địa điểm (Hotspot)

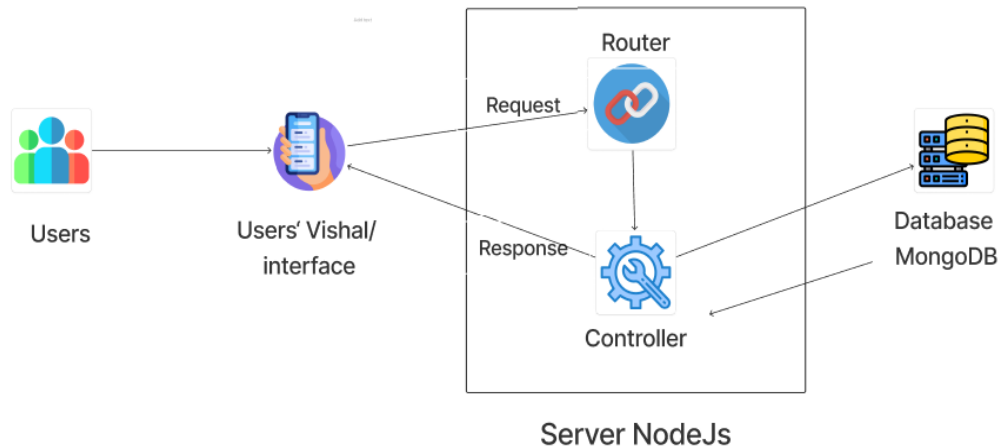
Tên UC	Chỉnh sửa địa điểm (Hotspot)	UC03_4
Điều kiện tiên quyết		
Hậu điều kiện	Hệ thống lưu thông tin địa điểm đã được chỉnh sửa	
Luồng xử lý chính	Quản trị viên	Hệ thống phản hồi
	<ul style="list-style-type: none"> - Ở trang xem chi tiết địa điểm nhập những thông tin cần thay đổi - Bấm nút save 	Lưu thông tin địa điểm đã được thay đổi vào hệ thống

Bảng 3.27 Mô tả chi tiết chức năng chỉnh sửa địa điểm (website quản trị Hotspotter)

CHƯƠNG 4. THIẾT KẾ HỆ THỐNG

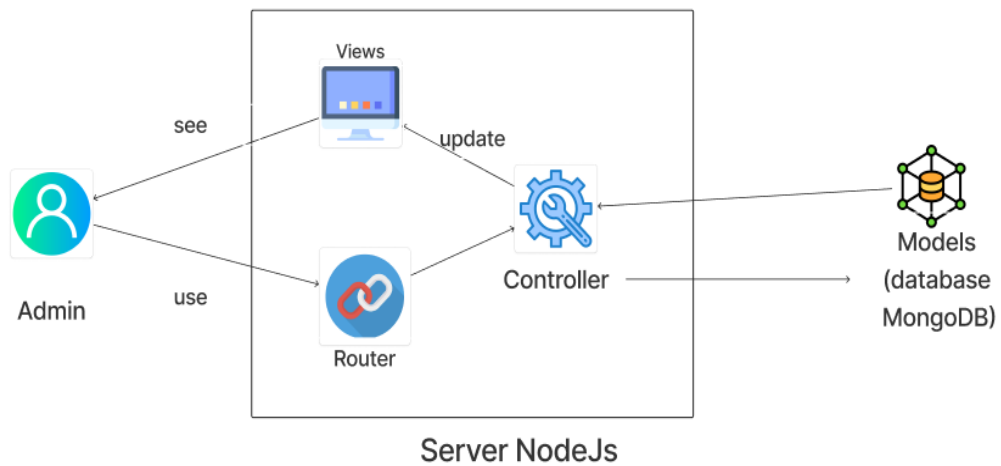
4.1. Thiết kế kiến trúc hệ thống

4.1.1. Thiết kế kiến trúc ứng dụng Hotspotter



Hình 4.1 Kiến trúc hệ thống ứng dụng Hotspotter

4.1.2. Thiết kế kiến trúc hệ thống website quản trị Hotspotter



Hình 4.2 Kiến trúc hệ thống website quản trị Hotspotter

4.2. Thiết kế cơ sở dữ liệu

- Nhóm phát triển chọn cơ sở dữ liệu MongoDB để triển khai cho dự án và lưu trữ dữ liệu được thực hiện trên server
- Vì lý do bảo mật từ phía công ty nên không thể đưa thông tin chi tiết về cơ sở dữ liệu vào trong báo cáo tốt nghiệp

4.3. Thiết kế lớp

- Vì lý do bảo mật từ phía công ty nên không thể đưa thông tin chi tiết về cơ sở dữ liệu và những thông tin liên quan đến cơ sở dữ liệu vào trong báo cáo tốt nghiệp

4.4. Thiết kế giao diện người dùng

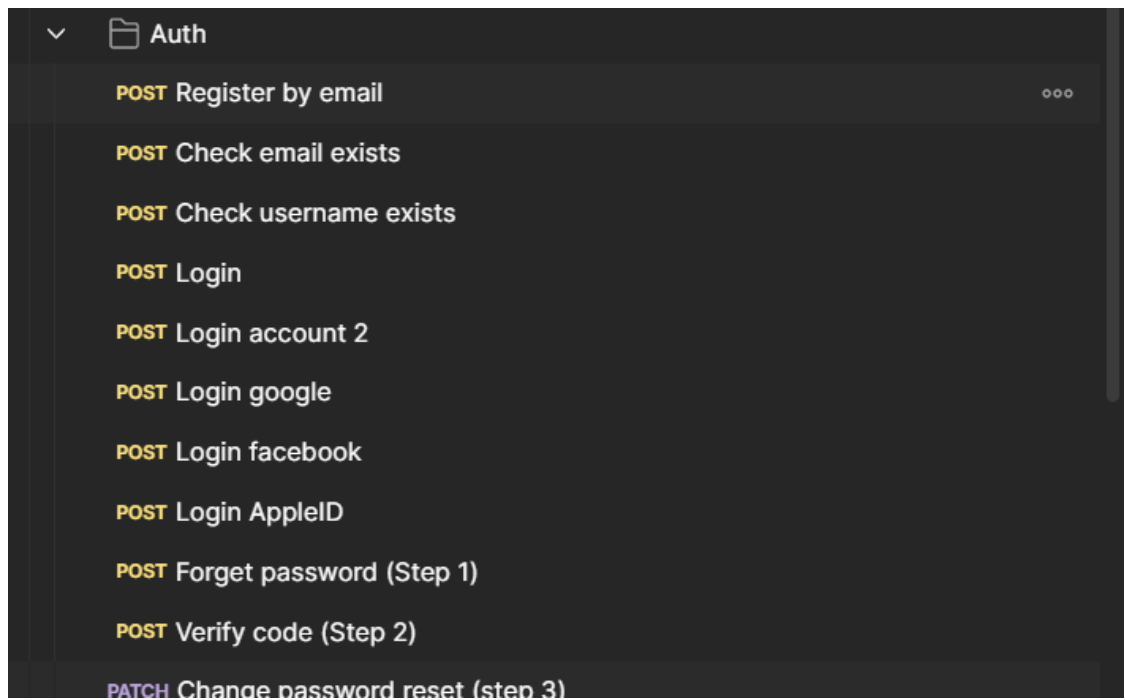
4.4.1. *Giao diện website quản trị Hotspotter*

Để đáp ứng với nhu cầu phải hoàn thành nhanh và tiết kiệm chi phí nhưng vẫn đáp ứng được những chức năng cần thiết cho website quản trị Hotspotter công ty đã đề xuất khách hàng sử dụng một admin template được phát triển dựa trên Bootstrap 3 có tên gọi là AdminLTE 3. Đây là đường dẫn đến template <https://adminlte.io/themes/v3/>. Nhóm phát triển đã tiến hành thiết kế giao diện website quản trị Hotspotter dựa trên giao diện AdminLTE 3, đồng thời tiếp thu và hoàn thiện dựa trên ý kiến phản hồi của khách hàng để mang đến trải nghiệm quản trị tối ưu nhất

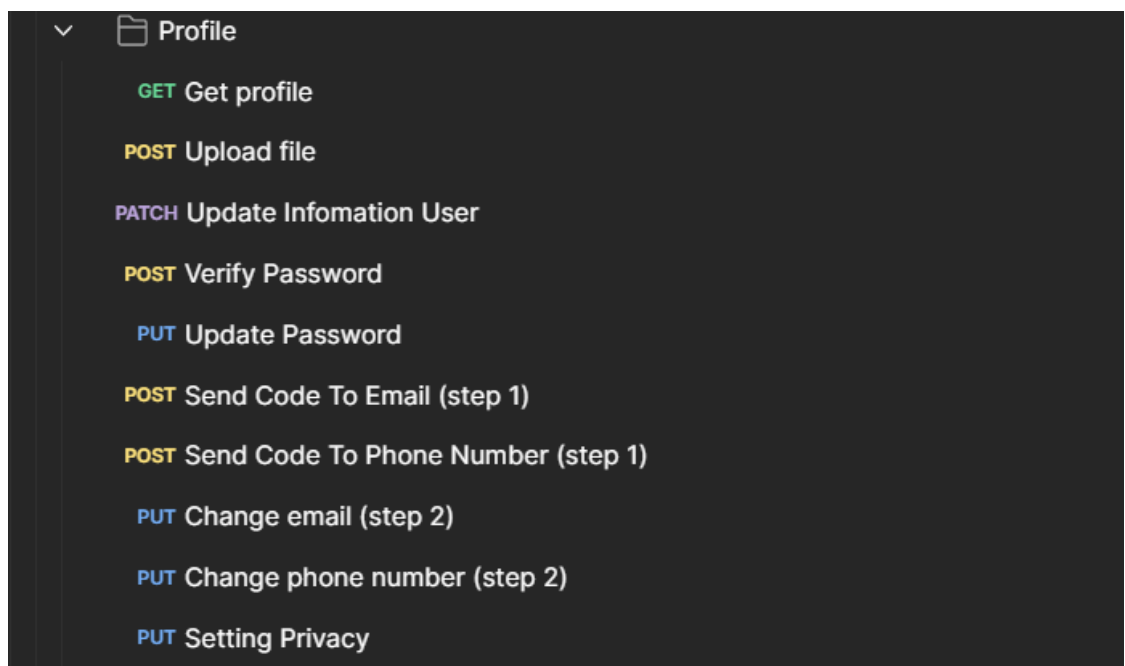
4.5. Thiết kế API

4.5.1. *Thiết kế API ứng dụng Hotspotter*

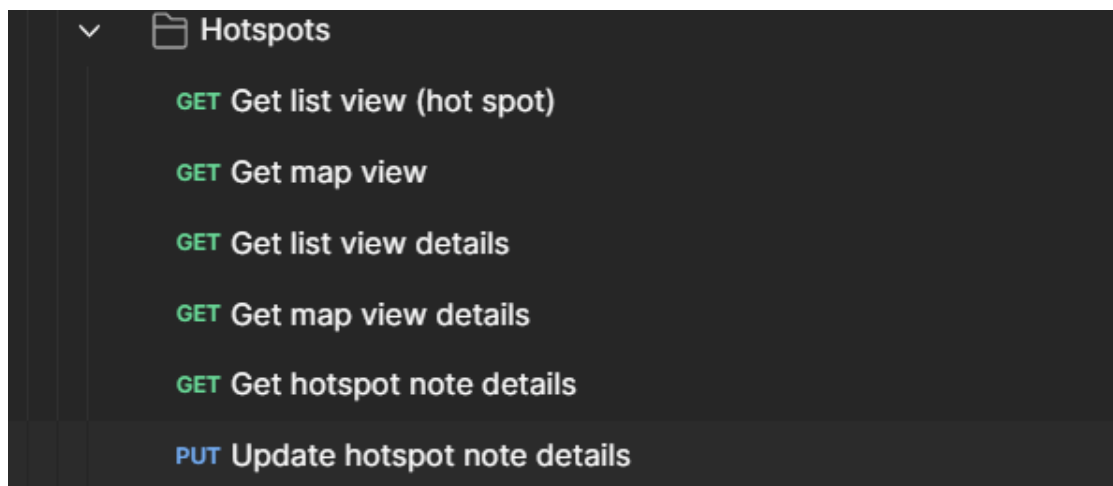
Dưới đây là hình ảnh danh sách các API được xây dựng dành riêng cho từng module của ứng dụng, giúp người dùng thực hiện các chức năng cụ thể:



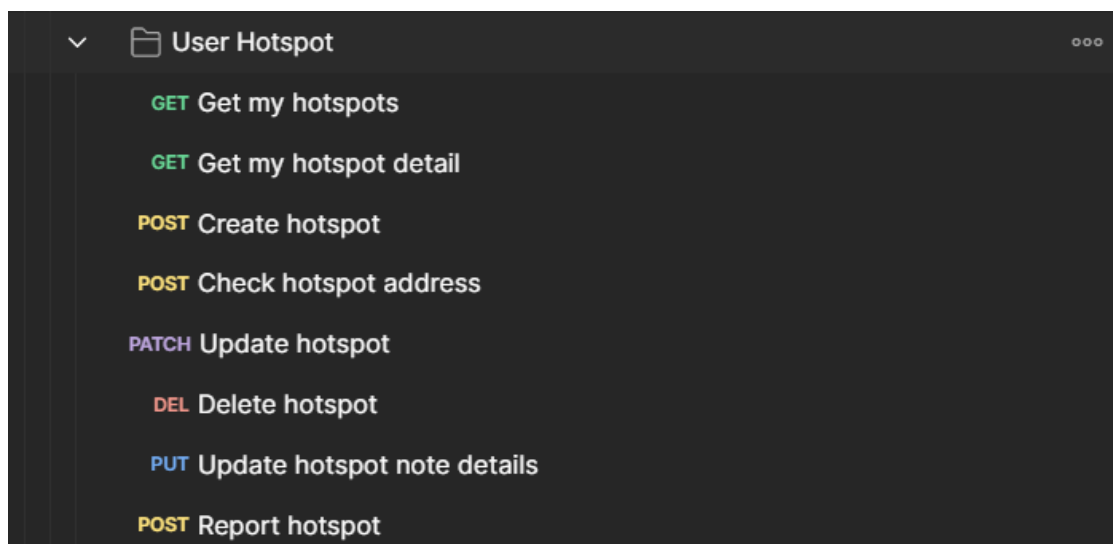
Hình 4.3 Module quản lý chứng thực (Authentication)



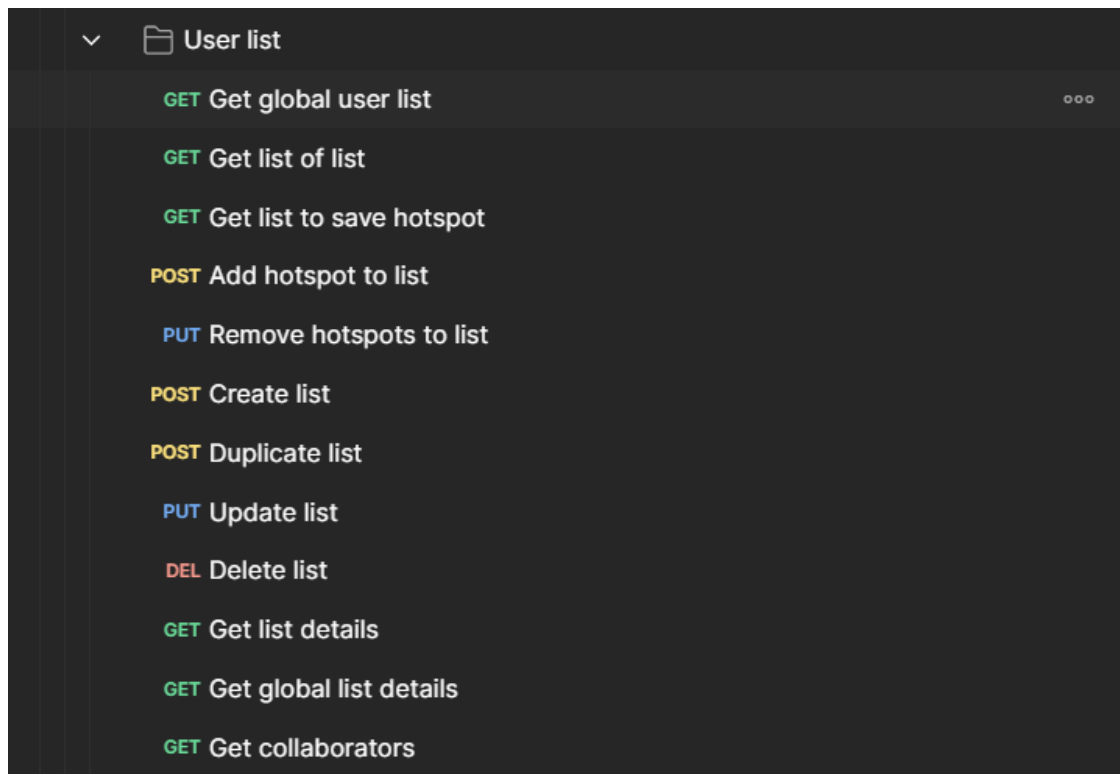
Hình 4.4 Module thông tin cá nhân (Profile)



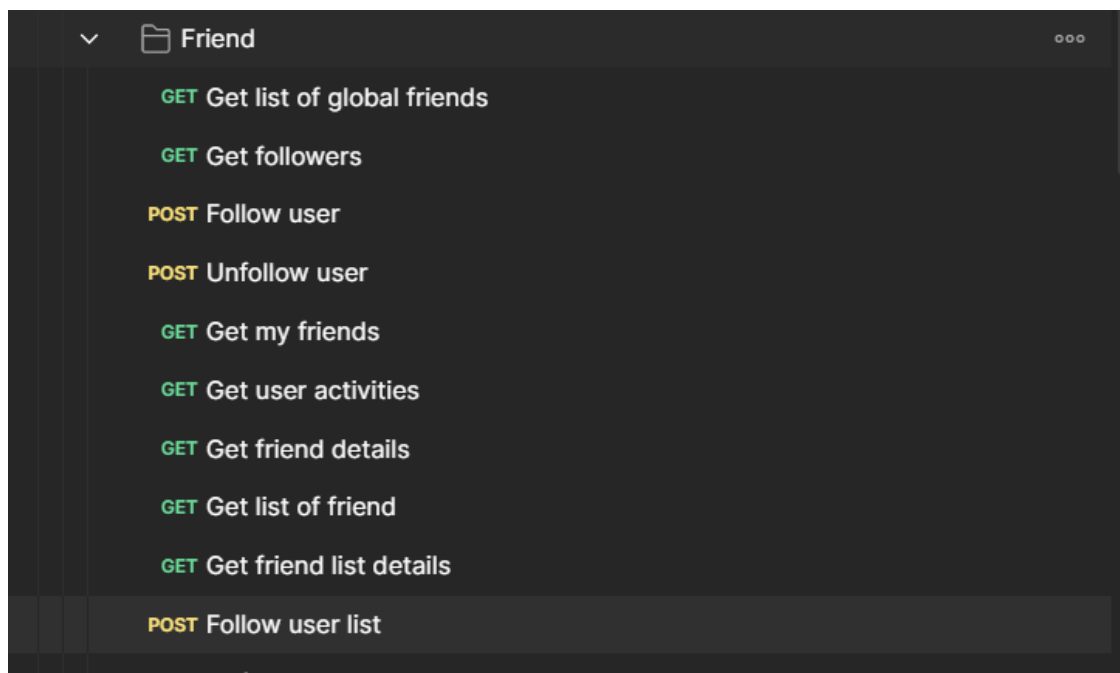
Hình 4.5 Module địa điểm (Hotspots)



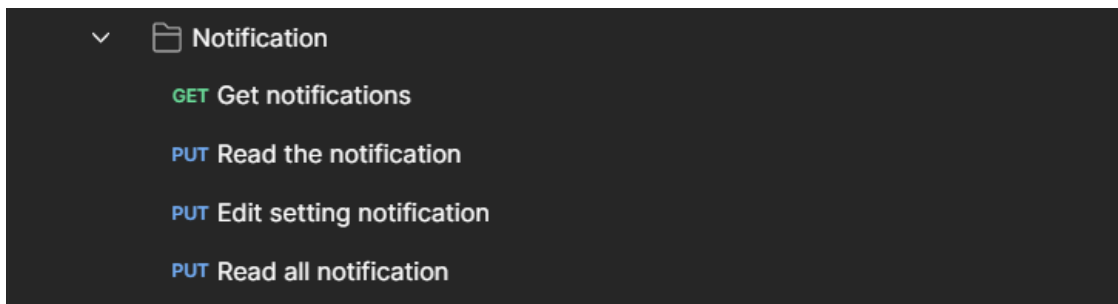
Hình 4.6 Module địa điểm cá nhân (User Hotspot)



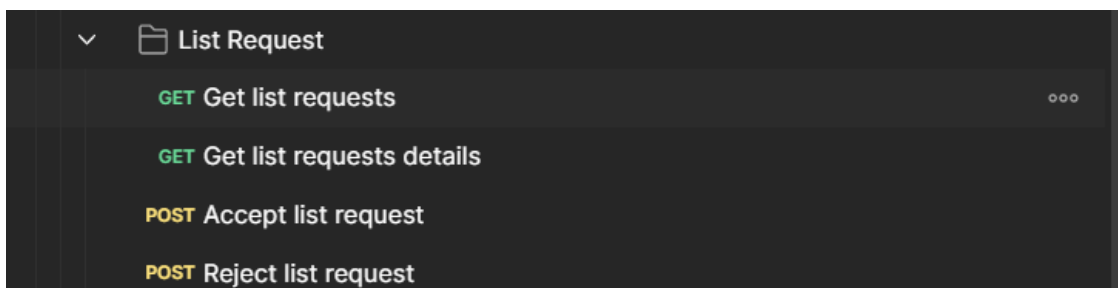
Hình 4.7 Module danh sách người dùng (User list)



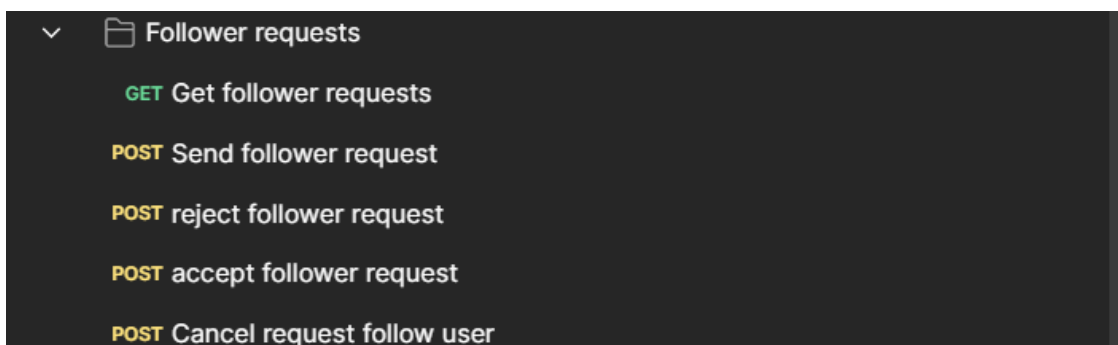
Hình 4.8 Module bạn bè (Friend)



Hình 4.9 Module thông báo



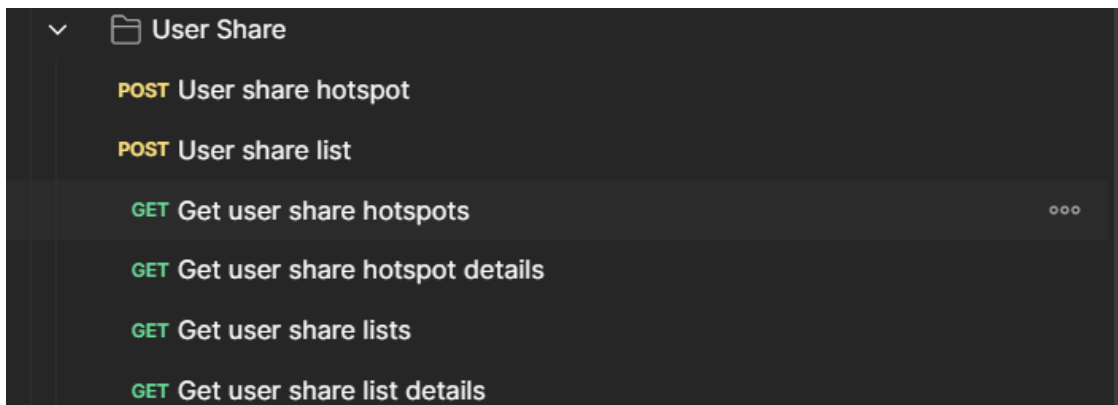
Hình 4.10 Module yêu cầu tham gia làm cộng tác viên danh sách (List Request)



Hình 4.11 Module yêu cầu theo dõi (Follower request)



Hình 4.12 Module gửi yêu cầu hỗ trợ đến quản trị viên (Help Request)



Hình 4.13 Module người dùng chia sẻ địa điểm hoặc danh sách (User Share)



Hình 4.14 Module bảng tin (Newsfeed)

CHƯƠNG 5. XÂY DỰNG HỆ THỐNG

5.1. Công cụ và môi trường phát triển

- Trello: Trello là công cụ quản lý công việc linh hoạt, nơi các nhóm có thể lập kế hoạch, cộng tác trên các dự án, tổ chức quy trình làm việc và theo dõi tiến độ một cách trực quan, hiệu quả và bổ ích. Từ lên ý tưởng, lập kế hoạch cho đến thực hiện, Trello quản lý các dấu mốc quan trọng và những nhiệm vụ hàng ngày khi làm việc cùng nhau và giúp nhóm hoàn thành công việc



Hình 5.1 Logo Trello

- Git/ Gitlab: GitLab là một nền tảng quản lý mã nguồn và dự án phần mềm dựa trên web. Nó cung cấp các tính năng như quản lý mã nguồn, theo dõi vấn đề, CI / CD và nhiều tính năng khác để hỗ trợ quản lý dự án phần mềm



Hình 5.2 Logo git



Hình 5.3 Logo gitlab

- Figma: Figma là một ứng dụng gồm nhiều công cụ thiết kế mạnh mẽ trên nền tảng website. Với Figma, chúng ta có thể sáng tạo không giới hạn giao diện người dùng (UI/UX), thiết kế tạo mẫu, tạo bài đăng trên các mạng xã hội và nhiều dự án thiết kế khác



Hình 5.4 Logo fima

- Clickup: ClickUp là ứng dụng tuyệt vời giúp quản lý sự kiện, công việc cá nhân và đặc biệt là quản lý công việc nhóm một cách hiệu quả. Đồng bộ công việc từ xa, ClickUp giúp chúng ta dễ dàng nhắc nhở, trao đổi về dự án, quản lý sự kiện và theo dõi tiến trình của từng nhiệm vụ



Hình 5.5 Logo clickup

- Visual studio code: Visual Studio Code là một trong những trình soạn thảo mã nguồn rất phổ biến được các lập trình viên sử dụng. Với các ưu điểm nổi bật là sự nhanh chóng, nhẹ, hỗ trợ đa nền tảng cùng nhiều tính năng và là mã nguồn mở chính. Visual Studio Code ngày càng được ưa chuộng sử dụng, là lựa chọn hàng đầu của các lập trình viên



Hình 5.6 Logo Visual Studio Code

- ExpressJs: ExpressJs được mệnh danh là bộ khung web nhanh gọn, linh hoạt và tối giản dành cho Node.js, được sử dụng rộng rãi trong các môi trường sản xuất trên toàn thế giới



Hình 5.7 Logo ExpressJs

- MongoDB: MongoDB là một hệ quản trị cơ sở dữ liệu theo kiểu tài liệu (document-oriented), khác biệt hoàn toàn so với mô hình quan hệ (relational)

truyền thống. Lý do chính cho sự thay đổi này là để việc mở rộng hệ thống (scale out) dễ dàng hơn, nhưng bên cạnh đó còn nhiều ưu điểm khác



Hình 5.8 Logo MongoDB

5.2. Những chức năng đã hoàn thành

TÀI LIỆU THAM KHẢO

3. Marijn Haverbeke, *Eloquent JavaScript: A Modern Introduction to Programming* (Nơi xuất bản: No Starch Press, 2024), trang 8, 269.
4. David Gutman, *Fullstack Node.js: The Complete Guide to Building Production Apps with Node.js* (Nơi xuất bản: Fullstack.io, 2019), trang 1, 20.
5. Kristina Chodorow, *MongoDB: The Definitive Guide, Second Edition* (Nơi xuất bản: O'Reilly Media, Inc., 2013), trang 3.
6. Mark Massé, *REST API Design Rulebook* (Nơi xuất bản: O'Reilly Media, Inc., 2012), trang 5-6, 11-13, 23-32.

CHECK LIST CỦA BÁO CÁO

STT	Nội dung công việc	Có	Không	Ghi chú
1	Báo cáo được trình bày (định dạng) đúng với yêu cầu.			
2	Báo cáo có số lượng trang đáp ứng đúng yêu cầu (50-80 trang)			
3	Báo cáo trình bày được đầy đủ phần mở đầu			
4	Báo cáo trình bày được cơ sở lý thuyết phù hợp với nội dung của đề tài và yêu cầu			
5	Nội dung chính của đề tài được trình bày hợp lý như đặt vấn đề rõ ràng, giải quyết vấn đề và kết quả.			
6	Báo cáo có phần kết luận và hướng phát triển của đề tài (Kết luận về kết quả đề tài và kết quả của bản thân thu được qua quá trình thực tập tại Doanh nghiệp)			

PHỤ LỤC

