Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-450-M2024/it114-milestone-4-chatroom-2024-m24/grade/yh68

IT114-450-M2024 - [IT114] Milestone 4 Chatroom 2024 M24

Submissions:

Submission Selection

1 Submission [active] 7/29/2024 10:38:14 PM

Instructions

COLLAPSE

- Implement the Milestone 4 features from the project's proposal document: https://docs.google.com/document/d/1ONmvEvel97GTFPGfVwwQC96xSsobbSbk56145XizQG4/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

Tasks: 7 Points: 10.00

Features (9 pts.)

COLLAPSE

COLLAPSE

Task #1 - Points: 3

Text: Client can export chat history of their current session (client-side)

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.
A StringBuilder must be used for consolidation.
Screenshots of editors must have the frame title visible with your ucid and the client name.
Code screenshots must have ucid/data comments.

#1) Show a few examples of

Caption (required) ✓
Describe/highlight what's being shown
Showing two examples of exported chat history

#2) Show the code related to building

Caption (required) ✓
Describe/highlight what's being shown
Showing the code related to building the export data

Explanation (required) ✓
Explain in concise steps how this logically works
PREVIEW RESPONSE
The exportChatHistory() method collects the text content from all JEditorPane components within chatArea, appending each pane's text to a StringBuilder. It then makes a file and names it using the current date and time, writes the collected chat history to this file, and outputs a success or error message based on the outcome.

#3) Show the UI interaction that will

Caption (required) ✓
Describe/highlight what's being shown
Showing the UI interaction that will trigger an export

Explanation (required) ✓
Explain where you put it any why
PREVIEW RESPONSE
I put it south, to the right of the Send button, to make it accessible and easy to see.

COLLAPSE

Task #2 - Points: 3

Text: Client's Mute List will persist across sessions (server-side)

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.

#1) Show multiple examples of

Caption (required) ✓
Describe/highlight what's being shown
Showing multiple examples of mutelist files and their content

#2) Show the code related to loading

Caption (required) ✓
Describe/highlight what's being shown
Showing the code related to loading the mutelist

Explanation (required) ✓
Explain in concise steps how this logically works

PREVIEW RESPONSE

The loadMuteList() method first checks if clientName is not null. It then clears the mutedClients set and attempts to read from a file named after client's name with a .txt extension. Each line read from the file is added to the mutedClients set. If an IOException occurs

#3) Show the code related to saving the

Caption (required) ✓
Describe/highlight what's being shown
Showing the code related to saving the mutelist

Explanation (required) ✓
Explain in concise steps how this logically works

PREVIEW RESPONSE

The saveMuteList() method saves the list of muted clients to a file named after the client that is doing the muting, writing each muted client's name on a new line. It uses a BufferedWriter to create and write to the file, adding each muted client on a new line. If an IOException occurs

during file reading, it logs an error message with the exception details.

during file operations, it logs an error message.

COLLAPSE

Task #3 - Points: 1

Text: Clients will receive a message when they get muted/unmuted by another user

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 they're unmuted. Similarly for /unmute Bob

#1) Show the code that generates

Caption (required) ✓

Describe/highlight what's being shown

Showing the code that generates message when the mute state changes

Explanation (required) ✓

Explain in concise steps how this logically works

PREVIEW RESPONSE

The handleMuteUnmute() method manages the muting or unmuting of a client based on the isMute bool. It first initializes a boolean variable which is initially

#2) Show a few examples of

Caption (required) ✓

Describe/highlight what's being shown

Showing examples of this feature.

false. If isMute is true, it mutes the target client by calling addMutedClient on the sender object, and assigns the result to changed. If isMute is false, it unmutes the client by calling removeMutedClient on the sender, and assigns this result to changed. After the mute/unmute operation, the method logs a message indicating whether the client was muted or unmuted, using the sender's client ID and the targetClientName. After that, the method returns the changed status, indicating whether the operation was successful or not.

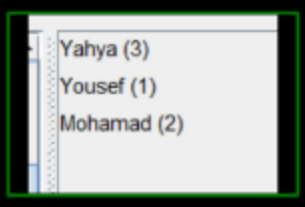
^COLLAPSE ^

Task #4 - Points: 3

Text: The user list on the Client-side should update per the status of each user

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

#1) Show the UI for Muted users appear

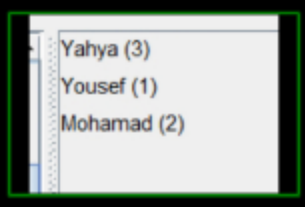


Caption (required) ✓
Describe/highlight what's being shown

#2) Show the code flow (client

Caption (required) ✓
Describe/highlight what's being shown

#3) Show the UI for Last person to



Caption (required) ✓
Describe/highlight what's being shown

#4) Show the code flow (client

Caption (required) ✓
Describe/highlight what's being shown

Showing the UI for this feature

Showing the code for this feature

Explanation (required)

✓

Explain in concise steps how this logically works

PREVIEW RESPONSE

The setMuted() method is supposed to update the client's mute status and trigger the updateAppearance() method to reflect this change visually and log the new mute status, update the isMuted variable, and then updateAppearance logs the current values of isMuted and isHighlighted. The updateAppearance method sets the text color to gray if isMuted is true and black otherwise, and changes the background color to yellow if isHighlighted is true and white otherwise, followed by revalidate() and repaint() to refresh the component's display. These updates are not appearing as expected, it is probably because I missed something with regards to the initialization or referencing of certain methods.

Showing the UI for this feature

Showing the code for this feature

Explanation (required)

✓

Explain in concise steps how this logically works

PREVIEW RESPONSE

The setHighlighted() method is supposed to update a client's highlighted status and invoke the updateAppearance() method to visually reflect this. It first logs the action with the new highlighted value, updates the isHighlighted field to the new value, and then calls updateAppearance to apply the changes. Some reasons it may not be functioning properly is something in the userlistpanel might not be initialized properly, but I'm not exactly sure otherwise it would be working.

Misc (1 pt.)

COLLAPSE

COLLAPSE

Task #1 - Points: 1

Text: Add the pull request link for the branch

Details:

Note: the link should end with /pull/#

URL #1

https://github.com/FreePalestine7/yh68-it114-450

URL

https://github.com/FreePalestine7/yh68-it114-4

ADD ANOTHER URL

COLLAPSE

Task #2 - Points: 1

Text: Talk about any issues or learnings during this assignment

Response:

My issues were mainly with the UI functions because I am super new to that aspect of java. I did learn a lot about java swing and it almost reminds me of front end development, I feel like once I get used to it I could honestly make anything I want. I brushed up on StringBuilder and bufferedwriter which I practically had to learn all over again.

COLLAPSE

Task #3 - Points: 1

Text: WakaTime Screenshot

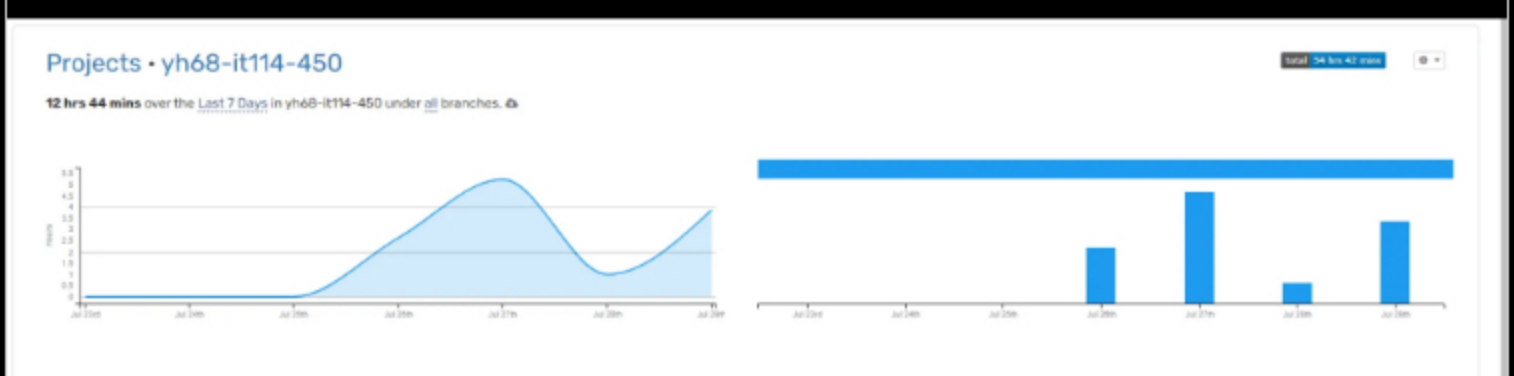
Details:

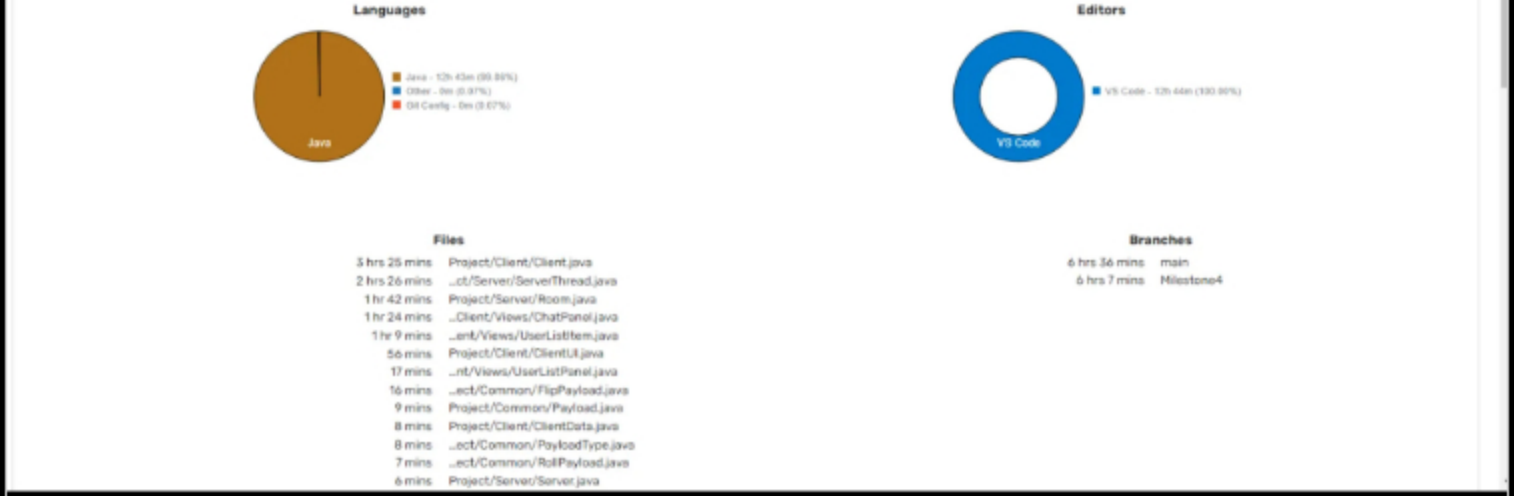
Grab a snippet showing the approximate time involved that clearly shows your repository. The duration isn't considered for grading, but there should be some time involved

Task Screenshots:

Gallery Style: Large View

Small Medium Large





Missing Caption

End of Assignment