TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

-----🙞🙜🕮🙞🙜-----

****

**ĐỀ TÀI**

**XÂY DỰNG WEBSITE CHIA SẺ KIỂN THỨC**

**KHÓA LUẬN TỐT NGHIỆP CHUYÊN NGÀNH CNTT**

**GIẢNG VIÊN HƯỚNG DẪN**

**ThS. TRẦN CÔNG TÚ**

**NHÓM SVTH:**

**NGUYỄN ĐÌNH PHÚ 18110175**

**VÕ ANH HUY 18110124**

TP HCM, THÁNG 09 NĂM 2022

**Trường ĐH Sư phạm Kỹ thuật TP.HCM CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**Khoa Đào tạo Chất lượng cao Độc lập – Tự do – Hạnh phúc**

**\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\***

**PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN**

**Họ và tên sinh viên:** Nguyễn Đình Phú **MSSV:** 18110175

**Họ và tên sinh viên:** Võ Anh Huy **MSSV:** 18110124

**Ngành:** Công nghệ thông tin

**Tên đề tài:** Xây dựng website chia sẻ kiển thức

**Họ và tên Giáo viên hướng dẫn:** ThS. Trần Công Tú

**Nhận xét:**

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

2. Ưu điểm:

3. Khuyết điểm:

4. Đề nghị cho bảo vệ hay không?

5. Đánh giá loại:

6. Điểm:

*Tp. Hồ Chí Minh, ngày… tháng… năm 2022*

*Giáo viên hướng dẫn*

*(Kỹ & ghi rõ họ tên)*

**Trường ĐH Sư phạm Kỹ thuật TP.HCM CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**Khoa Đào tạo Chất lượng cao Độc lập – Tự do – Hạnh phúc**

**\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\***

**PHIẾU NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN**

**Họ và tên sinh viên:** Nguyễn Đình Phú **MSSV:** 18110175

**Họ và tên sinh viên:** Võ Anh Huy **MSSV:** 18110124

**Ngành:** Công nghệ thông tin

**Tên đề tài:** Xây dựng website chia sẻ kiển thức

**Họ và tên Giáo viên phản biện:**

**Nhận xét:**

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

2. Ưu điểm:

3. Khuyết điểm:

4. Đề nghị cho bảo vệ hay không?

5. Đánh giá loại:

6. Điểm:

*Tp. Hồ Chí Minh, ngày… tháng… năm 2022*

*Giáo viên hướng dẫn*

*(Kỹ & ghi rõ họ tên)*

LỜI CẢM ƠN

*Lời đầu tiên nhóm xin gửi lời cảm ơn chân thành sâu sắc đến với khoa Công Nghệ Thông Tin, Khoa Đào tạo Chất lượng cao – Trường ĐH Sư phạm Kỹ thuật TP. HCM đã tạo điều kiện cho nhóm em được học tập, phát triển các kiến thức, kỹ năng để thực hiện đề tài này. Bên cạnh đó nhóm xin phép gửi đến thầy Trần Công Tú lời cảm ơn sâu sắc nhất. Trải qua một quá trình dài học tập và thực hiện đề tài trong học kỳ vừa qua, thầy đã tận tâm hướng dẫn nhóm một cách rất nhiệt tình để nhóm có thể hoàn thành được đề tài như ngày hôm nay. Mặc dù diễn ra gây rất nhiều khó khăn trong quá trình thực hiện đề tài nhưng thầy đã tận tâm chỉ bảo, hướng dẫn và động viên nhóm để giúp nhóm vượt qua.*

*Nhờ có một nền tảng kiến thức vững chắc trong suốt quá trình học tập và những kinh nghiệm có được nhờ học tập ở trường, các thầy cô đã tặng cho nhóm em một khối kiến thức vô cùng rộng lớn và vững chắc giúp nhóm có cơ hội phát triển bản thân trong tương lai và gần nhất là thực hiện thành công đề tài này.*

*Với lượng kiến thức là vô cùng rộng lớn và khả năng hạn hẹp của nhóm nên trong đề tài không tránh khỏi những thiếu sót, vì vậy nhóm rất mong nhận được sự góp ý tận tình của quý thầy cô để nhóm có thể hoàn thiện đề tài một cách tốt nhất. Cuối cùng nhóm em xin chúc các thầy cô có sức khỏe thật tốt và đạt được nhiều thành công.*

*Chúng em xin chân thành cảm ơn!*

Nhóm sinh viên thực hiện

Nguyễn Đình Phú

Võ Anh Huy

TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HCM

KHOA ĐÀO TẠO CHẤT LƯỢNG CAO

\*\*\*\*\*\*

**ĐỀ CƯƠNG KHÓA LUẬN TỐT NGHIỆP**

**Họ và tên Sinh viên**: Nguyễn Đình Phú **MSSV:** 18110175

**Họ và tên sinh viên:** Võ Anh Huy **MSSV:** 18110124

Thời gian làm khóa luận từ:

Ngành: Công nghệ thông tin

Chuyên ngành: Công nghệ phần mềm

**Tên đề tài:** XÂY DỰNG WEBSITE CHIA SẺ KIỂN THỨC

**Họ và tên Giáo viên hướng dẫn:** ThS. Trần Công Tú

**Nhiệm vụ của khóa luận:**

1. *Lý thuyết*
   1. Tìm hiểu RESTful API
   2. Tìm hiểu về Spring Boot
   3. Tìm hiểu về Spring Data JPA
   4. Tìm hiểu về ReactJS
   5. Tìm hiểu về Docker
2. *Thực hành*

Xây dựng hệ thống chia sẻ bài viết về các chủ đề kiến thức:

* 1. Khách vãng lai: Xem danh sách và nội dung chi tiết, bình luận, đánh giá của các bài viết. Ngoài ra khách vãng lai có thể đăng ký tài khoản người dùng cơ bản để thực hiện các tính năng cho phép tương tác với hệ thống.
  2. Người dùng cơ bản: Xem danh sách và nội dung chi tiết, bình luận, đánh giá của các bài viết, viết bình luận, đánh giá, báo cáo nội dung bài viết. Đồng thời người dùng cơ bản cũng có thể đăng bài viết lên hệ thống để chia sẻ với những người dùng khác.
  3. Người kiểm duyệt: Có mọi quyền hạn của người dùng cơ bản, thêm vào đó người kiểm duyệt có quyền xem các báo cáo về các bài viết và đưa ra quyết định xóa bài viết khỏi hệ thống. Tài khoản người kiểm duyệt chỉ có thể được tạo bởi quản trị viên.
  4. Quản trị viên: Là người dùng có quyền hạn cao nhất, thực hiện được mọi chức năng mà người kiểm duyệt có, đồng thời có thể quản lý các người kiểm duyệt khác như xem danh sách các người kiểm duyệt, tạo mới tài khoản, đổi mật khẩu hoặc vô hiệu hóa tài khoản người kiểm duyệt.

KẾ HOẠCH THỰC HIỆN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** |  |  |  |  |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
|  |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |
| 6 |  |  |  |  |
| 7 |  |  |  |  |
| 8 |  |  |  |  |
| 9 |  |  |  |  |
| 10 |  |  |  |  |
| 11 |  |  |  |  |
| 12 |  |  |  |  |
| 13 |  |  |  |  |
| 14 |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| **Ý kiến của giáo viên hướng dẫn**  (ký và ghi rõ họ tên) |  | Ngày … tháng … năm 2022  Người viết đề cương  Nguyễn Đình Phú |

DANH MỤC HÌNH

DANH MỤC BẢNG

MỤC LỤC

1. PHẦN MỞ ĐẦU
2. Tính cấp thiết của đề tài

“Xây dựng Website chia sẻ kiến thức”.

1. Giải pháp

Hệ thống cung cấp cho người đọc - chia sẻ, người quản lý một nền tảng chia sẻ kiến thức một cách linh động, quản lý một cách đơn giản, thân thiện.

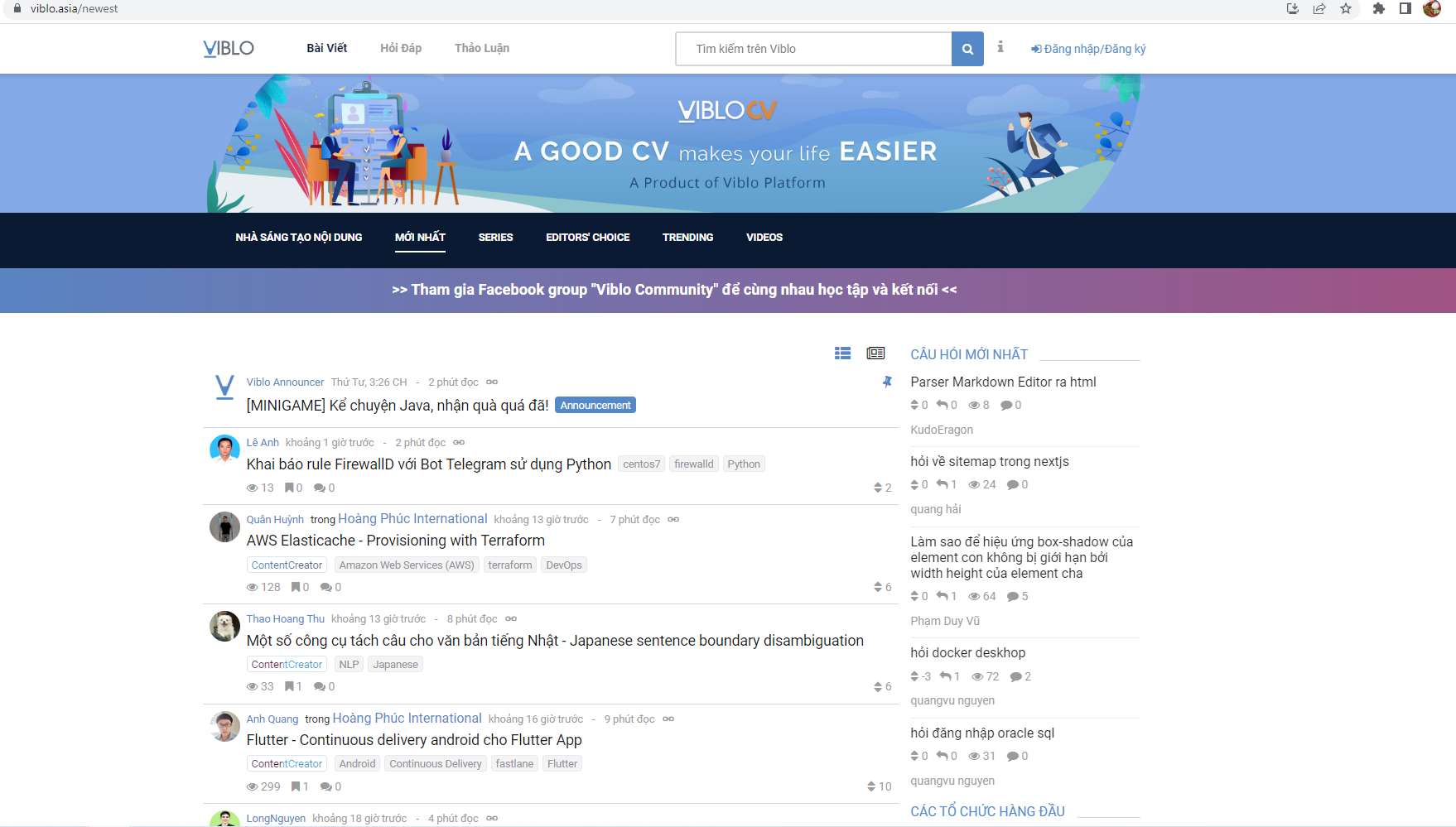
Về tổng quan nền tảng chia cơ bản 3 phần:

* + Khách vãng lai, người dùng cơ bản:
    - Cho phép khách vãng lai xem danh sách, nội dung chi tiết từng bài viết cũng như bình luận, đánh giá của người dùng khác.
    - Cho phép người dùng cơ bản xem danh sách, nội dung bài viết, bình luận, đánh giá, đồng thời có thể để lại đánh giá của bản thân, báo cáo bài viết, và đăng bài viết.
    - Tính năng chia sẻ / nghe nội dung âm thanh giúp người dùng thuận tiện hơn khi cần nghe âm thanh thay vì đọc.
* Quản lý chung (dành cho người kiểm duyệt và quản trị viên)
  + Cho phép quản trị viên quản lý danh sách người kiểm duyệt, thêm, thay đổi mật khẩu, vô hiệu hóa / kích hoạt tài khoản người kiểm duyệt.
  + Cho phép người kiểm duyệt và quản trị viên xem các báo cáo về các bài viết và đưa ra quyết định xóa bài viết.

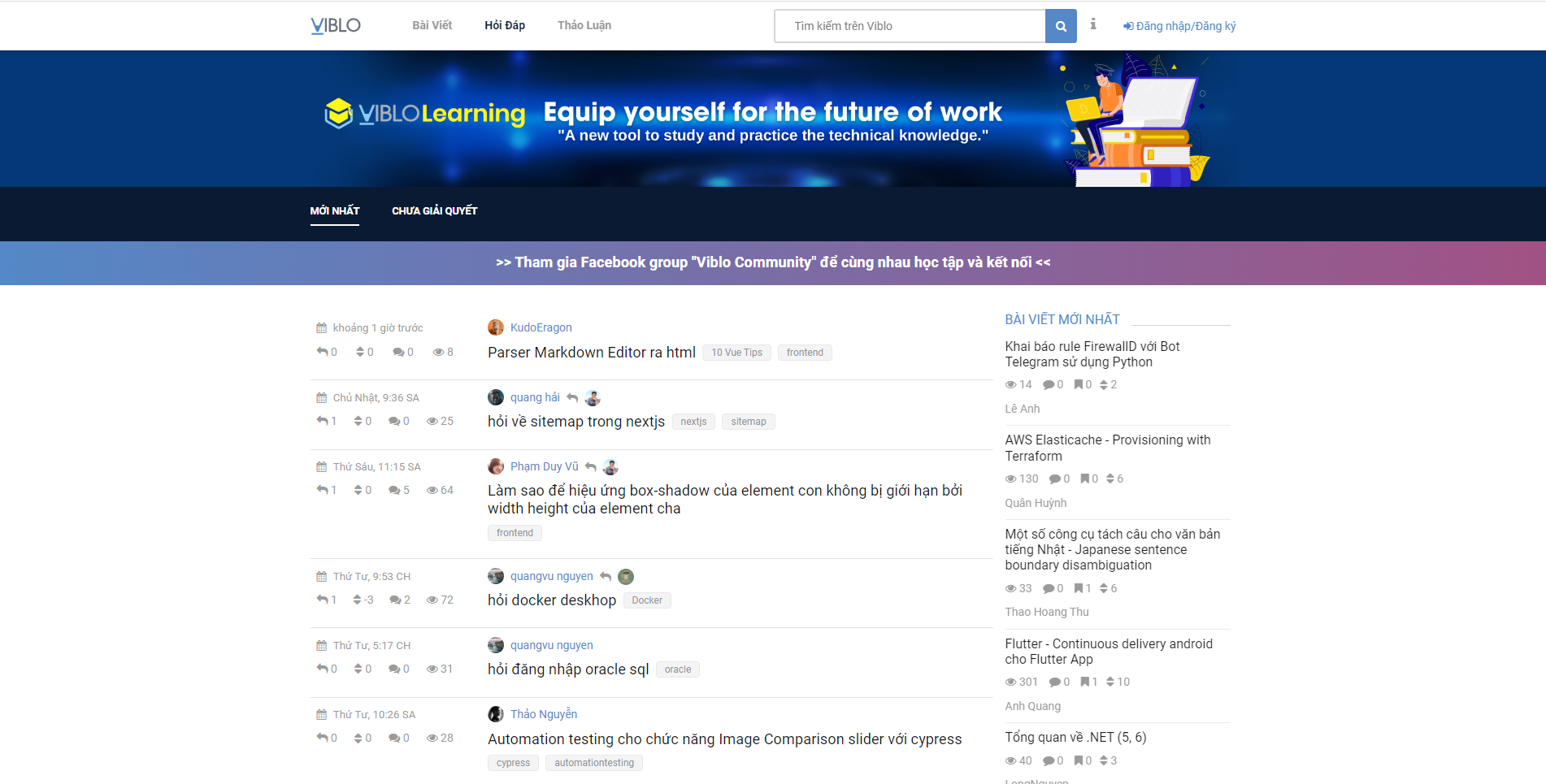
1. Mục tiêu của đề tài

* Tạo một trang web mạng xã hội hỗ trợ việc trao đổi, trau dồi kiến thức.
* Xây dựng hoàn thành trang mạng xã hội kiến thức cung cấp đầy đủ các tính năng cần thiết:
  + Xây dựng trang web mạng xã hội kiến thức với các tính năng chính:
    - Đăng ký tài khoản
    - Đăng nhập / Đăng xuất
    - Xem các bài viết
    - Tham gia bình luận
    - Đánh giá bài viết
    - Báo cáo bài viết
    - Đăng bài viết
  + Các chức năng riêng của người dùng cơ bản
    - Chỉnh sửa bài viết của bản thân
    - Xóa bài viết của bản thân
  + Các chức năng riêng của người kiểm duyệt
    - Xem danh sách các báo cáo về các bài viết
    - Xóa bài viết của người dùng khác
  + Các chức năng riêng của quản trị viên
    - Xem danh sách các báo cáo về các bài viết
    - Chỉnh sửa bài viết của người dùng khác
    - Xóa bài viết của người dùng khác
    - Quản lý các tài khoản người kiểm duyệt
* Áp dụng công nghệ Java - Spring Boot, ReactJS, SQL Server vào dự án.
* Giao diện thân thiện với người dùng, đơn giản

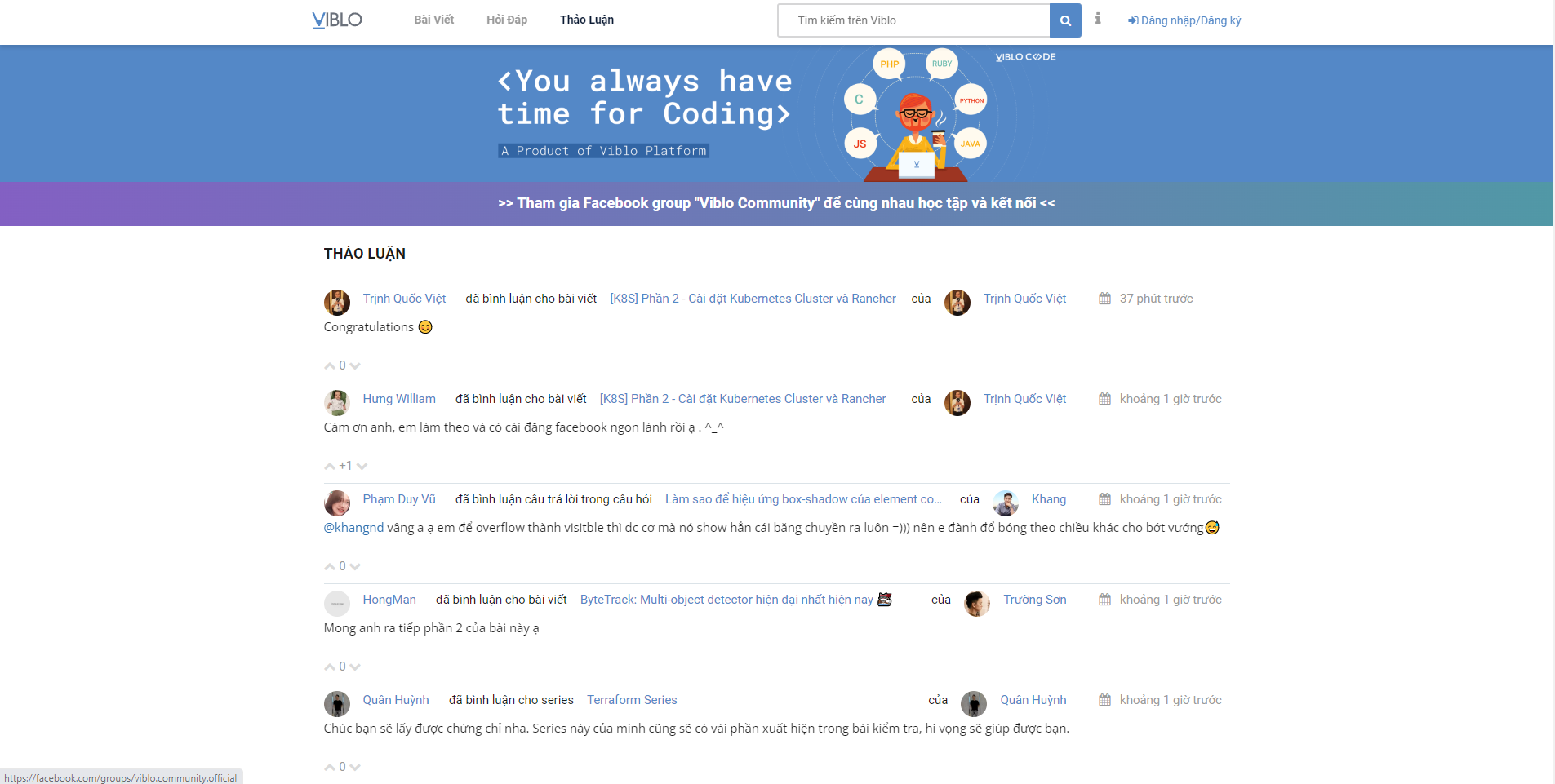
1. PHẦN NỘI DUNG
   1. KHẢO SÁT HIỆN TRẠNG VÀ XÁC ĐỊNH YÊU CẦU
      1. KHẢO SÁT HIỆN TRẠNG
         1. Viblo.asia



Hình khảo sát hiện trạng trang Viblo.asia 1



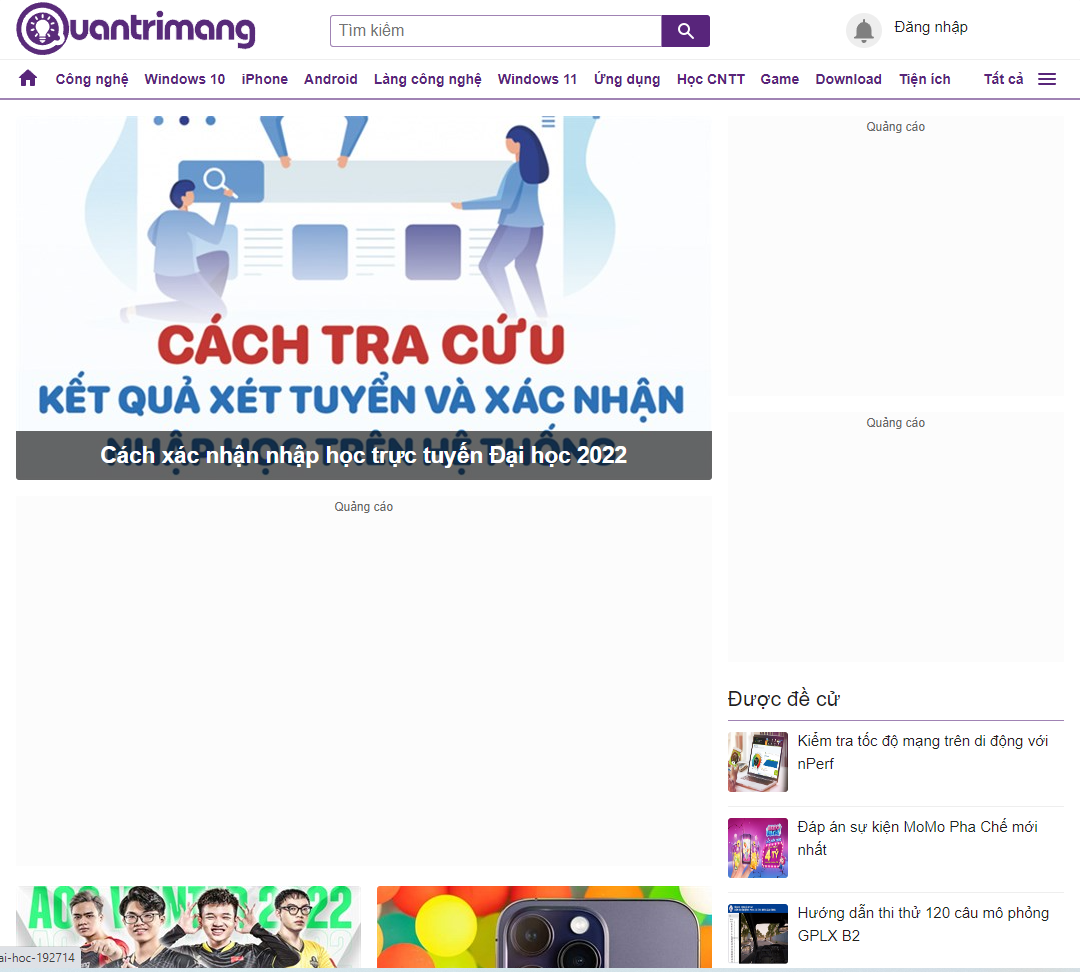
Hình khảo sát hiện trạng trang Viblo.asia 2



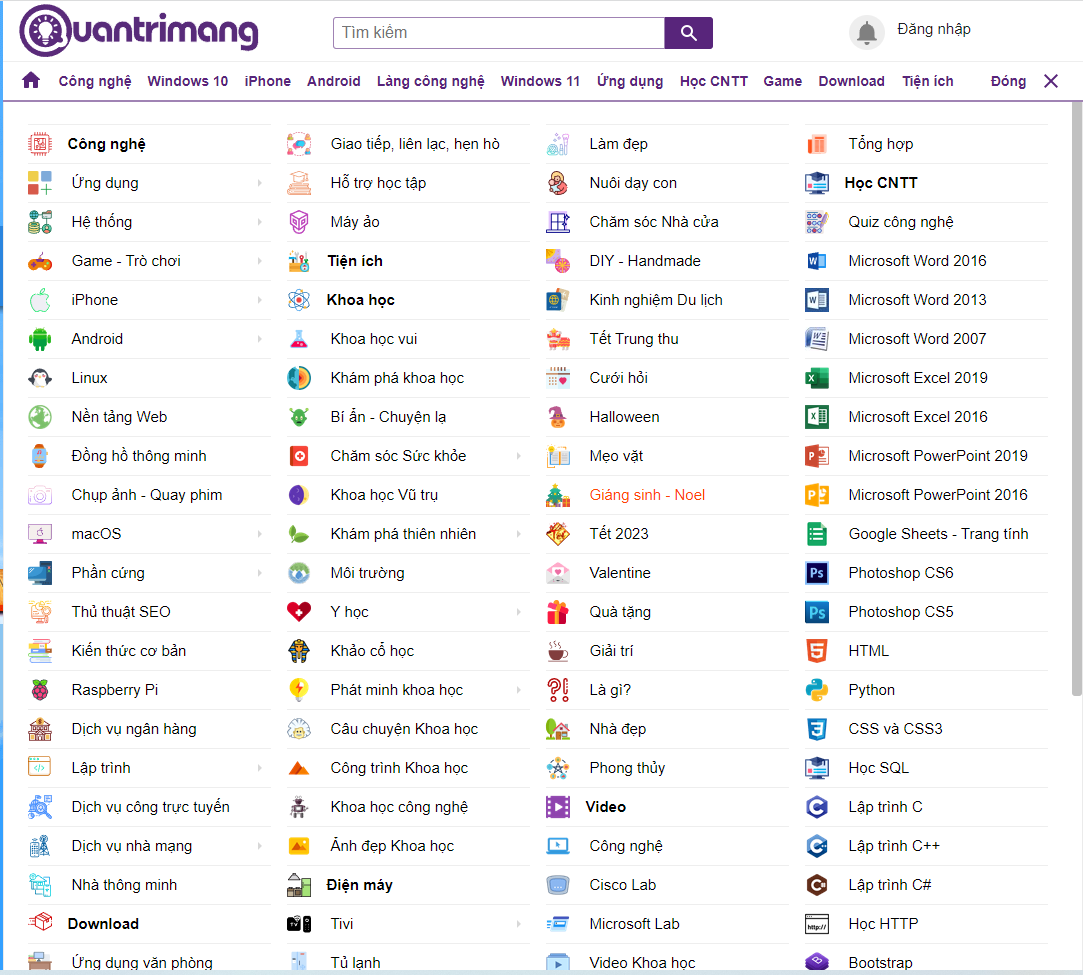
Hình khảo sát hiện trạng trang Viblo.asia 3

*Nguồn: https://viblo.asia/*

* + - * 1. Ưu điểm
* Giao diện:
  + Bố cục đơn giản.
  + Cung cấp màn hình giao diện tương tác dễ dàng cho những người mới dùng lần đầu tiên.
* Chức năng:
  + Cung cấp đầy đủ các chức năng cơ bản của trang chia sẻ kiến thức.
  + Độ bao phủ nội dung rộng.
    - * 1. Nhược điểm
* Giao diện:
  + Cách bố trí nội dung ở một số nơi không thích hợp, không cân đối.
* Chức năng:
  + Cần đăng nhập để xem toàn bộ nội dung bài viết
  + Bố cục trang chủ (newest) chưa thực sự cân đối và đẹp mắt.
  + Thiếu danh mục bài viết.
    - 1. Quantrimang.com



Hình khảo sát hiện trạng hệ thống Quantrimang.com 1



Hình khảo sát hiện trạng hệ thống Quantrimang.com 2

* + - * 1. Ưu điểm
* Giao diện:
  + Bố cục đơn giản.
  + Giao diện thân thiện, dễ sử dụng.
* Chức năng:
  + Đa dạng nội dung.
  + Hỗ trợ đánh giá bài viết
    - * 1. Nhược điểm
* Quảng cáo xuất hiện nhiều
* Không tương tác được.
* Chỉ cho phép chủ kênh đăng bài.
  + 1. Kết luận về những tiêu chí cần đạt được
       1. Ưu điểm
* Hỗ trợ các chức năng cơ bản của một trang chia sẻ kiến thức.
* Nội dung đa dạng.
* Giao diện dễ sử dụng.
  + - 1. Nhược điểm
* Bố cục phần chia còn chưa hợp lý.
* Nhiều nội dung dư thừa được hiển thị.
* Một số vẫn chưa hỗ trợ cho phép người dùng tương tác.
* Bố trí quảng cáo chưa hợp lí
  + - 1. Mong muốn của nhóm sau khi khảo sát
* Tạo một trang mạng xã hội cung cấp đầy đủ tính năng cơ bản phục vụ mục đích chia sẻ, trao đổi và trau dồi kiến thức.
* Nội dung đơn giản phù hợp, hạn chế nội dung không cần thiết.
* Dễ dàng sử dụng, đơn giản, thân thiện.
* Hỗ trợ người dùng đăng bài viết
  1. CƠ SỞ LÝ THUYẾT
     1. Tổng quan công nghệ sử dụng
        1. Giới thiệu chung

Xây dựng web:

* Backend: Java, Spring Boot, Spring Data JPA, RESTful API.
* Frontend: ReactJS và một số thư viện hỗ trợ.
* Database: Microsoft SQL Server (Docker).
  + - 1. Lý do chọn công nghệ
         1. Java – Spring Boot
* Java là một ngôn ngữ lập trình được triển khai rộng rãi và rất phổ biến để phát triển web.
* Java cũng là ngôn ngữ có tính hướng đối tượng đầy đủ nhất, hoạt động như một nền tảng tiêu chuẩn cho rất nhiều doanh nghiệp và nhà phát triển trên thế giới.
* Java có khả năng hoạt động tốt trên nhiều nền tảng khác nhau nhờ JVM.
* Là một ngôn ngữ mạnh mẽ, an toàn và có khả năng triển khai, thực thi ứng dụng với hiệu suất cao.
* Spring Boot là một module trong Spring Framework, giúp giảm tải cấu hình, số lượng hành động mà nhà phát triển phải thực hiện, hỗ trợ truy cập cơ sở dữ liệu như MySQL, SQL Server, Oracle, … với Spring Data JPA. Cùng với các ưu điểm mạnh mẽ khác của Spring Boot giúp quá trình phát triển ứng dụng trở nên dễ dàng và nhanh chóng hơn.
  + - * 1. Microsoft SQL Server
* Hiệu suất cao
* Tính sẵn sàng cao
* SQL vẫn là nền tảng cho các hệ thống lớn hiện nay
* Độ tin cậy cao
* Đặc trưng bởi 4 tính chất (ACID)
* Được phát triển bởi Microsoft và tuổi đời rất lâu
* Được Microsoft phát triển các công cụ hỗ trợ vô cùng mạnh mẽ và thuận tiện
  + - * 1. ReactJS
* ReactJS là một thư viện rất đơn giản, nhẹ và dễ học do chỉ thao tác với lớp view.
* ReactJS cung cấp một cấu trúc dựa trên JSX component, các component có thể được tái sử dụng giúp quá trình phát triển và duy trở nên dễ dàng hơn.
* Render nhanh với Virtual DOM. [4]
* Clean Abstraction, chỉ cần hiểu về life cycle, props, state thay vì các khái niệm phức tạp. [4]
* Được phát triển bởi Facebook (nay là Meta) cùng với cộng đồng hỗ trợ và thư viện mạnh mẽ.
  + - * 1. Docker
* Docker là một nền tảng cung cấp cách để build, deploy và run ứng dụng dễ dàng hơn bằng cách sử dụng các containers (trên nền ảo hóa). [5]
* Thuận tiện hơn khi cài đặt môi trường phát triển ứng dụng.
* Hỗ trợ tốt trên nhiều nền tảng (Linux, macOS, Windows).

Do là một nền tảng ảo hóa, Docker giúp nhà phát triển không cần quá quan tâm đến môi trường của máy chủ và các thiết lập phức tạp, thêm vào đó nó cũng giúp nhà phát triển dử dụng các dịch vụ không hỗ trợ trên nền tảng hệ điều hành của máy chủ. Có thể kể đến trong đề tài này là MS SQL Server chỉ có thể cài đặt trên Windows và Linux. Tuy nhiên nhờ Docker, MS SQL Server cũng có thể được cài trên máy chủ chạy hệ điều hành macOS.

* + 1. Tìm hiểu công nghệ
       1. Microsoft SQL Server
          1. Định nghĩa

Microsoft SQL Server (còn được gọi là MS SQL Server hay SQL Server) là một loại phần mềm được phát triển bởi Microsoft vào cuối thập niên 80 của thế kỷ XX (1989). SQL Server được sử dụng để có thể dễ dàng lưu trữ dữ liệu dựa theo tiểu chuẩn RDBMS.

SQL Server được xây dựng dựa trên ngôn ngữ T-SQL (Transact-SQL) và là một hệ cơ sở dữ liệu dạng quan hệ đối tượng. [1]

* + - * 1. Ưu điểm
* Có phiên bản miễn phí (SQL Express) có gần như đầy đủ các tính năng của bản tiêu chuẩn.
* Được phát triển bởi Microsoft.
* Hỗ trợ tốt cho người dùng chuyên nghiệp và doanh nghiệp lớn.
* Có các công cụ mạnh mẽ, đa dạng.
* Tài liệu hướng dẫn và hỗ trợ phong phú đến từ Microsoft.
* Độ tin cậy và ổn định cao.
* Là hệ quản trị cơ sở dữ liệu quan hệ (RDBMS).
  + - * 1. Nhược điểm
* Chi phí cao nếu nhu cầu sử dụng vượt quá nhu cầu của bản miễn phí.
* Hạn chế về cơ sở hạ tầng và phải cam kết đầu tư các sản phẩm trong hệ sinh thái của Microsoft để sử dụng SQL Server.
  + - 1. Docker
         1. Định nghĩa

Docker là một nền tảng cho nhà phát triển và quản trị viên hệ thống để build, deploy và run ứng dụng với containers. Docker cho phép tạo ra các môi trường độc lập để khởi chạy và phát triển ứng dụng trên nền tảng ảo hóa. Khi deploy lên server chỉ cần run Docker container thì ứng dụng sẽ được khởi chạy.

* + - * 1. Ưu điểm
* Có khả năng start và stop chỉ trong vài giây
* Containers có thể build và loại bỏ nhanh hơn máy ảo
* Có thể khởi chạy container trên nhiều môi trường máy chủ khác nhau (Linux, macOS, Windows)
* Dễ dàng thiết lập môi trường làm việc
* Containers yêu cầu ít tài nguyên phần cứng hơn máy ảo
* Hệ thống image trên docker hub phong phú, đa dạng
* Giữ môi trường máy chủ sạch sẽ hơn vì các container là độc lập và không ảnh hưởng trực tiếp đến môi trường máy chủ
  + - * 1. Nhược điểm
* Không nên dùng cho phát triển các ứng dụng desktop GUI
* Docker Data Volume vẫn còn chưa thực sự tốt cho việc lưu trữ dữ liệu
  + - 1. Java
         1. Định nghĩa

Java là một ngôn ngữ lập trình hướng đối tượng (OOP) dựa trên các lớp. Ban đầu Java được phát triển bởi Sun Microsystems và phát hành vào năm 1995. Khác với phần lớn các ngôn ngữ lập trình khác, thay vì biên dịch mã nguồn thành mã máy hoặc thông dịch mã nguồn khi chạy Java được thiết kế để biên dịch mã nguồn thành bytecode sau đó được môi trường thực thi (runtime environment) chạy. Nhờ thiết kế này, sau khi được biên dịch, sản phẩm có thể chạy trên bất kì môi trường nào miễn là hỗ trợ JVM. [2]

* + - * 1. Ưu điểm
* Dễ học
* Dễ đọc code
* Phát triển nhanh chóng
* Quản lý bộ nhớ tốt
* Là ngôn ngữ lập trình hướng đối tượng
* Hỗ trợ tốt cho việc phát triển với nhiều thư viện
* Là nền tảng cho Spring Framework – một hệ sinh thái mạnh mẽ
  + - * 1. Nhược điểm
* Vẫn là một ngôn ngữ cổ điển, không thường xuyên nâng cấp các tính năng mới so với các ngôn ngữ hiện đại
* Sử dụng nhiều bộ nhớ hơn so với C# và C++ [8]
* Không có số nguyên không dấu
* Không có con trỏ hoặc tính năng tương tự
* Không cho phép lập chỉ mục hoặc toán tử trên các đối tượng
  + - 1. Spring Boot
         1. Định nghĩa

Spring Boot là một dự án phát triển bởi JAV (ngôn ngữ Java) trong hệ sinh thái Spring Framework. Nó giúp cho lập trình viên đơn giản hóa quá trình xây dựng một ứng dụng với Spring, chỉ cần tập trung vào phát triển bussiness cho ứng dụng. [3]

* + - * 1. Ưu điểm
* Tự động cấu hình
* Opinionated Dependencies, giúp nhà phát triển có thể ngay lập tức xây dựng ứng dụng và tập trung vào phát triển logic cho ứng dụng.
* Embeded Servers
* Độc lập: các ứng dụng Spring Boot có thể được phát triển, thiết lập và chạy độc lập không cần qua nhiều bước phức tạp vì trong mỗi ứng dụng đã chứa đầy đủ các cấu hình cho riêng nó
* Hoạt động mạnh mẽ cùng với các bộ công cụ các của Spring
* Cộng đồng hỗ trợ mạnh mẽ
  + - * 1. Nhược điểm
* Kích thước tệp phát triển lớn do có nhiều thành phần không được sử dụng.
* Chỉ nên sử dụng cho các ứng dụng microservices, các ứng dụng vừa và nhỏ, không phù hợp cho các ứng dụng nguyên khối quy mô lớn
  + - 1. Spring Data JPA
         1. Định nghĩa

Spring Data JPA là một bản ghi chi tiết của Java để quản lý dữ liệu quan hệ trong các ứng dụng Java. Nó cho phép truy cập và lưu trữ dữ liệu giữa các object/class Java và các tables trong database quan hệ. JPA tuân theo Object Relation Mapping (ORM) và là một tập các interface.

JPA cung cấp một API Entity manager runtime để xử lý các câu query và transaction trên các object dựa trên database. Nó sử dụng ngôn ngữ truy vấn hướng đối tượng độc lập nền tảng JPQL (Java Persistnt Query Language).

JPA bao gồm 3 phần:

* Java Persistence API
* Object Relational metadata
* Bản thân API, được định nghĩa trong persistence package

JPA không phải là một framework, nó định nghĩa một khái niệm có thể thực thi bởi bất kỳ framework nào. [7]

* + - * 1. Ưu điểm
* Đơn giản, gọn gàng và ít tốn công sức hơn JDBC, SQL và ánh xạ thủ công.
* Thích hợp cho các ứng dụng phức tạp không tuân theo hiệu suất.
* Dữ liệu được biểu thị bằng object/class thay vì bảng và bản ghi.
* Cho phép truy vấn dưới dạng các entity Java thay vì cột và bảng trong SQL.
* Hỗ trợ truy vấn bằng phương thức (Method query)
  + - * 1. Nhược điểm
* Cần nắm rõ cơ chế hoạt động của Spring Data JPA và thiết kế database để viết code hiệu quả.
  + - 1. ReactJS
         1. Định nghĩa
* ReactJS là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook (nay là Meta) được sử dụng để xây dựng giao diện người dùng.
  + - * 1. Ưu điểm
* ReactJS giúp viết code dễ dàng hơn với JSX
* Có nhiều công cụ hỗ trợ cho việc phát triển
* Thư viện và cộng đồng hỗ trợ lớn
* Hỗ trợ cả Server Side Rendering và Client Side Rendering
* Dễ bảo trì và phát triển
  + - * 1. Nhược điểm
* ReactJS chỉ phục vụ cho tầng view
* ReactJS chỉ là View Library, không phải MVC Framework nên không có Model và Controller, cần kết hợp với các thư viện khác
* Cần cấu hình lại khi tích hợp ReactJS vào các Framework MVC truyền thống
  1. XÁC ĐỊNH VÀ MÔ HÌNH HÓA YÊU CẦU
     1. Lập danh sách yêu cầu

Bảng yêu cầu chức năng:

* + 1. Mô hình hóa yêu cầu
       1. Lược đồ Use-Case



Lược đồ tổng quan Use case

* + - 1. Đặc tả Use-Case
         1. Mô tả Use-Case “Đăng ký”

Bảng mô tả Use-Case “Đăng ký”

|  |  |
| --- | --- |
| **Tiêu đề** | **Nội dung** |
| Tên Use-Case | Dang ky |
| Mô tả | Cho phép người dùng đăng ký tài khoản để đăng nhập vào hệ thống. |
| Actor | Khách vãng lai |
| Điều kiện kích hoạt | Khi người dùng chọn chức năng đăng ký từ trang chủ của hệ thống. |
| Tiền điều kiện | Không có |
| Hậu điều kiện | Đăng ký thành công |
| Luồng sự kiện chính | 1. Hệ thống hiển thị màn hình đăng ký.  2. Người dùng nhập thông tin đăng ký.  3. Hệ thống kiểm tra thông tin đăng ký.  4. Nếu thông tin đăng ký hợp lệ, hệ thống thông báo đăng ký thành công và chuyển hướng đến trang đăng nhập.  5. Kết thúc Use-Case. |
| Luồng sự kiện phụ | A – Username đã tồn tại  1. Hệ thống hiển thị thông báo đăng ký thất bại.  2. Quay lại bước 2 trong luồng sự kiện chính. |

* + - * 1. Mô tả Use-Case “Đăng nhập”

Bảng mô tả Use-Case “Đăng nhập”

|  |  |
| --- | --- |
| **Tiêu đề** | **Nội dung** |
| Tên Use-Case | Dang nhap |
| Mô tả | Cho phép người dùng đăng nhập vào hệ thống để thực hiện các chức năng tương ứng với quyền hạn của mình. |
| Actor | Người dùng cơ bản, Người kiểm duyệt, Quản trị viên |
| Điều kiện kích hoạt | Khi người dùng chọn chức năng đăng nhập từ trang chủ của hệ thống. |
| Tiền điều kiện | Đã có tài khoản tương ứng trong hệ thống. |
| Hậu điều kiện | Đăng nhập thành công |
| Luồng sự kiện chính | 1. Hệ thống hiển thị màn hình đăng nhập.  2. Người dùng nhập thông tin đăng nhập.  3. Hệ thống kiểm tra thông tin đăng nhập.  4. Nếu thông tin đăng nhập không hợp lệ, hệ thống thông báo đăng nhập thất bại.  5. Kết thúc Use-Case. |
| Luồng sự kiện phụ |  |

* + - * 1. Mô tả Use-Case “Đăng xuất”

Bảng mô tả Use-Case “Đăng xuất”

|  |  |
| --- | --- |
| **Tiêu đề** | **Nội dung** |
| Tên Use-Case | Dang xuat |
| Mô tả | Cho phép người dùng đăng xuất khỏi hệ thống. |
| Actor | Người dùng cơ bản, Người kiểm duyệt, Quản trị viên |
| Điều kiện kích hoạt | Khi người dùng chọn đăng xuất từ menu cá nhân tại ảnh đại diện. |
| Tiền điều kiện | Đã đăng nhập vào hệ thống. |
| Hậu điều kiện | Đăng xuất thành công. |
| Luồng sự kiện chính | 1. Người dùng click đăng xuất  2. Hệ thống xóa thông tin phiên đăng nhập của người dùng khỏi hệ thống.  3. Kết thúc Use-Case. |
| Luồng sự kiện phụ |  |

* + - * 1. Mô tả Use-Case “Xem danh sách bài viết”

Bảng mô tả Use-Case “Xem danh sách bài viết”

|  |  |
| --- | --- |
| **Tiêu đề** | **Nội dung** |
| Tên Use-Case | Xem danh sach bai viet |
| Mô tả | Cho phép người dùng xem danh sách các bài viết có trong hệ thống. |
| Actor | Khách vãng lai, Người dùng cơ bản, Người kiểm duyệt, Quản trị viên |
| Điều kiện kích hoạt | Khi người dùng truy cập thành công vào trang chủ của hệ thống. |
| Tiền điều kiện | Không có. |
| Hậu điều kiện | Hiển thị danh sách các bài viết. |
| Luồng sự kiện chính | 1. Hệ thống hiển thị danh sách các bài viết tương ứng với các danh mục.  2. Kết thúc Use-Case. |
| Luồng sự kiện phụ |  |

* + - * 1. Mô tả Use-Case “Xem chi tiết bài viết”

Bảng mô tả Use-Case “Xem chi tiết bài viết”

|  |  |
| --- | --- |
| **Tiêu đề** | **Nội dung** |
| Tên Use-Case | Xem chi tiet bai viet |
| Mô tả | Cho phép người dùng xem nội dung của bài viết và các thông tin liên quan của bài viết. |
| Actor | Khách vãng lai, Người dùng cơ bản, Người kiểm duyệt, Quản trị viên |
| Điều kiện kích hoạt | Khi người dùng chọn xem một bài viết. |
| Tiền điều kiện | Không có |
| Hậu điều kiện | Hiển thị nội dung chi tiết bài viết |
| Luồng sự kiện chính | 1. Hệ thống hiển thị nội dung bài viết cùng các thông tin liên quan.  2. Kết thúc Use-Case. |
| Luồng sự kiện phụ |  |

* + - * 1. Mô tả Use-Case “Xem bình luận bài viết”

Bảng mô tả Use-Case “Xem bình luận bài viết”

|  |  |
| --- | --- |
| **Tiêu đề** | **Nội dung** |
| Tên Use-Case | Xem binh luan bai viet |
| Mô tả | Cho phép người dùng xem bình luận của mọi người đối với bài viết. |
| Actor | Khách vãng lai, Người dùng cơ bản, Người kiểm duyệt, Quản trị viên |
| Điều kiện kích hoạt | Khi người dùng chọn xem một bài viết. |
| Tiền điều kiện | Người dùng bấm vào xem bài viết. |
| Hậu điều kiện | Hiển thị danh sách các bình luận của bài viết được chọn. |
| Luồng sự kiện chính | 1. Hệ thống hiển thị nội dung bài viết.  2. Người dùng cuộn xuống cuối bài viết để xem bình luận.  3. Kết thúc Use-Case. |
| Luồng sự kiện phụ |  |

* + - * 1. Mô tả Use-Case “Xem đánh giá bài viết”

Bảng mô tả Use-Case “Xem đánh giá bài viết”

|  |  |
| --- | --- |
| **Tiêu đề** | **Nội dung** |
| Tên Use-Case | Xem danh gia bai viet |
| Mô tả | Cho phép người dùng xem đánh giá của mọi người đối với bài viết. |
| Actor | Khách vãng lai, Người dùng cơ bản, Người kiểm duyệt, Quản trị viên |
| Điều kiện kích hoạt | Khi người dùng chọn xem một bài viết. |
| Tiền điều kiện | Người dùng bấm vào xem bài viết. |
| Hậu điều kiện | Hiển thị số điểm được đánh giá của bài viết. |
| Luồng sự kiện chính | 1. Hệ thống hiển thị nội dung bài viết.  2. Người dùng xem thông tin bài viết tại phần tiêu đề và thông tin bài viết.  3. Kết thúc Use-Case |
| Luồng sự kiện phụ |  |

* + - * 1. Mô tả Use-Case “Đánh giá bài viết”

Bảng mô tả Use-Case “Đánh giá bài viết”

|  |  |
| --- | --- |
| **Tiêu đề** | **Nội dung** |
| Tên Use-Case | Danh gia bai viet |
| Mô tả | Cho phép người dùng đánh giá bài viết, cùng với các đánh giá từ nhiều người dùng khác giúp đánh giá được chất lượng bài viết. |
| Actor | Người dùng cơ bản, Người kiểm duyệt, Quản trị viên |
| Điều kiện kích hoạt | Khi người dùng chọn xem một bài viết. |
| Tiền điều kiện | Đã đăng nhập vào hệ thống. Người dùng bấm vào xem bài viết. |
| Hậu điều kiện | Người dùng gửi thành công đánh giá của mình cho bài viết được chọn. |
| Luồng sự kiện chính | 1. Hệ thống hiển thị nội dung bài viết.  2. Người dùng click “lên” để cộng 1 điểm, click “xuống” để trừ 1 điểm”.  3. Hệ thống thay đổi trạng thái đánh giá của người dùng đối với bài viết và tổng số điểm của bài viết.  4. Kết thúc Use-Case. |
| Luồng sự kiện phụ | A – Người dùng đã đánh giá trước đó và muốn đánh giá lại  1. Người dùng click lại vào trạng thái đã đánh giá trước đó để hủy.  2. Quay lại bược 2 trong luồng sự kiện chính. |

* + - * 1. Mô tả Use-Case “Bình luận bài viết”

Bảng mô tả Use-Case “Bình luận bài viết”

|  |  |
| --- | --- |
| **Tiêu đề** | **Nội dung** |
| Tên Use-Case | Binh luan bai viet |
| Mô tả | Cho phép người dùng để lại bình luận đánh giá bài viết hoặc cùng nhau thảo luận về chủ để bài viết cùng với các người dùng khác. |
| Actor | Người dùng cơ bản, Người kiểm duyệt, Quản trị viên |
| Điều kiện kích hoạt | Khi người dùng chọn bình luận vào bài viết. |
| Tiền điều kiện | Đã đăng nhập vào hệ thống. Người dùng chọn xem bài viết và để lại bình luận. |
| Hậu điều kiện | Hiển thị bình luận vừa gửi lên danh sách bình luận của bài viết. |
| Luồng sự kiện chính | 1. Hệ thống hiển thị nội dung bài viết, các bình luận trước đó và nơi nhập bình luận mới.  2. Người dùng nhập nội dung bình luận.  3. Người dùng click gửi bình luận.  4. Hệ thống hiển thị bình luận mới lên danh sách bình luận.  5. Kết thúc Use-Case. |
| Luồng sự kiện phụ |  |

* + - * 1. Mô tả Use-Case “Báo cáo bài viết”

Bảng mô tả Use-Case “Báo cáo bài viết”

|  |  |
| --- | --- |
| **Tiêu đề** | **Nội dung** |
| Tên Use-Case | Bao cao bai viet |
| Mô tả | Cho phép người dùng báo cáo một bài viết nếu cảm thấy bài viết không hợp lệ hay có các nội dung nhạy cảm, sai lệch văn hóa, nội dung, … |
| Actor | Người dùng cơ bản, Người kiểm duyệt, Quản trị viên |
| Điều kiện kích hoạt | Khi người dùng chọn xem nội dung bài viết và chọn chức năng báo cáo bài viết. |
| Tiền điều kiện | Đã đăng nhập vào hệ thống. Người dùng chọn xem nội dung bài viết. |
| Hậu điều kiện | Người dùng gửi thành công báo cáo bài viết đến hệ thống. |
| Luồng sự kiện chính | 1. Hệ thống hiển thị màn hình nhập thông tin nội dung báo cáo.  2. Người dùng nhập nội dung báo cáo.  3. Người dùng chọn gửi báo cáo.  4. Hệ thống lưu thông tin báo cáo về bài viết của người dùng.  5. Hệ thống đóng màn hình báo cáo bài viết và quay về màn hình nội dung bài viết.  6. Kết thúc Use-Case. |
| Luồng sự kiện phụ | A – Người dùng hủy bỏ thao tác báo cáo  1. Người dùng click chọn hủy bỏ.  2. Quay lại bước 5 trong luồng sự kiện chính. |

* + - * 1. Mô tả Use-Case “Đăng bài viết”

Bảng mô tả Use-Case “Đăng bài viết”

|  |  |
| --- | --- |
| **Tiêu đề** | **Nội dung** |
| Tên Use-Case | Dang bai viet |
| Mô tả | Cho phép người dùng đăng bài viết để chia sẻ đến mọi người. |
| Actor | Người dùng cơ bản, Người kiểm duyệt, Quản trị viên |
| Điều kiện kích hoạt | Khi người dùng chọn chức năng đăng bài viết trong hệ thống. |
| Tiền điều kiện | Đã đăng nhập vào hệ thống. Chọn chức năng đăng bài viết. |
| Hậu điều kiện | Bài viết được đăng tải lên hệ thống và hiển thị ở trang chủ hệ thống. |
| Luồng sự kiện chính | 1. Hệ thống hiển thị màn hình đăng bài viết.  2. Người dùng nhập và tải lên các các thông tin, nội dung về bài viết.  3. Người dùng click chọn đăng bài viết.  4. Hệ thống thông báo đăng bài viết thành công.  5. Hệ thống hiển thị màn hình đăng bài viết để người dùng tiếp tục đăng bài viết khác nếu muốn. (Quay lại bước 1)  6. Kết thúc Use-Case. |
| Luồng sự kiện phụ | A – Người dùng hủy bỏ thao tác đăng bài viết  1. Người dùng click vào logo hệ thống để quay lại màn hình trang chủ hệ thống.  2. Quay lại bước 6 trong luồng sự kiện chính. |

* + - * 1. Mô tả Use-Case “Chỉnh sửa bài viết sở hữu”

Bảng mô tả Use-Case “Chỉnh sửa bài viết sở hữu”

|  |  |
| --- | --- |
| **Tiêu đề** | **Nội dung** |
| Tên Use-Case | Chinh sua bai viet so huu |
| Mô tả | Cho phép người dùng chỉnh sửa nội dung bài viết bản thân đã đăng trước đó. |
| Actor | Người dùng cơ bản, Người kiểm duyệt, Quản trị viên |
| Điều kiện kích hoạt | Khi người dùng chọn chức năng chỉnh sửa bài viết từ màn hình nôi dung bài viết của bài viết bản thân đã đăng trước đó. |
| Tiền điều kiện | Đã đăng nhập vào hệ thống. Bài viết phải thuộc sở hữu của người dùng. |
| Hậu điều kiện | Nội dung bài viết được thay đổi và cập nhật lên hệ thống. |
| Luồng sự kiện chính | 1. Hệ thống hiển thị màn hình thông tin, nội dung cũ của bài viết.  2. Người dùng nhập thông tin mới.  3. Người dùng click chọn cập nhật bài viết.  4. Hệ thống thông báo thành công.  5. Hệ thống đóng màn hình chỉnh sửa bài viết và quay về màn hình nội dung bài viết trước đó.  6. Nội dung bài viết được cập nhật lên hệ thống.  7. Kết thúc Use-Case. |
| Luồng sự kiện phụ | A – Người dùng hủy bỏ thao tác cập nhật bài viết.  1. Người dùng click vào logo hệ thống để quay lại màn hình trang chủ hệ thống.  2. Quay lại bước 7 trong luồng sự kiện chính. |

* + - * 1. Mô tả Use-Case “Xóa bài viết sở hữu”

Bảng mô tả Use-Case “Xóa bài viết sở hữu”

|  |  |
| --- | --- |
| **Tiêu đề** | **Nội dung** |
| Tên Use-Case | Xoa bai viet so huu |
| Mô tả | Cho phép người dùng xóa bài viết bản thân đã đăng trước đó. |
| Actor | Người dùng cơ bản, Người kiểm duyệt, Quản trị viên |
| Điều kiện kích hoạt | Khi người dùng chọn chức năng xóa bài viết từ trang nội dung bài viết được chọn. |
| Tiền điều kiện | Đã đăng nhập vào hệ thống. Bài viết phải thuộc sở hữu của người dùng. |
| Hậu điều kiện | Bài viết bị xóa khỏi hệ thống. |
| Luồng sự kiện chính | 1. Hệ thống hiển thị nút xóa bài viết trong phần thông tin bài viết tại trang nội dung bài viết.  2. Người dùng click chọn xóa bài viết.  3. Hệ thống thông báo thành công và bài viết bị xóa khỏi hệ thống.  4. Hệ thống quay lại màn hình nội dung bài viết trước đó.  5. Kết thúc Use-Case. |
| Luồng sự kiện phụ |  |

* + - * 1. Mô tả Use-Case “Xem danh sách báo cáo bài viết”

Bảng mô tả Use-Case “Xem danh sách báo cáo bài viết”

|  |  |
| --- | --- |
| **Tiêu đề** | **Nội dung** |
| Tên Use-Case | Xem danh sach bao cao bai viet |
| Mô tả | Cho phép người dùng xem các báo cáo về các bài viết của người dùng trong hệ thống từ đó giúp phát hiện các bài viết không hợp lệ, vi phạm để đưa ra quyết định xóa bài. |
| Actor | Người kiểm duyệt, Quản trị viên |
| Điều kiện kích hoạt | Người dùng chọn chức năng xem danh sách báo cáo bài viết trong hệ thống. |
| Tiền điều kiện | Đã đăng nhập vào hệ thống. |
| Hậu điều kiện | Hệ thống hiển thị danh sách các báo cáo về các bài viết sắp xếp theo thời gian gần nhất. |
| Luồng sự kiện chính | 1. Hệ thống hiển thị danh sách các báo cáo về các bài viết.  2. Người dùng chọn xem các báo cáo khác của cùng bài viết.  3. Hệ thống hiển thị tất cả các báo cáo của cùng bài viết được chọn.  4. Kết thúc Use-Case |
| Luồng sự kiện phụ | A – Người dùng chọn: dẫn tới nội dung bài viết  1. Hệ thống hiển thị trang nội dung của bài viết được chọn cùng các thông tin liên quan.  2. Quay lại bước 4 trong luồng sự kiện chính. |

* + - * 1. Mô tả Use-Case “Xóa bài viết của người dùng khác”

Bảng mô tả Use-Case “Xóa bài viết của người dùng khác”

|  |  |
| --- | --- |
| **Tiêu đề** | **Nội dung** |
| Tên Use-Case | Xoa bai viet cua nguoi dung khac |
| Mô tả | Cho phép người dùng xóa bài viết trong hệ thống. |
| Actor | Người kiểm duyệt, Quản trị viên. |
| Điều kiện kích hoạt | Người dùng chọn chức năng xóa bài viết từ trang nội dung bài viết của bài viết được chọn. |
| Tiền điều kiện | Đã đăng nhập vào hệ thống. |
| Hậu điều kiện | Bài viết bị xóa khỏi hệ thống. |
| Luồng sự kiện chính | 1. Hệ thống hiển thị nút xóa bài viết trong phần thông tin bài viết tại trang nội dung bài viết.  2. Người dùng click chọn xóa bài viết.  3. Hệ thống thông báo thành công và bài viết bị xóa khỏi hệ thống.  4. Hệ thống quay lại màn hình nội dung bài viết trước đó.  5. Kết thúc Use-Case. |
| Luồng sự kiện phụ |  |

* + - * 1. Mô tả Use-Case “Thêm tài khoản người kiểm duyệt”

Bảng mô tả Use-Case “Thêm tài khoản người kiểm duyệt”

|  |  |
| --- | --- |
| **Tiêu đề** | **Nội dung** |
| Tên Use-Case | Them tai khoan nguoi kiem duyet |
| Mô tả | Cho phép người dùng thêm tài khoản Người kiểm duyệt vào hệ thống. |
| Actor | Quản trị viên |
| Điều kiện kích hoạt | Người dùng chọn chức năng thêm Người kiểm duyệt trong trang quản lý Người kiểm duyệt. |
| Tiền điều kiện | Đã đăng nhập vào hệ thống. |
| Hậu điều kiện | Tài khoản người kiểm duyệt được thêm vào hệ thống. |
| Luồng sự kiện chính | 1. Hệ thống hiển thị màn hình nhập thông tin cho tài khoản Người kiểm duyệt mới.  2. Người dùng nhập thông tin tài khoản.  3. Người dùng click xác nhận tạo tài khoản Người kiểm duyệt mới.  3. Hệ thống kiểm tra thông tin đăng ký.  4. Nếu thông tin đăng ký hợp lệ, hệ thống thông báo thành công.  5. Kết thúc Use-Case. |
| Luồng sự kiện phụ | A – Username đã tồn tại  1. Hệ thống hiển thị thông báo tạo thất bại.  2. Quay lại bước 2 trong luồng sự kiện chính. |

* + - * 1. Mô tả Use-Case “Xem danh sách tài khoản người kiểm duyệt”

Bảng mô tả Use-Case “Xem danh sách tài khoản người kiểm duyệt”

|  |  |
| --- | --- |
| **Tiêu đề** | **Nội dung** |
| Tên Use-Case | Xem danh sach tai khoan nguoi kiem duyet |
| Mô tả | Cho phép người dùng xem danh sách các tài khoản Người kiểm duyệt có trong hệ thống để thực hiện các thao tác quản lý khác. |
| Actor | Quản trị viên |
| Điều kiện kích hoạt | Người dùng chọn chức năng xem danh sách Người kiểm duyệt. |
| Tiền điều kiện | Đã đăng nhập vào hệ thống. |
| Hậu điều kiện | Hệ thống hiển thị danh sách các tài khoản Người kiểm duyệt. |
| Luồng sự kiện chính | 1. Hệ thống hiển thị danh sách các tài khoản Người kiểm duyệt và các thông tin liên quan.  2. Kết thúc Use-Case |
| Luồng sự kiện phụ |  |

* + - * 1. Mô tả Use-Case “Vô hiệu hóa tài khoản người kiểm duyệt”

Bảng mô tả Use-Case “Vô hiệu hóa tài khoản người kiểm duyệt”

|  |  |
| --- | --- |
| **Tiêu đề** | **Nội dung** |
| Tên Use-Case | Vo hieu hoa tai khoan nguoi kiem duyet |
| Mô tả | Cho phép người dùng vô hiệu hoặc tái kích hoạt hóa tài khoản Người kiểm duyệt được chọn. |
| Actor | Quản trị viên |
| Điều kiện kích hoạt | Người dùng chọn chức năng vô hiệu hóa tài khoản Người kiểm duyệt. |
| Tiền điều kiện | Đã đăng nhập vào hệ thống. Người dùng chọn xem danh sách các tài khoản Người kiểm duyệt. |
| Hậu điều kiện | Tài khoản người kiểm duyệt bị vô hiệu hóa / tái kích hoạt |
| Luồng sự kiện chính | 1. Hệ thống hiển thị danh sách các tài khoản Người kiểm duyệt.  2. Người dùng chọn vô hiệu hóa / kích hoạt tài khoản.  3. Hệ thống cập nhật và hiển thị trạng thái mới của tài khoản.  4. Kết thúc Use-Case. |
| Luồng sự kiện phụ |  |

* + - * 1. Mô tả Use-Case “Thay đổi mật khẩu tài khoản người kiểm duyệt”

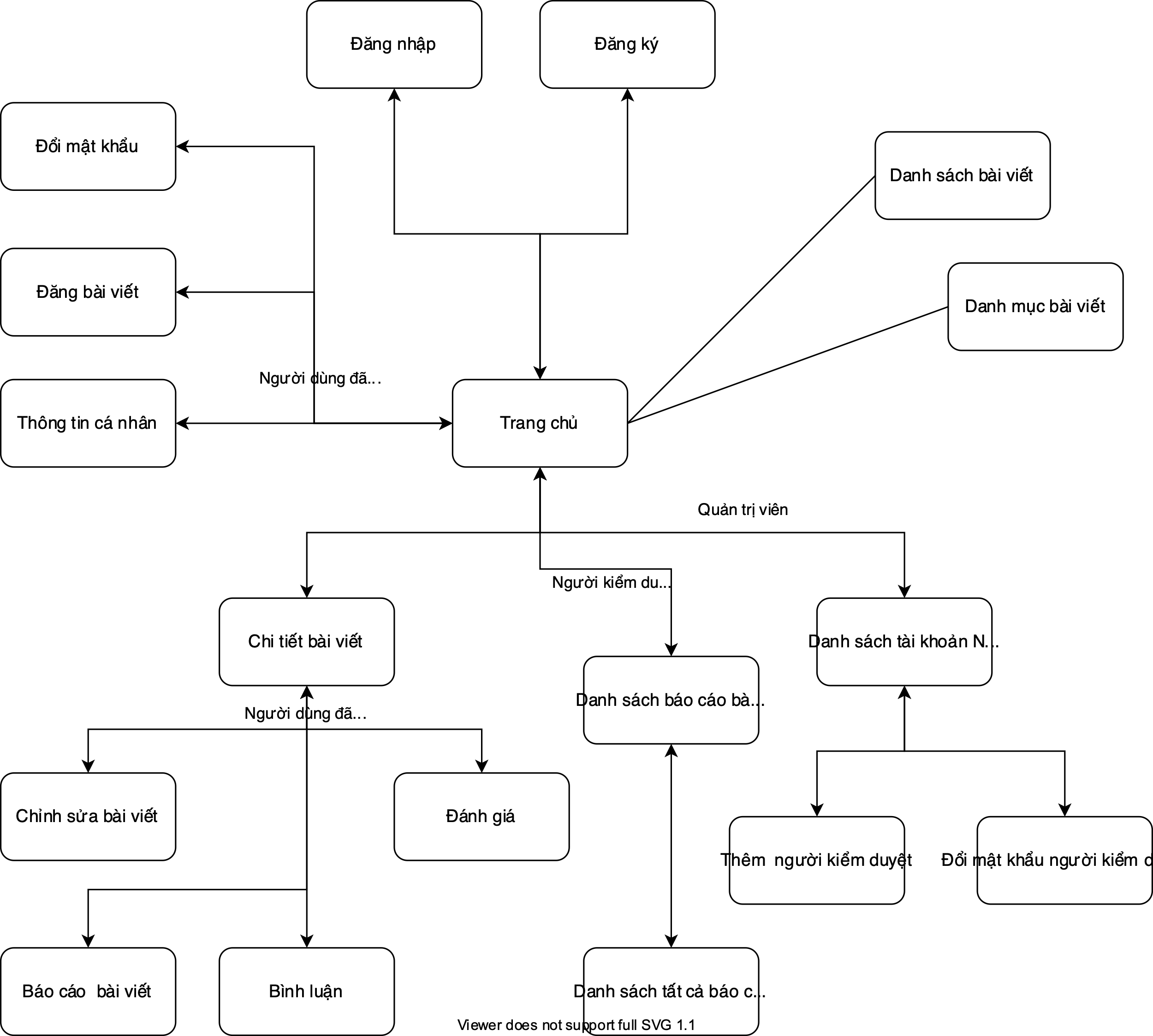
Bảng mô tả Use-Case “Thay đổi mật khẩu người kiểm duyệt”

|  |  |
| --- | --- |
| **Tiêu đề** | **Nội dung** |
| Tên Use-Case | Thay doi mat khau nguoi kiem duyet |
| Mô tả | Cho phép người dùng thay đổi mật khẩu của tài khoản Người kiểm duyệt được chọn. |
| Actor | Quản trị viên |
| Điều kiện kích hoạt | Khi người dùng chọn chức năng thay đổi mật khẩu tài khoản Người kiểm duyệt. |
| Tiền điều kiện | Đã đăng nhập vào hệ thống. Người dùng chọn xem danh sách các tài khoản Người kiểm duyệt. |
| Hậu điều kiện | Mật khẩu của tài khoản Người kiểm duyệt được chọn được thay đổi. |
| Luồng sự kiện chính | 1. Hệ thống hiển thị danh sách các tài khoản Người kiểm duyệt.  2. Người dùng click chọn thay đổi mật khẩu của một tài khoản Người kiểm duyệt.  3. Người dùng nhập mật khẩu mới cho tài khoản.  4. Người dùng click chọn xác nhận thay đổi mật khẩu.  5. Hệ thống thông báo thành công và mật khẩu được cập nhật lên hệ thống.  6. Kết thúc Use-Case. |
| Luồng sự kiện phụ |  |

* 1. THIẾT KẾ HỆ THỐNG
     1. Thiết kế cơ sở dữ liệu

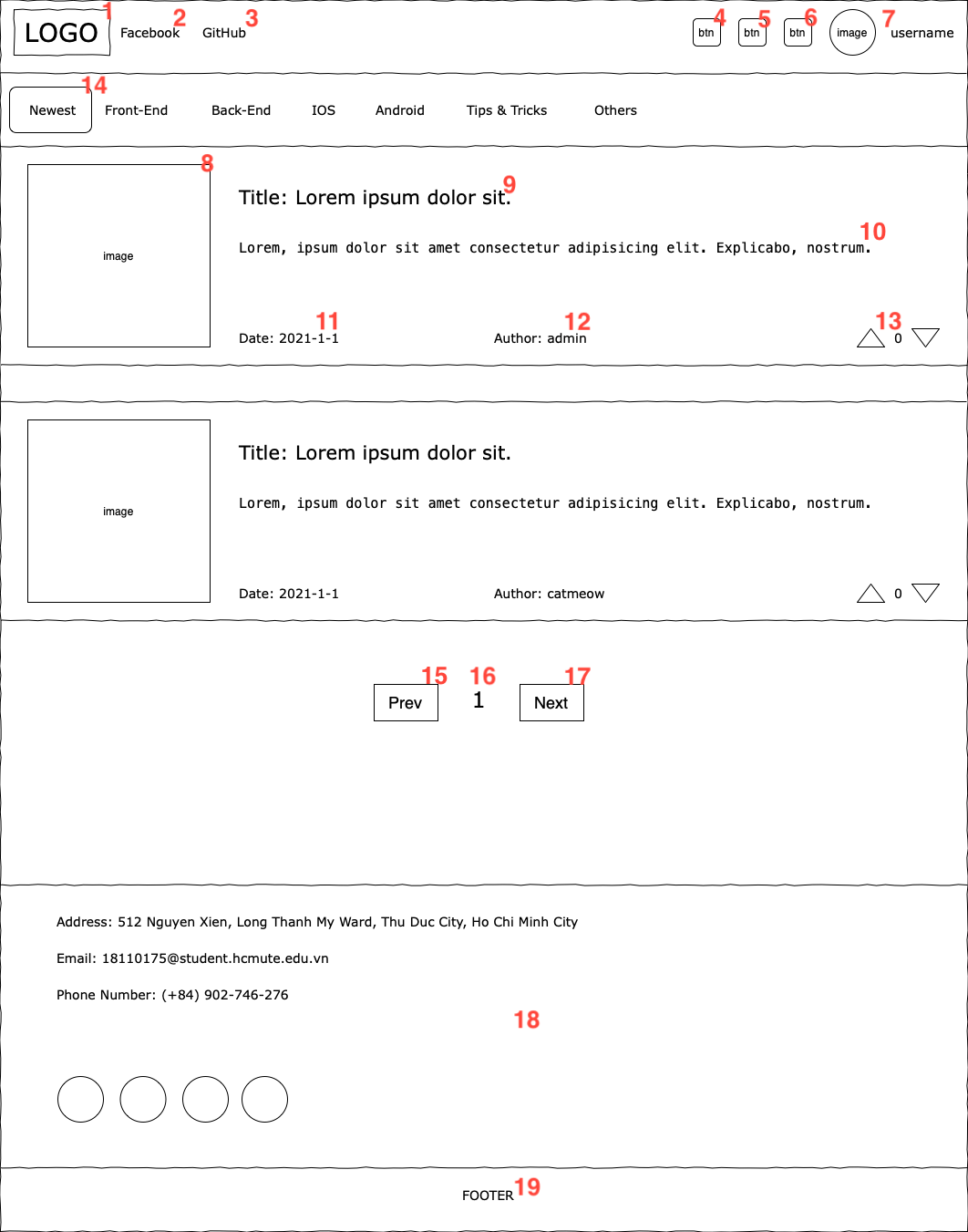
Thiết kế cơ dữ liệu

* + 1. Thiết kế giao diện



Sơ đồ màn hình

* + - 1. Màn hình chính



Hình màn hình chính

* + 1. Các lược đồ sequence chính
       1. Đăng ký tài khoản



Lược đồ sequence đăng ký tài khoản

**Mô tả hoạt động:**

* Người dùng vãng lai đăng ký tài khoản Người dùng cơ bản để thực hiện nhiều tính năng tương tác với hệ thống.
* Nhập thông tin username và mật khẩu.
* Gửi thông tin đăng ký lên hệ thống, hệ thống kiểm tra liệu username đã tồn tại.
* Nếu username chưa tồn tại, lưu thông tin tài khoản mới vào database và trả về thông báo lưu thành công, thông báo cho người dùng đăng ký thành công.
* Nếu username đã tồn tại, thông báo cho người dùng đăng ký thất bại.
  + - 1. Đăng nhập



Lược đồ sequence đăng nhập

**Mô tả hoạt động:**

* Người dùng đăng nhập hệ thống với username và mật khẩu của mình.
* Nhập thông tin username và mật khẩu.
* Gửi thông tin đăng nhập lên Back-End.
* Back-End truy vấn thông tin người dùng từ database với username.
* Nếu thông tin khớp với database, trả về token và thông báo đăng nhập thành công.
* Nếu thông tin không trùng khớp với database, trả về thông báo đăng nhập thất bại.
  + - 1. Đăng bài viết



Lược đồ sequence đăng bài viết

**Mô tả hoạt động:**

* Người dùng đăng nhập với username và mật khẩu của mình.
* Chọn chức năng đăng bài viết và gửi thông tin bài viết gồm tiêu đề, mô tả ngắn gọn, nội dung văn bản (markdown), ảnh tượng trưng bài viết, tệp âm thanh bài viết lên Back-End.
* Back-End lưu ảnh tượng trưng và tệp âm thanh vào StorageService và trả về thông tin đã lưu.
* Back-End gửi thông tin bài viết cùng với thông tin đã lưu của ảnh và âm thanh xuống database.
* Database thông báo lưu thành công.
* Back-End trả về thông báo lưu thành công bài viết cho người dùng.
  + - 1. Chỉnh sửa bài viết



Lược đồ sequence chỉnh sửa bài viết

**Mô tả hoạt động:**

* Người dùng đăng nhập với username và mật khẩu.
* Bài viết thuộc sở hữu của người dùng hoặc người dùng là admin.
* Người dùng gửi thông tin cập nhật bài viết lên Back-End.
* Back-End gửi ảnh tượng trưng và tệp âm thanh mới đến StorageService.
* StorageService lưu ảnh, tệp âm thanh và trả về thông tin đã lưu.
* Back-End gửi thông tin bài viết cùng thông tin ảnh, âm thanh đã lưu đến database.
* Database lưu thông tin bài viết và trả về thông báo lưu thành công.
* Back-End gửi thông báo lưu thành công về người dùng.
  + - 1. Gửi báo cáo bài viết



Lược đồ sequence gửi báo cáo bài viết

**Mô tả hoạt động:**

* Người dùng đăng nhập với username và mật khẩu.
* Người dùng gửi thông tin báo cáo bài viết đến Back-End.
* Back-End gửi thông tin báo cáo đến database.
* Database lưu thông tin báo cáo về bài viết và gửi thông báo lưu thành công đến Back-End.
* Back-End gửi thông báo thành công đến người dùng.
  1. CÀI ĐẶT VÀ THỬ NGHIỆM
     1. Cài đặt
        1. Cài đặt Database
* Cài đặt và cấu hình MS SQL Server
  + Cách 1: cài trực tiếp trên thiết bị (Windows hoặc Linux đều được Microsoft hỗ trợ cài native)
  + Cách 2: cài qua Docker (macOS không được Microsoft hỗ trợ bản cài native)
    - Cài đặt và cấu hình Docker
    - Mở powershell trên Windows hoặc terminal trên Linux hoặc macOS.
    - Thực thi lệnh:  
      *sudo docker run -e "ACCEPT\_EULA=Y" -e "SA\_PASSWORD=<YourStrong@Passw0rd>" \  
      -p 1433:1433 --name sql1 -h sql1 \  
      -d mcr.microsoft.com/mssql/server:2019-latest*
    - Thay đổi các thông tin tương ứng với database mong muốn như port, container’s name, … Nếu không cần thay đổi nhiều thông tin, ta thay đổi *=<YourStrong@Passw0rd>* thành mật khẩu sa cho database. [6]
* Kết nối đến database server bằng các công cụ như SSMS hoặc Azure Data Studio (Microsoft) hoặc DBeaver, … để thực thi query trong tệp ./DB\_SQL/full\_scripts.sql
* Cài đặt database hoàn tất.
  + - 1. Cài đặt Backend
* Cài đặt và cấu hình JDK 11.
* Cài đặt và cấu hình Maven.
* Kiểm tra port 8080 trên máy chủ cài đặt có thể hoạt động.
* Thư mục hiện tại là thư mục gốc của project.
* Chỉnh sửa tệp ./BE/audiosn/src/main/resources/application.properties
  + Thay đổi giá trị của: *file.upload-dir* từ */Users/dinhphu/tmp/test\_be\_file/savedfile* thành đường dẫn thư mục lưu trữ trên máy chủ cần cài đặt.
  + Thay đổi giá trị của: *spring.datasource.url* từ *jdbc:sqlserver://localhost:1433;databaseName=audiosn* thành thông tin kết nối đến database server.
    - Thay đổi *localhost* thành địa chỉ ip hoặc tên miền của database server.
    - Thay đổi *1433* thành port của database server tương ứng (mặc định là 1433)
    - Thay đổi giá trị databaseName từ *audiosn* thành giá trị database mới tương ứng (nếu cần đổi tên).
* Truy cập đường dẫn ./BE/audiosn/
* Mở powershell trên Windows hoặc terminal trên Linux hoặc macOS).
* Thực thi các lệnh sau:
  + *mvn clean -f pom.xml*
  + *mvn package -f pom.xml*
  + *mvn spring-boot:run*
* Cài đặt Back-End hoàn tất.
  + - 1. Cài đặt FrontEnd
* Cài đặt và cấu hình NodeJS v16.
* Kiểm tra port 3000 trên máy chủ cài đặt có thể hoạt động.
* Cài đặt và cấu hình ***npm*** (hoặc ***yarn***)
* Chỉnh sửa tệp ./FE/zound/.env
  + Thay đổi giá trị của biến môi trường REACT\_APP\_BE\_API\_V1\_URL=*http://localhost:8080/api/v1* từ *localhost:8080* thành địa chỉ ip và port của Back-End hoặc tên miền của Back-End tương ứng (đã cài đặt ở trên hoặc đã deploy).
* Nếu máy chủ không hoạt động với biến môi trường trong .env, có thể thay đổi trực tiếp giá trị BASE\_URL\_API\_BE trong tệp ./FE/zound/src/constants/global.js
* Truy cập đường dẫn ./FE/zound/
* Mở powershell trên Windows hoặc terminal trên Linux hoặc macOS.
* Thực thi các lệnh sau:
  + *yarn install*
  + *yarn start*
* Cài đặt Front-End hoàn tất.
  + 1. Kiểm thử

Điều kiện tiên quyết:

* Mở web browser
* Truy cập: <http://localhost:3000> (hoặc địa chỉ của Front-End tương ứng nếu đã deploy)
  + - 1. Kiểm thứ chức năng đăng nhập

1. PHẦN KẾT LUẬN
2. KẾT QUẢ ĐẠT ĐƯỢC

* Xây dựng trang mạng xã hội chia sẻ kiến thức – có âm thanh với các chức năng chính:
  + Xem danh sách và chi tiết nội dung bài viết
  + Xem, đăng bình các bình luận, thảo luận về bài viết
  + Đánh giá bài viết
  + Báo cáo bài viết
  + Đăng bài viết, sửa, xóa bài viết
  + Đăng ký / Đăng nhập / Đổi mật khẩu
  + Quản lý, kiểm duyệt bài viết (danh sách báo cáo, xóa bài)
  + Quản lý Người kiểm duyệt
  + Chỉnh sửa một số thông tin cá nhân cơ bản
* Hệ thống có tích hợp âm thanh thuận tiên cho người dùng có thể đọc hoặc nghe nội dung mà người đăng bài chia sẻ.
* Tạo dựng môi trường cho mọi người cùng nhau học hỏi, trao đổi, thảo luận và chia sẻ kiến thức. Kết nối con người – con người, con người – kiến thức trong thời buổi dịch bệnh Covid-19 hoành hành gây cản trở việc trao đổi, tìm hiểu kiến thức trực tiếp.
* Xây dựng được một công cụ đơn giản, thuận tiện cho mọi người có thể tìm hiểu kiến thức mọi lúc, mọi nơi dù đang ngồi xếp hàng hay trên xe buýt giúp tối ưu thời gian.

1. ƯU, NHƯỢC ĐIỂM
2. BÀI HỌC KINH NGHIỆM
3. HƯỚNG PHÁT TRIỂN
4. DANH MỤC TÀI LIỆU THAM KHẢO