

TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI
TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
BỘ MÔN TRUYỀN THÔNG VÀ MẠNG MÁY TÍNH

-----oOo-----



BÁO CÁO BÀI TẬP LỚN

XÂY DỰNG ỨNG DỤNG CHAT

GVHD: *ThS. Bùi Trọng Tùng*

SVTH: *Đào Đăng Sơn 20194362*

Đặng Phương Nam 20194335

TP. HÀ NỘI, THÁNG 2 NĂM 2022

Giới thiệu đề tài	3
Tổng quan về đề tài	3
Phân chia công việc	3
Thiết kế giao thức	4
Thiết kế chương trình	8
Client	8
Cấu trúc thư mục (packages)	8
Biểu đồ class	10
Lưu đồ	10
Tương tác giữa các thành phần của UI và ClientConnection (quản lý nhận và gửi thông điệp)	11
Server	13
Cấu trúc thư mục:	13
Cơ chế hoạt động của Server:	13
Lưu đồ hoạt động	14
Cách thức biên dịch, cài đặt chương trình	16
Client	16
Server	21
2.1 Cài đặt môi trường thực thi (Visual studio 2015):	21
2.2 Biên dịch và chạy chương trình	21
V. Tài liệu tham khảo:	23

I. Giới thiệu đề tài

1. Tổng quan về đề tài

Ứng dụng chat đơn giản sử dụng Winsock và TCP socket ở phía server xử lý việc trao đổi thông điệp và Javafx và thư viện socket của java để tạo giao diện người dùng ở phía client.

Ứng dụng có các chức năng sau:

- Khi 1 user đăng nhập thành công, cung cấp danh sách các user khác đang online
- User gửi và nhận thông điệp
- Khi có 1 trong 2 user ngắt kết nối hoặc yêu cầu dừng cuộc trò chuyện, thông báo cho bên còn lại biết.
- Cho phép user chuyển sang cuộc trò chuyện khác (với user khác nếu cần).
- Tạo nhóm chat:
 - User tạo nhóm chat
 - User thêm các user khác vào nhóm
 - Tham gia và rời nhóm chat
 - Gửi thông điệp tới các nhóm

2. Phân chia công việc

Chức năng	Người đảm nhiệm	Đánh giá mức độ hoàn thành	Chú thích
Giao diện người dùng	Đặng Phương Nam	100%	
Nhận và gửi thông điệp phía client	Đặng Phương Nam	100%	
Xử lý các thông điệp phía Client	Đặng Phương Nam	100%	
Xử lý nhận, trả lời và chuyển tiếp các thông điệp	Đào Đăng Sơn	100%	

Xử lý truyền dòng	Đào Đăng Sơn	100%	
Quản lý cấu trúc dữ liệu chứa các thông tin người dùng, nhóm	Đào Đăng Sơn	100%	
Xác thực tài khoản/mật khẩu	Đào Đăng Sơn	100%	
Server không giới hạn số lượng người dùng (đa luồng)	Đào Đăng Sơn, Đặng Phương Nam	100%	
Viết báo cáo phần thiết kế chương trình và các thức biên dịch, cài đặt của Server	Đào Đăng Sơn	100%	
Viết báo cáo phần thiết kế chương trình và các thức biên dịch, cài đặt của Client	Đặng Phương Nam	100%	
Viết báo cáo phần Giới thiệu đề tài và Thiết kế giao thức	Đặng Phương Nam, Đào Đăng Sơn	100%	

II. Thiết kế giao thức

Ghi chú:

- Ký hiệu ‘|1|’: “\r\n”- Delimiter mức 1: báo kết thúc 1 thông điệp
- Ký hiệu ‘|2|’: ký tự ASCII(6) - delimiter mức 2
- Ký hiệu ‘|3|’: ký tự ASCII(7) - delimiter mức 3

Sự kiện	Cấu trúc thông điệp yêu cầu từ Client	Cấu trúc thông điệp trả lời từ Server	Chú thích
---------	---------------------------------------	---------------------------------------	-----------

Client login	USER 2 username 2 password 1	USER 2 MATRALOI 1	
Client lấy danh sách người dùng	LISTUSER 2 FROM 1	LISTUSER 2 USER1 2 USER2 2 ...	Server gửi thông điệp cập nhật lại danh sách người dùng online hiện tại
Client lấy danh sách nhóm	LISTGROUP 2 FROM 1	LISTGROUP 2 GROUPNAME1 3 GROUPUSER11 3 GROUPUSER12 2 GROUPNAME2 3 GROUPUSER21 3 GROUPUSER22 3 GROUPUSER23 2 GROUPNAME3 3 1	Server gửi thông điệp chứa danh sách các nhóm hiện tại
Client gửi thông điệp đến 1 client khác	POST 2 FROM 2 TO 2 PAYLOAD 1	POST 2 MATRALOI 1	Server chuyển tiếp tin nhắn đến người nhận, sau đó gửi thông điệp phản hồi tới người gửi Thông điệp chuyển tiếp: MESSAGEUSER 2 FROM 2 PAYLOAD
Client tạo nhóm	GROUP 2 NAME 2 FROM 2 USER1 2 ... 1	GROUP 2 MATRALOI 1 LISTGROUP 2 GROUPNAME1 3 GROUPUSER11 3 GROUPUSER12 2 GROUPNAME2 3	Server gửi thông điệp cập nhật (LISTGROUP) đến tất cả các thành viên trong nhóm

		GROUPUSER21 3 GROUPUSER22 3 GROUPUSER23 2 GROUPNAME3 3 1	
Client gửi thông điệp đến nhóm	BROADCAST 2 FROM 2 GROUPNAME 2 P AYLOAD 1		Server chuyển tiếp tin nhắn đến các thành viên khác trong nhóm: Thông điệp chuyển tiếp: MESSAGEGROUP 2 NAME 2 FROM 2 PAYLOAD
Client rời nhóm	LEAVE 2 FROM 2 NAME 1	LEAVE 2 MATRALOI 2 NHOMCANROI LISTGROUP 2 GROUPNAME1 3 GROUPUSER11 3 GROUPUSER12 2 GROUPNAME2 3 GROUPUSER21 3 GROUPUSER22 3 GROUPUSER23 2 GROUPNAME3 3 1	Server gửi lại thông điệp cập nhật (LISTGROUP) đến tất cả các thành viên còn lại trong nhóm
Client đăng xuất / quit	QUIT 2 FROM 1	QUIT 2 MATRALOI 1	

Mời vào nhóm	ADD 2 FROM 2 GROUPNAME 2 NAME1 2 NAME2 2 ... 1	ADD 2 MATRALOI 2 NAME 1 2 NAME2.... 1 LISTGROUP 2 GROUPNAME1 3 GROUPUSER11 3 GROUPUSER12 2 GROUPNAME2 3 GROUPUSER21 3 GROUPUSER22 3 GROUPUSER23 2 GROUPNAME3 3 	Thông điệp trả lời sẽ thông báo cho người dùng đã add thành công những ai, đồng thời Server sẽ gửi lại thông điệp cập nhật (LISTGROUP) đến các thành viên khác trong nhóm.
--------------	---	---	--

2. Các loại mã trả lời

Chú thích mã trả lời	Mã trả lời	Ghi chú
USER login thành công	10	
USER login thất bại	11	Sai tên tài khoản/mật khẩu
USER login thất bại	12	Server không đọc được file account.txt
USER login thất bại	13	tài khoản đã đăng nhập ở nơi khác rồi

POST thành công	20	
POST thất bại	21	Người dùng phía còn lại không online
GROUP tạo thành công	30	
ADD thành công	40	Trong trường hợp người dùng được add đã có trong group, server sẽ bỏ qua và coi như việc add thành công
QUIT thành công	50	
GROUP tạo thất bại	31	Tên group đã tồn tại
LEAVE success	60	Thông điệp này sẽ được server chủ động gửi đến client cuối cùng trong group khi trong group chỉ còn 1 thành viên (ép xóa group)

III. Thiết kế chương trình

1. Client

a) Cấu trúc thư mục (packages)

```

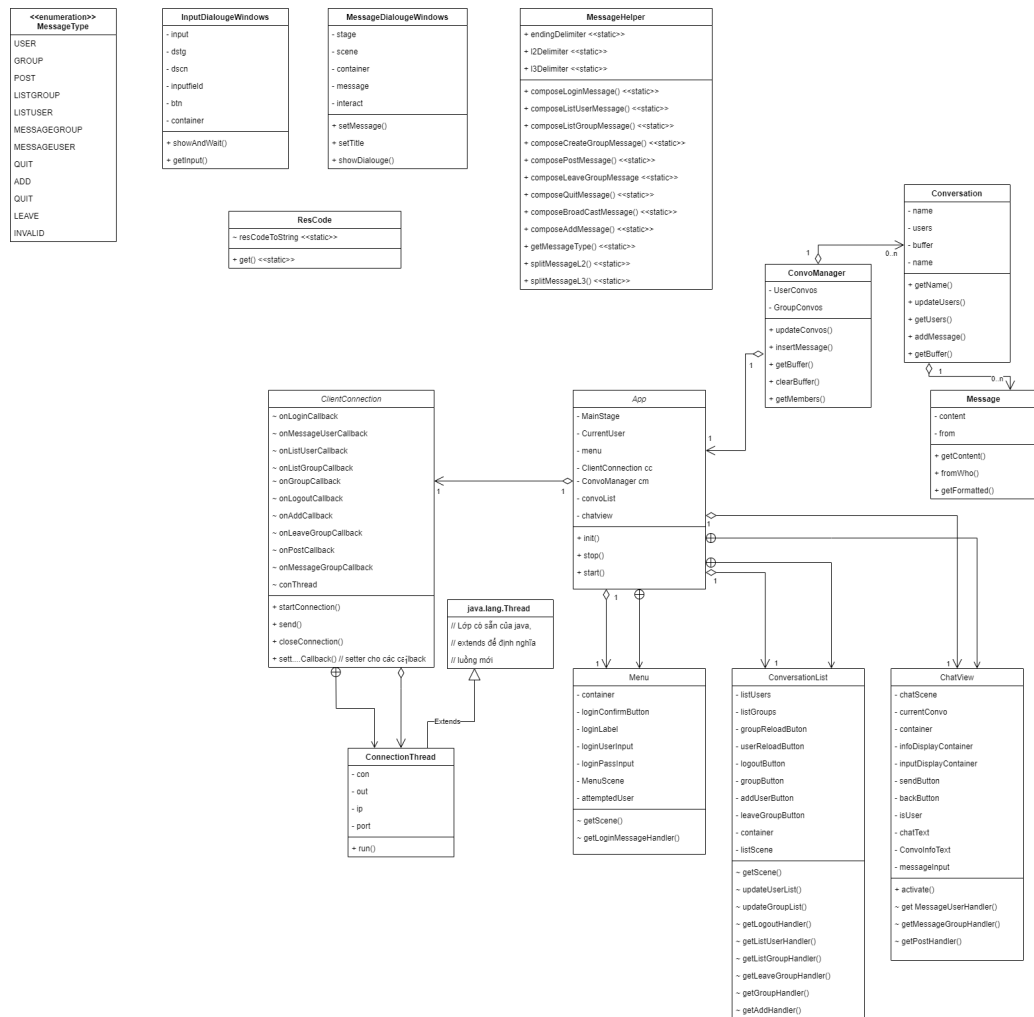
-----com.newclient      +
|
|      +-----models      +
|      |      |
|      |      +-- Conversation.java
|      |      +-- Message.java
|      |      +-- MessageType.java
|      |
|      +-----utils      +
|      |      |
|      |      +--ConvoManager.java
|      |      +--InputDialogue.java
|      |      +--MessageDialougeWindows.java

```



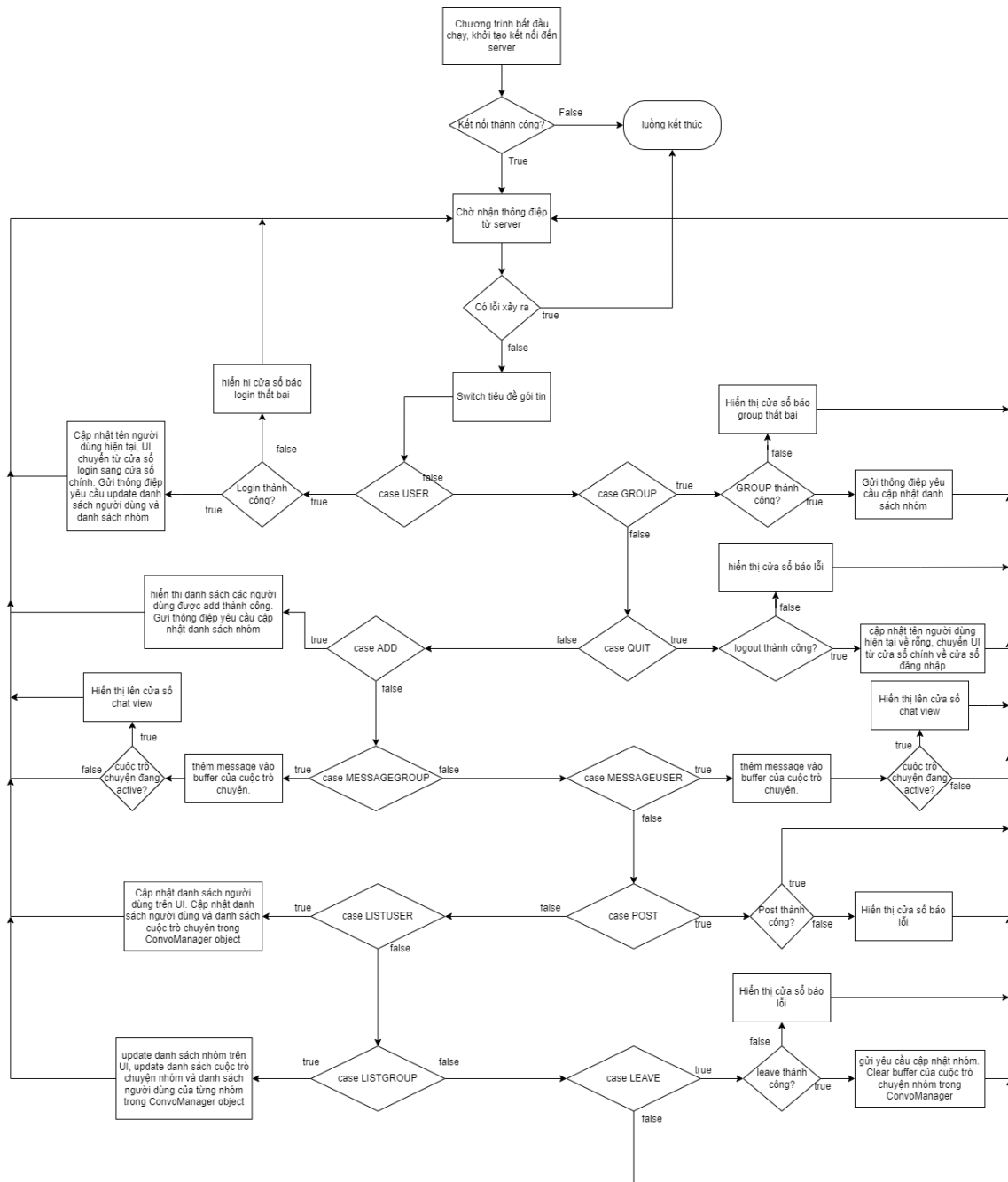
```
|          +---MessageHelper.java
|          +---ResCode.java
|
+-----App.java
+-----ClientConnection.java
```

b) Biểu đồ class

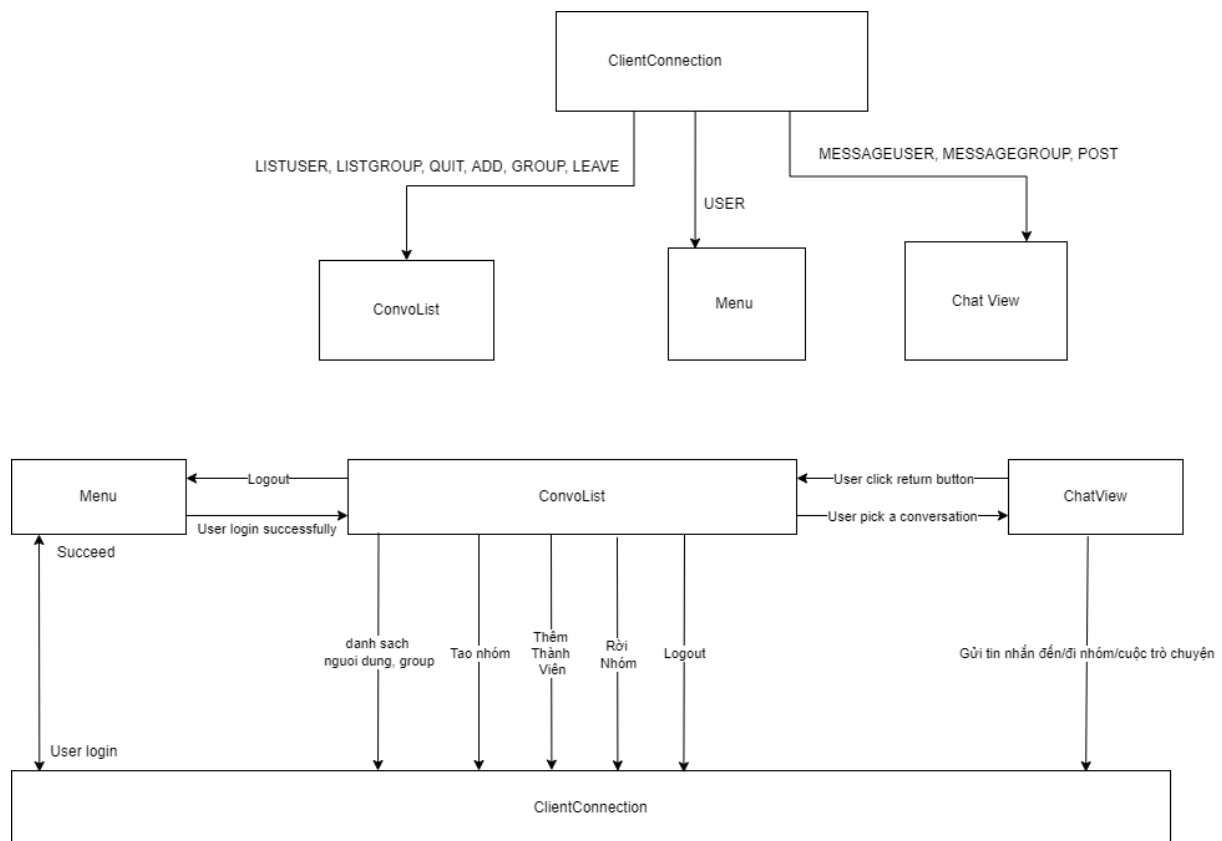


c) Lưu đồ

- Lưu đồ của xử lý các thông điệp được gửi từ server:



d) Tương tác giữa các thành phần của UI và ClientConnection (quản lý nhận và gửi thông điệp)



- Khi client nhận được thông điệp từ server, nó sẽ gọi các callback tương ứng với loại thông điệp mà nó nhận được, thông qua 1 object Consumer nằm trong context của các thành phần của UI tương ứng, từ đó nó có thể truy cập đến các biến, các giá trị, các thành phần trong UI để cập nhật theo nhu cầu của chương trình.
- Danh sách các Consumer object và UI class tương ứng:
 - ListUserCallback, ListGroupCallback, QuitCallback (LogoutCallback), LeaveGroupCallback, AddUserCallback, GroupCallback nằm trong UI class ConvoList
 - UserCallback (LoginCallback) nằm trong UI class Menu
 - MessageUserCallback, MessageGroupCallback, PostCallback nằm trong UI class ChatView
- Khi người dùng thao tác trên các thành phần của UI liên quan đến các chức năng thì các EventHandler sẽ gọi đến hàm send() của ClientConnection để gửi thông điệp tương ứng đến server (quá trình gửi vẫn xảy ra ở luồng chính). Client khi gửi thì chỉ quan tâm đến việc gửi thông điệp đi, không quan tâm đến thông điệp trả lời). Khi có trả lời từ server thì thông điệp sẽ được nhận và xử lý

ở phía luồng nhận, dữ liệu sẽ được cập nhật tức thời còn UI thì sẽ được scheduled để cập nhật sớm nhất có thể.

2. Server

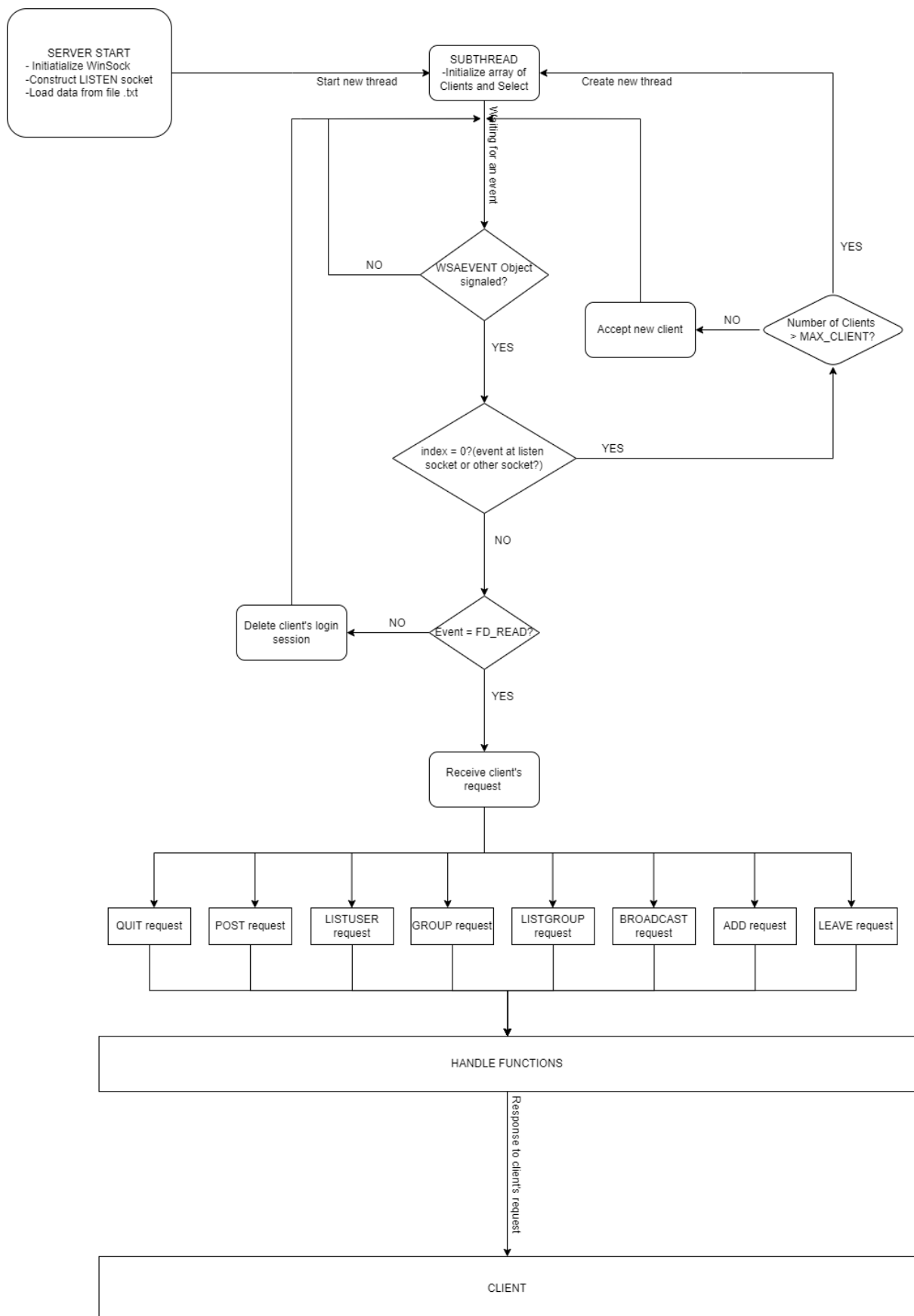
a) Cấu trúc thư mục:

- File dữ liệu: account.txt
- File header:
 - +) Protocol.h: Định nghĩa giao thức
 - +) MessageHandling.h: Định nghĩa các hàm xử lý, hỗ trợ thao tác
 - +) LoginFunction.h: Định nghĩa các hàm xử lý đăng nhập, đăng xuất
 - +) Communication.h: Định nghĩa các hàm xử lý các request bên client
- File cpp:
 - +) MessageHandling.cpp: Triển khai file MessageHandling.h
 - +) LoginFunction.cpp: Triển khai file LoginFunction.h
 - +) Communication.cpp: Triển khai file Communication.h
 - +) Server.cpp: File main của Server

b) Cơ chế hoạt động của Server:

- Server hoạt động dựa theo 2 chế độ vào ra EventSelect và Multi-thread
- Server thực hiện bắt đầu với một luồng con, mỗi luồng con xử lý được tối đa 64 clients cùng lúc, khi số lượng clients vượt quá mức tối đa, Server sẽ tạo thêm luồng mới và đưa kết nối mới vừa nhận được xử lý ở luồng mới
- Tại mỗi luồng con, Server hoạt động theo cơ chế **EventSelect**. Server khởi tạo 1 mảng quản lý thông tin Client(Socket,Address) và 1 mảng các **WSAEVENT** được gán tương ứng với các client. Các **WSAEVENT** được thiết lập lắng nghe các sự kiện đến (**FD_ACCEPT**, **FD_CLOSE** với listen socket; **FD_READ** và **FD_CLOSE** với các socket khác). Server được đặt vào trạng thái lắng nghe sự kiện với hàm **WSAWaitForMultipleEvents** được đặt trong vòng lặp vô tận. Hàm này sẽ trả về index tương ứng với vị trí của **WSAEVENT** xảy ra sự kiện đến. Server kiểm tra xem đó là sự kiện đến nào(accept,read,close) và xử lý tương ứng, sau đó trả kết quả lại cho bên client.

c) Lưu đồ hoạt động



IV. Cách thức biên dịch, cài đặt chương trình

1. Client

Đường dẫn đến github của client

<https://github.com/DustinDust/Chatapp-client-networkprogramming>


Client của project sử dụng javafx với maven là công cụ quản lý phần mềm. Để chạy được project, ta đầu tiên cần phải cài đặt maven.

a) Cài đặt môi trường Maven

- Link download maven: <https://maven.apache.org/download.cgi>. Ta sẽ lựa chọn file binaries - zip archive:

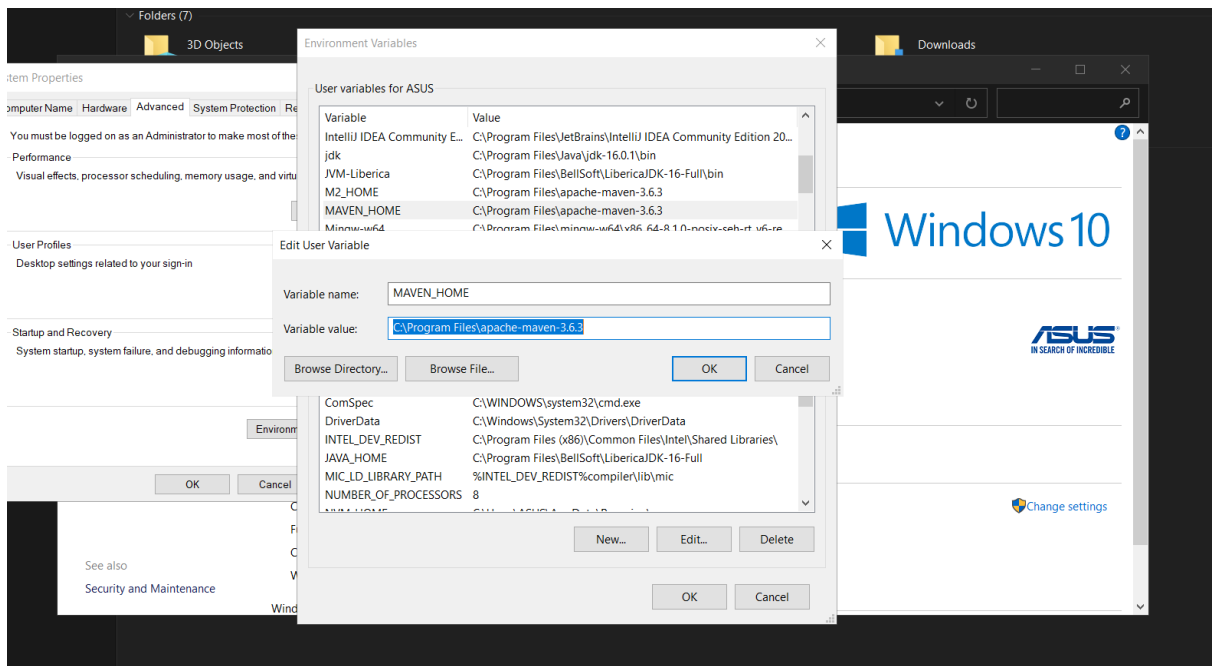
	Link	Checksums	Signature
Binary tar.gz archive	apache-maven-3.8.4-bin.tar.gz	apache-maven-3.8.4-bin.tar.gz.sha512	apache-maven-3.8.4-bin.tar.gz.asc
Binary zip archive	apache-maven-3.8.4-bin.zip	apache-maven-3.8.4-bin.zip.sha512	apache-maven-3.8.4-bin.zip.asc
Source tar.gz archive	apache-maven-3.8.4-src.tar.gz	apache-maven-3.8.4-src.tar.gz.sha512	apache-maven-3.8.4-src.tar.gz.asc
Source zip archive	apache-maven-3.8.4-src.zip	apache-maven-3.8.4-src.zip.sha512	apache-maven-3.8.4-src.zip.asc

- Sau khi giải nén, ta được thư mục như sau:

Name	Date modified	Type
 apache-maven-3.8.4	11/14/2021 9:12 AM	File folder

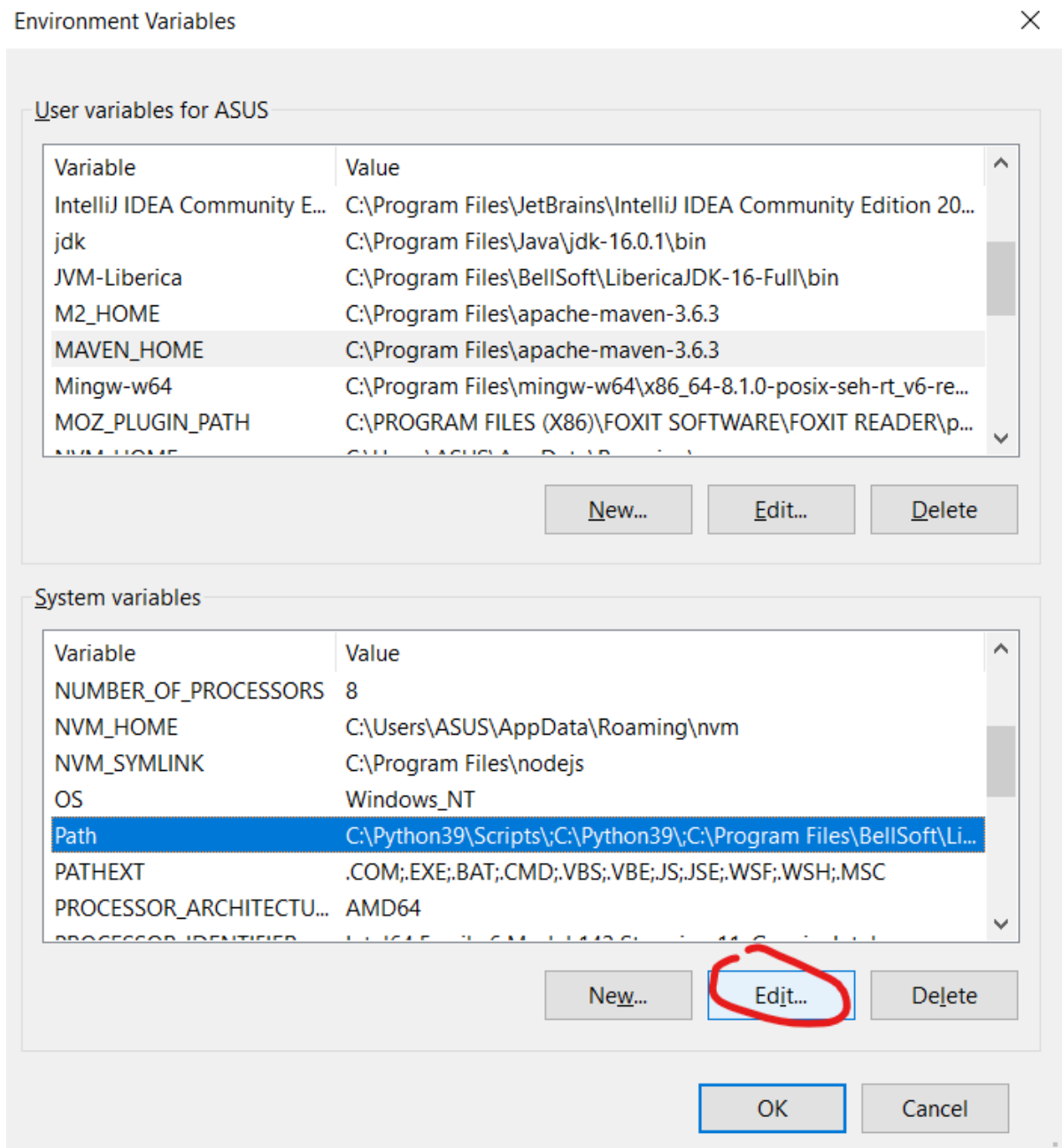
- Thêm MAVEN_HOME và biến môi trường cho windows: chuột phải on **MyComputer** -> **properties** -> **Advanced System Settings** ->

Environment variables -> click new button:

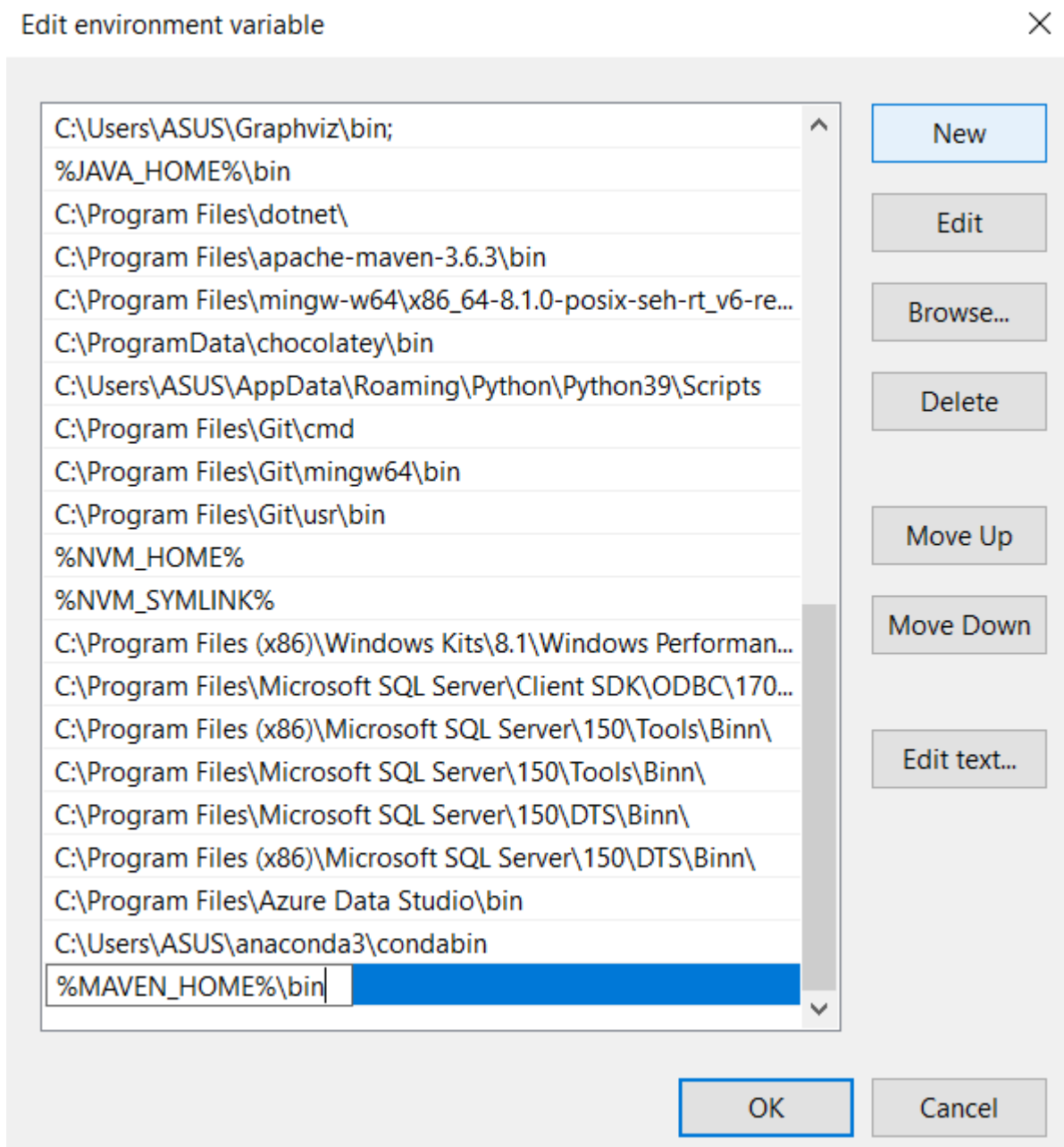


- Điền variable name là MAVEN_HOME và variable value là đường dẫn đến thư mục maven vừa được giải nén ra.

- Bổ sung maven vào path: ở phía dưới phần system variables, ta tìm variable có tên là Path, chọn edit:



- Chọn new, và thêm đường dẫn %MAVEN_HOME%\bin rồi xác nhận



- Kiểm tra maven: trên cửa sổ command prompt, gõ lệnh `mvn -version`

```
Command Prompt
Microsoft Windows [Version 10.0.19043.1526]
(c) Microsoft Corporation. All rights reserved.

C:\Users\ASUS>mvn -version
Apache Maven 3.8.1 (05c21c65bdfed0f71a2f2ada8b84da59348c4c5d)
Maven home: C:\Program Files\apache-maven-3.6.3\bin\..
Java version: 16, vendor: BellSoft, runtime: C:\Program Files\BellSoft\LibericaJDK-16-Full
Default locale: en_US, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"

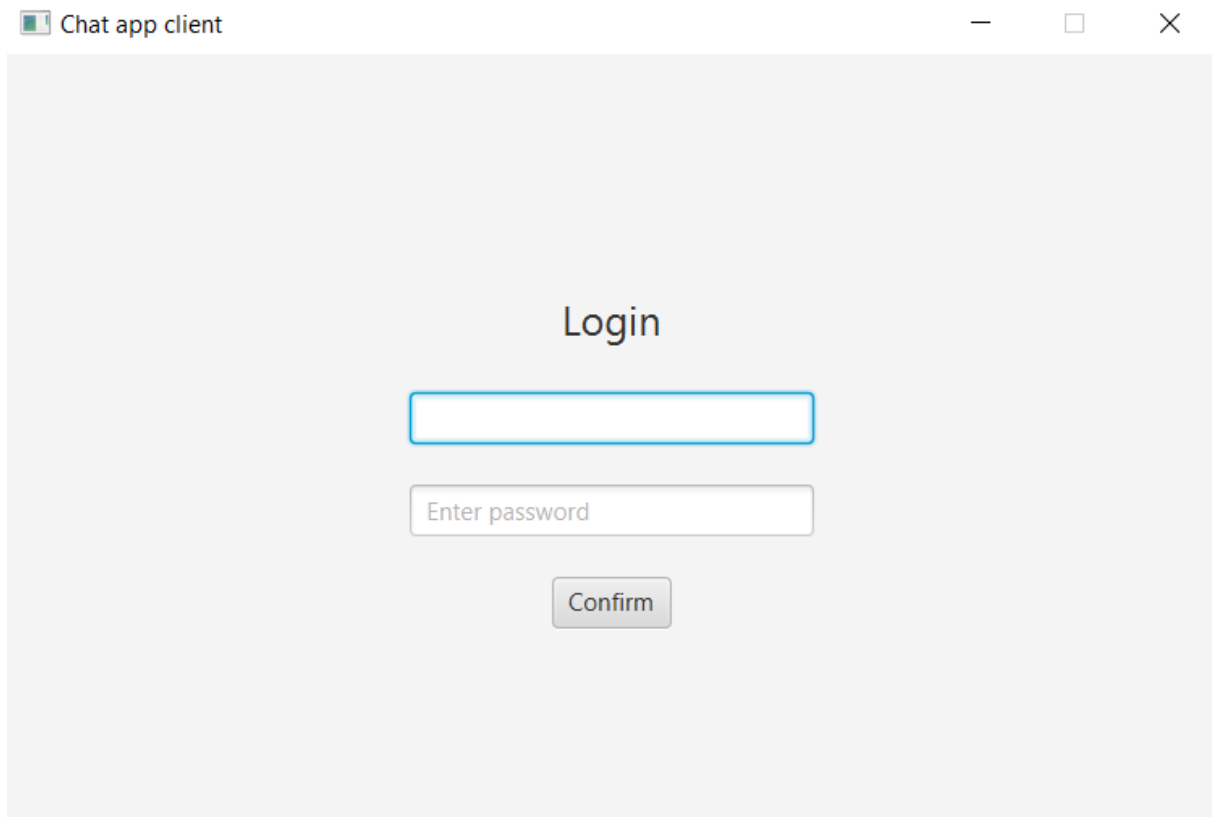
C:\Users\ASUS>
```

b) Tải và chạy project client

- Clone mã nguồn từ github: `git clone`
<https://github.com/DustinDust/Chatapp-client-networkprogramming.git>
- **Note:** cần đảm bảo trong thư mục mã nguồn có chứa file pom.xml
- Đọc, sửa code trong mã nguồn rồi chạy javafx app bằng cách chọn từ cửa sổ maven plugin của IDE tùy ý (Eclipse/IntelliJ/VSCode); hoặc có thể chạy từ command prompt (từ thư mục chứa pom.xml) với câu lệnh `mvn javafx:run`

```
Command Prompt - mvn javafx:run
C:\Users\ASUS\Vscode-projects\java-related>cd test
C:\Users\ASUS\Vscode-projects\java-related\test>ls
Chatapp-client-networkprogramming
C:\Users\ASUS\Vscode-projects\java-related\test>cd Chatapp-client-networkprogramming
C:\Users\ASUS\Vscode-projects\java-related\test\Chatapp-client-networkprogramming>ls
README.md pom.xml src target usecase.md
C:\Users\ASUS\Vscode-projects\java-related\test\Chatapp-client-networkprogramming>mvn javafx:run
[INFO] Scanning for projects...
[INFO] -----< com.newclient:demo >-----
[INFO] Building demo 1.0-SNAPSHOT
[INFO] -----[ jar ]-----
[INFO]
[INFO] >>> javafx-maven-plugin:0.0.6:run (default-cli) > process-classes @ demo >>>
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ demo ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory C:\Users\ASUS\Vscode-projects\java-related\test\Chatapp-client-networkprogram
ming\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:compile (default-compile) @ demo ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] <<< javafx-maven-plugin:0.0.6:run (default-cli) < process-classes @ demo <<<
[INFO]
```

- Kết quả (server cần phải chạy trước khi chạy client để có thể thực hiện các chức năng):



- **Note:** trong trường hợp có vấn đề về dependencies resolve, ta có thể ép maven cập nhật dependencies thông qua lệnh command prompt: `mvn dependency:resolve -U` (tại thư mục chứa pom.xml)

2. Server

2.1 Cài đặt môi trường thực thi (Visual studio 2015):

- Link tải file cài đặt:
https://my.visualstudio.com/Downloads?q=visual%20studio%202015&wt.mc_id=o~msft~vscom~older-downloads
- Truy cập vào link trên, sau đó chọn phiên bản Visual Studio Community 2015 để tải về
- Chạy file cài đặt, chọn các tính năng cần thiết cho VS 2015 sau đó cài đặt.

2.2 Biên dịch và chạy chương trình

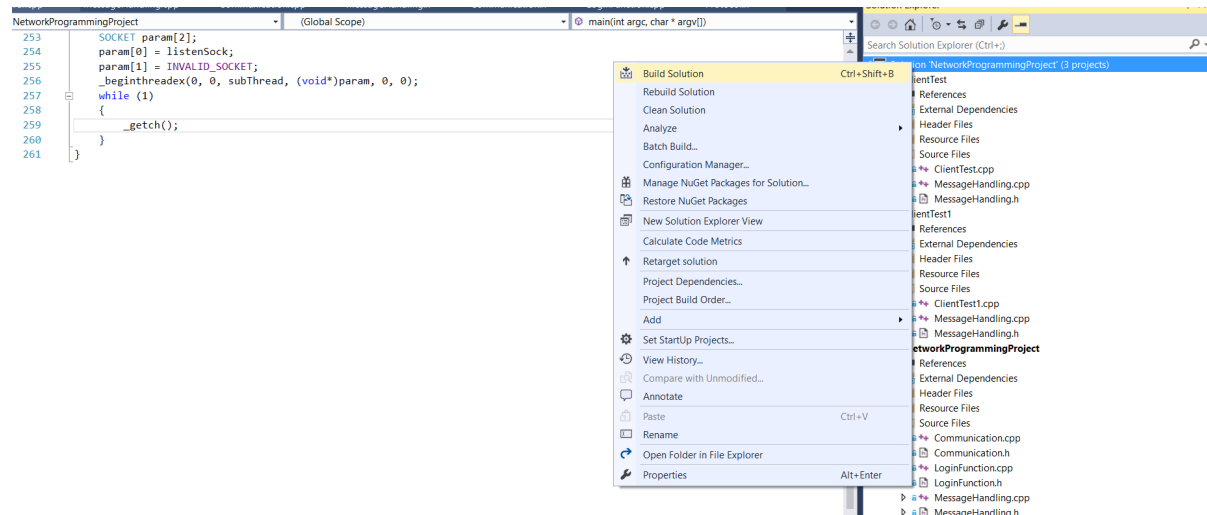
- Clone project từ link Github:
<https://github.com/lazyBoy1909/ChatApp.git>

- Chạy file **Microsoft Visual Studio Solution(.sln)** để bắt đầu project với Visual Studio 2015

name	Date modified	type	Size
ClientTest	1/11/2022 3:24 PM	File folder	
ClientTest1	12/28/2021 7:26 PM	File folder	
Debug	2/8/2022 4:52 PM	File folder	
NetworkProgrammingProject	2/17/2022 9:23 PM	File folder	
NetworkProgrammingProject	2/15/2022 5:54 PM	Microsoft Visual St...	3 KB
NetworkProgrammingProject.VC		base File	30,536 KB

Size: 2.81 KB
Type: Microsoft Visual Studio Solution
Version: Visual Studio 14
Date modified: 2/15/2022 5:54 PM

- Chuột phải vào tên Solution, chọn Build Solution để thực hiện build project



- Chạy file **NetworkProgrammingProject.exe** trong thư mục Debug hoặc chạy trên Command Prompt

account	2/8/2022 5:00 PM	Text Document	1 KB
ClientTest	2/15/2022 5:18 PM	Application	127 KB
ClientTest	2/15/2022 5:18 PM	Incremental Linker...	644 KB
ClientTest	2/15/2022 5:18 PM	Program Debug D...	748 KB
ClientTest	2/15/2022 5:18 PM	Application	127 KB
ClientTest1	2/15/2022 5:18 PM	Incremental Linker...	644 KB
ClientTest1	2/15/2022 5:18 PM	Program Debug D...	748 KB
NetworkProgrammingProject	2/15/2022 5:54 PM	Application	235 KB
NetworkProgrammingProject	2/15/2022 5:54 PM	Incremental Linker...	3,157 KB
NetworkProgrammingProject	2/15/2022 5:54 PM	Program Debug D...	3,684 KB

V. Tài liệu tham khảo:

- Microsoft Document about WSAEventSelect:
<https://docs.microsoft.com/en-us/windows/win32/api/winsock2/nf-winsock2-wsaeventselect>
- Microsoft Document about Error Codes:
<https://docs.microsoft.com/en-us/windows/win32/winsock/windows-sockets-error-codes-2>
- Example source code using WSAEventSelect:
<https://users.soict.hust.edu.vn/tungbt/it4060/src/WSAEventSelectServer.cpp>
- Javafx documentation: <https://openjfx.io/javadoc/17/>
- Javafx - showAndWait():
<https://stackoverflow.com/questions/65362590/what-is-the-difference-between-showandwait-and-show-in-java-fx-stage-class>
- Javafx - Platform.runLater() and task:
<https://stackoverflow.com/questions/13784333/platform-runlater-and-task-in-javafx>
- Maven lifecycle documentation:
<https://maven.apache.org/guides/introduction/introduction-to-the-lifecycle.html>
- Maven and Javafx for starter: <https://openjfx.io/openjfx-docs/maven>
- Getting started with java socket and stream:
<https://www.baeldung.com/a-guide-to-java-sockets>
- Java socket documentation:
<https://docs.oracle.com/javase/7/docs/api/java/net/Socket.html>