Linux Chat Application

Oscar Kwan

Krystle Bulalakaw

Test Document

March 23, 2016

Test Cases

Test ID Test Description Test Steps Expected Results Pass/Fail

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | Server compiles and runs without and warnings or errors. | 1. Navigate to the folder containing *server.cpp* 2. Type *gcc -Wall -o server server.cpp* 3. Type *./server* 4. Observe results | No warnings or errors are produced. A print statement indicates the server is listening. | Pass |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2 | Client connects to server successfully. | 1. Launch server application 2. Launch client application 3. Navigate to the connect tab 4. Enter in appropriate fields for username, ip address, and port. 5. Press connect button after these fields are entered | The server reports a connection of a client along with its name and IP address.  The client indicates the user is online and on the list of connected clients. | Pass |
| 3 | Client receives messages successfully. | 1. Launch server application 2. Launch multiple server applications on different hosts 3. Repeat Test #2 to connect all clients 4. Go to Connect tab, enter text in chat box and press <Enter> | Server indicates client messages sender and contents.  Each client receives messages sent from another, along with name and timestamp. | Pass |
| 4. | Timestamps displayed correctly in window. | 1. Observe chat log from Test ID #3. | Timestamp is displayed on the right of name whenever user sends or receives a message. | Pass |
| 5 | Client logs chat session to file if pressed. | 1. Launch server application 2. Launch client application 3. Go to Connect tab 4. Enter appropriate values for name, IP, and host and click Connect 5. Go to Chat tab 6. Send and receive text to other users (the chat is logged only if there is some history) 7. Click Export to file 8. The log file can be found in *linux-chat/Source/build-linuxchat-Desktop\_Qt\_5\_5\_1\_GCC\_64bit-Profile/log.txt* | The contents of log.txt should match the chat history from the beginning up to the point where **Export to file** was clicked. | Pass |
| 6 | User can disconnect. | 1. Launch a server and client application and connect as normal. 2. Enter some test text to ensure you are connected and receiving from the server. 3. Navigate to Connect tab and click **Disconnect** 4. Observe terminal and Chat tab. | The server should report that a disconnect message was received, and that it disconnected the client and updated the client/username list.  The client should appear as offline and not see any other online users. | Pass |
| 7 | User can reconnect after disconnecting. | 1. Repeat Test #6. 2. Click the **Connect** button once more. 3. Observe terminal and chat tab. | The server should report that a client connected, and it updated the client/username list.  The client should appear as online and see the list of connected users. | Pass |
| 8 | Attempting to connect with Invalid parameters results in an error message box. | 1. Launch server and client application 2. Attempt to connect with invalid parameters:    1. Blank    2. Invalid host    3. Invalid IP    4. Server not running | The client should indicated with an error message that it was not able to connect due to socket error, host resolution error, etc. | Pass |