Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-002-S2024/it114-chatroom-milestone-3-2024/grade/cw72

IT114-002-S2024 - [IT114] Chatroom Milestone 3 2024

Submissions:

Submission Selection

1 Submission [active] 4/30/2024 10:04:15 PM

Instructions

COLLAPSE

Implement the Milestone 3 features from the project's proposal document: https://docs.google.com/document/d/1ONmvEvel97GTFPGfVwwQC96xSsobbSbk56145Xt
Make sure you add your ucid/date as code comments where code changes are done
All code changes should reach the Milestone3 branch
Create a pull request from Milestone3 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: Milestone3

Tasks: 14 Points: 10.00

Basic UI (2 pts.)

COLLAPSE

Task #1 - Points: 1

Text: Screenshots of the following

Checklist *The checkbox are for your own tracking

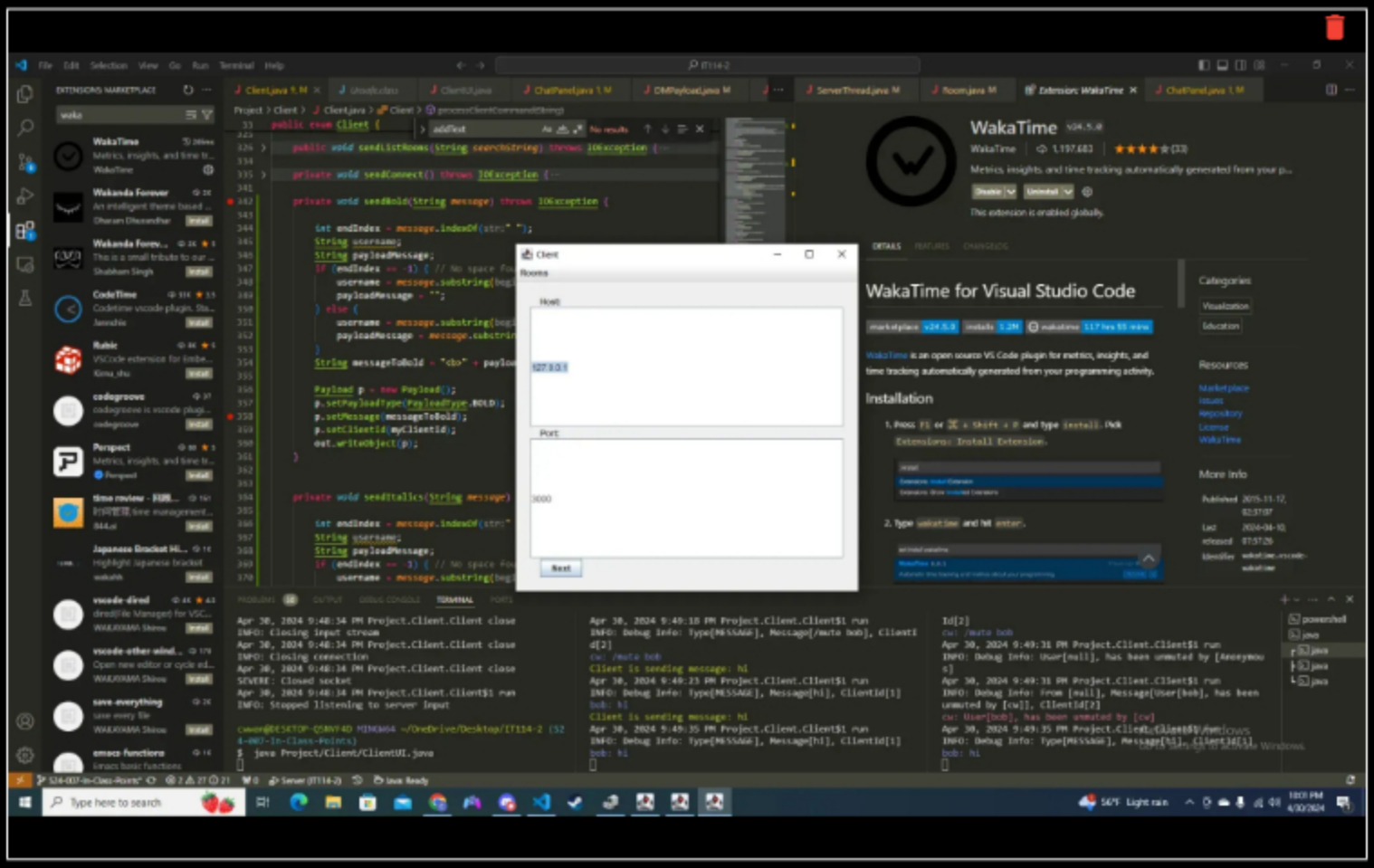
#	Points	Details

#1	1	Connection Panel
#2	1	User Details Panel
#3	1	Chat Panel
#4	1	Clearly caption screenshots

Task Screenshots:

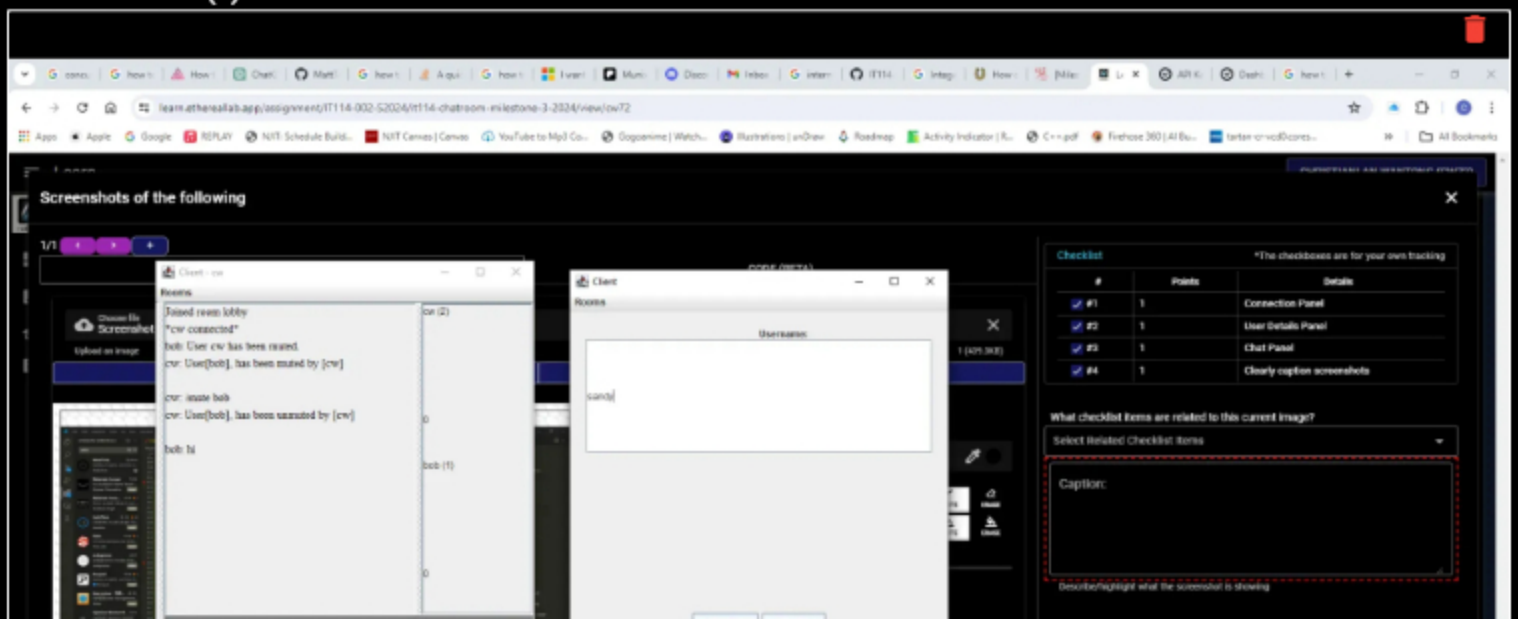
Gallery Style: Large View

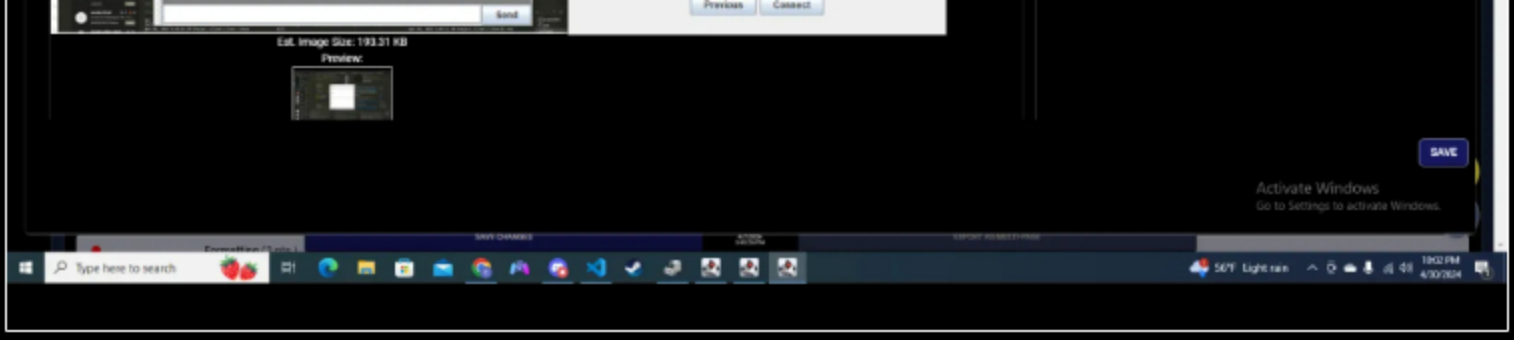
Small Medium Large



This is the screenshot of the connection panel

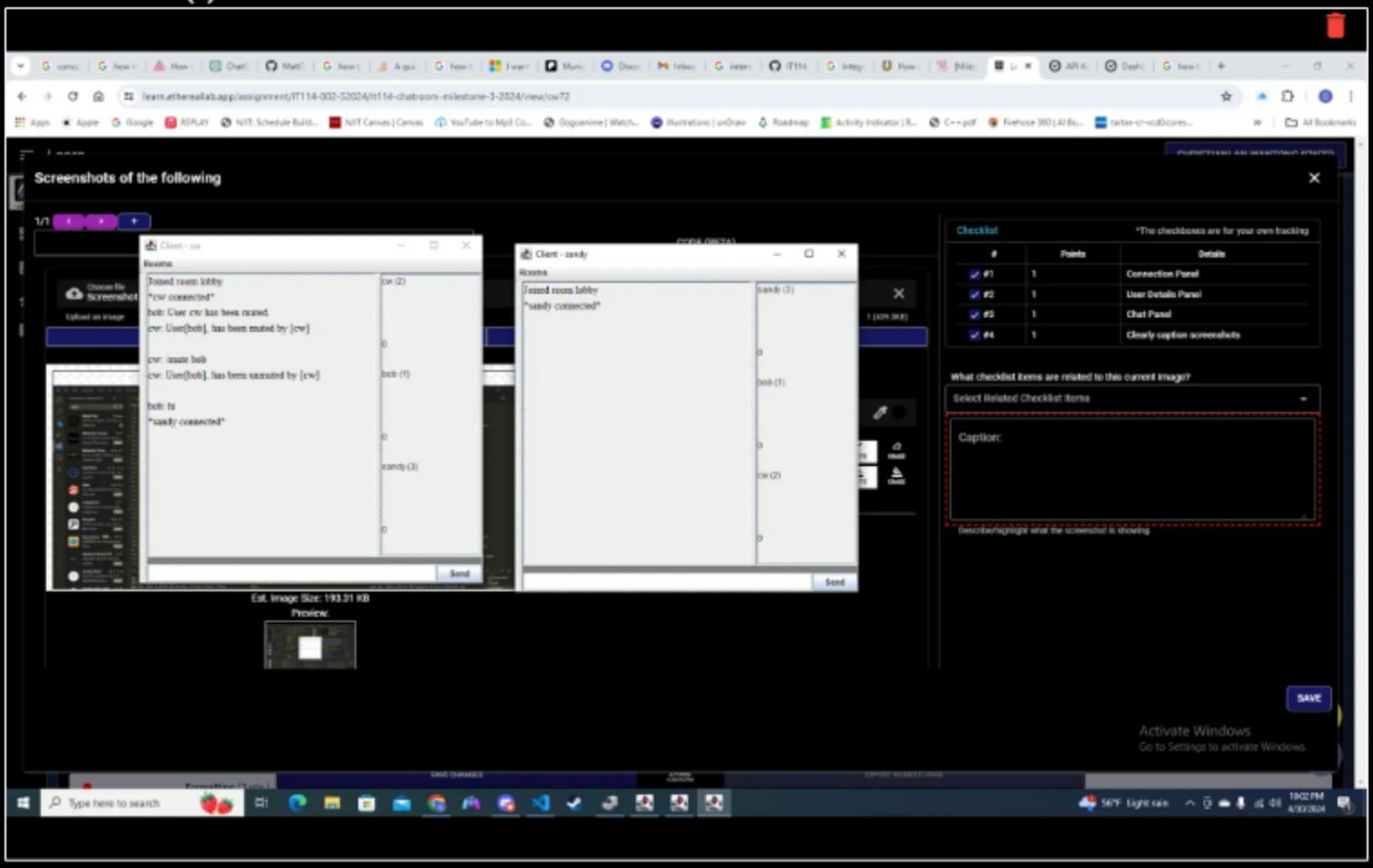
Checklist Items (0)





This is the screenshot of userDetails panel.

Checklist Items (0)



This is the screenshot of the caption panel.

Checklist Items (0)

Formatting (2 pts.)

COLLAPSE

COLLAPSE

Task #1 - Points: 1

Text: Screenshots demoing flip and roll commands

Checklist

*The checkboxes are for your own tracking

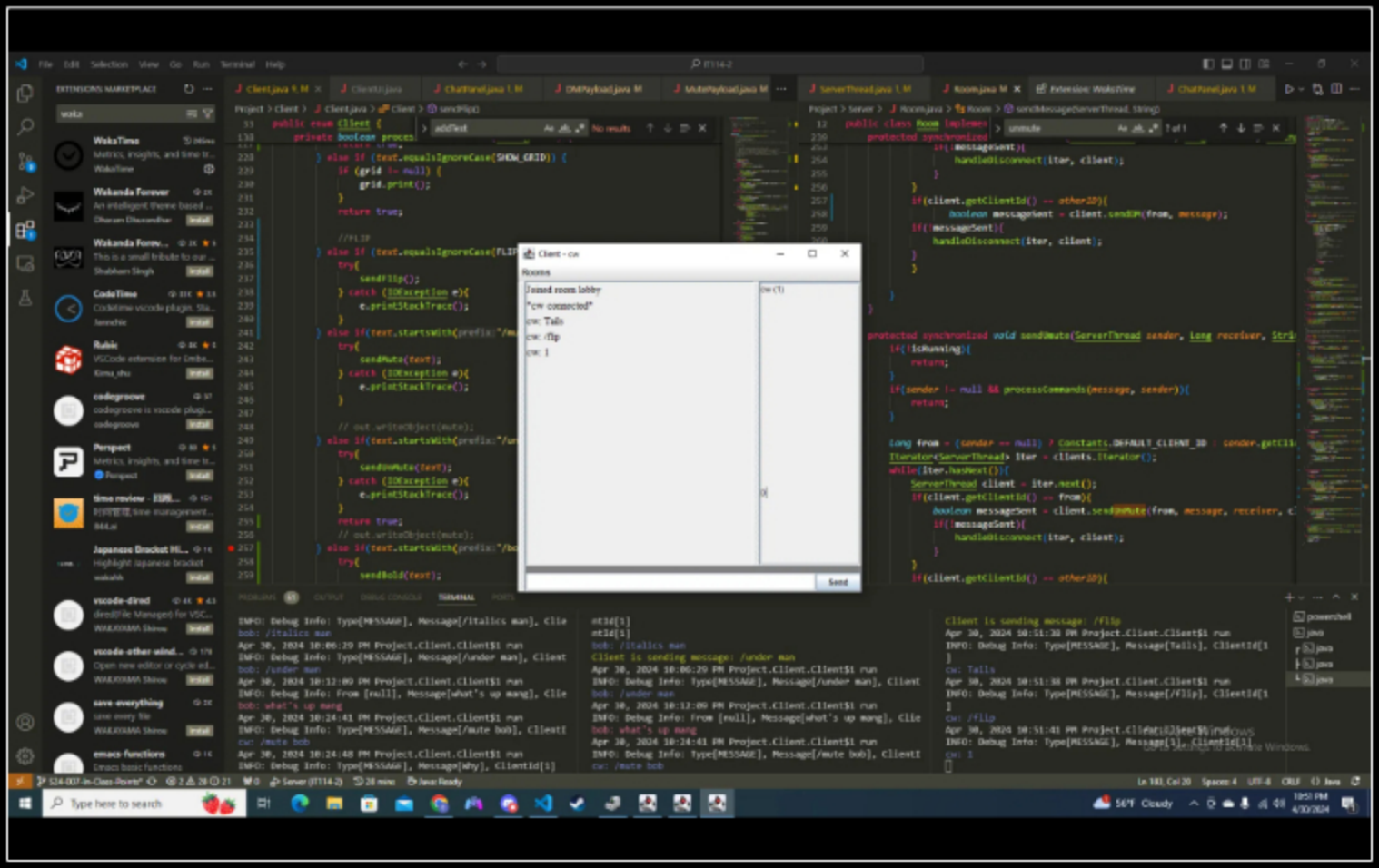
#	Points	Details
#1	1	Flip output in a different format than normal messages

#2	1	Roll # output in a different format than normal messages
#3	1	Roll #d# output in a different format than normal messages
#4	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large



These are the flip and roll commands

Checklist Items (4)

#1 Flip output in a different format than normal messages

#2 Roll # output in a different format than normal messages

#3 Roll #d# output in a different format than normal messages

#4 Clearly caption screenshots

COLLAPSE

Task #2 - Points: 1

Text: Screenshots demoing custom text formatting

Checklist

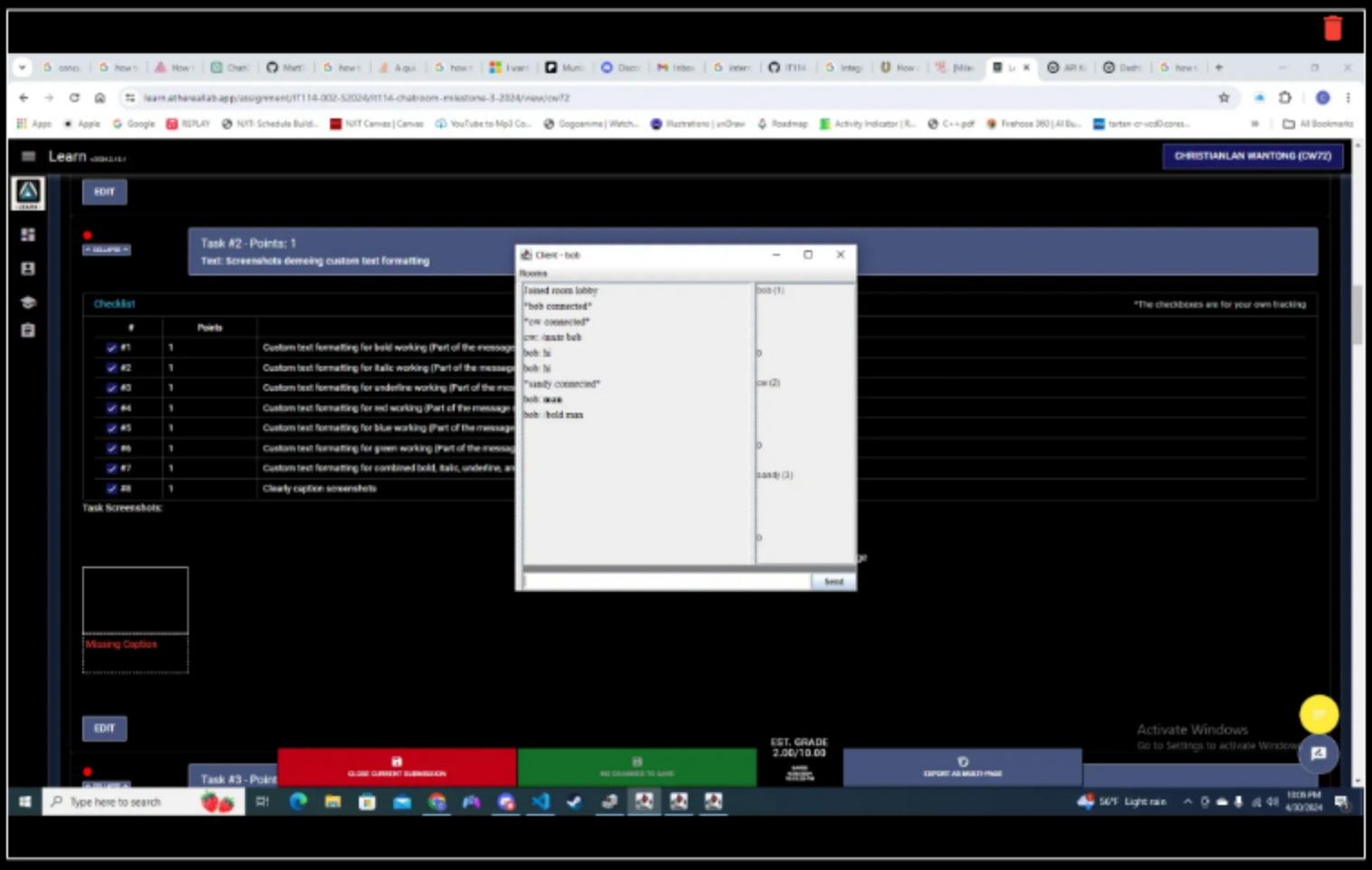
*The checkboxes are for your own tracking

#	Points	Details
#1	1	Custom text formatting for bold working (Part of the message should appear bold)
#2	1	Custom text formatting for italic working (Part of the message should appear italic)
#3	1	Custom text formatting for underline working (Part of the message should appear underline)
#4	1	Custom text formatting for red working (Part of the message should appear red)
#5	1	Custom text formatting for blue working (Part of the message should appear blue)
#6	1	Custom text formatting for green working (Part of the message should appear green)
#7	1	Custom text formatting for combined bold, italic, underline, and a color working (Part of the message should have all 4 formats applied at once)
#8	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

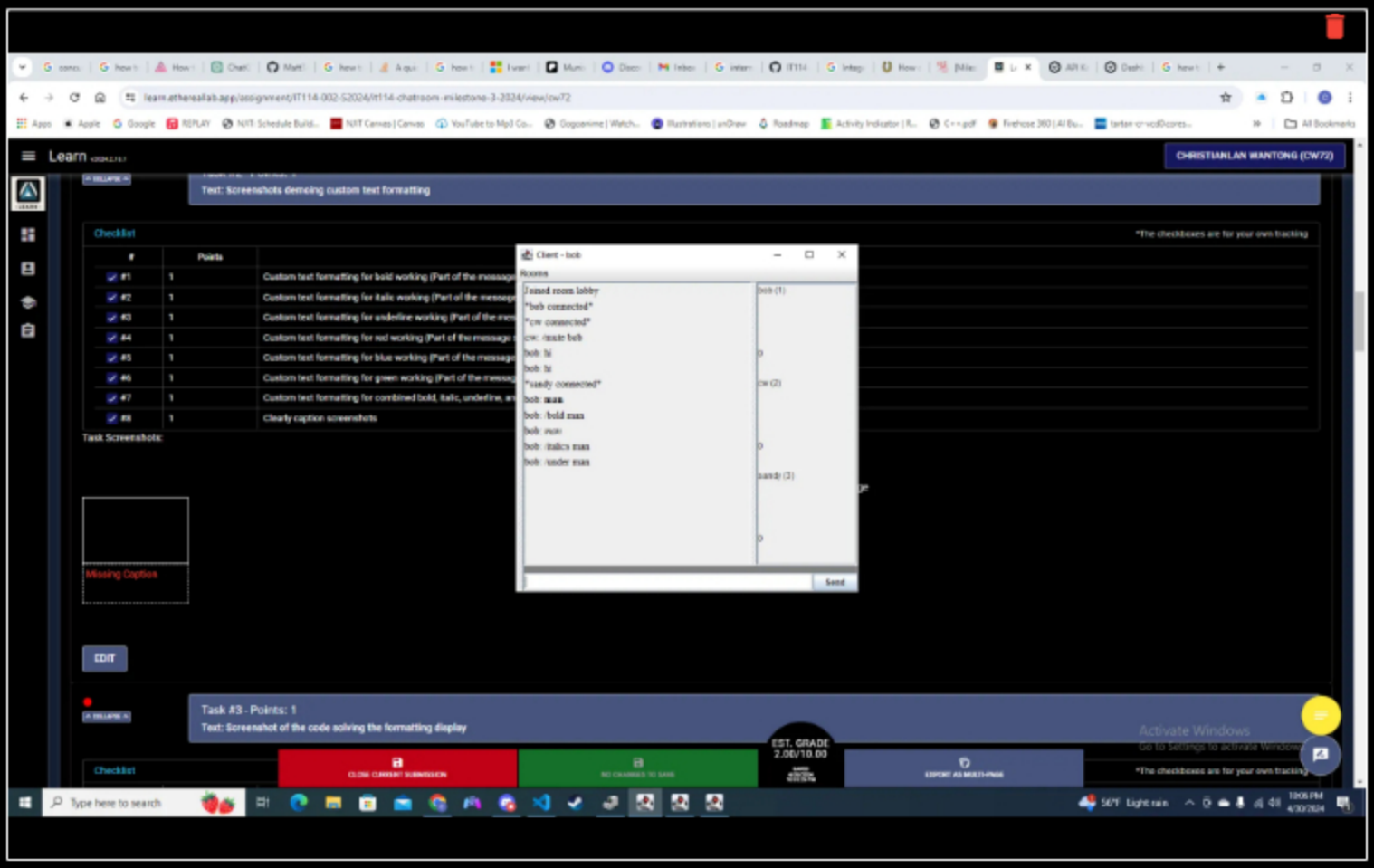
Small Medium Large



This is the screen of the bold text.

Checklist Items (1)

#1 Custom text formatting for bold working (Part of the message should appear bold)



Italic is working for this text

Checklist Items (1)

#2 Custom text formatting for italic working (Part of the message should appear italic)

^COLLAPSE^

Task #3 - Points: 1

Text: Screenshot of the code solving the formatting display

Checklist

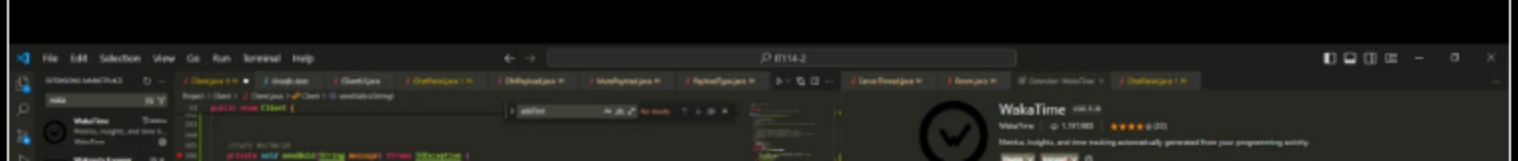
*The checkboxes are for your own tracking

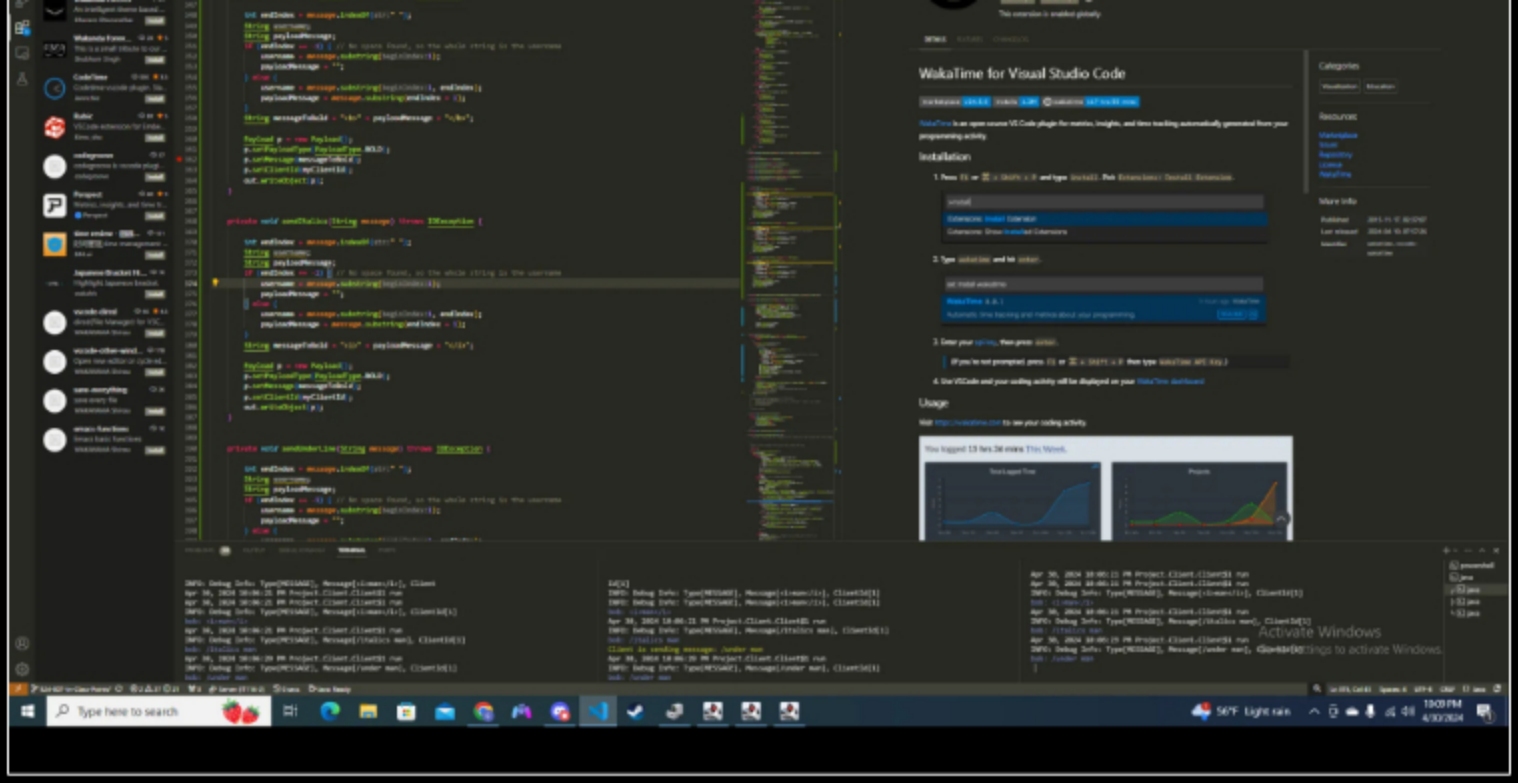
#	Points	Details
#1	1	Show each relevant file this was done in (may be one or more)
#2	1	Include ucid and date comment
#3	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large





Contains screen of relevant work done in file.

Checklist Items (3)

#1 Show each relevant file this was done in (may be one or more)

#2 Include ucid and date comment

#3 Clearly caption screenshots

COLLAPSE

Task #4 - Points: 1

Text: Explain how the formatting was made to be visible/rendered in the UI

Details:
Note each scenario

Response:

In order to make the text formatting appear in the UI. I wrapped the text in an html element and, JPanel I made sure that the textarea was on html so it was able to get the formatting.

Private Message with @ (2 pts.)

COLLAPSE

Task #1 - Points: 1

Text: Screenshots demoing private message

Checklist

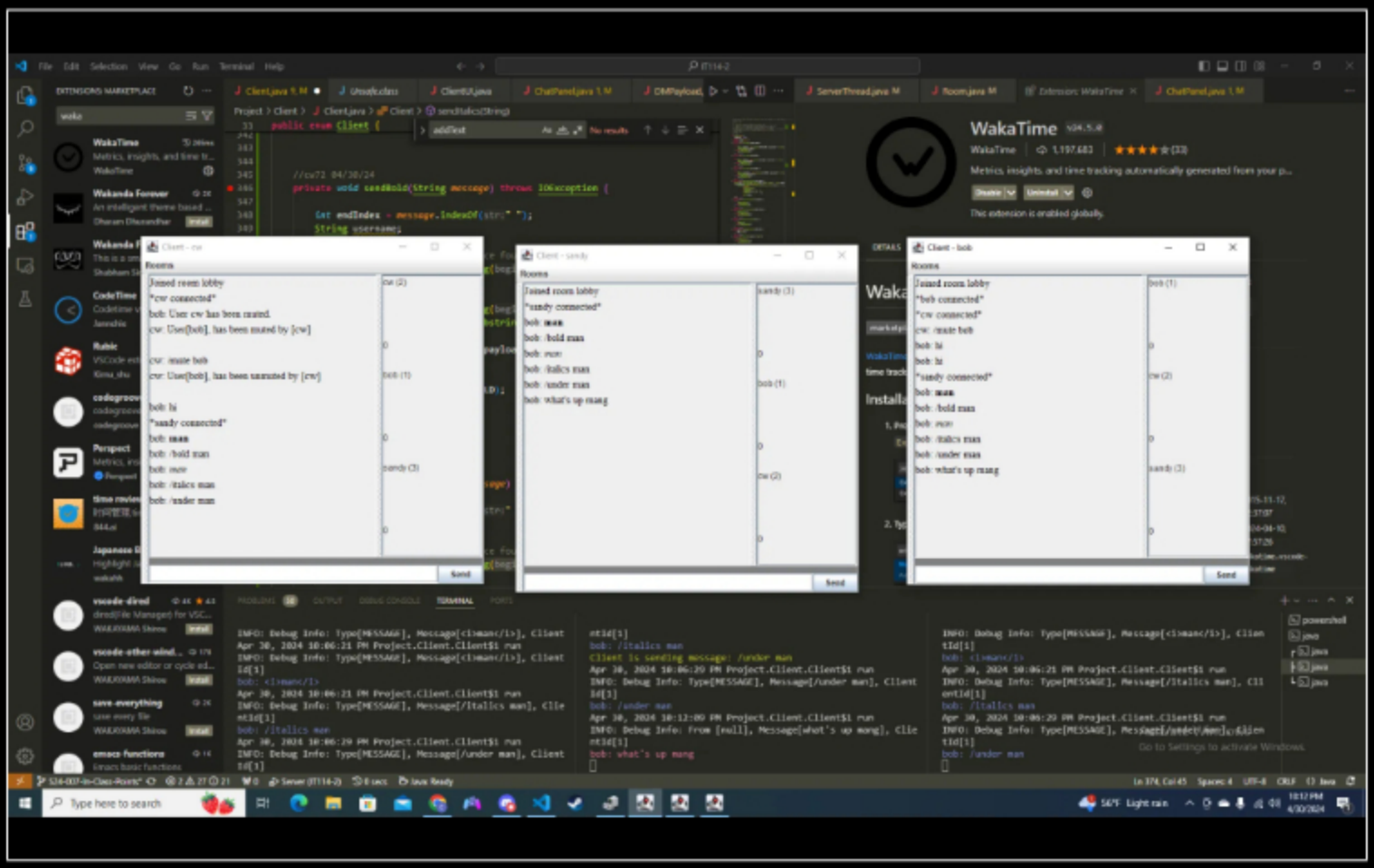
*The checkboxes are for your own tracking

#	Points	Details
#1	1	Should have 3 clients in the same room
#2	1	Demo a private message where only the sender and target see the message
#3	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large



Shows the that there is private message functionality.

Checklist Items (3)

#1 Should have 3 clients in the same room

#2 Demo a private message where only the sender and target see the message

#3 Clearly caption screenshots

COLLAPSE

Task #2 - Points: 1

Text: Screenshots of the related code

Checklist

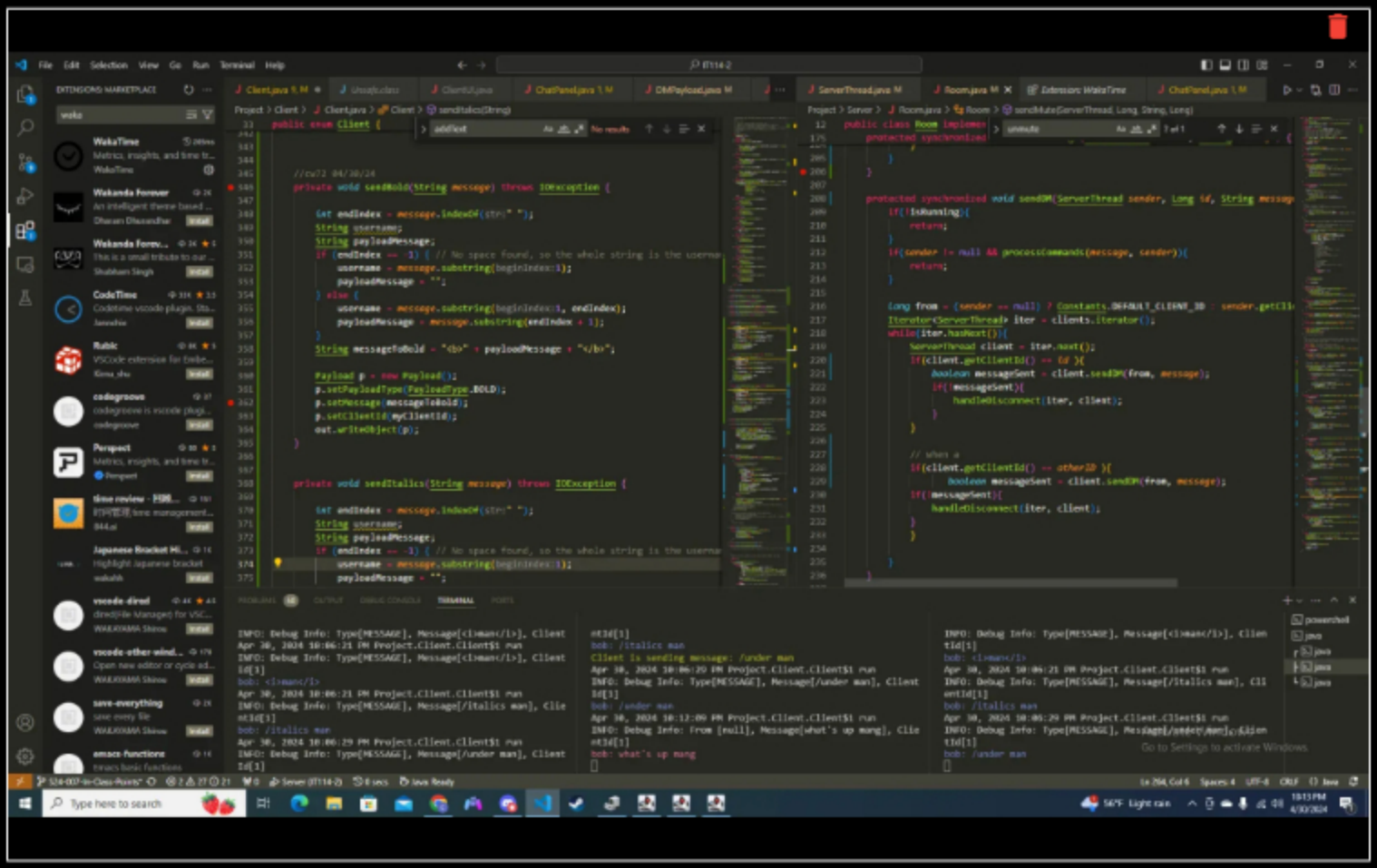
*The checkboxes are for your own tracking

#	Points	Details
#1	1	Show what code processes and handles the private message
#2	1	The message should only be sent to the receiver and the target
#3	1	The client should be targeting the username and the server side should be fetching the correct recipient
#4	1	Include ucid and date comment
#5	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large



Shows the logic in the Room.java that handles the private sending message functionality.

Checklist Items (5)

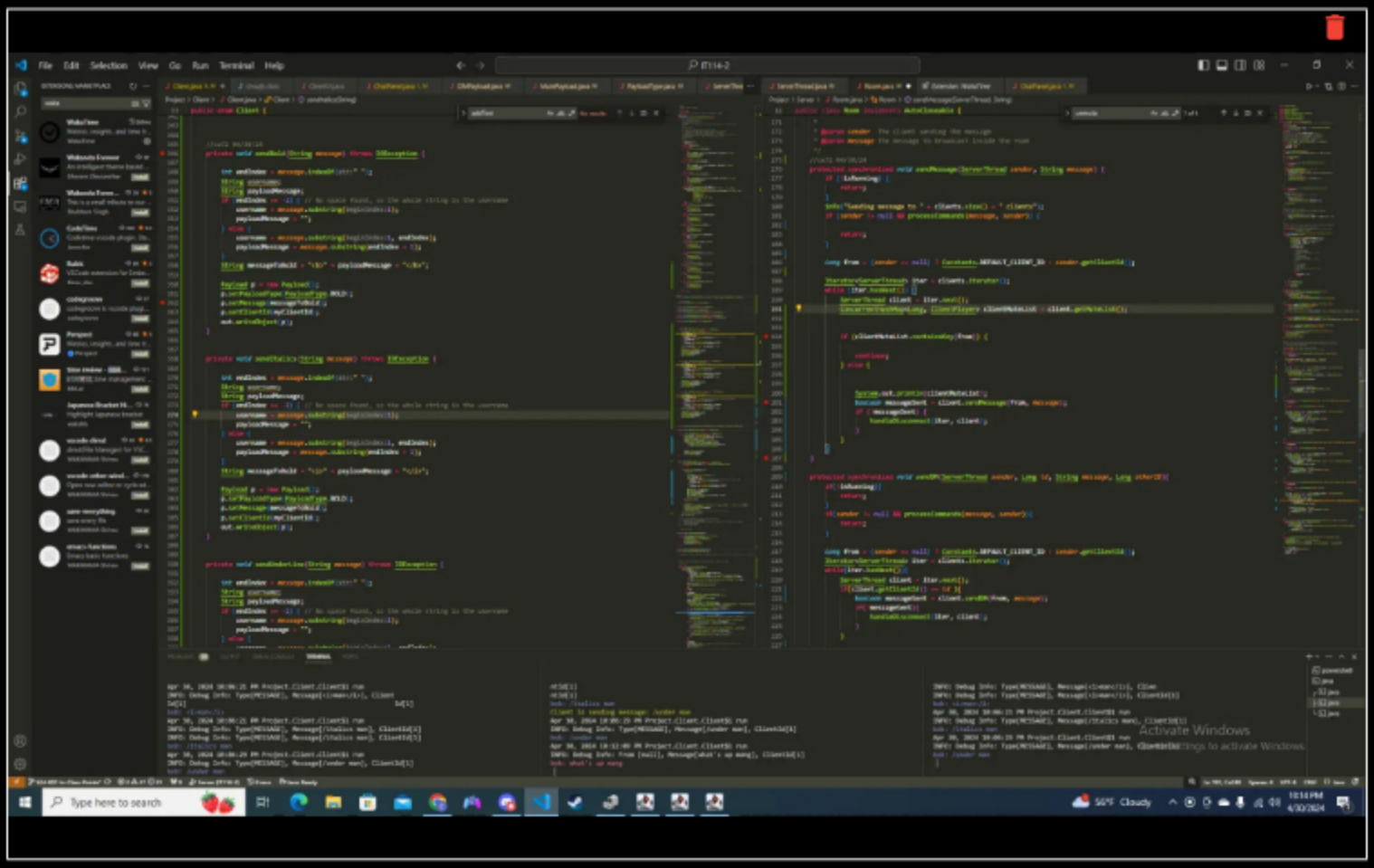
#1 Show what code processes and handles the private message

#2 The message should only be sent to the receiver and the target

#3 The client should be targeting the username and the server side should be fetching the correct recipient

#4 Include ucid and date comment

#5 Clearly caption screenshots



Part two of the screenshots that show private messaging functionality.

Checklist Items (5)

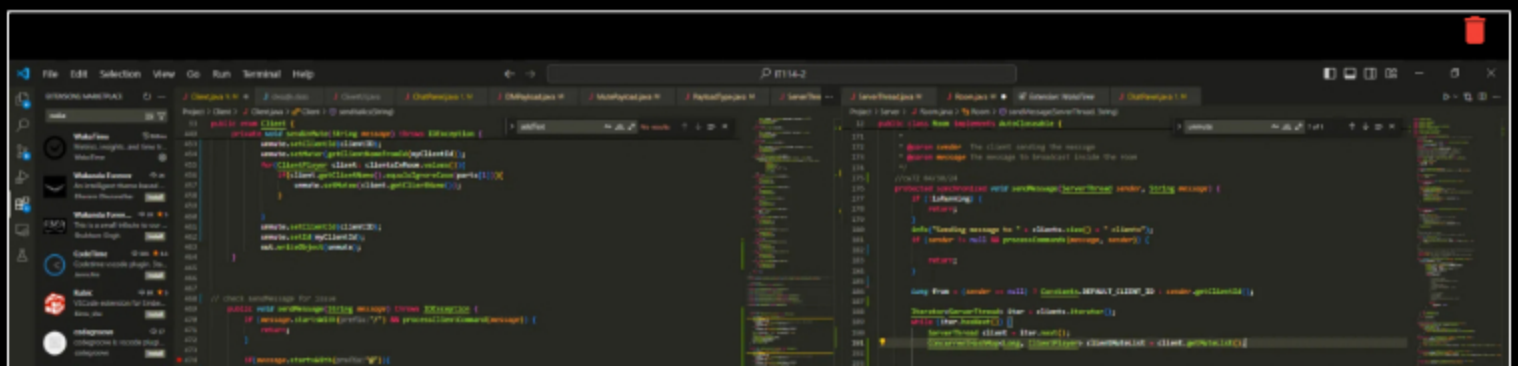
#1 Show what code processes and handles the private message

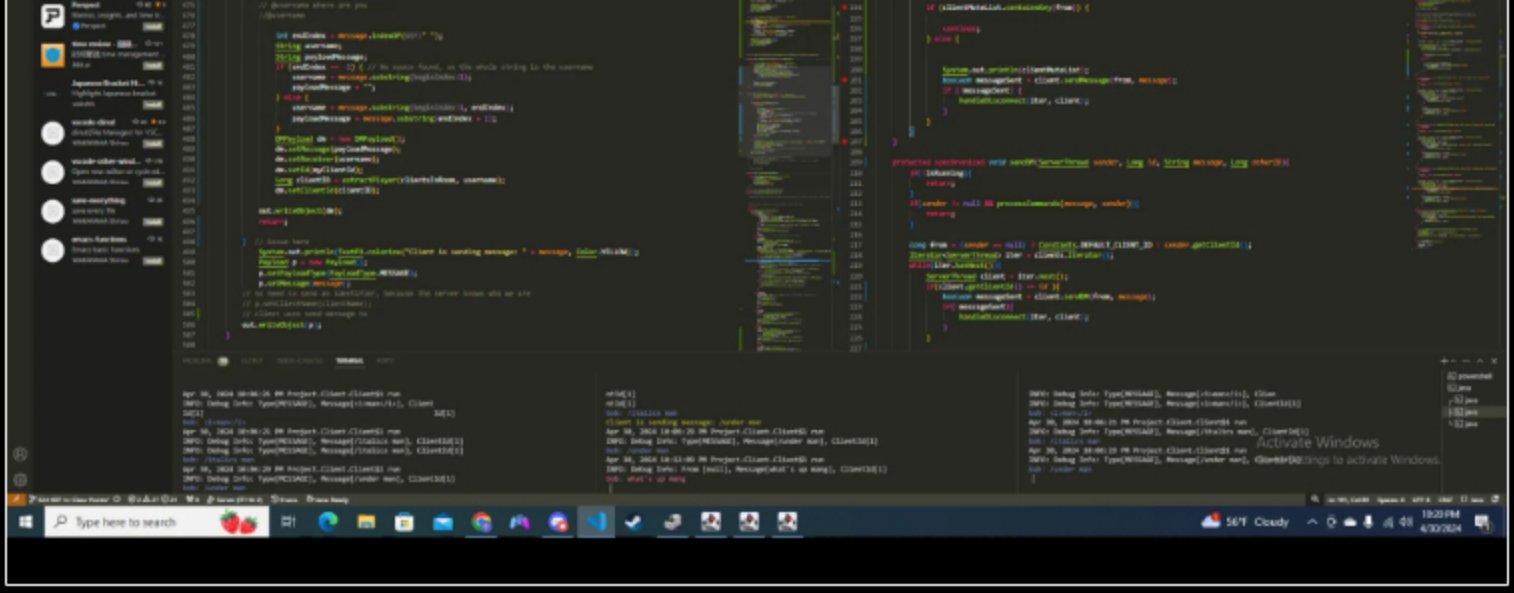
#2 The message should only be sent to the receiver and the target

#3 The client should be targeting the username and the server side should be fetching the correct recipient

#4 Include ucid and date comment

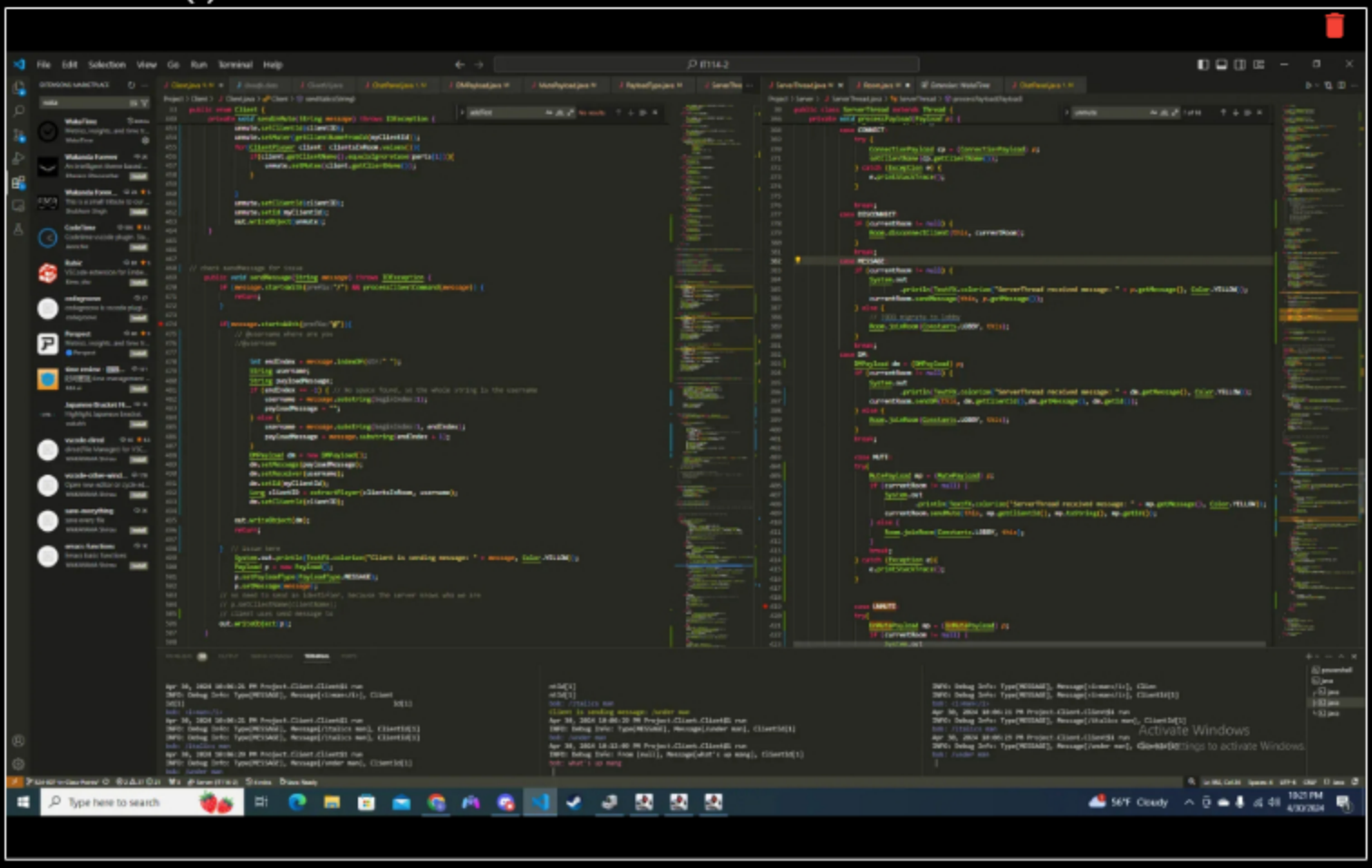
#5 Clearly caption screenshots





Missing Caption

Checklist Items (0)



Logic in the client that explains how to send the private messages to the ServerThread

Checklist Items (5)

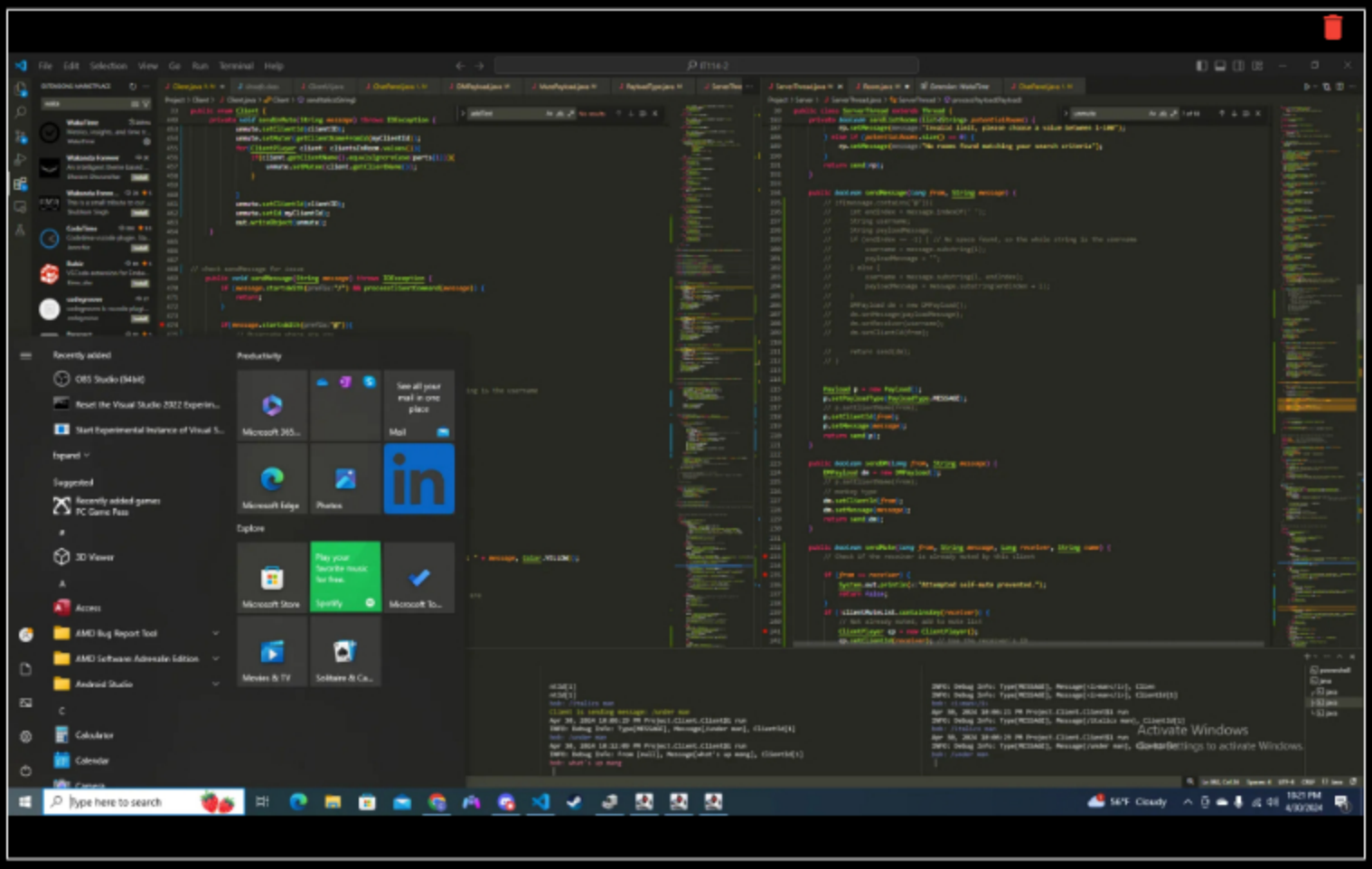
#1 Show what code processes and handles the private message

#2 The message should only be sent to the receiver and the target

#3 The client should be targeting the username and the server side should be fetching the correct recipient

#4 Include ucid and date comment

#5 Clearly caption screenshots



Shows the function being called in the ServerThread.jav a

Checklist Items (5)

#1 Show what code processes and handles the private message

#2 The message should only be sent to the receiver and the target

#3 The client should be targeting the username and the server side should be fetching the correct recipient

#4 Include ucid and date comment

#5 Clearly caption screenshots

COLLAPSE

Task #3 - Points: 1

Text: Explain how private message works related to the code above

Checklist

*The checkboxes are for your own tracking

#	Points	Details

#1	1	Include how the sender and receiver are handled
#2	1	Include how the username is used to get the proper id

Response:

It was a very complicated process. But essentially a payload was sent with PayloadType DM. That payload was sent from Client.java ui, and it check for "@" symbol. In ServerThread it processes the payload and calls the Room.java sendDM function, there I take information from I extract the client ID from which was set in the Client.java sendMessage function by looping through the clientsInRoom and I check if the receiver is comparing client.getClientId() to the receiver, if it is I then send the message to that client.

Mute/Unmute Users (3 pts.)

COLLAPSE

COLLAPSE

Task #1 - Points: 1

Text: Screenshots demoing feature working

Checklist

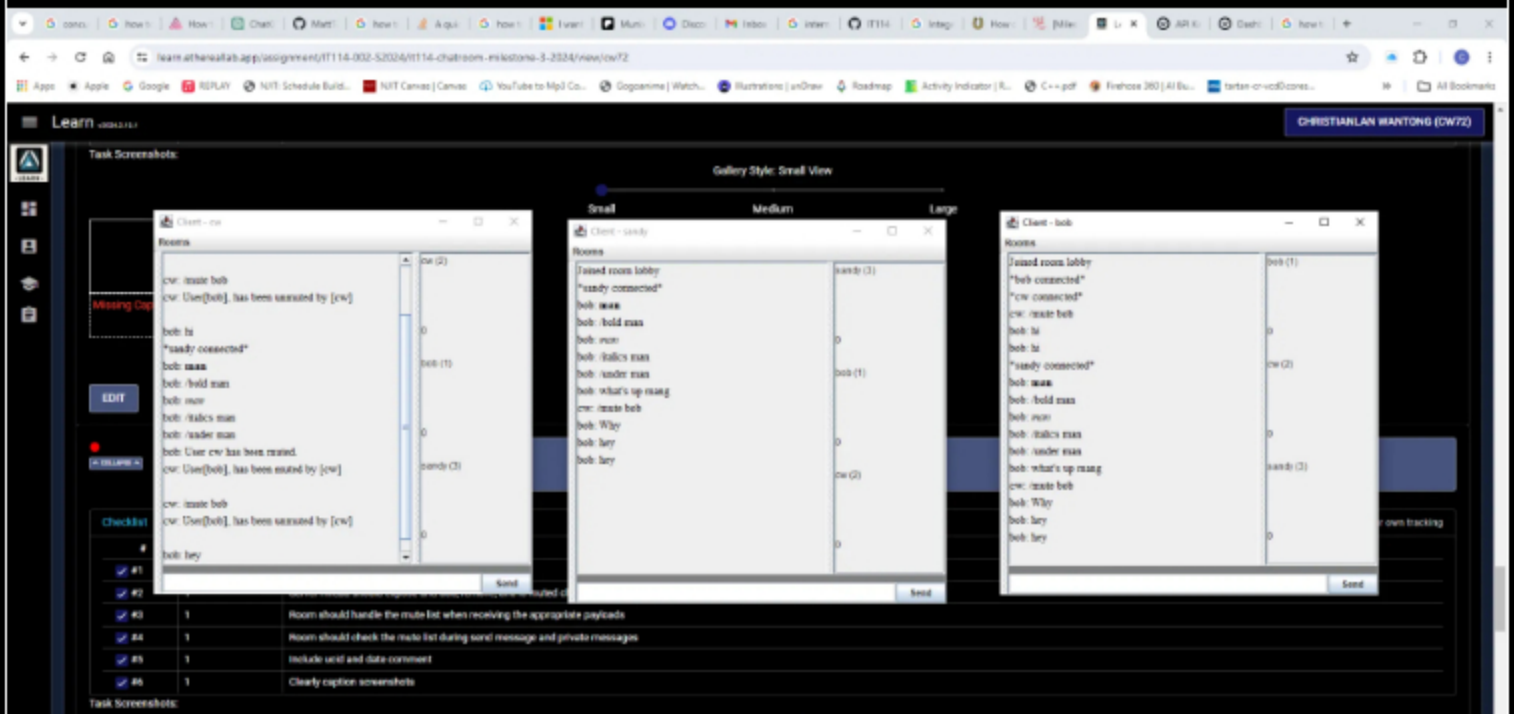
*The checkboxes are for your own tracking

#	Points	Details
#1	1	Should have 3 clients in the same room
#2	1	Demo mute preventing messages between the muter and the target
#3	1	Demo mute also being accounted for with private messages
#4	1	Demo unmute allowing the messages again from the target to the unmuter

Task Screenshots:

Gallery Style: Large View

Small Medium Large



Shows the mute and unmute function being used.

Checklist Items (4)

#1 Should have 3 clients in the same room

#2 Demo mute preventing messages between the muter and the target

#3 Demo mute also being accounted for with private messages

#4 Demo unmute allowing the messages again from the target to the unmuter

COLLAPSE

Task #2 - Points: 1

Text: Screenshots of the related code

Checklist

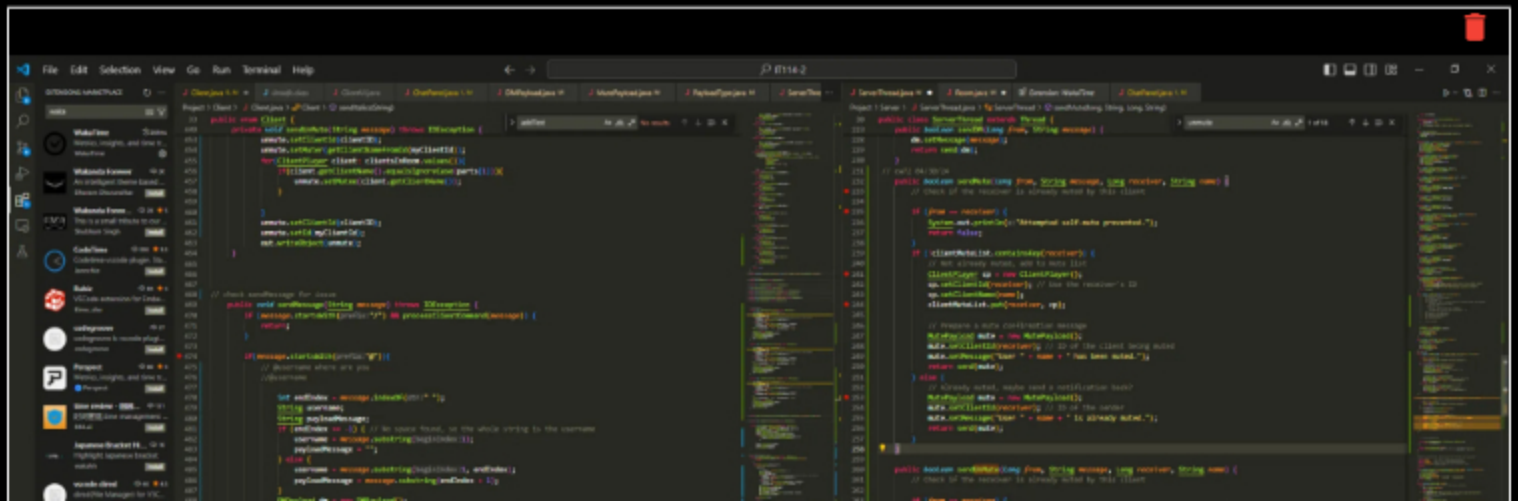
*The checkboxes are for your own tracking

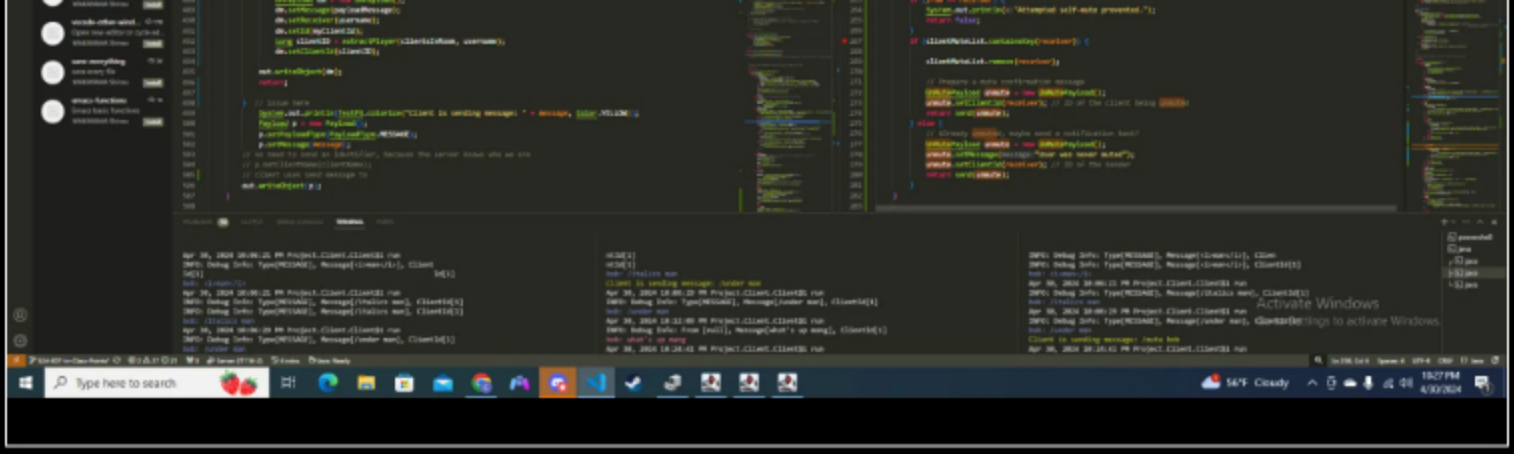
#	Points	Details
#1	1	ServerThread should have a list of who they muted
#2	1	ServerThread should expose and add, remove, and is muted check to room
#3	1	Room should handle the mute list when receiving the appropriate payloads
#4	1	Room should check the mute list during send message and private messages
#5	1	Include ucid and date comment
#6	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

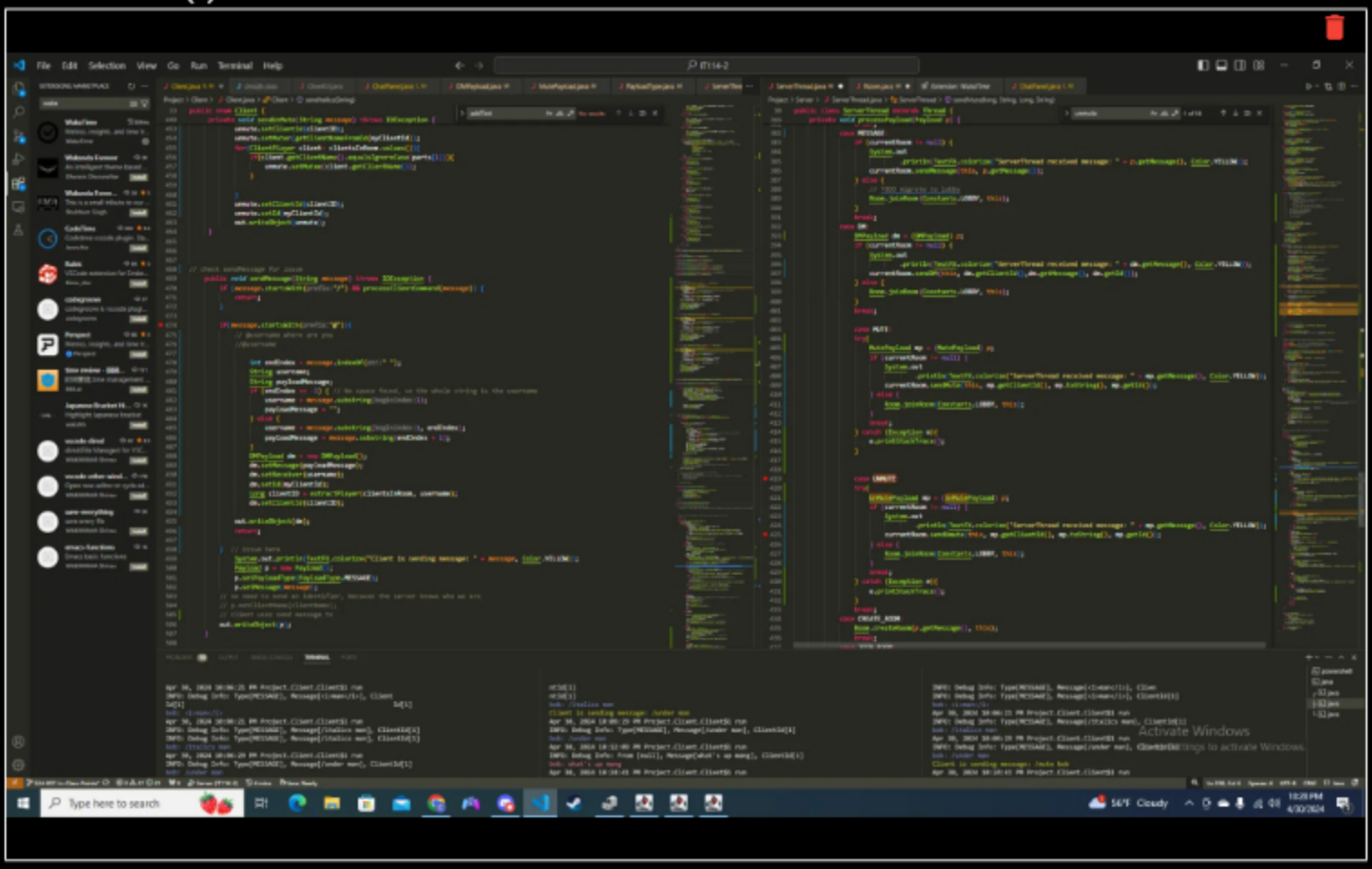
Small Medium Large





Shows the list of clients that are muted, How that room should handle payloads.

Checklist Items (0)



Continuation of the first screenshot.

Checklist Items (6)

#1 ServerThread should have a list of who they muted

#2 ServerThread should expose and add, remove, and is muted check to room

#3 Room should handle the mute list when receiving the appropriate payloads

#4 Room should check the mute list during send message and private messages

#5 Include ucid and date comment

#6 Clearly caption screenshots

COLLAPSE

Task #3 - Points: 1

Text: Explain how the mute and unmute logic works in relation to the code

Checklist *The checkboxes are for your own tracking

#	Points	Details
#1	1	Explain how your mute list is handled
#2	1	Explain how it's handled/processed in send message and private message

Response:

There was a command trigger in the client, in the client function both the muter and mutee clientId was added to the payload. Then in server thread, there was new list called clientsInMuteList and I added functionality to check if the clientID was in the list, if not I added the client Id, as well as the client name to the clientPlayer object in the map. Then when In Room.java, in the send message function, I iterated through the send message function and then iterated through the clientsInMuteList for each client and check if the sender was in the mute list and if so I just continued. For the unmute function it was similar logic but in the ServerThread I check if the client ID was in the mute list and if so I remove the key and the value from the map.

Misc (1 pt.)

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/cdubbx/IT114/pull/1

COLLAPSE

Task #2 - Points: 1

Text: Talk about any issues or learnings during this assignment

Response:

I believe the assignment was pretty robust, I feel like if I began the assignment back in February it would've given me and other people more time to complete the assignment. I believe the there was way too much to be able to understand. It is pretty realistic because if you want to become a program you need learn on how to adapt to a code

understand. It is pretty realistic because if you want to become a program you need learn on how to adapt to a code base.

COLLAPSE

Task #3 - Points: 1

Text: WakaTime Screenshot

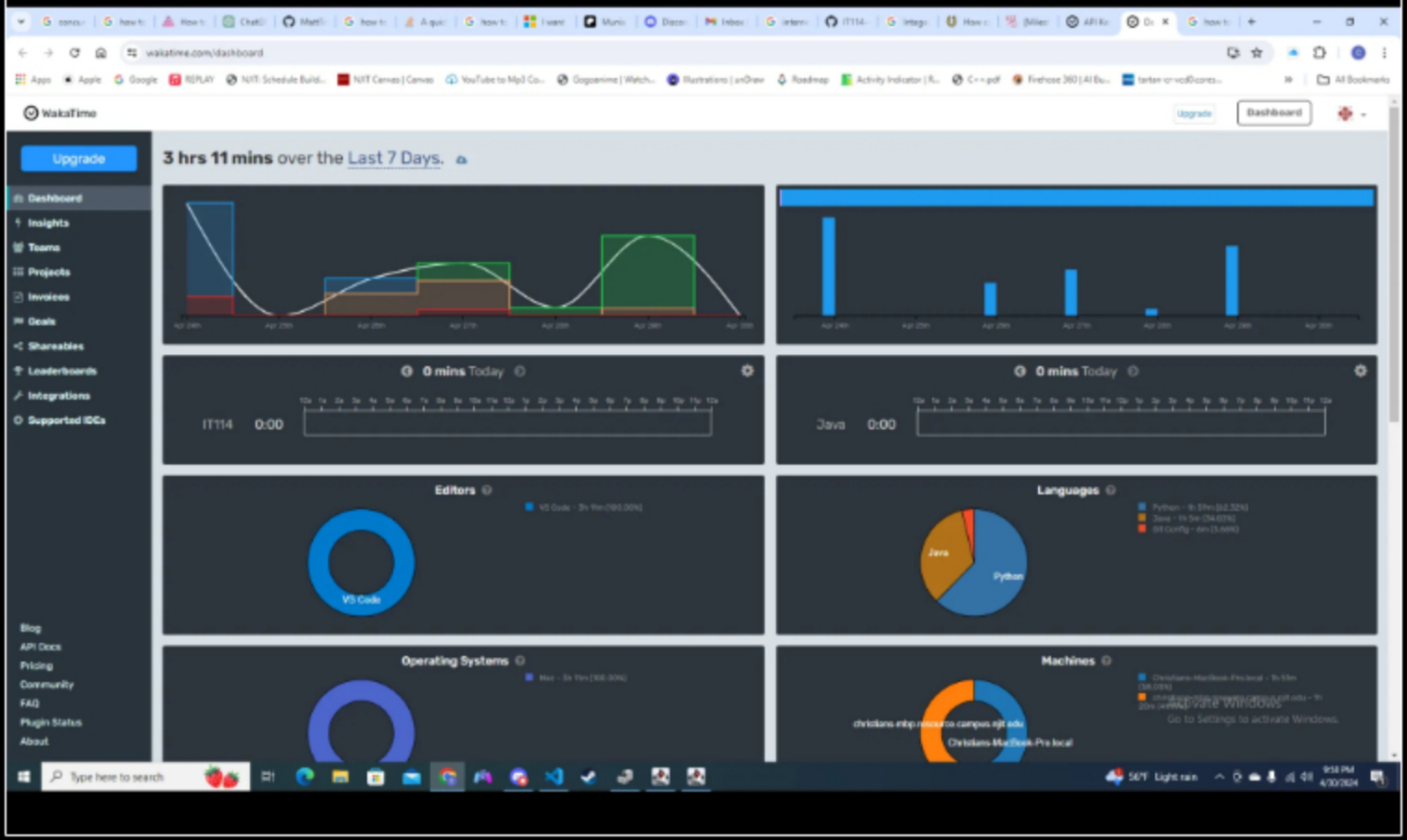
Details:

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



It is not an accurate representation because I did not have waka time on this computer and I have two computers.

End of Assignment