Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-450-M2024/it114-module-5-project-milestone-1/grade/vk686

IT114-450-M2024 - [IT114] Module 5 Project Milestone 1

Submissions:

Submission Selection

1 Submission [active] 6/24/2024 9:56:51 AM

Instructions

COLLAPSE

Overview Video: https://youtu.be/A2yDMS9TS1o

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. Copy in the code from Sockets Part 5 into the Project folder (just the files)
    2. https://github.com/MattToegel/IT114/tree/M24-Sockets-Part5
4. Fix the package references at the top of each file (these are the only edits you should do at this point)
5. Git add/commit the baseline and push it to github
6. Create a pull request from Milestone1 to main (don't complete/merge it yet, just have it in open status)
7. Ensure the sample is working and fill in the below deliverables 1. Note: Don't forget the client commands are /name and /connect
8. Generate the output file once done and add it to your local repository
9. Git add/commit/push all changes
10. Complete the pull request merge from the step in the beginning
11. Locally checkout main
12. git pull origin main

Branch name: Milestone1

Tasks: 8 Points: 10.00

Start Up (3 pts.)

^COLLAPSE^

^COLLAPSE^

Task #1 - Points: 1

Text: Start Up

Details:

Important: Code screenshots should be fairly concise (try to show only the sections of code relevant to the question)

Capturing all possible code (i.e., including a lot of irrelevant code) can lead to a reduced grade. The goal is to show you understand what segments are related to the prompts.

#1) Show the Server starting via

Caption (required) ✓

Describe/highlight what's being shown

Showing the server starting via command line and listening for connections

#2) Show the Server Code that listens

Caption (required) ✓

Describe/highlight what's being shown (ucid/date must be present)

Showing the server Code that listens for connections

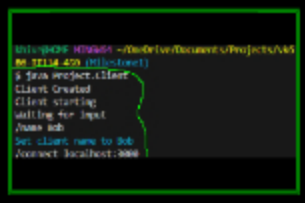
Explanation (required) ✓

Briefly explain the code related to starting up and waiting for connections

PREVIEW RESPONSE

"Start" method sets up the server socket and begins listening for client connections on our port(3000).

#3) Show the Client starting via



Caption (required) ✓

Describe/highlight what's being shown

Showing the client starting via command line

#4) Show the Client Code that

Caption (required) ✓

Describe/highlight what's being shown (ucid/date must be present)

Showing the client Code that prepares the client and waits for user input

Explanation (required) ✓

Briefly explain the code/logic/flow leading up to and including waiting for user input

PREVIEW RESPONSE

When the Client enum is created, the private constructor is called and

"try (ServerSocket serverSocket = new ServerSocket(port)) - is created to listen for incoming connections.

up. The result is "Client created" message.

The "start" method is the entry point for the client. It prints a message indicating the client is starting.

The "istenToInput" mthod runs in a loop, waiting for and processing user input, handling commands, and sending messages to the server if connected.

^COLLAPSE ^

Task #2 - Points: 1

Text: Connecting

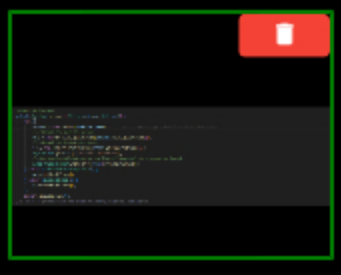
Details:

Important: Code screenshots should be fairly concise (try to show only the sections of code relevant to the question)

Capturing all possible code (i.e., including a lot of irrelevant code) can lead to a reduced grade. The goal is to show you understand what segments are related to the prompts.

#1) Show 3 Clients connecting to the Server

#2) Show the code related to Clients connecting to the Server (including the two needed commands)



Caption (required) ✓

Describe/highlight what's being shown (ucid/date

Caption (required) ✓
Describe/highlight what's being shown
Showing 3 Clients connecting to the Server

must be present)
Showing the code related to Clients connecting to the Server

Explanation (required) ✓
Briefly explain the code/logic/flow
PREVIEW RESPONSE
With the server's address and port, a "Socket" object is created. This object both requests to connect to the server and handles communication if the request is received.
Once the client and server can talk to each other, they both set up input and output streams to send and receive data.

The "isConnection method" is responsible for checking if a given string contains a valid connection command. It verifies if the input string matches the expected format for connecting to a server, which includes an IP address or "localhost" followed by a port number.

Communication (3 pts.)
COLLAPSE

COLLAPSE

Task #1 - Points: 1
Text: Communication

Details:
Important: Code screenshots should be fairly concise (try to show only the sections of code relevant to the question)

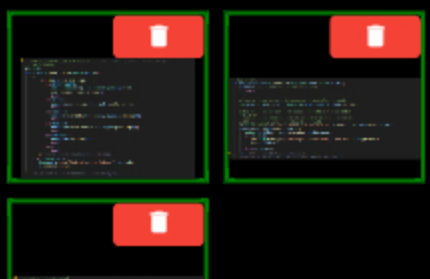
Capturing all possible code (i.e., including a lot of irrelevant code) can lead to a reduced grade. The goal is to show you understand what segments are related to the prompts.

#1) Show each Client sending and receiving messages

#2) Show the code related to the Client-side of getting a user

Caption (required) ✓

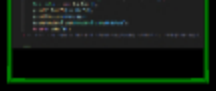
#3) Show the code related to the Server-side receiving the



Caption (required) ✓
Describe/highlight what's being shown
Showing each Client sending and receiving messages

Describe/highlight what's being shown (ucid/date must be present)
Showing the code related to the Client-side of getting a user message and sending it over the socket

Explanation (required) ✓
Briefly explain the code/logic/flow involved
PREVIEW RESPONSE
When a user types a message, the "sendMessage" method is called to create a payload and send it to the server.
Then it calls the "send" method to send the payload over the socket.



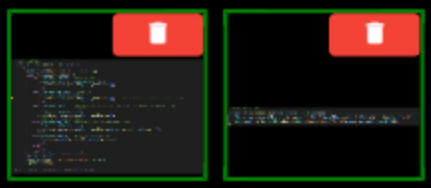
Caption (required) ✓
Describe/highlight what's being shown (ucid/date must be present)
Showing the code related to the Server-side receiving the message and relaying it to each connected Client

Explanation (required) ✓
Briefly explain the code/logic/flow involved
PREVIEW RESPONSE
The processPayload method in the ServerThread class processes different types of payloads received from the client. For message payloads, it calls the sendMessage method on the current room.

The sendMessage method in the Room class sends a message from one client to all clients in the room. It iterates over all clients in the room and calls the sendMessage method on each client's ServerThread instance.

The sendMessage method in ServerThread sends a message payload to the client.

#4) Show the code related to the Client receiving messages



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

shown (ucid/date must be present)
Showing the code related to the Client receiving messages from the Server-side and presenting them

Explanation (required) ✓
Briefly explain the code/logic/flow involved

PREVIEW RESPONSE

The processPayload method handles different types of payloads. Depending on the payload type, it calls specific methods to process.

The "processMessage" method handles regular messages from the server, printing them with the sender's name.

COLLAPSE

Task #2 - Points: 1

Text: Rooms

Details:
Important: Code screenshots should be fairly concise (try to show only the sections of code relevant to the question)

Capturing all possible code (i.e., including a lot of irrelevant code) can lead to a reduced grade. The goal is to show you understand what segments are related to the prompts.

#1) Show Clients can Create Rooms

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

#2) Show Clients can Join Rooms (leave/join messages

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

#3) Show the Client code related to the create/join room

Caption (required) ✓
Describe/highlight what's being shown (ucid/date must be present)

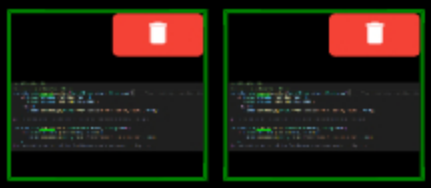
shown
Showing Clients can Create Rooms

shown
Showing Clients can Join Rooms (leave/join messages should be visible)

Showing the Client code related to the create/join room commands

Explanation (required) ✓
Briefly explain the code/logic/flow involved
PREVIEW RESPONSE
For /createroom, it extracts the room name and calls sendCreateRoom.
For /joinroom, it extracts the room name and calls sendJoinRoom.

#4) Show the ServerThread/Room code handling the



Caption (required) ✓
Describe/highlight what's being shown (ucid/date must be present)
Showing the ServerThread/Room code handling the create/join process

Explanation (required) ✓
Briefly explain the code/logic/flow involved
PREVIEW RESPONSE
For ROOM_CREATE payloads, it calls the handleCreateRoom method on the currentRoom instance.
For ROOM_JOIN payloads, it calls the handleJoinRoom method on the currentRoom instance.

The handleCreateRoom method attempts to create a new room via the Server class and moves the client to the new room.
The handleJoinRoom method

#5) Show the Server code for handling the create/join process

Caption (required) ✓
Describe/highlight what's being shown (ucid/date must be present)
Showing the Server code for handling the create/join process

Explanation (required) ✓
Briefly explain the code/logic/flow involved
PREVIEW RESPONSE
The "createRoom" attempts to create a new room and add it to the tracked rooms collection.
The "joinRoom" attempts to move a client to the specified room.

#6) Show that Client messages are constrained to the

Caption (required) ✓
Describe/highlight what's being shown
Showing that Client messages are constrained to the Room (clients in different Rooms can't talk to each other)

Explanation (required) ✓
Briefly explain why/how it works this way
PREVIEW RESPONSE
Each room maintains its own list of clients (clientsInRoom). When a message is sent, it is only broadcasted to clients in this list.

attempts to join the specified room via the Server class and moves the client to the specified room if it exists.

Disconnecting/Termination (3 pts.)

COLLAPSE

COLLAPSE

Task #1 - Points: 1

Text: Disconnecting

Details:

Important: Code screenshots should be fairly concise (try to show only the sections of code relevant to the question)

Capturing all possible code (i.e., including a lot of irrelevant code) can lead to a reduced grade. The goal is to show you understand what segments are related to the prompts.

#1) Show Clients gracefully disconnecting (should

Caption (required) ✓

Describe/highlight what's being shown

Showing Clients gracefully disconnecting (should not crash Server or other Clients)

#2) Show the code related to Clients disconnecting

Caption (required) ✓

Describe/highlight what's being shown (ucid/date must be present)

Showing the code related to Clients disconnecting

Explanation (required) ✓

Briefly explain the code/logic/flow involved

PREVIEW RESPONSE

The sendDisconnect method creates and sends the disconnect payload to the server.

#3) Show the Server terminating (Clients should be

Caption (required) ✓

Describe/highlight what's being shown

Showing the Server terminating (Clients

#4) Show the Server code related to handling termination

Caption (required) ✓

Describe/highlight what's being shown (ucid/date must be present)

Showing the Server code related to handling termination

Explanation (required) ✓

Briefly explain the code/logic/flow involved

PREVIEW RESPONSE

The "shutdown" method stops accepting new connections, closes the server socket, and disconnects all clients.

Misc (1 pt.)

COLLAPSE

COLLAPSE

Task #1 - Points: 1

Text: Add the pull request link for this branch

URL #1

https://github.com/VK686NJ/vk686-IT114-450/pull/9

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:

The material was well organized so I just had to pay attention and put time into studying previous material. I had no issues working on this assignment.

COLLAPSE

Task #3 - Points: 1

Text: WakaTime Screenshot

Details:

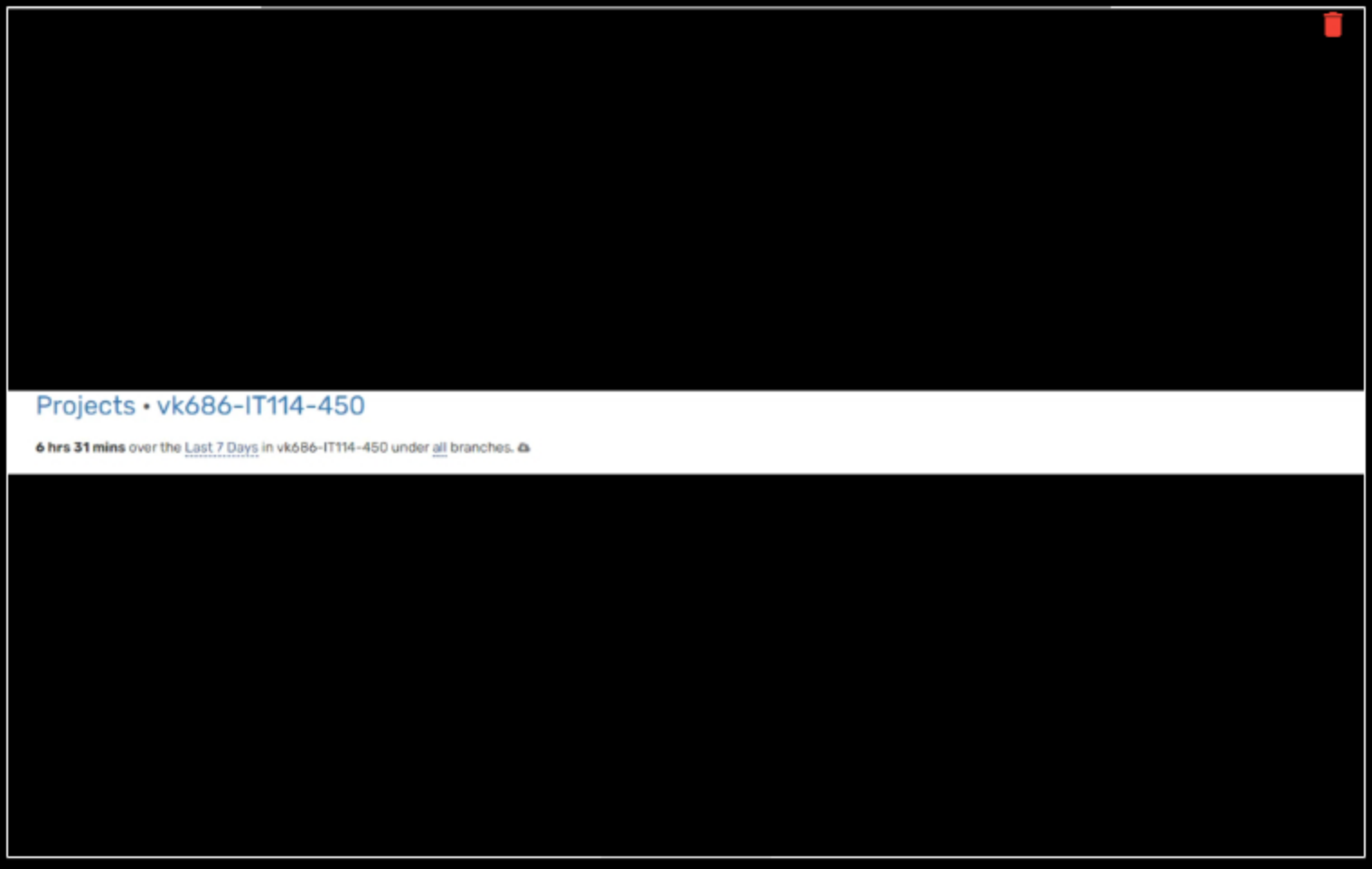
Grab a snippet showing the approximate time involved that clearly shows your repository.

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

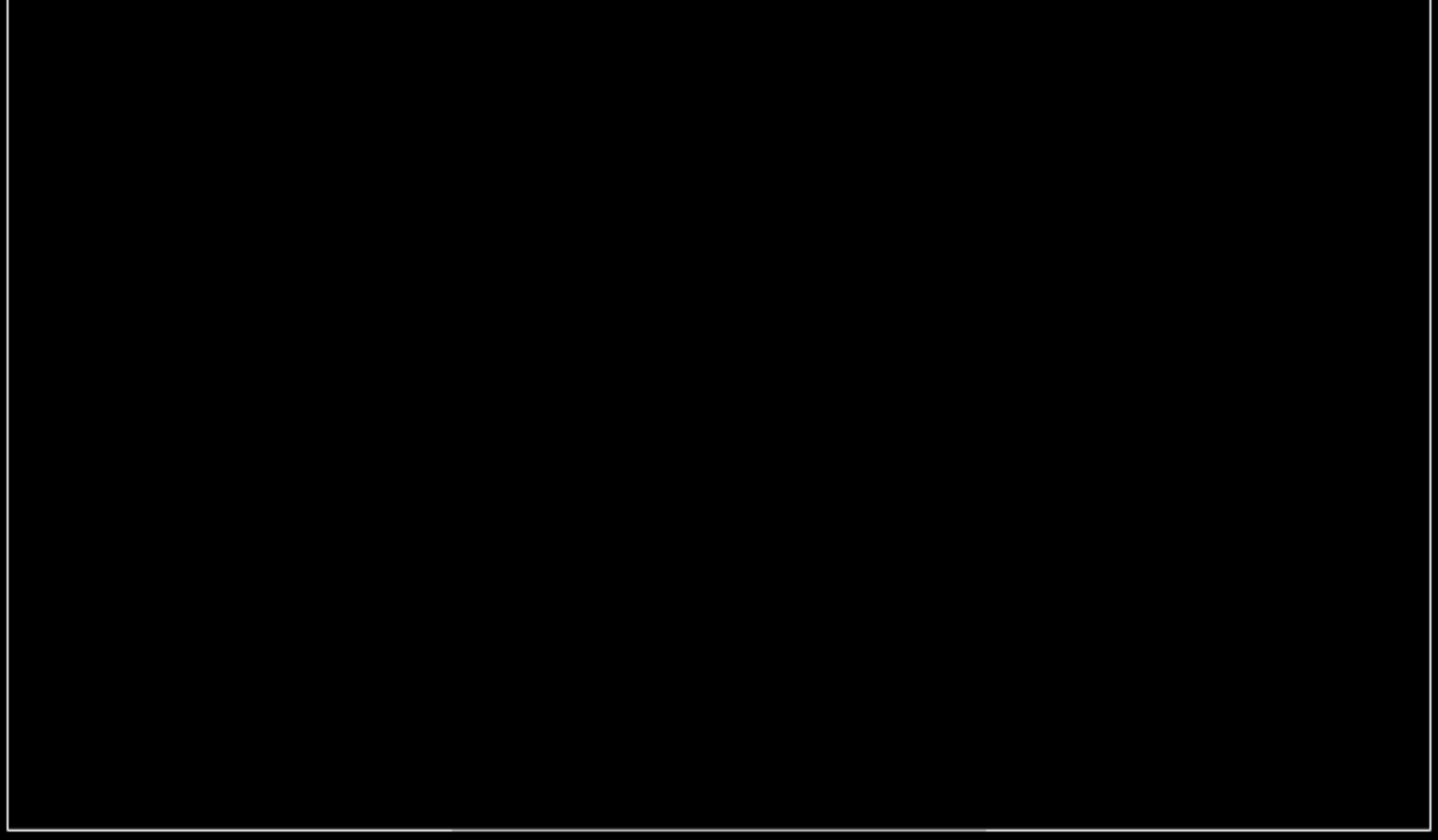
Task Screenshots:

Gallery Style: Large View

Small Medium Large



Overall



Detailed view

End of Assignment