

# BÁO CÁO ĐỒ ÁN

# LẬP TRÌNH SOCKET

**MÔN: MẠNG MÁY TÍNH – CSC10008 (18CLC5)**

**GIẢNG VIÊN:**

**LÝ THUYẾT: LÊ GIANG THANH**

**THỰC HÀNH: NGUYỄN THỊ THANH HUYỀN**

**THÀNH VIÊN:**

**18127175 – DƯƠNG TẤN PHÁT \* Nhóm trưởng**

**18127209 – HÀ MINH THANH**

## PHẦN I: GIỚI THIỆU

### 1. Yêu cầu đồ án

**Cơ bản:** Xây dựng ứng dụng cho mô hình chat client – server, sử dụng lập trình socket, đa tiến trình multi thread trong C++ và giao thức TCP cho các chức năng sau:

1. Đăng ký, đăng nhập tài khoản.
2. Kiểm tra một tài khoản qua username.
3. Thay đổi mật khẩu tài khoản.
4. Thay đổi thông tin người dùng.
5. Upload dữ liệu lên server và download dữ liệu từ server.
6. Tạo kênh chat riêng cho cặp người dùng đang online.

**Nâng cao:** Xây dựng cơ sở dữ liệu cho server quản lý thông tin các user. Lập trình giao diện Win32 API, hỗ trợ UTF-16.

### 2. Kịch bản đồ án

Sử dụng socket và multithread để gửi tin nhắn giữa client – server:

- Server tạo socket để sẵn sàng lắng nghe và kết nối mỗi khi có một client có nhu cầu.
- Mỗi client khi kết nối đến server, server sẽ tạo 1 socket và 1 thread tương ứng để nhận gói tin. Khi nhận được gói tin, server sẽ thực hiện theo yêu cầu đính kèm trong gói tin.
- Sau khi thực hiện xong, khi client ngắt kết nối thì server sẽ hủy các thông tin liên quan đến client đó.

Với mỗi yêu cầu chức năng, client sẽ gửi lên server 1 gói tin request. Server dựa theo FLAG và nội dung gói tin để thực hiện yêu cầu và trả kết quả về cho client. Sau khi xử lý, server tạo lại 1 gói trả lời lại cho client. Client dựa theo FLAG và nội dung để xử lý.

## PHẦN II: PHÂN TÍCH ĐỒ ÁN

### 1. Gửi nhận dữ liệu sử dụng socket kết hợp multithread

Sử dụng STREAM SOCKET của C++

Server tạo 1 socket nằm ở 1 luồng riêng biệt để lắng nghe khi nào có client mới kết nối. Với mỗi một client kết nối đến server thì server tạo ra 1 socket mới và 1 thread mới tương ứng để luôn chờ nhận gói tin và lắng nghe khi nào client không còn kết nối sẽ loại client đó khỏi danh sách. Khi nhận được gói tin thì thực hiện theo loại của gói tin yêu cầu.

Client tạo 1 socket để connect đến server đồng thời tạo các thread để luôn chờ nhận gói tin trả về của server.

### 2. Chức năng đăng ký và đăng nhập tài khoản

Client gửi username và password lên server với một dấu hiệu nhận biết. Server nhận được username và password sẽ kiểm tra trong cơ sở dữ liệu hiện tại. Nếu tài khoản tồn tại -> gửi mã đăng kí thất bại về client, nếu thành công thì cập nhật lại cơ sở dữ liệu cũng như danh sách client đang online và gửi mã đăng ký thành công về cho Client. Chức năng đăng nhập tương tự.

### 3. Chức năng Chat với 1 người

Client gửi username của đối tượng muốn chat lên server, server kiểm tra user có online hay không. Nếu có thì gửi mã lệnh thành công về client - client nhận được sẽ tạo phòng chat 2 người, ngược lại gửi mã lệnh thất bại – client báo thông tin thất bại ra cho người dùng.

Trong mỗi hành động chat thì client gửi văn bản chat lên server, server sẽ gửi lại cho user bên kia văn bản.

Một trong 2 rời khỏi phòng chat thì gửi yêu cầu lên server, server gửi thông tin kết thúc quá trình chat cho user còn lại.

### 4. Chức năng thay đổi password

Client gửi lên server gói tin chứa username, mật khẩu cũ và mật khẩu mới. Server dựa vào username và mật khẩu cũ trong database kiểm tra xem có hợp lệ không. Nếu hợp lệ thay đổi mật khẩu và gửi mã thay đổi thành công cho client, ngược lại gửi mã thất bại.

### 5. Chức năng kiểm tra thông tin người dùng

Client gửi lên server gói tin chứa username cần kiểm tra thông tin. Server dựa vào mã FLAG để biết kiểm tra thông tin loại nào. Sau đó kiểm tra trong database để gửi đoạn mã kết quả về cho client.

## 6. Chức năng thay đổi thông tin người dùng

Client gửi lên server gói tin chứa username và thông tin cần thay đổi. Server dựa vào mã FLAG để biết thay đổi thông tin loại nào. Sau đó kiểm tra trong database để thay đổi thông tin và lưu lại database mới đồng thời gửi đoạn mã kết quả về cho client.

## 7. Chức năng gửi nhận tập tin

*Upload:* Client gửi tên tập tin, kích thước lên cho server. Server kiểm tra trong database có tập tin đó chưa. Nếu chưa có gửi mã bắt accept và bắt đầu nhận dữ liệu. Gói tin được chia làm nhiều phần và gửi liên tục với 1 mã lệnh gửi tập tin, server nhận và ghép lại.

*Download:* Client gửi tên tập tin cho server. Server kiểm tra xem trong database có tồn tại tập tin đó hay không. Nếu có thì bắt đầu quá trình gửi dữ liệu cho client. Gói tin được chia làm nhiều phần và gửi liên tục với 1 mã lệnh gửi tập tin, client nhận và ghép lại.

## PHẦN III: TỔNG QUAN MÔ HÌNH

### 1. Ý tưởng

Người dùng tương tác với phần mềm thông qua giao diện, từ giao diện ở 1 client cho biết được yêu cầu của người dùng thực hiện 1 chức năng nào đó. Với mỗi yêu cầu chức năng client sẽ gửi lên server 1 gói tin theo 1 định dạng qui định trước. Server dựa theo FLAG và nội dung gói tin để thực hiện yêu cầu và trả kết quả về cho client (nếu có).

Server ngoài việc xử lý yêu cầu còn quản lý việc client nào đã kết nối, đã ngắt kết nối, người dùng nào đã đăng ký, đăng nhập,... Tất nhiên có áp dụng multi-threading vào cả client và server để thực hiện việc nhận và gửi tin đồng thời bởi các luồng riêng biệt.

### 2. Các loại gói tin

Tất cả các loại (FLAG) thuộc kiểu enum với tên là Flags

FLAG	Nội dung	Ý nghĩa
REGISTER	username   NULL   password   NULL	Đăng ký tài khoản
REGISTER_SUCCESS		Đăng ký thành công
REGISTER_FAILURE		Đăng ký thất bại
LOGIN	username   NULL   password   NULL	Đăng nhập
LOGIN_SUCCESS		Đăng nhập thành công
LOGIN_FAILURE		Đăng nhập thất bại
CHANGE_PASSWORD	username   NULL   old_password   NULL   new_password   NULL	Thay đổi mật khẩu
CHANGE_PASSWORD_SUCCESS		Thay đổi mật khẩu thành công
CHANGE_PASSWORD_FAILURE		Thay đổi mật khẩu thất bại
CHECK_USER_FIND	username   NULL	Kiểm tra người dùng có tồn tại
USER_FIND_SUCCESS		Người dùng tồn tại
USER_FIND_FAILURE		Người dùng không tồn tại
CHECK_USER_ONLINE	username   NULL	Kiểm tra người dùng online
USER_ONLINE_SUCCESS	0 / 1   NULL	Người dùng online / offline
USER_ONLINE_FAILURE		Người dùng không tồn tại
CHECK_USER_SHOW_DATE	username   NULL	Kiểm tra ngày sinh
USER_SHOW_DATE_SUCCESS	date   NULL	Thông tin về ngày sinh
USER_SHOW_DATE_FAILURE		Người dùng không tồn tại

CHECK_USER_SHOW_FULLNAME	username   NULL	Kiểm tra tên
USER_SHOW_FULLNAME_SUCCESS	fullname   NULL	Thông tin về tên
USER_SHOW_FULLNAME_FAILURE		Người dùng không tồn tại
CHECK_USER_SHOW_NOTE	username   NULL	Kiểm tra ghi chú
USER_SHOW_NOTE_SUCCESS	note   NULL	Thông tin ghi chú
USER_SHOW_NOTE_FAILURE		Người dùng không tồn tại
CHECK_USER_SHOW_ALL	username   NULL	Kiểm tra tất cả thông tin
USER_ONLINE_SUCCESS	0 / 1   fullname   NULL   date   NULL   note   NULL	Thông tin tất cả thông tin
USER_ONLINE_FAILURE		Người dùng không tồn tại
SETUP_INFO_FULLNAME	username   NULL   fullname   NULL	Thay đổi tên
SETUP_INFO_FULLNAME_SUCCESS		Thay đổi tên thành công
SETUP_INFO_FULLNAME_FAILURE		Thay đổi tên thất bại
SETUP_INFO_DATE	username   NULL   date   NULL	Thay đổi ngày sinh
SETUP_INFO_DATE_SUCCESS		Thay đổi ngày sinh thành công
SETUP_INFO_DATE_FAILURE		Thay đổi ngày sinh thất bại
SETUP_INFO_NOTE	username   NULL   note   NULL	Thay đổi ghi chú
SETUP_INFO_NOTE_SUCCESS		Thay đổi ghi chú thành công
SETUP_INFO_NOTE_FAILURE		Thay đổi ghi chú thất bại
CREATE_CHAT	partner   NULL	Tạo chat
CREATE_CHAT_SUCCESS	partner   NULL	Tạo chat thành công
CREATE_CHAT_FAILURE		Tạo chat thất bại
CHAT	sender   NULL   receiver   NULL   content   NULL	Nội dung chat
END_CHAT	sender   NULL   receiver   NULL	Kết thúc chat
SEND_FILE	file name   NULL   file size   NULL	Yêu cầu muốn gửi file
SEND_FILE_CANCEL	file name   NULL   file size   NULL	Không chấp nhận file
SEND_FILE_ACCEPT	file name   NULL   file size   NULL	Chấp nhận file
FILE_DATA	size   NULL   content	Nội dung file được chia nhỏ
CONTINUE		Yêu cầu nhận tiếp nội dung file
STOP		Đã nhập đủ nội dung file
DOWN_FILE	filename   NULL	Yêu cầu nhận file
DOWN_FILE_CANCEL	filename   NULL	Không có file
DOWN_MULTI	number   file1   NULL   file2   NULL	Yêu cầu down multi file

## PHẦN IV: CÀI ĐẶT YÊU CẦU ĐỒ ÁN

### 1. Chức năng đăng ký và đăng nhập tài khoản

#### a. Phía client

**LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)**

1. Bắt message khi người dùng nhấn vào button “REGISTER” hay “LOGIN” thông qua **WM\_COMMAND** với ID của button là **IDC\_REGISTER** và **IDC\_LOGIN**. Lấy username và password ở 2 ô edittext “edUsername” – “edPassword” kiểm tra hợp lệ và tạo 1 gói tin yêu cầu đăng ký gửi lên server với FLAG là **REGISTER** hay **LOGIN** bằng hàm **sendMessage**.

2. Nhận kết quả trả về thông qua **receiverMessage** và gửi message đến giao diện với **WM\_COMMAND** với ID là **IDC\_RECEIVE**. wParam là con trỏ tới tin nhắn nhận được. Kết quả trả về có thể là: **REGISTER\_SUCCESS**, **REGISTER\_FAILURE**, **LOGIN\_SUCCESS**, **LOGIN\_FAILURE**.

**int sendMessage(WCHAR\* message, int len);**

Gửi gói tin đến server. Được yêu cầu từ **WndProc** thông qua **GlobalClient**.

**int receiveMessage();**

Nhận gói tin từ server. Sau đó gửi message đến **WndProc** để xử lý với **IDC\_RECEIVE**

#### b. Phía Server

**int receiveMessage(SOCKET recSocket);**

Nhận tin nhắn từ Client và dựa là FLAG để xử lý. Ở đây FLAG: **REGISTER\_SUCCESS**, **REGISTER\_FAILURE**, **LOGIN\_SUCCESS**, **LOGIN\_FAILURE** ứng với từng loại để xử lý. Sau đó dùng phương thức **sendMessage**.

**int sendMessage(ClientPacket\* client, WCHAR\* message, int len);**

Gửi gói tin phản hồi đăng ký thành công hay thất bại đến client.

### 2. Chức năng thay đổi password

#### a. Phía client

**LRESULT CALLBACK ChangePasswordProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)**

Bắt message khi người dùng nhấn vào button “CHANGE PASSWORD” thông qua **WM\_COMMAND** với ID của button là **IDC\_ENTER**. Lấy oldpassword và newpassword, confirmpassword ở 3 ô edittext “edOldPass” – “edNewPass” – “edConfirmPass” kiểm tra hợp lệ và tạo 1 gói tin yêu cầu đăng ký gửi lên server với FLAG là **CHANG\_PASSWORD** bằng hàm **sendMessage**.

### **LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)**

Nhận kết quả trả về thông qua **receiveMessage** và gửi message đến giao diện với **WM\_COMMAND** với ID là **IDC\_RECEIVE**. wParam là con trỏ tới tin nhắn nhận được. Kết quả trả về có thể là: **CHANGE\_PASSWORD\_SUCCESS**, **CHANGE\_PASSWORD\_FAILURE**.

#### **int sendMessage(WCHAR\* message, int len);**

Gửi gói tin đến server. Được yêu cầu từ **ChangePasswordProc** thông qua **GlobalClient**.

#### **int receiveMessage();**

Nhận gói tin từ server. Sau đó gửi message đến **WndProc** để xử lý với **IDC\_RECEIVE**

### **b. Phía Server**

#### **int receiveMessage(SOCKET recSocket);**

Nhận tin nhắn từ Client và dựa là FLAG để xử lý. Ở đây FLAG: **CHANGE\_PASSOWRD** để xử lý. Sau đó dùng phương thức **sendMessage**.

#### **int sendMessage(ClientPacket\* client, WCHAR\* message, int len);**

Gửi gói tin phản hồi thay đổi thành công hay thất bại đến client.

## **3. Chức năng kiểm tra thông tin người dùng**

### **a. Phía client**

### **LRESULT CALLBACK CheckUserProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)**

Bắt message khi người dùng nhấn vào button “CHECK USER” thông qua **WM\_COMMAND** với ID của button là **IDC\_ENTER**. Lấy username ở ô edittext “edUsername” đồng thời lấy optoin ở các radiobutton **IDC\_FIND**, **IDC\_ONLINE**, **IDC\_SHOWNAME**, **IDC\_SHOWDATE**, **IDC\_SHOWNOTE**, **IDC\_SHOWALL**, kiểm tra hợp lệ và tạo 1 gói tin yêu cầu đăng ký gửi lên server với FLAG là **CHECK\_USER\_[OPTION\*]**, bằng hàm **sendMessage**.

### **LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)**

Nhận kết quả trả về thông qua **receiverMessage** và gửi message đến giao diện với **WM\_COMMAND** với ID là **IDC\_RECEIVE**. wParam là con trỏ tới tin nhắn nhận được. Kết quả trả về có thể là: thành công hay thất bại.

#### **int sendMessage(WCHAR\* message, int len);**

Gửi gói tin đến server. Được yêu cầu từ **CheckUserProc** thông qua **GlobalClient**.



**int receiveMessage();**

Nhận gói tin từ server. Sau đó gửi message đến **WndProc** để xử lý với **IDC\_RECEIVE**

#### **b. Phía Server**

**int receiveMessage(SOCKET recSocket);**

Nhận tin nhắn từ Client và dựa là FLAG để xử lý. Ở đây FLAG: **CHECK\_USER\_[OPTION\*]** để xử lý. Sau đó dùng phương thức **sendMessage**.

**int sendMessage(ClientPacket\* client, WCHAR\* message, int len);**

Gửi gói tin phản hồi thông tin thành công hay thất bại đến client.

### **4. Chức năng thay đổi thông tin người dùng**

#### **a. Phía client**

**LRESULT CALLBACK SetupInforProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)**

Bắt message khi người dùng nhấn vào button “SETUP INFORMATION” thông qua **WM\_COMMAND** với ID của button là **IDC\_ENTER**. Lấy infor ở ô edittext “edEnter” đồng thời lấy optoin ở các radiobutton **IDC\_NAME**, **IDC\_DATE**, **IDC\_NOTE**, kiểm tra hợp lệ và tạo 1 gói tin yêu cầu đăng ký gửi lên server với FLAG là **SETUP\_INFO\_[OPTION\*]**, bằng hàm **sendMessage**.

**LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)**

Nhận kết quả trả về thông qua **receiveMessage** và gửi message đến giao diện với **WM\_COMMAND** với ID là **IDC\_RECEIVE**. wParam là con trỏ tới tin nhắn nhận được. Kết quả trả về có thể là: thành công hay thất bại.

**int sendMessage(WCHAR\* message, int len);**

Gửi gói tin đến server. Được yêu cầu từ **SetupInforProc** thông qua **GlobalClient**.

**int receiveMessage();**

Nhận gói tin từ server. Sau đó gửi message đến **WndProc** để xử lý với **IDC\_RECEIVE**

#### **b. Phía Server**

**int recClient(SOCKET recSocket);**

Nhận tin nhắn từ Client và dựa là FLAG để xử lý. Ở đây FLAG: **SETUP\_INFO\_[OPTION\*]** để xử lý. Sau đó dùng phương thức **sendMessage**.

**int sendMessage(ClientPacket\* client, WCHAR\* message, int len);**

Gửi gói tin phản hồi thông tin đến client.

## 5. Chức năng gửi tập tin

### a. Phía client

**LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)**

Bắt message khi người dùng nhấn vào button “UPLOAD” thông qua **WM\_COMMAND** với ID của button là **IDC\_UPLOAD**. Lấy infor ở các radiobutton **IDC\_NORMAL**, **IDC\_CHANGENAME**, **IDC\_MULTI**, kiểm tra hợp lệ và mở cửa sổ mới thích hợp.

**LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)**

1. Nhận gói tin yêu cầu gửi file thông qua **receiveMessage** và gửi message đến giao diện với **WM\_COMMAND** và ID là **IDC\_RECEIVE**. **wParam** là con trỏ tới tin nhắn nhận được. Kết quả trả về là FLAG: **SEND\_FILE\_ACCEPT** hoặc **SEND\_FILE\_CANCEL** nếu đã có tên như vậy tồn tại trên server.
2. Nhận gói tin **SEND\_FILE\_ACCEPT** thì bắt đầu gửi phần đầu tiên của gói tin **FILE\_DATA** đến server.
3. Nhận gói tin **SEND\_FILE\_CANCEL** thì gọi hàm đóng file và dừng việc gửi gói tin.
4. Nhận gói tin **CONTINUE** thì gửi phần tiếp theo của gói tin **FILE\_DATA** đến server. Dùng hàm **onSendFile** để tiếp tục đọc tập tin.
5. Nhận gói tin **STOP** thì đã gửi xong file và dừng việc gửi file hiện tại lại. Xem trong **listMultiFile** có file không, nếu có tiếp tục gửi file trong **listMultiFile**.

**int sendMessage(WCHAR\* message, int len);**

Gửi gói tin đến server. Được yêu cầu từ **WndProc**, **ChangeNameUploadProc**, **MultiUploadProc** thông qua **GlobalClient**.

**int receiveMessage();**

Nhận gói tin từ server. Sau đó gửi message đến **WndProc** để xử lý với **IDC\_RECEIVE**

**BOOL myCreateOpenFile(HWND hwnd, WCHAR\* filename);**

Mở hộp thoại mở file để người dùng chọn file cần gửi.

### b. Phía Server

**int receiveMessage(SOCKET recSocket);**

Nhận tin nhắn từ Client và dựa là FLAG để xử lý.

1. Nhận được gói tin **SEND\_FILE** thì kiểm tra file có tồn tại hay không để gửi gói **SEND\_FILE\_ACCEPT** hoặc **SEND\_FILE\_CANCEL**.
2. Gói tin có FLAG **FILE\_DATA**. Đọc gói tin và ghi dữ liệu nhận được vào file. Nếu ghi đủ thì đóng file và gửi gói tin có FLAG **STOP** đến client để dừng gửi, ngược lại gửi gói **CONTINUE** để yêu cầu tiếp tục gửi.

**int sendMessage(ClientPacket\* client, WCHAR\* message, int len);**

Gửi gói tin phản hồi thông tin đến client.

## 6. Chức năng tải tập tin

### a. Phía client

**LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)**

Bắt message khi người dùng nhấn vào button “DOWNLOAD” thông qua **WM\_COMMAND** với ID của button là **IDC\_DOWNLOAD**. Lấy infor ở các radiobutton **IDC\_NORMAL**, **IDC\_CHANGENAME**, **IDC\_MULTI**, kiểm tra hợp lệ và mở cửa sổ mới thích hợp.

**LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)**

1. Nhận gói tin yêu cầu gửi file thông qua **receiveMessage** và gửi message đến giao diện với **WM\_COMMAND** và ID là **IDC\_RECEIVE**. **wParam** là con trỏ tới tin nhắn nhận được. Kết quả trả về là FLAG: **SEND\_FILE** thì mở chọn thư mục lưu nếu được thì gửi lại gói **SEND\_FILE\_ACCEPT** ngược lại **SEND\_FILE\_CANCEL**

2. Gói tin có FLAG **FILE\_DATA**. Đọc gói tin và ghi dữ liệu nhận được vào file. Nếu ghi đủ thì đóng file và gửi gói tin có FLAG **STOP** đến server để dừng gửi, ngược lại gửi gói **CONTINUE** để yêu cầu tiếp tục gửi.

3. Nếu chọn chức năng multi file thì gói tin gửi lần đầu sẽ là **DOWN\_MULTI** chứa tên tất cả các file muốn download.

**int sendMessage(WCHAR\* message, int len);**

Gửi gói tin đến server. Được yêu cầu từ **WndProc**, **NormalDownloadProc**, **ChangeNameDownloadProc**, **MultiDownloadProc** thông qua **GlobalClient**.

**int receiveMessage();**

Nhận gói tin từ server. Sau đó gửi message đến **WndProc** để xử lý với **IDC\_RECEIVE**

**BOOL myCreateSaveFile(HWND hwnd, WCHAR\* filename);**

Mở hộp thoại lưu file để người dùng chọn nơi file cần lưu.

### b. Phía Server

**int receiveMessage(SOCKET recSocket);**

Nhận tin nhắn từ Client và dựa là FLAG để xử lý.

1. Nhận gói tin **SEND\_FILE\_ACCEPT** thì bắt đầu gửi phần đầu tiên của gói tin đến user bên kia.

2. Nhận gói tin **SEND\_FILE\_CANCEL** thì gọi hàm đóng file và dừng việc gửi gói tin.

3. Nhận gói tin **CONTINUE** thì gửi phần tiếp theo của gói tin đến client Dùng hàm **onSendFile** để tiếp tục đọc tập tin.

4. Nhận gói tin **STOP** thì đã gửi xong file và dừng việc gửi file hiện tại lại. Xem trong **listMultiFile** có file không, nếu có tiếp tục gửi file trong **listMultiFile**.

5. Nếu gói tin **DOWN\_MULTI** thì đọc tên các file rồi thêm vào **listMultiFile**.

**int sendMessage(ClientPacket\* client, WCHAR\* message, int len);**

Gửi gói tin phản hồi thông tin đến client.

## 7. Chức năng chat với người dùng khác

### a. Phía client

**LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)**

1. Bắt message khi người dùng nhấn vào button “CHAT” thông qua **WM\_COMMAND** với ID của button là **IDC\_CHAT**. Lấy username ở ô edittext “edUsernae” kiểm tra hợp lệ và mở cửa sổ mới thích hợp, đồng thời gửi **CREATE\_CHAT**.

2. Nhận kết quả tạo chat trả về thông qua **receiveMessage** và gửi message đến giao diện với **WM\_COMMAND** và ID là **IDC\_RECEIVE**. **wParam** là con trỏ tới tin nhắn nhận được. Kết quả trả về có thể là: **CREATE\_CHAT\_SUCCESS** (tạo chat 1-1 thành công, người dùng có online) hoặc **CREATE\_CHAT\_FAILURE** (người dùng không tồn tại hoặc offline) -> hiện thông báo người dùng. Đối với thành công nếu chưa có cửa sổ chat thì tạo cửa sổ chat mới, còn có rồi thì hiện thị tin nhắn.

3. Nhận tin nhắn chat thông qua **receiveMessage** và gửi message đến giao diện với **WM\_COMMAND** và ID là **IDC\_RECEIVE**. **wParam** là con trỏ tới tin nhắn nhận được. Kết quả trả về là gói tin có FLAG: **CHAT**. Tìm đến ChatBox nhận tin nhắn và hiển thị tin nhắn.

4. Nhận tin có user bên kia đã ngừng chat thông qua **receiveMessage** và gửi message đến giao diện với **WM\_COMMAND** và ID là **IDC\_RECEIVE**.

**wParam** là con trỏ tới tin nhắn nhận được. Kết quả trả về là gói tin có FLAG: **END\_CHAT** và hiển thị user bên kia đã ngừng chat.

**LRESULT CALLBACK ChatBoxProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)**

1. Bắt message khi người dùng nhấn vào button “SEND” thông qua **WM\_COMMAND** với ID của button là **IDC\_SEND**. Lấy nội dung ô edittext – nội dung tin nhắn, kiểm tra hợp lệ và tạo 1 gói tin yêu cầu gửi tin nhắn đến user bên kia và gửi lên server với FLAG là **CHAT** bằng hàm **sendMessage**.

2. Bắt sự kiện **WM\_DESTROY** khi người dùng đóng chatbox. Gửi gói tin ngừng chat cho user bên kia thông qua **sendMessage** với gói tin có FLAG **END\_CHAT**

**int sendMessage(WCHAR\* message, int len);**

Gửi gói tin đến server. Được yêu cầu từ **WndProc**, **ChatBoxProc** thông qua **GlobalClient**.

**int receiveMessage();**

Nhận gói tin từ server. Sau đó gửi message đến **WndProc** để xử lý với **IDC\_RECEIVE**

## **b. Phía Server**

### **int receiveMessage(SOCKET recSocket);**

1. Nhận gói tin yêu cầu tạo chat. Kiểm tra user cần chat có online hay tồn tại hay không. Nếu có thì gửi tin nhắn có FLAG **CREATE\_CHAT\_SUCCESS** về client ngược lại gửi tin có FLAG **CREATE\_CHAT\_FAILURE** thông qua hàm **sendMessage**.
2. Nhận gói tin chat giữa 2 client. Dùng hàm **sendMessage** để gửi gói tin tin nhắn đến user bên kia.
3. Nhận gói tin kết thúc chat của 1 user (**END\_CHAT**) và gửi đến user còn lại

### **int sendMessage(ClientPacket\* client, WCHAR\* message, int len);**

Gửi gói tin phản hồi thông tin đến client.

## **PHẦN V: ĐÁNH GIÁ ĐỒ ÁN**

Hoàn tất 100% yêu cầu cơ bản và nâng cao.