BỘ CÔNG THƯƠNG

TRƯỜNG CAO ĐẲNG KỸ THUẬT CAO THẮNG

KHOA CÔNG NGHỆ THÔNG TIN

... ☙ 🕮 ❧ ...

ĐỒ ÁN TỐT NGHIỆP

WEBSITE TRAO ĐỔI THÔNG TIN TRỰC TUYẾN SEND IT

Giảng viên hướng dẫn: Trần Quang Khải

Sinh viên thực hiện:

Nguyễn Thế Vinh MSSV**: 0306191487**

Trương Văn Chiến MSSV**: 0306191107**

Lớp: **CĐTH 19 PMD – CDTH 19 PMB**

Khóa: **2019 – 2022**

TP. HỒ CHÍ MINH, tháng 7 năm 2022

LỜI CẢM ƠN

Chúng em xin chân thành cảm ơn quý thầy cô trong khoa Công Nghệ Thông Tin trường cao đẳng Kỹ Thuật Cao Thắng, để có thể hoàn thành được đồ án tốt nghiệp này là kết quả nền tảng từ kiến thức của sự giảng dạy tận tình của các quý thầy cô, trang bị cho chúng em những kiến thức quý báu qua các bài học, bài giảng và giúp chúng em vận dụng những kiến thức đã học vào công việc khi ra trường.

Đặc biệt với tấm lòng biết ơn sâu sắc cảm ơn thầy Trần Quang Khải đã trực tiếp hướng dẫn, đóng góp ý kiến, giúp đỡ chúng em hoàn thành đồ án tốt nghiệp này.

Mặc dù đã cố gắng nỗ lực thực hiện đồ án này nhưng với kiến thức và thời gian có hạn, chắc hẳn đồ án này sẽ không tránh khỏi sai sót. Chúng em rất mong nhận được sự thông cảm và đóng góp ý kiến của thầy cô để chúng em có thể hoàn thiện tốt hơn cho sau này.

Chúng em xin chân thành cám ơn !

Tp. Hồ Chí Minh, tháng 7 năm 2022

Nhóm sinh viên thực hiện

Nguyễn Thế Vinh & **Trương Văn Chiến**

LỜI NÓI ĐẦU

Thời gian gần đây, tình hình dịch bệnh COVID-19 diễn ra rất căng thẳng và phức tạp. Sự bùng phát dịch COVID-19 đã mang lại những thách thức chưa từng có, đem đến nhiều hệ lụy cả về kinh tế lẫn con người. Việc giao tiếp, trao đổi thông tin của tất cả mọi người tất cả đều quy về trao đổi trực tuyến.

Cũng trong khoảng thời gian dịch bệnh đang hoành hành và diễn ra hết sức phức tạp này chúng em nhận thấy rằng trao đổi thông tin trực tuyến giữa người với người là vô cùng quan trọng không chỉ là trao đổi bằng tin nhắn mà còn gặp mặt trực tuyến qua hình thức như cuộc gọi/meeting.

Xuất phát từ những nhu cầu thực tế, với mục đích phục vụ cho việc trao đổi thông tin trong thời gian mọi thứ bị đóng băng vì dịch bệnh. Chúng em quyết định chọn đề tài "Phát triển nền tảng trao đổi thông tin trực tuyến với công nghệ ReactJS, MongoDB (Mongoose), NodeJs và SocketIO", nhằm xây dựng một nền tảng ứng dụng có thể hỗ trợ cho việc trao đổi thông tin, gặp mặt được tốt hơn và thật đơn giản.

Nhóm sinh viên thực hiện đề tài :

Nguyễn Thế Vinh
MSSV: 0306191487
Điện Thoại: 0971707353
Email: 0306191487@caothang.edu.vn

Trương Văn Chiến
MSSV: 0306191107
Điện Thoại: 0358208351
Email: 0306191107@caothang.edu.vn

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

TP. HỒ CHÍ MINH, ngày.....tháng.....năm 2022

Giáo viên hướng dẫn

NHẬN XÉT CỦA GIÁO VIÊN PHẢN BIỆN

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

TP.HCM, Ngày ……. Tháng……. Năm 2022

Giáo viên phản biện

NHẬN XÉT CỦA HỘI ĐỒNG BẢO VỆ ĐỒ ÁN TỐT NGHIỆP

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

TP.HCM, Ngày ……. Tháng……. Năm 2022

Thư kí hội đồng | Chủ tịch hội đồng

MỤC LỤC

CHƯƠNG 1. GIỚI THIỆU 1
1.1. Giới thiệu về đồ án 1
1.2. Các công nghệ được ứng dụng trong đồ án 2
1.2.1. MongoDB 2
1.2.2. Mongoose 5
1.2.3. Công Nghệ ReactJS 7
1.2.4. NodeJS Platform 8
1.2.5. Framework React-Bootstrap 9
1.2.6. Material UI 11
1.2.7. Socket.io 11
CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ DỮ LIỆU 13
2.1. Đặc tả chức năng 13
2.1.1. Đối với người dùng 13
2.1.2. Đối với người quản trị 15
2.2. Use Case 16
2.2.1. Xác định tác nhân trong hệ thống 16
2.2.2. Use Case tổng quát của phân hệ User 17
2.2.3. Use Case tổng quát của phân hệ Admin 18
2.3. Cơ sở dữ liệu 19
2.3.1. Danh mục các bảng 19
CHƯƠNG 3. CÀI ĐẶT GIAO DIỆN VÀ CHƯƠNG TRÌNH 24
3.1. Cấu trúc thư mục đồ án 24
3.1.1. Cấu trúc thư mục ở Backend và Frontend 24
3.1.2. Cấu trúc thư mục Socket.io 25
3.2. Giao diện website phân hệ của User 26
3.2.1. Giao diện trang chủ website 26
3.2.2. Giao diện đăng ký 26

3.2.3. Giao diện đăng nhập 28
3.2.4. Giao diện Chính (sau khi đăng nhập thành công) 29
3.2.5. Giao diện tìm kiếm người dùng 29
3.2.6. Giao diện màn hình chính khi bạn có nhiều liên lạc 31
3.2.7. Giao diện gửi tin nhắn và emoji 32
3.2.8. Giao diện gửi hình ảnh / gửi file 33
3.2.9. Giao diện thu hồi tin nhắn, thả cảm xúc tin nhắn 35
3.2.10. Giao diện call video 37
3.2.11. Giao diện cập nhật thông tin người dùng 39
3.2.12. Giao diện thay đổi ảnh nền khung chat 41
3.2.13. Giao diện tạo nhóm chat 43
3.2.14. Giao diện trong một nhóm chat 44
3.3. Giao diện trang quản trị 45
3.3.1. Giao diện trang chủ 45
CHƯƠNG 4: TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN 47
4.1. Kết quả đạt được 47
4.2. Phần hạn chế của đề tài 47
4.3. Hướng phát triển 48
TÀI LIỆU THAM KHẢO 49

DANH MỤC CÁC KÝ TỰ, CÁC TỪ VIẾT TẮC

STT	Từ viết tắt	Ý nghĩa
1	HTML	Hypertext Markup Language
2	CSS	Cascading Style Sheets
3	RDBMS	Relational Database Management System
4	API	Application Programming Interface
5	IT	Information Technology
6	JSX	Javascript + XML
7	UI	User Interface
8	XML	Extensible Markup Language
9	JS	JavaScript
10	npm	Node package manager
11	JSON	JavaScript Object Notation
12	RAM	Random Access Memory
13	DOM	Document Object Model

DANH MỤC CÁC BẢNG

Bảng 1: Bảng Users.....19
Bảng 2: Bảng Conversations.....20
Bảng 3: Bảng Messages.....23

DANH MỤC HÌNH ẢNH

Hình 1: Tổng quan về MongoDB 2
Hình 2: Ưu điểm của MongoDB 4
Hình 3: Mongoose logo 5
Hình 4: ReactJS logo 7
Hình 5: NodeJS Logo 8
Hình 6: Logo React - Boostrap 10
Hình 7: SocketIO Logo 11
Hình 8: Tác nhân trrong hệ thống 16
Hình 9: Use Case User tổng quát 17
Hình 10: Use Case Admin tổng quát 18
Hình 11: Cơ sở dữ liệu 19
Hình 12. Cấu trúc thư mục Backend Hình 13: Cấu trúc thư mục Frontend 24
Hình 14: Cấu trúc thư mục SocketIO 25
Hình 15: Giao diện trang chủ website 26
Hình 16: Giao diện đăng ký 27
Hình 17: Giao diện đăng nhập 28
Hình 18: Giao diện màn hình chính (đăng nhập thành công) 29
Hình 19: Giao diện tìm kiếm người dùng 30
Hình 20: Giao diện màn hình chính(khi chưa chọn liên hệ) 31
Hình 21: Giao diện chat với bạn bè 31
Hình 22: Giao diện chọn emoji 32
Hình 23: Giao diện thao tác gửi ảnh 33
Hình 24: Giao diện gửi ảnh thành công 34
Hình 25: Giao diện gửi ảnh thành công 35
Hình 26: Giao diện các bước thu hồi và thả cảm xúc tin nhắn 36

Hình 27: Giao diện nút bắt đầu cuộc gọi 37
Hình 28: Giao diện chờ khi thực hiện cuộc gọi 37
Hình 29: Giao diện nhận thông báo khi có người gọi đến 38
Hình 30: Giao diện đang thực hiện cuộc gọi.....38
Hình 31: Giao diện các chức năng của cập nhật 39
Hình 32: Giao diện cập nhật thông tin người dùng.....40
Hình 33: Giao diện các chức năng của cập nhật 41
Hình 34: Giao diện chi tiết đơn hàng thành công 42
Hình 35: Giao diện tạo nhóm chat 43
Hình 36: Giao diện chat của nhóm chat.....44
Hình 37: Giao diện tất cả User đã tham gia 45
Hình 38: Giao diện thông tin chi tiết của User 46

CHƯƠNG 1. GIỚI THIỆU

1.1. Giới thiệu về đồ án

Thời gian gần đây, tình hình dịch bệnh COVID-19 diễn ra rất căng thẳng và phức tạp. Sự bùng phát dịch COVID-19 trong giai đoạn từ 2020 đến nay đã hơn 2 năm nhưng vẫn chưa hề chấm dứt hoàn toàn, đại dịch đã mang lại những thách thức chưa từng có, đem đến nhiều hệ lụy cả về kinh tế lẫn con người. Về yếu tố con người, COVID-19 đã làm ảnh hưởng cả sức khỏe, tinh thần, sự gắn kết với nhau giữa chúng ta và người thân. Việc giao tiếp, trao đổi thông tin của không chỉ gia đình mà còn tất cả mọi người đều quy về hình thức trao đổi trực tuyến.

Cũng trong khoảng thời gian dịch bệnh đang hoành hành và diễn ra hết sức phức tạp này chúng em nhận thấy rằng trao đổi thông tin trực tuyến giữa người với người là vô cùng quan trọng không chỉ là trao đổi bằng tin nhắn mà còn gặp mặt trực tiếp qua cuộc gọi/meeting.

Xuất phát từ những nhu cầu thực tế, với mục đích phục vụ cho việc trao đổi thông tin trong thời gian mọi thứ bị đóng băng vì dịch bệnh. Chúng em quyết định chọn đề tài "Phát triển nền tảng trao đổi thông tin trực tuyến với công nghệ ReactJS, MongoDB (Mongoose), NodeJs và SocketIO", nhằm xây dựng một nền tảng ứng dụng có thể hỗ trợ cho việc trao đổi thông tin, gặp mặt được tốt hơn và thật đơn giản.

1.2. Các công nghệ được ứng dụng trong đồ án

1.2.1. MongoDB

1.2.1.1. *Tổng quan về MongoDB*

Hình 1: Tổng quan về MongoDB

MongoDB là một hệ quản trị cơ sở dữ liệu NoSQL (cơ sở dữ liệu phi quan hệ) mã nguồn mở đa nền tảng viết bằng C++. Bản ghi trong MongoDB được lưu trữ dưới dạng một dữ liệu văn bản (Document), là một cấu trúc dữ liệu bao gồm các cặp giá trị và trường tương tự như các đối tượng JSON. MongoDB được phát triển bởi MongoDB Inc. dưới dạng giấy phép Server-Side Public License (SSPL).

1.2.1.2. *Lợi ích khi sử dụng MongoDB*

Với NoSQL - cụ thể là MongoDB, bạn có thể mở rộng dữ liệu mà không lo tới những việc như tạo khóa ngoại, khóa chính, kiểm tra ràng buộc. v.v ...NoSQL bỏ qua tính toàn vẹn của dữ liệu và transaction để đổi lấy hiệu suất nhanh và khả năng mở rộng.

NoSQL được sử dụng ở rất nhiều công ty, tập đoàn lớn, ví dụ như FaceBook sử dụng Cassandra do FaceBook phát triển, Google phát triển và sử dụng BigTable. Các collection trong MongoDB được cấu trúc rất linh hoạt, cho phép các dữ liệu lưu trữ không cần tuân theo một cấu trúc nhất định.

Thông tin liên quan được lưu trữ cùng nhau để truy cập, truy vấn nhanh thông qua ngôn ngữ truy vấn MongoDB

1.2.1.3. *Ưu điểm của MongoDB*

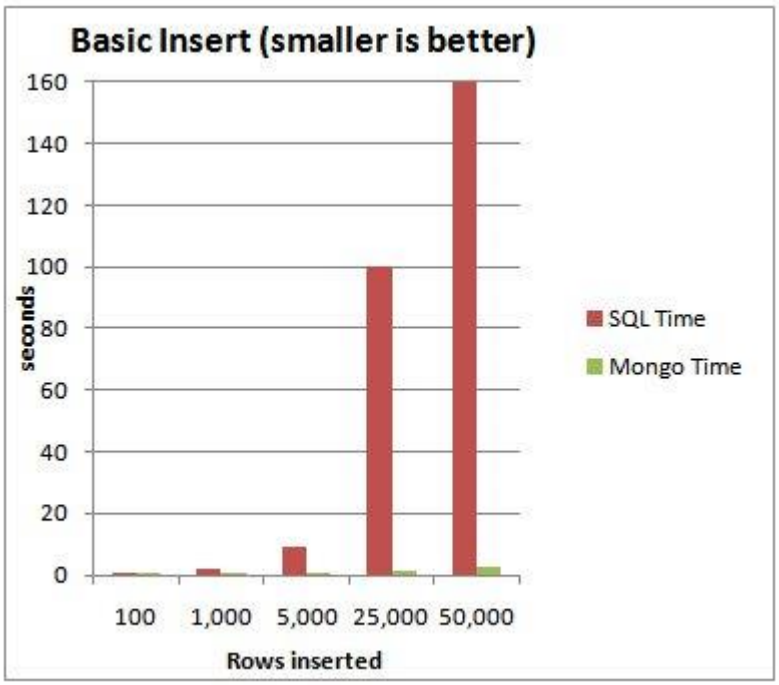
Do MongoDB sử dụng lưu trữ dữ liệu dưới dạng Document JSON nên mỗi một collection sẽ có các kích cỡ và các document khác nhau, linh hoạt trong việc lưu trữ dữ liệu, nên bạn muốn gì thì cứ insert vào thoải mái.

Dữ liệu trong MongoDB không có sự ràng buộc lẫn nhau, không có join như trong RDBMS nên khi insert, xóa hay update nó không cần phải mất thời gian kiểm tra xem có thỏa mãn các ràng buộc dữ liệu như trong RDBMS.

MongoDB rất dễ mở rộng (Horizontal Scalability). Trong MongoDB có một khái niệm cluster là cụm các node chứa dữ liệu giao tiếp với nhau, khi muốn mở rộng hệ thống ta chỉ cần thêm một node với vào cluster: Trường dữ liệu "_id" luôn được tự động đánh index (chỉ mục) để tốc độ truy vấn thông tin đạt hiệu suất cao nhất.

Khi có một truy vấn dữ liệu, bản ghi được cached lên bộ nhớ Ram, để phục vụ lượt truy vấn sau diễn ra nhanh hơn mà không cần phải đọc từ ổ cứng. Hiệu năng cao: Tốc độ truy vấn (find, update, insert, delete) của MongoDB nhanh hơn hẳn so với các hệ quản trị cơ sở dữ liệu quan hệ (RDBMS). Với một lượng dữ liệu đủ lớn thì thử nghiệm cho thấy

tốc độ insert của MongoDB có thể nhanh tới gấp 100 lần so với MySQL.



Hình 2: Ưu điểm của MongoDB

1.2.1.4. *Nhược điểm của MongoDB*

Một ưu điểm của MongoDB cũng chính là nhược điểm của nó. MongoDB không có các tính chất ràng buộc như trong RDBMS nên khi thao tác với mongoDB thì phải hết sức cẩn thận. Tốn bộ nhớ do dữ liệu lưu dưới dạng key-value, các collection chỉ khác về value do đó key sẽ bị lặp lại. Không hỗ trợ join nên dễ bị dữ thừa dữ liệu.

Khi insert/update/remove bản ghi, MongoDB sẽ chưa cập nhật ngay xuống ổ cứng, mà sau 60 giây MongoDB mới thực hiện ghi toàn bộ dữ liệu thay đổi từ RAM xuống ổ cứng điêù này sẽ là nhược điểm vì sẽ có nguy cơ bị mất dữ liệu khi xảy ra các tình huống như mất điện...

1.2.2. Mongoose

Hình 3: Mongoose logo

1.2.2.1. *Giới thiệu Mongoose*

Mongoose là một Object Document Mapper (ODM). Điều này có nghĩa là Mongoose cho phép bạn định nghĩa các object (đối tượng) với một Schema được định nghĩa rõ ràng, được ánh xạ tới một MongoDB document.

Mongoose cung cấp một số lượng đáng kinh ngạc các chức năng cho việc tạo ra và làm việc với các schema. Mongoose hiện có 8 SchemaTypes. Đó là:

- String
- Number
- Date
- Buffer
- Boolean
- Mixed
- ObjectId
- Array

Mỗi loại dữ liệu cho phép bạn xác định:

- Giá trị mặc định
- Hàm xác thực

- Cho biết một trường bắt buộc
- Hàm get cho phép bạn thao tác dữ liệu trước khi nó được trả về như một object
- Hàm set cho phép bạn thao tác dữ liệu trước khi nó được lưu vào cơ sở dữ liệu
- Tạo indexes cho phép dữ liệu được nạp nhanh hơn

Ngoài các tùy chọn phổ biến này, một số kiểu dữ liệu nhất định cho phép bạn tùy chỉnh thêm cách dữ liệu được lưu trữ và truy xuất từ một database. Ví dụ, một kiểu dữ liệu String cũng cho phép bạn chỉ định các tùy chọn bổ sung sau đây:

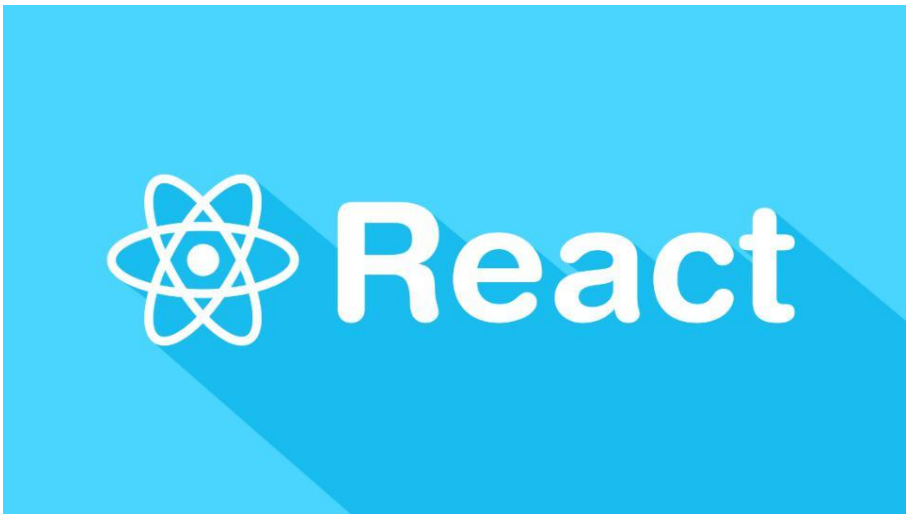
- Chuyển đổi nó thành chữ thường
- Chuyển đổi nó thành chữ in hoa
- Cắt dữ liệu trước khi lưu lại
- Một biểu thức chính quy có thể giới hạn dữ liệu được phép lưu trong quá trình xác thực
- Một enum có thể định nghĩa một danh sách các string (chuỗi) hợp lệ

Tóm tắt: Mongoose là một thư viện JavaScript cho phép bạn định nghĩa các Schema với dữ liệu được định kiểu rõ ràng. Khi một schema được định nghĩa, Mongoose cho phép bạn tạo một Model dựa trên một Schema cụ thể. Model của Mongoose sau đó được ánh xạ tới một MongoDB document thông qua định nghĩa Schema của Model.

Khi bạn đã xác định các Schema và model của mình, Mongoose chứa nhiều hàm khác nhau cho phép bạn xác thực tính hợp lệ, lưu, xóa và truy vấn dữ liệu của bạn bằng các hàm MongoDB phổ biến.

1.2.3. Công Nghệ ReactJS

1.2.3.1. *Đôi nét về ReactJS*



Hình 4: ReactJS logo

ReactJS là một thư viện Javascript mã nguồn mở hỗ trợ xây dựng các thành phần giao diện nhanh gọn và tiện lợi. Bình thường các lập trình viên sẽ nhúng Javascript vào code HTML thông qua các attribute như AngularJS nhưng với Reactjs làm việc như một thư viện cho phép nhúng HTML vào Javascript thông qua JSX. Qua đó có thể dễ dàng lồng các đoạn HTML vào trong JSX làm cho các component dễ hiểu và dễ sử dụng hơn.

1.2.3.2. *Tính năng của ReactJS*

- **JSX** – JSX là phần mở rộng cú pháp JavaScript. Không nhất thiết phải sử dụng JSX trong phát triển React.
- Các Component (Thành phần) – React là tất cả về các Component. Chúng ta cần nghĩ mọi thứ như một Component. Điều này sẽ giúp chúng ta duy trì mã khi làm việc trên các dự án quy mô lớn hơn.

1.2.3.3. *Ưu điểm và nhược điểm của ReactJS*

- **Ưu điểm:**

✓ Sử dụng DOM ảo là một đối tượng JavaScript. Điều này sẽ cải thiện hiệu suất ứng dụng, vì DOM ảo Javascript nhanh hơn DOM thông thường.

✓ Có thể được sử dụng trên máy khách và máy chủ cũng như với các khuôn khổ khác.

✓ Các mẫu thành phần và dữ liệu cải thiện khả năng đọc, giúp duy trì các ứng dụng lớn hơn.

- **Nhược điểm:**

✓ Chỉ bao gồm lớp xem của ứng dụng, do đó chúng ta vẫn cần chọn các công nghệ khác để có được bộ công cụ hoàn chỉnh để phát triển.

✓ Sử dụng khuôn mẫu nội tuyến và JSX, điều này có vẻ khó xử đối với một số nhà phát triển.

1.2.4. NodeJS Platform

1.2.4.1. *Giới thiệu NodeJS Platform*

Hình 5: NodeJS Logo

NodeJS là một nền tảng được xây dựng trên "V8 JavaScript Engine" được viết bằng C++ và JavaScript. Nền tảng này được phát triển bởi Ryan Lienhart Dahl vào năm 2009.

Node.js ra đời khi các developer đời đầu của JavaScript mở rộng nó từ một thứ bạn chỉ chạy được trên trình duyệt thành một thứ bạn có thể chạy trên máy của mình dưới dạng ứng dụng độc lập.

Cả trình duyệt JavaScript và Node.js đều chạy trên JavaScript runtime V8 engine. Công cụ này lấy code JavaScript của bạn và convert nó sang mã máy (bytecode) cho việc thực thi nhanh hơn. Mã máy là loại code thấp cấp hơn để máy tính có thể chạy mà không cần biên dịch nó.

1.2.4.2. *Tại sao lại sử dụng NodeJS Platform*

Node.js là một JavaScript runtime được build dựa trên engine JavaScript V8 của Chrome. Node.js sử dụng kiến trúc hướng sự kiện event-driven, mô hình non-blocking I/O làm cho nó nhẹ và hiệu quả hơn. Hệ thống nén của Node.js, npm, là hệ thống thư viện nguồn mở lớn nhất thế giới.

1.2.5. Framework React-Bootstrap

1.2.5.1. *Tổng quan về React-Bootstrap*

React-Bootstrap cung cấp sẵn một bộ các React component với look-and-feel của Twitter Bootstrap component từ đó giúp việc tạo dựng UI cho React app dễ dàng hơn bao giờ hết. Ví dụ khi ta mô tả một button nhỏ, text là "Something", click event trigger hàm callback "someCallback".

Hình 6: Logo React - Boostrap

1.1.5.2. Những lợi ích khi sử dụng React-Bootstrap

1. *Hạn chế sự phụ thuộc vào jQuery*

Bootstrap được đóng gói kèm theo bootstrap.js, thư viện này phụ thuộc vào jQuery và jQuery thực hiện các phép tính toán làm thay đổi DOM thực. Trong khi đó React không tương tác trực tiếp với DOM thực mà gián tiếp thông qua DOM ảo. Khi chúng ta bất cẩn để jQuery thay đổi DOM thực, React không hề biết đến sự thay đổi đó dẫn đến việc DOM ảo và DOM thực mất tính đồng bộ và phát sinh lỗi conflict. Sử dụng React-Bootstrap component chúng ta có thể tránh được vấn đề này.

2. *Cung cấp Bootstrap API hoàn thiện hơn*

Bootstrap code thường lặp lại ví dụ như ở đoạn code html trên chúng ta phải viết "btn" tới 3 lần trong "btn btn-success btn-sm" để mô tả một element "button". React quản lý toàn bộ quá trình render webpage thông qua Javascript. Vì vậy, chúng ta chỉ cần mô tả element được render như thế nào bằng một Javascript object và React sẽ làm phần việc còn lại, tạo ra element đó trong DOM và hiển thị trên trình duyệt cho chúng ta.

1.2.6. Material UI

Material UI là một thư viện các React Component đã được tích hợp thêm cả Google's Material Design. Theo như giới thiệu trên trang chủ thì được xây dựng nhờ tình cảm với React và Google's Material Design. Do đó mà phần hướng dẫn trên trang chủ của Material UI cũng đã nói nên sử dụng Material UI với React.

1.2.6.1. Lợi ích khi sử dụng Material UI

Material UI đem đến cho chúng ta và trang web của chúng ta một giao diện hoàn toàn mới, với những button, textfield, toogle... được design theo một phong cách mới lạ, thay vì việc nhà nhà người người dùng Bootstrap như hiện nay.

1.2.7. Socket.io

1.2.7.1. Tổng quan về Socket.io

Hình 7: SocketIO Logo

Để xây dựng một ứng dụng realtime thì chúng ta cần phải sử dụng Socketio. Socketio có khả năng giúp những bên có địa điểm khác nhau có thể thực hiện dễ dàng hơn các kết nối với nhau để có thể truyền tải được dữ liệu một cách nhanh chóng và lập tức nhất thông qua hệ thống server trung gian. Ngoài ra, Socketio cũng được sử dụng cho các mục đích khác như: chat, game online, cập nhật kết quả của một trận đấu đang xảy ra,...

Socketio không được xem là ngôn ngữ mà chỉ là một trong những công cụ có thể hỗ trợ thực hiện cho một số ứng dụng dạng realtime. Chính vì vậy, bạn thể sử dụng Socketio để thay thế hoàn toàn cho một ngôn ngữ khác và các loại ngôn ngữ đó có thể là: php, asp.net, nodejs,...

1.2.7.2. Những điểm nỗi bật của Socket.io

Socket.io hiện đang được cộng đồng các lập trình viên dùng ngày càng nhiều bởi sự tiện lợi của nó. Hiện nay, Socket.io giúp ta sở hữu nhiều tính năng nổi bật như: bảo mật, binary, kết nối tự động, ghép kênh, tạo phòng,..

➢ **Khả năng bảo mật** :

Ngay khi Socket.io xuất hiện thì nó có thể tự động tạo ra những kết nối dạng bảo mật như:

- Thực hiện Proxy và cân bằng cho tải.
- Tạo tường lửa cá nhân và các phần mềm chống lại virus.

Socket.io thực hiện xây dựng dựa vào Engine.IO như sau: nó sẽ khởi chạy theo phương thức long-polling để tự động kết nối, rồi dùng những phương thức giao tiếp khác để giao tiếp được tốt hơn.

➢ **Kết nối lại tự động và phát hiện tình trạng ngắt kết nối**:

Khi chạy, nếu như client bị ngắt kết nối thì nó có thể kết nối tự động cho đến khi server xác nhận đã được phản hồi. Đây là tính năng cho phép thiết bị có thể tự động kết nối lại mãi mãi cho đến khi server phản hồi. Socket.io còn cung cấp cho bạn những event có thể phát hiện ngắt kết nối giữa client và server.

CHƯƠNG 2. PHÂN TÍCH VÀ THIẾT KẾ DỮ LIỆU

2.1. Đặc tả chức năng

2.1.1. Đối với người dùng

2.1.1.1. Xem nội dung trang Landing Page

Người dùng truy cập vào website có thể xem được trang chủ của Website, trang sẽ có vài thông tin nhỏ của web như slogan hoặc quote của người sử dụng. Người dùng lúc này có thể chọn Đăng nhập nếu đã có tài khoản hoặc Get Started (Đăng ký) nếu chưa.

2.1.1.2. Đăng ký tài khoản

Người dùng có thể đăng ký tài khoản mới để có thể nhắn tin, gọi điện với mọi người khác đang tham gia web.

Người dùng sau khi đăng ký tài khoản sẽ chưa có liên hệ (Conversation) nào ngoài một Conversation có tên là Saved Messages – nơi mà người dùng có thể gửi tin nhắn, hình ảnh muốn lưu trữ ở đó.

2.1.1.3. Đăng xuất khỏi hệ thống

Là chức năng người dùng thoát khỏi phiên đăng nhập hiện tại

2.1.1.4. Tìm kiếm người dùng

Khi người dùng nhập Tên người dùng hoặc địa chỉ Email đăng ký của người dùng để tìm kiếm những người dùng khác và bắt đầu cuộc trò chuyện.

2.1.1.5. Đăng nhập

Người dùng đăng nhập bằng tài khoản được tạo trước đó. Nếu người dùng đã có đăng ký tài khoản trước đó thì người dùng nhập tài khoản bằng email và mật khẩu để đăng nhập.

2.1.1.6. Nhắn tin trực tuyến

Nếu người dùng chưa có cuộc trò chuyện nào thì người dùng có thể tìm kiếm người dùng khác để bắt đầu trò chuyện.

Nếu đã có rồi người dùng có thể nhắn tin với nhau và người ở bên kia cũng sẽ nhận được tin nhắn ngay lập tức.

2.1.1.7. Gửi biểu cảm, cảm xúc(emoji)

Ngoài việc gửi tin nhắn bằng chữ, kí tự đơn giản, người dùng còn có thể chọn những biểu cảm khuôn mặt, trái tim, v...v… dễ thương để gửi cho bạn bè.

2.1.1.8. Gửi ảnh, gửi file

Người dùng có thể gửi ảnh cho nhau, ảnh sau khi gửi sẽ được lưu lại trong danh sách các ảnh đã gửi trong hộp thoại đó. Bên cạnh đó người dùng có thể trao đổi tài liệu bằng cách gửi các file mềm cho nhau như các loại file có đuôi mở rộng: *.docx, .pdf,..*

Hình ảnh gửi đi cũng là một dạng tin nhắn nhưng hình ảnh này có thể phóng to lên bằng cách click vào hình ảnh đó.

File/tệp tin sau khi gửi đi cũng sẽ được lưu vào một danh sách file của cuộc trò chuyện đó, người dùng có thể ấn chọn vào để tải file về máy của mình.

2.1.1.9. Gọi điện Video trực tuyến

Người dùng có thể gọi điện cho đối phương của mình, lập tức nếu người dùng phía bên kia (nếu cũng đang trực tuyến sẽ nhận được cuộc gọi) lúc này người ở đầu bên kia sẽ nhận được cuộc gọi, nhấc máy và bắt đầu trò chuyện.

2.1.1.10. Thu hồi tin nhắn

Người dùng có thể thu hồi bất cứ tin nhắn nào của mình đã gửi đi, thời gian để thực hiện việc thu hồi là không giới hạn – người dùng có thể đã gửi tin nhắn đó rất lâu trước đó và vẫn có thể thu hồi, tuy nhiên hành động đó sẽ không thể hoàn tác được.

2.1.1.11. Thay đổi thông tin cá nhân

Người dùng có thể chọn vào Profile Icon của mình và tiếp tục chọn Profile để có thể thay đổi tên, hình ảnh đại diện, cập nhật số điện thoại.

2.1.1.12. Thay đổi ảnh nền chat

Người dùng có thể thay đổi hình nền chat của mình nếu cảm thấy nhàm chán bằng cách chọn vào Profile Icon, chọn ảnh mà mình muốn và chọn Save.

2.1.2. Đối với người quản trị

2.1.2.7. Quản lý tài khoản

Vì đây là Web trao đổi thông tin nên những thông tin như các cuộc hội thoại của người dùng là bảo mật, vì thế nên người quản trị chỉ có thể xem được các liên quan về người dùng đó (Tên, ngày tham gia,…).

2.1.2.8. Đăng nhập vào trang quản trị

Người quản trị nhập email và password của mình đã tạo trước đó.

2.1.2.9. Đăng xuất khỏi trang quản trị

Người quản trị đăng xuất khỏi trang quản trị.

2.2. Use Case

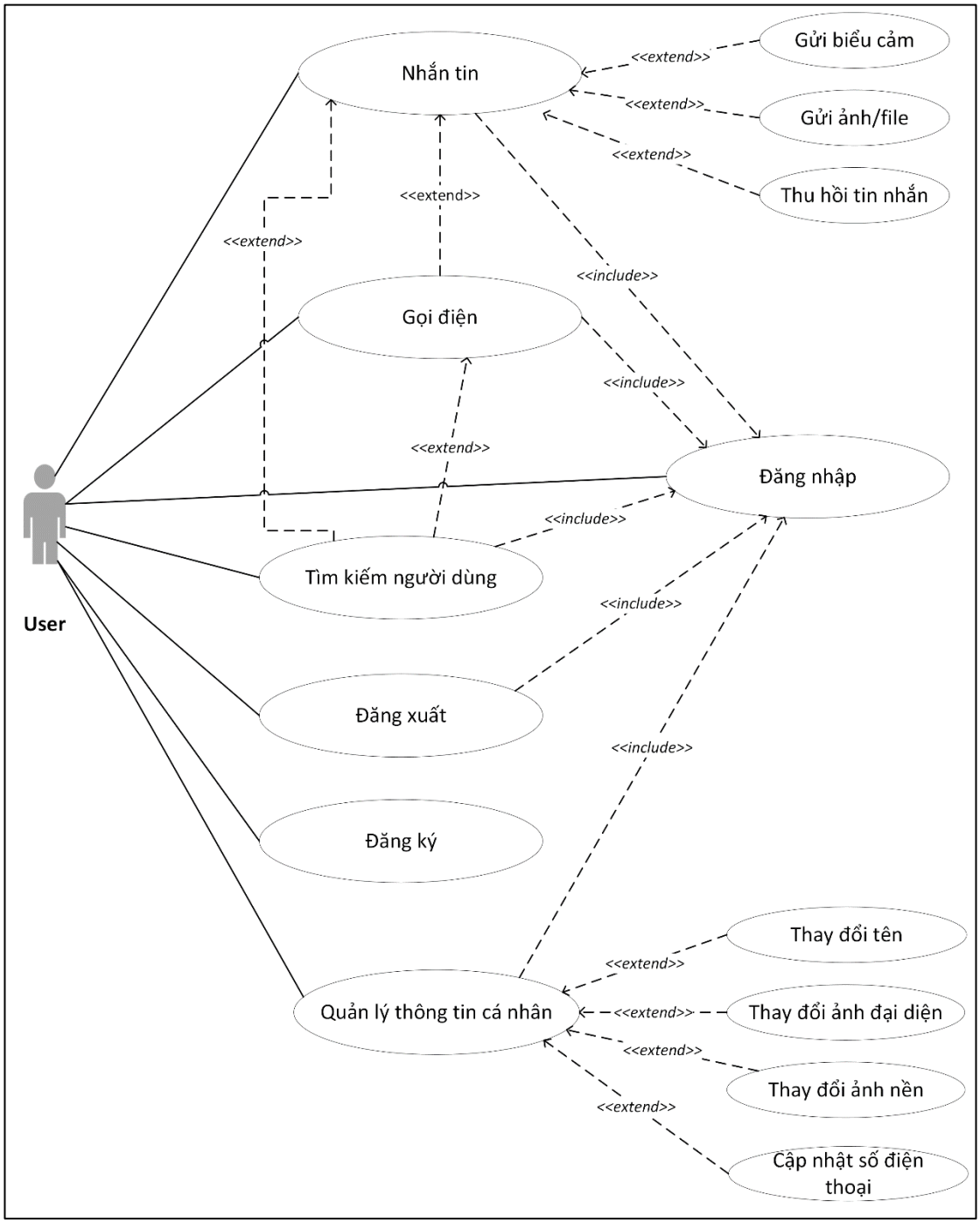
2.2.1. Xác định tác nhân trong hệ thống

Hình 8: Tác nhân trrong hệ thống

User (người dùng): là những người đăng ký tài khoản và sử dụng Web

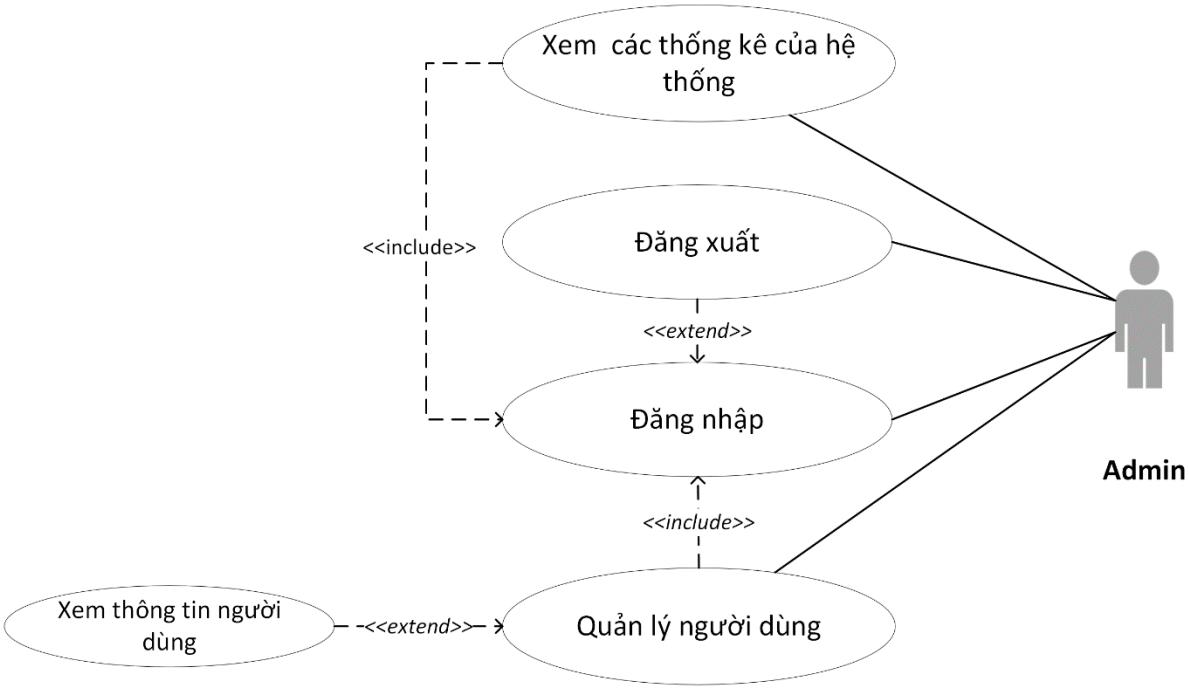
Admin : Là thành viên quản trị của hệ thống, có các quyền và chức năng, quản lý thông tin, thông số của người dùng, xem các thống kê liên quan.

2.2.2. Use Case tổng quát của phân hệ User



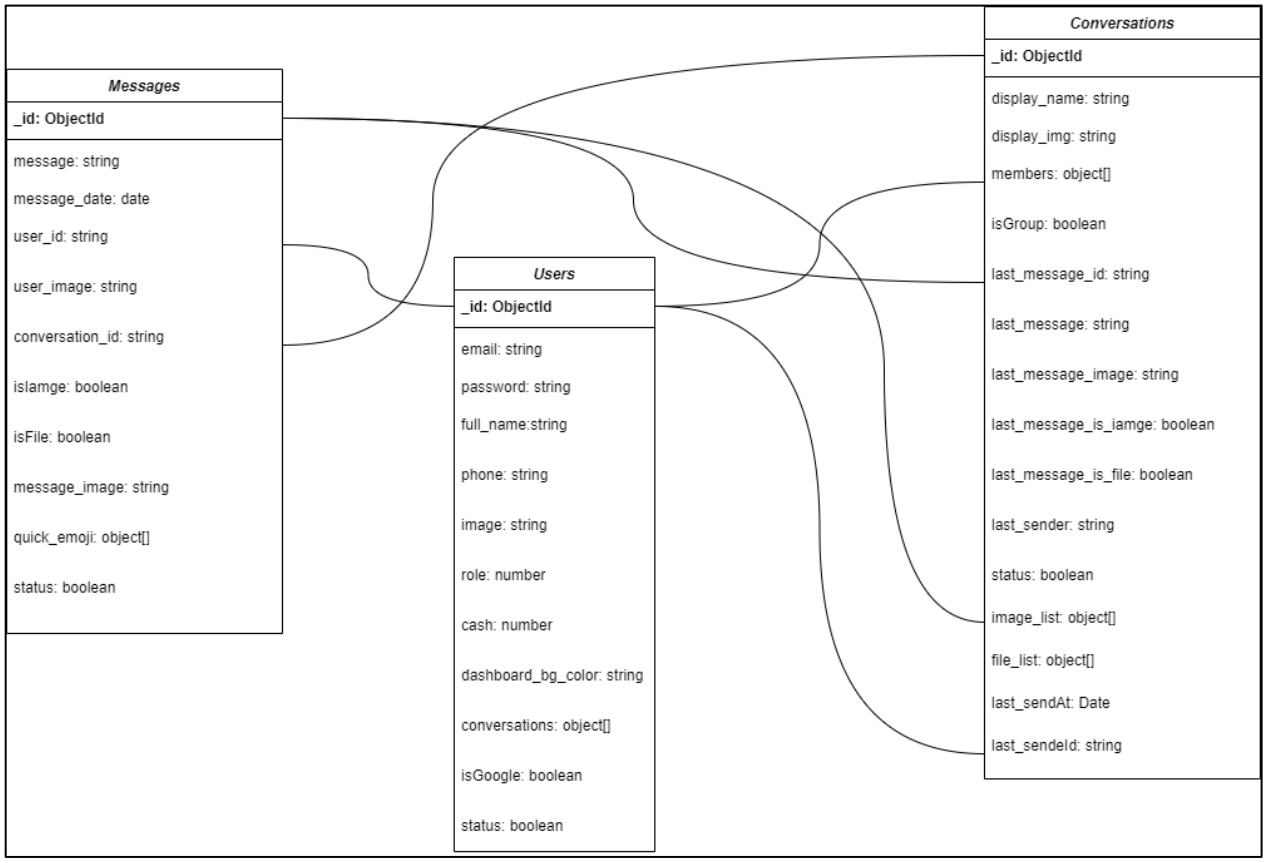
Hình 9: Use Case User tổng quát

2.2.3. Use Case tổng quát của phân hệ Admin



Hình 10: Use Case Admin tổng quát

2.3. Cơ sở dữ liệu



Hình 11: Cơ sở dữ liệu

2.3.1. Danh mục các bảng

2.3.1.1. Bảng Users (Người dùng)

Bảng 1: Bảng Users

Trường	Kiểu dữ liệu	Ràng buộc	Rỗng	Ghi chú
id	objectId	Khóa chính	không	Mã định danh, tự động tạo
email	string	N/A	không	Email đăng ký của user

password	string	N/A	không	Mật khẩu
full_name	stringl	N/A	không	Tên
phone	string	N/A	có	Số điện thoại
image	string	N/A	không	Ảnh đại diện
role	number	N/A	không	Phân cấp(user/admin)
cash	number	N/A	không	Số xu của user
dashboard_bg_color	string	N/A	không	Ảnh nền
conversations	object[]	NA	không	Danh sách liên hệ
isGoogle	boolean	N/A	không	Có phải là tài khoản đăng nhập bằng Google hay không.
status	boolean	N/A	không	Trạng thái tài khoản

2.3.1.2. Bảng Conversations (Liên hệ)

Bảng 2: Bảng Conversations

Trường	Kiểu dữ liệu	Ràng buộc	Rỗng	Ghi chú
id	objectId	Khóa chính	không	Id của liên hệ
display_name	string	N/A	không	Tên của liên hệ
members	object[]		không	Danh sách thành viên trong liên hệ

isGroup	boolean	N/A	không	Liên hệ này có phải Nhóm không
display_img	string	N/A	không	Hình ảnh của liên hệ
last_message_id	string	N/A	không	Id của tin nhắn cuối cùng được gửi đến liên hệ này
last_message	string	N/A	không	Nội dung của tin nhắn cuối cùng được gửi đến liên hệ này
last_message_image	string	N/A	không	Nếu tin nhắn cuối cùng là hình ảnh thì sẽ là giá trị của field này
last_message_is_image	boolean	N/A	không	Tin nhắn cuối cùng gửi đến liên hệ có phải là ảnh hay không
last_message_is_file	boolean	N/A	không	Tin nhắn cuối cùng gửi đến liên hệ có phải là file hay không
last_sender	string	N/A	không	Tên của người gửi tin nhắn cuối

				cùng đến liên hệ này
last_senderId	string	N/A	không	Id của người gửi tin nhắn cuối cùng đến liên hệ này
last_sendAt	date	N/A	không	Thời gian mà liên hệ nhận được tin nhắn cuối cùng
image_list	object[]	N/A	không	Danh sách hình ảnh mà người dùng đã gửi trong liên hệ
file_list	object[]	N/A	không	Danh sách file mà người dùng đã gửi trong liên hệ
status	boolean	N/A	không	Trạng thái

2.3.1.3. Bảng Messages (Tin nhắn)

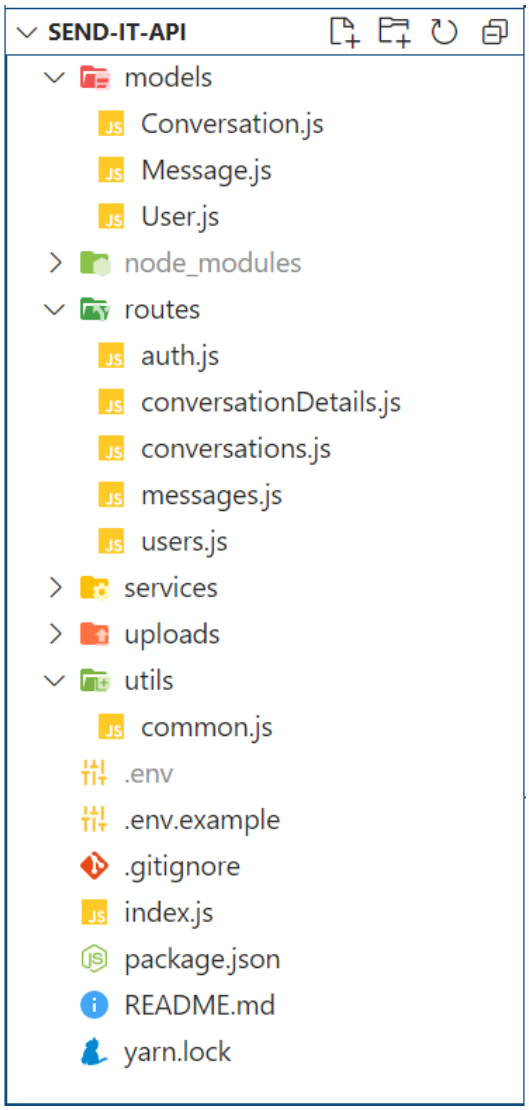
Bảng 3: Bảng Messages

Trường	Kiểu dữ liệu	Ràng buộc	Rỗng	Ghi chú
id	objectId	Khóa chính	không	Id của tin nhắn
message	string	N/A	không	Nội dung tin nhắn
message_date	int	N/A	không	Thời gian tin nhắn được gửi
user_id	string	N/A	không	Id của người gửi tin nhắn
user_image	string	N/A	không	Ảnh đại diện của người gửi tin nhắn
conversation_id	string	N/A	không	Id của liên hệ mà tin nhắn gửi đến
isImage	boolean	N/A	không	Tin nhắn này có phải là ảnh hay không
isFile	boolean	N/A	không	Tin nhắn này có phải là File hay không
message_image	string	N/A	không	Đường dẫn nơi lưu ảnh
quick_emoji	object[]	N/A	không	Biểu cảm/cảm xúc mà tin nhắn nhận được
status	boolean	N/A	không	Trạng thái

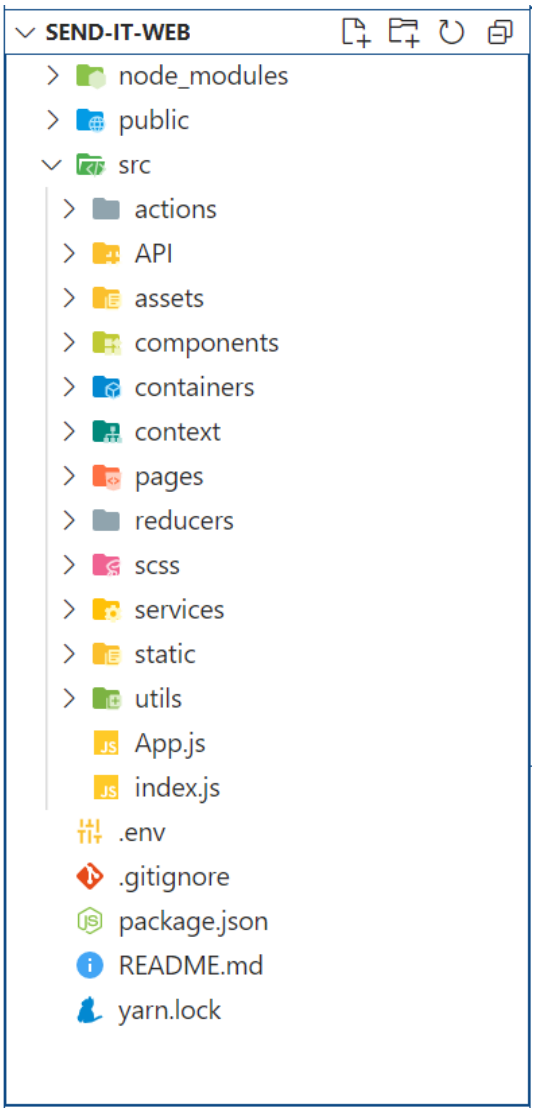
CHƯƠNG 3. CÀI ĐẶT GIAO DIỆN VÀ CHƯƠNG TRÌNH

3.1. Cấu trúc thư mục đồ án

3.1.1. Cấu trúc thư mục ở Backend và Frontend

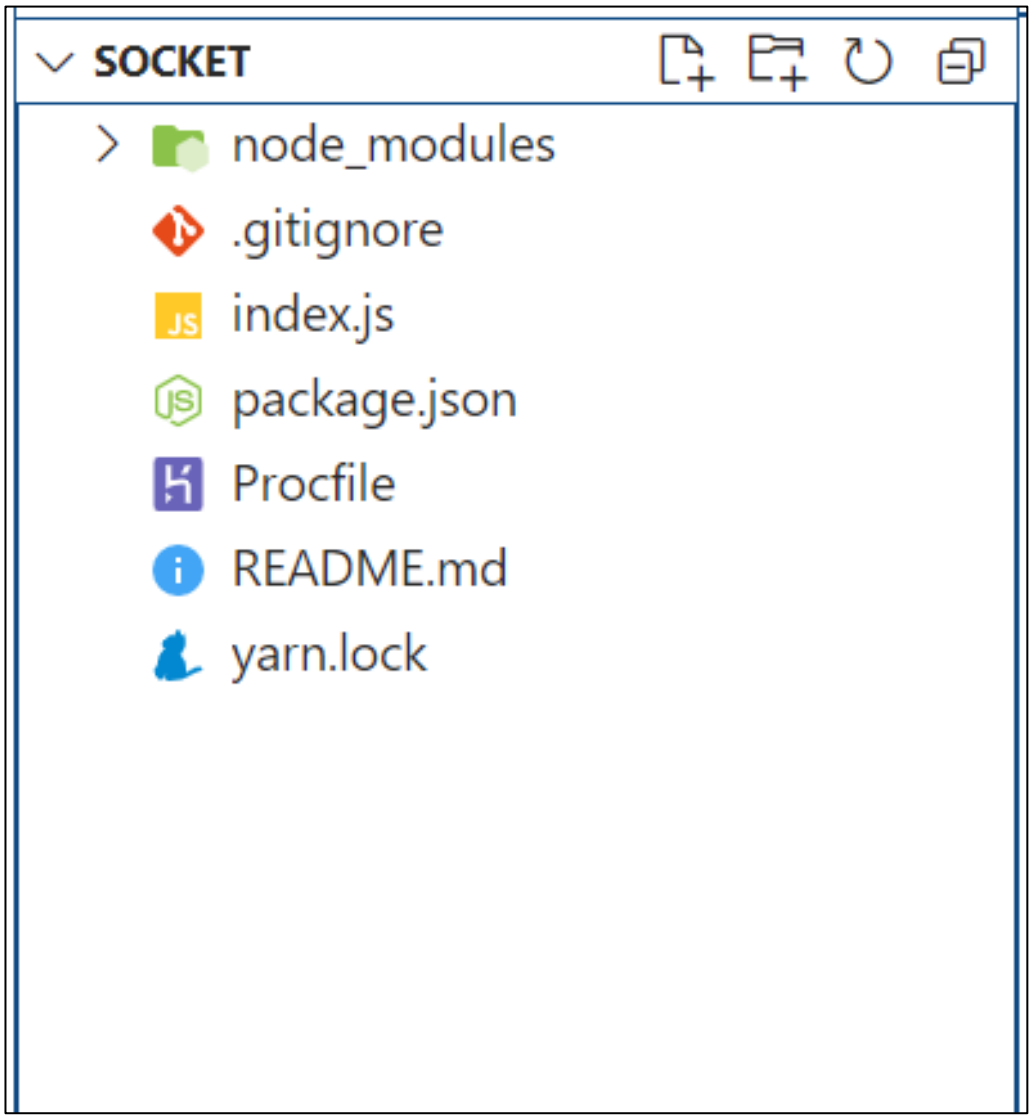


Hình 12. Cấu trúc thư mục Backend



Hình 13: Cấu trúc thư mục Frontend

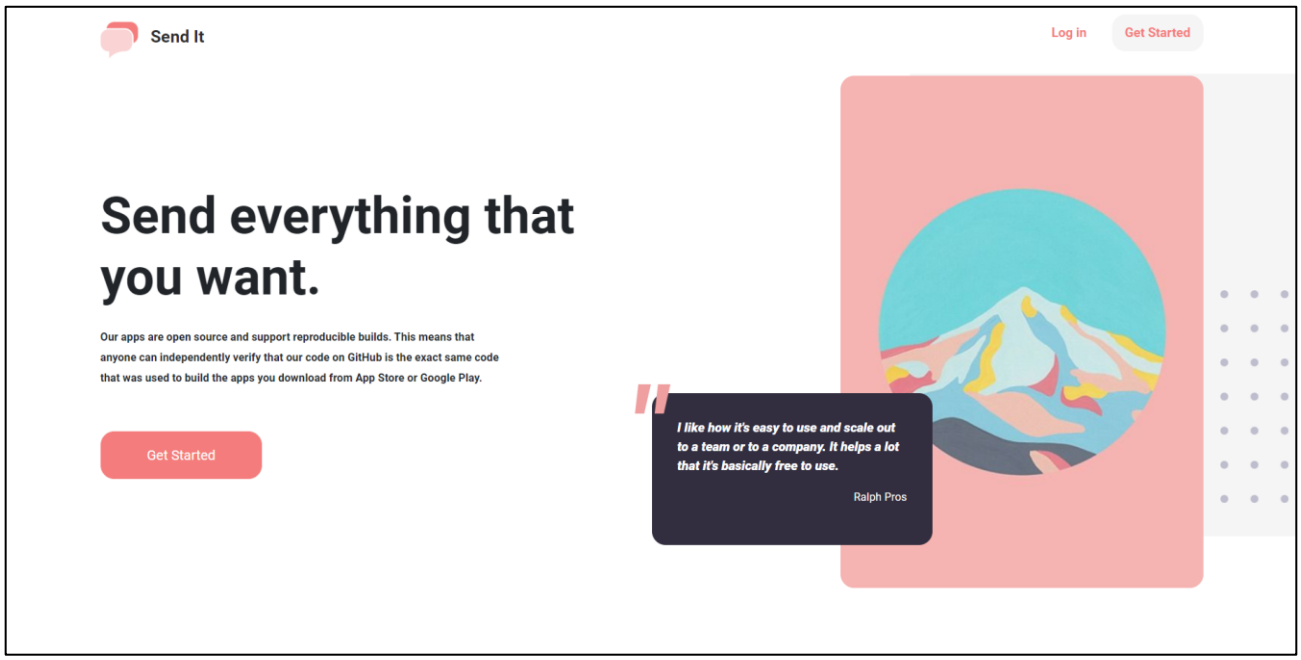
3.1.2. Cấu trúc thư mục Socket.io



Hình 14: Cấu trúc thư mục SocketIO

3.2. Giao diện website phân hệ của User

3.2.1. Giao diện trang chủ website



Hình 15: Giao diện trang chủ website

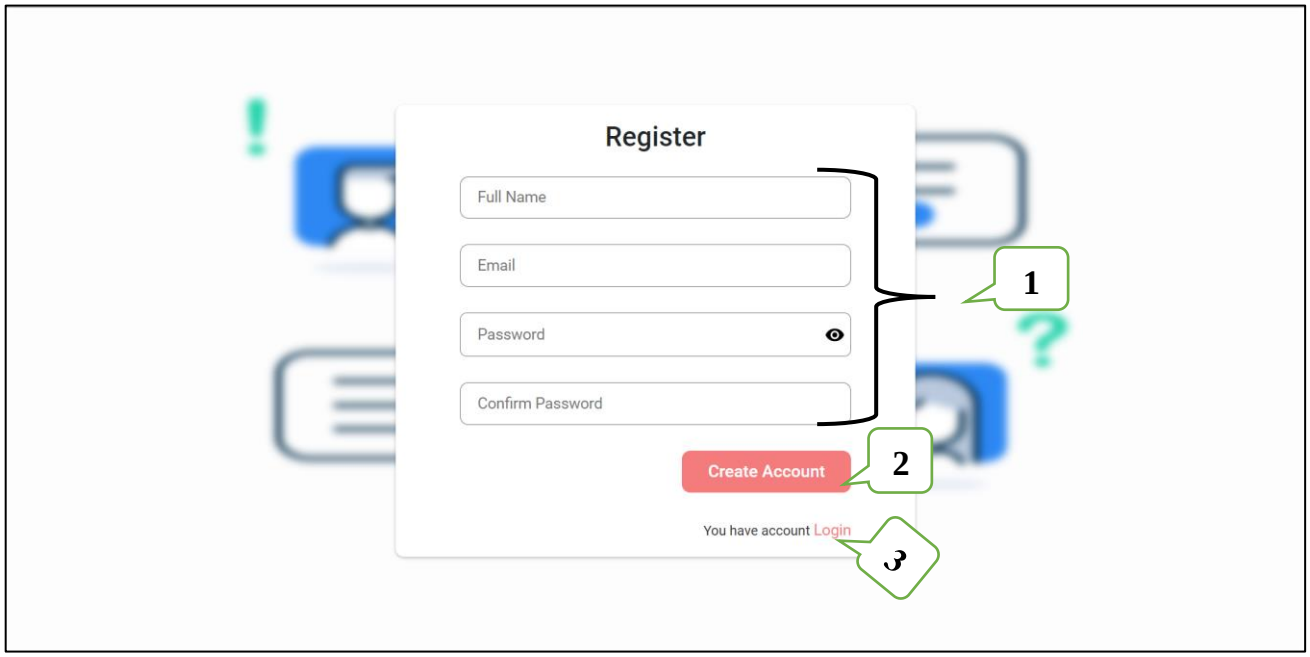
Ở giao diện trang chủ người dùng có thể xem được Slogan của Website và Quote của khách hàng sử dụng website, người dùng có thể chọn đăng nhập (Login) hoặc Đăng ký tài khoản mới (Get Started).

3.2.2. Giao diện đăng ký

Người dùng chọn vào nút Get Started ở màn hình trang chủ hoặc chọn "Create account" ở màn hình đăng nhập để có thể đăng ký.

Một số ràng buộc khi đăng ký thành viên:

- Mật khẩu: Độ dài mật khẩu lớn hơn 6 ký tự.
- Email: Phải đúng định dạng của email và không được trùng với tài khoản khác(unique).
- Tên người dùng: Tên người dùng phải lớn hơn 6 ký tự.



Hình 16: Giao diện đăng ký

Chú thích:

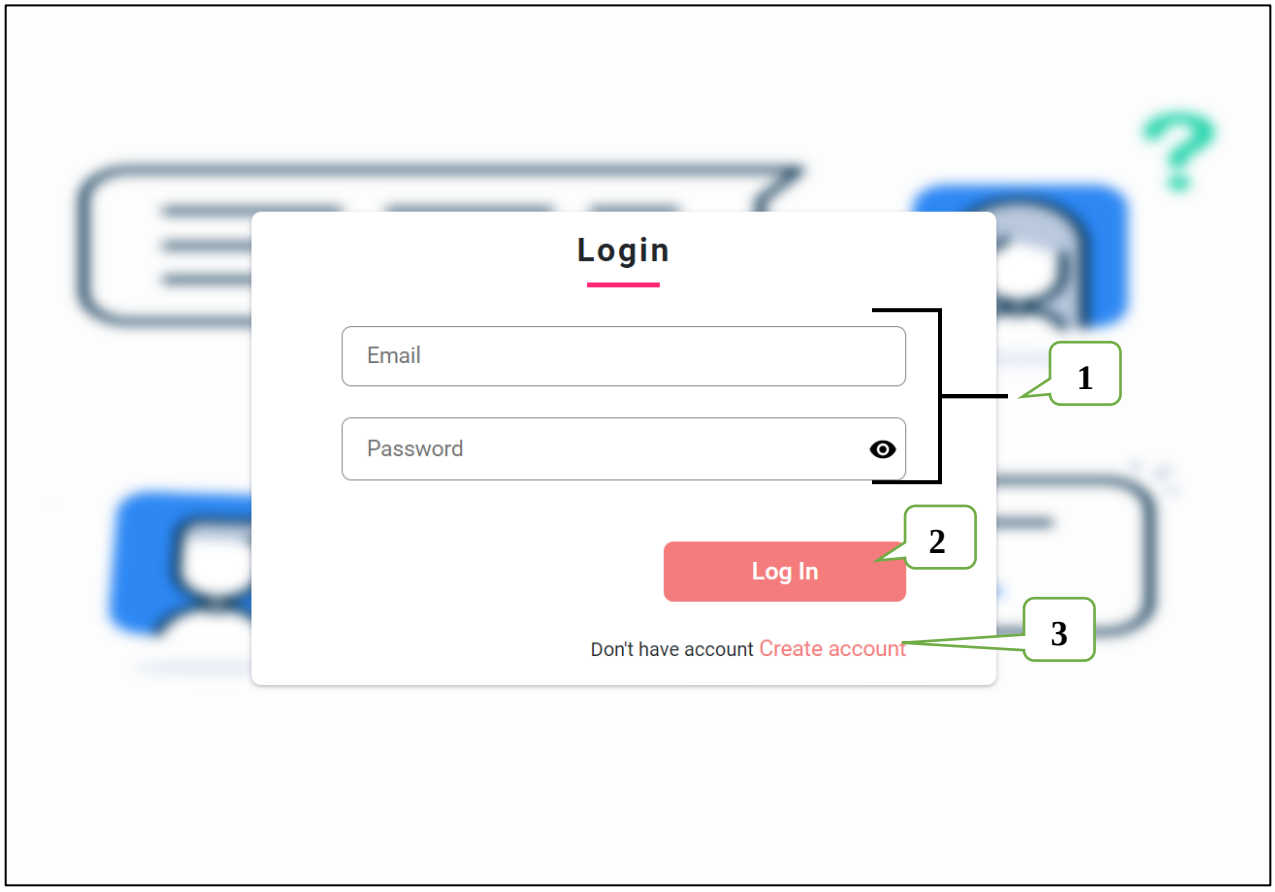
[1]: Vùng nhập thông tin người dùng

[2]: Chọn Đăng ký

[3]: Quay về đăng nhập

3.2.3. Giao diện đăng nhập

Người dùng đăng nhập tài khoản đã tạo trước đó để vào website.



Hình 17: Giao diện đăng nhập

Chú thích:

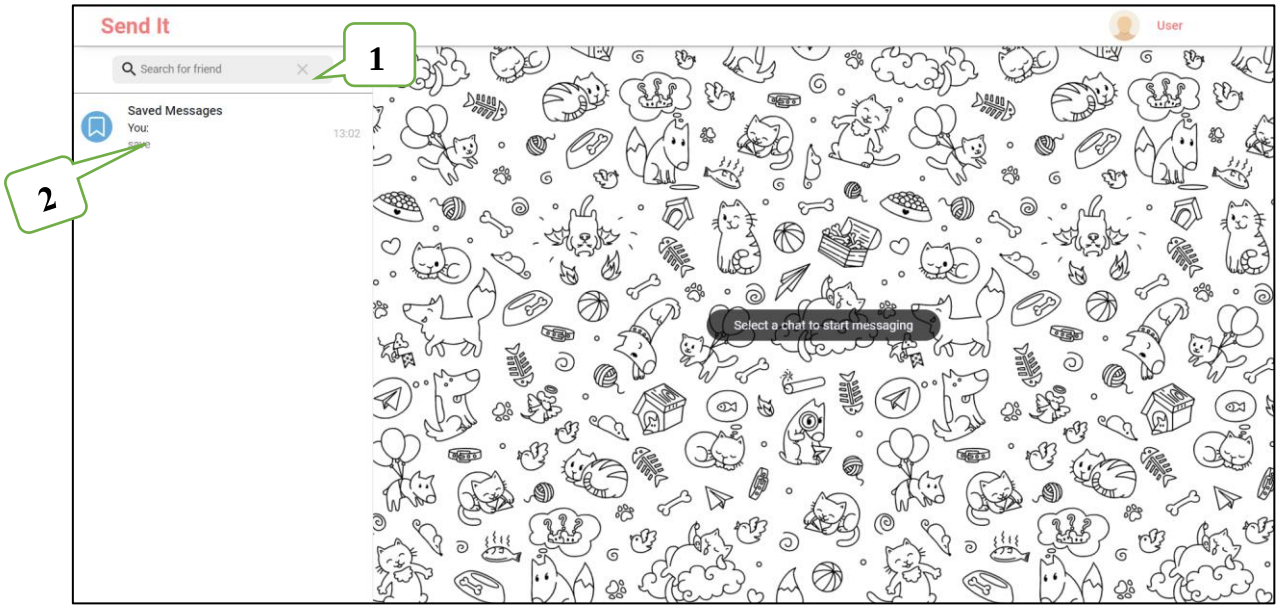
[1]: Vùng nhập tài khoản và mật khẩu

[2]: Nút đăng nhập

[3]: Nút quay lại trang đăng ký

3.2.4. Giao diện Chính (sau khi đăng nhập thành công)

Khi người dùng đăng nhập thành công thì đây là màn hình chính



Hình 18: Giao diện màn hình chính (đăng nhập thành công)

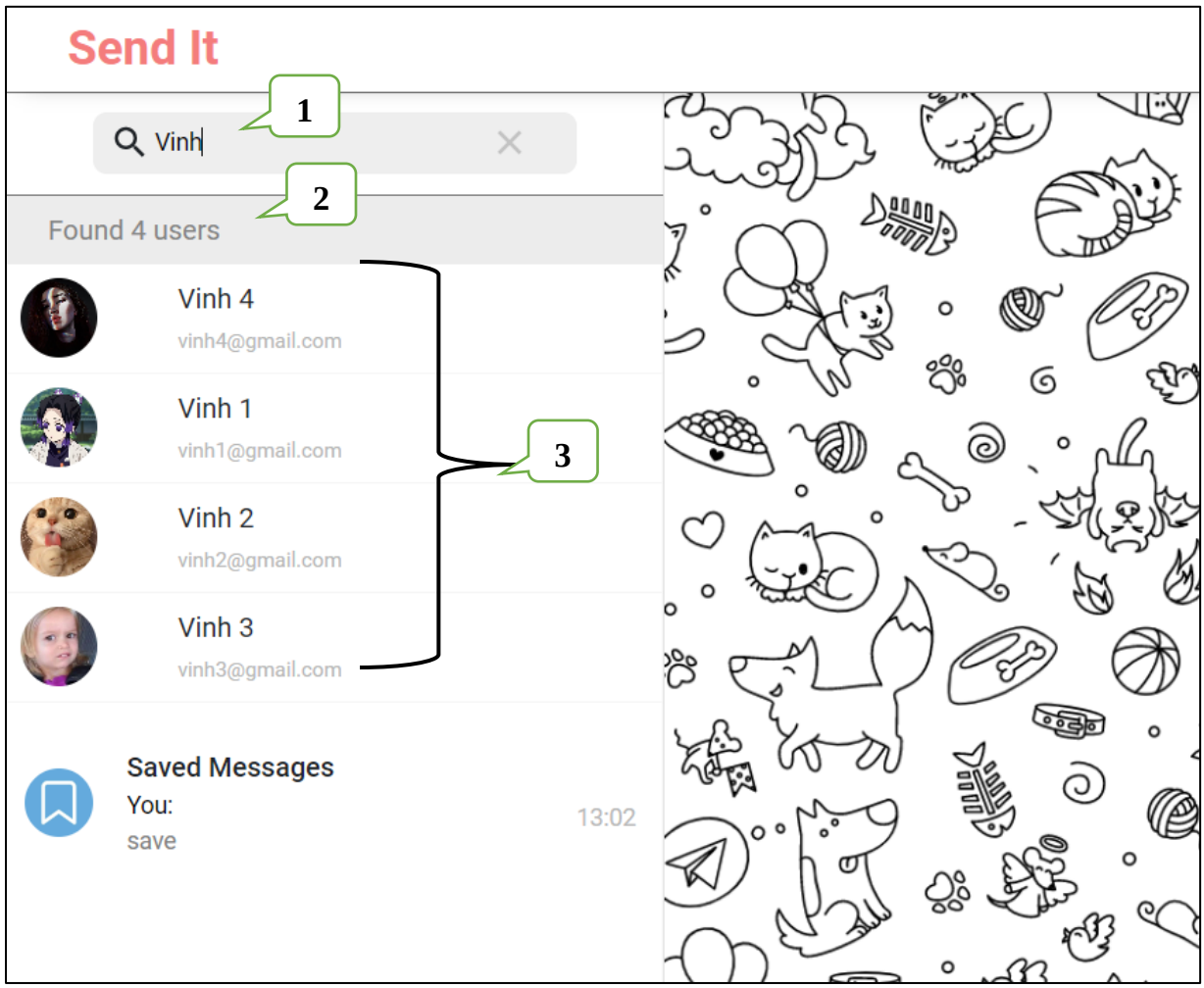
Chú thích:

[1]: Ô nhập thông tin tìm kiếm người dùng khác (Tên/Email).

[2]: Các liên hệ của người dùng (hình trên là mối liên hệ có sẵn của hệ thống mà người mới tham gia nào cũng có dùng để lưu tin nhắn của chính mình).

3.2.5. Giao diện tìm kiếm người dùng

Người dùng nhập từ khóa cần tìm ở ô tìm kiếm (Tên hoặc email của người dùng khác), sau khi người dùng ngừng nhập khoảng nửa giây hệ thống sẽ tự động hiện thông tin các người dùng khác.



Hình 19: Giao diện tìm kiếm người dùng

Chú thích:

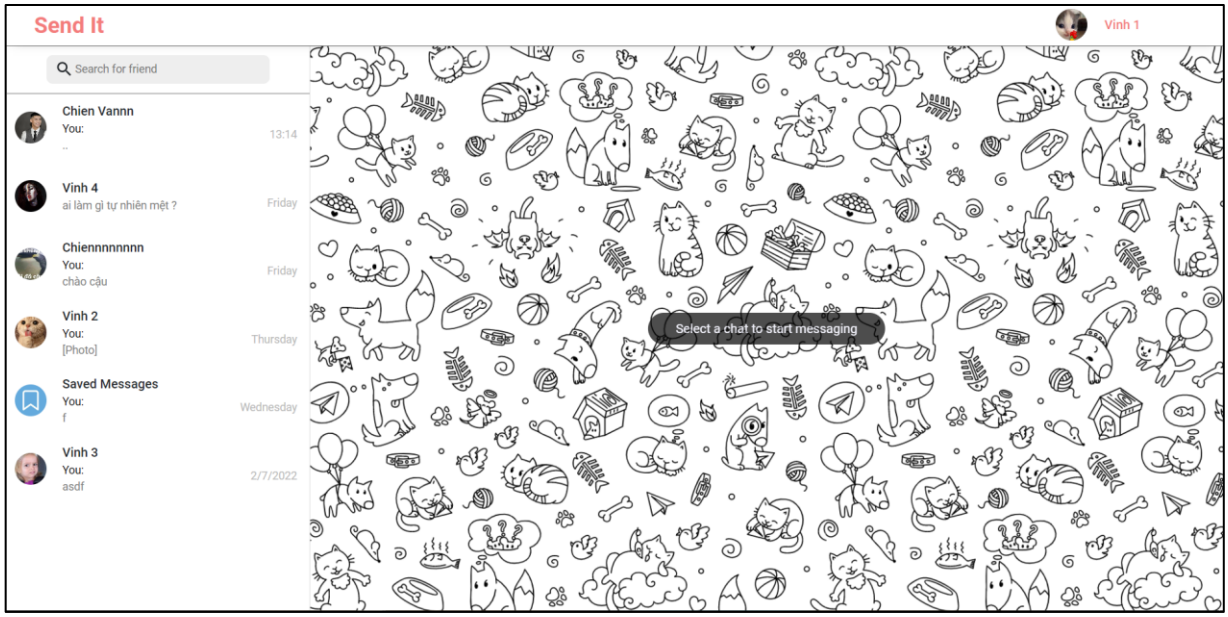
[1]: Từ khóa tìm kiếm

[2]: Tổng số người dùng được tìm thấy

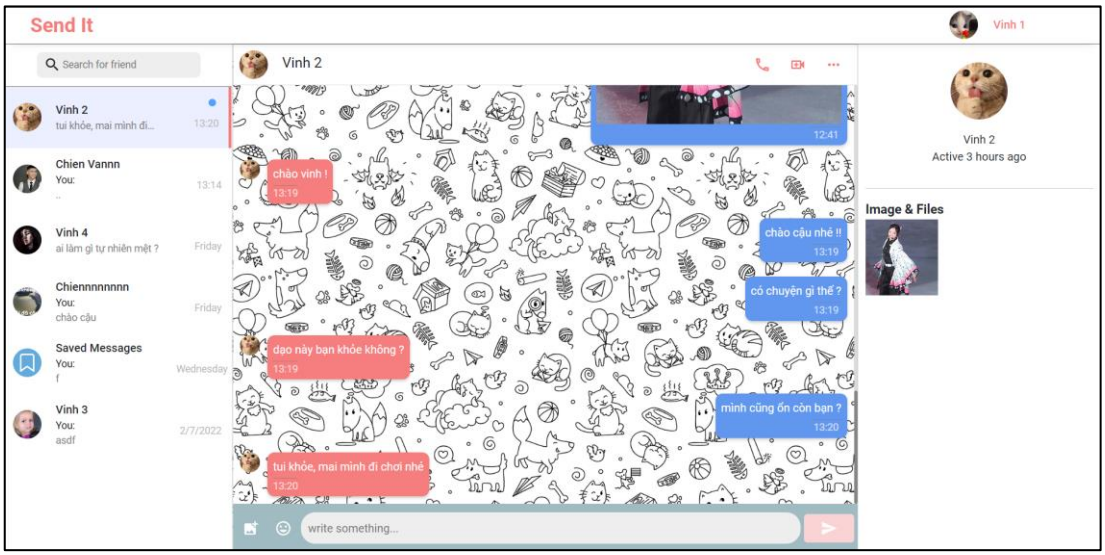
[3]: Các người dùng khác được tìm thấy theo từ khóa tìm kiếm

3.2.6. Giao diện màn hình chính khi bạn có nhiều liên lạc

Sau khi người dùng đăng nhập thành công hoặc mở app lên thì hệ thống sẽ hiện lên giao diện chờ chat. Khi người dùng chọn vào một box chat bất kì thì màn hình chat sẽ hiện lên và có thể chat với người trong box chat đó.



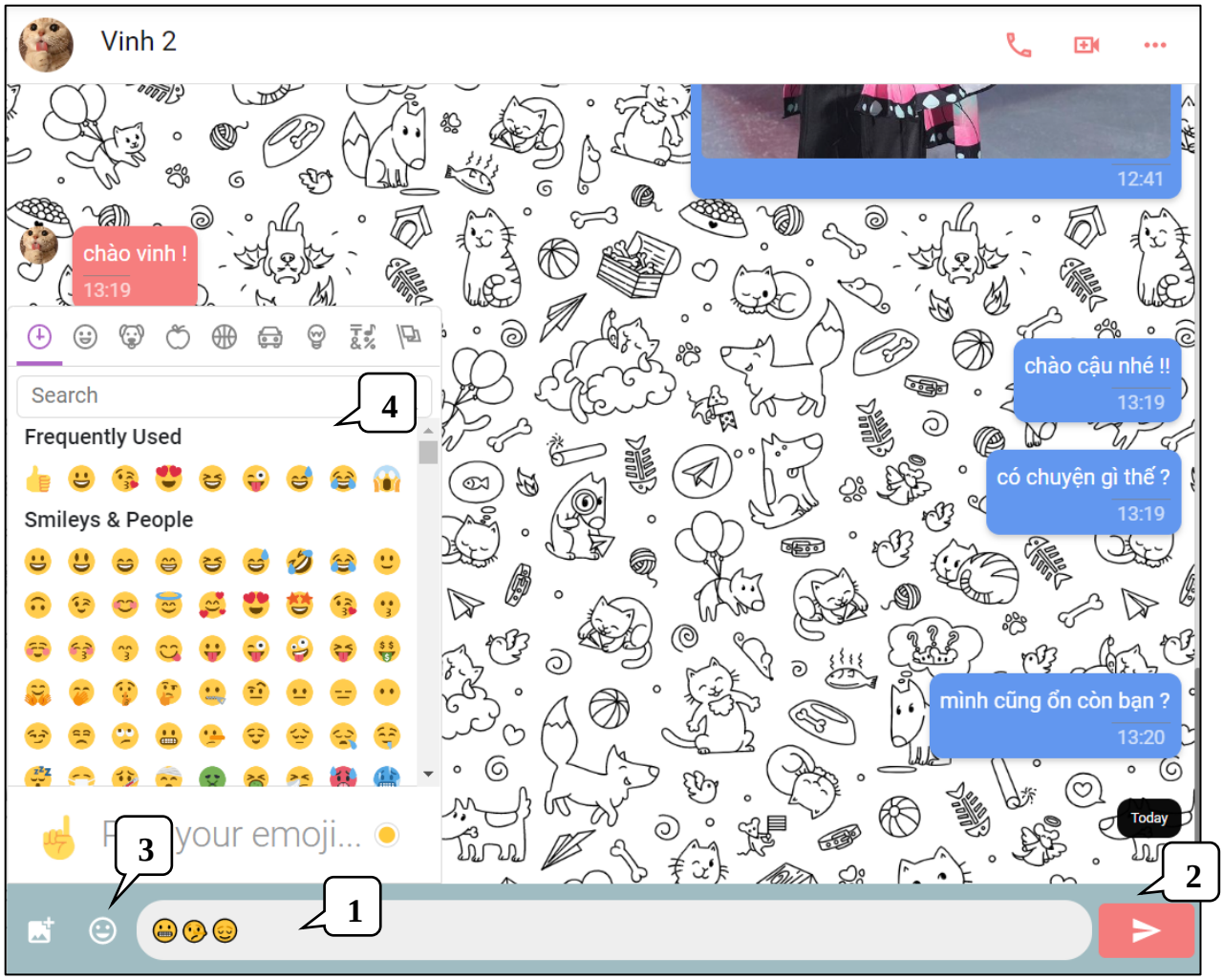
Hình 20: Giao diện màn hình chính(khi chưa chọn liên hệ)



Hình 21: Giao diện chat với bạn bè

3.2.7. Giao diện gửi tin nhắn và emoji

Sau khi chọn vào một box chat thì người dùng có thể nhập nội dung vào ô số [**1**] và nhấn gửi thì tin nhắn sẽ được gửi cho đối phương, ngoài nhập nội dung tin nhắn thì người dùng còn có thể chọn vào nút emoji để chọn emoji và gửi kèm với nội dung tin nhắn.



Hình 22: Giao diện chọn emoji

Chú thích:

[1]: Ô nhập nội dung tin nhắn

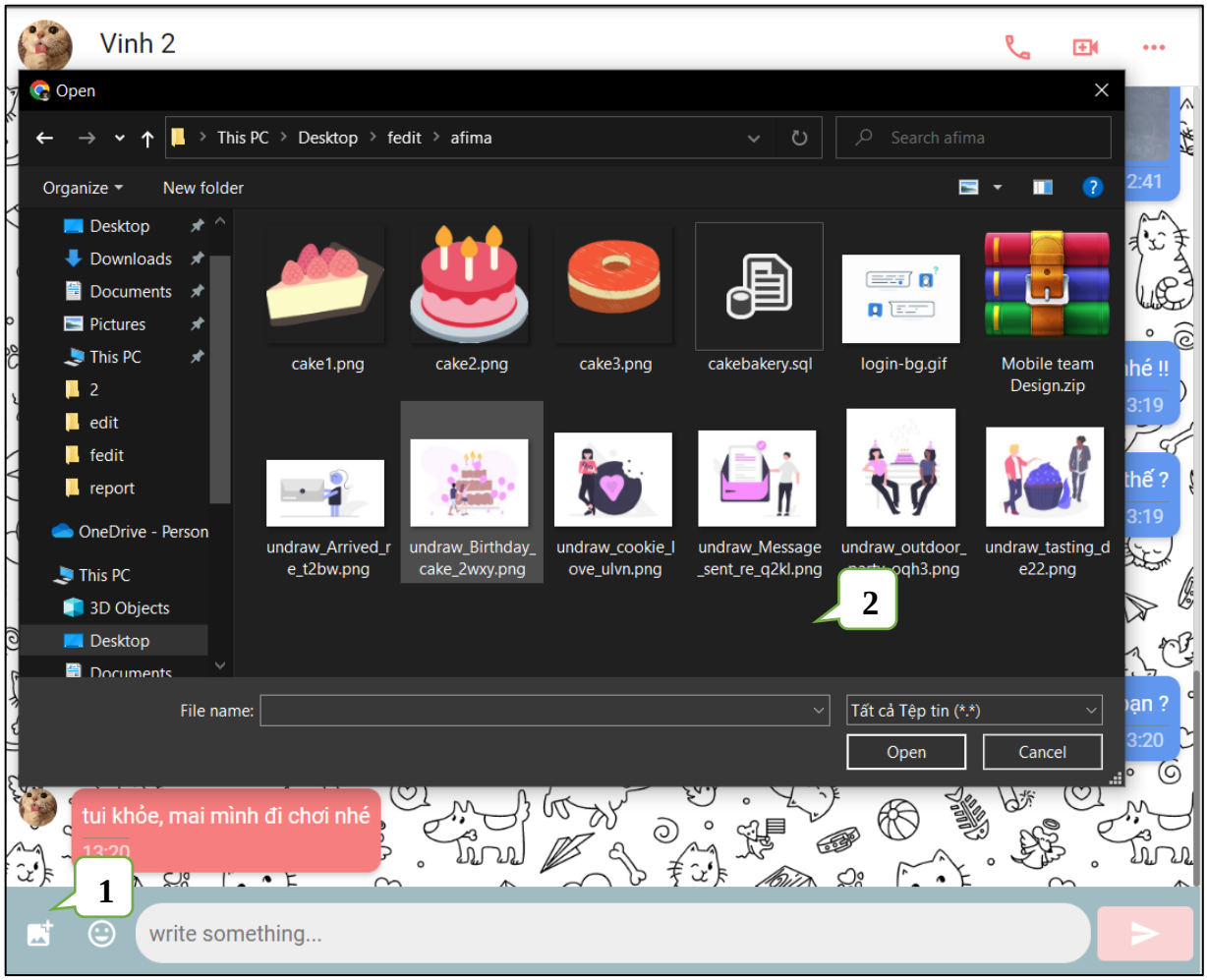
[2]: Nút gửi tin nhắn

[3]: Nút chọn emoji

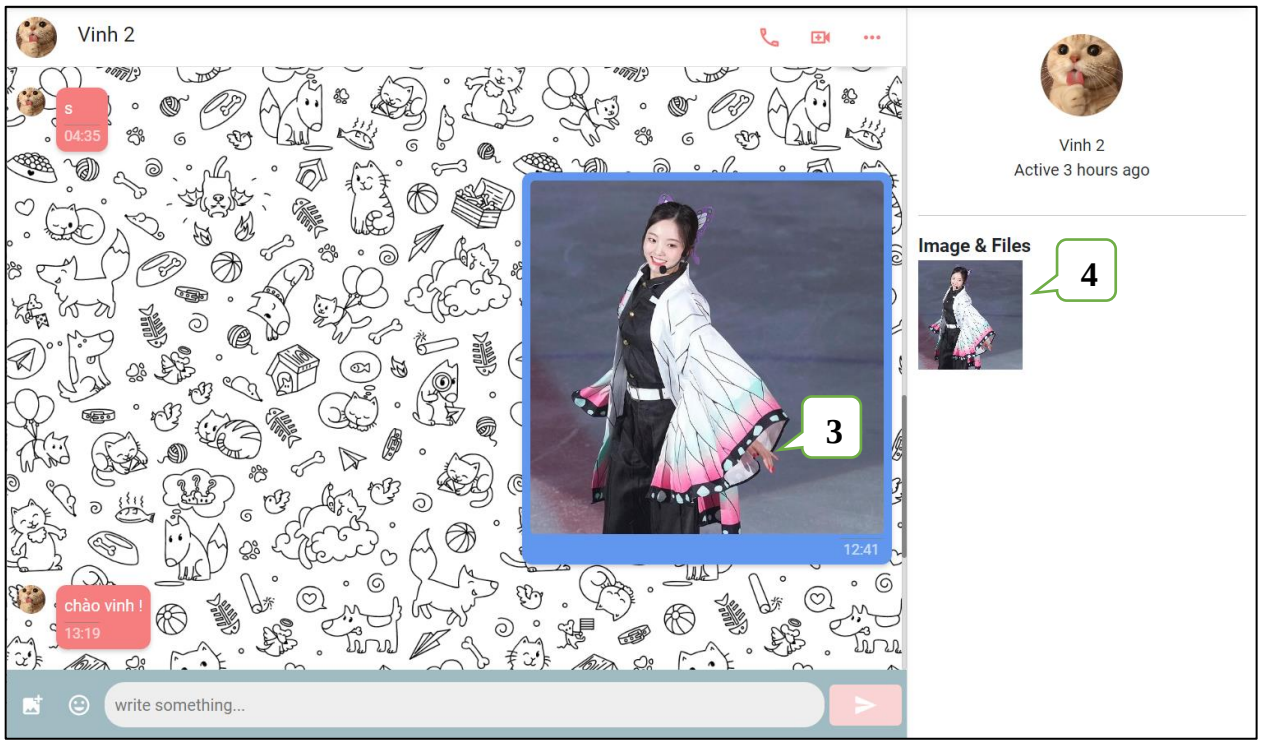
[4]: Khung chọn emoji

3.2.8. Giao diện gửi hình ảnh / gửi file

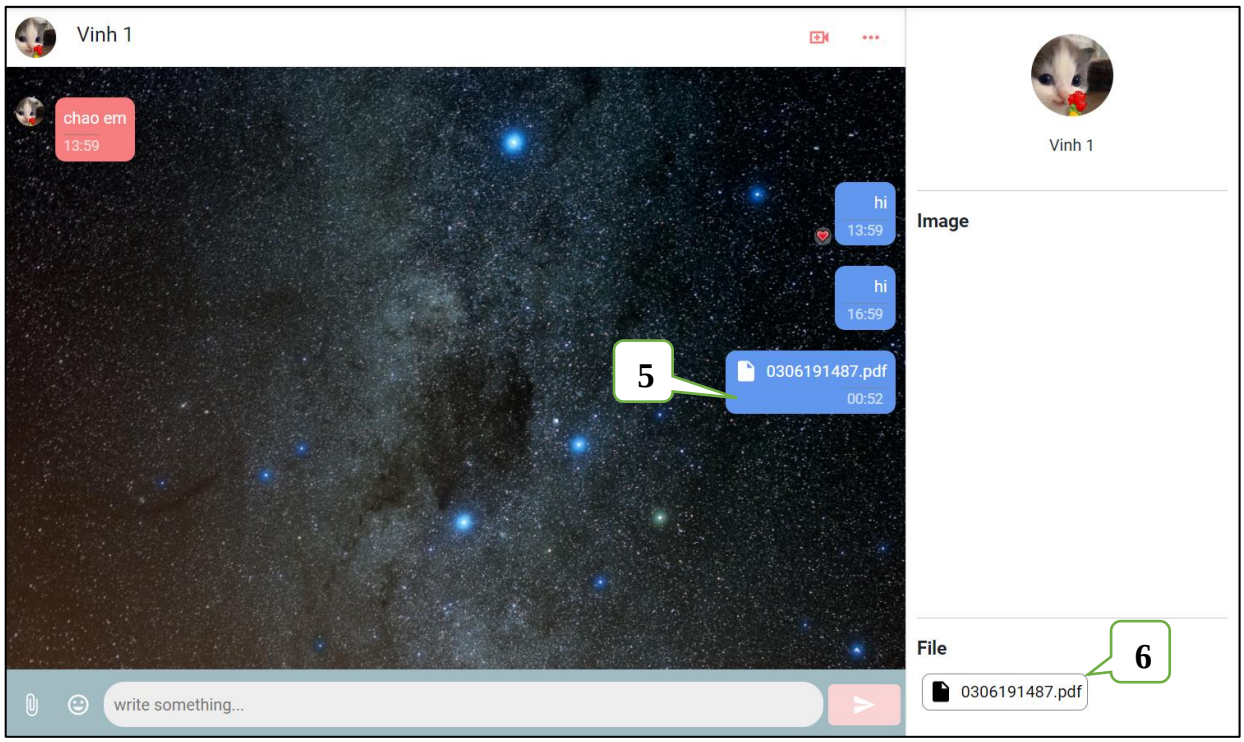
Sau khi chọn nút gửi ảnh thì một pop-up chọn hình ảnh sẽ được hiện lên và người dùng thực hiện thao tác chọn ảnh để gửi. Sau khi chọn thì hình ảnh sẽ được gửi đi và hiện lên khung chat và mục hình ảnh đã gửi.



Hình 23: Giao diện thao tác gửi ảnh



Hình 24: Giao diện gửi ảnh thành công



Hình 25: Giao diện gửi ảnh thành công

Chú thích:

[1]: Nút để thực hiện thao tác gửi ảnh

[2]: Nút chọn ảnh để gửi

[3]: Hình ảnh sau khi gửi thành công

[4]: Hình ảnh được lưu vào hình ảnh đã gửi

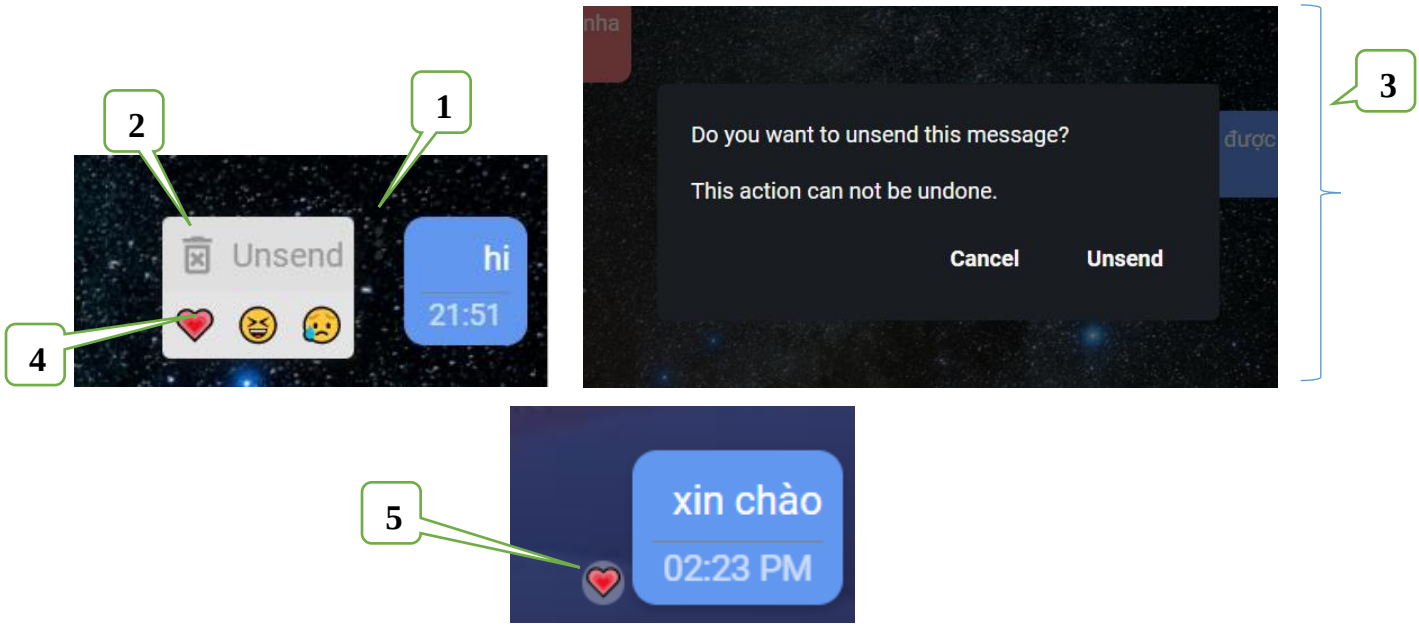
[5]: File sau khi được gửi

[6]: File được gửi trong liên hệ sẽ được lưu ở danh sách này

3.2.9. Giao diện thu hồi tin nhắn, thả cảm xúc tin nhắn

Mỗi tin nhắn sẽ có một nút để thực hiện các chức năng và trong đó có chức năng thu hồi tin nhắn đó. Khi người dùng muốn thu hồi tin nhắn nào thì nhấn vào nút menu ở cuối tin nhắn đó sau đó sẽ hiện ra các chức năng. Chọn vào chức năng “Unsend”, sau

khi chọn xong một pop-up sẽ hiện ra để xác nhận việc có thu hồi tin nhắn đó hay không. Xác nhận thu hồi tin nhắn sẽ không thể hoàn tác.



Hình 26: Giao diện các bước thu hồi và thả cảm xúc tin nhắn

Chú thích:

[1]: Nút chọn chức năng

[2]: Chọn chức năng thu hồi tin nhắn

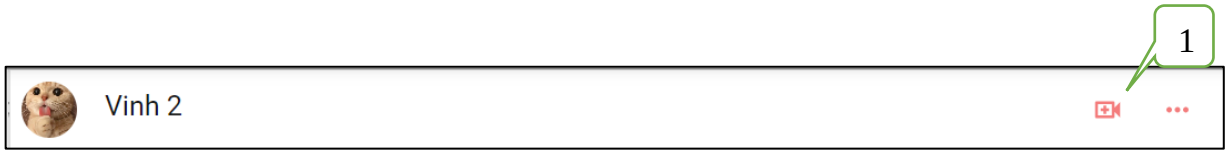
[3]: Pop-up xác nhận thu hồi tin nhắn

[4]: Thả cảm xúc tin nhắn

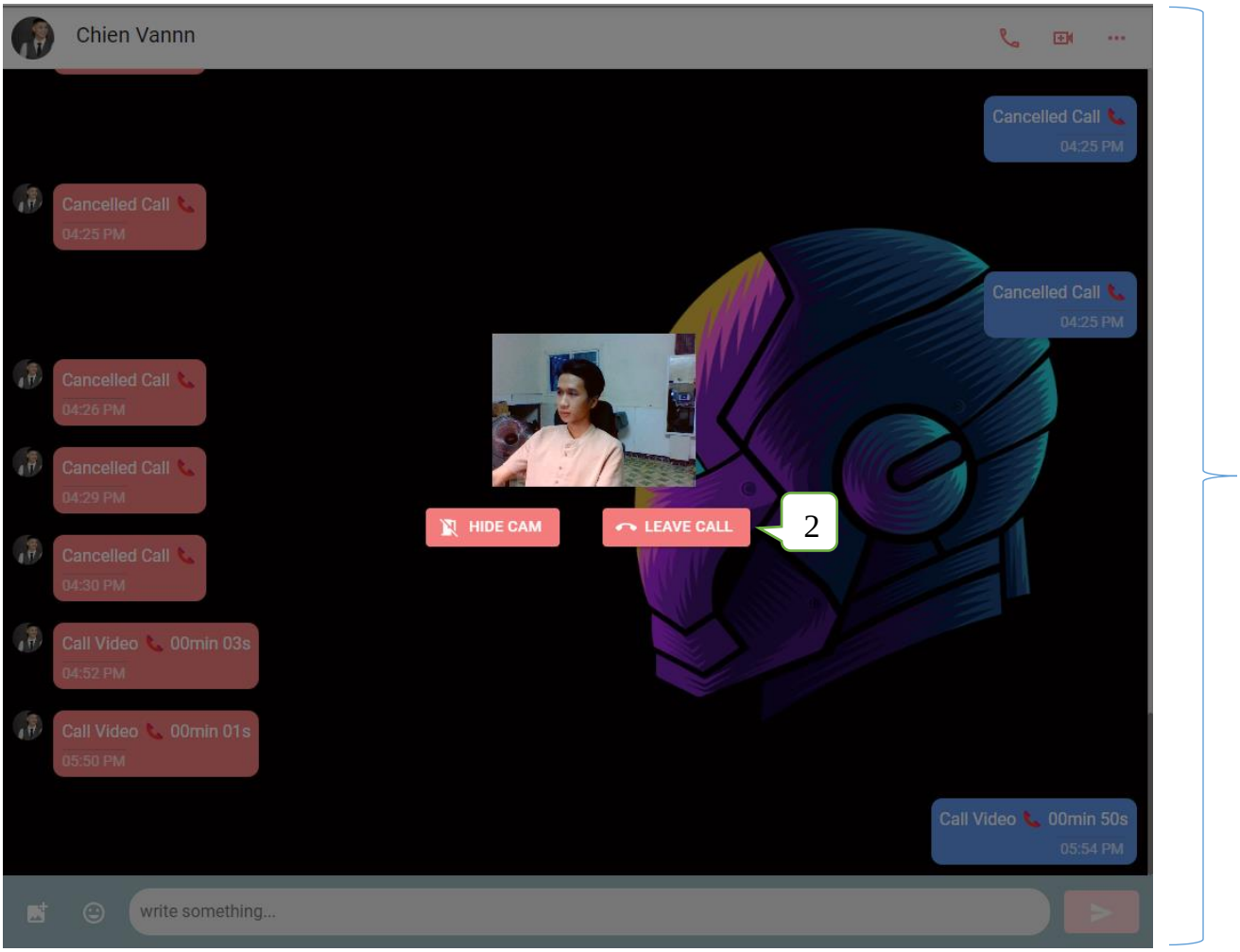
[5]: Tin nhắn sau khi được thả cảm xúc

3.2.10. Giao diện call video

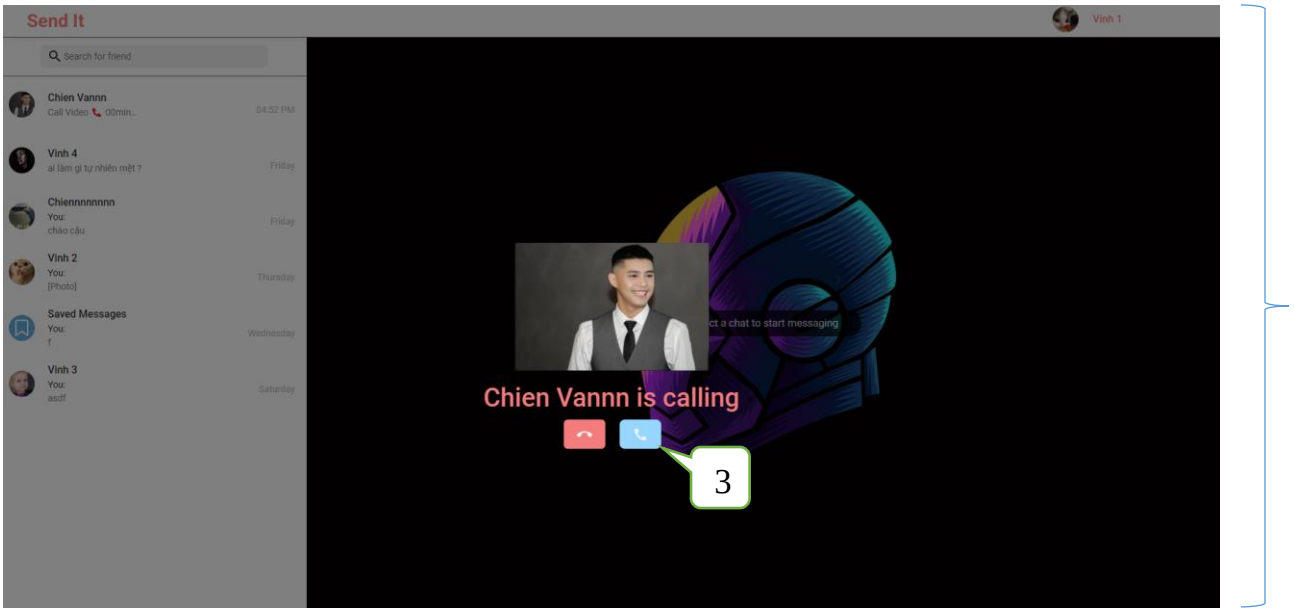
Khi nhấn vào nút thực hiện chức năng call video hệ thống sẽ tìm đến người được gọi và hiện một thông báo cho người đó. Có 2 lựa chọn là chấp nhận hoặc từ chối cuộc gọi. Trong quá trình gọi thì người dùng có thể sử dụng tính năng ẩn camera.



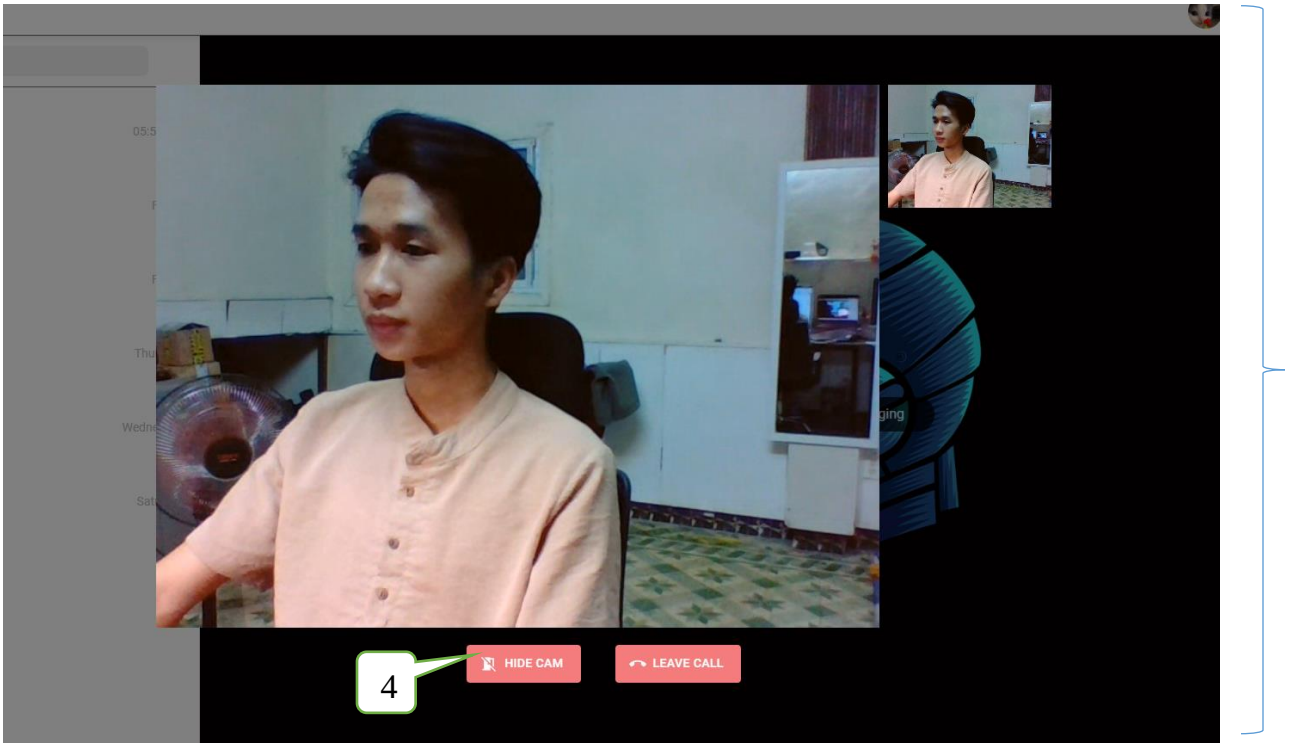
Hình 27: Giao diện nút bắt đầu cuộc gọi



Hình 28: Giao diện chờ khi thực hiện cuộc gọi



Hình 29: Giao diện nhận thông báo khi có người gọi đến



Hình 30: Giao diện đang thực hiện cuộc gọi

Chú thích:

[1]: Nút thực hiện cuộc gọi

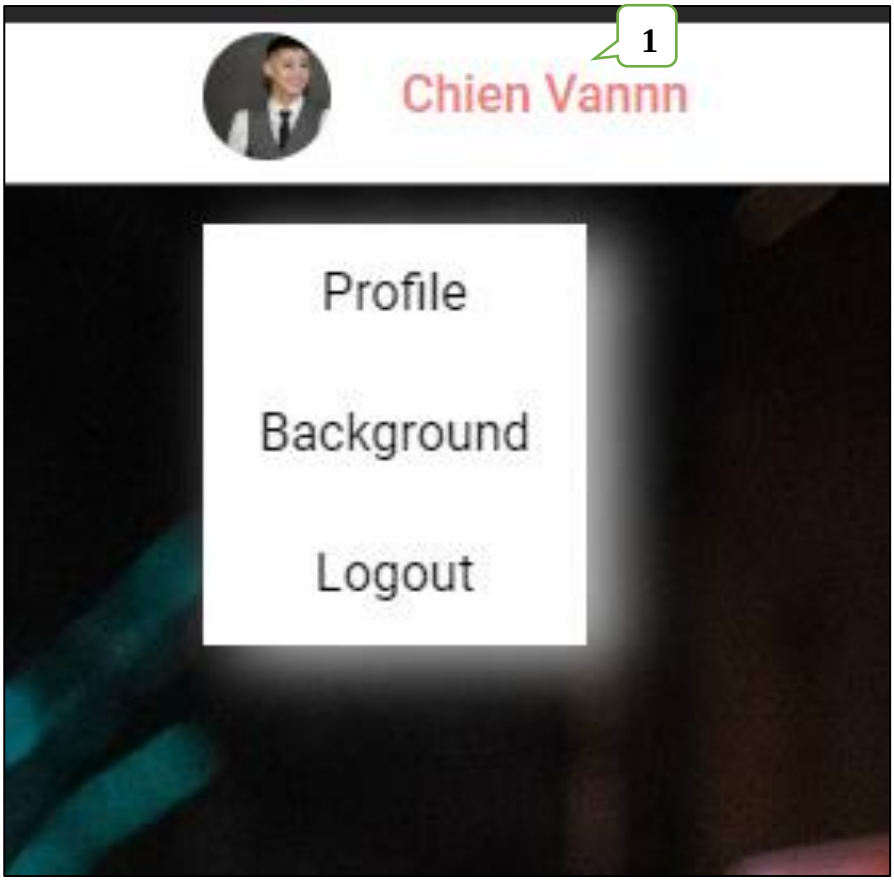
[2]: Nút ngưng cuộc gọi

[3]: Nút chấp nhận cuộc gọi

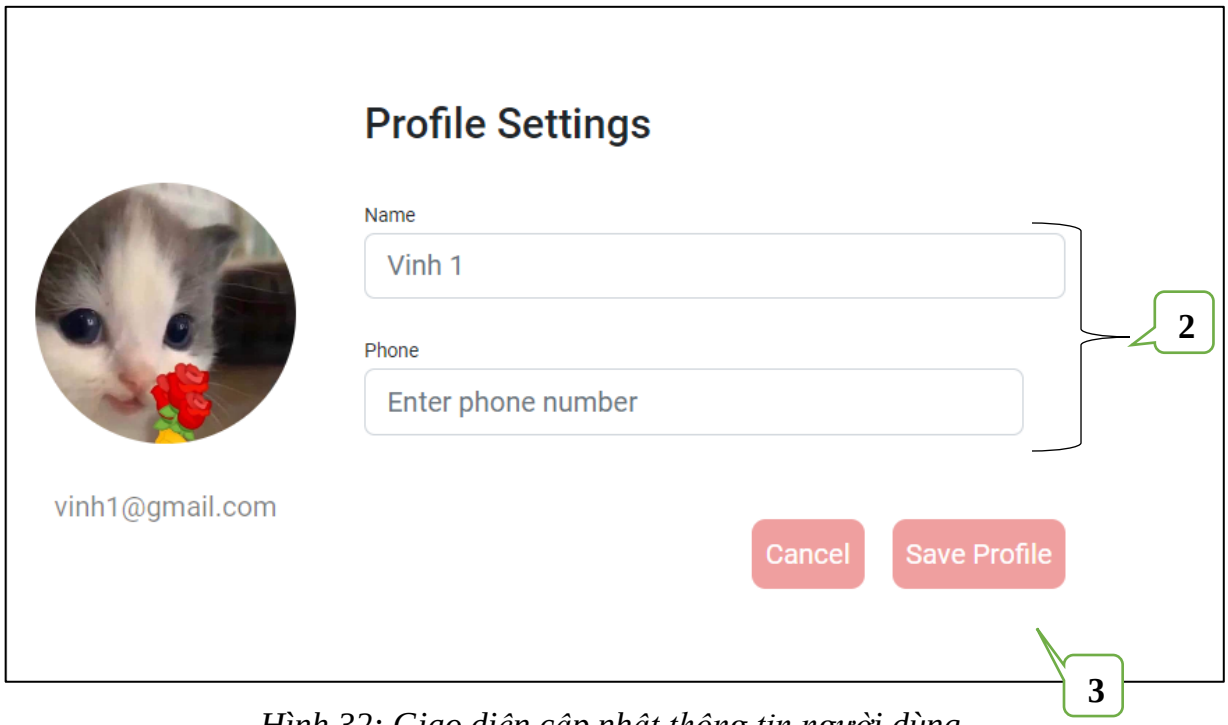
[4]: Nút ẩn camera

3.2.11. Giao diện cập nhật thông tin người dùng

Sau khi người dùng chọn vào nút hiển thị thông tin, hệ thống sẽ hiện lên giao diện thông tin của người dùng. Tại đây người dùng có thể cập nhật các thông tin cần thiết.



Hình 31: Giao diện các chức năng của cập nhật



Hình 32: Giao diện cập nhật thông tin người dùng

Chú thích:

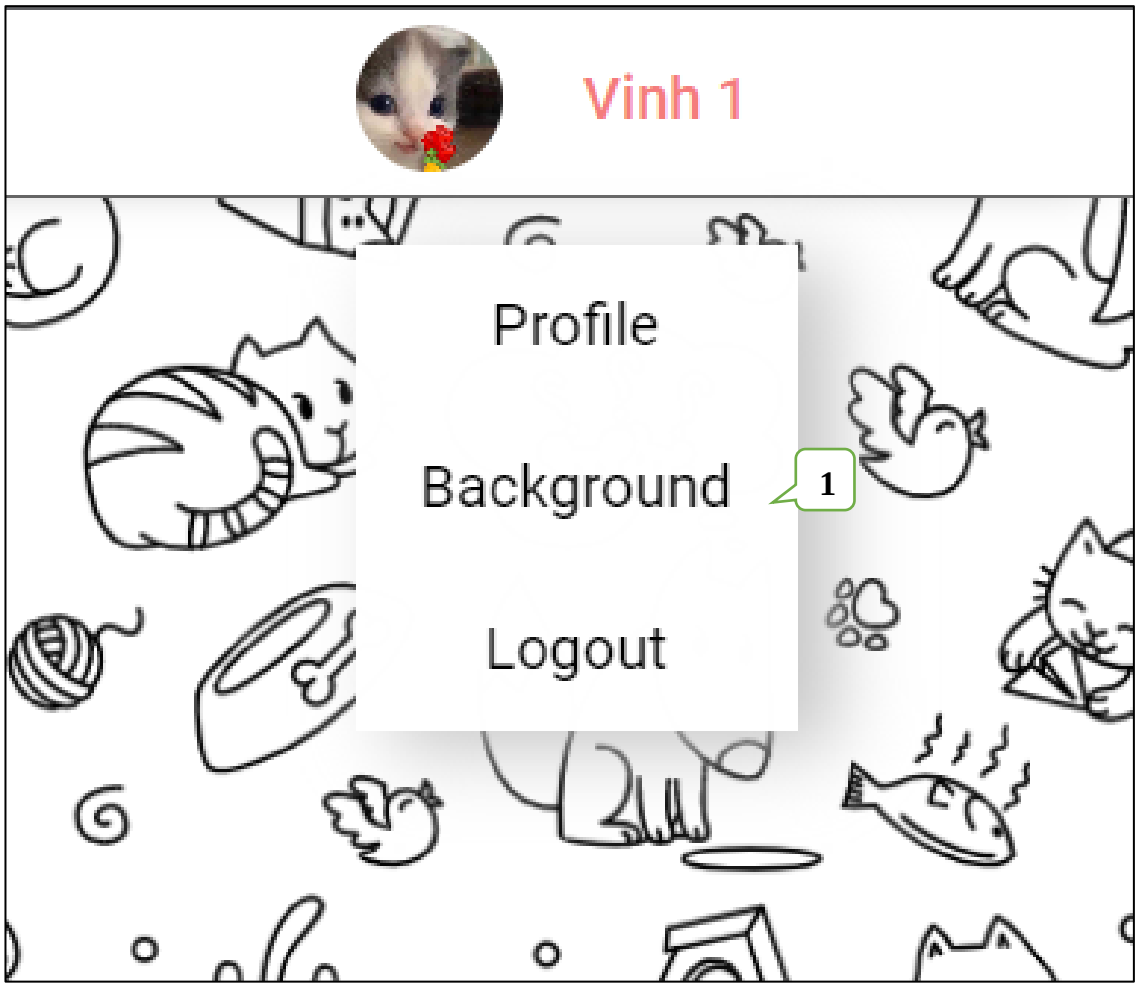
[1]: Nút gọi chức năng cập nhật thông tin ngươi dùng

[2]: Khung cập nhật thông tin người dùng

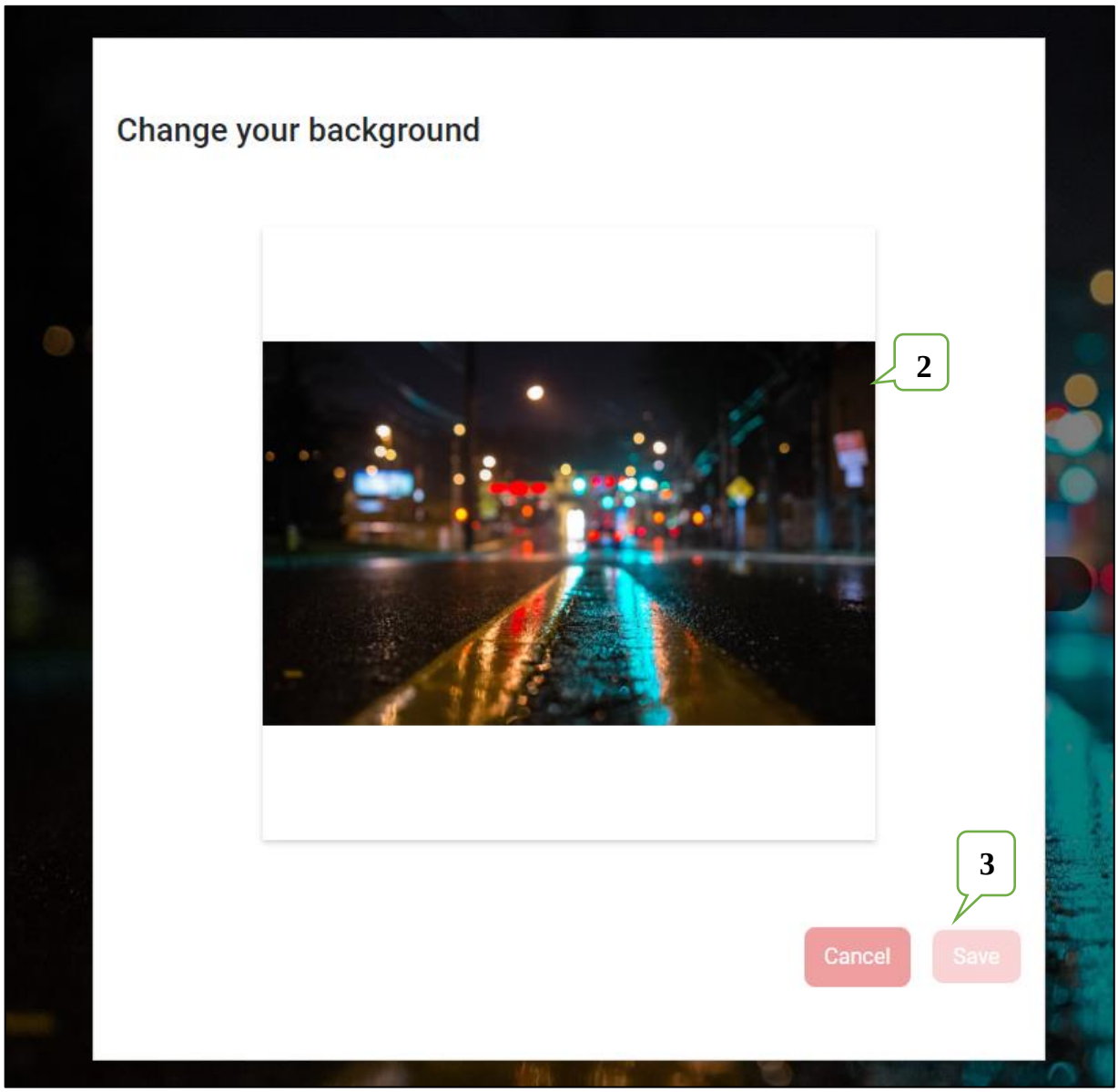
[3]: Nút lưu thông tin mới cập nhật

3.2.12. Giao diện thay đổi ảnh nền khung chat

Sau khi người dùng chọn vào nút cập nhật ảnh nền khung chat, hệ thống sẽ hiện lên giao diện để người dùng có thể cập nhật lại hình ảnh.



Hình 33: Giao diện các chức năng của cập nhật



Hình 34: Giao diện chi tiết đơn hàng thành công

Chú thích:

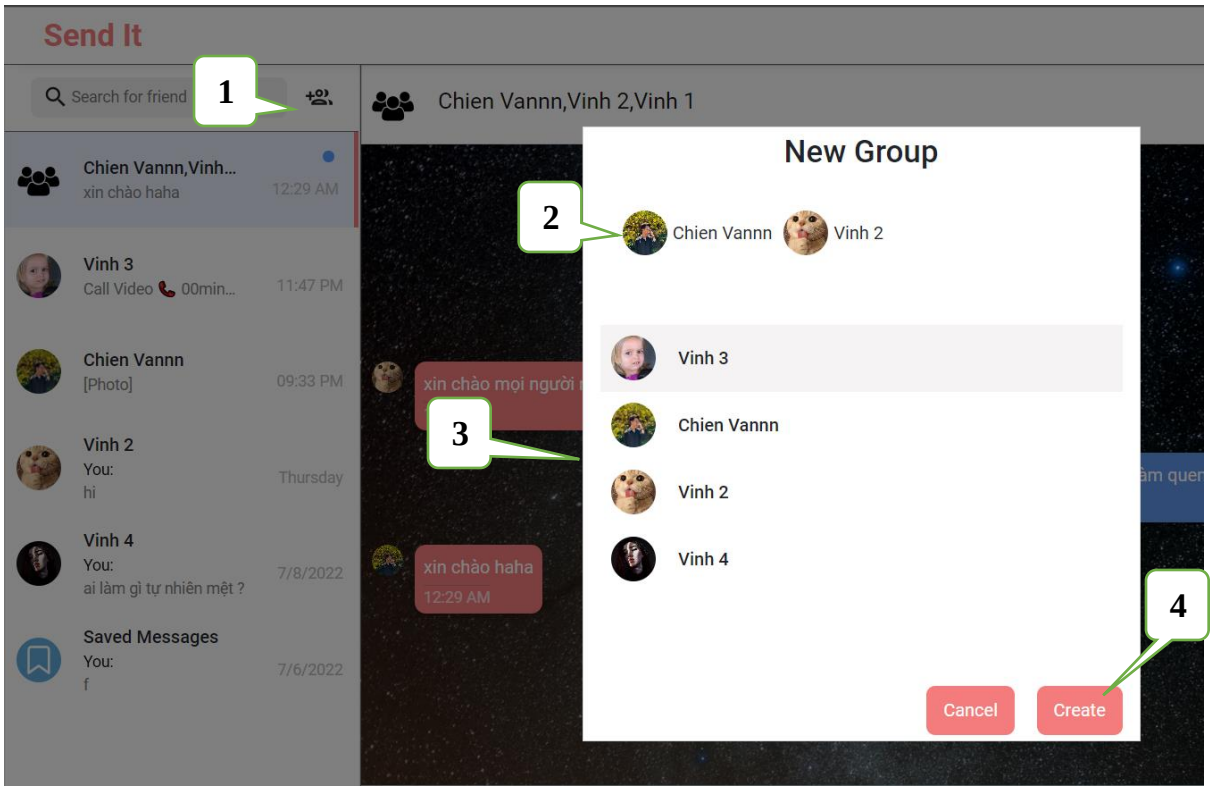
[1]: Nút gọi chức năng cập nhật ảnh nền khung chat

[2]: Hình ảnh hiện tại

[3]: Nút lưu thông tin mới cập nhật

3.2.13. Giao diện tạo nhóm chat

Bạn có thể sẽ thấy biểu tượng tạo nhóm chat gần với khung tìm kiếm như sau. Khi ấn chọn vào biểu tượng dưới bạn sẽ mở ra một khung chọn để thêm những người bạn muốn vào nhóm chat mà bạn muốn tạo. Tuy nhiên những người mà bạn có thể chọn là những người đang có liên hệ với bạn, điều này tránh việc bạn có thể mời người lạ vào nhóm chat vì hành động đó người khác có thể xem là làm phiền.



Hình 35: Giao diện tạo nhóm chat

Chú thích:

[1]: Nút tạo nhóm chat

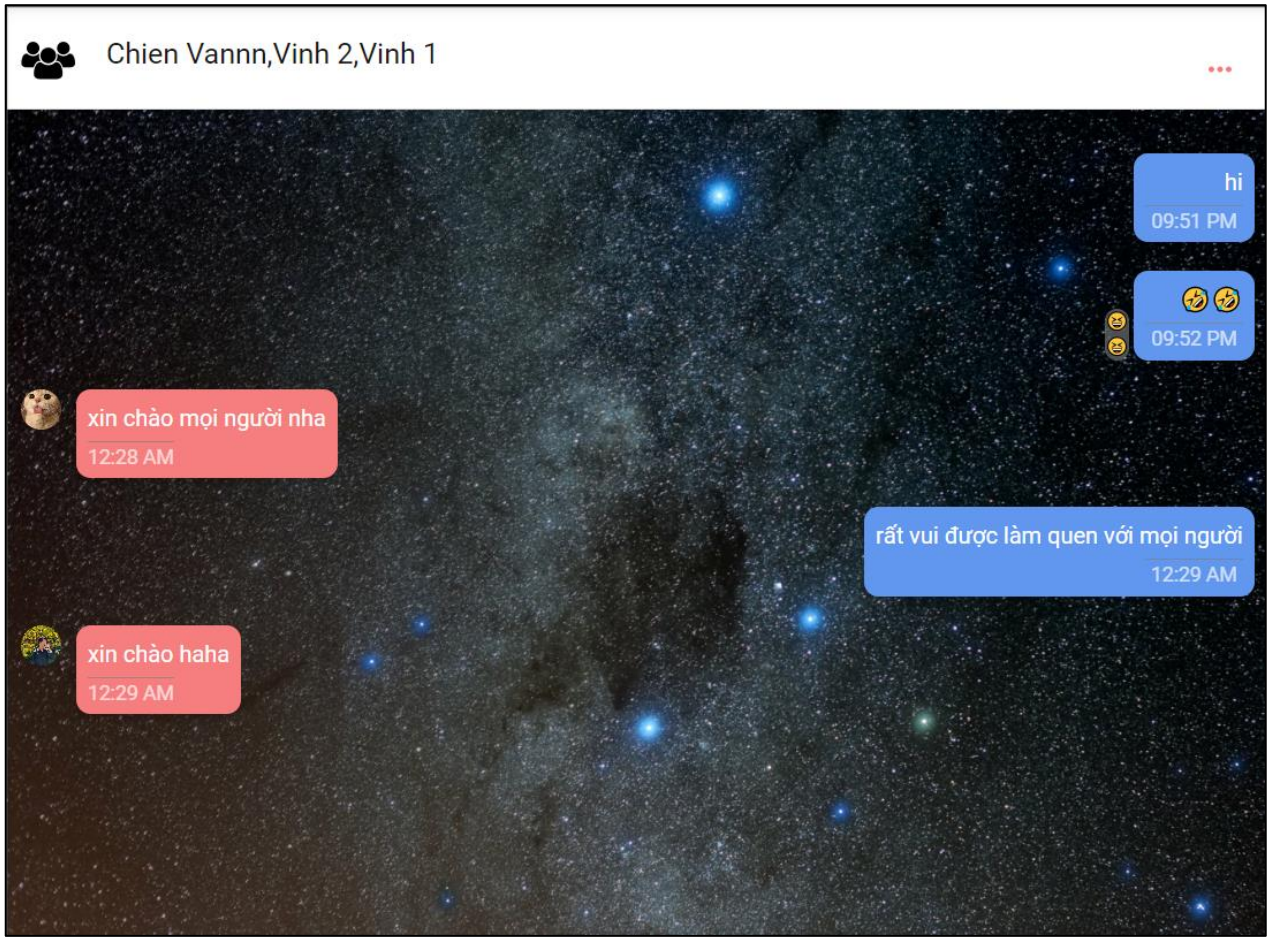
[2]: Những người được chọn để tạo nhóm chat

[3]: Những người đang liên lạc, họ có thể được bạn chọn để tạo nhóm chat

[4]: Nút tạo nhóm chat mới

3.2.14. Giao diện trong một nhóm chat

Về cơ bản nhóm chat cũng giống với cuộc trò chuyện riêng tư của hai người và nó có đầy đủ các chức năng của việc trò chuyện giữa hai người từ nhắn tin, gửi ảnh, gửi file,… ngoại trừ cuộc gọi nhóm.



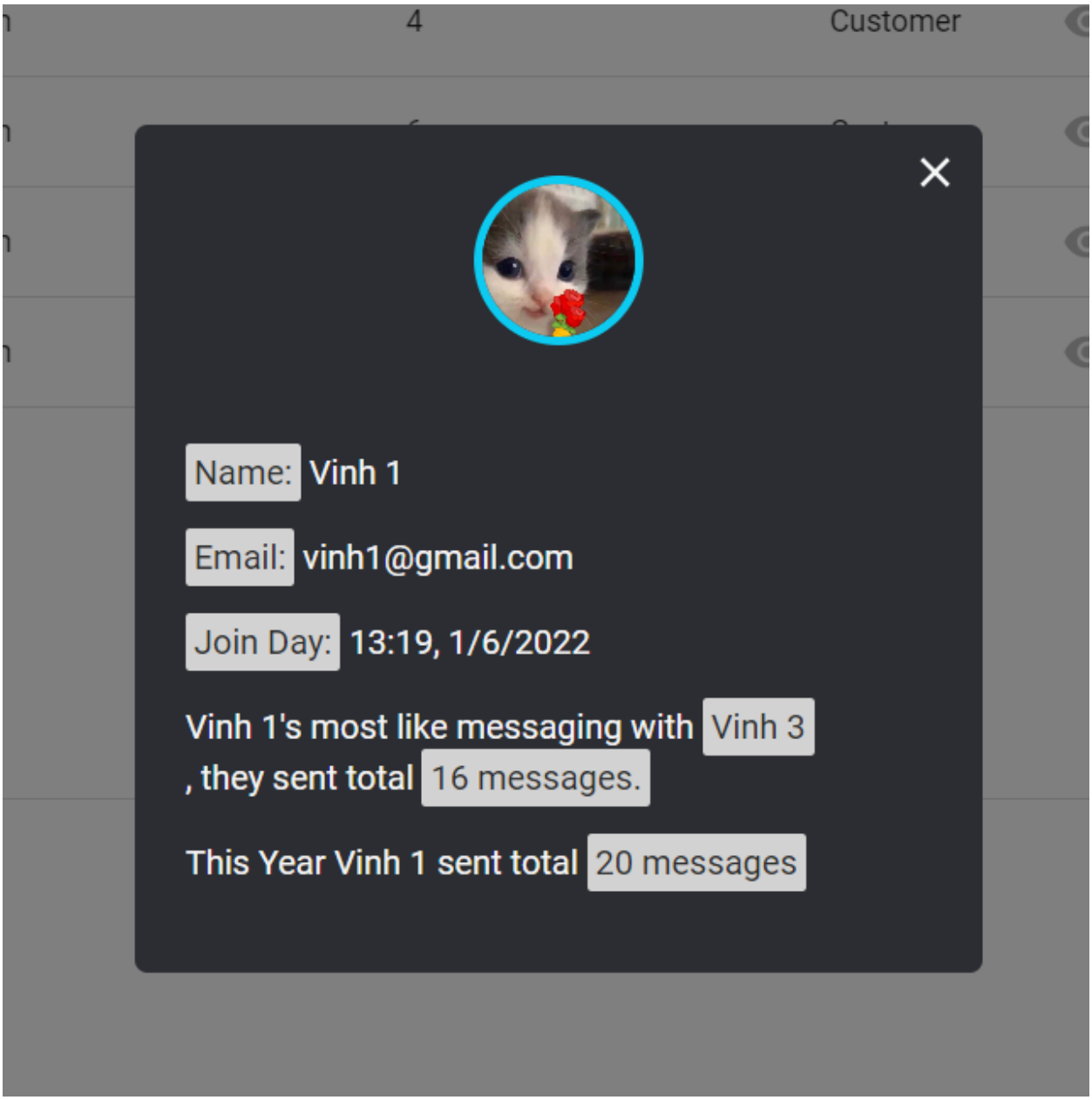
Hình 36: Giao diện chat của nhóm chat

3.3. Giao diện trang quản trị

3.3.1. Giao diện trang chủ

Ở giao diện trang chủ admin, người quản trị có thể xem được tất cả các Users đang hoạt động và có thể xem thông tin chi tiết của từng User, ngoài ra còn xem được người mà User đó trao đổi nhiều nhất hoặc User đó gửi bao nhiêu tin nhắn trong 1 n.

Hình 37: Giao diện tất cả User đã tham gia



Hình 38: Giao diện thông tin chi tiết của User

Chú thích:

[1]: Nút xem thông tin chi tiết của User

CHƯƠNG 4: TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN

4.1. Kết quả đạt được

- Website trao đổi thông tin trực tuyến Send It sau thời gian chúng em tập trung làm việc, về cơ bản đã hoàn thành những chức năng cho người dùng:

- Đối với người sử dụng webstite
  - ✓ Có thể thực hiện các chức năng cơ bản như tạo tài khoản, đăng nhập, tìm kiếm người dùng và chat (trò chuyện riêng và trò chuyện nhóm), call video, gửi hình ảnh, gửi emoji, reaction tin nhắn, thay đổi thông tin người dùng. Bên cạnh đó website sử dụng công nghệ mới (ReactJS).
- Đối với người quản trị
  - ✓ Các tác vụ quản lý người dùng cơ bản như: Xem thông tin thống kê về người dùng đó như: Người dùng đó nhắn tin với ai nhiều nhất, trong năm nay người dùng này đã nhắn bao nhiêu tin nhắn.

- Vận dụng tất cả kiến thức đã học ở trường và những kiến thức tự tìm hiểu được vào làm đồ án.
- Khả năng kết hợp kiến thức đã học và công nghệ mới tìm hiểu được.

4.2. Phần hạn chế của đề tài

Do thời gian với kiến thức, kinh nghiệm còn giới hạn nên đồ án của nhóm chúng em không tránh khỏi những thiếu sót. Bên cạnh đó thời gian phát triển ứng dụng và kiểm thử ứng dụng không nhiều nên những lỗi làm crash ứng dụng vẫn chưa được phát hiện hết.

Thời gian phát triển đề tài cộng với việc bắt đầu tìm hiểu đề tài và công nghê mới là một thử thách vô cùng lớn đối với chúng em để có thể làm cho website có thể hoàn thành một cách hoàn hảo, tuy nhiên chúng em đã có gắng hết sức và làm tốt nhất có thể.

Đề tài đã hoàn thành các chức năng căn bản, tuy nhiên bên cạnh đó vẫn có một số khuyết điểm :

- ✓ Chưa tối ưu các component
- ✓ Giao diện chưa được tối ưu
- ✓ Chưa sử dụng các hiệu ứng chuyển trang…
- ✓ Chưa hoàn thành được các chức năng nâng cao đã đề ra

4.3. Hướng phát triển

Trong tương lai nhóm chúng em sẽ tập trung vào phát triển những chức năng nâng cao tập trung vào việc liên hệ nhóm như:

- ✓ Phát triển tiếp các tính năng call video nhóm
- ✓ Thêm tính năng share screen khi call video
- ✓ Tích hợp tạo vote trong cuộc trò chuyện(bỏ phiếu)
- ✓ Tối ưu các component
- ✓ Hiện thông báo khi có tin nhắn hoặc cuộc gọi đến trên màn hình

TÀI LIỆU THAM KHẢO

- **Giao diện:**
  - ✓ **React-Bootstrap: https://react-bootstrap.github.io**
  - ✓ **Material-UI: https://material-ui.com**
  - ✓ **Styled-component:** https://styled-components.com
  - ✓ **SCSS/SASS: https://sass-lang.com/guide**
- **Package: https://www.npmjs.com**
- **Document:**
  - ✓ **ReactJS: https://reactjs.org**
  - ✓ **MongoDB: https://viblo.asia/p/tim-hieu-ve-mongodb-4P856ajGlY3**
  - ✓ **SocketIO: https://socket.io**
  - ✓ **NodeJS: https://nodejs.org/en/docs/**
  - ✓ **Mongoose: https://mongoosejs.com/docs/**
  - ✓ **Youtube: https://www.youtube.com/**