**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG THƯƠNG THÀNH PHỐ HỒ CHÍ MINH**

**A blue and white logo with a book and a square cap

Description automatically generated**

**BÁO CÁO ĐỒ ÁN**

**CÔNG NGHỆ JAVA**

**ĐỀ TÀI: Xây dựng ứng dụng chat**

**GVHD: Trần Trương Tấn Phát**

**NHÓM SINH VIÊN THỰC HIỆN:**

Bạch Minh Tuyên - 2001216289

Lâm Tấn Trọng - 2001216247

Nguyễn Thị Xuân Lan - 2001215904

Nguyễn Hoài Nam - 2001215971

**Thành phố Hồ Chí Minh, tháng 5 năm 2024**

**MỤC LỤC**

[CHƯƠNG I. TỔNG QUAN MÔ HÌNH CLIENT-SERVER VÀ SỬ DỤNG SOCKET 1](#_Toc167828894)

[1.1. Khái niệm 1](#_Toc167828895)

[1.2. Khái quát về ứng dụng chat và công nghệ sử dụng 2](#_Toc167828896)

[1.2.1. Ứng dụng chat 2](#_Toc167828897)

[1.2.2. Công nghệ sử dụng 3](#_Toc167828898)

[CHƯƠNG II. PHÂN TÍCH VÀ ĐẶC TẢ YÊU CẦU 4](#_Toc167828899)

[2.1. Yêu cầu và phân tích 4](#_Toc167828900)

[2.2. Yêu cầu chức năng 6](#_Toc167828901)

[2.3. Mô tả nghiệp vụ 8](#_Toc167828902)

[2.4. Thiết kế hệ thống 8](#_Toc167828903)

[2.4.1. Sơ đồ use case 8](#_Toc167828904)

[2.4.2. Thiết kế cơ sở dữ liệu 13](#_Toc167828905)

[CHƯƠNG III. MỘT SỐ GIAO DIỆN ỨNG DỤNG CHAT 17](#_Toc167828906)

[3.1.Giao diện … 17](#_Toc167828907)

[3.2. Giao diện 17](#_Toc167828908)

[3.3. Giao diện 17](#_Toc167828909)

[CHƯƠNG IV. KẾT LUẬN 17](#_Toc167828910)

[4.1. Kết quả đạt được 17](#_Toc167828911)

[4.1.1.Ưu điểm 17](#_Toc167828912)

[4.1.2. Nhược diểm 17](#_Toc167828913)

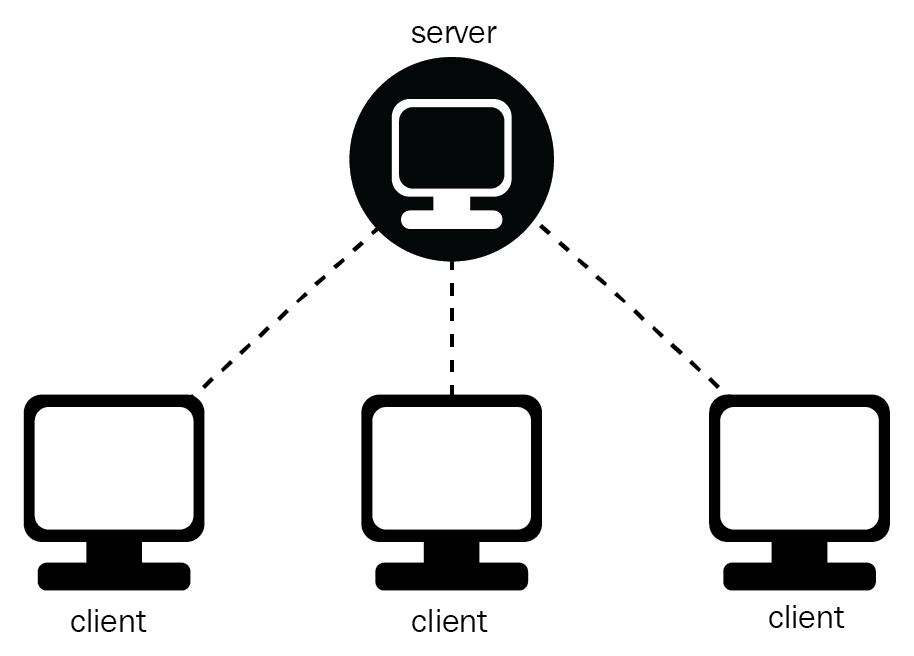
[4.2. Hướng phát triển 17](#_Toc167828914)

[TÀI LIỆU THAM KHẢO 17](#_Toc167828915)

# CHƯƠNG I. TỔNG QUAN MÔ HÌNH CLIENT-SERVER VÀ SỬ DỤNG SOCKET

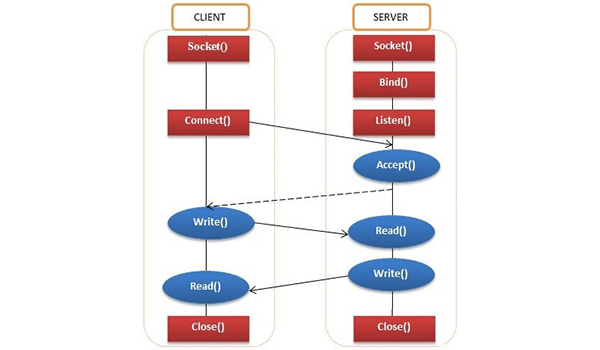
* 1. Khái niệm

Mô hình Client-Server (hay còn gọi là mô hình máy khách - máy chủ) là mô hình mà ở đây, máy khách là các máy tính, các thiết bị điện tử như máy in, điện thoại bàn, máy fax,… các máy khách này gửi yêu cầu đến server (hay còn được gọi là máy chủ). Máy chủ, xử lý các yêu cầu đó và trả về kết quả. Là một kiểu kiến trúc phổ biến trong phát triển ứng dụng mạng, trong đó có một máy chủ (server) quản lý và cung cấp dịch vụ cho các máy khách (clients).



Socket được sử dụng để thiết lập kết nối giữa client và server. Sockets là một giao diện lập trình ứng dụng (API) cho việc gửi và nhận dữ liệu qua mạng .Socket là một giao diện lập trình ứng dụng mạng được dùng để truyền và nhận dữ liệu trên internet. Giữa hai chương trình chạy trên mạng cần có một liên kết giao tiếp hai chiều để kết nối với nhau. Điểm cuối (endpoint) của liên kết này được gọi là socket, cho phép người dùng kết nối các máy tính truyền tải và nhận dữ liệu từ máy tính thông qua mạng. Vì vậy, hiểu đơn giản thì socket là thiết bị truyền thông hai chiều gửi và nhận dữ liệu từ máy khác.

Để xây dựng được một ứng dụng chat bằng Socket thì cần phải hiểu rõ cách thức hoạt động của nó, có rất nhiều loại Socket, nhưng ở đây mình sẽ lấy Stream Socket làm ví dụ, và sẽ dùng nó để lập trình nên phần mềm chat giữa client và server, Stream Socket hoạt động như sau:



Stream Socket hay còn gọi là socket hướng kết nối, là socket hoạt động thông qua giao thức TCP (Transmission Control Protocol). Stream Socket chỉ hoạt động khi server và client đã kết nối với nhau thông qua một IP và Port, dữ liệu truyền đi được đảm bảo truyền đến đúng nơi nhận, đúng thứ tự với thời gian nhanh chóng.

* 1. Khái quát về ứng dụng chat và công nghệ sử dụng
     1. Ứng dụng chat

Ứng dụng chat là một công cụ giao tiếp trực tuyến cho phép người dùng gửi và nhận tin nhắn văn bản, hình ảnh, video, và nhiều loại tệp khác. Các ứng dụng chat hiện đại thường hỗ trợ cả giao tiếp thời gian thực (real-time) và lưu trữ lịch sử hội thoại.

Bao gồm thành phần chính:

+ Giao diện người dùng: cửa sổ chat hiển thị cuộc hội thoại bao gồm tin nhắn văn bản , hình ảnh và các tệp đính kèm, thanh nhập liệu, danh sách bạn bè và nhóm.

+ Máy chủ : xử lí tin nhắn , quản lí người dùng, quản lý nhóm chat

+ Cơ sở dữ liệu : lưu trữ lịch sử tin nhắn người dùng, thông tin người dùng.

+Giao thức giao tiếp : Socket cho phép giao tiếp thời gian thực giữa máy khách và máy chủ.

* + 1. Công nghệ sử dụng

+ Sử dụng ngôn ngữ Java

Java là một ngôn ngữ lập trình hướng đối tượng mạnh mẽ và phổ biến, được phát triển bởi Sun Microsystems vào năm 1995 và hiện nay được quản lý bởi Oracle Corporation. Java được thiết kế để chạy trên nhiều nền tảng khác nhau, từ máy tính để bàn đến các hệ thống lớn và các thiết bị di động.

Với tính năng đơn giản thiết kế dễ học và dễ sử dụng , cú pháp rõ ràng và dễ hiểu. Hỗ trợ các nguyên tắc OOP như kết thừa, đa hình , đóng gói và trừu tượng giúp tổ chức và quản lý mã nguồn hiệu quả. Độc lập nền tảng với khẩu hiệu “Viết một lần, chạy mọi nơi” (Write Once, Run Anywhere -WORA), bảo mật , đa luồng , hiệu năng cao , thư viện phong phú cung cấp nhiều chức năng sẵn có cho lập trình viên.

Ngôn ngữ java được sử dụng rỗng rãi trong nhiều lĩnh vực như ứng dụng web , ứng dụng di động, ứng dụng doanh nghiệp , ứng dụng desktop và phát triển game. Ngoài ra , ava có một cộng đồng lập trình viên rộng lớn và rất nhiều tài liệu, khóa học, và diễn đàn hỗ trợ. Điều này giúp các lập trình viên dễ dàng học hỏi và giải quyết các vấn đề khi làm việc với Java.

Java là một ngôn ngữ lập trình mạnh mẽ, đa dụng và phổ biến, lý tưởng cho nhiều loại ứng dụng khác nhau, từ các ứng dụng nhỏ đến các hệ thống lớn và phức tạp. Với những đặc điểm nổi bật như độc lập nền tảng, bảo mật cao, và hỗ trợ lập trình hướng đối tượng, Java vẫn tiếp tục là lựa chọn hàng đầu cho các lập trình viên và các tổ chức phát triển phần mềm trên toàn thế giới.

+ Sử dụng kết hợp với Microsoft SQL Server

Microsoft SQL Server: Là một hệ quản trị cơ sở dữ liệu quan hệ (RDBMS) của Microsoft, cung cấp các giải pháp quản lý dữ liệu mạnh mẽ, bảo mật và hiệu suất cao.

Khi kết hợp, chúng tạo nên một giải pháp mạnh mẽ để phát triển các ứng dụng có khả năng quản lý và xử lý dữ liệu hiệu quả.

Kết hợp Java với Microsoft SQL Server mang lại nhiều lợi ích cho các ứng dụng doanh nghiệp, từ khả năng mở rộng, tính bảo mật, đến hiệu suất cao. Sử dụng JDBC và driver JDBC của Microsoft, lập trình viên có thể dễ dàng kết nối, truy vấn và thao tác dữ liệu trên SQL Server từ các ứng dụng Java.

+ Công cụ hỗ trợ : Eclipse , Netbean

Eclipse là một môi trường phát triển tích hợp (IDE) mã nguồn mở và miễn phí, được sử dụng rộng rãi để phát triển các ứng dụng Java và các ngôn ngữ lập trình khác.

NetBeans là một IDE mã nguồn mở và miễn phí, được Oracle tài trợ, chuyên dụng cho phát triển các ứng dụng Java, nhưng cũng hỗ trợ các ngôn ngữ khác.

# CHƯƠNG II. PHÂN TÍCH VÀ ĐẶC TẢ YÊU CẦU

* 1. Yêu cầu và phân tích
* Tính năng của ứng dụng chat:
* Chat theo từng người: Cho phép người dùng trò chuyện một cách riêng tư với một người khác.
* Chat theo nhóm: Người dùng có thể tham gia vào các nhóm chat cụ thể và trò chuyện với nhiều người cùng một lúc.
* Gửi tin nhắn văn bản: Cung cấp chức năng để gửi và nhận tin nhắn dạng văn bản.
* Gửi file: Cho phép người dùng gửi và nhận các file (ảnh, video, tài liệu văn bản...) trong quá trình trò chuyện.
* Gửi media: Hỗ trợ gửi và nhận các loại phương tiện khác nhau như hình ảnh, video, âm thanh.
* Cơ chế hoạt động:
* Khi một client kết nối đến server, server chấp nhận kết nối và tạo một thread hoặc tiến trình mới để xử lý các yêu cầu từ client đó.
* Mỗi client sẽ gửi các yêu cầu như gửi tin nhắn, gửi file, tham gia nhóm qua kết nối socket.
* Server sẽ đảm bảo phân phối tin nhắn đến các client phù hợp, bao gồm cả việc quản lý tin nhắn riêng tư và tin nhắn nhóm.
* Khi người dùng gửi file hoặc media, dữ liệu sẽ được truyền qua socket tương ứng và server sẽ chịu trách nhiệm lưu trữ và phân phối dữ liệu đó đến các client liên quan.
* Bảo mật:
* Đảm bảo tính riêng tư và bảo mật thông tin của người dùng trong quá trình trò chuyện.
* Sử dụng các biện pháp bảo mật như mã hóa dữ liệu để ngăn chặn việc đánh cắp thông tin trong quá trình truyền dữ liệu.
* Giao diện người dùng (UI):
* Cần có giao diện thân thiện, dễ sử dụng cho cả client và server.
* Cung cấp các tính năng như tạo nhóm mới, thêm thành viên vào nhóm, quản lý tin nhắn đã gửi/nhận...
* Quản lý kết nối và phiên làm việc:
* Đảm bảo ứng dụng có khả năng xử lý nhiều kết nối từ các client đồng thời.
* Xác định và duy trì các phiên làm việc (session) cho mỗi client để theo dõi trạng thái của họ và lưu trữ lịch sử trò chuyện.
* Tương thích đa nền tảng:
* Phát triển ứng dụng sao cho có thể hoạt động trên nhiều nền tảng, bao gồm desktop, web và di động.
  1. Yêu cầu chức năng
* Đăng nhập và Đăng ký:
* Cho phép người dùng đăng nhập vào hệ thống bằng tài khoản và mật khẩu hoặc thông qua các phương thức xác thực khác như OAuth.
* Cung cấp chức năng đăng ký tài khoản mới nếu người dùng chưa có tài khoản.
* Quản lý Danh bạ:
* Cho phép người dùng thêm, sửa đổi và xóa các liên hệ trong danh bạ của họ.
* Hiển thị trạng thái trực tuyến/offline của các liên hệ.
* Chat Riêng tư:
* Cho phép người dùng trò chuyện một cách riêng tư với một hoặc nhiều người khác.
* Hỗ trợ gửi và nhận tin nhắn văn bản, file và media trong cuộc trò chuyện riêng tư.
* Tạo và Quản lý Nhóm:
* Cho phép người dùng tạo nhóm chat và thêm thành viên vào nhóm.
* Hỗ trợ chức năng quản lý nhóm như thay đổi tên nhóm, thêm hoặc xóa thành viên khỏi nhóm.
* Chat Nhóm:
* Cho phép người dùng dùng trò chuyện một cách riêng tư với một hoặc nhiều người khác.
* Hỗ trợ gửi và nhận tin nhắn văn bản, file và media trong cuộc trò chuyện nhóm.
* Gửi và Nhận Tin Nhắn:
* Cung cấp giao diện đơn giản để nhập và hiển thị tin nhắn.
* Hỗ trợ gửi và nhận tin nhắn văn bản, biểu tượng cảm xúc, sticker...
* Gửi và Nhận File và Media:
* Cho phép người dùng gửi và nhận các loại file (ảnh, video, tài liệu...) và media (âm thanh) trong các cuộc trò chuyện.
* Hỗ trợ xem trước nội dung của file và media trước khi tải xuống.
* Thông báo và Thông báo Công cụ:
* Cung cấp cơ chế thông báo cho người dùng khi có tin nhắn mới, lời mời nhóm, hoặc sự kiện khác liên quan đến các hoạt động trò chuyện.
* Hỗ trợ cài đặt thông báo theo ý muốn của người dùng.
* Quản lý Trạng thái và Hiện trạng:
* Hiển thị trạng thái online/offline của người dùng.
* Cho phép người dùng đặt trạng thái cá nhân và thông báo trạng thái cho liên hệ.
* Tìm kiếm và Lọc Tin Nhắn:
* Cung cấp chức năng tìm kiếm để người dùng có thể tìm kiếm tin nhắn, liên hệ, hoặc nhóm.
* Hỗ trợ lọc tin nhắn theo thời gian, người gửi hoặc nội dung.
* Bảo mật và Quyền riêng tư:
* Đảm bảo tính bảo mật của dữ liệu truyền qua mạng bằng cách sử dụng giao thức mã hóa và bảo mật.
* Cung cấp các cài đặt quyền riêng tư để người dùng có thể kiểm soát ai có thể truy cập và gửi tin nhắn cho họ.
* Lịch sử Trò chuyện:
* Lưu trữ và hiển thị lịch sử trò chuyện cho mỗi cuộc trò chuyện riêng tư và nhóm.
* Cho phép người dùng xem lại và tải xuống lịch sử trò chuyện.
* Giao diện người dùng thân thiện:
* Xây dựng giao diện người dùng đơn giản, dễ sử dụng và thân thiện với người dùng.
* Hỗ trợ các tính năng như chế độ ban đêm, đa ngôn ngữ... để cải thiện trải nghiệm người dùng.
  1. Mô tả nghiệp vụ
* Người dùng mới tạo tài khoản để có thể sử dụng ứng dụng chat.

Quy trình:

Người dùng truy cập vào giao diện đăng ký.

Người dùng nhập thông tin cần thiết như tên người dùng, mật khẩu.

Hệ thống kiểm tra tính hợp lệ của thông tin (tên người dùng chưa tồn tại, mật khẩu và xác nhận mật khẩu trùng khớp).

Hệ thống lưu trữ thông tin người dùng mới vào cơ sở dữ liệu.

Người dùng nhận được thông báo đăng ký thành công và có thể đăng nhập vào hệ thống.

* Đăng nhập

Mô tả nghiệp vụ:

Người dùng nhập thông tin tài khoản để truy cập vào ứng dụng chat.

Quy trình:

Người dùng truy cập vào giao diện đăng nhập.

Người dùng nhập tên tài khoản và mật khẩu.

Hệ thống kiểm tra tính hợp lệ của thông tin đăng nhập (so khớp với cơ sở dữ liệu).

Nếu thông tin hợp lệ, hệ thống cho phép người dùng truy cập vào giao diện chat.

Nếu thông tin không hợp lệ, hệ thống hiển thị thông báo lỗi.

* Gửi và nhận tin nhắn

Mô tả nghiệp vụ:

Người dùng có thể gửi và nhận tin nhắn trong thời gian thực với các người dùng khác.

Quy trình:

Người dùng nhập tin nhắn vào trường nhập tin nhắn và nhấn nút gửi.

Ứng dụng client gửi tin nhắn đến server thông qua socket.

Server nhận tin nhắn và xác định người nhận.

Server chuyển tiếp tin nhắn đến ứng dụng client của người nhận qua socket.

Ứng dụng client của người nhận hiển thị tin nhắn mới trong giao diện chat.

* Thêm bạn và quản lý danh sách bạn bè

Mô tả nghiệp vụ:

Người dùng có thể thêm người khác vào danh sách bạn bè để dễ dàng trò chuyện.

Quy trình:

Người dùng gửi yêu cầu kết bạn đến một người dùng khác.

Server lưu trữ yêu cầu kết bạn và gửi thông báo đến người nhận.

Người nhận xem xét và chấp nhận hoặc từ chối yêu cầu kết bạn.

Nếu yêu cầu được chấp nhận, server cập nhật danh sách bạn bè của cả hai người dùng.

Người dùng có thể xem danh sách bạn bè và bắt đầu trò chuyện với họ.

* Quản lý thông tin tài khoản

Mô tả nghiệp vụ:

Người dùng có thể cập nhật thông tin cá nhân và thay đổi mật khẩu.

Quy trình:

Người dùng truy cập vào giao diện cài đặt tài khoản.

Người dùng cập nhật thông tin cá nhân như tên, email, hoặc ảnh đại diện.

Người dùng có thể thay đổi mật khẩu bằng cách nhập mật khẩu hiện tại và mật khẩu mới.

Hệ thống xác thực thông tin và cập nhật cơ sở dữ liệu.

Người dùng nhận được thông báo cập nhật thành công.

* Nhận thông báo

Mô tả nghiệp vụ:

Người dùng nhận thông báo về tin nhắn mới, yêu cầu kết bạn, và các sự kiện khác.

Quy trình:

Khi có tin nhắn mới hoặc yêu cầu kết bạn, server gửi thông báo đến ứng dụng client của người dùng.

Ứng dụng client hiển thị thông báo trong giao diện người dùng.

Người dùng có thể xem và tương tác với thông báo, ví dụ như chấp nhận yêu cầu kết bạn hoặc mở tin nhắn mới.

* Đăng xuất

Mô tả nghiệp vụ:

Người dùng có thể đăng xuất khỏi tài khoản của mình.

Quy trình:

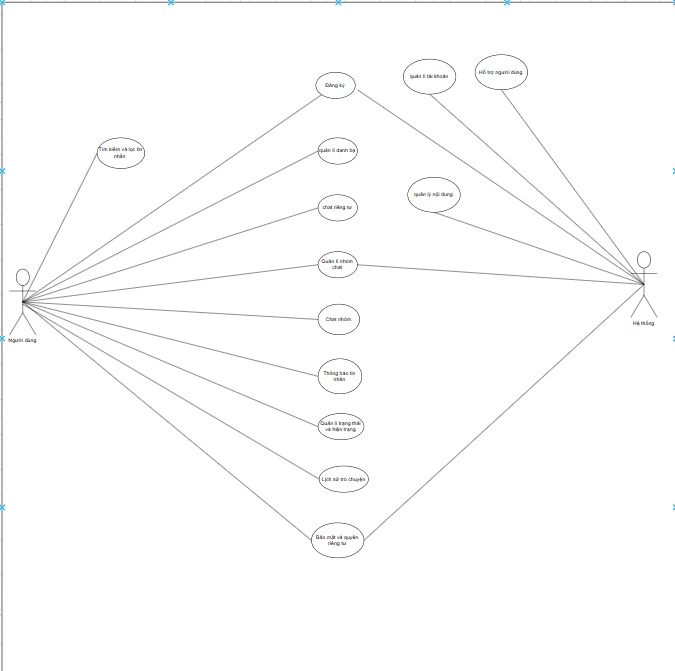
Người dùng nhấn nút đăng xuất trong giao diện cài đặt tài khoản.

Ứng dụng client gửi yêu cầu đăng xuất đến server.

Server hủy phiên làm việc của người dùng và cập nhật trạng thái trực tuyến.

Người dùng được đưa trở lại giao diện đăng nhập.

* 1. Thiết kế hệ thống
     1. Sơ đồ use case

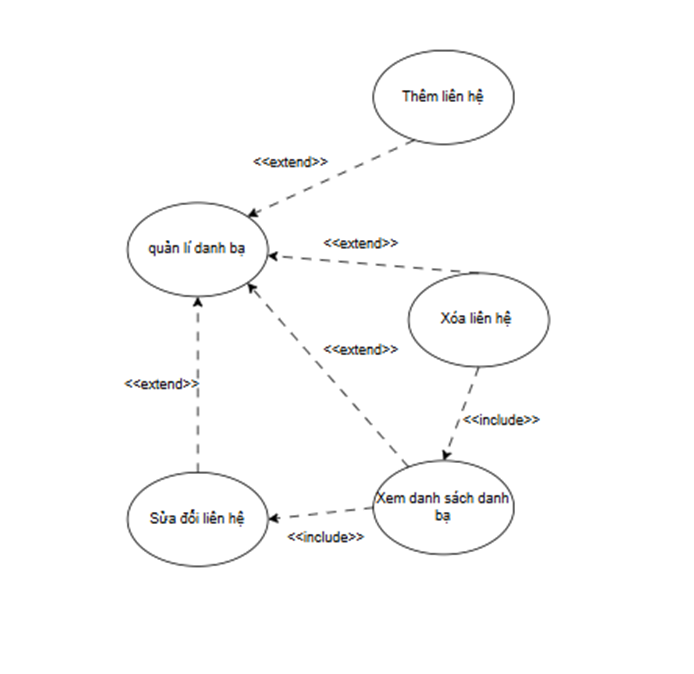


+ Usecase quản lý tài khoản

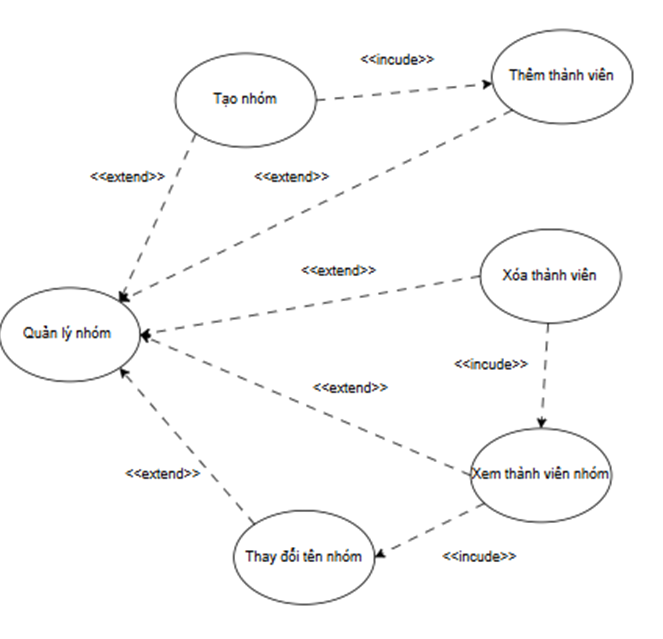
A diagram of a diagram

Description automatically generated

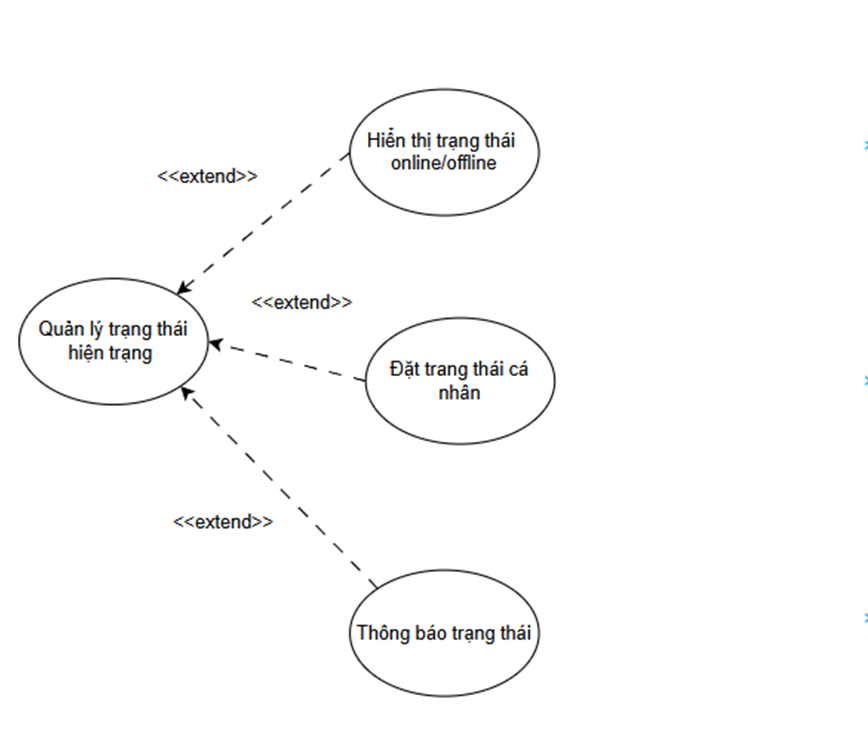
+ Use case quản lí danh bạ



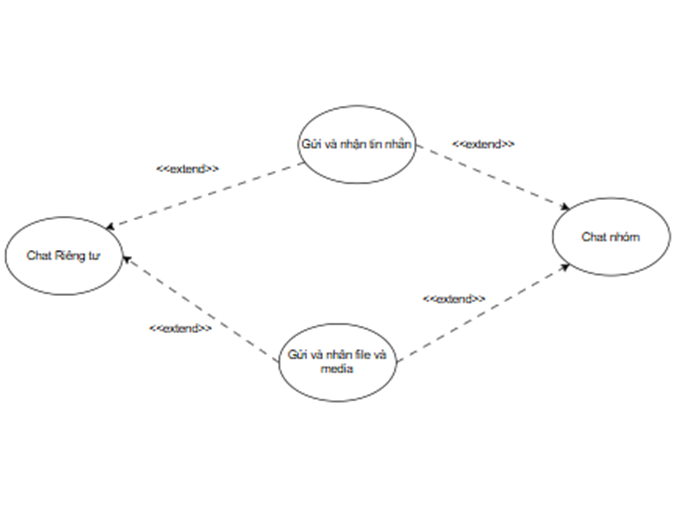
+ Use case quản lý nhóm



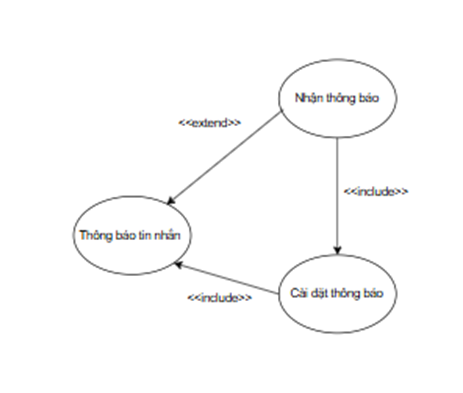
+ Use case quản lý trạng thái và hiện trạng



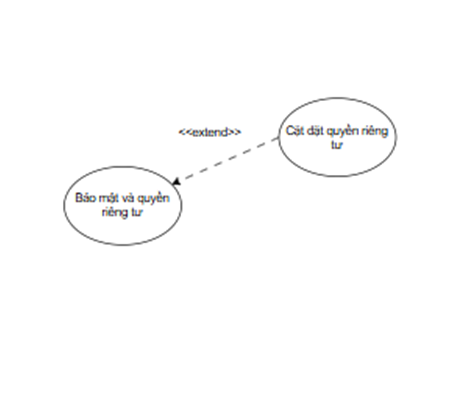
+ Use case chat riêng tư và chat nhóm



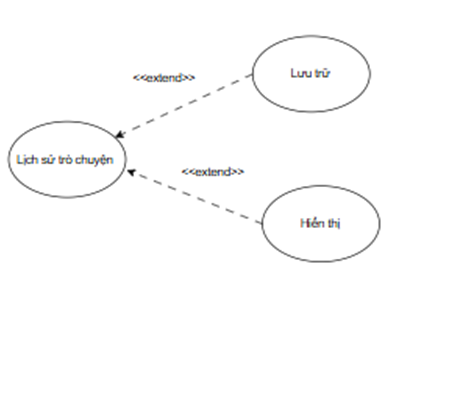
+ Use case thông báo tin nhắn

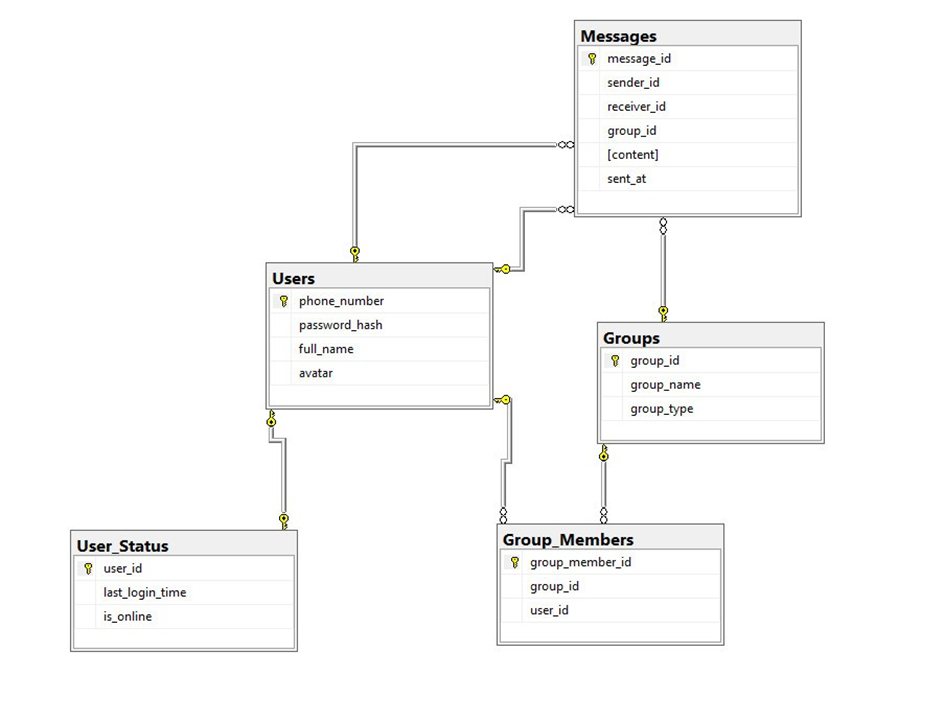


+ Use case bảo mật và quyền riêng tư



+ Use case lịch sử trò chuyện



* + 1. Thiết kế cơ sở dữ liệu
* Mô hình diagrams
* Đặc tả dữ liệu

+ Bảng Users

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Kích thước | Ràng buộc | Khuông dạng |
| Phone\_number | VARCHAR | 50 | Khóa chính | Văn Bản |
| password\_hash | VARCHAR | 255 |  | Văn Bản |
| full\_name | NVARCHAR | 50 |  | Văn Bản |
| avatar | NVARCHAR | 255 |  | Văn Bản |

+ Bảng User\_Status

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Kích thước | Ràng buộc | Khuông dạng |
| user\_id | VARCHAR | 50 | Khóa chính | Văn Bản |
| last\_login\_time | DATETIME2 | 0 |  |  |
| is\_online | BIT |  |  |  |

+Bảng Groups

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Kích thước | Ràng buộc | Khuông dạng |
| group\_id | INT |  | Khóa chính | Văn Bản |
| group\_name | NVARCHAR | 50 |  | Văn Bản |
| group\_type | NVARCHAR | 20 |  | Văn Bản |

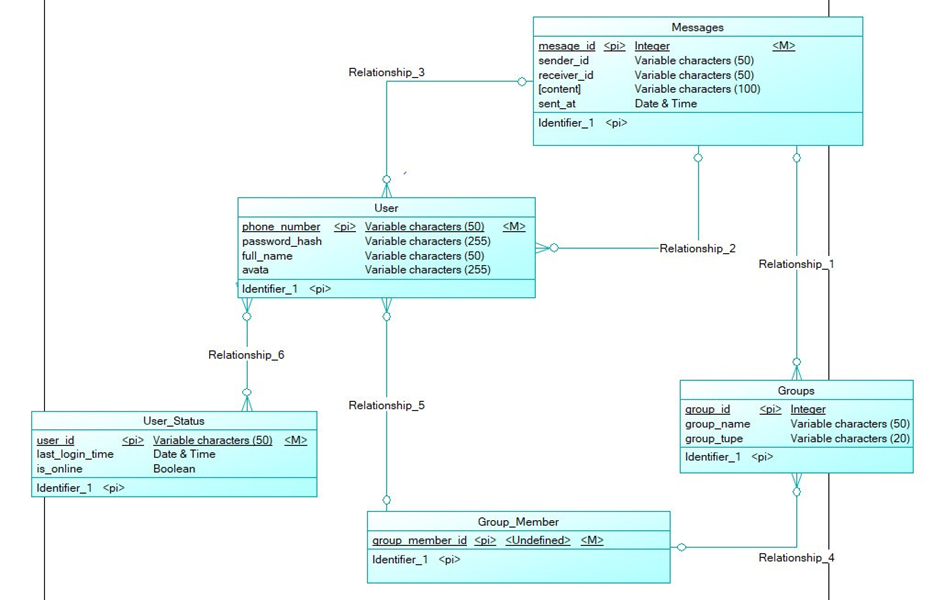
+ Bảng Group\_Members

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Kích thước | Ràng buộc | Khuông dạng |
| group\_member\_id | VARCHAR | 50 | Khóa chính | Văn Bản |
| group\_id | VARCHAR | 255 |  | Văn Bản |
| user\_id | NVARCHAR | 50 |  | Văn Bản |

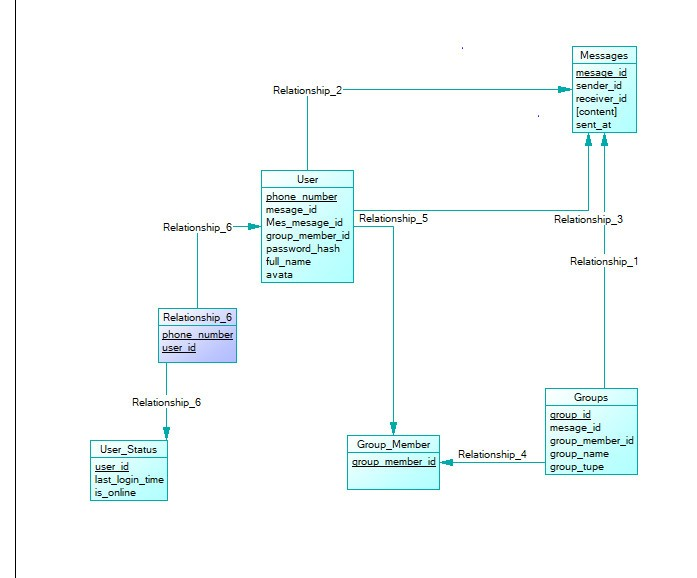
+ Bảng Messages

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Kích thước | Ràng buộc | Khuông dạng |
| message\_id | INT |  | Khóa chính | Kiểu số nguyên |
| sender\_id | VARCHAR | 50 |  | Văn Bản |
| receiver\_id | VARCHAR | 50 |  | Văn Bản |
| group\_id | INT |  |  | Kiểu số nguyên |
| content | VARCHAR | MAX |  | Văn bản |
| sent\_at | DATETIME2 | 0 |  | Kiểu ngày giờ |

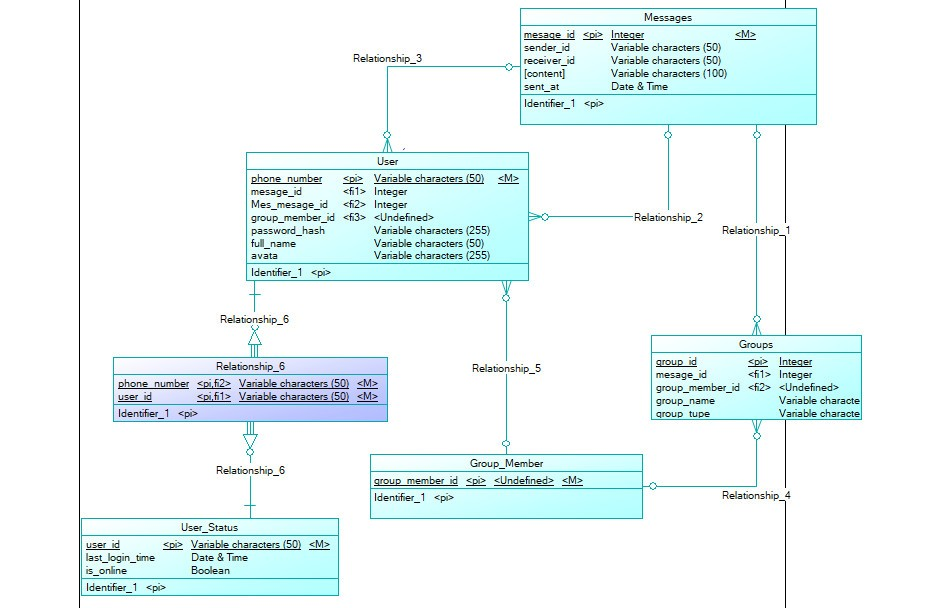
* Mô hình dữ liệu CDM:



Mô hình dữ liệu vật lý : PDM



Mô hình dữ liệu logic :LDM



CHƯƠNG III. MỘT SỐ GIAO DIỆN ỨNG DỤNG CHAT

3.1.Giao diện …

3.2. Giao diện

3.3. Giao diện

CHƯƠNG IV. KẾT LUẬN

4.1. Kết quả đạt được

* Ứng dụng được xây dựng dựa trên mô hình kết hợp giữa client-server cho việc quản lí các user và P2P cho việc trò chuyện giữa hai user với nhau.
* Ứng dụng có các tính năng đơn giản như: chat giữa hai user, một lúc đồng thời chat với nhiều user, gửi File trong quá trình chat.

4.2. Hướng phát triển

* Mã nguồn còn chưa tối ưu cho ứng dụng.
* Ứng dụng còn có thể thêm các tính năng như: chat nhóm, gọi video...

TÀI LIỆU THAM KHẢO