TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

BÁO CÁO ĐỀ TÀI MÔN PROJECT II

ỨNG DỤNG NHẮN TIN TRÊN NỀN TẢNG WEB

Họ và tên: Phan Trọng Tình MSSV: 20183997

Lớp: CNTT 01 – K63

Email: [tinh.pt183997@sis.hust.edu.vn](mailto:tinh.pt183997@sis.hust.edu.vn)

Tổng quan

Báo cáo trình bày về kết quả thực hiện đề tài bộ môn Project II, xây dựng ứng dụng nhắn tin trên nền tảng Web. Áp dụng những công nghệ như Node.js để xây dựng Server với một luồng xử lý yêu cầu, giúp đảm bảo vấn đề cân bằng I/O; Vue.js – Một front-end framework nhỏ gọn và đầy đủ tính năng, hỗ trợ Single-File Component và Vuex dựa trên mô hình Flux, giúp dễ dàng quản lý trạng thái ứng dụng; MongoDB – Cơ sở dữ liệu document, lưu trữ thông tin của một đối tượng gói gọn trong một document; WebRTC, API hỗ trợ bởi đa số trình duyệt, cho phép truy cập thiết bị đa phương tiện trên thiết bị, tạo kết nối Peer-to-Peer giữa các trình duyệt, truyền đa phương tiện thời gian thực. Ứng dụng nhắn tin cung cấp những chức năng cơ bản như: Đăng ký, đăng nhập, tìm kiếm người dùng, gửi, hủy, chấp nhận và từ chối yêu cầu kết bạn, tạo nhóm, rời nhóm, thêm bạn bè vào nhóm và gọi video.

Lời cảm ơn

Xin được gửi lời cảm ơn tới Thầy Nguyễn Nhật Quang, Giảng viên Trường Đại Học Bách Khoa Hà Nội, đã hết sức ủng hộ, kiên nhẫn và tạo điều kiện cho em có thể chủ động lựa chọn đề tài và tìm hiểu công nghệ để có thể hoàn thành đồ án môn học một cách tốt nhất.

Mục lục

[I. Giới thiệu: 5](#_Toc75116810)

[II. Nội dung: 5](#_Toc75116811)

[1. Phân tích yêu cầu hệ thống: 5](#_Toc75116812)

[*a. Các tác nhân hệ thống:* 5](#_Toc75116813)

[*b. Mô hình hệ thống:* 5](#_Toc75116814)

[*c. Một số nghiệp vụ cơ bản:* 5](#_Toc75116815)

[*d. Các biểu đồ ca sử dụng của hệ thống:* 6](#_Toc75116816)

[2. Các công nghệ sử dụng: 18](#_Toc75116817)

[*a. Node.js:* 18](#_Toc75116818)

[*b. Vue.js:* 19](#_Toc75116819)

[*c. MongoDB:* 20](#_Toc75116820)

[*d. WebRTC:* 21](#_Toc75116821)

[3. Kiến trúc tổng thể hệ thống: 22](#_Toc75116822)

[4. Thiết kế chi tiết: 22](#_Toc75116823)

[*a. Sơ đồ tổng quan và chi tiết tầng trình bày:* 22](#_Toc75116824)

[*b. Sơ đồ tổng quan và chi tiết tầng ứng dụng và nghiệp vụ:* 28](#_Toc75116825)

[*c. Sơ đồ tổng quan và chi tiết tầng truy xuất dữ liệu:* 30](#_Toc75116826)

[*d. Thiết kế tầng lưu trữ dữ liệu:* 33](#_Toc75116827)

[5. Kết quả: 33](#_Toc75116828)

# Giới thiệu:

Đề tài em chọn cho môn học là “Xây dựng ứng dụng nhắn tin trên nền tảng Web”. Lý do em chọn đề tài này là vì chúng ta đang sống giữa đại dịch, mọi người thực hiện giãn cách xã hội, khiến việc giao tiếp giữa mọi người trở nên khó khăn hơn. Đồng thời, hạ tầng hệ thống thông tin cũng đang phát triển mạnh, đặc biệt là sự phổ biến của điện thoại thông minh và khả năng kết nối Internet tốc độ cao 4G, và 5G đang được thử nghiệm ở nước ta, cho phép truy cập Internet nhanh chóng và ổn định. Vậy nên một ứng dụng cho phép người dùng kết nối và tương tác thời gian thực sẽ giúp cho việc giao tiếp trong mùa đại dịch trở nên dễ dàng hơn. Báo cáo trình bày kết quả đạt được, các công nghệ sử dụng khi thực hiện đề tài và một số hướng phát triển của ứng dụng.

# Nội dung:

## Phân tích yêu cầu hệ thống:

### *Các tác nhân hệ thống:*

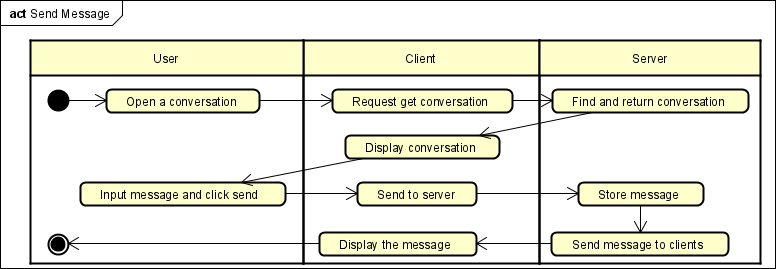
* Hệ thống gồm các tác nhân Guest và User.
* Guest: Tác nhân không đăng nhập, có thể đăng nhập và đăng ký.
* User: Tác nhân đã đăng nhập, có thể sử dụng đầy đủ tính năng.

### *Mô hình hệ thống:*

* Hệ thống sử dụng mô hình Client-Server.
* Client: Tương tác với Guest và User, qua Client, Guest và User gửi yêu cầu tới Server để xử lý theo mô hình Request-Response.
* Server: Nhận các Request của Client, xử lý và trả về Response.
* Ngoài ra, Server đảm nhận vai trò trung gian, chuyển tiếp tin nhắn giữa các Client, và đóng vai trò Signaling trong sử dụng WebRTC để thực hiện truyền thông thời gian thực với hình ảnh và âm thanh.

### *Một số nghiệp vụ cơ bản:*

* Gửi tin nhắn:



* Gọi điện:

Diagram

Description automatically generated

### *Các biểu đồ ca sử dụng của hệ thống:*

* Tổng quan:

Diagram

Description automatically generated

* Ca sử dụng “Quản lý bạn bè”:

Diagram

Description automatically generated

* Ca sử dụng “Quản lý nhóm”:

Diagram

Description automatically generated

* Ca sử dụng “Mở cuộc hội thoại với bạn bè”:

Diagram

Description automatically generated

* Ca sử dụng “Mở cuộc hội thoại với nhóm”:

Diagram

Description automatically generated

* Ca sử dụng “Tìm kiếm”:

Diagram

Description automatically generated

* Ca sử dụng “Kiểm tra yêu cầu kết bạn”:

Diagram

Description automatically generated

#### Đặc tả ca sử dụng:

* Ca sử dụng “Đăng nhập”:

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Usecase | UC01 | Tên Use case | Đăng nhập |
| Mục đích sử dụng | Giúp Guest đăng nhập để trở thành User | | |
| Tác nhân | Guest | | |
| Sự kiện kích hoạt | Khi Guest truy cập địa chỉ ứng dụng | | |
| Tiền điều kiện | Không | | |
| Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Client | Hiển thị trang đăng nhập | | 2 | Guest | Nhập tài khoản và mật khẩu | | 3 | Guest | Bấm “Login” | | 4 | Client | Gửi yêu cầu đăng nhập tới Server | | 5 | Server | Kiểm tra đăng nhập và gửi thông tin User | | 6 | Client | Hiển thị thông tin trang chủ với thông tin User | | | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 2a | Guest | Nhập tài khoản và mật khẩu sai | | 5a | Server | Kiểm tra đăng nhập và gửi về lỗi | | 6a | Client | Báo lỗi đăng nhập | | | |
| Hậu điều kiện | Guest đăng nhập thành công trở thành User | | |

* Ca sử dụng “Đăng ký”:

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Usecase | UC02 | Tên Use case | Đăng ký |
| Mục đích sử dụng | Giúp Guest đăng ký tài khoản | | |
| Tác nhân | Guest | | |
| Sự kiện kích hoạt | Khi Guest bấm nút “Sign Up” trên trang đăng nhập | | |
| Tiền điều kiện | Không | | |
| Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Client | Hiển thị trang đăng nhập | | 2 | Guest | Chọn nút “Sign Up” | | 3 | Client | Hiển thị trang đăng ký | | 4 | Guest | Nhập thông tin đăng ký và bấm “Register” | | 5 | Client | Kiểm tra nhập đủ các trường | | 6 | Client | Gửi thông tin đăng ký tới Server | | 7 | Server | Kiểm tra thông tin đăng ký và báo thành công | | 8 | Client | Hiển thị thông báo thành công | | | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 4a | Guest | Nhập thông tin không hợp lệ | | 6a | Client | Báo lỗi thông tin không hợp lệ | | 4b | Guest | Nhập tài khoản đã tồn tại | | 8b | System | Báo lỗi tài khoản đã tồn tại | | | |
| Hậu điều kiện | Tài khoản mới được tạo | | |

Dữ liệu đầu vào của thông tin đăng ký:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Trường dữ liệu | Bắt buộc | Điều kiện hợp lệ | Ví dụ |
| 1 | Email | Có | Dạng Email | tinh.pt183997@sis.hust.edu.vn |
| 2 | Tên người dùng | Có | Viết liền không dấu, 14 ký tự | Tinhbg0612 |
| 3 | Mật khẩu | Có | Độ dài hơn 6 ký tự | 12345678 |
| 4 | Ngày sinh | Có | dd/mm/YYYY | 06/12/2000 |
| 5 | Giới tính | Có | Nam/Nữ/Khác | Nam |

* Ca sử dụng “Thông báo online”:

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Usecase | UC03 | Tên Use case | Thông báo online |
| Mục đích sử dụng | Giúp thông báo User trực tuyến tới bạn bè | | |
| Tác nhân | Client | | |
| Sự kiện kích hoạt | Khi User vừa đăng nhập | | |
| Tiền điều kiện | Không | | |
| Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Client | Gửi thông báo online tới Server | | 2 | Server | Chuyển tiếp thông báo đó tới Client của bạn bè | | 3 | Server | Cập nhật trạng thái User | | 4 | Friend’s Client | Nhận thông báo, chuyển trạng thái User online | | 5 | Friend’s Client | Gửi thông báo “cùng online” tới Server | | 6 | Server | Chuyển tiếp thông báo tới Client của User | | 7 | Client | Chuyển trạng thái của bạn sang online | | | |
| Luồng sự kiện thay thế | Không | | |
| Hậu điều kiện | Trạng thái của User trở thành online, biết được người dùng nào online | | |

* Ca sử dụng “Thông báo offline”:

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Usecase | UC04 | Tên Use case | Thông báo offline |
| Mục đích sử dụng | Giúp thông báo User offline tới bạn bè | | |
| Tác nhân | Client | | |
| Sự kiện kích hoạt | Khi User vừa đăng xuất hay ngắt kết nối | | |
| Tiền điều kiện | Không | | |
| Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | Client | Gửi thông báo ngắt kết nối tới Server | | 2 | Server | Chuyển tiếp thông báo đó tới Client của bạn bè | | 3 | Server | Cập nhật trạng thái User | | 4 | Friend’s Client | Nhận thông báo, chuyển trạng thái User offline | | | |
| Luồng sự kiện thay thế | Không | | |
| Hậu điều kiện | Trạng thái của User trở thành offline | | |

* Ca sử dụng “Mở cuộc hội thoại với bạn bè”:

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Usecase | UC05 | Tên Use case | Mở cuộc hội thoại với bạn |
| Mục đích sử dụng | Mở cuộc hội thoại giữa User và bạn | | |
| Tác nhân | User | | |
| Sự kiện kích hoạt | Khi User bấm vào một liên lạc là bạn | | |
| Tiền điều kiện | Không | | |
| Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | User | Chọn một liên lạc là bạn bè | | 2 | Client | Gửi yêu cầu mở cuộc hội thoại đến Server | | 3 | Server | Kiểm tra thông tin và trả về cuộc hội thoại | | 4 | Client | Hiển thị cuộc hội thoại | | | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 3 | Server | Thông tin không hợp lệ, trả về lỗi | | 4 | Client | Báo lỗi | | | |
| Hậu điều kiện | Hiển thị cuộc hội thoại giữa User và bạn | | |

* Ca sử dụng “Mở cuộc hội thoại với nhóm”:

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Usecase | UC06 | Tên Use case | Mở cuộc hội thoại với nhóm |
| Mục đích sử dụng | Mở cuộc hội thoại giữa User và nhóm | | |
| Tác nhân | User | | |
| Sự kiện kích hoạt | Khi User bấm vào một liên lạc là nhóm | | |
| Tiền điều kiện | Không | | |
| Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | User | Bấm vào một liên lạc là nhóm | | 2 | Client | Gửi yêu cầu mở cuộc hội thoại đến Server | | 3 | Server | Kiểm tra thông tin và trả về cuộc hội thoại | | 4 | Client | Hiển thị cuộc hội thoại | | | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 3 | Server | Thông tin không hợp lệ, trả về lỗi | | 4 | Client | Báo lỗi | | | |
| Hậu điều kiện | Hiển thị cuộc hội thoại giữa User và nhóm | | |

* Ca sử dụng “Gửi tin nhắn”:

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Usecase | UC07 | Tên Use case | Gửi tin nhắm |
| Mục đích sử dụng | Giúp User gửi tin nhắn tới một cuộc hội thoại | | |
| Tác nhân | User | | |
| Sự kiện kích hoạt | Khi User nhấn vào ô nhập tin nhắn hoặc nút đính kèm tệp tin | | |
| Tiền điều kiện | User ở trong một cuộc hội thoại | | |
| Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | User | Nhập tin nhắn hoặc đính kèm tệp tin | | 2 | User | Bấm “Send” | | 3 | Client | Gửi tin nhắn tới Server | | 4 | Server | Lưu và gửi tới các bên trong cuộc hội thoại | | 5 | Client | Hiển thị tin nhắn mới | | | |
| Luồng sự kiện thay thế | Không | | |
| Hậu điều kiện | Tin nhắn mới trong cuộc hội thoại | | |

* Ca sử dụng “Tìm kiếm”:

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Usecase | UC08 | Tên Use case | Tìm kiếm |
| Mục đích sử dụng | Giúp User tìm kiếm người dùng hay nhóm đã tham gia | | |
| Tác nhân | User | | |
| Sự kiện kích hoạt | Khi User nhập vào ô tìm kiếm | | |
| Tiền điều kiện | Không | | |
| Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | User | Nhập tên tìm kiếm | | 2 | Client | Gửi yêu cầu tìm kiếm tới Server | | 3 | Server | Thực hiện tìm kiếm và trả về kết quả | | 4 | Client | Hiển thị kết quả | | | |
| Luồng sự kiện thay thế | Không | | |
| Hậu điều kiện | Danh sách liên lạc thay thế bằng danh sách kết quả tìm kiếm | | |

* Ca sử dụng “Cập nhật danh sách đã xem”:

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Usecase | UC09 | Tên Use case | Cập nhật danh sách đã xem |
| Mục đích sử dụng | Giúp cập nhật danh sách những người đã xem cuộc hội thoại | | |
| Tác nhân | User | | |
| Sự kiện kích hoạt | Khi User mở cuộc hội thoại hay nhận tin nhắn mới trong cuộc hội thoại | | |
| Tiền điều kiện | Không | | |
| Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | User | Mở một cuộc hội thoại, nhận tin nhắn mới | | 2 | Client | Thông báo đã xem tới Server | | 3 | Server | Cập nhật danh sách đã xem của cuộc hội thoại | | 4 | Server | Gửi danh sách đã xem tới các Client | | 5 | Client | Cập nhật danh sách đã xem | | | |
| Luồng sự kiện thay thế | Không | | |
| Hậu điều kiện | Danh sách những User đã xem tin nhắn được cập nhật | | |

* Ca sử dụng “Kiểm tra yêu cầu kết bạn”:

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Usecase | UC10 | Tên Use case | Kiểm tra yêu cầu kết bạn |
| Mục đích sử dụng | Kiểm tra yêu cầu kết bạn giữa hai User và hiển thị đề xuất | | |
| Tác nhân | User | | |
| Sự kiện kích hoạt | Khi User bấm vào một liên lạc không phải bạn bè hay nhóm | | |
| Tiền điều kiện | Sau khi User thực hiện tìm kiếm | | |
| Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | User | Chọn liên lạc trong kết quả tìm kiếm | | 2 | Client | Hiện thị giao diện đề xuất theo yêu cầu kết bạn | | | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1a | User | Chọn liên lạc là bạn bè hay nhóm | | 2a | Client | Thực hiện mở cuộc hội thoại với mục tiêu | | | |
| Hậu điều kiện | Giao diện đề xuất được hiển thị | | |

* Ca sử dụng “Gửi yêu cầu kết bạn”:

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Usecase | UC11 | Tên Use case | Gửi yêu cầu kết bạn |
| Mục đích sử dụng | Giúp User gửi yêu cầu kết bạn tới một User khác | | |
| Tác nhân | User | | |
| Sự kiện kích hoạt | Khi User bấm “Send” trong giao diện đề xuất | | |
| Tiền điều kiện | Kiểm tra yêu cầu kết bạn không có giữa hai User, không phải bạn bè | | |
| Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | User | Bấm vào “Send” trong giao diện đề xuất | | 2 | Client | Gửi yêu cầu kết bạn tới Server | | 3 | Server | Thêm vào các yêu cầu kết bạn User đã gửi của User và các yêu cầu kết bạn đối phương đã nhận | | 4 | Server | Thông báo thành công tới các bên | | 5 | Client | Yêu cầu cập nhât danh sách yêu cầu kết bạn | | 6 | Server | Trả về danh sách yêu cầu kết bạn | | 7 | Client | Cập nhật | | | |
| Luồng sự kiện thay thế | Không | | |
| Hậu điều kiện | Thêm yêu cầu kết bạn vào hai danh sách yêu cầu kết bạn | | |

* Ca sử dụng “Hủy bỏ yêu cầu kết bạn”:

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Usecase | UC12 | Tên Use case | Hủy bỏ yêu cầu kết bạn |
| Mục đích sử dụng | Giúp User hủy yêu cầu kết bạn tới một User khác | | |
| Tác nhân | User | | |
| Sự kiện kích hoạt | Khi User bấm “Cancel” trong giao diện đề xuất | | |
| Tiền điều kiện | Kiểm tra User đã gửi yêu cầu kết bạn từ đối phương | | |
| Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | User | Bấm “Cancel” trong giao diện đề xuất | | 2 | Client | Yêu cầu hủy bỏ yêu cầu kết bạn tới Server | | 3 | Server | Xóa khỏi các yêu cầu kết bạn User đã gửi và các yêu cầu kết bạn đối phương đã nhận | | 4 | Server | Thông báo thành công tới các bên | | 5 | Client | Yêu cầu cập nhật danh sách yêu cầu | | 6 | Server | Trả về danh sách yêu cầu kết bạn | | 7 | Client | Cập nhật | | | |
| Luồng sự kiện thay thế | Không | | |
| Hậu điều kiện | Xóa yêu cầu khỏi hai danh sách yêu cầu kết bạn | | |

* Ca sử dụng “Chấp nhận yêu cầu kết bạn”:

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Usecase | UC13 | Tên Use case | Chấp nhận yêu cầu kết bạn |
| Mục đích sử dụng | Giúp User chấp nhận yêu cầu kết bạn từ một User khác | | |
| Tác nhân | User | | |
| Sự kiện kích hoạt | Khi User bấm “Accept” trong giao diện đề xuất | | |
| Tiền điều kiện | Kiểm tra đối phương đã gửi yêu cầu kết bạn tới User | | |
| Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | User | Bấm “Accept” trong giao diện đề xuất | | 2 | Client | Yêu cầu chấp nhận lời mời kết bạn tới Server | | 3 | Server | Xóa yêu cầu từ danh sách yêu cầu kết bạn đã gửi của User và danh sách yêu cầu kết bạn đã nhận của đối phương | | 4 | Server | Tạo cuộc hội thoại mới giữa hai User | | 5 | Server | Thêm User vào danh sách bạn bè của đối phương, và thêm đối phương vào danh sách bạn bè của User | | 7 | Server | Thông báo thành công tới các bên | | 8 | Client | Yêu cầu cập nhật bạn bè và yêu cầu kết bạn | | 9 | Server | Trả về danh sách bạn bè và yêu cầu kết bạn | | 10 | Client | Cập nhật | | | |
| Luồng sự kiện thay thế | Không | | |
| Hậu điều kiện | Xóa yêu cầu khỏi hai danh sách yêu cầu kết bạn và thêm các User vào hai danh sách bạn bè | | |

* Ca sử dụng “Từ chối yêu cầu kết bạn”:

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Usecase | UC14 | Tên Use case | Từ chối yêu cầu kết bạn |
| Mục đích sử dụng | Giúp User từ chối yêu cầu kết bạn từ một User khác | | |
| Tác nhân | User | | |
| Sự kiện kích hoạt | Khi User bấm “Reject” trong giao diện đề xuất | | |
| Tiền điều kiện | Kiểm tra đối phương đã gửi yêu cầu kết bạn tới User | | |
| Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | User | Bấm “Reject” trong giao diện đề xuất | | 2 | Client | Yêu cầu từ chối lời mời kết bạn tới Server | | 3 | Server | Xóa yêu cầu từ danh sách yêu cầu kết bạn đã gửi của User và danh sách yêu cầu kết bạn đã nhận của đối phương | | 4 | Server | Thông báo thành công | | 5 | Client | Yêu cầu cập nhật danh sách yêu cầu kết bạn | | 6 | Server | Trả về danh sách yêu cầu kết bạn | | 7 | Client | Cập nhật | | | |
| Luồng sự kiện thay thế | Không | | |
| Hậu điều kiện | Xóa yêu cầu khỏi hai danh sách yêu cầu kết bạn | | |

* Ca sử dụng “Hủy kết bạn”:

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Usecase | UC15 | Tên Use case | Hủy kết bạn |
| Mục đích sử dụng | Giúp User hủy kết bạn với một User khác | | |
| Tác nhân | User | | |
| Sự kiện kích hoạt | Khi User bấm “Unfriend” khi ở trong cuộc hội thoại với đối phương | | |
| Tiền điều kiện | User ở trong cuộc hội thoại với đối phương | | |
| Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | User | Bấm “Unfriend” | | 2 | Client | Gửi yêu cầu hủy kết bạn | | 3 | Server | Xóa User khỏi danh sách bạn bè đối phương và đối phương khỏi danh sách bạn bè User | | 4 | Server | Xóa cuộc hội thoại giữa hai User | | 5 | Server | Thông báo thành công | | 6 | Client | Thoát User khỏi cuộc hội thoại, thoát đối phương nếu cùng trong cuộc hội thoại và yêu cầu cập nhật danh sách bạn bè của hai User | | 6 | Server | Trả về danh sách bạn bè | | 7 | Client | Cập nhật danh sách bạn bè và hiển thị | | | |
| Luồng sự kiện thay thế | Không | | |
| Hậu điều kiện | User bị xóa khỏi khỏi danh sách bạn bè đối phương và đối phương bị xóa khỏi danh sách bạn bè User | | |

* Ca sử dụng “Tạo nhóm”:

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Usecase | UC16 | Tên Use case | Tạo nhóm |
| Mục đích sử dụng | Giúp User tạo một nhóm | | |
| Tác nhân | User | | |
| Sự kiện kích hoạt | Khi User bấm “Create Group” | | |
| Tiền điều kiện | Không | | |
| Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | User | Bấm “Create Group” | | 2 | Client | Hiển thị giao diện tạo nhóm | | 3 | User | Nhập tên nhóm và bấm “Create” | | 4 | Client | Yêu cầu tạo nhóm tới Server | | 5 | Server | Tạo cuộc hội thoại nhóm | | 6 | Server | Thêm nhóm vào danh sách nhóm của User | | 7 | Server | Báo thành công | | 8 | Client | Yêu cầu cập nhật danh sách nhóm | | 9 | Server | Trả về danh sách nhóm | | 10 | Client | Cập nhật danh sách nhóm và hiển thị | | | |
| Luồng sự kiện thay thế | Không | | |
| Hậu điều kiện | Nhóm mới được thêm vào danh sách nhóm của User | | |

* Ca sử dụng “Thêm bạn vào nhóm”:

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Usecase | UC17 | Tên Use case | Thêm bạn vào nhóm |
| Mục đích sử dụng | Giúp User thêm bạn bè vào một nhóm | | |
| Tác nhân | User | | |
| Sự kiện kích hoạt | Khi User bấm “Add Friend to Group” trong một cuộc hội thoại nhóm | | |
| Tiền điều kiện | User đang ở trong cuộc hội thoại nhóm | | |
| Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | User | Bấm “Add Friend to Group” | | 2 | Client | Hiển thị danh sách bạn chưa ở trong nhóm. | | 3 | User | Bấm “Add” | | 4 | Client | Yêu cầu thêm người bạn vào nhóm | | 5 | Server | Thêm người bạn vào danh sách User trong cuộc hội thoại của nhóm | | 6 | Server | Thêm nhóm vào danh sách nhóm của bạn | | 7 | Server | Thêm các tin nhắn thông báo thêm thành viên vào cuộc hội thoại của nhóm | | 8 | Server | Thông báo thành công | | 9 | Client | Yêu cầu cập nhanh danh sách thành viên nhóm và danh sách nhóm | | 10 | Server | Trả về anh sách thành viên và danh sách nhóm | | 11 | Client | Cập nhật và hiển thị | | | |
| Luồng sự kiện thay thế | Không | | |
| Hậu điều kiện | Cập nhật danh sách nhóm của các User | | |

* Ca sử dụng “Rời nhóm”:

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Usecase | UC18 | Tên Use case | Rời nhóm |
| Mục đích sử dụng | Giúp User rời một nhóm | | |
| Tác nhân | User | | |
| Sự kiện kích hoạt | Khi User bấm “Leave Group” trong một cuộc hội thoại nhóm | | |
| Tiền điều kiện | User đang ở trong cuộc hội thoại nhóm | | |
| Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | User | Bấm “Leave Group” | | 2 | Client | Yêu cầu rời nhóm tới Server | | 2 | Server | Xóa nhóm khỏi danh sách nhóm của User | | 3 | Server | Kiểm tra User không phải thành viên cuối cùng | | 4 | Server | Xóa User khỏi danh sách thành viên của nhóm | | 5 | Server | Thêm tin nhắn User rời nhóm vào hội thoại | | 6 | Server | Thông báo với User | | 7 | Client | Thoát User khỏi cuộc hội thoại nhóm | | 8 | Client | Yêu cầu cập nhật danh sách nhóm | | 9 | Server | Trả về danh sách nhóm | | 10 | Client | Cập nhật và hiển thị | | | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 3a | Server | Kiểm tra User là thành viên cuối cùng | | 4a | Server | Xóa nhóm, xóa cuộc hội thoại nhóm | | 5a | Server | Thông báo với User và thoát User khỏi cuộc hội thoại nhóm | | | |
| Hậu điều kiện | Cập nhật danh sách nhóm của các User và danh sách thành viên nhóm | | |

* Ca sử dụng “Gọi video”:

|  |  |  |  |
| --- | --- | --- | --- |
| Mã Usecase | UC19 | Tên Use case | Gọi video |
| Mục đích sử dụng | Giúp User thực hiện cuộc gọi video với một người bạn | | |
| Tác nhân | User | | |
| Sự kiện kích hoạt | Khi User bấm “Call” trong cuộc hội thoại với bạn | | |
| Tiền điều kiện | User đang ở trong cuộc hội thoại với bạn | | |
| Luồng sự kiện chính | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 1 | User | Bấm “Call” | | 2 | Client | Gửi yêu cầu gọi tới Server | | 2 | Server | Chuyển tiếp yêu cầu đến Friend’s Client | | 3 | Friend’s Client | Hiển thị cuộc gọi tới | | 4 | Friend User | Bấm “Answer” | | 5 | Friend’s Client | Gửi lời đồng ý tới Server | | 6 | Server | Chuyển tiếp lời đồng ý tới User’s Client | | 7 | Client | Tạo kết nối Peer-to-Peer với Friend’s Client | | 8 | User | Bắt đầu cuộc hội thoại | | | |
| Luồng sự kiện thay thế | |  |  |  | | --- | --- | --- | | STT | Thực hiện bởi | Hành động | | 4a | Friend User | Bấm “Reject” | | 5a | Friend’s Client | Gửi lời từ chối đến Server | | 6a | Server | Chuyển tiếp lời từ chối tới User’s Client | | 7a | Client | Dừng cuộc gọi | | | |
| Hậu điều kiện | Không | | |

## Các công nghệ sử dụng:

### *Node.js:*

* Node.js là một JavaScript runtime, xây dựng trên nền tảng V8 JavaScript Engine, cho phép chạy mã nguồn JavaScript ngoài trình duyệt, thường dùng xây dựng các ứng dụng Internet mà chủ yếu là các ứng dụng Server.
* Vấn đề I/O Scaling: Việc truy xuất dữ liệu từ môi trường Internet hay ổ đĩa tiêu tốn tài nguyên hơn hẳn so với RAM hay cache. Trong khi đó, phần lớn các ứng dụng web phụ thuộc vào việc đọc dữ liệu từ đĩa hay từ một nguồn thông qua mạng (Ví dụ như truy vấn từ một cơ sở dữ liệu nằm trên thiết bị khác, cần kết nối qua mạng). Khi có một HTTP request được nhận, dữ liệu cần được nạp từ database hay đọc từ đĩa, nên việc chờ đợi là không tránh khỏi. Các kết nối mở và request tiêu tốn tài nguyên của Server. Và để có thể đáp ứng một số lượng lớn các yêu cầu từ nhiều Client dùng chung một Web Server, chúng ta có vấn đề I/O Scaling.
* Web Server truyền thống sử dụng mỗi tiến trình với một Request. Khi một Client gửi HTTP request tới, Server sử dụng một tiến trình mới để xử lý, tuy nhiên trong quá trình chờ kết quả truy vấn cơ sở dữ liệu, tiến trình ở trạng thái rảnh rỗi, nhưng vẫn tiêu tối tài nguyên là CPU và RAM. Ngoài ra, khởi động tiến trình mất tương đối thời gian và chiếu dụng nhiều không gian bộ nhớ. Web Server như vậy sẽ khó đáp ứng được nhiều request từ client. Đó là lý do các ứng dụng web hiện đại sử dụng thread pool.
* Web Server truyền thống sử dụng thread pool. Bằng cách sử dụng mỗi luồng từ thread pool cho phép thay thế thời gian bắt đầu hay kết thúc tiến trình bằng thời gian tạo hay kết thúc luồng với chi phí rẻ hơn. Tuy nhiên, vẫn có sự lãng phí RAM giữa các luồng. Đồng thời, Hệ điều hành cần chuyển giữa các luồng (dù có đang trong trạng thái nhàn rỗi), dẫn tới sự lãng phí tài nguyên CPU.
* Node.js sử dụng một luồng duy nhất để đáp ứng các request. Ý tưởng server sử dụng một luồng đã được áp dụng trước Node.js ở Nginx. Nginx là một web server đơn luồng có thể đáp ứng một lượng lớn các yêu cầu đồng thời với mức tài nguyên tiêu thụ ổn định.
* Javascript có một luồng thực thi. Đó là cách truyền thống mà trình duyệt hoạt động. Khi có một hoạt động cần chạy thời gian dài (Ví dụ như chờ kết quả truy vấn cơ sở dữ liệu), thì việc thực thi hoạt động cần được tiếp tục diễn ra trong hàm callback. Và Node.js có hiệu năng cao, sử dụng JavaScript, ngôn ngữ cho phép định nghĩa một hàm trong một hàm khác và sử dụng các biến của hàm bên ngoài, và khả năng truyền hàm như một tham số vào trong hàm khác.
* Event Loop trong Node.js: JavaScript có callstack chứa các hàm cần thực thi theo thứ tự Last In, First Out. Event Loop duyệt callstack và thực hiện lần lượt các hàm cho đến khi rỗng. Event Queue là một hàng đợi các hàm xử lý sự kiện, sau khi thực hiện hết callstack, các hàm từ Event Queue sẽ được nạp vào callstack để Event Loop thực hiện. Việc phải chờ đợi callstack thực hiện hết có thể gây ra đói luồng (Thread Starvation) do khi callstack quá lớn, các hàm xử lý sự kiện trong Event Queue sẽ phải đợi lâu, trong khi việc xử lý I/O cần phải nhanh trong mô hình Client-Server. ECMAScript 2015 hay ES6 cung cấp một Job Queue và Promise, cho phép thực thi các hàm bất đồng bộ, hàm xử lý sự kiện sớm nhất thay vì đẩy xuống cuối callstack và phải chờ callstack thực hiện xong. Sử dụng Promise cho phép thực hiện hàm ngay khi hàm hiện tại thực hiện xong trong callstack. async/await được xây dựng dựa trên Promise cũng có tác dụng tương tự. Event Loop cho phép ứng dụng nhận request từ client, thực hiện truy vấn với yêu cầu đó, trong khi chờ đợi truy vấn hoàn thành, ứng dụng tiếp tục nhận request và thực hiện truy vấn. Khi truy vấn thành công, tiếp tục thực hiện và hoàn thành các xử lý request.

### *Vue.js:*

* Vue.js là một front-end framework hỗ trợ những chức năng cơ bản như data-binding, list rendering, sorting và event listening.
* Tương tự như các front-end framework khác, Vue cung cấp khả năng tạo các component riêng biệt trong một ứng dụng nhằm đảm bảo khả năng tái sử dụng và dễ bảo trì. Single-File Component của Vue cho phép định nghĩa cả view và logic (HTML/CSS và JavaScript) trong một tệp .vue đơn lẻ. Việc chia sẻ dữ liệu giữa các component được chia làm ba dạng: Parent-Child, Child-Parent và giữa các Sibling. Từ parent tới child có thể truyền dữ liệu nhờ vào props, cho phép child component sử dụng dữ liệu từ parent mà không cho phép thay đổi trạng thái của parent. Tuy nhiên theo chiều ngược lại, child component chỉ có thể giao tiếp với parent thông qua các sự kiện tự định nghĩa, khi cần truyền dữ liệu, child component phát một sự kiện, còn parent lắng nghe sự kiện đó và thực hiện một công việc. Giữa các Sibling có thể sử dụng các sự kiện tự định nghĩa hoặc thư viện quản lý trạng thái Vuex.
* Vuex là một thư viện quản lý trạng thái dựa trên design pattern Flux. Flux là design pattern tạo ra bởi Facebook, gồm bốn phần: Action, Dispatcher, Store và View, tạo thành đường ống dữ liệu một chiều. Người dùng tương tác với view, view kích hoạt action mô tả chuyện gì diễn ra. Store ghi nhận action đó và quyết định trạng thái nào đã thay đổi. Flux giúp chia nhỏ trạng thái thành các phần nhỏ hơn, cô lập và dễ kiểm thử. Vuex xây dựng dựa trên Flux, xây dựng dành riêng cho Vue. Vuex có một store pattern đơn giản với sự định nghĩa riêng biệt các getters, mutations, và actions. Ý tưởng chính của Vuex là dữ liệu được lưu trữ trong một cấu trúc state, nằm trong store. Ứng dụng đọc dữ liệu state từ store và state sẽ không bao giờ được thay đổi ngoài store. Người dùng thông qua view kích hoạt các action để mô tả chuyện gì đã xảy ra. Action thực hiện các mutation để thay đổi state. Và khi state thay đổi, view được cập nhật.
* Vue-router là thư viện cho phép xây dựng một Single Page Application với Vue. Single Page Application là ứng dụng web chỉ tải một lần và việc chuyển giữa các trang sẽ do JavaScript xử lý động. Vue-router cho phép định tuyến phía client với các kỹ thuật như đặt guard, dynamic matching route.

### *MongoDB:*

* MongoDB là một cơ sở dữ liệu NoSQL, lưu trữ dữ liệu dưới dạng các document.
* NoSQL là một thuật ngữ chỉ thế hệ cơ sở dữ liệu kiểu mới mới, khác với các cơ sở dữ liệu quan hệ SQL truyền thống và được chia làm 4 dạng: Document database – MongoDB, Key-value database - Redis, Column-family database – Cassandra và Graph database – Neo4J. Tất cả được xây dựng với mục đích là Scalability.
* Document database là cơ sở dữ liệu hoạt động dựa trên khái niệm document – một gói thông tin về một thực thể nhất định, thường dùng định dạng JSON để lưu trữ, ngoài ra còn có XML hay dạng nhị phân.
* Lý do chọn NoSQL là khả năng Scalability. Trong khi các thiết kết quan hệ gặp khó trong vấn đề scale. Các cơ sở dữ liệu cần đáp ứng tính Availability – Khả năng cung cấp dữ liệu và Consistency – Tính đúng đắn và đồng nhất của dữ liệu. Thường trên một thiết bị duy nhất, Consistency và Availability gắn chặt với nhau, nếu dữ liệu sẵn có, thì sẽ đồng nhất. Tuy nhiên với một thiết bị thì khó có thể đáp ứng được tất cả yêu cầu từ client, cũng như phân chia các máy chủ để có thể hoạt động trên nhiều vùng địa lý, nhằm phục vụ nhanh nhất các client ở vị trí địa lý đó. Và khi đó dữ liệu được chia làm các phần. Các phần này phải được kết nối với nhau đảm bảo tính Consistency. Và trong trường hợp xảy ra mất kết nối giữa các phần, thì khi đó buộc phải lựa chọn khi có một yêu cầu từ người dùng, là phản hồi yêu cầu đó để đảm bảo availability, hay từ chối để duy trì consistency. Vấn đề này có thể giải quyết thông qua việc chia dữ liệu thành các đơn vị chứa toàn bộ các thông tin của nó mà không cần lấy từ bên ngoài, gọi là các shard. Ví dụ như nhóm người dùng ở Mỹ và nhóm người dùng Việt Nam sẽ có hai máy chủ chứa dữ liệu của Mỹ và Việt Nam. Khi bị mất kết nối, người dùng ở Việt Nam vẫn có thể sử dụng dữ liệu từ máy chủ Việt Nam. Với cơ sở dữ liệu quan hệ, việc có các ràng buộc chặt với nhau sẽ dẫn đến khó có thể tách rời. Khi mà một cơ sở dữ liệu quan hệ sẽ khuyến khích việc có nhiều quan hệ giữa các bảng.

### *WebRTC:*

* Giao tiếp thời gian thực đã trở nên phổ biến với các thiết bị di động thông minh với các ứng dụng như Apple’s Facetime, Google’s Hangout hay Facebook’s Messenger. WebRTC được xây dựng với mục tiêu mang giao tiếp thời gian thực tới trình duyệt.
* Kết nối thời gian thực với WebRTC là kết nối Peer-to-Peer do việc truyền dữ liệu hình ảnh và âm thanh qua một Server trung gian rất tốn kém và không hiệu quả. WebRTC sử dụng kết nối UDP để truyền dữ liệu để đảm bảo tốc độ, ứng dụng được phép mất một số khung dữ liệu, hay gửi dữ liệu mới nhất quan trọng hơn đảm bảo tất cả dữ liệu tới được phía bên kia. Dù sử dụng UDP, thì việc truyền dữ liệu hình ảnh và âm thanh trực tiếp vẫn là không thể do dữ liệu quá lớn, dẫn đến yêu cầu phải chia nhỏ dữ liệu và nén để truyền đi, bên nhận sẽ giải nén và sử dụng, thuật toán sử dụng ở đây là codec, và vì có nhiều loại codec, dẫn tới hai bên trong kết nối phải biết được đối phương dùng loại nào, cũng như nhiều thông tin quan trọng khác.
* Tạo kết nối giữa P2P giữa các trình duyệt được thực hiện thông qua quá trình Signaling và Negotiating. Signaling là việc các bên tạo các candidate, và gửi cho nhau thông qua một Signaling Server, có thể là Server của ứng dụng, và khi hai bên đồng ý kết nối, thì các candidate sẽ được duyệt với thứ tự ưu tiên, chọn ra kết nối tốt nhất cho cả hai bên. Negotiating diễn ra dưới mô hình offer/answer. Bên yêu cầu kết nối sẽ tạo và gửi một offer thông qua Signaling Server, ở đó offer được chuyển tiếp tới bên nhận. Bên nhận sẽ nhận offer và trả lời bằng một answer nếu đồng ý. Offer và answer ở đây sẽ chứa thông tin mô tả phiên mỗi trình duyệt (Session Description), bao gồm những thông tin như những loại thiết bị hỗ trợ hay video và audio codec.
* Việc tạo kết nối P2P giữa các trình duyệt gặp trở ngại do Network Address Translation (NAT). Do số lượng địa chỉ Ipv4 có hạn, trong một mạng cục bộ, các thiết bị sẽ có một địa chỉ cục bộ, và khi kết nối ra Internet, thiết bị cần một địa chỉ công cộng để có thể gửi và nhận dữ liệu từ ngoài Internet. Để tạo được kết nối P2P thì các trình duyệt phải biết được địa chỉ công cộng của nhau. Để biết địa chỉ công cộng của nhau, và thiết lập kết nối P2P, cần sử dụng các kỹ thuật: Session Traversal Utilities for NAT (STUN), Traversal Using Relays around NAT (TURN) và Interactive Connectivity Establishment (ICE). STUN giúp trình duyệt biết được địa chỉ công cộng bằng cách gửi thông điệp tới STUN Server, STUN Server sau đó trả về thông tin địa chỉ IP công cộng cho trình duyệt. TURN được áp dụng khi tường lửa quá nghiêm ngặt, không cho phép STUN được thực hiện. TURN sẽ thêm vào giữa các trình duyệt một Server trung gian, nhận, xử lý và chuyển gói tin giữa các bên, tuy nhiên TURN có chi phí rất cao nên thường là biện pháp cuối cùng. ICE là áp dụng STUN và TURN, đảm nhận việc nhận các candidate chứa thông tin về mạng của trình duyệt thông qua STUN Server, sau đó gửi qua Signaling Server. TURN được coi là biện pháp cuối cùng nếu STUN thất bại.
* WebRTC hiện tại được hỗ trợ trên đa số các trình duyệt phổ biến như Google Chrome, Mozilla Firefox, Opera…. Ứng dụng phổ biến nhất của WebRTC là truyền thông hình ảnh và âm thanh trên trình duyệt, và chỉ cho phép những địa chỉ có nguồn gốc đáng tin cậy như HTTPS, hoặc localhost. Ngoài ra còn một số ứng dụng khác như truyền tệp, nhưng sẽ không áp dụng trong đồ án môn học.

## Kiến trúc tổng thể hệ thống:

Diagram

Description automatically generated

## Thiết kế chi tiết:

### *Sơ đồ tổng quan và chi tiết tầng trình bày:*

* Sơ đồ tầng trình bày:

Diagram

Description automatically generated

* Chi tiết các lớp:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tên lớp | Login | | | | |
| Mục đích | Đăng nhập | | | | |
| Thuộc tính | Tên | Kiểu | Phạm vi | | Mô tả |
| Username | String | Public | | Tên đăng nhập |
| Password | String | Public | | Mật khẩu |
| Phương thức | Tên | Đầu vào | Đầu ra | Phạm vi | Mô tả |
| Login | Username,  Password | Void | Public | Đăng nhập |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tên lớp | Sign Up | | | | | |
| Mục đích | Đăng ký | | | | | |
| Thuộc tính | Tên | Kiểu | Phạm vi | | | Mô tả |
| Username | String | Public | | | Tên đăng nhập |
| Password | String | Public | | | Mật khẩu |
| Gender | String | Public | | | Giới tính |
| Email | String | Public | | | Email |
| DateOfBirth | Date | Public | | | Ngày sinh |
| Phương thức | Tên | Đầu vào | | Đầu ra | Phạm vi | Mô tả |
| Register | Username,  Password,  Gender,  Email,  DateOfBirth | | Void | Public | Đăng ký |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tên lớp | SidebarHeader | | | | | | | |
| Mục đích | Hiển thị thông tin người dùng và tìm kiếm | | | | | | | |
| Thuộc tính | Tên | | Kiểu | Phạm vi | | | | Mô tả |
| Username | | String | Public | | | | Tên tài khoản |
| SearchName | | String | Public | | | | Tên tìm kiếm |
| Phương thức | Tên | Đầu vào | | | Đầu ra | Phạm vi | Mô tả | |
| Search | SearchName | | | UserList, GroupList | Public | Tìm kiếm người dùng | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tên lớp | SidebarBody | | | | | |
| Mục đích | Hiển thị danh sách người dùng và kết quả tìm kiếm, truy cập cuộc hội thoại với người dùng | | | | | |
| Thuộc tính | Tên | Kiểu | Phạm vi | | Mô tả | |
| Friends | Friend[] | Public | | Danh sách bạn bè | |
| Groups | Group[] | Public | | Danh sách nhóm | |
| SearchResult | User[] | Public | | Kết quả tìm kiếm người dùng | |
| GroupSearchResult | Group[] | Public | | Kết quả tìm nhóm tham gia | |
| Phương thức | Tên | Đầu vào | Đầu ra | Phạm vi | | Mô tả |
| checkFriendRequest | UserId | Int | Public | | Kiểm tra yêu cầu kết bạn giữa các người dùng |
| accessContact | UserId,  TargetId | Void | Public | | Nếu mục tiêu chưa là bạn sẽ hiển thị giao diện đề xuất kết bạn, nếu mục tiêu là bạn hoặc nhóm sẽ mở cuộc hội thoại |
| openConversation | UserId,  ConId,  Name,  Type | Void | Public | | Mở cuộc hội thoại |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Tên lớp | SidebarFooter | | | | | | |
| Mục đích | Các phím chức năng đăng xuất, tạo nhóm | | | | | | |
| Thuộc tính | Tên | | Kiểu | | Phạm vi | | Mô tả |
| UserId | | String | | Public | | Id của người dùng |
| Phương thức | Tên | Đầu vào | | Đầu ra | Phạm vi | Mô tả | |
| Logout | UserId | | Void | Public | Đăng xuất | |
| CreateGroup | Không | | Void | Public | Mở giao diện tạo nhóm | |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tên lớp | ContentHeader | | | | | | | |
| Mục đích | Hiển thị tên cuộc hội thoại  Trong cuộc hội thoại với bạn bè: Hủy kết bạn, gọi điện.  Trong cuộc hội thoại nhóm: Thêm thành viên, rời nhóm | | | | | | | |
| Thuộc tính | Tên | Kiểu | | Phạm vi | | Mô tả | | |
| UserId | String | | Public | | Id của người dùng | | |
| Username | String | | Public | | Tên người dùng | | |
| OtherId | String | | Public | | Id của đối phương (bạn/nhóm) | | |
| OtherType | String | | Public | | Kiểu cuộc hội thoại (bạn/nhóm) | | |
| ConId | String | | Public | | Id của cuộc hội thoại | | |
| Phương thức | Tên | | Đầu vào | | Đầu ra | | Phạm vi | Mô tả |
| Unfriend | | UserId,  ConId | | Void | | Public | Hủy kết bạn |
| AddToGroup | | GroupName  ConId | | Void | | Public | Mở giao diện thêm bạn vào nhóm |
| LeaveGroup | | ConId | | Void | | Public | Rời nhóm |
| Call | | UserId,  OtherId | | Void | | Public | Mở giao diện gọi video, thực hiện cuộc gọi |
| OnCall | | UserId,  OtherId | | Void | | Public | Xử lý khi có cuộc gọi đến |
| Answer | | UserId,  CallerId | | Void | | Public | Trả lời cuộc gọi |
| Reject | | UserId,  CallerId | | Void | | Public | Từ chối cuộc gọi |
| OnCallerCandidate | | UserId,  CallerId,  Candidate | | Void | | Public | Nhận ICECandidate từ người gọi |
| OnICECandidate | | Candidate | | Void | | Public | Nhận ICECandidate từ ICE Server |
| SendICECandidate | | CalleeId,  Candidate | | Void | | Public | Gửi ICECandidate đã nhận từ ICE Server tới người nhận cuộc gọi |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Tên lớp | ContentBody | | | | | | |
| Mục đích | Hiển thị cuộc hội thoại | | | | | | |
| Thuộc tính | Tên | Kiểu | | | Phạm vi | Mô tả | |
| Messages | Message[] | | | Public | Danh sách tin nhắn | |
| Seen | User[] | | | Public | Danh sách người dùng đã xem | |
| UserId | String | | | Public | Id của người dùng | |
| Username | String | | | Public | Tên người dùng | |
| ConId | String | | | Public | Id của cuộc hội thoại | |
| Phương thức | Tên | | Đầu vào | Đầu ra | | Phạm vi | Mô tả |
| SeenUpdate | | UserId,  Username,  ConId | Void | | Public | Cập nhật danh sách đã xem tin nhắn cuộc hội thoại |
| OnNewMessage | | ConId,  Message | Void | | Public | Thêm tin nhắn vào cuộc hội thoại |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Tên lớp | ContentFooter | | | | | | |
| Mục đích | Nhập và gửi tin nhắn | | | | | | |
| Thuộc tính | Tên | | Kiểu | | Phạm vi | Mô tả | |
| Message | | String | | Public | Tin nhắn | |
| File | | File | | Public | Tệp | |
| UserId | | String | | Public | Id của của người dùng | |
| Username | | String | | Public | Tên người dùng | |
| ConId | | String | | Public | Id của cuộc hội thoại | |
| Phương thức | Tên | Đầu vào | | Đầu ra | | Phạm vi | Mô tả |
| Send | UserId,  Username,  ConId,  Message | | Void | | Public | Gửi tin nhắn |
| SendFile | UserId,  Username,  ConId,  File | | Void | | Public | Gửi tệp |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Tên lớp | SendRequest | | | | | | |
| Mục đích | Giao diện đề xuất gửi yêu cầu kết bạn | | | | | | |
| Thuộc tính | Tên | | Kiểu | | Phạm vi | Mô tả | |
| UserId | | String | | Public | Id của người dùng | |
| Username | | String | | Public | Tên người dùng | |
| OtherId | | String | | Public | Id của người dùng mục tiêu | |
| OtherName | | String | | Public | Tên người dùng mục tiêu | |
| Phương thức | Tên | Đầu vào | | Đầu ra | | Phạm vi | Mô tả |
| SendRequest | UserId,  Username,  OtherId,  OtherName | | Void | | Public | Gửi yêu cầu kết bạn |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Tên lớp | RequestSent | | | | | | |
| Mục đích | Giao diện đề xuất hủy yêu cầu kết bạn sau khi đã gửi | | | | | | |
| Thuộc tính | Tên | | Kiểu | | Phạm vi | Mô tả | |
| UserId | | String | | Public | Id của người dùng | |
| Username | | String | | Public | Tên người dùng | |
| OtherId | | String | | Public | Id của người dùng mục tiêu | |
| OtherName | | String | | Public | Tên người dùng mục tiêu | |
| Phương thức | Tên | Đầu vào | | Đầu ra | | Phạm vi | Mô tả |
| CancelRequest | UserId,  Username,  OtherId,  OtherName | | Void | | Public | Hủy yêu cầu kết bạn |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Tên lớp | RequestReceived | | | | | | |
| Mục đích | Giao diện đề xuất chấp nhận hoặc từ chối yêu cầu kết bạn | | | | | | |
| Thuộc tính | Tên | | Kiểu | | Phạm vi | Mô tả | |
| UserId | | String | | Public | Id của người dùng | |
| Username | | String | | Public | Tên người dùng | |
| OtherId | | String | | Public | Id của người dùng mục tiêu | |
| OtherName | | String | | Public | Tên người dùng mục tiêu | |
| Phương thức | Tên | Đầu vào | | Đầu ra | | Phạm vi | Mô tả |
| AcceptRequest | UserId,  Username,  OtherId,  OtherName | | Void | | Public | Chấp nhận yêu cầu kết bạn |
| RejectRequest | UserId,  Username,  OtherId,  Othername | | Void | | Public | Từ chối yêu cầu kết bạn |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Tên lớp | CreateGroup | | | | | | |
| Mục đích | Giao diện đề xuất chấp nhận hoặc từ chối yêu cầu kết bạn | | | | | | |
| Thuộc tính | Tên | | Kiểu | | Phạm vi | Mô tả | |
| UserId | | String | | Public | Id của người dùng | |
| Username | | String | | Public | Tên người dùng | |
| GroupName | | String | | Public | Tên nhóm | |
| Phương thức | Tên | Đầu vào | | Đầu ra | | Phạm vi | Mô tả |
| CreateGroup | UserId,  Username,  GroupName | | Void | | Public | Tạo nhóm |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Tên lớp | AddToGroup | | | | | | |
| Mục đích | Giao diện đề xuất chấp nhận hoặc từ chối yêu cầu kết bạn | | | | | | |
| Thuộc tính | Tên | | Kiểu | | Phạm vi | Mô tả | |
| UserId | | String | | Public | Id của người dùng | |
| Username | | String | | Public | Tên người dùng | |
| ConId | | String | | Public | Id của cuộc hội thoại nhóm | |
| GroupMembers | | User[] | | Public | Danh sách thành viên nhóm | |
| Friends | | Friend[] | | Public | Danh sách bạn bè | |
| Phương thức | Tên | Đầu vào | | Đầu ra | | Phạm vi | Mô tả |
| AddToGroup | UserId,  Username,  FriendId,  FriendName, ConId,  GroupName | | Void | | Public | Thêm bạn vào nhóm |

### *Sơ đồ tổng quan và chi tiết tầng ứng dụng và nghiệp vụ:*

* Sơ đồ tổng quan:

Diagram

Description automatically generated

* Chi tiết các lớp:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Tên lớp | Authenticator | | | | | | |
| Mục đích | Đăng nhập, đăng ký | | | | | | |
| Thuộc tính | Tên | | Kiểu | | Phạm vi | Mô tả | |
| Username | | String | | Public | Tên người dùng | |
| Password | | String | | Public | Mật khẩu | |
| Email | | String | | Public | Email | |
| Gender | | String | | Public | Giới tính | |
| DateOfBirth | | Date | | Public | Ngày sinh | |
| Phương thức | Tên | Đầu vào | | Đầu ra | | Phạm vi | Mô tả |
| Login | Username,  Password | | Void | | Public | Đăng nhập |
| Register | Username,  Password,  Email,  Gender,  DateOfBirth | | String | | Public | Đăng ký |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Tên lớp | Account | | | | | | |
| Mục đích | Thông tin tài khoản người dùng | | | | | | |
| Thuộc tính | Tên | | Kiểu | Phạm vi | | Mô tả | |
| Username | | String | Public | | Tên người dùng | |
| UserId | | String | Public | | Id của người dùng | |
| Friends | | Friend[] | Public | | Danh sách bạn bè | |
| Groups | | Group[] | Public | | Danh sách nhóm | |
| SentRequests | | User[] | Public | | Các yêu cầu kết bạn đã gửi | |
| ReceivedRequest | | User[] | Public | | Các yêu cầu kết bạn đã nhận | |
| SearchResult | | User[] | Public | | Kết quả tìm người dùng | |
| GroupSearchResult | | Group[] | Public | | Kết quả tìm nhóm tham gia | |
| Phương thức | Tên | Đầu vào | | | Đầu ra | Phạm vi | Mô tả |
| Initiate | UserId | | | Void | Public | Khởi tạo thông tin |
| SendRequest | UserId,  Username,  OtherId,  OtherName | | | Void | Public | Gửi yêu cầu kết bạn |
| CancelRequest | UserId,  Username,  OtherId,  OtherName | | | Void | Public | Hủy yêu cầu kết bạn |
| AcceptRequest | UserId,  Username,  OtherId,  OtherName | | | Void | Public | Đồng ý yêu cầu kết bạn |
| RejectRequest | UserId,  Username,  OtherId,  OtherName | | | Void | Public | Từ chối yêu cầu kết bạn |
| Unfriend | UserId,  Username,  OtherId,  OtherName | | | Void | Public | Hủy kết bạn |
| GetFriends | UserId | | | Void | Public | Lấy danh sách bạn bè |
| GetGroups | UserId | | | Void | Public | Lấy danh sách nhóm |
| GetRequests | UserId | | | Void | Public | Lấy danh sách yêu cầu |
| CreateGroup | UserId,  GroupName | | | Void | Public | Tạo nhóm |
| AddToGroup | UserId,  Username,  OtherId,  OtherName,  ConId | | | Void | Public | Thêm vào nhóm |
| LeaveGroup | UserId,  Username,  ConId | | | Void | Public | Rời nhóm |
| Search | UserId, SearchName | | | Void | Public | Tìm kiếm |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Tên lớp | Conversation | | | | | | | |
| Mục đích | Thông tin cuộc hội thoại | | | | | | | |
| Thuộc tính | Tên | | Kiểu | | Phạm vi | | Mô tả | |
| OtherName | | String | | Public | | Tên đối phương (bạn/nhóm) | |
| OtherId | | String | | Public | | Id của đối phương (bạn/nhóm) | |
| OtherType | | String | | Public | | Kiểu cuộc hội thoại (bạn/nhóm) | |
| ConId | | String | | Public | | Id cuộc hội thoại | |
| Users | | User[] | | Public | | Thành viên cuộc hội thoại | |
| Messages | | Message[] | | Public | | Danh sách tin nhắn | |
| Seen | | User[] | | Public | | Danh sách người dùng đã xem | |
| Phương thức | Tên | Đầu vào | | Đầu ra | | Phạm vi | | Mô tả |
| GetConversation | UserId,  ConId | | Void | | Public | | Lấy thông tin cuộc hội thoại |
| UpdateSeen | User[] | | Void | | Public | | Cập nhật danh sách đã xem |
| AddMessage | Message | | Void | | Public | | Đẩy tin nhắn vào danh sách |

### *Sơ đồ tổng quan và chi tiết tầng truy xuất dữ liệu:*

* Sơ đồ tổng quan tầng truy xuất dữ liệu:

Diagram, schematic

Description automatically generated

* Chi tiết lớp:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tên lớp | UserRouter | | | | |
| Mục đích | Xử lý các yêu cầu về thông tin người dùng | | | | |
| Thuộc tính | Không | | | | |
| Phương thức | Tên | Đầu vào | Đầu ra | Phạm vi | Mô tả |
| OnGetUserInfo | UserId | User | Public | Tìm và trả về thông tin người dùng |
| OnSearch | SearchName | User[] | Public | Tìm kiếm người dùng và trả về kết quả tìm kiếm |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tên lớp | AuthRouter | | | | |
| Mục đích | Xử lý yêu cầu đăng nhập, đăng xuất và đăng ký | | | | |
| Thuộc tính | Không | | | | |
| Phương thức | Tên | Đầu vào | Đầu ra | Phạm vi | Mô tả |
| OnLogin | Username,  UserId | String | Public | Đăng nhập người dùng vào hệ thống |
| OnLogout | UserId | Void | Public | Đăng xuất người dùng khỏi hệ thống |
| OnRegister | Username,  Password,  Email,  Gender,  DateOfBirth | Void | Public | Đăng ký tài khoản mới |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tên lớp | FriendRouter | | | | |
| Mục đích | Xử lý yêu cầu về yêu cầu kết bạn, hủy kết bạn, lấy danh sách bạn bè, yêu cầu | | | | |
| Thuộc tính | Không | | | | |
| Phương thức | Tên | Đầu vào | Đầu ra | Phạm vi | Mô tả |
| OnSendRequest | Username,  UserId,  OtherName,  OtherId | Bool | Public | Gửi yêu cầu kết bạn giữa các người dùng |
| OnCancelRequest | Username,  UserId,  OtherName,  OtherId | Bool | Public | Hủy yêu cầu kết bạn đã gửi |
| OnAcceptRequest | Username,  UserId,  OtherName  OtherId | Bool | Public | Chấp nhận yêu cầu kết bạn đã nhận |
| OnRejectRequest | Username,  UserId,  OtherName  OtherId | Bool | Public | Từ chối yêu cầu kết bạn đã nhận |
| OnUnfriend | Username,  UserId,  OtherName  OtherId | Bool | Public | Hủy kết bạn giữa các người dùng |
| OnGetFriends | UserId | User[] | Public | Lấy danh sách bạn bè |
| OnFriendsRequest | UserId | User[] | Public | Lấy danh sách yêu cầu |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tên lớp | GroupRouter | | | | |
| Mục đích | Xử lý yêu cầu về nhóm | | | | |
| Thuộc tính | Không | | | | |
| Phương thức | Tên | Đầu vào | Đầu ra | Phạm vi | Mô tả |
| OnCreateGroup | Username,  UserId,  GroupName | Bool | Public | Tạo nhóm |
| OnAddToGroup | Username,  UserId,  OtherName,  OtherId,  ConId,  GroupName | Bool | Public | Thêm bạn vào nhóm |
| OnLeave | Username,  UserId | Bool | Public | Chấp nhận yêu cầu kết bạn đã nhận |
| OnGetGroups | UserId | Group[] | Public | Lấy danh sách nhóm |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tên lớp | MessageRouter | | | | |
| Mục đích | Xử lý yêu cầu mở cuộc hội thoại | | | | |
| Thuộc tính | Không | | | | |
| Phương thức | Tên | Đầu vào | Đầu ra | Phạm vi | Mô tả |
| OnGetConversation | UserId,  ConId | Conversation | Public | Mở cuộc hội thoại |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Tên lớp | Socket.io | | | | |
| Mục đích | Xử lý yêu cầu thời gian thực | | | | |
| Thuộc tính | Không | | | | |
| Phương thức | Tên | Đầu vào | Đầu ra | Phạm vi | Mô tả |
| OnConnection | Socket | Void | Public | Kết nối từ client |
| OnInit | UserId | Void | Public | Thông báo online, Tham gia các phòng chat |
| OnOnline | UserId | Void | Public | Thông báo online tới bạn bè |
| OnOnlineToo | FromId,  ToId | Void | Public | Phản hồi thông báo online |
| OnJoinConversation | ConId | Void | Public | Tham gia phòng hội thoại |
| OnSeen | ConId,  UserId,  Username | Void | Public | Cập nhật danh sách đã xem cuộc hội thoại tới toàn bộ thành viên |
| OnMessage | ConId,  Message | Void | Public | Nhận tin nhắn, lưu trữ và chuyển tiếp |
| OnFileMessage | ConId,  Message,  File | Void | Public | Nhận tin nhắn, lưu trữ tin nhắn và tệp, chuyển tin nhắn |
| OnDisconnect | Không | Void | Public | Thông báo offline tới bạn bè |
| OnCall | FromId,  ToId | Void | Public | Chuyển tiếp lời gọi tới bên nhân |
| OnAnswer | FromId,  ToId | Void | Public | Chuyển tiếp lời hồi đáp đến bên nhận |
| OnReject | FromId,  ToId | Void | Public | Chuyển tiếp lời từ chối đến bên nhận |
| OnCandidate | FromId,  ToId | Void | Public | Chuyển tiếp ICE Candidate đến các bên |

### *Thiết kế tầng lưu trữ dữ liệu:*

A screenshot of a computer

Description automatically generated

## Kết quả:

* Chức năng “Đăng ký”:

Graphical user interface

Description automatically generated

* Chức năng “Đăng nhập”:

A screenshot of a computer

Description automatically generated

* Chức năng “Tìm kiếm”:

Graphical user interface, text, application

Description automatically generated

* Chức năng “Gửi yêu cầu kết bạn”:

A screenshot of a computer

Description automatically generated

* Chức năng “Hủy yêu cầu kết bạn”:

A screenshot of a computer

Description automatically generated

* Chức năng “Chấp nhận và từ chối yêu cầu kết bạn”:

A screenshot of a computer

Description automatically generated

* Chức năng “Gửi tin nhắn”:

A screenshot of a computer

Description automatically generated

* Chức năng “Gọi video”:

Graphical user interface, application, Teams

Description automatically generated

* Chức năng “Hủy kết bạn”:

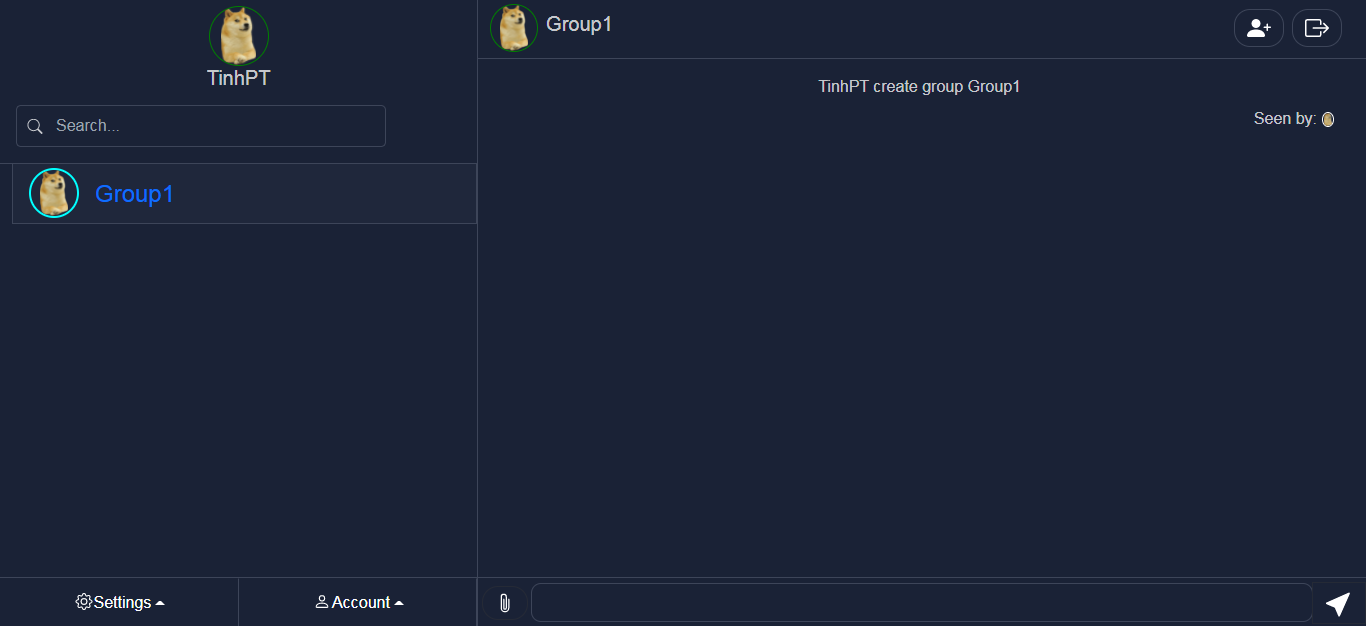
A screenshot of a computer

Description automatically generated

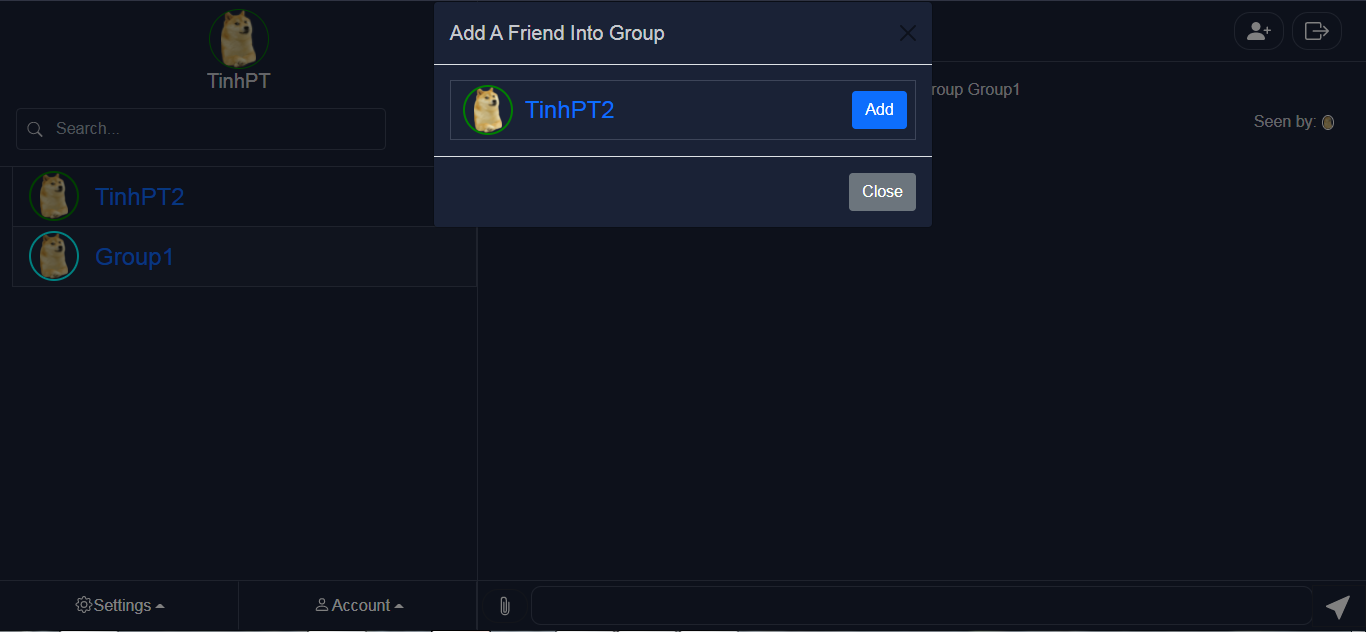
* Chức năng “Tạo nhóm”:

A screenshot of a computer

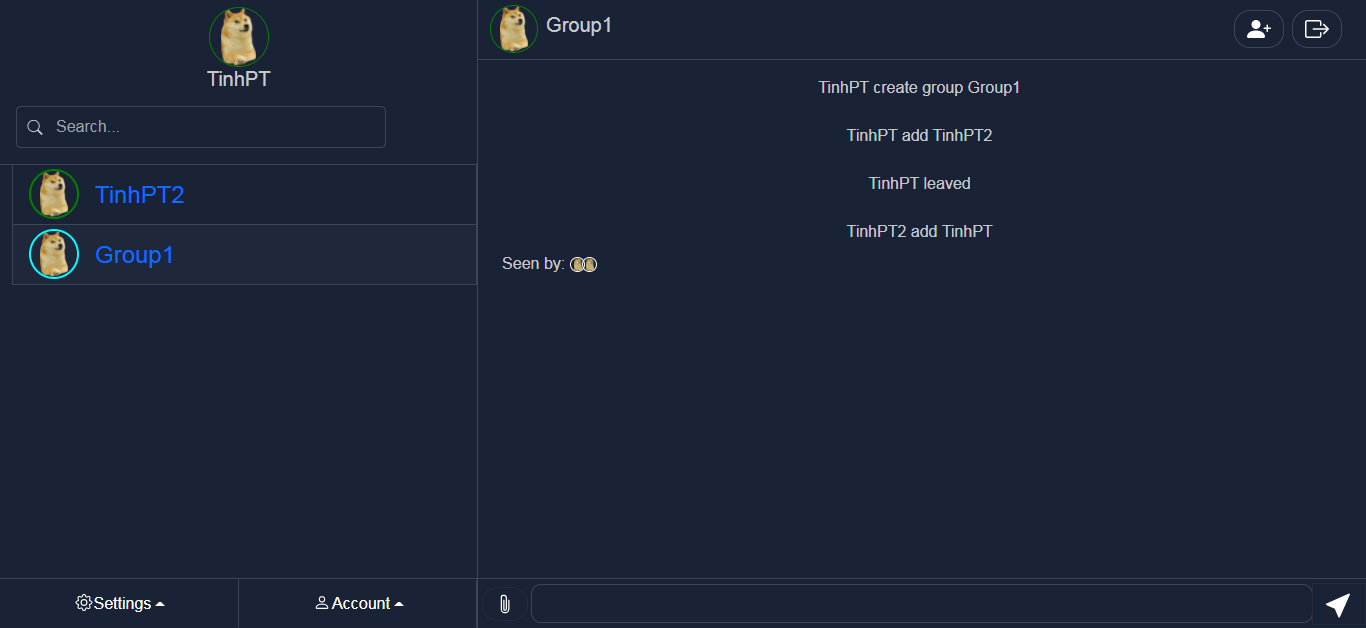
Description automatically generated



* Chức năng “Thêm bạn vào nhóm”:



* Chức năng “Rời nhóm”:



Kết luận và hướng phát triển

Ứng dụng nhắn tin trên nền tảng Web cung cấp những chức năng cơ bản, cho phép người dùng có thể kết bạn, nhắn tin và gọi video. Qua môn học, em đã làm quen và ứng dụng các công nghệ như Node.js để xây dựng ứng dụng phía Server, Vue.js xây dựng front-end, WebRTC truyền thông thời gian thực, Socket.io gửi thông điệp thời gian thực và cơ sở dữ liệu MongoDB NoSQL cũng như làm quen với quy trình phát triển một hệ thống phần mềm. Với ứng dụng nhắn tin, hướng phát triển có thể là trở thành ứng dụng họp mặt online, tương tự như Microsoft Team, Zoom, Facebook Messenger,….

Tài liệu tham khảo

[1] Beginning Node.js – Basarat Syed – 2014

[2] Fullstack Vue 3: The Complete Guide to Vue.js - Hassan Djirdeh – 2021

[3] Learning WebRTC – Dan Ristic – 2015