

**TRƯỜNG ĐẠI HỌC BÁCH KHOA TP.HCM**  
**KHOA ĐIỆN - ĐIỆN TỬ**



---

**BÁO CÁO BÀI TẬP LỚN**  
**MẠNG MÁY TÍNH**

---

**ĐỀ TÀI: NET – CHATTING**  
**GVHD: TS VÕ QUẾ SƠN**

**Sinh viên thực hiện:**

Võ Minh Hưng	1511419
Nguyễn Văn Quý	1512702
Nguyễn Văn Quảng	1512670
Nguyễn Tấn Sỹ	1512872
Nguyễn Quang Vinh	1514066
Lê Phú Nhân	1512256

TP.Hồ Chí Minh 5/2018

## Mục lục

<b>1. Giới thiệu các bước cần thiết để hoàn thành bài tập lớn:</b>	<b>1</b>
1.1. Socket Programming . . . . .	1
1.2. Đa luồng (Multi-thread): . . . . .	2
1.3. MySQL — Cơ sở dữ liệu . . . . .	2
1.4. GUI — Giao diện . . . . .	4
1.5. NAT Port . . . . .	6
<b>2. Mô hình hệ thống, thuật toán, chương trình bài tập lớn:</b>	<b>8</b>
2.1. Mô hình hệ thống: . . . . .	8
2.2. Sơ đồ thuật toán server: . . . . .	8
2.3. Sơ đồ thuật toán client: . . . . .	9
<b>3. Tổng kết, demo</b>	<b>10</b>
3.1. Public IP của server: . . . . .	10
3.2. Mở server: . . . . .	11
3.3. Mở kết nối từ client đến server. . . . .	11
3.4. Chat box: . . . . .	13
3.5. Tổng kết: . . . . .	15

## Danh sách hình vẽ

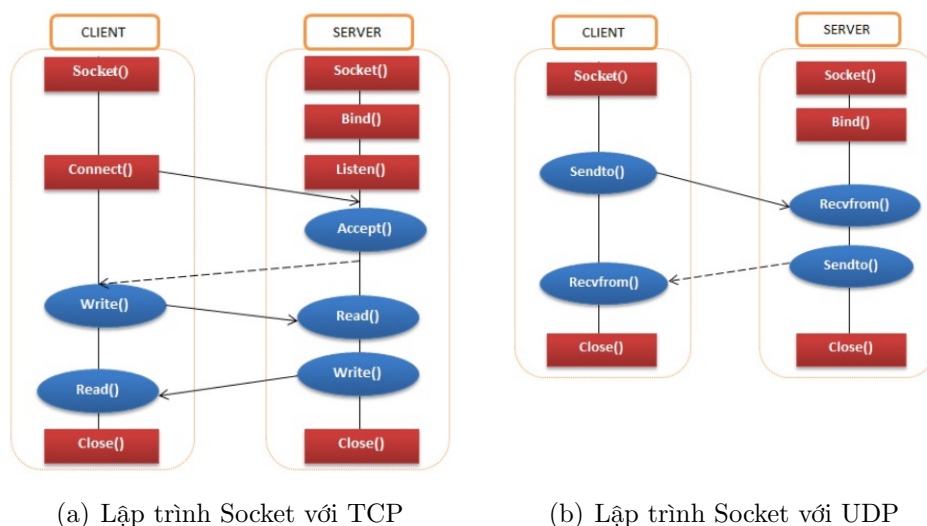
1. Cơ chế gọi hàm trong lập trình Socket . . . . .	1
2. Tạo một GUI mới . . . . .	4
3. Thuộc tính Object . . . . .	5
4. Kết quả GUI . . . . .	5
5. Hiện IP khi dùng lệnh ifconfig. . . . .	6
6. Bảng đăng nhập của router. . . . .	6
7. Thanh “Navigation” và "sidebar" của router web. . . . .	7
8. Điền thông tin NAT PORT . . . . .	7
9. Kết quả . . . . .	7
10. Thuật toán Sever . . . . .	9
11. Thuật toán Client . . . . .	10
12. Public IP được lấy bằng Code Python . . . . .	11
13. Thông báo trên server . . . . .	11
14. Sau khi nhập thành công, hiển thị trên client. . . . .	11
15. Thông báo client address ở server . . . . .	12
16. GUI tạo tài khoản. . . . .	12
17. GUI đăng nhập. . . . .	13
18. GUI chatbox . . . . .	14

## 1. Giới thiệu các bước cần thiết để hoàn thành bài tập lớn:

### 1.1. Socket Programming

a. Socket là một cổng logic mà một chương trình sử dụng để kết nối với một chương trình khác chạy trên một máy tính khác trên Internet. Chương trình mạng có thể sử dụng nhiều Socket cùng một lúc, nhờ đó nhiều chương trình có thể sử dụng Internet cùng một lúc. có 2 loại socket thường thấy đó là dùng giao thức TCP hoặc UDP

- Stream Socket Dựa trên giao thức TCP việc truyền dữ liệu chỉ thực hiện giữa 2 quá trình đã thiết lập kết nối. Giao thức này đảm bảo dữ liệu được truyền đến nơi nhận một cách đáng tin cậy, đúng thứ tự nhờ vào cơ chế quản lý luồng lưu thông trên mạng và cơ chế chống tắc nghẽn.
- Datagram Socket Dựa trên giao thức UDP việc truyền dữ liệu không yêu cầu có sự thiết lập kết nối giữa 2 quá trình. Ngược lại với giao thức TCP thì dữ liệu được truyền theo giao thức UDP không được tin cậy, có thể không đúng trình tự và lặp lại. Tuy nhiên vì nó không yêu cầu thiết lập kết nối không phải có những cơ chế phức tạp nên tốc độ nhanh. ... ứng dụng cho các ứng dụng truyền dữ liệu nhanh như chat, game ...



Hình 1: Cơ chế gọi hàm trong lập trình Socket

### b. Python với Socket Programming

Module socket trong Python sẽ giúp chúng ta thực hiện các kết nối client server để giao tiếp giữa các máy với nhau. Quy trình hoạt động của ứng dụng Server-Client như sau: Server có nhiệm vụ của là lắng nghe, chờ đợi kết nối từ Client trên địa chỉ IP của

mình với PORT được quy định sẵn. Khi Client gửi dữ liệu tới Server thì nó phải giải quyết một công việc là nhận dữ liệu đó → xử lý → trả kết quả lại cho Client.

## 1.2. Đa luồng (Multi-thread):

### a. Khái niệm

- Việc chạy đa luồng gần giống việc chạy các tiến trình riêng lẻ. Nhưng nó có các lợi thế sau:
  - Các thread chạy trên cùng một tiến trình (process), chia sẻ không gian lưu trữ dữ liệu, tiến trình chính. Vì vậy, việc cập nhật dữ liệu và trao đổi thông tin giữa các luồng cũng dễ dàng hơn là giữa các tiến trình riêng lẻ.
  - Chạy đa luồng không yêu cầu quá nhiều về nơi lưu trữ dữ liệu.
- Một luồng khi được khai báo bao gồm : sự bắt đầu, sự thực hiện và sự kết thúc. Việc chạy đa luồng yêu cầu luôn có một con trỏ để theo dõi và chuyển đổi giữa các luồng.

### b. Python với Multi-Threading

- Trong chương trình này có sử dụng đa luồng ở phần lệnh:
  - `clients[uname] = client`
    - `clients[uname] = client`
    - `threading.Thread(target = handleClient, args = (client, uname, )).start()`
- Chú thích code: Khi server đang hoạt động thì nếu có yêu cầu kết nối từ một client mới thì sẽ tạo một luồng mới để client này có thể hòa vào mạng lưới chat.

## 1.3. MySQL — Cơ sở dữ liệu

### a. Khái niệm

- Database (Cơ sở dữ liệu) là một tập hợp dữ liệu đã được tổ chức sắp xếp. Mục đích chính của Database là để tổ chức một lượng lớn thông tin bằng việc lưu trữ, thu thập, và quản lý.
- SQL là viết tắt của Structured Query Language, là ngôn ngữ truy vấn mang tính cấu trúc. Nó được thiết kế để quản lý dữ liệu trong một hệ thống quản lý cơ sở dữ liệu quan hệ (RDBMS). SQL là ngôn ngữ cơ sở dữ liệu, được sử dụng để tạo, xóa trong cơ sở dữ liệu, lấy các hàng và sửa đổi các hàng, ...
- MySQL là một SQL Database mã nguồn mở, sử dụng SQL như là ngôn ngữ cơ sở dữ liệu chuẩn.

▪ Một số ưu điểm của MySQL:

- Hiệu suất cao
- Tính khả dụng cao
- Khả năng mở rộng và linh hoạt
- Quản lý dễ dàng

b. Python với MySQL

▪ Sử dụng thư viện sqlite3:

Step 1. kết nối với database:

---

```
1 db = sqlite3.connect('users.db')
2 c = db.cursor()
```

---

Step 2. Tạo bảng với các cột name, age, username, password, gender:

---

```
1 c.execute('create table if not exists users(name TEXT
NOT NULL,age INT NOT NULL,username TEXT NOT
NULL,password TEXT NOT NULL, gender TEXT NOT
NULL)')
```

---

Step 3. Chèn dữ liệu username và password:

---

```
1 c.execute('SELECT * FROM users WHERE username = ? and
password = ?',(username,password))
```

---

Step 4. Nếu muốn chèn nhiều thông tin hơn:

---

```
1 c.execute('INSERT INTO users VALUES
(?,?,?,?,?)',(name, age, username, pasword,
gender))
```

---

Step 5. Lưu lại những gì đã làm:

---

```
1 db.commit()
```

---

Step 6. Kết thúc phiên làm việc:

---

```
1 db.close()
```

---

## 1.4. GUI — Giao diện

### a. Phần mềm hỗ trợ:

Trước tiên cài đặt PyQt và Qt Designer

Nhóm sẽ sử dụng PyQt5 với Python 3. Nếu đã cài đặt PyQt4 hoặc PySide thì không cần phải hạ cấp, chuyển đổi.

Trên Linux, các gói được yêu cầu có thể nằm trong kho lưu trữ distro, nếu đang sử dụng Ubuntu, Debian, chạy lệnh:

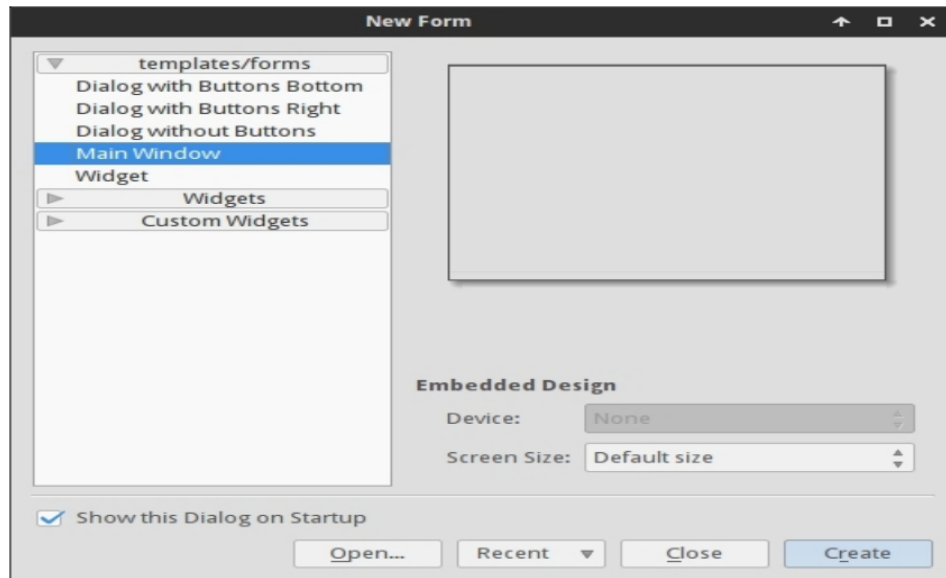
---

```
1 $ apt-get install python3-qt5 pyqt5-dev-tools qt5-designer
```

---

### b. Các bước tạo GUI:

Bước 1. Mở Qt Designer thấy hộp thoại biểu mẫu mới, chọn “Widget” và nhấp vào “Create”



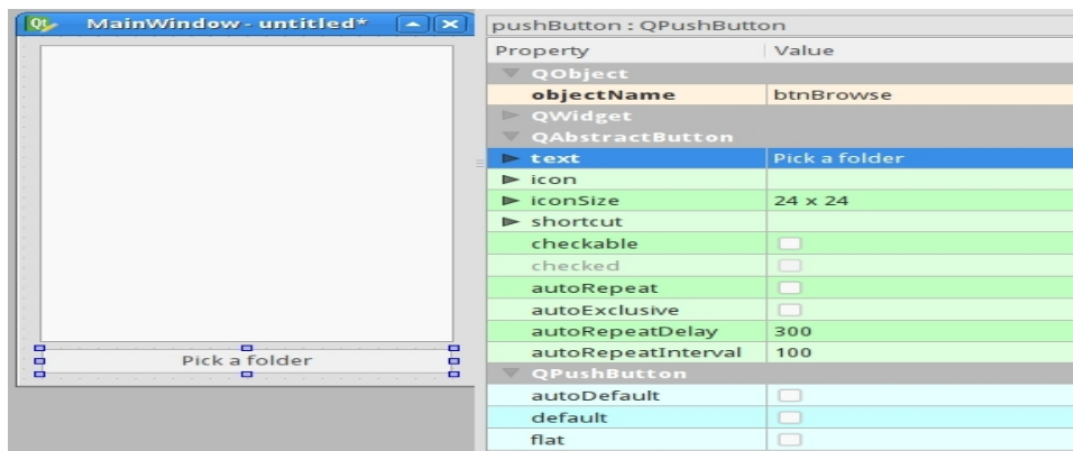
Hình 2: Tạo một GUI mới

Bước 2. Sử dụng động tác kéo thả để tạo giao diện theo yêu cầu cần có.

Các loại Object cần thiết để sử dụng nằm trong “Widget Box” gồm : Layouts, Spacers, Buttons, Item Views, Item Widgets, ...

Bước 3. Thay đổi thuộc tính và hoàn thiện.

Các thuộc tính của Object mới được tạo ra nằm trong bảng Property



Hình 3: Thuộc tính Object

c. Python với PyQt5

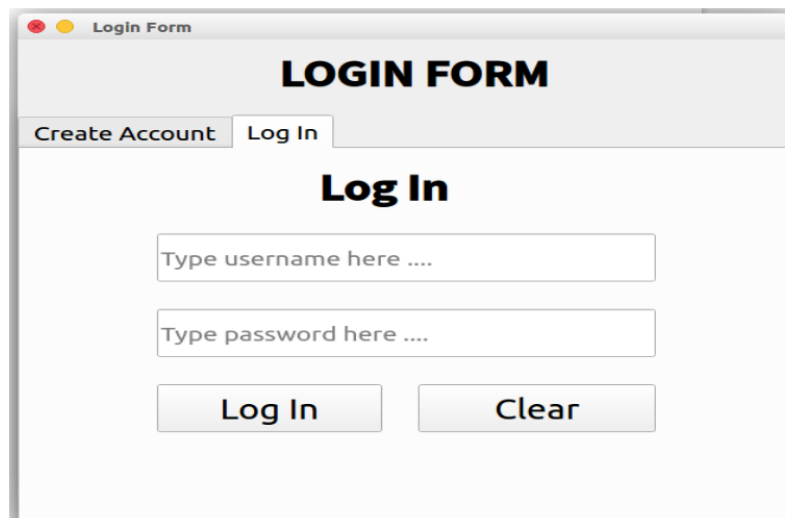
Có thể sử dụng các tệp .ui trực tiếp từ mã python, nhưng chuyển đổi mã .ui thành tệp python để có thể chỉnh sửa dễ dàng hơn. Để chuyển đổi tệp tin, sử dụng lệnh pyuic5 từ terminal, command prompt

---

```
1 $ pyuic5 -x design.ui -o design.py
```

---

Sau khi chỉnh sửa file python được kết quả hoàn chỉnh:



Hình 4: Kết quả GUI

## 1.5. NAT Port

### a. Khái niệm:

- Port Forwarding là quá trình chuyển tiếp một port cụ thể từ mạng này đến mạng khác, cho phép người dùng bên ngoài có thể truy cập vào mạng bên trong bằng cách sử dụng port đó từ bên ngoài thông qua bộ định tuyến (đã mở NAT – Network Address Translation).
- Các Port là được đánh số – được giới hạn từ 0 đến 65535 và bình thường thì tất cả các port này sẽ đóng.

### b. Các bước để thực hiện NAT Port:

Thực hiện ở kí túc xá Bách Khoa dùng Router loại: HG8045A của Huawei

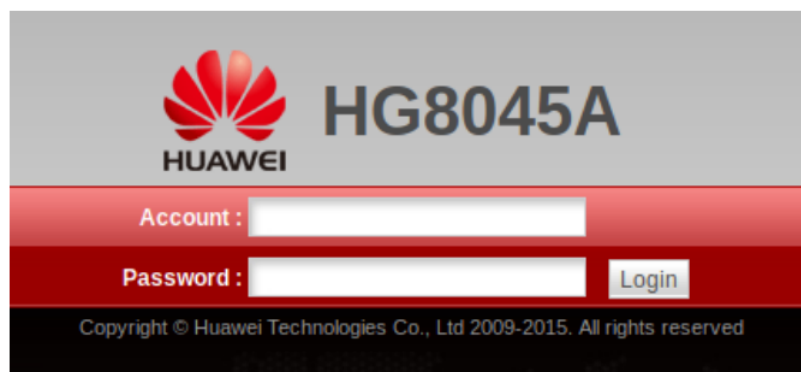
**Step 1.** Tìm kiếm địa chỉ IP của Router để nhập vào trình duyệt web: Đối với ubuntu, nhập: `ifconfig`, ta sẽ nhận được một thông báo.

```
wlp8s0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500  
inet 192.168.0.7 netmask 255.255.255.0 broadcast 192.168.0.255
```

Hình 5: Hiện IP khi dùng lệnh `ifconfig`.

Vậy địa chỉ IP của Router là: 192,168,0,1

**Step 2.** Đăng nhập vào router thông qua trang web và NAT PORT:



Hình 6: Bảng đăng nhập của router.

Sau đó click vào phần **Forward Rules** → **Port Mapping Configuration**





Hình 7: Thanh “Navigation” và “sidebar” của router web.

Ở đây, ta sẽ chọn mục **New**, điền các thông số:

- Mapping name: Là tên ánh xạ
- Internal Port number: Là số lượng port bên trong mạng local muốn ánh xạ.
- External Port number: Là số lượng port mà bên ngoài mạng global muốn ánh xạ.

The screenshot shows the 'New' NAT Port configuration form. It includes fields for 'Type' (User-defined selected), 'Enable Port Mapping' (checked), 'Mapping Name' (NatPort\_MangMayTinh), 'WAN Name' (1\_INTERNET\_R\_1), 'Internal Host' (192.168.0.7), 'External Source IP Address' (empty), 'Protocol' (TCP/UDP), 'Internal port number' (2000), 'External port number' (2000), and 'External source port number' (65535). There are 'Delete', 'Add', 'Apply', and 'Cancel' buttons at the bottom.

Hình 8: Điền thông tin NAT PORT

Sau khi điền xong, nhấn nút **Apply** là hoàn tất quá trình **NAT PORT** và nhận được thông báo.

	Mapping Name	WAN Name	Internal Host	External Host	Enable
<input type="checkbox"/>	Rasp_Qu	1_INTERNET_R_VID_10	192.168.0.2	--	Enable
<input type="checkbox"/>	NatPortQuang	1_INTERNET_R_VID_10	192.168.0.11	--	Enable
<input type="checkbox"/>	natphuongpi	1_INTERNET_R_VID_10	192.168.0.3	--	Enable
<input type="checkbox"/>	NatPort_MangMayT.....	1_INTERNET_R_VID_10	192.168.0.7	--	Enable

Hình 9: Kết quả

## 2. Mô hình hệ thống, thuật toán, chương trình bài tập lớn:

### 2.1. Mô hình hệ thống:

- Gồm 2 phần:
  - a. Server:
    - Chuyển đoạn chat qua lại cho các client khác.
    - Xử lý các lệnh thực thi do client gửi lên.
  - b. Client:
    - Kết hợp với GUI (giao diện) để thuận tiện cho người dùng.
    - Gửi lệnh thực thi, hoặc nội dung chat lên server.

### 2.2. Sơ đồ thuật toán server:

#### SERVER FOR NET CHATTING

step 1. Get Public IP

step 2. Socket TCP/IP

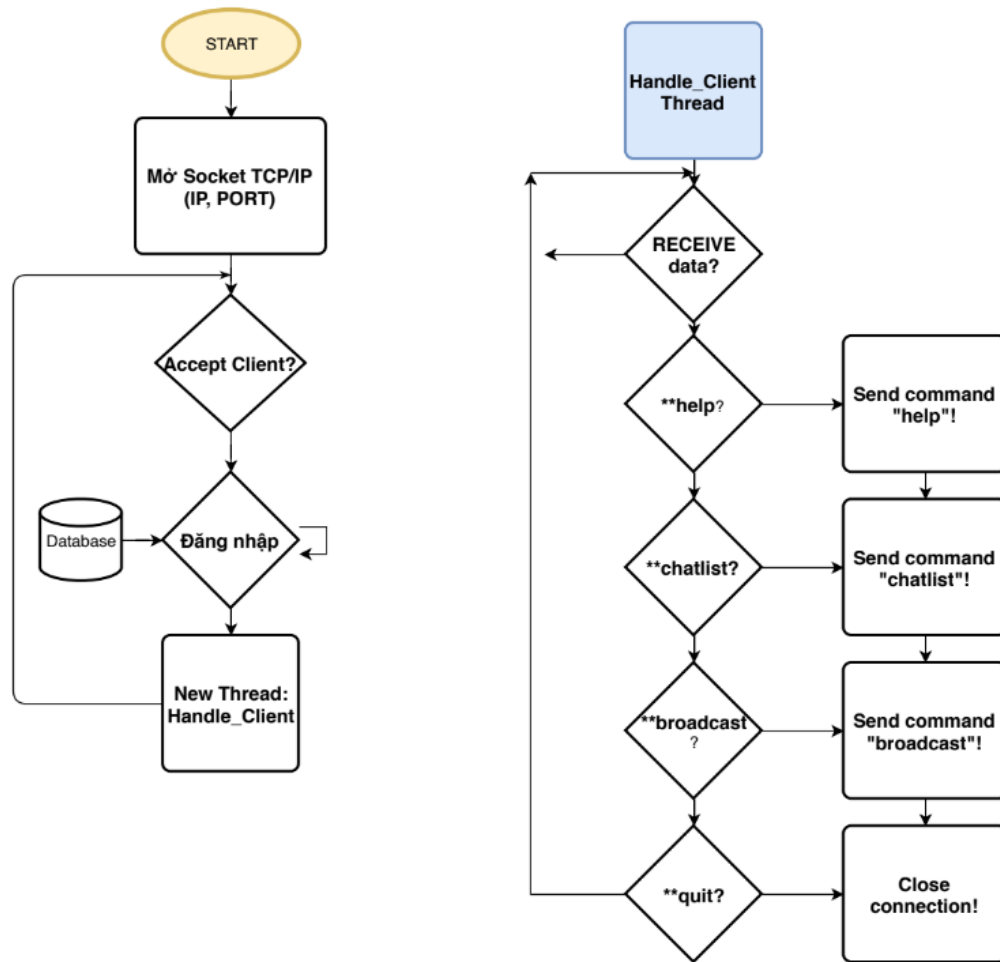
step 3. Connect Database Function

step 4. Handle Client Thread

step 5. Wait incoming connection and Login (Main)

step 6. After Login okay, Handle command from client!

Ta có mô hình thuật toán server sau:



Hình 10: Thuật toán Sever

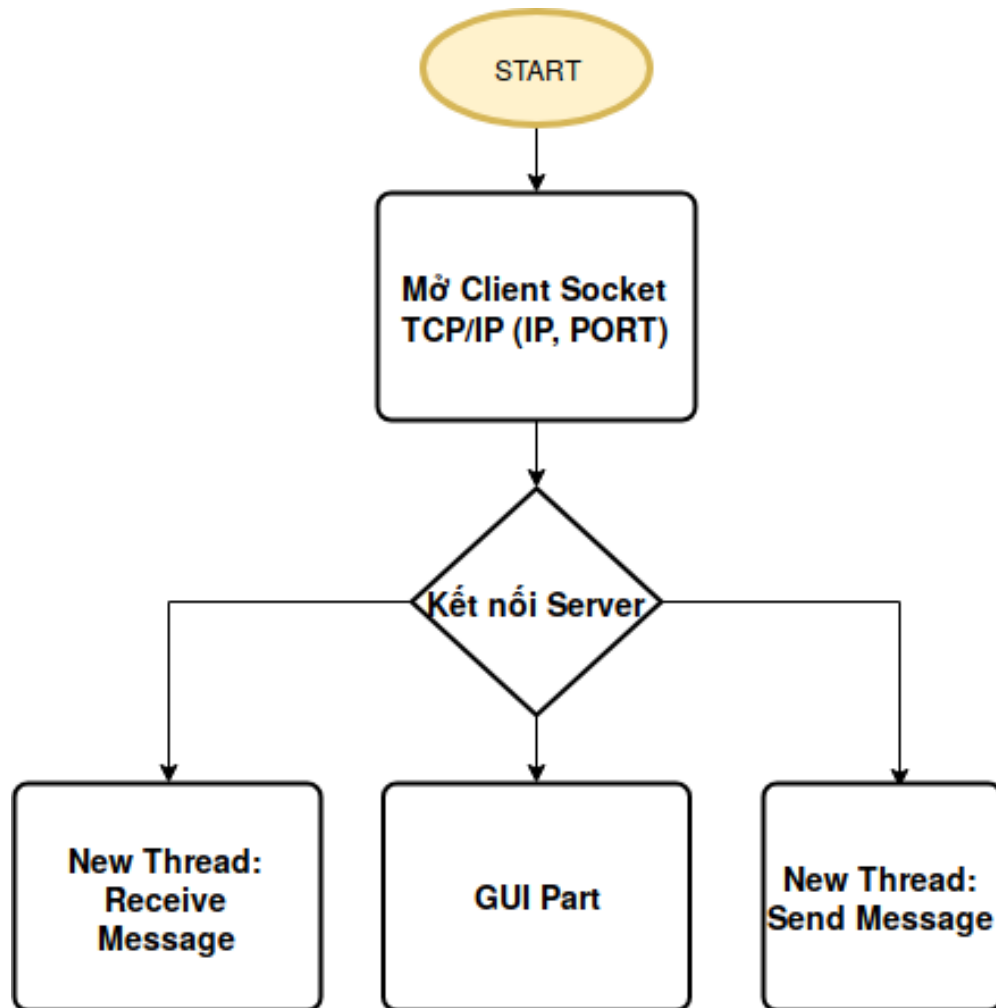
### 2.3. Sơ đồ thuật toán client:

#### CLIENT FOR CHATTING

1. Include Library for GUI
2. Initialise GUI
3. Draw GUI
4. Connect to server
5. Login
6. After login okay, Open GUI Chatbox!

## 7. Receive and Send message, command!

Ta mô tả thuật toán Client như sau:



Hình 11: Thuật toán Client

### 3. Tổng kết, demo

#### 3.1. Public IP của server:

Public IP của router do nhà dịch vụ VNPT cung cấp. Public IP được tìm bằng code như sau:

```
#Get Public IP
from requests import get
ip = str(get('https://api.ipify.org').text)
print("My public IP address is:")
print(ip)
```

Hình 12: Public IP được lấy bằng Code Python

### 3.2. Mở server:

---

```
1 $python3 server.py
```

---

Ngay sau đó, thông báo xuất hiện:

```
vanquang@vanquang-Vostro-3560:~/Desktop/MangMayTinh/OfficialCode/LoginForm$ python3 server.py
My public IP address is:
14.226.231.175
Server Ready...
Ip Address of the Server::14.226.231.175
Server Port: 5555
-----Listening! Imcoming connection-----
```

Hình 13: Thông báo trên server

### 3.3. Mở kết nối từ client đến server.

---

```
1 $python3 clientChat.py
```

---

Sau đó, bắt buộc nhập **địa chỉ** và **port** của server.

```
vanquang@vanquang-Vostro-3560:~/Desktop/MangMayTinh/OfficialCode/LoginForm$ python3 clientChat.py
Enter the Server Address:: 14.226.231.175
Enter PORT:: 5555
receive Message is ready!
Login first!
```

Hình 14: Sau khi nhập thành công, hiển thị trên client.

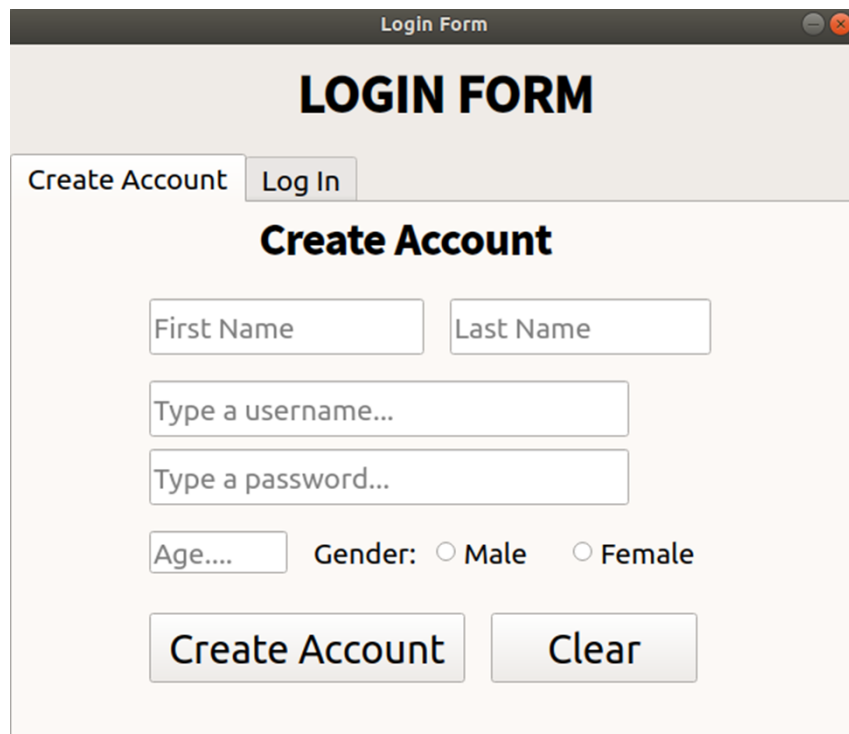
Ngay lúc kết nối thành công, server hiện thông báo: show IP của client kết nối lên.

```
vanquang@vanquang-Vostro-3560:~/Desktop/MangMayTinh/OfficialCode/LoginForm$ python3 server.py
My public IP address is:
14.226.231.175
Server Ready...
Ip Address of the Server::14.226.231.175
Server Port: 5555
-----Listening! Incoming connection-----
Client address:
('14.226.231.175', 55222)
-----
```

Hình 15: Thông báo client address ở server

Lúc này, GUI đăng nhập hiện lên ở client:

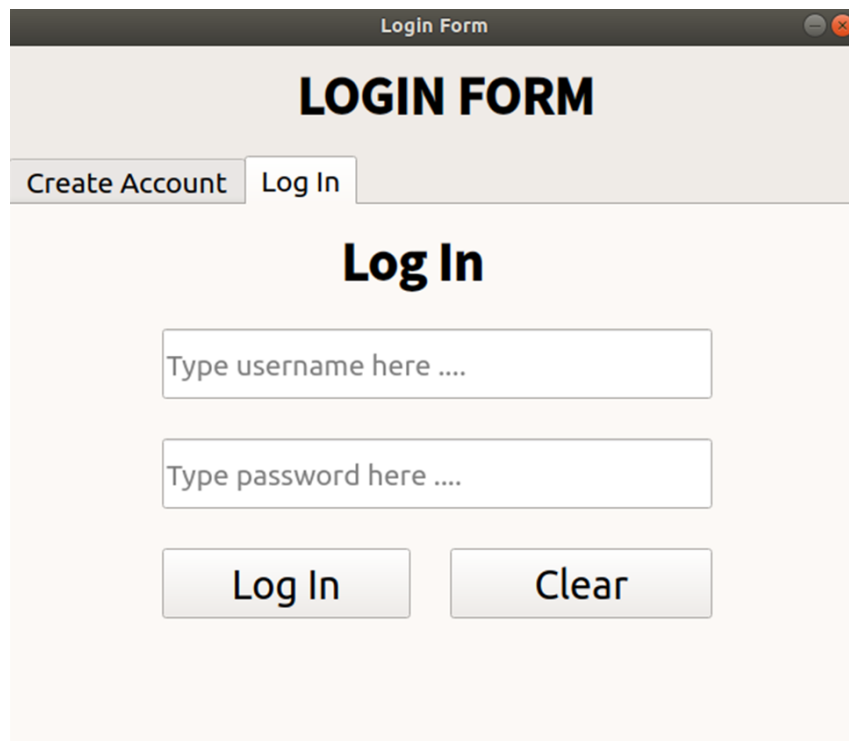
1. Tạo tài khoản:

The image shows a window titled "Login Form" with a light beige background. At the top, there's a dark header bar with the title. Below the header, the text "LOGIN FORM" is displayed in large, bold, black letters. Underneath, there are two tabs: "Create Account" (which is active) and "Log In". The "Create Account" section contains several input fields: "First Name", "Last Name", "Type a username...", and "Type a password...". Below these, there's an "Age...." field and a "Gender:" section with radio buttons for "Male" and "Female". At the bottom of the form, there are two buttons: "Create Account" and "Clear".

Hình 16: GUI tạo tài khoản.

Để tạo tài khoản, ta cần điền đầy đủ thông tin và nhấn nút “Create Account”

2. Đăng nhập:



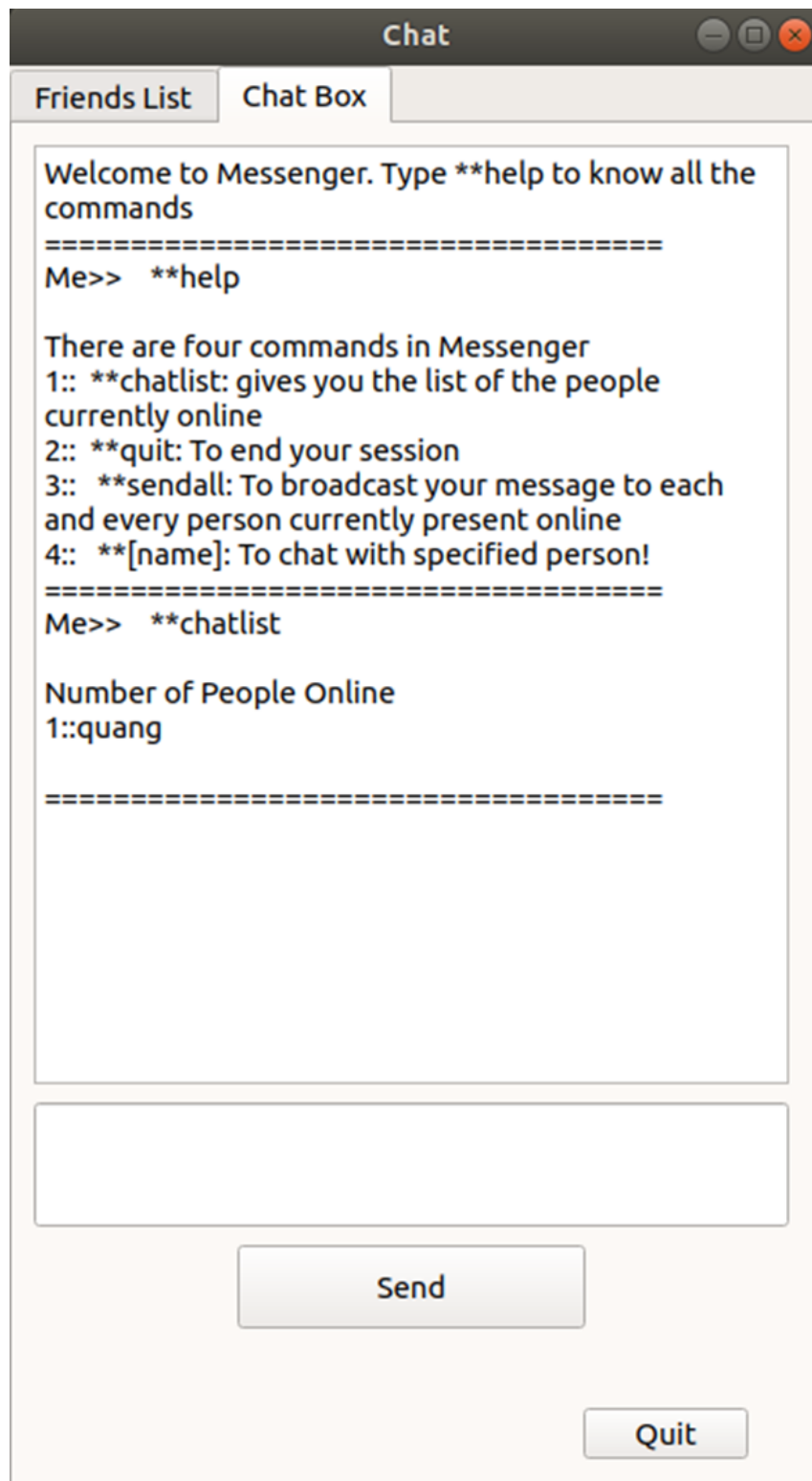
The image shows a graphical user interface for a login form. The window has a title bar that says "Login Form". Inside the window, at the top, is a large bold heading "LOGIN FORM". Below this heading are two tabs: "Create Account" and "Log In". The "Log In" tab is currently selected. Under the "Log In" tab, there is a sub-heading "Log In". Below the sub-heading are two text input fields. The first field has the placeholder text "Type username here ...." and the second field has the placeholder text "Type password here ....". At the bottom of the form are two buttons: "Log In" and "Clear".

Hình 17: GUI đăng nhập.

Đăng nhập đúng tên (**username**) và mật khẩu (**password**) và nhấn nút **Login**

### 3.4. Chat box:

Sau khi đăng nhập thành công, ta có giao diện được hiển thị lên:



Hình 18: GUI chatbox



Ở đây, ta có 4 lệnh để thực thi:

1. `**help`: Đưa ra thông tin chỉ dẫn
2. `**chatlist`: Show danh sách các thành viên đang online.
3. `**quit`: Kết thúc cuộc trò chuyện.
4. `**sendall`: Gửi tin nhắn đến tất cả các thành viên đang online.
5. `**tên`: Gửi tin nhắn riêng đến 1 thành viên đang online.

### 3.5. Tổng kết:

Như vậy, nhóm đã làm xong project net-chatting, sử dụng socket, thread, SQL, GUI và NatPort, là những công cụ rất cần thiết cho một lập trình viên, kiến thức cần phải hành trang ngay từ trong ghế nhà trường. Trong tương lai, nhóm sẽ tiếp tục nỗ lực xây dựng nhiều ứng dụng hơn nữa, ví dụ chơi game, buzz, emoticon... để có một kỉ niệm đẹp về Yahoo! Messenger thời đó.