

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



**NIÊN LUẬN CƠ SỞ**  
**XÂY DỰNG ỨNG DỤNG CHATBOX CƠ BẢN**



**Sinh viên thực hiện:**

**Nguyễn Trung Tín**

**MSSV : B2110979**

**Lớp : 21T9A2**

**Ngành : Mạng máy tính và Truyền thông dữ liệu**

**Khoá : 47**

**Giáo viên hướng dẫn:**

**Giảng viên: TS. Hà Duy An**

**HỌC KỲ 1, 2023-2024**

**Cần Thơ, 04/2024**

This image shows a full page of a handwriting practice worksheet. It consists of approximately 20 horizontal rows. Each row is defined by two parallel dotted lines, creating a series of uniform gaps for letter height. The entire page is otherwise blank, with no margins, text, or other markings.

## MỤC LỤC

Chương 1	GIỚI THIỆU .....	6
I.	TỔNG QUAN .....	6
1.	Mô tả .....	6
2.	Mục tiêu .....	6
3.	Hướng giải quyết .....	7
4.	Kế hoạch thực hiện .....	7
Chương 2	LÝ THUYẾT .....	8
I.	Khái niệm .....	8
1.	J2EE: .....	8
2.	HTML: .....	8
3.	CSS: .....	8
4.	JavaScript: .....	8
5.	MySQL: .....	9
6.	Java: .....	9
7.	Ajax: .....	9
8.	WebSocket: .....	10
9.	Mô hình MVC: .....	12
II.	Ứng dụng vào thực tiễn: .....	13
Chương 3	XÂY DỰNG ỨNG DỤNG .....	16
I.	PHÂN TÍCH YÊU CẦU .....	16
1.	Yêu cầu .....	16
2.	Phân tích: .....	16
3.	Đặc tả: .....	17

II.	THIẾT KẾ VÀ CÀI ĐẶT GIẢI PHÁP CHO CƠ SỞ DỮ LIỆU .....	19
1.	Sơ đồ tổng thể hệ thống.....	19
2.	Thiết kế thành phần dữ liệu .....	20
3.	Xây dựng các thực thể và mối quan hệ .....	23
III.	THIẾT KẾ CÀI ĐẶT GIAO DIỆN XỬ LÝ .....	25
1.	Login.....	25
2.	Register .....	27
3.	Giao diện trò chuyện chính.....	29
	CHƯƠNG 4 KẾT LUẬN VÀ ĐÁNH GIÁ.....	35
I.	KẾT LUẬN .....	35
1.	Kết quả.....	35
2.	Thu thập về mặt chuyên môn .....	35
3.	Kinh nghiệm .....	36
II.	ĐÁNH GIÁ .....	36
1.	Tổng quan .....	36
2.	Ưu điểm .....	36
3.	Nhược điểm .....	36
4.	Hướng phát triển.....	37

## MỤC LỤC HÌNH ẢNH

Hình 1 Mô hình hoạt động của Ajax.....	9
Hình 2 Cấu trúc WebSockets .....	10
Hình 3 Kết nối WebSocket so với kết nối HTTP.....	11
Hình 4 Mô hình MVC (Model - View - Controller).....	13
Hình 5 Cấu trúc dự án áp dụng mô hình MVC.....	14
Hình 6 Sơ đồ tổng quát chương trình .....	19
Hình 7 Sơ đồ hoạt vụ của tác nhân "Người dùng" .....	20
Hình 8 Mô hình thực thể - Mối quan hệ mở rộng (EER Diagram) .....	22
Hình 9 Giao diện đăng nhập (Login).....	26
Hình 10 Giao diện đăng ký (Register).....	27
Hình 11 Hộp thoại cảnh báo người dùng nhập không đồng nhất mật khẩu. ....	29
Hình 12 Giao diện trò chuyện chính.....	29
Hình 13 Nút đăng xuất.....	29
Hình 14 Giao diện trò chuyện với chức năng "Thêm bạn mới" .....	30
Hình 15 Nút đóng giao diện.....	30
Hình 16 Thanh tìm kiếm.....	30
Hình 17 Nút "Thêm bạn bè" theo từng người dùng đã thay đổi trạng thái .....	31
Hình 18 Giao diện trò chuyện với chức năng "Lời mời kết bạn" .....	31
Hình 19 Đồng ý kết bạn.....	31
Hình 20 Giao diện trò chuyện với hộp trò chuyện .....	32
Hình 21 Trạng thái hoạt động .....	32

## MỤC LỤC BẢNG

Bảng 1 Kế hoạch thực hiện.....	7
Bảng 2 Sự kiện của WebSocket.....	12
Bảng 3 Phương thức của WebSocket.....	12
Bảng 4 Thực thể USERS .....	24
Bảng 5 Thực thể Friends.....	24
Bảng 6 Thực thể BOXCHAT.....	25
Bảng 7 Thực thể MESSAGES .....	25

## **Chương 1**

### **GIỚI THIỆU**

#### **I. TỔNG QUAN**

##### **1. Mô tả**

Trong thời đại công nghệ 4.0 hiện nay, công nghệ thông tin đóng vai trò quan trọng, đặc biệt là trong lĩnh vực Internet đang trở thành một thành phần không thể thiếu đối với đời sống của mỗi chúng ta. Quá trình hình thành và phát triển của Internet cũng kéo theo sự phát triển của một số hình thức trao đổi thông tin. Vào thời xa xưa, việc trao đổi thông tin đã đóng một phần vai quan trọng trong cuộc sống xã hội. Từ thời kì trao đổi tin tức bằng thư từ thông qua những chú bồ câu tới thời kì bưu chính viễn thông, con người đóng vai trò quan trọng (người giao thư – giao liên) cung cấp các dịch vụ thư tín trong và ngoài nước. Khi chiếc điện thoại đầu tiên ra đời đã đẩy mạnh việc trao đổi thông tin. Trao đổi thông tin đóng vai trò quan trọng trong chính trị, giáo dục, kinh tế, quân sự, ... Sau cuộc các mạng công nghệ vấn đề trao đổi tin tức nở rộ thông qua nhiều hình thức như Video Call, Voice Call, Email, Social Network, ... Vai trò của trao đổi thông tin càng được làm rõ hơn qua Đại dịch Covid 2019, khi mọi người cách ly thì việc trao đổi thông tin là cần thiết như đồ ăn và thức uống. Đặc biệt là hình thức trao đổi bằng đoạn tin nhắn văn bản điển hình là Message, Zalo, Instagram, ... được mọi người ưa chuộng vì tính tiện lợi mà nó đem lại. Không như Email, việc trao đổi tin nhắn qua các ứng dụng như Message, Zalo gần như là tức thì (Realtime). Nhằm thực hiện nghiên cứu giải quyết vấn đề trên thì việc xây dựng “Ứng dụng Web Chat Realtime Đơn Giản” là thật sự cần thiết. Qua việc xây dựng giúp chúng ta hiểu hơn về cách xây dựng, vận hành một ứng dụng Chat Realtime.

##### **2. Mục tiêu**

Mục tiêu chính của đề tài “Xây dựng ứng dụng Web Chat Realtime Đơn Giản” là xây dựng một Website có thể cho hai người dùng chat trực tiếp với nhau thông qua giao diện một Website. Bên cạnh đó, ứng dụng sẽ quản lý thông tin tài khoản,

tên người dùng và mật khẩu nhằm đảm bảo sự an toàn bảo mật khi hai người dùng tương tác với nhau. Đơn giản hoá quá trình đăng ký, đăng nhập giúp người dùng dễ dàng tiếp cận. Giao diện thân thiện với người dùng. Đảm bảo các tính năng cơ bản của một ứng dụng chat hoạt động bình thường, nhanh chóng và hiệu quả.

### 3. Hướng giải quyết

- **Đối tượng nghiên cứu:** Ứng dụng Chat.
- **Phạm vi:** Giữa 2 người dùng chat với nhau.
- **Công nghệ:** J2EE, kỹ thuật Web (HTML, CSS, JavaScript, Ajax), Java core (JDK 17), WebSocket.
- **Mô hình xây dựng Server – Client :** Mô hình MVC (Model – View - Controller)
- **Thiết bị chạy demo:** Laptop Acer Aspire 7, CPU AMD RYZEN 5 5500U, Ram 16gb.
- **Các phần mềm sử dụng trong quá trình xây dựng:** Apache NetBeans IDE 20, MySQL Workbench 8.0 CE, Web Browser (Chrome, Edge Dev, Edge) phiên bản mới nhất.

### 4. Kế hoạch thực hiện

Bảng 1 Kế hoạch thực hiện

Tuần	Công việc	Ghi chú
1 - 4	Tìm hiểu công nghệ, phạm vi thực hiện	
5 - 6	Triển khai các lớp cơ bản. Xây dựng Database.	
7 - 8	Xây dựng giao diện ứng dụng. Xây dựng xử lý đăng nhập, đăng xuất	
9 - 11	Xây dựng lõi ứng dụng: Kết bạn (Gửi lời mời, đồng ý kết bạn). Chat (Gửi tin nhắn, hiển thị tin nhắn, thông báo tin nhắn mới).	
12 - 13	Sửa lỗi.	
13 - 14	Viết báo cáo word.	
15	Thiết kế slide báo cáo.	
16	Hoàn thiện demo chương trình	
17	Chuẩn bị các khâu báo cáo.	
18	Hoàn thành báo cáo.	



## Chương 2

### LÝ THUYẾT

#### I. Khái niệm

##### 1. J2EE:

**Java Platform Enterprise Edition** (hoặc Java EE), hiện này là **Jakarta EE**. J2EE còn được biết tới với tên “**Phát triển ứng dụng doanh nghiệp**”, mô hình ứng dụng J2EE dựa trên nền tảng ngôn ngữ lập trình Java và máy ảo Java. J2EE cung cấp kho APIs của công ty Oracle phục vụ quá trình xây dựng ứng dụng web trở nên dễ dàng hơn. Kho lưu trữ APIs có thể tái cấu trúc sử dụng nhiều lần. Bên cạnh đó, nó còn là một công nghệ hỗ trợ xây dựng hệ thống phân tán. [Theo **Jakarta.ee**]

##### 2. HTML:

HTML là viết tắt của **Hyper Text Markup Language** (Ngôn ngữ đánh dấu siêu văn bản), tiêu chuẩn để tạo ra các trang Web. Bên cạnh đó, HTML còn là mô tả cấu trúc của một trang Web, các thẻ (phần tử) HTML cho trình duyệt biết cách hiển thị nội dung, gắn nhãn cho các nội dung. [Theo **W3S HTML**]

##### 3. CSS:

Viết tắt của **Cascading Style Sheets**. CSS giúp thiết kế, kiểm soát bố cục dễ dàng, giúp hiển thị các phần tử HTML một cách sinh động hơn. Có thể nói HTML đóng vai trò là khung xương thì CSS là phần biểu bì bên ngoài cơ thể. [Theo **W3S CSS**]

##### 4. JavaScript:

JavaScript là một ngôn ngữ lập trình (Web) được phát minh bởi Brendan Eich vào năm 1995 và trở thành tiêu chuẩn **ECMA** vào năm 1997. ECMA-262 là tên chính thức của tiêu chuẩn. ECMAScript là tên chính thức của ngôn ngữ. JavaScript đóng vai trò là tuyến thân kinh của một trang Web. [Theo **W3S JavaScript**]

## 5. MySQL:

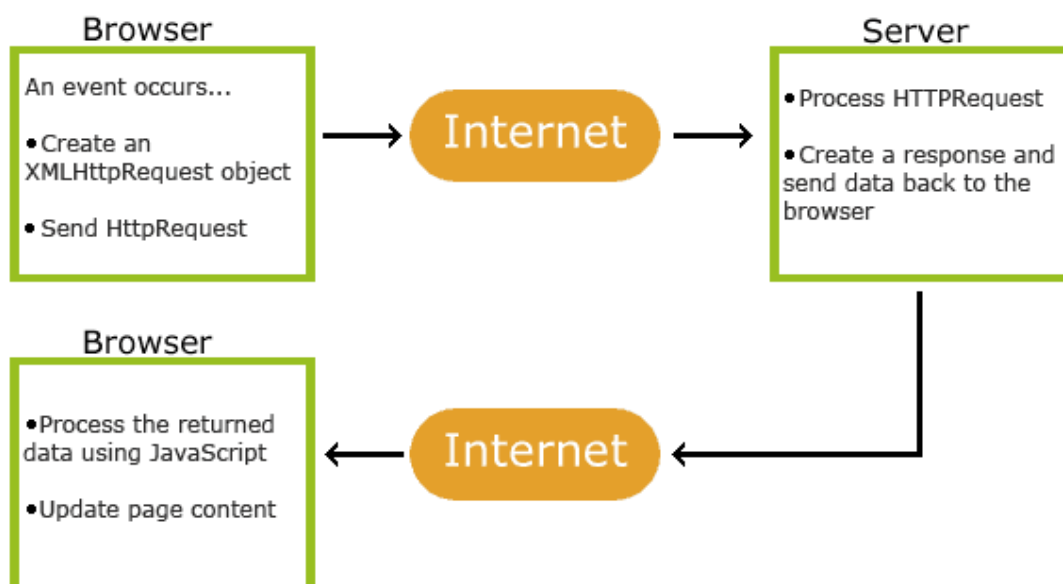
MySQL là một hệ thống quản lý cơ sở dữ liệu quan hệ, mã nguồn mở, miễn phí, rất thích hợp cho các ứng dụng nhỏ và lớn. Hỗ trợ đa nền tảng, tuân thủ các tiêu chuẩn của **ANSI SQL**. MySQL được phát hành lần đầu tiên vào năm 1995 phân phối và hỗ trợ bởi Tập đoàn Oracle. [Theo **W3S MySQL**]

## 6. Java:

Java là một ngôn ngữ lập trình hướng đối tượng. Được sử dụng để phát triển phần mềm, trang web, game hay ứng dụng trên các thiết bị di động, cũng bởi Java có tính chất đa nền nền có thể chạy hầu hết tất cả nền tảng khác nhau. Java được phát hành năm 1994, đến năm 2010 được Oracle mua lại từ **Sun MicroSystem**.

## 7. Ajax:

Ajax là viết tắt của **Asynchronous JavaScript And XML**. Không phải là một ngôn ngữ lập trình. Ajax có thể đọc dữ liệu từ máy chủ web – sau khi trang được tải. Cập nhật một trang web mà không cần tải lại trang. Có thể gửi dữ liệu tới máy chủ thông qua chế độ nền.



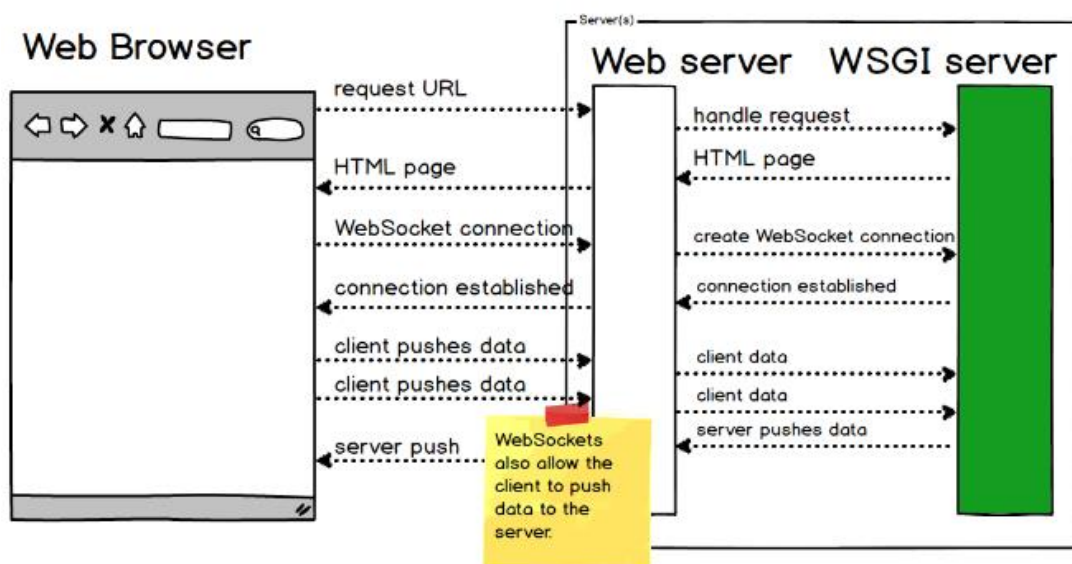
Hình 1 Mô hình hoạt động của Ajax.

## 8. WebSocket:

### – Định nghĩa:

Công nghệ hỗ trợ giao tiếp hai chiều giữa Client và Server bằng cách sử dụng một TCP Socket để tạo kết nối. WebSocket là một kiểu Reverse Ajax hỗ trợ giao tiếp Realtime giữa Client và Server. Không giống như HTTP truyền thông tuân theo mô hình phản hồi yêu cầu, giao thức này cho phép giao tiếp hai chiều. Điều này có nghĩa là Client và Server có thể tương tác dữ liệu cho nhau bất cứ lúc nào, giúp dữ liệu truyền đi nhanh chóng mà không cần phải reload trang web.

## WebSockets



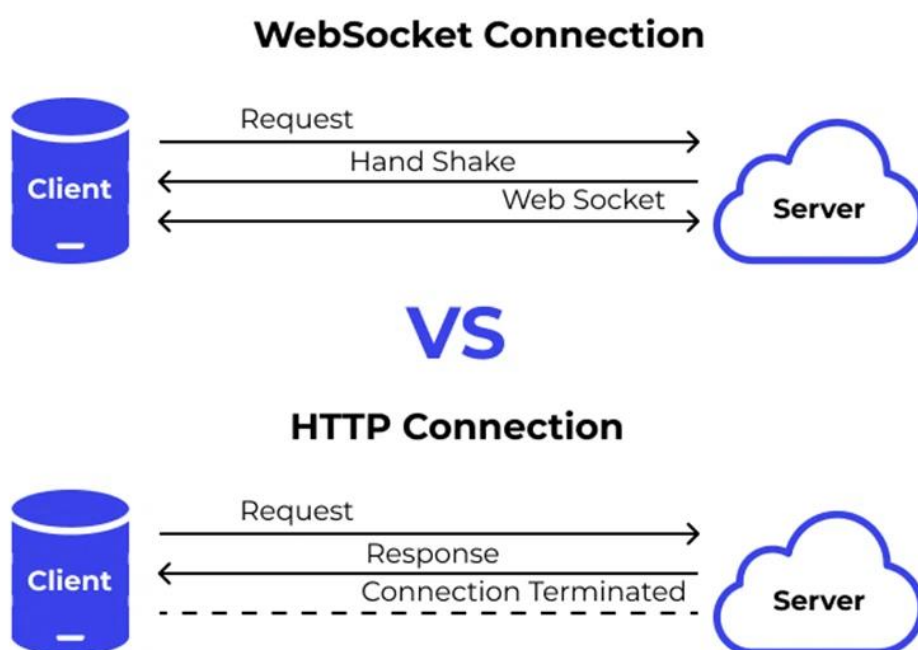
Hình 2 Cấu trúc WebSockets

Giao thức chuẩn thông thường của WebSocket là **ws://**, giao thức secure **wss://**. Chuẩn giao tiếp là String và hỗ trợ buffered arrays và blobs. (Vì niên luận này hướng tới nghiên cứu học tập ở mức đơn giản vậy nên **ws://** được lựa chọn làm nền tảng phát triển thay vì sử dụng giao thức secure).

### – Cách thức hoạt động:

Về bản chất, WebSocket có tính chất hai chiều và sử dụng giao thức TCP. Do đó, quá trình hoạt động của giao thức này vẫn cần bắt đầu bằng một “bắt tay” (handshake), sau đó là duy trì một kết nối liên tục, cho phép truyền dữ liệu một cách

nhANH chóng và hiệu quả. Toàn bộ quá trình hoạt động sẽ diễn ra như sau: Web Browser gửi một yêu cầu HTTP đặc biệt tới máy chủ. Yêu cầu này bao gồm một tiêu đề (Upgrade header) HTTP/1.1 thông báo máy chủ biết rằng trình duyệt muốn “nâng cấp” kết nối từ HTTP sang WebSocket. Nếu máy chủ có hỗ trợ giao thức WebSocket, nó sẽ phản hồi với một tiêu đề HTTP để xác nhận chuyển đổi. Khi tiêu đề được trình duyệt tiếp nhận, kết nối WebSocket sẽ chính thức được thiết lập. Sau khi kết nối được thiết lập, máy chủ và trình duyệt đã có thể gửi dữ liệu nhiều lần mà không cần yêu cầu hoặc phản hồi. Điều này giúp giảm thiểu độ trễ và tăng tốc truyền dữ liệu.



Hình 3 Kết nối WebSocket so với kết nối HTTP

– Các sự kiện của WebSocket

*Bảng 2 Sự kiện của WebSocket*

Sự kiện	Event Handler	Mô tả
open	onopen	Khi một WebSocket chuyển sang trạng thái mở, “onopen” sẽ được gọi.
message	onmessage	Khi WebSocket nhận dữ liệu từ Server.
error	onerror	Có bất kì lỗi nào trong giao tiếp.
close	onclose	Kết nối được đóng. Những sự kiện được truyền cho “onclose” có ba tham số là “code”, “reason”, và “wasClean”.

– Các phương thức của WebSocket

*Bảng 3 Phương thức của WebSocket*

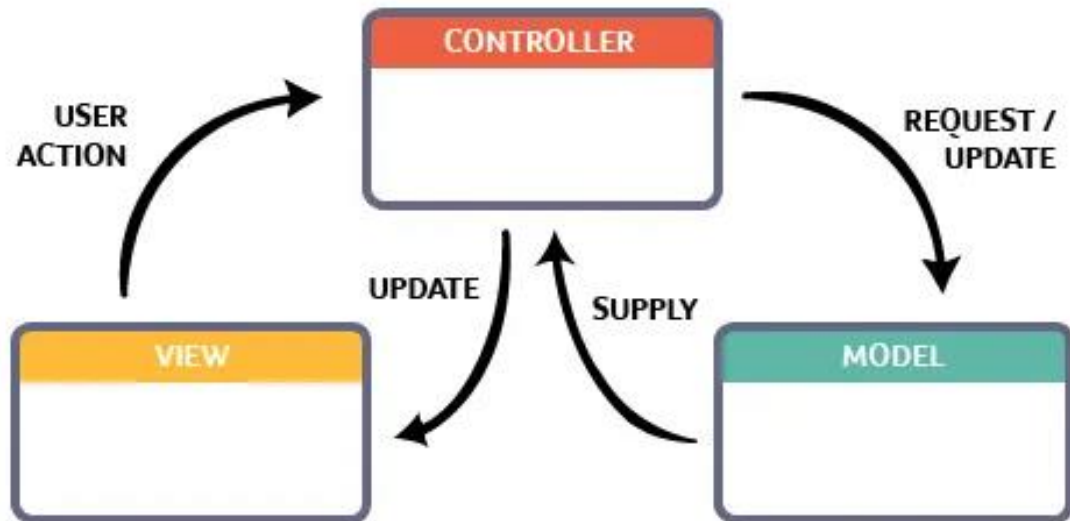
Phương thức	Mô tả
Send()	Send(data) gửi dữ liệu tới Server. Message data là String, ArrayBuffer, blob.
Close()	Đóng kết nối đang tồn tại.

## 9. Mô hình MVC:

MVC là viết tắt của cụm từ “Model - View - Controller”. Đây là mô hình thiết kế được sử dụng trong kỹ thuật phần mềm. MVC là một mẫu kiến trúc phần mềm để tạo lập giao diện người dùng trên máy tính. MVC chia làm ba phần được kết nối với nhau và mỗi thành phần đều có một nhiệm vụ riêng của nó và độc lập với các thành phần khác.

- **Model (dữ liệu):** Quản lý xử lý các dữ liệu.
- **View (giao diện):** Nơi hiện thị dữ liệu cho người dùng.
- **Controller (bộ điều khiển):** Điều khiển sự tương tác của hai thành phần Model và View.

[Trích của **Cao Lê Viết Tiến**]

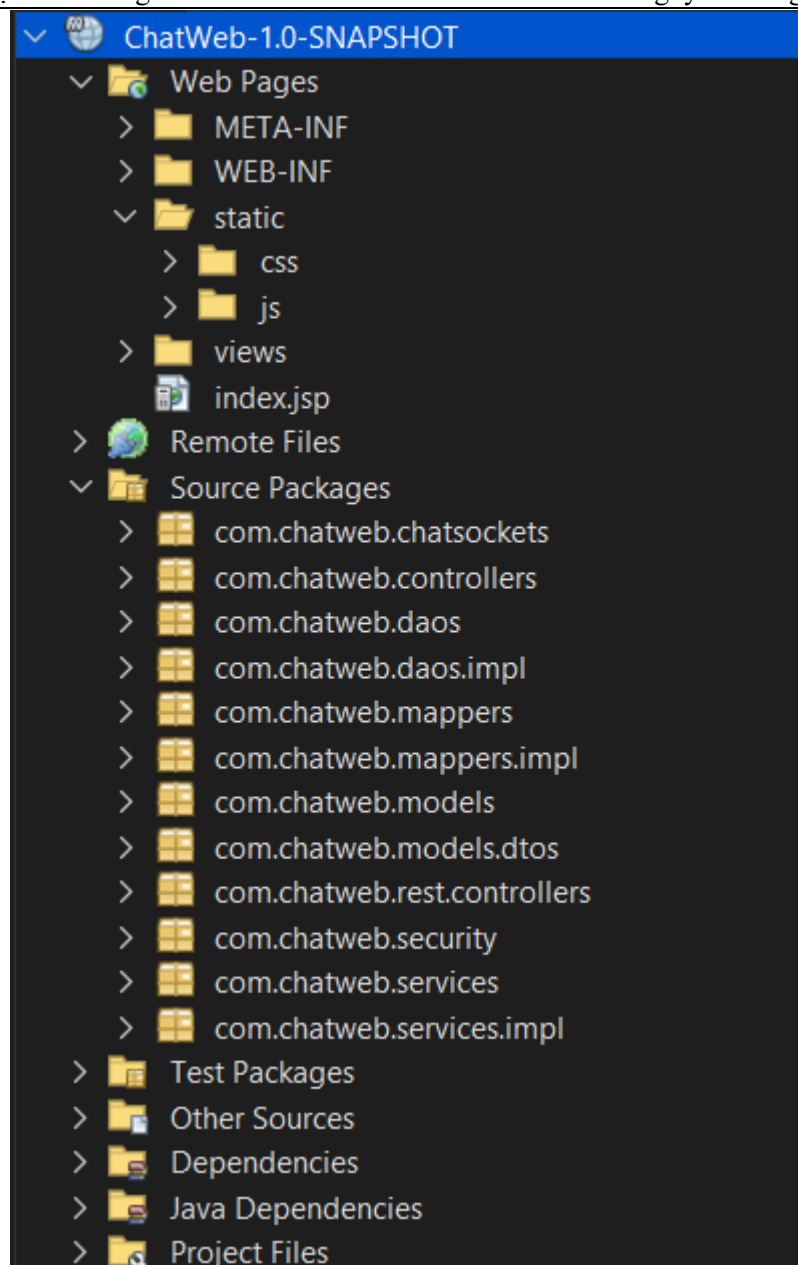


Hình 4 Mô hình MVC (Model - View - Controller)

## II. Ứng dụng vào thực tiễn:

Theo mô hình MVC xây dựng ứng dụng theo hướng đa tầng. Với thư mục Views quản lý tất cả giao diện của người dùng gồm có giao diện: Đăng nhập, Đăng ký, Chat Box. Thư mục static quản lý 2 thư mục con là css và js. Đây là nơi lưu trữ các style của từng trang giao diện cũng như những xử lý phần client thông qua javascript.

Xây dựng Model gồm có các packages: model (quản lý các lớp đối tượng), daos (chứa các interface kết nối với cơ sở dữ liệu), daos.impl (nơi triển khai các interface trong daos), model.dtos (chuyển đổi các model class thành json class nhằm dễ dàng xử lý dữ liệu đầu ra ở client khi nhận dữ liệu từ client hay từ server phản hồi client), các controllers và rest.controller (giúp điều khiển các trang, và nơi tiếp nhận yêu cầu người dùng), servers và services.impl (đóng vai trò là cầu nối giữa daos và controller).



Hình 5 Cấu trúc dự án áp dụng mô hình MVC

HTML, CSS, JavaScript được vận dụng vào xây dựng giao diện người dùng kết hợp với JSP của J2EE. Giao diện được xây dựng dựa trên nền tảng HTML, cấu trúc bởi CSS, sử dụng các function JavaScript để thực hiện các thao tác thay đổi trực tiếp trên giao diện, cung cấp các xử lý thô ở phía Client giúp giảm tải phía Server. Kết hợp JSP giúp quản lý hiển thị trở nên dễ dàng hơn với các taglib, function java được JSP biên dịch thành phần tử của HTML.

Sử dụng những kiến thức đã có được của MySQL để triển khai cơ sở dữ liệu đơn giản hợp lý về mặt logic nhằm quản lý các dữ liệu phát sinh của ứng dụng cần

lưu trữ lại để sử dụng nhiều lần như dữ liệu người dùng, dữ liệu tin nhắn, dữ liệu đoạn chat, dữ liệu bạn bè.

Ajax được dùng như công cụ kết nối với Server để lấy các thông tin về bạn bè, người dùng, tin nhắn để thực hiện cập nhật liên tục làm cho giao diện động có tính tương tác cao hơn, thân thiện hơn với người dùng.

WebSocket đóng vai trò cốt lõi trong ứng dụng việc áp dụng các kiến thức của WebSocket giúp xây dựng ứng dụng dễ dàng. Sử dụng các phương thức mà WebSocket cung cấp góp phần to lớn trong việc giảm thiểu thời gian xây dựng. Ở phía Server sử dụng thư viện API có sẵn do Jakarta cung cấp để xây dựng WebSocket, để đón nhận yêu cầu “bắt tay” từ phía Client. Thông qua các function event: onopen, onerror, onmessage và onclose. Điểm nổi bật là thông qua Servlet EndPoint và thông qua đoạn mã hoá và giải mã đoạn tin nhắn của Client. Sử dụng các lớp DTO đã được định nghĩa sẵn để phục vụ mã hoá.

Vì xây dựng dựa trên công nghệ J2EE lấy ngôn ngữ lập trình Java làm cốt lõi nên trong quá trình xây dựng kiến thức về Java core đóng vai trò quan trọng. Kiến thức về lập trình hướng đối tượng, xử lý kiểu dữ liệu mảng, các luồng dữ liệu.



## **Chương 3**

### **XÂY DỰNG ỨNG DỤNG**

#### **I. PHÂN TÍCH YÊU CẦU**

##### **1. Yêu cầu**

Dựa theo nhu cầu sử dụng của người dùng, ứng dụng Web xây dựng phải đảm bảo giao diện thân thiện, tối giản. Có trang đăng nhập, đăng ký và giao diện trò chuyện. Có danh sách bạn bè, danh sách bạn bè chưa kết bạn, tìm kiếm và kết bạn. Có cửa sổ trò chuyện. Hiển thị người dùng online và hiển thị thông báo khi có tin nhắn mới. Cuộc trò chuyện thời gian thực.

##### **2. Phân tích:**

Theo yêu cầu của người dùng, cần ít nhất 3 giao diện bao gồm: Đăng ký, Đăng Nhập, Giao diện trò chuyện.

Các chức năng cơ bản cần xây dựng gồm: chức năng đăng nhập, đăng ký, chức năng hiển thị danh sách bạn bè (đã kết bạn, chưa kết bạn), tìm kiếm bạn bè và kết bạn. Đặc biệt, là trò chuyện thời gian thực bằng hình thức văn bản (text). Các chức năng cần thiết khác làm cho ứng dụng trở nên thân thiện hơn gồm: hiển thị bạn bè online, cập nhật thông báo tin nhắn mới khi có người dùng gửi tin nhắn.

Dựa vào yêu cầu, cần xây dựng các bảng dữ liệu gồm: Users, Friends, Boxchat, Message. Để quản lý các dữ liệu phát sinh của ứng dụng.

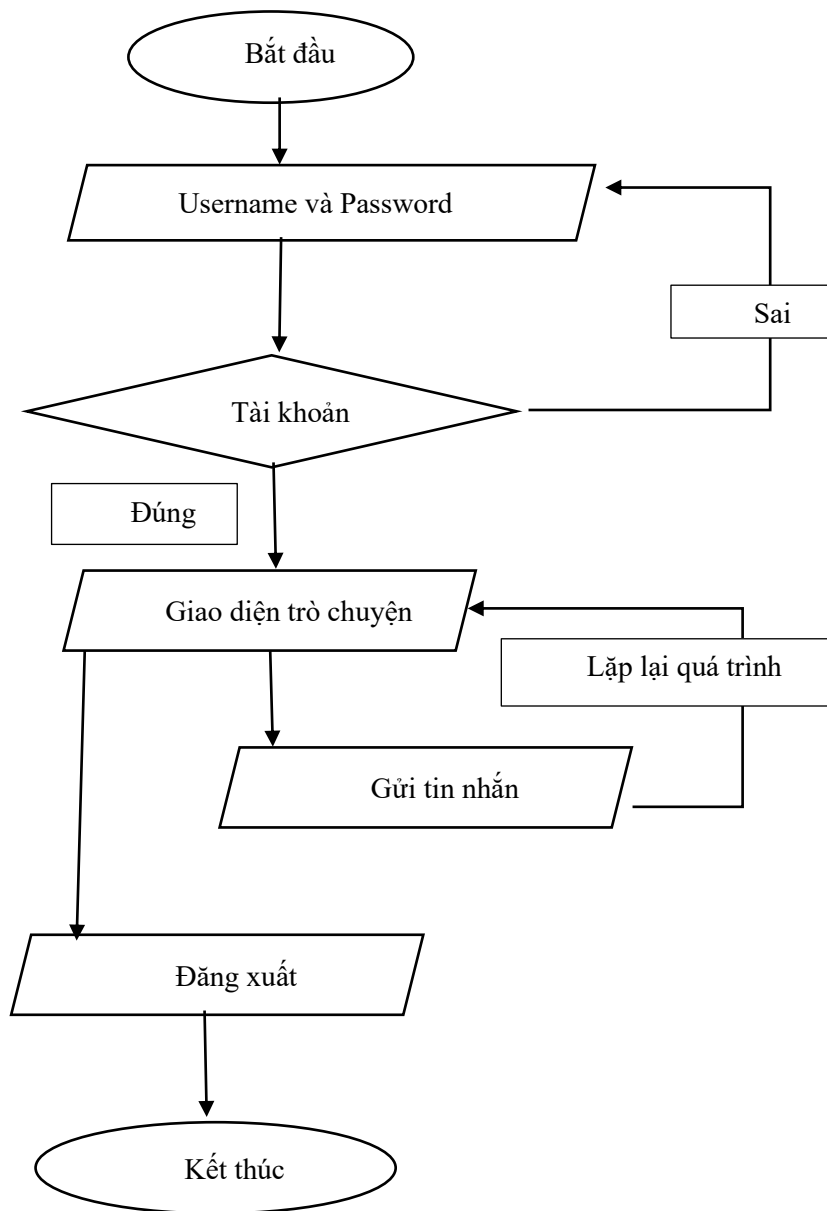
### 3. Đặc tả:

- Giao diện:
  - Trang Đăng nhập:
    - Username
    - Password
  - Trang Đăng ký:
    - Username
    - Name
    - Password 1
    - Password 2
    - Avatar
  - Trang Trò chuyện chính:
    - Danh sách online
    - Danh sách người dùng chưa đồng ý kết bạn
    - Tìm và thêm bạn mới
    - Cửa sổ trò chuyện
      - Tên người trò chuyện
      - Nội dung tin nhắn cũ
      - Nhập tin nhắn mới

- Cơ sở dữ liệu:
  - Users
    - Username
    - Password
    - Name
    - Avatar
    - isOnline
  - Friends
    - User1
    - User2
    - Status
  - BoxChat
    - Id
    - Name
  - Messages
    - Id
    - Sender
    - Receiver
    - Content
    - Created\_at
    - Box\_id
    - Status

## II. THIẾT KẾ VÀ CÀI ĐẶT GIẢI PHÁP CHO CƠ SỞ DỮ LIỆU

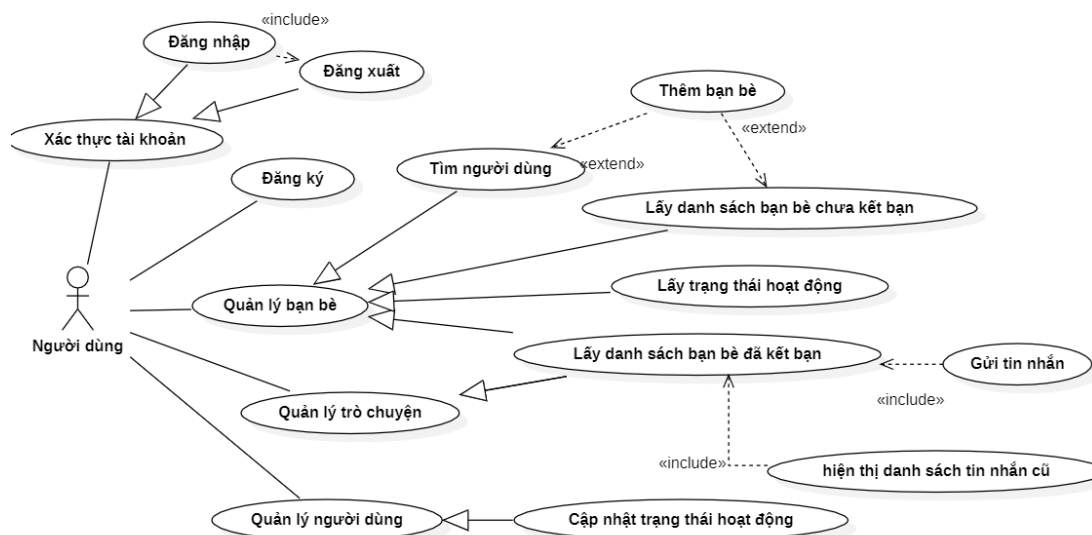
### 1. Sơ đồ tổng thể hệ thống



Hình 6 Sơ đồ tổng quát chương trình

## 2. Thiết kế thành phần dữ liệu

### – Sơ đồ hoạt vụ (Use case diagram)

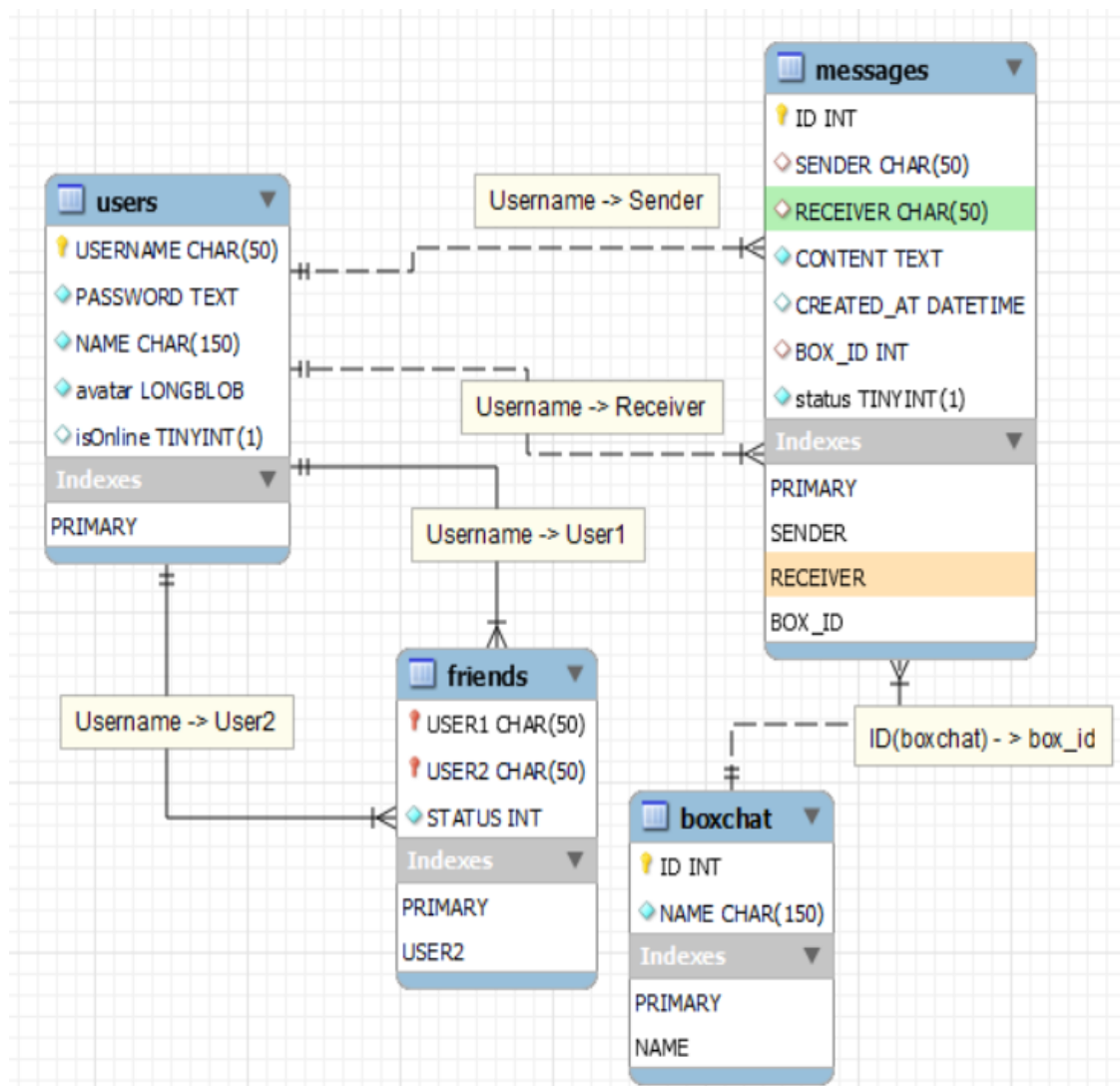


Hình 7 Sơ đồ hoạt vụ của tác nhân "Người dùng"

- Xác thực tài khoản: người dùng sẽ “Đăng nhập” vào hệ thống, khi đăng nhập thành công sẽ xuất hiện chức năng “Đăng xuất”.
- Đăng ký: giúp người dùng tạo tài khoản người dùng mới.
- Quản lý bạn bè:
  - Tìm người dùng: dựa vào tên người dùng hoặc username để tìm người dùng tồn tại trong hệ thống mà chưa kết bạn.
  - Lấy danh sách bạn bè chưa kết bạn: là danh sách người dùng đã gửi lời mời tới người dùng trong phiên đăng nhập nhưng chưa nhận phản hồi.
  - Lấy trạng thái hoạt động: thực hiện một các bị động khi người dùng hiển thị danh sách người dùng thì ứng dụng sẽ hiển thị thông tin hoạt động của người dùng đó.
  - Lấy danh sách bạn bè đã kết bạn: là danh sách những người dùng đã kết nối (là bạn bè).
  - Thêm bạn bè: gửi lời mời kết bạn hoặc đồng ý lời mời kết bạn của người dùng khác muốn kết nối với người dùng trong phiên làm việc.

- Quản lý trò chuyện:
  - Lấy danh sách bạn bè đã kết bạn: chức năng chính nằm trong quản lý người dùng nhưng đóng vai trò quan trọng trong quản lý trò chuyện. Vì chỉ thực hiện trò chuyện với người dùng đã kết bạn, bắt buộc người dùng trò chuyện nằm trong danh sách này.
  - Gửi tin nhắn: người dùng nhập tin nhắn bằng văn bản và bấm gửi tin nhắn sẽ hiện thị trên đoạn chat.
  - Hiện thị danh sách tin nhắn cũ: tin nhắn đã xem hoặc chưa xem nhưng không nằm trong cuộc trò chuyện trực tiếp sẽ được chức năng này hiện thị trên đoạn chat.
- Quản lý người dùng: khi người dùng đăng nhập thành công chức năng này sẽ tự động thực hiện cập nhật trạng thái người dùng thành online.

– Mô hình thực thể - Mối quan hệ mở rộng (EER Diagram)



Hình 8 Mô hình thực thể - Mối quan hệ mở rộng (EER Diagram)

▪ Bảng USERS:

- “USERNAME” là khoá chính không cho phép rỗng.
- “PASSWORD” lưu trữ mật khẩu đã hash của người dùng và không cho phép rỗng.
- “NAME” lưu trữ tên của người dùng và không cho phép rỗng.
- “Avatar” lưu trữ hình ảnh đại diện của người dùng dưới dạng longblob và không cho phép rỗng.
- “isOnline” lưu trạng thái trực tuyến của người dùng ở dưới dạng boolean.

- Bảng BOXCHAT: bảng này lưu trữ thông tin về các hộp trò chuyện
  - “ID” là khoá chính, tự động tăng và không được phép trống.
  - “NAME” lưu trữ tên của hộp trò chuyện và không được phép trống. cũng như là duy nhất. (tên được lưu bằng tên sender+receiver or receiver+sender và chỉ tồn tại duy nhất)
- Bảng MESSAGES: bảng này lưu trữ thông tin về các tin nhắn giữa các người dùng.
  - “ID” là khoá chính, tự động tăng và không cho phép trống.
  - “SENDER” tham chiếu đến cột “USERNAME” trong bảng USERS để xác định người gửi tin nhắn.
  - “RECEIVER” tham chiếu đến cột “USERNAME” trong bảng USERS để xác định người nhận tin nhắn.
  - “CONTENT” lưu trữ nội dung của tin nhắn và không cho phép trống.
  - “CREATED\_AT” lưu thời gian tạo của tin nhắn, mặc định là thời điểm hiện tại khi được thêm vào.
  - “BOX\_ID” tham chiếu đến cột “ID” trong bảng BOXCHAT để xác định hộp trò chuyện mà tin nhắn thuộc về.
  - “STATUS” lưu trữ trạng thái của tin nhắn (Ví dụ: Đã gửi, Đã nhận) dưới dạng boolean.

### 3. Xây dựng các thực thể và mối quan hệ

- Thực thể USERS là nơi lưu thông tin tài khoản của người dùng. Mỗi người dùng có một “Username”, “Password”, “Name” và “Avatar” riêng. Có lưu trữ trạng thái hoạt động của người dùng thông qua “isOnline”.



Bảng 4 Thực thể USERS

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải
1	USERNAME	Varchar(50)	Tên đăng nhập của người dùng.
2	PASSWORD	Text	Mật khẩu của người dùng.
3	NAME	Nvarchar(150)	Tên của người dùng.
4	AVATAR	LongBlob	Ảnh đại diện của người dùng.
5	ISONLINE	Boolean	Trạng thái online của người dùng.

- Thực thể FRIENDS là nơi lưu hoạt động kết nối giữa 2 người dùng khác nhau liên kết với nhau. Có “User1” và “User2” ràng buộc bởi “Username” trong bảng thực thể USERS. “Status” cung cấp trạng thái mối quan hệ giữa 2 người dùng. Nếu người dùng “User1” gửi lời mời kết bạn thì trạng thái mối quan hệ là 1, nhưng khi “User2” đồng ý kết bạn trạng thái sẽ chuyển về 2.

Bảng 5 Thực thể Friends

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải
1	USER1	Varchar(50)	Người dùng thứ nhất.
2	USER2	Varchar(50)	Người dùng thứ hai.
3	STATUS	int	Trạng thái của mối quan hệ.

- Thực thể BOXCHAT là nơi lưu trữ hộp các cuộc trò chuyện. Đóng vai trò quản lý các phần tử tin nhắn giữa hai người tương tác trực tiếp với nhau. Thực thể gồm 2 thuộc tính là “ID” và “Name”. Thuộc tính “Name” được cấu tạo bởi “Username” của hai người đó kết hợp lại với nhau. Mang tính duy nhất, không trùng lặp.

Bảng 6 Thực thể BOXCHAT

STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải
1	ID	int	Mã BoxChat tự động cập nhật luôn được thay mới
2	NAME	NVarchar(150)	Tên là sự kết hợp username của 2 người dùng và thiết hại quá lớn.

- Thực thể MESSAGES là nơi lưu trữ nội dung tin nhắn giữa các người dùng, được quản lý thông qua thực thể BOXCHAT bởi “BOX\_ID” tham chiếu tới “ID”. Người nhận “RECEIVER” và người gửi “SENDER” tham chiếu tới “USERNAME” của thực thể USERS. “CREATED\_AT” lưu thời gian tin nhắn gửi tới, qua đó giúp quản lý tin nhắn một cách hợp lý. “STATUS” thể hiện trạng thái tin nhắn, người nhận đã xem sẽ có trạng thái “2” còn người dùng chưa xem sẽ có trạng thái “1”.

Bảng 7 Thực thể MESSAGES

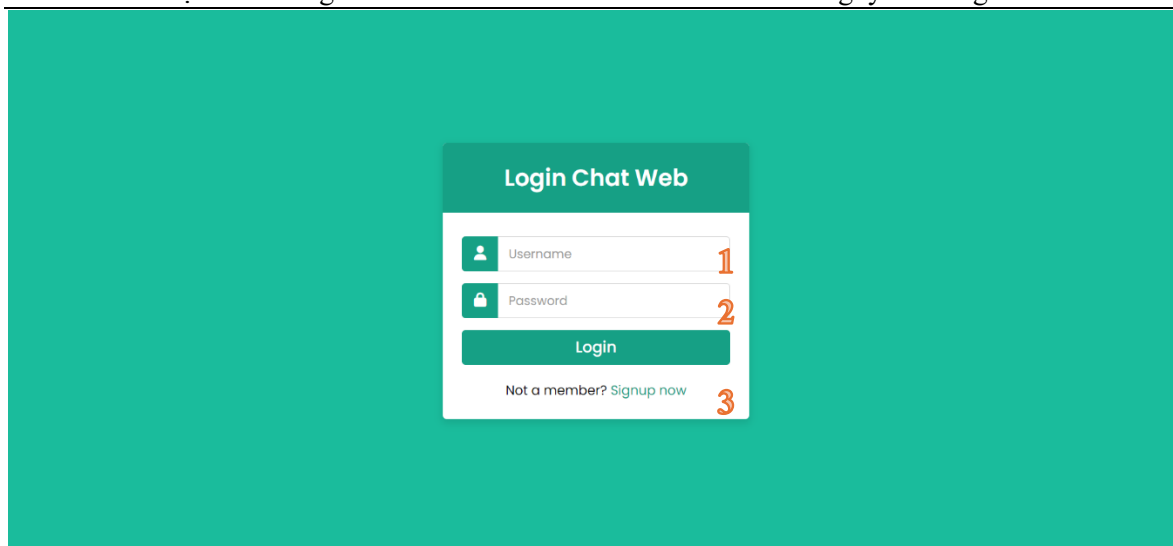
STT	Tên thuộc tính	Kiểu dữ liệu	Diễn giải
1	ID	Int	Tự động tăng khi thêm một tin nhắn mới. Khoá chính của thực thể.
2	SENDER	Varchar(50)	Người gửi tin nhắn.
3	RECEIVER	Varchar(50)	Người nhận tin nhắn.
4	CONTENT	Text	Nội dung tin nhắn.
5	CREATED_AT	DateTime	Thời gian tin nhắn được gửi.
6	BOX_ID	Int	Mã của đoạn Chat.
7	STATUS	boolean	Trạng thái của tin nhắn.

### III. THIẾT KẾ CÀI ĐẶT GIAO DIỆN XỬ LÝ

#### 1. Login

##### 1. Giao diện

Ở giao diện này người dùng sẽ điền các thông tin “Username” và “Password”. Sau khi nhập và nhấn Login, thông tin sẽ được xử lý, sau lý xử lý thành công và tài khoản hợp lệ người dùng sẽ được đưa đến giao diện trò chuyện.



Hình 9 Giao diện đăng nhập (Login)

- (1) “Username” người dùng sẽ nhập tối đa 50 kí tự, không dấu.
- (2) “Password” người dùng nhập tối thiểu 8 kí tự, và tối đa 50 kí tự.
- (3) Đường dẫn người dùng qua giao diện đăng ký tài khoản (Register).

## 2. Xử lý

Khi người dùng truy cập vào URL trang web sẽ gửi một doGet từ Servlet đến Server yêu cầu lấy giao diện Login hiển thị ra cho người dùng.

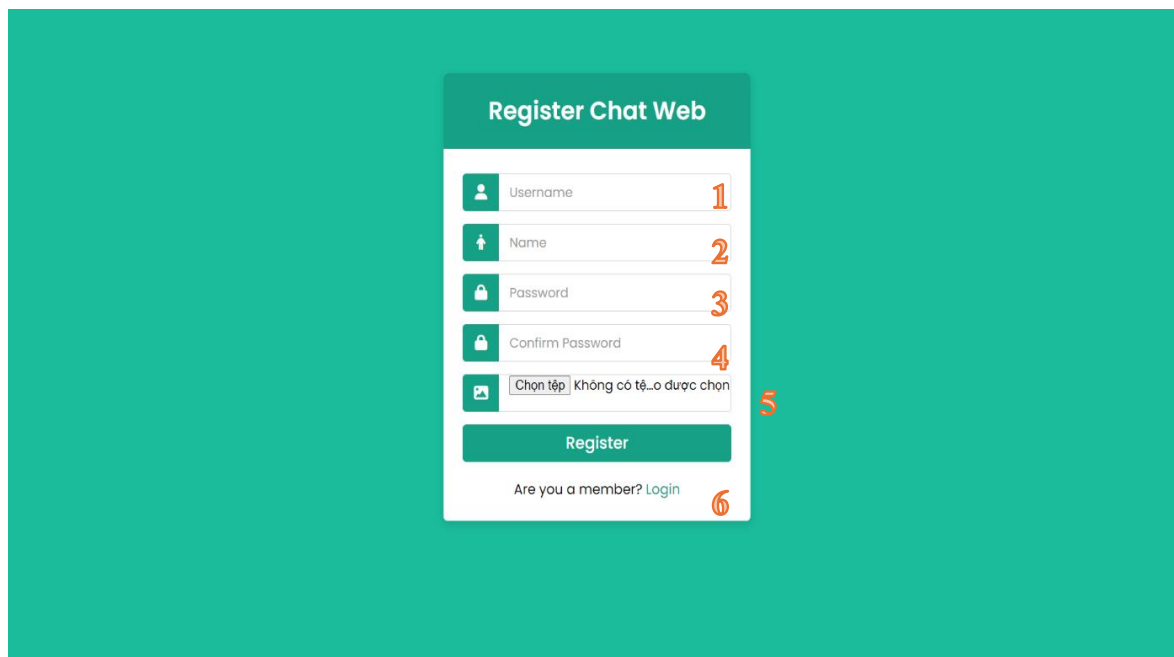
Sau khi người dùng nhập thông tin tài khoản và bấm Login, dữ liệu sẽ được gửi tới Server bằng phương thức Post. Ở Server, dữ liệu sẽ được nhận và đem so sánh với dữ liệu người dùng trong cơ sở dữ liệu. Nếu dữ liệu trùng khớp sẽ mở phiên làm việc (Session) cho người dùng, sau đó cập nhật trạng thái người dùng trong cơ sở dữ liệu và chuyển người dùng tới trang trò chuyện. Khi người dùng nhập sai hay dữ liệu không có trong cơ sở dữ liệu trang sẽ tải lại cho người dùng đăng nhập lại.

Khi người dùng chưa có tài khoản, người dùng có thể tạo tài khoản bằng đường dẫn (3) [Signup now](#).

## 2. Register

### 1. Giao diện

Ở giao diện Login có đường dẫn truy cập qua giao diện Register. Ở giao diện Register người dùng cần nhập đầy đủ thông tin để tiến hành đăng ký tài khoản mới để sử dụng ứng dụng chat.

The image shows a web registration form titled "Register Chat Web" on a teal background. The form is white with a green header. It contains several input fields: "Username" (1), "Name" (2), "Password" (3), "Confirm Password" (4), and an "Avatar" section (5) with a "Chọn tệp" button and a note "Không có tệp nào được chọn". Below these is a green "Register" button. At the bottom, there is a link "Are you a member? Login" (6). Red numbers 1 through 6 are placed to the right of each corresponding field or element to indicate specific requirements or features.

Hình 10 Giao diện đăng ký (Register)

- (1) “Username” tên tài khoản mà người dùng muốn tạo. Tối thiểu 50 kí tự, không gõ dấu tiếng việt. Ví dụ: Được phép ab12s, không được abc123s.
- (2) “Name” tên của người dùng tối đa 150 kí tự. Ví dụ: Nguyễn Trung Tín.
- (3) và (4) “Password” và “Comfirm Password” cần đồng nhất, người dùng nhập mật khẩu ở (3) thì ở (4) người dùng cần nhập giống y hệt. Tối thiểu cần nhập 8 kí tự và tối đa 50 kí tự.
- (5) “Avatar” người dùng sẽ tải lên một tấm hình để làm ảnh đại diện, bắt buộc người dùng phải tải hình lên. Định dạng hình ảnh nằm trong các định dạng sau: .jpg, .png, .jpeg, .gif, .raw
- (6) Đường dẫn người dùng quay lại trang đăng nhập (Login)

## 2. Xử lý

Khi người dùng truy cập vào trang, Server sẽ nhận được thông điệp doGet từ Client yêu cầu lấy giao diện đăng ký (Register). Giao diện sẽ được nạp lên cho người dùng. Trước đó, Server sẽ kiểm tra phiên làm việc (Session) xem người dùng đã đăng nhập chưa, nếu người dùng đã đăng nhập hệ thống sẽ đưa người dùng về giao diện trò chuyện chính.

Khi người dùng nhập đầy đủ thông tin để đăng ký tài khoản người dùng mới, dữ liệu sẽ được gửi lên Server thông qua Method “Post”. Ở Server, dữ liệu sẽ nhận ở phương thức doPost.

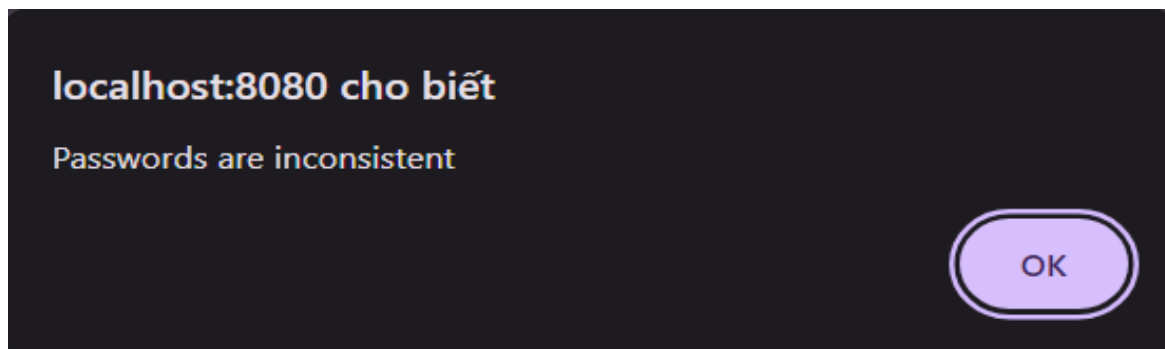
Mật khẩu sẽ được mã hoá bằng kỹ thuật mã hoá AES-256. AES được viết tắt từ Advanced Encryption Standard (chuẩn mã hoá cấp cao) đây là chuẩn mã hoá dữ liệu thuộc kiểu thuật toán “mã hoá khối” (block cipher) do Viện tiêu chuẩn và Công nghệ quốc gia Hoa Kỳ phát hành vào năm 2001.

Thông tin về chuẩn mã hoá AES “*Tiêu chuẩn AES mã hoá đối xứng là phổ biến và an toàn nhất, nó có ba tiêu chuẩn là AES 128, AES 192 và AES 256. Sự khác biệt giữa chúng là trong chìa khoá khi được xác định bởi số lượng bit. Với số lượng bit nhiều hơn thì mức bảo mật tốt hơn và ngược lại. Nói một cách đơn giản, mã hoá cấp Quân sự - mã hoá AES 256 có kích thước khoá 256 bit. Hiện nay, AES 256 là một trong những phương pháp mã hoá được áp dụng khá phổ biến vì phương thức mật mã nhanh và đáng tin cậy. Cho đến nay, chưa có máy tính hoặc con người nào có thể bẻ khoá mã hoá AES 256*” theo trang “<https://viettelidc.com.vn/tin-tuc/ma-hoa-aes-256-la-gi-moi-dieu-ve-ma-hoa-aes-256-ma-nguoi-dung-nen-biet>” .

Server sẽ dùng Username và Password (đã được mã hoá) để tìm trong cơ sở dữ liệu để chắc rằng chưa có người dùng với Username và Password như trên tồn tại trong hệ thống. Sau đó, Server sẽ tạo đối tượng người dùng mới và lưu vào cơ sở dữ liệu. Hoàn tất quá trình người dùng sẽ được chuyển đến trang đăng nhập (Login) để tiến hành đăng nhập và sử dụng hệ thống.

Khi người dùng có tài khoản và không muốn đăng ký tài khoản mới có thể truy cập đường dẫn (6) để trở lại trang đăng nhập để tiến hành đăng nhập hệ thống.

Khi người dùng nhập không đồng nhất mật khẩu hệ thống sẽ cảnh báo bằng Alert.

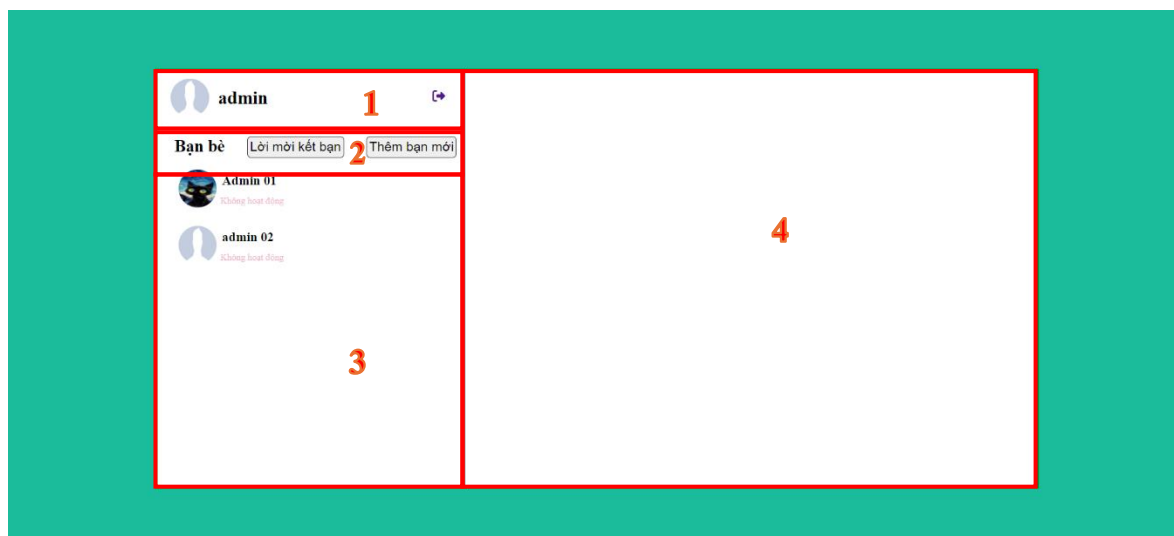


Hình 11 Hộp thoại cảnh báo người dùng nhập không đồng nhất mật khẩu.

### 3. Giao diện trò chuyện chính

#### 1. Giao diện

Sau khi đăng nhập người dùng sẽ được chuyển tới giao diện trò chuyện chính. Server sẽ lấy thông tin người dùng và đưa vào giao diện đây lên cho người dùng phía Client.



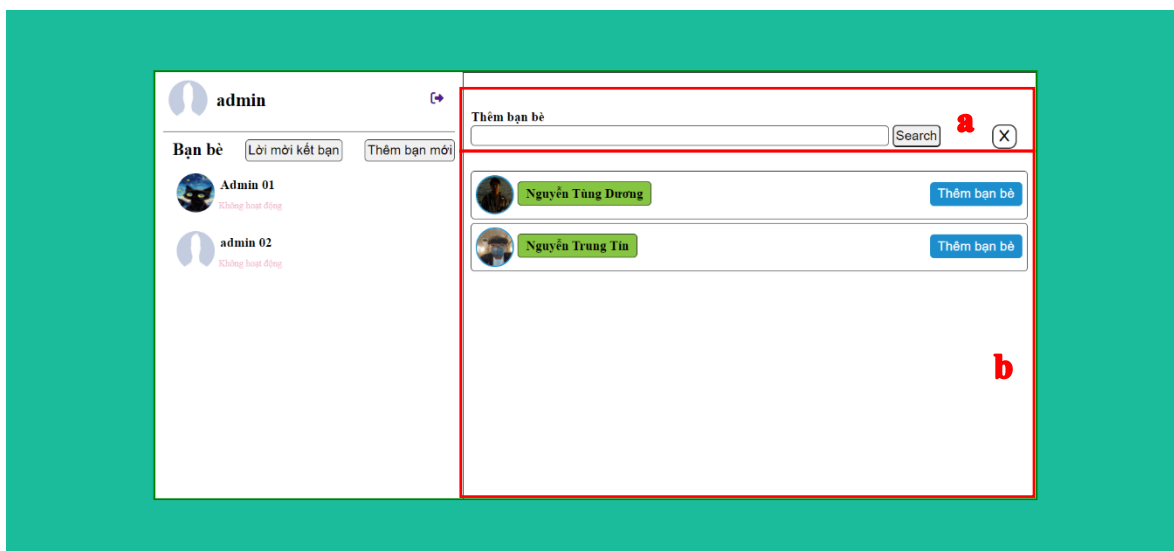
Hình 12 Giao diện trò chuyện chính



Hình 13 Nút đăng xuất

Ứng dụng sẽ chia thành 4 khu vực. Khu vực (1) dùng để hiển thị tên người dùng và nút đăng xuất. Ở khu vực (2) sẽ bao gồm tagname Bạn bè và 2 chức năng. “Lời

mời kết bạn” dùng để hiển thị danh sách những người dùng đã gửi lời mời kết bạn tới người dùng đang ở giao diện hiện tại. “Thêm bạn mới” chức năng này giúp người dùng tìm kiếm bạn bè theo [username] hoặc [name], với [name] (tên người dùng) có thể tìm gần đúng. Khu vực (3) sẽ bao gồm danh sách người dùng đã thành “Bạn bè”. Khu vực (4) dùng để hiển thị các chức năng ở khu vực (2) cũng như hiển thị hộp trò chuyện khi người dùng chọn Bạn bè ở khu vực (3) để thực hiện trò chuyện.



Hình 14 Giao diện trò chuyện với chức năng "Thêm bạn mới"

Ở giao diện trò chuyện chính với chức năng “Thêm bạn mới” chia khu vực (4) của chức năng trò chuyện chính thành 2 khu vực (a) và (b).

Khu vực (a) nơi hộp thoại tìm kiếm được đặt bên cạnh đó là nút đóng chức năng. Khu vực (b) là nơi hiển thị kết quả tìm kiếm của người dùng với nội dung là các tài khoản chưa gửi lời mời kết bạn.



Hình 15 Nút đóng giao diện

**Thêm bạn bè**

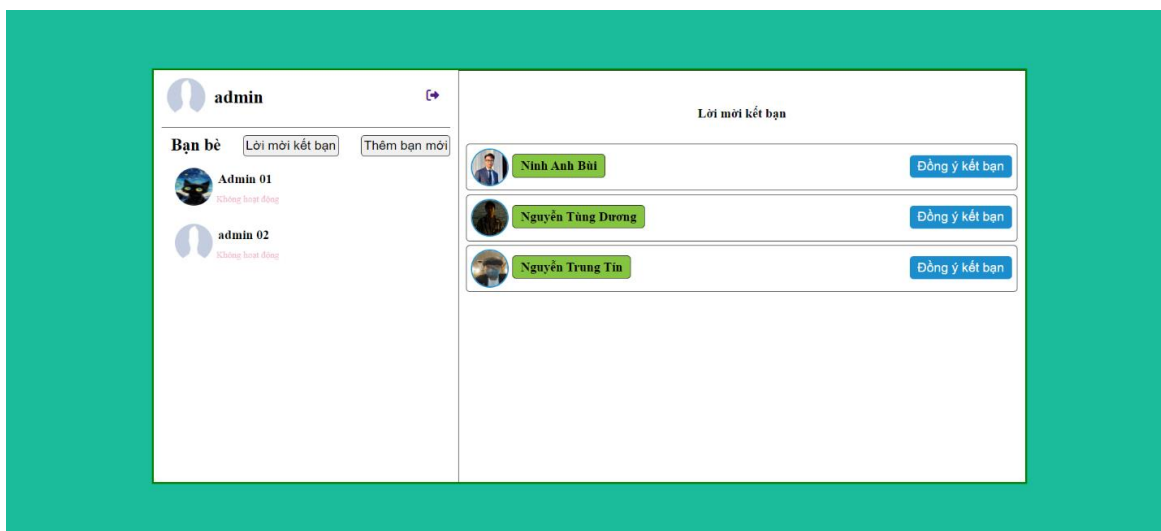
 

Hình 16 Thanh tìm kiếm

Khi người dùng gửi lời mời kết bạn nút “Thêm bạn bè” của người dùng sẽ thay đổi trạng thái.

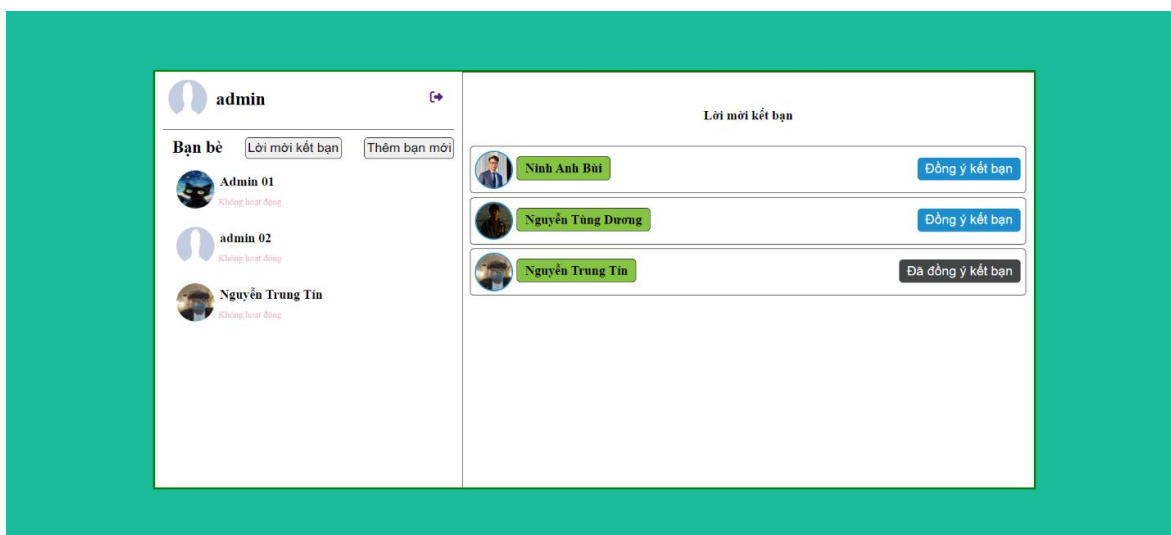


Hình 17 Nút "Thêm bạn bè" theo từng người dùng đã thay đổi trạng thái



Hình 18 Giao diện trò chuyện với chức năng "Lời mời kết bạn"

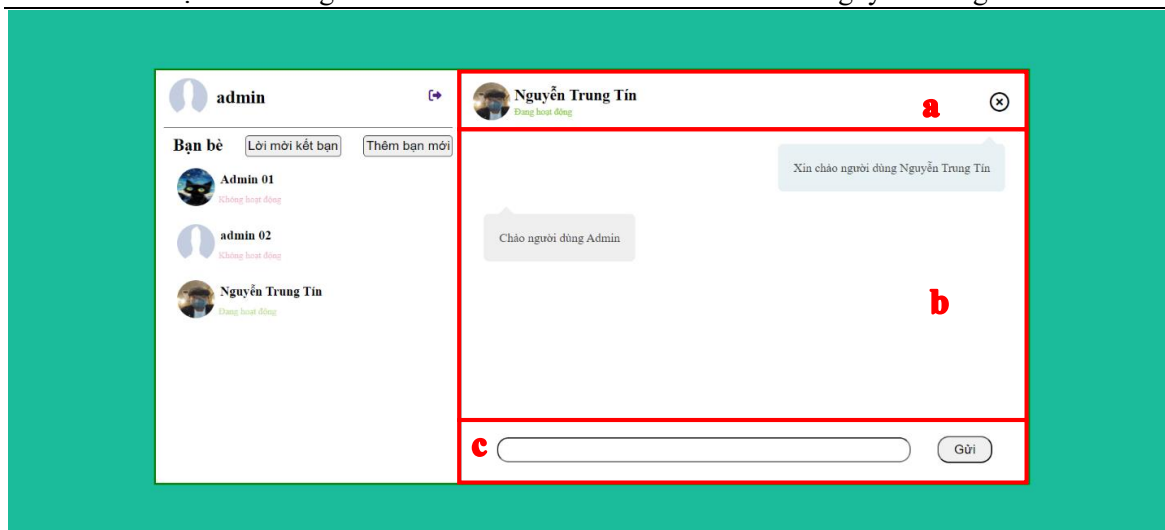
Ở chức năng “Lời mời kết bạn” sẽ biến khu vực (4) thành danh sách người dùng gửi lời mời kết bạn.



Hình 19 Đồng ý kết bạn

Khi người dùng đồng ý kết bạn nút “Đồng ý kết bạn” sẽ thay đổi trạng thái. Và người dùng đã được thêm vào bạn bè sẽ chuyển vào danh sách bạn bè.





Hình 20 Giao diện trò chuyện với hộp trò chuyện

Ở giao diện này, khu vực (4) sẽ chia thành 3 khu vực nhỏ gồm (a), (b) và (c). Khu vực (a) dùng hiển thị thông tin gồm tên và trạng thái hoạt động “người nhận” người tham gia trò chuyện với người dùng, nút đóng đoạn chat. Còn khu vực (b), đây là khu vực chứa cuộc trò chuyện gồm các đoạn tin nhắn của người dùng và “bạn bè” người dùng tham gia cuộc trò chuyện. Tin nhắn được hiển thị theo thứ tự và chia thành 2 bên gồm bên trái do “bạn bè” người dùng tham gia trò chuyện gửi và bên phải do người dùng đăng nhập gửi. Ở khu vực (c) đây là nơi để người dùng nhập tin nhắn.



Hình 21 Trạng thái hoạt động

Trạng thái hoạt động được đánh kèm bên dưới thông tin người dùng. Có 2 trạng thái là “Đang hoạt động” khi người dùng online và trạng thái “Không hoạt động” khi người dùng offline. Cho dù bạn bè có ở trạng thái nào thì tin nhắn vẫn được gửi.

## 2. Xử lý

Khi người dùng đăng nhập thành công Server sẽ nạp giao diện trò chuyện chính cùng với thông tin người dùng lên Client.

Giao diện chính là sự kết hợp giữa nhiều Servlet xử lý riêng lẻ và Servlet chính “ChatWebSocket”.

ChatWebSocket sử dụng API WebSocket đã được tích hợp sẵn trong kho thư viện của Jakarta. Khi người dùng đăng nhập thành công Client (Web Browser) sẽ gửi một yêu cầu nâng cấp HTTP thành WebSocket. ChatWebSocket sẽ mở phiên làm việc (Session). Để sử dụng cùng lúc nhiều người dùng ta sử dụng kỹ thuật EndPoint tạo điểm cuối bằng username của người dùng vì username của người dùng là duy nhất nên không có hiện tượng trùng username giữa 2 hay nhiều người dùng.

ChatWebSocket với phương thức “onMessage” sẽ lắng nghe các hoạt động **trò chuyện** của người dùng. Khi có một tin nhắn mới, tin nhắn sẽ gửi lên Server. Server sẽ kiểm tra người dùng gồm người gửi (Sender) và người nhận (Receiver) có boxchat trò chuyện riêng chưa, nếu chưa thì tạo boxchat trò chuyện cho 2 người dùng trên. Đã có boxchat thì sẽ lấy Id của boxchat gán vào đối tượng tin nhắn (message) và tiến thành lưu vào cơ sở dữ liệu. Đồng thời chuyển tin nhắn lên giao diện người nhận nếu người nhận cùng tồn tại trong boxchat. Các hoạt động diễn ra tức thì vẫn đảm bảo ứng dụng hoạt động thời gian thực.

Khi người dùng gặp lỗi khi trò chuyện Server sẽ quảng lỗi ra Terminal. Ở giao diện người dùng WebSocket sẽ khởi động lại.

Khi người dùng đóng giao diện trò chuyện hoặc tắt trình duyệt web (Web Browser), Client sẽ gọi phương thức “onClose” trên Server và tiến hành xóa phiên làm việc, cập nhật lại trạng thái hoạt động của người dùng trên cơ sở dữ liệu của hệ thống.

Chức năng “Thêm bạn bè” Client sẽ dùng Ajax kết nối với Server gọi phương thức tìm kiếm người dùng trên Server để trả về danh sách người dùng phù hợp với nội dung “từ khoá tìm kiếm”. Vì sử dụng Ajax nên không cần tải lại trang dữ liệu vẫn được nạp lên giao diện người dùng.

Chức năng “Lời mời kết bạn” cũng tương tự như “Thêm bạn bè”, Client cũng thông qua Ajax gửi thông điệp để lấy dữ liệu về người dùng đã gửi lời mời kết bạn.

Danh sách bạn bè được hiển thị bằng cách sử dụng Ajax gửi thông điệp lên Server lấy những người dùng đã thành bạn bè. Danh sách được cập nhật liên tục bằng hàm **setInterval(loadListUserContactMe, 1000);** cứ 1s sẽ gửi thông điệp 1 lần. Việc gửi thông điệp từ Client giúp giảm thiểu xử lý cho Server khi Server phải gửi thông điệp liên tục.

Đối với đăng xuất, khi người dùng bấm vào nút đăng xuất, Server sẽ tiến hành xoá phiên làm việc của người dùng và cập nhật trạng thái của người dùng trên cơ sở dữ liệu.

## **CHƯƠNG 4**

### **KẾT LUẬN VÀ ĐÁNH GIÁ**

#### **I. KẾT LUẬN**

##### **1. Kết quả**

Qua quá trình học tập nghiên cứu, áp dụng những kiến thức có được xây dựng được trang Website trò chuyện thời gian thực (Chat Realtime). Cung cấp cho người dùng những chức năng cơ bản như đăng nhập, đăng ký, tìm người dùng, kết bạn, hiển thị danh sách bạn bè, danh sách người dùng gửi lời mời kết bạn. Đáp ứng đủ cho người dùng với mục đích trò chuyện.

Xây dựng được giao diện cơ bản, thân thiện với người dùng. Có sự tương tác giữa Website và người dùng. Cung cấp gần như đầy đủ các chức năng trên giao diện của người dùng.

##### **2. Thu thập về mặt chuyên môn**

Sau khi hoàn thành dự án, giúp tôi có hiểu thêm về các xây dựng một trang Website. Áp dụng được kiến thức trên nhà trường cũng như những kiến thức tìm hiểu thêm trong quá trình cài đặt và kiểm thử ứng dụng.

Về mặt giao diện, áp dụng các kiến thức về HTML/CSS tạo giao diện người dùng. Qua đó hiểu hơn về bố cục – layout, các thẻ, các thuộc tính mà HTML/CSS cung cấp cho lập trình viên. Biết cách vận dụng những tài nguyên có sẵn từ HTML/CSS triển khai các giao diện cơ bản. Ứng dụng kỹ thuật Ajax và JavaScript làm giao diện trở nên linh động, có tính tương tác giữa người dùng và trang Web.

Về mặt kỹ thuật - xử lý dữ liệu, biết cách ứng dụng mô hình MVC vào thiết kế cài đặt ứng dụng Web. Ứng dụng Maven để quản lý dự án Java Web, khả năng tìm kiếm thư viện hỗ trợ quá trình xây dựng. Có thêm kiến thức về DAO và RowMapper tương tác với dữ liệu trong cơ sở dữ liệu. Khả năng tư duy logic giải quyết vấn đề về các thao tác. Biết thêm về mã hoá dữ liệu cụ thể là kỹ thuật mã hoá AES 256. Qua đó vận dụng vào mã hoá mật khẩu của người dùng.

Qua dự án giúp tôi khắc phục các mặt hạn chế như về giao diện, xử lý dữ liệu và xây dựng cơ sở dữ liệu. Không những vậy, sau khi nghiên cứu và xây dựng học thêm được nhiều kiến thức mới từ việc xây dựng quản lý dữ liệu, áp dụng mã hoá vào cơ sở dữ liệu, lưu trữ hình ảnh cũng như xử lý hình ảnh ở dạng byte.

### **3. Kinh nghiệm**

Có thêm nhiều kinh nghiệm về xử lý lỗi. Nâng cao khả năng bắt lỗi trong lập trình với ngôn ngữ Java. Biết thêm nhiều mã lỗi khi tương tác với cơ sở dữ liệu. Còn hơn thế, quá dự án cho tôi hiểu thêm về việc xây dựng một trang Web cần những gì và khả năng áp dụng các mô hình kỹ thuật vào xây dựng dự án điển hình như mô hình MVC. Học thêm nhiều về JavaScript. Có kinh nghiệm lưu trữ hình ảnh trong cơ sở dữ liệu.

## **II. ĐÁNH GIÁ**

### **1. Tổng quan**

Hoàn thành tốt các mục tiêu đã đề ra. Tuy còn nhiều bất cập trong việc thiết kế cài đặt nhưng vẫn đảm bảo các chức năng hoạt động bình thường.

### **2. Ưu điểm**

Ứng dụng sử dụng Java core làm nền tảng nên tính ổn định rất cao. Thừa hưởng sự đa nền tảng có thể chạy trên nhiều hệ điều hành khác nhau. Tính tương thích cao với Server. Xây dựng theo mô hình MVC nên việc bảo hành sửa chữa, nâng cấp dễ dàng thực hiện. Xử lý dữ liệu được tách riêng ở DAO, việc này giúp cho ứng dụng có thể thêm các cơ sở dữ liệu bên ngoài bằng cách thêm các DAO xử lý. Tính ổn định cao, vì ngôn ngữ Java được hình thành phát triển trong thời gian dài và thường xuyên cập nhật phiên bản mới hỗ trợ rất nhiều công nghệ mới. Bên cạnh đó, ứng dụng xây dựng các hàm xử lý riêng biệt làm đơn giản quá trình nâng cấp không ảnh hưởng tới những chức năng đã có trước đó. Giao diện đơn giản, dễ tiếp cận người sử dụng.

### **3. Nhược điểm**

So với các ứng dụng Realtime sử dụng Node Js để xây dựng thì ứng dụng Realtime sử dụng Java có phần lép vế về mặt hiệu năng, khả năng “Thời gian thực”. Bên cạnh đó, việc lưu trữ hình ảnh người dùng trong cơ sở dữ liệu dẫn đến rủi ro mất

dữ liệu khi cơ sở dữ liệu gặp rắc rối. Do lưu trữ trong cơ sở dữ liệu nên hiệu năng tải hình ảnh lên Client của ứng dụng có phần giảm. Với lưu trữ trên Server kết hợp lưu đường dẫn trong cơ sở dữ liệu sẽ trải qua các bước sau: lấy đường dẫn dán vào thẻ Img, thẻ Img truy cập lấy hình ảnh, hiển thị hình ảnh. Thì quá trình lấy hình ảnh lưu cơ sở dữ liệu sẽ trải qua: Client lấy id người dùng, sau đó gửi lên Server yêu cầu lấy ảnh người dùng, Server phản hồi lấy dữ liệu chuyển đổi dạng thành hình ảnh và “ghi vào Client”, Client nhận hình ảnh gán vào thẻ Img, hiển thị hình ảnh. Do đó, có sự chậm trễ trong xử lý.

Danh sách bạn bè online dùng kỹ thuật Ajax để gọi dữ liệu từ Server về cách nhau 1s. Việc gọi dữ liệu vậy nếu có nhiều người dùng truy cập cùng lúc sẽ ảnh hưởng tới hiệu năng của ứng dụng. Thiết bị làm Server khởi chạy cũng đóng vai trò quan trọng trong quá trình đem lại trải nghiệm sử dụng cho người dùng.

Do thời gian hạn hẹp, vẫn còn nhiều chức năng chưa hoàn thành như cập nhật thông tin người dùng, xoá tin nhắn, huỷ kết bạn, v.v

Giao diện còn nhiều hạn chế, cần bổ sung các tính năng thay đổi giao diện giúp tiếp cận nhiều người dùng hơn. Có thể áp dụng các Framework xử lý giao diện như Tailwin, Bootstrap để đạt được hiệu quả cao nhất.

Bên cạnh các vấn đề trên, vấn đề bảo mật vẫn đóng vai trò quan trọng. Cần áp dụng các kỹ thuật bảo mật để giúp người dùng an toàn khi sử dụng ứng dụng. Việc mã hoá dữ liệu đóng vai trò quan trọng đối với các ứng dụng trò chuyện thời gian thực. Việc bị đánh cắp dữ liệu ở các ứng dụng chat không còn quá xa lạ với mọi người. Để tăng tính tin cậy cần thường xuyên cập nhật các kỹ thuật bảo mật mới.

#### **4. Hướng phát triển**

Có thể dùng framework Spring Boot để tăng hiệu suất làm việc. Vì Spring Boot cung cấp nhiều thư viện API giúp quá trình code trở nên nhanh hơn khi sử dụng Java làm nền tảng phát triển.

Cải thiện giao diện người dùng hướng tới nhiều thiết bị sử dụng Chat Web hơn. Giao diện cần đổi mới phù hợp với nhu cầu người dùng. Tăng tính tương tác giữa người dùng và hệ thống.

Cập nhật thêm các tính năng bảo mật, mã hoá dữ liệu. Giữ người dùng tránh xa các đối tượng có ý đồ xấu. Mã hoá tin nhắn 2 chiều tăng tính bảo mật của cuộc trò chuyện.

Tiếp tục hoàn thiện các chức năng còn thiếu của ứng dụng như: cập nhật thông tin người dùng, xoá bạn bè, xoá tin nhắn, tìm kiếm tin nhắn v.v Khắc phục các điểm yếu về lưu trữ và xử lý. Tiếp cận các kỹ thuật lưu trữ hiệu quả nhằm tăng hiệu suất hoạt động của Chat Web.

### Tài liệu tham khảo

- [1] **Josh Juneau.** Website: <https://blogs.oracle.com/javamagazine/post/how-to-build-applications-with-the-websocket-api-for-java-ee-and-jakarta-ee>.
- [2] **Website:** <https://www.w3schools.com/>
- [3] **Website:** <https://jakarta.ee/specifications/websocket/>
- [4] **Viettel IDC.** Website: <https://viettelidc.com.vn/tin-tuc/ma-hoa-aes-256-la-gi-moi-dieu-ve-ma-hoa-aes-256-ma-nguoi-dung-nen-biet>
- [5] **Website:** <https://jakarta.ee/specifications/platform/9/apidocs/>