

# DA NANG UNIVERSITY OF SCIENCE AND TECHNOLOGY

LẬP TRÌNH MẠNG

---

## Báo cáo cuối kỳ

---

***Sinh viên: Nhóm 4***

Phạm Minh Tùng  
Trần Quốc Khánh  
Ngô Hoàng Gia

***Giáo viên hướng dẫn:***

TS. Nguyễn Văn Hiếu

Ngày 3 tháng 12 năm 2023



Bảng 1: **BẢNG PHÂN CÔNG CÔNG VIỆC**

STT	Họ và tên	Nhiệm vụ	Khối lượng
1	Phạm Minh Tùng	Code unicast, đa luồng thread, viết báo cáo phần 1,5	33.3
2	Ngô Hoàng Gia	Code broadcast server, đăng nhập, đăng ký, viết báo cáo phần 2,3, làm slide	33.3
3	Trần Quốc Khánh	Code hiển thị user online, đăng xuất, broadcast client, viết báo cáo 3,4	33.3

## Mục lục

<b>1</b>	<b>Tên đề tài</b>	<b>3</b>
<b>2</b>	<b>Giới thiệu</b>	<b>3</b>
<b>3</b>	<b>Các chức năng của ứng dụng</b>	<b>3</b>
3.1	Server . . . . .	3
3.1.1	Khởi tạo và Cấu hình Socket . . . . .	3
3.1.2	Quản lý người dùng . . . . .	4
3.1.3	Giao tiếp với client . . . . .	6
3.2	Client . . . . .	9
3.2.1	Khởi tạo và cấu hình Socket . . . . .	9
3.2.2	Giao diện người dùng và đăng nhập . . . . .	9
3.2.3	Giao tiếp với server . . . . .	10
<b>4</b>	<b>Mô tả các chức năng use-case</b>	<b>12</b>
4.1	Đăng Ký Người Dùng . . . . .	12
4.2	Đăng Nhập . . . . .	13
4.3	Gửi Tin Nhắn Riêng (Unicast) . . . . .	13
4.4	Gửi Tin Nhắn Broadcast . . . . .	14
4.5	Kiểm Tra Người Dùng Online . . . . .	15
4.6	Đăng Xuất . . . . .	16

<b>5</b>	<b>Thiết kế demo, test case</b>	<b>17</b>
5.1	Đăng nhập . . . . .	17
5.2	Đăng ký . . . . .	18
5.3	Broadcast . . . . .	19
5.4	Unicast . . . . .	19
5.5	Hiển thị người dùng online . . . . .	20
5.6	Đăng xuất . . . . .	20

# 1 Tên đề tài

Lập trình socket: Chat

## 2 Giới thiệu

Lập trình socket là một công cụ cơ bản cho giao tiếp mạng. Nó cho phép các tiến trình như server-client, client-client giao tiếp với nhau bằng cách sử dụng socket. Trong bài cuối kỳ, chúng em triển khai một ứng dụng nhỏ của socket là cho phép các client khác nhau có thể nhắn tin với nhau.

## 3 Các chức năng của ứng dụng

### 3.1 Server

Các chức năng chính:

#### 3.1.1 Khởi tạo và Cấu hình Socket

Sử dụng Winsock để khởi tạo và cấu hình socket để lắng nghe kết nối từ các client.

```
1      WSADATA wsaData;
2      if (WSAStartup(MAKEWORD(2, 2), &wsaData) != 0) {
3          std::cerr << "WSAStartup failed" << std::endl;
4          return -1;
5      }
6
7      int serverSocket = socket(AF_INET, SOCK_STREAM, 0);
8      if (serverSocket == INVALID_SOCKET) {
9          std::cerr << "Error creating socket" << std::endl;
10         WSACleanup();
11         return -1;
12     }
13
```

```

14     sockaddr_in serverAddr;
15     serverAddr.sin_family = AF_INET;
16     serverAddr.sin_port = htons(8080);
17     serverAddr.sin_addr.s_addr = INADDR_ANY;
18
19     if (bind(serverSocket, (struct sockaddr*)&serverAddr
20         , sizeof(serverAddr)) == SOCKET_ERROR) {
21         std::cerr << "Error_binding_socket" << std::endl
22             ;
23         closesocket(serverSocket);
24         WSACleanup();
25         return -1;
26     }
27
28     if (listen(serverSocket, SOMAXCONN) == SOCKET_ERROR)
29     {
30         std::cerr << "Error_listening_on_socket" << std
31             ::endl;
32         closesocket(serverSocket);
33         WSACleanup();
34         return -1;
35     }

```

### 3.1.2 Quản lý người dùng

Đọc danh sách người dùng từ file "users.txt" khi khởi động.

```

1     bool loadUsers() {
2         std::ifstream file("users.txt");
3         if (!file.is_open()) {
4             std::cerr << "Unable_to_open_users.txt_file" <<
5                 std::endl;
6             return false;
7         }
8         User user;
9         while (file >> user.username >> user.password) {
10             user.socket = INVALID_SOCKET; // Socket will be
11                 assigned after login

```

```

10         user.check = 0;
11         users.push_back(user);
12     }

```

Hỗ trợ đăng ký và đăng nhập cho người dùng.

```

1     if (choice == "1") {
2         // Register
3         recv(clientSocket, buffer, sizeof(buffer),
4             0);
5         std::string username(buffer);
6         recv(clientSocket, buffer, sizeof(buffer),
7             0);
8         std::string password(buffer);
9         User newUser;
10        bool ktra=true;
11        for (User& user : users) {
12            if (strcmp(username.c_str(),user.
13                username.c_str())==0)
14            {
15                send(clientSocket, "
16                    register_fail,username_has_
17                    been_used", sizeof("
18                    register_fail,username_has_
19                    been_used"), 0);
20                ktra=false;
21                break;
22            }
23        }
24        if(ktra){
25            newUser.username = username;
26            newUser.password = password;
27            newUser.socket = INVALID_SOCKET;
28            newUser.check = 0;
29            users.push_back(newUser);
30            if (saveUser(newUser)) {
31                send(clientSocket, "register_success",
32                    sizeof("register_success"), 0);
33            } }

```

```

26         } else if (choice == "2") {
27             // Login
28             recv(clientSocket, buffer, sizeof(buffer),
29                 0);
30             std::string username = buffer;
31             recv(clientSocket, buffer, sizeof(buffer),
32                 0);
33             std::string password = buffer;
34             User b;
35             b.username = username;
36             b.password = password;
37             bool loginSuccess = false;
38
39             for (User& user : users) {
40                 if (user.username == b.username && user.
41                     password == b.password && user.socket
42                     == INVALID_SOCKET &&
43                     user.check == 0) {
44                     user.socket = clientSocket;
45                     user.check = 1;
46                     loginSuccess = true;
47                     break;
48                 }
49             }
50         }

```

Gửi thông báo đăng ký thất bại, nếu tên người dùng đã được đăng ký.

Gửi thông báo đăng nhập thành công hoặc thất bại về client.

Lưu trạng thái đăng nhập và thông tin socket của người dùng.

### 3.1.3 Giao tiếp với client

Xử lý yêu cầu gửi tin nhắn riêng (Unicast) từ client.

```

1 void sendMessageToUser(const std::string& message, const
2     std::string& recipientUsername, SOCKET senderSocket)
3 {
4     for (const User& user : users) {

```

```

3         if (user.socket != INVALID_SOCKET && user.socket
4             != senderSocket && user.username ==
5             recipientUsername) {
6             send(user.socket, message.c_str(), message.
7                 size() + 1, 0);
            }
        }
    }
}

```

Xử lý yêu cầu gửi tin nhắn cho tất cả mọi người (Broadcast) từ client.

```

1     void broadcastMessage(const std::string& message,
2                           SOCKET senderSocket) {
3     for (const User& user : users) {
4         if (user.socket != INVALID_SOCKET && user.socket
5             != senderSocket) {
6             send(user.socket, message.c_str(), message.
7                 size() + 1, 0);
            }
        }
    }
}

```

Xử lý yêu cầu lấy danh sách người dùng đang online.

```

1     recv(clientSocket, buffer, sizeof(buffer), 0);
2     string message = buffer;
3     string onlineUsers = "Users are online: ";
4     for (User& user : users) {
5         if (user.check == 1) {
6             if (strcmp(message.c_str(), user.username.
7                 c_str()) == 0) {
8                 continue;
9             }
            onlineUsers += "[" +
                user.username + "]" +
                " ";
        }
    }
}

```



```

10         }
11     }
12     send(clientSocket,onlineUsers.
        c_str(),onlineUsers.size()
        +1,0);
13 }

```

Xử lý yêu cầu đăng xuất và đưa người dùng về trạng thái đăng nhập.

```

1     recv(clientSocket, buffer, sizeof(buffer), 0);
2     string message = buffer;
3     for (User& user : users) {
4         if (strcmp(message.c_str(),
5             user.username.c_str())
6             ==0) {
7             user.check = 0;
8             user.socket =
9                 INVALID_SOCKET;
10            send(clientSocket,"You_
                was_logged_out",sizeof
                ("You_was_logged_out")
                ,0);
11            break;
12        }
13    }

```

Xử dụng đa luồng để có thể xử lý nhiều client.

```

1 std::vector<std::thread> clientThreads;
2 std::thread clientThread(handleClient, clientSocket);
3     clientThreads.push_back(std::move(clientThread))
4     ;

```

## 3.2 Client

### 3.2.1 Khởi tạo và cấu hình Socket

Sử dụng Winsock để khởi tạo và cấu hình socket để kết nối đến server.

### 3.2.2 Giao diện người dùng và đăng nhập

Hiển thị menu cho phép người dùng đăng ký hoặc đăng nhập.

Gửi yêu cầu đăng ký hoặc đăng nhập đến server.

Hiển thị thông báo đăng ký hoặc đăng nhập thành công hoặc thất bại.

```
1      std::cout << "1. Register\n2. Login\nChoose_\noption: ";
2      int choice;
3      std::cin >> choice;
4
5      send(clientSocket, std::to_string(choice).c_str(), sizeof(choice), 0);
6
7      if (choice == 1) {
8          // Register
9          std::string username, password;
10         std::cout << "Enter_username: ";
11         std::cin >> username;
12         send(clientSocket, username.c_str(), strlen(username.c_str()) + 1, 0);
13         std::cout << "Enter_password: ";
14         std::cin >> password;
15         send(clientSocket, password.c_str(), strlen(password.c_str()) + 1, 0);
16
17         char registerResponse[256];
```

```

18         recv(clientSocket, registerResponse, sizeof(
19             registerResponse), 0);
20         std::cout << "Server_response:_" <<
21             registerResponse << std::endl;
22     } else if (choice == 2) {
23         // Login
24         std::string username, password;
25         std::cout << "Enter_username:_" <<
26             std::cin.ignore();
27         std::getline(std::cin, username);
28         send(clientSocket, username.c_str(), strlen(
29             username.c_str()) + 1, 0);
30         std::cout << "Enter_password:_" <<
31             std::getline(std::cin, password);
32
33         send(clientSocket, password.c_str(), strlen(
34             password.c_str()) + 1, 0);
35
36         char loginResponse[256];
37         recv(clientSocket, loginResponse, sizeof(
38             loginResponse), 0);
39         std::cout << "Server_response:_" <<
40             loginResponse << std::endl;

```

### 3.2.3 Giao tiếp với server

Người dùng gửi tin nhắn riêng với một người mình chọn (unicast) trong server.

```

1     if (choose == "1") {
2         std::cout << "Enter_the_username
3             _of_the_recipient:";
4         std::getline(cin,
5             recipientUsername);
6         send(clientSocket,
7             recipientUsername.c_str(),
8             strlen(recipientUsername.
9                 c_str()) + 1, 0);

```

```

5         std::cout << "Enter your message
           : ";
6         std::getline(cin, message);
7         message = "\nUser [" + username +
           "] send message to YOU: " +
           message + "\n";
8         send(clientSocket, message.c_str
              (), strlen(message.c_str()) +
              1, 0);
9     }

```

Người dùng gửi tin nhắn với tất cả mọi người trong server (broadcast).

```

1     else if (choose == "2") {
2         std::cout << "Enter your message
           : ";
3         std::getline(cin, message);
4         message = "\nUser [" + username
           + "] send message to ALL: " +
           message + "\n";
5         send(clientSocket, message.c_str
              (), strlen(message.c_str()) +
              1, 0);
6     }

```

Người dùng có thể kiểm tra người dùng đang online và đăng xuất.

```

1     if (choose == "3") {
2         message = username;
3         send(clientSocket, message.c_str
              (), strlen(message.c_str()) +
              1, 0);
4     }

```

Sử dụng luồng riêng (thread) để có thể nhận và hiển thị tin nhắn từ client khác.

```

1      std::thread receiveThread(receiveMessages ,
2      clientSocket);
      receiveThread.join();

```

## 4 Mô tả các chức năng use-case

### 4.1 Đăng Ký Người Dùng

Người dùng mở ứng dụng client và chọn chức năng đăng ký.

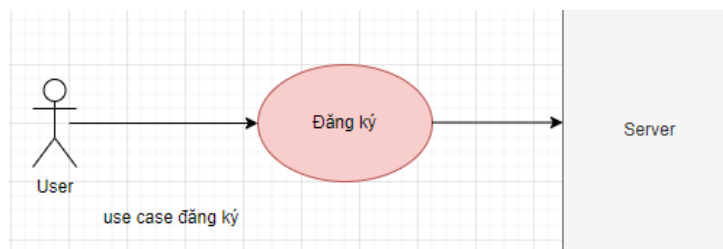
Nhập Thông Tin Đăng Ký: Người dùng nhập thông tin đăng ký bao gồm username và password.

Gửi Yêu Cầu Đăng Ký Đến Server: Client gửi yêu cầu đăng ký kèm theo thông tin người dùng đến server.

Xử Lý Yêu Cầu Ở Server: Server nhận yêu cầu, kiểm tra thông tin hợp lệ và thêm người dùng mới vào danh sách các người dùng và tên lẫn mật khẩu người dùng sẽ được lưu vào file users.txt.

Trả Kết Quả Đăng Ký: Server gửi thông báo đăng ký thành công hoặc thất bại về client.

Hiển Thị Kết Quả Cho Người Dùng: Client hiển thị thông báo đăng ký thành công hoặc thất bại cho người dùng.



Hình 1: use-case đăng ký

## 4.2 Đăng Nhập

Người dùng mở ứng dụng client và chọn chức năng đăng nhập.

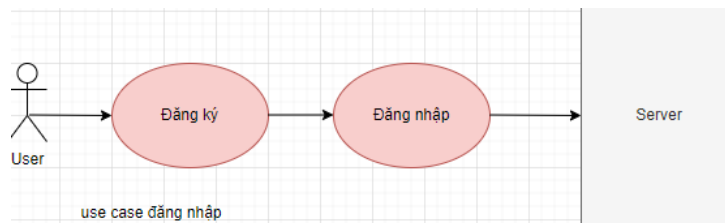
Nhập Thông Tin Đăng Nhập: Người dùng nhập thông tin đăng nhập bao gồm username và password.

Gửi Yêu Cầu Đăng Nhập Đến Server: Client gửi yêu cầu đăng nhập kèm theo thông tin người dùng đến server.

Xử Lý Yêu Cầu Ở Server: Server nhận yêu cầu, kiểm tra thông tin và đăng nhập người dùng nếu thông tin hợp lệ.

Trả Kết Quả Đăng Nhập: Server gửi thông báo đăng nhập thành công hoặc thất bại về client.

Hiển Thị Kết Quả Cho Người Dùng: Client hiển thị thông báo đăng nhập thành công hoặc thất bại và chuyển sang menu chính nếu thành công.



Hình 2: use-case đăng nhập

## 4.3 Gửi Tin Nhắn Riêng (Unicast)

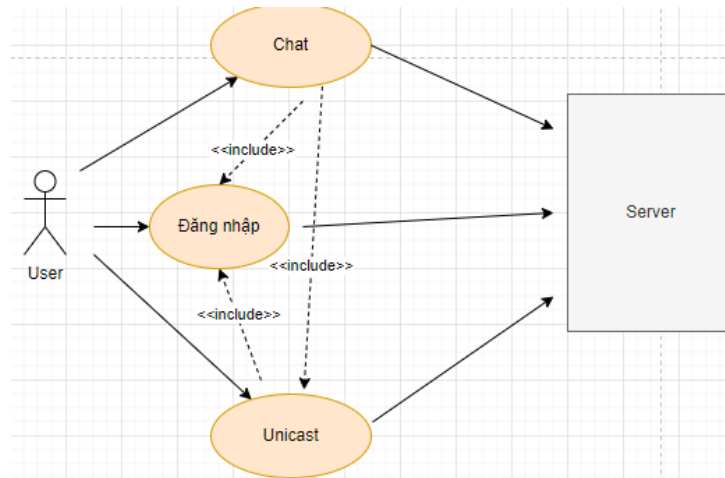
Người Dùng Chọn Gửi Tin Nhắn Riêng: Sau khi đăng nhập, người dùng chọn chức năng gửi tin nhắn riêng.

Nhập Người Nhận và Tin Nhắn: Người dùng nhập tên người nhận và nội dung tin nhắn. Nội dung tin nhắn sẽ được lưu vào file (message.txt)

Gửi Yêu Cầu Gửi Tin Nhắn Đến Server: Client gửi yêu cầu gửi tin nhắn riêng kèm theo thông tin đến server.

Xử Lý Yêu Cầu Ở Server: Server nhận yêu cầu, tìm người nhận và gửi tin nhắn đến người đó.

Hiển Thị Kết Quả Cho Người Dùng: Server trả kết quả về client, client hiển thị thông báo thành công hoặc thất bại.



Hình 3: use-case unicast

#### 4.4 Gửi Tin Nhắn Broadcast

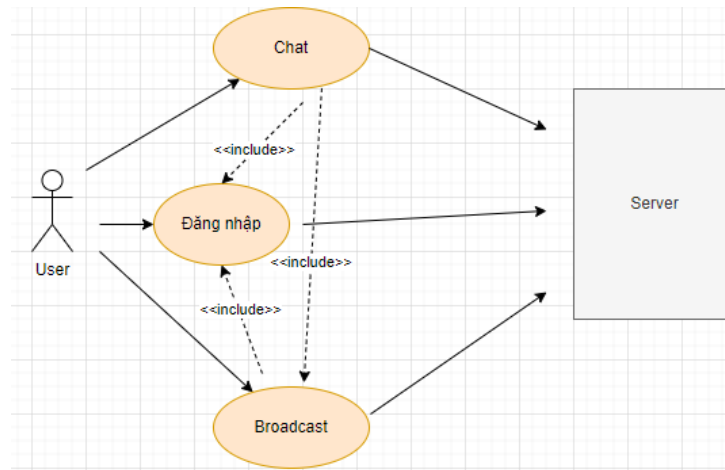
Người Dùng Chọn Gửi Tin Nhắn Broadcast: Sau khi đăng nhập, người dùng chọn chức năng gửi tin nhắn broadcast.

Nhập Nội Dung Tin Nhắn: Người dùng nhập nội dung tin nhắn broadcast. Tin nhắn sẽ được lưu vào file (message.txt)

Gửi Yêu Cầu Gửi Tin Nhắn Đến Server: Client gửi yêu cầu gửi tin nhắn broadcast kèm theo thông tin đến server.

Xử Lý Yêu Cầu Ở Server: Server nhận yêu cầu và gửi tin nhắn đến tất cả người dùng đang kết nối.

Hiển Thị Kết Quả Cho Người Dùng: Server trả kết quả về client, tất cả client khác sẽ nhận được tin nhắn nếu thành công.



Hình 4: use-case broadcast

## 4.5 Kiểm Tra Người Dùng Online

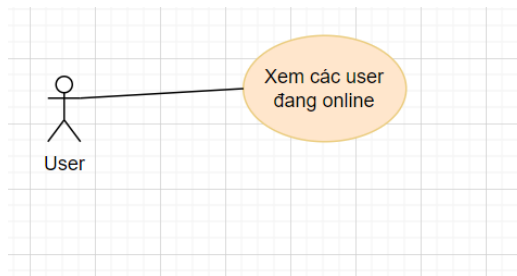
Người Dùng Chọn Kiểm Tra Người Dùng Online: Sau khi đăng nhập, người dùng chọn chức năng kiểm tra người dùng online.

Gửi Yêu Cầu Kiểm Tra Đến Server: Client gửi yêu cầu kiểm tra người dùng online kèm theo thông tin đến server.

Xử Lý Yêu Cầu Ở Server: Server nhận yêu cầu, kiểm tra người dùng đang online và gửi danh sách về client.

Hiển Thị Kết Quả Cho Người Dùng: Server trả danh sách người dùng online về client, client hiển thị danh sách này cho người dùng.





Hình 5: use-case kiểm tra người dùng online

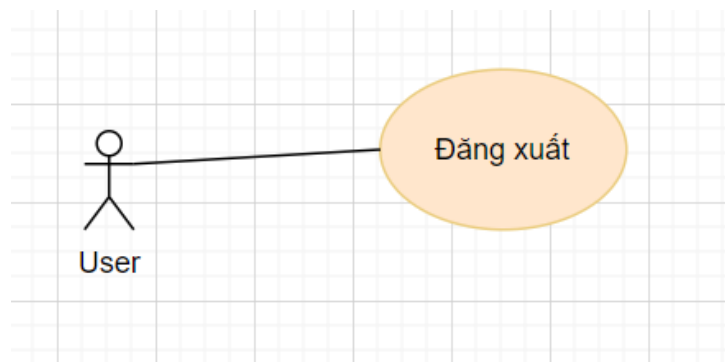
## 4.6 Đăng Xuất

Người Dùng Chọn Đăng Xuất: Người dùng chọn chức năng đăng xuất.

Gửi Yêu Cầu Đăng Xuất Đến Server: Client gửi yêu cầu đăng xuất kèm theo thông tin đến server.

Xử Lý Yêu Cầu Ở Server: Server nhận yêu cầu và đưa người dùng về trạng thái đăng nhập.

Hiển Thị Kết Quả Cho Người Dùng: Server trả thông báo đăng xuất về client, client hiển thị thông báo và chuyển về màn hình đăng nhập.



Hình 6: use-case đăng xuất

## 5 Thiết kế demo, test case

### 5.1 Đăng nhập

Thành công: Server trả về thành công

```
Connected to server
1. Register
2. Login
Choose option: 2
Enter username: tien
Enter password: 123
Server response: login_success:
```

Hình 7: Đăng nhập thành công

Thất bại: Server trả về thất bại

```
Connected to server
1. Register
2. Login
Choose option: 2
Enter username: gg
Enter password: 123
Server response: login_fail
```

Hình 8: Đăng nhập thất bại

## 5.2 Đăng ký

Thành công: Tên và mật khẩu người dùng được lưu vào file users.txt nếu đăng ký thành công.

```
1. Register
2. Login
Choose option: 1
Enter username: Nam
Enter password: 123
Server response: register_success
6    Nam 123
```

Hình 9: Đăng ký và tên, mật khẩu tự động lưu vào file

Thất bại: Trả về client thất bại nếu tên người dùng đã được đăng ký

```
1. Register
2. Login
Choose option: 1
Enter username: tung
Enter password: 123
Server response: register_fail,username has been used
```

Hình 10: Đăng ký thất bại

## 5.3 Broadcast

```
Enter username: tung
Enter password: 123
Server response: login_success:
|-----|
|Unicast enter 1--Broadcast enter 2--User are online enter 3--Logout enter 4|
|-----|

User [dat] send message to ALL: chao tung

User [gia] send message to ALL: Hi tung and khanh
```

```
Enter username: gia
Enter password: 123
Server response: login_success:
|-----|
|Unicast enter 1--Broadcast enter 2--User are online enter 3--Logout enter 4|
|-----|

2
Enter your message: Hi tung and khanh

User [gia] send message to ALL: Hi tung and khanh
```

```
Enter username: khanh
Enter password: 123
Server response: login_success:
|-----|
|Unicast enter 1--Broadcast enter 2--User are online enter 3--Logout enter 4|
|-----|

User [gia] send message to ALL: Hi tung and khanh
```

Hình 11: 2 người dùng còn lại sẽ nhận được tin nhắn người gửi(ở giữa)

## 5.4 Unicast

```
Choose option: 2
Enter username: khanh
Enter password: khanh1
Server response: login_success:
|-----|
|Unicast enter 1--Broadcast enter 2--User are online enter 3--Logout enter 4|
|-----|

1
Enter the username of the recipient:tung
Enter your message: chao tung nhe!
|-----|
|Unicast enter 1--Broadcast enter 2--User are online enter 3--Logout enter 4|
|-----|

■
```

---

```
Choose option: 2
Enter username: tung
Enter password: tung1
Server response: login_success:
|-----|
|Unicast enter 1--Broadcast enter 2--User are online enter 3--Logout enter 4|
|-----|

User [khanh] send message to YOU: chao tung nhe!

□
```

---

```
Choose option: 2
Enter username: gia
Enter password: gia1
Server response: login_success:
|-----|
|Unicast enter 1--Broadcast enter 2--User are online enter 3--Logout enter 4|
|-----|

□
```

Hình 12: user khanh (ở đầu) gửi tin nhắn tới user tung và chỉ một user tung nhận được tin nhắn

## 5.5 Hiện thị người dùng online

```
Choose option: 2
Enter username: gia
Enter password: gia1
Server response: login_success:
|-----|
|Unicast enter 1--Broadcast enter 2--User are online enter 3--LogOut enter 4|
|-----|
3
|-----|
|Unicast enter 1--Broadcast enter 2--User are online enter 3--LogOut enter 4|
|-----|
Users are online: [khanh]      [tung]
█
```

Hình 13: user gia yêu cầu hiển thị người dùng đang hoạt động server sẽ trả về tung và khanh

## 5.6 Đăng xuất

```
Choose option: 2
Enter username: tung
Enter password: tung1
Server response: login_success:
|-----|
|Unicast enter 1--Broadcast enter 2--User are online enter 3--LogOut enter 4|
|-----|

User [khanh] send message to YOU: chao tung nhe!

4
You was logged out
1. Register
2. Login
Choose option: █
```

Hình 14: user khanh đăng xuất, sau khi đăng xuất sẽ trả về màn hình để ta có thể tiếp tục đăng ký và đăng nhập