

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM
KHOA ĐÀO TẠO CHẤT LƯỢNG CAO



HCMUTE

TIÊU LUẬN CHUYÊN NGÀNH

XÂY DỰNG WEBSITE CHIA SẺ HÌNH ẢNH

SVTH

Ngô Thị Trang

Lê Huy Hiệp

Khoá 2017 - 2021

Ngành CÔNG NGHỆ THÔNG TIN

GVHD ThS. TRẦN CÔNG TÚ

MSSV

17110241

17110137

TP.HCM, ngày 20, tháng 12, năm 2020



CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

Tp. Hồ Chí Minh, 23 tháng 12, 2020

NHIỆM VỤ TIỂU LUẬN CHUYÊN NGÀNH

Họ tên sinh viên: Ngô Thị Trang

MSSV: 17110241

Họ tên sinh viên: Lê Huy Hiệp

MSSV: 17110137

Chuyên ngành: Công nghệ thông tin

Lớp: 17110CLST3

Giảng viên hướng dẫn: ThS. TRẦN CÔNG TÚ

Điện thoại: 0983 674 375

Ngày nhận đề tài: 27/08/2020

Ngày nộp đề tài:

1. Tên đề tài: Xây dựng website chia sẻ hình ảnh.

2. Các số liệu, tài liệu ban đầu:

Hiện thực thuật toán bằng HTML, CSS, JS, Bootstrap, ASP.NET CORE 3.x, SQL Server, Angular.

Thiết kế trang web chia sẻ hình ảnh.

3. Nội dung thực hiện đề tài:

Nghiên cứu các thiết kế được một website chia sẻ hình ảnh với đầy đủ các tính năng như: đăng bài viết, xem và bình luận hình ảnh,...

Phân tích chức năng của website và lựa chọn phương án thiết kế website phù hợp.

4. Sản phẩm:

Thiết kế và Xây dựng website chia sẻ hình ảnh có đầy đủ các chức năng cơ bản.

TRƯỞNG NGÀNH

(Kí và ghi rõ họ tên)

GIẢNG VIÊN HƯỚNG DẪN

(Kí và ghi rõ họ tên)



CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

Tp. Hồ Chí Minh, ____ tháng 12, 2020

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên sinh viên: Ngô Thị Trang

Mã số sinh viên: 17110241

Họ và tên sinh viên: Lê Huy Hiệp

Mã số sinh viên: 17110137

Ngành: Công nghệ thông tin.

Tên đề tài: Xây dựng website chia sẻ hình ảnh.

Họ và tên giáo viên hướng dẫn: ThS. TRẦN CÔNG TÚ

NHẬN XÉT

1. Về nội dung đề tài & khối lượng thực hiện:

.....
.....
.....

2. Ưu điểm:

.....
.....
.....

3. Khuyết điểm:

.....
.....
.....

4. Đề nghị cho bảo vệ hay không?

.....

5. Đánh giá loại:

.....

6. Điểm: (bằng chữ.....)

Tp. Hồ Chí Minh, ngày ____ tháng 12, 2020

GIẢNG VIÊN HƯỚNG DẪN



CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM
Độc lập – Tự do – Hạnh phúc

Tp. Hồ Chí Minh, ____ tháng 12, 2020

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

Họ và tên sinh viên: Ngô Thị Trang

Mã số sinh viên: 17110241

Họ và tên sinh viên: Lê Huy Hiệp

Mã số sinh viên: 17110137

Ngành: Công nghệ thông tin.

Tên đề tài: Xây dựng website chia sẻ hình ảnh.

Họ và tên giáo viên phản biện:

NHẬN XÉT

7. Về nội dung đề tài & khối lượng thực hiện:

.....

.....

.....

8. Ưu điểm:

.....

.....

.....

9. Khuyết điểm:

.....

.....

.....

10. Đề nghị cho bảo vệ hay không?

.....

11. Đánh giá loại:

.....

12. Điểm: (bằng chữ.....)

Tp. Hồ Chí Minh, ngày ____ tháng 12, 2020

GIẢNG VIÊN PHẢN BIỆN

LỜI CẢM ƠN

Lời đầu tiên, nhóm thực hiện xin được phép gửi lời cảm ơn chân thành đến khoa Đào tạo Chất Lượng Cao – Trường Đại học Sư phạm Kỹ thuật Thành phố Hồ Chí Minh đã tạo mọi điều kiện thuận lợi nhất cho nhóm thực hiện có cơ hội được tự do tiếp cận, tham khảo, mở rộng thêm kiến thức trong lĩnh vực Công nghệ thông tin nói chung và môn Tiểu Luận Chuyên Ngành nói riêng.

Lời cảm ơn trân trọng nhất nhóm thực hiện xin chân thành gửi đến Thầy **Trần Công Tú** – người đã dùng mọi tâm huyết và tri thức của người Thầy, cùng đồng hành và trực tiếp giảng dạy, hướng dẫn và tạo mọi điều kiện thuận lợi giúp đỡ cho nhóm phát huy hết khả năng cũng như nâng cao kiến thức trong suốt quá trình học tập, đặc biệt là trong quá trình chuẩn bị và thực hiện đề tài môn học. Cảm ơn sự nhiệt tình của Thầy, là động lực vô cùng to lớn giúp nhóm thực hiện thực hiện, kiên trì trong suốt quá trình thực hiện đề tài và khám phá ra những kiến thức mới đầy thú vị và bổ ích liên quan đến đề tài mà cụ thể là đề tài **Xây dựng website Chia Sẻ Hình Ảnh**.

TPHCM, ngày 20 tháng 12 năm 2020

Nhóm sinh viên thực hiện

LỜI MỞ ĐẦU

Hiện nay internet đã trở nên thông dụng và không thể thiếu trong cuộc sống. Và sự ra đời của website đã đóng vai trò rất quan trọng trong việc đưa mọi người đến với những thông tin mà họ đang cần. Website rút ngắn đi khoảng cách, không chỉ gói gọn trong phạm vi nhỏ mà còn phát triển ở mức toàn cầu. Website chính là sản phẩm tốt nhất của internet, nhằm mang đến những lợi ích vô cùng to lớn cũng như đem lại rất nhiều cơ hội cho chính người sử dụng. Website mang đến cho người dùng những thông tin cần thiết, những vấn đề khác nhau trong cuộc sống đều có thể tìm kiếm trên website, giải đáp mọi thắc mắc với thời gian cực nhanh chóng và cực kỳ hiệu quả.

Vào những năm gần đây, website phát triển ngày càng đa dạng trong đó có website theo hướng mạng xã hội. Có thể kể ra một số website phổ biến nhất hiện nay như Facebook, Youtube, Instagram, Pinterest,... Rất khó để người dùng cưỡng lại sự thu hút của các website này, nhất là khi được dùng miễn phí. Thậm chí Facebook, Gmail đã trở thành một phần không thể thiếu trong cuộc sống của nhiều người.

Từ đó thấy được website theo hướng mạng xã hội đã dần trở nên vô cùng quan trọng trong cuộc sống. Dựa trên điều đó nhóm quyết định xây dựng một website chia sẻ hình ảnh để tạo điều kiện cho mọi người giao lưu, liên kết, chia sẻ những sở thích, sự quan tâm, những ý tưởng dễ dàng với nhau. Website này sẽ là một kho ảnh cực kỳ lớn với vô số các hình ảnh đẹp, thú vị, độc đáo; người dùng có thể thoải mái xem và tải miễn phí bất cứ hình ảnh nào; và có thể tìm kiếm các ý tưởng thông qua các hình ảnh theo chủ đề mà họ muốn.

MỤC LỤC

LỜI CẢM ƠN	VI
LỜI MỞ ĐẦU	VII
MỤC LỤC	VIII
DANH MỤC CÁC CHỮ VIẾT TẮT	XI
DANH MỤC CÁC BẢNG BIỂU.....	XII
DANH MỤC CÁC HÌNH ẢNH, BIỂU ĐỒ.....	XIII
CHƯƠNG 1: TỔNG QUAN.....	1
1.1. Lý do chọn đề tài.....	1
1.2. Mục tiêu	1
1.3. Công nghệ sử dụng	1
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT	2
2.1. Angular Framework.....	2
2.2. ASP.NET Core	2
2.3. Microsoft SQL Server 2017	3
2.4. Swagger UI	4
CHƯƠNG 3: NỘI DUNG	5
3.1. Khảo sát các sản phẩm liên quan tới đề tài.....	5
3.1.1. Pinterest.com.....	5
3.1.2. Facebook.com	6
3.1.3. Cooky.vn.....	8
3.1.4. Kết luận.....	8
3.2. Mô hình hoá yêu cầu	10
3.2.1. Thiết kế use case diagram.....	10
3.2.2. Đặc tả use case.....	10
3.3. Sơ đồ trình tự (Sequence diagram).....	29
3.3.1. Tìm bài viết.....	29
3.3.2. Bình luận.....	29
3.3.3. Đăng nhập	30
3.3.3. Tìm kiếm.....	31

3.4. Thiết kế cơ sở dữ liệu	32
3.4.1. Mô hình liên kết thực thể.....	32
3.4.2. Mô hình quan hệ các bảng trong cơ sở dữ liệu.....	32
3.5. Thiết kế giao diện.....	33
3.5.1. Giao diện “Người quản trị - Admin”	33
3.5.2. Giao diện “Khách”	35
3.5.3. Giao diện “Người dùng đã xác thực”	36
3.5.3. Giao diện Swagger UI	38
CHƯƠNG 4: CÀI ĐẶT SẢN PHẨM	41
4.1. Front-end.....	41
4.1.1. Cấu trúc ứng dụng.....	41
4.1.2. Các đoạn xử lý chính trong front-end.....	44
4.2. Back-end	45
4.2.1. Cấu trúc project API	45
4.2.2. Một số Function	46
CHƯƠNG 5: TỔNG KẾT	50
5.1. Kết quả đạt được	50
5.2. Ưu điểm	50
5.3. Nhược điểm	50
5.4. Khó khăn	51
5.5. Bài học kinh nghiệm.....	51
5.6. Hướng phát triển	51
TÀI LIỆU THAM KHẢO.....	52
PHỤ LỤC	53
1. Cài đặt Node.js và npm	53
2. Cài đặt Visual Studio Code.....	56
3. Cài đặt Angular CLI	56
3. Tạo một ứng dụng Web Angular	57
4. Cài đặt .NET Core SDK.....	59
5. Cài đặt Visual Studio Code.....	60

DANH MỤC CÁC CHỮ VIẾT TẮT

Stt	Từ viết tắt	Nghĩa
1	HTML	Hypertext Markup Language
2	CSS	Cascading Style Sheets
3	JS	JavaScript
4	ASP-NET	Active Server Pages - Network Enabled Technologies
5	SQL	Structured Query Language
6	MVC	Model-View-Controller
7	SPA	Single Page Application
8	DOM	Document Object Model
9	API	Application Programming Interface
10	UI	User Interface
11	DI	Dependency Injection
12	IIS	Internet Information Services
13	OLTP	On-line Transactional Processing
14	APIs	Application Programming Interfaces
15	AI	Artificial Intelligence
16	OTP	One Time Password
17	CLI	Command Line Interface
18	SDK	Software Development Kit

DANH MỤC CÁC BẢNG BIỂU

Bảng 3.2. Đặc tả usecase Đăng ký bằng email	11
Bảng 3.3. Đặc tả usecase Đăng nhập bằng tài khoản hệ thống	12
Bảng 3.4. Đặc tả usecase Tìm kiếm hình ảnh.....	16
Bảng 3.5. Đặc tả usecase Xem thông báo.....	17
Bảng 3.6. Đặc tả usecase Xem chi tiết bài viết.....	18
Bảng 3.7. Đặc tả usecase Bình luận bài viết.....	20
Bảng 3.8. Đặc tả usecase Xem trang cá nhân	21
Bảng 3.9. Đặc tả usecase Tạo bài viết	22
Bảng 3.10. Đặc tả usecase Xóa bài viết.....	25
Bảng 3.11. Đặc tả usecase Xóa bình luận.....	27

DANH MỤC CÁC HÌNH ẢNH, BIỂU ĐỒ

Hình 1. Usecase Diagram	10
Hình 2. Usecase tổng quát của Khách	10
Hình 3. Usecase tổng quát của Người dùng đã xác thực	15
Hình 4. Usecase bình luận bài viết	20
Hình 5. Usecase tổng quát của Người quản trị	25
Hình 6. Sơ đồ trình tự - Tìm bài viết	29
Hình 7. Sơ đồ trình tự - Bình luận	29
Hình 8. Sơ đồ trình tự - Đăng nhập	30
Hình 9. Sơ đồ trình tự - Tìm kiếm	31
Hình 10. Mô hình liên kết thực thể.....	32
Hình 11. Mô hình quan hệ.	32
Hình 12. Giao diện Danh sách người dùng	33
Hình 13. Giao diện Thêm mới người dùng.....	33
Hình 14. Giao diện Chỉnh sửa người dùng	33
Hình 15. Giao diện Danh sách bài viết	34
Hình 16. Giao diện Tạo bài viết.....	34
Hình 17. Giao diện Danh sách bình luận	34
Hình 18. Giao diện Đăng nhập	35
Hình 19. Giao diện Đăng ký	35
Hình 20. Giao diện Trang chủ	36
Hình 21. Giao diện Trang cá nhân.....	36
Hình 22. Giao diện Chi tiết bài viết.....	37
Hình 23. Giao diện Tạo bài viết.....	37
Hình 24. Giao diện Chỉnh sửa hồ sơ.....	38
Hình 25. Giao diện Swagger – Tài khoản.....	38
Hình 26. Giao diện Swagger – Bài viết	39
Hình 27. Giao diện Swagger – Bình luận	39
Hình 28. Giao diện Swagger - Thông báo	39
Hình 29. Giao diện Swagger - Tìm kiếm.....	39
Hình 30. Giao diện Swagger - Lược đồ.....	40

Hình 31. Cấu trúc tổng quát của Angular Project.....	41
Hình 32. Cấu trúc thư mục src	42
Hình 33. Cấu trúc thư mục app	43
Hình 34. HTML của upload file	44
Hình 35. Xử lý dữ liệu	44
Hình 36. Xử lý giao diện khi upload.	45
Hình 37. HTML cho hiện hình ảnh sau khi upload	45
Hình 38. AutoMapperProfile.cs.....	46
Hình 39. Post.cs	47
Hình 40. PostReponse.cs	47
Hình 41. PostController.cs.....	47
Hình 42. PostService.cs	48
Hình 43. PostService.cs	49
Hình 44. CreateNotificationRequest.cs	49
Hình 45. Tải Node.js.....	53
Hình 46. Cài đặt Node.js.....	53
Hình 47. Cài đặt Node.js.....	54
Hình 48. Cài đặt Node.js.....	54
Hình 49. Cài đặt Node.js.....	55
Hình 50. Cài đặt Node.js.....	55
Hình 51. Cài đặt Node.js.....	56
Hình 52, Cài đặt Angular.....	57
Hình 53. Kết quả cài đặt Angular	57
Hình 54. Tạo project Angular	57
Hình 55. Cấu trúc project Angular	58
Hình 56. Chạy project Angular.....	58
Hình 57 Giao diện khi tạo project trong Angular	59
Hình 58. Tải .NET Core 3.1	60
Hình 59. Cài đặt .NET Core SDK	60
Hình 60. Cài đặt Visual Studio 2019.....	61

Chương 1: TỔNG QUAN

1.1. Lý do chọn đề tài

Cùng với quá trình toàn cầu hóa và sự phát triển của công nghệ thông tin, mạng internet trên thế giới và Việt Nam ngày càng phát triển mạnh mẽ. Sự tham gia của các cá nhân trên mạng ngày càng tích cực và nhu cầu chia sẻ thông tin, kết nối bạn bè là nhu cầu thiết yếu thúc đẩy sự ra đời và phát triển của các website. Hiện nay, tình hình phát triển của các website đang ngày trở nên mạnh mẽ. Số lượng người truy cập và đăng ký thành viên ở các mạng xã hội này ngày càng tăng. Điển hình như một số website mạng xã hội: facebook.com, cooky.vn, pinterest.com,... Tuy nhiên, dù đã có rất nhiều các trang mạng xã hội như thế nhưng nhu cầu của người dùng vẫn rất cao nên nhóm đã nghiên cứu xây dựng website chia sẻ hình ảnh nhằm chia sẻ nhu cầu và tạo nhiều lựa chọn cho người dùng.

1.2. Mục tiêu

Thiết kế và xây dựng được một website Chia Sẻ Hình Ảnh với đầy đủ các tính năng như: đăng tải hình ảnh, tìm kiếm và xem các hình ảnh của nhiều người.

Thiết kế giao diện người dùng gần gũi, dễ sử dụng. Với giao diện người quản trị đơn giản trực quan có các chức năng để quản lý người dùng.

1.3. Công nghệ sử dụng

- Front-end: Angular 9 có sử dụng HTML, CSS, JS, Bootstrap 4.0
- Back-end: ASP.NET Core.
- Database: SQL Server.

Chương 2: CƠ SỞ LÝ THUYẾT

2.1. Angular Framework

Angular là một JavaScript framework dùng để viết giao diện web (Front-end) do google phát triển. Hiểu đơn giản thì Angular là một khung làm việc của JavaScript MVC phía máy khách (client) nhằm phát triển ứng dụng web động.[1]

Angular là một JavaScript framework rất mạnh. Lập trình viên thường sử dụng Angular để xây dựng project Single Page Application (SPA). Angular cung cấp các tính năng tích hợp cho animation, http service và có các tính năng như auto-complete, navigation, toolbar, menus,... Code được viết bằng TypeScript, biên dịch thành JavaScript và hiển thị tương tự trong trình duyệt.[2]

Để sử dụng tốt Angular, người dùng cần nắm các khái niệm cơ bản như: HTML, CSS, JavaScript, TypeScript, Document Object Model (DOM).[1]

Ưu điểm khi sử dụng Angular:

- Ứng dụng được giảm tối đa kích thước và tăng tối đa hiệu suất.[1]
- HTML linh hoạt hơn.[1]
- Code HTML mạnh mẽ hơn với những đặc trưng như IF, FOR, LOCAL VARIABLES,...[1]
- Dễ dàng hiển thị các fields từ data model của website và theo dõi những thay đổi, cập nhật lại từ người dùng nhờ binding data.[1]
- Xây dựng và tái sử dụng nhờ vào những khối module độc lập.[1]
- Nhanh chóng giải quyết các bài toán logic nhờ back-end service hỗ trợ giao tiếp.[1]

2.2. ASP.NET Core

ASP.NET Core là một web framework mã nguồn mở và được tối ưu hóa cho cloud để phát triển các ứng dụng web chạy trên nhiều nền tảng như Windows, Linux và Mac. Hiện tại, nó bao gồm MVC framework được kết hợp các tính năng của MVC và Web API thành một web framework duy nhất.[3]

Ưu điểm khi sử dụng ASP.NET Core:

- ASP.NET Core không phải là phiên bản tiếp theo của ASP.NET. Nó là một cái tên mới được xây dựng từ đầu. Nó có một sự thay đổi lớn về kiến trúc và kết quả là nó gọn hơn, phân chia module tốt hơn.[3]
- ASP.NET Core không còn dựa trên System.Web.dll. Nó dựa trên một tập hợp nhiều yếu tố của Nuget packages. Điều này cho phép bạn tối ưu ứng dụng của mình chỉ cần những NuGet packages cần thiết.[3]
- Lợi ích của diện tích bề mặt ứng dụng nhỏ hơn thì bảo mật chặt chẽ hơn, giảm dịch vụ, cải thiện hiệu suất và giảm chi phí.[3]
- ASP.NET Core có thể chạy trên cả .NET Core hoặc full .NET Framework.
- Xây dựng và chạy được các ứng dụng ASP.NET Core trên nhiều nền tảng như Windows, Mac và Linux.[3]
- Được xây dựng trên .NET Core, hỗ trợ side-by-side app versioning.[3]
- Công cụ mới giúp đơn giản hóa việc phát triển web hiện đại.[3]
- Liên kết đơn các web stack như Web UI và API Web.[3]
- Cấu hình dựa trên môi trường đám mây sẵn có.[3]
- Được xây dựng dựa trên cho DI (Dependency Injection).[3]
- Tag Helpers làm cho các Razor markup trở nên tự nhiên hơn với HTML.[3]
- Có khả năng host trên IIS hoặc self-host.[3]

2.3. Microsoft SQL Server 2017

Microsoft SQL Server là một hệ quản trị cơ sở dữ liệu quan hệ được phát triển bởi Microsoft. Là một máy chủ cơ sở dữ liệu, nó là một sản phẩm phần mềm có chức năng chính là lưu trữ và truy xuất dữ liệu theo yêu cầu của các ứng dụng phần mềm khác. Có thể chạy trên cùng một máy tính hoặc trên một máy tính khác trên mạng (bao gồm cả Internet).[4]

Microsoft SQL Server có nhiều phiên bản khác nhau, ở phiên bản SQL Server 2017 có các tính năng đột phá sau:

- Tăng cường hiệu suất mà không cần điều chỉnh: Tăng hiệu suất tốc độ truy vấn bằng cách sử dụng Adaptive Query Processing, giao dịch nhanh hơn với bộ nhớ trong OLTP và phân tích nhanh hơn 100x với Columnstore bộ nhớ trong.[4]

- Bảo vệ dữ liệu với Luôn luôn Mã hóa.[4]
- Truy vấn bất kỳ dữ liệu nào có hỗ trợ đồ thị.[4]
- Hỗ trợ bất kỳ nền tảng, bất kỳ đám mây.[4]

2.4. Swagger UI

APIs(Application Programming Interfaces) đang ngày càng trở nên phổ biến, các dịch vụ trên Internet hầu hết đều sử dụng chuẩn RESTfull APIs để cung cấp cho các đối tác 1 phần tài nguyên của mình sử dụng. Để cho các đối tác biết mình được cung cấp những tài nguyên gì và sử dụng những thông tin nào để có thể lấy được tài nguyên đó, ta cần phải có 1 công cụ hỗ trợ việc tạo document APIs giúp thuận tiện cho việc cung cấp về cách sử dụng tài nguyên thông qua APIs 1 cách hiệu quả và Swagger ra đời nhằm phục vụ cho công việc đó.[5]

Swagger là 1 open source dùng để phát triển, thiết kế, xây dựng và làm tài liệu cho các hệ thống RESTfull Web Service. Có một số phần mềm swagger như Swagger Editor, Swagger Codegen, Swagger Inspector, Swagger UI. Trong đó swagger UI được sử dụng nhiều nhất.[5]

Swagger UI là một tool cho phép bất kỳ ai - từ developers cho đến end users - có thể hình dung và tương tác với các tài nguyên API của dự án. Tool này sẽ tự động generates ra API documents từ file config Swagger, với cái nhìn trực quan và việc triển khai trở nên dễ dàng hơn cho phía client. Ngoài ra, công cụ này còn cho phép ta mockup đến api đó để xem kết quả.[6]

Chương 3: NỘI DUNG

3.1. Khảo sát các sản phẩm liên quan tới đề tài

3.1.1. Pinterest.com

Có các chức năng chính:

- Đăng ký, đăng nhập: bằng email, facebook hoặc google.
- Đăng xuất.
- Tìm kiếm hình ảnh: theo chủ đề.
- Nhắn tin: Xem và gửi tin nhắn.
- Tìm kiếm người dùng khác: bằng email hoặc tên.
- Xem thông báo.
- Cài đặt:
 - Chỉnh sửa hồ sơ.
 - Cài đặt tài khoản.
 - Thông báo.
 - Quyền riêng tư và dữ liệu.
 - Bảo mật.
- Tương tác ghim:
 - Xem chi tiết ghim.
 - Lưu ghim.
 - Tải xuống hình ảnh.
 - Thả tim cho ghim.
 - Nhận xét ghim.
 - Trả lời nhận xét của người dùng khác.
 - Thêm ảnh feedback cho ghim.
 - Báo cáo ghim.
- Theo dõi người dùng khác.
- Tạo bảng.
- Tạo ghim:
 - Tải lên hình ảnh/video.
 - Chỉnh sửa hình ảnh.

Tạo carousel.

Ưu điểm: Giao diện thân thiện. Có áp dụng AI tìm hiểu thói quen của người dùng để đưa ra các gợi ý tìm kiếm. Ngoài ra chất lượng tải lên hình ảnh tương đối tốt.

Nhược điểm:

- Các biểu tượng cho chức năng không có chú giải, trong trường hợp người dùng không biết về các biểu tượng đó thì phải click vô mới biết đó chức năng gì.
- Ở bước đăng ký bằng email không có xác thực tài khoản email có phải chính chủ không.

3.1.2. Facebook.com

Có các chức năng chính:

- Đăng ký, đăng nhập:
Bằng email hoặc số điện thoại.
Xác thực số điện thoại bằng mã OTP.
- Đăng xuất.
- Tìm kiếm: có thể tìm kiếm tên người dùng, tên trang, tên nhóm, bài viết bằng các từ khoá.
- Xem thông báo.
- Tương tác với bài viết:
Xem chi tiết bài viết.
Thả cảm xúc cho bài viết.
Chỉnh sửa bình luận.
Chia sẻ bài viết lên tường nhà cá nhân.
Ẩn bài viết.
Lưu bài viết.
Bật thông báo về bài viết.
Báo cáo bài viết.
Chỉnh sửa bài viết (Chỉ khi đó là bài của bản thân).
- Tương tác với người dùng khác:
Xem trang cá nhân.
Xem danh sách bạn bè.

Kết bạn.

Theo dõi.

Nhắn tin.

Gọi điện.

Tìm kiếm trên trang cá nhân.

Báo cáo người dùng.

Chặn người dùng.

- Tạo bài viết:

Thiết lập chế độ riêng tư cho bài viết.

Viết cảm nghĩ.

Tải lên hình ảnh/video (1 hoặc nhiều).

Chỉnh sửa hình ảnh.

Gắn thẻ bạn bè.

Thiết lập cảm xúc/hoạt động.

Check in.

- Tương tác với trang cá nhân:

Thêm ảnh đại diện.

Thêm ảnh bìa.

Chỉnh sửa giới thiệu.

Tạo album.

Thêm ảnh vào album.

Quản lý nhật ký hoạt động.

Cài đặt dòng thời gian.

Tạo trang.

Quản lý bài viết đã lưu.

- Tạo phòng họp.

- Phát trực tiếp.

- Cài đặt:

Cài đặt tài khoản.

Cài đặt bảo mật và đăng nhập.

Thông báo.

Ưu điểm: Giao diện thân thiện với người dùng. Đây đủ chức năng để đáp ứng mọi nhu cầu của người dùng.

Nhược điểm: Hình ảnh/video khi tải lên chất lượng có thể bị giảm đi.

3.1.3. Cooky.vn

Có các chức năng chính:

- Đăng ký, đăng nhập: bằng email, facebook hoặc google.
- Đăng xuất.
- Tìm kiếm: tên món ăn.
- Tương tác với bài viết:

Xem chi tiết.

Thả tim/Lưu bài viết.

Bình luận bài viết.

Xem tất cả bình luận của bài viết.

- Tương tác với người dùng khác:

Xem trang cá nhân.

Quan tâm

Xem tất cả công thức, bộ sưu tập và tài khoản người dùng quan tâm.

- Đăng bài viết (công thức).
- Xem các công thức đã lưu.
- Quản lý tài khoản:

Thông tin cá nhân.

Chỉnh sửa ảnh bìa, ảnh đại diện.

Ưu điểm: Giao diện thân thiện, đầy đủ chức năng để sử dụng.

Nhược điểm: Chưa có chế độ bảo mật cho tài khoản. Ở bước đăng ký bằng email không có xác thực tài khoản email có phải chính chủ không.

3.1.4. Kết luận

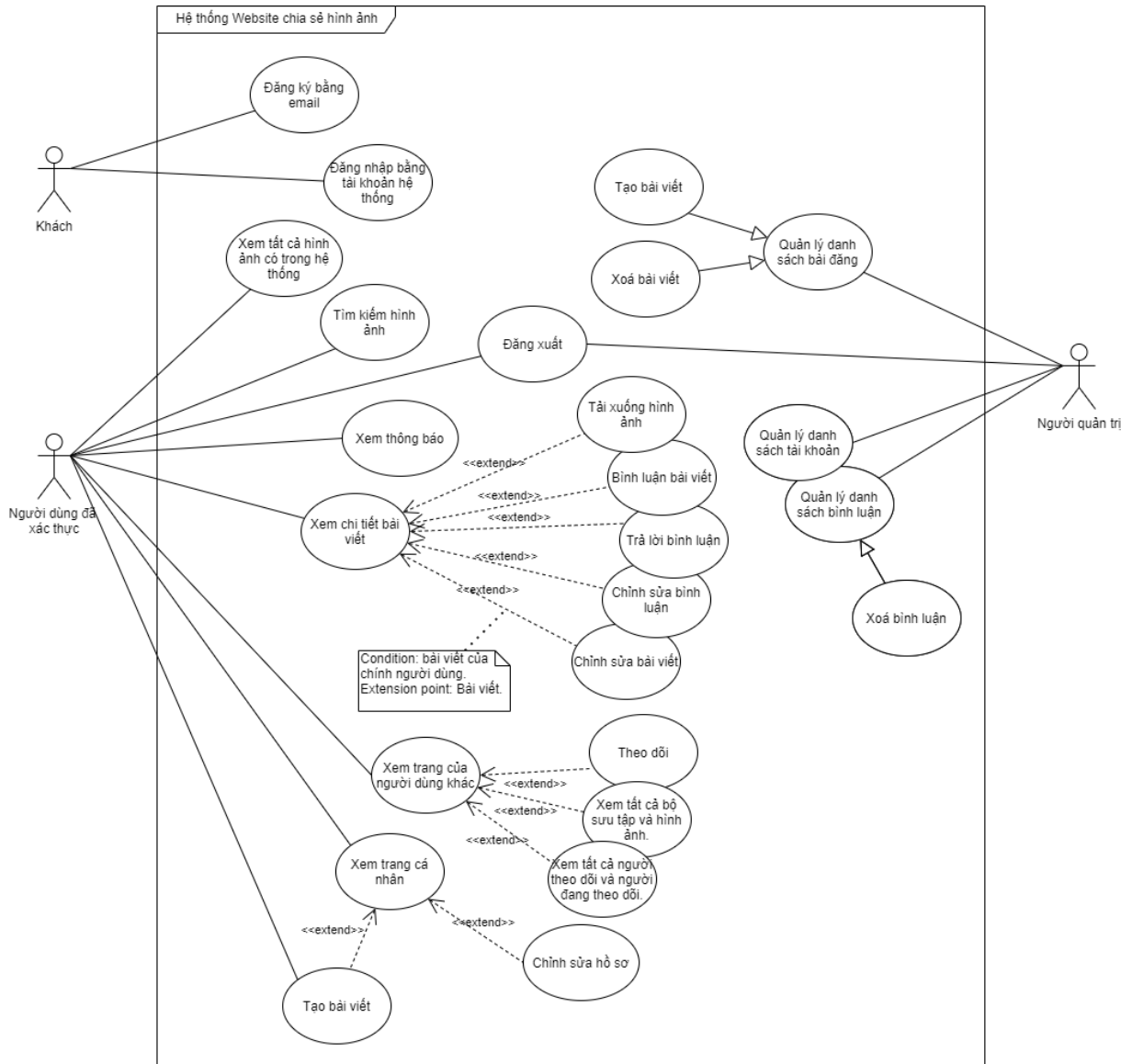
Quan khảo sát các sản phẩm trên nhóm rút ra được các chức năng cần có:

- Đăng ký, đăng nhập: bằng email, facebook hoặc google.
- Đăng xuất.
- Tìm kiếm hình ảnh: theo chủ đề.

- Xem thông báo.
- Cài đặt: Chỉnh sửa hồ sơ
- Tương tác với bài viết:
 - Xem chi tiết bài viết.
 - Chỉnh sửa bài viết (khi là của chính người dùng).
 - Tải xuống hình ảnh.
 - Bình luận cho bài viết.
 - Thả tim hoặc trả lời bình luận.
 - Chỉnh sửa bình luận (bình luận của chính người dùng).
- Tương tác với trang người dùng khác:
 - Theo dõi.
 - Xem tất cả hình ảnh của người dùng.
 - Xem tất cả người theo dõi và đang theo dõi của người dùng
- Tạo bài viết:
 - Tạo tiêu đề.
 - Tải lên hình ảnh.

3.2. Mô hình hoá yêu cầu

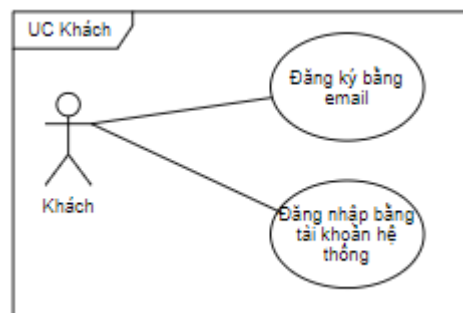
3.2.1. Thiết kế use case diagram



Hình 1. Usecase Diagram

3.2.2. Đặc tả use case

3.2.2.1. <Khách>



Hình 2. Usecase tổng quát của Khách

a. <Khách> Đăng ký bằng email

Bảng 3.1. Đặc tả usecase Đăng ký bằng email

USE CASE – CSHA_UC_01			
Use Case No.	CSHA_UC_01	Use Case Version	1.0
Use Case Name	Đăng ký		
Author	Ngô Thị Trang		
Date	9/2020	Priority	Bình thường
Actor: <ul style="list-style-type: none">- Khách.			
Summary: <ul style="list-style-type: none">- Cho phép khách có thể đăng ký 1 tài khoản của hệ thống.			
Goal: <ul style="list-style-type: none">- Khách có thể tạo thành công tài khoản của hệ thống.			
Triggers: <ul style="list-style-type: none">- Khách gửi lệnh đăng ký bằng cách nhấn vào nút Đăng ký.			
Preconditions: <ul style="list-style-type: none">- Thiết bị của khách phải có kết nối Internet.			
Post Conditions: <ul style="list-style-type: none">- Thành công: Một tài khoản mới được tạo.- Thất bại: Hệ thống hiển thị thông báo lỗi.			
Main Success Scenario:			
Step	Actor Action	System Response	
1	Khách truy cập trang web bằng URL.	Hệ thống hiển thị trang chủ của web. [Exception 1]	
2	Khách nhấn vào nút “Đăng ký”.	Hệ thống chuyển hướng đến trang đăng ký. [Exception 1]	
3	Khách nhập tên, email và mật khẩu vào các ô tương ứng.	Hệ thống kiểm tra. [Exception 3] [Exception 4] [Exception 1]	

4	Khách nhấn nút “Đăng ký”.	Hệ thống gửi tin nhắn xác thực tới email. [Exception 1]
5	Khách nhấn vào link xác thực trong tin nhắn.	Hệ thống kiểm tra và chuyển hướng sang trang đăng nhập. [Exception 1]

Alternative Scenario: N/A

Exceptions:

No	Cause	System Response
1	Internet bị mất.	Hệ thống hiển thị thông báo “Vui lòng kiểm tra kết nối internet của bạn”.
2	Email đã được đăng ký trước đó.	Hệ thống hiển thị thông báo “Email đã được đăng ký.”
3	Mật khẩu có độ dài nhỏ hơn 6 ký tự hoặc không chứa ký tự viết hoa hay không chứa ký tự chữ thường hoặc không chứa ký tự số.	Hệ thống hiển thị thông báo “Mật khẩu có độ dài ít nhất là 6 ký tự và chứa ít nhất 1 ký tự viết hoa, 1 ký tự viết thường và 1 ký tự số.”

Relationships: N/A

Business Rules:

- Email: phải là duy nhất.
- Mật khẩu: Mật khẩu có độ dài ít nhất là 6 ký tự và chứa ít nhất 1 ký tự viết hoa, 1 ký tự viết thường và 1 ký tự số.
- Hệ thống phải đảm bảo rằng mật khẩu không ai có thể đọc được.

b. <Khách> Đăng nhập bằng tài khoản hệ thống

Bảng 3.2. Đặc tả usecase Đăng nhập bằng tài khoản hệ thống

USE CASE – CSHA_UC_02			
Use Case No.	CSHA_UC_02	Use Case Version	1.0
Use Case Name	Đăng nhập bằng tài khoản hệ thống		

Author	Ngô Thị Trang		
Date	9/2020	Priority	Bình thường
Actor: <ul style="list-style-type: none">- Khách.			
Summary: <ul style="list-style-type: none">- Cho phép khách có thể đăng nhập vào hệ thống.			
Goal: <ul style="list-style-type: none">- Khách có thể tạo thành công tài khoản của hệ thống.			
Triggers: <ul style="list-style-type: none">- Khách nhập Email và Mật khẩu.- Khách gửi lệnh đăng nhập bằng cách nhấn vào nút Đăng nhập.			
Preconditions: <ul style="list-style-type: none">- Thiết bị của khách phải có kết nối Internet.			
Post Conditions: <ul style="list-style-type: none">- Thành công: Tài khoản chuyển sang trạng thái đang hoạt động.- Thất bại: Hệ thống hiển thị thông báo lỗi.			
Main Success Scenario:			
Step	Actor Action	System Response	
1	Khách nhấn vào nút “Đăng nhập” trên thanh menu.	Hệ thống chuyển hướng sang trang đăng nhập. [Exception 1]	
2	Khách nhập Email và Mật khẩu rồi nhấn nút “Đăng nhập”.	Hệ thống kiểm tra xác nhận thành công và chuyển hướng sang trang chủ. [Exception 2] [Exception 1]	
Alternative Scenario: N/A			
Exceptions:			
No	Cause	System Response	
1	Internet bị mất.	Hệ thống hiển thị thông báo “Vui lòng kiểm tra kết nối internet của bạn”.	

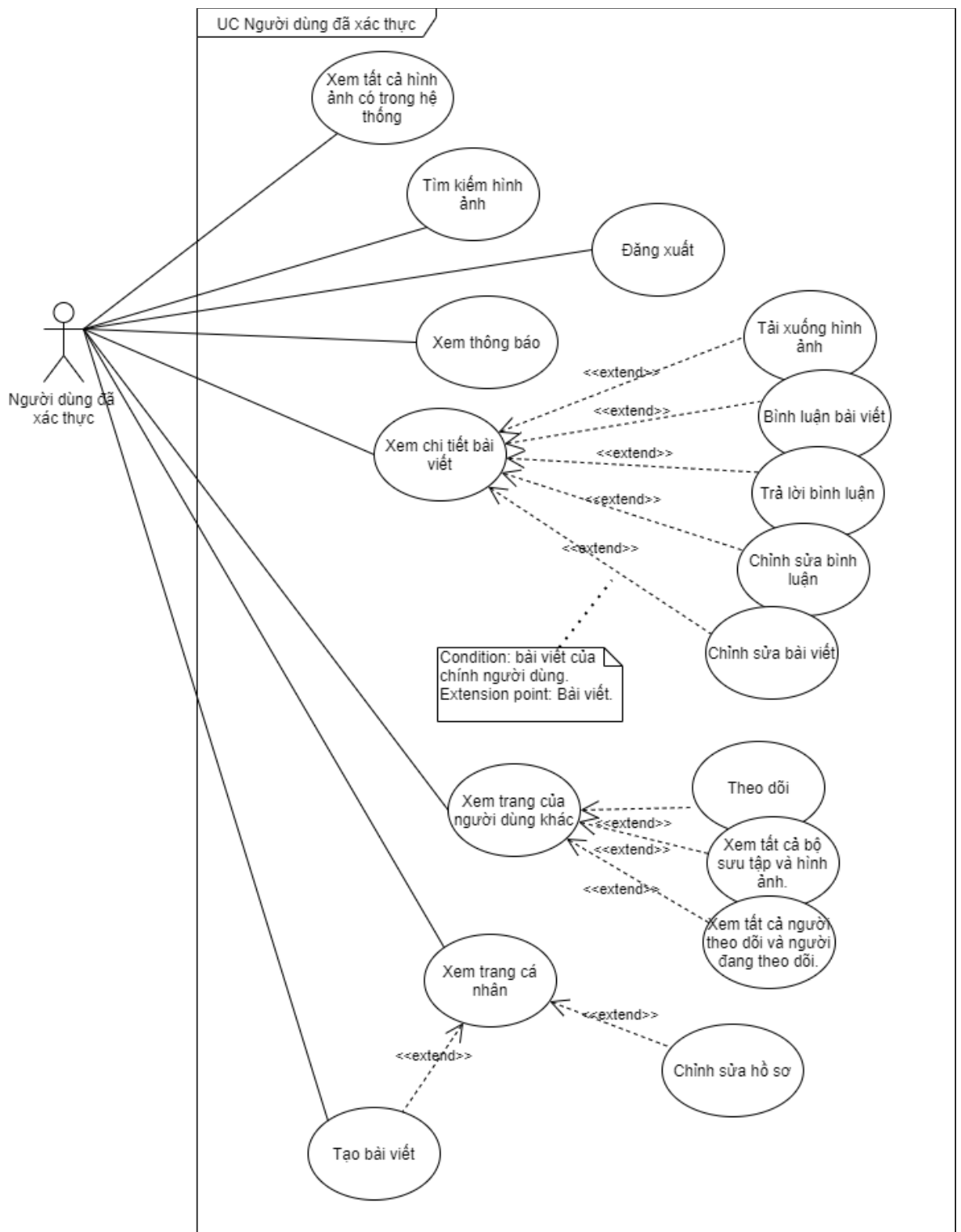
2	Khách nhập sai Email hoặc Mật khẩu.	Hệ thống hiển thị thông báo “Email hoặc Mật khẩu sai. Vui lòng nhập lại.”
---	-------------------------------------	---

Relationships: N/A

Business Rules:

- Email: không được để trống.
- Mật khẩu: không được để trống.

3.2.2.2. <Người dùng đã xác thực>



Hình 3. Usecase tổng quát của Người dùng đã xác thực

a. <Người dùng đã xác thực> Tìm kiếm hình ảnh

Bảng 3.3. Đặc tả usecase Tìm kiếm hình ảnh

USE CASE – CSHA_UC_03			
Use Case No.	CSHA_UC_03	Use Case Version	1.0
Use Case Name	Tìm kiếm hình ảnh		
Author	Ngô Thị Trang		
Date	9/2020	Priority	Cao
Actor: <ul style="list-style-type: none">- Người dùng đã xác thực.			
Summary: <ul style="list-style-type: none">- Cho phép người dùng đã xác thực có thể tìm kiếm hình ảnh trong hệ thống.			
Goal: <ul style="list-style-type: none">- Người dùng đã xác thực có thể tìm hình ảnh thành công trong hệ thống.			
Triggers: <ul style="list-style-type: none">- Người dùng đã xác thực nhập chủ đề hình ảnh muốn tìm.- Người dùng đã xác thực gửi lệnh tìm kiếm bằng cách nhấn Enter trên bàn phím.			
Preconditions: <ul style="list-style-type: none">- Thiết bị của người dùng đã xác thực phải có kết nối Internet.- Tài khoản người dùng ở trạng thái đang hoạt động.			
Post Conditions: <ul style="list-style-type: none">- Thành công: Hệ thống hiển thị tất cả hình ảnh mà tiêu đề có chứa các từ mà người dùng nhập.- Thất bại: Hệ thống hiển thị thông báo lỗi.			
Main Success Scenario:			
Step	Actor Action	System Response	
1	Người dùng nhập chủ đề hình ảnh vào ô tìm kiếm trên Menu, rồi nhấn Enter.	Hệ thống kiểm tra và hiển thị danh sách các hình ảnh phù hợp. [Exception 2] [Exception 1]	
Alternative Scenario: N/A			

Exceptions:

No	Cause	System Response
1	Internet bị mất.	Hệ thống hiển thị thông báo “Vui lòng kiểm tra kết nối internet của bạn”.
2	Không có hình ảnh nào trong hệ thống có chứa các từ người dùng nhập.	Hệ thống hiển thị thông báo “Không có kết quả phù hợp.”

Relationships: N/A**Business Rules:**

- Tìm kiếm: không được để trống.

b. <Người dùng đã xác thực> Xem thông báo*Bảng 3.4. Đặc tả usecase Xem thông báo*

USE CASE – CSHA_UC_04			
Use Case No.	CSHA_UC_04	Use Case Version	1.0
Use Case Name	Xem thông báo		
Author	Ngô Thị Trang		
Date	9/2020	Priority	Cao

Actor:

- Người dùng đã xác thực.

Summary:

- Cho phép người dùng đã xác thực có thể xem thông báo mới nhất.

Goal:

- Người dùng đã xác thực có thể xem thành công các thông báo mới nhất trong hệ thống.

Triggers:

- Người dùng đã xác thực gửi lệnh Xem thông báo bằng cách nhấn vào biểu tượng thông báo.

Preconditions:

- Tài khoản người dùng ở trạng thái đang hoạt động.

- Thiết bị của người dùng đã xác thực phải có kết nối Internet.

Post Conditions:

- Thành công: Hệ thống hiển thị tất cả thông báo mới nhất của người dùng.
- Thất bại: Hệ thống hiển thị thông báo lỗi.

Main Success Scenario:

Step	Actor Action	System Response
1	Người dùng nhấn vào biểu tượng thông báo trên thanh Menu.	Hệ thống kiểm tra và hiển thị danh sách tất cả các thông báo mới nhất. [Exception 2] [Exception 1]

Alternative Scenario: N/A

Exceptions:

No	Cause	System Response
1	Internet bị mất.	Hệ thống hiển thị thông báo “Vui lòng kiểm tra kết nối internet của bạn”.
2	Không có thông báo từ hệ thống tới người dùng.	Hệ thống hiển thị thông báo “Không có thông báo.”

Relationships: N/A

Business Rules: N/A

c. <Người dùng đã xác thực> Xem chi tiết bài viết

Bảng 3.5. Đặc tả usecase Xem chi tiết bài viết

USE CASE – CSHA_UC_05			
Use Case No.	CSHA_UC_05	Use Case Version	1.0
Use Case Name	Xem chi tiết bài viết		
Author	Ngô Thị Trang		
Date	9/2020	Priority	Cao
Actor: <ul style="list-style-type: none"> - Người dùng đã xác thực. Summary: <ul style="list-style-type: none"> - Cho phép người dùng đã xác thực có thể xem chi tiết bài viết. 			

Goal:

- Người dùng đã xác thực có thể xem thành công chi tiết của bài viết.

Triggers:

- Người dùng đã xác thực gửi lệnh Xem chi tiết bài viết bằng cách nhấn vào hình ảnh.

Preconditions:

- Tài khoản người dùng ở trạng thái đang hoạt động.
- Thiết bị của người dùng đã xác thực phải có kết nối Internet.

Post Conditions:

- Thành công: Hệ thống hiển thị chi tiết của bài viết tương ứng với hình ảnh người dùng nhấn.
- Thất bại: Hệ thống hiển thị thông báo lỗi.

Main Success Scenario:

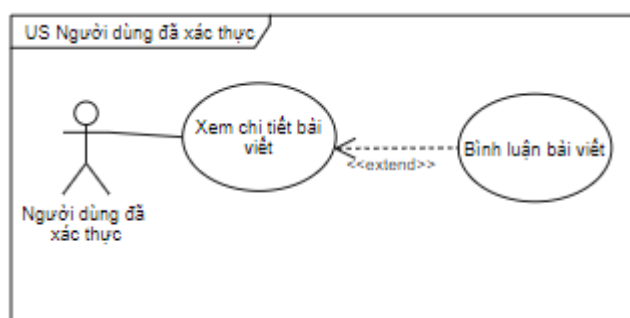
Step	Actor Action	System Response
1	Người dùng nhấn hình ảnh muốn xem chi tiết.	Hệ thống kiểm tra và hiển thị chi tiết của hình ảnh gồm các thông tin sau: <ul style="list-style-type: none"> - Tiêu đề. - Avatar và Tên tài khoản người tạo. - Danh sách các bình luận của bài viết. [Exception 1]

Alternative Scenario: N/A**Exceptions:**

No	Cause	System Response
1	Internet bị mất.	Hệ thống hiển thị thông báo “Vui lòng kiểm tra kết nối internet của bạn”.

Relationships: CSHA_UC_06**Business Rules:** N/A

d. <Người dùng đã xác thực> Bình luận bài viết



Hình 4. Usecase bình luận bài viết

Bảng 3.6. Đặc tả usecase Bình luận bài viết

USE CASE – CSHA_UC_06			
Use Case No.	CSHA_UC_06	Use Case Version	1.0
Use Case Name	Bình luận bài viết		
Author	Ngô Thị Trang		
Date	9/2020	Priority	Cao
Actor: <ul style="list-style-type: none"> - Người dùng đã xác thực. Summary: <ul style="list-style-type: none"> - Cho phép người dùng đã xác thực có thể bình luận bài viết. Goal: <ul style="list-style-type: none"> - Người dùng đã xác thực có thể bình luận bài viết thành công. Triggers: <ul style="list-style-type: none"> - Người dùng đã xác thực nhập nội dung bình luận. - Người dùng đã xác thực gửi lệnh Bình luận bài viết bằng cách nhấn vào nút “Đã xong”. Preconditions: <ul style="list-style-type: none"> - Tài khoản người dùng ở trạng thái đang hoạt động. - Thiết bị của người dùng đã xác thực phải có kết nối Internet. Post Conditions: <ul style="list-style-type: none"> - Thành công: Hệ thống hiển thị bình luận vừa đăng ở khung nhận xét của bài viết. - Thất bại: Hệ thống hiển thị thông báo lỗi. 			

Main Success Scenario:		
Step	Actor Action	System Response
1	Người dùng nhập bình luận vào ô “Viết nhận xét”.	Hệ thống kiểm tra và hiển thị bình luận ở khung nhận xét của bài viết. [Exception 1]
Alternative Scenario: N/A		
Exceptions:		
No	Cause	System Response
1	Internet bị mất.	Hệ thống hiển thị thông báo “Vui lòng kiểm tra kết nối internet của bạn”.
Relationships: CSHA_UC_05		
Business Rules:		
<ul style="list-style-type: none"> - Viết nhận xét: không được để trống. 		

e. <Người dùng đã xác thực> Xem trang cá nhân

Bảng 3.7. Đặc tả usecase Xem trang cá nhân

USE CASE – CSHA_UC_07			
Use Case No.	CSHA_UC_07	Use Case Version	1.0
Use Case Name	Xem trang cá nhân		
Author	Ngô Thị Trang		
Date	9/2020	Priority	Cao
Actor: <ul style="list-style-type: none"> - Người dùng đã xác thực. 			
Summary: <ul style="list-style-type: none"> - Cho phép người dùng đã xác thực có thể xem trang cá nhân của mình. 			
Goal: <ul style="list-style-type: none"> - Người dùng đã xác thực có thể xem trang cá nhân thành công. 			
Triggers: <ul style="list-style-type: none"> - Người dùng đã xác thực gửi lệnh Xem trang cá nhân bằng cách nhấn vào hình đại diện hoặc tên. 			

Preconditions:

- Tài khoản người dùng ở trạng thái đang hoạt động.
- Thiết bị của người dùng đã xác thực phải có kết nối Internet.

Post Conditions:

- Thành công: Hệ thống hiển thị trang cá nhân của người dùng.
- Thất bại: Hệ thống hiển thị thông báo lỗi.

Main Success Scenario:

Step	Actor Action	System Response
1	Người dùng nhấn vào hình đại diện hoặc tên của tài khoản muốn xem trang cá nhân.	Hệ thống kiểm tra và chuyển hướng sang trang cá nhân tương ứng với các thông tin sau: <ul style="list-style-type: none"> - Hình đại diện. - Tổng số người theo dõi và người đang theo dõi. - Danh sách tất cả các bài viết (nếu có). [Exception 1]

Alternative Scenario: N/A**Exceptions:**

No	Cause	System Response
1	Internet bị mất.	Hệ thống hiển thị thông báo “Vui lòng kiểm tra kết nối internet của bạn”.

Relationships: N/A**Business Rules:** N/A**f. <Người dùng đã xác thực> Tạo bài viết***Bảng 3.8. Đặc tả usecase Tạo bài viết*

USE CASE – CSHA_UC_08			
Use Case No.	CSHA_UC_08	Use Case Version	1.0
Use Case Name	Tạo bài viết		

Author	Ngô Thị Trang		
Date	9/2020	Priority	Cao
Actor: <ul style="list-style-type: none">- Người dùng đã xác thực.			
Summary: <ul style="list-style-type: none">- Cho phép người dùng đã xác thực có thể tạo bài viết mới.			
Goal: <ul style="list-style-type: none">- Người dùng đã xác thực tạo bài viết mới thành công.			
Triggers: <ul style="list-style-type: none">- Người dùng đã xác thực gửi lệnh Tạo bài viết bằng cách nhấn vào biểu tượng Tạo bài viết (+).			
Preconditions: <ul style="list-style-type: none">- Tài khoản người dùng ở trạng thái đang hoạt động.- Thiết bị của người dùng đã xác thực phải có kết nối Internet.			
Post Conditions: <ul style="list-style-type: none">- Thành công: Hệ thống hiển thị trang cá nhân của người dùng.- Thất bại: Hệ thống hiển thị thông báo lỗi.			
Main Success Scenario:			
Step	Actor Action	System Response	
1	Người dùng nhấn vào biểu tượng Tạo bài viết (+) ở góc dưới bên phải của trang chủ hoặc ở dưới hình đại diện ở trang cá nhân.	Hệ thống chuyển hướng sang trang tạo bài viết có: <ul style="list-style-type: none">- Ô “Nhập tiêu đề”.- Nút “Chọn tệp”.- Khung hiện hình ảnh.- Nút “Đăng”. [Exception 1]	
2	Người dùng nhập tiêu đề rồi nhấn vào nút “Chọn tệp” để tải hình ảnh lên.	Hệ thống xuất hiện khung File Explorer để người dùng chọn hình ảnh muốn tải lên. [Exception 1]	

3	Người dùng điều hướng đến tệp hình ảnh muốn tải lên rồi nhấn Open.	Hệ thống kiểm tra và hiển thị hình ảnh người dùng vừa chọn ở khung hình. [Exception 1]
4	Người dùng nhấn nút “Đăng” để đăng tải bài viết.	Hệ thống kiểm tra thành công và hiển thị thông báo “Tạo bài viết thành công!”. [Exception 1] [Exception 2]
5	Người dùng nhấn nút “OK”.	Hệ thống điều hướng quay lại trang trước đó và hiển thị bài viết. [Exception 1]

Alternative Scenario: N/A

Exceptions:

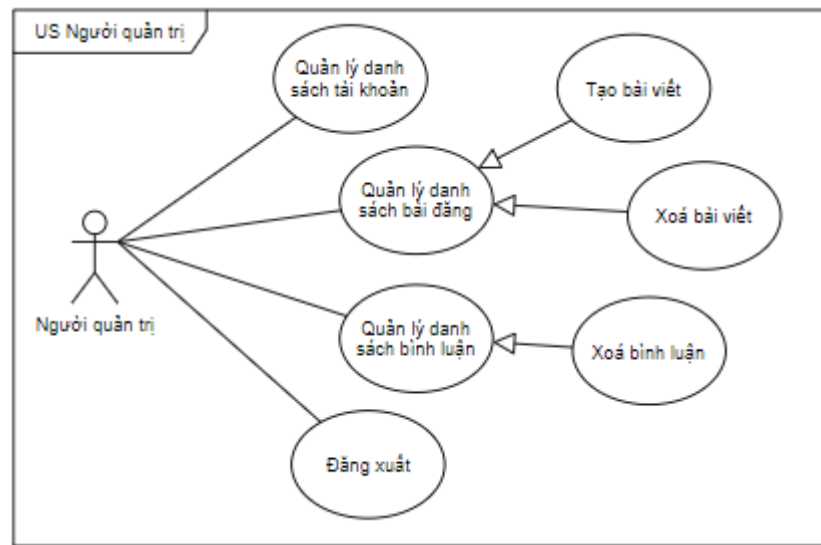
No	Cause	System Response
1	Internet bị mất.	Hệ thống hiển thị thông báo “Vui lòng kiểm tra kết nối internet của bạn”.

Relationships: N/A

Business Rules:

- Tiêu đề: không được để trống.
- Tệp hình: không được để trống.

3.2.2.3. <Người quản trị>



Hình 5. Usecase tổng quát của Người quản trị

a. <Người quản trị> Xoá bài viết

Bảng 3.9. Đặc tả usecase Xoá bài viết

USE CASE – CSHA_UC_09			
Use Case No.	CSHA_UC_09	Use Case Version	1.0
Use Case Name	Xoá bài viết		
Author	Ngô Thị Trang		
Date	9/2020	Priority	Bình thường
Actor: <ul style="list-style-type: none"> - Người quản trị. Summary: <ul style="list-style-type: none"> - Cho phép người quản trị có thể xoá bài viết. Goal: <ul style="list-style-type: none"> - Người quản trị xoá bài viết thành công. Triggers: <ul style="list-style-type: none"> - Người quản trị gửi lệnh Xoá bài viết bằng cách nhấn vào biểu tượng Xoá. Preconditions: <ul style="list-style-type: none"> - Tài khoản người quản trị ở trạng thái đang hoạt động. - Thiết bị của người dùng phải có kết nối Internet. Post Conditions:			

- Thành công: Hệ thống xoá bài viết khỏi danh sách.
- Lỗi: Hệ thống hiển thị thông báo lỗi.

Step	Actor Action	System Response
1	Người quản trị nhấn vào “Posts” trên thanh menu.	Hệ thống chuyển hướng sang trang danh sách tất cả các bài viết có trong hệ thống. [Exception 1]
2	Người quản trị nhấn vào biểu tượng “Xoá” ở hàng bài viết muốn xoá.	Hệ thống xuất hiện thông báo “Are you sure you want to delete this post?” với 2 nút “OK” và “Huỷ” để người quản trị lựa chọn. [Exception 1]
3	Người quản trị nhấn nút “OK”.	Hệ thống kiểm tra và hiển thị danh sách các bài viết sau khi đã xoá bài viết đó. [Exception 1]

Alternative Scenario:

Step	Actor Action	System Response
1	Người dùng nhấn nút “Huỷ”.	Hệ thống tắt thông báo.

Exceptions:

No	Cause	System Response
1	Internet bị mất.	Hệ thống hiển thị thông báo “Vui lòng kiểm tra kết nối internet của bạn”.

Relationships: N/A

Business Rules: N/A

b. <Người quản trị> Xóa bình luận

Bảng 3.10. Đặc tả usecase Xóa bình luận

USE CASE – CSHA_UC_10												
Use Case No.	CSHA_UC_10	Use Case Version	1.0									
Use Case Name	Xóa bình luận											
Author	Ngô Thị Trang											
Date	9/2020	Priority	Bình thường									
Actor: <ul style="list-style-type: none">- Người quản trị. Summary: <ul style="list-style-type: none">- Cho phép người quản trị có thể xóa bình luận. Goal: <ul style="list-style-type: none">- Người quản trị xóa bình luận thành công. Triggers: <ul style="list-style-type: none">- Người quản trị gửi lệnh Xóa bình luận bằng cách nhấn vào biểu tượng Xóa. Preconditions: <ul style="list-style-type: none">- Tài khoản người quản trị ở trạng thái đang hoạt động.- Thiết bị của người dùng phải có kết nối Internet. Post Conditions: <ul style="list-style-type: none">- Thành công: Hệ thống xóa bình luận khỏi danh sách.- Lỗi: Hệ thống hiển thị thông báo lỗi. <table><tr><th>Step</th><th>Actor Action</th><th>System Response</th></tr><tr><td>1</td><td>Người quản trị nhấn vào “Comments” trên thanh menu.</td><td>Hệ thống chuyển hướng sang trang danh sách tất cả các bình luận có trong hệ thống. [Exception 1]</td></tr><tr><td>2</td><td>Người quản trị nhấn vào biểu tượng “Xóa” ở hàng bình luận muốn xóa.</td><td>Hệ thống xuất hiện thông báo “Are you sure you want to delete this comments?” với 2 nút “OK” và “Huỷ” để người quản trị lựa chọn.</td></tr></table>				Step	Actor Action	System Response	1	Người quản trị nhấn vào “Comments” trên thanh menu.	Hệ thống chuyển hướng sang trang danh sách tất cả các bình luận có trong hệ thống. [Exception 1]	2	Người quản trị nhấn vào biểu tượng “Xóa” ở hàng bình luận muốn xóa.	Hệ thống xuất hiện thông báo “Are you sure you want to delete this comments?” với 2 nút “OK” và “Huỷ” để người quản trị lựa chọn.
Step	Actor Action	System Response										
1	Người quản trị nhấn vào “Comments” trên thanh menu.	Hệ thống chuyển hướng sang trang danh sách tất cả các bình luận có trong hệ thống. [Exception 1]										
2	Người quản trị nhấn vào biểu tượng “Xóa” ở hàng bình luận muốn xóa.	Hệ thống xuất hiện thông báo “Are you sure you want to delete this comments?” với 2 nút “OK” và “Huỷ” để người quản trị lựa chọn.										

			[Exception 1]	
	3	Người quản trị nhấn nút “OK”.	Hệ thống kiểm tra và hiển thị danh sách các bình luận sau khi đã xóa bình luận đó. [Exception 1]	

Alternative Scenario:

Step	Actor Action	System Response
1	Người dùng nhấn nút “Hủy”.	Hệ thống tắt thông báo.

Exceptions:

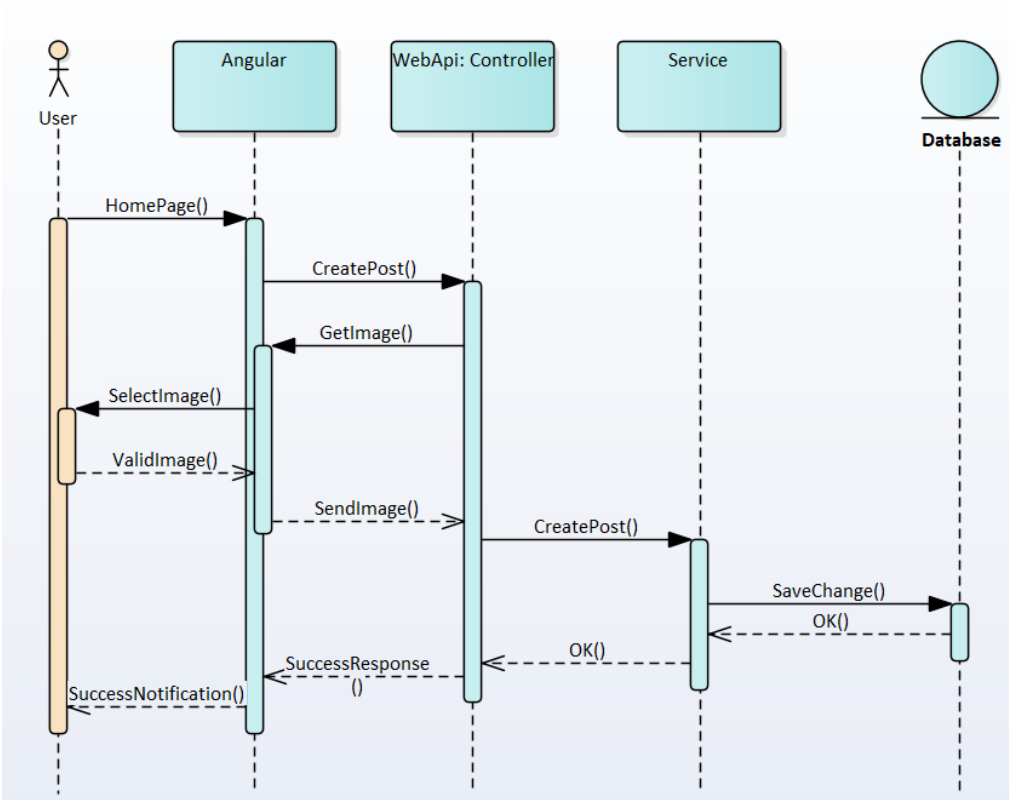
No	Cause	System Response
1	Internet bị mất.	Hệ thống hiển thị thông báo “Vui lòng kiểm tra kết nối internet của bạn”.

Relationships: N/A

Business Rules: N/A

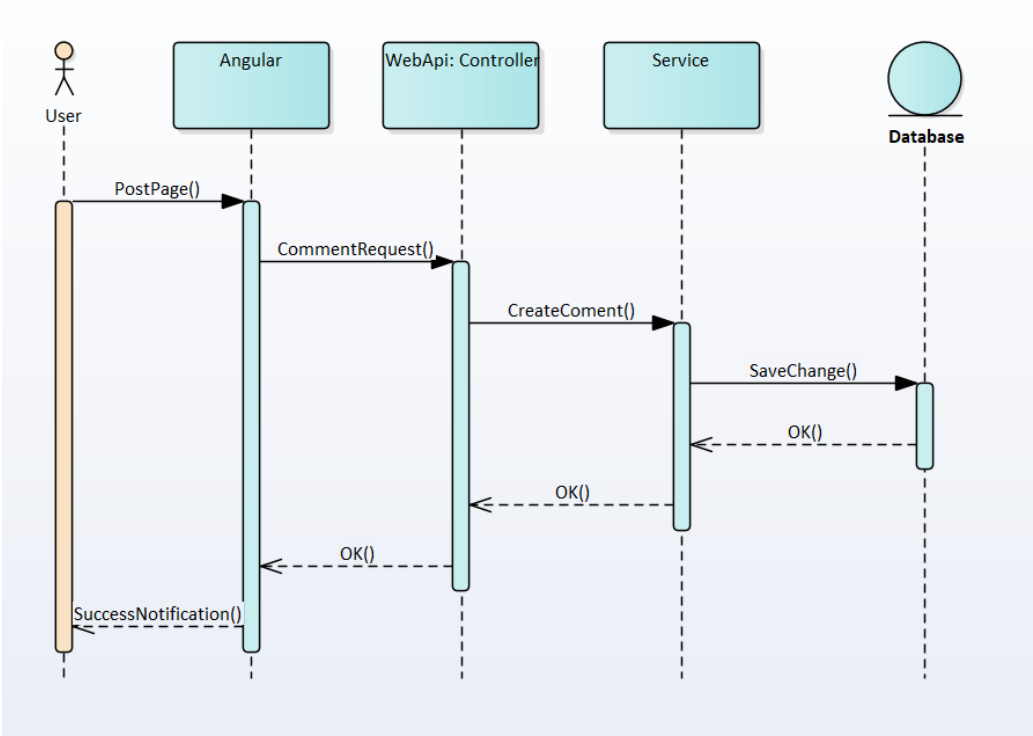
3.3. Sơ đồ trình tự (Sequence diagram)

3.3.1. Tìm bài viết



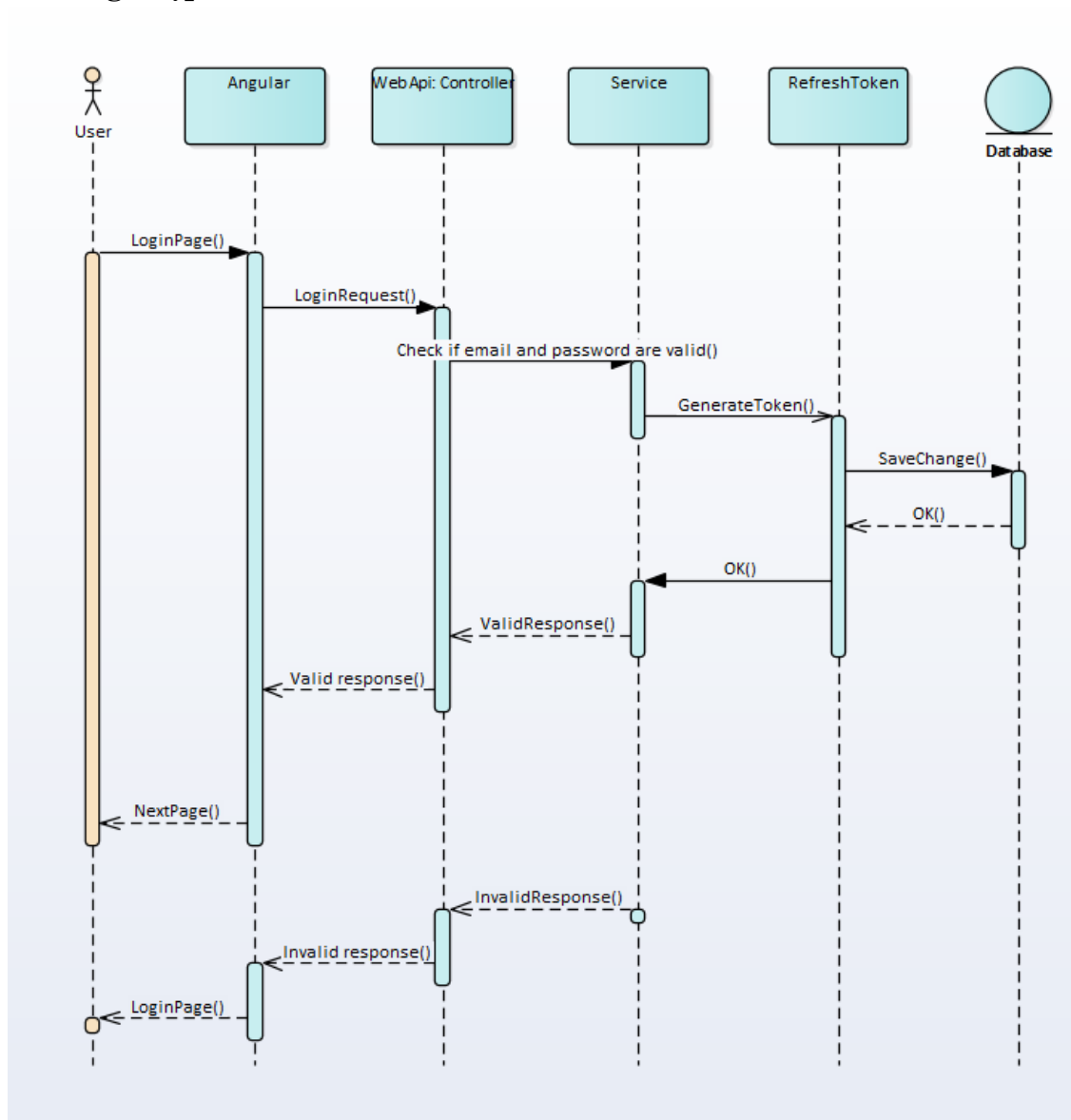
Hình 6. Sơ đồ trình tự - Tìm bài viết

3.3.2. Bình luận



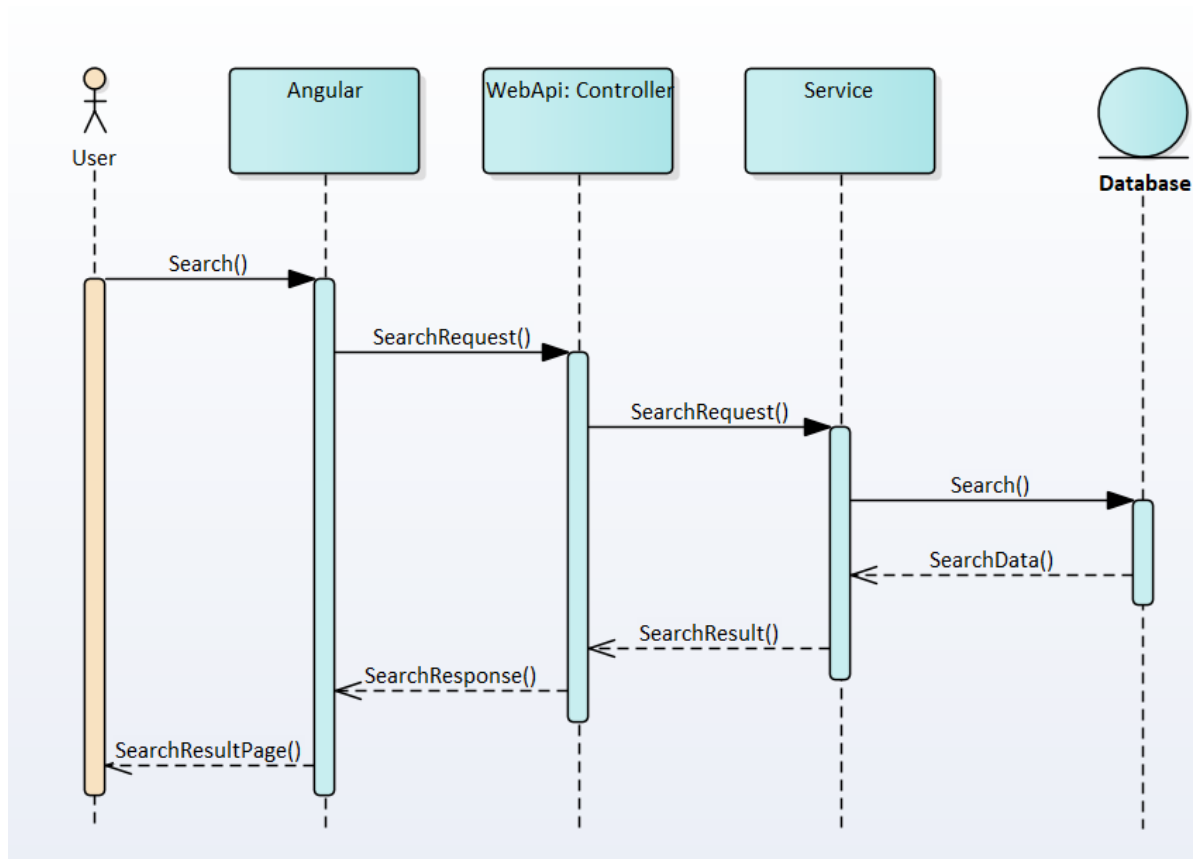
Hình 7. Sơ đồ trình tự - Bình luận

3.3.3. Đăng nhập



Hình 8. Sơ đồ trình tự - Đăng nhập

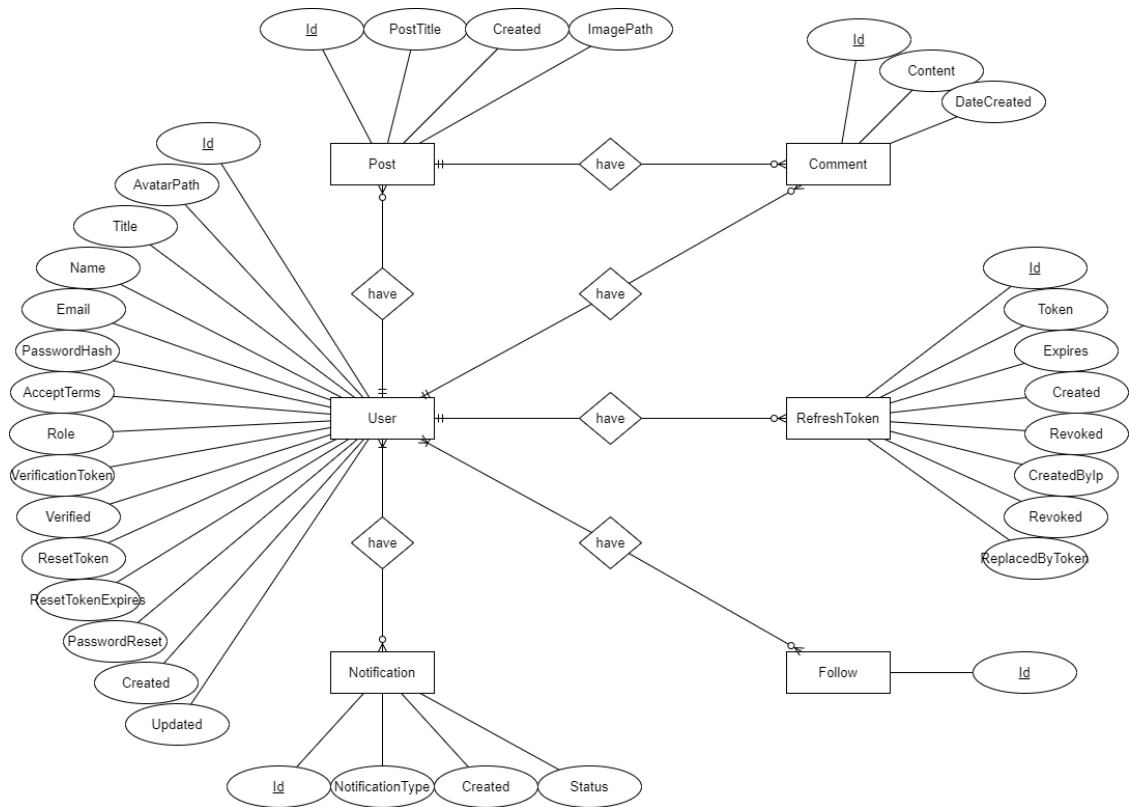
3.3.3. Tìm kiếm



Hình 9. Sơ đồ trình tự - Tìm kiếm

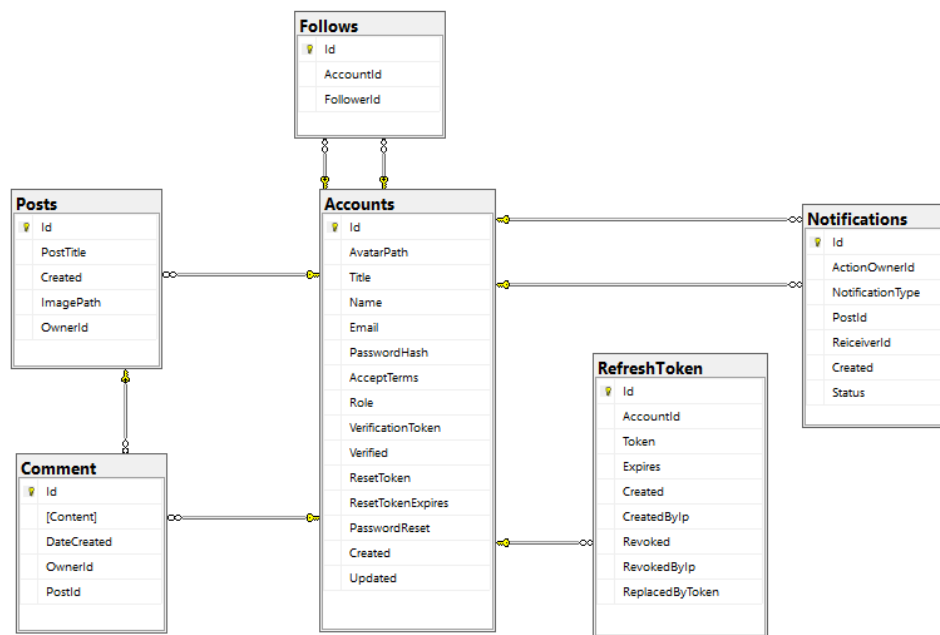
3.4. Thiết kế cơ sở dữ liệu

3.4.1. Mô hình liên kết thực thể



Hình 10. Mô hình liên kết thực thể.

3.4.2. Mô hình quan hệ các bảng trong cơ sở dữ liệu



Hình 11. Mô hình quan hệ.

3.5. Thiết kế giao diện









3.5.1. Giao diện “Người quản trị - Admin”

3.5.1.1. Danh sách người dùng

Accounts	Posts	Comments	Log out
----------	-------	----------	---------

Accounts

+ Create Account

Name	Email	Role	
Mr Hiep Le	admin@gmail.com	Admin	 
Mr Trang Ngô	trang@gmail.com	User	 
Miss Trang Ngô	user1@gmail.com	User	 
Mr Lê Hiệp	user2@gmail.com	User	 

Hình 12. Giao diện Danh sách người dùng

3.5.1.2. Thêm mới người dùng

Accounts	Posts	Comments	Log out
----------	-------	----------	---------

← Create Account

Title

Name

Email

Role

Password

Confirm Password

Save

Hình 13. Giao diện Thêm mới người dùng

3.5.1.3. Chỉnh sửa người dùng

Accounts	Posts	Comments	Log out
----------	-------	----------	---------

← Edit Account

Title

Name

Mr

Trang Ngô

Email

Role

trang@gmail.com

User

Change Password

Leave blank to keep the same password







Password

Confirm Password

Save

Hình 14. Giao diện Chỉnh sửa người dùng

3.5.1.4. Danh sách bài viết

Accounts	Posts	Comments	Log out
Posts			
+ Create Post			
Account Id	Post Title	Image	Created Date
2	Đẹp chịu hết nổi		18/12/2020 09:02:35 
2			18/12/2020 09:02:56 
2	Hello		18/12/2020 09:03:13 



Hình 15. Giao diện Danh sách bài viết

3.5.1.5. Tạo bài viết

Accounts	Posts	Comments	Log out
Create Post			
←			
Title			
<input type="text" value="Nhập tiêu đề"/>			
File			
<input type="button" value="Chọn tệp"/> Không có tệp nào được chọn			
<input type="button" value="Save"/>			

Hình 16. Giao diện Tạo bài viết

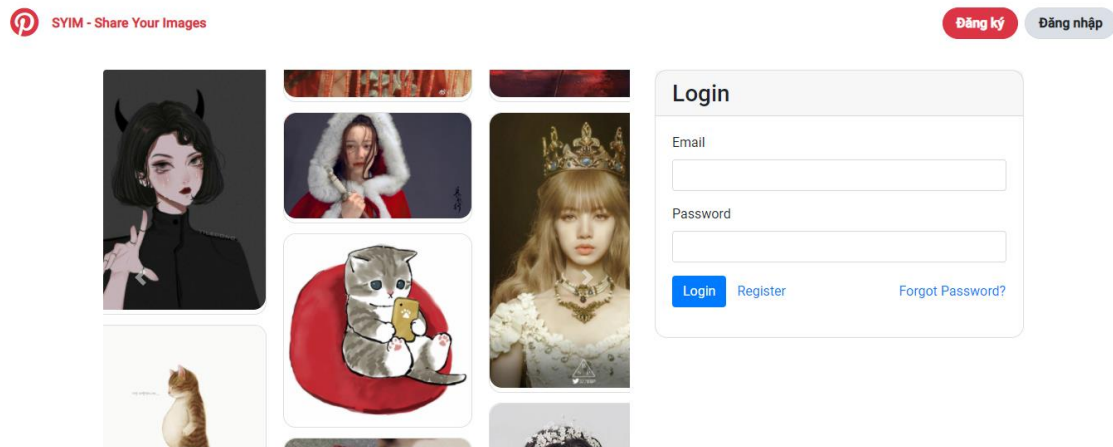
3.5.1.6. Danh sách bình luận

Accounts	Posts	Comments	Log out
Comments			
Post Id	Owner Id	Content	Created Date
3	2	xinh quá đi	18/12/2020 09:07:22 
3	1	cute	18/12/2020 09:39:51 

Hình 17. Giao diện Danh sách bình luận

3.5.2. Giao diện “Khách”

3.5.2.1. Đăng ký



SYIM - Share Your Images

Đăng ký Đăng nhập

Login

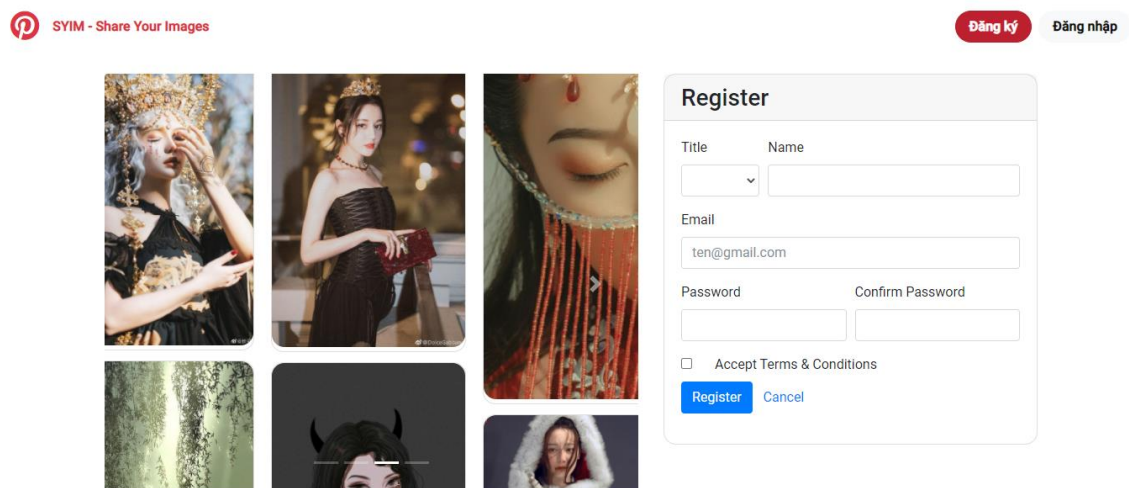
Email

Password

Login Register Forgot Password?

Hình 18. Giao diện Đăng nhập

3.5.2.2. Đăng nhập



SYIM - Share Your Images

Đăng ký Đăng nhập

Register

Title Name

Email

ten@gmail.com

Password Confirm Password

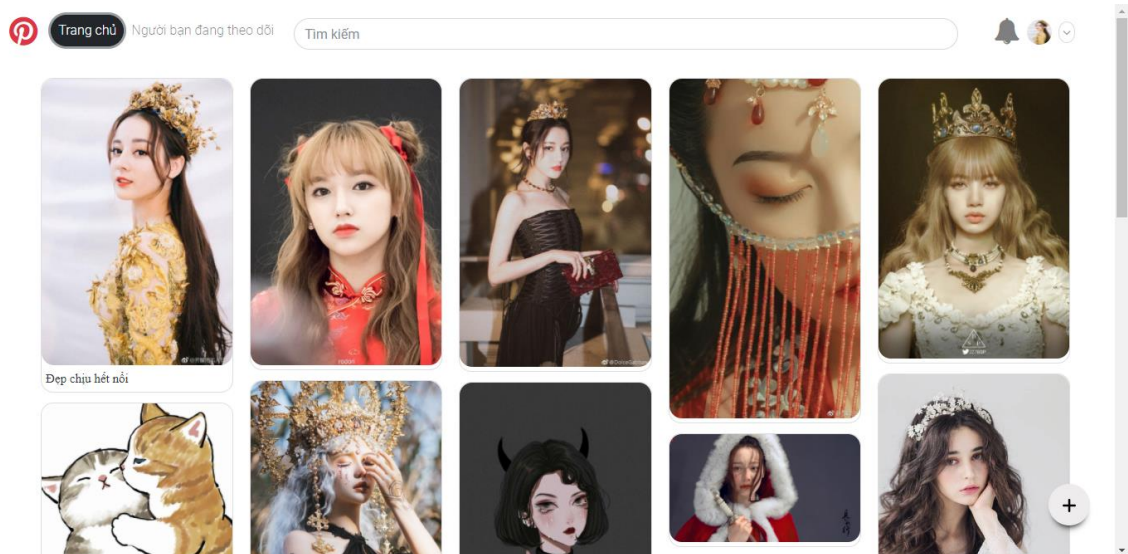
☐ Accept Terms & Conditions

Register Cancel

Hình 19. Giao diện Đăng ký

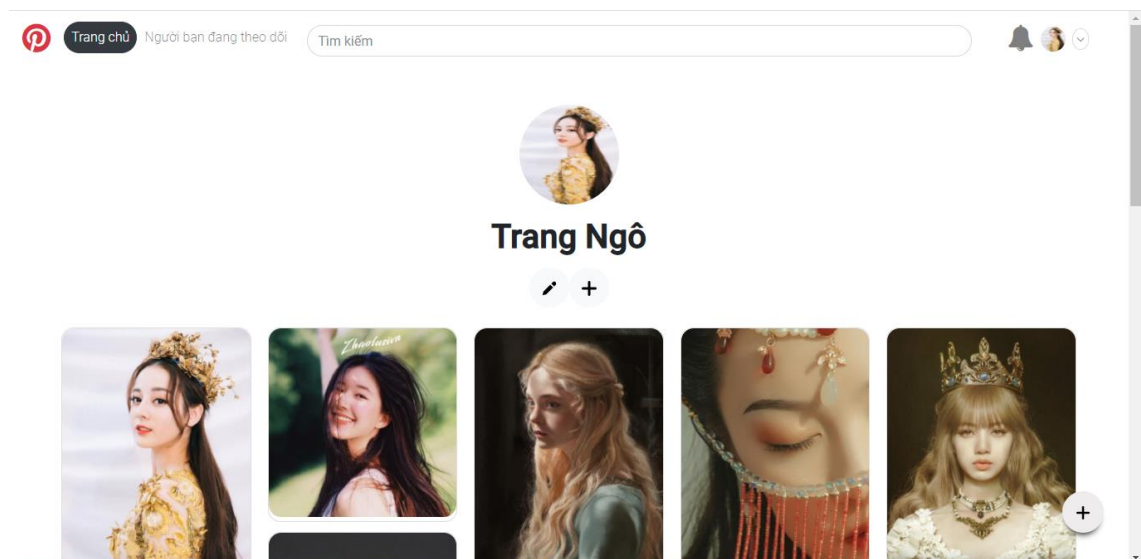
3.5.3. Giao diện “Người dùng đã xác thực”

3.5.3.1. Trang chủ



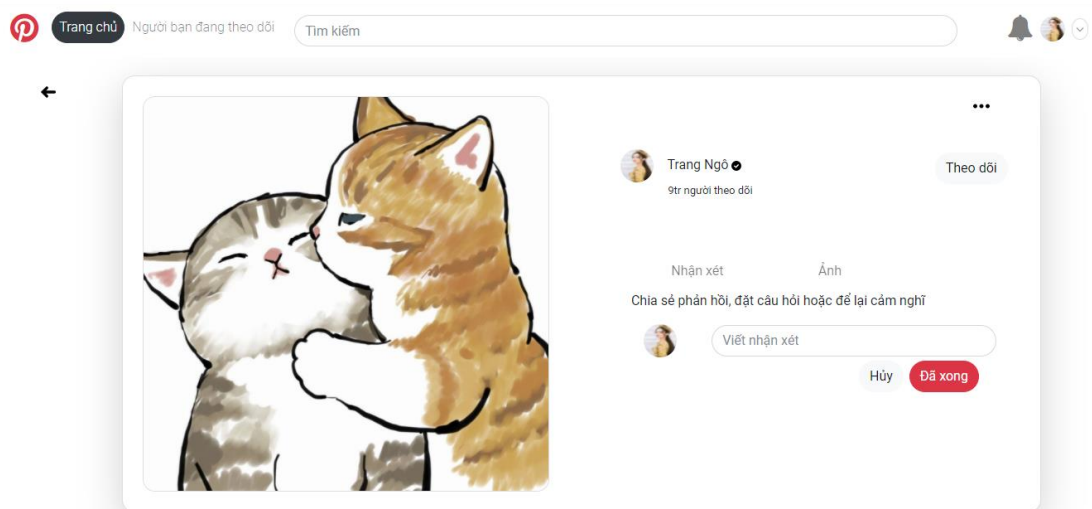
Hình 20. Giao diện Trang chủ

3.5.3.2. Trang cá nhân



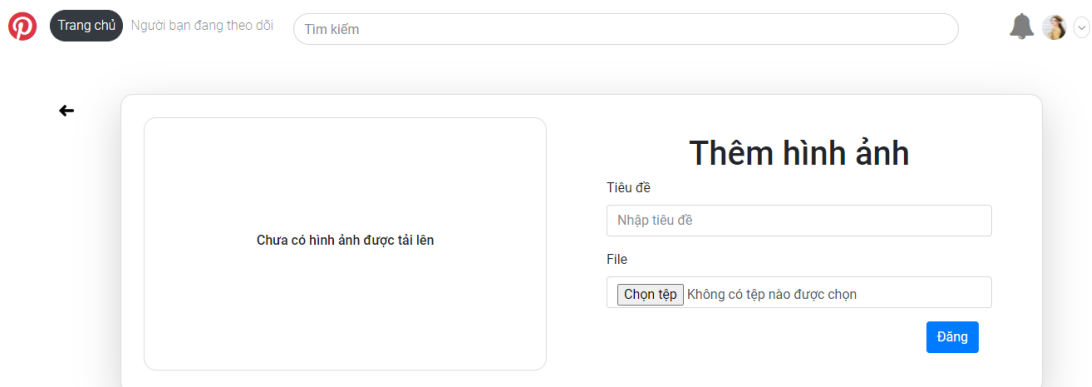
Hình 21. Giao diện Trang cá nhân

3.5.3.3. Chi tiết bài viết



Hình 22. Giao diện Chi tiết bài viết

3.5.3.4. Tạo bài viết



Hình 23. Giao diện Tạo bài viết

3.5.3.5. Chỉnh sửa hồ sơ

Trang chủ Người bạn đang theo dõi Tìm kiếm

Update Profile

Title Name

Mr Trang Ngô

Email

trang@gmail.com

Change Password

Leave blank to keep the same password


Password Confirm Password

Delete Account Update

Hình 24. Giao diện Chỉnh sửa hồ sơ

3.5.3. Giao diện Swagger UI

3.5.2.1. Tài khoản

Accounts 

POST	/Accounts/authenticate
POST	/Accounts/refresh-token
POST	/Accounts/revoke-token
POST	/Accounts/register
POST	/Accounts/verify-email
POST	/Accounts/forgot-password
POST	/Accounts/validate-reset-token
POST	/Accounts/reset-password
PUT	/Accounts/SetAvatar/{id}
GET	/Accounts
POST	/Accounts
GET	/Accounts/{id}
PUT	/Accounts/{id}
DELETE	/Accounts/{id}

Hình 25. Giao diện Swagger – Tài khoản

3.5.2.2. Bài viết

Posts		▼
GET	/Posts	
POST	/Posts	
GET	/Posts/GetPostById/{id}	
GET	/Posts/GetAllByUserId/{id}	
PUT	/Posts/{id}	

Hình 26. Giao diện Swagger – Bài viết

3.5.2.3. Bình luận

Comment		▼
GET	/Comment	
POST	/Comment	
GET	/Comment/GetAllByPostId/{id}	
PUT	/Comment/{id}	
DELETE	/Comment/{id}	

Hình 27. Giao diện Swagger – Bình luận

3.5.2.4. Thông báo

Notification		▼
GET	/Notification	
GET	/Notification/{id}	

Hình 28. Giao diện Swagger - Thông báo

3.5.2.5. Tìm kiếm

Search		▼
GET	/Search	

Hình 29. Giao diện Swagger - Tìm kiếm

3.5.2.6. Lược đồ



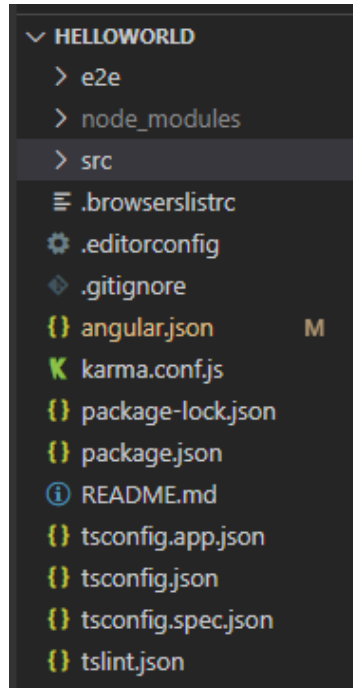
Hình 30. Giao diện Swagger - Lược đồ

Chương 4: CÀI ĐẶT SẢN PHẨM

4.1. Front-end

4.1.1. Cấu trúc ứng dụng

Sau khi tạo project thành công thì ở thư mục root có cấu trúc như hình dưới:



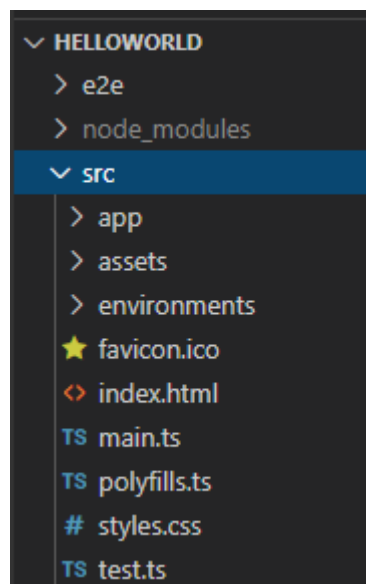
Hình 31. Cấu trúc tổng quát của Angular Project

Mục đích của các thư mục và tập tin này như sau:

- **e2e**: Thư mục này dùng để chứa các tập tin dành cho mục đích kiểm thử (testing).
- **node_modules**: Chứa các Node.js module cần thiết cho ứng dụng.
- **src**: Đây là thư mục sẽ chứa toàn bộ source code của ứng dụng.
- **.editorconfig**: Chứa các cấu hình liên quan đến phần Editor để chỉnh sửa source code như: indent_size, max_line_length,...
- **.gitignore**: Đây là tập tin metadata của Git, chứa thông tin những tập tin hoặc thư mục sẽ bị bỏ qua (ignore) không được commit lên Git Repository.
- **angular-cli.json**: Đây là tập tin chứa cấu hình cho Angular CLI, giúp chúng ta có thể build ứng dụng Angular.
- **karma.conf.js**: Tập tin cấu hình cho Karma, liên quan nhiều đến phần testing.

- **package-lock.json**: Dùng để lock version cho các Node.js module dependencies.
- **package.json**: Tập tin cấu hình cho Node.js module dependencies.
- **protractor.conf.js**: Tập tin cấu hình cho Protractor, liên quan nhiều đến phần testing.
- **README.md**: Tập tin này thường được sử dụng để cho các hệ thống Git hiển thị thông tin về Git Repository của chúng ta.
- **tsconfig.json**: Tập tin định nghĩa việc compile cho TypeScript.
- **tslint.json**: Tập tin cấu hình để kiểm tra lỗi cho các tập tin .ts (TypeScript) trong Angular project.

Trong các tập tin, thư mục được liệt kê ở trên thì thư mục src là thư mục quan trọng, là nơi để chúng ta có thể thêm code, modify code để phát triển ứng dụng Angular của mình. Nội dung của thư mục này như sau:



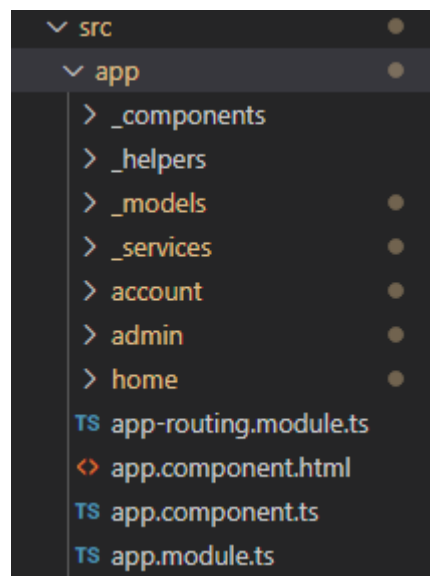
Hình 32. Cấu trúc thư mục src

Mục đích của các thư mục, tập tin trong thư mục này như sau:

- **app**: Đây là thư mục sẽ chứa toàn bộ code của ứng dụng Angular.
- **assets**: Thư mục này sẽ chứa các file ảnh, CSS, custom JavaScript của ứng dụng Angular.

- **environments:** Chúng ta có thể viết ứng dụng chạy trên nhiều môi trường khác nhau, đây chính là thư mục giúp chúng ta làm định nghĩa các tập tin cấu hình cho những môi trường khác nhau đó.
- **favicon.ico:** Icon của ứng dụng Angular.
- **index.html:** Trang chủ của ứng dụng Angular.
- **main.ts:** Chứa code bootstrapping cho ứng dụng Angular.
- **polyfill.ts:** Dùng để định nghĩa các chuẩn để ứng dụng của chúng ta có thể chạy được trên mọi trình duyệt.
- **style.css:** Định nghĩa style CSS cho ứng dụng Angular.
- **test.ts:** Code để chạy test.

Như đã nói ở trên thì app là thư mục chứa toàn bộ code của ứng dụng và khi tạo project trong thư mục này có các tập tin thể hiện mẫu cho 1 module. Thư mục app có cấu trúc như sau:



Hình 33. Cấu trúc thư mục app

- **_components:** Dùng để chứa các component dùng chung cho toàn ứng dụng như alert.
- **_helpers:** Nơi để các file TypeScript phục vụ cho việc kiểm tra, xác thực.
- **_models:** Chứa các file khai báo model tương ứng với database.
- **account:** Là module với các component phục vụ cho bước đăng ký đăng nhập, còn có thể gọi là giao diện cho Khách.

- **admin:** Là module với các component cho Người quản trị.
- **home:** Là module với các component cho Người dùng đã xác thực.
- **app-routing.module.ts:** Dùng để định nghĩa routing cho module.
- **app-module.ts:** Dùng để nhóm các component, directive, pipe, và service có trong module
- **app-component.html:** Dùng để viết giao diện cho component bằng ngôn ngữ html.
- **app-component.ts:** Dùng để khai báo cho component.

4.1.2. Các đoạn xử lý chính trong front-end

Xử lý khi upload hình ảnh:

```
<input
  FormControlName="file"
  id="file"
  type="file"
  class="form-control"
  (change)="onFileChange($event)"
  style="word-wrap:break-word;">
```

Hình 34. HTML của upload file

Input sẽ nhận đầu vào với dạng file, sau khi người dùng click vào và chọn tải lên hình ảnh, giao diện sẽ có sự thay đổi tương ứng với thiết lập trong onFileChange(\$event).

```
ngOnInit(): void {
  this.myForm = this.formBuilder.group({
    postTitle: ['', Validators.required],
    file: ['', Validators.required],
    fileSource: ['', Validators.required]
  });
  this.testForm = new FormData();
}
```

Hình 35. Xử lý dữ liệu

Vì form sẽ có dữ liệu dạng JSON nên phải chuyển sang FormData() để lấy dữ liệu nhận được.

```

onFileChange(event) {
  const reader = new FileReader();
  if (event.target.files && event.target.files.length) {
    const [file] = event.target.files;
    console.log('1');

    this.testForm.append("file",file);
    this.testForm.append("postTitle", this.myForm.get("postTitle").value);
    reader.readAsDataURL(file);
    reader.onload = () => {
      this.imageSrc = reader.result as string;
      console.log(this.myForm.value);
    };
    console.log(this.myForm.get('postTitle').value);
  }
}

```

Hình 36. Xử lý giao diện khi upload.

Khi file được upload thành công sẽ lấy thông tin thêm vào FormData tương ứng với title “file”. Sau khi lấy được thông tin thì dùng `FileReader()` để đọc source của hình. Và phía giao diện ta chỉ cần thêm `imageSrc` vào `[src]` để hiện hình ảnh vừa được upload.

```

<div *ngIf="imageSrc" class="col-6 text-center" style="outline-style: none; ">
  <div class="card mt-4 mb-4 ml-4 mr-4" style="justify-content: center; align-items: center; display: flex;">
    <img style="height: 100%; width: 100%;" [src]="imageSrc" alt="">
  </div>
</div>

```

Hình 37. HTML cho hiện hình ảnh sau khi upload

4.2. Back-end

4.2.1. Cấu trúc project API

Project được chia thành các thư mục sau:

- **Controllers**

Xác định các end points / routes cho web api, controllers là đầu vào cho web api từ client applications thông qua http requests.

- **Models**

Gồm các Request Model và Response Model cho các chức năng của Controller, request model khai báo các tham số cho các yêu cầu của các request được gửi đến, và response models sẽ khai báo cấu trúc của dữ liệu được trả về.

- **Services**

Chứa code cho business logic chính, xác thực và truy cập dữ liệu.

- **Entities**

Đại diện cho dữ liệu được lưu trữ dưới database. Entity Framework Core sẽ ánh xạ dữ liệu quan hệ từ cơ sở dữ liệu đến các phiên bản của đối tượng thực thể C#.

- **Migrations**

Database migration files dựa trên các lớp Entities classes được dùng để tự động tạo cơ sở dữ liệu MSSQLserver cho api và tự động cập nhật mỗi khi Entities có thay đổi. Migrations được tạo ra với Package Managment Console, the migrations in this example were generated with the command Add-Migration InitialCreate.

- **Middleware**

Các class Middleware được dùng để tạo các yêu cầu pipeline để xử lý các tác vụ như xử lý global error và xác thực token.

- **Helpers**

Một số class hỗ trợ cho việc ánh xạ dữ liệu, xác thực,...

4.2.2. Một số Function

Dùng AutoMapper để ánh xạ Entity sang Model, nhờ đó ta có thể tự do xử lý dữ liệu mà không ảnh hưởng đến cấu trúc của Entity cũng như quan hệ giữa chúng. Model phản ánh cấu trúc dữ liệu trả về phía client không nhất thiết phải hoàn toàn giống Entity, mà có thể có thuộc tính của nhiều Entity khác nhau.

```
public AutoMapperProfile()
{
    //Reaction
    CreateMap<Reaction, AccountResponse>();

    CreateMap<CreateReactionRequest, Reaction>();

    CreateMap<UpdateReactionRequest, Reaction>();
    //Notification
    CreateMap<Notification, NotificationResponse>();

    CreateMap<CreateNotificationRequest, Notification>();

    CreateMap<UpdateNotificationRequest, Notification>();
    //Post
    CreateMap<Post, PostResponse>();

    CreateMap<CreatePostRequest, Post>();

    CreateMap<UpdatePostRequest, Post>();
}
```

Hình 38. AutoMapperProfile.cs

```

public class Post
{
    public int Id { get; set; }
    public string PostTitle { get; set; }
    public DateTime Created { get; set; }
    public string ImagePath { get; set; }
    public int OwnerId {get; set;}
}

```

Hình 39. Post.cs

```

public class PostResponse
{
    public int Id { get; set; }
    public string PostTitle { get; set; }
    public DateTime Created { get; set; }
    public string ImagePath { get; set; }
    public int OwnerId { get; set; }
    public string OwnerName { get; set; }
    public string OwnerAvatar { get; set; }
}

```

Hình 40. PostReponse.cs

Sử dụng các Class service để xử lý dữ liệu trước khi gửi về client thông qua HttpRequest. Nhờ vào việc map các thuộc tính của entity vào các model, ta có thể đơn giản hoá dữ liệu trả về nhưng vẫn đảm bảo đầy đủ thông tin cần thiết cho mỗi request.

```

[HttpPost, DisableRequestSizeLimit]
public IActionResult CreatePost([FromForm] CreatePostRequest post)
{
    try
    {
        var file = Request.Form.Files[0];
        var folderName = Path.Combine("Resources", "Images");
        var pathToSave = Path.Combine(Directory.GetCurrentDirectory(), folderName);
        if (file.Length > 0)
        {
            var fileName = ContentDispositionHeaderValue.Parse(file.ContentDisposition).FileName.Trim('"');
            var fullPath = Path.Combine(pathToSave, fileName);
            var dbPath = Path.Combine(folderName, fileName);

            using (var stream = new FileStream(fullPath, FileMode.Create))
            {
                var model = new CreatePostRequest
                {
                    ImagePath = dbPath
                };
                var temp = _postService.CreatePost(model);
                return Ok(new { dbPath });
            }
        }
        else
        {
            return BadRequest();
        }
    }
    catch (Exception ex)
    {
        return StatusCode(500, $"Internal server error: {ex}");
    }
}

```

Hình 41. PostController.cs

Sử dụng các service khác nhau để hỗ trợ việc lấy các thuộc tính của các entity khác nhau sau đó map vào Response Model.

```
public class PostService : IPostService
{
    private readonly DataContext _context;
    private readonly IMapper _mapper;
    private readonly INotificationService _notificationService;
    private readonly IFollowService _followService;
    public PostService(
        DataContext context,
        IMapper mapper,
        INotificationService notificationService,
        IFollowService followService)
    {
        _context = context;
        _mapper = mapper;
        _notificationService = notificationService;
        _followService = followService;
    }
    //Get specific post by its Id
    public PostResponse GetPostById(int postId) ...

    //Create
    public PostResponse CreatePost(CreatePostRequest model)
    {
        var post = _mapper.Map<Post>(model);
        if (post == null) throw new ApplicationException("Create Post failed");
        _context.Posts.Add(post);
        _context.SaveChanges();
        SendNotification(post);
        return _mapper.Map<PostResponse>(post);
    }
}
```

Hình 42. PostService.cs

Ngoài ra ta còn có thể dùng các service để xử lý dữ liệu một cách tự động mỗi khi cần mà không cần tạo request mới.

```

private void SendNotification(Post post)
{
    var followers = _followService.GetAllByUserId(post.OwnerId);
    //Create notification for each follower of Post owner
    foreach (FollowResponse follower in followers)
    {
        var notification = new CreateNotificationRequest
        {
            //Post Owner
            ActionOwnerId = post.OwnerId,
            NotificationType = NotificationType.Posted,
            PostId = post.Id,
            ReiceiverId = follower.Id,
            Created = DateTime.Now,
            Status = Status.Created
        };
        _notificationService.CreateNotification(notification);
    }
}

```

Hình 43. PostService.cs

Các Request model giúp chúng ta định dạng dữ liệu truyền vào cho các hàm xử lý của service nhằm đảm bảo tính đúng đắn của dữ liệu, giảm tối đa việc xuất hiện exception.

```

public class CreateNotificationRequest
{
    [Required]
    public int Id { get; set; }
    [Required]
    public int ActionOwnerId { get; set; }
    [Required]
    public NotificationType NotificationType { get; set; }
    [Required]
    public int? PostId { get; set; }
    [Required]
    public int ReiceiverId { get; set; }
    [Required]
    public DateTime Created { get; set; }
    [Required]
    public Status Status { get; set; }
}

```

Hình 44. CreateNotificationRequest.cs

Chương 5: TỔNG KẾT

5.1. Kết quả đạt được

Nhóm đã xây dựng được một trang website chia sẻ hình ảnh với các chức năng cơ bản mà nhóm đã đề ra lúc đầu như:

- Đăng ký, đăng nhập: bằng email, facebook hoặc google.
- Đăng xuất.
- Tìm kiếm hình ảnh.
- Xem thông báo.
- Chỉnh sửa hồ sơ
- Tương tác với bài viết:

Xem chi tiết bài viết.

Chỉnh sửa bài viết (khi là của chính người dùng).

Tải xuống hình ảnh.

Bình luận cho bài viết.

Trả lời bình luận.

Chỉnh sửa bình luận (bình luận của chính người dùng).

- Tương tác với trang người dùng khác:

Theo dõi.

Xem tất cả bộ sưu tập và hình ảnh trong đó của người dùng.

Xem tất cả người theo dõi và đang theo dõi của người dùng.

- Tạo bài viết:

Tạo tiêu đề.

Tải lên hình ảnh.

5.2. Ưu điểm

Giao diện thân thiện dễ sử dụng.

5.3. Nhược điểm

Giao diện chưa hoàn thiện.

Chưa có độ bảo mật cao.

Còn thiếu các chức năng thương mại.

5.4. Khó khăn

Không có nhiều thời gian để website trở nên hoàn hảo hơn.

Không đủ kiến thức và kỹ năng dẫn đến hoàn thành công việc trễ hơn dự kiến.

5.5. Bài học kinh nghiệm

Nên phân chia thời gian thực hiện công việc tốt hơn.

5.6. Hướng phát triển

Trong tương lai gần nhóm sẽ phát triển:

- Hoàn thiện các chức năng đã có.
- Thêm phân tích người dùng, đề xuất tìm kiếm.
- Dùng AI để phân loại ảnh.
- Thêm chatbot để phục vụ cho chức năng nhắn tin.
- Thêm một số chức năng mang tính thương mại.
- Thêm các cơ chế bảo mật.

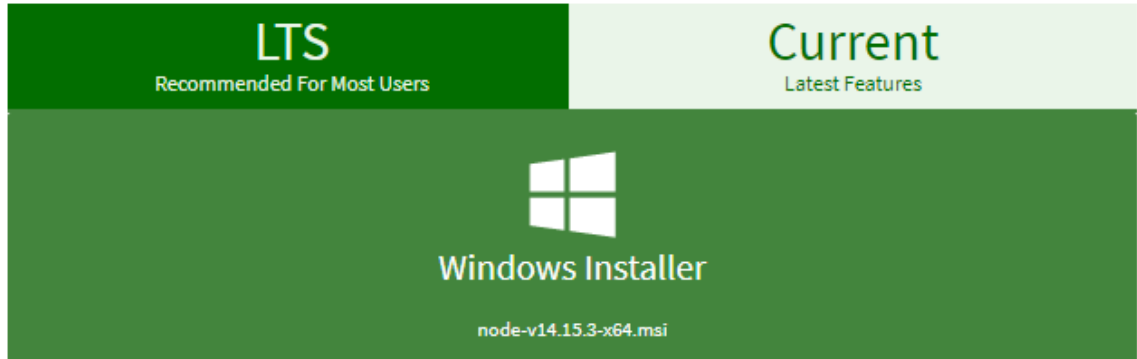
TÀI LIỆU THAM KHẢO

1. Angular là gì? Giới thiệu toàn tập về Angular. Truy cập tại: https://blog.tinohost.com/angular-la-gi/#Angular_la_gi
2. ANGULAR LÀ GÌ? TẠI SAO NÊN SỬ DỤNG ANGULAR? Truy cập tại: <https://tuyendung.ominext.com/angular-la-gi-tai-sao-can-hoc-lap-trinh-angular/>
3. Tổng quan về ASP.NET Core. Truy cập tại: <https://netcore.vn/bai-viet/tong-quan-ve-aspnet-core>
4. Microsoft SQL Server. Truy cập tại: https://vi.wikipedia.org/wiki/Microsoft_SQL_Server
5. SQL Server 2017 có gì mới? So sánh các ấn bản, phiên bản SQL 2017. Truy cập tại: <https://help.pacisoft.com/knowledgebase/sql-server-2017-co-gi-moi-so-sanh-cac-phien-ban-sql-server/>
6. Tìm hiểu về Swagger - Công cụ viết document cho RESTfull APIs. Truy cập tại: <https://medium.com/velacorpblog/>
7. Giới thiệu tool Swagger UI - Viết document cho API. Truy cập tại: <https://viblo.asia/p/gioi-thieu-tool-swagger-ui-viet-document-cho-api-Qbq5QWRwZD8>
8. Cài đặt, tạo dự án, dịch và chạy ứng dụng ASP.NET Core. Truy cập tại: <https://tuhocict.com/cai-dat-moi-truong-va-cong-cu-cho-aspnet-core/>
9. Tổng quan về cấu trúc của Angular project. Truy cập tại: <https://huongdanjava.com/vi/tong-quan-ve-cau-truc-cua-angular-project.html>
10. Cài Đặt và Tạo Project Angular Đầu Tiên. Truy cập tại: <https://www.stdio.vn/javascript/cai-dat-va-tao-project-angular-dau-tien-FlmLz>

PHỤ LỤC

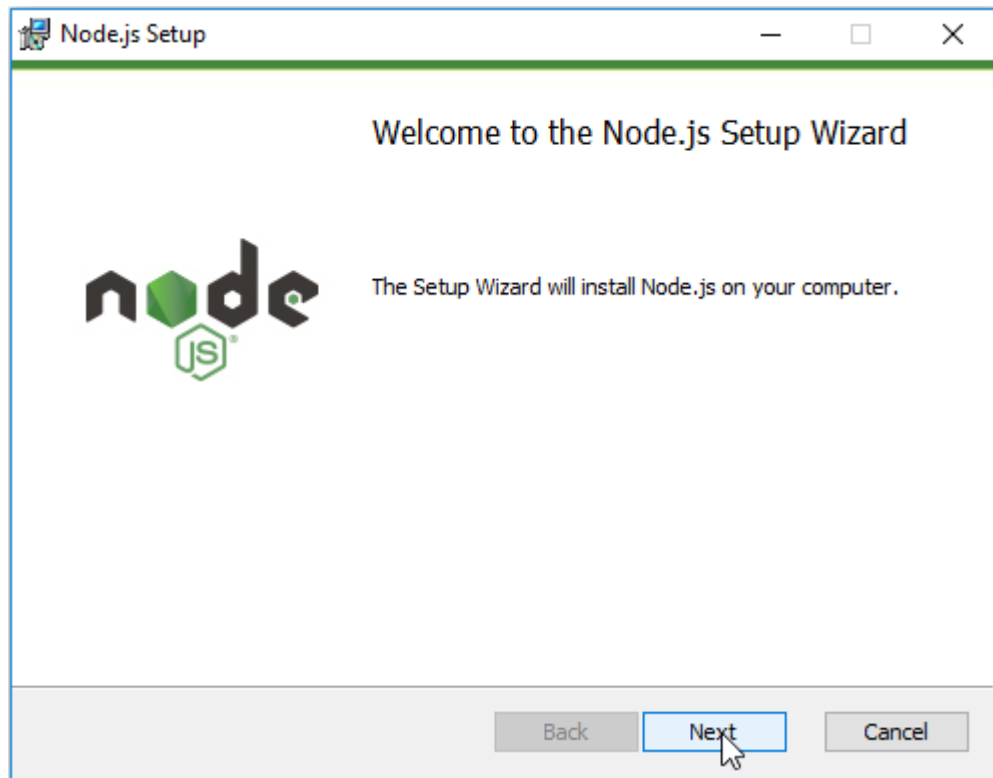
1. Cài đặt Node.js và npm

Tải Node.js tại: <https://nodejs.org/en/download/>

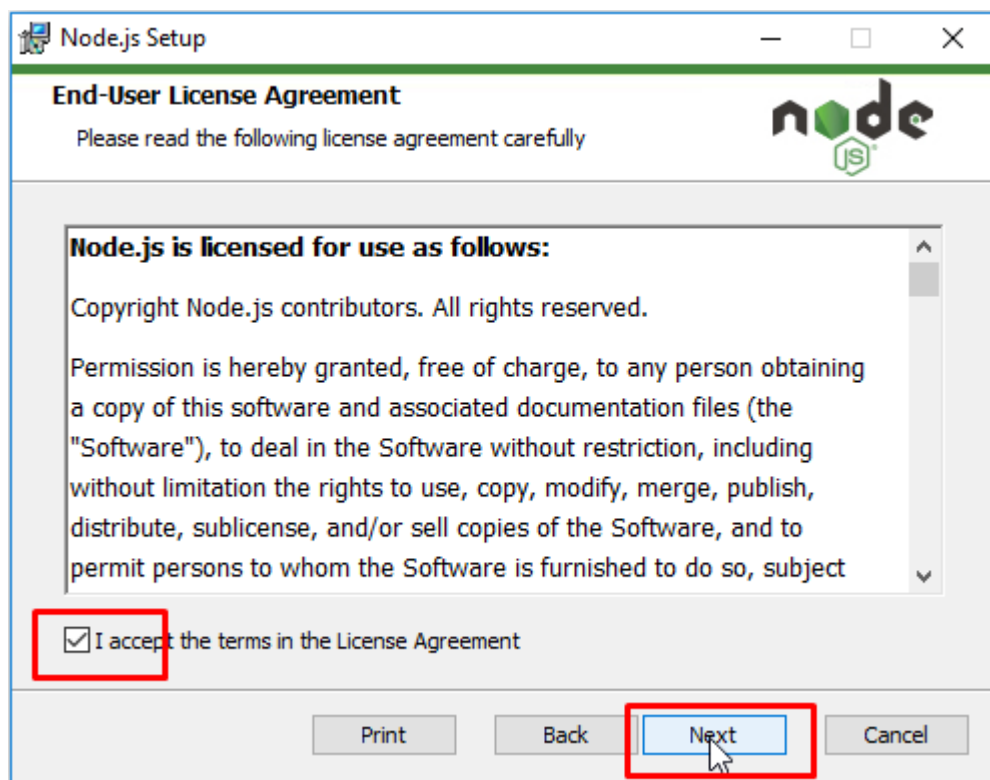


Hình 45. Tải Node.js

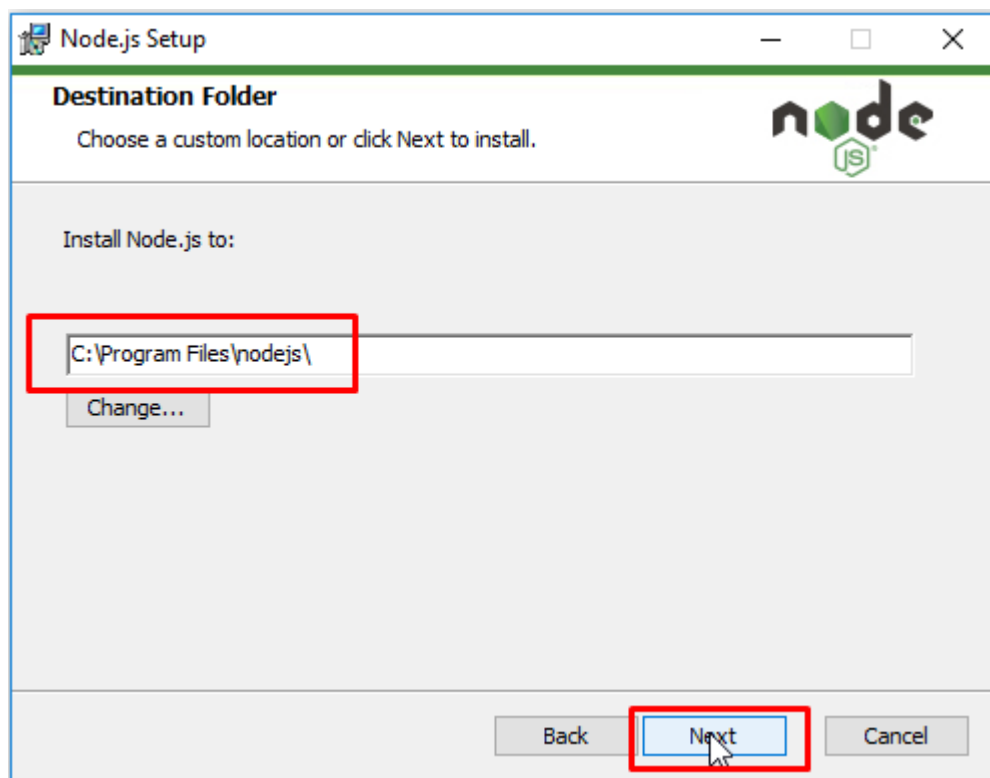
Tải hoàn tất và tiến hành chạy file cài đặt như hình dưới:



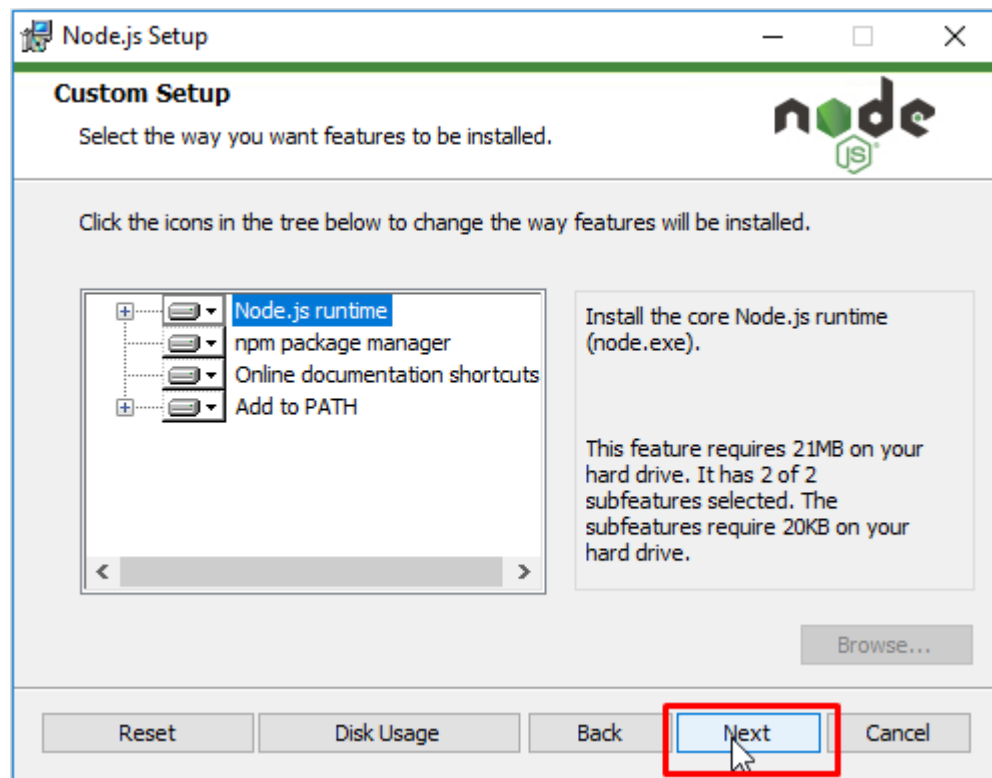
Hình 46. Cài đặt Node.js



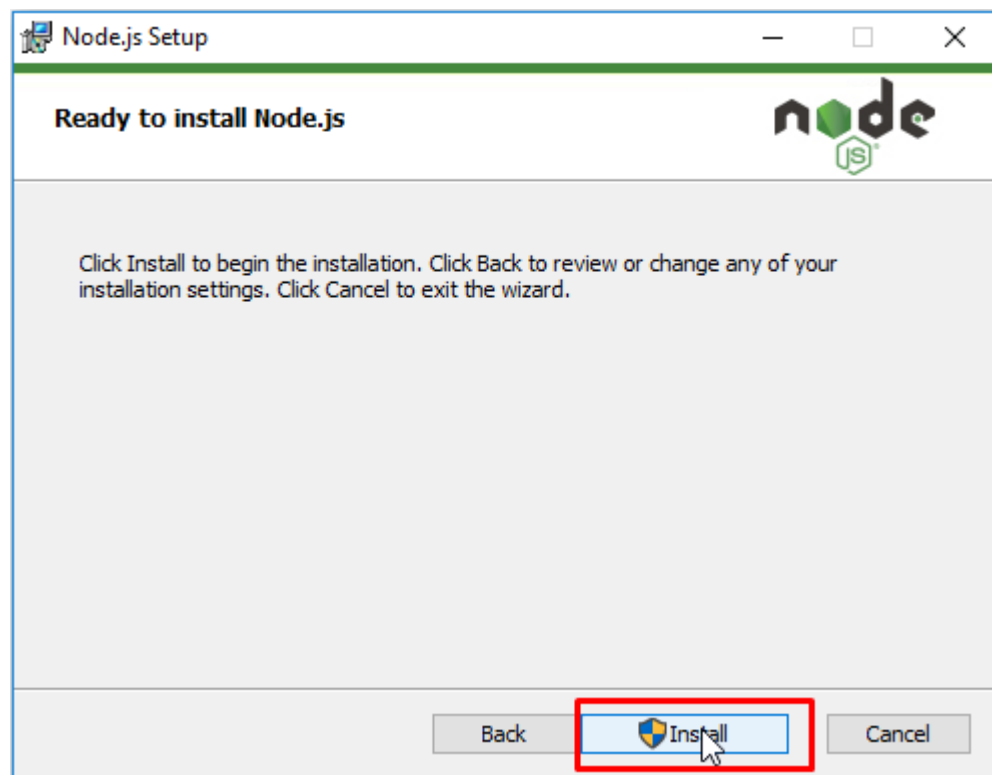
Hình 47. Cài đặt Node.js



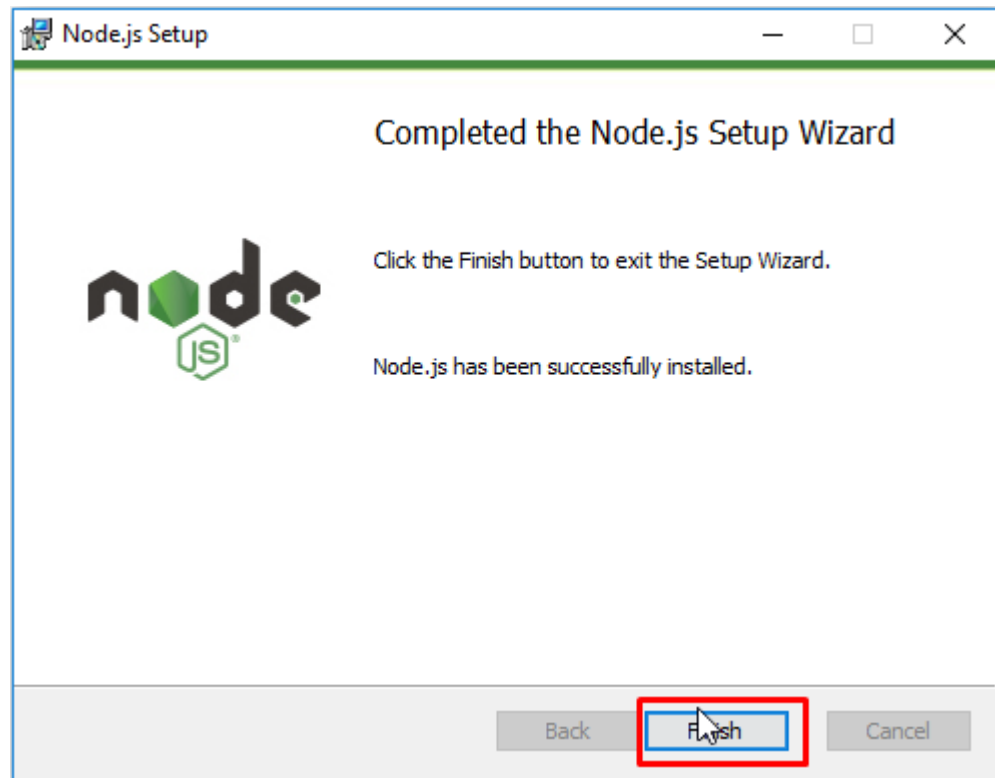
Hình 48. Cài đặt Node.js



Hình 49. Cài đặt Node.js



Hình 50. Cài đặt Node.js



Hình 51. Cài đặt Node.js

2. Cài đặt Visual Studio Code

Cài đặt Visual Studio Code chủ yếu yêu cầu:

- Máy tính phải được cài đặt sẵn **.NET Framework 4.5.2**.

Có thể kiểm tra bằng cách vào Control Panel => Programs and Features.

Nếu chưa có thì tải tại <https://www.microsoft.com/en-us/download/>

- Visual Studio Code bao gồm phiên bản User và System, có thể lựa chọn phiên bản nào cũng được. Phiên bản User phù hợp cho các user trong Windows không có quyền Administrator.

Tải Visual Studio Code tại <https://code.visualstudio.com>

Sau khi tải hoàn tất, chạy file cài đặt VSCodeUserSetup-...exe và tiến hành cài đặt, để mặc định Visual Studio Code được cài đặt với các cấu hình như đường dẫn.

3. Cài đặt Angular CLI

Angular CLI thực chất là một Node.js package, cài đặt thông qua trình quản lý package của Node.js - npm. Mở cửa sổ console và gõ vào như sau:

```
npm install -g @angular/cli
```

```
Select Command Prompt
Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Admin>npm install -g @angular/cli
npm WARN deprecated debug@4.2.0: Debug versions >=3.2.0 <3.2.7 || >=4 <4.3.1 have a low-severity ReDos regression when used in a Node.js environment. It is recommended you upgrade to 3.2.7 or 4.3.1. (https://github.com/visionmedia/debug/issues/797)
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
C:\Users\Admin\AppData\Roaming\npm\ng -> C:\Users\Admin\AppData\Roaming\npm\node_modules\@angular\cli\bin\ng

> @angular/cli@11.0.5 postinstall C:\Users\Admin\AppData\Roaming\npm\node_modules\@angular\cli
> node ./bin/postinstall/script.js

+ @angular/cli@11.0.5
added 1 package, removed 26 packages and updated 22 packages in 15.496s

New patch version of npm available! 6.14.6 -> 6.14.10
Changelog: https://github.com/npm/cli/releases/tag/v6.14.10
Run npm install -g npm to update!
```

Hình 52, Cài đặt Angular

Kiểm tra cài đặt thành công và số phiên bản bằng cách nhập: `ng version`

```
Command Prompt
Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Admin>ng version

Angular CLI
Angular CLI: 11.0.5
Node: 12.18.3
OS: win32 x64

Angular:
...
```

Hình 53. Kết quả cài đặt Angular

3. Tạo một ứng dụng Web Angular

Mở cửa sổ console và tạo project với cú pháp: `ng new <project-name>`

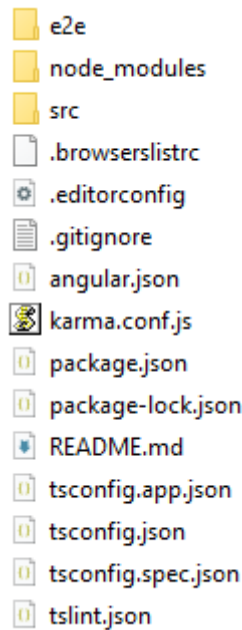
Ví dụ: `ng new HelloWorld`

```
Select Command Prompt
Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Admin>ng new HelloWorld
? Do you want to enforce stricter type checking and stricter bundle budgets in the workspace?
  This setting helps improve maintainability and catch bugs ahead of time.
  For more information, see https://angular.io/strict Yes
? Would you like to add Angular routing? Yes
? Which stylesheet format would you like to use? CSS
CREATE HelloWorld/angular.json (3655 bytes)
CREATE HelloWorld/package.json (1201 bytes)
CREATE HelloWorld/README.md (1019 bytes)
CREATE HelloWorld/tsconfig.json (737 bytes)
CREATE HelloWorld/tslint.json (3185 bytes)
CREATE HelloWorld/.editorconfig (274 bytes)
CREATE HelloWorld/.gitignore (631 bytes)
```

Hình 54. Tạo project Angular

Quá trình tạo mất khoảng từ 3 - 5 phút tùy thuộc tốc độ mạng do phải tải các gói phụ thuộc. Cấu trúc của project sau khi được tạo thành công như sau:



Hình 55. Cấu trúc project Angular

Nhờ Node.js, project có thể được host và chạy thử với lệnh:

```
C:\Users\Admin\HelloWorld>ng serve
? Would you like to share anonymous usage data about this project with the Angular Team at
Google under Google's Privacy Policy at https://policies.google.com/privacy? For more
details and how to change this setting, see http://angular.io/analytics. Yes

Thank you for sharing anonymous usage data. Would you change your mind, the following
command will disable this feature entirely:

  ng analytics project off

Compiling @angular/core : es2015 as esm2015
Compiling @angular/common : es2015 as esm2015
Compiling @angular/platform-browser : es2015 as esm2015
Compiling @angular/router : es2015 as esm2015
Compiling @angular/platform-browser-dynamic : es2015 as esm2015
✓ Browser application bundle generation complete.

Initial Chunk Files | Names | Size
vendor.js | vendor | 2.66 MB
polyfills.js | polyfills | 484.58 kB
styles.css, styles.js | styles | 344.14 kB
main.js | main | 59.52 kB
runtime.js | runtime | 6.15 kB
| Initial Total | 3.53 MB

Build at: 2020-12-20T04:56:47.468Z - Hash: 22ddf95e8a0b5641ad65 - Time: 11694ms

** Angular Live Development Server is listening on localhost:4200, open your browser on http://localhost:4200/ **

✓ Compiled successfully.
✓ Browser application bundle generation complete.

Initial Chunk Files | Names | Size
styles.css, styles.js | styles | 344.14 kB

4 unchanged chunks

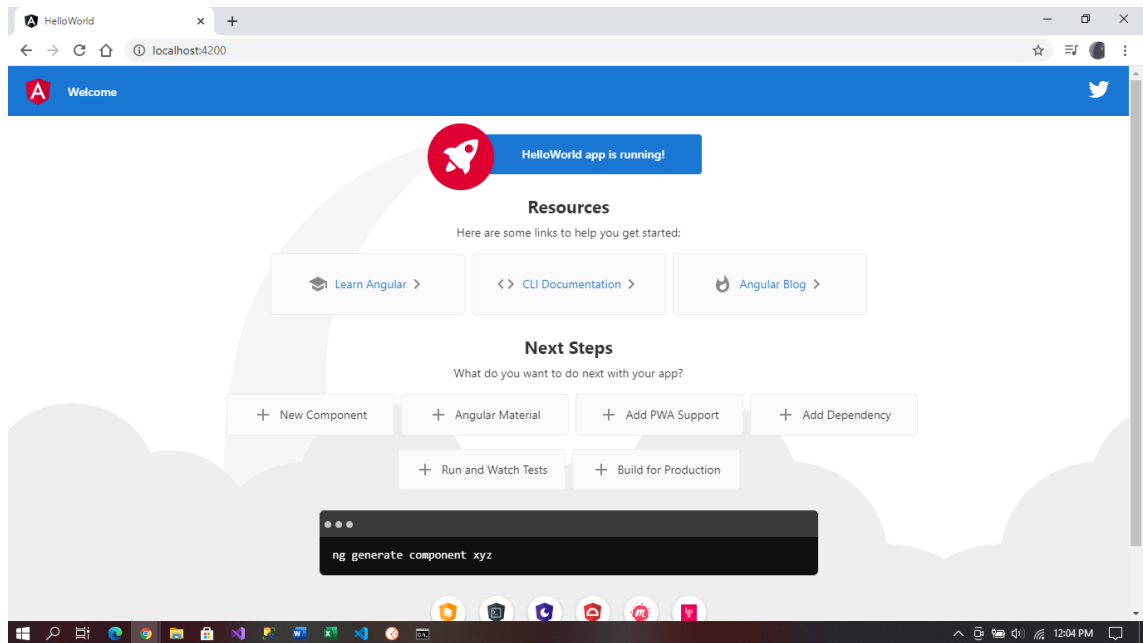
Build at: 2020-12-20T04:56:48.802Z - Hash: b94fe622f28f17ee8b04 - Time: 833ms
✓ Compiled successfully.
```

Hình 56. Chạy project Angular

Mặc định project sẽ được chạy trên địa chỉ `http://localhost:4200`. Có thể thay đổi port mặc định nhờ tham số `--port`.

Ví dụ muốn đổi về port 1234 gõ: `ng serve --port 1234`

Kết quả thực thi với port 4200 như sau:

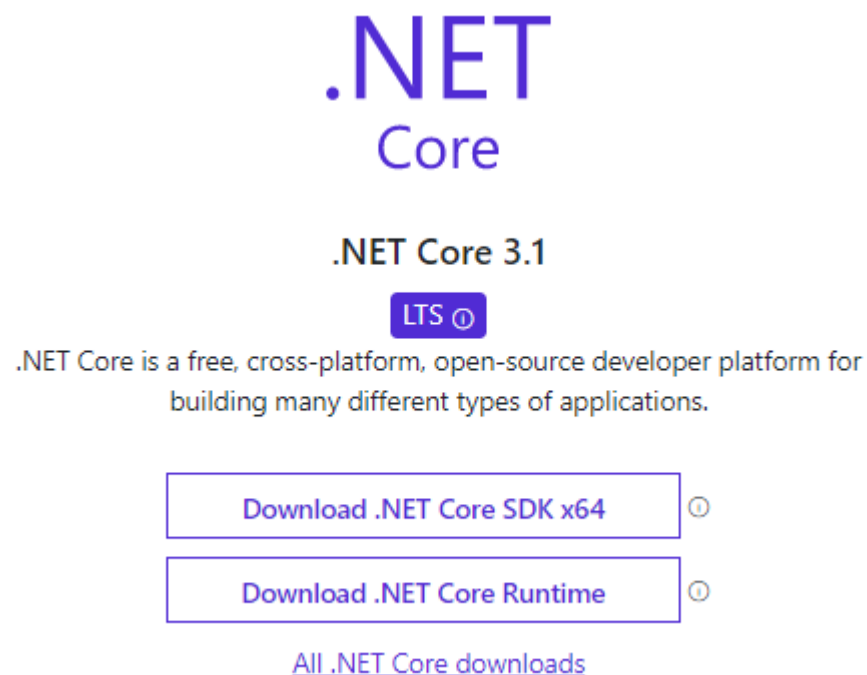


Hình 57 Giao diện khi tạo project trong Angular

4. Cài đặt .NET Core SDK

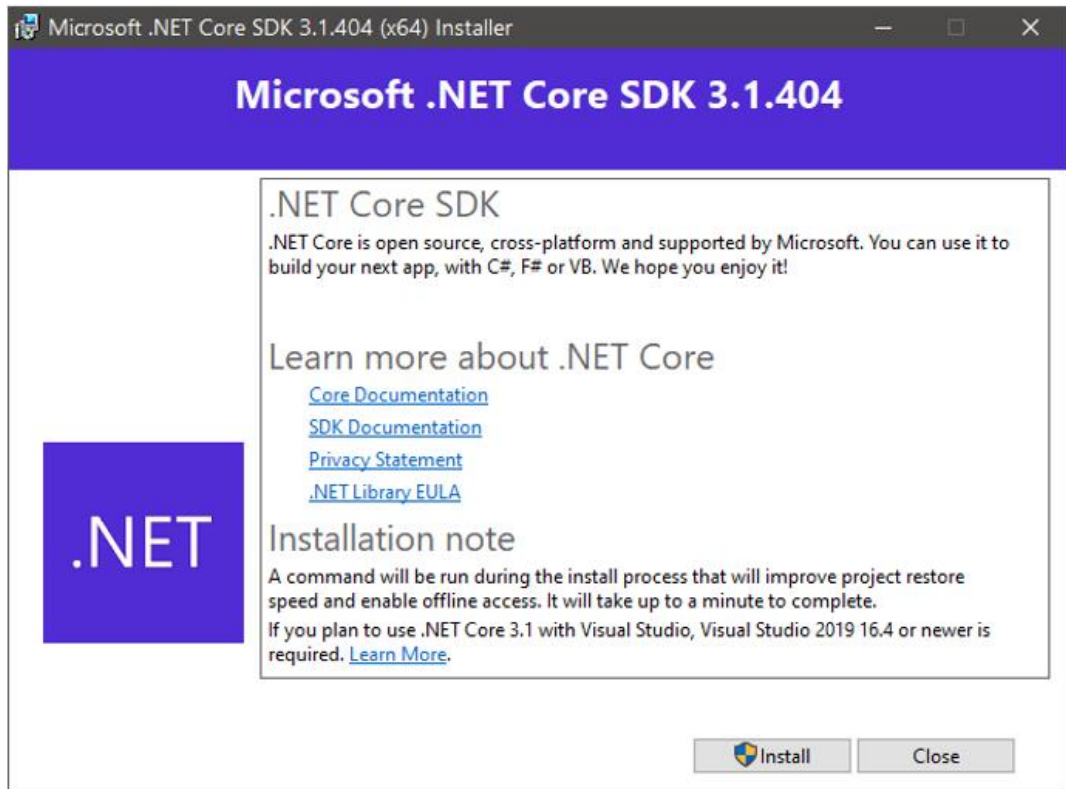
Tải .NET Core 3.1 từ <https://www.microsoft.com/net/download/core>

Để phát triển ứng dụng cần cài đặt ASP.NET Core SDK.



Hình 58. Tải .NET Core 3.1

Cài đặt bằng file installer đã tải về, giao diện cài đặt như hình dưới:



Hình 59. Cài đặt .NET Core SDK

5. Cài đặt Visual Studio Code

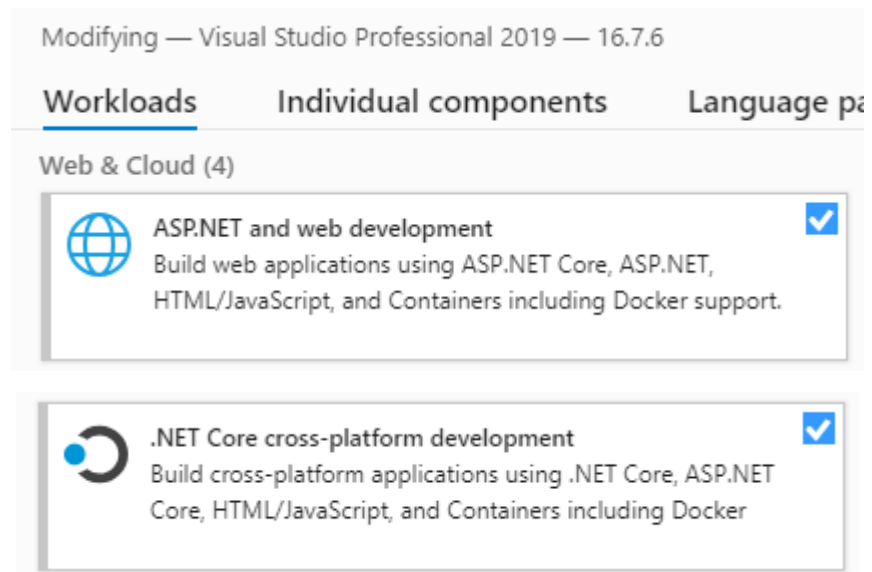
Do sử dụng .NET Core 3.1 (phiên bản mới nhất hiện nay) nên đòi hỏi phải sử dụng Visual Studio 2019 (v16.3 trở lên).

- Đang dùng bản Visual Studio cũ:

Cài đặt Visual Studio (community) 2019 tại

<https://visualstudio.microsoft.com/downloads/> .

Trong quá trình cài đặt, chọn Workloads **ASP.NET and web development** (phát triển ứng dụng trên cả ASP.NET và ASP.NET Core) hoặc **.NET Core cross-platform development** (phát triển ứng dụng trên .NET Core và ASP.NET Core).



Hình 60. Cài đặt Visual Studio 2019

- Đã cài đặt Visual Studio 2019:

Mở **Visual Studio Installer** update mới nhất rồi chọn **Modify** và chọn tab **Workloads** và cũng chọn một trong hai mục như trên. Cài đặt một trong hai workload trên sẽ giúp cài đặt tất cả các thành phần cần thiết cho phát triển và thực thi ứng dụng ASP.NET Core trên Windows.