BÁO CÁO THỰC HÀNH

**Môn học: Lập trình mạng căn bản**

**Buổi báo cáo: Lab 03**

**Tên chủ đề:** Lập trình Sockets trong C#

*GVHD:* Nguyễn Xuân Hà

*Ngày thực hiện: 10/05/2024*

**THÔNG TIN CHUNG:**

Lớp: NT106.O23.1

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Lê Minh Hoàng | 22520463 | 22520463@gm.uit.edu.vn |

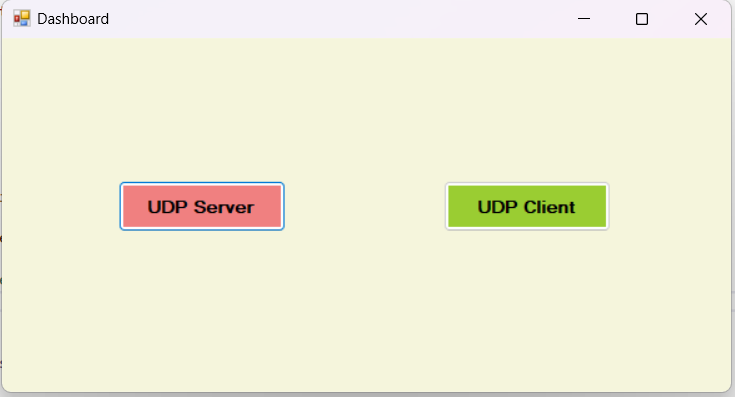
1. **ĐÁNH GIÁ KHÁC:**

|  |  |
| --- | --- |
| **Nội dung** | **Kết quả** |
| Tổng thời gian thực hiện bài thực hành trung bình | Khoảng hơn 1 tuần |
| Link Github | https://github.com/MHabc |
| Link Video thực hiện  *(nếu có)* | https://bom.so/lgdYJ0 |
| Ý kiến *(nếu có)*  + Khó khăn  + Đề xuất … |  |
| Điểm tự đánh giá | 95/100 |

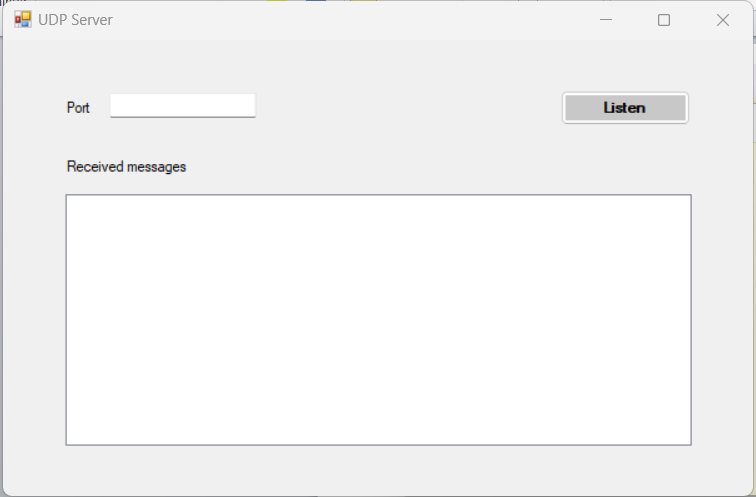
BÁO CÁO CHI TIẾT

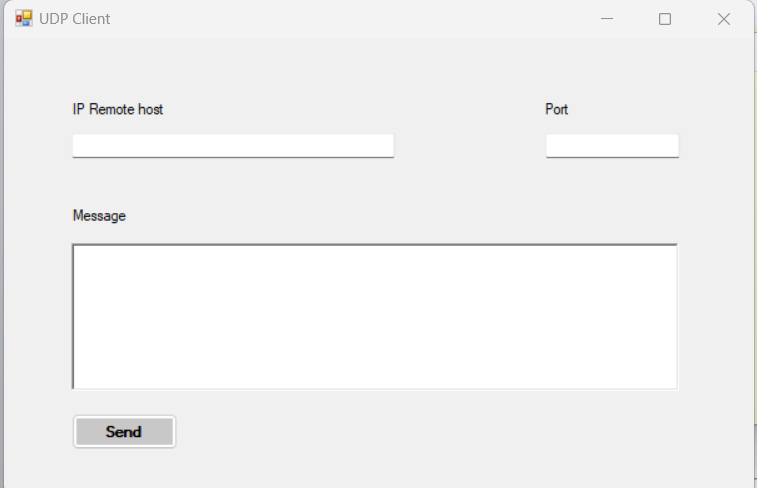
**1. Bài 01**

**\* Giao diện Dashboard :**



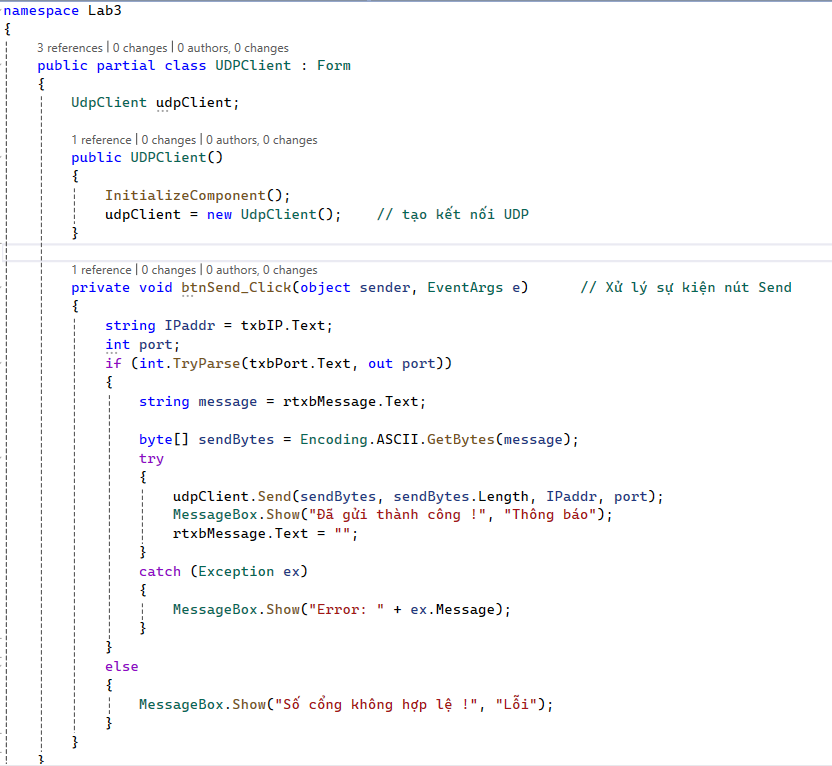
- Khi nhấn vào các button UDP Server và UDP Client sẽ dẫn đến các form tương ứng.





**\* Giải thích code :**

**- Code UDP Client :**



- Đầu tiên, khởi tạo và thiết lập kết nối UDP.

- Bắt sự kiện Click cho nút Send :

+ Lấy địa chỉ IP host, số cổng, nội dung message được người dùng nhập vào từ các textBox và richTextBox.

+ Chuyển nội dung message thành mảng byte để gửi đi.

+ Gửi message mà không cần thiết lập kết nối với Server với địa chỉ IP và cổng được chỉ định bằng phương thức Send của ‘UdpClient’.

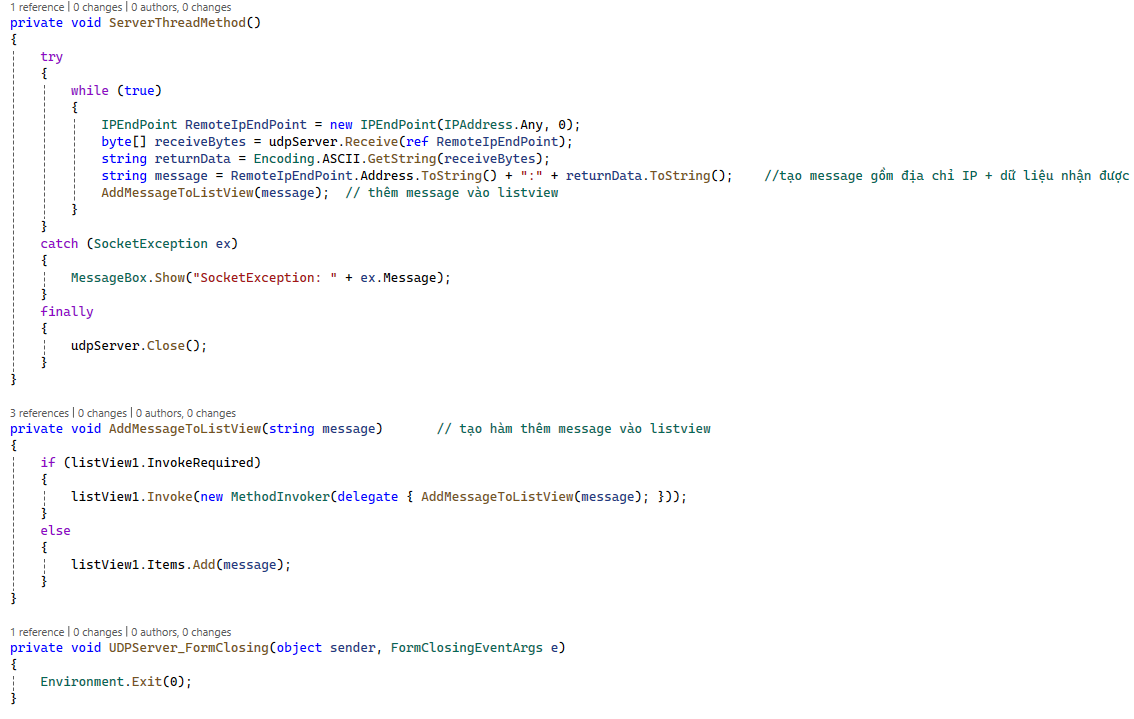
+ Thông báo nếu gửi thành công hoặc thông báo lỗi nếu gửi thất bại.

**- Code UDP Server :**



- Bắt sự kiện Click cho nút Listen : chương trình sẽ tạo 1 kết nối UDP để lắng nghe trên 1 cổng cụ thể.

- Khởi tạo phương thức StartListening : tạo 1 luồng mới để lắng nghe kết nối từ máy khách.



- Khởi tạo phương thức ServerThreadMethod :

+ Tạo 1 IPEndPoint để lưu thông tin về địa chỉ IP và cổng của máy khách.

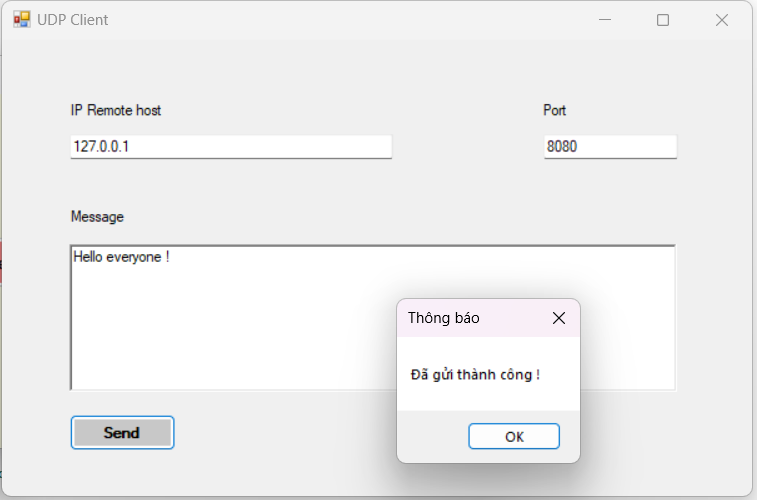
+ Nhận dữ liệu từ máy khách và chuyển thành mảng byte, sau đó chuyển sang chuỗi.

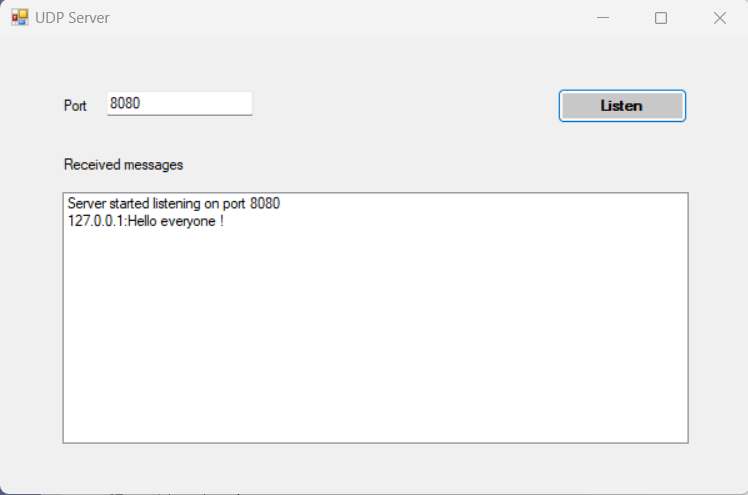
+ Tạo message gồm địa chỉ IP + dữ liệu nhận được.

+ Hiển thị ra ListView tương ứng bằng cách gọi đến phương thức AddMessageToListView.

**\* Kết quả thực thi chương trình :**

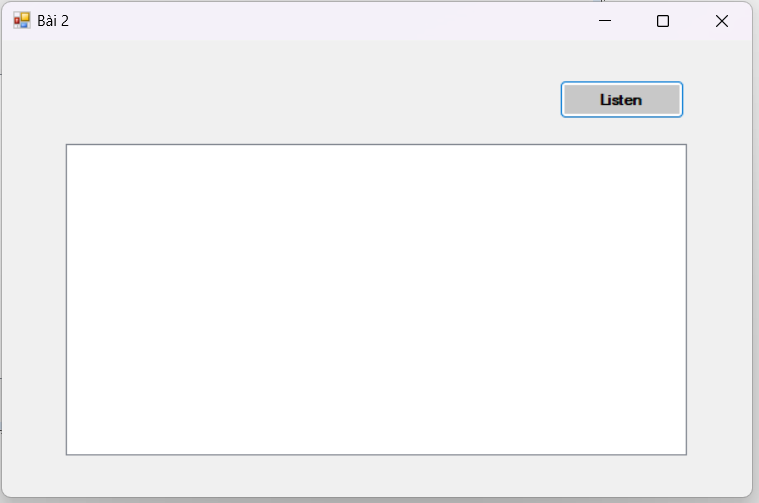
- Khi Client nhập vào địa chỉ IP, số cổng và gửi thông điệp đến Server thì bên Server sẽ nhận được thông điệp và hiển thị lên List View tương ứng.

****

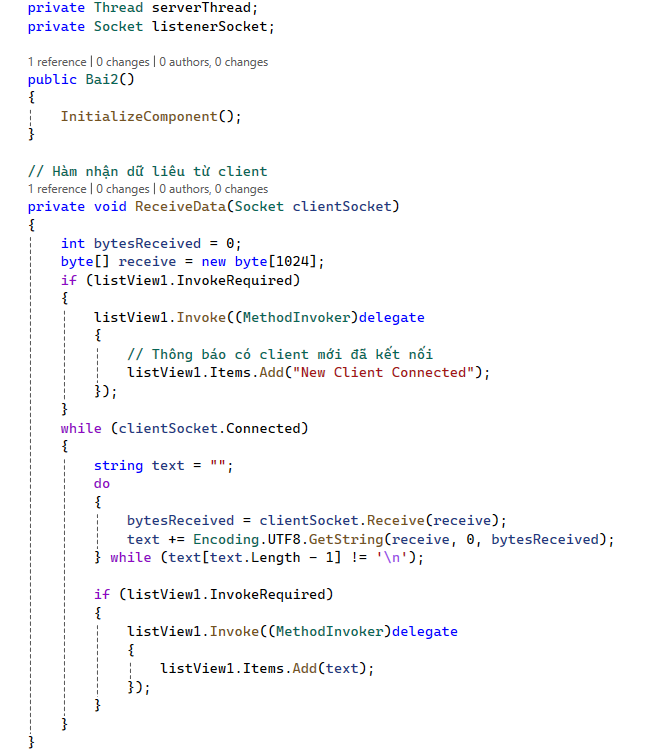
****

**2. Bài 02**

**\* Giao diện bài 2 :**



**\* Giải thích code :**



- Tạo hàm ReceiveData :

+ Nhận dữ liệu từ client thông qua một kết nối socket đã được thiết lập.

+ Sử dụng vòng lặp nhận dữ liệu từ client và hiển thị lên list view nếu client vẫn còn kết nối.

+ Việc hiển thị dữ liệu được thực hiện bằng cách sử dụng Invoke để đảm bảo an toàn trên luồng giao diện người dùng.



- Tạo hàm StartListen :

+ Bắt đầu lắng nghe kết nối từ client. Nó tạo một socket listener và liên tục chấp nhận kết nối mới từ client bằng cách sử dụng một vòng lặp vô hạn.

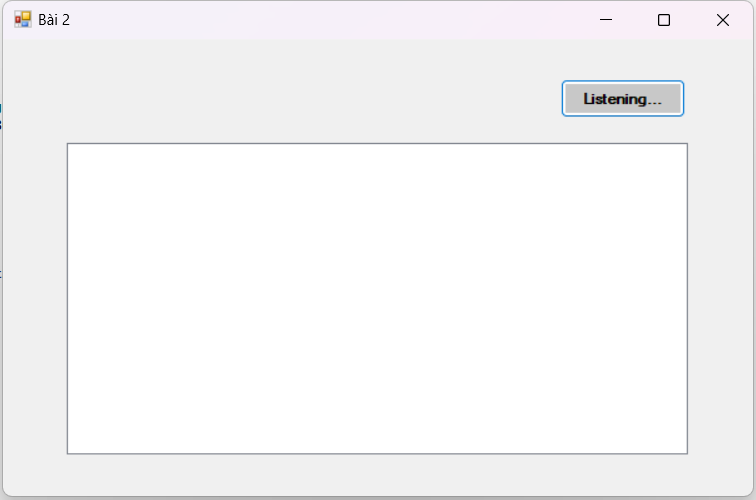
+ Mỗi khi có một kết nối mới được chấp nhận, nó tạo một luồng mới để xử lý kết nối đó bằng cách gọi hàm ReceiveData.

- Bắt sự kiện Click cho nút Listen : Khi nút được nhấn, chương trình sẽ bắt đầu lắng nghe kết nối từ client bằng cách tạo một tiến trình mới gọi hàm StartListen.

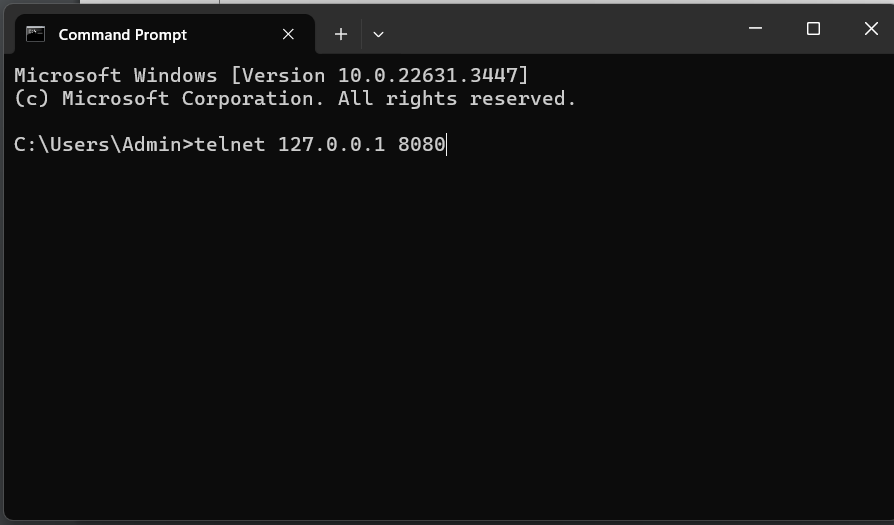
- Bắt sự kiện đóng form để kết thúc chương trình.

**\* Kết quả khi thực thi chương trình :**

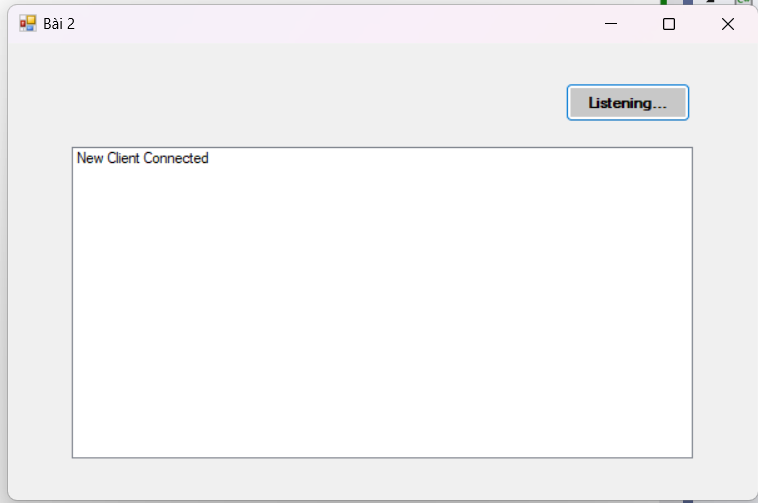
- Đầu tiên nhấn vào nút Listen để lắng nghe kết nối tại địa chỉ IP và port của máy. Khi đó nút Listen sẽ chuyển thành trạng thái Listening…

****

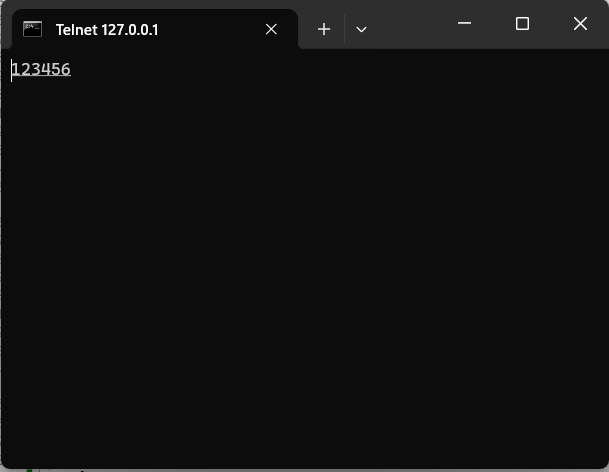
- Trong cmd, gõ lệnh : telnet 127.0.0.1 8080

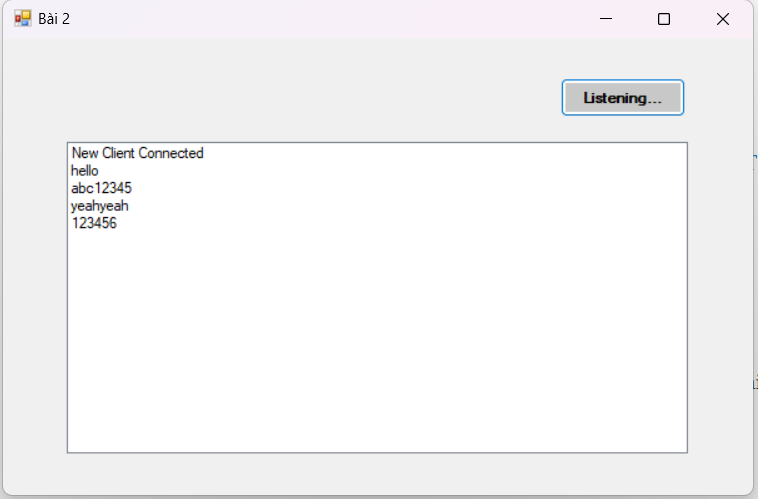
****

- Khi đó, bên server nhận được thông báo đã có client mới kết nối



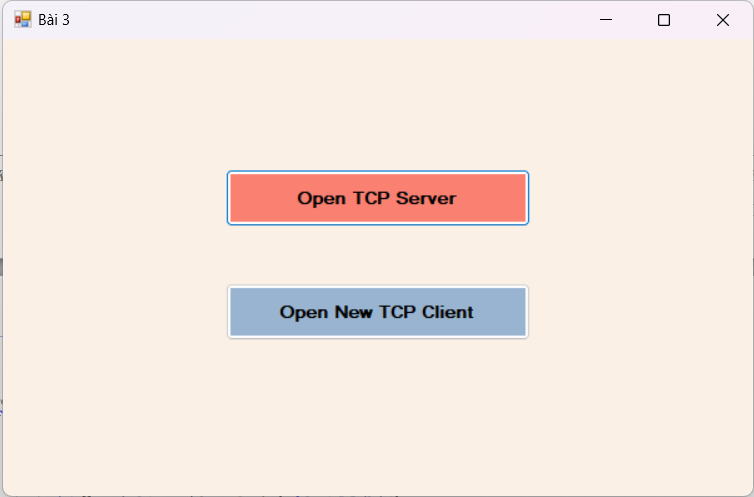
- Thực hiện gửi thông điệp tùy ý, chương trình sẽ nhận được và hiện lên form.





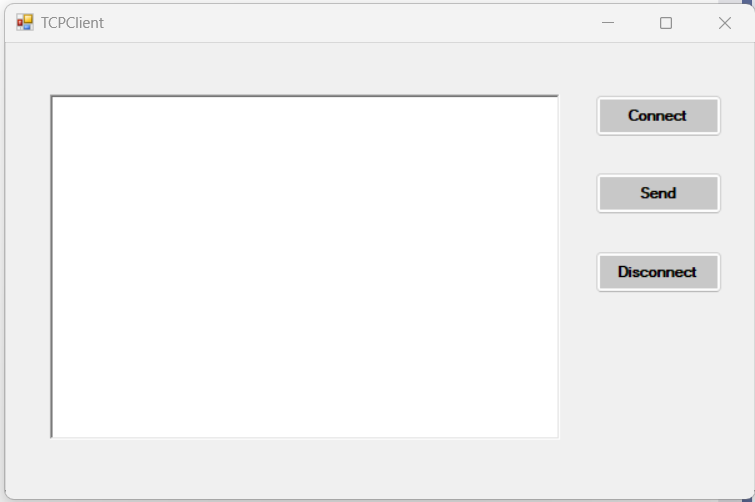
**3. Bài 03**

**\* Giao diện bài 3 :**

****

- Khi nhấn vào các button Open TCP Server và Open New TCP Client sẽ dẫn đến các form tương ứng.



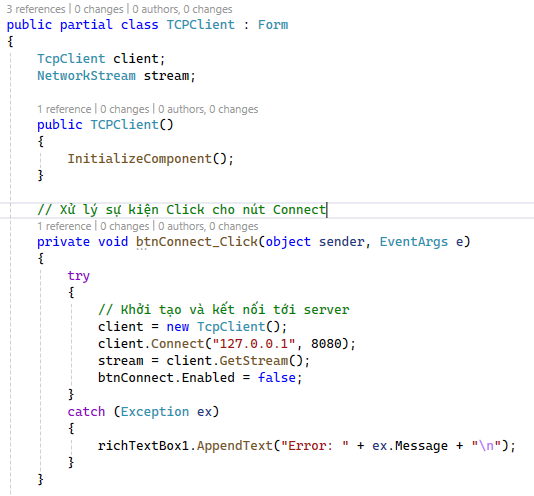


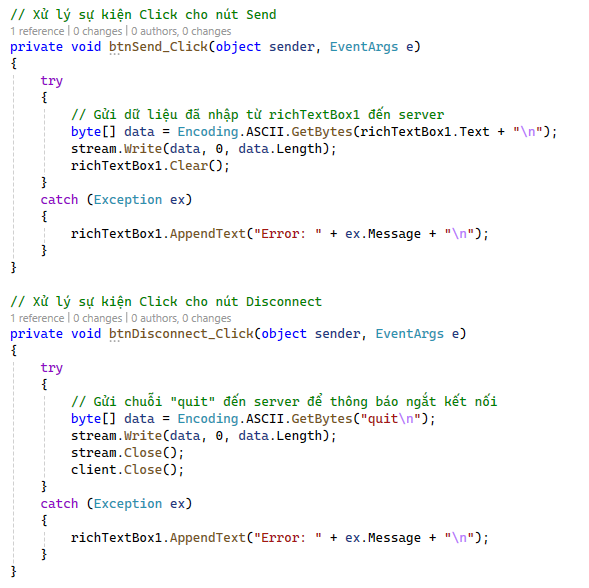
**\* Giải thích code :**

**- Code TCP Server :**

Phần code và giao diện tương tự như TCP Server ở bài 2.

**- Code TCP Client :**





- Đầu tiên, tạo đối tượng TcpClient và tạo luồng để đọc và ghi dữ liệu dựa trên NetworkStream.

- Bắt sự kiện Click cho nút Connect :

+ Tạo đối tượng TcpClient, kết nối đến tới server tại địa chỉ IP và port xác định.

+ Lấy ‘NetworkStream’ để giao tiếp với server, sau đó vô hiệu hóa nút Connect.

- Bắt sự kiện Click cho nút Send :

+ Lấy dữ liệu đã nhập ở richTextBox1 chuyển sang mảng byte rồi gửi đến server bằng phương thức Write.

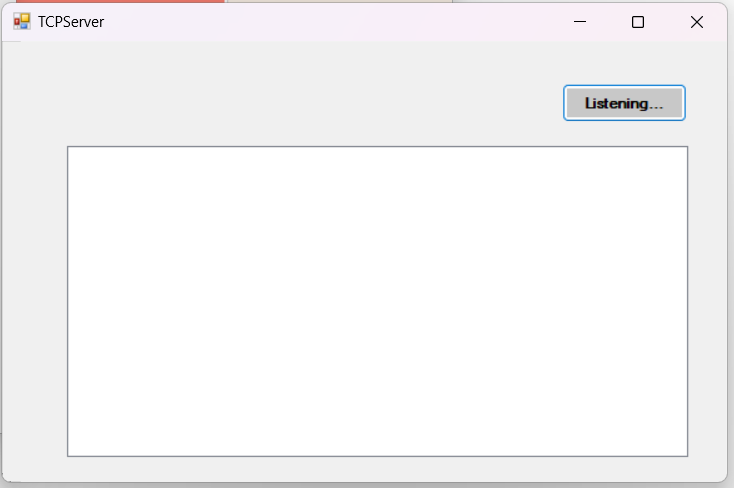
- Bắt sự kiện Click cho nút Disconnect :

+ Dùng phương thức Write để gửi dữ liệu mang dấu hiệu kết thúc cho Server biết và đóng kết nối.

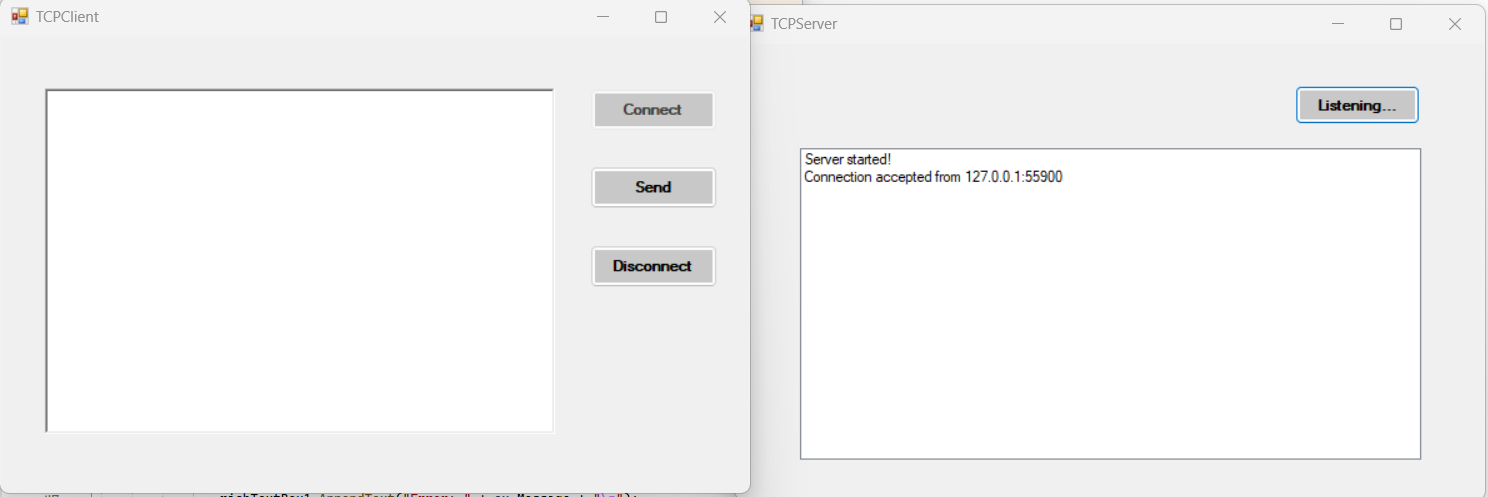
+ Đóng kết nối TcpClient và NetworkStream.

**\* Kết quả khi thực thi chương trình :**

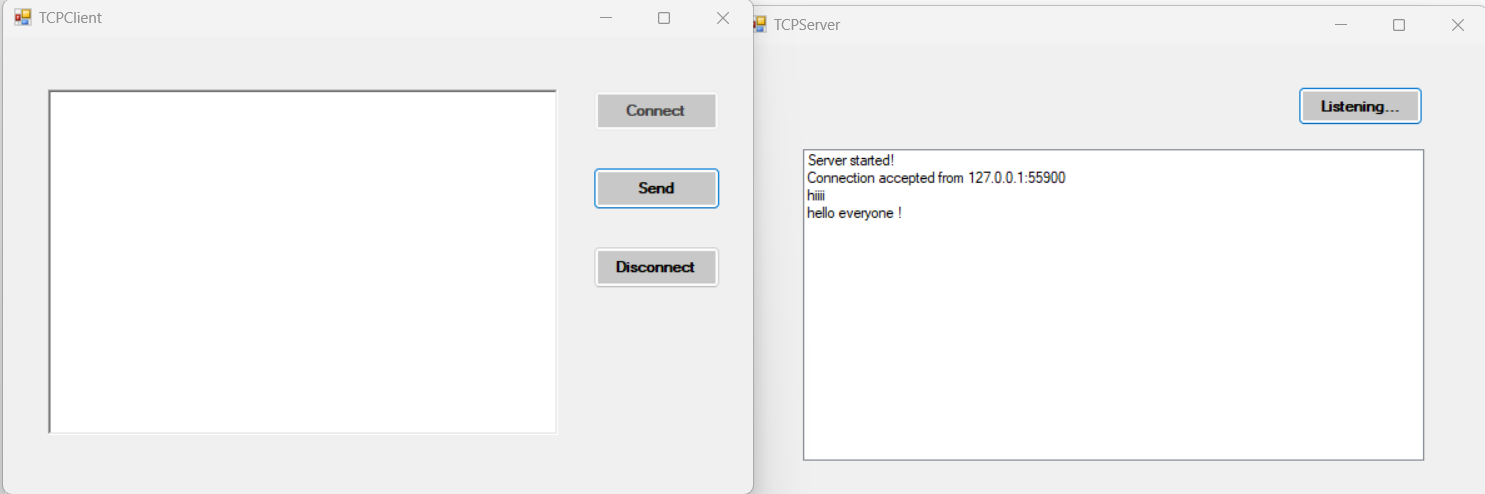
- Đầu tiên, TCP Server lắng nghe kết nối :

****

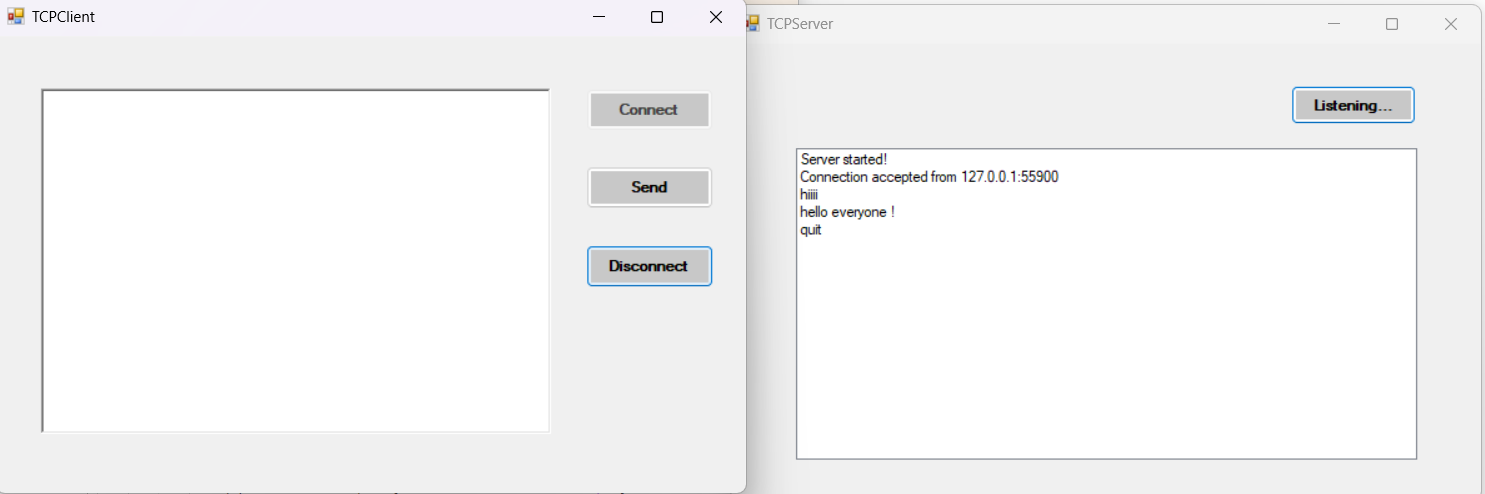
- Khi client nhấn nút Connect thì bên server sẽ hiển thị đã nhận kết nối :

****

- Khi client nhập thông điệp và Send thì server sẽ nhận được và hiển thị lên list view :



- Khi client nhấn nút Disconnect thì chuỗi “quit” sẽ được gửi đến báo cho server biết và đóng kết nối :



**4. Bài 04**

**\* Giao diện bài 4**

**\* Giải thích code :**

**\* Kết quả khi thực thi chương trình :**

**5. Bài 05**

**\* Giao diện bài 5 :**

**\* Giải thích code :**

**\* Kết quả khi thực thi chương trình :**

**6. Bài 06**

**\* Giao diện bài 6 :**

**\* Giải thích code :**

**\* Kết quả khi thực thi chương trình :**