ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC CNTT VÀ TT VIỆT – HÀN

**KHOA KHOA HỌC MÁY TÍNH**



**BÁO CÁO CÔNG NGHỆ VÀ LẬP TRÌNH WEB (7)**

**ĐỀ TÀI: XÂY DỰNG WEBSITE CHIA SẺ CÔNG NGHỆ LAPTOP**

**Sinh viên thực hiện:** Tôn Thất Hội – 20SE1

Hoàng Thị Kiều Giang – 20SE1

**Giảng viên hướng dẫn :** Ths. Lương Khánh Tý

*Đà Nẵng, tháng 10, năm 2021*

ĐẠI HỌC ĐÀ NẴNG

TRƯỜNG ĐẠI HỌC CNTT VÀ TT VIỆT – HÀN

**KHOA KHOA HỌC MÁY TÍNH**



**BÁO CÁO CÔNG NGHỆ VÀ LẬP TRÌNH WEB (7)**

**ĐỀ TÀI: XÂY DỰNG WEBSITE CHIA SẺ CÔNG NGHỆ LAPTOP**

**Sinh viên thực hiện:** Tôn Thất Hội – 20SE1

Hoàng Thị Kiều Giang – 20SE1

**Giảng viên hướng dẫn :** Ths. Lương Khánh Tý

*Đà Nẵng, tháng 10, năm 2021***NHẬN XÉT**

**(Của Giảng viên hướng dẫn)**

…………………………………………………………………………………………………………………………………………

…………………………………………………………………………………………………………………………………………

…………………………………………………………………………………………………………………………………………

…………………………………………………………………………………………………………………………………………

…………………………………………………………………………………………………………………………………………

…………………………………………………………………………………………………………………………………………

……………………………………………………………………

**Chữ ký của GVHD**

**LỜI CẢM ƠN**

Để dự án này đạt kết quả tốt đẹp, chúng em đã nhận được sự hỗ trợ, giúp đỡ của thầy cô. Với tình cảm sâu sắc, chân thành, cho phép chúng em được bày tỏ lòng biết ơn sâu sắc đến tất cả quý thầy cô đã tạo điều kiện giúp đỡ trong quá trình học tập và nghiên cứu và phát triển đề tài.

Trước hết chúng em xin gửi tới các thầy cô trường Đại học Công nghệ thông tin và truyền thông Việt – Hàn lời chào trân trọng, lời chúc sức khỏe và lời cảm ơn sâu sắc. Với sự quan tâm, dạy dỗ, chỉ bảo tận tình chu đáo của các giảng viên đã truyền tải các kĩ năng cần thiết, đến nay chúng em đã có thể hoàn thành đồ án cơ sở 2.

Đặc biệt chúng em xin gửi lời cảm ơn chân thành nhất tới cô giáo – Ths. Lương Khánh Tý đã quan tâm giúp đỡ tận tình chúng em hoàn thành tốt đồ án này trong thời gian qua.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế, đồ án này không thể tránh được những thiếu sót. Chúng em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các thầy cô để chúng em có điều kiện bổ sung, nâng cao ý thức của mình, phục vụ tốt hơn công tác thực tế sau này.

Chúng em xin chân thành cảm ơn!

**MỤC LỤC**

[LỜI CẢM ƠN](#_heading=h.3dy6vkm) 4

[CHƯƠNG 1. TỔNG QUAN](#_heading=h.147n2zr) 9

[1.1 Giới thiệu](#_heading=h.17dp8vu) 9

[1.1.1 Yêu cầu thực tế](#_heading=h.3rdcrjn) 9

[1.2 Mô tả bài toán.](#_heading=h.3o7alnk) 9

[1.2.1 Định nghĩa bài toán.](#_heading=h.23ckvvd) 9

[1.2.2 Các yêu cầu hệ thống.](#_heading=h.ihv636) 9

[1.3 Các nhóm chức năng của hệ thống](#_heading=h.32hioqz) 10

1.4 [Công cụ hỗ trợ. 10](#_heading=h.41mghml)

1.4[.1 Ngôn ngữ sử dụng.](#_heading=h.2grqrue) 10

1[.5. MongoDB là gì?](#_heading=h.vx1227) 10

1[.5.1 Giới thiệu về cơ sở dữ liệu MongoDB 10](#_heading=h.3fwokq0)

1[.5.2 Ưu điểm của MongoDB là gì? 11](#_heading=h.1v1yuxt)

1[.5.3 Khuyết điểm của MongoDB là gì? 1](#_heading=h.4f1mdlm)2

1[.5.4 Giới thiệu về cơ sở dữ liệu MongoDB compass 1](#_heading=h.2u6wntf)2

1[.6. Tìm hiểu ExpressJs 1](#_heading=h.19c6y18)2

1[.6.1. Giới thiệu về ExpressJs 12](#_heading=h.3tbugp1)

1[.6.2. Tại sao nên dùng ExpressJs 1](#_heading=h.28h4qwu)3

1[.6.3. Những tính năng của Expressjs 13](#_heading=h.2xcytpi)

1[.7. Tìm hiểu về NodeJs 14](#_heading=h.1mrcu09)

1[.8. Mô hình MVC là gì? 14](#_heading=h.46r0co2)

[CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 1](#_heading=h.1ksv4uv)5

[2.1. Các tác nhân của hệ thống. 1](#_heading=h.1hmsyys)5

[2.2](#_heading=h.2lwamvv). [Các use case 1](#_heading=h.2lwamvv)5

[2.2.1 Đặc tả các use case 1](#_heading=h.111kx3o)6

[2.2.2 Biểu đồ use case mức chi tiết](#_heading=h.111kx3o) 21

[2.2.3 Biểu đồ use case viết blog](#_heading=h.3l18frh) 22

[2.2.4 Biểu đồ use case tìm kiếm bài viết](#_heading=h.3l18frh) 22

[2.2.5 Biểu đồ use case xóa bài viết 2](#_heading=h.4k668n3)3

[2.3 Biểu đồ hoạt động 2](#_heading=h.1egqt2p)3

[2.3.1 Biểu đồ hoạt động viết blogs 2](#_heading=h.3ygebqi)3

[2.3.2 Biểu đồ hoạt động tìm kiếm bài viết 2](#_heading=h.sqyw64)4

[2.3.3 Biểu đồ hoạt động xóa bài viết 2](#_heading=h.1rvwp1q)5

[2.4 Biểu đồ trạng thái](#_heading=h.2r0uhxc) 26

[2.4.1 Biểu đồ trạng thái viết blogs 2](#_heading=h.1664s55)6

[2.4.2 Biểu đồ trạng thái tìm kiếm bài viết 2](#_heading=h.25b2l0r)7

[2.4.3 Biểu đồ trạng thái xóa bài viết 2](#_heading=h.34g0dwd)8

[2.5 Biểu đồ tuần tự 2](#_heading=h.43ky6rz)9

[2.5.1 Biểu đồ tuần tự viết blogs 2](#_heading=h.2iq8gzs)9

[2.5.2 Biểu đồ tuần tự tìm kiếm bài viết](#_heading=h.3hv69ve) 30

[2.5.3 Biểu đồ tuần tự xóa bài viết](#_heading=h.4h042r0) 31

[2.6 Biểu đồ giao tiếp 3](#_heading=h.1baon6m)2

[2.6.1 Biểu đồ giao tiếp viết blogs 3](#_heading=h.3vac5uf)2

[2.6.2 Biểu đồ giao tiếp tìm kiếm bài viết](#_heading=h.pkwqa1) 33

[2.6.3 Biểu đồ giao tiếp xóa bài viết 3](#_heading=h.1opuj5n)4

[CHƯƠNG 3. XÂY DỰNG CHƯƠNG TRÌNH](#_heading=h.zu0gcz) 37

3[.1.Giao diện của ứng dụng 3](#_heading=h.48pi1tg)7

3[.1.1.Trang chủ 3](#_heading=h.2nusc19)7

3[.1.2.Trang tin tức 3](#_heading=h.3mzq4wv)7

3[.1.3.Trang hỏi đáp 3](#_heading=h.haapch)8

3[.1.4.Trang liên hệ 3](#_heading=h.1gf8i83)8

3[.1.5. Trang viết blogs 3](#_heading=h.2fk6b3p)9

3[.1.6. Trang tổng quan 3](#_heading=h.3ep43zb)9

3[.1.7. Trang quản lý người dùng](#_heading=h.4du1wux) 40

3[.1.8. Trang quản lý bài viết](#_heading=h.184mhaj) 40

3[.1.9. Trang quản lý quản lý giao diện user](#_heading=h.279ka65) 41

[CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 4](#_heading=h.zu0gcz)3

4[.1 Kết quả đạt được và hạn chế của website 4](#_heading=h.3jtnz0s)3

4[.2 Hướng phát triển, khắc phục](#_heading=h.1yyy98l) 43

4[.3 Kết luận](#_heading=h.4iylrwe) 43

## DANH MỤC HÌNH ẢNH

Hình 0 Use case mức chi tiết 22

[Hình 1 Use case viết vlog](#_heading=h.2y3w247) 22

[Hình 2 Use case tìm kiếm bài viết](#_heading=h.49x2ik5) 23

[Hình 3 Use case xóa bài viết](#_heading=h.2zbgiuw) 23

[Hình 4 Biểu đồ hoạt động viết blogs](#_heading=h.2dlolyb) 24

[Hình 5 Biểu đồ hoạt động tìm kiếm bài viết](#_heading=h.3cqmetx) 25

[Hình 6 Biểu đồ hoạt động xóa bài viết](#_heading=h.4bvk7pj) 26

[Hình 7 Biểu đồ trạng thái viết blogs](#_heading=h.3q5sasy) 27

[Hình 8 Biểu đồ trạng thái tìm kiếm bài viết](#_heading=h.kgcv8k) 28

[Hình 9 Biểu đồ trạng thái xóa bài viết](#_heading=h.1jlao46) 29

[Hình 10 Biểu đồ tuần tự viết blogs](#_heading=h.xvir7l) 30

[Hình 11 Biểu đồ tuần tự tìm kiếm bài viết](#_heading=h.1x0gk37) 31

[Hình 12 Biểu đồ tuần tự xóa bài viết](#_heading=h.2w5ecyt) 32

[Hình 13 Biểu đồ giao tiếp viết blogs](#_heading=h.2afmg28) 33

[Hình 14 Biểu đồ giao tiếp tìm kiếm bài viết](#_heading=h.39kk8xu) 34

[Hình 15 Biểu đồ giao tiếp xóa bài viết](#_heading=h.2p2csry) 35

[Hình 16 Database mongodb compass tổng quan](#_heading=h.1302m92) 35

[Hình 17 Bảng blogusers](#_heading=h.2250f4o) 36

[Hình 18 Bảng loginregisters](#_heading=h.319y80a) 36

[Hình 19 Bảng slideshows](#_heading=h.40ew0vw) 37

[Hình 20 Trang tin tức](#_heading=h.2250f4o) 36

[Hình 21 Trang hỏi đáp](#_heading=h.319y80a) 36

[Hình 22 Trang liên hệ](#_heading=h.40ew0vw) 37

[Hình 23 Trang viết blogs](#_heading=h.upglbi) 37

[Hình 24 Trang tổng quan](#_heading=h.1tuee74) 38

[Hình 25 Trang quản lý người dùng](#_heading=h.2szc72q) 38

[Hình 26 Trang quản lý bài viết](#_heading=h.3s49zyc) 39

[Hình 27 Trang quản lý quản lý giao diện user](#_heading=h.meukdy) 39

**CHƯƠNG 1. TỔNG QUAN**

**1.1 Giới thiệu**

**1.1.1 Yêu cầu thực tế**

- Thế giới ngày một phát triển ,đời sống con người ngày càng hiện đại hóa, nhu cầu về đời sống vật chất và tinh thần của con người mỗi ngày càng một cao và có nhiều đòi hỏi khắt khe hơn.

- Các yêu cầu thiết yếu của chúng ta hắng ngày như : ăn uống , mua sắm, xem phim , nghe nhạc, đọc báo, xem tin tức…. Ở thời điểm công nghệ 4.0, các yêu cầu này dường như đang dần được đáp ứng một cách rất dễ dàng.

- Nắm bắt được nhu cầu đó và xét thấy sự phát triển vượt bậc của mạng Internet ở khắp nơi trên thế giới nói chung và tại nước ta trong những năm gần đây nói riêng nên việc ứng dụng công nghệ thông tin vào lĩnh vực tìm kiếm thông tin, truy cập dữ liệu từ xa không còn khó khăn. Người ta tận dụng tối đa những lợi ích mà lĩnh vực này mang lại.

- GHBlogs là một trong những trang web đặc thù áp dụng công nghệ thông tin vào hoạt động: cung cấp những thông tin mới nhất và chính xác nhất về các hãng laptop qua mạng và sau này sẽ có xu hường mở rộng. Mục tiêu của việc xây dựng trang web này nhằm giúp cho khách hàng có thể có thêm nhiều hiểu biết hơn về laptop, đây cũng là nơi mà người dùng có thể chia sẽ lại những kinh nghiệm, những hiểu biết của mình đến với mọi người xung quanh. Chỉ cần đăng nhập vào hệ thống với tài khoản đã có (nếu khách hàng đã là hội viên của cửa hàng) hay chỉ cần vài thao tác đăng kí đơn giản là khách hàng có thể tự do thảo luận, trao đổi tại hệ thống ( ) của chúng tôi.

* 1. **Mô tả bài toán.**

**1.2.1 Định nghĩa bài toán.**

- GHBlogs là một một website cung cấp cho người sử dụng, nhà quản lý các kiến thức bổ ích về các dòng laptop hiện hành hiện nay.

- Đối với khách hàng, hệ thống cho phép xem được các bài viết cũ cũng như mới nhất của website. Sau khi đăng kí làm thành viên của website, khách hàng có thể tự do chia sẻ lại những kiến thức mà mình có được cho mọi người xung quanh bằng những bài viết mà người dung tự viết ra.

**1.2.2 Các yêu cầu hệ thống.**

**-** Yêu cầu chức năng: Thiết kế giao diện của ứng dụng dễ nhìn và dễ hiểu, thiết kế cơ sở dữ liệu phù hợp với các chức năng xây dựng được hệ thống trang web hoạt động ổn định và đúng yêu cầu.

## + Xem thông tin người dùng.

## +Thêm, sửa, xóa các bài viết, quản lý được thông tin các người dùng, v.v…

**-** Yêu cầu phi chức năng**:** Tính bảo mật cao, giao diện thân thiện, hoạt động 24/7…

**1.3 Các nhóm chức năng của hệ thống**

- Các chức năng hệ thông bao gồm:

+ Nhóm chức năng đăng ký, đăng nhập thành viên.

+ Nhóm chức năng xem bài viết: xem nội dung bài viết, xem nội dung bình luận.

+ Nhóm chức năng quản lý thông tin: gồm quản lý thông tin cá nhân, quản lý danh sách bài viết, quản lý danh sách bình luận.

**1.4 Công cụ hỗ trợ.**

**1.4.1 Ngôn ngữ sử dụng.**

- Công cụ và ngôn ngữ thiết kế giao diện website :

+ Website được viết bằng các ngôn ngữ thông dụng như HTML/CSS/JS/SASS và có sử dụng template engine (handlebars).

- Xây dựng backEnd : ngôn ngữ NodeJs, ExpressJs và một vài thư viện hỗ trợ khác như( passport, nodemon, passport-local, express-session … ).

- Đặc biệt dự án được chia theo mô hình MVC.

- Cơ sở dữ liệu: MongoDB và MongoDB compass ( đây là mongodb client, là một công cụ giúp truy xuất được vào database mongoDB 1 cách dễ dàng hơn).

**1.5. MongoDB là gì?**

**1.5.1 Giới thiệu về cơ sở dữ liệu MongoDB**

- MongoDB là một dạng phần mềm cơ sở dữ liệu sử dụng mã nguồn mở NoSQL. Nó có thể hỗ trợ trên nhiều nền tảng khác nhau và được thiết kế với mục đích hướng đến đối tượng. MongoDB hoạt động dựa vào các khái niệm Collection và Document. Đồng thời, nó có hiệu suất cao cùng với tính khả dụng tốt và dễ dàng mở rộng.

- Các Collection trong MongoDB có cấu trúc cực kỳ linh hoạt. Điều này cho phép dữ liệu không cần thiết phải tuân theo bất kỳ một dạng cấu trúc nào. Vì thế, MongoDB có thể lưu trữ những dữ liệu có cấu trúc đa dạng và phức tạp. Dữ liệu trong MongoDB được lưu bằng định dạng kiểu JSON.



**1.5.2 Ưu điểm của MongoDB là gì?**

- Linh hoạt trong lưu trữ các kích cỡ dữ liệu khác nhau. Nhờ chúng được lưu dưới dạng JSON nên bạn thoải mái chèn bất kỳ thông tin nào tùy theo nhu cầu sử dụng.

- Tiết kiệm thời gian trong việc kiểm tra sự tương thích về cấu trúc khi thêm, xóa hoặc cập nhật dữ liệu. Nhờ MongoDB không có sự ràng buộc trong một khuôn khổ, quy tắc nhất định nào.

- Bạn dễ dàng mở rộng hệ thống thông qua việc thêm node vào cluster. Cụm các node này đóng vai trò như thư viện chứa các dữ liệu giao tiếp với nhau.

- Tốc độ truy vấn của MongoDB nhanh hơn so với RDBMS do toàn bộ dữ liệu truy vấn đã được ghi đệm lên bộ nhớ RAM. Nhờ thế, những lượt truy vấn sau sẽ được rút ngắn thời gian vì chúng không cần đọc từ ổ cứng.

- Trường dữ liệu “\_id” (đại diện cho giá trị duy nhất trong mỗi document) được tự động đánh chỉ mục nên hiệu suất luôn đạt mức cao nhất.

**1.5.3 Khuyết điểm của MongoDB là gì?**

- Bên cạnh các ưu điểm, MongoDB vẫn còn tồn tại một số điểm hạn chế mà bạn cần chú ý khi cài đặt và sử dụng:

+ Vì dữ liệu không bị ràng buộc nên trong quá trình sử dụng, bạn cần cẩn thận trong mọi thao tác nhằm tránh xảy ra những điều không mong muốn, làm ảnh hưởng đến dữ liệu.

+ Chương trình MongoDB tiêu tốn khá nhiều dung lượng bộ nhớ do dữ liệu được lưu dưới dạng key và value. Bên cạnh đó, một số collection chỉ có sự khác biệt về value nên việc lặp lại key là điều khó tránh khỏi. Điều này dẫn đến thừa dữ liệu.

+ Thông thường, thời gian để dữ liệu chuyển đổi từ RAM xuống ổ cứng khoảng 60s nên nguy cơ bị mất dữ liệu nếu xảy ra mất điện là điều có thể xảy ra.

**1.5.4 Giới thiệu về cơ sở dữ liệu MongoDB compass**

- Đây là một công cụ dùng để quản lý hệ cơ sở dữ liệu MongoDB.

- Là một tool mongo client giúp người dùng có thể truy xuất vào mongodb dễ hàng hơn.

- Giao diện trực quan.

**1.6. Tìm hiểu ExpressJs**

**1.6.1. Giới thiệu về ExpressJs**

**- Expressjs** hay còn được viết là Express js, Express.js. Đây là một framework được xây dựng trên nền tảng của **Nodejs**. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. **Expressjs** hỗ trợ các method HTTP và midleware tạo ra API vô cùng mạnh mẽ và dễ sử dụng.

- Vì Express js chỉ yêu cầu ngôn ngữ lập trình Javascript nên việc xây dựng các ứng dụng web và API trở nên đơn giản hơn với các lập trình viên và nhà phát triển. Expressjs cũng là một khuôn khổ của Node.js do đó hầu hết các mã code đã được viết sẵn cho các lập trình viên có thể làm việc.

**-** Nhờ có Expressjs mà các nhà lập trình có thể dễ dàng tạo các ứng dụng 1 web, nhiều web hoặc kết hợp. Do có dung lượng khá nhẹ, Expressjs giúp cho việc tổ chức các ứng dụng web thành một kiến trúc MVC có tổ chức hơn. Để có thể sử dụng được mã nguồn này, chúng ta cần phải biết về Javascript và HTML.

**1.6.2. Tại sao nên dùng ExpressJs**

\* Sự phổ biến của Javascript

- Javascript là một ngôn ngữ lập trình được sử dụng rộng rãi hiện nay, dễ đọc và được hỗ trợ ở mọi nơi. Mà Expressjs lại hỗ trợ Javascript, do đó nếu bạn đã biết đến Javascript thì chắc chắn việc lập trình bằng Express.js là vô cùng đơn giản. Ngay cả những người mới bắt đầu tham gia vào lĩnh vực phát triển web này cũng có thể sử dụng Expressjs.

- Javascript là ngôn ngữ lập trình dễ học với cả những người không có bất kỳ kiến thức gì về ngôn ngữ lập trình khác. Chính vì tính phổ biến, dễ học và dễ sử dụng này mà Express.js cho phép các tài năng trẻ tham gia và đạt được nhiều thành công trong phát triển ứng dụng web..

### - Hỗ trợ xây dựng website một cách nhanh chóng

### - Thời gian chính là tài sản quý giá của bất kỳ doanh nghiệp nào. Hơn thế, nhiều lập trình viên còn phải chịu áp lực xây dựng các ứng dụng web một cách hiệu quả trong thời gian ngắn. Nhưng để thực hiện tốt công việc này cần rất nhiều thời gian và Express.js chính là công cụ cứu cánh cho các nhà lập trình.

### - Express.js có thể giúp làm giảm một nửa thời gian viết mã mà vẫn xây dựng lên các ứng dụng web hiệu quả. Không chỉ trợ giúp về mặt thời gian Expressjs còn làm giảm những áp lực cần thiết để xây dựng với sự trợ giúp của các tính năng khác nhau của nó. Express js còn cung cấp một phần mềm trung gian đảm nhận nhiệm vụ đưa ra các quyết định để phản hồi chính xác những yêu cầu của khách hàng.

**1.6.3. Những tính năng của Expressjs**

- Phát triển máy chủ nhanh chóng: Expressjs cung cấp nhiều tính năng dưới dạng các hàm để dễ dàng sử dụng ở bất kỳ đâu trong chương trình. Điều này đã loại bỏ nhu cầu viết mã từ đó tiết kiệm được thời gian.

- Phần mềm trung gian Middleware: Đây là phần mềm trung gian có quyền truy cập vào cơ sở dữ liệu, yêu cầu của khách hàng và những phần mềm trung gian khác. Phần mềm Middleware này chịu trách nhiệm chính cho việc tổ chức có hệ thống các chức năng của Express.js

- Định tuyến - Routing: Express js cung cấp cơ chế định tuyến giúp duy trì trạng thái của website với sự trợ giúp của URL.

- Tạo mẫu - Templating: Các công cụ tạo khuôn mẫu được Express.js cung cấp cho phép các nhà xây dựng nội dung động trên các website bằng cách tạo dựng các mẫu HTML ở phía máy chủ.

- Gỡ lỗi - Debugging: Để phát triển thành công các ứng dụng web không thể thiết đi việc gỡ lỗi. Giờ đây với Expressjs việc gỡ lỗi đã trở nên dễ dàng hơn nhờ khả năng xác định chính xác các phần ứng dụng web có lỗi.

**1.7. Tìm hiểu về NodeJs**

- NodeJS là một nền tảng được xây dựng trên V8 JavaScript Engine – trình thông dịch thực thi mã JavaScript, giúp xây dựng các ứng dụng web một cách đơn giản và dễ dàng mở rộng.

- NodeJS được phát triển bởi Ryan Dahl vào năm 2009 và có thể chạy trên nhiều hệ điều hành khác nhau: OS X, Microsoft Windows, Linux.

**1.8 Mô hình MVC là gì?**

- MVC= Model + View + Controller Model: Chính là dự liệu (cụ thể hơn trong ZF sẽ là các class làm nhiệm vụ thao tác trực tiếp xuống DBMS)

- Mô hình Views có nhiệm vụ liên kết với Mô hình Model và xuất các dữ liệu ra trình duyệt theo nhu cầu đòi hỏi của người tiêu dùng(user). Điển hình là các văn bản HTML.

- View: Làm nhiệm vụ render trang web từ các action do Controller truyền sang + dữ liệu từ model (có thể hiểu nó như template render). ∙ Controller: chính là phần cốt lõi, điều hành trang web của bạn, 1 trang web có thể có nhiều module (có thể hiểu như compoment của  Joomla), một module có thể có nhiều controller. Một controller sẽ gồm nhiều action.

- Ví dụ trong Forum sẽ có Post Controller bao gồm các action như listAction, readAction, writeAction,…

**CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG**

* 1. **Các tác nhân của hệ thống.**

- Khách xem: là người ghé qua ứng dụng, xem thông tin sản phẩm.

- Người dùng: là những người đã thăm đã đăng kí thành viên của hệ thống.

- Người quản trị: điều hành, quản lý và theo dõi mọi hoạt động của hệ thống:

- Thêm sửa xóa thành viên, thêm sửa xóa bài viết.

**2.2 Các use case**

\* Actor:

- Người dùng

- Quản trị

\* Use case:

- Đăng nhập

- Đăng kí

- Cập nhập bình luận

+ Thêm bình luận

+ Chỉnh sửa bình luận

+ Xóa bình luận

- Viết blogs

+ Tìm kiếm bài viết

+ Cập nhập thông tin cá nhân

- Thêm một/Nhiều thông tin

- Chỉnh sửa một/nhiều thông tin

- Xóa một/nhiều thông tin

+ Cập nhập bài viết

- Thêm bài viết

- Chỉnh sửa bài viết

- Xóa bài viết

+ Liên hệ

+ Quản lý bài viết

- Thêm bài viết

- Chỉnh sửa bài viết

- Xóa bài viết

+ Quản lý bình luận

- Thêm bình luận

- Chỉnh sửa bình luận

- Xóa bình luận

+ Quản lý người dùng

- Thêm người dùng

- Chỉnh sửa người dùng

- Xóa người dùng

**2.2.1 Đặc tả các use case**

## - Use-case : Đăng kí thành viên.

## - Tác nhân : Khách xem.

## - Mô tả: sau khi truy cập vào ứng dụng và nhấn nút đăng kí hệ thống cung cấp các thông tin cần thiết để đăng kí nếu đăng kí thành công thông tin sẽ được lưu vào database và tài khoản được khởi tạo.

**- Use-case:** Đăng nhập hệ thống

**- Tác nhân**: Người quản trị, Người dùng

**- Mô tả**: sau khi đăng kí thành công khách hàng tiến hành đăng nhập vào hệ thống, hệ thống sẽ hiển thi ra form nhập thông tin để khách hàng có thể tiến hành đăng nhập vào trang chủ và sử dụng website

**- Use-case:** Thêm bình luận

**- Actor**: Admin, người dùng

**- Object**: Mô tả admin/người dùng thêm bình luận vào hệ thống.

**- Description**: Khi admin/người dùng muốn thêm bình luận vào hệ thống thì admin/người dùng sẽ truy cập vào hệ thống bình luận chọn chức năng bình luận, thực hiện nhập nội dung và xác nhận.

**- Use-case:** Chỉnh sửa bình luận

**- Actor**: Admin, người dùng

**- Object**: Mô tả admin/người dùng sửa nội dung bình luận có trong hệ thống.

**- Description**: Khi admin/người dùng muốn thay đổi nội dung bình luận thì admin/người dùng sẽ truy cập vào hệ thống bình luận chọn chức năng sửa, thực hiện nhập nội dung vào bình luận và xác nhận thay đổi bình luận đó.

**- Use-case:** Xóa bình luận

**- Actor**: Admin, người dùng

**- Object**: Mô tả admin/người dùng xóa bình luận có trong hệ thống.

**- Description**: Khi admin/người dùng muốn xóa bình luận trong hệ thống thì admin/người dùng sẽ truy cập vào hệ thống bình luận chọn chức năng xóa bình luận, thực hiện nhập nội dung và xác nhận.

**- Use-case:** Viết blogs

**- Tác nhân**: khách hàng

**- Mô tả**: sau khi đăng nhập thành công, khách hàng có thể click vào mục viết blogs trên thanh điều hướng để có thể được điều hướng vào trang viết blogs. Nơi đây, khách hàng có thể chia sẻ những kinh nghiệm, những kiến thức mà mình tích lũy được về các hãng laptop

**- Use-case:** Tìm kiếm bài viết

**- Tác nhân**: Người dùng

**- Mô tả**: Người dùng muốn tìm kiếm bài viết có thể chọn mục tìm kiếm hiển thị trên giao diện hệ thống sẽ giúp người dùng lọc ra được những bài viết mới nhất để người dùng có thể tùy ý lựa chọn

**- Use-case:** Thêm thông tin cá nhân

**- Actor**: Người dùng

**- Object**: Mô tả người dùng thêm thông tin cá nhân vào hệ thống.

**- Description**: Khi người dùng muốn thêm thông tin vào hệ thống thì người dùng sẽ truy cập vào hệ thống cá nhân chọn chức năng thêm thông tin, thực hiện nhập thông tin cá nhân và xác nhận

**- Use-case:** Chỉnh sửa thông tin cá nhân

**-  Actor**: Người dùng

**- Object**: Mô tả người dùng sửa thông tin cá nhân có trong hệ thống.

**- Description**: Khi người dùng trong hệ thống muốn thay đổi thông tin thì người dùng sẽ truy cập vào hệ thống cá nhân chọn chức năng sửa thông tin, thực hiện nhập thông tin vào hệ thống và xác nhận thay đổi thông tin đó.

**- Use-case:** Xóa thông tin cá nhân

**- Actor**: Người dùng

**- Object**: Mô tả người dùng xóa thông tin cá nhân có sẵn trong hệ thống.

**- Description**: Khi người dùng muốn xóa thông tin cá nhân vào hệ thống thì người dùng sẽ truy cập vào hệ thống cá nhân chọn chức năng xóa thông tin cá nhân và xác nhận.

**- Use-case:** Thêm bài viết

**- Actor**: Admin, người dùng

**- Object**: Mô tả admin/người dùng thêm bài viết vào hệ thống.

**- Description**: Khi admin/người dùng muốn thêm bài viết vào hệ thống thì admin/người dùng sẽ truy cập vào hệ thống bài viết chọn chức năng thêm bài viết, thực hiện nhập bài viết và xác nhận.

**- Use-case:** Chỉnh sửa bài viết

**- Actor**: Admin, người dùng

**- Object**: Mô tả admin/người dùng sửa nội dung bài viết có trong hệ thống.

**- Description**: Khi admin/người dùng muốn thay đổi nội dung bài viết thì admin/người dùng sẽ truy cập vào hệ thống bài viết chọn chức năng sửa, thực hiện nhập nội dung vào bài viết và xác nhận thay đổi nội dung đó.

**- Use-case:** Xóa bài viết

**- Actor**: admin, người dùng

**- Object**: Mô tả admin/người dùng xóa bài viết có trong hệ thống.

**- Description**: Khi admin/người dùng muốn xóa bài viết thì admin/người dùng sẽ truy cập vào hệ thống bài viết chọn chức năng xóa bài viết, thực hiện xóa bài viết và xác nhận.

**- Use-case:** Liên hệ

**- Actor**: người dùng

**- Object**: mô tả người dùng có thể liên hệ với admin mỗi khi có sự cố về website

**- Description**: Khi người dùng muốn liên hệ với admin để nhờ trợ giúp một thứ gì đò, người dùng sẽ truy cập vào trang liên hệ và tiến hành điền đầy đủ thông tin và mô tả vấn đề mình đang gặp phải, form sẽ được gửi vào mail admin

**- Use-case:** Quản lý thêm bài viết

**- Actor**: Quản lý

**- Object**: Mô tả quản lý thêm bài viết vào hệ thống.

**- Description**: Khi quản lý muốn thêm bài viết vào hệ thống thì quản lý sẽ truy cập vào hệ thống bài viết chọn chức năng thêm bài viết, thực hiện nhập nội dung và xác nhận.

**- Use-case:** Quản lý chỉnh sửa bài viết

**- Actor**: Quản lý

**- Object**: Mô tả quản lý sửa nội dung bài viết có trong hệ thống.

**- Description**: Khi quản lý muốn thay đổi nội dung bài viết thì quản lý sẽ truy cập vào hệ thống bài viết chọn chức năng sửa, thực hiện nhập nội dung vào bài viết và xác nhận thay đổi bài viết đó.

**- Use-case:** Quản lý xóa bài viét

**- Actor**: Quản lý

**- Object**: Mô tả quản lý xóa bài viết có trong hệ thống.

**- Description**: Khi quản lý muốn xóa bài viết thì quản lý sẽ truy cập vào hệ thống bài viết chọn chức năng xóa, thực hiện xóa bài viết và xác nhận.

**- Use-case:** Quản lý thêm bình luận

**- Actor**: Quản lý

**- Object**: Mô tả quản lý thêm bình luận vào hệ thống.

**- Description**: Khi quản lý muốn thêm bình luận vào hệ thống thì quản lý sẽ truy cập vào hệ thống bình luận chọn chức năng bình luận, thực hiện nhập nội dung và xác nhận.

**- Use-case:** Quản lý chỉnh sửa bình luận

**- Actor**: Quản lý

**- Object**: Mô tả quản lý sửa nội dung bình luận có trong hệ thống.

**- Description**: Khi quản lý muốn thay đổi nội dung bình luận thì quản lý sẽ truy cập vào hệ thống bình luận chọn chức năng sửa, thực hiện nhập nội dung vào bình luận và xác nhận thay đổi bình luận đó.

**- Use-case:** Quản lý xóa bình luận

**- Actor**: Quản lý

**- Object**: Mô tả quản lý xóa bình luận có trong hệ thống.

**- Description**: Khi quản lý muốn xóa bình luận thì quản lý sẽ truy cập vào hệ thống bình luận chọn chức năng xóa, thực hiện xóa bình luận và xác nhận.

**- Use-case:** Quản lý thêm người dùng

**- Actor**: Quản lý

**- Object**: Mô tả người quản lý thêm thông tin một người dùng vào hệ thống

**- Description**: Khi người dùng đăng ký thì người quản lý sẽ truy cập vào hệ thống quản lý chọn chức năng thêm người dùng, thực hiện nhập thông tin người dùng và xác nhận

**- Use-case:** Quản lý chỉnh sửa người dùng

**- Actor**: Quản lý

**- Object**: Mô tả người quản lý sửa thông tin một người dùng có trong hệ thống

**- Description**: Khi người dùng trong hệ thống có thay đổi thông tin người quản lý sẽ thực hiện truy cập trang quản lý và chọn chức năng sửa thông tin cá nhân của người dùng đó, thực hiện nhập thông tin vào hệ thống xác nhận thay đổi thông tin đó

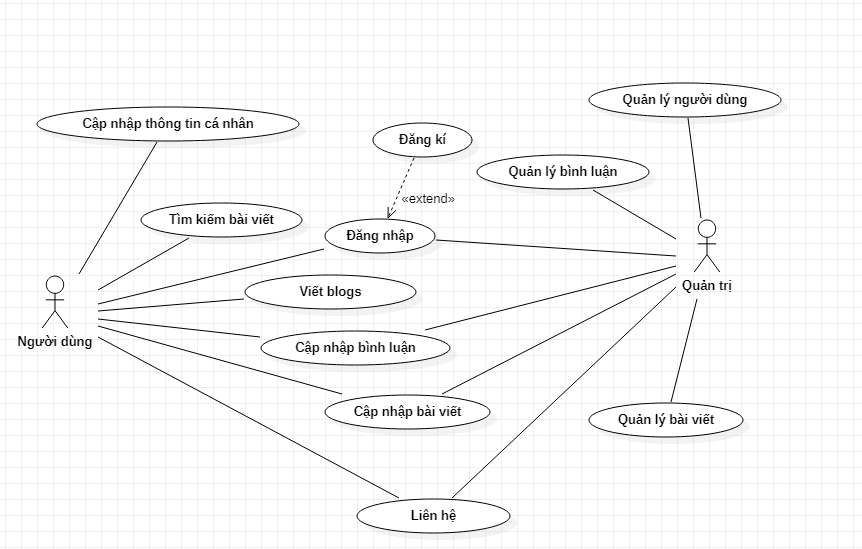
**- Use-case:** Quản lý xóa người dùng

**- Actor**: Quản lý

**- Object**: Mô tả người quản lý xóa thông tin một người dùng có trong hệ thống

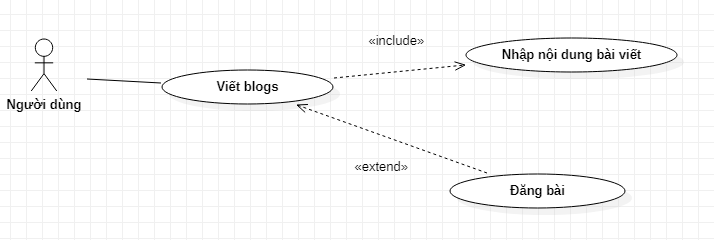
**- Description**: Khi người dùng không làm trong cửa hàng nữa thì người quản lý sẽ truy cập vào hệ thống quản lý chọn chức năng xóa người dùng, thực hiện xóa thông tin người dùng và xác nhận.

**2.2.2** **Biểu đồ use case mức chi tiết**

****

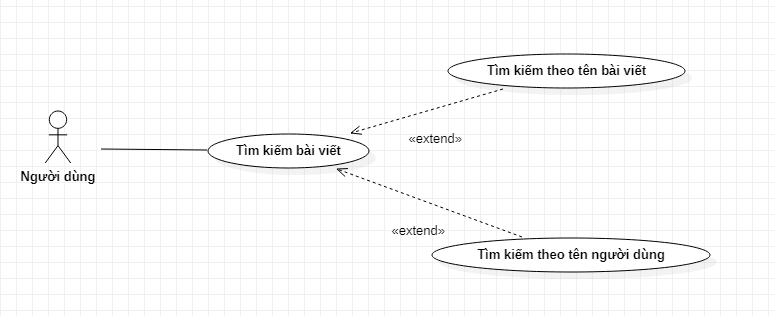
*Hình 0 Use case mức chi tiết*

**2.2.3 Biểu đồ use case viết blogs.**



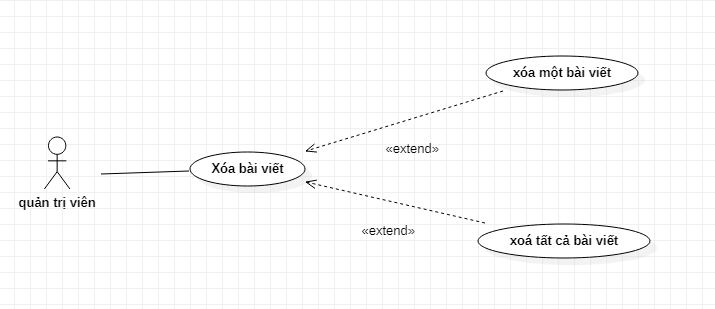
*Hình 1 Use case viết vlog*

**2.2.4 Biểu đồ use case tìm kiếm bài viết**

****

*Hình 2 Use case tìm kiếm bài viết*

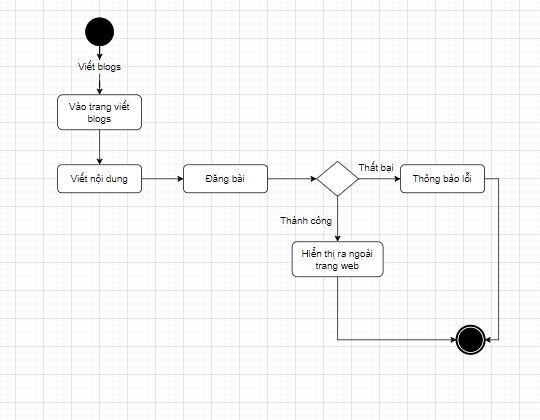
**2.2.5 Biểu đồ use case xóa bài viết**

****

*Hình 3 Use case xóa bài viết*

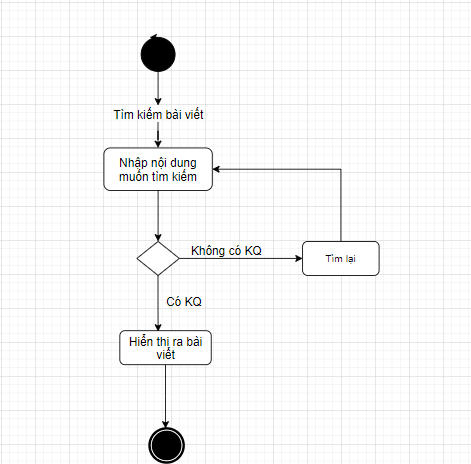
**2.3 Biểu đồ hoạt động**

**2.3.1 Biểu đồ hoạt động viết blogs**

****

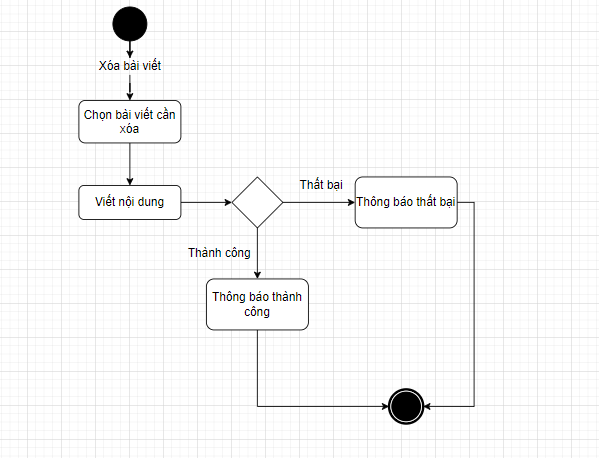
*Hình 4 Biểu đồ hoạt động viết blogs*

**2.3.2 Biểu đồ hoạt động tìm kiếm bài viết**

****

*Hình 5 Biểu đồ hoạt động tìm kiếm bài viết*

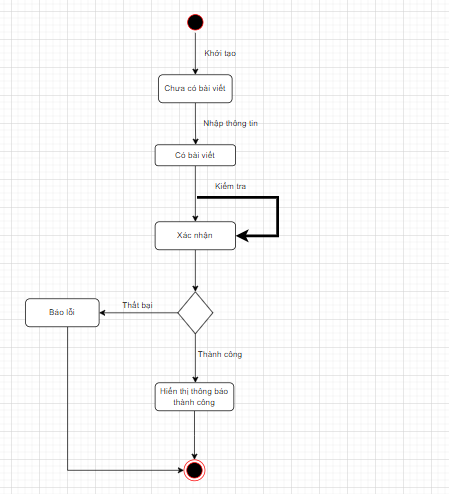
**2.3.3 Biểu đồ hoạt động xóa bài viết**

****

*Hình 6 Biểu đồ hoạt động xóa bài viết*

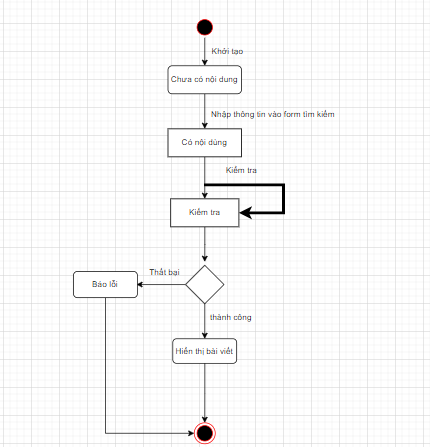
**2.4 Biểu đồ trạng thái**

**2.4.1 Biểu đồ trạng thái viết blogs**

****

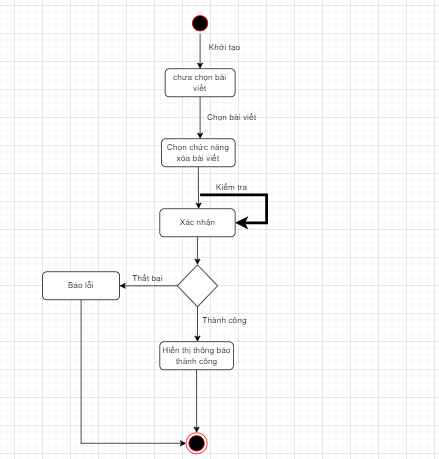
*Hình 7 Biểu đồ trạng thái viết blogs*

**2.4.2 Biểu đồ trạng thái tìm kiếm bài viết**

****

*Hình 8 Biểu đồ trạng thái tìm kiếm bài viết*

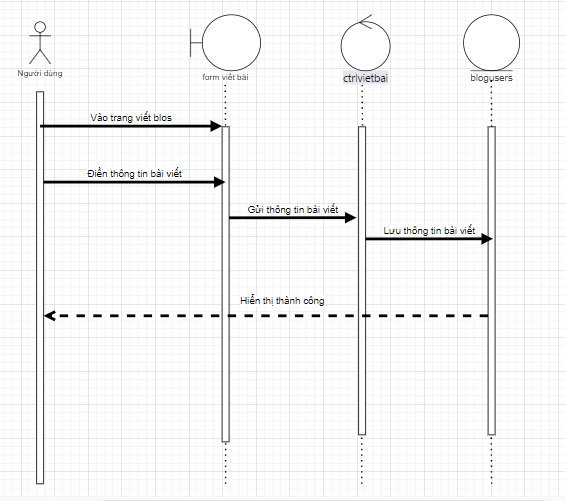
**2.4.3 Biểu đồ trạng thái xóa bài viết**

****

*Hình 9 Biểu đồ trạng thái xóa bài viết*

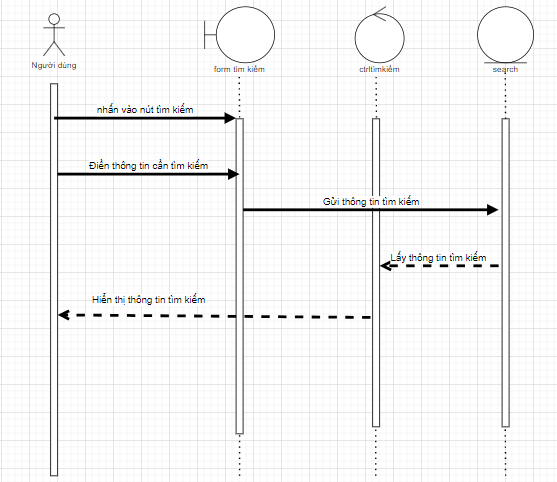
**2.5 Biểu đồ tuần tự**

**2.5.1 Biểu đồ tuần tự viết blogs**

****

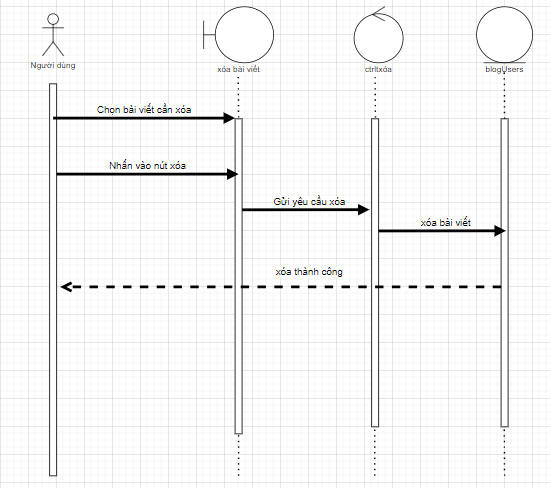
*Hình 10 Biểu đồ tuần tự viết blogs*

**2.5.2 Biểu đồ tuần tự tìm kiếm bài viết**

****

*Hình 11 Biểu đồ tuần tự tìm kiếm bài viết*

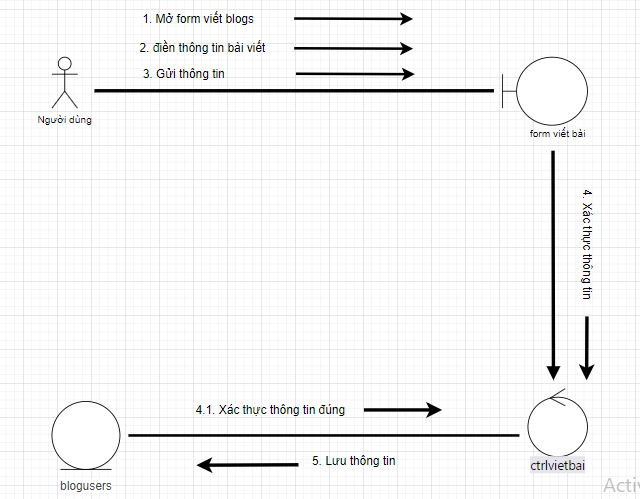
**2.5.3 Biểu đồ tuần tự xóa bài viết**

****

*Hình 12 Biểu đồ tuần tự xóa bài viết*

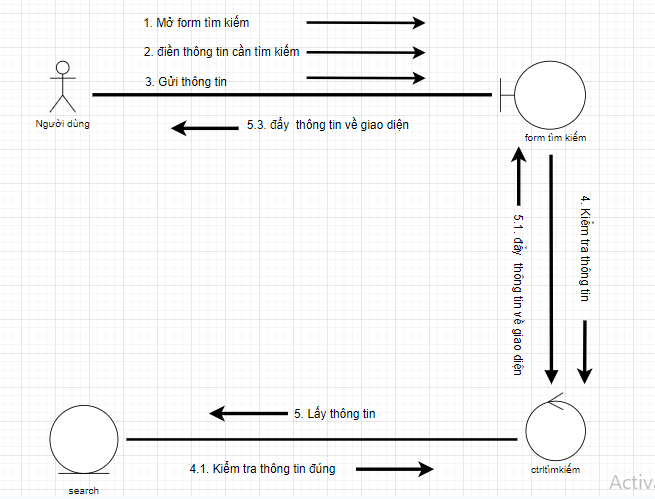
**2.6 Biểu đồ giao tiếp**

**2.6.1 Biểu đồ giao tiếp viết blogs**

****

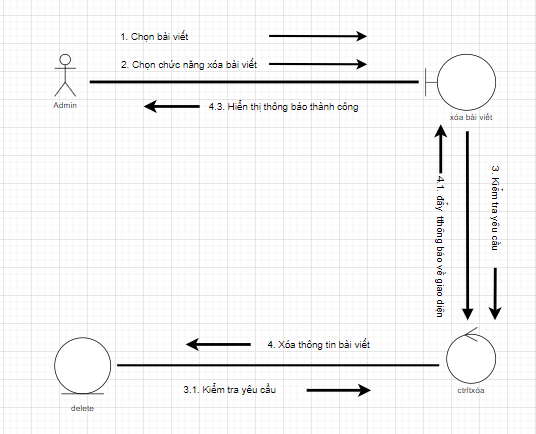
*Hình 13 Biểu đồ giao tiếp viết blogs*

**2.6.2 Biểu đồ giao tiếp tìm kiếm bài viết**

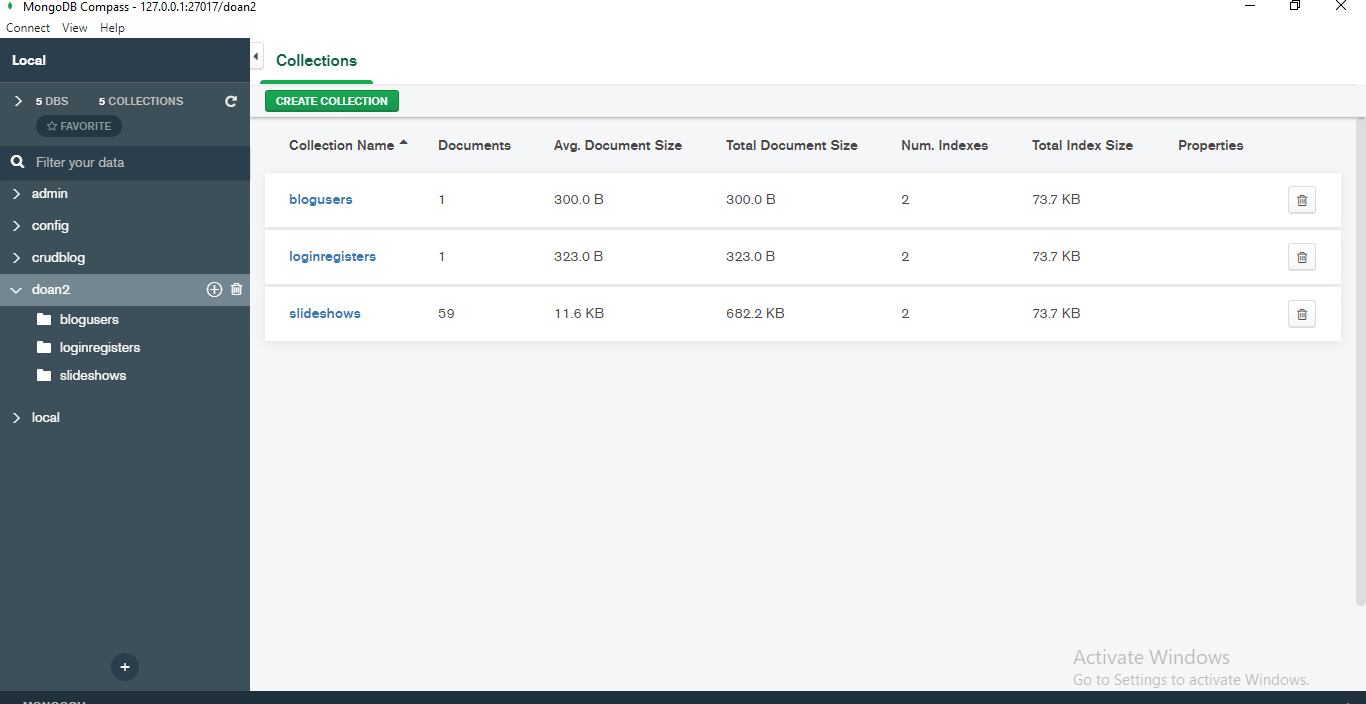
****

*Hình 14 Biểu đồ giao tiếp tìm kiếm bài viết*

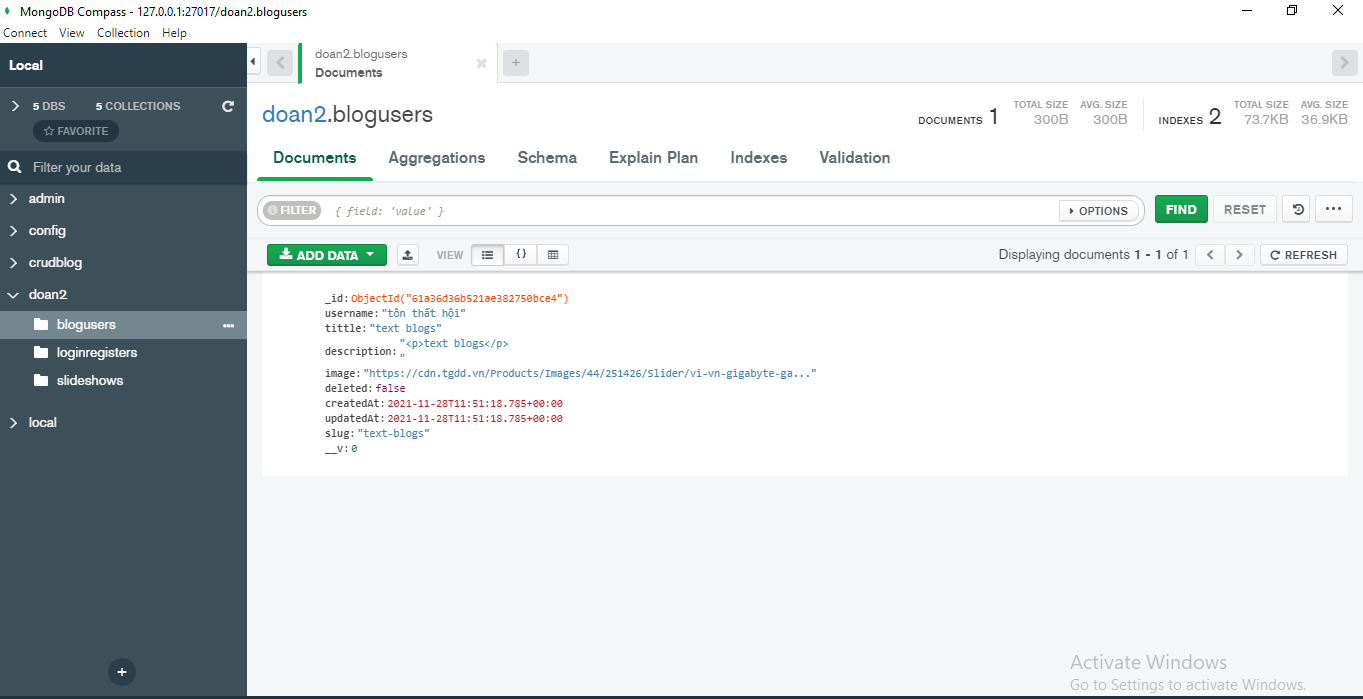
**2.6.3 Biểu đồ giao tiếp xóa bài viết**

****

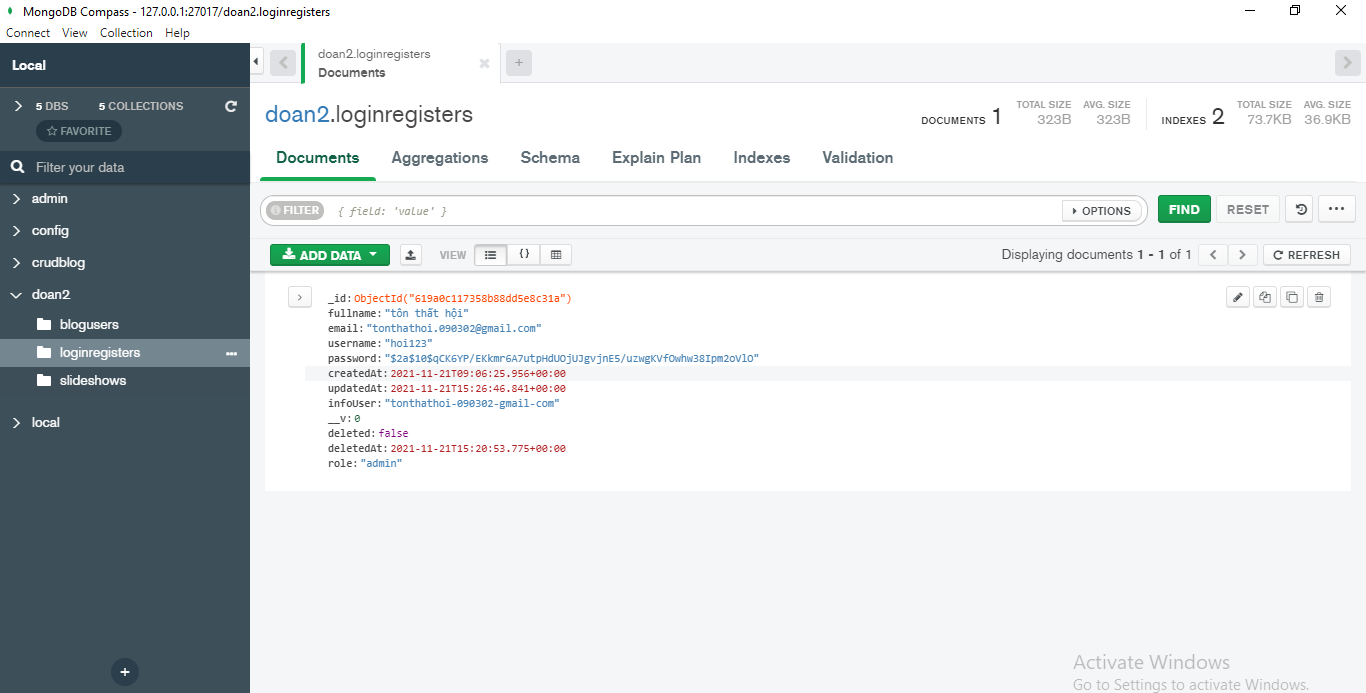
*Hình 15 Biểu đồ giao tiếp xóa bài viết*

**

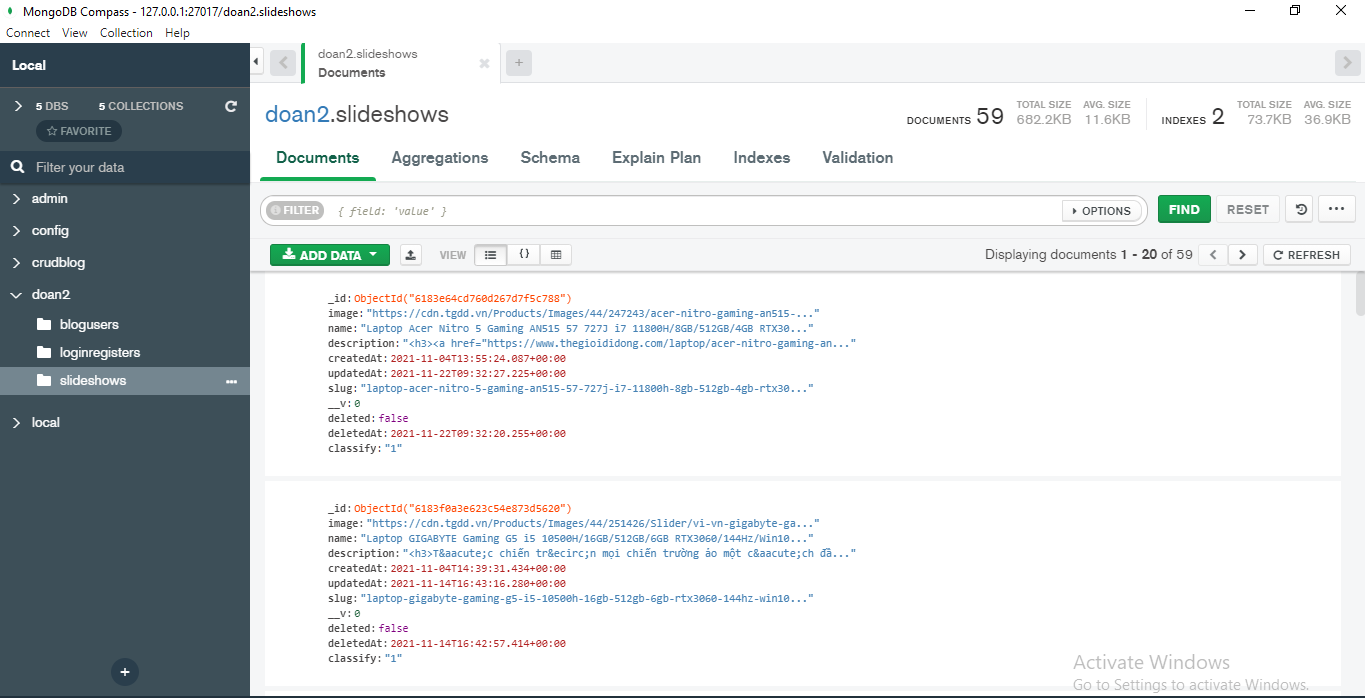
*Hình 16. Database mongodb compass tổng quan*

**

*Hình 17. Bảng blogusers*

**

*Hình 18. Bảng loginregisters*

**

*Hình 19. Bảng slideshows*

**CHƯƠNG 3. XÂY DỰNG CHƯƠNG TRÌNH**

**3.1.Giao diện của ứng dụng**

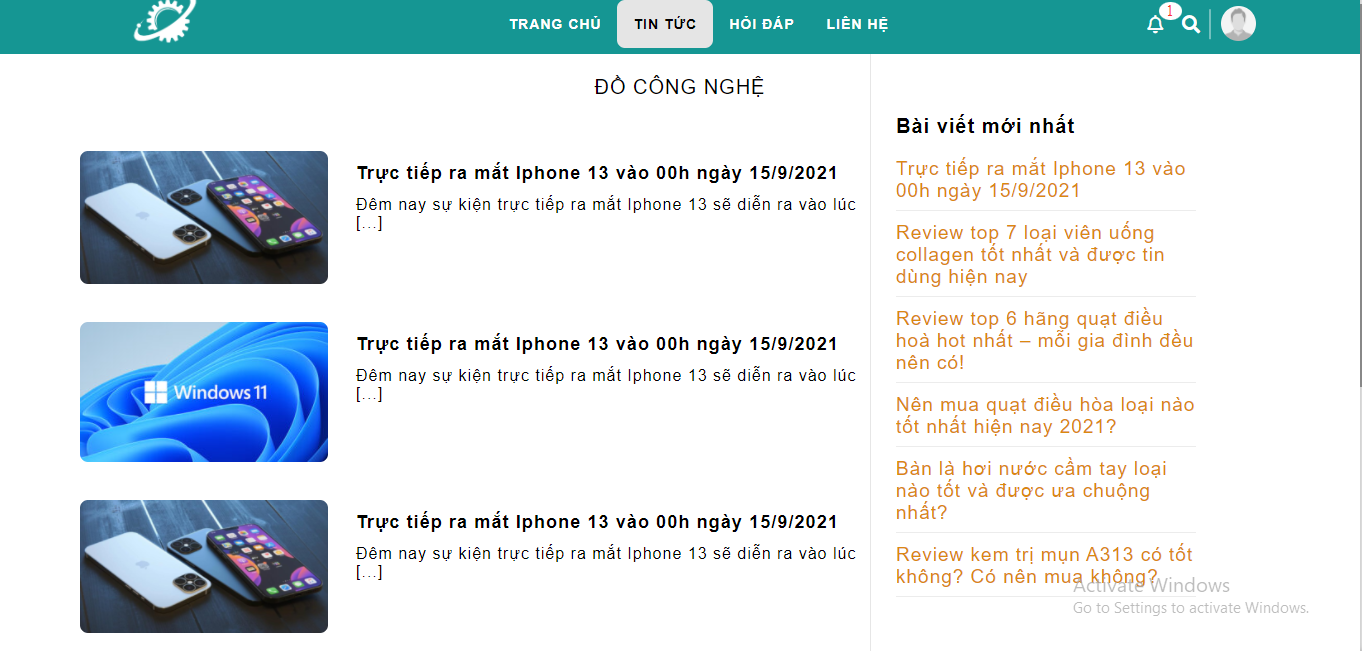
**Giao diện user**

**3.1.1.Trang chủ**

****

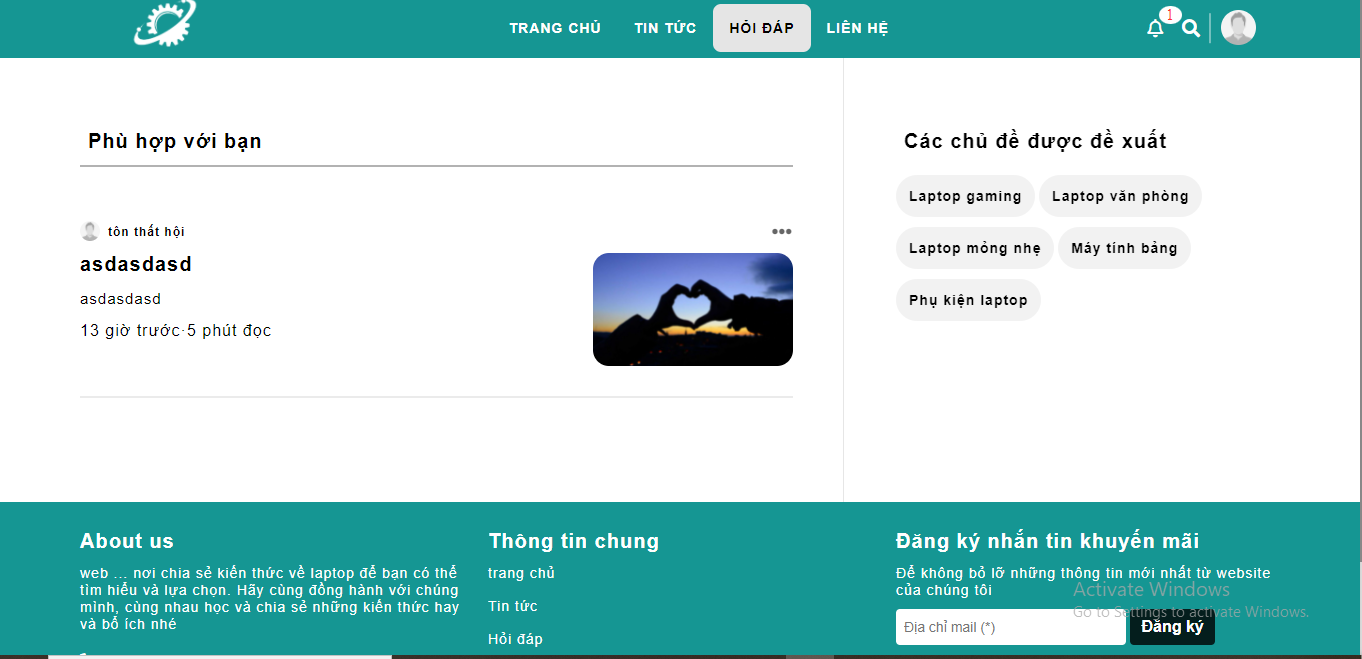
*Hình 20 Trang chủ*

**3.1.2.Trang tin tức**

****

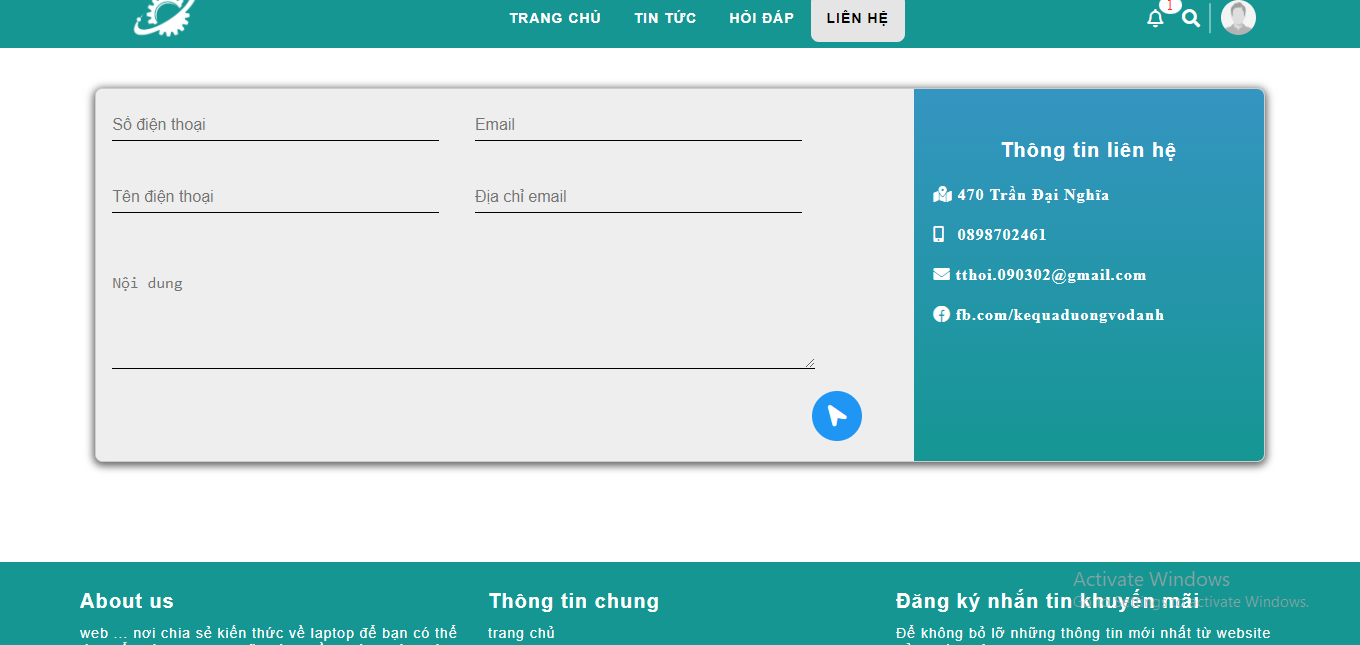
*Hình 21 Trang tin tức*

**3.1.3.Trang hỏi đáp**

****

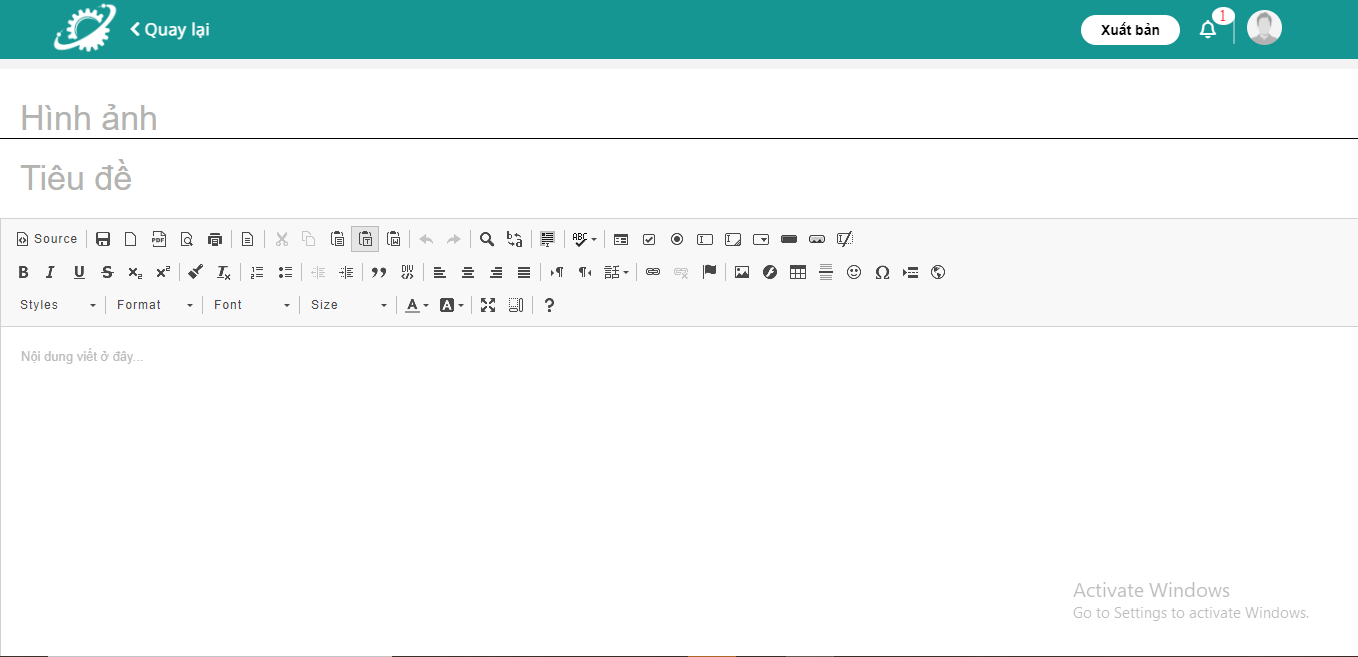
*Hình 22 Trang hỏi đáp*

**3.1.4.Trang liên hệ**

****

*Hình 23 Trang liên hệ*

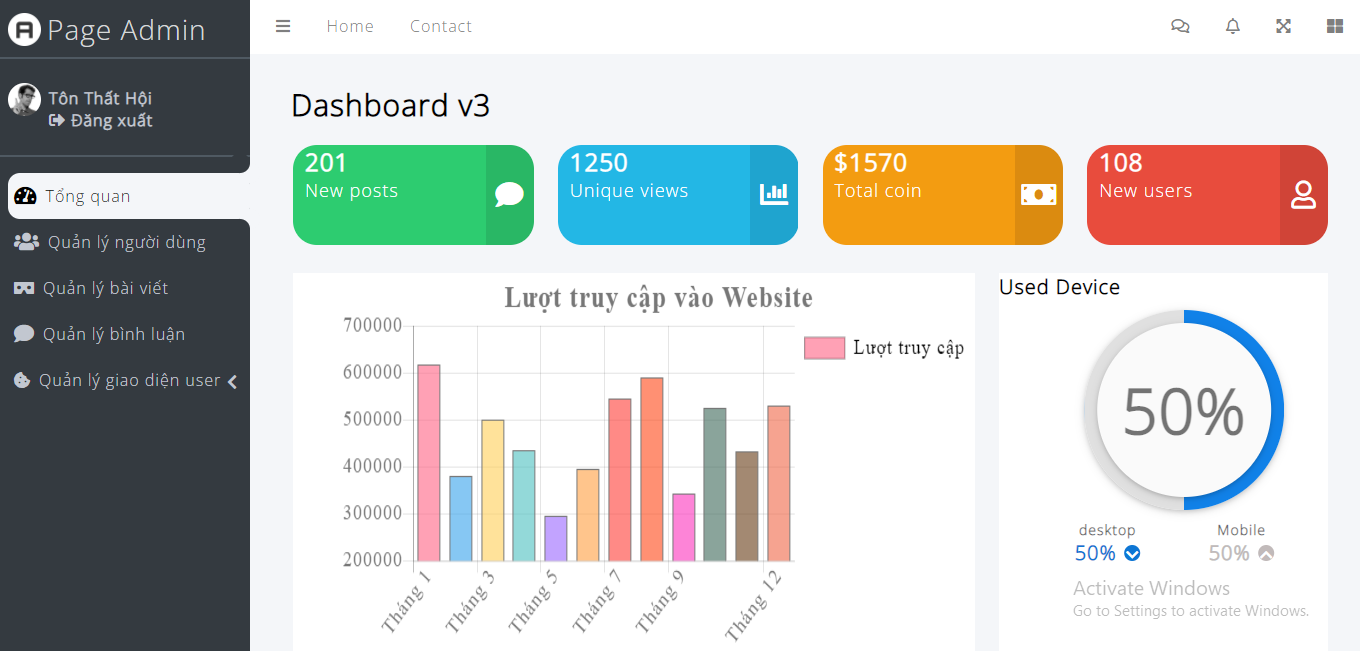
**3.1.5. Trang viết blogs**

****

*Hình 24 Trang viết blogs*

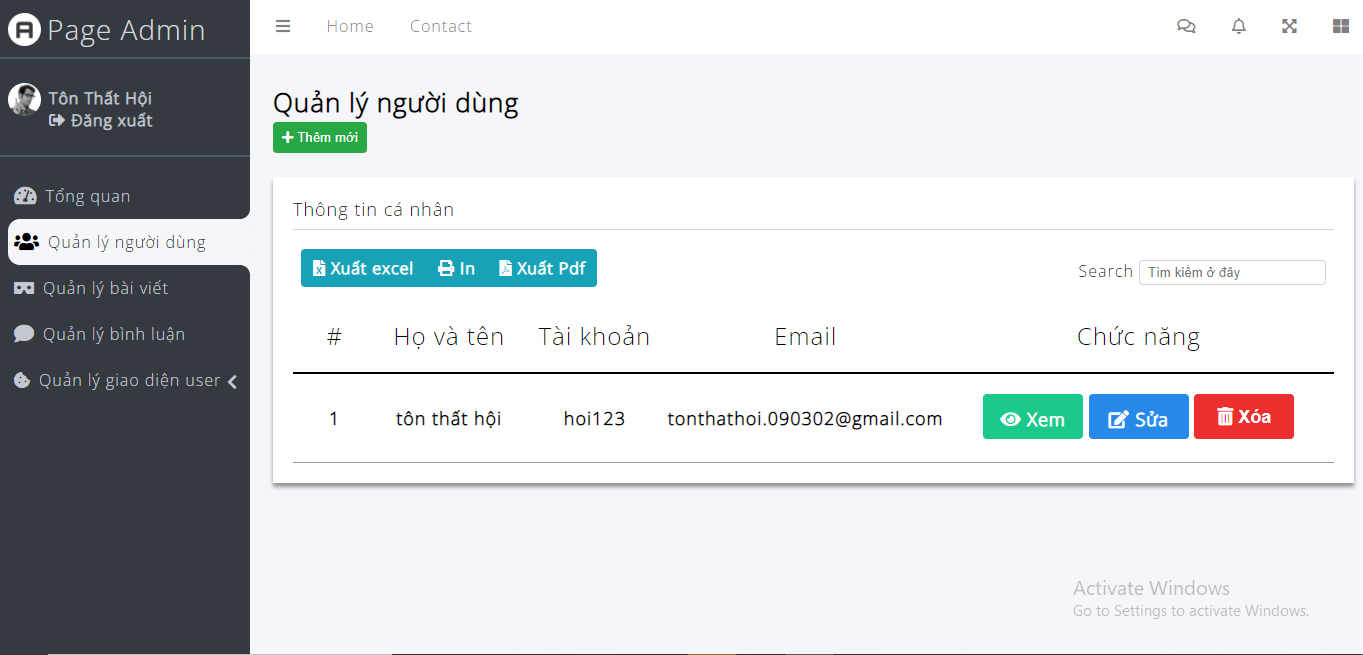
**Giao diện admin**

**3.1.6. Trang tổng quan**

****

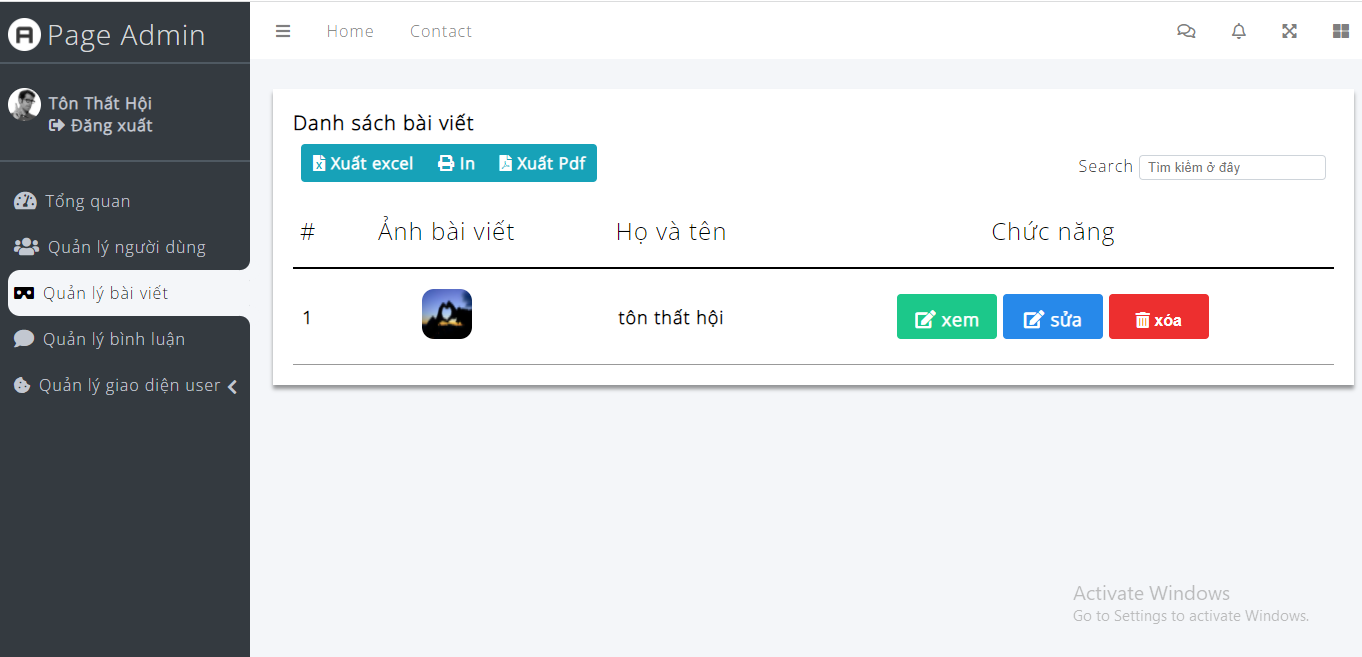
*Hình 25 Trang tổng quan*

**3.1.7. Trang quản lý người dùng**

****

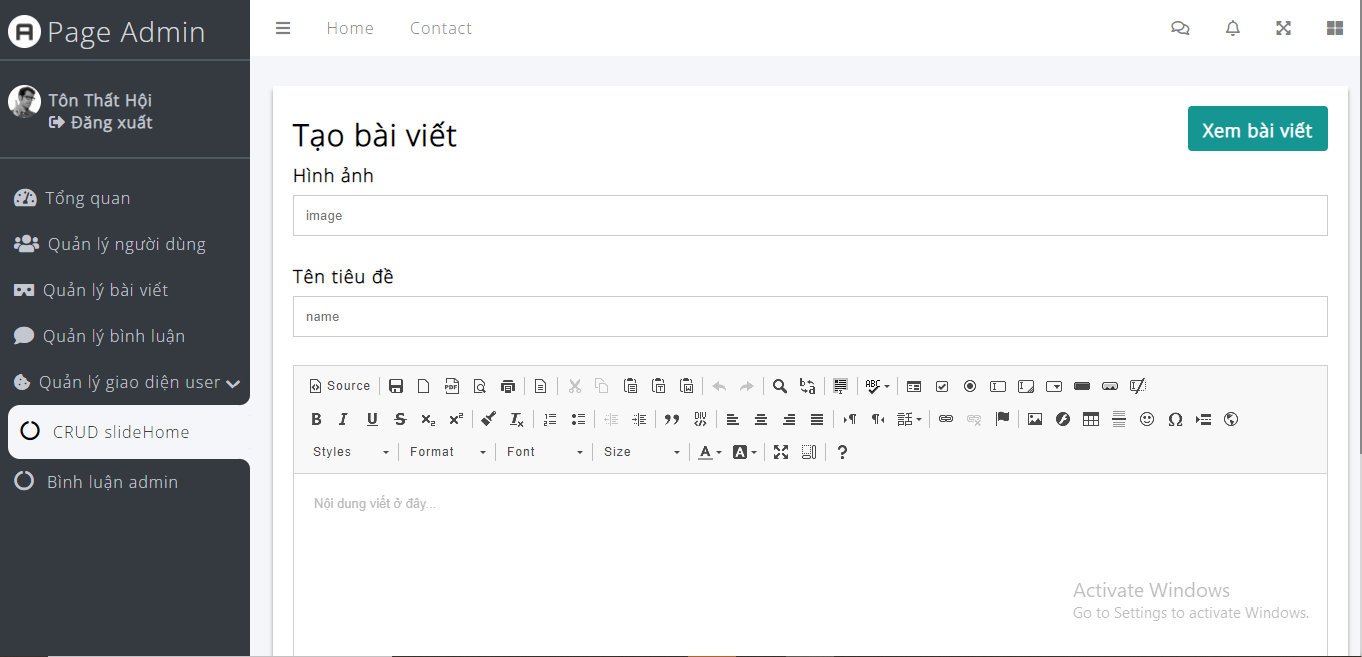
*Hình 26 Trang quản lý người dùng*

**3.1.8. Trang quản lý bài viết**

****

*Hình 27 Trang quản lý bài viết*

**3.1.9. Trang quản lý quản lý giao diện user**

****

*Hình 28 Trang quản lý quản lý giao diện user*

**CHƯƠNG 4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

**4.1 Kết quả đạt được và hạn chế của website**

**-** Qua quá trình làm đồ án chúng em đã được trao dồi thêm kiến thức về cách một ứng dụng được làm ra như thế nào, cách thức hoạt động ra sao qua đó chung em còn được bổ sung một số kiến thức về ngôn ngữ lập trình và nâng cao kiến thức về cơ sở dữ liệu Thứ hai tuy đã hoàn thành sản phẩm nhưng ứng dụng bán hàng điển tử của chúng em còn nhiều mặt hạn chế như:

+ Cơ sở dữ liệu còn sơ sài.

+ Giao diện còn đơn giản.

+ Còn một số chức năng chưa hoàn thành.

**4.2 Hướng phát triển, khắc phục**

- Bổ sung các chức năng còn thiếu của sản phẩm.

- Cải thiện cơ sở dữ liệu cũng như giao diện.

**4.3 Kết luận**

- Với những kiến thức mà em đã học được ở trường, những kinh nghiệm thực tế từ thời gian qua kèm với sự hướng dẫn của các thầy cô, em đã hoàn thành đề tài xây dựng ứng dụng bán hàng hàng điện tử online. Đề tài này tuy không mới nhưng nó đã giúp em được học hỏi và bổ sung thêm nhiều kiến thức mới để em có thể hoàn thiện mình trong tương lai. Tuy nhiên, khi làm sẽ không tránh khỏi những sai sót. Em rất mong thầy cô sẽ giúp đỡ em hoàn thiện hơn. Em xin chân thành cảm ơn!