Final Report For:

Virtual Chat Room(VCR) Instant Messaging System

Submitted By:

Joshua O. Lee
November 24, 2012

Joshua O. Lee
3971 HWY 310
Joliet, MT 59041
(406)962-3726

November 24, 2012

To whom it may concern,

Enclosed is the final report in response to your request for an instant messaging system to be used internally on local networks. I hope you will find that this software, currently under the working name of Virtual Chat Room or VCR, not only meets all of your desired specifications, but exceeds your expectations.

The final report for VCR includes the following:

- Introduction
- Review of Specifications
- Design Approach
- Debugging and Stress Test Results
- Conclusion

It has been a pleasure working with you and your company. If you have any questions, concerns, or would like to discuss this report in further detail please do not hesitate to contact me. I look forward with working with you and your company again in the future.

Sincerely,

Joshua O. Lee

Table Of Contents

Subject:	Page Number
Introduction	**4**
Review of Specifications	**4**
Design Approach	**5**
• Basic Overview of Design	5
• Changes form Original Proposed Design	5
• Information Storage Design	5
• Network Communications Design	6
• Multithreading Design	7
Debugging and Stress Test Results	**8**
Conclusion	**8**

Introduction

The main purpose of this project was to provide a lightweight instant messaging system for commercial and consumer use. VCR is designed to work only on the local subnet of a network and without having to set up and designate a server for it. Without a need for a server, this program is able to turn any subnet that at least one VCR is running on into a virtual chat room hence its name. Since it only works on local networks it keeps, users from contacting other users outside of the network.

Review of Specifications

I feel the need to provide you and your company with high quality software that not only fully meets your needs and specifications but also exceeds them. Outlined below is the specifications that this project must meet.

VCR must:

- Be lightweight and portable
- Work without a server
- Secure two-way chat features
- Have the ability to create a Virtual Chat Room on the current subnet
- Actively broadcast user information, such as IP and user name, to all other clients on the network
- Have simple and easy to use graphical user interface

Design Approach

Basic Overview of Design:

VCR contains three major parts that will be explained in detail below: Information Storage, Networking and Multithreading. VCR's user interface contains two windows: a log in form and a chat window. The log in form is used to get the users display name and check if the name is valid or already taken. As soon as the user enters their name, it starts a thread that listens to the pings to check if the user name is already in use. If the user name entered is valid and not in use, the user name is transferred to a global variable in the chat window and the chat window is opened. When the chat window is loaded, three threads are started to handle receiving messages and pings and to send pings. When a new user is detected on the network, that users information is stored into a data structure array and displayed in the list box in an easy to read format. While all received messages are displayed on a read only rich text box. All messages to be sent are entered into a text box below it and is sent when the user presses enter.

Changes from Original Proposed Design:

The only deviation from the original proposed design is peer to peer communication is handled by UDP instead of TCP. This was done to save on memory and overhead, lessen network load, avoid unnecessarily opening and tying up ports, and to simplify the program overall.

Information Storage Design:

All essential information is stored within three global variables. First, an integer named mode holds the current mode code for the user. These mode codes are represented by enums, that can be found in the ActiveUsers.h file, that range from 0 to 7 allowing for 8 different modes. Next, a string named user is used to store the users screen name. This variable is only assigned a value once and is not altered any further after the form loads. The final variable is a dynamically resizing array called OnlineUsers. OnlineUsers is an array of a data structure call ActiveUsers, and is used to hold another users information like screen name, IP address, and the user's current mode. The data structure ActiveUsers can be found in ActiveUsers.h and consist of two strings, the users name and IP, and a single integer, the users mode. The mode variable in ActiveUsers is also represented with enums just like the global mode variable mentioned above. The only difference between both mode variables is ActiveUsers mode variable is used for other users on the network and not for the user of the local client.

Network Communications Design:

All communication done between clients is done via UDP. Two types of communications are preformed by the client: pings and messaging communications.

Pings are constantly broadcast from the client at an interval of 10 milliseconds on port 2100. These pings broadcast information like the users current mode and the user's name as an ASCII string in the following format: modeUserName. The user's IP address is also extracted from these pings using an IPEndpoint. The user name is only read once and is never updated till the user disconnects while any changes in a users current mode are imminently updated on all clients. The modes sent by pings are modes 0 to 2 which stand for public chat, private chat, and away respectively.

Messaging broadcasts messages from the user to all available clients when in public chat mode while only sending messages to a specified IP address when in private chat mode. Messaging communications are also used to handle sending and responding to private chat invites from other users, alerting users when private chat has been terminated, and communicate to the clients when a user closes VCR's connection. All messaging communications are sent on port 2200 as an ASCII string. Messaging communications follows two different formats. All general messages from the user follow a format of modeUserName: SentMessage\n. For general messages mode is just used for filtering out public messages from private messages while in private chat and filtering out the utility messages so they are not displayed onscreen. A substring consisting of everything after mode is extracted and displayed for the user to see. All general messages send the same modes as pings, modes 0 to 2 .

Utility messages are for communicating private chat invites, responses to invites, termination of private chats, and alerting the clients when a user leaves the chat so they can be deleted form the system and memory freed up. They are sent as messaging communications but are sent in a format different from general messages. Instead of using the format of modeUserName: SentMessage\n they use the same format as the pings, modeUsername. Unlike pings though the modes sent by utility messages are modes 3 to 7. Respectively these modes stand for disconnect from chat, invite to private chat, accept invite, decline invite, and terminate private chat.

Multithreading Design:

VCR currently uses four threads, apart from the main thread, to carry out background tasks. The four threads are used for checking if the user name is valid, broadcasting user information, and receiving user information and messaging communications. The chat window only runs three of the four threads while the fourth thread runs in the log in screen. The threads for broadcasting user information and receiving user information and messaging communications are all started as soon as the chat window loads, and run continuously in an infinite loop in the background until the chat windows is closed. Upon closing all threads are forcefully terminated and all back ground tasks cease. If an error is encountered on one of the thread's loops, the loops are broken an the error is displayed in the chat window along with the name of the thread the error occurred in.

The thread that runs on the log in screen is different from the threads that run in the chat window. Instead of running in the background this thread blocks the execution of the main thread until a time limit has been reached or a duplicate name has been found. It also runs on an On Click event as opposed to when the form loads. When the user attempts to log in with a user name, the user name is written to a global variable and a new thread is created and started. As soon as the thread is started it is joined with the main thread causing it to block the execution of the main thread for 5 seconds or when the thread terminates and gives control back to the main thread. As soon as the main thread regains control it changes a boolean value to false and waits half a second for the thread to terminate “gracefully”. After this half a second it checks if the thread is still alive. If the thread is still alive thread is then forcefully terminated.

Debugging and Stress Test Results

During debugging, an alpha version of VCR was given to a group of about 15 student and they where told to break it. To even my surprise this unfinished version proved extremely stable, and while some minor bugs where found and fixed no program crashing bugs where found. The only major error that came up during debugging is one student could not receive messages unless VCR was run with administrative privileges. Since he was the only one to encounter such a problem thus far the error was most likely caused by an outside source, such as a windows firewall related setting or the required ports not being opened correctly. Stress test results where also equally surprising for an alpha version of the software. Even when every one was told to hold enter and spam messages, creating what effectively amount to a denial of service attack on the system, we had a hard time breaching 0.5% of network usage and only hit a max 0.6%. VCR also remained extremely responsive, and only showed some unresponsiveness when one user tried to open one hundred instance on his computer. This was only momentary though and after all the initial information was processed from that user the clients normal operation resumed as if nothing happened. The current version of VCR is even more stable, fixing may of the exploits that where used to try to flood the network.

Conclusion

As can be seen VCR has been designed and tested to meet all your specifications. VCR is able to work without a server, has a simple and easy to use user interface, allows for private peer to peer communications as well as public messages that can be broadcast to all users on the current subnet, actively broadcasts user information in the form of “pings”, and is very lightweight and portable, only consisting of two files that combined are only a mere 488KB in size. All current stress test have also shown VCR to be a dependable, stable program that uses very little bandwidth. The only major problem to have come up thus far is that some users will need to run this program with administrative privileges to get it through the firewall. I would like to thank you for allowing me to work on such an interesting project for your company. I look forward to conducting further business with you and your company in the future.