**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM**



**TÊN ĐỀ TÀI MÔN CÔNG NGHỆ MỚI TRONG PHÁT TRIỂN ỨNG DỤNG CNTT**

***Nhóm 04 - Sinh viên thực hiện***

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Họ và tên | Mã số sinh viên | Nhiệm vụ |
| 1 | Nguyễn Khánh An (NT) | 20122281 | Back-end |
| 2 | Lê Hữu Hiệp | 20115701 | Web |
| 3 | Lê Thanh Tùng | 20008831 | Web |
| 4 | Phan Nguyễn Hoài Hiệp | 20111601 | App |
| 5 | Nguyễn Tuấn Hiệp | 20111311 | App |

# MỤC LỤC

[MỤC LỤC 1](#_Toc164413465)

[DANH MỤC CÁC HÌNH VẼ 3](#_Toc164413466)

[DANH MỤC CÁC BẢNG BIỂU 4](#_Toc164413467)

[CHƯƠNG 1 : GIỚI THIỆU 5](#_Toc164413468)

[1.1 Tổng quan 5](#_Toc164413469)

[1.2 Mục tiêu đề tài 5](#_Toc164413470)

[1.3 Phạm vi đề tài 5](#_Toc164413471)

[1.4 Mô tả yêu cầu chức năng 5](#_Toc164413472)

[1.4.1. Yên cầu chức năng 5](#_Toc164413473)

[1.4.2. Yêu cầu phi chức năng 6](#_Toc164413474)

[CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT 7](#_Toc164413475)

[2.1 Cloud Computing với AWS 7](#_Toc164413476)

[2.1.1. Cloud computing: 7](#_Toc164413477)

[2.1.2. Amazon Web Services (AWS): 7](#_Toc164413478)

[2.1.3. Tích hợp AWS vào Ứng dụng Chat: 7](#_Toc164413479)

[2.2 Node.js 7](#_Toc164413480)

[2.2.1. Giới thiệu về Node.js: 7](#_Toc164413481)

[2.2.2. Cấu trúc và Mô hình Lập trình trong Node.js 7](#_Toc164413482)

[2.2.3. Ứng dụng của Node.js trong Ứng dụng Chat 7](#_Toc164413483)

[2.3 MongoDB / DynamoDB 7](#_Toc164413484)

[2.3.1. MongoDB 7](#_Toc164413485)

[2.3.2. Amazon DynamoDB 8](#_Toc164413486)

[2.4 React native 9](#_Toc164413487)

[2.5 NextJS 11](#_Toc164413488)

[CHƯƠNG 3 : PHÂN TÍCH VÀ THIẾT KẾ 13](#_Toc164413489)

[3.1 Phân tích yêu cầu bằng UML 13](#_Toc164413490)

[3.1.1 Usecase tổng quát 13](#_Toc164413491)

[3.1.2 Danh sách tác nhân và mô tả 13](#_Toc164413492)

[3.1.3 Danh sách các tình huống hoạt động (Use cases) 13](#_Toc164413493)

[3.1.4 Tình huống hoạt động 15](#_Toc164413494)

[3.2 Đặc tả các yêu cầu chức năng 15](#_Toc164413495)

[3.2.1 Usecase đăng kí 15](#_Toc164413496)

[3.2.2 Usecase đăng nhập 18](#_Toc164413497)

[3.2.3 Usecase lấy lại mật khẩu 21](#_Toc164413498)

[3.2.4 Usecase gửi tin nhắn 24](#_Toc164413499)

[3.2.5 Usecase tạo nhóm chat 26](#_Toc164413500)

[3.2.6 Usecase gửi file 28](#_Toc164413501)

[3.3 Class diagram 30](#_Toc164413502)

[3.4 Deployment diagram 31](#_Toc164413503)

[CHƯƠNG 4 : HIỆN THỰC 31](#_Toc164413504)

[4.1 Cấu hình phần cứng, phần mềm 31](#_Toc164413505)

[4.1.1. Thông tin 31](#_Toc164413506)

[4.1.2. Cấu hình phần cứng và phần mềm 32](#_Toc164413507)

[4.2 Giao diện của hệ thống 32](#_Toc164413508)

[4.2.1. Giao diện mobile 32](#_Toc164413509)

[4.2.2. Giao diện của website 32](#_Toc164413510)

[4.3 Kế hoạch và hiện thực kiểm thử hệ thống 32](#_Toc164413511)

[4.3.1 Kế hoạch kiểm thử 32](#_Toc164413512)

[4.3.2 Kiểm thử hệ thống 33](#_Toc164413513)

[CHƯƠNG 5 : KẾT LUẬN 36](#_Toc164413514)

[5.1 Kết quả đạt được 36](#_Toc164413515)

[5.2 Hạn chế của đồ án 36](#_Toc164413516)

[5.3 Hướng phát triển 36](#_Toc164413517)

[TÀI LIỆU THAM KHẢO 37](#_Toc164413518)

# DANH MỤC CÁC HÌNH VẼ

**Chương 2**

[Hình 2 - 1 Amazon Web Service (AWS) 6](#_Toc164388601)

[Hình 2 - 2 Node.js development 6](#_Toc164388602)

[Hình 2 - 3 MongoDB 6](#_Toc164388603)

[Hình 2 - 4 Amazon DynamoDB 7](#_Toc164388604)

[Hình 2 - 5 Logo React Native 8](#_Toc164388605)

[Hình 2 - 6 Logo Next.js 10](#_Toc164388606)

**Chương 3**

[Hình 3 - 1 Usecase tổng quát 12](#_Toc164295013)

# DANH MỤC CÁC BẢNG BIỂU

[Bảng 3‑1 Danh sách tác nhân và mô tả 1](#_Toc14682192)

[Bảng 3‑2 Danh sách các tình huống trong hệ thống 1](#_Toc14682193)

# : GIỚI THIỆU

## Tổng quan

Ứng dụng Chat là một ứng dụng di động (sử dụng React Native) và web (sử dụng Nex.js) cũng kết nối với giao qua một server (sử dụng Node.js) cho phép người dùng trò chuyện và giao tiếp với nhau thông qua tin nhắn văn bản, hình ảnh và tệp đính kèm. Ứng dụng này sử dụng dịch vụ lưu trữ đám mây của AWS để lưu trữ dữ liệu người dùng, bao gồm cả tin nhắn và tệp đính kèm.

## Mục tiêu đề tài

Mục tiêu của dự án này là tạo ra một ứng dụng chat linh hoạt, dễ sử dụng và bảo mật, mang lại trải nghiệm giao tiếp tốt nhất cho người dùng. Ứng dụng sẽ cung cấp các tính năng như tạo tài khoản người dùng, đăng nhập, tìm kiếm bạn bè, tạo nhóm chat, gửi tin nhắn, chia sẻ hình ảnh và tệp đính kèm, và xem thông tin người dùng.

## Phạm vi đề tài

* Ứng dụng được tạo ra cho để cho các người dùng có thể trao đổi thông tin, tài liệu thông qua việc kết bạn với nhau.
* Ứng dụng có thể được sử dụng qua web hoặc app di động
* Sử dụng cơ sở dữ liệu NoSQL.
* Sử dụng React-Native để tạo ra ứng dụng di động, Next.js để tạo web và Node.js để viết Back-end.
* Các phần mềm hỗ trợ: Visual Studio Code, draw.io, DBdiagram.

## Mô tả yêu cầu chức năng

1. **Yên cầu chức năng**

* Tính năng Người dùng:
* Đăng nhập và đăng ký tài khoản người dùng.
* Quản lý thông tin cá nhân của người dùng.
* Tìm kiếm và thêm bạn bè.
* Xem danh sách bạn bè và thông tin của họ.
* Tính năng Nhóm Chat:
* Tạo và quản lý nhóm chat.
* Thêm và xóa thành viên khỏi nhóm.
* Gửi tin nhắn trong nhóm chat.
* Tính năng Tin nhắn:
* Gửi và nhận tin nhắn văn bản.
* Gửi và nhận hình ảnh và tệp đính kèm.
* Xóa tin nhắn.
* Tính năng Bảo mật:
* Xác thực người dùng.
* Bảo vệ dữ liệu người dùng với mã hóa.
* Quản lý quyền truy cập vào nhóm chat và tin nhắn.
* Tính năng khác: …

1. **Yêu cầu phi chức năng**
   * Tốc độ xử lý nhanh
   * Hệ thống phải có giao diện đẹp, dễ sử dụng và chạy ổn định
   * Có tính bảo mật, tính đúng đắn, độ tin cậy cao
   * Ràng buộc về ngân sách
   * Phù hợp với các chính sách của tổ chức sử dụng hệ thống
   * Yêu cầu tương thích giữa phần cứng và phần mềm

# : CƠ SỞ LÝ THUYẾT

## Cloud Computing với AWS



Hình 2 - 1 Amazon Web Service (AWS)

1. **Cloud computing:**

+ **Khái niệm:** Cloud Computing là mô hình cung cấp và sử dụng các dịch vụ máy tính thông qua internet. Thay vì sử dụng tài nguyên máy tính cục bộ hoặc máy chủ riêng, người dùng có thể truy cập vào các dịch vụ máy tính như lưu trữ, máy chủ, cơ sở dữ liệu và ứng dụng từ các nhà cung cấp dịch vụ đám mây.

+ **Lợi ích:**

* Linh hoạt và mở rộng dễ dàng: Người dùng có thể tăng giảm tài nguyên theo nhu cầu một cách linh hoạt và dễ dàng.
* Tiết kiệm chi phí: Không cần phải đầu tư lớn cho cơ sở hạ tầng máy tính vì người dùng chỉ trả tiền cho tài nguyên mà họ sử dụng.
* Khả năng mở rộng toàn cầu: Các nhà cung cấp đám mây như AWS cung cấp cơ sở hạ tầng trên toàn thế giới, giúp các ứng dụng có thể mở rộng một cách toàn cầu.

1. **Amazon Web Services (AWS):**

+ **Giới thiệu:** AWS là một trong những nhà cung cấp dịch vụ đám mây lớn nhất và phổ biến nhất trên thế giới. Nó cung cấp một loạt các dịch vụ phong phú, bao gồm tính toán, lưu trữ, cơ sở dữ liệu, máy chủ ảo, dịch vụ AI/ML và nhiều dịch vụ khác.

+ Các Dịch vụ Chính của AWS:

* Amazon EC2 (Elastic Compute Cloud): Dịch vụ máy ảo đa năng cho phép triển khai ứng dụng và tài nguyên tính toán.
* Amazon S3 (Simple Storage Service): Dịch vụ lưu trữ đối tượng với khả năng lưu trữ dữ liệu không giới hạn.
* Amazon RDS (Relational Database Service): Dịch vụ cơ sở dữ liệu quan hệ quản lý được, bao gồm MySQL, PostgreSQL, SQL Server và Oracle.
* Amazon DynamoDB: Dịch vụ cơ sở dữ liệu NoSQL quản lý được, với khả năng mở rộng tự động và hiệu suất cao.
* Amazon API Gateway: Dịch vụ quản lý và xây dựng các API cho ứng dụng.

+Lợi ích của việc sử dụng AWS:

* Đa dạng và linh hoạt: AWS cung cấp một loạt các dịch vụ phong phú, cho phép người dùng tùy chỉnh và cấu hình theo nhu cầu cụ thể của họ.
* Độ tin cậy cao: AWS cung cấp một môi trường đám mây phân tán và có sự đa dạng hóa, giúp đảm bảo sự ổn định và tin cậy của ứng dụng.
* Tiết kiệm chi phí: Với mô hình thanh toán theo sử dụng, người dùng chỉ trả tiền cho tài nguyên thực sự sử dụng, giúp tiết kiệm chi phí hạ tầng.

1. **Tích hợp AWS vào Ứng dụng Chat:**

* Sử dụng Amazon S3 cho Lưu trữ Dữ liệu: Dữ liệu hình ảnh và tệp đính kèm có thể được lưu trữ trong Amazon S3 để đảm bảo tính nhất quán và tin cậy.
* Sử dụng Amazon EC2 cho Máy Chủ Ứng dụng: Máy chủ ứng dụng của ứng dụng chat có thể được triển khai trên Amazon EC2 để đảm bảo khả năng mở rộng và tính sẵn sàng cao.
* Sử dụng Amazon RDS hoặc Amazon DynamoDB cho Cơ sở Dữ liệu: Cơ sở dữ liệu cho ứng dụng chat có thể được quản lý bằng Amazon RDS (cho cơ sở dữ liệu quan hệ) hoặc Amazon DynamoDB (cho cơ sở dữ liệu NoSQL) để đảm bảo hiệu suất và khả năng mở rộng.

## Node.js



Hình 2 - 2 Node.js development

* 1. **Giới thiệu về Node.js:**

**+ Khái niệm:** Node.js là một nền tảng phát triển ứng dụng máy chủ xây dựng trên JavaScript runtime của Chrome V8 Engine. Nó cho phép chúng ta viết mã JavaScript phía máy chủ để xây dựng các ứng dụng web và API.

**+ Lịch sử**: Node.js được phát triển bởi Ryan Dahl vào năm 2009 và đã nhanh chóng trở thành một trong những công nghệ phát triển web phổ biến nhất thế giới.

**+ Tính năng nổi bật**:

* Single-threaded và non-blocking I/O: Node.js sử dụng mô hình xử lý không đồng bộ để xử lý các yêu cầu I/O mà không chặn luồng chính.
* Cộng đồng lớn và hỗ trợ mạnh mẽ: Node.js có một cộng đồng phát triển rất lớn và nhiều công cụ hỗ trợ mạnh mẽ từ cộng đồng.

1. **Cấu trúc và Mô hình Lập trình trong Node.js**

* Module và NPM (Node Package Manager): Node.js sử dụng hệ thống module để tổ chức và tái sử dụng mã. NPM là một công cụ quản lý gói package) cho phép bạn tải về, cài đặt và quản lý các thư viện và công cụ được viết bằng JavaScript.
* Event Loop: Node.js sử dụng một Event Loop để xử lý các sự kiện và yêu cầu không đồng bộ một cách hiệu quả. Event Loop là một cơ chế quan trọng trong Node.js giúp đảm bảo tính nhất quán và hiệu suất cao.
* Express.js: Express.js là một framework phổ biến cho Node.js, cung cấp các tính năng mạnh mẽ để xây dựng các ứng dụng web và API một cách dễ dàng và nhanh chóng.

1. **Ứng dụng của Node.js trong Ứng dụng Chat**

* Xử lý Yêu cầu và Phản hồi: Node.js cho phép xử lý các yêu cầu HTTP từ các client và gửi phản hồi trả về.
* Xử lý Dữ liệu Thời gian thực: Node.js cung cấp khả năng xử lý dữ liệu thời gian thực như tin nhắn từ người dùng và gửi lại các thông báo mới đến client.
* Tích hợp với Cơ sở Dữ liệu: Node.js có thể tích hợp với các cơ sở dữ liệu phổ biến như MongoDB hoặc MySQL để lưu trữ và truy xuất dữ liệu.

## MongoDB / DynamoDB

* 1. **MongoDB**



Hình 2 - 3 MongoDB

+ Giới thiệu về MongoDB: MongoDB là một hệ thống quản lý cơ sở dữ liệu NoSQL, mã nguồn mở, được thiết kế để lưu trữ dữ liệu dưới dạng các tài liệu JSON linh hoạt.

+ Lợi ích của MongoDB:

* Linh hoạt: MongoDB cho phép lưu trữ dữ liệu ở dạng tài liệu có cấu trúc linh hoạt, giúp dễ dàng mở rộng và điều chỉnh cấu trúc dữ liệu theo nhu cầu của ứng dụng.
* Tính sẵn sàng cao: MongoDB hỗ trợ replica set và sharding để đảm bảo tính sẵn sàng và hiệu suất cao.
* Dễ dàng tích hợp: MongoDB có nhiều thư viện và driver cho các ngôn ngữ lập trình phổ biến, giúp dễ dàng tích hợp với các ứng dụng hiện có.

+ Sử dụng MongoDB trong Ứng dụng Chat: MongoDB có thể được sử dụng để lưu trữ dữ liệu tin nhắn, thông tin người dùng và dữ liệu nhóm chat trong ứng dụng Chat của bạn. Dưới đây là một số ví dụ về cách sử dụng MongoDB:

* Lưu trữ tin nhắn: Mỗi tin nhắn có thể được lưu trữ dưới dạng một bản ghi JSON trong một collection riêng.
* Lưu trữ thông tin người dùng: Thông tin người dùng như tên, email, và hình đại diện có thể được lưu trữ trong một collection người dùng.
* Lưu trữ nhóm chat: Thông tin về các nhóm chat, bao gồm thành viên và tin nhắn trong nhóm, có thể được lưu trữ trong một collection riêng.

1. **Amazon DynamoDB**



Hình 2 - 4 Amazon DynamoDB

+Giới thiệu về Amazon DynamoDB: Amazon DynamoDB là một dịch vụ cơ sở dữ liệu NoSQL quản lý được, được cung cấp bởi Amazon Web Services (AWS). Nó được thiết kế để cung cấp hiệu suất cao và tính sẵn sàng cao cho các ứng dụng web và di động.

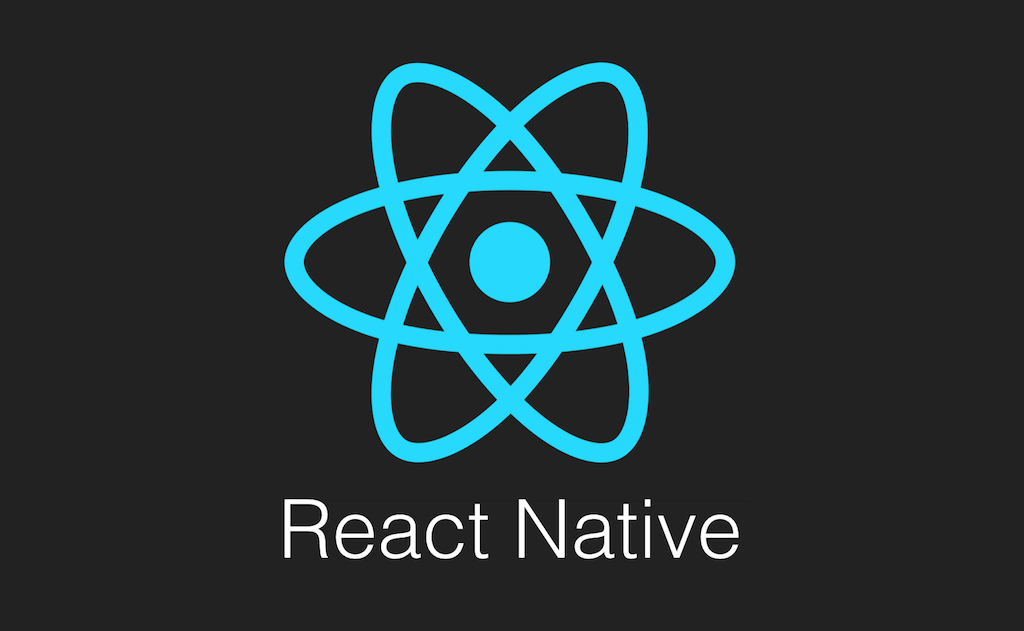
+Lợi ích của Amazon DynamoDB:

* Hiệu suất cao: DynamoDB cung cấp hiệu suất cao và có khả năng mở rộng tự động để xử lý hàng triệu yêu cầu mỗi giây.
* Tính sẵn sàng cao: DynamoDB được xây dựng để đảm bảo tính sẵn sàng cao và không có thời gian ngừng hoạt động dự kiến.
* Dễ sử dụng: DynamoDB cung cấp một giao diện đơn giản và dễ sử dụng, bao gồm cả các SDK (Software Development Kit) cho nhiều ngôn ngữ lập trình.

+Sử dụng Amazon DynamoDB trong Ứng dụng Chat: DynamoDB có thể được sử dụng để lưu trữ dữ liệu tin nhắn, thông tin người dùng và dữ liệu nhóm chat trong ứng dụng Chat của bạn. Dưới đây là một số ví dụ về cách sử dụng DynamoDB:

* Lưu trữ tin nhắn: Mỗi tin nhắn có thể được lưu trữ dưới dạng một mục trong một bảng DynamoDB.
* Lưu trữ thông tin người dùng: Thông tin người dùng như tên, email và hình đại diện có thể được lưu trữ trong một bảng DynamoDB.
* Lưu trữ nhóm chat: Thông tin về các nhóm chat, bao gồm thành viên và tin nhắn trong nhóm, có thể được lưu trữ trong một bảng DynamoDB.

## React native



Hình 2 - 5 Logo React Native

React Native là một framework phát triển ứng dụng di động được phát triển bởi Facebook. Nó cho phép các nhà phát triển sử dụng JavaScript và React để xây dựng các ứng dụng di động cho cả iOS và Android từ một mã nguồn duy nhất. Dưới đây là một số ưu và nhược điểm của React Native:

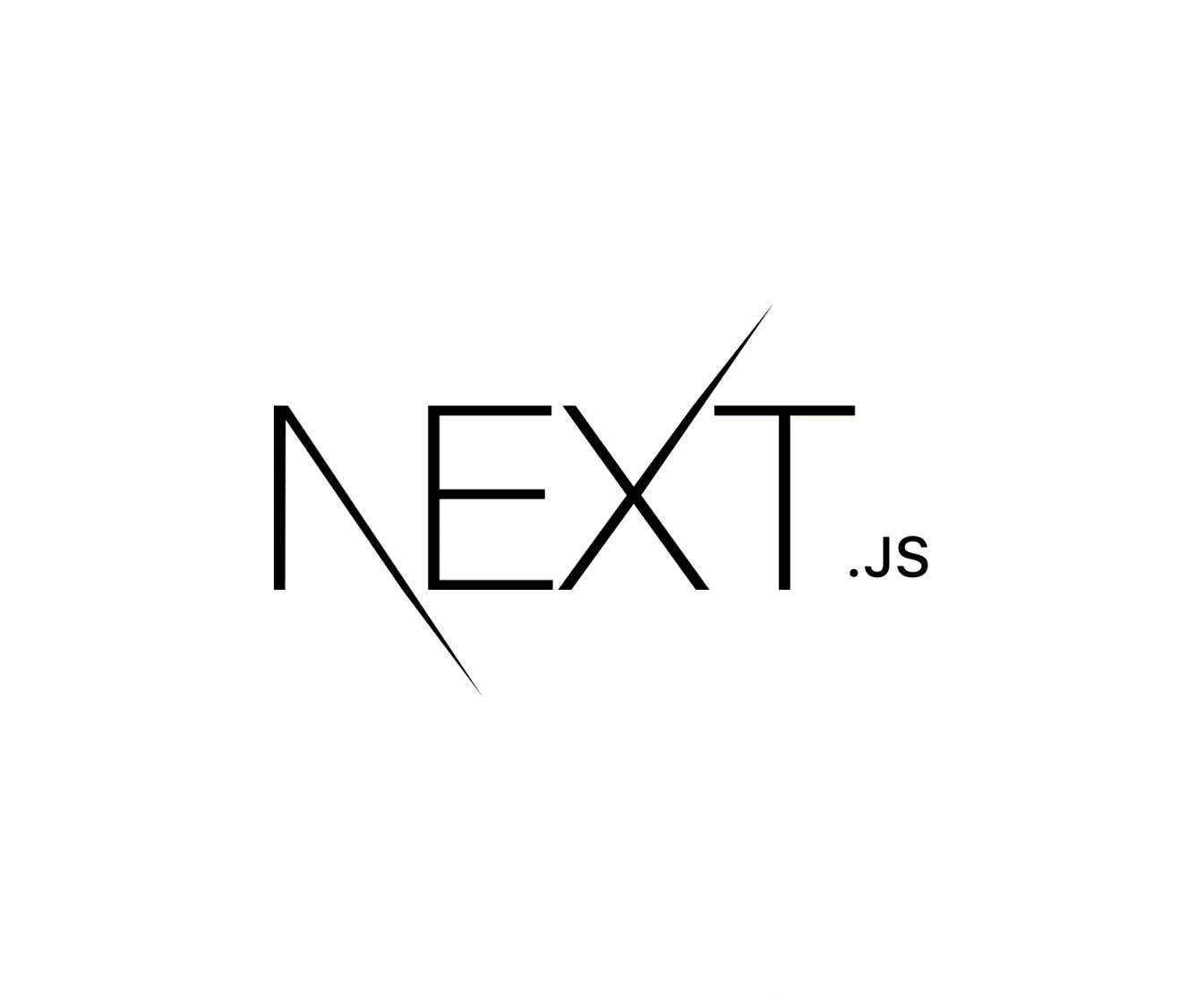
- Ưu điểm:

* Cross-platform Development: React Native cho phép bạn xây dựng ứng dụng cho cả iOS và Android từ một mã nguồn duy nhất, giúp tiết kiệm thời gian và công sức so với việc phát triển từng platform một.
* Hiệu suất tốt: Với việc sử dụng JavaScript, React Native sử dụng mã nguồn hiệu quả và cho phép các ứng dụng chạy mượt mà trên cả hai nền tảng.
* Cộng đồng lớn: React Native có một cộng đồng phát triển mạnh mẽ, điều này có nghĩa là có nhiều tài liệu, hướng dẫn và thư viện sẵn có để giúp bạn giải quyết các vấn đề và phát triển ứng dụng một cách hiệu quả.
* Hot Reloading: React Native hỗ trợ hot reloading, cho phép bạn xem kết quả ngay lập tức khi thay đổi mã nguồn mà không cần tải lại ứng dụng.

- Nhược điểm:

* Hiệu suất có thể giảm: Mặc dù hiệu suất của React Native khá tốt, nhưng trong một số trường hợp, đặc biệt là với các ứng dụng đòi hỏi hiệu suất cao, có thể cần phải viết các modules native để tối ưu hóa hiệu suất.
* Thư viện bên thứ ba: Mặc dù có một số thư viện và modules sẵn có cho React Native, nhưng không phải tất cả chúng đều được hỗ trợ tốt và có thể gây ra các vấn đề khi tích hợp vào dự án.
* Khó khăn trong việc debugging: Debugging trong React Native có thể phức tạp hơn so với các framework khác do sự kết hợp của mã JavaScript và mã native.
* Yêu cầu kiến thức native: Đôi khi, để xử lý các yêu cầu đặc biệt hoặc tối ưu hóa hiệu suất, bạn có thể cần kiến thức về lập trình native (Objective-C/Swift cho iOS và Java/Kotlin cho Android).

## NextJS



Hình 2 - 6 Logo Next.js

Next.js là một framework React được sử dụng rộng rãi trong cộng đồng phát triển web hiện nay. Được phát triển bởi Vercel, Next.js cung cấp một cách tiếp cận hiệu quả và linh hoạt để xây dựng các ứng dụng web phức tạp và hiệu suất cao.

-Ưu điểm:

* Server-side Rendering (SSR) và Static Site Generation (SSG): Next.js hỗ trợ cả SSR và SSG, cho phép bạn tạo các ứng dụng web có hiệu suất cao, tối ưu cho SEO và thời gian tải trang nhanh hơn.
* Routing tự động: Next.js cung cấp một hệ thống routing tự động, giúp bạn tạo các ứng dụng web có cấu trúc linh hoạt và dễ bảo trì.
* Tối ưu cho SEO: Nhờ vào SSR và SSG, Next.js cho phép các trang web được render trên máy chủ trước khi được gửi đến trình duyệt, giúp cải thiện SEO và tăng khả năng tìm thấy của các trang web trên các công cụ tìm kiếm.
* Hỗ trợ TypeScript nền tảng: Next.js hỗ trợ TypeScript nền tảng, giúp tăng tính tin cậy và dễ bảo trì của mã nguồn.

-Nhược điểm:

* Khó khăn trong việc học hỏi ban đầu: Đối với những người mới bắt đầu, có thể cần một thời gian để làm quen với cách Next.js hoạt động, đặc biệt là với các khái niệm như SSR và SSG.
* Có thể phức tạp với các dự án lớn: Trong các dự án lớn, việc quản lý các trạng thái và side effects có thể trở nên phức tạp hơn so với các ứng dụng nhỏ.
* Có thể có hiệu suất kém trên máy chủ yếu: SSR và SSG có thể tạo ra các yêu cầu lớn đối với máy chủ, đặc biệt là trong các ứng dụng có lượng truy cập lớn
* Yêu cầu hiểu biết về React: Next.js là một framework dựa trên React, do đó, việc hiểu biết về React là cần thiết để làm việc hiệu quả với Next.js.

## Firebase



Hình 2 - 7 Logo Firebase

Firebase là nền tảng phát triển cho ứng dụng web và ứng dụng di động. Với Firebase, bạn có thể nhanh chóng xây dựng các ứng dụng chất lượng cao, gia tăng số người dùng hoạt động tích cực và kiếm được nhiều tiền hơn. Nền tảng này có nhiều tính năng được tích hợp chặt chẽ mà bạn có thể sử dụng kết hợp với nhau – trong đó có tính năng phân tích dữ liệu, chương trình phụ trợ ưu tiên thiết bị di động, công cụ phát triển ứng dụng và công cụ kiếm tiền – để giúp ứng dụng của bạn đạt được kết quả tốt nhất.

* + Firebase Authentication

Hầu hết các ứng dụng cần biết danh tính của người dùng. Việc biết danh tính của người dùng cho phép ứng dụng lưu dữ liệu người dùng trên đám mây một cách an toàn và cung cấp trải nghiệm được cá nhân hóa giống nhau trên tất cả các thiết bị của người dùng.

Xác thực Firebase cung cấp các dịch vụ phụ trợ, SDK dễ sử dụng và thư viện giao diện người dùng được tạo sẵn để xác thực người dùng với ứng dụng của bạn. Nó hỗ trợ xác thực bằng mật khẩu, số điện thoại, nhà cung cấp nhận dạng liên kết phổ biến như Google, Facebook và Twitter, v.v.

Xác thực Firebase tích hợp chặt chẽ với các dịch vụ Firebase khác và tận dụng các tiêu chuẩn ngành như OAuth 2.0 và OpenID Connect để có thể dễ dàng tích hợp với chương trình phụ trợ tùy chỉnh của bạn.

## Redis



Hình 2 – 8 Logo redis

Redis (REmote DIctionary Server) là một mã nguồn mở được dùng để lưu trữ dữ liệu có cấu trúc, có thể sử dụng như một database, bộ nhớ cache hay một message broker.

* + Các ứng dụng của Redis:
    - Caching: Sử dụng làm bộ nhớ đệm. Chính tốc độ đọc ghi nhanh mà Redis có thể làm bộ nhớ đệm, nơi chia sẻ dữ liệu giữa các ứng dụng hoặc làm database tạm thời. Ngoài ra Redis có thể sử dụng để làm Full Page Cache cho website. Cũng vì tính nhất quán của Redis, cho dù restart Redis thì người dùng cũng không có cảm nhận chậm khi tải trang.
    - Counter: Sử dụng làm bộ đếm. Với thuộc tính tăng giảm thông số rất nhanh trong khi dữ liệu được lưu trên RAM, sets và sorted sets được sử dụng thực hiện đếm lượt view của một website, các bảng xếp hạng trong game chẳng hạng. Redis hỗ trợ thread safe do đó nó có thể đồng bộ dữ liệu giữa các request.
    - Publish/Suscribe (Pub/Sub): Tạo kênh chia sẻ dữ liệu. Redis hỗ trợ tạo các channel để trao đổi dữ liệu giữa publisher và subscriber giống như channel trong Socket Cluster hay topic trong Apache Kafka. Ví dụ: Pub/Sub được sử dụng theo dõi các kết nối trong mạng xã hội hoặc các hệ thống chat.
    - Queues: Tạo hàng đợi để xử lý lần lượt các request. Redis cho phép lưu trữ theo list và cung cấp rất nhiều thao tác với các phần tử trong list, vì vậy nó còn được sử dụng như một message queue.

## Socket.io



Socket.IO là một thư viện mạnh mẽ và phổ biến dành cho các ứng dụng web thời gian thực. Nó cung cấp khả năng truyền thông tin hai chiều giữa máy chủ và máy khách theo thời gian thực. Dưới đây là tổng quan về Socket.IO cũng như những ưu điểm nổi bật của nó:

**Tổng quan về Socket.IO:**

* Giao tiếp hai chiều: Socket.IO cho phép máy chủ và máy khách giao tiếp với nhau theo hai chiều, nghĩa là cả hai đều có thể gửi và nhận dữ liệu bất cứ lúc nào.
* Dựa trên WebSockets: Socket.IO sử dụng WebSockets để truyền dữ liệu, nhưng cũng có thể tự động chuyển đổi sang các cơ chế khác như HTTP long-polling nếu WebSockets không khả dụng.
* Hỗ trợ nhiều nền tảng: Socket.IO có thể được sử dụng trên nhiều nền tảng khác nhau bao gồm Node.js, các trình duyệt web, và các thiết bị di động.

Thư viện đơn giản và dễ sử dụng: API của Socket.IO được thiết kế đơn giản và dễ hiểu, giúp cho việc phát triển ứng dụng thời gian thực trở nên dễ dàng hơn.

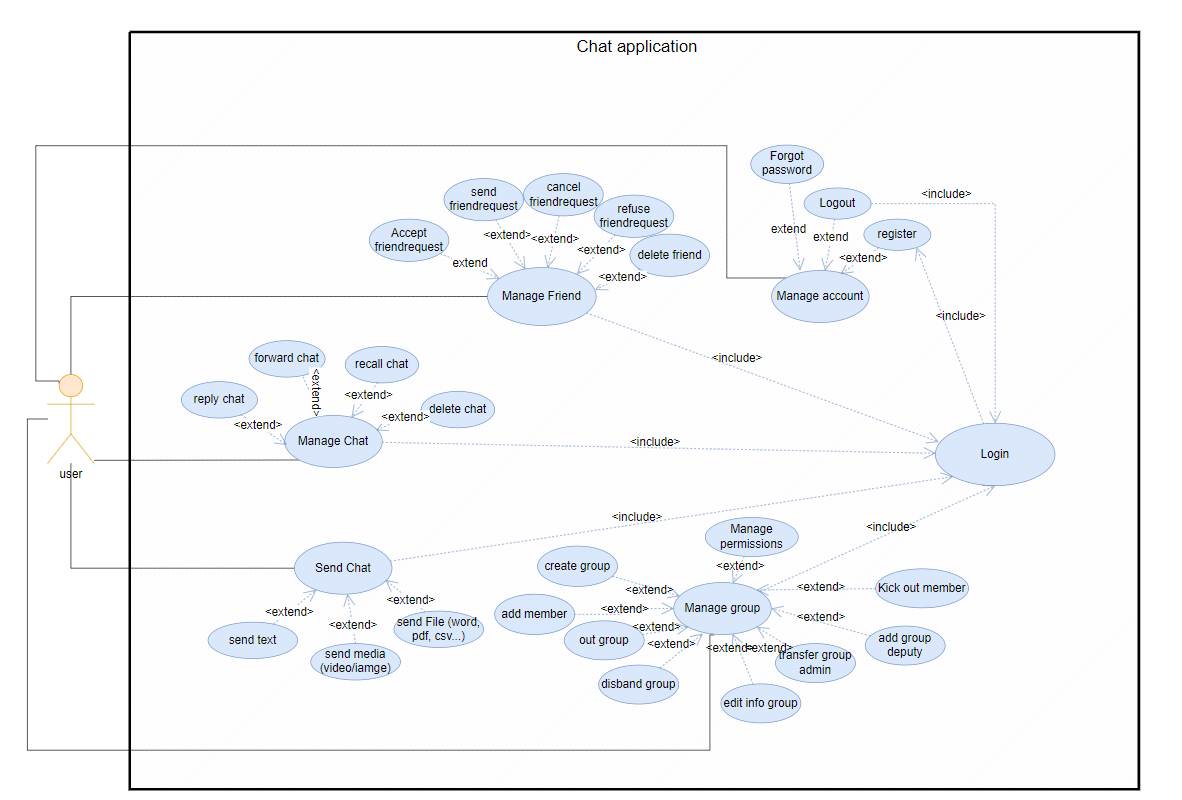
**Ưu điểm của Socket.IO:**

* Đồng bộ dữ liệu theo thời gian thực: Socket.IO cho phép đồng bộ dữ liệu giữa máy chủ và máy khách ngay lập tức khi có thay đổi, điều này rất hữu ích trong các ứng dụng như chat, trò chơi trực tuyến, và các ứng dụng cộng tác thời gian thực.
* Khả năng xử lý nhiều kết nối cùng lúc: Socket.IO có khả năng quản lý nhiều kết nối đồng thời một cách hiệu quả, điều này rất quan trọng đối với các ứng dụng có nhiều người dùng truy cập cùng lúc.
* Khả năng chuyển đổi tự động: Socket.IO tự động chuyển đổi giữa WebSockets và các cơ chế khác như HTTP long-polling, giúp đảm bảo kết nối ổn định ngay cả khi một số công nghệ không khả dụng.
* Quản lý sự kiện dễ dàng: Socket.IO cung cấp mô hình quản lý sự kiện tương tự như Node.js, giúp cho việc xử lý các sự kiện (như kết nối, ngắt kết nối, và các sự kiện tùy chỉnh) trở nên dễ dàng và trực quan.
* Bảo mật tốt: Socket.IO hỗ trợ SSL/TLS và có thể tích hợp với các cơ chế xác thực và bảo mật khác, giúp đảm bảo an toàn cho dữ liệu truyền tải.
* Khả năng mở rộng: Socket.IO có thể được mở rộng dễ dàng thông qua các tính năng như phòng (rooms) và không gian tên (namespaces), giúp quản lý và tổ chức các kết nối một cách hiệu quả.

# : PHÂN TÍCH VÀ THIẾT KẾ

## Phân tích yêu cầu bằng UML

### Usecase tổng quát



Hình 3 - 1 Usecase tổng quát

### Danh sách tác nhân và mô tả

Danh sách tác nhân và mô tả xem ở Bảng 3-1.

|  |  |
| --- | --- |
| Tác nhân | Mô tả tác nhân |
| User |  |
| Member |  |
| Admin |  |
|  |  |
|  |  |
| … |  |

Bảng 3‑1 Danh sách tác nhân và mô tả

### Danh sách các tình huống hoạt động (Use cases)

|  |  |
| --- | --- |
| ID | Tên Use case |
| UC01 | Đăng ký |
| UC02 | Đăng nhập |
| UC03 | Lấy lại mật khẩu |
| UC04 | Gửi tin nhắn |
| UC05 | Tạo nhóm chat |
| UC06 | Gửi file |
| UC07 | Chỉnh sửa thông tin cá nhân |
| UC08 | Gửi lời mời kết bạn |
| UC09 | Chấp nhận lời mời kết bạn |
| UC10 | Từ chối lời mời kết bạn |
| UC11 | Xoá bạn bè |
| UC12 | Chuyển tiếp tin nhắn |
| UC13 | Thu hồi tin nhắn |
| UC14 | Xoá tin nhắn phía bạn |
| UC15 | Tạo nhóm chat |
| UC16 | Thêm thành viên vào nhóm chat |
| UC17 | Xoá thành viên khỏi nhóm chat |
| UC18 | Thêm phó nhóm |
| UC19 | Rời khỏi nhóm |
| UC20 | Giải tán nhóm |
| UC21 | Chuyền quyền admin |
| UC22 | Chỉnh sửa thông tin nhóm |

Bảng 3‑2 Danh sách các tình huống trong hệ thống

### Tình huống hoạt động

Đặc tả use case.

Activity diagram.

Sequence diagram cho từng tình huống.

## Đặc tả các yêu cầu chức năng

### Usecase đăng kí

|  |  |
| --- | --- |
| * **Tên use case:** Đăng ký | |
| * **Mô tả sơ lược:** Usecase giúp người dùng đăng ký tài khoản để sử dụng ứng dụng zola | |
| * **Actor chính:** Người dùng | |
| * **Actor phụ:** Không | |
| * **Tiền điều kiện:** Đang ở trang đăng ký, đăng nhập | |
| * **Hậu điều kiện:** Đăng ký thành công sẽ tự động đăng nhập và đưa người dùng đến giao diện chat | |
| * **Luồng sự kiện chính (Main Flow):** | |
| **Actor** | **System** |
| 1. Vào trang đăng ký |  |
| 2. Nhập tên muốn đăng ký |  |
| 3. Chọn tiếp tục | 4. Hệ thống kiểm tra tên |
|  | 5. Hệ thống chuyển sang giao diện nhập số điện thoại và mật khẩu |
| 6. Nhập số điện thoại và mật khẩu |  |
| 7. Chọn tiếp tục | 8. Hệ thống kiểm tra số điện thoại và mật khẩu |
|  | 9. Hệ thống chuyển sang giao diện xác thực otp |
| 10. Nhập recpacha | 11. Hệ thống gửi mã otp về số điện thoại đã nhập |
| 12. Nhập mã otp | 13. Hệ thống kiểm tra mã otp |
|  | 14. Chuyển sang giao diện chat |
| * **Luồng sự kiện thay thế (Alternate Flow):** | |
|  | 4.1. Hệ thống thông báo name nhập sai cấu trúc |
| 4.2. Người dùng quay lại bước 2 |  |
|  | 8.1 Hệ thống thông báo nhập sai cấu trúc số điện thoại, hoặc mật khẩu |
| 8.2 Người dùng quay lại bước 6 |  |
|  | 13.1 Hệ thống thông báo nhập sai otp |
| 13.2 Người dùng quay lại bước 12 |  |

Activity

A diagram of a flowchart

Description automatically generated

Sequence

A diagram of a diagram

Description automatically generated

### Usecase đăng nhập

1. Đặc tả

|  |  |
| --- | --- |
| * **Tên use case:** Đăng nhập | |
| * **Mô tả sơ lược:** Uc giúp người dùng đăng nhập vào hệ thống zola | |
| * **Actor chính:** Người dùng | |
| * **Actor phụ:** Không | |
| * **Tiền điều kiện:** Đang ở trang yêu cầu đăng nhập hoặc đăng ký | |
| * **Hậu điều kiện:** Đăng nhập thành công sẽ vào hệ thống zola | |
| * **Luồng sự kiện chính (Main Flow):** | |
| **Actor** | **System** |
| 1. Vào trang đăng nhập |  |
| 2. Nhập thông số điện thoại và mật khẩu |  |
| 3. Click nút --> | 4. Hệ thống kiểm tra thông tin đăng nhập |
|  | 5. Hệ thống chuyển sang giao diện home của hệ thống zola |
| * **Luồng sự kiện thay thế (Alternate Flow):** | |
|  | 4.1. Hệ thống báo số điện thoại hoặc mật khẩu không chính xác |
| 4.2. Người dùng quay lại bước 2 |  |

*b. Biểu đồ* Activity

A diagram with blue text

Description automatically generated

*c. Biểu đồ Sequence Diagram*

A blue and black diagram

Description automatically generated

### Usecase lấy lại mật khẩu

a. Đặt tả:

|  |  |
| --- | --- |
| * **Tên use case:** Lấy lại mật khẩu | |
| * **Mô tả sơ lược:** Uc giúp người dùng lấy lại mật khẩu khi quên mật khẩu bằng số điện thoại | |
| * **Actor chính:** Người dùng | |
| * **Actor phụ:** Không | |
| * **Tiền điều kiện:** Đang ở trang đăng nhập | |
| * **Hậu điều kiện:** Lấy lại mật khẩu thành công hệ thống sẽ tự động đăng nhập vào hệ thống với thông tin mới | |
| * **Luồng sự kiện chính (Main Flow):** | |
| **Actor** | **System** |
| 1. Vào trang lấy lại mật khẩu |  |
| 2. Nhập số điện thoại, mật khẩu mới, xác nhận mật khẩu mới |  |
| 3. Click nút tiếp tục | 4. Hệ thống kiểm tra thông tin nhập |
|  | 5. Hệ thống chuyển sang giao diện nhập mã otp |
| 6. Nhập recapcha | 7. Hệ thống gửi mã otp |
| 8. Nhập mã otp |  |
| 9. nhấn nút tiếp tục | 10. Hệ thống kiểm tra mã otp |
|  | 11. Hệ thống chuyển sang giao diện home của zola |
| * **Luồng sự kiện thay thế (Alternate Flow):** | |
|  | 4.1. Hệ thống thông báo số điên thoại nhập sai hoặc sai mật khẩu hoặc mật khẩu không khớp |
| 4.2. Người dùng quay lại bước 2 |  |
|  | 10.1 Hệ thống thông báo nhập sai otp |
| 10.2 Người dùng quay lại bước 8 |  |

b. Biểu đồ Activity

A diagram of a flowchart

Description automatically generated

*c. Biểu đồ Sequence Diagram*

A computer screen shot of a diagram

Description automatically generated

### Usecase gửi tin nhắn

a. Đặc tả:

|  |  |
| --- | --- |
| * **Tên use case:** Gửi tin nhắn | |
| * **Mô tả sơ lược:** Người dùng có thể gửi tin nhắn văn bản | |
| * **Actor chính:** Người dùng | |
| * **Actor phụ:** Không | |
| * **Tiền điều kiện:** Đăng nhập thành công và khởi tạo đoạn chat | |
| * **Hậu điều kiện:** Gửi được tin nhắn để hiển thị bên bản thân và những người khác | |
| * **Luồng sự kiện chính (Main Flow):** | |
| **Actor** | **System** |
| 1. Nhập tin nhắn văn bản |  |
| 2. Nhấn nút gửi | 3. Hệ thống kiểm tra dữ liệu rỗng |
|  | 4. Hệ thống chạy socket và hiển thị tin nhắn ra khung chat, đồng thời lưu vào csdl. |
| * **Luồng sự kiện thay thế (Alternate Flow):** | |
|  | 3.1 Tin nhắn rỗng, hệ thống không lưu |
|  | 3.1.2. Usecase kết thúc |

b. Biểu đồ Activity

A screenshot of a computer

Description automatically generated

c. *Biểu đồ Sequence Diagram*

A diagram of a diagram

Description automatically generated

### Usecase tạo nhóm chat

a. Đặc tả:

|  |  |
| --- | --- |
| * **Tên use case:** Tạo nhóm chat | |
| * **Mô tả sơ lược:** Chức năng giúp người dùng tạo nhóm chat | |
| * **Actor chính:** Người dùng | |
| * **Actor phụ:** Không | |
| * **Tiền điều kiện:** Đăng nhập thành công | |
| * **Hậu điều kiện:** Người dùng và các thành viên được chọn sẽ được thêm vào nhóm chat | |
| * **Luồng sự kiện chính (Main Flow):** | |
| **Actor** | **System** |
| 1.Người dùng chọn biểu tượng tạo nhóm chat | 2.Hệ thống hiển thị giao diện model tạo nhóm chat |
| 3.Người dùng nhập tên nhóm chat  4. Người dùng chọn thành viên được thêm vào nhóm | 5.Hệ thống kiểm tra số lượng thành viên |
| 6.Người dùng chọn vào biểu tượng tạo nhóm | 7.Tạo nhóm chat thành công sẽ chuyển tới cuộc hội thoại của nhóm |
|  |  |
| * **Luồng sự kiện thay thế (Alternate Flow):** | |
|  | 5.1. Không đủ thành viên tối thiểu để tạo nhóm biểu tượng tạo nhóm sẽ không sáng. |
|  | 5.1.2 Hệ thống quay lại bước 4. |

b. Biểu đồ Activity

A screenshot of a diagram

Description automatically generated

*c. Biểu đồ Sequence Diagram*

A diagram of a project

Description automatically generated

### Usecase gửi file

a. Đặc tả:

|  |  |
| --- | --- |
| * **Tên use case:** Gửi file(word,pdf,excel,…) | |
| * **Mô tả sơ lược:** Chức năng giúp người dùng gửi được file | |
| * **Actor chính:** Người dùng | |
| * **Actor phụ:** Không | |
| * **Tiền điều kiện:** Đăng nhập thành công và đã thiết lập kết nối trước khi chat | |
| * **Hậu điều kiện:** Người dùng sẽ gửi được file. | |
| * **Luồng sự kiện chính (Main Flow):** | |
| **Actor** | **System** |
| 1.Người dùng chọn biểu tượng gửi file | 2.Hệ thống hiển thị giao diện chọn file |
| 3.Người dùng chọn file cần gửi | 4.Hệ thống kiểm tra dung lượng của file |
|  | 5.File được chọn sẽ hiển thị vào đoạn chat |
|  |  |
| * **Luồng sự kiện thay thế (Alternate Flow):** | |
|  | 4.1. Dung lượng file >20MB hệ thống sẽ hiển thị thông báo file quá lớn! |
|  | 4.1.2 Hệ thống kết thúc. |
|  |  |

b. Biểu đồ Activity

A screenshot of a computer

Description automatically generated

*c. Biểu đồ Sequence Diagram*

A diagram of a project

Description automatically generated

### Usecase chỉnh sửa thông tin cá nhân

a. Đặc tả:

|  |  |
| --- | --- |
| * **Tên use case:** Chỉnh sửa thông tin cá nhân | |
| * **Mô tả sơ lược:** Chức năng dùng để chỉnh | |
| * **Actor chính:** Người dùng | |
| * **Actor phụ:** Không | |
| * **Tiền điều kiện:** Đăng nhập thành công và ở trang thông tin cá nhân | |
| * **Hậu điều kiện:** Người dùng sẽ chỉnh sửa được thông tin cá nhân bao gồm ảnh đại diện, tên, giới tính, ngày sinh. | |
| * **Luồng sự kiện chính (Main Flow):** | |
| **Actor** | **System** |
| 1.Người dùng nhấn vào nút chỉnh sửa thông tin cá nhân | 2.Hệ thống hiển thị giao diện chỉnh sửa thông tin cá nhân |
| 3.Người dùng nhập những thông tin cần chỉnh sửa |  |
| 4.Người dùng nhấn nút cập nhật thông tin | 5.Hệ thống thông báo cập nhật thông tin thành công |
|  | 6. Hệ thống quay lại trang thông tin cá nhân |
|  |  |
| * **Luồng sự kiện thay thế (Alternate Flow):** | |

b. Biểu đồ Activity

A screenshot of a computer

Description automatically generated

*c. Biểu đồ Sequence Diagram*

A diagram of a diagram

Description automatically generated

### Usecase gửi lời mời kết bạn

a. Đặc tả:

|  |  |
| --- | --- |
| * **Tên use case:** Gửi lời mời kết bạn | |
| * **Mô tả sơ lược:** Chức năng giúp người dùng gửi lời mời kết bạn cho 1 người khác | |
| * **Actor chính:** Người dùng | |
| * **Actor phụ:** Không | |
| * **Tiền điều kiện:** Đăng nhập thành công | |
| * **Hậu điều kiện:** Người dùng sẽ gửi lời mời kết bạn cho người khác. | |
| * **Luồng sự kiện chính (Main Flow):** | |
| **Actor** | **System** |
| 1.Người dùng nhấn vào thanh tìm kiếm | 2.Hệ thống hiển thị giao diện tìm kiếm |
| 3.Người dùng nhập số điện thoại người cần gửi | 4.Hệ thống hiển thị thông tin người cần gửi lời mời ( ảnh đại diện, tên) |
| 5. Người dùng nhấn vào nút gửi lời mời kết bạn | 6. Hệ thống chuyển đổi nút kết bạn thành nút thu hồi |
|  |  |
| * **Luồng sự kiện thay thế (Alternate Flow):** | |

b. Biểu đồ Activity

A diagram with blue text

Description automatically generated

*c. Biểu đồ Sequence Diagram*

A diagram of a diagram

Description automatically generated

### Usecase chấp nhận lời mời kết bạn

a. Đặc tả:

|  |  |
| --- | --- |
| * **Tên use case:** Chấp nhận lời mời kết bạn | |
| * **Mô tả sơ lược:** Chức năng giúp người dùng trở thành bạn bè với người gửi kết bạn với mình | |
| * **Actor chính:** Người dùng | |
| * **Actor phụ:** Không | |
| * **Tiền điều kiện:** Đăng nhập thành công và đã nhận được 1 lời mời kết bạn từ ai đó, ở trang danh bạ | |
| * **Hậu điều kiện:** Người dùng sẽ trở thành bạn bè với người gửi lời mời kết bạn. | |
| * **Luồng sự kiện chính (Main Flow):** | |
| **Actor** | **System** |
| 1.Người dùng chọn nút lời mời kết bạn | 2.Hệ thống hiển thị giao diện lời mời kết bạn |
| 3.Người dùng chọn nút đồng ý | 4.Hệ thống thông báo kết bạn thành công |
|  |  |
| * **Luồng sự kiện thay thế (Alternate Flow):** | |
|  |  |

b. Biểu đồ Activity

A screenshot of a diagram

Description automatically generated

*c. Biểu đồ Sequence Diagram*

A diagram of a diagram

Description automatically generated

### Usecase từ chối lời mời kết bạn

a. Đặc tả:

|  |  |
| --- | --- |
| * **Tên use case:** Từ chối lời mời kết bạn | |
| * **Mô tả sơ lược:** Chức năng giúp người dùng từ chối lời mời kết bạn của người khác gửi tới | |
| * **Actor chính:** Người dùng | |
| * **Actor phụ:** Không | |
| * **Tiền điều kiện:** Đăng nhập thành công và nhận được 1 lời mời kết bạn, đang ở trang danh bạ | |
| * **Hậu điều kiện:** Người dùng sẽ từ chối được lời mời kết bạn của người khác | |
| * **Luồng sự kiện chính (Main Flow):** | |
| **Actor** | **System** |
| 1.Người dùng chọn nút lời mời kết bạn | 2.Hệ thống hiển thị giao diện lời mời kết bạn |
| 3.Người dùng chọn nút từ chối | 4.Hệ thống xóa lời mời kết bạn khỏi danh sách |
|  |  |
| * **Luồng sự kiện thay thế (Alternate Flow):** | |
|  |  |

b. Biểu đồ Activity

A screenshot of a diagram

Description automatically generated

*c. Biểu đồ Sequence Diagram*

A diagram of a diagram

Description automatically generated

### Usecase xóa bạn bè

a. Đặc tả:

|  |  |
| --- | --- |
| * **Tên use case:** Xóa bạn bè | |
| * **Mô tả sơ lược:** Chức năng giúp người dùng xóa được bạn bè | |
| * **Actor chính:** Người dùng | |
| * **Actor phụ:** Không | |
| * **Tiền điều kiện:** Đăng nhập thành công và đã có bạn bè trước đó | |
| * **Hậu điều kiện:** Người dùng sẽ xóa được 1 bạn bè khỏi danh sách bạn bè. | |
| * **Luồng sự kiện chính (Main Flow):** | |
| **Actor** | **System** |
|  |  |
| 1.Người dùng nhấn vào nút danh bạ | 2.Hệ thống hiển thị giao diện danh bạ |
| 3.Người dùng nhấn giữ người bạn cần xóa bạn bè | 4.Hệ thống hiển thị thông báo bạn có chắc chắn muốn xóa bạn |
| 5.Người dùng chọn đồng ý | 6.Hệ thống xóa người đó khỏi danh sách bạn bè |
|  |  |
| * **Luồng sự kiện thay thế (Alternate Flow):** | |
| 5.1Người dùng chọn từ chối | 5.2.Hệ thống quay lại bước 2 |
|  |  |

b. Biểu đồ Activity

A diagram of a diagram

Description automatically generated

*c. Biểu đồ Sequence Diagram*

A diagram of a diagram

Description automatically generated

### Usecase chuyển tiếp tin nhắn

1. Đặc tả:

|  |  |
| --- | --- |
| * **Tên use case:** Chuyển tiếp tin nhắn | |
| * **Mô tả sơ lược:** Chức năng giúp người dùng chuyển tiếp được tin nhắn | |
| * **Actor chính:** Người dùng | |
| * **Actor phụ:** Không | |
| * **Tiền điều kiện:** Đăng nhập thành công và đã thiết lập kết nối trước khi chat | |
| * **Hậu điều kiện:** Người dùng sẽ chuyển tiếp được tin nhắn. | |
| * **Luồng sự kiện chính (Main Flow):** | |
| **Actor** | **System** |
| 1.Người dùng chọn vào biểu tượng chuyển tiếp tin nhắn. | 2.Hệ thống hiển thị giao diện chọn danh sách bạn bè cần chuyển tiếp tin nhắn. |
| 3.Người dùng chọn người cần chuyển tiếp tin nhắn. |  |
| 4.Người dùng chọn gửi. |  |
|  | 5.Hệ thống thông báo chuyển tiếp tin nhắn thành công. |
| * **Luồng sự kiện thay thế (Alternate Flow):** | |
| 3.1.Không có người nào để chuyển tiếp tin nhắn. | 3.2.Hệ thống kết thúc. |

b. Biểu đồ Activity

A diagram of a diagram

Description automatically generated

*c. Biểu đồ Sequence Diagram*

A diagram of a diagram

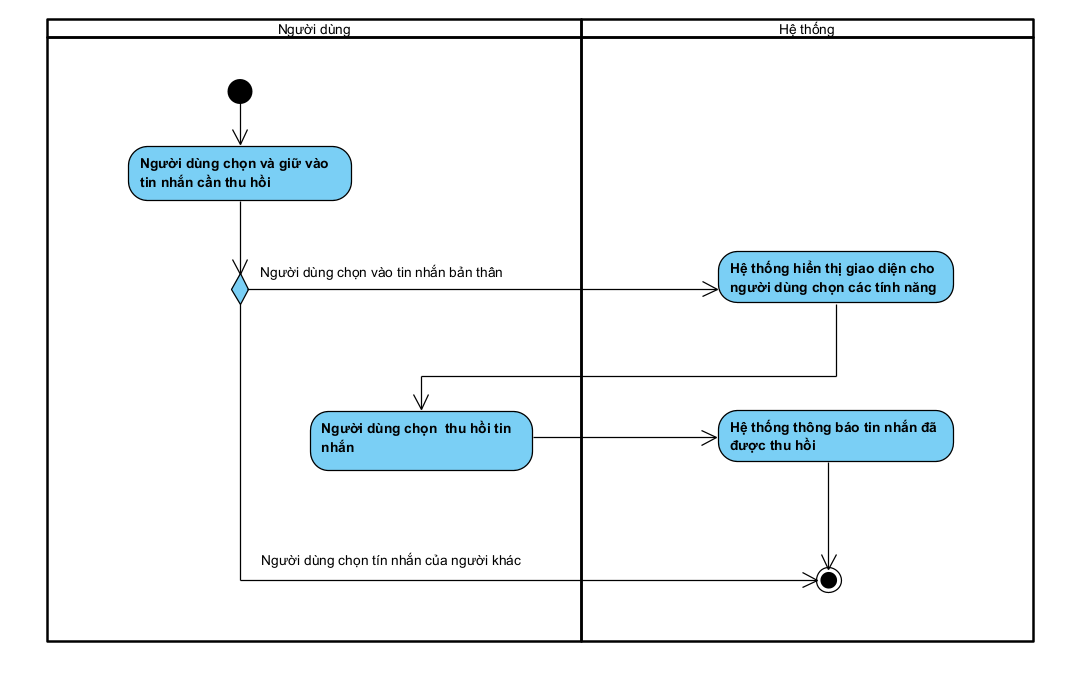
Description automatically generated

### Usecase thu hồi tin nhắn

a. Đặc tả:

|  |  |
| --- | --- |
| * **Tên use case:** Thu hồi tin nhắn | |
| * **Mô tả sơ lược:** Chức năng giúp người dùng thu hồi được tin nhắn | |
| * **Actor chính:** Người dùng | |
| * **Actor phụ:** Không | |
| * **Tiền điều kiện:** Đăng nhập thành công và đã thiết lập kết nối trước khi chat | |
| * **Hậu điều kiện:** Người dùng sẽ thu hồi được tin nhắn. | |
| * **Luồng sự kiện chính (Main Flow):** | |
| **Actor** | **System** |
| 1.Người dùng chọn và giữ vào tin nhắn cần thu hồi. | 2.Hệ thống hiển thị giao diện cho người dùng chọn các tính năng. |
| 3.Người dùng chọn thu hồi tin nhắn. | 4.Hệ thống thông báo tin nhắn đã được thu hồi. |
| * **Luồng sự kiện thay thế (Alternate Flow):** | |
| 1.1.Nếu người dùng chọn vào tin nhắn của người khác. | 1.2.Hệ thống hiển thị giao diện cho người dùng chọn các tính năng, nhưng không có tính năng thu hồi. |
|  | 1.3.Hệ thống kết thúc. |

b. Biểu đồ Activity



*c. Biểu đồ Sequence Diagram*

A diagram of a project

Description automatically generated

### Usecase xóa tin nhắn phía bạn

a. Đặc tả:

|  |  |
| --- | --- |
| * **Tên use case:** Xóa tin nhắn phía bạn | |
| * **Mô tả sơ lược:** Chức năng giúp người dùng xóa được tin nhắn của mình gửi, nhưng người khác vẫn thấy | |
| * **Actor chính:** Người dùng | |
| * **Actor phụ:** Không | |
| * **Tiền điều kiện:** Đăng nhập thành công và đã thiết lập kết nối trước khi chat | |
| * **Hậu điều kiện:** Người dùng sẽ xóa được tin nhắn của mình gửi. | |
| * **Luồng sự kiện chính (Main Flow):** | |
| **Actor** | **System** |
| 1.Người dùng chọn và giữ vào tin nhắn cần xóa ở phía mình. | 2.Hệ thống hiển thị giao diện gồm nhiều chức năng. |
| 3.Người dùng chọn vào xóa tin nhắn. | 4.Hệ thống hiển thị giao diện xác nhận xóa tin nhắn. |
| 5.Người dùng chọn xác nhận. | 6.Hệ thống thông báo tin nhắn đã được xóa. |
| * **Luồng sự kiện thay thế (Alternate Flow):** | |
| 5.1.Người dùng chọn hủy. | 5.2.Hệ thống kết thúc. |

b. Biểu đồ Activity

A diagram with blue text

Description automatically generated

*c. Biểu đồ Sequence Diagram*

A diagram of a diagram

Description automatically generated

### Usecase tạo nhóm chat

a. Đặc tả:

|  |  |
| --- | --- |
| * **Tên use case:** Tạo nhóm chat | |
| * **Mô tả sơ lược:** Chức năng giúp người dùng tạo được nhóm chat | |
| * **Actor chính:** Người dùng | |
| * **Actor phụ:** Không | |
| * **Tiền điều kiện:** Đăng nhập thành công và đã thiết lập kết nối trước khi chat | |
| * **Hậu điều kiện:** Người dùng sẽ tạo được nhóm chat. | |
| * **Luồng sự kiện chính (Main Flow):** | |
| **Actor** | **System** |
| 1.Người dùng chọn vào biểu tượng tạo nhóm chat. | 2.Hệ thống hiển thị giao diện tạo nhóm chat. |
| 3.Người dùng chọn người cần thêm vào nhóm và đặt tên cho nhóm. |  |
| 4.Người dùng chọn tạo nhóm. |  |
|  | 5. Hệ thống thông báo tạo nhóm chat thành công. |
| * **Luồng sự kiện thay thế (Alternate Flow):** | |
| 3.1.Người dùng chọn hủy | 3.2.Hệ thống kết thúc |

b. Biểu đồ Activity

*A diagram of a chat

Description automatically generated*

*c. Biểu đồ Sequence Diagram*

A diagram of a project

Description automatically generated

### Usecase thêm thành viên vào nhóm chat

a. Đặc tả:

|  |  |
| --- | --- |
| * **Tên use case:** Thêm thành viên vào nhóm chat | |
| * **Mô tả sơ lược:** Chức năng giúp người dùng thêm được thành viên vào nhóm chat | |
| * **Actor chính:** Người dùng | |
| * **Actor phụ:** Không | |
| * **Tiền điều kiện:** Đăng nhập thành công và đã thiết lập kết nối trước khi chat, là nhóm trưởng và nhóm chat đã được tạo trước. | |
| * **Hậu điều kiện:** Người dùng sẽ thêm được thành viên vào nhóm chat. | |
| * **Luồng sự kiện chính (Main Flow):** | |
| **Actor** | **System** |
| 1.Người dùng chọn vào biểu tượng thêm thành viên vào nhóm chat. | 2.Hệ thống hiển thị danh sách bạn bè chưa có trong nhóm chat. |
| 3.Người dùng chọn người cần thêm vào nhóm chat. |  |
| 4.Người dùng chọn thêm thành viên. | 5.Hệ thống thông báo thêm thành viên vào nhóm chat thành công. |
|  |  |
| * **Luồng sự kiện thay thế (Alternate Flow):** | |
| 3.1.Người dùng chọn hủy. | 3.2.Hệ thống kết thúc. |

b. Biểu đồ Activity

*A diagram of a flowchart

Description automatically generated*

*c. Biểu đồ Sequence Diagram*

A diagram of a diagram

Description automatically generated

### Usecase xóa thành viên khỏi nhóm chat

a. Đặc tả:

|  |  |
| --- | --- |
| * **Tên use case:** Xóa thành viên khỏi nhóm chat | |
| * **Mô tả sơ lược:** Chức năng giúp người dùng xóa thành viên khỏi nhóm chat. | |
| * **Actor chính:** Người dùng | |
| * **Actor phụ:** Không | |
| * **Tiền điều kiện:** Đăng nhập thành công và đã thiết lập kết nối trước khi chat, là nhóm trưởng và thành viên đã ở trong nhóm | |
| * **Hậu điều kiện:** Người dùng sẽ xóa được thành viên ra khỏi nhóm. | |
| * **Luồng sự kiện chính (Main Flow):** | |
| **Actor** | **System** |
| 1.Người dùng chọn xem danh sách thành viên nhóm. | 2.Hệ thống hiển thị danh sách thành viên có trong nhóm. |
| 3.Người dùng chọn vào thành viên muốn xóa ra khỏi nhóm. | 4.Hệ thống hiển thị danh sách các chức năng. |
| 5.Người dùng chọn xóa thành viên khỏi nhóm. | 6.Hệ thống hiển thị thông báo xác nhận xóa thành viên khỏi nhóm. |
| 7.Người dùng chọn xác nhận. | 8.Hệ thống thông báo thành viên đã bị xóa khỏi nhóm. |
|  |  |
| * **Luồng sự kiện thay thế (Alternate Flow):** | |
| 7.1.Người dùng chọn hủy. | 7.2.Hệ thống kết thúc. |
|  |  |

b. Biểu đồ Activity

*A diagram with text and blue bubbles

Description automatically generated with medium confidence*

*c. Biểu đồ Sequence Diagram*

A diagram of a diagram

Description automatically generated

### Usecase thêm phó nhóm

a. Đặc tả:

|  |  |
| --- | --- |
| * **Tên use case:** Thêm phó nhóm | |
| * **Mô tả sơ lược:** Chức năng giúp nhóm trưởng thay đổi quyền của thành viên trong nhóm từ thành viên lên nhóm phó (phó nhóm có nhiều quyền hơn). | |
| * **Actor chính:** Người dùng | |
| * **Actor phụ:** Không | |
| * **Tiền điều kiện:** Đăng nhập thành công và đã thiết lập kết nối trước khi chat. Đang nhóm trưởng của nhóm chat muốn thêm phó nhóm, thành viên ở trong nhóm. | |
| * **Hậu điều kiện:** Thành viên được chỉ được trở thành phó nhóm. Thành viên sẽ có thêm nhiều quyền hạn (phó nhóm) trong nhóm chat được cấp quyền. | |
| * **Luồng sự kiện chính (Main Flow):** | |
| **Actor** | **System** |
| 1.Người dùng chọn xem danh sách thành viên nhóm. | 2. Hệ thống hiển thị danh sách thành viên có trong nhóm. |
| 3. Người dùng chọn vào thành viên muốn trở thành phó nhóm. | 4. Hệ thống hiển thị các tuỳ chọn đối với thành viên. |
| 5.Người dùng chọn chức năng “Thêm nhóm phó”. | 6.Hệ thống thêm thành viên thành phó nhóm. |
|  | 7.Hệ thống thông báo thành viên bổ nhiệm trở thành phó nhóm. |
| * **Luồng sự kiện thay thế (Alternate Flow):** | |
| 3.1 Người dùng chọn thành viên đã là phó nhóm. | 3.2 Hệ thống hiển thị các tuỳ chọn không có chức năng thêm phó nhóm. |
|  |  |

b. Biểu đồ Activity

A screenshot of a diagram

Description automatically generated

*c. Biểu đồ Sequence Diagram*

A diagram of a diagram

Description automatically generated

### Usecase rời khỏi nhóm

a. Đặc tả:

|  |  |
| --- | --- |
| * **Tên use case:** Rời khỏi nhóm | |
| * **Mô tả sơ lược:** Chức năng giúp người rời khỏi nhóm | |
| * **Actor chính:** Người dùng | |
| * **Actor phụ:** Không | |
| * **Tiền điều kiện:** Đăng nhập thành công và đã thiết lập kết nối trước khi chat. Đang tham gia nhóm. | |
| * **Hậu điều kiện:** Người dùng rời khỏi nhóm chỉ định. | |
| * **Luồng sự kiện chính (Main Flow):** | |
| **Actor** | **System** |
| 1.Người dùng mở tuỳ chọn. | 2.Hệ thống hiển thị giao diện tuỳ chọn. |
| 3.Người dùng chọn “Rời nhóm”. | 4.Hệ thống hiển thị thông báo xác nhận rời nhóm. |
| 5.Người dùng chọn xác nhận. | 6.Hệ thống xoá bạn người dùng ra khỏi nhóm. |
|  | 7.Hệ thống hiển thị thông báo rời nhóm. |
|  |  |
| * **Luồng sự kiện thay thế (Alternate Flow):** | |
| 5.1.Người dùng chọn huỷ. | 5.2.Hệ thống kết thúc. |
|  | 6.1.Hệ thống hiển thị tuỳ chọn trưởng nhóm mới khi rời nhóm (Khi người dùng đang là trưởng nhóm). |
| 6.2.Người dùng chọn trưởng nhóm mới và xác nhận. | 6.3.Quay về bước 7. |

b. Biểu đồ Activity

A screenshot of a diagram

Description automatically generated

*c. Biểu đồ Sequence Diagram*

A diagram of a diagram

Description automatically generated

### Usecase giải tán nhóm

a. Đặc tả:

|  |  |
| --- | --- |
| * **Tên use case:** Chức năng giải tán nhóm | |
| * **Mô tả sơ lược:** Chức năng giúp người dùng giải tán nhóm, loại bỏ tất cả thành viên và xoá nhóm khỏi hệ thống. | |
| * **Actor chính:** Người dùng | |
| * **Actor phụ:** Không | |
| * **Tiền điều kiện:** Đăng nhập thành công và đã thiết lập kết nối trước khi chat. Đang là nhóm trưởng của nhóm. | |
| * **Hậu điều kiện:** Tất cả thành viên trong nhóm đều rời khỏi, nhóm chat bị xoá. | |
| * **Luồng sự kiện chính (Main Flow):** | |
| **Actor** | **System** |
| 1. Người dùng mở tuỳ chọn. | 2.Hệ thống hiển thị giao diện tuỳ chọn. |
| 3.Người dùng chọn cài đặt nhóm. | 4.Hệ thống hiển thị giao diện cài đặt nhóm. |
| 5.Người dùng chọn “Giải tán nhóm”. | 6.Hệ thống hiển thị thông báo xác nhận. |
| 7.Người dùng xác nhận. | 8.Hệ thống giải tán nhóm và hiển thị thông báo. |
| * **Luồng sự kiện thay thế (Alternate Flow):** | |
| 7.1.Người dùng chọn huỷ. | 7.2.Hệ thống kết thúc. |

b. Biểu đồ Activity

A screenshot of a diagram

Description automatically generated

*c. Biểu đồ Sequence Diagram*

A diagram of a diagram

Description automatically generated

### Usecase chuyển quyền trưởng nhóm

a. Đặc tả:

|  |  |
| --- | --- |
| * **Tên use case:** Chuyển quyền trưởng nhóm | |
| * **Mô tả sơ lược:** Chức năng giúp trưởng nhóm chuyển nhượng quyền nhóm cho một thành viên khác trong nhóm. | |
| * **Actor chính:** Người dùng | |
| * **Actor phụ:** Không | |
| * **Tiền điều kiện:** Đăng nhập thành công và đã thiết lập kết nối trước khi chat. Đang là trưởng nhóm, trong nhóm có thành viên khác. | |
| * **Hậu điều kiện:** Người dùng sẽ trở thành thành viên và thành viên được chọn trở thành trưởng nhóm. | |
| * **Luồng sự kiện chính (Main Flow):** | |
| **Actor** | **System** |
| 1.Người dùng chọn tuỳ chọn. | 2.Hệ thống hiển thị giao diện tuỳ chọn. |
| 3.Người dùng chọn “Chuyển quyền trưởng nhóm”. | 4.Hệ thống hiển thị danh sách thành viên khác. |
| 5.Người dùng chọn trưởng nhóm mới. | 6.Hệ thống hiển thị thông báo xác nhận. |
| 7.Người dùng xác nhận. | 8.Hệ thống chuyển quyển trưởng nhóm cho thành viên được chọn. |
|  | 9.Hệ thống thông báo cho bổ nhiệm mới. |
| * **Luồng sự kiện thay thế (Alternate Flow):** | |
| 7.1.Người dùng chọn huỷ. | 7.2.Hệ thống kết thúc. |

b. Biểu đồ Activity

A diagram of a chat

Description automatically generated

*c. Biểu đồ Sequence Diagram*

A diagram of a project

Description automatically generated

### Usecase chỉnh sửa thông tin nhóm

a. Đặc tả:

|  |  |
| --- | --- |
| * **Tên use case:** Gửi file(word,pdf,excel,…) | |
| * **Mô tả sơ lược:** Chức năng giúp người dùng gửi được file | |
| * **Actor chính:** Người dùng | |
| * **Actor phụ:** Không | |
| * **Tiền điều kiện:** Đăng nhập thành công và đã thiết lập kết nối trước khi chat | |
| * **Hậu điều kiện:** Người dùng sẽ gửi được file. | |
| * **Luồng sự kiện chính (Main Flow):** | |
| **Actor** | **System** |
| 1.Người dùng chọn tuỳ chọn. | 2.Hệ thống hiển thị tuỳ chọn. |
| 3.Người dùng chọn thông tin cần thay đổi. | 4.Hệ thống hiển thị form nhập thông tin tin mới. |
| 5.Người dùng nhập thông tin mới. |  |
| 6.Người dùng chọn nút “Lưu”. | 7.Hệ thống cập nhập thông tin mới và hiển thị thông báo về cập nhập. |
| * **Luồng sự kiện thay thế (Alternate Flow):** | |
|  |  |

b. Biểu đồ Activity

A screenshot of a diagram

Description automatically generated

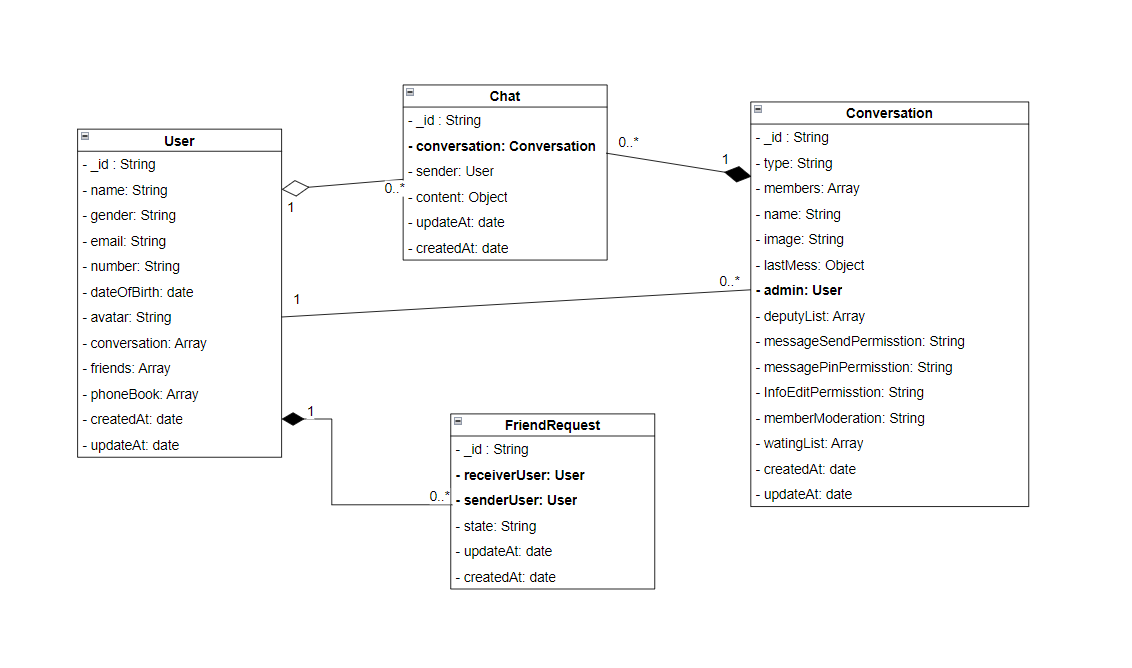
*c. Biểu đồ Sequence Diagram*

A diagram of a project

Description automatically generated

## Class diagram

Mô hình lớp (Mô hình lớp đầy đủ + Mô hình lớp bao gồm các lớp Entity)



*Hình 3.3 Class diagram*

## Deployment diagram

Mô hình triển khai hệ thống.

A computer diagram of a server

Description automatically generated

*Hình 3.4 Sơ đồ deployment*

1. Client (Web browser, mobile): Người dùng truy cập trang web từ trình duyệt web hoặc thiết bị di động. Khi truy cập, người dùng gửi yêu cầu tới máy chủ để lấy hình ảnh hoặc tệp.
2. DNS: Domain Name System (DNS) chịu trách nhiệm phân giải tên miền (ví dụ: www.mysite.com) thành địa chỉ IP của máy chủ để định tuyến yêu cầu.
3. Virtual Private Server (VPS):
   1. EC2: Một phiên bản máy chủ ảo EC2 của Amazon Web Services (AWS) được sử dụng làm máy chủ ảo.
   2. Nginx: Một máy chủ web và máy chủ proxy ngược, xử lý các yêu cầu từ khách hàng và chuyển tiếp chúng tới máy chủ web.
   3. Web server: Máy chủ web Node.js xử lý các yêu cầu từ Nginx và tương tác với các thành phần khác của hệ thống.
4. Redis: Hệ thống lưu trữ dữ liệu đệm (cache) cục bộ, được sử dụng để lưu trữ dữ liệu tạm thời nhằm tăng tốc độ truy cập. Khi máy chủ web nhận được yêu cầu, nó sẽ kiểm tra dữ liệu trong Redis trước. Nếu dữ liệu tồn tại trong cache, nó sẽ đọc từ cache. Nếu không, nó sẽ truy xuất dữ liệu từ cơ sở dữ liệu và lưu vào cache.
5. MongoDB Cloud: Một cơ sở dữ liệu NoSQL được lưu trữ trên đám mây, lưu trữ dữ liệu lâu dài và bền vững. Khi dữ liệu không tồn tại trong cache, máy chủ web sẽ truy xuất dữ liệu từ MongoDB và lưu dữ liệu này vào cache để phục vụ các yêu cầu sau.
6. S3 Storage: Dịch vụ lưu trữ đối tượng của Amazon Web Services, được sử dụng để lưu trữ các tệp hình ảnh và tệp khác. Khi người dùng tải lên tệp, tệp này sẽ được lưu trữ trong S3 và đường dẫn tệp sẽ được trả về cho người dùng.

Quá trình hoạt động chi tiết của hệ thống như sau:

1. Người dùng gửi yêu cầu truy cập hình ảnh hoặc tệp thông qua trình duyệt web hoặc thiết bị di động tới DNS để phân giải tên miền thành địa chỉ IP của máy chủ.
2. Yêu cầu được chuyển tới máy chủ Nginx thông qua giao thức HTTPS hoặc socket.
3. Nginx chuyển tiếp yêu cầu tới máy chủ web.
4. Máy chủ web kiểm tra Redis để xem dữ liệu yêu cầu có tồn tại trong cache không:
5. Nếu có, dữ liệu sẽ được đọc từ Redis và trả về cho người dùng.
6. Nếu không, máy chủ web sẽ truy xuất dữ liệu từ MongoDB, lưu trữ vào Redis, và trả dữ liệu về cho người dùng.
7. Khi người dùng tải lên tệp, máy chủ web sẽ lưu trữ tệp này vào S3 và trả về đường dẫn tệp cho người dùng.
8. Hệ thống này giúp tăng tốc độ phản hồi và hiệu quả truy cập bằng cách sử dụng caching (Redis) và cung cấp khả năng lưu trữ linh hoạt và bền vững (S3 và MongoDB).

# : HIỆN THỰC

## Cấu hình phần cứng, phần mềm

### Thông tin

Amazon EC2 (Elastic Compute Cloud): Đã sử dụng EC2 để triển khai máy chủ cho ứng dụng.

Amazon S3 (Simple Storage Service): Dùng S3 để lưu trữ dữ liệu, tệp tin đa phương tiện.

MongoDB: Sử dụng MongoDB Atlas Database để quản lý cơ sở dữ liệu

1. **Cấu hình phần cứng và phần mềm**

Thông số chi tiết EC2 và phần mềm sử dụng

|  |  |
| --- | --- |
| **Phần cứng** | **Phần mềm** |
| Hệ điều hành: Ubuntu 22.04.3 LTS  Kernel: 6.5.0-1018-aws  Architecture: x86\_64  Bộ xử lý (CPU): Intel(R) Xeon(R) CPU E5-2676 v3 @ 2.40GHz  Số CPU: 1  Số luồng/core: 1  Số core/socket: 1  Số socket: 1  Dung lượng RAM: 949 MiB  Dung lượng ổ đĩa /dev/root: 7.6G  Virtualization: Xen | Node.js: v20.11.0  Redis server: v=6.0.16 |
|  |  |

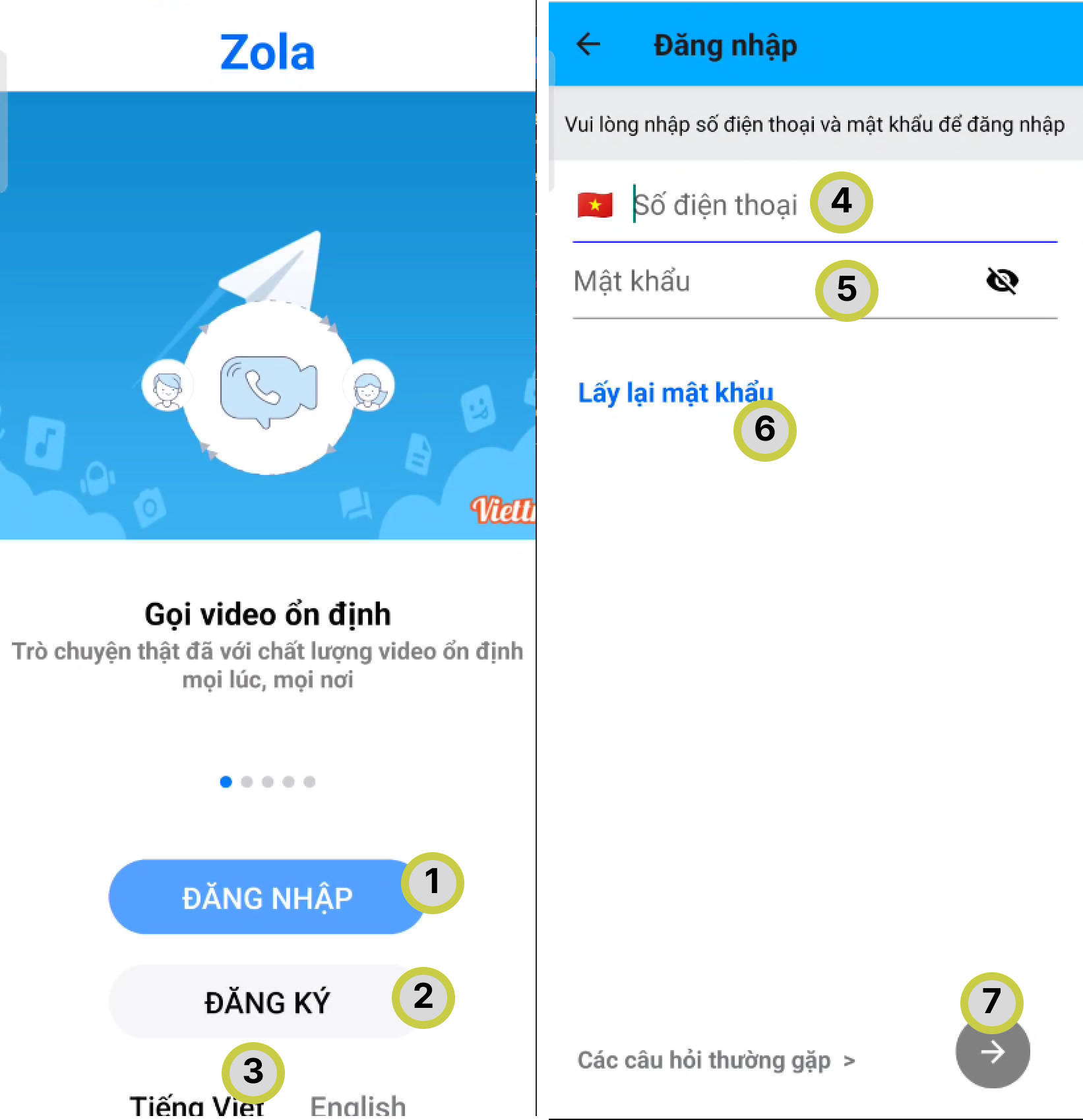
Phần mềm phát triển dự án:

Vscode, Android studio, Redis inside, Mongo atlas, Postman, Nodejs

## Giao diện của hệ thống

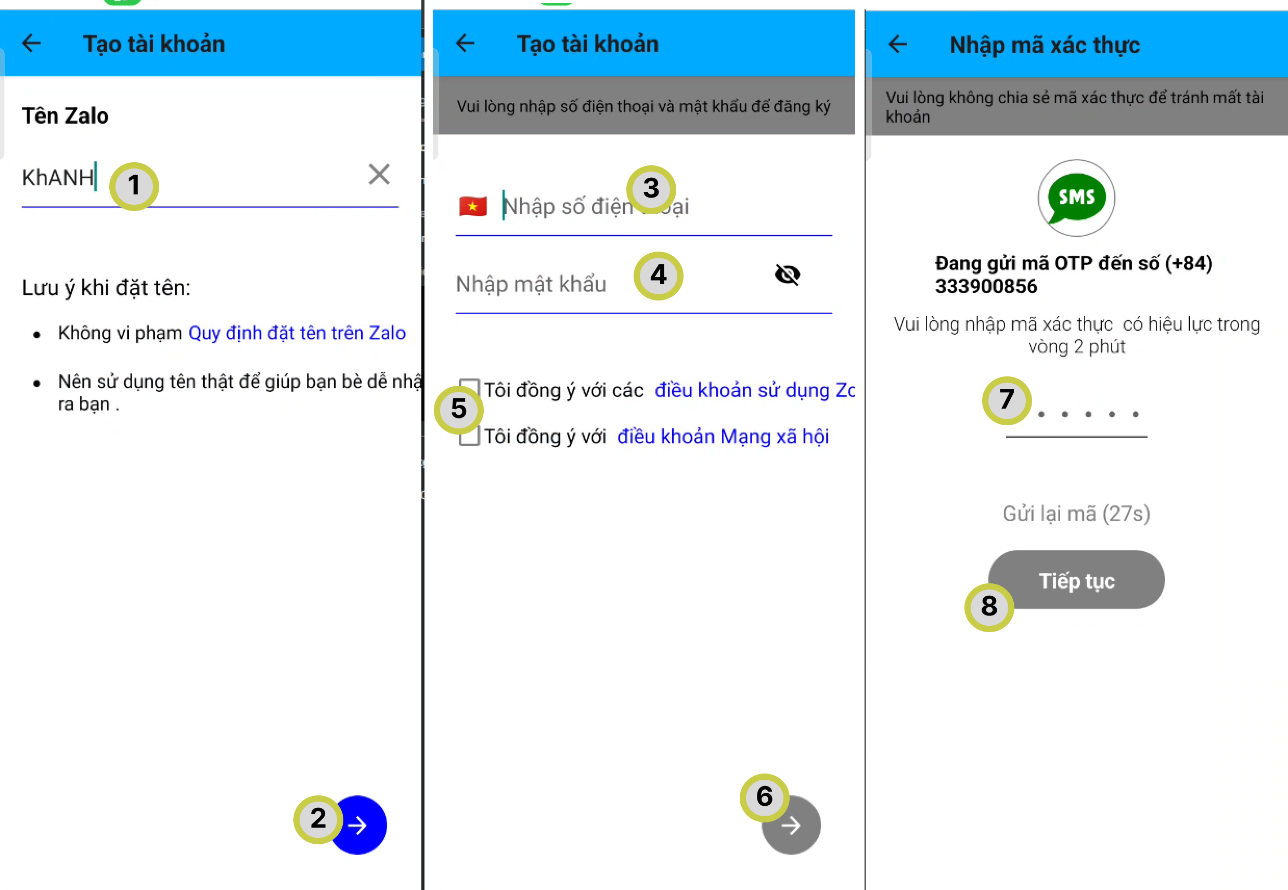
## Giao diện mobile

Màn hình đăng nhập:



* 1. button: Tuỳ chọn đăng nhập
  2. button: Tuỳ chọn đăng ký tài khoản
  3. toggle: Chuyển đổi ngôn ngữ
  4. textfield: Nhập số điện thoại
  5. textfield: Nhập mật khẩu
  6. button: Lấy lại mật khẩu khi quên mật khẩu
  7. button: Sau khi nhập thông tin tài khoản hợp lệ, di chuyển đến trang tiếp theo

Màn hình đăng ký tài khoản:



1. textfield: Nhập tên tài khoản đăng ký
2. button: Tên hợp lệ, nhấn nút tiếp tục để di chuyển đến trang kế tiếp
3. textfield: Nhập số điện thoại
4. textfield: Nhập mật khẩu
5. checkbox: Chọn đồng ý các điều khoản
6. button: Thông tin tài khoản, nhấn nút tiếp tục để di chuyển đến trang kế tiếp
7. textfield: Nhập mã OTP gửi về số điện thoại
8. button: Nhấn ‘tiếp túc’ để xác nhận và di chuyển vào trang home.

Màn hình nhắn tin, tìm kiếm:

A screenshot of a group of chat messages

Description automatically generated

1. textfield: Chọn tìm kiếm bạn bè theo số điện thoại và tên.
2. Chọn cuộc trò chuyện để nhắn tin.
3. Tab nhắn tin: Các cuộc hội thoại.
4. Tab contacts: Danh sách bạn bè, danh bạ, lời mời kết bạn…
5. Tab discover:
6. Tab timeline:
7. Tab user: Thông tin cá nhân người dùng.
8. Button: Gọi điện video.
9. Button: Xem chi tiết cuộc trò chuyện
10. Button: Trả lời tin nhắn
11. Button: Chuyển tiếp tin nhắn
12. Button: Thu hồi tin nhắn
13. Button: Xoá tin nhắn phía mình
14. Textfield: Gửi tin nhắn text
15. Button: Gửi tin nhắn file
16. Button: Gửi tin nhắn hình ảnh, video
17. Textfield: Nhập thông tin tìm kiếm
18. Button: Nút kết bạn.

Màn hình bạn bè, lời mời kết bạn:

Screens screenshot of a chat

Description automatically generated

1. Danh sách lời mời kết bạn
2. Danh bạ load từ thiết bị
3. Button: Gọi điện cho bạn bè
4. Button: Xoá bạn bè
5. Danh sách lời mời kết bạn đã nhận
6. Danh sách lời mời kết bạn đã gửi
7. Button: Thu hồi lời mời kết bạn đã gửi
8. Button: Từ chối lời mời kết bạn.
9. Button: Chập nhận lời mời kết bạn

Màn hình trang cá nhân:

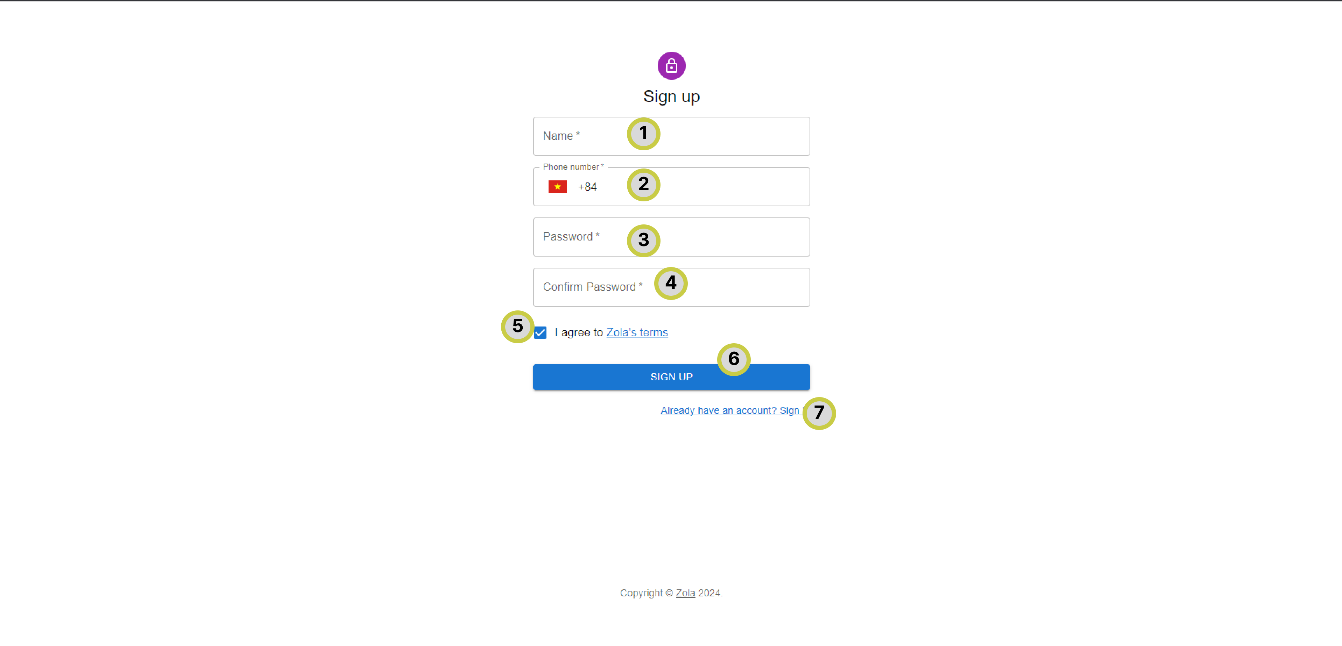
Screens screenshot of a chat

Description automatically generated

1. Xem trang cá nhân
2. Button: Đổi mật khẩu tài khoản.
3. Button: Đăng xuất khỏi thiết bị.
4. Đổi ảnh đại diện.
5. Button: Chỉnh sửa thông tin cá nhân.

## Giao diện của website

Màn hình đăng ký:



A screenshot of a computer

Description automatically generated

1. Textfield: Nhập tên
2. Textfield: Nhập số điện thoại
3. Textfield: Nhập mật khẩu
4. Textfield: Nhập lại mật khẩu
5. Checkbox: Đồng ý các điều khoản của website
6. Button: Nút đăng ký tài khoản
7. Đã có tài khoản => chuyển đến trang đăng nhập

Màn hình đăng nhập:

A screenshot of a computer

Description automatically generated

1. Textfield: Nhập tên
2. Textfield: Nhập số điện thoại
3. Button: Đăng nhập vào tài khoản
4. Quên mật khẩu => chuyển đến trang lấy lại mật khẩu.
5. Chưa có tài khoàn => chuyển đến trang đăng ký.
6. Textfield: Nhập số điện thoại
7. Button: Chọn submit.
8. Đã có tài khoản => quay lại trang đăng nhập

Màn hình nhắn tin:



1. Vào trang cá nhân
2. Trang tin nhắn, hiển thị các cuộc trò chuyện
3. Button Danh sách bạn bè, nhóm, lời mời kết bạn
4. Button Cài đặt của trang web
5. Button Chọn chi tiết cuộc trò chuyện.
6. Textfield Tìm kiếm bạn bè theo tên.
7. Button Tìm kiếm bạn bè theo số điện thoại, thêm bạn.
8. Button Tạo nhóm chat.
9. Button Gọi video.
10. Button Chi tiết cuộc trò chuyện.
11. Button Tuỳ chọn các hành động của tin nhắn
12. Button Chuyền tiếp tin nhắn.
13. Button Sao chép tin nhắn.
14. Button Xoá tin nhắn phía mình.
15. Button Thu hồi tin nhắn.
16. Button Gửi emoji
17. Button Nút like
18. Button Gửi ảnh
19. Button Gửi file, video
20. Text: Gửi tin nhắn text
21. Button: Đăng xuất

Màn hình danh sách nhóm, danh sách lời mời kết bạn:

A screenshot of a computer

Description automatically generated

1. Danh sách bạn bè
2. Danh sách nhóm
3. Danh sách lời mời kết bạn
4. Textfield: Tìm kiếm theo tên
5. Sắp xếp danh sách theo tên
6. Filter
7. Chọn cuộc trò chuyện -> vào chi tiết trò chuyện
8. Danh sách lời mời kết bạn đã nhận
9. Từ chối lời mời kết bạn
10. Chấp nhận lời mời kết bạn
11. Danh sách lời mời kết bạn đã gửi
12. Thu hồi lời mời kết bạn

Màn hình tạo nhóm chat, tìm bạn:

A screenshot of a computer

Description automatically generated

1. Textfield: Nhập số điện thoại cần tìm
2. Button: Huỷ tuỳ chọn.
3. Button: Tìm kiếm bạn bè
4. Textfield: Nhập tên nhóm cần tạo
5. Textfield: Nhập số điện thoại cần tìm
6. Chọn member cho nhóm.
7. Button: Huỷ tuỳ chọn.
8. Button: Tạo nhóm chat.

Màn hình chỉnh sửa thông tin cá nhân

A screenshot of a computer

Description automatically generated

1. Chỉnh sửa ảnh đại diện
2. Thay đổi thông tin cá nhân
3. Nhập tên mới
4. Chọn giới tính
5. Chọn ngày sinh
6. Button: submit thông tin
7. Textfield: Nhập mật khẩu cũ.
8. Textfield: Nhập mật khẩu mới
9. Textfield: Nhập lại mật khẩu mới
10. Button: Tuỳ chọn huỷ hành động hiện tại
11. Button: Lưu thông tin.

## Kế hoạch và hiện thực kiểm thử hệ thống

### Kế hoạch kiểm thử

**Mục tiêu kiểm thử:**

* Đảm bảo tính ổn định và đáng tin cậy của hệ thống phần mềm chat realtime.
* Xác nhận rằng hệ thống hoạt động đúng theo yêu cầu chức năng và yêu cầu phi chức năng.

**Phạm vi kiểm thử:**

* Kiểm thử tích hợp: Đảm bảo các thành phần của hệ thống hoạt động một cách mượt mà khi được kết hợp với nhau.
* Kiểm thử chức năng: Xác nhận rằng các chức năng của hệ thống hoạt động đúng theo yêu cầu.
* Kiểm thử hiệu suất: Đánh giá hiệu suất của hệ thống trong điều kiện tải cao.
* Kiểm thử bảo mật: Đảm bảo rằng hệ thống được bảo vệ khỏi các mối đe dọa và tấn công mạng.

**Kế hoạch kiểm thử**:

* Chuẩn bị môi trường kiểm thử: Triển khai một môi trường kiểm thử tách biệt để thực hiện các bước kiểm thử mà không ảnh hưởng đến môi trường sản xuất.
* Phát triển kịch bản kiểm thử: Xác định các trường hợp kiểm thử chức năng và phi chức năng, bao gồm cả trường hợp biên và trường hợp lỗi.
* Thực hiện kiểm thử: Thực hiện các bước kiểm thử theo kịch bản đã xác định và ghi lại kết quả kiểm thử.
* Đánh giá và báo cáo: Phân tích kết quả kiểm thử, bao gồm cả lỗi phát hiện được và đánh giá hiệu suất và bảo mật của hệ thống. Tạo báo cáo về kết quả kiểm thử để chia sẻ với các bên liên quan.

### Kiểm thử hệ thống

**Bước 1: Kiểm thử tích hợp**

Kiểm tra tích hợp giữa máy chủ và client.

Xác định và giải quyết các vấn đề tích hợp như xung đột tài nguyên.

**Bước 2: Kiểm thử chức năng**

Đăng ký tài khoản mới và xác thực đăng nhập.

Gửi tin nhắn và nhận tin nhắn từ các người dùng khác nhau.

Tạo và tham gia các cuộc trò chuyện nhóm.

Xác định và sửa các lỗi giao diện người dùng.

**Bước 3: Kiểm thử hiệu suất**

Tăng dần tải lên hệ thống để đánh giá hiệu suất và thời gian phản hồi của nó.

Đảm bảo hệ thống vẫn hoạt động mượt mà dưới áp lực tải cao.

**Bước 4: Kiểm thử bảo mật**

Kiểm tra xác thực và quyền truy cập của người dùng.

Thử các kỹ thuật tấn công mạng như SQL injection hoặc cross-site scripting để đảm bảo hệ thống có thể chống lại các cuộc tấn công này.

**Bước 5: Đánh giá và báo cáo**

**Bảng testing:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test ID | Chức năng | Mô tả | Điều kiện trước | Dữ liệu test | Kết quả mong muốn |
| UC01 | Đăng nhập | Nhập sai số điện thoại và mật khẩu |  | Sdt: 0002123121  Mật khẩu: password | Số điện thoại không tồn tại |
| UC02 | Đăng nhập | Nhập đúng số điện thoại, không nhập mật khẩu |  | Sđt: 0974067552  Mật khẩu: trống | Mật khẩu không được để trống |
| UC03 | Đăng nhập | Số điện thoại để trống, mật khẩu nhập đúng |  | Sđt: trống  Mật khẩu: 123456Aa | Số điện thoại không được để trống. |
| UC04 | Đăng nhập | Số điện thoại đúng, mật khẩu đúng |  | Sđt: 0339717984  Mật khẩu:  123456Hh | Thông báo đăng nhập thành công và chuyển vào trong. |
| UC05 | Đăng ký | Không nhập số điện thoại |  | Sđt: trống | Thông báo số điện thoại không được để trống |
| UC06 | Đăng ký | Không phải số điện thoại |  | Sđt: 0001231@# | Thông báo số điện thoại không hợp lệ |
| UC07 | Đăng ký | Nhập số điện thoại đã tồn tại |  | Sđt: 0339717894 | Thông báo số điện thoại đã tồn tại |
| UC08 | Đăng ký | Nhập số điện thoại hợp lệ |  | Sđt: 0974067552 | Chuyển đến trang đặt mật khẩu |
| UC09 | Đăng ký | Không nhập mật khẩu |  | Mật khẩu: trống | Thông báo mật khẩu không được để trống |
| UC010 | Đăng ký | Nhập mật khẩu không hợp lệ |  | Mật khẩu: %$%$%dsadsa | Thông báo mật khẩu không hợp lệ |
| UC011 | Đăng ký | Nhập mật khẩu hợp lệ |  | Mật khẩu: 123456Aa | Thông báo đăng ký tài khoản thành công. |

# : KẾT LUẬN

## Kết quả đạt được

* Triển khai thành công một hệ thống phần mềm chat realtime sử dụng công nghệ AWS.
* Xây dựng và kiểm thử các chức năng cơ bản của hệ thống như đăng nhập, gửi và nhận tin nhắn, quản lý cuộc trò chuyện nhóm.
* Đảm bảo tính ổn định và hiệu suất của hệ thống dưới tải cao.
* Nắm vững quy trình kiểm thử và đánh giá bảo mật của hệ thống

## Hạn chế của đồ án

* Thiếu sự phong phú và đa dạng trong các tính năng của hệ thống so với các ứng dụng chat realtime hàng đầu.
* Hệ thống chưa được tối ưu hoá đầy đủ về giao diện người dùng và trải nghiệm người dùng.
* Bản kiểm thử bảo mật chưa đủ chi tiết và không thể nắm bắt được tất cả các loại tấn công

## Hướng phát triển

* Tối ưu hoá giao diện người dùng và trải nghiệm người dùng để thu hút người dùng mới và giữ chân người dùng hiện tại.
* Mở rộng kiểm thử bảo mật để bao gồm nhiều loại tấn công hơn và tìm kiếm các lỗ hổng bảo mật tiềm ẩn.
* Nghiên cứu và triển khai tích hợp các công nghệ mới như machine learning để cải thiện chức năng phát hiện spam và các hành vi không mong muốn khác

# TÀI LIỆU THAM KHẢO

Các tài liệu Tiếng Anh

1. Kevin Faaborg & Sandro Pasquali - Mastering Node.js Second Edition – December 2017
2. David Herron, Diogo Resende & Valentin Bojinov - Node.js Complete Reference Guide – December 2018

Các tài liệu từ Internet

1. <https://nodejs.org/en/docs/>
2. <https://expressjs.com/en/guide/routing.html>/
3. <https://topdev.vn/blog/redis-la-gi/>
4. <https://firebase.google.com/docs/>