**TRƯỜNG ĐẠI HỌC KINH TẾ QUỐC DÂN**

VIỆN CÔNG NGHỆ THÔNG TIN VÀ KINH TẾ SỐ

Logo

Description automatically generated

**BÁO CÁO CHUYÊN ĐỀ THỰC TẬP**

**Đề tài**

**XÂY DỰNG WEBSITE CHIA SẺ THÔNG TIN REDDIT VIỆT NAM**

|  |  |
| --- | --- |
| Tên:  Mã sinh viên:  Lớp: | Đỗ Tuấn Đức  11200832  Công nghệ thông tin 62A |
| Giảng viên hướng dẫn: | TS. Đặng Minh Quân |

*Hà Nội, tháng 4 năm 2024*

MỤC LỤC

[*LỜI MỞ ĐẦU 3*](#_Toc164345679)

[*LỜI CAM ĐOAN 4*](#_Toc164345680)

[*CÁC THUẬT NGỮ VIẾT TẮT 5*](#_Toc164345681)

[*DANH MỤC HÌNH 6*](#_Toc164345682)

[*DANH MỤC BẢNG 8*](#_Toc164345683)

[*CHƯƠNG 1: TỔNG QUAN CÔNG TY FPT SOFTWARE 9*](#_Toc164345684)

[*1.1. Tổng quan Công ty FPT SOFTWARE 9*](#_Toc164345685)

[*1.1.1. Thông tin chung về Công ty FPT Software 9*](#_Toc164345686)

[*1.1.2. Văn hóa của công ty và cơ cấu tổ chức 10*](#_Toc164345687)

[*CHƯƠNG 2: GIỚI THIỆU VÀ KHẢO SÁT ĐỀ TÀI 13*](#_Toc164345688)

[*2.1. Tổng quan về đề tài 13*](#_Toc164345689)

[*2.1.1. Lý do chọn đề tài 13*](#_Toc164345690)

[*2.1.2. Mục tiêu của đề tài 14*](#_Toc164345691)

[*2.1.3. Phạm vi của đề tài 14*](#_Toc164345692)

[*2.1.4. Đối tượng nghiên cứu của đề tài 14*](#_Toc164345693)

[*2.2. Khảo sát hệ thống 14*](#_Toc164345694)

[*2.2.1. Khảo sát hiện trạng và yêu cầu đặt ra 14*](#_Toc164345695)

[*2.2.2. Quy trình nghiệp vụ 15*](#_Toc164345696)

[*2.3. Mô hình hóa nghiệp vụ người dùng 17*](#_Toc164345697)

[*2.4. Công cụ thực hiện 17*](#_Toc164345698)

[*2.4.1. MySQL 17*](#_Toc164345699)

[*2.4.2. Spring Boot 18*](#_Toc164345700)

[*2.4.3. Spring Data JPA 19*](#_Toc164345701)

[*2.4.4. Spring Security 20*](#_Toc164345702)

[*2.4.5. Spring Validation và Spring Mail 21*](#_Toc164345703)

[*2.4.6. Postman 22*](#_Toc164345704)

[*2.4.7. Angular 22*](#_Toc164345705)

[*CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 24*](#_Toc164345706)

[*3.1. Giới thiệu về phương pháp phân tích 24*](#_Toc164345707)

[*3.2. Biểu đồ Use Case 25*](#_Toc164345708)

[*3.2.1. Use Case tổng quan của hệ thống 25*](#_Toc164345709)

[*3.2.2. Use Case Đăng nhập 26*](#_Toc164345710)

[*3.2.2. Use Case Xem bài viết 27*](#_Toc164345711)

[*3.2.3. Use Case Tạo bài viết, tạo chủ đề 28*](#_Toc164345712)

[*3.2.4. Use Case Tạo bình luận 29*](#_Toc164345713)

[*3.2.5. Use Case Bình chọn bài viết. 30*](#_Toc164345714)

[*3.3. Biểu đồ thực thể ERD 31*](#_Toc164345715)

[*3.4. Thiết kế cơ sở dữ liệu 32*](#_Toc164345716)

[*3.5. Triển khai Java Backend trên IntelliJ Idea 35*](#_Toc164345717)

[*3.6. Sử dụng Postman để test API được cung cấp từ Back-end 37*](#_Toc164345718)

[*CHƯƠNG 4: TRIỂN KHAI THỬ NGHIỆM VÀ ĐÁNH GIÁ 44*](#_Toc164345719)

[*4.1. Giới thiệu website 44*](#_Toc164345720)

[*1.1.1. Giao diện đăng nhập 44*](#_Toc164345721)

[*1.1.2. Giao diện sau khi đăng nhập thành công 45*](#_Toc164345722)

[*1.1.3. Giao diện đăng kí 45*](#_Toc164345723)

[*1.1.4. Giao diện tạo bài viết 46*](#_Toc164345724)

[*1.1.6. Giao diện chi tiết bài viết và bình luận 47*](#_Toc164345725)

[*4.2. Đánh giá 47*](#_Toc164345726)

[*KẾT LUẬN 49*](#_Toc164345727)

[*TÀI LIỆU THAM KHẢO 50*](#_Toc164345728)

# LỜI MỞ ĐẦU

Trải qua 4 năm học tại trường Đại học Kinh tế Quốc dân, em xin gửi lời tri ân sâu sắc đến quý thầy cô Viện Công nghệ thông tin và Kinh tế số. Nhờ sự tận tâm giảng dạy, truyền đạt kiến thức và kinh nghiệm quý báu của quý thầy cô, em đã có được hành trang vững vàng để sẵn sàng cho con đường phía trước. Đặc biệt, em xin bày tỏ lòng biết ơn sâu sắc đến thầy Đặng Minh Quân - người thầy đã trực tiếp hướng dẫn và đồng hành cùng em trong suốt quá trình thực hiện chuyên đề. Nhờ sự dìu dắt tận tình của thầy, em đã hoàn thành được bài luận một cách trọn vẹn nhất.

Do kinh nghiệm và kiến thức còn hạn chế, chuyên đề tốt nghiệp của em vẫn tồn tại nhiều thiếu sót. Em kính mong nhận được sự nhận xét và chỉ dẫn từ thầy cô để chuyên đề của em được hoàn thiện hơn.

Em xin trân trọng cảm ơn!

# LỜI CAM ĐOAN

Tôi xin cam đoan rằng:

Kết quả nghiên cứu của chuyên đề này là hoàn toàn trung thực hoặc chưa từng được sử dụng và công bố trong bất kỳ văn bản nào khác.

*Hà Nội, ngày 15 tháng 4 năm 2024*

Sinh viên

Đỗ Tuấn Đức

# CÁC THUẬT NGỮ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **STT** | **Thuật ngữ viết tắt** | **Giải thích** |
| 1 | UML | Unified Modeling Language - Ngôn ngữ mô  hình hoá thống nhất |
| 2 | RDBMS | Hệ quản trị Cơ sở dữ liệu quan hệ |
| 3 | CSDL | Cơ sở dữ liệu |
| 4 | CNTT | Công nghệ thông tin |
| 5 | ERD | Entity Relationship Diagram – Sơ đồ quan hệ thực thể |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

# DANH MỤC HÌNH

[Hình 2. 1 Mô hình hóa nghiệp vụ của người đọc 18](#_Toc153290658)

[Hình 2. 2 Mô hình hóa nghiệp vụ của thủ thư 19](#_Toc153290659)

[Hình 2. 3 Entity Framework Core là một thư viện ánh xạ đối tượng - quan hệ (ORM) 20](#_Toc153290660)

[Hình 3. 1 Biểu đồ Use case tổng quan hệ thống 24](#_Toc153290633)

[Hình 3. 2 Biểu đồ Use case đăng nhập 25](#_Toc153290634)

[Hình 3. 3 Biểu đồ Use case đọc sách 26](#_Toc153290635)

[Hình 3. 4 Biểu đồ Use case gửi yêu cầu mượn sách 27](#_Toc153290636)

[Hình 3. 5 Biểu đồ Use case quản lý yêu cầu mượn sách 28](#_Toc153290637)

[Hình 3. 6 Biểu đồ Use case quản lý sách 29](#_Toc153290638)

[Hình 3. 7 Biểu đồ Use case quản lý thể loại sách 30](#_Toc153290639)

[Hình 3. 8 Biểu đồ Use case thống kê 31](#_Toc153290640)

[Hình 3. 9 Biểu đồ hoạt động chức năng đăng nhập 33](#_Toc153290641)

[Hình 3. 10 Biểu đồ hoạt động chức năng đọc sách 34](#_Toc153290642)

[Hình 3. 11 Biểu đồ hoạt động chức năng gửi yêu cầu mượn sách 35](#_Toc153290643)

[Hình 3. 12 Biểu đồ hoạt động chức năng đọc sách 36](#_Toc153290644)

[Hình 3. 13 Biểu đồ hoạt động chức năng quản lý sách 37](#_Toc153290645)

[Hình 3. 14 Biểu đồ hoạt động chức năng quản lý thể loại sách 38](#_Toc153290646)

[Hình 3. 15 Biểu đồ hoạt động chức năng thống kê 39](#_Toc153290647)

[Hình 3. 16 Biểu đồ tuần tự chức năng đăng nhập 40](#_Toc153290648)

[Hình 3. 17 Biểu đồ tuần tự chức năng đọc sách 40](#_Toc153290649)

[Hình 3. 18 Biểu đồ tuần tự chức năng gửi yêu cầu mượn sách 41](#_Toc153290650)

[Hình 3. 19 Biểu đồ tuần tự chức năng quản lý yêu cầu mượn sách 41](#_Toc153290651)

[Hình 3. 20 Biểu đồ tuần tự chức năng quản lý sách 42](#_Toc153290652)

[Hình 3. 21 Biểu đồ tuần tự chức năng quản lý thể loại sách 43](#_Toc153290653)

[Hình 3. 22 Biểu đồ tuần tự chức năng thống kê 43](#_Toc153290654)

[Hình 3. 23 Biểu đồ lớp 44](#_Toc153290655)

[Hình 3. 24 Sơ đồ thiết kế cơ sở dữ liệu 44](#_Toc153290656)

[Hình 4. 1 Giao diện đăng nhập………………………………………………………….…………..47](#_Toc152625527)

[Hình 4. 2 Giao diện thông tin người dùng 47](#_Toc152625528)

[Hình 4. 3 Giao diện đổi mật khẩu 48](#_Toc152625529)

[Hình 4. 4 Giao diện quản lý yêu cầu mượn sách 48](#_Toc152625530)

[Hình 4. 5 Giao diện chi tiết yêu cầu mượn sách 49](#_Toc152625531)

[Hình 4. 6 Giao diện quản lý sách 49](#_Toc152625532)

[Hình 4. 7 Giao diện thông tin người đọc mượn sách 50](#_Toc152625533)

[Hình 4. 8 Giao diện quản lý thể loại sách 50](#_Toc152625534)

[Hình 4. 9 Giao diện thống kê 51](#_Toc152625535)

[Hình 4. 10 Giao diện sách 51](#_Toc152625536)

[Hình 4. 11 Giao diện chi tiết sách 52](#_Toc152625537)

[Hình 4. 12 Giao diện sách đã mượn 52](#_Toc152625538)

[Hình 4. 13 Giao diện đọc sách 53](#_Toc152625539)

# DANH MỤC BẢNG

[Bảng 3. 1 Đặc tả Use case đăng nhập 25](#_Toc2368)

[Bảng 3. 2 Đặc tả Use case đọc sách 26](#_Toc16164)

[Bảng 3. 3 Đặc tả Use case gửi yêu cầu mượn sách 27](#_Toc18931)

[Bảng 3. 4 Đặc tả Use case quản lý yêu cầu mượn sách 28](#_Toc424)

[Bảng 3. 5 Đặc tả Use case quản lý sách 29](#_Toc6024)

[Bảng 3. 6 Đặc tả Use case quản lý thể loại sách 30](#_Toc19327)

[Bảng 3. 7 Đặc tả Use case thống kê 31](#_Toc9597)

[Bảng 3. 8 Bảng User 44](#_Toc9451)

[Bảng 3. 9 Bảng BookBorrowingRequest 45](#_Toc4477)

[Bảng 3. 10 Bảng BookBorrowingRequestDetails 45](#_Toc10902)

[Bảng 3. 11 Bảng Book 45](#_Toc10764)

[Bảng 3. 12 Bảng Category 46](#_Toc21978)

# CHƯƠNG 1: TỔNG QUAN CÔNG TY FPT SOFTWARE

# Tổng quan Công ty FPT SOFTWARE

## ***Thông tin chung về Công ty FPT Software***

FPT Software là tên gọi khác của công ty TNHH Phần Mềm FPT với nhiệm vụ chính là gia công phần mềm tại Việt Nam và nước ngoài.

Thành lập từ năm 1988 đến nay với 3 Trụ sở chính FPT Software đặt tại Việt Nam và một số nước trên toàn thế giới như Hoa Kỳ, Nhật Bản, Malaysia, Đức, Úc, Singapore, Malaysia, Thái Lan và Philipines hiện FPT Software đáp ứng nhu cầu gia công phần mềm lớn cho hơn 150 công ty hàng đầu tại 20 quốc gia lớn nhất trên thế giới hiện nay với các hợp đồng lớn có khi đạt cả 1 triệu USD dành cho một số đối tác lớn như Hitachi, NEOPOST, Petronas, Deutsche Bank, và Unilever với hơn 4000 nhân viên trên toàn thế giới.

FPT Software theo đuổi mục tiêu gia công phần mềm để đáp ứng cho nhu cầu phát triển CNTT của các hãng phần mềm trong nước, các công ty lớn trong nước và tham vọng hơn là xuất khẩu phần mềm trên toàn thế giới cho các công ty nước ngoài biết đến tập đoàn FPT, mục đích chính là vươn đến tầm cao mới thông qua công nghệ nhằm nâng cao năng suất lao động.

Đến với FPT Software, nhân viên được phép phát huy tối đa khả năng sáng tạo, nỗ lực đưa các sáng tạo vào các giải pháp công nghệ thông tin nhằm nắm bắt các cơ hội đổi mới và phát triển nhằm đạt được mục tiêu chung của tập đoàn là OneFPT - Công ty xuất khẩu phần mềm hàng đầu tại Việt Nam.

Kinh doanh của FPT Sofware xuất phát từ Công nghệ thông tin và [Viễn Thông FPT](http://fpt.center/) là cốt lõi để đáp ứng cho khách hàng trên 63 tỉnh thành trên toàn lãnh thổ Việt Nam và mở rộng đến khách hàng toàn cầu. Từ khi được thành lập cho đến nay, FPT Software đã là công ty phần mềm số 1 tại Việt Nam trong lĩnh vực tích hợp hệ thống, phân phối và bán lẻ, dịch vụ Công Nghệ Thông tin, xuất khẩu gia công phần mềm, bán lẻ sản phẩm CNTT.

## ***Văn hóa của công ty và cơ cấu tổ chức***

FPT Software cung cấp môi trường phát triển năng lực tuyệt vời trong đa dạng lĩnh vực: Manufacturing and Automotive, Utilities and Energy, Banking Finance, Securities & Insurance, Telecommunication & Media, Logistics and Transportation, Healthcare... cùng những công nghệ xu hướng: Artificial Intelligence (AI), Low-code, Robotic Process Automation, Cloud, AR/VR, Blockchain, Data Analytics, Enterprise Mobilization, Internet of Things…

FPT Software mang đến trải nghiệm đặc biệt cho nhân viên, đáp ứng nhu cầu và truyền cảm hứng, thúc đẩy tài năng phát triển. Chính sách đãi ngộ của FPT Software không chỉ áp dụng cho nhân viên mà còn mở rộng đến người nhà. Trước đại dịch, công ty hỗ trợ lãi suất mua nhà hay ôtô cho nhân viên, hỗ trợ bảo hiểm cho người thân nhân viên. Trong dịch bệnh, công ty công bố nhiều chính sách đãi ngộ như không áp dụng thời gian thử việc, luôn ký hợp đồng chính thức và ưu đãi 4% lãi suất mua nhà cho nhân viên mới… Những tâm niệm từ người “thuyền trưởng” như truyền lửa cho mỗi cá nhân để tiếp tục nâng cao tiêu chuẩn, rèn luyện bản thân và viết tiếp giấc mơ nâng tầm vị thế của những sản phẩm “Make in Vietnam” trên thị trường quốc tế. Trên chặng đường đó, FPT Software không thể thiếu cánh tay của những người đồng đội cùng chung chí hướng.

Bắt đầu là một Trung tâm phần mềm chiến lược của FPT, năm 1999 FPT Software bắt đầu đẩy mạnh xuất khẩu dịch vụ phần mềm với thực trạng “không kinh nghiệm, không thương hiệu, không đủ người”. Từ 13 thành viên trụ cột ban đầu, năm 2001 FPT Software với sự dẫn dắt của Chủ tịch Trương Gia Bình đã nhận thức được tiềm năng vô cùng lớn từ đất nước "Mặt trời mọc" và chọn Nhật Bản là đất nước mở đầu cho hành trình khai phá tiềm năng toàn cầu hóa của mình. Chính nhờ tầm nhìn xa và sự đồng lòng của những thanh niên trẻ giàu hoài bão khi đó, FPT Software như vận động viên vượt rào lần lượt cán đích các mốc kế hoạch đặt ra trong nhiều năm tiếp theo, khi trở thành công ty phần mềm Việt Nam đầu tiên đạt 200 triệu đô vào năm 2015 và cán mốc 10000 nhân viên vào năm 2016.

Sứ mệnh chung: FPT Software nỗ lực trở thành nhà cung cấp dịch vụ công nghệ thông tin hàng đầu Việt Nam, đồng thời vươn tầm quốc tế, góp phần khẳng định vị thế Việt Nam trên bản đồ công nghệ số toàn cầu. FPT Software tiên phong trong việc hỗ trợ khách hàng chuyển đổi số toàn diện, từ chiến lược đến triển khai, vận hành, góp phần nâng cao hiệu quả hoạt động, gia tăng sức cạnh tranh cho khách hàng.

Tầm hình: FPT Software hướng đến trở thành nhà cung cấp dịch vụ công nghệ thông tin hàng đầu Việt Nam và khu vực, đồng thời vươn tầm quốc tế, góp phần khẳng định vị thế Việt Nam trên bản đồ công nghệ số toàn cầu. FPT Software cam kết mang đến cho khách hàng những giải pháp và dịch vụ công nghệ thông tin tiên tiến, chất lượng cao, góp phần nâng cao hiệu quả hoạt động, gia tăng sức cạnh tranh cho khách hàng. FPT Software luôn đặt con người làm trung tâm, xây dựng môi trường làm việc chuyên nghiệp, sáng tạo, thu hút và phát triển nhân tài.

Văn hóa: FPT Software có văn hóa doanh nghiệp độc đáo, lấy con người làm trọng tâm, với những giá trị cốt lõi:

- Chí công vô tư: Luôn đặt lợi ích chung của công ty và khách hàng lên hàng đầu.

- Hợp tác đồng đội: Làm việc hiệu quả theo nhóm, hỗ trợ lẫn nhau cùng phát triển.

- Cầu tiến học hỏi: Luôn học hỏi, trau dồi kiến thức và kỹ năng để phát triển bản thân.

- Sáng tạo đổi mới: Dám nghĩ dám làm, không ngừng sáng tạo và đổi mới.

- Trách nhiệm: Luôn hoàn thành tốt công việc được giao, chịu trách nhiệm cho hành động của bản thân.

FPT Software còn có nhiều chương trình văn hóa, thể thao và giải trí phong phú, góp phần tạo môi trường làm việc năng động, sáng tạo và gắn kết cho cán bộ nhân viên. Ngoài ra, FPT Software còn chú trọng phát triển nguồn nhân lực, xây dựng đội ngũ cán bộ nhân viên có trình độ chuyên môn cao, đạo đức nghề nghiệp tốt, đáp ứng yêu cầu ngày càng cao của thị trường. Nhờ có sứ mệnh, tầm hình và văn hóa rõ ràng, FPT Software đã đạt được nhiều thành công trong những năm qua, trở thành một trong những công ty công nghệ thông tin hàng đầu Việt Nam và khu vực.

# CHƯƠNG 2: GIỚI THIỆU VÀ KHẢO SÁT ĐỀ TÀI

# 2.1. Tổng quan về đề tài

## ***2.1.1. Lý do chọn đề tài***

Trong bối cảnh kinh tế toàn cầu vẫn đang phải đối mặt với sự không chắc chắn và diễn biến phức tạp, việc chuyển đổi số và chuyển đổi xanh đang trở thành xu hướng không thể phủ nhận. Tại Việt Nam, mục tiêu của Chính phủ là tạo ra một nền kinh tế bền vững hơn thông qua việc đẩy mạnh chuyển đổi số và tận dụng tối đa tiềm năng của mạng lưới internet và người dùng di động đang gia tăng. Dân số trẻ và sự phát triển nhanh chóng của công nghệ thông tin ở Việt Nam mở ra cơ hội lớn cho việc phát triển các dịch vụ số như thương mại điện tử, giáo dục trực tuyến, y tế số và quản lý trực tuyến... Chuyển đổi số tại Việt Nam đang đối mặt với nhiều cơ hội cũng như thách thức. Với lợi thế là một thị trường có dân số đông và chủ yếu là đối tượng trẻ, Việt Nam có hơn 70 triệu người dùng internet, 143 triệu thuê bao di động và 64 triệu người dùng mạng xã hội. Đây là một lợi thế để phát triển các dịch vụ số, thương mại điện tử, giáo dục trực tuyến, y tế số, quản lý online v.v. Tuy nhiên, việc xây dựng và duy trì hạ tầng công nghệ thông tin, đảm bảo an ninh mạng và bảo vệ dữ liệu cá nhân vẫn là những thách thức đối với quá trình chuyển đổi số tại Việt Nam.

Một trong những nhiệm vụ quan trọng của việc chuyển đổi số là đáp ứng nhu cầu của xã hội hiện đại. Đó cũng là lý do trang cung cấp thông tin Reddit Việt Nam ra đời. Đây là một hình thức cung cấp thông tin về tất cả những chủ đề mà người dùng muốn cho công chúng thông qua các phương tiện điện tử, như internet, máy tính, điện thoại thông minh, máy tính bảng, v.v. Trang thông tin có nhiều ưu điểm bởi vì có thể cung cấp thông tin từ nhiều nguồn và có thể nắm bắt được thông tin. Tuy vậy, tại Việt Nam, các trang chia sẻ thông tin chưa thực sự đáp ứng được sự kì vọng của người sử dụng.

Nhận thấy thực trạng đó, trong quá trình thực tập và làm việc, tác giả đã có những nghiên cứu để xây dựng một website cho phép người dùng truy cập vào trang thông tin một cách dễ dàng, thuận tiện. Từ đó nâng cao khả năng tiếp cận thông tin của người sử dụng. Đây là một giải pháp giúp cho người dùng Internet tại Việt Nam có những thông tin và chủ đề để bàn luận một cách thuận tiện hơn. Xuất phát từ những lý do trên, bản thân tác giả được đào tạo và có những kiến thức cơ bản về Công nghệ thông tin, qua thời gian thực tập và làm việc tại công ty, tác giả quyết định lựa chọn đề tài “Xây dựng website chia sẻ thông tin Reddit Việt Nam”.

## ***2.1.2. Mục tiêu của đề tài***

*Mục tiêu tổng thể*

Nghiên cứu, thiết kế và xây dựng một website chia sẻ thông tin, cho phép người dùng truy cập để thảo luận về một chủ đề cụ thể một cách dễ dàng, thuận tiện, từ đó nâng cao khả năng tiếp cận thông tin và bàn luận vấn đề.

*Mục tiêu chi tiết*

Thiết kế giao diện, cơ sở dữ liệu, chức năng và bảo mật của website.

Lập trình, kiểm thử và triển khai website trên môi trường thực tế.

Đánh giá hiệu quả, ưu nhược điểm và đề xuất hướng phát triển.

## ***2.1.3. Phạm vi của đề tài***

- Các chức năng của website: Website được sử dụng cho đối tượng chính là người người sử dụng website. Chỉ khi người dùng đăng nhập vào website thì mới xem được các thông tin về các bài viết, bình luận, thời gian đăng, ...

- Các yếu tố kỹ thuật của website: Sử dụng các công nghệ, ngôn ngữ, thư viện và hệ quản trị cơ sở dữ liệu phù hợp để xây dựng, thiết kế, phát triển, kiểm thử, bảo trì và cải tiến website, đảm bảo tính năng, hiệu năng, bảo mật, khả năng mở rộng và thân thiện với người dùng.

## ***2.1.4. Đối tượng nghiên cứu của đề tài***

Người dùng: Là những người có nhu cầu tiếp nhận, trao đổi, truyền đạt thông tin về các chủ đề khác nhau. Người dùng cũng là người tạo chủ đề để những người dùng khác bàn luận với nhau.

# 2.2. Khảo sát hệ thống

## ***2.2.1. Khảo sát hiện trạng và yêu cầu đặt ra***

*Khảo sát hiện trạng*

Sau khi nghiên cứu và tìm hiểu về các nền tảng liên quan đến hội nhóm, cộng đồng hiện nay, tác giả đưa ra những nhận định cơ bản như sau:

- Chưa có một hệ thống có thế đáp ứng được nhu cầu của các hội nhóm, các chủ đề và bài viết chưa được rõ ràng. Gây ra khó khăn cho người dùng khi tìm kiếm các bài viết liên quan đến chủ đề. Theo tham khảo, cộng đồng người sử dụng Reddit tại Việt Nam khá lớn. Việc sử dụng trang Reddit của nước ngoài sẽ có những bất lợi về ngôn ngữ. Điều ấy khiến cho người Việt không thể hiểu hết được những thông tin liên quan đến cá nhân hay bảo mật cá nhân.

- Đã có những Website nhưng đều về những chủ đề cụ thể ví dụ như thảo luận về công ty, thảo luận về lương thưởng, thảo luận về trò chơi điện tử… Chưa thực sự có một Website có thể bao quát tổng thể tất cả các thông tin.

- Các cộng đồng chưa liên kết với nhau. Với những người dùng có sở thích khác nhau sẽ phải truy cập nhiều trang web hay nhiều cộng đồng để thảo luận.

- Gặp nhiều khó khăn trong việc xây dựng và vận hành hạ tầng công nghệ thông tin, đảm bảo an ninh mạng và bảo vệ dữ liệu cá nhân.

*Yêu cầu đặt ra*

Dựa trên tình hình hiện trạng, tác giả rút ra một số yêu cầu cần phải đáp ứng của đề tài như sau:

- Yêu cầu chức năng: Cho phép người dùng tạo chủ đề để đăng các bài viết. Đối với các bài viết của mỗi người dùng, các người dùng khác có thể thực hiện bình luận về bài viết, bày tỏ thái độ đối với bài viết. Người dùng có thể theo dõi các thông tin liên quan đến số lượng bài đăng và bình luận mình đã đăng tải…

- Yêu cầu phi chức năng: Đảm bảo tính năng và hiệu suất tối ưu, bảo mật, khả năng mở rộng. Bất kì người dùng nào cũng có thể sử dụng dễ dàng.

- Yêu cầu giao diện: giao diện thân thiện, tối giản và dễ sử dụng cũng như quản lý để có thể tiếp cận đến nhiều đối tượng mục tiêu, kể cả những đối tượng có kỹ năng số cơ bản hay kỹ năng số nâng cao và chuyên sâu.

## ***2.2.2. Quy trình nghiệp vụ***

- Khảo sát bài toán: Là bước đầu tiên trong quy trình, nhằm hiểu rõ nhu cầu và yêu cầu về website, bao gồm các thông tin về mục tiêu, đối tượng, chức năng, bài toán, v.v. Bước này sử dụng phương pháp như khảo sát, quan sát, v.v. để thu thập dữ liệu.

- Thiết kế website: Là bước tiếp theo, nhằm xây dựng một bản thiết kế cho website, bao gồm các thành phần như cơ sở dữ liệu, chức năng và bảo mật, giao diện. Bước này sử dụng các công cụ như bản vẽ, sơ đồ, mô hình, v.v. để mô tả các tầng của website như tầng dữ liệu, tầng thực thi, tầng cho người sử dụng.

- Lập trình website: Là bước thực hiện, nhằm biến bản thiết kế thành một website hoạt động, bao gồm các công việc như viết mã, nhập dữ liệu, cài đặt phần mềm, v.v. Bước này sử dụng các ngôn ngữ lập trình, cơ sở dữ liệu, giao diện, bảo mật để phát hành Website.

- Thử nghiệm và đánh giá Website: Là bước cuối cùng, nhằm kiểm thử Website sau đó đưa ra những đánh giá ưu, nhược điểm của Website và đề xuất hướng phát triển tương lai của đề tài.

# 2.3. Mô hình hóa nghiệp vụ người dùng

A diagram of a diagram

Description automatically generated

*Hình 2.1. Mô hình hóa nghiệp vụ người dùng*

# 2.4. Công cụ thực hiện

## ***2.4.1. MySQL***

MySQL là một hệ quản trị cơ sở dữ liệu quan hệ mã nguồn mở (RDBMS) phổ biến nhất thế giới. Nó được sử dụng cho nhiều ứng dụng khác nhau, từ các trang web nhỏ đến các doanh nghiệp lớn. MySQL được biết đến với độ tin cậy, hiệu suất và khả năng mở rộng cao. Nó cũng tương đối dễ sử dụng và cài đặt, khiến nó trở thành một lựa chọn phổ biến cho người mới bắt đầu và các nhà phát triển có kinh nghiệm.

MySQL được sử dụng cho nhiều ứng dụng khác nhau, bao gồm:

- Trang web: MySQL là một lựa chọn phổ biến cho các trang web vì nó có thể xử lý lượng truy cập cao và lưu trữ nhiều dữ liệu.

- Ứng dụng web: MySQL được sử dụng rộng rãi cho các ứng dụng web, chẳng hạn như thương mại điện tử, mạng xã hội và ứng dụng di động.

- Phần mềm doanh nghiệp: MySQL được sử dụng cho nhiều phần mềm doanh nghiệp, chẳng hạn như hệ thống quản lý quan hệ khách hàng (CRM), hệ thống quản lý tài nguyên doanh nghiệp (ERP) và hệ thống quản lý nội dung (CMS).

- Ứng dụng khoa học dữ liệu: MySQL được sử dụng cho các ứng dụng khoa học dữ liệu để lưu trữ và phân tích dữ liệu lớn.

## ***2.4.2. Spring Boot***

Spring Boot là một framework mã nguồn mở dựa trên nền tảng Spring giúp phát triển ứng dụng Java dễ dàng và nhanh chóng hơn. Nó cung cấp nhiều tính năng hữu ích giúp đơn giản hóa quá trình cấu hình, triển khai và quản lý ứng dụng Spring.

Ưu điểm của Spring Boot:

- Đơn giản hóa: Spring Boot giúp đơn giản hóa quá trình phát triển ứng dụng Spring bằng cách loại bỏ nhiều cấu hình boilerplate.

- Nhanh chóng: Spring Boot giúp bạn tạo ứng dụng Spring nhanh chóng hơn bằng cách cung cấp các starter dependency và tính năng khởi động tự động.

- Dễ sử dụng: Spring Boot dễ sử dụng và dễ học đối với các nhà phát triển Java.

- Có thể mở rộng: Spring Boot có thể mở rộng để đáp ứng nhu cầu của các ứng dụng lớn và phức tạp.

- Hỗ trợ đa nền tảng: Spring Boot có thể chạy trên nhiều nền tảng khác nhau, bao gồm Linux, Windows và macOS.

Tính năng chính của Spring Boot:

- Khởi động tự động: Spring Boot có thể tự động khởi động ứng dụng của bạn mà không cần cấu hình thủ công.

- Starter dependency: Spring Boot cung cấp các starter dependency giúp bạn dễ dàng thêm các thư viện và tính năng phổ biến vào ứng dụng của mình.

- Embedded servers: Spring Boot có thể nhúng các server web như Tomcat, Jetty hoặc Undertow vào ứng dụng của bạn.

- Actuator: Spring Boot cung cấp Actuator, một bộ công cụ giúp bạn giám sát và quản lý ứng dụng đang chạy.

- Spring Cloud: Spring Boot có thể được sử dụng kết hợp với Spring Cloud để xây dựng các ứng dụng microservices.

Để sử dụng Spring Boot, bạn cần cài đặt Java Development Kit (JDK) và Maven hoặc Gradle. Sau đó, tạo một dự án Spring Boot mới bằng cách sử dụng Spring Initializr hoặc công cụ dòng lệnh Spring Boot.

## ***2.4.3. Spring Data JPA***

Spring Data JPA là một phần quan trọng trong hệ sinh thái Spring, giúp đơn giản hóa việc truy cập và quản lý dữ liệu trong các ứng dụng Java. Nó cung cấp một lớp trừu tượng trên Hibernate, giúp bạn thao tác với các đối tượng Java thay vì viết SQL trực tiếp.

Tính năng chính:

- Lớp Repository: Cung cấp các phương thức tiện lợi để truy cập dữ liệu như findById(), findAll(), save(), delete(), v.v. Hỗ trợ truy vấn JPQL: Cho phép bạn viết truy vấn truy cập dữ liệu bằng ngôn ngữ JPQL (Java Persistence Query Language) thay vì SQL.

- Tính năng tự động tạo: Tự động tạo các truy vấn và câu lệnh SQL dựa trên cấu trúc của lớp Entity và tên phương thức.

- Hỗ trợ Paging và Sorting: Cung cấp các phương thức tiện lợi để phân trang và sắp xếp dữ liệu.

- Hỗ trợ Transactions: Quản lý transactions một cách tự động.

- Tích hợp với Spring Boot: Dễ dàng tích hợp với Spring Boot để tạo ra các ứng dụng web RESTful.

Các lợi ích cảu Spring Data JPA:

- Giảm thiểu boilerplate code: Viết ít code hơn, tập trung vào logic nghiệp vụ.

- Tăng cường bảo trì: Code dễ đọc, dễ hiểu, dễ bảo trì.

- Tăng cường tính an toàn: Giảm thiểu lỗi SQL và lỗi truy cập dữ liệu.

- Tăng hiệu suất: Tối ưu hóa truy vấn và truy cập dữ liệu.

Muốn sử dụng Spring Data JPA, chúng ta cần thêm thư viện Spring Data JPA. Thêm thư viện spring-data-jpa vào classpath của project. Tạo các lớp thực thể để biểu diễn các bản ghi trong database. Sau đó tạo các interface Repository để khai báo các phương thức truy cập dữ liệu. Cuối cùng, cấu hình datasource để kết nối với database.

## ***2.4.4. Spring Security***

Spring Security là một framework bảo mật cấp doanh nghiệp được xây dựng trên nền tảng Spring Framework cho Java. Nó cung cấp các tính năng xác thực, ủy quyền, bảo mật ứng dụng web và quản lý truy cập nguồn lực. Spring Security được sử dụng rộng rãi trong các ứng dụng Java để bảo vệ dữ liệu và tài nguyên khỏi truy cập trái phép.

Các tính năng chính của Spring Security:

- Xác thực: Xác minh danh tính người dùng bằng cách sử dụng tên người dùng và mật khẩu, chứng chỉ SSL, token JWT hoặc các cơ chế khác.

- Ủy quyền: Kiểm soát quyền truy cập của người dùng vào các tài nguyên của ứng dụng dựa trên vai trò và quyền hạn của họ.

- Bảo mật ứng dụng web: Bảo vệ ứng dụng web khỏi các cuộc tấn công phổ biến như tấn công chèn mã SQL, tấn công site-cross scripting (XSS) và tấn công giả mạo yêu cầu trang web (CSRF).

- Quản lý truy cập nguồn lực: Kiểm soát quyền truy cập vào các tài nguyên hệ thống như tệp, thư mục và cơ sở dữ liệu.

Lợi ích của việc sử dụng Spring Security:

- Cải thiện bảo mật: Spring Security giúp bảo vệ ứng dụng khỏi các mối đe dọa bảo mật phổ biến.

- Giảm thiểu rủi ro: Spring Security giúp giảm thiểu rủi ro vi phạm dữ liệu và các vấn đề bảo mật khác.

- Tăng năng suất: Spring Security giúp bạn tiết kiệm thời gian và công sức bằng cách cung cấp các tính năng bảo mật được xây dựng sẵn.

- Dễ sử dụng: Spring Security dễ học và sử dụng, ngay cả đối với những người mới bắt đầu.

- Có thể mở rộng: Spring Security có thể được mở rộng để đáp ứng các nhu cầu bảo mật cụ thể của bạn.

## ***2.4.5. Spring Validation và Spring Mail***

Spring Validation là một framework mã nguồn mở được sử dụng để xác thực dữ liệu đầu vào trong các ứng dụng Java. Nó cung cấp các chú thích và API để giúp dễ dàng xác minh xem dữ liệu đầu vào có hợp lệ hay không. Spring Validation được tích hợp tốt với Spring Framework, giúp bạn dễ dàng sử dụng nó trong các ứng dụng web của mình.

Các tính năng chính của Spring Validation:

- Chú thích xác thực: Spring Validation cung cấp các chú thích để xác minh các loại dữ liệu khác nhau, chẳng hạn như @NotEmpty, @NotNull, @Size, @Email và @Pattern.

- Trình xác thực tùy chỉnh: Có thể tạo trình xác thực tùy chỉnh để xác minh các quy tắc xác thực phức tạp.

- Hỗ trợ nhóm xác thực: Có thể nhóm các quy tắc xác thực lại với nhau để dễ dàng sử dụng lại.

- Hỗ trợ getMessageResolver: Có thể tùy chỉnh thông báo lỗi xác thực.

Spring Mail là một framework mã nguồn mở được sử dụng để gửi email trong các ứng dụng Java. Nó cung cấp các lớp và API để giúp bạn dễ dàng gửi email đơn giản và phức tạp. Spring Mail được tích hợp tốt với Spring Framework, giúp bạn dễ dàng sử dụng nó trong các ứng dụng web của mình.

## ***2.4.6. Postman***

Công cụ toàn diện cho API Postman là một nền tảng phát triển và quản lý API được sử dụng rộng rãi, cung cấp nhiều tính năng mạnh mẽ để hỗ trợ lập trình viên và chuyên gia API trong suốt vòng đời phát triển API. Một số đặc điểm chính của API như sau:

- Giao diện thân thiện: Postman sở hữu giao diện trực quan, dễ sử dụng, cho phép người dùng dễ dàng tạo, gửi và kiểm tra các yêu cầu API.

- Hỗ trợ nhiều phương thức HTTP: Postman hỗ trợ đầy đủ các phương thức HTTP phổ biến như GET, POST, PUT, PATCH, DELETE, giúp người dùng thực hiện các thao tác CRUD (Tạo, Đọc, Cập nhật, Xóa) với dữ liệu API.

- Tạo và quản lý Collections: Postman cho phép người dùng tạo và quản lý các Collections, nơi lưu trữ các yêu cầu API được nhóm theo chức năng hoặc dự án.

- Thử nghiệm và gỡ lỗi API: Postman cung cấp các công cụ mạnh mẽ để thử nghiệm và gỡ lỗi API, bao gồm khả năng kiểm tra trạng thái phản hồi, xem nhật ký yêu cầu và phản hồi, và xác định lỗi.

- Môi trường: Postman hỗ trợ nhiều môi trường, cho phép người dùng mô phỏng hành vi của API trong các môi trường khác nhau (ví dụ: phát triển, thử nghiệm, sản xuất).

- Biến: Postman hỗ trợ sử dụng biến để lưu trữ và truy cập các giá trị thường xuyên sử dụng, giúp đơn giản hóa các yêu cầu API phức tạp.

- Mở rộng: Postman có hệ sinh thái mở rộng với nhiều plugin và thư viện, cung cấp các tính năng bổ sung và tích hợp với các công cụ và dịch vụ khác.

- Hỗ trợ cộng đồng: Postman sở hữu cộng đồng người dùng lớn và tích cực, cung cấp nhiều tài liệu, hướng dẫn và hỗ trợ trực tuyến.

## ***2.4.7. Angular***

Angular là một framework mã nguồn mở miễn phí được sử dụng để xây dựng các ứng dụng web hiện đại. Nó được phát triển bởi Google và được sử dụng rộng rãi bởi các công ty lớn như Forbes, General Motors và Upwork. Angular là framework front-end dựa trên TypeScript, cung cấp nhiều tính năng để giúp xây dựng các ứng dụng web mạnh mẽ và có thể mở rộng. Một số tính năng chính của Angular bao gồm:

- Kiến trúc MVC: Angular sử dụng mô hình kiến trúc Model-View-Controller (MVC) để giúp bạn phân tách mã của mình thành các phần riêng biệt, dễ quản lý hơn.

- Liên kết dữ liệu: Angular cung cấp tính năng liên kết dữ liệu mạnh mẽ giúp bạn dễ dàng giữ cho giao diện người dùng của mình được cập nhật với dữ liệu của ứng dụng.

- Thành phần: Angular sử dụng các thành phần để xây dựng các khối xây dựng giao diện người dùng có thể tái sử dụng được.

# CHƯƠNG 3: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

# 3.1. Giới thiệu về phương pháp phân tích

Ngôn ngữ Mô hình hóa Thống nhất (UML) nổi lên như một công cụ đắc lực trong lĩnh vực phân tích thiết kế hệ thống, được đánh giá cao bởi khả năng mô hình hóa trực quan và hiệu quả. UML được xem như tiêu chuẩn cho việc phát triển phần mềm và hệ thống nhờ những ưu điểm sau:

- Tạo mô hình trực quan: UML sử dụng các biểu đồ, ký hiệu sinh động để mô tả hệ thống, giúp người dùng dễ dàng nắm bắt cấu trúc, chức năng và mối quan hệ giữa các thành phần. Nhờ vậy, việc quản lý độ phức tạp của hệ thống trở nên đơn giản và hiệu quả hơn.

- Tập trung vào các khía cạnh quan trọng: UML cho phép người dùng tập trung vào những khía cạnh thiết yếu trong quá trình phân tích và thiết kế hệ thống, loại bỏ những chi tiết rườm rà, không cần thiết. Nhờ vậy, việc mô hình hóa trở nên rõ ràng và chính xác hơn.

- Mô hình hóa trừu tượng: UML hỗ trợ mô hình hóa hệ thống một cách trừu tượng, tách biệt khỏi các chi tiết kỹ thuật cụ thể. Điều này giúp người dùng dễ dàng hình dung bức tranh toàn cảnh của hệ thống và đưa ra những quyết định thiết kế sáng suốt.

Phiên bản UML hiện hành là UML 2.0, cung cấp đa dạng các loại mô hình và biểu đồ để đáp ứng nhu cầu linh hoạt trong quá trình phân tích hệ thống. Một số biểu đồ phổ biến bao gồm:

- Biểu đồ Trường hợp sử dụng (Use Case): Mô tả tương tác giữa người dùng và hệ thống hoặc giữa các hệ thống với nhau.

- Biểu đồ Hoạt động (Activity): Thể hiện các hành động tuần tự hoặc song song diễn ra trong hệ thống.

- Biểu đồ Trình tự (Sequence): Mô tả sự tương tác giữa các đối tượng và thứ tự thực hiện tương tác.

- Biểu đồ Lớp (Class): Thể hiện các lớp, giao diện, kiểu và mối quan hệ giữa chúng trong hệ thống. Nhờ những ưu điểm nổi bật, UML đóng vai trò quan trọng trong việc tối ưu hóa quá trình phân tích thiết kế hệ thống, giúp tạo ra những hệ thống phần mềm hiệu quả, đáp ứng nhu cầu người dùng và mục tiêu đề ra.

# 3.2. Biểu đồ Use Case

## ***3.2.1. Use Case tổng quan của hệ thống***

**A blue background with black text

Description automatically generated**

*Hình 3.1. Biểu đồ Use case tổng quan hệ thống.*

## ***3.2.2. Use Case Đăng nhập***

**A diagram of a diagram

Description automatically generated**

*Hình 3.2. Biểu đồ Use case đăng nhập.*

Bảng 3.1. Đặc tả Use case đăng nhập.

|  |  |  |
| --- | --- | --- |
| Tiêu đề | | Nội dung |
| Tên Usecase | | Đăng nhập |
| Tác nhân | | Người dùng Website |
| Mô tả | | Cho phép người dùng đăng nhập vào hệ thống để thực hiện những chức năng của Website |
| Sự kiện kích hoạt | | Nhấn nút “Đăng nhập” trên màn hình. |
| Tiền điều kiện | | Người dùng phải có tài khoản đã được đăng kí và có trong database |
| Hậu điều kiện | Thành công | Nếu thành công thì sẽ chuyển đến giao diện ứng dụng sau đăng nhập. |
| Lỗi | Nếu thông tin đăng nhập không đúng, backend sẽ trả về thông báo không thành công. |
| Luồng sự kiện chính | | 1. Hệ thống hiển thị giao diện đăng nhập.  2. Người dùng nhập tài khoản và mật khẩu.  3. Hệ thống gửi thông tin đến Backend để thực hiện kiểm tra thông tin đăng nhập.  4. Kết thúc chức năng. |
| Luồng sự kiện khác | | N/A |

## ***3.2.2. Use Case Xem bài viết***

A diagram of a person with a blue oval

Description automatically generated

*Hình 3.3 Biểu đồ Use case Xem bài viết, xem chủ đề.*

Bảng 3. 2 Đặc tả Use case Xem bài viết, xem chủ đề.

|  |  |  |
| --- | --- | --- |
| Tiêu đề | | Nội dung |
| Tên Usecase | | Xem bài viết, chủ đề |
| Tác nhân | | Người dùng Website |
| Mô tả | | Cho phép người dùng sau khi đăng nhập vào hệ thống có thể xem các bài viết và các chủ đề ở trên Website |
| Sự kiện kích hoạt | |  |
| Tiền điều kiện | | Người dùng đã đăng nhập được vào Website |
| Hậu điều kiện | Thành công | Backend trả về các bài viết đã được đăng tải  Backend trả về các chủ đề đã được tạo |
| Lỗi | Không gửi được dữ liệu đến Back-end và báo lỗi |
| Luồng sự kiện chính | | 1. Hệ thống gửi thông tin đến Backend để lấy dữ liệu về bài viết và chủ đề cho Front-end.  2. Sau khi người dùng đăng nhập sẽ hiển thị các bài viết và các chủ đề  4. Kết thúc chức năng. |
| Luồng sự kiện khác | | N/A |

## ***3.2.3. Use Case Tạo bài viết, tạo chủ đề***

**A blue oval with black text

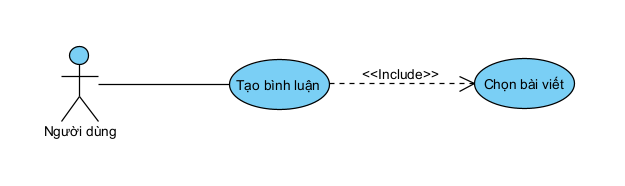
Description automatically generated**

*Hình 3.4. Biểu đồ Use case Tạo bài viết, tạo chủ đề.*

Bảng 3.3. Đặc tả Use case *Tạo bài viết, tạo chủ đề.*

|  |  |  |
| --- | --- | --- |
| Tiêu đề | | Nội dung |
| Tên Usecase | | Tạo bài viết, tạo chủ đề |
| Tác nhân | | Người dùng Website |
| Mô tả | | Cho phép người dùng tạo các chủ đề để bàn luận. Sau khi tạo chủ đề, người dùng sẽ tạo bài viết và chọn chủ đề để đăng tải. |
| Sự kiện kích hoạt | | Tạo chủ đề, tạo bài viết |
| Tiền điều kiện | | Người dùng đã đăng nhập |
| Hậu điều kiện | Thành công | Tạo chủ đề thành công  Tạo bài viết theo chủ đề đã tạo thành công |
| Lỗi | Không gửi được dữ liệu đến Back-end và báo lỗi |
| Luồng sự kiện chính | | 1. Người dùng chọn nút tạo chủ đề trên Website 2. Người dùng cung cấp các thông tin liên quan đến chủ đề 3. Người dùng ấn tạo để tạo chủ đề hoặc chọn hủy không tạo chủ đề nữa. 4. Sau khi người dùng tạo chủ đề xong sẽ chọn tạo bài viết 5. Người dùng cung cấp các thông tin của bài viết và chọn chủ đề phù hợp 6. Người dùng ấn tạo bài viết 7. Kết thúc chức năng. |
| Luồng sự kiện khác | | N/A |

## ***3.2.4. Use Case Tạo bình luận***

****

*Hình 3.5. Biểu đồ Use case Tạo bình luận.*

Bảng 3.4. Đặc tả Use case Tạo bình luận.

|  |  |  |
| --- | --- | --- |
| Tiêu đề | | Nội dung |
| Tên Usecase | | Tạo bình luận |
| Tác nhân | | Người dùng |
| Mô tả | | Cho phép người dùng tạo bình luận ở trên bài viết đã được đăng tải |
| Sự kiện kích hoạt | | Chọn bình luận ở bài đăng |
| Tiền điều kiện | | Người dùng đã đăng nhập |
| Hậu điều kiện | Thành công | Bình luận được đăng tải vào bài viết đã chọn |
| Lỗi | Không gửi được dữ liệu đến Back-end và báo lỗi |
| Luồng sự kiện chính | | 1. Người dùng ấn vào nút bình luận của một bài viết bất kì trên Website 2. Hệ thống cung cấp tất cả các bình luận đã được đăng ở bài viết đó 3. Người dùng nhập bình luận 4. Chọn bình luận để gửi bình luận vào bài viết 5. Kết thúc chức năng |
| Luồng sự kiện khác | | N/A |

## ***3.2.5. Use Case Bình chọn bài viết.***

**A blue oval with black text

Description automatically generated**

*Hình 3.6. Biểu đồ Use case Bình chọn bài viết.*

Bảng 3.5. Đặc tả Use case Bình chọn bài viết.

|  |  |  |
| --- | --- | --- |
| Tiêu đề | | Nội dung |
| Tên Usecase | | Binh chọn bài viết |
| Tác nhân | | Người dùng |
| Mô tả | | Cho phép người dùng bình chọn ủng hộ hoặc không ủng hộ bài viết |
| Sự kiện kích hoạt | | Vào trang chủ trên Website |
| Tiền điều kiện | | Người dùng đăng nhập vào Website |
| Hậu điều kiện | Thành công | Người dùng bình chọn thành công và số lượng bình chọn sẽ được cập nhật |
| Lỗi | Không gửi được dữ liệu đến Back-end và báo lỗi |
| Luồng sự kiện chính | | 1. Người dùng truy cập trang chủ trên website. 2. Hệ thống hiện thị tất cả các bài viết đã đăng tải 3. Người dùng chọn biểu tượng mũi tên hướng lên (ủng hộ) hoặc mũi tên hướng xuống (không ủng hộ) 4. Hệ thống cập nhật thông tin về lượng ủng hộ và không ủng hộ. Với lượt ủng hộ thì được +1, lượt không ủng hộ sẽ -1 5. Hệ thống lưu thông tin vào cơ sở dữ liệu 6. Kết thúc chức năng |
| Luồng sự kiện khác | |  |

# 3.3. Biểu đồ thực thể ERD

Dựa trên các Use Case đã phân tích và các tác nhân. Em đưa ra sơ đồ thực thể như sau:

A diagram of a flowchart

Description automatically generated

*Hình 3.7. Sơ đồ ERD của hệ thống*

# 3.4. Thiết kế cơ sở dữ liệu

A computer screen shot of a computer

Description automatically generatedVới biểu đồ đã phân tích ở bên trên, tác giả đưa ra các thông tin về cơ sở dữ liệu và các thông tin cần thiết cho cơ sở dữ liệu như sau:

*Hình 3.8. Cơ sỏ dữ liệu của hệ thống.*

Mô tả của các bảng trong cơ sở dữ liệu như sau:

* **User**: Lưu dữ liệu người dùng trên hệ thống

Bảng 3.6. Bảng User

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| user\_id | BIGINT | Mã tài khoản của người dùng |
| username | VARCHAR | Tên đăng nhập của người dùng |
| password | VARCHAR | Mật khẩu của người dùng |
| email | VARCHAR | Email của người dùng |
| createdDate | DATETIME | Ngày tạo tài khoản của người dùng |
| enable | BIT | Cờ đánh dấu kích hoạt tài khoản |

* **Post**: Lưu dữ liệu bài viết trên hệ thống

Bảng 3.7. Bảng Post

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| post\_id | BIGINT | Mã bài viết |
| description | TINYTEXT | Mô tả bài viết |
| post\_name | VARCHAR | Tên bài viết |
| url | VARCHAR | Đường dẫn |
| voteCount | DATETIME | Tổng số lượng bình chọn |
| id | BIGINT | Mã chủ đề |
| user\_id | BIGINT | Mã người dùng |
| created\_date | DATETIME | Ngày tạo bài viết |

* **Comment**: Lưu dữ liệu bình luận trên hệ thống

Bảng 3.8. Bảng Comment

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| id | BIGINT | Mã bình luận |
| text | VARCHAR | Mô tả bình luận |
| post\_id | BIGINT | Mã bài viết |
| user\_id | BIGINT | Mã người dùng |
| created\_date | DATETIME | Ngày tạo bình luận |

* **Subreddit**: Lưu dữ liệu chủ đề trên hệ thống

Bảng 3.9. Bảng Subreddit

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| id | BIGINT | Mã chủ đề |
| description | VARCHAR | Mô tả chủ đề |
| name | VARCHAR | Tên chủ đề |
| user\_user\_id | BIGINT | Mã người dùng |
| created\_date | DATETIME | Ngày tạo chủ đề |

* **Subreddit\_post**: bảng sinh ra dựa trên quan hệ giữa Subreddit và Post trên hệ thống

Bảng 3.10. Bảng Subreddit\_post

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| subreddit\_id | BIGINT | Mã chủ đề |
| post\_post\_id | BIGINT | Mã bài viết |

* **Vote**: Lưu thông tin bình chọn trên hệ thống

Bảng 3.11. Bảng Vote

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| vote\_id | BIGINT | Mã chủ đề |
| vote\_type | SMALLINT | Loại bình chọn |
| post\_id | BIGINT | Mã bài viết |
| user\_id | BIGINT | Mã người dùng |

* **Token**: Lưu thông tin token để kích hoạt tài khoản trên hệ thống

Bảng 3.12. Bảng Vote

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| id | BIGINT | Mã |
| expiry\_date | DATETIME | Ngày hết hạn |
| token | VARCHAR | Mã token |
| user\_user\_id | BIGINT | Mã người dùng |

* **RefreshToken**: Lưu thông tin refresh token trên hệ thống

Bảng 3.13. Bảng Refresh\_Token

|  |  |  |
| --- | --- | --- |
| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
| id | BIGINT | Mã |
| created\_date | DATETIME | Ngày tạo |
| token | VARCHAR | Mã token |

# 3.5. Triển khai Java Backend trên IntelliJ Idea

***3.5.1. Xây dựng gói Model:***

Dựa trên các thông tin đã phân tích và dựa trên các kiểu dữ liệu trên database, tác giả đã đưa ra các model của chương trình để sử dụng ở ngôn ngữ Java. Ngoài ra, tác giả sử dụng các Annotation cho các lớp ở Model. Điều này khiến cho việc sự sinh các hàm cần thiết cảu model như @Getter, @Setter, @Builder, @AllArgsConstructor, @NoArgConstructor,… Các Model của bài bao gồm:

- Comment: nhận dữ liệu của bình luận

- NotificationEmail: phục vụ cho việc nhận dữ liệu để gửi Mail

- Post: nhận dữ liệu của của bài đăng

- RefreshToken: nhận dữ liệu của refresh token

- Subreddit: Nhận dữ liệu của chủ đề

- User: Nhận dữ liệu của người dùng

- VericationToken: Nhận dữ liệu của mã xác thực

- Vote: Nhận dữ liệu của bình chọn

- VoteType: Nhận dữ liệu của kiểu bình chọn (ủng hộ, không ủng hộ)

***3.5.2. Xây dựng gói DTO để gửi và nhận dữ liệu của hệ thống:***

Các DTO được lập ra gồm: AuthenticationResponse, CommentsDto, LoginRequest, LogoutRequest, PostRequest, PostResponse, RefreshTokenRequest, RegisterRequest, SubredditDTO, VoteDTO.

***3.5.3. Xây dựng gói Service để triển khai các chức năng của hệ thống:***

Các Service được triển khai trong hệ thông bao gồm: AuthService, CommentService, MailContentBuilder, MailService, PostService, RefreshTokenService, SubredditService, UserDetailServiceIml, VoteService.

Với các Service được liệt kê bên trên, tác giả đã triển khai các chức năng của hệ thống và được mô tả dưới đây:

- Chức năng Đăng kí: Người dùng sẽ nhập thông tin đăng kí sau đó nhấn đăng kí. Nếu form đăng kí thiếu thông tin nào sẽ được hệ thống thông báo cho người dùng để bổ sung thông tin. Sau khi nhấn đăng kí, người dùng sẽ phải kiểm tra e-mail của mình để kích hoạt tài khoản.

- Chức năng Xác thức tài khoản: Sau khi đăng kí tài khoản, người dùng nhận được e-mail trong đó có chưa đường dẫn để kích hoạt tài khoản. Người dùng sẽ bấm vào đường dẫn ấy để kích hoạt tài khoản.

- Chức năng Đăng xuất: Sau khi đăng nhập, hệ thống sẽ cung cấp refreshToken cho người dùng. Mã này sẽ được xóa khi người dùng bấm đăng xuất ra hỏi Website.

- Chức năng Đăng tải bài viết: Người dùng nhập các thông tin cần thiết để có thể đăng tải bài viết. Sau khi ấn Đăng bài, hệ thống sẽ nhận dữ liệu và xử lý.

- Chức năng Lấy tất cả bài viết: Người dùng sẽ gửi yêu cầu đến hệ thống để lấy ra tất cả bài viết theo thứ tự từ ngày đăng gần đây nhất ngay sau khi đăng nhập. Các bài viết sẽ được xử lý và hiện thị cho người dùng.

- Chức năng Lấy bài viết (theo chủ đề, lấy một bài viết, lấy bài viết theo người dùng): Sau khi ấn lấy bài viết yêu một trong ba yêu cầu trên, người dùng sẽ nhận được các bài viết tương ứng. Bài viết sẽ được lấy theo mã chủ đề, tên người dùng, mã bài viết

- Chức năng Tạo chủ đề: Người dùng nhập các thông tin cần thiết để có thể tạo chủ đề. Sau khi ấn Tạo chủ đề, hệ thống sẽ nhận dữ liệu và xử lý.

- Chức năng Lấy chủ đề: Cung cấp cho Front-end để hiển thị thông tin

- Chức năng bình chọn: Người dùng sau khi đăng nhập sẽ nhìn thấy bài viết. Trong mỗi bài viết sẽ có mũi tên màu xanh hướng lên thể hiện cho ủng hộ và mũi tên màu đỏ hướng xuống là không ủng hộ. Mỗi người dùng được chọn một trong hai loại bình chọn. Lần bình chọn được gửi đến hệ thống để tính toán số lượng.

***3.5.3. Xây dựng gói Controller để triển khai các RestAPI của hệ thống:***

REST API (Representational State Transfer Application Programming Interface) là một kiến trúc phần mềm mô tả giao diện lập trình ứng dụng (API) cho các ứng dụng web. REST API được thiết kế để đơn giản, dễ sử dụng và có thể mở rộng, giúp các ứng dụng giao tiếp và trao đổi dữ liệu hiệu quả với nhau.

Trong phần Back-end, tác giả đã viết mã để cung cấp API cho Front-end để xử lý giao diện. Mỗi API sẽ tương ứng với một chức năng đã được nêu ở trên để xử lý ở trên Front-end.

***3.5.4. Cấu hình Spring Security cho hệ thống***

Spring Security ở dự án được cấu hình như sau:

- Vô hiệu hóa CSRF và CORS của Website

- Cho phép tất cả các yêu cầu đến các đường dẫn bắt đầu bằng "/api/auth/" mà không cần xác thực. Dòng này có thể định nghĩa các điểm cuối liên quan đến xác thực người dùng (đăng nhập, đăng ký, ...).

- Cấu hình Spring Security để sử dụng xác thực dựa trên JWT (JSON Web Token) cho máy chủ tài nguyên OAuth2. Điều này có nghĩa là máy chủ cần một mã thông báo JWT hợp lệ trong tiêu đề ủy quyền yêu cầu để truy cập các tài nguyên được bảo vệ.

- Cấu hình Spring Security STATELESS: máy chủ sẽ không quản lý phiên người dùng và chỉ dựa vào mã thông báo JWT để xác thực.

- Thiết lập điểm nhập tùy chỉnh để xử lý các trường hợp người dùng cố gắng truy cập tài nguyên được bảo vệ mà không có mã thông báo hợp lệ. Dòng này có thể trả về phản hồi 401 (Unauthorized) với thông tin thích hợp.

- Thiết lập trình xử lý tùy chỉnh để xử lý các trường hợp người dùng có mã thông báo hợp lệ nhưng không có quyền truy cập đủ vào tài nguyên cụ thể. Dòng này có thể trả về phản hồi 403 (Forbidden).

# 3.6. Sử dụng Postman để test API được cung cấp từ Back-end

Chức năng đăng kí (POST): /api/auth/signup

**Bảng 3.14. Dữ liệu nhận đi và gửi về của chức năng đăng kí**

|  |  |
| --- | --- |
| Request | Response |
| username | Người dùng đăng kí thành công |
| email |
| password |

Chức năng kích hoạt tài khoản (GET): /api/auth/accountVerification/ + token

**Bảng 3.15. Dữ liệu nhận đi và gửi về của chức năng kích hoạt tài khoản**

|  |  |
| --- | --- |
| Request | Response |
|  | Tài khoản đã được kích hoạt |

Chức năng kích hoạt đăng nhập (POST): /api/auth/login

**Bảng 3.16. Dữ liệu nhân đi và gửi về của chức năng đăng nhập**

|  |  |
| --- | --- |
| Request | Response |
| username | authenticationToken |
| password | refreshToken |
|  | expiresAt |
|  | username |

Chức năng làm mới token (POST): /api/auth/refresh/token

**Bảng 3.17. Dữ liệu nhân đi và gửi về của chức năng làm mới token**

|  |  |
| --- | --- |
| Request | Response |
| refreshToken | authenticationToken |
| username | refreshToken |
|  | expiresAt |
|  | username |

Chức năng đăng xuất (POST): /api/auth/logout

**Bảng 3.18. Dữ liệu nhân đi và gửi về của chức năng đăng nhập**

|  |  |
| --- | --- |
| Request | Response |
| refreshToken |  |
| username |

Chức năng tạo chủ đề (POST): /api/subreddit

**Bảng 3.19. Dữ liệu nhân đi và gửi về của chức năng tạo chủ đề**

|  |  |
| --- | --- |
| Request | Response |
| name | id |
| description | name |
|  | description |
|  | numberOfPost |

Chức năng lấy tất cả chủ đề (GET): /api/subreddit

**Bảng 3.20. Dữ liệu nhân đi và gửi về của chức năng lấy tất cả chủ đề**

|  |  |
| --- | --- |
| Request | Response |
|  | id |
| name |
| description |
| numberOfPost |

Chức năng lấy một chủ đề (GET): /api/subreddit/{id}

**Bảng 3.20. Dữ liệu nhân đi và gửi về của chức năng lấy cụ thể chủ để**

|  |  |
| --- | --- |
| Request | Response |
|  | id |
| name |
| description |
| numberOfPost |

Chức năng tạo bài viết (POST): /api/posts

**Bảng 3.21. Dữ liệu nhân đi và gửi về của chức năng tạo bài viết**

|  |  |
| --- | --- |
| Request | Response |
| subredditName |  |
| postName |
| url |
| description |

Chức năng lấy tất cả bài viết (GET): /api/posts

**Bảng 3.22. Dữ liệu nhân đi và gửi về của chức năng lấy tất cả bài viết**

|  |  |
| --- | --- |
| Request | Response |
|  | id |
| postName |
| url |
| description |
| userName |
| subredditName |
| voteCount |
| commentCount |
| duration |
| upVote |
| downVote |

Chức năng lấy bài viết theo người dùng (GET): /api/posts/by-user/{username}

**Bảng 3.23. Dữ liệu nhân đi và gửi về của chức năng lấy bài viết theo người dùng**

|  |  |
| --- | --- |
| Request | Response |
|  | id |
| postName |
| url |
| description |
| userName |
| subredditName |
| voteCount |
| commentCount |
| duration |
| upVote |
| downVote |

Chức năng lấy bài viết theo chủ đề (GET): /api/posts/by-subreddit/{id}

**Bảng 3.24. Dữ liệu nhân đi và gửi về của chức năng lấy bài viết theo chủ đề**

|  |  |
| --- | --- |
| Request | Response |
|  | id |
| postName |
| url |
| description |
| userName |
| subredditName |
| voteCount |
| commentCount |
| duration |
| upVote |
| downVote |

Chức năng tạo bình luận (POST): /api/comments

**Bảng 3.25. Dữ liệu gửi đi và nhận về của chức năng bình luận**

|  |  |
| --- | --- |
| Request | Response |
| postID |  |
| text |
| userName |

Chức năng lấy bình luận theo bài viết (GET): /api/comments/by-post/{postId}

**Bảng 3.26. Dữ liệu gửi đi và nhận về của chức năng lấy bình luận theo bài viết**

|  |  |
| --- | --- |
| Request | Response |
|  | id |
| postId |
| createdDate |
| text |
| userName |

Chức năng bình chọn bài viết (POST): /api/votes

**Bảng 3.26. Dữ liệu gửi đi và nhận về của chức năng bình chọn**

|  |  |
| --- | --- |
| Request | Response |
| voteType |  |
| postId |

# CHƯƠNG 4: TRIỂN KHAI THỬ NGHIỆM VÀ ĐÁNH GIÁ

# 4.1. Giới thiệu website

## *Giao diện đăng nhập*

A screenshot of a computer

Description automatically generated*Hình 4.1. Giao diện đăng nhập*

## *Giao diện sau khi đăng nhập thành công*

A screenshot of a computer

Description automatically generated*Hình 4.2. Giao diện đăng nhập thành công*

## *Giao diện đăng kí*

A screenshot of a computer

Description automatically generated*Hình 4.3. Giao diện đăng kí*

## *Giao diện tạo bài viết*

A screenshot of a computer

Description automatically generated*Hình 4.4. Giao diện tạo bài viết*

* + 1. *Giao diện tạo chủ đề*

A screenshot of a computer

Description automatically generated*Hình 4.5. Giao diện tạo chủ đề*

## *Giao diện chi tiết bài viết và bình luận*

A screenshot of a computer

Description automatically generated*Hình 4.6. Giao diện chi tiết bài viết và bình luận*

# 4.2. Đánh giá

Website đã cung cấp cho người dùng giao diện dễ dàng với các chức năng được liệt kê. Đảm bảo chức năng bảo mật thông tin cá nhân người dùng.

*Những mục tiêu đạt được:*

Xây dựng được Website cung cấp các chức năng để người dùng có thể đăng các bài đăng theo một chủ đề. Người dùng cũng có thể xem các bình luận của chủ đề ấy để xem các người dùng khác có thông tin gì khác không. Người dùng có thể bình chọn tích cực hay tiêu cực cho bài viết. Người dùng cũng quản lý được các bài đăng của mình. Đã sử dụng được Postman để test API cho các API được tạo ra ở phần Back-end.

*Hạn chế của đề tài*

Mặc dù đề tài đã đáp ứng được về cơ bản mục tiêu chính đề ra ban đầu nhưng đề tài vẫn còn một số điểm hạn chế:

- Một số chức năng đã hoàn thành ở Back-end nhưng chưa được triển khai cụ thể ở Front-end

- Giao diện cần được hoàn thiện hơn.

- Các chức năng chưa đa dạng

- Cần bổ sung các thông tin liên quan đến xác thực dữ liệu trước khi gửi

- Vẫn còn một số lỗi ở Front-end trong quá trình chạy thử nghiệm. *Phương hướng phát triển*

- Tích hợp thêm các chức năng liên quan đến phân quyền để có người quản lý Website, quản lý bình luận, quản lý bài viết, ...

- Tiếp tục hoàn thiện thêm về giao diện để tăng thêm trải nghiệm người dùng.

- Phát triển cộng đồng người dùng để có nhiều chủ đề, bài viết để bàn luận

- Phát triển, mở rộng thêm chức năng báo cáo, tạo biểu đồ

- Nghiên cứu để phát triển website đúng với tiêu chuẩn cộng đồng cũng như ngăn chặn các bài viết mang tính tiêu cực, spam, …

# KẾT LUẬN

Trong chuyên đề này, em đã nghiên cứu và phát triển một website dành cho người dùng, với mục tiêu là cung cấp cho người dùng một cộng đồng để có thể chia sẻ và thảo luận về các chủ đề đa dạng mà người dùng có thể tự tạo ra chủ đề. Website đã xây dựng được các chức năng cơ bản cho người dùng bao gồm các chức năng liên quan đến xác thực, các chức năng liên quan đến chủ đề, bài viết, bình luận, bình chọn. Đề tài của em có ý nghĩa thực tiễn và đóng góp cho sự chuyển đổi số.

Tuy nhiên, đề tài của em vẫn còn nhiều hạn chế như còn thiếu các chức năng do phần khảo sát chưa kĩ, giao diện chưa thân thiện và hấp dẫn với người dùng và cần phải cung cấp chức năng phân quyền để quản lý các thông tin liên quan đến Website. Trong tương lai em sẽ tiếp tục nghiên cứu và phát triển thêm các chức năng này để nâng cao chất lượng và hiệu suất của Website.

Dù đã cố gắng hết sức để hoàn thiện, nhưng với thời gian có hạn và kiến thức hạn hẹp, website của em vẫn còn đơn giản và có thể chưa đáp ứng đầy đủ yêu cầu vì việc xây dựng một Website chất lượng đòi hỏi nhiều nguồn lực chất lượng cao liên quan đến nhân sự cũng như cơ sở vật chất và kiến thức về phần Front-end và phần Back-end. Đồng thời cũng không tránh khỏi những thiếu sót, nên em rất mong nhận được sự góp ý của thầy cô và các bạn để có thể tiếp tục hoàn thiện hơn trong tương lai.

Và một lần nữa, em xin chân thành cảm ơn các thầy cô trong Viện, đặc biệt là thầy Đặng Minh Quân đã nhiệt tình hướng dẫn và góp ý cho em trong suốt quá trình hoàn thiện đề tài.

Em xin chân thành cảm ơn!

# TÀI LIỆU THAM KHẢO

[1] Bài giảng, slide môn học Phân tích và Thiết kế hệ thống. ThS. Trần Thị Mỹ Diệp.Trường Đại học Kinh tế Quốc dân.

[2] Một số trang web khác:

[Spring Initializr](https://start.spring.io/)

[Spring Boot Tutorial - Learn Spring Boot (geeksforgeeks.org)](https://www.geeksforgeeks.org/spring-boot/)

<Angular>

[Spring Boot - Sending Email via SMTP - GeeksforGeeks](https://www.geeksforgeeks.org/spring-boot-sending-email-via-smtp/)

[Introduction to Spring Data JPA | Baeldung](https://www.baeldung.com/the-persistence-layer-with-spring-data-jpa)

[Spring Security](https://spring.io/projects/spring-security)

[Postman API Platform | Sign Up for Free](https://www.postman.com/)

[MySQL Tutorial (w3schools.com)](https://www.w3schools.com/mysql/)