

Thành viên và phân chia công việc

Họ và tên	Nhiệm vụ	Mức độ hoàn thành
Nguyễn Hoàng Duy Tân		100%
Trần Nguyễn Minh Tuệ		100%
Nguyễn Minh Lộc		100%

Phân tích yêu cầu

Functional requirements

Xây dựng một ứng dụng chia sẻ file đơn giản với giao thức được định nghĩa sẵn, sử dụng những giao thức trong TCP/IP stack.

Client Functions

Basic functions

Đăng ký trong kho lưu trữ

- Máy khách có thể gửi yêu cầu đăng ký file có trong kho lưu trữ cho máy chủ.
- Thông điệp đăng ký file: “publish “. File trên máy khách sẽ được thêm vào kho lưu trữ dưới tên .
- Các file sau khi đăng ký sẽ được lưu trữ trong kho lưu trữ của tài khoản được liên kết với máy chủ.

Gửi yêu cầu tải file cho server

- Máy khách có thể gửi yêu cầu tải file không có sẵn trong kho lưu trữ của mình. Lúc này máy chủ sẽ phản hồi lại danh sách các máy khách khác có file được yêu cầu.

Tải file trực tiếp từ nguồn muốn chọn

- Máy khách sau khi nhận được phản hồi từ máy chủ danh sách máy khách có sẵn file được yêu cầu có thể chọn một nguồn thích hợp và gửi yêu cầu tải file tới đó.
- Các máy khách được cung cấp một danh sách yêu cầu tải file từ các máy khách khác, máy khách có thể chọn 1 file trong danh sách và gửi yêu cầu tải file tới máy khách đó.
- Thông điệp tải file: “fetch “. Trong đó fname là 1 trong những tên file muốn chọn sau khi server đã phản hồi.

Extended functions

Đăng ký tài khoản

- Người dùng đăng ký địa chỉ của máy vào hệ thống của máy chủ.

Đăng nhập tài khoản và xác thực

- Người dùng đăng nhập tài khoản để sử dụng các chức năng của hệ thống.

Liệt kê danh sách lưu trữ

- Máy khách có thể kiểm tra danh sách các file có trong kho lưu trữ của mình.

Tìm kiếm bằng từ khóa

- Server sẽ hỗ trợ người dùng tìm kiếm file theo từ khóa.

Server Functions

Basic functions

Kiểm tra trạng thái máy chủ

- Máy chủ có thể kiểm tra trạng thái của máy chủ khác thông qua lệnh “ping <hostname>”

Xem danh sách file của máy khách khác

- Máy chủ có thể xem danh sách file trong kho lưu trữ của các máy khách thông qua lệnh “discover <hostname>”

Gửi thông tin cần thiết sau khi nhận yêu cầu tải file từ clients

- Sau khi nhận được yêu cầu tìm file từ người dùng, server sẽ tiến hành theo dõi và tìm kiếm để trả về các thông tin nơi đang lưu trữ các file đó cho clients: ID peer, thời gian file được cập nhật.

Extended functions

Xem file log

- Máy chủ có thể xem file log của máy khách khác thông qua.

Non-functional requirements

Giao diện người dùng – Cung cấp giao diện người dùng dễ sử dụng cho máy khách để nhập các lệnh và theo dõi quá trình tải tệp. **Multi-threading** – Triển khai đa luồng trong máy khách để có thể xử lý nhiều tải xuống cùng lúc. **Hiệu năng và tích hợp** – Đảm bảo rằng hệ thống hoạt động hiệu quả và có khả năng tích hợp với các mạng internet và hệ thống người dùng khác nhau.

Phân tích kiến trúc

Kiến trúc Peer-to-Peer (P2P)

- Kiến trúc Peer-to-Peer (P2P) là một mô hình mạng máy tính trong đó các máy tính (được gọi là nút hoặc “peers”) kết nối trực tiếp với nhau để chia sẻ tài nguyên và thông tin mà không cần sự tương tác trung tâm từ máy chủ. Trong kiến trúc P2P, mỗi máy tính có thể đồng thời hoạt động như máy khách và máy chủ, có nghĩa là chúng có khả năng yêu cầu tài nguyên từ các máy tính khác và chia sẻ tài nguyên với người khác.

Kiến trúc client-server

- Kiến trúc client-server (còn được gọi là mô hình client-server) là một kiến trúc máy tính phổ biến được sử dụng trong việc tổ chức và quản lý các dịch vụ và tài nguyên trên mạng. Nó dựa trên sự phân chia các vai trò chính trong hệ thống thành hai phần: máy khách (client) và máy chủ (server). Hai phần này tương tác với nhau để cung cấp các dịch vụ, ứng dụng, và tài nguyên cho người dùng.

Giới thiệu Protocol

HyperText Trànser Protocol

HyperText Transfer Protocol (HTTP) là một giao thức ở tầng ứng dụng trong mô hình OSI để gửi và nhận tài liệu, hình ảnh, văn bản như HTML document. Về cơ bản, giao thức HTTP xây dựng trên cơ chế request-response trong mô hình client-server. Trong mô hình này, client có thể là một process thuộc máy tính này, server có thể là process thuộc máy tính khác, hai process thuộc hai phần cứng khác nhau khi muốn giao tiếp với nhau thì sẽ thông qua HTTP để giao tiếp. Ai là người request thì đó là client, người nhận request để response sẽ là server.

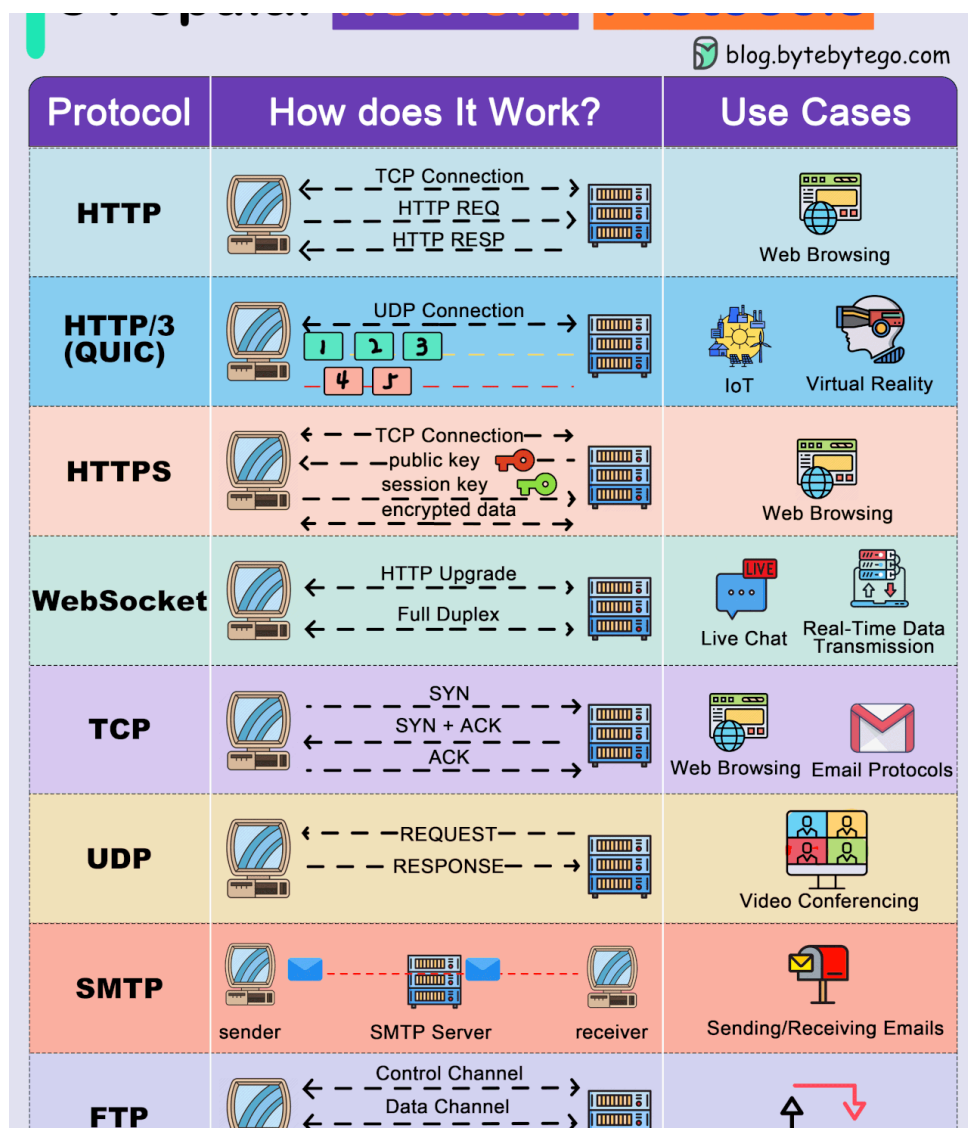


Figure 1: 8 giao thức mạng máy tính phổ biến (Nguồn Bytebytego)

Transmission Control Protocol

Transmission Control Protocol (TCP) là một trong các giao thức cốt lõi của bộ giao thức TCP/IP. Nhờ có TCP, các ứng dụng trên các host được nối mạng có thể tạo các kết nối với nhau, mà qua đó chúng có thể trao đổi dữ liệu hoặc các gói tin. Giao thức này đảm bảo chuyển giao dữ liệu tới nơi nhận một cách đáng tin cậy và đúng thứ tự.

Cách đặc tính cơ bản của TCP:

- Point-to-point: Trong một giao thức TCP, chỉ có một sender và một server được kết nối với nhau bằng 3-way handshaking.
- Reliable, in-order bit stream: Hỗ trợ truyền tin cậy và đúng thứ tự.
- Pipelined: Truyền song song nhằm tăng hiệu quả gửi nhận
- Flow control: Receiver kiểm soát tốc độ gửi của sender để tránh làm quá tải receiver.
- Congestion control: Tự động điều chỉnh tốc độ gửi ở mức tối đa mà không làm tắc nghẽn hệ thống.
- Full-duplex connection: hỗ trợ truyền hai chiều trong cùng một thời điểm trong một kết nối.

Socket programming with Python

Socket là cánh cổng ngăn cách giữa Application Layer với Transport Layer.

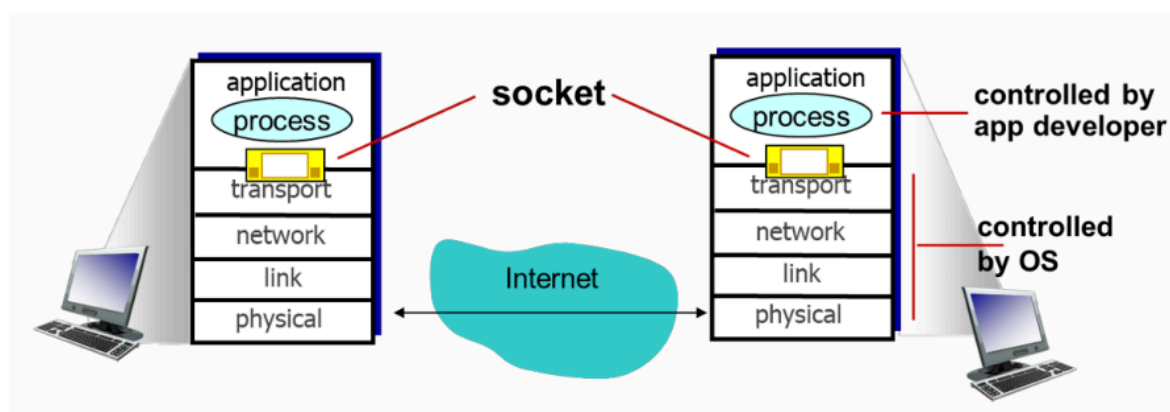


Figure 2: Mô hình socket

Thư viện socket của Python hỗ trợ một số API sau để thiết lập kết nối socket TCP/UDP:

- socket()
- bind()
- listen()
- accept()
- connect()
- connect_ex()
- send()
- recv()
- close()

Thư viện socket của Python hỗ trợ cả TCP socket lẫn UDP socket. Trong project này, nhóm dự định sử dụng TCP socket để hiện thực dự án.

Flow của một TCP socket connection có thể được thể hiện như hình dưới đây:

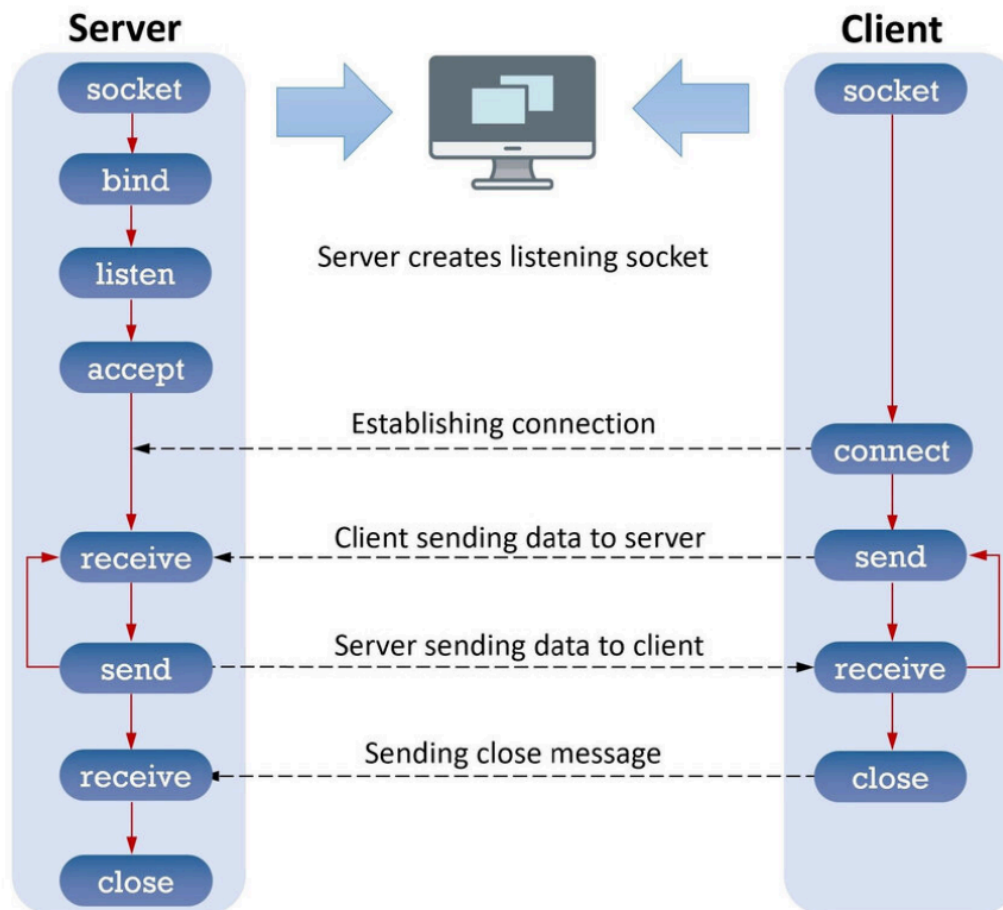


Figure 3: Quy trình một kết nối socket để gửi nhận dữ liệu (Nguồn ResearchGate)

Application design

Architecture design

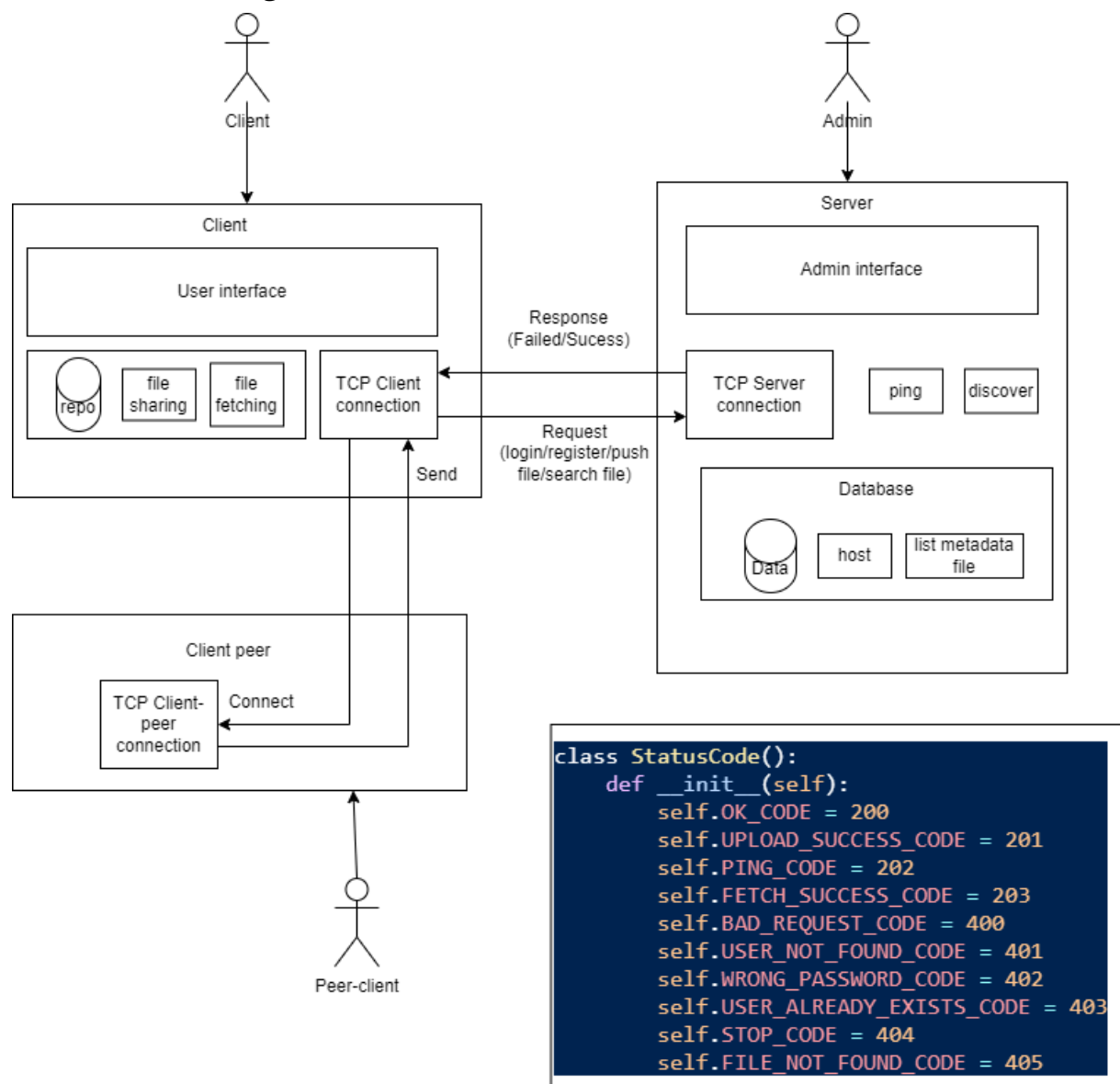


Figure 4: Kiến trúc tổng quan của hệ thống

Flow protocol design

Register

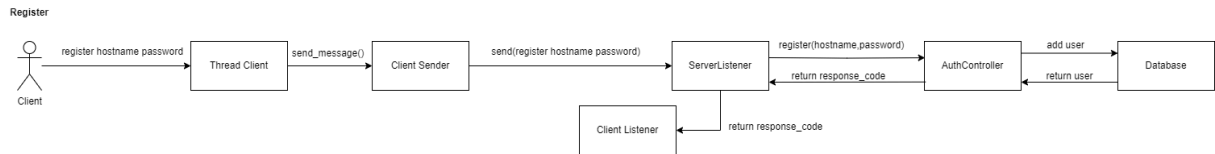


Figure 5: Sơ đồ luồng của chức năng đăng ký

Login

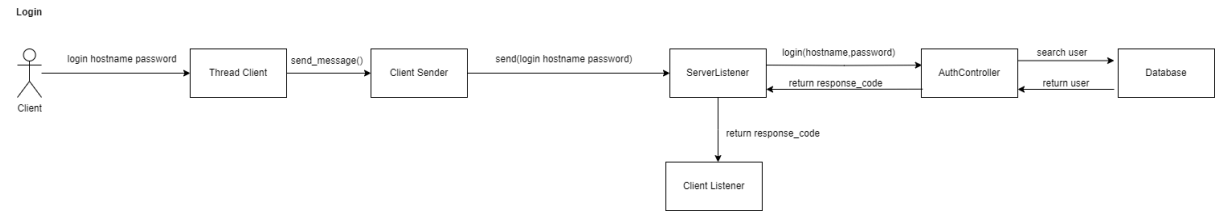


Figure 6: Sơ đồ luồng của chức năng đăng nhập

Ping

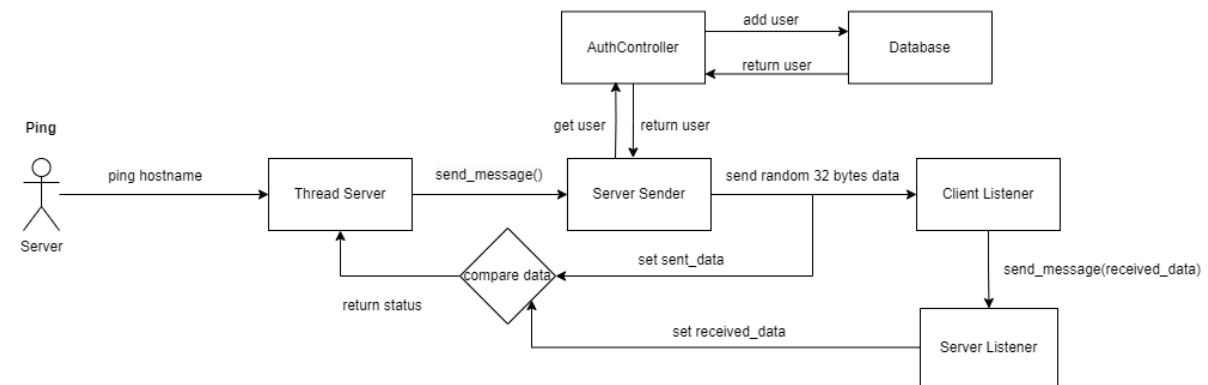


Figure 7: Sơ đồ luồng của chức năng ping

Discover

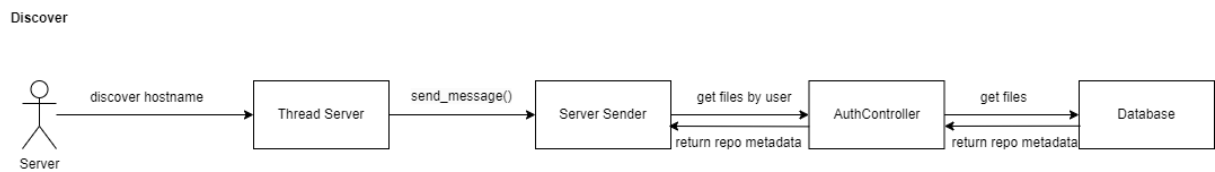


Figure 8: Sơ đồ luồng của chức năng discover

Publish

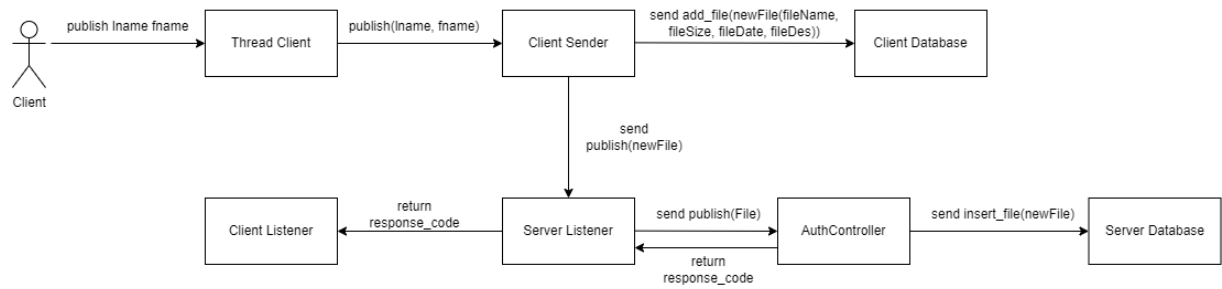


Figure 9: Sơ đồ luồng của chức năng publish

Fetch

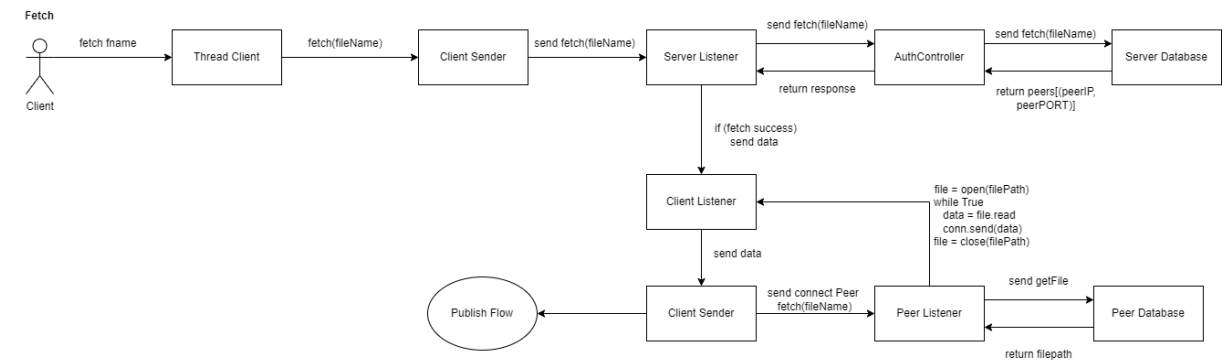


Figure 10: Sơ đồ luồng của chức năng fetch

Activity diagram

Register

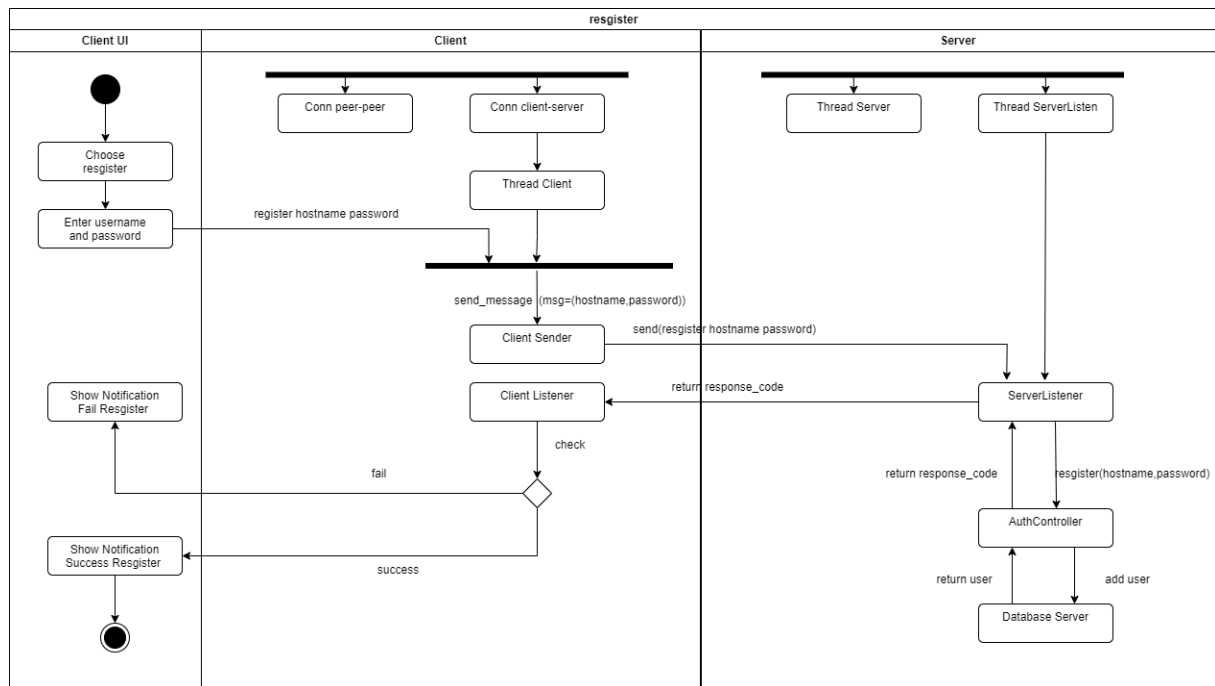


Figure 11: Sơ đồ hoạt động của chức năng đăng ký

Login

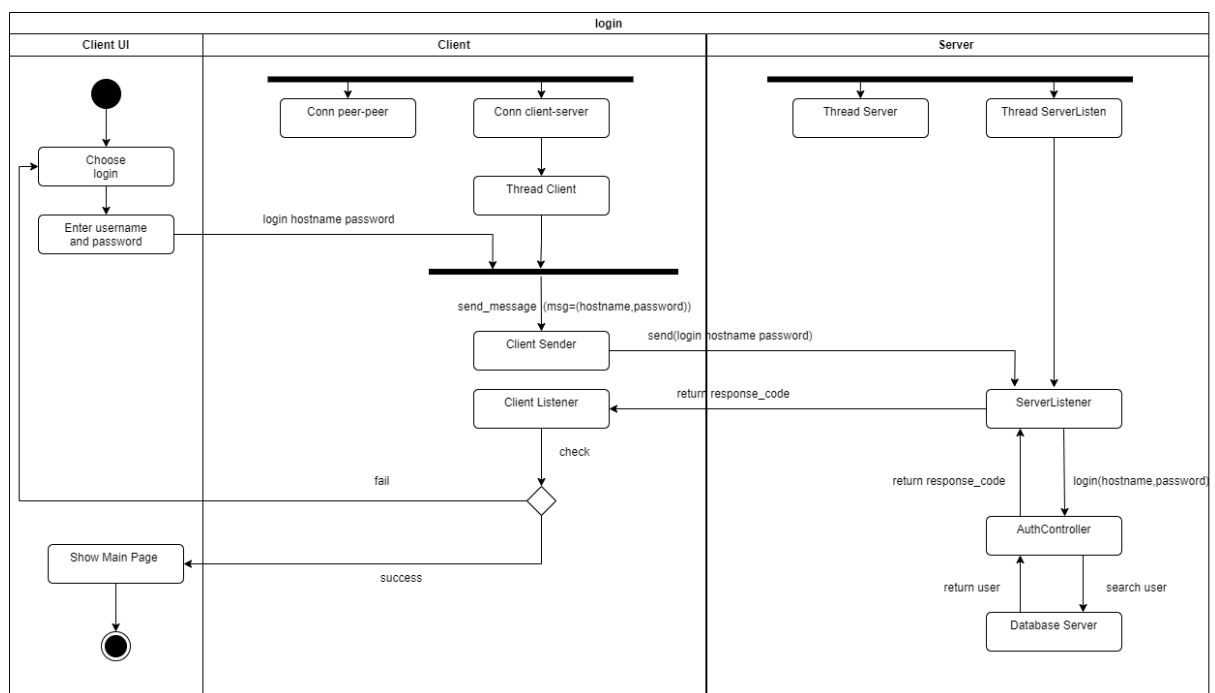


Figure 12: Sơ đồ hoạt động của chức năng đăng nhập

Ping

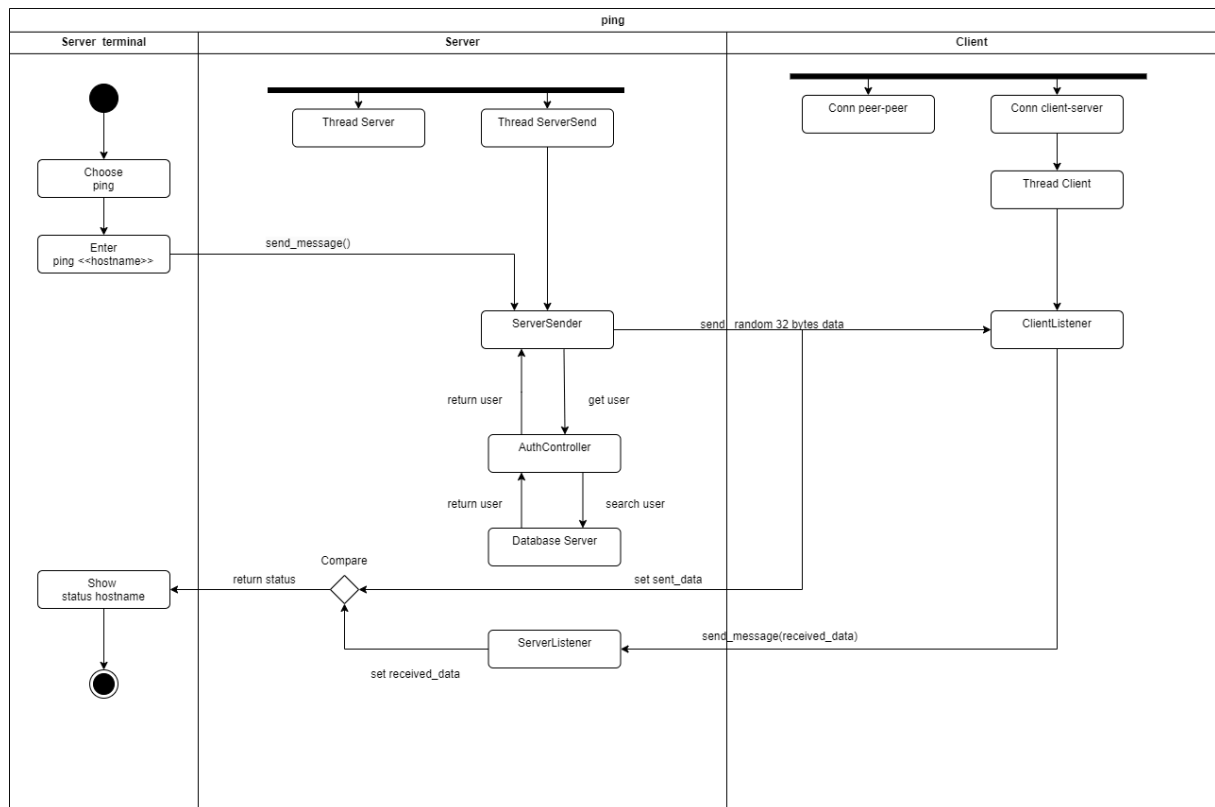


Figure 13: Sơ đồ hoạt động của chức năng ping

Discover

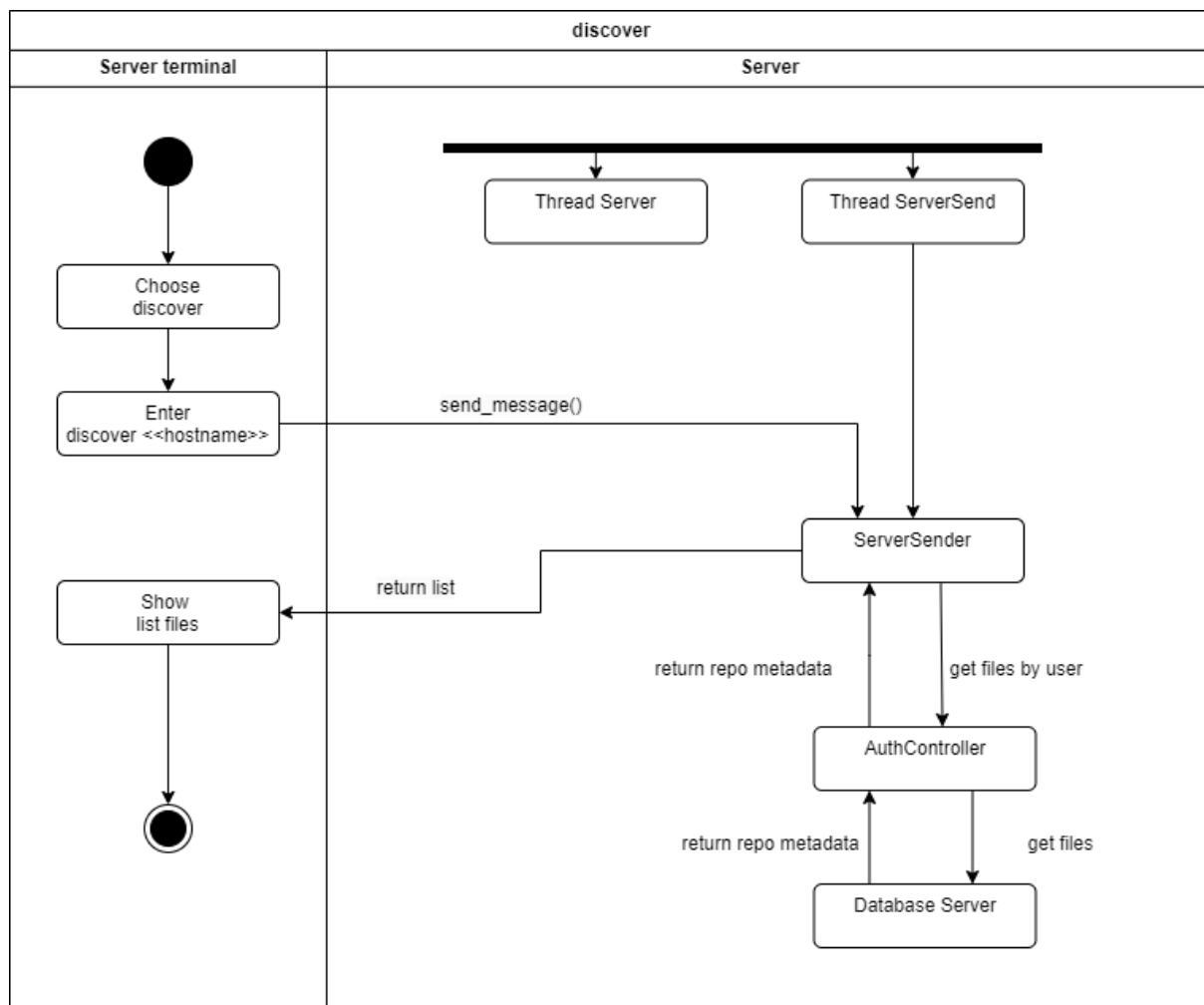


Figure 14: Sơ đồ hoạt động của chức năng discover

Publish

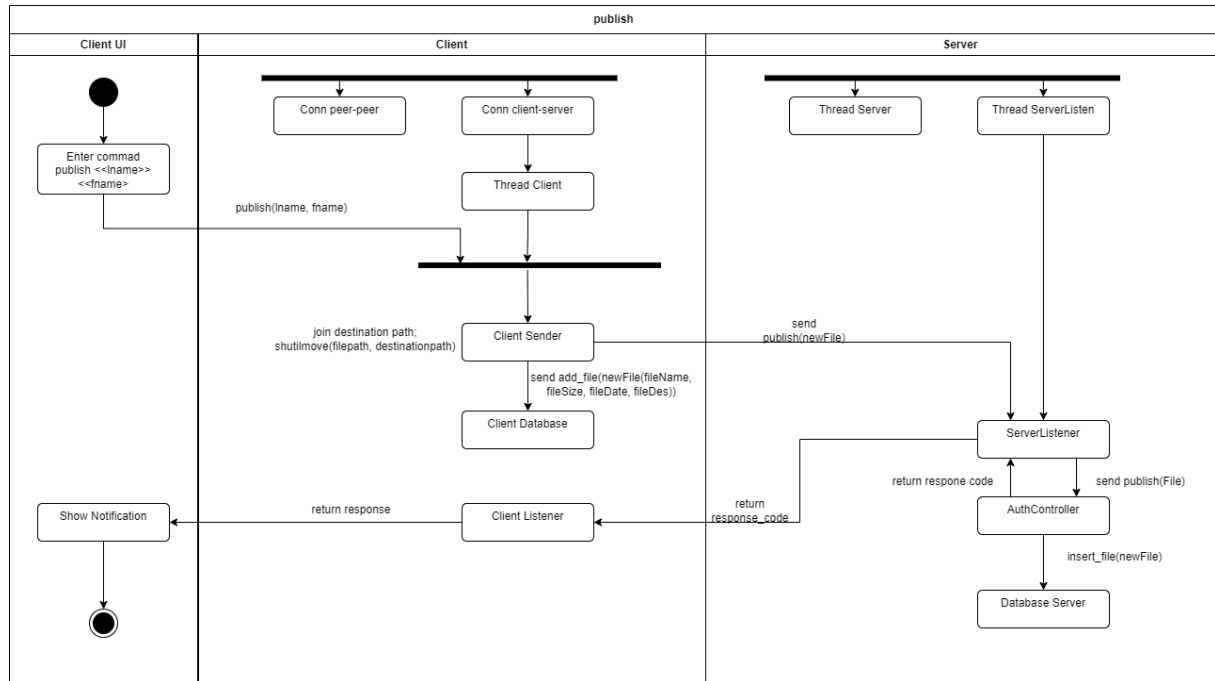


Figure 15: Sơ đồ hoạt động của chức năng publish

Fetch

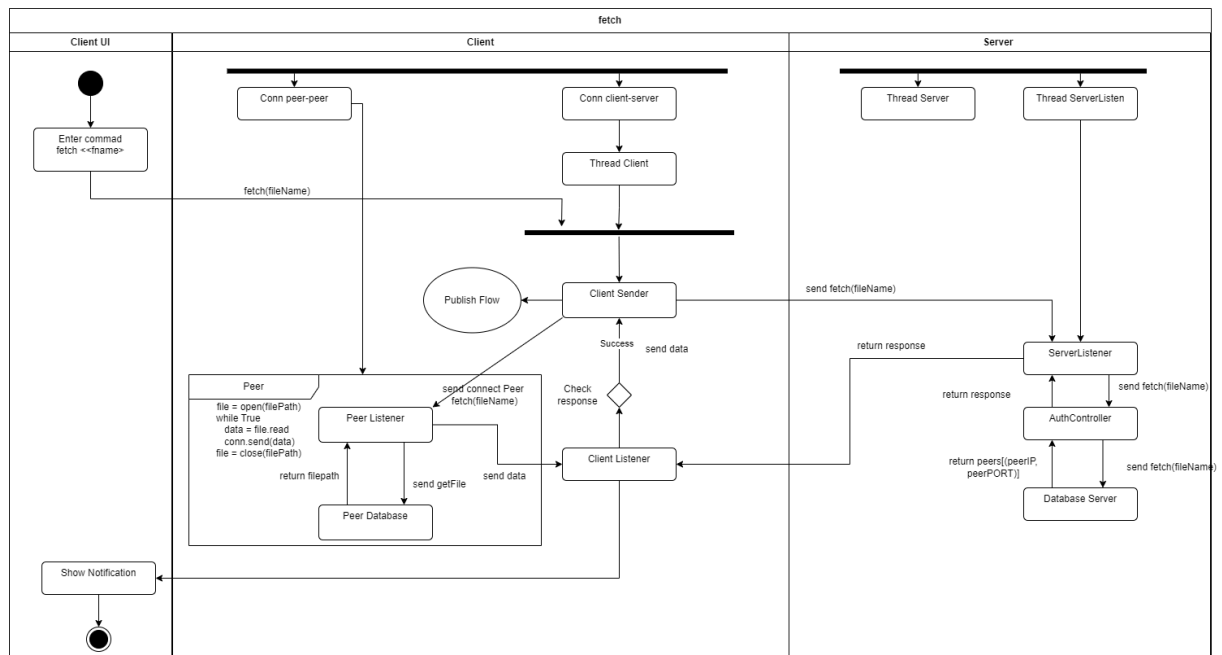


Figure 16: Sơ đồ hoạt động của chức năng fetch

Design UI

Register and Login

The screenshot shows a web application titled "File Sharing Application - Computer Networks Assignment - 2023 - HCMUT". The main heading is "Connect to the server". Below this, there are five input fields with labels: "Server:" (value: 10.0.104.92), "Username:", "Password:", "IP:" (value: 10.0.104.92), and "Peer Port:" (value: 5001). To the right of the "Server:" field is a label "IP Server". To the right of the "Username:" field is a label "Username". To the right of the "Peer Port:" field is a label "Peerport". To the left of the "Password:" field is a label "Password". To the left of the "IP:" field is a label "Peer IP". At the bottom of the form are two buttons: "Register" and "Login". The footer of the application shows "By: Minh Tuệ - Minh Lộc - Duy Tân" and "v1.0.0".

Figure 17: UI Register and Login

Đây là trang đăng ký, đăng nhập vào hệ thống bao gồm: địa chỉ IP Server, username, password, địa chỉ IP Peer, địa chỉ Peer Port.

Phía dưới có 2 nút: Register và Login.

Khi nhấn nút Register, hệ thống sẽ gửi thông tin đăng ký lên Server.

Khi nhấn nút Login, hệ thống sẽ gửi thông tin đăng nhập lên Server.

Publish Page

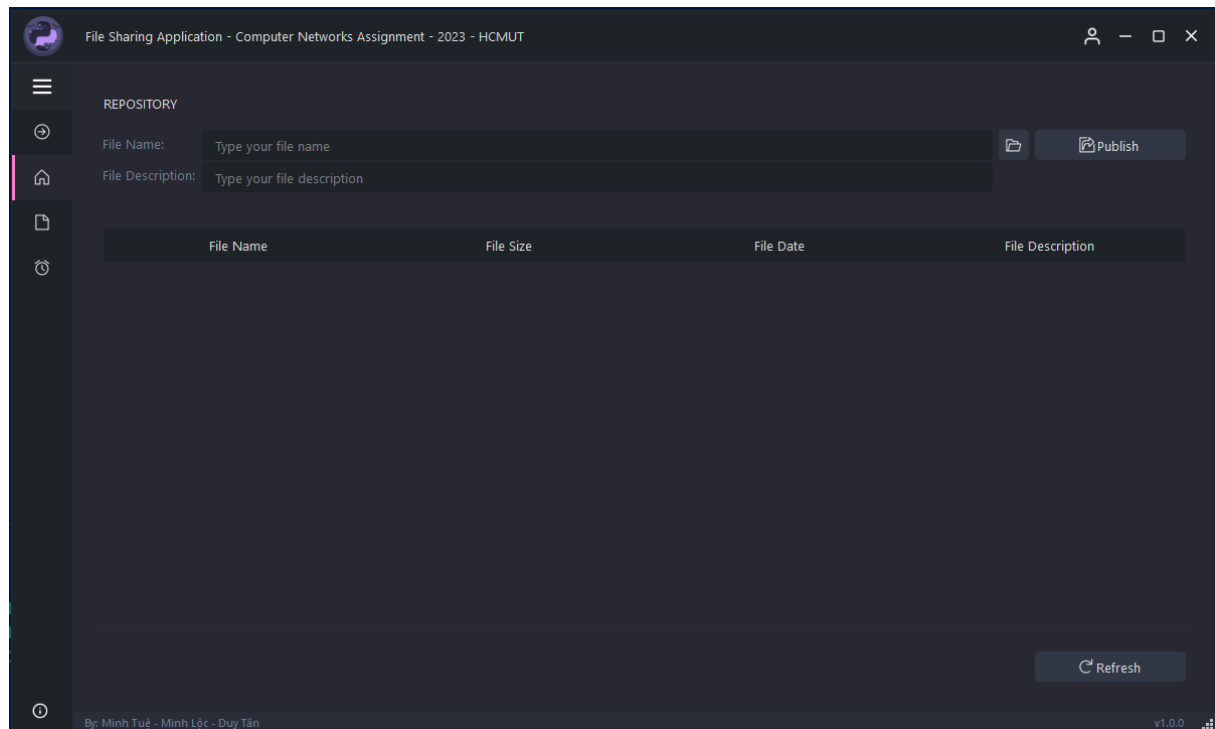


Figure 18: UI Publish File

Đây là trang Publish File, bao gồm: đường dẫn file, tên file fname sau khi publish, mô tả file, nút Publish.

Khi nhấn nút Publish, hệ thống sẽ gửi thông tin Publish lên Server.

Phía dưới là lưới hiển thị danh sách các file đã Publish.

Fetch Page

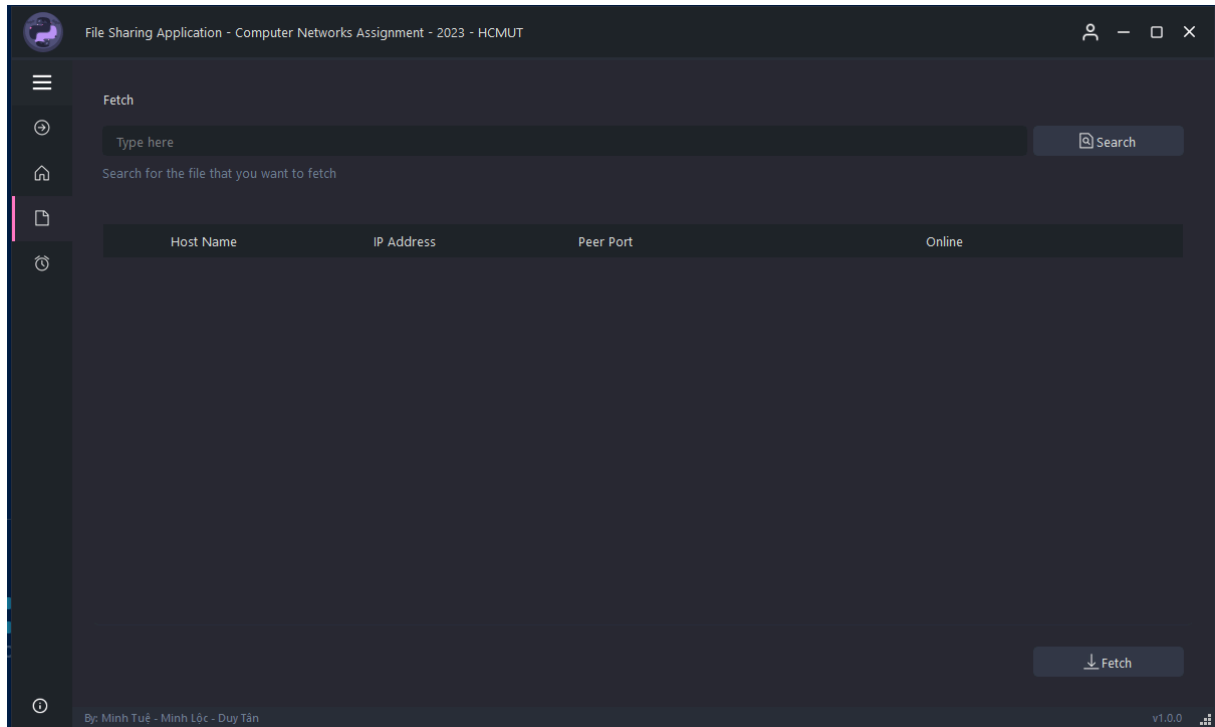


Figure 19: UI Fetch File

Đây là trang Fetch File, bao gồm: tên file fname cần Fetch, nút Search.
Sau khi Search, giao diện sẽ hiển thị danh sách các file có tên fname.
Người dùng có thể chọn 1 file trong danh sách và nhấn nút Fetch để Fetch file về.

Ngoài ra còn hiển thị thông tin người dùng: username, địa chỉ IP Peer, địa chỉ Peer Port.

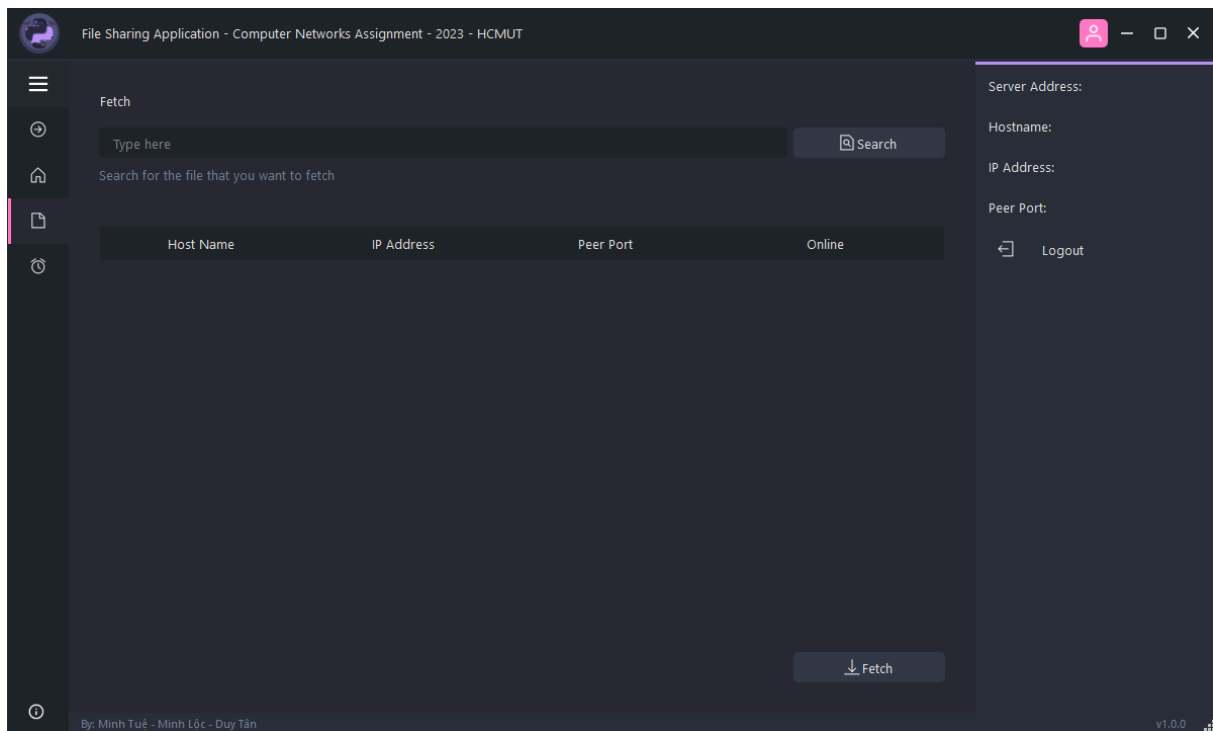


Figure 20: UI Profile

Tutorial - Guides

Server

Người dùng có thể start server bằng cách chạy file server.py

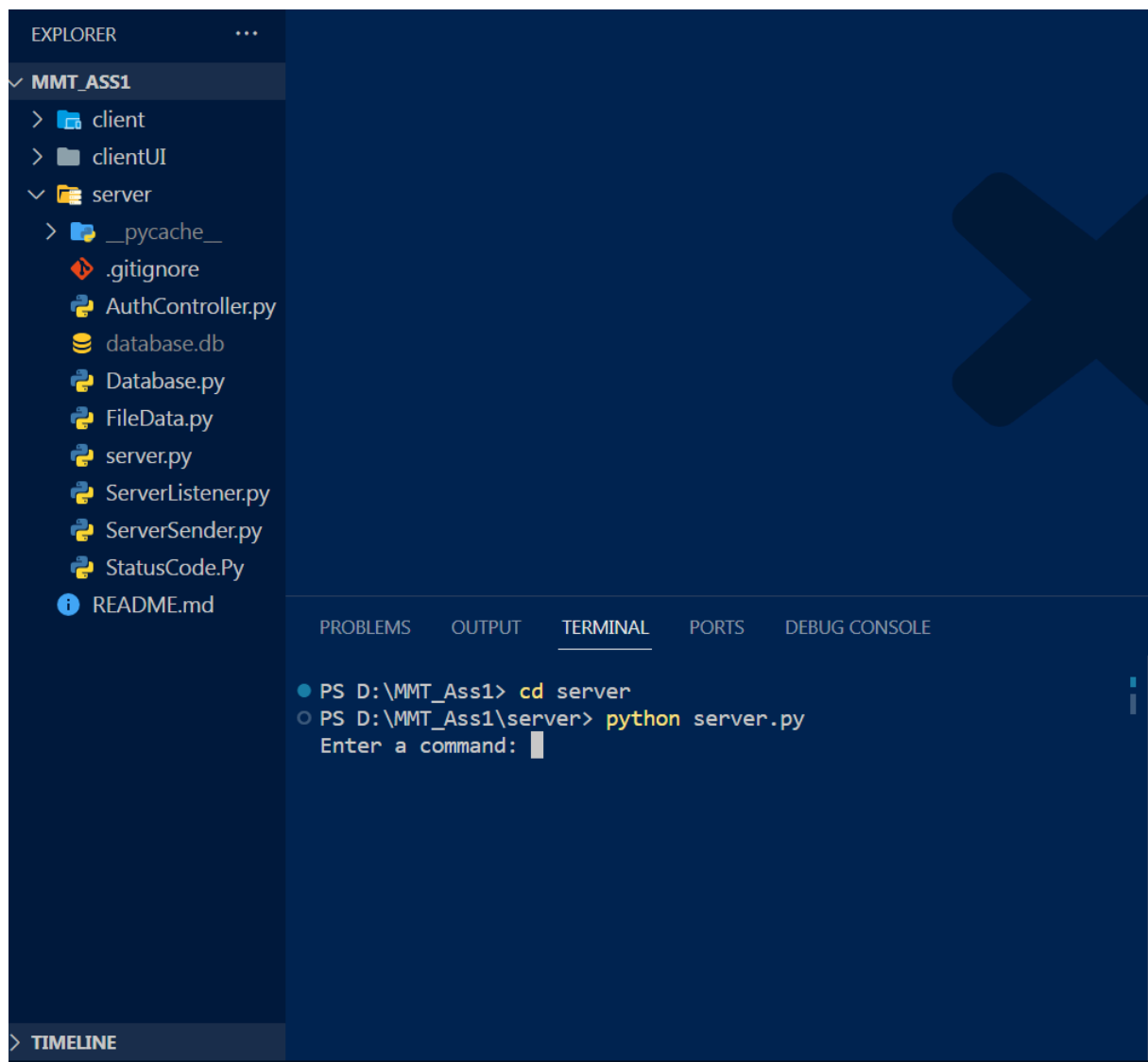


Figure 21: Start Server

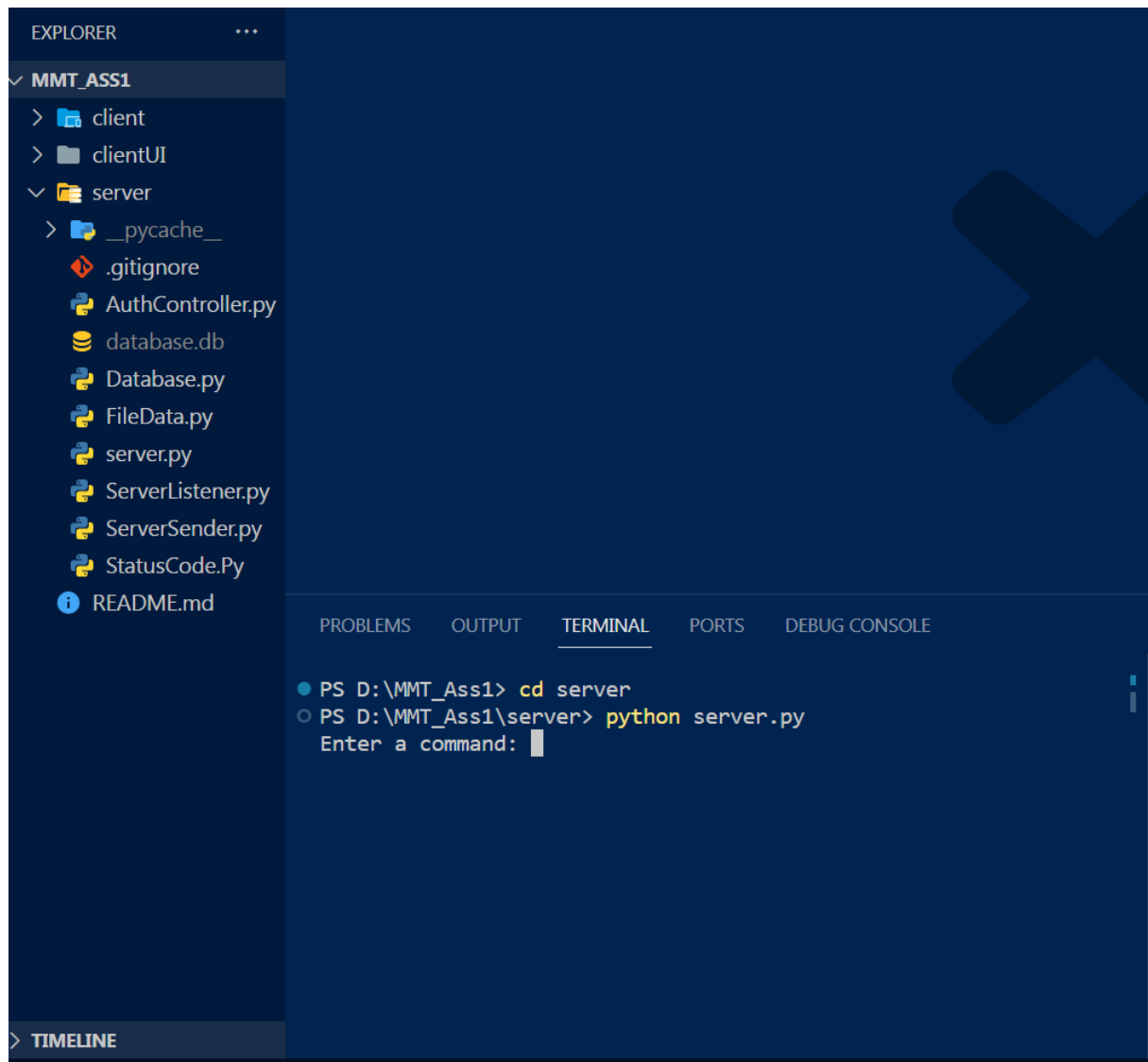


Figure 22: Start Server

ping

Server có thể ping tới các Peer khác bằng cách nhập ping <>

Nếu ping thành công, sẽ hiển thị thông báo client đang hoạt động

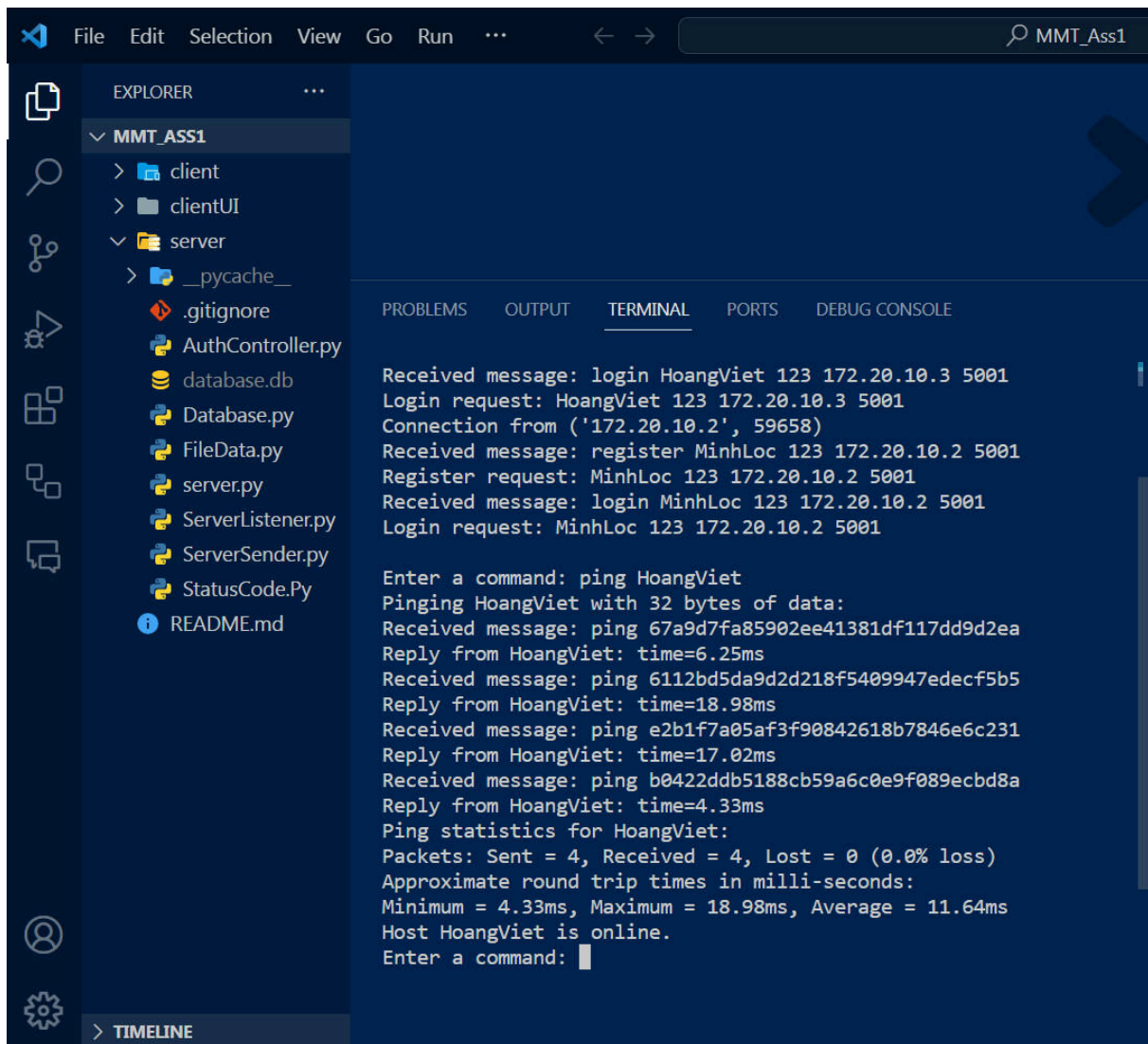
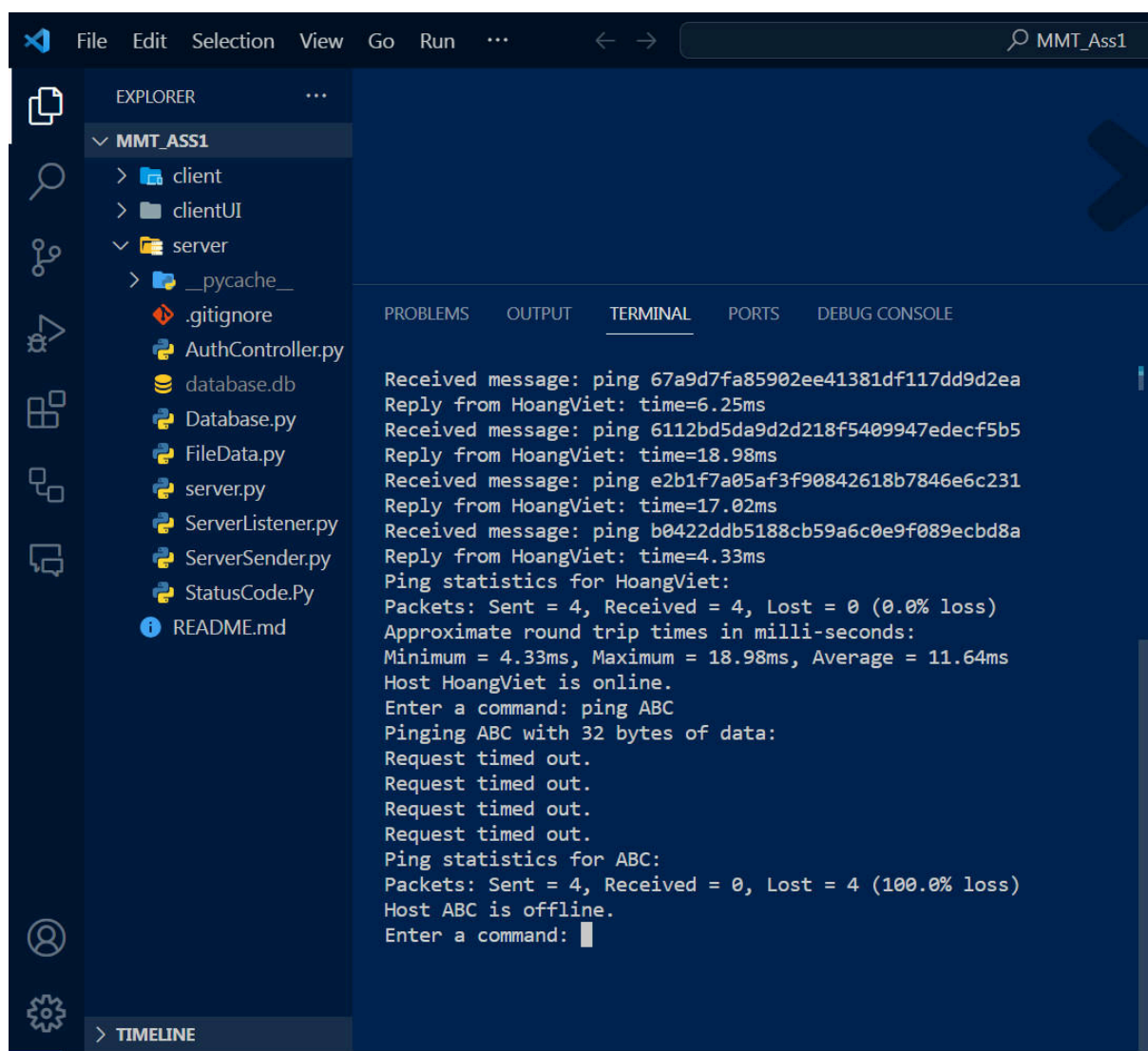


Figure 23: Ping

Nếu ping thất bại, sẽ hiển thị thông báo client không hoạt động



The screenshot shows a VS Code interface with a project named 'MMT_Ass1'. The Explorer panel on the left shows a file structure with folders 'client' and 'clientUI', and a 'server' folder containing files like '__pycache__', '.gitignore', 'AuthController.py', 'database.db', 'Database.py', 'FileData.py', 'server.py', 'ServerListener.py', 'ServerSender.py', 'StatusCode.py', and 'README.md'. The Terminal panel on the right shows the following output:

```
Received message: ping 67a9d7fa85902ee41381df117dd9d2ea
Reply from HoangViet: time=6.25ms
Received message: ping 6112bd5da9d2d218f5409947edecf5b5
Reply from HoangViet: time=18.98ms
Received message: ping e2b1f7a05af3f90842618b7846e6c231
Reply from HoangViet: time=17.02ms
Received message: ping b0422ddb5188cb59a6c0e9f089ecbd8a
Reply from HoangViet: time=4.33ms
Ping statistics for HoangViet:
Packets: Sent = 4, Received = 4, Lost = 0 (0.0% loss)
Approximate round trip times in milli-seconds:
Minimum = 4.33ms, Maximum = 18.98ms, Average = 11.64ms
Host HoangViet is online.
Enter a command: ping ABC
Pinging ABC with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for ABC:
Packets: Sent = 4, Received = 0, Lost = 4 (100.0% loss)
Host ABC is offline.
Enter a command: 
```

Figure 24: Ping No Online

Discover

Server có thể discover các Peer khác bằng cách nhập discover

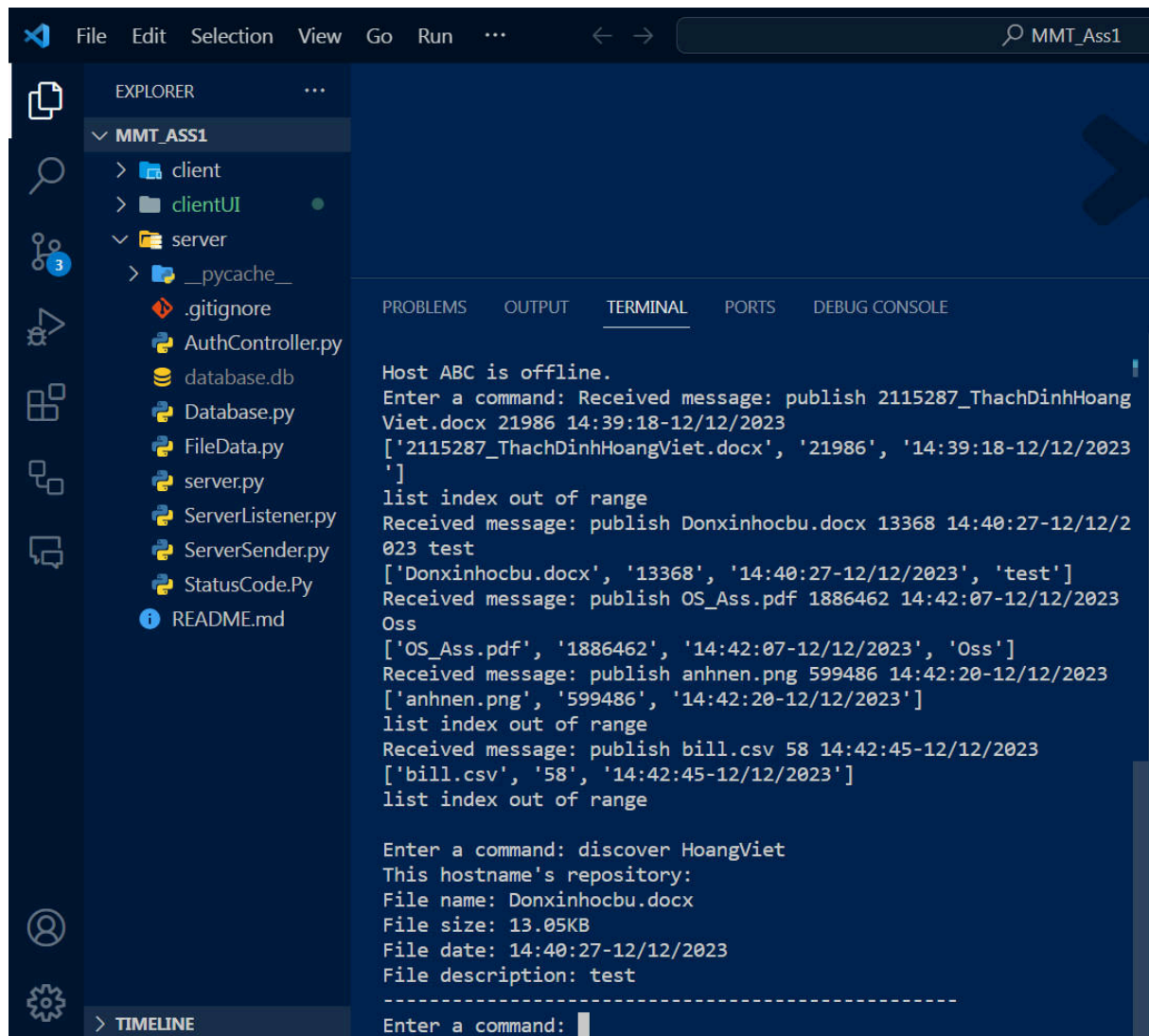
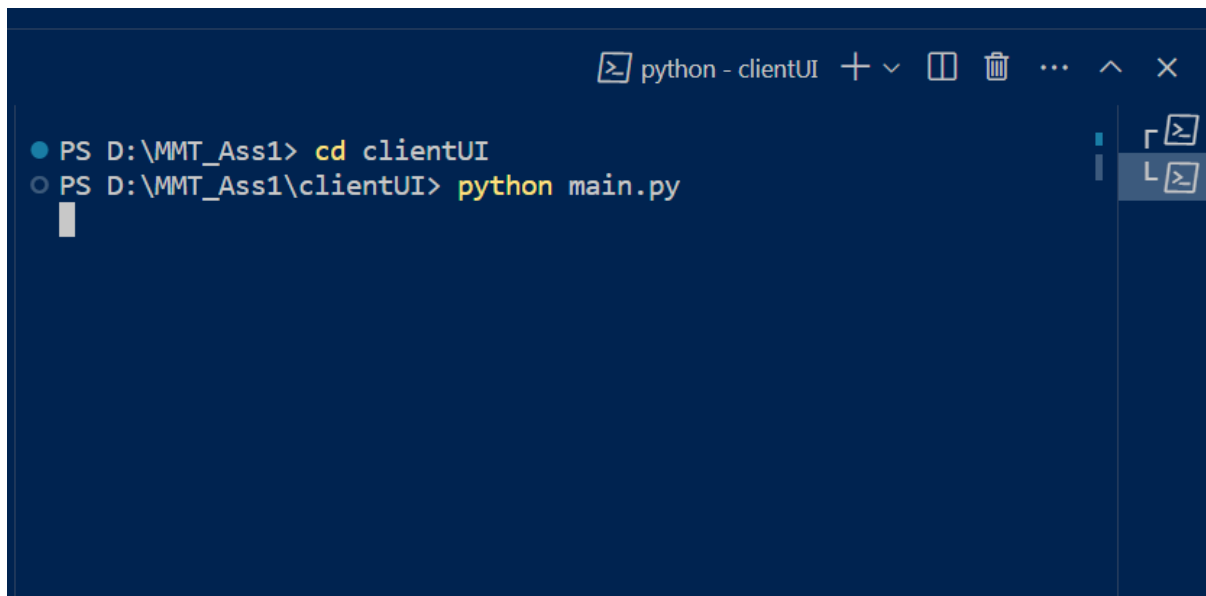


Figure 25: Discover

Client

Người dùng có thể start client bằng cách chạy file main.py

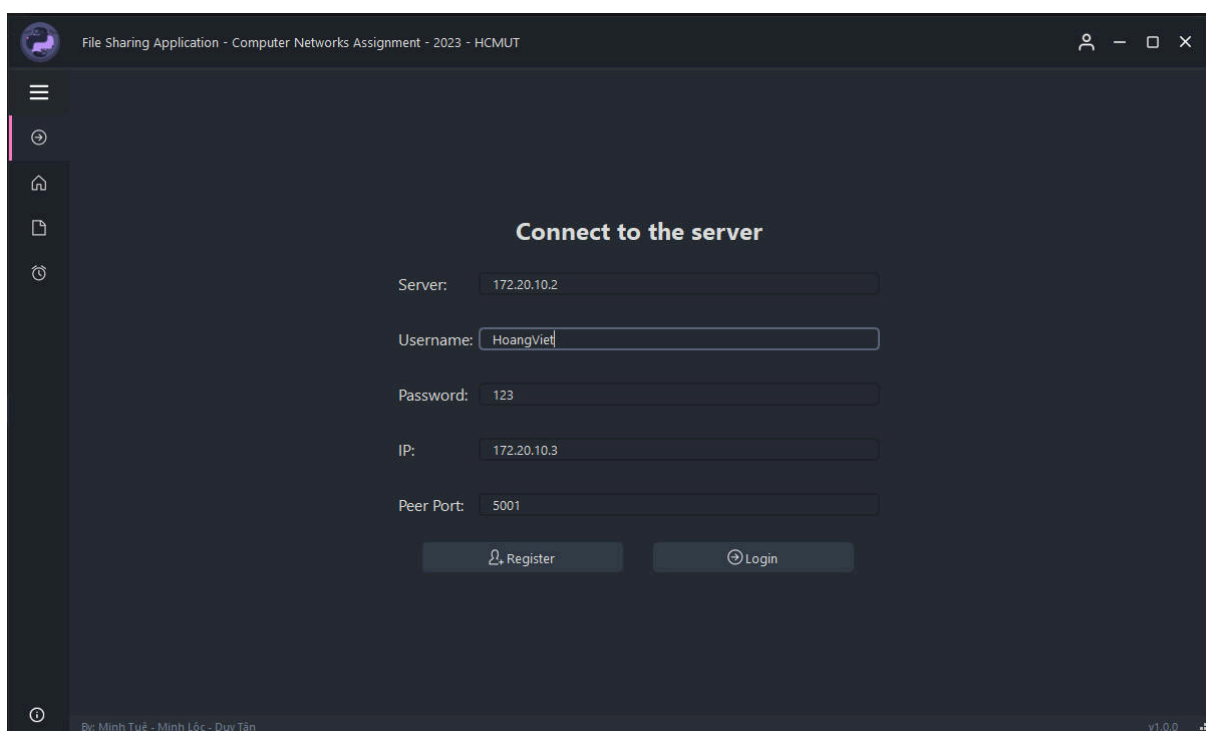


```
python - clientUI
PS D:\MMT_Ass1> cd clientUI
PS D:\MMT_Ass1\clientUI> python main.py
```

Figure 26: Start Client

Register

Người dùng sẽ nhập IP Server, username, password, địa chỉ IP Peer, địa chỉ Peer Port để đăng ký tài khoản trên hệ thống.



File Sharing Application - Computer Networks Assignment - 2023 - HCMUT

Connect to the server

Server: 172.20.10.2

Username: HoangViet

Password: 123

IP: 172.20.10.3

Peer Port: 5001

Register Login

By: Minh Tuệ - Minh Lộc - Duy Tân v1.0.0

Figure 27: Client Register

Giao diện sẽ hiển thị thông báo đăng ký thành công

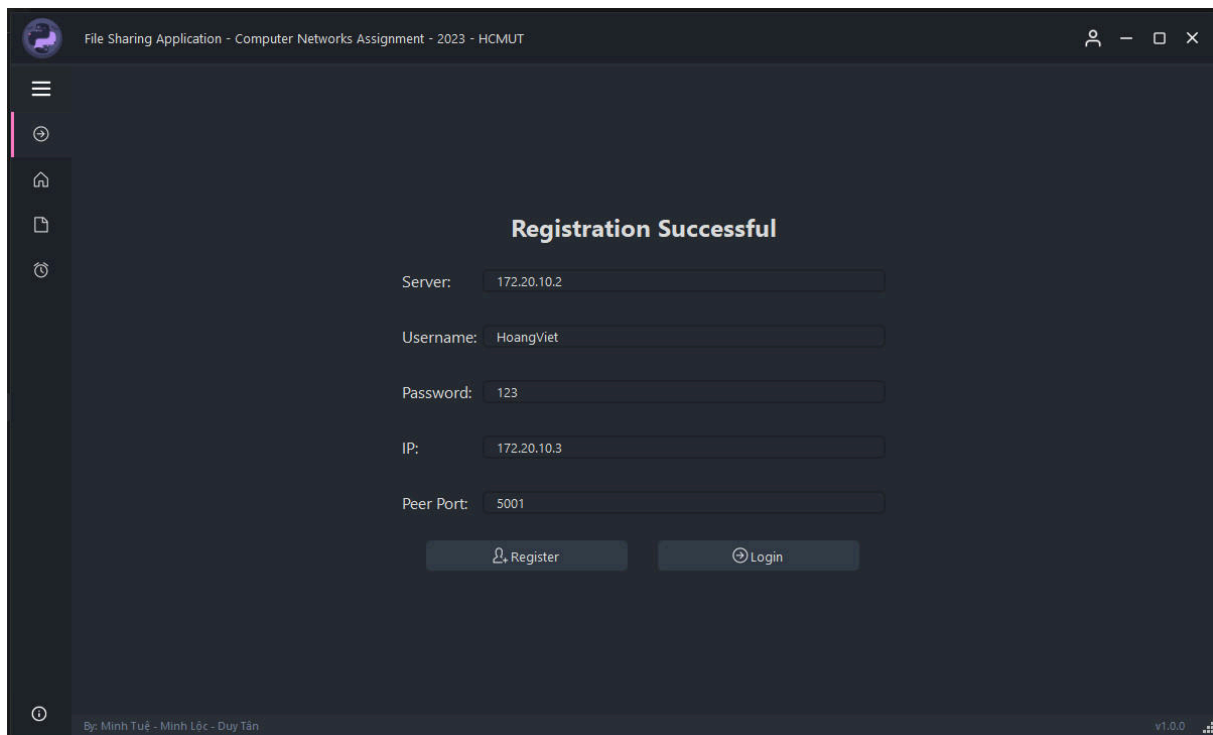


Figure 28: Client Register Success

Phía Server sẽ hiển thị thông báo đăng ký thành công

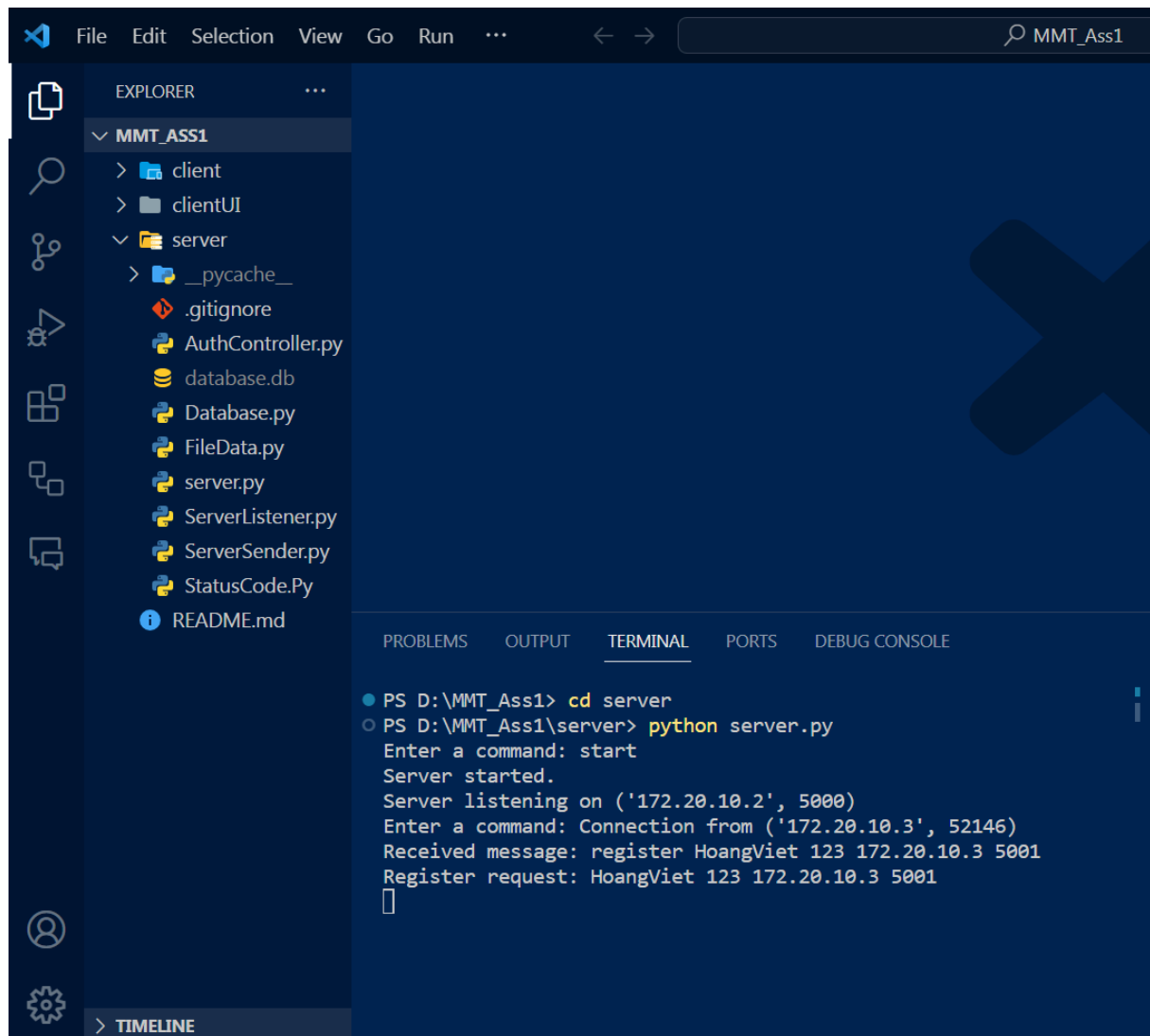


Figure 29: Server Register Success

Login

Người dùng sẽ nhập IP Server, username, password, địa chỉ IP Peer, địa chỉ Peer Port để đăng nhập vào hệ thống.

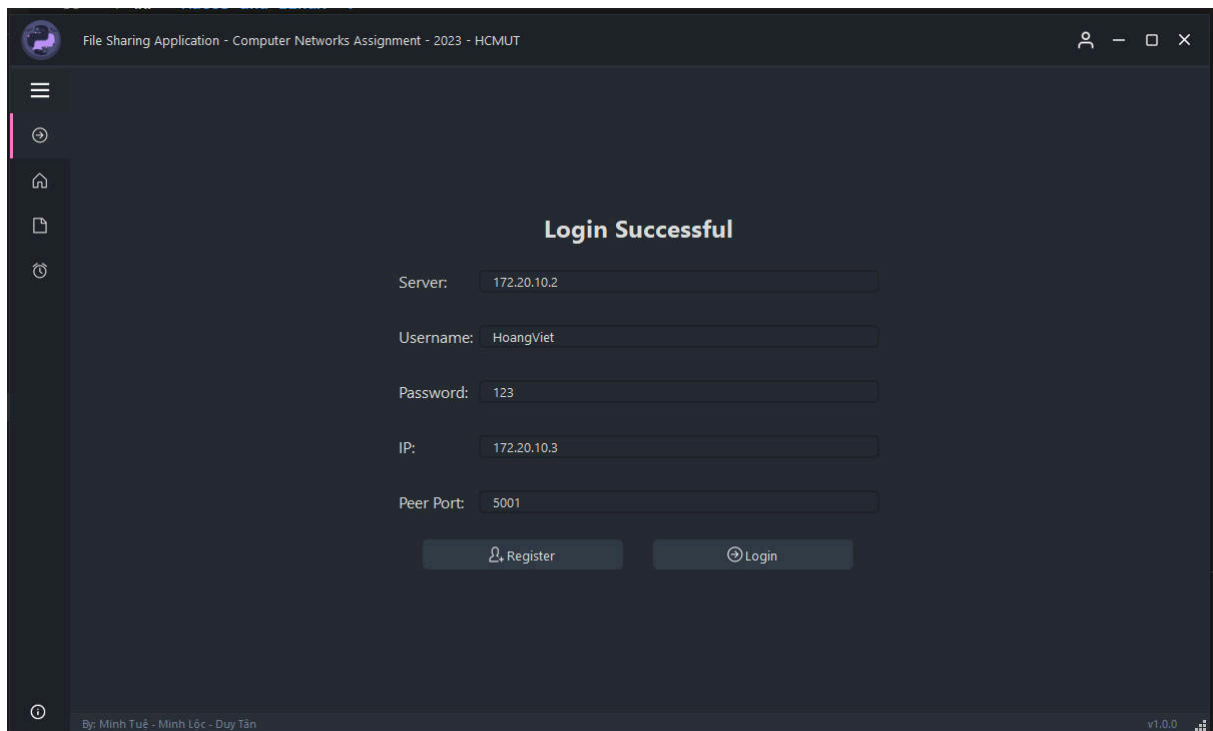


Figure 30: Client Login

Phía Server sẽ hiển thị thông báo đăng nhập thành công

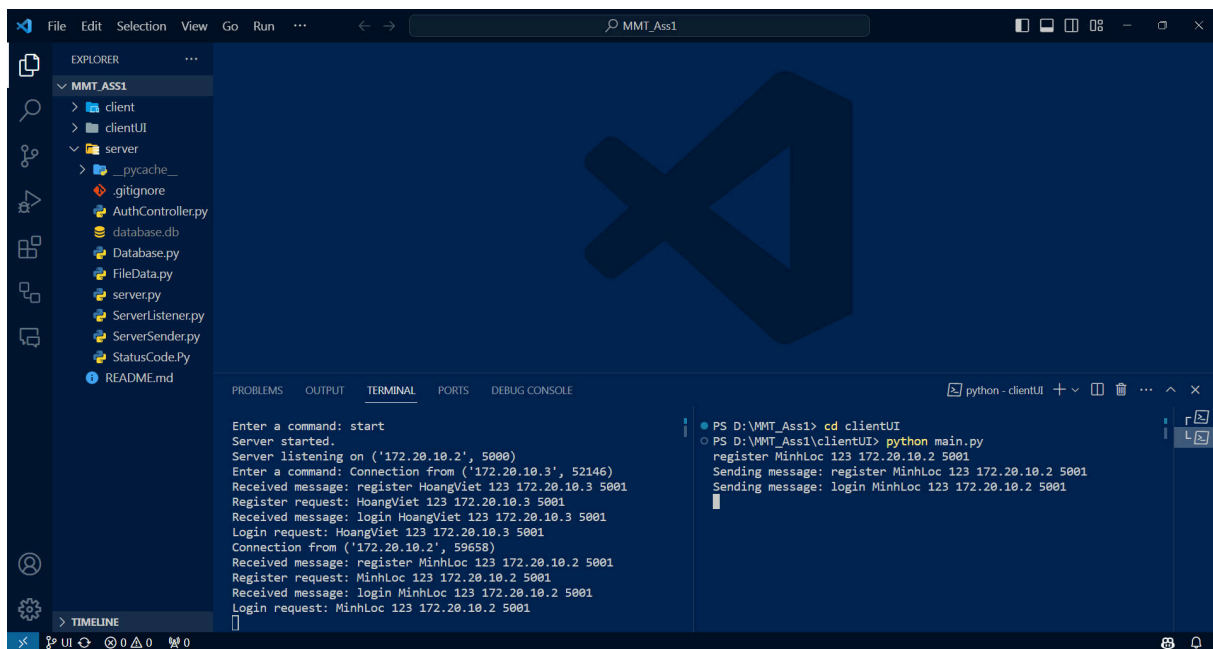


Figure 31: Login Success

Publish

Người dùng sẽ nhập đường dẫn file, mô tả file và nhấn nút Publish để Publish file lên hệ thống.

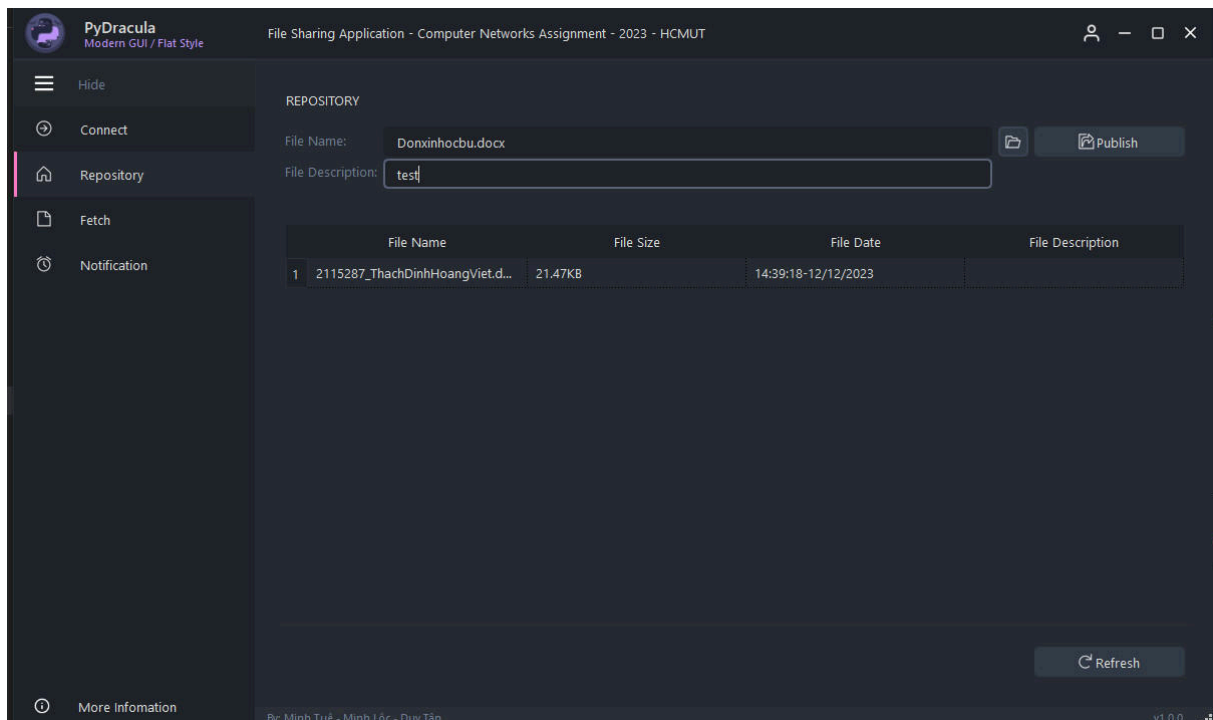


Figure 32: Publish

Giao diện sẽ hiển thị file đã Publish vào danh sách

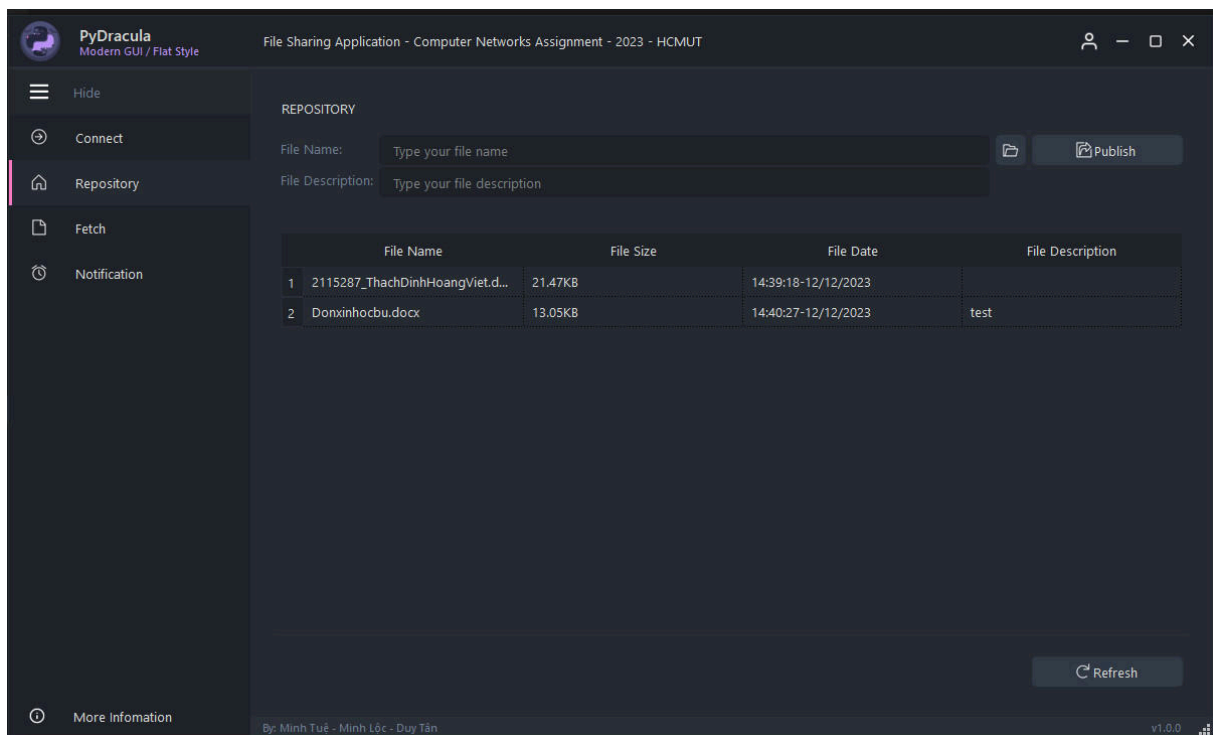


Figure 33: Publish Success

Fetch

Người dùng sẽ nhập tên file cần Fetch và nhấn nút Search để tìm kiếm file.

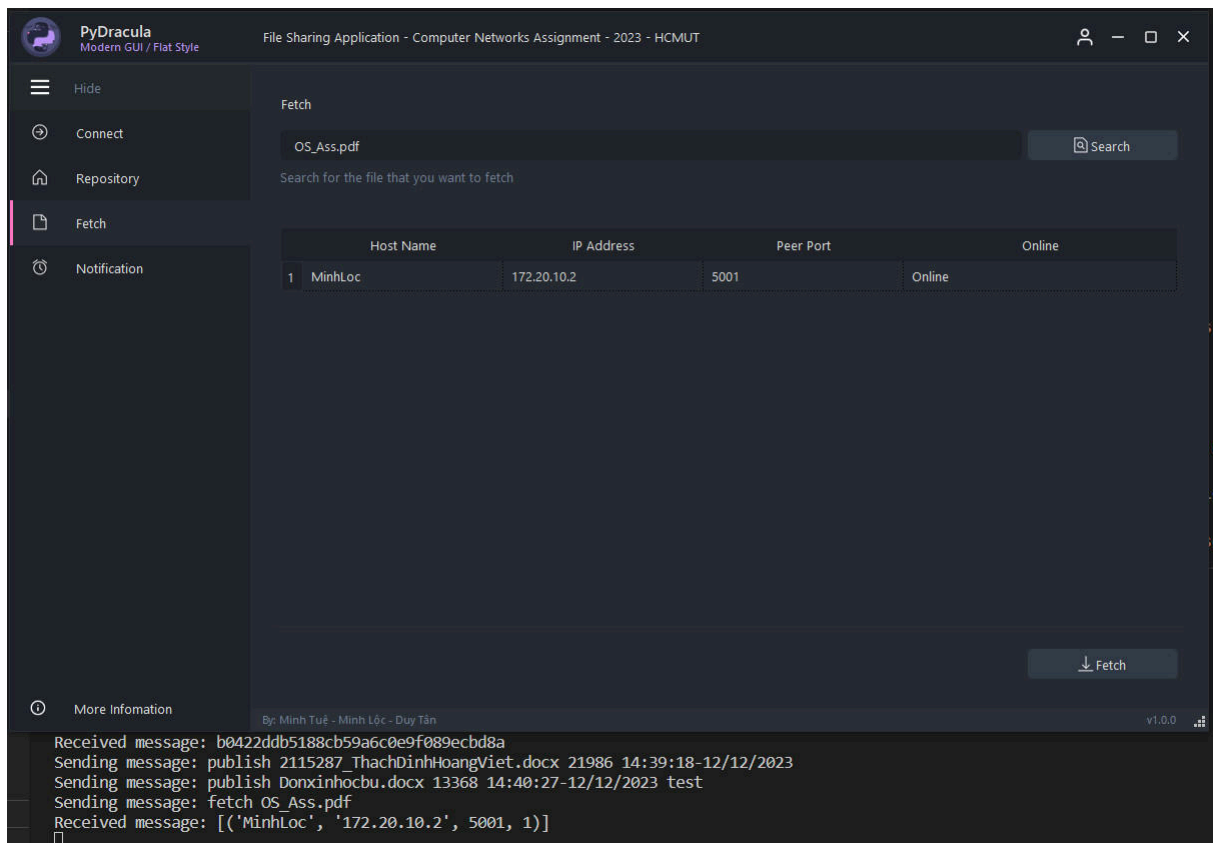


Figure 34: Fetch

Sau khi chọn file cần Fetch và nhấn nút Fetch, file sẽ được Fetch về.

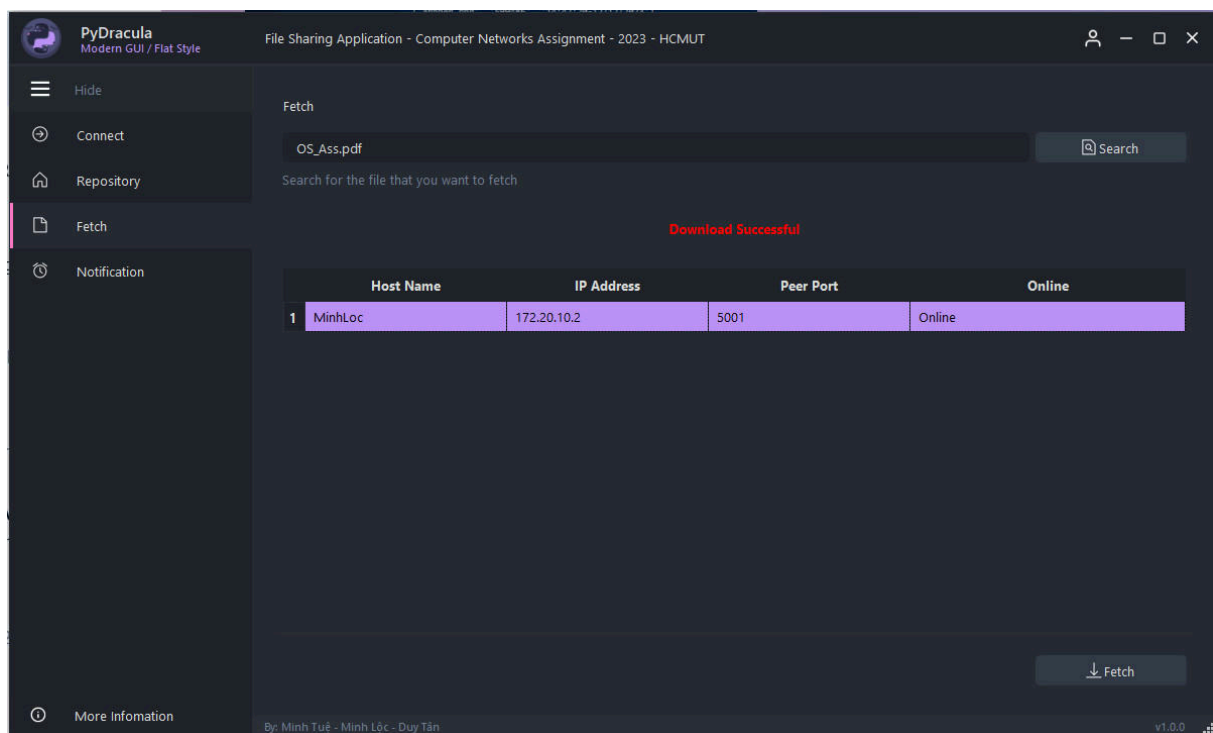


Figure 35: Fetch Success

Lưới danh sách file sẽ hiển thị file đã Fetch về.

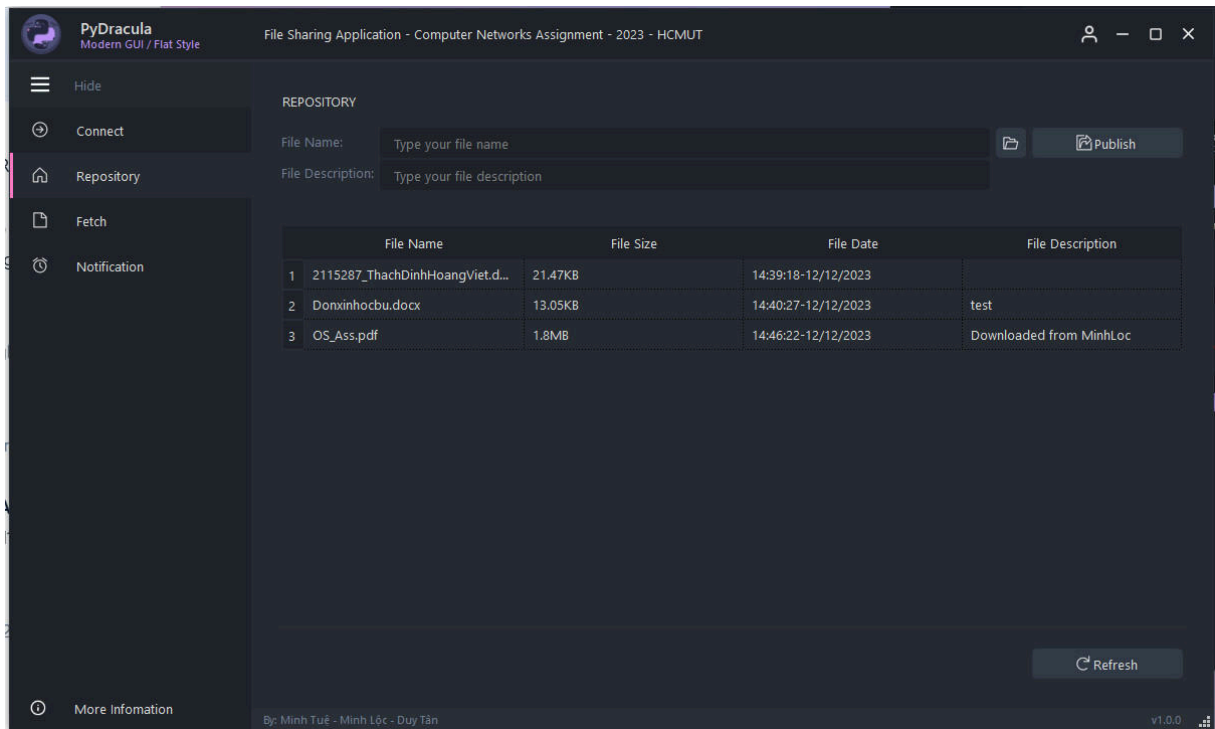


Figure 36: Fetch Success

Error Handling