Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-002-S2024/it114-project-milestone-1/grade/am3485

IT114-002-S2024 - [IT114] Project Milestone 1

Submissions:

Submission Selection

1 Submission [active] 4/17/2024 9:25:18 AM

Instructions

COLLAPSE

1. Create a new branch called Milestone1
2. At the root of your repository create a folder called Project if one doesn't exist yet
    1. You will be updating this folder with new code as you do milestones
    2. You won't be creating separate folders for milestones; milestones are just branches
3. Create a pull request from Milestone1 to main (don't complete/merge it yet, just have it in open status)
4. Copy in the latest Socket sample code from the most recent Socket Part example of the lessons
    1. Recommended Part 5 (clients should be having names at this point and not ids)
    2. https://github.com/MattToegel/IT114/tree/Module5/Module5
5. Fix the package references at the top of each file (these are the only edits you should do at this point)
6. Git add/commit the baseline and push it to github
7. Create a pull request from Milestone1 to main (don't complete/merge it yet, just have it in open status)
8. Ensure the sample is working and fill in the below deliverables
    1. Note: The client commands likely are different in part 5 with the /name and /connect options instead of just "connect"
9. Generate the worksheet output file once done and add it to your local repository
10. Git add/commit/push all changes
11. Complete the pull request merge from step 7
12. Locally checkout main
13. git pull origin main

Branch name: Milestone1

Tasks: 9 Points: 10.00

Start Up (3 pts.)

COLLAPSE

COLLAPSE

Task #1 - Points: 1

Text: Server and Client Initialization

Checklist

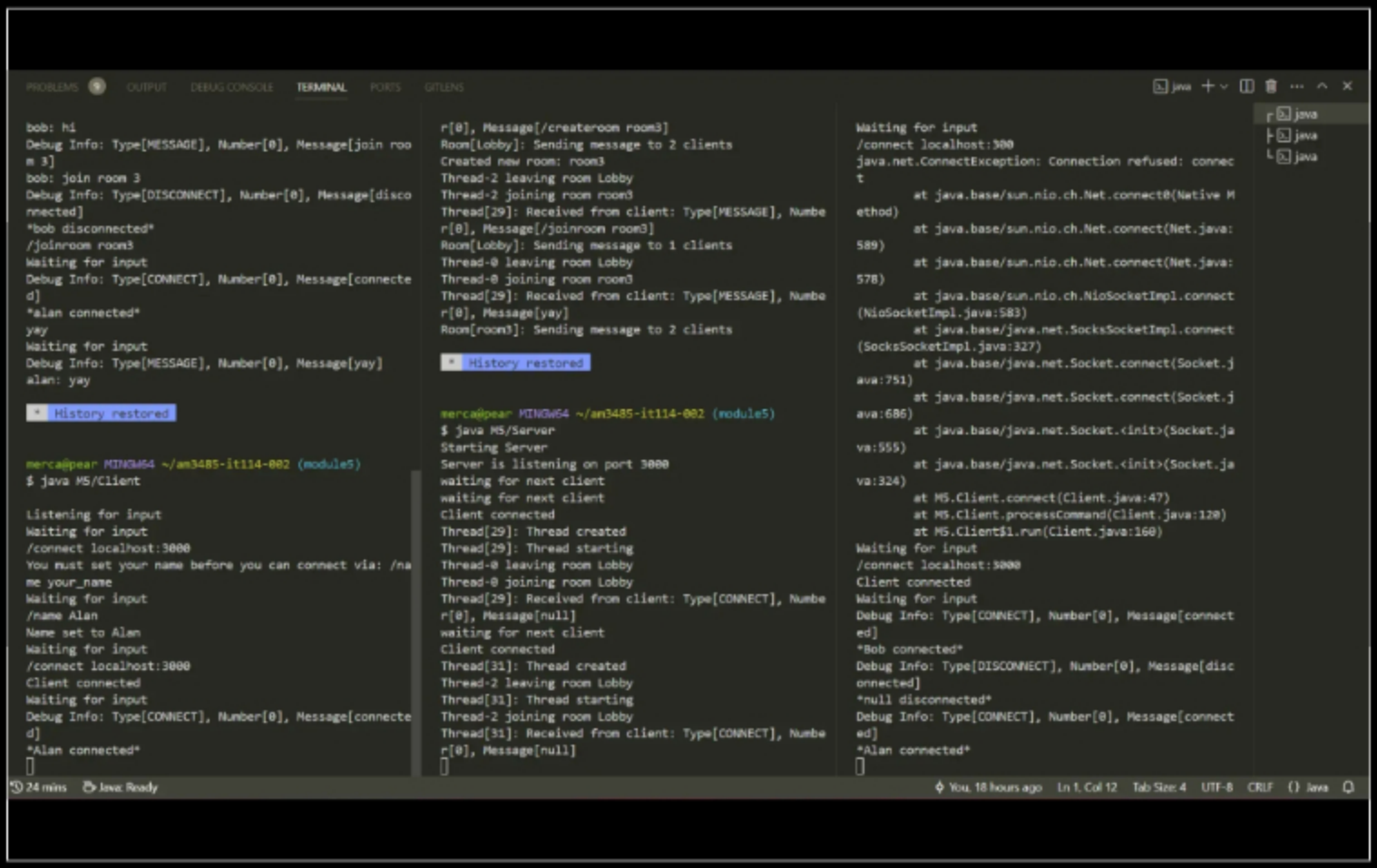
*The checkboxes are for your own tracking

#	Points	Details
☐ #1	1	Server should properly be listening to its port from the command line (note the related message)
☐ #2	1	Clients should be successfully waiting for input
☐ #3	1	Clients should have a name and successfully connected to the server (note related messages)

Task Screenshots:

Gallery Style: Large View

Small Medium Large



the screen shot

Checklist Items (0)

COLLAPSE

Task #2 - Points: 1

Text: Explain the connection process

Details:

Note the various steps from the beginning to when the client is fully connected and able to communicate in the room.

Emphasize the code flow and the sockets usage.

Checklist

*The checkboxes are for your own tracking

#	Points	Details
☐ #1	1	Mention how the server-side of the connection works
☐ #2	1	Mention how the client-side of the connection works
☐ #3	1	Describe the socket steps until the server is waiting for messages from the client

Response:

In order to connect the client and the server you must first make sure each part is running then from the server type/ connect localhost:3000, localhost is how it sounds and means you are hosting it locally and the 3000 part is a port number that is usually available. next name the two clients and then connect both to the same port as the server.

Communication (3 pts.)

COLLAPSE

COLLAPSE

Task #1 - Points: 1

Text: Add screenshot(s) showing evidence related to the checklist

Checklist

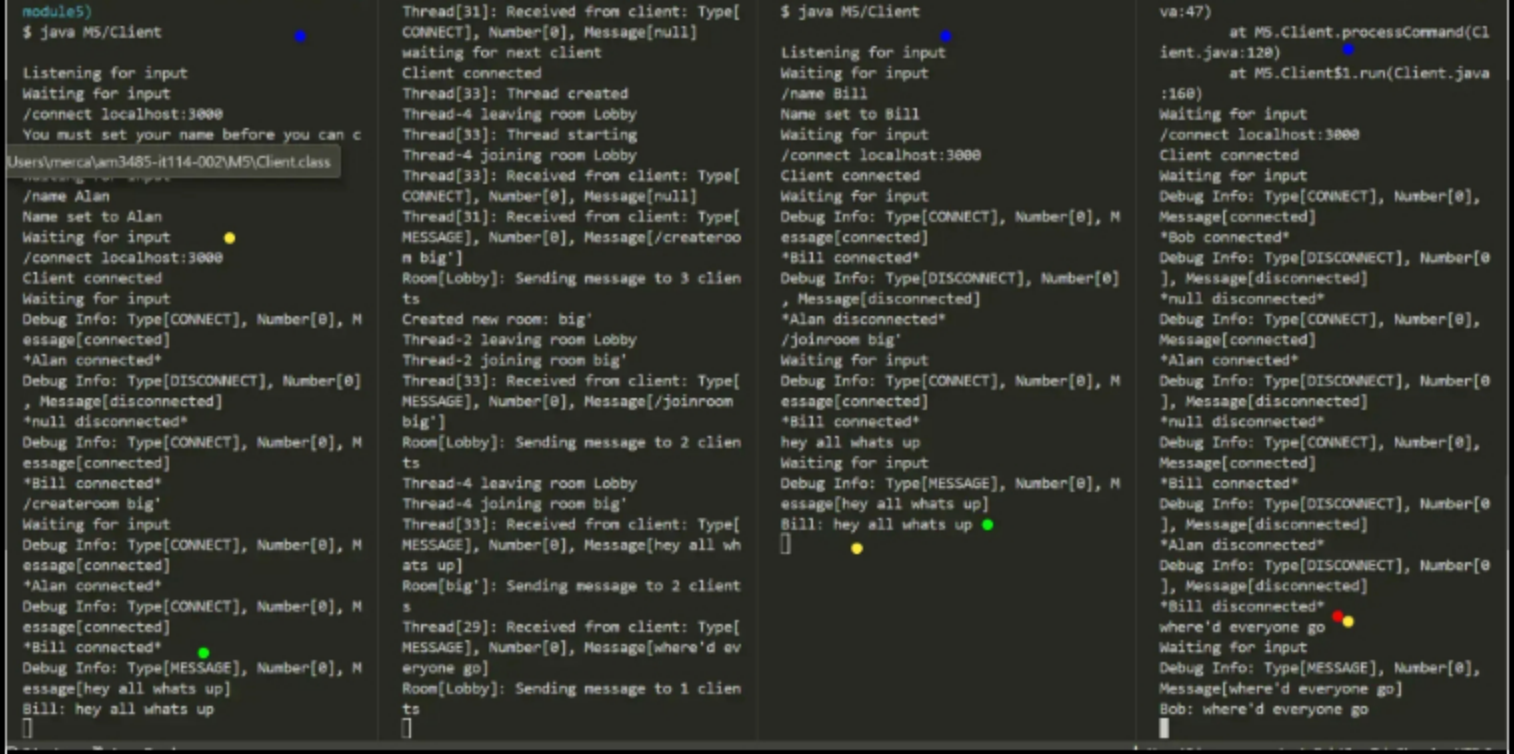
*The checkboxes are for your own tracking

#	Points	Details
☐ #1	1	At least two clients connected to the server
☐ #2	1	Client can send messages to the server
☐ #3	1	Server sends the message to all clients in the same room
☐ #4	1	Messages clearly show who the message is from (i.e., client name is clearly with the message)
☐ #5	2	Demonstrate clients in two different rooms can't send/receive messages to each other (clearly show the clients are in different rooms via the commands demonstrated in the lessons
☐ #6	1	Clearly caption each image regarding what is being shown

Task Screenshots:

Gallery Style: Large View

Small Medium Large



red = message shown blue = proves connection green = shows messages being sent from inside the same room yellow = shows different rooms can't see each others messages

Checklist Items (0)

COLLAPSE

Task #2 - Points: 1

Text: Explain the communication process

Details:

How are messages entered from the client side and how do they propagate to other clients?

Note all the steps involved and use specific terminology from the code.
Don't just translate the code line-by-line to plain English, keep it concise.

Checklist

*The checkboxes are for your own tracking

#	Points	Details
☐ #1	1	Mention the client-side (sending)
☐ #2	1	Mention the ServerThread's involvement
☐ #3	1	Mention the Room's perspective
☐ #4	1	Mention the client-side (receiving)

Response:

from the client side all that you have to do is type something out and press enter, on the server thread side its forwards the message so people can see it, on the room side it makes it so that only people in the same room can see a message, and the receiving side they are displayed the message and who its from.

Disconnecting/Termination (3 pts.)

COLLAPSE

COLLAPSE

Task #1 - Points: 1

Text: Add screenshot(s) showing evidence related to the checklist

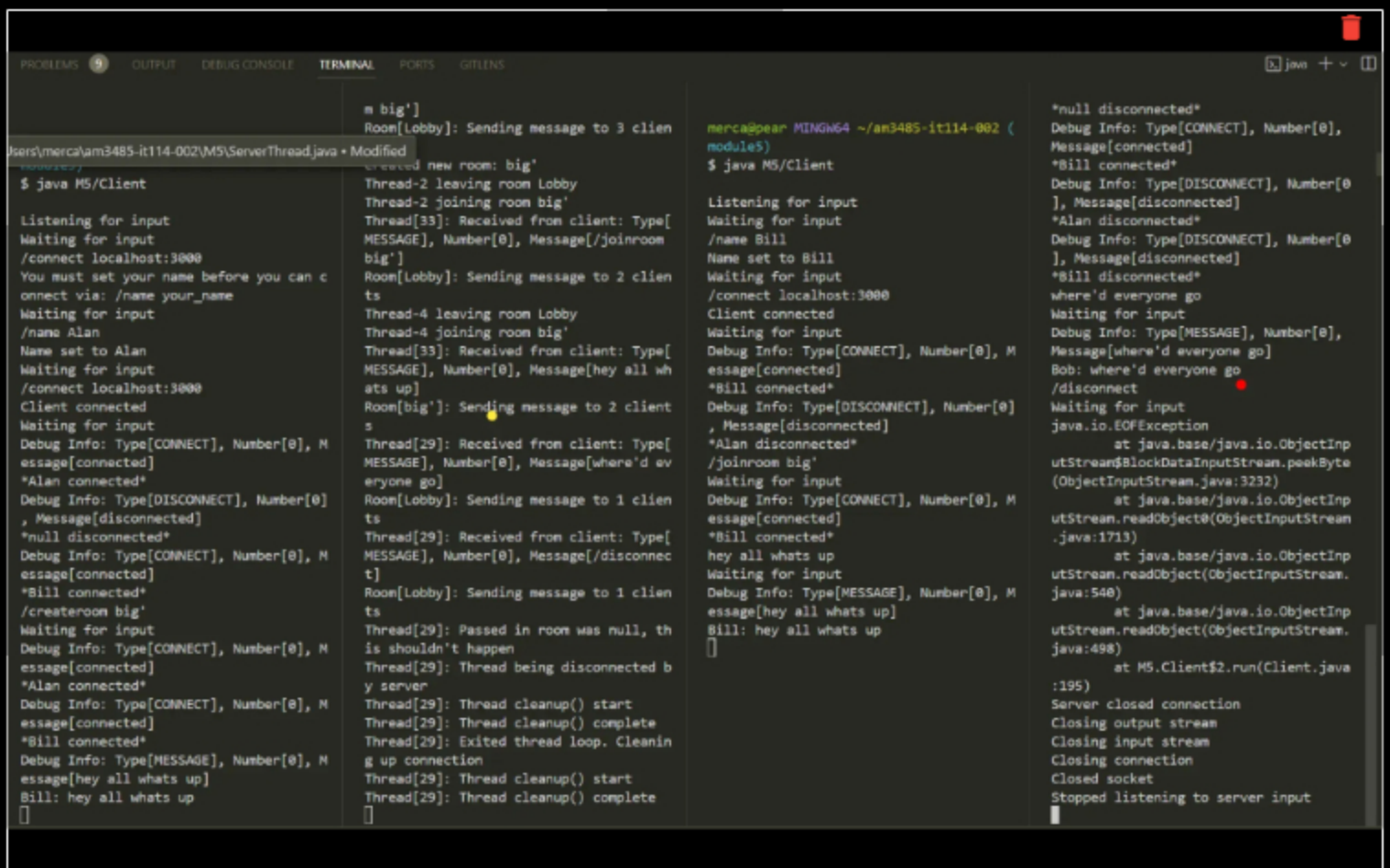
Checklist *The checkboxes are for your own tracking

#	Points	Details
☐ #1	1	Show a client disconnecting from the server; Server should still be running without issue (it's ok if an exception message shows as it's part of the lesson code, the server just shouldn't terminate)
☐ #2	1	Show the server terminating; Clients should be disconnected but still running and able to reconnect when the server is back online (demonstrate this)
☐ #3	1	For each scenario, disconnected messages should be shown to the clients (should show a different person disconnected and should show the specific client disconnected)
☐ #4	1	Clearly caption each image regarding what is being shown

Task Screenshots:

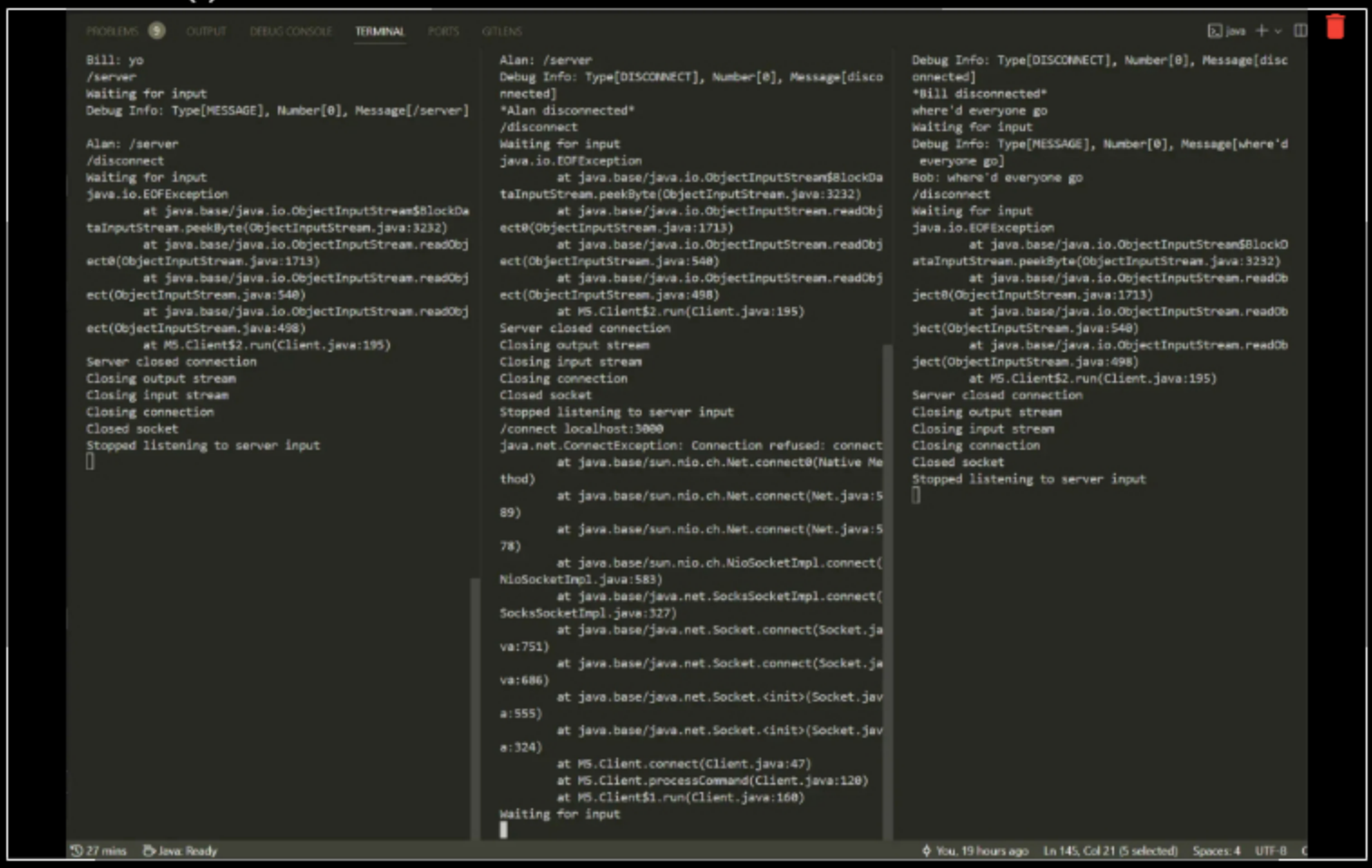
Gallery Style: Large View

Small Medium Large



red disconnected from yellow

Checklist Items (0)



when server disconnects

Checklist Items (0)

COLLAPSE

Task #2 - Points: 1

Text: Explain the various Disconnect/termination scenarios

Details:
Include the various scenarios of how a disconnect can occur. There should be around 3 or so.

Checklist

*The checkboxes are for your own tracking

#	Points	Details
☐ #1	1	Mention how a client gets disconnected from a Socket perspective
☐ #2	1	Mention how/why the client program doesn't crash when the server disconnects/terminates.
☐ #3	1	Mention how the server doesn't crash from the client(s) disconnecting

Response:

the scenarios of a disconnect are quiting the client, terminating the server, or pressing control C from either the client or the server

one way to terminate is to x the instance of the terminal out or close it lets say you did this from the server side it wouldn't crash the client because of the check connection function that uses a try catch loop to stop this from happening so you don't have to rerun the program anytime a disconnect might happen the client will be set back to waiting for input like when you start
another way to terimnate is by pressing control c in the client or server lets say you do this on the client side the server wouldn't crash because of a try catch loop that would catch when a client disconnects and send the server back to the same point where it started as waiting for client and all the server threads and rooms will be closed if nobody is using them

Misc (1 pt.)

COLLAPSE

COLLAPSE

Task #1 - Points: 1

Text: Add the pull request link for this branch

URL #1
https://github.com/alanpear/am3485-it114-002/pull/10

COLLAPSE

Task #2 - Points: 1

Text: Talk about any issues or learnings during this assignment

Details:

Few related sentences about the Project/sockets topics

Response:

I had no issues with this milestone it was more of an assignment that helped me get familiar with the process and helped me recount what i learned in class.

COLLAPSE

Task #3 - Points: 1

Text: WakaTime Screenshot

Details:

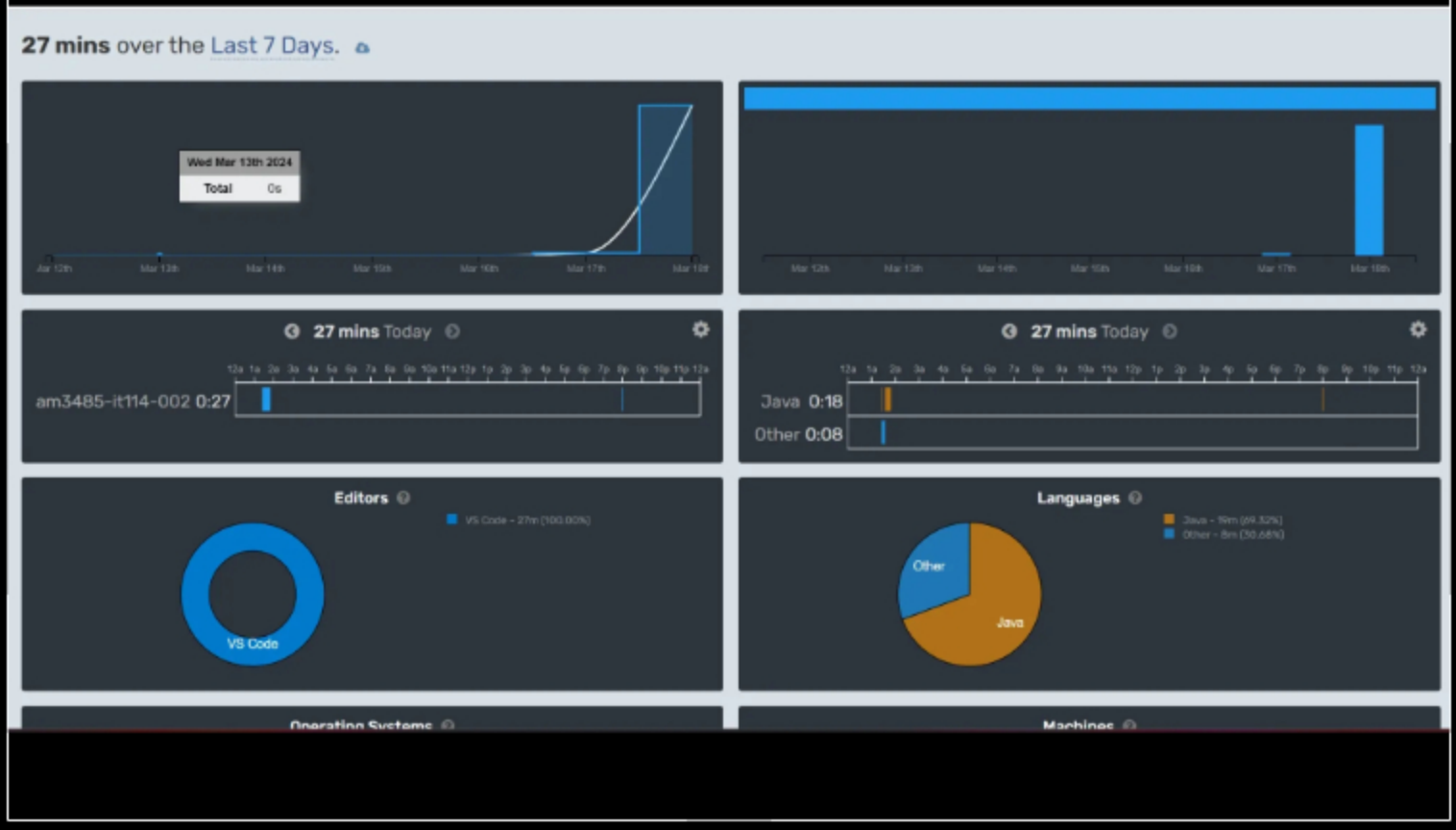
Grab a snippet showing the approximate time involved that clearly shows your repository.

The duration isn't considered for grading, but there should be some time involved.

Task Screenshots:

Gallery Style: Large View

Small Medium Large



waka time ss

End of Assignment