

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN MẠNG MÁY TÍNH
ĐỀ 2

Thông tin nhóm

18127004 – Nguyễn Vũ Thu Hiền

18127027- Trần Minh Đức

18127208 - Ngô Thanh Phương Thái

Thành phố Hồ Chí Minh, 2020

MỤC LỤC

Contents

MỤC LỤC	2
CHƯƠNG I: THÔNG TIN CHUNG	3
Tên đề tài:	3
Môi trường phát triển ứng dụng:	3
Thông tin về nhóm:	3
CHƯƠNG II: PHÁT BIỂU BÀI TOÁN	4
1. Khảo sát hiện trạng	4
1.1. Nhu cầu thực tế	4
1.2. Các phần mềm cùng loại hiện có	4
1.3. Hạn chế tồn tại của các phần mềm trên.....	4
2. Yêu cầu của hệ thống	4
CHƯƠNG III. MÔ HÌNH USE CASE	5
1. Sơ đồ Use-case	5
2. Danh sách các Actor	5
3. Danh sách các Use-case	5
4. Đặc tả Use-case.....	6
4.1. Đặc tả Use-case “Cấu hình địa chỉ Server”	6
4.2. Đặc tả Use-case “Xem hội thoại đã chat”	7
4.3. Đặc tả Use-case “Chat nhóm”	7
4.4. Đặc tả Use-case “Gửi File”	8
4.5. Đặc tả Use-case “Xem danh sách online”	9
4.6. Đặc tả Use-case “Xem log”	10
4.7. Đặc tả Use-case “Cấu hình trên Server”	10
4.8. Đặc tả Use-case “Chat riêng tư”	11
CHƯƠNG IV. THIẾT KẾ DỮ LIỆU	13
CHƯƠNG V. HƯỚNG DẪN SỬ DỤNG	20
CHƯƠNG VI: ĐÁNH GIÁ	27
CHƯƠNG V: Tham khảo	28

CHƯƠNG I: THÔNG TIN CHUNG

Tên đề tài:

Chương trình mô phỏng ứng dụng chat conference gồm một server và nhiều client.

Môi trường phát triển ứng dụng:

- IDE: Visual Studio 2019 Community (Free Version)
- Hệ điều hành: Window 10
- Nền tảng: Win32 x86
- Framework: MFC

Thông tin về nhóm:

STT	MSSV	Họ và tên	Điện thoại	Email
1	18127004	Nguyễn Vũ Thu Hiền		
2	18127027	Trần Minh Đức		
3	18127208	Ngô Thanh Phương Thái		

CHƯƠNG II: PHÁT BIỂU BÀI TOÁN

1. Khảo sát hiện trạng

1.1. Nhu cầu thực tế

Qua việc khảo sát thu thập nhu cầu liên quan đến việc giao tiếp qua kênh chat online, nhóm đã đưa ra một số yêu cầu thực tế sau:

- Người sử dụng chat có nhu cầu sử dụng kênh giao tiếp trực tuyến để liên lạc với nhau qua các dạng nội dung text, hình ảnh, tệp tin,...
- Ngoài liên lạc trực tiếp 1 – 1 thì người dùng còn muốn sử dụng để họp chung với nhiều người dùng khác để làm việc nhóm.
- Do lý do đặc thù về tính bảo mật của doanh nghiệp hay cá nhân, người sử dụng muốn xây dựng server chat riêng để sử dụng
- Ngoài ra người dùng còn muốn tăng tính bảo mật bằng cách tin nhắn sẽ tự hủy trong một khoảng thời gian nhất định trong một số trường hợp nhất định

1.2. Các phần mềm cùng loại hiện có

- Messenger
- Zalo
- ...

1.3. Hạn chế tồn tại của các phần mềm trên

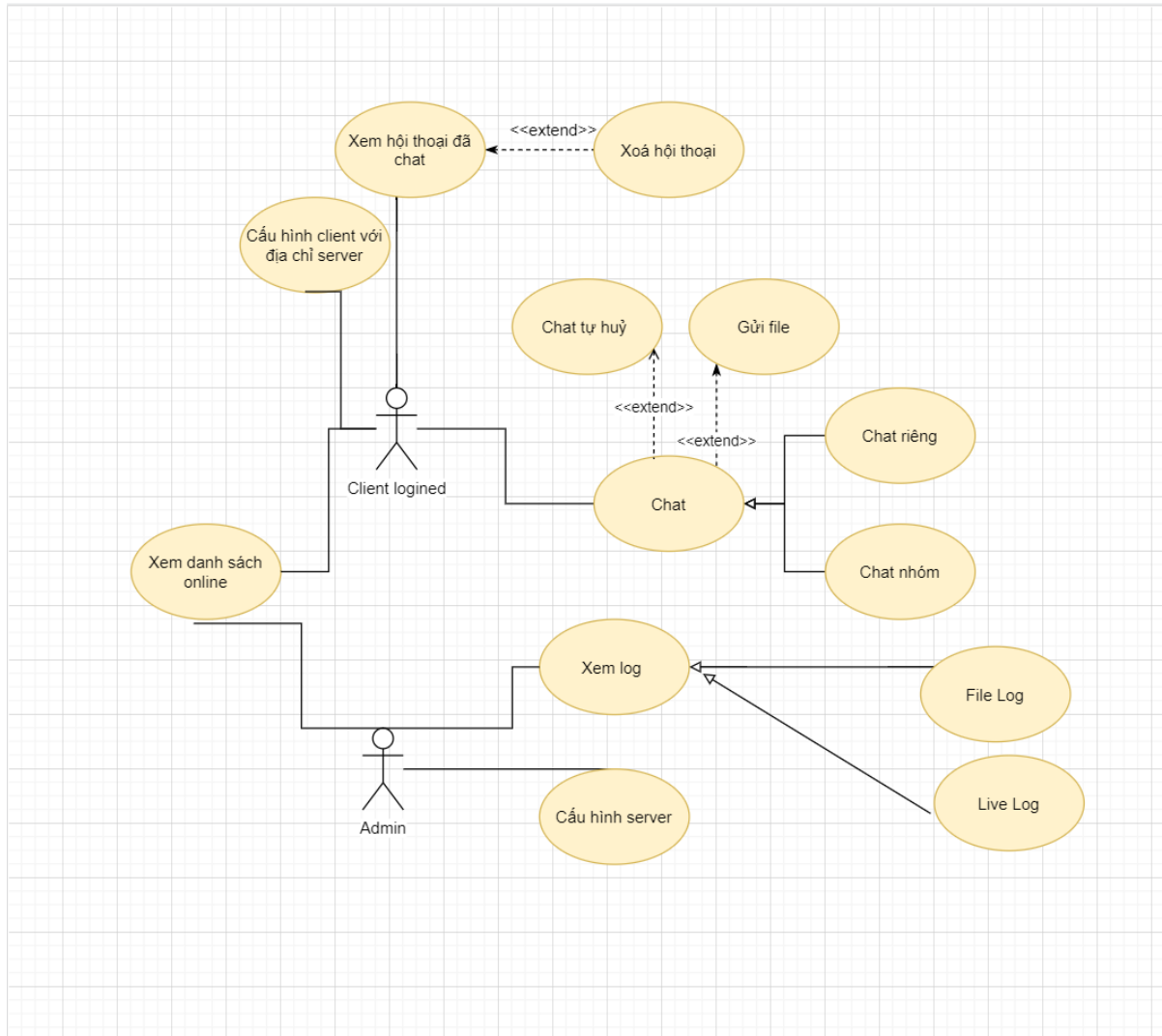
- Các phần mềm trên dùng trong một hệ sinh thái của nhà phát hành, vì vậy sẽ không phù hợp với mục tiêu bảo mật đề ra.
- Các phần mềm do sử dụng phổ cập với lượng lớn người sử dụng nên sẽ giới hạn trong việc truyền file dung lượng lớn, chặn các tệp tin thực thi đặc biệt...
- Không thể triển khai nội bộ trong cùng 1 nhóm hoặc cùng 1 công ty

2. Yêu cầu của hệ thống

- **Yêu cầu cấu hình**
 - CPU: Bộ xử lý 1.2 ghz hoặc hơn
 - RAM: dung lượng 1GB trở lên
 - HDD/SDD: dung lượng trống 20MB trở lên
 - Hệ điều hành: Window 7 trở lên (ưu tiên 64 bits)

CHƯƠNG III. MÔ HÌNH USE CASE

1. Sơ đồ Use-case



2. Danh sách các Actor

STT	Tên Actor	Ý nghĩa/Ghi chú
1	Người sử dụng Client	Người sử dụng tính năng trong client
2	Người quản trị Admin	Quản lý hoạt động trên server

3. Danh sách các Use-case

STT	Tên Use-case	Ý nghĩa/Ghi chú
1	Cấu hình client với địa chỉ server	Cho phép người dùng cấu hình các cài đặt kết nối với server
2	Xem hội thoại đã chat	Tải các nội dung hội thoại đã chat từ người sử dụng

[Type here]

[Type here]

[Type here]

3	Xoá hội thoại	Xoá dữ liệu hội thoại đã lưu
4	Chat riêng	Cho phép người sử dụng tạo cuộc hội thoại riêng với 1 client khác
5	Chat nhóm	Cho phép người sử dụng tạo hội thoại nhóm với nhiều client khác
6	Xem danh sách online	Xem các danh sách các client đã đăng nhập có online trên hệ thống
7	Chat tự hủy	Hệ thống sẽ tự xoá các dữ liệu đã gửi sau thời gian nhất định
8	Gửi file	Cho phép người sử dụng có thể gửi file cho client khác thông qua server
9	Cấu hình server	Cho phép Admin có thể cấu hình các thông số ip listen, port cho server
10	Xem log	Cho phép Admin có thể xem log thông qua file hoặc log trực tiếp trên Server.exe

4. Đặc tả Use-case

4.1. Đặc tả Use-case “Cấu hình địa chỉ Server”

4.1.1. Tóm tắt

Người dùng Client để cấu hình địa chỉ và port server chat

4.1.2. Dòng sự kiện

Dòng sự kiện chính

1. Người dùng nhập IP và Port server chat
2. Người dùng ấn OK để lưu cấu hình hoặc ấn nút Cancel để thoát
3. Hiện thị giao diện dưới dạng 1 form dialog

Các dòng sự kiện khác

- a. Người dùng chưa nhập IP hoặc PORT, ấn OK
 1. Hệ thống thông báo thông tin chưa nhập
 2. Yêu cầu cung cấp lại thông tin
- b. Người dùng nhập sai IP
 1. Thông báo nhập sai IP hoặc PORT
 2. Yêu cầu nhập lại

Trình bày các dòng sự kiện khác. Nếu không có dòng sự kiện khác, ghi “Không có”

4.1.3. Các yêu cầu đặc biệt

Không

Ghi nhận các yêu cầu đặc biệt khi thực hiện Use-case. Nếu không có yêu cầu đặc biệt nào, ghi “Không có”

4.1.4. Trạng thái hệ thống khi bắt đầu thực hiện Use-case

Khởi chạy phần mềm, click vào menu Setting > Setting Server Information

[Type here]

[Type here]

[Type here]

Mô tả rõ điều kiện trước khi bắt đầu thực hiện Use-case (ví dụ có đòi hỏi người sử dụng phải đăng nhập thành công trước đó hay không...)

4.1.5. *Trạng thái hệ thống sau khi thực hiện Use-case*

Lưu cấu hình vào tệp tin setting_client.ini, tải lại cấu hình mới cho phần mềm.

Hiển thị màn hình client chính cho người sử dụng.

Mô tả rõ tình trạng hệ thống sau khi thực hiện Use-case (bao gồm cả trường hợp Use-case thực hiện thành công, hoặc thất bại).

4.1.6. *Điểm mở rộng*

Không

Mô tả những tình huống xuất hiện các Use-case khác có quan hệ <<extend>> với Use-case đang xét.

4.2. *Đặc tả Use-case “Xem hội thoại đã chat”*

4.2.1. *Tóm tắt*

Cho phép người sử dụng xem lại hội thoại mình đã liên lạc với người khác hoặc nhóm khác

4.2.2. *Dòng sự kiện*

Dòng sự kiện chính

1. Người dùng Click vào các đoạn hội thoại trên List Control
2. Hiển thị đoạn hội thoại lên màn hình như một cuộc hội thoại mới thông thường nhưng có dữ liệu cũ đã được tải lại

Các dòng sự kiện khác

- a. Có thông báo tin nhắn mới
 1. Hệ thống hiển thị icon có tin nhắn mới ở thanh list control chứa các hội thoại đã chat
 2. Người dùng click vào thì sẽ load dữ liệu cũ cùng với tin nhắn mới

4.2.3. *Các yêu cầu đặc biệt*

1. Người dùng phải đăng nhập trước khi thực hiện chức năng
2. Thư mục storage phải tồn tại

4.2.4. *Trạng thái hệ thống khi bắt đầu thực hiện Use-case*

Hệ thống đã đăng nhập tài khoản client, hiển thị form chính chưa có dữ liệu hội thoại cũ.

4.2.5. *Trạng thái hệ thống sau khi thực hiện Use-case*

Hiển thị màn hình chính với đoạn hội thoại đã được chọn tải từ tệp tin dữ liệu trong máy Client.

4.2.6. *Điểm mở rộng*

- Người dùng có thể xóa cuộc hội thoại hiện tại (dữ liệu lưu trong client) bằng cách ấn chọn xóa trong thanh công cụ
- Người dùng có trả lời hoặc gửi tin nhắn tiếp tục tới tài khoản đang xem hội thoại nếu người đó đang trực tuyến

4.3. *Đặc tả Use-case “Chat nhóm”*

4.3.1. *Tóm tắt*

Cho phép người dùng có thể tạo cuộc hội thoại nhóm với nhiều client đang online khác

4.3.2. Dòng sự kiện

Dòng sự kiện chính

1. Người sử dụng chọn button tạo hội thoại mới hoặc click chuột chọn các user đang online và click chuột phải chọn tạo hội thoại nhóm
 - Trường hợp dùng button tạo hội thoại thì cần điền tên các thành viên cần thêm vào hội thoại
2. Người dùng có thể bắt đầu gửi tin nhắn ở khung nội dung chat
3. Nhập xong nội dung có thể ấn gửi hoặc phím enter

Các dòng sự kiện khác

- a. Người dùng chưa nhập nội dung chat
 1. Hệ thống thông báo nội dung chat chưa được nhập
 2. Yêu cầu người dùng nhập nội dung
- b. Nội dung chat quá giới hạn 2000 ký tự
 1. Hệ thống báo nội dung chat quá giới hạn
 2. Yêu cầu người dùng chỉnh sửa nội dung chat hiện tại (Client sẽ không xoá các nội dung cũ trong khung nội dung để cho người sử dụng sửa trực tiếp)

4.3.3. Các yêu cầu đặc biệt

1. Người dùng phải đăng nhập
2. Chỉ thêm thành viên online vào chat nhóm hiện tại

4.3.4. Trạng thái hệ thống khi bắt đầu thực hiện Use-case

Người sử dụng đã đăng nhập và có danh sách online

4.3.5. Trạng thái hệ thống sau khi thực hiện Use-case

Hiện thị hội thoại chat nhóm cho người sử dụng

4.3.6. Điểm mở rộng

- Người sử dụng có thể xoá đoạn hội thoại hiện tại bằng button trên màn hình
- Người sử dụng có thể gửi file bằng button đính kèm trên màn hình
- Người sử dụng có thể gửi icon dạng unicode bằng chọn icon trên màn hình
- Người sử dụng có thể chọn chức năng gửi tạm thời để tự huỷ sau thời gian nhất định

4.4. Đặc tả Use-case “Gửi File”

4.4.1. Tóm tắt

Người sử dụng có thể gửi tệp tin trong cuộc hội thoại cá nhân hoặc nhóm

4.4.2. Dòng sự kiện

Dòng sự kiện chính

1. Trong cuộc hội thoại cá nhân hoặc nhóm, người sử dụng click chọn button đính kèm
2. Hộp thoại chọn file mở ra, người dùng click chọn file cần gửi
3. Chọn OK để tiếp tục hoặc Cancel
4. Thanh thông tin quá trình tải file hiện lên, người dùng có thể gửi tin nhắn mà không cần đợi gửi hết file

Các dòng sự kiện khác

- a. File không tồn tại
 - Nếu file sau khi chọn bị xoá dẫn đến không tồn tại thì hệ thống sẽ gửi thông báo cho người sử dụng biết file không tồn tại
- b. Người nhận mất kết nối giữa chừng
 - Nếu người nhận mất kết nối thì Client sẽ hiển thị thông báo người nhận đã đăng xuất
 - Hệ thống sẽ tự động khoá cuộc trò chuyện hiện tại và huỷ quá trình truyền file

4.4.3. Các yêu cầu đặc biệt

1. Người dùng phải đăng nhập
2. Người dùng phải chọn gửi file trong 1 cuộc hội thoại có thành viên online (đơn lẻ hoặc nhóm)

4.4.4. Trạng thái hệ thống khi bắt đầu thực hiện Use-case

- Người dùng đã đăng nhập thành công
- Danh sách online được cập nhật
- Mở một cuộc hội thoại

4.4.5. Trạng thái hệ thống sau khi thực hiện Use-case

- Màn hình hiển thị quá trình truyền file

4.4.6. Điểm mở rộng

1. Trong quá trình truyền, nếu muốn huỷ thì người dùng click chọn button “x” cạnh thanh trạng thái truyền file.
2. Trong quá trình truyền người dùng hoạt động chat bằng nội dung bình thường, nội dung text chat sẽ được ưu tiên truyền.

4.5. Đặc tả Use-case “Xem danh sách online”

4.5.1. Tóm tắt

Người dùng có thể xem danh sách các user đang online trên Client

Quản trị viên có thể xem danh sách các user đang online trên Server

4.5.2. Dòng sự kiện

Dòng sự kiện chính

1. Sau khi người dùng đăng nhập thì danh sách user online sẽ được tải từ trên server về
2. Người dùng có thể ấn Refresh để yêu cầu tải lại danh sách user online

Các dòng sự kiện khác

- c. Khi có người đăng nhập mới
 - Client sẽ tự động cập nhật danh sách online khi có người đăng nhập mới
- d. Khi có người đăng xuất
 - e. Client sẽ tự động cập nhật danh sách online khi có người đăng xuất

4.5.3. Các yêu cầu đặc biệt

- Người dùng phải đăng nhập thành công

4.5.4. Trạng thái hệ thống khi bắt đầu thực hiện Use-case

- Người dùng đăng nhập thành công, form chính được hiển thị

4.5.5. *Trạng thái hệ thống sau khi thực hiện Use-case*

- Màn hình hiển thị danh sách online hiện tại

4.5.6. *Điểm mở rộng*

- Người dùng có thể click chọn các user online và chuột phải để tạo cuộc hội thoại

4.6. *Đặc tả Use-case “Xem log”*

4.6.1. *Tóm tắt*

Quản trị viên có thể xem lại log người sử dụng server bằng file hoặc màn hình trên Server

4.6.2. *Dòng sự kiện*

Dòng sự kiện chính

1. Hiển thị thông tin theo thời gian thực hoạt động trên client và server
2. Quản trị viên click chọn xuất file thì dữ liệu sẽ được lưu vào 1 file .log

Các dòng sự kiện khác

a. Quản trị viên xoá nội dung log thời gian thực server

1. Quản trị viên click chọn button “Clear Log” trên màn hình server
2. Thông báo hỏi xác nhận việc xoá log, click OK để tiếp tục, Cancel để dừng lại use -case
3. Thông báo hiện lên xác nhận việc đã xoá thành công

4.6.3. *Các yêu cầu đặc biệt*

- Quản trị viên phải đăng nhập bằng tài khoản quản trị trên server

4.6.4. *Trạng thái hệ thống khi bắt đầu thực hiện Use-case*

- Quản trị viên đăng nhập thành công, server hoạt động và các client trao đổi dữ liệu được ghi lại trên màn hình log của server

4.6.5. *Trạng thái hệ thống sau khi thực hiện Use-case*

- Màn hình hiển thị log hoạt động của server và client, dữ liệu trao đổi giữa các đối tượng.

4.6.6. *Điểm mở rộng*

Không

4.7. *Đặc tả Use-case “Cấu hình trên Server”*

4.7.1. *Tóm tắt*

Quản trị viên cấu hình địa chỉ, port server lắng nghe, thư mục chứa file tài khoản

4.7.2. *Dòng sự kiện*

Dòng sự kiện chính

1. Quản trị viên click chọn Setting > Setting Server để mở form dialog cài đặt
2. Điền các thông tin địa chỉ, port server, thư mục account, thư mục chứa log, thư mục chứa file tệp tin tạm, thư mục chứa file hội thoại
3. Click OK để xác nhận, cancel để huỷ cài đặt hiện tại.

Các dòng sự kiện khác

- a. Người dùng chưa một trường nào đó, ấn OK
 - 1. Hệ thống thông báo thông tin chưa nhập
 - 2. Yêu cầu cung cấp lại thông tin
- b. Người dùng nhập sai IP/PORT
 - 1. Thông báo nhập sai IP hoặc PORT
 - 2. Yêu cầu nhập lại
- c. Đường dẫn thư mục không tồn tại
 - 1. Thông báo người dùng nhập sai đường dẫn hoặc đường dẫn không tồn tại
 - 2. Yêu cầu nhập lại

4.7.3. Các yêu cầu đặc biệt

Quản trị viên phải đăng nhập thành công để quản lý server

4.7.4. Trạng thái hệ thống khi bắt đầu thực hiện Use-case

Quản trị viên đăng nhập thành công, màn hình hiển thị thông tin cấu hình hiện tại

4.7.5. Trạng thái hệ thống sau khi thực hiện Use-case

Lưu cấu hình vào tệp tin setting_server.ini trên server, tải lại cấu hình mới cho phần mềm.

Hiển thị màn hình server chính cho người sử dụng với các thông tin cấu hình đã được cài mới.

4.7.6. Điểm mở rộng

Không

4.8. Đặc tả Use-case “Chat riêng tư”

4.8.1. Tóm tắt

Cho phép người dùng có thể tạo cuộc hội thoại mới với 1 tài khoản đang online khác

4.8.2 Dòng sự kiện

Dòng sự kiện chính

- 1. Người dùng chọn button tạo hội thoại mới hoặc click chuột phải vào item và chọn tạo hội thoại trong danh sách online
- 2. Người dùng có thể bắt đầu gửi tin nhắn ở khung nội dung chat
- 3. Để gửi tin nhắn ấn nút gửi hoặc phím enter

Các dòng sự kiện khác

- b. Nhận tin nhắn
 - 1. Khi nhận được tin nhắn thì nội dung tin nhắn sẽ được hiển thị trong khung chat
- c. Người sử dụng gửi tin nhắn có độ dài > 2000 ký tự
 - 1. Thông báo nhập nội dung người sử dụng nhập quá giới hạn
 - 2. Yêu cầu người sử dụng nhập lại trong khoảng giới hạn

4.8.3. Các yêu cầu đặc biệt

Người sử dụng phải đăng nhập

[Type here]

[Type here]

[Type here]

Chỉ có thể chat với người đang online

4.8.4. Trạng thái hệ thống khi bắt đầu thực hiện Use-case

Người sử dụng đã đăng nhập và có danh sách online

4.8.5. Trạng thái hệ thống sau khi thực hiện Use-case

Hiện thị hội thoại chat riêng cho người sử dụng

[Type here]

[Type here]

[Type here]

CHƯƠNG IV. THIẾT KẾ DỮ LIỆU

Gói tin gửi nhận Server Client

Cấu trúc gói tin

HEADER REGION (256 bytes)	DATA REGION (7936 bytes)
PACKAGE (8192 bytes)	

- Header Region (256 bytes)**

Tên trường	Vị trí	Kích thước	Diễn giải
COMMAND	0 (0x00)	4 bytes	Loại gói tin (EmessageCommand)
TOTAL_SIZE	4 (0x04)	4 bytes	Kích thước gói tin
CUR_PACK	8 (0x08)	4 bytes	Thứ tự gói tin hiện tại
SRC	12 (0x0C)	64 bytes	Hash địa chỉ nguồn (SHA256)
DES	76 (0x4C)	64 bytes	Hash địa chỉ đích (SHA256)
Flag Raw	140 (0x8C)	1 bytes	0x00 : Không sử dụng Raw Src, Des 0x01 : Sử dụng Raw Src, Des
Raw SRC	141 (0x8D)	11 bytes	Địa chỉ nguồn
Raw DES	152 (0x98)	11 bytes	Địa chỉ đích
Num Data Item	163 (0xA3)	4 bytes	Số lượng trường dữ liệu trong vùng Data
Delim	167 (0xA7)	1 bytes	Ký tự phân cách giữa các trường dữ liệu trong vùng Data
Reversed	168 (0xA8)	88 bytes	Dữ liệu dự trữ (dùng khi cần thiết)

- Data Region (8192 bytes)**

DATA 1	DELIM	DATA2	DELIM	...
---------------	--------------	--------------	--------------	-----

- Dữ liệu được phân tách bởi Delim trong header, mặc định là \0

Các cấu dựa trên các mục đích sử dụng

FLAG	Các trường header sử dụng	Ý nghĩa
CLIENT_SIGN_IN	Src : client Des: server	Username: tên tài khoản Password: mật khẩu
SERVER_REPONSE_SIGN_IN_SUCESS	Src: server Des: client	Tên user
SERVER_RESPONSE_SIGN_IN_ERROR	Src: server Des: client	Tên user
CLIENT_SIGN_UP	Src : client Des: server	Username: tên tài khoản Password: mật khẩu

[Type here]

[Type here]

[Type here]

CLIENT_REQUEST_PRIVATE_CHAT	Src: client Des: server	Username: Tên tài khoản cần tạo hội thoại riêng
	Src: server Des: client	Des_user: Tài khoản người chat Src_user: Tên tài khoản người tạo hội thoại riêng Hash_conversation: Hash id cuộc trò chuyện đã được tạo từ server
CLIENT_SEND_PRIVATE_CHAT	Src: tên tài khoản gửi Des: Server HashSrc: SHA256 của tài khoản gửi HashDes: ID hash của cuộc hội thoại	Message: Nội dung cuộc trò chuyện dưới dạng chuỗi không chứa NULL “\0”
CLIENT_REQUEST_LIST_ONLINE	Src: Client Des: Server	Yêu cầu server gửi danh sách tài khoản online
	Src: Server Des: Client	U1,U2,...Un là tên các tài khoản đang online
CLIENT_REQUEST_GROUP_CHAT	Src: client Des: server	ChatName: tên nhóm hội thoại U1,U2,...Un Danh sách thêm vào nhóm
SERVER_RESPONSE_HASH_KEY	Src: server Des: client	HashID: SHA256 Hash id của cuộc trò chuyện ChatName: Tên cuộc trò chuyện U1,U2,U3,...Un Danh sách thành viên cuộc trò chuyện
CLIENT_SEND_GROUP_CHAT	Src: tên tài khoản gửi Des: Server HashSrc: SHA256 của tài khoản gửi HashDes: ID hash của cuộc hội thoại	Message: Nội dung cuộc trò chuyện dưới dạng chuỗi không chứa NULL “\0”
CLIENT_REQUEST_TRANSFER_FILE	Src: Tên tài khoản gửi Des: “Server” SrcHash: SHA256 tên tài khoản gửi DesHash: ID của cuộc trò chuyện NumPackage: ChunkSize (sử dụng đặc biệt) TotalSize: Kích thước file truyền	FileName: Tên file truyền
CLIENT_BEGIN_TRANSFER_FILE	Src: Tên tài khoản gửi Des: “Server” SrcHash: SHA256 tên tài khoản gửi DesHash: ID của cuộc trò chuyện NumPackage: Số thứ tự của chunk TotalSize: Kích thước của File	Dữ liệu file (có thể chứa NULL) kích thước bằng chunksize yêu cầu
CLIENT_END_TRANSFER_FILE	Src: Tên tài khoản gửi Des: “Server” SrcHash: SHA256 tên tài khoản gửi	Không sử dụng

[Type here]

[Type here]

[Type here]

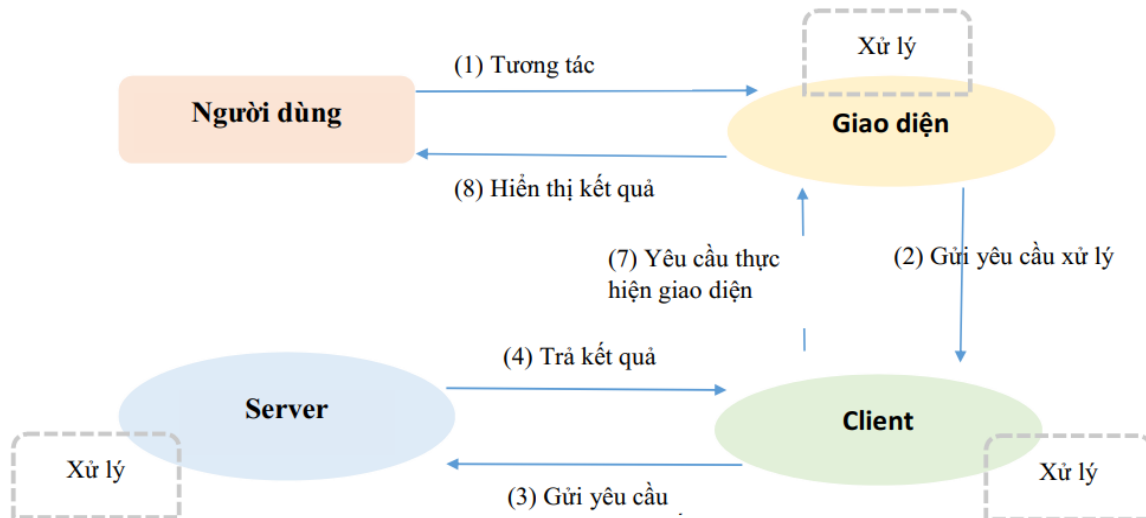
[Type here]

[Type here]

[Type here]

	DesHash: ID của cuộc trò chuyện	
--	---------------------------------	--

1.3. Mô hình đối tượng



- Thành phần trong hệ thống
 - **Server**: Cấu trúc xử lý trong lớp **ServerSocket**, tiếp nhận và xử lý trực tiếp đối với các gói tin nhận từ **Client**
 - **Client**:
 - + **ClientBackgroundService**: Tiếp nhận trung gian điều khiển quá trình xử lý. Được tạo ra một lần duy nhất và được truyền giữa các Form khác nhau để sử dụng
 - + **ClientSocket**: Tiếp nhận và gửi gói tin giao tiếp với **Server**. Các sự kiện nhận được sẽ gửi qua lớp **Giao diện** để xử lý thông qua

```
SendMessageW(  
    _In_ HWND hWnd,  
    _In_ UINT Msg,  
    _Pre_maybenull_ _Post_valid_ WPARAM wParam,  
    _Pre_maybenull_ _Post_valid_ LPARAM lParam);
```

1. Kịch bản và Luồng xử lý dữ liệu

1.1. Đăng ký tài khoản

- Sau khi người dùng click vào đăng ký tài khoản, **Client** sẽ gọi hàm **void Register(string username, string password)** để gửi gói tin với cấu trúc **CLIENT_SIGN_UP** lên server
 - **Tham số**:
 - o **username**: tên tài khoản
 - o **password**: mật khẩu
- ServerSocket** tiếp nhận gói tin trên và kiểm tra bằng hàm trong **AccountManagement**

[Type here]

[Type here]

[Type here]

[Type here]

[Type here]

[Type here]

- Nếu tài khoản tồn tại, phản hồi gói tin **SERVER_RESPONSE_SIGN_UP_ERROR**
 - Nếu tài khoản chưa tồn tại, thêm vào tệp tin dữ liệu tài khoản với dữ liệu tài khoản mới **username,password** và gửi phản hồi **SERVER_RESPONSE_SIGN_UP_SUCCESS**
3. *ClientSocket* tiếp nhận phản hồi và gửi gói tin tới lớp giao diện để xử lý
- Thành công:

```
SendMessageW(hwnd, IDC_FORM_LOGIN_MSG_HANDLER,  
IDC_FORM_LOGIN_MSG_HANDLER_REGISTER_SUCCESS, 0);
```

- Thất bại:

```
SendMessageW(hwnd, IDC_FORM_LOGIN_MSG_HANDLER,  
IDC_FORM_LOGIN_MSG_HANDLER_REGISTER_ERROR, 0);
```

4. Lớp giao diện tiếp nhận message gửi từ *SendMessageW*
- Thành công: Thông báo thành công và yêu cầu người đăng nhập login để thực hiện các chức năng của client
 - Thất bại: Thông báo thất bại, vui lòng chọn tài khoản khác.

1.2. Đăng nhập tài khoản

1. Sau khi người dùng click vào nhập tài khoản, Client sẽ gọi hàm

void Login(string username, string password) để gửi gói tin với cấu trúc **CLIENT_SIGN_UP** lên server

- Tham số:
 - o **username: tên tài khoản**
 - o **password: mật khẩu**

2. **ServerSocket** tiếp nhận gói tin trên và kiểm tra bằng hàm trong **AccountManagement**

- Nếu tài khoản sai mật khẩu hoặc không tồn tại, phản hồi gói tin **SERVER_RESPONSE_SIGN_IN_ERROR**
- Nếu kiểm tra tài khoản đúng server sẽ phản hồi gói tin thành công **SERVER_RESPONSE_SIGN_IN_SUCCESS**

3. *ClientSocket* tiếp nhận phản hồi và gửi gói tin tới lớp giao diện để xử lý
- Thành công:

```
SendMessageW(hwnd, IDC_FORM_LOGIN_MSG_HANDLER,  
IDC_FORM_LOGIN_MSG_HANDLER_LOGIN_SUCCESS, 0);
```

- Thất bại:

```
SendMessageW(hwnd, IDC_FORM_LOGIN_MSG_HANDLER,  
IDC_FORM_LOGIN_MSG_HANDLER_LOGIN_ERROR, 0);
```

[Type here]

[Type here]

[Type here]

[Type here]

[Type here]

[Type here]

4. Lớp giao diện tiếp nhận message gửi từ **SendMessageW**

- Thành công: Khởi tạo lớp **messenger** để mở form cho phép người dùng sử dụng các chức năng với tài khoản đã đăng nhập
- Thất bại: Thông báo thất bại, vui lòng nhập lại mật khẩu

1.3. Danh sách tài khoản online

- Kịch bản gửi danh sách tài khoản online được gửi từ Server xuống Client trong các tình huống sau:

+ Có client bị mất kết nối hoặc đăng xuất

+ Có client đăng nhập mới

+ 1 Client yêu cầu lấy danh sách online mới

- Kịch bản xử lý
 1. Có yêu cầu hoặc gặp tình huống cần lấy danh sách online
 2. Server tiếp nhận yêu cầu và lấy danh sách qua hàm **ServerSocket::GetListOnline()**;
 3. **ServerSocket** gửi gói tin với các Client hoặc 1 client chỉ định với FLAG **CLIENT_REQUEST_LIST_ONLINE**
 4. **ClientSocket** nhận được gói tin và lưu danh sách online vào trong **ClientSocket::_list_online** và gửi 1 message tới lớp giao diện

SendMessageW(hwnd, IDC_FORM_CHAT_MSG_HANDLER,
IDC_FORM_CHAT_MSG_HANDLER_LIST_ONLINE, 0);

5. Lớp giao diện nhận được message và cập nhật lại **ListControl** trên giao diện và hiển thị thông báo tới các cuộc hội thoại có user logout

1.4. CHAT riêng tư

- Tạo chat riêng tư
 1. Người sử dụng chọn 1 tài khoản đang online bằng cách double click trên **ListControl** friends.
 2. Client sẽ gửi 1 gói tin với cấu trúc **CLIENT_REQUEST_PRIVATE_CHAT** với dữ liệu
 - Tài khoản nhận
 - Tên cuộc trò chuyện (mặc định là tên tài khoản người nhận)
 3. Server tiếp nhận gói tin **CLIENT_REQUEST_PRIVATE_CHAT** xử lý tạo **PrivateConversation** và lưu vào danh sách **_lcs** quản lý các cuộc hội thoại đã tạo với 1 khoá SHA256 xây dựng dựa trên tên tài khoản khởi tạo cuộc trò chuyện + random 1 chuỗi ngẫu nhiên

[Type here]

[Type here]

[Type here]

[Type here]

[Type here]

[Type here]

4. Server gửi gói tin **SERVER_RESPONSE_HASH_KEY** chứa mã **HASH_ID** của cuộc trò chuyện và danh sách thành viên trong cuộc trò chuyện
 5. Các client có tham gia cuộc trò chuyện nhận được gói tin **SERVER_RESPONSE_HASH** xử lý và thêm cuộc trò chuyện vào ListControl bên phải. Nếu cuộc trò chuyện đang được chọn thì hiển thị trực tiếp vào khu vực chat
- Gửi tin nhắn chat riêng tư:
1. Client nhập đoạn hội thoại và ấn button có hình máy bay để gửi.
 - Gói tin gửi có cấu trúc **CLIENT_SEND_PRIVATE_CHAT**
 2. Server nhận được sẽ lấy hash_id của cuộc trò chuyện và gửi tới tất cả client với nội dung gói tin giữ nguyên
 3. Các Client nhận gói tin sẽ tìm kiếm cuộc trò chuyện và thêm tin nhắn vào **list_mess**
 - Nếu khung chat hiện tại đang là cuộc trò chuyện hiện thời thì hiển thị nội dung tin nhắn lên khung chat

1.5. Tạo CHAT nhóm

- Tạo chat nhóm
1. Người sử dụng 2 tài khoản đang online trở lên bằng cách click chuột trái trên **ListControl** friends chọn các tài khoản đang online và click chuột phải điền các thông tin tên cuộc trò chuyện.
 2. Client sẽ gửi 1 gói tin với cấu trúc **CLIENT_REQUEST_GROUP_CHAT** với dữ liệu
 - Tên cuộc trò chuyện
 - Danh sách thành viên trong cuộc trò chuyện
 3. Server tiếp nhận gói tin **CLIENT_REQUEST_GROUP_CHAT** xử lý tạo **GroupConversation** và lưu vào danh sách **_lcs** quản lý các cuộc hội thoại đã tạo với 1 khoá SHA256 xây dựng dựa trên tên tài khoản khởi tạo cuộc trò chuyện + random 1 chuỗi ngẫu nhiên
 4. Server gửi gói tin **SERVER_RESPONSE_HASH_KEY** chứa mã **HASH_ID** của cuộc trò chuyện và danh sách thành viên trong cuộc trò chuyện
 5. Các client có tham gia cuộc trò chuyện nhận được gói tin **SERVER_RESPONSE_HASH** xử lý và thêm cuộc trò chuyện vào **ListControl** bên phải. Nếu cuộc trò chuyện đang được chọn thì hiển thị trực tiếp vào khu vực chat
- Gửi tin nhắn nhóm
1. Client nhập đoạn hội thoại và ấn button có hình máy bay để gửi.
 - Gói tin gửi có cấu trúc **CLIENT_SEND_GROUP_CHAT**
 2. Server nhận được sẽ lấy hash_id của cuộc trò chuyện và gửi tới tất cả client với nội dung gói tin giữ nguyên

[Type here]

[Type here]

[Type here]

[Type here]

[Type here]

[Type here]

3. Các Client nhận gói tin sẽ tìm kiếm cuộc trò chuyện và thêm tin nhắn vào **list_mess**
 - Nếu khung chat hiện tại đang là cuộc trò chuyện hiện thời thì hiển thị nội dung tin nhắn lên khung chat

1.6. Gửi file

1. Người sử dụng chọn button đính kèm file và chọn file cần đính kèm
2. Client sẽ gửi gói tin với cấu trúc **CLIENT_REQUEST_TRANSFER_FILE** với một số dữ liệu đặc biệt:
 - NumPackage = Chunksize (kích thước 1 chunk truyền file)
 - TotalSize = Kích thước tệp tin
 - FileName : Tên file
3. Server nhận được sẽ tìm kiếm dựa trên Hash_id của cuộc trò chuyện và gửi yêu cầu truyền file tới tất cả các Client thành viên trừ người gửi
4. Các Client thành viên nhận được và gọi hàm **void ClientConversation::InitFileTransferManagement(string file_name, int total_size, int chunk_size)** và gán **transferring = true** để khoá không cho truyền file khác
5. Client gửi file sẽ gửi tiếp các gói tin **CLIENT_BEGIN_TRANSFER_FILE** với
 - NumPackage = vị trí gói tin
 - TotalSize = kích thước tệp tin
 - Dữ liệu: ChunkSize (bytes) của gói tin đang truyền
6. Sau khi gửi hết dữ liệu thì Client gửi file sẽ gửi gói tin **CLIENT_END_TRANSFER_FILE** để kết thúc truyền file
7. Các client sẽ đóng buffer, thông báo MessageBox truyền file thành công và chuyển **transferring = false**.

Lưu ý:

- File có thể mở trực tiếp bằng click vào danh sách file truyền hoặc tin nhắn trên khung chat
- Trong 1 cuộc hội thoại chỉ 1 client truyền và truyền trong 1 thời điểm. Các client khác ấn truyền sẽ bị thông báo lỗi

[Type here]

[Type here]

[Type here]

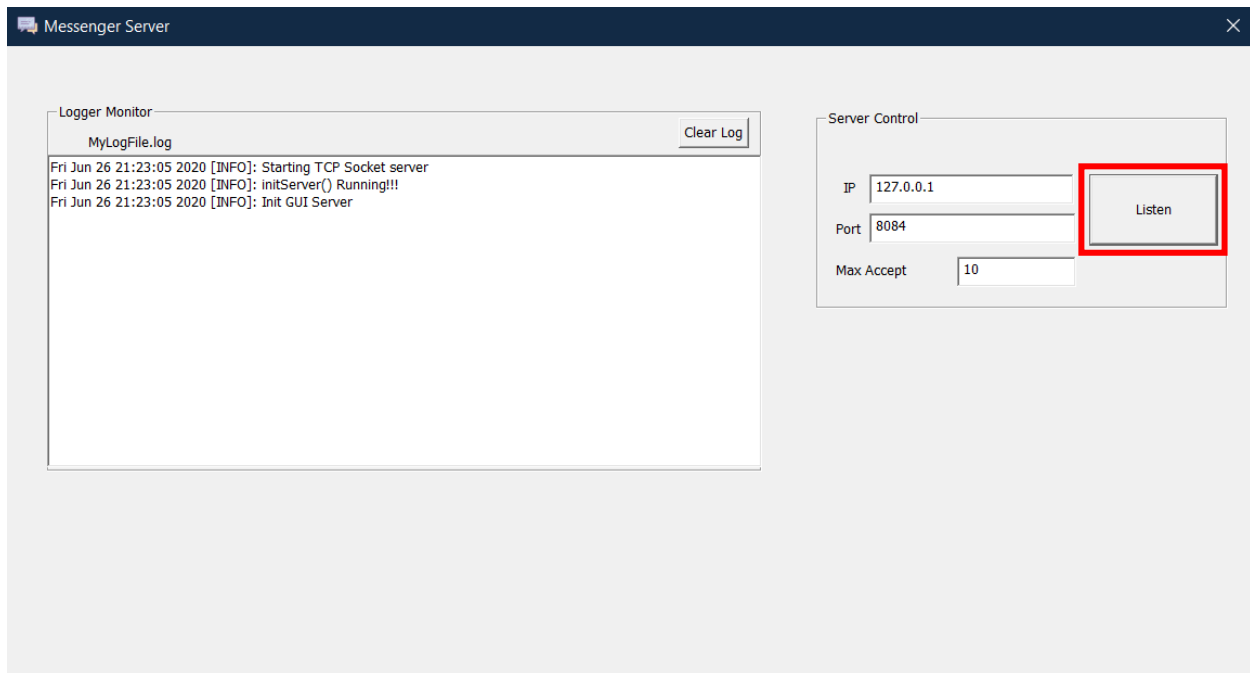
[Type here]

[Type here]

[Type here]

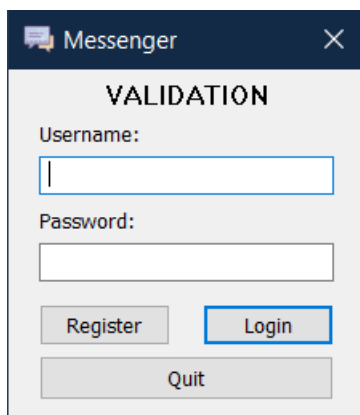
CHƯƠNG V. HƯỚNG DẪN SỬ DỤNG

Phía Server



- Chọn Listen để server lắng nghe kết nối từ client

Phía Client



Cửa sổ Message yêu cầu người dùng đăng nhập

Nếu chưa có account thì phải đăng kí

[Type here]

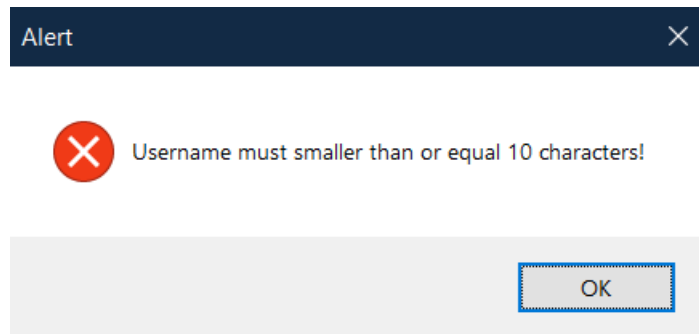
[Type here]

[Type here]

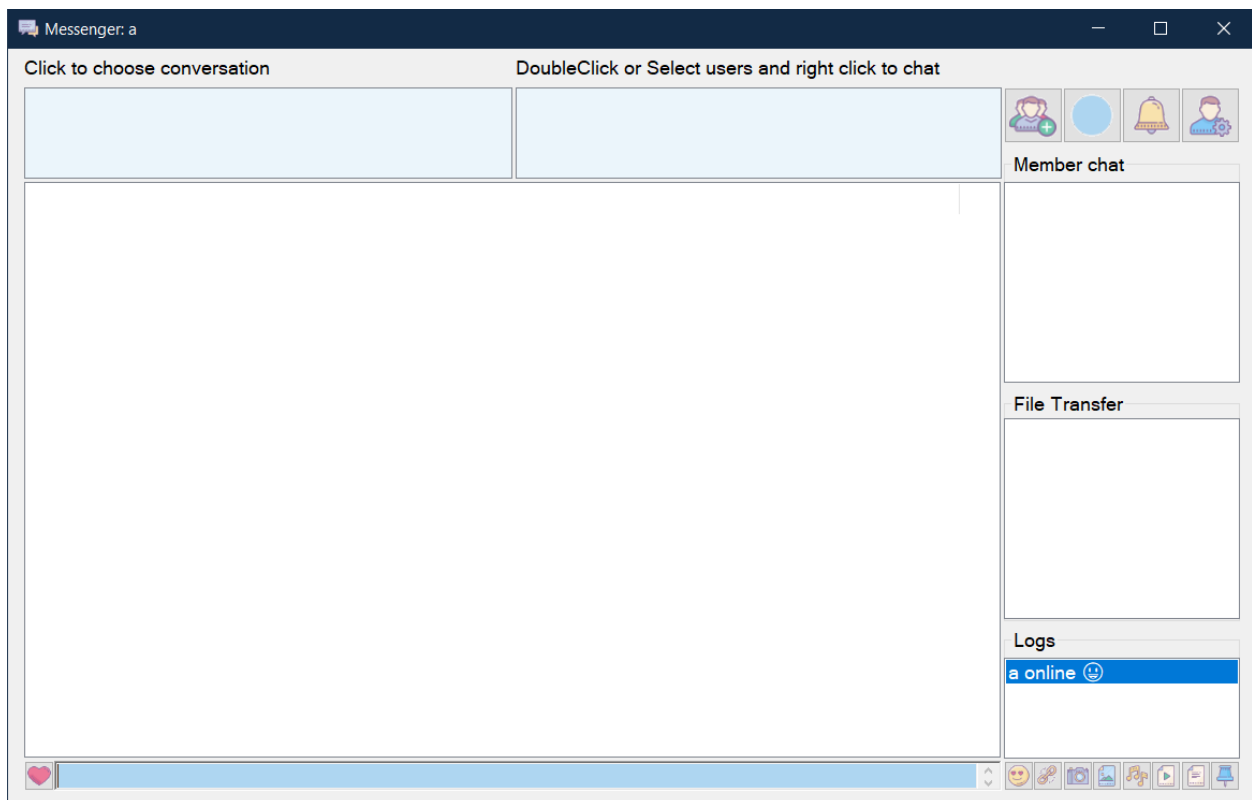
[Type here]

[Type here]

[Type here]



Username yêu cầu người dùng nhập *không quá 10 kí tự*. Nếu hơn ứng dụng sẽ báo lỗi



Đây là giao diện của ứng dụng sau khi đăng nhập thành công



Phía trên cùng góc phải có các chức năng:

- Tạo nhóm
- Thay đổi màu khung chat
- Bật/tắt âm thông báo
- Cài đặt tùy chọn

[Type here]

[Type here]

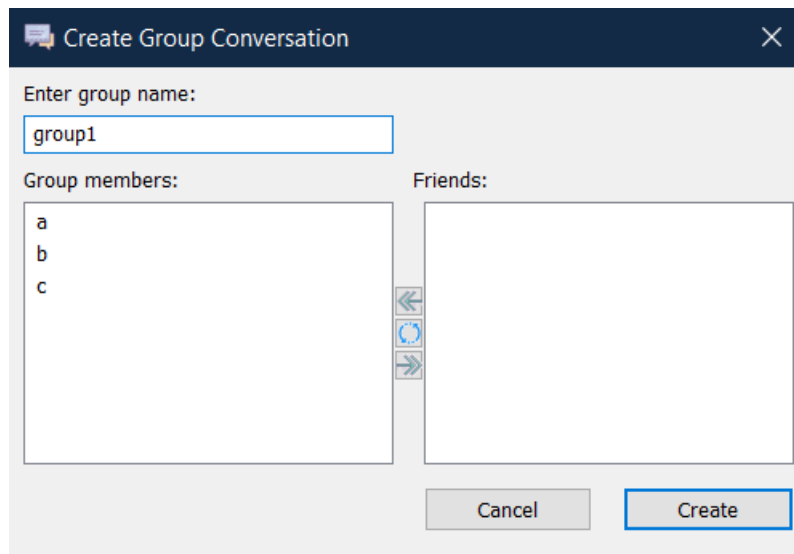
[Type here]

[Type here]

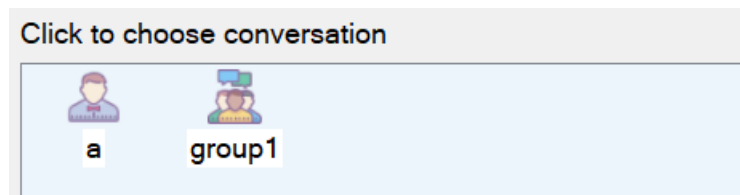
[Type here]

[Type here]

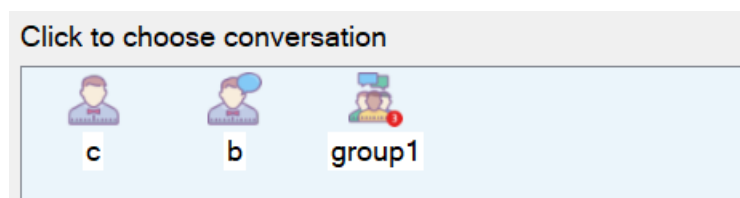
1. TẠO NHÓM



Khi muốn tạo group, ứng dụng sẽ hiện ra cửa sổ để người dùng nhập tên group và chọn các thành viên để thêm vào nhóm đó. Chính giữa là các nút chọn để người dùng có thể dễ dàng chuyển danh sách qua lại một cách nhanh chóng.



Sau khi tạo xong thì group sẽ được hiển thị ở khung trên cùng bên trái giao diện. Nếu có tin nhắn mới đến thì đây cũng là nơi thông báo



2. THAY ĐỔI MÀU KHUNG CHAT

[Type here]

[Type here]

[Type here]

[Type here]

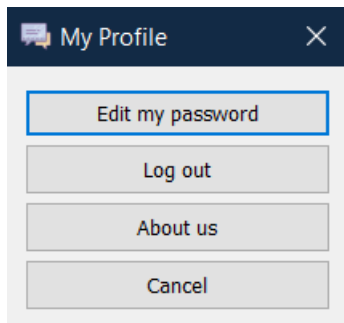
[Type here]

[Type here]



Có 9 màu để người dùng có thể lựa chọn để khung chat thêm sinh động

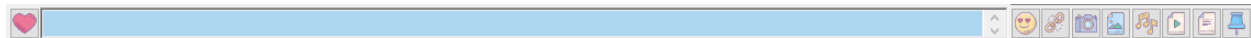
3. CÀI ĐẶT TUÝ CHỌN



Có các chức năng:

- Thay đổi password
- Đăng xuất
- Thông tin ứng dụng

4. PHẦN CHAT



Phía dưới cùng của ứng dụng là phần chat, có một số chức năng:

- Chọn emoji
- Nhập link muốn gửi
- Camera
- Chọn ảnh/âm thanh/video/text muốn gửi từ máy tính

[Type here]

[Type here]

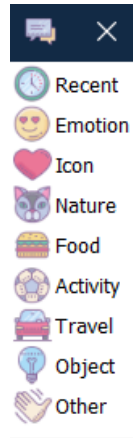
[Type here]

[Type here]

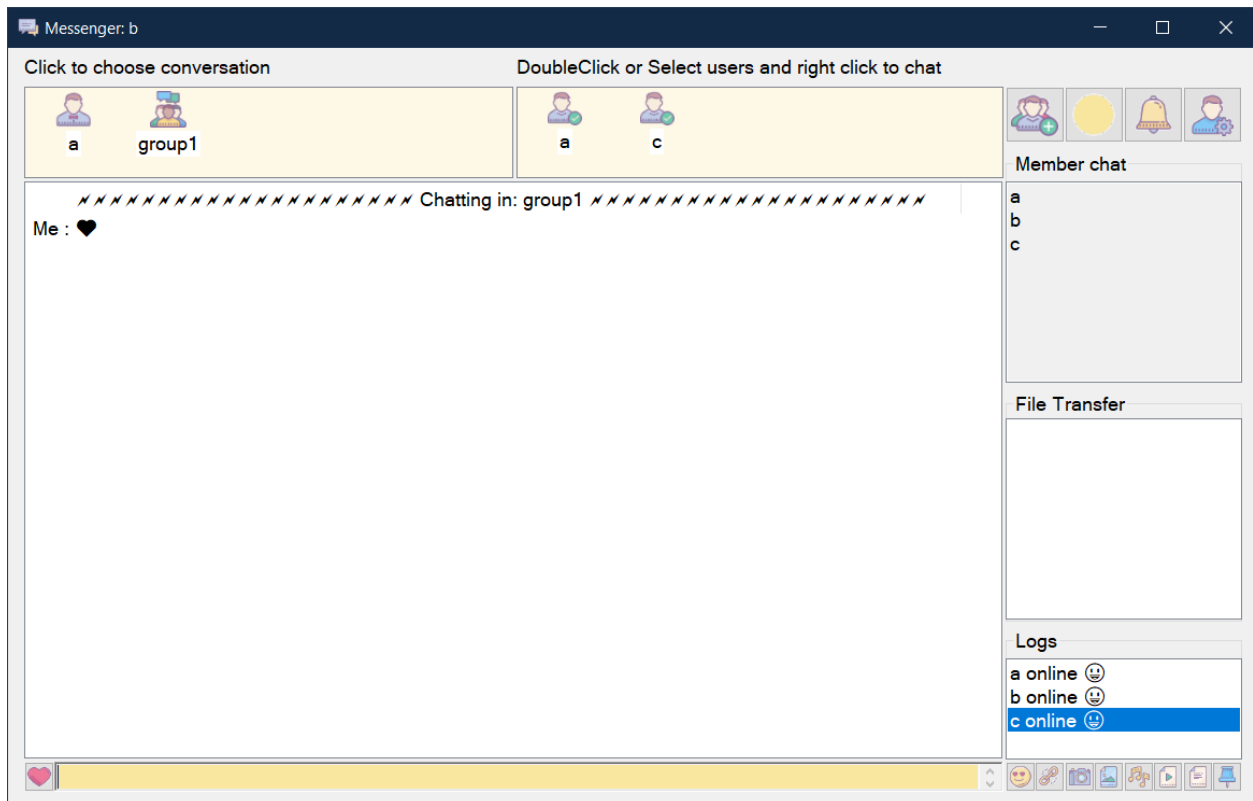
[Type here]

[Type here]

Chọn emoji



GIAO DIỆN CHAT



Khi **enter**, icon hình trái tim sẽ được gửi mặc định

[Type here]

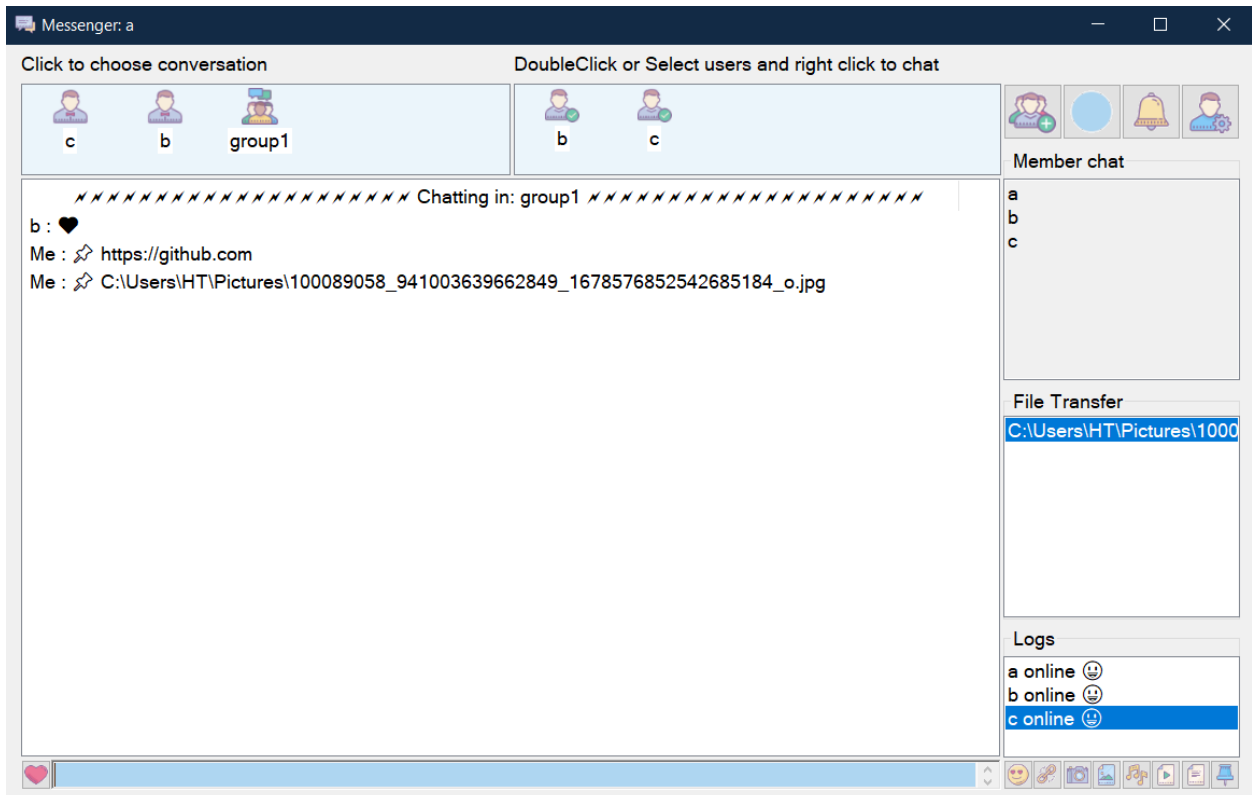
[Type here]

[Type here]

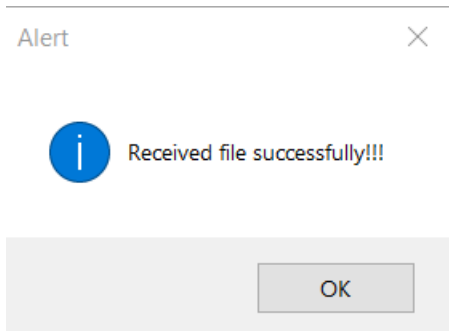
[Type here]

[Type here]

[Type here]

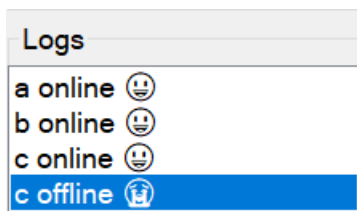


Đây là giao diện khi người dùng muốn gửi đi 1 link hoặc 1 file ảnh nào đó. File được gửi đi sẽ được xuất hiện trong khung File Transfer. Và khi bên nhận nhận được file sẽ hiện thông báo nhận file thành công.



Cùng với đó, ta cũng thấy số lượng thành viên được hiển thị ở khung Member Chat

Khung Logs hiển thị trạng thái của người dùng

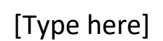


[Type here]

[Type here]

[Type here]

[Type here]



CHƯƠNG VI: ĐÁNH GIÁ

MSSV	Họ & Tên	Tổng quan	Chi tiết	Phần trăm đồ án	Phần trăm hoàn thành
18127208	Ngô Thanh Phương Thái	Giao diện MFC & Quản lí Account (text)	<ul style="list-style-type: none">Nghiên cứu + hướng dẫn nhóm sử dụng MFC cơ bảnThiết kế các dialog cần thiết, xây dựng bộ cục chương trình và xử lý thao tác chuột, hiển thị tin nhắn gửi và nhậnXây dựng các hàm để thay đổi icon, thay đổi màu sắc + animationThiết kế các hàm cơ bản phục vụ việc quản lí tài khoản, bật tắt âm thanh, mở file dựa trên đường dẫn		100%
18127004	Nguyễn Vũ Thu Hiền	Designer + Xử lí file + Client logs + Hướng dẫn sử dụng	<ul style="list-style-type: none">Tìm kiếm icon + thiết kế các pending message iconXây dựng các nút gửi file, các cờ xác định file + mở file khi nhận được dựa trên cờLogs ở client, hiển thị ai online / offlineViết hướng dẫn sử dụng chi tiết		100%
18127027	Trần Minh Đức	Core socket + Kết hợp core vào UI + Báo cáo chi tiết socket	<ul style="list-style-type: none">Nghiên cứu thư viện Socket của C++Thiết kế các cấu trúc gói tin + Xử lý core		100%

[Type here]

[Type here]

[Type here]

CHƯƠNG V: Tham khảo

1. <https://github.com/elhoangvu/ClientServerChat-C-Socket>

2. Demo được cung cấp

[Type here]

[Type here]

[Type here]