**Client\_App.cs**

|  |  |  |  |
| --- | --- | --- | --- |
| **Original Author** | [**www.geekpedia.com/tutorial239\_Csharp-Chat-Part-1---Building-the-Chat-Client.html**](http://www.geekpedia.com/tutorial239_Csharp-Chat-Part-1---Building-the-Chat-Client.html) | **Date** | **3/12/2012** |
| **Author** | **A.J. Montgomery** | **Date** | **3/12/2012** |
| **File Name** | **Client\_App.cs** |  |  |
| **File Location** | **/CinderellaLauncher/Forms/Chat/Client\_App.cs** |  |  |
| **File Uses** | **MainMenu.cs, FGCheckin.cs, Alterations.cs, CheckIn.cs, CheckOut.cs, ShoppingMgt.cs** |  |  |
| **File Used By** | **Program.cs** |  |  |
| **Reason For Use** | **Use to implement for the application, ensuring that users are able to chat.** |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Method(line #)** | **Purpose** | **Precondition** | **Postcondition** |
| **connectButton(37)** | **The connect button is designed for the client app to connect to the chat server, as long as it is running.** | **User enters their username and clicks the connect button. During this time the connection is initialized.** | **Once username is entered in the textbox, they are connect and are able to chat. The button is set to disconnect.** |
| **InitializeConnection(55)** | **To ensure that the connection is established with its respective IP address and the user can either connect or disconnect.** | **The IP address is generated for connection. During this process, the chat server is running and monitoring for connections.** | **When the IP address and username is entered, the connection status is changed and the connect button is set to disconnect.** |
| **RecieveMessage(94)** | **This determines the connection status from the server. It happens after the server is monitoring for connection and after the connection is initialized.** | **The chat server is monitoring for connection and the username and IP address are both entered.** | **If the connection is initialized, then the connection is successful, otherwise the IP address is either entered wrong or the chat server is not running or crashed.** |
| **UpdateLog(107)** | **This happens when the message is sent on the Client\_App after the user clicks the send button.** | **The user writes a message prior to clicking the send button.** | **Once the user clicks on the send button after the message is written, the log will update onto the screen.** |
| **sendButton\_Click(176)** | **Executes when the message is written out and ready to be sent to the screen.** | **The message is written and waiting to be sent.** | **Once the button is clicked, the message is sent to the screen.** |
| **txtMessage\_KeyPress(183)** | **Using the enter key to send message to the screen after it has been written.** | **The message is written and waiting to be sent (using the enter key).** | **Once the enter key is pressed, the message is sent to the screen.** |
| **SendMessage(193)** | **The message is written in the textbox waiting to be sent to the screen once the enter key or the send button is clicked.** | **The textbox is waiting on a message to be written.** | **Once the message is written, then it can be sent off to the screen.** |
| **CloseConnection(207)** | **Designed for user to disconnect once user has finished chatting.** | **A user is finish chatting with other user and clicks on the disconnect button.** | **Once the button is clicked the user is disconnected and the button is set back to connect.** |
| **OnApplicationExit(234)** | **To ensure that the connection is close.** | **The user is disconnected and therefore is finish.** | **The connection is closed.** |
| **Client\_App\_FormClosing(252)** | **To close the form once it has been disconnected.** | **The connection is close.** | **Once the connection is close, then the form closes.** |
| **Client\_App\_Load(285)** | **The cursor focus on the username textbox once the form is loaded.** | **User wants to chat.** | **As soon as the user clicks to chat, the cursor blinks on the username textbox.** |