

**ĐẠI HỌC CẦN THƠ**  
**TRƯỜNG CÔNG NGHỆ THÔNG TIN – TRUYỀN THÔNG**  
**KHOA MẠNG MÁY TÍNH VÀ TRUYỀN THÔNG**



**NIÊN LUẬN CƠ SỞ**  
**ỨNG DỤNG CHAT CƠ BẢN**

**Sinh viên thực hiện:**

**Họ tên:** Phạm Minh Sáng  
**MSSV:** B2110976  
**Lớp:** DI21T9A2  
**Ngành:** Mạng Máy Tính Và Truyền  
Thông Dữ Liệu  
**Khóa:** 47

**Giáo viên hướng dẫn:**

**Giảng viên:** TS. Hà Duy An

**HỌC KỲ 2, 2023-2024**

## This image shows a single sheet of white paper with horizontal ruling lines. The lines are evenly spaced and run across the width of the page. There are no margins, text, or other markings on the paper.

## MỤC LỤC

Chương 1: Tổng quan.....	6
I. Mô tả bài toán.....	6
1. Mô tả.....	6
2. Các chức năng chính.....	6
3. Các yêu cầu phi chức năng.....	7
4. Môi trường vận hành .....	7
5. Các giả định phụ thuộc .....	7
II. Mục tiêu cần đạt được.....	7
III. Hướng giải quyết .....	8
IV. Kế hoạch thực hiện.....	8
Chương 2: Lý thuyết.....	9
I. Khái niệm .....	9
1. HTML .....	9
2. CSS .....	9
3. JavaScript .....	9
4. JQuery.....	9
5. Node.js .....	10
6. Express.js .....	10
7. EJS .....	10
8. Socket.IO .....	10
9. MySQL .....	11
II. Ứng dụng .....	11
Chương 3: Kết quả ứng dụng .....	12
I. Thiết kế thành phần dữ liệu.....	12
1. Sơ đồ hoạt vụ (Usecase) .....	12
2. Mô hình mức quan niệm (CDM).....	13
3. Mô hình mức luận lý (LDM).....	14
4. Mô hình mức vật lý (PDM) .....	14
II. Thiết kế giao diện .....	16
1. Trang đăng ký.....	16
2. Trang đăng nhập .....	16
3. Trang chat .....	17
Chương 4: Kết luận – đánh giá.....	20

I. Kết quả đạt được .....	20
II. Thu hoạch.....	20
III. Ưu điểm .....	20
IV. Hạn chế - nguyên nhân.....	20
V. Hướng phát triển.....	21

## DANH MỤC HÌNH ẢNH

Hình 1: Sơ đồ hoạt vụ (Usecase).....	12
Hình 2: Mô hình mức quan niệm (CDM).....	13
Hình 3: Mô hình mức vật lý (PDM).....	14
Hình 4: Trang đăng ký.....	16
Hình 5: Trang đăng nhập.....	16
Hình 6: Trang chat - Home.....	17
Hình 7: Trang chat - Tìm kiếm người dùng để kết bạn.....	17
Hình 8: Trang chat - Lời mời kết bạn.....	18
Hình 9: Trang chat - Gửi và nhận tin nhắn.....	18
Hình 10: Trang chat - Hiện thị thông báo.....	18
Hình 11: Trang chat - Hiện thị có người dùng đang nhập tin .....	19
Hình 12: Trang chat - Tìm kiếm người dùng muốn gửi tin.....	19
Hình 13: Trang chat - Hiện thị kết quả tìm kiếm người dùng.....	19

## **DANH MỤC BẢNG**

Bảng 1: Kế hoạch thực hiện .....	8
Bảng 2: PDM - USERS .....	15
Bảng 3: PDM - FRIENDS .....	15
Bảng 4: PDM - MESSAGES .....	15

## CHƯƠNG 1: TỔNG QUAN

### I. MÔ TẢ BÀI TOÁN

#### 1. Mô tả

Hiện nay, việc trao đổi thông tin, dữ liệu một cách nhanh chóng và tức thì là một trong những vấn đề thiết yếu của xã hội hiện nay. Do vậy, khi chúng ta trao đổi thông tin với nhau bằng văn bản trên mạng xã hội cũng cần được tức thì để hai bên có thể cập nhật được thông tin một cách tốt nhất và không phải chờ đợi quá lâu.

Với ứng dụng Chat cơ bản, lần đầu khi người dùng truy cập sẽ được dẫn đến trang đăng nhập, để người dùng có thể đăng nhập được vào ứng dụng Chat thì cần có một tài khoản cá nhân cho riêng mình với các thông tin cơ bản như: *tên tài khoản người dùng, mật khẩu*. Với việc bảo mật thông tin cá nhân của người dùng ngày càng quan trọng, cho nên mật khẩu của người dùng sẽ được mã hóa để tránh rò rỉ hoặc đánh cắp thông tin, mỗi tên tài khoản trong hệ thống tồn tại là duy nhất.

Sau khi đăng ký thành công một tài khoản, ứng dụng sẽ điều hướng đến trang đăng nhập, khi người dùng đã có cho mình một tài khoản cá nhân, họ sẽ tiến hành đăng nhập vào sử dụng ứng dụng Chat điền đầy đủ và đúng với những gì mà người dùng đã đăng ký trước đó.

Khi đăng nhập thành công vào ứng dụng một hộp thoại Chat hiện lên và dẫn đến phần thêm đoạn hội thoại với người dùng khác trong ứng dụng, lần đầu tiên người dùng sử dụng ứng dụng họ sẽ không tồn tại một hộp thoại nào. Lúc này người dùng sẽ tiến hành tìm và kết bạn với người dùng khác trong ứng dụng, chờ đợi người dùng được kết bạn đồng ý lời mời thì cả hai mới có thể bắt đầu tiến hành trò chuyện với nhau. Lời mời kết bạn sẽ được lưu vào bảng với thông tin như: *mã lời mời, người gửi kết bạn, người chấp nhận lời mời, tình trạng lời mời*.

Bắt đầu cuộc trò chuyện người dùng cần chọn vào hộp thoại của người cần nhận tin nhắn, sau đó người dùng sẽ nhập nội dung tin nhắn cần gửi đi vào ô nhập tin, nội dung tin nhắn sẽ được lưu vào bảng gồm: *mã tin nhắn, người gửi, người nhận, nội dung tin nhắn, thời gian, tình trạng nhận tin nhắn*. Trong lúc trò chuyện thì người dùng có thể thấy được là người kia có đang nhắn một đoạn tin nào đó không, khi có một tin nhắn mới hộp thoại của người dùng sẽ có một chấm đỏ hiển thị thông báo mới của người dùng khác gửi đến. Sau khi người dùng không sử dụng ứng dụng nữa họ có thể đăng xuất ra khỏi ứng dụng và sẽ được chuyển về trang đăng nhập.

#### 2. Các chức năng chính

- + Đăng nhập.
- + Đăng ký.
- + Tìm kiếm người dùng muốn kết bạn.
- + Gửi lời mời kết bạn và chấp nhận lời mời kết bạn.
- + Tìm người dùng đã kết bạn.
- + Chọn hộp thoại muốn nhắn tin.

- + Gửi và nhận tin nhắn.
- + Hiện thị thông báo tin nhắn chưa đọc.
- + Đăng xuất.

### 3. Các yêu cầu phi chức năng

- + Giao diện thân thiện, hài hòa màu sắc, dễ dàng sử dụng.
- + Phông chữ: sử dụng phông chữ Roboto tiếng Việt, tiêu chuẩn Unicode.
- + Có tính mở rộng để thuận tiện cho việc phát triển ứng dụng về sau.
- + Hiệu quả về mặt tốc độ xử lý cập nhật, tra cứu thông tin trả về cho người dùng không quá 5 giây.
- + Bảo mật tài khoản của người dùng bằng các giải thuật tránh các trường lộ, đánh cắp tài khoản.

### 4. Môi trường vận hành

Máy tính cá nhân

- + Phần cứng:
  - CPU: Intel Core i5 10300H
  - Ram: 16GB
  - SSD: 512GB
- + Phần mềm:
  - Hệ điều hành Window 11
  - Visual Studio Code
  - Node.js 18.18.2
  - MySQL Workbench 8.0

### 5. Các giả định phụ thuộc

- + Máy chủ có thể bị nhiễm virus hoặc hacker tấn công làm hỏng dữ liệu.
- + Phần cứng bị hỏng đột ngột
- + Tài khoản người dùng có thể bị tấn công
- + Nguồn điện không ổn định hoặc thiếu giải pháp cho nguồn điện dự phòng
- + Đường truyền kết nối Internet hỏng hoặc không ổn định làm mất nối kết
- + Các trình duyệt không hỗ trợ tương thích ứng dụng

## II. MỤC TIÊU CẦN ĐẠT ĐƯỢC

Mục tiêu tổng quan: Sau khi thực hiện đề tài có thể hiểu rõ được cách thức hoạt động của mô hình client – server, nghiên cứu về Node.js cũng như là hệ quản trị cơ sở dữ liệu MySQL. Bên cạnh đó là nghiên cứu về Socket.IO xử lý các tác vụ theo thời gian thực, cách phân tích bài toán, giải quyết vấn đề.

Mục tiêu cụ thể: Xây dựng được một ứng dụng web có thể kết bạn, trao đổi thông tin văn bản giữa hai người dùng với nhau theo thời gian thực trên nền tảng là Node.js kết hợp thư viện Socket.IO cùng với MySQL để lưu trữ dữ liệu.



### III. HƯỚNG GIẢI QUYẾT

Sử dụng Node.js làm server để nhận thông tin từ người dùng như: thông tin tài khoản đăng nhập, danh sách bạn bè trao đổi tin nhắn trên ứng dụng, các đoạn tin nhắn của người dùng gửi đi, sau đó lưu trữ vào trong hệ quản trị cơ sở dữ liệu MySQL. Kết hợp sử dụng thư viện Socket.IO để có thể phản hồi tin nhắn ngược lại cho người dùng cũng như là người nhận được tin nhắn tức thì theo thời gian thực.

### IV. KẾ HOẠCH THỰC HIỆN

Tuần	Công việc	Thời gian báo cáo
1 - 4	Tìm hiểu công nghệ, phạm vi thực hiện.	Tuần 4
5 - 6	Triển khai giao diện chat máy client bằng HTML, CSS.	
7 - 8	Xây dựng server socket trao đổi tin nhắn giữa 2 client với nhau.	
9	Xử lý sự kiện kết nối tên người dùng tới ứng dụng, thoát khỏi người dùng hiện tại, hiển thị số người dùng	Tuần 9
10 - 11	Xử lý sự kiện người dùng tìm kiếm gửi lời mời, chấp nhận lời kết bạn, tìm kiếm người cần trao đổi tin nhắn	
12 - 13	Sửa lỗi và bổ sung các chức năng.	Tuần 12
14	Hoàn thiện demo chương trình và sửa lại lỗi nếu có.	Tuần 14
15	Thiết kế slide báo cáo.	
16	Làm văn bản in nên cuốn niên luận	Tuần 16
17	Chuẩn bị các khâu báo cáo.	
18	Hoàn thành báo cáo, trình bày kết quả đạt được.	Tuần 18

Bảng 1: Kế hoạch thực hiện

## CHƯƠNG 2: LÝ THUYẾT

### I. KHÁI NIỆM

#### 1. HTML

HTML là một trong những hệ thống mã hóa quan trọng nhất trên thế giới. Đằng sau mỗi trang web là một trang mã HTML. HTML hay Hypertext Markup Language (Ngôn ngữ đánh dấu siêu văn bản), là ngôn ngữ đánh dấu được sử dụng để tạo các trang web. Ngôn ngữ đánh dấu là ngôn ngữ máy tính được tạo ra để làm việc với văn bản. Ngôn ngữ đánh dấu cung cấp định dạng cho tệp văn bản. Điều này xác định kiểu văn bản và bố cục của trang web.

Mã định dạng được gọi là thẻ, chúng sử dụng các ký tự đặc biệt để giữ các lệnh định dạng - thẻ. Trong HTML, sử dụng cặp thẻ bắt đầu và kết thúc, ví dụ như: bắt đầu trang html bằng `<html>` và kết thúc bằng `</html>`.

#### 2. CSS

CSS (Cascading Style Sheets) được sử dụng để tìm và định dạng lại các phần tử được tạo ra bởi các ngôn ngữ đánh dấu. Cách mà trang web được hiển thị như thế nào là dựa vào cách trình bày CSS. Phong chữ, màu sắc, hình nền, khoảng cách dòng, bố cục trang, v.v., đều được kiểm soát bằng CSS. Thậm chí có thể thêm các hiệu ứng đặc biệt và hoạt ảnh cơ bản vào trang web. CSS cũng cung cấp các phương pháp để kiểm soát cách trình bày tài liệu trong các ngữ cảnh không phải là trình duyệt, chẳng hạn như ở dạng in hoặc trình đọc màn hình.

#### 3. JavaScript

JavaScript là ngôn ngữ kịch bản giúp bổ sung tính tương tác và hành vi cho các trang web, bao gồm: kiểm tra hợp lệ các mục nhập biểu mẫu, hoán đổi kiểu cho một phần tử hoặc toàn bộ trang web, tự động tải các nguồn cấp dữ liệu cuộn với nhiều nội dung hơn, làm cho trình duyệt ghi nhớ thông tin về người dùng, xây dựng các tiện ích giao diện.

Thuật ngữ tập lệnh DOM được sử dụng liên quan đến JavaScript. DOM là viết tắt của Document Object Mode và nó đề cập đến danh sách tiêu chuẩn hóa các thành phần trang web có thể được truy cập và thao tác bằng JavaScript (hoặc ngôn ngữ kịch bản khác).

#### 4. JQuery

Jquery là thư viện JavaScript được thiết kế để đơn giản hóa việc lập trình giao diện web, cho phép tương tác các phần tử DOM trên HTML, kiểm soát các hành động của người dùng một cách dễ dàng. Với các tính năng chính như: truy cập, thay đổi vào các phần tử DOM, sửa đổi, tạo hoạt ảnh giao diện trang web, trả lời tương tác của người dùng, truy xuất thông tin từ máy chủ thông qua AJAX

## 5. Node.js

Node.js không phải là một ngôn ngữ mới và không chỉ là một framework trên Javascript. Node.js là một môi trường JavaScript runtime mã nguồn mở và đa nền tảng. NodeJS là một nền tảng được xây dựng trên “Chrome 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 đòi đầu của JavaScript mở rộng nó từ một thứ bạn chỉ chạy được trên trình duyệt thành một thứ bạn có thể chạy “ở bất cứ nơi nào” như một ứng dụng độc lập. Node.js sử dụng cơ chế bất đồng bộ, thực thi đơn luồng, không nghẽn và sử dụng bộ nhớ một cách hiệu quả.

## 6. Express.js

Express.js là một web application framework tối giản và linh hoạt, cung cấp một bộ tính năng mạnh mẽ để phát triển các ứng dụng di động và web dựa trên Node.js. Là một trong những web application framework phổ biến nhất trong hệ sinh thái Node.js. Express.js cung cấp tất cả các tính năng của một web application framework, như templating, xử lý tệp tĩnh, kết nối với cơ sở dữ liệu SQL và NoSQL.

## 7. EJS

EJS viết tắt của Embedded JavaScript là một ngôn ngữ mẫu (template language) được sử dụng trong phát triển ứng dụng web. Nó giúp phát triển các trang web động bằng cách cho phép tạo ra các mẫu HTML được kết hợp với mã JavaScript. EJS thường được sử dụng cùng với Node.js và các framework web như Express để tạo giao diện người dùng động dựa trên dữ liệu từ máy chủ.

## 8. Socket.IO

Socket.IO là một thư viện mã nguồn mở trong Node.js được phát triển từ năm 2010. Mục đích nhằm tạo ra một môi trường giao tiếp thời gian thực và hai chiều giữa máy chủ và máy khách trên Internet. Nền tảng cho phép gửi và nhận dữ liệu một cách tức thì và tối ưu hóa việc sử dụng tài nguyên. Tích hợp với nhiều ngôn ngữ khác nhau như PHP, ASP.NET, Node.js.

Socket.IO xây dựng dựa trên Engine.IO và hỗ trợ việc thiết lập kết nối bảo mật thông qua sự hỗ trợ của các phương thức giao tiếp tốt hơn như Websocket. Ngoài ra, Socket.IO cũng cung cấp một số cơ chế bảo mật bổ sung, bao gồm:

- + Bảo mật dựa trên TLS/SSL: Socket.IO hỗ trợ kết nối an toàn thông qua việc sử dụng giao thức HTTPS và WSS (WebSocket over TLS/SSL). Điều này có tác dụng mã hóa dữ liệu trên đường truyền và ngăn chặn các cuộc tấn công theo dõi hoặc thay đổi dữ liệu.
- + Hỗ trợ cân bằng tải và proxy: Socket.IO có thể dễ dàng tích hợp với các cơ chế cân bằng tải và proxy để tăng cường bảo mật và khả năng chịu tải của hệ thống.
- + Hỗ trợ cho tường lửa: Socket.IO cung cấp các cơ chế để vượt qua các tường lửa và phần mềm chống virus thông qua việc sử dụng các cổng mạng phổ biến như cổng 443 (HTTPS).

Cùng với một số tính năng đặc trưng khác của Socket.IO như:

- + Cơ chế kết nối server tự động: tính năng tự động kết nối đến server giúp việc giao tiếp giữa client và server trở nên linh hoạt và dễ dàng hơn. Khi mất kết nối trong quá trình khởi chạy, Socket.IO sẽ liên tục cố gắng kết nối đến server cho đến khi kết nối lại thành công hoặc đạt đến ngưỡng thời gian chờ đặt trước.
- + Mã hóa nhị phân: giúp người dùng có thể truyền các dữ liệu nhị phân như hình ảnh video hoặc âm thanh.
- + Tiện ích tạo kênh và phòng.

Phương thức hoạt động: Socket.IO hoạt động bằng cách sử dụng mô hình giao thức "Long Polling" hoặc WebSockets để tạo và duy trì kết nối real time giữa client và server. Quy trình này hoạt động như sau:

- + Trong trường hợp trình duyệt hoặc client không hỗ trợ WebSockets, Socket.IO sẽ sử dụng phương pháp long-polling. Khi client gửi yêu cầu tới server, server sẽ không trả về ngay lập tức mà duy trì kết nối. Hệ thống chỉ trả về khi có dữ liệu mới hoặc sau một khoảng thời gian nhất định. Sau đó, client sẽ gửi yêu cầu mới ngay lập tức để duy trì kết nối. Quá trình này tiếp diễn để tạo ra một luồng thông tin realtime giữa client và server qua HTTP.
- + Trong trường hợp trình duyệt hoặc client hỗ trợ WebSockets, Socket.IO sẽ sử dụng giao thức này để tạo kết nối full-duplex giữa client và server. Điều này cho phép cả server và client gửi, nhận dữ liệu một cách tức thì mà không cần phải tạo các yêu cầu HTTP mới.

## 9. MySQL

MySQL là một hệ thống quản lý cơ sở dữ liệu quan hệ mã nguồn mở (RDBMS) dựa trên ngôn ngữ truy vấn có cấu trúc (SQL) được phát triển, phân phối và hỗ trợ bởi tập đoàn Oracle. Hoạt động dựa trên mô hình tiêu chuẩn là Client (Máy khách) – Server (Máy chủ). MySQL chạy trên hầu hết tất cả các nền tảng, bao gồm cả Linux, UNIX và Windows. MySQL thường được kết hợp với các ứng dụng web.

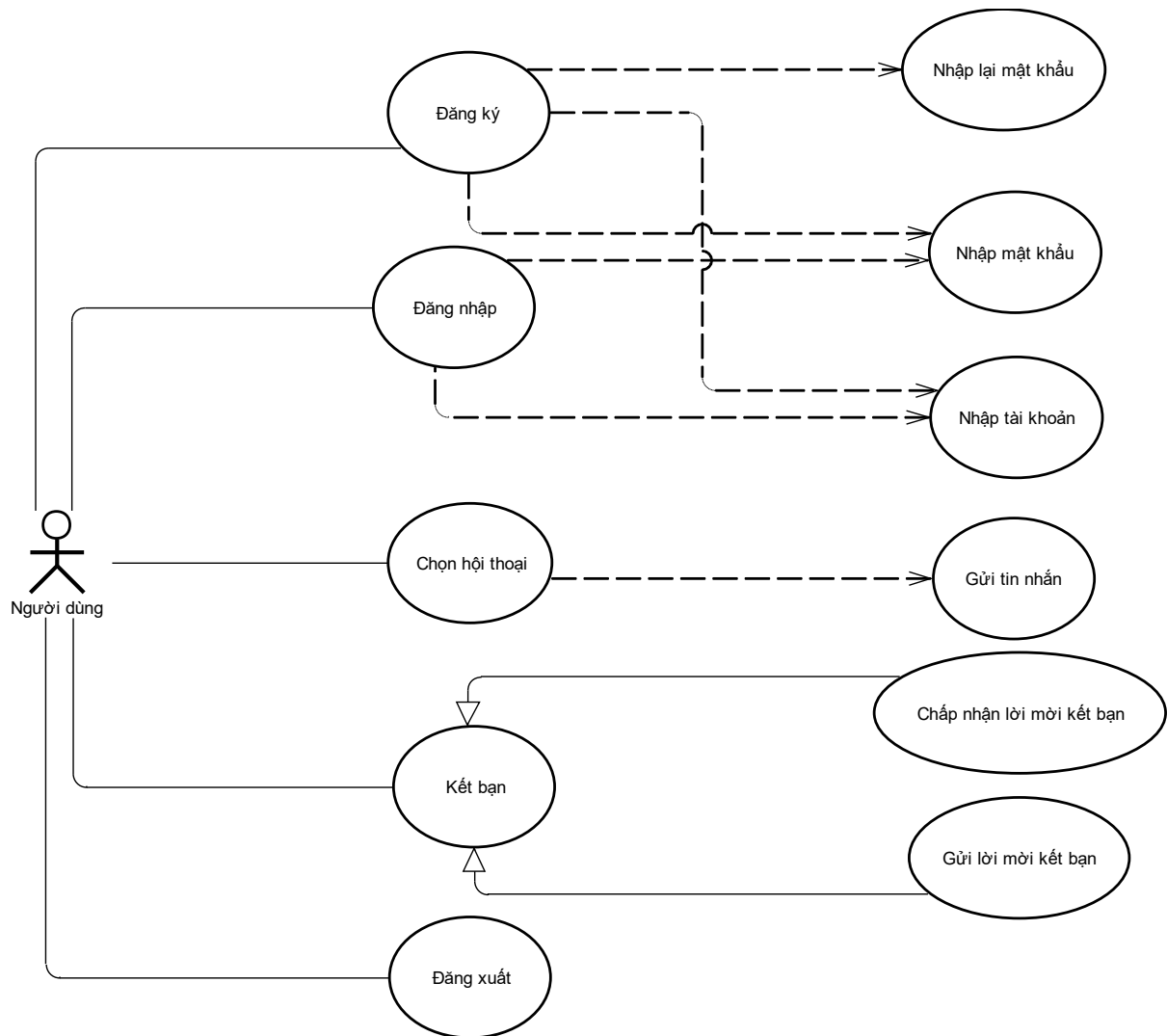
## II. ỨNG DỤNG

Kết hợp HTML, CSS, JavaScript để làm một giao diện đăng ký, đăng nhập, Chat hoàn chỉnh, thân thiện với người dùng, có thể dễ dàng sử dụng khi tiếp xúc ứng dụng. Sử dụng mẫu trên vào EJS để kết hợp lấy dữ liệu trả về từ phía server. Tạo một server Node.js để xử lý thông tin người dùng gửi lên server cùng với đó là lưu trữ những thông tin ấy vào hệ quản trị cơ sở dữ liệu MySQL để dễ dàng truy xuất, tạo mới, sửa đổi. Trong quá trình người dùng trao đổi tin nhắn văn bản với người dùng khác thì sử dụng kết hợp Node.js và thư viện Socket.IO để việc gửi, nhận thông tin theo thời gian thực một cách nhanh chóng, hiệu quả.

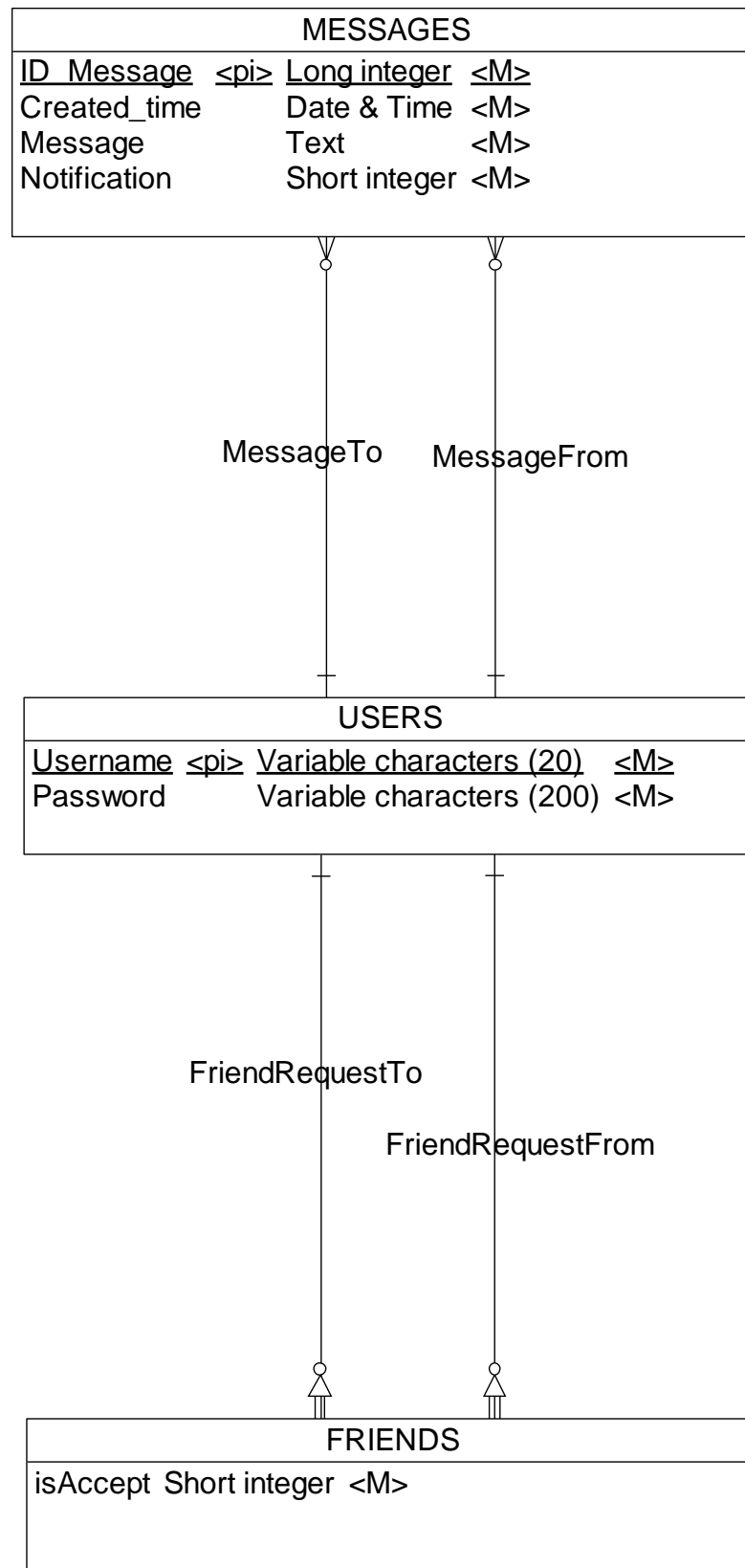
## CHƯƠNG 3: KẾT QUẢ ỨNG DỤNG

### I. THIẾT KẾ THÀNH PHẦN DỮ LIỆU

#### 1. Sơ đồ hoạt vụ (Usecase)



Hình 1: Sơ đồ hoạt vụ (Usecase)

**2. Mô hình mức quan niệm (CDM)**

Hình 2: Mô hình mức quan niệm (CDM)

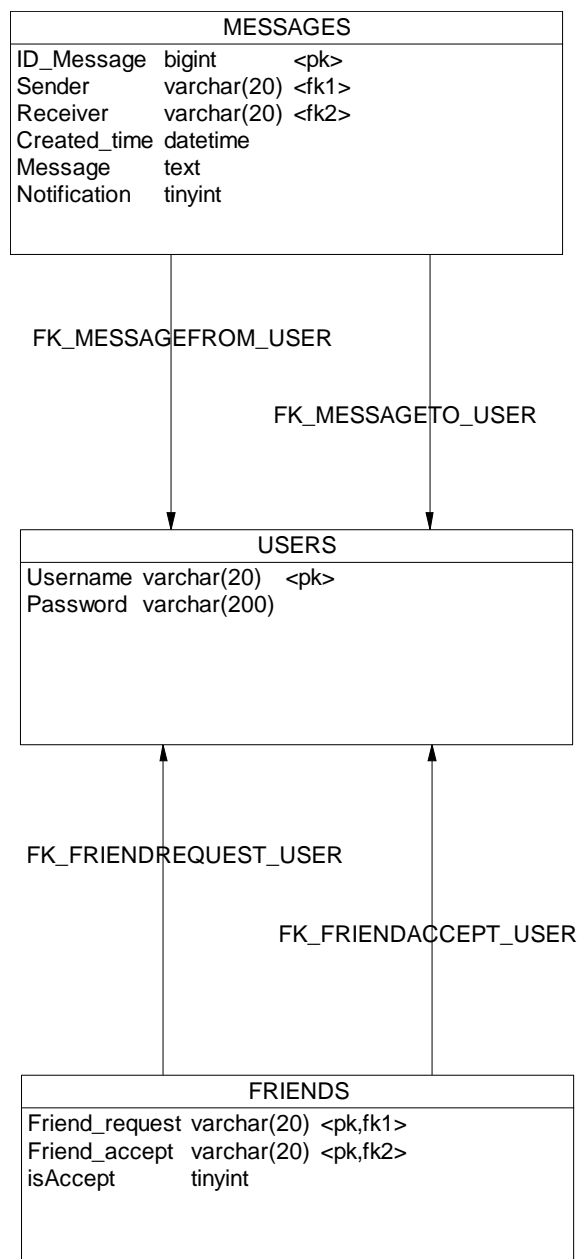
### 3. Mô hình mức luận lý (LDM)

- + USERS(Username, Password)
- + FRIENDS(Friend\_request, Friend\_accept, isAccept)
- + MESSAGES(ID\_Message, Sender, Receiver, Created\_time, Message, Notification)

Các ràng buộc toàn vẹn tham chiếu

- + FRIEND(Friend\_request) → USERS(Username)
- + FRIEND(Friend\_accept) → USERS(Username)
- + MESSAGES(Sender) → USERS(Username)
- + MESSAGES(Receiver) → USERS(Username)

### 4. Mô hình mức vật lý (PDM)



Hình 3: Mô hình mức vật lý (PDM)

## Bảng USERS

Bảng 2: PDM - USERS

Tên thuộc tính	Kiểu dữ liệu	Khóa chính	Khóa ngoại	Giá trị mặc định	Identity	NN	Diễn giải
Username	varchar(20)	x				x	Tên tài khoản của người dùng.
Password	varchar(200)					x	Mật khẩu của người dùng được mã hóa.

## Bảng FRIENDS

Bảng 3: PDM - FRIENDS

Tên thuộc tính	Kiểu dữ liệu	Khóa chính	Khóa ngoại	Giá trị mặc định	Identity	NN	Diễn giải
Friend_request	varchar(20)	x	x			x	Tên người dùng đã gửi lời mời kết bạn.
Friend_accept	varchar(20)	x	x			x	Tên người dùng được gửi lời mời kết bạn, chấp nhận lời mời kết bạn.
isAccept	tinyint			'0'		x	Trạng thái gửi yêu cầu kết bạn. '0' là đã gửi nhưng chưa chấp nhận, '1' là đã chấp nhận yêu cầu.

## Bảng MESSAGES

Bảng 4: PDM - MESSAGES

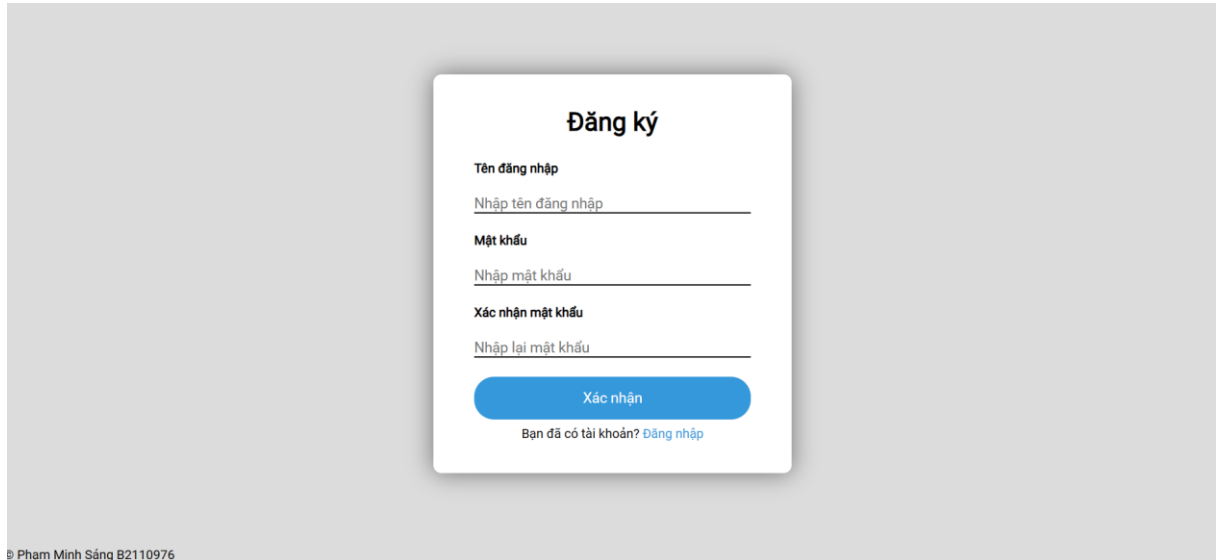
Tên thuộc tính	Kiểu dữ liệu	Khóa chính	Khóa ngoại	Giá trị mặc định	Identity	NN	Diễn giải
ID_Message	bigint	x				x	Khóa chính của tin nhắn.
Sender	varchar(20)		x			x	Tên người dùng đã gửi tin nhắn.
Receiver	varchar(20)		x			x	Tên người dùng nhận được tin nhắn.
Created_time	datetime					x	Thời gian mà tin nhắn được gửi đi.
Message	text					x	Nội dung tin nhắn.
Notification	tinyint			'1'		x	Trạng thái của tin nhắn. '1' là tin nhắn chưa được đọc, '0' tin nhắn đã đọc.



## II. THIẾT KẾ GIAO DIỆN

### 1. Trang đăng ký

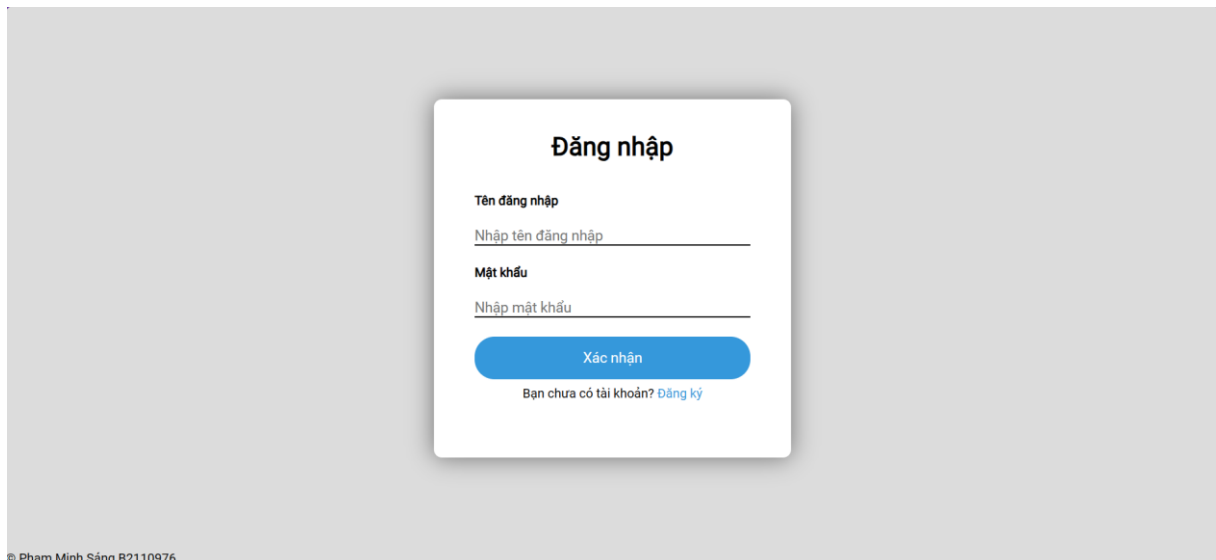
Đăng ký tài khoản để sử dụng ứng dụng với các thông tin bao gồm: tên đăng nhập, mật khẩu và nhập lại mật khẩu lần nữa để xác nhận.



Hình 4: Trang đăng ký

### 2. Trang đăng nhập

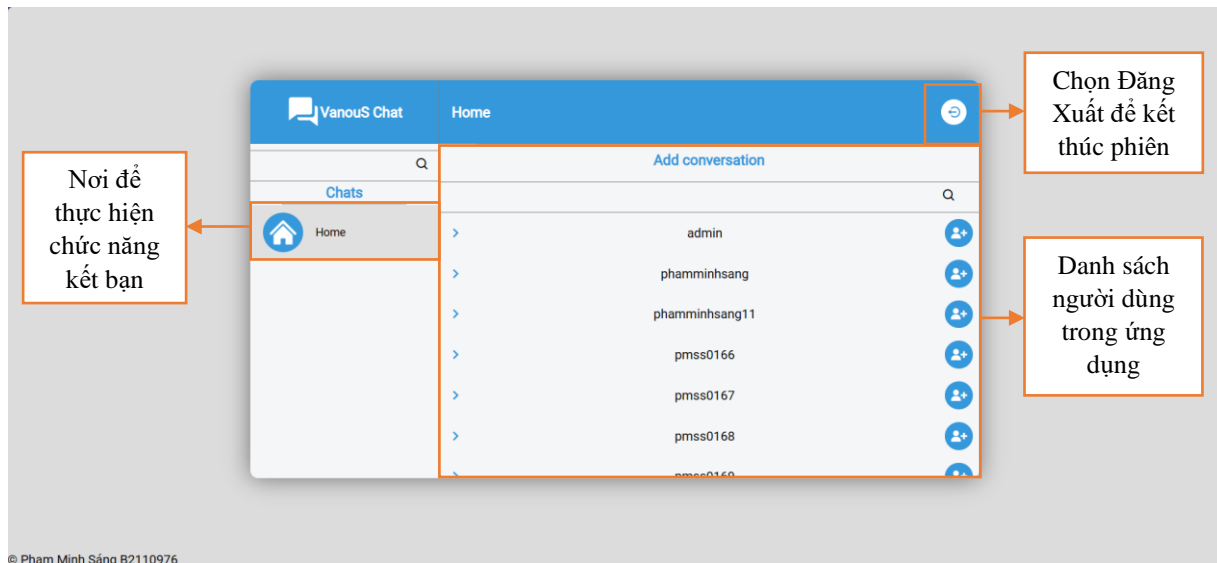
Đăng nhập bằng tài khoản mà người dùng đã đăng ký trước đó để có thể sử dụng ứng dụng và bắt đầu kết bạn, trao đổi tin nhắn với người dùng khác.



Hình 5: Trang đăng nhập

### 3. Trang chat

- + Trang chat chính của ứng dụng với một tài khoản lần đầu tiên sử dụng ứng dụng. Tại trang này người dùng chọn tab home để kết bạn với người dùng khác trong hệ thống và bắt đầu trao đổi tin nhắn.



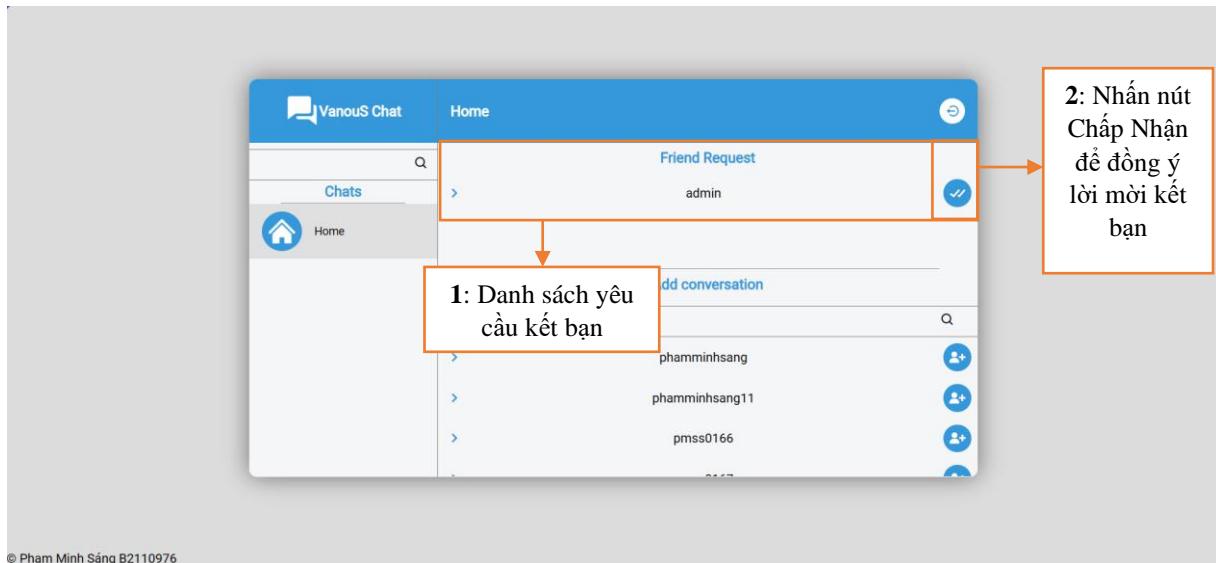
Hình 6: Trang chat - Home

- + Tại đây người dùng có thể sử dụng chức năng tìm kiếm người cụ thể cần kết bạn dễ dàng hơn.



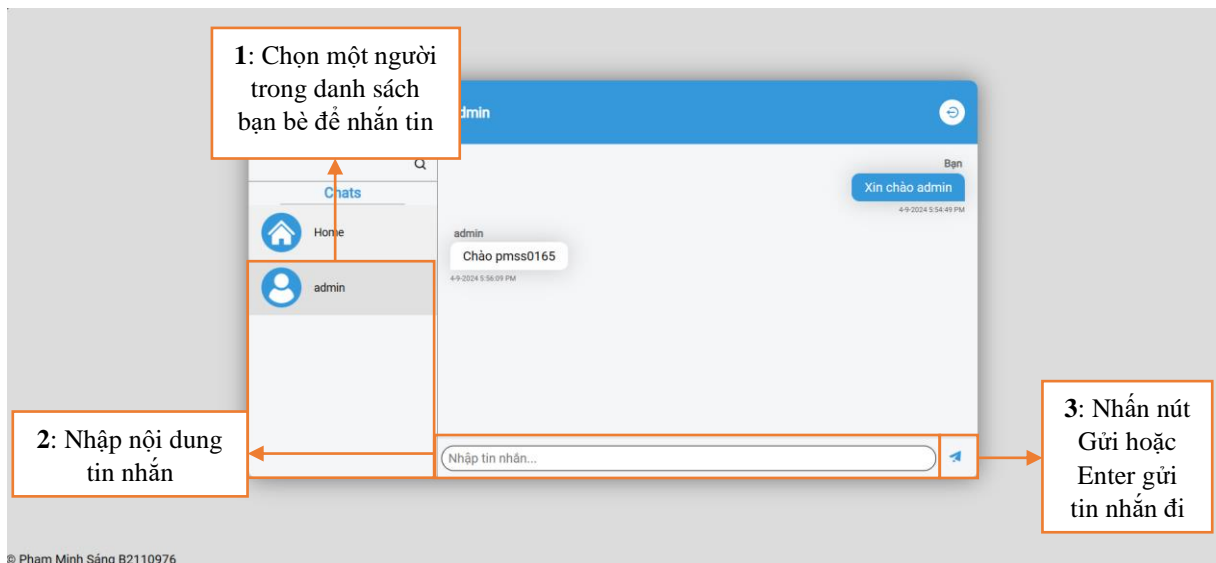
Hình 7: Trang chat - Tìm kiếm người dùng để kết bạn

- + Khi có một người dùng muốn kết bạn sẽ hiển thị lên màn hình ứng dụng lời mời kết bạn của người đó. Để đồng ý kết bạn nhấn vào nút tích kế bên lời mời và có thể bắt đầu trao tin nhắn với nhau.



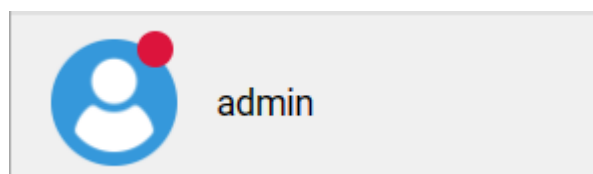
Hình 8: Trang chat - Lời mời kết bạn

- + Bên tab Chats chọn người muốn nhận tin nhắn từ bạn và bắt đầu trao đổi những thông tin mà bạn muốn gửi đến cho người nhận.



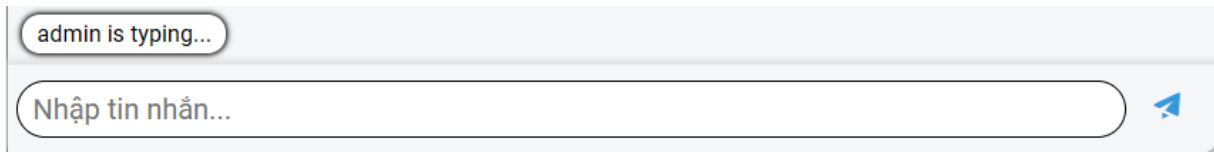
Hình 9: Trang chat - Gửi và nhận tin nhắn

- + Khi có một tin nhắn mới mà người dùng chưa đọc, sẽ có một chấm đỏ cạnh bên hộp thoại của người gửi tin nhắn đến để người dùng dễ dàng nhận biết.



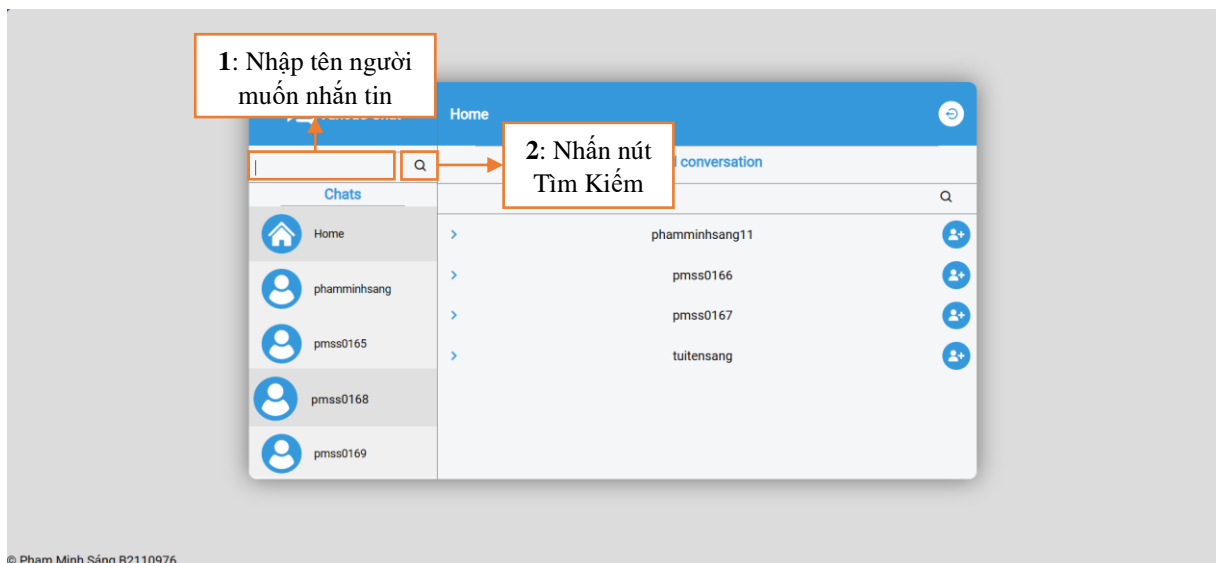
Hình 10: Trang chat - Hiển thị thông báo

- + Người dùng đang trong cuộc trò chuyện với một người dùng khác, khi người dùng đó đang nhập tin vào ô nhập tin, sẽ có một ô hiển thị là người dùng đó đang nhập tin nhắn.

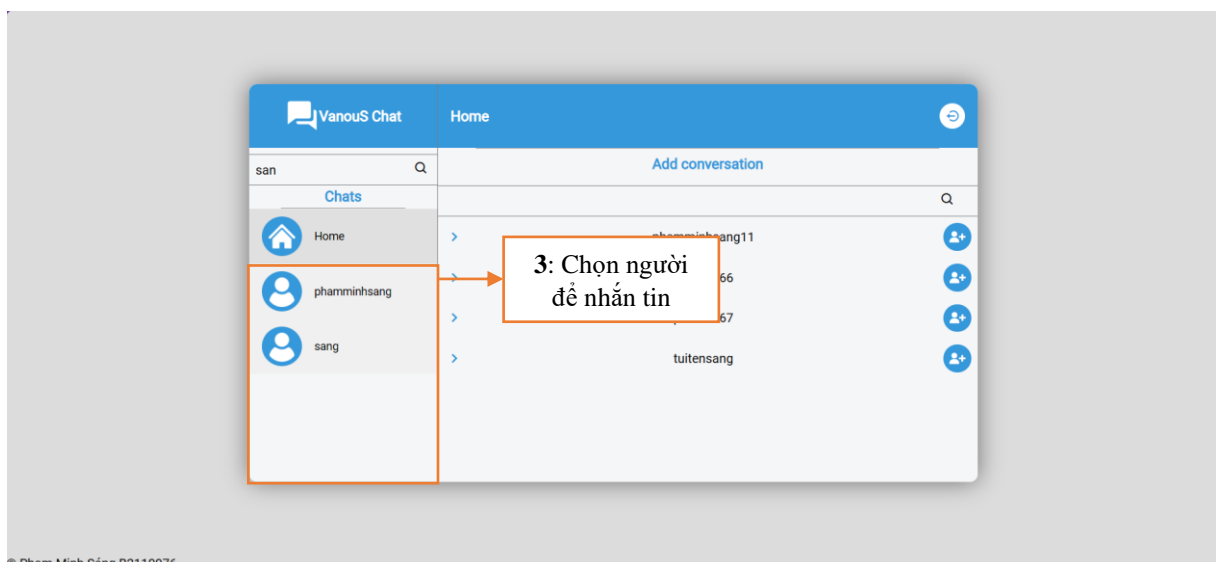


Hình 11: Trang chat - Hiển thị có người dùng đang nhập tin

- + Người dùng có thể sử dụng chức năng tìm kiếm người muốn gửi, trao đổi tin nhắn khi số lượng bạn bè quá nhiều.



Hình 12: Trang chat - Tìm kiếm người dùng muốn gửi tin



Hình 13: Trang chat - Hiển thị kết quả tìm kiếm người dùng

## CHƯƠNG 4: KẾT LUẬN – ĐÁNH GIÁ

### I. KẾT QUẢ ĐẠT ĐƯỢC

Sau khi xác định phạm vi nghiên cứu, áp dụng các kỹ thuật, kiến thức đã tìm hiểu, học hỏi được đã có thể xây dựng được một ứng dụng Chat với các chức năng cơ bản như:

- + Đăng ký được tài khoản, có thể sử dụng tài khoản đã đăng ký để đăng nhập ứng dụng.
- + Trao đổi tin nhắn giữa hai người dùng theo thời gian thực.
- + Kết bạn, đồng ý kết bạn giữa hai người dùng để bắt đầu trao đổi tin nhắn.
- + Thông báo khi có một tin nhắn mới mà người dùng chưa đọc.
- + Thiết kế được giao diện Chat cơ bản.

### II. THU HOẠCH

Qua quá trình thực hiện đề tài bản thân đã học hỏi, tiếp thu được nhiều công nghệ và kiến thức mới có thể giúp ích cho bản thân trong việc nghiên cứu, học tập sau này cụ thể như:

- + Hiểu được cách sử dụng Node.js để làm một webserver.
- + Sử dụng được thư viện Socket.IO để tạo trao đổi thông tin từ client này tới client khác theo thời gian thực .
- + Thiết kế được giao diện người dùng động bằng HTML, CSS, JavaScript kết hợp với cả thư viện JQuery.
- + Biết cách kết nối từ server tới hệ quản trị cơ sở dữ liệu MySQL để lưu thông tin người dùng, dữ liệu người dùng gửi đi.
- + Tự thiết kế được một mô hình cơ sở dữ liệu chứa dữ liệu.
- + Học hỏi được các cách giải quyết vấn đề trong việc trao đổi thông tin theo thời gian thực, cách truy xuất, sửa đổi các thông tin trong cơ sở dữ liệu.

### III. ƯU ĐIỂM

Ứng dụng đã xây dựng có được những ưu điểm như sau:

- + Trao đổi thông tin văn bản giữa hai người dùng nhanh chóng theo thời gian thực.
- + Có chức năng kết bạn qua lại người các người dùng sử dụng ứng dụng.
- + Hiện thị thông báo khi người dùng có tin nhắn từ người dùng khác chưa đọc.
- + Thanh tìm kiếm trả về kết quả danh sách cần tìm nhanh chóng cho người dùng.

### IV. HẠN CHẾ - NGUYÊN NHÂN

Tuy đã cố gắng tìm hiểu các giải pháp, cách giải quyết vấn đề khác nhau để tối ưu, hoàn thiện ứng dụng nhưng vẫn còn tồn tại một số hạn chế chưa thể khắc phục để giúp quá trình sử dụng của người dùng thoải mái, mượt mà hơn, chẳng hạn như:

- + Giao diện chưa mượt mà, chưa có các hiệu ứng tương tác phục vụ cho trải nghiệm người dùng tốt hơn. Do kiến thức về hiểu biết về thiết kế, xây dựng giao diện người dùng còn hạn chế.

- + Các hộp thoại chứa tin nhắn mới chưa được chèn lên trên đầu danh sách hộp thoại của người dùng trao đổi với người dùng khác. Do không đủ thời gian để thực hiện
- + Xây dựng cơ sở dữ liệu còn đơn sơ chưa đầy đủ hết tất cả các thông tin.

## **V. HƯỚNG PHÁT TRIỂN**

Ứng dụng Chat cơ bản vẫn còn có thể phát triển thêm để trở thành một ứng dụng có thể áp dụng vào đời sống thực tế, chẳng hạn như:

- + Xây dựng chức năng tạo nhóm cũng như là giao tiếp với những người dùng trong cùng nhóm đó.
- + Thêm chức năng video call để có thể giao tiếp trực quan hơn, thú vị hơn.
- + Gửi các tin nhắn có chứa sticker, icon vui nhộn tạo sự hứng thú, vui tươi, đa dạng cho cuộc trò chuyện.
- + Gửi hình ảnh cho người khác để cuộc trò chuyện thêm trực quan, rõ ràng thông tin của người dùng muốn truyền tải.
- + Phát triển ứng dụng trở thành ứng dụng đa nền tảng.

**TÀI LIỆU THAM KHẢO**

- [1] **Harris, Patricia.** *What is HTML Code?.* Pages: 4-6. The Rosen Publishing Group, Inc, 2017.
- [2] **Boduch, Adam, Chaffer, Jonathan, Swedberg, Karl.** *Learning JQuery 3.* Pages: 9-10. Packt Publishing Ltd, 2017.
- [3] **Satheesh, Mithun, Bruno Joseph D'mello, and Jason Krol.** *Web development with MongoDB and NodeJs.* Chapter 1. Packt Publishing Ltd, 2015.
- [4] **Rai, Rohit.** *Socket.IO real-time web application development.* Chapter 2, 3 and 5. Packt Publishing, 2013.
- [5] **Robbins, Jennifer Niederst.** *Learning web design: A beginner's guide to HTML, CSS, JavaScript, and web graphics.* Pages: 11-12. " O'Reilly Media, Inc.", 2012.
- [6] **Node.js - Run JavaScript Everywhere:** <https://nodejs.org/en>
- [7] **Socket.IO:** <https://socket.io/>
- [8] **StackOverflow:** <https://stackoverflow.com/>