**Mục lục**

[**1.** **Tổng quan đề tài** 2](#_Toc44683956)

[**2.** **Cấu trúc chương trình** 2](#_Toc44683957)

[**2.1.** **CLIENT(empty project, WinMain)** 2](#_Toc44683958)

[**2.2.** **SERVER(empty project, main)** 3](#_Toc44683959)

[**3.** **Khuôn dạng gói tin, cách truyền tin** 3](#_Toc44683960)

[**3.1.** **Khuôn dạng gói tin** 3](#_Toc44683961)

[**3.2.** **Cách truyền tin** 4](#_Toc44683962)

[**4.** **Use case** 4](#_Toc44683963)

[**4.1.** **Use case kết nối tới server** 4](#_Toc44683964)

[**4.2.** **Use case yêu cầu chuyển tiếp file** 4](#_Toc44683965)

[**4.3.** **Use case yêu cầu tìm kiếm file** 5](#_Toc44683966)

[**5.** **Biểu đồ hoạt động** 5](#_Toc44683967)

[**5.1.** **Kết nối tới server** 6](#_Toc44683968)

[**5.2.** **Chuyển tiếp file** 6](#_Toc44683969)

[**5.3.** **Tìm kiếm file** 7](#_Toc44683970)

[**6.** **Giao diện chương trình** 8](#_Toc44683971)

[**6.1.** **SERVER** 8](#_Toc44683972)

[**6.2.** **CLIENT** 8](#_Toc44683973)

[**Tài Liệu Tham Khảo** 10](#_Toc44683974)

# **Tổng quan đề tài**

1. **Tên đề tài**: đề tài 7, xây dựng ứng dụng chia sẻ file 1

* Server có các chức năng:
* Xử lý yêu cầu kết nối của client, trả về ID cho client đó
* Nhận yêu cầu tìm kiếm file theo tên của client A nào đó
* Gửi lệnh tìm kiếm tới các client khác đang kết nối
* Gửi lại cho client A danh sách các client có file mà client A tìm kiếm
* Client có các chức năng:
* Gửi yêu cầu tìm kiếm một file lên server
* Nhận danh sách các client có file mà client đó yêu cầu tìm kiếm
* Lựa chọn ra một client trong số đó để yêu cầu download file
* Yêu cầu server chuyển tiếp file đến client khác
* Thực hiện tìm kiếm file theo yêu cầu của client khác(sẽ tìm kiếm file trong thư mục Data nằm cùng thư mục với file .exe của client). Cụ thể, client sẽ tìm kiếm đệ quy vào thư mục Data(tìm kiếm tất cả các file nằm trong Data và các thư mục con của Data)

1. **Ngôn ngữ sử dụng**: C++(winsock, Win32)
2. **Trình biên dịch sử dụng**: Visual Studio 2015
3. **Phân chia công việc**

* Phạm Hồng Phúc
* Xây dựng khuôn dạng gói tin, cách truyền tin, cách truyền file
* Tìm hiểu cách tạo giao diện
* Tìm hiểu các hàm tương tác với file
* Đỗ Viết Trí
* Tìm hiểu các hàm tương tác với file
* Tìm hiểu cách tạo giao diện

1. **Cấu trúc chương trình**

Chương trình xây dựng theo hướng cấu trúc, bao gồm hai project là CLIENT và SERVER

* 1. **CLIENT(empty project, WinMain)**
* Header Files
* CONST.h: chứa những hằng số phục vụ cho client
* Common.h: chứa những cấu trúc(struct) và tiêu đề hàm có chức năng chung phục vụ client
* InteractFile.h: chứa những cấu trúc và tiêu đề hàm có chức năng tương tác với file
* TCP\_SOCKET.h: chứa những cấu trúc và tiêu đề hàm có chức năng định nghĩa khuôn dạng gói tin, cách đóng gói và cách truyền tin trong chương trình
* Resource Files
* Common.cpp: định nghĩa các hàm được khai báo trong Common.h
* InteractFile.cpp: định nghĩa các hàm được khai báo trong InteractFile.h
* TCP\_SOCKET.cpp: định nghĩa các hàm được khai báo trong TCP\_SOCKET.h
* Source Files
* WinMain.cpp: chứa hàm WinMain của chương trình phía client
  1. **SERVER(empty project, main)**
* Header Files
* CONST.h: chứa những hằng số phục vụ cho server
* Common.h: chứa những cấu và tiêu đề hàm có chức năng chung phục vụ server
* TCP\_SOCKET.h: chứa những cấu trúc và tiêu đề có chức năng định nghĩa khuôn dạng gói tin, cách đóng gói và cách truyền tim trong chương trình
* Resource Files
* Common.cpp: định nghĩa các hàm được khai báo trong Common.h
* TCP\_SOCKET.cpp: định nghĩa các hàm được khai báo trong TCP\_SOCKET.h
* Source Files
* SERVER.cpp: chứa hàm main của chương trình phía server

1. **Khuôn dạng gói tin, cách truyền tin**
   1. **Khuôn dạng gói tin**

Định dạng struct, gồm 5 trường

struct Message {

int type;

char fileName[BUFF\_SIZE];

char ID[BUFF\_SIZE];

char data[BUFF\_SIZE + 1];

int dataLen;

};

(BUFF\_SIZE là 10240)

* Type: xác định chức năng mà client và server phải thực hiện

**SERVER**

* 100: kết nối thành công, gửi ID lại cho client
* 110: gửi danh sách các client đang kết nối
* 111: gửi danh sách các client có file yêu cầu về client đang tìm kiếm file đấy
* 112: tải từng gói dữ liệu của file tìm kiếm về client đi kèm kích thức của gói dữ liệu đó
* 1120: giống 112 nhưng là dấu hiệu kết thúc tải file
* 120: gửi yêu cầu tìm kiếm file từ một client đến các client còn lại
* 121: gửi yêu cầu tải file lên server đến client được chỉ định trong chức năng tìm kiếm file
* 200: yêu cầu chuyển tiếp file về client
* 201: tải từng gói dữ liệu của file chuyển tiếp về client đi kèm với kích thước của gói dữ liệu đó
* 2010: giống 201 nhưng là dấu hiệu kết thúc tải file về client
* 202: ID do client gửi lên có thể kết nối và được phép chuyển tiếp
* 203: ID do client gửi lên không thể kết nối hoặc bị từ chối chuyển tiếp

**CLIENT**

* 300**:**  yêu cầu kết nối và yêu cầu server gửi ID
* 310: gửi yêu cầu tìm kiếm file
* 311: tải từng gói dữ liệu file từ client được chỉ định bởi client khác lên server đi kèm với kích thức của gói dữ liệu đó
* 3110: giống 311 nhưng là dấu hiệu kết thúc tải file lên server
* 312: chỉ định một client để tải file đang tìm kiếm
* 320: không tìm thấy tên file mà server yêu cầu
* 321: tìm thấy tên file mà server yêu cầu
* 400: gửi ID và fileName để yêu cầu chuyển tiếp file đến client tương ứng với ID được gửi
* 401: gửi lần lượt từng gói dữ liệu file chuyển tiếp lên server đi kèm với kích thước của gói dữ liệu đó
* 4010: giống 401 nhưng là dấu hiệu kết thúc tải file lên server
* 410: không cho chuyển tiếp file về client
* 411: cho phép chuyển tiếp file về client
* fileName: chứa thông tin tên file
* ID: chứa thông tin ID của client(chưa chắc là ID của client gửi gói tin đó)
* data: chứa dữ liệu(có thể là dữ liệu file cần truyền đi)
* dataLen: kích thức của trường data
  1. **Cách truyền tin**

Sử dụng tcp socket và kỹ thuật truyền dòng để truyền tin giữa client và server

1. **Use case**
   1. **Use case kết nối tới server**
2. tóm tắt: người dùng sử dụng use case này để kết nối tới server và nhận ID từ server
3. Dòng sự kiện

* Hệ thống hiển thị giao diện ban đầu
* Người dùng thực hiện kết nối bằng việc nhập thông tin IP và số hiệu cổng của server, sau đó bấm Connect
* Nếu kết nối thành công , server sẽ gửi trả ID của người dùng
* Nếu không thành công hệ thống sẽ báo lỗi, yêu cầu kết nối lại
  1. **Use case yêu cầu chuyển tiếp file**

1. Tóm tắt: người dùng sử dụng use case để chuyển tiếp file đến client khác
2. Dòng sự kiện

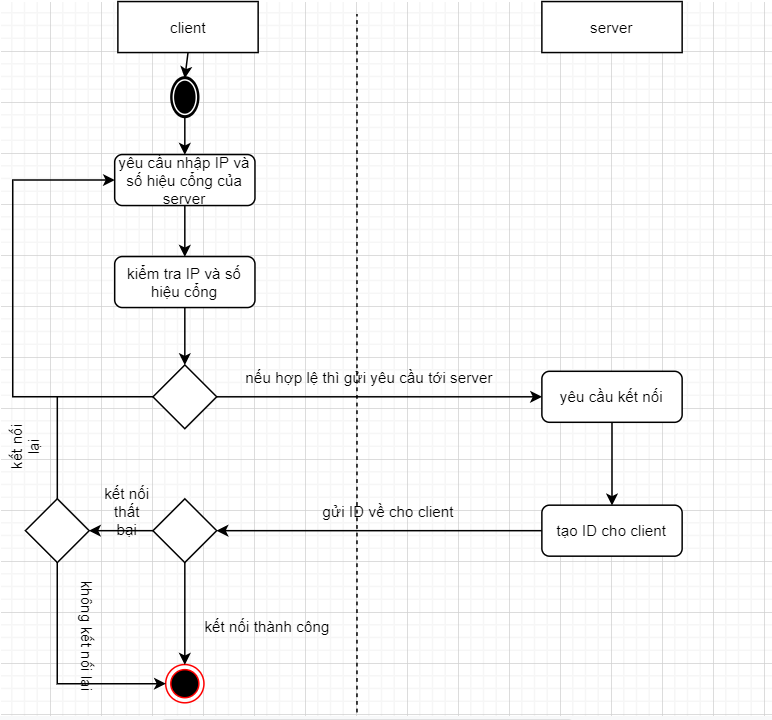
* Use case bắt đầu khi người dùng bấm browse chọn file cần chuyển tiếp và bấm forward để chuyển tiếp file
* Hệ thống sẽ yêu cầu nhập ID của người muốn chuyển tiếp
* Sau khi nhập ID, bấm forward một lần nữa
* Server sẽ tìm kiếm client được client kia yêu cầu, nếu tìm thấy server sẽ yêu cầu được chuyển tiếp file về client đó, nếu không tìm thấy thông báo về cho client yêu cầu chuyển tiếp, kết thúc use case
* Nếu yêu cầu chuyển tiếp được chấp nhận, bắt đầu chuyển tiếp file về client đó, nếu không chấp nhận thì thông báo về cho client yêu cầu chuyển tiếp, kết thúc use case

1. Luồng ngoại lệ: nếu client được yêu cầu chuyển tiếp chính là client yêu cầu chuyển tiếp thì báo lỗi cho người dùng, kết thúc use case
   1. **Use case yêu cầu tìm kiếm file**
2. Tóm tắt: người dùng sử dụng use case để tìm kiếm file ở các client khác
3. Dòng sự kiện

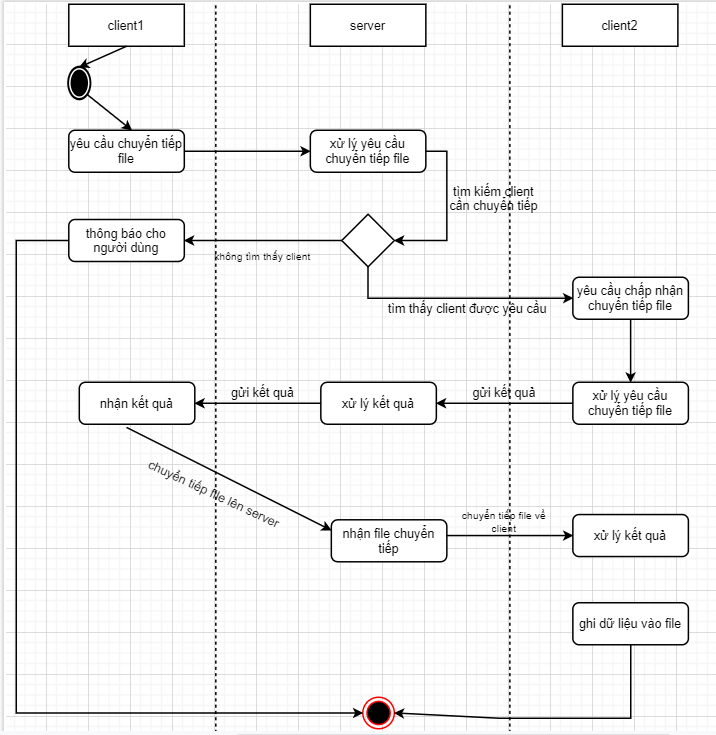
* Use case bắt đầu khi người dùng nhập tên file cần chuyển tiếp và bấm nút search
* Server sẽ gửi yêu cầu tìm kiếm này tới các client khác
* Nếu tìm thấy, server sẽ gửi trả client yêu cầu tìm kiếm danh sách các client có thể tải file cần tìm kiếm, client sẽ chọn một chient trong số đó để yêu cầu tải file
* Nếu không tìm thấy, server sẽ thông báo cho client, client sẽ hiện thông báo cho người dùng

1. **Biểu đồ hoạt động**

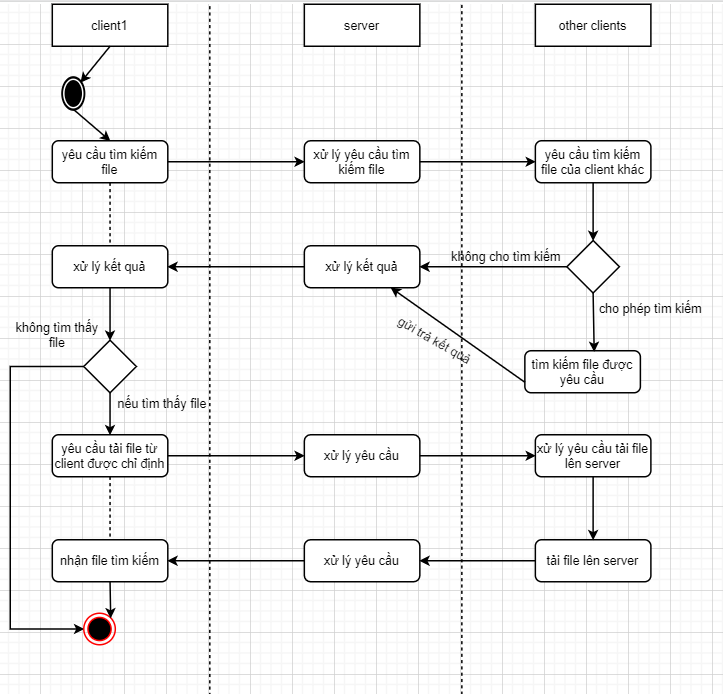
## **Kết nối tới server**



* 1. **Chuyển tiếp file**



* 1. **Tìm kiếm file**



1. **Giao diện chương trình**
   1. **SERVER**

Server là console, không có giao diện

* 1. **CLIENT**

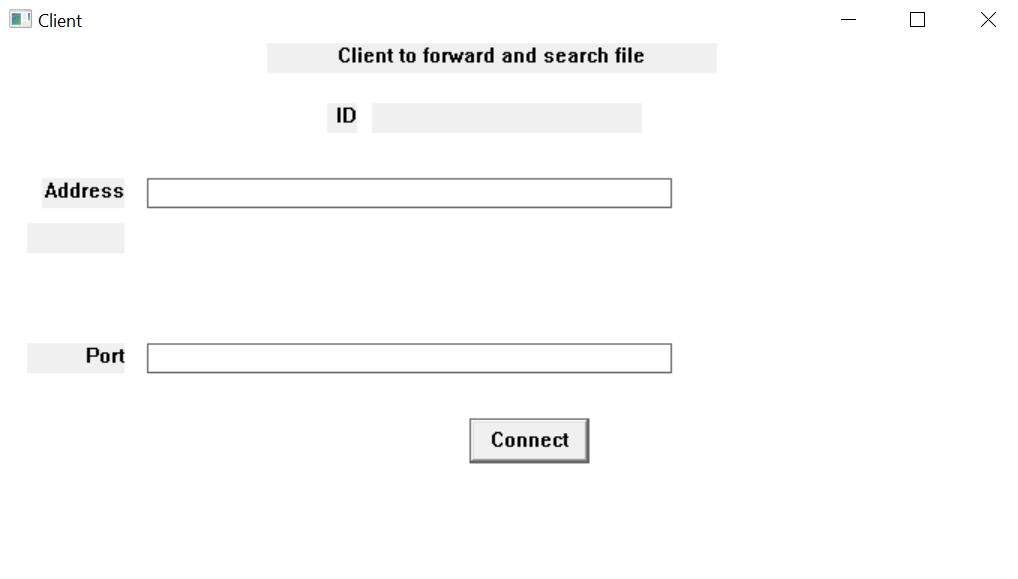


Figure 1: giao diện khởi tạo

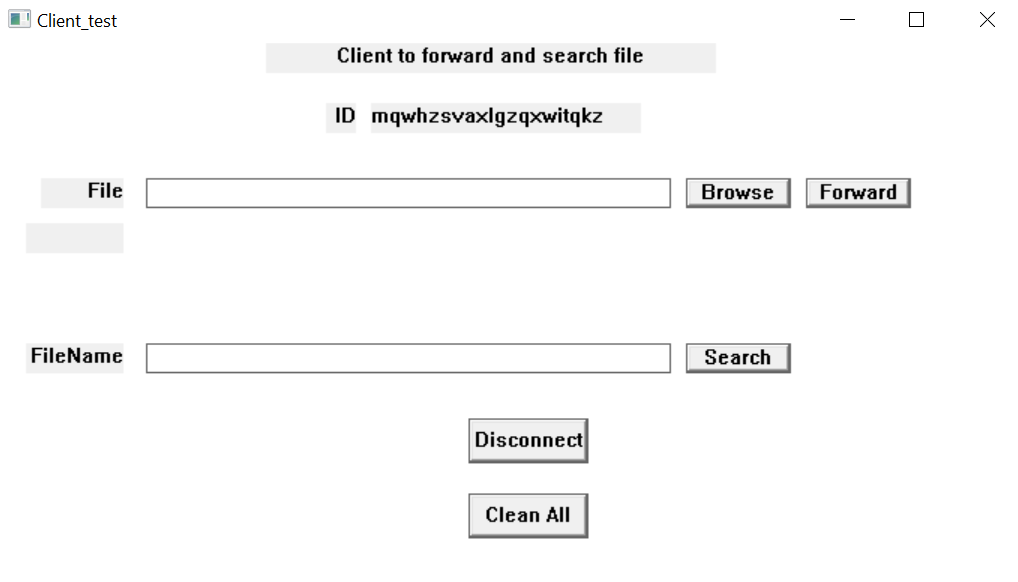


Figure 2: giao diện sau khi kết nối thành công

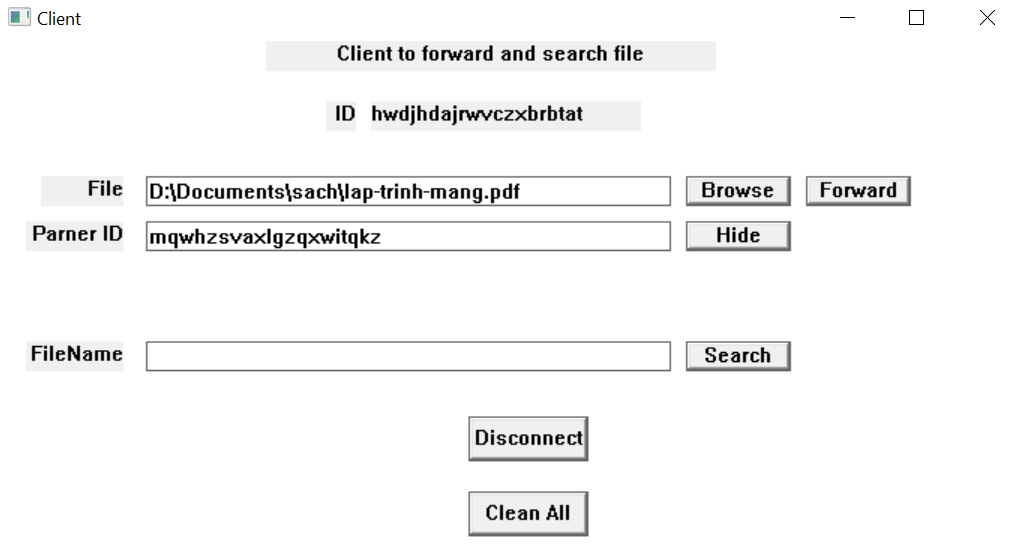


Figure 3: giao diện khi muốn chuyển tiếp file



Figure 4: giao diện danh sách client có file tìm kiếm

# **Tài Liệu Tham Khảo**

* <https://docs.microsoft.com/en-us/windows/win32/controls/user-controls-intro>
* <https://stackoverflow.com/questions/7598067/how-to-create-a-windows-style-textbox-in-a-c-win32-application>
* <https://stackoverflow.com/questions/8520560/get-a-file-name-from-a-path>
* <https://stackoverflow.com/questions/2886609/how-to-make-multiple-windows-using-win32-api#:~:text=5%20Answers&text=You%20can%20hit%20CreateWindow(),1st%20one%20if%20you%20want>.
* <https://docs.microsoft.com/en-us/windows/win32/api/winuser/nf-winuser-messagebox>
* <https://stackoverflow.com/questions/42046666/test-project-exe-has-triggered-a-breakpoint>
* <https://stackoverflow.com/questions/42438135/c-winapi-listbox-getting-selected-item-using-lb-getsel-lb-getcursel>
* <https://docs.microsoft.com/en-us/windows/win32/learnwin32/closing-the-window>