S

**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM**



**MÔN HỌC : CÔNG NGHỆ MỚI TRONG PHÁT TRIỂN ỨNG DỤNG CNTT**

**ĐỀ TÀI : ỨNG DỤNG NHẮN TIN ( CHAT APP)**

GVHD: Tôn Long Phước

Trần Thế Trung

***Nhóm 02 - Sinh viên thực hiện***

1. Trần Văn Tú– 20054351
2. Huỳnh Văn Duy – 20062071
3. Ngô Minh Thuận – 20053391
4. Hồ Thị Mỹ Quỳnh - 20053791

# MỤC LỤC

[MỤC LỤC 1](#_Toc166964998)

[CHƯƠNG 1 : GIỚI THIỆU 3](#_Toc166964999)

[1.1 Tổng quan 3](#_Toc166965000)

[1.2 Mục tiêu đề tài 3](#_Toc166965001)

[1.3 Phạm vi đề tài 4](#_Toc166965002)

[1.4 Mô tả yêu cầu chức năng 4](#_Toc166965003)

[1.4.1 Phân tích yêu cầu của hệ thống 4](#_Toc166965004)

[1.4.2 Mô tả hoạt động của hệ thống 5](#_Toc166965005)

[CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT 6](#_Toc166965006)

[2.1 Node.js 6](#_Toc166965007)

[Hình 2‑1 Node.js developmentttt 6](#_Toc166965008)

[2.2 ReactJs 9](#_Toc166965009)

[Hình 2-2: React Js 9](#_Toc166965010)

[2.3 React Native 11](#_Toc166965011)

[Hình 2-3: React Native 11](#_Toc166965012)

[2.4 Firebase 13](#_Toc166965013)

[Hình 2-4: Firebase 13](#_Toc166965014)

[CHƯƠNG 3 : PHÂN TÍCH VÀ THIẾT KẾ 16](#_Toc166965015)

[3.1 Phân tích yêu cầu bằng UML 16](#_Toc166965016)

[3.1.1 Usecase 16](#_Toc166965017)

[3.1.2 Danh sách tác nhân và mô tả 18](#_Toc166965018)

[Bảng 3‑1 Danh sách tác nhân và mô tả 19](#_Toc166965019)

[3.1.3 Danh sách các tình huống hoạt động (Use cases) 19](#_Toc166965020)

[3.1.4 Tình huống hoạt động 21](#_Toc166965021)

[3.1.4.1 UC01\_Đăng nhập 21](#_Toc166965022)

[3.1.4.2 UC02\_Đăng ký 23](#_Toc166965023)

[3.1.4.3 UC03\_ Quên mật khẩu 25](#_Toc166965024)

[3.1.4.4 UC04\_Cập Nhập Thông Tin Người Dùng 28](#_Toc166965025)

[3.1.4.5 UC05\_Tìm người dùng 29](#_Toc166965026)

[3.1.4.6 UC06\_Tạo cuộc trò truyện nhóm 32](#_Toc166965027)

[3.1.4.7 UC07\_Thêm thành viên vào nhóm 34](#_Toc166965028)

[3.1.4.8 UC08\_Xóa thành viên khỏi nhóm 35](#_Toc166965029)

[3.1.4.9 UC09\_Giải tán nhóm 37](#_Toc166965030)

[3.1.4.10 UC10\_Nhắn tin 39](#_Toc166965031)

[3.1.4.11 UC11\_Gửi file, hình ảnh, video 42](#_Toc166965032)

[3.1.4.12 UC12\_Thu hồi tin nhắn 44](#_Toc166965033)

[3.1.4.13 UC13\_Xóa tin nhắn 45](#_Toc166965034)

[3.1.4.14 UC14\_ Tìm kiếm bạn bè 47](#_Toc166965035)

[3.1.4.15 UC15\_Gửi yêu cầu kết bạn 49](#_Toc166965036)

[3.1.4.16 UC16\_ Chấp nhận yêu cầu kết bạn 51](#_Toc166965037)

[3.2 Class diagram 52](#_Toc166965038)

[3.3 Deployment diagram 53](#_Toc166965039)

[CHƯƠNG 4 : HIỆN THỰC 54](#_Toc166965040)

[4.1 Cấu hình phần cứng, phần mềm 54](#_Toc166965041)

[4.2 Giao diện của hệ thống 54](#_Toc166965042)

[4.2.1 Giao diện web 54](#_Toc166965043)

[4.2.2 Giao diện app 69](#_Toc166965044)

[4.2.3 Kế hoạch kiểm thử 92](#_Toc166965045)

[4.2.4 Kiểm thử hệ thống 98](#_Toc166965046)

[CHƯƠNG 5 : KẾT LUẬN 100](#_Toc166965047)

[5.1 Kết quả đạt được 100](#_Toc166965048)

[5.2 Hạn chế của đồ án 100](#_Toc166965049)

[5.3 Hướng phát triển 100](#_Toc166965050)

[TÀI LIỆU THAM KHẢO 101](#_Toc166965051)

# : GIỚI THIỆU

## Tổng quan

Trong bối cảnh hiện nay, thị trường đòi hỏi không chỉ sự tiện lợi, tốc độ và chất lượng mà còn mong muốn sự đa dạng và cá nhân hóa. Sự bùng nổ của công nghệ thông tin đã mở ra một cánh cửa rộng lớn cho sự đổi mới và phát triển của các ứng dụng trò chuyện trực tuyến. Điều này đã tạo ra một áp lực lớn đối với các nhà phát triển để không chỉ tạo ra những giải pháp dễ sử dụng mà còn phải đáp ứng được những yêu cầu riêng biệt của từng người dùng.

Dựa trên nhận định này và sự hỗ trợ từ giáo viên, nhóm của chúng tôi đã đặt mục tiêu xây dựng một ứng dụng nhắn tin (chat app) mang tính độc đáo và phù hợp với nhu cầu thị trường. Việc này không chỉ đòi hỏi chúng tôi phải hiểu rõ về công nghệ mà còn phải tiếp cận và đáp ứng được những mong muốn và đòi hỏi của người dùng. Dự án của chúng tôi sẽ không chỉ tạo ra một nền tảng chat trực tuyến mà còn là một không gian tương tác đa chiều, nơi mà người dùng có thể trải nghiệm sự kết nối và giao tiếp một cách tự nhiên và sinh động trên cả web và di động. Điều này sẽ mang lại cho họ trải nghiệm trao đổi thông tin không chỉ nhanh chóng mà còn phong phú và sáng tạo, từ đó nâng cao hiệu suất làm việc và tạo ra giá trị độc đáo cho mỗi người dùng.

## Mục tiêu đề tài

Ứng dụng nhắn tin hỗ trợ người dùng những chức năng sau:

Với nền tảng website người dùng có thể :

* Xem và chỉnh sửa thông tin cá nhân.
* Kết bạn dựa trên tên người dùng.
* Xem danh sách bạn bè, danh sách nhóm.
* Chat cá nhân
* Chat nhóm

Với nền tảng mobile người dùng có thể:

* Xem và chỉnh sửa thông tin cá nhân.
* Kết bạn dựa trên tên người dùng.
* Xem danh sách bạn bè, danh sách nhóm.
* Chat cá nhân
* Chat nhóm

## Phạm vi đề tài

* Giao diện dễ sử dụng và thân thiện, dễ thao tác và sử dụng.
* Cho phép khách hàng đăng kí tài khoản và đảm bảo thông tin khách hàng.
* Cho phép khách hàng thay đổi thông tin cá nhân, ảnh đại diện, ảnh bìa…
* Xem và thay đổi các thông tin về tài khoản.
* Chat nhóm và group, video call với người khác
* Xoá, thu hồi tin nhắn, gửi các sticker, emoji
* Xem và tìm kiếm theo nhiều danh mục.
* lưu trữ các tệp tin đa phương tiện như hình ảnh, video...

## Mô tả yêu cầu chức năng

### Phân tích yêu cầu của hệ thống

* Người dùng có thể thay đổi thông tin cá nhân như hình ảnh, tên, giới tính…
* Người dùng thực hiện chat trên ứng dụng, hệ thống sẽ tiếp nhận tin nhắn và gửi cho đối tượng chỉ định
* Cho phép người dùng gửi tin nhắn, emoji, hình ảnh và video.
* Người dùng có thể thu hồi hoặc xoá tin nhắn

Từ mô tả trên, có thể đưa ra yêu cầu của hệ thống với hai đối tượng chính tương tác với hệ thống như sau:

* Đối với người dùng:
* Tạo tài khoản, xem và chỉnh sửa thông tin cá nhân.
* Chat đơn hoặc chat group.
* Thu hồi, xoá tin nhắn
* Gửi emoij, hình ảnh và video…
* Kết bạn từ việc tìm kiếm tên người dùng.

### ***Mô tả hoạt động của hệ thống***

* Đối với người dùng:
* Người dùng cần phải đăng ký 1 tài khoản gồm email hoặc số điện thoại và mật khẩu trước khi sử dùng ứng dụng chat.
* Người dùng có thể kết bạn dựa trên việc tìm kiếm tên, sau khi có bạn bè trong danh sách, người dùng có thể chat với bạn bè của mình.
* Thành viên trong hộp thoại chat có thể gửi tin nhắn văn bản, đi kèm với hình ảnh, video, emoij .
* Khi gửi nhầm hoặc vô tình gửi tin nhắn, người dùng có thể thu hồi hoặc xoá tin nhắn mà mình vừa gửi.
* Ngoài ra, người dùng có thể tạo group chat để tiện trong việc trao đổi thông tin theo nhóm, với cách thức hoạt động tương tự như chat đơn

# : CƠ SỞ LÝ THUYẾT

## Node.js



## Hình 2‑1 Node.js developmentttt

1. Tổng quan:

Node.js là một nền tảng phát triển được xây dựng trên JavaScript runtime của Chrome V8 Engine, cho phép viết mã JavaScript chạy phía máy chủ. Nó ra đời vào năm 2009 bởi Ryan Dahl và ngày nay đã trở thành một trong những công cụ phổ biến nhất cho việc xây dựng ứng dụng web và dịch vụ mạng.

Dưới đây là một số điểm tổng quan về Node.js:

JavaScript Everywhere: Node.js cho phép sử dụng JavaScript ở cả phía máy chủ và phía người dùng, tạo ra một môi trường phát triển đồng nhất và giảm thiểu sự chuyển đổi giữa các ngôn ngữ lập trình.

Non-blocking, Asynchronous I/O: Node.js sử dụng mô hình xử lý không chặn và bất đồng bộ (non-blocking, asynchronous) để xử lý nhiều yêu cầu cùng một lúc mà không cần tạo ra các luồng mới cho mỗi yêu cầu. Điều này làm cho Node.js rất hiệu quả trong việc xử lý các ứng dụng web thời gian thực và có khả năng mở rộng tốt.

Nền tảng mã nguồn mở: Node.js được phát triển dựa trên mã nguồn mở và được hỗ trợ bởi một cộng đồng lớn các nhà phát triển, điều này giúp nó phát triển nhanh chóng và linh hoạt.

Modules và npm: Node.js có hệ thống modules mạnh mẽ, cho phép phát triển ứng dụng một cách modul và tái sử dụng code dễ dàng. npm (Node Package Manager) là một công cụ quản lý gói mở rộng của Node.js, cung cấp hàng ngàn thư viện và công cụ để giúp phát triển ứng dụng một cách nhanh chóng và hiệu quả.

Đa nhiệm: Node.js có thể được sử dụng để xây dựng các ứng dụng web, API, ứng dụng di động, ứng dụng desktop và nhiều hơn nữa. Điều này tạo ra sự linh hoạt và đa năng cho người phát triển.

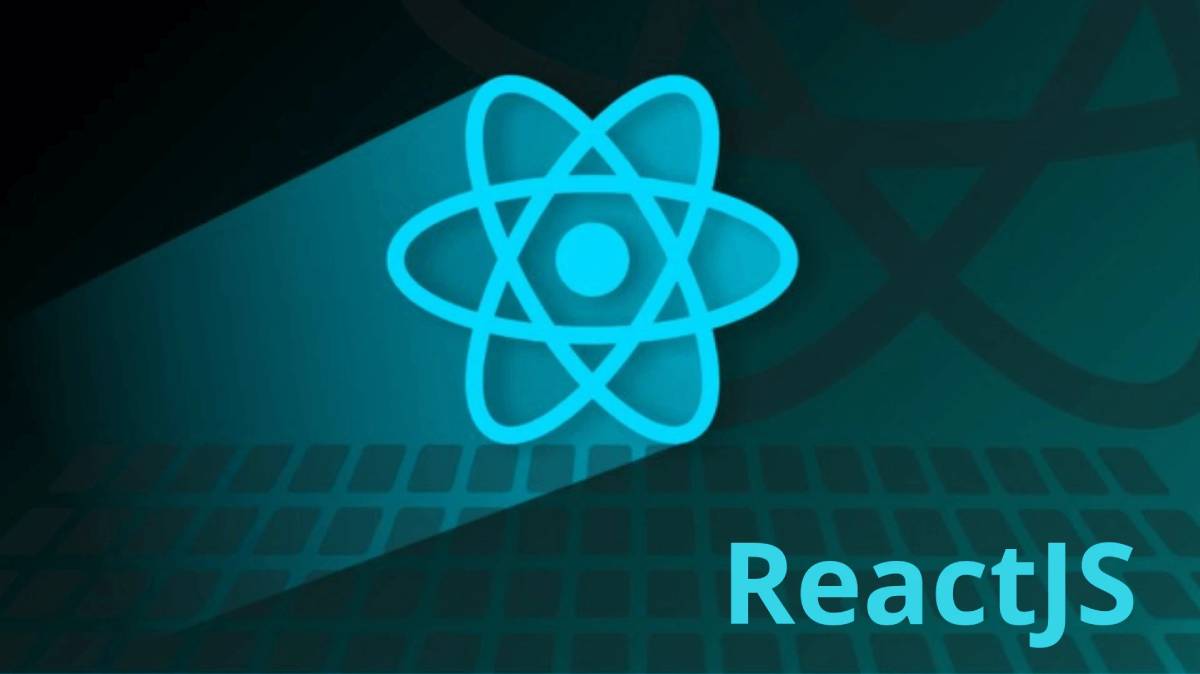
1. Ưu điểm :

* Hiệu suất cao: Node.js sử dụng mô hình không chặn và bất đồng bộ, cho phép xử lý hàng ngàn kết nối cùng một lúc mà không làm chậm tốc độ của ứng dụng.
* JavaScript đồng nhất: Sử dụng JavaScript ở cả phía máy chủ và phía người dùng tạo ra một môi trường phát triển đồng nhất, giảm thiểu sự phân mảnh và giúp đơn giản hóa việc quản lý code.
* Nền tảng mã nguồn mở và cộng đồng lớn: Node.js được phát triển và hỗ trợ bởi cộng đồng lớn các nhà phát triển, điều này tạo điều kiện thuận lợi cho việc chia sẻ kiến thức, học hỏi và giải quyết vấn đề.
* Modules và npm: Có hệ thống modules mạnh mẽ và npm cung cấp hàng ngàn thư viện và công cụ hữu ích giúp phát triển ứng dụng một cách nhanh chóng và hiệu quả.
* Đa nhiệm: Node.js có thể được sử dụng để xây dựng nhiều loại ứng dụng khác nhau bao gồm ứng dụng web, API, ứng dụng di động và nhiều hơn nữa.

1. Nhược điểm :

* Không thích hợp cho các ứng dụng CPU-intensive: Node.js thích hợp cho các ứng dụng I/O-intensive nhưng không hiệu quả cho các ứng dụng CPU-intensive vì nó không sử dụng đa luồng.
* Callback hell: Sự lồng ghép của các hàm callback có thể dẫn đến callback hell, khiến code trở nên khó đọc và bảo trì.
* Phát triển không đồng đều: Một số tính năng và thư viện của Node.js có thể phát triển không đồng đều hoặc không được hỗ trợ tốt bởi cộng đồng.
* Nhược điểm về bảo mật: Do sự phổ biến của Node.js, có nhiều lỗ hổng bảo mật có thể được tìm thấy và khai thác nếu không được quản lý và bảo vệ cẩn thận.
* Khó khăn trong việc theo dõi và gỡ lỗi: Do tính bất đồng bộ và các quá trình xử lý song song, việc theo dõi và gỡ lỗi các lỗi có thể trở nên khó khăn hơn so với các nền tảng khác.

## ReactJs



## Hình 2-2: React Js

1. Tổng quan

Component-based: React.js sử dụng một mô hình lập trình dựa trên components, cho phép phân chia giao diện người dùng thành các thành phần độc lập và tái sử dụng được, giúp tăng khả năng bảo trì và phát triển.

Virtual DOM: React.js sử dụng Virtual DOM để tối ưu hóa hiệu suất của ứng dụng. Thay vì cập nhật trực tiếp DOM mỗi khi có thay đổi, React tạo ra một bản sao ảo của DOM và chỉ cập nhật những phần thay đổi, giúp tăng tốc độ và hiệu suất của ứng dụng.

JSX: React.js sử dụng JSX (JavaScript XML) để viết code, cho phép viết HTML trong JavaScript, tạo ra một cú pháp dễ đọc và dễ hiểu hơn.

Cộng đồng lớn và hỗ trợ tốt: React.js có một cộng đồng lớn và sự hỗ trợ mạnh mẽ từ Facebook và cộng đồng, điều này giúp giải quyết vấn đề nhanh chóng và dễ dàng.

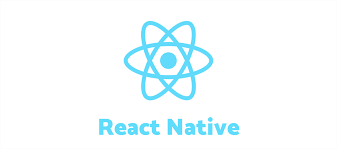
1. Ưu điểm

* Hiệu suất cao: Nhờ vào Virtual DOM và cách React cập nhật DOM hiệu quả, ứng dụng React thường có hiệu suất cao và đáp ứng tốt đối với người dùng.
* Component-based architecture: Các thành phần tái sử dụng giúp giảm mã lặp lại, tăng sự bảo trì và phát triển của ứng dụng.
* JSX: JSX tạo ra một cú pháp rõ ràng và dễ đọc, giúp làm cho code dễ hiểu hơn và giảm thiểu lỗi.
* Thư viện và công cụ hỗ trợ: React.js có một hệ sinh thái đa dạng của các thư viện và công cụ hỗ trợ như Redux, React Router, Styled Components, giúp phát triển ứng dụng một cách nhanh chóng và hiệu quả.

1. Nhược điểm

* Học phức tạp ban đầu: JSX và cách tiếp cận component-based có thể làm cho việc học React ban đầu khó khăn đối với một số người mới bắt đầu.
* Có thể phức tạp với ứng dụng lớn: Với các ứng dụng lớn và phức tạp, quản lý trạng thái và dữ liệu có thể trở nên phức tạp và khó khăn hơn.
* Độ lớn của cộng đồng và công nghệ mới: Mặc dù React có một cộng đồng lớn và mạnh mẽ, nhưng đôi khi có thể khá khó để theo kịp với các công nghệ mới và thay đổi trong hệ sinh thái React.

## React Native



## Hình 2-3: React Native

1. Tổng quan

React Native là một framework phát triển ứng dụng di động sử dụng JavaScript và React, được phát triển bởi Facebook. Nó cho phép nhà phát triển xây dựng ứng dụng di động đa nền tảng (iOS và Android) với một lượng mã nguồn chung. Dưới đây là một tổng quan và các ưu và nhược điểm của React Native:

Tổng quan:

Cross-platform development: React Native cho phép phát triển ứng dụng di động cho cả iOS và Android từ một mã nguồn duy nhất. Điều này giúp tiết kiệm thời gian và công sức cho việc phát triển và bảo trì ứng dụng.

JavaScript-based: React Native sử dụng JavaScript làm ngôn ngữ lập trình chính, giúp cho các nhà phát triển đã có kinh nghiệm với JavaScript có thể nhanh chóng chuyển sang phát triển ứng dụng di động.

Native performance: React Native sử dụng các thành phần UI native, giúp cho ứng dụng có thể đạt được hiệu suất gần như native, mặc dù vẫn sử dụng JavaScript để điều khiển logic.

Hot reloading: React Native hỗ trợ hot reloading, cho phép các nhà phát triển xem các thay đổi ngay lập tức khi code được chỉnh sửa, giúp tăng tốc độ phát triển.

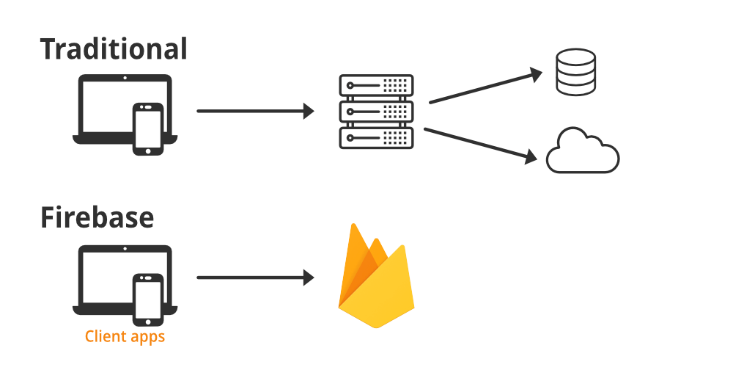
1. Ưu điểm

* Tích hợp dễ dàng: React Native dễ dàng tích hợp với các thư viện và frameworks khác như Redux, React Navigation, và nhiều thư viện khác, giúp mở rộng và tùy chỉnh ứng dụng một cách linh hoạt.
* Hiệu suất gần như native: Với việc sử dụng các thành phần UI native, React Native cho phép ứng dụng đạt được hiệu suất gần như native, đồng thời giảm thiểu sự chậm trễ và giảm tải cho người dùng.
* Cộng đồng lớn và hỗ trợ mạnh mẽ: React Native có một cộng đồng lớn và sự hỗ trợ mạnh mẽ từ Facebook và cộng đồng, điều này giúp giải quyết vấn đề nhanh chóng và dễ dàng.

1. Nhược điểm

* Khả năng tùy chỉnh giới hạn: Mặc dù React Native cho phép tích hợp với các thư viện native, nhưng đôi khi việc tùy chỉnh giao diện và tính năng có thể bị giới hạn so với việc sử dụng công nghệ native.
* Kích thước ứng dụng lớn: Do việc sử dụng các thành phần UI native, có thể ứng dụng React Native có kích thước lớn hơn so với các ứng dụng native truyền thống.
* Phụ thuộc vào phiên bản React Native: Việc phát triển của React Native có thể bị ảnh hưởng bởi sự thay đổi trong phiên bản framework, đòi hỏi các nhà phát triển phải duy trì và cập nhật ứng dụng của mình để đảm bảo tính tương thích.

## Firebase



## Hình 2-4: Firebase

1. Tổng quan

Firebase là một nền tảng phát triển ứng dụng di động và web do Google phát triển. Nó cung cấp một loạt các dịch vụ đám mây như cơ sở dữ liệu thời gian thực, lưu trữ tệp, xác thực người dùng, phân tích, thông báo đẩy, và nhiều dịch vụ khác, giúp nhà phát triển xây dựng và triển khai ứng dụng một cách nhanh chóng và dễ dàng. Dưới đây là một tổng quan và các ưu và nhược điểm của Firebase:

Tổng quan:

* Dịch vụ toàn diện: Firebase cung cấp một loạt các dịch vụ đám mây cho phép phát triển ứng dụng từ đầu đến cuối, từ lưu trữ dữ liệu cho đến xác thực người dùng, phân tích và thông báo đẩy.
* Thời gian thực: Firebase Realtime Database là một cơ sở dữ liệu thời gian thực, cho phép dữ liệu được đồng bộ hóa ngay lập tức giữa các thiết bị và máy chủ.
* Xác thực người dùng: Firebase Authentication cung cấp các phương tiện xác thực người dùng bảo mật như email/password, OAuth, và các dịch vụ xác thực mạng xã hội.
* Lưu trữ tệp: Firebase Storage cho phép lưu trữ và quản lý các tệp đa phương tiện như hình ảnh, video và tệp âm thanh trên đám mây.
* Dễ sử dụng và tích hợp: Firebase cung cấp SDK dễ sử dụng cho nhiều nền tảng, bao gồm JavaScript, iOS và Android, và có thể dễ dàng tích hợp vào các ứng dụng hiện có.

1. Ưu điểm

* Tăng tốc độ phát triển: Firebase cung cấp các công cụ mạnh mẽ và dễ sử dụng, giúp tăng tốc quá trình phát triển và triển khai ứng dụng.
* Khả năng mở rộng và linh hoạt: Firebase cho phép mở rộng ứng dụng của bạn một cách dễ dàng, từ việc xử lý hàng trăm đến hàng nghìn người dùng.
* Tính đồng nhất và đồng bộ: Firebase cung cấp một nền tảng đồng nhất và đồng bộ, giúp đảm bảo rằng dữ liệu và tác vụ của bạn luôn được đồng bộ và đảm bảo.
* Hỗ trợ tốt từ cộng đồng: Firebase có một cộng đồng rộng lớn và tích cực, với các tài liệu hướng dẫn, diễn đàn, và nguồn tài nguyên phong phú.

1. Nhược điểm

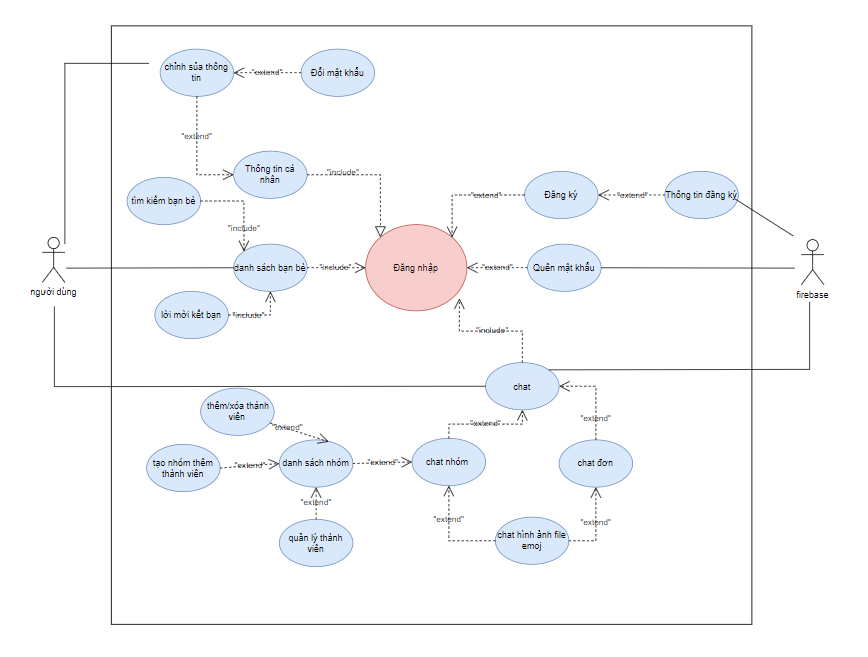
* Giới hạn về tùy chỉnh: Firebase có thể giới hạn trong việc tùy chỉnh và mở rộng cho các ứng dụng phức tạp và đòi hỏi tính linh hoạt cao.
* Chi phí: Mặc dù Firebase cung cấp một số dịch vụ miễn phí, nhưng chi phí có thể tăng lên khi sử dụng các tính năng cao cấp hoặc khi ứng dụng của bạn mở rộng.
* Phụ thuộc vào nền tảng của bên thứ ba: Firebase là một sản phẩm của Google, do đó, sự phụ thuộc vào nền tảng của một công ty lớn có thể là một vấn đề cho một số tổ chức.

# : PHÂN TÍCH VÀ THIẾT KẾ

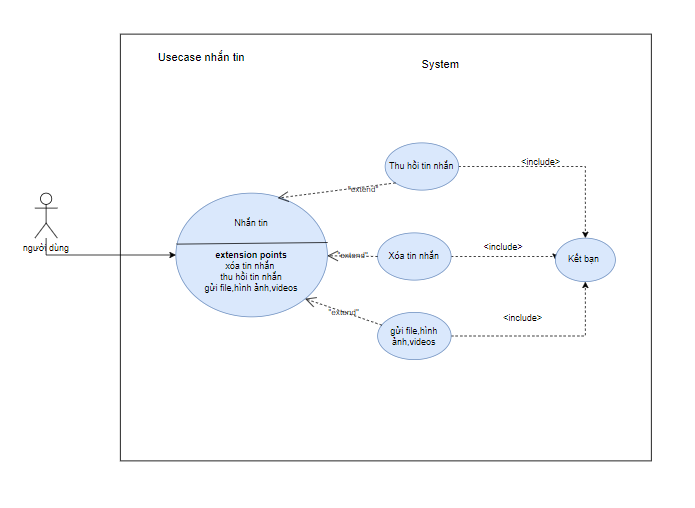
## Phân tích yêu cầu bằng UML

### Usecase

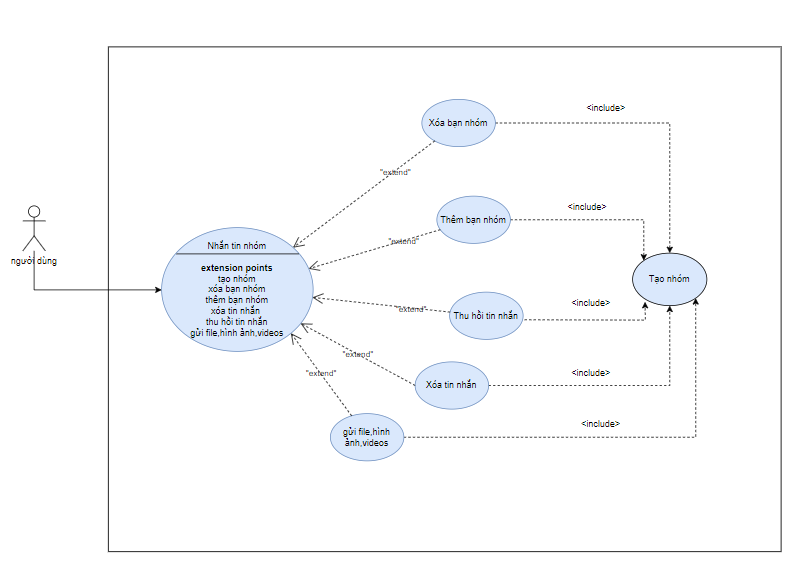
***Use case tổng quát***

******

***Usecase nhắn tin đơn :***

******

***Usecase nhắn tin nhóm:***

******

### 3.1.2 Danh sách tác nhân và mô tả

|  |  |
| --- | --- |
| Tác nhân | Mô tả tác nhân |
| User | Đăng ký tài khoản   * Đăng nhập * Quản lí thông tin cá nhân : * Cập nhật thông tin cá nhân * Tìm người dùng * Kết bạn với người dùng * Quản lý bạn bè:   + Tìm kiếm bạn bè   + Thêm bạn bè * Quản lí tin nhắn * Gửi tin nhắn * Thu hồi tin nhắn * Xóa tin nhắn * Gửi file(ảnh, video, word, pdf,…) * Quản lý nhóm * Tạo nhóm * Thêm thành viên nhóm * Xóa thành viên nhóm |

#### 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 nhập |
| UC02 | Đăng ký |
| UC03 | Quên mật khẩu |
| UC04 | Cập Nhập Thông Tin Người Dùng |
| UC05 | Tìm người dùng |
| UC06 | Tạo cuộc trò truyện nhóm |
| UC07 | Thêm thành viên vào nhóm |
| UC08 | Xóa thành viên khỏi nhóm |
| UC09 | Giải tán nhóm |
| UC10 | Nhắn tin |
| UC11 | Gửi file, hình ảnh, video |
| UC12 | Thu hồi tin nhắn |
| UC13 | Xóa tin nhắn |
| UC14 | Tìm kiếm bạn bè |
| UC15 | Gửi yêu cầu kết bạn |
| UC16 | Chấp nhận yêu cầu kết bạn |

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

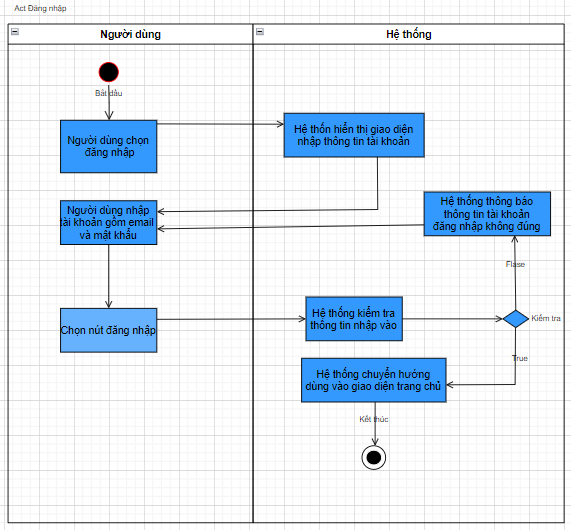
### Tình huống hoạt động

Đặc tả use case.

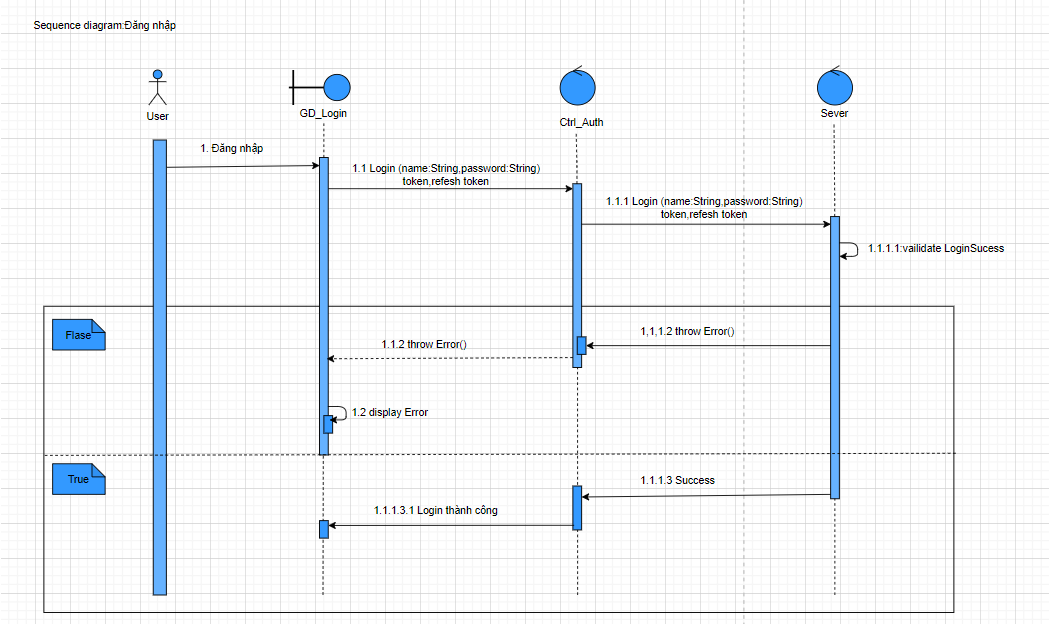
#### UC01\_Đăng nhập

|  |  |
| --- | --- |
| * **Tên use case**: UC01\_Đăng nhập | |
| * **Mô tả sơ lược**: Người dùng muốn đăng nhập vào hệ thống để sử dụng các dịch vụ | |
| * **Actor chính**: Người dùng | |
| * **Actor phụ**: | |
| * **Tiền điều kiện (Pre-condition):** Người dùng đã có tài khoản | |
| * **Hậu điều kiện (Post-condition):** | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Người dùng chọn chức năng đăng nhập |  |
|  | 2. Hệ thống hiển thị giao diện nhập thông tin tài khoản |
| 1. Người dùng nhập tài khoản gồm email và mật khẩu |  |
| 1. Chọn nút đăng nhập |  |
|  | 1. Hệ thống kiểm tra thông tin nhập vào |
|  | 1. Hệ thống chuyển hướng người dùng vào giao diện trang chủ |
|  |  |
| * **Luồng sự kiện thay thế (alternate flow):** | |
|  | 6.1. Hệ thống thông báo thông tin tài khoản đăng nhập không đúng |
|  | 6.2. Hệ thống quay lại bước 3 |

Activity diagram:



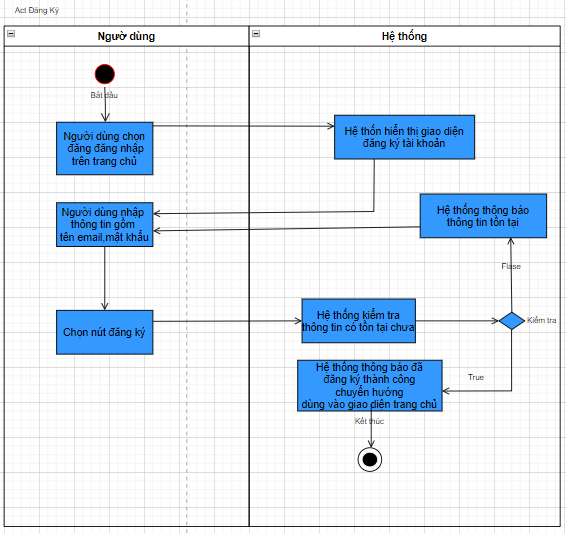
Sequence diagram:



#### UC02\_Đăng ký

|  |  |
| --- | --- |
| * **Tên use case**: UC02\_Đăng ký | |
| * **Mô tả sơ lược**: Người dùng muốn đăng ký vào hệ thống để sử dụng các dịch vụ | |
| * **Actor chính**: Người dùng | |
| * **Actor phụ**: Không. | |
| * **Tiền điều kiện (Pre-condition):** Người dùng chưa có tài khoản | |
| * **Hậu điều kiện (Post-condition):** Người dùng tạo tài khoản thành công | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Người dùng chọn chức năng đăng ký trên trang chủ |  |
|  | 1. Hiển thị giao diện đăng ký |
| 1. Người dùng nhập thông tin gồm tên, email, mật khẩu |  |
| 1. Người dùng nhấn đăng ký |  |
|  | 1. Hệ thống kiểm tra có tồn tại tài khoản chưa |
|  | 1. Thông báo đang ký thành công và chuyển hướng tới giao diện trang chủ |
| * **Luồng sự kiện thay thế (alternate flow):** | |
|  | 5.1. Hệ thống thông báo đã tồn tại |
|  | 5.2 Hệ thống quay lại bước 3 |

Activity diagram:



Sequence diagram:

#### UC03\_ Quên mật khẩu

|  |  |
| --- | --- |
| * **Tên use case**: UC03\_ Quên mật khẩu | |
| * **Mô tả sơ lược**: Người dùng lấy lại mật khẩu đã mất | |
| * **Actor chính**: Người dùng | |
| * **Actor phụ**: Không. | |
| * **Tiền điều kiện (Pre-condition):** Người dùng đã có tài khoản | |
| * **Hậu điều kiện (Post-condition):** Người dùng lấy lại được mật khẩu mới | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Người dùng chọn chức năng quên mật khẩu. |  |
|  | 1. Hiển thị giao diện lấy lại mật khẩu |
| 1. Người dùng nhập email và bấm xác nhận |  |
|  | 1. Hệ thống kiểm tra tài khoản |
|  | 1. Hệ thống hiển thị giao diện nhập mật khẩu |
|  | 1. Mật khẩu mới được gửi về gmail |
| 1. Người dùng nhập mật khẩu mới từ gmail |  |
|  | 1. Hệ thống kiểm tra mật khẩu |
|  | 1. Hệ thống xác nhận trang đăng nhập thành công. |
| * **Luồng sự kiện thay thế (alternate flow):** | |
|  | 4.1. Hệ thống thông báo không tồn tại |
|  | 4.2. Hệ thống quay lại bước 3 |
|  | * 1. Hệ thống thông báo mật khẩu không hợp lệ |
|  | 7.2 Hệ thống quay lại bước 2 |

Activity diagram:

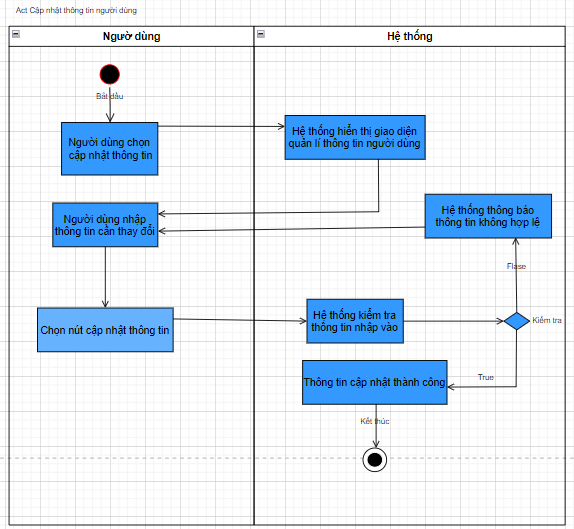
A diagram of a computer

Description automatically generated with medium confidence

#### UC04\_Cập Nhập Thông Tin Người Dùng

|  |  |
| --- | --- |
| * **Tên use case**: UC04\_Cập Nhập Thông Tin Người Dùng | |
| * **Mô tả sơ lược**: Giúp người cập nhập thông tin của người dùng trong hệ thông | |
| * **Actor chính**: người dùng | |
| * **Actor phụ**: Không. | |
| * **Tiền điều kiện (Pre-condition):** Người dùng đăng nhập thành công vào hệ thống | |
| * **Hậu điều kiện (Post-condition):** Thông tin người dùng được cập nhập | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Người dùng chọn chức năng cập nhật thông tin |  |
|  | 1. Hệ thống hiển thị giao diện quản lí thông tin người dùng |
| 1. Người dùng nhập thông tin tài khoản cần thay đổi |  |
| 1. Người dùng nhấn cập nhật |  |
|  | 1. Hệ thống kiểm tra thông tin nhập vào |
|  | 1. Thông báo cập nhập thành công |
| * **Luồng sự kiện thay thế (alternate flow):** | |
|  | 5.1. Hệ thống thông báo thông tin không hợp lệ |
|  | 5.2. Hệ thống quay lại bước 3 |

Activity diagram:



#### UC05\_Tìm người dùng

|  |  |
| --- | --- |
| **Use case:** UC05\_Tìm người dùng | |
| **Mô tả sơ lược**: Giúp người sử dụng tìm một người dùng khác thông qua tên người dùng | |
| **Actor chính**: Người dùng. | |
| **Actor phụ**: Không. | |
| **Tiền điều kiện (Pre-condition):** Người dùng đã đăng nhập vào hệ thống và đang ở trang chủ. | |
| **Hậu điều kiện (Post-condition):** | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Người dùng chọn vào biểu tượng “thêm bạn” |  |
|  | 1. Hệ thống hiển thị thanh tìm kiếm |
| 1. Người dùng nhập Tên của người dùng mà mình muốn tìm kiếm |  |
| 1. Người dùng bấm nút tìm kiếm |  |
|  | 1. Hệ thống kiểm tra tính hợp lệ của Tên |
|  | 1. Hệ thống gửi yêu cầu tìm kiếm đến server |
|  | 1. Hệ thống nhận kết quả trả về và hiển thị thông tin người dùng |
| * **Luồng sự kiện thay thế (alternate flow):** | |
|  | 5.1 Hệ thống thông báo tên không hợp lệ |
|  | 5.2 Quay lại bước 3 |
| * **Luồng sự kiện ngoại lệ (exception flow):** | |

Activity diagram:

A diagram of a workflow

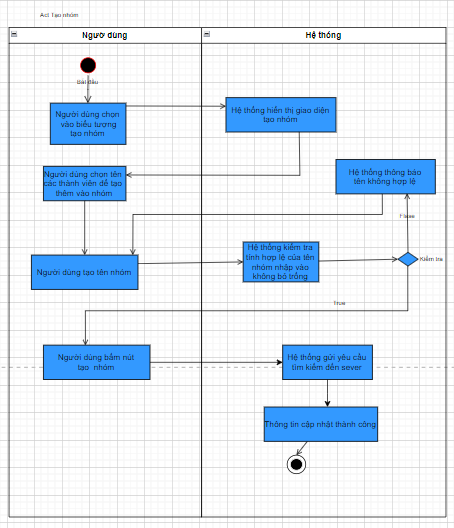
Description automatically generated

Sequence diagram:

#### UC06\_Tạo cuộc trò truyện nhóm

|  |  |
| --- | --- |
| **Use case:** UC06\_Tạo cuộc trò truyện nhóm | |
| **Mô tả sơ lược**: Giúp người dùng tạo một cuộc trò chuyện với nhiều thành viên | |
| **Actor chính**: Người dùng. | |
| **Actor phụ**: Không. | |
| **Tiền điều kiện (Pre-condition):** Người dùng đã đăng nhập vào hệ thống và đang ở trang chủ. | |
| **Hậu điều kiện (Post-condition):** | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Người dùng chọn vào biểu tượng tạo nhóm |  |
|  | 1. Hệ thống hiển thị Modal tạo nhóm |
| 1. Người dùng chọn các thành viên để thêm vào nhóm |  |
| 1. Người dùng nhập tên nhóm |  |
|  | 1. Hệ thống kiểm tra tính hợp lệ của tên nhóm(không bỏ trống) |
| 1. Người dùng bấm vào nút “tạo nhóm” |  |
|  | 1. Hệ thống gửi yêu cầu đến server |
|  | 1. Hệ thống tạo nhóm mới và thêm nhóm vừa tạo vào danh sách nhóm. |
| * **Luồng sự kiện thay thế (alternate flow):** | |
|  | 5.1 Hệ thống thông báo tên nhập không hợp lệ |
|  | 5.2 Hệ thống quay lại bước 4 |
| * **Luồng sự kiện ngoại lệ (exception flow):** | |

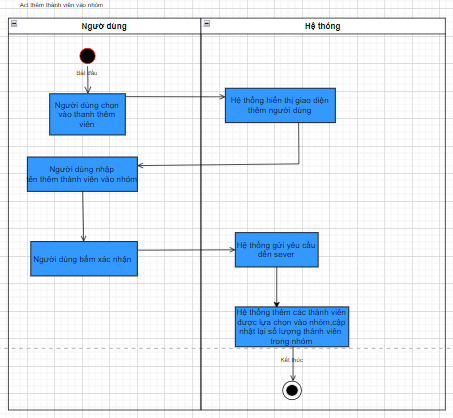
Activity diagram:



#### UC07\_Thêm thành viên vào nhóm

|  |  |
| --- | --- |
| **Use case:** UC07\_Thêm thành viên vào nhóm | |
| **Mô tả sơ lược**: Thêm một người dùng vào nhóm | |
| **Actor chính**: Người dùng. | |
| **Actor phụ**: Không. | |
| **Tiền điều kiện (Pre-condition):** Người dùng (trưởng nhóm)đã đăng nhập vào hệ thống và đang ở giao diện chat của cuộc hội thoại | |
| **Hậu điều kiện (Post-condition):** | |
| * **Luồng sự kiện chính (main flow):** | |
| **Actor** | **System** |
| 1. Người dùng chọn vào biểu tượng “thêm thành viên vào nhóm” |  |
|  | 1. Hệ thống hiển thị Modal thêm thành viên vào nhóm |
| 1. Người dùng bấm thêm thành viên vào nhóm |  |
| 1. Người dùng bấm vào nút xác nhận |  |
|  | 1. Hệ thống gửi yêu cầu đến server |
|  | 1. Hệ thống thêm các thành viên được lựa chọn vào nhóm, cập nhật lại số lượng thành viên nhóm |
| * **Luồng sự kiện thay thế (alternate flow):** | |
| * **Luồng sự kiện ngoại lệ (exception flow):** | |

Activity diagram:



#### UC08\_Xóa thành viên khỏi nhóm

|  |  |  |
| --- | --- | --- |
| **Use case:** UC08\_Xóa thành viên khỏi nhóm | | |
| **Mô tả sơ lược**: Xóa thành viên ra khỏi nhóm trò chuyện | | |
| **Actor chính**: Người dùng. | | |
| **Actor phụ**: Không. | | |
| **Tiền điều kiện (Pre-condition):** Người dùng đã đăng nhập vào hệ thống và đang ở giao diện chat của cuộc hội thoại và là nhóm trưởng của cuộc hội thoại. | | |
| **Hậu điều kiện (Post-condition):** | | |
| * **Luồng sự kiện chính (main flow):** | | |
| **Actor** | **System** | |
| 1. Người dùng chọn vào mục thành viên ở phần thông tin nhóm | 1. Hệ thống hiển thị giao diện các thành viên trong nhóm. | |
| 1. Người dùng bấm vào thành viên muốn xóa và chọn xóa thành viên | 1. Hệ thống gửi yêu cầu đến server | |
|  | 1. Hệ thống xóa thành viên được chọn ra khỏi nhóm trò chuyện | |
|  | 1. Hệ thống cập nhật lại số lượng các thành viên trong nhóm | |
| * **Luồng sự kiện thay thế (alternate flow):** | | |
| 5.1 người dùng chọn ‘huỷ’ để dừng tác vụ | |  |
| * **Luồng sự kiện ngoại lệ (exception flow):** | | |

Activity diagram:

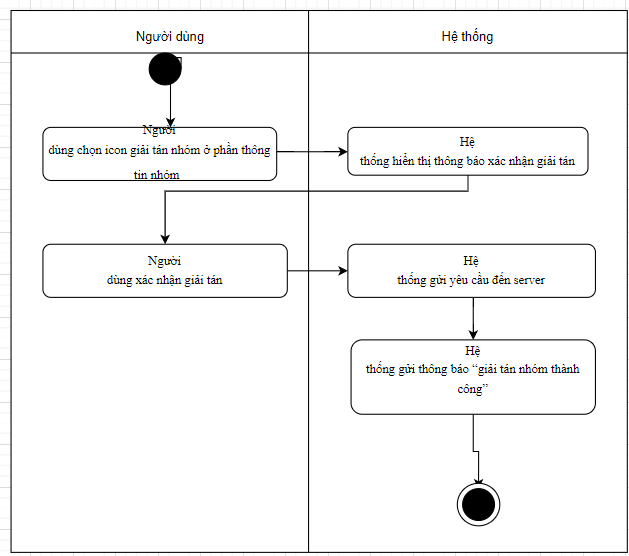
A diagram with blue squares and black dots

Description automatically generated

#### 3.1.4.9 UC09\_Giải tán nhóm

|  |  |  |
| --- | --- | --- |
| **Use case:** UC09\_Giải tán nhóm | | |
| **Mô tả sơ lược**: giải tán nhóm trò chuyện | | |
| **Actor chính**: Người dùng. | | |
| **Actor phụ**: Không. | | |
| **Tiền điều kiện (Pre-condition):** Người dùng đã đăng nhập vào hệ thống và đang ở giao diện chat của cuộc hội thoại và là nhóm trưởng của cuộc hội thoại. | | |
| **Hậu điều kiện (Post-condition):** | | |
| * **Luồng sự kiện chính (main flow):** | | |
| **Actor** | **System** | |
| * + - 1. Người dùng chọn icon giải tán nhóm ở phần thông tin nhóm |  | |
|  | * + - 1. Hệ thống hiển thị thông báo xác nhận giải tán | |
| * + - 1. Người dùng xác nhận giải tán |  | |
|  | * + - 1. Hệ thống gửi yêu cầu đến server | |
|  | * + - 1. Hệ thống gửi thông báo “giải tán nhóm thành công” | |
| * **Luồng sự kiện thay thế (alternate flow):** | | |
| 3.1 người dùng chọn ‘huỷ’ để dừng tác vụ | |  |
| * **Luồng sự kiện ngoại lệ (exception flow):** | | |

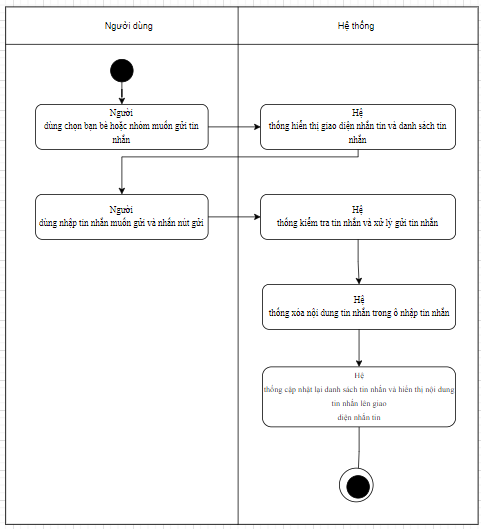
Activity diagram:

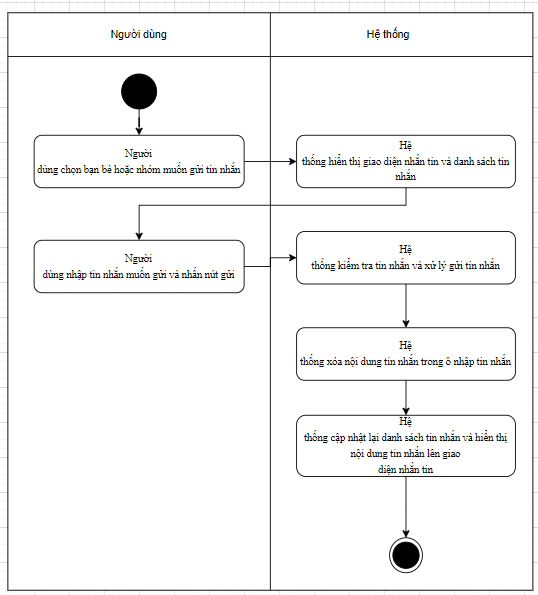


#### UC10\_Nhắn tin

|  |  |
| --- | --- |
| **Use case:** UC11\_Nhắn tin | |
| **Mô tả sơ lược**: Nhắn tin với bạn bè hoặc trong một nhóm chat | |
| **Actor chính**: Người dùng | |
| **Actor phụ**: Không. | |
| **Tiền điều kiện (Pre-condition):** Người dùng đã đăng nhập vào hệ thống và đã tạo cuộc trò chuyện với bạn bè hoặc nhóm muốn nhắn tin. | |
| **Hậu điều kiện (Post-condition):** Người dùng có thể nhắn tin với bạn bè hoặc nhóm trong cửa sổ chat. | |
| * **Luồng sự kiện chính (main flow):** | |
| **Người dùng** | **Hệ thống** |
| 1. Người dùng chọn bạn bè hoặc nhóm muốn gửi tin nhắn |  |
|  | 1. Hệ thống hiển thị giao diện nhắn tin và danh sách tin nhắn |
| 1. Người dùng nhập tin nhắn muốn gửi và nhấn nút gửi |  |
|  | 1. Hệ thống kiểm tra tin nhắn và xử lý gửi tin nhắn |
|  | 1. Hệ thống xóa nội dung tin nhắn trong ô nhập tin nhắn |
|  | 1. Hệ thống cập nhật lại danh sách tin nhắn và hiển thị nội dung tin nhắn lên giao diện nhắn tin |
| * **Luồng sự kiện thay thế (alternate flow):** | |
|  | 4.1. Hệ thống kiểm tra bội dung tin nhắn không hợp lệ (rỗng, chỉ chứa kí tự khoảng trắng) |
| * **Luồng sự kiện ngoại lệ (exception flow):** | |
|  |  |

Activity diagram:

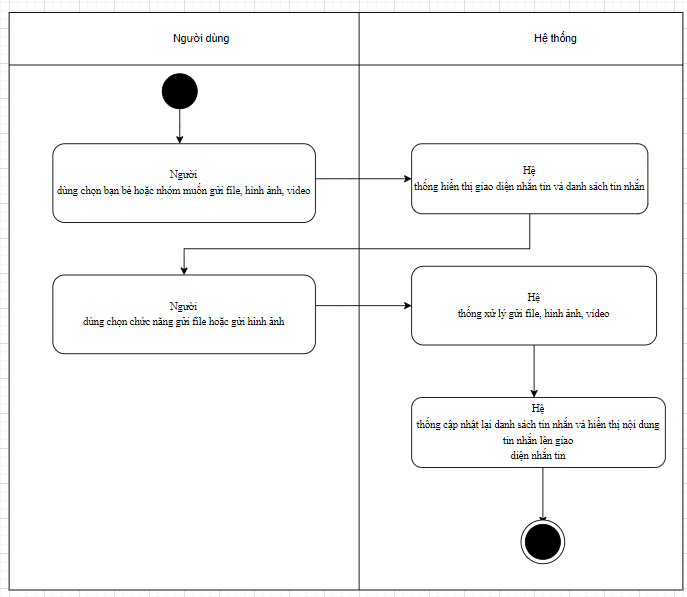




#### UC11\_Gửi file, hình ảnh, video

|  |  |
| --- | --- |
| **Use case:** UC11\_Gửi file, hình ảnh, video | |
| **Mô tả sơ lược**: Gửi file, hình ảnh, video với bạn bè hoặc trong một nhóm chat | |
| **Actor chính**: Người dùng | |
| **Actor phụ**: không. | |
| **Tiền điều kiện (Pre-condition):** Người dùng đã đăng nhập vào hệ thống và đã tạo cuộc trò chuyện với bạn bè hoặc nhóm muốn gửi file, hình ảnh, video. | |
| **Hậu điều kiện (Post-condition):** Người dùng có thể gửi file, hình ảnh, video với bạn bè trong cửa sổ chat. | |
| * **Luồng sự kiện chính (main flow):** | |
| **Người dùng** | **Hệ thống** |
| 1. Người dùng chọn bạn bè hoặc nhóm muốn gửi file, hình ảnh, video |  |
|  | 1. Hệ thống hiển thị giao diện nhắn tin và danh sách tin nhắn |
| 1. Người dùng chọn chức năng gửi file hoặc gửi hình ảnh |  |
| 1. Người dùng tìm và chọn file, hình ảnh, video cần gửi |  |
|  | 1. Hệ thống xử lý gửi file, hình ảnh, video |
|  | 1. Hệ thống cập nhật lại danh sách tin nhắn và hiển thị nội dung tin nhắn lên giao diện nhắn tin |
| * **Luồng sự kiện thay thế (alternate flow):** | |
|  |  |
| * **Luồng sự kiện ngoại lệ (exception flow):** | |
|  | * 1. Hệ thống thông báo gửi thất bại |

Activity diagram:



#### UC12\_Thu hồi tin nhắn

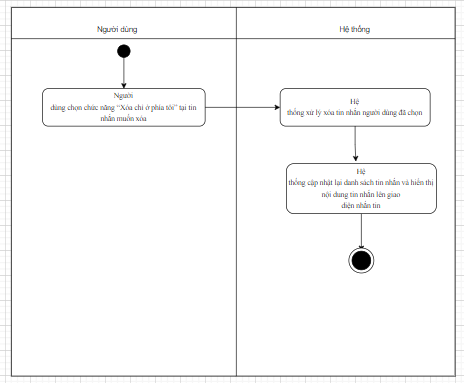
|  |  |
| --- | --- |
| **Use case:** UC13\_Thu hồi tin nhắn | |
| **Mô tả sơ lược**: Thu hồi tin nhắn đã gửi với bạn bè hoặc trong một nhóm chat | |
| **Actor chính**: Người dùng | |
| **Actor phụ**: Không. | |
| **Tiền điều kiện (Pre-condition):** Người dùng đã đăng nhập vào hệ thống và đang ở giao diện nhắn tin. | |
| **Hậu điều kiện (Post-condition):** Người dùng có thể thu hồi tin nhắn đã gửi với bạn bè trong cửa sổ chat. | |
| * **Luồng sự kiện chính (main flow):** | |
| **Người dùng** | **Hệ thống** |
| 1. Người dùng chọn chức năng “thu hồi” tại tin nhắn muốn thu hồi |  |
|  | 1. Hệ thống xử lý thu hồi tin nhắn |
|  | 1. Hệ thống cập nhật lại danh sách tin nhắn và hiển thị nội dung tin nhắn lên giao diện nhắn tin |
| * **Luồng sự kiện thay thế (alternate flow):** | |
|  |  |
| * **Luồng sự kiện ngoại lệ (exception flow):** | |
|  |  |

Activity diagram:

#### UC13\_Xóa tin nhắn

|  |  |
| --- | --- |
| **Use case:** UC14\_Xóa tin nhắn | |
| **Mô tả sơ lược**: Xóa tin nhắn trong cửa sổ chat của người dùng (“Xóa chỉ ở phía tôi”) | |
| **Actor chính**: Người dùng | |
| **Actor phụ**: Không. | |
| **Tiền điều kiện (Pre-condition):** Người dùng đã đăng nhập vào hệ thống và đang ở giao diện nhắn tin. | |
| **Hậu điều kiện (Post-condition):** Người dùng có thể xóa tin nhắn bất kì chỉ trong cửa sổ chat của mình | |
| * **Luồng sự kiện chính (main flow):** | |
| **Người dùng** | **Hệ thống** |
| 1. Người dùng chọn chức năng “Xóa chỉ ở phía tôi” tại tin nhắn muốn xóa |  |
|  | 1. Hệ thống xử lý xóa tin nhắn người dùng đã chọn |
|  | 1. Hệ thống cập nhật lại danh sách tin nhắn và hiển thị nội dung tin nhắn lên giao diện nhắn tin |
| * **Luồng sự kiện thay thế (alternate flow):** | |
|  |  |
| * **Luồng sự kiện ngoại lệ (exception flow):** | |
|  |  |

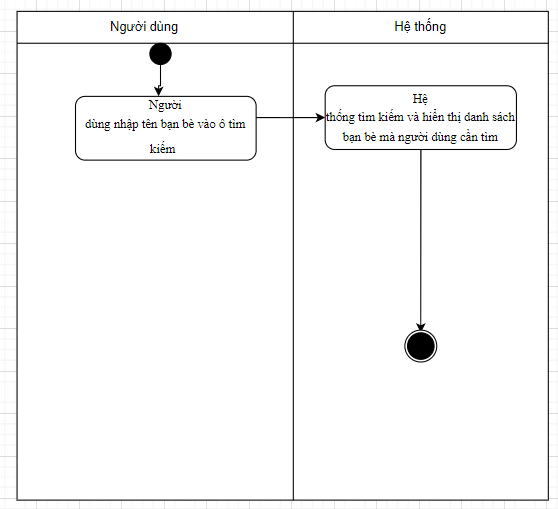
Activity diagram:



#### UC14\_ Tìm kiếm bạn bè

|  |  |
| --- | --- |
| **Use case:** UC15\_Tìm kiếm bạn bè | |
| **Mô tả sơ lược**: Người dùng tìm kiếm bạn bè | |
| **Actor chính**: Người dùng | |
| **Actor phụ**: Không. | |
| **Tiền điều kiện (Pre-condition):** Người dùng đã đăng nhập vào hệ thống và đang ở giao diện bạn bè; | |
| **Hậu điều kiện (Post-condition):** Người dùng tìm kiếm bạn bè trong danh sách bạn bè. | |
| * **Luồng sự kiện chính (main flow):** | |
| **Người dùng** | **Hệ thống** |
| 1. Người dùng nhập tên bạn bè vào ô tìm kiếm |  |
|  | 1. Hệ thống tìm kiếm và hiển thị danh sách bạn bè mà người dùng cần tìm |
| * **Luồng sự kiện thay thế (alternate flow):** | |
|  |  |
| * **Luồng sự kiện ngoại lệ (exception flow):** | |
|  |  |

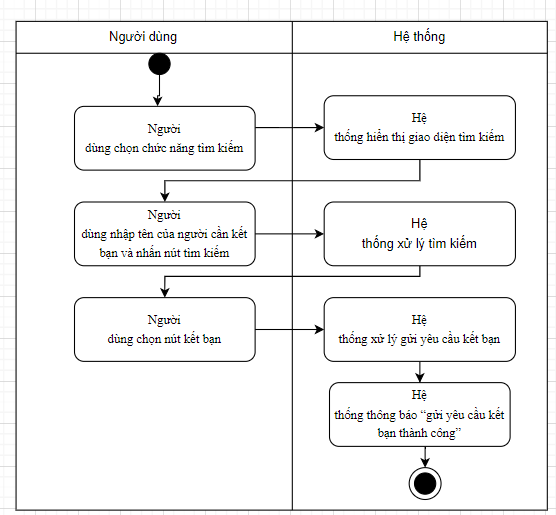
Activity diagram:



#### UC15\_Gửi yêu cầu kết bạn

|  |  |
| --- | --- |
| **Use case:** UC16\_Gửi yêu cầu kết bạn | |
| **Mô tả sơ lược**: Người dùng gửi yêu cầu kết bạn | |
| **Actor chính**: Người dùng | |
| **Actor phụ**: Không. | |
| **Tiền điều kiện (Pre-condition):** Người dùng đã đăng nhập vào hệ thống và đang ở giao diện bạn bè; | |
| **Hậu điều kiện (Post-condition):** Người dùng có thể gửi yêu cầu kết bạn cho bạn bè | |
| * **Luồng sự kiện chính (main flow):** | |
| **Người dùng** | **Hệ thống** |
| 1. Người dùng chọn chức năng tìm kiếm |  |
|  | 1. Hệ thống hiển thị giao diện tìm kiếm |
| 1. Người dùng nhập tên của người cần kết bạn và nhấn nút tìm kiếm |  |
|  | 1. Hệ thống xử lý tìm kiếm |
| 1. Người dùng chọn nút kết bạn |  |
|  | 1. Hệ thống xử lý gửi yêu cầu kết bạn |
|  | 1. Hệ thống thông báo “gửi yêu cầu kết bạn thành công” |
| * **Luồng sự kiện thay thế (alternate flow):** | |
|  | 4.1 Hệ thống thông báo “không tìm thấy” |
| * **Luồng sự kiện ngoại lệ (exception flow):** | |
|  |  |

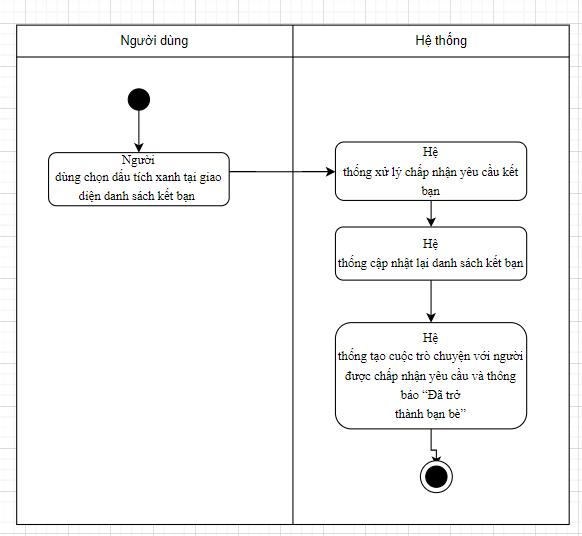
Activity diagram:



#### UC16\_ Chấp nhận yêu cầu kết bạn

|  |  |
| --- | --- |
| **Use case:** UC17\_Chấp nhận yêu cầu kết bạn | |
| **Mô tả sơ lược**: Người dùng từ chấp nhận yêu cầu kết bạn | |
| **Actor chính**: Người dùng | |
| **Actor phụ**: Không. | |
| **Tiền điều kiện (Pre-condition):** Người dùng đã đăng nhập vào hệ thống và đang ở giao diện bạn bè; người dùng đã nhận được yêu cầu kết bạn từ người khác. | |
| **Hậu điều kiện (Post-condition):** Người dùng có thể chấp nhận yêu cầu kết bạn. | |
| * **Luồng sự kiện chính (main flow):** | |
| **Người dùng** | **Hệ thống** |
| 1. Người dùng chọn dấu tích xanh tại giao diện danh sách kết bạn |  |
|  | 1. Hệ thống xử lý chấp nhận yêu cầu kết bạn |
|  | 1. Hệ thống cập nhật lại danh sách kết bạn |
|  | 1. Hệ thống tạo cuộc trò chuyện với người được chấp nhận yêu cầu và thông báo “Đã trở thành bạn bè” |
| * **Luồng sự kiện thay thế (alternate flow):** | |
|  |  |
| * **Luồng sự kiện ngoại lệ (exception flow):** | |
|  |  |

Activity diagram:



## Class diagram

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

A diagram of a user

Description automatically generated

## Deployment diagram

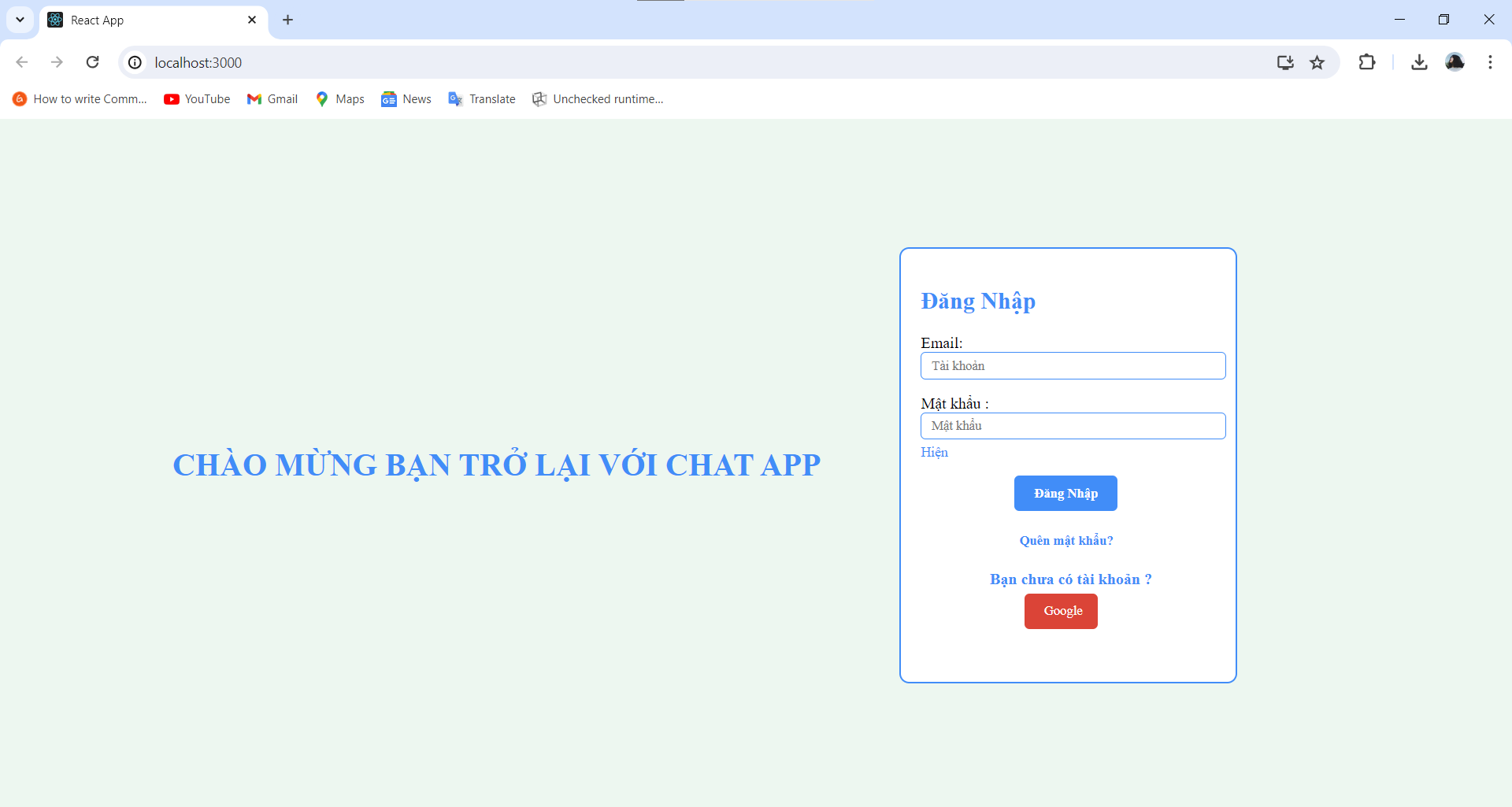
# : HIỆN THỰC

## Cấu hình phần cứng, phần mềm

* Các dịch vụ FireBase:
  + Storage:lưu trữ ảnh
  + FireStore:lưu trữ dữ liệu
  + Authentication: lưu trữ tài khoản
* Cấu hình Sever: Visual code studio
* Cấu hình Client: Visual code studio

## Giao diện của hệ thống

### Giao diện web



6

5

4

3

2

1

Hình 4.2.1.1 Đăng nhập

1 : Nhập email

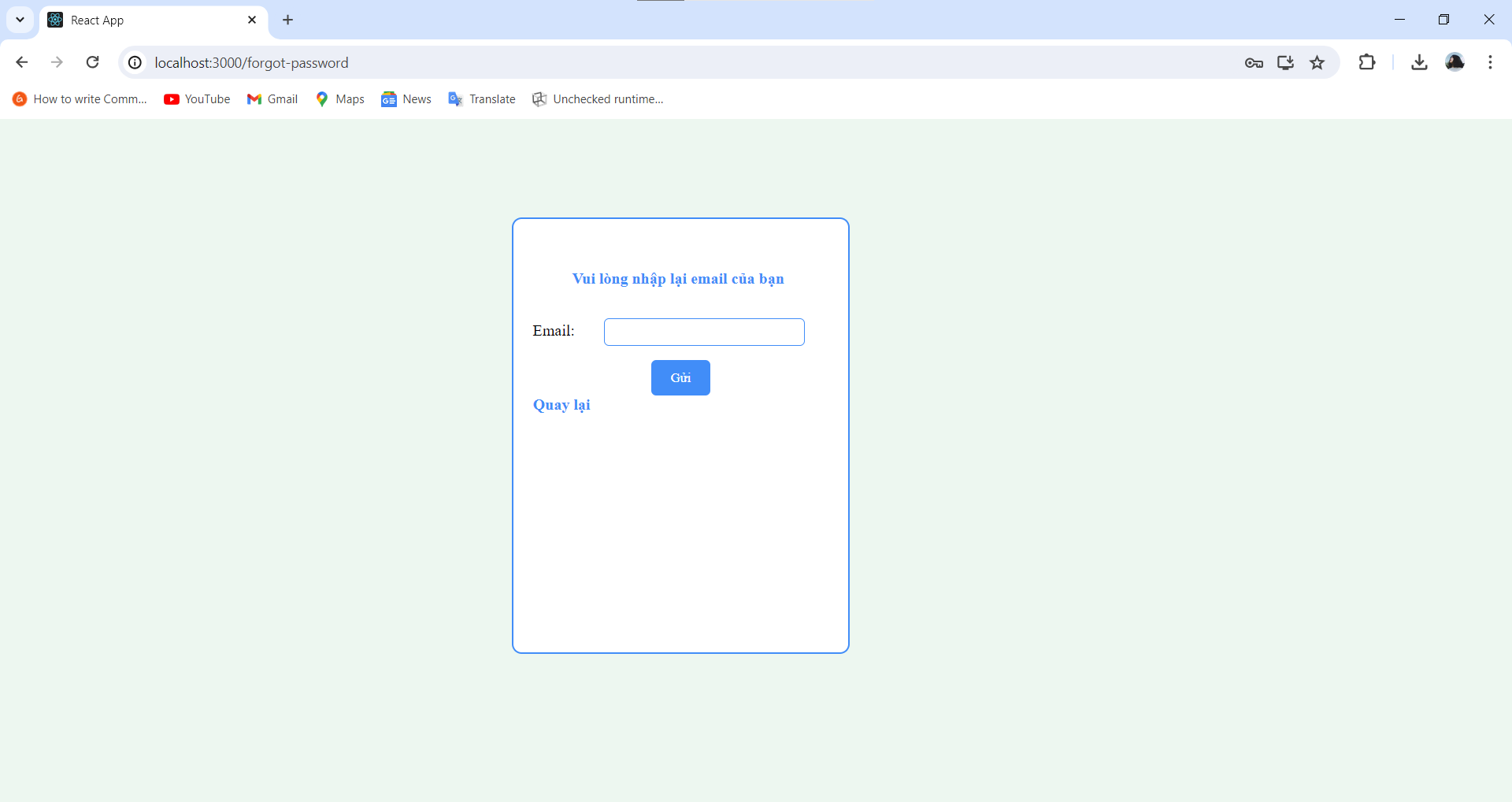
2 : Nhập mật khẩu

3: Nhấn để đăng nhập

4 : Nhấn “Quên mật khẩu “ nếu quên mật khẩu

5: Nếu chưa có tài khoản thì nhấn chuyển trang

6: Đăng nhập bằng tài khoản google



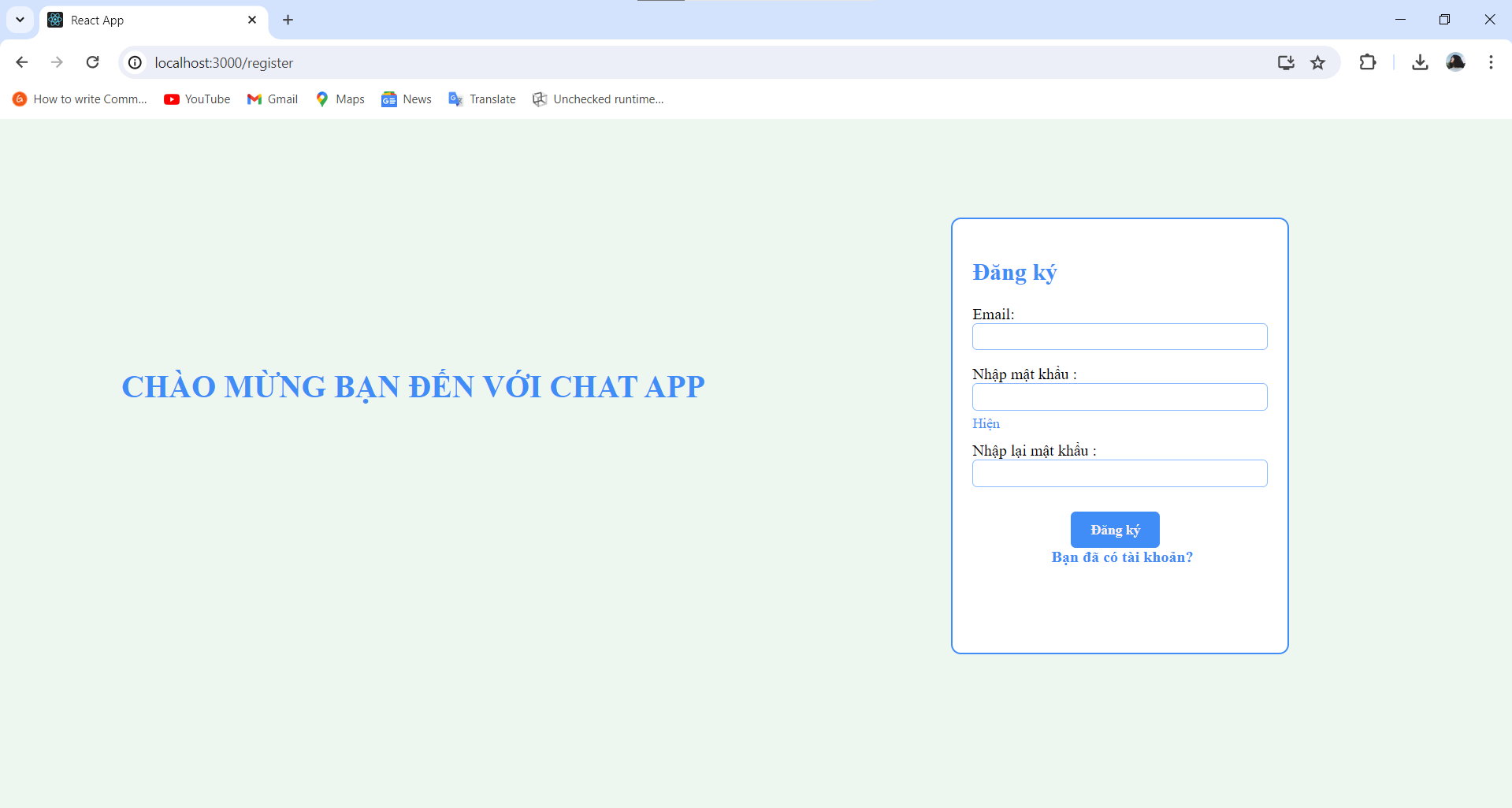
2

1

Hình 4.2.1.2 Quên mật khẩu

1: Nhập email

2: Nhấn nút gửi



5

4

3

2

1

Hình 4.2.1.3. Đăng kí

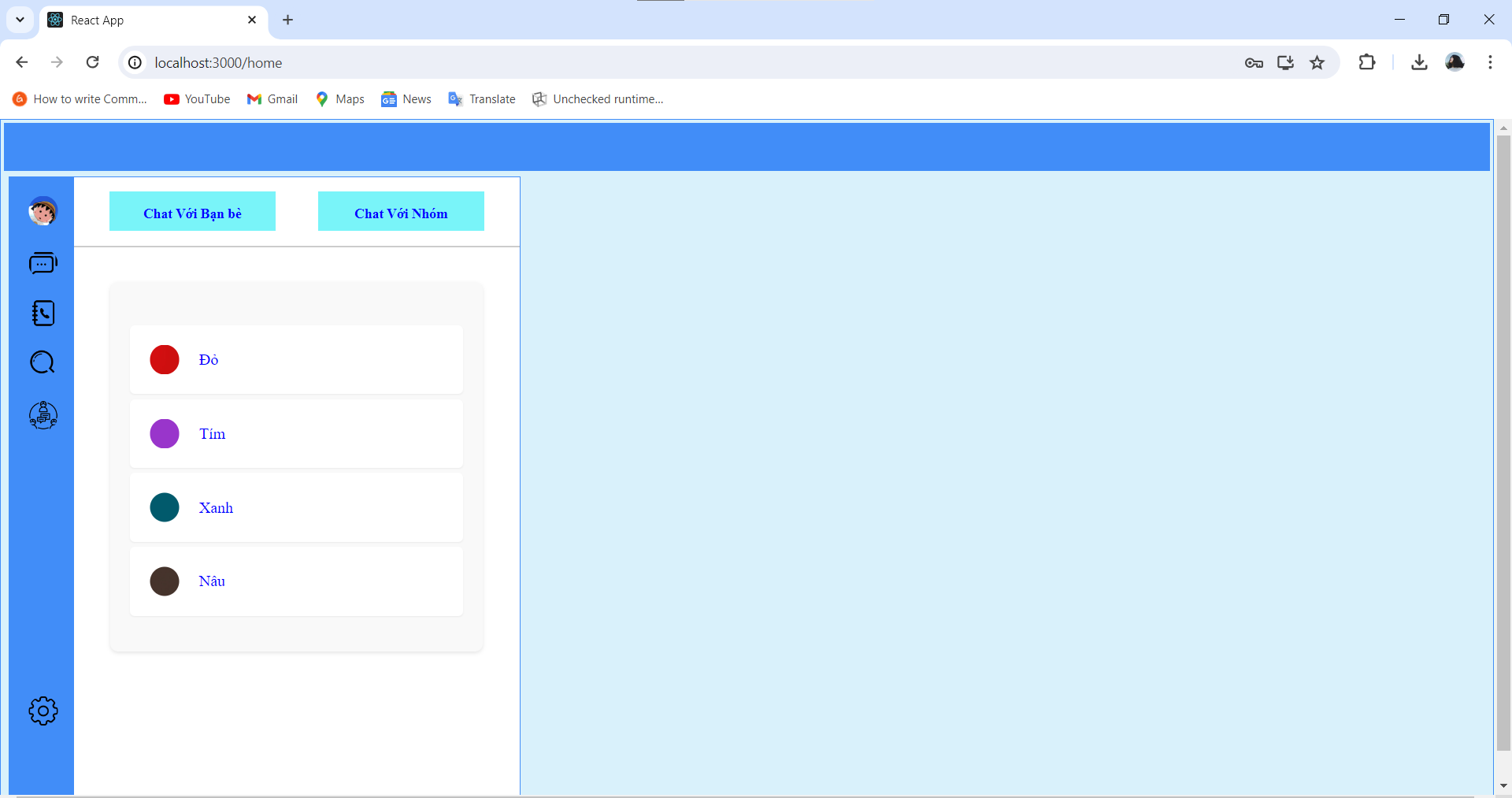
1 : Nhập email

2: Nhập mật khẩu

3: Nhập lại mật khẩu đã nhập

4: Nhấn để đăng kí tài khoản

5: Nếu có tài khoản thì nó chuyển hướng sang trang đăng nhập



6

3

2

1

5

4

Hình 4.2.1.4 Trang chính

1: Nhấn vào ảnh đại diện để xem thông tin và cập nhật thông tin giống như hình 4.2.1.4

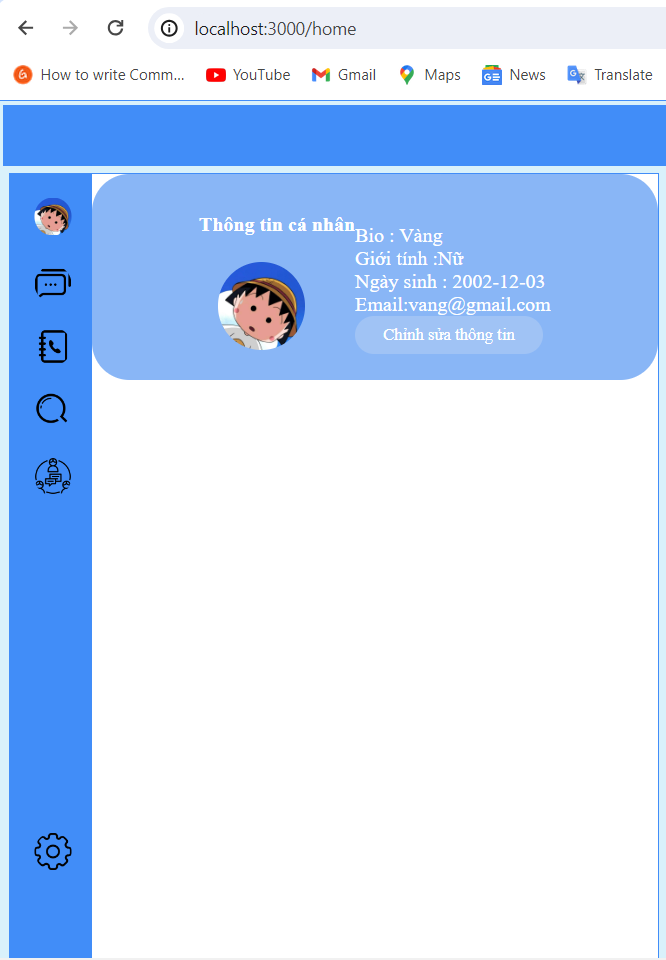
2: Chứa danh sách tin nhắn

3: Chứa lời mời gửi/nhận

4: Tìm kiếm bạn bè

5: Nhóm/ tạo nhóm

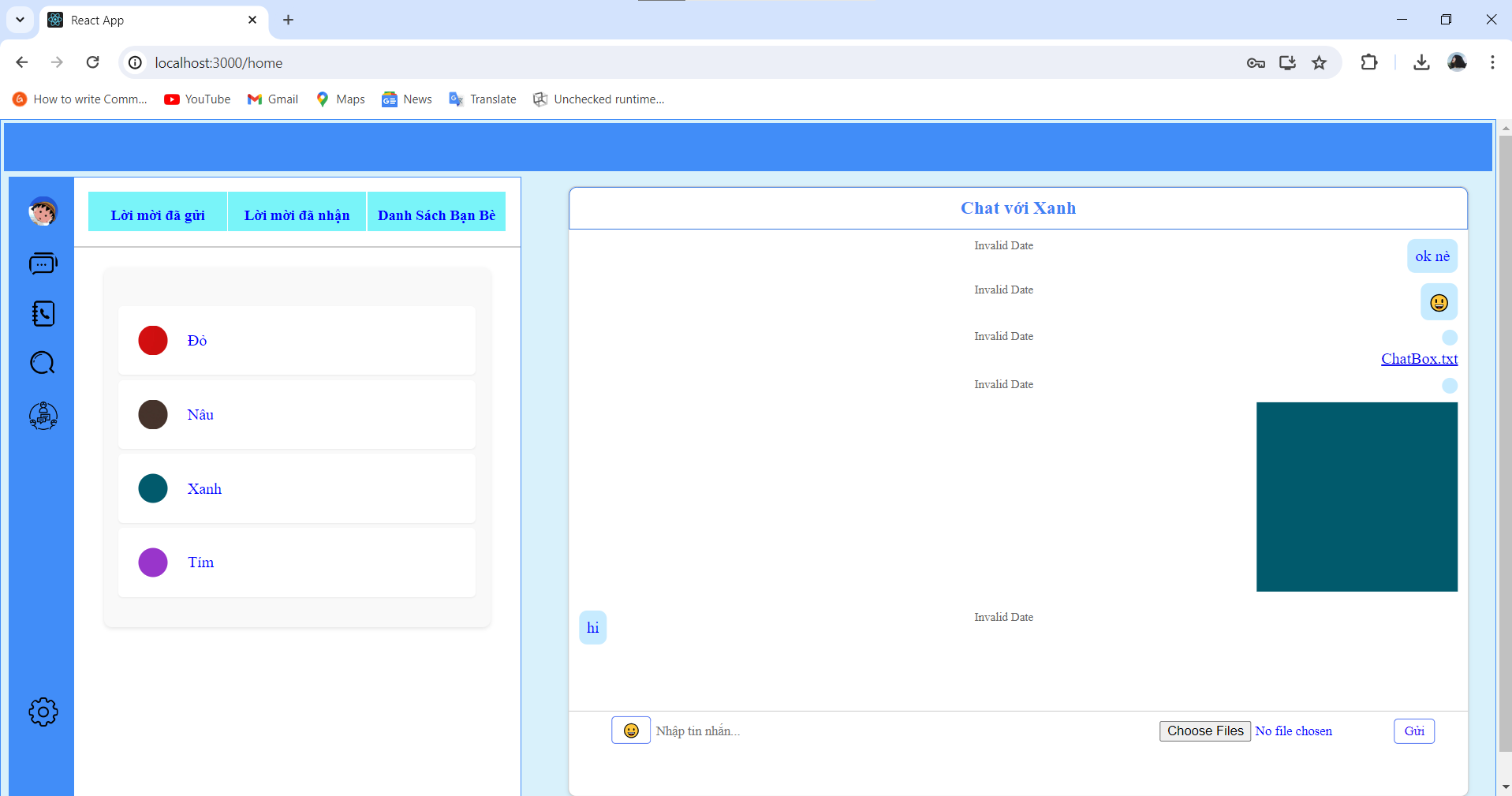
5: Cài đặt chứa đăng xuất



1

Hình 4.2.1.5 Thông tin cá nhân

1: Nút chỉnh sửa thông tin



10

9

8

7

6

5

4

3

2

1

Hình 4.2.1.6 Chat với bạn bè

1 : Chứa lời mời đã gửi, khi nhấn vào sẽ hiện giống như Hình 4.2.1.7

2: Chứa lời mời đã nhận, khi nhấn vào sẽ hiện giống như Hình 4.2.1.8

3: Chứa danh sách bạn bè, khi nhấn vào sẽ hiện giống Hình 4.2.1.8

4: Đoạn chat với bạn bè

5: Gửi emoij

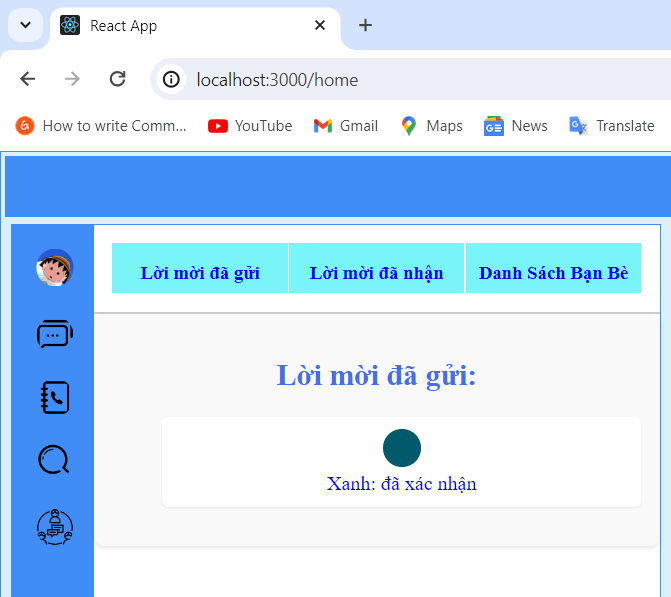
6: Nhập text

7: Chọn file/ ảnh/ video

8: Nút gửi tin nhắn

9: Tin nhắn của bạn bè

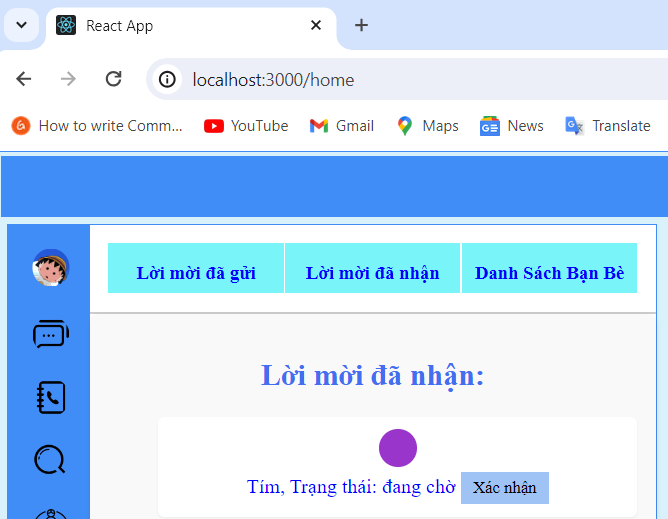
10: Tin nhắn của mình



1

Hình 4.2.1.7 Lời mời đã gửi

1: Lời mời đã gửi kèm trạng thái



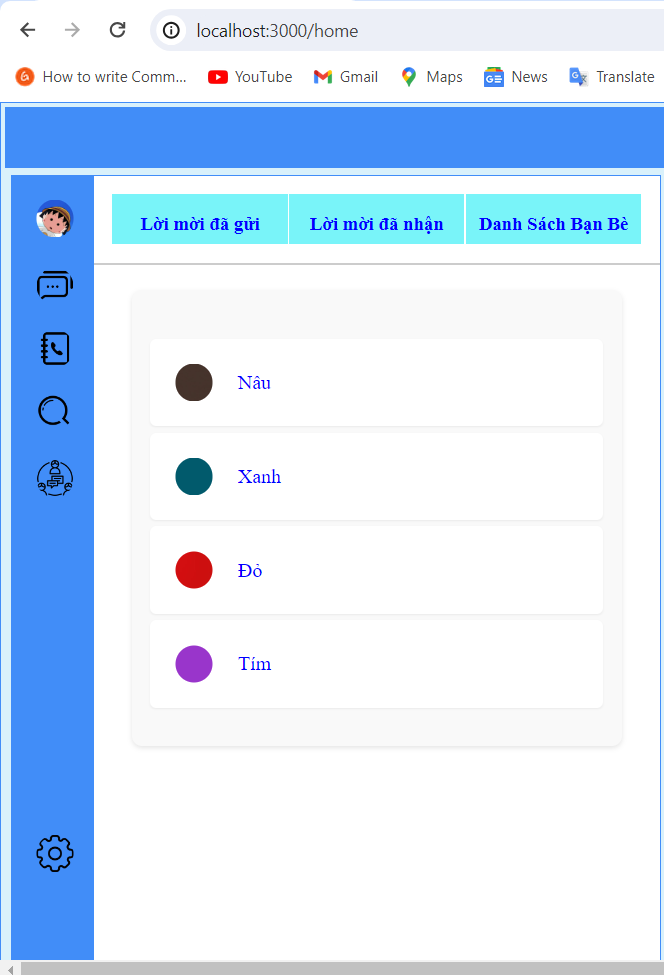
1

2

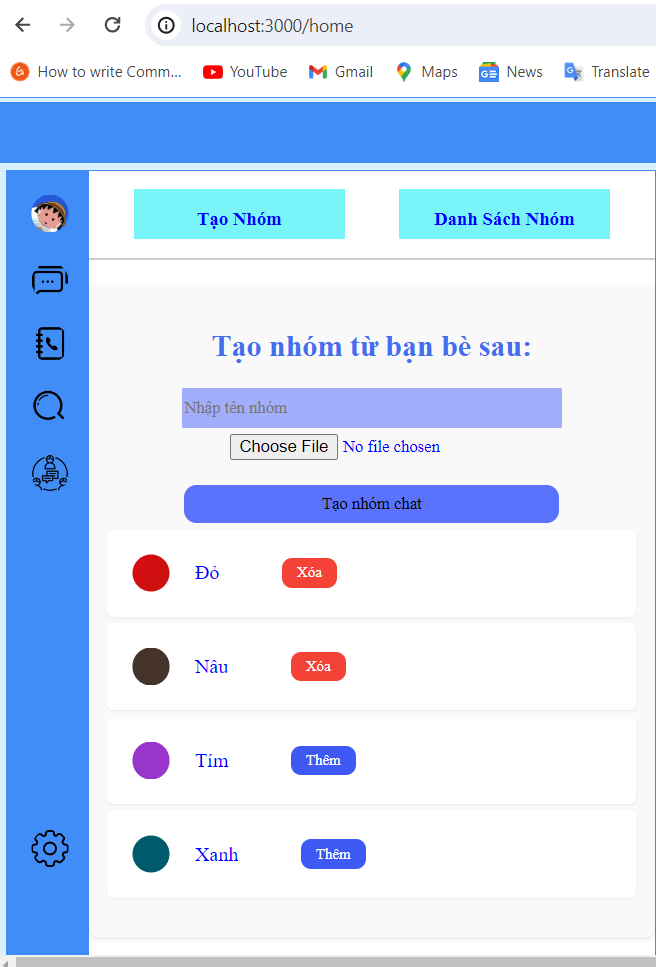
Hình 4.2.1.8 Xác nhận lời mời

1: Lời mời nhận được kèm trạng thái

2: Nếu đồng ý thì nhấn xác nhận



Hình 4.2.1.9 Danh sách bạn bè



4

3

2

1

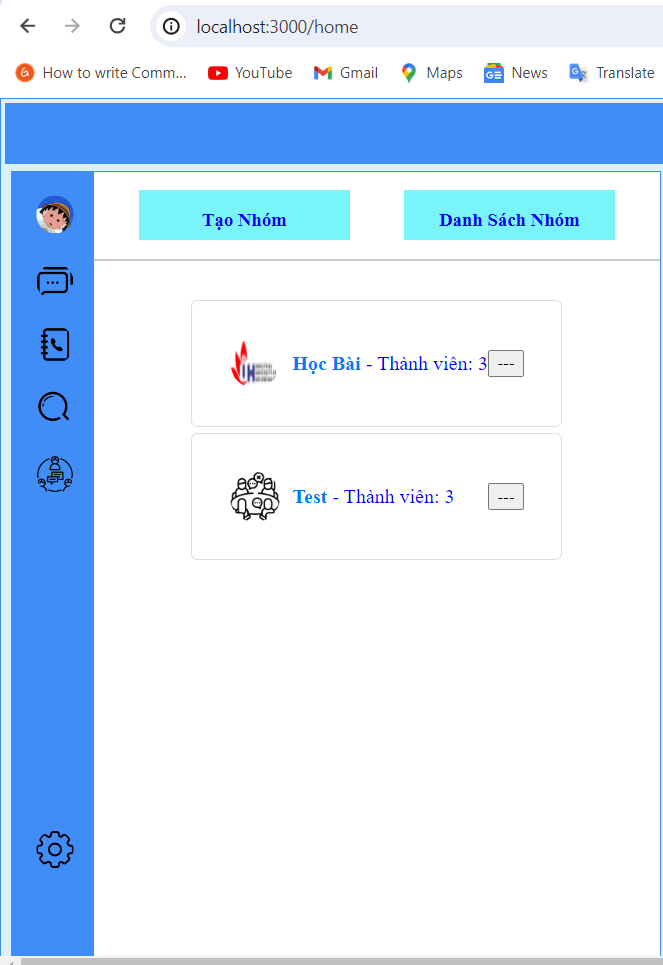
Hình 4.2.1.10 Tạo nhóm

1: Chọn thành viên

2:Nhập tên nhóm

3: Chọn ảnh đại diện nhóm

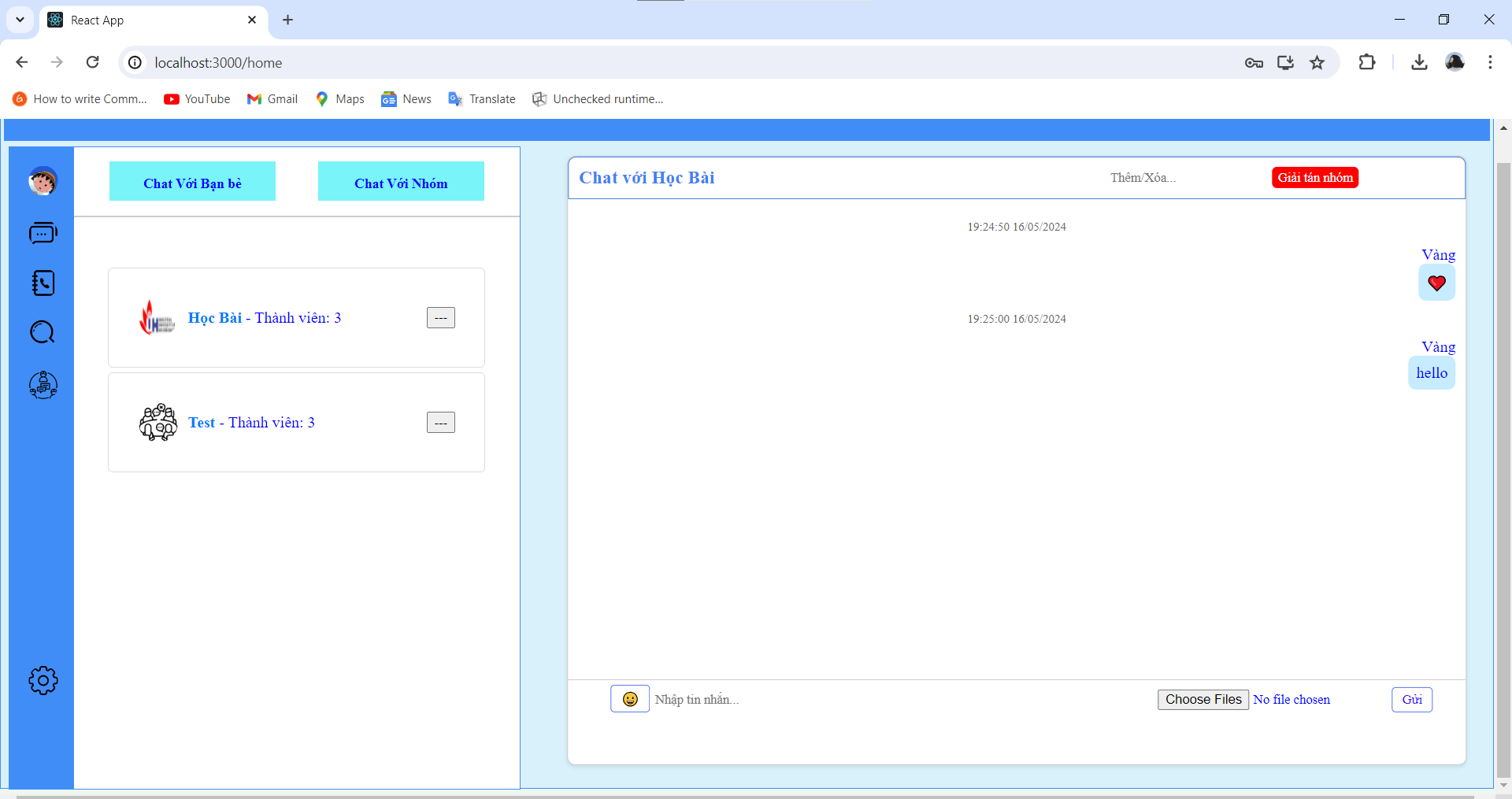
4: Nút tạo nhóm



1

Hình 4.2.1.11 Danh sách nhóm

1: Danh sách nhóm



3

2

1

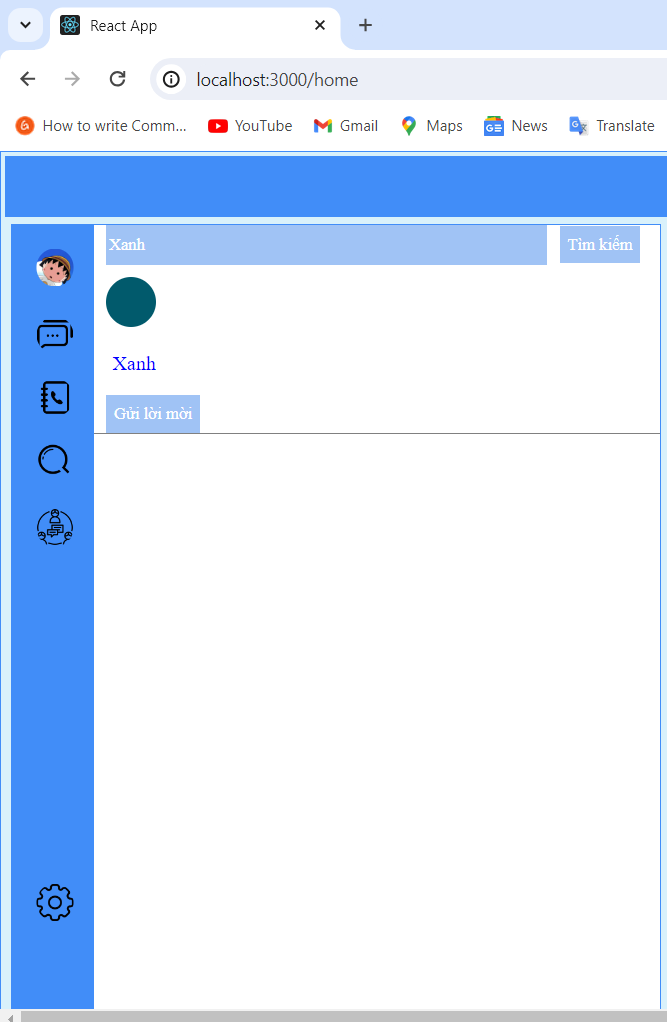
Hình 4.2.1.12 Chat với nhóm

1: Nút chat với nhóm

2: Thêm / xóa thành viên

3: Giải tán nhóm

Những chức năng còn lại tương tự với chat với bạn bè



3

2

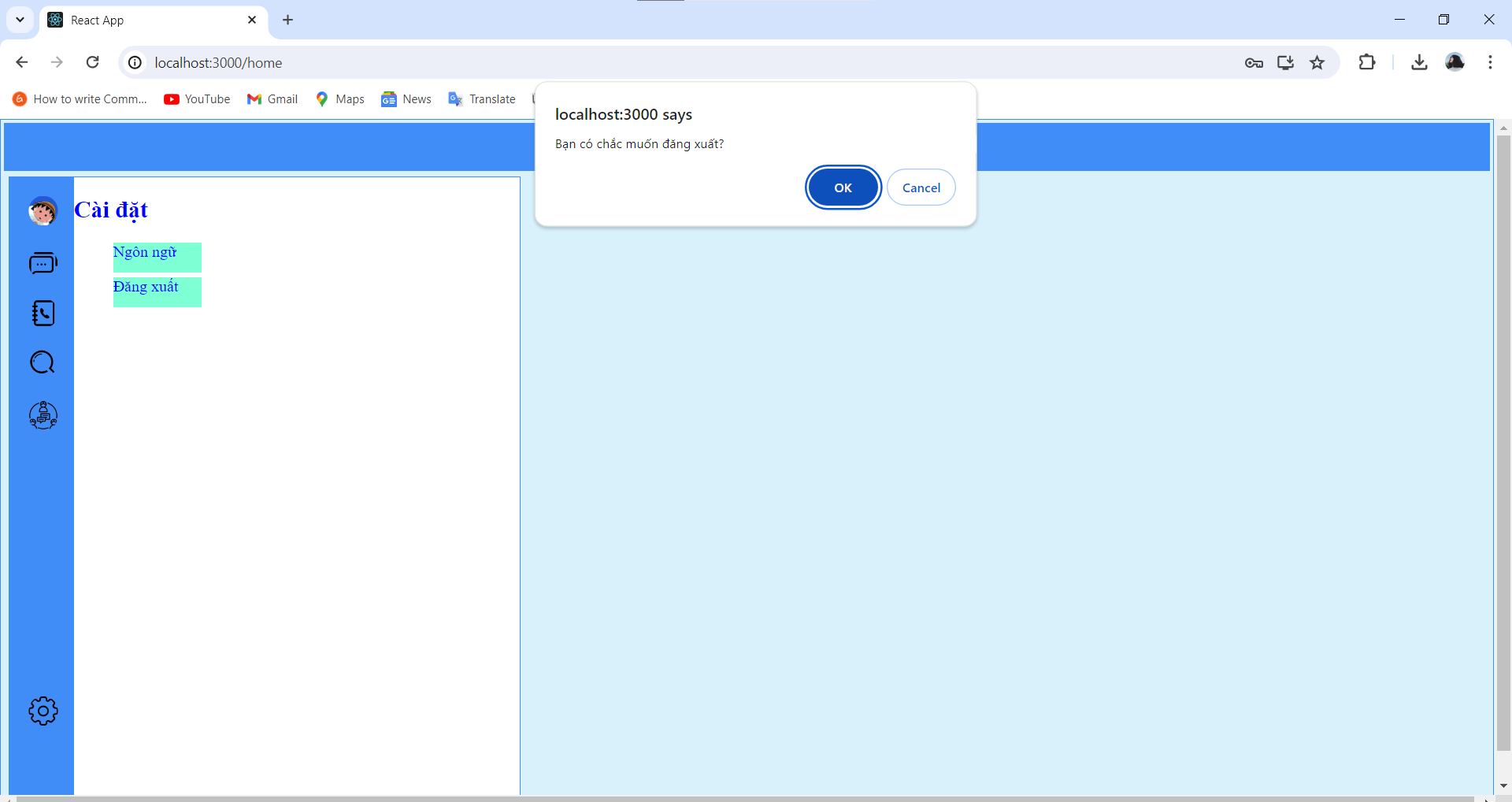
1

Hình 4.2.1.13 Tìm kiếm bạn bè

1: Nhập tên người bạn muốn tìm kiếm

2: Nút kìm kiếm

3: Kết quả kèm theo nút gửi lời mời



2

1

Hình 4.2.1.14 Đăng xuất

1: Chọn đăng xuất

2: Nhấn OK hoặc Cancel

### Giao diện app

A screenshot of a phone

Description automatically generated

Hình 4.2.1.1 Đăng nhập

1: Nhấn để đăng nhập

2: Nhấn để đăng ký

A screenshot of a phone

Description automatically generated

8

7

6

5

4

3

2

1

Hình 4.2.2.2. Đăng kí

1: Quay lại trang đăng nhập

2: Nhập tên người dùng

3 : Nhập email

4 : Nhập mật khẩu

5 : Nhập lại mật khẩu đã nhập

6 : Chọn ngày, tháng , năm sinh

7 : Chọn giới tính

A screenshot of a phone

Description automatically generated

3

2

1

Hình 4.2.2.3 Chọn avatar

1 : Quay trở lại phần nhập thông tin lúc đăng ký

2 : Chọn avatar

3 : Chuyển tới trang tiếp theo

A screenshot of a phone

Description automatically generated

5

4

3

2

1

Hình 4.2.2.4 Giao diện đăng nhập

1 : Nhập email

2 : Nhập mật khẩu

3: Nhấn để đăng nhập

4: Nhấp vào nếu quên mặt khẩu

5: Nhấn đề đăng nhập và sử dụng ứng dụng

A screenshot of a phone

Description automatically generated

2

1

Hình 4.2.2.5 giao diện nếu bạn quên mặt khẩu

1 : Nhập email lúc đăng ký

2 : chọn vào để reset mặt khẩu

A black rectangular frame with a white screen

Description automatically generated

4

3

2

1

Hình 4.2.2.6 Giao diện chat

1: Tìm kiếm các người dùng khác

2 : Hiện thị các đoạn chat của bạn với người dùng khác

3 : Nhấn chọn danh bạ

4 : Nhấn chọn trang cá nhân

A screenshot of a phone

Description automatically generated

3

2

1

Hình 4.2.2.7 Giao diện danh bạ

1 : Trang hiển thị bạn bè

2 : Trang hiển thị các nhóm đã tạo và tham gia

3 : Trang hiển thị lời mời kết bạn

A screenshot of a chat

Description automatically generated

1

2

Hình 4.2.2.8 Trang hiển thị các lời mời kết bạn

1: Quay lại hình 4.2.2.7

2: Xác nhận kết bạn

A screenshot of a phone

Description automatically generated

2

1

Hình 4.2.2.9 Nhóm

1 : Tạo nhóm

2 : Nhấn vào sẽ chuyển qua group chat

A screenshot of a chat

Description automatically generated

3

2

1

Hình 4.2.2.10 Tạo Nhóm

1: Chọn các thành viên sẽ gia nhập nhóm

2 : Nhập tên nhóm

3 : Tạo nhóm

A screenshot of a phone

Description automatically generated

5

4

3

2

1

Hình 4.2.2.11 Giao diện group chat

1: Thêm và xóa thành viên

2 : Soạn tin nhắn

3 : Gửi tin nhắn

4 : Gửi ảnh

5 : Gửi file

A screenshot of a phone

Description automatically generated

3

2

1

Hình 4.2.2.11 Giao diện chat đơn

1 : Soạn tin nhắn sao khi soạn sẽ hiển thị nút gửi

2 : Gửi ảnh

3 : Gửi file

A screenshot of a phone

Description automatically generated

1

Hình 4.2.2.12 Giao diện cá nhân

1: Chuyển sang trang cá nhân hiển thị thông tin

A screenshot of a phone

Description automatically generated

1

Hình 4.2.2.13 Giao diện cá nhân

1 : chuyển sang trang thông tin cá nhân

Hình 4.2.2.13 Giao diện thông tin cá nhân

A screenshot of a phone

Description automatically generated

1

Hình 4.2.2.14 Giao diện thông tin cá nhân

1 : Chỉnh sửa các thông tin cá nhân

A screenshot of a computer

Description automatically generated

4

3

2

1

Hình 4.2.2.15 Giao diện thông tin cá nhân

1: Chỉnh sửa tên

2: Chỉnh sửa giới tính

3 : Chỉnh sửa ngày sinh, tháng , năm sinh

4: Cập nhật thông tin đã sửa

Kế hoạch và hiện thực kiểm thử hệ thống

### Kế hoạch kiểm thử

**Bước 1:** Lập kế hoạch và kiểm soát việc kiểm thử

**1. Xác định phạm vi, rủi ro và mục đích kiểm thử:**

* **Phạm vi:** Ứng dụng chat bao gồm các chức năng đăng ký,đăng nhập tài khoản, kết bạn, chat đơn (text, video, file), chat nhóm.
* **Rủi ro:** Tham khảo phần "Xác định rủi ro" trong các phản hồi trước (2. Kết bạn, 3. Chat đơn, 4. Chat nhóm).
* **Mục đích:** Đảm bảo ứng dụng chat hoạt động ổn định, đầy đủ chức năng và mang lại trải nghiệm tốt cho người dùng.

**2. Xác định cách tiếp cận kiểm thử:**

* **Kiểm thử hộp đen:** Kiểm thử dựa trên giao diện người dùng mà không quan tâm đến cấu trúc bên trong của ứng dụng.
* **Kiểm thử hộp trắng:** Kiểm thử dựa trên cấu trúc bên trong của ứng dụng, bao gồm mã nguồn, cơ sở dữ liệu, v.v.
* **Kiểm thử thủ công:** Kiểm thử viên thực hiện kiểm thử bằng tay.
* **Kiểm thử tự động:** Sử dụng công cụ tự động để thực hiện các bài kiểm thử.

**3. Xác định chiến lược kiểm thử:**

* **Mục tiêu kiểm thử:**
  + Đảm bảo 100% chức năng cơ bản hoạt động chính xác.
  + Phát hiện và sửa lỗi tối đa.
  + Nâng cao trải nghiệm người dùng.
* **Phương pháp kiểm thử:**
  + Kiểm thử đơn vị: Kiểm thử từng thành phần riêng lẻ của ứng dụng.
  + Kiểm thử tích hợp: Kiểm thử sự tương tác giữa các thành phần của ứng dụng.
  + Kiểm thử hệ thống: Kiểm thử toàn bộ ứng dụng trong môi trường thực tế.
  + Kiểm thử hồi quy: Kiểm thử lại các chức năng sau khi sửa lỗi.
  + Kiểm thử hiệu suất: Đánh giá hiệu suất của ứng dụng dưới tải cao.
  + Kiểm thử bảo mật: Đánh giá khả năng chống tấn công của ứng dụng.
  + Kiểm thử khả năng sử dụng: Đánh giá mức độ dễ sử dụng của ứng dụng.
* **Tổng thời gian và nguồn lực:**
  + 3 tuần: Phân tích yêu cầu, thiết kế test case, thực thi test case, đánh giá kết quả.
  + 2 tester: Thực hiện kiểm thử thủ công và tự động.
  + 1 máy chủ test: Cài đặt và cấu hình ứng dụng để kiểm thử.
* **Môi trường test:**
  + Hệ điều hành: Windows, Android.

**4. Xác định nguồn lực cần thiết:**

* **Nhân lực:**
  + 2 tester
  + 2 developer (hỗ trợ cài đặt và cấu hình môi trường test)
* **Phần cứng:**
  + 2 máy tính
  + 1 điện thoại
* **Phần mềm:**
  + Hệ điều hành: Windows, Android.
  + Công cụ kiểm thử:
* **Môi trường test:**
  + Môi trường test thực tế

**5. Lên lịch kiểm thử:**

* **Tuần 1:**
  + Phân tích yêu cầu và thiết kế test case cho chức năng đăng ký,đăng nhập tài khoản.
  + Thực thi test case cho chức năng đăng ký tài khoản.
  + Đánh giá kết quả và báo cáo lỗi.
* **Tuần 2:**
  + Phân tích yêu cầu và thiết kế test case cho chức năng chat đơn.
  + Thực thi test case cho chức năng chat đơn.
  + Đánh giá kết quả và báo cáo lỗi.
* **Tuần 3:**
  + Phân tích yêu cầu và thiết kế test case cho chức năng chat nhóm.
  + Thực thi test case cho chức năng chat nhóm.
  + Đánh giá kết quả và báo cáo lỗi.
  + Kiểm thử hồi quy cho tất cả các chức năng.

**Bước 2: Phân tích và Thiết kế**

**1. Xác định điều kiện kiểm thử:**

* Liệt kê các điều kiện tiên quyết trước khi thực hiện kiểm thử cho từng chức năng.
* Ví dụ:
  + Đăng ký tài khoản: Người dùng cần có kết nối mạng, địa chỉ email hợp lệ, v.v.
  + Kết bạn: Người dùng cần đã đăng ký tài khoản, v.v.
  + Chat đơn: Người dùng cần đã kết bạn với người khác, v.v.
  + Chat nhóm: Người dùng cần được thêm vào nhóm, v.v.

**2. Thiết kế trường hợp kiểm thử (test case):**

* Viết test case cho từng chức năng, bao gồm:
  + Mô tả test case: Nêu rõ mục đích và nội dung của test case.
  + Dữ liệu đầu vào: Liệt kê dữ liệu cần sử dụng để thực hiện test case.
  + Bước kiểm tra: Mô tả từng bước thực hiện để kiểm tra chức năng.
  + Kỹ vọng kết quả: Liệt kê kết quả mong đợi sau khi thực hiện test case.
  + Kết quả thực tế: Ghi lại kết quả thực tế thu được khi thực hiện test case.
  + Đánh giá: Xác định test case thành công hay thất bại.
* Sử dụng bảng tính hoặc công cụ quản lý test case để lưu trữ và quản lý test case hiệu quả.

**3. Đánh giá tính khả thi:**

* Phân tích xem các yêu cầu và chức năng của ứng dụng có thể kiểm thử được hay không.
* Xác định những hạn chế hoặc rủi ro tiềm ẩn trong quá trình kiểm thử.
* Đề xuất giải pháp khắc phục cho các vấn đề tiềm ẩn.

**4. Chuẩn bị môi trường test:**

* Cài đặt và cấu hình môi trường test phù hợp với yêu cầu của ứng dụng.
* Chuẩn bị dữ liệu test cần thiết cho từng chức năng.
* Đảm bảo môi trường test ổn định và sẵn sàng cho việc thực hiện kiểm thử.

**5. Xác định yêu cầu về công cụ:**

* Lựa chọn công cụ kiểm thử phù hợp với từng loại kiểm thử (ví dụ: kiểm thử thủ công, kiểm thử tự động).
* Cài đặt và cấu hình các công cụ kiểm thử đã chọn.
* Đào tạo tester sử dụng các công cụ kiểm thử hiệu quả.

**Bước 3: Thực thi Kiểm thử Ứng dụng Chat**

**1. Chuẩn bị dữ liệu kiểm thử:**

* Tạo dữ liệu test cho từng chức năng, bao gồm dữ liệu hợp lệ và không hợp lệ.
* Ví dụ:
  + Đăng ký tài khoản: Dữ liệu hợp lệ (email), dữ liệu không hợp lệ (email không hợp lệ).
  + Chat đơn: Dữ liệu hợp lệ (tin nhắn văn bản, hình ảnh, tệp tin), dữ liệu không hợp lệ (tin nhắn trống, hình ảnh/tệp tin không hợp lệ).
  + Chat nhóm: Dữ liệu hợp lệ ,dữ liệu không hợp lệ ,Không đọc được tin nhắn, tin nhắn gửi thất bại.

**2. Thực hiện kiểm thử:**

* **Kiểm thử thủ công:**
  + Thực hiện từng bước kiểm tra theo các trường hợp kiểm thử đã thiết kế trong Bước 2.
  + Ghi nhận kết quả thực tế thu được cho từng bước kiểm tra.
  + So sánh kết quả thực tế với kết quả mong đợi và đánh giá test case là Passed/Failed.
  + Báo cáo lỗi cho những test case có kết quả thực tế không khớp với kết quả mong đợi.

**3. Chạy lại test:**

* Chạy lại các test case bị failed để xác nhận lỗi đã được sửa hay chưa.
* Cập nhật kết quả kiểm thử và báo cáo lỗi sau khi chạy lại test.

**Bước 4: Đánh giá kết quả thực thi và báo cáo kết quả cho Kiểm thử Ứng dụng Chat**

**Mục đích:** Đánh giá toàn bộ quá trình kiểm thử ứng dụng chat, bao gồm đánh giá kết quả kiểm thử, xác định yêu cầu thay đổi và thống kê số liệu liên quan.

**Tiêu chí đánh giá:**

* **Số lượng test case Passed:**
  + Đánh giá tỷ lệ phần trăm test case Passed so với tổng số test case được thực thi.
  + Mức độ Passed cao cho thấy chất lượng ứng dụng tốt và ít lỗi.
* **Hoàn thành đúng deadline:**
* Đánh giá việc hoàn thành công việc kiểm thử đúng theo thời gian đề ra trong kế hoạch.
* Việc hoàn thành đúng deadline đảm bảo tiến độ chung của dự án.

**Kết thúc hoạt động kiểm thử:**

* **Đối chiếu kết quả thực thi test case với các tiêu chí kết thúc kiểm thử đã đề ra trong Bước 1.**
  + Phân tích xem kết quả có đạt được các tiêu chí đề ra hay không.
* **Đánh giá xem có cần thiết phải thực hiện thêm kiểm thử hay điều chỉnh các tiêu chí kết thúc kiểm thử.**
  + Căn cứ vào kết quả đánh giá, quyết định có cần kiểm thử thêm hay không và điều chỉnh các tiêu chí cho phù hợp.
* **Viết báo cáo tóm tắt hoạt động kiểm thử và kết quả kiểm thử cho các bên liên quan.**
  + Báo cáo bao gồm:
    - Mục tiêu và phạm vi kiểm thử.
    - Phương pháp và công cụ kiểm thử sử dụng.
    - Kết quả test case chi tiết (Passed/Failed).
    - Phân tích lỗi và đề xuất sửa lỗi.
    - Đánh giá chung về chất lượng ứng dụng.
    - Kết luận và đề xuất.

**Bước 5: Đóng hoạt động kiểm thử cho Kiểm thử Ứng dụng Chat**

**Mục đích:** Kết thúc hoạt động kiểm thử ứng dụng chat và bàn giao ứng dụng cho khách hàng.

**Hoạt động:**

**1. Xác nhận hoàn thành các cam kết:**

* Kiểm tra lại xem tất cả các chức năng đã được cam kết trong yêu cầu ban đầu đã được hoàn thiện và kiểm thử đầy đủ.
* Xác nhận với bộ phận phát triển rằng tất cả các lỗi nghiêm trọng đã được sửa chữa.

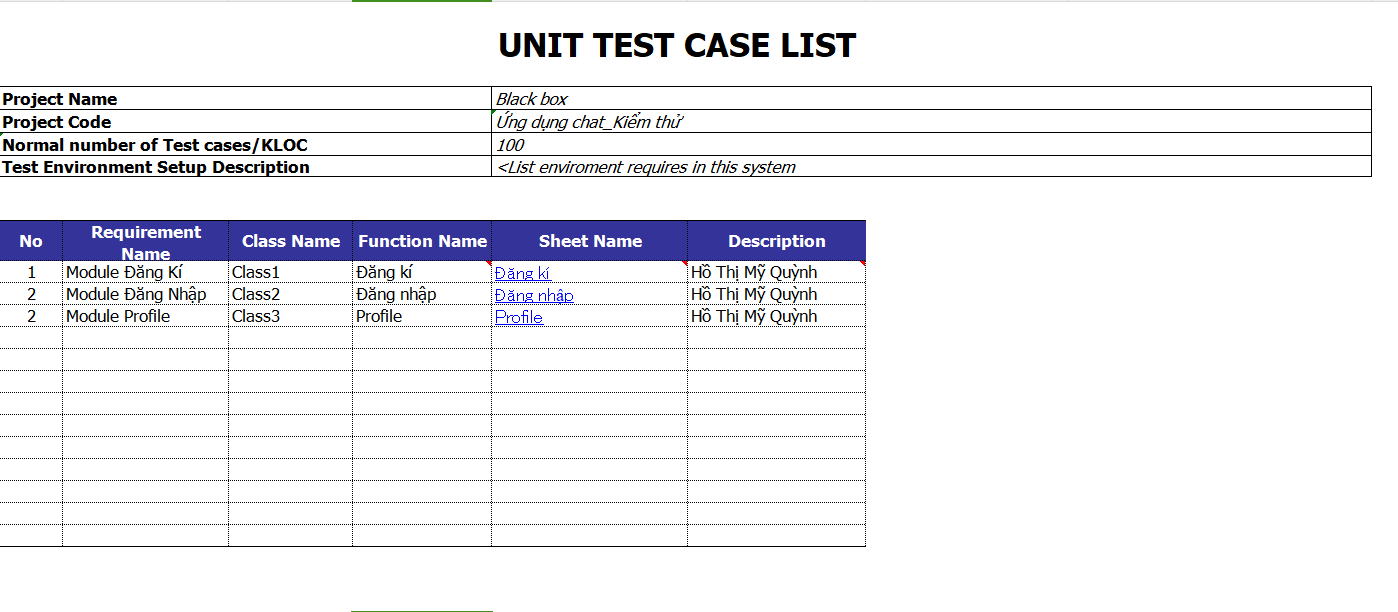
**2. Lưu trữ tài liệu kiểm thử:**

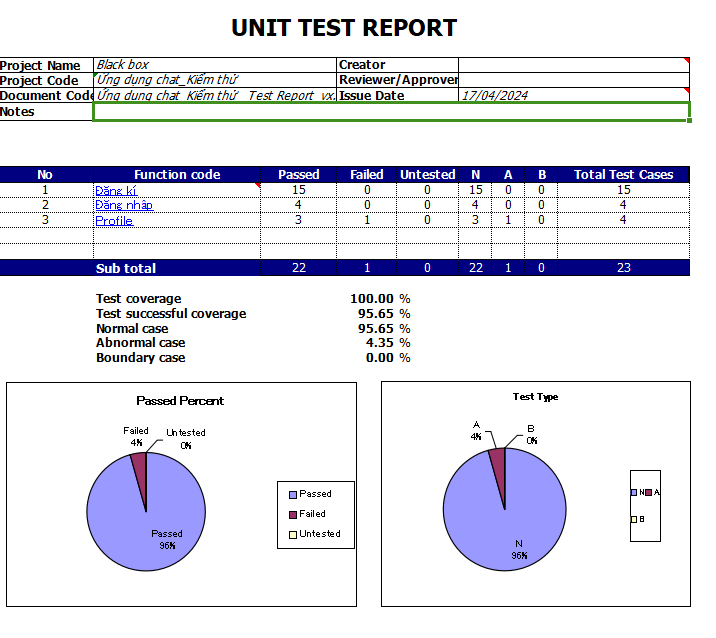
* Đóng gói tất cả các tài liệu kiểm thử, bao gồm:
  + Kế hoạch kiểm thử
  + Trường hợp kiểm thử
  + Báo cáo kết quả kiểm thử
  + Báo cáo lỗi
* Lưu trữ tài liệu kiểm thử ở một nơi an toàn và dễ truy cập để sử dụng cho các mục đích sau này hoặc các dự án tương tự.

**3. Đánh giá quá trình kiểm thử:**

* Nhóm họp rút kinh nghiệm để đánh giá hiệu quả của hoạt động kiểm thử.
* Phân tích những điểm mạnh, điểm yếu của quy trình kiểm thử hiện tại.
* Xác định những bài học kinh nghiệm cần áp dụng cho các dự án trong tương lai.

### Kiểm thử hệ thống





# : KẾT LUẬN

## Kết quả đạt được

Đã hoàn thành các chức năng đã đặt ra như:

Người dùng có thể :

- Thực hiện đăng ký, đăng nhập tài khoản, sửa đổi thông tin cá nhân và ảnh đại diện

- Thực hiện chat đơn, chat theo nhóm, có thể gửi kèm file, hình ảnh, video và emoij… xoá, thu hồi .

- Có thể kết bạn, tạo và tham gia các group chat

## Hạn chế của đồ án

* Chưa gọi, gọi video được.

## Hướng phát triển

Trong tương lai sẽ tiếp tục hoàn thiện project, sửa chữa các lỗi và thêm vào 1 số chức năng như gọi, gọi video , và cho người dùng chia sẻ nhật kí của mình.

# 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>