

ĐẠI HỌC BÁCH KHOA HÀ NỘI

BÁO CÁO BÀI TẬP LỚN MÔN HỌC LẬP TRÌNH MẠNG

Xây dựng ứng dụng chat

NGUYỄN DUY KHÁNH 20204992

khanh.nd204992@sis.hust.edu.vn

LƯU KHẮC ĐĂNG DƯƠNG 20205076

duong.lkd205076@sis.hust.edu.vn

Giảng viên hướng dẫn: ThS. ThS. Bùi Trọng Tùng _____

Chữ kí GVHD

Trường: Công nghệ Thông tin và Truyền thông

HÀ NỘI, 01/2024

GIỚI THIỆU VỀ BÀI TẬP LỚN

Bài tập lớn nằm trong nội dung môn học Thực hành Lập trình mạng máy tính (HEDSPI) - IT4062.

Mục tiêu của bài tập lớn là xây dựng được một ứng dụng chat triển khai được dựa vào các kiến thức đã được cung cấp trong môn học từ đó giúp sinh viên áp dụng tốt kiến thức về Lập trình mạng, có cơ hội áp dụng kiến thức đã học trong công việc thực tế.

Ứng dụng chat thiết kế cần đảm bảo theo các nội dung chức năng:

- Khi 1 user đăng nhập thành công, cung cấp danh sách các bạn bè khác đang online
- User gửi và nhận thông điệp
- Khi có 1 trong 2 user ngắt kết nối hoặc yêu cầu dừng cuộc trò chuyện, thông báo cho bên còn lại biết.
- Tạo nhóm chat:
 - User tạo nhóm chat
 - User gửi yêu cầu mời tham gia nhóm tới các user khác
 - Tham gia và rời nhóm chat
 - Gửi thông điệp tới các nhóm"

Sinh viên thực hiện
(Ký và ghi rõ họ tên)

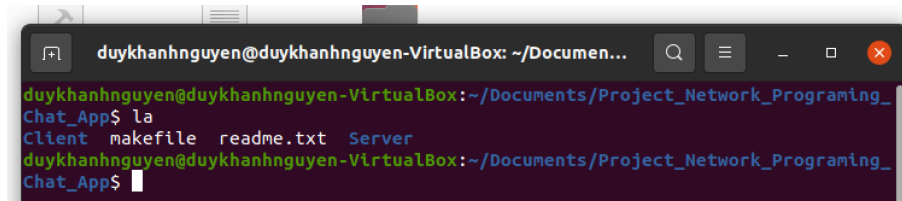
MỤC LỤC

CHƯƠNG 1. HƯỚNG DẪN CHẠY ỨNG DỤNG TRÊN MÁY ẢO VDI...	1
CHƯƠNG 2. MÔ TẢ GIAO THỨC	2
2.1 Mã giao thức của ứng dụng.....	2
2.1.1 Mô tả quy tắc đặt mã giao thức.....	2
2.1.2 Các mã giao thức được quy định trong ứng dụng.....	2
2.2 Kịch bản theo giao thức.....	5
2.2.1 Xử lý truyền dòng	5
2.2.2 Cài đặt cơ chế vào ra trên Server.....	5
2.2.3 Đăng ký và quản lý tài khoản	5
2.2.4 Đăng nhập và quản lý phiên	5
2.2.5 Gửi lời mời kết bạn.....	6
2.2.6 Chấp nhận, từ chối lời mời kết bạn.....	7
2.2.7 Huỷ kết bạn	7
2.2.8 Lấy danh sách bạn bè và trạng thái.....	7
2.2.9 Gửi tin nhắn giữa hai người dùng.....	8
2.2.10 Ngắt kết nối	9
2.2.11 Thêm người dùng vào nhóm chat	9
2.2.12 Xóa người dùng ra khỏi nhóm	10
2.2.13 Gửi nhận thông điệp trong nhóm chat (Chat nhóm)	10
2.2.14 Log hoạt động.....	11
CHƯƠNG 3. CẤU TRÚC PHẦN MỀM CỦA CHƯƠNG TRÌNH.....	12
3.0.1 Cấu trúc phía Client	12
3.0.2 Cấu trúc phía Server.....	12

CHƯƠNG 1. HƯỚNG DẪN CHẠY ỨNG DỤNG TRÊN MÁY ẢO VDI

Các bước hướng dẫn cài đặt chương trình và khởi chạy chương trình

- **Bước 1:** Truy cập vào đường dẫn: /Document/Project_Network_Programing_Chat_App



Hình 1.1: Vị trí thư mục Project

- **Bước 2:** Build ứng dụng bằng lệnh make.

Sau khi biên dịch. Sẽ có hai tệp được tạo ra: server_chat (Có đường dẫn /Server/src/server_chat) và client_chat (Có đường dẫn /Client/src/client_chat)

- **Bước 3:** Khởi chạy ứng dụng. Mở 2 Terminal tại hai vị trí file vừa được build và chạy các lệnh theo cú pháp:
 - ./server_chat <Port_number>
 - ./client_chat <Server_IP_Address> <Server_Port_Number>
 - Trường hợp sử dụng nhiều client thì mở nhiều Terminal và chạy như hướng dẫn chạy client ở trên.

CHƯƠNG 2. MÔ TẢ GIAO THỨC

2.1 Mã giao thức của ứng dụng

2.1.1 Mô tả quy tắc đặt mã giao thức

Trong ứng dụng chat, chúng em sử dụng mã giao thức dựa vào thao khảo mã giao thức của FTP(File Transfer Protocol) và có biến đổi đi một chút.

Cụ thể, mã giao thức có 4 ký tự số (xxxx) bao gồm 3 phần chính:

- **Phần thứ nhất** (1 chữ số đầu tiên của bên trái) Sự tích cực hoặc không tích cực của nội dung Server muốn trả lời. Hay có thể hiểu là lỗi hoặc thành công.

Ở đây trong quá trình làm mới chỉ sử dụng hai mã là:

- "2": Phản hồi xử lý thành công (tích cực)
- "4": Phản hồi xử lý không thành công (phủ quyết)

Tuy nhiên, nếu tương ứng được phát triển hơn nữa thì có thể dùng thêm cách mã khác 2 và 4 để phân chia rõ hơn các mức độ tích cực và phủ quyết hoặc ý nghĩa khác (như FTP).

- **Phần thứ hai** (1 chữ số tiếp theo): Lý do trả về của thông điệp. Lý do trả về của thông điệp sẽ mang ý nghĩa dựa vào bảng mã giao thức. Được quy định trong quá trình Server xử lý, nếu cùng mang một ý nghĩa là tích cực hoặc phủ quyết thì phần này giúp cho Client hiểu được tại sao Server lại trả về mã giao thức đó.

Phần này hiệu quả đối với một số chức năng có đồng thời nhiều mã mang ý nghĩa (tích cực/ phủ quyết) nhưng lý do xử lý trên Server lại khác nhau.

Ví dụ: trong quá trình gửi lời mời kết bạn, Server trả về lỗi không thành công có hai trường hợp là: 4008 và 4108. Trong đó 4008: Không tìm thấy người dùng, 4108: Đã kết bạn từ trước đó.

- **Phần thứ 3** (2 chữ số còn lại): Mã theo chức năng, ban đầu chúng em dự tính phần này chỉ có 1 chữ số nhưng do trong quá trình thực hiện số các chức năng cần phân biệt lớn hơn 10 nên chúng em đã dành ra 2 chữ số để biểu thị nội dung chức năng này. Các mã giao thức của cùng một chức năng thì sẽ có phần này giống nhau.

2.1.2 Các mã giao thức được quy định trong ứng dụng

Các mã giao thức trả lời từ Server

CHƯƠNG 2. MÔ TẢ GIAO THỨC

Mã giao thức	Ý nghĩa	Lý do
2000	Kết nối tới Server thành công	
2001	Phản hồi đăng ký thành công	
4001	Phản hồi đăng ký không thành công	
2002	Phản hồi đăng nhập thành công	
4102	Phản hồi đăng nhập không thành công	Không tìm thấy tên đăng nhập
4202	Phản hồi đăng nhập không thành công	Sai mật khẩu
4302	Phản hồi đăng nhập không thành công	Tài khoản đã được đăng nhập
2003	Gửi tin nhắn thành công (Giữa hai người)	
4103	Gửi tin nhắn không thành công (Giữa hai người)	
2004	Tạo nhóm thành công	
4004	Tạo nhóm không thành công	
2005	Thêm thành viên vào nhóm thành công	
4105	Thêm thành viên vào nhóm không thành công	Không có user_id trong friend_list
4205	Thêm thành viên vào nhóm không thành công	Không tồn tại Group id
4305	Thêm thành viên vào nhóm không thành công	Đã tồn tại trong nhóm
4405	Thêm thành viên vào nhóm không thành công	userid không hợp lệ
4505	Thêm thành viên vào nhóm không thành công	Chưa đăng nhập
2006	Xóa người dùng ra khỏi nhóm thành công	
4106	Xóa người dùng ra khỏi nhóm không thành công	Không tồn tại userid
4206	Xóa người dùng ra khỏi nhóm không thành công	Không tồn tại groupid
4306	Xóa người dùng ra khỏi nhóm không thành công	Lỗi mở file trên Server
4406	Xóa người dùng ra khỏi nhóm không thành công	Không có user trong nhóm
4506	Xóa người dùng ra khỏi nhóm không thành công	Bạn chưa tham gia vào nhóm chat này! (no role)
4606	Xóa người dùng ra khỏi nhóm không thành công	Chưa đăng nhập
2007	Gửi tin nhắn nhóm thành công	
4007	Gửi tin nhắn nhóm không thành công	Không tồn tại groupid
4107	Gửi tin nhắn nhóm không thành công	Chưa tham gia vào nhóm
4207	Gửi tin nhắn nhóm không thành công	Chưa đăng nhập
2008	Gửi lời mời kết bạn thành công	
4008	Gửi lời mời kết bạn không thành công	userid không hợp lệ. không online
4108	Gửi lời mời không thành công	Đã có kết bạn từ trước
2108	Kết bạn thành công, gửi tới cả hai userid	
2208	Kết bạn không thành công, gửi tới cả hai userid	
2009	Hủy kết bạn thành công	
4009	Hủy kết bạn không thành công	Không tìm thấy userid
2010	Rời khỏi nhóm thành công	
4010	Rời khỏi nhóm không thành công	Không tồn tại groupid
4110	Rời khỏi nhóm không thành công	Chưa là thành viên của nhóm
2011	Lấy danh sách nhóm thành công	Danh sách không trống
2111	Lấy danh sách nhóm thành công	Danh sách trống
4011	Lấy danh sách nhóm không thành công	
2012	Lấy danh sách bạn bè thành công	Danh sách không trống
2112	Lấy danh sách bạn bè thành công	Danh sách trống
4012	Lấy danh sách bạn bè không thành công	Lỗi trên server (mở file)
4112	Lấy danh sách bạn bè không thành công	Chưa đăng nhập
2013	Lấy danh sách thành viên nhóm thành công	Danh sách không trống
2113	Lấy danh sách thành viên nhóm thành công	Danh sách trống
4013	Lấy danh sách thành viên nhóm không thành công	Không tồn tại groupid
2014	Lấy nội dung tin nhắn giữa hai người thành công	Danh sách không trống
2114	Lấy nội dung tin nhắn giữa hai người thành công	Danh sách trống
4014	Lấy nội dung tin nhắn giữa hai người không thành công	Không tìm thấy Tập data chứa tin nhắn
2015	Ngắt kết nối thành công	
4015	Ngắt kết nối không thành công	Không tìm thấy userid
2016	Lấy nội dung tin nhắn nhóm thành công	
4016	Lấy nội dung tin nhắn nhóm không thành công	Không tìm thấy groupid
4116	Lấy nội dung tin nhắn nhóm không thành công	Chưa tham gia vào nhóm
4216	Lấy nội dung tin nhắn nhóm không thành công	Chưa đăng nhập

Bảng 2.1: Mã giao thức của Ứng dụng Chat

2.2 Kịch bản theo giao thức

2.2.1 Xử lý truyền dòng

Xử lý truyền dòng: Xử lý cả client và Server: Thêm ký tự “\r \n” vào cuối phần dữ liệu từ Client gửi lên Server.

Từ Server gửi về Client đối với một số dữ liệu có kích thước lớn, xử lý truyền dòng bằng cách gửi kích thước thông điệp trước và gửi thông điệp sau (Phía từ Server → Client). Tuy nhiên trong ứng dụng chưa xử lý truyền dòng theo hướng này do kích thước các thông điệp chat trả về chưa lớn, nhưng khi ứng dụng phục vụ nhiều người và với lượng dữ liệu lớn thì cần xử lý truyền dòng theo chiều này hoặc sử dụng lưu tạm vào bộ nhớ Client để tránh việc tra đổi quá nhiều dữ liệu từ Server trả về Client.

2.2.2 Cài đặt cơ chế vào ra trên Server

Sử dụng TCP Socket

Mỗi client kết nối tới Server thì Server sẽ tạo một luồng để xử lý với Client đó tới khi Client ngắt kết nối thì Server đóng luồng dành cho Client.

Sử dụng cơ chế đa luồng (Multi Thread)

Nên cần phải quản lý đồng bộ phải tốt giữa các tài nguyên chung giữa các luồng

2.2.3 Đăng ký và quản lý tài khoản

Kịch bản:

- Khi người dùng lựa chọn chức năng đăng ký thì sẽ hiển thị các thành phần để người dùng nhập gồm: Tên người dùng, Tài khoản và Mật khẩu
- Client sẽ gửi thông tin đến cho server.
- Server sẽ bắt đầu kiểm tra trong file account.txt. Nếu tài khoản đã tồn tại ở trong cơ sở dữ liệu thì sẽ gửi lại response “4001” cho client rằng đăng ký không thành công
- Nếu tài khoản không có trong cơ sở dữ liệu thì sẽ gửi response “2001” đến cho client thông báo đăng ký thành công và lưu account vào trong cơ sở dữ liệu.

2.2.4 Đăng nhập và quản lý phiên

Đăng nhập:

Tài nguyên: Cơ sở dữ liệu: account.txt

Mỗi dòng sẽ lưu trữ thông tin một tài khoản

account.txt


```
<fullname(Không có dấu cách, nên viết hoa với các chữ cái đầu)
(vô hạn)> <username(vô hạn)> <password(vô hạn)> <User_id -
gồm 5 số> <Status: 0/1 band/activite, good>
```

```
NguyenDuyKhanh khanhnd 123456 00657 1
```

```
LuuKhacDangDuong duonglk 1234aa 00658 0
```

```
.
.
.
```

Quản lý phiên: Phiên được cài đặt dưới dạng Link List, mỗi node có các thuộc tính:

```
// Cấu trúc lưu trữ thông tin của mỗi phiên
```

```
struct Session
{
    int socketId; // Định danh socket trên server
    char *userId;
    struct Session *next; // Con trỏ đến phiên
    tiếp theo trong danh sách liên kết
};
```

Phiên chứa các người đang hoạt động, hay các tài khoản đang kết nối tới Server hay như ứng với trên Facebook có nghĩa là Đang hoạt động

Kịch bản:

- Người dùng chọn chức năng đăng nhập
- Người dùng nhập tên người dùng, mật khẩu
- Client gửi thông điệp lên Server: “login\n<username>\n<pass>\r\n”
- Server xử lý và trả về mã giao thức

2.2.5 Gửi lời mời kết bạn

Kịch bản:

- Người dùng chọn chức năng gửi lời mời kết bạn Khi đó client sẽ gửi response cho server “add_friend\n<user_id_recieved>\r\n” và chờ phản hồi
- Server sẽ kiểm tra đầu vào và gửi response “2008” cho người gửi và người nhận nếu như gửi lời mời thành công.
- Với trường hợp server kiểm tra thấy rằng 2 người đã là bạn bè (đã có trong file friend.txt) thì sẽ gửi response “4108” cho người gửi

- Nếu không tìm thấy id người nhận hoặc người nhận không online thì sẽ thông báo response “4008”.

2.2.6 Chấp nhận, từ chối lời mời kết bạn

Kịch bản:

- Người dùng chọn chức năng thông báo lời mời kết bạn
- Khi đó client sẽ gửi response
- Hiện thị ra danh sách lời mời kết bạn
- Nếu chấp nhận lời mời kết bạn thì nhấn thông điệp User_ID 1
- Nếu từ chối lời mời kết bạn thì nhấn thông điệp User_ID 0
- Khi đó client sẽ gửi thông điệp đến cho server với “accept

2.2.7 Huỷ kết bạn

- Người dùng chọn chức năng huỷ kết bạn.
- Client sẽ gọi hàm getFriendsListAndStatus để lấy ra danh sách của bạn bè
- Sau người dùng sẽ nhập id của người cần huỷ kết bạn
- Client gửi thông điệp cho server có dạng “un_friend\n<id1>\n<id2>\n\r”
- Server nhận thông điệp.
- Kiểm tra xem 2 người có là bạn bè hay không (mở file firriend.txt và kiểm tra từng dòng xem có nội dung chứa “id1 id2” hay không.
- Nếu server tìm thấy nội dung thì sẽ gửi response “2009” và xoá nội dung “user_id1 user_id2” đi
- Nếu không tìm thấy thì sẽ gửi response “4009” là không tìm thấy ID hoặc chưa kết bạn

2.2.8 Lấy danh sách bạn bè và trạng thái

Kịch bản:

- Người dùng chọn chức năng/ Hoặc chức năng này được Client gọi để phục vụ cho các chức năng khác.
- Client gửi tới Server thông điệp: “get_friend_list\r\n” và chờ đợi hồi âm từ Server
- Server trả về thông điệp mã giao thức
- Nếu mã giao thức là thành công và báo danh sách không trống. Sau khi nhận mã giao thức từ Server, Client gửi "OK\r\n tới Server để biểu thị sẵn sàng nhận

danh sách bạn bè kèm trạng thái. Tiếp theo: Server gửi về danh sách bạn bè kèm trạng thái có dạng: “\n<userid1>\n<usernameid1>\n<status: online - 1/ offline: 0>...”

2.2.9 Gửi tin nhắn giữa hai người dùng

Tin nhắn được quản lý dữ liệu bằng file có tên id1_id2.txt <userid> <hh:mm:ss-dd:mm:yyyy> <content có thể chứa dấu cách> Khanh 04:09:23-09:01:2024 Hello World

Kịch bản:

- Người dùng chọn chức năng
- Client gửi thông điệp tới Server: “get_friend_list\n<userid>\r\n” và chờ đợi hồi âm từ Server
- Server trả về thông điệp mã giao thức
- Nếu mã giao thức là thành công và báo danh sách nhóm không trống, Server gửi tiếp Danh sách bạn bè kèm trạng thái có dạng: “<numberoffriend>\n<userid1>\n<usernameid1>\n<status: online - 1/ offline: 0>\n. . . \r\n” Người dùng nhập id cần nhắn tin
- Client gửi lên Server thông điệp: “send_message_online_private\nUSER_ID1\nUSER_ID2\n\r” và chờ đợi hồi âm từ Server
- Nếu mã giao thức là thành công và báo danh sách nhóm không trống, Server gửi tiếp nội dung tin nhắn kèm trạng thái có dạng: “<userid1>\n<hh:mm:ss-dd:mm:yyyy>\n<content có thể chứa dấu cách>\n<userid2>\n<hh:mm:ss-dd:mm:yyyy>\n<content có thể chứa dấu cách>\n. . . \r\n”
- Client hiển thị nội dung tin nhắn trước đó ra terminal
- Người dùng nhập nội dung tin nhắn và nhấn enter tới khi xâu nhập vào là “!q” nghĩa là thoát khỏi chức năng chat
- Mỗi lần người dùng nhắn tin nhắn mới. Client gửi lên Server: “send_message_online_private\nUSER_ID1\nUSER_ID2\n<content Có thể có dấu cách>”
- Nhận về mã giao thức từ Server và hiển thị gửi thành công hoặc không.
- **Chức năng này đã xử lý luôn việc người dùng còn lại không online nhưng vẫn có thể xem được tin nhắn ngay khi đăng nhập trở lại."Gửi tin nhắn offline"**

2.2.10 Ngắt kết nối

Kịch bản:

- Người dùng chọn chức năng
- Client gửi thông điệp tới Server
- Client hiển thị danh sách bạn bè
- Client gửi thông điệp đến cho server “disconnect\n<userid2>\n\r”
- Server nhận thông điệp và đọc message để lấy userid1 và userid2
- Tìm kiếm trong folder “..\message_private” và tìm kiếm user_id1_userid2.txt
- Nếu tìm thấy thì xoá file đó đi và gửi response “2015”
- Nếu không tìm thấy thì gửi response “4015”

2.2.11 Thêm người dùng vào nhóm chat

Nhóm chat được quản lý dữ liệu bằng file group.txt:

group.txt

<idgroup> <iduser1> <iduser2> ...

Các idgroup và iduser đều được tạo bởi hàm genId

Kịch bản:

- Người dùng chọn chức năng
- Client gửi thông điệp tới Server: “get_list_group\n<userid>\r\n” và chờ đợi hồi âm từ Server
- Server trả về thông điệp mã giao thức
- Nếu mã giao thức là thành công và báo danh sách nhóm không trống, Server gửi tiếp Danh sách bạn bè kèm trạng thái có dạng:
“\n<groupid1>\n<groupid2>”
- Client gửi tới Server thông điệp: “get_friend_list\r\n” và chờ đợi hồi âm từ Server
- Server trả về thông điệp mã giao thức
- Nếu mã giao thức là thành công và báo danh sách không trống, Server gửi tiếp. Danh sách bạn bè kèm trạng thái có dạng: “\n<userid1>\n<usernameid1>\n<status: online - 1/ offline: 0>”
- Người dùng nhập idgroup và userid muốn thêm vào
- Client gửi lên Server thông điệp: ”add_user_to_group\n<groupid>\n<userid>\r\n”

Server xử lý mà trả lời về client mã giao thức

2.2.12 Xóa người dùng ra khỏi nhóm

Kịch bản:

- Người dùng chọn chức năng
- Client gửi thông điệp tới Server: “get_list_group\n<userid>\r\n” và chờ đợi hồi âm từ Server
- Server trả về thông điệp mã giao thức
- Nếu mã giao thức là thành công và báo danh sách nhóm không trống, Server gửi tiếp Danh sách bạn bè kèm trạng thái có dạng: “<numberofgroup>\n<groupid1>\n<groupid2>\n... \r\n”
- Người dùng nhập groupid
- Client gửi lên Server thông điệp: “get_list_member_of_group\n<groupid>\r\n” và chờ đợi hồi âm từ Server
- Server trả về thông điệp mã giao thức
- Nếu mã giao thức là thành công và báo danh sách không trống, Server gửi tiếp
- Danh sách thành viên có dạng: “<numberofmember>\n<userid1>\n<userid2>\n... \r\n”
- Người dùng nhập userid muốn xóa khỏi nhóm chat
- Client gửi thông điệp lên Server: “remove_user_to_group\n<groupid>\n<userid>\r\n”
- Server xử lý và gửi lại client mã giao thức

2.2.13 Gửi nhận thông điệp trong nhóm chat (Chat nhóm)

Tin nhắn nhóm được quản lý dữ liệu bằng file có tên group_<idgroup>.txt

group_<idgroup>.txt

<userid> <hh:mm:ss-dd:mm:yyyy> <content có thể chứa dấu cách>
khanhnd 04:09:23-09:01:2024 Hello World

Kịch bản:

- Người dùng chọn chức năng
- Client gửi thông điệp tới Server: “get_list_group\r\n” và chờ đợi hồi âm từ Server
- Server trả về thông điệp mã giao thức Nếu mã giao thức là thành công và báo danh sách nhóm không trống, Server gửi tiếp Danh sách bạn bè kèm trạng thái có dạng: “\n<groupid1>\n<groupid2>\n...” Người dùng nhập groupid

- Client gửi lên Server thông điệp: “get_content_message_of_group\n<groupid>\r\n” và chờ đợi hồi âm từ Server
- Nếu mã giao thức là thành công và báo danh sách nhóm không trống. Client gửi lên Server mã báo sẵn sàng nhận dữ liệu tin nhắn nhóm "OK\r\n". Tiếp theo, Server gửi về dữ liệu tin nhắn nhóm có dạng: “<userid1>\n<hh:mm:ss-dd:mm:yyyy>\n<content có thể chứa dấu cách>\n<userid2>\n<hh:mm:ss-dd:mm:yyyy>\n<content có thể chứa dấu cách> \n...”
- Hiển thị nội dung tin nhắn trước đó ra terminal
- Người dùng nhập nội dung tin nhắn và nhấn enter tới khi xâu nhập vào là “!q” nghĩa là thoát khỏi chức năng chat
- Mỗi lần người dùng nhấn tin nhắn mới. Client gửi lên Server: “send_message_group\n<groupid>\n<userid>\n<content Có thể có dấu cách>”
- Nhận về mã giao thức từ Server và hiển thị gửi thành công hoặc không.

2.2.14 Log hoạt động

Log hoạt động được quản lý dữ liệu bằng file: log.txt

log.txt

[dd/mm/yyyy hh:mm:ss]\$<clientaddress>\$<userid :

trống nếu chưa tìm được userid - chưa đăng nhập>\$<+/-><result>

CHƯƠNG 3. CẤU TRÚC PHẦN MỀM CỦA CHƯƠNG TRÌNH

Ứng dụng có mã nguồn phân chia thành hai phía Client và Server

3.0.1 Cấu trúc phía Client

- Thư mục lib: Chứa các file định nghĩa của các hàm tự định nghĩa
- Thư mục src: Chứa định nghĩa các hàm trong Client

3.0.2 Cấu trúc phía Server

- Thư mục database:
 - Thư mục message_group: Chứa dữ liệu tin nhắn của các nhóm
 - Thư mục message_private: Chứa dữ liệu tin nhắn giữa hai người
 - Các tệp dữ liệu: danh sách tài khoản, danh sách bạn bè, danh sách nhóm, Mã số GenID
- Thư mục lib: Chứa các khai báo của các hàm sử dụng xử lý chức năng của Server
- Thư mục src: Chứa các file định nghĩa của các hàm sử dụng xử lý chức năng của Server