**Dyscord Architecture and Pseudo-Code**

The following components are provided:

1. The following class-scoped variables are provided:

string targetUser; : the username to communicate with

string targetIp; : the ip address of the user to communicate with  
int targetPort; : the port of the user to communicate with

string myIp; : your ip address  
int myPort; : my listener port

Thread thread:  
Socket listener;

1. A PHP application at http://people.rit.edu/dxsigm/php/login.php

Using the WebBrowser control to go to:

<http://people.rit.edu/dxsigm/php/login.php?login=dschuh&ip=1.1.1.1>:2222

adds a login entry to the server with user's ip and port

<http://people.rit.edu/dxsigm/php/login.php?logins>

returns HTML similar to:

<button name=dschuh title='192.168.1.5:2222'> dschuh</button><button name=harry title='192.168.1.13:3333'> harry</button>

This HTML can be interacted with to choose a user to communicate with using the HtmlEntity.Click event. Note that the "name" attribute is the username, and the "title" attribute is their ip address and port.

1. The form constructor first displays another form to select the port to open the listener on. Then starts a new thread with a listener socket at the defined port. So every user running this program has their own listener process which is "listening" for messages.

Upon pressing "Send", any text entered in the textbox should be sent to the listener socket at the targetIp address, where the targetIp and targetSocket was fetched from pressing the "Users" button.

**SettingsForm Controls and Fields**

* public int myPort
* Port TextBox
* Port Label
* Start Server Button

**DyscordForm Controls and Fields**

* string targetUser; : the username to communicate with
* string targetIp; : the ip address of the user to communicate with
* int targetPort; : the port of the user to communicate with
* string myIp; : your ip address
* int myPort; : my listener port
* Thread thread:
* Socket listener;
* Send Button
* Login Button
* Users Button
* Exit Button
* Conversation RichTextBox
* Current Message to Send RichTextBox
* UserName TextBox
* Conversation GroupBox
* WebBrowser
* Split Container for bottom 2 panels of controls

**Pseudocode**

**SettingsForm( Form owner, int nPort ) Constructor**

{

* // set owner to owner form
* // center to parent
* // init myPort = nPort and portTextBox.Text
* // set StartServer button click event handler
* // set PortTextBox KeyPress event handler

}

StartServerButton\_\_Click

{

* // set myPort to contents of portTextBox
* // close the form

}

PortTextBox\_\_KeyPress()

{

* // if a digit or backspace was pressed, don't handle it and let .NET handle it (ie. add it to the portTextBox contents

}

**DyscordForm Constructor**

{

* // create SettingsForm
* // display settings form
* // create listener thread and delegate to listen or messages from other users and add them to the conversation RichTextBox
* fetch the v4 IP address of my computer
* // add Login\_\_Click eventhandler to Login button
* // add Users\_\_Click eventhandler to Users button
* // add Send\_\_Click eventhandler to Send button
* // add CompletedDocument eventhandler to webBrowser control

}

Login\_Click()

{

* webBrowser opens [http://people.rit.edu/dxsigm/php/login.php?login= loginTextBox.Text&ip=myIp](http://people.rit.edu/dxsigm/php/login.php?login=%20loginTextBox.Text&ip=myIp):myPort
* disable loginTextBox and Login button
* webBrowser.visible = false

}

Users\_Click()

{

* WebBrowser.Visible = true
* webBrowser opens <http://people.rit.edu/dxsigm/php/login.php?logins>

}

WebBrowser\_\_CompletedDocument()

{

// fetch the array HtmlElements by the Button Tag

HtmlElementCollection hc = document.GetElementsByTag("button");

foreach( HtmlElement he in hc )

{

// add the HE\_Click event handler to each button

he.Click += HE\_\_Click;

}

}

HE\_\_Click()

{

// when a user button is clicked

// split title into ip and port

ipPort = HtmlElement.title.Split(':');

// set the targetIp address to the title attribute up to the ":"

targetIp = ipPort[0];

// set the targetPort to the number after the ":"

targetPort = ipPort[1];

// set the targetUser to the name attribute

targetUser = HtmlElement.name

// update the GroupBox title to indicate who I am conversing with

GroupBox.Text = "Conversing with " + targetUser;

// hide the WebBrowser control

WebBrowser.Visible = false;  
}

Send\_Click

{

* Add ">"+targetUser+messageTextBox.Text to the Conversation RichTextBox
* Send the message to the user over the socket using the targetIP address and targetPort
* Clear the message text box

}

UpdateConversationDisplay( string text )

{

* append text to the convRichTextBox

}

Listen()

{

* create delegate function to reference UpdateConversationDisplay
* open listener socket at myPort
* while(true) listen for client messages and add them to the convRichTextBox

}