|  |  |
| --- | --- |
| **HỌC VIỆN KỸ THUẬT QUÂN SỰ**  **KHOA: CÔNG NGHỆ THÔNG TIN**  **BỘ MÔN: CÔNG NGHỆ PHẦN MỀM** | **CỘNG HÒA XÃ HỘI CỦ NGHĨA VIỆT NAM**  **Độc lập – Tự do – Hạnh phúc** |

**NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP**

Họ và tên: Hứa Hoàng Linh

# **CHƯƠNG I. KHẢO SÁT ĐỀ TÀI**

1. **Tổng quan về đề tài**

Giao tiếp là một trong những hoạt động thiết yếu trong cuộc sống của con người. Sự giao tiếp đã phát triền từ hàng thế kỷ trước, khi con người vẫn còn dùng thư và người đưa thư để gửi đi những thông tin quan trọng. Ngày nay việc giao tiếp đã đơn giản hơn khi tin nhắn điện tử ra đời.

Xã hội ngày càng phát triền thì nhu cầu giao tiếp giữa còn người và con người cần thiết hơn bao giờ hết. Giao tiếp là một trong những hoạt động thiết yếu nhất. Từ chuyện công việc tới chuyện cá nhân hay những cuộc trò chuyện tán gẫu bình thường. Tuy nhiên, sử dụng tin nhắn SMS gặp phải một số hạn chế như tốn chi phí cho từng tin nhắn. Việc phủ sóng mạng 4G, sự phổ biến của wifi và cả sự phát triển của công nghệ làm cho những ứng dụng nhắn tin trực tuyến trở nên hữu ích hơn bao giờ hết.

Với ứng dụng nhắn tin trực tuyến, người dùng không phải tốn chi phí cho từng tin nhắn. Chưa kể những tính năng mới, tiện ích được phát triển để phù hợp với nhu cầu của người dùng.

1. **Mục đích của đề tài**

Hệ thống được xây dựng với mục đích giúp người dùng có thể giao tiếp được với nhau thông qua Internet. Các phương thức giao tiếp bao gồm: tin nhắn văn bản, tin nhắn tệp, voice/video call. Không những vậy, hệ thống cũng được thiết kế với giao diện đơn giản mà vẫn đầy đủ tính năng, thân thiện ngay cả với người sử dụng lần đầu tiên.

Hệ thống cũng được xây dựng với công nghệ và những kiến trúc tối ưu nhất để có thể vận hành được với một số lượng người dùng khổng lồ và dễ dàng mở rộng cũng như bảo trì.

Ngoài ra, cũng là kết quả của quá trình tự học và tự nghiên cứu. Tạo nền tảng tốt hơn cho việc tự nghiên cứu cũng như tìm hiều các công nghệ khác sau này.

1. **Các ứng dụng trong thực tế**
2. **Skype**
   1. **Giới thiệu**

Skype là dịch vụ gọi điện miễn phí trên nền IP (Voice Over IP) – được phát hành lần đầu tiên vào năm 2003 bởi sự hợp tác của các thành viên từ nhiều quốc gia (Đan Mạch, Thụy Điển…). Skype đã có những bước phát triển mạnh mẽ, sau 7 năm phát hành, đến năm 2010, đã có gần 7 trăm triệu người sử dụng. Năm 2011, Microsoft mua lại Skype với giá 8.5 tỉ USD và hiện Skype vẫn đang thuộc sở hữu của Microsoft.

Hiện nay, Skype đang dần được tích hợp với các dịch vụ khác của Microsoft (bạn có thể đăng nhập Skype bằng tài khoàn Outlook). Chức năng cơ bản của Skype là chat (instant messaging), free call, chia sẻ màn hình… Bạn cũng có thể sử dụng Skype trên các smartphone bằng cách tải ứng dụng Skype từ kho ứng dụng (ví dụ, Goolge Play nếu bạn dùng smartphone Android)

* 1. **Ưu điểm/Nhược điểm**

Ưu điểm:

* Đơn giản, dễ sử dụng
* Hỗ trợ nhiều nền tảng. Hiện nay có rất nhiều ứng dụng gọi điện, nhắn tin miễn phí như Viber, Facebook, Yahoo Messenger,… nhưng dưới góc độ người dùng trong các doanh nghiệp thì Skype vẫn được sử dụng phổ biến hơn cả. Skype có lúc là công cụ chính thống để trao đổi công việc trong các doanh nghiệp, trong khi các ứng dụng gọi điện nhắn tin miễn phí trên mobile chỉ mang tính chất giải trí, cá nhân.
* Nhiều tính năng hay, thiết thực: Quản lý nhóm rất tốt, quản lý lịch sử chat (đặc biệt là tính năng cho phép sửa/xóa nội dung đã gửi),gửi nhận file rất tốt (nhanh – do tính năng tự động nhận diện mạng nội bộ, nếu bạn gửi file qua Skype từ các máy trong cùng mạng LAN thì tốc độ không kém gì khi bạn copy file giữa các máy bằng cách thông thường trong Windows Explorer)

Nhược điểm:

* không hỗ trợ tốt việc gửi tin nhắn offline. Ví dụ, A gửi cho B một tin nhắn trong khi B đang offline, rồi sau đó A cũng offline => thì khi B online trở lại sẽ không nhận được ngay tin nhắn của A mà phải chờ đến khi A và B cùng online.
  1. **Một số tính năng nổi bật**
* Kết nối mọi người
* Gửi tin nhắn
* Voice/Video call
* Gửi tệp
* Trò chuyện nhóm
* Livestream trong nhóm

1. **Facebook Messenger**
   1. **Giới thiệu**

Facebook Messenger là một dịch vụ và ứng dụng phần mềm tin nhắn tức thời chia sẻ giao tiếp bằng ký tự và giọng nói. Được tích hợp trên ứng dụng Chat (trò chuyện) của [Facebook](https://vi.wikipedia.org/wiki/Facebook) và được xây dựng trên giao thức [MQTT](https://vi.wikipedia.org/w/index.php?title=MQTT&action=edit&redlink=1). Messenger cho phép người dùng Facebook trò chuyện với bạn bè trên cả di động và trang web chính.

Theo báo các của Facebook vào tháng 3 năm 2015 Facebook Messenger đạt 600 triệu người sử dụng. [David A. Marcus](https://vi.wikipedia.org/w/index.php?title=David_A._Marcus&action=edit&redlink=1) đứng đầu Facebook Messenger và tham gia Facebook dựa trên lời mời của [Mark Zuckerberg](https://vi.wikipedia.org/wiki/Mark_Zuckerberg), CEO của Facebook. Đến năm 2016 Facebook Messenger đã lên tới 1 tỉ người

* 1. **Ưu điểm/Nhược điểm**

Ưu điểm

* Trao đổi thông tin với người thân dễ dàng
* Chia sẻ tệp, thông báo địa điểm

Nhược điểm

* Thường xuyên nhận được tin nhắn rác
  1. **Một số tính năng nổi bật**
* Trò truyện với bạn bè hoặc người lạ
* Gửi tệp
* Voice/Video call
* Trò chuyện nhóm
* Trò chuyện bí mật
* Chơi game

1. **Kết luận chương**

Trong chương 1 đã nêu ra được các vấn đề chính sau:

* Đưa ra cái nhìn tổng quan và mục đích của đề tài
* Khảo sát các ứng dụng nổi tiếng trên thế giới tại thời điểm hiện tại

1. **Công nghệ và môi trường phát triển**
2. **Java**

Java là một ngôn ngữ lập trình hướng đối tượng (OOP) và dựa trên các lớp (class). Khác với phần lớn ngôn ngữ lập trình thông thường, thay vì biên dịch mã nguồn thành mã máy hoặc thông dịch mã nguồn khi chạy, Java được thiết kế để biên dịch mã nguồn thành bytecode, bytecode sau đó sẽ được môi trường thực thi (runtime environment) chạy.

Trước đây, Java chạy chậm hơn những ngôn ngữ dịch thẳng ra mã máy như C và C++, nhưng sau này nhờ công nghệ "biên dịch tại chỗ" - Just in time compilation, khoảng cách này đã được thu hẹp, và trong một số trường hợp đặc biệt Java có thể chạy nhanh hơn. Java chạy nhanh hơn những ngôn ngữ thông dịch như Python, Perl, PHP gấp nhiều lần. Java chạy tương đương so với C#, một ngôn ngữ khá tương đồng về mặt cú pháp và quá trình dịch/chạy

Cú pháp Java được vay mượn nhiều từ C & C++ nhưng có cú pháp hướng đối tượng đơn giản hơn và ít tính năng xử lý cấp thấp hơn. Do đó việc viết một chương trình bằng Java dễ hơn, đơn giản hơn, đỡ tốn công sửa lỗi hơn.

Trong Java, hiện tượng rò rỉ bộ nhớ hầu như không xảy ra do bộ nhớ được quản lý bởi Java Virtual Machine (JVM) bằng cách tự động "dọn dẹp rác". Người lập trình không phải quan tâm đến việc cấp phát và xóa bộ nhớ như C, C++. Tuy nhiên khi sử dụng những tài nguyên mạng, file IO, database (nằm ngoài kiểm soát của JVM) mà người lập trình không đóng (close) các streams thì rò rỉ dữ liệu vẫn có thể xảy ra.

1. **Embedded Jetty**

Eclipse Jetty vừa là Java HTTP Server và Java Servet Container được phát triển bởi Eclipse Foudation. Jetty được phát triển cho mọi người sử dụng miễn phí và là phần mềm mã nguồn mở. Jetty được biết đến là một trong những công cụ mạnh mẽ trong việc phát triển phần mềm theo kiến trúc Mirco Service

Embedded Jetty là một thư viện giúp cho lập trình viên có thể nhúng Jetty Web Server vào chương trình của mình thay vì triển khai ứng dụng trên 1 Web Server.

1. **MongoDB**

MongoDB là một trong những cơ sở dữ liệu mã nguồn mở NoSQL phổ biến nhất được biết bằng C++. Tính đến tháng 2/2015, MongoDB được xếp thứ 4 trong số các hệ thống cơ sở dữ liệu phổ biến nhất. Nó được pahts triển bởi công ty 10gen sau này được biết đến với tên MongoDB Inc..

MongoDB hỗ trợ đa nền tảng, nó có thể chạy trên Windows, Linux và Mac...Nó hỗ trợ hầu hết các ngôn ngữ lập trình phổ biến như C#, Java, PHP, Javascript...và các môi trường phát triển khác nhau.

MongoDB là cơ sở dữ liệu hướng tài liệu, nó lưu trữ dữ liệu trong các bản ghi dạng BSON với cấu trúc động rất linh hoạt. Nghĩa là bạn có thể lưu các bản ghi mà không cần lo lắng về cấu trúc dữ liệu như là số trường, kiểu của trường lưu trữ. Cấu trúc bản ghi của MongoDB tương tự như các đối tượng JSON.

1. **ReactJS**

React là một thư viện UI phát triển tại Facebook để hỗ trợ việc xây dựng những thành phần (components) UI có tính tương tác cao, có trạng thái và có thể sử dụng lại được.

ReactJS đang nổi lên trong những năm gần đây với xu hướng Single Page Application. Trong khi những framework khác cố gắng hướng đến một mô hình MVC hoàn thiện thì React nổi bật với sự đơn giản và dễ dàng phối hợp với những thư viện Javascript khác

React được sử dụng tại Facebook trong production, và trang web nổi tiếng www.instagram.com được viết hoàn toàn trên React.

1. **WebRTC**

WebRTC (Web Real-Time Communication) là một dự án nguồn mở, miễn phí, cung cấp các trình duyệt web và ứng dụng di động với giao tiếp thời gian thực (RTC) thông qua các 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 plugin hoặc tải xuống ứng dụng gốc. WebRTC cược hỗ trợ bởi Google, Microsoft, Mozilla và Opera, WebRTC đang được chuẩn hóa thông qua World Wide Web Consortium (W3C) và Lực lượng đặc nhiệm kỹ thuật Internet (IETF).

# **CHƯƠNG II. CƠ SỞ LÝ THUYẾT**

1. **Ngôn ngữ Java**
2. **Java là gì?**

Java là một ngôn ngữ lập lập trình, được phát triển bởi Sun Microsystem vào năm 1995, là ngôn ngữ kế thừa trực tiếp từ C/C++ và là một ngôn ngữ lập trình hướng đối tượng.

Vì sao ngôn ngữ này lại được đặt tên là Java? Java là tên một hòn đảo ở Indonesia - hòn đảo nổi tiếng với loại coffee Peet và cũng là loại nước uống phổ biến của các kỹ sư Sun. Ban đầu Ngôn ngữ này được đặt tên là "Oak" (có nghĩa là "Cây sồi" - 1991), nhưng các luật sư của Sun xác định rằng tên đó đã được đăng ký nhãn hiệu nên các nhà phát triển đã phải thay thế bằng một tên mới - và cũng vì lý do trên mà cái tên Java đã ra đời và trở thành tên gọi chính thức của Ngôn ngữ này - Ngôn ngữ Lập trình Java.

1. **Ngôn ngữ Java**

Java là một ngôn ngữ lập trình hướng đối tượng (OOP) và dựa trên các lớp (class). Khác với phần lớn ngôn ngữ lập trình thông thường, thay vì biên dịch mã nguồn thành mã máy hoặc thông dịch mã nguồn khi chạy, Java được thiết kế để biên dịch mã nguồn thành bytecode, bytecode sau đó sẽ được môi trường thực thi (runtime environment) chạy.

Trước đây, Java chạy chậm hơn những ngôn ngữ dịch thẳng ra mã máy như C và C++, nhưng sau này nhờ công nghệ "biên dịch tại chỗ" - Just in time compilation, khoảng cách này đã được thu hẹp, và trong một số trường hợp đặc biệt Java có thể chạy nhanh hơn. Java chạy nhanh hơn những ngôn ngữ thông dịch như Python, Perl, PHP gấp nhiều lần. Java chạy tương đương so với C#, một ngôn ngữ khá tương đồng về mặt cú pháp và quá trình dịch/chạy

Cú pháp Java được vay mượn nhiều từ C & C++ nhưng có cú pháp hướng đối tượng đơn giản hơn và ít tính năng xử lý cấp thấp hơn. Do đó việc viết một chương trình bằng Java dễ hơn, đơn giản hơn, đỡ tốn công sửa lỗi hơn.

Trong Java, hiện tượng rò rỉ bộ nhớ hầu như không xảy ra do bộ nhớ được quản lý bởi Java Virtual Machine (JVM) bằng cách tự động "dọn dẹp rác". Người lập trình không phải quan tâm đến việc cấp phát và xóa bộ nhớ như C, C++. Tuy nhiên khi sử dụng những tài nguyên mạng, file IO, database (nằm ngoài kiểm soát của JVM) mà người lập trình không đóng (close) các streams thì rò rỉ dữ liệu vẫn có thể xảy ra.

1. **Đặc điểm cơ bản của Java**

Tiêu chí hàng đầu của Ngôn ngữ Lập trình Java là "Write Once, Run Anywhere" (Viết một lần, chạy mọi nơi), nghĩa là Java cho phép chúng ta viết code một lần và thực thi được trên các hệ điều hành khác nhau. Ví dụ, chúng ta viết code trên Hệ điều hành Windows và nó có thể thực thi được trên các Hệ điều hành Linux và Mac OS...

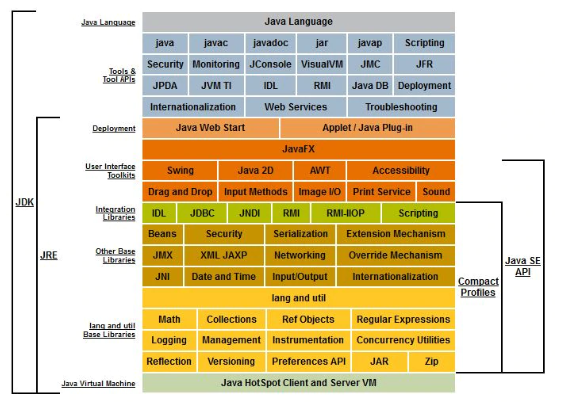
Với đặc điểm nổi bật đó, Java có những đặc điểm cơ bản như sau:

* Đơn giản và quen thuộc: Vì Java kế thừa trực tiếp từ C/C++ nên nó có những đặc điểm của ngôn ngữ này, Java đơn giản vì mặc dù dựa trên cơ sở C++ nhưng Sun đã cẩn thận lược bỏ các tính năng khó nhất của của C++ để làm cho ngôn ngữ này dễ sử dụng hơn.
* Hướng đối tượng và quen thuộc.
* Mạnh mẽ (thể hiện ở cơ chế tự động thu gom rác - Garbage Collection) và an toàn.
* Kiến trúc trung lập, độc lập nền tảng và có tính khả chuyển (Portability).
* Hiệu suất cao.
* Máy ảo (biên dịch và thông dịch).
* Phân tán.
* Đa nhiệm: Ngôn ngữ Java cho phép xâ dựng trình ứng dụng, trong đó nhiều quá trình có thể xảy ra đồng thời. Tính đa nhiệm cho phép các nhà lập trình có thể biên soạn phần mềm đáp ứng tốt hơn, tương tác tốt hơn và thực hiện theo thời gian thực.

1. **Các nền tảng cơ bản của Java**

Java gồm có 3 thành phần chính:

* Java Virtual Machine (Java VM): Máy ảo Java.
* Java Application Programming Interface (Java API).
* Java Development Kit (JDK) gồm trình biên dịch, thông dịch, trợ giúp, soạn tài liệu... và các thư viện chuẩn.



1. **Tại sao chọn Java để làm ngôn ngữ để phát triển phía Server?**
2. **Embedded Jetty**
3. **Embedded Jetty là gì?**

Eclipse Jetty vừa là Java HTTP Server và Java Servet Container được phát triển bởi Eclipse Foudation. Jetty được phát triển cho mọi người sử dụng miễn phí và là phần mềm mã nguồn mở. Jetty được biết đến là một trong những công cụ mạnh mẽ trong việc phát triển phần mềm theo kiến trúc Mirco Service

Embedding Jetty có một slogan là “Don’t deploy your application in Jetty, deploy Jetty in your application!” nghĩa là thay vì đóng gói ứng dụng và deploy nó trên Web Server thì chúng ta có thể nhúng trực tiếp Web Server vào ứng dụng của mình

Embedded Jetty là một thư viện giúp cho lập trình viên có thể nhúng Jetty Web Server vào chương trình của mình thay vì triển khai ứng dụng trên 1 Web Server.

1. **So sánh Jetty với Tomcat**

Ở đây chúng ta sẽ so sánh Jetty với Tomcat. Tại sao lại là Tomcat mà không phải một Web Server nào khác? Đơn giản là bởi vì Tomcat là Web Server có cộng đồng lập trình viên sử dụng đông đảo nhất và cũng là một trong những công nghệ nổi tiếng nhất

|  |  |
| --- | --- |
| **Tomcat** | **Jetty** |
| Tomcat thuộc sở hữu của Apache | Jetty thuộc sở hữu của Eclipse |
| Tomcat được sử dụng rộng rãi và có một cộng đồng lớn có thể hỗ trợ mọi vấn đề liên quan tới Tomcat | Jetty ít được biết tới hơn, tuy nhiên cũng được sử dụng khá rộng rãi bởi khả năng nhúng của nó |
| Tomcat là Java Servlet cho các server khác, bao gồm JBoss | Jetty là thư viện viết bằng Java và có thể sử dụng trực tiếp trong mã nguồn |
| Tomcat khá là nặng và khó tìm hiểu | Jetty dễ dàng nhúng và nhẹ hơn rất nhiều |

1. **Ưu điểm của Jetty**

Tuy có nhiều điểm không bằng Tomcat, nhưng Jetty cũng có những ưu điểm vượt trội hơn so với các Web Server khác:

* Sử dụng ít bộ nhớ hơn và nhẹ hơn do đó cung cấp tốc độ và khả năng mở rộng
* Có thể được nhúng dễ dàng trong ứng dụng Java Web cũng như phục vụ như máy chủ không đồng bộ
* Mã nguồn mở và có sự hỗ trợ tốt từ cộng đồng
* Có vùng phủ bộ nhớ (memory footprint) nhỏ nên có thể khởi chạy và khởi động lại nhanh chóng
* Tùy biến cao
* Quy mô nhỏ cùng với hiệu quả cao, chi phí bảo trì và tổng chi phí sở hữu thấp

1. **MongoDB**
2. **Giới thiệu về MongoDB**

MongoDB là một trong những cơ sở dữ liệu mã nguồn mở NoSQL phổ biến nhất được biết bằng C++. Tính đến tháng 2/2015, MongoDB được xếp thứ 4 trong số các hệ thống cơ sở dữ liệu phổ biến nhất. Nó được phát triển bởi công ty 10gen sau này được biết đến với tên MongoDB Inc..

MongoDB hỗ trợ đa nền tảng, nó có thể chạy trên Windows, Linux và Mac...Nó hỗ trợ hầu hết các ngôn ngữ lập trình phổ biến như C#, Java, PHP, Javascript...và các môi trường phát triển khác nhau.

MongoDB là cơ sở dữ liệu hướng tài liệu, nó lưu trữ dữ liệu trong các bản ghi dạng BSON với cấu trúc động rất linh hoạt. Nghĩa là bạn có thể lưu các bản ghi mà không cần lo lắng về cấu trúc dữ liệu như là số trường, kiểu của trường lưu trữ. Cấu trúc bản ghi của MongoDB tương tự như các đối tượng JSON.

1. **Ưu điểm và nhược điểm của MongoDB**
   1. **Ưu điểm**

* Mã nguồn mở: MongoDB là phần mềm mã nguồn mở miễn phí và có cộng đồng phát triển rất lớn
* Hiệu năng cao: Tốc độ truy vấn của MongoDB nhanh hơn hẳn so với Hệ quản trị CSDL quan hệ (RDBMS). Thử nghiệm cho thấy tốc độ thêm bản ghi mới nhanh hơn SQL khoảng 100 lần
* Dữ liệu linh hoạt: MongoDB là dạng cơ sở dữ liệu hướng tài liệu (document database), lưu dữ liệu dướng dạng BSON, không bó buộc về số lượng trường và kiểu dữ liệu.
* Rich Query Language: MongoDB có sẵn các phương thức CRUD khiến cho việc sử dụng trở nên dễ dàng hơn
* Tính sẵn có: Giống như các RDBMS khác, MongoDB cũng hỗ trợ Replica Set nhằm việc sao lưu và khôi phục dữ liệu
* Khả năng mở rộng: Trong MongoDB có một khái niệm cluster là cụm các node chứa dữ liệu giao tiếp với nhau, khi muốn mở rộng hệ thống ta chỉ cần thêm một node với vào cluster
  1. **Nhương điểm**

Tuy có rất nhiều ưu điểm đáng chú ý, nhưng MongoDB vẫn có một số những nhược điểm sau:

* MongoDB không có các tính chất ràng buộc như trong RDBMS –> dễ bị làm sai dữ liệu
* Không hỗ trợ join giống như RDBMS nên khi viết function join trong code ta phải làm bằng tay khiến cho tốc độ truy vấn bị giảm.
* Sử dụng nhiều bộ nhớ: do dữ liệu lưu dưới dạng key-value, các collection chỉ khác về value do đó key sẽ bị lặp lại. Không hỗ trợ join nên sẽ bị dữ thừa dữ liệu (trong RDBMS thì ta chỉ cần lưu 1 bản ghi rồi các bản ghi khác tham chiếu tới còn trong MongoDB thì không)

1. **So sánh MongoDB với MySQL**

Ở đây lựa chọn MySQL để so sánh trực tiếp vì MySQL là một trong những Hệ quản trị CSDL quan hệ rất mạnh và có cộng đồng cũng rất phát triển

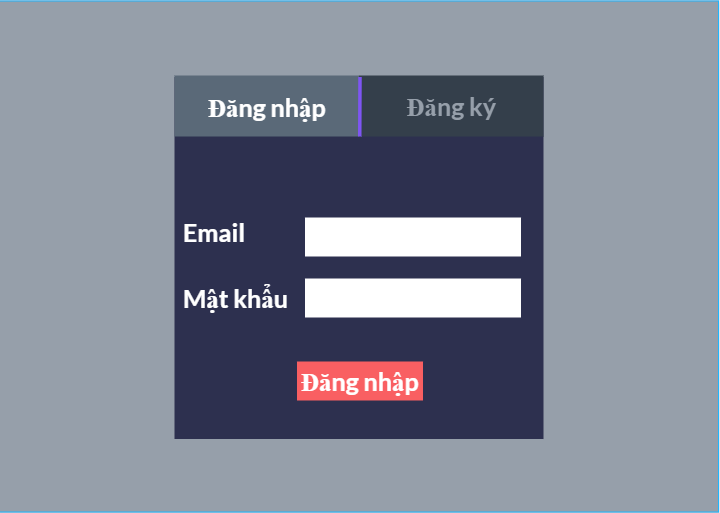
|  |  |  |
| --- | --- | --- |
|  | **MySQL** | **MongoDB** |
| Viết bởi ngôn ngữ | C, C++ | C, C++, JS |
| Kiểu | RDBMS | Document Oriented |
| Đặc điểm chính | Table, Row, Column | Collection, Document, Field |
| Giấy phép | GPL v2 / Giấy phép thương mại có sẵn OD | OD GNU AGPL v3.0 / Giấy phép thương mại có sẵn OD |
| Lược đồ | Strict | Dynamic |
| Mở rộng | Theo chiều dọc | Theo chiều ngang |
| Các tính năng chính | - Tìm kiếm và đán chỉ mục  - Hỗ trợ replication  - Trigger  - Truy vấn lồng  - Truy vấn bộ nhớ đệm  - Hỗ trợ SSL  - Hỗ trợ Unicode  - Công cụ lưu trữ khác nhau với các đặc tính hiệu suất khác nhau | - Auto-sharding  - Hỗ trợ replication  - Hỗ trợ dữ liệu nhúng  - Chỉ mục phụ toàn diện  - Ngôn ngữ truy vấn phong phú  - Hỗ trợ công cụ lưu trữ khác nhau |
| Sử dụng tốt nhất cho | - Cấu trúc dữ liệu phù hợp với bảng và hàng  - Cập nhật thường xuyên và sửa đổi khối lượng lớn bản ghi  - Bộ dữ liệu tương đối nhỏ | - Chịu tải read/write tốt  - Lược đồ không ổn định  - phù hợp với hệ thống dữ liệu lớn  - Tính sẵn sàng cao  - Không có CSDL của quản trị viên |

1. **Tại sao chọn MongoDB?**

* Vì đây là hệ thống trò chuyện trực tuyến nên yếu tố realtime là quan trọng nhất. Do đó với đặc điểm truy vấn cực nhanh của MongoDB là một trong những yếu tố quan trọng nhất
* Hệ thống phải lưu lại rất nhiều dữ liệu người dùng, lịch sử trò chuyện… nên với số lượng người dùng lớn đây sẽ trở thành hệ thống bigdata. MongoDB làm việc rất hiệu quả với bigdata hiệu quả hơn RDBMS rất nhiều
* Tần suất read/write của hệ thống là rất lớn

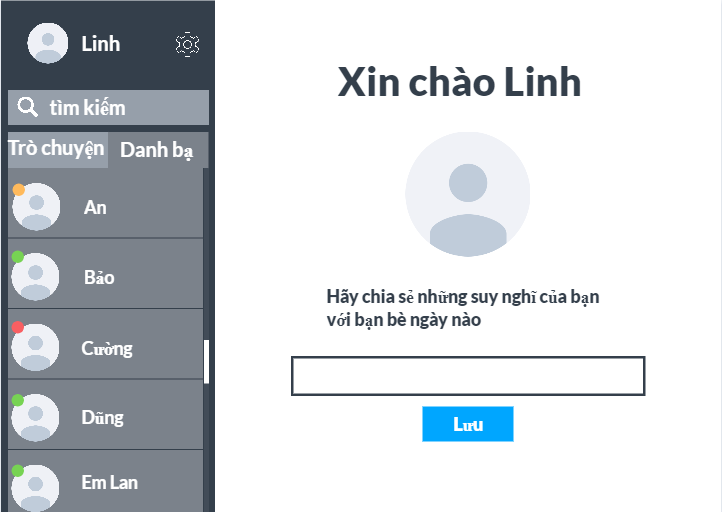
# **CHƯƠNG III. ĐẶC TẢ YÊU CẦU PHẦN MỀM**

1. **Đặc tả yêu cầu giao diện**

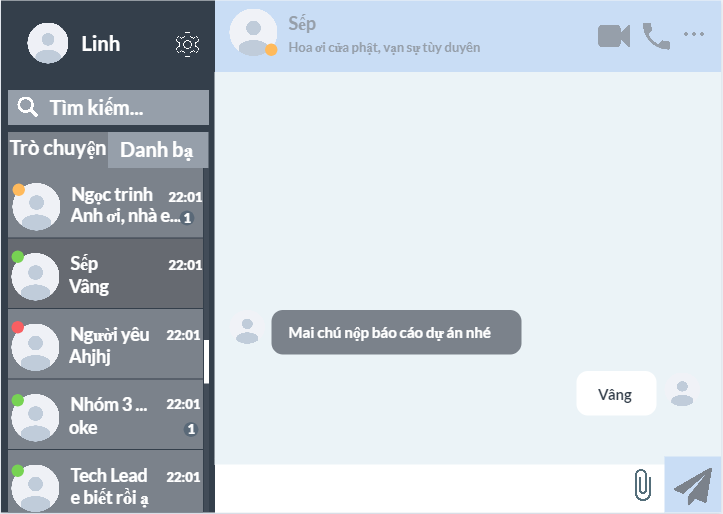




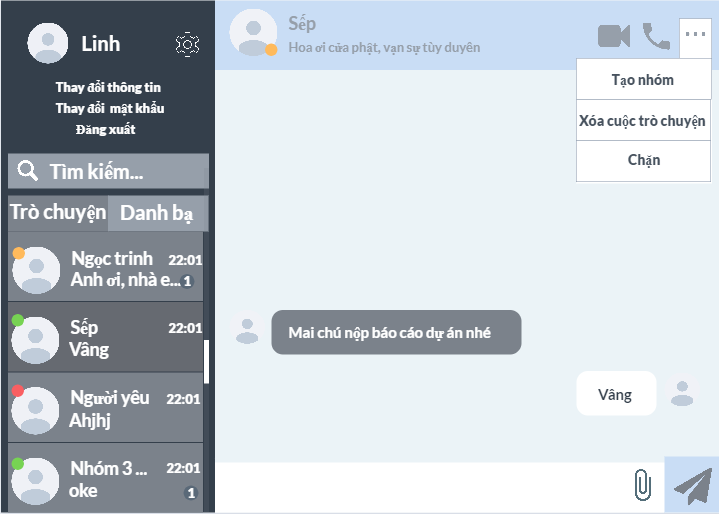
*Màn hình đăng ký/đăng nhập*



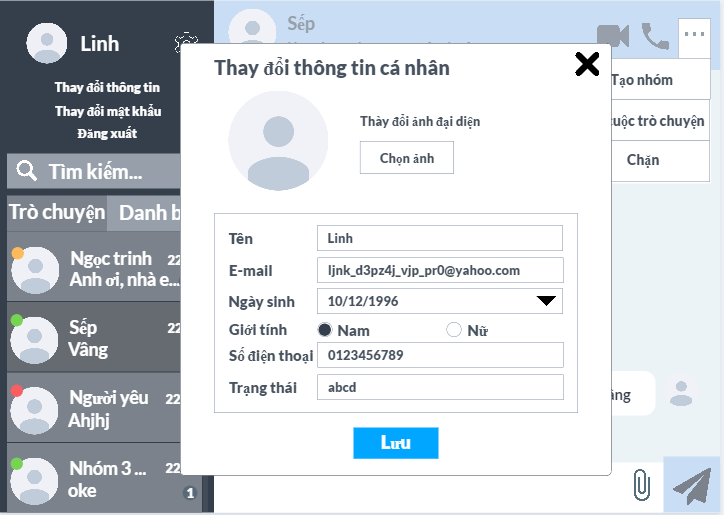
*Màn hình chính*



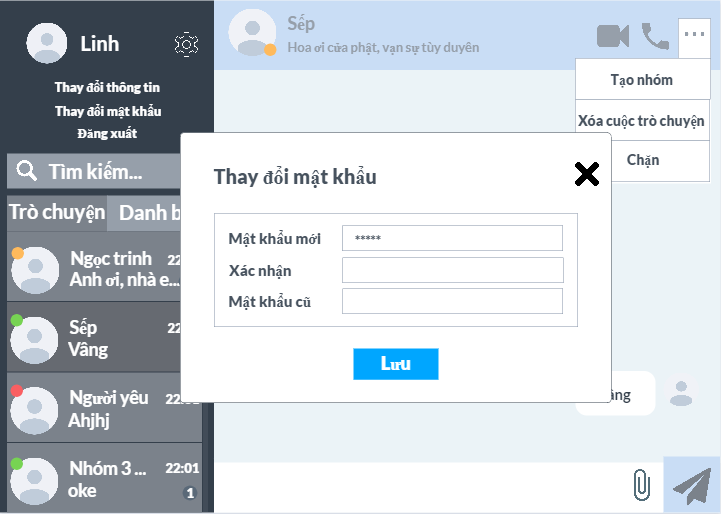
*Tin nhắn với bạn bè*



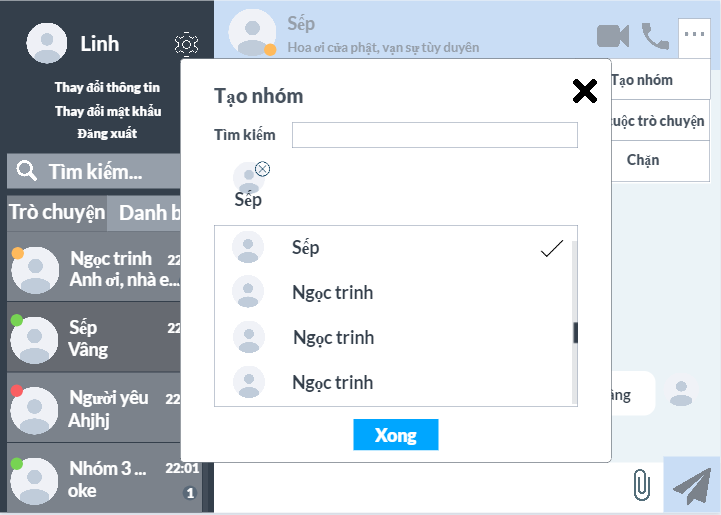
*Menu thông tin tài khoản và menu trong cuộc trò chuyện*



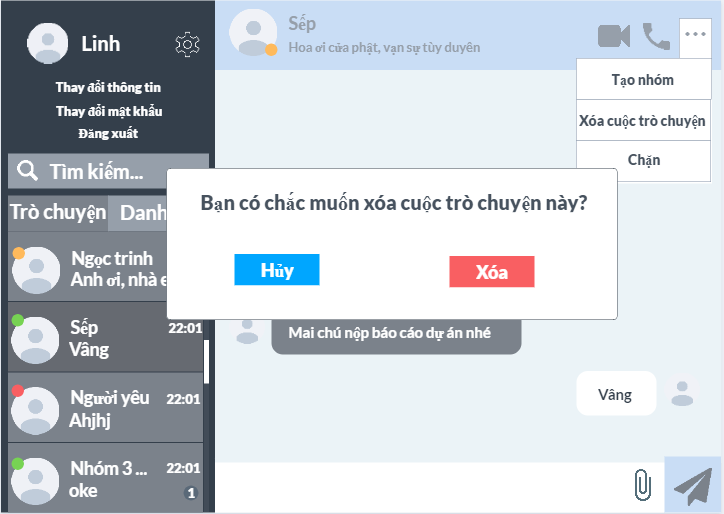
*Thay đổi thông tin cá nhân*



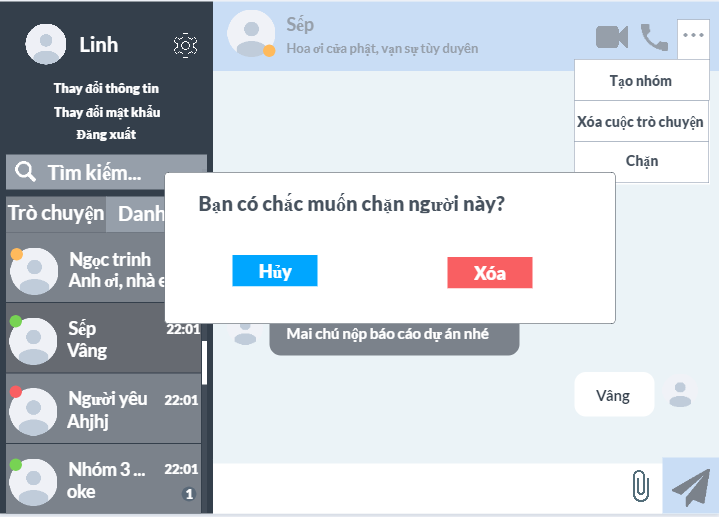
*Thay đổi mật khẩu*



*Tạo cuộc trò chuyện nhóm*



*Xóa cuộc trò chuyện*



*Chặc liên lạc*

1. **Đặc tả yêu cầu phi chức năng**

Các yêu cầu chính chính:

1. **Đặc tả yêu cầu chức năng**
2. **Các chức năng chính**

* Đăng ký
* Đăng nhập
* Đăng xuất
* Lấy lại tài khoản
* Hiển thị thông tin người dùng
* Cập nhật thông tin
* Thay đổi mật khẩu
* Tìm kiếm bạn bè
* Thêm liên lạc mới
* Xóa liên lạc
* Hiển thị danh sách liên lạc
* Hiển thị danh sách tin nhắn gần nhất
* Hiển thị lịch sử tin nhắn
* Gửi tin nhắn văn bản
* Gửi tệp đính kèm
* Tạo cuộc trò chuyện nhóm
* Thêm thành viên vào cuộc trò chuyện nhóm
* Xóa cuộc trò chuyện
* Voice call
* Video call

1. **Mô tả chi tiết yêu cầu chức năng**

* **ID: FR01**

TITLE: Đăng ký

DESC: Người dùng đăng ký tài khoản mới

RAT: Chức năng cơ bản bắt buộc phải có

* **ID: FR02**

TITLE: Đăng nhập

DESC: Người dùng sử dụng tài khoản đã đăng ký trước đó để đăng nhập vào hệ thống

RAT: Chức năng cơ bản bắt buộc phải có

DEP: FR01

* **ID: FR03**

TITLE: Đăng xuất

DESC: Người dùng đăng xuất ra khỏi tài khoản hiện tại để đăng ký hoặc đăng nhập với tài khoản khác

RAT: Chức năng cơ bản bắt buộc phải có

DEP: FR01, FR02

* **ID: FR04**

TITLE: Lấy lại tài khoản

DESC: Người dùng sử dụng chức năng này để có thể lấy lại tài khoản trong trường hợp quên tài khoản hoặc tài khoản bị đánh cắp. Người dùng sẽ phải cung cấp một số thông tin để có thể lấy lại tài khoản của mình

RAT: Sử dụng để lấy lại tài khoản của người dùng

* **ID: FR05**

TITLE: Hiển thị thông tin người dùng

DESC: Người dùng sử dụng tính năng này để hiển thị thông tin của bản thân hoặc của bạn bè. Các thông tin cơ bản sẽ được hiển thị bao gồm: tên, giới tính, ngày sinh, số điện thoại, ảnh đại diện

RAT: Sử dụng để hiển thị thông tin của chính mình hoặc người khác

DEP: FR01, FR02

* **ID: FR06**

TITLE: Cập nhật thông tin

DESC: Người dùng có nhu cầu thay đổi các thông tin cá nhân sẽ sử dụng tính năng này để thay đổi. Người dùng sẽ được phép thay đổi các thông tin cá nhân cơ bản như: tên, giới tính, ngày sinh, số điện thoại, ảnh đại diện

RAT: Sử dụng để thay đổi thông tin người dùng

DEP: FR01, FR02

* **ID: FR07**

TITLE: Thay đổi mật khẩu

DESC: Người dùng có thể thay đổi mật khẩu của mình trong trường hợp nghi ngờ người khác biết mật khẩu

RAT: Sử dụng để thay đổi mật khẩu

DEP: FR01, FR02

* **ID: FR08**

TITLE: Tìm kiếm bạn bè

DESC: Người dùng sẽ sử dụng tính năng này để tìm kiếm bạn bè mới và có thể xem một số thông tin cơ bản của họ

RAT: Sử dụng để người dùng tìm kiếm bạn bè

DEP: FR01, FR02

* **ID: FR09**

TITLE: Thêm liên lạc mới

DESC: Người dùng sẽ thêm một người dùng khác vào danh sách liên lạc của mình

RAT: Sử dụng để người dùng thêm bạn bè

DEP: FR01, FR02, FR08

* **ID: FR10**

TITLE: Xóa liên lạc

DESC: Người dùng sẽ xóa người dùng khác khỏi danh sách liên lạc của mình

RAT: Sử dụng để người dùng xóa liên lạc

DEP: FR01, FR02, FR11

* **ID: FR11**

TITLE: Hiển thị danh sách liên lạc

DESC: Hiển thị danh sách liên lạc của người dùng

RAT: Người dùng phải biết danh sách liên lạc của mình gồm những ai

DEP: FR01, FR02, FR10, FR09

* **ID: FR12**

TITLE: Hiển thị danh sách tin nhắn gần nhất

DESC: Người dùng sẽ có thể xem danh sách tin nhắn gần nhất của mình, các tin nhắn sẽ được sắp xếp theo thứ tự thời gian

RAT: Người dùng có thể xem danh sách tin nhắn gần nhất

DEP: FR01, FR02

* **ID: FR13**

TITLE: Hiển thị lịch sử tin nhắn

DESC: Khi người dùng bấm vào chi tiết trong danh sách tin nhắn gần nhất, người dùng sẽ xem được lịch sử cuộc trò chuyện đó

RAT: Sử dụng để xem tin nhắn của cuộc trò chuyện

DEP: FR01, FR02, FR12

* **ID: FR14**

TITLE: Gửi tin nhắn văn bản

DESC: Tính năng quan trọng nhất của phần mềm giúp người dùng có thể trò chuyện với những người khác

RAT: Sử dụng để gửi tin nhắn cho những người khác

DEP: FR01, FR02

* **ID: FR15**

TITLE: Gửi tin tệp đính kèm

DESC: Tính năng quan trọng nhất của phần mềm giúp người dùng có thể gửi các tệp tới những người khác

RAT: Sử dụng để gửi tệp cho những người khác

DEP: FR01, FR02

* **ID: FR16**

TITLE: Tạo cuộc trò chuyện nhóm

DESC: Trong trường hợp người dùng muốn tạo cuộc trò chuyện có nhiều hơn 2 người sẽ sử dụng tính năng này để thêm những người dùng khác vào cuộc trò chuyện

RAT: Sử dụng để tạo cuộc trò chuyện nhóm

DEP: FR01, FR02

* **ID: FR17**

TITLE: Thêm thành viên vào cuộc trò chuyện nhóm

DESC: Trong trường hợp người dùng muốn thêm một người khác vào cuộc trò chuyện nhóm đã tạo trước đó

RAT: Sử dụng để thêm người và cuộc trò chuyện nhóm

DEP: FR01, FR02, FR16

* **ID: FR18**

TITLE: Xóa cuộc trò chuyện

DESC: Người dùng sử dụng tính năng để xóa toàn bộ lịch sử của cuộc trò chuyện

RAT: Sử dụng để xóa lịch sử trò chuyện

DEP: FR01, FR02

* **ID: FR19**

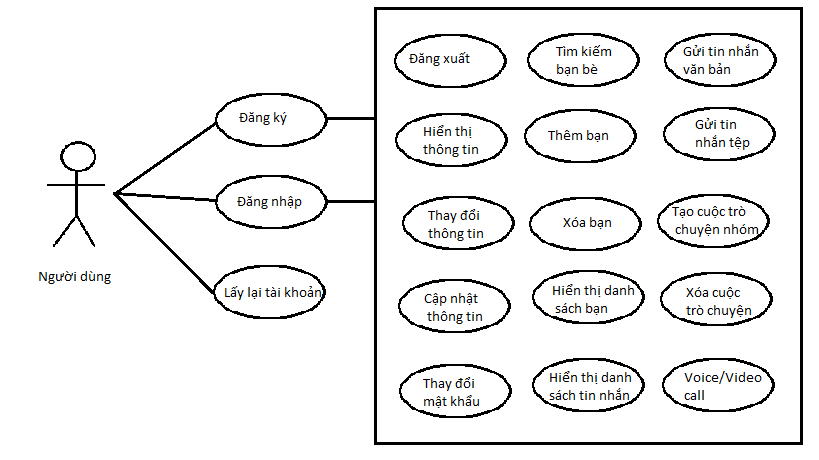
TITLE: Voice/video call

DESC: Người dùng sử dụng tính năng gọi điện bằng âm thanh hoặc hình ảnh cho người khác

RAT: Sử dụng để tạo cuộc gọi với người khác

DEP: FR01, FR02

1. **Đặc tả yêu cầu hệ thống**
2. **Mô hình Use Case tổng thể**

****

1. **Đặc tả chi tiết các Use Case**

|  |  |
| --- | --- |
| **Use Case** | **Đăng ký** |
| **Điều kiện trước** | * Người dùng truy cập vào trang web |
| **Điều kiện sau** | * Người dùng chuyển sang màn hình chính |
| **Luồng sử kiện chính** | 1. Người dùng bấm vào tab “Đăng ký” 2. Người dùng nhập các thông tin: e-mail, tên người dùng, ngày sinh, giới tính, số điện thoại, mật khẩu, mật khẩu xác nhận 3. Người dùng bấm vào nút “Đăng ký” 4. Hệ thống đăng ký thành công và chuyển người dùng tới trang chính |
| **Luồng sự kiện thay thế** | Tại bước 4, ở các trường hợp:   * E-mail đã đăng ký * E-mail không hợp lệ * Mật khẩu không phù hợp * Mật khẩu xác nhận không đúng * Ngày sinh không hợp lệ   Hệ thống sẽ trả thông báo lỗi ở màn hình đăng ký và người dung sẽ phải nhập thông tin phù hợp |

|  |  |
| --- | --- |
| **Use Case** | **Đăng nhập** |
| **Điều kiên trước** | * Người dùng truy cập vào trang web |
| **Điều kiện sau** | * Người dùng chuyển sang màn hình chính |
| **Luồng sử kiện chính** | 1. Người dùng bấm vào tab “Đăng nhập” 2. Người dùng nhập các thông tin: e-mail, mật khẩu 3. Người dùng bấm vào nút “Đăng nhập” 4. Hệ thống kiểm tra thông tin người dùng và chuyển người dùng tới trang chính |
| **Luồng sự kiện thay thế** | Tại bước 4, ở các trường hợp:   * E-mail không hợp lệ * Mật khẩu không đúng   Hệ thống sẽ trả thông báo lỗi ở màn hình đăng nhập và người dùng sẽ phải nhập thông tin phù hợp |

|  |  |
| --- | --- |
| **Use Case** | **Đăng xuất** |
| **Điều kiên trước** | * Người dùng đã đăng nhập vào trang chính |
| **Điều kiện sau** | * Người dùng được chuyển về màn hình đăng ký/đăng nhập |
| **Luồng sử kiện chính** | 1. Người dùng bấm vào biểu tượng bánh răng bên cạnh tên của mình 2. Menu hiển thị các chức năng trong đó có chức năng đăng xuất 3. Người dùng bấm vào nút “Đăng xuất” 4. Hệ thống xóa dữ liệu đăng nhập và chuyển người dùng tới trang đăng ký/đăng nhập |

|  |  |
| --- | --- |
| **Use Case** | **Lấy lại tài khoản** |
| **Điều kiện trước** | * Người dùng truy cập vào trang web * Người dùng đã đăng ký tài khoản trong hệ thống nhưng quên mất mật khẩu |
| **Điều kiện sau** | * Người dùng nhận thay đổi mật khẩu mới thành công và được chuyển tới trang chính |
| **Luồng sử kiện chính** | 1. Người dùng bấm vào chức năng “Quên mật khẩu” 2. Người dùng nhập thông tin e-mail 3. Người dùng nhận được 1 email có chứa mã xác nhận 4. Người dùng nhập mã xác nhận cùng với mật khẩu mới và mật khẩu xác nhận 5. Hệ thống thay đổi mật khẩu mới cho người dùng và chuyển người dùng tới trang chính |
| **Luồng sự kiện thay thế** | Tại bước 2, ở các trường hợp:   * E-mail không hợp lệ * E-mail không tồn tại trong hệ thống   Hệ thống đưa ra thông báo “E-mail không hợp lệ, vui lòng kiểm tra lại”  Tại bước 4, ở các trường hợp:   * Mã xác nhận không đúng * Mật khẩu và mật khẩu xác nhận không trùng khớp   Người dung cũng sẽ được nhận thông báo lỗi tương ứng trên màn hình |

|  |  |
| --- | --- |
| **Use Case** | **Hiển thị thông tin người dùng** |
| **Điều kiện trước** | * Người dùng truy cập vào trang web * Người dùng đã đăng nhập vào hệ thống |
| **Điều kiện sau** | * Hiển thị thông tin chi tiết của người dùng |
| **Luồng sử kiện chính** | 1. Người dùng bấm vào tên người dùng hoặc ảnh đại diện của bất kỳ người dùng nào (bao gồm chính mình) 2. Popup mới hiện lên trên màn hình và hiển thị thông tin về người dùng bao gồm: E-mail, tên người dùng, ngày sinh, giới tính, số điện thoại |

|  |  |
| --- | --- |
| **Use Case** | **Cập nhật thông tin** |
| **Điều kiên trước** | * Người dùng truy cập vào trang web * Người dùng đã đăng nhập vào hệ thống |
| **Điều kiện sau** | * Thông tin mới của người dùng được cập nhật thành công |
| **Luồng sử kiện chính** | 1. Người dùng bấm vào hình bánh răng bên cạnh tên của mình 2. Menu mở rộng và hiển thị chức năng “Thay đổi thông tin cá nhân”, người dùng bấm vào chức năng này 3. Popup mới hiện lên trên màn hình và hiển thị thông tin về người dùng bao gồm: tên người dùng, ngày sinh, giới tính, số điện thoại. Ngoài ra người dùng sẽ phải nhập thêm mật khẩu hiện tại để xác nhận 4. Người dùng thanh đổi thông tin cá nhân của mình 5. Người dùng bấm vào nút cập nhật 6. Hệ thống thông báo người dùng cập nhật thành công và ẩn popup |
| **Luồng sự kiện thay thế** | Tại bước 5, ở các trường hợp:   * Ngày sinh không hợp lệ * Tên người dùng không phù hợp * Mật khẩu hiện tại không đúng   Người dùng cũng sẽ được nhận thông báo lỗi tương ứng trên màn hình |

|  |  |
| --- | --- |
| **Use Case** | **Thay đổi mật khẩu** |
| **Điều kiện trước** | * Người dùng truy cập vào trang web * Người dùng đã đăng nhập vào hệ thống |
| **Điều kiện sau** | * Thông tin mới của người dùng được cập nhật thành công |
| **Luồng sử kiện chính** | 1. Người dùng bấm vào hình bánh răng bên cạnh tên của mình 2. Menu mở rộng và hiển thị chức năng “Thay đổi thông mật khẩu”, người dùng bấm vào chức năng này 3. Popup mới hiện lên trên màn hình và hiển thị 3 ô: mật khẩu mới, mật khẩu xác nhận, mật khẩu cũ. 4. Người dùng điền đủ thông tin và bấn nút “Cập nhật” 5. Hệ thống thông báo người dùng thay đổi mật khẩu thành công và ẩn popup |
| **Luồng sự kiện thay thế** | Tại bước 4, ở các trường hợp:   * Mật khẩu xác nhận không đúng * Mật khẩu không phù hợp * Mật khẩu cũ không chính xác   Người dùng cũng sẽ được nhận thông báo lỗi tương ứng trên màn hình |

|  |  |
| --- | --- |
| **Use Case** | **Tìm kiếm bạn bè** |
| **Điều kiện trước** | * Người dùng truy cập vào trang web * Người dùng đã đăng nhập vào hệ thống |
| **Điều kiện sau** | * Hiển thị danh sách những người có tên hoặc email gần giống với chuỗi mà người dùng tìm kiếm |
| **Luồng sử kiện chính** | 1. Nhập ký tự bất kỳ vào thanh tìm kiếm 2. Hệ thống sẽ tự động hiển thị ra danh sách những người có tên và email gần giống với chuỗi được tìm kiếm |

|  |  |
| --- | --- |
| **Use Case** | **Thêm liên lạc mới** |
| **Điều kiện trước** | * Người dùng truy cập vào trang web * Người dùng đã đăng nhập vào hệ thống * Người dùng sử dụng chức năng tìm kiếm bạn bè và được hệ thống trả kết quả |
| **Điều kiện sau** | * Những người được thêm và sẽ được hiển thị ở danh sách liên lạc |
| **Luồng sử kiện chính** | 1. Người dùng bấm vào biểu tượng “Thêm” bên cạnh tên của những người trong danh sách tìm kiếm 2. Sau khi bấm, nút thêm sẽ bị khóa lại và tên liên lạc mới được hiển thị ở danh sách liên lạc |

|  |  |
| --- | --- |
| **Use Case** | **Hiển thị danh sách liên lạc** |
| **Điều kiện trước** | * Người dùng truy cập vào trang web * Người dùng đã đăng nhập vào hệ thống |
| **Điều kiện sau** | * Hiển thị danh sách những người dùng trong danh sách liên lạc |
| **Luồng sử kiện chính** | 1. Người dùng bấm vào chức năng “Liên lạc” 2. Hiển thị danh sách liên lạc của người dùng |

|  |  |
| --- | --- |
| **Use Case** | **Xóa liên lạc** |
| **Điều kiện trước** | * Người dùng truy cập vào trang web * Người dùng đã đăng nhập vào hệ thống * Người dùng bấm vào chức năng “Liên lạc” |
| **Điều kiện sau** | * Liên lạc bị xóa không xuất hiện trong danh sách liên lạc |
| **Luồng sử kiện chính** | 1. Người dùng bấm vào nút xóa trong danh sách liên lạc tương ứng với người cần xóa 2. Danh sách không hiển thị người vừa xóa |

|  |  |
| --- | --- |
| **Use Case** | **Xóa liên lạc** |
| **Điều kiện trước** | * Người dùng truy cập vào trang web * Người dùng đã đăng nhập vào hệ thống * Người dùng bấm vào chức năng “Liên lạc” |
| **Điều kiện sau** | * Liên lạc bị xóa không xuất hiện trong danh sách liên lạc |
| **Luồng sử kiện chính** | 1. Người dùng bấm vào nút xóa trong danh sách liên lạc tương ứng với người cần xóa 2. Danh sách không hiển thị người vừa xóa |

|  |  |
| --- | --- |
| **Use Case** | **Hiển thị danh sách tin nhắn gần nhất** |
| **Điều kiện trước** | * Người dùng truy cập vào trang web * Người dùng đã đăng nhập vào hệ thống |
| **Điều kiện sau** | * Hiên thị danh sách tin nhắn gần nhất |
| **Luồng sử kiện chính** | 1. Người dùng bấm vào chức năng “Tin nhắn” 2. Hiển thị danh sách liên lạc của người dùng 3. Các thông tin được hiện bao gồm: tên, thời gian, nội dung tin nhắn, số tin nhắn chưa đọc |

|  |  |
| --- | --- |
| **Use Case** | **Hiển thị lịch sử tin nhắn** |
| **Điều kiện trước** | * Người dùng truy cập vào trang web * Người dùng đã đăng nhập vào hệ thống |
| **Điều kiện sau** | * Hiên thị lịch sử tin nhắn của người dùng với liên lạc đó |
| **Luồng sử kiện chính** | 1. Người dùng bấm 2 lần liên tiếp vào các phần tử được hiển thị trong danh sách tin nhắn gần nhất hoặc trong danh sách liên lạc 2. Hiển thị lịch sử tin nhắn của người dùng |

|  |  |
| --- | --- |
| **Use Case** | **Gửi tin nhắn văn bản** |
| **Điều kiện trước** | * Người dùng truy cập vào trang web * Người dùng đã đăng nhập vào hệ thống * Người dùng sử dụng chức năng lịch sử tin nhắn |
| **Điều kiện sau** | * Gửi tin nhắn thành công tới người nhận |
| **Luồng sử kiện chính** | 1. Ở màn hình lịch sử tin nhắn, người dùng nhập chuỗi bất kỳ và bấm “Gửi” hoặc phím Enter 2. Tin nhắn mới sẽ được hiển thị trên màn hình lịch sử tin nhắn 3. Phía người nhận nhận được tin nhắn |

|  |  |
| --- | --- |
| **Use Case** | **Gửi tệp đính kèm** |
| **Điều kiện trước** | * Người dùng truy cập vào trang web * Người dùng đã đăng nhập vào hệ thống * Người dùng sử dụng chức năng lịch sử tin nhắn |
| **Điều kiện sau** | * Gửi tin nhắn thành công tới người nhận |
| **Luồng sử kiện chính** | 1. Ở màn hình lịch sử tin nhắn, bấm vào icon gửi tệp đính kèm 2. Người dùng chọn tệp cần gửi và gửi cho người nhận 3. Phía người nhận nhận được tệp |

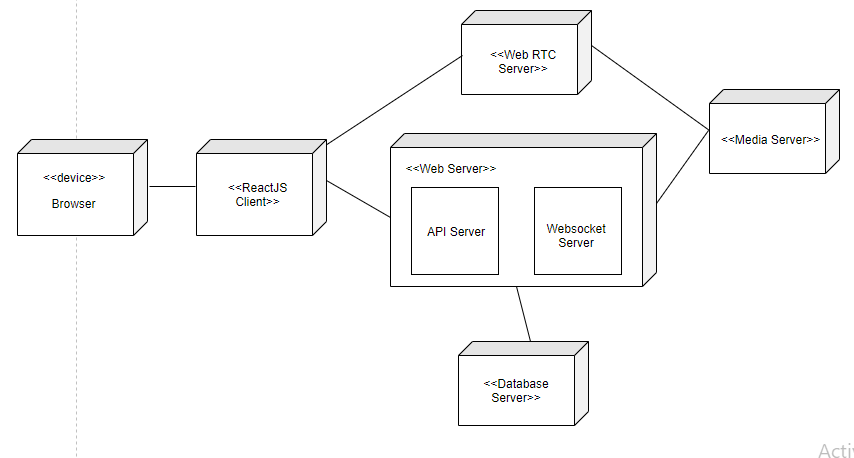
|  |  |
| --- | --- |
| **Use Case** | **Tạo cuộc trò chuyện nhóm** |
| **Điều kiện trước** | * Người dùng truy cập vào trang web * Người dùng đã đăng nhập vào hệ thống |
| **Điều kiện sau** | * Tạo một nhóm chat mới bao gồm các liên lạc đã được thêm |
| **Luồng sử kiện chính** | 1. Người dùng bấm vào nút “Tạo nhóm mới” 2. Hiển thi popupcho phép tìm kiếm và chọn liên lạc trong danh sách liên lạc để thêm vào nhóm mới 3. Người dùng bấm nút tạo nhóm |

|  |  |
| --- | --- |
| **Use Case** | **Xóa cuộc trò chuyện** |
| **Điều kiện trước** | * Người dùng truy cập vào trang web * Người dùng đã đăng nhập vào hệ thống |
| **Điều kiện sau** | * Cuộc trò chuyện biến mất khỏi danh sách tin nhắn gần nhất và không còn lịch sử chat giữa 2 người |
| **Luồng sử kiện chính** | 1. Người dùng bấm vào nút “Xóa cuộc trò chuyện” trên menu mở rộng của lịch sử tin nhắn |

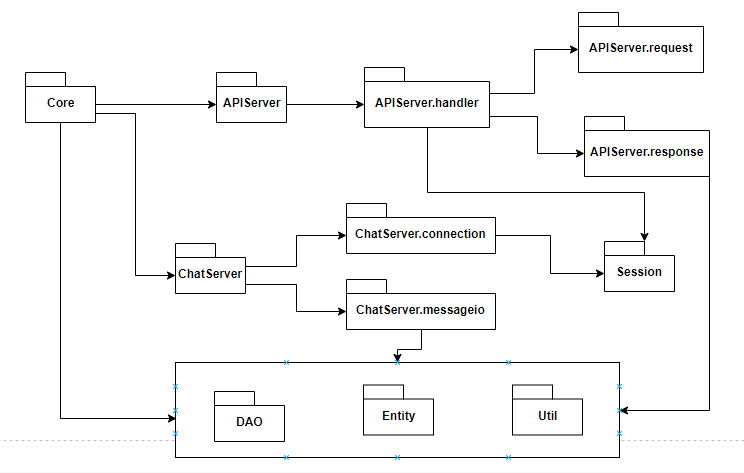
|  |  |
| --- | --- |
| **Use Case** | **Voice/Video call** |
| **Điều kiện trước** | * Người dùng truy cập vào trang web * Người dùng đã đăng nhập vào hệ thống |
| **Điều kiện sau** | * 2 người có thể nhìn, nghe thấy nhau và trò chuyện bình thường |
| **Luồng sử kiện chính** | 1. Người dùng bấm vào nút “Gọi” trên menu mở rộng của lịch sử tin nhắn 2. Phía người nhận sẽ hiển thị màn hình chờ khi người nhận chưa bắt máy 3. Khi người nhận bắt máy sẽ hiển thị một frame chứa hình ảnh của người nhận |
| **Luồng sự kiện thay thế** | Ở bước 3, Nếu người nhận không bắt máy sẽ không hiện bất kỳ frame nào và thông báo cho người dùng |

# **CHƯƠNG IV. THIẾT KẾ HỆ THỐNG**

1. **Mô hình kiến trúc**
2. **Deployment Diagram**



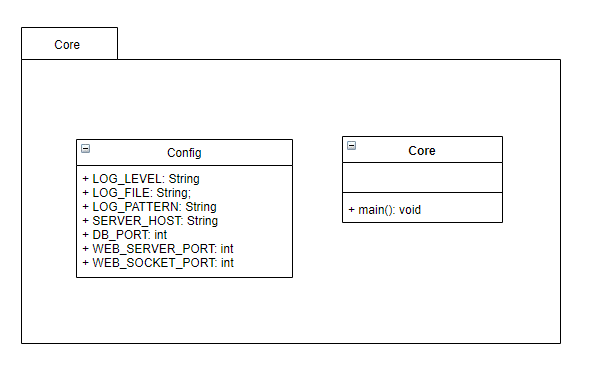
1. **Package Diagram**



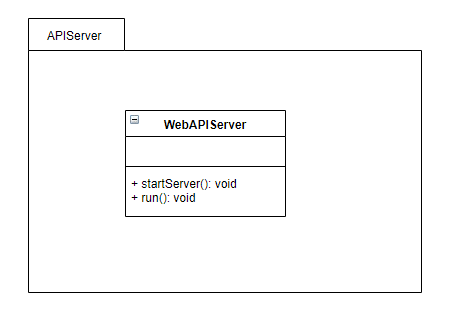
* Core: Chứa các configuration của server và chứa hàm main để khởi động toàn server
* DAO: Thực hiện việc kết nối tới DB
* Entity: Chứa các lớp thực thể
* Util: Chứa các hàm dùng chung và các constant
* APIServer: Thực hiện khởi tạo API Server
* APIServer.handler: Xử lý các request được gửi tới từ client
* ApiServer.request: Chứa thông tin request của Client
* ApiServer.response: Chứa các class xử lý api chuyên biệt
* ChatServer: Thực hiện khởi tạo WebSocket Server
* ChatServer.connection: Chứa các websocket connection đã được authen
* ChatServer.messageio: Thực hiện việc nhận và gửi dữ liệu qua socket
* Session: Chứa các phiên làm việc của người dùng, sử dụng để xác thực khi có yêu cầu xử lý hoặc yêu cầu kết nối từ của phía client

1. **Class Diagram**

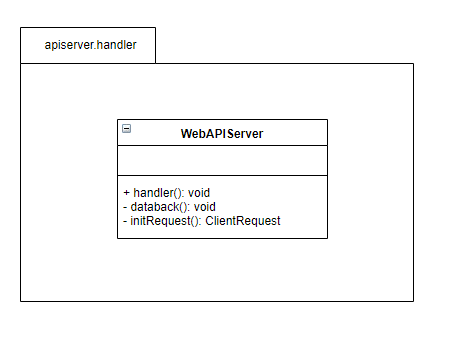
* Package core



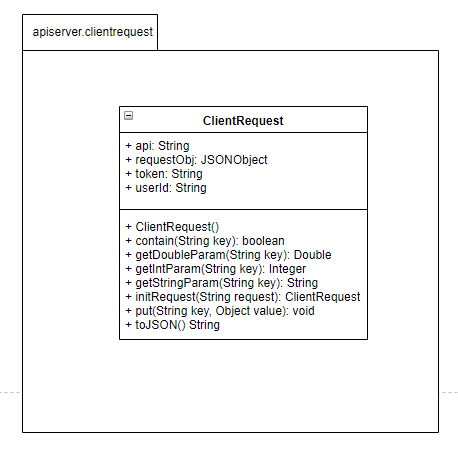
* Package apiserver



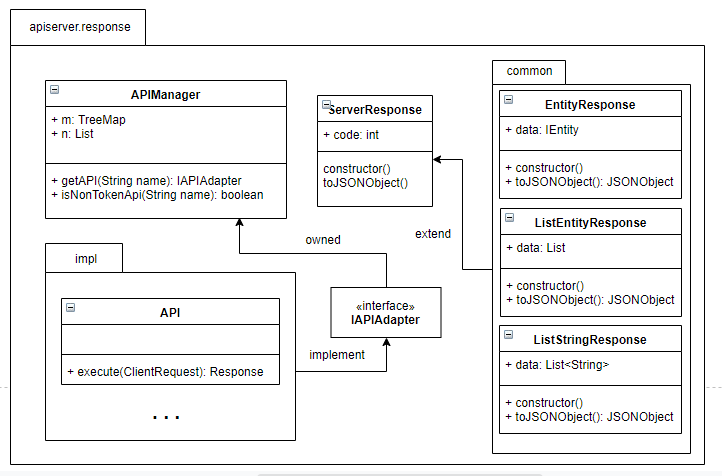
* Package apiserver.handler



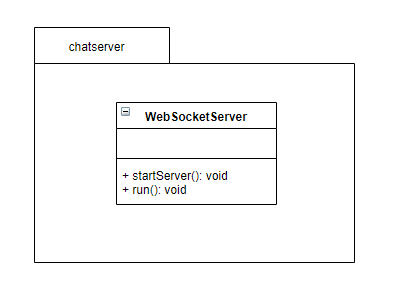
* Package apiserver.request



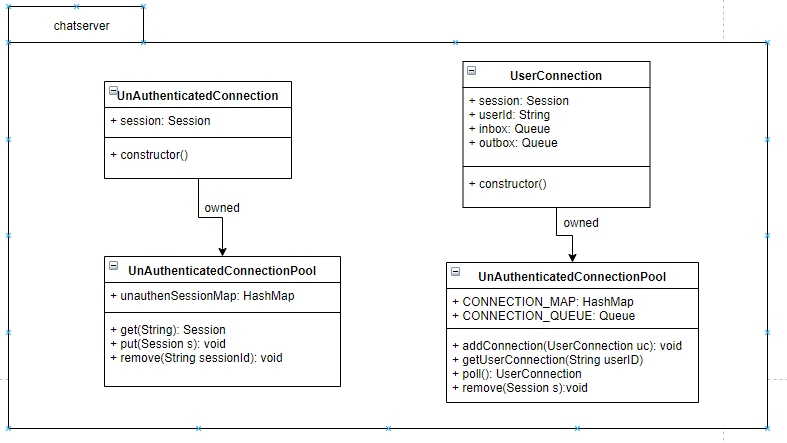
* Package apiserver.response



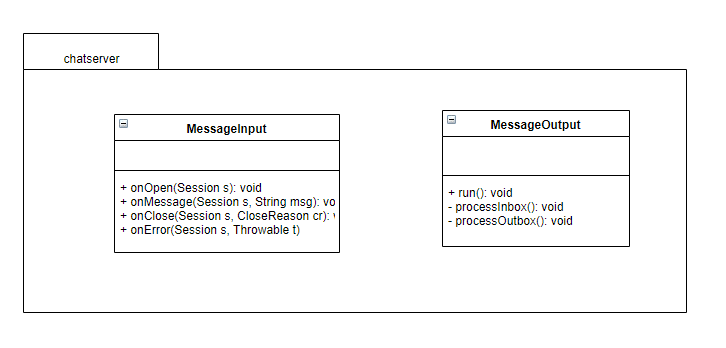
* Package chatserver



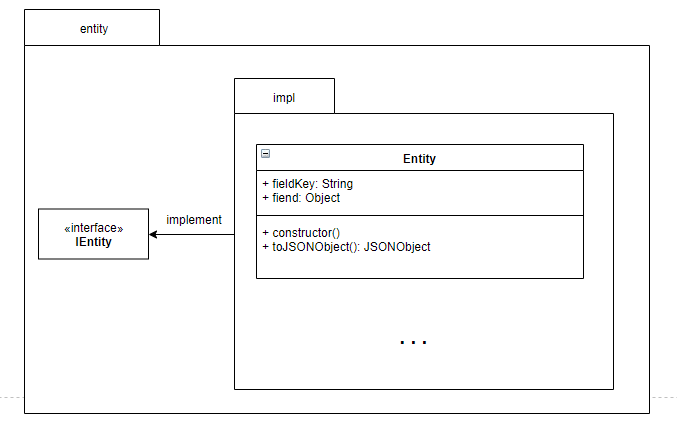
* Package chatserver.connection



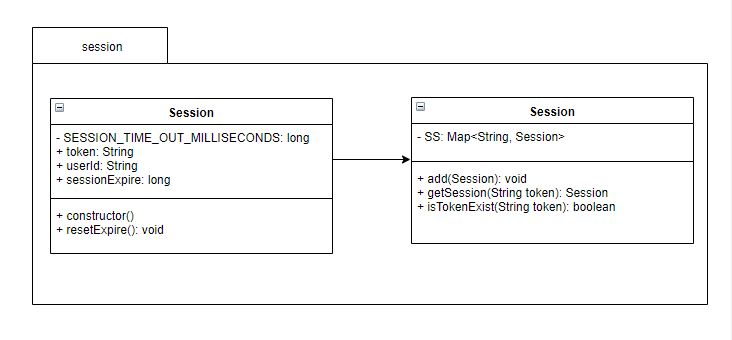
* Package chatserver.messageio



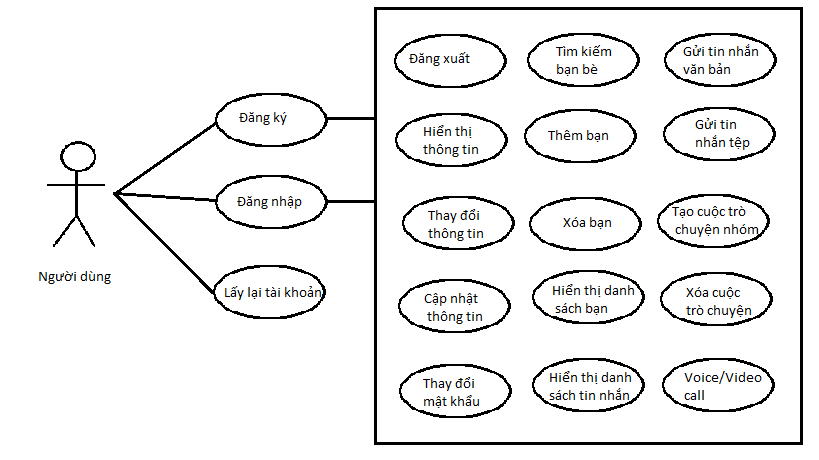
* Package entity



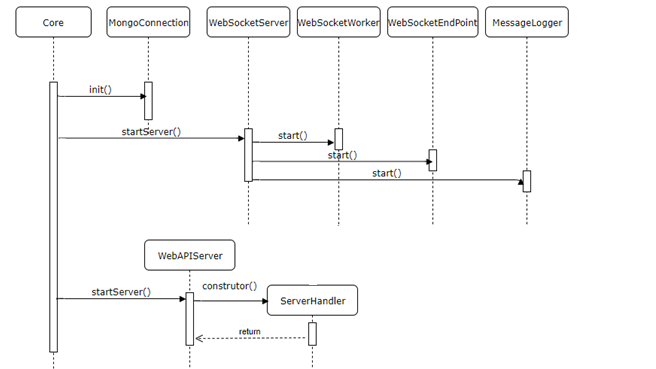
* Package session



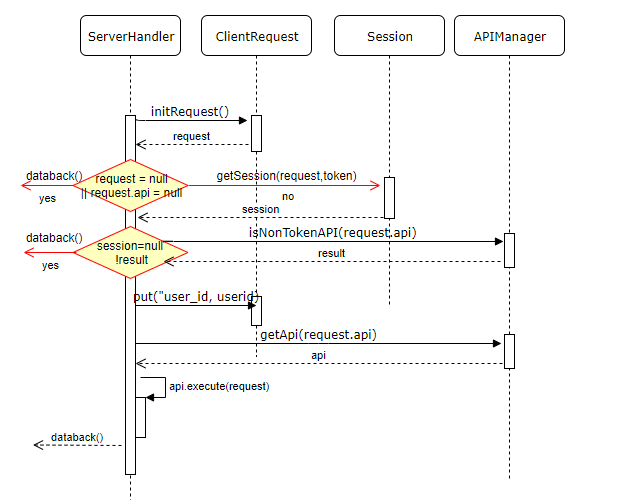
1. **Mô hình hoạt động**
2. **Use Case Diagram**

****

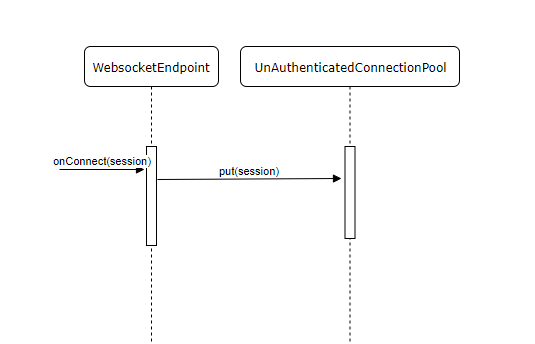
1. **Sequence Diagram**

****

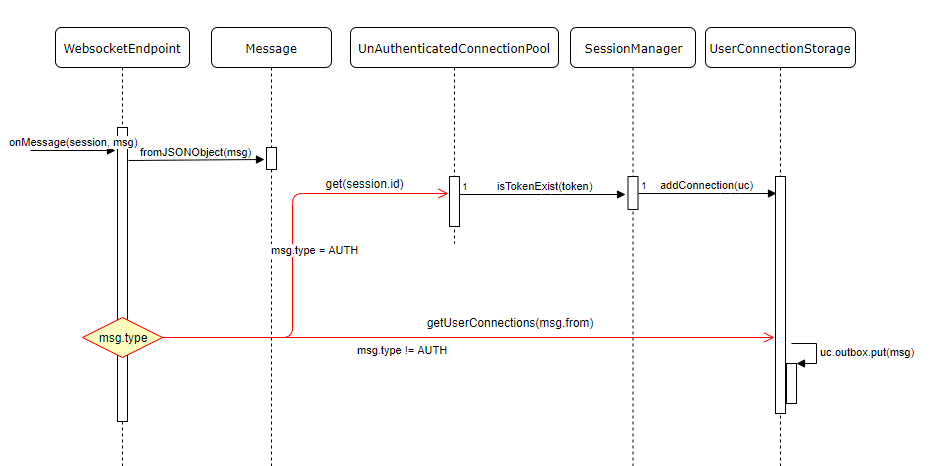
*Khởi tạo server*



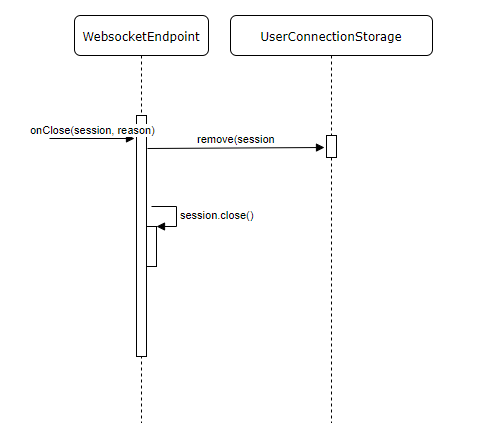
*Tiếp nhận http request từ client*



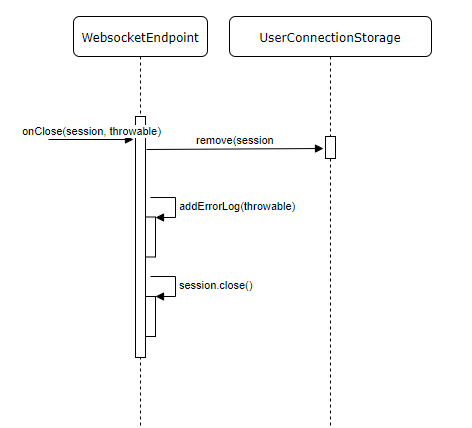
*Tiếp nhận yêu cầu kết nối qua qua Websocket*



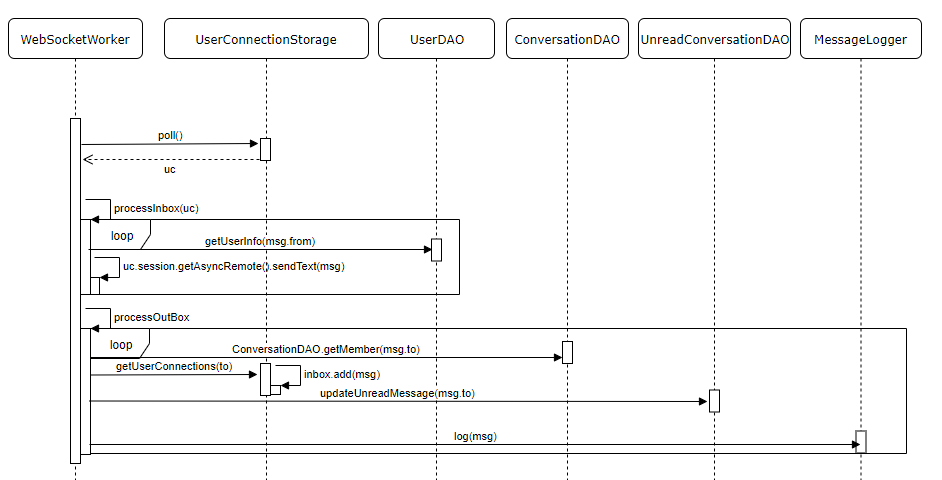
*Tiếp nhận tin nhắn qua socket*



*Tiếp nhận sự kiện đóng kết nối*



*Kết nối gặp Exception*



*Luồng xử lý tin nhắn và gửi cho người nhận*

1. **Thiết kế CSDL**
2. **Database user**

* Collection user: lưu thông tin người dùng

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| user\_id | String | Mã người dùng |
| email | String | Email của người dùng |
| user\_name | String | Tên người dùng, sử dụng để hiển thị |
| sort\_name | String | Tên người dùng, sử dụng để tìm kiếm |
| gender | Int32 | Giới tính  Nam: 0  Nữ: 1 |
| dob | String | Ngày sinh, theo format  yyyy-MM-dd |
| ava\_id | String | Mã ảnh đại diện |
| original\_pwd | String | Mật khẩu |
| pwd | String | Mât khẩu dạng mã hóa, sử dụng để kiểm tra đăng nhập |
| phone\_number | String | Số điện thoại |

* Collection contact: Lưu thông tin danh sách liên lạc

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| user\_id | String | Mã người dùng |
| friend\_id | String | Mã của liên lạc |
| time | String | Ngày thêm liên lạc, theo format yyyyMMddHHmmss |

* Collection converstion: Lưu thông tin cuộc trò chuyện

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| \_id | ObjectId | Mã người dùng |
| lst\_user | List | Danh sách những người trong cuộc trò chuyện |
| user\_id | String | Mã người dùng trong cuộc trò chuyện |
| conversation\_type | Int32 | Phân loại cuộc trò chuyện  0: tin nhắn riêng tư  1: nhóm |
| time | String | Ngày thêm cuộc trò chuyện, theo format yyyyMMddHHmmss |
| last\_message\_from | String | Người gửi tin nhắn cuối cùng |
| last\_message\_time | String | Thời gian gửi tin nhắn cuối cùng, theo format yyyyMMddHHmmss |
| last\_message\_type | String | Loại tin nhắn cuối cùng được gửi |
| last\_message\_value | String | Nội dung tin nhắn cuối cùng |

* Collection unread\_conversation: Lưu số lượng tin nhắn chưa đọc theo từng user

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| conversation\_id | String | Mã cuộc trò chuyện |
| user\_id | String | Mã người dùng |
| unread\_number | Int32 | Số tin nhắn chưa đọc |

1. **Database chatlog**

Collection trong database này sẽ lưu theo mã cuộc trò chuyện. Tức là collection name = conversation\_id

|  |  |  |
| --- | --- | --- |
| Tên trường | Kiểu dữ liệu | Mô tả |
| msg\_id | String | Mã tin nhắn |
| from | String | Mã người dùng gửi tin nhắn |
| type | String | Loại tin nhắn |
| value | String | Nội dung tin nhắn |
| time | String | Thời gian gửi tin nhắn, theo format yyyyMMddHHmmss |
| flag | Int32 | Trạng thái tin nhắn |