

Name: Phan Phạm Trọng Phúc

ID: 21520402

Class: NT216.N21.ANTN

Lập trình mạng căn bản LAB 3'S REPORT

Tạo dashboard để chuyển giữa các bài:

- Phía client

```
namespace Lab03
{
    3 references
    public partial class DashBoard : Form
    {
        1 reference
        public DashBoard()
        {
            InitializeComponent();
        }

        1 reference
        private void button1_Click(object sender, EventArgs e)
        {
            Bai1 bai1 = new Bai1();
            bai1.Show();
        }

        1 reference
        private void button2_Click(object sender, EventArgs e)
        {
            Bai3_Client bai3 = new Bai3_Client();
            bai3.Show();
        }

        1 reference
        private void button3_Click(object sender, EventArgs e)
        {
            Bai4 bai4 = new Bai4();
            bai4.Show();
        }
    }
}
```

- Phía server:

```
namespace Server
{
    public partial class DashBoard : Form
    {
        public DashBoard()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            Bai1 bai1 = new Bai1();
            bai1.Show();
        }

        1 reference
        private void button2_Click(object sender, EventArgs e)
        {
            Bai2 bai2 = new Bai2();
            bai2.Show();
        }

        1 reference
        private void button3_Click(object sender, EventArgs e)
        {
            Bai3 bai3 = new Bai3();
            bai3.Show();
        }

        1 reference
        private void button4_Click(object sender, EventArgs e)
        {
            Bai4 bai4 = new Bai4();
            bai4.Show();
        }
    }
}
```

Bài 1:

- Phía client:

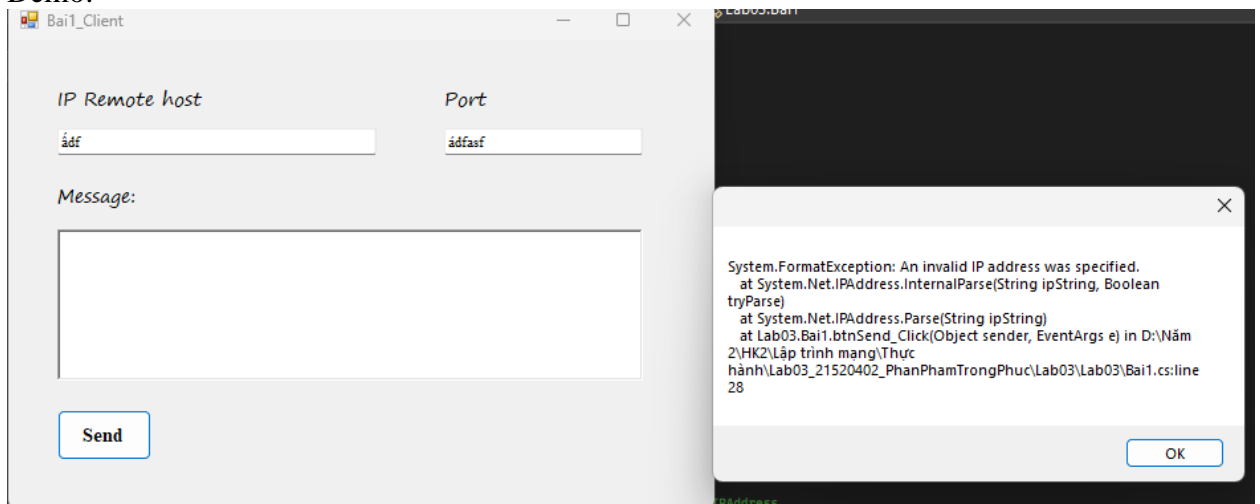
Code:

```
namespace Lab03
{
    4 references
    public partial class Bai1 : Form
    {
        1 reference
        public Bai1()
        {
            InitializeComponent();
        }

        1 reference
        private void btnSend_Click(object sender, EventArgs e)
        {
            try
            {
                UdpClient udpClient = new UdpClient();
                //Lấy địa chỉ IP từ textbox và chuyển thành kiểu IPAddress
                IPAddress ipadd = IPAddress.Parse(tbIP.Text);
                int port = Convert.ToInt32(tbPort.Text);
                IPEndPoint ipend = new IPEndPoint(ipadd, port);
                //Chuyển chuỗi dữ liệu nhập sang kiểu byte
                Byte[] sendBytes = Encoding.UTF8.GetBytes(rtbMessage.Text);
                //Gửi dữ liệu đến IPEndPoint đã định nghĩa địa chỉ IP và Port
                udpClient.Send(sendBytes, sendBytes.Length, ipend);

                udpClient.Close();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.ToString());
            }
        }
    }
}
```

Demo:



Có chức năng kiểm tra IP và Port trước khi gửi

Chỉ định Ip và Port để kết nối đến server

- Phía server:

Code:

```

4 references
public partial class Bai1 : Form
{
    private Thread listenThread;
    1 reference
    public Bai1()
    {
        InitializeComponent();
    }

    1 reference
    private void btnListen_Click(object sender, EventArgs e)
    {
        if (listenThread == null || !listenThread.IsAlive)
        {
            listenThread = new Thread(new ThreadStart(Listen));
            listenThread.IsBackground = true;
            listenThread.Start();
        }
    }

    1 reference
    private void Listen()
    {
        int port;
        if (!int.TryParse(tbPort.Text, out port))
        {
            MessageBox.Show("Vui lòng nhập Port đúng.");
            return;
        }

        UdpClient udpClient = new UdpClient(port);

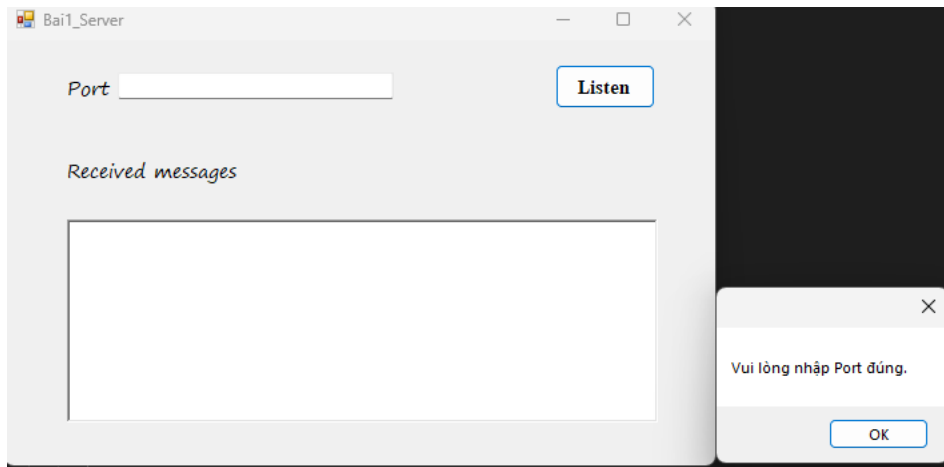
        while (true)
        {
            IPEndPoint remoteIpEndPoint = new IPEndPoint(IPAddress.Any, 0);
            byte[] receiveBytes = udpClient.Receive(ref remoteIpEndPoint);
            string returnData = Encoding.UTF8.GetString(receiveBytes);
            string message = remoteIpEndPoint.Address.ToString().Trim() + ": " + returnData.ToString().Trim();

            // Sử dụng phương thức Invoke để gọi hàm AddMessageToListView trên UI Thread
            rtbRecMess.Invoke(new Action(() => AddMessageToRichTextBox(message)));
        }
    }

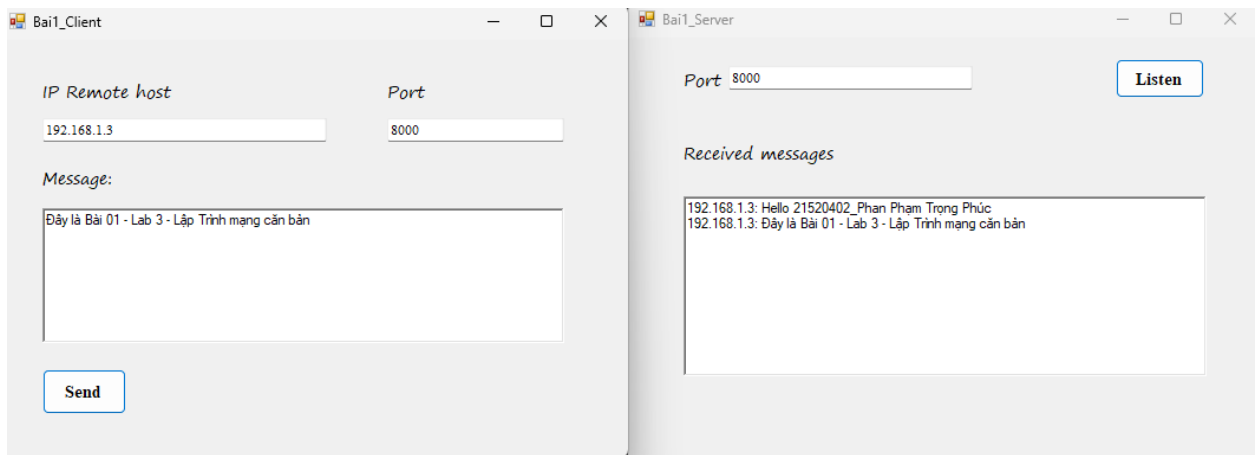
    1 reference
    private void AddMessageToRichTextBox(string message)
    {
        rtbRecMess.AppendText(message.Trim() + "\n");
    }
}

```

Demo:



Kiểm tra Port trước khi listen



Liên tục lắng nghe từ client

Bài 2:

Code:

```

namespace Server
{
    4 references
    public partial class Bai2 : Form
    {
        private TcpListener tcpListener;
        private TcpClient tcpClient;
        private NetworkStream networkStream;
        private byte[] receiveBuffer = new byte[1024];
        1 reference
        public Bai2()
        {
            InitializeComponent();
        }

        1 reference
        private void btnListen_Click(object sender, EventArgs e)
        {
            CheckForIllegalCrossThreadCalls = false;
            Thread serverThread = new Thread(new ThreadStart(StartUnsafeThread));
            serverThread.Start();
        }

        1 reference
        void StartUnsafeThread()
        {
            int port;
            if (!int.TryParse(tbPort.Text, out port))
            {
                MessageBox.Show("Vui lòng nhập Port đúng.");
                return;
            }

            if (tcpListener == null)
            {
                tcpListener = new TcpListener(IPAddress.Any, port);
                tcpListener.Start();

                rtbMessage.AppendText("Đang lắng nghe kết nối từ Telnet...\n");
                while (true)
                {
                    tcpListener.BeginAcceptTcpClient(HandleTcpClient, tcpListener);
                }
            }
        }
    }
}

```

Sử dụng thread để liên tục lắng nghe

```

    else
    {
        MessageBox.Show("Đã có kết nối đang được lắng nghe.");
    }
}

1 reference
private void HandleTcpClient(IAsyncResult ar)
{
    try
    {
        tcpClient = tcpListener.EndAcceptTcpClient(ar);
        networkStream = tcpClient.GetStream();

        string ip = ((IPEndPoint)tcpClient.Client.RemoteEndPoint).Address.ToString();

        rtbMessage.Invoke(new Action(() => rtbMessage.AppendText("Đã kết nối đến: " + ip + "\n")));

        Receive();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

2 references
private void Receive()
{
    networkStream.BeginRead(receiveBuffer, 0, receiveBuffer.Length, HandleReceive, networkStream);
}

1 reference
private void HandleReceive(IAsyncResult ar)
{
    try
    {
        int bytesRead = networkStream.EndRead(ar);

        if (bytesRead > 0)
        {

```

Khi kết nối thành công thì server sẽ hiển thị thông tin kết nối

```

private void HandleReceive(IAsyncResult ar)
{
    try
    {
        int bytesRead = networkStream.EndRead(ar);

        // Nếu nhận được dữ liệu thì
        if (bytesRead > 0)
        {
            string message = Encoding.UTF8.GetString(receiveBuffer, 0, bytesRead);

            // Hiển thị lên màn hình
            rtbMessage.Invoke(new Action(() => rtbMessage.AppendText(message)));

            Array.Clear(receiveBuffer, 0, receiveBuffer.Length);

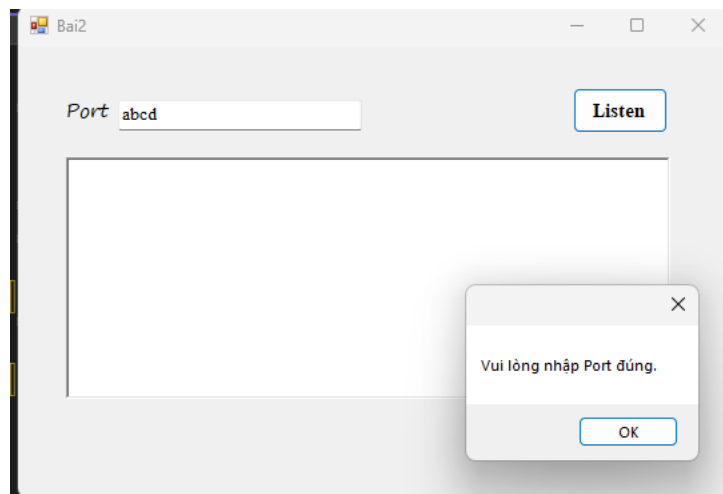
            Receive();
        }
        else // Kết nối đã bị đóng
        {
            string ip = ((IPEndPoint)tcpClient.Client.RemoteEndPoint).Address.ToString();
            rtbMessage.Invoke(new Action(() => rtbMessage.AppendText(ip + " đã ngắt kết nối.\n")));

            tcpClient.Close();
            tcpClient = null;
            networkStream.Close();
            networkStream = null;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

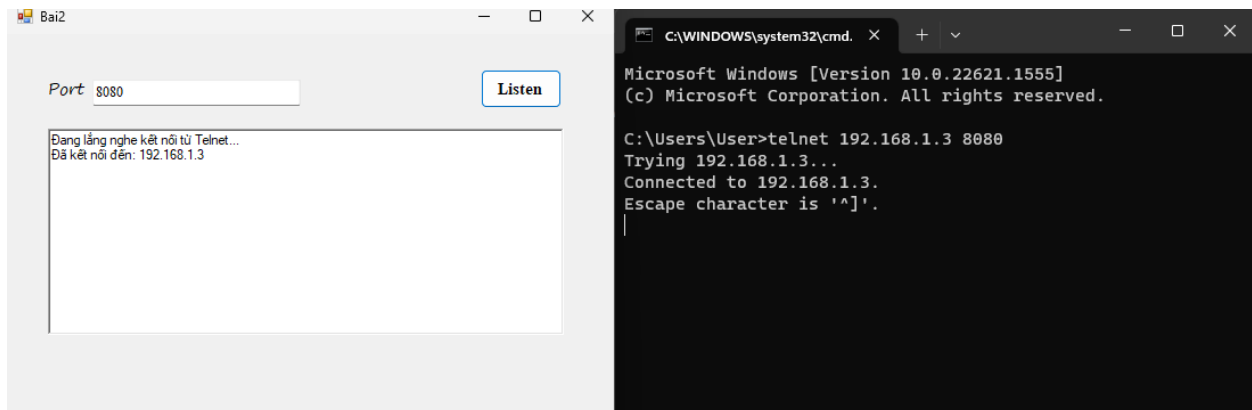
```

Hàm này nhận và kiểm tra dữ liệu để hiển thị thông tin

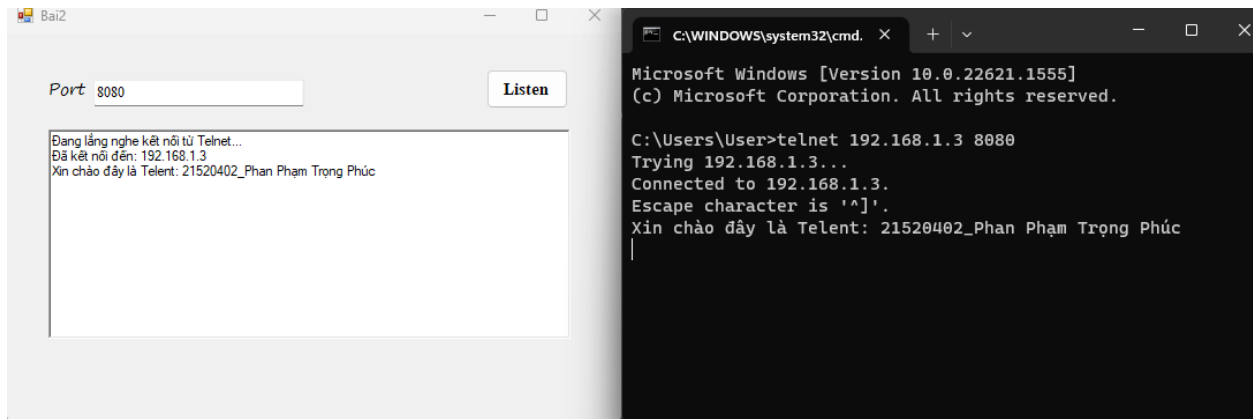
Demo:



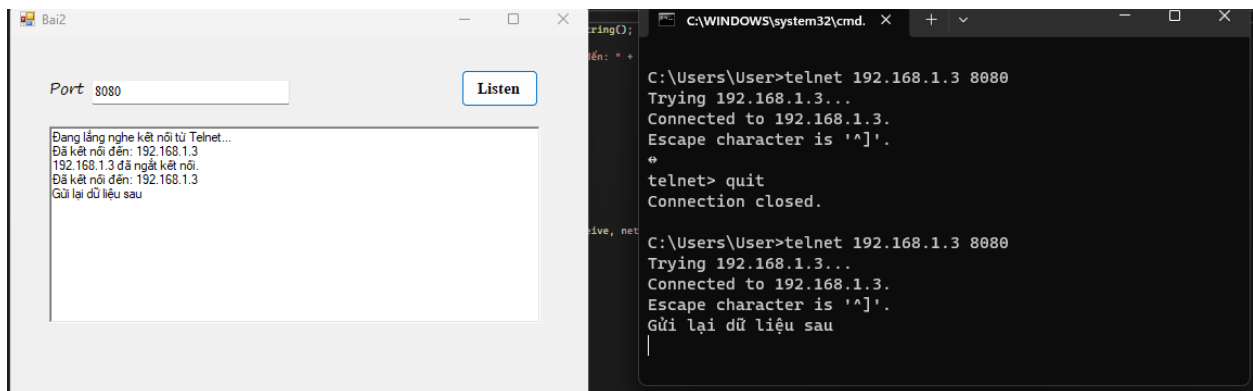
Kiểm tra Port trước khi lắng nghe



Kết nối thành công



Server nhận thông tin và hiển thị



Server liên tục lắng nghe các kết nối

Bài 3:

- Phía client:
- Code:

```

namespace Lab03
{
    4 references
    public partial class Bai3_Client : Form
    {
        private TcpClient client;
        private NetworkStream stream;
        1 reference
        public Bai3_Client()
        {
            InitializeComponent();
        }

        1 reference
        private void btnConnect_Click(object sender, EventArgs e)
        {
            try
            {
                // Tạo một kết nối tới server
                client = new TcpClient(tbIP.Text, int.Parse(tbPort.Text));
                stream = client.GetStream();

                // Thông báo trạng thái kết nối đến người dùng
                rtbSentMess.AppendText("Đã kết nối tới server " + tbIP.Text + ":" + tbPort.Text + "\n");
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }

        1 reference
        private void btnSend_Click(object sender, EventArgs e)
        {
            try
            {
                if (client == null || !client.Connected)
                {
                    MessageBox.Show("Chưa thiết lập kết nối tới server.");
                    return;
                }

                // Lấy dữ liệu từ richTextBox và gửi đi
                string message = rtbMess.Text;
                byte[] data = Encoding.UTF8.GetBytes(message);
                stream.Write(data, 0, data.Length);

                stream.Close();

                // Thông báo cho người dùng biết rằng dữ liệu đã được gửi
                rtbSentMess.AppendText("Đã gửi dữ liệu: " + message + "\n");
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }
    }
}

```

Sử dụng thread để liên tục lắng nghe

```

    else
    {
        MessageBox.Show("Đã có kết nối đang được lắng nghe.");
    }
}

1 reference
private void HandleTcpClient(IAsyncResult ar)
{
    try
    {
        tcpClient = tcpListener.EndAcceptTcpClient(ar);
        networkStream = tcpClient.GetStream();

        string ip = ((IPEndPoint)tcpClient.Client.RemoteEndPoint).Address.ToString();

        rtbMessage.Invoke(new Action() => rtbMessage.AppendText("Đã kết nối đến: " + ip + "\n"));

        Receive();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

2 references
private void Receive()
{
    networkStream.BeginRead(receiveBuffer, 0, receiveBuffer.Length, HandleReceive, networkStream);
}

1 reference
private void HandleReceive(IAsyncResult ar)
{
    try
    {
        int bytesRead = networkStream.EndRead(ar);

        if (bytesRead > 0)
        {

```


Khi kết nối thành công thì server sẽ hiển thị thông tin kết nối

```
private void HandleReceive(IAsyncResult ar)
{
    try
    {
        int bytesRead = networkStream.EndRead(ar);

        // Nếu nhận được dữ liệu thì
        if (bytesRead > 0)
        {
            string message = Encoding.UTF8.GetString(receiveBuffer, 0, bytesRead);

            // Hiển thị lên màn hình
            rtbMessage.Invoke(new Action(() => rtbMessage.AppendText(message)));

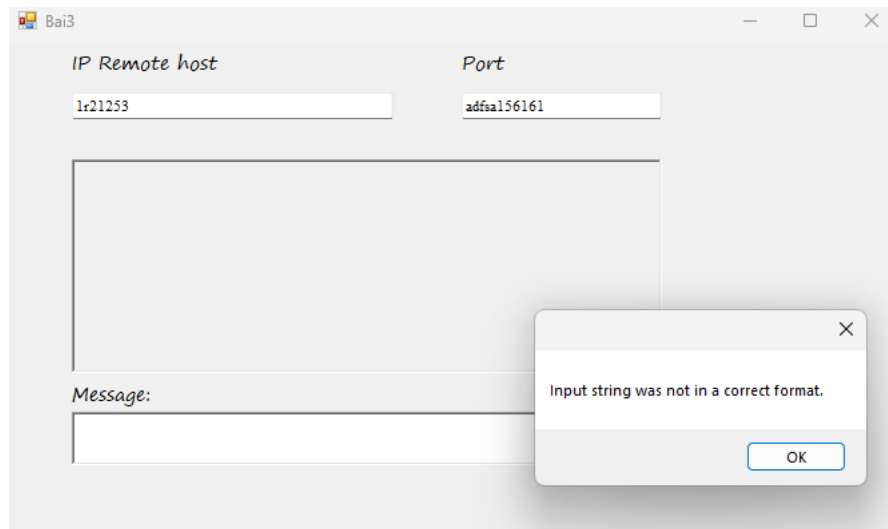
            Array.Clear(receiveBuffer, 0, receiveBuffer.Length);

            Receive();
        }
        else // Kết nối đã bị đóng
        {
            string ip = (((IPEndPoint)tcpClient.Client.RemoteEndPoint).Address.ToString());
            rtbMessage.Invoke(new Action(() => rtbMessage.AppendText(ip + " đã ngắt kết nối.\n")));

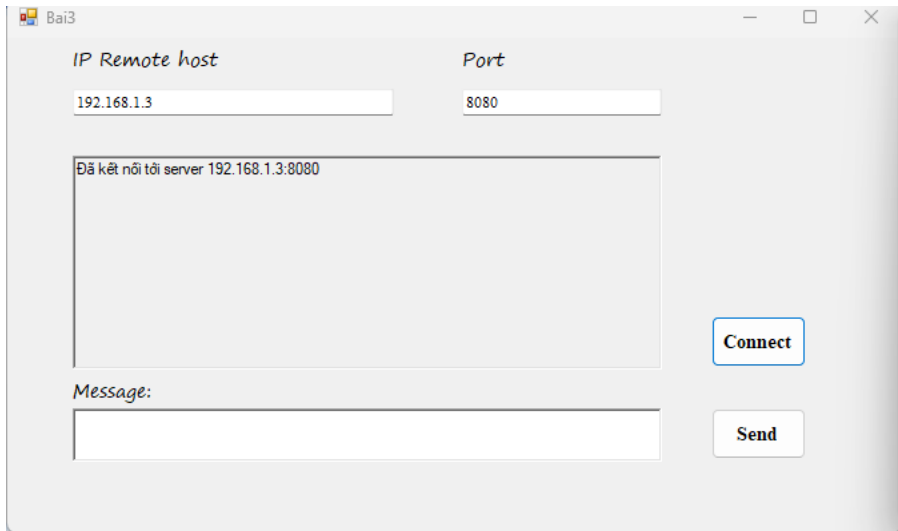
            tcpClient.Close();
            tcpClient = null;
            networkStream.Close();
            networkStream = null;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

Hàm này nhận và kiểm tra dữ liệu để hiển thị thông tin

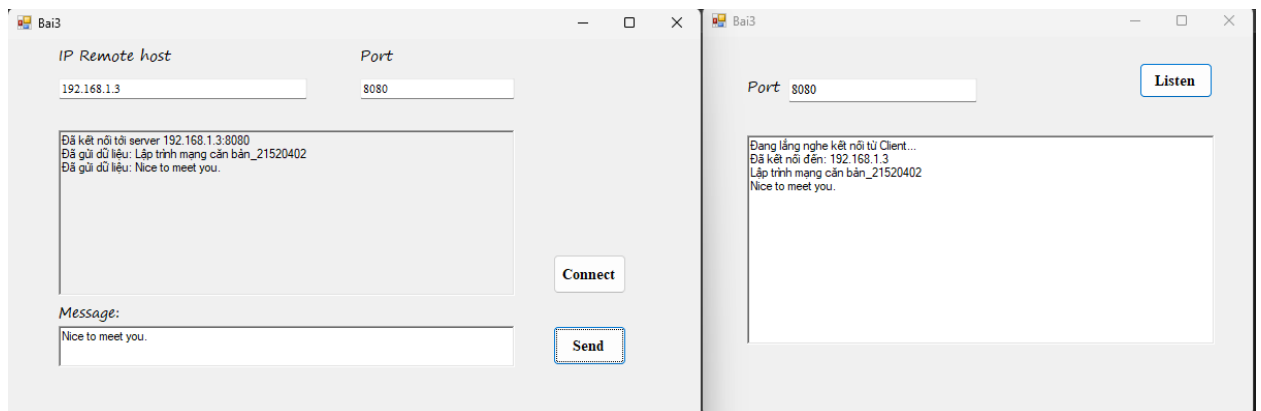
Demo:



Kiểm tra các trường dữ liệu



Kết nối thành công đến server



Gửi dữ liệu đến server

- Phía server:
Code:

```

namespace Lab03
{
    4 references
    public partial class Bai3_Client : Form
    {
        private TcpClient client;
        private NetworkStream stream;
        1 reference
        public Bai3_Client()
        {
            InitializeComponent();
        }

        1 reference
        private void btnConnect_Click(object sender, EventArgs e)
        {
            try
            {
                // Tạo một kết nối tới server
                client = new TcpClient(tbIP.Text, int.Parse(tbPort.Text));
                stream = client.GetStream();

                // Thông báo trạng thái kết nối đến người dùng
                rtbSentMess.AppendText("Đã kết nối tới server " + tbIP.Text + ":" + tbPort.Text + "\n");
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }

        1 reference
        private void btnSend_Click(object sender, EventArgs e)
        {
            try
            {
                if (client == null || !client.Connected)
                {
                    MessageBox.Show("Chưa thiết lập kết nối tới server.");
                    return;
                }

                // Lấy dữ liệu từ richTextBox và gửi đi
                string message = rtbMess.Text;
                byte[] data = Encoding.UTF8.GetBytes(message);
                stream.Write(data, 0, data.Length);

                stream.Close();

                // Thông báo cho người dùng biết rằng dữ liệu đã được gửi
                rtbSentMess.AppendText("Đã gửi dữ liệu: " + message + "\n");
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }
    }
}

```

Sử dụng thread để liên tục lắng nghe

```

    else
    {
        MessageBox.Show("Đã có kết nối đang được lắng nghe.");
    }
}

1 reference
private void HandleTcpClient(IAsyncResult ar)
{
    try
    {
        tcpClient = tcpListener.EndAcceptTcpClient(ar);
        networkStream = tcpClient.GetStream();

        string ip = ((IPEndPoint)tcpClient.Client.RemoteEndPoint).Address.ToString();

        rtbMessage.Invoke(new Action() => rtbMessage.AppendText("Đã kết nối đến: " + ip + "\n"));

        Receive();
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

2 references
private void Receive()
{
    networkStream.BeginRead(receiveBuffer, 0, receiveBuffer.Length, HandleReceive, networkStream);
}

1 reference
private void HandleReceive(IAsyncResult ar)
{
    try
    {
        int bytesRead = networkStream.EndRead(ar);

        if (bytesRead > 0)
        {

```

Khi kết nối thành công thì server sẽ hiển thị thông tin kết nối

```
private void HandleReceive(IAsyncResult ar)
{
    try
    {
        int bytesRead = networkStream.EndRead(ar);

        // Nếu nhận được dữ liệu thì
        if (bytesRead > 0)
        {
            string message = Encoding.UTF8.GetString(receiveBuffer, 0, bytesRead);

            // Hiển thị lên màn hình
            rtbMessage.Invoke(new Action(() => rtbMessage.AppendText(message)));

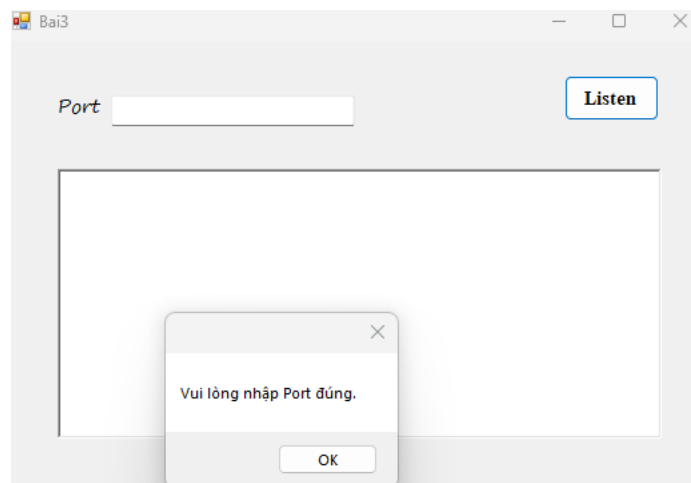
            Array.Clear(receiveBuffer, 0, receiveBuffer.Length);

            Receive();
        }
        else // Kết nối đã bị đóng
        {
            string ip = (((IPEndPoint)tcpClient.Client.RemoteEndPoint).Address.ToString());
            rtbMessage.Invoke(new Action(() => rtbMessage.AppendText(ip + " đã ngắt kết nối.\n")));

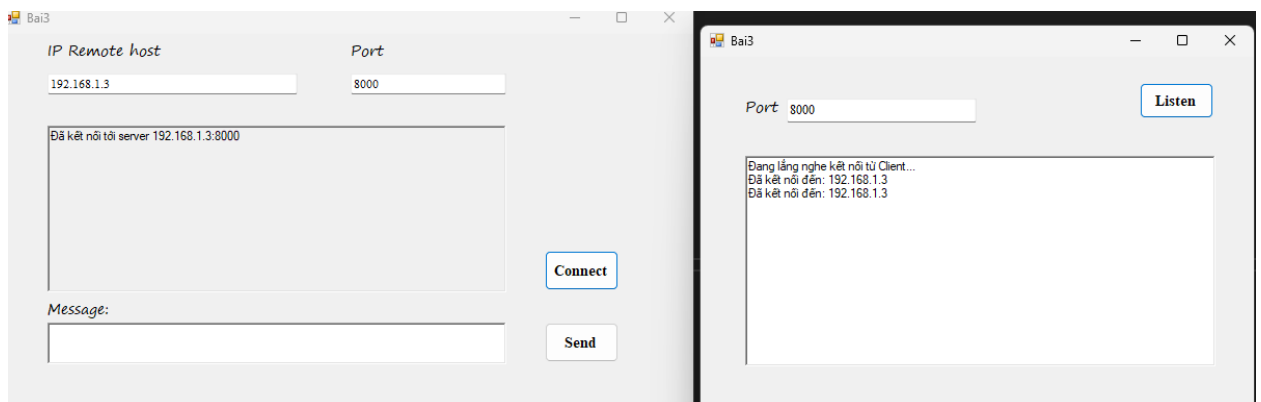
            tcpClient.Close();
            tcpClient = null;
            networkStream.Close();
            networkStream = null;
        }
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}
```

Hàm này nhận và kiểm tra dữ liệu để hiển thị thông tin

Demo:



Kiểm tra địa chỉ Port



Lắng nghe các kết nối

Bài 4:

- Phía clients:
Code:

```
namespace Lab03
{
    4 references
    public partial class Bai4 : Form
    {
        private TcpClient client;
        private NetworkStream stream;
        private string receivedData;

        1 reference
        public Bai4()
        {
            InitializeComponent();
        }

        1 reference
        private void btnConnect_Click(object sender, EventArgs e)
        {
            try
            {
                // Tao một kết nối tới server
                client = new TcpClient(tbIP.Text, int.Parse(tbPort.Text));
                stream = client.GetStream();

                // Khởi tạo thread nhận dữ liệu từ server
                Thread receiveThread = new Thread(new ThreadStart(ReceiveData));
                receiveThread.Start();
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }

        1 reference
        private void btnSend_Click(object sender, EventArgs e)
        {
            try
            {
                if (client == null || !client.Connected)
                {
                    MessageBox.Show("Chưa thiết lập kết nối tới server.");
                    return;
                }

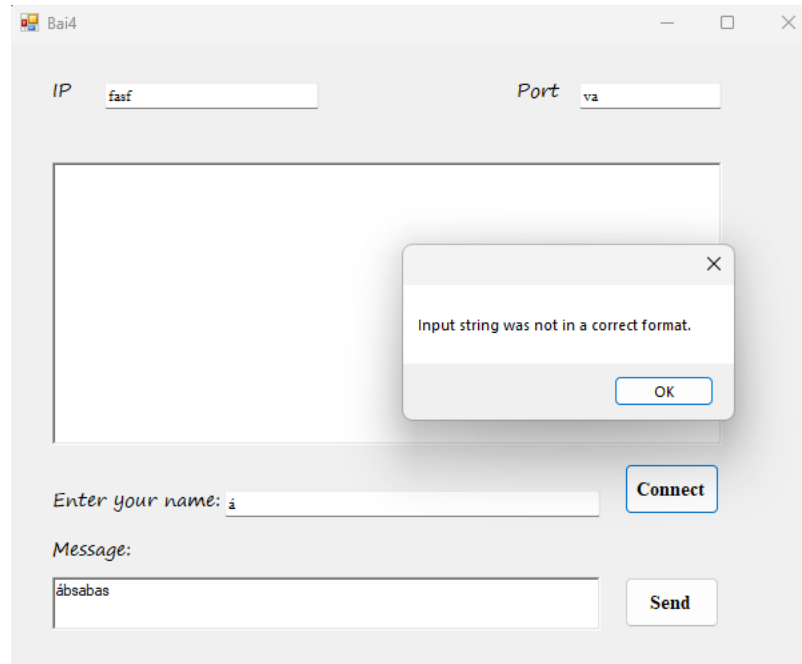
                // Lấy dữ liệu từ RichTextBox và gửi đi
                string message = tbUsr.Text + ": " + rtbMess.Text;
                byte[] data = Encoding.UTF8.GetBytes(message);
                stream.Write(data, 0, data.Length);

                // Lưu dữ liệu đã gửi đi vào biến receivedData
                receivedData = message;
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }

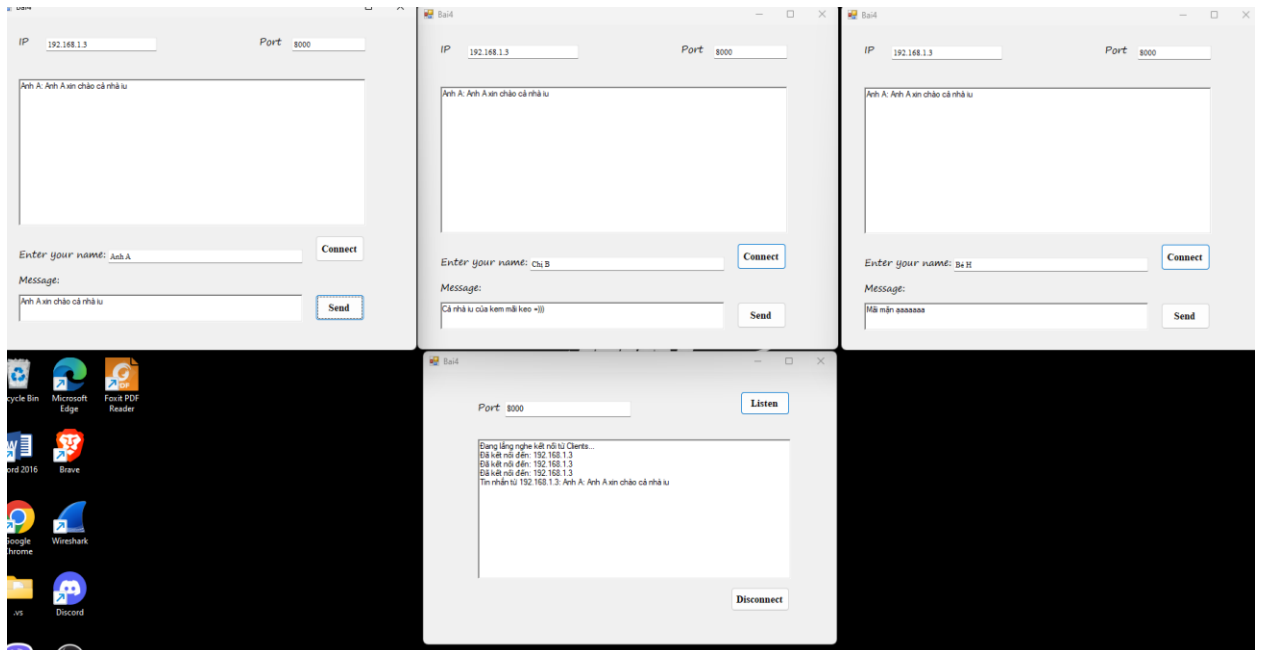
        1 reference
        private void ReceiveData()
        {
            try
            {
                while (true)
                {
                    // Đọc dữ liệu từ server và gửi lại cho các clients khác
                    byte[] buffer = new byte[1024];
                    int bytesRead = stream.Read(buffer, 0, buffer.Length);
                    string response = Encoding.UTF8.GetString(buffer, 0, bytesRead);
                    Invoke((MethodInvoker)delegate
                    {
                        rtbSentMess.AppendText(response + "\n");
                    });
                }
            }
            catch (Exception ex)
            {
                MessageBox.Show(ex.Message);
            }
        }
    }
}
```

Hàm để đọc dữ liệu từ server gửi về

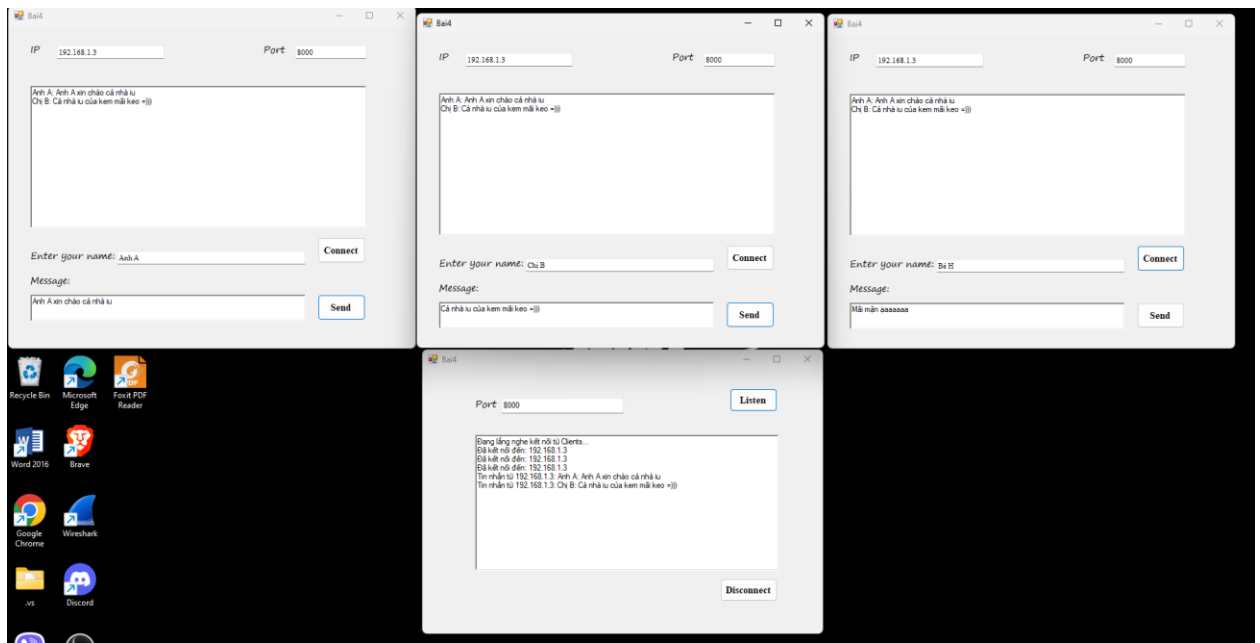
Demo:



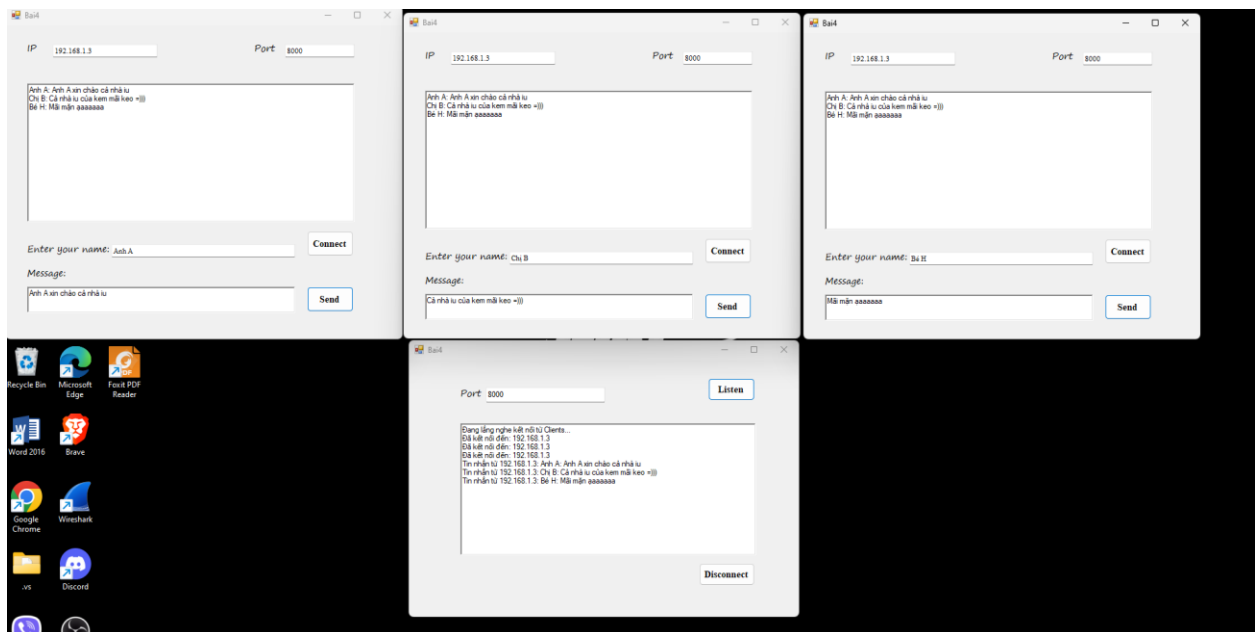
Kiểm tra các trường dữ liệu



Client thứ nhất gửi dữ liệu



Client thứ hai gửi dữ liệu



Client thứ ba gửi dữ liệu

- Phía server:
Code:

```
public partial class Bai4 : Form
{
    private static TcpListener tcpListener;
    private static readonly List<TcpClient> clients = new List<TcpClient>();

    1 reference
    public Bai4()
    {
        InitializeComponent();
    }
}
```

Dùng List<TcpClient> để lưu các clients kết nối

```

private void btnListen_Click(object sender, EventArgs e)
{
    CheckForIllegalCrossThreadCalls = false;
    Thread serverThread = new Thread(new ThreadStart(StartUnsafeThread));
    serverThread.Start();
}

1 reference
void StartUnsafeThread()
{
    int port;
    if (!int.TryParse(tbPort.Text, out port))
    {
        MessageBox.Show("Vui lòng nhập Port đúng.");
        return;
    }

    if (tcpListener == null)
    {
        tcpListener = new TcpListener(IPAddress.Any, port);
        tcpListener.Start();

        rtbMessage.AppendText("Đang lắng nghe kết nối từ Clients...\n");
        while (true)
        {
            tcpListener.BeginAcceptTcpClient(new AsyncCallback(HandleTcpClient), tcpListener);
        }
    }
    else
    {
        MessageBox.Show("Đã có kết nối đang được lắng nghe.");
    }
}

1 reference
private void HandleTcpClient(IAsyncResult ar)
{
    try
    {
        TcpListener listener = (TcpListener)ar.AsyncState;
        TcpClient client = listener.EndAcceptTcpClient(ar);
        string ip = ((IPEndPoint)client.Client.RemoteEndPoint).Address.ToString();

        rtbMessage.Invoke(new Action(() => rtbMessage.AppendText("Đã kết nối đến: " + ip + "\n")));

        clients.Add(client);

        // Tạo một thread mới để xử lý dữ liệu đến từ client
        Thread clientThread = new Thread(new ParameterizedThreadStart(HandleClientComm));
        clientThread.Start(client);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message);
    }
}

private void HandleClientComm(object clientObj)
{
    TcpClient client = (TcpClient)clientObj;
    NetworkStream stream = client.GetStream();

    byte[] buffer = new byte[1024];
    int bytesRead;

    while (true)
    {
        try
        {
            bytesRead = stream.Read(buffer, 0, buffer.Length);
            if (bytesRead > 0)
            {
                string message = Encoding.UTF8.GetString(buffer, 0, bytesRead);
                rtbMessage.Invoke(new Action(() => rtbMessage.AppendText("Tin nhắn từ " + ((IPEndPoint)client.Client.RemoteEndPoint).Address.ToString() + ": " + message + "\n")));
                BroadcastMessage(message);
            }
        }
        catch
        {
            // Kết nối đã bị đóng
            clients.Remove(client);
            rtbMessage.Invoke(new Action(() => rtbMessage.AppendText(((IPEndPoint)client.Client.RemoteEndPoint).Address.ToString() + " đã ngắt kết nối.\n")));
            break;
        }
    }
}

1 reference
private void BroadcastMessage(string message)
{
    byte[] buffer = Encoding.UTF8.GetBytes(message);

    foreach (TcpClient client in clients)
    {
        NetworkStream stream = client.GetStream();
        stream.Write(buffer, 0, buffer.Length);
    }
}

```

Quản lý các clients