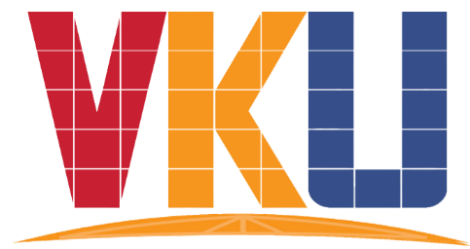
**ĐẠI HỌC ĐÀ NẴNG**

ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

VIỆT HÀN



**BÁO CÁO ĐỒ ÁN CƠ SỞ 4**

**TÌM HIỂU VÀ XÂY DỰNG ỨNG DỤNG CHAT**

**Giảng viên hướng dẫn : TRẦN ĐÌNH SƠN**

**Sinh viên thực hiện : NGUYỄN THỊ PHƯƠNG NHI**

**VÕ THỊ THÌN**

**Lớp : 18IT1**

***Đà Nẵng, tháng 01 năm 2021***

**ĐẠI HỌC ĐÀ NẴNG**

**ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT HÀN**

**BÁO CÁO ĐỒ ÁN CƠ SỞ 4**

**TÌM HIỂU VÀ XÂY DỰNG ỨNG DỤNG CHAT**

***Đà Nẵng, tháng 01 năm 2021***

**MỞ ĐẦU**

Trong những năm gần đây, bắt đầu từ 2010 trở lại đây. Cơn vũ bão smart phone đã tràn ngập trong nước ta. Ai ai cũng có một chiếc smart phone. Smart phone là điện thoại thông minh tích hợp một nền tảng di động với nhiều tính năng hỗ trợ tiên tiến về điện toán và kết nối trên nền tảng cơ bản của điện thoại di động thông thường.

Ban đầu điện thoại thông minh bao gồm các tính năng như điện thoại di động thông thường kết hợp với các thiết bị phổ biến như PDA, thiết bị điện tử cầm tay, máy ảnh kỹ thuật số, hệ thống định vị GPS. Điện thoại thông minh ngày nay bao gồm tất cả chức năng của laptop như duyệt web, Wi-Fi, các ứng dụng của bên thứ 3 trên di động và các phụ kiện đi kèm trên máy. Việc mua và dùng smart phone rất tirnj lợi vì nó có thể thay thế hoàn toàn một chiếc máy tính bình thường. Ngoài việc đó ra nó còn là công cụ để kết nối mọi người với nhau với những chức năng của một chiếc điện thoại thông thường hoặc dùng ứng dụng của bên thứ 3.

Mặc dù các ứng dụng để kết nối mọi người như Mạng xã hội rất tiện ích những nó cũng đem lại những cái hại như chúng ta mất quá nhiều thời gian vào Mạng xã hội... Chính vì vậy, chúng em đã lựa chọn đề tài “Tìm hiểu và xây dựng ứng dụng chat” trên hệ điều hành Android

**NHẬN XÉT**

MỤC LỤC

[CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI 7](#_Toc60251238)

[**1.** **Giới thiệu đề tài** 7](#_Toc60251239)

[**2.** **Khảo sát các ứng dụng tương tự** 7](#_Toc60251240)

[**2.1.** **Messenger** 7](#_Toc60251241)

[**2.2.** **Zalo** 7](#_Toc60251242)

[CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ 9](#_Toc60251243)

[**1.** **Cơ sở lý thuyết** 9](#_Toc60251244)

[**1.1.** **Hệ điều hành Android** 9](#_Toc60251245)

[**1.1.1.** **Giới thiệu** 9](#_Toc60251246)

[1.1.2. Các tính năng nổi bật của Android 10](#_Toc60251247)

[**1.2.** **Java** 12](#_Toc60251248)

[**1.2.1.** **Giới thiệu về ngôn ngữ Java** 12](#_Toc60251249)

[**1.2.2.** **Tính năng** 13](#_Toc60251250)

[**1.3.** **NodeJS** 15](#_Toc60251251)

[**1.3.1. Khái niệm NodeJS** 15](#_Toc60251252)

[**1.3.2.** **Các đặc tính của NodeJS** 16](#_Toc60251253)

[**1.3.3.** **Những ứng dụng viết bằng NodeJS** 17](#_Toc60251254)

[**1.4.** **MongoDB** 17](#_Toc60251255)

[**1.4.1.** **Định nghĩa** 17](#_Toc60251256)

[**1.4.2.** **Cách thức hoạt động** 17](#_Toc60251257)

[**1.4.3.** **Lợi thế** 18](#_Toc60251258)

[**1.5.** **Socketio** 18](#_Toc60251259)

[**2.** **Đặc tả yêu cầu** 21](#_Toc60251260)

[**2.3.** **Xác định các actor** 21](#_Toc60251261)

[**2.4.** **Xác định yêu cầu người sử dụng** 21](#_Toc60251262)

[**3.** **Biểu đồ use case** 21](#_Toc60251263)

[**3.1.** **Xây dựng biểu đồ use case** 21](#_Toc60251264)

[**3.2.** **Mô tả các use case** 21](#_Toc60251265)

[**4.** **Biểu đồ lớp** 24](#_Toc60251266)

[**5.** **Biểu đồ ER** 25](#_Toc60251267)

[**6.** **Thiết kế cơ sở dữ liệu** 25](#_Toc60251268)

[CHƯƠNG 3: TRIỂN KHAI VÀ KẾT QUẢ 25](#_Toc60251269)

[CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 25](#_Toc60251270)

[**1.** **Kết luận** 25](#_Toc60251271)

[**2.** **Ưu, nhược điểm** 25](#_Toc60251272)

[**3.** **Đánh giá, nhận xét** 25](#_Toc60251273)

[**4.** **Hướng phát triển** 25](#_Toc60251274)

# CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI

1. **Giới thiệu đề tài**
2. **Khảo sát các ứng dụng tương tự**
   1. **Messenger**

Facebook Messenger là một ứng dụng phần mềm tin nhắn tức thời chia sẻ giao tiếp qua văn bản và file (tập tin). Được tích hợp trên tính năng 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, máy tính và trên web.

Theo báo cáo 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 theo lời mời của [Mark Zuckerberg](https://vi.wikipedia.org/wiki/Mark_Zuckerberg), CEO của Facebook. Đến năm 2016, số người sử dụng Facebook Messenger đã lên tới hơn 1 tỉ người.

Messenger cho di động được phát hành vào ngày 9 tháng 8 năm 2011 cho [iOS](https://vi.wikipedia.org/wiki/IOS_(Apple)) và [Android](https://vi.wikipedia.org/wiki/Android_(h%E1%BB%87_%C4%91i%E1%BB%81u_h%C3%A0nh)), và với một ứng dụng nâng cấp vào ngày 11 tháng 10 cho [BlackBerry OS](https://vi.wikipedia.org/wiki/BlackBerry_OS). Vào tháng 12 năm 2012, Ứng dụng Facebook Messenger cho Android ở vài nơi (chẳng hạn như Úc, Nam Á, Indonesia, Nam Phi và Venezuela) được thêm vào Facebook chỉ cần tạo tên tài khoản và cung cấp số điện thoại. Sau đó "Chat Heads"được cải tiến và thêm vào trên Facebook trên Android.

Độ phủ sóng chưa đủ rộng rãi vì mới chỉ có phiên bản [Messenger cho iPhone](https://taimienphi.vn/download-messenger-cho-iphone-73469/173-0-phien-ban), Messenger cho Android, Messenger cho Windows Phone và phiên bản PC.

Messenger có dung lượng tối đa lên tới 80 MB, một con số khá lớn.

Nếu sử dụng Messenger bằng 3G sẽ hay xả ra tình trạng giật lag do server máy chủ đặt ở nước ngoài, đặc biệt tình trạng ngốn RAM trên Messenger đã được khẳng định là một trong những căn nguyên dẫn tới tình trạng giật lag.

* 1. **Zalo**

Zalo là ứng dụng đa chức năng hoạt động trên nền tảng di động và máy tính, được sử dụng tại các nước [Việt Nam](https://vi.wikipedia.org/wiki/Vi%E1%BB%87t_Nam), [Hoa Kỳ](https://vi.wikipedia.org/wiki/Hoa_K%E1%BB%B3), [Myanmar](https://vi.wikipedia.org/wiki/Myanmar), [Nhật Bản](https://vi.wikipedia.org/wiki/Nh%E1%BA%ADt_B%E1%BA%A3n), [Đài Loan](https://vi.wikipedia.org/wiki/%C4%90%C3%A0i_Loan), [Hàn Quốc](https://vi.wikipedia.org/wiki/H%C3%A0n_Qu%E1%BB%91c), [Malaysia](https://vi.wikipedia.org/wiki/Malaysia), [Ả Rập Xê Út](https://vi.wikipedia.org/wiki/%E1%BA%A2_R%E1%BA%ADp_X%C3%AA_%C3%9At), [Angola](https://vi.wikipedia.org/wiki/Angola), [Sri Lanka](https://vi.wikipedia.org/wiki/Sri_Lanka), [Cộng hòa Séc](https://vi.wikipedia.org/wiki/C%E1%BB%99ng_h%C3%B2a_S%C3%A9c), [Nga](https://vi.wikipedia.org/wiki/Nga).

Zalo được phát triển và phát hành bởi công ty cổ phần [VNG](https://vi.wikipedia.org/wiki/VNG). Phiên bản đầu tiên được ra mắt vào ngày 08/08/[2012](https://vi.wikipedia.org/wiki/2012) không nhận được sự quan tâm nhiều từ người dùng.[[5]](https://vi.wikipedia.org/wiki/Zalo#cite_note-5)

Tháng 12/[2012](https://vi.wikipedia.org/wiki/2012), Zalo chính thức ra mắt, đi theo mô hình mobile-first và nhanh chóng thu hút được một lượng lớn người dùng Việt Nam nhờ việc sản phẩm hoạt động tốt, ổn định trên hạ tầng mạng Việt Nam.

Tháng 02/2013 Zalo được bình chọn vào top những ứng dụng di động sáng tạo nhất châu Á trên Techinasia.

*Tính năng*

* Nhắn tin miễn phí (có thể chia sẻ biểu tượng cảm xúc, hình ảnh và video).
* Gọi điện miễn phí (gọi thoại và gọi video).
* Chia sẻ trạng thái (viết, đọc, đăng và bình luận trạng thái, nhưng chỉ xem được bình luận của bạn bè chung).
* Chia sẻ tập tin, video và lưu trữ thêm nhiều thông tin tại mục ***Truyền file***.
* Kết bạn (thông qua liên kết với danh bạ trên điện thoại và [Facebook](https://vi.wikipedia.org/wiki/Facebook), quét mã QR trực tiếp hoặc mục ***Tìm quanh đây***; lấy vị trí thiết bị của bạn làm tâm điểm rồi truy quét những người dùng Zalo khác trong bán kính lên đến hơn 2 km).
* Thanh toán bằng thẻ tín dụng qua ZaloPay.

Hiện nay Zalo đang hỗ trợ trên rất nhiều nền tang bao gồm phiên bản [Zalo cho Android](https://taimienphi.vn/download-zalo-cho-android-73278), Zalo cho iPhone, Zalo cho Windows Phone, thậm chí cũng đã có phiên bản cho Windows, Mac OS X, Linux và phiên bản web.

Nếu với kết nối Wifi, cả hai đều cho thấy khả năng xử lý tin nhắn khá tốt, độ trễ thấp, trong khi đó, Zalo có ưu thế hơn về khả năng sử dụng 3G do phần mềm này đã được nhà phát triển VNG tối ưu hóa cho kết nối mạng ở Việt Nam.

# CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ

1. **Cơ sở lý thuyết**
   1. **Hệ điều hành Android**
      1. **Giới thiệu**

Android là hệ điều hành di động được phát triển bởi Google (GOOGL), được sử dụng chủ yếu trong các thiết bị màn hình cảm ứng, điện thoại di động và máy tính bảng. Nó cho phép người dùng thao tác với các thiết bị một cách trực quan thông qua chuyển động của ngón tay như: chạm, vuốt. Google cũng sử dụng phần mềm Android trên tivi, ô tô và đồng hồ đeo tay - Mỗi thiết bị đều được thiết kế giao diện riêng.

*Lịch sử phát triển của HĐH Android*

Hệ điều hành này đầu tiên được Android.Inc. - một công ty phần mềm đặt tại Thung lũng Silicon - phát triển, sau đó được Google mua lại vào năm 2005. Mặc dù mã nguồn Android được phát hành ở định dạng mã nguồn mở để thuận tiện cho người dùng, nhưng nó vẫn có thể sử dụng cố định cho các phần mềm độc quyền trên các thiết bị di động. Các nhà đầu tư và chuyên gia trong lĩnh vực công nghiệp điện tử lúc đó đã đặt dấu chẩm hỏi về ý định thật sự của Google trong việc thâm nhập thị trường di động kể từ khi mua lại. Nhưng chỉ không lâu sau, Google đã công bố thiết bị chạy Android đầu tiên của mình ra thị trường vào năm 2007.

[](https://techvccloud.mediacdn.vn/2020/8/20/208a-1597909541132464454114.jpg)

*Android liên kết với nhiều ứng dụng Google*

Kể từ đó, các nhà phát triển phần mềm có thể sử dụng công nghệ Android để phát triển các ứng dụng dành cho thiết bị di động, được bán trên các cửa hàng ứng dụng. Bởi vì nó được phát triển như một sản phẩm của Google, nên người dùng Android có cơ hội liên kết thiết bị di động của họ với các sản phẩm khác của Google, chẳng hạn như nền tảng email, Google Drive.

Tính đến tháng 1 năm 2020, Android là hệ điều hành phổ biến nhất được sử dụng trên các thiết bị di động, với 74,3% thị phần toàn cầu. IOS của Apple đứng thứ hai với 24,8%.

### Các tính năng nổi bật của Android

Giao diện người dùng mặc định của Android sử dụng các thao tác trực tiếp như chạm, vuốt và kéo để bắt đầu giao tiếp. Thiết bị cung cấp phản hồi xúc giác cho người dùng thông qua các cảnh báo như rung để phản hồi các hành động. Ví dụ: nếu người dùng nhấn vào nút điều hướng, thiết bị sẽ rung.

Khi người dùng khởi động thiết bị, hệ điều hành Android sẽ hiển thị màn hình chính, đây là trung tâm điều hướng chính cho thiết bị Android, bao gồm các tiện ích và biểu tượng ứng dụng. Widget là màn hình thông tin tự động cập nhật nội dung như thời tiết hoặc tin tức. Màn hình chính có thể khác nhau tùy theo nhà sản xuất của thiết bị. Người dùng cũng có thể chọn các chủ đề khác nhau cho màn hình chính thông qua các ứng dụng của bên thứ ba trên Google Play .

Thanh trạng thái ở phía trên cùng màn hình chính hiển thị các thông tin và trạng thái kết nối của thiết bị: mạng Wi-Fi đang kết nối, sóng 3G, cường độ sóng điện thoại. Người dùng có thể vuốt thanh trạng thái theo chiều đi xuống để xem màn hình thông báo.

Hệ điều hành Android cũng sở hữu các tính năng giúp tiết kiệm pin. Hệ điều hành tạm dừng các ứng dụng không được sử dụng để tiết kiệm pin và mức sử dụng CPU. Android có các tính năng quản lý bộ nhớ giúp tự động đóng các trình xử lý không hoạt động lưu trong bộ nhớ.

[](https://techvccloud.mediacdn.vn/2020/8/20/208-15979096172241970906105.jpg)

* Android với tính năng picture-in-picture cho phép thao tác đa nhiệm
* Bluetooth
* Các giao thức truyền thông 3G như EVDO và HSDPA
* Wifi
* Tự động vá lỗi
* Tin nhắn SMS và MMS
* Camera tĩnh/Camera kỹ thuật số
* GPS
* Bản đồ
* Ứng dụng đa nhiệm/đa tác vụ (multitask)

*Phần cứng*

Android sử dụng ARM cho nền tảng phần cứng; các phiên bản sau của hệ điều hành Android hỗ trợ kiến trúc x86 và x86-64. Bắt đầu từ năm 2012, các nhà sản xuất thiết bị đã phát hành điện thoại thông minh và máy tính bảng Android với bộ vi xử lý Intel.

Yêu cầu phần cứng tối thiểu của Android phụ thuộc vào kích thước màn hình của thiết bị, loại và mật độ CPU. Ban đầu, Google yêu cầu bộ xử lý 200MHz, 32MB bộ nhớ và 32 MB RAM.

*So sánh hệ điều hành Android với các hệ điều hành di động khác*

Đối thủ cạnh tranh chính của Android là Apple iOS. Cả iOS và Android đều cung cấp các tính năng hữu ích cho người dùng. Apple iOS là hệ điều hành độc quyền với giao diện cố định, trong khi Android là hệ điều hành mã nguồn mở cung cấp nhiều tùy biến và linh hoạt hơn.

Android là hệ điều hành điện thoại thông minh bán chạy nhất kể từ năm 2011. Thị phần toàn cầu của Android từ năm 2018 đến 2019 là 74,45%, theo Statcounter. Thị phần toàn cầu của Apple iOS là 22,85%. Tuy nhiên, tại Mỹ, Apple thống trị thị phần với 57,22%; Samsung tuyên bố sở hữu 24,27%, tiếp theo là LG (5,49%) và Motorola (3,66%).

* 1. **Java**
     1. **Giới thiệu về ngôn ngữ Java**

Năm 1990, [James Gosling](https://vi.wikipedia.org/wiki/James_Gosling) tạo ra ngôn ngữ Oak trong dự án Green của Sun MicroSystems nhằm phát triển phần mềm cho các thiết bị dân dụng.Năm 1995, Oak được đổi tên thành **Java** và dần trở thành tiêu chuẩn cho công ngiệp Internet. Vì Java phát triển quá mạnh mẽ nên năm 2009, [Oracle](https://www.oracle.com/index.html) đã mạnh tay mua lại cả Sun MicorSystems để nắm ngôn ngữ Java trong tay.

Java là một [ngôn ngữ lập trình hướng đối tượng](https://niithanoi.edu.vn/hieu-ngay-9-khai-niem-lap-trinh-huong-doi-tuong-trong-java-qua-vi-du.html) (OOP) và dựa trên các lớp (class), thay vì biên dịch mã nguồn thành mã máy, Java được biên dịch thành bytecode, sau đó được chạy trên môi trường thực thi. Java vẫn đang là ngôn ngữ thống trị trong lĩnh vực lập trình mặc cho sự trỗi dậy của các ngôn ngữ như là [Python](https://niithanoi.edu.vn/lap-trinh-python.html) hay JavaScript.

Java hầu như được sử dụng rộng rãi trong hầu hết các lĩnh vực từ ngân hàng, tài chính, bán lẻ, ứng dụng di động, thị trường chứng khoán hay thậm chí là [Big data](https://niithanoi.edu.vn/khoa-hoc-big-data.html).

* + 1. **Tính năng**

Java sở hữu vốn tính năng đầy đủ nhất của một đàn anh thực thụ trong ngành. Bất cứ ai đã học qua Java rồi cũng cơ bản điểm qua được một số tính năng nổi bậc như sau:

  
  
Ngôn ngữ lập trình Java cung cấp 12 tính năng chính

***Java có tính Đơn giản***  
Java đã cải tiến dễ dàng hơn bằng cách loại bỏ tất cả những thứ phức tạp của các anh hàng xóm (chẳng hạn như con trỏ trong C++,..) để bản thân trở nên đơn giản và dễ tiếp cận với người học mà vẫn có hiệu suất hàng đầu.

***Java có tính Hướng đối tượng***  
Mọi thứ trong Java đều có thể xem là đối tượng – sở hữu các thuộc tính và hành vi.  
Java còn được coi là ngôn ngữ hỗ trợ lập trình hướng đối tượng đầy đủ và hoàn thiện nhất.

***Java có tính di động***  
Bạn hoàn toàn có thể đóng gói toàn bộ chương trình Java và di chuyển nó linh hoạt.

***Java có tính Độc lập với nền tảng***  
Java độc lập, không phụ thuộc vào nền tảng, có nghĩa là mọi ứng dụng sau khi được biên dịch có thể mang đi và thực thi trên các nền tảng khác. (Write Once, Run Anywhere).

***Java có tính Bảo mật***  
Tất cả các mã Java được chuyển đổi sang byteCode sau khi biên dịch, cho phép phát triển các hệ thống/ứng dụng không có virus, giả mạo.  
Hơn nữa, cú pháp của Java rất chặt chẽ nên giúp cho các chương trình Java thô sơ khi sinh ra đã có một mức độ bảo mật nhất định (Cho dù chưa thực hiện các biện pháp bảo mật)

***Java rất Mạnh mẽ***  
Java có khả năng thích ứng với môi trường phát triển hỗ trợ cấp phát bộ nhớ động do giảm lãng phí bộ nhớ và hiệu suất của ứng dụng được tăng lên.  
Có bộ gom rác tự động trong java chạy trên [Máy ảo Java](https://niithanoi.edu.vn/hieu-ngay-may-ao-java-jvm-va-kien-truc-cua-may-ao-java.html) để loại bỏ các đối tượng không còn được sử dụng bởi ứng dụng Java nữa.  
Có xử lý ngoại lệ và cơ chế kiểm tra kiểu trong Java. Tất cả những điểm này làm cho Java mạnh mẽ.

***Java có kiến trúc trung lập***  
Không giống như nhiều ngôn ngữ lập trình khác bao gồm C và C ++, khi Java được biên dịch, nó không được biên dịch thành máy cụ thể nền tảng, thay vào đó là biên dịch thành bytecode độc ​​lập với nền tảng.  
Bytecode này được phân phối trên web và được Máy ảo (JVM) giải thích trên bất kỳ nền tảng nào nó đang được chạy.  
Do đó, khi bạn viết một đoạn mã Java trong một nền tảng cụ thể và tạo ra một tệp. Class mã thực thi. Bạn có thể thực thi / chạy tệp. Class này trên bất kỳ hệ thống nào, điều kiện duy nhất là hệ thống đích phải được cài đặt JVM (JRE) trong đó.  
Nói tóm lại, trình biên dịch Java tạo ra một định dạng tệp đối tượng có kiến ​​trúc trung lập, làm cho mã được biên dịch có thể thực thi được trên nhiều bộ xử lý, với sự hiện diện của hệ thống Java Runtime.

***Java có tính động***  
Java là một ngôn ngữ động. Nó hỗ trợ tải động của các lớp. Nó có nghĩa là các lớp được tải theo yêu cầu. Nó cũng hỗ trợ các chức năng từ các ngôn ngữ bản địa của nó, tức là, C và C ++.  
Java hỗ trợ biên dịch động và quản lý bộ nhớ tự động (bộ sưu tập rác).

***Java có tính năng Biên dịch + Thông dịch***  
Java được biên dịch thành byteCode sau đó được thông dịch bởi môi trường Java runtime.

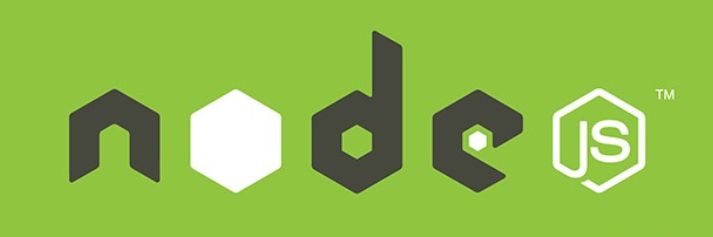
***Java có Hiệu suất cao***  
Java đạt được hiệu suất cao thông qua việc sử dụng byteCode có thể dễ dàng dịch sang mã máy và việc sử dụng các trình biên dịch JIT (Just-In-Time) giúp Java mang lại hiệu năng cao.

***Java hỗ trợ Đa luồng***Java hỗ trợ nhiều luồng thực thi, cho phép xử lý và thực hiện các tính toán song song.

***Java có tính phân tán***  
Java cung cấp một tính năng giúp tạo các ứng dụng phân tán. Sử dụng phương thức từ xa (RMI - Remote Method Invocation), một chương trình có thể gọi một phương thức của một chương trình khác thông qua và nhận được đầu ra, cho phép các đối tượng của một ứng dụng được phân bố và thực thi trên các máy tính khác nhau.

* 1. **NodeJS**

**1.3.1. Khái niệm NodeJS**



* Nodejs là một **nền tảng** (Platform) phát triển độc lập được xây dựng ở trên Javascript Runtime của Chrome mà chúng ta có thể xây dựng được các ứng dụng mạng một cách nhanh chóng và dễ dàng mở rộng.
* Nodejs được xây dựng và phát triển từ năm 2009, bảo trợ bởi công ty Joyent, trụ sở tại California, Hoa Kỳ.Phần Core bên dưới của Nodejs được viết hầu hết bằng C++ nên cho tốc độ xử lý và hiệu năng khá cao.
* Nodejs tạo ra được các ứng dụng có tốc độ xử lý nhanh, realtime thời gian thực.
* Nodejs áp dụng cho các sản phẩm có lượng truy cập lớn, cần mở rộng nhanh, cần đổi mới công nghệ, hoặc tạo ra các dự án Startup nhanh nhất có thể.
  + 1. **Các đặc tính của NodeJS**
* Không đồng bộ: Tất cả các API của NodeJS đều không đồng bộ (none-blocking) , nó chủ yếu dựa trên nền của NodeJS Server và chờ đợi Server trả dữ liệu về. Việc di chuyển máy chủ đến các API tiếp theo sau khi gọi và cơ chế thông báo các sự kiện của Node.js giúp máy chủ để có được một phản ứng từ các cuộc gọi API trước (Realtime).
* Chạy rất nhanh: NodeJ được xây dựng dựa vào nền tảng V8 Javascript Engine nên việc thực thi chương trình rất nhanh.
* Đơn luồng nhưng khả năng mở rộng cao: Node.js sử dụng một mô hình luồng duy nhất với sự kiện lặp. cơ chế tổ chức sự kiện giúp các máy chủ để đáp ứng một cách không ngăn chặn và làm cho máy chủ cao khả năng mở rộng như trái ngược với các máy chủ truyền thống mà tạo đề hạn chế để xử lý yêu cầu. Node.js sử dụng một chương trình đơn luồng và các chương trình tương tự có thể cung cấp dịch vụ cho một số lượng lớn hơn nhiều so với yêu cầu máy chủ truyền thống như Apache HTTP Server.
* Không đệm: NodeJS không đệm bất kì một dữ liệu nào và các ứng dụng này chủ yếu là đầu ra dữ liệu.
* Có giấy phép: NodeJS đã được cấp giấy phép bởi MIT License.
  + 1. **Những ứng dụng viết bằng NodeJS**
* **Websocket server:** Các máy chủ web socket như là Online Chat, Game Server…
* **Fast File Upload Client:** là các chương trình upload file tốc độ cao.
* **Ad Server:** Các máy chủ quảng cáo.
* **Cloud Services:** Các dịch vụ đám mây.
* **RESTful API:** đây là những ứng dụng mà được sử dụng cho các ứng dụng khác thông qua API.
* **Any Real-time Data Application:** bất kỳ một ứng dụng nào có yêu cầu về tốc độ thời gian thực. Micro Services: Ý tưởng của micro services là chia nhỏ một ứng dụng lớn thành các dịch vụ nhỏ và kết nối chúng lại với nhau. Nodejs có thể làm tốt điều này.
  1. **MongoDB**
     1. **Định nghĩa**
* MongoDB là một cơ sở dữ liệu mã nguồn mở và là cơ sở dữ liệu NoSQL hàng đầu, được hàng triệu người sử dụng. MongoDB được viết bằng C++.
* Ngoài ra, MongoDB là một cơ sở dữ liệu đa nền tảng, hoạt động trên các khái niệm Collection và Document, nó cung cấp hiệu suất cao, tính khả dụng cao và khả năng mở rộng dễ dàng.
  + 1. **Cách thức hoạt động**
* MongoDB hoạt động dưới một tiến trình ngầm service, luôn mở một cổng (Cổng mặc định là 27017) để lắng nghe các yêu cầu truy vấn, thao tác từ các ứng dụng gửi vào sau đó mới tiến hành xử lý.
* Mỗi một bản ghi của MongoDB được tự động gắn thêm một field có tên “\_id” thuộc kiểu dữ liệu ObjectId mà nó quy định để xác định được tính duy nhất của bản ghi này so với bản ghi khác, cũng như phục vụ các thao tác tìm kiếm và truy vấn thông tin về sau. Trường dữ liệu “\_id” luôn được tự động đánh index (chỉ mục) để tốc độ truy vấn thông tin đạt hiệu suất cao nhất.
* Mỗi khi có một truy vấn dữ liệu, bản ghi được cache (ghi đệm) lên bộ nhớ Ram, để phục vụ lượt truy vấn sau diễn ra nhanh hơn mà không cần phải đọc từ ổ cứng.
* Khi có yêu cầu thêm/sửa/xóa bản ghi, để đảm bảo hiệu suất của ứng dụng mặc định MongoDB sẽ chưa cập nhật xuống ổ cứng ngay, mà sau 60 giây MongoDB mới thực hiện ghi toàn bộ dữ liệu thay đổi từ RAM xuống ổ cứng.
  + 1. **Lợi thế**
* Ít schema hơn: Vì schema được sinh ra là để nhóm các đối tượng vào 1 cụm, dễ quản lý. Ví dụ như tạo 1 schema tên là Students chẳng hạn thì chỉ có những gì liên quan đến student thì mới được cho vào schema này. Trong khi đó trong mongodb thì chỉ 1 collection ta có thể chứa nhiều document khác nhau . Với mỗi document thì số trường, nội dung, kích thước lại có thể khác nhau.
* Cấu trúc của một đối tượng rõ ràng.
* Không có các Join phức tạp.
* Khả năng mở rộng cực lớn: việc mở rộng dữ liệu mà không phải lo đến các vấn đề như khóa ngoại, khóa chính, kiểm tra ràng buộc, ... MongoDB cho phép thực hiện replication và sharding nên việc mở rộng cũng thuận lợi hơn.
  + - Sử dụng bộ nhớ trong để lưu giữ cửa sổ làm việc cho phép truy cập dữ liệu nhanh hơn. Việc cập nhật được thực hiện nhanh gọn nhờ update tại chỗ (in-place).
  1. **Socketio**

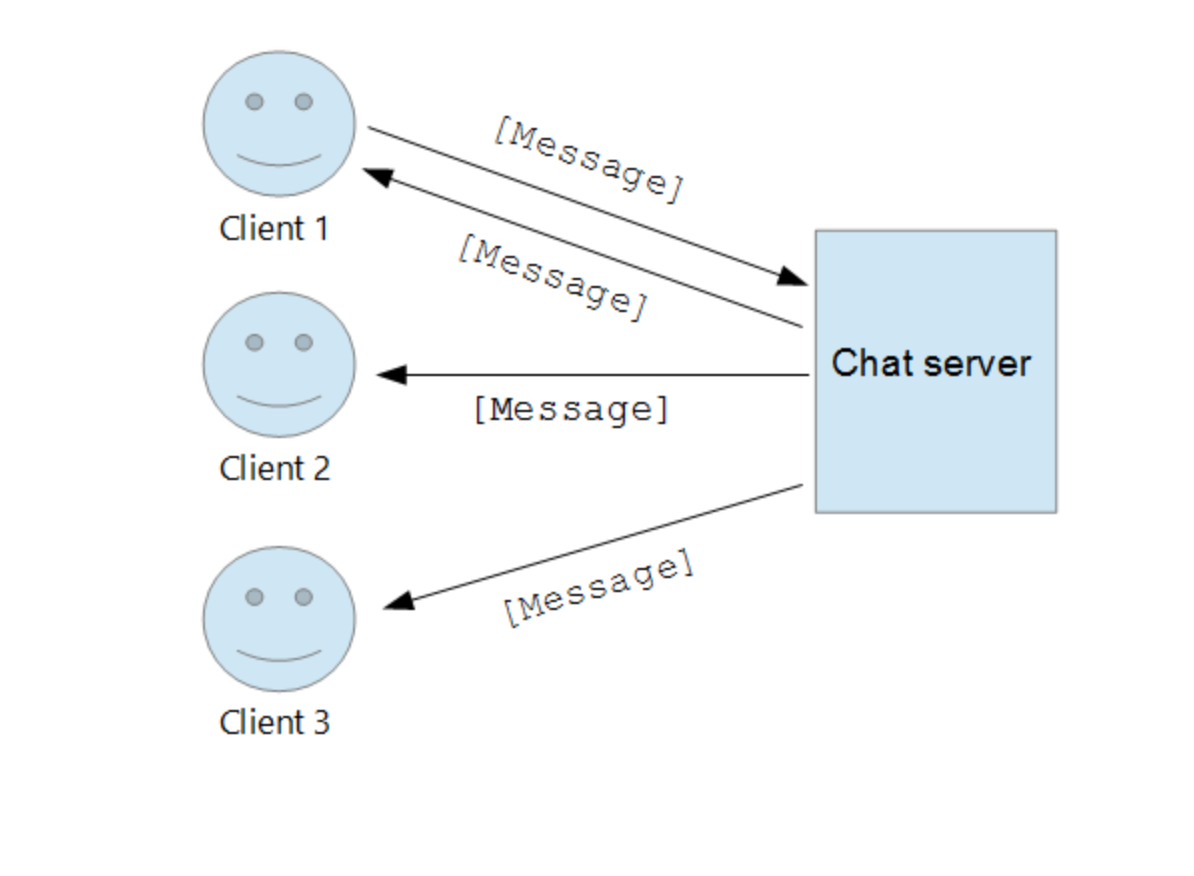
Socket.io là một module trong Node.js được phát triển vào năm 2010. Nó được phát triển để sử dụng các kết nối mở để tạo điều kiện giao tiếp thời gian thực, trả về giá trị thực ở tại thời điểm đó. Socket.io cho phép giao tiếp hai chiều giữa máy khách và máy chủ. Giao tiếp hai chiều được bật khi máy khách có Socket.io trong trình duyệt và máy chủ cũng đã tích hợp gói Socket.io

Nó được sử dụng trong việc xây dựng các ứng dụng web real-time cần tốc độ phản hồi ngay lập tức như: chat, trực tiếp bóng đá,.... Socket.io xây dựng dựa trên Engine.IO, đầu tiên nó sẽ thiết lập một kết nối *long-polling*, sau đó cố gắng nâng cấp lên các kết nối khác tốt hơn giống như WebSocket.

* + *Socket.IO - Đang hoạt động*

Một cách phổ biến để chứng minh giao tiếp hai chiều mà Socket.IO cung cấp là một ứng dụng trò chuyện cơ bản (chúng tôi nói về một số trường hợp sử dụng khác bên dưới). Với socket, khi máy chủ nhận được thông báo mới, nó sẽ gửi đến máy khách và thông báo cho họ, bỏ qua việc phải gửi yêu cầu giữa máy khách và máy chủ. Một ứng dụng trò chuyện đơn giản cho thấy cách này hoạt động.

Thời gian thực bạn có thể phụ thuộc vào. Ably cho phép bạn dễ dàng xây dựng các ứng dụng thời gian thực hoàn chỉnh, đơn giản hóa theo quy mô.



*Duy trì & Vận hành Socket.IO*

Như đã giải thích ở trên, việc bắt đầu với Socket.IO tương đối đơn giản - tất cả những gì bạn cần là một máy chủ Node.js để chạy nó. Nếu bạn muốn bắt đầu với một ứng dụng thời gian thực cho một số lượng người dùng hạn chế, Socket.IO là một lựa chọn tốt. Các vấn đề xảy ra khi làm việc ở quy mô lớn. Ví dụ: giả sử bạn muốn xây dựng một ứng dụng giống như CRM cho phép giao tiếp giữa các doanh nghiệp. Socket.IO được xây dựng trên các thư viện mạng không đồng bộ và sẽ gây ra tải trên máy chủ của bạn. Việc duy trì kết nối với người dùng cũng như gửi và nhận tin nhắn sẽ gây thêm căng thẳng và nếu khách hàng bắt đầu gửi một lượng lớn dữ liệu qua Socket.IO, nó sẽ truyền dữ liệu theo từng phần, giải phóng tài nguyên khi truyền dữ liệu. Vì vậy, khi ứng dụng của bạn thu hút nhiều người dùng hơn và máy chủ của bạn đạt mức tải tối đa, bạn sẽ cần chia kết nối qua nhiều máy chủ.

Thật không may, điều này không đơn giản như thêm một máy chủ khác. Sockets là một kết nối mở giữa máy chủ và máy khách. Máy chủ chỉ biết về các máy khách đã kết nối trực tiếp với nó chứ không phải những máy khách đã kết nối với các máy chủ khác. Quay trở lại chức năng trò chuyện, hãy tưởng tượng bạn muốn phát một thông báo đến tất cả người dùng rằng ai đó đã tham gia trò chuyện. Nếu họ được kết nối với một máy chủ khác, họ sẽ không nhận được thông báo này.

Để giải quyết vấn đề này, bạn cần có một quán rượu / cửa hàng phụ (ví dụ: Redis). Cửa hàng này sẽ giải quyết vấn đề nói trên bằng cách thông báo cho tất cả các máy chủ rằng họ cần gửi tin nhắn khi ai đó tham gia trò chuyện. Thật không may, điều này có nghĩa là một cơ sở dữ liệu bổ sung để duy trì mà rất có thể sẽ yêu cầu máy chủ riêng của nó.

Socket.IO đã tạo bộ điều hợp socket.io-adapter hoạt động với cửa hàng pub / sub và máy chủ để chia sẻ thông tin. Bạn có thể viết cách triển khai bộ điều hợp này của riêng mình hoặc bạn có thể sử dụng bộ điều hợp họ đã cung cấp cho Redis, may mắn thay, Socket.IO rất dễ tích hợp.

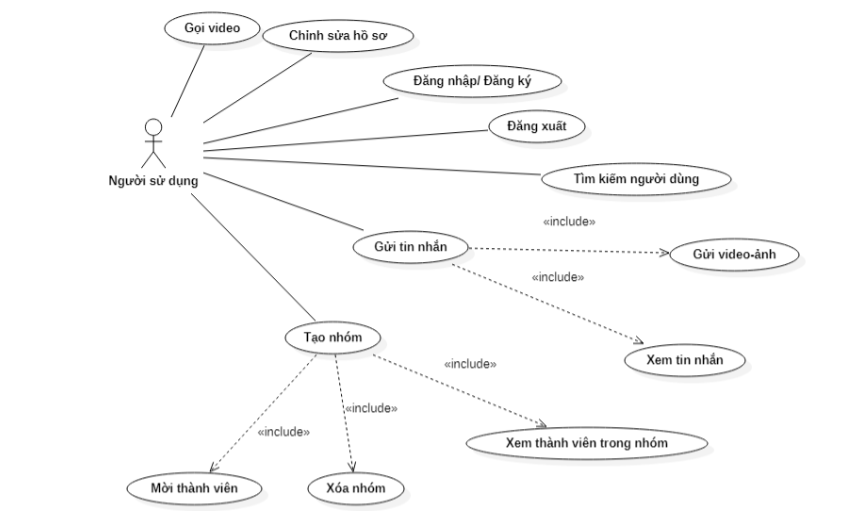
Các công cụ nâng cao độ tin cậy khác cho Socket.IO có thể bao gồm CoreOS để chia nhỏ kiến ​​trúc thành các đơn vị có thể được phân phối trên phần cứng có sẵn, giới thiệu các phiên bản mới khi tải tăng lên.

Một vấn đề khác với việc mở rộng quy mô Socket.IO là trong khi WebSockets giữ kết nối của họ đang mở, nếu kết nối trở lại trạng thái thăm dò thì sẽ có nhiều yêu cầu trong suốt thời gian kết nối. Khi một trong những yêu cầu này chuyển đến một máy chủ khác, bạn sẽ nhận được lỗi `Lỗi trong quá trình bắt tay WebSocket: Mã phản hồi không mong muốn: 400`.

Hai cách chính để giải quyết vấn đề này là định tuyến các máy khách dựa trên địa chỉ gốc của họ hoặc cookie. Socket.IO có  [tài liệu tuyệt vời](https://socket.io/docs/using-multiple-nodes)  về cách giải quyết vấn đề này cho các môi trường khác nhau.

Mặc dù Socket.IO có xu hướng có tài liệu tốt để khắc phục những hạn chế của nó, nhưng chúng thường được coi là 'biện pháp khắc phục' thay vì giải pháp. Nếu bạn có ý định mở rộng quy mô hơn nữa, những cách được đề xuất này sẽ làm tăng thêm độ phức tạp và thêm lợi nhuận do lỗi cho ngăn xếp của bạn.

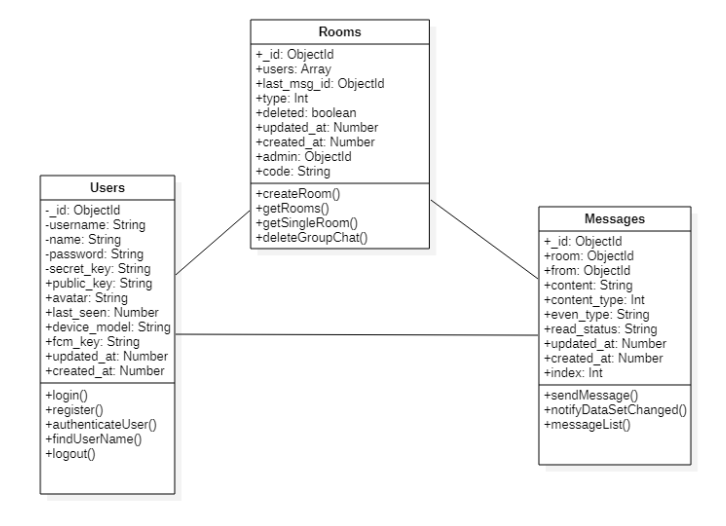
1. **Đặc tả yêu cầu**
   1. **Xác định các actor**
   2. **Xác định yêu cầu người sử dụng**
2. **Biểu đồ use case**
   1. **Xây dựng biểu đồ use case**



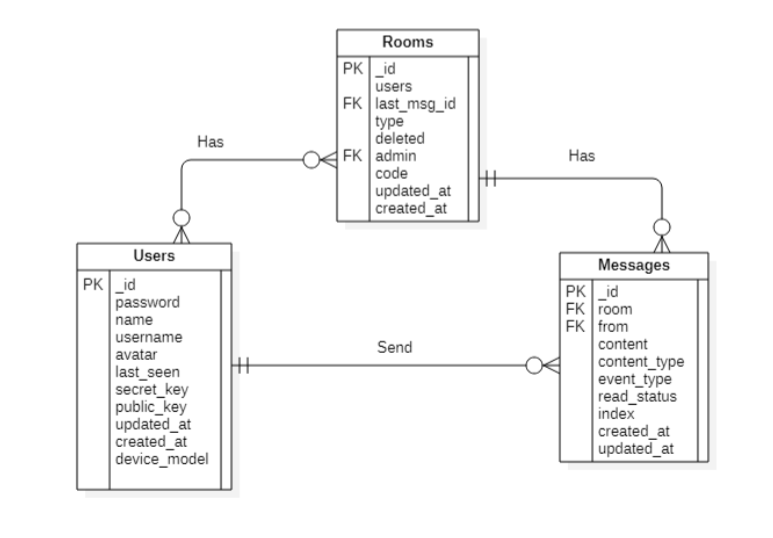
* 1. **Mô tả các use case**

|  |  |  |
| --- | --- | --- |
| **Use case** | **Mô tả** | **Các bước thực hiện** |
| Đăng nhập | Mô tả các bước đăng nhập của người dùng | Hệ thống yêu cầu người dùng cung cấp thông tin đăng nhập gồm tên đăng nhâp,mật khẩu.  Nhập xong bấm nút đăng nhập  Hệ thống kiểm tra thông tin và gửi thông báo thành công/thất bại.  Nếu thành công, chuyển qua trang chính. Nếu thất bại, yêu cầu đăng nhập lại. |
| Đăng xuất | Mô tả việc đăng xuất khỏi hệ thống | Người dùng thực hiện chức năng đăng xuất khỏi hệ thống.  Hệ thống hiện thị yêu cầu xác nhận từ người dùng  Người dùng xác nhận  Hệ thống đăng xuất tài khoản khỏi hệ thống. Nếu người dùng không xác nhận thì hệ thống sẽ giữ nguyên hiện trạng. |
| Gọi video | Mô tả việc gọi video với người dùng, nhóm | Người dùng đăng nhập vào hệ thống.  Chọn đối tượng hoặc nhóm cần gọi  Bấm vào biểu tượng call video  Hệ thống tự động kết nối người dùng với nhau.  Nếu có người nhận thì cuộc gọi được kết nối thành công.  Nếu thất bại, hệ thống hiển thị không kết nối được |
| Chỉnh sửa hồ sơ | Mô tả việc sửa thông tin, hồ sơ cá nhân | Người dùng đăng nhập vào hệ thống.  Vào mục hồ sơ cá nhân  Chọn mục cần chỉnh sửa  Thực hiện chỉnh sửa vào bấm nút lưu  Hệ thống kiểm tra thông tin và lưu thông tin vào cơ sở dữ liệu |
| Tìm kiếm người dùng | Mô tả việc tìm kiếm người dùng | Người dùng đăng nhập vào hệ thống  Vào mục tìm kiếm  Nhập tên hoặc các thông tin nhận dạng người dùng và bấm nút tìm kiếm  Hệ thống đối chiếu thông tin vừa nhập với cơ sở dữ liệu  Nếu tìm được thì hiển thị danh sách phù hợp với thông tin được nhập  Nếu thất bại thì hiện thông báo không có kết quả nào phù hợp |
| Gửi tin nhắn | Mô tả việc gửi tin nhắn cho người dùng hoặc nhóm | Người dùng đăng nhập vào hệ thống  Tìm kiếm người dùng hoặc nhóm cần liên hệ và bấm vào  Hệ thống hiển thị trang chat  Người dùng nhập đoạn tin nhắn vào text box phía dưới và nhấn nút gửi  Hệ thống xử lý thông tin và lưu thông tin  Nếu thành công, hiển thị thông báo đã gửi  Nếu thất bại, hiển thị thông báo lỗi |
| Gửi video-ảnh | Mô tả các bước gửi video hoặc ảnh cho người dùng | Người dùng đăng nhập hệ thống  Tìm kiếm người dùng hoặc nhóm cần liên hệ và bấm vào  Hệ thống hiển thị trang chat |
| Xem tin nhắn | Mô tả việc xem tin nhắn | Người dùng đăng nhập vào hệ thống  Tìm kiếm người dùng hoặc nhóm cần liên hệ và bấm vào  Hệ thống hiển thị trang chat |

1. **Biểu đồ lớp**



1. **Biểu đồ ER**



1. **Thiết kế cơ sở dữ liệu**

# CHƯƠNG 3: TRIỂN KHAI VÀ KẾT QUẢ

# CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. **Kết luận**
2. **Ưu, nhược điểm**
3. **Đánh giá, nhận xét**
4. **Hướng phát triển**