

Feature	Points
Overall Program Structure	20
<ul style="list-style-type: none"> ● GUI code still works! 5 ● Correct "main" handling 5 ● Ability to run multiple instances of program on same host, using different ports and JmDNS names 5 ● Good modularity, with most networking code in files other than gui.py 5 	
Basic Protocol Functionality	10
<ul style="list-style-type: none"> ● Correct JmDNS Discovery functionality 5 <i>Register your service with JmDNS</i> <i>Correctly process discovery and resolution</i> ● Correct GET_USER_INFO handling 5 <i>Should generate GET_USER_INFO messages for discovered peers, and correctly handle results</i> <i>Should correctly process GET_USER_INFO messages from peers</i> <i>Should integrate information returned from GET_USER_INFO into GUI</i> 	
File Sharing Behavior	30
<ul style="list-style-type: none"> ● Correct LIST_FILES handling 10 <i>Should be able to generate LIST_FILES to peers to browse their shared files</i> <i>Should be able to accept and process LIST_FILES from peers</i> <i>Integrate results of LIST_FILES into GUI</i> ● Correct GET_FILE functionality 10 <i>Should be able to generate GET_FILE to a peer to request a file</i> <i>Should be able to respond to GET_FILE from a peer, returning file contents</i> <i>Integration into GUI: show status info about files fetched from you; show some indication of files retrieved from peers (display file, save to filesystem, etc.)</i> ● Correct OFFER_FILE behavior 10 <i>Generate an OFFER_FILE message to peers as a result of drag-and-drop</i> <i>Integrate received OFFER_FILE messages into GUI by popping up a confirmation dialog</i> <i>Respond to accepted OFFER_FILE messages by generating a GET_FILE; returned file should be reflected in the GUI somehow (through a status message, or by saving to the filesystem, etc.)</i> 	
Conversation Management	40
<ul style="list-style-type: none"> ● Should be able to support multiple chats at one time 10 <i>In other words, everything below should work when there are two or more chats going</i> ● Correct INVITE handling 5 <i>Send INVITE message when new conversation is attempted</i> <i>Display invitation window for INVITES received from peers</i> <i>Respond with ACK or NACK to received INVITES</i> ● Correct CHAT_STATUS handling 10 <i>If you are the initiator of the chat, generate CHAT_STATUS messages to all peers when the chat membership list changes</i> ● Correct MESSAGE handling 5 <i>Generate MESSAGES to all peers when you send text to them</i> <i>Properly receive MESSAGES from peers, and update the GUI display</i> ● Correct GOODBYE handling 5 <i>Generate a GOODBYE message to the initiator when you exit a chat</i> 	

<u>Feature</u>	<u>Points</u>
<ul style="list-style-type: none">• Correct CLOSE_CHAT handling <i>If you are the initiator, send a CLOSE_CHAT message to all peers when the chat is stopped</i>	5
Bonus	10
<ul style="list-style-type: none">• Fancy protocol enhancements (such as sending images in MESSAGEs for example).• Exception handling above and beyond the call of duty (survive server crashes, malformed messages from server, etc.)	5