BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG  
KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN TỐT NGHIỆP**

**XÂY DỰNG ỨNG DỤNG PHÒNG HỌP TRỰC TUYẾN VÀ NHẮN TIN VỚI BẠN BÈ**

**Giảng viên hướng dẫn: Ths. Bùi Chí Thành**

**Sinh viên thực hiện: Nguyễn Anh Hải**

**Mã số sinh viên: 59130550**

Khánh Hòa - 2021

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG  
KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN TỐT NGHIỆP**

**XÂY DỰNG ỨNG DỤNG PHÒNG HỌP TRỰC TUYẾN VÀ NHẮN TIN VỚI BẠN BÈ**

Giảng viên hướng dẫn: Ths. Bùi Chí Thành

Sinh viên thực hiện: Nguyễn Anh Hải

Mã số sinh viên: 59130550

Khánh Hòa, tháng 7/2021

LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu của tôi và được sự hướng dẫn khoa học của Ths.Bùi Chí Thành. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những nội dung trong phần khảo sát, phân tích, nhận xét và đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo. Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình. Trường Đại học Nha Trang không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

Nha Trang, ngày 01 tháng 7 năm 2021

Sinh viên

Nguyễn Anh Hải

LỜI CẢM ƠN

Để hoàn thành đồ án tốt nghiệp này trước tiên em xin gửi đến các quý thầy, cô giáo trường Đại học Nha Trang lời cảm ơn chân thành và sâu sắc nhất.

Đặc biệt, em xin gửi đến thầy Ths Bùi Chí Thành – người đã tận tình hướng dẫn, giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình để giúp cho em hoàn thành đồ án tốt nghiệp này lời cảm ơn chân thành.

Em xin cảm ơn sự giúp đỡ của ban lãnh đạo, các anh chị và toàn thể nhân viên công ty 9thWonder Việt Nam đã tạo điều kiện cho em tìm hiểu về môi trường làm việc thực tế và tìm hiểu thêm những kiến thức mới trong quá trình thực tập và làm đồ án thực tập.

Vì kiến thức bản thân còn hạn chế, trong quá trình thực tập, hoàn thiện đồ án tốt nghiệp này em không tránh khỏi những sai sót, kính mong nhận được những ý kiến đóng góp từ quý thầy cô để em có thể có bổ sung và nâng cao kiến thức của mình góp phần phục vụ tốt hơn cho công tác của bản thân sau này.

Sau cùng, em xin kính chúc quý thầy cô trong khoa Công nghệ Thông tin và   
ban giám hiệu trường Đại học Nha Trang thật dồi dào sức khỏe để tiếp tục truyền đạt kiến thức cho thế hệ mai sau. Em xin chân thành cảm ơn.

Nha Trang, ngày 01 tháng 7 năm 2021

Sinh viên

Nguyễn Anh Hải

**MỤC LỤC**

[LỜI CAM ĐOAN i](#_Toc75607405)

[LỜI CẢM ƠN ii](#_Toc75607406)

[DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ viii](#_Toc75607407)

[DANH MỤC BẢNG BIỂU xi](#_Toc75607408)

[DANH MỤC CÁC TỪ VIẾT TẮT xiii](#_Toc75607409)

[PHẦN MỞ ĐẦU 1](#_Toc75607410)

[CHƯƠNG 1. TỔNG QUAN VỀ VẤN ĐỀ NGHIÊN CỨU 2](#_Toc75607411)

[1.1 Lý do chọn đề tài 2](#_Toc75607412)

[1.2 Xác định yêu cầu của đề tài 2](#_Toc75607413)

[1.3 Mục tiêu của đề tài 8](#_Toc75607414)

[1.4 Phương pháp nghiên cứu 8](#_Toc75607415)

[1.5 Ý nghĩa khoa học và thực tiễn 8](#_Toc75607416)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 9](#_Toc75607417)

[2.1 Tổng quan về NodeJs 9](#_Toc75607418)

[2.1.1 Tổng quan 9](#_Toc75607419)

[2.1.2 NPM 11](#_Toc75607420)

[2.2 Tổng quan về ExpressJS 11](#_Toc75607421)

[2.2.1 Tổng quan 12](#_Toc75607422)

[2.2.2 Các thành phần cơ bản 13](#_Toc75607423)

[2.3 Tổng quan về MongoDB 17](#_Toc75607424)

[2.3.1 Tổng quan về NoSQL 17](#_Toc75607425)

[2.3.2 Tổng quan về MongoDB 19](#_Toc75607426)

[2.3.3 Tổng quan về Mongoose 23](#_Toc75607427)

[2.4 Tổng quan về giao thức Websocket 25](#_Toc75607428)

[2.5 Tổng quan về Socket.io 28](#_Toc75607429)

[2.5.1 Tổng quan 28](#_Toc75607430)

[2.5.2 Phiên bản Socket (phía máy chủ) 29](#_Toc75607431)

[2.5.3 Phiên bản Socket (phía máy khách) 31](#_Toc75607432)

[2.6 Tổng quan về mạng ngang hàng (peer to peer network - P2P) 38](#_Toc75607433)

[2.7 Tổng quan về WebRTC (Web real time communication) 39](#_Toc75607434)

[2.7.1 Tổng quan 39](#_Toc75607435)

[2.7.2 Thư viện simple-peer 40](#_Toc75607436)

[2.8 Pug 42](#_Toc75607437)

[2.8.1 Tổng quan về template engine 42](#_Toc75607438)

[2.8.2 Tổng quan về Pug template 42](#_Toc75607439)

[2.9 SASS/SCSS 45](#_Toc75607440)

[2.9.1 Tổng quan 45](#_Toc75607441)

[2.9.2 Cú pháp 45](#_Toc75607442)

[2.10 Tổng quan về Cloudinary 47](#_Toc75607443)

[2.10.1 Tổng quan 47](#_Toc75607444)

[2.10.2 Các API chính 47](#_Toc75607445)

[CHƯƠNG 3. PHÂN TÍCH HỆ THỐNG 49](#_Toc75607446)

[3.1 Sơ đồ use case 49](#_Toc75607447)

[3.1.1 Nhóm use case đăng nhập 49](#_Toc75607448)

[3.1.2 Nhóm use case phòng họp 49](#_Toc75607449)

[3.1.3 Nhóm use case trò chuyện với bạn bè 49](#_Toc75607450)

[3.2 Bộ mô tả use case 50](#_Toc75607451)

[3.2.1 Use case đăng nhập 50](#_Toc75607452)

[3.2.2 Use case đăng ký tài khoản 51](#_Toc75607453)

[3.2.3 Use case quên mật khẩu 53](#_Toc75607454)

[3.2.4 Use case tạo phòng họp 54](#_Toc75607455)

[3.2.5 Use case tham gia vào phòng họp 55](#_Toc75607456)

[3.2.6 Use case quản lý phòng họp 56](#_Toc75607457)

[3.2.7 Use case giao tiếp trong phòng họp 59](#_Toc75607458)

[3.2.8 Use case rời phòng họp 61](#_Toc75607459)

[3.2.9 Use case kết thúc phiên họp 61](#_Toc75607460)

[3.2.10 Use case nhắn tin với bạn bè 62](#_Toc75607461)

[3.2.11 Use case tìm kiếm bạn bè 64](#_Toc75607462)

[3.2.12 Use case quản lý bạn bè 65](#_Toc75607463)

[3.2.13 Use case cập nhật thông tin cá nhân 67](#_Toc75607464)

[3.3 Sơ đồ hoạt động 69](#_Toc75607465)

[3.3.1 Use case đăng ký tài khoản 69](#_Toc75607466)

[3.3.2 Use case xác nhận tài khoản 69](#_Toc75607467)

[3.3.3 Use case đăng nhập 70](#_Toc75607468)

[3.3.4 Use case đăng nhập bằng Facebook 70](#_Toc75607469)

[3.3.5 Use case quên mật khẩu 71](#_Toc75607470)

[3.3.6 Use case tạo phòng họp 71](#_Toc75607471)

[3.3.7 Use case tham gia phòng họp 71](#_Toc75607472)

[3.3.8 Use case nhắn tin trong phòng phòng họp 72](#_Toc75607473)

[3.3.9 Use case bật camera 72](#_Toc75607474)

[3.3.10 Use case Chia sẻ màn hình 73](#_Toc75607475)

[3.3.11 Use case quay màn hình 73](#_Toc75607476)

[3.3.12 Use case quay tạo văn bản 73](#_Toc75607477)

[3.3.13 Use case giơ tay 74](#_Toc75607478)

[3.3.14 Use case khóa phòng họp 74](#_Toc75607479)

[3.3.15 Use case cấm chat 74](#_Toc75607480)

[3.3.16 Use case đuổi thành viên khỏi phòng 75](#_Toc75607481)

[3.3.17 Use case lưu tin nhắn 75](#_Toc75607482)

[3.3.18 Use case lưu danh sách thành viên trong phòng 76](#_Toc75607483)

[3.3.19 Use case rời phòng họp 76](#_Toc75607484)

[3.3.20 Use case kết thúc phòng họp 76](#_Toc75607485)

[3.3.21 Use case gửi lời mời kết bạn 77](#_Toc75607486)

[3.3.22 Use case chấp nhận lời mời kết bạn 77](#_Toc75607487)

[3.3.23 Use case hủy kết bạn 78](#_Toc75607488)

[3.3.24 Use case tìm kiếm bạn bè 78](#_Toc75607489)

[3.3.25 Use case nhắn tin với bạn bè 79](#_Toc75607490)

[3.3.26 Use case gọi video 79](#_Toc75607491)

[3.3.27 Use case nhắn tin bằng chụp ảnh 80](#_Toc75607492)

[3.3.28 Use case nhắn tin bằng trợ lý ảo 80](#_Toc75607493)

[3.3.29 Use case cập nhật thông tin cá nhân 81](#_Toc75607494)

[3.4 Sơ đồ tuần tự 81](#_Toc75607495)

[3.4.1 Use case đăng ký tài khoản 81](#_Toc75607496)

[3.4.2 Use case đăng nhập 83](#_Toc75607497)

[3.4.3 Use case đăng nhập bằng Facebook 83](#_Toc75607498)

[3.4.4 Use case quên mật khẩu 84](#_Toc75607499)

[3.4.5 Use case tạo phòng họp 86](#_Toc75607500)

[3.4.6 Use case tham gia phòng họp 86](#_Toc75607501)

[3.4.7 Use case nhắn tin trong phòng phòng họp 87](#_Toc75607502)

[3.4.8 Use case bật camera 88](#_Toc75607503)

[3.4.9 Use case chia sẻ màn hình 89](#_Toc75607504)

[3.4.10 Use case quay màn hình 90](#_Toc75607505)

[3.4.11 Use case tạo văn bản 91](#_Toc75607506)

[3.4.12 Use case giơ tay 92](#_Toc75607507)

[3.4.13 Use case khóa phòng họp 93](#_Toc75607508)

[3.4.14 Use case cấm nhắn tin 94](#_Toc75607509)

[3.4.15 Use case đuổi thành viên khỏi phòng 95](#_Toc75607510)

[3.4.16 Use case lưu tin nhắn 95](#_Toc75607511)

[3.4.17 Use case lưu danh sách thành viên trong phòng 96](#_Toc75607512)

[3.4.18 Use case rời phòng họp 96](#_Toc75607513)

[3.4.19 Use case kết thúc phòng họp 97](#_Toc75607514)

[3.4.20 Use case gửi lời mời kết bạn 97](#_Toc75607515)

[3.4.21 Use case chấp nhận lời mời kết bạn 98](#_Toc75607516)

[3.4.22 Use case hủy kết bạn 99](#_Toc75607517)

[3.4.23 Use case tìm kiếm bạn bè 99](#_Toc75607518)

[3.4.24 Use case nhắn tin với bạn bè 100](#_Toc75607519)

[3.4.25 Use case gọi video 101](#_Toc75607520)

[3.4.26 Use case nhắn tin bằng chụp ảnh 103](#_Toc75607521)

[3.4.27 Use case nhắn tin bằng trợ lý ảo 104](#_Toc75607522)

[3.4.28 Use case nhắn tin cập nhật thông tin 105](#_Toc75607523)

[CHƯƠNG 4. THIẾT KẾ HỆ THỐNG 108](#_Toc75607524)

[4.1 Lược đồ ODM 108](#_Toc75607525)

[4.2 Sơ đồ màn hình 115](#_Toc75607526)

[4.2.1 Sơ đồ màn hình 115](#_Toc75607527)

[4.2.2 Mô tả màn hình 115](#_Toc75607528)

[CHƯƠNG 5. CÀI ĐẶT HỆ THỐNG VÀ KẾT QUẢ ĐẠT ĐƯỢC 130](#_Toc75607529)

[5.1 Nhóm giao diện đăng nhập 130](#_Toc75607530)

[5.2 Nhóm giao diện phòng họp 132](#_Toc75607531)

[5.3 Nhóm giao diện nhắn tin với bạn bè 140](#_Toc75607532)

[CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 147](#_Toc75607533)

[6.1 Kết luận 147](#_Toc75607534)

[6.1.1 Kết quả đạt được 147](#_Toc75607535)

[6.1.2 Hạn chế tồn tại 147](#_Toc75607536)

[6.2 Hướng phát triển 147](#_Toc75607537)

[DANH MỤC TÀI LIỆU THAM KHẢO 148](#_Toc75607538)

DANH MỤC CÁC HÌNH VẼ, ĐỒ THỊ

[Hình 2.1 Quá trình thực thi các tác vụ không đồng bộ trong hệ thống Nodejs 10](#_Toc75607339)

[Hình 2.2 Sơ đồ mô tả về luồng dữ liệu chính được triển khai khi xử lý một yêu cầu / phản hồi HTTP trong Express 13](#_Toc75607340)

[Hình 2.3 Middleware (phần trung gian) 15](#_Toc75607341)

[Hình 2.4 Chi phí trên mỗi MB dữ liệu theo thời gian 18](#_Toc75607342)

[Hình 2.5 MongoDB collection 20](#_Toc75607343)

[Hình 2.6 MongoDB document 21](#_Toc75607344)

[Hình 2.7 Thêm document trong MongoDB 22](#_Toc75607345)

[Hình 2.8 Thêm document trong MongoDB 22](#_Toc75607346)

[Hình 2.9 Thêm document trong MongoDB 23](#_Toc75607347)

[Hình 2.10 Xóa document trong MongoDB 23](#_Toc75607348)

[Hình 2.11 Mongoose Object Data Modeling 23](#_Toc75607349)

[Hình 2.12 Phương thức bắt tay và truyền dữ liệu của giao thức WebSocket 27](#_Toc75607350)

[Hình 2.13 Thành phần của Socket.io 28](#_Toc75607351)

[Hình 2.14 Không gian tên Socket.io 29](#_Toc75607352)

[Hình 2.15 Vòng đời của Socket.io phía máy khách 33](#_Toc75607353)

[Hình 2.16 Sự kiện quảng bá toàn máy khách trong Socket.io 37](#_Toc75607354)

[Hình 2.17 Sự kiện quảng bá trong Socket.io 37](#_Toc75607355)

[Hình 2.18 Room trong Socket.io 37](#_Toc75607356)

[Hình 2.19 Sự kiện quảng bá của room trong socket.io 38](#_Toc75607357)

[Hình 2.20 Mạng ngang hàng 38](#_Toc75607358)

[Hình 2.21 Mạng lưới kết nối nhiều hơn hai đồng đẳng trong simple-peer 41](#_Toc75607359)

[Hình 2.22 Cấu trúc của một template engine 42](#_Toc75607360)

[Hình 3.1 Nhóm use case đăng nhập 49](#_Toc75607361)

[Hình 3.2 Nhóm use case phòng họp 49](#_Toc75607362)

[Hình 3.3 Nhóm use case trò chuyện với bạn bè 50](#_Toc75607363)

[Hình 4.1 Lược đồ ODM 108](#_Toc75607364)

[Hình 4.2 Sơ đồ màn hình nhắn tin với bạn bè 115](#_Toc75607365)

[Hình 4.3 Sơ đồ màn hình phòng họp 115](#_Toc75607366)

[Hình 5.1 Giao diện đăng nhập 130](#_Toc75607367)

[Hình 5.2 Giao diện đăng ký tài khoản 130](#_Toc75607368)

[Hình 5.3 Giao diện email xác nhận đăng ký tài khoản 131](#_Toc75607369)

[Hình 5.4 Giao diện đăng nhập bằng Google 131](#_Toc75607370)

[Hình 5.5 Giao diện quên mật khẩu bước 1 131](#_Toc75607371)

[Hình 5.6 Giao diện email xác nhận tài khoản quên mật khẩu 132](#_Toc75607372)

[Hình 5.7 Giao diện quên mật khẩu bước 2 132](#_Toc75607373)

[Hình 5.8 Giao diện tạo phòng họp 133](#_Toc75607374)

[Hình 5.9 Giao diện tham gia phòng họp 133](#_Toc75607375)

[Hình 5.10 Giao diện tham gia phòng chờ 134](#_Toc75607376)

[Hình 5.11 Giao diện phòng họp cho chủ phòng 134](#_Toc75607377)

[Hình 5.12 Giao diện phòng họp cho người tham gia 135](#_Toc75607378)

[Hình 5.13 Giao diện quản lý phòng 135](#_Toc75607379)

[Hình 5.14 Giao diện thành viên phòng họp 136](#_Toc75607380)

[Hình 5.15 Giao diện thông tin phòng 136](#_Toc75607381)

[Hình 5.16 Giao diện nhắn tin trong phòng họp 137](#_Toc75607382)

[Hình 5.17 Giao diện sử dụng microphone, camera 137](#_Toc75607383)

[Hình 5.18 Giao diện chia sẻ màn hình của người chia sẻ 138](#_Toc75607384)

[Hình 5.19 Giao diện màn hình chia sẻ màn hình 138](#_Toc75607385)

[Hình 5.20 Giao diện rời phòng họp của chủ phòng 139](#_Toc75607386)

[Hình 5.21 Giao diện rời phòng họp của người tham gia 139](#_Toc75607387)

[Hình 5.22 Kết quả tệp xuất tin nhắn trong phòng họp 140](#_Toc75607388)

[Hình 5.23 Kết quả tệp xuất thông tin cuộc họp 140](#_Toc75607389)

[Hình 5.24 Kết quả tệp xuất thông tin thành viên cuộc họp 140](#_Toc75607390)

[Hình 5.25 Giao diện nhắn tin với bạn bè 140](#_Toc75607391)

[Hình 5.26 Giao diện nhắn tin với bạn bè loại họp thoại 141](#_Toc75607392)

[Hình 5.27 Giao diện có cuộc gọi đến 141](#_Toc75607393)

[Hình 5.28 Giao diện gọi video 142](#_Toc75607394)

[Hình 5.29 Giao diện tìm kiếm người dùng 142](#_Toc75607395)

[Hình 5.30 Giao diện kết quả tìm kiếm người dùng 143](#_Toc75607396)

[Hình 5.31 Giao diện hộp thoại thông báo 143](#_Toc75607397)

[Hình 5.32 Giao diện hộp thoại mở rộng 144](#_Toc75607398)

[Hình 5.33 Giao diện trang cá nhân 144](#_Toc75607399)

[Hình 5.34 Giao diện trang cá nhân người dùng 145](#_Toc75607400)

[Hình 5.35 Giao diện cài đặt chung 145](#_Toc75607401)

[Hình 5.36 Giao diện cài đặt phòng họp tĩnh 145](#_Toc75607402)

[Hình 5.37 Giao diện cài đặt bảo mật 146](#_Toc75607403)

[Hình 5.38 Giao diện soạn thảo văn bản 146](#_Toc75607404)

**DANH MỤC BẢNG BIỂU**

[Bảng 2.1 Mối quan hệ của các thuật thuật ngữ RDBMS với MongoDB 22](#_Toc75607297)

[Bảng 2.2 Danh sách các lý do Socket máy chủ ngắt kết nối có thể xảy ra 31](#_Toc75607298)

[Bảng 2.3 Danh sách các lý do Socket máy chủ ngắt kết nối có thể xảy ra 34](#_Toc75607299)

[Bảng 4.1 Member Collection 109](#_Toc75607300)

[Bảng 4.2 Text Collection 111](#_Toc75607301)

[Bảng 4.3 Notification Collection 111](#_Toc75607302)

[Bảng 4.4 GroupMessage Collection 112](#_Toc75607303)

[Bảng 4.5 Message Collection 112](#_Toc75607304)

[Bảng 4.6 Room Collection 113](#_Toc75607305)

[Bảng 4.7 User Collection 114](#_Toc75607306)

[Bảng 4.8 Mô tả đối tượng trên màn hình đăng nhập 115](#_Toc75607307)

[Bảng 4.9 Danh sách biến cố và xử lý tương ứng trên màn hình đăng nhập 116](#_Toc75607308)

[Bảng 4.10 Mô tả đối tượng trên màn hình đăng ký tài khoản 116](#_Toc75607309)

[Bảng 4.11 Danh sách biến cố và xử lý tương ứng trên màn hình đăng ký tài khoản 117](#_Toc75607310)

[Bảng 4.12 Mô tả đối tượng trên màn hình quên mật khẩu bước 1 117](#_Toc75607311)

[Bảng 4.13 Mô tả đối tượng trên màn hình quên mật khẩu bước 2 117](#_Toc75607312)

[Bảng 4.14 Danh sách biến cố và xử lý tương ứng trên màn hình quên mật khẩu bước 1 118](#_Toc75607313)

[Bảng 4.15 Danh sách biến cố và xử lý tương ứng trên màn hình quên mật khẩu bước 2 118](#_Toc75607314)

[Bảng 4.16 Mô tả đối tượng trên màn hình nhắn tin với bạn bè 118](#_Toc75607315)

[Bảng 4.17 Danh sách biến cố và xử lý tương ứng trên màn hình nhắn tin với bạn bè 119](#_Toc75607316)

[Bảng 4.18 Mô tả đối tượng trên màn hình gọi audio 120](#_Toc75607317)

[Bảng 4.19 Danh sách biến cố và xử lý tương ứng trên màn hình gọi audio 120](#_Toc75607318)

[Bảng 4.20 Mô tả đối tượng trên màn hình gọi video 120](#_Toc75607319)

[Bảng 4.21 Danh sách biến cố và xử lý tương ứng trên màn hình gọi video 120](#_Toc75607320)

[Bảng 4.22 Mô tả đối tượng trên màn hình trang cá nhân 121](#_Toc75607321)

[Bảng 4.23 Danh sách biến cố và xử lý tương ứng trên màn hình trang cá nhân 122](#_Toc75607322)

[Bảng 4.24 Mô tả đối tượng trên màn hình tìm kiếm người dùng 122](#_Toc75607323)

[Bảng 4.25 Danh sách biến cố và xử lý tương ứng trên màn hình tìm kiếm người dùng 122](#_Toc75607324)

[Bảng 4.26 Mô tả đối tượng trên màn hình cài đặt 123](#_Toc75607325)

[Bảng 4.27 Danh sách biến cố và xử lý tương ứng trên màn hình cài đặt 123](#_Toc75607326)

[Bảng 4.28 Mô tả đối tượng trên màn hình soạn thảo văn bản 124](#_Toc75607327)

[Bảng 4.29 Danh sách biến cố và xử lý tương ứng trên màn hình soạn thảo văn bản 124](#_Toc75607328)

[Bảng 4.30 Mô tả đối tượng trên màn hình tạo phòng họp 124](#_Toc75607329)

[Bảng 4.31 Danh sách biến cố và xử lý tương ứng trên màn hình tạo phòng họp 124](#_Toc75607330)

[Bảng 4.32 Mô tả đối tượng trên màn hình tham gia phòng họp 125](#_Toc75607331)

[Bảng 4.33 Danh sách biến cố và xử lý tương ứng trên màn hình tham gia phòng họp 125](#_Toc75607332)

[Bảng 4.34 Mô tả đối tượng trên màn hình phòng chờ 125](#_Toc75607333)

[Bảng 4.35 Danh sách biến cố và xử lý tương ứng trên màn hình phòng chờ 125](#_Toc75607334)

[Bảng 4.36 Mô tả đối tượng trên màn hình phòng họp cho chủ phòng 125](#_Toc75607335)

[Bảng 4.37 Danh sách biến cố và xử lý tương ứng trên màn hình phòng họp cho chủ phòng 126](#_Toc75607336)

[Bảng 4.38 Mô tả đối tượng trên màn hình phòng họp 128](#_Toc75607337)

[Bảng 4.39 Danh sách biến cố và xử lý tương ứng trên màn hình phòng họp 129](#_Toc75607338)

DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |
| --- | --- |
| **Ký hiệu** | **Thuật ngữ** |
| JSON | JavaScript Object Notation |
| BSON | Binary JSON |
| CSDL | Cơ sở dữ liệu. |
| DBMS | Database Management System |
| APP | Application |
| SQL | Structured Query Language |
| NPM | Node Package Manager |
| HTTP | Hypertext Transfer Protocol |
| HTML | HyperText Markup Language |
| CSS | Cascading Style Sheets |
| URL | Uniform Resource Locator |
| ODM | Object Data Modeling |
| ORM | Object Relation Modeling |
| RDBMS | Relational Database Management System |
| CRUD | Create Read Update Delete |
| TCP | Transmission Control Protocol |
| IETF | Internet Engineering Task Force |
| API | Application Programming Interface |
| W3C | World Wide Web Consortium |
| WebRTC | Web Real-Time Communication |

PHẦN MỞ ĐẦU

Hiện nay công nghệ hiện đại đang ngày càng phát triển nhanh chóng trên mọi lĩnh vực trên thế giới. Trong đó nhu cầu giao tiếp, trao đổi thông tin ngày càng tăng. Cùng với sự phát triển của khoa học công nghệ, các cuộc họp trực tuyến thay vì tổ chức các cuộc họp truyền thống đang diễn ra ngày càng nhiều, nhất là trong giai đoạn thế giới đang ở trong giai đoạn dịch bệnh hoành hành trên khắp thế giới, việc giao tiếp truyền thống gặp nhiều khó khăn.

Việc giao tiếp trực tuyến giúp ích rất nhiều cho sự phát triển của nhân loại. Nó giúp tiết kiệm thời gian, tiện lợi, bỏ qua rào cản địa lý của các đối tác giao tiếp, và có thể giao tiếp vào bất kỳ thời gian và địa điểm nào.

Đề tài trò chuyện trực tuyến một đề tài không phải là mới, hiện tại trên thị trường cũng có rất nhiều các phần mềm trò chuyện trực tuyến với nhiều chức năng và chuyên nghiệp hơn. Chỉ cần gõ tìm kiếm trên Google về các phần mềm trò chuyện trực tuyến sẽ cho ra rất nhiều kết quả. Tuy nhiên, em muốn đi sâu tìm hiểu các kiến thức, công nghệ liên quan đến lĩnh vực này để tự mình tạo ra một phần mềm cho riêng mình, trước mắt là giúp bản thân nâng cao kinh nghiệm lập trình và sau đó có thể giúp đỡ cho trường nếu có thể. Em rất mong nhận được sự góp ý của quý thầy cô.

Em xin chân thành cảm ơn.

# TỔNG QUAN VỀ VẤN ĐỀ NGHIÊN CỨU

## Lý do chọn đề tài

Trong xã hội phát triển, nhu cầu giao tiếp của con người tăng lên rất nhanh, các cuộc họp diễn ra ngày càng nhiều. Đặc biệt trong những năm gần đây, tình hình dịch bệnh Covid-19 đang diễn ra hết sức phức tạp, nó tàn phá nền kinh tế thế giới, con người thì hạn chế tiếp xúc với nhau, khả năng giao tiếp trực tiếp bị hạn chế, việc dạy và học gặp rất nhiều khó khăn, việc tổ chức các cuộc họp trực tiếp gặp nhiều bất lợi. Trước tình hình thực tế cùng với sự phát triển của khoa học công nghệ, xã hội chuyển sang phương thức giao tiếp trực tuyến, các trường học chuyển sang phương thức giảng dạy trực tuyến, các công ty chuyển sang phương thức làm việc tại nhà, giao tiếp và tổ chức các cuộc họp online. Do đó các ứng dụng giao tiếp trực tuyến phát triển mạnh mẽ. Ngày càng có nhiều ứng dụng được phát triển nhằm đảm bảo nhu cầu giao tiếp của xã hội. Có thể kể đến như Zoom, Google Meet, Teams, Facebook... Chính vì những lý do trên, tôi quyết định tìm hiểu và xây dựng một ứng dụng giao tiếp trực tuyến thông qua Internet trên nền tảng web.

## Xác định yêu cầu của đề tài

* Đăng ký và đăng nhập.
* Đăng ký tài khoản.
* Đăng ký tài khoản bằng email, tên và mật khẩu.
* Yêu cầu xác nhận tài khoản bằng email đăng ký.
* Có thể đăng ký tài khoản thông qua bên thứ ba gồm Facebook và Google.
* Thông báo về trạng thái đăng ký tài khoản (thành công hay thất bại).
* Đăng nhập tài khoản.
* Đăng nhập tài khoản bằng email và mật khẩu.
* Không thể đăng nhập vào ứng dụng khi chưa xác minh tài khoản bằng email trừ trường hợp đăng nhập ứng dụng thông qua bên thứ 3.
* Có thể đăng đăng nhập tài khoản bằng các phương thức đăng nhập khác thông qua bên thứ ba gồm Facebook và Google.
* Thông báo về trạng thái đăng nhập tài khoản (thành công hay thất bại).
* Đăng xuất khỏi ứng dụng.
* Quên mật khẩu:
* Cho phép người dùng đổi lại mật khẩu khi họ quên mật khẩu bằng email đã đăng ký tài khoản.
* Chỉ tiến hành cho phép đổi mật khẩu khi người dùng xác nhận đổi mật khẩu trong email đăng ký.
* Thông báo về trạng thái đổi mật khẩu (thành công hay thất bại).
* Tạo phòng họp trực tuyến.
* Có thể sử dụng chức năng này khi chưa đăng ký tài khoản.
* Tạo phòng họp.
* Cần có mã phòng, mật khẩu vào phòng (sinh ra ngẫu nhiên) và tên người tạo hiển thị trong phòng.
* Người tạo phòng họp cũng chính là chủ phòng, chủ phòng có đầy đủ các chức năng của các thành viên bình thường và có thêm chức năng quản lý phòng.
* Chức năng quản lý phòng của chủ phòng:
* Mở phòng: Mở phòng họp, bất kì ai cũng có thể vào phòng họp.
* Khóa phòng: Khóa phòng họp, không thể vào phòng khi phòng đang ở trạng thái khóa.
* Đặt trạng thái phòng chờ: Các thành viên tham gia vào phòng khi phòng đang ở trạng trạng thái phòng chờ sẽ được đưa vào phòng chờ, các yêu cầu tham gia sẽ được gửi đến chủ phòng. Chủ phòng có quyền cho phép thành viên vào phòng hay không.
* Cấm trò chuyện: Cấm mọi thành viên trong phòng trò chuyện bằng tin nhắn, trừ mình (chủ phòng).
* Cấm bật microphone: Cấm mọi thành viên trong phòng bật microphone, nếu có thành viên đang bật microphone thì tắt microphone của họ, trừ mình (chủ phòng).
* Cấm bật camera: Cấm mọi thành viên trong phòng bật camera, nếu có thành viên đang bật camera thì tắt camera của họ, trừ mình (chủ phòng).
* Cấm chia sẻ màn hình: Cấm mọi thành viên trong phòng chia sẻ màn hình, nếu có thành viên đang chia sẻ màn hình thì tắt màn hình chia sẻ của họ, trừ mình (chủ phòng).
* Cấm quay màn hình: Cấm mọi thành viên trong phòng quay màn hình, nếu có thành viên đang quay màn hình thì kết thúc phiên quay màn hình của họ, trừ mình (chủ phòng).
* Đuổi thành viên ra khỏi phòng: Chỉ định một thành viên nào đó rời khỏi phòng mà không cần sự chấp nhận của họ.
* Cấm giao tiếp của thành viên trong phòng: Cấm sự giao tiếp của một thành viên vào đó trong phòng. Các nội dung cấm bao gồm: nhắn tin, bật microphone, bật camera, chia sẻ màn hình, quay màn hình. Nếu họ đang dùng các chức năng đó thì dừng lại.
* Kết thúc phòng họp: Kết thúc phiên họp cho tất cả mọi thành viên trong phòng.
* Cho phép chủ phòng xuất báo cáo về thông tin cuộc họp:
* Thời gian bắt đầu cuộc họp (được tính khi chủ phòng vào phòng).
* Thời gian kết thúc phòng họp (được tính khi chủ phòng rời phòng).
* Số người tham gia vào phòng họp.
* Danh sách thông tin của các thành viên tham gia vào cuộc họp.
* Tên thành viên.
* Thời điểm vào phòng.
* Thời điểm rời phòng.
* Thời gian ở trong phòng.
* Tham gia vào phòng họp.
* Tham gia vào phòng đã được tạo bằng mã phòng, mật khẩu và tên của người tham gia hiển thị trong phòng.
* Nếu phòng bị khóa thì thông báo không thể vào phòng.
* Nếu phòng đang ở trạng thái phòng chờ thì chuyển người tham gia vào phòng chờ, có sự cho phép của chủ phòng thì người tham gia mới được vào phòng. Người tham gia có thể rời khỏi phòng chờ.
* Khi mới vào phòng thì mặc định microphone và camera sẽ bị tắt.
* Các phương thức giao tiếp trong phòng:
* Thông qua tin nhắn: Nhắn tin bằng bàn phím, cho phép gửi biểu tượng cảm xúc. Tin nhắn sẽ được gửi đến các thành viên khác trong phòng. Chỉ nhắn tin được khi chủ phòng cho phép.
* Thông qua gửi tệp tin: Chia sẻ tệp tin có trong thiết bị, cho phép gửi nhiều tệp cùng một lúc (tối đa 5 tệp tin) và kích thước mỗi tập tin không quá 1MB. Tệp tin sẽ được gửi đến các thành viên khác trong phòng. Chỉ được gửi tệp tin khi chủ phòng cho phép nhắn tin.
* Thông qua sử dụng microphone: Truyền âm thanh lấy được từ microphone đến các thành viên khác trong phòng. Chỉ được bật microphone khi được chủ phòng cho phép sử dụng microphone.
* Thông qua sử dụng camera: Truyền hình ành lấy được từ camera đến các thành viên khác trong phòng. Chỉ được bật camera khi được chủ phòng cho phép sử dụng camera.
* Thông qua chia sẻ màn hình: Cho phép chia sẻ màn hình làm việc của mình cho các thành viên trong phòng. Cùng một thời điểm chỉ có một thành viên được chia sẻ màn hình và chỉ được chia sẻ màn hình khi được chủ phòng cho phép.
* Thông qua giơ tay: Các thành viên trong phòng có thể giơ tay trình bày ý kiến. Khi giơ tay sẽ có thông báo cho các thành viên khác và người giơ tay sẽ được đưa lên đầu danh sách thành viên.
* Cho phép quay lại nội dung cuộc họp: Mọi thành viên trong phòng đều có thể quay lại nội dung cuộc họp khi được chủ phòng cho phép.
* Cho phép thảo luận trên bộ soạn thảo văn bản online.
* Tạo văn bản và gửi đường dẫn của văn bản đến các thành viên trong phòng. Khi kết thúc cuộc họp, văn bản không bị xóa đi. Danh sách văn bản được hiển thị trong trang cá nhân của tác giả.
* Chỉ có tác giả của bài viết mới có quyền chỉnh sửa, những người còn lại chỉ được xem.
* Cho phép xuất nội dung trò chuyện của cuộc họp: Nội dung kết xuất bao gồm: tên thành viên viết nhắn tin, thời điểm nhắn tin và nội dung tin nhắn.
* Cho phép rời khỏi phòng họp: Mọi thành viên có thể rời khỏi phòng họp bất bì lúc nào. Khi thành viên đó vào lại phòng thì là một thành viên khác chứ không phải là người trước.
* Có thông báo về tình trạng phòng họp (có người tham gia vào phòng, có người rời khỏi phòng, có người giơ tay, các thay đổi quản lý phòng của chủ phòng).
* Hiển thị danh sách các thành viên trong phòng, thứ tự thành viên được sắp xếp theo vai trò (bạn, chủ phòng) và tên.
* Hiển thị thông tin phòng họp và có thể chia sẻ cho người khác vào phòng.
* Cho phép tạo phòng họp tĩnh.
* Khi tạo tài khoản, mỗi user sẽ có một phòng họp tĩnh, gồm số phòng và mật khẩu phòng.
* Số phòng sẽ không thay đổi.
* Mật khẩu phòng có thể thay đổi.
* User có thể sử dụng phòng họp tĩnh hoặc không. Nếu user sử dụng phòng họp tĩnh thì khi tạo phòng sẽ vào phòng họp tĩnh của mình, còn nếu không sử dụng phòng họp tĩnh thì khi tạo phòng sẽ tạo một phòng họp ngẫu nhiên theo phiên làm việc (sẽ bị xóa khi kết thúc cuộc họp).
* Trò chuyện trực tuyến với bạn bè.
* Tìm kiếm bạn bè.
* Thông qua tên của bạn bè.
* Thông qua số điện thoại của bạn bè.
* Thông qua địa chỉ của bạn bè.
* Các hành động liên quan đến bạn bè.
* Gửi lời mời kết bạn: Gửi lời mời kết bạn đến bạn bè, phải chờ sự xác nhận của bạn bè mới có thể trò chuyện với nhau.
* Chấp nhận lời mời kết bạn.
* Không chấp nhận lời mời kết bạn.
* Xóa bạn bè: Xóa bạn bè từ một phía, không cần sự chấp nhận từ bên còn lại. Mọi tin nhắn giữa hai bên sẽ bị xóa.
* Gửi thông báo về các hành động của bạn bè (chấp nhận hay không chấp nhận lời mời kết bạn).
* Hiển thị trạng thái của bạn bè (đang hoạt động hay đang không hoạt động).
* Hiển thị danh sách bạn bè cùng với tin nhắn mới nhất với họ, danh sách được sắp xếp theo giảm dần theo thời gian của tin nhắn mới nhất.
* Trò chuyện với bạn bè.
* Chỉ trò chuyện được với bạn bè của mình.
* Trò chuyện thông qua tin nhắn, có thể gửi biểu tượng cảm xúc.
* Cho phép gửi tệp đính kèm, cho phép gửi nhiều tệp đính kèm cùng một lúc (tối đa 5 tệp) và kích thước mỗi tệp không vượt quá 1MB. Cho phép xem trước tệp đính kèm nếu tệp đính kèm có định dạng là tệp tin ảnh, video, audio.
* Cho phép gọi audio: Không thể gọi cho bạn bè khi họ không trực tuyến. Người được gọi có quyền nhận hay không nhận cuộc gọi. Trong khi gọi cho phép tắt bật microphone và không cho phép sử dụng camera.
* Cho phép gọi video: Không thể gọi cho bạn bè khi họ không trực tuyến. Người được gọi có quyền nhận hay không nhận cuộc gọi. Trong khi gọi cho phép tắt bật microphone và camera.
* Gửi đoạn ghi âm: Lấy đoạn âm thanh từ microphone và gửi cho bạn bè
* Gửi ảnh chụp trực tiếp: Sử dụng camera chụp một ảnh trực tiếp và gửi cho bạn bè.
* Cho phép xóa tin nhắn: Xóa tin nhắn cho cả hai bên, tin nhắn không hoàn toàn bị xóa đi, nhưng không thể hoàn tác.
* Chỉnh sửa tin nhắn: Cho phép chỉnh sửa các tin nhắn dạng văn bản
* Sử dụng trợ lý ảo để giao hỗ trợ trò chuyện bằng giọng nói.
* Nhận diện giọng nói và chuyển sang dạng chữ và gửi đoạn tin nhắn đó cho bạn bè.
* Các hình thức xác nhận gửi tin nhắn bằng trợ lý ảo.
* Tự động gửi.
* Xác nhận bằng hộp thoại.
* Xác nhận bằng chỉ thị lời nói.
* Cho phép chạy nền chức năng này hoặc khi kích hoạt khi có hành động của user.
* Cho phép chuyển đổi ngôn ngữ của trợ lý giữa tiếng Anh và tiếng Việt.
* Cho phép tắt, bật chức năng này.
* Chỉnh sửa thông tin cá nhân:
* Chỉnh sửa ảnh đại diện, họ tên, số điện thoại, ngày tháng năm sinh, giới tính, địa chỉ.
* Đổi mật khẩu.
* Thay đổi tên trên thanh địa chỉ của trình duyệt.
* Thêm, xóa, sửa các văn bản soạn thảo.
* Xem, xóa, thay đổi trạng thái các thông báo được gửi đến người dùng.
* Có âm thanh cho một số hành động, các xử lý của ứng dụng.
* Đưa ứng dụng lên Heroku hosting.

## Mục tiêu của đề tài

* Xây dựng được ứng dụng phục vụ nhu cầu giao tiếp trực tuyến, đáp ứng được các yêu cầu đã đề ra của đề tài.
* Giúp bản thân có thêm nhiều kiến thức mới về công nghệ để phục vụ cho công việc sau này.

## Phương pháp nghiên cứu

* Nghiên cứu, tìm hiểu tổng quan về các công nghệ liên quan đến ứng dụng: Ngôn ngữ lập trình JavaScript, NodeJs, ExpressJs, MongoDB, WebSocket, Socket.io, Peer to Peer Network, WebRTC, Pug template, SCSS, Cloudinary, cách đưa ứng dụng lên Hosting, sử dụng phần mềm Visual Studio Code để phát triển ứng dụng.
* Phân tích ứng dụng theo phương pháp hướng đối tượng.
* Thiết kế ứng dụng theo mô hình NoSQL.
* Cài đặt và thử nghiệm ứng dụng.
* Dựa vào kiến thức học được, kinh nghiệm trong các đề tài trước để khắc phục các hạn chế còn tồn tại, phát triển thêm chức năng, tối ưu hóa hệ thống.

## Ý nghĩa khoa học và thực tiễn

Việc xây dựng ứng dụng phòng họp trực tuyến và nhắn tin với bạn bè sẽ hỗ trợ trong việc giao tiếp trực tuyến, nhất là trong thời điểm hiện tại khi tình hình dịch bệnh đang diễn ra rất phức tạp, con người thực hiện giãn cách, hạn chế tiếp xúc, học tập và làm việc tại nhà thì việc giao tiếp trực tuyến là không thể thiếu. Giúp việc giao tiếp diễn ra một cách dễ dàng mà không cần phải gặp mặt trực tiếp. Tiết kiệm thời gian tổ chức các cuộc họp ngắn và có thể thực hiện giao tiếp vào bất kỳ thời gian nào.

# CƠ SỞ LÝ THUYẾT

1. **Tổng quan về NodeJs**
2. **Tổng quan**

Node.js là một môi trường runtime (là môi trường mà một chương trình hoặc ứng dụng được thực thi. Đó là cơ sở hạ tầng phần cứng và phần mềm hỗ trợ việc chạy một cơ sở mã cụ thể trong thời gian thực) dành cho JavaScript đa nền tảng và mã nguồn mở. Nó là một công cụ phổ biến cho hầu hết mọi loại dự án.

Node.js chạy công cụ JavaScript V8, cốt lõi của Google Chrome, bên ngoài trình duyệt. Điều này cho phép Node.js hoạt động rất hiệu quả.

Ứng dụng Node.js chạy trong một luồng duy nhất mà không cần tạo luồng mới cho mọi yêu cầu. Nodejs cung cấp một tập hợp các nguyên thủy I/O không đồng bộ trong thư viện chuẩn của nó.

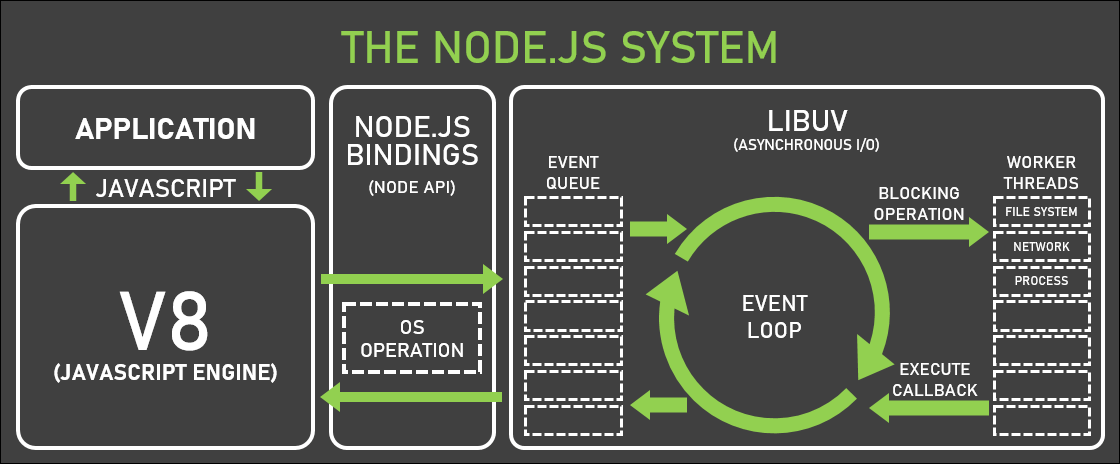
Khi Nodejs thực hiện một thao tác I/O, chẳng hạn như đọc từ mạng, truy cập cơ sở dữ liệu hoặc hệ thống tệp, thay vì chặn luồng và lãng phí chu kỳ chờ đợi CPU, Node.js sẽ tiếp tục hoạt động khi phản hồi trở lại. Điều này cho phép Node.js xử lý hàng nghìn kết nối đồng thời với một máy chủ duy nhất mà không gây ra gánh nặng quản lý đồng thời luồng.

Node.js có một lợi thế là người phát triển giao diện người dùng viết JavaScript cho trình duyệt hiện có thể viết mã phía máy chủ mà không cần phải học một ngôn ngữ hoàn toàn khác.

Trong Node.js, các tiêu chuẩn ECMAScript mới có thể được sử dụng mà không gặp vấn đề gì.

Việc thực thi mã JavaScript phụ thuộc vào công cụ Chrome V8. Nó hoạt động như một cốt lõi của NodeJs và nó có thể được nhúng vào ứng dụng C++. V8 Engine nhận đầu vào là một chuỗi trong JavaScript, xử lý nó và sau đó in đầu ra ra stdout.

Quá trình thực thi các tác vụ không đồng bộ trong hệ thống Nodejs.



Hình . Quá trình thực thi các tác vụ không đồng bộ trong hệ thống Nodejs

Toàn bộ quy trình của sơ đồ có thể được chia thành các bước như sau:

* Mã JavaScript chạy trên V8 engine, khi có một tác vụ không đồng bộ để cần thực thi.
* Một yêu cầu được gửi bởi thư viện libuv Node.js tới hệ điều hành để thực hiện tác vụ.
* Tác vụ này sau đó được đặt vào một hàng đợi các sự kiện (event queue) sẽ hoàn thành nhiệm vụ sau đó trong tương lai.
* Vòng lặp sự kiện (event loop) kiểm tra hàng đợi sự kiện một cách nhất quán để xem liệu bất kỳ tác vụ nào trong hàng đợi đã được hoàn thành hay chưa.
* Nếu không có bất kỳ tác vụ đồng bộ chưa hoàn thành nào được tìm thấy trong hàng đợi sự kiện, vòng lặp sự kiện sẽ trả về để tiếp tục thực hiện lệnh gọi lại của tác vụ không đồng bộ.

Từ quan điểm phát triển máy chủ web, Node có một số lợi ích:

* Hiệu suất cao. Nodejs được thiết kế để tối ưu hóa thông lượng và khả năng mở rộng trong các ứng dụng web và là giải pháp tốt cho nhiều vấn đề phát triển web phổ biến (ví dụ: ứng dụng web thời gian thực).
* Mã được viết bằng JavaScript thuần túy, có nghĩa là các nhà phát triển dành ít thời gian hơn để dịch chuyển ngữ cảnh giữa các ngôn ngữ khi họ đang viết cả mã phía máy khách và phía máy chủ.
* JavaScript là một ngôn ngữ lập trình tương đối mới và được hưởng lợi từ những cải tiến trong thiết kế ngôn ngữ khi so sánh với các ngôn ngữ máy chủ web truyền thống khác (ví dụ: Python, PHP...). Nhiều ngôn ngữ mới và phổ biến khác được biên dịch/chuyển đổi thành JavaScript như TypeScript, CoffeeScript, ClojureScript, Scala, LiveScript….

1. **NPM**

NPM (trình quản lý gói) cung cấp quyền truy cập vào hàng trăm nghìn gói có thể sử dụng lại.

Nó có một hệ sinh thái bên thứ ba rất tích cực và cộng đồng nhà phát triển, với rất nhiều người sẵn sàng trợ giúp.

* Các gói (module) cơ bản của Nodejs
* http (xây dựng một máy chủ HTTP)

const http = require('http')

const port = process.env.PORT || 3000

const server = http.createServer((req, res) => {

  res.statusCode = 200

  res.setHeader('Content-Type', 'text/html')

  res.end('<h1>Hello, World!</h1>')

})

server.listen(port, () => {

  console.log(`Server running at port ${port}`)

})

* fs (xử lý file)
* Đọc file

const fs = require('fs')

fs.readFile('/Users/joe/test.txt', 'utf8' , (err, data) => {

  if (err) {

    console.error(err)

    return

  }

  console.log(data)

})

* Ghi file

const fs = require('fs')

const content = 'Some content!'

fs.writeFile('/Users/joe/test.txt', content, err => {

  if (err) {

    console.error(err)

    return

  }

})

* Xóa file

const fs = require('fs')

fs.unlink('path/file.txt', (err) => {

  if (err) throw err

  console.log('path/file.txt was deleted')

})

1. **Tổng quan về ExpressJS**
2. **Tổng quan**

Express js là một Framework phổ biến, nhỏ, nhưng linh hoạt được xây dựng trên nền tảng của Nodejs. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. Và là thư viện cơ bản cho một số khung web Node phổ biến khác. Nó cung cấp các cơ chế để:

* Viết trình xử lý cho các yêu cầu với các phương thức HTTP khác nhau tại các đường dẫn URL (routes) khác nhau.
* Tích hợp với các công cụ kết xuất (template engines) để tạo phản hồi bằng cách chèn dữ liệu vào các mẫu (template).
* Thêm phần mềm trung gian (middleware) xử lý yêu cầu bổ sung tại bất kỳ thời điểm nào trong đường dẫn xử lý yêu cầu.

Mặc dù bản thân Express khá tối giản, nhưng các nhà phát triển đã tạo ra các gói phần mềm trung gian tương thích để giải quyết hầu hết mọi vấn đề phát triển web. Có các thư viện để làm việc với cookie, phiên làm việc (session), thông tin đăng nhập của người dùng, tham số URL, dữ liệu POST, bảo mật...

Trong một trang web theo hướng dữ liệu truyền thống, một ứng dụng web chờ các yêu cầu HTTP từ trình duyệt web (hoặc ứng dụng khách khác). Khi nhận được yêu cầu, ứng dụng sẽ tìm ra hành động cần thiết dựa trên mẫu URL và có thể là thông tin liên quan có trong dữ liệu POST hoặc dữ liệu GET. Tùy thuộc vào những gì được yêu cầu, nó có thể đọc hoặc ghi thông tin từ cơ sở dữ liệu hoặc thực hiện các tác vụ khác cần thiết để đáp ứng yêu cầu. Sau đó, ứng dụng sẽ trả về phản hồi cho trình duyệt web, thường tạo động một trang HTML để trình duyệt hiển thị bằng cách chèn dữ liệu đã truy xuất vào các trình giữ chỗ trong một mẫu HTML.

Express cung cấp các phương thức để chỉ định hàm nào được gọi cho một phương thức HTTP cụ thể như GET, POST, PUT... và mẫu URL (route) và các phương thức để chỉ định công cụ mẫu (view) nào được sử dụng, nơi mẫu các tệp được định vị và sử dụng mẫu nào để hiển thị phản hồi. Người phát triển có thể sử dụng phần mềm trung gian để thêm hỗ trợ cho cookie, phiên và người dùng, nhận tham số POST, GET... Ứng dụng có thể sử dụng bất kỳ cơ sở dữ liệu nào được Node hỗ trợ (Express không xác định bất kỳ hành vi nào liên quan đến cơ sở dữ liệu).

* Cách khởi tạo một máy chủ expressjs đơn giản

const express = require('express')

const app = express()

const port = 3000

app.get('/', (req, res) => {

  res.send('Hello World!')

})

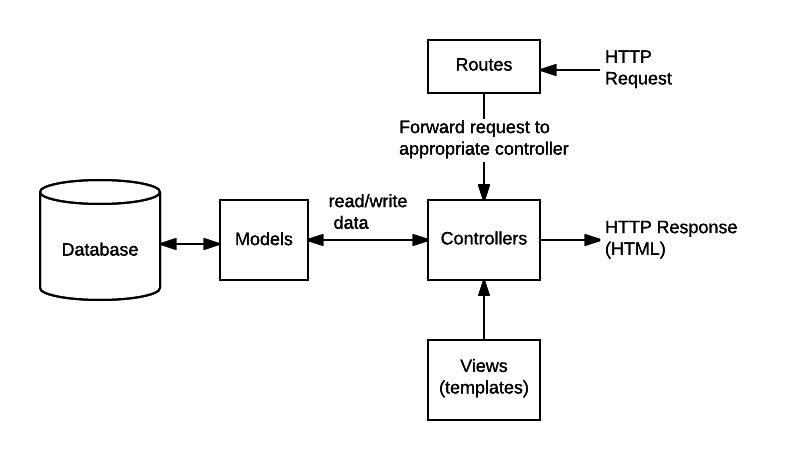
app.listen(port, () => {

  console.log(`Example app listening on port ${port}!`)

})

1. **Các thành phần cơ bản**

Sơ đồ dưới đây mô tả về luồng dữ liệu chính và những thứ cần được triển khai khi xử lý một yêu cầu / phản hồi HTTP.



Hình 2.2 Sơ đồ mô tả về luồng dữ liệu chính được triển khai khi xử lý một yêu cầu / phản hồi HTTP trong Express

* Routes có chức năng chuyển tiếp các yêu cầu được hỗ trợ (và bất kỳ thông tin nào được mã hóa trong URL yêu cầu) đến các chức năng của bộ điều khiển thích hợp.
* Controller có chức năng lấy dữ liệu được yêu cầu từ các Models, tạo trang HTML hiển thị dữ liệu và trả lại cho người dùng để xem trong trình duyệt.
* Views (templates) được bộ điều khiển sử dụng để hiển thị dữ liệu.
* Models đại diện cho dữ liệu của trang web dưới dạng các đối tượng JavaScript, sau đó được ánh xạ tới cơ sở dữ liệu bên dưới.

1. **Routes**

Routes là một phần của mã Express liên kết với các phương thức HTTP (GET, POST, PUT, DELETE...), đường dẫn / mẫu URL và một hàm được gọi để xử lý mẫu đó.

* Ví dụ định nghĩa routes

const express = require('express')

const router = express.Router()

// Home page route.

router.get('/', function (req, res) {

  res.send('Wiki home page')

})

// About page route.

router.get('/about', function (req, res) {

  res.send('About this wiki')

})

module.exports = router

* Sử dụng route

const wiki = require('./routers/wiki.js')

app.use('/wiki', wiki)

Các phương thức cơ bản của Routes (tương ứng với các phương thức của HTTP)

router.get(): phản hồi các yêu cầu HTTP GET (lấy dữ liệu từ một nguồn dữ liệu cụ thể).

router.post(): phản hồi các yêu cầu HTTP POST (gửi dữ liệu tới một server để tạo hoặc cập nhật tài nguyên trên đó).

Router.put(): phản hồi các yêu cầu HTTP PUT (gửi dữ liệu tới server để tạo hoặc cập nhật tài nguyên trên đó).

Router.delete(): phản hồi các yêu cầu HTTP DELETE (xóa tài nguyên cụ thể nào đó).

Đường dẫn routes: Các đường dẫn tuyến xác định các điểm cuối (endpoint) mà tại đó các yêu cầu có thể được thực hiện. Nó có thể là một chuỗi cụ thể (‘/about’) hay là một mẫu chuỗi (‘/ab\*cd’) để định nghĩa.

Thông số routes: Tham số routes là các phân đoạn URL được đặt tên được sử dụng để nắm bắt các giá trị tại các vị trí cụ thể trong URL. Các phân đoạn được đặt tên được bắt đầu bằng dấu hai chấm và sau đó là tên (ví dụ: ‘/:parameter’). Các giá trị thu được được lưu trữ trong đối tượng req.params bằng cách sử dụng tên tham số làm khóa (ví dụ: req.params.parameter).

* Ví dụ

app.get('/users/:userId/books/:bookId', function (req, res) {

  // truy cập userId qua: req.params.userId

  // truy cập bookId qua: req.params.bookId

  res.send(req.params)

})

1. **Controller**

Controller dùng để xử lý logic cho các routes, tách việc định nghĩa routes với xử lý bên trong (lấy dữ liệu từ cơ sở dữ liệu, tạo trang HTML hiển thị cho người dùng) giúp cho việc phát triển và bảo trì ứng dụng dễ dàng hơn.

Ví dụ

// router.js

const express = require('express')

const router = express.Router()

const controller = require(‘./controller’)

// Home page route.

router.get('/',  controller.getIndex)

// About page route.

router.get('/about', controller.getAbout)

module.exports = router

// controller.js

module.export.getIndex = (req, res) => {

  res.send('Wiki home page')

}

module.export.getAbout = (req, res) => {

  res.send('About this wiki')

}

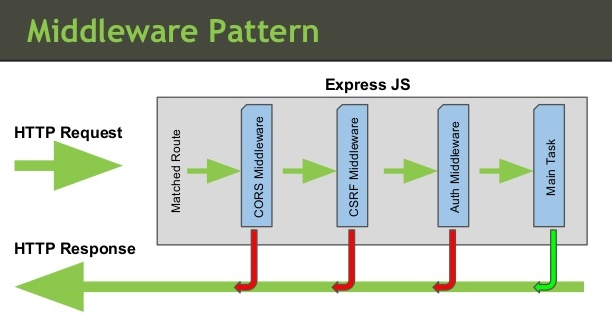
1. **Models**

Là một mô hình dữ liệu đối tượng (“ODM”) hoặc mô hình quan hệ đối tượng (“ORM”) đại diện cho dữ liệu của trang web dưới dạng các đối tượng JavaScript, sau đó được ánh xạ tới cơ sở dữ liệu bên dưới.

1. **Views**

Là nơi chứa một template engine cho phép sử dụng các tệp mẫu tĩnh trong ứng dụng. Trong thời gian chạy, các mẫu thay thế các biến trong tệp mẫu bằng các giá trị thực tế (được truyền vào từ controller) và chuyển mẫu thành tệp HTML được gửi đến máy khách. Cách tiếp cận này giúp thiết kế một trang HTML dễ dàng hơn. Trình tạo ứng dụng Express sử dụng Jade làm mặc định, nhưng nó cũng hỗ trợ một số template engine khác. Một số template engine phổ biến hoạt động với Express là Pug, Mustache và Ejs.

1. **Middleware (phần trung gian)**



Hình . Middleware (phần trung gian)

ExpressJs khi hoạt động sẽ là một loạt các hàm middleware được thực hiện liên tiếp nhau. Sau khi đã thiết lập, các request từ phía người dùng khi gửi lên, Express sẽ thực hiện lần lượt qua các hàm middleware cho đến khi trả về response cho người dùng. Các hàm này sẽ được quyền truy cập đến các đối tượng đại diện cho request, response, hàm middleware tiếp theo, và đối tượng lỗi nếu cần thiết. Một hàm middleware sau khi hoạt động xong, nếu chưa phải là cuối cùng trong chuỗi các hàm cần thực hiện, sẽ cần gọi lệnh next() để chuyển sang hàm tiếp theo, bằng không xử lý sẽ bị treo tại hàm đó.

Các chức năng mà middleware có thể thực hiện trong ExpressJS sẽ bao gồm:

* Thực hiện bất cứ đoạn code nào.
* Thay đổi các đối tượng request và response.
* Kết thúc một quá trình request-response.
* Gọi hàm middleware tiếp theo trong stack.
* Trong Express, có 5 kiểu middleware có thể sử dụng:
* Application-level middleware (middleware cấp ứng dụng)

Ví dụ khai báo một middleware không khai báo đường dẫn cụ thể, nó sẽ được thực hiện mỗi lần request.

const app = express()

app.use(function (req, res, next) {

  console.log('Time:', Date.now())

  next()

})

Ví dụ khai báo một middle không khai báo đường dẫn cụ thể, nó sẽ được thực hiện mỗi khi request đến đường dẫn /user/:id bất kể phương thức nào (GET, POST…).

app.use('/user/:id', function (req, res, next) {

  console.log('Request Type:', req.method)

  next()

})

Ví dụ khai báo một middle không khai báo đường dẫn và phương thức cụ thể, nó sẽ được thực hiện mỗi khi request đến đường dẫn /user/:id bằng phương thức GET.

app.get('/user/:id', function (req, res, next) {

  res.send('USER')

})

* Router-level middleware (middleware cấp điều hướng – router)

Các middleware này về chức năng không khác gì so với application-level middleware, tuy nhiên thay vì dùng biến app có thể gây nhầm lẫn với các thiết lập, phần router có thể không rõ ràng và khó phân biệt, ExpressJS cung cấp một đối tượng router chuyên dùng để khai báo route bằng cách gọi hàm sau:

const router = express.Router()

* Error-handling middleware (middleware xử lý lỗi)

Đây là các middleware phục vụ cho việc xử lý lỗi.

app.use(function (err, req, res, next) {

  console.error(err.stack)

  res.status(500).send('Something broke!')

})

* Built-in middleware (middleware sẵn có)

Các middleware được tích hợp trong Express.

Ví dụ: app.use(express.static('public'))

* Third-party middleware (middleware của bên thứ ba)

Sử dụng middleware của bên thứ ba để thêm chức năng cho ứng dụng Express.

Ví dụ sử dụng middleware phân tích cú pháp cookie.

const express = require('express')

const app = express()

const cookieParser = require('cookie-parser')

// load the cookie-parsing middleware

app.use(cookieParser())

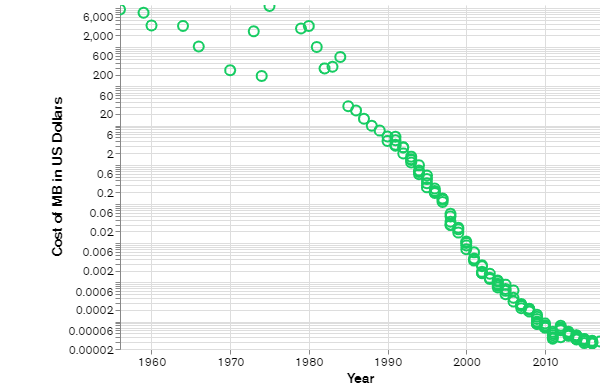
1. **Tổng quan về MongoDB**
2. **Tổng quan về NoSQL**

Cơ sở dữ liệu NoSQL (hay còn gọi là "not only SQL" hoặc “non-SQL”) là cơ sở dữ liệu lưu trữ dữ liệu ở định dạng không phải là bảng quan hệ

Một quan niệm sai lầm phổ biến là cơ sở dữ liệu NoSQL hoặc cơ sở dữ liệu không quan hệ không lưu trữ tốt dữ liệu mối quan hệ. Cơ sở dữ liệu NoSQL có thể lưu trữ dữ liệu mối quan hệ, chúng chỉ lưu trữ nó khác với cơ sở dữ liệu quan hệ. Trên thực tế, khi so sánh với cơ sở dữ liệu SQL, nhiều người thấy dữ liệu mối quan hệ mô hình hóa trong cơ sở dữ liệu NoSQL dễ dàng hơn trong cơ sở dữ liệu SQL, vì dữ liệu liên quan không cần phải được phân chia giữa các bảng.

Các mô hình dữ liệu NoSQL cho phép dữ liệu liên quan được lồng trong một cấu trúc dữ liệu duy nhất.

Cơ sở dữ liệu NoSQL xuất hiện vào cuối những năm 2000 khi chi phí lưu trữ giảm đáng kể. Các nhà phát triển (thay vì lưu trữ) đang trở thành chi phí chính của việc phát triển phần mềm, do đó, cơ sở dữ liệu NoSQL được tối ưu hóa cho năng suất của nhà phát triển.



Hình . Chi phí trên mỗi MB dữ liệu theo thời gian

Khi chi phí lưu trữ giảm nhanh chóng, số lượng ứng dụng dữ liệu cần thiết để lưu trữ và truy vấn tăng lên. Dữ liệu này có đủ hình dạng và kích thước, có cấu trúc, bán cấu trúc và đa hình, và việc xác định trước lược đồ trở nên gần như không thể. Cơ sở dữ liệu NoSQL cho phép các nhà phát triển lưu trữ một lượng lớn dữ liệu phi cấu trúc, mang lại cho chúng rất nhiều tính linh hoạt.

Điện toán đám mây cũng trở nên phổ biến và các nhà phát triển bắt đầu sử dụng các đám mây công cộng để lưu trữ các ứng dụng và dữ liệu của họ. Họ muốn khả năng phân phối dữ liệu trên nhiều máy chủ và khu vực để làm cho ứng dụng của họ có khả năng phục hồi, mở rộng quy mô và định vị địa lý thông minh cho dữ liệu của họ. Một số cơ sở dữ liệu NoSQL như MongoDB cung cấp các khả năng này.

* Các loại NoSQL DBMS
* Key-value (khóa-giá trị)

Là một loại cơ sở dữ liệu đơn giản trong đó mỗi mục chứa các khóa và giá trị. Một giá trị thường chỉ có thể được truy xuất bằng cách tham chiếu đến khóa của nó, vì vậy việc học cách truy vấn một cặp khóa-giá trị cụ thể thường đơn giản. Cơ sở dữ liệu khóa-giá trị rất phù hợp cho các trường hợp sử dụng mà cần lưu trữ lượng lớn dữ liệu nhưng không cần thực hiện các truy vấn phức tạp để truy xuất dữ liệu đó. Các trường hợp sử dụng phổ biến bao gồm lưu trữ tùy chọn người dùng hoặc bộ nhớ đệm. Redis và DynanoDB là các cơ sở dữ liệu khóa-giá trị phổ biến.

* Document (tài liệu)

Lưu trữ dữ liệu trong các tài liệu tương tự như các đối tượng JSON. Mỗi tài liệu chứa các cặp trường và giá trị. Các giá trị thường có thể là nhiều loại bao gồm những thứ như chuỗi, số, boolean, mảng hoặc đối tượng và cấu trúc của chúng. Do có nhiều loại giá trị trường và ngôn ngữ truy vấn mạnh mẽ, cơ sở dữ liệu tài liệu rất phù hợp cho nhiều trường hợp sử dụng và có thể được sử dụng làm cơ sở dữ liệu mục đích chung. Họ có thể mở rộng quy mô theo chiều ngang để chứa khối lượng dữ liệu lớn. MongoDB liên tục được xếp hạng là cơ sở dữ liệu NoSQL phổ biến nhất thế giới theo DB-engine và là một ví dụ về cơ sở dữ liệu tài liệu.

Ví dụ DBMS: MongoDB, CouchDB, Elasticsearch, Solr.

* Wide column stores

Lưu trữ dữ liệu trong bảng, hàng và cột động. Wide column stores cung cấp nhiều tính linh hoạt so với cơ sở dữ liệu quan hệ vì mỗi hàng không bắt buộc phải có các cột giống nhau. Nhiều người coi các Wide column stores là cơ sở dữ liệu khóa-giá trị hai chiều. Cửa hàng cột rộng rất phù hợp khi cần lưu trữ lượng lớn dữ liệu và có thể dự đoán các mẫu truy vấn sẽ như thế nào. Wide column stores thường được sử dụng để lưu trữ dữ liệu Internet of Things và dữ liệu hồ sơ người dùng.

Ví dụ DBMS: Cassandra, HBase, Microsoft Azure Cosmos DB.

* Graph databases (biểu đồ)

Lưu trữ dữ liệu trong các nút và các cạnh. Các nút thường lưu trữ thông tin về người, địa điểm và mọi thứ trong khi các cạnh lưu trữ thông tin về mối quan hệ giữa các nút. Cơ sở dữ liệu đồ thị vượt trội trong các trường hợp sử dụng mà cần duyệt qua các mối quan hệ để tìm kiếm các mẫu như mạng xã hội, phát hiện gian lận và công cụ khuyến nghị. Neo4j và JanusGraph là những ví dụ về cơ sở dữ liệu đồ thị.

1. **Tổng quan về MongoDB**

MongoDB là một cơ sở dữ liệu định hướng tài liệu, đa nền tảng, hiệu suất cao, tính sẵn sàng cao và khả năng mở rộng dễ dàng. MongoDB hoạt động dựa trên khái niệm về bộ sưu tập và tài liệu.

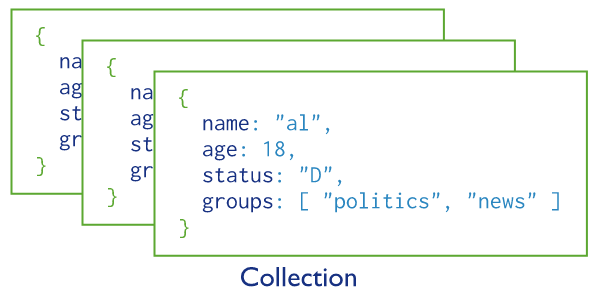
* Đặc điểm chính
* Lược đồ ít hơn: MongoDB là một cơ sở dữ liệu tài liệu trong đó một bộ sưu tập chứa các tài liệu khác nhau. Số lượng trường, nội dung và kích thước của tài liệu có thể khác nhau giữa các tài liệu.
* Cấu trúc của một đối tượng duy nhất là rõ ràng.
* Không có liên kết phức tạp.
* Khả năng truy vấn sâu. MongoDB hỗ trợ các truy vấn động trên tài liệu bằng ngôn ngữ truy vấn dựa trên tài liệu gần như mạnh mẽ như SQL.
* Khả năng điều chỉnh.
* Dễ mở rộng quy mô.
* Sử dụng bộ nhớ trong để lưu bộ làm việc (cửa sổ), cho phép truy cập dữ liệu nhanh hơn.
* Các khái niệm chính
* Database - Cơ sở dữ liệu

MongoDB lưu trữ các bản ghi dữ liệu dưới dạng tài liệu (cụ thể là tài liệu BSON) được tập hợp lại với nhau trong bộ sưu tập. Cơ sở dữ liệu lưu trữ một hoặc nhiều bộ sưu tập tài liệu. Mỗi cơ sở dữ liệu có một bộ tệp riêng trên hệ thống tệp. Một máy chủ MongoDB duy nhất thường có nhiều cơ sở dữ liệu.

Tạo cơ sở dữ liệu mới: use myDB

* Collection - Bộ sưu tập

Collection là một nhóm các tài liệu MongoDB. Nó tương đương với một bảng RDBMS. Một bộ sưu tập tồn tại trong một cơ sở dữ liệu duy nhất. Bộ sưu tập không thực thi một giản đồ. Các tài liệu trong một bộ sưu tập có thể có các trường khác nhau. Thông thường, tất cả các tài liệu trong bộ sưu tập đều có mục đích tương tự hoặc có liên quan.



Hình 2.5 MongoDB collection

Tạo bộ sưu tập:

db.createCollection(

  <name>,

  {

    capped: <boolean>,

    autoIndexId: <boolean>,

    size: <number>,

    max: <number>,

    storageEngine: <document>,

    validator: <document>,

    validationLevel: <string>,

    validationAction: <string>,

    indexOptionDefaults: <document>,

    viewOn: <string>,

    pipeline: <pipeline>,

    collation: <document>,

    writeConcern: <document>

  }

)

* Document - Tài liệu

Tài liệu là một tập hợp các cặp khóa-giá trị. Tài liệu có lược đồ động. Lược đồ động có nghĩa là các tài liệu trong cùng một bộ sưu tập không cần phải có cùng một nhóm trường hoặc cấu trúc và các trường chung trong tài liệu của một bộ sưu tập có thể chứa các loại dữ liệu khác nhau. MongoDB lưu trữ các bản ghi dữ liệu dưới dạng tài liệu BSON. BSON là bản đại diện nhị phân của tài liệu JSON, mặc dù nó chứa nhiều kiểu dữ liệu hơn JSON.



Hình 2.6 MongoDB document

Tài liệu MongoDB bao gồm các cặp trường-và-giá trị và có cấu trúc sau:

{

  field1: value1,

  field2: value2,

  ...

}

Giá trị của một trường có thể là bất kỳ kiểu dữ liệu BSON nào, bao gồm các tài liệu, mảng và mảng tài liệu khác. Ví dụ: tài liệu sau chứa các giá trị thuộc nhiều loại khác nhau.

var mydoc = {

  \_id: ObjectId("5099803df3f4948bd2f98391"),

  name: { first: "Alan", last: "Turing" },

  birth: new Date('Jun 23, 1912'),

  death: new Date('Jun 07, 1954'),

  contribs: [ "Turing machine", "Turing test", "Turingery" ],

  views : NumberLong(1250000)

}

Tên trường

Tài liệu có các ràng buộc sau đối với tên trường.

* Tên trường \_id được dành riêng để sử dụng làm khóa chính; giá trị của nó phải là duy nhất trong bộ sưu tập, là bất biến, và có thể thuộc bất kỳ kiểu nào khác ngoài mảng.
* Tên trường là chuỗi không được chứa ký tự rỗng.
* Tên trường cấp cao nhất không được bắt đầu bằng ký tự đô la ($).
* Tên các trường phải khác nhau trong một document.

BSON là một định dạng tuần tự hóa nhị phân được sử dụng để lưu trữ tài liệu và thực hiện các cuộc gọi thủ tục từ xa trong MongoDB. Một số loại BSON cơ bản: Double, String, Object, Array, ObjectId, Boolean, Date, Null, Regular Expression…

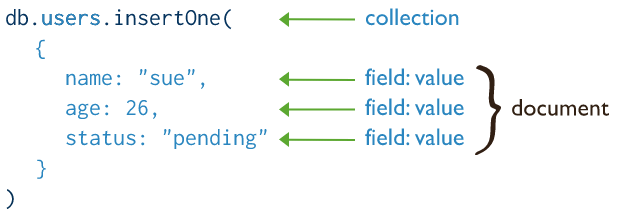
Bảng 2.1 Mối quan hệ của các thuật thuật ngữ RDBMS với MongoDB

|  |  |
| --- | --- |
| **RDBMS** | **MongoDB** |
| Database | Database |
| Table | Collection |
| Row | Document |
| Column | Field |
| Table Join | Embedded Documents |
| Primary Key | Primary Key (Default key \_id) |

* Xử lý CRUD của MongoDB
* Thêm

db.collection.insertOne()

db.collection.insertMany()



Hình 2.7 Thêm document trong MongoDB

* Truy vấn

db.collection.find()



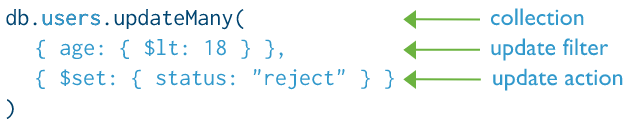
Hình 2.8 Thêm document trong MongoDB

* Sửa

db.collection.updateOne()

db.collection.updateMany()

db.collection.replaceOne()

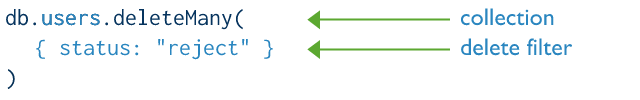


Hình 2.9 Thêm document trong MongoDB

* Xóa

db.collection.deleteOne()

db.collection.deleteMany()



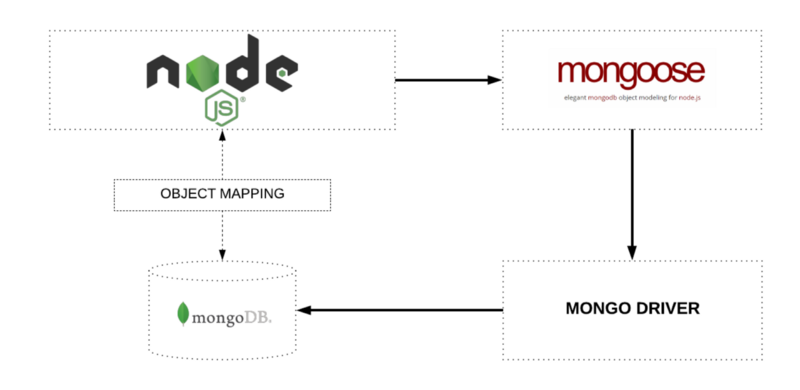
Hình 2.10 Xóa document trong MongoDB

1. **Tổng quan về Mongoose**

Có hai cách tiếp cận phổ biến để tương tác với cơ sở dữ liệu:

* Sử dụng ngôn ngữ truy vấn gốc của cơ sở dữ liệu (ví dụ: SQL).
* Sử dụng mô hình dữ liệu đối tượng (“ODM”) hoặc mô hình quan hệ đối tượng (“ORM”). ODM/ORM đại diện cho dữ liệu của trang web dưới dạng các đối tượng JavaScript, sau đó được ánh xạ tới cơ sở dữ liệu bên dưới. Một số ORM được gắn với một cơ sở dữ liệu cụ thể, trong khi những ORM khác cung cấp một chương trình phụ trợ cơ sở dữ liệu bất khả tri (database-agnostic).

Mongoose là một thư viện tạo Object Data Modeling (ODM) cho MongoDB và Node.js. Nó quản lý các mối quan hệ giữa dữ liệu, cung cấp xác nhận lược đồ và được sử dụng để dịch giữa các đối tượng trong mã và biểu diễn của các đối tượng đó trong MongoDB.



Hình 2.11 Mongoose Object Data Modeling

Ánh xạ đối tượng giữa Node và MongoDB được quản lý thông qua Mongoose. Mô hình Mongoose là một trình bao bọc trên lược đồ Mongoose. Một lược đồ Mongoose xác định cấu trúc của tài liệu, giá trị mặc định, trình xác thực... mô hình Mongoose cung cấp giao diện cho cơ sở dữ liệu để tạo, truy vấn, cập nhật, xóa bản ghi...

Tạo mô hình Mongoose.

let mongoose = require('mongoose')

let userSchema = new mongoose.Schema({

  name: String,

  email: String,

  gender: {

    type: Boolean,

    default: true

  }

})

module.exports = mongoose.model('User', userSchema)

Các kiểu dữ liệu được sử dụng trong lược đồ

* Array
* Boolean
* Buffer
* Date
* Mixed (A generic / flexible data type)
* Number
* ObjectId
* String

Xuất một mô hình cần gọi hàm tạo mô hình trên cá thể Mongoose và truyền cho nó tên của bộ sưu tập và một tham chiếu đến định nghĩa lược đồ.

* Các xử lý chính
* Tạo document

const User = require('./user')

User.create({ name: 'Mongoose', email: ‘test@gmail.com’ }, (err, user) => {

  if (err) return handleError(err)

  // đã thêm thành công

})

* Truy vấn

User.find()            // find all users

  .skip(100)           // skip the first 100 items

  .limit(10)           // limit to 10 items

  .sort({name: 1}      // sort ascending by firstName

  .select({name: true} // select firstName only

  .exec()              // execute the query

  .then(docs => {

    console.log(docs)

  })

  .catch(err => {

    console.error(err)

  })

* Sửa document

User

  .findOneAndUpdate(

    {

      email: 'ada.lovelace@gmail.com'  // search query

    },

    {

      email: 'theoutlander@live.com'   // field: values to update

    },

    {

      new: true,                       // return updated doc

      runValidators: true              // validate before update

    })

  .then(doc => {

    console.log(doc)

  })

  .catch(err => {

    console.error(err)

  })

* Xóa document

User

  .findOneAndRemove({

    email: 'theoutlander@live.com'

  })

  .then(response => {

    console.log(response)

  })

  .catch(err => {

    console.error(err)

  })

1. **Tổng quan về giao thức Websocket**

Trước đây, việc tạo các ứng dụng web cần hai chiều giao tiếp giữa máy khách và máy chủ (ví dụ: nhắn tin tức thì và các ứng dụng trò chơi) đã yêu cầu lạm dụng HTTP để thăm dò ý kiến máy chủ để cập nhật trong khi gửi thông báo ngược dòng là riêng biệt cuộc gọi HTTP.

Điều này dẫn đến một loạt các vấn đề:

* Máy chủ buộc phải sử dụng một số TCP cơ bản khác nhau kết nối cho mỗi khách hàng, một kết nối để gửi thông tin đến khách hàng và một khách hàng mới cho mỗi tin nhắn đến.
* Giao thức dây có chi phí cao, với mỗi máy khách đến máy chủ thông báo có tiêu đề HTTP.
* Tập lệnh phía máy khách buộc phải duy trì ánh xạ từ kết nối đi đến kết nối đến để theo dõi trả lời.

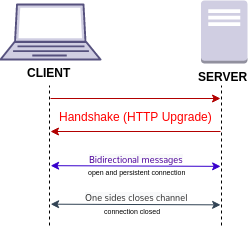
Một giải pháp đơn giản hơn sẽ là sử dụng một kết nối TCP duy nhất cho giao thông theo cả hai hướng được gọi là giao thức WebSocket, nó cung cấp một thay thế cho thăm dò HTTP để giao tiếp hai chiều từ một máy khách đến một máy chủ từ xa. Nó có thể được sử dụng cho nhiều ứng dụng web: trò chơi, mã chứng khoán, ứng dụng nhiều người dùng đồng thời chỉnh sửa, giao diện người dùng hiển thị các dịch vụ phía máy chủ trong thời gian thực....

WebSocket là một giao thức truyền thông máy tính, cung cấp các kênh truyền thông song công toàn phần (full-duplex: chế độ truyền xác định hướng của luồng tín hiệu giữa hai thiết bị được kết nối, việc truyền tín hiệu diễn ra theo hai hướng cùng một lúc và kênh được sử dụng đồng thời bởi cả hai thiết bị đã kết nối) trên một kết nối TCP duy nhất. Giao thức WebSocket đã được IETF chuẩn hóa thành RFC 6455 vào năm 2011 và API WebSocket trong Web IDL (Web interface description language) đang được W3C chuẩn hóa.

WebSocket khác với HTTP. Cả hai giao thức đều nằm ở lớp 7 trong mô hình OSI và phụ thuộc vào TCP ở tầng thứ 4. Mặc dù chúng khác nhau, RFC 6455 nói rằng WebSocket được thiết kế để hoạt động trên các cổng HTTP 443 và 80 cũng như để hỗ trợ proxy và trung gian HTTP, do đó làm cho nó tương thích với giao thức HTTP. Để đạt được tính tương thích, quá trình bắt tay WebSocket sử dụng tiêu đề nâng cấp HTTP (HTTP/1.1) để thay đổi từ giao thức HTTP sang giao thức WebSocket.

Giao thức WebSocket cho phép tương tác giữa trình duyệt web (hoặc ứng dụng khách khác) và máy chủ web với chi phí thấp hơn so với các lựa chọn thay thế bán song công như bỏ phiếu HTTP, tạo điều kiện truyền dữ liệu theo thời gian thực từ và đến máy chủ. Điều này có thể thực hiện được bằng cách cung cấp một cách tiêu chuẩn hóa để máy chủ gửi nội dung đến máy khách mà không cần khách hàng yêu cầu trước, đồng thời cho phép các thông báo được truyền qua lại trong khi vẫn giữ kết nối mở. Bằng cách này, một cuộc trò chuyện diễn ra hai chiều có thể diễn ra giữa máy khách và máy chủ. Giao tiếp thường được thực hiện qua cổng TCP số 443 (hoặc 80 trong trường hợp kết nối không an toàn), điều này có lợi cho các môi trường chặn các kết nối Internet không phải web bằng cách sử dụng tường lửa.

Giao thức có hai phần: bắt tay và truyền dữ liệu.



Hình 2.12 Phương thức bắt tay và truyền dữ liệu của giao thức WebSocket

Để thiết lập kết nối WebSocket, máy khách sẽ gửi một yêu cầu bắt tay WebSocket, máy chủ trả về phản hồi bắt tay WebSocket.

Yêu cầu của máy khách trông như sau:

GET /chat HTTP/1.1

  Host: server.example.com

  Upgrade: websocket

  Connection: Upgrade

  Sec-WebSocket-Key: dGhlIHNhbXBsZSBub25jZQ==

  Origin: http://example.com

  Sec-WebSocket-Protocol: chat, superchat

  Sec-WebSocket-Version: 13

Phản hồi của máy chủ trông như sau:

HTTP/1.1 101 Switching Protocols

  Upgrade: websocket

  Connection: Upgrade

  Sec-WebSocket-Accept: s3pPLMBiTxaQ9kYGzzhZRbK+xOo=

  Sec-WebSocket-Protocol: chat

Quá trình bắt tay bắt đầu bằng một yêu cầu / phản hồi HTTP, cho phép các máy chủ xử lý các kết nối HTTP cũng như các kết nối WebSocket trên cùng một cổng. Khi kết nối được thiết lập, giao tiếp sẽ chuyển sang giao thức nhị phân hai chiều.

Ngoài tiêu đề HTTP/1.1, máy khách gửi một tiêu đề Sec-WebSocket-Key chứa các byte ngẫu nhiên được mã hóa base64 và máy chủ trả lời bằng một hàm băm của khóa trong tiêu đề Sec-WebSocket-Accept. Điều này nhằm ngăn chặn proxy bộ nhớ đệm gửi lại cuộc hội thoại WebSocket trước đó và không cung cấp bất kỳ xác thực, quyền riêng tư hoặc tính toàn vẹn nào.

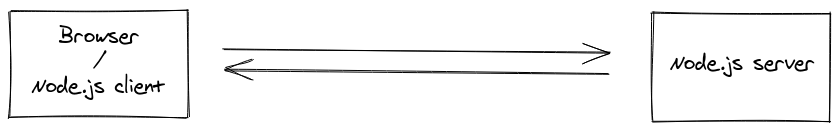
Sau khi kết nối được thiết lập, máy khách và máy chủ có thể gửi dữ liệu WebSocket hoặc khung văn bản qua lại ở chế độ song công. Dữ liệu được đóng khung tối thiểu, với một tiêu đề nhỏ theo sau là tải trọng (payload). Truyền WebSocket được mô tả là “messages”, trong đó một messages có thể được tùy chọn chia thành nhiều khung dữ liệu. Điều này có thể cho phép gửi tin nhắn khi có sẵn dữ liệu ban đầu nhưng độ dài hoàn chỉnh của tin nhắn là không xác định (nó gửi hết khung dữ liệu này đến khung dữ liệu khác cho đến khi kết thúc và đi kèm với bit FIN).

Hầu hết các trình duyệt đều hỗ trợ giao thức này, bao gồm Google Chrome, Firefox, Microsoft Edge, Internet Explorer, Safari và Opera.

1. **Tổng quan về Socket.io**
2. **Tổng quan**

Socket.io là một thư viện cho phép giao tiếp theo thời gian thực, hai chiều và dựa trên sự kiện giữa trình duyệt và máy chủ. Nó bao gồm:

* Thư viện phía máy chủ cho Node.js.
* Thư viện phía máy khách chạy trong trình duyệt (cũng có thể chạy từ Node.js).



Hình 2.13 Thành phần của Socket.io

Cách thức hoạt động.

Kênh hai chiều giữa máy chủ Socket.io (Node.js) và máy khách Socket.io (trình duyệt, Node.js hoặc ngôn ngữ lập trình khác) được thiết lập bằng kết nối WebSocket bất cứ khi nào có thể và sẽ sử dụng tính năng thăm dò ý kiến ​​dài HTTP làm dự phòng.

Cơ sở mã Socket.io được chia thành hai lớp riêng biệt:

* Hệ thống bên dưới (cấp thấp): được gọi là Engine.io, động cơ bên trong Socket.io.
* API cấp cao: chính là Socket.io.

Socket.io không phải là một triển khai của WebSocket. Mặc dù Socket.io thực sự sử dụng WebSocket như một phương tiện truyền tải khi có thể, nó sẽ thêm siêu dữ liệu bổ sung vào mỗi gói. Đó là lý do tại sao máy khách WebSocket sẽ không thể kết nối thành công với máy chủ Socket.io và máy khách Socket.io cũng sẽ không thể kết nối với máy chủ WebSocket thuần túy.

Socket.io cung cấp các tính năng bổ sung trên một đối tượng WebSocket thuần túy.

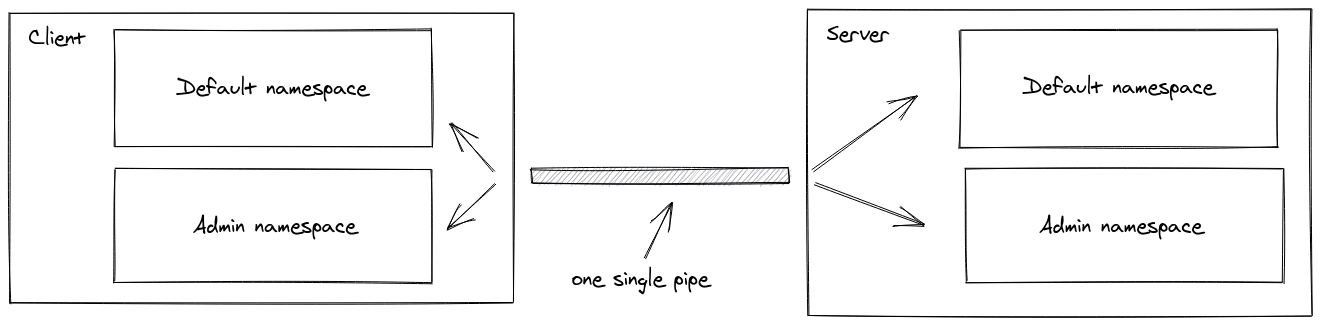
* Độ tin cậy (dự phòng cho việc thăm dò dài hạn HTTP trong trường hợp không thể thiết lập kết nối WebSocket).
* Kết nối lại tự động.
* Bộ đệm gói.

Theo mặc định, bất kỳ sự kiện nào phát ra trong khi Socket không được kết nối sẽ được lưu vào bộ đệm cho đến khi kết nối lại.

Mặc dù hữu ích trong hầu hết các trường hợp (khi độ trễ kết nối lại ngắn), nó có thể dẫn đến một lượng lớn các sự kiện khi kết nối được khôi phục.

* Sự xác nhận: Cho phép xác nhận việc truyền thông điệp từ máy khách tới máy chủ đã thực hiện hoàn thành chưa.
* Truyền phát cho tất cả máy khách hoặc một nhóm nhỏ máy khách (Room - Phòng).
* Ghép kênh (Namespace - Không gian tên).

Không gian tên là một kênh giao tiếp cho phép phân chia logic của ứng dụng qua một kết nối được chia sẻ duy nhất (còn được gọi là “ghép kênh”).



Hình 2.14 Không gian tên Socket.io

1. **Phiên bản Socket (phía máy chủ)**
2. **Đối tượng máy chủ**

Thường được gọi là io.

* Sự kiện

Đối tượng máy chủ (io) tạo ra một sự kiện duy nhất là connect (hoặc tên khác là connection). Sự kiện này được kích hoạt khi có kết nối mới.

io.on("connection", (socket) => {

  // ...

})

1. **Đối tượng soket máy chủ**

* Chức năng chính:
* Phát ra và lắng nghe các sự kiện sự kiện.
* Phát sóng quảng bá (broadcasting).
* Tham gia và rời khỏi phòng.
* Phiên bản Socket có một số thuộc tính có thể được sử dụng trong ứng dụng.

**Socket#id**

Mỗi kết nối mới được gán một mã định danh 20 ký tự ngẫu nhiên.

Giá trị nhận dạng này được đồng bộ hóa với giá trị ở phía máy khách.

// server-side

io.on("connection", (socket) => {

  console.log(socket.id) // ojIckSD2jqNzOqIrAGzL

});

// client-side

socket.on("connect", () => {

  console.log(socket.id) // ojIckSD2jqNzOqIrAGzL

});

Sau khi tạo, Socket tham gia vào phòng được xác định bằng id riêng của nó

Lưu ý: không thể ghi đè id nhận dạng này.

**Socket#handshake**

Đối tượng này chứa một số chi tiết về quá trình bắt tay xảy ra ở đầu phiên Socket.io.

{

  headers: /\* tiêu đề của yêu cầu ban đầu \*/

  query: /\* các tham số truy vấn của yêu cầu ban đầu \*/

  auth: /\* dữ liệu xác thực \*/

  time: /\* ngày tạo (dưới dạng chuỗi) \*/

  issued: /\* ngày tạo (dấu thời gian unix) \*/

  url: /\* chuỗi URL yêu cầu \*/

  address: /\* ip của máy khách \*/

  xdomain: /\* kết nối có phải là giữa nhiều miền hay không \*/

  secure: /\* kết nối có an toàn không \*/

}

**Socket#rooms**

Đây là tham chiếu đến các phòng mà soket hiện đang ở.

io.on("connection", (socket) => {

  console.log(socket.rooms) // Set { <socket.id> }

  socket.join("room1")

  console.log(socket.rooms) // Set { <socket.id>, "room1" }

})

**Socket middlewares**

Các phần mềm trung gian có thể được thêm vào socket máy chủ, chúng sẽ được gọi cho mỗi gói tin đến.

io.on("connection", (socket) => {

  socket.use(([event, ...args], next) => {

    if (isUnauthorized(event)) {

      return next(new Error("unauthorized event"));

    }

    next()

  })

  socket.on("error", (err) => {

    if (err && err.message === "unauthorized event") {

      socket.disconnect()

    }

  })

})

Phương thức next cũng có thể được gọi với một đối tượng lỗi. Trong trường hợp đó, sự kiện sẽ không đến được các trình xử lý sự kiện đã đăng ký và thay vào đó, một sự kiện lỗi sẽ được phát ra.

* Sự kiện

Ở phía máy chủ, phiên bản Socket phát ra hai sự kiện đặc biệt:

* disconnect

Sự kiện này được kích hoạt bởi đối tượng Socket khi ngắt kết nối.

Bảng 2.2 Danh sách các lý do Socket máy chủ ngắt kết nối có thể xảy ra

|  |  |
| --- | --- |
| **Lý do** | **Mô tả** |
| server namespaces disconnect | Soket đã bị ngắt kết nối cưỡng bức bởi máy chủ với socket.disconnect() |
| client namespaces disconnect | Máy khách đã ngắt kết nối theo cách thủ công bằng socket.disconnect() |
| server shutting down | Máy chủ đang tắt |
| ping timeout | Máy khách không gửi gói PING trong độ trễ pingTimeout |
| transport close | Kết nối đã bị đóng (ví dụ: người dùng bị mất kết nối hoặc mạng đã được thay đổi từ WiFi thành 4G) |
| transport error | Kết nối đã gặp lỗi |

* disconnecting

Sự kiện này tương tự như disconnect nhưng được kích hoạt sớm hơn một chút, khi bộ Socket#room chưa trống.

io.on("connection", (socket) => {

  socket.on("disconnecting", (reason) => {

    for (const room of socket.rooms) {

      if (room !== socket.id) {

        socket.to(room).emit("user has left", socket.id)

      }

    }

  })

})

Lưu ý: những sự kiện trên, cùng với connect, connect\_error, newListener và removeListener là các sự kiện đặc biệt không được sử dụng trong ứng dụng.

1. **Phiên bản Socket (phía máy khách)**

Bên cạnh việc phát ra và lắng nghe các sự kiện, phiên bản Socket có một số thuộc tính có thể được sử dụng trong ứng dụng.

**Socket#id**

Mỗi kết nối mới được gán một mã định danh 20 ký tự ngẫu nhiên.

Giá trị nhận dạng này được đồng bộ hóa với giá trị ở phía máy chủ.

// server-side

io.on("connection", (socket) => {

  console.log(socket.id) // x8WIv7-mJelg7on\_ALbx

})

// client-side

socket.on("connect", () => {

  console.log(socket.id) // x8WIv7-mJelg7on\_ALbx

})

socket.on("disconnect", () => {

  console.log(socket.id) // undefined

})

**Socket#connected**

Thuộc tính này mô tả liệu socket hiện có được kết nối với máy chủ hay không.

socket.on("connect", () => {

  console.log(socket.connected) // true

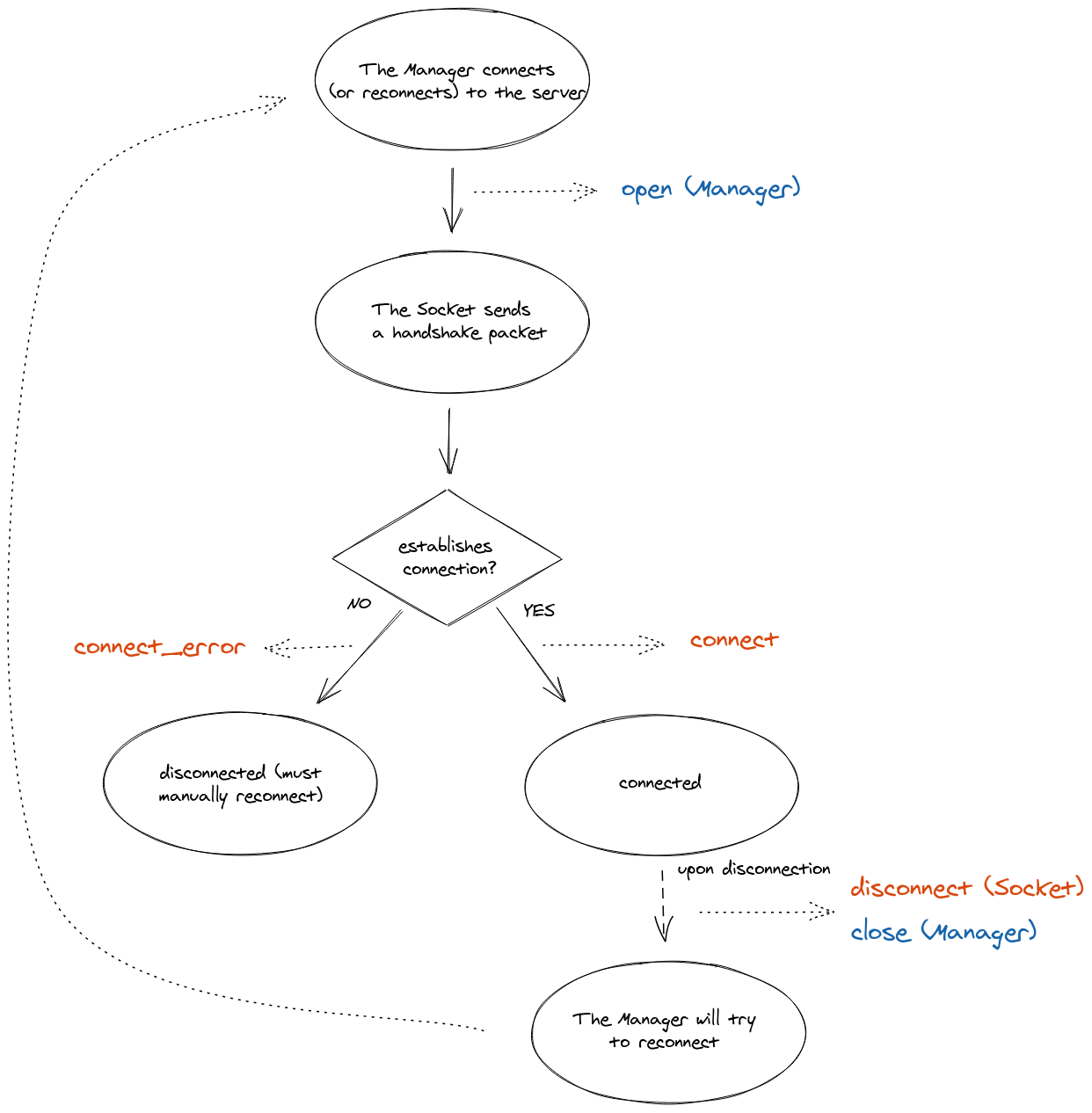
})

socket.on("disconnect", () => {

  console.log(socket.connected) // false

})

* Vòng đời



Hình 2.15 Vòng đời của Socket.io phía máy khách

* Sự kiện

Socket phát ra ba sự kiện đặc biệt.

* connect

Sự kiện này được kích hoạt bởi máy khách khi kết nối và kết nối lại. Sự kiện này được kích hoạt bởi thể hiện Socket khi kết nối và kết nối lại.

socket.on("connect", () => {

  // ...

})

* connect\_error

Sự kiện này được kích hoạt khi:

* Kết nối cấp thấp không thể được thiết lập: Socket sẽ tự động cố gắng kết nối lại sau một khoảng thời gian nhất định.
* Kết nối bị máy chủ từ chối trong chức năng phần mềm trung gian (middleware): cần kết nối lại theo cách thủ công.

socket.on("connect\_error", () => {

  // ...

})

* disconnect

Sự kiện này được kích hoạt khi ngắt kết nối.

socket.on("disconnect", (reason) => {

  // ...

})

Bảng 2.3 Danh sách các lý do Socket máy chủ ngắt kết nối có thể xảy ra

|  |  |
| --- | --- |
| **Lý do** | **Mô tả** |
| io server disconnect | Máy chủ đã ngắt kết nối với máy khách với socket.disconnect() |
| io client disconnect | Máy khách đã được ngắt kết nối theo cách thủ công bằng socket.disconnect() |
| ping timeout | Máy chủ không gửi PING trong phạm vi pingInterval + pingTimeout |
| transport close | Kết nối đã bị đóng (ví dụ: người dùng bị mất kết nối hoặc mạng đã được thay đổi từ WiFi thành 4G) |
| transport error | Kết nối đã gặp lỗi (ví dụ: máy chủ đã bị ngắt trong chu kỳ bỏ phiếu dài hạn HTTP) |

Trong hai trường hợp đầu tiên (ngắt kết nối rõ ràng), máy khách sẽ không thể tự kết nối lại mà cần gọi thủ công socket.connect(). Trong các trường hợp còn lại, máy khách sẽ tự động cố gắng kết nối lại sau một khoảng thời gian nhất định.

Lưu ý: những sự kiện ở trên, cùng với sự kiện disconnecting, newListener và removeListener là những sự kiện đặc biệt không được sử dụng trong ứng dụng.

* Các sự kiện chính
* Sự kiện phát (emitting events)

Có một số cách để gửi sự kiện giữa máy chủ và máy khách.

* Phát cơ bản

Ứng dụng có thể phát ra các sự kiện ở một bên và đăng ký bộ nghe ở bên kia.

Phát ra sự kiện có tên eventName cùng với data cần truyền: socket.emit(eventName, data)

// server-side

io.on("connection", (socket) => {

  socket.emit("hello", "world")

})

// client-side

socket.on("hello", (arg) => {

  console.log(arg) // world

})

Điều này cũng hoạt động theo hướng ngược lại.

// server-side

io.on("connection", (socket) => {

  socket.on("hello", (arg) => {

    console.log(arg) // world

  })

})

// client-side

socket.emit("hello", "world")

* Sự xác nhận

socket.emit(eventName, data, callback)

Socket cho phép thêm một lệnh gọi lại làm đối số cuối cùng của hàm emit() và lệnh gọi lại này sẽ được gọi khi phía bên kia xác nhận sự kiện.

// server-side

io.on("connection", (socket) => {

  socket.on("update item", (arg1, arg2, callback) => {

    console.log(arg1) // 1

    console.log(arg2) // { name: "updated" }

    callback({

      status: "ok"

    })

  })

})

// client-side

socket.emit("update item", "1", { name: "updated" }, (response) => {

  console.log(response.status) // ok

})

* Lắng nghe các sự kiện

Có một số cách để xử lý các sự kiện được truyền giữa máy chủ và máy khách.

Lắng nghe sự kiện có tên eventName, listener là hàm xử lý sự kiện với tham số nhận vào là data mà bên phát ra sự kiện truyền theo: socket.on(eventName, listener)

socket.on("details", (...args) => {

  // ...

})

Lắng nghe sự kiện có tên eventName một lần duy nhất: socket.once(eventName, listener)

socket.once("details", (...args) => {

  // ...

})

socket.off(eventName, listener)

Loại bỏ trình nghe đã chỉ định khỏi mảng trình nghe cho sự kiện có tên eventName.

const listener = (...args) => {

  console.log(args)

}

socket.on("details", listener)

// and then later...

socket.off("details", listener)

Loại bỏ tất cả trình nghe hoặc trình nghe mà eventName được chỉ định.

socket.removeAllListeners([eventName])

// for a specific event

socket.removeAllListeners("details")

// for all events

socket.removeAllListeners()

* Xử lý lỗi

Hiện không có xử lý lỗi tích hợp sẵn trong thư viện Socket.IO, có nghĩa là người phát triển ứng dụng phải bắt bất kỳ lỗi nào có thể xảy ra trong trình nghe.

io.on("connection", (socket) => {

  socket.on("list items", async (callback) => {

    try {

      const items = await findItems();

      callback({

        status: "OK",

        items

      })

    } catch (e) {

      callback({

        status: "NOK"

      })

    }

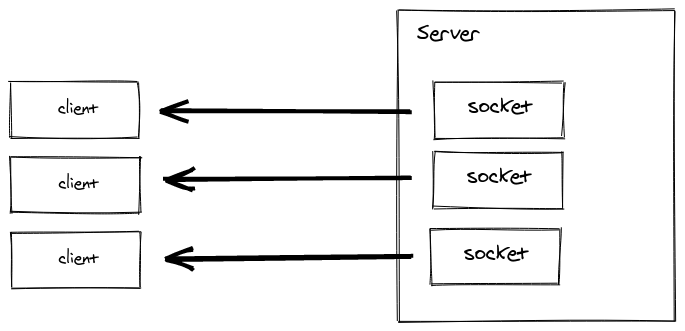
  })

})

* Sự kiện quảng bá

Socket.io giúp dễ dàng gửi các sự kiện đến tất cả các máy khách được kết nối.

Lưu ý: Sự kiện quảng bá là một tính năng chỉ dành cho máy chủ.



Hình 2.16 Sự kiện quảng bá toàn máy khách trong Socket.io

io.on("connection", (socket) => {

  io.emit("hello", "world")

})

Cho tất cả các máy khách được kết nối ngoại trừ người gửi.



Hình 2.17 Sự kiện quảng bá trong Socket.io

// server-side

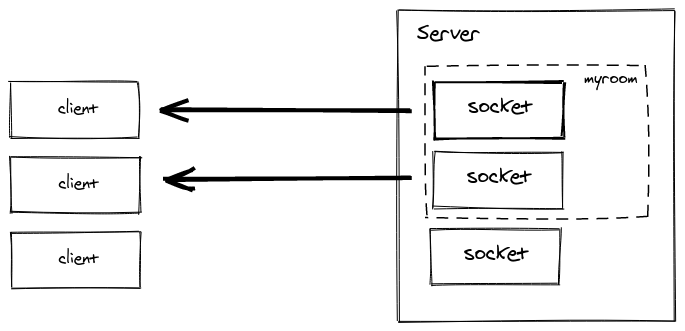
io.on("connection", (socket) => {

  socket.broadcast.emit("hello", "world")

})

* Room

Phòng (room) là một kênh tùy ý mà các socket có thể tham gia và rời khỏi. Nó có thể được sử dụng để truyền phát các sự kiện cho một nhóm nhỏ máy khách.



Hình 2.18 Room trong Socket.io

Lưu ý: room là khái niệm chỉ dành cho máy chủ (tức là máy khách không có quyền truy cập vào danh sách các phòng đã tham gia).

**Tham gia vào phòng**

Cho socket tham gia vào phòng có id là roomName: socket.join(roomName)

io.on('connection', socket => {

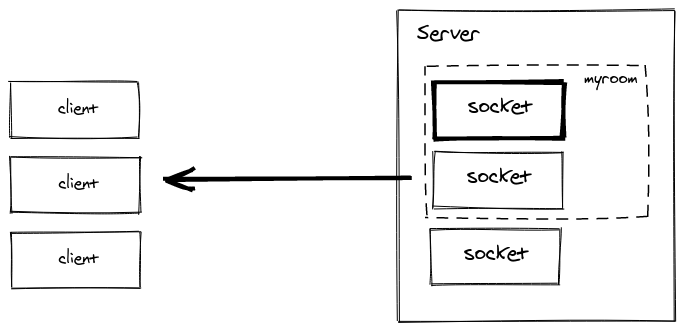
  socket.join('room')

})

Sử dụng ***to*** hoặc ***in*** để phát sự kiện trong phòng: io.to(‘room').emit('someEvent')

Thư viện cũng cho phép phát sự kiện nhiều phòng cùng một lúc (mỗi máy khách nằm trong phòng chỉ nhận được một sự kiện được phát ra cho dù nó có thể nằm trong nhiều phòng trong danh sách): io.to('room1').to('room2').to('room3').emit('someEvent');

Thư viện cũng cho phép phát ra sự kiện quảng bá đến một phòng (trừ người gửi).



Hình 2.19 Sự kiện quảng bá của room trong socket.io

io.on('connection', (socket) => {

  socket.to('room').emit('someEvent')

})

**Phòng mặc định**

Được sử dụng để gửi tin nhắn riêng tư

io.on("connection", socket => {

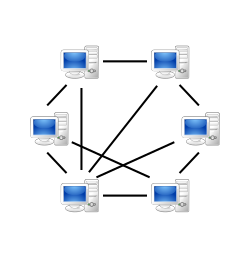
  socket.on("private message", (anotherSocketId, msg) => {

    socket.to(anotherSocketId).emit("private message", socket.id, msg)

  })

})

1. **Tổng quan về mạng ngang hàng (peer to peer network - P2P)**



Hình 2.20 Mạng ngang hàng

Mạng ngang hàng là mạng trong đó hai hoặc nhiều máy tính kết nối với nhau, chia sẻ tệp và truy cập vào các thiết bị như máy in trong mạng mà không yêu cầu máy chủ riêng biệt hoặc phần mềm máy chủ.

Ở dạng đơn giản nhất, mạng ngang hàng được tạo khi hai hoặc nhiều PC được kết nối và chia sẻ tài nguyên mà không cần thông qua một máy chủ riêng biệt. Mạng P2P có thể là kết nối đặc biệt - một vài máy tính được kết nối qua Bus nối tiếp đa năng để truyền tệp. Mạng P2P cũng có thể là một cơ sở hạ tầng cố định liên kết nửa tá máy tính trong một văn phòng nhỏ qua dây đồng. Hoặc mạng P2P có thể là một mạng ở quy mô lớn hơn, trong đó các giao thức và ứng dụng đặc biệt thiết lập mối quan hệ trực tiếp giữa những người dùng qua Internet.

Trong khi các mô hình điện toán tập trung trước đó và hệ thống máy khách / máy chủ ngày nay thường được coi là môi trường được kiểm soát, trong đó các cá nhân sử dụng PC của họ theo cách được xác định bởi cơ quan có thẩm quyền cao hơn, thì mạng nhóm làm việc P2P cổ điển là chia sẻ công khai các tệp và thiết bị.

Trong môi trường P2P, quyền truy cập được điều chỉnh bằng cách thiết lập quyền chia sẻ trên các máy riêng lẻ.

1. **Tổng quan về WebRTC (Web real time communication)**
2. **Tổng quan**

WebRTC là một dự án mã nguồn mở, miễn phí cung cấp cho các trình duyệt web và ứng dụng di động khả năng giao tiếp thời gian thực (RTC) thông qua các giao diện lập trình ứng dụng đơn giản (API). Nó cho phép giao tiếp âm thanh và video hoạt động bên trong các trang web bằng cách cho phép giao tiếp ngang hàng trực tiếp, loại bỏ nhu cầu cài đặt phần mềm hoặc tải xuống các ứng dụng gốc. Được hỗ trợ bởi Apple, Google, Microsoft, Mozilla và Opera, các thông số kỹ thuật của WebRTC đã được xuất bản bởi W3C và IETF.

Các thành phần chính của WebRTC bao gồm một số API JavaScript:

* getUserMedia lấy phương tiện âm thanh và video (bằng cách truy cập vào máy ảnh và micrô của thiết bị).
* MediaStream là một luồng dữ liệu âm thanh và / hoặc video. Khi làm việc cục bộ, có thể lấy được bằng cách gọi getUserMedia. Sau khi thiết lập kết nối WebRTC thành công, quyền truy cập vào luồng phương tiện của trình duyệt từ xa sẽ khả dụng.

const stream = await navigator.mediaDevices.getUserMedia({

  video: true,

  audio: true

})

* RTCDataChannel cho phép giao tiếp hai chiều dữ liệu tùy ý giữa các đồng nghiệp. Nó sử dụng cùng một API như WebSockets và có độ trễ rất thấp.
* RTCPeerConnection cho phép giao tiếp âm thanh và video giữa các đồng nghiệp. Nó thực hiện xử lý tín hiệu, giao tiếp ngang hàng, bảo mật và quản lý băng thông.

API WebRTC không bao gồm điều khoản nào cho việc báo hiệu, tức là kiểm tra các đồng nghiệp để kết nối và xác định cách thiết lập kết nối giữa chúng.

1. **Thư viện simple-peer**

Thư viện simple-peer là một thư viện ngắn gọn, API kiểu NodeJs cho WebRTC

* Hoạt động trong Node và trình duyệt.
* Hỗ trợ luồng video / audio
* Hỗ trợ kênh dữ liệu
* Dữ liệu văn bản và nhị phân
* Luồng song công NodeJs
* Các xử lý cơ bản trong thư viện
* Tạo cuộc gọi (peer call)

const peerCall = new SimplePeer({

  initiator: true,

  trickle: false

})

* Tạo cuộc nghe (peer answer)

const peerAnswer = new SimplePeer({

  initiator: false,

  trickle: false

})

* Kênh dữ liệu

peerCall.on('signal', data => {

  // khi peerCall có tín hiệu kết nối (signal), hãy cung cấp nó cho peerAnswer bằng cách nào đó

  peerAnswer.signal(data)

})

peerAnswer.on('signal', data => {

  // khi peerAnswer có tín hiệu kết nối (signal), hãy cung cấp nó cho peerCall bằng cách nào đó

  peerCall.signal(data)

})

peerCall.on('connect', () => {

  // đợi sự kiện ''connect'' trước khi sử dụng kênh dữ liệu

  peerCall .send('hey answer, how is it going?')

})

peerAnswer .on('data', data => {

  // nhận được một tin nhắn kênh dữ liệu

  console.log('got a message from peerCall: ' + data)

})

* Video / Audio

const stream = await navigator.mediaDevices.getUserMedia({

  video: true,

  audio: true

})

peerCall.addStream(stream)

peerCall.on('signal', data => {

  peerAnswer .signal(data)

})

peerAnswer .on('signal', data => {

  peerCall.signal(data)

})

peerAnswer .on('stream', stream => {

  // có luồng video từ xa, bây giờ hãy hiển thị nó trong thẻ video

  const video = document.querySelector('video')

  if ('srcObject' in video) {

    video.srcObject = stream

  } else {

    video.src = window.URL.createObjectURL(stream)

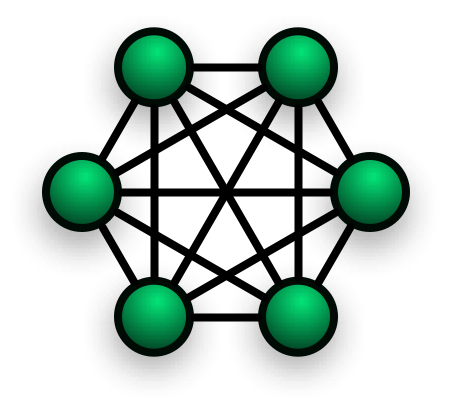
  }

  video.play()

})

* Vấn đề kết nối nhiều hơn hai đồng đẳng

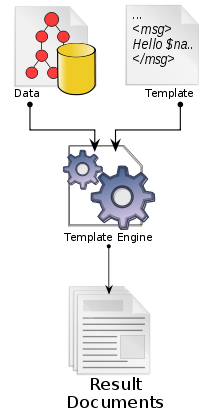
Cách đơn giản nhất để làm điều đó là tạo một cấu trúc liên kết lưới. Điều đó có nghĩa là mọi đồng đẳng sẽ mở ra một kết nối với mọi đồng đẳng khác.



Hình 2.21 Mạng lưới kết nối nhiều hơn hai đồng đẳng trong simple-peer

Để phát một tin nhắn, chỉ cần lặp lại trên tất cả các đồng nghiệp ngang hàng và gọi peer.send. Vì vậy, giả sử có 3 đồng nghiệp. Sau đó, khi một đồng đẳng muốn gửi một số dữ liệu, nó phải gửi nó 2 lần, một lần cho mỗi đồng nghiệp khác. Vì vậy, phải nên cẩn thận một chút về kích thước của dữ liệu muốn gửi. Các cấu trúc liên kết đầy đủ của lưới không mở rộng quy mô tốt khi số lượng các đồng đẳng là rất lớn. Tổng số cạnh trong mạng sẽ là với n là số đồng đẳng.

1. **Pug**
2. **Tổng quan về template engine**



Hình 2.22 Cấu trúc của một template engine

Template engine là phần mềm được thiết kế để kết hợp các mẫu với một mô hình dữ liệu để tạo ra nhiều trang có cùng giao diện trên toàn bộ trang web. Đây là views trong một dự án MVC.

1. **Tổng quan về Pug template**

Pug là một template engine cho Node và cho trình duyệt. Nó biên dịch sang HTML và có cú pháp đơn giản, giúp làm việc hiệu quả hơn và mã dễ đọc hơn. Pug giúp cho việc dễ dàng viết HTML có thể tái sử dụng cũng như kết xuất dữ liệu được lấy từ cơ sở dữ liệu hoặc API.

* Cú pháp
* Cú pháp viết 1 thẻ html trong pug

tagName#id.class(attribute1=”value”, attribute2=”value”)

  tagNameSub#id.class(attribute1=”value”, attribute2=”value”) content

Ví dụ

ul#list

  li.item

    a(href='/') Home

  li.item

    a(href='/page-1') Page 1

  li.item

    a(href='/page-2') Page 2

Kết quả

<ul id="list">

    <li class="item"><a href="/">Home</a></li>

    <li class="item"><a href="/page-1">Page 1</a></li>

    <li class="item"><a href="/page-2">Page 2</a></li>

</ul>

* Cú pháp chia layout trong Pug template

Sử dụng từ khóa block để định nghĩa 1 khối nội dung trong template cha.

//- layout.pug

html

  head

    title My Site - #{title}

    block scripts

      script(src='/jquery.js')

  body

    block content

    block foot

      #footer

        p some footer content

Sử dụng từ khóa block template con để ghi đè nội dung của block đó trong template cha.

//- page-a.pug

extends layout.pug

block scripts

  script(src='/jquery.js')

  script(src='/pets.js')

block content

  h1= title

  - var pets = ['cat', 'dog']

  each petName in pets

    include pet.pug

Sử dụng từ khóa append / prepend trong template con để gộp nội dung thêm vào block vào sau/trước nội dung đã có trong block đó trong template cha.

//- layout.pug

html

  head

    block head

      script(src='/vendor/jquery.js')

      script(src='/vendor/caustic.js')

  body

    block content

//- page.pug

extends layout

append head

  script(src='/vendor/three.js')

  script(src='/game.js')

* Includes

Chèn nội dung của một template vào 1 template khác

//- index.pug

doctype html

html

  include includes/head.pug

  body

    h1 My Site

    p Welcome to my super lame site.

    include includes/foot.pug

//- includes/head.pug

head

  title My Site

  script(src='/javascripts/jquery.js')

  script(src='/javascripts/app.js')

//- includes/foot.pug

footer#footer

  p Copyright (c) foobar

Kết quả

<!DOCTYPE html>

<html>

<head>

  <title>My Site</title>

  <script src="/javascripts/jquery.js"></script>

  <script src="/javascripts/app.js"></script>

</head>

<body>

  <h1>My Site</h1>

  <p>Welcome to my super lame site.</p>

  <footer id="footer">

    <p>Copyright (c) foobar</p>

  </footer>

</body>

</html>

- Khai báo biến

- var title = "On Dogs: Man's Best Friend";

- var author = "enlore";

h1= title

p Written with love by #{author.toUpperCase()}

Kết quả

<h1>On Dogs: Man's Best Friend</h1>

<p>Written with love by ENLORE</p>

- Câu điều kiện

- var user = {description: 'foo bar baz'}

- var authorised = true

#user

  if !user.description

    h2.green Description

    p.description= user.description

  else if authorised

    h2.blue Description

    p.description.

      User has no description,

      why not add one...

  else

    h2.red Description

    p.description User has no description

Kết quả

<div id="user">

  <h2 class="blue">Description</h2>

  <p class="description">User has no description,

    why not add one...</p>

</div>

* Vòng lặp

ul

  each val, index in ['zero', 'one', 'two']

    li= index + ': ' + val

Kết quả

<ul>

  <li>0: zero</li>

  <li>1: one</li>

  <li>2: two</li>

</ul>

1. **SASS/SCSS**
2. **Tổng quan**

CSS Preprocessors là ngôn ngữ tiền xử lý CSS. Là một ngôn ngữ kịch bản mở rộng của CSS và được biên dịch thành cú pháp CSS giúp viết CSS nhanh hơn và có cấu trúc rõ ràng hơn. CSS Preprocessor có thể giúp tiết kiệm thời gian viết CSS, dễ dàng bảo trì và phát triển CSS.

SASS/SCSS là một chương trình tiền xử lý CSS (CSS preprocessor). Nó giúp viết CSS theo cách của một ngôn ngữ lập trình, có cấu trúc rõ ràng, rành mạch, dễ phát triển và bảo trì code hơn. Ngoài ra nó có rất nhiều các thư viện hỗ trợ kèm theo giúp viết code CSS một cách dễ dàng vào đơn giản hơn. Có rất nhiều loại CSS Preprocessor trong đó bao gồm SASS, Stylus hay LESS.

1. **Cú pháp**

SASS và SCSS về bản chất vấn đề là giống nhau, chỉ khác nhau ở cách viết.

Ví dụ:

Trong SASS (.sass)

.main

  background-color: #f3f3f3

    .headline

      color: #000

Trong SCSS (.scss)

.main {

  background-color: #f3f3f3;

  &.headline {

    color: #000;

  }

}

* Một số cú pháp trong SCSS
* Khai báo biến

Ví dụ

$base-color: #c6538c;

.alert {

  border: 1px solid $base-color;

}

* Mixin: giúp định nghĩa những style được sử dụng lại và có thể truyền được tham số và xử lý logic

Ví dụ

@mixin square($size, $radius: 0) {

  width: $size;

  height: $size;

  @if $radius !=0 {

    border-radius: $radius;

  }

}

.avatar {

  @include square(100px, $radius: 4px);

}

* Function: định nghĩa các hàm trong SCSS

Ví dụ

@function pow($base, $exponent) {

  $result: 1;

  @for $\_ from 1 through $exponent {

    $result: $result \* $base;

  }

  @return $result;

}

.sidebar {

  float: left;

  margin-left: pow(4, 3) \* 1px;

}

* Placeholder: viết những đoạn style giống nhau của các selector

Ví dụ

%toolbelt {

  box-sizing: border-box;

  border-top: 1px rgba(#000, .12) solid;

  padding: 16px 0;

  width: 100%;

  &:hover {

    border: 2px rgba(#000, .5) solid;

  }

}

.action-buttons {

  @extend %toolbelt;

  color: #4285f4;

}

.reset-buttons {

  @extend %toolbelt;

  color: #cddc39;

}

1. **Tổng quan về Cloudinary**
2. **Tổng quan**

Cloudinary là một giải pháp phần mềm dưới dạng dịch vụ (SaaS) để quản lý tất cả nội dung phương tiện (file, video, ảnh...) web hoặc ứng dụng di động trên đám mây. Nội dung phương tiện được phân phối từ các máy chủ hiệu suất cao thông qua Mạng phân phối nội dung (CDN).

Các lợi ích chính cho nhà phát triển

* Loại bỏ công việc R&D liên quan đến truyền thông khỏi bộ phận CNTT và DevOps, đồng thời giảm chi phí lên đến 90%.
* Các tính năng thao tác hình ảnh tiên tiến.
* Giảm thời gian đưa sản phẩm mới ra thị trường.
* Thư viện phương tiện dựa trên đám mây và các API REST để tải lên, phân phối và quản trị.
* Cải thiện trải nghiệm người dùng: trang web trông tuyệt vời và tải nhanh trên mọi thiết bị.

1. **Các API chính**

Quá trình sử dụng các API phải có chữ ký đám mây được thực hiện qua HTTPS bằng giao thức bảo mật dựa trên các thông số cloud\_name, api\_key và api\_secret trong tài khoản của nhà phát triển.

* Tải nội dung lên đám mây

cloudinary.v2.uploader.upload(file, options, callback)

Ví dụ: tải lên tệp hình ảnh cục bộ có tên sample.jpg

cloudinary.v2.uploader.upload(

  'sample.jpg',

  {

    public\_id: 'sample',

    resource\_type: 'auto',

  },

  function (error, result) {

    console.log(result, error)

  }

)

Dữ liệu trả về

{

  public\_id: 'sample',

  version: '1312461204',

  width: 864,

  height: 564,

  format: 'jpg',

  created\_at: '2017-08-10T09:55:32Z',

  resource\_type: 'image',

  tags: [],

  bytes: 9597,

  type: 'upload',

  etag: 'd1ac0ee70a9a36b14887aca7f7211737',

  url: 'http://res.cloudinary.com/demo/image/upload/v1312461/sample.jpg',

  secure\_url: 'https://res.cloudinary.com/demo/image/upload/v1312461/sample.jpg',

  signature: 'abcdefgc024acceb1c1baa8dca46717137fa5ae0c3',

  original\_filename: 'sample'

}

* Xóa tài nguyên

cloudinary.v2.api.delete\_resources(public\_ids, options, callback)

Xóa hình ảnh đã tải lên theo public IDs

cloudinary.v2.api.delete\_resources(

  ['image1', 'image2'],

  {

    resource\_type: 'image',

  },

  function (error, result) {

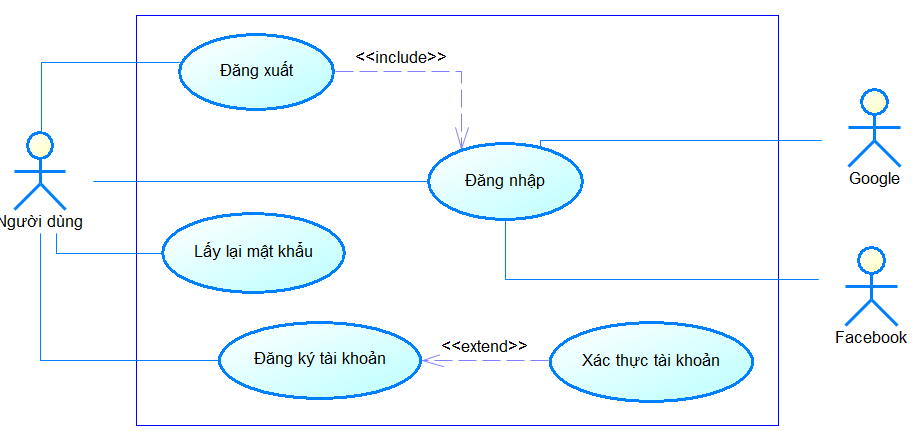
    console.log(result, error)

  }

)

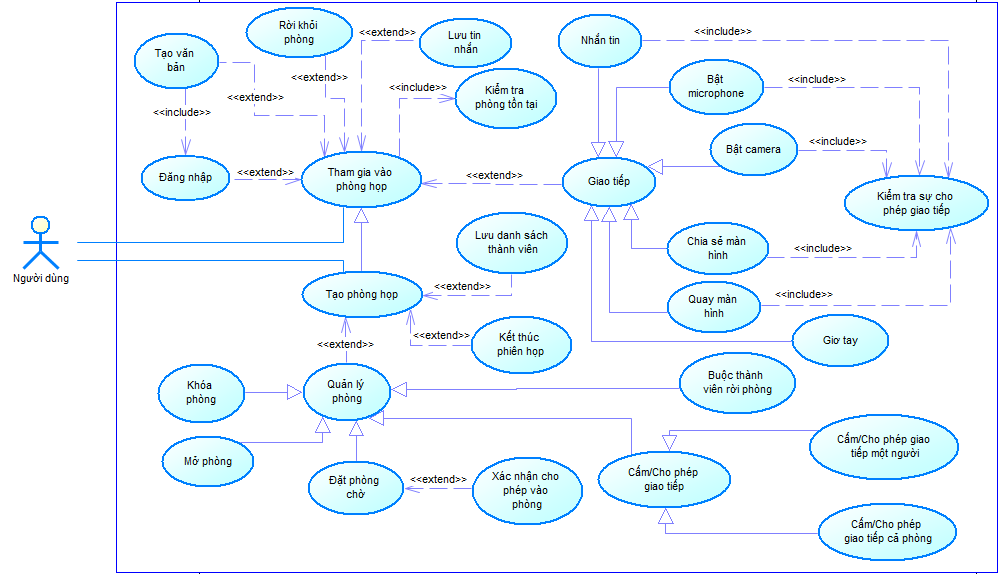
# PHÂN TÍCH HỆ THỐNG

1. **Sơ đồ use case**
2. **Nhóm use case đăng nhập**



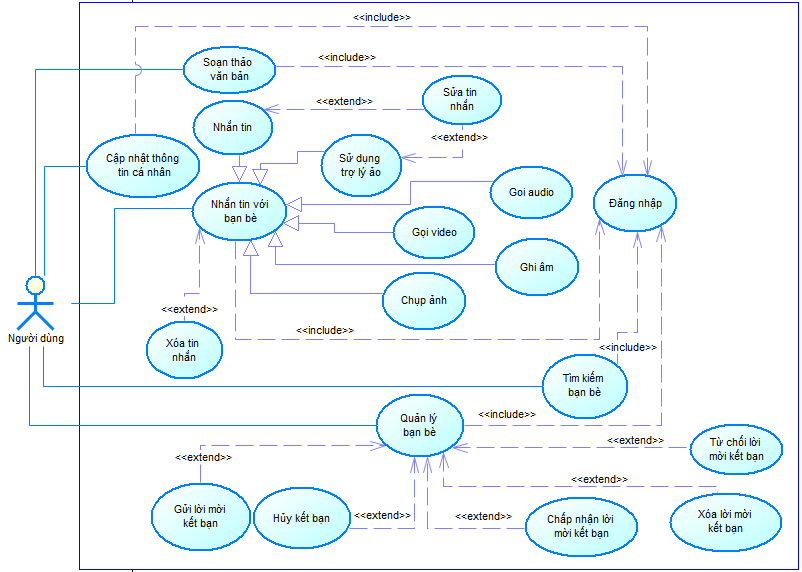
Hình 3.1 Nhóm use case đăng nhập

1. **Nhóm use case phòng họp**



Hình 3.2 Nhóm use case phòng họp

1. **Nhóm use case trò chuyện với bạn bè**



Hình 3.3 Nhóm use case trò chuyện với bạn bè

1. **Bộ mô tả use case**
2. **Use case đăng nhập**

|  |  |
| --- | --- |
| **Tên use case** | Đăng nhập |
| **Mô tả** | Người dùng muốn đăng nhập vào ứng dụng để sử dụng dịch vụ từ ứng dụng |
| **Tác nhân chính** | Người dùng, Google, Facebook |
| **Mức** | Phải có |
| **Kích hoạt** | Người dùng đăng nhập vào ứng dụng |
| **Tiền điều kiện** | * Tài khoản người dùng đã được đăng ký và xác thực kích hoạt * Thiết bị của người dùng đã được kết nối internet khi thực hiện đăng nhập |
| **Đảm bảo tối thiểu** | Hệ thống thông báo lỗi đăng nhập thất bại |
| **Đảm bảo thành công** | * Người dùng đăng nhập ứng dụng thành công * Hệ thống tạo phiên đăng nhập cho người dùng |
| **Chuỗi sự kiện chính** | 1. Người dùng truy cập ứng dụng  2. Người dùng nhập tài khoản và chọn lệnh đăng nhập  3. Hệ thống xác thực thông tin đăng nhập thành công và cho phép người dùng truy cập vào ứng dụng  4. Hệ thống tạo phiên đăng nhập cho người dùng |
| **Chuỗi sự kiện thay thế** | 2a. Người dùng chọn phương thức đăng nhập bằng tài khoản Google  2a1. Hệ thống chuyển sang màn hình đăng nhập của Google  2a2. Người dùng nhập tài khoản Google và chọn lệnh đăng nhập  2a3. Google xác thực thông tin đăng nhập thành công và cho phép người dùng truy cập ứng dụng  *Use case tiếp tục bước 4*  2b. Người dùng chọn phương thức đăng nhập bằng tài khoản Facebook  2b1. Hệ thống chuyển sang màn hình đăng nhập của Facebook  2b2. Người dùng nhập tài khoản Facebook và chọn lệnh đăng nhập  2b3. Facebook xác thực thông tin đăng nhập thành công và cho phép người dùng truy cập ứng dụng  *Use case tiếp tục bước 4* |
| **Chuỗi sự kiện ngoại lệ** | 3c. Hệ thống xác thực thông tin đăng nhập không thành công và hiển thị thông báo  3ca. Người dùng nhập lại thông tin đăng nhập vào form đăng nhập  *Use case tiếp tục bước 3*  3cb. Người dùng không đăng nhập lại và thoát khỏi hệ thống  *Use case dừng lại* |

1. **Use case đăng ký tài khoản**

|  |  |
| --- | --- |
| **Tên use case** | Đăng ký tài khoản |
| **Mô tả** | Người dùng muốn đăng ký tài khoản để truy cập vào ứng dụng |
| **Tác nhân chính** | Người dùng, Google, Facebook |
| **Mức** | Phải có |
| **Kích hoạt** | Người dùng chọn lệnh đăng ký tài khoản |
| **Tiền điều kiện** | Thiết bị của người dùng đã được kết nối internet khi thực hiện đăng ký |
| **Đảm bảo tối thiểu** | Thông báo đăng ký tài khoản thất bại |
| **Đảm bảo thành công** | * Người dùng đăng ký tài khoản thành công * Hệ thống chuyển sang giao diện đăng nhập |
| **Chuỗi sự kiện chính** | 1. Người dùng truy cập ứng dụng  2. Người dùng chọn chức năng đăng ký tài khoản  3. Người dùng nhập thông tin đăng ký vào form đăng ký  4. Hệ thống xác thực thông tin đăng đăng ký thành công và thêm tài khoản mới đăng ký vào CSDL  5. Hệ thống gửi email xác thực tài khoản cho người người dùng  6. Người dùng vào email đăng ký để xác thực tài khoản  7. Hệ thống xác thực xác nhận đăng ký tài khoản thành công và cập nhật CSDL |
| **Chuỗi sự kiện thay thế** | 3a. Người dùng chọn phương thức đăng ký bằng tài khoản Google  3a1. Hệ thống chuyển sang màn hình đăng nhập của Google  3a2. Người dùng nhập tài khoản Google và chọn lệnh đăng nhập  3a3. Google xác thực thông tin đăng nhập thành công và tạo tài khoản nếu chưa có  3a4. Hệ thống tạo phiên đăng nhập cho người dùng  3b. Người dùng chọn phương thức đăng nhập bằng tài khoản Facebook  3b1. Hệ thống chuyển sang màn hình đăng nhập của Facebook  3b2. Người dùng nhập tài khoản Facebook và chọn lệnh đăng nhập  3b3. Facebook xác thực thông tin đăng nhập thành công và tạo tài khoản nếu chưa có  3b4. Hệ thống tạo phiên đăng nhập cho người dùng |
| **Chuỗi sự kiện ngoại lệ** | 4a. Hệ thống xác thực thông tin đăng đăng ký không thành công và hiển thị thông báo  4aa. Người dùng nhập lại thông tin đăng đăng ký vào form đăng đăng ký  *Use case tiếp tục bước 4*  4ab. Người dùng không đăng ký lại và thoát khỏi hệ thống  *Use case dừng lại* |

1. **Use case quên mật khẩu**

|  |  |
| --- | --- |
| **Tên use case** | Quên mật khẩu |
| **Mô tả** | Người dùng muốn đổi mật khẩu tài khoản vì quên mật khẩu cũ |
| **Tác nhân chính** | Người dùng |
| **Mức** | Không bắt buộc |
| **Kích hoạt** | Người dùng chọn lệnh quên mật khẩu |
| **Tiền điều kiện** | * Tài khoản người dùng đã được tạo sẵn * Tài khoản không phải là tài khoản đăng nhập bằng Google hay Facebook * Thiết bị của người dùng đã được kết nối internet khi thực hiện lấy lại mật khẩu |
| **Đảm bảo tối thiểu** | Thông báo đổi mật khẩu thất bại |
| **Đảm bảo thành công** | * Người dùng đổi mật khẩu thành công * Hệ thống chuyển sang giao diện đăng nhập |
| **Chuỗi sự kiện chính** | 1. Người dùng truy cập ứng dụng  2. Người dùng chọn chức năng quên mật khẩu  3. Người dùng nhập thông tin xác thực vào form yêu cầu xác thực tài khoản  4. Hệ thống xác thực thông tin yêu cầu đổi mật khẩu thành công và gửi email xác thực đổi mật khẩu cho người dùng  5. Người dùng vào email đăng ký để xác thực đổi mật khẩu  6. Hệ thống chuyển sang giao diện đổi mật khẩu  7. Người dùng nhập thông tin đổi mật khẩu vào form đổi mật khẩu  8. Hệ thống xác thực thông tin đổi mật khẩu thành công và cập nhật CSDL  9. Hệ thống quay lại trang đăng nhập |
| **Chuỗi sự kiện ngoại lệ** | 4a. Hệ thống xác thực thông tin yêu cầu đổi mật khẩu không thành công và hiển thị thông báo  4aa. Người dùng nhập lại thông tin xác thực vào form yêu cầu xác thực tài khoản  *Use case tiếp tục bước 4*  4ab. Người dùng không nhập lại thông tin xác thực đổi mật khẩu và thoát khỏi hệ thống  *Use case dừng lại*  8a. Hệ thống xác thực thông tin đổi mật khẩu không thành công và hiển thị thông báo  8aa. Người dùng nhập lại thông tin đổi mật khẩu vào form đổi mật khẩu  *Use case tiếp tục bước 8*  8ab. Người dùng không nhập lại thông tin đổi mật khẩu và thoát khỏi hệ thống  *Use case dừng lại* |

1. **Use case tạo phòng họp**

|  |  |
| --- | --- |
| **Tên use case** | Tạo phòng họp |
| **Mô tả** | Người dùng muốn tạo phòng họp để thảo luận |
| **Tác nhân chính** | Người dùng |
| **Mức** | Phải có |
| **Kích hoạt** | Người dùng chọn lệnh tạo phòng họp |
| **Tiền điều kiện** | Thiết bị của người dùng đã được kết nối internet khi thực hiện tạo phòng họp |
| **Đảm bảo tối thiểu** | Thông báo tạo phòng thất bại |
| **Đảm bảo thành công** | * Người dùng tạo phòng họp thành công * Hệ thống chuyển sang giao diện phòng họp cho chủ phòng |
| **Chuỗi sự kiện chính** | 1. Người dùng truy cập ứng dụng  2. Người dùng chọn chức năng tạo phòng họp  3. Người dùng nhập thông tin phòng họp muốn tạo vào form tạo phòng họp  4. Hệ thống xác thực thông tin tạo phòng họp thành công và lưu vào CSDL  5. Hệ thống tạo phiên họp cho người dùng  6. Hệ thống chuyển sang giao diện phòng họp cho chủ phòng |
| **Chuỗi sự kiện ngoại lệ** | 4a. Hệ thống xác thực thông tin tạo phòng họp không thành công và hiển thị thông báo  4aa. Người dùng nhập lại thông tin tạo phòng họp vào form tạo phòng họp  *Use case tiếp tục bước 4*  4ab. Người dùng không nhập lại thông tin tạo phòng họp và thoát khỏi hệ thống  *Use case dừng lại* |

1. **Use case tham gia vào phòng họp**

|  |  |
| --- | --- |
| **Tên use case** | Tham gia vào phòng họp |
| **Mô tả** | Người dùng muốn tham gia vào phòng họp để thảo luận |
| **Tác nhân chính** | Người dùng |
| **Mức** | Không bắt buộc |
| **Kích hoạt** | Người dùng chọn lệnh tham gia phòng họp |
| **Tiền điều kiện** | * Phòng họp đã được tạo * Thiết bị của người dùng đã được kết nối internet khi thực hiện tham gia vào phòng họp |
| **Đảm bảo tối thiểu** | Thông báo tham gia phòng thất bại |
| **Đảm bảo thành công** | * Người dùng tham gia phòng họp thành công * Hệ thống chuyển sang giao diện phòng họp cho người tham gia |
| **Chuỗi sự kiện chính** | 1. Người dùng truy cập ứng dụng  2. Người dùng chọn chức năng tham gia phòng họp  3. Người dùng nhập thông tin phòng họp muốn tham gia vào form tham gia phòng họp  4. Hệ thống xác thực thông tin tham gia phòng họp thành công và lưu vào CSDL  5. Hệ thống tạo phiên họp cho người dùng  6. Hệ thống chuyển sang giao diện phòng họp cho người tham gia |
| **Chuỗi sự kiện ngoại lệ** | 4a. Hệ thống xác thực thông tin tham gia phòng họp không thành công và hiển thị thông báo  4aa. Người dùng nhập lại thông tin tham gia phòng họp vào form tham gia phòng họp  *Use case tiếp tục bước 4*  4ab. Người dùng không nhập lại thông tin tham gia phòng họp và thoát khỏi hệ thống  *Use case dừng lại* |

1. **Use case quản lý phòng họp**

|  |  |
| --- | --- |
| **Tên use case** | Quản lý phòng |
| **Mô tả** | Chủ phòng muốn quản lý phòng họp để điều khiển hoạt động của phòng họp |
| **Tác nhân chính** | Người dùng |
| **Mức** | Không bắt buộc |
| **Kích hoạt** | Người dùng chọn một lệnh quản lý phòng họp |
| **Tiền điều kiện** | * Người dùng là chủ phòng của phòng cần quản lý * Thiết bị của người dùng đã được kết nối internet khi thực hiện quản lý phòng |
| **Đảm bảo tối thiểu** | Hệ thống thông báo lỗi quản lý thất bại |
| **Đảm bảo thành công** | * Chủ phòng thực hiện lệnh quản lý phòng thành công * Hệ thống gửi lệnh quản lý đến thành viên bị áp dụng lệnh quản lý |
| **Chuỗi sự kiện chính** | 1. Chủ phòng chọn lệnh quản lý mở phòng  2. Hệ thống xác thực lệnh quản lý thành công và cập nhật CSDL  3. Hệ thống gửi lệnh quản lý cho các thành viên trong phòng |
| **Chuỗi sự kiện thay thế** | 1a. Chủ phòng chọn lệnh quản lý đặt phòng chờ  *Use case tiếp tục bước 3*  1a. Chủ phòng chọn lệnh quản lý mở phòng  *Use case tiếp tục bước 3*  1c. Chủ phòng chọn lệnh quản lý cấm nhắn tin  *Use case tiếp tục bước 3*  1d. Chủ phòng chọn lệnh quản lý cho phép nhắn tin  *Use case tiếp tục bước 3*  1e. Chủ phòng chọn lệnh quản lý cấm bật microphone  *Use case tiếp tục bước 3*  1f. Chủ phòng chọn lệnh quản lý cho phép bật microphone  *Use case tiếp tục bước 3*  1g. Chủ phòng chọn lệnh quản lý cấm bật camera  *Use case tiếp tục bước 3*  1h. Chủ phòng chọn lệnh quản lý cho phép bật camera  *Use case tiếp tục bước 3*  1i. Chủ phòng chọn lệnh quản lý cấm chia sẻ màn hình  *Use case tiếp tục bước 3*  1j. Chủ phòng chọn lệnh quản lý cho phép chia sẻ màn hình  *Use case tiếp tục bước 3*  1k. Chủ phòng chọn lệnh quản lý cấm quay màn hình  *Use case tiếp tục bước 3*  1l. Người dùng chọn lệnh quản lý cho phép quay màn hình  *Use case tiếp tục bước 3*  1m. Chủ phòng chọn lệnh cấm giao tiếp một thành viên  1m1. Hệ thống xác thực lệnh quản lý thành công và cập nhật CSDL  1m2. Hệ thống gửi lệnh quản lý cho thành viên bị chỉ định lệnh quản lý  1n. Chủ phòng chọn lệnh cho phép giao tiếp một thành viên  1n1. Hệ thống xác thực lệnh quản lý thành công và cập nhật CSDL  1n2. Hệ thống gửi lệnh quản lý cho thành viên bị chỉ định lệnh quản lý  1o. Chủ phòng chọn lệnh cho phép một thành viên trong phòng chờ tham gia vào phòng  1o1. Hệ thống xác thực lệnh quản lý thành công và cập nhật CSDL  1o2. Hệ thống gửi lệnh quản lý cho thành viên bị chỉ định lệnh quản lý  1p. Chủ phòng chọn lệnh không cho phép một thành viên trong phòng chờ tham gia vào phòng  1p1. Hệ thống xác thực lệnh quản lý thành công và cập nhật CSDL  1p2. Hệ thống gửi lệnh quản lý cho thành viên bị chỉ định lệnh quản lý |
| **Chuỗi sự kiện ngoại lệ** | 2a. Hệ thống xác thực lệnh quản lý không thành công và hiển thị thông báo  2aa. Người dùng chọn lại lệnh quản lý  *Use case tiếp tục bước 2*  2ab. Người dùng không chọn lại lệnh quản lý  *Use case dừng lại* |

1. **Use case giao tiếp trong phòng họp**

|  |  |
| --- | --- |
| **Tên use case** | Giao tiếp |
| **Mô tả** | Người dùng muốn giao tiếp để thảo luận trong phòng họp |
| **Tác nhân chính** | Người dùng |
| **Mức** | Không bắt buộc |
| **Kích hoạt** | Người dùng chọn một lệnh giao tiếp |
| **Tiền điều kiện** | * Người dùng phải là thành viên trong phòng họp * Thiết bị của người dùng đã được kết nối internet khi thực hiện giao tiếp |
| **Đảm bảo tối thiểu** | Hệ thống thông báo lỗi giao tiếp thất bại |
| **Đảm bảo thành công** | * Người dùng thực hiện giao tiếp thành công * Hệ thống gửi thông tin giao tiếp đến các thành viên khác trong phòng |
| **Chuỗi sự kiện chính** | 1. Người dùng soạn tin nhắn  2. Người dùng chọn lệnh gửi tin nhắn  3. Hệ thống xác thực thông tin giao tiếp thành công và cập nhật CSDL  4. Hệ thống gửi thông tin giao tiếp cho các thành viên trong phòng |
| **Chuỗi sự kiện thay thế** | 1a. Người dùng chọn lệnh bật microphone  *Use case tiếp tục bước 3*  1b. Người dùng chọn lệnh tắt microphone  *Use case tiếp tục bước 3*  1c. Người dùng chọn lệnh bật camera  *Use case tiếp tục bước 3*  1d. Người dùng chọn lệnh tắt camera  *Use case tiếp tục bước 3*  1e. Người dùng chọn lệnh chia sẻ màn hình  *Use case tiếp tục bước 3*  1f. Người dùng chọn lệnh tắt chia sẻ màn hình  *Use case tiếp tục bước 3*  1g. Người dùng chọn lệnh giơ tay  *Use case tiếp tục bước 3*  1h. Người dùng chọn lệnh bỏ giơ tay  *Use case tiếp tục bước 3*  1i. Người dùng chọn lệnh quay màn hình  1i1. Hệ thống xác thực lệnh quay màn hình thành công  1i2. Hệ thống thực hiện quay màn hình  1j. Người dùng chọn lệnh dừng quay màn hình  1j1. Hệ thống thực hiện dừng quay màn hình |
| **Chuỗi sự kiện ngoại lệ** | 3a. Hệ thống xác thực thông tin giao tiếp không thành công và hiển thị thông báo  3aa. Người dùng chọn lại lệnh giao tiếp  *Use case tiếp tục bước 3*  3ab. Người dùng không chọn lại lệnh giao tiếp  *Use case dừng lại* |

1. **Use case rời phòng họp**

|  |  |
| --- | --- |
| **Tên use case** | Rời khỏi phòng |
| **Mô tả** | Người dùng muốn rời khỏi phòng họp |
| **Tác nhân chính** | Người dùng |
| **Mức** | Không bắt buộc |
| **Kích hoạt** | Người dùng chọn lệnh rời phòng họp |
| **Tiền điều kiện** | * Người dùng phải là thành viên trong phòng họp * Thiết bị của người dùng đã được kết nối internet khi thực hiện rời phòng họp |
| **Đảm bảo tối thiểu** | Hệ thống thông báo lỗi rời phòng thất bại |
| **Đảm bảo thành công** | * Người dùng rời phòng họp thành công * Hệ thống gửi thông tin rời phòng đến các thành viên trong phòng họp |
| **Chuỗi sự kiện chính** | 1. Người dùng chọn lệnh rời phòng họp  2. Hệ thống thực hiện rời phòng thành công và cập nhật CSDL  3. Hệ thống gửi thông tin rời phòng cho các thành viên trong phòng |
| **Chuỗi sự kiện ngoại lệ** | 2a. Hệ thống thực hiện rời phòng không thành công và hiển thị thông báo  2aa. Người dùng chọn lại lệnh rời phòng  *Use case tiếp tục bước 2*  2ab. Người dùng không chọn lại lệnh rời phòng  *Use case dừng lại* |

1. **Use case kết thúc phiên họp**

|  |  |
| --- | --- |
| **Tên use case** | Kết thúc phiên họp |
| **Mô tả** | Chủ phòng muốn kết thúc phiên họp |
| **Tác nhân chính** | Người dùng |
| **Mức** | Không bắt buộc |
| **Kích hoạt** | Chủ phòng chọn lệnh kết thúc phiên họp |
| **Tiền điều kiện** | * Người dùng phải là chủ phòng của phòng muốn kết thúc phiên họp * Thiết bị của người dùng đã được kết nối internet khi thực hiện kết thúc phiên họp |
| **Đảm bảo tối thiểu** | Hệ thống thông báo lỗi kết thúc phiên họp thất bại |
| **Đảm bảo thành công** | * Hệ thống gửi thông tin kết thúc phiên đến các thành viên trong phòng họp * Phiên họp kết thúc thành công |
| **Chuỗi sự kiện chính** | 1. Chủ phòng chọn lệnh kết thúc phiên họp  2. Hệ thống thực hiện xác thực kết thúc phiên họp thành công và cập nhật CSDL  3. Hệ thống gửi thông tin kết thúc phiên họp cho các thành viên trong phòng |
| **Chuỗi sự kiện ngoại lệ** | 2a. Hệ thống thực hiện xác thực kết thúc phiên họp không thành công và hiển thị thông báo  2aa. Chủ phòng chọn lại lệnh kết thúc phiên họp  *Use case tiếp tục bước 3*  2ab. Chủ phòng không chọn lại lệnh kết thúc phiên họp  *Use case dừng lại* |

1. **Use case nhắn tin với bạn bè**

|  |  |
| --- | --- |
| **Tên use case** | Nhắn tin với bạn bè |
| **Mô tả** | Người dùng muốn nhắn tin với bạn bè bằng tin nhắn, gọi audio, gọi video, ghi âm, chụp ảnh, trợ lý ảo |
| **Tác nhân chính** | Người dùng |
| **Mức** | Không bắt buộc |
| **Kích hoạt** | Người dùng chọn một lệnh nhắn tin với bạn bè |
| **Tiền điều kiện** | * Người dùng phải đăng nhập vào ứng dụng * Thiết bị của người dùng đã được kết nối internet khi thực hiện nhắn tin với bạn bè |
| **Đảm bảo tối thiểu** | Hệ thống thông báo lỗi nhắn tin thất bại |
| **Đảm bảo thành công** | Người dùng nhắn tin thành công cho bạn bè |
| **Chuỗi sự kiện chính** | 1. Người dùng chọn người bạn muốn nhắn tin  2. Người dùng soạn tin nhắn  3. Người dùng chọn lệnh gửi tin nhắn  4. Hệ thống xác thực thông tin tin nhắn thành công và cập nhật CSDL  5. Hệ thống gửi thông tin tin nhắn cho người dùng nhận tin nhắn |
| **Chuỗi sự kiện thay thế** | 2a. Người dùng chọn lệnh gọi audio  2a1. Người nhận chấp nhận cuộc gọi  2a2. Hệ thống kết nối cuộc gọi thành công và cập nhật CSDL  2a3. Người dùng ngắt kết nối cuộc gọi và cập nhật CSDL  2a4. Hệ thống xử lý cuộc gọi kết thúc  *Use case kết thúc*  2b. Người dùng chọn lệnh gọi video  2b1. Người nhận chấp nhận cuộc gọi  2b2. Hệ thống kết nối cuộc gọi thành công và cập nhật CSDL  2b3. Người dùng ngắt kết nối cuộc gọi và cập nhật CSDL  2b4. Hệ thống xử lý cuộc gọi kết thúc  *Use case kết thúc*  2c. Người dùng chọn lệnh ghi âm  2c1. Hệ thống thực hiện việc ghi âm  *Use case tiếp tục bước 3*  2d. Người dùng chọn lệnh chụp ảnh  2d1. Hệ thống thực hiện việc chụp ảnh  *Use case tiếp tục bước 3*  2e. Người dùng chọn lệnh nhắn tin bằng trợ lý ảo  2e1. Người dùng sử dụng trợ lý ảo để soạn tin nhắn  2e2. Hệ thống xử lý dữ liệu từ trợ lý ảo  *Use case tiếp tục bước 3* |
| **Chuỗi sự kiện ngoại lệ** | 4a. Hệ thống xác thực thông tin tin nhắn không thành công và hiển thị thông báo  4aa. Người dùng chọn lại lệnh nhắn tin  *Use case tiếp tục bước 4*  4ab. Người dùng không chọn lại lệnh nhắn tin  *Use case dừng lại*  2a2a. Hệ thống kết nối cuộc gọi không thành công và hiển thị thông báo  2a2aa. Người dùng chọn lại lệnh gọi audio  *Use case tiếp tục bước 2a1*  2a2ab. Người dùng không chọn lại lệnh gọi audio  *Use case dừng lại*  2b2a. Hệ thống kết nối cuộc gọi không thành công và hiển thị thông báo  2b2aa. Người dùng chọn lại lệnh gọi audio  *Use case tiếp tục bước 2b1*  2b2ab. Người dùng không chọn lại lệnh gọi audio  *Use case dừng lại* |

1. **Use case tìm kiếm bạn bè**

|  |  |
| --- | --- |
| **Tên use case** | Tìm kiếm bạn bè |
| **Mô tả** | Người dùng muốn tìm kiếm bạn bè theo tên, số điện thoại, địa chỉ |
| **Tác nhân chính** | Người dùng |
| **Mức** | Không bắt buộc |
| **Kích hoạt** | Người dùng chọn lệnh tìm kiếm bạn bè |
| **Tiền điều kiện** | * Người dùng phải đăng nhập vào ứng dụng * Thiết bị của người dùng đã được kết nối internet khi thực hiện tìm kiếm bạn bè |
| **Đảm bảo tối thiểu** | Hệ thống thông báo không tìm thấy bạn bè nào thỏa tiêu chí tìm kiếm |
| **Đảm bảo thành công** | Hệ thống trả về kết quả tìm kiếm bạn bè |
| **Chuỗi sự kiện chính** | 1. Người dùng nhập thông tin bạn bè cần tìm kiếm vào form tìm kiếm  2. Người dùng chọn lệnh tìm kiếm  3. Hệ thống thực hiện tìm kiếm bạn bè  4. Hệ thống trả về kết quả tìm kiếm |

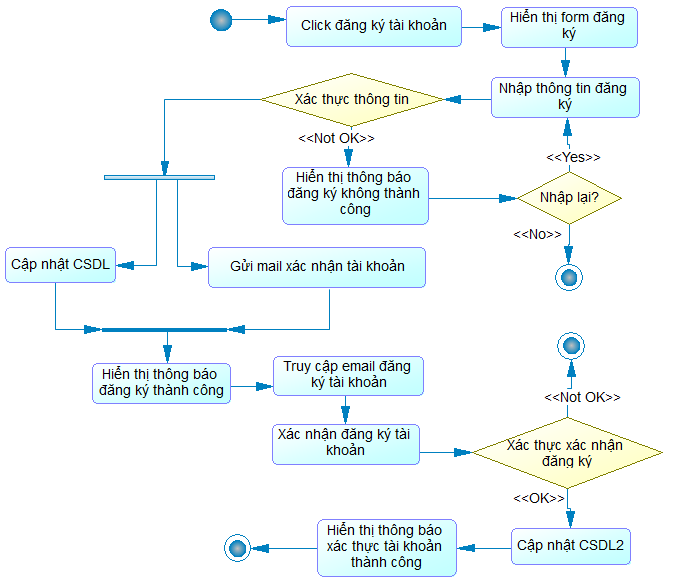
1. **Use case quản lý bạn bè**

|  |  |
| --- | --- |
| **Tên use case** | Quản lý bạn bè |
| **Mô tả** | Người dùng muốn quản lý bạn bè để gửi lời mời kết bạn, chấp nhận lời mời kết bạn, từ chối lời mời kết bạn, hủy lời mời kết bạn, hủy bạn bè |
| **Tác nhân chính** | Người dùng |
| **Mức** | Không bắt buộc |
| **Kích hoạt** | Người dùng chọn lệnh kết bạn |
| **Tiền điều kiện** | * Người dùng phải đăng nhập vào ứng dụng * Thiết bị của người dùng đã được kết nối internet khi thực hiện kết bạn |
| **Đảm bảo tối thiểu** | Hệ thống thông báo lỗi kết bạn thất bại |
| **Đảm bảo thành công** | Hệ thống thực hiện thành công lệnh quản lý bạn bè và cập nhật CSDL |
| **Chuỗi sự kiện chính** | 1. Người dùng chọn bạn bè muốn quản lý bạn bè  2. Người dùng chọn lệnh gửi lời mời kết bạn  3. Hệ thống xác thực lệnh gửi lời mời kết bạn thành công và cập nhật CSDL  4. Hệ thống gửi thông báo kết bạn cho người dùng được kết bạn |
| **Chuỗi sự kiện thay thế** | 2a1. Người dùng chọn lệnh chấp nhận lời mời kết bạn  2a2. Hệ thống xác thực lệnh chấp nhận lời mời kết bạn thành công và cập nhật CSDL  2a3. Hệ thống gửi thông báo kết bạn cho người gửi lời mời kết bạn  *Use case kết thúc*  2b1. Người dùng chọn lệnh từ chối lời mời kết bạn  2b2. Hệ thống xác thực lệnh từ chối lời mời kết bạn thành công và cập nhật CSDL  *Use case kết thúc*  2c1. Người dùng chọn lệnh xóa lời mời kết bạn  2c2. Hệ thống xác thực lệnh xóa lời mời kết bạn thành công và cập nhật CSDL  *Use case kết thúc*  2d1. Người dùng chọn lệnh hủy kết bạn  2d2. Hệ thống xác thực lệnh hủy kết bạn thành công và cập nhật CSDL  *Use case kết thúc* |
| **Chuỗi sự kiện ngoại lệ** | 3a. Hệ thống xác thực lệnh gửi lời mời kết bạn không thành công  3aa. Người dùng chọn lại lệnh gửi lời mời kết bạn  *Use case tiếp tục bước 3*  3ab. Người dùng không chọn lại lệnh gửi lời mời kết bạn  *Use case kết thúc*  2a2a. Hệ thống xác thực lệnh chấp nhận lời mời kết bạn không thành công  2a2aa. Người dùng chọn lại lệnh chấp nhận lời mời kết bạn  *Use case tiếp tục bước 2a2*  2a2ab. Người dùng không chọn lại lệnh chấp nhận lời mời kết bạn  *Use case kết thúc*  2b2a. Hệ thống xác thực lệnh từ chối lời mời kết bạn không thành công  2b2aa. Người dùng chọn lại lệnh từ chối lời mời kết bạn  *Use case tiếp tục bước 2b2*  2b2ab. Người dùng không chọn lại lệnh từ chối lời mời kết bạn  *Use case kết thúc*  2c2a. Hệ thống xác thực lệnh xóa lời mời kết bạn không thành công  2c2aa. Người dùng chọn lại lệnh xóa lời mời kết bạn  *Use case tiếp tục bước 2c2*  2c2ab. Người dùng không chọn lại lệnh xóa lời mời kết bạn  *Use case kết thúc*  2d2a. Hệ thống xác thực lệnh hủy kết bạn không thành công  2d2aa. Người dùng chọn lại lệnh hủy kết bạn  *Use case tiếp tục bước 2d2*  2d2ab. Người dùng không chọn lại lệnh hủy kết bạn  *Use case kết thúc* |

1. **Use case cập nhật thông tin cá nhân**

|  |  |
| --- | --- |
| **Tên use case** | Cập nhật thông tin cá nhân |
| **Mô tả** | Người dùng muốn cập nhật thông tin cá nhân trong ứng dụng |
| **Tác nhân chính** | Người dùng |
| **Mức** | Không bắt buộc |
| **Kích hoạt** | Người dùng chọn lệnh cập nhật thông tin cá nhân |
| **Tiền điều kiện** | * Người dùng phải đăng nhập vào ứng dụng * Thiết bị của người dùng đã được kết nối internet khi thực hiện cập nhật thông tin cá nhân |
| **Đảm bảo tối thiểu** | Hệ thống thông báo cập nhật thông tin cá nhân thất bại |
| **Đảm bảo thành công** | Hệ thống cập nhật thành công thông tin cá nhân của người dùng |
| **Chuỗi sự kiện chính** | 1. Người dùng truy cập vào trang thông tin cá nhân  2. Người dùng cung cấp thông tin cập nhật vào form thông tin cá nhân  3. Người dùng chọn lệnh cập nhật thông tin cá nhân  4. Hệ thống xác thực thông tin cập nhật thành công và cập nhật CSDL  5. Hệ thống thông báo cập nhật thành công |
| **Chuỗi sự kiện thay thế** | 1a. Người dùng truy cập vào trang đổi mật khẩu  1a1. Người dùng cung cấp mật khẩu mới  1a2. Người dùng chọn lệnh cập nhật mật khẩu  *Use case tiếp tục bước 4*  1b. Người dùng truy cập vào trang đổi tên url  1b1. Người dùng cung cấp tên url mới  1b2. Người dùng chọn lệnh cập nhật tên url  *Use case tiếp tục bước 4* |
| **Chuỗi sự kiện ngoại lệ** | 4a. Hệ thống xác thực thông tin cập nhật không thành công và hiển thị thông báo lỗi  4aa. Người dùng nhập lại thông tin cập nhật  *Use case* tiếp *tục bước 2*  4ab. Người dùng không nhập lại thông tin cập nhật và thoát khỏi hệ thống  *Use* case *kết thúc* |

1. **Sơ đồ hoạt động**
2. **Use case đăng ký tài khoản**

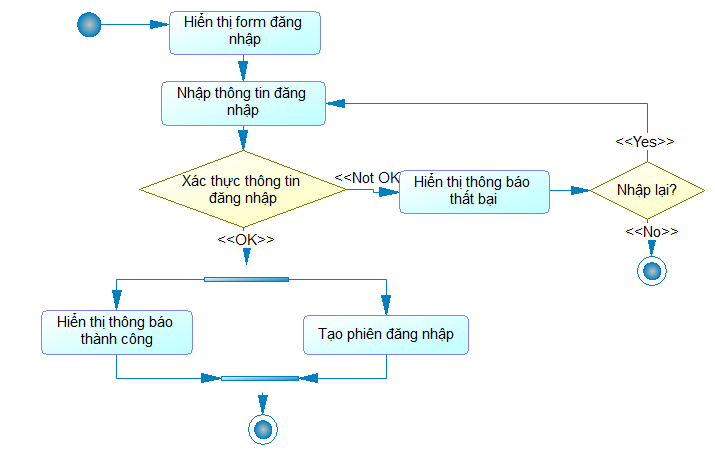


1. **Use case xác nhận tài khoản**

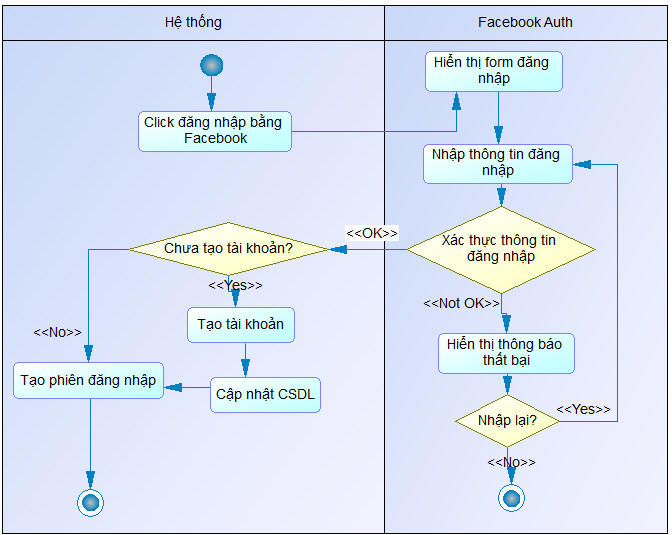
Diagram

Description automatically generated

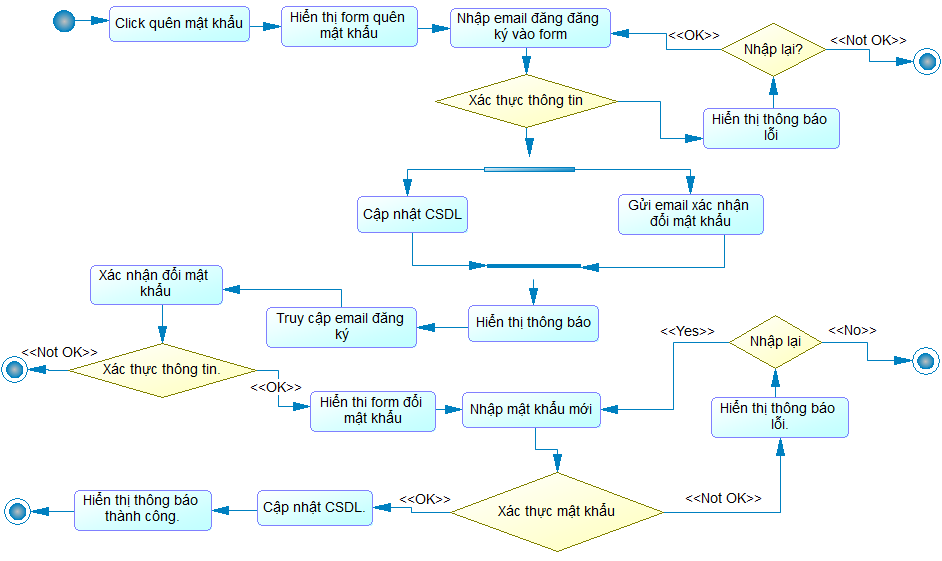
1. **Use case đăng nhập**



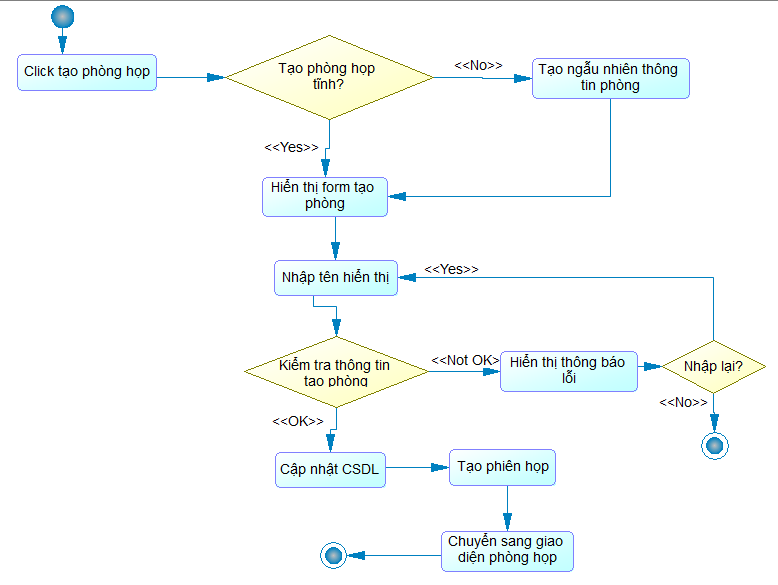
1. **Use case đăng nhập bằng Facebook**



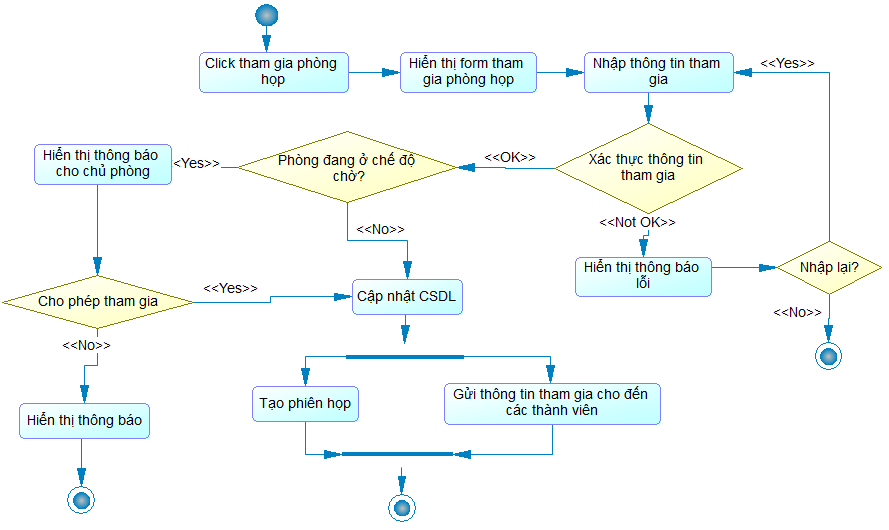
1. **Use case quên mật khẩu**



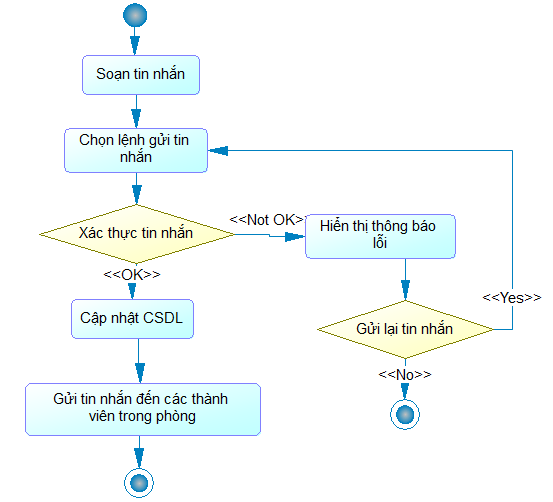
1. **Use case tạo phòng họp**



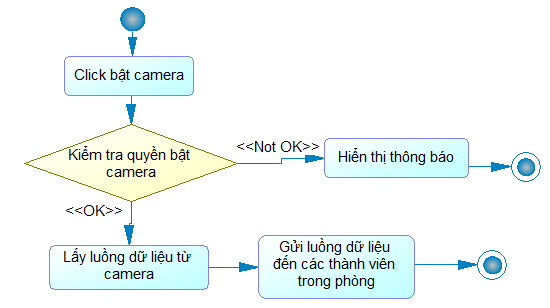
1. **Use case tham gia phòng họp**



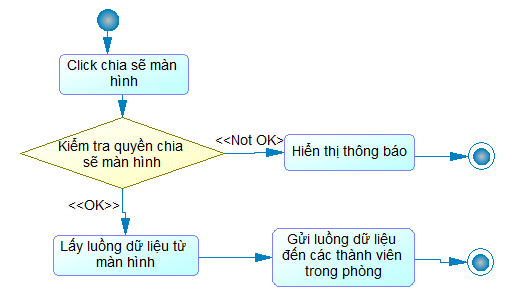
1. **Use case nhắn tin trong phòng phòng họp**



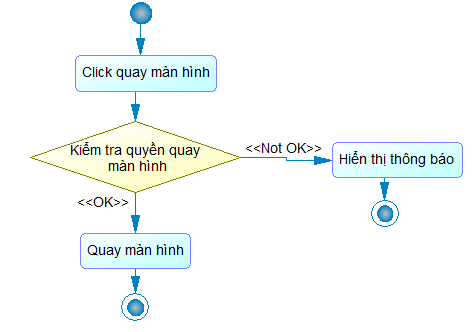
1. **Use case bật camera**



1. **Use case Chia sẻ màn hình**



1. **Use case quay màn hình**



1. **Use case quay tạo văn bản**

Diagram

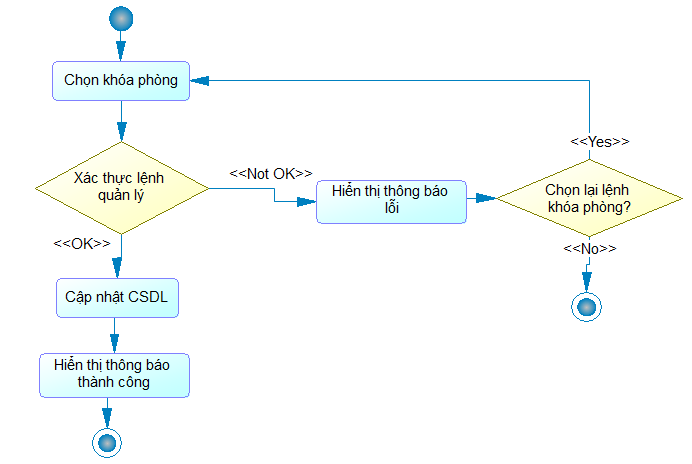
Description automatically generated

1. **Use case giơ tay**

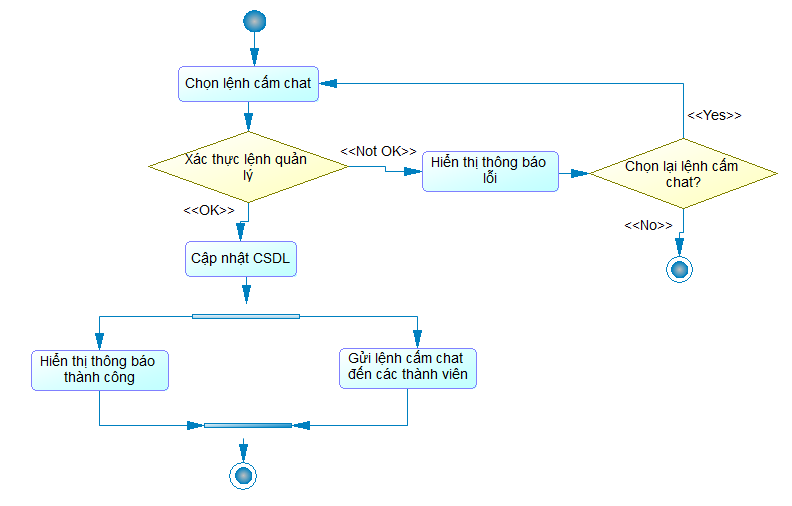
Diagram

Description automatically generated

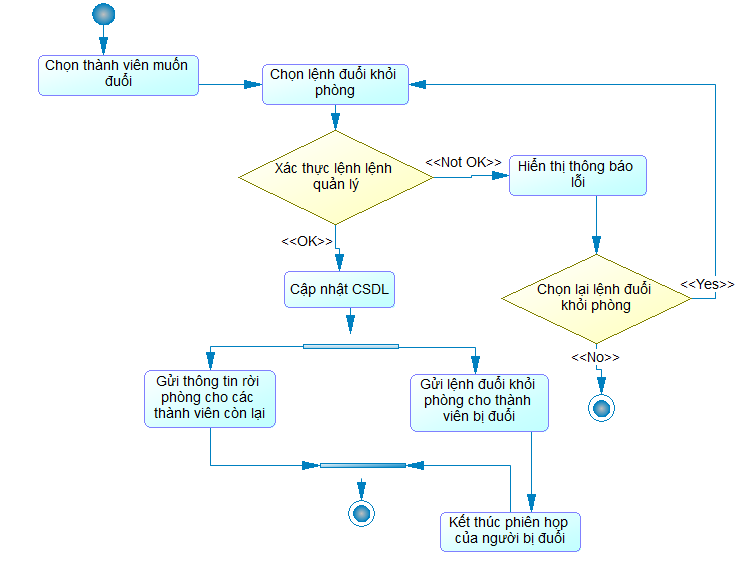
1. **Use case khóa phòng họp**



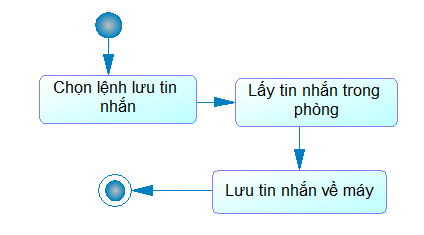
1. **Use case cấm chat**



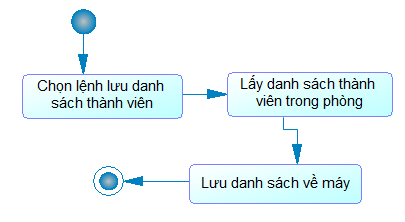
1. **Use case đuổi thành viên khỏi phòng**



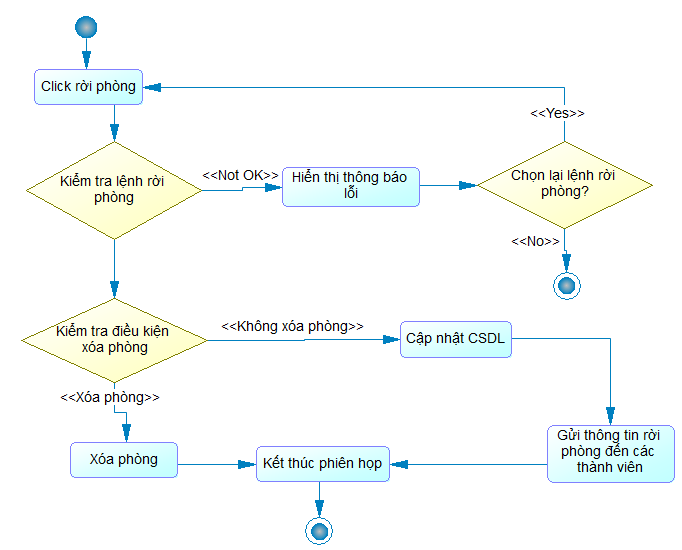
1. **Use case lưu tin nhắn**



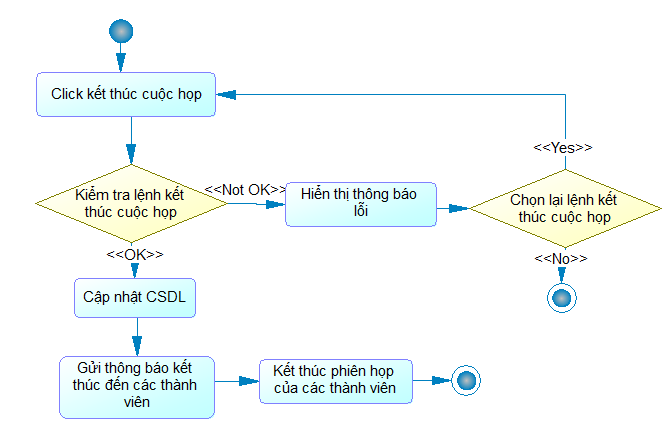
1. **Use case lưu danh sách thành viên trong phòng**



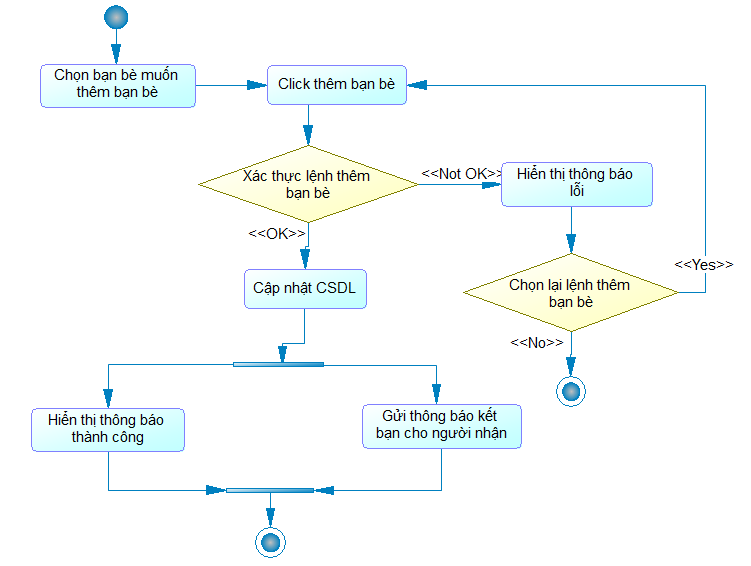
1. **Use case rời phòng họp**



1. **Use case kết thúc phòng họp**



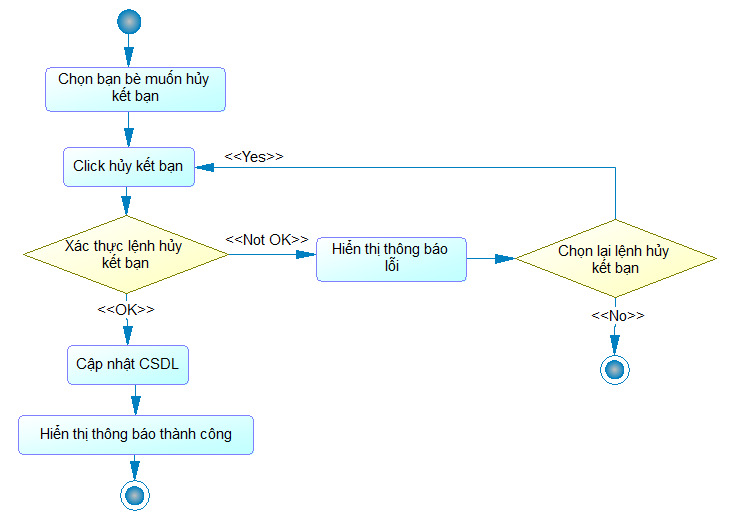
1. **Use case gửi lời mời kết bạn**



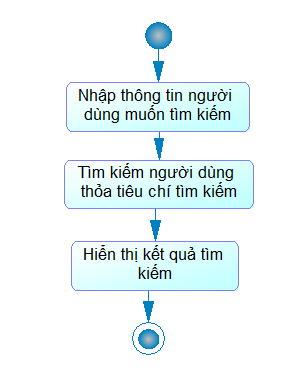
1. **Use case chấp nhận lời mời kết bạn**



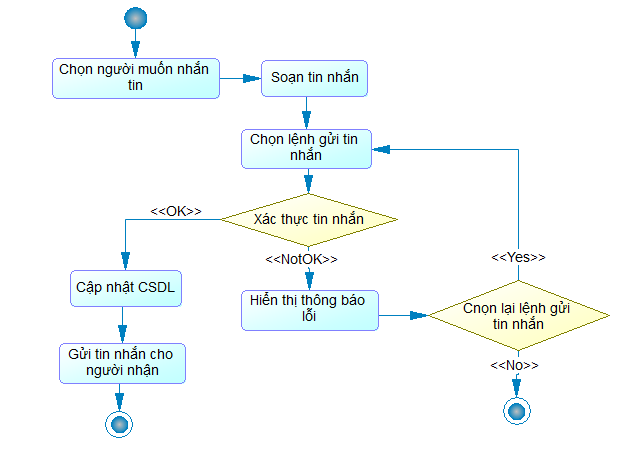
1. **Use case hủy kết bạn**



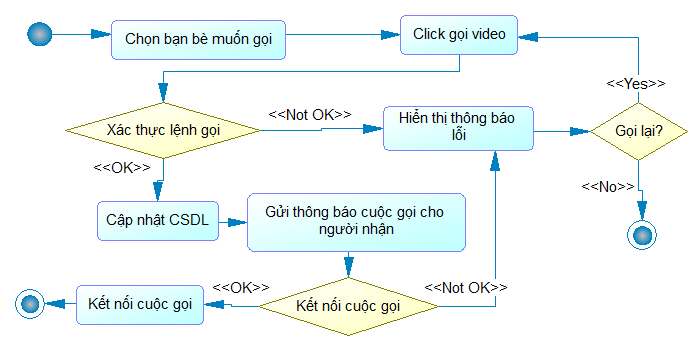
1. **Use case tìm kiếm bạn bè**



1. **Use case nhắn tin với bạn bè**



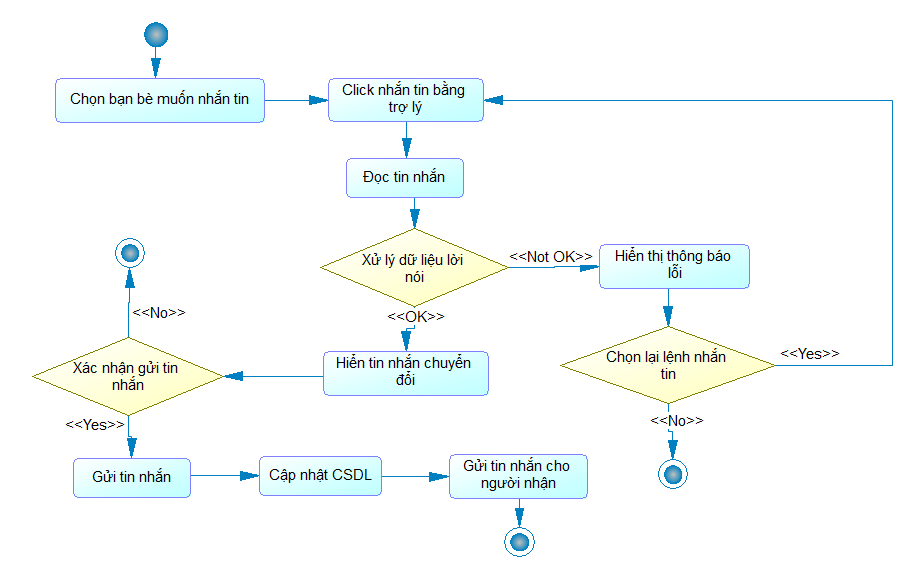
1. **Use case gọi video**



1. **Use case nhắn tin bằng chụp ảnh**



1. **Use case nhắn tin bằng trợ lý ảo**



1. **Use case cập nhật thông tin cá nhân**

Diagram

Description automatically generated

1. **Sơ đồ tuần tự**
2. **Use case đăng ký tài khoản**

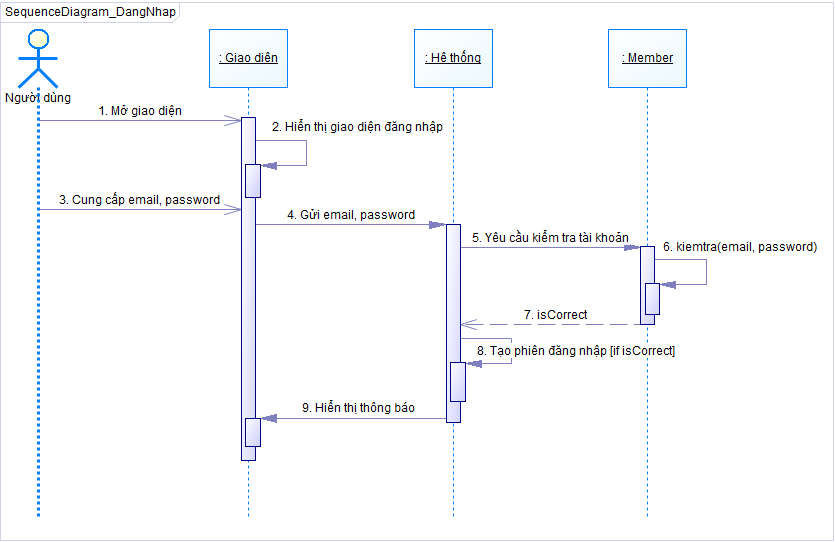


Mô tả: Người dùng mở giao diện đăng ký tài khoản mới và cung cấp thông tin đăng ký tài khoản, thông tin đăng ký bao gồm: họ tên, email, mật khẩu và xác nhận mật khẩu. Người dùng nhấn vào nút đăng đăng ký. Hệ thống xác thực thông tin đăng ký của người dùng. Quy định xác thực bao gồm:

* Trường họ tên: Không được rỗng, phải là chuỗi, không dài quá 40 ký tự, không chứa các ký tự đặc biệt (trừ - và \_).
* Trường email: Không được rỗng, đúng định dạng email, email chưa được tài khoản khác sử dụng.
* Trường mật khẩu: Không được rỗng, dài ít nhất 6 ký tự.
* Trường xác nhận mật khẩu: Phải khớp với trường mật khẩu.

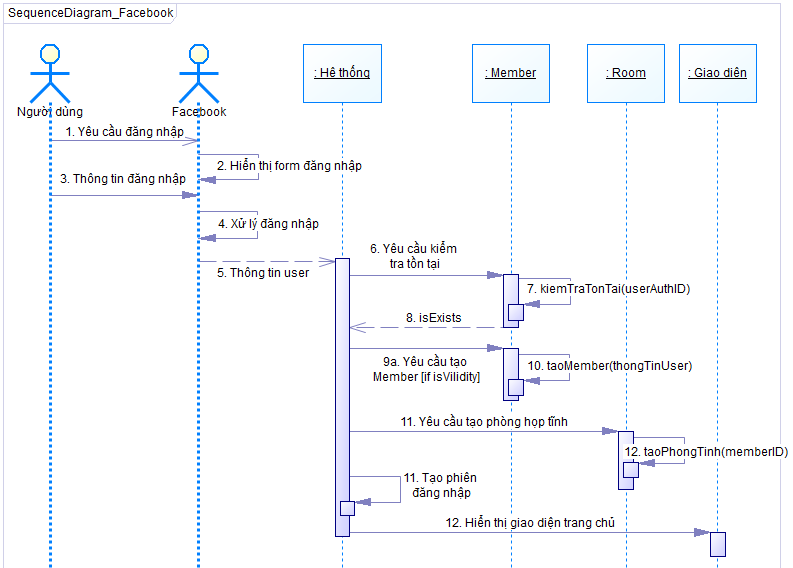
Nếu thông tin đăng ký của người dùng không thỏa các quy định trên thì hiển thị thông báo lỗi. Ngược lại, hệ thống tạo tài khoản mới và tạo phòng họp tĩnh cho tài khoản này. Phòng họp tĩnh bao gồm số phòng và mật khẩu được sinh ngẫu nhiên. Số phòng là một chuỗi gồm 9 chữ số, mật khẩu phòng là một chuỗi 4 chữ số. Số phòng không thể thay đổi, mật khẩu phòng có thể thay đổi bởi người dùng. Sau khi tạo tài khoản, hệ thống gửi một email xác nhận kích hoạt tài khoản vào email vừa đăng ký và thông báo đăng ký tài khoản thành công. Người dùng không thể đăng nhập vào ứng dụng bằng tài khoản chưa được kích hoạt. Người dùng phải truy cập vào email đã dùng để đăng ký tài khoản để kích hoạt tài khoản. Yêu cầu kích hoạt tài khoản được gửi đến hệ thống. Hệ thống xác nhận thông tin kích hoạt và kích hoạt tài khoản. Hệ thống hiển thị thông báo kích hoạt tài khoản thành công. Hệ thống chuyển sang giao diện đăng nhập.

1. **Use case đăng nhập**



Mô tả: Người dùng mở giao diện đăng nhập và nhập thông tin đăng nhập vào form đăng nhập. Thông tin đăng nhập bao gồm: email và mật khẩu. Người dùng nhấp nút đăng nhập. Thông tin đăng nhập được gửi đến hệ thống. Hệ thống kiểm tra thông tin đăng nhập. Nếu thông tin đăng nhập không hợp lệ, hệ thống hiển thị thông báo đăng nhập không thành công. Ngược lại, hệ thống tạo phiên đăng nhập cho người dùng và cho phép người dùng truy cập vào ứng dụng.

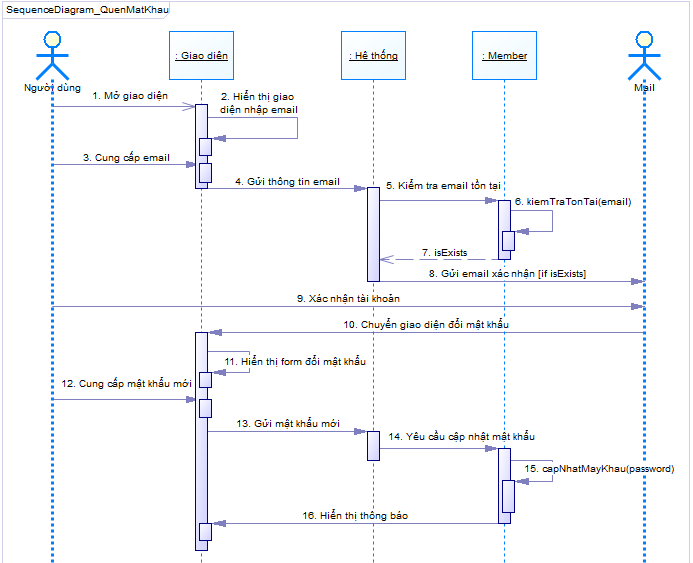
1. **Use case đăng nhập bằng Facebook**



Mô tả: Người dùng có thể đăng nhập vào ứng dụng bằng tài khoản Facebook. Người dùng chọn chức năng đăng nhập bằng Facebook. Hệ thống chuyển sang giao diện đăng nhập của Facebook. Người dùng nhập thông tin đăng nhập của Facebook. Nếu thông tin đăng nhập hợp hệ, hệ thống kiểm tra tài khoản Facebook này đã được sử dụng để đăng nhập vào ứng dụng hay chưa, nếu tài khoản chưa được sử dụng, hệ thống tạo tài khoản mới từ thông tin trả về của Facebook và tạo phòng họp tĩnh (tương tự use case đăng ký tài khoản). Tài khoản loại này không cần phải kích hoạt, không thể đổi mật khẩu. Hệ thống tạo phiên đăng nhập và cho phép người dùng truy cập vào ứng dụng.

Hệ thống cũng cho phép người dùng sử dụng tài khoản Google để đăng nhập vào ứng dụng. Các bước thực hiện tương tự như việc đăng nhập bằng Facebook.

1. **Use case quên mật khẩu**

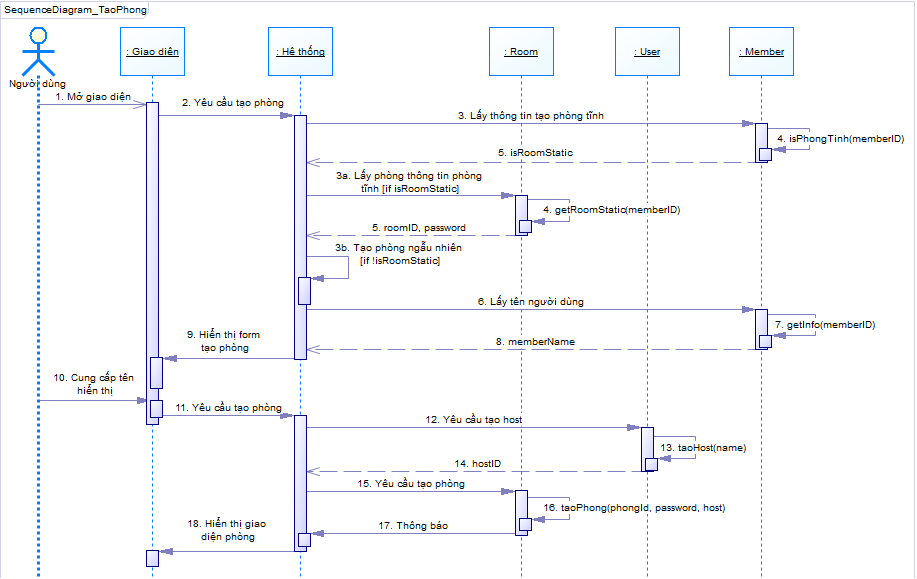


Mô tả: Người dùng chọn chức năng quên mật khẩu. Hệ thống yêu cầu cung cấp email đăng ký tài khoản để xác thực việc lấy lại mật khẩu. Người dùng cung cấp email vào form và nhấn nút submit. Hệ thống xác nhận email người dùng cung cấp có tồn tại trong ứng dụng hay không (những tài khoản được tạo bằng cách đăng nhập bằng Facebook và Google sẽ không có chức năng lấy lại mật khẩu). Nếu email không tồn tại, hệ thống sẽ thông báo cho người dùng. Ngược lại, hệ thống tạo mã xác thực cho tài khoản và gửi một email đến email vừa nhập để xác nhận việc lấy lại mật khẩu. Người dùng cần truy cập vào email đăng ký tài khoản để xác nhận việc lấy lại mật khẩu. Hệ thống kiểm tra mã xác thực và hiển thị giao diện đổi mật khẩu (hệ thống không có khả năng lấy lại mật khẩu trước đó). Người dùng nhập thông tin đổi mật khẩu vào form đổi mật khẩu. Thông tin bao gồm: mật khẩu mới và xác nhận mật khẩu. Hệ thống kiểm tra các quy định về thông tin đổi mật khẩu đối với thông tin mà người dùng cung cấp. Quy định bao gồm:

* Trường mật khẩu mới: Không được rỗng, dài ít nhất 6 ký tự.
* Trường xác nhận mật khẩu: Phải khớp với trường mật khẩu mới.

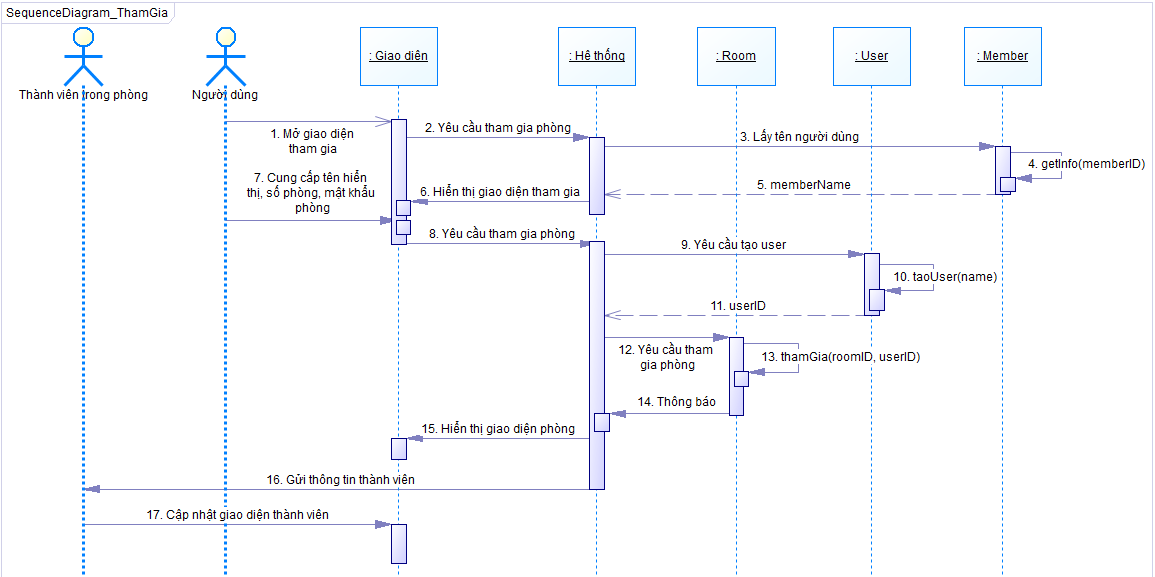
Nếu thông tin đổi không thỏa các quy định trên, hệ thống sẽ thông báo cho người dùng. Ngược lại, hệ thống cập nhật lại mật khẩu mới cho tài khoản và thông báo cho người dùng.

1. **Use case tạo phòng họp**



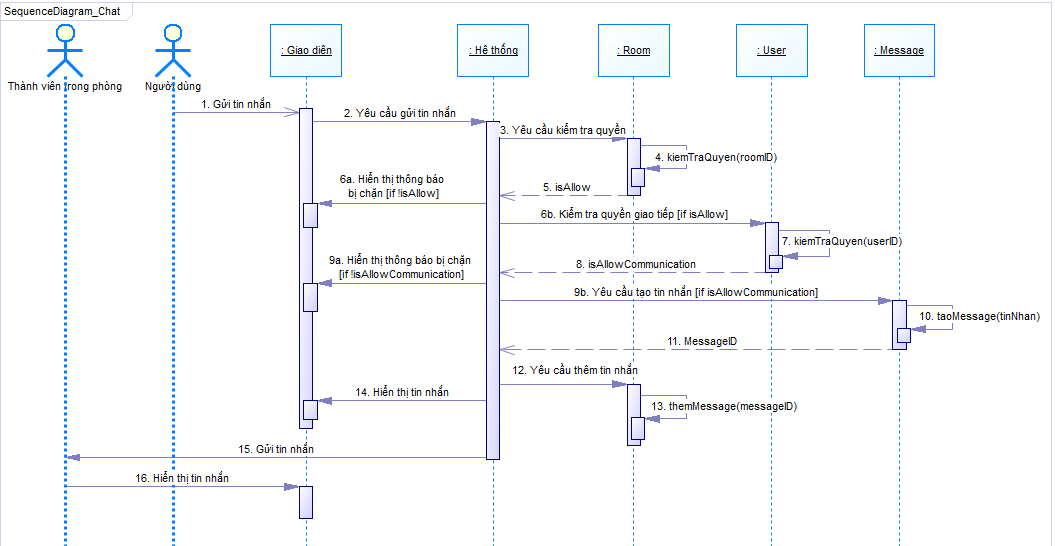
Mô tả: Người dùng mở giao diện tạo phòng họp mới. Form tạo phòng bao gồm: tên hiển thị trong phòng, số phòng, mật khẩu phòng. Tất cả trường đều không được bỏ trống. Nếu người dùng đang ở phiên đăng nhập, hệ thống lấy tên người dùng làm tên hiển thị trong phòng họp, hình đại diện người dùng là hình đại diện trong phòng họp và hệ thống kiểm tra người dùng có đang sử dụng tạo phòng họp tĩnh. Nếu có hệ thống lấy thông tin phòng họp tĩnh (gồm số phòng và mật khẩu phòng) điền vào form tạo phòng. Ngược lại, hệ thống sinh ngẫu nhiên số phòng và mật khẩu phòng, hình đại diện sẽ được lấy hình ảnh mặt định. Số phòng là chuỗi 9 chữ số, mật khẩu là chuỗi 4 chữ số. Tên người dùng có thể được thay đổi, số phòng và mật khẩu phòng không thể thay đổi. Người dùng nhấn nút tạo phòng. Hệ thống xác thực thông tin tạo phòng, nếu thông tin không hợp lệ hệ thống sẽ thông báo cho người dùng. Ngược lại, hệ thống tạo phòng họp và tạo chủ phòng. Hệ thống tạo phiên họp cho chủ phòng, thông báo thành công và chuyển sang giao diện phòng họp dành cho chủ phòng.

1. **Use case tham gia phòng họp**



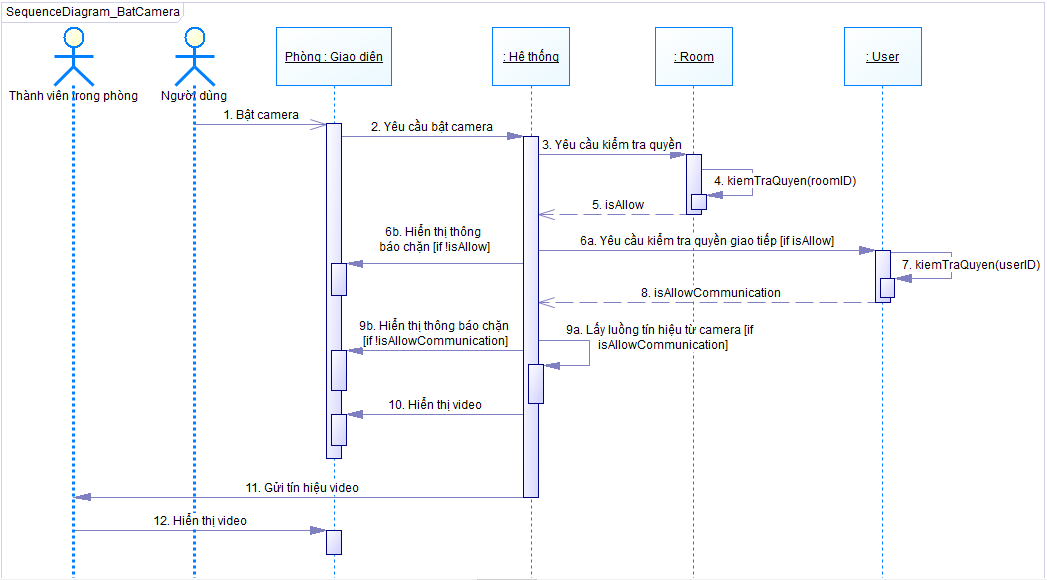
Mô tả: Người dùng mở giao diện tham gia phòng họp. Form tham gia phòng bao gồm: tên hiển thị trong phòng, số phòng và mật khẩu phòng. Tất cả trường không được bỏ trống. Nếu người dùng đang trong phiên đăng nhập, hệ thống sẽ lấy tên người dùng làm tên hiển thị trong phòng, hình đại diện người dùng làm hình đại diện trong phòng họp. Người dùng cung cấp thông tin tham gia phòng vào form tham gia và nhấn nút tham gia phòng. Hệ thống kiểm tra thông tin tham gia phòng (phòng tồn tại). Nếu thông tin không hợp lệ, hệ thống thông báo lỗi cho người dùng. Ngược lại, nếu phòng đang ở trạng thái khóa, hệ thống sẽ thông báo cho người dùng. Nếu phòng đang ở trạng thái mở, hệ thống tạo mới thành viên và cho thành viên vào phòng. Khi có thành viên vào phòng, hệ thống sẽ cập nhật lại danh sách thành viên và gửi danh sách cho các thành viên trong phòng. Nếu phòng đang ở trạng thái phòng chờ, hệ thống sẽ tạo mới thành viên và cho thành viên vào phòng chờ. Hệ thống sẽ thông báo cho chủ phòng. Chủ phòng có quyền cho phép hoặc không cho phép thành viên trong phòng chờ vào phòng. Nếu chủ phòng không cho phép, hệ thống sẽ xóa thành viên yêu cầu tham gia và thông báo cho người yêu cầu tham gia. Ngược lại, hệ thống cho thành viên yêu cầu vào phòng.

1. **Use case nhắn tin trong phòng phòng họp**



Mô tả: Các thành viên trong phòng chỉ có thể nhắn tin khi được phép (phòng đang cho phép nhắn tin, bản thân không bị cấm giao tiếp bởi chủ phòng) trừ chủ phòng. Người dùng soạn tin nhắn và chọn gửi tin nhắn. Hệ thống kiểm tra quyền nhắn tin của người nhắn. Nếu người nhắn có quyền nhắn tin, hệ thống sẽ tạo tin nhắn mới với nội dung tin nhắn. Hệ thống gửi tin nhắn đến các thành viên trong phòng. Người dùng có thể gửi tệp đính kèm cho các thành viên trong phòng. Hệ thống cho phép gửi nhiều tệp cùng một lúc (tối đa 5 tệp tin) và kích thước mỗi tập tin không quá 1MB. Khi người dùng gửi tệp, hệ thống kiểm tra tệp có thỏa các quy định về gửi tệp tin ở trên. Nếu không thỏa quy định hệ thống sẽ thông báo cho người dùng. Ngược lại hệ thống tải tệp lưu trữ trên đám mây và gửi đường dẫn đến tệp cho các thành viên trong phòng. Hệ thống cung cấp một số biểu tượng cảm xúc phục vụ cho việc nhắn tin.

1. **Use case bật camera**



Mô tả: Các thành viên trong phòng chỉ có thể bật camera khi được phép (phòng đang cho phép bật camera, bản thân không bị cấm giao tiếp bởi chủ phòng) trừ chủ phòng. Người dùng chọn bật camera, hệ thống kiểm tra quyền bật camera của người yêu cầu. Nếu người yêu cầu có quyền bật camera, hệ thống sẽ yêu cầu người dùng cung cấp quyền sử dụng camera của thiết bị. Nếu người dùng không cho phép, hệ thống sẽ thông báo cho người dùng, chức năng kết thúc. Ngược lại, hệ thống lấy luồng dữ liệu video từ camera của thiết bị người dùng, hiển thị luồng dữ liệu ra giao diện người dùng và đồng thời hệ thống gửi luồng dữ liệu video đến các thành viên trong phòng. Khi các thành viên trong phòng nhận được luồng dữ liệu video của một thành viên nào đó trong phòng gửi đến, hệ thống sẽ hiển thị ra giao diện thành viên đó. Khi người dùng chọn tắt camera, hệ thống sẽ tắt hiển thị video và đồng thời gửi tín hiệu tắt camera đến các thành viên trong phòng. Khi các thành viên trong phòng nhận được tín hiệu tắt camera của một thành viên, hệ thống sẽ tắt video của người yêu cầu tương ứng.

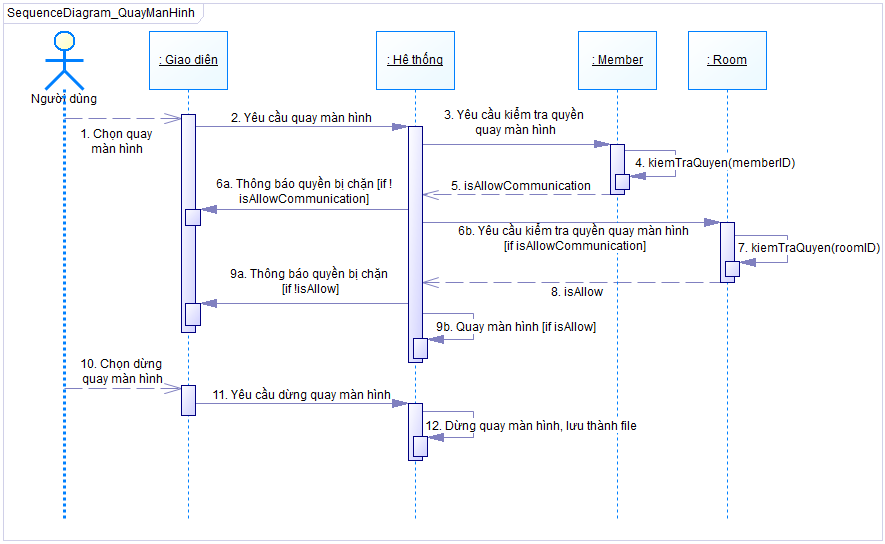
Hệ thống xử lý tương tự cho chức năng bật và tắt microphone. Hệ thống sẽ lấy luồng dữ liệu âm thanh từ microphone của thiết bị người dùng.

1. **Use case chia sẻ màn hình**



Mô tả: Các thành viên trong phòng chỉ có Chia sẻ màn hình khi được phép (phòng đang cho phép Chia sẻ màn hình, bản thân không bị cấm giao tiếp bởi chủ phòng (trừ chủ phòng) và trong phòng hiện không có thành viên nào đang Chia sẻ màn hình). Người dùng chọn chức năng Chia sẻ màn hình. Hệ thống kiểm tra quyền Chia sẻ màn hình của thành viên yêu cầu. Nếu người yêu cầu không có quyền Chia sẻ màn hình, hệ thống sẽ thông báo cho người dùng. Ngược lại, hệ thống yêu cầu người dùng cung cấp quyền truy cập vào dữ liệu màn hình. Nếu người dùng không chấp nhận yêu cầu, hệ thống sẽ thông báo cho người dùng, chức năng kết thúc. Ngược lại, hệ thống lấy luồng dữ liệu từ màn hình thiết bị người dùng, hiển thị giao diện Chia sẻ màn hình và đồng thời gửi luồng dữ liệu màn hình đến các thành viên trong phòng. Khi các thành viên trong phòng nhận được luồng dữ liệu Chia sẻ màn hình của một thành viên trong phòng, hệ thống sẽ hiển thị giao diện Chia sẻ màn hình tương ứng. Khi người dùng chọn tắt Chia sẻ màn hình, hệ thống sẽ tắt hiển thị giao diện Chia sẻ màn hình và đồng thời gửi tín hiệu tắt Chia sẻ màn hình đến các thành viên trong phòng. Khi các thành viên trong phòng nhận được tín hiệu tắt Chia sẻ của một thành viên, hệ thống sẽ tắt giao diện Chia sẻ của người yêu cầu.

1. **Use case quay màn hình**



Mô tả: Các thành viên trong phòng chỉ có thể quay màn hình khi được phép (phòng đang cho phép quay màn hình, bản thân không bị cấm giao tiếp bởi chủ phòng) trừ chủ phòng. Người dùng chọn quay màn hình, hệ thống kiểm tra quyền quay màn hình của người yêu cầu. Nếu người yêu cầu không có quyền quay màn hình, hệ thống sẽ thông báo cho người dùng. Ngược lại, hệ thống yêu cầu người dùng cung cấp quyền truy cập vào dữ liệu màn hình. Nếu người dùng không phép truy cập, hệ thống sẽ thông báo cho người dùng. Ngược lại, hệ thống tiến hành quay màn hình. Khi người dùng chọn dừng quay màn hình, hệ thống dừng quay màn hình và kết xuất kết quả quay màn hình ra tệp và tải về máy người dùng.

1. **Use case tạo văn bản**

Graphical user interface

Description automatically generated

Mô tả: Chỉ có người dùng đang trong phiên đăng nhập mới được phép tạo văn bản. Người dùng chọn tạo Text (file văn bản), hệ thống kiểm tra phiên đăng nhập của người dùng. Nếu người dùng không đang trong phiên đăng nhập, hệ thống sẽ thông báo cho người dùng. Ngược lại, hệ thống sẽ tạo Text và hiển thị giao diện soạn thảo văn bản đồng thời gửi đường dẫn đến tệp văn bản vừa tạo đến các thành viên trong phòng. Nội dung chỉnh sửa bao gồm nội dung văn bản, định dạng văn bản và tên văn bản. Tệp văn bản chỉ có thể chỉnh sửa bởi người tạo. Khi có người truy cập vào văn bản, thông tin chỉnh sửa văn bản cũng được gửi đến những người truy cập. Các tệp văn bản có thể được tìm lại trong trang cá nhân của người dùng và cho phép xóa đi.

1. **Use case giơ tay**

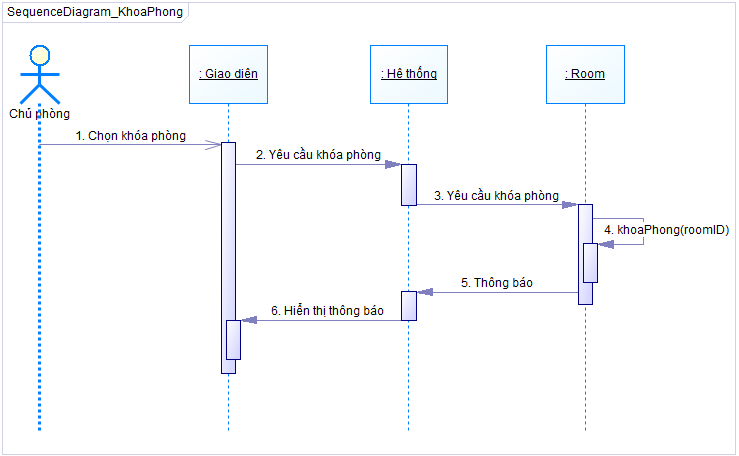
Graphical user interface, diagram, application

Description automatically generated

Mô tả: Người dùng chọn giơ tay, hệ thống cập nhật trạng thái người dùng thành đang giơ tay và gửi tín hiệu giơ tay đến các thành viên trong phòng. Hệ thống sẽ thông báo cho các thành viên trong phòng và sắp xếp lại danh sách thành viên theo sự ưu tiên dành cho thành viên đang giơ tay.

Người dùng cũng có thể chọn bỏ giơ tay, hệ thống cập nhật trạng thái người dùng thành không giơ tay và gửi tín hiệu bỏ giơ tay đến các thành viên trong phòng. Hệ sắp xếp lại danh sách thành viên trong phòng.

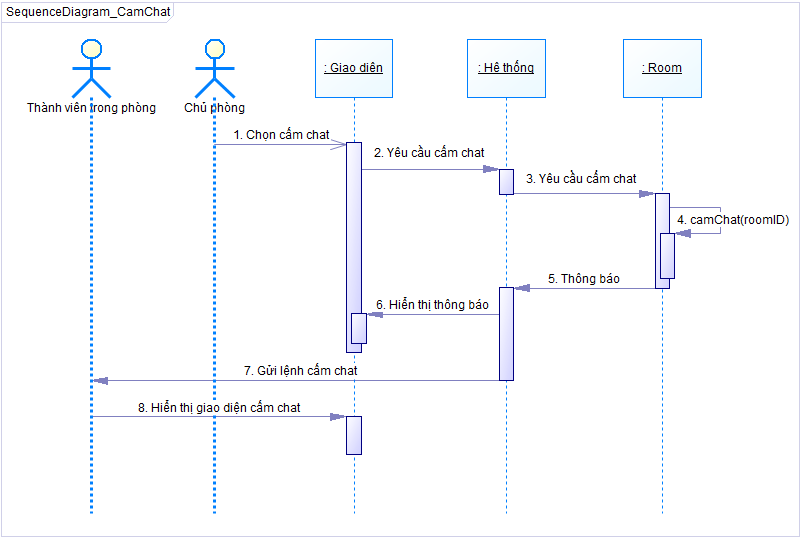
1. **Use case khóa phòng họp**



Mô tả: Chủ phòng chọn khóa phòng. Hệ thống sẽ khóa phòng lại. Người dùng không thể vào phòng khi phòng đang khóa.

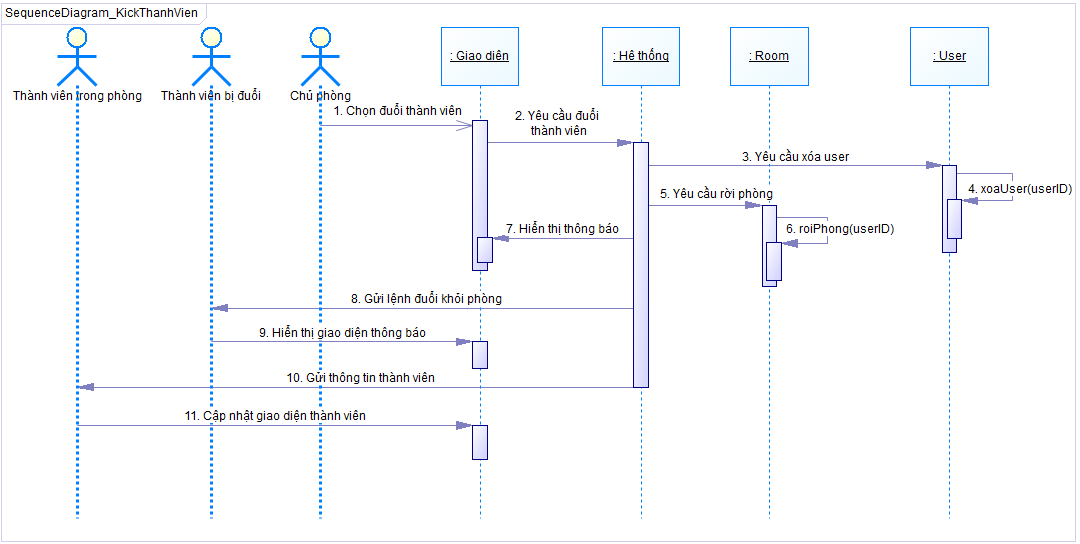
Hệ thống còn cho phép chủ mở phòng và đặt trạng thái phòng chờ. Ở chế độ mở phòng, bất kì ai cũng có thể vào phòng. Ở chế độ phòng chờ, hệ thống sẽ xử lý như use case tham gia phòng ở trạng thái phòng chờ.

1. **Use case cấm nhắn tin**



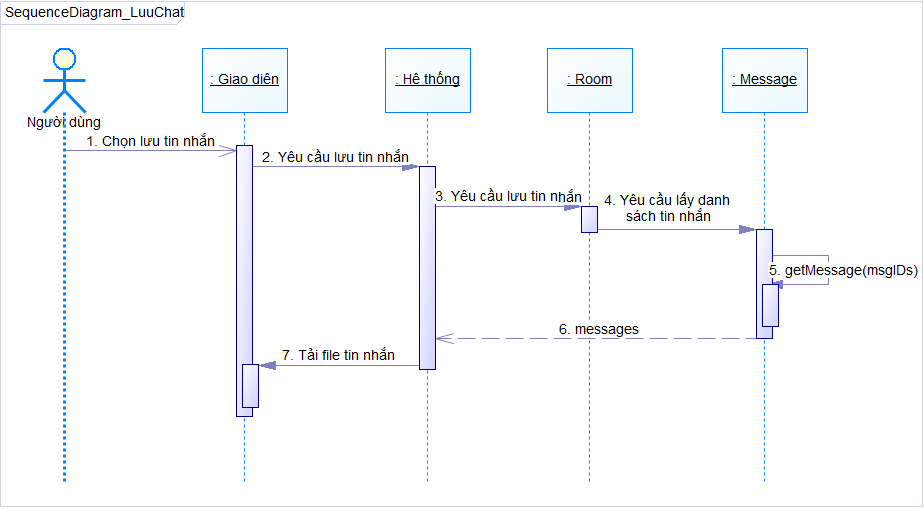
Mô tả: Chủ phòng chọn cấm nhắn tin. Hệ thống gửi lệnh cấm nhắn tin và thông báo lệnh cấm nhắn tin đến các thành viên trong phòng. Các thành viên trong phòng không thể nhắn tin khi phòng đang ở trạng thái cấm nhắn tin, trừ chủ phòng. Chủ phòng cũng có thể cho phép nhắn tin sau khi cấm, hệ thống sẽ gửi lệnh cho phép nhắn tin và thông báo đến các thành viên trong phòng. Chủ phòng cũng có thể cấm hoặc cho phép các thành viên trong phòng bật camera, bật microphone, Chia sẻ màn hình, quay màn hình. Các bước xử lý tương tự như chức năng cấm hoặc cho phép nhắn tin. Thêm vào đó, nếu thành viên đang thực chức năng bị cấm thì chức năng đó sẽ dừng lại ngay lập tức. Chủ phòng cũng có thể cấm việc giao tiếp của một thành viên cụ thể. Nội dung cấm giao tiếp bao gồm: cấm nhắn tin, cấm bật camera, cấm bật microphone, cấm Chia sẻ màn hình, cấm quay màn hình. Nếu thành viên bị chỉ định lệnh cấm đang thực hiện các chức năng trên thì hệ thống sẽ dừng chức năng đó lại và thông báo cho thành viên đó.

1. **Use case đuổi thành viên khỏi phòng**



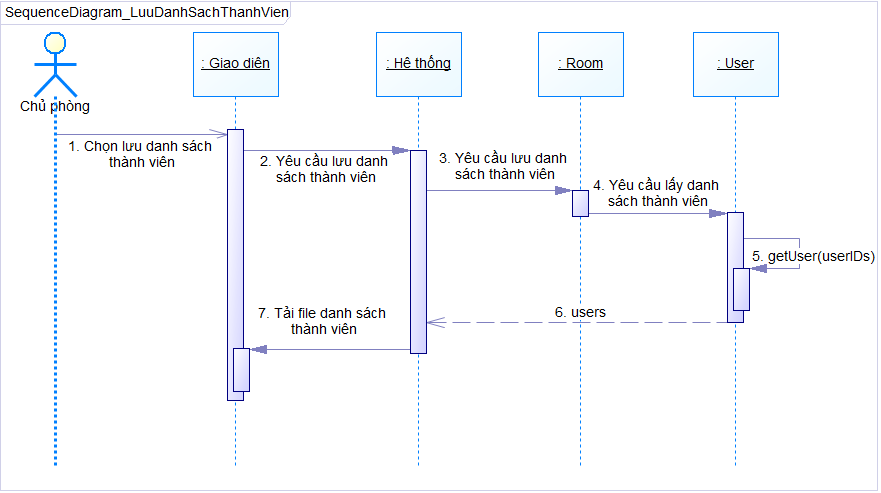
Mô tả: Chủ phòng chọn thành viên muốn đuổi khỏi phòng và chọn lệnh đuổi khỏi phòng. Hệ thống thực hiện xóa thành viên, xóa thành viên khỏi phòng và hiển thị thông báo cho chủ phòng. Hệ thống gửi lệnh đuổi khỏi phòng cho thành viên được chỉ định. Hệ thống thực hiện rời phòng họp cho thành viên đó và hiển thị thông báo đuổi khỏi phòng. Hệ thống thực hiện cập nhật giao diện danh sách thành viên cho các thành viên còn lại trong phòng.

1. **Use case lưu tin nhắn**



Mô tả: Các thành viên trong phòng chọn lưu tin nhắn. Hệ thống lấy ra danh sách tin nhắn trong phòng từ lúc phòng bắt đầu đến hiện tại kết xuất vào file và tải về thiết bị người dùng. Nội dung kết xuất gồm thời gian nhắn tin, tên người nhắn và nội dung tin nhắn.

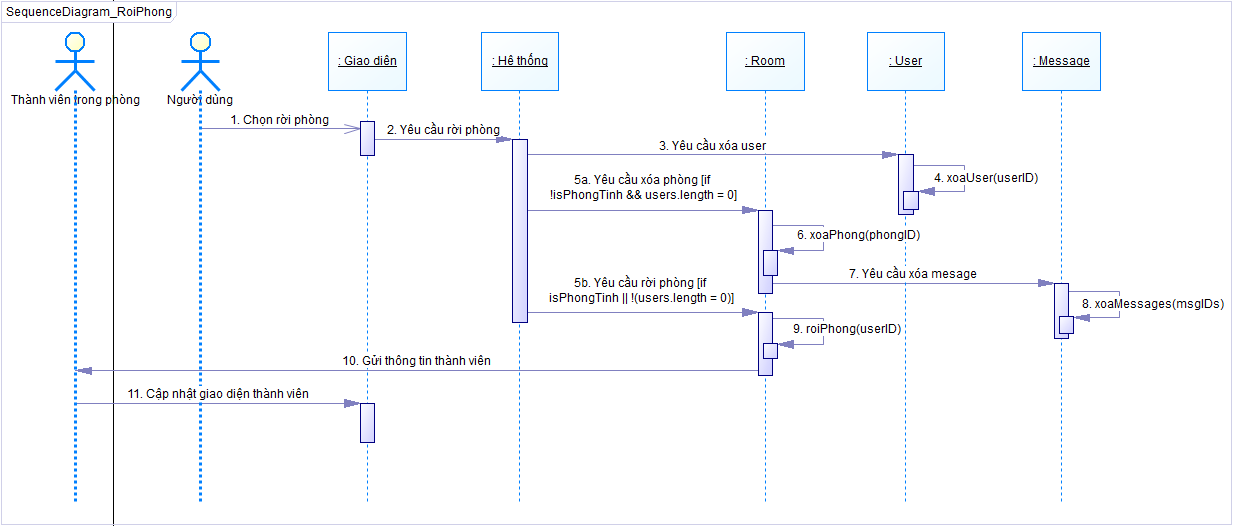
1. **Use case lưu danh sách thành viên trong phòng**



Mô tả: Chủ phòng chọn lưu danh sách thành viên trong phòng. Hệ thống lấy ra danh sách thành viên kể từ lúc phòng bắt đầu đến hiện tại kết xuất vào file và tải về thiết bị người dùng. Nội dung kết xuất gồm:

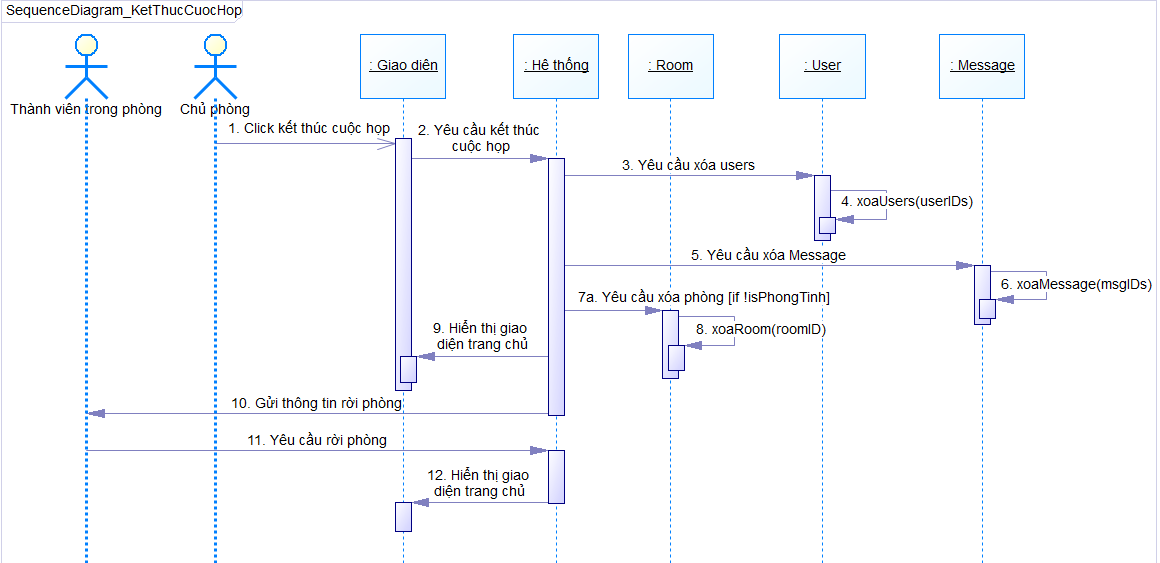
* Thông tin phòng họp: Số phòng, thời điểm bắt đầu, thời điểm kết thúc (thời điểm kết xuất), thời gian cuộc họp, tổng số thành viên tham gia.
* Danh sách thành viên tham gia: Tên thành viên, thời điểm vào, thời điểm ra (hoặc thời điểm kết xuất).

1. **Use case rời phòng họp**



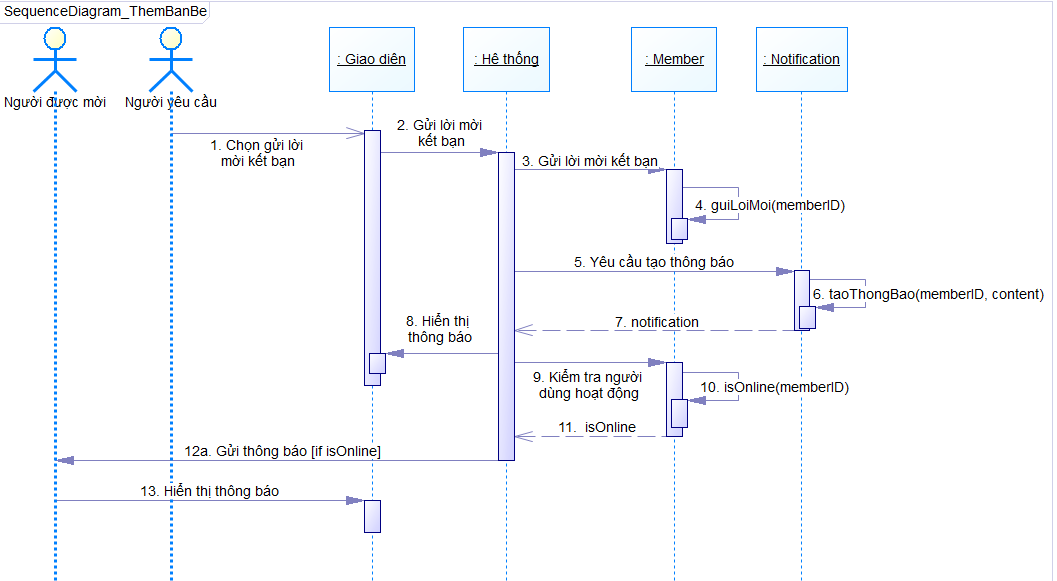
Mô tả: Người dùng chọn rời phòng. Hệ thống kết thúc phiên họp cho thành viên đó. Hệ thống kiểm tra điều kiện xóa phòng gồm: phòng không phải là phòng họp tĩnh, hiện không có thành viên nào trong phòng. Nếu thỏa điều kiện xóa phòng, hệ thống xóa danh sách thành viên trong phòng, xóa phòng, xóa danh sách tin nhắn trong phòng, xóa các tệp đính kèm trên đám mây liên quan đến phòng, xóa các thành viên hiện đang ở trong phòng chờ nếu phòng đang ở trạng thái phòng chờ và thông báo cho những người dùng đó. Ngược lại, nếu không thỏa điều kiện xóa phòng, hệ thống sẽ cập nhật danh sách thành viên và gửi đến đến các thành viên còn lại trong phòng và thông báo với họ.

1. **Use case kết thúc phòng họp**



Mô tả: Chủ phòng chọn kết thúc cuộc họp. Hệ thống kết thúc cuộc họp cho tất cả thành viên trong phòng. Hệ thống xóa danh sách thành viên trong phòng, xóa phòng, xóa danh sách tin nhắn trong phòng, xóa các tệp đính kèm trên đám mây liên quan đến phòng, xóa các thành viên hiện đang ở trong phòng chờ nếu phòng đang ở trạng thái phòng chờ và thông báo cho những người dùng đó. Hệ thống thông báo kết thúc cuộc họp đến các thành viên trong phòng.

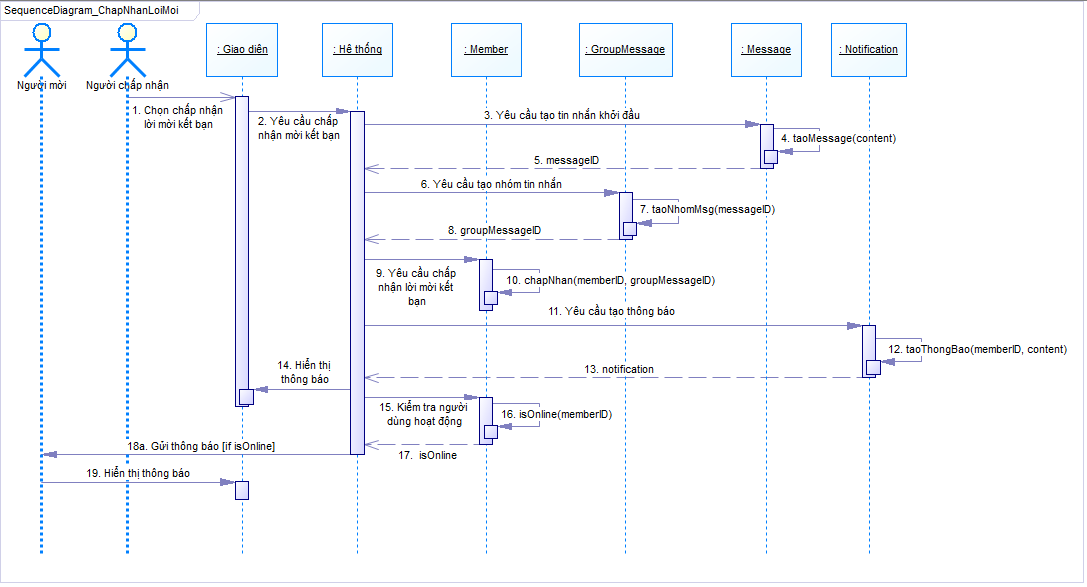
1. **Use case gửi lời mời kết bạn**



Mô tả: Người dùng chọn bạn bè và chọn thêm lời mời kết bạn. Hệ thống tạo yêu cầu kết bạn cho người dùng và tạo thông một thông báo kết bạn cho người được mời và hiển thị thông báo. Và nếu người được mời đang online, hệ thống sẽ gửi thông báo thời gian thực và hiển thị thông báo cho người đó. Đối với thông báo, người dùng có thể xóa và thay đổi trạng thái (đã đọc, chưa đọc).

Ngoài ra, người dùng có thể xóa lời mời kết bạn đã gửi. Hệ thống hiển thị thông báo cảnh báo. Nếu người dùng tiếp tục, hệ thống sẽ xóa lời mời kết bạn và yêu cầu kết bạn đối với người nhận lời mời. Hệ thống không tạo thông báo xóa lời mời kết bạn.

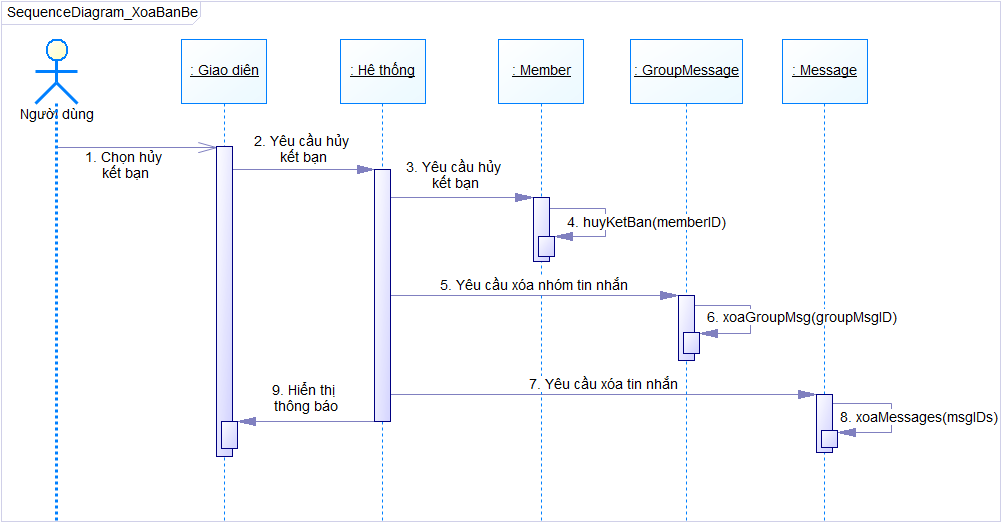
1. **Use case chấp nhận lời mời kết bạn**



Mô tả: Người dùng chọn chấp nhận lời mời kết bạn. Hệ thống thực hiện việc kết bạn bao gồm các bước: Xóa yêu cầu kết bạn và lời mời kết bạn đối với người gửi lời mời, tạo nhóm tin nhắn (đối tượng này chứa danh sách tin nhắn của hai người dùng), tạo tin nhắn khởi đầu cuộc trò chuyện, tạo thông báo chấp nhận yêu cầu kết bạn cho người gửi lời mời và gửi thông báo thời gian thực cho người đó nếu hệ thống kiểm tra họ đang trực tuyến. Hệ thống hiển thị thông báo cho người dùng.

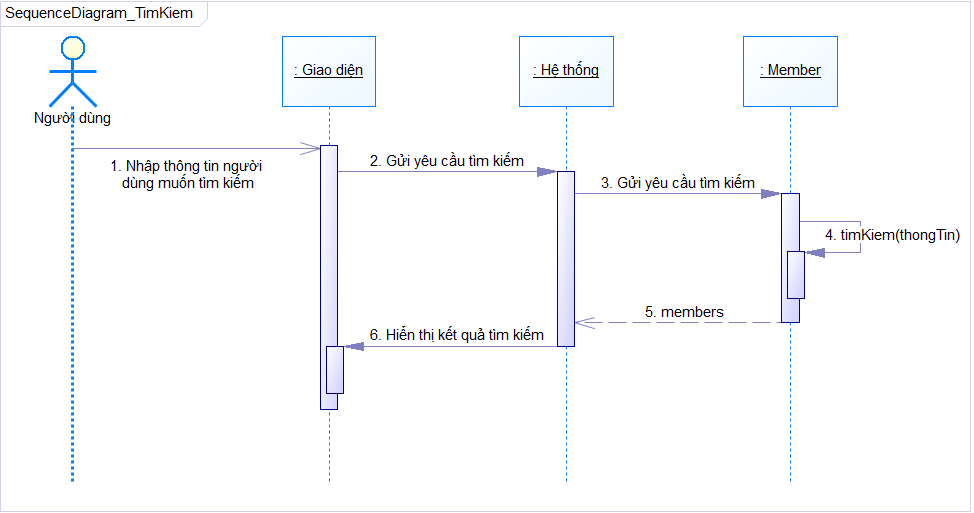
Ngoài ra, người dùng còn có thể không chấp nhận lời mời kết bạn. Hệ thống sẽ xóa yêu cầu kết bạn và lời mời kết bạn đối với người gửi lời mời. Hệ thống không tạo thông báo không chấp nhận lời mời kết bạn.

1. **Use case hủy kết bạn**



Mô tả: Người dùng chọn hủy kết bạn. Hệ thống hiển thị thông báo cảnh báo. Nếu người dùng tiếp tục, hệ thống thực hiện việc hủy bạn bè bao gồm: xóa bạn bè, xóa nhóm tin nhắn, xóa toàn bộ tin nhắn giữa hai người, xóa các tệp tin có liên quan đến tin nhắn của họ trên đám mây và hiển thị thông báo cho người dùng. Hệ thống không tạo thông báo hủy kết bạn.

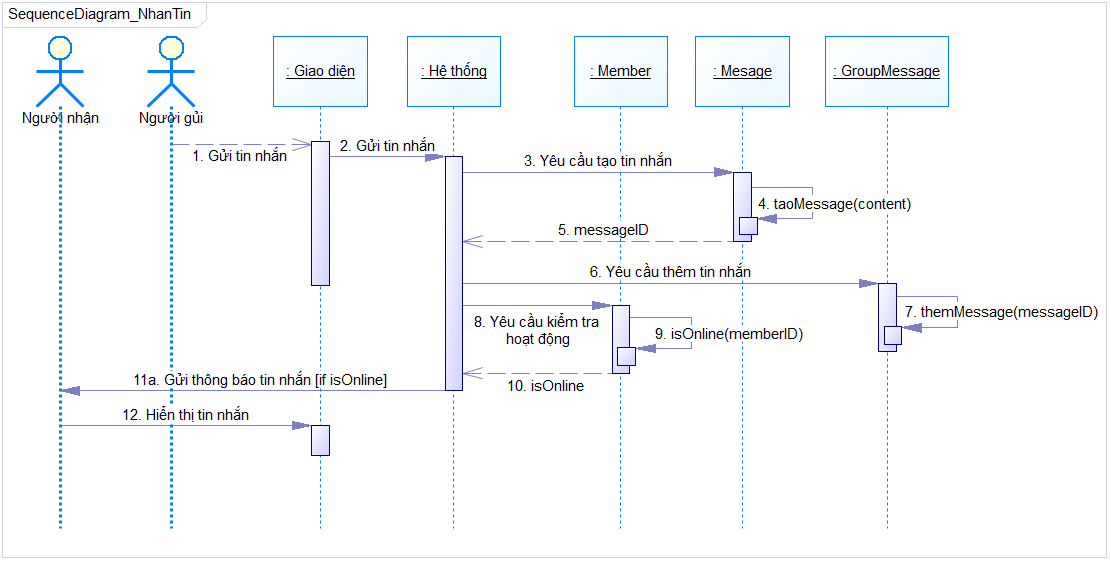
1. **Use case tìm kiếm bạn bè**



Mô tả: Người dùng nhập thông tin bạn bè muốn tìm kiếm vào form tìm kiếm bạn bè. Thông tin có thể là tên bạn bè, số điện thoại, địa chỉ của bạn bè. Hệ thống thực hiện tìm kiếm bạn bè từ thông tin người dùng cung cấp và hiển thị danh sách bạn bè. Có hai loại tìm kiếm:

* Tìm kiếm bạn bè.
* Tìm kiếm mọi người (kể cả bạn bè).

1. **Use case nhắn tin với bạn bè**

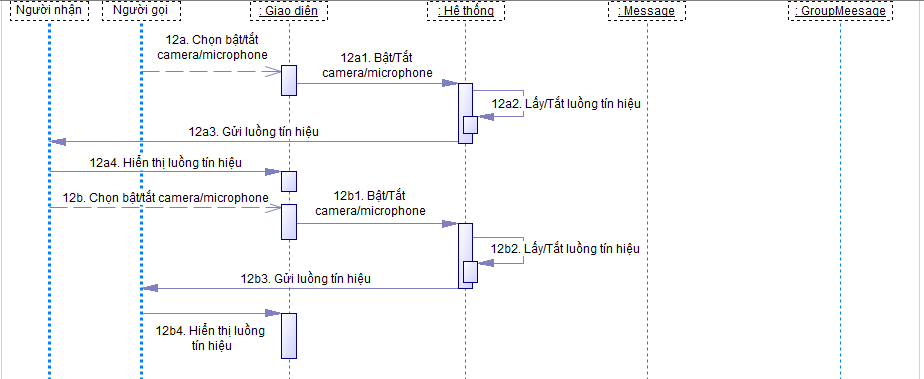


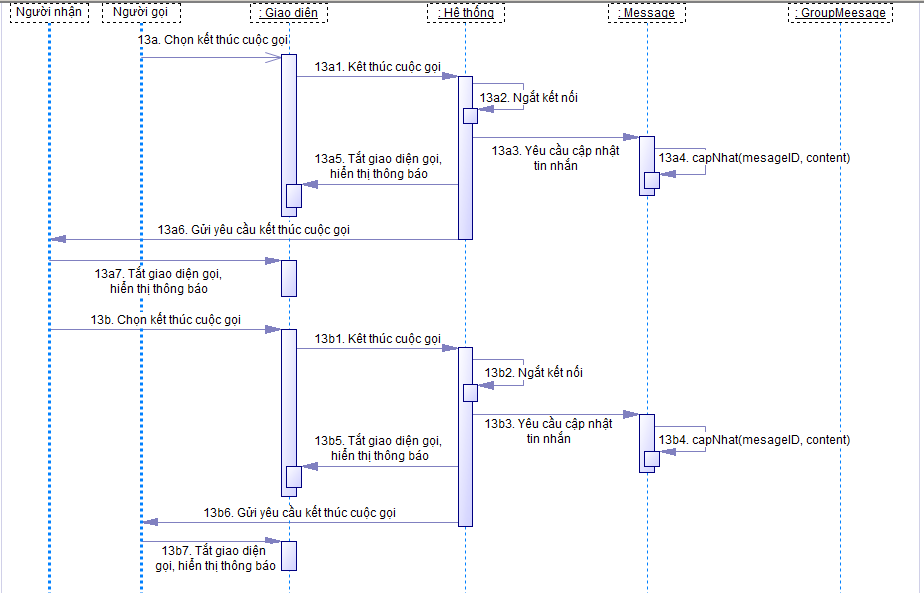
Mô tả: Người dùng soạn tin nhắn và chọn gửi tin nhắn. Hệ thống hiển thị tin nhắn phía người gửi và tạo tin nhắn từ nội dung tin nhắn và thêm tin nhắn vào nhóm tin nhắn. Hệ thống kiểm tra người nhận có đang trực tuyến không. Nếu có, hệ thống gửi tin nhắn thời gian thực và hiển thị tin nhắn cho người nhận.

Người dùng có thể gửi tệp đính kèm cho các thành viên trong phòng. Hệ thống cho phép gửi nhiều tệp cùng một lúc (tối đa 5 tệp tin) và kích thước mỗi tập tin không quá 1MB. Khi người dùng gửi tệp, hệ thống kiểm tra tệp có thỏa các quy định về gửi tệp tin ở trên. Nếu không thỏa quy định hệ thống sẽ thông báo cho người dùng. Ngược lại hệ thống tải tệp lưu trữ trên đám mây và gửi đường dẫn đến tệp cho các thành viên trong phòng. Hệ thống cung cấp một số biểu tượng cảm xúc phục vụ cho việc nhắn tin.

1. **Use case gọi video**







Mô tả:

* Giai đoạn kết nối cuộc gọi

Người dùng chọn gọi video cho bạn bè. Quy định về thực hiện cuộc gọi: người nhận đang trực tuyến và đang không trong bất kỳ cuộc gọi nào khác. Hệ thống kiểm tra quy định tạo cuộc gọi ở trên. Nếu không thỏa, hệ thống sẽ thông báo cho người gọi và use kết thúc. Ngược lại, hệ thống tạo tin nhắn loại cuộc gọi và thêm vào nhóm tin nhắn. Hệ thống tạo cuộc gọi và gửi tín hiệu đến người nhận và hiển thị hiển thị giao diện chờ kết nối bên người gọi và giao diện yêu cầu kết nối cuộc gọi bên người nhận. Người nhận có quyền chấp nhận hay không chấp nhận cuộc gọi đó. Nếu người dùng không chấp nhận cuộc gọi, hệ thống cập nhật lại tin nhắn cuộc gọi và gửi tín hiệu từ chối cuộc gọi đến người gọi và hiển thị thông báo từ chối cuộc gọi. Ngược lại, nếu người nhận chấp nhận cuộc gọi, hệ thống xử lý kết nối cuộc gọi và gửi tín hiệu chấp nhận cuộc gọi cho người gọi và chuyển sang giao diện cuộc gọi video cho cả hai bên người dùng. Trường hợp, người dùng không từ chỗi cũng như chấp nhận cuộc gọi, cuộc gọi sẽ tự động kết thúc sau một khoảng thời gian, xử lý như người dùng từ chối cuộc gọi.

* Giai đoạn gọi

Trong cuộc gọi, cả hai bên người dùng đều có quyền bật/tắt sử dụng camera để Chia sẻ hình ảnh và microphone để Chia sẻ âm thanh. Khi người dùng chọn bật camera, hệ thống sẽ yêu cầu người dùng cung cấp quyền sử dụng camera của thiết bị. Nếu người dùng không cho phép, hệ thống sẽ thông báo cho người dùng. Ngược lại, hệ thống lấy luồng dữ liệu video từ camera của thiết bị người dùng, hiển thị luồng dữ liệu ra giao diện cuộc gọi và đồng thời hệ thống gửi luồng dữ liệu video đến đối phương. Khi người dùng nhận được luồng dữ liệu video của đối phương gửi đến, hệ thống sẽ hiển thị video ra giao diện cuộc gọi. Khi người dùng chọn tắt camera, hệ thống sẽ tắt hiển thị video và đồng thời gửi tín hiệu tắt camera đến đối phương. Khi người dùng nhận được tín hiệu tắt camera của đối phương, hệ thống sẽ tắt video của đối phương.

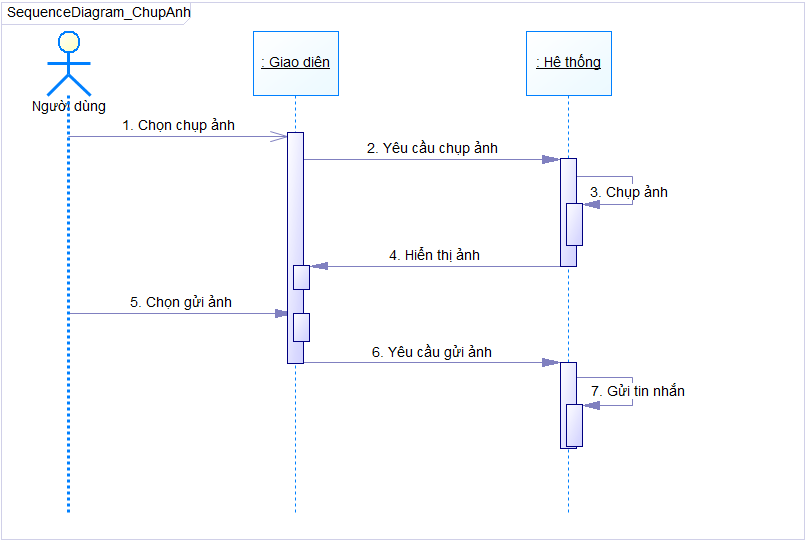
Hệ thống sử lý tương tự cho chức năng bật và tắt microphone. Hệ thống sẽ lấy luồng dữ liệu âm thanh từ microphone của thiết bị người dùng.

* Giai đoạn kết thúc cuộc gọi.

Trong cuộc gọi, cả hai bên người dùng đều có quyền kết thúc cuộc gọi. Người dùng chọn kết thúc cuộc gọi, hệ thống ngắt kết nối cuộc gọi và cập nhật lại tin nhắn cuộc gọi và gửi tín hiệu ngắt kết nối cuộc gọi cho đối phương. Hệ thống hiển thị thông báo cho hai bên người dùng.

Người dùng có thể thực hiện cuộc gọi audio. Xử lý tương tự như thực hiện cuộc gọi video. Trong cuộc gọi audio, người dùng không thể sử dụng chức năng bật camera.

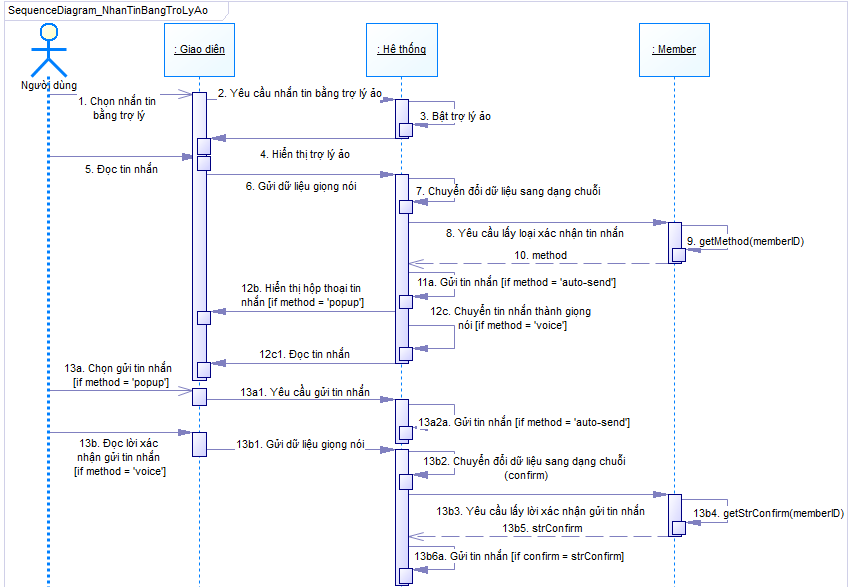
1. **Use case nhắn tin bằng chụp ảnh**



Mô tả: Người dùng chọn nhắn tin bằng chụp ảnh trực tiếp. Hệ thống sử dụng camera chụp lại hình ảnh và hiển thị kết quả chụp. Nếu người dùng chọn gửi ảnh, hệ thống gửi tin nhắn hình ảnh đến người nhận như cách xử lý của use case nhắn tin với bạn bè. Hệ thống tải ảnh lên đám mây và hiển thị hình ảnh cho cả hai bên người dùng.

Người dùng có thể gửi tin nhắn bằng ghi âm âm thanh bằng microphone. Người dùng chọn nhắn tin bằng ghi âm. Hệ thống tiến hành ghi âm bằng microphone của thiết bị. khi kết thúc việc ghi âm, hệ thống gửi tin nhắn âm thanh cho người nhận như cách xử lý của use case nhắn tin với bạn bè. Hệ thống tải âm thanh lên đám mây và hiển thị tin nhắn âm thanh cho cả hai bên người dùng.

1. **Use case nhắn tin bằng trợ lý ảo**



Mô tả: Người dùng chọn nhắn tin bằng trợ lý ảo (chuyển đổi giọng nói thành chuỗi tin nhắn). Người dùng có thể tắt bật việc sử dụng trợ lý ảo trong nhắn tin. Có hai loại nhắn tin bằng trợ lý ảo:

* Trợ lý ảo kích hoạt khi chọn: Kích hoạt trợ lý ảo bằng chỉ thị hành động của người dùng.
* Trợ lý ảo chạy ngầm: Kích hoạt trợ lý ảo bằng chỉ thị giọng nói. Chỉ thị giọng nói này có thể được thiết lập bởi người dùng.

Hệ thống hỗ trợ ngôn ngữ tiếng Anh và tiếng Việt cho trợ lý ảo và người dùng có thể chuyển đổi qua lại gửi hai ngôn ngữ này.

Các loại xác nhận gửi tin nhắn và người dùng có thể chuyển đổi qua lại giữa các kiểu xác nhận:

* Tự động gửi: Không hiển thị tin nhắn chuyển đổi, thực hiện gửi tin nhắn đến người nhận sau khi chuyển đổi tin nhắn.
* Hiển thị hộp thoại xác nhận tin nhắn. Nếu người dùng cho phép gửi tin nhắn, hệ thống sẽ gửi tin nhắn đến người nhận.
* Xác nhận gửi tin nhắn bằng giọng nói: Hệ thống sẽ chuyển tin nhắn vừa được chuyển đổi sang âm thành và phát ra cho người dùng. Người dùng đọc chỉ thị xác nhận gửi tin nhắn. Hệ thống chuyển đổi dữ liệu giọng nói chỉ thị thành chuỗi chỉ thị. Nếu chuỗi chỉ thị là xác nhận gửi tin nhắn, hệ thống sẽ gửi tin nhắn đến người dùng. Ngược lại hệ thống sẽ dừng việc gửi tin nhắn.

Quá trình nhắn tin bằng trợ ảo: Người dùng kích hoạt trợ lý ảo. Hệ thống kích hoạt trợ lý ảo. Người dùng đọc tin nhắn và hệ thống chuyển đổi dữ liệu giọng nói sang dạng chuỗi. Hệ thống hiển thị tin nhắn và yêu cầu xác nhận gửi tin nhắn (như các cách xác nhận ở trên). Nếu người dùng xác nhận gửi tin nhắn hệ thống gửi tin nhắn đến người nhận.

1. **Use case nhắn tin cập nhật thông tin**

Graphical user interface

Description automatically generated

Mô tả: Người dùng nhập tin cập nhật vào form cập nhật thông tin người dùng. Form bao gồm các trường và các quy định cho mỗi trường như sau:

* Trường họ tên: Là chuỗi, không được bỏ trống, dài không quá 40 ký tự, không chứa các kí tự đặc biệt (trừ - và \_).
* Trường giới tính: Có 2 giá trị là “Nam” và “Nữ”, có thể bỏ trống.
* Trường ngày sinh: Có định dạng ngày, nhỏ nhất là ngày 01/01/1900 và lớn nhất là ngày hiện tại, có thể bỏ trống.
* Trường số điện thoại: Có định dạng số điện thoại và có thể bỏ trống.
* Trường địa chỉ: Là chuỗi không chứa các ký tự đặc biệt (trừ - và \_) và có thể bỏ trống.

Người dùng chọn cập nhật thông tin. Hệ thống kiểm tra thông tin người dùng cung cấp có thỏa các quy định trên. Nếu không thỏa, hệ thống sẽ thông báo cho người dùng. Ngược lại, hệ thống cập nhật thông tin và thông báo cho người dùng.

Người dùng có thể cập nhật ảnh đại diện. Có hai cách caaph nhật ảnh đại diện:

* Chọn hình ảnh trong thiết bị người dùng.
* Chụp ảnh từ camera thiết bị người dùng.

Người dùng chọn cập nhật ảnh đại diện. Hệ thống tải ảnh lên đám mây và cập nhật ảnh đại diện người dùng.

Người dùng có thể cập nhật đường dẫn trên thanh địa chỉ. Form cập nhật đường dẫn bao gồm trường đường dẫn và các quy định cho trường như sau: là chuỗi không chứa các ký tự đặc biệt (trừ - và \_), không được bỏ trống, dài không quá 15 kí tự và chưa được người dùng khác sử dụng. Người dùng chọn cập nhật đường dẫn. Hệ thống kiểm tra thông tin người dùng cung cấp. Nếu không thỏa các quy định, hệ thống sẽ thông báo cho người dùng. Ngược lại, hệ thống cập nhật đường dẫn người dùng và thông báo cho người dùng.

Người dùng có thể cập nhật mật khẩu cho tài khoản. Form đổi mật khẩu bao gồm các trường và các quy định cho các trường như sau:

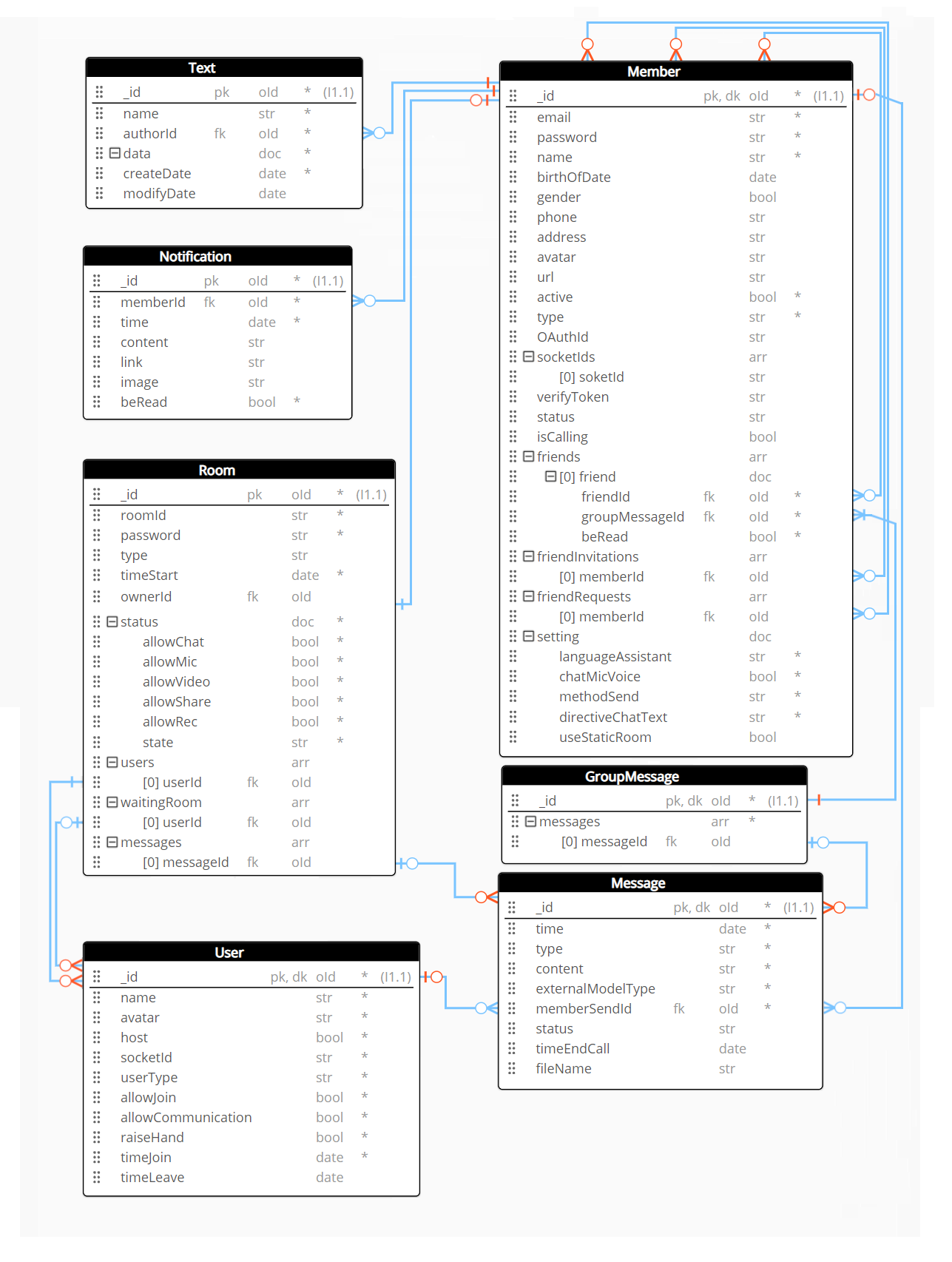
* Trường mật khẩu cũ: Là chuỗi, không được bỏ trống và trùng với mật khẩu của tài khoản.
* Trường mật khẩu: Không được rỗng, dài ít nhất 6 ký tự.
* Trường xác nhận mật khẩu: Phải khớp với trường mật khẩu.

Người dùng chọn đổi mật khẩu. Hệ thống kiểm tra thông tin đổi mật khẩu người dùng cung cấp có thỏa các quy định trên. Nếu không thỏa, hệ thống sẽ thông báo cho người dùng. Ngược lại, hệ thống cập nhật mật khẩu và thông báo cho người dùng.

Người dùng có thể cập nhật mật khẩu cho phòng họp tĩnh. Form đổi mật khẩu phòng họp tĩnh bao gồm trường mật khẩu và các quy định: là chuỗi gồm 4 chữ số và không được bỏ trống. Người dùng chọn cập nhật mật khẩu phòng họp tĩnh. Hệ thống kiểm tra thông tin người dùng cung cấp có thỏa các quy định. Nếu không thỏa, hệ thống sẽ thông báo cho người dùng. Ngược lại, hệ thống cập nhật mật khẩu phòng họp tĩnh và thông báo cho người dùng.

# THIẾT KẾ HỆ THỐNG

1. **Lược đồ ODM**



Hình 4.1 Lược đồ ODM

* Chi tiết các bộ sưu tập

Bảng . Member Collection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Ý nghĩa** | **Kiểu dữ liệu** | **Bắt buộc** | **Ghi chú** |
| \_id | Khóa chính | ObjectId | True |  |
| email | Email người dùng | String | True |  |
| password | Mật khẩu người dùng | String | True |  |
| name | Tên người dùng | String | True |  |
| birthOfDate | Ngày sinh người dùng | Date | False | Giá trị mặc định: 1/1/1970 |
| gender | Giới tính | Boolean | False | Giá trị true: Name  Giá trị false: Nữ |
| phone | Điện thoại người dùng | String | False |  |
| address | Địa chỉ người dùng | String | False |  |
| avatar | Đường dẫn ảnh đại diện người dùng | String | False |  |
| url | Đường dẫ người dùng | String | False |  |
| active | Trạng thái active tài khoản người dùng | Boolean | True | Giá trị true: Tài khoản đã kích hoạt  Giá trị false: Tài khoản chưa kích hoạt |
| type | Loại tài khoản | String | True | Giá trị ‘local’: Tài khoản đăng ký từ ứng dụng  Giá trị ‘google’: Tài khoản đăng nhập bằng Google  Giá trị ‘facebook’: Tài khoản đăng nhập bằng Facebook |
| OAuthId | Id đăng nhập từ bên thứ 3 | String | False | Chỉ có ở loại tài khoản ‘google’ và ‘facebook’ |
| socketIds | Danh sách socketId người dùng khi trực tuyến | Array [String] | False | Người dùng có thể mở nhiều tab khác nhau trên trình duyệt, một tab tương ứng một socketId |
| verifyToken | Mã xác nhận tài khoản | String | False | Được dùng khi kích hoạt tài khoản và quên mật khẩu |
| status | Trạng thái của tài khoản | String | False | Giá trị ‘online’: Người dùng đang trực tuyến  Giá trị chuỗi thời gian: Người dùng đang ngoại tuyến, thời gian là thời điểm người dùng ngoại tuyến |
| isCalling | Người dùng đang có cuộc gọi | Boolean | Fasle |  |
| friends | Danh sách bạn bè | Array [Document] |  |  |
| friends.friendId | Id bạn bè | ObjectId | True | Reference: Member |
| friends. groupMessageId | Id nhóm tin nhắn | ObjectId | True | Reference: GroupMessage |
| friends.beRead | Đã đọc tin nhắn | Boolean | True | Giá trị true: Người dùng đã đọc tin nhắn của bạn bè  Giá trị false: Người dùng chưa đọc tin nhắn của bạn bè |
| friendRequests | Danh sách lời mời kết bạn | Array [ObjectId] | False | Reference: Member |
| friendInvitations | Danh sách yêu cầu kết bạn | Array [ObjectId] | Fasle | Reference: Member |
| setting | Các cài đặt tài khoản | Document | True |  |
| setting. languageAssistant | Ngôn ngữ của trợ lý ảo | String | True | Giá trị ‘vi’: Tiếng Việt  Giá trị ‘en’: Tiếng Anh |
| setting. chatMicVoice | Nhắn tin bằng ghi âm | Boolean | True | Giá trị true: Nhắn tin bằng ghi âm  Giá trị fasle: Nhắn tin bằng trợ lý ảo |
| setting. methodSend | Phương thức xác nhận gửi tin nhắn bằng trợ lý ảo | String | True | Giá trị ‘confirm-popup’: Hiển thị hộp thoại xác nhận  Giá trị ‘confirm-voice’: Xác nhận bằng giọng nói  Giá trị ‘auto-send’: Tự động gửi tin nhắn |
| setting. isChatAssistant | Kích hoạt trợ lý ảo chạy ngầm | Boolean | True |  |
| setting. directiveChatText | Chỉ thị kích hoạt nhắn tin bằng trợ lý ảo chạy ngầm | String | True | Giá trị mặc định: “chat” |
| setting. useStaticRoom | Sử dụng phòng họp tĩnh khi tạo phòng họp | Boolean | True | Giá trị true: Sử dụng phòng họp tĩnh khi tạo phòng họp  Giá trị false: Sử dụng phòng họp ngẫu nhiên khi tạo phòng họp |

Bảng . Text Collection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Ý nghĩa** | **Kiểu dữ liệu** | **Bắt buộc** | **Ghi chú** |
| \_id | Khóa chính | ObjectId | True |  |
| name | Tên văn bản | String | True |  |
| authorId | Id tác gỉả | ObjectId | True | Reference: Member |
| data | Dữ liệu văn bản | Document | True |  |
| createDate | Thời điểm tạo văn bản | Date | True |  |
| modifyDate | Thời điểm sửa văn bản gần nhất | Date | False |  |

Bảng . Notification Collection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Ý nghĩa** | **Kiểu dữ liệu** | **Bắt buộc** | **Ghi chú** |
| \_id | Khóa chính | ObjectId | True |  |
| memberId | Id người dùng | ObjectId | True | Reference: Member |
| time | Thời điểm tạo thông báo | Date | True |  |
| content | Nội dung thông báo | String | True |  |
| link | Đường dẫn đến thông báo | String | True |  |
| image | Hình ảnh của thông báo | String | False |  |
| beRead | Người dùng đã đọc thông báo | Boolean | False | Giá trị mặc định: false |

Bảng . GroupMessage Collection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Ý nghĩa** | **Kiểu dữ liệu** | **Bắt buộc** | **Ghi chú** |
| \_id | Khóa chính | ObjectId | True |  |
| messages | Danh sách id tin mesage | Array [ObjectId] | False | Reference: Message |

Bảng . Message Collection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Ý nghĩa** | **Kiểu dữ liệu** | **Bắt buộc** | **Ghi chú** |
| \_id | Khóa chính | ObjectId | True |  |
| time | Danh sách id tin mesage | Array [ObjectId] | False | Reference: Message |
| type | Loại tin nhắn | String | True | Các giá trị:  ‘start’: Tin nhắn mở đầu  ‘text’: Tin nhắn chuỗi kí tự  ‘edited’: Tin nhắn chuỗi kí tự đã được sửa  ‘raw’: Tin nhắn file dạng text  ‘image’: Tin nhắn file dạng hình ảnh  ‘video’: Tin nhắn file dạng video  ‘audio’: Tin nhắn file dạng âm thanh  ‘call-audio’: Tin nhắn cuộc gọi audio  ‘call-video’: Tin nhắn cuộc gọi video  ‘call-audio-refuse’: Tin nhắn từ chối cuộc gọi audio  ‘call-video-refuse’: Tin nhắn từ chối cuộc gọi video |
| content | Nội dung tin nhắn | String | True |  |
| externalModelType | Tên model reference | String | True | Giá trị “Member”: Member reference  Giá trị “User”: User reference |
| memberId | Id người gửi | ObjectId | True | refPath: externalModelType |
| timeEndCall | Thời điểm kết thúc cuộc gọi | Date | False | Chỉ được dùng khi là tin nhắn cuộc gọi |
| fileName | Tên tệp đính kèm | String | False | Chỉ được dùng khi là tin nhắn tệp đính kèm |

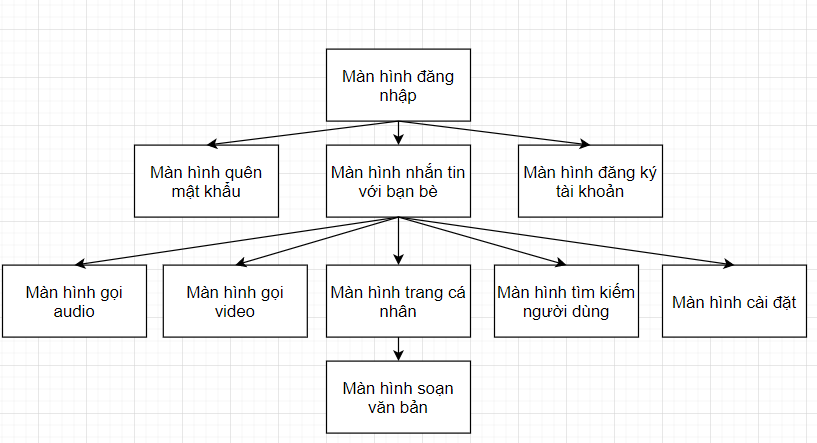
Bảng . Room Collection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Ý nghĩa** | **Kiểu dữ liệu** | **Bắt buộc** | **Ghi chú** |
| \_id | Khóa chính | ObjectId | True |  |
| roomId | Số phòng | String | True | Unique |
| password | Mật khẩu phòng | String | True |  |
| timeStart | Thời điểm bắt đầu phòng họp | Date | True |  |
| ownerId | Id người dùng là chủ phòng họp tĩnh | ObjectId | False | Reference: Member  Nếu có ownerId thì phòng là phòng tĩnh |
| status | Trạng thái, các thiết lập của phòng | Document | True |  |
| status. allowChat | Cho phép thành viên trong phòng nhắn tin | Boolean | True | Giá trị true: cho phép  Giá trị false: không chó phép |
| status. allowMic | Cho phép thành viên trong phòng bật microphone | Boolean | True | Giá trị true: cho phép  Giá trị false: không chó phép |
| status. allowVideo | Cho phép thành viên trong phòng bật camera | Boolean | True | Giá trị true: cho phép  Giá trị false: không chó phép |
| status. allowShare | Cho phép thành viên trong phòng Chia sẻ màn hình | Boolean | True | Giá trị true: cho phép  Giá trị false: không chó phép |
| status. allowRec | Cho phép thành viên trong phòng quay màn hình | Boolean | True | Giá trị true: cho phép  Giá trị false: không chó phép |
| status. isShareScreen | Phòng đang có thành viên Chia sẻ màn hình | Boolean | True | Giá trị true: đang có thành viên Chia sẻ màn hình  Giá trị false: đang không có thành viên Chia sẻ màn hình |
| status.state | Trạng thái của phòng | String | True | Giá trị ‘open’: mở đang phòng  Giá trị ‘looked’: Phòng đang khóa  Giá trị ‘waiting’: phòng đang ở chế độ phòng chờ |
| users | Danh sách id thành viên trong phòng | Array [ObjectId] | True | Reference: User |
| waitingRoom | Danh sách id thành viên trong phòng chờ | Array [ObjectId] | True | Reference: User |
| messages | Danh sách id tin nhắn | Array [ObjectId] | True | Reference: Message |

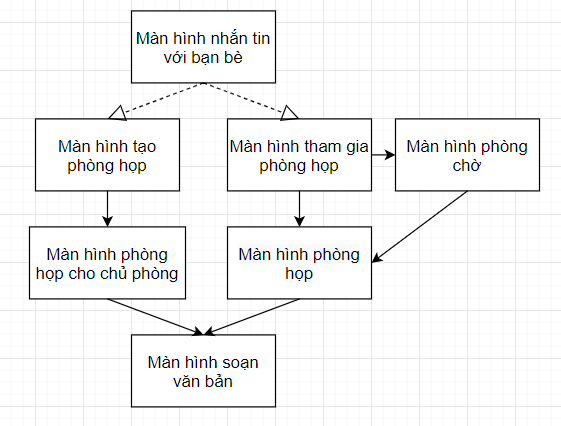
Bảng . User Collection

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tên trường** | **Ý nghĩa** | **Kiểu dữ liệu** | **Bắt buộc** | **Ghi chú** |
| \_id | Khóa chính | ObjectId | True |  |
| name | Tên thành viên | String | True |  |
| avatar | Đường dẫn đến ảnh đại diện của thành viên | String | True |  |
| host | Thành viên là chủ phòng | Boolean | True | Giá trị true: thành viên là chủ phòng  Giá trị false: thành viên không là chủ phòng |
| socketId | Socket Id của thành viên | String | True |  |
| userType | Loại thành viên | String | True | Giá trị ‘session’: loại thành viên không đang trong phiên đăng nhập  Giá trị ‘member’: loại thành viên đang trong phiên đăng nhập |
| allowJoin | Cho phép vào phòng | Boolean | True | Giá trị true: thành viên đang ở trong phòng họp  Giá trị false: thành viên đang ở trong phòng chờ |
| allowComunication | Quyền giao tiếp trong phòng họp của thành viên | Boolean | True | Giá trị true: thành viên có quyền giao tiếp  Giá trị false: thành viên không có quyền giao tiếp |
| raiseHand | Thành viên đang giơ tay | Boolean | True | Giá trị true: thành viên đang giơ tay  Giá trị false: thành viên đang không giơ tay |
| timeJoin | Thời điểm vào phòng | Date | True |  |
| timeLeave | Thời điểm rời phòng | Date | False |  |

1. **Sơ đồ màn hình**
2. **Sơ đồ màn hình**



Hình 4.2 Sơ đồ màn hình nhắn tin với bạn bè



Hình 4.3 Sơ đồ màn hình phòng họp

1. **Mô tả màn hình**
2. **Màn hình đăng nhập**

Bảng . Mô tả đối tượng trên màn hình đăng nhập

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Ràng buộc** | **Chức năng** |
| 1 | Email | Textbox | Không rỗng, có định dạng email | Nhập email đăng nhập tài khoản |
| 2 | Mật khẩu | Textbox | Không rỗng | Nhập mật khẩu đăng nhập tài khoản |
| 3 | Đăng nhập | Button |  | Submit form đăng nhập |
| 4 | Đăng ký | Button |  | Chuyển sang màn hình đăng ký tài khoản |
| 5 | Quên mật khẩu | Button |  | Chuyển sang màn hình quên mật khẩu |
| 6 | Facebook | Button |  | Chuyển sang màn hình đăng nhập bằng Facebook |
| 7 | Google | Button |  | Chuyển sang màn hình đăng nhập bằng Google |
| 8 | Tạo phòng | Button |  | Chuyển sang màn hình tạo phòng họp |
| 9 | Tham gia | Button |  | Chuyển sang màn hình tham gia phòng họp |

Bảng . Danh sách biến cố và xử lý tương ứng trên màn hình đăng nhập

|  |  |  |
| --- | --- | --- |
| **STT** | **Biến cố** | **Xử lý** |
| 1 | Chọn button Đăng nhập | Xử lý đăng nhập, nếu đăng nhập thành công chuyển sang màn hình nhắn tin với bạn bè |
| 2 | Chọn button Đăng ký | Chuyển sang màn hình đăng ký tài khoản |
| 3 | Chọn button Quên mật khẩu | Chuyển sang màn hình quên mật khẩu |
| 4 | Chọn button Facebook | Chuyển sang màn hình đăng nhập bằng Facebook |
| 5 | Chọn button Google | Chuyển sang màn hình đăng nhập bằng Google |
| 6 | Chọn button Tạo phòng | Chuyển sang màn hình tạo phòng họp |
| 7 | Chọn button Tham gia | Chuyển sang màn hình tham gia phòng họp |

1. **Màn hình đăng ký tài khoản**

Bảng . Mô tả đối tượng trên màn hình đăng ký tài khoản

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Ràng buộc** | **Chức năng** |
| 1 | Họ tên | Textbox | Không rỗng, không chứa ký tự đặc biệt (trừ - và \_), dài không quá 40 ký tự | Nhập tên người dùng đăng ký tài khoản |
| 2 | Email | Textbox | Không rỗng, có định dạng email | Nhập email người dùng đăng ký tài khoản |
| 3 | Mật khẩu | Textbox | Không rỗng, dài ít nhất 6 ký tự | Nhập mật khẩu đăng ký tài khoản |
| 4 | Xác nhận MK | Textbox | Khớp với trường mật khẩu | Nhập lại mật khẩu đăng ký tài khoản |
| 5 | Đăng ký | Button |  | Submit form đăng ký |
| 6 | Đăng nhập | Button |  | Chuyển sang màn hình đăng nhập |
| 7 | Facebook | Button |  | Chuyển sang màn hình đăng nhập bằng Facebook |
| 8 | Google | Button |  | Chuyển sang màn hình đăng nhập bằng Google |

Bảng . Danh sách biến cố và xử lý tương ứng trên màn hình đăng ký tài khoản

|  |  |  |
| --- | --- | --- |
| **STT** | **Biến cố** | **Xử lý** |
| 1 | Chọn button Đăng ký | Xử lý đăng ký tài khoản, nếu đăng ký thành công chuyển sang màn hình đăng nhập |
| 2 | Chọn button Đăng nhập | Chuyển sang màn hình đăng nhập |
| 3 | Chọn button Facebook | Chuyển sang màn hình đăng nhập bằng Facebook |
| 4 | Chọn button Google | Chuyển sang màn hình đăng nhập bằng Google |

1. **Màn hình quên mật khẩu**

Bảng . Mô tả đối tượng trên màn hình quên mật khẩu bước 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Ràng buộc** | **Chức năng** |
| 1 | Email | Textbox | Không rỗng, có định dạng email | Nhập email đăng nhập tài khoản |
| 2 | Xác nhận | Button |  | Submit form xác nhận tài khoản |
| 3 | Đăng nhập | Button |  | Chuyển sang màn hình đăng nhập |

Bảng . Mô tả đối tượng trên màn hình quên mật khẩu bước 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Ràng buộc** | **Chức năng** |
| 1 | Mật khẩu | Textbox | Không rỗng, dài ít nhất 6 ký tự | Nhập mật khẩu đăng ký tài khoản |
| 2 | Xác nhận MK | Textbox | Khớp với trường mật khẩu | Nhập lại mật khẩu đăng ký tài khoản |
| 3 | Đổi mật khẩu | Button |  | Submit form đổi mật khẩu |
| 4 | Đăng nhập | Button |  | Chuyển sang màn hình đăng nhập |

Bảng . Danh sách biến cố và xử lý tương ứng trên màn hình quên mật khẩu bước 1

|  |  |  |
| --- | --- | --- |
| **STT** | **Biến cố** | **Xử lý** |
| 1 | Chọn button Xác nhận | Xử lý xác nhận quên mật khẩu, nếu xác nhận thành công chuyển sang màn hình quên mật khẩu bước 2 |
| 2 | Chọn button Đăng nhập | Chuyển sang mà hình đăng nhập |

Bảng . Danh sách biến cố và xử lý tương ứng trên màn hình quên mật khẩu bước 2

|  |  |  |
| --- | --- | --- |
| **STT** | **Biến cố** | **Xử lý** |
| 1 | Chọn button Đổi mật khẩu | Xử lý đổi mật khẩu, nếu đổi mật khẩu thành công chuyển sang màn hình đăng nhập |
| 2 | Chọn button Đăng nhập | Chuyển sang màn hình đăng nhập |

1. **Màn hình nhắn tin với bạn bè**

Bảng . Mô tả đối tượng trên màn hình nhắn tin với bạn bè

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Ràng buộc** | **Chức năng** |
| 1 | Tin nhắn | Textbox | Không rỗng | Nhập tin nhắn |
| 2 | Biểu tượng cảm xúc | Button |  | Chọn biểu tượng cảm xúc |
| 3 | File | Button |  | Chọn gửi tệp đính kèm |
| 4 | Chụp ảnh | Button |  | Chọn nhắn tin chụp ảnh |
| 5 | Ghi âm | Button |  | Chọn nhắn tin bằng ghi âm hoặc nhắn tin bằng trợ lý ảo |
| 6 | Gửi tin nhắn | Button |  | Gửi tin nhắn |
| 7 | Gọi audio | Button |  | Chuyển sang màn hình gọi audio |
| 8 | Gọi video | Button |  | Chuyển sang màn hình gọi video |
| 9 | Tìm kiếm người dùng [\*] | Textbox | Không rỗng | Nhập thông tin người dùng muốn tìm kiếm |
| 10 | Tìm kiếm bạn bè | Textbox | Không rỗng | Nhập thông tin bạn bè muốn tìm kiếm |
| 11 | Tạo phòng họp | Button |  | Chuyển sang màn hình tạo phòng họp |
| 12 | Tham gia phòng họp | Button |  | Chuyển sang màn hình tham gia phòng họp |
| 13 | Trang cá nhân [\*] | Button |  | Chuyển sang màn hình trang cá nhân |
| 14 | Cài đặt [\*] | Button |  | Chuyển sang màn hình cài đặt |
| 15 | Đăng xuất [\*] | Button |  | Đăng xuất tài khoản |
| 16 | Thông báo [\*] | Button |  | Hiển thị danh sách thông báo thông báo. Mỗi thông báo có button xóa, chuyển trạng thái thông báo |
| 17 | Bạn bè | List |  | Hiển thị danh sách bạn bè. Mỗi bạn bè có button Trang cá nhân |

Các đối tượng [\*] sử dụng chung cho các màn hình nhắn tin với bạn bè, màn hình trang cá nhân, màn hình tìm kiếm người dùng, màn hình cài đặt.

Bảng . Danh sách biến cố và xử lý tương ứng trên màn hình nhắn tin với bạn bè

|  |  |  |
| --- | --- | --- |
| **STT** | **Biến cố** | **Xử lý** |
| 1 | Chọn button Gửi tin nhắn | Xử lý gửi tin nhắn |
| 2 | Chọn button Biểu tượng cảm xúc | Xử lý thêm biểu tượng cảm xúc vào tin nhắn |
| 3 | Chọn button File | Xử lý gửi tệp đính kèm |
| 4 | Chọn button Chụp ảnh | Xử lý gửi tin nhắn bằng chụp ảnh |
| 5 | Chọn giữ button Ghi âm | Xử lý nhắn tin bằng ghi âm nếu người dùng đang sử dụng nhắn tin bằng ghi âm |
| 6 | Chọn button Ghi âm | Xử lý nhắn tin bằng trợ lý ảo nếu người dùng đang sử dụng nhắn tin bằng trợ lý ảo |
| 7 | Chọn button Gọi audio | Xử lý gọi audio cho bạn bè và chuyển sang màn hình gọi audio |
| 8 | Chọn button Gọi video | Xử lý gọi video cho bạn bè và chuyển sang màn hình gọi video |
| 9 | Nhập nội dung tìm kiếm người dùng vào textbox Tìm kiếm người dùng | Nhận thông tin tìm kiếm, xử lý tìm kiếm người dùng và trả về danh sách người dùng tìm kiếm được và chuyển sang màn hình tìm kiếm người dùng |
| 10 | Nhập nội dung tìm kiếm người dùng vào textbox Tìm kiếm bạn bè | Nhận thông tin tìm kiếm, xử lý tìm kiếm bạn bè và trả về danh sách bạn bè tìm kiếm được |
| 11 | Chọn button Tạo phòng họp | Xử lý tạo phòng có phiên đăng nhập và chuyển sang màn hình tạo phòng họp |
| 12 | Chọn button Tham gia phòng họp | Xử lý tham gia phòng có phiên đăng nhập và chuyển sang màn hình tham gia phòng họp |
| 13 | Chọn button Trang cá nhân | Chuyển sang màn hình trang cá nhân |
| 14 | Chọn button Cài đặt | Chuyển sang màn hình cài đặt |
| 15 | Chọn button Đăng xuất | Xử lý đăng xuất tài khoản và chuyển sang màn hình đăng nhập |
| 16 | Chọn button Thông báo | Hiển thị danh sách thông báo của người dùng. Nếu người dùng chọn button xóa thông báo, xử lý xóa thông báo. Nếu người dùng chọn button chuyển trạng thái thông báo, xử lý chuyển trạng thái của thông báo tương ứng |
| 17 | Chọn bạn bè | Chuyển sang giao diện nhắn tin với bạn bè tương ứng. Nếu người dùng chọn xem trang cá nhân bạn bè, chuyển sang giao diện trang cá nhân của bạn bè tương ứng |

1. **Màn hình gọi audio**

Bảng . Mô tả đối tượng trên màn hình gọi audio

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Ràng buộc** | **Chức năng** |
| 1 | Kết thúc cuộc gọi | Button |  | Kết thúc cuộc gọi |
| 2 | Microphone | Button |  | Bật/Tắt sử dụng microphone |

Bảng . Danh sách biến cố và xử lý tương ứng trên màn hình gọi audio

|  |  |  |
| --- | --- | --- |
| **STT** | **Biến cố** | **Xử lý** |
| 1 | Chọn button Kết thúc cuộc gọi | Xử lý kết thúc cuộc gọi và đóng màn hình gọi audio |
| 2 | Chọn button Microphone | Nếu người dùng đang bật microphone, xử lý tắt microphone. Ngược lại, xử lý bật microphone |

1. **Màn hình gọi video**

Bảng . Mô tả đối tượng trên màn hình gọi video

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Ràng buộc** | **Chức năng** |
| 1 | Kết thúc cuộc gọi | Button |  | Kết thúc cuộc gọi |
| 2 | Microphone | Button |  | Bật/Tắt sử dụng microphone |
| 3 | Video | Button |  | Bật/Tắt sử dụng camera |

Bảng . Danh sách biến cố và xử lý tương ứng trên màn hình gọi video

|  |  |  |
| --- | --- | --- |
| **STT** | **Biến cố** | **Xử lý** |
| 1 | Chọn button Kết thúc cuộc gọi | Xử lý kết thúc cuộc gọi và đóng màn hình gọi video |
| 2 | Chọn button Microphone | Nếu người dùng đang bật microphone, xử lý tắt microphone. Ngược lại, xử lý bật microphone |
| 3 | Chọn button Camera | Nếu người dùng đang bật camera, xử lý tắt camera. Ngược lại, xử lý bật camera |

1. **Màn hình trang cá nhân**

Bảng . Mô tả đối tượng trên màn hình trang cá nhân

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Ràng buộc** | **Chức năng** |
| 1 | Họ tên | Textbox | Không rỗng, không chứa các ký tự đặc biệt (trừ - và \_), dài không quá 40 ký tự | Nhập tên người dùng |
| 2 | Ngày sinh | Textbox | Không rỗng, có định dạng Date | Nhập ngày sinh người dùng |
| 3 | Giới tính | Checkbox |  | Chọn giới tính người dùng |
| 5 | Điện thoại | Textbox | Có định dạng số điện thoại | Nhập số điện thoại người dùng |
| 6 | Địa chỉ | Textbox | Không chứa các ký tự đặc biệt (trừ - và \_) | Nhập địa chỉ người dùng |
| 7 | Lưu | Button |  | Submit form cập nhật thông tin người dùng |
| 8 | Đổi ảnh đại diện | Button |  | Đổi ảnh đại diện người dùng. Có button chụp ảnh và button chọn ảnh |
| 9 | Bạn bè | List |  | Hiển thị danh sách bạn bè. Mỗi bạn bè có button Trang cá nhân, button Nhắn tin và button Hủy kết bạn |
| 9 | Yêu cầu kết bạn | List |  | Hiển thị danh sách yêu cầu kết bạn. Mỗi yêu cầu kết bạn có button Trang cá nhân và button Hủy yêu cầu kết bạn |
| 10 | Lời mời kết bạn | List |  | Hiển thị danh sách lời mời kết bạn. Mỗi lời mời kết bạn có button Trang cá nhân, button Chấp nhận lời mời kết bạn và button Xóa lời mời kết bạn |
| 11 | Tạo văn bản mới | Button |  | Chuyển sang màn hình soạn thảo văn bản |
| 12 | Văn bản | List |  | Hiển thị danh sách văn bản. Mỗi văn bản có button xóa và sửa văn bản |

Bảng . Danh sách biến cố và xử lý tương ứng trên màn hình trang cá nhân

|  |  |  |
| --- | --- | --- |
| **STT** | **Biến cố** | **Xử lý** |
| 1 | Chọn button Lưu | Xử lý cập nhật thông tin người dùng và hiển thị thông báo |
| 2 | Chọn button đổi ảnh đại diện | Nếu người dùng chọn button Chụp ảnh, xử lý chụp ảnh và cập nhật ảnh đại diện. Nếu người dùng chọn button Chọn file, xử lý chọn file ảnh và cập nhật ảnh đại diện |
| 3 | Chọn button Xem trang cá nhân người dùng | Chuyển sang màn hình trang cá nhân người dùng |
| 4 | Chọn button Nhắn tin | Chuyển sang màn hình nhắn tin với bạn bè được chọn |
| 5 | Chọn button Hủy kết bạn | Xử lý hủy kết bạn tương ứng |
| 6 | Chọn button Hủy yêu cầu kết bạn | Xử lý hủy yêu cầu kết bạn tương ứng |
| 7 | Chọn button Xóa yêu cầu kết bạn | Xử lý xóa yêu cầu kết bạn tương ứng |
| 8 | Chọn button Chấp nhận lời mời kết bạn | Xử lý chấp nhận lời mời kết bạn tương ứng |
| 9 | Chọn button Xóa lời mời kết bạn | Xử lý xóa lời mời kết bạn tương ứng |
| 10 | Chon button Tạo văn bản mới | Xử lý tạo văn bản mới và chuyển sang giao diện soạn thảo văn bản |
| 11 | Chọn button Xóa văn bản | Xử lý xóa văn bản tương ứng |
| 12 | Chọn button Sửa văn bản | Chuyển sang giao diện soạn thảo văn bản tương ứng |

1. **Màn hình đăng màn hình tìm kiếm người dùng**

Bảng . Mô tả đối tượng trên màn hình tìm kiếm người dùng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Ràng buộc** | **Chức năng** |
| 1 | Người dùng | List |  | Hiển thị kết quả tìm kiếm người dùng. Mỗi người dùng có button Trang cá nhân; button Nhắn tin và Hủy kết bạn nếu là bạn bè; button Hủy yêu cầu kết bạn nếu người dùng đã gửi yêu cầu kết bạn; button Chấp nhận lời mời kết bạn và button Xóa lời mời kết bạn nếu người dụng được gửi lời mời kết bạn |

Bảng . Danh sách biến cố và xử lý tương ứng trên màn hình tìm kiếm người dùng

|  |  |  |
| --- | --- | --- |
| **STT** | **Biến cố** | **Xử lý** |
| 1 | Chọn button Xem trang cá nhân người dùng | Chuyển sang màn hình trang cá nhân người dùng |
| 2 | Chọn button Nhắn tin | Chuyển sang màn hình nhắn tin với bạn bè được chọn |
| 3 | Chọn button Hủy kết bạn | Xử lý hủy kết bạn tương ứng |
| 4 | Chọn button Hủy yêu cầu kết bạn | Xử lý hủy yêu cầu kết bạn tương ứng |
| 5 | Chọn button Xóa yêu cầu kết bạn | Xử lý xóa yêu cầu kết bạn tương ứng |
| 6 | Chọn button Chấp nhận lời mời kết bạn | Xử lý chấp nhận lời mời kết bạn tương ứng |
| 7 | Chọn button Xóa lời mời kết bạn | Xử lý xóa lời mời kết bạn tương ứng |

1. **Màn hình cài đặt**

Bảng . Mô tả đối tượng trên màn hình cài đặt

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Ràng buộc** | **Chức năng** |
| 1 | Ngôn ngữ trợ lý ảo | Select |  | Chọn ngôn ngữ cho trợ lý ảo |
| 2 | Nhắn tin bằng ghi âm | Checkbox |  | Chọn phương thức nhắn tin bằng ghi âm hay bằng trợ lý ảo |
| 3 | Phương thức xác nhận gửi tin nhắn | Select |  | Chọn phương thức xác nhận gửi tin nhắn bằng trợ lý ảo |
| 4 | Chạy nền trợ lý ảo | Checkbox |  | Bật chế độ chạy nền cho trợ lý ảo |
| 5 | Chỉ thị xác nhận | Button |  | Thiết lập chuỗi chỉ thị xác nhận gửi tin nhắn bằng giọng nói |
| 6 | Lưu cài đặt nhắn tin | Button |  | Submit form cài đặt nhắn tin |
| 7 | Mật khẩu | Textbox | Không rỗng, dài ít nhất 6 ký tự | Nhập mật khẩu đăng ký tài khoản |
| 8 | Xác nhận MK | Textbox | Khớp với trường mật khẩu | Nhập lại mật khẩu đăng ký tài khoản |
| 9 | Đổi mật khẩu | Button |  | Submit form đổi mật khẩu |
| 10 | Url | Textbox | Không rỗng, dài không quá 15 ký tự, không chứa các ký tự đặc biệt (trừ - và \_) | Nhập đường dẫn người dùng |
| 11 | Đổi url | Button |  | Submit form đổi url |

Bảng . Danh sách biến cố và xử lý tương ứng trên màn hình cài đặt

|  |  |  |
| --- | --- | --- |
| **STT** | **Biến cố** | **Xử lý** |
| 1 | Chọn button Lưu cài đặt nhắn tin | Xử lý cập nhật cài đặt nhắn tin |
| 2 | Chọn button Đổi mật khẩu | Xử lý đổi mật khẩu người dùng |
| 3 | Chọn button Đổi url | Xử lý đổi đường dẫn người dùng |

1. **Màn hình soạn thảo văn bản**

Bảng . Mô tả đối tượng trên màn hình soạn thảo văn bản

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Ràng buộc** | **Chức năng** |
| 1 | Tên văn bản | Textbox | Không rỗng | Nhập tên văn bản |
| 2 | Lưu tên văn bản | Button |  | Submit form lưu tên văn bản |
| 3 | Văn bản | TextEditor |  | Soạn thảo văn bản |
| 4 | Format | List Button |  | Định dạng văn bản |

Bảng . Danh sách biến cố và xử lý tương ứng trên màn hình soạn thảo văn bản

|  |  |  |
| --- | --- | --- |
| **STT** | **Biến cố** | **Xử lý** |
| 1 | Chọn button Lưu tên văn bản | Xử lý cập nhật tên văn bản |
| 2 | Chọn các button Format | Xử lý định dạng văn bản |
| 3 | Nhấn Ctrl + S | Xử lý lưu văn bản |

1. **Màn hình tạo phòng họp**

Bảng . Mô tả đối tượng trên màn hình tạo phòng họp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Ràng buộc** | **Chức năng** |
| 1 | Tên hiển thị | Textbox | Không rỗng, không chứa các kí tự đặc biệt (trừ - và \_), dài không quá 40 ký tự | Nhập tên hiển thị trong phòng họp |
| 2 | Số phòng | Textbox | Được thiết lập sẵn | Hiển thị số phòng họp |
| 3 | Mật khẩu | Textbox | Được thiết lập sẵn | Hiển thị mật khẩu phòng họp |
| 4 | Tạo phòng | Button |  | Submit form tạo phòng họp |

Bảng . Danh sách biến cố và xử lý tương ứng trên màn hình tạo phòng họp

|  |  |  |
| --- | --- | --- |
| **STT** | **Biến cố** | **Xử lý** |
| 1 | Chọn button Tạo phòng | Xử lý tạo phòng họp và chuyển sang màn hình phòng họp cho chủ phòng |

1. **Màn hình tham gia phòng họp**

Bảng . Mô tả đối tượng trên màn hình tham gia phòng họp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Ràng buộc** | **Chức năng** |
| 1 | Tên hiển thị | Textbox | Không rỗng, không chứa các kí tự đặc biệt (trừ - và \_), dài không quá 40 ký tự | Nhập tên hiển thị trong phòng họp |
| 2 | Số phòng | Textbox | Không rỗng, dài 9 ký số | Nhập số phòng cần tham gia |
| 3 | Mật khẩu | Textbox | Không rỗng, dài 4 ký số | Nhập mật khẩu phòng cần tham gia |
| 4 | Tham gia | Button |  | Submit form tham gia phòng họp |

Bảng . Danh sách biến cố và xử lý tương ứng trên màn hình tham gia phòng họp

|  |  |  |
| --- | --- | --- |
| **STT** | **Biến cố** | **Xử lý** |
| 1 | Chọn button Tham gia | Xử lý tham gia phòng họp và chuyển sang màn hình phòng họp nếu phòng đang mở và hoặc chuyển sang màn hình phòng chờ nếu phòng đang ở trạng thái phòng chờ |

1. **Màn hình phòng chờ**

Bảng . Mô tả đối tượng trên màn hình phòng chờ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Ràng buộc** | **Chức năng** |
| 1 | Thoát phòng chờ | Button |  | Thoát khỏi phòng chờ |

Bảng . Danh sách biến cố và xử lý tương ứng trên màn hình phòng chờ

|  |  |  |
| --- | --- | --- |
| **STT** | **Biến cố** | **Xử lý** |
| 1 | Chọn button Thoát phòng chờ | Xử lý thoát khỏi phòng chờ và chuyển sang màn hình đăng nhập nếu người dùng đang không trong phiên đăng nhập, hoặc chuyển sang màn hình nhắn tin với bạn bè nếu người dùng đang ở trong phiên đăng nhập |

1. **Màn hình phòng họp cho chủ phòng**

Bảng . Mô tả đối tượng trên màn hình phòng họp cho chủ phòng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Ràng buộc** | **Chức năng** |
| 1 | Microphone | Button |  | Bật/Tắt sử dụng microphone |
| 2 | Camera | Button |  | Bật/Tắt sử dụng camera |
| 3 | Chia sẻ màn hình | Button |  | Bật/Tắt Chia sẻ màn hình |
| 4 | Quay màn hình | Button |  | Quay/Dừng quay màn hình |
| 5 | Giơ tay | Button |  | Giơ tay/Bỏ giơ tay |
| 6 | Tạo văn bản | Button |  | Tạo văn bản soạn thảo, mở thêm màn hình soạn thảo văn bản |
| 7 | Thành viên trong phòng | List |  | Hiển thị danh sách thành viên trong phòng. Trừ chủ phòng, mỗi thành viên có button Đuổi khỏi phòng, button Cấm giao tiếp |
| 8 | Thành viên trong phòng chờ | List |  | Hiển thị danh sách thành viên trong phòng chờ. Mỗi thành viên có button Cho phép vào phòng và button Không cho phép vào phòng |
| 9 | Lưu danh sách | Button |  | Lưu danh sách thành viên |
| 10 | Tin nhắn | Textbox | Không rỗng | Nhập tin nhắn |
| 11 | File | Button |  | Chọn tệp đính kèm cần gửi |
| 12 | Biểu tượng cảm xúc | Button |  | Chọn biểu tượng cảm xúc thêm vào tin nhắn |
| 13 | Gửi tin nhắn | Button |  | Gửi tin nhắn |
| 14 | Lưu tin nhắn | Button |  | Lưu danh sách tin nhắn trong phòng |
| 15 | Ẩn thời gian | Button |  | Ẩn/Hiện thời gian tin nhắn |
| 16 | Thông tin phòng | Button |  | Hiển thị thông tin phòng họp. Thông tin gồm số phòng và mật khẩu phòng |
| 17 | Quản lý phòng | Checkbox List |  | Quản lý trạng thái phòng. Gồm 3 lựa chọn: Mở phòng, khóa phòng và đặt trạng thái phòng chờ |
| 18 | Quản lý tin nhắn | Checkbox List |  | Quản lý việc giao tiếp trong phòng. Gồm: Tắt trò chuyện, Cấm bật microphone, Cấm bật camera, Cấm Chia sẻ màn hình, Cấm quay màn hình |
| 19 | Rời phòng | Button |  | Rời phòng họp |
| 20 | Kết thúc cuộc họp | Button |  | Kết thúc cuộc họp |

Bảng . Danh sách biến cố và xử lý tương ứng trên màn hình phòng họp cho chủ phòng

|  |  |  |
| --- | --- | --- |
| **STT** | **Biến cố** | **Xử lý** |
| 1 | Chọn button Microphone | Nếu người dùng đang bật microphone, xử lý tắt microphone. Ngược lại, xử lý bật microphone |
| 2 | Chọn button Camera | Nếu người dùng đang bật camera, xử lý tắt camera. Ngược lại, xử lý bật camera |
| 3 | Chọn button Chia sẻ màn hình | Nếu người dùng đang Chia sẻ màn hình, xử lý dừng Chia sẻ màn hình. Ngược lại, xử lý Chia sẻ màn hình |
| 4 | Chọn button Quay màn hình | Nếu người dùng đang quay màn hình, xử lý bật quay màn hình. Ngược lại, xử lý tắt quay màn hình |
| 5 | Chọn button Giơ tay | Nếu người dùng đang giơ tay, xử lý bỏ giơ tay. Ngược lại, xử lý bật giơ tay |
| 6 | Chọn button Tạo văn bản | Xử lý tạo văn bản soạn thảo, mở màn hình soạn thảo văn bản |
| 7 | Chọn button Đuổi khỏi phòng | Xử lý đuổi thành viên tương ứng khỏi phòng họp |
| 8 | Chọn button Cấm giao tiếp | Xử lý cầm giao tiếp đối với thành viên được chọn nếu thành viên đó đang được phép giao tiếp. Ngược lại, xử lý cho phép giao tiếp đồi với thành viên được chọn |
| 9 | Chọn button Cho phép vào phòng | Xử lý cho phép thành viên tương ứng ở phòng chờ vào phòng họp và chuyển sang màn hình phòng họp cho thành viên đó |
| 10 | Chọn button Không cho phép vào phòng | Xử lý không cho phép thành viên tương ứng ở phòng chờ vào phòng họp |
| 11 | Chọn button Lưu danh sách | Xử lý xuất danh sách thành viên trong phòng và tải về máy người dùng |
| 12 | Chọn button Gửi tin nhắn | Xử lý gửi tin nhắn |
| 13 | Chọn button Ẩn thời gian | Xử lý ẩn thời gian của tin nhắn nếu đang hiển thị thời gian tin nhắn. Ngược lại, xử lý hiện thời gian tin nhắn |
| 14 | Chọn button Thông tin phòng | Xử lý hiển thị thông tin phòng |
| 15 | Chọn checkbox Mở phòng | Xử lý mở phòng họp |
| 16 | Chọn checkbox Khóa phòng | Xử lý khóa phòng họp |
| 17 | Chọn checkbox Đặt trạng thái phòng chờ | Xử lý đặt trạng thái phòng chờ |
| 18 | Chọn checkbox Tắt trò chuyện | Xử lý tắt trò chuyện đối với các thành viên trong phòng nếu phòng đang cho phép trò chuyện, Ngược lại, xử lý cho phép trò chuyện đối với các thành viên trong phòng |
| 19 | Chọn checkbox Cấm bật microphone | Xử lý cấm bật microphone đối với các thành viên trong phòng nếu phòng đang cho phép bật microphone, Ngược lại, xử lý cho phép bật microphone đối với các thành viên trong phòng |
| 20 | Chọn checkbox Cấm bật camera | Xử lý cấm bật camera đối với các thành viên trong phòng nếu phòng đang cho phép bật camera, Ngược lại, xử lý cho phép bật camera đối với các thành viên trong phòng |
| 21 | Chọn checkbox Cấm Chia sẻ màn hình | Xử lý cấm Chia sẻ màn hình đối với các thành viên trong phòng nếu phòng đang cho phép cho phép Chia sẻ màn hình, Ngược lại, xử lý cho phép Chia sẻ màn hình đối với các thành viên trong phòng |
| 22 | Chọn checkbox Cấm quay màn hình | Xử lý cấm quay màn hình đối với các thành viên trong phòng nếu phòng đang cho phép quay màn hình, Ngược lại, xử lý cho phép quay màn hình đối với các thành viên trong phòng |
| 23 | Chọn button Rời phòng | Xử lý rời phòng họp |
| 24 | Chọn button kết thúc cuộc họp | Xử lý kết thúc cuộc họp |

1. **Màn hình phòng họp**

Bảng . Mô tả đối tượng trên màn hình phòng họp

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Ràng buộc** | **Chức năng** |
| 1 | Microphone | Button |  | Bật/Tắt sử dụng microphone |
| 2 | Camera | Button |  | Bật/Tắt sử dụng camera |
| 3 | Chia sẻ màn hình | Button |  | Bật/Tắt Chia sẻ màn hình |
| 4 | Quay màn hình | Button |  | Quay/Dừng quay màn hình |
| 5 | Giơ tay | Button |  | Giơ tay/Bỏ giơ tay |
| 6 | Tạo văn bản | Button |  | Tạo văn bản soạn thảo, mở thêm màn hình soạn thảo văn bản |
| 7 | Thành viên trong phòng | List |  | Hiển thị danh sách thành viên trong phòng. |
| 8 | Tin nhắn | Textbox | Không rỗng | Nhập tin nhắn |
| 9 | File | Button |  | Chọn tệp đính kèm cần gửi |
| 10 | Biểu tượng cảm xúc | Button |  | Chọn biểu tượng cảm xúc thêm vào tin nhắn |
| 11 | Gửi tin nhắn | Button |  | Gửi tin nhắn |
| 12 | Lưu tin nhắn | Button |  | Lưu danh sách tin nhắn trong phòng |
| 13 | Ẩn thời gian | Button |  | Ẩn/Hiện thời gian tin nhắn |
| 14 | Thông tin phòng | Button |  | Hiển thị thông tin phòng họp. Thông tin gồm số phòng và mật khẩu phòng |
| 15 | Rời phòng | Button |  | Rời phòng họp |

Bảng . Danh sách biến cố và xử lý tương ứng trên màn hình phòng họp

|  |  |  |
| --- | --- | --- |
| **STT** | **Biến cố** | **Xử lý** |
| 1 | Chọn button Microphone | Nếu người dùng đang bật microphone, xử lý tắt microphone. Ngược lại, xử lý bật microphone |
| 2 | Chọn button Camera | Nếu người dùng đang bật camera, xử lý tắt camera. Ngược lại, xử lý bật camera |
| 3 | Chọn button Chia sẻ màn hình | Nếu người dùng đang Chia sẻ màn hình, xử lý dừng Chia sẻ màn hình. Ngược lại, xử lý Chia sẻ màn hình |
| 4 | Chọn button Quay màn hình | Nếu người dùng đang quay màn hình, xử lý bật quay màn hình. Ngược lại, xử lý tắt quay màn hình |
| 5 | Chọn button Giơ tay | Nếu người dùng đang giơ tay, xử lý bỏ giơ tay. Ngược lại, xử lý bật giơ tay |
| 6 | Chọn button Tạo văn bản | Xử lý tạo văn bản soạn thảo, mở màn hình soạn thảo văn bản |
| 7 | Chọn button Gửi tin nhắn | Xử lý gửi tin nhắn |
| 8 | Chọn button Ẩn thời gian | Xử lý ẩn thời gian của tin nhắn nếu đang hiển thị thời gian tin nhắn. Ngược lại, xử lý hiện thời gian tin nhắn |
| 9 | Chọn button Thông tin phòng | Xử lý hiển thị thông tin phòng |
| 10 | Chọn button Rời phòng | Xử lý rời phòng họp |

# CÀI ĐẶT HỆ THỐNG VÀ KẾT QUẢ ĐẠT ĐƯỢC

1. **Nhóm giao diện đăng nhập**

Graphical user interface, text, application

Description automatically generated

Hình 5.1 Giao diện đăng nhập

Graphical user interface, text, application

Description automatically generated

Hình 5.2 Giao diện đăng ký tài khoản

Graphical user interface, application

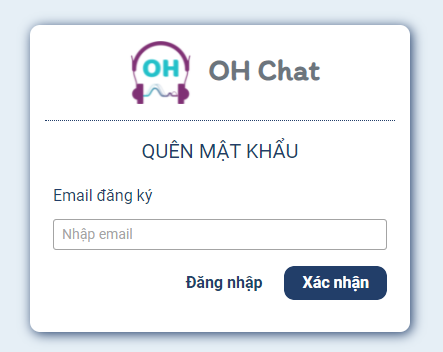
Description automatically generated

Hình 5.3 Giao diện email xác nhận đăng ký tài khoản

Graphical user interface, text, application

Description automatically generated

Hình 5.4 Giao diện đăng nhập bằng Google



Hình 5.5 Giao diện quên mật khẩu bước 1

Graphical user interface, application

Description automatically generated

Hình 5.6 Giao diện email xác nhận tài khoản quên mật khẩu

Graphical user interface, text, application

Description automatically generated

Hình 5.7 Giao diện quên mật khẩu bước 2

1. **Nhóm giao diện phòng họp**

Graphical user interface, text, application, chat or text message

Description automatically generated

Hình 5.8 Giao diện tạo phòng họp

Graphical user interface, text, application, chat or text message

Description automatically generated

Hình 5.9 Giao diện tham gia phòng họp

Graphical user interface, text, application, chat or text message

Description automatically generated

Hình 5.10 Giao diện tham gia phòng chờ

Graphical user interface, application

Description automatically generated

Hình 5.11 Giao diện phòng họp cho chủ phòng

Graphical user interface, application

Description automatically generated

Hình 5.12 Giao diện phòng họp cho người tham gia

Graphical user interface, application

Description automatically generated

Hình 5.13 Giao diện quản lý phòng

Graphical user interface, application

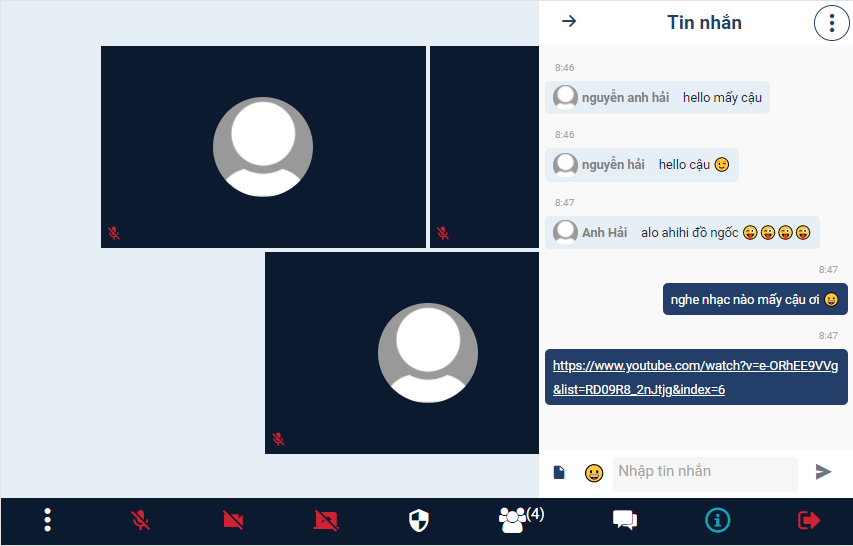
Description automatically generated

Hình 5.14 Giao diện thành viên phòng họp

Graphical user interface, text, application

Description automatically generated

Hình 5.15 Giao diện thông tin phòng



Hình 5.16 Giao diện nhắn tin trong phòng họp

A screenshot of a computer

Description automatically generated with medium confidence

Hình 5.17 Giao diện sử dụng microphone, camera

Graphical user interface

Description automatically generated

Hình 5.18 Giao diện chia sẻ màn hình của người chia sẻ

Graphical user interface, application, Word

Description automatically generated

Hình 5.19 Giao diện màn hình chia sẻ màn hình

Graphical user interface, text, application

Description automatically generated

Hình 5.20 Giao diện rời phòng họp của chủ phòng

Graphical user interface, application

Description automatically generated

Hình 5.21 Giao diện rời phòng họp của người tham gia

Graphical user interface, text, application

Description automatically generated

Hình 5.22 Kết quả tệp xuất tin nhắn trong phòng họp

Graphical user interface, application

Description automatically generated

Hình 5.23 Kết quả tệp xuất thông tin cuộc họp

Graphical user interface, text

Description automatically generated

Hình 5.24 Kết quả tệp xuất thông tin thành viên cuộc họp

1. **Nhóm giao diện nhắn tin với bạn bè**

Graphical user interface, text, application, chat or text message

Description automatically generated

Hình 5.25 Giao diện nhắn tin với bạn bè

Graphical user interface, text, application, chat or text message

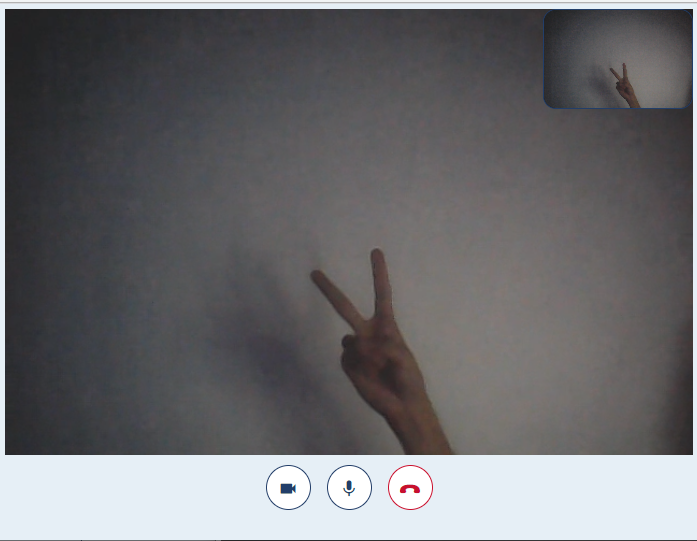
Description automatically generated

Hình 5.26 Giao diện nhắn tin với bạn bè loại họp thoại

Graphical user interface, text, application, chat or text message

Description automatically generated

Hình 5.27 Giao diện có cuộc gọi đến



Hình 5.28 Giao diện gọi video

Graphical user interface, text, application, chat or text message

Description automatically generated

Hình 5.29 Giao diện tìm kiếm người dùng

Graphical user interface, text, application

Description automatically generated

Hình 5.30 Giao diện kết quả tìm kiếm người dùng

Graphical user interface, text, application, chat or text message

Description automatically generated

Hình 5.31 Giao diện hộp thoại thông báo

Graphical user interface, text, application, chat or text message

Description automatically generated

Hình 5.32 Giao diện hộp thoại mở rộng

Graphical user interface, text, website

Description automatically generated

Hình 5.33 Giao diện trang cá nhân

Graphical user interface, text, application, chat or text message

Description automatically generated

Hình 5.34 Giao diện trang cá nhân người dùng

Graphical user interface, application

Description automatically generated

Hình 5.35 Giao diện cài đặt chung

Graphical user interface, application, website

Description automatically generated

Hình 5.36 Giao diện cài đặt phòng họp tĩnh

Graphical user interface, text, application, email

Description automatically generated

Hình 5.37 Giao diện cài đặt bảo mật

Graphical user interface, application, Word

Description automatically generated

Hình 5.38 Giao diện soạn thảo văn bản

# KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. **Kết luận**
2. **Kết quả đạt được**

Sau quá trình nghiên cứu và thực hiện đề tài “Xây dựng ứng dụng phòng họp trực tuyến và nhắn tin với bạn bè”, kết quả đã đạt được như sau:

* Hiểu cách thức thức hoạt động của một hệ thống trò chuyện trực tuyến.
* Xây dựng thành công ứng dụng trò chuyện trực tuyến trên nền tảng web, đáp ứng các yêu cầu đề tài đã đề ra.
* Biết và hiểu hơn các công nghệ phục vụ cho việc phát triển đề tài (NodeJs, ExpressJs, MongoDB, WebRTC API….).
* Giao diện ứng dụng thân thiện, dễ sử dụng.
* Thực hiện kiểm thử ứng dụng ở mức độ lập trình viên.
* Đã đưa ứng dụng lên hosting để công khai ứng dụng ra cộng đồng với tên miền:

<https://oh-chatapp.herokuapp.com/>

1. **Hạn chế tồn tại**

Ứng dụng hiện tại còn có một số hạn chế như sau:

* Chưa triển khai ứng dụng trên các nền tảng khác như mobile app, desktop app.
* Chưa hỗ trợ được hết các trình duyệt khác nhau.
* Chưa có một số chức năng nâng cao như: điều khiển màn hình, xóa nền video….

1. **Hướng phát triển**

* Tiếp tục phát triển ứng dụng ngày càng hoàn thiện hơn, xây dựng nhiều chức năng hơn.
* Xây dựng ứng dụng trên các nền tảng mobile app và desktop app và công khai ứng dụng ra cộng đồng.
* Nỗ lực phát triển ứng dụng để trở thành ứng dụng được nhà trường sử dụng trong việc giảng dạy online.

DANH MỤC TÀI LIỆU THAM KHẢO

[1] Hà Thị Thanh Ngà, trường đại học Nha Trang. Bài giảng Phân tích và thiết kế hệ thống thông tin.

[2] Internet Engineering Task Force (IETF). The WebSocket Protocol.

<<https://datatracker.ietf.org/doc/html/rfc6455>>. [Truy cập ngày 20/6/2021].

[3] Socket.IO. Introduction. <<https://socket.io/docs/v4/index.html>>. [Truy cập ngày 20/6/2021].

[4] Tutorialandexample, 6/11/2020. Node.js Event Loop.

<<https://www.tutorialandexample.com/node-js-event-loop/>>. [Truy cập ngày 20/6/2021].

[5] Wikipedia, 8/6/2021. Peer-to-peer. <<https://en.wikipedia.org/wiki/Peer-to-peer>>. [Truy cập ngày 20/6/2021].

[6] Mongodb. Introduction to MongoDB.

<<https://docs.mongodb.com/manual/introduction/>>. [Truy cập ngày 20/6/2021].

[7] Tutorialspoint. MongoDB – Overview.

<<https://www.tutorialspoint.com/mongodb/mongodb_overview.htm>>. [Truy cập ngày 20/6/2021].

[8] Mongodb. What is NoSQL? <<https://www.mongodb.com/nosql-explained>>. [Truy cập ngày 20/6/2021].

[9] Mongodb. MongoDB CRUD Operations.

<<https://docs.mongodb.com/manual/crud/#mongodb-crud-operations>>. [Truy cập ngày 20/6/2021].

[10] Freecodecamp, 11/02/2018. Introduction to Mongoose for MongoDB.

<<https://www.freecodecamp.org/news/introduction-to-mongoose-for-mongodb-d2a7aa593c57/>>. [Truy cập ngày 20/6/2021].

[11] Arvind Prajapati, 20/6/2018. What is middleware in the context of Nodejs.

<<https://www.livescript.in/2018/06/what-is-middleware-in-context-of-nodejs.html>>. [Truy cập ngày 20/6/2021].

[12] MDN Web Docs, 28/4/2021. Express Tutorial Part 4: Routes and controllers.

<<https://developer.mozilla.org/en-US/docs/Learn/Server-side/Express_Nodejs/routes>>. [Truy cập ngày 20/6/2021].