ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

KHOA CÔNG NGHỆ THÔNG TIN

--oOo--

ĐỒ ÁN MÔN HỌC JAVA

Sinh viên thực hiện:	20120065 – Lâm Quang Duy
Lớp:	21_6
Môn học:	Lập trình ứng dụng Java
Tên đồ án:	Networking
Giảng viên lý thuyết:	Nguyễn Văn Khiết
Giảng viên thực hành:	Nguyễn Lê Hoàng Dũng
	Trương Phước Lộc

Thành phố Hồ Chí Minh, ngày 6, tháng 04, năm 2023

I. Contents

II. Danh sách chức năng

Tên chức năng	Mức độ hoàn thành
Quản lý danh sách client	100%
Giám sát hoạt client đang kết nối	100%
Giám sát thư mục của client	100%
Giám sát nhiều client cùng lúc	100%
Giao diện đồ hoạ	100%

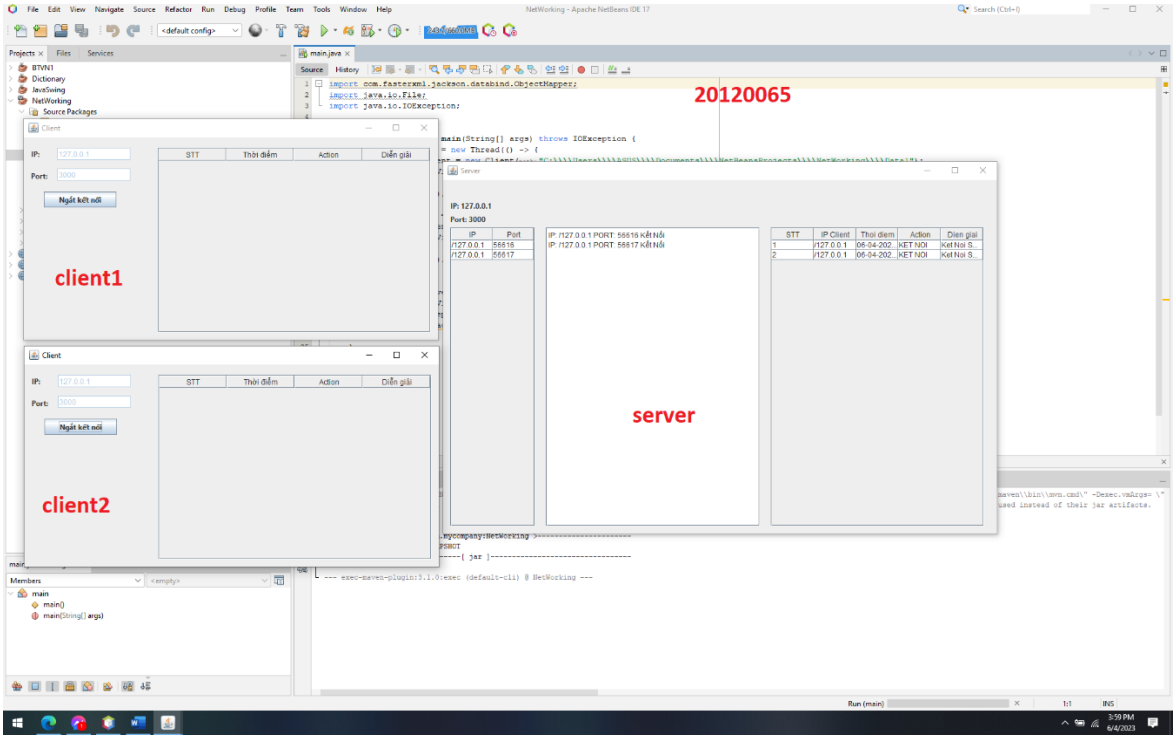
III. Điểm tự đánh giá

Đồ án hoàn thành đầy đủ các yêu cầu về mặt chức năng, tuy nhiên chưa xuất được file jar theo yêu cầu. Do đó em hi vọng sẽ được nhận điểm số ở mức 9.5+.

IV. Giao diện

1. Tổng quan

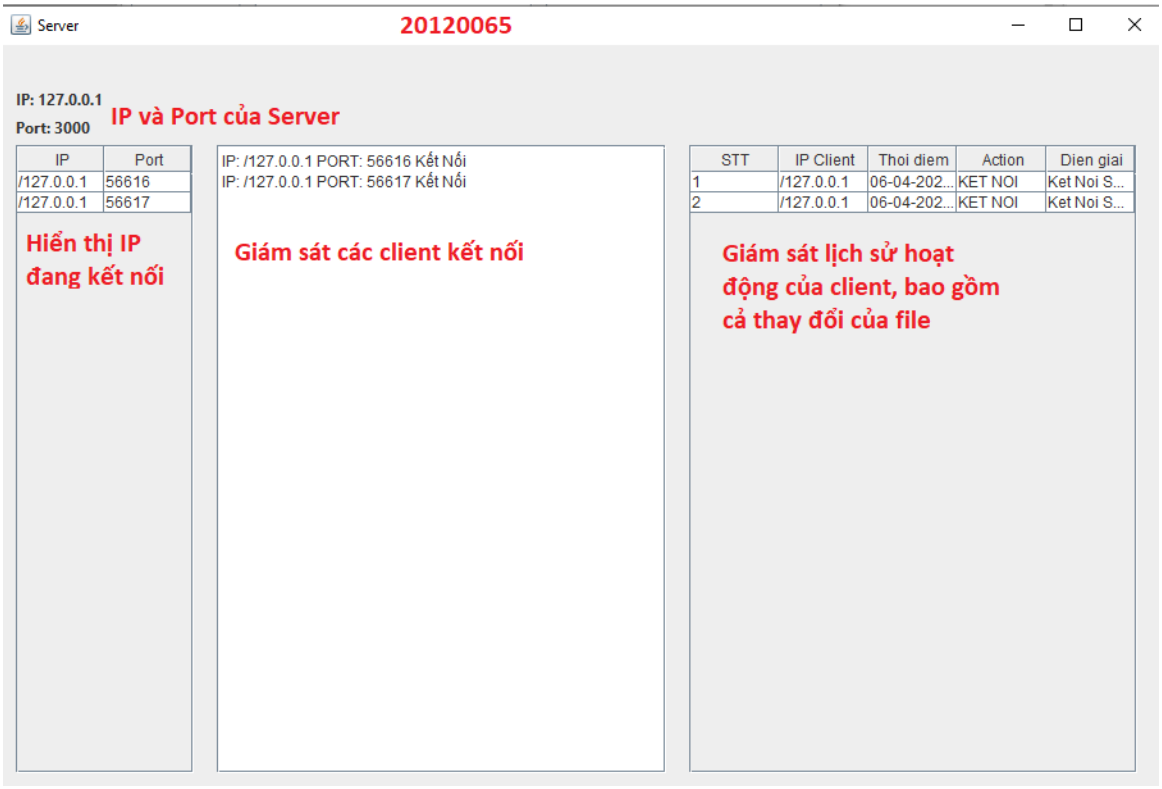
Giao diện tổng quan của ứng dụng khi khởi động gồm 1 server và 2 client.



2. Server

Server được tạo thành từ 4 phần chính:

- Hiển thị IP, Port.
- Danh sách IP kết nối.
- Giám sát các client kết nối.
- Giám sát lịch sử hoạt động của client.



3. Client

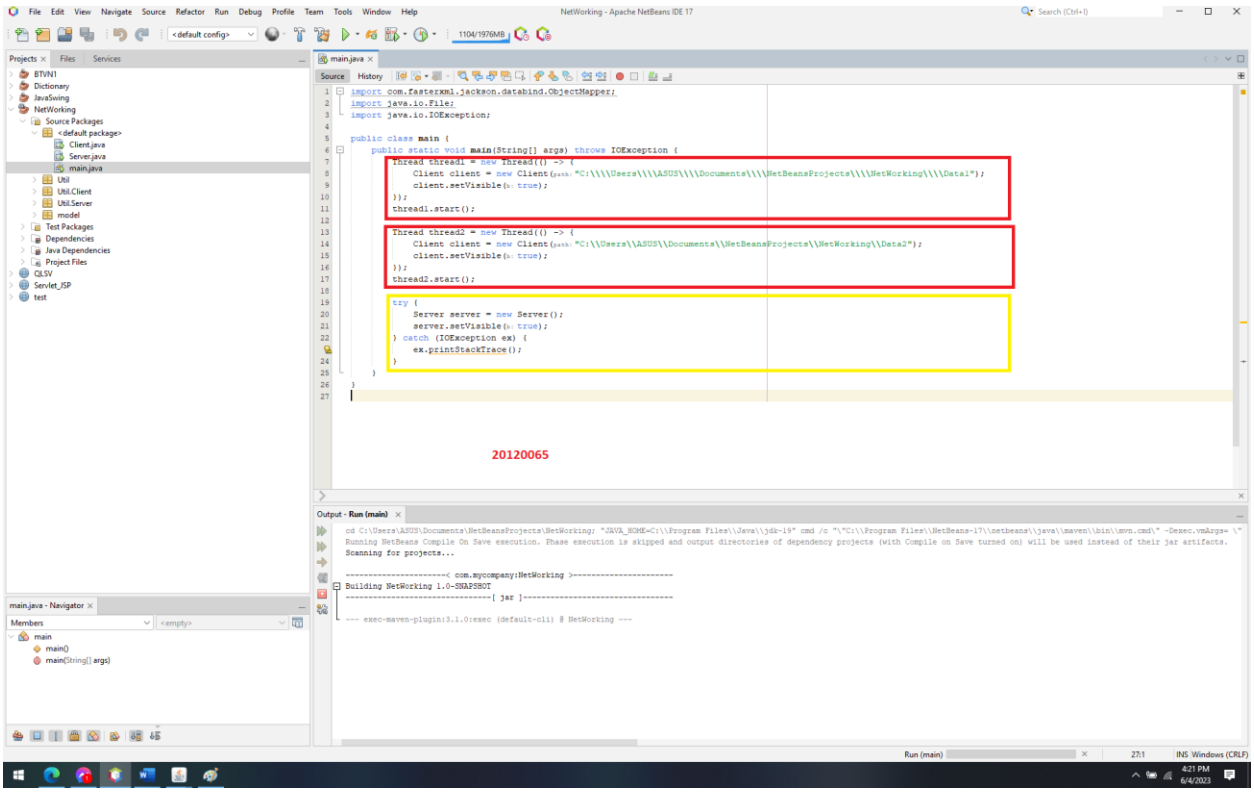
Client được chia thành 3 phần chính:

- Hiển thị ô trống để người dùng nhập IP, Port để kết nối với Server.
- Button kết nối
- Hiển thị giám sát thư mục

V. Giải thích code

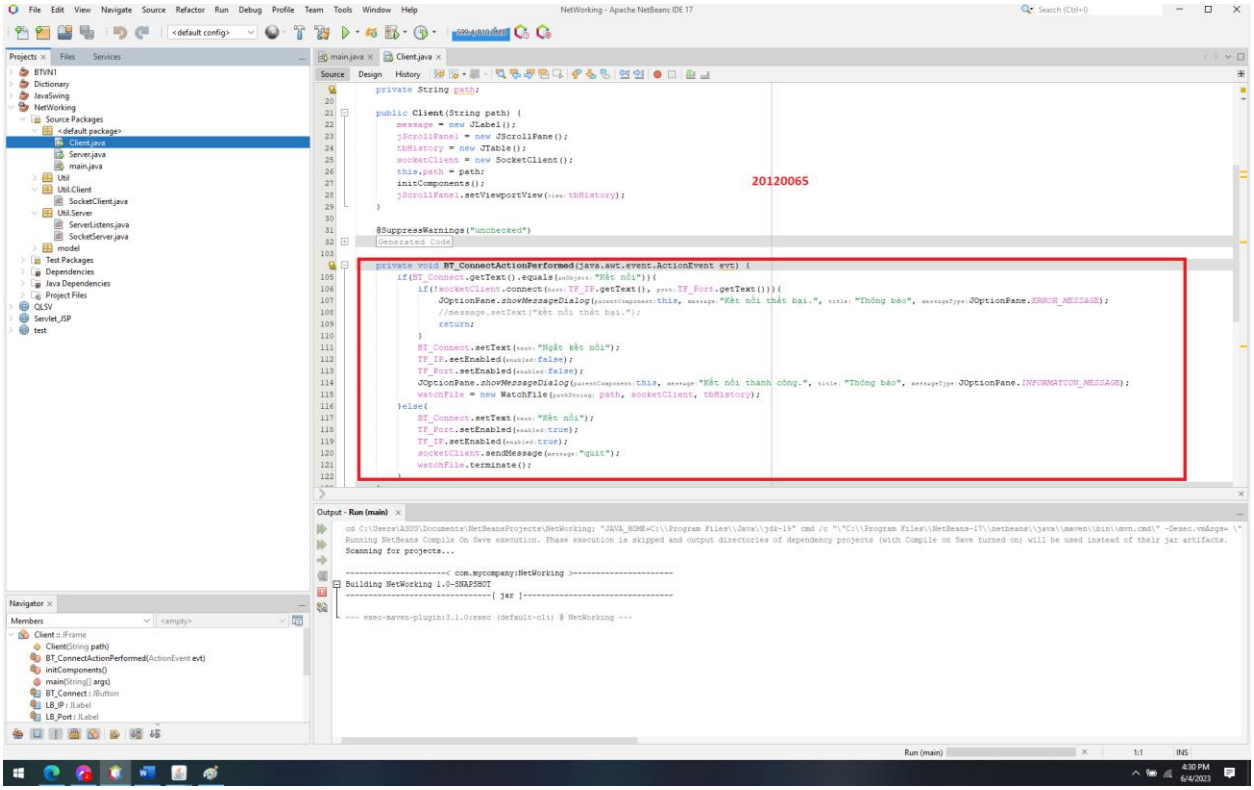
1. Main.java

Hàm main khởi tạo 2 jFrame Clinet (phần khoanh đỏ) và 1 jFrame Server (phần khoanh vàng). Để thực thi chương trình chỉ cần cho khởi chạy hàm main. Cùng lúc khởi tạo 2 Client, chương trình đồng thời gán đường dẫn tĩnh để giám sát các thư mục, client1 và client2 sẽ giám sát các thư mục tương ứng là Data1 và Data2. Do đó, trong trường hợp người dùng cài đặt chương trình về máy để sử dụng, cần phải thay đổi đường dẫn sang thư mục phù hợp cần giám sát.



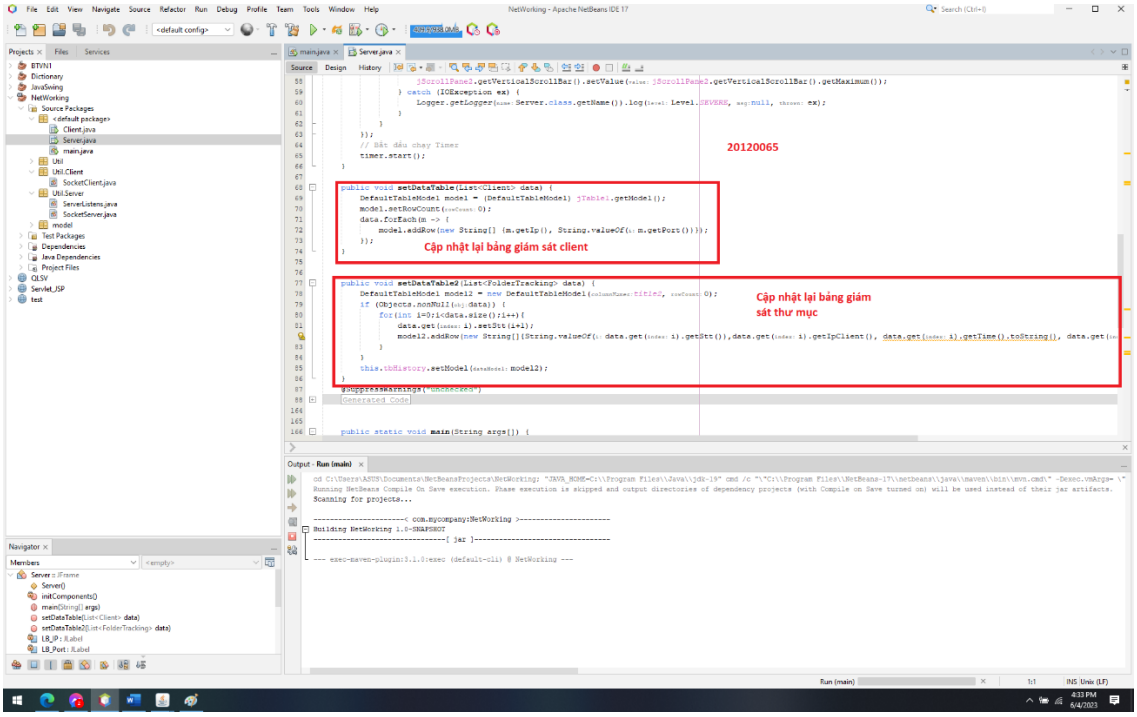
2. Client.java

Jframe này được tạo ra để thiết kế giao diện cũng như các sự kiện của Client. Trong đó sự kiện đối với button kết nối là một sự kiện quan trọng giúp Client trao đổi thông tin được với server, hàm sử lý trao đổi thông qua Socket Client (nằm trong package Util.Client).

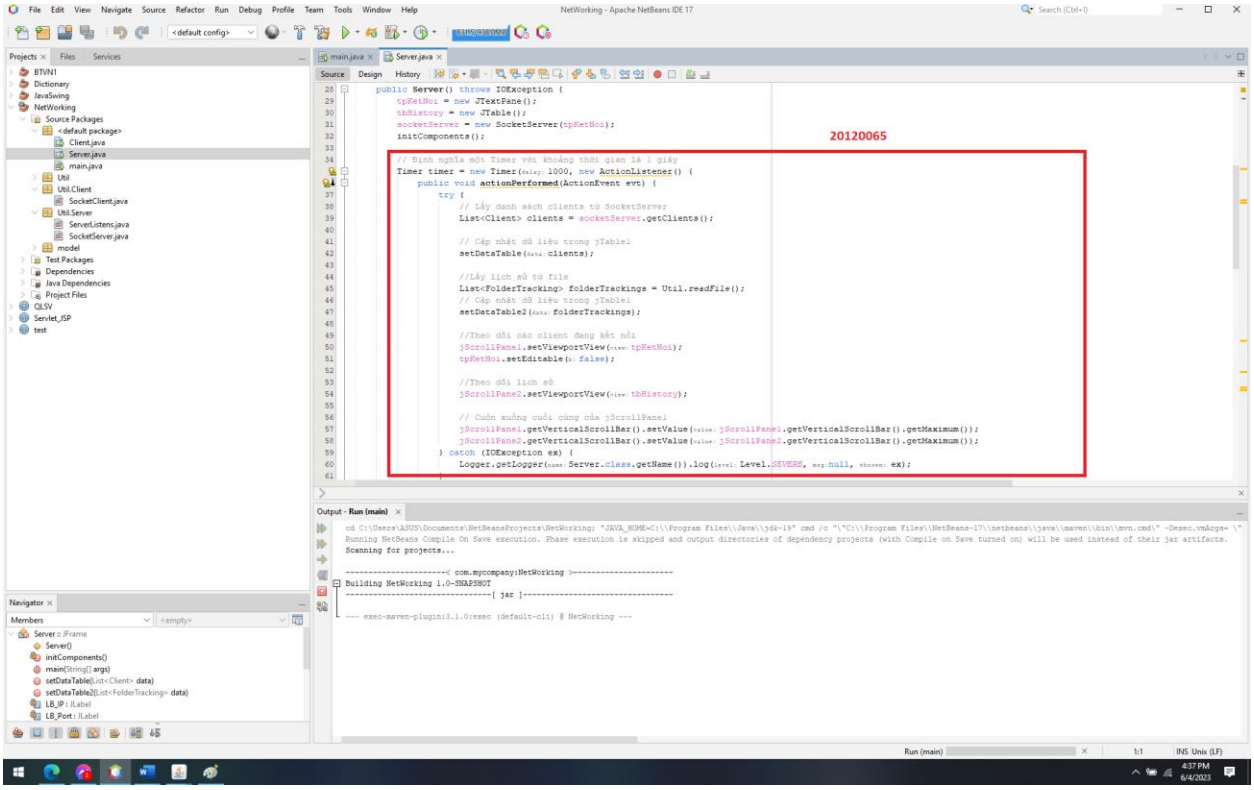


3. Server.java

jFrame này được thiết kế để tạo giao diện và xử lý các sự kiện của Server. Ảnh phía dưới được sử dụng để cập nhật 2 thành phần quan trọng là hiển thị các hoạt động của client và giám sát file.



Ngoài ra, trong file còn có xử lý cập nhật danh sách các client đã kết nối và gọi hàm thực thi của 2 hàm cập nhật đã nói phía trên. Ở đây có sử dụng kỹ thuật Timer để các hàm cập nhật được thực thi liên tục mỗi giây 1 lần.



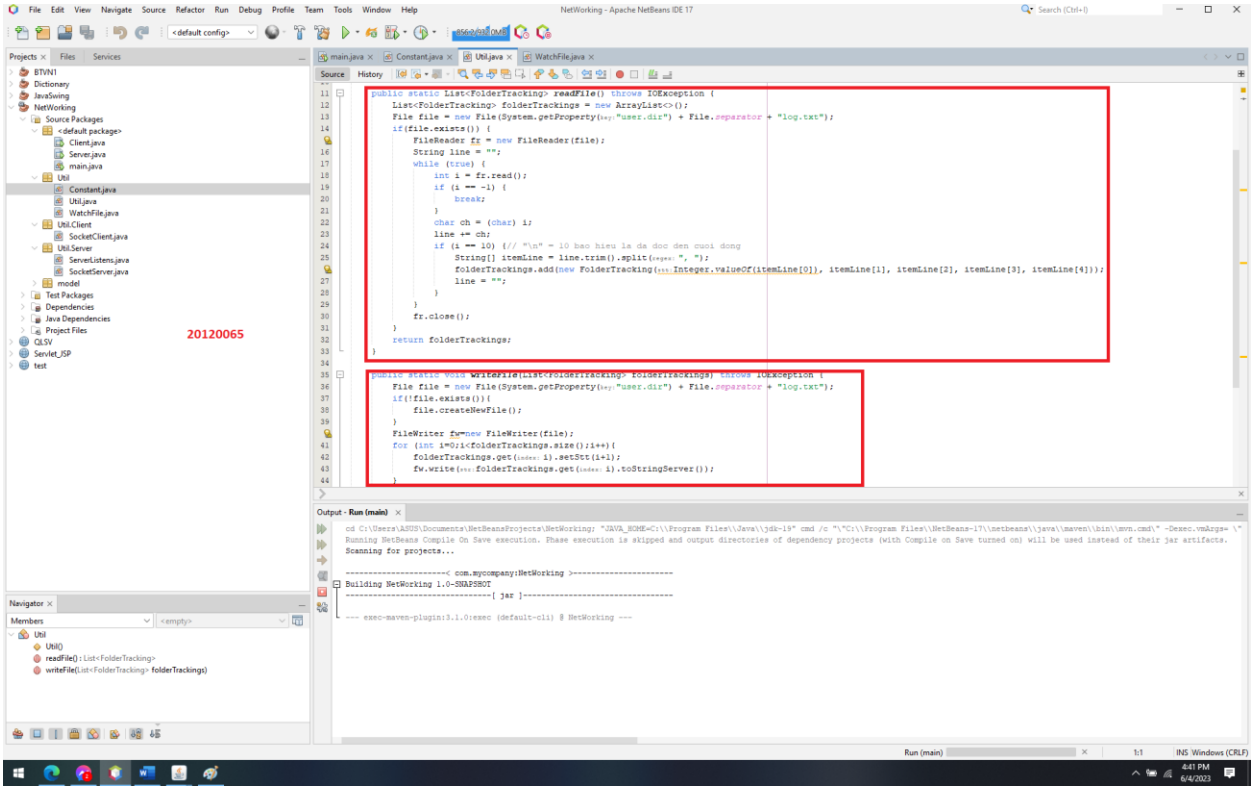
4. Package Util

Package này được tạo ra để xử lý các hoạt động liên quan tới việc giám sát thư mục, và lưu lại lịch sử của việc giám sát.

a) Util.java

File này xử lý 2 phần đọc và ghi dữ liệu vào tệp log.txt để phục vụ cho việc lưu trữ và hiển thị lịch sử giám sát thư mục.

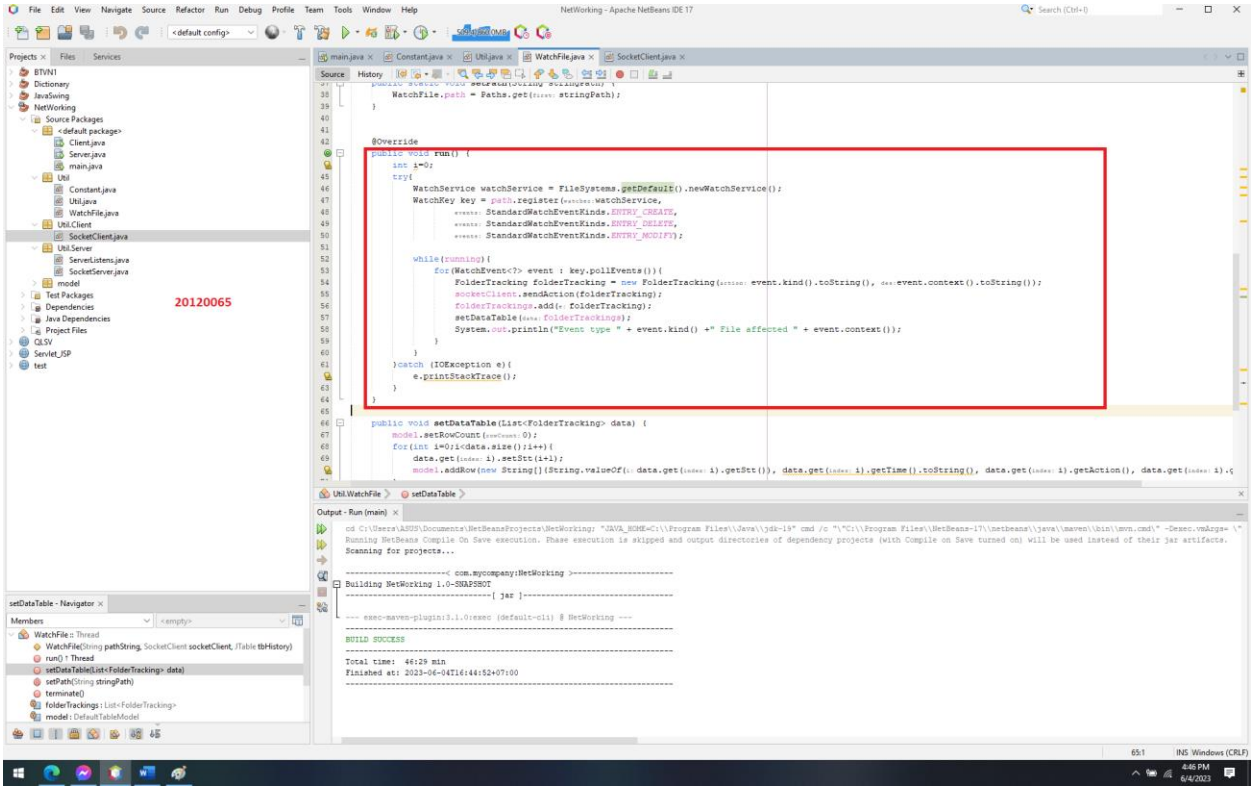
- Phương thức readFile() đọc nội dung từ tệp tin log.txt và trả về một danh sách các đối tượng FolderTracking. Quá trình đọc tệp tin được thực hiện bằng cách sử dụng lớp FileReader để đọc từng ký tự và chia dòng thành các mục bằng dấu ,. Sau đó, các mục này được sử dụng để tạo các đối tượng FolderTracking, và được thêm vào danh sách folderTrackings.
- Phương thức writeFile() ghi dữ liệu từ danh sách các đối tượng FolderTracking vào tệp tin log.txt. Quá trình ghi dữ liệu được thực hiện bằng cách sử dụng lớp FileWriter để ghi các chuỗi tương ứng với các đối tượng FolderTracking vào tệp tin.



b) WatchFile.java

File này được sử dụng để giám sát sự thay đổi trong thư mục.

- Lớp này được kế thừa từ lớp Thread, cho phép nó chạy riêng biệt.
- Lớp này sử dụng Watch Service API của Java để giám sát sự thay đổi trong thư mục được chỉ định.
- Trong phương thức run(), lớp WatchFile lặp lại việc kiểm tra các sự kiện xảy ra trong thư mục được giám sát. Mỗi khi có sự kiện xảy ra (tạo, xóa hoặc sửa đổi tệp tin), lớp ghi lại thông tin về sự kiện và gửi nó cho một đối tượng SocketClient thông qua phương thức sendAction().
- Lớp WatchFile cũng duy trì một danh sách các đối tượng FolderTracking, chứa thông tin về các sự kiện xảy ra trong thư mục. Phương thức setDataTable() được sử dụng để cập nhật dữ liệu trong một JTable (được truyền vào qua tham số tbHistory) với thông tin từ danh sách FolderTracking.
- Lớp WatchFile có một phương thức terminate() để dừng quá trình giám sát thư mục bằng cách đặt cờ running thành false.



5. Package Util.Client

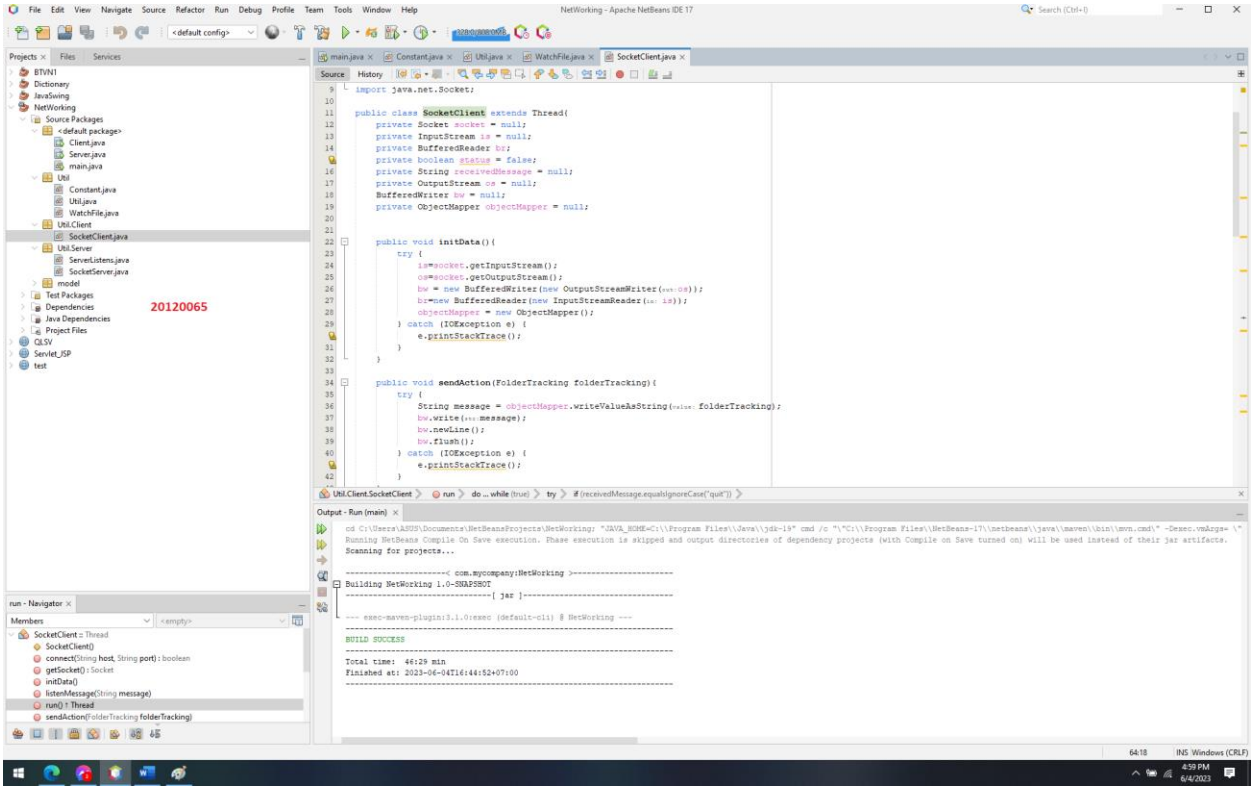
a) SocketClient.java

File này được tạo ra để xử lý phía backend của Client, cung cấp các phương thức để tạo và quản lý kết nối tới một máy chủ socket, gửi và nhận thông điệp thông qua giao thức Socket.

- Lớp SocketClient kế thừa từ lớp Thread, cho phép nó chạy trên một luồng riêng biệt.
- Trong phương thức initData(), lớp SocketClient thiết lập luồng dữ liệu vào ra (input/output stream) từ socket để gửi và nhận dữ liệu.
- Phương thức sendAction() nhận một đối tượng FolderTracking và chuyển đổi nó thành chuỗi JSON bằng thư viện ObjectMapper. Sau đó, chuỗi JSON được gửi đi qua socket.
- Phương thức sendMessage() nhận một chuỗi thông điệp và gửi nó đi qua socket.
- Trong phương thức run(), lớp SocketClient lắng nghe các thông điệp nhận được từ socket thông qua BufferedReader. Nếu thông điệp là "quit", kết nối

sẽ được đóng. Nếu không, phương thức listenMessage() được gọi để xử lý thông điệp.

- Phương thức listenMessage() phân tích thông điệp nhận được và thực hiện các hành động tương ứng. Nếu loại thông điệp là Constant.trackingFile, phương thức setPath() của lớp WatchFile được gọi để cập nhật đường dẫn theo thông điệp nhận được.
- Phương thức connect() thiết lập kết nối tới máy chủ socket thông qua địa chỉ host và cổng được cung cấp. Nếu kết nối thành công, lớp SocketClient khởi tạo luồng dữ liệu và trả về true. Ngược lại, nó trả về false.
- Phương thức getSocket() trả về đối tượng Socket hiện tại của lớp SocketClient.

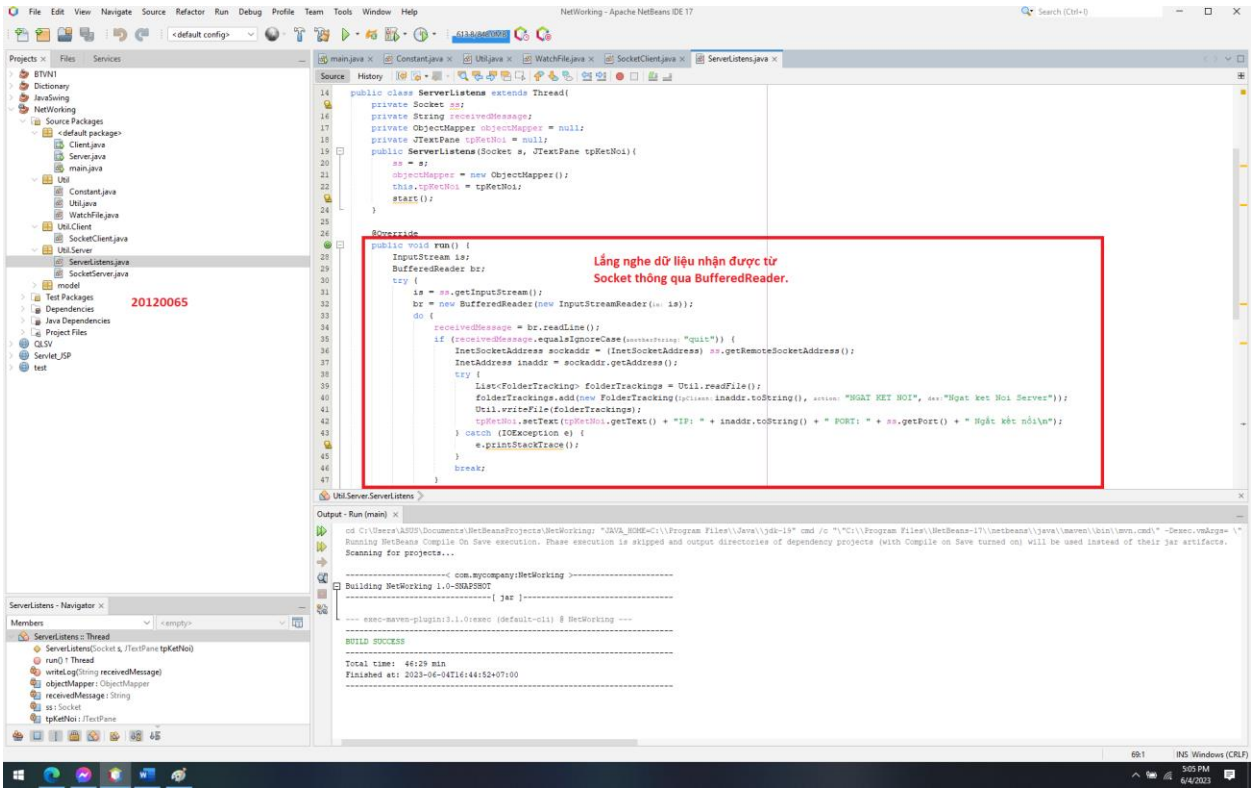


6. Package Util.Server

a) ServerListenes

File này chịu trách nhiệm lắng nghe kết nối từ phía máy khách, xử lý thông điệp nhận được và ghi log vào tệp tin.

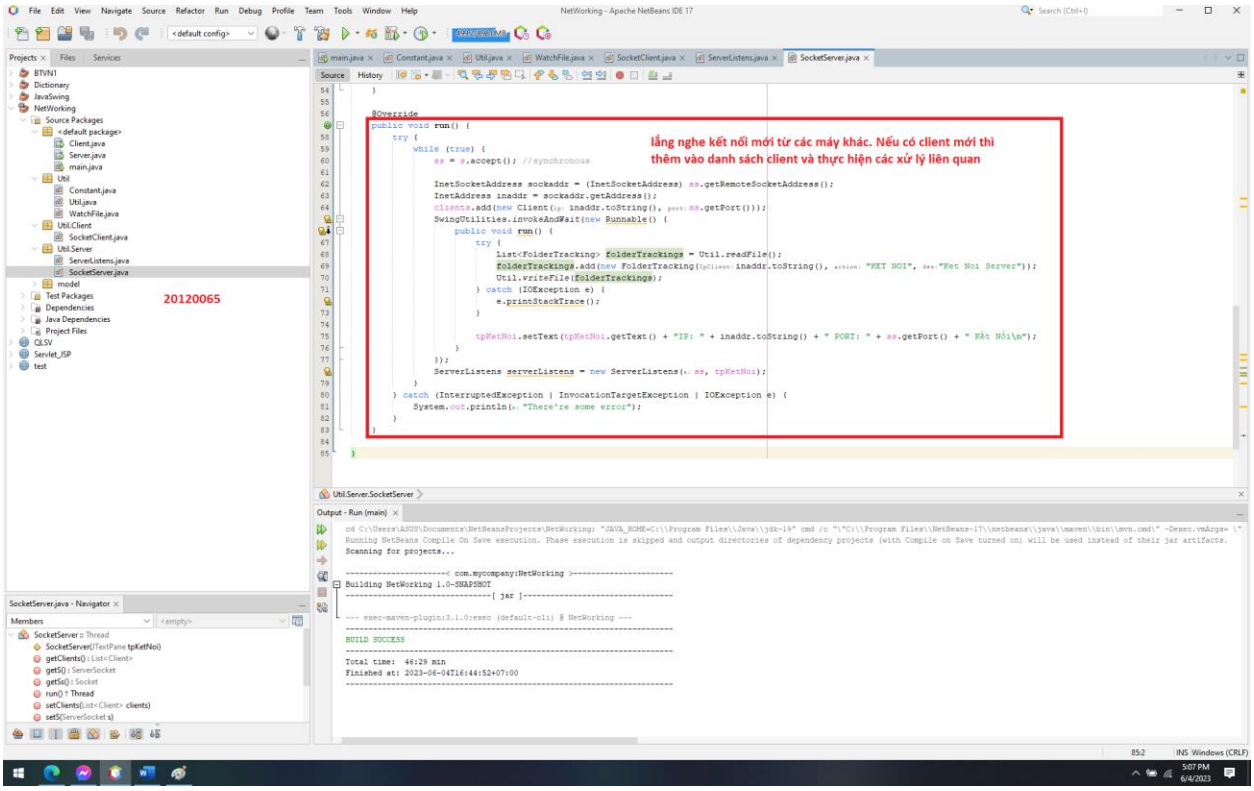
- Lớp ServerListens kế thừa từ lớp Thread, cho phép nó chạy trên một luồng riêng biệt.
- Constructor của lớp ServerListens nhận một đối tượng Socket và một JTextPane. Trong constructor, lớp ObjectMapper được khởi tạo để chuyển đổi JSON thành đối tượng Java. Luồng mới cũng được khởi chạy.
- Trong phương thức run(), lớp ServerListens lắng nghe dữ liệu nhận được từ Socket thông qua BufferedReader. Nếu dữ liệu là "quit", kết nối sẽ được đóng. Nếu không, phương thức writeLog() được gọi để xử lý thông điệp nhận được.
- Phương thức writeLog() chuyển đổi thông điệp JSON nhận được thành đối tượng FolderTracking bằng ObjectMapper. Sau đó, địa chỉ IP của máy khách được lấy từ Socket và đặt vào đối tượng FolderTracking. Cuối cùng, phương thức Util.writeFile() được gọi để ghi thông tin vào tệp log.txt.
- Trong phương thức writeLog(), phương thức Util.readFile() được sử dụng để đọc dữ liệu từ tệp log.txt và phương thức Util.writeFile() được sử dụng để ghi dữ liệu mới vào tệp log.txt.



b) SocketServer.java

File này chịu trách nhiệm tạo và quản lý máy chủ socket, lắng nghe kết nối từ các máy khách, ghi log và hiển thị thông tin kết nối.

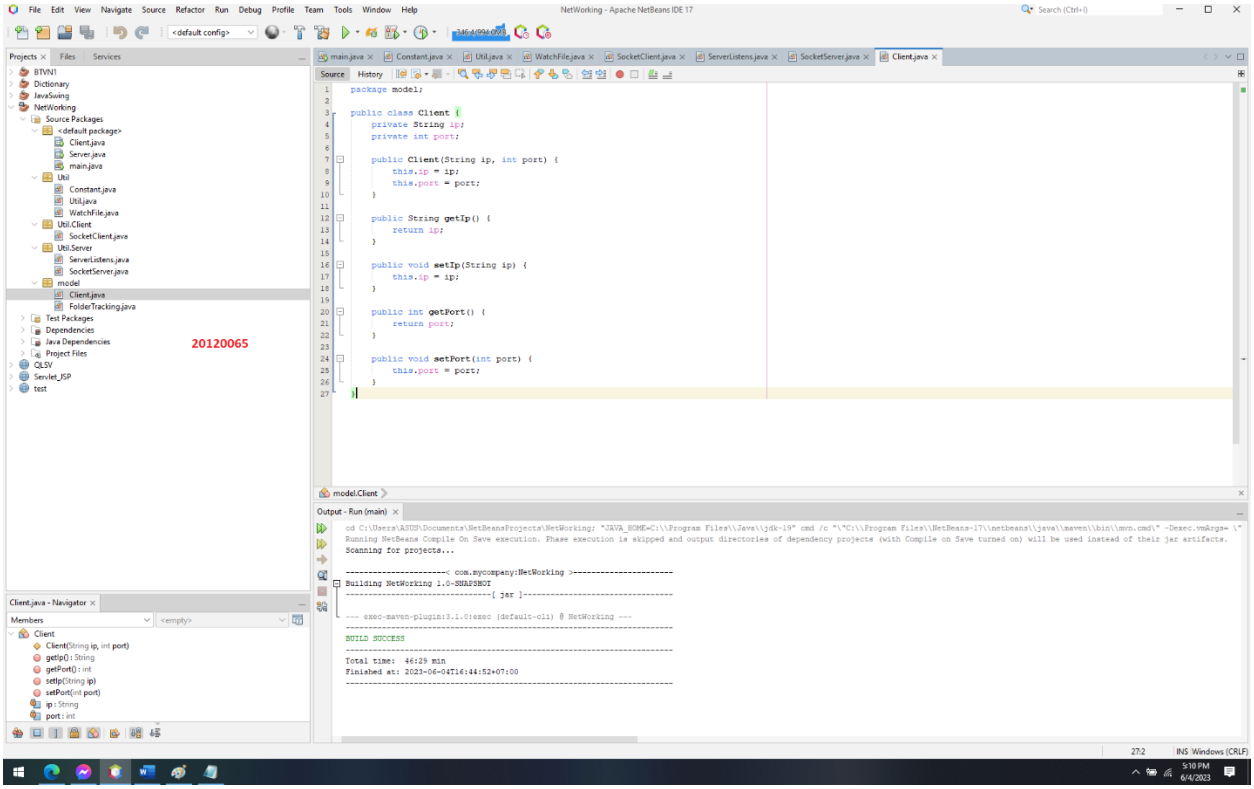
- Lớp SocketServer kế thừa từ lớp Thread, cho phép nó chạy trên một luồng riêng biệt.
- Constructor của lớp SocketServer nhận một đối tượng JTextPane để hiển thị thông tin kết nối. Trong constructor, một ServerSocket được khởi tạo và lớp được khởi chạy.
- Trong phương thức run(), lớp SocketServer lắng nghe kết nối từ các máy khách bằng cách sử dụng accept() trên ServerSocket. Mỗi khi có kết nối mới, một đối tượng Client mới được thêm vào danh sách clients và thông tin kết nối được ghi vào tệp log.txt thông qua Util.writeFile(). Đồng thời, thông tin kết nối cũng được hiển thị trên JTextPane. Cuối cùng, một đối tượng ServerListens được tạo ra để lắng nghe thông điệp từ máy khách đó.
- Trong phương thức run(), phương thức Util.readFile() được sử dụng để đọc dữ liệu từ tệp log.txt và phương thức Util.writeFile() được sử dụng để ghi dữ liệu mới vào tệp log.txt.



7. Package Model

a) Client.java

Chứa các phương thức của Client, bao gồm các getter setter liên quan tới IP và Port.



b) FolderTracking.java

File này chịu trách nhiệm lưu trữ thông tin về các hành động và sự kiện liên quan đến thư mục, cung cấp các phương thức để biểu diễn và truy xuất thông tin này.

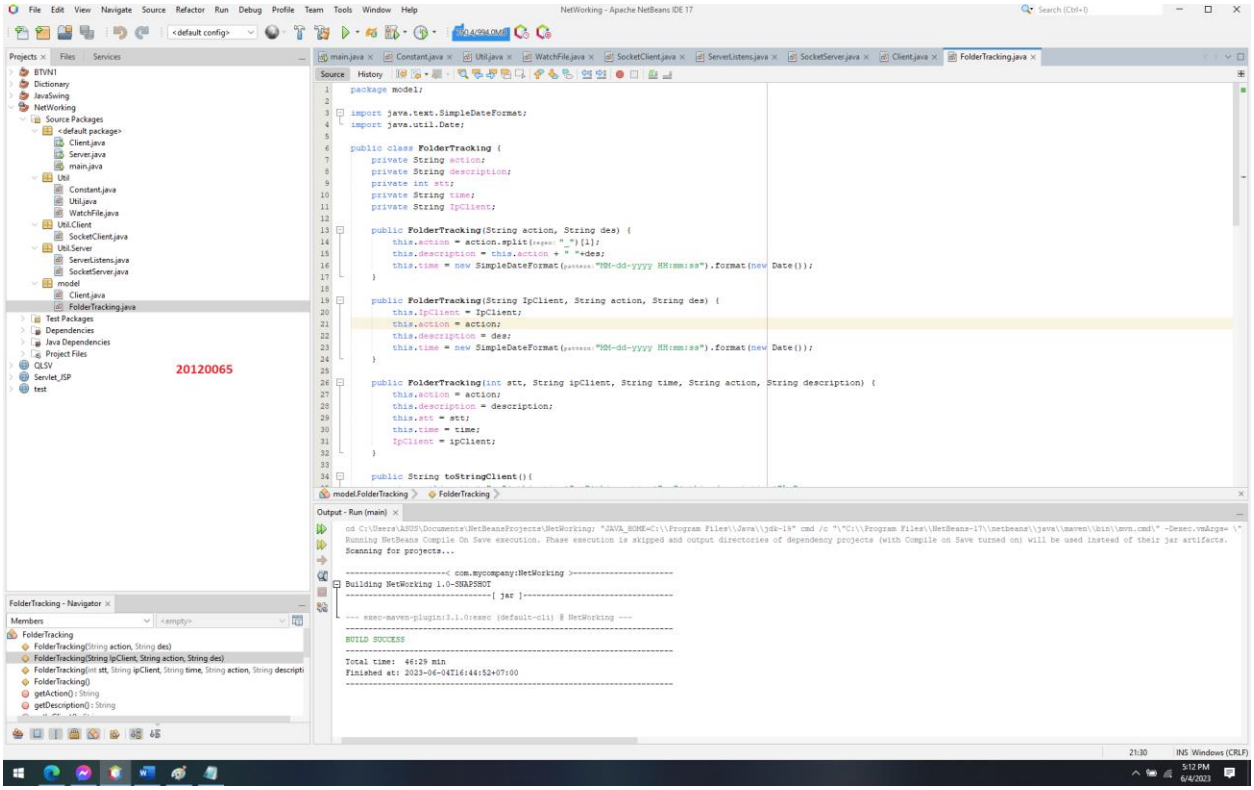
Các thuộc tính:

- action: Đại diện cho hành động được thực hiện trên thư mục.
- description: Mô tả chi tiết về hành động và thư mục liên quan.
- stt: Số thứ tự, đại diện cho vị trí của hành động trong danh sách.
- time: Thời gian thực hiện hành động, được định dạng theo định dạng ngày/giờ.
- IpClient: Địa chỉ IP của máy khách thực hiện hành động.

Các phương thức:

- toStringClient(): Trả về một chuỗi biểu diễn của đối tượng FolderTracking dành cho máy khách.
- toStringServer(): Trả về một chuỗi biểu diễn của đối tượng FolderTracking dành cho máy chủ.

Các getter và setter cho các thuộc tính của lớp.



VI. Hướng dẫn cài đặt và triển khai

Người dùng tải toàn bộ source code về máy và khởi chạy file main.java là có thể sử dụng chương trình. Ngoài ra, trong trường hợp IDE bất đồng bộ không thể cài đặt chương trình, người dùng có thể copy từng file từ source code sang IDE cá nhân vào các vị trí tương ứng phù hợp.

VII. Hướng dẫn sử dụng

Chi tiết cách sử dụng xem tại:

https://drive.google.com/file/d/1x6K5Y_xYIXrKs6bao92Yg4qer1Xeny2x/view?usp=sharing