

UNIVERSITY OF SCIENCE
FALCUTY OF INFORMATION TECHNOLOGY



SUBJECT: COMPUTER NETWORKING

SOCKET PROGRAMMING PROJECT CLASS 20CLC11

Group ID: 1_20127013_20127370

Đặng Nguyễn Duy – 20127013

Mai Quý Trung – 20127370

Lecturers: **NGUYỄN THANH QUÂN, LÊ HÀ MINH, LÊ NGỌC SƠN**

Table of Contents

<i>I) THÔNG TIN CỦA NHÓM</i>	<i>3</i>
<i>II) TIẾN ĐỘ CÔNG VIỆC.....</i>	<i>3</i>
<i>III) KỊCH BẢN GIAO TIẾP CHƯƠNG TRÌNH.....</i>	<i>3</i>
<i>IV) MÔI TRƯỜNG LẬP TRÌNH, FRAMEWORK HỖ TRỢ.....</i>	<i>7</i>
<i>V) HƯỚNG DẪN SỬ DỤNG.....</i>	<i>8</i>
<i>VI) BẢNG PHÂN CHIA CÔNG VIỆC</i>	<i>12</i>
<i>VII) NGUỒN, TÀI LIỆU THAM KHẢO</i>	<i>14</i>

I) THÔNG TIN CỦA NHÓM

Group ID: 1_20127013_20127370

- Đặng Nguyễn Duy – 20127013
- Mai Quý Trung – 20127370

Đề án: Lập trình socket (socket programming)

Môn học: Mạng máy tính (computer networking)

II) TIẾN ĐỘ CÔNG VIỆC

Đề tài: Tỷ giá vàng Việt Nam (Đề 1).

Nội dung: Chương trình mô phỏng ứng dụng trực tuyến gồm một server và nhiều client. Server sẽ lưu trữ thông tin giá vàng giúp client có thể tra cứu. Mọi quá trình client đăng nhập, client đăng xuất, ... đều được thể hiện trên màn hình của server.

Tiến độ công việc: Hoàn thành.

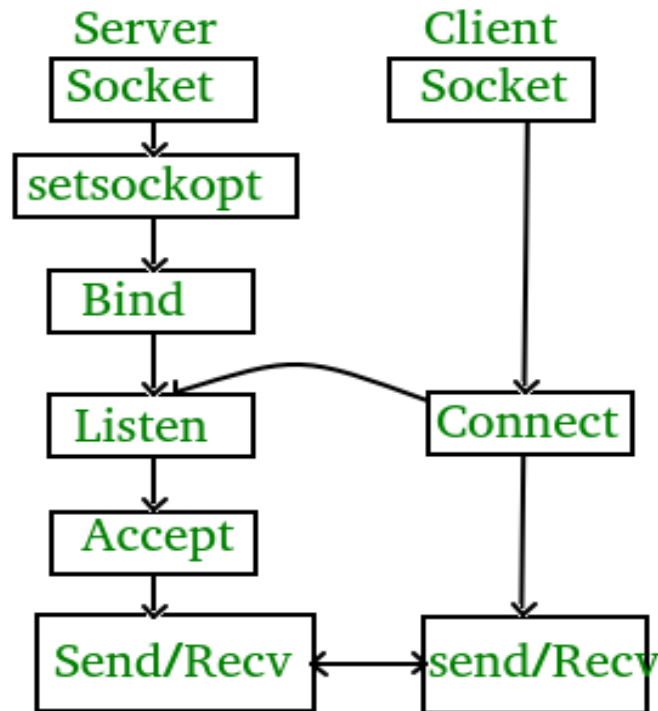
Phân công việc nhóm:

- Đặng Nguyễn Duy: Xây dựng kết nối giữa server và client, thread và thiết kế giao diện.
- Mai Quý Trung: Thu thập dữ liệu từ trang web thứ 3 (third-party) và xử lý data cho server.

III) KỊCH BẢN GIAO TIẾP CHƯƠNG TRÌNH

1. Giao tiếp giữa Server-Client:

- Sử dụng kết nối TCP (Transmission Control Protocol).
- Thư viện sử dụng cho việc kết nối: Winsock2.h (Microsoft Visual Studio).
- Giao tiếp giữa server và client sẽ được thực hiện thông qua việc gửi tín hiệu đi từ client đến server thông qua địa chỉ IP của server. Về phía server, tín hiệu sẽ được lắng nghe và xử lý, rồi gửi phản hồi ngược lại cho client.
- Kịch bản giao tiếp chi tiết của server-client được biểu diễn theo biểu đồ sau:



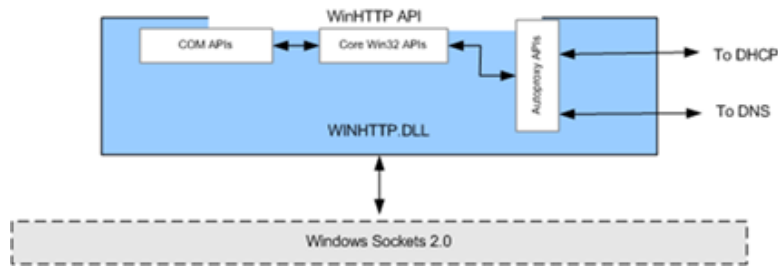
(Hình 3.1.1: Mô hình giao tiếp giữa server-client)

- **Setsockopt:**
 - + Server sẽ thiết lập lựa chọn cho socket.
 - + Về phía client: struct addinfo được khởi tạo và chứa cấu trúc địa chỉ socket và khai báo các giá trị của chúng.
- **Bind:** Hàm bind sẽ kết nối cấu trúc địa chỉ socket này với các lựa chọn socket đã được thiết lập, nhằm cho phép server chấp nhận các kết nối từ client.
- **Connect:** Hàm connect sẽ bao gồm socket đã được tạo ra từ server và cấu trúc địa chỉ socket mà client vừa khai báo thành công.
- **Listen:** Sau khi client đã kết nối với server thông qua địa chỉ IP và địa chỉ cổng trên hệ thống, server sẽ lắng nghe từ địa chỉ IP và cổng đó để nhận được các yêu cầu của client.
- **Accept:** Một khi socket đã được kết nối và lắng nghe, server sẽ phải tiếp nhận các yêu cầu từ các kết nối trên socket đó.
- **Send & Receive:** Hàm send và receive sẽ trả về giá trị ứng với số byte gửi hoặc số byte nhận. 2 hàm hoạt động giống nhau khi điều nhận biến clientsocket đang hoạt động, chuỗi giá trị gửi hoặc nhận, độ dài chuỗi và các biến cờ.
- **Disconnect:** Sau khi server và client kết nối, trao đổi các thông tin gửi và nhận xong thì client có thể ngưng kết nối khỏi server và server đóng socket.

- **Đăng kí, đăng nhập:** Client sẽ giao tiếp đến server thông qua 2 chuỗi kí tự request có độ dài mặc định. 2 chuỗi bao gồm username và password của client, sau đó được truyền đến server thông qua hàm send. Server sẽ so sánh chuỗi để xác định xem có chấp nhận đăng kí/đăng nhập cho client đó không. Server sẽ gửi về cho client chuỗi kí tự respond, client sẽ nhận được respond thông hàm receive, so khớp với điều kiện và đưa ra kết quả cho client. Tóm lại, giao tiếp đăng kí, đăng nhập giữa server và client là thông qua chuỗi kí tự có độ dài được cho trước.
- **Tra cứu:** Tương tự đăng kí và đăng nhập, client sẽ giao tiếp đến server thông qua 1 chuỗi request chung, chứa thông tin bao gồm loại vàng và ngày cần tra cứu, chuỗi với độ dài được cho trước sẽ được gửi đến thông qua hàm send. Về mặt giao diện (GUI), client sẽ chọn loại vàng trong 1 textbox và chọn thời gian trong calendar. Sau đó, server sẽ nhận được chuỗi đó, lấy ra data mà client cần và gửi về cho client. Data sau khi được client nhận về sẽ được hiện trên listview của client GUI.
- **File:** Thông tin chính giao tiếp giữa server và client sẽ bao gồm đăng kí, đăng nhập, đăng xuất và thông tin lưu trữ tỷ giá vàng.
 - + Account.txt: File lưu trữ các tài khoản của client. Giao tiếp gửi nhận gồm username và password được gửi từ client, nếu là đăng kí, server sẽ kiểm tra từ file xem tài khoản này đã tồn tại hay chưa, nếu đăng nhập, server sẽ kiểm tra username và password có trùng khớp không, và trả về chuỗi tài khoản tương ứng để xác định chấp nhận hoặc từ chối đăng kí, đăng nhập của client.
 - + GoldData: File lưu trữ data tỷ giá vàng được lấy từ third party.

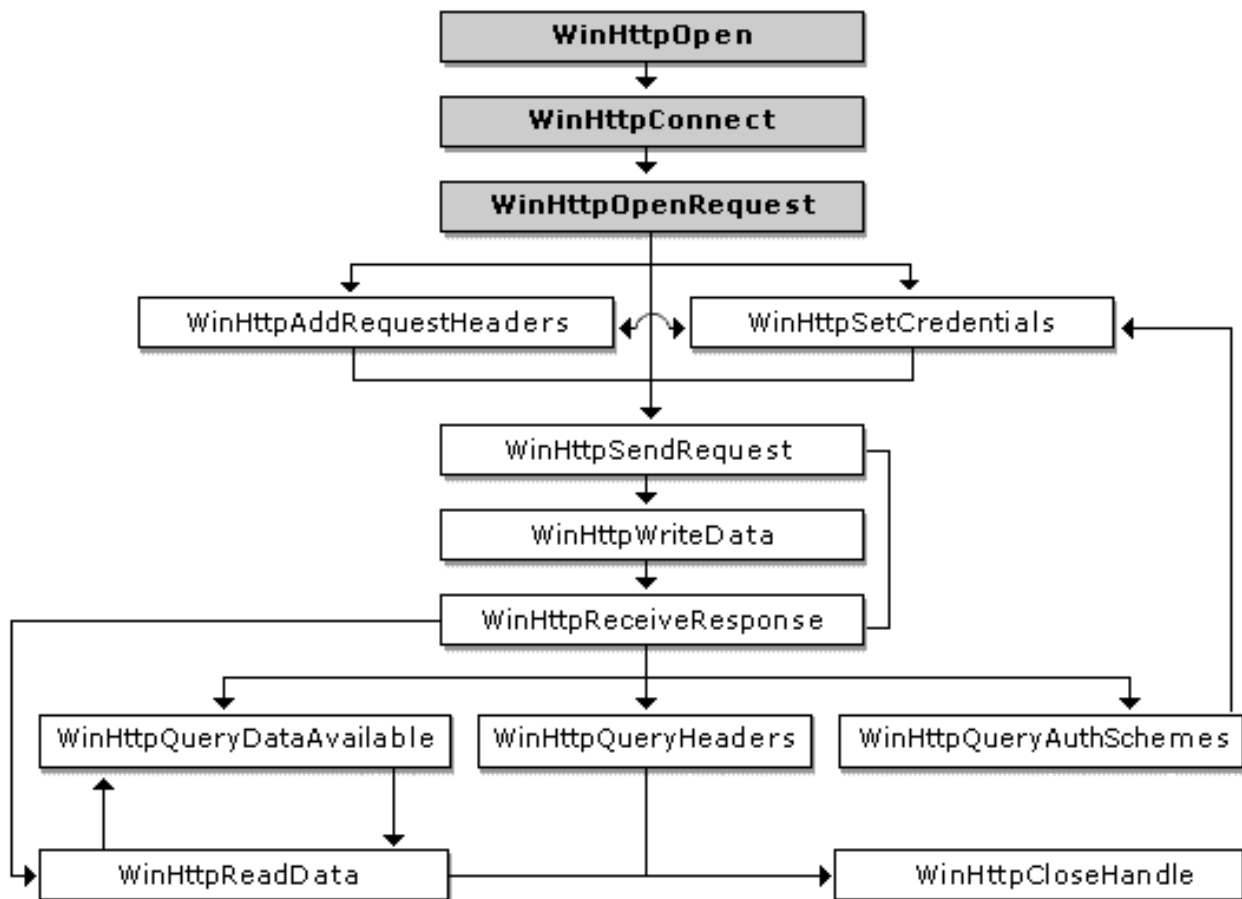
2. Giao tiếp giữa Server-Third party:

- Thư viện sử dụng: Winhttp.h (Microsoft Visual Studio).
- Third party: <https://vapi.vnappmob.com>
- Sử dụng kết nối TCP (Transmission Control Protocol).
- Giao tiếp giữa server và third party được diễn ra thông qua việc server sẽ kết nối thông qua winhttp.h, sau đó sẽ gửi request đến bên thứ 3 về api key để mở khoá và lấy về dữ liệu tỷ giá vàng. Sau khi có được api key và request được chấp nhận, server sẽ nhận response về, file dữ liệu chứa thông tin về tỷ giá vàng.
- Database: Cơ sở dữ liệu được sử dụng khi lấy data tỷ giá vàng và dùng để lưu trữ là JSON.



(Hình 3.2.1: Liên kết giữa server và third party qua Windows Socket 2.0 và WinHttp)

- Thông qua sự liên kết giữa winsock2.h và winhttp.h, server sẽ truy cập được vào mạng lưới kết nối của third party, từ đó lấy được api để trích xuất data về cho server.
- Sẽ có 2 công việc chính cần được thực hiện:
 - + Kết nối đến third party để lấy api key thông qua lệnh GET.
 - + Server sẽ tiếp tục kết nối đến third party 1 lần nữa để lấy file data.



(Hình 3.2.2: Sơ đồ biểu diễn cách thức trích xuất data từ third party của WinHttp)

- 1 biến đặc biệt với kiểu dữ liệu là HINTERNET được sử dụng để có thể truy cập vào data.
- WinHttpOpen dùng để mở đến trang web của bên thứ 3
- WinHttpConnect sau đó được dùng để kết nối với trang web thông qua HINTERNET
- WinHttpRequest cho phép server gửi request.
- WinHttpSendRequest cho server bắt đầu gửi request đến trang web.
- Sau khi xử lý request, WinHttpReceiveResponse dùng để trang web gửi response về cho server
- WinHttpQueryDataAvailable và WinHttpReadData được sử dụng để kiểm tra xem có gặp trục trặc hay lỗi gì khi server đang lấy data về
- Sau khi lấy được toàn bộ data cần thiết, WinHttpCloseHandle sẽ đóng và ngắt kết server khỏi third party, trang web cũng ngừng nhận các handle.
- Giao tiếp lấy data giữa server và third party: Server gửi đi và nhận về data được chứa trong 1 chuỗi kí tự buffer, thực hiện thao tác gửi request và nhận response sau khi third party trích xuất data và trả về chuỗi buffer, sau đó server có thể lưu chuỗi về dạng string hoặc ghi vào file.

IV) MÔI TRƯỜNG LẬP TRÌNH, FRAMEWORK HỖ TRỢ

1. Môi trường lập trình: C++ (Visual Studio)

- Ưu điểm:
 - + Dễ sử dụng, ngôn ngữ căn bản, nền tảng.
 - + Giúp luyện tập, nâng cao khả năng lập trình bằng C++.
- Nhược điểm:
 - + Thiếu nhiều thư viện, phải sử dụng thêm từ nguồn bên ngoài (rapidjson).
 - + Số lượng các hàm hỗ trợ ít nên hầu như phải viết hàm riêng.

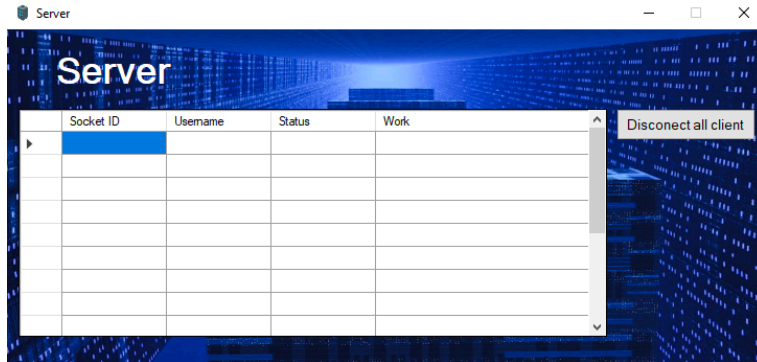
2. Framework hỗ trợ: GUI components in Visual C++/CLR, C++/CLR Windows (.NET 4.7.2)

- Ưu điểm:
 - + Tiếp cận tương đối dễ, framework có sẵn trong Visual Studio.
 - + Có thể thêm các label, button, textbox, trang trí, chỉnh sửa dễ dàng, thao tác đơn giản.
- Nhược điểm:
 - + Cú pháp của phần back end cho GUI khó sử dụng, cú pháp câu lệnh dài.
 - + Các kiểu dữ liệu thông thường của C++ bị thay đổi, thao tác cài đặt xử lý đa luồng gặp khó khăn.
 - + Ít tài liệu tham khảo, chủ yếu là tài liệu cho C#.

V) HƯỚNG DẪN SỬ DỤNG

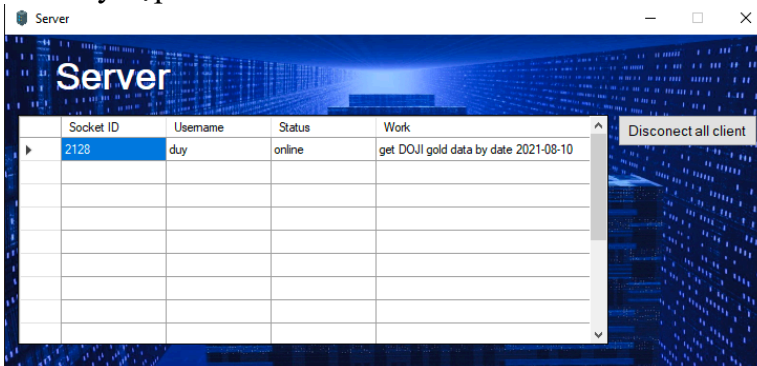
1. Server

- Khởi động: Server.exe



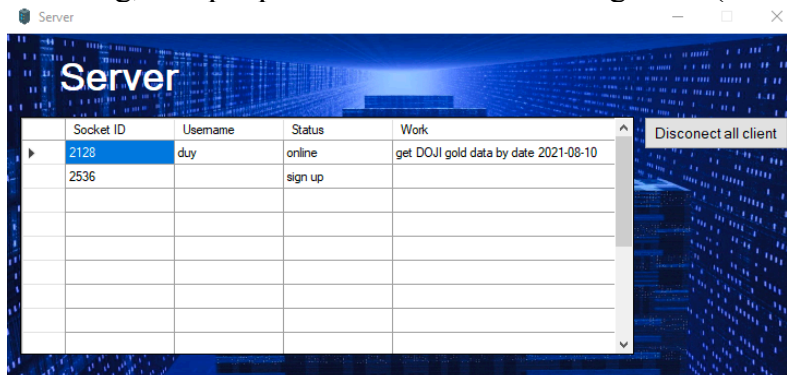
(Hình 5.1.1: Khởi động server)

- Cho client truy cập vào socket:



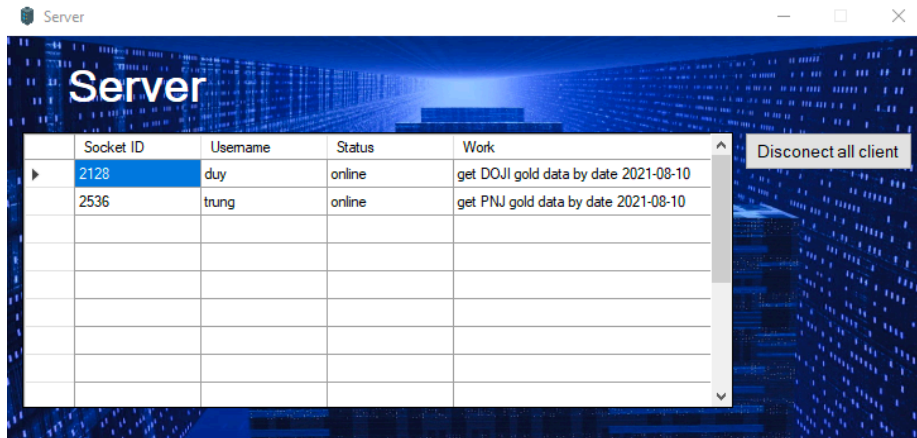
(Hình 5.1.2: Client kết nối vào server)

- Xử lý đa luồng, cho phép nhiều client kết nối cùng 1 lúc (thread):



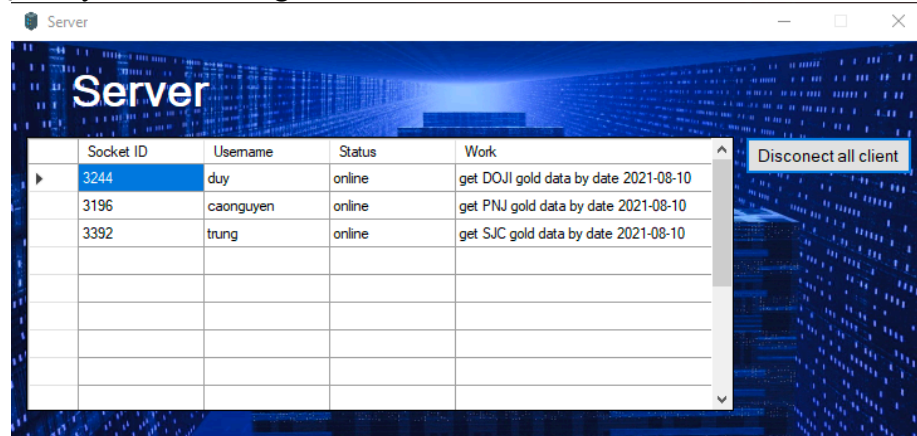
(Hình 5.1.3: Nhiều client có thể cùng lúc truy cập)

- Xử lý request từ client: cho phép client tìm kiếm thông tin về tỷ giá vàng

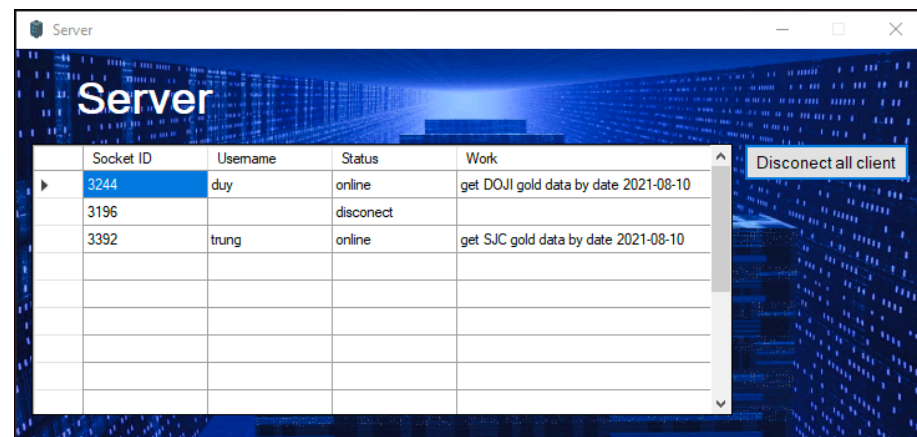


(Hình 5.1.4: Xử lý request cho các client)

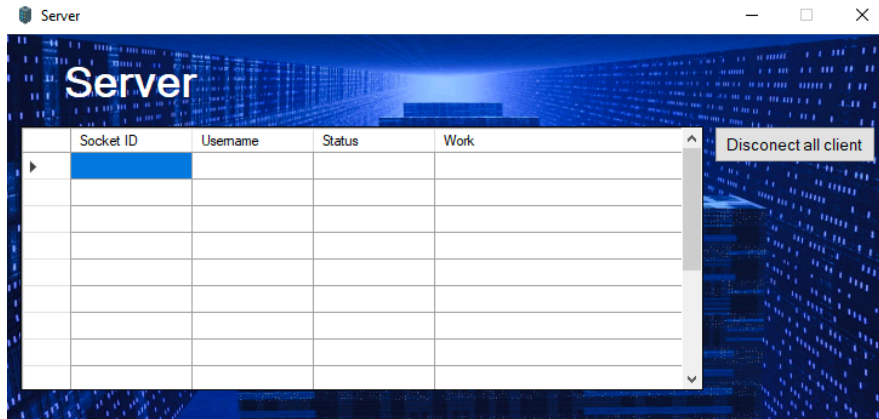
- Quản lý các client, ngắt kết nối:



(Hình 5.1.5: Các client hiện đang kết nối)



(Hình 5.1.6: 1 client disconnect khỏi server)



(Hình 5.1.7: Server disconnect toàn bộ client)

2. Client

- Khởi động:



(Hình 5.2.1: Khởi động client)

- Truy cập vào server:



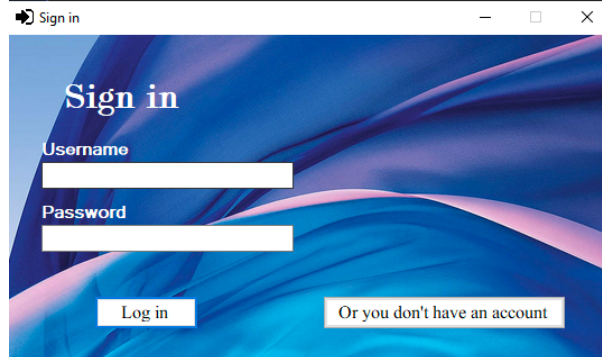
(Hình 5.2.2: Client truy cập vào server)

- Đăng kí:



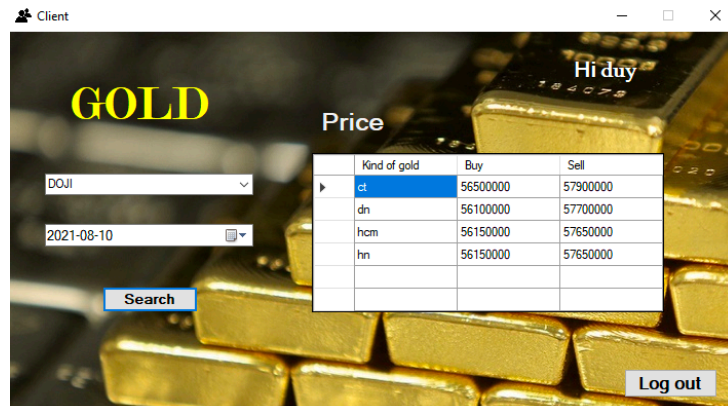
(Hình 5.2. : Đăng ký cho client)

- Đăng nhập:



(Hình 5.2. : Đăng nhập cho client)

- Đăng xuất: Sau khi đăng xuất, client sẽ trở về màn hình đăng nhập/đăng ký.
- Tra cứu:



	Kind of gold	Buy	Sell
▶	ct	56500000	57900000
	dn	56100000	57700000
	hcm	56150000	57650000
	hn	56150000	57650000

(Hình 5.2. : Tra cứu cho client)

File hướng dẫn sử dụng ngoài ra còn được bao gồm trong file readme.txt trong project

VI) BẢNG PHÂN CHIA CÔNG VIỆC

Chức năng\Người thực hiện	Nguyễn Duy	Quý Trung
KẾT NỐI		X
QUẢN LÝ KẾT NỐI	X	
ĐĂNG NHẬP	X	
ĐĂNG KÝ	X	
TRA CỨU		X
QUẢN LÝ CƠ SỞ DỮ LIỆU		X
THOÁT	X	
GIAO DIỆN SERVER	X	
GIAO DIỆN CLIENT	X	
REPORT		X

Cụ thể:

1. KẾT NỐI:

- Cho phép client kết nối đến server thông qua kết nối TCP.
- Cho phép client và server đặt tại các host khác nhau (cho phép client nhập IP của server để kết nối).

2. QUẢN LÝ KẾT NỐI:

- Khi client hoặc server mất kết nối đột ngột, không làm chương trình treo hay xảy ra lỗi.

- Nếu một client mất kết nối không làm ảnh hưởng đến các client khác.
- Quản lý kết nối đa tiểu trình (thread).
- Nhiều client có thể kết nối đến server cùng lúc.

3. ĐĂNG NHẬP:

- Client đăng nhập bằng cách gửi username, password cho server.
- Server nhận thông tin username, password từ client và kiểm tra với thông tin đã lưu trữ tại server.
- Client phải nhập đúng username và password để có thể đăng nhập, có thể chuyển sang mục đăng ký nếu chưa có tài khoản.

4. ĐĂNG KÝ:

- Client đăng ký bằng cách gửi username, password cho server.
- Server nhận thông tin username, password từ client và kiểm tra với thông tin đã lưu trữ tại server, nếu đã tồn tại, gửi thông báo đến client, yêu cầu đăng ký tài khoản khác.
- Client phải tạo username và password không trùng với tài khoản nào hiện có trong data của server, có thể chuyển sang mục đăng nhập nếu đã có tài khoản.

5. TRA CỨU:

- Cho phép client tra cứu theo ngày, theo loại vàng (SJC Ha Noi, SJC HCM, PNJ SJC...).
- Server sẽ kết nối tới một website khác (third party) để lấy thông tin (JSON), sau đó rút trích thông tin và lưu trữ dữ liệu dưới server để phục vụ request của client.
- Server cập nhật thông tin liên tục 30 phút 1 lần của ngày hôm đó.

6. QUẢN LÝ CƠ SỞ DỮ LIỆU:

- Sử dụng dữ liệu lưu trữ trong file (json).

7. THOÁT:

- Client được phép gửi thông báo ngừng kết nối đến server.
- Server có thể gửi thông báo ngừng kết nối đến tất cả client đang hoạt động.

8. GIAO DIỆN:

- Console Application (GUI):
 - + Thiết kế giao diện đồ họa cho Client.
 - + Thiết kế giao diện đồ họa cho Server.

VII) NGUỒN, TÀI LIỆU THAM KHẢO

- **Microsoft:**
 - + <https://docs.microsoft.com/en-us/windows/win32/winsock/getting-started-with-winsock>
 - + <https://docs.microsoft.com/en-us/windows/win32/winsock/complete-client-code>
 - + <https://docs.microsoft.com/en-us/windows/win32/winsock/complete-server-code>
 - + <https://docs.microsoft.com/en-us/windows/win32/api/winsock2/nf-winsock2-socket>
 - + <https://docs.microsoft.com/en-us/windows/win32/api/winsock2/nf-winsock2-accept>
- **GeeksForGeeks:**
 - + <https://www.geeksforgeeks.org/socket-programming-cc/>
- **RapidJson:**
 - + https://rapidjson.org/md_doc_tutorial.html
- **Stdio:**
 - + <https://www.stdio.vn/article/huong-dan-su-dung-rapidjson-de-xu-ly-json-trong-cpp-0ZsuX>
- **Youtube:**
 - + <https://www.youtube.com/watch?v=QCnyqMWPkQk>
 - + <https://www.youtube.com/watch?v=gB51Tla5pPI>
 - + <https://www.youtube.com/watch?v=xTRTY-fOIe8&t=5953s>