

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN

LUẬN VĂN TỐT NGHIỆP

Tên đề tài:

XÂY DỰNG ỨNG DỤNG WEB HẸN HÒ
TRÊN LARAVEL

Giáo viên hướng dẫn: **ThS. Trần Văn Hùng**

Sinh viên thực hiện:

- 1. Nguyễn Bạch Nhật Long**
- 2. Lê Văn Hiếu**

TP HỒ CHÍ MINH – Năm 2020

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ SÀI GÒN
KHOA CÔNG NGHỆ THÔNG TIN

LUẬN VĂN TỐT NGHIỆP

Tên đề tài:

**XÂY DỰNG ỨNG DỤNG WEB HẸN HÒ
TRÊN LARAVEL**

Giáo viên hướng dẫn: **ThS. Trần Văn Hùng**

Sinh viên thực hiện:

- 1. Nguyễn Bạch Nhật Long**
- 2. Lê Văn Hiếu**

TP HỒ CHÍ MINH – Năm 2020

BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ SÀI GÒN
TP. HỒ CHÍ MINH

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

KHOA : Công Nghệ Thông Tin

NHIỆM VỤ BÀI THI TỐT NGHIỆP

(Sinh viên phải dán tờ này vào trang thứ nhất của Bản thuyết minh)

HỌ VÀ TÊN : 1/ Nguyễn Bạch Nhật Long..... MSSV: DH51603225 LỚP: D16_TH03 ...

2/ Lê Văn Hiếu..... MSSV: DH51601561 LỚP: D16_TH03 ..

NGÀNH : **Tin học**.....

1. ĐÀU ĐỀ BÀI THI :

Xây dựng ứng dụng web hẹn hò trên Laravel

2. NHIỆM VỤ YÊU CẦU VỀ NỘI DUNG VÀ SỐ LIỆU BAN ĐẦU:

- Tìm hiểu các chức năng của một ứng dụng hẹn hò
- Tìm hiểu Laravel framework
- Xây dựng CSDL
- Xây dựng ứng dụng hẹn hò trên web bằng Laravel

3. CÁC HỒ SƠ VÀ TÀI LIỆU CUNG CẤP BAN ĐẦU :

4. NGÀY GIAO NHIỆM VỤ BÀI THI : 11/05/2020.....

5. NGÀY HOÀN THÀNH NHIỆM VỤ : 29/08/2020.....

6. HỌ TÊN NGƯỜI HƯỚNG DẪN:

PHẦN HƯỚNG DẪN

(1) Trần Văn Hùng.....

Toàn bộ.....

(2)

Nội dung và yêu cầu của Bài thi Tốt nghiệp đã được thông qua Khoa.

Tp. Hồ Chí Minh, ngày ... tháng ... năm 2020

TRƯỞNG KHOA
(Ký và ghi rõ họ tên)

NGƯỜI HƯỚNG DẪN CHÍNH
(Ký và ghi rõ họ tên)

Trần Văn Hùng

LỜI CẢM ƠN

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Trong suốt thời gian từ khi bắt đầu học tập ở giảng đường đại học đến nay, chúng em đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý thầy cô, gia đình và bạn bè.

Với lòng biết ơn sâu sắc nhất, chúng em xin gửi đến quý Thầy Cô ở Khoa Công Nghệ Thông Tin – Trường Đại Học Công Nghệ Sài Gòn với tri thức và tâm huyết của mình để truyền đạt vốn kiến thức quý báu cho chúng em trong suốt thời gian học tập tại trường.

Chúng em xin chân thành cảm ơn Thầy Trần Văn Hùng đã tận tâm hướng dẫn chúng em qua từng buổi học trên lớp cũng như những buổi nói chuyện, thảo luận về lĩnh vực sáng tạo trong nghiên cứu khoa học. Nếu không có những lời hướng dẫn, dạy bảo của thầy thì em nghĩ đề tài này của chúng em rất khó có thể hoàn thiện được. Một lần nữa, chúng em xin chân thành cảm ơn thầy.

Bước đầu đi vào thực tế, tìm hiểu về lĩnh vực sáng tạo trong nghiên cứu khoa học, kiến thức của chúng em còn hạn chế và còn nhiều ngỡ ngàng. Do vậy, không tránh khỏi những thiếu sót là điều chắc chắn, chúng em rất mong nhận được những ý kiến đóng góp quý báu của quý thầy cô và các bạn học cùng lớp để kiến thức của em trong lĩnh vực này được hoàn thiện hơn.

Lời sau cùng, chúng em xin kính chúc quý thầy cô trong khoa Công Nghệ Thông Tin, thầy hiệu trưởng – Cao Hào Thi sức khỏe, niềm tin, vững bước đi đầu cho chúng em trưởng thành.

Trân trọng. Thành phố Hồ Chí Minh, ngày 12 tháng 08 năm 2020

Nhóm sinh viên thực hiện

MỤC LỤC

NHIỆM VỤ BÀI THI TỐT NGHIỆP	4
MỤC LỤC	6
MỤC LỤC CÁC HÌNH VẼ.....	10
MỤC LỤC CÁC BẢNG.....	13
Chương 1. GIỚI THIỆU	15
1.1 ĐẶT VẤN ĐỀ	15
1.2 MỤC TIÊU CỦA ĐỀ TÀI.....	15
1.3 ĐỐI TƯỢNG NGHIÊN CỨU	16
1.4 PHẠM VI NGHIÊN CỨU.....	16
1.5 MỤC ĐÍCH NGHIÊN CỨU.....	16
1.6 NHIỆM VỤ NGHIÊN CỨU.....	16
1.7 PHƯƠNG PHÁP NGHIÊN CỨU	16
1.8 Ý NGHĨA LUẬN VÀ THỰC TIỄN CỦA ĐỀ TÀI	17
Chương 2. MÔ TẢ NGHIỆP VỤ	18
2.1 TỔNG QUAN	18
2.2 QUY TRÌNH THAM GIA HẸN HÒ.....	18
2.3 QUY TRÌNH QUẢN TRỊ HỆ THỐNG	20
2.3.1 Quản trị viên.....	20
2.3.2 Quản trị viên cấp cao.....	20
Chương 3. PHÂN TÍCH CHỨC NĂNG	21
3.1 MODULE USER	21
3.2 MODULE ADMIN	22
Chương 4. GIẢI PHÁP ĐỀ XUẤT	25
4.1 KIẾN TRÚC TỔNG THỂ	25
4.2 MÔ HÌNH XÂY DỰNG.....	26
4.2.1 Mô tả kiến trúc MVC	26
4.2.2 Sự tương tác giữa các thành phần trong mô hình MVC.....	27

4.2.3 Ưu – Nhược điểm của mô hình MVC	27
4.3 GIẢI PHÁP CÔNG NGHỆ	28
4.3.1 Công nghệ xử lý phía Front-end: Framework Bootstrap	28
4.3.2 Công nghệ xử lý phía Back-end: Framework Laravel	29
4.3.3 Hệ quản trị cơ sở dữ liệu: MySQL	30
4.3.4 Giao diện lập trình ứng dụng: API	31
4.4 SƠ ĐỒ CHỨC NĂNG	34
4.4.1 Sơ đồ chức năng tổng quát	34
4.4.2 Sơ đồ chức năng User	35
4.4.3 Sơ đồ chức năng Admin	35
4.5 SƠ ĐỒ USE-CASE TỔNG QUÁT	36
Chương 5. PHÂN TÍCH DỮ LIỆU	37
5.1 PHÂN TÍCH Ở MỨC QUAN NIỆM	37
5.1.1 Sơ đồ ER	37
5.1.2 Mô tả các loại thực thể	38
5.2 THIẾT KẾ CƠ SỞ DỮ LIỆU	42
5.2.1 Sơ đồ logic dữ liệu (mô hình quan hệ)	42
5.2.2 Mô tả các bảng dữ liệu	43
Chương 6. THIẾT KẾ CHỨC NĂNG MODULE USER	47
6.1 CHỨC NĂNG QUẢN LÝ ĐĂNG NHẬP	47
6.1.1 Sơ đồ use case chức năng quản lý đăng nhập	47
6.1.2 Sơ đồ tuần tự chức năng quản lý đăng nhập	50
6.2 CHỨC NĂNG TÌM KIẾM BẠN BÈ	51
6.2.1 Sơ đồ use case chức năng tìm kiếm bạn bè	51
6.2.2 Sơ đồ tuần tự chức năng tìm kiếm bạn bè	53
6.3 CHỨC NĂNG QUẢN LÝ TRANG CÁ NHÂN	54
6.3.1 Sơ đồ use case chức năng quản lý trang cá nhân	54
6.3.2 Sơ đồ tuần tự chức năng quản lý trang cá nhân	56
6.4 CHỨC NĂNG QUẢN LÝ BẠN BÈ	57
6.4.1 Sơ đồ use case chức năng quản lý bạn bè	57
6.4.2 Sơ đồ tuần tự chức năng quản lý bạn bè	59

6.5 CHỨC NĂNG CHAT.....	60
6.5.1 Sơ đồ use case chức năng Chat	60
6.5.2 Sơ đồ tuần tự chức năng Chat	61
6.6 CHỨC NĂNG STATUS.....	62
6.6.1 Sơ đồ use case chức năng Status	62
6.6.2 Sơ đồ tuần tự chức năng Status	64
6.7 CHỨC NĂNG REVIEW	66
6.7.1 Sơ đồ use case chức năng Review.....	66
6.7.2 Sơ đồ tuần tự chức năng Review.....	67
6.8 CHỨC NĂNG LIKE.....	68
6.8.1 Sơ đồ use case chức năng quản lý đăng nhập	68
6.8.2 Sơ đồ tuần tự chức năng quản lý đăng nhập	69
6.9 CHỨC NĂNG BÌNH LUẬN.....	71
6.9.1 Sơ đồ use case chức năng bình luận.....	71
6.9.2 Sơ đồ tuần tự chức năng quản lý đăng nhập	72
 Chương 7. THIẾT KẾ CHỨC NĂNG MODULE ADMIN.....	 74
7.1 CHỨC NĂNG QUẢN LÝ THỂ LOẠI BLOG.....	74
7.1.1 Sơ đồ use case quản lý thể loại blog	74
7.1.2 Sơ đồ tuần tự quản lý thể loại blog	76
7.2 CHỨC NĂNG QUẢN LÝ BLOG.....	78
7.2.1 Sơ đồ use case chức năng quản lý blog.....	78
7.2.2 Sơ đồ tuần tự chức năng quản lý blog.....	80
7.3 CHỨC NĂNG XỬ LÝ PHIẾU LIÊN HỆ	82
7.3.1 Sơ đồ use case chức năng xử lý phiếu liên hệ.....	82
7.3.2 Sơ đồ tuần tự chức năng xử lý phiếu liên hệ.....	83
7.4 CHỨC NĂNG XỬ LÝ ĐÁNH GIÁ.....	84
7.4.1 Sơ đồ use case chức năng xử lý đánh giá.....	84
7.4.2 Sơ đồ tuần tự chức năng xử lý đánh giá.....	85
7.5 CHỨC NĂNG QUẢN LÝ NGƯỜI DÙNG	87
7.5.1 Sơ đồ use case chức năng quản lý người dùng	87
7.5.2 Sơ đồ tuần tự chức năng quản lý người dùng.....	88
7.6 CHỨC NĂNG QUẢN LÝ QUẢN TRỊ.....	90

7.6.1 Sơ đồ use case quản lý quản trị	90
7.6.2 Sơ đồ tuần tự quản lý quản trị	91
Chương 8. THÀNH PHẦN GIAO DIỆN	93
8.1 GIAO DIỆN ĐĂNG NHẬP	93
8.2 GIAO DIỆN TRANG CHỦ	94
8.3 GIAO DIỆN USER	96
8.4 GIAO DIỆN ADMIN	100
Chương 9. KẾT LUẬN	104
9.1 KẾT QUẢ ĐẠT ĐƯỢC	104
9.2 NHỮNG CHỨC NĂNG CHƯA HOÀN THÀNH	105
9.3 HƯỚNG PHÁT TRIỂN	105
TÀI LIỆU THAM KHẢO	106

MỤC LỤC CÁC HÌNH VẼ

Hình 4-1: Kiến trúc tổng thể của hệ thống.....	25
Hình 4-2: Mô hình xây dựng của hệ thống.....	26
Hình 4-3: Laravel theo mô hình MVC	29
Hình 4-4: Sơ đồ chức năng tổng quát của hệ thống	34
Hình 4-5: Sơ đồ chức năng của User	35
Hình 4-6: Sơ đồ chức năng Admin	35
Hình 5-1: Sơ đồ ER của hệ thống.....	37
Hình 5-2: Lược đồ quan hệ cơ sở dữ liệu.....	42
Hình 6-1: Sơ đồ usecase chức năng đăng nhập.....	47
Hình 6-2: Sơ đồ tuần tự chức năng Đăng nhập.....	50
Hình 6-3: Sơ đồ tuần tự chức năng Đăng ký.....	50
Hình 6-4: Sơ đồ chức năng Quên mật khẩu	51
Hình 6-5: Sơ đồ use case tìm kiếm bạn bè	51
Hình 6-6: Sơ đồ tuần tự chức năng tìm kiếm theo tên	53
Hình 6-7: Sơ đồ tuần tự chức năng tìm kiếm theo sở thích	53
Hình 6-8: Sơ đồ use case quản lý trang cá nhân.....	54
Hình 6-9: Sơ đồ tuần tự chức năng Đổi ảnh đại diện.....	56
Hình 6-10: Sơ đồ tuần tự chức năng Cập nhật thông tin.....	56
Hình 6-11: Sơ đồ tuần tự chức năng Đổi mật khẩu.....	57
Hình 6-12: Sơ đồ use case quản lý bạn bè	57
Hình 6-13: Sơ đồ tuần tự chức năng Kết bạn.....	59
Hình 6-14: Sơ đồ tuần tự chức năng Hủy kết bạn.....	59
Hình 6-15: Sơ đồ use case tìm kiếm bạn bè	60
Hình 6-16: Sơ đồ tuần tự chức năng Chat với bạn bè	61
Hình 6-17: Sơ đồ use case Status	62
Hình 6-18: Sơ đồ tuần tự chức năng Đăng status.....	64
Hình 6-19: Sơ đồ tuần tự chức năng Xóa status.....	64
Hình 6-20: Sơ đồ tuần tự chức năng Sửa status	65

Hình 6-21: Sơ đồ use case Review	66
Hình 6-22: Sơ đồ tuần tự chức năng Gửi đánh giá	67
Hình 6-23: Sơ đồ usecase chức năng Like	68
Hình 6-24: Sơ đồ tuần tự chức năng Thích status	69
Hình 6-25: Sơ đồ tuần tự chức năng Hủy thích status	70
Hình 6-26: Sơ đồ chức năng Đếm số lượt thích.....	70
Hình 6-27: Sơ đồ usecase chức năng bình luận	71
Hình 6-28: Sơ đồ tuần tự chức năng Xem bình luận.....	72
Hình 6-29: Sơ đồ tuần tự chức năng Phản hồi bình luận	73
Hình 7-1: Sơ đồ use case chức năng quản lý thể loại blog	74
Hình 7-2: Sơ đồ tuần tự chức năng Thêm thể loại blog	76
Hình 7-3: Sơ đồ tuần tự chức năng Xóa thể loại Blog	76
Hình 7-4: Sơ đồ tuần tự chức năng Sửa thể loại Blog.....	77
Hình 7-5: Sơ đồ use case quản lý blog.....	78
Hình 7-6: Sơ đồ tuần tự chức năng Thêm Blog	80
Hình 7-7: Sơ đồ tuần tự chức năng Xóa Blog.....	80
Hình 7-8: Sơ đồ tuần tự chức năng Sửa Blog	81
Hình 7-9: Sơ đồ use case xử lý phiếu liên hệ	82
Hình 7-10: Sơ đồ tuần tự chức năng Gửi phản hồi	83
Hình 7-11: Sơ đồ use case xử lý đánh giá	84
Hình 7-12: Sơ đồ tuần tự chức năng Duyệt đánh giá.....	85
Hình 7-13: Sơ đồ tuần tự chức năng Xóa đánh giá	86
Hình 7-14: Sơ đồ use case chức năng quản lý người dùng	87
Hình 7-15: Sơ đồ tuần tự chức năng Khóa tài khoản	88
Hình 7-16: Sơ đồ tuần tự chức năng Mở khóa tài khoản	89
Hình 7-17: Sơ đồ use case chức năng quản lý quản trị	90
Hình 7-18: Sơ đồ tuần tự chức năng Thêm quản trị.....	91
Hình 7-19: Sơ đồ tuần tự chức năng Xóa quản trị	92
Hình 8-1: Giao diện đăng nhập	93

Hình 8-2: Giao diện trang chủ.....	94
Hình 8-3: Giao diện trang Blog.....	94
Hình 8-4: Giao diện trang Giới thiệu	95
Hình 8-5: Giao diện Trang Liên hệ	95
Hình 8-6: Giao diện Trang Đăng ký	96
Hình 8-7: Giao diện Trang cá nhân	97
Hình 8-8: Giao diện Bạn bè.....	97
Hình 8-9: Giao diện chat	98
Hình 8-10: Giao diện Status	99
Hình 8-11: Giao diện Đánh giá	99
Hình 8-12: Giao diện sửa thông tin cá nhân.....	100
Hình 8-13 : Giao diện trang chủ Admin.....	100
Hình 8-14 : Giao diện Quản lý thể loại blog	101
Hình 8-15 : Giao diện Thêm loại blog	101
Hình 8-16 : Giao diện Liên hệ.....	102
Hình 8-17 : Giao diện Quản lý User.....	102
Hình 8-18 : Giao diện Quản lý đánh giá	103

MỤC LỤC CÁC BẢNG

Bảng 5-1: Bảng mô tả thực thể User	38
Bảng 5-2: Bảng mô tả thực thể Albums	38
Bảng 5-3: Bảng mô tả thực thể Images	39
Bảng 5-4: Bảng mô tả thực thể Friends	39
Bảng 5-5: Bảng mô tả thực thể Messages	39
Bảng 5-6: Bảng mô tả thực thể Post	39
Bảng 5-7: Bảng mô tả thực thể Review	40
Bảng 5-8: Bảng mô tả thực thể Likes	40
Bảng 5-9: Bảng mô tả thực thể Comment	40
Bảng 5-10: Bảng mô tả thực thể Blog	40
Bảng 5-11: Bảng mô tả thực thể Blog_Album	41
Bảng 5-12: Bảng mô tả thực thể Blog_Category	41
Bảng 5-13: Bảng mô tả dữ liệu User	43
Bảng 5-14: Bảng mô tả dữ liệu Albums	43
Bảng 5-15: Bảng mô tả dữ liệu Images	44
Bảng 5-16: Bảng mô tả dữ liệu Friends	44
Bảng 5-17: Bảng mô tả dữ liệu Messages	44
Bảng 5-18: Bảng mô tả dữ liệu Post	44
Bảng 5-19: Bảng mô tả dữ liệu Review	45
Bảng 5-20: Bảng mô tả dữ liệu Likes	45
Bảng 5-21: Bảng mô tả dữ liệu Comment	45
Bảng 5-22: Bảng mô tả dữ liệu Blog	46
Bảng 5-23: Bảng mô tả dữ liệu Blog_Album	46
Bảng 5-24: Bảng mô tả dữ liệu Blog_Category	46
Bảng 6-1: Bảng mô tả usecase quản lý đăng nhập	49
Bảng 6-2: Bảng mô tả usecase tìm kiếm bạn bè	52
Bảng 6-3: Bảng mô tả usecase quản lý trang cá nhân	55
Bảng 6-4: Bảng mô tả usecase quản lý bạn bè	58

Bảng 6-5: Bảng mô tả usecase Chat.....	61
Bảng 6-6: Bảng mô tả usecase Status.....	63
Bảng 6-7: Bảng mô tả usecase Review	67
Bảng 6-8: Bảng mô tả usecase Like	69
Bảng 6-9: Bảng mô tả usecase Bình luận.....	72
Bảng 7-1: Bảng mô tả usecase quản lý thể loại blog.....	75
Bảng 7-2: Bảng mô tả usecase quản lý Blog.....	79
Bảng 7-3: Bảng mô tả usecase xử lý phiếu liên hệ	83
Bảng 7-4: Bảng mô tả use case xử lý đánh giá.....	85
Bảng 7-5: Bảng mô tả use case quản lý người dùng	88
Bảng 7-6: Bảng mô tả usecase quản trị	91

Chương 1. GIỚI THIỆU

1.1 ĐẶT VẤN ĐỀ

Thế kỷ 21 được coi là kỷ nguyên phát triển của công nghệ thông tin. Chúng ta đã và đang được chứng kiến sự phát triển với tốc độ phi thường của công nghệ thông tin. Năng lực xử lý và lưu trữ của từng cá nhân, từng tổ chức ngày càng gia tăng nhờ việc sử dụng máy tính và Internet. Công nghệ thông tin đang được áp dụng trong mọi lĩnh vực hoạt động của con người từ đơn giản cho tới phức tạp nhất.

Việc tìm kiếm bạn đời online trên mạng ngày nay rất phổ biến với hàng trăm, hàng nghìn ứng dụng, website hẹn hò. Hoà chung với xu hướng của thế giới cùng với việc công nghệ ngày càng phát triển thì việc giao lưu, kết bạn hẹn hò trên mạng ở Việt Nam cũng đã phổ biến hơn và bạn trẻ độc thân đã có thêm nhiều cơ hội trong việc tìm nửa ấy cho mình.

Với mong muốn có thể tìm thấy người ấy của mình, nhiều bạn trẻ đã tìm đến 1 cách khác để có thể kết bạn làm quen và hẹn hò thông qua mạng internet, các trang mạng xã hội.

Hiện nay mọi người đều quen dần với việc sử dụng các phần mềm ứng dụng, website để làm việc, giải trí, lưu trữ thông tin. Với việc giao lưu, kết bạn dựa trên hình thức website sẽ giúp cho người sử dụng thao tác dễ dàng hơn, quản lý chặt chẽ hơn và bảo mật được thông tin.

Nhận thức được việc đó nên nhóm em đã quyết định chọn đề tài “**Xây dựng ứng dụng web hẹn hò trên Laravel**” để thử sức bản thân vào việc xây dựng một website giúp cho việc kết nối mọi người trở nên đơn giản hơn.

1.2 MỤC TIÊU CỦA ĐỀ TÀI

Xây dựng một website tương đối hoàn chỉnh phục vụ có hiệu quả cho người dùng. Đảm bảo mọi chức năng hoạt động tốt, mang lại hiệu quả cao, phù hợp với nhu cầu người dùng.

1.3 ĐỐI TƯỢNG NGHIÊN CỨU

Ngôn ngữ lập trình PHP, framework Laravel, framework Bootstrap, mô hình MVC, hệ cơ sở dữ liệu MySQL.

Cách thức thiết kế một dự án website và triển khai hệ thống trong thực tế.

1.4 PHẠM VI NGHIÊN CỨU

Đề tài nằm trong phạm vi môn học Luận văn tốt nghiệp.

1.5 MỤC ĐÍCH NGHIÊN CỨU

Nắm được quy trình thực hiện một dự án thiết kế website.

Nắm được các kiến thức cơ bản về framework Laravel, framework Bootstrap và cách thao tác thực hiện của nó trong việc xây dựng ứng dụng web.

Ứng dụng kết quả nghiên cứu xây dựng website để giới thiệu hình ảnh, các hoạt động... của trường Đại học Công nghệ Sài Gòn.

1.6 NHIỆM VỤ NGHIÊN CỨU

Tìm hiểu quy trình thiết kế, các chức năng của một website hện hò.

Tìm hiểu và nghiên cứu ngôn ngữ lập trình PHP, phần mềm nguồn mở.

Tìm hiểu Laravel framework.

Phân tích thiết kế hệ thống website theo hướng đối tượng.

Xây dựng cơ sở dữ liệu.

Xây dựng ứng dụng hện hò trên web bằng Laravel.

1.7 PHƯƠNG PHÁP NGHIÊN CỨU

Nghiên cứu qua Internet: Một số website hện hò khác.

Phương pháp làm việc: Lập ra kế hoạch thực hiện đề tài.

Thiết kế các chức năng của ứng dụng và tìm ra hướng giải quyết tối ưu nhất.

Tìm và đưa ra biện pháp khắc phục khi gặp các sự cố, khó khăn trong khi thực hiện đề tài.

1.8 Ý NGHĨA LUẬN VÀ THỰC TIỄN CỦA ĐỀ TÀI

Ý nghĩa lý luận:

Rèn được kỹ năng phân tích và thiết kế dự án.

Học hỏi nghiên cứu về công nghệ lập trình web Laravel và cơ sở dữ liệu MySQL phục vụ cho việc thiết kế website hện hò.

Ý nghĩa thực tiễn

Sản phẩm của đồ án thu được là một website hện hò với nhiều chức năng cho các đối tượng có nhu cầu tìm kiếm bạn đời.

Chương 2. MÔ TẢ NGHIỆP VỤ

2.1 TỔNG QUAN

Website hẹn hò gồm 2 bộ phận chính với các chức năng khác nhau, song hoạt động dựa vào nhau.

1. Bộ phận hẹn hò, kết bạn giao lưu	2. Bộ phận quản trị hệ thống
-------------------------------------	------------------------------

2.2 QUY TRÌNH THAM GIA HẸN HÒ

Người dùng trước khi sử dụng hệ thống phải đăng ký tài khoản bằng thông tin thật và sẽ được xác nhận mức độ chính xác. Quy trình đăng ký rất đơn giản, quan trọng nhất là người dùng cần phải xác thực email đăng ký thì hệ thống mới chấp thuận tài khoản của người dùng.

Hệ thống có thể hủy tài khoản ngay lập tức mà không cần phải thông báo trước nếu tài khoản bị nghi ngờ rằng giả mạo. Ngoài ra việc khóa tài khoản dựa trên một số hành vi như sau:

- Ảnh không phù hợp: khiêu dâm, bạo lực, kinh dị, động vật, truyện tranh, game, hoạt hình, ảnh của người nổi tiếng,...
- Mục đích sử dụng dịch vụ không đúng với trang web đã đề ra.
- Tài khoản giả mạo, mạo danh người khác, không chính chủ.
- Hành vi phá hoại cộng đồng: đe dọa, spam, quấy rối người dùng khác, nội dung chính trị, tôn giáo, sắc tộc và đòi truy,...
- Sử dụng ngôn từ không phù hợp trong giao tiếp.
- Mục đích xấu: kinh doanh, lợi dụng để trục lợi, phá hoại,...
- Báo cáo vi phạm của người dùng khác. Thông tin sẽ được xem xét và kiểm chứng và có biện pháp xử lý kịp thời.

Khi đã hoàn tất các bước đăng ký, người dùng sẽ có được tài khoản hợp lệ để bắt đầu việc kết bạn – hẹn hò.

Việc đăng nhập có thể diễn ra trong 3 cách sau:

- Đăng nhập bằng tài khoản email đã được xác thực.
- Đăng nhập bằng tài khoản Facebook.
- Đăng nhập bằng tài khoản Google.

Sau khi đăng nhập, người dùng có thể cập nhật lại thông tin cá nhân của mình để hiển thị lên trang cá nhân.

Người dùng có thể tìm kiếm bạn bè theo một hoặc nhiều tiêu chí, có thể theo tên, theo sở thích, theo nghề nghiệp,... hệ thống sẽ ghi nhận và sẽ tiến hành tìm kiếm trong hệ thống. Hệ thống sẽ trả về một danh sách người dùng phù hợp với các từ khóa mà người dùng tìm kiếm hoặc trả về một danh sách rỗng nếu không có đối tượng phù hợp với từ khóa mà người dùng đưa ra.

Khi người dùng ưng ý một trong các kết quả trả về, chọn Kết bạn và chờ đối phương chấp nhận lời mời kết bạn sau đó cả hai mới có thể bắt đầu trò chuyện với nhau. Việc trò chuyện với nhau chỉ có hiệu lực khi cả hai cùng là Bạn bè. Ngoài ra hệ thống còn hiển thị trạng thái online/offline của bạn bè cho người dùng thuận tiện trò chuyện. Sau một khoảng thời gian trò chuyện, nếu cả hai bên đều muốn tìm hiểu thêm thì có thể chọn Ghép Đôi để trở thành một cặp. Hệ thống sẽ ghi nhận kết quả và sẽ thay đổi trạng thái của cả hai đối tượng. Hệ thống ghi nhận trường hợp này là ghép đôi thành công. Hệ thống cần phải lưu lịch sử đoạn chat nhằm người dùng yêu cầu.

Trường hợp đối tượng đang trong quá trình Ghép Đôi có vấn đề trong chuyện tình cảm và muốn dừng có thể nhấn chọn Chia Tay. Hệ thống sẽ gửi đề nghị chia tay cho cả hai cùng xác nhận. Hệ thống ghi nhận trường hợp này là ghép đôi thất bại.

Việc thu thập thông tin cá nhân được thực hiện trên cơ sở tự khai báo để đăng ký thành viên, tùy từng thời điểm, thông tin thu thập sẽ bao gồm nhưng không giới hạn ở:

- Thông tin cá nhân như: tên, tuổi, giới tính, ngày sinh, email, số điện thoại, quê quán, thành phố hiện đang sinh sống...

2.3 QUY TRÌNH QUẢN TRỊ HỆ THỐNG

2.3.1 Quản trị viên

Quản trị viên tham gia việc quản trị hệ thống với một số quyền hạn nhất định.

Quản trị viên là những người quản trị hỗ trợ Quản trị viên cao cấp quản lý thông tin, nội dung đăng tải trên website.

Quản trị viên cần đăng nhập để sử dụng hệ thống. Tài khoản của quản trị viên sẽ do quản trị viên cấp cao cung cấp. Trong trường hợp quên mật khẩu quản trị viên có thể lấy lại mật khẩu bằng email đăng nhập của mình.

Quản trị viên có thể tạo mới một blog để đăng lên trang chủ với các thông tin cơ bản như tiêu đề, nội dung, hình ảnh kèm theo, tác giả, v.v.. và tính năng ẩn/hiện tin.

Ngoài ra còn có chức năng duyệt đánh giá và phản hồi liên hệ từ khách hàng. Hệ thống phải hiển thị toàn bộ các đánh giá và phản hồi chưa được duyệt, từ đó quản trị viên có thể duyệt hoặc xóa các đánh giá và phản hồi trên. Sau khi xử lý xong thì phải cập nhật lại trạng thái của danh sách đánh giá và liên hệ.

Hệ thống hiển thị đầy đủ thông tin của quản trị viên, cung cấp cho quản trị viên chức năng đổi mật khẩu tài khoản của mình.

2.3.2 Quản trị viên cấp cao

Quản trị viên cấp cao là quản trị viên có quyền hạn lớn nhất trong hệ thống, sử dụng toàn quyền trong hệ thống.

Tuy là quản trị viên cấp cao nhưng vẫn cần phải đăng nhập để thực hiện việc quản trị của mình.

Quản trị viên cấp cao là người cung cấp tài khoản quản trị viên và quản lý các thông tin của viên cao cấp.

Chương 3. PHÂN TÍCH CHỨC NĂNG

3.1 MODULE USER

Đăng ký, đăng nhập, quản lý thông tin: Đăng tải thông tin cho phép thành viên đăng ký làm thành viên, quản lý thông tin yêu cầu của mình. Thành viên có trang quản lý thông tin của mình, có thể chỉnh sửa thông tin như: các thông tin cá nhân, hình ảnh, giới thiệu bản thân,...

Đăng xuất: Kết thúc phiên làm việc của người dùng, cho người dùng trở về trang chủ.

Xác nhận đăng ký qua email. Người đăng ký phải xác thực việc đăng ký qua email. Điều này giúp tránh các đăng ký ảo, các đăng ký spam hệ thống.

Tính năng khôi phục mật khẩu qua email: tránh việc mất thông tin tài khoản, mật khẩu tài khoản.

Trang thông tin cá nhân: Hiện thị đầy đủ các thông tin cá nhân của thành viên, có thể bổ sung, cập nhập một số thông tin đầy đủ.

Quản lý hình ảnh giới thiệu, album ảnh

- Tính năng đặt làm avatar: cho phép chọn ảnh làm hình đại diện.
- Tính năng sắp xếp album: có tính năng sắp xếp album theo ngày tạo.
- Tính năng tạo album ảnh: cho phép người dùng tạo album gồm các hình ảnh được đăng tải lên.
- Tính năng xóa album: cho phép xóa album mà người dùng không dùng nữa.
- Tính năng hiển thị hình ảnh theo album: hiển thị các hình ảnh trong album của người dùng.

Quản lý danh sách bạn bè

- Gửi yêu cầu kết bạn tới người khác.
- Có danh sách những người đã gửi yêu cầu kết bạn tới bạn và chờ chấp nhận.
- Danh sách bạn bè: Cho phép xem nhanh các thông tin của bạn bè.

- Danh sách đen – chặn. Những người thuộc danh sách đen này không gửi được tin nhắn, comment đến bạn.

Status: Cho phép thành viên cập nhật, đăng tải các tâm trạng, tâm sự của mình lên trang cá nhân.

Hệ thống tin nhắn: hệ thống cung cấp tính năng để các thành viên trao đổi tin nhắn với nhau.

Tìm bạn theo các tiêu chí: Giới tính: nam, nữ. Năm sinh (cho phép lựa chọn năm sinh), Chiều cao, khu vực như Hà Nội, Đà Nẵng ...

Hiển thị thông tin về bạn khi tìm kiếm theo tiêu chí: tên, ảnh đại diện. Click vào ảnh, tên để xem chi tiết thông tin.

Đánh giá: Cho phép thành viên đánh giá ứng dụng khi sử dụng. Mỗi thành viên chỉ được đánh giá 01 lần. Nội dung đánh giá sẽ được gửi về trang quản trị để kiểm duyệt và sẽ được đăng tải lên trang chủ khi được duyệt.

3.2 MODULE ADMIN

Quản lý thể loại blog

- Thêm/xóa/sửa các thể loại blog.
- Hiển thị danh sách các thể loại blog đang có để dàng quản lý.
- Tìm kiếm thể loại dễ dàng.
- Phân trang cho thể loại blog.
- Sắp xếp tăng/giảm dần theo các cột header.
- Xuất các file CSV/Excel/PDF để dàng lưu trữ.

Quản lý blog

- Thêm/xóa/sửa các blog.
- Hiển thị danh sách các blog đang có để dàng quản lý.
- Tìm kiếm blog dễ dàng.
- Phân trang cho blog.
- Sắp xếp tăng/giảm dần theo các cột header.

- Xuất các file CSV/Excel/PDF dễ dàng lưu trữ.

Quản lý liên hệ

- Hiển thị danh sách các liên hệ được gửi tới và đang chờ được xử lý.
- Tìm kiếm liên hệ dễ dàng.
- Phân trang cho liên hệ.
- Sắp xếp tăng/giảm dần theo các cột header.
- Xuất các file CSV/Excel/PDF dễ dàng lưu trữ.

Quản lý đánh giá

- Hiển thị danh sách các đánh giá mà người dùng gửi tới.
- Duyệt hoặc xóa đánh giá.
- Tìm kiếm đánh giá dễ dàng.
- Phân trang cho đánh giá.
- Sắp xếp tăng/giảm dần theo các cột header.
- Xuất các file CSV/Excel/PDF dễ dàng lưu trữ.

Quản lý User

- Hiển thị danh sách các người dùng thuận lợi cho việc quản lý.
- Khóa/Mở khóa tài khoản người dùng.
- Tìm kiếm người dùng dễ dàng.
- Phân trang cho người dùng.
- Sắp xếp tăng/giảm dần theo các cột header.
- Xuất các file CSV/Excel/PDF dễ dàng lưu trữ.

Quản lý quản trị

- Thêm các quản trị viên khác với quyền hạn ít hơn.
- Hiển thị danh sách các quản trị viên dưới quyền.
- Tìm kiếm quản trị viên dễ dàng.
- Phân trang cho quản trị viên.
- Sắp xếp tăng/giảm dần theo các cột header.

- Xuất các file CSV/Excel/PDF dễ dàng lưu trữ.

Đổi mật khẩu

Các quản trị con đổi mật khẩu ngay sau khi quản trị viên tạo tài khoản để quản trị.

Đăng nhập

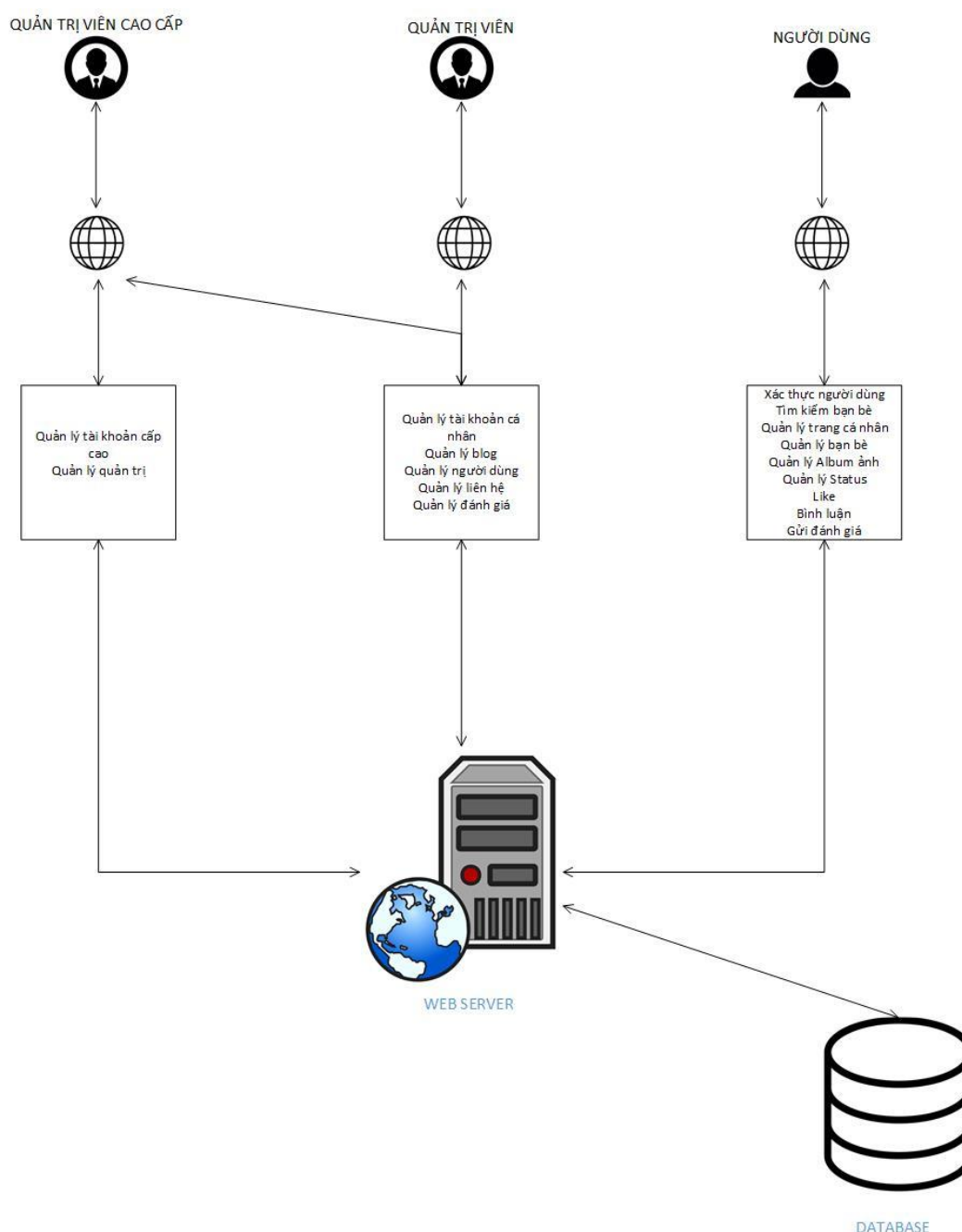
Các quản trị viên đăng nhập bằng thông tin tài khoản đã được cấp để vào trang quản trị.

Đăng xuất

Kết thúc phiên làm việc của quản trị viên, trở về trang chủ.

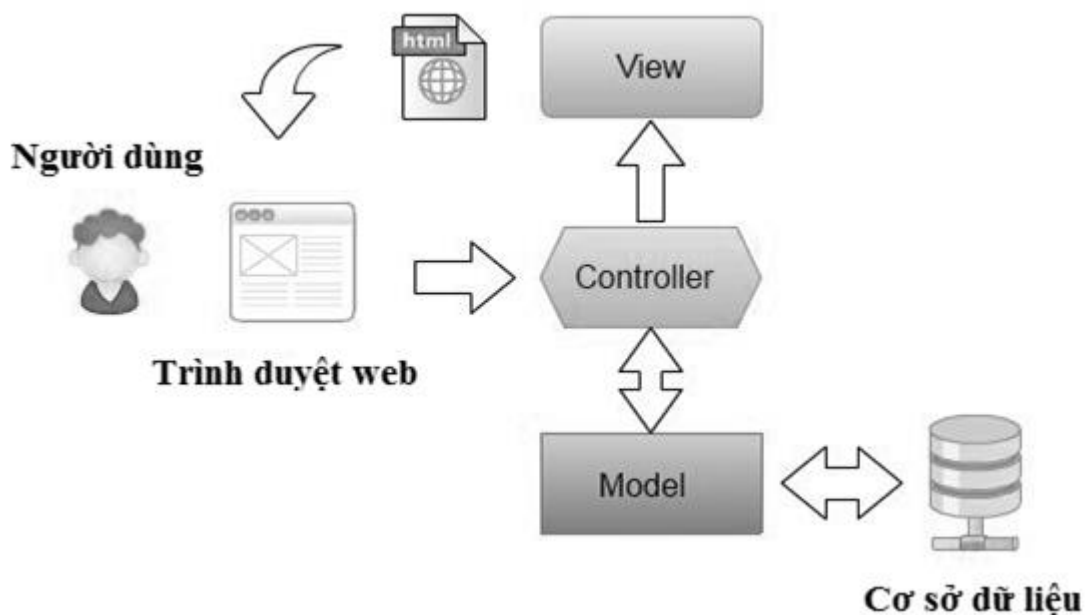
Chương 4. GIẢI PHÁP ĐỀ XUẤT

4.1 KIẾN TRÚC TỔNG THỂ



Hình 4-1: Kiến trúc tổng thể của hệ thống

4.2 MÔ HÌNH XÂY DỰNG



Hình 4-2: Mô hình xây dựng của hệ thống

4.2.1 Mô tả kiến trúc MVC

MVC là viết tắt của Model, View và Controller. MVC tách ứng dụng thành ba thành phần: Model, View và Controller.

- **Models:** Các đối tượng Models là một phần của ứng dụng, các đối tượng này thiết lập logic của phần dữ liệu của ứng dụng. Thông thường, các đối tượng model lấy và lưu trạng thái của model trong CSDL.
- **Views:** Views là các thành phần dùng để hiển thị giao diện người dùng (UI). Thông thường, view được tạo dựa vào thông tin dữ liệu model. View là giao diện người dùng.
- **Controllers:** Controller là các thành phần dùng để quản lý tương tác người dùng, làm việc với model và chọn view để hiển thị giao diện người dùng. Trong một ứng dụng MVC, view chỉ được dùng để hiển thị thông tin, controller chịu trách nhiệm

quản lý và đáp trả nội dung người dùng nhập và tương tác với người dùng. Controller là cầu nối giữa người sử dụng và ứng dụng.

4.2.2 Sự tương tác giữa các thành phần trong mô hình MVC

Controller tương tác với qua lại với View.

Controller tương tác qua lại với Model.

Model và View không có sự tương tác với nhau mà nó tương tác với nhau thông qua Controller.

4.2.3 Ưu – Nhược điểm của mô hình MVC

- **Ưu điểm:**
 - Các dự án có thể áp dụng ngay mô hình MVC mà không phụ thuộc môi trường, nền tảng xây dựng hay ngôn ngữ lập trình được sử dụng.
 - Có khả năng vận chuyển, phân chia các class/ function vào các thành phần riêng biệt Controller – Model – View, giúp cho việc xây dựng – phát triển – quản lý – vận hành và bảo trì một dự án dễ dàng hơn cũng như kiểm soát được các luồng xử lý và tạo ra các thành phần xử lý nghiệp vụ chuyên biệt hóa.
 - Tạo mô hình chuẩn cho dự án, khi người có chuyên môn ngoài dự án tiếp cận với dự án dễ dàng hơn.
 - Đây là mô hình đơn giản, xử lý những nghiệp vụ đơn giản, và dễ dàng triển khai với các dự án nhỏ.
- **Nhược điểm:**
 - Đối với các dự án có tính phức tạp cao, mô hình MVC trở nên không khả dụng.
 - Mô hình MVC cũng có nhiều bất cập, cụ thể là phân chia công việc không đồng đều. Phần Controller và Model thực hiện quá nhiều tác vụ, đặc biệt là

Model. Trong khi View chỉ chứa các mẫu hiển thị, thì Model xử lý mọi user request do Controller gửi đến và lo liệu cả UI logic cho lớp View.

4.3 GIẢI PHÁP CÔNG NGHỆ

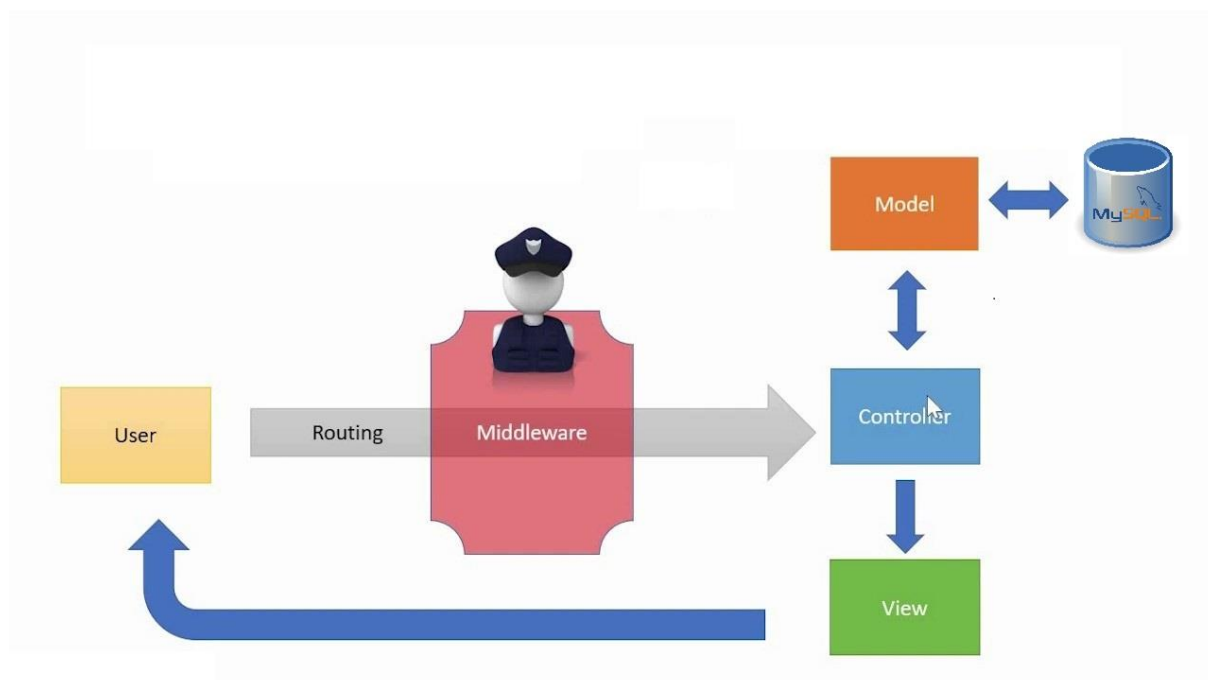
4.3.1 Công nghệ xử lý phía Front-end: Framework Bootstrap

Bootstrap là một trong những CSS Framework phổ biến nhất hiện nay, do Twitter phát triển. CSS Framework này có ưu điểm là hỗ trợ khả năng responsive (tức là trang web sẽ tự động co giãn theo kích thước của cửa sổ trình duyệt), tương thích tốt với thiết bị cỡ nhỏ, với sự phổ biến của smartphone, tablet hiện nay, đây là một yếu tố vô cùng quan trọng giúp các nhà thiết kế, phát triển ứng dụng web không cần phải design một bản riêng cho mobile hay PC, mà chỉ cần thiết kế một lần duy nhất cho mọi thiết bị. Việc này giúp tiết kiệm chi phí cho người chủ ứng dụng

Lý do chọn Bootstrap:

- Tiết kiệm thời gian, dễ sử dụng.
- Khả năng tùy chỉnh.
- Hệ thống grid responsive tốt nhất.
- Tính nhất quán cao.
- Khả năng tương thích với hầu như tất cả các trình duyệt hiện đại.
- Có sẵn tài liệu để học cùng với ví dụ và các bản demo.

4.3.2 Công nghệ xử lý phía Back-end: Framework Laravel



Hình 4-3: Laravel theo mô hình MVC

Laravel là một PHP Framework mã nguồn mở và miễn phí, được phát triển bởi Taylor Otwell và nhằm mục tiêu hỗ trợ phát triển các ứng dụng web theo cấu trúc model-view-controller (MVC). Những tính năng nổi bật của Laravel bao gồm cú pháp dễ hiểu rõ ràng, một hệ thống đóng gói Modular và quản lý gói phụ thuộc, nhiều cách khác nhau để truy cập vào các cơ sở dữ liệu quan hệ, nhiều tiện ích khác nhau hỗ trợ việc triển khai vào bảo trì ứng dụng.

- Có thể tạo liên kết đến các route được đặt tên. Vì vậy, khi tạo các liên kết, bạn chỉ cần sử dụng tên của bộ định tuyến, framework sẽ tự động chèn URL chính xác. Nếu bạn cần thay đổi các route trong tương lai, những thay đổi sẽ tự động được thực hiện ở mọi nơi. Cú pháp và cấu trúc của PHP tương đối dễ hiểu và áp dụng.
- Tài liệu của Laravel rất thân thiện với nhà phát triển. Tất cả các phiên bản của Laravel được phát hành cùng với các tài liệu phù hợp, bạn sẽ tìm thấy những giải thích chi tiết về coding style, methods và classes.

- Tích hợp với dịch vụ mail: Các ứng dụng web bắt buộc phải thông báo cho người dùng về các deal và offer mới. Đăng ký người dùng mới cũng rất quan trọng (nghĩa là thông báo cho người dùng mới khi anh ấy/ cô ấy đăng ký). Framework được trang bị API sạch trên thư viện SwiftMailer. Và có các driver cho SMTP, Mailgun, SparkPost, Mandrill, PHP's "mail" function, Amazon SES, và "sendmail" giúp bạn gửi thư qua các dịch vụ dựa trên đám mây hoặc dịch vụ local.
- Công cụ tích hợp cho dòng lệnh – Artisan: Các nhà phát triển thường thấy tẻ nhạt khi thực hiện các nhiệm vụ lập trình lặp đi lặp lại vì chúng rất tốn thời gian. Công cụ dòng lệnh có tên Artisan giúp họ tạo khung code và quản lý hệ thống cơ sở dữ liệu hiệu quả. Artisan giúp tạo các tệp MVC cơ bản và quản lý tài sản, bao gồm cả các cấu hình tương ứng của chúng.
- Eloquent ORM: Đây là một trong những tính năng tốt nhất của Laravel, cung cấp một triển khai ActiveRecord đơn giản và hiệu quả, làm việc với cơ sở dữ liệu. Các mô hình bạn tạo trong MVC sẽ có một bảng tương ứng trong cơ sở dữ liệu. ORM có các mối quan hệ, nếu bạn thao tác một bảng trong cơ sở dữ liệu, bạn cũng sẽ quản lý các dữ liệu liên quan. Các mối quan hệ đó là: one-to-one, one-to-many, many-to-many, has-many-through, polymorphic, và many-to-many polymorphic.

4.3.3 Hệ quản trị cơ sở dữ liệu: MySQL

MySQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng.

Tại sao sử dụng MySQL?

- MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng.
- Bảo vệ dữ liệu mạnh mẽ.
- MySQL là sự chọn lựa tốt nhất cho cả ứng dụng web vì MySQL là nơi cho các website trao đổi thường xuyên bởi nó có engine xử lý tốc độ cao, khả

năng chèn dữ liệu cực nhanh và hỗ trợ mạnh các chức năng chuyên dụng của web.

- Mã nguồn mở tự do và hỗ trợ 24/7.

4.3.4 Giao diện lập trình ứng dụng: API

API là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác. Nó là viết tắt của Application Programming Interface – giao diện lập trình ứng dụng. API cung cấp khả năng cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Và từ đó có thể trao đổi dữ liệu giữa các ứng dụng.

API hiện nay đều tuân thủ theo tiêu chuẩn REST và HTTP, tạo sự thân thiện để sử dụng với nhà phát triển. Giúp người dùng dễ dàng truy cập, dễ hiểu hơn. Web API hiện đại dùng cho các đối tượng cụ thể, chẳng hạn như mobile developer với document, version khác nhau.

Web API: là hệ thống API được sử dụng trong các hệ thống website. Hầu hết các website đều ứng dụng đến Web API cho phép bạn kết nối, lấy dữ liệu hoặc cập nhật cơ sở dữ liệu. Ví dụ: Bạn thiết kế chức năng login thông Google, Facebook, Twitter, Github... Điều này có nghĩa là bạn đang gọi đến API của. Hoặc như các ứng dụng di động đều lấy dữ liệu thông qua API.

Web API là một phương thức dùng để cho phép các ứng dụng khác nhau có thể giao tiếp, trao đổi dữ liệu qua lại. Dữ liệu được Web API trả lại thường ở dạng JSON hoặc XML thông qua giao thức HTTP hoặc HTTPS.

Những điểm nổi bật của Web API

Web API hỗ trợ restful đầy đủ các phương thức: Get/Post/put/delete dữ liệu. Nó giúp bạn xây dựng các HTTP service một cách rất đơn giản và nhanh chóng. Nó cũng có khả năng hỗ trợ đầy đủ các thành phần HTTP: URI, request/response headers, caching, versioning, content format.

Tự động hóa sản phẩm

Với web API, chúng ta sẽ tự động hóa quản lý công việc, cập nhật luồng công việc, giúp tăng năng suất và tạo hiệu quả công việc cao hơn.

Khả năng tích hợp linh động

API cho phép lấy nội dung từ bất kỳ website hoặc ứng dụng nào một cách dễ dàng nếu được cho phép, tăng trải nghiệm người dùng. API hoạt động như một chiếc cổng, cho phép các công ty chia sẻ thông tin được chọn nhưng vẫn tránh được những yêu cầu không mong muốn.

Cập nhật thông tin thời gian thực

API có chức năng thay đổi và cập nhật thay đổi theo thời gian thực. Với công nghệ này, dữ liệu sẽ được truyền đi tốt hơn, thông tin chính xác hơn, dịch vụ cung cấp linh hoạt hơn.

Có tiêu chuẩn chung để sử dụng

Bất kỳ người dùng, công ty nào sử dụng cũng có thể điều chỉnh nội dung, dịch vụ mà họ sử dụng.

Hỗ trợ đầy đủ các thành phần MVC như: routing, controller, action result, filter, model binder, IoC container, dependency injection, unit test.

Web API hoạt động như thế nào?

Đầu tiên là xây dựng URL API để bên thứ ba có thể gửi request dữ liệu đến máy chủ cung cấp nội dung, dịch vụ thông qua giao thức HTTP hoặc HTTPS.

Tại web server cung cấp nội dung, các ứng dụng nguồn sẽ thực hiện kiểm tra xác thực nếu có và tìm đến tài nguyên thích hợp để tạo nội dung trả về kết quả.

Server trả về kết quả theo định dạng JSON hoặc XML thông qua giao thức HTTP/HTTPS.

Tại nơi yêu cầu ban đầu là ứng dụng web hoặc ứng dụng di động, dữ liệu JSON/XML sẽ được parse để lấy data. Sau khi có được data thì thực hiện tiếp các hoạt động như lưu dữ liệu xuống Cơ sở dữ liệu, hiển thị dữ liệu...

Ưu và nhược điểm của Web API

Mỗi một ứng dụng bất kỳ đều có những ưu nhược điểm riêng, hỗ trợ tốt cho các ứng dụng. Vì vậy mà web API cũng không ngoại lệ:

1. Ưu điểm

Web API được sử dụng hầu hết trên các ứng dụng desktop, ứng dụng mobile và ứng dụng website.

Linh hoạt với các định dạng dữ liệu khi trả về client: JSON, XML hay định dạng khác.

Nhanh chóng xây dựng HTTP service: URI, request/response headers, caching, versioning, content formats và có thể host trong ứng dụng hoặc trên IIS.

Mã nguồn mở, hỗ trợ chức năng RESTful đầy đủ, sử dụng bởi bất kì client nào hỗ trợ XML, JSON.

Hỗ trợ đầy đủ các thành phần MVC như: routing, controller, action result, filter, model binder, IoC container, dependency injection, unit test.

Giao tiếp hai chiều được xác nhận trong các giao dịch, đảm bảo độ tin cậy cao.

2. Nhược điểm

Do web API còn khá mới nên chưa thể đánh giá nhiều về nhược điểm của mô hình này. Tuy nhiên, có hai nhược điểm dễ dàng nhận thấy:

Web API chưa hoàn toàn phải là RESTful service, mới chỉ hỗ trợ mặc định GET, POST

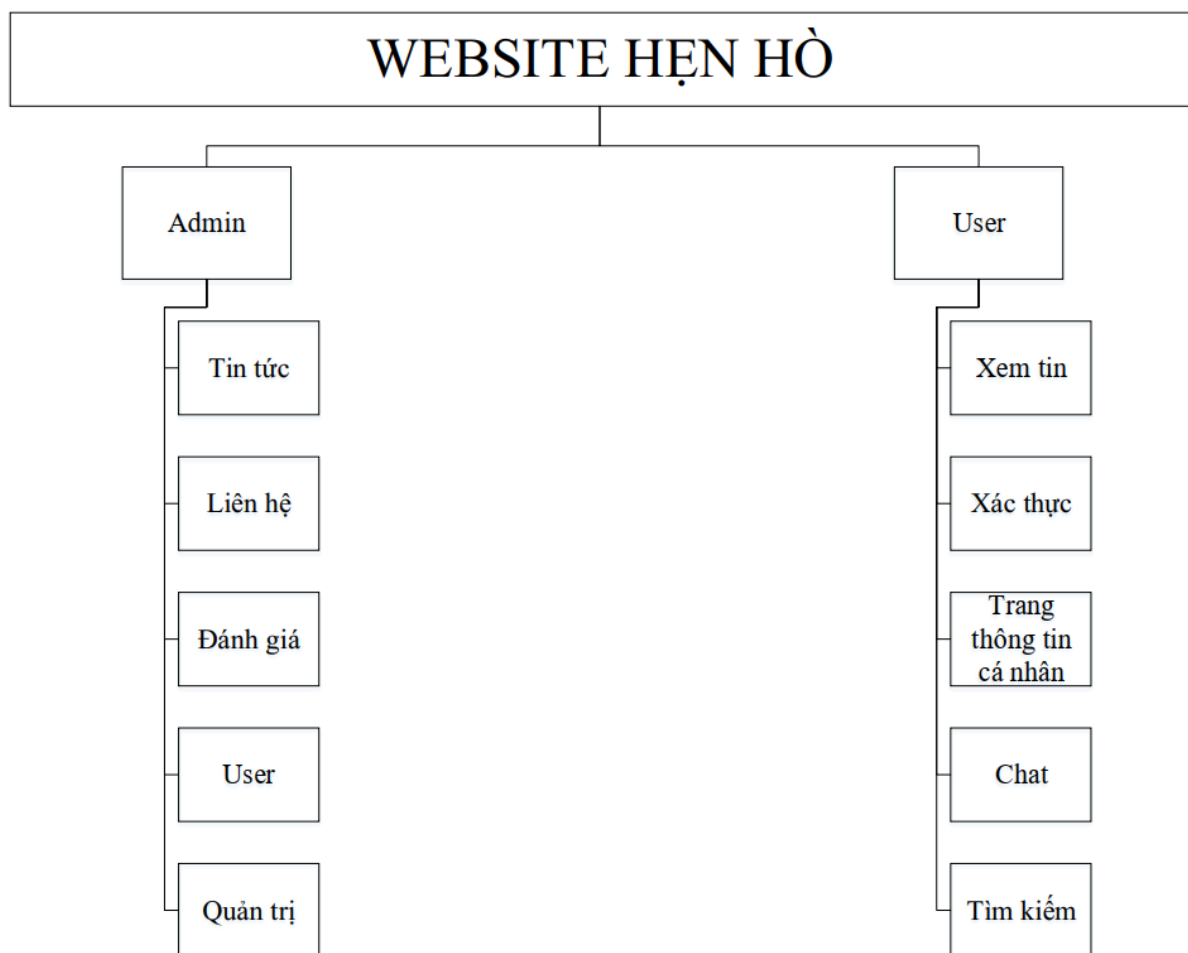
Để sử dụng hiệu quả cần có kiến thức chuyên sâu, có kinh nghiệm backend tốt

Tốn thời gian và chi phí cho việc phát triển, nâng cấp và vận hành

Có thể gặp vấn đề về bảo mật khi hệ thống bị tấn công nếu không giới hạn điều kiện kỹ.

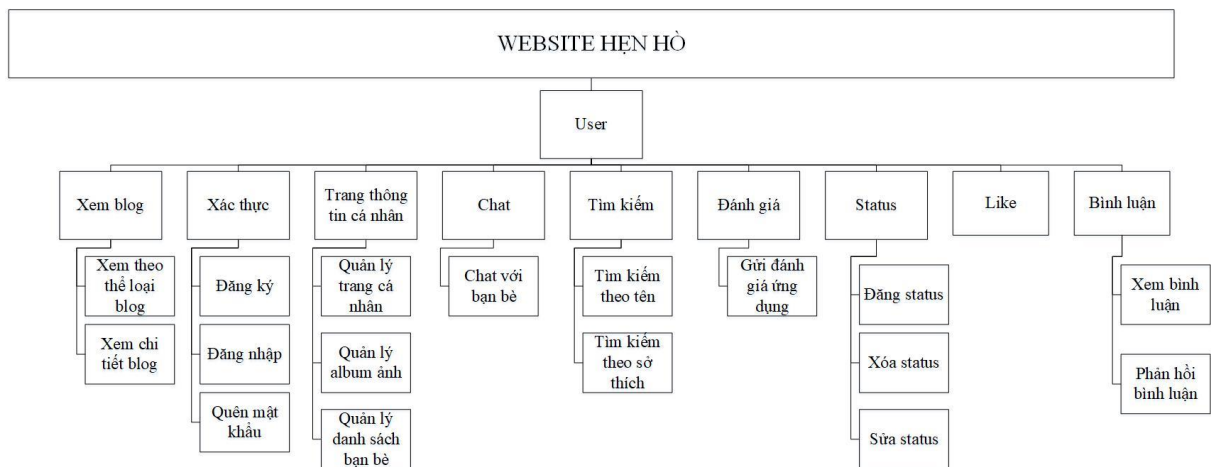
4.4 SƠ ĐỒ CHỨC NĂNG

4.4.1 Sơ đồ chức năng tổng quát



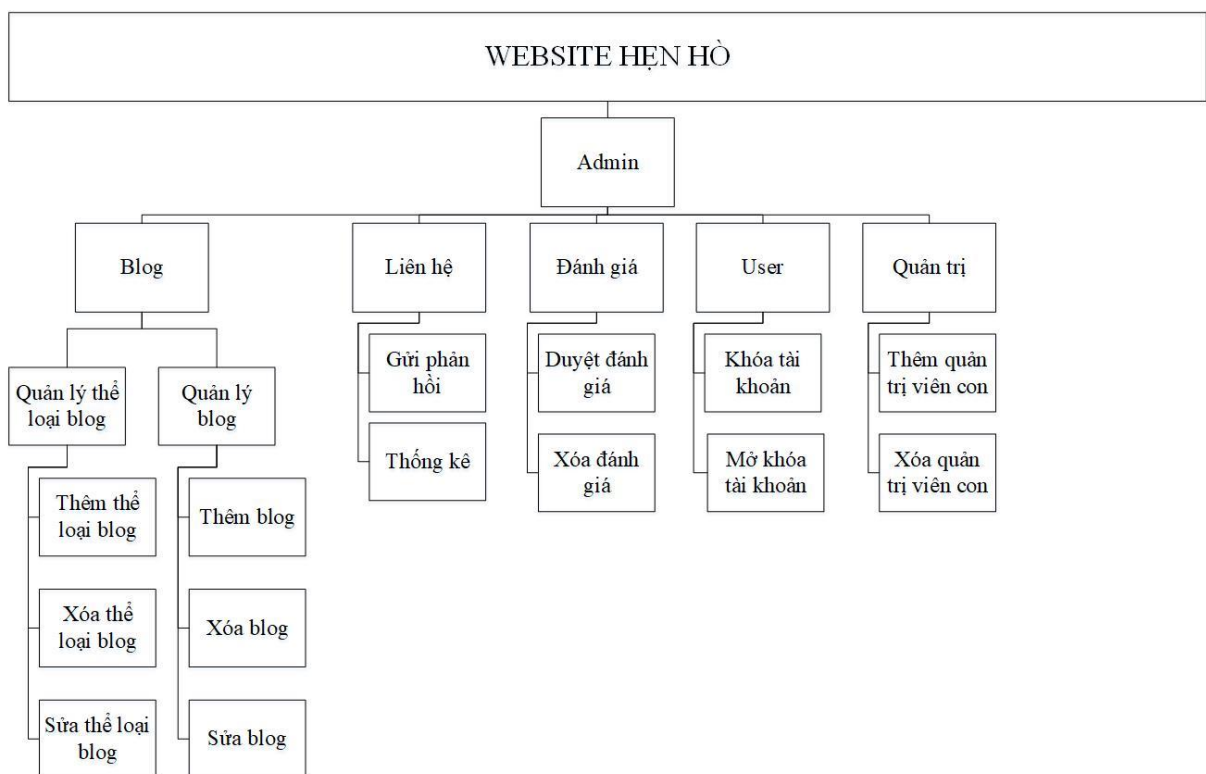
Hình 4-4: Sơ đồ chức năng tổng quát của hệ thống

4.4.2 Sơ đồ chức năng User



Hình 4-5: Sơ đồ chức năng của User

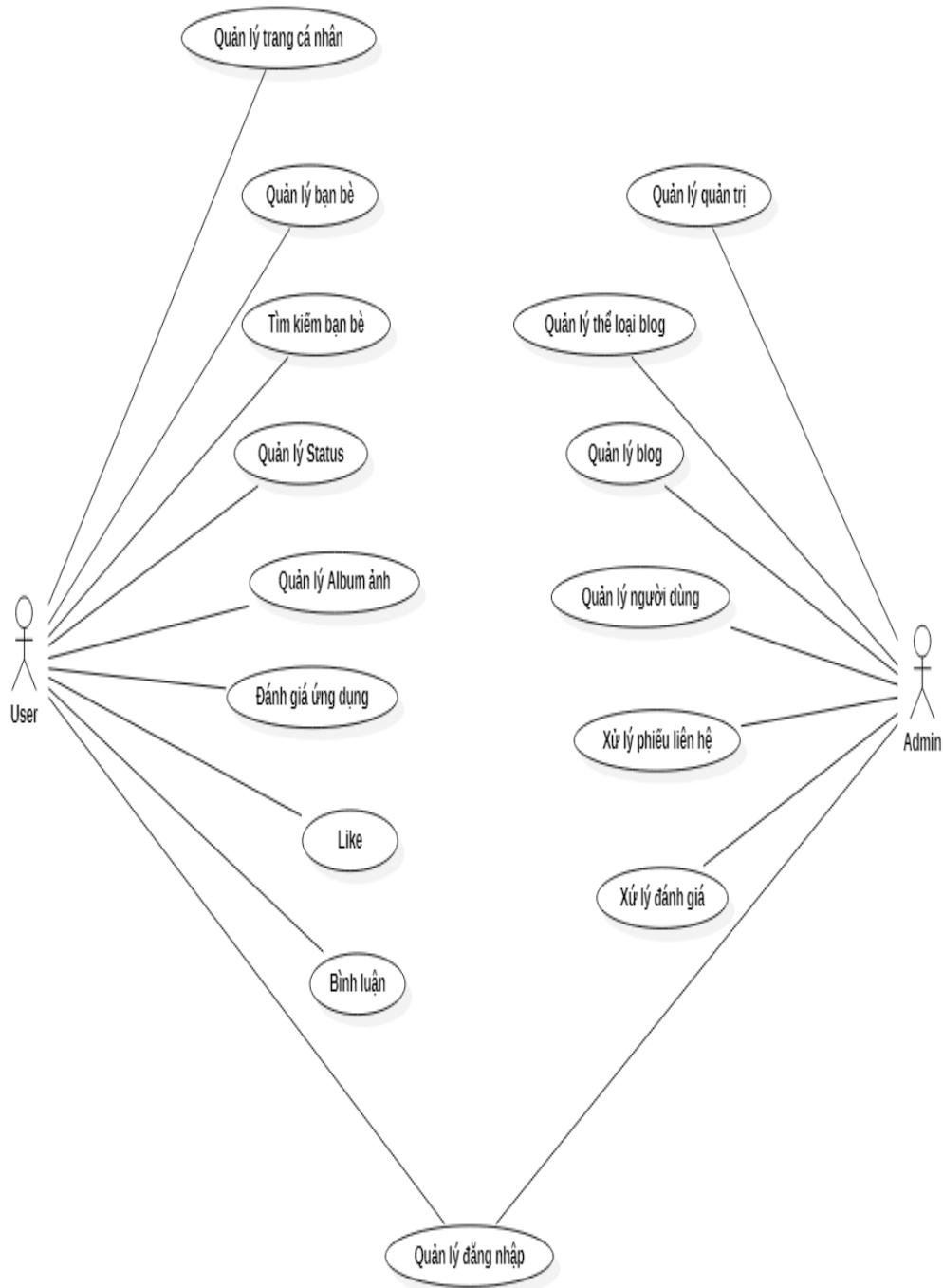
4.4.3 Sơ đồ chức năng Admin



Hình 4-6: Sơ đồ chức năng Admin

4.5 SƠ ĐỒ USE-CASE TỔNG QUÁT

Usecase tổng quát hệ thống

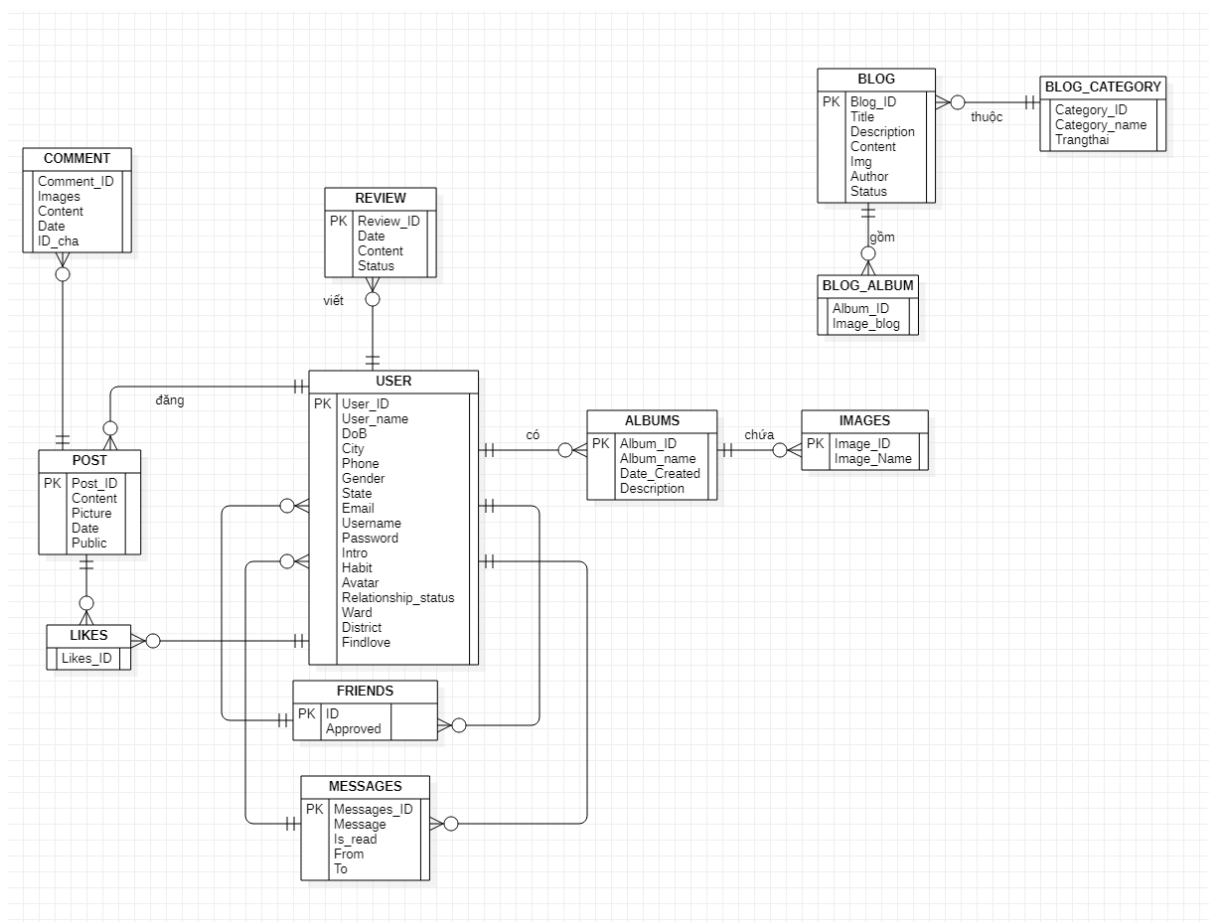


Hình 4-4: Use case tổng quát hệ thống

Chương 5. PHÂN TÍCH DỮ LIỆU

5.1 PHÂN TÍCH Ở MỨC QUAN NIỆM

5.1.1 Sơ đồ ER



Hình 5-1: Sơ đồ ER của hệ thống

5.1.2 Mô tả các loại thực thể

□ Loại thực thể USER

Mô tả: Loại thực thể USER gồm người dùng sử dụng hệ thống					
Thuộc tính	Kiểu	K	U	M	Diễn giải
User_ID	Varchar(10)	x	x	x	Mã người dùng
User_Name	Varchar(30)			x	Tên người dùng
DoB	date			x	Ngày sinh của người dùng
Gender	Tinyint(1)			x	Giới tính của người dùng
Religion	Varchar(30)			x	Tôn giáo của người dùng
Phone	Varchar(11)			x	Số điện thoại của người dùng
City	Varchar(191)			x	Thành phố sinh sống
District	Varchar(191)			x	Phường
Ward	Varchar(191)			x	Quận
Job	Varchar(50)			x	Nghề nghiệp người dùng
Habit	Text			x	Thói quen, sở thích người dùng
Intro	Text			x	Giới thiệu về bản thân người dùng
Findlove	Text			x	Nhu cầu tìm kiếm bạn bè của người dùng
Email	Varchar(191)			x	Email đăng nhập của người dùng
Password	Varchar(191)			x	Mật khẩu tài khoản người dùng
Status	Tinyint(1)			x	Trạng thái tài khoản người dùng
Role	Tinyint(1)			x	Phân loại người dùng
Img	Varchar(1000)			x	Ảnh đại diện người dùng

Bảng 5-1: Bảng mô tả thực thể User

□ Loại thực thể ALBUMS

Mô tả: Loại thực thể ALBUM gồm những album mà người dùng tạo ra					
Thuộc tính	Kiểu	K	U	M	Diễn giải
Album_ID	int(10)	x	x	x	Mã album
Album_Name	Varchar(191)			x	Tên album
Date_Created	date			x	Ngày tạo album
Description	Varchar(191)			x	Mô tả album

Bảng 5-2: Bảng mô tả thực thể Albums

❑ Loại thực thể IMAGES

Mô tả: Loại thực thể IMAGES gồm hình ảnh có trong album					
Thuộc tính	Kiểu	K	U	M	Diễn giải
Id_image	int(10)	x	x	x	Mã hình ảnh
Name_image	Varchar(191)			x	Tên hình ảnh

Bảng 5-3: Bảng mô tả thực thể Images

❑ Loại thực thể FRIENDS

Mô tả: Loại thực thể FRIENDS gồm những bạn bè của người dùng					
Thuộc tính	Kiểu	K	U	M	Diễn giải
id	int(10)	x	x	x	Mã bạn bè
approved	Tinyint(1)			x	Tình trạng bạn bè

Bảng 5-4: Bảng mô tả thực thể Friends

❑ Loại thực thể MESSAGES

Mô tả: Loại thực thể MESSAGES gồm tin nhắn của người dùng					
Thuộc tính	Kiểu	K	U	M	Diễn giải
id	bigint(20)	x	x	x	Mã tin nhắn
from	bigint(20)			x	Người gửi tin nhắn
to	bigint(20)			x	Người nhận tin nhắn
message	text			x	Tin nhắn
Is_read	Tinyint(4)			x	Kiểm tra tin nhắn đã được xem hay chưa

Bảng 5-5: Bảng mô tả thực thể Messages

❑ Loại thực thể POST

Mô tả: Loại thực thể POST gồm những bài đăng của người dùng					
Thuộc tính	Kiểu	K	U	M	Diễn giải
Post_ID	int(10)	x	x	x	Mã bài đăng
Content	text			x	Nội dung bài đăng
Picture	Varchar(191)			x	Ảnh của bài đăng
Date	date			x	Ngày đăng bài
Public	Tinyint(1)			x	Chế độ hiển thị của bài đăng

Bảng 5-6: Bảng mô tả thực thể Post

❑ Loại thực thể REVIEW

Mô tả: Loại thực thể REVIEW gồm những bài đánh giá về trang web của người dùng					
Thuộc tính	Kiểu	K	U	M	Diễn giải
Review_ID	int(10)	x	x	x	Mã bài đánh giá
Date	datetime			x	Ngày gửi bài đánh giá
Content	text			x	Nội dung bài đánh giá
Status	Tinyint(1)			x	Trạng thái hiển thị của bài đánh giá

Bảng 5-7: Bảng mô tả thực thể Review

❑ Loại thực thể LIKES

Mô tả: Loại thực thể LIKES gồm thông tin về lượt thích của bài đăng					
Thuộc tính	Kiểu	K	U	M	Diễn giải
Like_ID	int(10)	x	x	x	Mã like

Bảng 5-8: Bảng mô tả thực thể Likes

❑ Loại thực thể COMMENT

Mô tả: Loại thực thể COMMENT gồm những bình luận trong bài đăng người dùng					
Thuộc tính	Kiểu	K	U	M	Diễn giải
Comment_ID	int(10)	x	x	x	Mã bình luận
Images	Varchar(100)			x	Ảnh bình luận
Content	Varchar(191)			x	Nội dung bình luận
Date	Date			x	Ngày bình luận
ID_cha	Int(11)			x	ID bình luận trước để phản hồi

Bảng 5-9: Bảng mô tả thực thể Comment

❑ Loại thực thể BLOG

Mô tả: Loại thực thể BLOG gồm các bài blog hiển thị lên trang chủ					
Thuộc tính	Kiểu	K	U	M	Diễn giải
Blog_ID	bigint(20)	x	x	x	Mã bài blog
Title	Varchar(100)			x	Tiêu đề bài blog
Description	Text			x	Mô tả bài blog
Content	Text			x	Nội dung bài blog
Img	Varchar(191)			x	Hình ảnh bài blog
Datesubmitted	date			x	Ngày đăng bài blog
Author	Varchar(30)			x	Tác giả viết bài blog
Status	Tinyint(4)			x	Trạng thái của bài blog

Bảng 5-10: Bảng mô tả thực thể Blog

❑ Loại thực thể BLOG_ALBUM

Mô tả: Loại thực thể BLOG_ALBUM gồm album hình có trong bài blog					
Thuộc tính	Kiểu	K	U	M	Diễn giải
Album_ID	int(11)	x	x	x	Mã album ảnh của bài blog
Image_blog	Varchar(100)			x	Ảnh trong bài blog

Bảng 5-11: Bảng mô tả thực thể Blog_Album

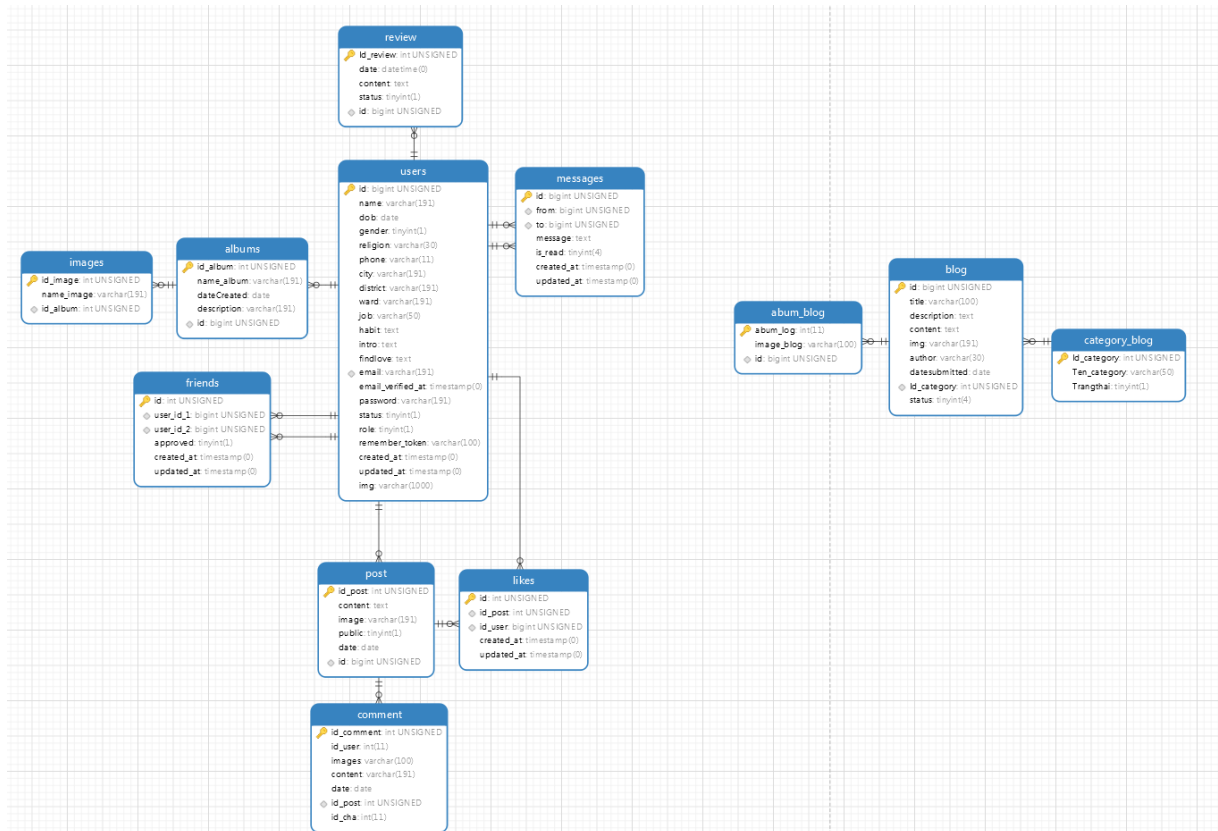
❑ Loại thực thể BLOG_CATEGORY

Mô tả: Loại thực thể BLOG_CATEGORY gồm các thể loại của bài blog					
Thuộc tính	Kiểu	K	U	M	Diễn giải
Category_ID	int(10)	x	x	x	Mã thể loại
Category_Name	Varchar(50)			x	Tên thể loại
Trangthai	Tinyint(1)			x	Trạng thái thể loại

Bảng 5-12: Bảng mô tả thực thể Blog_Category

5.2 THIẾT KẾ CƠ SỞ DỮ LIỆU

5.2.1 Sơ đồ logic dữ liệu (mô hình quan hệ)



Hình 5-2: Lược đồ quan hệ cơ sở dữ liệu

5.2.2 Mô tả các bảng dữ liệu

□ Bảng USER

Mô tả: Loại thực thể USER gồm người dùng sử dụng hệ thống					
Thuộc tính	Kiểu	K	U	M	Diễn giải
User_ID	Varchar(10)	x	x	x	Mã người dùng
User_Name	Varchar(30)			x	Tên người dùng
DoB	date			x	Ngày sinh của người dùng
Gender	Tinyint(1)			x	Giới tính của người dùng
Religion	Varchar(30)			x	Tôn giáo của người dùng
Phone	Varchar(11)			x	Số điện thoại của người dùng
City	Varchar(191)			x	Thành phố sinh sống
District	Varchar(191)			x	Phường
Ward	Varchar(191)			x	Quận
Job	Varchar(50)			x	Nghề nghiệp người dùng
Habit	Text			x	Thói quen, sở thích người dùng
Intro	Text			x	Giới thiệu về bản thân người dùng
Findlove	Text			x	Nhu cầu tìm kiếm bạn bè của người dùng
Email	Varchar(191)			x	Email đăng nhập của người dùng
Password	Varchar(191)			x	Mật khẩu tài khoản người dùng
Status	Tinyint(1)			x	Trạng thái tài khoản người dùng
Role	Tinyint(1)			x	Phân loại người dùng
Img	Varchar(1000)			x	Ảnh đại diện người dùng

Bảng 5-13: Bảng mô tả dữ liệu User

□ Bảng ALBUMS

Mô tả: Loại thực thể ALBUM gồm những album mà người dùng tạo ra					
Thuộc tính	Kiểu	K	U	M	Diễn giải
Id_album	int(10)	x	x	x	Mã album
Album_Name	Varchar(191)			x	Tên album
Date_Created	date			x	Ngày tạo album
Description	Varchar(191)			x	Mô tả album
Id	Bigint(20)			x	Mã người dùng

Bảng 5-14: Bảng mô tả dữ liệu Albums

❑ Bảng IMAGES

Mô tả: Loại thực thể IMAGES gồm hình ảnh có trong album					
Thuộc tính	Kiểu	K	U	M	Diễn giải
Id_image	int(10)	x	x	x	Mã hình ảnh
Name_image	Varchar(191)			x	Tên hình ảnh
Id_album	int(10)			x	Mã album chứa ảnh

Bảng 5-15: Bảng mô tả dữ liệu Images

❑ Bảng FRIENDS

Mô tả: Loại thực thể FRIENDS gồm những bạn bè của người dùng					
Thuộc tính	Kiểu	K	U	M	Diễn giải
id	int(10)	x	x	x	Mã bạn bè
approved	Tinyint(1)			x	Tình trạng bạn bè
Id_user_1	Bigint(20)			x	Mã người dùng 1
Id_user_2	Bigint(20)			x	Mã người dùng 2

Bảng 5-16: Bảng mô tả dữ liệu Friends

❑ Bảng MESSAGES

Mô tả: Loại thực thể MESSAGES gồm tin nhắn của người dùng					
Thuộc tính	Kiểu	K	U	M	Diễn giải
id	bigint(20)	x	x	x	Mã tin nhắn
from	bigint(20)			x	Người gửi tin nhắn
to	bigint(20)			x	Người nhận tin nhắn
message	text			x	Tin nhắn
Is_read	Tinyint(4)			x	Kiểm tra tin nhắn đã được xem hay chưa

Bảng 5-17: Bảng mô tả dữ liệu Messages

❑ Bảng POST

Mô tả: Loại thực thể POST gồm những bài đăng của người dùng					
Thuộc tính	Kiểu	K	U	M	Diễn giải
Post_ID	int(10)	x	x	x	Mã bài đăng
Content	Varchar(30)			x	Nội dung bài đăng
Picture	Varchar(30)			x	Ảnh của bài đăng
Date	date			x	Ngày đăng bài
Public	Tinyint(1)			x	Chế độ hiển thị của bài đăng
Id	Bigint(20)			x	Mã người dùng đăng bài

Bảng 5-18: Bảng mô tả dữ liệu Post

❑ Bảng REVIEW

Mô tả: Loại thực thể REVIEW gồm những bài đánh giá về trang web của người dùng					
Thuộc tính	Kiểu	K	U	M	Diễn giải
Review_ID	int(10)	x	x	x	Mã bài đánh giá
Date	date			x	Ngày gửi bài đánh giá
Content	text			x	Nội dung bài đánh giá
Status	Tinyint(1)			x	Trạng thái hiển thị của bài đánh giá
Id	Bigint(20)			x	Mã người dùng gửi bài đánh giá

Bảng 5-19: Bảng mô tả dữ liệu Review

❑ Bảng LIKES

Mô tả: Loại thực thể LIKES gồm thông tin về lượt thích của bài đăng					
Thuộc tính	Kiểu	K	U	M	Diễn giải
Like_ID	int(10)	x	x	x	Mã like
Post_ID	Int(10)			x	Mã bài post
User_ID	bigint(20)			X	Mã người dùng

Bảng 5-20: Bảng mô tả dữ liệu Likes

❑ Bảng COMMENT

Mô tả: Loại thực thể COMMENT gồm những bình luận trong bài đăng người dùng					
Thuộc tính	Kiểu	K	U	M	Diễn giải
Comment_ID	Varchar(10)	x	x	x	Mã bình luận
Images	int(10)			x	Ảnh bình luận
Content	int(10)			x	Nội dung bình luận
Date	Varchar(30)			x	Ngày bình luận
ID_cha	Varchar(30)			x	ID bình luận trước để phản hồi
Post_ID	Int(10)			x	Mã bài đăng chứa các bình luận

Bảng 5-21: Bảng mô tả dữ liệu Comment

❑ Bảng BLOG

Mô tả: Loại thực thể BLOG gồm các bài blog hiển thị lên trang chủ					
Thuộc tính	Kiểu	K	U	M	Diễn giải
Blog_ID	bigint(20)	x	x	x	Mã bài blog
Title	Varchar(100)			x	Tiêu đề bài blog
Description	Text			x	Mô tả bài blog
Content	Text			x	Nội dung bài blog
Img	Varchar(191)			x	Hình ảnh bài blog
Datesubmitted	date			x	Ngày đăng bài blog
Author	Varchar(30)			x	Tác giả viết bài blog
Status	Tinyint(4)			x	Trạng thái của bài blog
Category_ID	Int(10)			x	Mã thể loại bài blog

Bảng 5-22: Bảng mô tả dữ liệu Blog

❑ Bảng BLOG_ALBUM

Mô tả: Loại thực thể BLOG_ALBUM gồm album hình có trong bài blog					
Thuộc tính	Kiểu	K	U	M	Diễn giải
Album_ID	Int(11)	x	x	x	Mã album ảnh của bài blog
Image_blog	Varchar(100)			x	Ảnh trong bài blog
Blog_ID	Int(10)			x	Mã bài blog

Bảng 5-23: Bảng mô tả dữ liệu Blog_Album

❑ Bảng BLOG_CATEGORY

Mô tả: Loại thực thể BLOG_CATEGORY gồm các thể loại của bài blog					
Thuộc tính	Kiểu	K	U	M	Diễn giải
Category_ID	Varchar(10)	x	x	x	Mã thể loại
Category_Name	Varchar(30)			x	Tên thể loại
Trangthai				x	Trạng thái thể loại

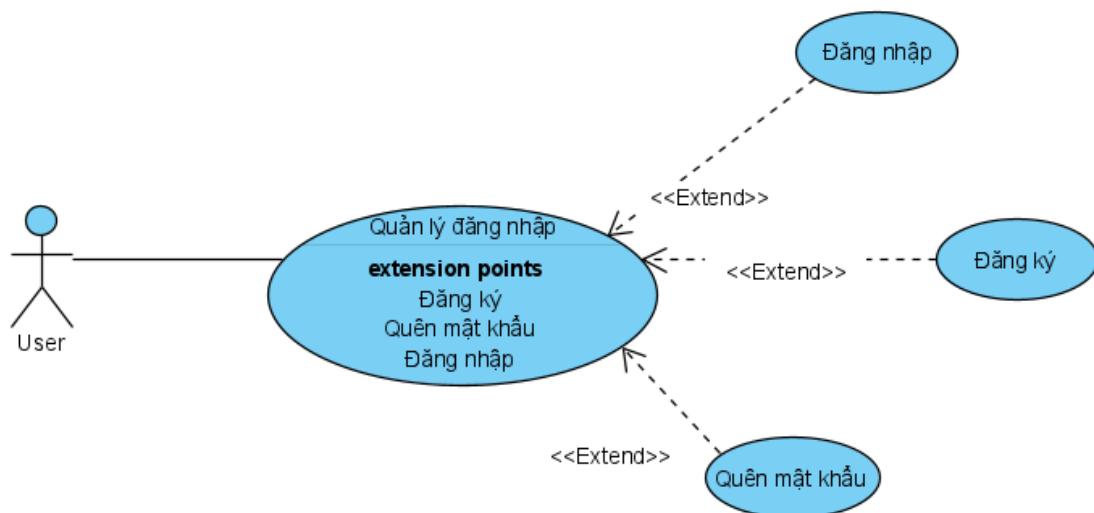
Bảng 5-24: Bảng mô tả dữ liệu Blog_Category

Chương 6. THIẾT KẾ CHỨC NĂNG

MODULE USER

6.1 CHỨC NĂNG QUẢN LÝ ĐĂNG NHẬP

6.1.1 Sơ đồ use case chức năng quản lý đăng nhập



Hình 6-1: Sơ đồ usecase chức năng đăng nhập

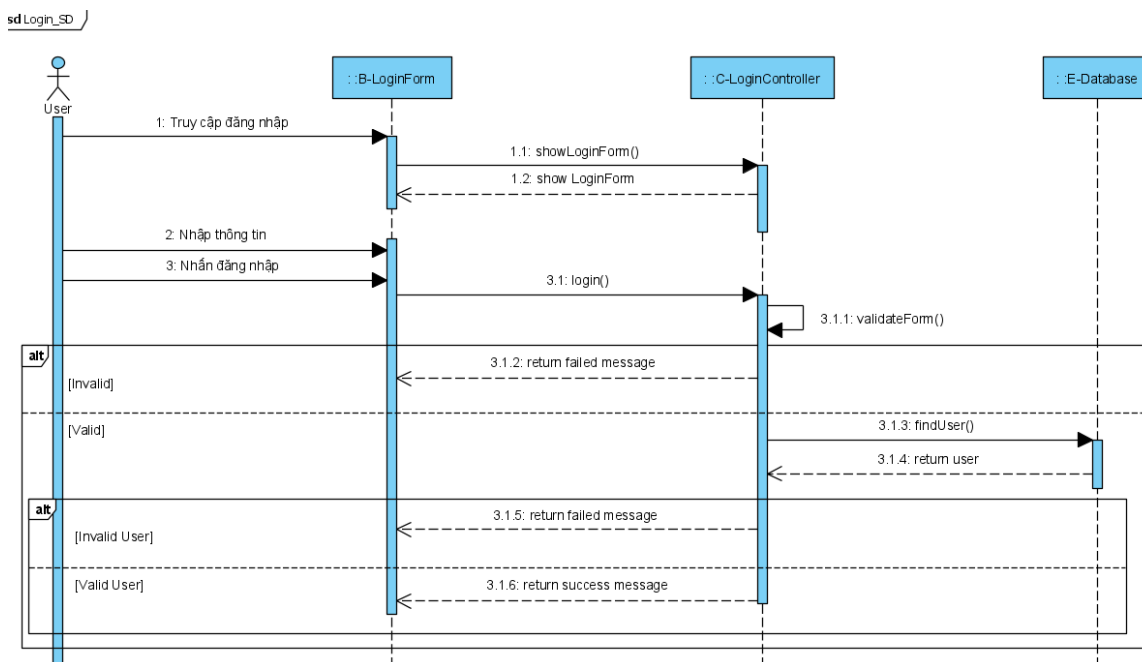
Tên Use case	Quản lý đăng nhập
Actor	User
Mô tả	Actor đăng nhập vào hệ thống.
Pre-conditions	Actor chưa đăng nhập vào hệ thống và có tài khoản
Post-conditions	Thành công: User đăng nhập vào trang cá nhân của mình Không thành công: Thông báo lỗi đăng nhập

Luồng sự kiện chính	<p>Actor vào trang đăng nhập có thể sử dụng các chức năng Đăng nhập, đăng ký, quên mật khẩu.</p> <ul style="list-style-type: none"> - Extend use case Đăng nhập - Extend use case Đăng ký - Extend use case Quên mật khẩu
Luồng sự kiện phụ	Actor đăng xuất, hệ thống trở về trang chủ
<< Extend use case Đăng nhập >>	<p>Đăng nhập</p> <ol style="list-style-type: none"> 1. Actor chọn chức năng đăng nhập 2. Hệ thống hiển thị form đăng nhập 3. Actor nhập thông tin đăng nhập 4. Actor click Đăng nhập 5. Hệ thống kiểm tra thông tin hợp lệ 6. Hệ thống tìm kiếm thông tin của actor 7. Hệ thống thông báo kết quả đăng nhập <p>Rẽ nhánh 1: 5.1. Kiểm tra thông tin không hợp lệ thì xuất thông báo lỗi và lặp lại bước 3</p> <p>Rẽ nhánh 2: 6.1. Kiểm tra thông tin không hợp lệ thì xuất thông báo lỗi và lặp lại bước 3</p>
<< Extend use case Đăng ký >>	<p>Đăng ký</p> <ol style="list-style-type: none"> 1. Actor chọn đăng ký tài khoản 2. Hệ thống sẽ hiển thị trang đăng ký 3. Actor điền thông tin đăng ký vào biểu mẫu 4. Actor nhấn đăng ký 5. Hệ thống kiểm tra thông tin hợp lệ. 6. Hệ thống kiểm tra email đã sử dụng hay chưa 7. Hệ thống lưu thông tin và gửi đường dẫn xác nhận email cho người dùng 8. Actor bấm vào đường dẫn xác nhận 9. Chuyển sang trang cá nhân của actor <p>Rẽ nhánh 1: 5.1. Kiểm tra thông tin không hợp lệ thì xuất thông báo lỗi và lặp lại bước 3</p> <p>Rẽ nhánh 2: 6.1. Kiểm tra email đã tồn tại và lặp lại bước 3</p>

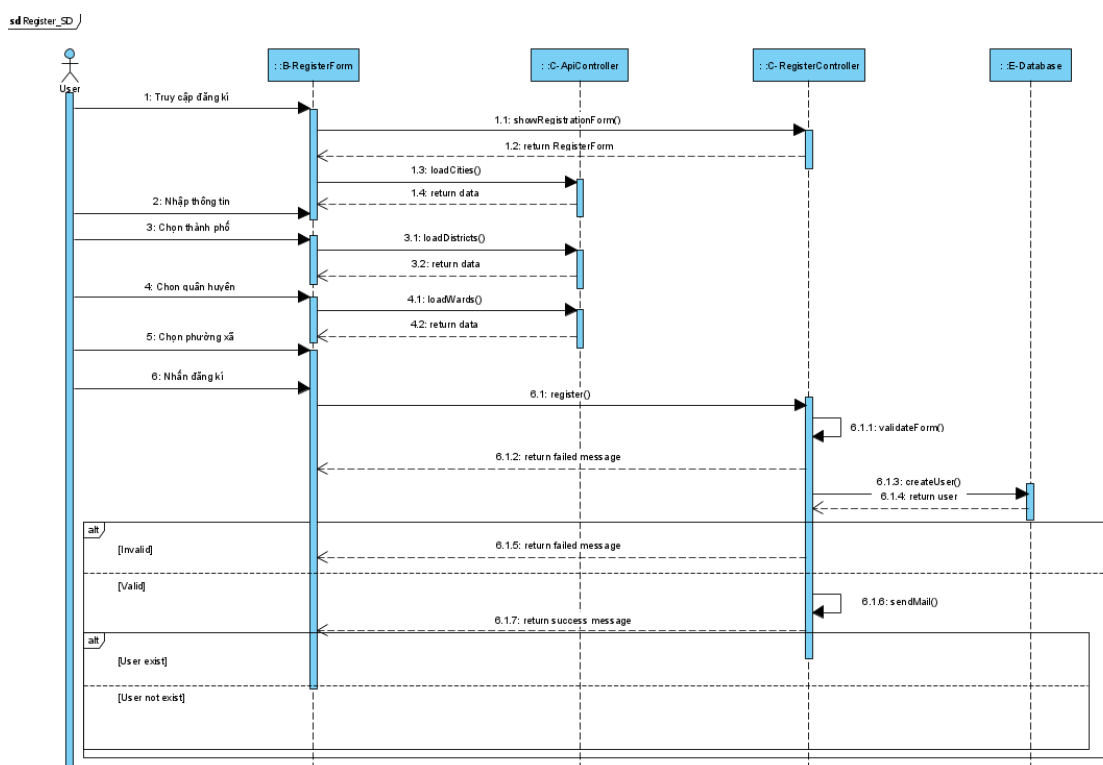
<p><< Extend use case Quên mật khẩu >></p>	<p>Quên mật khẩu</p> <ol style="list-style-type: none"> 1. Actor chọn quên mật khẩu 2. Actor nhập email tài khoản cần cấp lại mật khẩu 3. Hệ thống kiểm tra thông tin nhập vào 4. Hệ thống kiểm tra email có tồn tại hay chưa 5. Hệ thống sẽ gửi email cập nhật lại mật khẩu 6. Actor vào mail đã đăng ký để cập nhật lại mật khẩu 7. Hệ thống hiển thị giao diện 8. Actor nhập thông tin vào form 9. Actor nhấn cấp lại mật khẩu 10. Hệ thống kiểm tra thông tin hợp lệ 11. Cập nhật CSDL 12. Chuyển sang trang cá nhân <p>Rẽ nhánh 1: 3.1. Kiểm tra email không hợp lệ thì xuất thông báo lỗi và lặp lại bước 2</p> <p>Rẽ nhánh 2: 4.1. Kiểm tra email không tồn tại và lặp lại bước 2</p> <p>Rẽ nhánh 3 : 9.1. Kiểm tra thông tin mật khẩu mới không hợp lệ và lặp lại bước 7</p>
--	--

Bảng 6-1: Bảng mô tả usecase quản lý đăng nhập

6.1.2 Sơ đồ tuần tự chức năng quản lý đăng nhập

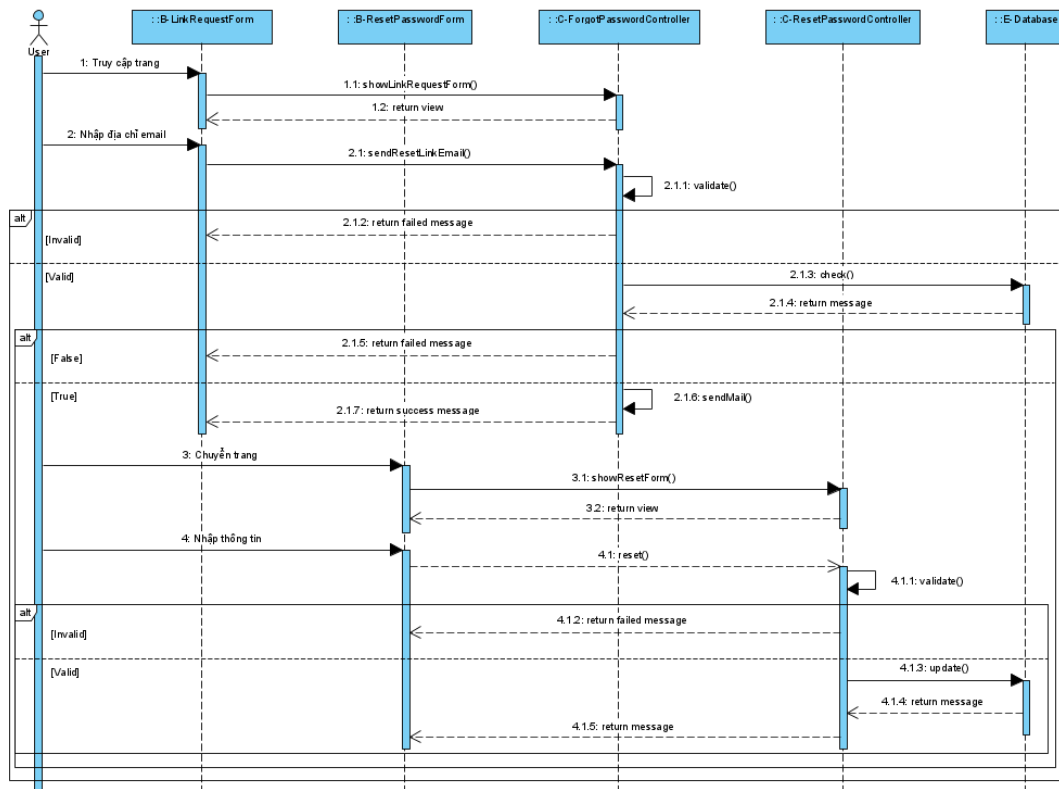


Hình 6-2: Sơ đồ tuần tự chức năng Đăng nhập



Hình 6-3: Sơ đồ tuần tự chức năng Đăng ký

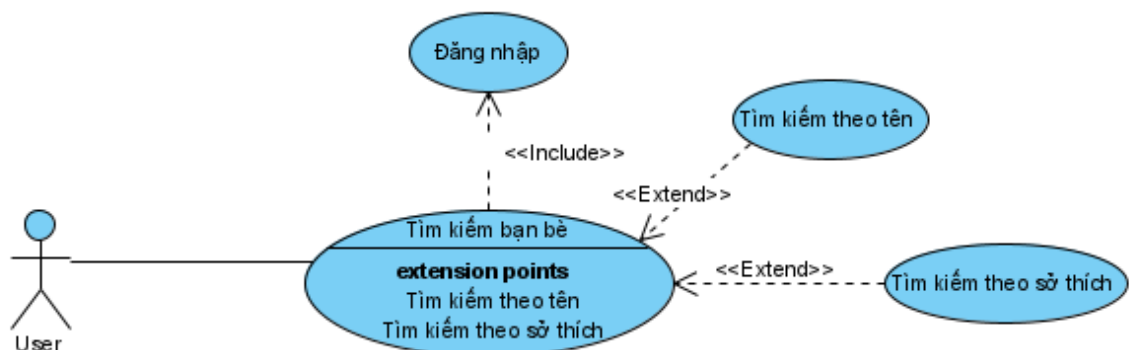
sđ QuênMậtKhau_SD



Hình 6-4: Sơ đồ chức năng Quên mật khẩu

6.2 CHỨC NĂNG TÌM KIẾM BẠN BÈ

6.2.1 Sơ đồ use case chức năng tìm kiếm bạn bè



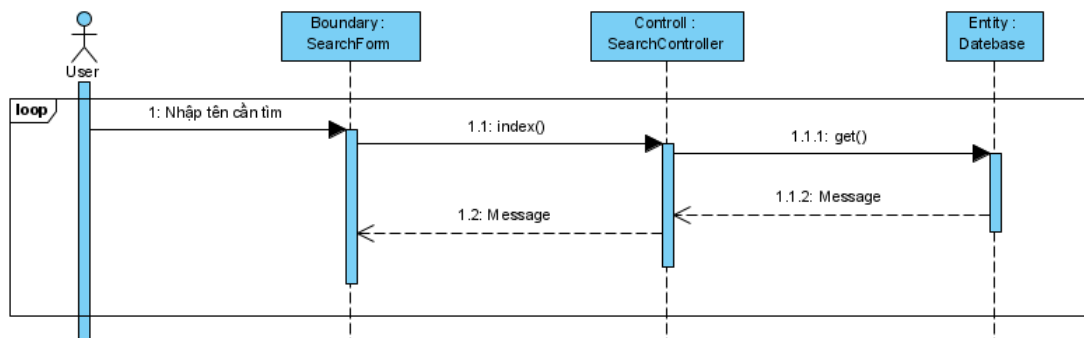
Hình 6-5: Sơ đồ use case tìm kiếm bạn bè

Tên Use case	Tìm kiếm bạn bè
Actor	User
Mô tả	User có thể tìm kiếm bạn bè theo 2 tiêu chí là tên và sở thích
Pre-conditions	Đăng nhập
Post-conditions	Thành công: Hiển thị danh sách kết quả tìm được Không thành công: Hiển thị danh sách rỗng và xuất thông báo không tìm thấy
Luồng sự kiện chính	Actor chọn chức năng Tìm kiếm <ul style="list-style-type: none"> - Include use case Đăng nhập - Extend use case Tìm kiếm theo tên - Extend use case Tìm kiếm theo sở thích
Luồng sự kiện phụ	
<< Extend use case Tìm kiếm theo tên >>	Tìm kiếm theo tên <ol style="list-style-type: none"> 1. Actor nhập tên người dùng cần tìm 2. Hệ thống kiểm tra trong CSDL 3. Hiển thị danh sách kết quả
<< Extend use case Tìm kiếm theo sở thích >>	Tìm kiếm theo sở thích <ol style="list-style-type: none"> 1. Actor nhập sở thích cần tìm 2. Hệ thống kiểm tra từ khóa trong CSDL 3. Hiển thị danh sách kết quả

Bảng 6-2: Bảng mô tả usecase tìm kiếm bạn bè

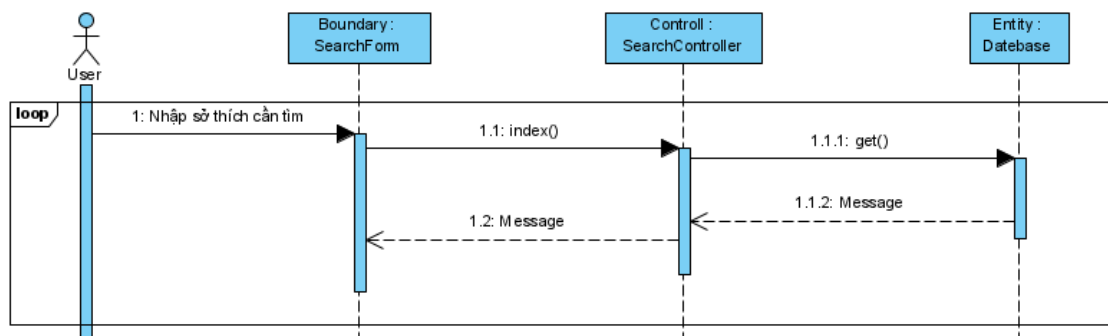
6.2.2 Sơ đồ tuần tự chức năng tìm kiếm bạn bè

sd Tìm kiếm theo tên



Hình 6-6: Sơ đồ tuần tự chức năng tìm kiếm theo tên

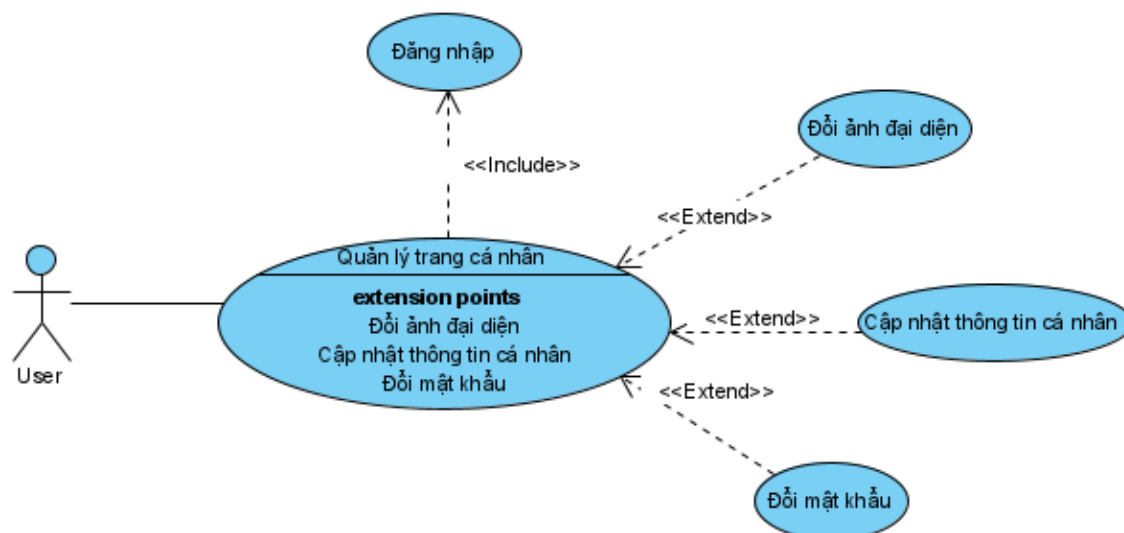
sd Tìm kiếm theo sở thích



Hình 6-7: Sơ đồ tuần tự chức năng tìm kiếm theo sở thích

6.3 CHỨC NĂNG QUẢN LÝ TRANG CÁ NHÂN

6.3.1 Sơ đồ use case chức năng quản lý trang cá nhân



Hình 6-8: Sơ đồ use case quản lý trang cá nhân

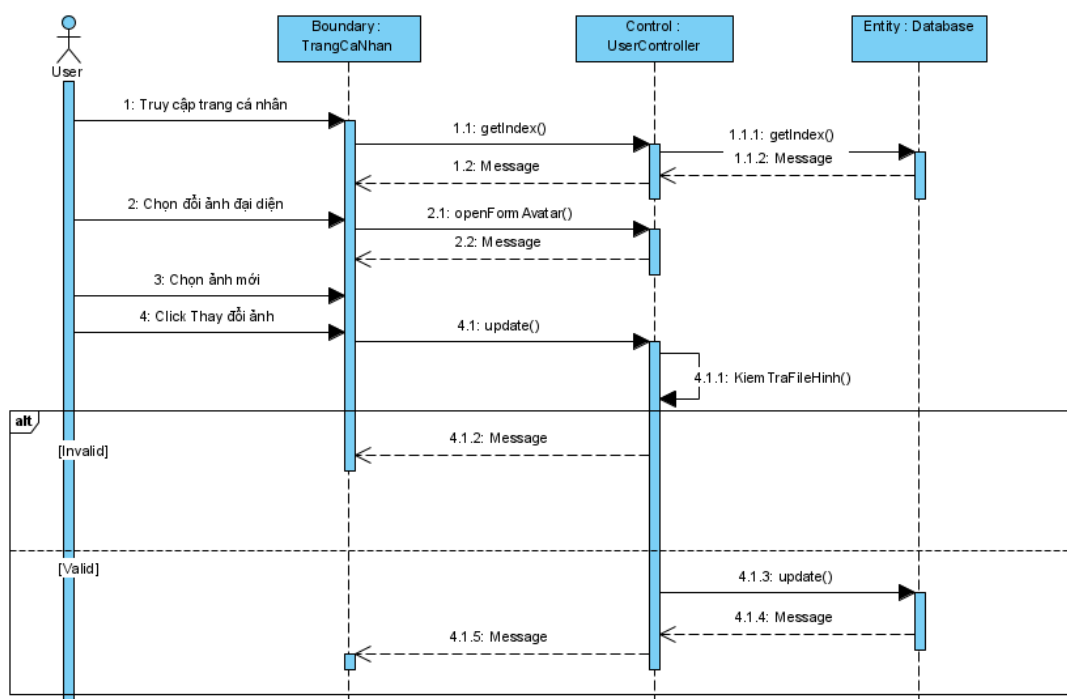
Tên Use case	Quản lý trang cá nhân
Actor	User
Mô tả	Actor có thể thay đổi ảnh đại diện, cập nhật thêm một số thông tin để hiển thị, chỉnh sửa tường nhà. Ngoài ra Actor có thể đổi mật khẩu tài khoản của mình.
Pre-conditions	Đăng nhập
Post-conditions	Thành công: Trang cá nhân được cập nhật thông tin mới Không thành công: Thông báo lỗi
Luồng sự kiện chính	Actor chọn chức năng Quản lý trang cá nhân Hệ thống mở trang quản lý trang cá nhân <ul style="list-style-type: none"> - Include use case Đăng nhập - Extend use case Đổi ảnh đại diện - Extend use case Cập nhật thông tin cá nhân - Extend use case Đổi mật khẩu
Luồng sự kiện phụ	Actor click sang chức năng khác, hệ thống không lưu lại những thông tin đã điền

<p><< Extend use case Đổi ảnh đại diện >></p>	<p>Đổi ảnh đại diện</p> <ol style="list-style-type: none"> 1. Actor chọn chức năng đổi ảnh đại diện 2. Hệ thống mở form đổi ảnh 3. Actor chọn ảnh mới 4. Actor click Thay đổi ảnh 5. Hiện thị kiểm tra file ảnh 6. Cập nhật CSDL 7. Hiện thị trang chủ với hình đại diện mới của Actor <p>Rẽ nhánh 1: 5.1. Kiểm tra file ảnh không hợp lệ thì không đổi ảnh và lặp lại bước 1</p>
<p><< Extend use case Cập nhật thông tin cá nhân >></p>	<p>Cập nhật thông tin cá nhân</p> <ol style="list-style-type: none"> 1. Actor chọn chức năng Sửa thông tin 2. Hệ thống mở trang sửa thông tin 3. Actor nhập thông tin mới 4. Actor click Cập nhật 5. Hệ thống kiểm tra thông tin Actor đã nhập 6. Cập nhật CSDL 7. Trở về trang cá nhân với những thông tin đã cập nhật <p>Rẽ nhánh 1: 5.1. Thông tin không hợp lệ sẽ xuất thông báo lỗi và quay về bước 3</p>
<p><< Extend use case Đổi mật khẩu >></p>	<p>Đổi mật khẩu</p> <ol style="list-style-type: none"> 1. Actor chọn Đổi mật khẩu 2. Hệ thống mở trang Đổi mật khẩu 3. Actor nhập mật khẩu mới 4. Hệ thống kiểm tra thông tin hợp lệ 5. Cập nhật CSDL 6. Hiện thị thông tin cập nhật <p>Rẽ nhánh 1: 4.1, Hệ thống kiểm tra thông tin không hợp lệ thì xuất thông báo lỗi và lặp lại bước 3</p>

Bảng 6-3: Bảng mô tả usecase quản lý trang cá nhân

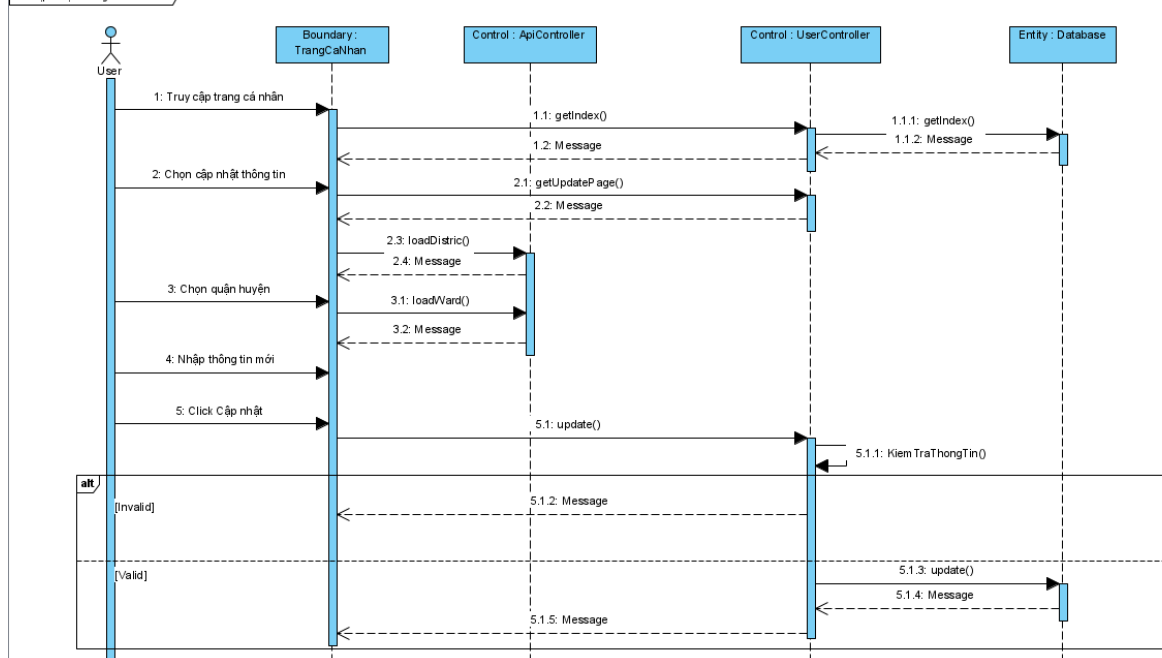
6.3.2 Sơ đồ tuần tự chức năng quản lý trang cá nhân

sd Đổi ảnh đại diện



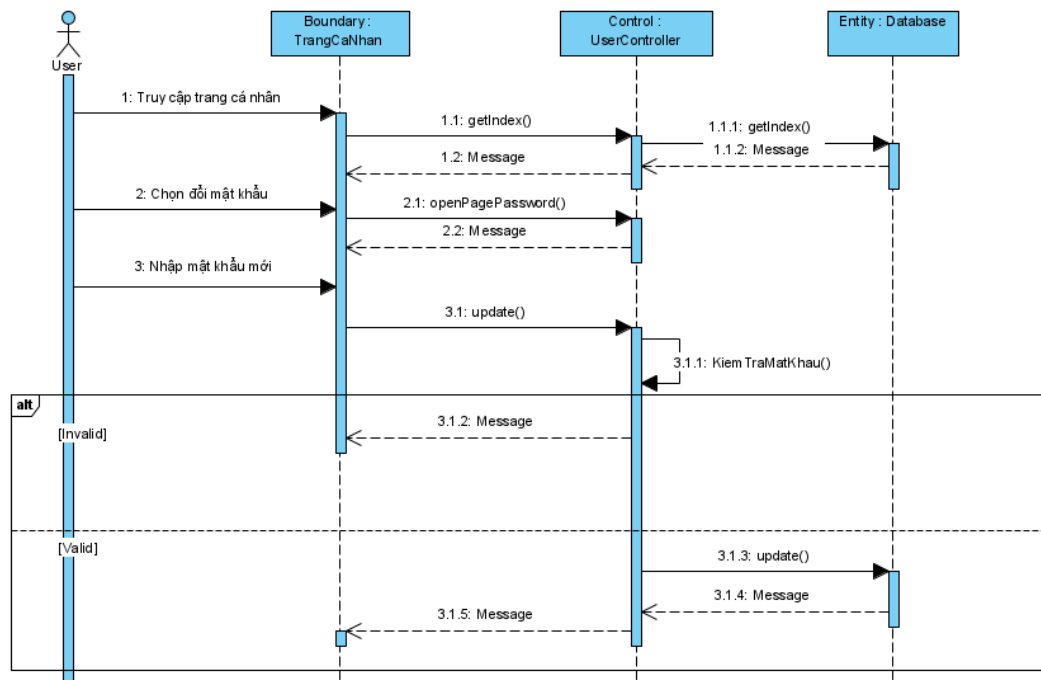
Hình 6-9: Sơ đồ tuần tự chức năng Đổi ảnh đại diện

sd Cập nhật thông tin cá nhân



Hình 6-10: Sơ đồ tuần tự chức năng Cập nhật thông tin

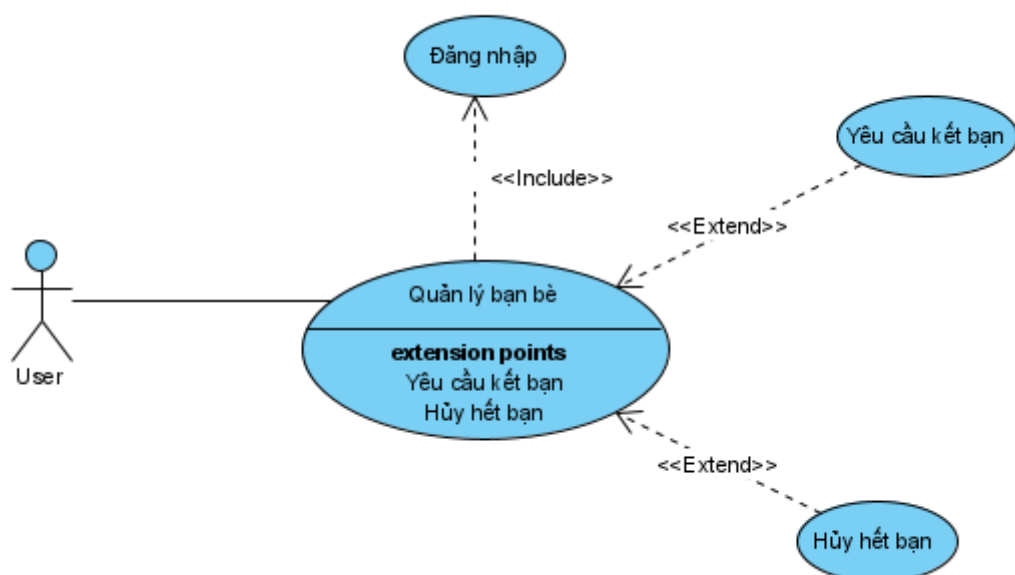
sd Đổi mật khẩu



Hình 6-11: Sơ đồ tuần tự chức năng Đổi mật khẩu

6.4 CHỨC NĂNG QUẢN LÝ BẠN BÈ

6.4.1 Sơ đồ use case chức năng quản lý bạn bè

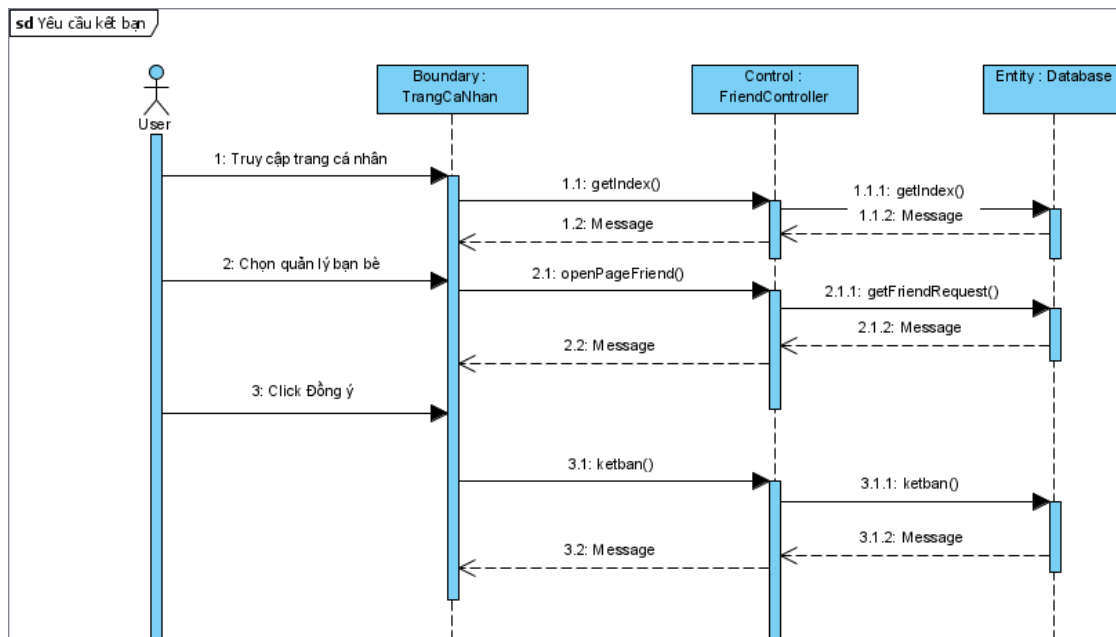


Hình 6-12: Sơ đồ use case quản lý bạn bè

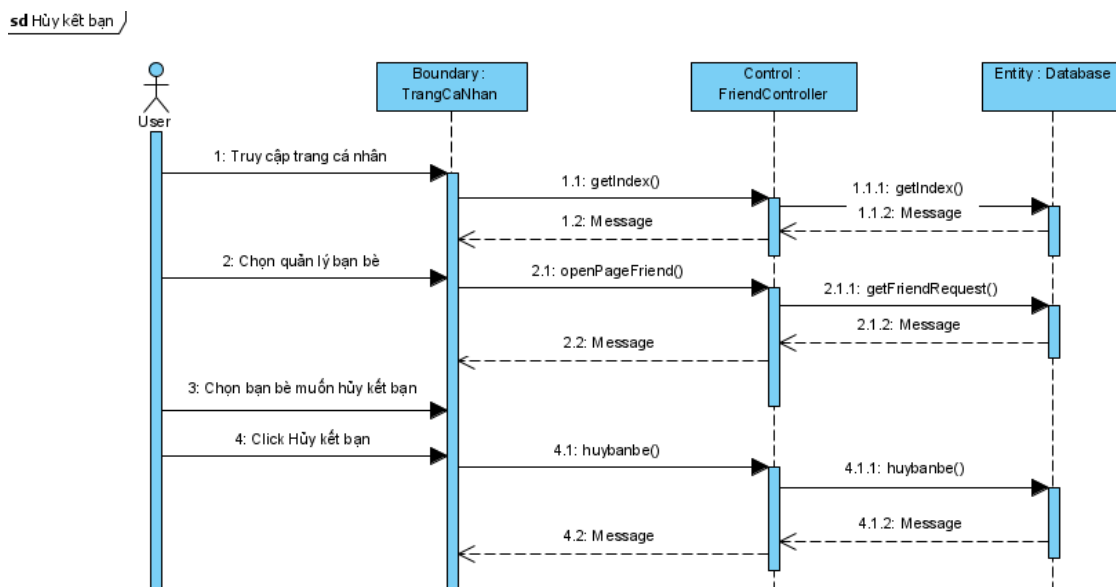
Tên Use case	Quản lý bạn bè
Actor	User
Mô tả	Hiển thị các lời mời kết bạn đến Actor, danh sách bạn bè và chức năng hủy kết bạn trong danh sách.
Pre-conditions	Đăng nhập
Post-conditions	Thành công: Danh sách bạn bè được cập nhật Không thành công: Thông báo lỗi
Luồng sự kiện chính	Actor chọn chức năng Quản lý bạn bè Hệ thống mở trang quản lý bạn bè <ul style="list-style-type: none"> - Include use case Đăng nhập - Extend use case Yêu cầu kết bạn - Extend use case Hủy kết bạn
Luồng sự kiện phụ	Actor đăng xuất hoặc dùng chức năng khác trong hệ thống
<< Extend use case Yêu cầu kết bạn >>	Yêu cầu kết bạn <ol style="list-style-type: none"> 1. Hệ thống hiển thị các lời mời kết bạn tới người dùng 2. Actor click Đồng ý 3. Cập nhật CSDL 4. Hiển thị danh sách bạn bè cập nhật
<< Extend use case Hủy kết bạn >>	Hủy kết bạn <ol style="list-style-type: none"> 1. Actor chọn hủy kết bạn trong danh sách bạn bè 2. Cập nhật CSDL 3. Hiển thị danh sách bạn bè cập nhật

Bảng 6-4: Bảng mô tả usecase quản lý bạn bè

6.4.2 Sơ đồ tuần tự chức năng quản lý bạn bè



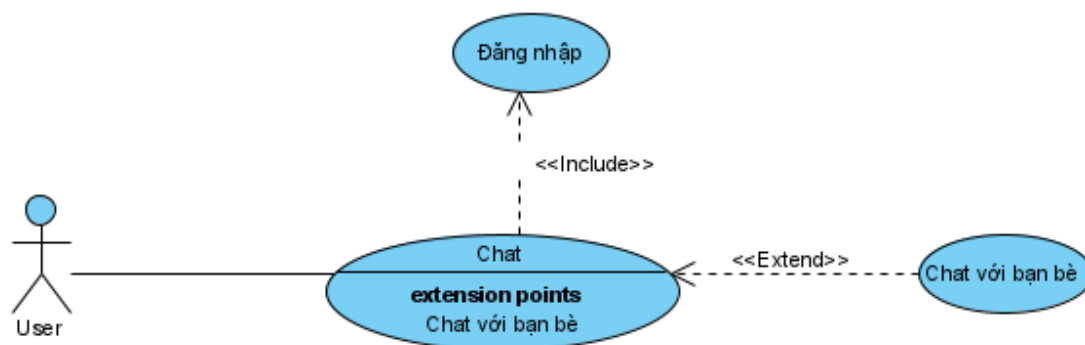
Hình 6-13: Sơ đồ tuần tự chức năng Kết bạn



Hình 6-14: Sơ đồ tuần tự chức năng Hủy kết bạn

6.5 CHỨC NĂNG CHAT

6.5.1 Sơ đồ use case chức năng Chat



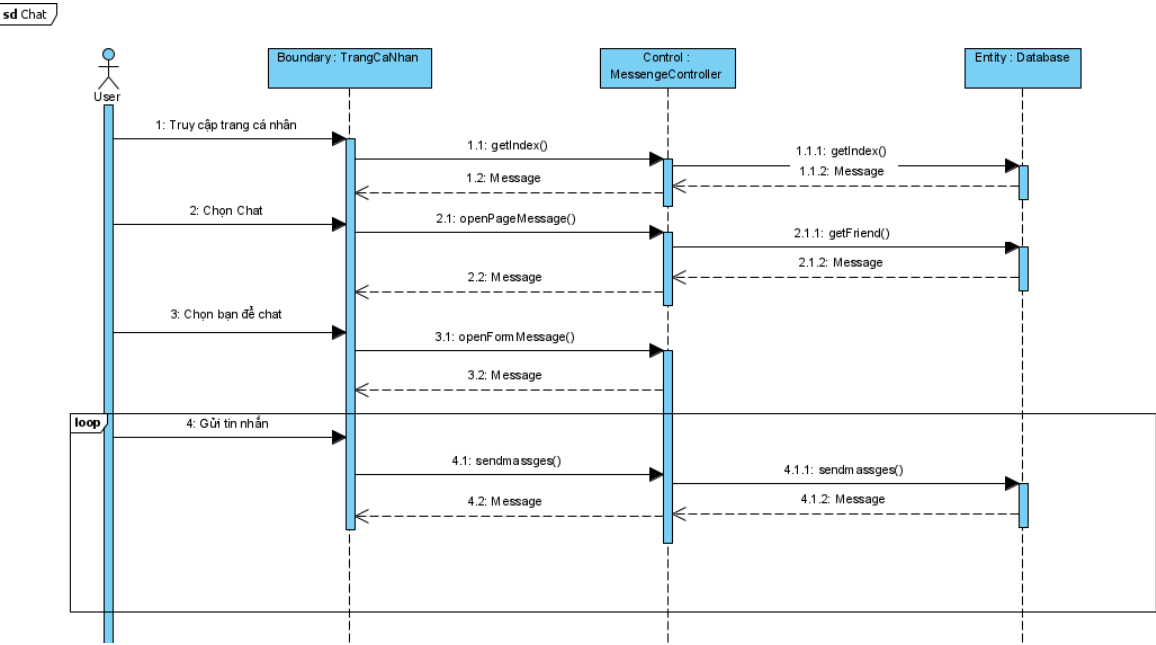
Hình 6-15: Sơ đồ use case tìm kiếm bạn bè

Tên Use case	Chat
Actor	User
Mô tả	Actor trò chuyện với bạn bè của mình
Pre-conditions	Đăng nhập
Post-conditions	Thành công: Hiển thị đoạn hội thoại của actor và bạn Không thành công: Thông báo lỗi
Luồng sự kiện chính	Actor chọn chức năng Chat Hệ thống mở trang Chat <ul style="list-style-type: none"> - Include use case Đăng nhập - Extend use case Chat với bạn bè
Luồng sự kiện phụ	Actor đăng xuất hoặc dùng chức năng khác
<< Extend use case Chat với bạn bè >>	Chat với bạn bè <ol style="list-style-type: none"> 1. Hệ thống hiển thị danh sách bạn bè của Actor 2. Actor click vào bạn của mình để bắt đầu chat 3. Hệ thống mở cửa sổ chat 4. Actor gửi tin nhắn cho bạn mình 5. Cập nhật CSDL

	6. Hiện thị tin nhắn của actor
--	--------------------------------

Bảng 6-5: Bảng mô tả usecase Chat

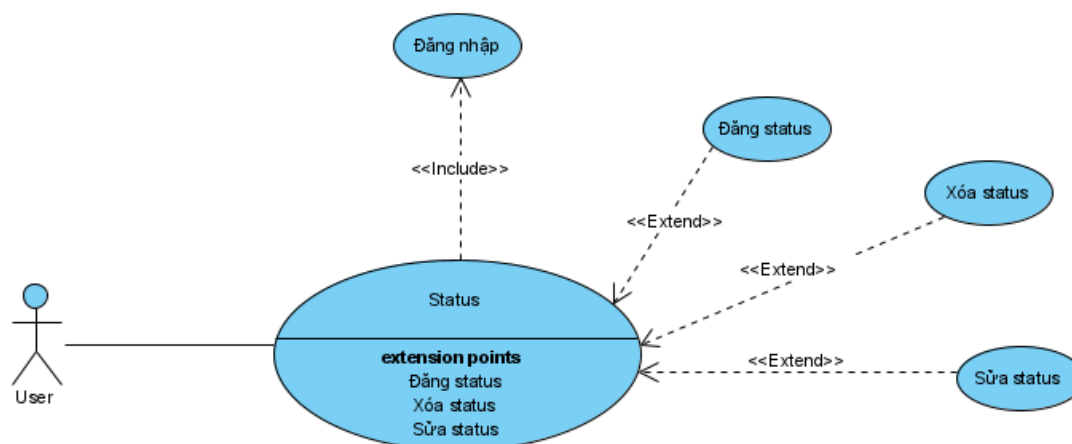
6.5.2 Sơ đồ tuần tự chức năng Chat



Hình 6-16: Sơ đồ tuần tự chức năng Chat với bạn bè

6.6 CHỨC NĂNG STATUS

6.6.1 Sơ đồ use case chức năng Status



Hình 6-17: Sơ đồ use case Status

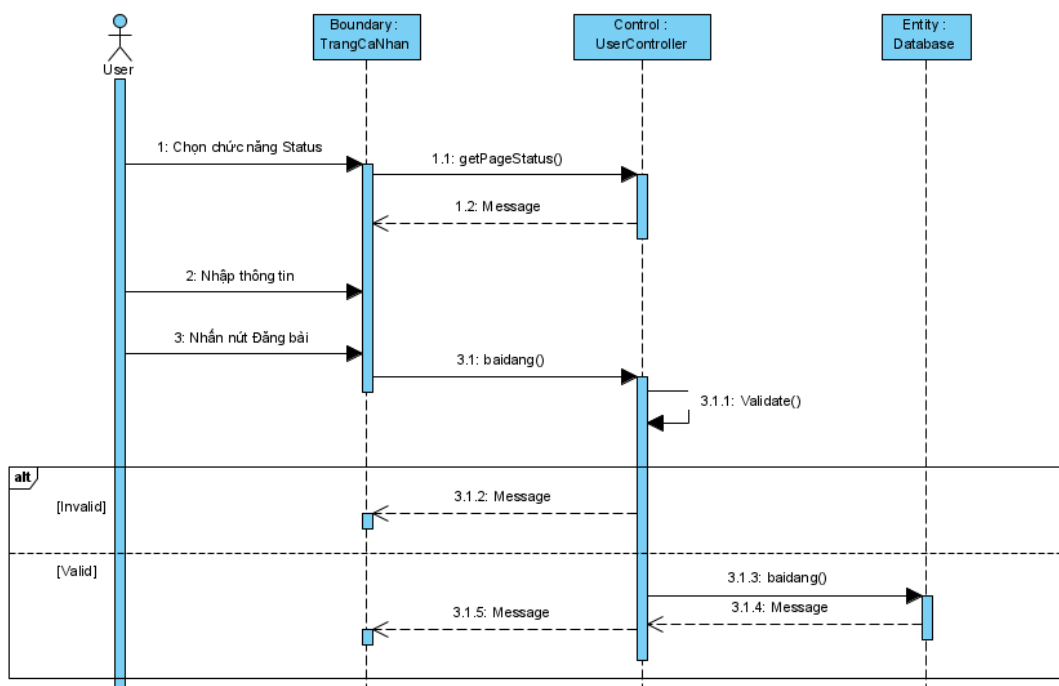
Tên Use case	Status
Actor	User
Mô tả	Actor có thể đăng các bài status kèm theo hình ảnh. Sau khi đăng có thể chỉnh sửa lại nội dung bên trong hoặc hình ảnh đi kèm. Nếu không thích nữa có thể xóa status đó đi. Ngoài ra actor còn có thể like bài viết của mình hoặc của bạn bè, hệ thống hiển thị số lượt like của bài status. Hệ thống cung cấp thêm cho actor chức năng bình luận bài viết, phản hồi bình luận và xem các phản hồi có trong bài.
Pre-conditions	Đăng nhập
Post-conditions	Thành công: Hiển thị thông tin các bài đăng của actor lên trang chủ, trang cá nhân. Không thành công: Thông báo lỗi
Luồng sự kiện chính	Actor chọn chức năng Status Hệ thống mở trang Status <ul style="list-style-type: none"> - Include use case Đăng nhập - Extend use case Đăng status

	<ul style="list-style-type: none"> - Extend use case Xóa status - Extend use case Sửa status - Extend use case Like - Extend use case Bình luận
Luồng sự kiện phụ	Actor đăng xuất hoặc dùng chức năng khác trong hệ thống
<< Extend use case Đăng status >>	Đăng status <ol style="list-style-type: none"> 1. Actor nhập thông tin cho Status 2. Actor click Đăng 3. Hệ thống kiểm tra thông tin hợp lệ 4. Cập nhật CSDL 5. Hiện thị danh sách Status ở trang chủ <p>Rẽ nhánh 1: 3.1, Kiểm tra thông tin không hợp lệ thì xuất thông báo lỗi và lặp lại bước 1</p>
<< Extend use case Xóa status >>	Xóa status <ol style="list-style-type: none"> 1. Actor truy cập trang cá nhân của mình 2. Hệ thống hiển thị các status của actor 3. Actor chọn status muốn xóa 4. Actor click Xóa bài viết 5. Hệ thống kiểm tra quyền sở hữu bài viết 6. Cập nhật CSDL 7. Hiện thị danh sách bài đăng cập nhật <p>Rẽ nhánh 1: 5.1, Bài viết không thuộc sở hữu của Actor thì không được và quay về trang chủ</p>
<< Extend use case Sửa status >>	Sửa status <ol style="list-style-type: none"> 1. Actor chọn status muốn sửa 2. Hệ thống load thông tin status được chọn 3. Actor nhập thông tin cần sửa 4. Hệ thống kiểm tra thông tin hợp lệ 5. Cập nhật CSDL 6. Hiện thị bài đăng sau khi cập nhật <p>Rẽ nhánh 1: 4.1, Hệ thống kiểm tra thông tin không hợp lệ thì xuất thông báo lỗi và lặp lại bước 3</p>

Bảng 6-6: Bảng mô tả usecase Status

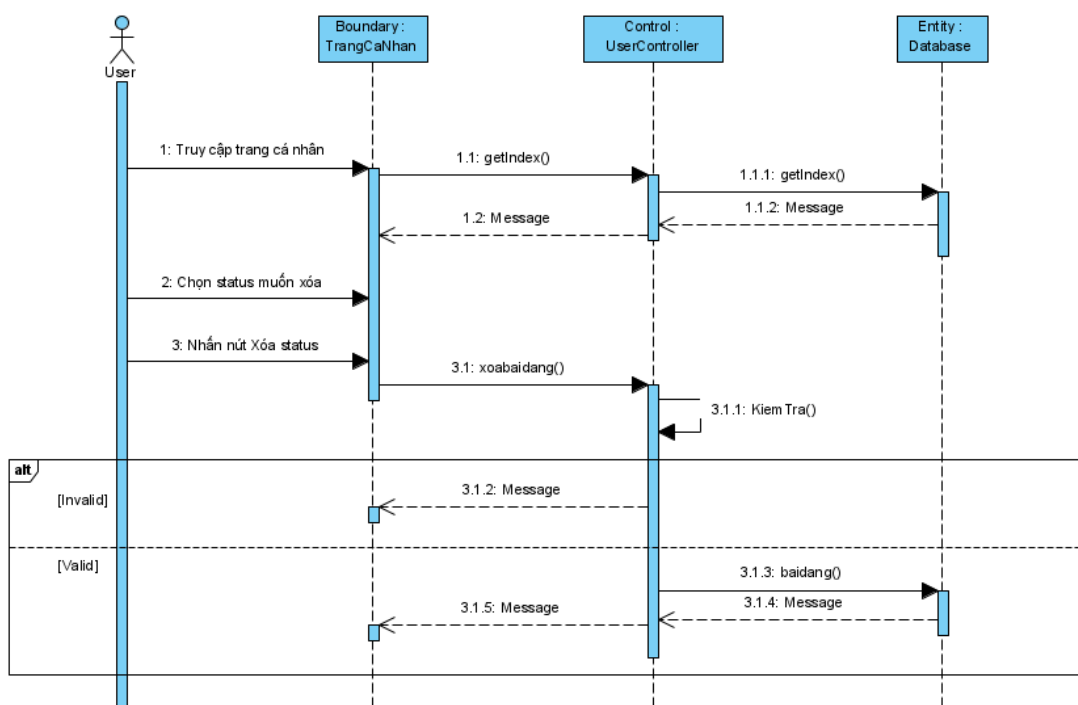
6.6.2 Sơ đồ tuần tự chức năng Status

sd Đăng status /

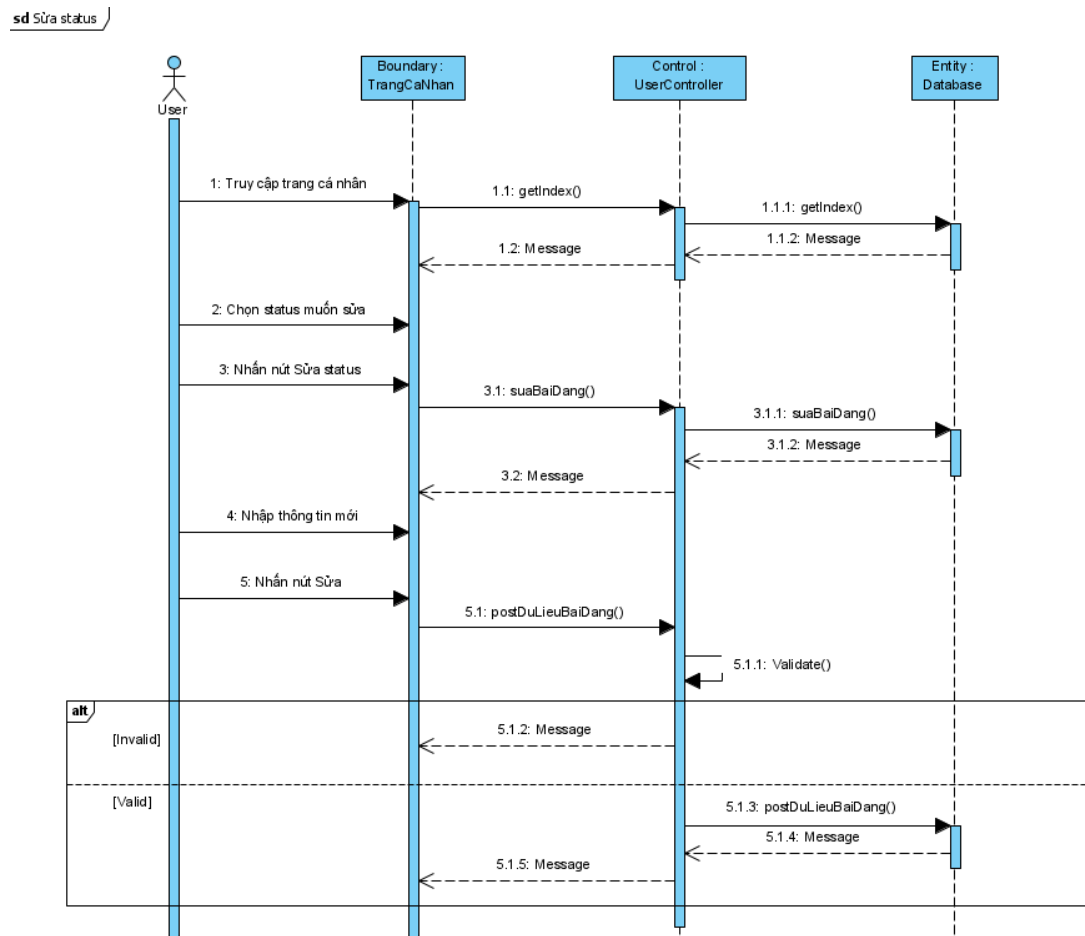


Hình 6-18: Sơ đồ tuần tự chức năng Đăng status

sd Xóa status /



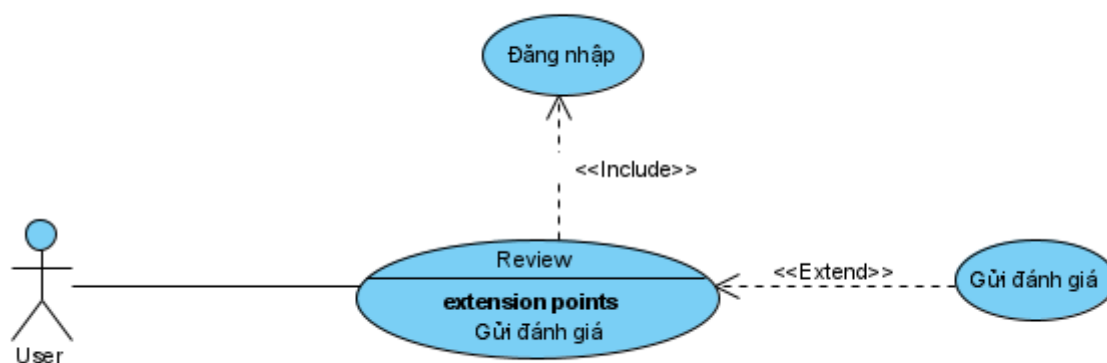
Hình 6-19: Sơ đồ tuần tự chức năng Xóa status



Hình 6-20: Sơ đồ tuần tự chức năng Sửa status

6.7 CHỨC NĂNG REVIEW

6.7.1 Sơ đồ use case chức năng Review



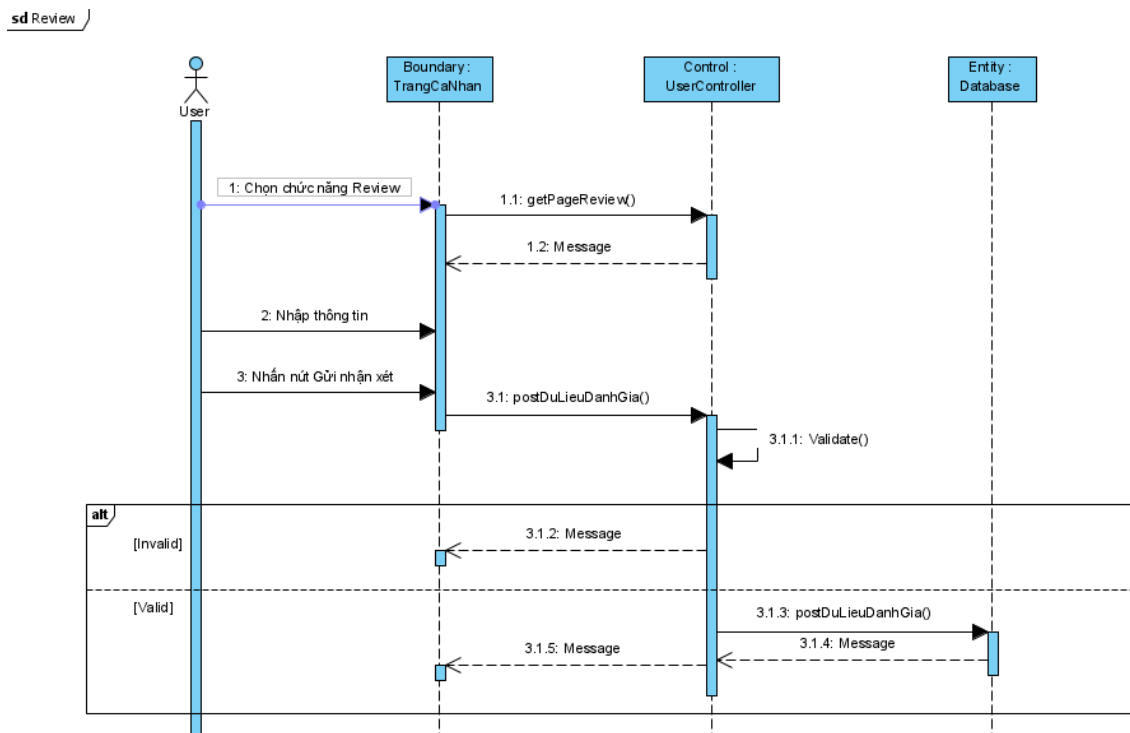
Hình 6-21: Sơ đồ use case Review

Tên Use case	Review
Actor	User
Mô tả	Actor có thể gửi đánh giá của mình về việc trải nghiệm ứng dụng cho Admin. Mỗi actor chỉ được thực hiện đánh giá 01 lần.
Pre-conditions	Đăng nhập
Post-conditions	Thành công: Gửi đánh giá tới Admin Không thành công: Thông báo lỗi
Luồng sự kiện chính	Actor chọn chức năng Review Hệ thống mở trang Review <ul style="list-style-type: none"> - Include use case Đăng nhập - Extend use case Gửi đánh giá
Luồng sự kiện phụ	Actor đăng xuất hoặc dùng chức năng khác trong hệ thống
<< Extend use case Gửi đánh giá >>	Gửi đánh giá <ol style="list-style-type: none"> 1. Actor nhập thông tin cảm nhận của mình về website 2. Actor click Gửi đánh giá 3. Hệ thống kiểm tra thông tin hợp lệ

	<p>4. Cập nhật CSDL</p> <p>Rẽ nhánh 1:</p> <p>3.1, Kiểm tra thông tin không hợp lệ thì xuất thông báo lỗi và lặp lại bước 1</p>
--	---

Bảng 6-7: Bảng mô tả usecase Review

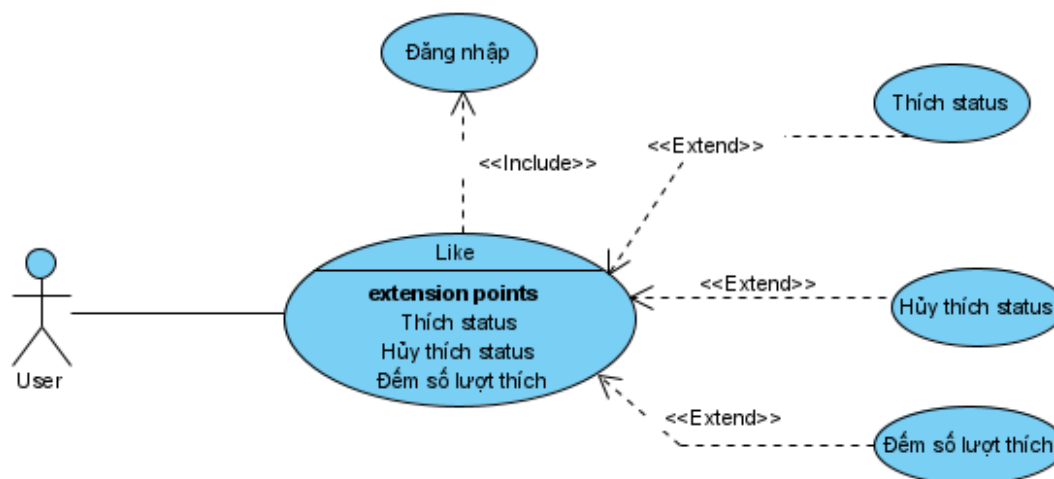
6.7.2 Sơ đồ tuần tự chức năng Review



Hình 6-22: Sơ đồ tuần tự chức năng Gửi đánh giá

6.8 CHỨC NĂNG LIKE

6.8.1 Sơ đồ use case chức năng quản lý đăng nhập



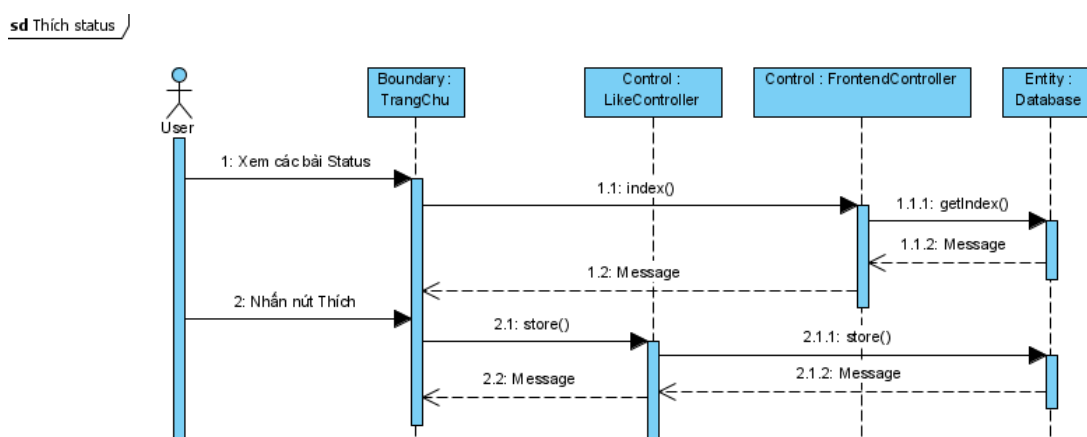
Hình 6-23: Sơ đồ usecase chức năng Like

Tên Use case	Like
Actor	User
Mô tả	Actor có thể thích/hủy thích một bài status bất kỳ. Hệ thống hiển thị số lượt thích của bài viết đó cho các actor.
Pre-conditions	Đăng nhập
Post-conditions	Thành công: Cập nhật số lượt thích của status Không thành công: Thông báo lỗi
Luồng sự kiện chính	Actor nhấn Thích status. <ul style="list-style-type: none"> - Extend use case Thích status - Extend use case Hủy thích status - Extend use case Đếm số lượt thích
Luồng sự kiện phụ	Actor đăng xuất, hệ thống trở về trang chủ
<< Extend use case Thích status >>	Thích status <ol style="list-style-type: none"> 1. Actor xem các bài status ở trang chủ 2. Hệ thống hiển thị các bài status lên trang chủ 3. Actor click Thích

	4. Cập nhật CSDL
<< Extend use case Hủy thích status >>	Hủy thích status <ol style="list-style-type: none"> 1. Hệ thống hiển thị các bài status lên trang chủ 2. Hiển thị thông tin Thích của actor 3. Actor nhấn lại Thích 4. Cập nhật CSDL Rẽ nhánh 1: 3.1. Kiểm tra nếu actor đã thích status và nhấn tiếp nút Thích thì sẽ hủy Thích bài viết và quay lại bước 2
<< Extend use case Đếm số lượt thích >>	Đếm số lượt thích <ol style="list-style-type: none"> 1. Hệ thống hiển thị số lượt like của status

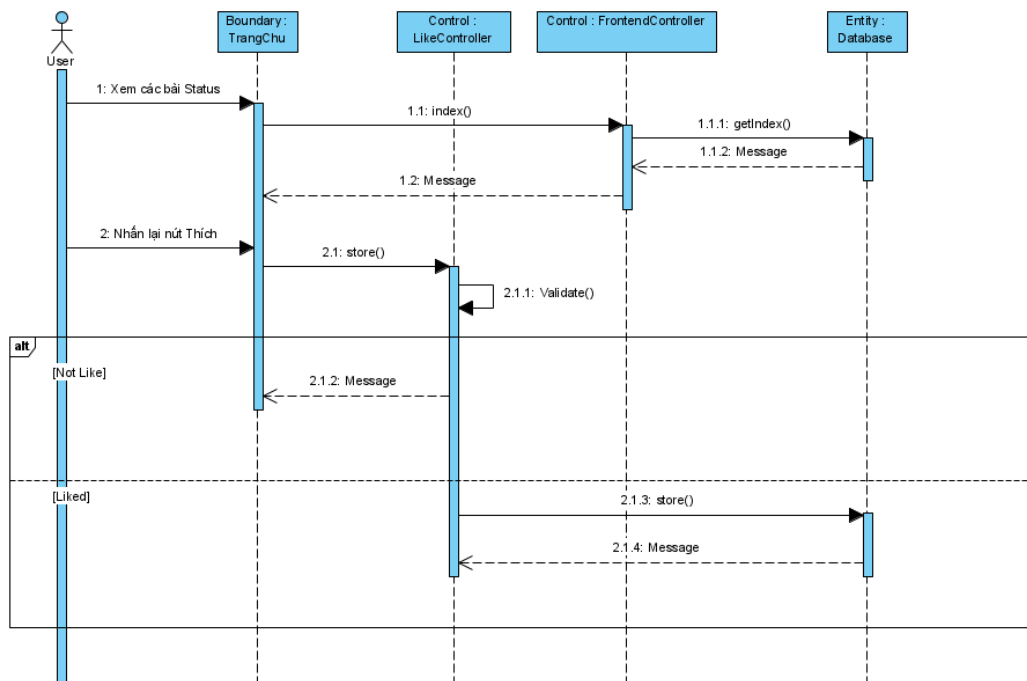
Bảng 6-8: Bảng mô tả usecase Like

6.8.2 Sơ đồ tuần tự chức năng quản lý đăng nhập



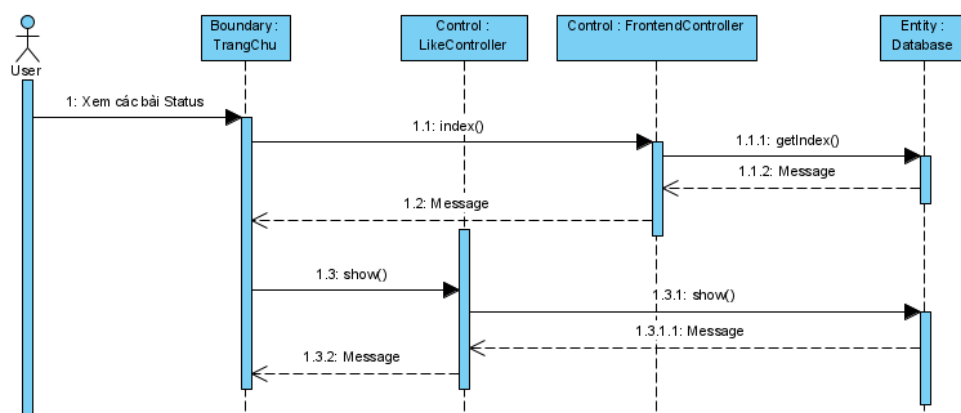
Hình 6-24: Sơ đồ tuần tự chức năng Thích status

sd Hủy thích Status



Hình 6-25: Sơ đồ tuần tự chức năng Hủy thích status

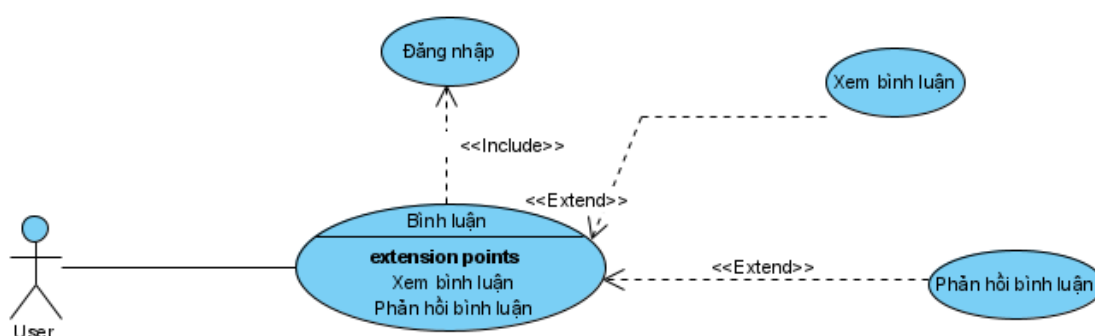
sd Đếm số lượt Thích



Hình 6-26: Sơ đồ chức năng Đếm số lượt thích

6.9 CHỨC NĂNG BÌNH LUẬN

6.9.1 Sơ đồ use case chức năng bình luận



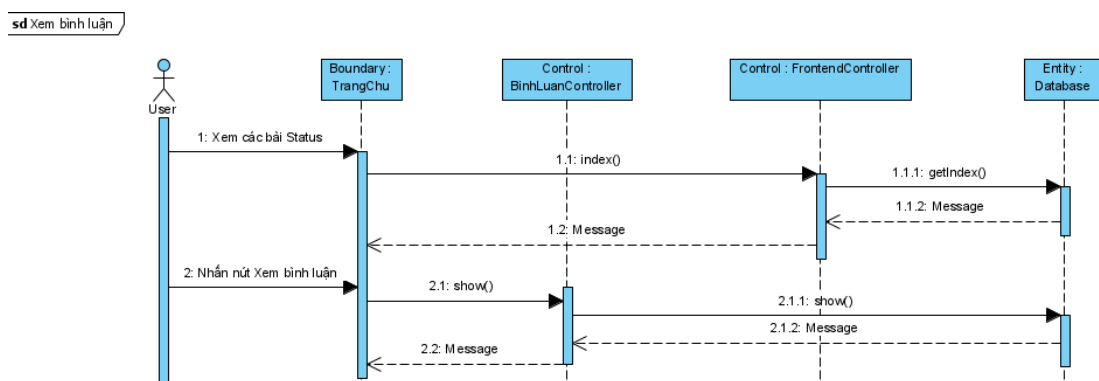
Hình 6-27: Sơ đồ usecase chức năng bình luận

Tên Use case	Bình luận
Actor	User
Mô tả	Actor có thể bình luận vào các bài status của bạn bè hoặc của chính mình. Ngoài ra có thể xem các phản hồi bình luận từ các actor khác vào bài status của mình.
Pre-conditions	Đăng nhập
Post-conditions	Thành công: Hiển thị tất cả các bình luận của status Không thành công: Thông báo lỗi
Luồng sự kiện chính	Actor bình luận vào các bài status. - Extend use case Xem bình luận - Extend use case Phản hồi bình luận
Luồng sự kiện phụ	Actor đăng xuất hoặc dùng chức năng khác trong hệ thống

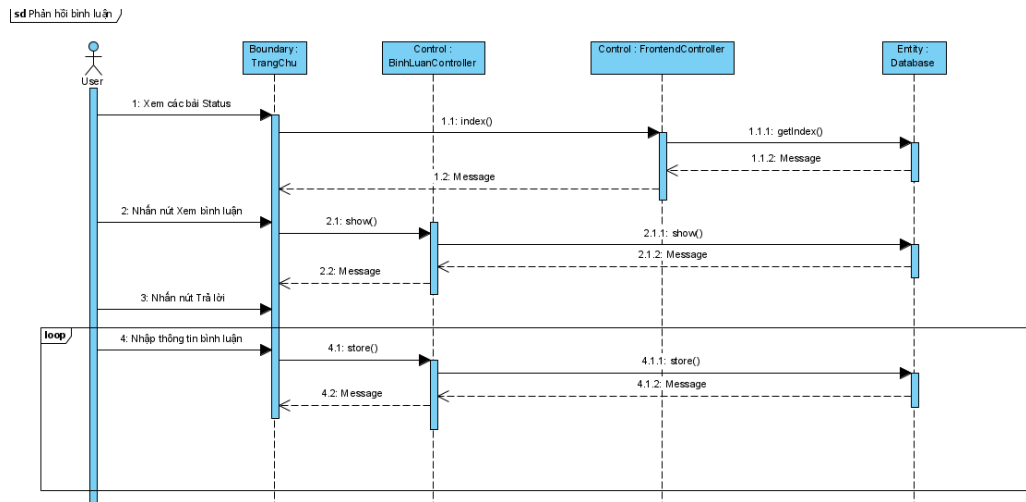
<< Extend use case Xem bình luận >>	Xem bình luận <ol style="list-style-type: none"> 1. Hệ thống hiển thị các bài status lên trang chủ 2. Actor click Xem bình luận 3. Hệ thống hiển thị tất cả bình luận có trong bài status
<< Extend use case Phản hồi bình luận >>	Phản hồi bình luận <ol style="list-style-type: none"> 1. Actor chọn đăng ký tài khoản 2. Hệ thống sẽ hiển thị trang đăng ký 3. Actor điền thông tin đăng ký vào biểu mẫu 4. Actor nhấn đăng ký 5. Hệ thống kiểm tra thông tin hợp lệ. 6. Hệ thống kiểm tra email đã sử dụng hay chưa 7. Hệ thống lưu thông tin và gửi đường dẫn xác nhận email cho người dung 8. Actor bấm vào đường dẫn xác nhận 9. Chuyển sang trang cá nhân của actor Rẽ nhánh 1: 5.1. Kiểm tra thông tin không hợp lệ thì xuất thông báo lỗi và lặp lại bước 3 Rẽ nhánh 2: 6.1. Kiểm tra email đã tồn tại và lặp lại bước 3

Bảng 6-9: Bảng mô tả usecase Bình luận

6.9.2 Sơ đồ tuần tự chức năng quản lý đăng nhập



Hình 6-28: Sơ đồ tuần tự chức năng Xem bình luận

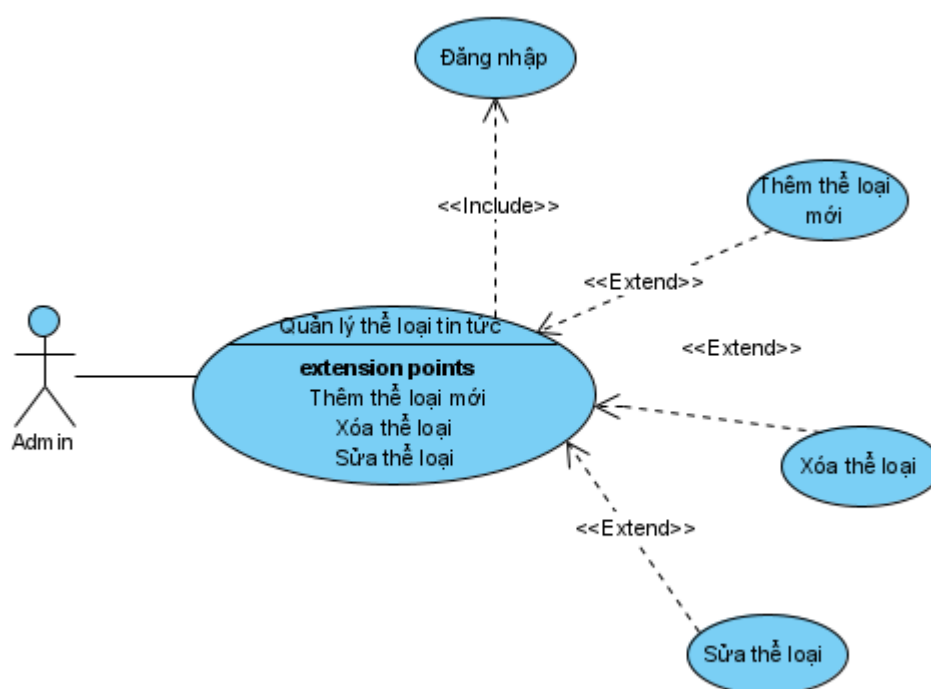


Hình 6-29: Sơ đồ tuần tự chức năng Phản hồi bình luận

Chương 7. THIẾT KẾ CHỨC NĂNG MODULE ADMIN

7.1 CHỨC NĂNG QUẢN LÝ THẺ LOẠI BLOG

7.1.1 Sơ đồ use case quản lý thẻ loại blog



Hình 7-1: Sơ đồ use case chức năng quản lý thẻ loại blog

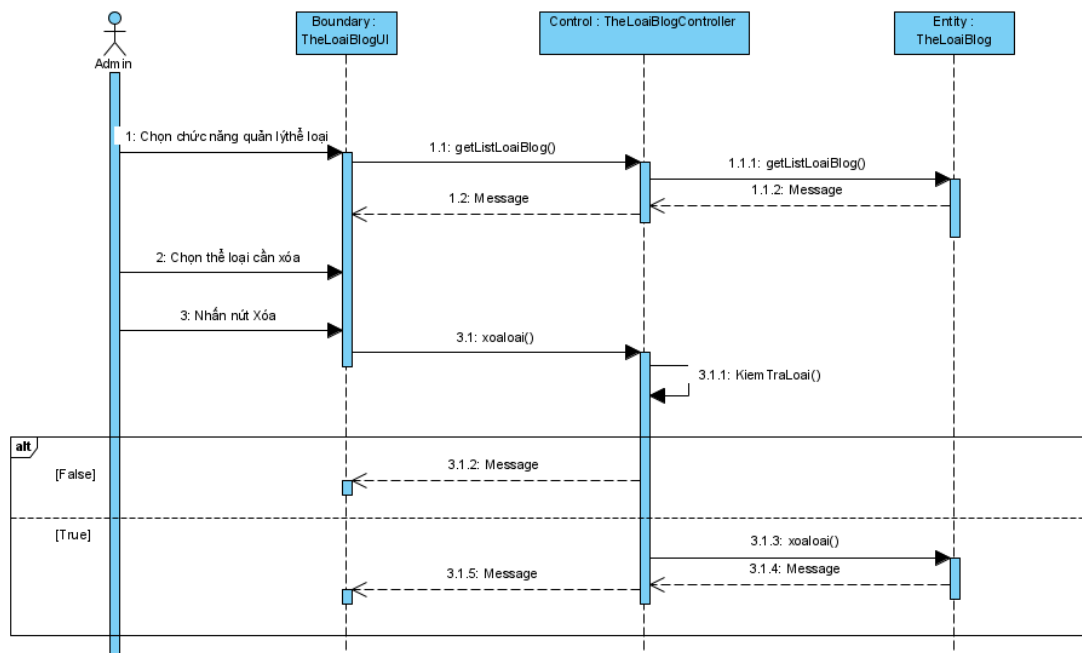
Tên Use case	Quản lý thẻ loại blog
Actor	Admin
Mô tả	Admin có thể thêm, xóa, cập nhật thông tin thẻ loại blog.

Pre-conditions	Đăng nhập
Post-conditions	Thành công: Danh sách được cập nhật Không thành công: Thông báo lỗi
Luồng sự kiện chính	Actor chọn chức năng Quản lý thể loại blog Hệ thống mở trang quản lý thể loại blog <ul style="list-style-type: none"> - Include use case Đăng nhập - Extend use case Thêm thể loại mới - Extend use case Xóa thể loại - Extend use case Sửa thể loại
Luồng sự kiện phụ	
<< Extend use case Thêm thể loại mới >>	Thêm thể loại mới <ol style="list-style-type: none"> 1. Actor nhập thông tin 2. Actor click Thêm 3. Hệ thống kiểm tra thông tin hợp lệ 4. Cập nhật CSDL 5. Hiển thị danh sách cập nhật Rẽ nhánh 1: 3.1, Kiểm tra thông tin không hợp lệ thì xuất thông báo lỗi và lặp lại bước 1
<< Extend use case Xóa thể loại >>	Xóa thể loại <ol style="list-style-type: none"> 1. Actor chọn thể loại muốn xóa 2. Actor click Xóa 3. Cập nhật CSDL 4. Hiển thị danh sách cập nhật Rẽ nhánh 1: 3.1, Actor không đồng ý xóa thì quay về trang quản lý thể loại
<< Extend use case Sửa thể loại >>	Sửa thể loại <ol style="list-style-type: none"> 1. Actor chọn thể loại muốn sửa 2. Hệ thống load thông tin thể loại được chọn 3. Actor nhập thông tin cần sửa 4. Hệ thống kiểm tra thông tin hợp lệ 5. Cập nhật CSDL 6. Hiển thị thông tin cập nhật Rẽ nhánh 1: 5.1, Hệ thống kiểm tra thông tin không hợp lệ thì xuất thông báo lỗi và lặp lại bước 3

Bảng 7-1: Bảng mô tả usecase quản lý thể loại blog

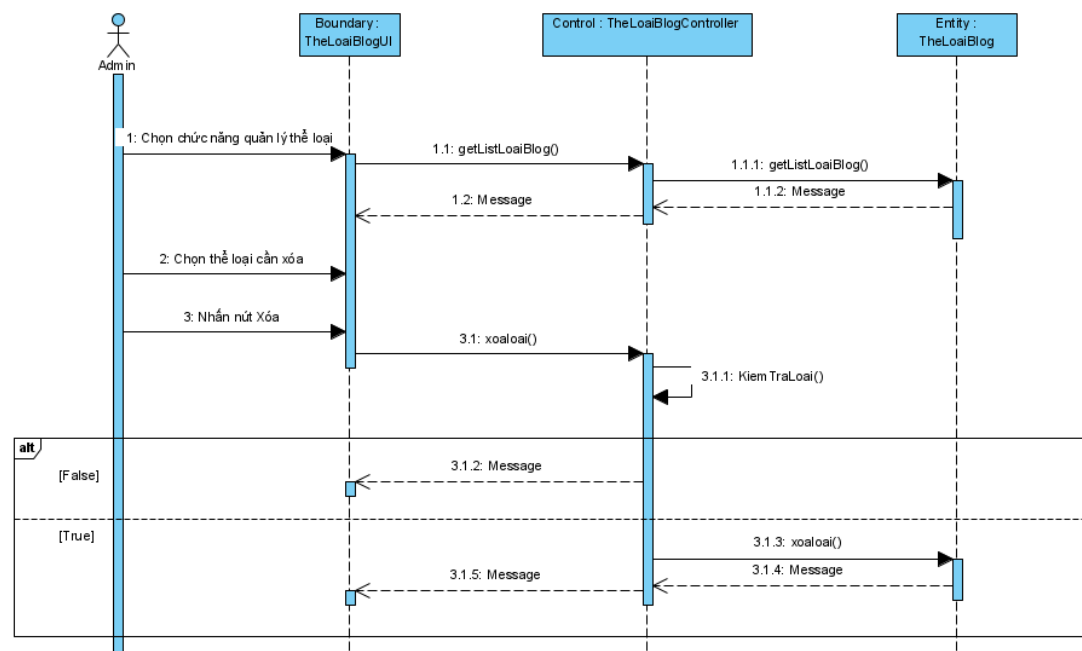
7.1.2 Sơ đồ tuần tự quản lý thể loại blog

sd Xóa thể loại Blog



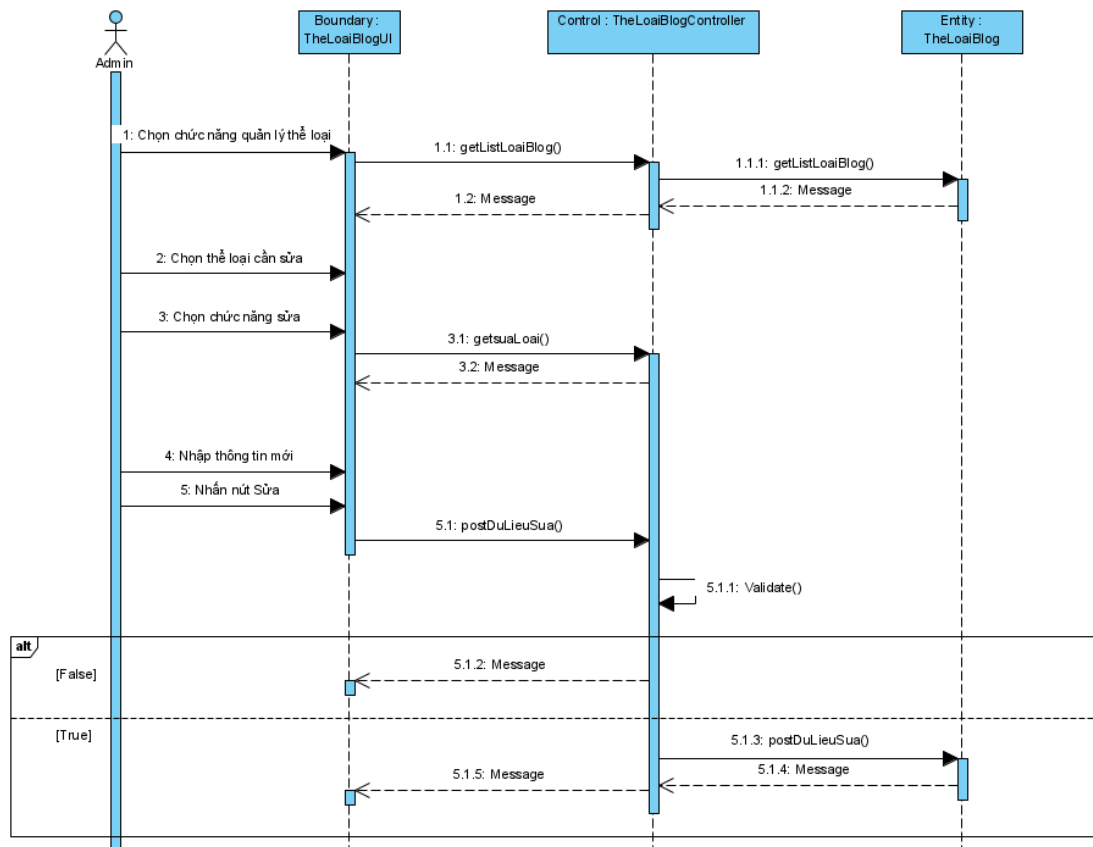
Hình 7-2: Sơ đồ tuần tự chức năng Thêm thể loại blog

sd Xóa thể loại Blog



Hình 7-3: Sơ đồ tuần tự chức năng Xóa thể loại Blog

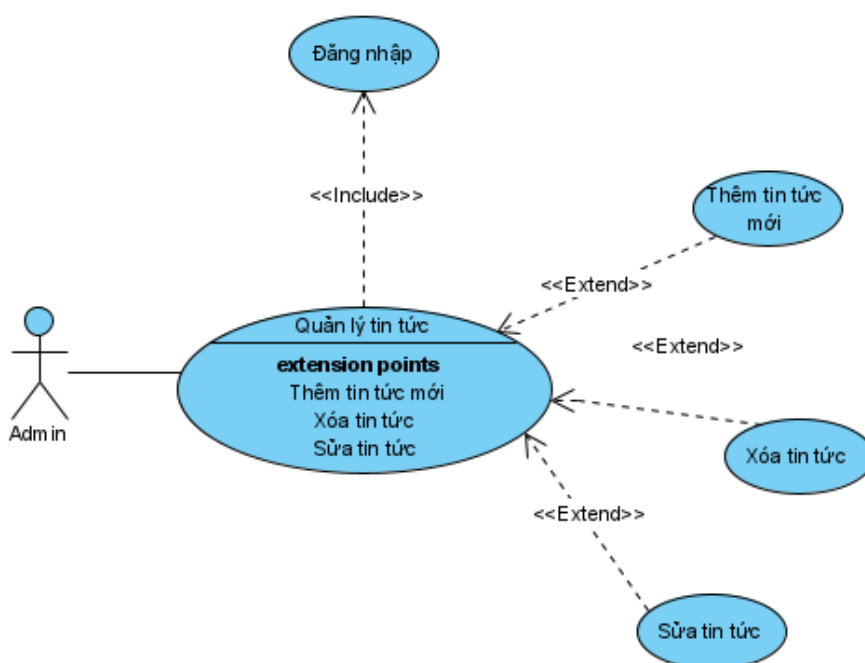
sd Sửa thể loại Blog



Hình 7-4: Sơ đồ tuần tự chức năng Sửa thể loại Blog

7.2 CHỨC NĂNG QUẢN LÝ BLOG

7.2.1 Sơ đồ use case chức năng quản lý blog



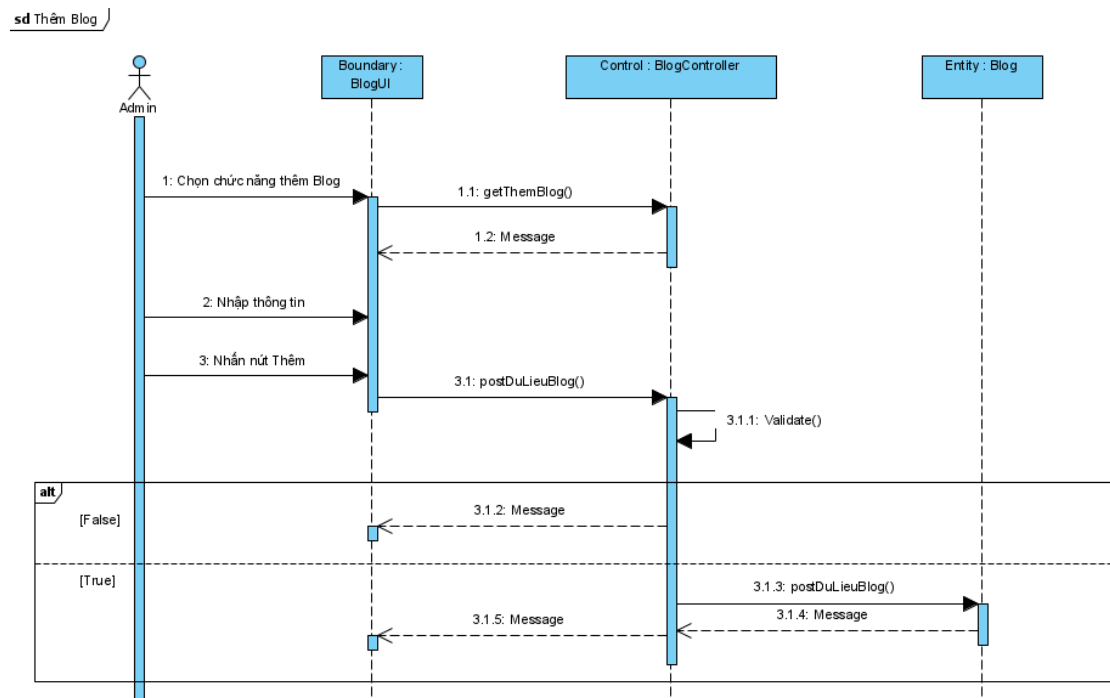
Hình 7-5: Sơ đồ use case quản lý blog

Tên Use case	Quản lý blog
Actor	Admin
Mô tả	Admin có thể thêm, xóa, cập nhật thông tin blog.
Pre-conditions	Đăng nhập
Post-conditions	Thành công: Danh sách được cập nhật Không thành công: Thông báo lỗi
Luồng sự kiện chính	Actor chọn chức năng Quản lý blog Hệ thống mở trang quản lý blog

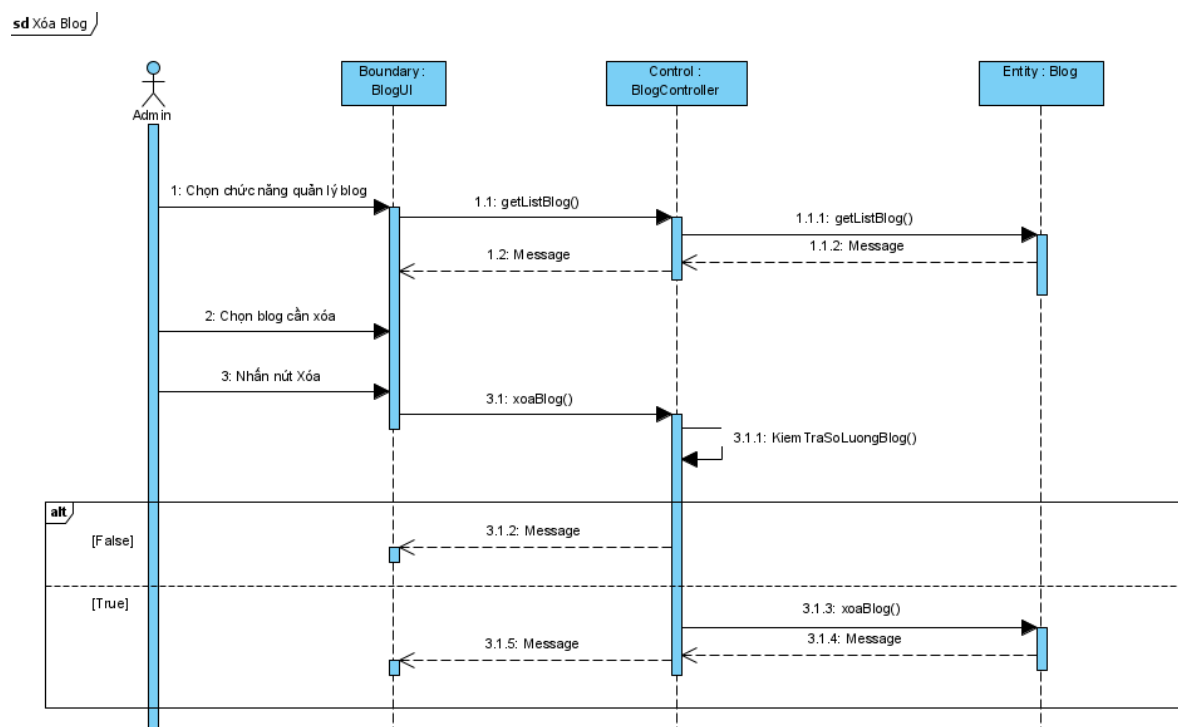
	<ul style="list-style-type: none"> - Include use case Đăng nhập - Extend use case Thêm blog mới - Extend use case Xóa blog - Extend use case Sửa blog
Luồng sự kiện phụ	
<< Extend use case Thêm blog mới >>	<p>Thêm blog mới</p> <ol style="list-style-type: none"> 1. Actor nhập thông tin blog mới vào form 2. Actor click Thêm 3. Hệ thống kiểm tra thông tin hợp lệ 4. Cập nhật CSDL 5. Hiển thị danh sách cập nhật <p>Rẽ nhánh 1: 4.1, Kiểm tra thông tin không hợp lệ thì xuất thông báo lỗi và lặp lại bước 1</p>
<< Extend use case Xóa blog >>	<p>Xóa blog</p> <ol style="list-style-type: none"> 1. Actor chọn blog muốn xóa 2. Actor click Xóa 3. Cập nhật CSDL 4. Hiển thị danh sách cập nhật <p>Rẽ nhánh 1: 3.1, Hệ thống kiểm tra số lượng blog còn lại nếu còn duy nhất 01 bài blog sẽ không xóa về trang quản lý blog</p>
<< Extend use case Sửa blog >>	<p>Sửa blog</p> <ol style="list-style-type: none"> 1. Actor chọn blog muốn sửa 2. Hệ thống load thông tin blog được chọn 3. Actor nhập thông tin cần sửa 4. Hệ thống kiểm tra thông tin hợp lệ 5. Cập nhật CSDL 6. Hiển thị thông tin cập nhật <p>Rẽ nhánh 1: 5.1, Hệ thống kiểm tra thông tin không hợp lệ thì xuất thông báo lỗi và lặp lại bước 3</p>

Bảng 7-2: Bảng mô tả usecase quản lý Blog

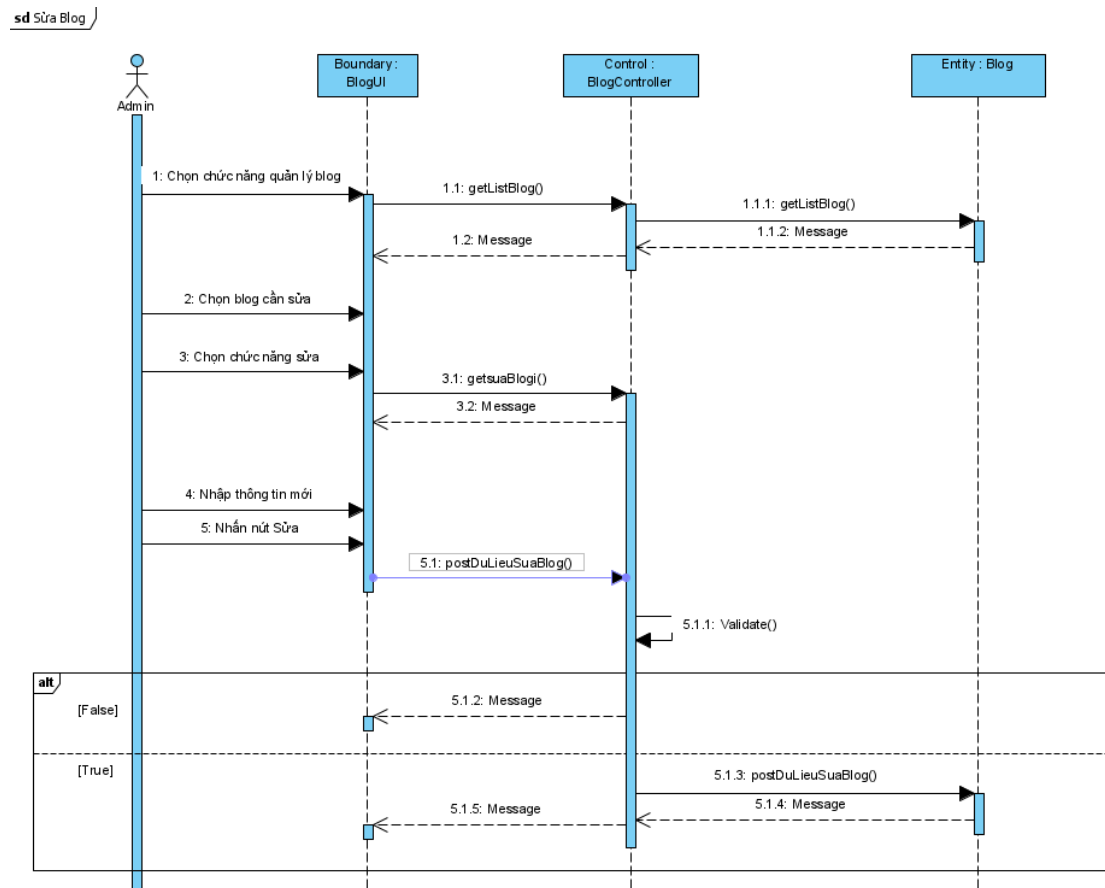
7.2.2 Sơ đồ tuần tự chức năng quản lý blog



Hình 7-6: Sơ đồ tuần tự chức năng Thêm Blog



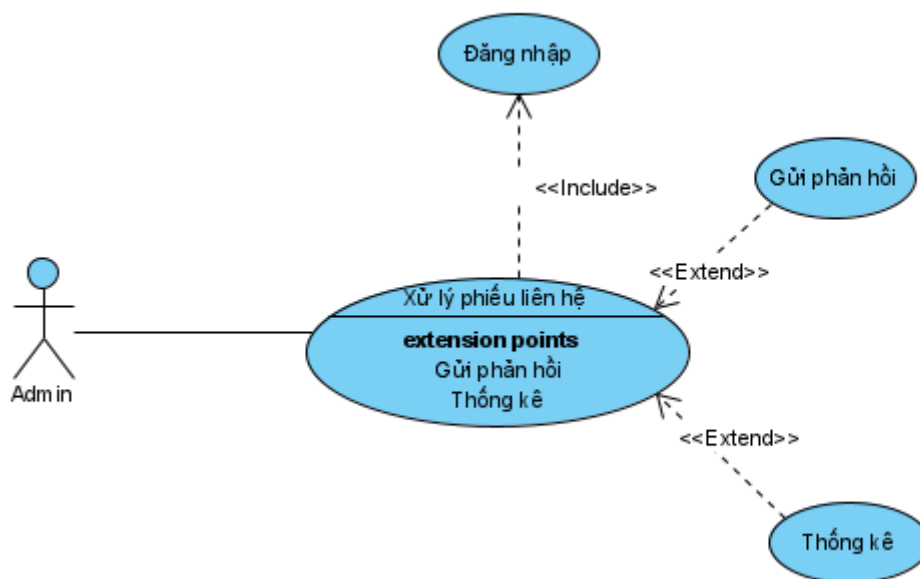
Hình 7-7: Sơ đồ tuần tự chức năng Xóa Blog



Hình 7-8: Sơ đồ tuần tự chức năng Sửa Blog

7.3 CHỨC NĂNG XỬ LÝ PHIẾU LIÊN HỆ

7.3.1 Sơ đồ use case chức năng xử lý phiếu liên hệ



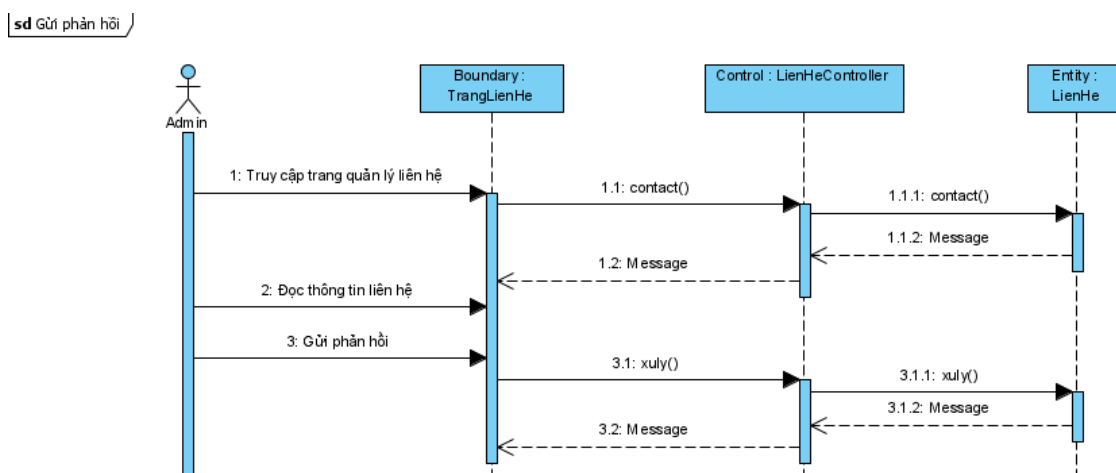
Hình 7-9: Sơ đồ use case xử lý phiếu liên hệ

Tên Use case	Xử lý phiếu liên hệ
Actor	Admin
Mô tả	Admin có thể gửi phản hồi liên hệ và thống kê số lượng liên hệ
Pre-conditions	Đăng nhập
Post-conditions	Thành công: Danh sách được cập nhật Không thành công: Thông báo lỗi
Luồng sự kiện chính	Actor chọn chức năng Quản lý Liên hệ Hệ thống mở trang quản lý liên hệ <ul style="list-style-type: none"> - Include use case Đăng nhập - Extend use case Gửi phản hồi - Extend use case Thống kê
Luồng sự kiện phụ	

<< Extend use case Gửi phản hồi >>	Gửi phản hồi <ol style="list-style-type: none"> 1. Hệ thống hiển thị thông tin liên hệ 2. Actor đọc thông tin liên hệ 3. Actor click Gửi phản hồi 4. Hệ thống gửi email phản hồi tới người dùng 5. Cập nhật CSDL 6. Hiển thị danh sách cập nhật
<< Extend use case Thống kê >>	Thống kê <ol style="list-style-type: none"> 1. Actor đăng nhập vào trang quản trị 2. Hiển thị số lượng liên hệ được gửi đến trong ngày

Bảng 7-3: Bảng mô tả usecase xử lý phiếu liên hệ

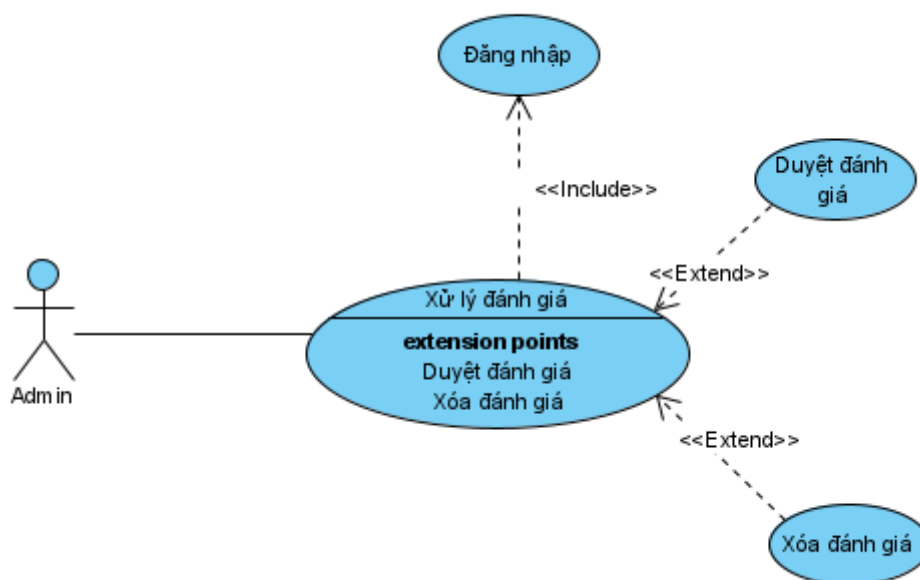
7.3.2 Sơ đồ tuần tự chức năng xử lý phiếu liên hệ



Hình 7-10: Sơ đồ tuần tự chức năng Gửi phản hồi

7.4 CHỨC NĂNG XỬ LÝ ĐÁNH GIÁ

7.4.1 Sơ đồ use case chức năng xử lý đánh giá



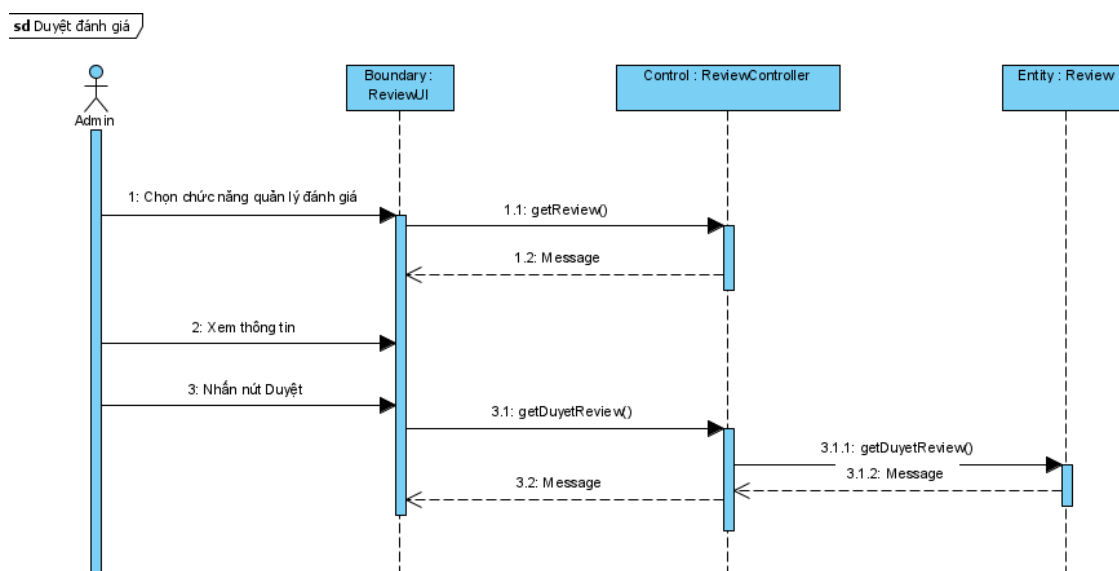
Hình 7-11: Sơ đồ use case xử lý đánh giá

Tên Use case	Xử lý đánh giá
Actor	Admin
Mô tả	Admin có thể duyệt hoặc xóa các đánh giá của người dùng về trang web được gửi tới admin.
Pre-conditions	Đăng nhập
Post-conditions	Thành công: Danh sách được cập nhật Không thành công: Thông báo lỗi
Luồng sự kiện chính	Actor chọn chức năng Đánh giá Hệ thống mở trang quản lý đánh giá <ul style="list-style-type: none"> - Include use case Đăng nhập - Extend use case Duyệt đánh giá - Extend use case Xóa đánh giá
Luồng sự kiện phụ	

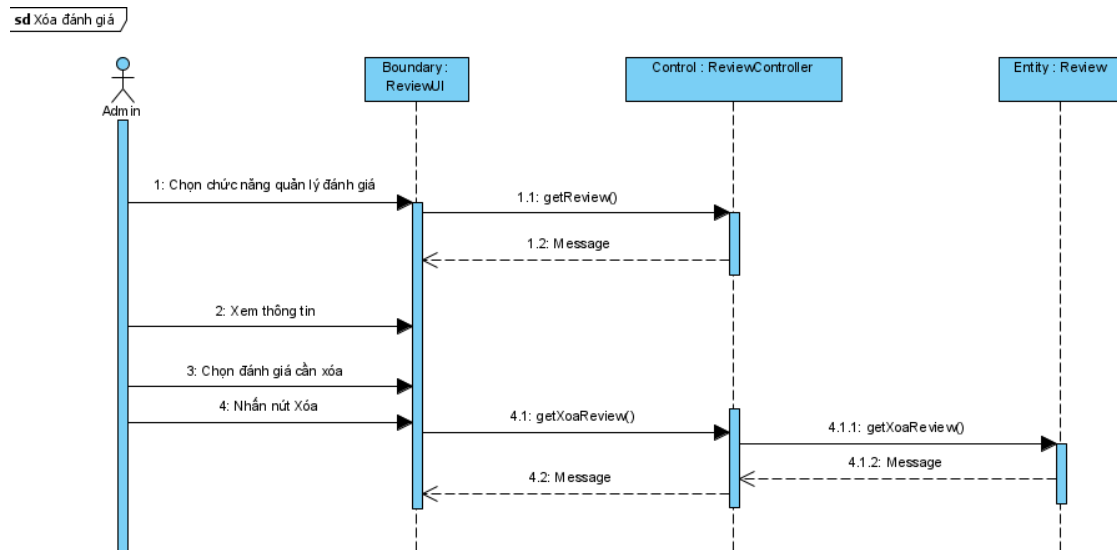
<< Extend use case Duyệt đánh giá >>	Duyệt đánh giá <ol style="list-style-type: none"> 1. Actor xem thông tin đánh giá 2. Actor click Duyệt 3. Cập nhật CSDL 4. Hiện thị danh sách cập nhật
<< Extend use case Xóa đánh giá >>	Xóa đánh giá <ol style="list-style-type: none"> 3. Actor chọn đánh giá muốn xóa 4. Actor click Xóa 5. Hiện thị thông báo xác nhận xóa 6. Actor xác nhận xóa 7. Cập nhật CSDL 8. Hiện thị danh sách cập nhật <p>Rẽ nhánh 1: 4.1, Actor không đồng ý xóa thì quay về trang quản lý đánh giá</p>

Bảng 7-4: Bảng mô tả use case xử lý đánh giá

7.4.2 Sơ đồ tuần tự chức năng xử lý đánh giá



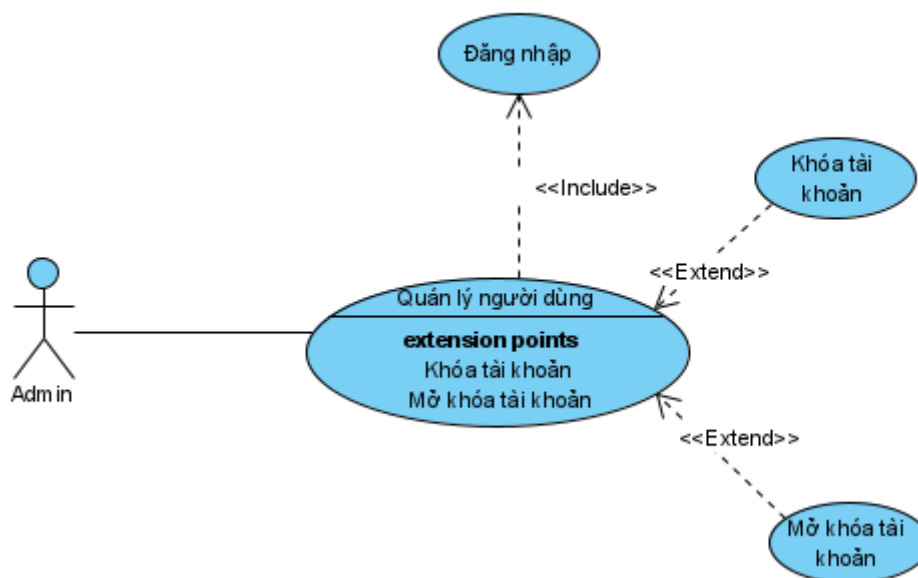
Hình 7-12: Sơ đồ tuần tự chức năng Duyệt đánh giá



Hình 7-13: Sơ đồ tuần tự chức năng Xóa đánh giá

7.5 CHỨC NĂNG QUẢN LÝ NGƯỜI DÙNG

7.5.1 Sơ đồ use case chức năng quản lý người dùng



Hình 7-14: Sơ đồ use case chức năng quản lý người dùng

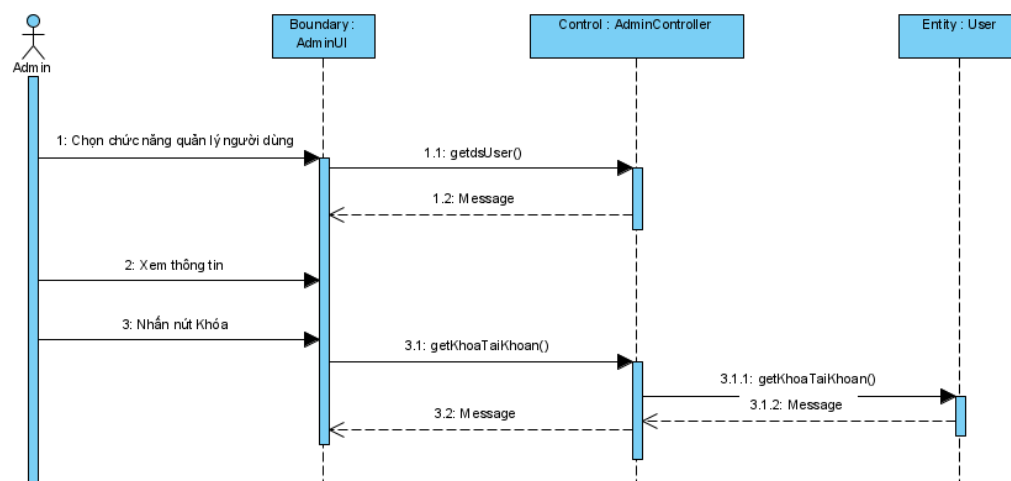
Tên Use case	Quản lý người dùng
Actor	Admin
Mô tả	Admin có thể khóa/ mở khóa tài khoản của người dùng.
Pre-conditions	Đăng nhập
Post-conditions	Thành công: Danh sách được cập nhật Không thành công: Thông báo lỗi
Luồng sự kiện chính	Actor chọn chức năng Quản lý người dùng Hệ thống mở trang quản lý người dùng <ul style="list-style-type: none"> - Include use case Đăng nhập - Extend use case Khóa tài khoản - Extend use case Mở khóa tài khoản
Luồng sự kiện phụ	

<< Extend use case Khóa tài khoản >>	Khóa tài khoản <ol style="list-style-type: none"> 1. Actor kiểm tra danh sách người dùng 2. Actor click Khóa tài khoản 3. Hiện thị thông báo xác nhận Khóa 4. Actor xác nhận Khóa 5. Cập nhật CSDL 6. Hiện thị danh sách cập nhật
<< Extend use case Mở khóa tài khoản >>	Mở khóa tài khoản <ol style="list-style-type: none"> 1. Actor kiểm tra danh sách người dùng 2. Actor click Mở khóa tài khoản đang bị khóa 3. Hiện thị thông báo xác nhận Mở khóa 4. Actor xác nhận Mở khóa 5. Cập nhật CSDL 6. Hiện thị danh sách cập nhật Rẽ nhánh 1: 4.1, Actor không đồng ý xóa thì quay về trang quản lý người dùng

Bảng 7-5: Bảng mô tả use case quản lý người dùng

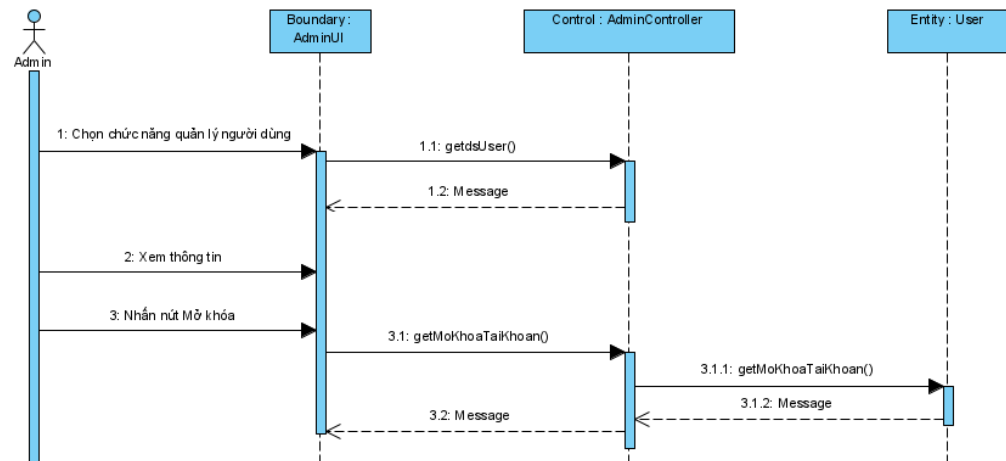
7.5.2 Sơ đồ tuần tự chức năng quản lý người dùng

sd khóa tài khoản



Hình 7-15: Sơ đồ tuần tự chức năng Khóa tài khoản

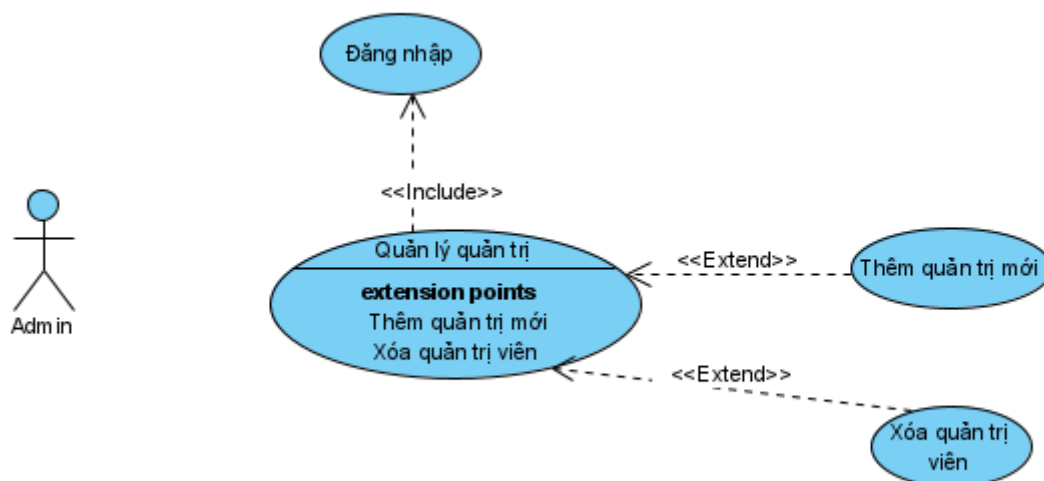
sd Mở khóa tài khoản



Hình 7-16: Sơ đồ tuần tự chức năng Mở khóa tài khoản

7.6 CHỨC NĂNG QUẢN LÝ QUẢN TRỊ

7.6.1 Sơ đồ use case quản lý quản trị



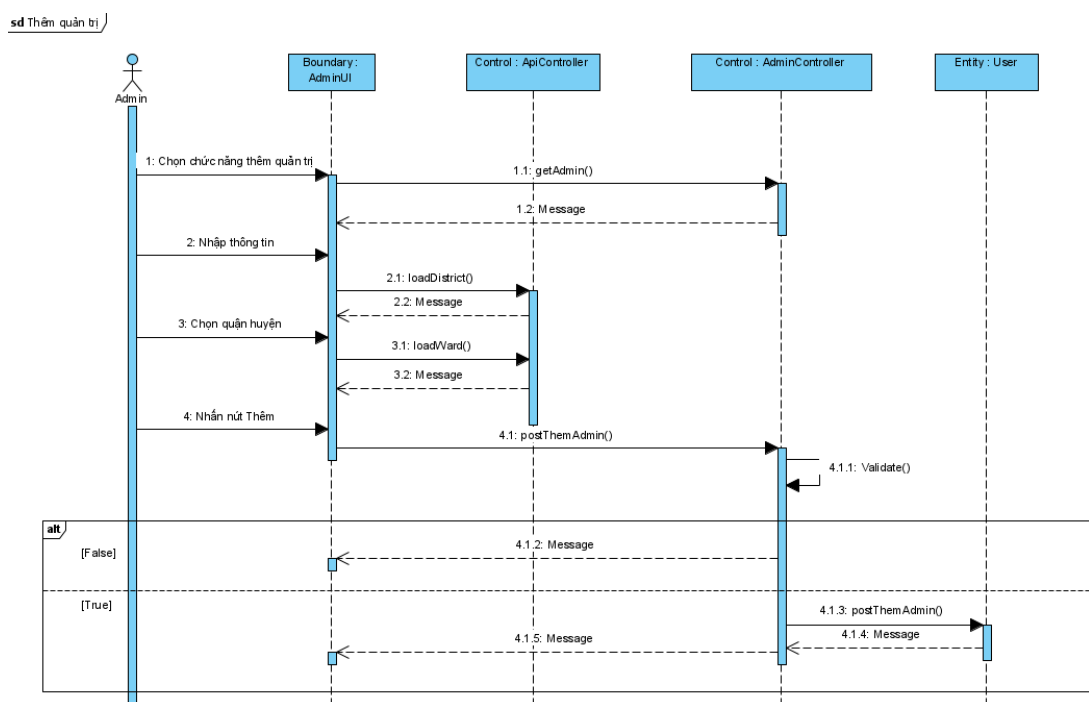
Hình 7-17: Sơ đồ use case chức năng quản lý quản trị

Tên Use case	Quản lý thể loại tin tức
Actor	Admin
Mô tả	Admin có thể thêm, xóa, các quản trị viên khác.
Pre-conditions	Đăng nhập
Post-conditions	Thành công: Danh sách được cập nhật Không thành công: Thông báo lỗi
Luồng sự kiện chính	Actor chọn chức năng Quản lý quản trị Hệ thống mở trang quản lý quản trị <ul style="list-style-type: none"> - Include use case Đăng nhập - Extend use case Thêm quản trị mới - Extend use case Xóa quản trị viên -
Luồng sự kiện phụ	

<p><< Extend use case Thêm quản trị mới >></p>	<p>Thêm quản trị mới</p> <ol style="list-style-type: none"> 1. Actor nhập thông tin của quản trị viên mới 2. Actor click Thêm 3. Hệ thống kiểm tra thông tin hợp lệ 4. Cập nhật CSDL 5. Hiện thị danh sách cập nhật <p>Rẽ nhánh 1: 3.1, Kiểm tra thông tin không hợp lệ thì xuất thông báo lỗi và lặp lại bước 1</p>
<p><< Extend use case Xóa quản trị >></p>	<p>Xóa quản trị</p> <ol style="list-style-type: none"> 1. Actor chọn quản trị muốn xóa ra khỏi hệ thống 2. Actor click Xóa 3. Hiện thị thông báo xác nhận xóa 4. Actor xác nhận xóa 5. Cập nhật CSDL 6. Hiện thị danh sách cập nhật <p>Rẽ nhánh 1: 4.1, Actor không đồng ý xóa thì quay về trang quản lý quản trị</p>

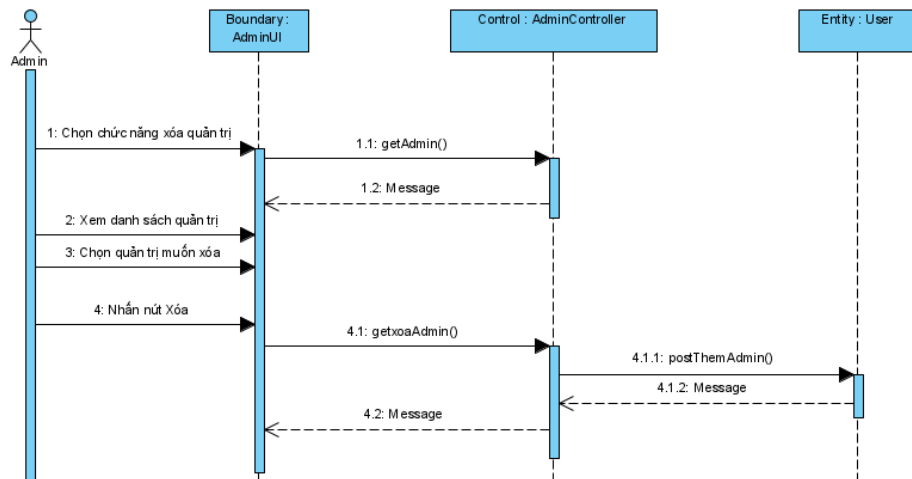
Bảng 7-6: Bảng mô tả usecase quản trị

7.6.2 Sơ đồ tuần tự quản lý quản trị



Hình 7-18: Sơ đồ tuần tự chức năng Thêm quản trị

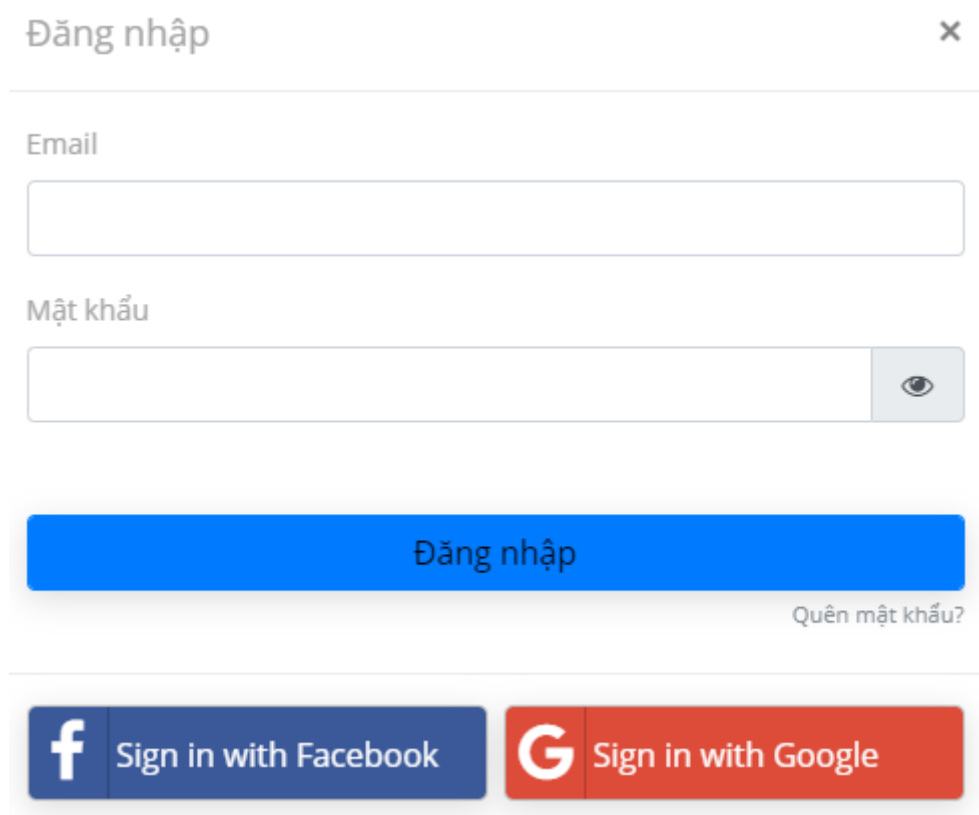
sd Xóa quản trị /



Hình 7-19: Sơ đồ tuần tự chức năng Xóa quản trị

Chương 8. THÀNH PHẦN GIAO DIỆN

8.1 GIAO DIỆN ĐĂNG NHẬP

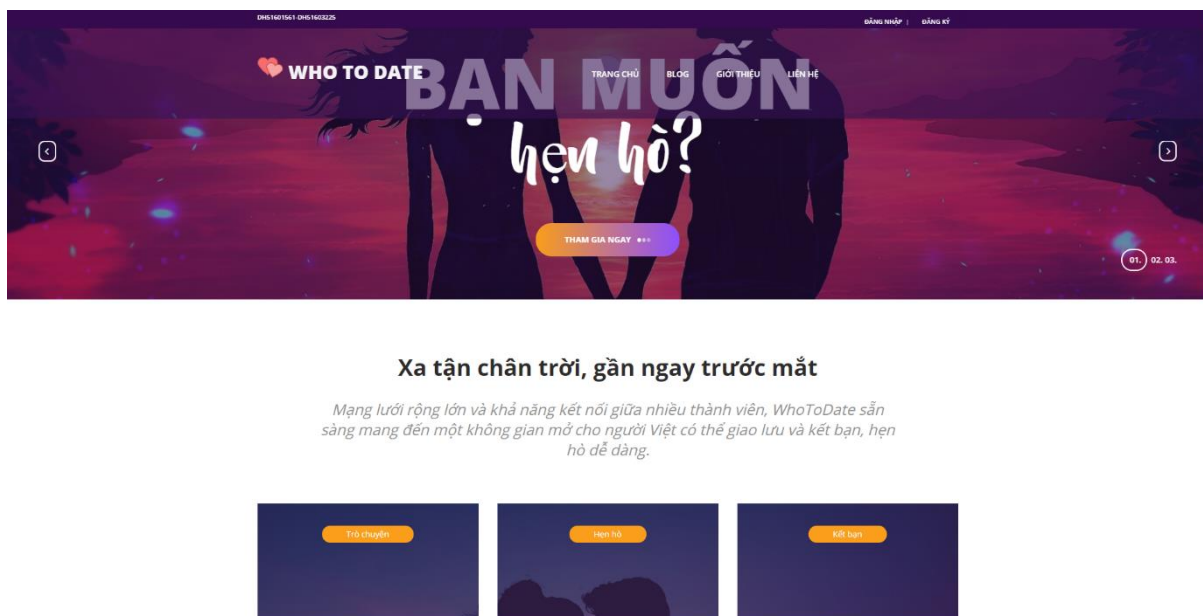
The image shows a login form with a title bar 'Đăng nhập' and a close button 'x'. It contains two input fields: 'Email' and 'Mật khẩu' (Password). The password field has a toggle button with an eye icon. Below the fields is a blue 'Đăng nhập' button and a link 'Quên mật khẩu?'. At the bottom, there are two social login buttons: 'Sign in with Facebook' and 'Sign in with Google'.

Hình 8-1: Giao diện đăng nhập

Mô tả: Người dùng đăng nhập thông qua giao diện trên, bằng cách nhập email đã được xác nhận và mật khẩu đã được đăng ký. Sau đó nhấn nút Đăng nhập để truy cập vào trang cá nhân của mình.

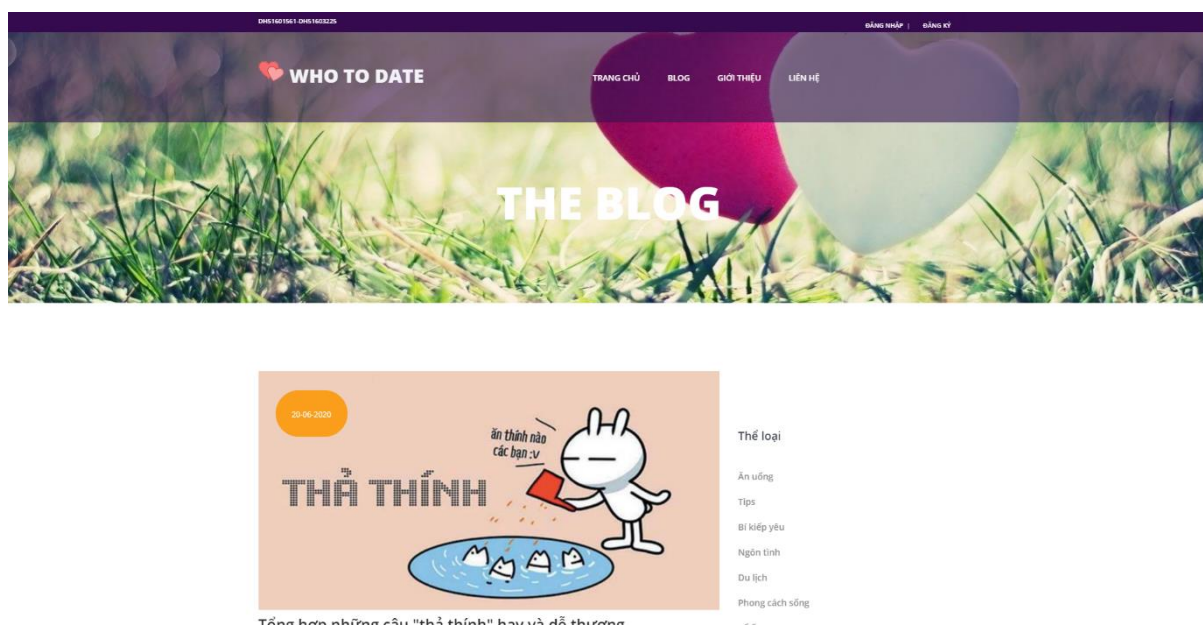
Người dùng có thể đăng nhập bằng nhiều hình thức, hệ thống cung cấp thêm đăng nhập bằng Facebook và Goolge. Ngoài ra nếu người dùng quên mật khẩu có thể tìm chức năng Quên mật khẩu ở trong giao diện này.

8.2 GIAO DIỆN TRANG CHỦ



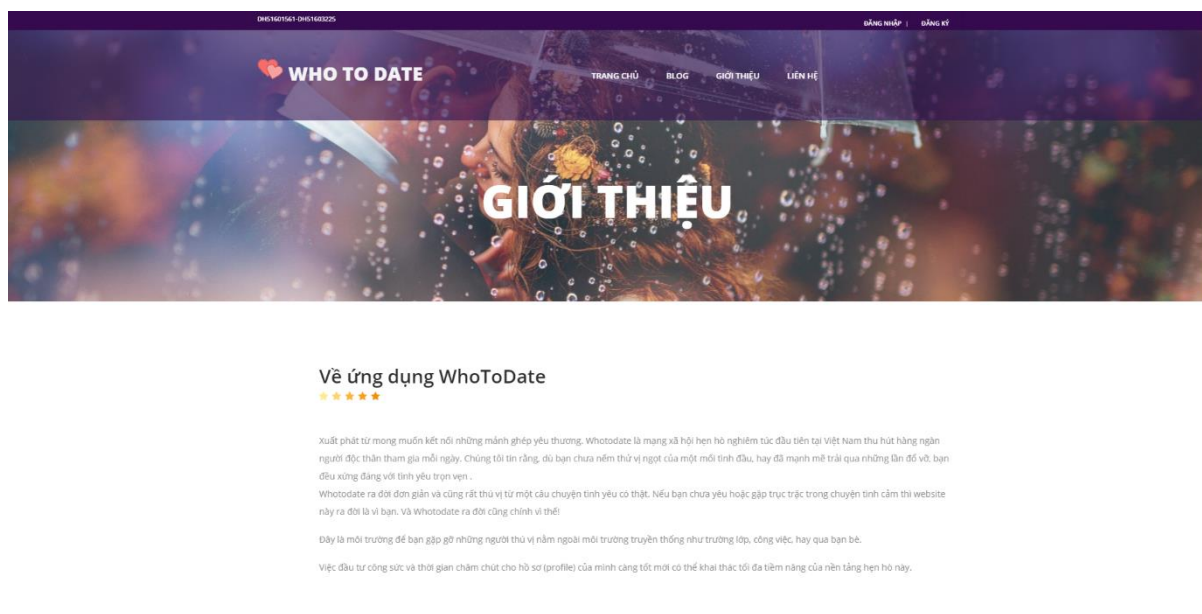
Hình 8-2: Giao diện trang chủ

Mô tả: Đây là giao diện đầu tiên khi người dùng truy cập hệ thống.



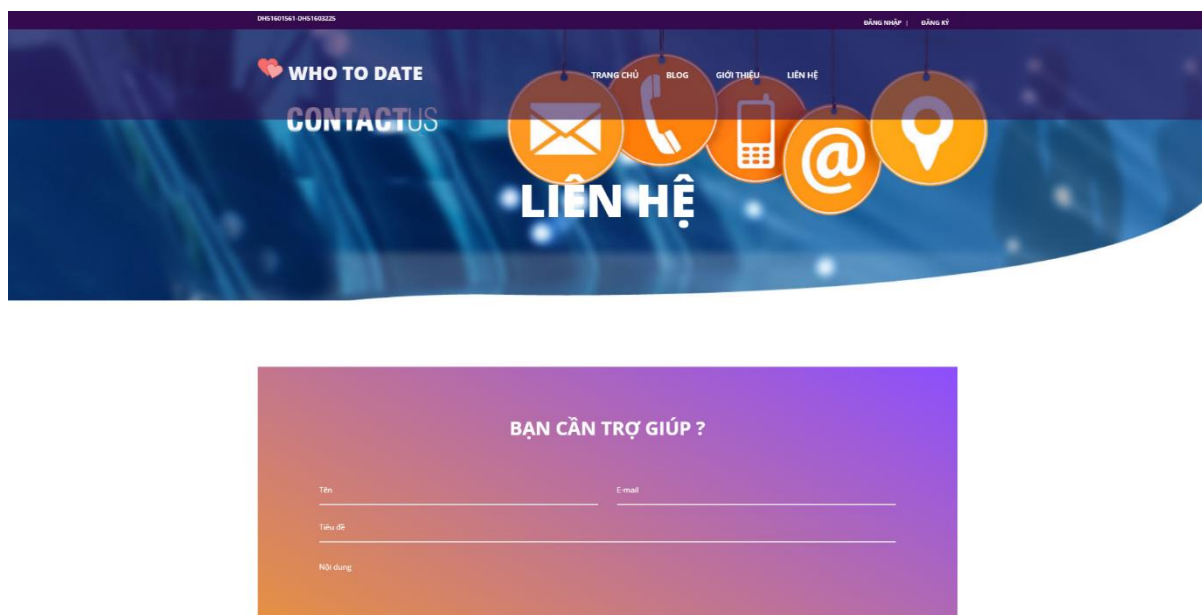
Hình 8-3: Giao diện trang Blog

Mô tả: Đây là giao diện để người đọc các bài Blog được đăng ở trang chủ



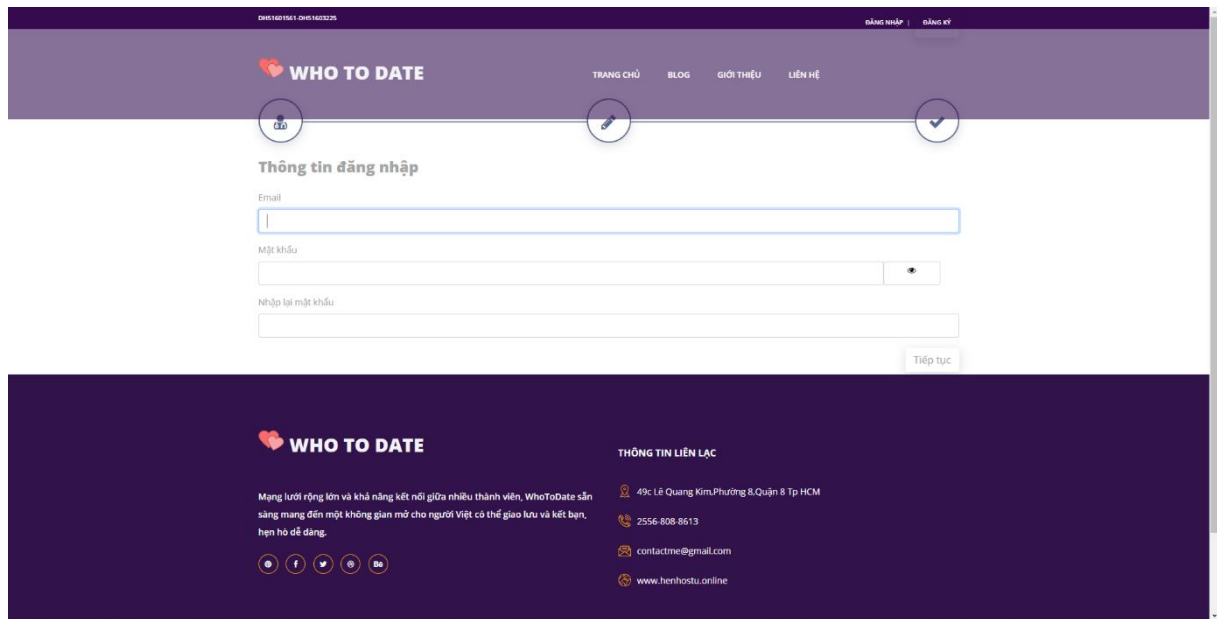
Hình 8-4: Giao diện trang Giới thiệu

Mô tả: Đây là giao diện giới thiệu tổng quan về ứng dụng web và thông tin của đội ngũ phát triển.



Hình 8-5: Giao diện Trang Liên hệ

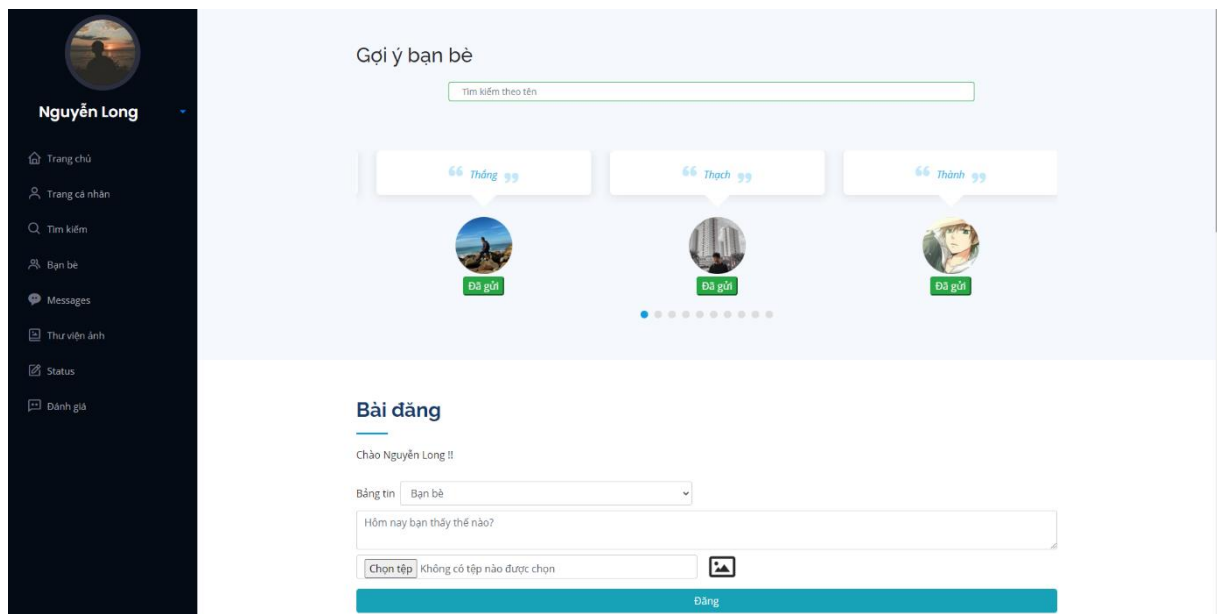
Mô tả: Đây là giao diện dùng để người dùng liên hệ với đội ngũ admin.



Hình 8-6: Giao diện Trang Đăng ký

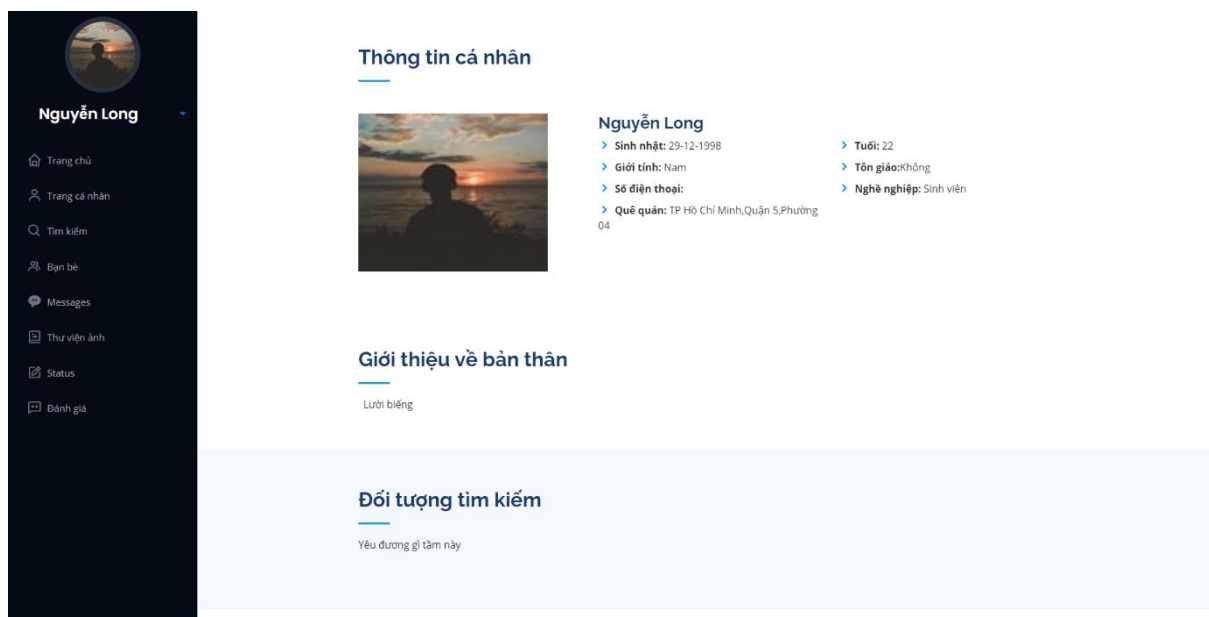
Mô tả: Đây là giao diện để người dùng thực hiện việc đăng ký tài khoản.

8.3 GIAO DIỆN USER



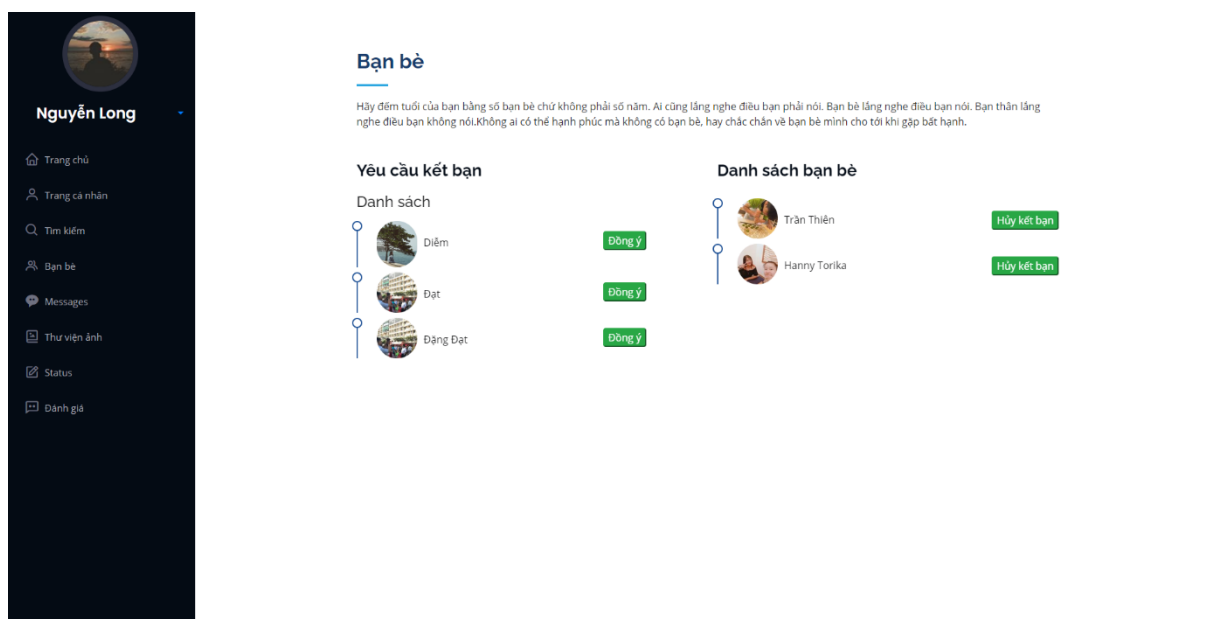
Hình 8-7: Giao diện trang chủ User

Mô tả: Đây là giao diện trang chủ sau khi đăng nhập.



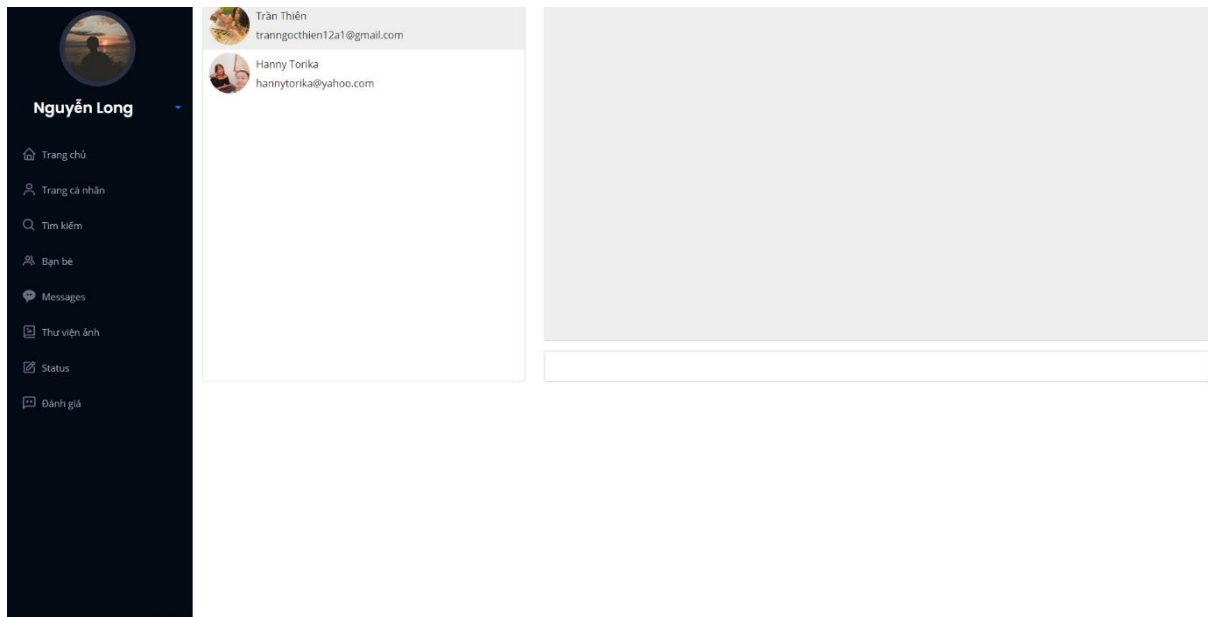
Hình 8-7: Giao diện Trang cá nhân

Mô tả: Đây là giao diện trang cá nhân của người dùng. Hiện thị các thông tin đăng ký của người dùng, cho phép người dùng chỉnh sửa thông tin cá nhân.



Hình 8-8: Giao diện Bạn bè

Mô tả: Đây là giao diện hiển thị những người dùng là bạn bè. Tại đây có hiển thị các lời mời kết bạn và chờ được chấp nhận.



Hình 8-9: Giao diện chat

Mô tả: Đây là giao diện chức năng trò chuyện với người dùng đã là bạn của mình. Cửa sổ trò chuyện sẽ hiển thị đoạn tin nhắn của hai người. Ngoài ra còn có hiển thị trạng thái hoạt động của bạn bè.

The screenshot shows a user interface for creating a post. On the left is a dark sidebar with the user's profile picture and name 'Nguyễn Long', and a list of navigation items: Trang chủ, Trang cá nhân, Tìm kiếm, Bạn bè, Messages, Thư viện ảnh, Status, and Đánh giá. The main content area is titled 'Bài đăng' (Post). Below the title, it says 'Chào Nguyễn Long !!'. There is a dropdown menu for 'Bảng tin' (Feed) with 'Bạn bè' (Friends) selected. Below this is a text input field with the placeholder 'Hôm nay bạn thấy thế nào?' (How are you feeling today?). Under the text field is a file upload section with a 'Chọn tệp' (Choose file) button, a message 'Không có tệp nào được chọn' (No files selected), and a small image icon. At the bottom of the form is a blue 'Đăng' (Post) button.

Hình 8-10: Giao diện Status

Mô tả: Đây là giao diện cho phép người dùng đăng tải nội dung lên trang chủ, trang cá nhân. Có 2 chế độ hiển thị status: Bạn bè và Chỉ mình tôi.

The screenshot shows a user interface for providing feedback about the website. On the left is the same dark sidebar as in the previous image. The main content area is titled 'Cảm nhận về Website' (Feedback about Website). Below the title, it says 'Bạn có cảm nhận và nhận xét gì sau khi sử dụng website này?' (What are your feelings and comments after using this website?). There is a large text input field with the placeholder 'Đây là vùng nhập text' (This is the text input area). To the right of the input field is a blue button labeled 'Gửi nhận xét' (Submit feedback).

Hình 8-11: Giao diện Đánh giá

Mô tả: Đây là giao diện chức năng đánh giá trải nghiệm sử dụng website của người dùng. Mỗi người dùng chỉ được đánh giá 01 lần.

Thông tin cá nhân

Họ và tên
Nguyễn Long

Ngày sinh
29/12/1998

Đại chỉ
Tiền Giang Quận Phường

Giới tính
☒ Nam ☐ Nữ

Tôn giáo
Không

SĐT

Nghề nghiệp
Sinh viên

Sở thích
Ngủ

Giới thiệu sơ lược bản thân
Lười biếng

Mẫu người bạn thích ?

Hình 8-12: Giao diện sửa thông tin cá nhân

Mô tả: Đây là giao diện chức năng cập nhật thông tin của người dùng. Người dùng có thể thêm hoặc xóa bớt thông tin để hiển thị lên trang cá nhân.

8.4 GIAO DIỆN ADMIN

Quản trị website - henhoSTU

Admin
admin@gmail.com

Trang chủ

Thể loại Blog

Blog

Liên hệ

Đánh giá

User

Quản trị

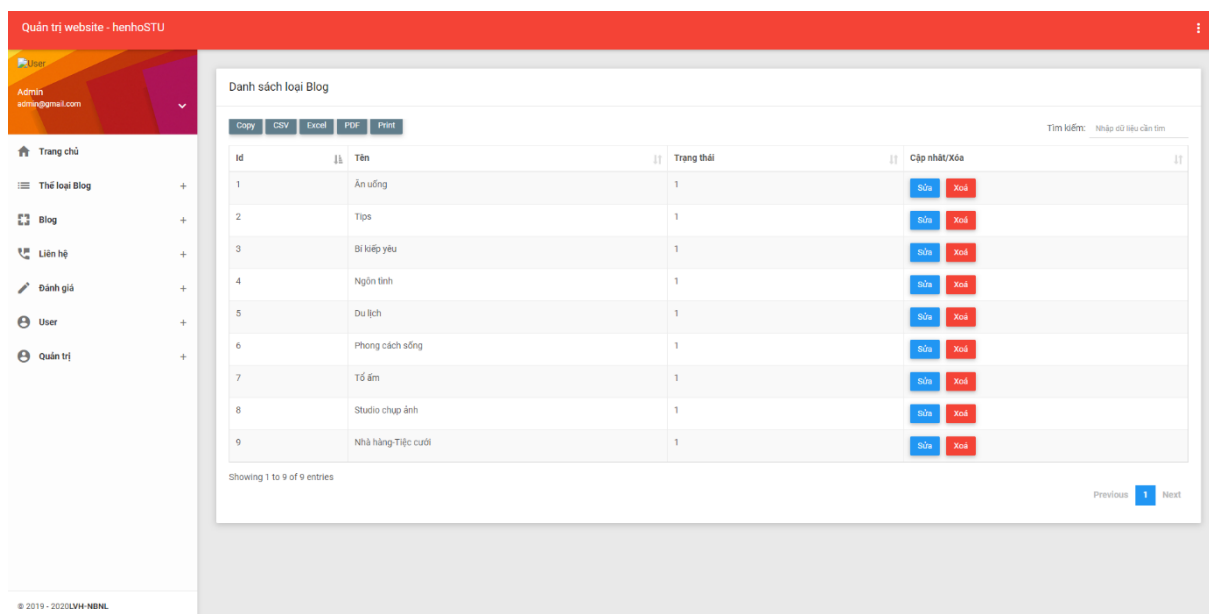
Trang chủ

Bảng thống báo hằng ngày

Ngày 09-08-2020	Thống báo
Liên hệ	0
Đánh giá	0

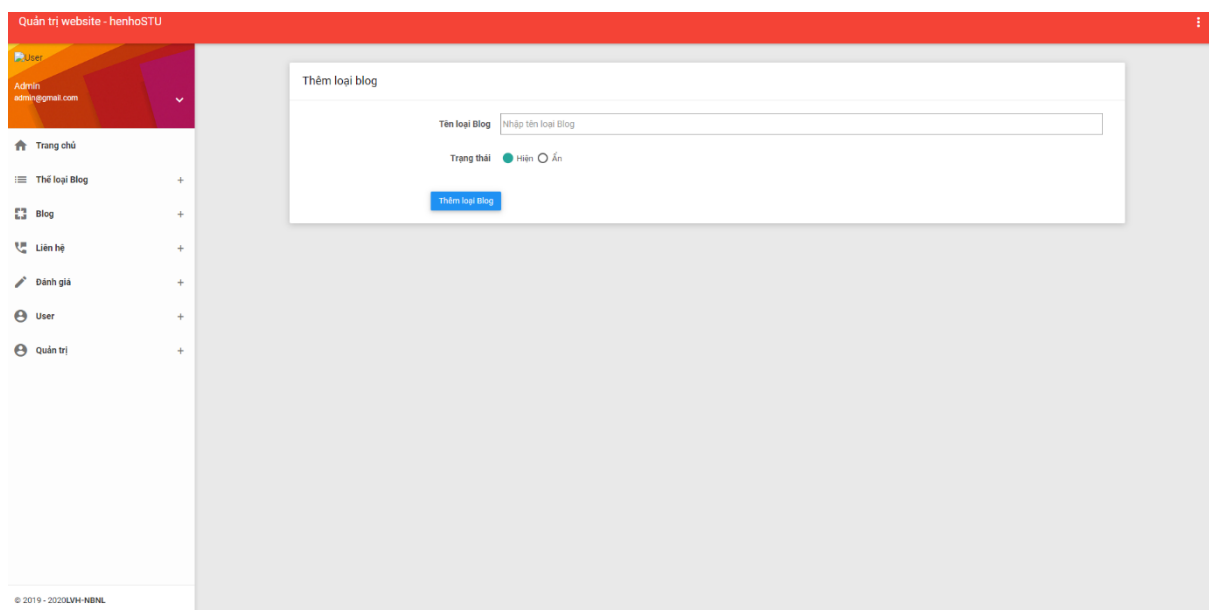
Hình 8-13 : Giao diện trang chủ Admin

Mô tả: Đây là giao diện trang chủ của quản trị viên sau khi đăng nhập bằng tài khoản quản trị.



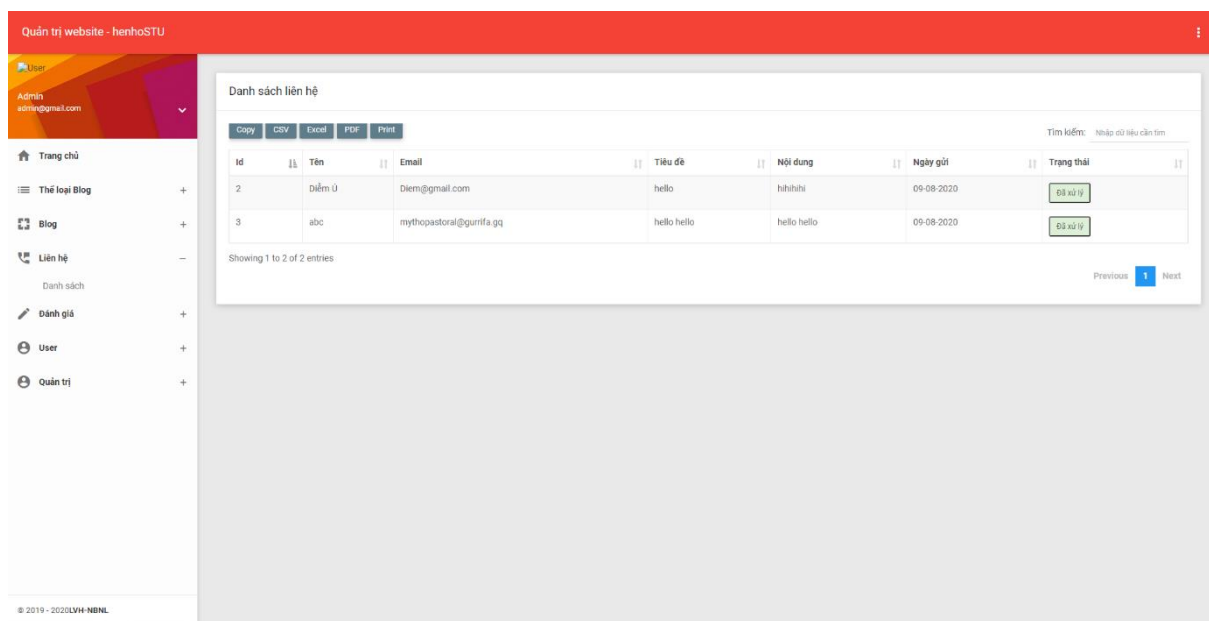
Hình 8-14 : Giao diện Quản lý thể loại blog

Mô tả: Đây là giao diện chức năng quản lý thể loại blog. Hiện thị danh sách các thể loại blog hiện hành và 2 chức năng xóa/sửa thể loại.



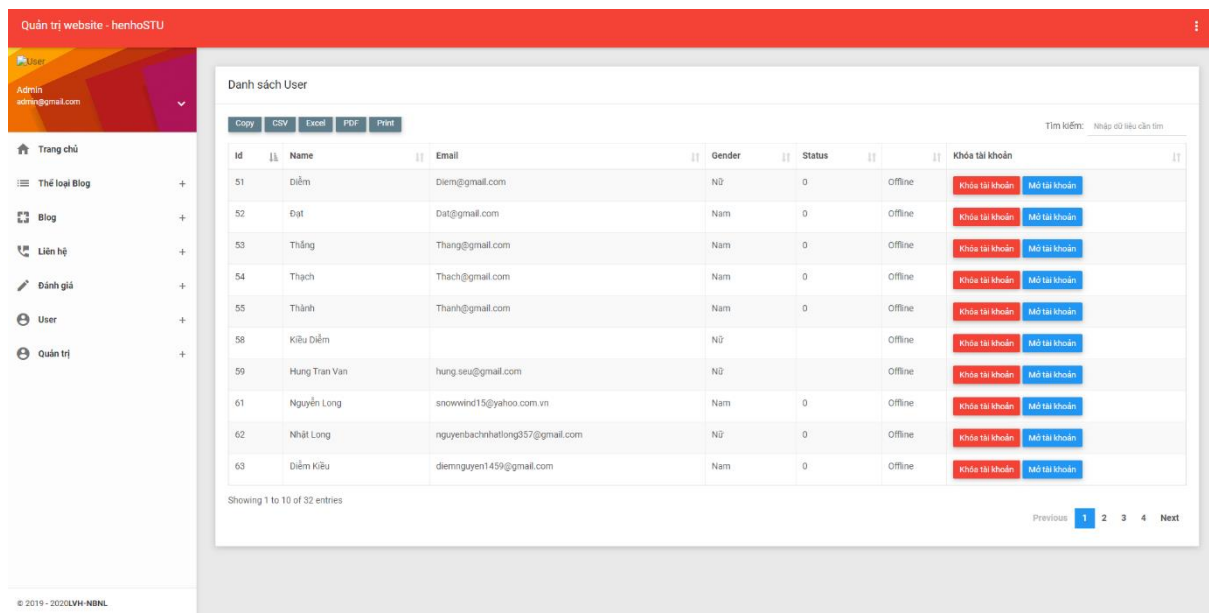
Hình 8-15 : Giao diện Thêm loại blog

Mô tả: Đây là giao diện chức năng Thêm blog mới. Admin nhập thông tin cho thể loại blog mới tại đây và chọn ẩn/hiện thể loại.



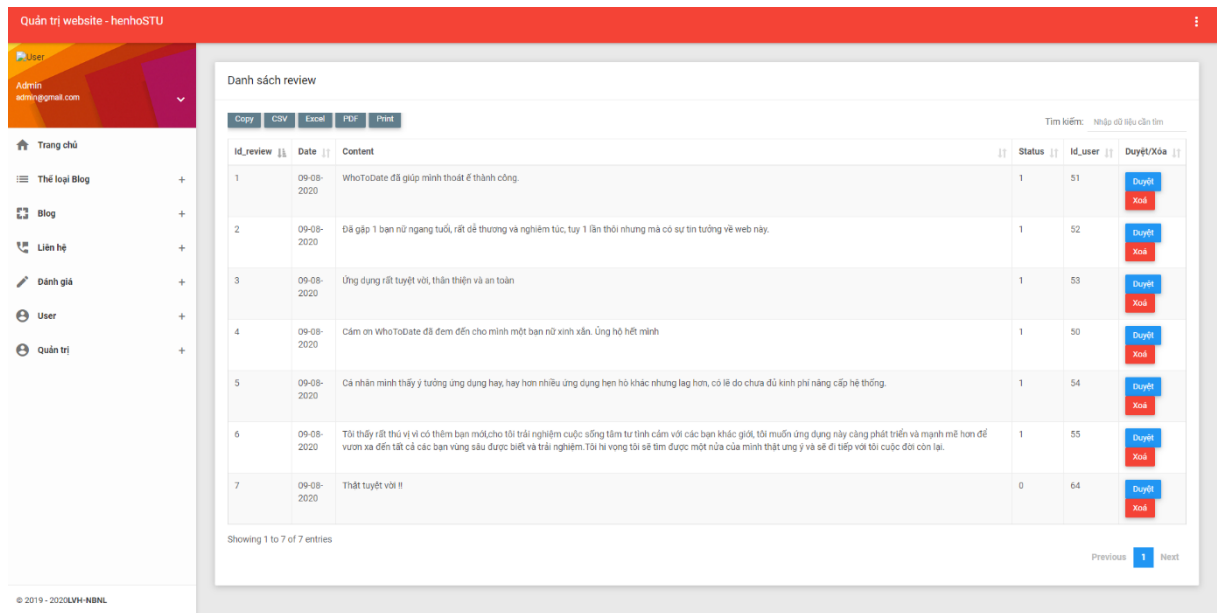
Hình 8-16 : Giao diện Liên hệ

Mô tả: Đây là giao diện chức năng liên hệ với admin. Các dữ liệu được lấy từ form Liên hệ ở trang Liên hệ. Tại đây admin có thể gửi phản hồi thông qua việc click vào button Xử lý. Sau đó cập nhật lại trạng thái Đã xử lý.



Hình 8-17 : Giao diện Quản lý User

Mô tả: Đây là giao diện chức năng Khóa tài khoản người dùng nếu có phát hiện người dùng vi phạm các chính sách của hệ thống. Ngoài ra còn có chức năng Mở khóa lại nếu người dùng chứng minh trong sạch.



Hình 8-18 : Giao diện Quản lý đánh giá

Mô tả: Đây là giao diện chức năng đánh giá mà người dùng gửi về cho admin. Admin sau khi đọc thông tin đánh giá có thể chọn Duyệt đánh giá để hiển thị lên trang chủ hoặc có thể Xóa các đánh giá xấu.

Chương 9. KẾT LUẬN

9.1 KẾT QUẢ ĐẠT ĐƯỢC

Thu được một trang web có đầy đủ các chức năng cơ bản ở hai module User và Admin.

1. Module User

- Xác thực: Đăng ký thành viên, đăng nhập hệ thống, quên mật khẩu, đổi mật khẩu.
- Trang cá nhân: cập nhật thông tin cá nhân, đổi ảnh đại diện, chi tiết trang cá nhân của một người dùng bất kỳ.
- Bạn bè: Gửi lời mời kết bạn, hiển thị danh sách bạn bè, hủy bạn bè.
- Chat: Trò chuyện với bạn bè.
- Tìm kiếm: Tìm kiếm theo tên hoặc sở thích.
- Album ảnh: Tạo album ảnh lưu trữ kỷ niệm của User.
- Status: Đăng status, xóa status, sửa status, like, bình luận, xem bình luận, phản hồi bình luận.
- Đánh giá: Gửi đánh giá về trải nghiệm sử dụng ứng dụng của User

2. Module Admin

- Xác thực: Đăng ký thành viên, đăng nhập hệ thống, quên mật khẩu, đổi mật khẩu.
- Quản lý thể loại blog: Thêm/Xóa/Sửa loại blog.
- Quản lý blog: Thêm/Xóa/Sửa blog.
- Xử lý liên hệ: Gửi phản hồi liên hệ.
- Xử lý đánh giá: Duyệt/Xóa đánh giá.
- Quản lý người dùng: Khóa/Mở khóa tài khoản.
- Quản lý quản trị: Thêm/Xóa quản trị viên.

9.2 NHỮNG CHỨC NĂNG CHƯA HOÀN THÀNH

Ghép Đôi

Hiển thị thông báo bình luận

Tìm kiếm nâng cao

9.3 HƯỚNG PHÁT TRIỂN

Để chương trình có thể hoạt động hiệu quả hơn cần dữ liệu phải được lưu trữ bằng hệ quản trị cơ sở dữ liệu có tính bảo mật hơn. Cần có sự quản lý người dùng một cách chặt chẽ hơn để việc theo dõi và quản lý của quản trị viên được tốt hơn.

Trong tương lai, cần phải xây dựng một hệ thống xác thực người dùng tốt hơn. Cụ thể là cần thêm các minh chứng xác định người dùng (CMND, giấy phép lái xe, Thẻ sinh viên,...).

Xây dựng một cộng đồng văn minh lịch sự, kiểm duyệt nội dung đăng tải, kiểm tra các tài khoản lợi dụng ứng dụng với mục đích cá nhân. Cải thiện môi trường mạng xã hội phù hợp với hệ sinh thái của trường.

TÀI LIỆU THAM KHẢO

- [1] <https://vi.wikipedia.org/wiki/Wikipedia>.
- [2] <https://www.w3schools.com/>
- [3] <https://gurayyarar.github.io/AdminBSBMaterialDesign/>
- [4] <https://getbootstrap.com/>
- [5] <https://jquery.com/>
- [6] <https://fontawesome.com/>
- [7] <https://laravel.com/docs/7.x>

