

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH



COMPUTER NETWORKING - CO3093

CHAT APPLICATION

GVHD : Hoàng Lê Hải Thanh

L08_Group 4:	1. Lô Hoàng Khôi Nguyên (NT)	2013914
	2. Ngô Thanh Phúc	2011848
	3. Trần Quốc Dũng	2011026
	4. Trần Đăng Khoa	2013515

TP. Hồ Chí Minh, tháng 11 năm 2022



Mục lục

1	Phân tích yêu cầu	2
1.1	Yêu cầu chức năng	2
1.1.1	Yêu cầu hệ thống	2
1.1.2	Yêu cầu người dùng	2
1.2	Yêu cầu phi chức năng	2
2	Mô tả chức năng	3
3	Mô hình luồng dữ liệu	4
4	Đánh giá kết quả đạt được	4
5	Hướng dẫn sử dụng	4
5.1	Hướng dẫn khởi tạo Server	4
5.2	Các thao tác trên client	5
6	Source code	9

1 Phân tích yêu cầu

1.1 Yêu cầu chức năng

1.1.1 Yêu cầu hệ thống

- Hệ thống có thể cho phép người dùng trao đổi tin nhắn với nhau
- Việc trao đổi tin nhắn của người dùng phải thông qua giao thức P2P
- Việc xác thực của người dùng yêu cầu kết nối đến một server

1.1.2 Yêu cầu người dùng

- Người dùng có thể cùng lúc kết nối với nhiều người khác
- Người dùng có danh sách các người dùng đã kết nối trước đó (danh sách bạn bè)

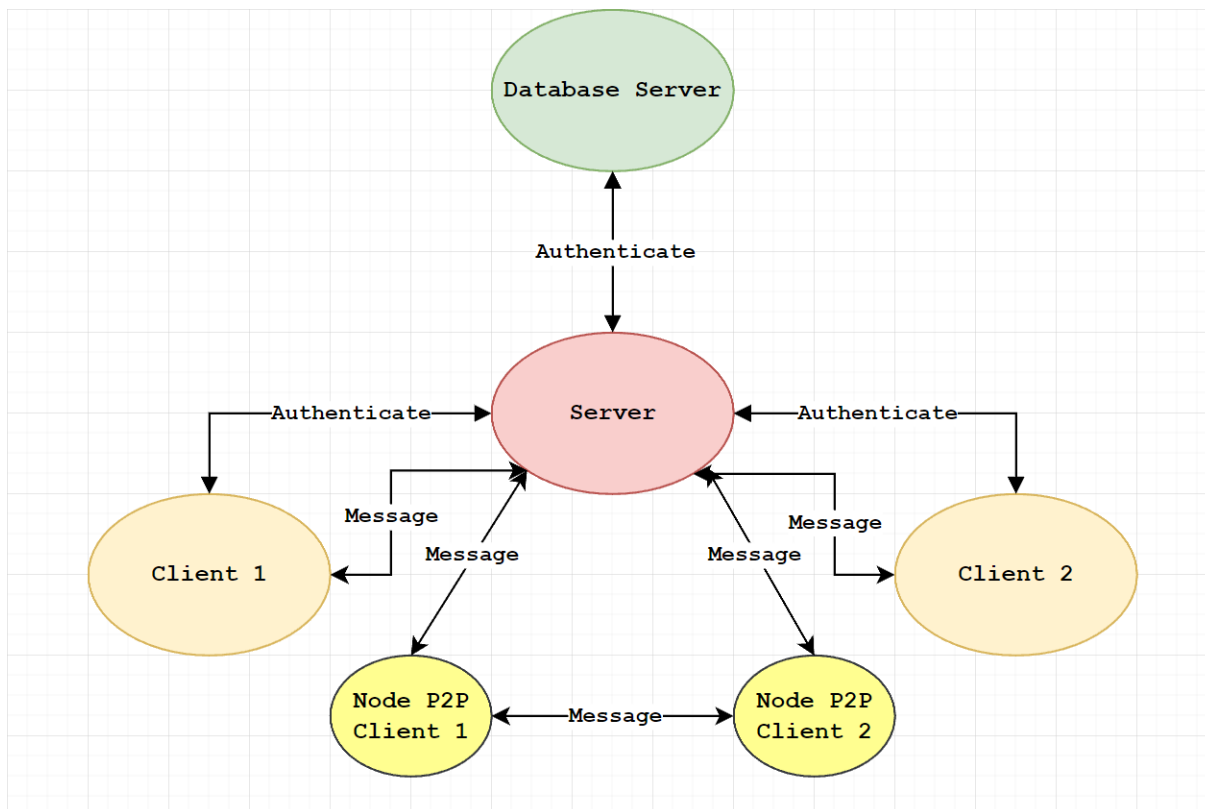
1.2 Yêu cầu phi chức năng

- Việc trao đổi tin nhắn giữa những người dùng phải dưới 1.0s

2 Mô tả chức năng

Scope	Class	Hàm	Mô tả
Server	socket	on(event, callback)	Dựa vào event truyền vào để callback cho webSocket giao tiếp giữa client và server
	node	listen(port, callback)	Khởi tạo node P2P cho một client với PORT định sẵn
	ws	on(event, callback)	Dựa vào event trong giao tiếp giữa client và server để callback
		send(data)	Gửi tin nhắn xuống client
		authenticate(uname, pwd, port)	Gửi thông tin xác thực của người dùng cùng với IP Address, PORT của Node P2P mà người dùng được server khởi tạo tới Database
		goOffline(uname)	Thay đổi trạng thái của người dùng trên Database khi người dùng ngắt kết nối
		getUserAddress(name, uname)	Lấy PORT và các trạng thái của người cần kết nối
		saveMessage(mes, uname)	Lưu lại tin nhắn được gửi từ người dùng này tới người dùng khác trên Database
Database Server		returnUserData(uname, pwd, ip, port)	Trả về server thông tin xác thực được người dùng gửi lên
		setOffline(uname)	Thay đổi trạng thái của người dùng trên Database khi có yêu cầu
		getUserAddress(name, username)	Lấy PORT và các trạng thái của người cần kết nối sau đó trả về cho server
		saveMessage(mes, username)	Lưu lại tin nhắn được gửi từ người dùng này tới người dùng khác trên Database
Client	socket	onopen(event, callback)	Thông báo mở kết nối giữa client và server
		onclose(event, callback)	Thông báo ngắt kết nối giữa client và server
		onmessage(event, callback)	Nhận dữ liệu giao tiếp từ server và callback
		send(data)	Gửi tin nhắn lên server
		startUp(data)	Khởi tạo client nếu như authenticate thành công
		sendAuthenticate()	Gửi thông tin xác thực nhập từ người dùng lên server thông qua webSocket
		checkInput(e)	Kiểm tra thông tin người dùng nhập có phải rỗng không
		createInputBar()	Tạo element input của chat
		createUsersListContainer()	Tạo element danh sách bạn bè
		createUsersOnList(name)	Tạo element từng người dùng thuộc danh sách
		createChat()	Tạo element khung chat
		createMessage(message, isUser)	Tạo element bubble chat

3 Mô hình luồng dữ liệu



4 Đánh giá kết quả đạt được

- Người dùng có thể được xác thực bằng tài khoản và mật khẩu
- Những người dùng có thể kết nối và liên lạc với nhau qua các Node P2P trên server

5 Hướng dẫn sử dụng

5.1 Hướng dẫn khởi tạo Server

- **Yêu cầu**
 - Máy đã được cài sẵn Node.js
 - Đã cài đặt các node_modules bằng lệnh `npm i`
- **Bước 1: Khởi chạy Web Server**

Để khởi chạy Web Server, ta sẽ cần vào thư mục source code của Web Server, và nhập lệnh `node server.js`. Lúc này, Server sẽ được chạy trên PORT 7270.

```
E:\testp2p>nodemon server.js
[nodemon] 2.0.20
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node server.js`
Client started at 7270
```

- **Bước 2:** Khởi chạy Database Server

Trước tiên, để có thể sử dụng được Database Server, ta cần thêm access URI của database vào trong file `mongoURI.json`. Vì lý do bảo mật nên URI này không được public trên GitHub mà chỉ xuất hiện ở trong bản báo cáo này.

Access URI:

`mongodb+srv://tryz:tryz1812@sldr.ors6dui.mongodb.net/`

Để chạy Database Server, ta cũng cần vào thư mục source code của Database Server, nhập lệnh `node index.js`. Lúc này, Database Server sẽ chạy trên PORT 60727.

```
E:\chatAppServer>nodemon index.js
[nodemon] 2.0.20
[nodemon] to restart at any time, enter `rs`
[nodemon] watching path(s): *.*
[nodemon] watching extensions: js,mjs,json
[nodemon] starting `node index.js`
```

5.2 Các thao tác trên client

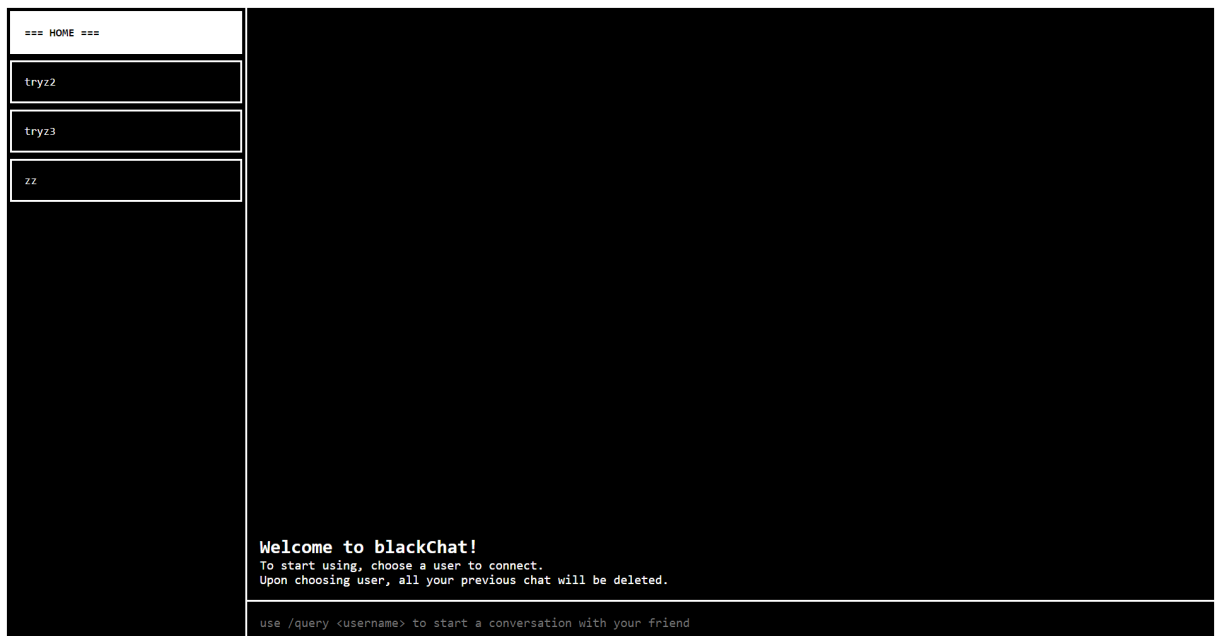
- **Chạy client trên trình duyệt**

Truy cập vào địa chỉ `http://localhost:7270` trên trình duyệt để bắt đầu sử dụng ứng dụng. Lúc này, ứng dụng sẽ yêu cầu chúng ta đăng nhập.



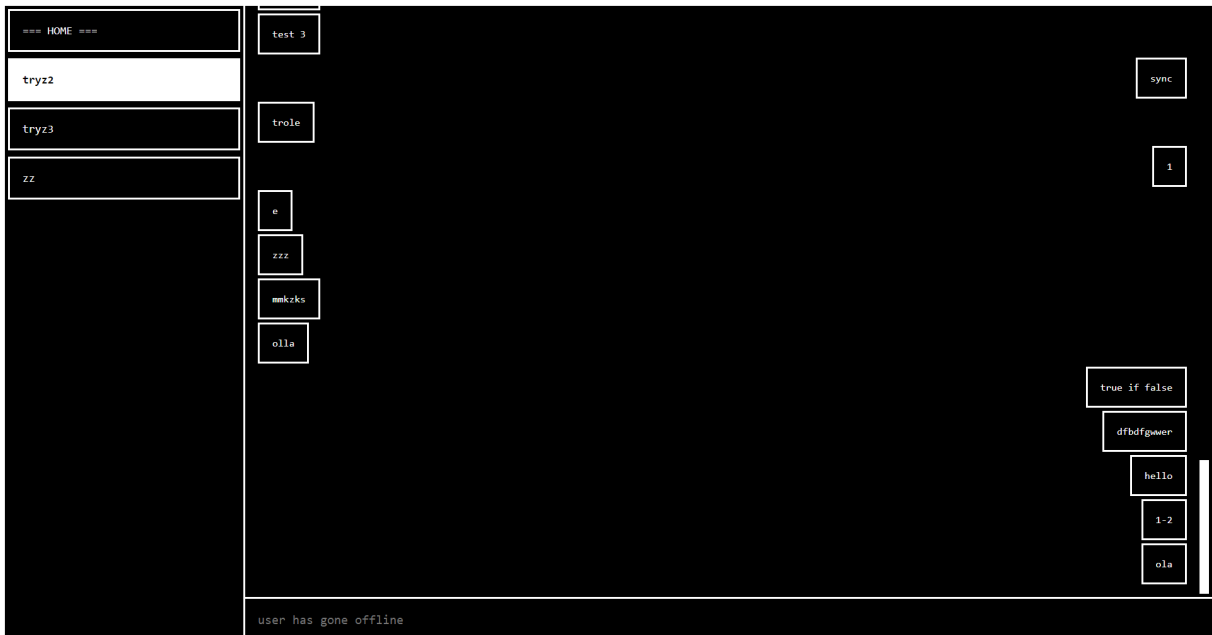
- **Giao diện chính**

Sau khi đăng nhập, người dùng sẽ được hướng tới trang giao diện chính của ứng dụng.

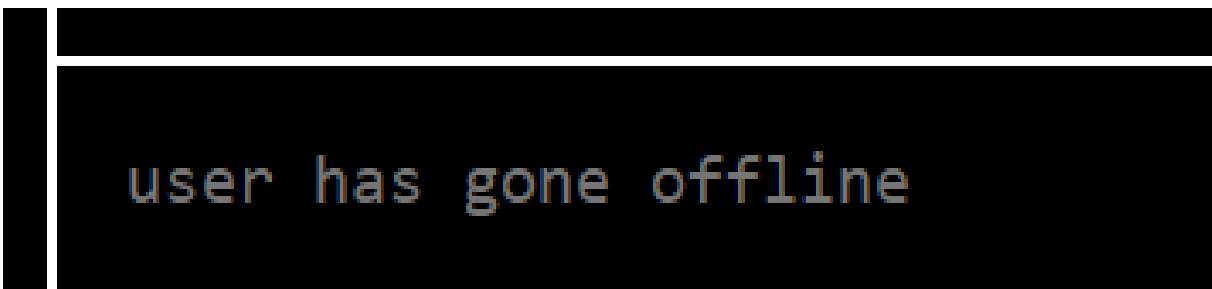


- **Kết nối tới một người là bạn bè**

Để kết nối tới một trong những người dùng là bạn bè, ta có thể click chuột vào tên của các người dùng bên cạnh.



Nếu người dùng còn lại đang Offline, ta sẽ không thể chat được với người dùng đó



Nếu người dùng Online, lúc này ta mới có thể chat



- **Gửi tin nhắn**

Để gửi tin nhắn, người dùng có thể nhập bất kỳ và ấn Enter. Phía đối diện cũng sẽ nhận được tin nhắn.



test2245

true if false

dfbdfgwwer

hello

1-2

ola

test1123

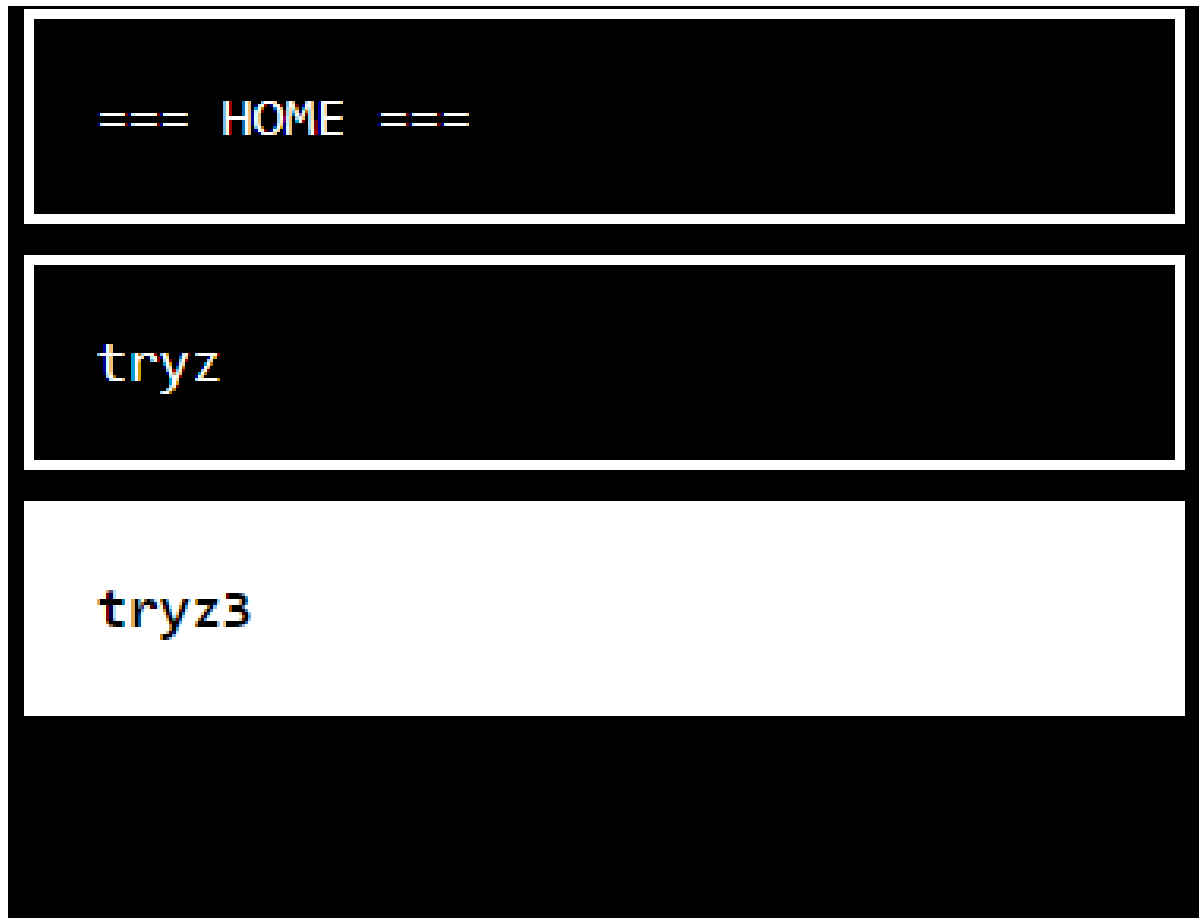
test2245

- **Kết bạn**

Để tiến hành kết bạn và gửi tin nhắn với một người nào đó, ta có thể dùng lệnh /query <ten_nguoi_dung> ở trang chủ.

/query tryz3|

Lúc này, người dùng mới sẽ được thêm vào



6 Source code

- **Web Server:** <https://github.com/FukutoTojido/blackChat>
- **Database Server:** <https://github.com/FukutoTojido/blackChatServer>