Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-003-F2024/it114-module-5-project-milestone-1/grade/js2637

Course: IT114-003-F2024

Assigment: [IT114] Module 5 Project Milestone 1

Student: Jack S. (js2637)

Submissions:

Submission Selection

1 Submission [submitted] 10/14/2024 7:47:59 PM

•

Instructions

^ COLLAPSE ^

Overview Video: https://youtu.be/A2yDMS9TS10

- Create a new branch called Milestone1
- 2. At the root of your repository create a folder called Project if one doesn't exist yet
 - 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
- 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
- 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
- Create a pull request from Milestone1 to main (don't complete/merge it yet, just have it in open status)
- 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
- 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

Group



Group: Start Up

Tasks: 2 Points: 3

^ COLLAPSE ^

Task



Group: Start Up Task #1: Start Up

Weight: ~50% Points: ~1.50

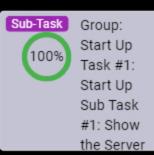
^ COLLAPSE ^

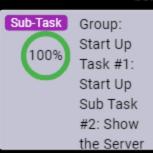
Details:

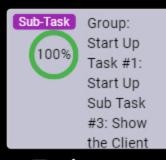
Important: Code screenshots should be fairly concise (try to show only the sections of code relevant to the question)

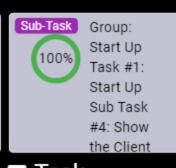
Capturing all possible code (i.e., including a lot of irrelevant code) can lead to a reduced grade.

Columns: 4









Screenshots

Gallery Style: 2 Columns

Screenshots

Gallery Style: 2 Columns

4 2

⊾ Task Screenshots

Gallery Style: 2 Columns

⊾ Task Screenshots

Gallery Style: 2 Columns

2



Server Starting

Caption(s) (required) <

Caption Hint: Describe/highlight what's being shown

connections

Code listen for

Caption(s) (required) <

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



Client Starting Via command line

being shown

Caption(s) (required) < Caption Hint:



wait for the User Input

Caption(s) (required) ~

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

≡,∕Task

≡ √Task

Response

Prompt

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

A ServerSocket object is created and waits for the clients to connect. Once connected it makes a Serverthread which handles the communication with the client. It also create the default room name LOBBY which all new user connect to at first.

Response

Prompt

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

Response:

This is The Listen to Input method which is called on by the Start method. The Start method which prepares the Client and the listenToInput method is called which waits for the user to input something and then check to see if it is a command.

End of Task 1

Task



Group: Start Up Task #2: Connecting

Weight: ~50% Points: ~1.50

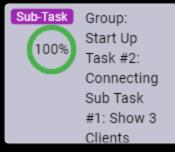


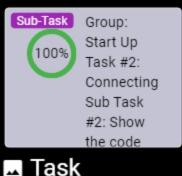


Important: Code screenshots should be fairly concise (try to show only the sections of code relevant to the question) ⇣

Capturing all possible code (i.e., including a lot of irrelevant code) can lead to a reduced grade.

Columns: 4





Screenshots

Screenshots

Gallery Style: 2 Columns Gallery Style: 2 Columns

4 2 1 4 2 1



3 Client Connect to Server

Caption(s) (required) <

Caption Hint:

Describe/highlight what's being shown



Code related to connecting to Server

Caption(s) (required) ~

Caption Hint:

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

≡, Task

Response

Prompt

Briefly explain the code/logic/flow Response:

The first image is the code that processes commands to connect to the Server while the second one is connecting to the Server. The processCommand section of the code helps the client create a name for the user and allows the user to /connect, the command to establish a connection with the server. The connect section of the code connects the client to the server using the port info provided by the command in the process commands section.

End of Task 2

End of Group: Start Up Task Status: 2/2

Group

100%

Group: Communication

Tasks: 2

^ COLLAPSE ^

Task



Group: Communication Task #1: Communication

Weight: ~50% Points: ~1.50

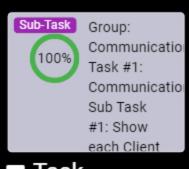
^ COLLAPSE ^

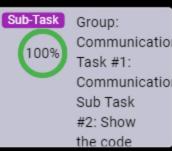


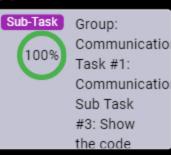
Important: Code screenshots should be fairly concise (try to show only the sections of code relevant to the question)

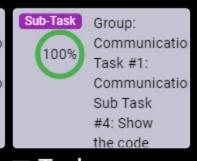
Capturing all possible code (i.e., including a lot of irrelevant code) can lead to a reduced grade.

Columns: 4









Screenshots

Gallery Style: 2 Columns

Task Screenshots

Gallery Style: 2 Columns

⊾ Task Screenshots

Gallery Style: 2 Columns

⊾ Task Screenshots

Gallery Style: 2 Columns

2 4



Sending messages between 3

client

Caption(s) (required) 🗸

Caption Hint: Describe/highlight what's being shown

2



Create a payload and send the payload to the socket

Caption(s) (required) <

Caption Hint: Describe/highlight what's being shown (ucid/date must

≡, Task

Daenanea

be present)

4 2



Server code message

Caption(s) (required) <

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

≡, Task

Response

Prompt Briefly explain the



code client receiving message

Caption(s) (required) <

Caption Hint:

being shown (ucid/date must be present)

≡, Task

Response **Prompt**

rcsponse **Prompt**

Briefly explain the code/logic/flow involved Response:

Creates a Payload then sends it over to the socket using the send(payload) method

code/logic/flow involved Response:

Receive payload from the client and send them back depending on the condition

Briefly explain the code/logic/flow involved Response:

It receives a payload from the server and then uses the process Payload method to get info.

End of Task 1

Task



Group: Communication

Task #2: Rooms Weight: ~50% Points: ~1.50

^ COLLAPSE ^

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.

Columns: 4

Sub-Task Group: Communication 100% Task #2: Rooms Sub Task #1: Show Clients can

Sub-Task Group: Communicatio 100% Task #2: Rooms Sub Task #2: Show Clients can

Sub-Task Group: Communicatio 100% Task #2: Rooms Sub Task #3: Show the Client

Sub-Task Group: Communicatio 100% Task #2: Rooms Sub Task #4: Show the

Screenshots

Gallery Style: 2 Columns

Task Screenshots

Gallery Style: 2 Columns

Screenshots

Gallery Style: 2 Columns

⊾ Task Screenshots

> 4 2

Gallery Style: 2 Columns

2



Client 1 created a room

4 2



Client 2 left lobby room and join test room with

2



create room

Caption(s) (required) <



Code

Caption(s) (required) < Caption Hint:

Caption Hint:

Caption(s) (required) <

Caption Hint:

Describe/highlight what's being shown

client 1

Caption(s) (required) <

Caption Hint:

Describe/highlight what's being shown

Describe/highlight what's

be present)

≡, Task

Response

Prompt

Briefly explain the code/logic/flow involved Response:

It is in the process command section of the code that will create a payload and which will allow you to set the room name or allow you to join.

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

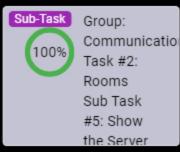
≡, Task

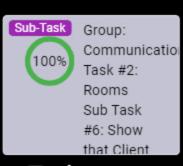
Response

Prompt

Briefly explain the code/logic/flow involved Response:

The code will create and room and make sure that none already exist and if someone try to join a room that doesn't exist they will be given a message that it doesn't exist.





Screenshots

Screenshots

Gallery Style: 2 Columns

4 2 1

Gallery Style: 2 Columns



create/join

room

Caption(s) (required) 🗸

Caption Hint: Describe/highlight what's being shown (ucid/date must other

be present)

≡, Task

Response Prompt

4 2 1

Client

messages 1

and 2 are in

the same room, so they

can see each

messages. client 3 is in the lobby and

can't see it.

Caption(s) (required) <

code/logic/flow involved
Response:

make sure room doesn't exist before creating one, and second part add a client too new room Caption Hint: Describe/highlight what's being shown

≡, Task

Response

Prompt

Briefly explain why/how it works this way Response:

It uses the currentRoom.sendmessage so it only sends the message to the current room.

End of Task 2

End of Group: Communication

Task Status: 2/2

Group



Group: Disconnecting/Termination

Tasks: 1 Points: 3

^ COLLAPSE ^

Task



Group: Disconnecting/Termination

Task #1: Disconnecting

Weight: ~100% Points: ~3.00

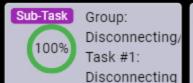
^ COLLAPSE ^

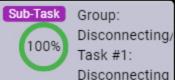
Details:

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

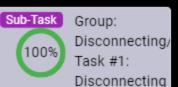
Capturing all possible code (i.e., including a lot of irrelevant code) can lead to a reduced grade.

Columns: 4









Sub Task #1: Show Clients

Sub Task #2: Show the code

Sub Task #3: Show the Server Sub Task #4: Show the Server

Gallery Style: 2 Columns

Screenshots

Task Screenshots

Gallery Style: 2 Columns

Screenshots

Gallery Style: 2 Columns

2

I Task ■ Task Screenshots

Gallery Style: 2 Columns

2



Client 1 disconnected and informed the other user in its room

Caption(s) (required) <

Caption Hint: Describe/highlight what's being shown

4 2



Server disconnecting

Caption(s) (required) <

Caption Hint:

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

≡ √Task

Response

Prompt

Briefly explain the code/logic/flow involved Response:

Sends the Server info to the server that the client is going to disconnect and then handles the client disconnecting by resetting the data and making sure the are no resource leaks.



Server Terminating

Caption(s) (required) <

Caption Hint:

Describe/highlight what's

2



shut down

Caption(s) (required) ~

Caption Hint:

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

≡ ∕Task

Response

Prompt

Briefly explain the code/logic/flow involved Response:

shutdown() remove all Users from Room from the server when the server is terminated.

End of Task 1

End of Group: Disconnecting/Termination

Task Status: 1/1

Group



Group: Misc Tasks: 3 Points: 1



Task



Group: Misc

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

Weight: ~33% Points: ~0.33

^ COLLAPSE ^

⇔Task URLs

URL #1

https://github.com/Jackshii/Js2637-IT114-003/pull/10

UR

https://github.com/Jackshii/Js2637-IT114-003/p

End of Task 1

Task



Group: Misc

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

Weight: ~33% Points: ~0.33

^ COLLAPSE ^



Few related sentences about the Project/sockets topics



∓ √ Task Response Prompt

Response:

I learn about where certain command are located in which part of the project and how to access them in the future. it was hard finding some of the part because i was look in the wrong places, i need to study more about server on my free time.

End of Task 2

Task



Group: Misc

Task #3: WakaTime Screenshot

Weight: ~33% Points: ~0.33

^ COLLAPSE ^



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

4 2 1



WakaTime

End of Task 3

End of Group: Misc Task Status: 3/3

End of Assignment