EE 4321 Computer Networks

Project 3: Online Chat Room

4/7/2019

Sam Petron

Mat Maternowsky

**Introduction**

This project demonstrates the use of socket programming to design an online chat room. Through the use of socket programming and gui application, the goal is to develop a single server that can support multiple clients.

**Client/Server Protocol**

We used a simple source:destination:type:message protocol as it was a very simple and sensible protocol. The message from who is checked first followed by checking the destination it is to be sent to. Afterwards there is a message type that the program checks to know which function to perform. Finally, there is the message which the message being sent from a client.

**User Manual**

To use the program start by initiating the server by opening the server file and pressing the start button. Next, open up as many client programs as you wish, just make sure that the server is running as the client will not connect otherwise. Once the server is up and the client is running, press the login button, this will initiate a connection with the server and ask you to enter a username. This name will pop up in the current user list, and the list will update as new users connect. Then simply write a message in the bottom text box and hit send. The message will show up in your display and be sent to everyone. Once you are done make sure to hit the logout, it will remove your name from the list of users, and shutdown your client program. Make sure all clients are off before shutting down the server, as the clients will not close properly if the server shuts down first.

**Important Notes**

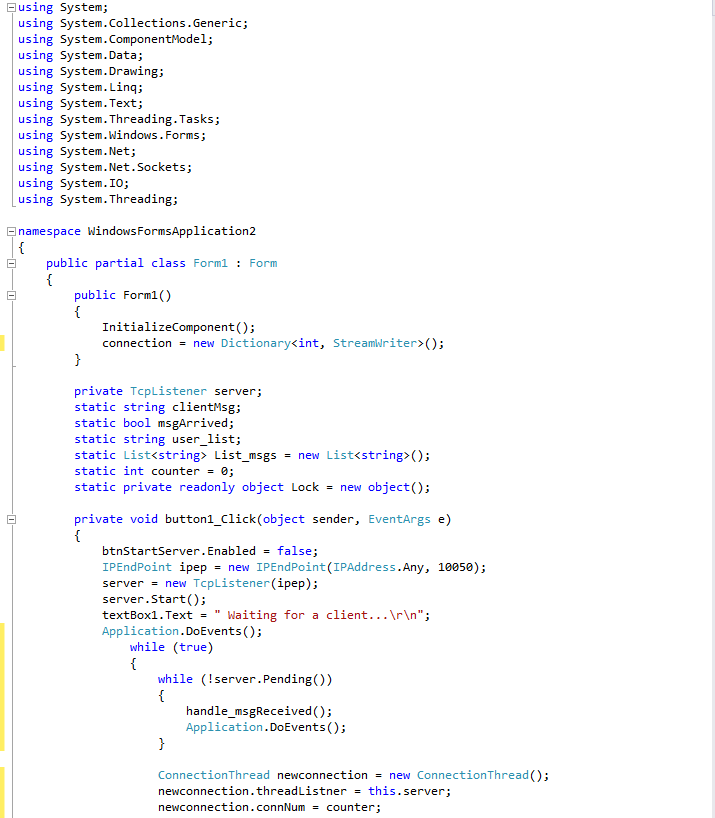
For the server side of the code the important features are methods for handling a new client thread connection, a method for storing and sending a user list to the client, and a method that notifies when the server connection has closed. On the client side the important features are a method for logging out, a method for sending a message to the server, a method for decoding the incoming message from the server into the correct protocol category, a method for updating the received userlist from the server, a method for printing messages from users on the client chat screen, and a method for notifying when the server is down.

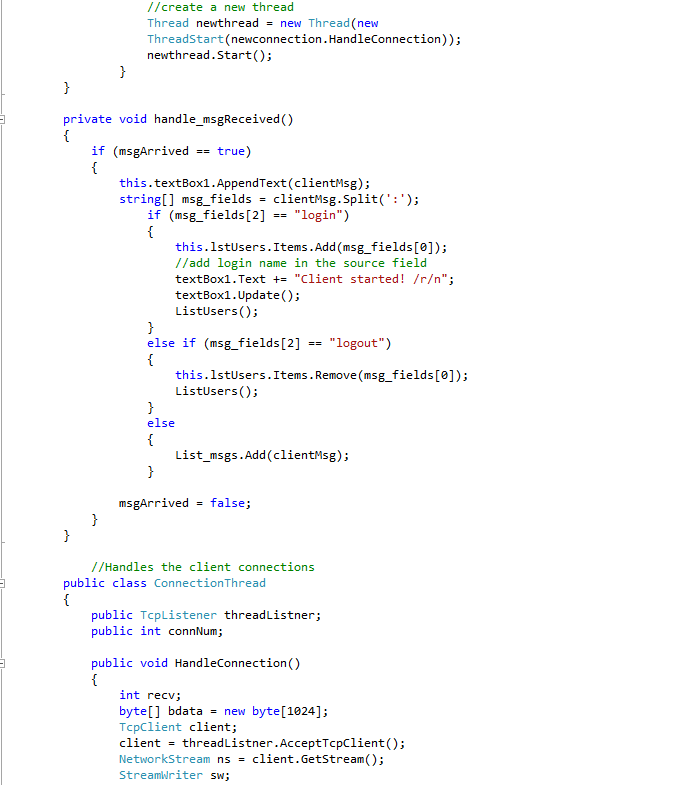
**Conclusion**

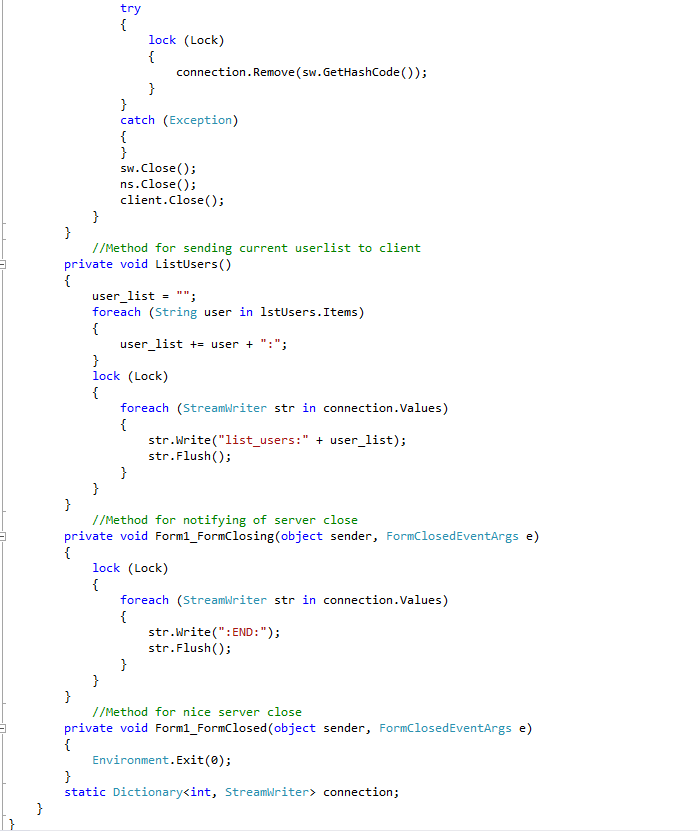
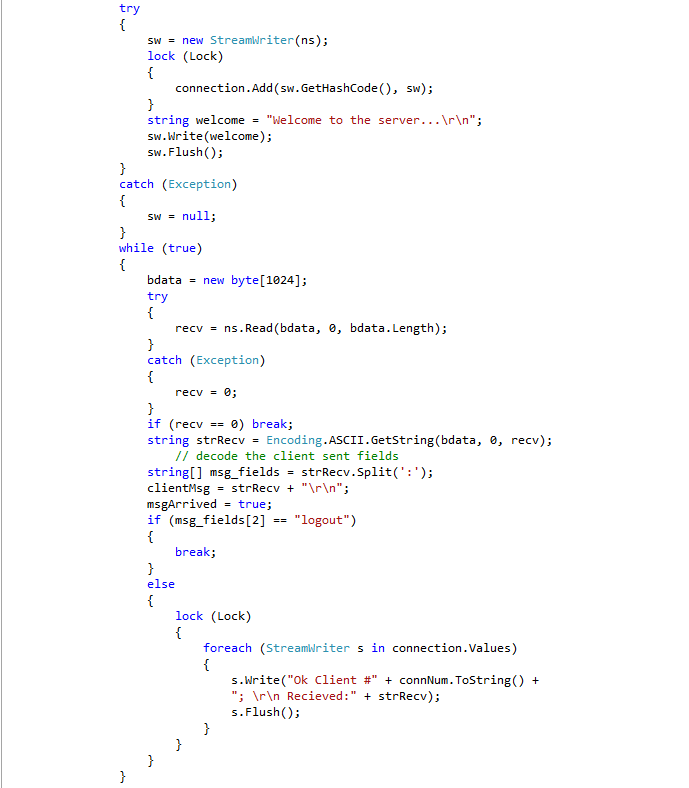
This project was useful in understanding how to use multiple threads to create a connection between clients and a server. It is useful information to know when thinking about designing networked applications. One application is to create a server client program that can send computer or computer application updates through the program. The server could also store important user information and free up space on the client's computer.

**Source Code**

**Server:**







**Client:**