

BỘ GIÁO DỤC VÀ ĐÀO TẠO

BỘ NÔNG NGHIỆP VÀ PTNT

TRƯỜNG ĐẠI HỌC THỦY LỢI



NGUYỄN ĐÌNH DU

BUILDING A CHAT APPLICATION

USING REACTJS & NESTJS

ĐỒ ÁN TỐT NGHIỆP

HÀ NỘI, NĂM 2022

BỘ GIÁO DỤC VÀ ĐÀO TẠO BỘ NÔNG NGHIỆP VÀ PTNT
TRƯỜNG ĐẠI HỌC THỦY LỢI

NGUYỄN ĐÌNH DU

BUILDING A CHAT APPLICATION
USING REACTJS & NESTJS

Ngành: Công Nghệ Thông Tin

Mã số: 7480201

NGƯỜI HƯỚNG DẪN: ThS.Trương Xuân Nam

HÀ NỘI, NĂM 2022



CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập - Tự do - Hạnh phúc



NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

**1. TÊN ĐỀ TÀI: BUILDING A CHAT APPLICATION USING
REACTJS & NESTJS**

2. NỘI DUNG

Nội dung	Tỷ lệ(%)
Chương 1. Giới thiệu	20%
Chương 2. Cơ sở lý thuyết	20%
Chương 3. Phân tích và thiết kế hệ thống	20%
Chương 4. Xây dựng và kiểm thử	20%
Chương 5. Kết luận và hướng phát triển	20%

3. GIÁO VIÊN HƯỚNG DẪN TỪNG PHẦN

Nội dung	Họ tên giảng viên hướng dẫn
Chương 1. Giới thiệu	ThS. Trương Xuân Nam
Chương 2. Cơ sở lý thuyết	
Chương 3. Phân tích và thiết kế hệ thống	
Chương 4. Kết luận và hướng phát triển	
Chương 5. Kết luận và hướng phát triển	

4. NGÀY GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Ngày tháng năm 2022

Trưởng Bộ môn
(Ký và ghi rõ Họ tên)

Giảng viên hướng dẫn chính
(Ký và ghi rõ Họ tên)

Nhiệm vụ Đồ án tốt nghiệp đã được Hội đồng thi tốt nghiệp của Khoa thông qua.

Ngày tháng năm 2022

Giảng viên hướng dẫn chính
(Ký và ghi rõ Họ tên)

Sinh viên đã hoàn thành và nộp bản Đồ án tốt nghiệp cho Hội đồng thi.

Ngày tháng năm 2022

Giảng viên hướng dẫn chính
(Ký và ghi rõ Họ tên)

Nguyễn Đình Du

GẤY BÌA ĐỒ ÁN TỐT NGHIỆP

NGUYỄN ĐÌNH DƯ

ĐỒ ÁN TỐT NGHIỆP

HÀ NỘI, NĂM 2022

LỜI CAM ĐOAN

Tác giả xin cam đoan đây là Đồ án tốt nghiệp của bản thân tác giả. Các kết quả trong Đồ án tốt nghiệp này là trung thực, và không sao chép từ bất kỳ một nguồn nào và dưới bất kỳ hình thức nào. Việc tham khảo các nguồn tài liệu đã được thực hiện trích dẫn và ghi nguồn tài liệu tham khảo đúng quy định.

Tác giả ĐATN

Nguyễn Đình Du

LỜI CẢM ƠN

Trong thời gian thực hiện đồ án em đã được học hỏi nhiều kiến thức bổ ích trong giáo trình và thực tế từ thầy cô. Qua đây em xin gửi lời cảm ơn chân thành đến:

Quý thầy cô trong khoa Công nghệ thông tin của trường Đại Học Thủy Lợi đã truyền đạt kiến thức, những kinh nghiệm quý báu trong chuyên môn cũng như trong các lĩnh vực khác. Sự tận tụy say mê, lòng nhân ái, nhiệt tình của thầy cô là động lực lớn giúp em trau dồi thêm kiến thức giúp em vượt qua khó khăn trong học tập cũng như trong cuộc sống.

Em xin gửi lời cảm ơn chân thành nhất đến thầy Trương Xuân Nam đã giúp đỡ và hướng dẫn em hoàn thành đồ án tốt nghiệp này.

Đồng thời cũng cảm ơn đến tất cả các bạn bè đã gắn bó và giúp đỡ nhau trong học tập cũng như trong suốt quá trình thực hiện đồ án tốt nghiệp.

Xin chân thành cảm ơn!

MỤC LỤC

DANH MỤC HÌNH ẢNH	vi
DANH MỤC BẢNG BIỂU	viii
DANH MỤC CÁC TỪ VIẾT TẮT VÀ GIẢI THÍCH CÁC THUẬT NGỮ	x
1. CHƯƠNG 1 GIỚI THIỆU	1
1.1. Giới thiệu chung	1
1.2. Mô tả bài toán	2
1.3. Phương pháp nghiên cứu	2
1.3.1. Nghiên cứu lý thuyết	2
1.3.2. Nghiên cứu thực nghiệm	2
2. CHƯƠNG 2 CƠ SỞ LÝ THUYẾT	3
2.1. MÔ HÌNH CLIENT - SERVER	3
2.2. CÔNG NGHỆ VÀ NỀN TẢNG PHÁT TRIỂN	4
2.2.1. Cơ sở dữ liệu SQL	4
2.2.2. NodeJs (NestJs)	5
2.2.3. Socket.Io, WEB-RTC	10
2.2.4. ReactJS	13
2.3. GIỚI THIỆU VỀ MÔ HÌNH VÀ QUY TRÌNH VỀ CÔNG NGHỆ	Error!
Bookmark not defined.	
2.3.1. Mô hình quy trình thác nước	Error! Bookmark not defined.
2.3.2. Phương pháp phân tích và thiết kế phần mềm	Error! Bookmark not defined.
3. CHƯƠNG 3 PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	15
3.1. XÁC ĐỊNH TÁC NHÂN HỆ THỐNG	15

3.1.1.	Tác nhân hệ thống	15
3.1.2.	Tác nhân và chức năng tương ứng	15
3.2.	YÊU CẦU PHI CHỨC NĂNG	16
3.3.	SƠ ĐỒ USECASE.....	16
3.3.1.	Sơ đồ usecase tổng quát	16
3.3.2.	Sơ đồ usecase phân rã.....	17
3.4.	QUY TRÌNH NGHIỆP VỤ	19
3.4.1.	Quy trình truy cập hệ thống.....	19
3.4.2.	Quy trình quản lý bạn bè	20
3.4.3.	Quy trình đăng nhập	21
3.4.4.	Quy trình đăng ký	22
3.4.5.	Quy trình quên mật khẩu	23
3.4.6.	Quy trình nhắn tin.....	24
3.4.7.	Quy trình gọi điện.....	25
3.5.	ĐẶC TẢ YÊU CẦU HỆ THỐNG	26
3.5.1.	Đăng nhập.....	26
3.5.2.	Đăng xuất.....	30
3.5.3.	Đăng ký	32
3.5.4.	Quên mật khẩu.....	36
3.5.5.	Chức năng nhắn tin/ gọi điện	38
3.5.6.	Chức năng tạo nhóm trò chuyện.....	41
3.5.7.	Chức năng thêm bạn bè	43
3.5.8.	Chức năng xóa bạn bè	46
3.6.	BIỂU ĐỒ LỚP THỰC THỂ.....	48

3.6.1.	Biểu đồ thực thể liên kết(ERD)	48
3.6.2.	Biểu đồ lớp	49
3.7.	THIẾT KẾ CƠ SỞ DỮ LIỆU	50
3.7.1.	Chi tiết bảng conversations	50
3.7.2.	Chi tiết bảng friends	51
3.7.3.	Chi tiết bảng friend_requests.....	51
3.7.4.	Chi tiết bảng groups	52
3.7.5.	Chi tiết bảng groups_users_users	53
3.7.6.	Chi tiết bảng group_messages.....	53
3.7.7.	Chi tiết bảng group_message_attachments	54
3.7.8.	Chi tiết bảng messages	54
3.7.9.	Chi tiết bảng message_attachments.....	55
3.7.10.	Chi tiết bảng peer	55
3.7.11.	Chi tiết bảng profiles	55
3.7.12.	Chi tiết bảng sessions	56
3.7.13.	Chi tiết bảng users	56
3.7.14.	Chi tiết bảng user_presence.....	Error! Bookmark not defined.
3.8.	THIẾT KẾ GIAO DIỆN.....	59
4.	CHƯƠNG 4 XÂY DỰNG VÀ KIỂM THỬ'	64
4.1.	XÂY DỰNG.....	64
4.2.	KIỂM THỬ'.....	64
5.	CHƯƠNG 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	68
5.1.	KẾT LUẬN	68
5.2.	HƯỚNG PHÁT TRIỂN	68

TÀI LIỆU THAM KHẢO.....	69
-------------------------	----

DANH MỤC HÌNH ẢNH

Hình 2.1. Truyền thông tin giữa client và server	3
Hình 2.2. Các góc nhìn OOAD sử dụng UML	Error! Bookmark not defined.
Hình 2.3. Ký hiệu usecase.....	Error! Bookmark not defined.
Hình 2.4. Ký hiệu về class	Error! Bookmark not defined.
Hình 2.5. Ký hiệu về actor	Error! Bookmark not defined.
Hình 3.1. Sơ đồ usecase tổng quát	16
Hình 3.2. Sơ đồ usecase quản lý danh sách bạn bè.....	17
Hình 3.3. Sơ đồ usecase gửi tin nhắn, gọi điện.....	17
Hình 3.4 Sơ đồ usecase tạo nhóm trò chuyện	18
Hình 3.5 Biểu đồ hoạt động sử dụng phần mềm.....	19
Hình 3.6 Biểu đồ hoạt động quản lý bạn bè.....	20
Hình 3.7 Biểu đồ hoạt động chức năng đăng nhập	21
Hình 3.8 Biểu đồ hoạt động chức năng đăng ký	22
Hình 3.9 Biểu đồ hoạt động chức năng quên mật khẩu	23
Hình 3.10 Biểu đồ hoạt động chức năng nhắn tin.....	24
Hình 3.11. Biểu đồ hoạt động chức năng gọi điện.....	25
Hình 3.12. Usecase đăng nhập	26
Hình 3.13. Biểu đồ luồng xử lý chức năng đăng nhập.....	27
Hình 3.14. Biểu đồ tuần tự chức năng đăng nhập.....	28
Hình 3.15. Usecase đăng xuất.....	30
Hình 3.16. Biểu đồ luồng xử lý chức năng đăng xuất.....	31
Hình 3.17. Biểu đồ tuần tự chức năng đăng xuất.....	31
Hình 3.18. Usecase đăng ký	32
Hình 3.19 Biểu đồ luồng xử lý chức năng đăng ký	34

Hình 3.20 Biểu đồ tuần tự chức năng đăng ký.....	34
Hình 3.21. Biểu đồ luồng xử lý chức năng quên mật khẩu.....	37
Hình 3.22. Biểu đồ tuần tự chức năng quên mật khẩu.....	37
Hình 3.23. Biểu đồ luồng xử lý chức năng nhắn tin/gọi điện	39
Hình 3.24. Biểu đồ tuần tự chức năng nhắn tin/gọi điện	40
Hình 3.25. Biểu đồ luồng xử lý chức năng tạo nhóm trò chuyện	42
Hình 3.26. Biểu đồ tuần tự chức năng tạo nhóm trò chuyện	42
Hình 3.27. Biểu đồ luồng xử lý chức năng thêm bạn bè.....	44
Hình 3.28. Biểu đồ tuần tự chức năng thêm bạn bè.....	45
Hình 3.29. Biểu đồ luồng xử lý chức năng xóa bạn bè.....	47
Hình 3.30. Biểu đồ tuần tự chức năng xóa bạn bè	47
Hình 3.31. Biểu đồ thực thể liên kết	48
Hình 3.32. Biểu đồ lớp.....	49
Hình 3.33. Cơ sở dữ liệu	58
Hình 3.34. Giao diện đăng nhập.....	59
Hình 3.35. Giao diện đăng ký	59
Hình 3.36. Giao diện Trang chủ.....	60
Hình 3.37. Giao diện nhắn tin	60
Hình 3.38. Giao diện danh sách bạn bè.....	61
Hình 3.39. Giao diện thêm bạn bè.....	61
Hình 3.40. Giao diện Profile	62
Hình 3.41. Giao diện cuộc gọi đến.....	62
Hình 3.42. Giao diện Video Call.....	63
Hình 3.43. Giao diện cuộc gọi thoại	63

DANH MỤC BẢNG BIỂU

Bảng 3.1. Bảng mô tả thông tin về tác nhân và các chức năng tương ứng	15
Bảng 3.2. Thông tin chức năng đăng nhập.....	26
Bảng 3.3. Bảng mô tả dòng sự kiện chính chức năng đăng nhập	28
Bảng 3.4. Bảng mô tả dòng sự kiện phụ chức năng đăng nhập	29
Bảng 3.5. Thông tin chức năng đăng xuất	30
Bảng 3.6. Bảng mô tả dòng sự kiện chính chức năng đăng xuất	32
Bảng 3.7. Bảng thông tin chức năng đăng ký	33
Bảng 3.8. Bảng mô tả dòng sự kiện chính chức năng đăng ký	35
Bảng 3.9. Bảng mô tả dòng sự kiện phụ chức năng đăng ký	36
Bảng 3.10. Bảng thông tin chức năng quên mật khẩu	36
Bảng 3.11. Bảng mô tả dòng sự kiện chính chức năng quên mật khẩu	38
Bảng 3.12. Bảng thông tin chức năng nhắn tin/gọi điện.....	38
Bảng 3.13. Bảng mô tả dòng sự kiện chính chức năng nhắn tin/gọi điện.....	40
Bảng 3.14. Bảng thông tin chức năng tạo nhóm trò chuyện.....	41
Bảng 3.15. Bảng mô tả dòng sự kiện chính chức năng tạo nhóm trò chuyện.....	43
Bảng 3.16. Bảng thông tin chức năng thêm bạn bè	43
Bảng 3.17. Bảng mô tả dòng sự kiện chính chức năng thêm bạn bè	45
Bảng 3.18. Bảng thông tin chức năng xóa bạn bè.....	46
Bảng 3.19. Bảng mô tả dòng sự kiện chính chức năng danh sách tin tuyển dụng.....	48
Bảng 3.20. Bảng conversations	50
Bảng 3.21. Bảng friends.....	51
Bảng 3.22. Bảng friend_requests	51
Bảng 3.23. Bảng groups	52
Bảng 3.24. Bảng groups_users_users	53
Bảng 3.25. Bảng group_messages	53
Bảng 3.26. Bảng group_message_attachments.....	54
Bảng 3.27. Bảng messages.....	54

Bảng 3.28. Bảng message_attachments	55
Bảng 3.29. Bảng peer	55
Bảng 3.30. Bảng profiles.....	55
Bảng 3.31. Bảng sessions.....	56
Bảng 3.32. Bảng user_presence	56
Bảng 3.33. Bảng users.....	57
Bảng 4.1. Bảng kiểm thử.....	64

DANH MỤC CÁC TỪ VIẾT TẮT VÀ GIẢI THÍCH CÁC THUẬT NGỮ

STT	Ký hiệu từ viết tắt	Chữ viết đầy đủ
1	IoT	Internet of Things
2	NPM	Node Package Manager
3	REST	REpresentational State Transfer
4	SPA	Single-page application
5	SQL	Structured Query Language

CHƯƠNG 1 GIỚI THIỆU

1.1. Giới thiệu chung

Hiện nay, thế giới đang chứng kiến sự phát triển vượt bậc của Internet và các ứng dụng trên Internet. Cùng với sự phát triển đó cộng với nhu cầu trao đổi và thông tin liên lạc một cách nhanh chóng và tiện lợi đã thúc đẩy sự phát triển các phần mềm để trao đổi thông tin một cách tức thì. Điển hình cho các phần mềm đó là ứng dụng “chat”. Các ứng dụng cho phép người dùng gửi và nhận các thông điệp nhanh chóng một cách trực tiếp với nhau. Vì thế những ứng dụng chat ra đời. Nhưng vấn đề đặt ra là với các phần mềm “chat” như trên thì một yêu cầu gần như bắt buộc là việc phải cài đặt ứng dụng để có thể sử dụng chúng. Từ đó nảy sinh ra ý tưởng đưa các ứng dụng chat lên Web. Một câu hỏi đặt ra là tại sao lại là web và sự tiện lợi có được là gì khi đưa những ứng dụng “chat” lên web. Câu trả lời rất đơn giản: Với “Web” bạn có thể thao tác bất cứ đâu có Internet mà không yêu cầu phải cài đặt ứng dụng. Điều đó mang lại sự thuận tiện cho người dùng và đồng thời là sự tiết kiệm tài nguyên máy tính một cách đáng kể. Thử tưởng tượng nếu bạn phải làm việc với một máy tính được kết nối Internet nhưng lại không cài sẵn một chương trình ứng dụng chat mà bạn cần cho việc trao đổi thông tin với người khác. Thì việc phải tải ứng dụng và cài đặt ứng dụng đó lên máy tính rõ ràng là rắc rối và phiền phức hơn rất nhiều so với việc dùng một trang web có cùng chức năng.

Cùng với đó, với sự phát triển bùng nổ của công nghệ web và sự hỗ trợ ngày càng mạnh của các ngôn ngữ lập trình. Việc tạo ra một trang web có khả năng hoạt động với các chức năng như một ứng dụng được cài trên máy tính là hoàn toàn có thể. Vì vậy, việc đưa một ứng dụng trên máy tính lên thành một trang web trở thành một nhu cầu thiết thực và cần thiết.

Với những lý do trên, đề án sẽ tập trung để giải quyết vấn đề xây dựng một ứng dụng web chat với đề tài “Building a chat application using reactjs & nestjs”. Với mục tiêu đề tài là xây dựng một trang web có khả năng tương tự như những phần mềm chat được cài đặt trên máy tính đã có trước đây.

1.2. Mô tả bài toán

Một website có chức năng phục vụ người dùng trao đổi thông tin trực tiếp với nhau. Đầu tiên người dùng truy cập trang web, đăng ký một tài khoản trên website. Sau đó người dùng đăng nhập hệ thống. Sau khi người dùng đăng nhập vào hệ thống. Hệ thống sẽ hiển thị danh sách bạn của người dùng, các thông điệp được gửi đến cho người dùng khi người dùng không online (offline message). Khi tham gia hệ thống người dùng có thể thực hiện một số thao tác:

- Đăng nhập.
- Đăng ký.
- Thêm bạn mới.
- Xóa một hay nhiều người trong danh sách bạn
- Gửi message tới một người dùng khác.
- Lập nhóm trò chuyện với nhiều bạn bè.
- Chỉnh sửa lại thông tin cá nhân của mình sau khi đăng ký.
- Thay đổi mật khẩu đăng nhập của tài khoản.

1.3. Phương pháp nghiên cứu

1.3.1. Nghiên cứu lý thuyết

- Phương pháp quan sát: Từ quy trình nghiệp vụ trong quá trình khảo sát, trao đổi với khách hàng để có được những yêu cầu của khách hàng phục vụ cho việc thiết kế và xây dựng hệ thống.
- Phương pháp thu thập tài liệu: Dựa trên các yêu cầu của khách hàng thu thập được
- Phương pháp phân tích thiết kế hệ thống: Phân tích và thiết kế theo hướng đối tượng

1.3.2. Nghiên cứu thực nghiệm

Xây dựng được 1 website đáp ứng đầy đủ các yêu cầu đề ra

CHƯƠNG 2 CƠ SỞ LÝ THUYẾT

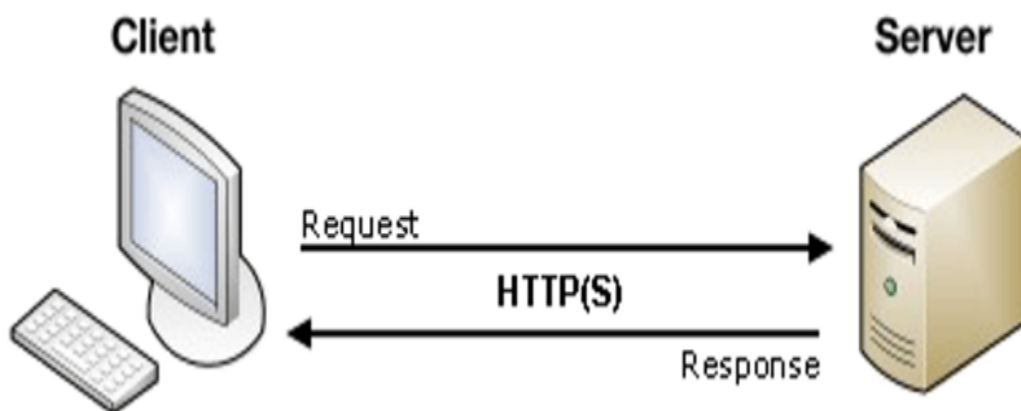
2.1. MÔ HÌNH CLIENT - SERVER

Client server là mô hình mạng máy tính gồm có hai thành phần chính đó là máy khách (client) và máy chủ (server). Server chính là nơi giúp lưu trữ tài nguyên cũng như cài đặt các chương trình dịch vụ theo đúng như yêu cầu của client. Ngược lại, client bao gồm máy tính cũng như các loại thiết bị điện tử nói chung sẽ tiến hành gửi yêu cầu đến server. Mô hình mạng Client Server sẽ cho phép mạng tập trung các ứng dụng có cùng chức năng tại một hoặc nhiều dịch vụ file chuyên dụng. Chúng sẽ trở thành trung tâm của hệ thống. Hệ điều hành của mô hình client server sẽ cho phép người dùng chia sẻ đồng thời cùng một loại tài nguyên mà không giới hạn vị trí địa lý.

- Nguyên tắc hoạt động của mô hình Client - Server

Trong mô hình Client Server, server chấp nhận tất cả các yêu cầu hợp lệ từ mọi nơi khác nhau trên internet, sau đó trả kết quả về máy tính đã gửi yêu cầu đó.

Máy tính được coi là máy khách khi chúng làm nhiệm vụ gửi yêu cầu đến các máy chủ và đợi câu trả lời được gửi về.



Hình 2.1. Truyền thông tin giữa client và server

Để máy khách và máy chủ có thể giao tiếp được với nhau thì giữa chúng phải có một chuẩn nhất định và chuẩn đó được gọi là giao thức. Một số giao thức phổ biến hiện nay như HTTPS, TCP/ IP, FTP, ...

Nếu máy khách muốn lấy được thông tin từ máy chủ, chúng phải tuân theo một giao thức mà máy chủ đó đưa ra. Nếu yêu cầu đó được chấp nhận thì máy chủ sẽ thu thập thông tin và trả về kết quả cho máy khách yêu cầu. Bởi vì server – máy chủ luôn luôn trong trạng thái sẵn sàng để nhận request từ client nên chỉ cần client gửi yêu cầu tín hiệu và chấp nhận yêu cầu đó thì server sẽ trả kết quả về phía client trong thời gian ngắn nhất.

2.2. CÔNG NGHỆ VÀ NỀN TẢNG PHÁT TRIỂN

- Backend Module
 - Sử dụng SQL để lưu trữ dữ liệu
 - Sử dụng NodeJs(NestJs)
 - Sử dụng Socket.IO và WebRTC để giao tiếp thời gian thực
- Frontend Module
 - Sử dụng ReactJS để dựng giao diện website

2.2.1. Cơ sở dữ liệu SQL

SQL là viết tắt của **Structured Query Language** là ngôn ngữ truy vấn có cấu trúc. Nó là một ngôn ngữ, là tập hợp các lệnh để tương tác với cơ sở dữ liệu. Dùng để lưu trữ, thao tác và truy xuất dữ liệu được lưu trữ trong một cơ sở dữ liệu quan hệ. Trong thực tế, SQL là ngôn ngữ chuẩn được sử dụng hầu hết cho hệ cơ sở dữ liệu quan hệ. Tất cả các hệ thống quản lý cơ sở dữ liệu quan hệ (RDMS) như MySQL, MS Access, Oracle, Postgres và SQL Server... đều sử dụng SQL làm ngôn ngữ cơ sở dữ liệu chuẩn.

Hầu như công ty nào lớn cũng cần xây dựng một hệ thống để lưu trữ cơ sở dữ liệu. Mọi thứ trong cơ sở dữ liệu này sẽ được diễn tả ra thành nhiều bảng, có mối quan hệ với nhau. Để truy vấn và lấy dữ liệu từ các bảng này nhằm tổng hợp thành thông tin nào đó, người ta dùng đến SQL thông qua các câu query.

SQL bắt đầu từ những năm 1970, khi các kỹ sư của IBM là Donald Chamberlin và Raymond Boyce thiết kế phiên bản đầu tiên để tương tác và lấy dữ liệu được lưu trữ trong hệ thống cơ sở dữ liệu của công ty. Họ gọi nó là SEQUEL, mặc dù sau đó họ buộc phải thay đổi nó do các vấn đề bản quyền.

SQL được sử dụng phổ biến vì nó có các ưu điểm sau:

- Cho phép truy cập dữ liệu trong các hệ thống quản lý cơ sở dữ liệu quan hệ.
- Cho phép mô tả dữ liệu.
- Cho phép xác định dữ liệu trong cơ sở dữ liệu và thao tác dữ liệu đó.
- Cho phép nhúng trong các ngôn ngữ khác sử dụng mô-đun SQL, thư viện và trình biên dịch trước.
- Cho phép tạo và thả các cơ sở dữ liệu và bảng.
- Cho phép tạo chế độ view, thủ tục lưu trữ, chức năng trong cơ sở dữ liệu.
- Cho phép thiết lập quyền trên các bảng, thủ tục và view.

SQL sẽ giúp quản lý hiệu quả và truy vấn thông tin nhanh hơn, giúp bảo trì, bảo mật thông tin dễ dàng hơn.

2.2.2. NodeJs (NestJs)

❖ NodeJS

NodeJS là một môi trường runtime chạy JavaScript đa nền tảng và có mã nguồn mở, được sử dụng để chạy các ứng dụng web bên ngoài trình duyệt của client. Nền tảng này được phát triển bởi Ryan Dahl vào năm 2009, được xem là một giải pháp hoàn hảo cho các ứng dụng sử dụng nhiều dữ liệu nhờ vào mô hình hướng sự kiện (event-driven) không đồng bộ.

Ban đầu, tác giả đặt tên dự án là web.js vì mục đích đơn thuần chỉ là sử dụng như một ứng dụng web thay cho Apache hay các server khác. Tuy nhiên, dự án nhanh chóng phát triển rất mạnh, vượt qua những khả năng của một webserver thông thường và thậm chí

có thể được xem như một nền tảng đa mục đích. Vì vậy, sau này dự án đã được đổi tên thành Node.JS và sử dụng đến ngày nay.

- Ưu điểm của NodeJs
 - IO hướng sự kiện không đồng bộ, cho phép xử lý nhiều yêu cầu đồng thời.
 - Sử dụng JavaScript – một ngôn ngữ lập trình dễ học.
 - Chia sẻ cùng code ở cả phía client và server.
 - NPM(Node Package Manager) và module Node đang ngày càng phát triển mạnh mẽ.
 - Cộng đồng hỗ trợ tích cực.
 - Cho phép stream các file có kích thước lớn.
- Nhược điểm của NodeJs
 - Không có khả năng mở rộng, vì vậy không thể tận dụng lợi thế mô hình đa lõi trong các phần cứng cấp server hiện nay.
 - Khó thao tác với cơ sở dữ liệu quan hệ.
 - Mỗi callback sẽ đi kèm với rất nhiều callback lồng nhau khác.
 - Cần có kiến thức tốt về JavaScript.
 - Không phù hợp với các tác vụ đòi hỏi nhiều CPU.
- Những ứng dụng nên viết bằng Node.JS

NodeJS được sử dụng để xây dựng rất nhiều loại ứng dụng khác nhau, trong đó phổ biến nhất gồm có:

- **Ứng dụng trò chuyện trong thời gian thực:** Nhờ vào cấu trúc không đồng bộ đơn luồng, Node.JS rất thích hợp cho mục đích xử lý giao tiếp trong thời gian thực. Nền tảng này có thể dễ dàng mở rộng quy mô và thường dùng để tạo ra các chatbot. Bên cạnh đó, các tính năng liên quan đến ứng dụng trò chuyện như: chat nhiều người, thông báo đẩy... cũng có thể dễ dàng được bổ sung nhờ NodeJS.

- **Internet of Things (IoT):** Các ứng dụng IoT thường bao gồm nhiều bộ cảm biến phức tạp để gửi những phần dữ liệu nhỏ. Node.JS là một lựa chọn lý tưởng để xử lý các yêu cầu đồng thời này với tốc độ cực nhanh.
- **Truyền dữ liệu:** Netflix là một trong số những công ty lớn trên thế giới chuyên sử dụng Node.JS cho mục đích truyền dữ liệu. Sở dĩ vì đây là một nền tảng nhẹ và cực nhanh, đồng thời còn cung cấp một API chuyên dùng để stream.
- **Các SPA (Single-page application) phức tạp:** Trong SPA, toàn bộ ứng dụng được load vào trong một trang duy nhất, do đó sẽ có một số request được thực hiện trong nền. Vòng lặp sự kiện (event loop) của Node.JS cho phép xử lý các request theo hướng non-blocking.
- **Các ứng dụng REST dựa trên API:** JavaScript được sử dụng trong cả frontend lẫn backend của trang. Do đó một server có thể dễ dàng giao tiếp với frontend qua REST API bằng Node.js. Bên cạnh đó, Node.JS còn cung cấp nhiều package như Express.js hay Koa để việc xây dựng ứng dụng web trở nên dễ dàng hơn bao giờ hết.

- Một số lý do nên sử dụng NodeJS:

Node.JS là một trong những nền tảng phổ biến nhất hiện nay cho mục đích phát triển ứng dụng mạng phía server. Vậy lý do nên sử dụng NodeJS là gì? Hãy cùng tìm hiểu những đặc điểm khiến Node.JS là lựa chọn hàng đầu cho các developer hiện nay:

- **Tốc độ cực nhanh:** Được xây dựng dựa trên engine JavaScript V8 của Google Chrome, do đó các thư viện của nó có khả năng thực thi code chỉ rất nhanh.
- **NPM:** Với hơn 50,000 package khác nhau, các developer có thể dễ dàng lựa chọn bất kỳ tính năng nào để xây dựng cho ứng dụng của mình.
- **Lập trình không đồng bộ:** Mọi API của Node.JS đều có tính không đồng bộ (non-blocking), do đó một server dựa trên Node.JS không cần phải đợi API trả về dữ liệu.

- **Không có buffering:** Node.JS giúp tiết kiệm thời gian xử lý file khi cần upload âm thanh hoặc video vì các ứng dụng này không bao giờ buffer dữ liệu mà chỉ xuất dữ liệu theo từng phần (chunk).
- **Đơn luồng:** NodeJS sử dụng mô hình đơn luồng với vòng lặp sự kiện. Do đó các ứng dụng có thể xử lý số lượng request lớn hơn rất nhiều so với các server truyền thống như Apache HTTP Server.

❖ NestJS

NestJS là một framework Node.JS cho phép xây dựng ứng dụng phía server. Nest mở rộng các framework Node.js như Express hay Fastify để bổ sung thêm nhiều module hay thư viện hỗ trợ việc xử lý tác vụ. Đây là một framework mã nguồn mở, sử dụng TypeScript và rất linh hoạt để xây dựng các hệ thống backend.

Bên cạnh đó, NestJS còn giúp mở rộng các server Node để hỗ trợ những cơ sở dữ liệu như MongoDB, Redis hay Apache Cassandra.

- Tại sao nên sử dụng NestJS?

Java là một ngôn ngữ lập trình với tuổi đời hơn 20 năm, từng là framework chính cho nhiều ứng dụng kinh doanh. Tuy nhiên những năm gần đây các developer đang dần chuyển sự chú ý của mình sang một cái tên mới: Node.JS. Đây là một nền tảng có tốc độ cực nhanh, hỗ trợ JavaScript và cho phép các developer xây dựng cả frontend lẫn backend chỉ với một ngôn ngữ lập trình duy nhất.

- Cho phép develop nhanh và hiệu quả hơn.
- Khả năng mở rộng tốt, dễ bảo trì ứng dụng.
- Là framework Node.js phát triển mạnh nhất trong 3 năm trở lại đây.
- Cộng đồng hỗ trợ lớn, tích cực.

- Kết hợp phát triển front-end và mid-tier, một đặc điểm vượt trội so với hầu hết các ngôn ngữ khác.
 - Sử dụng TypeScript, cho phép thích ứng nhanh chóng với các thay đổi khi JavaScript đang ngày càng phát triển mạnh mẽ.
 - Nguồn tài liệu hướng dẫn phong phú, chi tiết.
 - Quá trình unit testing trở nên đơn giản hơn.
 - Được xây dựng chuyên dùng cho các ứng dụng doanh nghiệp có quy mô lớn.
 - Cung cấp kiến trúc ứng dụng độc lập, cho phép các developer tạo ra những ứng dụng dễ test, dễ mở rộng và dễ bảo trì.
 - Cho phép xây dựng ứng dụng Rest API, MVC, microservices, GraphQL, Web Socket hay CRON job.
 - Cấu trúc chủ yếu dựa vào Angular – rất đơn giản và cho phép tập trung vào việc thiết kế endpoint thay vì cấu trúc của ứng dụng.
 - Cung cấp các module, dịch vụ và controller giống Angular, cho phép ứng dụng có khả năng mở rộng và test tốt hơn so với Express hay Koa.
 - Có tính chất kết nối lỏng, do đó những lỗi cũ trong dự án không ảnh hưởng đến cấu trúc của codebase sau này.
- Các thành phần chính của NestJs
 - Controller
 - Provider
 - Module

2.2.3. Socket.Io, WEB-RTC

❖ Socket.Io

Khi truy cập vào 1 trang web hoặc ứng dụng bất kỳ thì việc giao tiếp giữa máy chủ (Server) và máy khách (Client) là việc rất quan trọng. Để máy chủ và máy khách có thể nhận biết được sự thay đổi của đối phương thì cần sử dụng những cách thức như AJAX, long-polling, short-polling, & HTML5 server-sent events... Việc sử dụng cách giao tiếp bằng những công cụ kể trên tồn tại nhiều nhược điểm trong đó có thể kể đến là kết quả trả về chậm và tốn rất nhiều tài nguyên.

Để khắc phục những nhược điểm này, công cụ Socket.io ra đời để giúp cho việc giao tiếp giữa Server và Client diễn ra tức khắc và chiếm ít tài nguyên nhất.

Socket io là gì? Nó được xem là 1 module trong Node.js được nhà sáng chế tạo ra và phát triển từ năm 2010. Mục đích lớn nhất của Socket io là để tạo môi trường giao tiếp thuận lợi trên Internet giúp trả về các giá trị thực ngay tại thời điểm giao tiếp giữa các bên với nhau (thường là giữa server và client).

Việc giao tiếp 2 chiều giữa máy khách và máy chủ được thực hiện bởi socket io khi và chỉ khi máy khách có module này trong trình duyệt và máy chủ cũng đã tích hợp sẵn gói socket io. Các ứng dụng sử dụng socket io thường đòi hỏi tốc độ phản hồi ngay lập tức. Một số ví dụ điển hình như xổ số, trực tiếp bóng đá, chat...

Cần phải nói thêm rằng socket io không phải là 1 ngôn ngữ, vì vậy nó phải được sử dụng kết hợp với những ngôn ngữ khác như php, asp.net, nodejs... Đến đây nếu bạn còn thắc mắc socket io là gì, hãy cùng tìm hiểu tiếp về đặc điểm của nó ở phần tiếp theo.

- Ưu điểm của SocketIO
 - Bảo mật cao.
 - Kết nối tự động tới server.
 - Mã hóa nhị phân.

- Cho phép tạo kênh và phòng.

❖ WEBRTC

Webrtc được viết tắt từ cụm “Web realtime communication” và được hiểu là giao tiếp với website theo thời gian thực. Nó bao gồm các hàm lập trình (web API) và được viết bằng JavaScript cho phép mọi người trên website trò chuyện trực tiếp thông qua video trên tất cả các trình duyệt.

Webrtc là một mã nguồn mở được duy trì bởi nhóm **Google Webrtc** dưới sự hỗ trợ của Apple, Google, Microsoft và Mozilla cùng những ông lớn trong lĩnh vực công nghệ khác. Nó mang đến rất nhiều lợi ích như sau.

- Webrtc cho phép người dùng gọi điện video và trò chuyện trực tiếp với nhau thông qua trình duyệt web
 - Webrtc cho phép người dùng chia sẻ các file dữ liệu, hình ảnh, video trong thời gian giao tiếp thực. Ví dụ như chức năng share màn hình hoặc gửi file của Zoom
 - Webrtc cho phép bạn có thể tổ chức các buổi hội thảo trực tuyến với đông thành viên cùng giao tiếp với nhau. Để dễ hiểu thì bạn có thể hình dung nó giống như chức năng họp online của ứng dụng Google meet vậy.
 - Webrtc còn cho phép người dùng tạo ra và chơi các game trực tuyến ứng với thời gian thực.
- Các thành phần làm nên công dụng của Webrtc
 - RTCdatachannel
 - Media stream
 - RTCPeerConnection

- Ưu điểm của WebRTC
 - WebRTC là thân thiện với tất cả các hệ điều hành và ngôn ngữ lập trình
 - WebRTC miễn phí
 - WebRTC thân thiện với tất cả các thiết bị
 - Dùng WebRTC không cần các ứng dụng hoặc Plugin hỗ trợ
 - WebRTC có tính bảo mật cao
 - WebRTC giúp tiết kiệm chi phí đi lại, giao tiếp do người dùng có thể kết nối trao đổi trực tiếp qua internet
- Nhược điểm của WebRTC
 - Khi cố gắng thực hiện kết nối P2P (Peer to peer - hình thức giao tiếp giữa các máy tính với nhau không thông qua máy chủ) WebRTC có thể bị cản bởi tường lửa (firewall) và NAT
 - Với mỗi trình duyệt WebRTC hỗ trợ các hàm API khác nhau nên có khả năng phát sinh lỗi khi kết nối giữa 2 máy khác trình duyệt.
 - Video truyền giữa 2 máy chưa được các hàng trình duyệt thống nhất nên video trên WebRTC có thể có chất lượng khác nhau.

2.2.4. ReactJS

ReactJs là một thư viện JavaScript, được sử dụng trong develop web. Nó được sử dụng nhằm xây dựng các phần tử tương tác trên trang web. Tuy nhiên, trước tiên ta nên tìm hiểu JavaScript và thư viện JavaScript là gì...

- Lý do các developer JavaScript sử dụng ReactJS:

React là một thư viện JavaScript, chuyên giúp các developer xây dựng giao diện người dùng (UI). Trong lĩnh vực website và ứng dụng web, UI là tập hợp các menu on-screen, search bar, button..., tức là mọi thứ mà người dùng sử dụng để tương tác với website hoặc app.

Trước khi ReactJS xuất hiện, các developer thường khó khăn với việc xây dựng UI thủ công bằng “vanilla JavaScript” (tức là ngôn ngữ JavaScript thô). Trước đó, các developer cũng thường sử dụng các tiện ích của React JS như jQuery. Tuy nhiên các thư viện này lại không thực sự tập trung vào UI. Do đó, quá trình develop sẽ lâu hơn, xuất hiện nhiều lỗi và bug hơn. Và vào năm 2011, một kỹ sư Facebook – Jordan Walke đã tạo ra React JS, tập trung vào mảng develop UI.

Ngoài việc cung cấp các code có thể tái sử dụng, React cũng đi kèm với hai tính năng chính, giúp nó nổi bật hơn trong mắt các developer JavaScript:

- JSX

Trung tâm của mọi trang web cơ bản chính là tài liệu HTML. Các trình duyệt web sẽ đọc những tài liệu này, sau đó hiển thị chúng trên các thiết bị output dưới dạng trang web. Trong quá trình này, các trình duyệt tạo ra một thứ gọi là Document Object Model (DOM). Đây chính là một biểu đồ cây đại diện cách trang web được sắp xếp. Các developer có thể thêm nội dung động vào dự án bằng cách sửa đổi DOM với các ngôn ngữ như JavaScript.

JSX (JavaScript eXtension) là một tiện ích React. Nó giúp các web developer dễ dàng sửa đổi DOM bằng cách sử dụng code HTML-style. Hơn nữa, vì trình duyệt React JS hỗ trợ tất cả trình duyệt hiện đại, nên JSX có thể tương thích với mọi nền tảng trình duyệt hiện nay.

Bên cạnh đó, sử dụng JSX để update một DOM sẽ cải thiện đáng kể hiệu suất trang web và hiệu quả develop. Nhưng bằng cách nào? Đó chính là nhờ vào tính năng thứ hai, Virtual DOM.

- Virtual DOM

Nếu không sử dụng React JS (và JSX), website sẽ dùng HTML để update DOM của nó. Việc này về cơ bản là có thể hoạt động tốt với các trang web tĩnh và đơn giản. Tuy nhiên, thực hiện nó với các trang web động, liên quan đến sự tương tác lớn từ phía người dùng, lại là một vấn đề. Bởi vì toàn bộ DOM cần được reload mỗi khi người dùng click vào tính năng yêu cầu refresh trang.

Tuy nhiên, nếu một developer sử dụng JSX để thao tác và update DOM, React JS sẽ tạo ra Virtual DOM. Virtual DOM là một bản sao của DOM từ trang web. Và React JS sẽ sử dụng bản sao này để xem những phần nào của DOM thực sự cần thay đổi.

Lấy ví dụ khi người dùng nhập một comment trên blog nào đó, và nhấp vào nút “Comment” để đăng. Nếu không sử dụng React JS, toàn bộ DOM sẽ phải được update lại. Mặt khác, React sẽ quét Virtual DOM để kiểm tra điều gì sẽ thay đổi sau hành động cụ thể của người dùng. Trong trường hợp này là khi người dùng nhấp vào nút “Comment”. Từ đó, ReactJS sẽ chỉ update phần tương ứng của DOM một cách chọn lọc. Việc này sẽ giúp tiết kiệm đáng kể sức mạnh tính toán cũng như thời gian tải.

CHƯƠNG 3 PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

3.1. XÁC ĐỊNH TÁC NHÂN HỆ THỐNG

3.1.1. Tác nhân hệ thống

- Người dùng thường

3.1.2. Tác nhân và chức năng tương ứng

Bảng 3.1. Bảng mô tả thông tin về tác nhân và các chức năng tương ứng

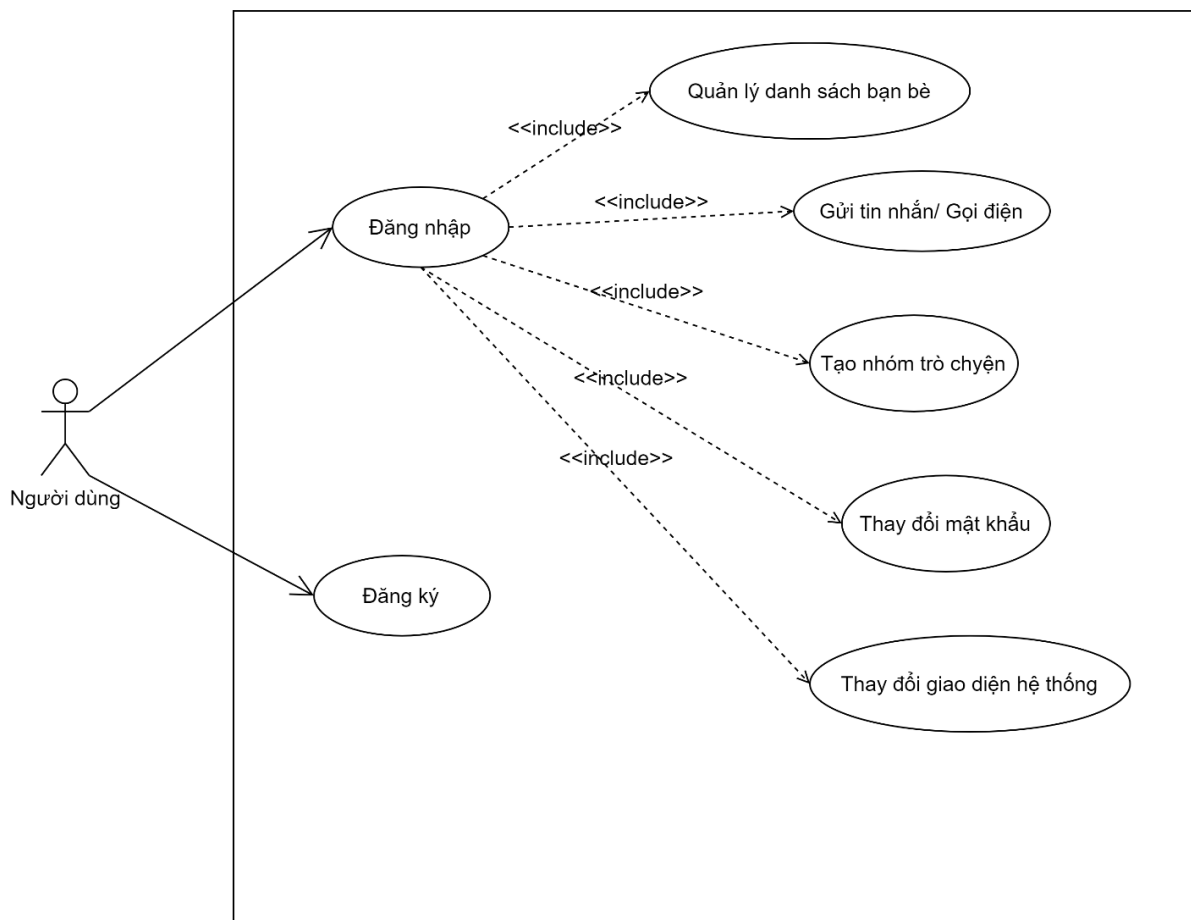
STT	Tác nhân	Chức năng	Ghi chú
1	Người dùng	<ul style="list-style-type: none">• Truy cập• Đăng nhập.• Đăng ký.• Thêm bạn mới.• Xóa một hay nhiều người trong danh sách bạn.• Gửi message/Video Call tới một người dùng khác.• Lập nhóm trò chuyện với nhiều bạn bè.• Thay đổi mật khẩu đăng nhập của tài khoản.• Thay đổi giao diện của hệ thống	Người dùng

3.2. YÊU CẦU PHI CHỨC NĂNG

- Hệ thống dễ dàng sử dụng, thân thiện với người dùng
- Cho phép một lượng lớn người truy cập và cùng một lúc
- Dễ dàng bảo trì, thời gian bảo trì không quá lâu tránh gây ảnh hưởng tới người dùng
- Tốc độ tải trang, phản hồi nhanh
- Sử dụng được trên nhiều nền tảng khác nhau

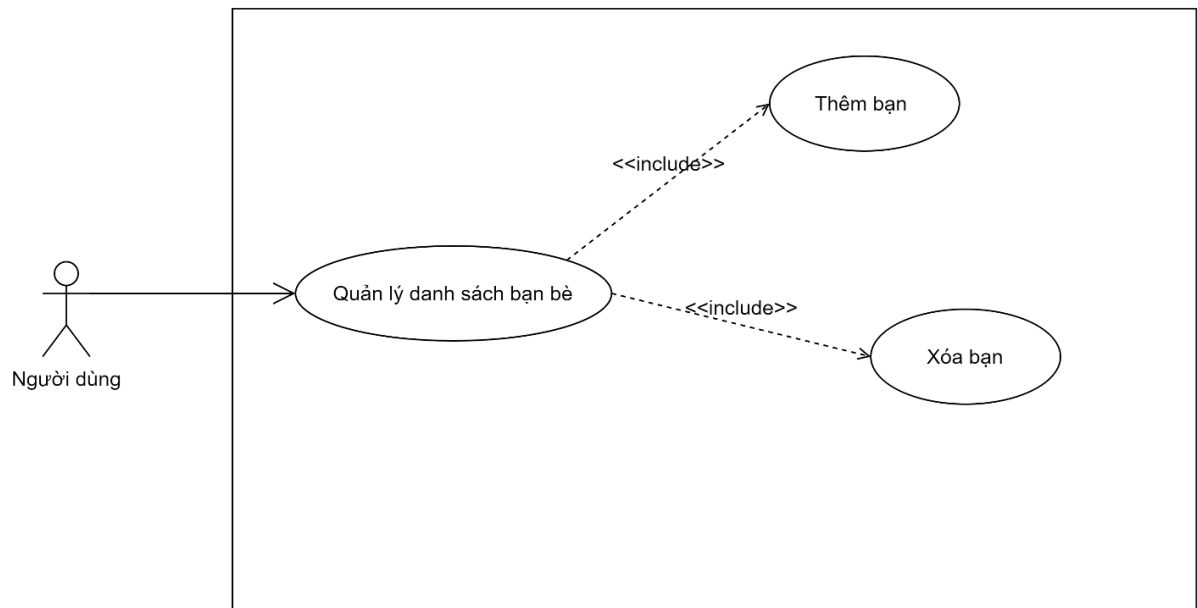
3.3. SƠ ĐỒ USECASE

3.3.1. Sơ đồ usecase tổng quát

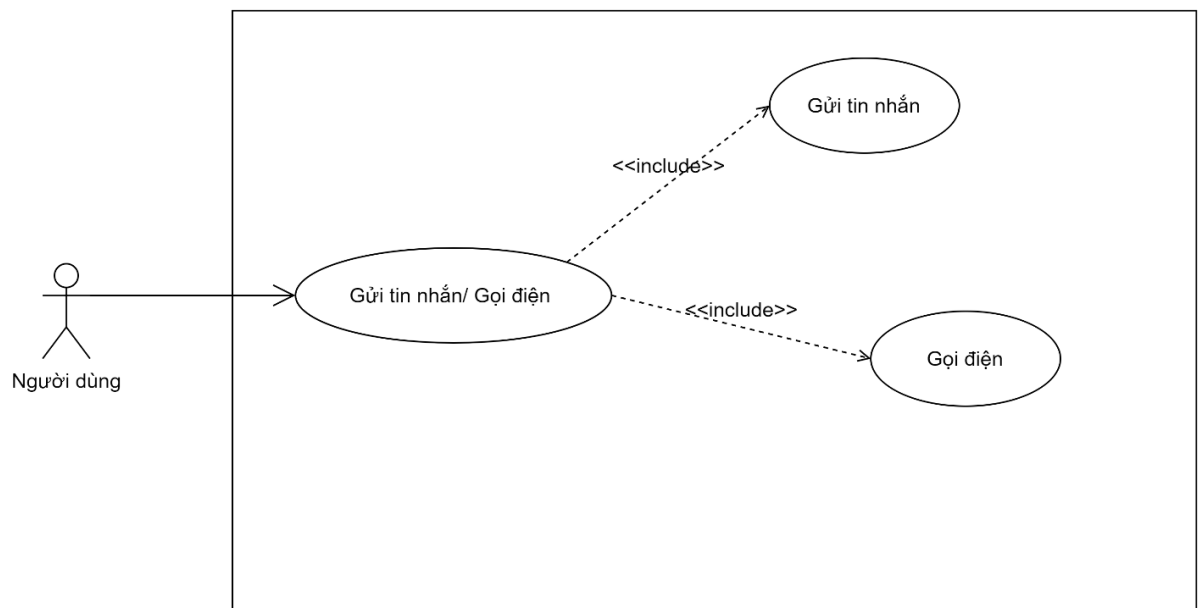


Hình 3.1. Sơ đồ usecase tổng quát

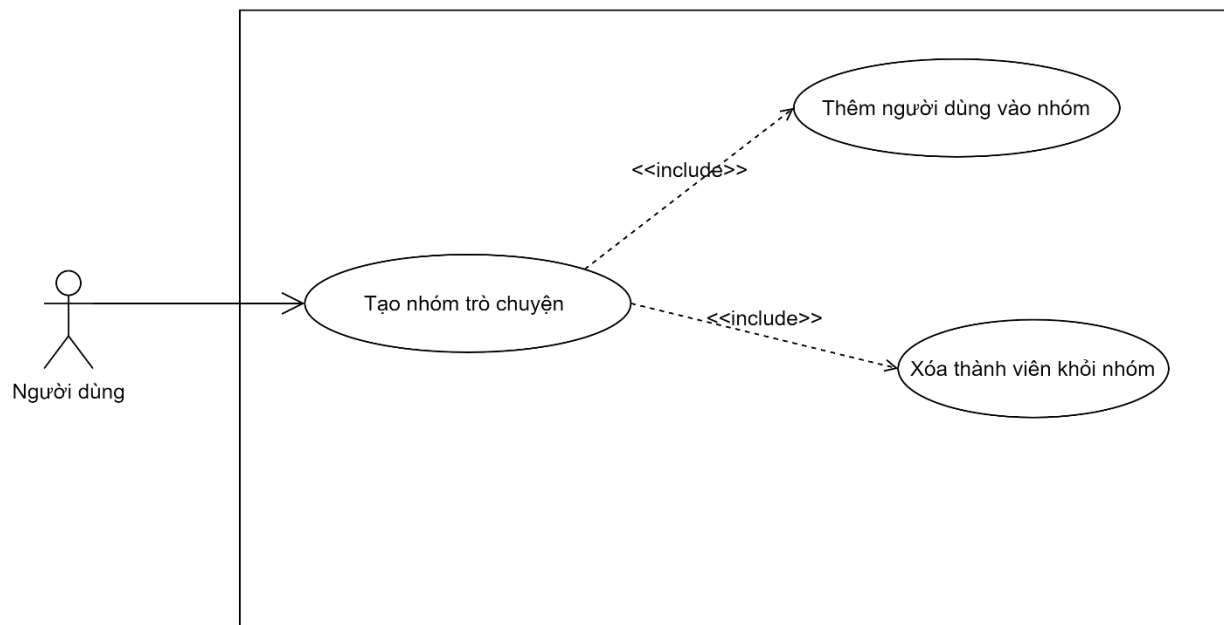
3.3.2. Sơ đồ usecase phân rã



Hình 3.2. Sơ đồ usecase quản lý danh sách bạn bè



Hình 3.3. Sơ đồ usecase gửi tin nhắn, gọi điện

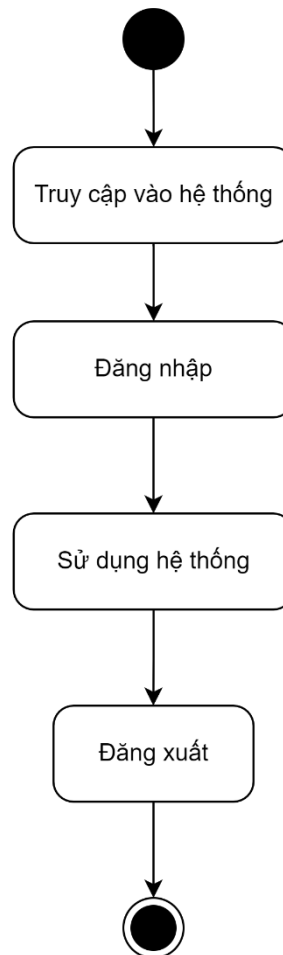


Hình 3.4 Sơ đồ usecase tạo nhóm trò chuyện

3.4. QUY TRÌNH NGHIỆP VỤ

3.4.1. Quy trình truy cập hệ thống

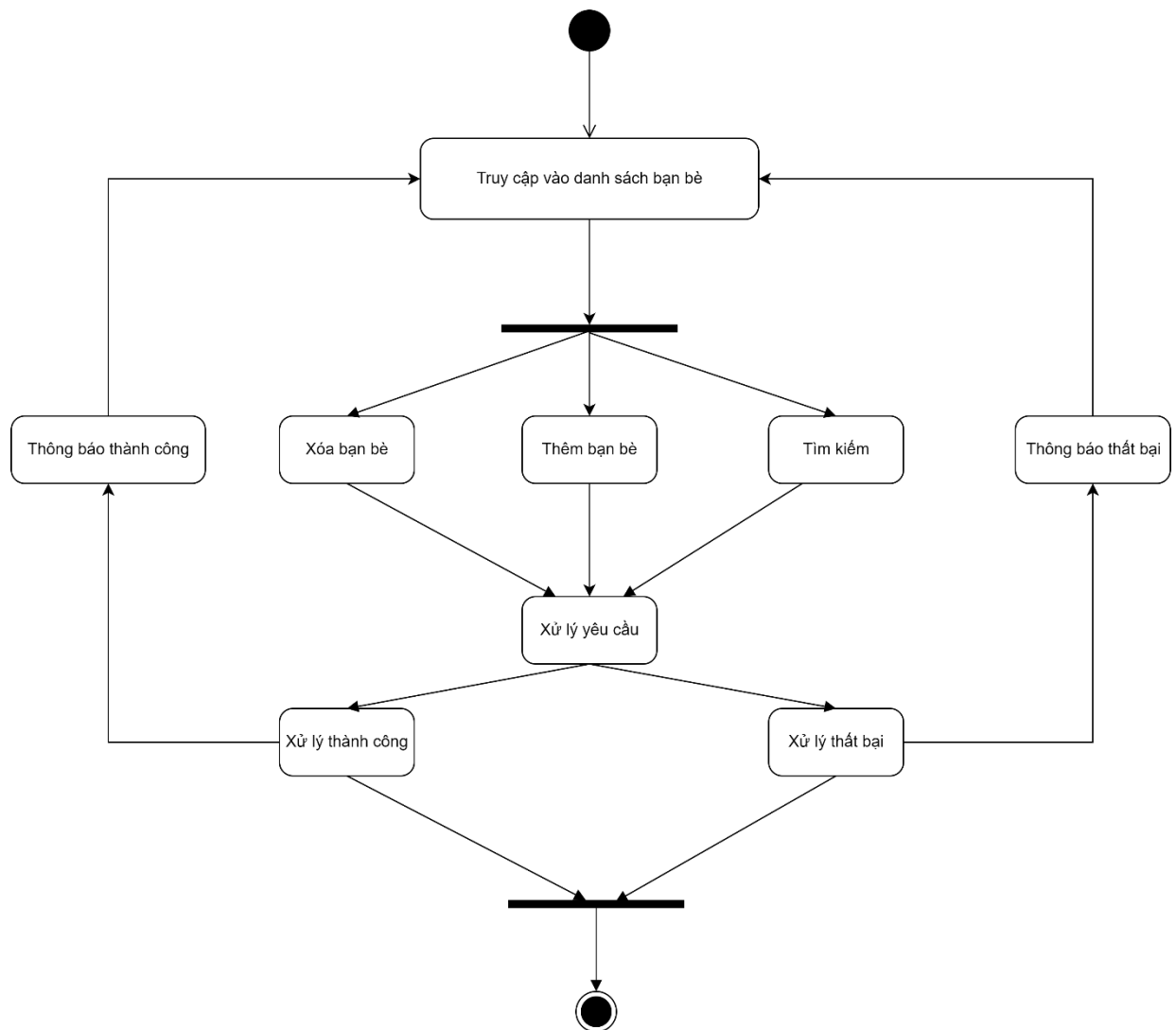
Người sử dụng cần phải đăng nhập vào hệ thống để có thể sử dụng các chức năng của hệ thống dựa theo quyền được hệ thống cấp.



Hình 3.5 Biểu đồ hoạt động sử dụng phần mềm

3.4.2. Quy trình quản lý bạn bè

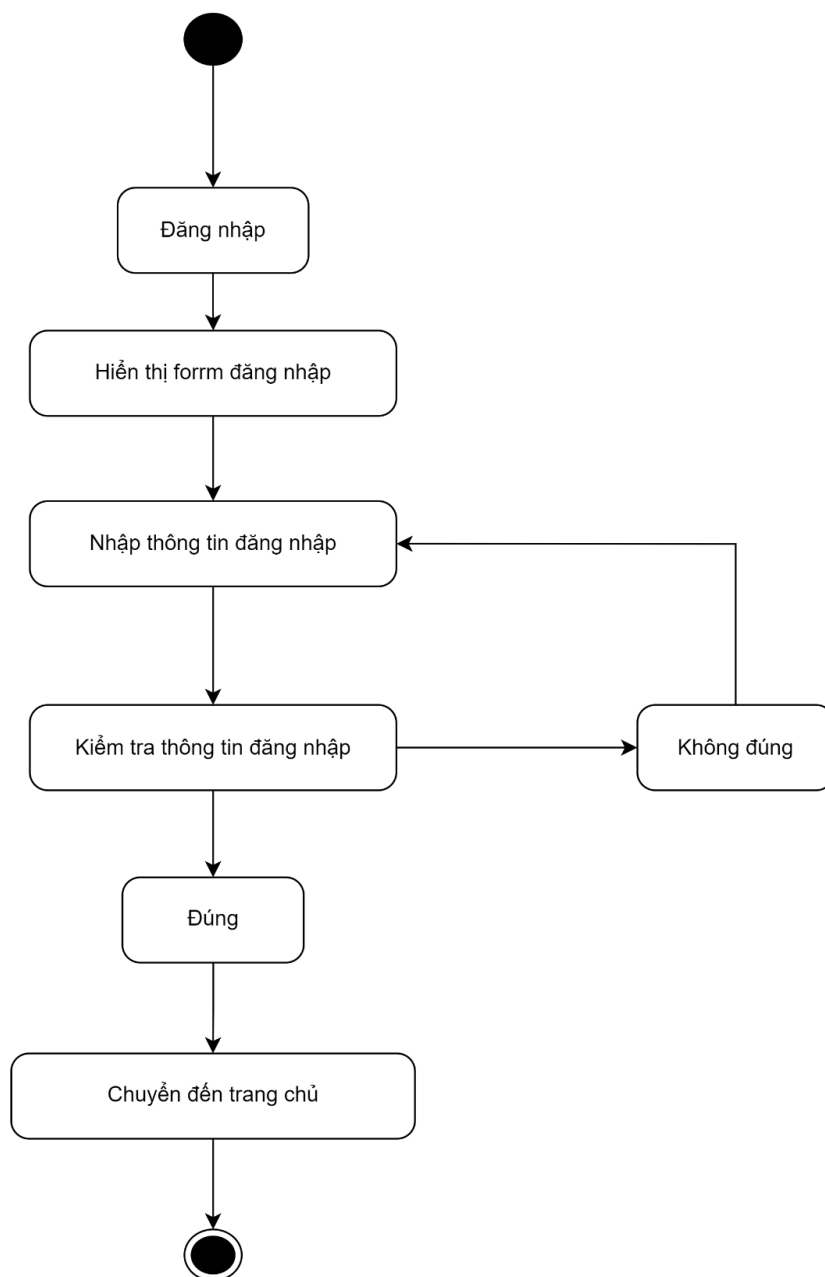
Người dùng truy cập trang danh sách bạn bè và thực hiện các hành động như thêm bạn bè, xóa bạn bè, tìm kiếm.



Hình 3.6 Biểu đồ hoạt động quản lý bạn bè

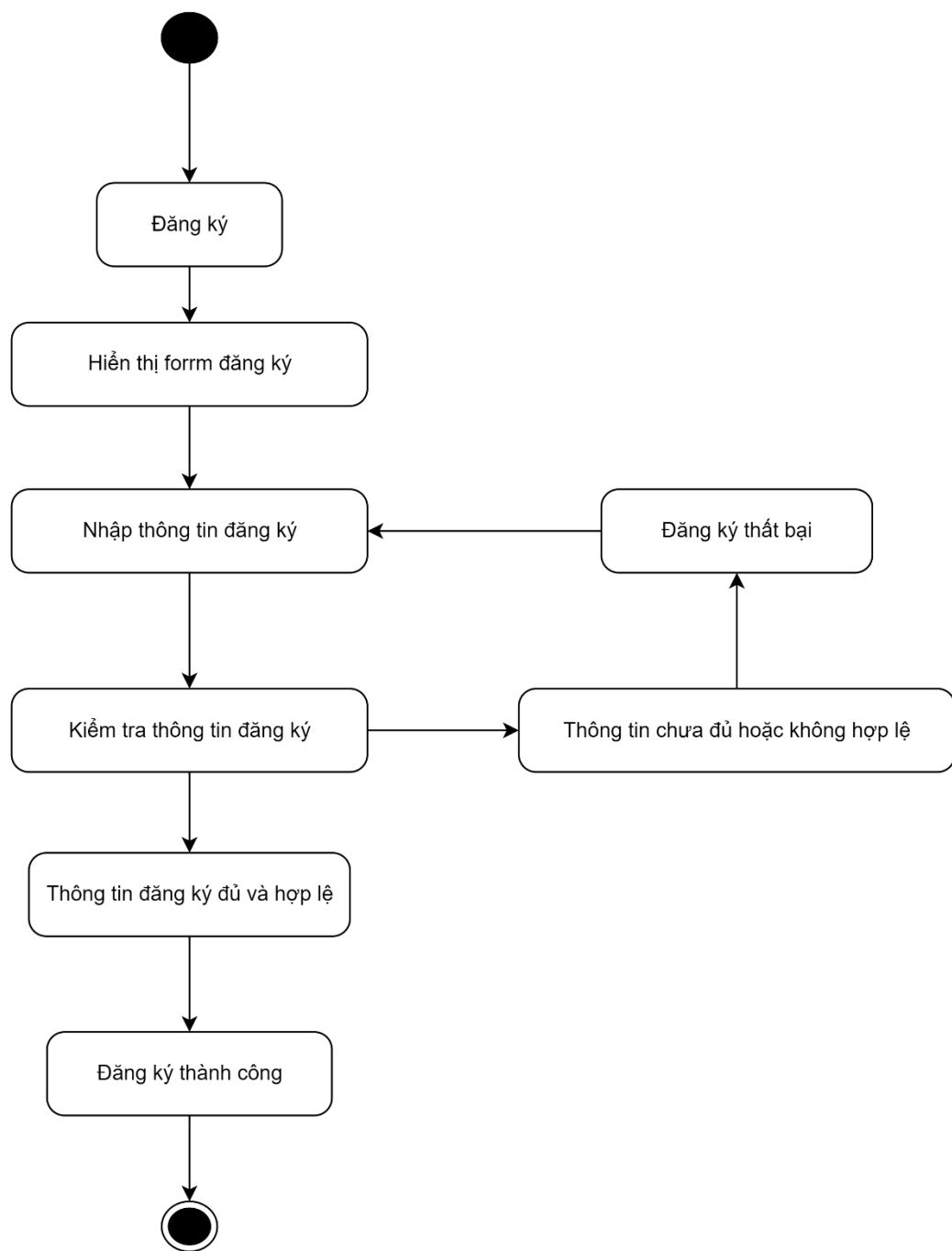
3.4.3. Quy trình đăng nhập

Người dùng bắt buộc phải đăng nhập để có thể thao tác các chức năng trên web.



Hình 3.7 Biểu đồ hoạt động chức năng đăng nhập

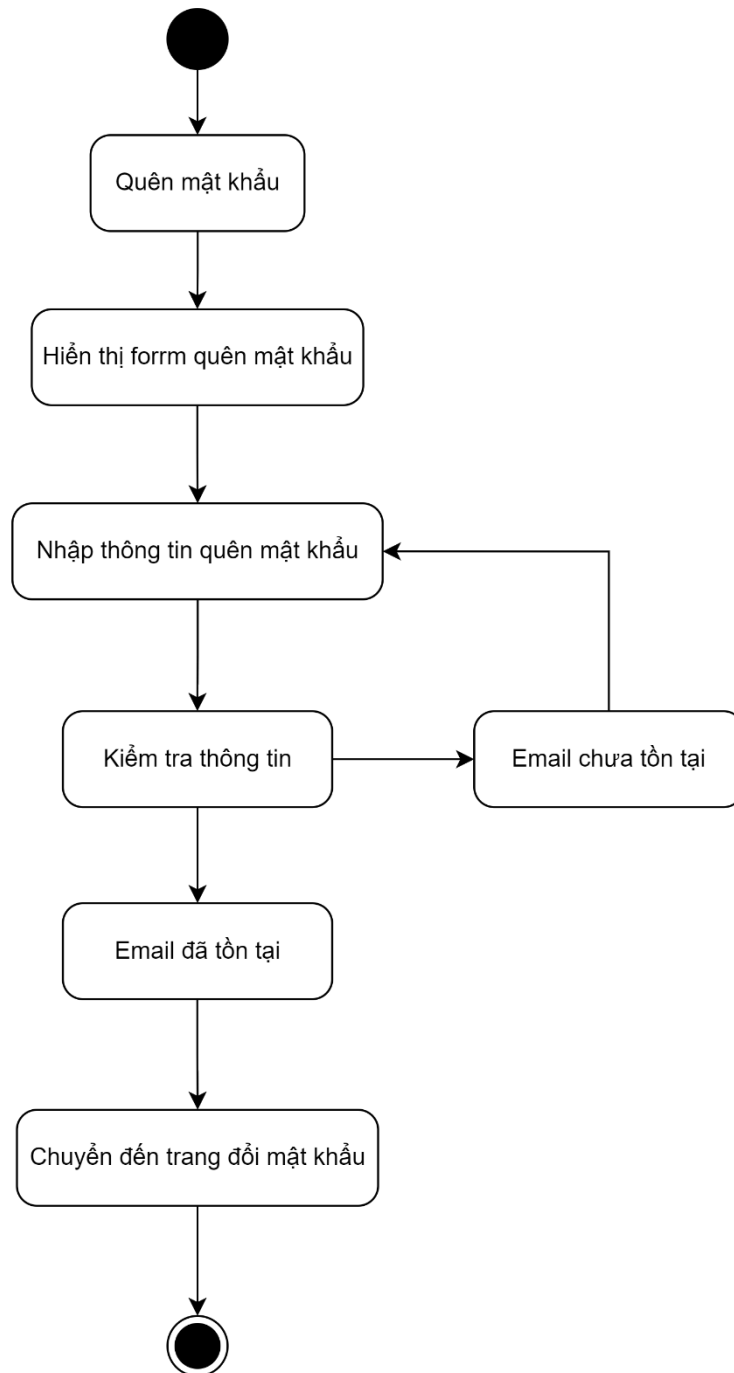
3.4.4. Quy trình đăng ký



Hình 3.8 Biểu đồ hoạt động chức năng đăng ký

3.4.5. Quy trình quên mật khẩu

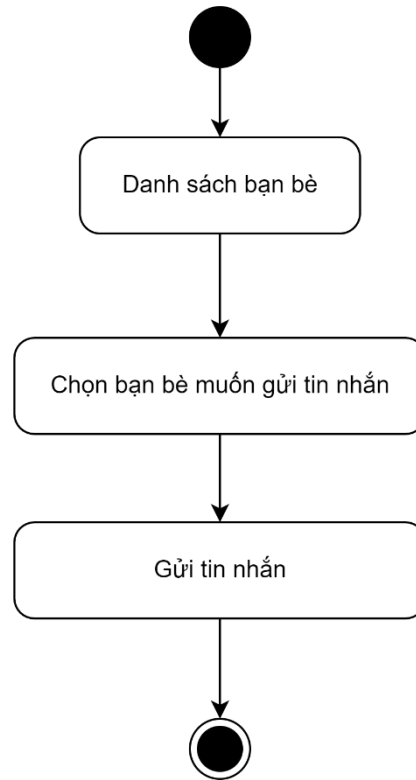
Tất cả người dùng đều có thể dùng chức năng quên mật khẩu để lấy lại mật khẩu



Hình 3.9 Biểu đồ hoạt động chức năng quên mật khẩu

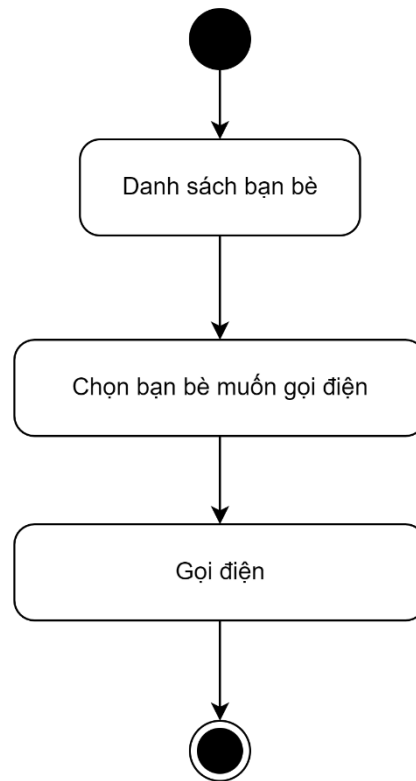
3.4.6. Quy trình nhắn tin

Người dùng nhắn tin với người dùng khác



Hình 3.10 Biểu đồ hoạt động chức năng nhắn tin

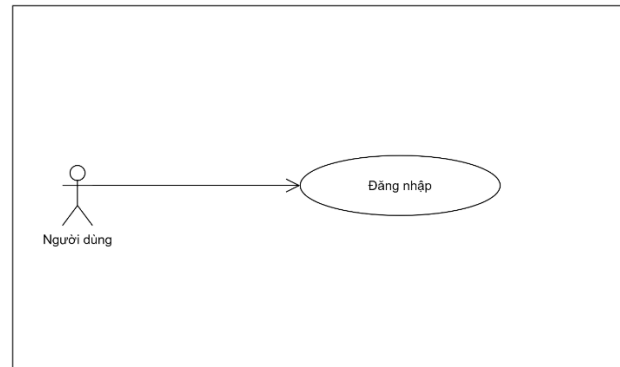
3.4.7. Quy trình gọi điện



Hình 3.11. Biểu đồ hoạt động chức năng gọi điện

3.5. ĐẶC TẢ YÊU CẦU HỆ THỐNG

3.5.1. Đăng nhập



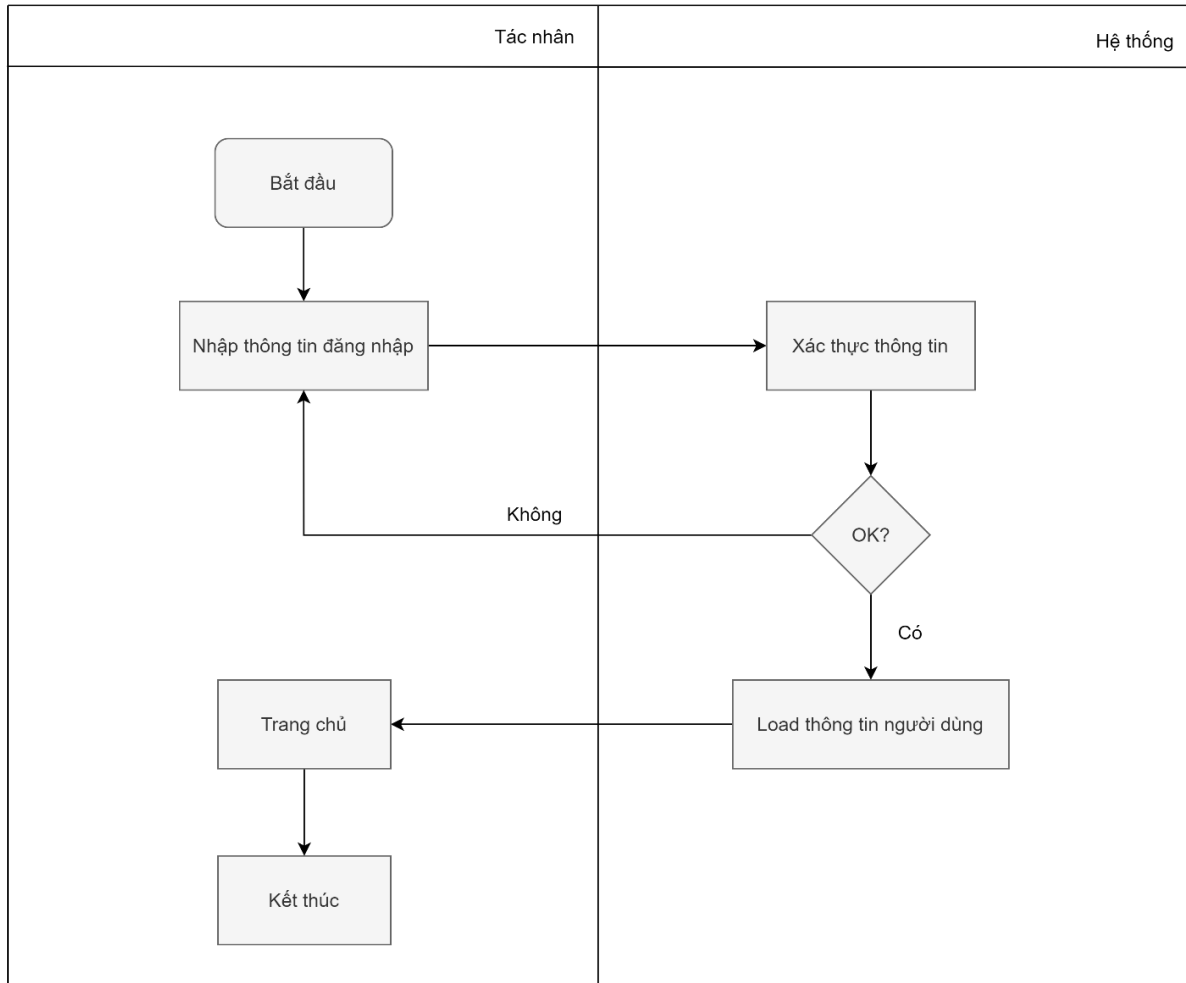
Hình 3.12. Usecase đăng nhập

3.5.1.1. Thông tin chức năng

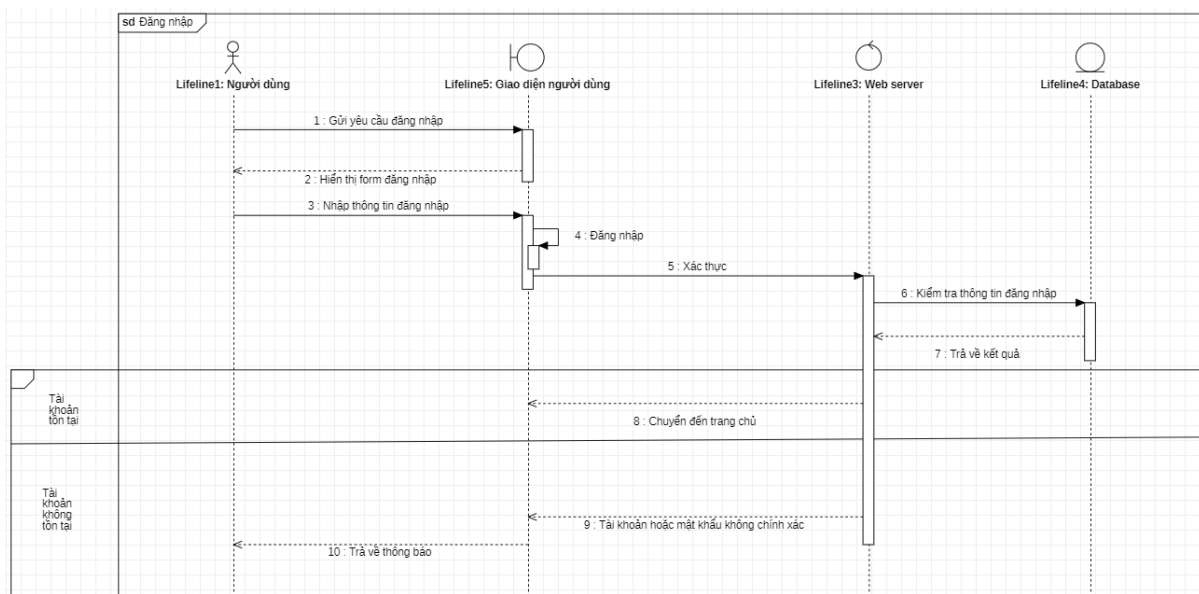
Bảng 3.2. Thông tin chức năng đăng nhập

Tên chức năng	Đăng nhập
Mô tả	Cho phép người dùng đăng nhập vào hệ thống để sử dụng các chức năng của hệ thống.
Tác nhân	Tất cả người dùng hệ thống
Điều kiện trước	N/A
Điều kiện sau	Nếu đăng nhập thành công: Người dùng được phép sử dụng các chức năng của hệ thống. Nếu đăng nhập không thành công: Yêu cầu đăng nhập lại
Ngoại lệ	Sai tên đăng nhập hoặc mật khẩu

3.5.1.2. Biểu đồ chức năng



Hình 3.13. Biểu đồ luồng xử lý chức năng đăng nhập



Hình 3.14. Biểu đồ tuần tự chức năng đăng nhập

3.5.1.3. Mô tả dòng sự kiện chính

Bảng 3.3. Bảng mô tả dòng sự kiện chính chức năng đăng nhập

Hành động của tác nhân	Phản ứng của hệ thống
1. Nhập thông tin đăng nhập: Email, mật khẩu	2. Xác thực thông tin: Kiểm tra thông tin tên người dùng và mật khẩu có tồn tại trên hệ thống hay không?
	3. Load thông tin người dùng: <ul style="list-style-type: none"> - Thông tin người dùng - Đơn vị người dùng - Loại đơn vị người dùng thường

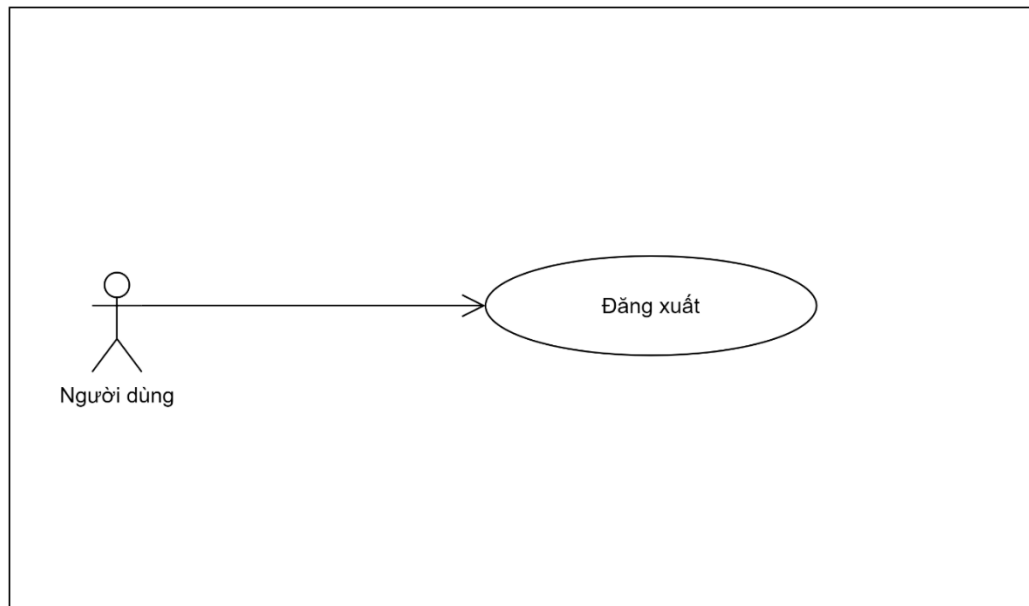
4. Trang chủ: Người dùng bắt đầu thao tác các chức năng trên hệ thống	
---	--

3.5.1.4. Mô tả dòng sự kiện phụ

Bảng 3.4. Bảng mô tả dòng sự kiện phụ chức năng đăng nhập

Hành động của tác nhân	Phản ứng của hệ thống
1. Người dùng chọn “Quên mật khẩu”	3. Chuyển đến màn hình quên mật khẩu
2. Người dùng chọn “Đăng ký ngay”	4. Chuyển đến màn hình đăng ký

3.5.2. Đăng xuất



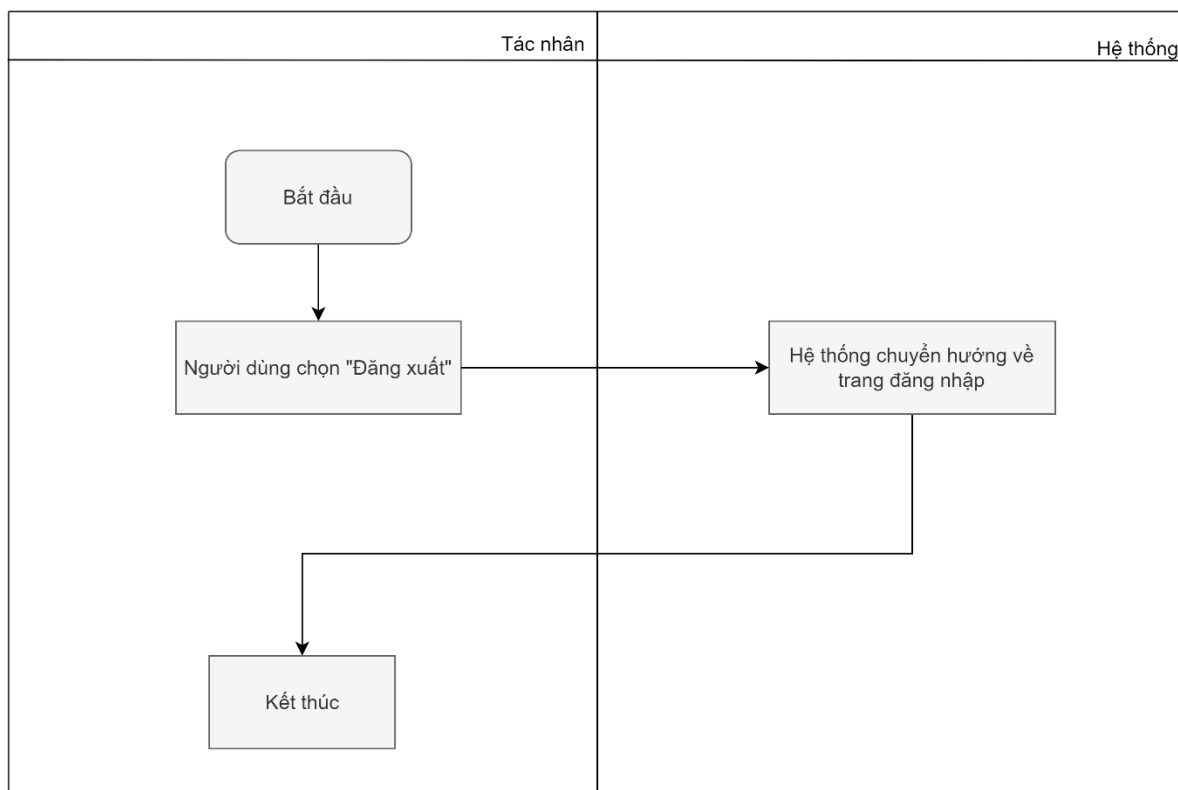
Hình 3.15. Usecase đăng xuất

3.5.2.1. Thông tin chức năng

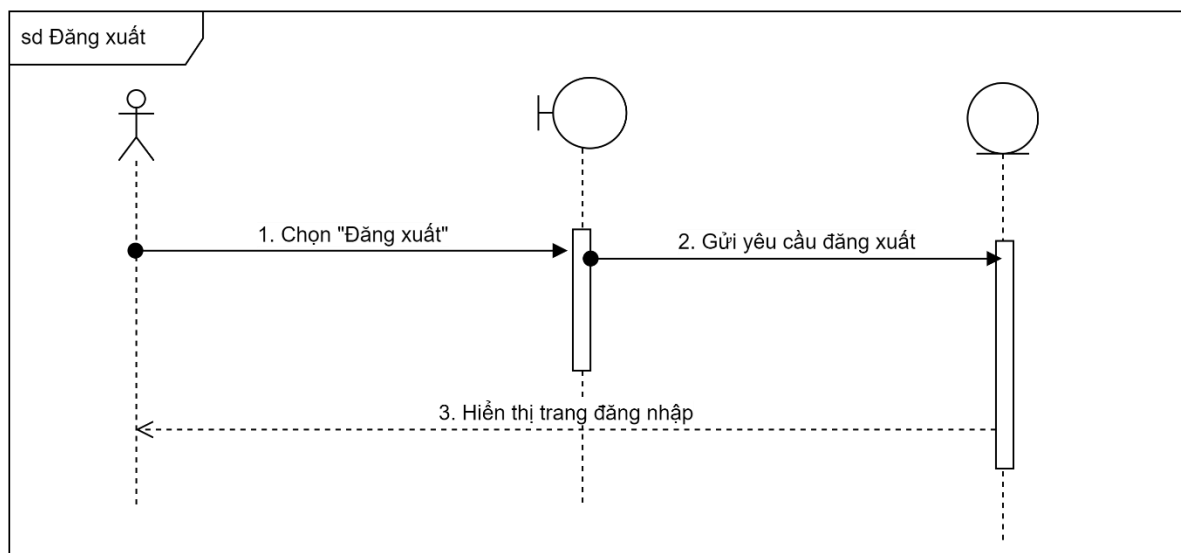
Bảng 3.5. Thông tin chức năng đăng xuất

Tên chức năng	Đăng xuất
Mô tả	Cho phép người dùng đăng xuất khỏi hệ thống.
Tác nhân	Tất cả người dùng hệ thống
Điều kiện trước	Người dùng đăng nhập thành công
Điều kiện sau	Đăng xuất thành công. Người dùng chuyển hướng về trang đăng nhập
Ngoại lệ	N/A

3.5.2.2. Biểu đồ chức năng



Hình 3.16. Biểu đồ luồng xử lý chức năng đăng xuất



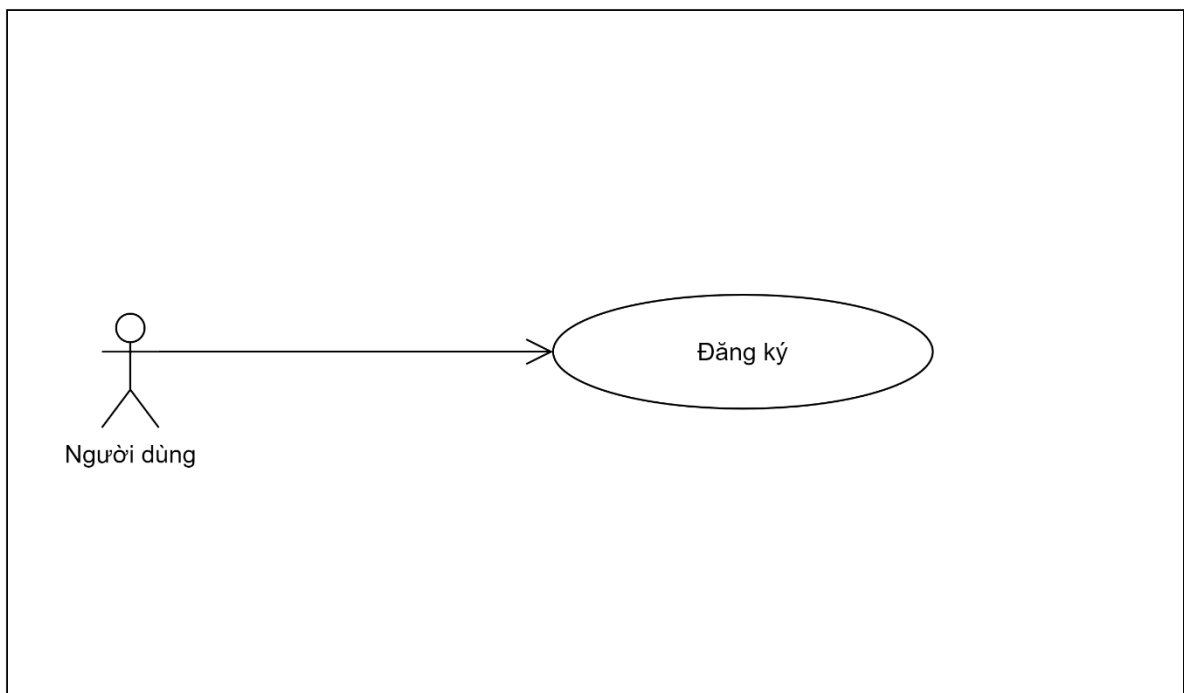
Hình 3.17. Biểu đồ tuần tự chức năng đăng xuất

3.5.2.3. Mô tả dòng sự kiện chính

Bảng 3.6. Bảng mô tả dòng sự kiện chính chức năng đăng xuất

Hành động của tác nhân	Phản ứng của hệ thống
1. Người dùng chọn đăng xuất	2. Hệ thống chuyển hướng về trang đăng nhập

3.5.3. Đăng ký



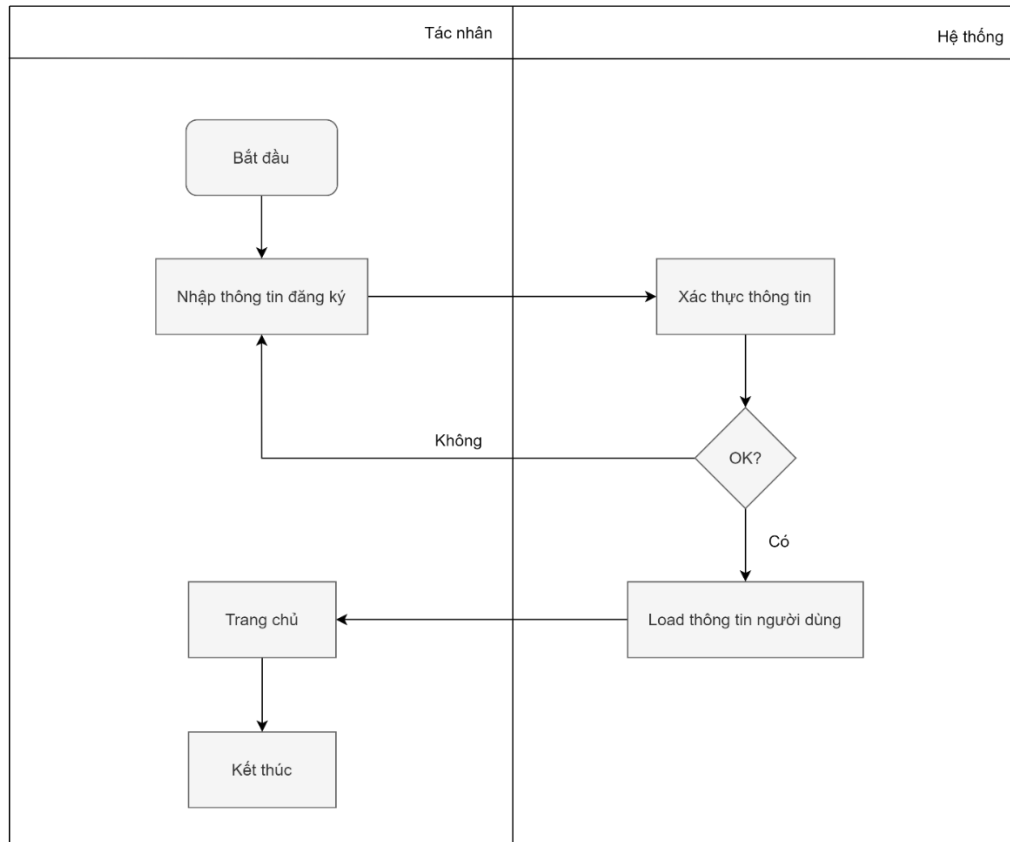
Hình 3.18. Usecase đăng ký

3.5.3.1. Thông tin chức năng

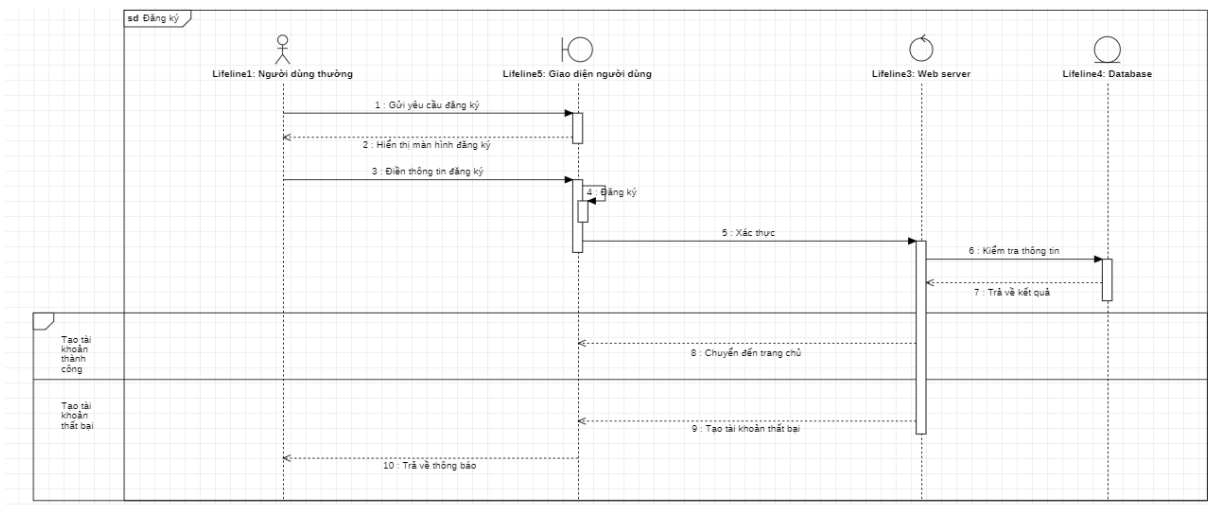
Bảng 3.7. Bảng thông tin chức năng đăng ký

Tên chức năng	Đăng ký
Mô tả	Chức năng này dùng để đăng ký tài khoản khi sử dụng trang web
Tác nhân	Người dùng.
Điều kiện trước	N/A
Điều kiện sau	<ul style="list-style-type: none">- Nếu đăng ký thành công: người dùng được chuyển đến trang đăng nhập.- Nếu đăng ký không thành công: Yêu cầu nhập lại các thông tin bị sai.
Ngoại lệ	<ul style="list-style-type: none">- Tên người dùng đã được sử dụng- Mật khẩu không đủ an toàn (độ dài từ 8 đến 16 ký tự, gồm các ký tự số, chữ, ký tự đặc biệt)- Họ tên quá dài (tối thiểu 50 ký tự)

3.5.3.2. Biểu đồ xử lý chức năng



Hình 3.19 Biểu đồ luồng xử lý chức năng đăng ký



Hình 3.20 Biểu đồ tuần tự chức năng đăng ký

3.5.3.3. Mô tả dòng sự kiện chính

Bảng 3.8. Bảng mô tả dòng sự kiện chính chức năng đăng ký

Hành động của tác nhân	Phản ứng của hệ thống
1. Nhập thông tin đăng ký: Tên người dùng, tên đệm, mật khẩu, mật khẩu xác nhận	<p>2. Xác thực thông tin nếu thông tin đạt yêu cầu chuyển đến 3, nếu sai thì chuyển đến 4:</p> <p>Kiểm tra thông tin tên người dùng có tồn tại trên hệ thống hay không?</p> <p>Kiểm tra mật khẩu đã đủ an toàn hay chưa?</p> <p>Kiểm tra mật khẩu và mật khẩu xác nhận có giống nhau hay chưa?</p>
	<p>3. Lưu thông tin người dùng vào bảng users. Chuyển đến trang đăng nhập.</p> <p>4. Hiển thị lỗi tương ứng “Tên người dùng đã được sử dụng”, “Mật khẩu không đủ an toàn”, “Hai mật khẩu phải giống nhau”</p>
5. Trang chủ: Người dùng bắt đầu thao tác các chức năng trên hệ thống	

3.5.3.4. Mô tả dòng sự kiện phụ

Bảng 3.9. Bảng mô tả dòng sự kiện phụ chức năng đăng ký

Hành động của tác nhân	Phản ứng của hệ thống
1. Người dùng chọn “Hủy bỏ”	3. Chuyển đến màn hình trang chủ khi chưa đăng nhập của người dùng thường
2. Người dùng chọn “Đăng nhập ngay”	4. Chuyển đến màn hình đăng nhập

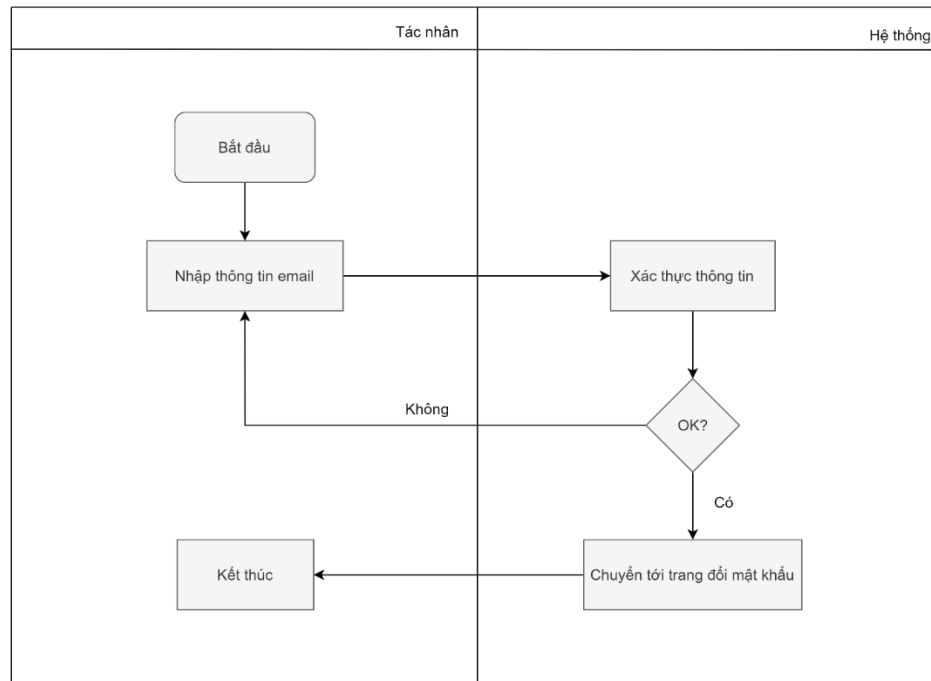
3.5.4. Quên mật khẩu

3.5.4.1. Thông tin chức năng

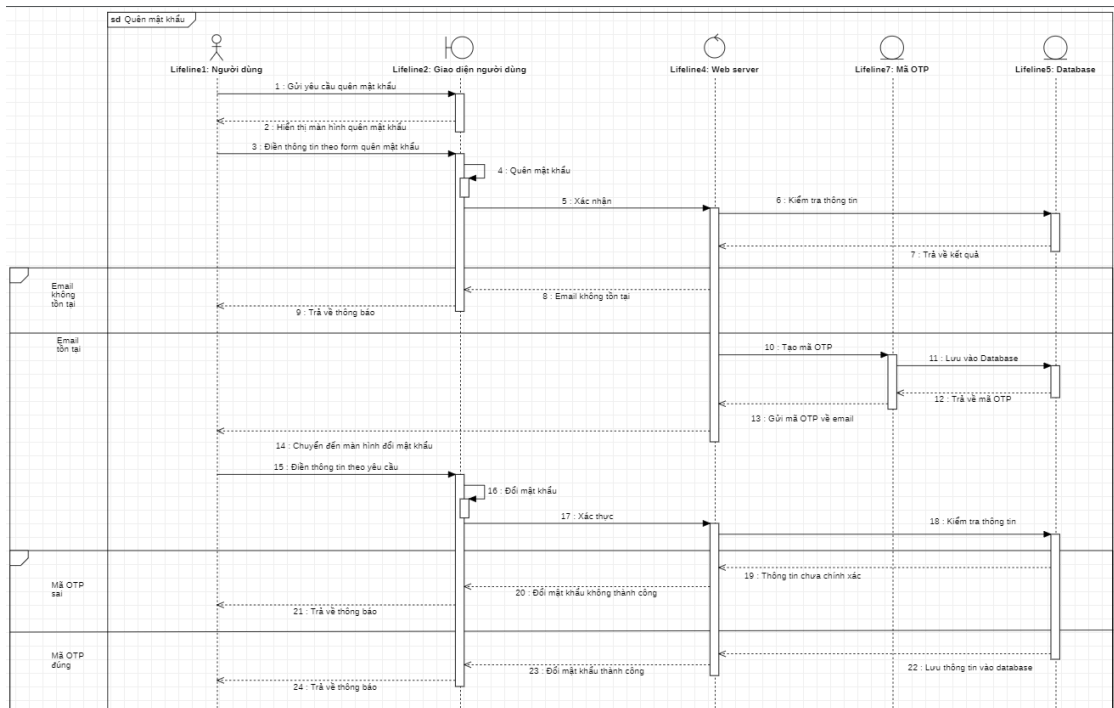
Bảng 3.10. Bảng thông tin chức năng quên mật khẩu

Tên chức năng	Quên mật khẩu
Mô tả	Chức năng này cho phép người dùng lấy mã otp để đổi mật khẩu khi người dùng quên mật khẩu
Tác nhân	Tất cả người dùng hệ thống có tài khoản hệ thống
Điều kiện trước	Người dùng ở trang đăng nhập chọn quên mật khẩu
Điều kiện sau	Chuyển đến màn hình Lấy lại mật khẩu
Ngoại lệ	N/A

3.5.4.2. Biểu đồ xử lý chức năng



Hình 3.21. Biểu đồ luồng xử lý chức năng quên mật khẩu



Hình 3.22. Biểu đồ tuần tự chức năng quên mật khẩu

3.5.4.3. Mô tả dòng sự kiện chính

Bảng 3.11. Bảng mô tả dòng sự kiện chính chức năng quên mật khẩu

Hành động của tác nhân	Phản ứng của hệ thống
1. Nhập email	2. Xác thực thông tin: Kiểm tra thông tin email có tồn tại ở DB chưa? Nếu email đã tồn tại chuyển đến 4, nếu email không tồn tại chuyển đến 3
	3. Hiện thị lỗi “Email không tồn tại” 4. Hiện thị thông báo “Mã otp để đổi mật khẩu đã được gửi vào email, vui lòng truy cập email để lấy mã.”, chuyển đến màn hình lấy lại mật khẩu

3.5.5. Chức năng nhắn tin/ gọi điện

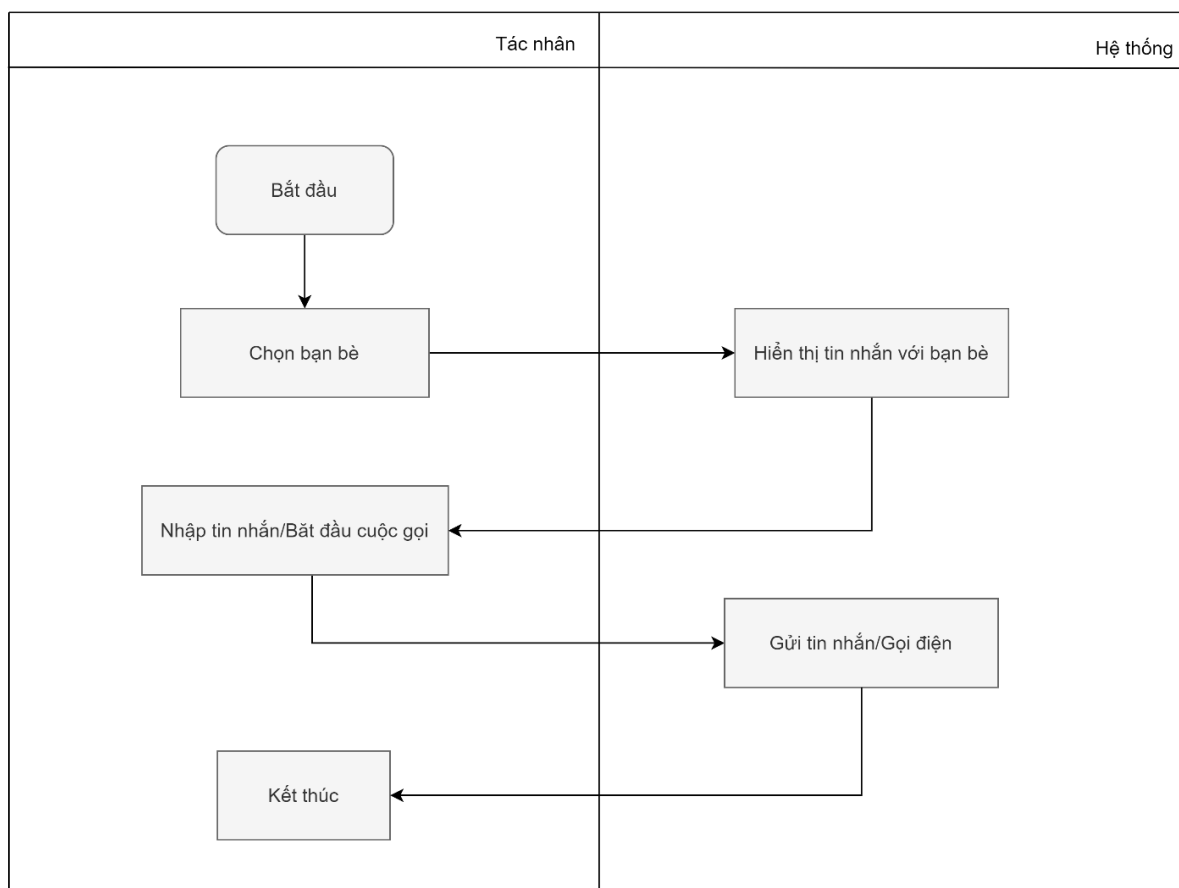
3.5.5.1. Thông tin chức năng

Bảng 3.12. Bảng thông tin chức năng nhắn tin/gọi điện

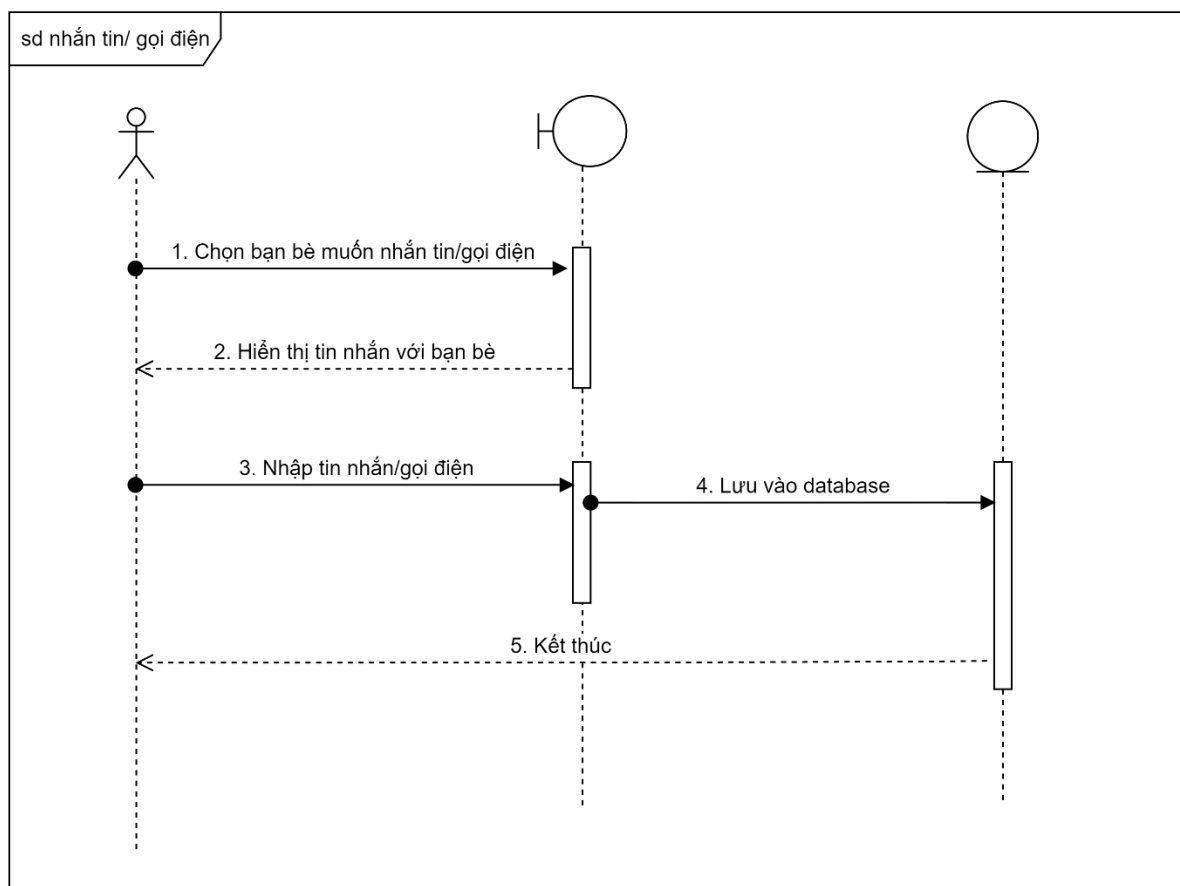
Tên chức năng	Nhắn tin
Mô tả	Chức năng này cho phép người dùng có thể gửi tin nhắn, gọi điện tới người dùng khác
Tác nhân	Người dùng

Điều kiện trước	Đăng nhập
Điều kiện sau	N/A
Ngoại lệ	N/A

3.5.5.2. Biểu đồ luồng xử lý chức năng



Hình 3.23. Biểu đồ luồng xử lý chức năng nhắn tin/gọi điện



Hình 3.24. Biểu đồ tuần tự chức năng nhắn tin/gọi điện

3.5.5.3. Mô tả dòng sự kiện chính

Bảng 3.13. Bảng mô tả dòng sự kiện chính chức năng nhắn tin/gọi điện

Hành động của tác nhân	Phản ứng của hệ thống
1. Chọn bạn bè	2. Hiển thị tin nhắn bạn bè
3. Nhập tin nhắn/gọi điện	4. Gửi tin nhắn/Gọi điện

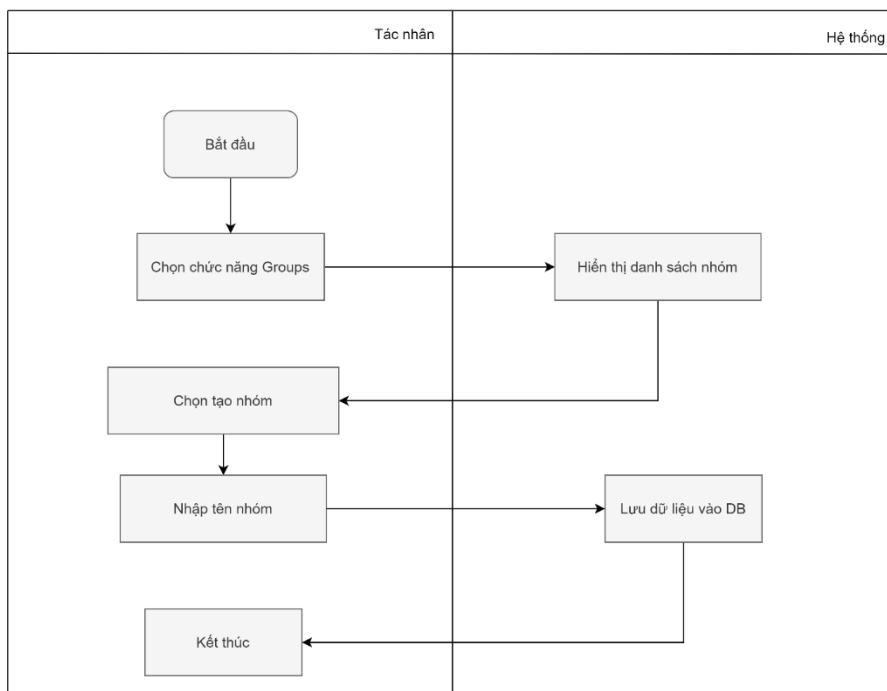
3.5.6. Chức năng tạo nhóm trò chuyện

3.5.6.1. Thông tin chức năng

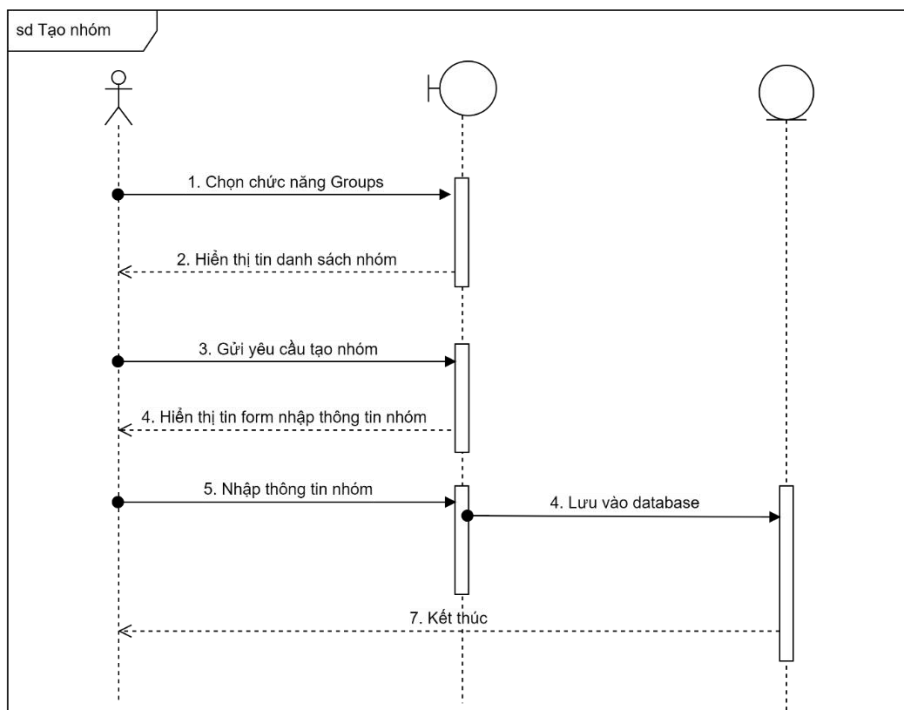
Bảng 3.14. Bảng thông tin chức năng tạo nhóm trò chuyện

Tên chức năng	Tạo nhóm trò chuyện
Mô tả	Chức năng này cho phép người dùng có thể tạo nhóm trò chuyện với nhiều người cùng lúc
Tác nhân	Người dùng
Điều kiện trước	Đăng nhập
Điều kiện sau	N/A
Ngoại lệ	N/A

3.5.6.2. Biểu đồ xử lý chức năng



Hình 3.25. Biểu đồ luồng xử lý chức năng tạo nhóm trò chuyện



Hình 3.26. Biểu đồ tuần tự chức năng tạo nhóm trò chuyện

3.5.6.3. Mô tả dòng sự kiện chính

Bảng 3.15. Bảng mô tả dòng sự kiện chính chức năng tạo nhóm trò chuyện

Hành động của tác nhân	Phản ứng của hệ thống
1. Vào chức năng Groups	2. Hiện thị danh sách nhóm
3. Chọn tạo nhóm mới 4. Nhập tên nhóm	5. Lưu dữ liệu vào DB

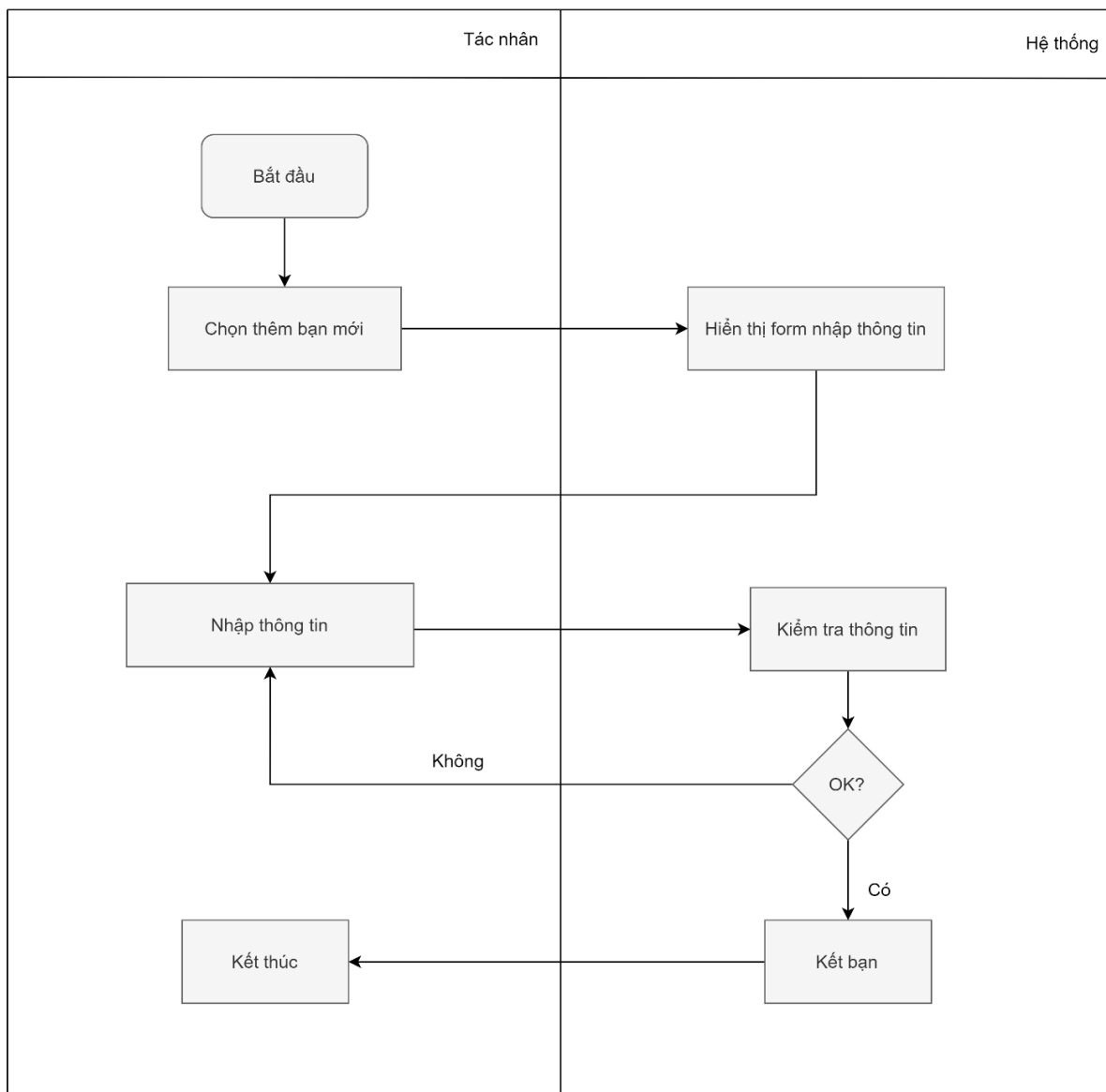
3.5.7. Chức năng thêm bạn bè

3.5.7.1. Thông tin chức năng

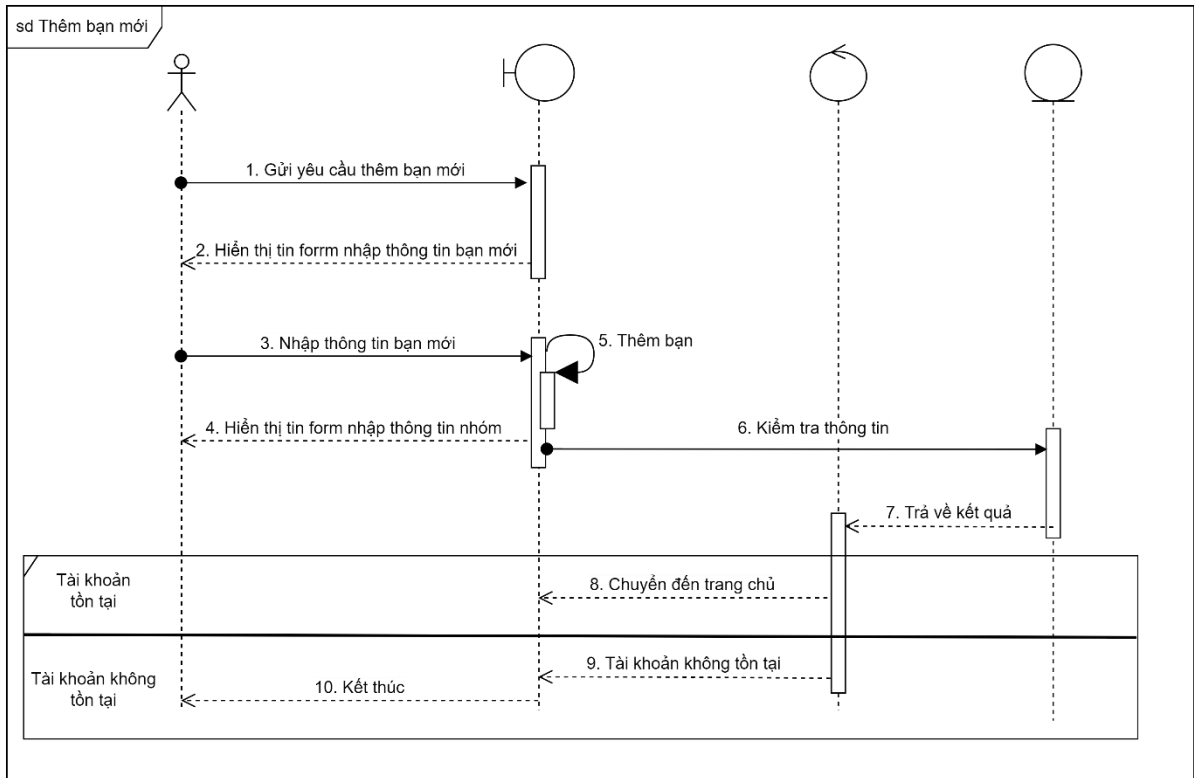
Bảng 3.16. Bảng thông tin chức năng thêm bạn bè

Tên chức năng	Thêm bạn bè
Mô tả	Chức năng này cho phép người dùng kết bạn với mọi người
Tác nhân	Người dùng
Điều kiện trước	Đăng nhập
Điều kiện sau	Kết bạn thành công
Ngoại lệ	Sai tên bạn bè

3.5.7.2. Biểu đồ xử lý chức năng



Hình 3.27. Biểu đồ luồng xử lý chức năng thêm bạn bè



Hình 3.28. Biểu đồ tuần tự chức năng thêm bạn bè

3.5.7.3. Mô tả dòng sự kiện chính

Bảng 3.17. Bảng mô tả dòng sự kiện chính chức năng thêm bạn bè

Hành động của tác nhân	Phản ứng của hệ thống
1. Vào chức năng thêm bạn mới	2. Hiển thị form nhập thông tin
3. Nhập thông tin bạn bè	4. Kiểm tra thông tin: Kiểm tra thông tin người dùng có tồn tại hay không?
5. Kết bạn	

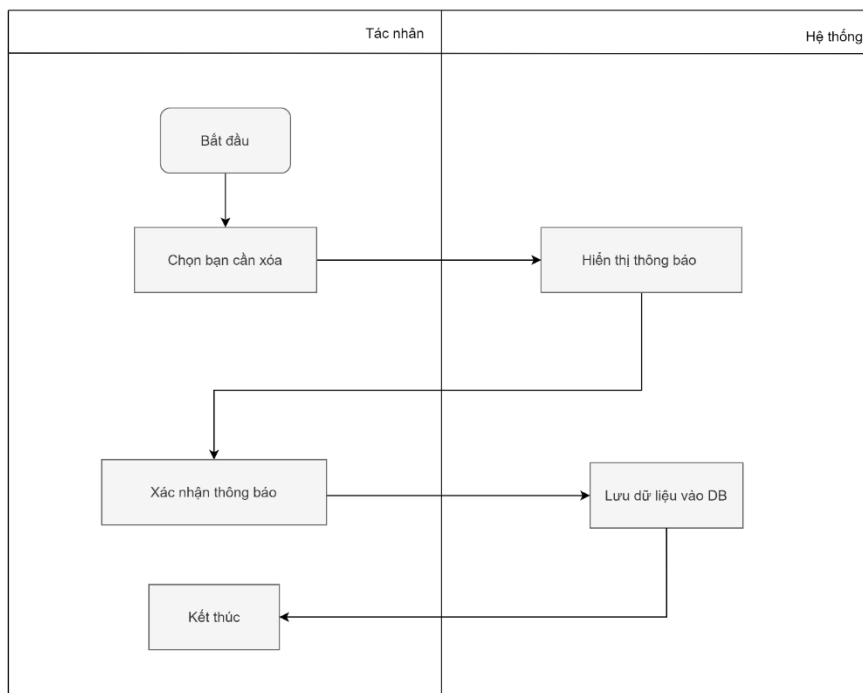
3.5.8. Chức năng xóa bạn bè

3.5.8.1. Thông tin chức năng

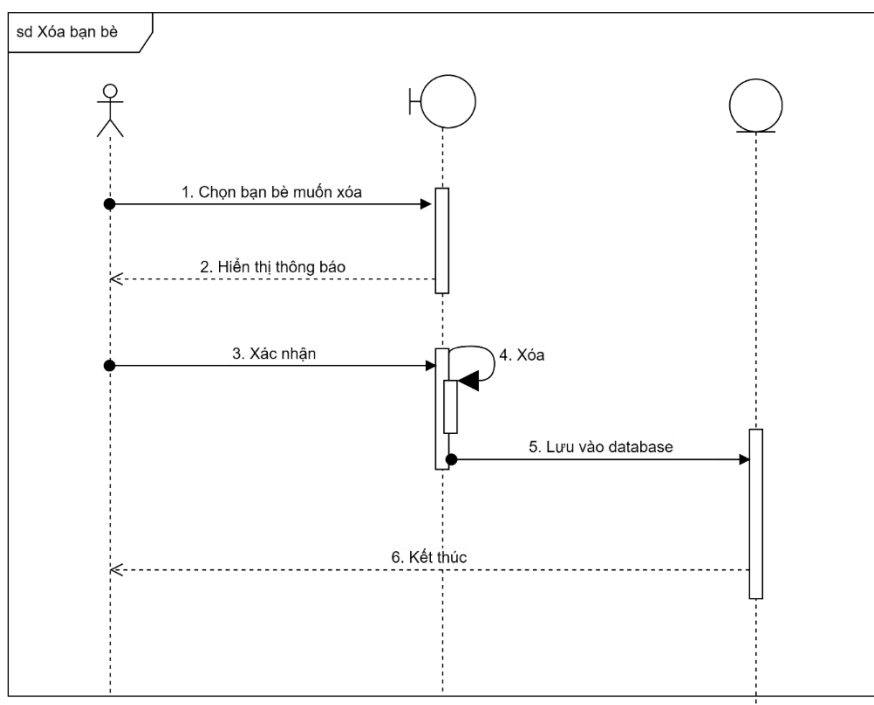
Bảng 3.18. Bảng thông tin chức năng xóa bạn bè

Tên chức năng	Xóa bạn bè
Mô tả	Chức năng này cho phép Xóa bạn bè
Tác nhân	Người dùng
Điều kiện trước	Đăng nhập
Điều kiện sau	Xóa bạn bè thành công
Ngoại lệ	N/A

3.5.8.2. Biểu đồ xử lý chức năng



Hình 3.29. Biểu đồ luồng xử lý chức năng xóa bạn bè



Hình 3.30. Biểu đồ tuần tự chức năng xóa bạn bè

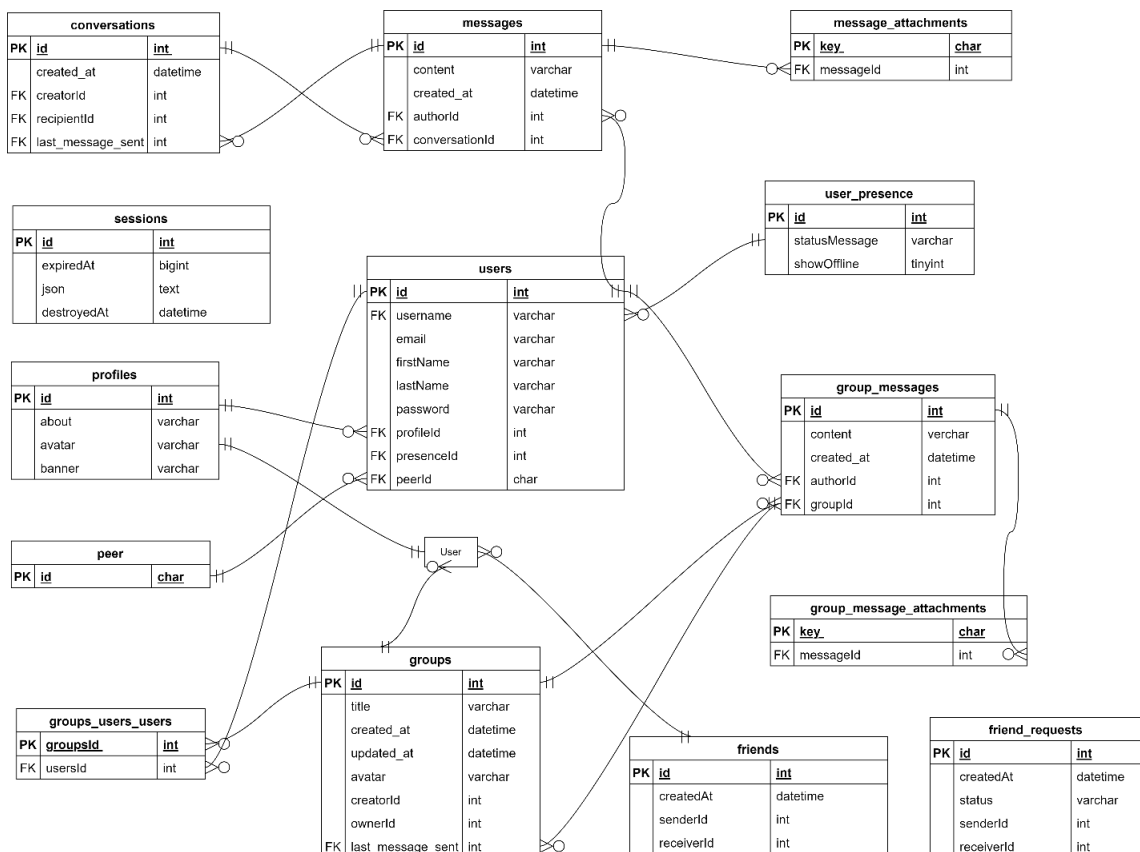
3.5.8.3. Mô tả dòng sự kiện chính

Bảng 3.19. Bảng mô tả dòng sự kiện chính chức năng danh sách tin tuyển dụng

Hành động của tác nhân	Phản ứng của hệ thống
1. Chọn bạn bè	2. Hiện thị thông báo có xác nhận xóa hay không
3. Xác nhận xóa bạn bè	4. Lưu dữ liệu vào DB

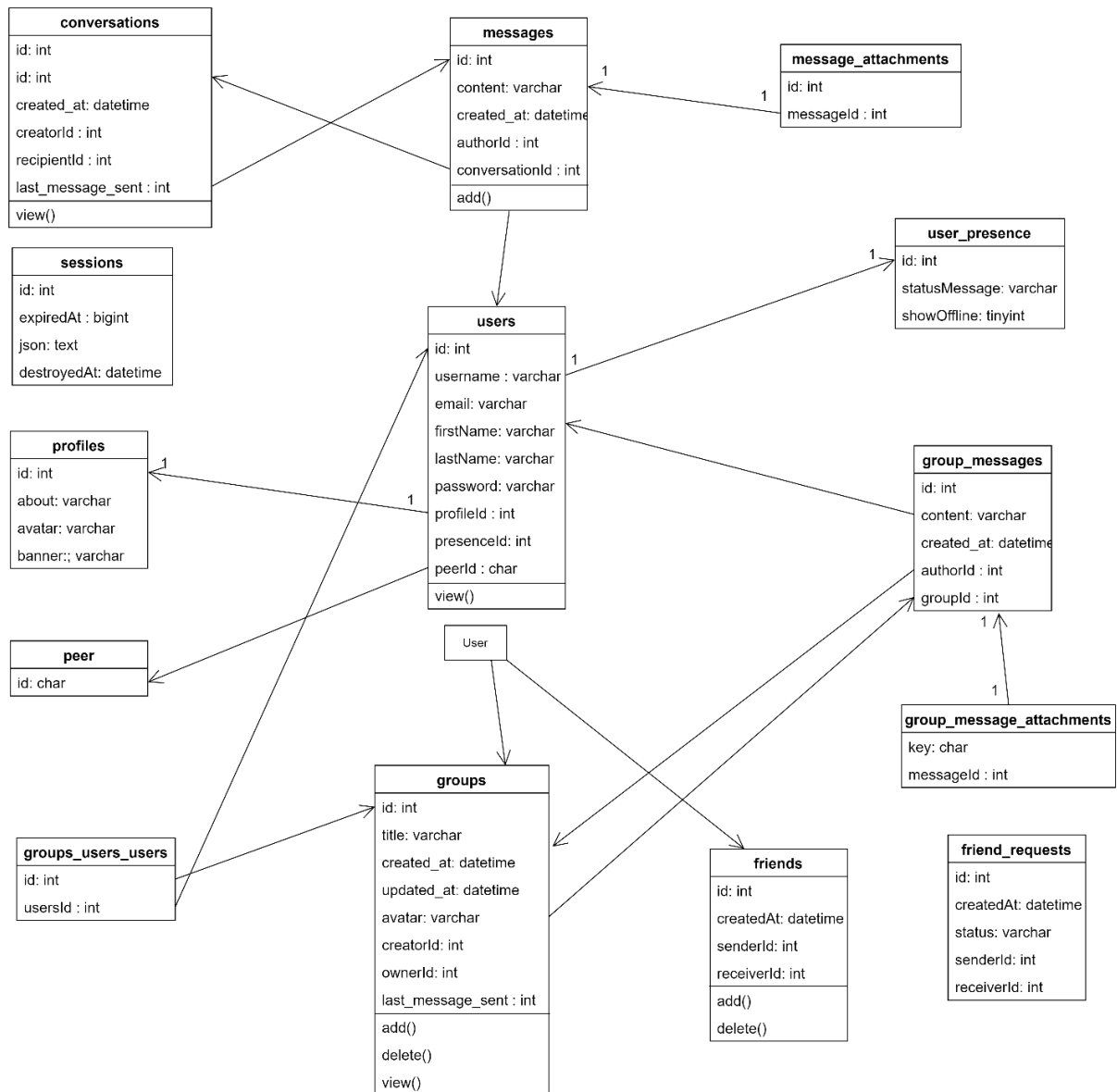
3.6. BIỂU ĐỒ LỚP THỰC THỂ

3.6.1. Biểu đồ thực thể liên kết (ERD)



Hình 3.31. Biểu đồ thực thể liên kết

3.6.2. Biểu đồ lớp



Hình 3.32. Biểu đồ lớp

3.7. THIẾT KẾ CƠ SỞ DỮ LIỆU

3.7.1. Chi tiết bảng conversations

Bảng 3.20. Bảng conversations

Thuộc tính	Kiểu dữ liệu	Nullable	Mô tả
ID	int	No	ID
created_at	datetime	No	Ngày tạo
updated_at	datetime	No	Ngày cập nhật
creatorId	int	Yes	Id người gửi
recipientId	int	Yes	Id người nhận
last_message_sent	int	Yes	Số tin đã gửi

3.7.2. Chi tiết bảng friends

Bảng 3.21. Bảng friends

Thuộc tính	Kiểu dữ liệu	Nullable	Mô tả
ID	int	No	ID
createdAt	datetime	No	Ngày kết bạn
senderId	int	Yes	Id người gửi
receiverId	int	Yes	Id người nhận

3.7.3. Chi tiết bảng friend_requests

Bảng 3.22. Bảng friend_requests

Thuộc tính	Kiểu dữ liệu	Nullable	Mô tả
ID	int	No	ID
Status	varchar	No	Trạng thái
createdAt	datetime	No	Ngày kết bạn
senderId	int	Yes	Id người gửi
receiverId	int	Yes	Id người nhận

3.7.4. Chi tiết bảng groups

Bảng 3.23. Bảng groups

Thuộc tính	Kiểu dữ liệu	Nullable	Mô tả
ID	int	No	ID
Title	Varchar	Yes	Tên nhóm
created_at	datetime	No	Ngày tạo
updated_at	datetime	No	Ngày chỉnh sửa
avatar	Varchar	Yes	Ảnh nhóm
creatorId	Int	Yes	Id người tạo
ownerId	Int	Yes	Id chủ sở hữu
last_message_sent	int	Yes	Số tin đã gửi

3.7.5. Chi tiết bảng groups_users_users

Bảng 3.24. Bảng groups_users_users

Thuộc tính	Kiểu dữ liệu	Nullable	Mô tả
groupsId	Int	No	Id nhóm
usersId	int	No	Id người dùng

3.7.6. Chi tiết bảng group_messages

Bảng 3.25. Bảng group_messages

Thuộc tính	Kiểu dữ liệu	Nullable	Mô tả
ID	int	No	ID
Content	Varchar	Yes	Nội dung
created_at	Datetime	No	Ngày tạo
authorId	Number	Yes	Id người gửi
groupId	Number	Yes	Id nhóm

3.7.7. Chi tiết bảng group_message_attachments

Bảng 3.26. Bảng group_message_attachments

Thuộc tính	Kiểu dữ liệu	Nullable	Mô tả
key	Char	No	Khóa
messageId	int	Yes	Id tin nhắn

3.7.8. Chi tiết bảng messages

Bảng 3.27. Bảng messages

Thuộc tính	Kiểu dữ liệu	Nullable	Mô tả
id	Int	No	Id
content	Varchar	Yes	Nội dung
created_at	Datetime	No	Ngày nhắn
authorId	Int	Yes	Id chủ sở hữu
conversationId	int	Yes	Id hội thoại

3.7.9. Chi tiết bảng message_attachments

Bảng 3.28. Bảng message_attachments

Thuộc tính	Kiểu dữ liệu	Nullable	Mô tả
key	Char	No	Khóa
messageId	int	Yes	Id tin nhắn

3.7.10. Chi tiết bảng peer

Bảng 3.29. Bảng peer

Thuộc tính	Kiểu dữ liệu	Nullable	Mô tả
ID	Char	No	ID

3.7.11. Chi tiết bảng profiles

Bảng 3.30. Bảng profiles

Thuộc tính	Kiểu dữ liệu	Nullable	Mô tả
ID	Int	No	ID
about	Varchar	No	Mô tả
Avartar	Varchar	Yes	Ảnh đại diện
baner	Varchar	Yes	Ảnh bìa

3.7.12. Chi tiết bảng sessions

Bảng 3.31. Bảng sessions

Thuộc tính	Kiểu dữ liệu	Nullable	Mô tả
expiredAt	bigint	No	Thời gian hết hạn
id	int	No	Id
json	Text	No	Json
destroyedAt	datetime	Yes	Ngày xóa

3.7.13. Chi tiết bảng user_presence

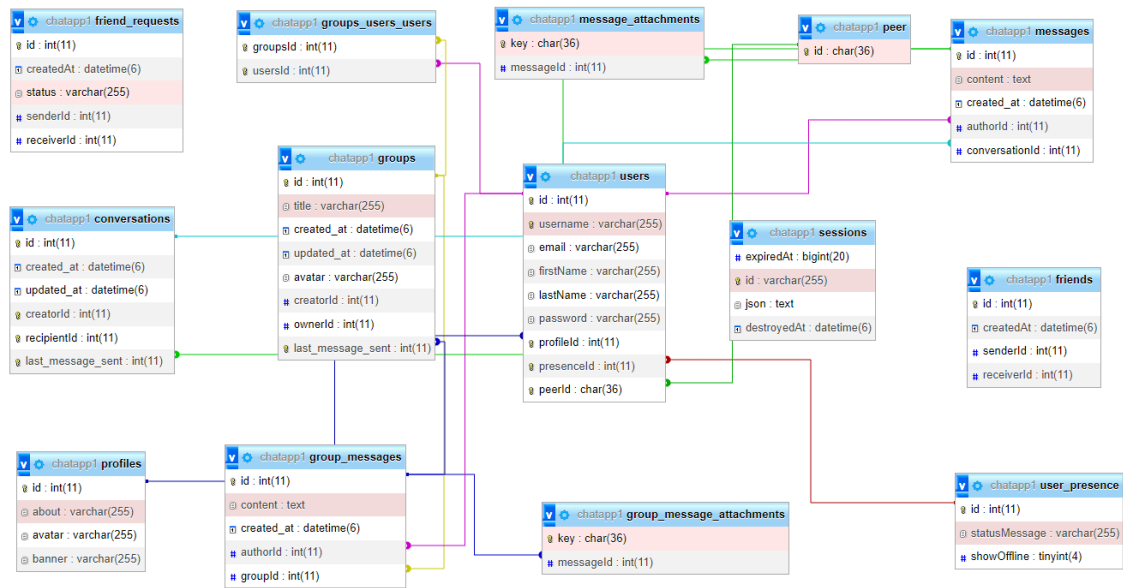
Bảng 3.32. Bảng user_presence

Thuộc tính	Kiểu dữ liệu	Nullable	Mô tả
ID	Int	No	ID
statusMessage	Varchar	Yes	Trạng thái tin nhắn
showOffline	Tinyint	No	Trạng thái hoạt động

3.7.14. Chi tiết bảng users

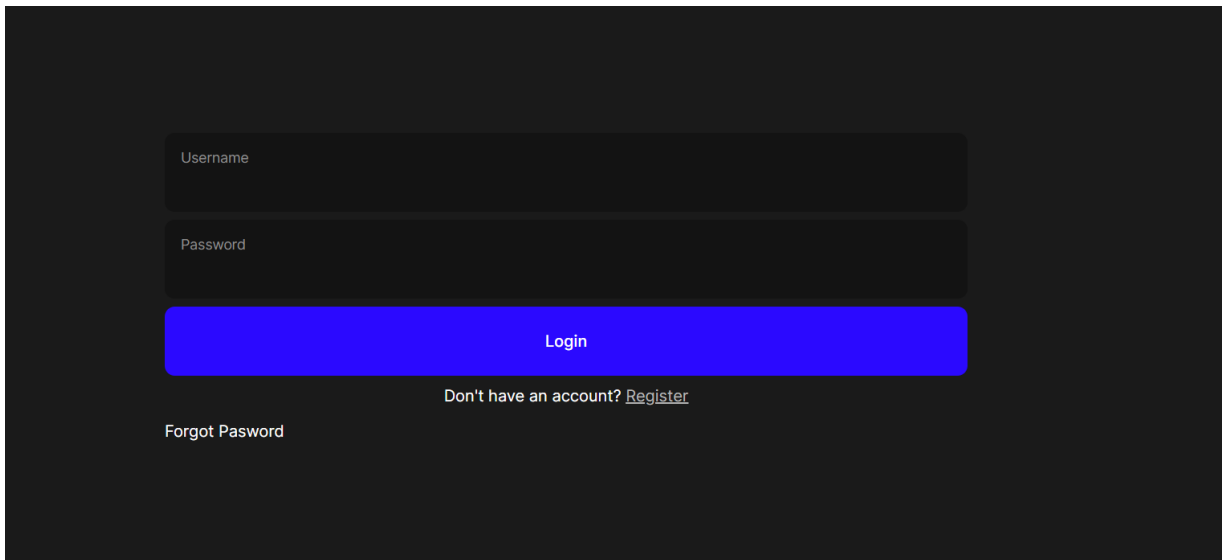
Bảng 3.33. Bảng users

Thuộc tính	Kiểu dữ liệu	Nullable	Mô tả
ID	int	No	ID
username	Varchar	No	Tên người dùng
email	Varchar	Yes	Email
firstName	Varchar	No	Họ
lastName	Varchar	No	Tên
password	Varchar	No	Mật khẩu
profileId	Int	Yes	Id hồ sơ
presenceId	Int	Yes	Id hiện diện
peerId	Char	Yes	Id peer



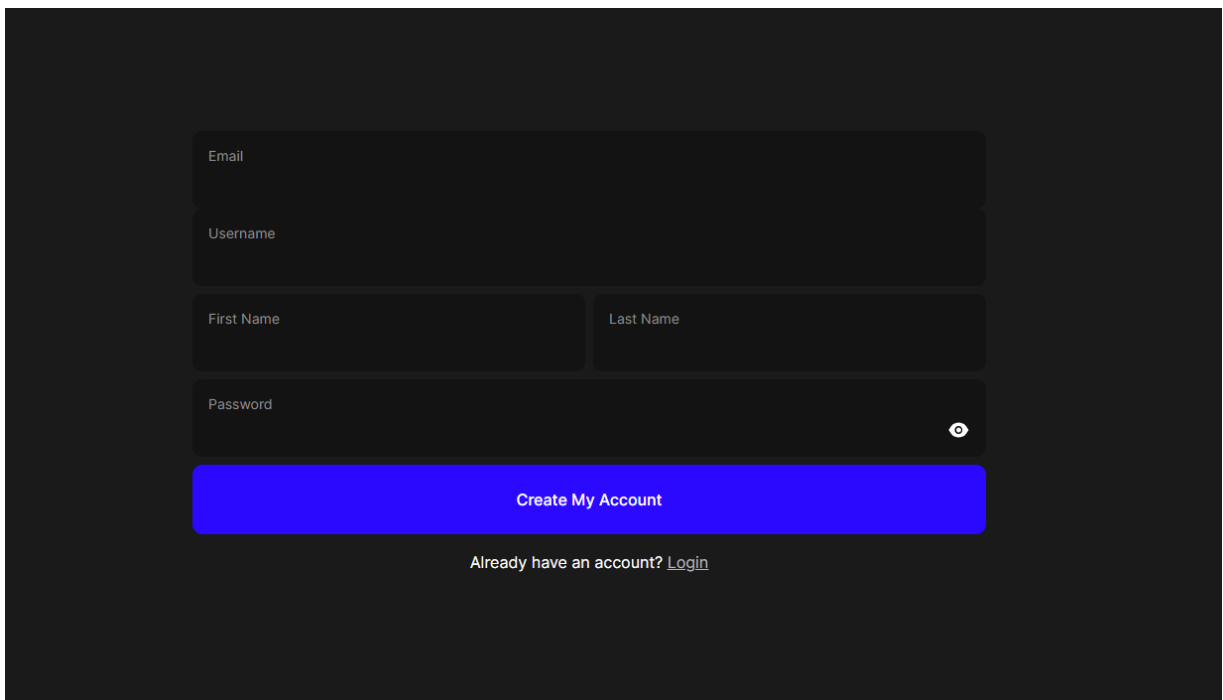
Hình 3.33. Cơ sở dữ liệu

3.8. THIẾT KẾ GIAO DIỆN



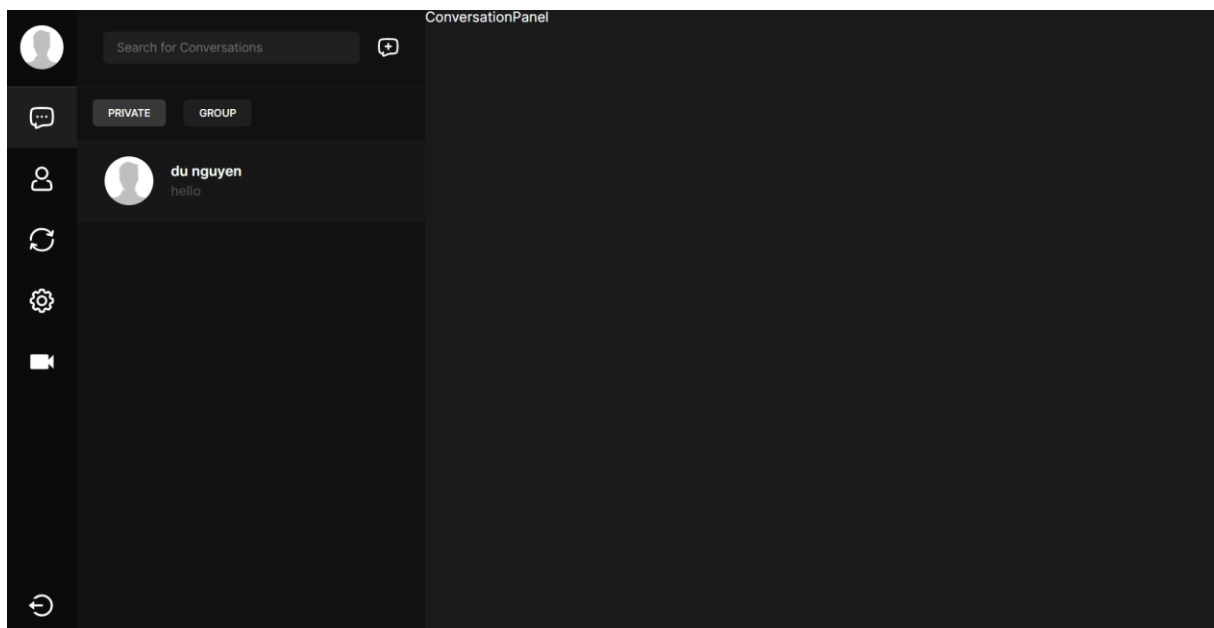
A login form with a dark background. It features two input fields: 'Username' and 'Password'. Below these is a large blue button labeled 'Login'. Under the button, there is a link 'Don't have an account? Register' and a link 'Forgot Password'.

Hình 3.34. Giao diện đăng nhập

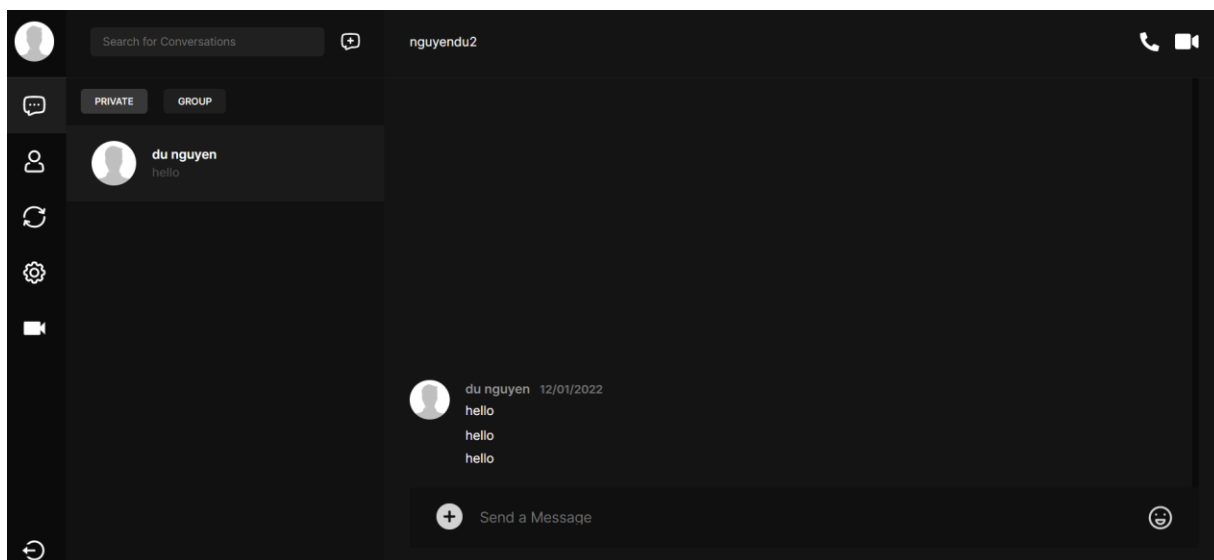


A registration form with a dark background. It features five input fields: 'Email', 'Username', 'First Name', 'Last Name', and 'Password'. The 'Password' field has a toggle icon. Below these is a large blue button labeled 'Create My Account'. Under the button, there is a link 'Already have an account? Login'.

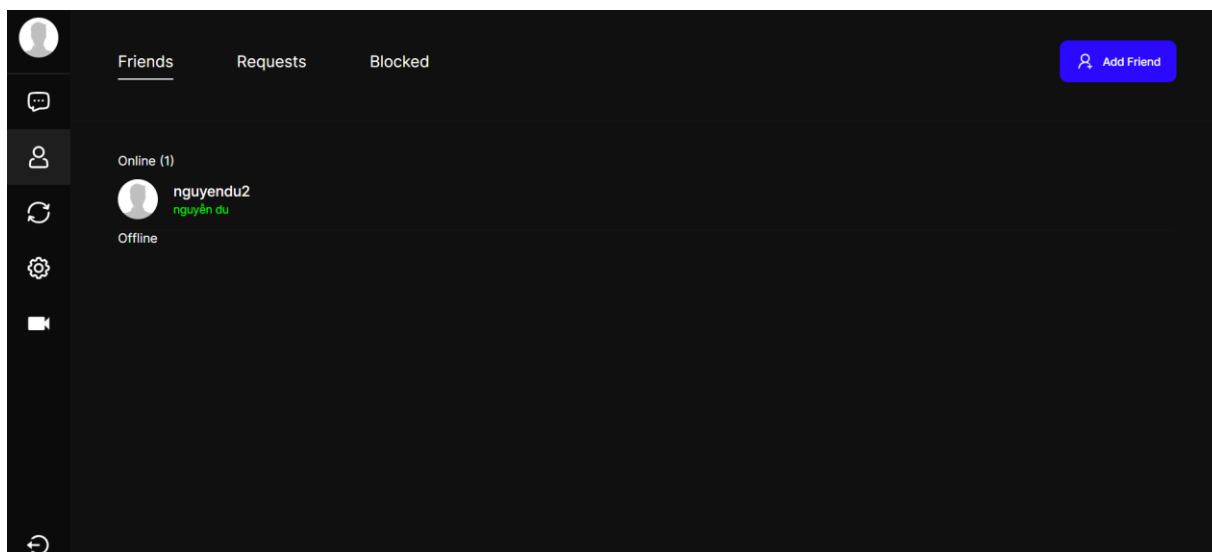
Hình 3.35. Giao diện đăng ký



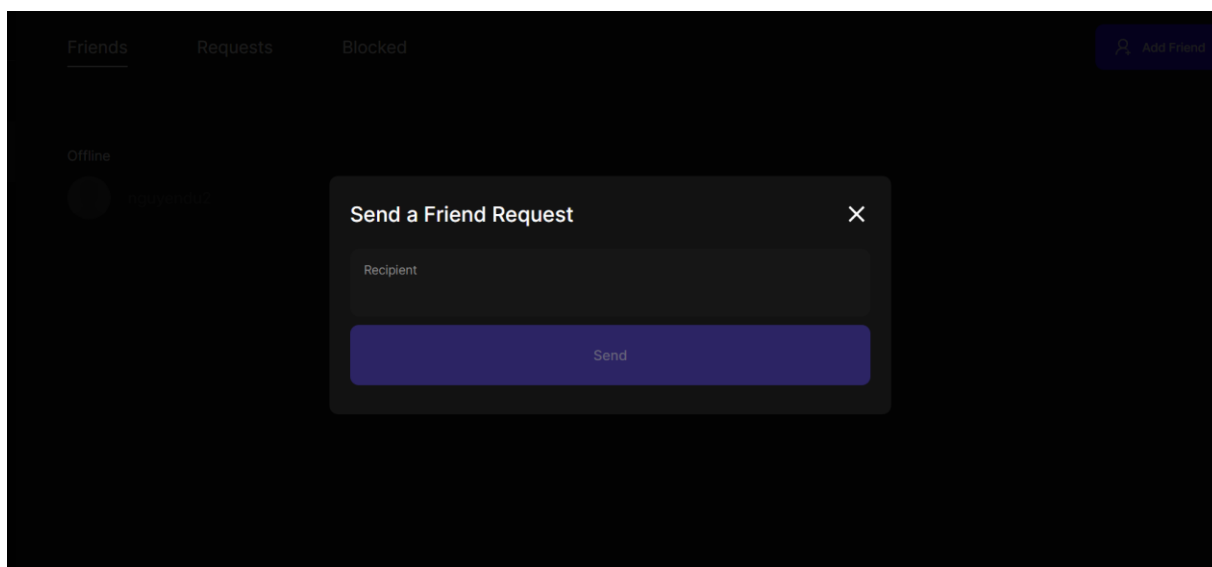
Hình 3.36. Giao diện Trang chủ



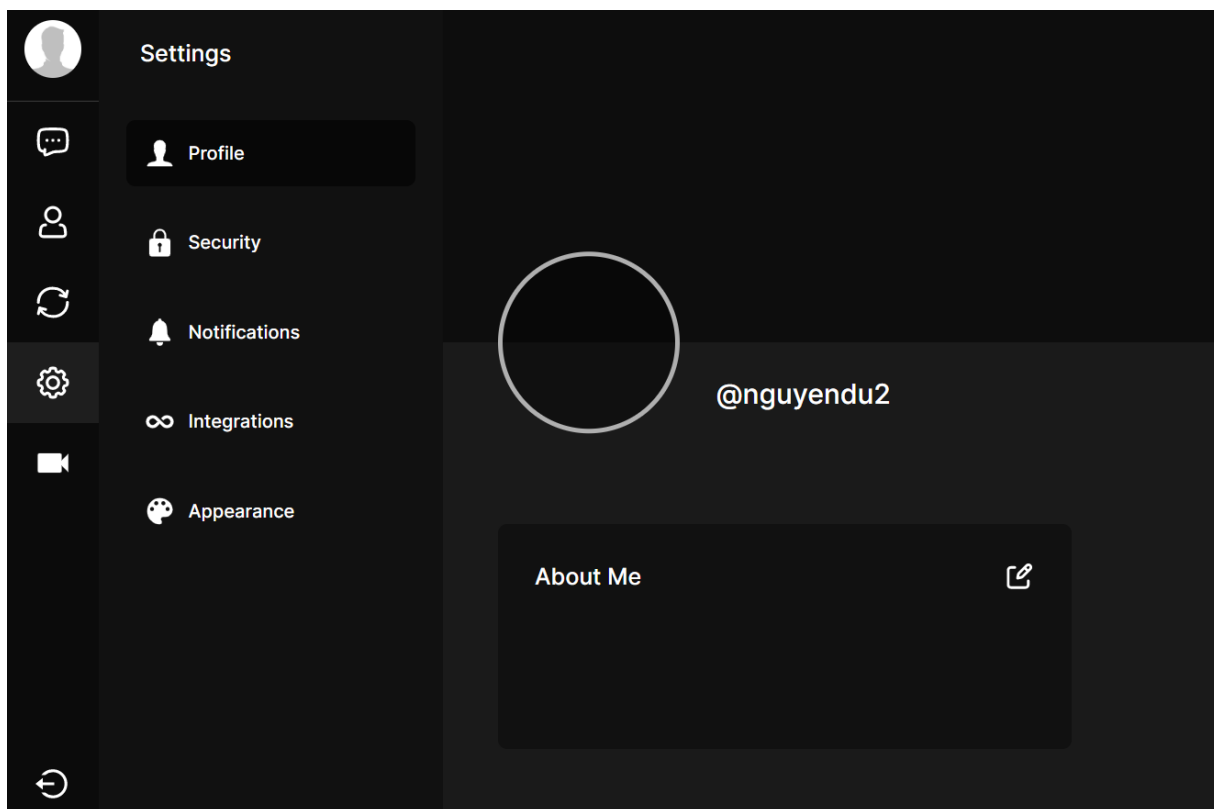
Hình 3.37. Giao diện nhắn tin



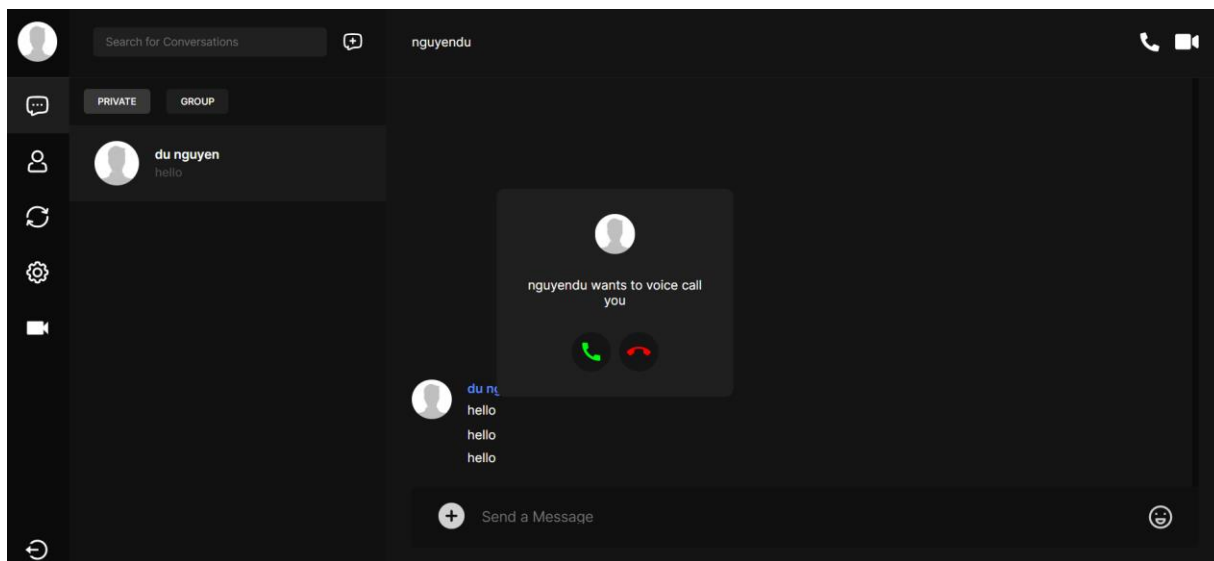
Hình 3.38. Giao diện danh sách bạn bè



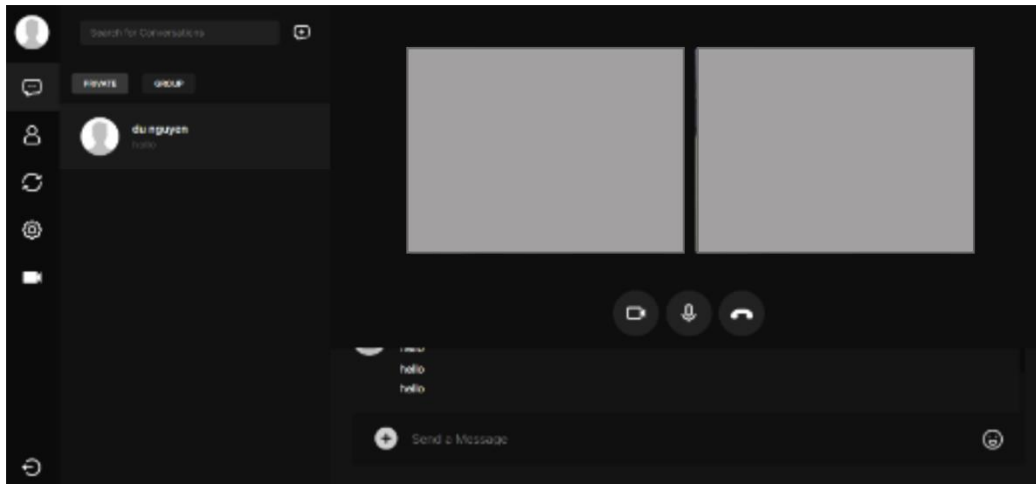
Hình 3.39. Giao diện thêm bạn bè



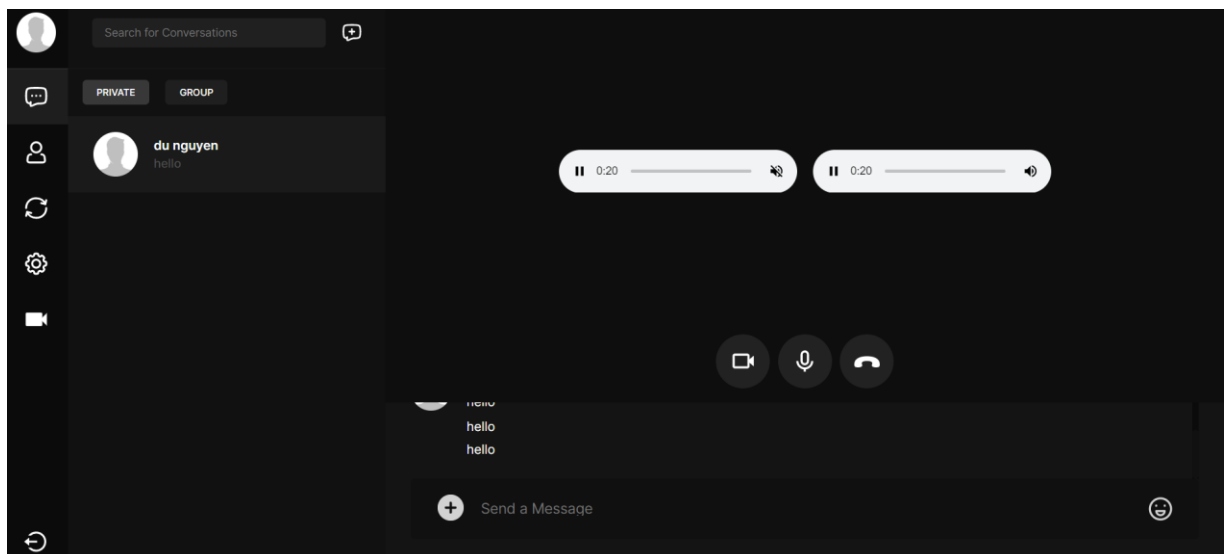
Hình 3.40. Giao diện Profile



Hình 3.41. Giao diện cuộc gọi đến



Hình 3.42. Giao diện Video Call



Hình 3.43. Giao diện cuộc gọi thoại

CHƯƠNG 4 XÂY DỰNG VÀ KIỂM THỬ

4.1. XÂY DỰNG

Link source code: <https://github.com/nguyendu334/chapApplication>

4.2. KIỂM THỬ

Bảng 4.1. Bảng kiểm thử

STT	Nội dung kiểm thử	Kết quả kì vọng	Kết quả thực tế	Kết luận
1	Đăng ký với Username trùng lặp	Hiển thị thông báo “USERNAME ALREADY EXISTS”	Hiển thị thông báo “USERNAME ALREADY EXISTS”	Pass
2	Đăng ký với Email trùng lặp	Hiển thị thông báo “EMAIL ALREADY EXISTS”	Hiển thị thông báo “EMAIL ALREADY EXISTS”	Pass
3	Đăng ký với Username chưa có	Hiển thị thông báo “Account created!”	Hiển thị thông báo “Account created!”	Pass
4	Đăng ký với Email chưa đúng định dạng	Hiển thị thông báo “Error creating user”	Hiển thị thông báo “Error creating user”	Pass

5	Đăng ký thiếu thông tin Email	Hiển thị thông báo “Email is Required”	Hiển thị thông báo “Email is Required”	Pass
6	Đăng ký thiếu thông tin Username	Hiển thị thông báo “Username is Required”	Hiển thị thông báo “Username is Required”	Pass
7	Đăng ký thiếu thông tin First name	Hiển thị thông báo “First name is Required”	Hiển thị thông báo “First name is Required”	Pass
8	Đăng ký thiếu thông tin Last name	Hiển thị thông báo “Last name is Required”	Hiển thị thông báo “Last name is Required”	Pass
9	Đăng ký thiếu thông tin Password	Hiển thị thông báo “Password is Required”	Hiển thị thông báo “Password is Required”	Pass
10	Đăng ký với mật khẩu quá ngắn	Hiển thị thông báo “MUST BE AT LEAST 8 CHARACTERS”	Hiển thị thông báo “MUST BE AT LEAST 8 CHARACTERS”	Pass
11	Đăng nhập với tên tài khoản và mật khẩu chính xác	Hiển thị thông báo “Success”	Hiển thị thông báo “Success”	Pass

12	Đăng nhập với tên tài khoản hoặc mật khẩu không chính xác	Hiển thị thông báo lỗi “Login failed”	Hiển thị thông báo lỗi “Login failed”	Pass
13	Người dùng gửi tin nhắn	Hiển thị màn hình tin nhắn	Hiển thị màn hình tin nhắn	Pass
14	Người dùng gọi video call	Hiển thị màn hình gọi video call và thông báo tới người nhận	Hiển thị màn hình gọi video call và thông báo tới người nhận	Pass
15	Quên mật khẩu với email chưa được đăng ký sử dụng	Hiển thị thông báo email chưa tồn tại	Không hiển thị thông báo	Fail
16	Quên mật khẩu với email đã được đăng ký sử dụng	Hiển thị thông báo “Vui lòng kiểm tra email để lấy mã otp đổi mật khẩu” và chuyển đến màn hình đổi mật khẩu	Không hiển thị	Fail
17	Đổi mật khẩu	Hiển thị thông báo “Đổi mật khẩu thành công”	Không hiển thị	Fail

18	Người dùng tạo group trò chuyện	Hiển thị Group và trò chuyện với mọi người	Hiển thị Group và trò chuyện với mọi người	Pass
19	Thêm thành viên vào group trò chuyện	Thêm thành viên thành công	Thêm thành viên thành công	Pass
20	Người dùng thay đổi ảnh đại diện Group	Thay đổi ảnh đại diện thành công	Thay đổi ảnh đại diện không thành công	Fail
21	Người dùng thay đổi avartar	Cập nhật Avartar thành công	Cập nhật Avartar không thành công	Fail
22	Người dùng cập nhật trạng thái	Cập nhật trạng thái thành công	Cập nhật trạng thái thành công	Pass
23	Người dùng thay đổi giao diện hệ thống	Thay đổi giao diện thành công	Thay đổi giao diện thành công	Pass
24	Thêm bạn bè đã có trong danh sách	Hiển thị thông báo "Friend Already Exists"	Hiển thị thông báo "Friend Already Exists"	Pass
25	Thêm bạn bè chưa có tài khoản	Hiển thị thông báo "Friend Not Found"	Hiển thị thông báo "Friend Not Found"	Pass

CHƯƠNG 5 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. KẾT LUẬN

Trong quá trình hoàn thiện đồ án tốt nghiệp, em đã vận dụng các kiến thức đã học, kết hợp tra cứu các nguồn tài liệu tham khảo để hoàn thành đồ án tốt nghiệp nhưng do kiến thức còn hạn chế cũng như kinh nghiệm chưa đủ nên website chỉ hoàn thiện ở mức độ nào đó cũng như các kiến thức thu hoạch được như:

- Hiểu được quy trình thực hiện xây dựng 1 website.
- Hoàn thành đặc tả yêu cầu, thiết kế giao diện website, thiết kế CSDL...

5.2. HƯỚNG PHÁT TRIỂN

Hiện tại website chỉ được dùng gửi tin nhắn và gọi điện, gọi video call cơ bản. Hướng phát triển của website là sẽ tiến đến một website chuyên nghiệp có thể đáp ứng được nhu cầu của người dùng và sẽ phát triển thêm nhiều chức năng. Từ đó công việc thương mại của website sẽ thuận lợi hơn thu hút nhiều cá nhân và doanh nghiệp sử dụng.

Mặc dù đã cố gắng hoàn thiện nhưng vẫn còn nhiều thiếu sót, mong các thầy và các cô sẽ đưa thêm ý kiến bổ sung để bổ sung cũng như khắc phục các vấn đề cho việc xây dựng hoàn chỉnh hơn.

TÀI LIỆU THAM KHẢO

- [1] <https://gabrieltanner.org/blog/webrtc-video-broadcas>.
- [2] <https://docs.nestjs.com/>.
- [3] <https://viblo.asia/p/nestjs-framework-thanh-thanh-cho-nodejs-RQqKLL7OK7z>.
- [4] <https://reactjs.org/docs/getting-started.html>.
- [5] <https://webRTC.org/getting-started/overview>.
- [6] <https://tmarketing.vn/thiet-ke-database-mysql-co-so-du-lieu/>.
- [7] <https://viblo.asia/p/huong-dan-xay-dung-ung-dung-chat-real-time-voi-node-js-WAyK8peoKxX>.