Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-003-F2024/it114-milestone-4-chatroom-2024-m24/grade/rra23

Course: IT114-003-F2024

Assigment: [IT114] Milestone 4 Chatroom 2024 M24

Student: Rahid A. (rra23)

Submissions:

Submission Selection

1 Submission [submitted] 12/10/2024 7:33:45 PM

•

Instructions

∧ COLLAPSE ∧

- Implement the Milestone 4 features from the project's proposal document:
 https://docs.google.com/document/d/10NmvEvel97GTFPGfVwwQC96xSsobbSbk56145XizQG4/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 ^





Group: Features

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

Weight: ~0% Points: ~0.01



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



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

| The property of the control of the control

example 1 of export

example 2 of export



example 3 of export

Caption(s) (required) 🗸

Caption Hint: Describe/highlight what's being shown



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

1



code for exporting chat

Caption(s) (required) ~

Caption Hint: Describe/highlight what's being shown

Task Response Prompt

Explain in concise steps how this logically works

Response:

I created a button that the client can press to export. When the button is clicked, it gets the date and time when it is clicked so it can put that in the filename. Then it creates the file and writes the messages into the file using the text that is in the JEditorPane. The text is written into the file using FileWriter and StringBuilder.



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



code for export button

export button next to the send button in the ui

Caption(s) (required) ~

Caption Hint: Describe/highlight what's being shown

■ Task Response Prompt

Explain where you put it any why

Response:

I put it in the input JPanel next to the send button.

End of Task 1

Task



Group: Features

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

Weight: ~0% Points: ~0.01

^ COLLAPSE ^



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

100%

Group: Features

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

4

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

2



example of mutelist file (I could not get it to work, but this is how I was trying to format it)

Caption(s) (required) 🗸

Caption Hint: Describe/highlight what's being shown



100%

Group: Features

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 1 public void loadFuteList(){ //rra2% 12/18/24
 String fileName = this.getClientName() + "muteList";
 File file = new File(fileName); if (!file.exists()) (case CLIENT_CONNECT: ConnectionPayload cp = (ConnectionPayload) try (Scanner scan = new Scanner(file)){
 int lineNumber = 1;
 while(scan.hasNextline()){ setClientName(cp.getClientName()); loadMuteList(); //rra23 12/10/24 break; String[] mutedUsers = text.split(",");)catch(Exception e){
 e.printStackTrace();

attempted code for loading mutelist

loadMutelist

Caption(s) (required) 🗸

Caption Hint: Describe/highlight what's being shown

■, Task Response Prompt

Explain in concise steps how this logically works

Response:

I could not get this feature to work, but I was trying to do something similar to what was used in the loadState method from NumberGuesser4.java from earlier in the semester



Group: Features

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

2 4 private void saveMuteList(){| //rra23 12/10/24 String fileName = this.getClientName() + "muteList";
try(FileWriter fw = new FileWriter(fileName)){ if (payload.getPayloadType().equals(PaylaodType.MUTE))(//rrs23 12/10/24 fw.write("muted User,"); e.printStackTrace();

attempted code for saving mutelist

mutelist

Caption(s) (required) 🗸

Caption Hint: Describe/highlight what's being shown

■ Task Response Prompt

Explain in concise steps how this logically works

Response:

I also could not get this feature to work, but like the loadMuteList, I tried to do something like the loadState method

from Numberguesser4

End of Task 2

Task



Group: Features

Task #3: 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



Group: Features

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)

1

Task Screenshots

Gallery Style: 2 Columns

2

4

case PayloadType.NuTE: //rearx 1z/10/28
onPuteWeiServe();
String tempflessage = this.getClientNameProwld(this.getRyClientId()) * " muted you";
sendNameQu(impRomsage);
break;
case PayloadType.WeiPutE:
onMuteWeiServe(); //rearx 1z/10/24
%tring tempflessage? = this.getClientNameProwld(this.getMyClientId()) * "unewbod you";
sendNotsage(ImpRomsage?);

formatted message for mute and unmute

Caption(s) (required) ~

Caption Hint: Describe/highlight what's being shown

■ Task Response Prompt

Explain in concise steps how this logically works

Response:

I could not get this feature to work fully, but I tried to check if the user that the client is trying to mute is already on the mutelist, and if they are, it would not send the mute message. For the unmute, it would check if they are not on the list and if they are not, it would not send any message



Group: Features

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)

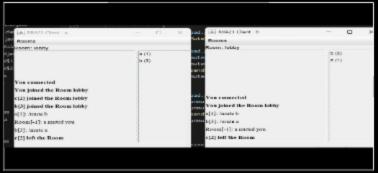
1

Task Screenshots

Gallery Style: 2 Columns

2

4



example

Caption(s) (required) ~

Caption Hint: Describe/highlight what's being shown

End of Task 3

Task



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 ^

① 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

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



example of muted indication

Caption(s) (required) ~

Caption Hint: Describe/highlight what's being shown



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

Gallery Style: 2 Columns

public void highlightMutedUser(long clientId, String clientName){ //rrs23 12/18/24 String displayName = "X" * clientName * "X"; premovelaces intitem(clientId); addMoserListItem(clientId), displayName); }

public void onMuteReceive(long clientId){ chatPanel.highlightMutedUser(clientId, clientName); //rrs22 12/19/24 }
}

code for indicating muted users

new method that calls the highlightMutedUser

Caption(s) (required) <

Caption Hint: Describe/highlight what's being shown

■ Task Response Prompt

Explain in concise steps how this logically works

Response:

I could not get this feature to fully work, but I tried to make it so that if someone is muted, their name appears with an x before it. I tried to get it to be grayed out, but I could not get the color to work in the userlistpanel.



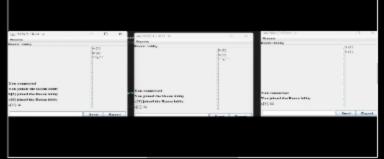
Group: Features

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

Gallery Style: 2 Columns



last message highlight

Caption(s) (required) <

Caption Hint: Describe/highlight what's being shown



Group: Features

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 void onMessageReceive(long clientId, String message) {
(currentCard.ordinal() >= CardView.CHAT.ordinal()) {
 String elientHame = Client.HSTANCt.getClientHameProxId(clientId);
 chetPanel.addText(String.format("%)(%): %); clientHame, clientId, message));
 chatPanel.infghlightLastHessage(clientId, clientName); //ren2% 12/16/24

updated onMessageReceive

highlight last message method

Caption(s) (required) 🗸

Caption Hint: Describe/highlight what's being shown

■ Task Response Prompt

Explain in concise steps how this logically works

Response:

When the message comes in, the onMessageReceive method calls the highlightLastMessage method in the ChatPanel.java which updates the userList putting asterisks in front of the name of the last person who sent a message.

I tried to get it to change the color but I could not get it, so I just put asterisks to highlight the user who sent the last message

End of Task 4

End of Group: Features

Task Status: 4/4

Group



Group: Misc Tasks: 3 Points: 1

^ COLLAPSE ^

Task



Group: Misc

Task #1: Add the pull request link for the branch

Weight: ~33% Points: ~0.33

^ COLLAPSE ^



Note: the link should end with /pull/#



⇔Task URLs

URL #1

https://github.com/Rahid-Ahmed/rra23-IT114-003/pull/16

UHL

https://github.com/Rahid-Ahmed/rra23-IT114-003

End of Task 1

Task



Group: Misc

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

Weight: ~33% Points: ~0.33

^ COLLAPSE ^

=, Task Response Prompt

Response:

I could not get all of the features to work properly. I tried to get them working but I could not.

End of Task 2

Task



Group: Misc

Task #3: WakaTime Screenshot

Weight: ~33% Points: ~0.33



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

Projects • rra23-IT114-003

4 hrs 18 mins over the Last 7 Days in rra25-IT114-003 under all branches.

| This is mins | Project/Client/News/Chet/Prositions | Project/Client/News/Chet/Prositions | A first 38 mins | Project/Client/Client/Semigram | Project/Client/Client/Semigram | Project/Client/Client/Semigram | Project/Client/Client/Semigram | Project/Client/Client/Semigram | Project/Client/Semigram | Pr

wakatime overview

wakatime detailed view

End of Task 3

End of Group: Misc Task Status: 3/3

End of Assignment