

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN



BÁO CÁO ĐỒ ÁN LÝ THUYẾT
VIẾT CHƯƠNG TRÌNH
ĐIỀU KHIỂN MÁY TÍNH TỪ XA
MÔN MẠNG MÁY TÍNH



NHÓM THỰC HIỆN

STT	Họ và tên	MSSV
1	Bùi Hồng Đăng	21120045
2	Trương Tấn Đạt	21120050
3	Nguyễn Trang Mai Hạnh	21120065
4	Nguyễn Trọng Hiếu – Nhóm trưởng	21120068
5	Tạ Công Hoàng	21120074
6	Lê Thanh Thái Quảng	21120123

Giảng viên: Đỗ Hoàng Cường

Lớp: 21CTT2

Học kỳ 1 – Niên khóa 2022 – 2023

Lời cảm ơn

Để hoàn thành được đồ án này, nhóm chúng em xin gửi lời cảm ơn chân thành đến:

Thầy Đỗ Hoàng Cường, giảng viên môn Mạng máy tính lớp 21CTT2, thầy đã trang bị giúp chúng em những kỹ năng cơ bản và kiến thức cần thiết về Mạng máy tính để có thể hoàn thành được đồ án này.

Thầy Lê Hà Minh đã giúp chúng em hệ thống lại những kiến thức về socket và áp dụng vào việc viết chương trình trong đồ án thông qua Seminar.

Cô Huỳnh Thụy Bảo Trân đã cho lời khuyên giúp chúng em tổ chức các tài liệu cần nộp trong đồ án tốt hơn.

Lời cuối cùng, chúng em xin kính chúc thầy, cô có thật nhiều sức khỏe và gặt hái được nhiều thành quả trên con đường giảng dạy của mình!

Chúng em xin chân thành cảm ơn!

MỤC LỤC

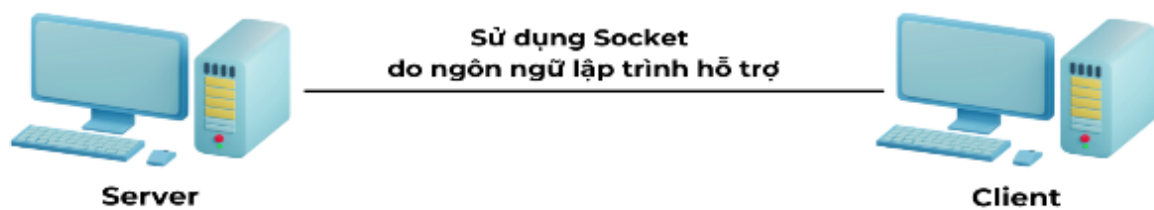
A. THÔNG TIN CHUNG VỀ ĐỒ ÁN.....	5
1. Thông tin chung	5
2. Các chức năng điều khiển cơ bản của chương trình	5
3. Video demo chương trình	6
B. MÔ TẢ CHƯƠNG TRÌNH	7
I. Tổng quan.....	7
II. Cấu trúc chương trình	7
III. Cách thực hiện các chức năng.....	19
1. Kết nối Client – Server	19
2. Liệt kê các application đang chạy trong máy, start/stop 1 app.....	20
a. Chức năng Watch: Xem danh sách các ứng dụng đang chạy	21
b. Chức năng Start: Bật một ứng dụng	21
c. Chức năng Kill: Tắt một ứng dụng.....	22
d. Chức năng Clear: Xóa danh sách ứng dụng đang chạy.....	24
e. Chức năng Save: (Chức năng bổ sung thêm)	24
3. Liệt kê các processes đang chạy trong máy, start/stop 1 process	25
a. Chức năng Watch: Xem tất cả chương trình đang chạy	26
b. Chức năng Kill: Tắt một ứng dụng đang chạy.....	27
c. Chức năng Start: Bật một process.	29
d. Chức năng Clear: Xóa bảng đã tạo từ chức năng Watch.	31
e. Chức năng Save: (Chức năng bổ sung thêm).....	31
4. Bật phím nhân	32
a. Chức năng Hook & Unhook.....	33
b. Chức năng In phím.....	34
c. Chức năng Xóa.....	34
d. Chức năng Save (Chức năng bổ sung thêm).....	34
5. Chụp màn hình	35
a. Chức năng Chụp.....	36
b. Chức năng Lưu.....	36
6. Lấy thông tin OS và File system (Get info) (Chức năng bổ sung thêm)	36

a. Chức năng Get: Lấy thông tin OS và File system.....	38
b. Chức năng Save: Lưu thông tin lấy được.....	38
c. Chức năng Clear: Xóa thông tin đã lấy được.....	39
7. Truyền màn hình: Display (Chức năng bổ sung thêm).....	39
a. Chức năng xem trực tiếp màn hình của server	40
b. Chức năng Chụp: Chụp màn hình, tạm dừng truyền và lưu ra file .jpg.....	40
c. Chức năng Exit: Thoát cửa sổ truyền màn hình	40
8. Shutdown	41
9. Exit: Thoát cửa sổ client, ngắt kết nối	41
C. HƯỚNG DẪN SỬ DỤNG CHƯƠNG TRÌNH.....	42
I. Kết nối Server – Client.....	42
II. Liệt kê các application đang chạy trong máy, start/stop 1 app	44
III. Liệt kê các processes đang chạy trong máy, start/stop 1 process	49
IV. Bắt phím nhấn	53
V. Chụp màn hình	55
VI. Lấy thông tin OS và File system (Get info).....	56
VII. Truyền màn hình (Display).....	58
VIII. Shutdown	61
IX. Exit	61
D. TÀI LIỆU THAM KHẢO	62
E. HOẠT ĐỘNG NHÓM ĐỒ ÁN	63
I. Giới thiệu thành viên nhóm.....	63
II. Nguyên tắc hoạt động nhóm	63
III. Kế hoạch làm việc nhóm.....	63
IV. Đánh giá hoạt động nhóm	64
1. Đánh giá mức độ hoàn thành chung của đồ án.....	64
2. Phân công công việc và đánh giá tỉ lệ đóng góp của từng thành viên.....	65
F. BIÊN BẢN HỌP NHÓM.....	66

A. THÔNG TIN CHUNG VỀ ĐỒ ÁN

1. Thông tin chung

- Viết chương trình điều khiển máy tính từ xa bằng ngôn ngữ Java với mô hình Client-Server. Người dùng sẽ thực hiện các thao tác trên Client để điều khiển Server. Server cho phép Client truy cập tới và thực hiện các chức năng điều khiển.



2. Các chức năng điều khiển cơ bản của chương trình

(Các chức năng bổ sung thêm so với yêu cầu của đề được in đỏ)

- Liệt kê các processes đang chạy trong máy tính, start/stop 1 process.

Gồm các chức năng:

- + WATCH để xem danh sách các process đang chạy.
- + START để bật một process.
- + KILL để diệt process.
- + CLEAR để xóa danh sách.

+ SAVE để lưu danh sách ra file .xlsx. (Chức năng bổ sung thêm)

- Liệt kê các application đang chạy trong máy tính, start/stop 1 app.

Gồm các chức năng:

- + WATCH để xem danh sách các ứng dụng đang chạy.
- + START để bật một ứng dụng.
- + KILL để tắt một ứng dụng.
- + CLEAR để xóa danh sách.

+ SAVE để lưu danh sách ra file .xlsx. (Chức năng bổ sung thêm)

- Chụp màn hình (Print Screen)

- + Chụp để chụp màn hình hiện tại của server
- + Lưu để lưu màn hình vừa chụp thành file .jpg
- Bắt phím nhấn (KeyStroke).
Gồm các chức năng:
 - + Hook: Bắt đầu bắt phím.
 - + Unhook: Huỷ bắt phím.
 - + In phím: In các phím vừa bắt được.
 - + Xóa: Xóa bảng các phím vừa in sau khi bắt được.
 - + Save: Lưu văn bản bắt được ra file .docx (Chức năng bổ sung thêm)
- Lấy thông tin OS và File system (Get info) (Chức năng bổ sung thêm)
 - + Get: Lấy thông tin và in ra 2 table
 - + Save: Lưu ra file .xlsx
 - + Clear: Xóa table
- Truyền màn hình: Display (Chức năng bổ sung thêm)
 - + Xem trực tiếp màn hình của server
 - + Chụp: Chụp lại màn hình, tạm dừng truyền màn hình và lưu ra file .jpg
 - + Exit: Thoát cửa sổ truyền màn hình
- Shutdown: Shutdown server sau 10s
- Exit: Thoát cửa sổ client, ngắt kết nối

3. Video demo chương trình

❖ Video demo – Người thực hiện: Nguyễn Trọng Hiếu - 21120068

- Link_drive:

https://drive.google.com/file/d/13_1g3gapWZjqFfaf6oCRBjr90AKfBMaW/view?usp=share_link

- (Hoặc) link youtube: https://youtu.be/9Scgk_6MpVo

❖ Các file đã lưu đề cập trong video (Folder Output)

- Link files output:

https://drive.google.com/drive/folders/1_8uDXGJ_7jLeNLNFfts6UX4-DKj0KUxu?usp=share_link

B. MÔ TẢ CHƯƠNG TRÌNH

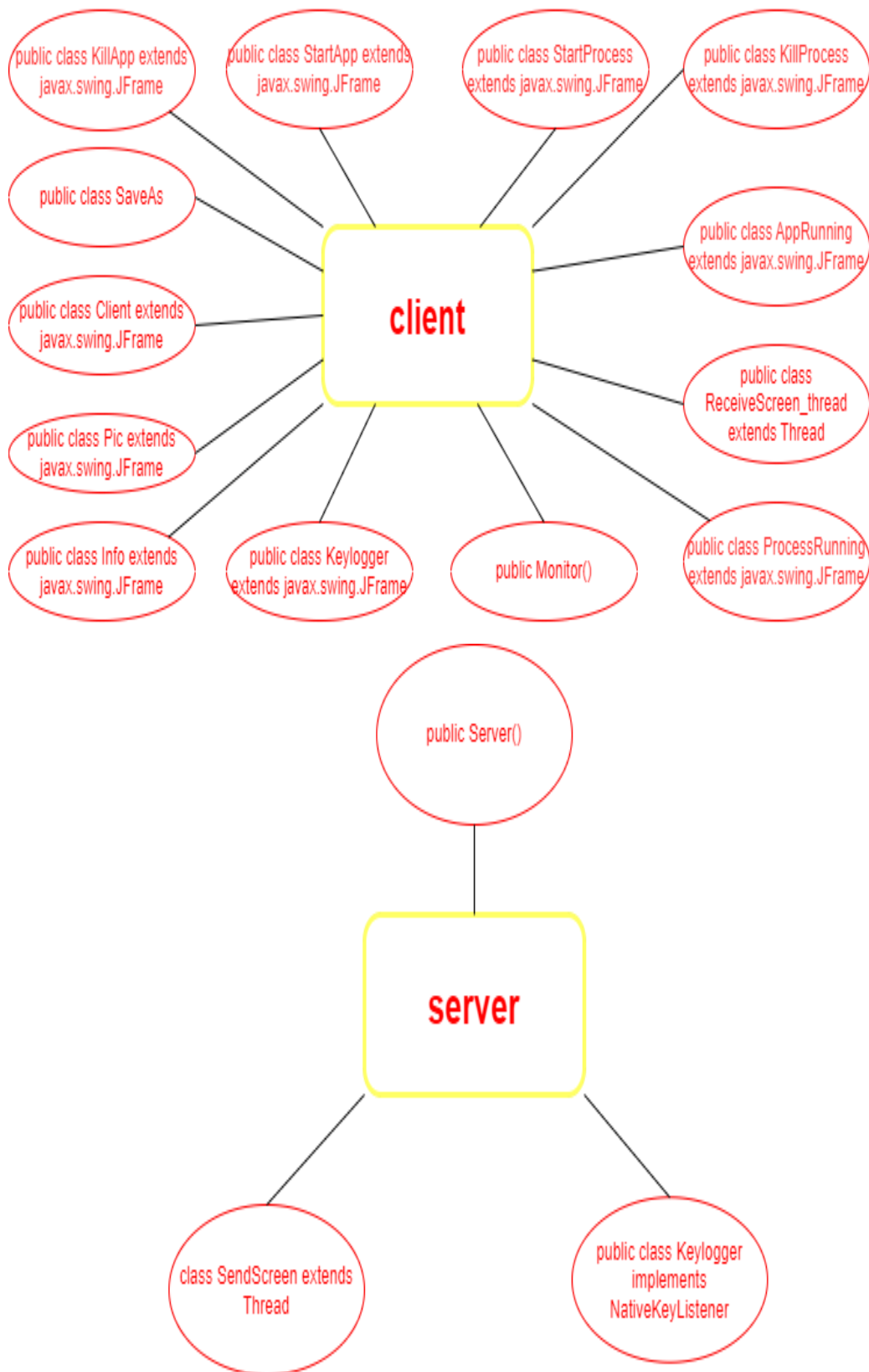
I. Tổng quan

- **Kiến trúc mạng sử dụng:** Client – Server
- **Giao thức sử dụng tại tầng transport:** TCP (Transmission Control Protocol)
Với giao thức TCP/IP, chúng là một giao thức không có trạng thái, trong mỗi một yêu cầu sau nào đó thì chúng đều tách biệt với các yêu cầu trước đó. Các Server/Client được nối mạng tạo kết nối với nhau, chúng có thể dễ dàng trao đổi dữ liệu hoặc các gói tin. Nhờ vậy, thao tác truyền thông tin đến nơi nhận sẽ chính xác và hiệu quả hơn.
- **Port sử dụng cho Server:** 5656
- **Môi trường lập trình:** Apache NetBeans IDE 15
- **Framework hỗ trợ:**
 - java.net.Socket;
 - java.io
 - javax.swing;
 - java.util
 - org.apache.logging.log4j;
 - java.awt
 - java.nio
 - javax.imageio
 - org.apache.poi.ss
 - org.apache.poi.xssf
 - org.apache.poi.xwpf
 - com.github.kwhat.jnativehook

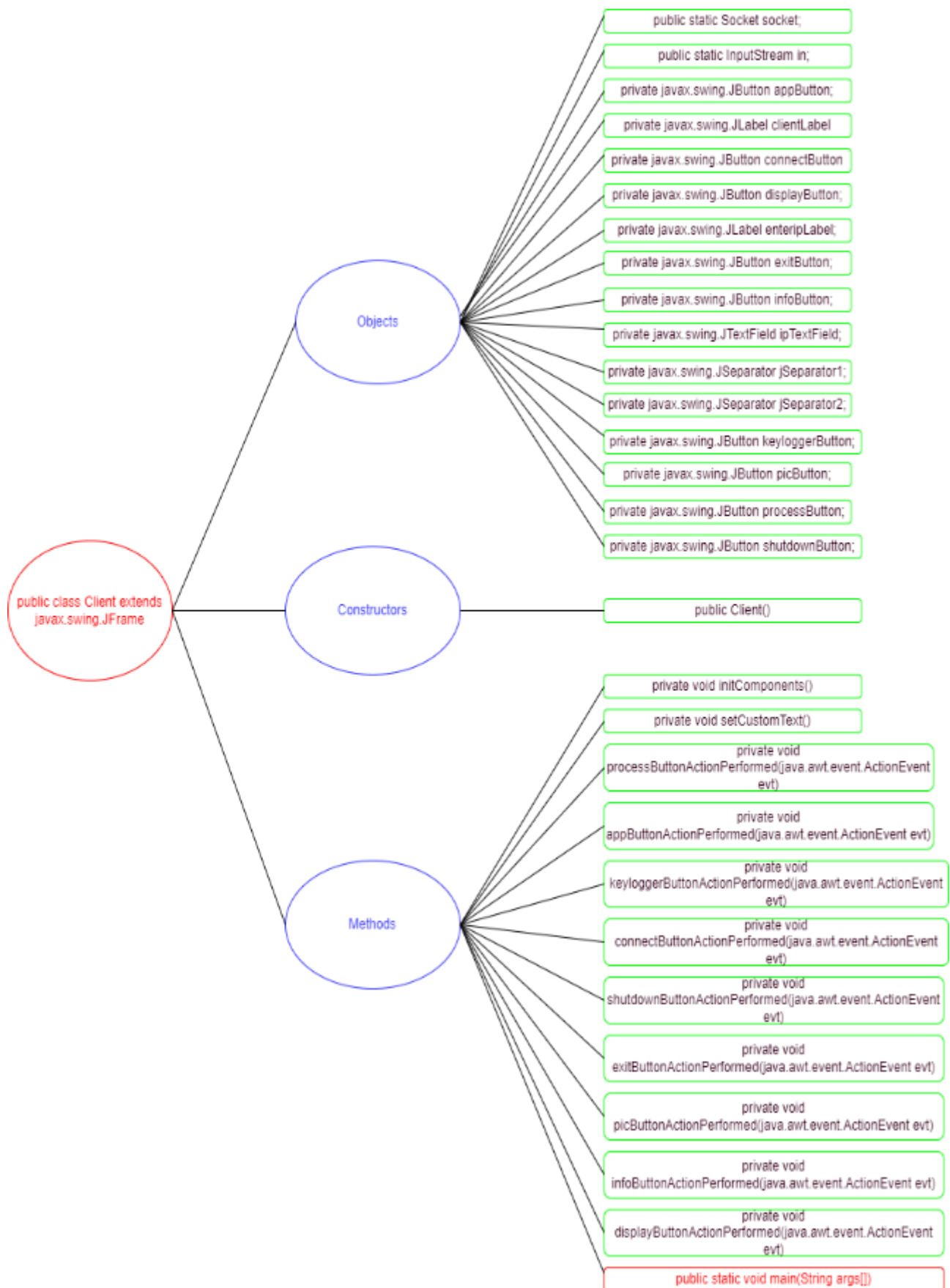
II. Cấu trúc chương trình

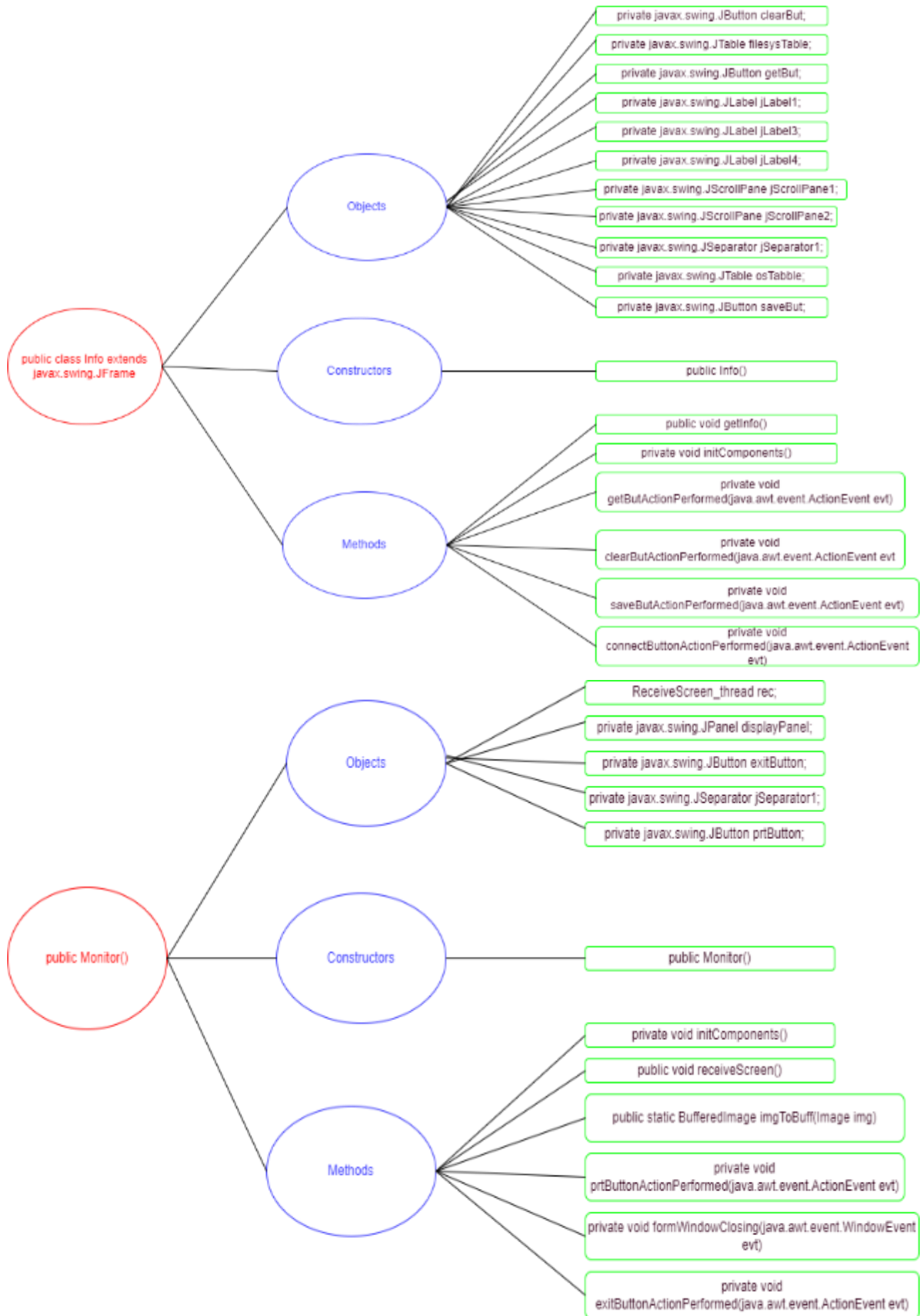
- Project Client chứa package client. Trong package client chứa 13 classes: Client, AppRunning, ProcessRunning, KillApp, KillProcess, StartApp, StartProcess, Info, Keylogger, Monitor, Pic, ReceiveScreen_thread, SaveAs

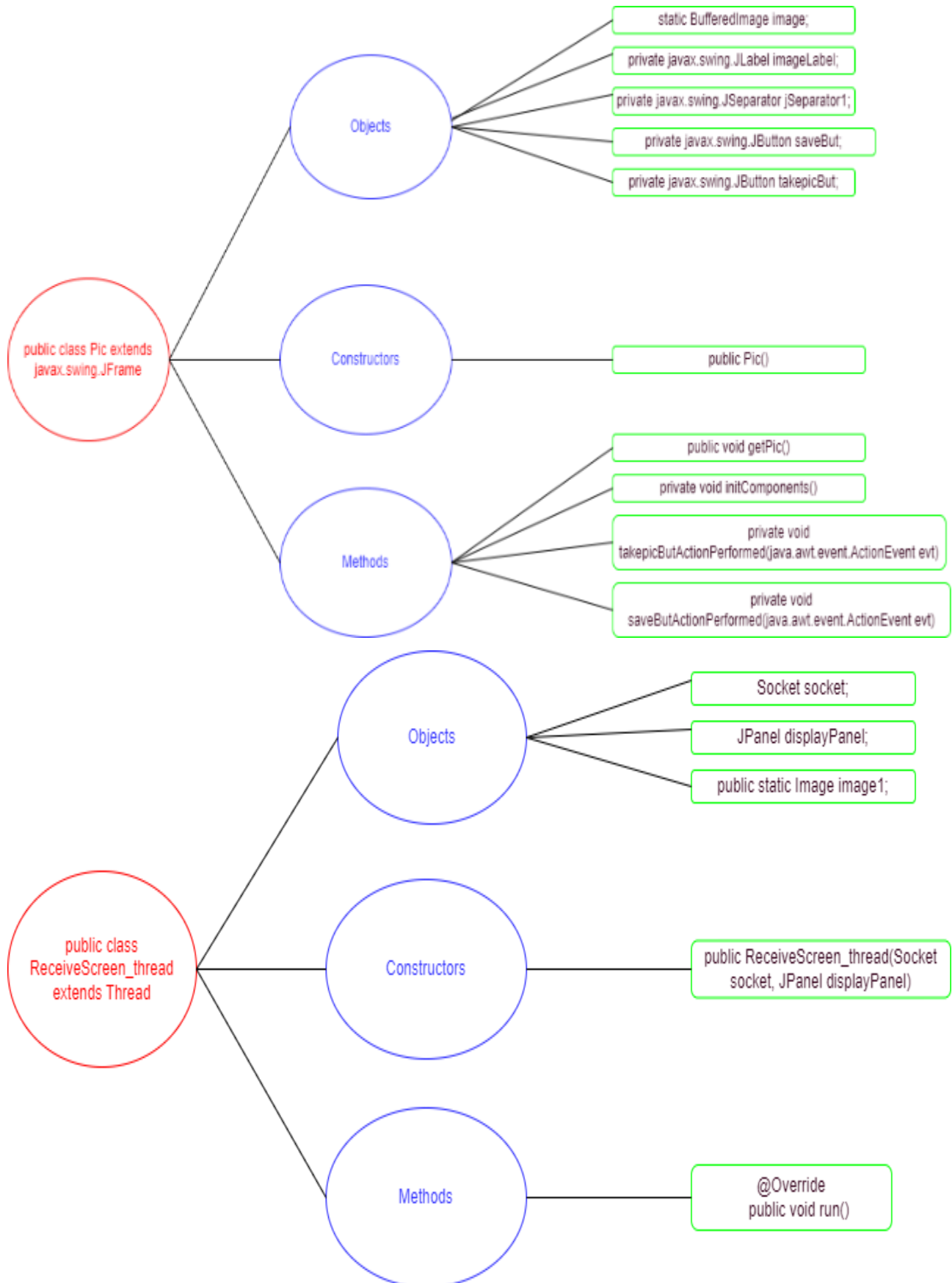
- Project Server chứa package server. Trong package server chứa 3 classes: Server, Keylogger, SendScreen



• Cấu trúc các classes của client









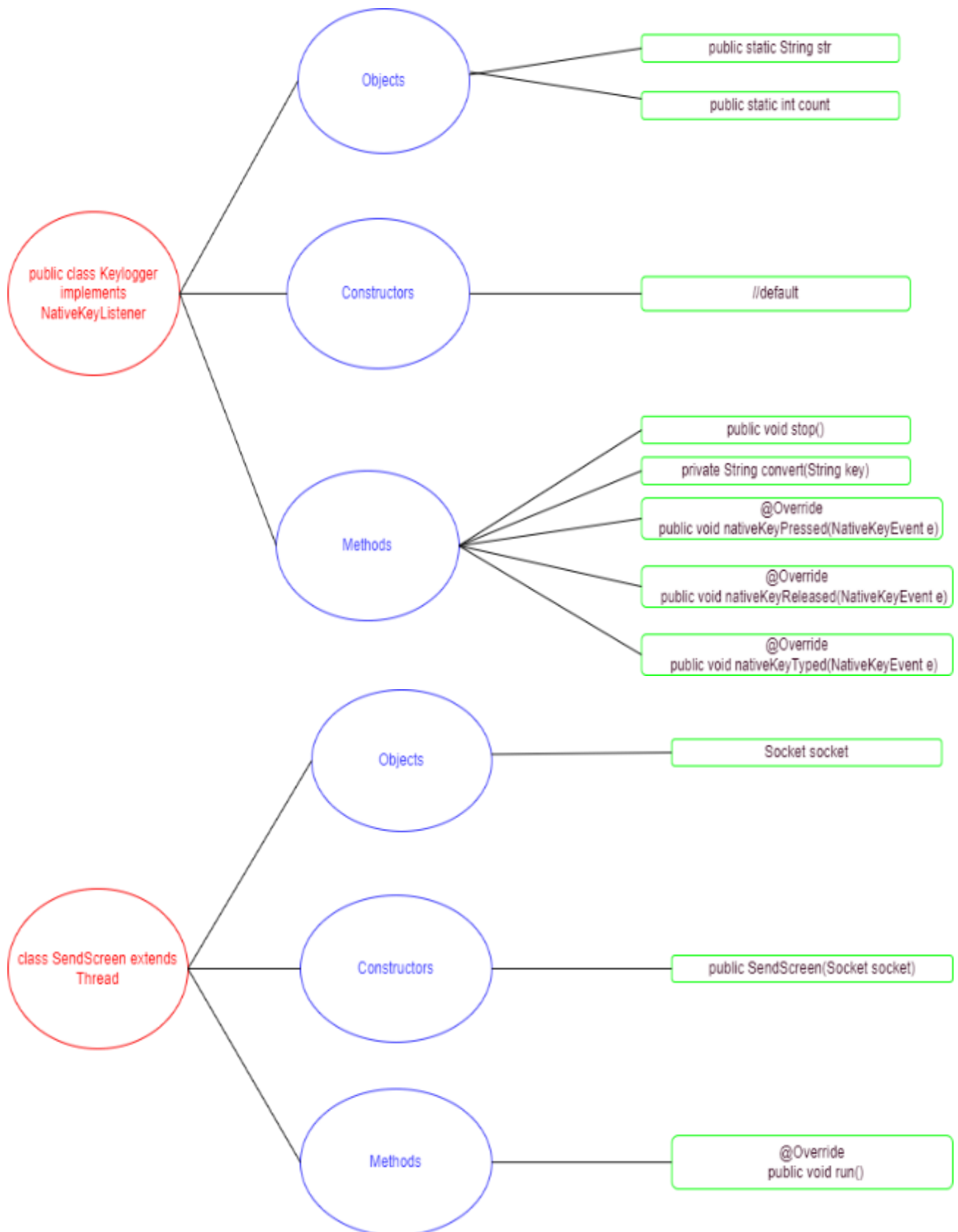


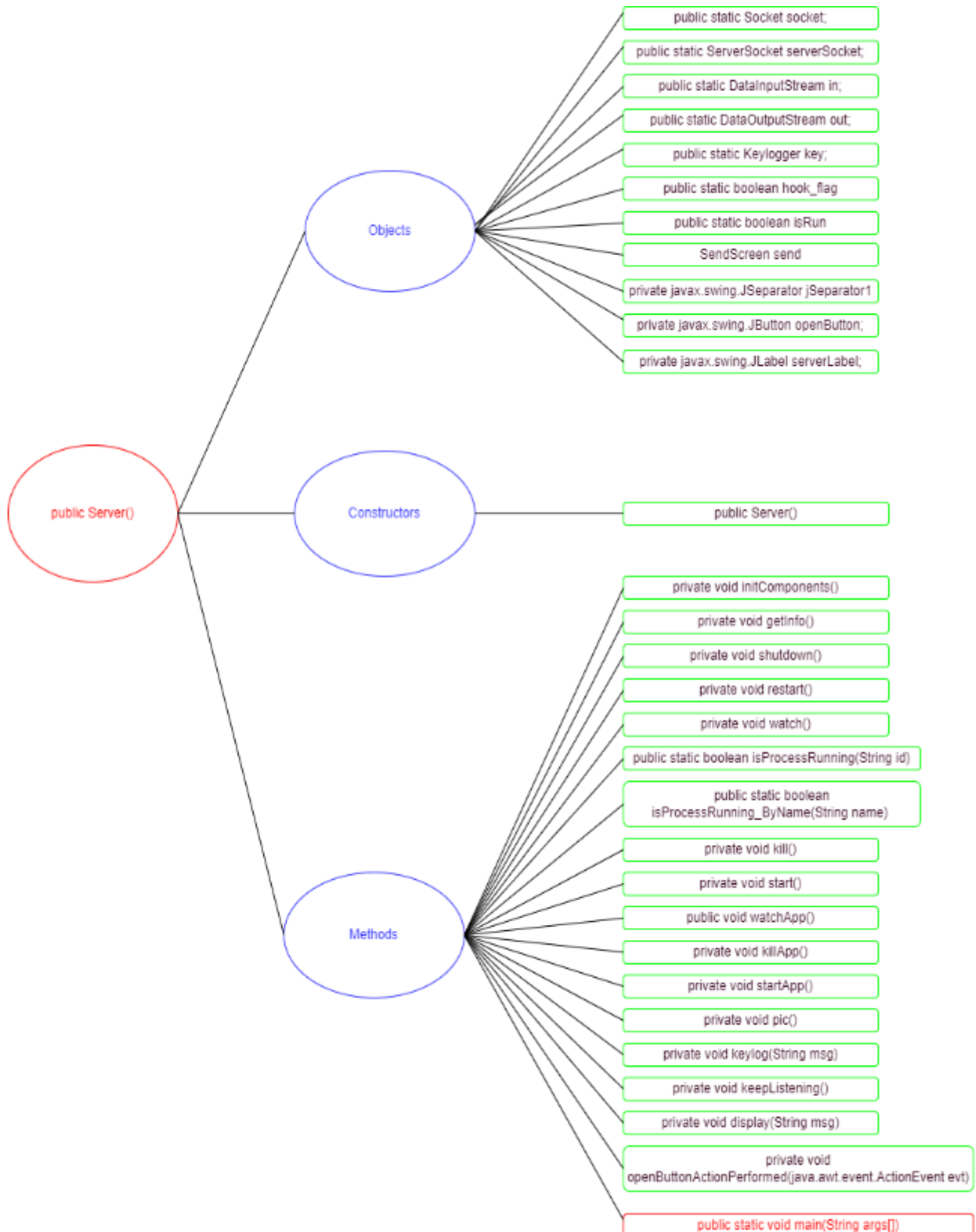






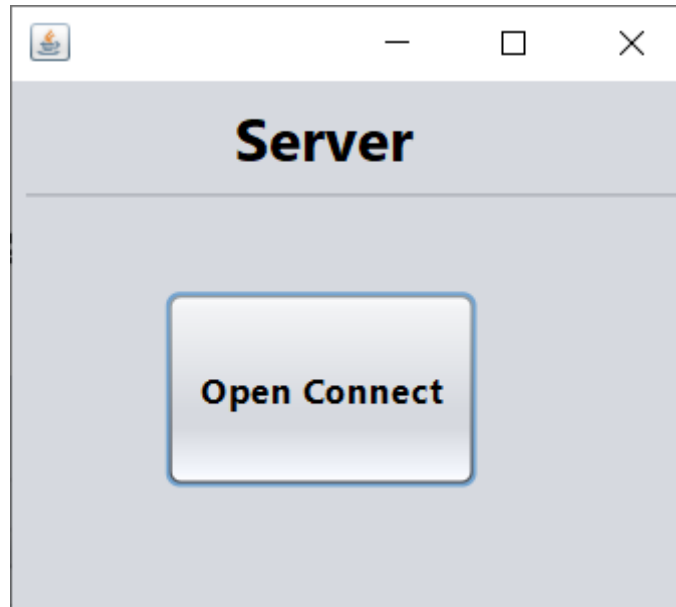
- Cấu trúc các classes của server



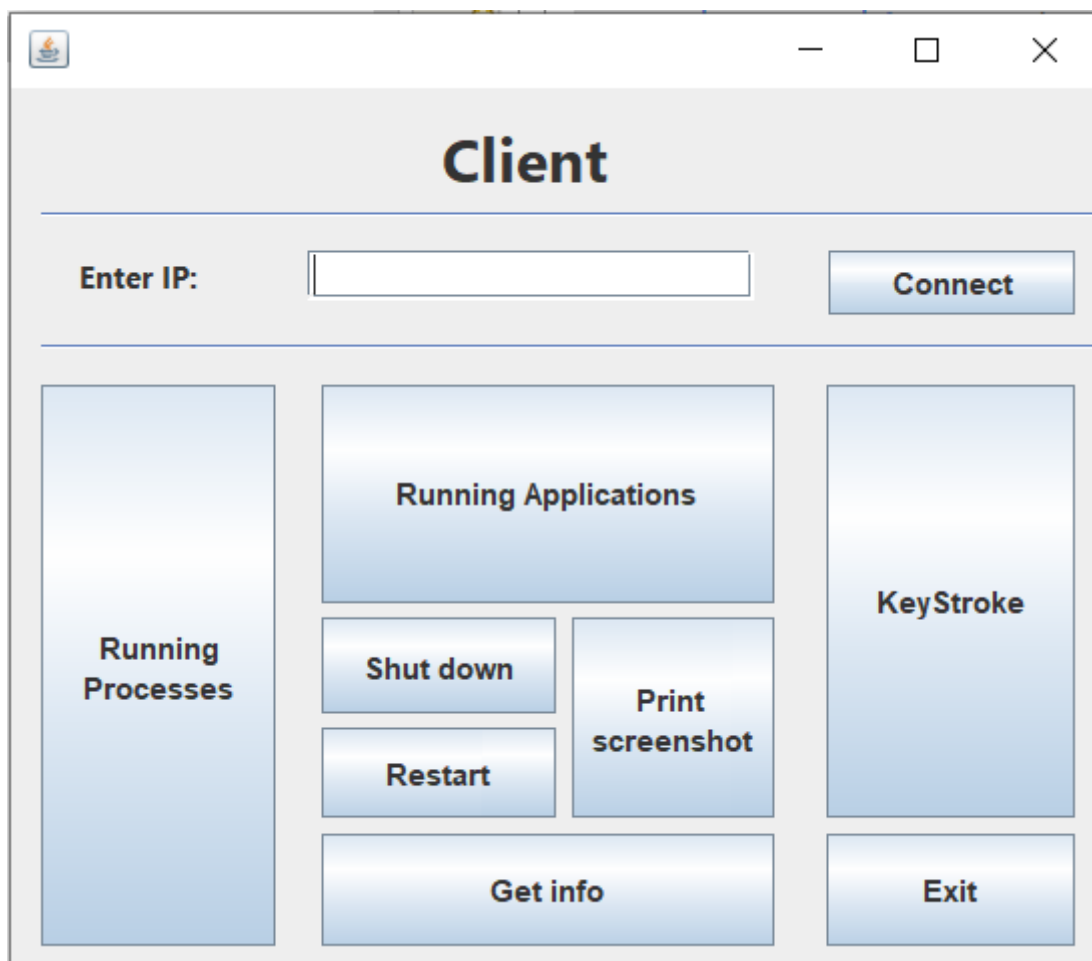


III. Cách thực hiện các chức năng

1. Kết nối Client – Server



Giao diện Server



Giao diện Client

Methods:

❖ Bên phía Client:

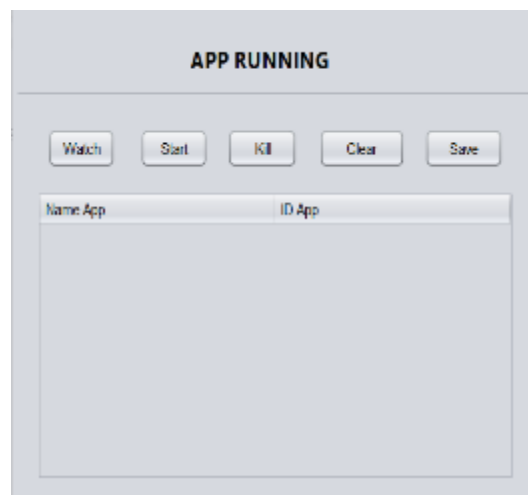
- *void connectButActionPerformed(java.awt.event.ActionEvent evt)*: Sau khi người dùng nhập địa chỉ IP, đọc và tiến hành kết nối với Server có địa chỉ IP tương ứng.

❖ Bên phía Server:

- *void openButActionPerformed(java.awt.event.ActionEvent evt)*: Mở Server và đợi kết nối từ Client.

2. Liệt kê các application đang chạy trong máy, start/stop 1 app

Tiến hành gọi hàm AppRunning để thực hiện các chức năng bên trong:



Class “AppRunning” gồm:

- Constructor:
 - *AppRunning()*: Gọi phương thức khởi tạo các thành phần giao diện bên trong class
- Methods:
 - *void initComponents()*: Khởi tạo các thành phần giao diện bên trong class.
 - *void killButtonActionPerformed(java.awt.event.ActionEvent evt)*: Chạy các sự kiện sau khi nhấn nút “Kill” ở giao diện Running Application.
 - *void watchButtonActionPerformed(java.awt.event.ActionEvent evt)*: Xuất hiện bảng chứa thông tin các chương trình đang chạy.

- *void clearButtonActionPerformed(java.awt.event.ActionEvent evt)*: Xóa bảng chứa thông tin các chương trình đang chạy (nếu có).
- *void startButtonActionPerformed(java.awt.event.ActionEvent evt)*: Chạy các sự kiện sau khi nhấn nút “Start” ở giao diện Running Application.
- *void saveButActionPerformed(java.awt.event.ActionEvent evt)*: Chạy các sự kiện sau khi nhấn nút “Save” ở giao diện Running Application.

a. Chức năng Watch: Xem danh sách các ứng dụng đang chạy

❖ Hàm bên phía Client:

- Gửi tín hiệu nút bấm “Watch App” đến server.
- Mỗi lần bấm Watch sẽ tạo một table.
- Biến line sẽ đọc thông tin về Application từ server, biến đổi và được thêm vào table. Lặp lại đến khi nhận được tín hiệu kết thúc ở đây là “@@@”.

❖ Hàm bên phía Server:

- Gọi hàm watchApp() khi nhận được tín hiệu “Watch App” từ client.
- Biến line để lưu dữ liệu về Application mà được lưu trong biến sc đọc từ powershell.
- Đọc, lưu và gửi dữ liệu khi vẫn còn có thể đọc được (sc.hasNextline()) qua client.
- Sau khi gửi hết dữ liệu, gửi thêm tín hiệu thông báo kết thúc, ở đây là “@@@”

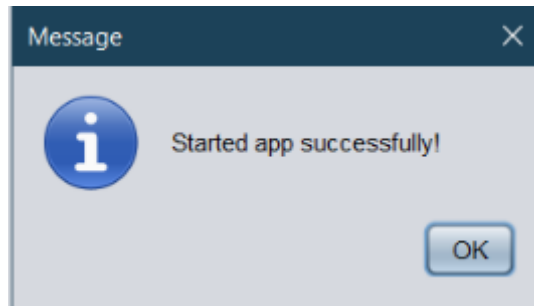
b. Chức năng Start: Bật một ứng dụng

Class StartApp gồm 1 constructor và 2 methods:

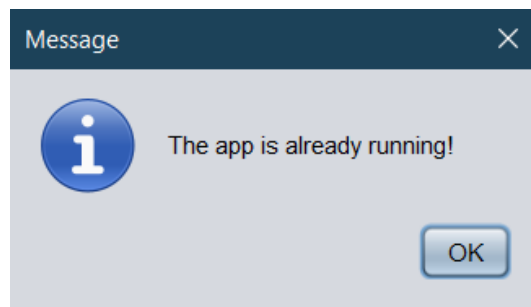
- Constructor:
 - *StartApp()*: Gọi phương thức khởi tạo các thành phần của giao diện Start App.
- Methods:
 - *void initComponents()*: Khởi tạo các thành phần của giao diện Start App.
 - *void startAppActionPerformed(java.awt.event.ActionEvent evt)*: Nhận tên của chương trình đã nhập và tiến hành bật chương trình đó.

❖ Hàm bên phía client:

- Gửi tín hiệu “Start App” đến server.
- Biến id lưu tên ứng dụng đã được nhập.
- Sau khi server chạy chức năng sẽ gửi đến client thông báo trong fromSV:
 - “Yes”: Đã bật ứng dụng thành công và thông báo.



- “Running”: Ứng dụng đang chạy và thông báo.



- “No”: Không thể bật ứng dụng và thông báo.

❖ Hàm bên phía server:

- Gọi hàm startApp() khi nhận tín hiệu “Start App”.
- Biến name lưu tên ứng dụng đã nhập bên client.
- Kiểm tra ứng dụng có đang chạy qua hàm isProcessRunning_ByName().
 - Nếu ứng dụng đang không chạy, bật ứng dụng đó và gửi cho client “Yes”.
 - Nếu ứng dụng đang chạy, gửi “Running” cho client.
 - Nếu không thể chạy ứng dụng, gửi “No” cho client.

c. Chức năng Kill: Tắt một ứng dụng

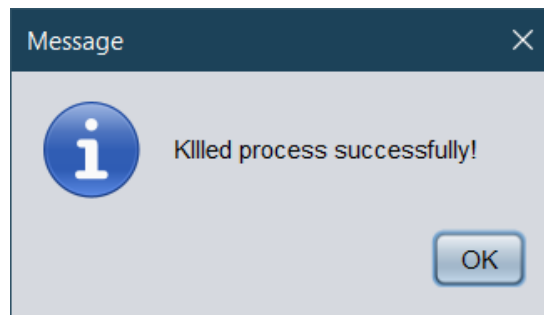
Class KillApp gồm 1 constructor và 2 methods:

- Constructor:
 - *KillApp()*: Gọi phương thức khởi tạo thành phần của giao diện Kill App.
- Methods:
 - *void initComponents()*: Khởi tạo thành phần của giao diện Kill App.

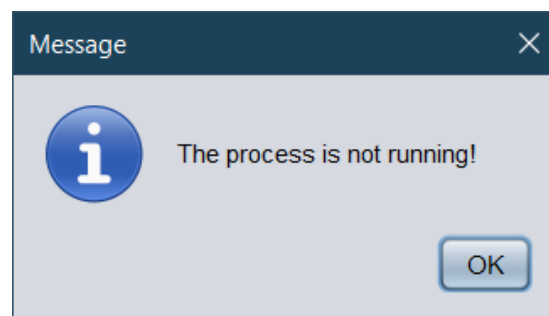
- `void killAppActionPerformed(java.awt.event.ActionEvent evt)`: Nhận ID của chương trình đã nhập và tiến hành tắt chương trình đó.

❖ Hàm bên phía client:

- Gửi tín hiệu “Kill App” đến server.
- Biến id lưu tên ứng dụng đã được nhập.
- Sau khi server chạy chức năng sẽ gửi đến client thông báo trong fromSV:
 - “Yes”: Đã tắt ứng dụng thành công và thông báo.



- “Not found”: Không tìm thấy ứng dụng theo ID và thông báo.



- “No”: Không thể tắt ứng dụng và thông báo.
- Thông báo lỗi nếu không thể tắt ứng dụng.

❖ Hàm bên phía server:

- Gọi hàm `killApp()` khi nhận tín hiệu “Kill App”
- Biến id lưu tên ứng dụng đã nhập bên client
- Kiểm tra ứng dụng có đang chạy qua hàm `isProcessRunning ()`
 - Nếu ứng dụng đang chạy, tắt ứng dụng đó và gửi cho client “Yes”
 - Nếu không tìm thấy ứng dụng theo ID, gửi “Not found” cho client
 - Nếu không thể tắt ứng dụng, gửi “No” cho client

d. Chức năng Clear: Xóa danh sách ứng dụng đang chạy**❖ Hàm bên phía client:**

Phương thức `void clearButtonActionPerformed(java.awt.event.ActionEvent evt)`:

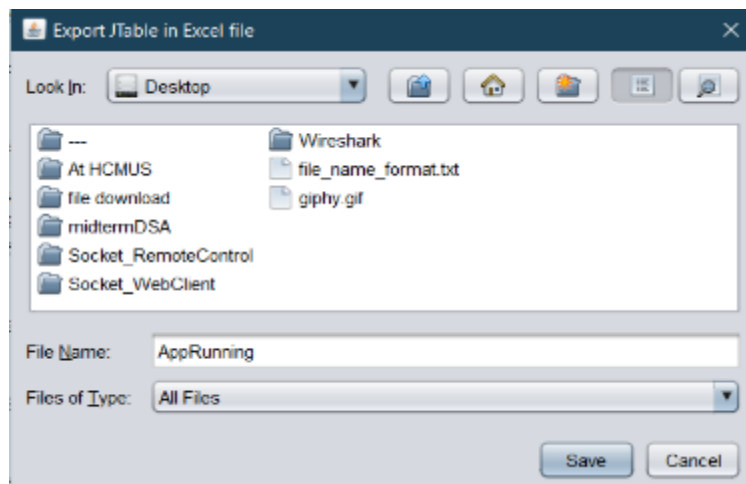
Xóa bảng chứa thông tin các chương trình.

Hàm này gọi lại bảng đã tạo ở chức năng “Watch”, sau đó đặt lại số hàng của bảng về bằng 0 và kết thúc hàm.

e. Chức năng Save: (Chức năng bổ sung thêm)**❖ Hàm bên phía client:**

Phương thức `void saveButActionPerformed(java.awt.event.ActionEvent evt)`: Lưu bảng chứa thông tin của các ứng dụng vào file Excel.

Sau khi nhấn nút “Save” giao diện sau sẽ hiện lên để người dùng có thể chọn nơi lưu file và đặt tên cho nó



3. Liệt kê các processes đang chạy trong máy, start/stop 1 process

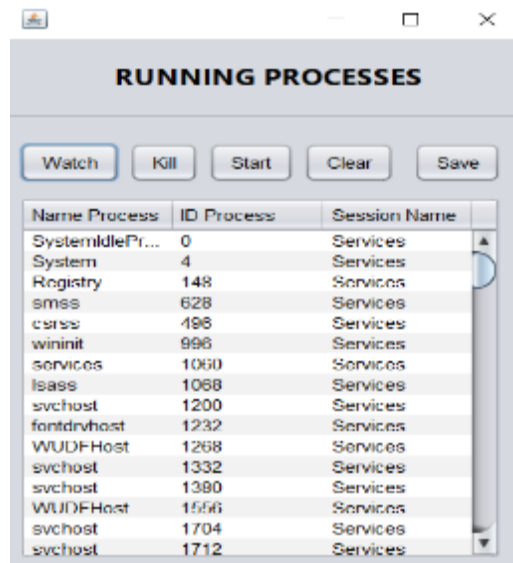
Tiến hành gọi hàm `ProcessRunning` để thực hiện các chức năng bên trong:



Class “`ProcessRunning`” gồm 1 constructor và 6 methods:

- Constructor:
 - `ProcessRunning()`: Gọi phương thức khởi tạo các thành phần giao diện bên trong class.
- Methods:
 - `void initComponents()`: Khởi tạo các thành phần giao diện bên trong class.
 - `void killButtonActionPerformed(java.awt.event.ActionEvent evt)`: Chạy các sự kiện sau khi bấm nút “Kill” ở giao diện Running Processes.
 - `void watchButtonActionPerformed(java.awt.event.ActionEvent evt)`: Xuất hiện bảng chứa thông tin các chương trình đang chạy.
 - `void clearButtonActionPerformed(java.awt.event.ActionEvent evt)`: Xóa bảng chứa thông tin các chương trình đang chạy (nếu có).
 - `void startButtonActionPerformed(java.awt.event.ActionEvent evt)`: Chạy các sự kiện sau khi bấm nút “Start” ở giao diện Running Processes.
 - `void saveButActionPerformed(java.awt.event.ActionEvent evt)`: Chạy các sự kiện sau khi bấm nút “Save” ở giao diện Running Processes.

a. Chức năng Watch: Xem tất cả chương trình đang chạy

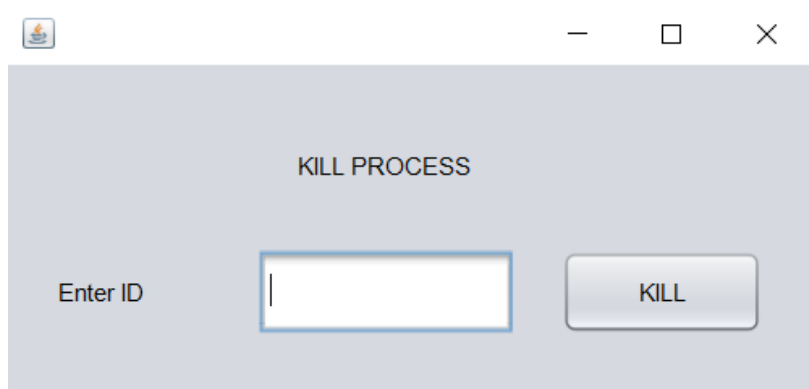


❖ Hàm bên phía client:

- Gửi tín hiệu của nút bấm đến cho server (ở đây là “Watch”).
- Tạo một bảng và làm cho bảng tự động làm mới mỗi lần bấm nút “Watch”.
- Dùng 1 biến (ở đây là biến *line*) để đọc thông tin về các chương trình mà server gửi qua, sau đó qua vài bước biến đổi cho phù hợp, ta sẽ thêm thông tin của chương trình vừa đọc đó vào bảng ở trên. Lặp lại bước này cho đến khi nhận được tín hiệu kết thúc bên phía server (ở đây là “@@@”).
- Kết thúc hàm

❖ Hàm bên phía server:

- Phương thức *watch()*: Đọc dữ liệu từ dòng lệnh chạy trên command line và gửi tất cả dữ liệu đó qua cho client.
- Nếu server nhận được tín hiệu là “Watch”, tiến hành gọi *watch()* để thực hiện chức năng xem các chương trình đang chạy.
- Dùng 1 biến (ở đây là biến *line*) để chứa dữ liệu về chương trình mà server đọc được từ command line, nếu biến dữ liệu đọc được vẫn khác **null** (vẫn đang đọc chương trình) thì gửi dữ liệu đấy cho client.
- Khi đã đọc và chuyển dữ liệu của tất cả chương trình cho client, gửi thêm một tín hiệu để thông báo cho client đã đọc hết (ở đây là “@@@”).
- Kết thúc hàm.

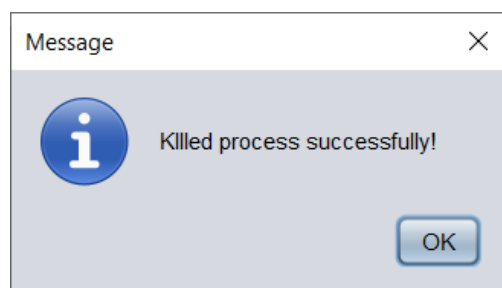
b. Chức năng Kill: Tắt một ứng dụng đang chạy

Class “KillProcess” gồm 1 constructor và 2 methods:

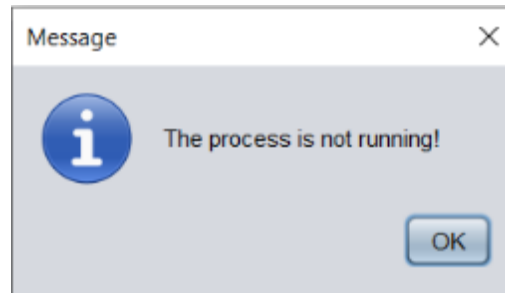
- Constructor:
 - *KillProcess()*: Gọi phương thức khởi tạo các thành phần của giao diện Kill Process.
- Methods:
 - *void initComponents()*: Khởi tạo các thành phần của giao diện Kill Process.
 - *void killActionPerformed(java.awt.event.ActionEvent evt)*: Nhận ID của chương trình đã nhập và tiến hành diệt chương trình đó.

❖ Hàm bên phía client:

- Gửi tín hiệu của nút bấm đến cho server (ở đây là “Kill”).
- Dùng 1 biến (ở đây là biến *id*) để lưu giá trị ID mà người dùng nhập vào ô và cũng được gửi qua cho server.
- Sau khi bên server thực hiện chức năng xong sẽ gửi đến client một thông báo được lưu trong biến *fromSV*:
 - Nếu *fromSV* là “Yes”: Đã diệt chương trình thành công và thông báo cho người dùng biết.



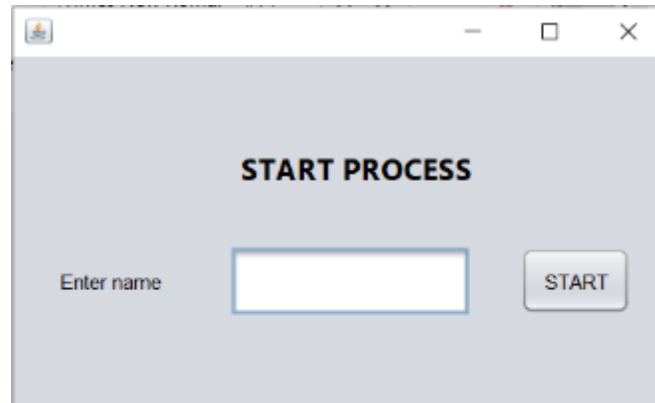
- Nếu *from SV* là “Not found”: Không tìm thấy chương trình nào có ID như đã nhập đang chạy và thông báo cho người dùng biết.



- Nếu *from SV* là “No”: Không diệt chương trình có id như đã nhập và thông báo cho người dùng biết lỗi đã gặp.
- Kết thúc hàm.
 - ❖ Hàm bên phía server:
- Phương thức:
 - *boolean isProcessRunning(String id)*: Nhận vào ID của chương trình, kiểm tra xem chương trình có ID đó có đang chạy hay không
 - *void kill()*: Diệt chương trình có ID đã nhập.
- Nếu server nhận được tín hiệu là “Kill” sẽ tiến hành gọi kill() để thực hiện chức năng.
- Dùng 1 biến (ở đây là biến *id*) lưu giá trị ID mà người dùng đã nhập bên phía client, còn biến *cmd* được gán bằng dòng lệnh để chương chạy trên command line.
- Tiến hành kiểm tra xem có chương trình có *id* đã nhập đang chạy hay không:
 - Nếu có chương trình đó đang chạy, diệt chương trình và gửi phản hồi cho bên client là đã diệt thành công (“Yes”).
 - Nếu không tìm thấy chương trình, gửi phản hồi cho bên client là không tìm thấy chương trình đang chạy (“Not found”).
 - Nếu không thể diệt chương trình, gửi phản hồi cho bên client là không thể diệt thấy chương trình đang chạy (“No”).
- Kết thúc hàm.

c. Chức năng Start: Bật một process.

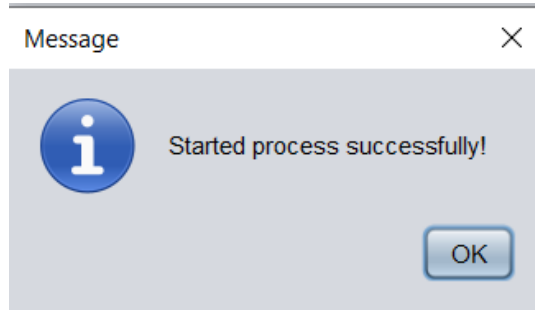
Ấn nút “Start” sẽ xuất hiện 1 giao diện để thao tác tiếp:



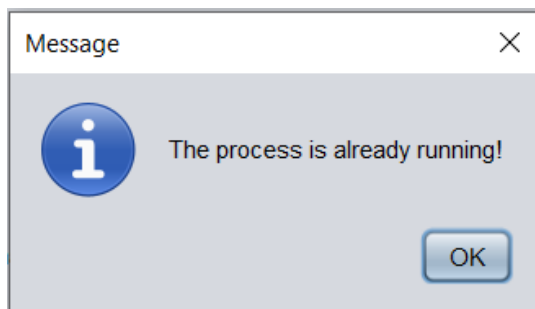
Tiến hành nhập tên của process muốn chạy và nhấn nút “START” để bắt đầu chạy chương trình đó. Class “StartProcess” gồm 1 constructor và 2 methods:

- Constructor:
 - *StartProcess()*: Gọi phương thức khởi tạo các thành phần của giao diện Start Process.
 - Methods:
 - *void initComponents()*: Khởi tạo các thành phần của giao diện Start Process.
 - *void startActionPerformed(java.awt.event.ActionEvent evt)*: Nhận tên của chương trình đã nhập và chạy chương trình đó.
- ❖ Hàm bên phía client:
- Gửi tín hiệu của nút bấm đến cho server (ở đây là “Start”).
 - Dùng 1 biến (ở đây là biến *id*) để lưu trữ tên của chương trình mà người dùng đã nhập vào ở phía trên.

- Sau khi bên server thực hiện chức năng xong sẽ gửi đến client một thông báo được lưu trong biến *fromSV*:
 - Nếu *fromSV* là “Yes”: Đã chạy chương trình thành công và thông báo cho người dùng biết.



- Nếu *fromSV* là “Running”: Chương trình đã nhập đang chạy và thông báo cho người dùng biết.



- Nếu *fromSV* là “No”: Không thể chạy chương trình và thông báo cho người dùng biết lỗi đã gặp.
- Kết thúc hàm.
 - ❖ Hàm bên phía server:
- Phương thức:
 - *boolean isProcessRunning_ByName(String name)*: Nhận vào tên của chương trình đã nhập, kiểm tra xem chương trình đó có đang chạy hay không
 - *void start()*: Chạy chương trình có tên đã nhập
- Nếu server nhận được tín hiệu là “Start” sẽ tiến hành gọi hàm *start()* để thực hiện chức năng.
- Dùng 1 biến (ở đây là *id*) lưu tên của chương trình mà người dùng đã nhập bên phía client.

- Tiến hành kiểm tra xem có chương trình có *id* đã nhập đang chạy hay không:
 - o Nếu chương trình đó không đang chạy, tiến hành chạy chương trình đó và gửi phản hồi cho bên client đã chạy chương trình thành công (“Yes”).
 - o Nếu chương trình đó đang chạy, gửi phản hồi cho bên client (“Running”).
 - o Nếu không thể chạy chương trình, gửi phản hồi cho bên client (“No”) đồng thời thông báo lỗi đã gặp.
- Kết thúc hàm.

d. Chức năng Clear: Xóa bảng đã tạo từ chức năng Watch.

Phương thức *void clearButtonActionPerformed(java.awt.event.ActionEvent evt)*:

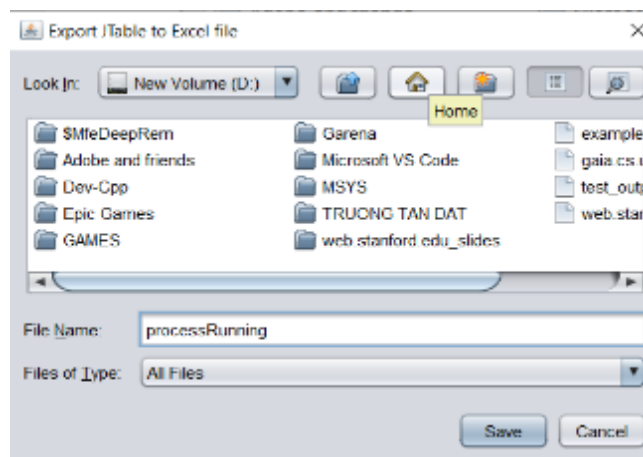
Xóa bảng chứa thông tin các chương trình.

Hàm này gọi lại bảng đã tạo ở chức năng “Watch”, sau đó đặt lại số hàng của bảng về bằng 0 và kết thúc hàm.

e. Chức năng Save: (Chức năng bổ sung thêm)

Phương thức *void saveButActionPerformed(java.awt.event.ActionEvent evt)*: Lưu bảng chứa thông tin của các chương trình vào file Excel.

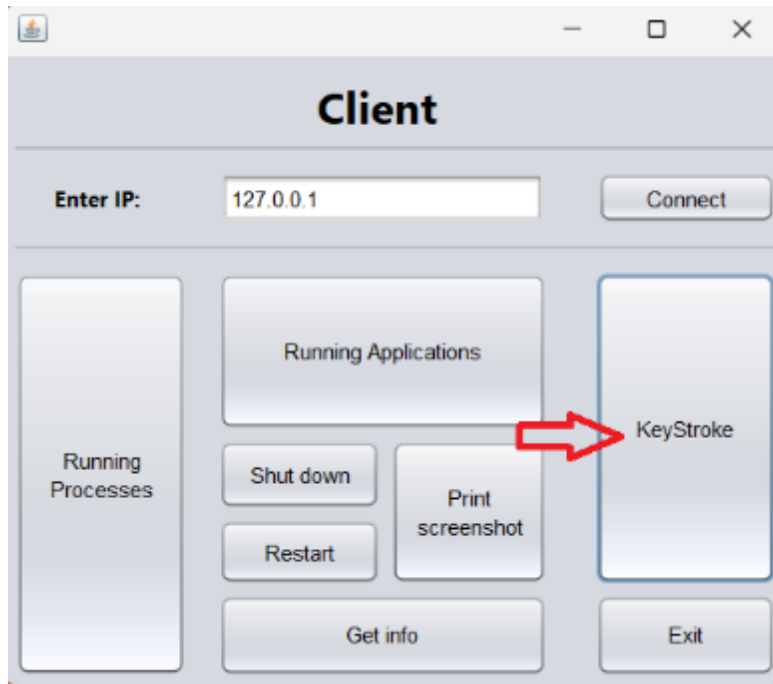
Hàm này giúp người dùng chọn địa điểm để lưu lại bảng thông tin thành file và đặt tên cho file đó



4. Bắt phím nhấn

Class Keylogger extends javax.swing.JFrame (Phía Client)

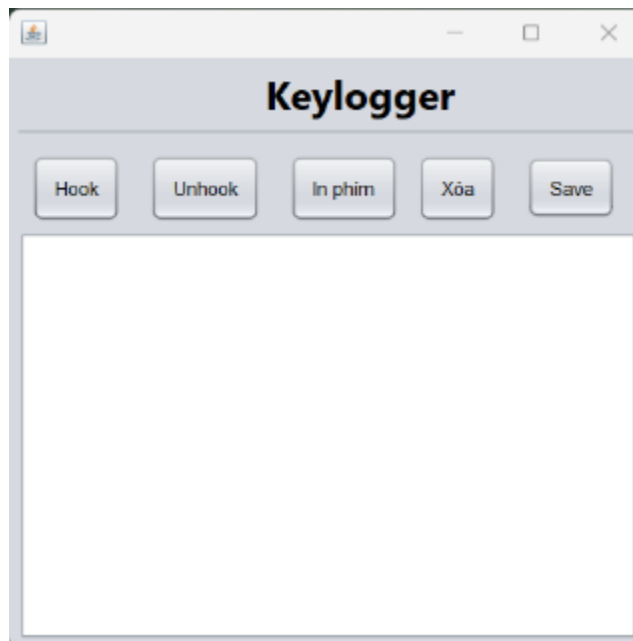
- Objects:
 - private javax.swing.JButton delKeylog;
 - private javax.swing.JButton hookBut;
 - private javax.swing.JLabel jLabel1;
 - private javax.swing.JScrollPane jScrollPane1;
 - private javax.swing.JSeparator jSeparator1;
 - private javax.swing.JTextArea keyTextArea;
 - private javax.swing.JButton printkeyBut;
 - private javax.swing.JButton saveBut;
 - private javax.swing.JButton unhookBut;
- Constructor: public Keylogger()
- Methods:
 - public void hook()
 - public void unhook()
 - public void printKey()
 - private void initComponents()
 - private void printkeyButActionPerformed
 - private void delKeylogActionPerformed
 - private void unhookButActionPerformed(java.awt.event.ActionEvent evt)
 - private void hookButActionPerformed(java.awt.event.ActionEvent evt)
 - private void saveButActionPerformed(java.awt.event.ActionEvent evt)
- Mô tả class Keylogger.java
 - Các object : str, count,
 - Sử dụng thư viện jnativehook
 - Các phương thức:
 - Constructor: Keylogger()
 - Hàm void run() để bắt đầu quá trình bắt phím
 - Sử dụng phương thức addNativeKeyListener(NativeKeyListener listener) để thêm trình nghe khoá đặc biệt để bắt phím từ hệ máy chủ server.
 - Và Field Logger để ghi lại nhật ký cho thư viện gốc.
 - Hàm void nativeKeyPressed(NativeKeyEvent e) để bắt các phím đã gõ trên máy Server và lưu vào biến string key và biến str (để gửi cho client)
 - Hàm void stop() dùng để dừng quá trình bắt phím.
 - Hàm String convert(String key) dùng để chuyển các phím đặc biệt thành kí tự của chúng ví dụ như [Period] thành dấu “.” thay vì hiển thị [Period]



❖ Cách hoạt động:

a. Chức năng Hook & Unhook

Khi click vào “KeyStroke” trên giao diện app thì Client, nếu như chưa kết nối với Server thì sẽ hiện lên message “Chưa kết nối đến server”, nếu đã kết nối đến Server thì sẽ hiện lên cửa sổ Keylogger



Tiếp theo Click vào “Hook”. Khi Click vào “Hook” thì Client sẽ gửi đến Server 1 message “Hook”. Server sẽ nhận được message “Hook” và tiến hành bắt phím ở máy Server.

- Cách gửi 1 đoạn tin nhắn từ Client sang Server
 - Khởi tạo một String có giá trị là “Hook”
 - Khởi tạo DataOutputStream, DataOutputStream để gửi dữ liệu sang Server thông qua Socket
 - Mô tả đoạn chương trình gửi message qua socket
 - o Tạo một chuỗi kí tự `message` = “Hook”
 - o Tạo DataOutputStream `out` = Socket Output Stream
 - o Gửi `message` = “Hook” thông qua `out`
- Cách Server nhận tin nhắn từ Client
 - Khởi tạo DataInputStream, DataInputStream để nhận dữ liệu từ Client thông qua Socket
 - Mô tả đoạn chương trình nhận message thông qua socket
 - o Tạo DataInputStream `in` = Socket Input Stream
 - o Tạo một chuỗi kí tự `message`
 - o Nhận `message` = “Hook” từ socket bằng `in`
- Cách bắt phím trên Server. Mô tả hàm void keylog() trong Server:
 - Tạo một String `res` để lưu các phím bắt được.
 - Khởi tạo class Keylogger để tiến hành bắt phím và bật cờ `hook_flag` thành true để đánh dấu đã bắt phím.

Khi Click vào “Unhook” thì Client sẽ gửi đến Server 1 message “Unhook”. Server sẽ nhận được message “Unhook” và tiến hành dừng bắt phím ở máy Server.

- Tương tự như cách gửi tin nhắn từ Client sang Server. Client sẽ gửi message “Unhook” cho Server và Server nhận message sẽ bật cờ `hook_flag` thành false và sẽ dừng quá trình bắt phím.

b. Chức năng In phím

Click vào “In phím” thì Client sẽ gửi message “Printkey” cho Server và Server nhận được message sẽ gửi lại cho client String str đã bắt phím khi ấn Hook nếu cờ `hook_flag` là true ngược lại thì sẽ gửi cho Client message “Unhook”.

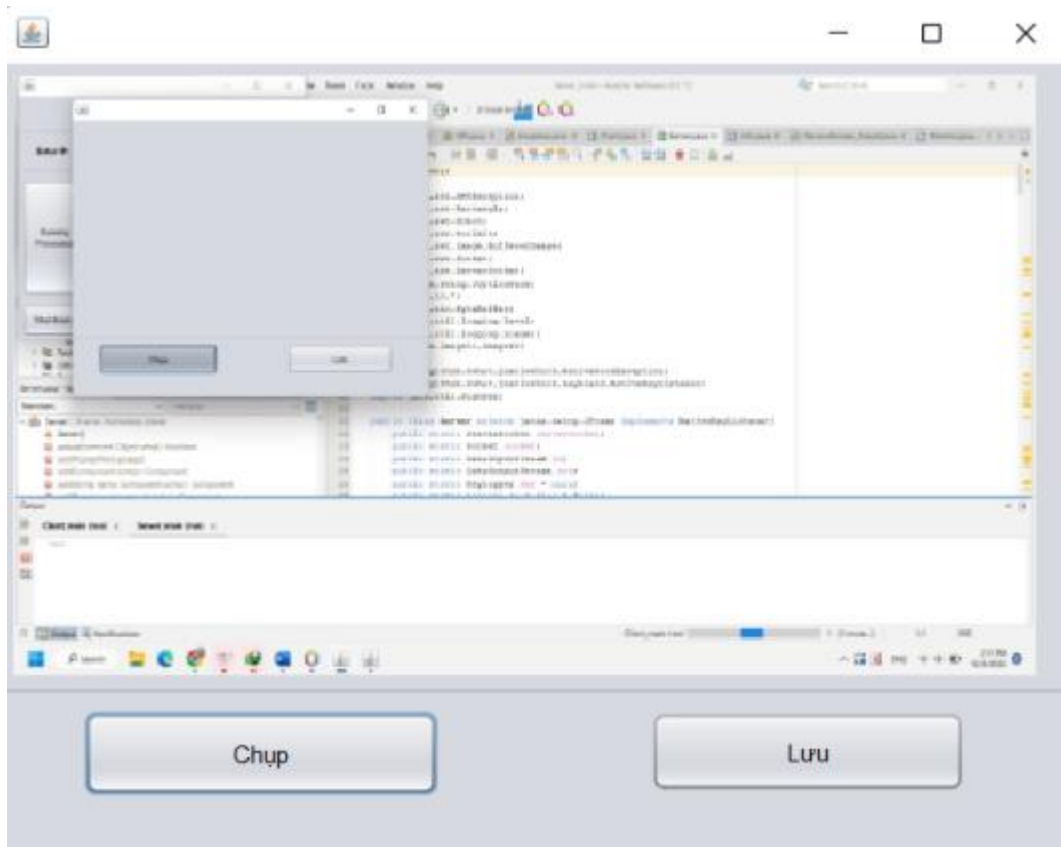
c. Chức năng Xóa

Khi click vào “Xóa” thì Client sẽ set là keyTextArea thành chuỗi rỗng bằng phương thức `setText(“”)`.

d. Chức năng Save (Chức năng bổ sung thêm)

Khi click vào “Save” thì Client sẽ lưu chuỗi bắt phím được thành file .docx vào đường dẫn và tên do người dùng chọn bằng phương thức `saveButActionPerformed(java.awt.event.ActionEvent evt)`.

5. Chụp màn hình



class Pic extends javax.swing.JFrame

- Các objects:
 - Static BufferedImage image
 - private javax.swing.JLabel imageLabel
 - private javax.swing.JSeparator jSeparator1
 - private javax.swing.JButton saveBut
 - private javax.swing.JButton takepicBut
- Constructor: public Pic()
- Method:
 - public void getPic()
 - private void takepicButActionPerformed(java.awt.event.ActionEvent evt)
 - private void saveButActionPerformed(java.awt.event.ActionEvent evt)
 - private void imageLabelAncestorAdded(javax.swing.event.AncestorEvent evt)

❖ Cách hoạt động

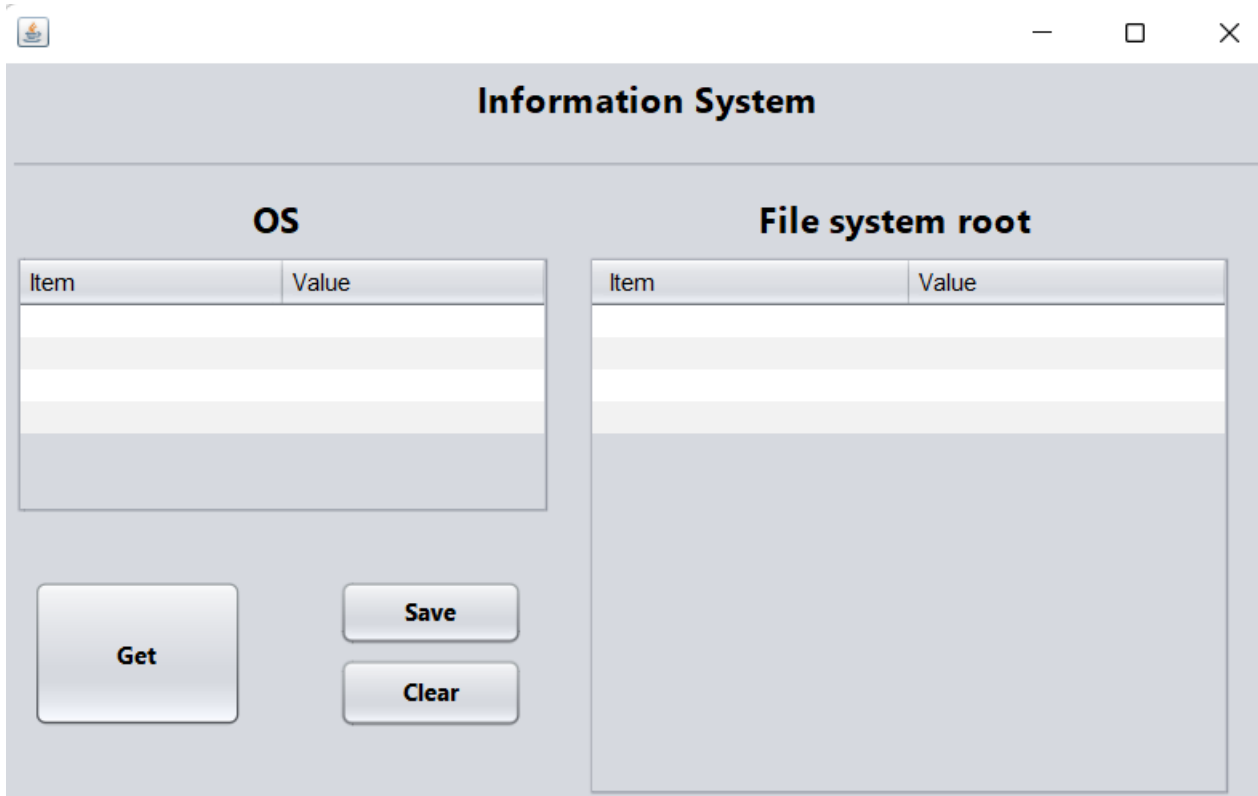
a. Chức năng Chụp

- Khi click vào “Print screenshot” trên giao diện app thì Client, hàm `picButtonActionPerformed` sẽ bắt sự kiện. Nếu như chưa kết nối với Server thì sẽ hiện lên message “Chưa kết nối đến server”, nếu đã kết nối đến Server thì tạo ra 1 đối tượng `Pic` thông qua class `Pic`.
- Sau khi đối tượng được tạo ra, sẽ khởi tạo giao diện và tạo ra đối tượng `BufferedImage`
- Khi click vào “Chụp” thì hàm `getPic()` sẽ được gọi để gửi message “Pic” đến cho Server thông qua Socket.
- Khi Server nhận được message “Pic” thì sẽ tiến hành chụp lại màn hình và gửi về cho Client. Hàm `getPic()` cũng làm nhiệm vụ nhận hình ảnh từ Server gửi đến và đưa ảnh đó vào `BufferedImage`.

b. Chức năng Lưu

- Khi click vào “Lưu”, đối tượng `SaveAs` thông qua class `SaveAs` được tạo ra để tiến hành lưu hình ảnh trong `BufferedImage` vào vị trí mong muốn.
- Hàm `saveJPG()` trong class `SaveAs` để lưu hình ảnh.

6. Lấy thông tin OS và File system (Get info) (Chức năng bổ sung thêm)



Public class Info extends javax.swing.JFrame bao gồm:

- Các objects:
 - private javax.swing.JButton clearBut;
 - private javax.swing.JTable filesysTable;
 - private javax.swing.JButton getBut;
 - private javax.swing.JLabel jLabel1;
 - private javax.swing.JLabel jLabel3;
 - private javax.swing.JLabel jLabel4;
 - private javax.swing.JScrollPane jScrollPane1;
 - private javax.swing.JScrollPane jScrollPane2;
 - private javax.swing.JSeparator jSeparator1;
 - private javax.swing.JTable osTable;
 - private javax.swing.JButton saveBut;
- Constructor
 - public Info() gọi lại method private void initComponents() dùng để khởi tạo các object và tạo thành giao diện Information System

- Các Methods:

- `private void initComponents()`: dùng để khởi tạo các object để tạo thành giao diện
- `private void getButActionPerformed (java.awt.event.ActionEvent evt)`: dùng để bắt sự kiện khi bấm vào button Get, lấy thông tin và hiện trên bảng
- `private void saveButActionPerformed(java.awt.event.ActionEvent evt)`: dùng để bắt sự kiện khi bấm vào button Save, giúp lưu thông tin ra file .xlsx
- `private void clearButActionPerformed(java.awt.event.ActionEvent evt)`: dùng để bắt sự kiện khi nhấn vào button Clear, giúp xóa thông tin table

a. Chức năng Get: Lấy thông tin OS và File system.

- Sau khi nhấn Get, bắt sự kiện bằng hàm `getButActionPerformed()`
- Gửi 1 chuỗi "Info" qua server thông qua socket
- Server sau khi nhận chuỗi sẽ gọi hàm `getInfo()`; gửi các thông tin cần thiết về client thông qua socket (các thông tin được ngăn cách bởi dấu ",",)
- Client lấy thông tin và ghi lên 2 tables

b. Chức năng Save: Lưu thông tin lấy được.

- Sau khi nhấn Save bắt sự kiện bằng hàm `saveButActionPerformed`
- Gọi `showSaveDialog` và tạo biến `fileChooser` để lưu đường dẫn chọn lưu file .xlsx
- Tạo đối tượng `SaveAs` truyền vào 2 tables, đường dẫn để lưu dữ liệu trên 2 tables vào 2 sheets của file excel

c. Chức năng Clear: Xóa thông tin đã lấy được.

- Sau khi nhấn Clear, bắt sự kiện bằng hàm clearButActionPerformed
- Set số hàng về 0 để xóa table

7. Truyền màn hình: Display (Chức năng bổ sung thêm)**class Monitor extends javax.swing.JFrame**

- Các objects:
- ReceiveScreen_thread rec;
- private javax.swing.JPanel displayPanel;
- private javax.swing.JButton exitButton;
- private javax.swing.JSeparator jSeparator1;
- private javax.swing.JButton prtButton;
- Constructor
- public Monitor()
- Methods:
- public void receiveScreen()
- private void initComponents()
- public static BufferedImage imgToBuff(Image img)
- private void prtButtonActionPerformed (java.awt.event.ActionEvent evt)
- private void formWindowClosing(java.awt.event.WindowEvent evt)
- private void exitButtonActionPerformed(java.awt.event.ActionEvent evt)

class ReceiveScreen_thread extends Thread

- Các objects:
- Socket socke;
- JPanel displayPanel;
- public static Image image1;
- Constructor
- public ReceiveScreen_thread(Socket socket, JPanel displayPanel)
- Methods: @Override public void run()

Class SendScreen extends Thread

- Các objects: Socket socket = null;
- Constructor: public SendScreen(Socket socket)
- Methods: @Override public void run()

a. Chức năng xem trực tiếp màn hình của server

- Sau khi bấm Display hàm displayButtonActionPerformed sẽ bắt sự kiện và tạo ra 1 đối tượng Monitor thông qua class Monitor
- Sau khi đối tượng được tạo ra, sẽ khởi tạo giao diện và tạo một đối tượng ReceiveScreen_thread giúp dùng nhận ảnh truyền về thông qua 1 thread mới
- Gọi hàm receiveScreen() để gửi chuỗi “Display” qua server thông qua socket và bắt đầu chạy đối tượng ReceiveScreen_thread bằng lệnh start()
- Khi Server nhận chuỗi sẽ tạo một đối tượng SendScreen chạy trên thread mới giúp gửi hình ảnh màn hình liên tục cho Client
- Hàm run() trong ReceiveScreen_thread sẽ liên tục nhận hình ảnh và cập nhật lên Frame

b. Chức năng Chụp: Chụp lại màn hình, tạm dừng truyền màn hình và lưu ra file .jpg

- Khi bấm Chụp sẽ dừng truyền màn hình, dừng thread của ReceiveScreen_thread. Đồng thời gửi chuỗi "stop display" cho server. Server nhận được chuỗi cũng đồng thời dừng thread của SendScreen. Màn hình sẽ hiện ảnh vừa chụp
- Tạo một đối tượng SaveAs truyền vào image. Đối tượng SaveAs lúc này gọi showSaveDialog và tạo biến fileChooser để lưu đường dẫn chọn lưu file .jpg
- Sau đó ảnh được lưu lại

c. Chức năng Exit: Thoát cửa sổ truyền màn hình

- Bắt sự kiện khi nhấn Exit, đóng Frame Monitor bằng lệnh dispose();

8. Shutdown

❖ Bên phía Client

- Sau khi bấm vào button Shutdown, sự kiện sẽ được bắt bằng hàm private void shutdownButtonActionPerformed giúp gửi chuỗi “Shutdown” thông qua socket.

❖ Bên phía Server

- Sau khi nhận chuỗi “Shutdown” từ server gọi hàm shutdown(), hàm shutdown sẽ shutdown server bằng lệnh của lớp Runtime.

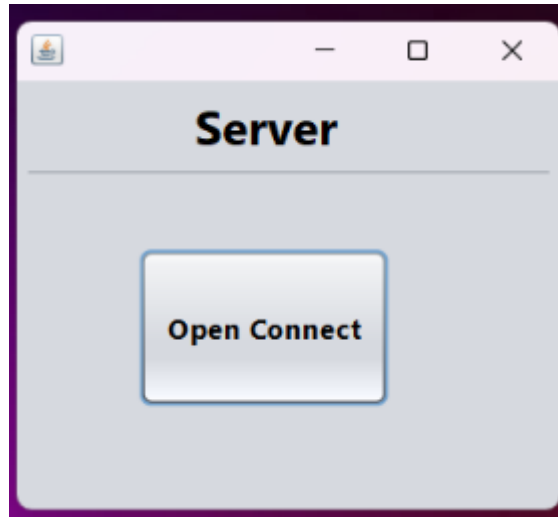
9. Exit: Thoát cửa sổ client, ngắt kết nối

- Sau khi bấm vào Exit, sự kiện sẽ được bắt bằng hàm private void shutdownButtonActionPerformed, hàm này gọi lệnh System.exit(0); giúp kết thúc chương trình phía Client, Server sẽ trở về trạng thái chờ kết nối.

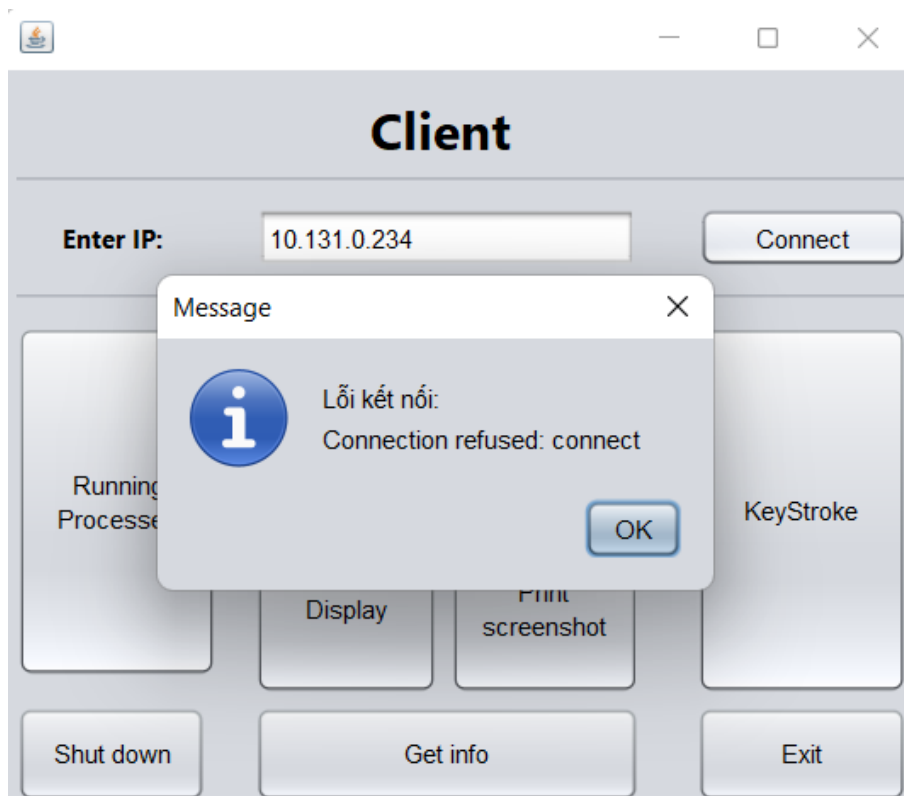
C. HƯỚNG DẪN SỬ DỤNG CHƯƠNG TRÌNH

I. Kết nối Server – Client

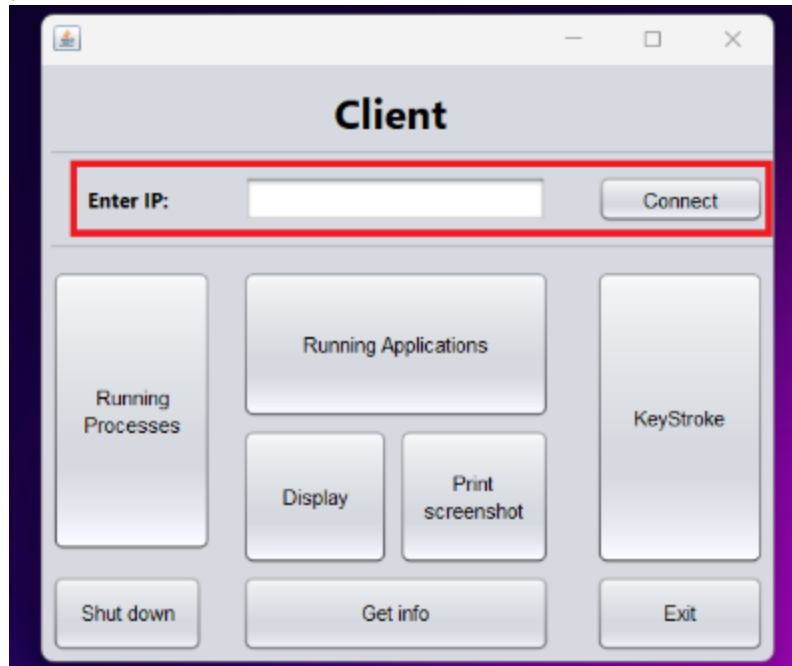
- Bước 1: Trên giao diện Server, nhấp vào **Open Connect** để Server vào trạng thái chờ kết nối.



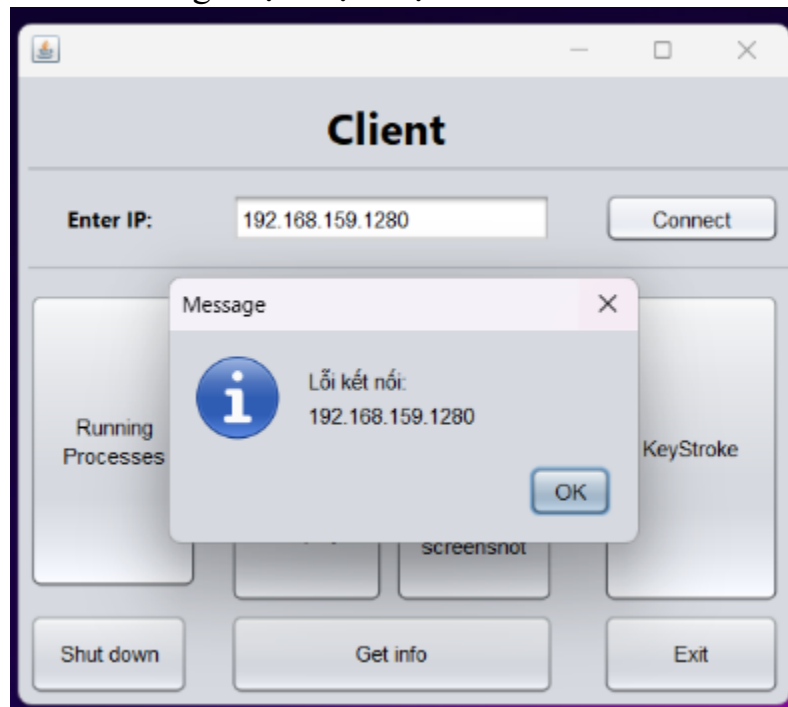
- Nếu Server chưa được mở trạng thái chờ kết nối, thì sẽ hiện ra thông báo **“Lỗi kết nối: Connection refused: connect”**.



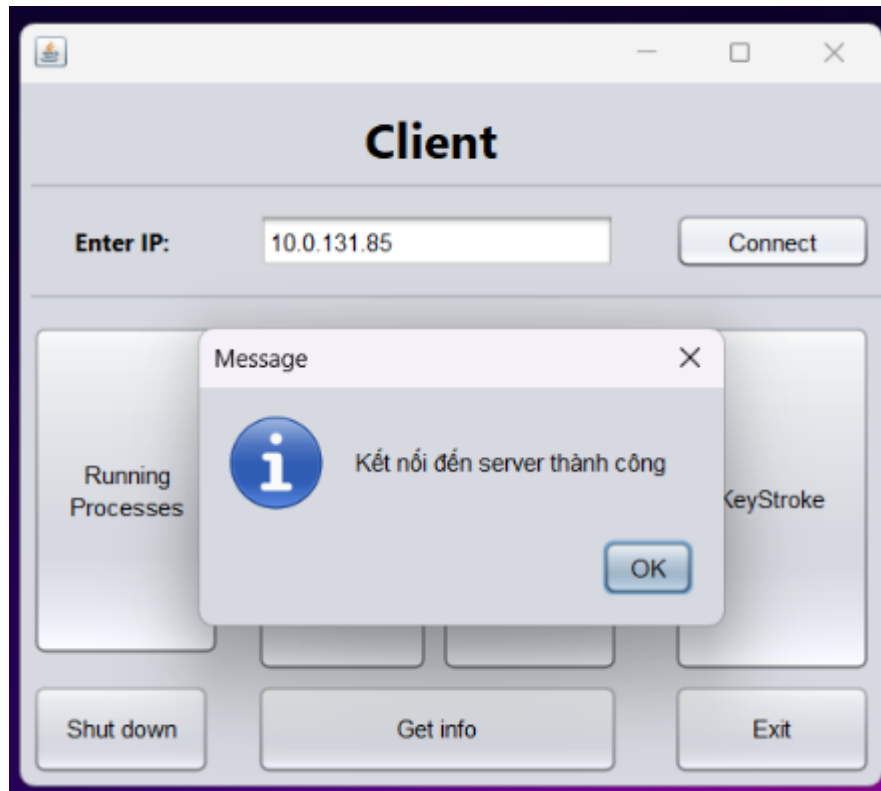
- Bước 2: Trên giao diện Client, tiến hành nhập địa chỉ IP của Server và nhấp chọn **Connect** để tạo kết nối.



- Nếu nhập sai địa chỉ IP của Server, màn hình Client sẽ hiện ra thông báo **“Lỗi kết nối”** và kết nối không được thực hiện.



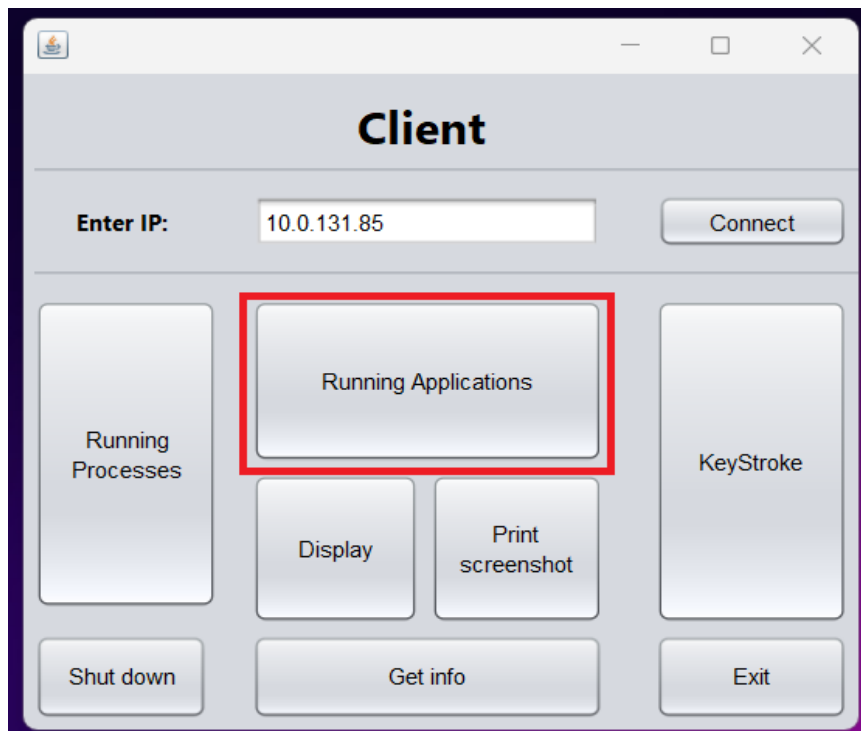
- Khi nhập đúng địa chỉ IP của Server, màn hình sẽ hiện ra thông báo **“Kết nối đến server thành công”**.

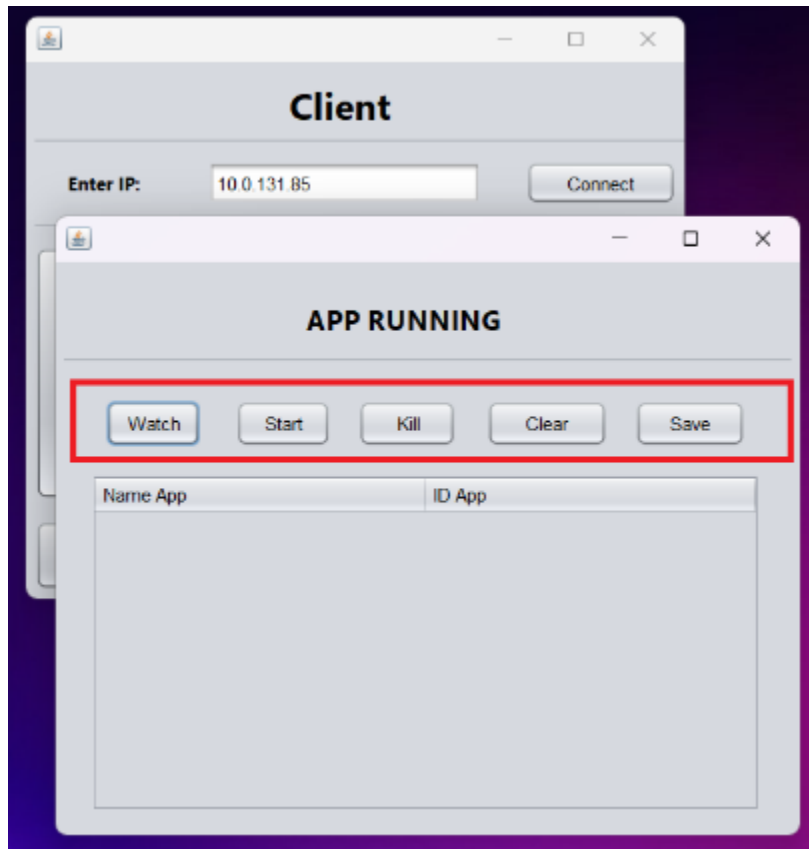


❖ Lưu ý: Để thực hiện các chức năng khác của chương trình, cần phải thực hiện kết nối thành công Client đến Server.

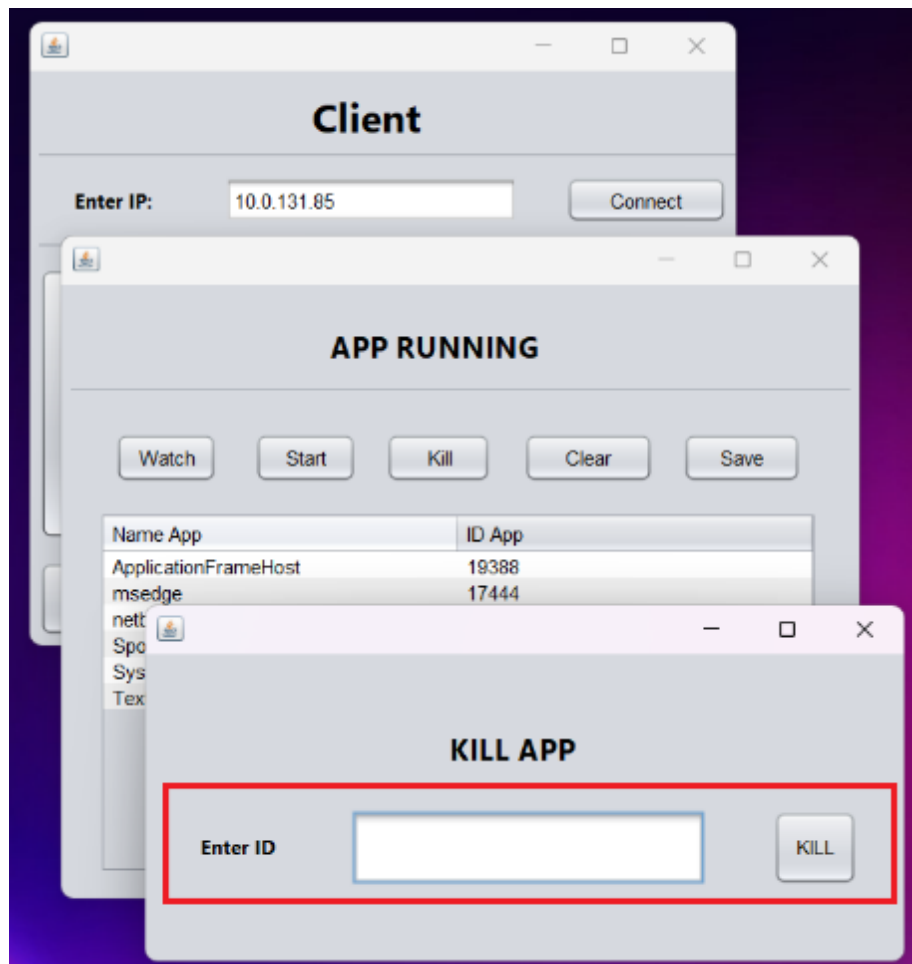
II. Liệt kê các application đang chạy trong máy, start/stop 1 app

Trên cửa sổ giao diện của Client, nhấp chọn **Running Application** để mở cửa sổ chức năng App running.



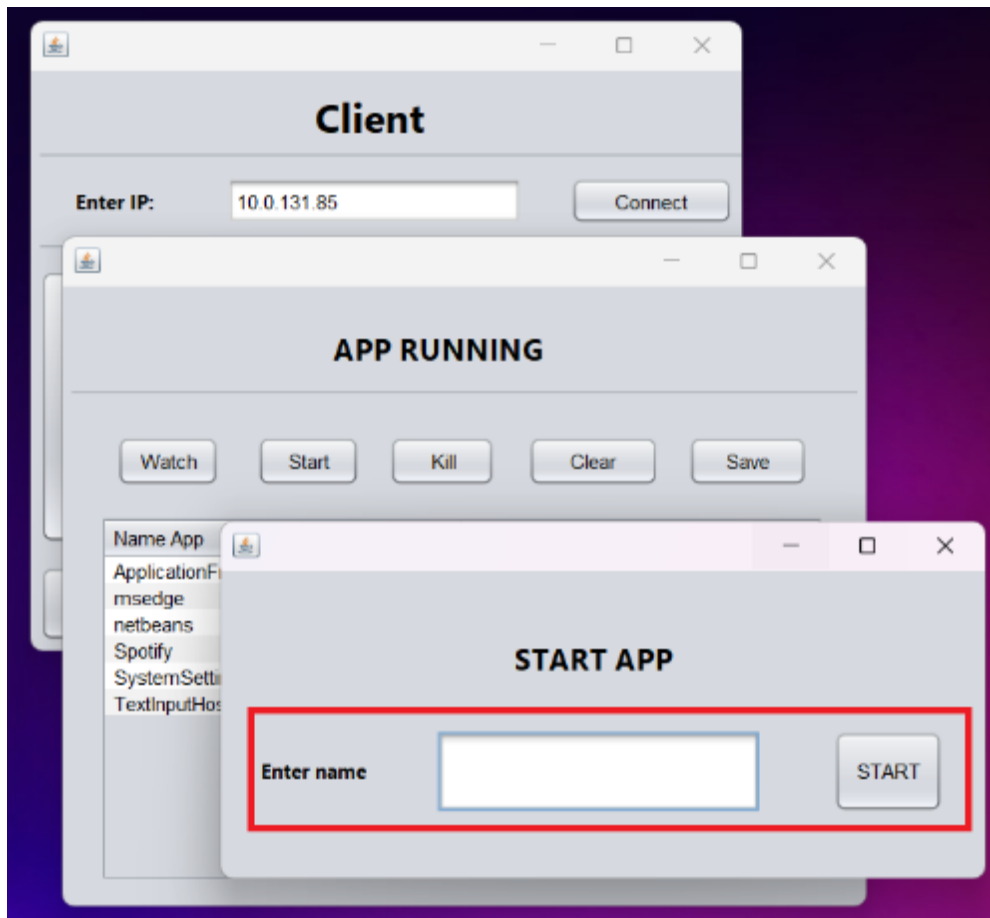


- Để xem danh sách app đang chạy trên máy, nhấp chọn **Watch**.
- Để tắt một app đang chạy trên máy, nhấp chọn Kill. Sau đó nhập ID của app cần tắt ở cửa sổ *Kill app* và nhấp **Kill**.

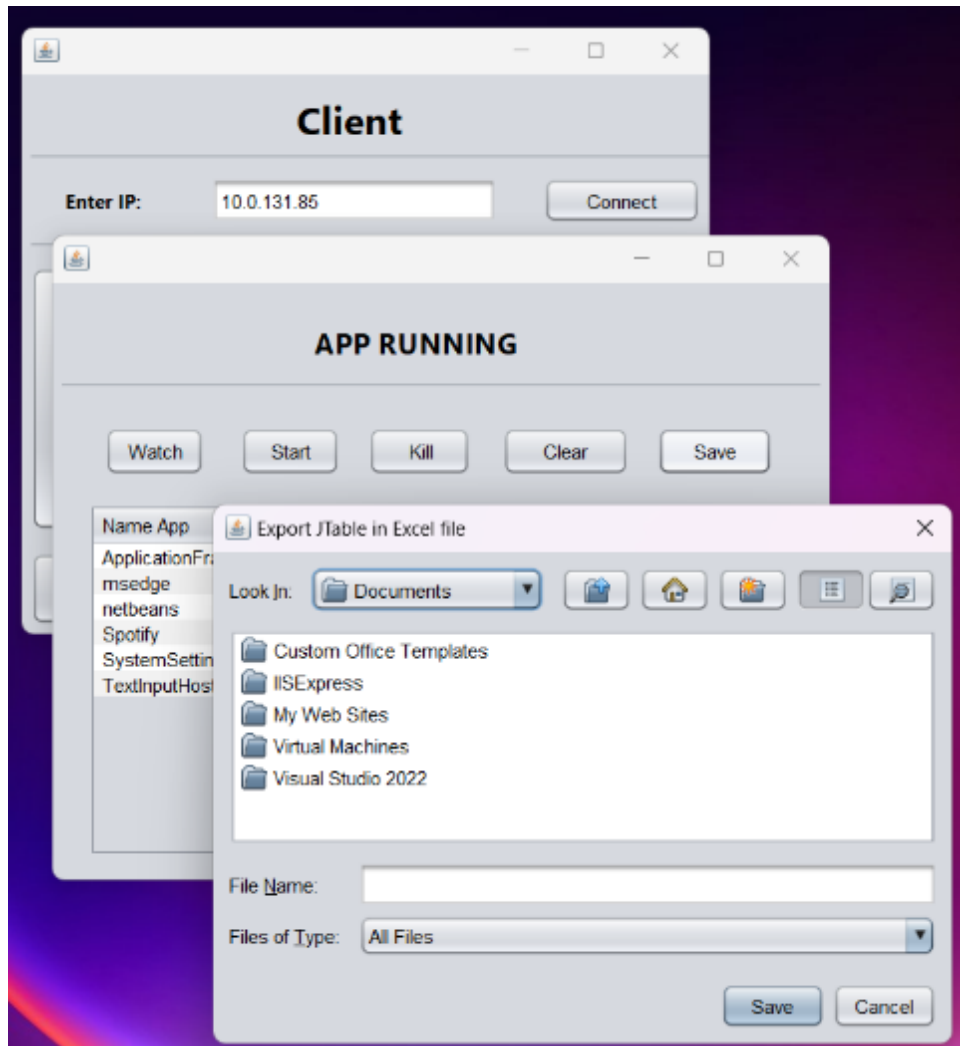


- Nếu không có app đang chạy nào ứng với ID được nhập vào, màn hình Client sẽ hiện ra thông báo ***“The process is not running!”***.
- Nếu có app đang chạy ứng với ID được nhập vào, màn hình Client sẽ hiện ra thông báo ***“Killed process successfully!”*** và app đó sẽ được tắt đi.

- Để bật một app, nhấp chọn **Start**. Sau đó cửa sổ *Start app* hiện ra, nhập tên của app cần bật và nhấp **Start** để bật app.



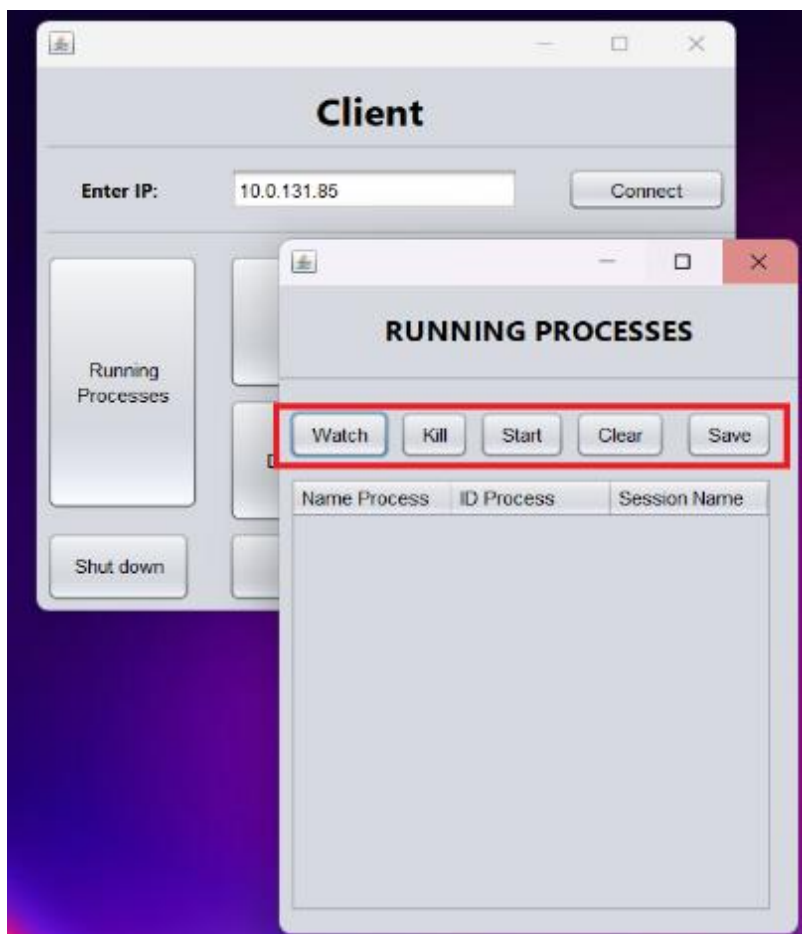
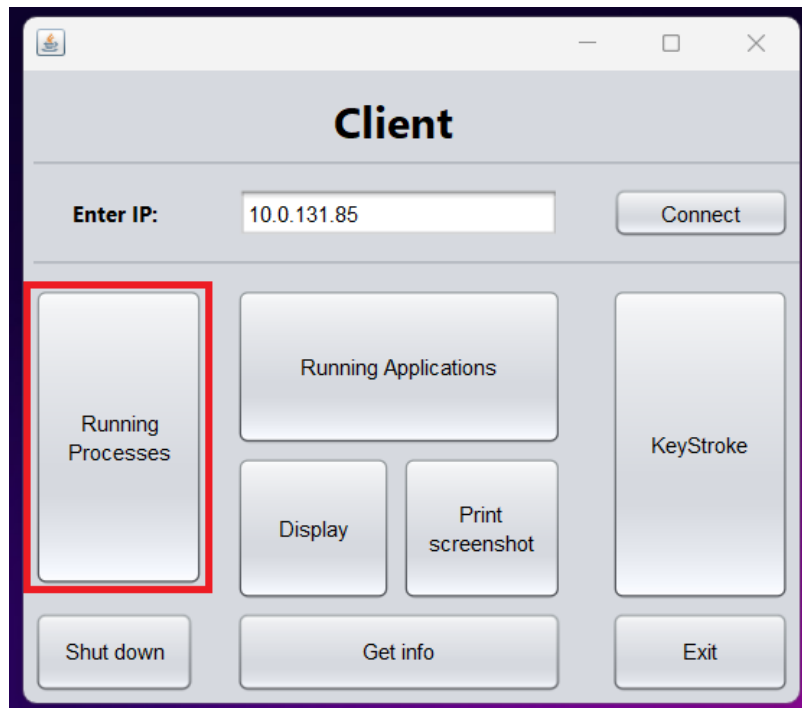
- Để lưu lại tiến trình các app này, nhấp chọn **Save**. Cửa sổ *Export/Table to Excel file* hiện ra, chọn thư mục nơi lưu trữ, sau đó chọn **Save** để tiến hành lưu lại dưới dạng file Excel.



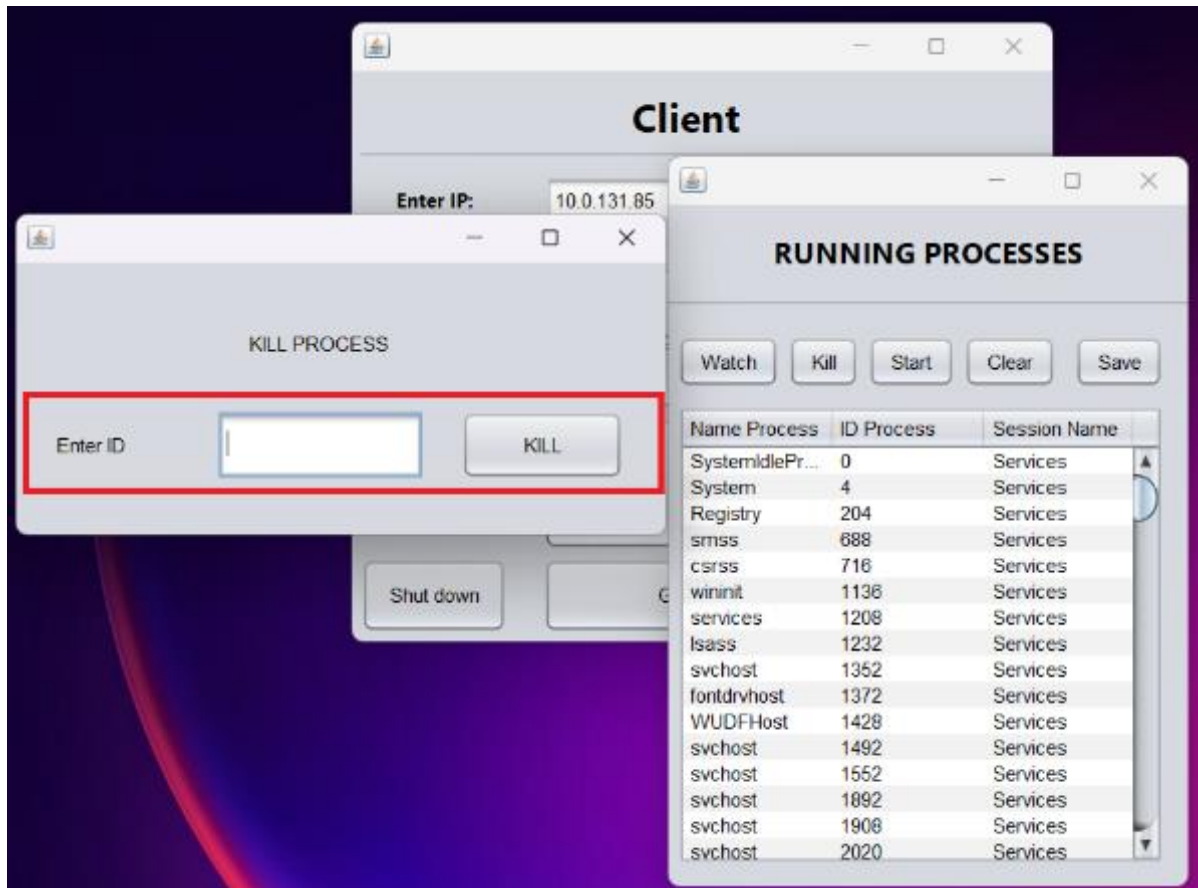
- Để xóa bảng danh sách các app đang chạy, nhấp chọn **Clear**.

III. Liệt kê các processes đang chạy trong máy, start/stop 1 process

Trên cửa sổ giao diện của Client, nhấp chọn **Running Processes** để mở cửa sổ chức năng *Running Processes*.

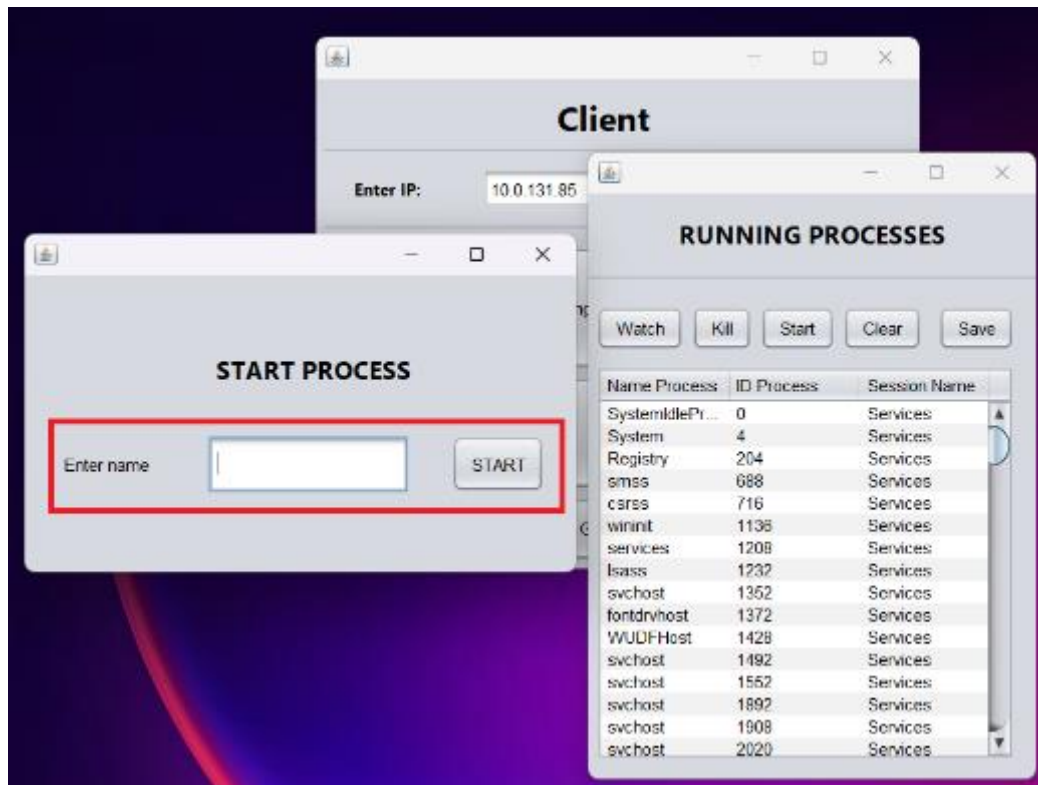


- Để xem danh sách processes đang chạy trên máy, nhấp chọn **Watch**.
- Để tắt một process đang chạy trên máy, nhấp chọn Kill. Sau đó nhập ID của process cần tắt ở cửa sổ **Kill process** và nhấp **Kill**.

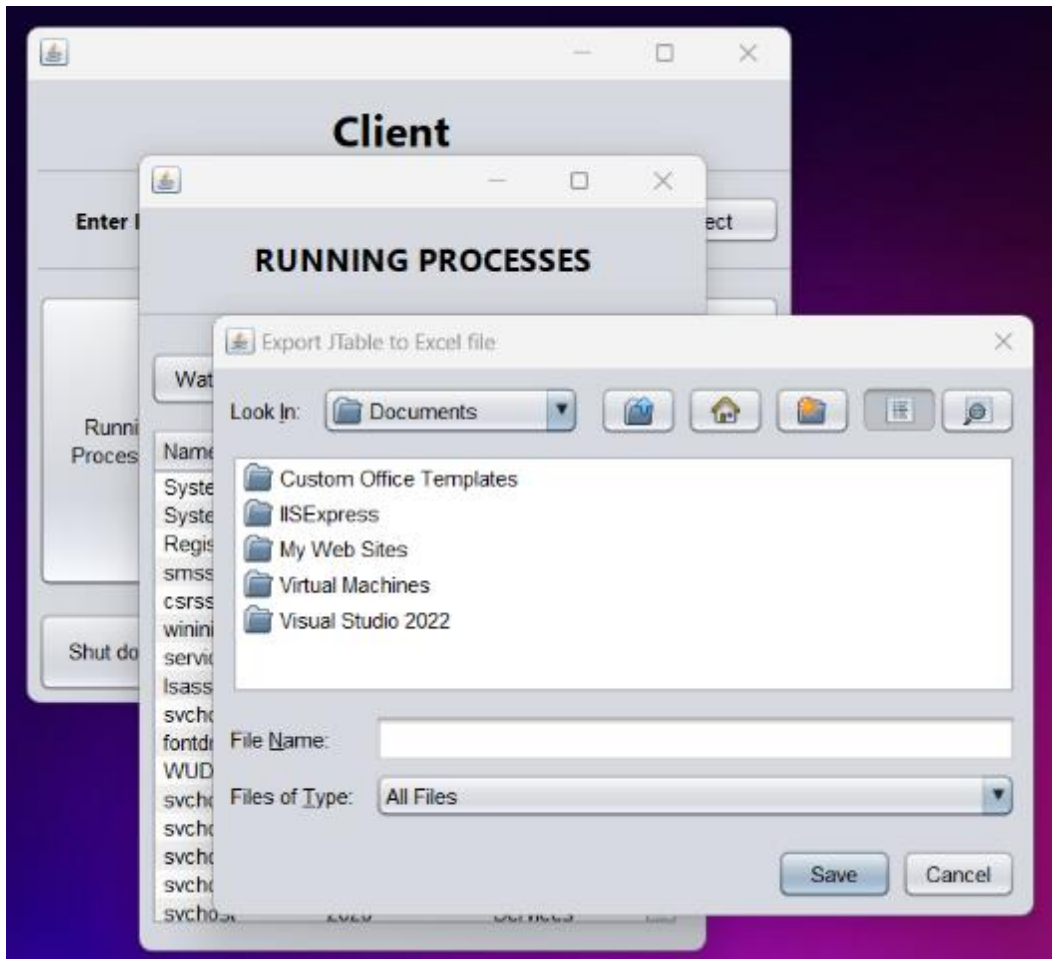


- Nếu không có process đang chạy nào ứng với ID được nhập vào, màn hình Client sẽ hiện ra thông báo ***"The process is not running!"***.
- Nếu có process đang chạy ứng với ID được nhập vào, màn hình Client sẽ hiện ra thông báo ***"Killed process successfully!"*** và process đó sẽ được tắt đi.

- Để bật một process, nhấp chọn **Start**. Sau đó cửa sổ *Start Process* hiện ra, nhập tên của process cần bật và nhấp **Start** để bật process.



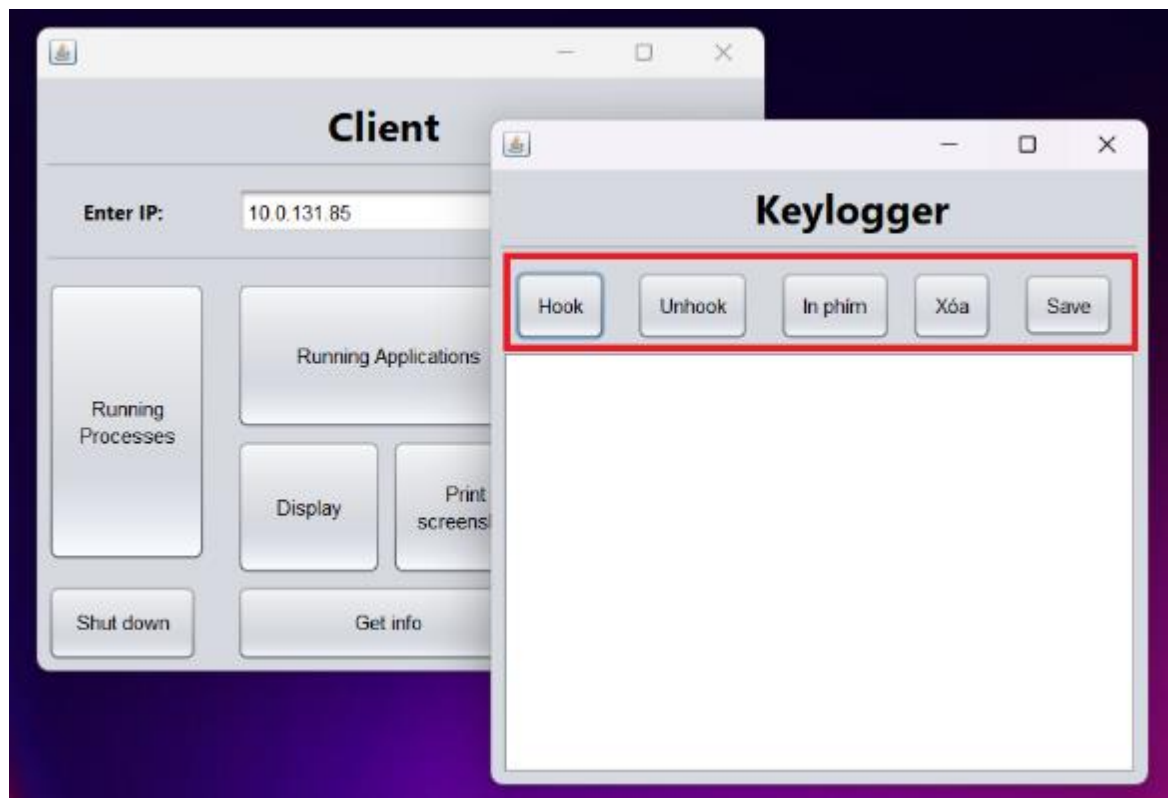
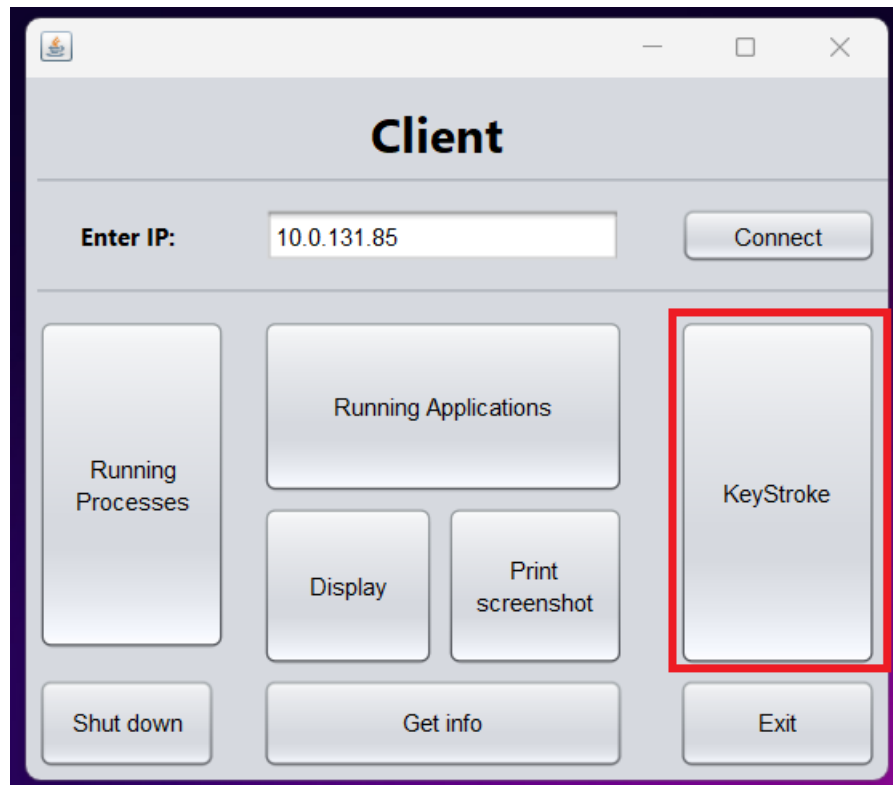
- Để lưu lại tiến trình các process này, nhấp chọn **Save**. Cửa sổ *Export/Table to Excel file* hiện ra, chọn thư mục nơi lưu trữ, sau đó chọn **Save** để tiến hành lưu lại dưới dạng file Excel.



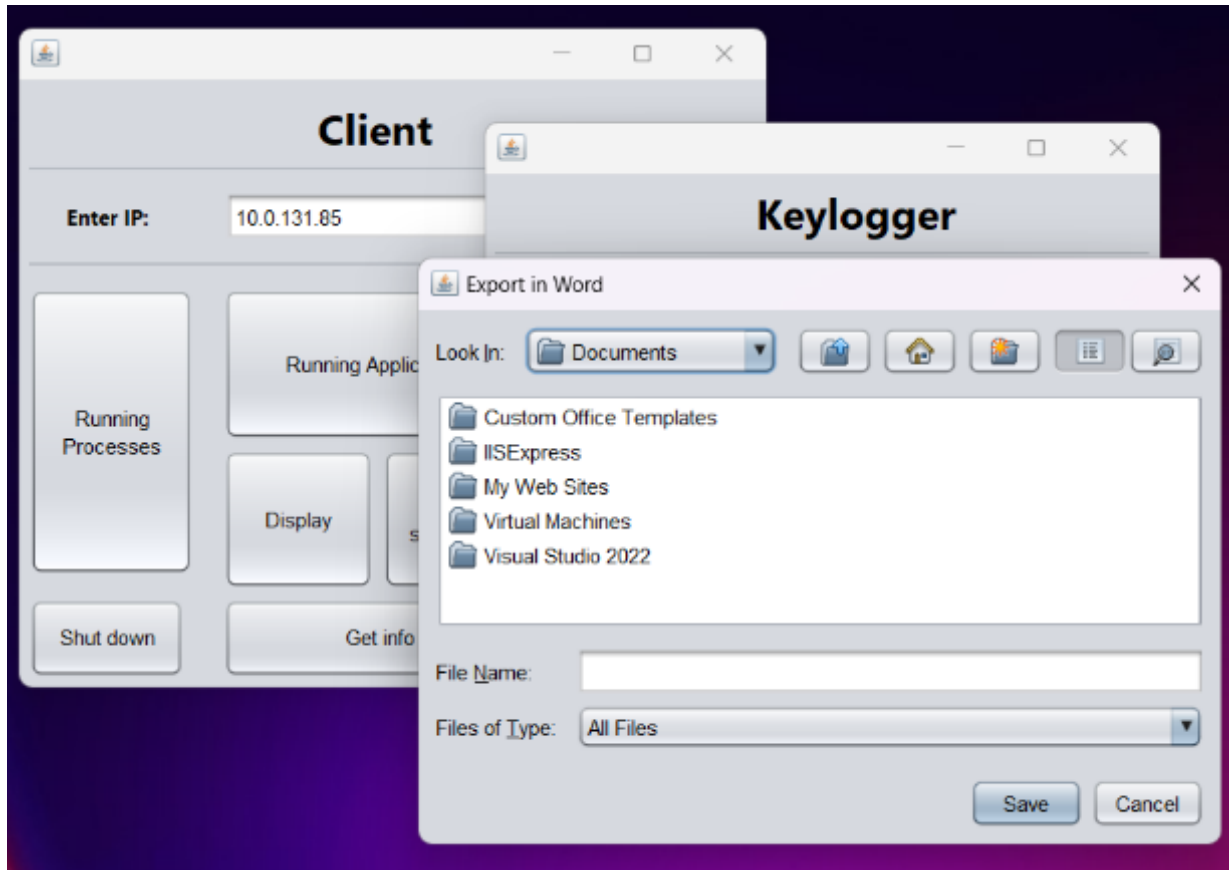
- Để xóa bảng danh sách các processes đang chạy, nhấp chọn **Clear**.

IV. Bắt phím nhấn

Trên cửa sổ giao diện của Client, nhấp chọn **KeyStroke** để mở cửa sổ chức năng *Keylogger*.

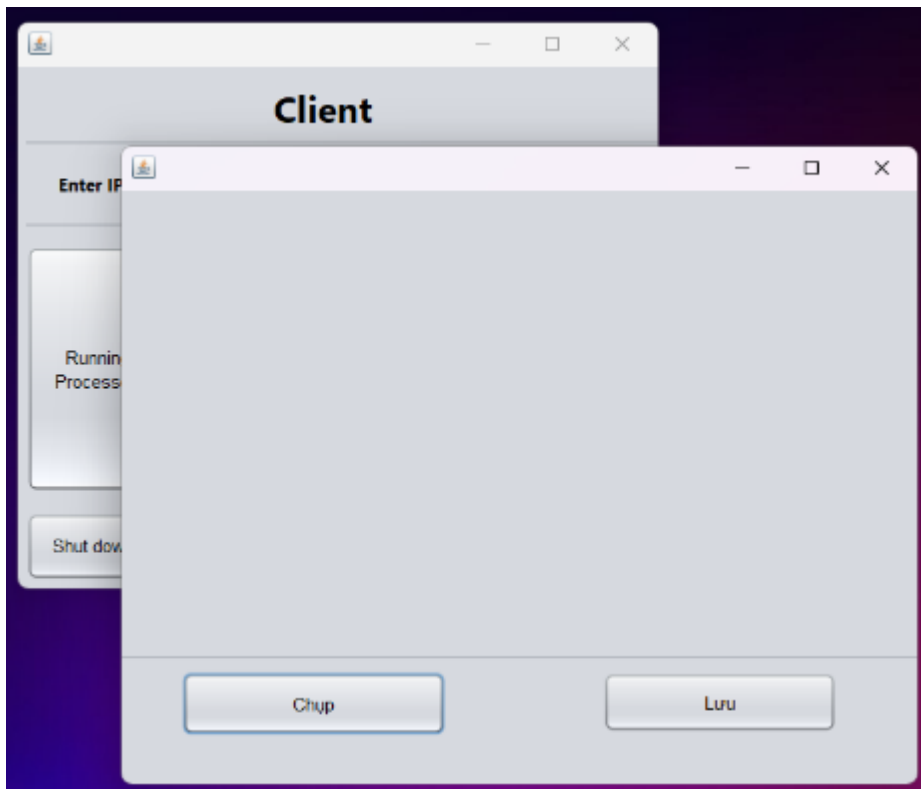
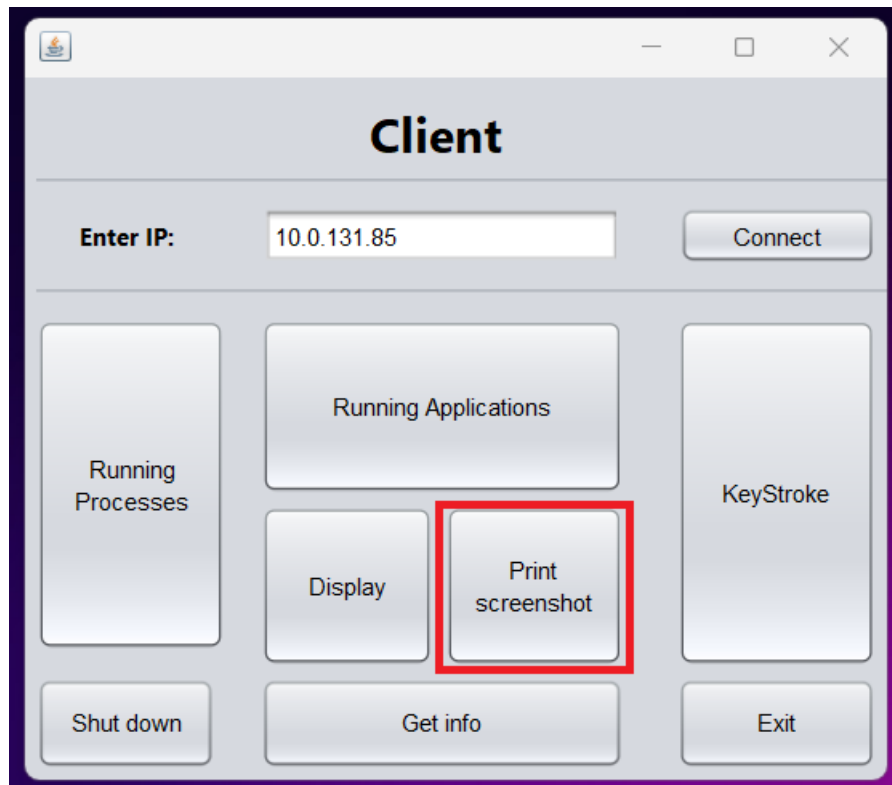


- Để tiến hành bắt phím nhấn bên Server, nhấp chọn **Hook**.
- Để in ra các phím đã bắt được, nhấp chọn **In phím**.
- Để tạm dừng bắt phím, nhấp chọn **Unhook**.
- Để lưu lại, nhấp chọn **Save**. Cửa sổ *Export in Word* hiện ra, chọn thư mục nơi lưu trữ, sau đó chọn **Save** để tiến hành lưu lại dưới dạng file Word.



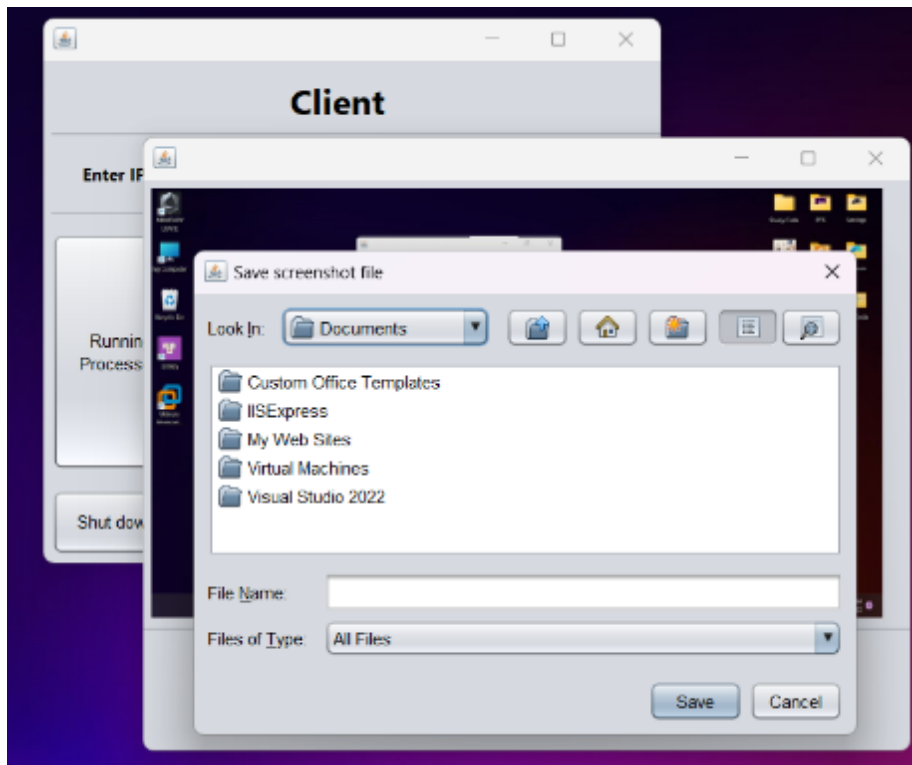
V. Chụp màn hình

Trên cửa sổ giao diện của Client, nhấp chọn **Print Screenshot** để mở cửa sổ chức năng *Chụp màn hình*.



- Để chụp màn hình của Server, nhấp chọn **Chụp**. Hình ảnh chụp được sẽ được hiển thị trên màn hình của Client.

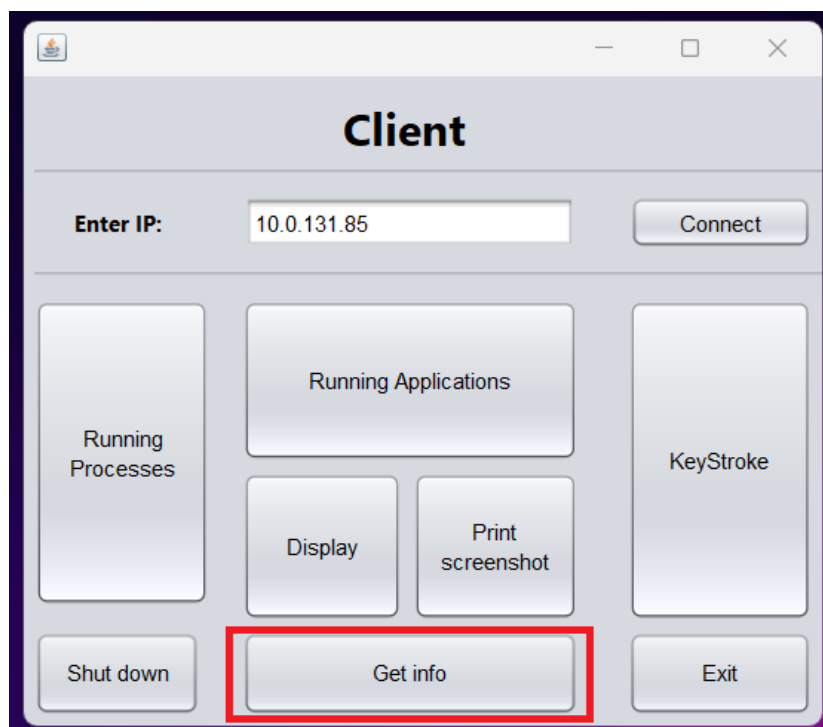
- Để lưu lại hình ảnh đã chụp, nhấp chọn **Lưu**. Cửa sổ *Save screenshot file* hiện ra, chọn thư mục nơi lưu trữ, sau đó chọn **Save** để tiến hành lưu lại.

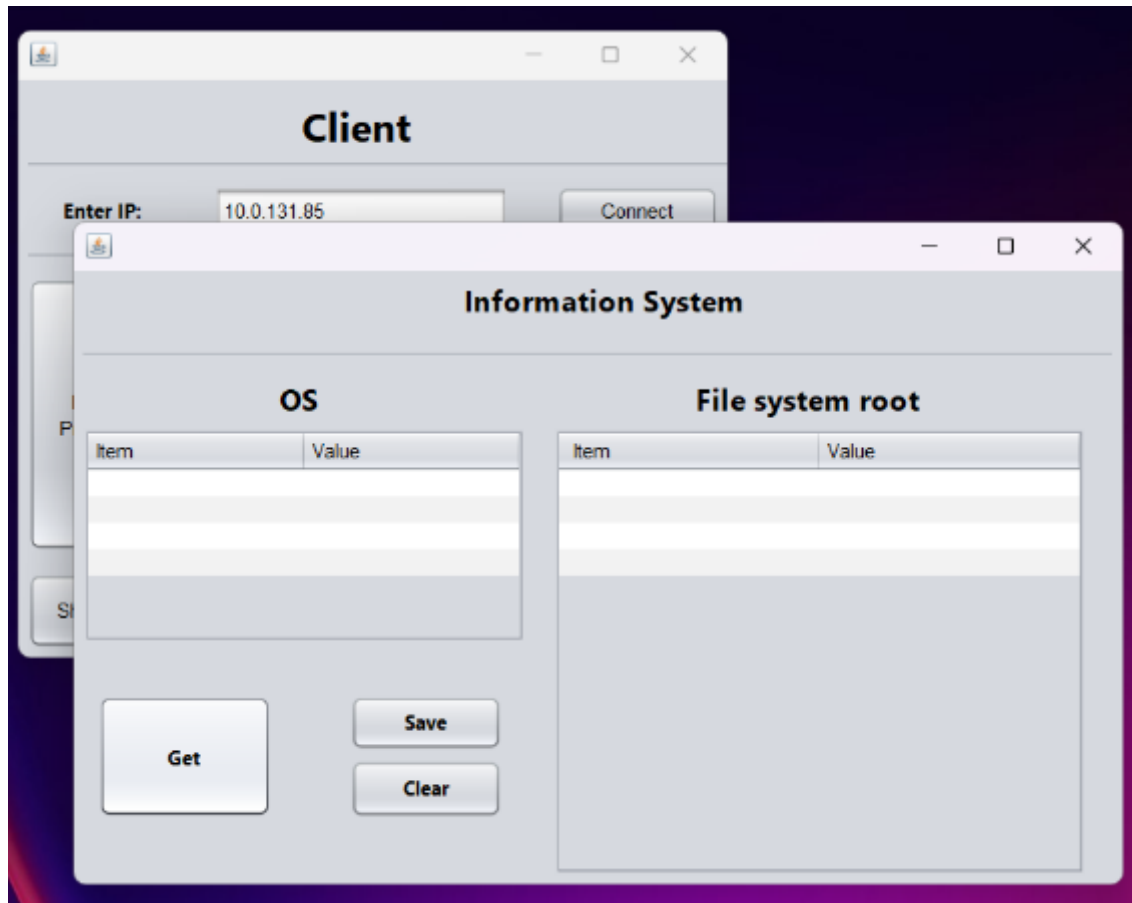


❖ *Thao tác Chụp và Lưu có thể thực hiện lặp lại nhiều lần.*

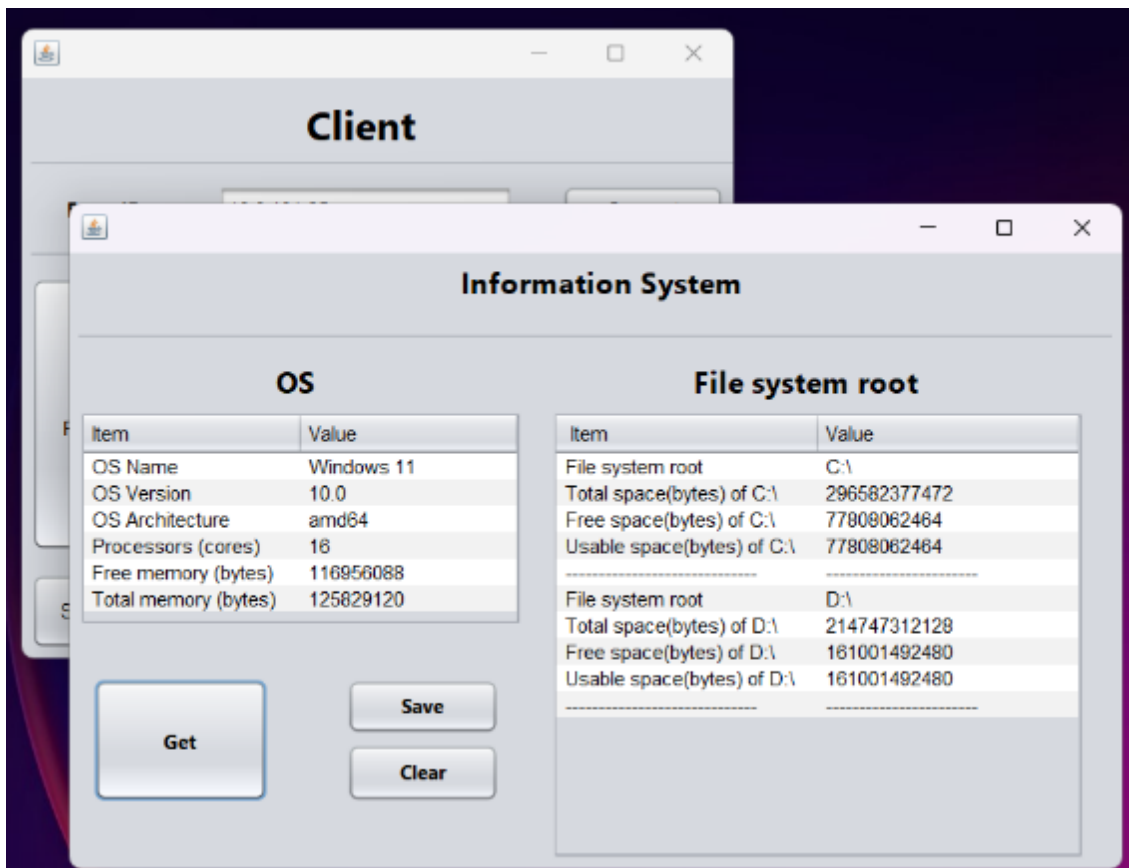
VI. Lấy thông tin OS và File system (Get info)

Trên cửa sổ giao diện của Client, nhấp chọn **Get info** để mở cửa sổ chức năng *Information System*.

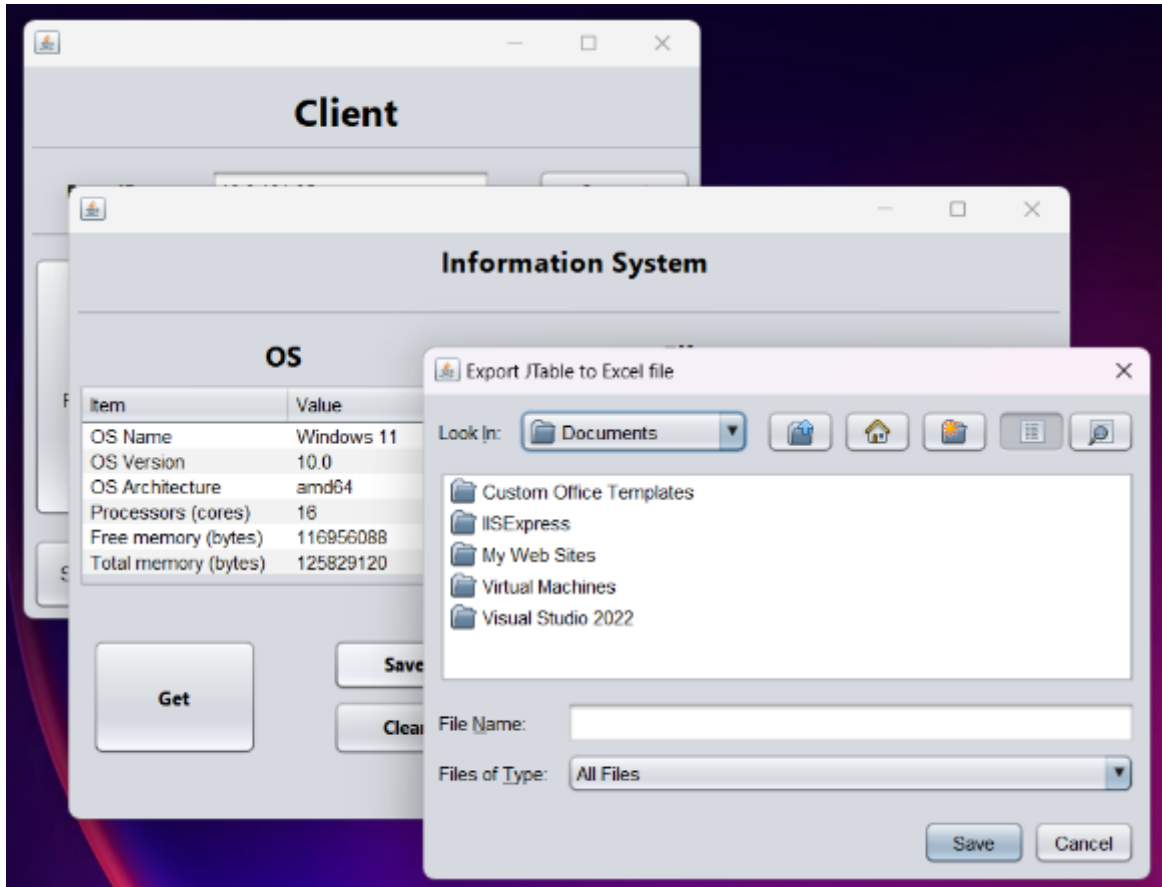




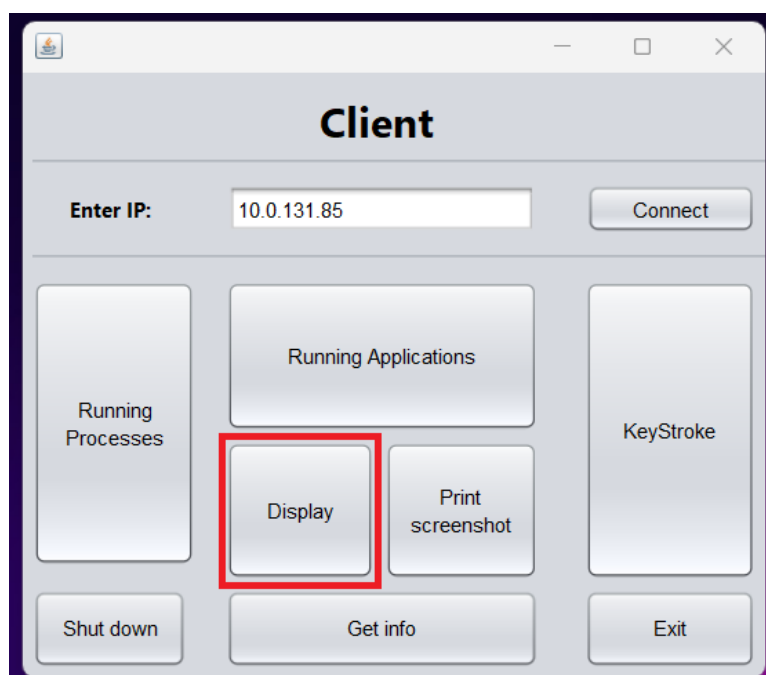
- Để xem thông tin của OS và File system của Server, nhấp chọn **Get**.



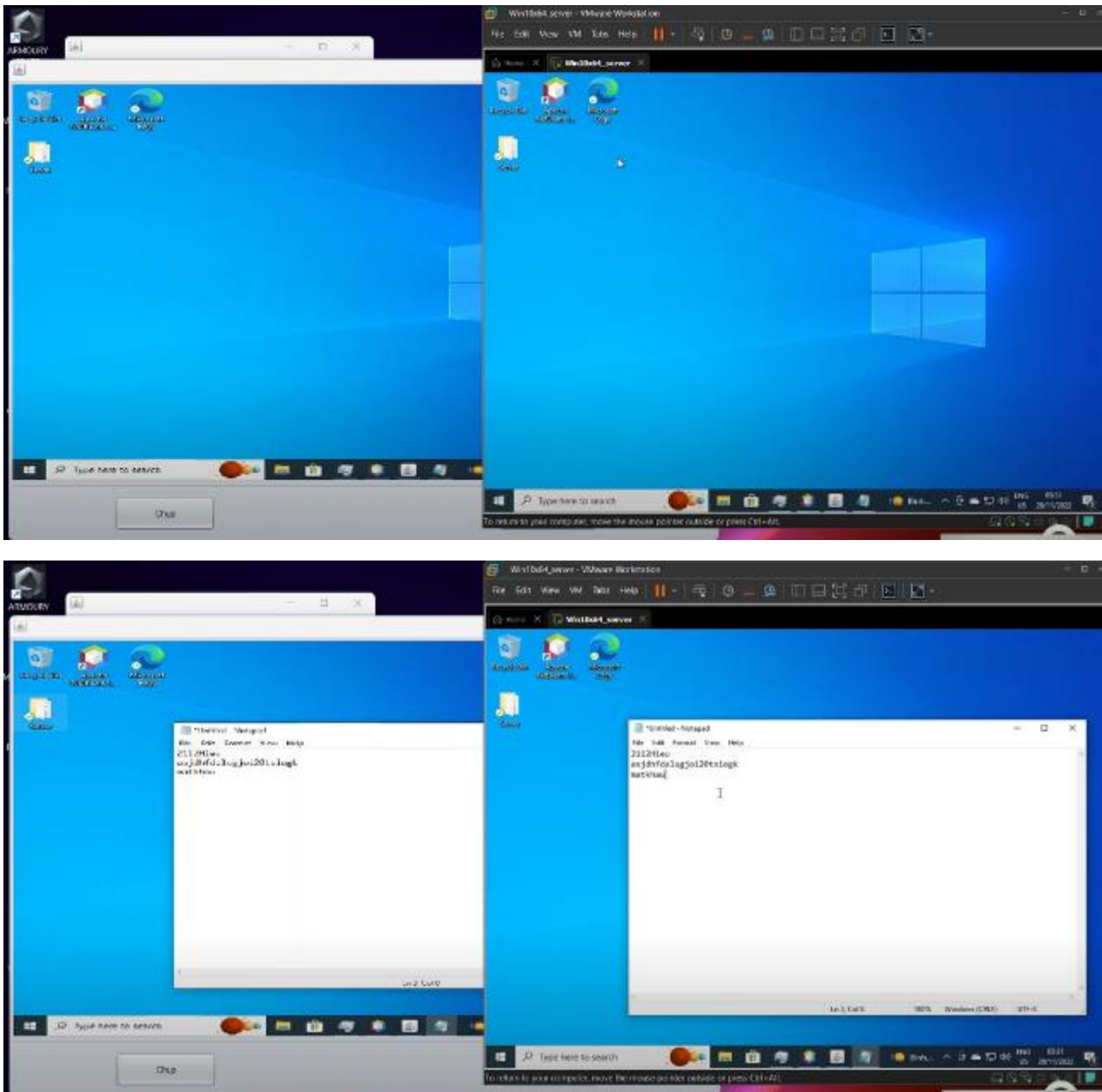
- Để xóa bảng thông tin của OS và File system của Server, nhấp chọn **Clear**.
- Để lưu lại tiến trình các thông tin này, nhấp chọn **Save**. Cửa sổ *Export/Table to Excel file* hiện ra, chọn thư mục nơi lưu trữ, sau đó chọn **Save** để tiến hành lưu lại dưới dạng file Excel.



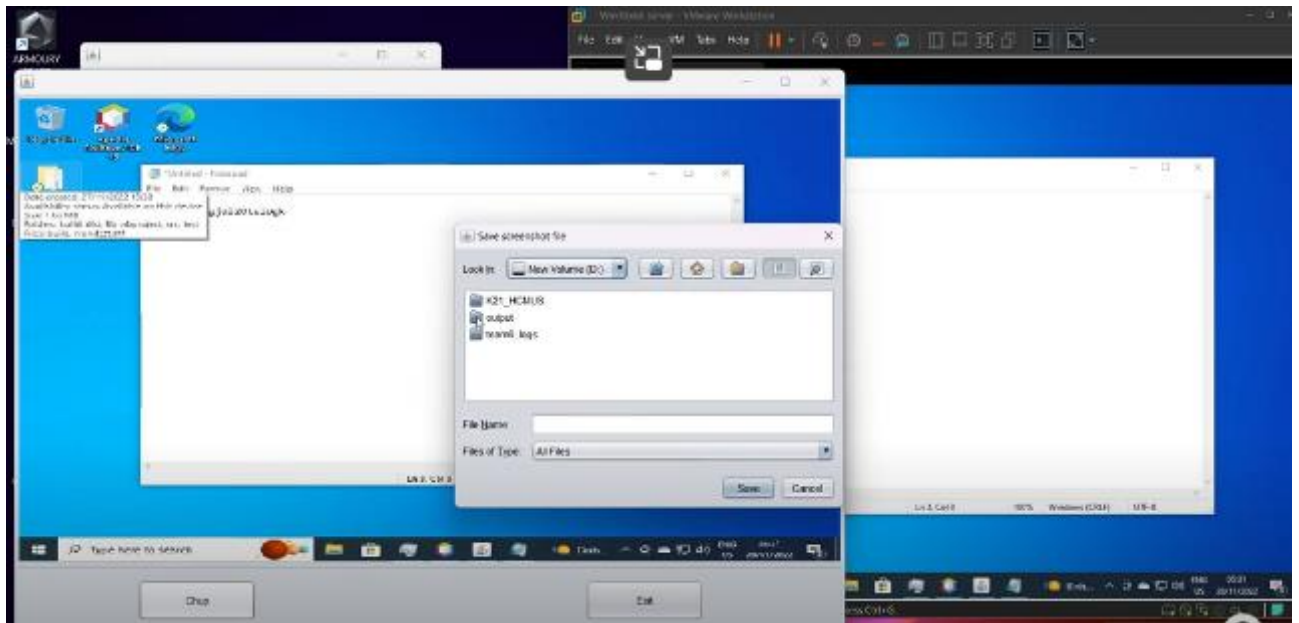
VII. Truyền màn hình (Display)



- Trên cửa sổ giao diện của Client, nhấp chọn **Display**. Một cửa sổ hiện ra hiển thị màn hình của Server được truyền liên tục đến Client, bất cứ thay đổi nào trên màn hình của Server cũng được truyền liên tục, giống như đang xem trực tiếp.



- Để chụp màn hình của Server, nhấp chọn **Chụp**. Hình ảnh chụp được sẽ được hiển thị trên màn hình của Client. Cửa sổ *Save screenshot file* hiện ra, chọn thư mục nơi lưu trữ, sau đó chọn **Save** để tiến hành lưu lại.

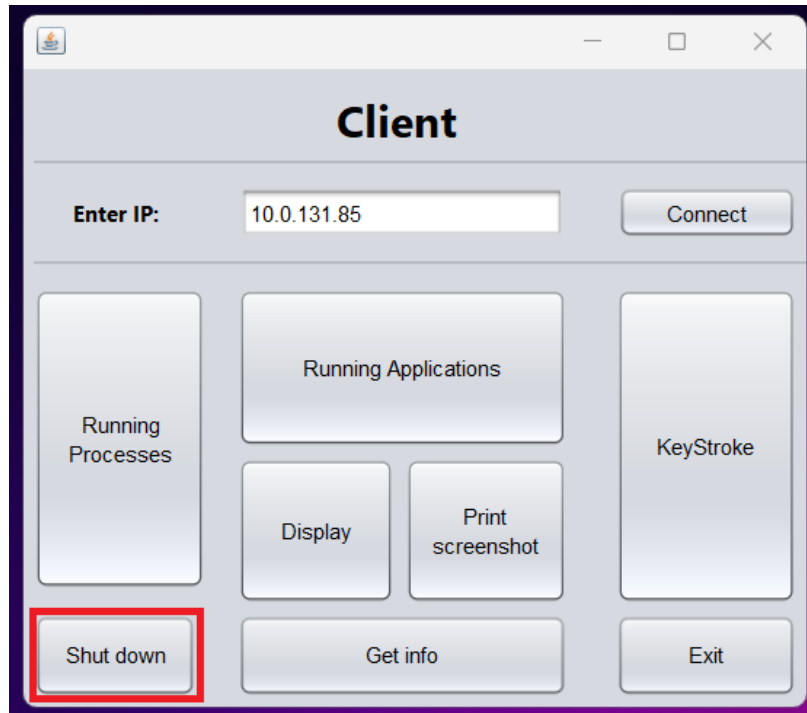


Thao tác Chụp và Lưu có thể thực hiện lặp lại nhiều lần.

- Để thoát khỏi chức năng, nhấp chọn **Exit** trên cửa sổ chức năng.

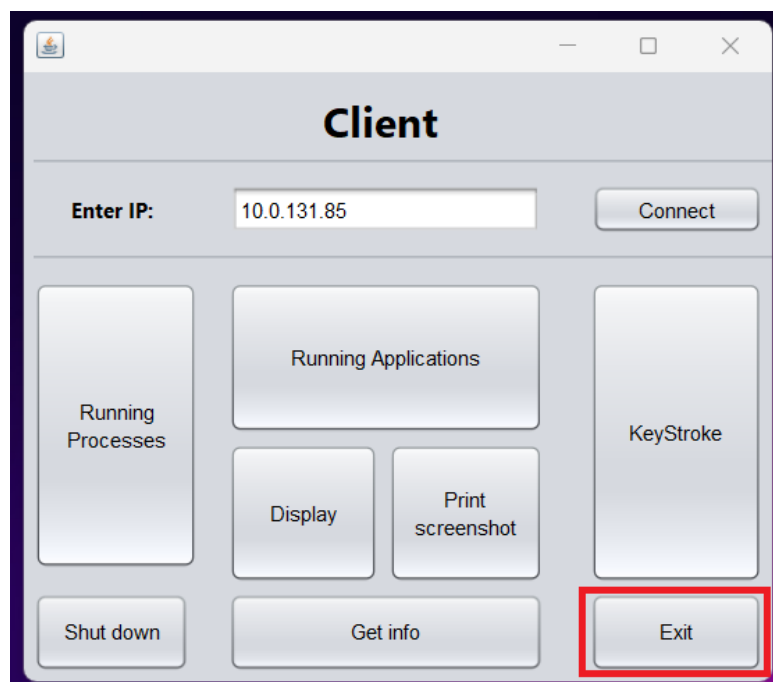
VIII. Shutdown

Trên cửa sổ giao diện của Client, nhấp chọn **Shutdown**. Server sẽ được shutdown sau vài giây.



IX. Exit

Để ngắt kết nối Server – Client và thoát chương trình điều khiển máy tính, nhấp chọn **Exit** trên cửa sổ giao diện của Client.



D. TÀI LIỆU THAM KHẢO

- Giáo trình Mạng máy tính (Mai Văn Cường, Trần Trung Dũng, Trần Hồng Ngọc, Lê Ngọc Sơn, Lê Giang Thanh, Trương Thị Mỹ Trang, Đào Anh Tuấn)
- <https://viettuts.vn/java-string/phuong-thuc-split>
- <https://www.anyviewer.com/how-to/kill-process-powershell-remote-computer-0007.html>
- <https://openplanning.net/13375/java-dataoutputstream>
- <https://www.javaguides.net/2018/08/datainputstream-class-in-java.html>
- <https://docs.oracle.com/javase/tutorial/uiswing/components/table.html>
- <https://stackhowto.com/how-to-set-column-width-in-jtable-in-java/>
- https://stackoverflow.com/questions/81902/how-to-find-and-kill-running-win-processes-from-within-java?fbclid=IwAR288f3St96cLR2nqV4QzX9JfJlrzEPDPwn-_Tfwk8OfDS6cJF1Jzs6t5FE
- https://stackoverflow.com/questions/9573696/kill-a-process-based-on-pid-in-java?fbclid=IwAR0q9ab17Q415A2iPVUcM7objLdX-i9dp8JIofBkCFV_wY4ezbRr1-rIBM
- <https://stackoverflow.com/questions/55469548/how-to-get-list-of-current-running-application-like-task-manager-except-applicat>
- <https://github.com/kwhat/jnativehook>
- <https://javadoc.io/static/com.1stleg/jnativehook/2.1.0/org/jnativehook/GlobalScreen.html>
- <https://stackoverflow.com/questions/4704073/how-can-i-get-system-hardware-info-via-java>
- <https://youtu.be/GMncZNs9Zrw>
- <https://stackoverflow.com/questions/22560566/export-jtable-in-excel-file>
- <https://stackoverflow.com/questions/3571203/what-are-runtime-getruntime-totalmemory-and-freememory>

E. HOẠT ĐỘNG NHÓM ĐỒ ÁN

I. Giới thiệu thành viên nhóm

Nhóm gồm 6 thành viên đến từ lớp 21CTT2

STT	Họ và tên	MSSV
1	Bùi Hồng Đăng	21120045
2	Trương Tấn Đạt	21120050
3	Nguyễn Trang Mai Hạnh	21120065
4	Nguyễn Trọng Hiếu – Nhóm trưởng	21120068
5	Tạ Công Hoàng	21120074
6	Lê Thanh Thái Quảng	21120123

II. Nguyên tắc hoạt động nhóm

- Các thành viên phải hoàn thành công việc đúng thời hạn đã giao.
- Các thành viên phải tham gia đúng giờ và đầy đủ các buổi họp (được phép trễ tối đa 15 phút với điều kiện phải thông báo trước cho cả nhóm biết).
- Các source code sau khi hoàn thành phải được up lên github chung của nhóm.
- Khi gặp khó khăn trong quá trình thực hiện công việc được giao, các thành viên chủ động liên lạc với nhau để cùng thảo luận.
- Các bảng phân công, kiểm tra và đánh giá tiến độ công việc đều được sự đồng ý và chấp nhận của tất cả các thành viên.

III. Kế hoạch làm việc nhóm

- *Trước ngày 22/10/2022*: các thành viên tự tìm hiểu đồ án.
- *Ngày 22/10/2022*: Họp nhóm lần 1.
- Giai đoạn 1 - *Từ ngày 22/10/2022 đến ngày 12/11/2022*: Thực hiện các phần:
 - + Connect server & client.
 - + Liệt kê các application đang chạy trong máy tính, start/stop 1 app.
 - + Liệt kê các processes đang chạy trong máy tính, start/stop 1 process.
 - + Get info (Lấy thông tin máy chủ).



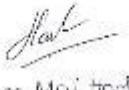

- + Keylogger.
- + Screen capture.
- + Shutdown.
- Ngày 12/11/2022: Họp nhóm lần 2. Kiểm tra công việc thực hiện ở giai đoạn 1.
- Giai đoạn 2 - Từ ngày 12/11/2022 đến ngày 30/11/2022: Thực hiện các phần:
 - + Display (Truyền màn hình).
 - + Save dữ liệu ra file .jpg, .xlsx, .docx
 - + Tổng hợp source code.
 - + Làm giao diện các phần khác Process, App và ghép code.
 - + Viết báo cáo.
 - + Quay video demo.
- Ngày 30/11/2022: Họp nhóm lần 3. Kiểm tra, chỉnh sửa lần cuối và hoàn thành đồ án.



IV. Đánh giá hoạt động nhóm

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

- Mức độ hoàn thành của cả đồ án: 100%
- Chi tiết: Hoàn thành, đáp ứng đủ yêu cầu của đề và bổ sung thêm những tính năng mới: Get Info (Lấy thông tin hệ điều hành và file system root của server); Save dữ liệu ra word, excel. Display (truyền màn hình)

2. Phân công công việc và đánh giá tỉ lệ đóng góp của từng thành viên

STT	Họ tên	MSSV	Nội dung	Tỉ lệ đóng góp	Chữ ký
1	Bùi Hồng Đăng	21120045	<ul style="list-style-type: none"> Liệt kê các applications đang chạy trong máy tính, start/stop app 	16%	 Bùi Hồng Đăng - 21120045 Đồng ý với tỉ lệ đóng góp 16%
2	Trương Tấn Đạt	21120050	<ul style="list-style-type: none"> Liệt kê các processes đang chạy trong máy tính, start/stop process Làm giao diện phần Process, App và ghép code 	17%	 Trương Tấn Đạt - 21120050 Chấp nhận và tỉ lệ đóng góp 17%
3	Nguyễn Trang Mai Hạnh	21120065	<ul style="list-style-type: none"> Connect server & client Viết báo cáo, biên bản mỗi cuộc họp 	16%	 Nguyễn Trang Mai Hạnh - 21120065 Đồng ý với tỉ lệ đóng góp 16%
4	Nguyễn Trọng Hiếu	21120068	<ul style="list-style-type: none"> Shut down, Get info (Lấy thông tin máy chủ) Display (Truyền màn hình) Save dữ liệu ra file .jpg, .xlsx, .docx Làm giao diện các phần khác Process, App và ghép code Quay video demo 	19%	 Nguyễn Trọng Hiếu - 21120068 Chấp nhận và tỉ lệ đóng góp 19%

5	Tạ Công Hoàng	21120074	<ul style="list-style-type: none"> Keylogger (Bắt phím nhấn) 	16%	 Tạ Công Hoàng - 21120074 Đồng ý với tỷ lệ đóng góp 16%
6	Lê Thanh Thái Quảng	21120123	<ul style="list-style-type: none"> Screen capture (Chụp màn hình và lưu ra file .jpg) 	16%	 Lê Thanh Thái - 21120123 Đồng ý với tỷ lệ đóng góp 16%

F. BIÊN BẢN HỌP NHÓM

S

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

BIÊN BẢN HỌP NHÓM LẦN 1**ĐỒ ÁN LÝ THUYẾT****VIẾT CHƯƠNG TRÌNH ĐIỀU KHIỂN MÁY TÍNH TỪ XA****❖ Thành viên tham gia:**

1. Bùi Hồng Đăng – 21120045
2. Trương Tấn Đạt – 21120050
3. Nguyễn Trang Mai Hạnh – 21120065
4. Nguyễn Trọng Hiếu – 21120068
5. Tạ Công Hoàng – 21120074
6. Lê Thanh Thái Quảng – 21120123

❖ Thời gian: lúc 21 giờ 00 phút, ngày 22 tháng 10 năm 2022**❖ Hình thức:** online qua Zoom**❖ Nội dung cuộc họp**

- Đọc và thảo luận đề bài
- Thảo luận định hướng làm đồ án
- Phân công công việc

❖ Bảng phân công

STT	Họ tên	MSSV	Nội dung
1	Bùi Hồng Đăng	21120045	<ul style="list-style-type: none"> Liệt kê các application đang chạy trong máy tính, start/stop 1 app
2	Trương Tấn Đạt	21120050	<ul style="list-style-type: none"> Liệt kê các processes đang chạy trong máy tính, start/stop 1 process
3	Nguyễn Trang Mai Hạnh	21120065	<ul style="list-style-type: none"> Connect server & client
4	Nguyễn Trọng Hiếu	21120068	<ul style="list-style-type: none"> Shutdown Get info (Lấy thông tin máy chủ)
5	Tạ Công Hoàng	21120074	<ul style="list-style-type: none"> Keylogger
6	Lê Thanh Thái Quảng	21120123	<ul style="list-style-type: none"> Screen capture

Các bảng phân công, kiểm tra và đánh giá tiến độ công việc đều được sự đồng ý và chấp nhận của tất cả các thành viên.

❖ Thời gian kiểm tra tiến độ: Thứ 7, ngày 12 tháng 11 năm 2022.

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

BIÊN BẢN HỌP NHÓM LẦN 2

ĐỒ ÁN LÝ THUYẾT

VIẾT CHƯƠNG TRÌNH ĐIỀU KHIỂN MÁY TÍNH TỪ XA

❖ **Thành viên tham gia:**

1. Bùi Hồng Đăng – 21120045
2. Trương Tấn Đạt – 21120050
3. Nguyễn Trang Mai Hạnh – 21120065
4. Nguyễn Trọng Hiếu – 21120068
5. Tạ Công Hoàng – 21120074
6. Lê Thanh Thái Quảng – 21120123

❖ **Thời gian:** lúc 21 giờ 00 phút, ngày 12 tháng 11 năm 2022

❖ **Hình thức:** online qua Zoom

❖ **Nội dung:**

Kiểm tra tiến độ và đánh giá mức độ hoàn thành công việc đã giao vào ngày 22/10/2022 của các thành viên.

- Thảo luận, góp ý kiến các vấn đề liên quan đến đồ án giữa các thành viên.
- Phân công các công việc tiếp theo của đồ án.

❖ Bảng kiểm tra và đánh giá mức độ hoàn thành công việc được giao vào ngày 22/10/2022

STT	Họ tên	MSSV	Nội dung	Kiểm tra và đánh giá tiến độ hoàn thành
1	Bùi Hồng Đăng	21120045	<ul style="list-style-type: none"> Liệt kê các application đang chạy trong máy tính, start/stop 1 app 	Hoàn thành đúng tiến độ.
2	Trương Tấn Đạt	21120050	<ul style="list-style-type: none"> Liệt kê các processes đang chạy trong máy tính, start/stop 1 process 	Hoàn thành đúng tiến độ.
3	Nguyễn Trang Mai Hạnh	21120065	<ul style="list-style-type: none"> Connect server & client 	Hoàn thành đúng tiến độ.
4	Nguyễn Trọng Hiếu	21120068	<ul style="list-style-type: none"> Shutdown Get info (Lấy thông tin máy chủ) 	Hoàn thành đúng tiến độ.
5	Tạ Công Hoàng	21120074	<ul style="list-style-type: none"> Keylogger 	Hoàn thành đúng tiến độ.
6	Lê Thanh Thái Quảng	21120123	<ul style="list-style-type: none"> Screen capture 	Hoàn thành đúng tiến độ.

❖ Bảng phân công công việc tiếp theo

STT	Họ tên	MSSV	Nội dung
1	Bùi Hồng Đăng	21120045	<ul style="list-style-type: none"> Tổng hợp source code
2	Trương Tấn Đạt	21120050	<ul style="list-style-type: none"> Làm giao diện các phần khác Process, App và ghép code
3	Nguyễn Trang Mai Hạnh	21120065	<ul style="list-style-type: none"> Viết báo cáo
4	Nguyễn Trọng Hiếu	21120068	<ul style="list-style-type: none"> Display (Truyền màn hình) Save dữ liệu ra file .jpg, .xlsx, .docx Làm giao diện các phần khác Process, App và ghép code Quay video demo
5	Tạ Công Hoàng	21120074	<ul style="list-style-type: none"> Tổng hợp source code
6	Lê Thanh Thái Quảng	21120123	<ul style="list-style-type: none"> Tổng hợp source code

Các bảng phân công, kiểm tra và đánh giá tiến độ công việc đều được sự đồng ý và chấp nhận của tất cả các thành viên.

❖ Thời gian các thành viên cần hoàn thành công việc được giao:

Thứ 7 ngày 30 tháng 11 năm 2022

CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM

Độc lập – Tự do – Hạnh phúc

BIÊN BẢN HỌP NHÓM LẦN 3

ĐỒ ÁN LÝ THUYẾT

VIẾT CHƯƠNG TRÌNH ĐIỀU KHIỂN MÁY TÍNH TỪ XA

❖ **Thành viên tham gia:**

1. Bùi Hồng Đăng – 21120045
2. Trương Tấn Đạt – 21120050
3. Nguyễn Trang Mai Hạnh – 21120065
4. Nguyễn Trọng Hiếu – 21120068
5. Tạ Công Hoàng – 21120074
6. Lê Thanh Thái Quảng – 21120123

❖ **Thời gian:** lúc 19 giờ 00 phút, ngày 30 tháng 11 năm 2022

❖ **Hình thức:** Trực tiếp.

❖ **Nội dung:**

- Chạy thử chương trình, kiểm tra lỗi.
- Các thành viên góp ý, chỉnh sửa video demo và bản báo cáo.
- Các thành viên cho ý kiến và đánh giá mức độ hoàn thành các công việc trong cả quá trình thực hiện đồ án.

❖ **Các file source code, file word báo cáo đồ án, video demo sẽ do bạn**

Nguyễn Trọng Hiếu - MSSV 21120068 đại diện nhóm nộp cho giảng viên.

❖ **Đánh giá mức độ hoàn thành công việc trong cả quá trình thực hiện đồ án của từng thành viên**

STT	Họ tên	MSSV	Nội dung	Đánh giá mức độ hoàn thành
1	Bùi Hồng Đăng	21120045	<ul style="list-style-type: none"> Liệt kê các applications đang chạy trong máy tính, start/stop app 	Hoàn thành tốt.
2	Trương Tấn Đạt	21120050	<ul style="list-style-type: none"> Liệt kê các processes đang chạy trong máy tính, start/stop process Làm giao diện phần Process, App và ghép code 	Hoàn thành tốt.
3	Nguyễn Trang Mai Hạnh	21120065	<ul style="list-style-type: none"> Connect server & client Viết báo cáo, biên bản mỗi cuộc họp 	Hoàn thành tốt.
4	Nguyễn Trọng Hiếu	21120068	<ul style="list-style-type: none"> Shut down, Get info (Lấy thông tin máy chủ) Display (Truyền màn hình) Save dữ liệu ra file .jpg, .xlsx, .docx Làm giao diện các phần khác Process, App và ghép code Quay video demo 	Hoàn thành tốt.
5	Tạ Công Hoàng	21120074	<ul style="list-style-type: none"> Keylogger (Bắt phím nhấn) 	Hoàn thành tốt.
6	Lê Thanh Thái Quảng	21120123	<ul style="list-style-type: none"> Screen capture (Chụp màn hình và lưu ra file .jpg) 	Hoàn thành tốt.