

Chương này sẽ tập trung vào việc khảo sát các hệ thống tương tự, phân tích các yêu cầu chức năng và phi chức năng của hệ thống Chat App. Từ việc khảo sát, đồ án sẽ đi sâu vào thiết kế tổng quan các chức năng thông qua biểu đồ Use Case, đồng thời đặc tả chi tiết các quy trình nghiệp vụ chính để làm cơ sở cho việc thiết kế và cài đặt ở các chương sau.

0.1 Khảo sát hiện trạng

Hiện nay, nhu cầu trao đổi thông tin trực tuyến đóng vai trò thiết yếu trong đời sống và công việc. Các ứng dụng nhắn tin tức thời (Instant Messaging) không chỉ đơn thuần là gửi văn bản mà còn tích hợp đa phương tiện, gọi thoại/video chất lượng cao và khả năng làm việc nhóm.

Qua khảo sát các ứng dụng phổ biến trên thị trường như Facebook Messenger, Zalo, và Telegram, nhóm nhận thấy các ứng dụng này đều có điểm chung là tập trung vào trải nghiệm người dùng mượt mà, tốc độ phản hồi nhanh (real-time) và khả năng xử lý dữ liệu đa phương tiện tốt. Tuy nhiên, việc xây dựng một hệ thống tương tự đòi hỏi sự kết hợp phức tạp của nhiều công nghệ hiện đại.

Dựa trên khảo sát và mục tiêu nghiên cứu, đồ án tập trung xây dựng một ứng dụng Chat trên nền web với các nhóm tính năng quan trọng sau:

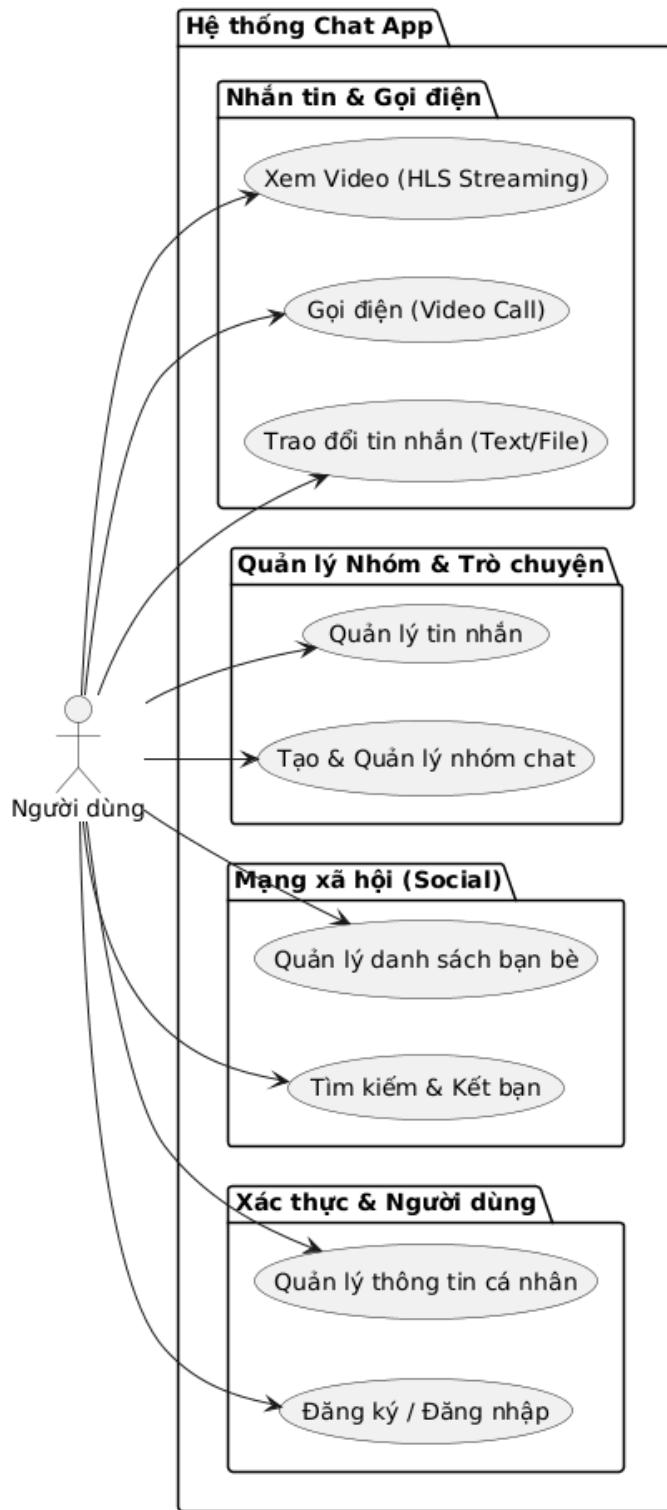
- **Xác thực & Người dùng:** Đảm bảo bảo mật tài khoản, hỗ trợ đăng nhập đa nền tảng (OAuth2 Google), quản lý thông tin cá nhân và trạng thái hoạt động (Online/Offline).
- **Mạng xã hội (Social):** Xây dựng mạng lưới kết nối thông qua tìm kiếm, kết bạn, quản lý danh sách bạn bè và tạo nhóm chat.
- **Nhắn tin (Messaging):** Hỗ trợ trao đổi tin nhắn văn bản, biểu cảm (Emoji), và đặc biệt là chia sẻ tài liệu, hình ảnh, video (hỗ trợ Streaming HLS cho video dung lượng lớn).
- **Gọi điện (Calling):** Tích hợp tính năng gọi Video Call trực tiếp giữa người dùng thông qua công nghệ WebRTC.
- **Hệ thống:** Các tính năng nền tảng như thông báo thời gian thực (Notification) và tìm kiếm toàn cục.

0.2 Tổng quan chức năng

Phần này tóm tắt các nhóm chức năng chính mà hệ thống hướng tới, đáp ứng nhu cầu giao tiếp toàn diện của người dùng. Hệ thống không chỉ cung cấp các tính năng nhắn tin cơ bản mà còn tích hợp các tiện ích nâng cao như gọi video chất lượng cao, chia sẻ tệp tin dung lượng lớn và quản lý nhóm hiệu quả. Các chức năng được chia thành các phân hệ rõ ràng để thuận tiện cho việc thiết kế và phát triển.

0.2.1 Biểu đồ use case tổng quát

Hệ thống được thiết kế xoay quanh tác nhân chính là **Người dùng (User)**. Dưới đây là biểu đồ Use Case tổng quát mô tả các chức năng chính của hệ thống:



Hình 0.1: Biểu đồ Use Case tổng quát của hệ thống

Danh sách các tác nhân:

- **Người dùng (User)**: Là người đã đăng ký tài khoản thành công và đăng nhập vào

hệ thống. Người dùng có toàn quyền sử dụng các chức năng giao tiếp và quản lý cá nhân.

Mô tả các nhóm Use Case chính:

1. Nhóm Xác thực (Authentication):

- Đăng ký, Đăng nhập (tài khoản thường và Google OAuth2).
- Quên mật khẩu, Đổi mật khẩu.
- Cập nhật Profile (Avatar, thông tin cá nhân) và quản lý trạng thái Online/Offline.

2. Nhóm Bạn bè & Tìm kiếm (Social):

- Tìm kiếm người dùng khác trong hệ thống.
- Gửi lời mời kết bạn, Chấp nhận hoặc Từ chối lời mời.
- Xem danh sách bạn bè và Hủy kết bạn.

3. Nhóm Nhóm & Trò chuyện (Group & Chat):

- Tạo cuộc hội thoại mới hoặc mở cuộc hội thoại đã có.
- Tạo nhóm chat mới, xem danh sách nhóm.
- Quản lý thành viên nhóm (Thêm, Xóa thành viên).
- Xóa cuộc hội thoại lịch sử.

4. Nhóm Nhắn tin & Gọi điện (Messaging & Calling):

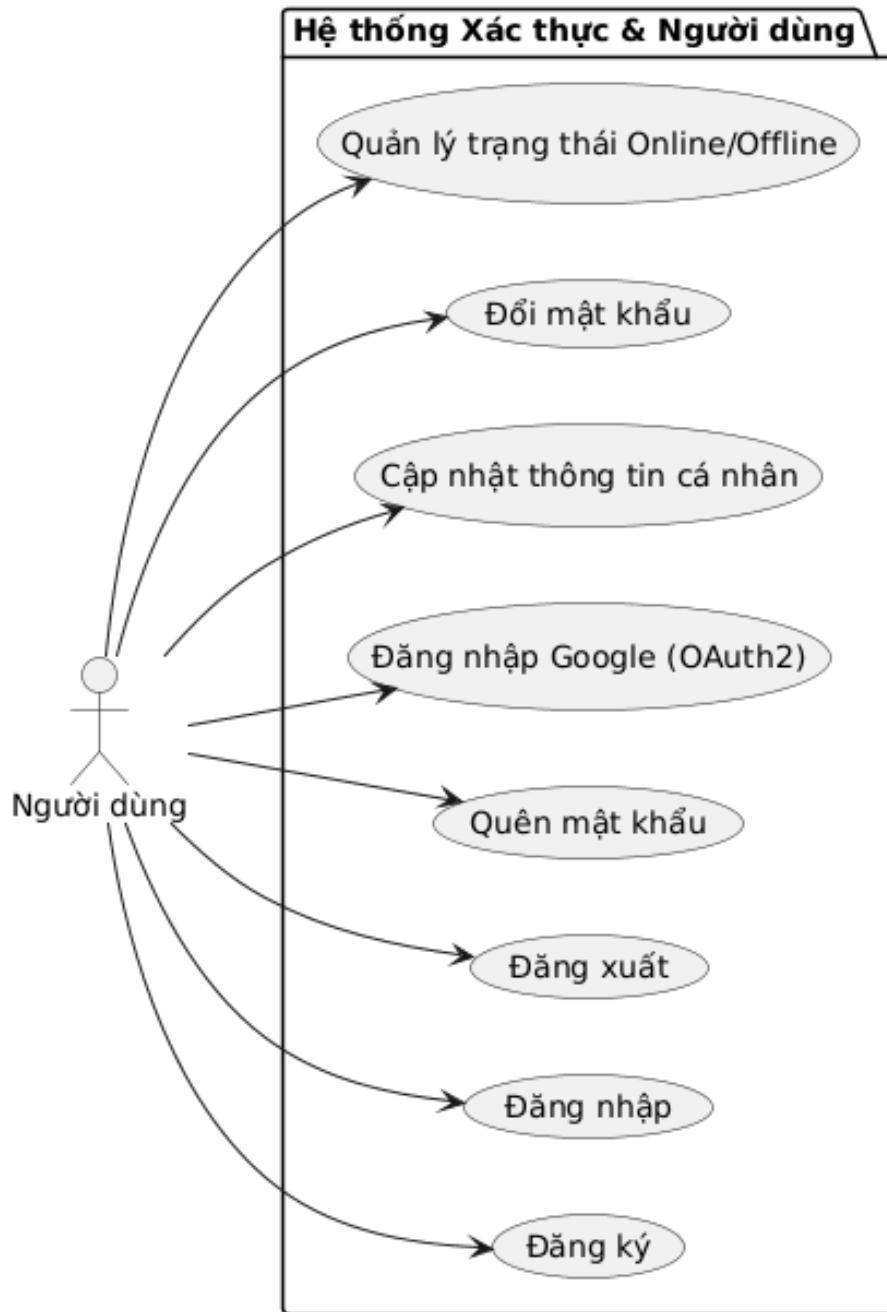
- Gửi nhận tin nhắn văn bản (Text), biểu tượng cảm xúc (Emoji).
- Gửi nhận tệp tin đa phương tiện: Ảnh, Video, Audio.
- Xem video chất lượng cao thông qua luồng phát trực tuyến (HLS Streaming).
- Thực hiện cuộc gọi Video trực tuyến (Video Call) sử dụng WebRTC.

0.2.2 Biểu đồ use case phân rã

Để làm rõ hơn các chức năng của hệ thống, phần này sẽ trình bày chi tiết các biểu đồ use case phân rã cho từng phân hệ chính.

a, Phân hệ Xác thực & Người dùng

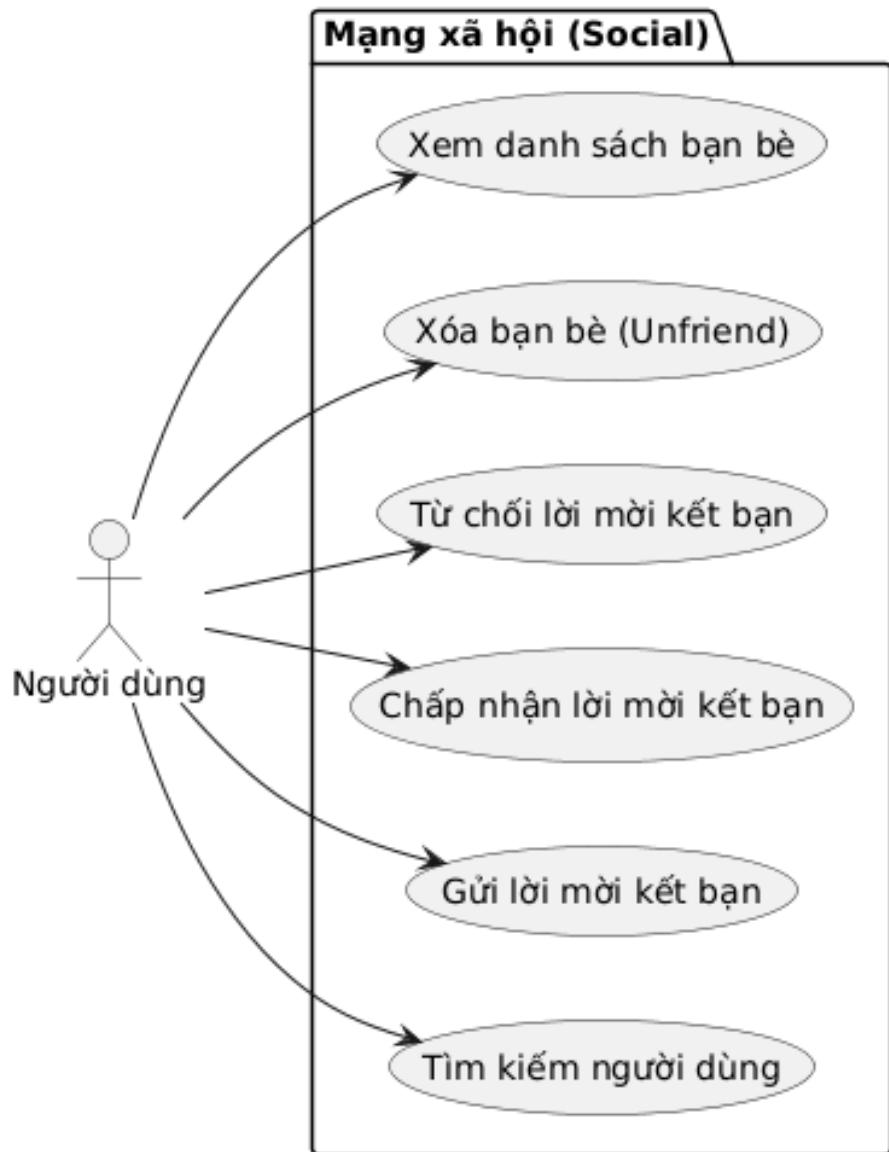
Phân hệ này quản lý việc truy cập và thông tin cá nhân của người dùng. Các chức năng bao gồm Đăng ký, Đăng nhập (hỗ trợ cả tài khoản thường và Google OAuth2), Đăng xuất, Quên mật khẩu. Ngoài ra, người dùng có thể cập nhật thông tin cá nhân (Avatar, thông tin cơ bản) và quản lý trạng thái online/offline của mình.



Hình 0.2: Biểu đồ Use Case phân rã: Xác thực & Người dùng

b, Phân hệ Mạng xã hội (Social)

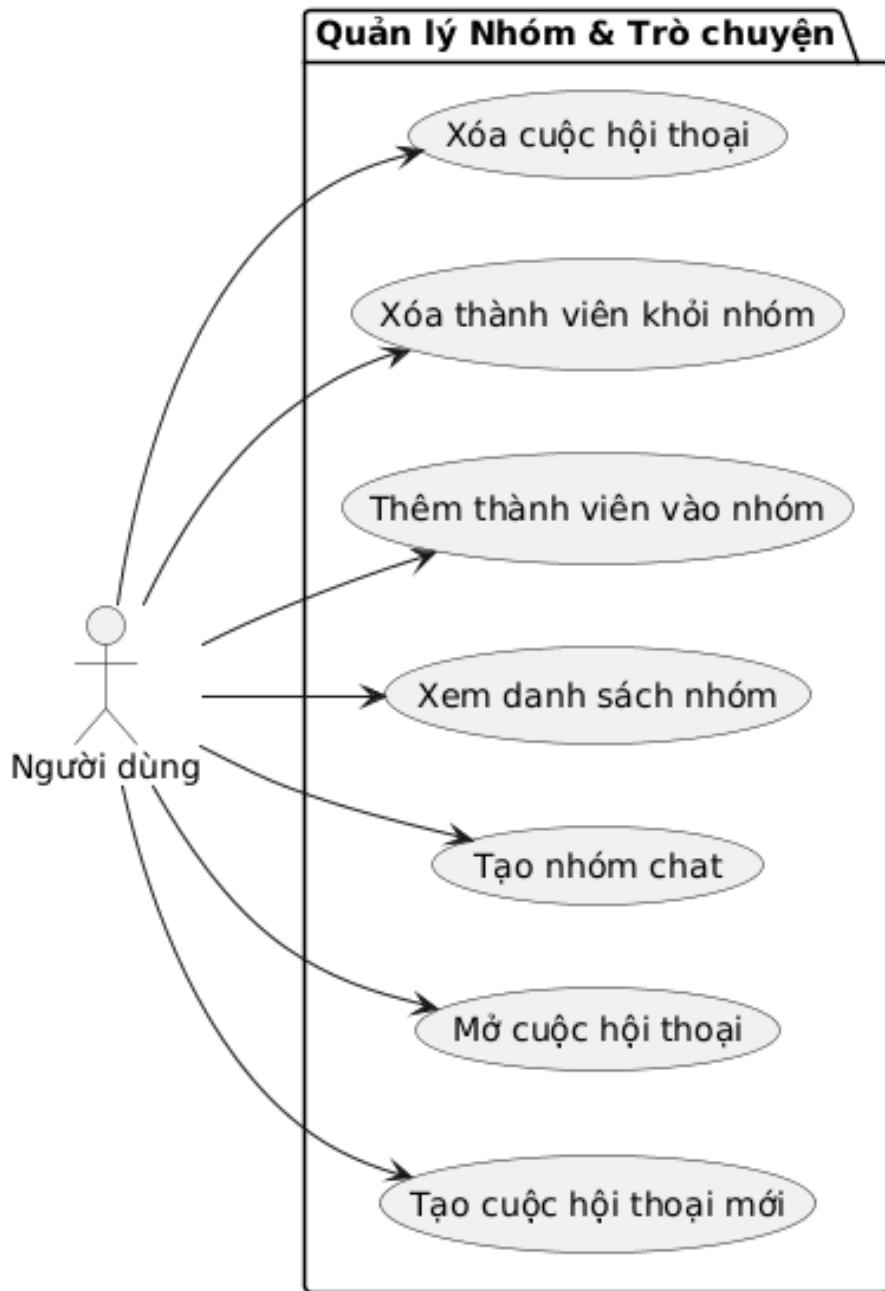
Phân hệ này tập trung vào tính năng kết nối giữa các người dùng. Người dùng có thể tìm kiếm người khác trong hệ thống, gửi lời mời kết bạn, và quản lý danh sách bạn bè (chấp nhận, từ chối hoặc hủy kết bạn).



Hình 0.3: Biểu đồ Use Case phân rã: Mạng xã hội

c, Phân hệ Quản lý Nhóm & Trò chuyện

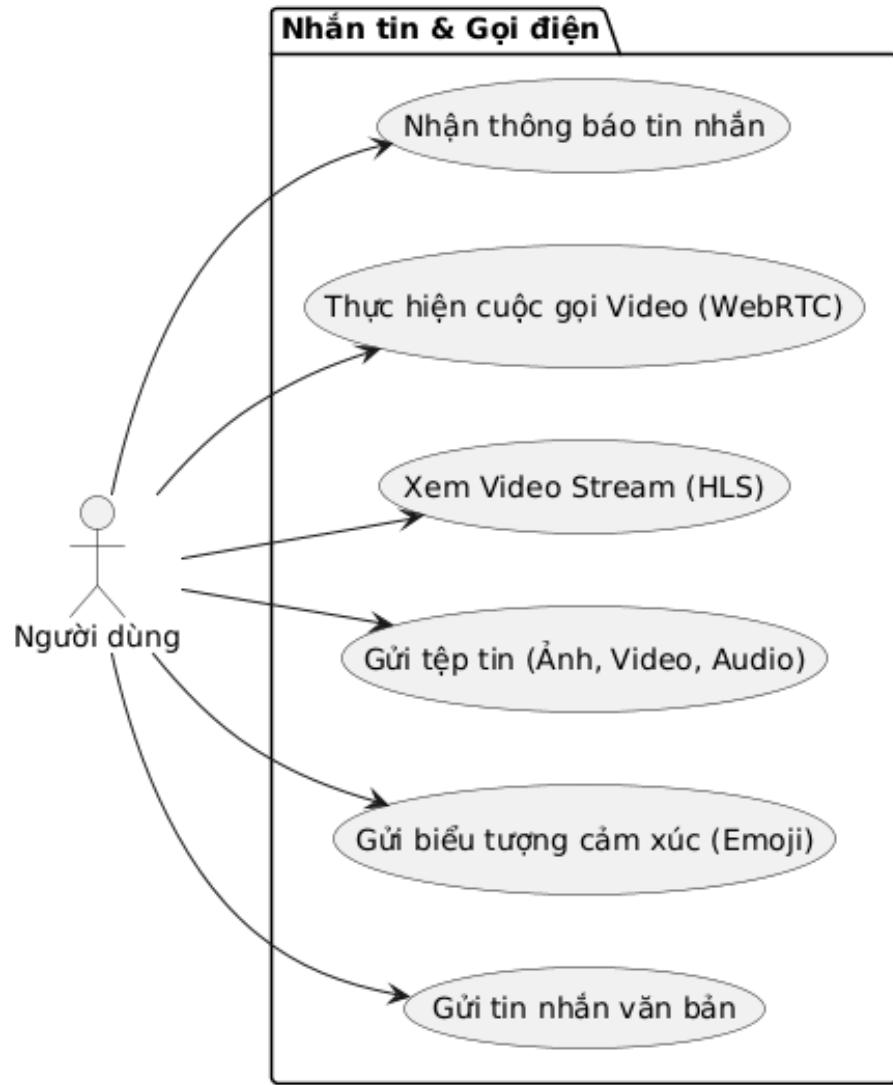
Phân hệ này cho phép người dùng quản lý các cuộc hội thoại và nhóm chat. Người dùng có thể tạo cuộc hội thoại mới, xem danh sách nhóm, tạo nhóm mới, và quản lý thành viên trong nhóm (thêm hoặc xóa thành viên).



Hình 0.4: Biểu đồ Use Case phân rã: Quản lý Nhóm & Trò chuyện

d, Phân hệ Nhắn tin & Gọi điện

Đây là phân hệ cốt lõi của ứng dụng, cung cấp khả năng trao đổi thông tin đa phương tiện. Người dùng có thể gửi tin nhắn văn bản, emoji, và các tệp tin (ảnh, video, audio). Hệ thống hỗ trợ xem video chất lượng cao qua HLS streaming và thực hiện cuộc gọi video thời gian thực qua WebRTC.



Hình 0.5: Biểu đồ Use Case phân rã: Nhắn tin & Gọi điện

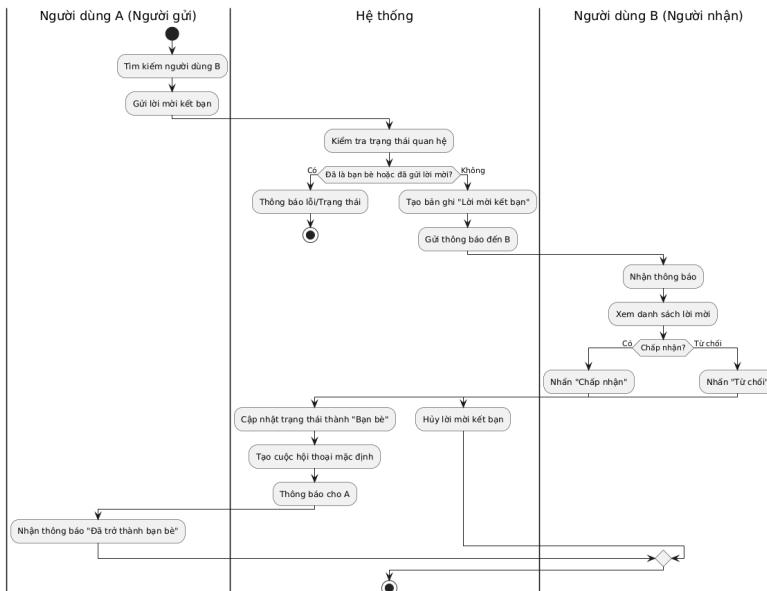
0.2.3 Quy trình nghiệp vụ

Nếu sản phẩm/hệ thống cần xây dựng có quy trình nghiệp vụ quan trọng/đáng chú ý, sinh viên cần mô tả và vẽ biểu đồ hoạt động minh họa quy trình nghiệp vụ đó. Sinh viên lưu ý đây không phải là luồng sự kiện của từng use case, mà là luồng hoạt động kết hợp nhiều use case để thực hiện một nghiệp vụ nào đó.

Đồ án tập trung vào hai quy trình nghiệp vụ tiêu biểu đại diện cho logic nghiệp vụ (Kết bạn) và logic kỹ thuật phức tạp (Gọi Video).

a, Quy trình Kết bạn (Friend Request Process)

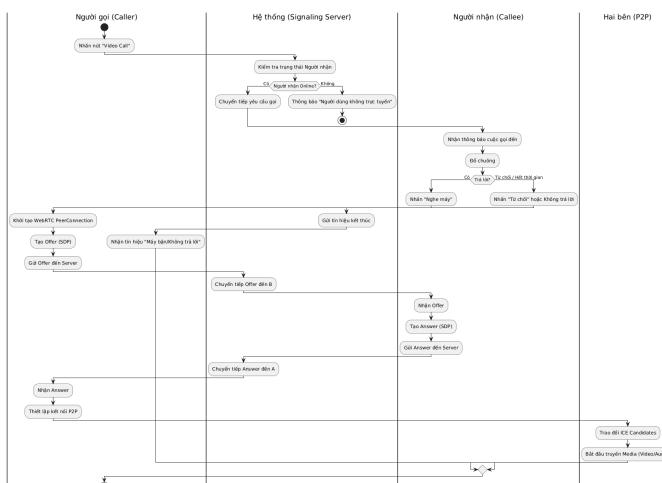
Đây là quy trình cơ bản để mở rộng mạng lưới xã hội của người dùng. Quy trình bắt đầu từ việc người dùng tìm kiếm, gửi lời mời, và kết thúc khi hai người trở thành bạn bè và có thể nhắn tin cho nhau.



Hình 0.6: Biểu đồ hoạt động: Quy trình Kết bạn

b, Quy trình thực hiện cuộc gọi Video (Video Call Process)

Quy trình này minh họa sự tương tác phức tạp giữa hai người dùng và hệ thống (Signaling Server) để thiết lập một phiên kết nối thời gian thực thông qua WebRTC.



Hình 0.7: Biểu đồ hoạt động: Quy trình gọi Video Call

0.3 Đặc tả chức năng

Sinh viên lựa chọn từ 4 đến 7 use case quan trọng nhất của đồ án để đặc tả chi tiết. Mỗi đặc tả bao gồm ít nhất các thông tin sau: (i) Tên use case, (ii) Luồng sự kiện (chính và phát sinh), (iii) Tiền điều kiện, và (iv) Hậu điều kiện. Sinh viên chỉ vẽ bổ sung biểu đồ hoạt động khi đặc tả use case phức tạp.

0.3.1 Đặc tả use case A

0.3.2 Đặc tả use case B

0.4 Yêu cầu phi chức năng

Các yêu cầu phi chức năng đóng vai trò quan trọng trong việc đảm bảo chất lượng và trải nghiệm người dùng của hệ thống. Dưới đây là các yêu cầu chi tiết:

- **Hiệu năng (Performance):**

- Hệ thống phải đảm bảo tính thời gian thực (Real-time). Độ trễ khi gửi/nhận tin nhắn không được vượt quá 200ms trong điều kiện mạng ổn định.
- Video streaming phải mượt mà, hỗ trợ tự động điều chỉnh chất lượng (Adaptive Bitrate Streaming) tùy theo băng thông mạng của người dùng.
- Thời gian tải trang ban đầu không quá 3 giây.

- **Bảo mật (Security):**

- Tài khoản người dùng phải được bảo vệ, mật khẩu lưu trữ trong cơ sở dữ liệu phải được mã hóa (hashing).
- Giao tiếp giữa Client và Server phải được mã hóa thông qua giao thức HTTPS và WSS (Secure WebSocket).
- Cơ chế xác thực qua Token (JWT) phải đảm bảo tính toàn vẹn và hết hạn hợp lý.
- Dữ liệu riêng tư của người dùng không được rò rỉ hoặc truy cập trái phép.

- **Tính khả dụng và Trải nghiệm người dùng (Usability):**

- Giao diện phải thân thiện, dễ sử dụng, hỗ trợ Responsive để hiển thị tốt trên cả máy tính và thiết bị di động.
- Hệ thống phải có cơ chế phản hồi rõ ràng cho các hành động của người dùng (ví dụ: thông báo loading, thông báo lỗi, thông báo thành công).

- **Độ tin cậy (Reliability):**

- Hệ thống hoạt động ổn định 24/7.
- Dữ liệu tin nhắn và tệp tin không được thất lạc trong quá trình truyền tải.
- Hệ thống có khả năng phục hồi sau khi gặp sự cố gián đoạn kết nối mạng.

- **Khả năng bảo trì và mở rộng (Maintainability & Scalability):**

- Mã nguồn phải được tổ chức rõ ràng theo kiến trúc định sẵn (ví dụ: Client-Server, MVC) để dễ dàng nâng cấp và sửa lỗi.
- Hệ thống có khả năng mở rộng để hỗ trợ thêm tính năng mới mà không làm ảnh hưởng đến các chức năng hiện có.

- Sử dụng các công nghệ containerization (Docker) để dễ dàng triển khai trên các môi trường khác nhau.