

Submission Worksheet

CLICK TO GRADE

<https://learn.ethereallab.app/assignment/IT114-003-F2024/it114-milestone-4-chatroom-2024-m24/grade/js2637>

Course: IT114-003-F2024
Assignment: [IT114] Milestone 4 Chatroom 2024 M24
Student: Jack S. (js2637)

Submissions:

Submission Selection

1 Submission [submitted] 12/10/2024 7:57:35 PM

Instructions

^ COLLAPSE ^

- Implement the Milestone 4 features from the project's proposal document:
<https://docs.google.com/document/d/1ONmvEveI97GTFPGfVwwQC96xSsobbSbk56145XizQG4/view>
- Make sure you add your ucid/date as code comments where code changes are done
- All code changes should reach the Milestone4 branch
- Create a pull request from Milestone4 to main and keep it open until you get the output PDF from this assignment.
- Gather the evidence of feature completion based on the below tasks.
- Once finished, get the output PDF and copy/move it to your repository folder on your local machine.
- Run the necessary git add, commit, and push steps to move it to GitHub
- Complete the pull request that was opened earlier
- Upload the same output PDF to Canvas

Branch name: Milestone4

Group



Group: Features
Tasks: 4
Points: 9

^ COLLAPSE ^

Task

100%

Group: Features

Task #1: Client can export chat history of their current session (client-side)

Weight: ~0%

Points: ~0.01

^ COLLAPSE ^

Details:

For this requirement it's not valid to have another list keep track of messages. The goal is to utilize the location where messages are already present.



This must be a client-side implementation.

Columns: 1

Sub-Task

100%

Group: Features

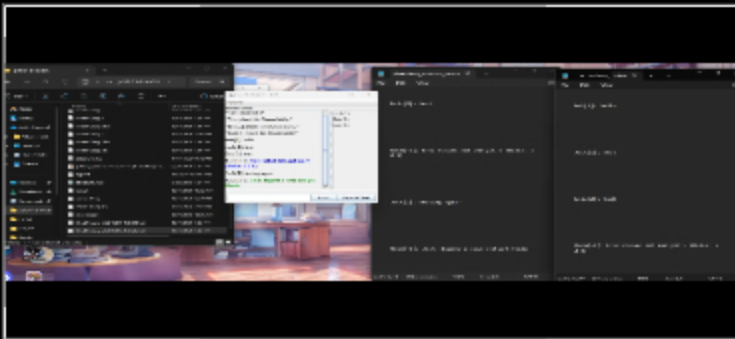
Task #1: Client can export chat history of their current session (client-side)

Sub Task #1: Show a few examples of exported chat history (include the filename showing that there are multiple copies)

Task Screenshots

Gallery Style: 2 Columns

4 2 1



chat history example

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Sub-Task

100%

Group: Features

Task #1: Client can export chat history of their current session (client-side)

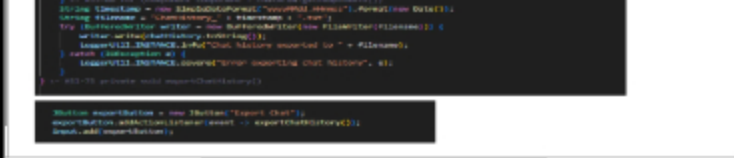
Sub Task #2: Show the code related to building the export data (where the messages are gathered from, the StringBuilder, and the file generation)

Task Screenshots

Gallery Style: 2 Columns

4 2 1





export code

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Explain in concise steps how this logically works

Response:

The second part of the code was the just add the button to the chat panel UI. The frist part of the code use to go through the text and create a file with the chat history the block of code in the for loop removes all the unnessary part of the htlm forming in the text file, this does create a lot of spaces in the text file, but makes it more read able when the file is open.

Sub-Task

100%

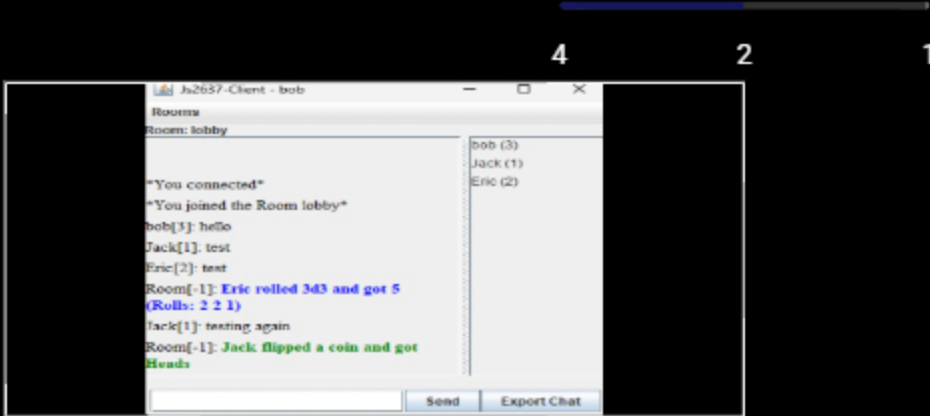
Group: Features

Task #1: Client can export chat history of their current session (client-side)

Sub Task #3: Show the UI interaction that will trigger an export

Task Screenshots

Gallery Style: 2 Columns



the export chat next to send button

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Explain where you put it any why

Response:

I put the export chat button next to the send button because it was easier to code and didn't look too weird.

End of Task 1

Task

Group: Features

100%

Task #2: Client's Mute List will persist across sessions (server-side)

Weight: ~0%

Points: ~0.01

^ COLLAPSE ^

i Details:

This must be a server-side implementation.

Screenshots of editors must have the frame title visible with your ucid and the client name.
Code screenshots must have ucid/data comments.

Columns: 1

Sub-Task

Group: Features

100%

Task #2: Client's Mute List will persist across sessions (server-side)

Sub Task #1: Show multiple examples of mutelist files and their content (their names should have/include the user's client name)

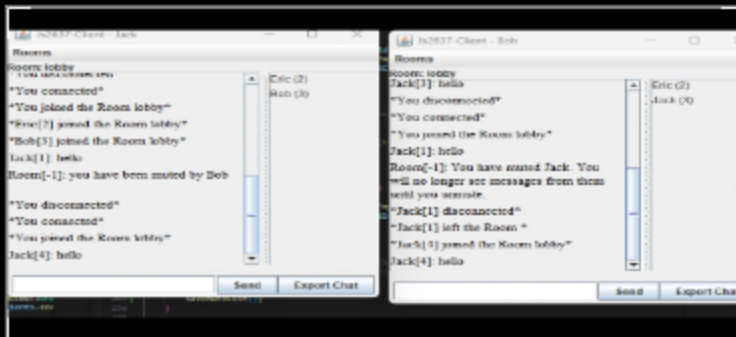
Task Screenshots

Gallery Style: 2 Columns

4

2

1



mute

Caption(s) (required) ✓Caption Hint: *Describe/highlight what's being shown***Sub-Task**

Group: Features

100%

Task #2: Client's Mute List will persist across sessions (server-side)

Sub Task #2: Show the code related to loading the mutelist for a connecting client (and logic that handles if there's no file)

Task Screenshots

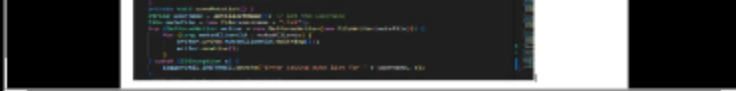
Gallery Style: 2 Columns

4

2

1





mute

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

≡ Task Response Prompt

Explain in concise steps how this logically works

Response:

I wasn't able to get the mute list to work, but I tried coding it so that when the client connects, it will load a mute list. and the name will be saved to it and it will load and save the mute list when called upon, but i don't know why my code doesn't work.

Sub-Task

Group: Features

100%

Task #2: Client's Mute List will persist across sessions (server-side)

Sub Task #3: Show the code related to saving the mutelist whenever the list changes for a client

🖼 Task Screenshots

Gallery Style: 2 Columns

4

2

1



Mute list

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

≡ Task Response Prompt

Explain in concise steps how this logically works

Response:

I wasn't able to get the mute list to work, but I tried coding it so that when the client connects, it will load a mute list. and the name will be saved to it and it will load and save the mute list when called upon, but i don't know why my code doesn't work.

End of Task 2

Task

Group: Features

Task #2: Clients will receive a message when they get muted/unmuted by another user

100%

Task #5: Clients will receive a message when they get muted/unmuted by another user
 Weight: ~0%
 Points: ~0.00

^ COLLAPSE ^

Details:

Screenshots of editors must have the frame title visible with your ucid and the client name.
 Code screenshots must have ucid/data comments.



I.e., /mute Bob followed by a /mute Bob should only send one message because Bob can only be muted once until

Columns: 1

Sub-Task

Group: Features

100%

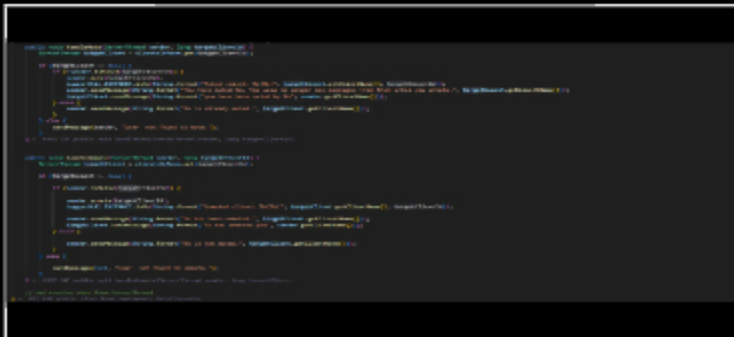
Task #3: Clients will receive a message when they get muted/unmuted by another user

Sub Task #1: Show the code that generates the well formatted message only when the mute state changes (see notes in the details above)

Task Screenshots

Gallery Style: 2 Columns

4 2 1



mute message

Caption(s) (required) ✓

Caption Hint: Describe/highlight what's being shown

Task Response Prompt

Explain in concise steps how this logically works

Response:

I just added the `targetClient.sendMessage(String.format("you have been muted by %s", sender.getClientName()));` and `targetClient.sendMessage(String.format("%s has unmuted you", sender.getClientName()));` to in the `handlemute` logic to handle sending the message out to the target client that they have been muted/unmuted.

Sub-Task

Group: Features

100%

Task #3: Clients will receive a message when they get muted/unmuted by another user

Sub Task #2: Show a few examples of this occurring and demonstrate that two mutes of the same user in a row generate only one message, do the same for unmute)

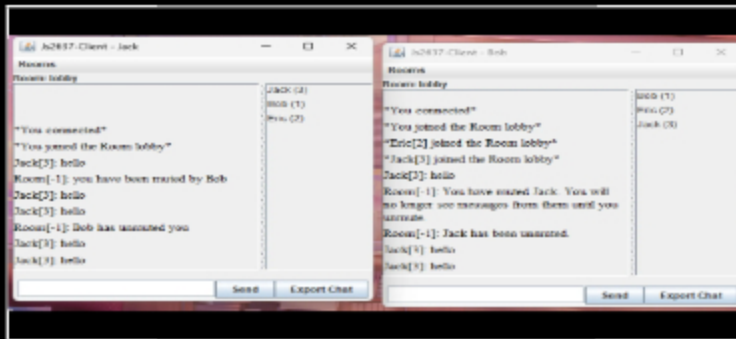
Task Screenshots

Gallery Style: 2 Columns

4

2

1



mute and unmute

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

End of Task 3

Task

100%

Group: Features

Task #4: The user list on the Client-side should update per the status of each user

Weight: ~0%

Points: ~0.01

^ COLLAPSE ^

Details:

Screenshots of editors must have the frame title visible with your ucid and the client name.
Code screenshots must have ucid/data comments.



Columns: 1

Sub-Task

100%

Group: Features

Task #4: The user list on the Client-side should update per the status of each user

Sub Task #1: Show the UI for Muted users appear grayed out (or similar indication of your choosing) include a few examples showing it updates correctly when changing from mute/unmute and back

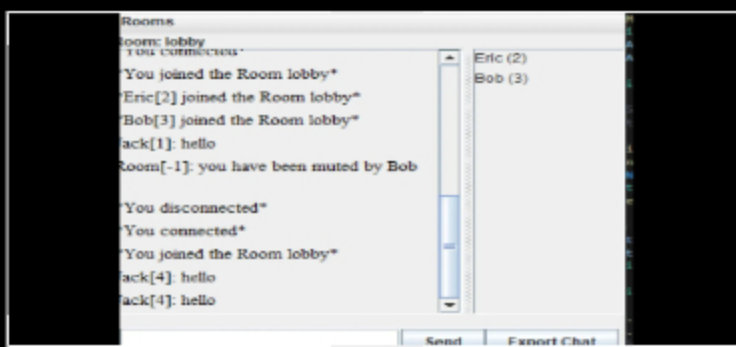
Task Screenshots

Gallery Style: 2 Columns

4

2

1



ui of room

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Sub-Task

Group: Features

Task #4: The user list on the Client-side should update per the status of each user

Sub Task #2: Show the code flow (client receiving -> UI) for Muted users appear grayed out (or similar indication of your choosing)

Task Screenshots



code

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Explain in concise steps how this logically works

Response:

I tried to change the appearance of the foreground to a new color in the Userlist Panel and UserList Items by using `.setforeground` and then call on the function when people type or is muted, but i wasn't able to get it to work correctly

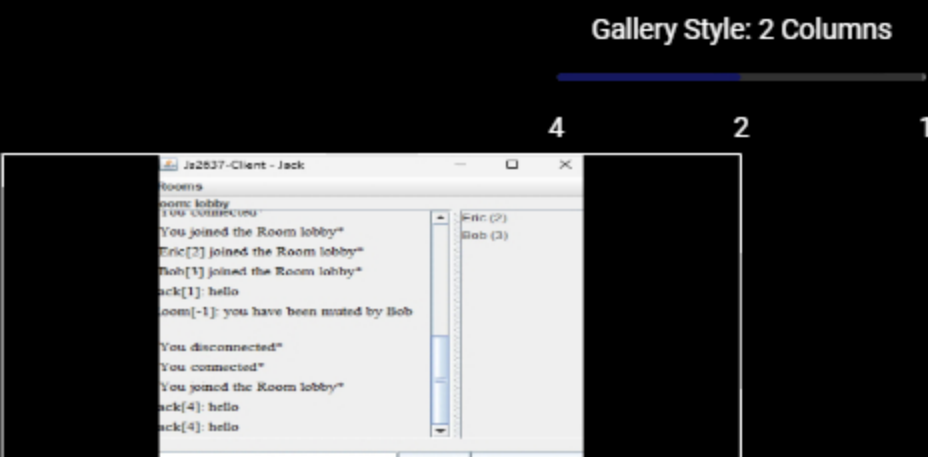
Sub-Task

Group: Features

Task #4: The user list on the Client-side should update per the status of each user

Sub Task #3: Show the UI for Last person to send a message gets highlighted (or similar indication of your choosing)

Task Screenshots



ui fo client

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Sub-Task

Group: Features

100%

Task #4: The user list on the Client-side should update per the status of each user

Sub Task #4: Show the code flow (client receiving -> UI) for Last person to send a message gets highlighted (or similar indication of your choosing)

Task Screenshots

Gallery Style: 2 Columns

4

2

1



Code

Caption(s) (required) ✓

Caption Hint: *Describe/highlight what's being shown*

Task Response Prompt

Explain in concise steps how this logically works

Response:

I tried to change the appearance of the foreground to a new color in the Userlist Panel and UserList Items by using `.setforeground` and then call on the function when people type or is muted, but i wasn't able to get it to work correctly

End of Task 4

End of Group: Features

Task Status: 4/4

Group

100%

Group: Misc

Tasks: 3

Points: 1

^ COLLAPSE ^

Task

Group: Misc

100%

Task #1: Add the pull request link for the branch
Weight: ~33%
Points: ~0.33

^ COLLAPSE ^

Details:

Note: the link should end with /pull/#



Task URLs

URL #1

<https://github.com/Jackshii/Js2637-IT114-003/pull/13>

URL

<https://github.com/Jackshii/Js2637-IT114-003/p>

End of Task 1

Task

100%

Group: Misc

Task #2: Talk about any issues or learnings during this assignment

Weight: ~33%

Points: ~0.33

^ COLLAPSE ^

Task Response Prompt

Response:

I wasn't able to figure out how to make the mute list and set the foreground of the UI I believe the mute list needed some file creation with the username and to pull the username from the file, but I wasn't able to figure it out, and I tried to solve the UI user color with .setforeground but nothing I tried worked.

End of Task 2

Task

100%

Group: Misc

Task #3: WakaTime Screenshot

Weight: ~33%

Points: ~0.33

^ COLLAPSE ^

Details:

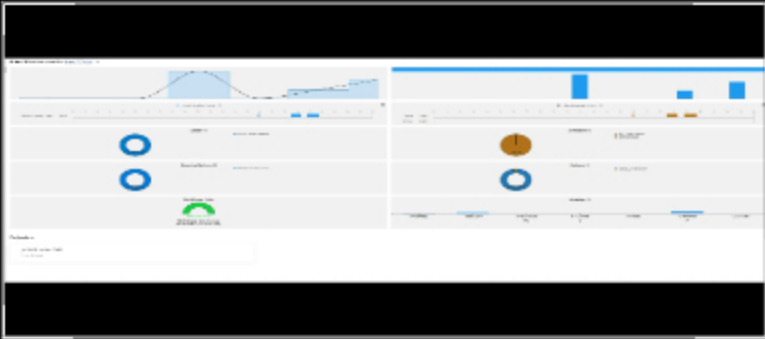
Grab a snippet showing the approximate time involved that clearly shows your repository. The duration isn't considered for grading, but there should be some time involved



Task Screenshots

Gallery Style: 2 Columns

4 2 1



Waka Time

End of Task 3

End of Group: Misc
Task Status: 3/3

End of Assignment