**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA CÔNG NGHỆ PHẦN MỀM**



BÁO CÁO ĐỒ ÁN MÔN HỌC

**LẬP TRÌNH TRỰC QUAN**

TÊN ĐỀ TÀI

**Môn học : Lập trình trực quan**

**Giảng viên lý thuyết : Phan Nguyệt Minh**

**Giảng viên thực hành : Huỳnh Hồ Thị Mộng Trinh**

**Nhóm thực hiện : Hồ Thái Ngọc - 16520825**

**Phạm Thị Hoàng Mai - 16520717**

**Nguyễn Công Minh - 16520740**

**Phan Vĩnh Long - 16520695**

***TP. Hồ Chí Minh, tháng 6 năm 2018***

**LỜI CẢM ƠN**

Trong suốt quá trình học tập môn **Lập trình trực quan** và hoàn thành đồ án **Xây dựng phần mềm Lan Chat**, nhóm đã nhận được những kiến thức vô cùng bổ ích từ **Cô Phan Nguyệt Minh,** **Cô Huỳnh Hồ Thị Mộng Trinh** và  **Thầy Phạm Thi Vương**. Thông qua việc hoàn thành đồ án, nhóm chúng em xin được gửi lời cám ơn đến các thầy cô vì sự tận tâm và vì những gì mà thầy cô đã truyền đạt cho chúng em.

Trân trọng.

TP. Hồ Chí Minh, ngày 18 tháng 06 năm 2018

NHÓM THỰC HIỆN

**NHẬN XÉT CỦA GIẢNG VIÊN**

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

..........................................................................................................................

TP Hồ Chí Minh, ngày 18 tháng 6 năm 2018

GIẢNG VIÊN HƯỚNG DẪN

**MỤC LỤC**

[MỞ ĐẦU 5](#_Toc517065932)

[I. GIỚI THIỆU ĐỀ TÀI 6](#_Toc517065933)

[1. Lý do chọn đề tài 6](#_Toc517065934)

[2. Mục đích nghiên cứu đề tài 6](#_Toc517065935)

[3. Phạm vi nghiên cứu đề tài 6](#_Toc517065936)

[4. Công nghệ sử dụng 6](#_Toc517065937)

[II. KIẾN TRÚC CHƯƠNG TRÌNH 7](#_Toc517065938)

[III. GIAO DIỆN CHƯƠNG TRÌNH VÀ CHỨC NĂNG CHÍNH 8](#_Toc517065939)

[1. Màn hình giao diện chính 8](#_Toc517065940)

[2. Thanh công cụ 9](#_Toc517065941)

[***2.1. Messenger*** 9](#_Toc517065942)

[***2.2. Extend*** 10](#_Toc517065943)

[***2.3. About*** 10](#_Toc517065944)

[3. Các chức năng chính: 10](#_Toc517065945)

[4. Hiện thực các chức năng chính: 11](#_Toc517065946)

[***4.1. Server:*** 11](#_Toc517065947)

[- Hashtable trong C# sử dụng một cặp **key-value** để truy cập các phần tử trong collection này. 11](#_Toc517065948)

[***4.2. Gửi text*** 12](#_Toc517065949)

[***4.3. Cho phép thao tác trên danh bạ bằng menu:*** 15](#_Toc517065950)

[***4.4. Thông báo trạng thái của user (Online/Offline/Invisible)*** 17](#_Toc517065951)

[***4.5. Chat group:*** 18](#_Toc517065952)

[***4.6. Hỗ trợ xem lịch sử chat*** 20](#_Toc517065953)

[***4.7. Gửi file*** 21](#_Toc517065954)

[***4.8. Show webcam*** 21](#_Toc517065955)

[***4.9. Hỗ trợ các tùy chỉnh cho người dùng*** 23](#_Toc517065956)

[5. Các chức năng nổi bật: 25](#_Toc517065957)

[***5.1. Transparent form:*** 25](#_Toc517065958)

[***5.2. Chơi game Caro*** 26](#_Toc517065959)

[IV. CHƯƠNG CÀI ĐẶT VÀ KIỂM THỬ 29](#_Toc517065960)

[1. Môi trường 29](#_Toc517065961)

[2. Ngôn ngữ cài đặt 29](#_Toc517065962)

[3. Thử nghiệm 29](#_Toc517065963)

[4. Đánh giá kết quả 29](#_Toc517065964)

[V. CHƯƠNG KHÁC 30](#_Toc517065965)

[1. Thuận lợi và khó khăn 30](#_Toc517065966)

[***1.1. Thuận lợi:*** 30](#_Toc517065967)

[***1.2. Khó khăn:*** 30](#_Toc517065968)

[2. Phân công công việc 30](#_Toc517065969)

[3. Hướng phát triển 31](#_Toc517065970)

[4. Tài liệu tham khảo 31](#_Toc517065971)

# **MỞ ĐẦU**

Trong bối cảnh hiện nay, ngành công nghệ thông tin đóng vai trò rất quan trọng vì nó xuất hiện hầu hết ở mọi lĩnh vực như kinh tế, giáo dục, chính trị,… Giúp con người giải quyết công việc nhẹ nhàng hơn cũng giúp mọi người có thể kết nối với nhau dễ dàng hơn. Hiện nay có rất nhiều ứng dụng, phần mềm chat, giúp mọi người có thể dễ dàng kết bạn và trao đổi, nói chuyện với nhau về sở thích bản thân, tâm sự với nhau những lời mà khi gặp mặt trực tiếp vẫn chưa thể nói hết.

Nhóm chúng em đã quyết định chọn đề tài ***Xây dựng phần mềm LanChat*** để tìm hiểu và nắm rõ về cách một tin nhắn được gửi, ứng dụng được ngôn ngữ C# trong việc hiện thực một đồ án hoàn chỉnh. Do kiến thức còn hạn hẹp và thời gian không nhiều, đây cũng là lần đầu nhóm thực hiện một đồ án nên không thể tránh khỏi những sai sót, chúng em rất hy vọng thầy cô sẽ góp ý để nhóm có thể hoàn thiện hơn.

# **I. GIỚI THIỆU ĐỀ TÀI**

1. **Lý do chọn đề tài**

Ngày nay, việc gửi thông tin qua Internet còn mang nhiều rủi ro, không đảm bảo an ninh trong khi đó đối với các công ty hay phòng ban nhỏ trong một bộ phận của một tổ chức, việc bảo mật thông tin lại mang tính chất sống còn như: thông tin về một dự án mới, những ý tưởng kinh doanh mới hay những thông tin mang tính bảo mật cao không thể để đối thủ nắm được. Việc trao đổi giữa các thành viên trong tổ chức khi thảo luận vấn đề hay làm việc với nhau là không thể tránh khỏi, nhưng trao đổi qua Internet lại quá mạo hiểm, trao đổi theo cách truyền thống lại mất thời gian. Vậy cách giải quyết cho vấn đề này là gì? Ta cần tạo cho họ một môi trường mà chỉ những người trong cùng một mạng cục bộ mới có thể kết nối với nhau để họ có thể dễ dàng trao đổi, nói chuyện. Chính từ suy nghĩ này, nhóm đã quyết định chọn đề tài ***Xây dựng phần mềm Lan Chat*** để giúp đỡ giải quyết vấn đề trên

1. **Mục đích nghiên cứu đề tài**

Đề tài ***Xây dựng phần mềm LanChat*** được nhóm chọn nhằm hiểu được cách giao tiếp giữa các cá thể trong cùng một mạng Lan và phát triển các tính năng mở rộng

1. **Phạm vi nghiên cứu đề tài**

Giới hạn trong những xử lý cơ bản của một phần mềm Chat có thể làm được. Hỗ trợ thêm các tính năng khác: Chat Webcam, gửi file, gửi hình ảnh, chơi game.

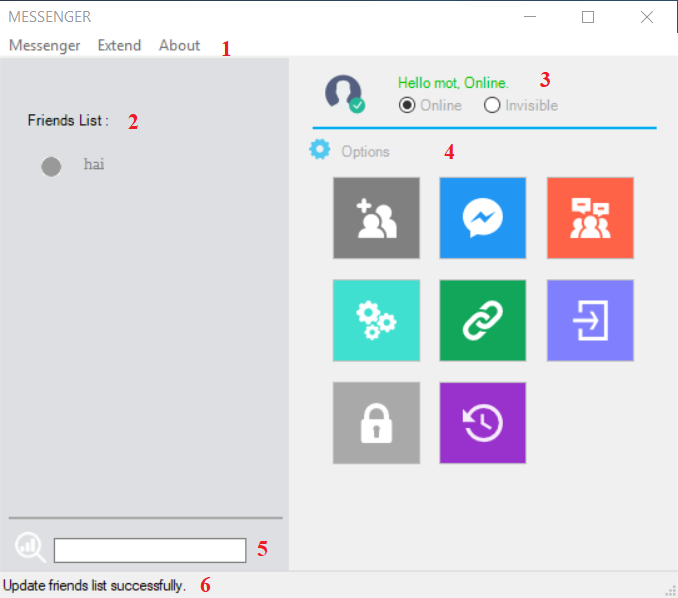
1. **Công nghệ sử dụng**

Phần mềm sử dụng .NET Framework 4. ,Bunifu Framework và được code trên Visual Studio 2015, Visual Studio 2017

# **II. KIẾN TRÚC CHƯƠNG TRÌNH**

# **III. GIAO DIỆN CHƯƠNG TRÌNH VÀ CHỨC NĂNG CHÍNH**

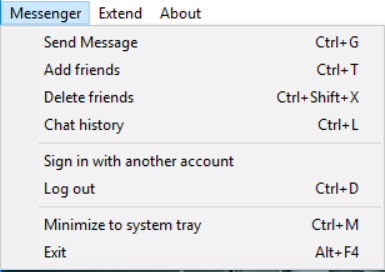
## **Màn hình giao diện chính**



|  |  |  |
| --- | --- | --- |
| STT | Tên gọi | Chức năng |
| 1 | Thanh công cụ | Cung cấp các chức năng đối với người dùng |
| 2 | Danh sách bạn bè | Hiển thị danh sách bạn bè |
| 3 | Tên và trạng thái người dùng | Hiển thị tên và trạng thái hiện tại của người dùng |
| 4 | Thanh công cụ truy cập nhanh | Các nút tắt và một số chức năng khác |
| 5 | Thanh tìm kiếm | Tìm kiếm bạn bè trong danh sách |
| 6 | Thanh trạng thái | Thông báo cập nhật danh sách bạn bè thành công |

## **2. Thanh công cụ**

### ***2.1. Messenger***

**Send Message:** Chọn người và bắt đầu chat

**Add friends:** Thêm một người bạn mới vào danh sách bạn bè

**Delete friends:** Xóa một người bạn khỏi danh sách bạn bè

**Chat history:** Xem lịch sử trò chuyện

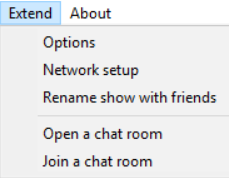
**Sign in with another account:** Đăng nhập bằng một tài khoản khác

**Log out:** Đăng xuất khỏi tài khoản hiện tại

**Minimize to system tray:** Thu nhỏ ứng dụng vào khay hệ thống

**Exit:** Thoát khỏi chương trình

### ***2.2. Extend***

**Option:** Xem các cài đặt của người dùng

**Network setup:** Chọn cài đặt mạng

**Rename show with friends:** Đổi tên hiển thị với bạn bè

**Open a chat room:** Tạo một phòng chat

**Join a chat room:** Tham gia một phòng chat

### ***2.3. About***

**Information:** Các thông tin về ứng dụng Lan Chat

## **3. Các chức năng chính:**

Các chức năng chính của chương trình là:

* Có server quản lí client
* Có thể chat qua mạng LAN với nội dung text
* Cho phép thao tác danh bạ bằng menu
* Thông báo trạng thái của user (Online/Offline/Invisible)
* Hỗ trợ chat group
* Hiển thị lịch sử chat
* Hỗ trợ gửi file
* Hỗ trợ xem webcam
* Hỗ trợ các tùy chỉnh cho người dùng

## **4. Hiện thực các chức năng chính:**

### ***4.1. Server:***

* Hashtable trong C# sử dụng một cặp **key-value** để truy cập các phần tử trong collection này.
* Phương thức và Thuộc tính của lớp Hashtable trong C#
* Bảng dưới liệt kê các **thuộc tính** được sử dụng phổ biến của lớp **Hashtable** trong C#:

|  |  |
| --- | --- |
| **Thuộc tính** | **Miêu tả** |
| Count | Lấy số cặp key và value được chứa trong Hashtable |
| IsFixedSize | Lấy một giá trị chỉ rằng có hay không Hashtable là có một kích cỡ cố định |
| IsReadOnly | Lấy một giá trị chỉ rằng có hay không Hashtable là read-only |
| Item | Lấy hoặc thiết lập giá trị được liên kết với key đã xác định |
| Keys | Lấy một ICollection chứa các key trong Hashtable |
| Values | Lấy một ICollection chứa các value trong Hashtable |

* Bảng dưới liệt kê các **phương thức** được sử dụng phổ biến của lớp **Hashtable** trong C#:

|  |  |
| --- | --- |
| **STT** | **Phương thức** |
| 1 | **public virtual void Add(object key, object value);**  Thêm một phần tử với key và value đã xác định vào trong Hashtable |
| 2 | **public virtual void Clear();**  Gỡ bỏ tất cả phần tử từ Hashtable |
| 3 | **public virtual bool ContainsKey(object key);**  Xác định có hay không Hashtable là chứa một key cụ thể |
| 4 | **public virtual bool ContainsValue(object value);**  Xác định có hay không Hashtable là chứa một value cụ thể |
| 5 | **public virtual void Remove(object key);**  Gỡ bỏ phần tử với key đã xác định từ Hashtable |

* Ứng dụng phương thức Hashtable

// Thêm một người dùng mới

internal bool Add(string username,string password,string fullName, string IP)

{

if(database.Contains(username))

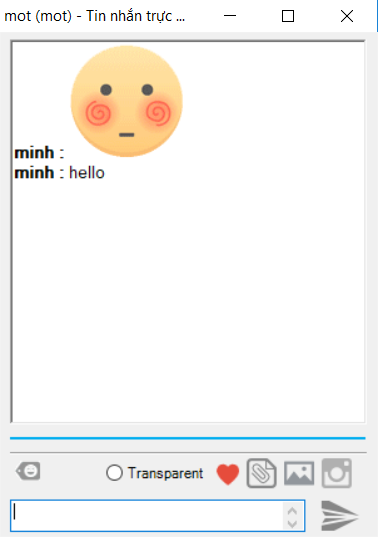
return false;

database.Add(username,new AccountField(password,false,fullName, IP));

return Save();

}

### ***4.2. Gửi text***



* Lọc địa chỉ IP từ máy khác

// Lọc địa chỉ IP từ Message

public string GetIPofContact(string message)

{

string pattern = @"(([0-1]?[0-9]{1,2}\.)|(2[0-4][0-9]\.)|(25[0-

5]\.)){3}(([0-1]?[0-9]{1,2})|(2[0-4][0-9])|(25[0-5]))";

Regex check = new Regex(pattern);

Match m = check.Match(message);

if (m.Success) return m.Value;

return "Error";

}

* Kiểm tra nội dung text khác null

// Khi txtMessage không chứa chữ thì làm mở nút Gửi (ko thể gửi tn được)

private void txtMessage\_TextChanged(object sender, System.EventArgs e)

{

if (txtMessage.Text == "")

btnSend.Enabled = false;

else

btnSend.Enabled = true;

}

* Định dạng font chữ nội dung của text, sau đó tìm kiếm fullName của user trong sever và đưa nội dung text ra MessageBox của người nhận. Đồng thời lưu lịch sử chat vào logs

rtbConversation.SelectionFont=new System.Drawing.Font("Microsoft Sans Serif",9.75f,System.Drawing.FontStyle.Bold);

rtbConversation.AppendText(Global.server.GetfullName(person)+" : ");

rtbConversation.SelectionFont=new System.Drawing.Font("Microsoft Sans Serif",9.75f,System.Drawing.FontStyle.Regular);

rtbConversation.AppendText(message+" \n");

CharactertoIcon(rtbConversation);

txtMessage.Focus();

if (setting[5] == "1" || setting[5] == "2")

WriteLogs(person+" : " + message); // Ghi logs của mình nhận

* Để xử lí thêm trường hợp không cho phép gửi tin nhắn tới “tài khoản không tồn tại”

Global.server.Send(Global.username, contact, txtMessage.Text); // Có thể xử lý thêm trường hợp ko cho phép gửi tin nhắn tới "tài koản ko tồn tại" ở đây.

txtMessage.Focus();

txtMessage.Clear();

* Sử dụng hàm public bool Send(string from,string to,string message) để hoàn thành việc gửi tin nhắn

public bool Send(string from,string to,string message)

{

// Kiểu tra sự tồn tại của đối tượng gửi.

if(database.ContactExists(to)==false)

return false;

// Nếu bạn Chat đã đăng nhập thì gửi tin nhắn trực tiếp (kể cả ẩn nick)

if(database.IsLoggedIn(to))

letters.Add(new Letter(from,to,message));

else

{

// Ngược lại, gửi tin nhắn Offline

database.offlineMessages.Add(new Letter(from,to,message));

if(database.offlineMessages.Count>maxOfflineMessages)

{

database.offlineMessages.RemoveAt(0);

}

//Console.WriteLine(database.Save());

Console.WriteLine(DateTime.Now.ToString() + ": Da gui tin nhan Offline

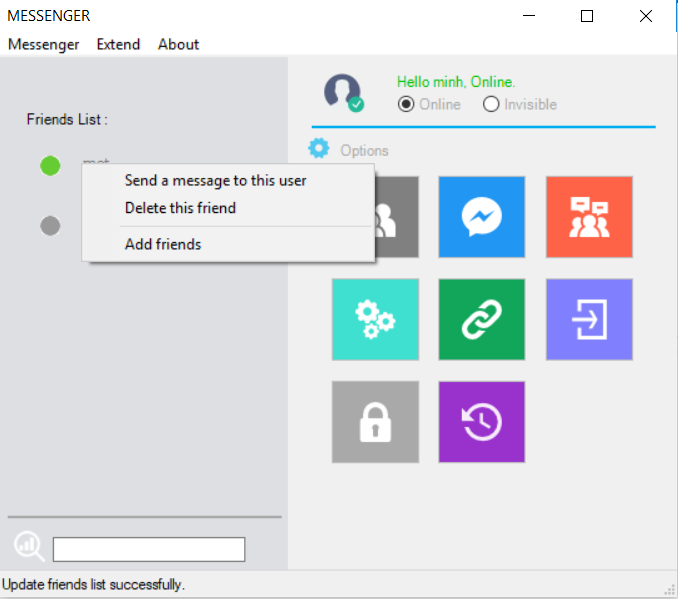
toi " + to);

}

return true;

}

### ***4.3. Cho phép thao tác trên danh bạ bằng menu:***



* Gửi một tin nhắn cho người bạn

private void menuSendMessage\_Click(object sender, System.EventArgs e)

{

using(FormSelectContact frmContact = new FormSelectContact())

{

if(frmContact.ShowDialog(this)==DialogResult.OK)

{

if(frmContact.txtContact.Text==Global.username)

{

MessageBox.Show("Bạn không thể gửi tin nhắn tới chính bạn được!");

return;

} if(Global.windowList.Contains(frmContact.txtContact.Text))

{

((FormMessage)Global.windowList[frmContact.txtContact.Text]).Focus();

}

else

{

FormMessage frmMessage = new FormMessage(); frmMessage.contact=frmContact.txtContact.Text;

frmMessage.Text=frmContact.txtContact.Text+" - Tin nhắn trực tiếp."; Global.windowList.Add(frmContact.txtContact.Text,frmMessage);

frmMessage.Show();

}

}

}

}

* Xóa bạn bè khỏi danh sách bạn

private void menuRemoveContact\_Click(object sender, System.EventArgs e)

{

FormSelectContact frmSelectContact = new FormSelectContact();

if (frmSelectContact.ShowDialog() == DialogResult.OK)

{

try

{

Global.server.RemoveContact(Global.username,

frmSelectContact.txtContact.Text);

UpdatePanelContact();

}

catch

{

AbortConnection();

return;

}

}

}

* Thêm bạn bè mới vào danh sách bạn

private void menuAddContact\_Click(object sender, System.EventArgs e)

{

FormAddContact frmAddContact = new FormAddContact();

if(frmAddContact.ShowDialog()==DialogResult.OK)

{

UpdatePanelContact();

}

}

### ***4.4. Thông báo trạng thái của user (Online/Offline/Invisible)***

* Thông báo trạng thái Online: Hiển thị thông báo ở góc trái bên trên màn hình



// Thông báo tên 1 Contact Online

private void GoOnline(string ContactName)

{

TrayBalloon.TrayBalloon tb = new TrayBalloon.TrayBalloon();

tb.BackgroundLocation = "background.bmp";

if (File.Exists(setting[1]))

tb.SoundLocation = setting[1];

else

tb.SoundLocation = "sounds/Online.wav";

tb.Title = "Lan Messenger!";

tb.Message = Global.server.GetfullName(ContactName) + " Online!";

tb.Run();

}

* Thông báo trạng thái Offline:



// Thông báo tên 1 Contact Offline

private void GoOffline(string ContactName)

{

TrayBalloon.TrayBalloon tb = new TrayBalloon.TrayBalloon();

tb.BackgroundLocation = "background.bmp";

if (File.Exists(setting[2]))

tb.SoundLocation = setting[2];

else

tb.SoundLocation = "sounds/Offline.wav";

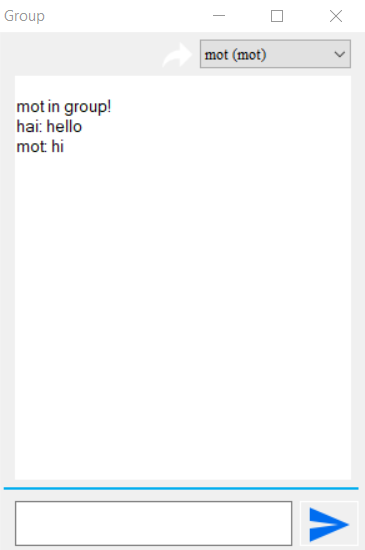
tb.Title = "Lan Messenger!";

tb.Message = Global.server.GetfullName(ContactName) + " Offline!";

tb.Run();

}

### ***4.5. Chat group:***



* Sử dụng phương thức TCP port 7070 để khởi tạo kênh chat

private void btn\_group\_Click(object sender, EventArgs e)

{

// Khởi tạo kênh liên lạc với remote Object.

channelChatRoom = new TcpChannel(7070);

ChannelServices.RegisterChannel(channelChatRoom, false);

// Ở đây sẻ sử dụng Singleton để duy trì trạng thái với nhiều kết nối Client

RemotingConfiguration.RegisterWellKnownServiceType(typeof(SampleObject),

Global.username.ToString(), WellKnownObjectMode.Singleton); // Cho Server chat

room

// Mở Room

if (OpenRoom() == 0)

{

ChannelServices.UnregisterChannel(channelChatRoom);

channelChatRoom = null;

}

}

* Dùng hàm OpenRoom để mở group chat

private int OpenRoom()

{

ArrayList alOnlineUser = new ArrayList();

FormChatRoom objChatWin;

if (true)

{

chan = new TcpChannel();

//ChannelServices.RegisterChannel(chan, false);

chan = null;

IPHostEntry temp = Dns.GetHostByName(Dns.GetHostName().ToString());

string IP = temp.AddressList[0].ToString();

objChatWin = new FormChatRoom();

objChatWin.MyGetData = new FormChatRoom.GetString(GetValue);

objChatWin.remoteObj = (SampleObject)Activator.GetObject(typeof(LanMessengerChatRoomBase.SampleObject), "tcp://"+IP+":7070/" + Global.username);

if (!objChatWin.remoteObj.JoinToChatRoom(Global.username))

{

MessageBox.Show(Global.username + " đã đăng nhập rồi!. Có thể

mạng bị lag, hãy thử lại sau!");

ChannelServices.UnregisterChannel(chan);

chan = null;

objChatWin.Dispose();

return 1; // 1: da dang nhap roi

}

objChatWin.key = objChatWin.remoteObj.CurrentKeyNo();

objChatWin.yourName = Global.username;

objChatWin.Show();

return 2; // 2 : Dang nhap cong

}

else

{

MessageBox.Show("Đã có lỗi xảy ra khi tạo Room Chat, vui lòng

thử lại!");

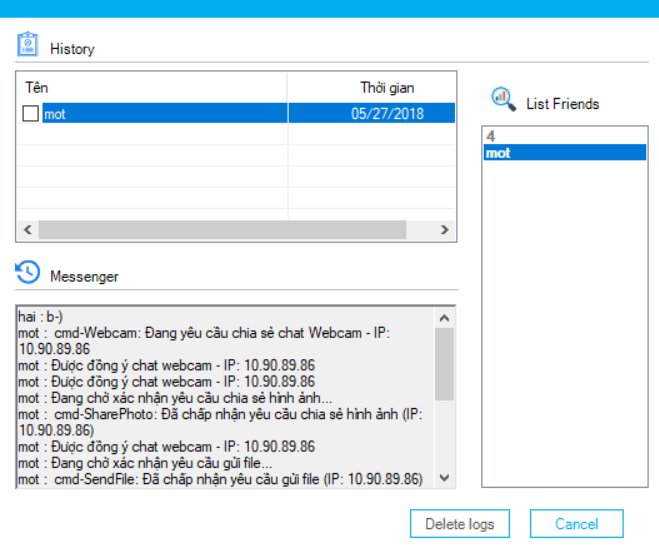
chan = null;

return 0; // 0: Loi

}

}

### ***4.6. Hỗ trợ xem lịch sử chat***



* Ghi lịch sử chat bằng file .dat

// Lọc tên file logs chat của 1 Contact xác định

private string GetFileContact(string path)

{

string pattern = "[0-9]+.dat";

Regex filename = new Regex(pattern);

Match m = filename.Match(path);

if (m.Success)

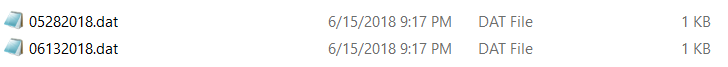
{

return m.Value;

}

return "";

}



### ***4.7. Gửi file***

* Gửi dữ liệu với kiểu dữ liệu bất kì bằng socket

public bool Send(Object data)

{

byte[] sendData = SerializeData(data);

return SendData(client, sendData);

}

private bool SendData(Socket target, byte[] data)

{

return target.Send(data) == 1 ? true : false;

}

### ***4.8. Show webcam***

* Để chat webcam thì đầu tiên cần phải bật webcam

private void Start\_Webcam()

{

try

{

if (capture != null)

{

if (capture.PreviewWindow != panelVideo)

{

capture.PreviewWindow = panelVideo;

}

capture.FrameEvent2 += new Capture.HeFrame(CaptureDone);

capture.GrapImg();

isSending = true;

}

}

catch (Exception) { }

}

* Người gửi khởi tạo server và mở port để quá trình chat webcam có thể bắt đầu

// Khởi tạo server, mở port - Vai trò là người gửi

Socket server\_sock;

void server()

{

try

{

server\_sock = new Socket(AddressFamily.InterNetwork, SocketType.Stream,

ProtocolType.Tcp);

server\_sock.Bind(new IPEndPoint(IPAddress.Any, 6000));

server\_sock.Listen(-1);

while (true)

{

try

{

Socket new\_socket = server\_sock.Accept();

NetworkStream ns = new NetworkStream(new\_socket);

pictureBox\_Remote.Image = Image.FromStream(ns);

ns.Close();

new\_socket.Close();

}

catch (Exception)

{

}

}

}

catch (Exception) { }

}

* Hàm giúp gửi hình ảnh trong bộ nhớ đệm đi

// Hàm gửi hình ảnh trong bộ nhớ đệm đi.

void SendVideoBuffer(object bufferIn)

{

try

{

TcpClient tcp = new TcpClient(IP, 6000);

NetworkStream ns = tcp.GetStream();

Image buffer = (Image)bufferIn;

buffer.Save(ns, System.Drawing.Imaging.ImageFormat.Jpeg);

ns.Close();

tcp.Close();

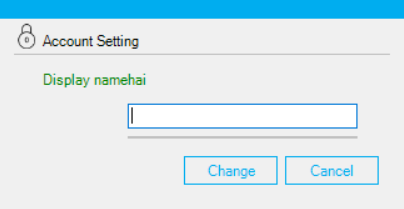
}

catch (Exception) { }

}

### ***4.9. Hỗ trợ các tùy chỉnh cho người dùng***

* Đổi tên hiển thị với bạn bè:



// Thay đổi tên hiển thị

internal bool ChangeDisplayName(string username, string displayName)

{

if (database.Contains(username))

{

((AccountField)database[username]).fullName = displayName;

return true;

}

else

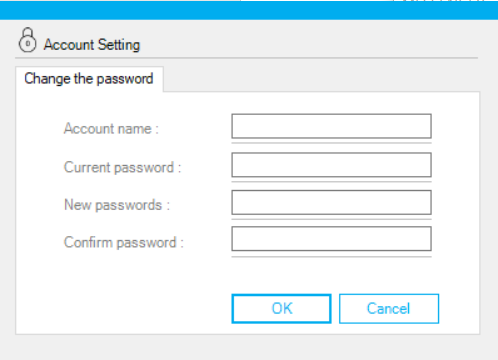
{

return false;

}

}

* Đổi mật khẩu người dùng:



// Thay đổi mật khẩu người dùng

internal bool ChangePassword(string username,string curPassword,string newPassword)

{

if(database.Contains(username))

{

if(((AccountField)database[username]).password==curPassword)

{

((AccountField)database[username]).password=newPassword;

Save();

return true;

}

else

{

return false;

}

}

else

{

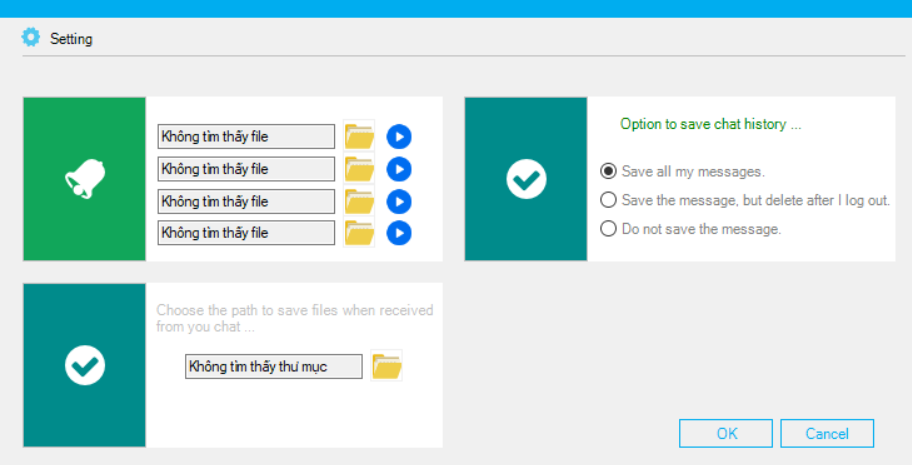
return false;

}

}



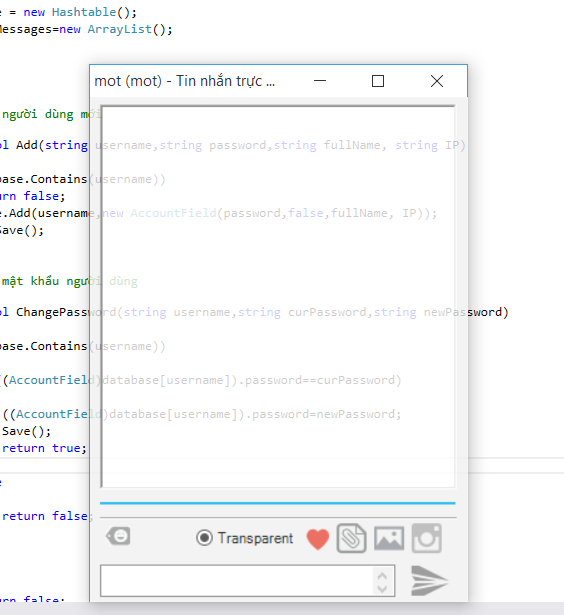
* Ngoài ra còn một số tùy chỉnh khác như tùy chỉnh âm thanh (thông báo online, offline, chat,…)



## **5. Các chức năng nổi bật:**

### ***5.1. Transparent form:***

* Điều chỉnh thuộc tính opacity của form. Chuyển form sang trạng thái transparent



private void rbtn\_opacity\_CheckedChanged(object sender, EventArgs e)

{

if(rbtn\_opacity.Checked=true)

{

this.Opacity = .80;

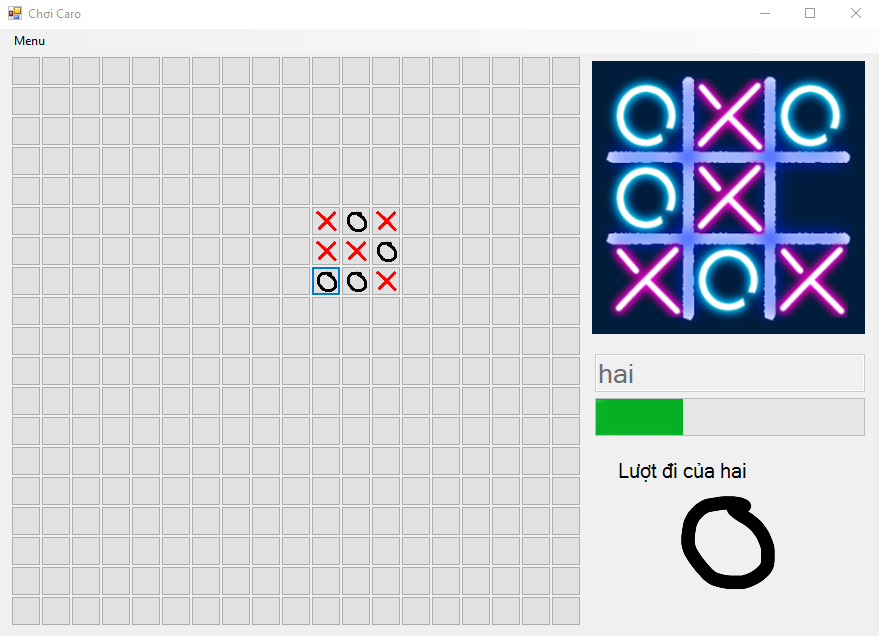
}

else

this.Opacity = .100;

}

### ***5.2. Chơi game Caro***



* Nếu là server thì sẽ tạo server, nếu là client thì sẽ lắng nghe thông tin từ server

private void Connect()

{

if (!socket.connectServer())

{

socket.isServer = true;

pnlChessBoard.Enabled = true;

socket.createServer();

}

else

{

socket.isServer = false;

pnlChessBoard.Enabled = false;

Listen();

}

}

* Hàm createServer để khởi tạo server

public void createServer()

{

IPEndPoint IPEP = new IPEndPoint(IPAddress.Parse(IP), port);

server = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);

server.Bind(IPEP);

server.Listen(10);

Thread acceptClient = new Thread(() =>

{

client = server.Accept();

});

acceptClient.IsBackground = true;

acceptClient.Start();

}

* Hàm Listen() lắng nghe thông tin giữa client và server

void Listen()

{

Thread listenThread = new Thread(() =>

{

try

{

SocketData data = (SocketData)socket.Recieve();

ProcessData(data);

}

catch

{

}

});

listenThread.IsBackground = true;

listenThread.Start();

}

* Xử lí data nhận được từ đối thủ

private void ProcessData(SocketData data)

{

switch(data.Command)

{

case (int)SocketCommand.NEW\_GAME:

this.Invoke((MethodInvoker)(() =>

{

NewGame();

pnlChessBoard.Enabled = false;

}));

break;

case (int)SocketCommand.NOTIFY:

MessageBox.Show(data.Message);

break;

case (int)SocketCommand.SEND\_POINT:

this.Invoke((MethodInvoker)(() =>

{

pnlChessBoard.Enabled = true;

prcbCoolDown.Value = 0;

tmCountDown.Start();

chessBoard.OtherPlayerMark(data.Point);

}));

break;

case (int)SocketCommand.QUIT:

tmCountDown.Stop();

MessageBox.Show("Đối thủ đã thoát!");

Close();

break;

default:

break;

}

Listen();

}

# **IV. CHƯƠNG CÀI ĐẶT VÀ KIỂM THỬ**

1. **Môi trường**

* Hệ điều hành Windows 10 Pro
* Visual Studio 2017, .NET Framework 4.0
* Máy tính xách tay ASUS, Intel core i7-45100 CPU @ 2.00GHz 2.60GHz, RAM 4.00 GB

## **2. Ngôn ngữ cài đặt**

* Ngôn ngữ lập trình C#

## **3. Thử nghiệm**

* Thử nghiệm các tính năng của phần mềm: Gửi tin nhắn, Chat Webcam, Thêm bạn bè, Tạo group chat, Gửi ảnh, Chơi game, Chat trong group, Xem lịch sử chat

1. **Đánh giá kết quả**

* Gửi tin nhắn, gửi hình ảnh, biểu tượng cảm xúc ổn
* Có thể tạo group chat để dễ dàng trao đổi
* Chat Webcam chưa có âm thanh, mới chỉ có hình ảnh
* Thêm bạn bè vào danh sách
* Có thông báo khi một người bạn trên hệ thống online hay offline
* Có Transparent Form để làm mờ Messenger Box
* Có thể chơi game caro

# **V. CHƯƠNG KHÁC**

## **1. Thuận lợi và khó khăn**

### ***1.1. Thuận lợi:***

* Có tool hỗ trợ: Bunifu Framework
* Một số xử lý đã có trên mạng nên có thể tham khảo
* Sử dụng winform nên thiết kế giao diện đơn giản

### ***1.2. Khó khăn:***

* Xung đột .NET Framework khi lập trình Socket: Không thể cài đặt và sử dụng Framework cũ ( .NET Framework 2.0) nên không thể làm được tính năng gửi file như nhóm dự tính lúc ban đầu
* Chưa đủ kiến thức chuyên môn về mạng để có thể lập trình được các chức năng phức tạp
* Chưa hiện thực được hoàn toàn tính năng Webcam ( chưa có âm thanh, mới chỉ có hình ảnh)

## **2. Phân công công việc**

|  |  |
| --- | --- |
| **Công việc** | **Người phụ trách** |
| Database Sever | Hồ Thái Ngọc |
| LanMessengerContact |
| Form menu |
| LanMessengerLib |
| LanMessengerNotify |
| Form đăng nhập | Phạm Thị Hoàng Mai |
| Selected Contacts |
| FormAbout |
| FormOption | Nguyễn Công Minh |
| FormAddContact |
| FormChangeDisplayName |
| FormChangePassword |
| Thiết kế Poster | Phan Vĩnh Long |
| FormSendFile |
| Game caro |
| Viết báo cáo | Nguyễn Công Minh, Phạm Thị Hoàng Mai, Hồ Thái Ngọc, Phan Vĩnh Long |

## **3. Hướng phát triển**

* Hoàn thiện chức năng Webcam ( âm thanh)
* Hỗ trợ việc gửi và nhận hình ảnh
* Phát triển thêm các game phức tạp hơn

## **4. Tài liệu tham khảo**

* [Microsoft Documentation](https://msdn.microsoft.com/enus/library/system.net.iphostentry(v=vs.110).aspx)
* [Microsoft Documentation](https://msdn.microsoft.com/enus/library/system.net.dns.gethostbyname(v=vs.110).aspx?query=) (link 2)
* [Lập trình Socket giao tiếp TCP client/ server](https://aptech.vn/kien-thuc-tin-hoc/c-lap-trinh-socket-giao-tiep-tcpclientserver.html)
* [Cơ bản về Thread](https://yinyangit.wordpress.com/2011/04/11/c-thread-basic/)
* [Stackoverflow](https://Stackoverflow.com)
* [HowKTeam](https://howkteam.com)