## TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT – HÀN

**KHOA KHOA HỌC MÁY TÍNH**



# ĐỒ ÁN MÔN HỌC LẬP TRÌNH MẠNG

**ĐỀ TÀI:**

**XÂY DỰNG ỨNG DỤNG HỌP TRỰC TUYẾN SỬ DỤNG GIAO THỨC TCP VÀ**

**CÔNG NGHỆ WEBRTC, KẾT HỢP VỚI MÔ HÌNH CLIENT-SERVER.**

|  |  |
| --- | --- |
| Sinh viên thực hiện: | **LÊ ĐỨC ANH VIỆT** Mã: **22IT338 NGUYỄN ĐÌNH QUỐC KHÁNH** Mã: **22IT14141** |
| Lớp: | **22JIT** |
| Giảng viên hướng dẫn: | **ThS. NGUYỄN THANH CẨM** |

## TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT – HÀN

**KHOA KHOA HỌC MÁY TÍNH**

**ĐỒ ÁN MÔN HỌC**

### SỬ DỤNG CÔNG NGHỆ WEBRTC VỚI GIAO THỨC HTTPS, TCP, UDP VÀ MÔ HÌNH CLIENT/SERVER ĐỂ XÂY DỰNG CHƯƠNG TRÌNH MẠNG XÃ HỘI TRỰC TUYẾN

**LỜI CẢM ƠN**

Để hoàn thành báo cáo đồ án này trước tiên chúng em xin gửi đến ThS. Nguyễn Thanh Cẩm - người trực tiếp hướng dẫn đồ án môn học lời cảm ơn sâu sắc nhất.

Cảm ơn thầy đã quan tâm, hướng dẫn và định hướng cho chúng em về hướng đi và phát triển của đề tài, cách triển khai báo cáo, cũng như cung cấp kiến thức về lý thuyết để thực hiện sản phẩm.

Vì khả năng sinh viên còn hạn chế, trong quá trình học tập, hoàn thiện báo cáo bài tập này em không tránh khỏi những sai sót, kính mong nhận được những ý kiến đóng góp từ các thầy(cô).

Cuối cùng chúng em kính chúc thầy(cô) dồi dào sức khỏe và đạt được nhiều thành công tốt đẹp trong công việc.

***Em xin chân thành cảm ơn!***

**LỜI GIỚI THIỆU**

Ngày nay, với sự phát triển mạnh mẽ của công nghệ thông tin và Internet, nhu cầu kết nối và trao đổi thông tin trong thời gian thực đã trở thành một phần không thể thiếu trong cuộc sống và công việc. Các ứng dụng meeting trực tuyến không chỉ giúp người dùng duy trì liên lạc mà còn tạo ra môi trường tương tác chân thực, nơi mọi người có thể thảo luận, học hỏi và chia sẻ ý tưởng. Vì vậy, việc xây dựng một ứng dụng meeting online với các tính năng hiện đại, thân thiện là một nhu cầu cấp thiết trong bối cảnh hiện nay.

Đối tượng hướng tới của ứng dụng là tất cả người dùng có nhu cầu kết nối, giao tiếp trong các buổi họp trực tuyến, đặc biệt là các doanh nghiệp, tổ chức giáo dục và các nhóm cộng đồng. Ứng dụng sẽ cung cấp một nền tảng linh hoạt, dễ sử dụng giúp người dùng dễ dàng tạo phòng họp, mời người tham gia, chia sẻ tài liệu và tổ chức cuộc họp một cách thuận tiện và hiệu quả.

Về phạm vi, ứng dụng bao gồm các tính năng cơ bản của một nền tảng meeting trực tuyến như tạo và quản lý phòng họp, hệ thống nhắn tin, thông báo, chia sẻ màn hình và ghi lại cuộc họp. Bên cạnh đó, ứng dụng sẽ tích hợp các tính năng nâng cao như gọi video chất lượng cao, chia sẻ tài liệu trực tuyến và sử dụng công nghệ WebRTC để tối ưu hóa trải nghiệm người dùng. Phiên bản đầu tiên của ứng dụng sẽ tập trung triển khai trên nền tảng web để mô phỏng hiệu quả môi trường làm việc trực tuyến.

Phương pháp tiếp cận để phát triển ứng dụng là kết hợp nghiên cứu lý thuyết về công nghệ mạng với các phương pháp lập trình hiện đại. Các công cụ và nền tảng như ExpressJs, WebRTC, HTML, CSS, JavaScript và công nghệ Socket sẽ được sử dụng nhằm đảm bảo hiệu năng và khả năng mở rộng trong tương lai.

Dự kiến, kết quả của đề tài là xây dựng một ứng dụng meeting online hoàn chỉnh, đáp ứng các nhu cầu cơ bản và nâng cao của người dùng. Ứng dụng sẽ cung cấp giao diện trực quan, dễ sử dụng, cùng với các tính năng hỗ trợ giao tiếp và kết nối hiệu quả, tạo ra một môi trường họp trực tuyến an toàn và tiện lợi cho người dùng.

**NHẬN XÉT CỦA GIẢNG VIÊN**

..................................................................................................................................

............................................................................................................................. ..............

...........................................................................................................................................

...........................................................................................................................................

............................................................................................................................. ..............

...........................................................................................................................................

............................................................................................................................. ..............

...........................................................................................................................................

............................................................................................................................. ..............

...........................................................................................................................................

............................................................................................................................. ..............

...........................................................................................................................................

............................................................................................................................. ..............

...........................................................................................................................................

............................................................................................................................. ..............

............................................................................................................................. ...

Đà Nẵng, tháng 10 năm 2023

### Giảng viên

**MỤC LỤC**

[CHƯƠNG 1: SƠ LƯỢC VỀ LẬP TRÌNH MẠNG 1](#_bookmark0)

* 1. [Khái niệm về lập trình mạng 1](#_bookmark1)
  2. [Các kiến trúc mạng 1](#_bookmark2)
     1. [Các mô hình mạng 1](#_bookmark3)
     2. [Các mô hình ứng dụng mạng 4](#_bookmark6)
  3. [Các giao thức mạng 5](#_bookmark7)
     1. [Các giao thức tầng ứng dụng (Tầng 7) 5](#_bookmark8)
     2. [Các giao thức tầng trình bày (Tầng 6) 6](#_bookmark9)
     3. [Các giao thức tầng phiên (Tầng 5) 6](#_bookmark10)
     4. [Các giao thức tầng giao vận (Tầng 4) 6](#_bookmark11)
     5. [Các giao thức tầng mạng (Tầng 3) 7](#_bookmark12)
     6. [Các giao thức tầng liên kết dữ liệu (Tầng 2) 8](#_bookmark13)
     7. [Các giao thức tầng vật lí (Tầng 1) 8](#_bookmark14)
  4. [Sơ lược về một số công nghệ 8](#_bookmark15)
     1. [WebRTC 8](#_bookmark16)
     2. [ElectronJs 10](#_bookmark18)
     3. [ReactJs 11](#_bookmark20)
     4. [SocketIO 12](#_bookmark22)
     5. [Redis 13](#_bookmark24)

[Chương 2 PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG 13](#_TOC_250000)

* 1. [Giới thiệu sơ lược về bài toán 13](#_bookmark25)
  2. Sơ đồ Usecase 15

Chương 3 Phân tích và thiết kế hệ thống 16

* 1. [Biểu đồ use case 16](#_bookmark26)
  2. [Đặc tả usecase 16](#_bookmark28)
     1. [Đăng kí 16](#_bookmark29)
     2. Đăng nhập 17
     3. [Thêm bài đăng 18](#_bookmark32)
     4. Thêm bình luận 18
     5. [Gửi yêu cầu tham gia nhóm 19](#_bookmark33)
     6. [Xóa thành viên 20](#_bookmark34)
     7. [Nhắn tin 20](#_bookmark35)
     8. [Gửi lời mời kết bạn 21](#_bookmark36)
     9. [Xử lí yêu cầu tham gia nhóm 22](#_bookmark37)
     10. [Xử lí yêu cầu kết bạn 23](#_bookmark38)
  3. Biểu đồ lớp 25
  4. [Biểu đồ tuần tự 26](#_bookmark39)
     1. [Đăng kí 26](#_bookmark40)
     2. [Đăng nhập 27](#_bookmark41)
     3. Tạo bài đăng 28
     4. [Thêm bình luận 29](#_bookmark42)
     5. [Gửi yêu cầu tham gia nhóm 29](#_bookmark43)
     6. [Xử lí yêu cầu tham gia nhóm 30](#_bookmark44)
     7. [Xóa thành viên 31](#_bookmark45)
     8. Gửi yêu cầu kết bạn 32
     9. [Xử lí yêu cầu kết bạn 33](#_bookmark46)
     10. [Nhắn tin 34](#_bookmark47)
  5. [Biểu đồ hoạt động 35](#_bookmark48)
     1. [Đăng kí 35](#_bookmark49)
     2. [Đăng nhập 36](#_bookmark50)
     3. [Tạo bài đăng 37](#_bookmark51)
     4. [Thêm bình luận 38](#_bookmark52)
     5. [Gửi yêu cầu tham gia nhóm 39](#_bookmark53)
     6. [Xử lí yêu cầu tham gia nhóm 40](#_bookmark54)
     7. [Xóa thành viên 41](#_bookmark55)
     8. Nhắn tin 42
     9. [Gửi yêu cầu kết bạn 43](#_bookmark56)
     10. [Xử lí yêu cầu kết bạn 43](#_bookmark57)

[Chương 4 CHƯƠNG TRÌNH DEMO 45](#_bookmark58)

* 1. [Giao diện trang Đăng nhập & Đăng kí 45](#_bookmark59)
  2. [Giao diện màn hình chính 45](#_bookmark60)
  3. [Giao diện cửa sổ bình luận 46](#_bookmark61)
  4. [Giao diện cửa sổ tạo bài viết 47](#_bookmark62)
  5. Giao diện tìm kiếm 48
  6. Giao diện nhắn tin 49
  7. Giao diện gọi video 50
  8. Giao diện trang khám phá nhóm 52
  9. Giao diện trang chi tiết nhóm 52
  10. Giao diện trang Livestream 52
  11. [Kết quả đạt được 53](#_bookmark63)
  12. [Những kĩ năng học được 53](#_bookmark64)
  13. [Hướng phát triển 53](#_bookmark65)

**DANH MỤC HÌNH**

|  |  |  |
| --- | --- | --- |
| [**Hình**](#_bookmark4) | [**1.2.1. Các mô hình mạng**](#_bookmark4) | [3](#_bookmark4) |
| [**Hình**](#_bookmark5) | [**1.2.2. Các mô hình mạng TCP/IP**](#_bookmark5) | [4](#_bookmark5) |
| [**Hình**](#_bookmark17) | [**1.4.1. Công nghệ WebRTC**](#_bookmark17) | [10](#_bookmark17) |
| [**Hình**](#_bookmark19) | [**1.4.2. Công nghệ Electron**](#_bookmark19) | [11](#_bookmark19) |
| [**Hình**](#_bookmark21) | [**1.4.3. Framework NESTJS**](#_bookmark21) | [12](#_bookmark21) |
| [**Hình**](#_bookmark23) | [**1.4.4. Thư viện Socket.io**](#_bookmark23) | [13](#_bookmark23) |

**DANH MỤC BẢNG**

### [Bảng 1.2.2. Biểu đồ use case 15](#_bookmark27)

### [Bảng 2.3.1. Đăng kí 16](#_bookmark30)

### [Bảng 2.3.2. Đăng nhập 18](#_bookmark31)

# CHƯƠNG 1: SƠ LƯỢC VỀ LẬP TRÌNH MẠNG

## Khái niệm về lập trình mạng

Lập trình mạng có thể được hiểu đơn giản là xây dựng các ứng dụng, chương trình hoạt động thông qua mạng máy tính. Là hành động sử dụng mã máy tính để viết các chương trình hoặc quy trình có thể giao tiếp với các chương trình hoặc quy trình khác trên mạng.

## Các kiến trúc mạng

Các hệ thống mạng ngày nay được thiết kế dựa trên những mô hình kiến trúc cụ thể, nhằm đảm bảo sự hiệu quả, an toàn và tính ổn định khi truyền tải thông tin. Các mô hình mạng như OSI (7 tầng), TCP/IP (4 tầng) và các mô hình ứng dụng như client- server, peer-to-peer đóng vai trò quan trọng trong việc định hình cách thức hoạt động của các mạng và ứng dụng trên toàn thế giới. Hiểu rõ về chúng sẽ giúp việc thiết kế và triển khai hệ thống mạng trở nên hiệu quả và có hệ thống hơn.

### Các mô hình mạng

Một trong những mô hình mạng cơ bản và lâu đời nhất là mô hình OSI (Open Systems Interconnection). Đây là một mô hình lý thuyết được phát triển nhằm chuẩn hóa cách các thiết bị kết nối và giao tiếp với nhau trên mạng. Mô hình này chia quá trình truyền tải dữ liệu thành bảy tầng, mỗi tầng đảm nhận một chức năng cụ thể, từ việc truyền dữ liệu vật lý qua dây cáp đến việc quản lý các ứng dụng cuối. Mô hình OSI giúp phân tách các vấn đề phức tạp thành từng phần nhỏ dễ quản lý, giúp các nhà phát triển có thể xác định và giải quyết sự cố nhanh chóng, tập trung vào từng tầng riêng biệt.

Mô hình OSI gồm: 4 tầng cao (Ứng dụng – Trình bày – Phiên – Vận chuyển) và 3 tầng thấp (Vật lí – Kết nối dữ liệu – Mạng). Trong khi các tầng cao tập trung vào những vấn đề liên quan đến ứng dụng, là nơi mà người dùng tương tác với phần mềm thì đối với các tầng thấp, tập trung vào việc xử lý các vấn đề liên quan đến truyền dẫn dữ liệu.

Về cụ thể các tầng theo thứ tự từ cao xuống thấp:

Tầng 7 - Tầng ứng dụng (Application) đặt ở vị trí cao nhất, thiết lập giao diện người dùng với mô hình OSI. Tầng này cung cấp một giao diện và các dịch vụ cho phép người dùng tương tác và sử dụng ứng dụng mạng một cách hiệu quả.

Tầng tiếp theo, Tầng 6 - Tầng Trình bày (Presentation), là nơi xử lý các vấn đề liên quan đến cú pháp và ngữ nghĩa của dữ liệu được truyền. Tầng Trình bày lấy dữ liệu từ Tầng Ứng dụng và chuẩn bị cho việc truyền qua Tầng Phiên. Nhiệm vụ của tầng này là biến dữ liệu thành dạng hiển thị đúng cho các ứng dụng sử dụng. Tầng này thực hiện nhiệm vụ dịch, mã hóa và nén dữ liệu.

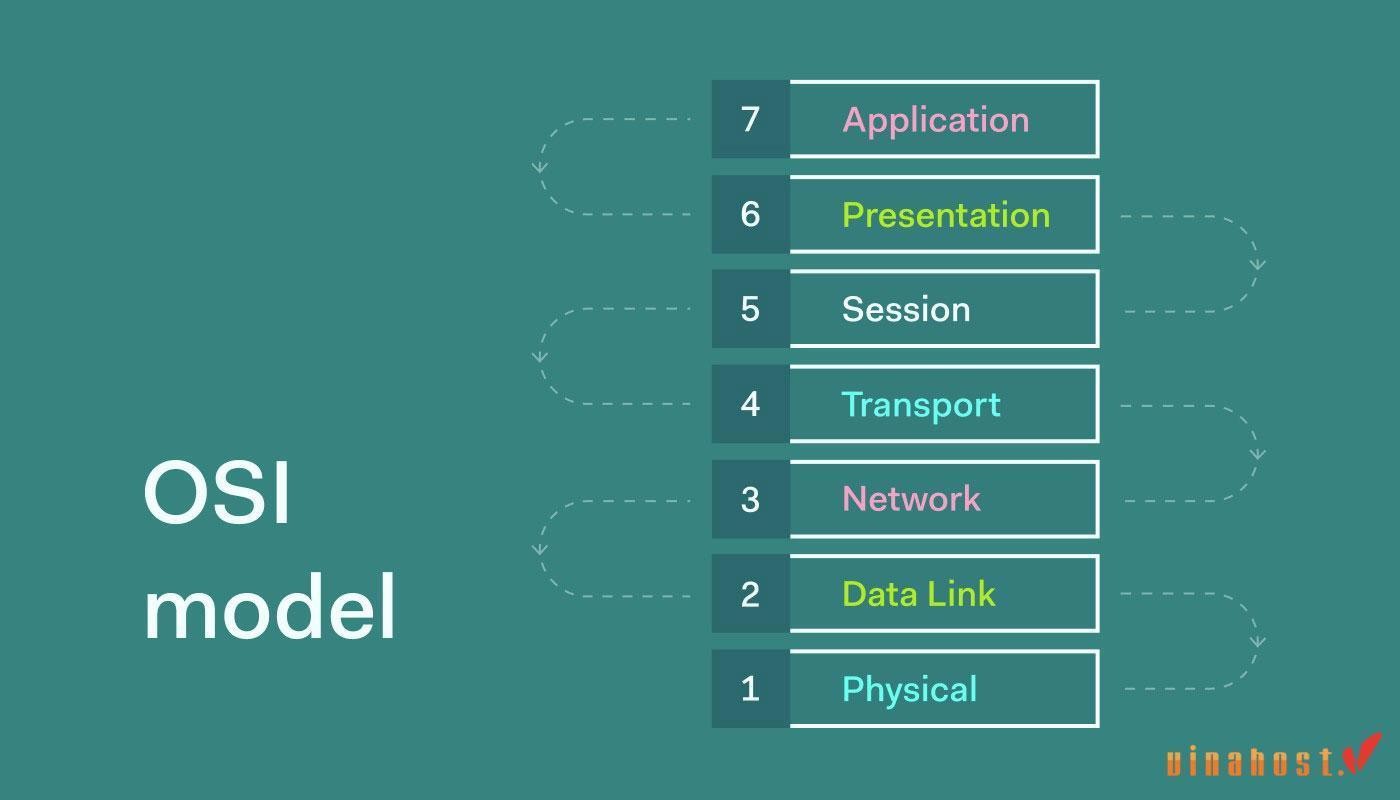
Tầng 5 - Tầng phiên (Session) đảm nhận vai trò điều khiển việc mở và đóng kết nối giữa hai thiết bị. Thời gian từ khi kết nối mở đến khi đóng được gọi là một phiên. Tầng này đảm bảo rằng phiên được mở đủ lâu để hoàn thành việc truyền dữ liệu và sau đó nhanh chóng đóng lại để tránh lãng phí tài nguyên.

Tầng 4 - Tầng vận chuyển (Transport), tầng thứ 4 trong mô hình OSI, chịu trách nhiệm đảm bảo dữ liệu truyền đi là đáng tin cậy. Nhiệm vụ của Tầng 4 là quản lý giao tiếp cuối cùng giữa hai thiết bị, bao gồm việc thu thập dữ liệu từ lớp phiên và phân chia nó thành các phân đoạn (segments) trước khi gửi xuống Tầng 3, đánh số gói tin để đảm bảo thông tin truyền đi theo thứ tự mong muốn. Tại thiết bị đích, Tầng vận chuyển phụ trách việc tái hợp các phân đoạn thành dữ liệu mà lớp phiên có thể sử dụng.

Tầng 3 - Tầng Mạng (Network) là tầng thứ ba trong mô hình OSI, có nhiệm vụ quản lý định tuyến và chuyển tiếp dữ liệu giữa các mạng khác nhau trong hệ thống mạng. Tầng mạng phân chia các phân đoạn từ tầng truyền tải thành các đơn vị nhỏ hơn, được gọi là gói (packets), trên thiết bị của người gửi và tái hợp các gói này trên thiết bị nhận. Ngoài ra, tầng mạng cũng xác định con đường tối ưu để dữ liệu đến đích của nó; hay còn được gọi là định tuyến. Trong trường hợp hai thiết bị giao tiếp trên cùng một mạng, tầng mạng trở nên không cần thiết.

Tầng 2 - Tầng Liên kết dữ liệu (Data link), được đặt ở vị trí thứ hai trong mô hình OSI, có trách nhiệm xác định địa chỉ vật lý (Mac Address) và đảm bảo việc truyền dữ liệu an toàn qua các đường truyền vật lý.

Cuối cùng là tầng 1 - tầng Vật lý (Physical) đặt ở tầng thấp nhất trong mô hình OSI, có trách nhiệm đảm bảo việc truyền dữ liệu qua môi trường vật lý.



**Hình 1.2.1. Các mô hình mạng**

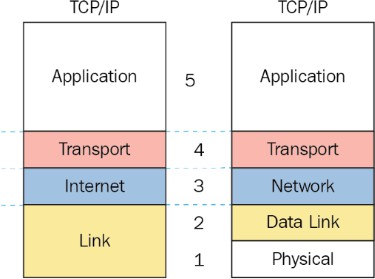
Còn đối với mô hình TCP/IP được cấu thành từ 4 tầng: Tầng ứng dụng (là sự kết hợp giữa tầng Ứng dụng – Trình bày – Phiên của mô hình OSI), tầng giao vận, tầng mạng, tầng vật lí (là sự kết hợp giữa tầng Vật lí – Kết nối dữ liệu). Mô hình TCP/IP được xem là sự thay thế chính thức cho mô hình OSI trong việc mô tả cách mạng hoạt động. Mặc dù mô hình OSI vẫn được sử dụng rộng rãi cho mục đích giáo dục, nhưng trong thực tế, mô hình TCP/IP với tính thực tiễn cao đã trở nên phổ biến và được xem là chính xác, hiệu quả hơn khi ánh xạ đến cấu trúc và giao thức của Internet. Sau đây là chi tiết các tầng của mô hình TCP/IP:

Tầng 4 – Tầng ứng dụng (Application): Đây là lớp giao tiếp trên cùng của mô hình. Đúng với tên gọi, tầng Ứng dụng đảm nhận vai trò giao tiếp dữ liệu giữa 2 máy khác nhau thông qua các dịch vụ mạng khác nhau (duyệt web, chat, gửi email, một sốgiao thức trao đổi dữ liệu: SMTP, SSH, FTP, ...). Dữ liệu khi đến đây sẽ được định dạng theo kiểu Byte nối Byte, cùng với đó là các thông tin định tuyến giúp xác định đường đi đúng của một gói tin.

Tầng 3 - Tầng Giao vận (Transport): Chức năng chính của tầng 3 là xử lý vấn đềgiao tiếp giữa các máy chủ trong cùng một mạng hoặc khác mạng được kết nối vớinhau thông qua bộ định tuyến. Tại đây dữ liệu sẽ được phân đoạn, mỗi đoạn sẽ khôngbằng nhau nhưng kích thước phải nhỏ hơn 64KB. Cấu trúc đầy đủ của một Segment lúc này là Header chứa thông tin điều khiển và sau đó là dữ liệu.

Tầng 2 - Tầng mạng (Internet): Gần giống như tầng mạng của mô hình OSI. Tạiđây, nó cũng được định nghĩa là một giao thức chịu trách nhiệm truyền tải dữ liệu mộtcách logic trong mạng. Các phân đoạn dữ liệu sẽ được đóng gói (Packets) với kíchthước mỗi gói phù hợp với mạng chuyển mạch mà nó dùng để truyền dữ liệu. Lúc này, các gói tin được chèn thêm phần Header chứa thông tin của tầng mạng và tiếp tục được chuyển đến tầng tiếp theo. Các giao thức chính trong tầng là IP, ICMP và ARP.

Tầng 1 - Tầng Vật lý (Physical): Là sự kết hợp giữa tầng Vật lý và tầng liên kếtdữ liệu của mô hình OSI. Chịu trách nhiệm truyền dữ liệu giữa hai thiết bị trong cùngmột mạng. Tại đây, các gói dữ liệu được đóng vào khung (gọi là Frame) và được định tuyến đi đến đích đã được chỉ định ban đầu.



**Hình 1.2.2. Các mô hình mạng TCP/IP**

### Các mô hình ứng dụng mạng

Bên cạnh các mô hình mạng cơ bản, cách thức các ứng dụng triển khai giao tiếp trên mạng cũng đóng vai trò quan trọng không kém. Mô hình client-server là một trong những kiến trúc phổ biến nhất, nơi một máy chủ cung cấp dịch vụ cho nhiều máy khách kết nối đến. Các dịch vụ như web, email, và cơ sở dữ liệu thường được thiết kế theo cách này, vì nó cho phép quản lý tập trung và đảm bảo rằng các yêu cầu được xử lý một cách chính xác, nhanh chóng. Máy khách gửi yêu cầu đến máy chủ, và máy chủ sẽ xử lý và trả về kết quả, tạo ra một quy trình đơn giản nhưng hiệu quả.

Ngược lại với mô hình client-server, mô hình peer-to-peer (P2P) không cần một máy chủ trung gian. Các thiết bị trong mạng P2P có quyền ngang hàng và có thể trao đổi dữ liệu trực tiếp với nhau. Điều này làm tăng tính phân tán và độc lập của hệ thống, giúp nó dễ dàng mở rộng mà không cần phụ thuộc vào một điểm trung tâm. Mô hình P2P được ứng dụng rộng rãi trong các hệ thống chia sẻ tệp tin, các dịch vụ truyền dữ liệu trực tiếp và các mạng blockchain. Tuy nhiên, sự phân tán này cũng đồng nghĩa với việc quản lý và bảo mật trở nên phức tạp hơn so với mô hình client-server.

## Các giao thức mạng

Trong kiến trúc mạng, các giao thức mạng đóng vai trò quan trọng trong việc

đảm bảo các thiết bị có thể giao tiếp và truyền tải dữ liệu một cách hiệu quả, an toàn. Mỗi tầng trong mô hình OSI đều có các giao thức riêng, đảm nhận các nhiệm vụ khác nhau từ việc truyền bit dữ liệu cho đến việc quản lý các phiên kết nối. Dưới đây là một số các giao thức tiêu biểu qua 7 tầng của mô hình OSI.

### Các giao thức tầng ứng dụng (Tầng 7)

Tầng Ứng dụng chứa đựng các giao thức và dịch vụ đa dạng như gửi tệp, email, truy cập trang web, phát dữ liệu đa phương tiện, và nhiều chức năng ứng dụng khác. Các giao thức và dịch vụ nổi bật trong tầng này bao gồm:

### HTTP (Hypertext Transfer Protocol)

HTTP là giao thức tầng ứng dụng, chủ yếu dùng để truyền tải dữ liệu trên web. Cấu trúc của một yêu cầu HTTP bao gồm ba phần chính: dòng yêu cầu (request line), phần tiêu đề (header), và phần nội dung (body). Khi một trình duyệt yêu cầu một tài nguyên từ máy chủ web, trình duyệt gửi một yêu cầu HTTP chứa các thông tin về

phương thức (GET, POST), đường dẫn tài nguyên và phiên bản giao thức.

HTTP hoạt động theo mô hình client-server, trong đó máy khách (client) gửi yêu cầu và máy chủ (server) trả về kết quả. HTTP là giao thức kết nối không trạng thái, có nghĩa là mỗi yêu cầu được gửi riêng biệt và không có thông tin lưu trữ về các yêu cầu trước đó.

Ưu điểm: Đơn giản, dễ sử dụng và triển khai, phù hợp cho các ứng dụng web.

Nhược điểm: Không có bảo mật dữ liệu, thông tin truyền tải có thể bị chặn và thay đổi trong quá trình truyền.

### HTTPS (HTTP Secure)

HTTPS là phiên bản bảo mật của HTTP, sử dụng mã hóa SSL/TLS để bảo vệ dữ liệu truyền tải giữa máy khách và máy chủ. Khi sử dụng HTTPS, một kênh mã hóa

được thiết lập trước khi truyền tải dữ liệu, đảm bảo rằng thông tin không bị nghe trộm hoặc sửa đổi trong quá trình truyền.

HTTPS hoạt động tương tự như HTTP, nhưng trước khi truyền dữ liệu, máy khách và máy chủ sẽ thực hiện bắt tay SSL/TLS để thiết lập kênh mã hóa. Giao thức này sử dụng các chứng chỉ số để xác thực máy chủ và mã hóa dữ liệu bằng khóa công khai.

Ưu điểm: Tăng cường bảo mật và bảo vệ dữ liệu khỏi bị nghe trộm, xác thực tính hợp lệ của máy chủ.

Nhược điểm: Tốc độ chậm hơn HTTP do quá trình mã hóa và giải mã dữ liệu, cần phải có chứng chỉ SSL/TLS, có chi phí triển khai cao hơn.

### Các giao thức tầng trình bày (Tầng 6)

Tầng trình bày chịu trách nhiệm định dạng dữ liệu, mã hóa, và nén dữ liệu. Các giao thức ở tầng này giúp chuyển đổi dữ liệu thành các định dạng có thể hiểu được giữa các hệ thống khác nhau. Ví dụ:

SSL/TLS (Secure Sockets Layer/Transport Layer Security) là các giao thức mã hóa dữ liệu giữa các ứng dụng mạng, giúp bảo vệ dữ liệu khỏi nghe lén và giả mạo. Chúng hoạt động bằng cách thiết lập một phiên bảo mật trước khi truyền dữ liệu, đảm bảo dữ liệu chỉ có thể được giải mã bởi các bên tham gia.

### Các giao thức tầng phiên (Tầng 5)

Tầng phiên quản lý các phiên kết nối giữa hai thiết bị mạng, đảm bảo rằng các

cuộc hội thoại diễn ra liên tục mà không bị gián đoạn. Các giao thức như SMB (Server Message Block) và NetBIOS được sử dụng tại tầng này, đặc biệt trong các hệ thống

chia sẻ tài nguyên và kết nối máy tính với máy chủ.

SMB (Server Message Block) là giao thức cho phép các máy tính trong cùng một mạng chia sẻ tệp, máy in, và các tài nguyên khác. Nó quản lý các phiên kết nối, duy trì kết nối mở để truyền dữ liệu qua lại giữa các máy.

NetBIOS (Network Basic Input/Output System) cung cấp các dịch vụ liên kết và giao tiếp cho các ứng dụng trong mạng LAN, bao gồm đặt tên thiết bị, giao tiếp phiên và truyền dữ liệu. Nó tạo ra các kênh giao tiếp tạm thời giữa các máy để trao đổi dữ liệu.

### Các giao thức tầng giao vận (Tầng 4) TCP (Transmission Control Protocol)

TCP là giao thức kết nối và tin cậy, được sử dụng để đảm bảo rằng dữ liệu truyền tải giữa hai điểm cuối được nhận đầy đủ và đúng thứ tự. Cấu trúc của một gói tin TCP bao gồm phần tiêu đề với các trường như số thứ tự (sequence number), số xác nhận (acknowledgment number), và cờ điều khiển (control flags).

TCP hoạt động theo mô hình ba bước bắt tay (three-way handshake) để thiết lập kết nối. Sau khi kết nối được thiết lập, dữ liệu được chia thành các gói tin, gửi đi và nhận về. TCP sử dụng cơ chế xác nhận (ACK) và điều khiển lưu lượng (flow control) để đảm bảo dữ liệu được truyền tải một cách tin cậy và đầy đủ.

Ưu điểm: Đảm bảo dữ liệu được truyền tải chính xác, có cơ chế phục hồi khi có mất mát gói tin, hỗ trợ kiểm soát lưu lượng và điều chỉnh tốc độ truyền tải.

Nhược điểm: Tốc độ truyền tải chậm hơn do phải đảm bảo độ tin cậy, không thích hợp cho các ứng dụng yêu cầu thời gian thực.

### UDP (User Datagram Protocol)

UDP là giao thức không kết nối, không đảm bảo tin cậy, và không có cơ chế phục hồi lỗi. Mỗi gói tin UDP được gửi độc lập và không có sự xác nhận từ phía

người nhận. Cấu trúc của gói tin UDP đơn giản hơn TCP, chỉ bao gồm phần tiêu đề với các trường cơ bản như cổng nguồn (source port), cổng đích (destination port), và chiều dài gói tin.

UDP gửi các gói tin một cách nhanh chóng mà không cần thiết lập kết nối. Điều này giúp giảm độ trễ nhưng không đảm bảo rằng gói tin sẽ đến nơi hoặc đến đúng thứ tự.

Ưu điểm: Tốc độ nhanh hơn TCP, thích hợp cho các ứng dụng yêu cầu thời gian thực như video call, game online.

Nhược điểm: Không có cơ chế đảm bảo độ tin cậy, dữ liệu có thể bị mất hoặc đến sai thứ tự.

### Các giao thức tầng mạng (Tầng 3)

Giao thức chính tại tầng mạng là IP (Internet Protocol). IP chịu trách nhiệm định tuyến và truyền tải các gói tin từ nguồn đến đích. Có hai phiên bản của IP đang được sử dụng: IPv4 và IPv6. IPv4 sử dụng địa chỉ 32-bit, còn IPv6 sử dụng địa chỉ 128-bit, giúp giải quyết vấn đề cạn kiệt địa chỉ IPv4.

Mặc dù IP là giao thức không tin cậy và không kết nối, nó đảm bảo rằng các gói tin được chuyển đến đúng địa chỉ IP đích, dựa trên thông tin định tuyến.

### Các giao thức tầng liên kết dữ liệu (Tầng 2)

Các giao thức phổ biến ở tầng liên kết dữ liệu bao gồm Ethernet và PPP (Point- to-Point Protocol). Ethernet là giao thức truyền dữ liệu phổ biến trên các mạng LAN (Local Area Network). Nó quản lý việc truyền dữ liệu giữa các thiết bị trên cùng một

mạng vật lý thông qua các gói tin Ethernet, đảm bảo các gói tin được truyền đúng đích thông qua địa chỉ MAC, trong khi PPP được sử dụng chủ yếu trên các kết nối điểm-

điểm.

### Các giao thức tầng vật lí (Tầng 1)

Tầng vật lý liên quan đến các tiêu chuẩn về cáp, tín hiệu điện, và tần số vô tuyến. Các giao thức như IEEE 802.3 cho Ethernet và IEEE 802.11 cho Wi-Fi quy định cách truyền tín hiệu trong môi trường vật lý. Tầng này đảm bảo dữ liệu được chuyển thành các tín hiệu để truyền qua cáp quang, cáp đồng, hoặc sóng vô tuyến.

## Sơ lược về một số công nghệ

### WebRTC

WebRTC, viết tắt của Web Real-Time Communication, là một công nghệ mã nguồn mở cho phép trình duyệt và ứng dụng di động truyền trực tiếp dữ liệu âm thanh, hình ảnh và video mà không cần thông qua máy chủ trung gian. Được phát triển bởi Google vào năm 2011 và sau đó trở thành một chuẩn của W3C và IETF, WebRTC cung cấp khả năng truyền thông tin theo thời gian thực giữa các thiết bị thông qua mô hình ứng dụng mạng ngang hàng (Peer-to-Peer). Điều này giúp các ứng dụng như gọi

video, gọi thoại và chia sẻ tệp tin trực tiếp trên web mà không cần cài đặt thêm phần mềm bên ngoài, đơn giản hóa quá trình kết nối và tương tác.

WebRTC sử dụng một tập hợp các API và giao thức để thiết lập, duy trì và kết thúc kết nối trực tiếp giữa các thiết bị. Các thành phần chính của WebRTC bao gồm:

MediaStream API: Đây là một phần của WebRTC API cho phép truy cập vào các thiết bị đầu vào của người dùng như microphone và camera, thu thập dữ liệu âm thanh và video. MediaStream cho phép WebRTC truy xuất luồng dữ liệu trực tiếp từ

các thiết bị và truyền chúng đến các thiết bị khác, đảm bảo rằng thông tin âm thanh và hình ảnh có thể được truyền tải một cách mượt mà và liên tục.

RTCDataChannel: Là một kênh dữ liệu hai chiều trong WebRTC,

RTCDataChannel chịu trách nhiệm trao đổi dữ liệu thời gian thực. Nó cho phép chia sẻ các loại dữ liệu không chứa âm thanh và video, như tệp tin, văn bản, và hình ảnh, mà không cần thông qua máy chủ trung gian. Khả năng trao đổi dữ liệu của

RTCDataChannel được đánh giá là rất nhanh chóng, an toàn và đáng tin cậy, giúp giảm thiểu độ trễ và chi phí truyền dữ liệu.

RTCPeerConnection: Đây là một API quan trọng trong WebRTC, giúp thiết lập và quản lý kết nối giữa các thiết bị ngang hàng. RTCPeerConnection kết hợp các luồng MediaStream và các kênh dữ liệu từ RTCDataChannel, giúp hai trình duyệt có thể tương tác trực tiếp với nhau mà không cần qua trung gian. Việc sử dụng RTCPeerConnection giúp tối ưu hóa quy trình kết nối và đảm bảo chất lượng cao cho các cuộc gọi video, thoại và truyền tải dữ liệu.

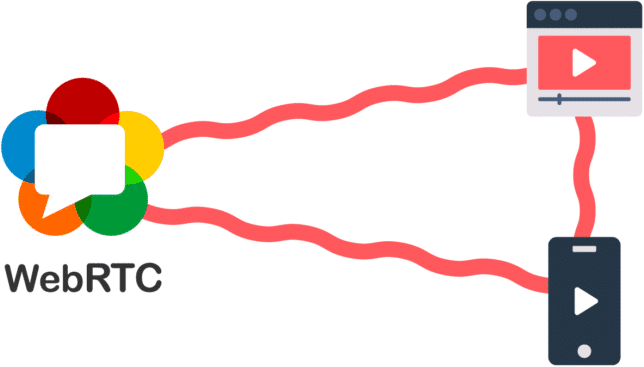
Ưu điểm lớn nhất của WebRTC là không yêu cầu cài đặt phần mềm bên ngoài, cho phép người dùng dễ dàng truy cập và sử dụng các tính năng trực tiếp từ trình duyệt. Điều này không chỉ đơn giản hóa trải nghiệm người dùng mà còn mở rộng khả năng tương thích trên nhiều nền tảng. WebRTC cung cấp độ trễ thấp và chất lượng cao nhờ khả năng truyền dữ liệu trực tiếp theo thời gian thực, đảm bảo trải nghiệm âm thanh, video rõ nét và liên tục. Ngoài ra, hệ thống còn tích hợp mã hóa dữ liệu end-to- end (E2E), giúp bảo vệ thông tin khỏi sự truy cập trái phép. Tính linh hoạt và dễ dàng tích hợp của WebRTC cũng cho phép nó hỗ trợ nhiều mô hình truyền thông, từ hội nghị video đa chiều đến truyền tải dữ liệu P2P.

Tuy nhiên, WebRTC cũng có những nhược điểm. Chất lượng cuộc gọi và truyền dữ liệu phụ thuộc vào kết nối internet, nên khi mạng không ổn định, trải nghiệm người dùng có thể bị ảnh hưởng đáng kể. Dù WebRTC tích hợp mã hóa cho luồng dữ liệu, quá trình signaling - vốn là bước thiết lập kết nối - không được quy định cụ thể, có thể tạo ra lỗ hổng bảo mật nếu không được xử lý đúng cách. Một thách thức khác là khả năng tương thích giữa các trình duyệt. Dù các trình duyệt hiện đại đã hỗ trợ WebRTC,

một số tính năng có thể không đồng nhất trên các nền tảng khác nhau, khiến việc phát triển và triển khai ứng dụng gặp khó khăn.

Với những đặc điểm kỹ thuật mạnh mẽ và linh hoạt, WebRTC đã trở thành một công nghệ cốt lõi trong các ứng dụng truyền thông trực tuyến hiện đại. Công nghệ này được ứng dụng rộng rãi trong nhiều lĩnh vực, từ hội nghị video như Google Meet,

Zoom và Microsoft Teams, đến các dịch vụ trò chuyện video như WhatsApp, Facebook Messenger, mang lại sự tiện lợi và nâng cao trải nghiệm cho người dùng trên toàn cầu.



**Hình 1.4.1. Công nghệ WebRTC**

### ElectronJs

ElectronJs là một framework mã nguồn mở được phát triển bởi GitHub, cho phép các nhà phát triển xây dựng các ứng dụng desktop đa nền tảng bằng cách sử dụng công nghệ web quen thuộc như HTML, CSS, và JavaScript. ElectronJs kết hợp công cụ trình duyệt Chromium với Node.js, cho phép ứng dụng có thể chạy trên các hệ điều hành phổ biến như Windows, macOS và Linux. Với ElectronJs, các nhà phát triển có thể tạo ra các ứng dụng desktop có giao diện và trải nghiệm tương tự như các ứng dụng web, nhưng vẫn tận dụng được các tài nguyên và tính năng của hệ điều hành, như truy cập hệ thống tệp, thông báo đẩy, và tích hợp với các ứng dụng khác.

ElectronJs cũng cung cấp khả năng dễ dàng tích hợp với các thư viện và module từ Node.js. Điều này giúp các nhà phát triển tận dụng hệ sinh thái phong phú của Node.js để mở rộng tính năng của ứng dụng mà không cần học thêm ngôn ngữ hoặc

framework mới. Các nhà phát triển có thể sử dụng các module như fs để quản lý tệp,

http để xây dựng các máy chủ đơn giản, hoặc các thư viện bên thứ ba như axios để thực hiện các yêu cầu HTTP một cách dễ dàng.

Tuy nhiên, ElectronJs cũng có một số nhược điểm đáng lưu ý. Một trong những vấn đề thường được nhắc đến là kích thước ứng dụng lớn và mức tiêu thụ bộ nhớ cao hơn so với các ứng dụng native. Điều này là do ElectronJs tích hợp Chromium và Node.js vào mỗi ứng dụng, dẫn đến dung lượng tải xuống lớn hơn và yêu cầu nhiều tài nguyên hệ thống hơn khi chạy. Ngoài ra, việc phụ thuộc vào các công nghệ web cũng có thể dẫn đến hiệu suất không ổn định, đặc biệt là với các ứng dụng yêu cầu xử lý đồ họa phức tạp hoặc thực thi nhiều tác vụ cùng lúc. Các vấn đề này đã được cải thiện dần theo thời gian, nhưng vẫn là điều cần cân nhắc khi quyết định sử dụng ElectronJs cho một dự án.



**Hình 1.4.2. Công nghệ Electron**

### ReactJS

ReactJS là một thư viện mã nguồn mở được sử dụng để xây dựng giao diện người dùng cho các ứng dụng web. Được phát triển bởi Facebook, ReactJS cho phép xây dựng các ứng dụng với giao diện người dùng phản ứng nhanh và dễ dàng mở rộng.

Với việc sử dụng JavaScript làm ngôn ngữ chính, ReactJS giúp các nhà phát triển xây dựng các thành phần giao diện tái sử dụng được (components), từ đó tối ưu hóa mã nguồn và nâng cao hiệu suất phát triển.

Một trong những điểm mạnh của ReactJS là mô hình dựa trên "Virtual DOM" (Virtual Document Object Model), giúp tăng hiệu suất bằng cách chỉ cập nhật những phần tử thay đổi thay vì toàn bộ giao diện. Điều này giúp cải thiện tốc độ và trải nghiệm người

dùng. ReactJS cũng dễ dàng tích hợp với các thư viện và công nghệ khác, như Redux hoặc Axios, giúp phát triển ứng dụng trở nên linh hoạt và dễ dàng mở rộng.

ReactJS tuân thủ các nguyên tắc lập trình hiện đại như tính tái sử dụng mã nguồn (component-based architecture), cho phép các nhà phát triển xây dựng các giao diện

người dùng dễ bảo trì và dễ mở rộng. Hệ sinh thái của ReactJS rộng lớn, cung cấp một loạt các công cụ và thư viện hỗ trợ phát triển, từ việc quản lý trạng thái ứng dụng đến việc tối ưu hóa hiệu suất. Với sự hỗ trợ mạnh mẽ từ cộng đồng, ReactJS là một công cụ tuyệt vời cho việc phát triển các ứng dụng web hiện đại và phức tạp.



**Hình 1.4.3. Framework ReactJS**

### SocketIO

Socket.IO là một thư viện mạnh mẽ và phổ biến dành cho việc xây dựng các ứng dụng thời gian thực trên nền tảng Node.js. Được thiết kế để giúp các nhà phát triển dễ dàng tạo ra các kết nối hai chiều giữa máy chủ và máy khách, Socket.IO cho phép truyền tải dữ liệu ngay lập tức mà không cần chờ đợi các yêu cầu HTTP thông thường. Sự ra đời của Socket.IO đã mở ra một hướng đi mới cho việc phát triển các ứng dụng như chat trực tuyến, cập nhật dữ liệu theo thời gian thực, trò chơi đa người chơi và các ứng dụng truyền phát dữ liệu trực tiếp khác.

Ưu điểm lớn nhất của Socket.IO là khả năng hỗ trợ kết nối thời gian thực với độ trễ thấp. Điều này làm cho nó trở thành sự lựa chọn lý tưởng cho các ứng dụng yêu cầu tốc độ phản hồi nhanh và tính tương tác cao, như trò chơi trực tuyến, hệ thống giao dịch tài chính, và các dịch vụ hỗ trợ khách hàng. Bên cạnh đó, việc tự động chuyển đổi giữa các giao thức giúp đảm bảo kết nối ổn định ngay cả khi môi trường mạng không tối ưu. Socket.IO cũng dễ dàng tích hợp với các hệ thống hiện có và có

thể hoạt động trên nhiều nền tảng khác nhau, từ máy tính để bàn đến di động.



**Hình 1.4.4. Thư viện Socket.io**

# CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

## Giới thiệu sơ lược về bài toán

Bài toán đặt ra là xây dựng một hệ thống mạng xã hội thời gian thực, nơi nhiều client (người dùng) có thể tương tác các client khác thông qua server.

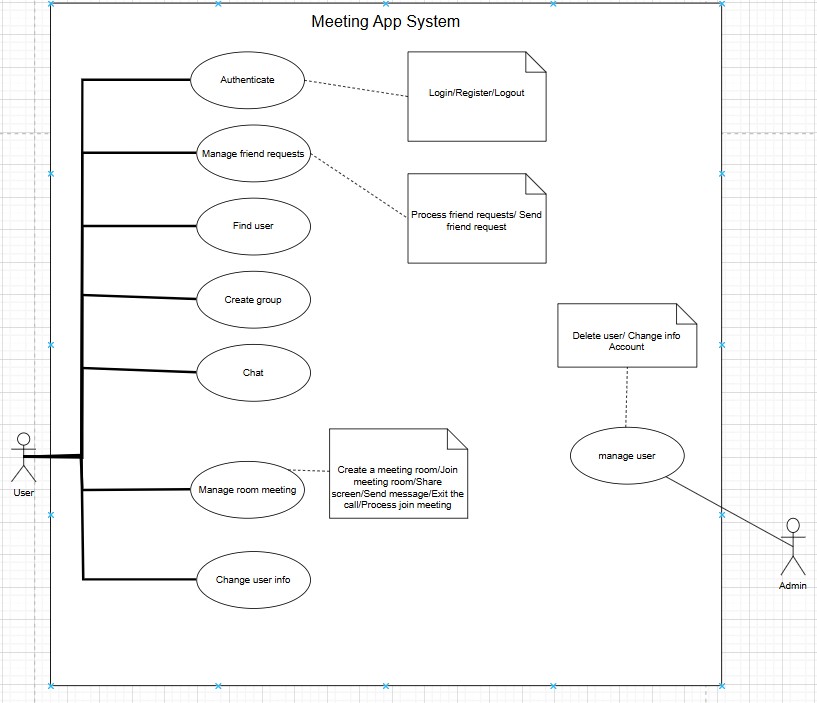
Chương trình cần đáp ứng nhu cầu cơ bản của một nền tảng mạng xã hội, bao gồm việc người dùng có thể đăng ký tài khoản và đăng nhập vào hệ thống. Thông qua tài khoản này, người dùng sẽ được cung cấp các tính năng như trò chuyện, chia sẻ tài liệu và tham gia họp trực tuyến.

Hệ thống cũng cho phép người dùng tạo ra các nhóm riêng, nơi họ có thể đăng các nội dung chỉ dành cho các thành viên trong nhóm. Chủ nhóm sẽ có quyền quản lý các yêu cầu gia nhập nhóm từ những người dùng khác, cũng như có khả năng loại bỏ thành viên trong nhóm.

Người dùng có thể tìm kiếm các người dùng khác để bắt đầu cuộc trò chuyện và từ đó có thể thêm thành viên vào cuộc trò chuyện đó. Những người trong cuộc trò chuyện có thể nhắn tin, gọi video call với nhau.

Phần server của hệ thống đóng vai trò quan trọng trong việc lưu trữ dữ liệu, xử lý các yêu cầu từ phía client và gửi thông báo đến người dùng khi có tương tác mới trên bài viết hoặc khi có yêu cầu gia nhập nhóm. Server cần đảm bảo khả năng xử lý đồng thời các yêu cầu từ nhiều người dùng mà vẫn duy trì hiệu suất cao, đồng thời bảo vệ dữ liệu và bảo mật thông tin cá nhân của người dùng.

## Biểu đồ use case



**Bảng 1.2.2. Biểu đồ use case**

## Đặc tả Use Case

### Use Case Register

|  |  |
| --- | --- |
| Tên Use-case | Register |
| Tác nhân | User |
| Mô tả | Người dùng đăng ký tài khoản. |

|  |  |
| --- | --- |
| Điều kiện trước | Người dùng vào ứng dụng. |
| Điều kiện sau | Tài khoản mới được tạo. |
| Bối cảnh | Người dùng muốn đăng ký tài khoản mới trong ứng dụng. |
| Luồng chính | Hành động của Người dùng:  Người dùng nhập thông tin tài khoản.  Bấm nút đăng ký. Hành động của hệ thống:  Tài khoản mới được thêm vào cơ sở dữ liệu.  Điều hướng vào trang chủ. |

**Bảng 2.3.1. Use Case Register**

### Use Case Login

|  |  |
| --- | --- |
| Tên Use-case | Login |
| Tác nhân | User |
| Mô tả | Người dùng đăng nhập vào hệ thống. |
| Điều kiện trước | Người dùng vào ứng dụng. |
| Điều kiện sau | Đăng nhập thành công. |
| Bối cảnh | Người dùng muốn đăng nhập vào ứng dụng. |
| Luồng chính | Hành động của Người dùng:  Người dùng nhập thông tin đăng nhập.  Người dùng bấm nút đăng nhập.  Hành động của hệ thống:  Trả về token để lưu lần đăng nhập sau. |

**Bảng 2.3.2. Use Case Login**

### Use Case Logout

|  |  |
| --- | --- |
| Tên Use-case | Logout |
| Tác nhân | User |
| Mô tả | Người dùng đăng nhập vào hệ thống. |
| Điều kiện trước | Người dùng vào ứng dụng. |
| Điều kiện sau | Đăng nhập thành công. |
| Bối cảnh | Người dùng muốn thoát ra khỏi ứng dụng. |
| Luồng chính | Hành động của Người dùng:  Người dùng nhập thông tin đăng nhập.  Người dùng bấm nút đăng nhập.  Hành động của hệ thống:  Trả về token để lưu lần đăng nhập sau. |

* + 1. **Use Case Forget password**

|  |  |
| --- | --- |
| Tên Use-case | Forget password |
| Tác nhân | User |
| Mô tả | Người dùng yêu cầu đặt lại mật khẩu khi quên mật khẩu đăng nhập vào hệ thống. |
| Điều kiện trước | Người dùng đã có tài khoản trong hệ thống.  Người dùng phải đăng nhập vào hệ thống (hoặc không thể đăng nhập vì quên mật khẩu). |
| Điều kiện sau | Mật khẩu được cập nhật thành công.  Người dùng có thể đăng nhập với mật khẩu mới. |
| Bối cảnh | Người dùng không thể đăng nhập vào hệ thống vì quên mật khẩu. Họ cần yêu cầu hệ thống gửi liên kết đặt lại mật khẩu hoặc mã xác nhận để có thể cập nhật mật khẩu mới. |

|  |  |
| --- | --- |
| Luồng chính | Hành động của Người dùng:  Người dùng vào trang "Quên mật khẩu" và nhập email hoặc số điện thoại đã đăng ký.  Nhấn nút "Gửi yêu cầu đặt lại mật khẩu".  Nhập mã xác nhận hoặc nhấn vào liên kết và nhập mật khẩu mới.  Hành động của hệ thống:  Xác thực email hoặc số điện thoại.  Gửi mã xác nhận hoặc liên kết để đặt lại mật khẩu.  Kiểm tra mã xác nhận, cập nhật mật khẩu mới trong cơ sở dữ  liệu.  Hiển thị thông báo thành công và điều hướng người dùng đến trang đăng nhập. |

**Bảng 2.3.3. Use Case Forget Password**

### Use Case Chat

|  |  |
| --- | --- |
| Tên Use-case | Chat |
| Tác nhân | User |
| Mô tả | Người dùng gửi tin nhắn. |
| Điều kiện trước | Người dùng mở hộp chat. |
| Điều kiện sau | Hệ thống tải các tin nhắn cũ trong |

|  |  |
| --- | --- |
|  | hộp chat (nếu có).  Tin nhắn được gửi thành công và lưu trữ trong cơ sở dữ liệu.  Người nhận nhận được tin nhắn trong giao diện hộp chat. |
| Bối cảnh | Người dùng sử dụng ứng dụng để gửi tin nhắn tới bạn bè hoặc nhóm chat trong giao diện chat của ứng dụng.  Người dùng có thể nhập và gửi tin nhắn văn bản, hình ảnh hoặc file. |
| Luồng chính | Hành động của Người dùng:  Người dùng chọn người muốn trò chuyện cùng.  Người dùng nhập tin nhắn. Người dùng nhấn nút gửi.  Hành động của hệ thống:  Lưu tin nhắn vào CSDL. Hiển thị ra app. |

**Bảng 2.3.4. Use Case Send Message**

* + 1. **Use Case Create Meeting Room**

|  |  |
| --- | --- |
| Tên Use-case | Create Meeting Room |
| Tác nhân | User |
| Mô tả | Người dùng tạo một cuộc họp để giao tiếp với các thành viên khác |
| Điều kiện trước | Người dùng đăng nhập vào hệ thống |
| Điều kiện sau | Cuộc họp được tạo thành công, và người dùng trở thành quản trị viên của phòng họp.  Giao diện phòng họp hiển thị để  người dùng có thể mời người tham gia. |
| Bối cảnh | Người dùng muốn tạo một cuộc họp trực tuyến để giao tiếp và mời người tham gia. Hệ thống sẽ tạo phòng họp, lưu thông tin và cung cấp giao diện  để người dùng mời người tham gia. |

|  |  |
| --- | --- |
| Luồng chính | Hành động của Người dùng:  Người dùng bấm vào nút “Bắt đầu cuộc họp tức thì”  Hành động của hệ thống:  Hệ thống tạo một phòng họp mới và lưu thông tin phòng vào cơ sở dữ liệu.  Hiển thị giao diện phòng họp vừa tạo để người dùng có thể bắt đầu mời người tham gia.  Tạo một liên kết phòng họp  hoặc mã phòng để người dùng có thể chia sẻ với những người tham gia. |

* + 1. **Use Case Exit Room**

|  |  |
| --- | --- |
| Tên Use-case | Exit Room |
| Tác nhân | User |
| Mô tả | Thoát khỏi phòng đang tham gia. |
| Điều kiện trước | Người dùng đã đăng nhập vào hệ thống và đang tham gia một phòng họp. |
| Điều kiện sau | Người dùng thoát khỏi phòng họp, không còn tham gia vào phòng đó. |
| Bối cảnh | Người dùng không còn họp nữa hoặc không muốn than gia phòng họp |
| Luồng chính | Hành động của Người dùng: Nhấn vào nút "Thoát".  Hành động của hệ thống:  Cập nhật thông tin để loại bỏ người dùng ra khỏi danh sách thành viên đang tham gia phòng họp.  Lưu thay đổi vào cơ sở dữ liệu.  Hiển thị thông báo xác nhận rằng người dùng đã rời khỏi phòng họp và điều hướng  người dùng trở lại trang chính |

|  |  |
| --- | --- |
|  | hoặc danh sách các cuộc họp. |

* + 1. **Use Case Join Meeting Room**

|  |  |
| --- | --- |
| Tên Use-case | Join Meeting Room |
| Tác nhân | User |
| Mô tả | Người dùng tham gia vào một cuộc họp trực tuyến để trao đổi thông tin. |
| Điều kiện trước | Người dùng đã đăng nhập vào hệ thống và có mã phòng họp hoặc được mời tham gia người dùng đã kết bạn |
| Điều kiện sau | Người dùng tham gia thành công vào phòng họp và có thể trao đổi thông tin với các thành viên khác. |
| Bối cảnh | Người dùng muốn tham gia vào một cuộc họp trực tuyến đã được lên lịch hoặc được mời tham gia, và hệ thống cung cấp giao diện để tham gia và  tương tác trong cuộc họp. |

|  |  |
| --- | --- |
| Luồng chính | Hành động của Người dùng:  Nhập mã phòng họp hoặc chọn phòng họp từ danh sách các cuộc họp đã được mời.  Nhấn nút "Tham gia" để vào phòng họp.  Người dùng được chuyển vào giao diện cuộc họp, nơi có thể trao đổi trực tiếp với các thành viên khác qua âm thanh, video, hoặc tin nhắn.  Hành động của hệ thống:  Xác thực mã phòng họp hoặc thông tin mời tham gia.  Kết nối người dùng vào phòng họp và cập nhật danh sách  người tham gia.  Hiển thị giao diện phòng họp cho người dùng.  Lưu lại lịch sử tham gia vào cơ sở dữ liệu. |

* + 1. **Use Case Send Friend Request**

|  |  |
| --- | --- |
| Tên Use-case | Send Friend Request |
| Tác nhân | User |

|  |  |
| --- | --- |
| Mô tả | Gửi lời lời kết bạn cho người dùng khác |
| Điều kiện trước | Người dùng đăng nhập vào hệ thống |
| Điều kiện sau | Gửi lời mời thành công |
| Bối cảnh | Người dùng muốn kết bạn với một người dùng khác và gửi yêu cầu kết bạn qua hệ thống. |
| Luồng chính | Hành động của Người dùng:  Người dùng tìm kiếm và chọn người muốn kết bạn.  Người dùng nhấn nút "Gửi lời mời kết bạn".  Hành động của hệ thống:  Xác nhận lời mời kết bạn của người dùng.  Gửi thông báo cho người dùng được mời kết bạn.  Lưu thông tin yêu cầu kết bạn vào cơ sở dữ liệu. |

* + 1. **Use Case Find User**

|  |  |
| --- | --- |
| Tên Use-case | Find User |
| Tác nhân | User |

|  |  |
| --- | --- |
| Mô tả | Người dùng tìm kiếm người khác để thực hiện các hành vi như kết bạn, mời tham gia nhóm, hoặc các hành động khác. |
| Điều kiện trước | Người dùng đã đăng nhập vào hệ thống. |
| Điều kiện sau | Danh sách người dùng và thông tin  chi tiết của người dùng được hiển thị để người dùng có thể xem xét và thực hiện các hành động tiếp theo. |
| Bối cảnh | Người dùng muốn tìm kiếm và xem thông tin chi tiết của người dùng  khác trong hệ thống. |
| Luồng chính | Hành động của người dùng:  Người dùng nhập từ khóa tìm kiếm (tên, email, số điện thoại, v.v.) vào công cụ tìm kiếm.  Hành động của Hệ thống:  Hệ thống tìm kiếm và hiển thị danh sách người dùng phù hợp với từ khóa tìm kiếm.  Hệ thống hiển thị thông tin chi tiết của người dùng được chọn (bao gồm tên, thông tin liên hệ, v.v.). |

**Bảng 2.3.9. Tìm người dùng**

* + 1. **Use Case Change User Info**

|  |  |
| --- | --- |
| Tên Use-case | Change User Info |
| Tác nhân | User |
| Mô tả | Người dùng thay đổi thông tin cá nhân trong hệ thống. |
| Điều kiện trước | Người dùng đăng nhập vào hệ thống. |
| Điều kiện sau | Thông tin cá nhân của người dùng được cập nhật thành công. |
| Bối cảnh | Người dùng muốn cập nhật các thông tin cá nhân của mình như tên, email, số điện thoại, v.v. |
| Luồng chính | Hành động của Người dùng:  Người dùng truy cập trang "Cập nhật thông tin cá nhân".  Người dùng thay đổi thông tin mong muốn (ví dụ: tên, email, mật khẩu, số điện thoại).  Người dùng nhấn nút "Lưu thay đổi".  Hành động của Hệ thống:  Hệ thống xác nhận các thông |

|  |  |
| --- | --- |
|  | tin đã thay đổi.  Hệ thống cập nhật các thông tin mới vào cơ sở dữ liệu.  Hệ thống thông báo thành công và hiển thị thông tin đã được cập nhật. |
| Luồng phụ | Luồng phụ 1: Thông tin không hợp lệ  Người dùng: Nhập thông tin không  hợp lệ (ví dụ: email không đúng định dạng, số điện thoại thiếu ký tự).  Hệ thống: Hiển thị thông báo lỗi cho người dùng, chỉ rõ trường thông tin nào cần sửa đổi.  Kết quả:  Người dùng phải sửa thông tin và thử lại. |

* + 1. **Use Case Share Screen**

|  |  |
| --- | --- |
| Tên Use-case | Share Screen |
| Tác nhân | User |
| Mô tả | Người dùng chia sẻ màn hình với các người tham gia khác trong phòng họp trực tuyến. |
| Điều kiện trước | Người dùng đã đăng nhập vào hệ |

|  |  |
| --- | --- |
|  | thống và tham gia vào một phòng họp trực tuyến. |
| Điều kiện sau | Màn hình của người dùng được chia sẻ thành công. |
| Bối cảnh | Người dùng muốn chia sẻ màn hình của mình trong một cuộc họp trực tuyến để trao đổi thông tin hoặc trình bày. |
| Luồng chính | Hành động của Người dùng:  Người dùng đăng nhập vào hệ thống và tham gia vào phòng họp trực tuyến.  Người dùng chọn nút "Bắt đầu chia sẻ" để chia sẻ màn hình  của mình.  Người dùng có thể bấm nút "Dừng chia sẻ" khi muốn kết thúc việc chia sẻ màn hình.  Hành động của Hệ thống:  Nếu người dùng chọn "Bắt đầu chia sẻ", hệ thống sẽ chia sẻ nội dung màn hình của  người dùng cho tất cả các thành viên trong phòng họp.  Nếu người dùng chọn "Dừng chia sẻ", hệ thống sẽ ngừng  chia sẻ và kết thúc phiên chia sẻ màn hình. |

|  |  |
| --- | --- |
|  |  |

* + 1. **Use Case Meeting Chat**

|  |  |
| --- | --- |
| Tên Use-case | Meeting Chat |
| Tác nhân | User |
| Mô tả | Người dùng sử dụng ô trò chuyện để gửi văn bản trong phòng họp trực tuyến. |
| Điều kiện trước | Người dùng đăng nhập vào hệ thống và đã tham gia vào một phòng họp  trực tuyến. |
| Điều kiện sau | Tin nhắn của người dùng được gửi thành công và hiển thị trong phòng trò chuyện của cuộc họp. |
| Bối cảnh | Người dùng muốn gửi tin nhắn văn bản trong cuộc họp để trao đổi thông tin với các thành viên khác. |

|  |  |
| --- | --- |
| Luồng chính | 1. **Người dùng:**   Tham gia vào phòng họp trực tuyến.  Nhập nội dung tin nhắn vào ô trò chuyện và nhấn nút "Gửi".   1. **Hệ thống:**    * Gửi tin nhắn đến tất cả các thành viên hiện đang tham gia phòng.    * Hiển thị tin nhắn trong giao diện trò chuyện   của từng thành viên trong thời gian thực.   1. **Người dùng khác:** Xem tin nhắn trong giao diện phòng họp của họ. |

* + 1. **Use Case Process Join Meeting**

|  |  |
| --- | --- |
| Tên Use-case | Process Join Meeting |
| Tác nhân | User (Owner of the Meeting) |
| Mô tả | Xử lý yêu cầu tham gia cuộc họp được gửi tới. |
| Điều kiện trước | Người chủ phòng đã đăng nhập vào hệ thống và một yêu cầu tham gia |

|  |  |
| --- | --- |
|  | cuộc họp đã được gửi tới. |
| Điều kiện sau | Yêu cầu tham gia cuộc họp được chấp nhận hoặc từ chối. |
| Bối cảnh | Người chủ phòng cần xử lý yêu cầu tham gia cuộc họp từ các thành viên khác. |
| Luồng chính | Hành động của Người dùng:  Người chủ phòng đăng nhập vào hệ thống.  Người chủ phòng xem danh sách các yêu cầu tham gia  cuộc họp.  Người chủ phòng chọn một yêu cầu và nhấn nút "Chấp nhận" hoặc "Từ chối".  Hành động của Hệ thống:  Nếu yêu cầu được chấp nhận, hệ thống thêm người gửi yêu cầu vào phòng họp và xóa yêu cầu khỏi danh sách.  Nếu yêu cầu bị từ chối, hệ thống hủy bỏ yêu cầu và thông báo cho người gửi yêu cầu rằng họ không được chấp nhận, đồng thời xóa yêu cầu khỏi danh sách. |

**Bảng 3.2-10 Xử lí yêu cầu tham gia cuộc họp**

### Use Case Process Friend Request

|  |  |
| --- | --- |
| Tên Use-case | Process Friend Request |
| Tác nhân | User |
| Mô tả | Xử lý yêu cầu kết bạn được gửi tới |
| Điều kiện trước | Người dùng đăng nhập vào hệ thống và có yêu cầu kết bạn đang chờ xử lý. |
| Điều kiện sau | Yêu cầu kết bạn được chấp nhận hoặc từ chối thành công. |
| Bối cảnh | Người dùng nhận được yêu cầu kết bạn và phải xử lý (chấp nhận hoặc từ chối) yêu cầu đó. |
| Luồng chính | Hành động của Người dùng:  Người dùng vào trang "Yêu cầu kết bạn" để xem các yêu cầu  Người dùng nhấn nút "Chấp nhận" hoặc "Từ chối" yêu cầu  Hành động của Hệ thống:  Nếu người dùng chọn "Chấp nhận", yêu cầu kết bạn sẽ  được thêm vào danh sách bạn bè và xóa khỏi danh sách yêu cầu.  Nếu người dùng chọn "Từ chối", yêu cầu kết bạn sẽ bị |

|  |  |
| --- | --- |
|  | xóa khỏi hệ thống. |

**Bảng 3.2-10 Xử lí yêu cầu kết bạn**

### Use Case Create Group Chat

|  |  |
| --- | --- |
| Tên Use-case | Create Group Chat |
| Tác nhân | User |
| Mô tả | Người dùng tạo một nhóm trò chuyện để giao tiếp với nhiều người dùng cùng lúc. |
| Điều kiện trước | Người dùng đã đăng nhập vào hệ thống và có danh sách bạn bè. |
| Điều kiện sau | Nhóm trò chuyện được tạo thành công và người dùng có thể tham gia vào nhóm để giao tiếp. |
| Bối cảnh | Người dùng muốn tạo một nhóm trò chuyện mới và thêm những người bạn vào nhóm. |
| Luồng chính | Hành động của Người dùng:  Người dùng đăng nhập vào hệ thống.  Người dùng vào trang nhóm và chọn "Tạo nhóm trò chuyện".  Người dùng chọn các thành viên muốn mời tham gia nhóm.  Người dùng nhấn nút "Tạo nhóm". |

|  |  |
| --- | --- |
|  | Hành động của Hệ thống:  Hệ thống tạo nhóm trò chuyện mới..  Hệ thống thêm các thành viên vào nhóm.  Hiển thị giao diện nhóm trò chuyện để người dùng có thể bắt đầu giao tiếp. |

**Bảng 3.2-10 Xử lí yêu cầu kết bạn**

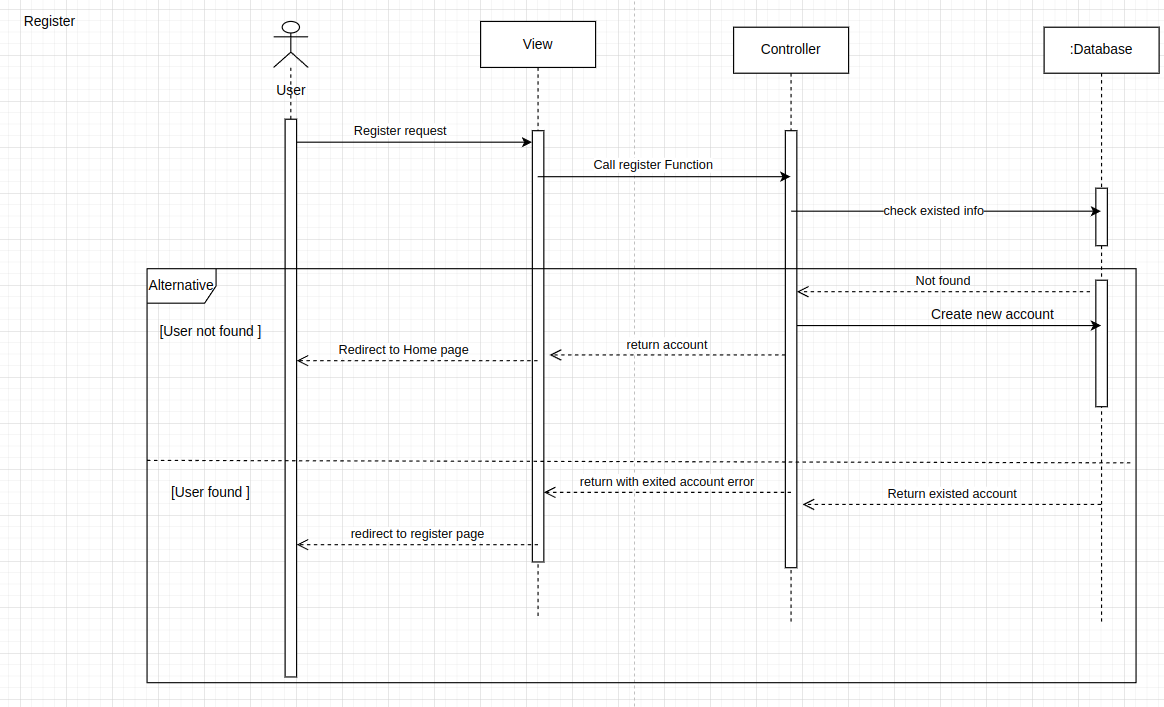
* + 1. **Use Case Delete User**

|  |  |
| --- | --- |
| Tên Use-case | Delete user |
| Tác nhân | Admin |
| Mô tả | Admin xử lý yêu cầu xóa tài khoản người dùng. |
| Điều kiện trước | Admin đăng nhập vào hệ thống, có yêu cầu xóa tài khoản người dùng. |
| Điều kiện sau | Tài khoản người dùng được xóa thành công, hoặc yêu cầu bị từ chối. |
| Bối cảnh | Admin nhận được yêu cầu xóa tài khoản từ người dùng và xử lý yêu cầu này. |
| Luồng chính | Hành động của Admin:  Admin đăng nhập vào hệ thống.  Admin xem danh sách yêu cầu xóa tài khoản.  Admin chọn yêu cầu xóa tài khoản cần xử lý.  Admin nhấn nút "Chấp nhận" hoặc "Từ chối".  Hành động của Hệ thống:  Nếu chọn "Chấp nhận", hệ thống xóa tài khoản người dùng khỏi cơ sở dữ liệu và |

|  |  |
| --- | --- |
|  | thông báo thành công.  Nếu chọn "Từ chối", hệ thống xóa yêu cầu xóa tài khoản và thông báo từ chối. |

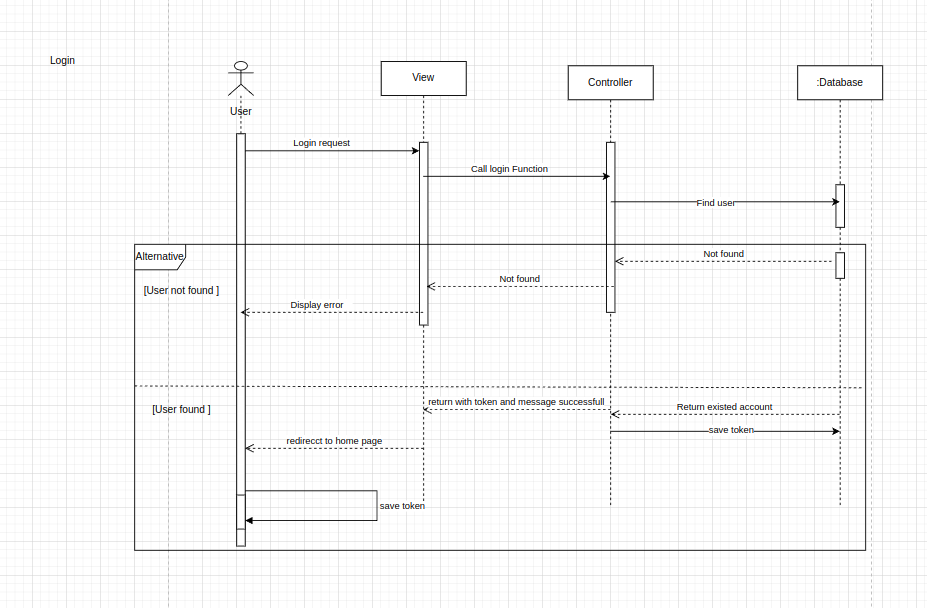
## Biểu đồ tuần tự

### Sequence Diagram Register



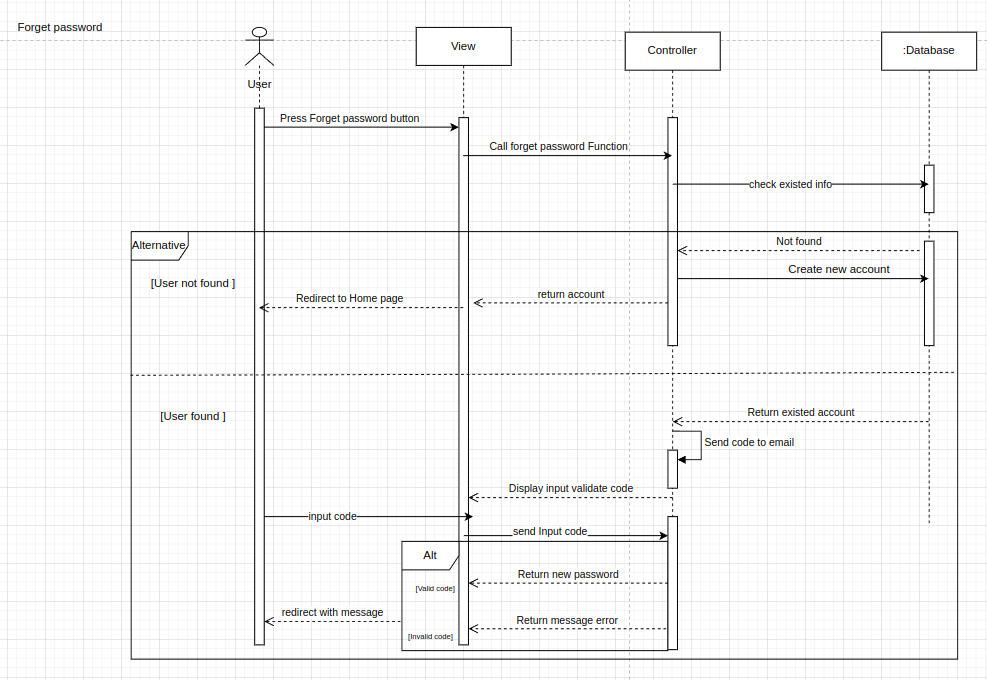
**Hình 3.4.1 Sequence Diagram Register**

### Sequence Diagram Login



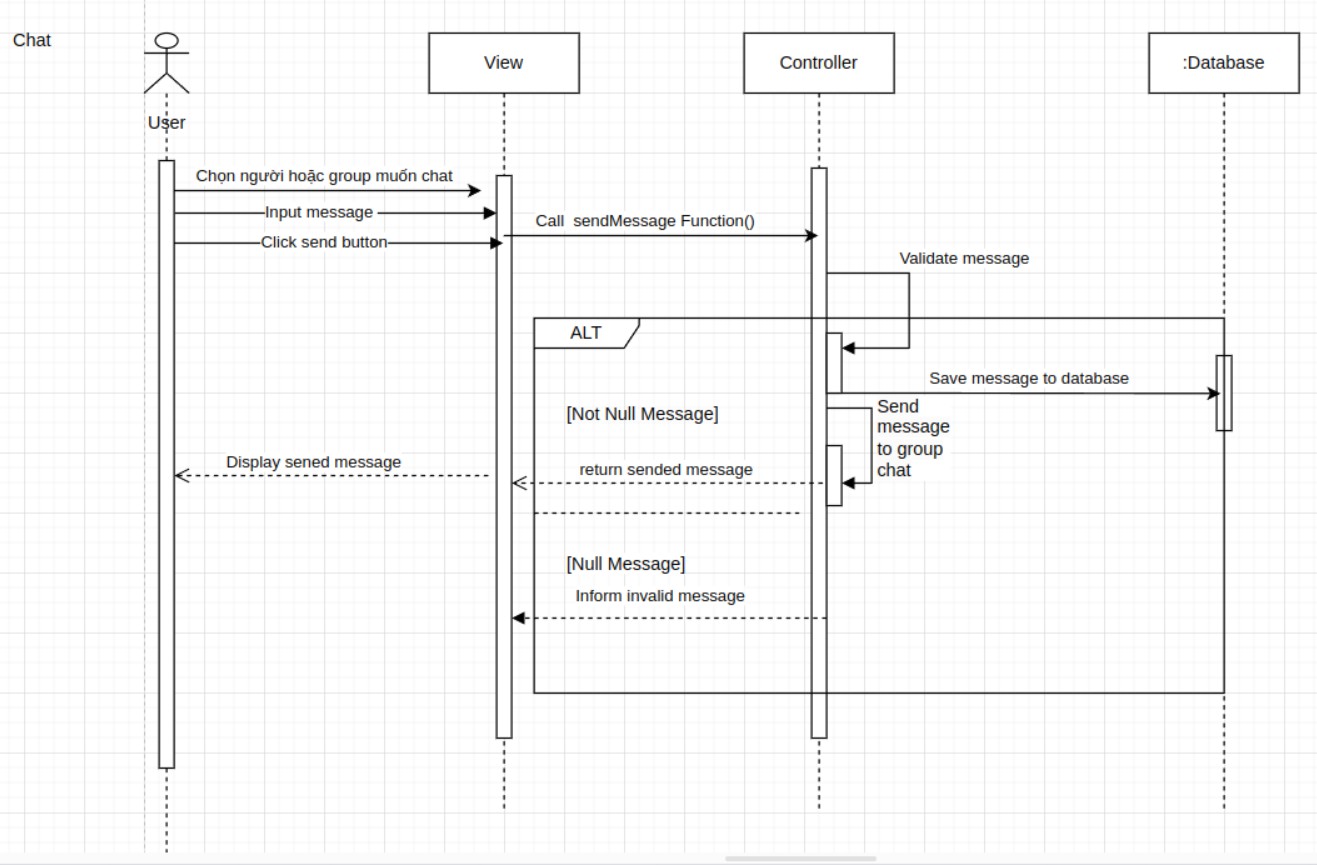
**Hình 3.4.2 Sequence Diagram Login**

### Sequence Diagram Forget Password



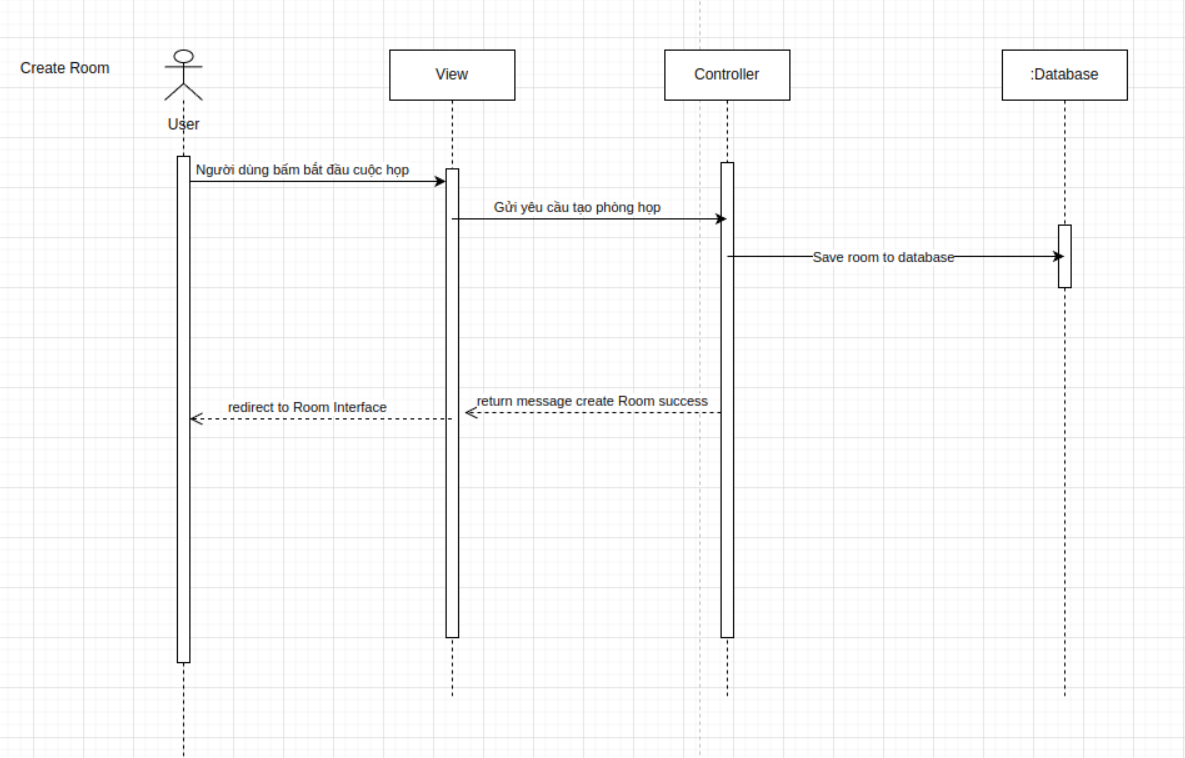
**Hình 3.4.3 Sequence Diagram Forget Password**

### Sequence Diagram Chat



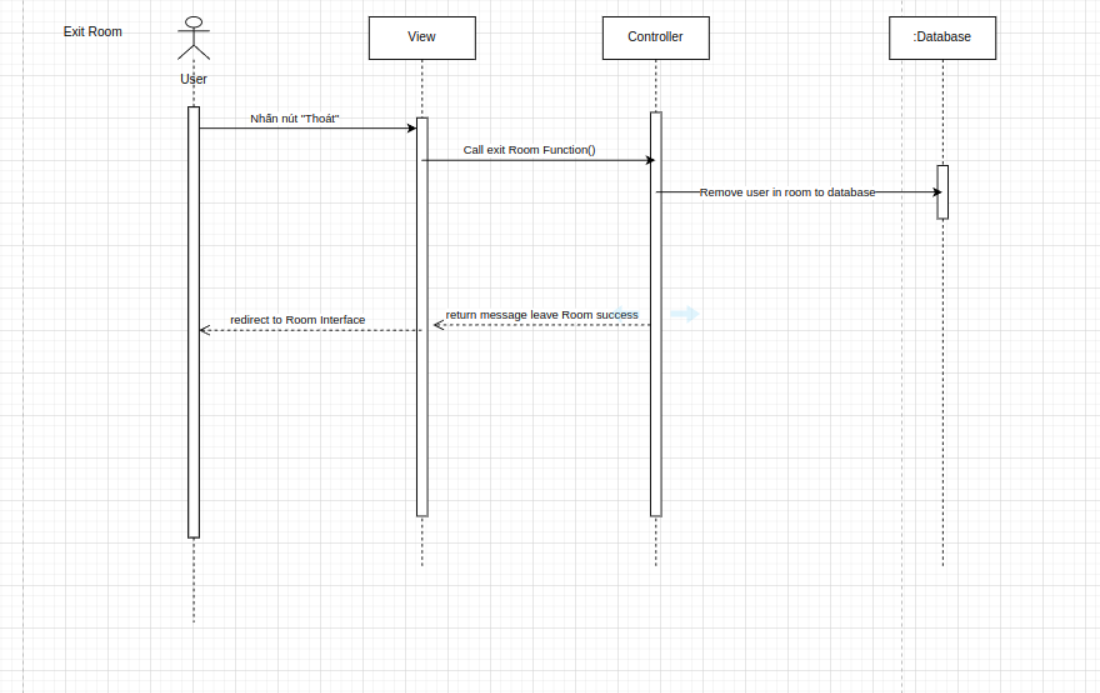
**Hình 3.4.4 Sequence Diagram Chat**

### Sequence Diagram Create Meeting Room



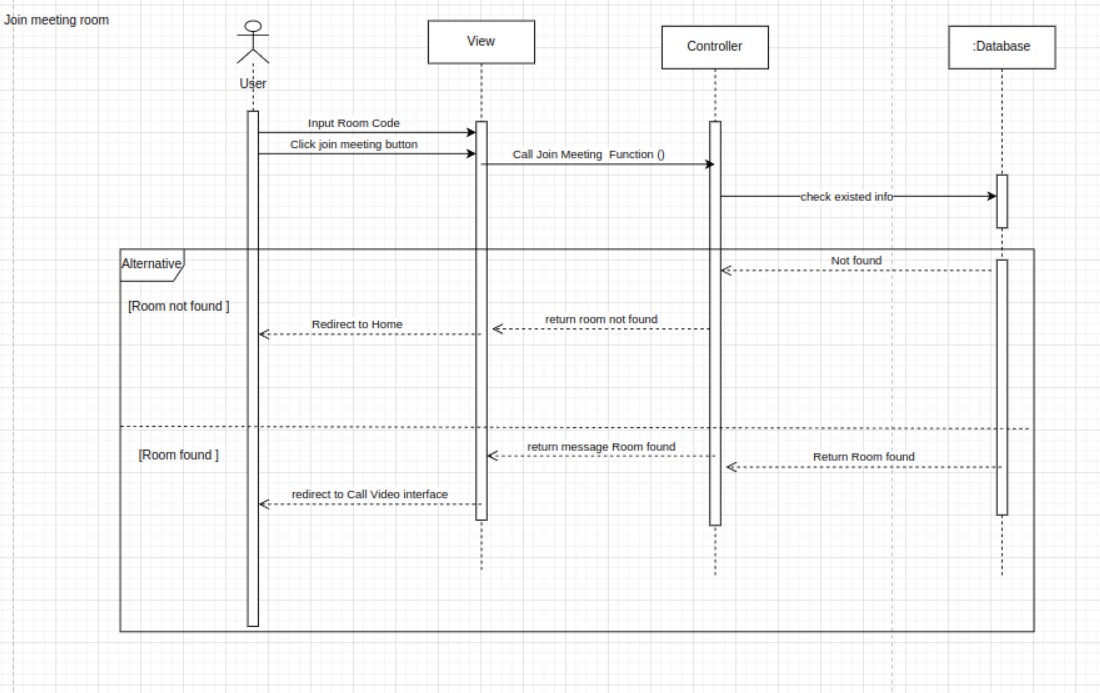
**Hình 3.4.5 Biểu đồ tuần tự tạo phòng họp**

### Sequence Diagram Exit Room



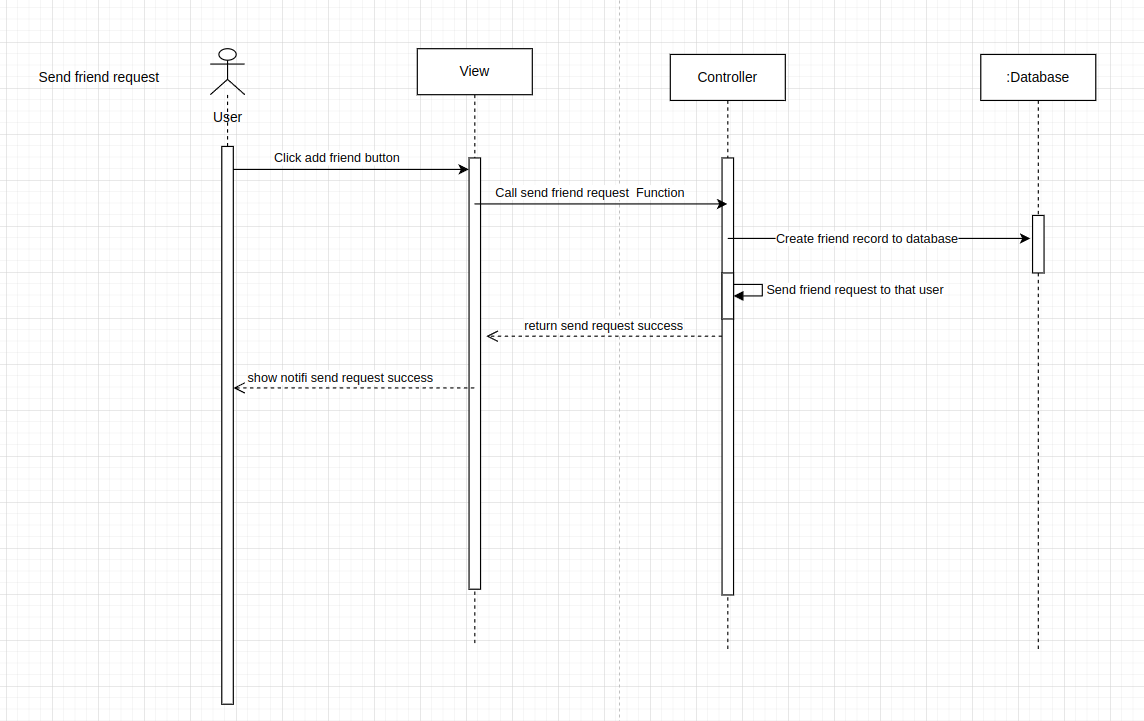
**Hình 3.4.6 Sequence Diagram Exit Room**

### Sequence Diagram Join Meeting Room



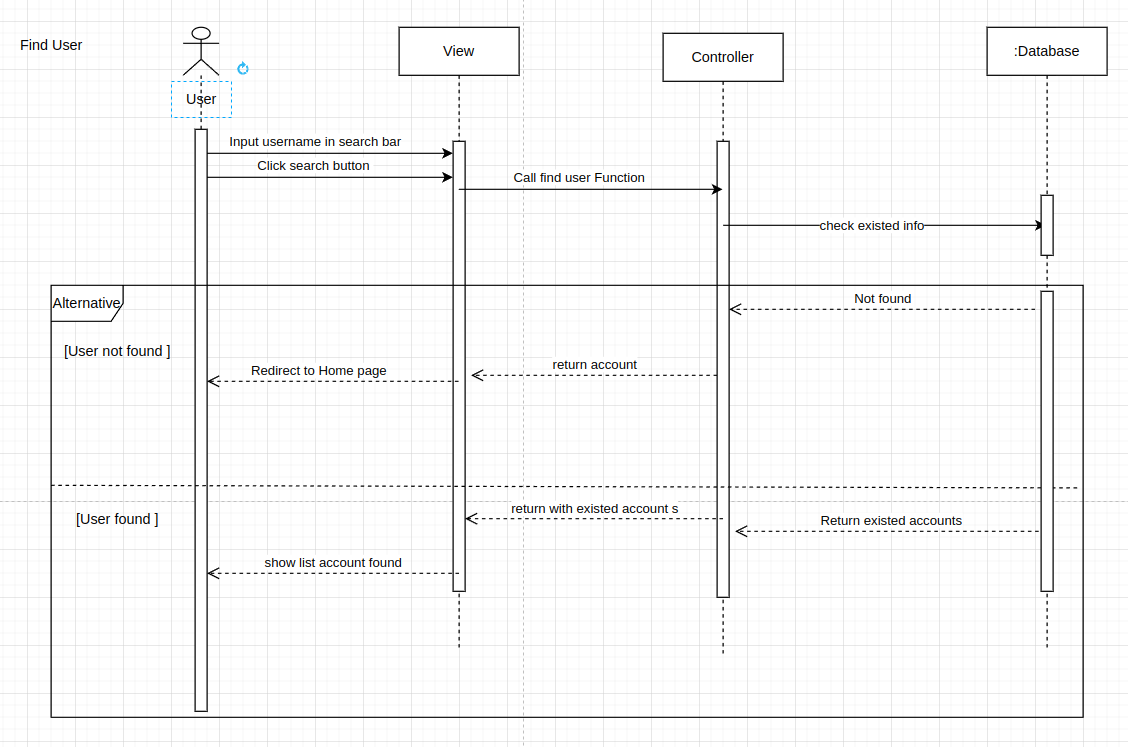
**Hình 3.4.7 Sequence Diagram Join Meeting Room**

### Sequence Diagram Send Friend Request



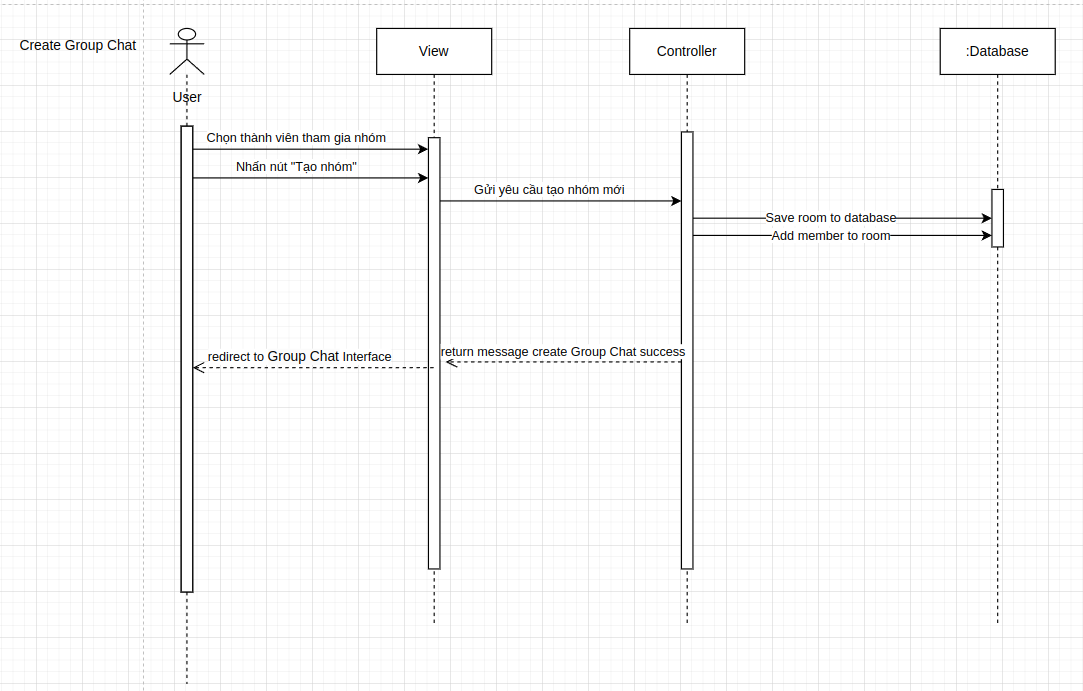
**Hình 3.4.8 Sequence Diagram Send Friend Request**

### Sequence Diagram Find User



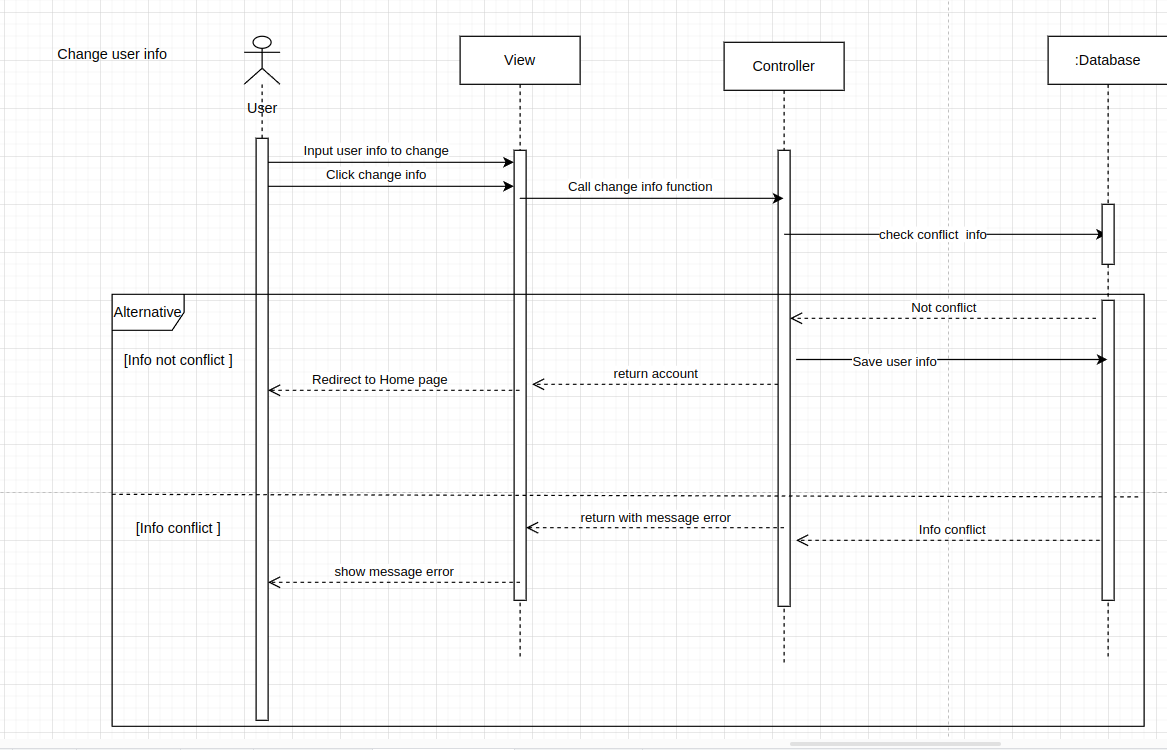
**Hình 3.4.9 Sequence Diagram Find User**

### Sequence Diagram Create Group Chat



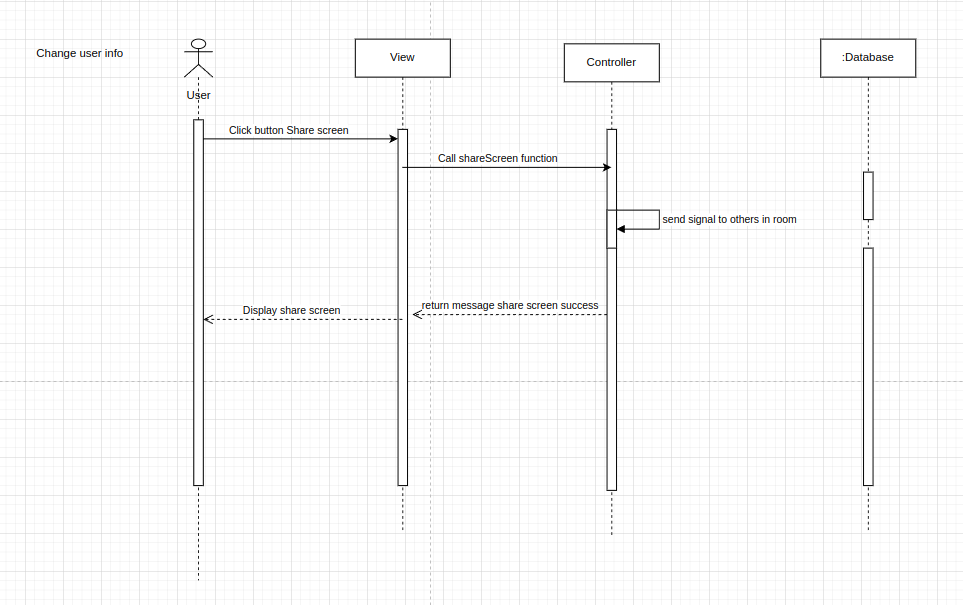
**Hình 3.4.10 Sequence Diagram Create Group Chat**

### Sequence Diagram Change User Info



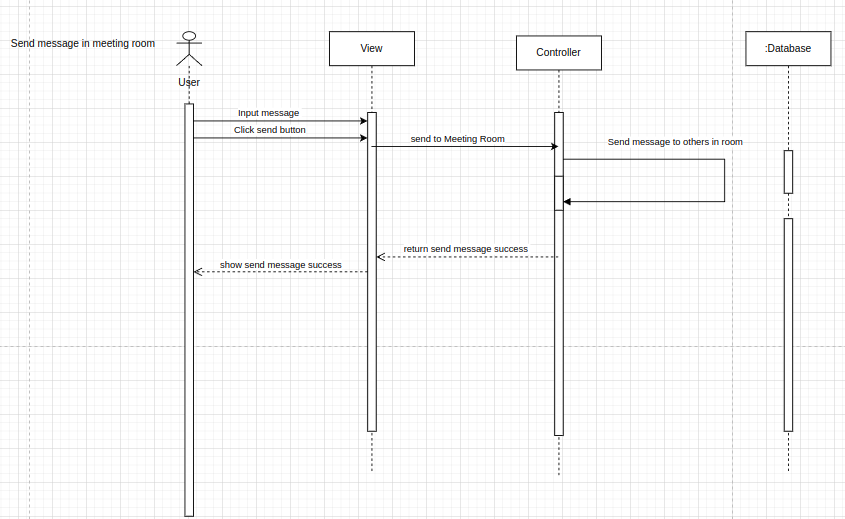
**Hình 3.4.10 Sequence Diagram Change User Info**

### Sequence Diagram Share Screen



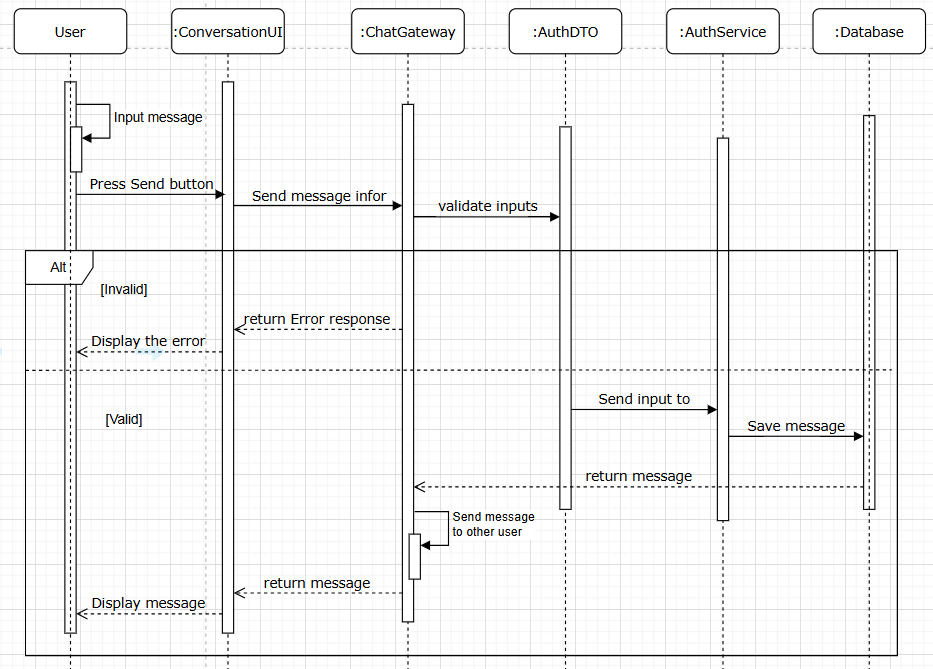
**Hình 3.4.12 Sequence Diagram Share Screen**

### Sequence Diagram Meeting Chat



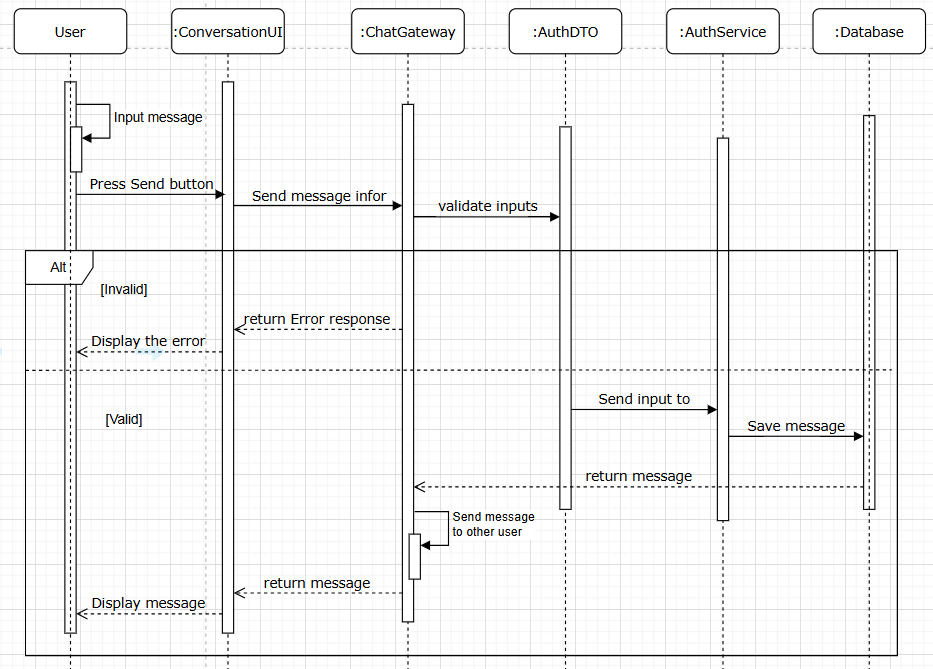
**Hình 3.4.10 Biểu đồ tuần tự trò chuyện trong lúc họp**

### Sequence Diagram Delete User



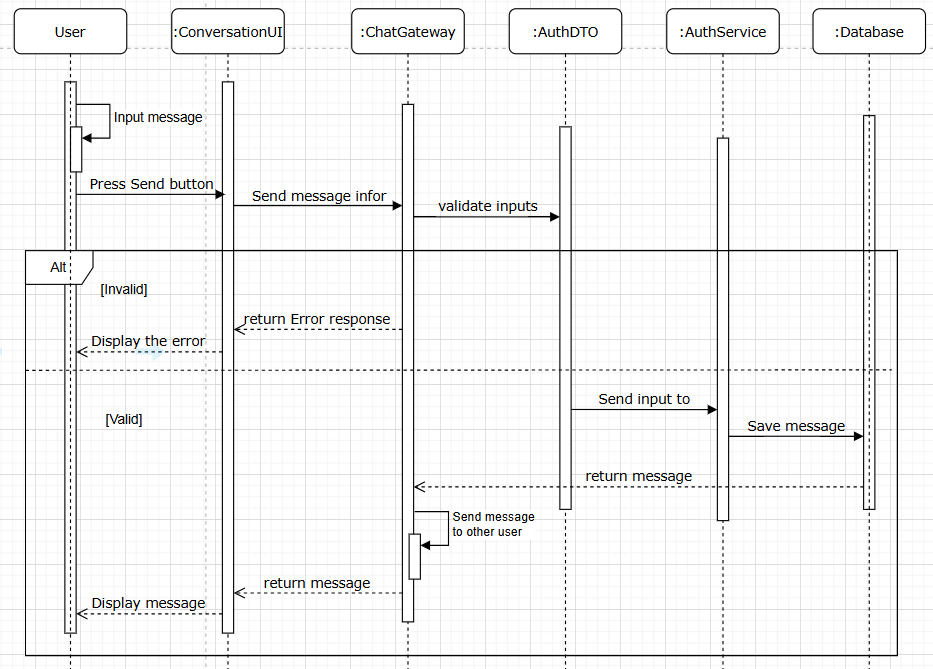
**Hình 3.4.10 Sequence Diagram Delete User**

### Sequence Diagram Logout



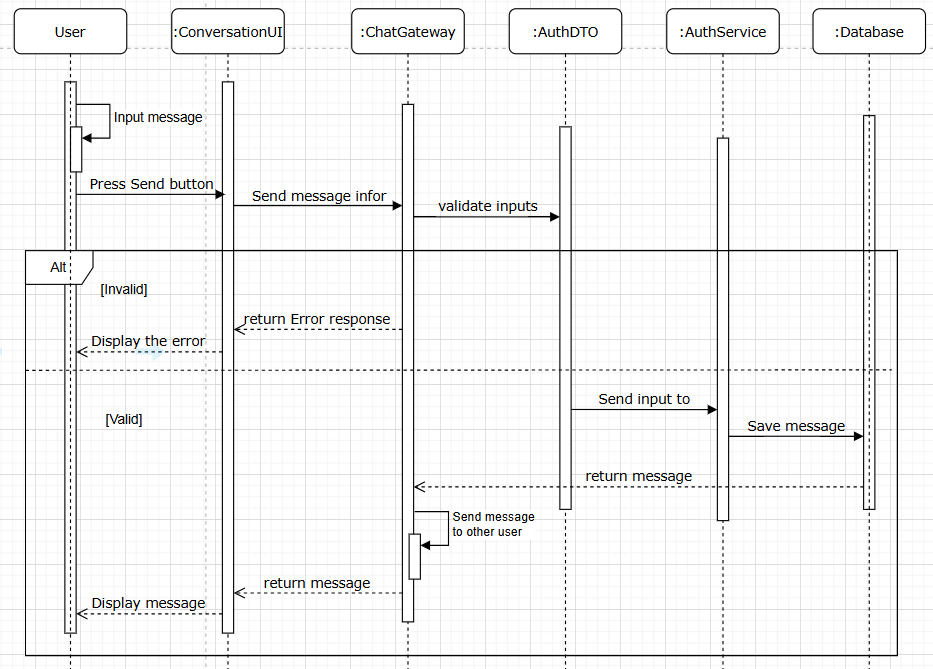
**Hình 3.4.10 Sequence Diagram Delete User**

### Sequence Diagram Delete User



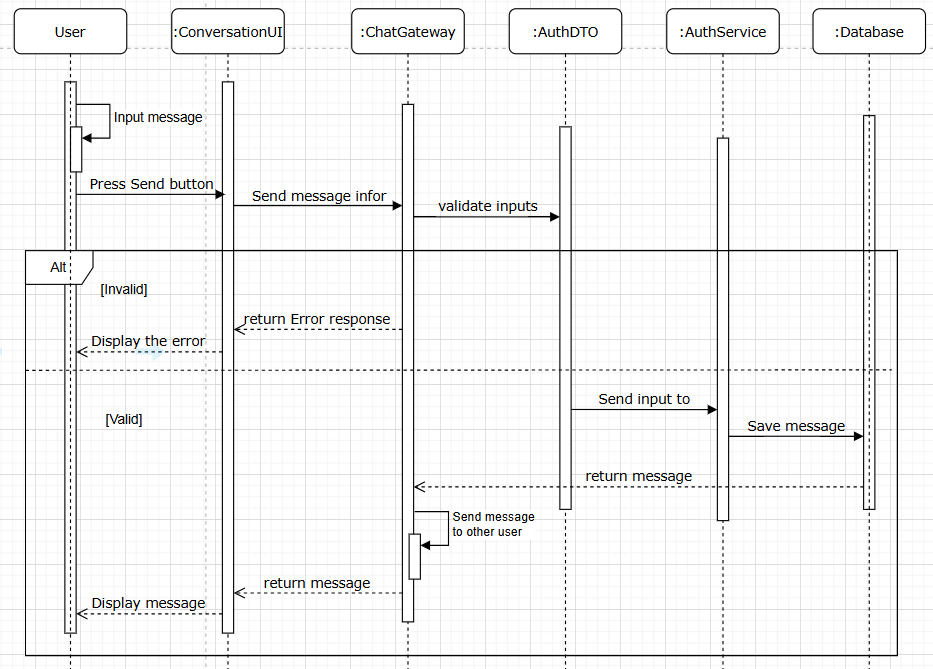
**Hình 3.4.10 Sequence Diagram Delete User**

### Sequence Diagram Delete User



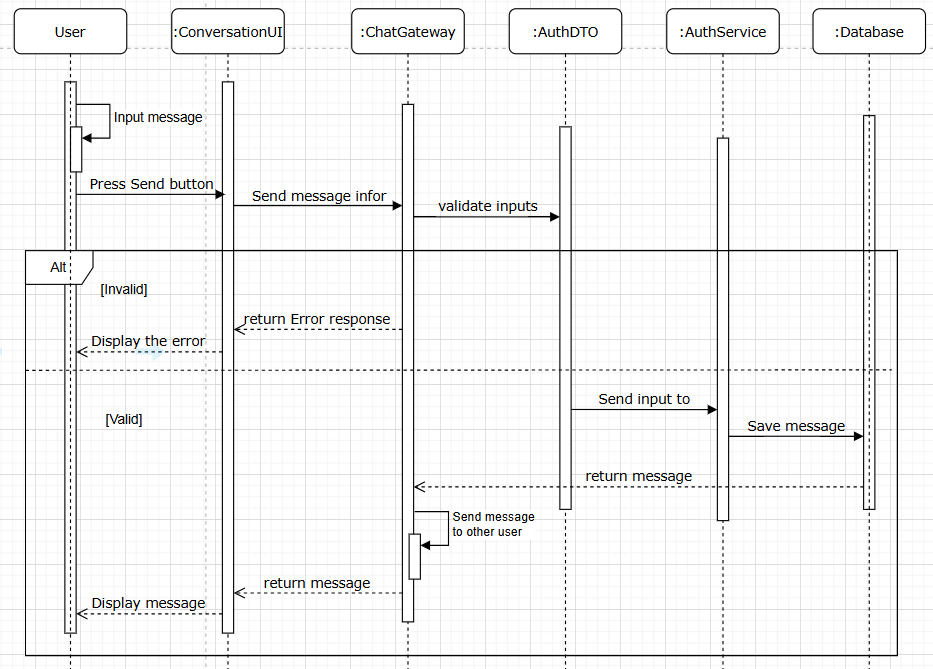
**Hình 3.4.10 Sequence Diagram Delete User**

### Sequence Diagram Delete User



**Hình 3.4.10 Sequence Diagram Delete User**

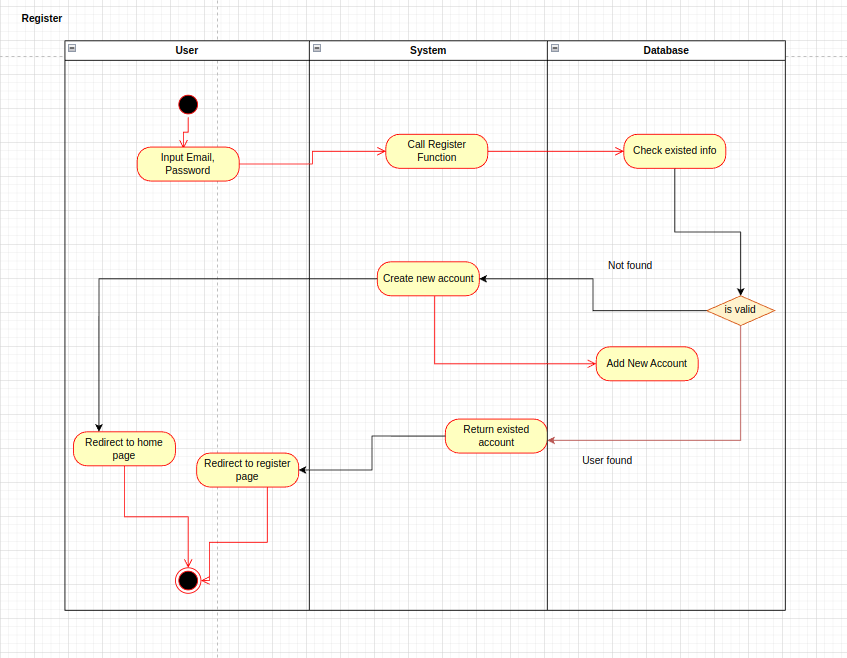
### Sequence Diagram Delete User



**Hình 3.4.10 Sequence Diagram Delete User**

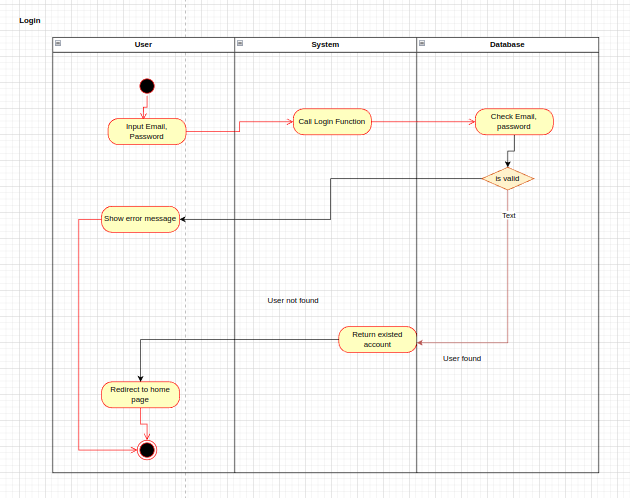
## Activity diagrams

* + 1. **Activity diagram Register**



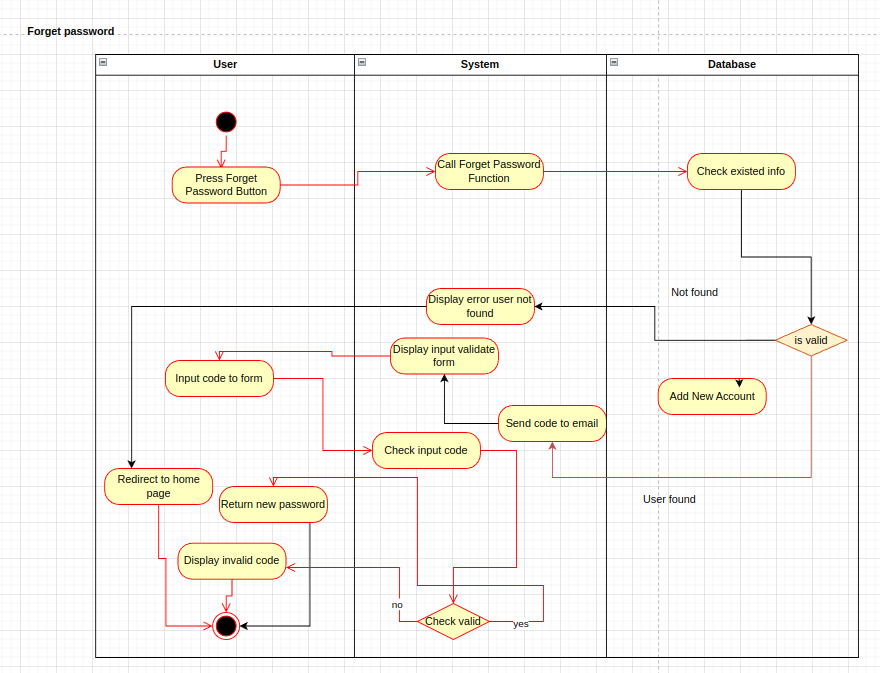
**Hình 3.5.1 Biểu đồ hoạt động Đăng kí**

### Activity Diagram Login



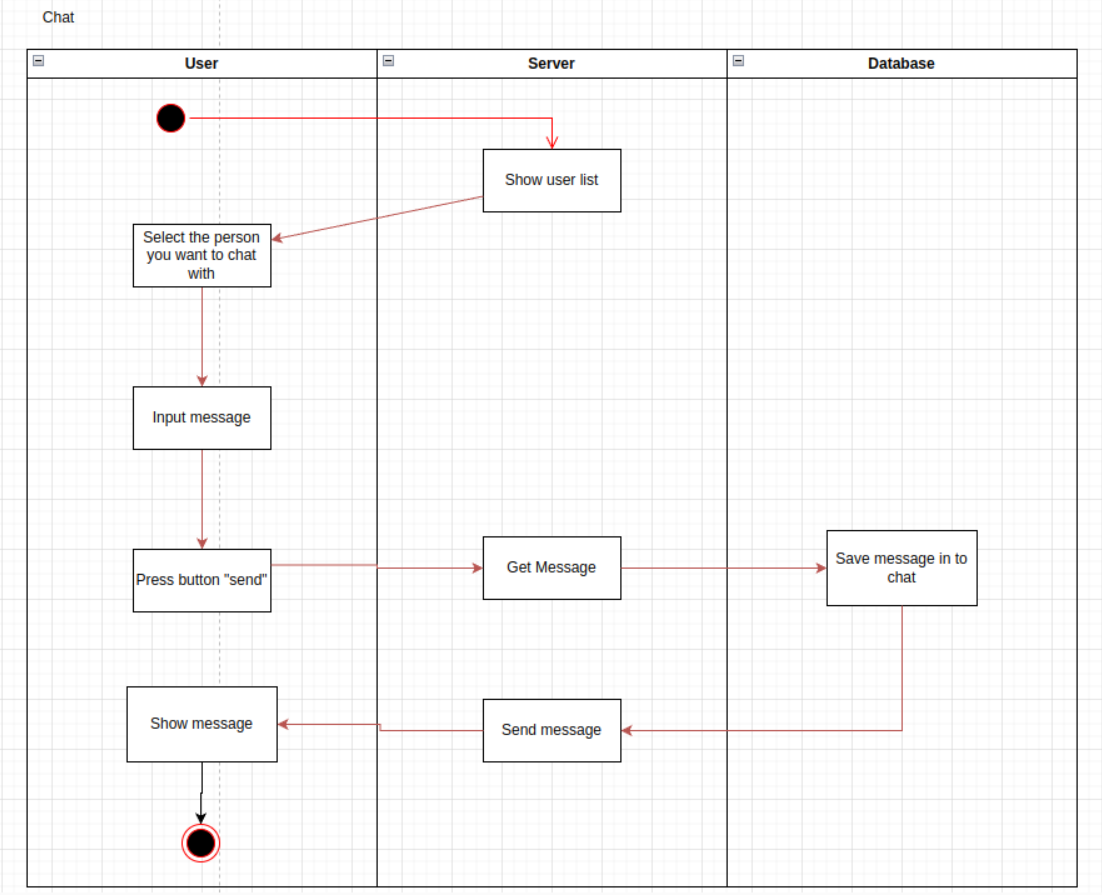
**Hình 3.5.2 Biểu đồ hoạt động Đăng nhập**

### Activity Diagram Forget Password



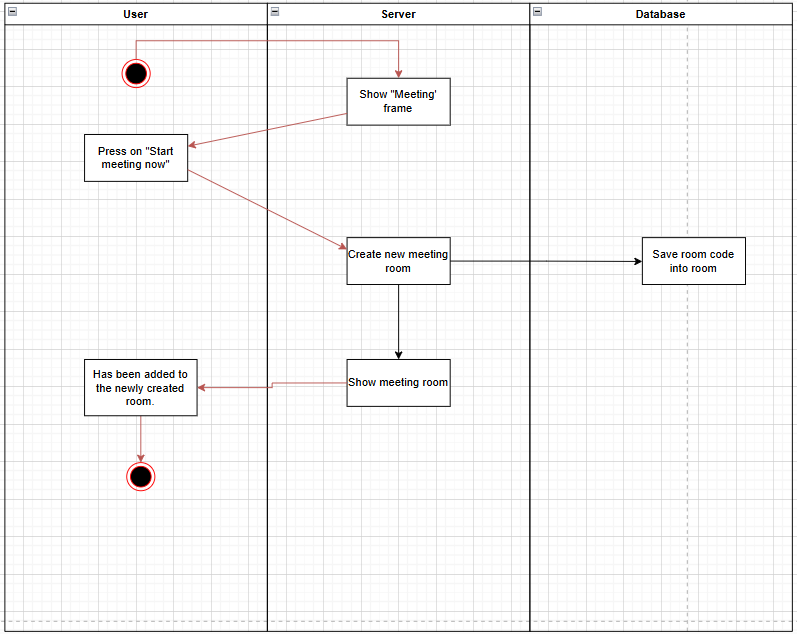
**Hình 3.5.3 Biểu đồ hoạt động Forget Password**

### Activity Diagram Chat



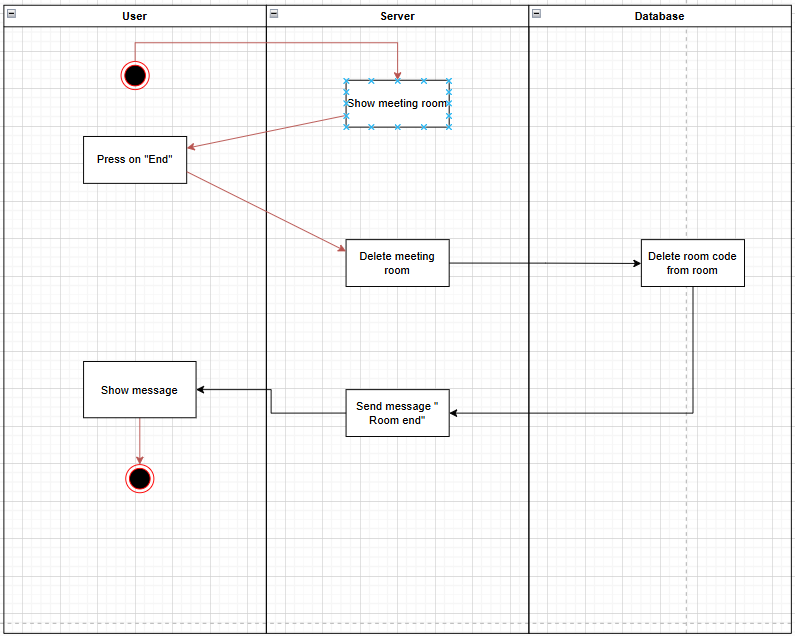
**Hình 3.5.4 Activity Diagram Chat**

### Activity Diagram Forget Password



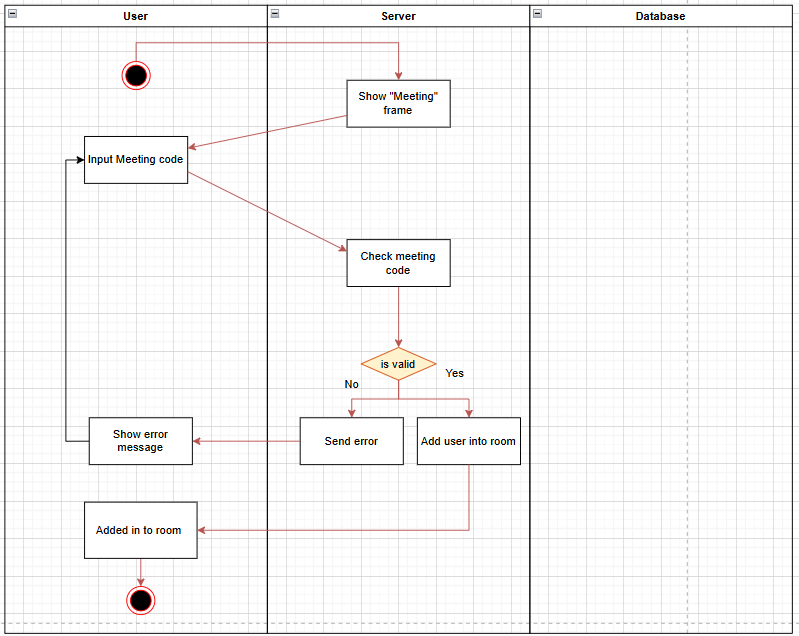
**Hình 3.5.5 Activity Diagram Forget Password**

### Activity Diagram Exit Room



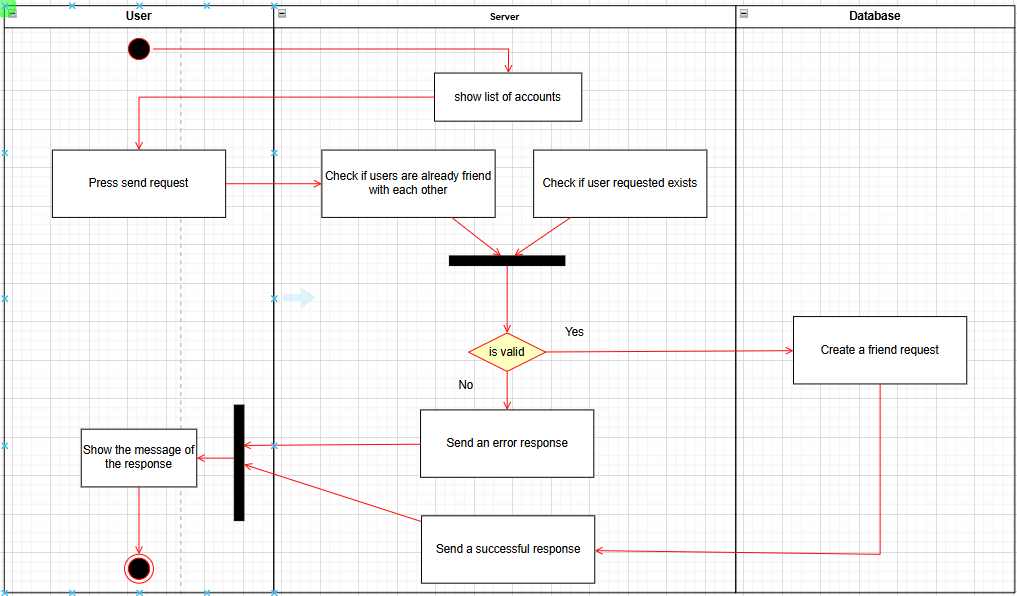
**Hình 3.5.6 Activity Diagram Exit Room**

### Activity Diagram Join Meeting Room



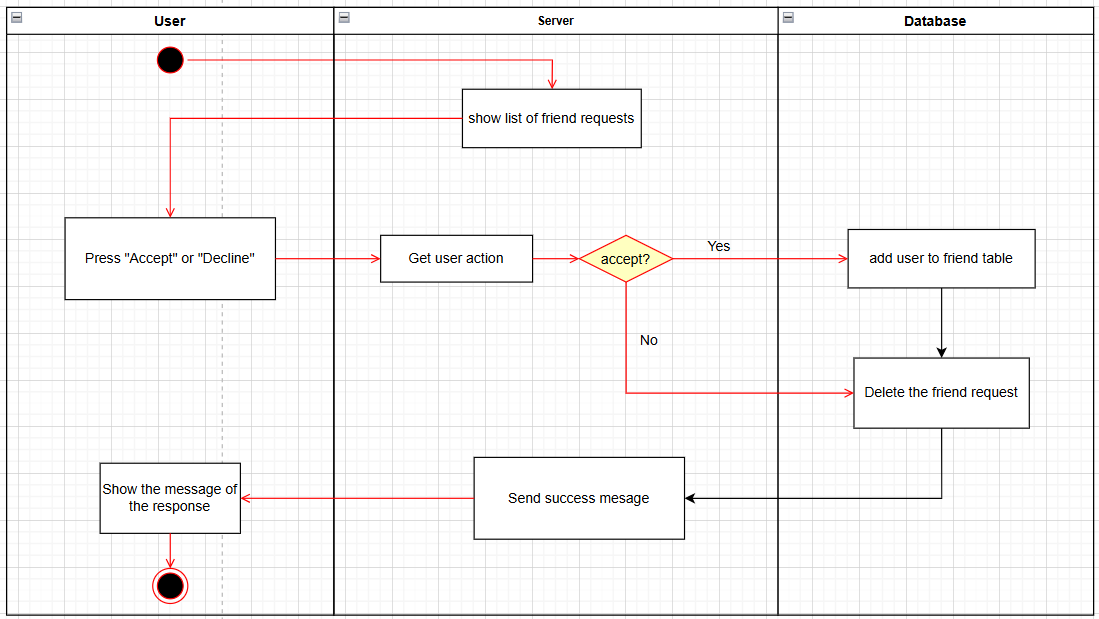
**Hình 3.5.7 Activity Diagram Join Meeting Room**

### Activity Diagram Send Friend Request



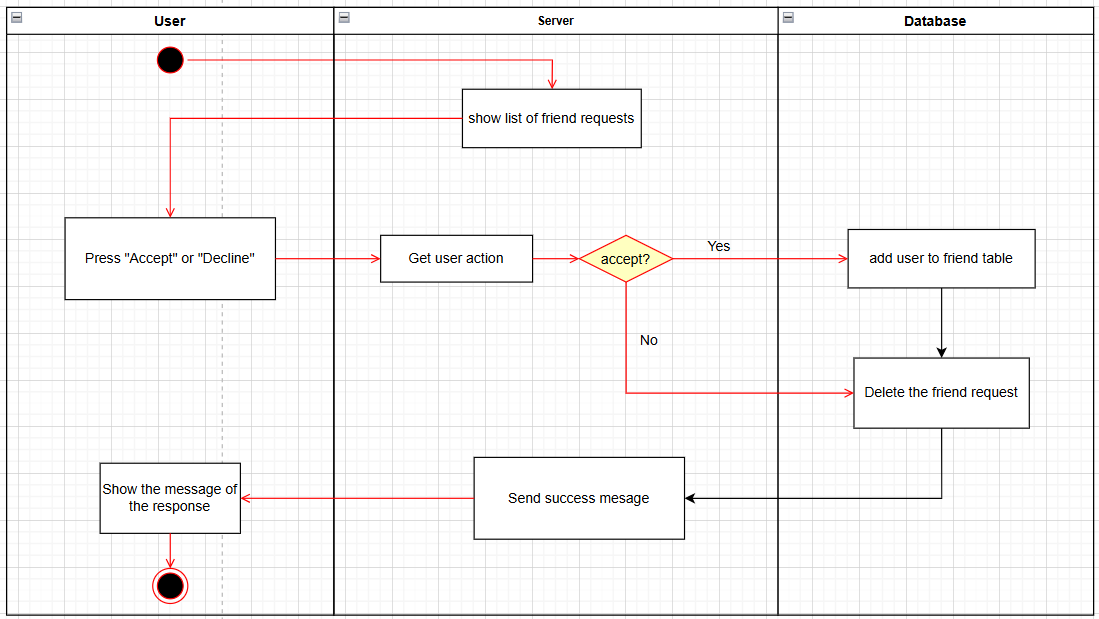
**Hình 3.5.9 Activity Diagram Send Friend Request**

### Activity Diagram Process Friend Requests



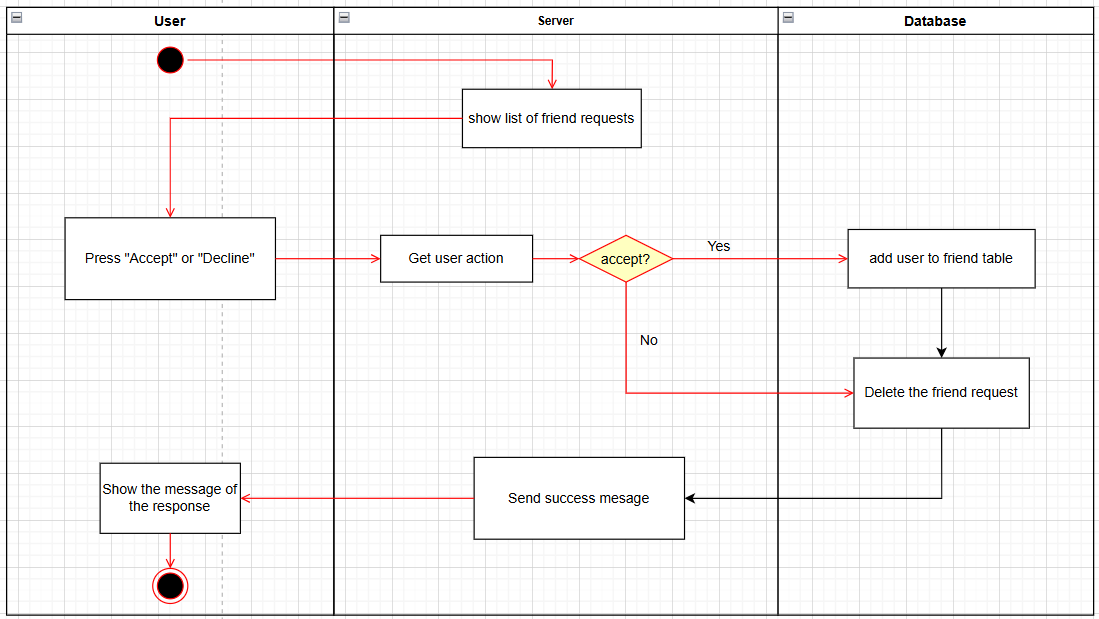
**Hình 3.5.10 Biểu đồ hoạt động Xử lý yêu cầu kết bạn**

### Activity Diagram Process Join Meeting



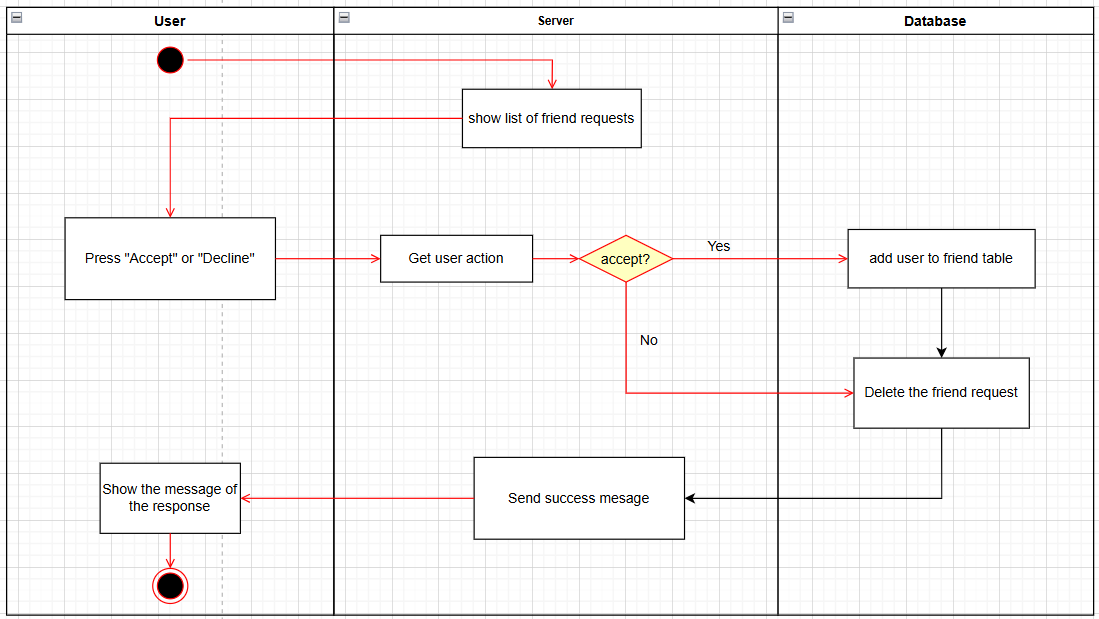
**Hình 3.5.10 Activity Diagram Process Friend Requests**

### Activity Diagram Meeting Chat



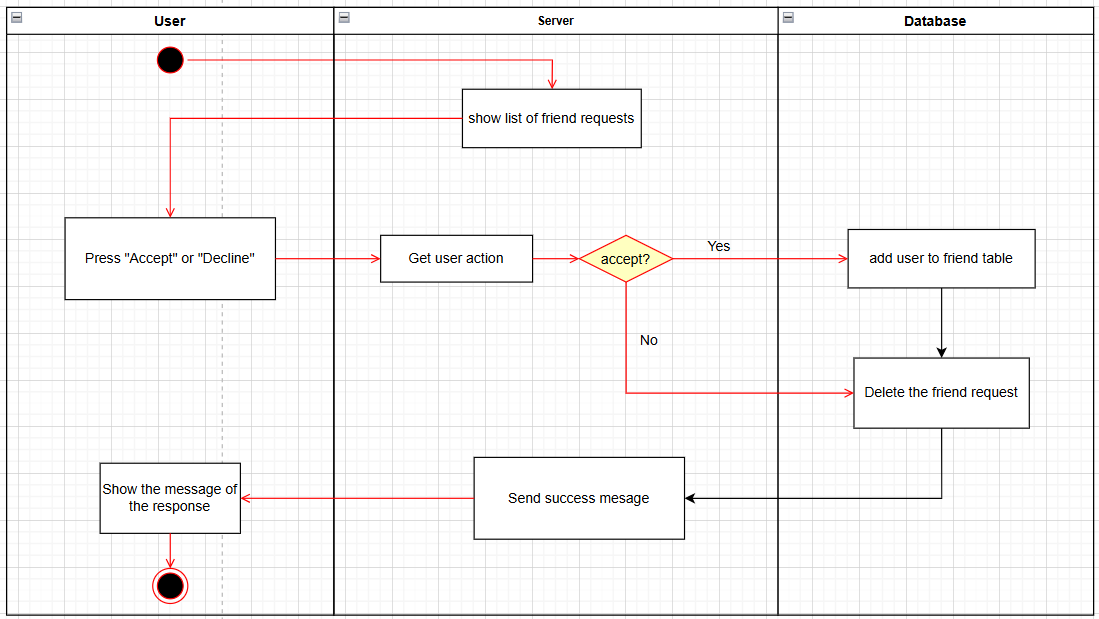
**Hình 3.5.10 Biểu đồ hoạt động Xử lý yêu cầu kết bạn**

### Activity Diagram Meeting Chat



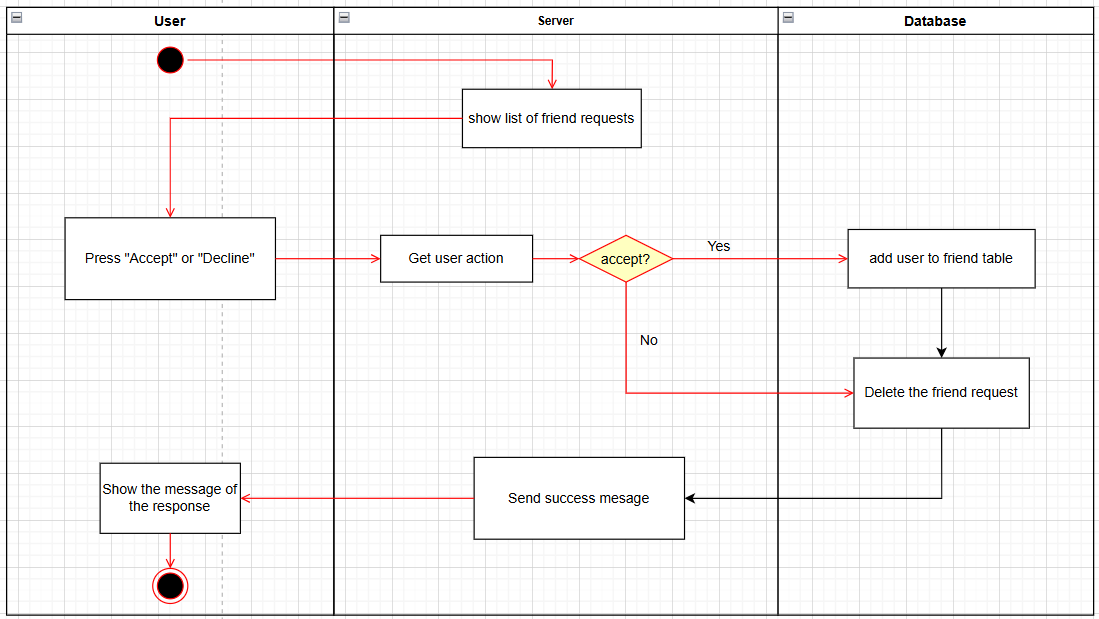
**Hình 3.5.10 Biểu đồ hoạt động Xử lý yêu cầu kết bạn**

### Activity Diagram Share Screen



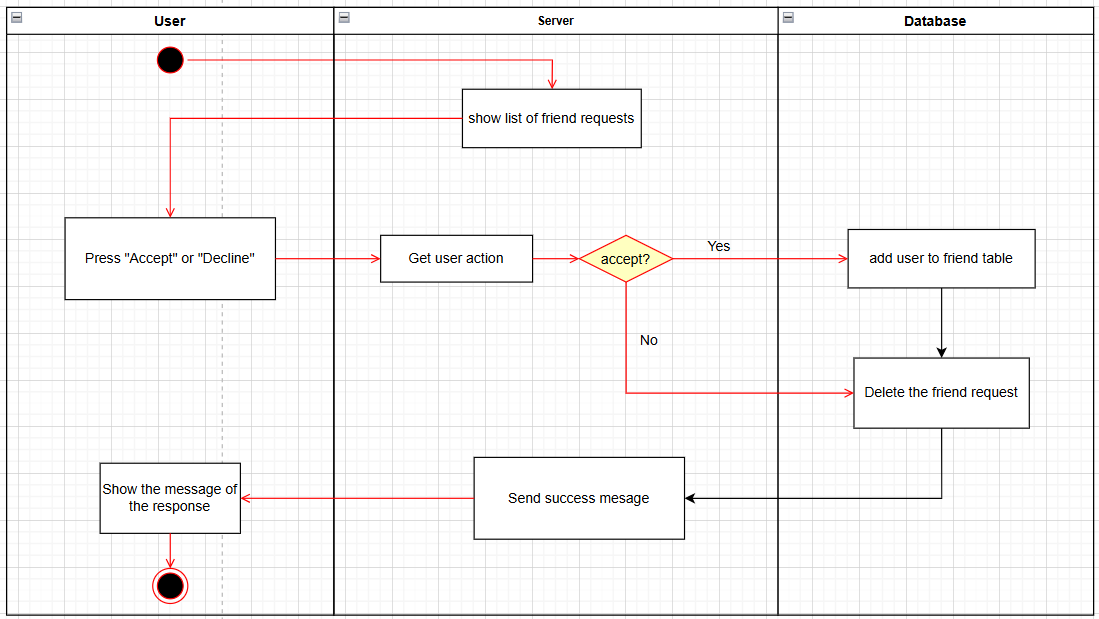
**Hình 3.5.10 Activity Diagram Share Screen**

### Activity Diagram Create Meeting Room



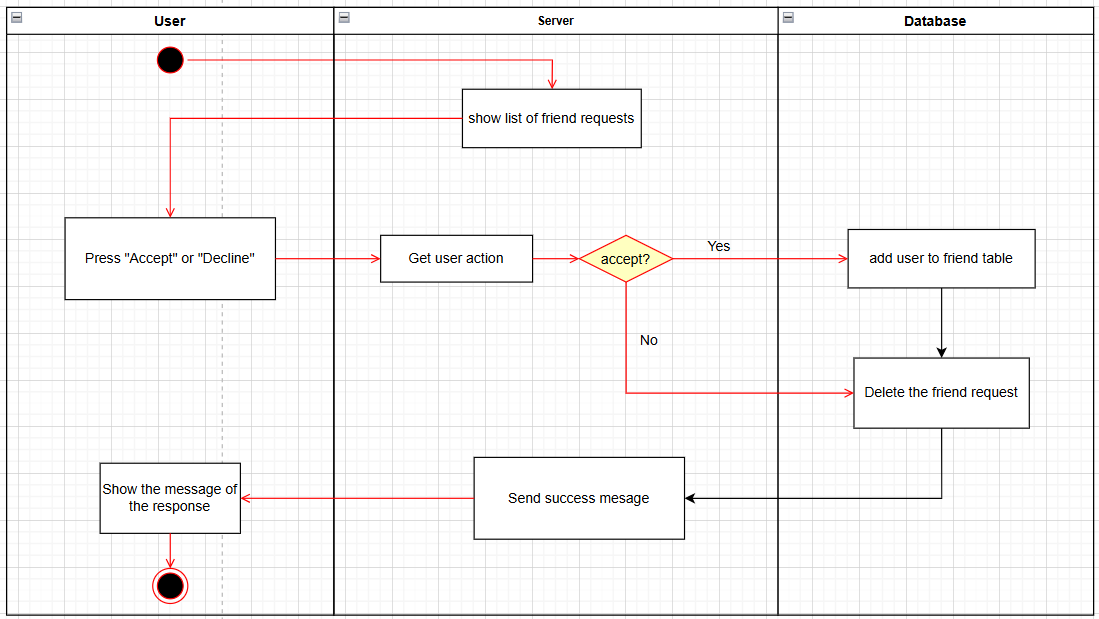
**Hình 3.5.14 Activity Diagram Create Meeting Room**

### Activity Diagram Find user



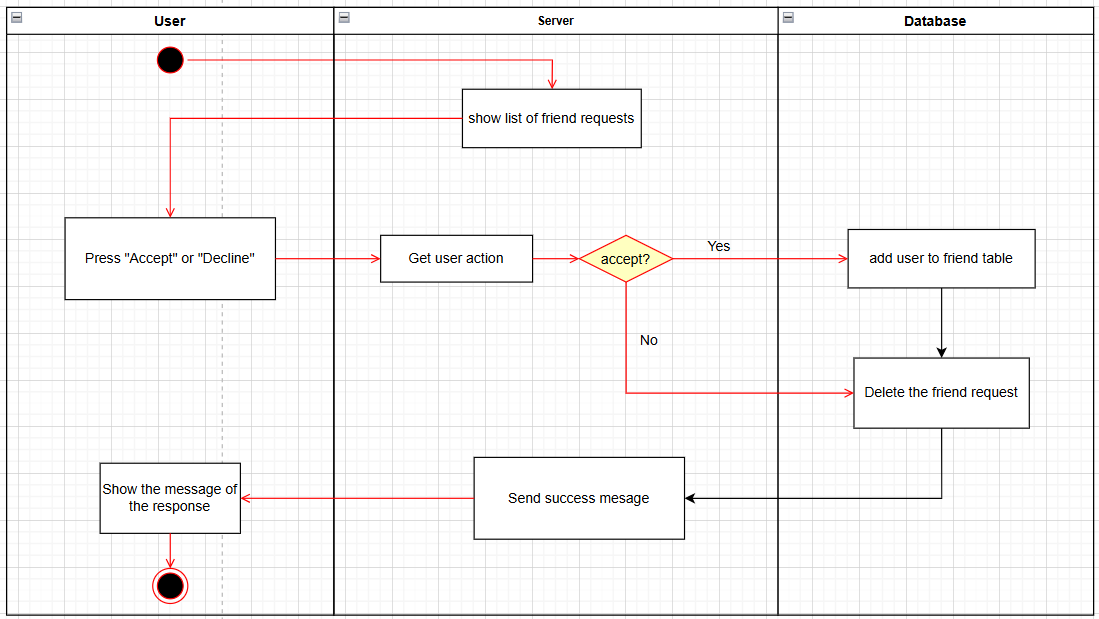
**Hình 3.5.14 Activity Diagram Find user**

### Activity Diagram Create Group Chat



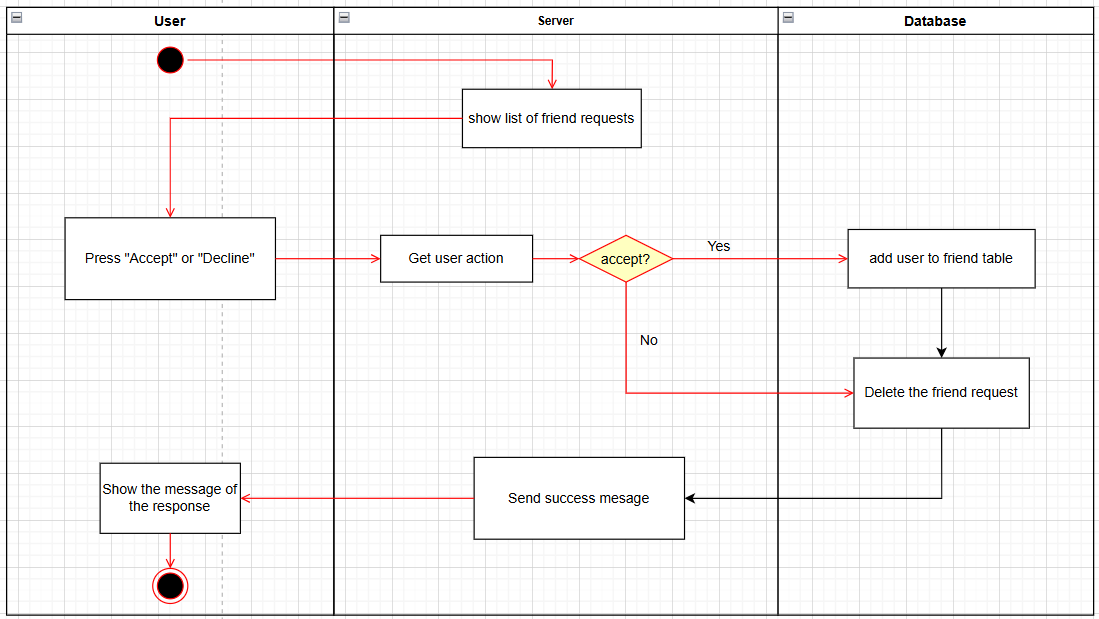
**Hình 3.5.16 Activity Diagram Create Group Chat**

### Activity Diagram Change User Info



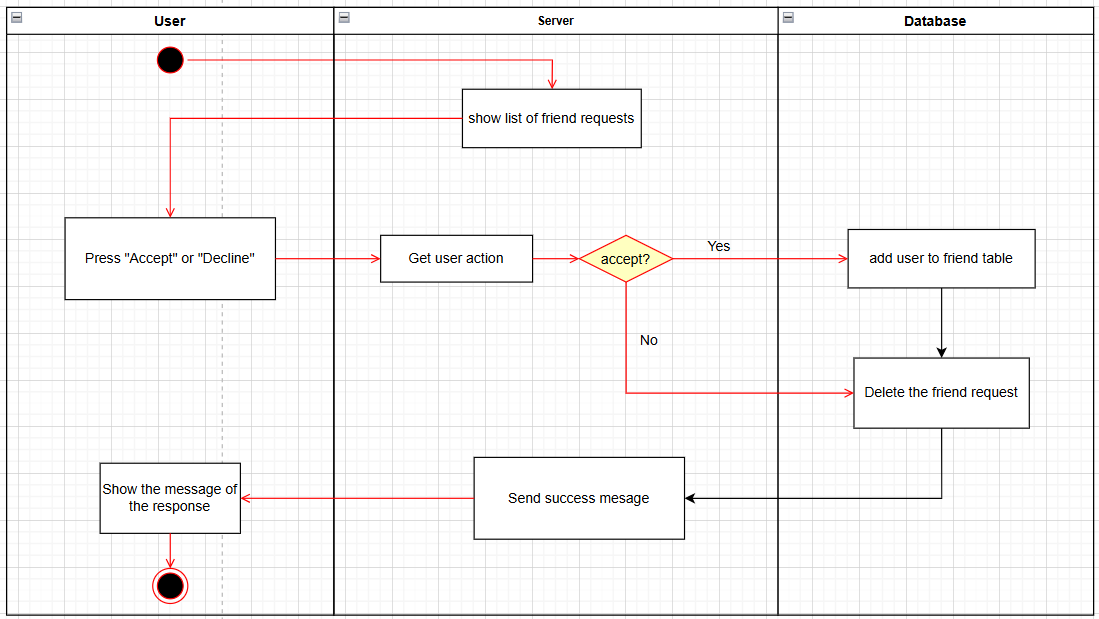
**Hình 3.5.17 Activity Diagram Change User Info**

### Activity Diagram Change User Info



**Hình 3.5.18 Activity Diagram Change User Info**

### Activity Diagram Delete User



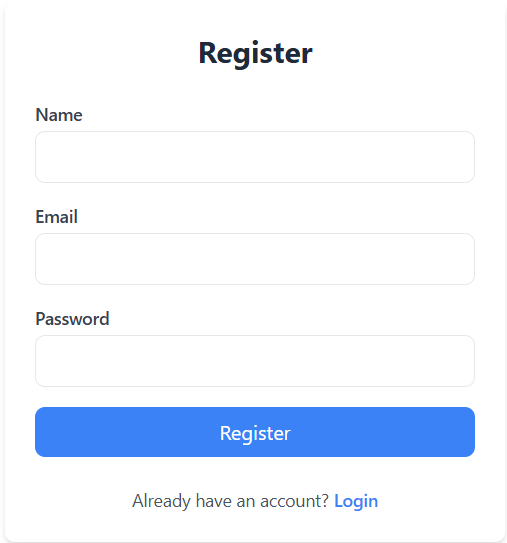
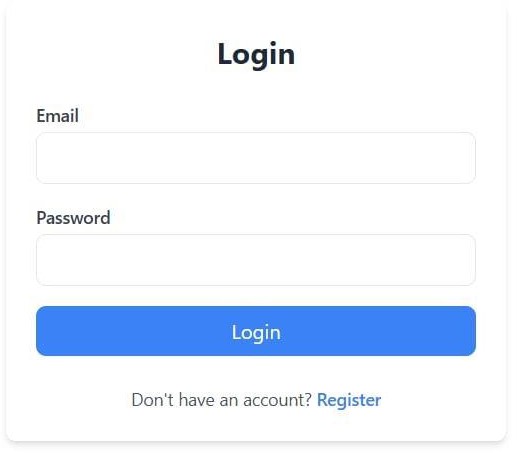
**Hình 3.5.19 Activity Diagram Delete User**

* 1. **Biểu đồ lớp**

# CHƯƠNG 3: CHƯƠNG TRÌNH DEMO

## Giao diện trang Đăng nhập & Đăng kí

Người dùng thấy giao diện đăng nhập hoặc đăng ký khi mới mở ứng dụng. Để tiến hành đăng nhập, người dùng nhập thông tin tài khoản và mật khẩu. Sau khi đăng ký thành công, hệ thống sẽ chuyển sang trang đăng nhập.



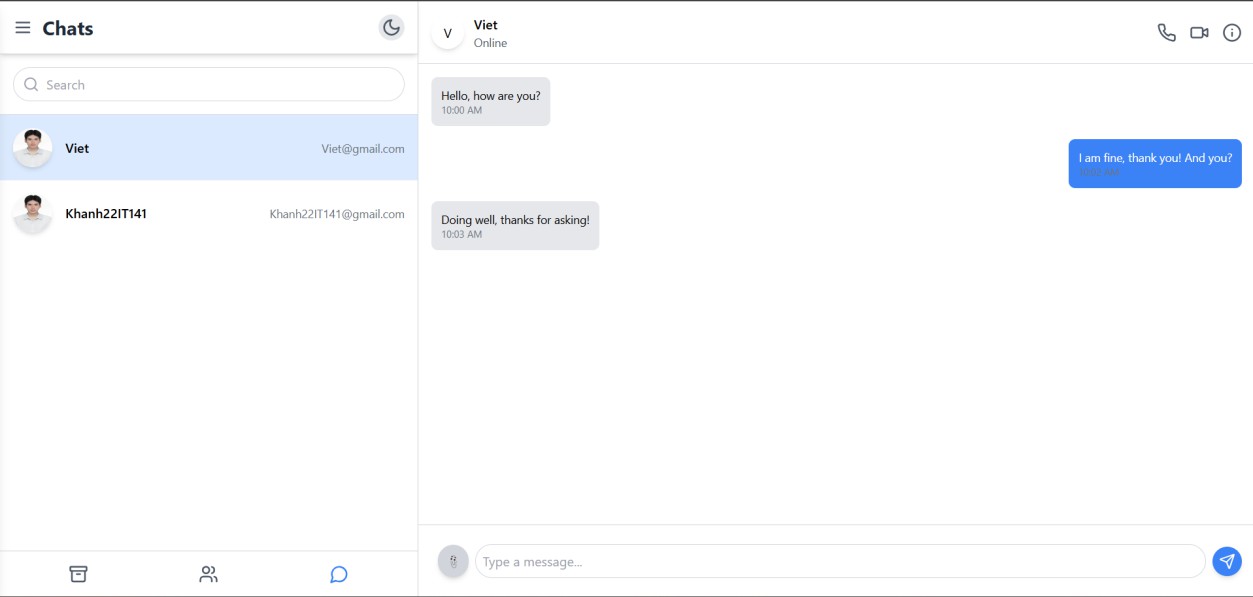
**Hình 4.1.1 Đăng kí & Đăng nhập**

## Giao diện màn hình trò chuyện

Giao diện trò chuyện của ứng dụng Boom-Meeting này bao gồm: Thanh bên trái gồm thanh điều hướng, thanh tìm kiếm và hiển thị những người dùng đã kết bạn. Dưới cùng thanh bên trái có các thành phần như biểu tượng menu cho phép mở và đóng thanh điều hướng phụ, và nút chuyển đổi giữa chế độ sáng và tối. Bên dưới là thanh tìm kiếm, cho phép người dùng nhập từ khóa để tìm kiếm nhanh các cuộc trò chuyện và danh bạ bạn bè.

**Phần chính giữa** là khu vực hiển thị nội dung cuộc trò chuyện. Tại đây, người dùng có thể xem các tin nhắn đã gửi và nhận. Mỗi tin nhắn đều có ghi chú thời gian và được căn lề trái hoặc phải tùy thuộc vào người gửi. Khu vực này cũng bao gồm thanh tiêu đề ở trên cùng, hiển thị tên của người dùng đang trò chuyện, trạng thái trực tuyến của họ, cùng với các biểu tượng để thực hiện cuộc gọi thoại, gọi video hoặc xem thông tin chi tiết về người đó.

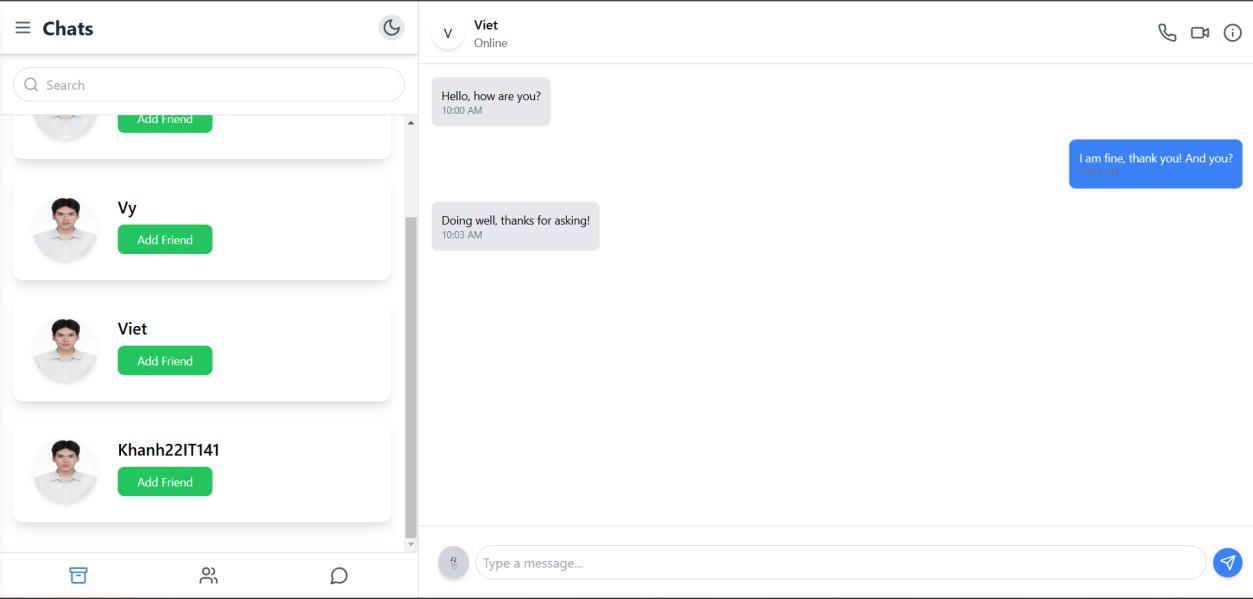
**Thanh nhập liệu** nằm ở dưới cùng của giao diện trò chuyện, nơi người dùng có thể gõ tin nhắn và gửi đi bằng cách nhấn biểu tượng gửi màu xanh.



**Hình 4.2.1 Màn hình trang trò chuyện**

## Giao diện màn hình gửi lời mời kết bạn:

Màn hình này bao gồm một danh sách người dùng với thông tin cơ bản của họ như tên và ảnh đại diện. Mỗi mục trong danh sách đi kèm với nút "Add Friend" để gửi yêu cầu kết bạn. Người dùng có thể sử dụng thanh tìm kiếm ở trên cùng để tìm kiếm bạn bè theo tên.



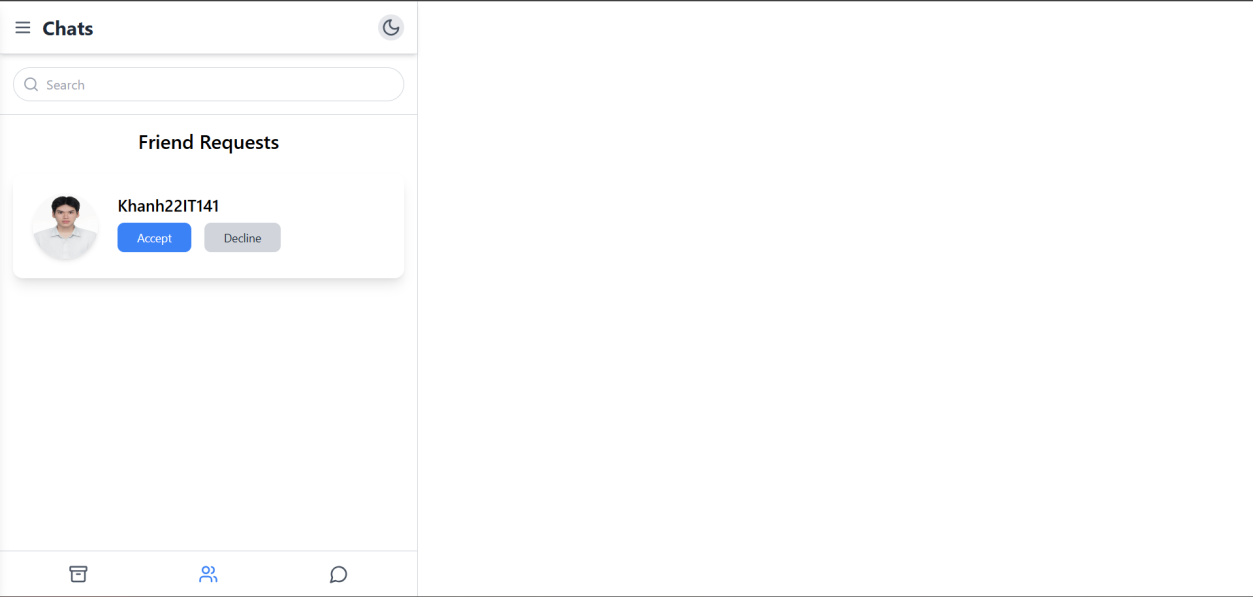
**Hình 4.3.1 Màn hình gửi lời mời kết bạn**

## Giao diện màn hình xử lý lời mời kết bạn:

Màn hình này hiển thị danh sách các yêu cầu kết bạn mà người dùng đã nhận được. Mỗi yêu cầu được trình bày cùng với tên và ảnh đại diện của người gửi.

Người dùng có thể chọn "Accept" (Chấp nhận) để đồng ý kết bạn hoặc "Decline" (Từ chối) nếu không muốn kết bạn. Thanh tìm kiếm ở trên cùng cho phép người dùng tìm kiếm nhanh các yêu cầu. Phía bên trái vẫn là thanh điều hướng chứa

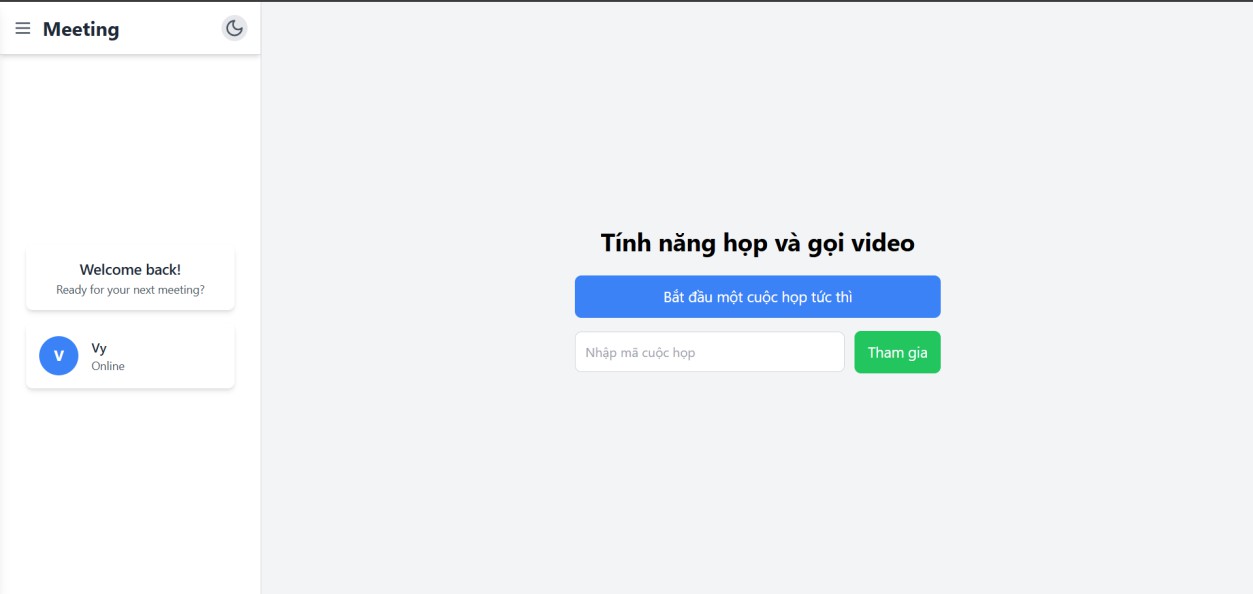
các biểu tượng như menu điều khiển và nút chuyển đổi chế độ sáng-tối, giúp người dùng dễ dàng thao tác và quản lý các yêu cầu kết bạn của mình.



**Hình 4.4.1 Màn hình xử lý lời mời kết bạn**

## Giao diện tạo cuộc họp

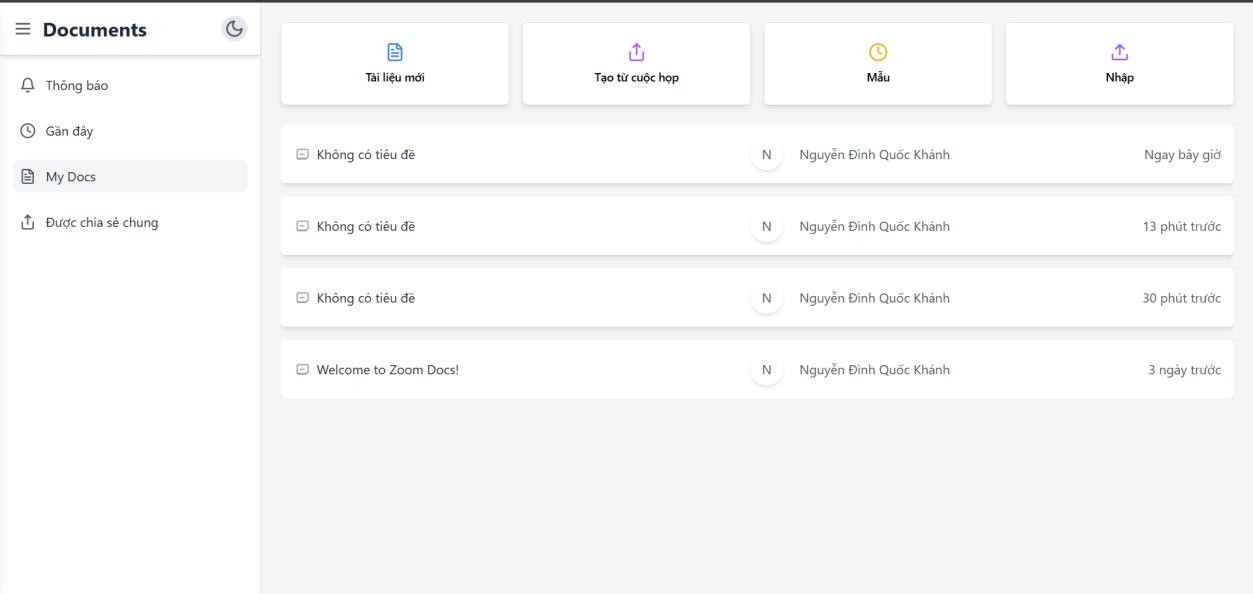
Màn hình này cung cấp cho người dùng hai tùy chọn chính: tạo mới một cuộc họp hoặc tham gia vào cuộc họp có sẵn. Ở phía bên phải, người dùng có thể nhấn nút "Bắt đầu một cuộc họp tức thì" để khởi tạo một cuộc họp mới ngay lập tức. Bên dưới, có trường nhập mã cuộc họp và nút "Tham gia" màu xanh lá, cho phép người dùng nhập mã để tham gia vào một cuộc họp đã được tạo trước đó.



**Hình 4.5.1 Màn hình giao diện tạo cuộc họp**

## Giao diện quản lý tài liệu

Giao diện quản lý tài liệu cung cấp một cách dễ dàng để người dùng tổ chức và truy cập các tài liệu của mình. Thanh điều hướng bên trái giúp chuyển nhanh giữa "Thông báo", "Gần đây", "My Docs" và các tài liệu chia sẻ. Phần chính hiển thị các chức năng như tải lên tài liệu mới, tạo từ cuộc họp, chọn mẫu, và nhập liệu. Danh sách tài liệu bao gồm tên, tác giả, và thời gian chỉnh sửa, giúp người dùng quản lý hiệu quả các tài liệu của mình.



**Hình 4.6.1 Màn hình quản lý tài liệu**

Kết luận

## Kết quả đạt được

Ứng dụng với giao diện thân thiện với người dùng, dễ sử dụng, các thao tác được thực hiện khá dễ dàng, đầy đủ những tính năng mà một trang web trao đổi sách nên có.

## Những kĩ năng học được

Cách để thiết kế và tạo dựng một ứng dụng di động từ giao diện tới chức năng. Những kiến thức về những ngôn ngữ lập trình có liên quan trong việc tạo dựng một ứng dụng di động.

Kinh nghiệm về trải nghiệm người dùng từ đó có thể nắm bắt tâm lí khách hàng khi xây dựng ứng dụng.

Cách quản lí một cơ sở dữ liệu, sự liên kết giữa Frontend và Backend Cách viết đồ án, trình bày Word cũng như kĩ năng làm Slide.

## Hướng phát triển

Cải thiện giao diện.

Chế độ sáng tối.

Thêm chức năng tạo sự kiện cho nhóm Thêm chức năng gợi ý bài đăng

Hỗ trợ đa ngôn ngữ.

# Tài liệu tham khảo:

* + 1. BÀI GIẢI LẬP TRÌNH MẠNG - PGS.TS.Huỳnh Công Pháp; Nguyễn Anh Tuấn; Lê Tân; Nguyễn Thanh Cẩm;Hoàng Hữu Đức – vku.udn.vn
    2. Advanced Network Programming - Principles and Techniques 2013 - Bogdan Ciubotaru; Gabriel-Miro Muntean.
    3. 2021-Giao\_trinh\_Lap\_trinh\_mang\_TCPSocket.
    4. Mô hình OSI là gì? Chức năng của mỗi tầng trong mô hình OSI - https://vietnix.vn/mo-hinh-osi-la-gi/
    5. Những giao thức mạng được sử dụng phổ biến - https://kb.pavietnam.vn/nhung- giao-thuc-mang-duoc-su-dung-pho-bien.html
    6. Mô hình mạng máy tính? Tổng hợp các mô hình mạng phổ biến nhất hiện nay

- https://viettuans.vn/mo-hinh-mang-may-tinh

* + 1. Mạng máy tính là gì? Lợi ích và phân loại mạng máy tính - https://viettuans.vn/mang-may-tinh-la-gi