



BÁO CÁO ĐỒ ÁN THỰC HÀNH 1:

LẬP TRÌNH SOCKET

Môn: Mạng máy tính

Đề bài: Chat Conference

Sinh viên thực hiện:

Mã số sinh viên	Họ tên
1712441	Trần Đình Tôn Hiếu
1712266	Võ Trường An
1712645	Bùi Thị Cẩm Nhung

1. Đánh giá mức độ hoàn thành của đồ án:

- Đánh giá tổng thể: **95%**

- Chi tiết từng yêu cầu:

STT	Yêu cầu	Mức độ hoàn thành
1	Server cho phép nhiều client đăng nhập và thảo luận cùng lúc	100%
2	Client cần đăng ký một username và password để đăng nhập	100%
3	Kiểm tra username đã tồn tại hay chưa, nếu tồn tại thì phải chọn username mới	100%
4	Mọi thông điệp trao đổi của client phải được thể hiện trên màn hình của server và mọi client khác, dưới dạng “username: Thông điệp chat”.	100%
5	Có thông báo trên màn hình server và client khi một client thoát	100%
6	Nếu server thoát, các client đều phải nhận biết là ngắt kết nối	0%
7	Tính năng chat riêng tư	100%
8	Tính năng gửi file	100%

- Tính năng bổ sung thêm:

+ File được gửi có kích thước không giới hạn, tuy nhiên tốc độ sẽ chậm. Tốc độ lý tưởng là khi gửi file ≤ 5 MB.

+ Client có thể chọn nơi lưu file được nhận với giao diện trực quan.

+ Client có thể đặt tên cho file khi tải về tùy ý.

2. Kịch bản giao tiếp của chương trình:

- Server luôn chạy port 1234 để tiếp nhận kết nối của Client.
- Client sau khi kết nối với Server sẽ phải Log in bằng một tài khoản. Khi Log in sẽ gửi username, địa chỉ IP và Port kết nối cho Server → Server sẽ lưu thông tin user vừa kết nối vào userCache, đồng thời báo cho tất cả các Client biết vừa có user kết nối để cập nhật lại danh sách user đang online ở giao diện Client.
- Sau khi Client kết nối vào Server, Server sẽ chạy thêm 2 luồng ở Port 1235 và 1236 để xử lý request từ Client. Việc xử lý này tuy gọi là “đồng thời” nhưng thực chất là tuần tự, và khi đang xử lý request từ một Client thì Server sẽ bị “blocking” tạm thời.
- Người dùng có thể chọn gửi tin nhắn riêng đến một người dùng khác, hoặc là tạo nhóm chat.
- Khi tạo nhóm chat, Server sẽ nhận thông tin những thành viên có trong nhóm chat, tiến hành tạo nhóm và cấp phát ID của nhóm tự động và thông báo về cho tất cả Client có trong nhóm chat.
- Khi người dùng gửi tin nhắn lên nhóm chat, Server sẽ nhận được ID của nhóm chat, tiến hành tra cứu trong groupCache xem những thành viên nào có trong nhóm chat đó, rồi gửi tin nhắn đó đến tất cả những người có trong nhóm chat.
- Nếu gửi tin nhắn riêng, Server sẽ nhận ID của người nhận, tìm dữ liệu của người nhận trong userCache rồi gửi trực tiếp tin nhắn đó đến người nhận.
- Khi người dùng gửi file, Server sẽ lưu file đó vào trong máy, rồi gửi thông báo về tên file, kích thước file lên trên màn hình chat của các Client. Người dùng nào muốn tải file về thì sẽ nhấn đúp vào file, chọn địa chỉ lưu file, sau đó sẽ nhận file từ Server.

3. Môi trường lập trình và framework sử dụng:

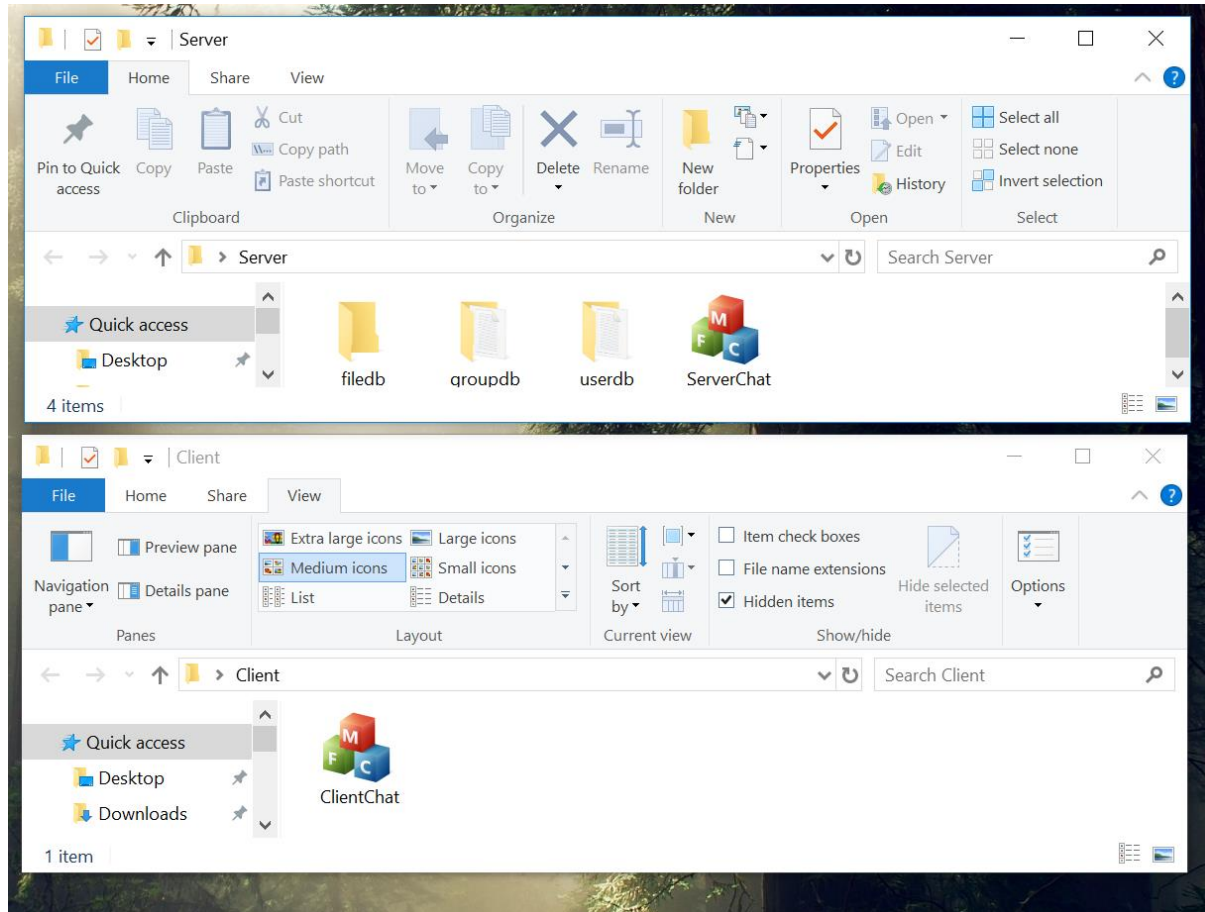
- Ở trong đề án này, môi trường lập trình mà nhóm sử dụng là Visual Studio.
- Nhóm sử dụng bộ thư viện MFC của C++ để làm đề án.
- Cụ thể hơn, ngoài việc dùng MFC để thiết kế giao diện Winform, nhóm còn sử dụng thư viện <afxsock.h> của MFC để sử dụng các Socket API có sẵn.

4. Bảng phân công công việc:

MSSV	Họ tên	Công việc thực hiện	Tỉ lệ đóng góp
1712441	Trần Đình Tôn Hiếu	<ul style="list-style-type: none"> - Viết các chức năng của Server: Đăng ký tài khoản, Log in, tạo nhóm, Gửi tin nhắn riêng, tin nhắn nhóm, gửi file và nhận file. - Viết báo cáo đề án. 	33.33%
1712266	Võ Trường An	<ul style="list-style-type: none"> - Thiết kế Server, cách các luồng của Server hoạt động. - Thiết kế CommonData thuận tiện cho việc gửi, nhận thông tin. - Giao diện Server, Ghi file Log, gửi danh sách User đang hoạt động. 	33.33%
1712645	Bùi Thị Cẩm Nhung	<ul style="list-style-type: none"> - Thiết kế giao diện Client, lập trình cho Client hoạt động theo từng sự kiện. - Viết các chức năng gửi và nhận thông điệp của Client. 	33.33%

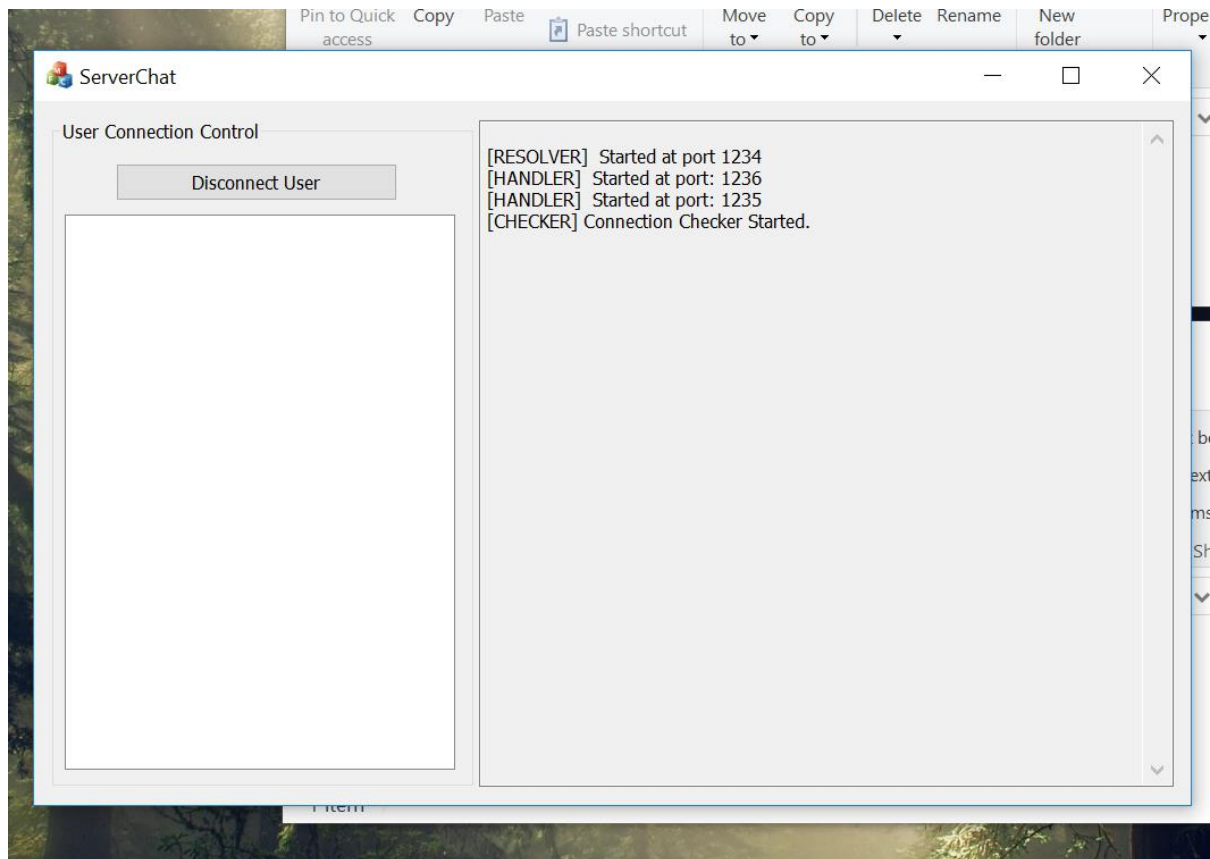
5. Hướng dẫn sử dụng:

Ta có 2 thư mục chứa chương trình như sau:



2 chương trình Client và Server

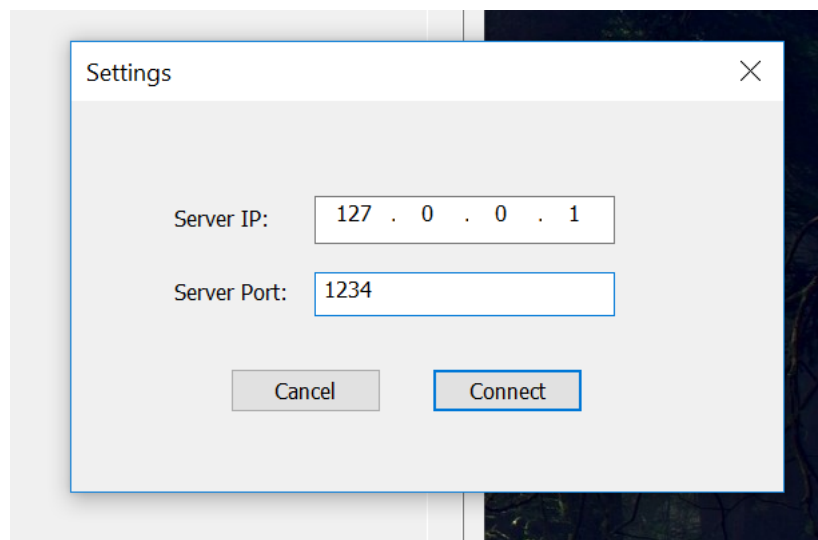
- Đầu tiên, người dùng chạy file ServerChat.exe để tạo một server điều khiển các client chat. Tại cửa sổ chương trình của Server, người dùng có thể nhìn thấy hầu hết hoạt động của Server và Client qua phần Log ở bên phải cửa sổ chương trình.
- Server sẽ tự động chạy trên các port 1234 nhằm nhận yêu cầu xử lý kết nối từ Client và các port 1235, 1236 để nhận các yêu cầu cũng như dữ liệu yêu cầu từ Client.



Giao diện Server lúc chạy

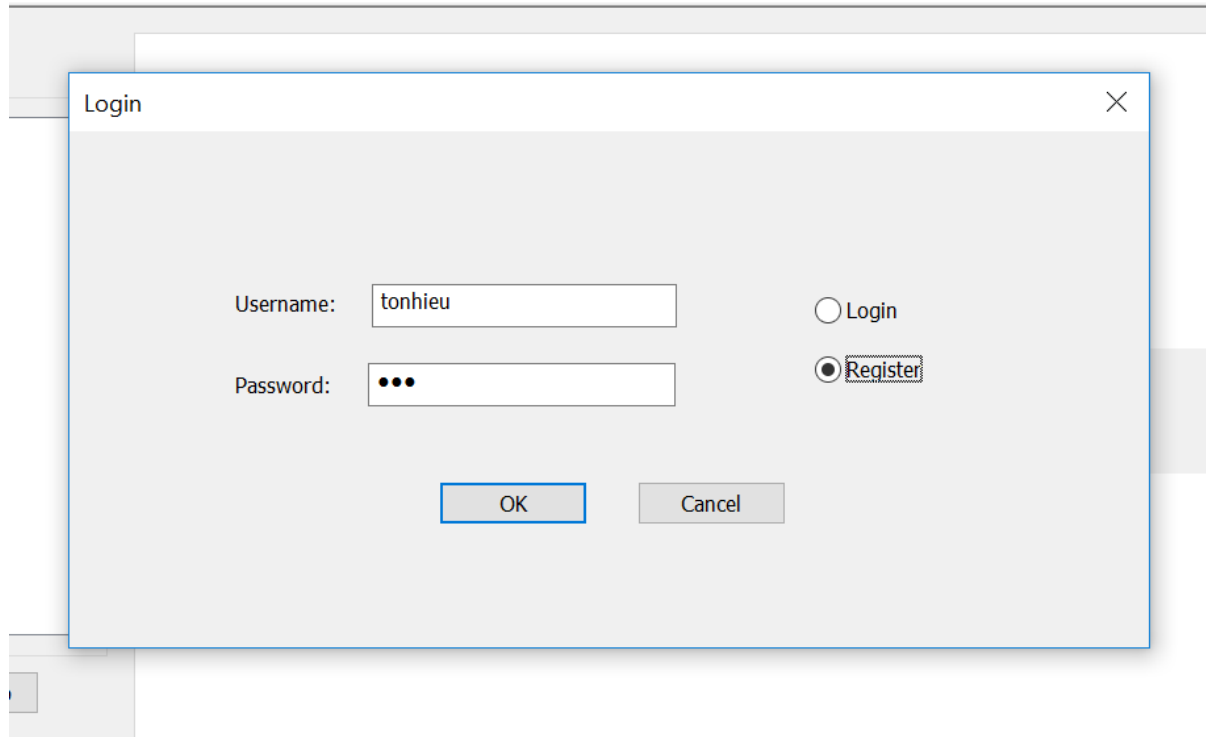
- Tại Server, nếu muốn ngắt kết nối với một người dùng nào đó, Server chọn người dùng tại danh sách User Connection Control và nhấn nút “Disconnect User”.

Người dùng ở các máy tính khác nhau sẽ bật Client.exe để chạy chương trình Client. Sau đó, nhập vào địa chỉ IP của server và port phân giải (mặc định là 1234).



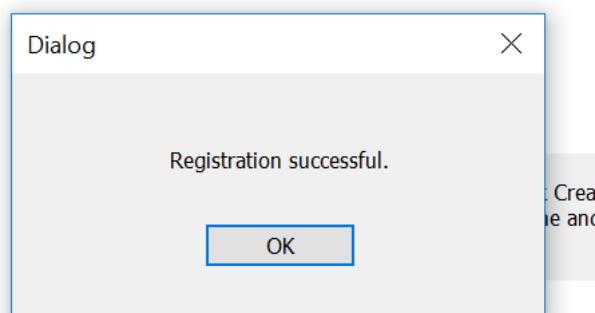
Địa chỉ IP: 127.0.0.1 (localhost) và Port 1234

- Khi kết nối đến chương trình server thành công, cửa sổ đăng nhập sẽ hiện lên để người dùng nhập vào tên đăng nhập và mật khẩu. Nếu như không thể kết nối đến chương trình Server, cửa sổ kết nối sẽ vẫn tiếp tục hiện lên để người dùng thực hiện việc kết nối lại.



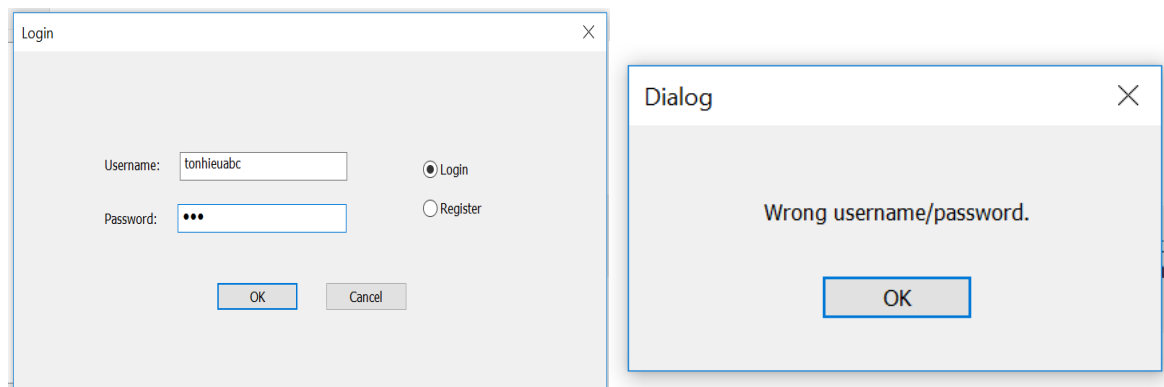
Đăng ký tài khoản mới

Nếu như chưa có tài khoản, người dùng cần đăng kí tên đăng nhập và mật khẩu bằng cách lựa chọn “Register”, sau đó nhập tên đăng nhập và mật khẩu của mình. Cuối cùng nhấn OK để hệ thống thực hiện việc đăng kí tài khoản. Nếu như việc đăng kí thành công, Client sẽ hiện lên thông báo “Registration successful”. Nếu như người dùng đã tồn tại thì không thể đăng kí được tài khoản.

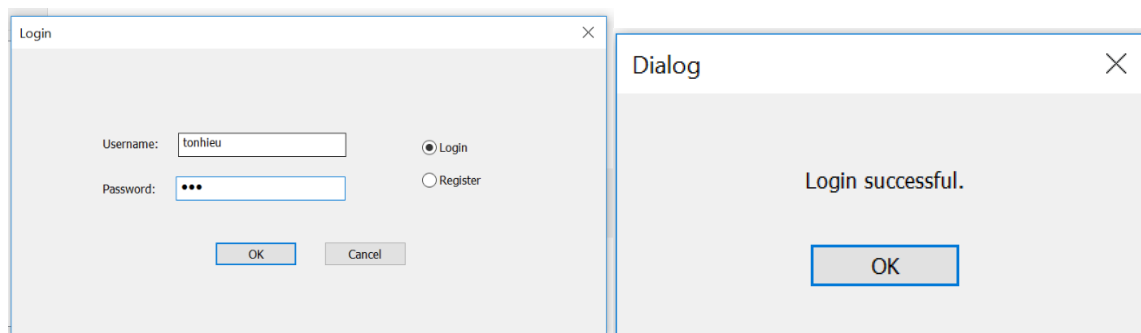


Đăng ký thành công

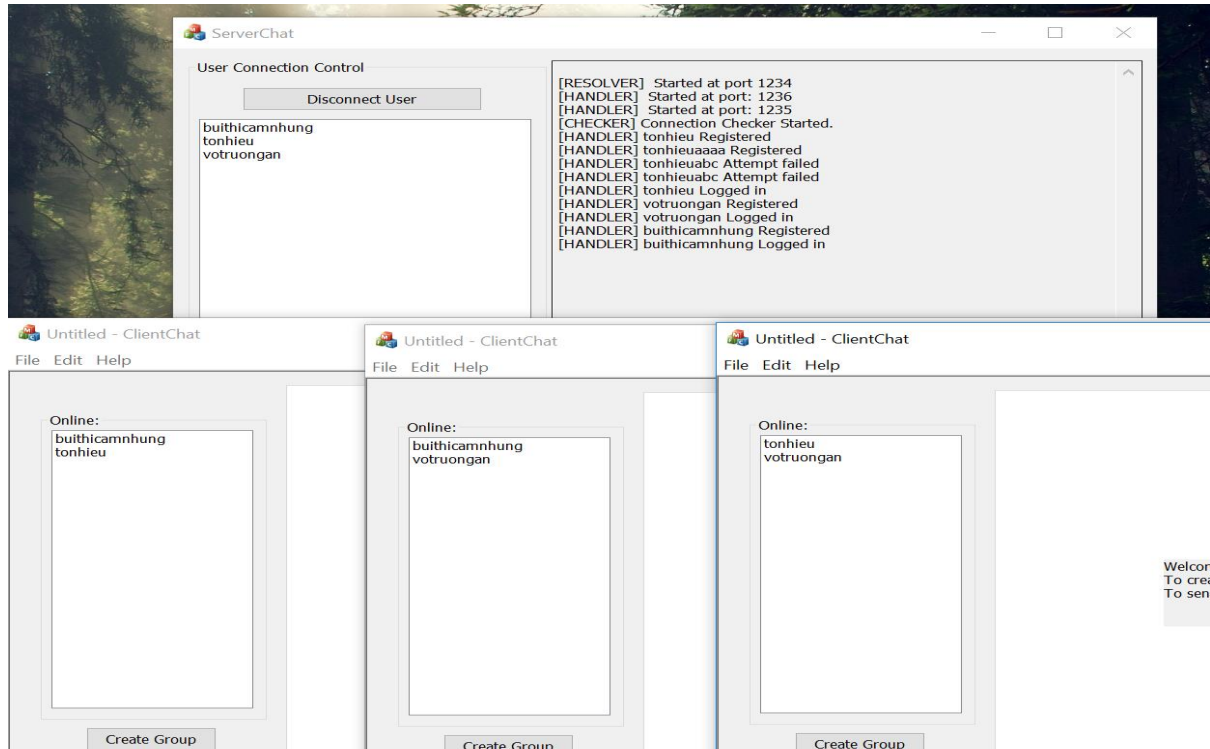
Để đăng nhập vào chương trình, người dùng nhập vào tên đăng nhập và mật khẩu, sau đó chọn lựa chọn “Login” và nhấn nút OK. Nếu đăng nhập thành công, Client sẽ hiện lên cửa sổ thông báo “Login successful”. Nếu như người dùng nhập vào tài khoản không tồn tại hoặc nhập sai mật khẩu, Client chỉ báo sai tên đăng nhập hoặc mật khẩu.



Đăng nhập thất bại vì sai Tài khoản

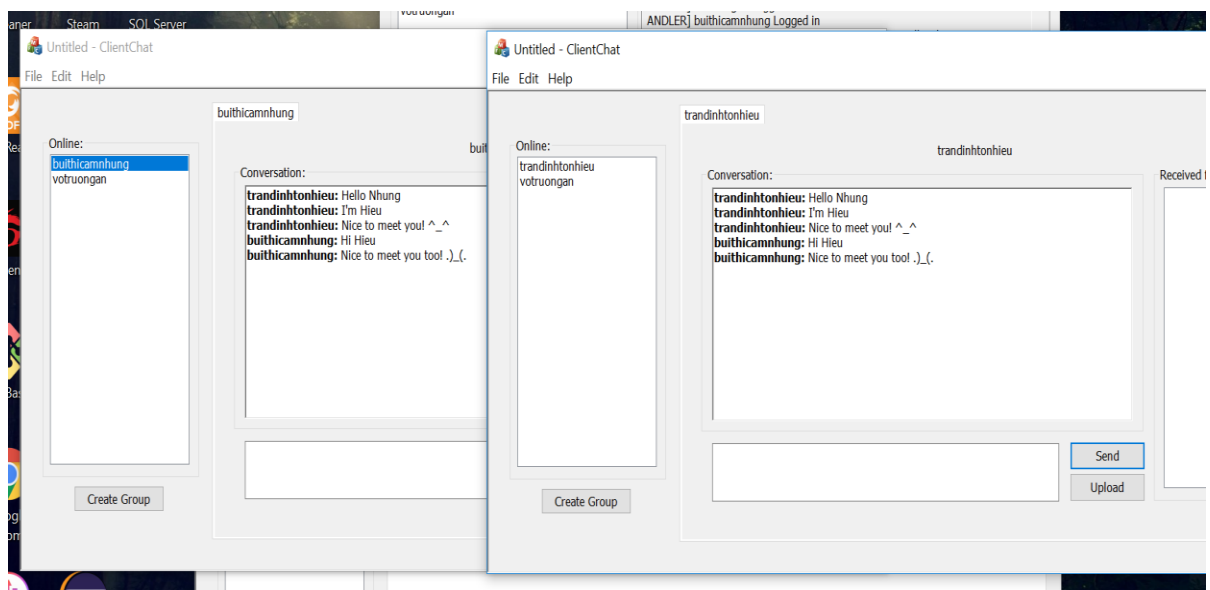


Đăng nhập thành công

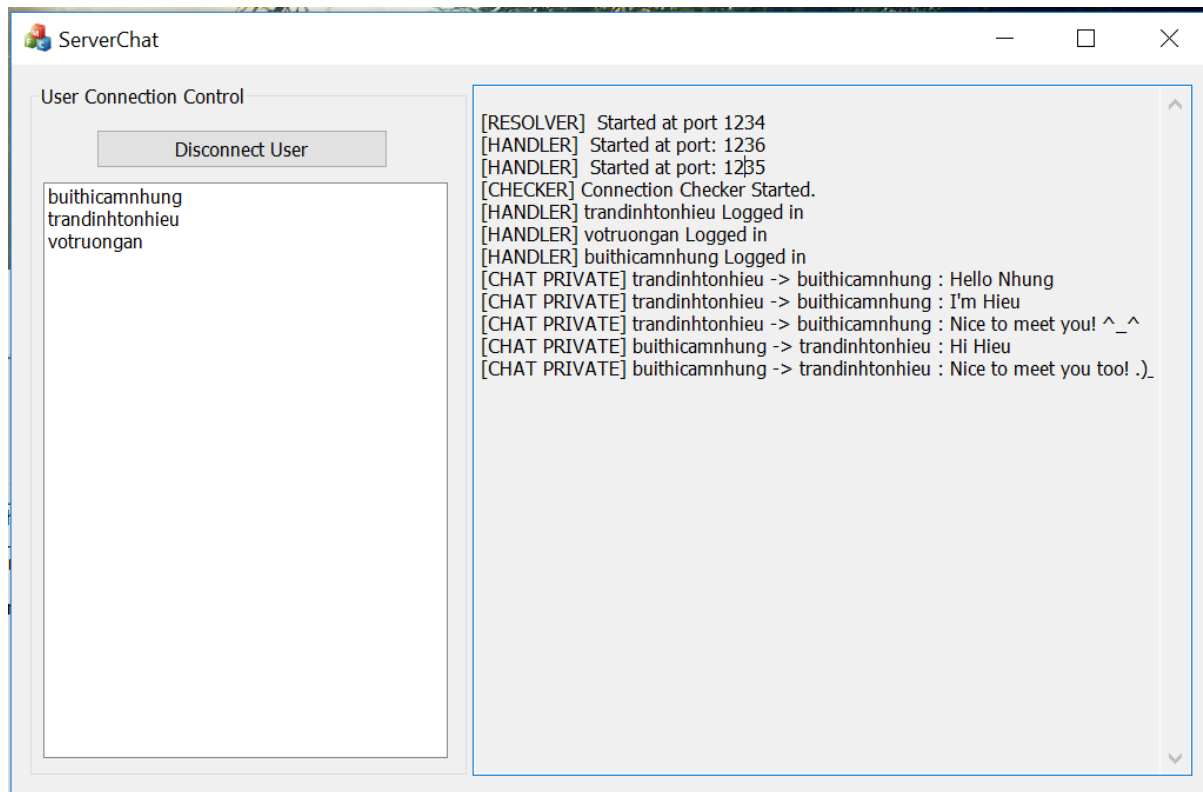


Khi mở 3 Client và đăng nhập 3 tài khoản cùng lúc

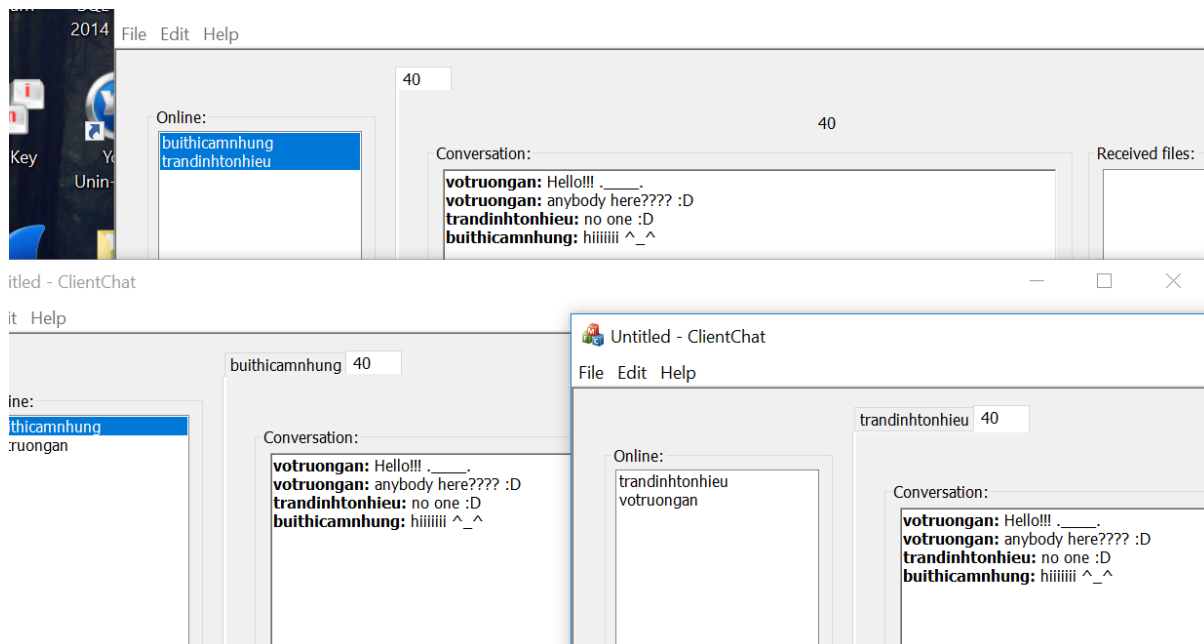
Sau khi đã đăng nhập vào chương trình, người dùng có thể chat với người dùng khác đang kết nối với Server bằng cách nhấn đúp chuột vào tên người dùng đó ở danh sách Online. Nếu người dùng muốn chat nhiều người dùng cùng lúc (chat nhóm), người dùng nhấp chuột vào tên những người dùng mình muốn tạo nhóm và nhấn nút "Create Group" để tạo nhóm.



Ví dụ về Chat riêng



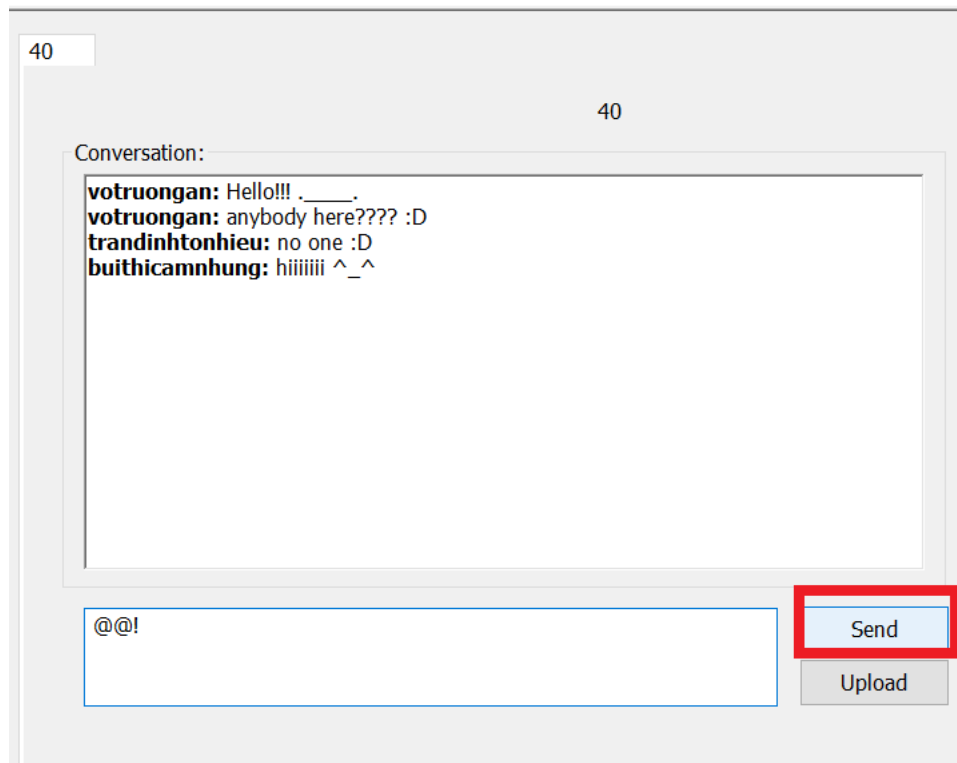
Server sẽ ghi log ra tất cả nội dung chat



Ví dụ về chat nhóm, ID nhóm chat là 40 (Được Server tự tạo)

Mỗi cửa sổ nói chuyện sẽ là một tab riêng ở phần bên phải của chương trình. Người dùng có thể chuyển đổi các tab với nhau để chuyển sang các cuộc trò chuyện khác nhau.

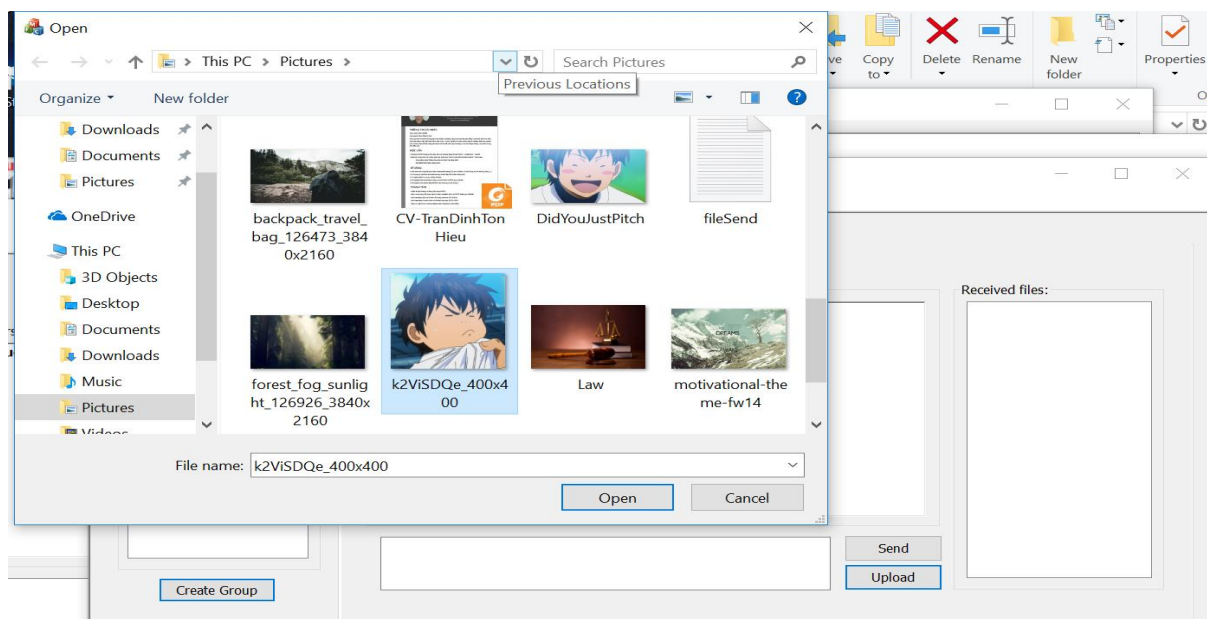
Để nhắn tin, người dùng nhập tin cần nhắn vào ô tin nhắn và nhấn nút “Send” để gửi tin nhắn.



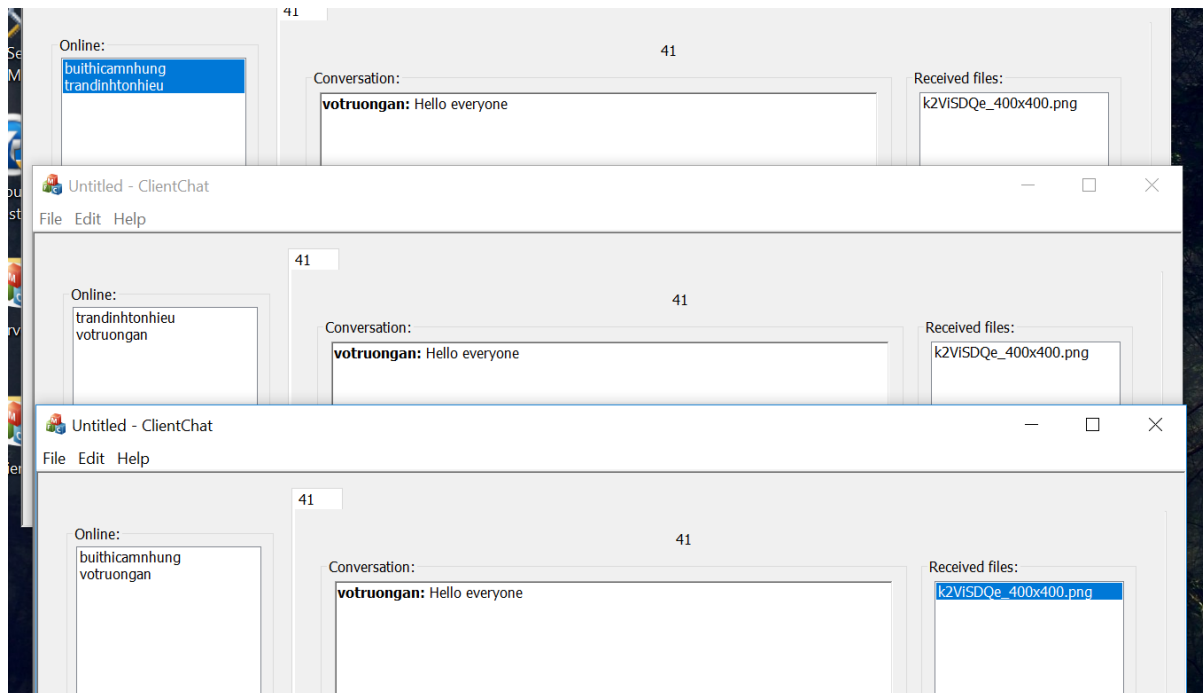
Ấn vào nút send để gửi tin nhắn

Để gửi file, người dùng nhấn vào nút Upload ở ngoài cùng bên phải của chương trình.

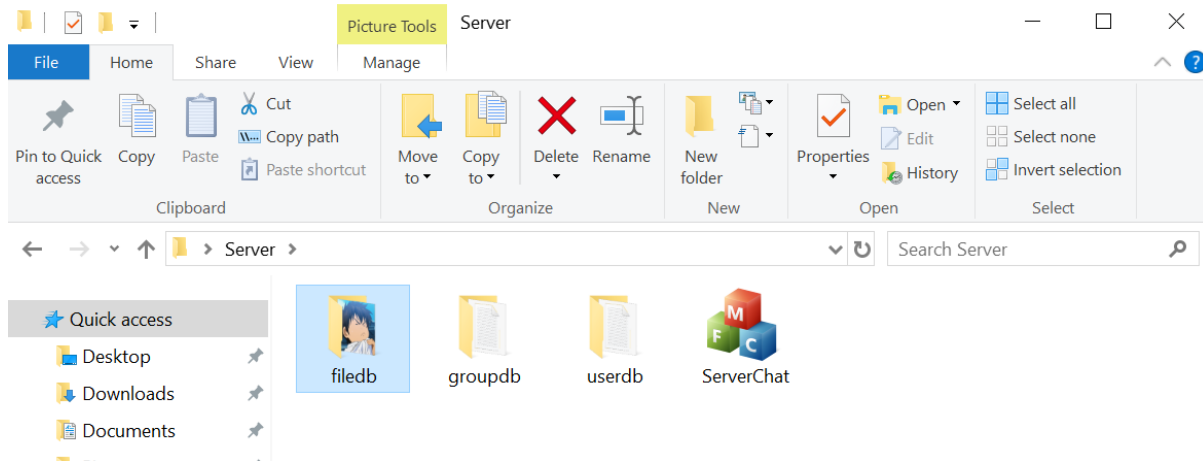
Cửa sổ duyệt file sẽ hiện lên để người dùng chọn file cần gửi



Sau khi ấn gửi file, ta sẽ thấy thông báo gửi file thành công và tên file xuất hiện trong hộp thoại bên phải:

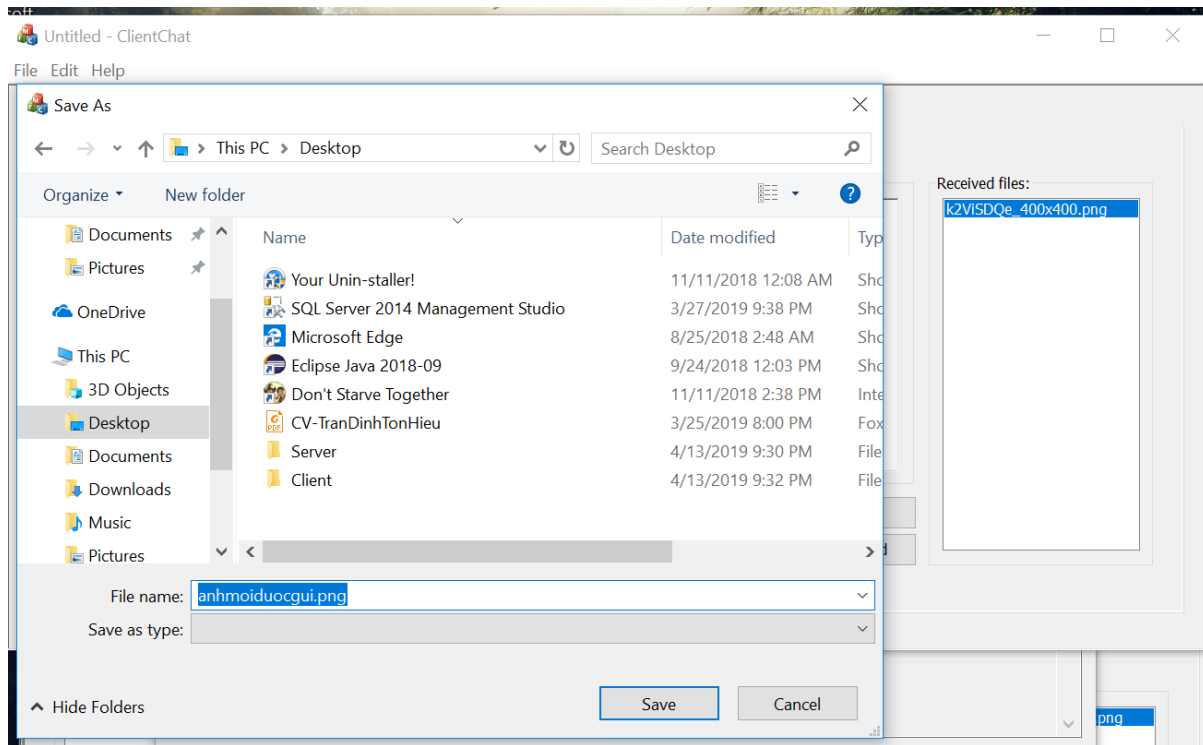


Lúc này, file vốn dĩ đã được lưu vào trong thư mục filedb của Server. Ta có thể kiểm tra:

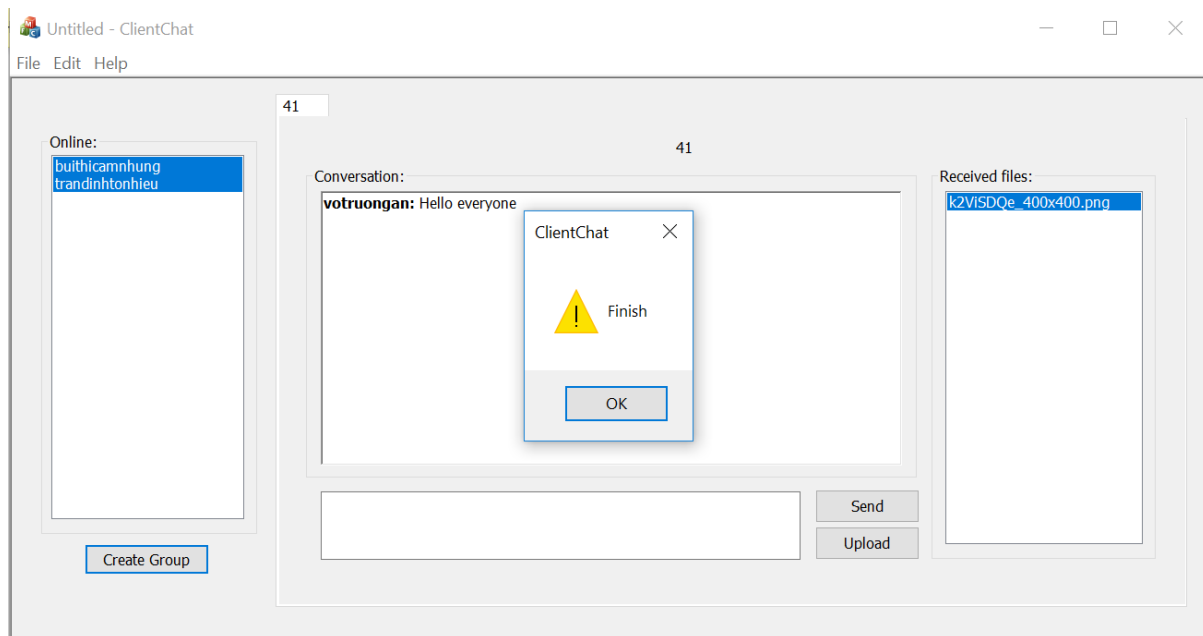


File đã được lưu trong thư mục filedb của Server

Client có thể click đúp vào tên file ở phía bên phải giao diện để tải file về:



Lúc này, ta có thể chọn nơi mà file sẽ được lưu và đặt tên cho file.

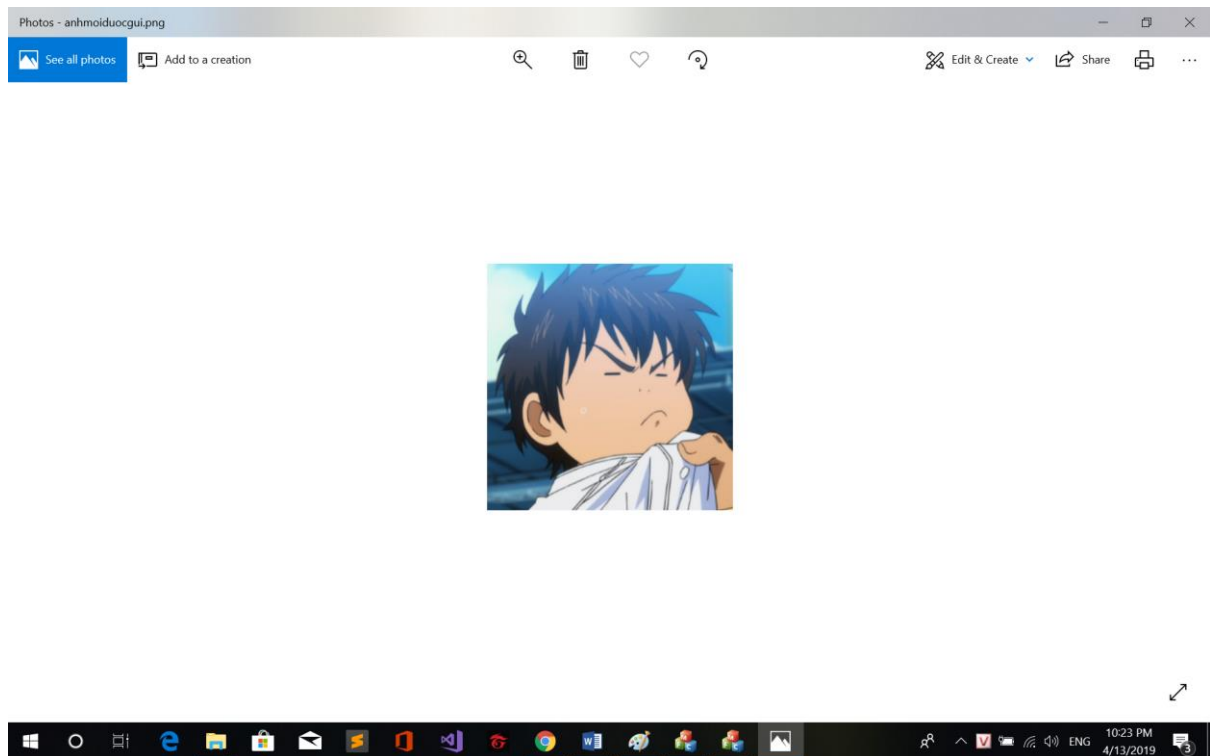


Hoàn thành tải file

Ta có thể thấy, lúc này ở Desktop (nơi ta chọn để lưu file vừa này) đã xuất hiện file ảnh:



Ta sẽ mở ảnh lên để kiểm tra xem có bị hỏng hay không:



Đây chính là ảnh ban đầu