Submission Worksheet

CLICK TO GRADE

https://learn.ethereallab.app/assignment/IT114-004-S2024/it114-project-milestone-1/grade/rn364

IT114-004-S2024 - [IT114] Project Milestone 1

Submissions:

Submission Selection

1 Submission [active] 3/19/2024 11:34:20 AM

Instructions

△ COLLAPSE △

Create a new branch called Milestone1

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

You won't be creating separate folders for milestones; milestones are just branches

Create a pull request from Milestone1 to main (don't complete/merge it yet, just have it in open status)

Copy in the latest Socket sample code from the most recent Socket Part example of the lessons Recommended Part 5 (clients should be having names at this point and not ids)

https://github.com/MattToegel/IT114/tree/Module5/Module5

Fix the package references at the top of each file (these are the only edits you should do at this point)

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

Note: The client commands likely are different in part 5 with the /name and /connect options instead of just "connect"

Generate the worksheet output file once done and add it to your local repository

Git add/commit/push all changes

Complete the pull request merge from step 7

Locally checkout main

git pull origin main

Branch name: Milestone1

Tasks: 9 Points: 10.00





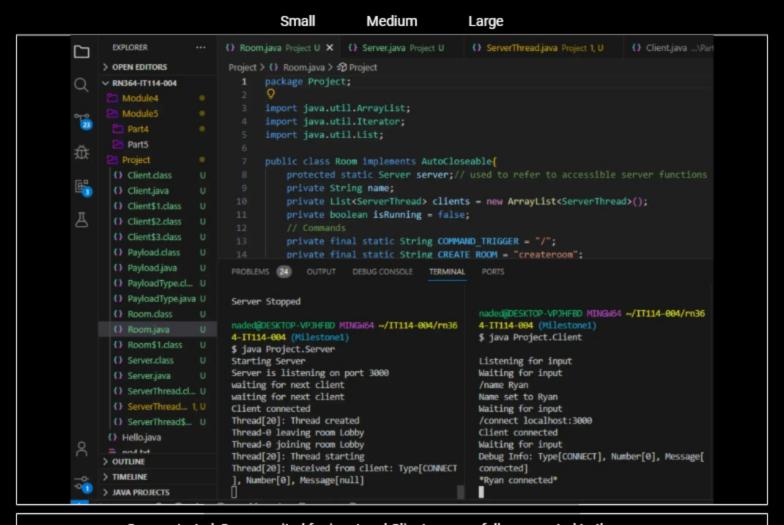
Task #1 - Points: 1

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



Server started, Server waited for input and Client successfully connected to the server

Checklist Items (3)

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



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:

The server gets set up to receive input from the client server and waits on port 3000 after compiling. Before the client can connect, it needs to pick a name. so on the client side i typed /name Ryan to create a name for the user. After choosing a name you will be able to connect to the server. I was able to connect by typing /connect localhost:3000 and Once connected, both the client side and server side are linked up.





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

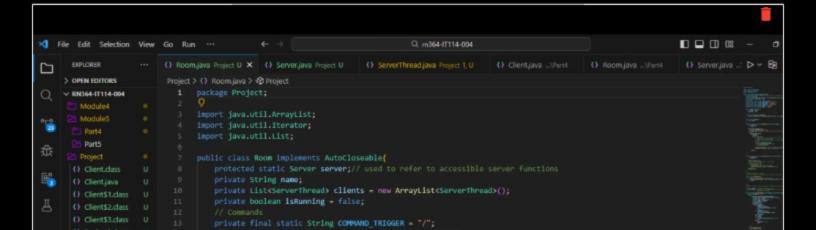
Small Medium Large ★ File Edit Selection View Go Run … () Server.java ...' ▷ ∨ 🙉 🗓 EXPLORER () Room,java Project U X () Server,java Project U Project > () Room.java > 30 Project package Project; V RN364-IT114-004 ° 128 Part5 盘 protected static Server server;// used to refer to accessible server functions () Client.class private String name; private List<ServerThread> clients = new ArrayList<ServerThread>(); () Client\$1.class private boolean isRunning = false; () Client\$3.class private final static String COMMAND_TRIGGER = "/";
private final static String CREATE ROOM = "createrpoom"; PROBLEMS (24) OUTPUT DEBUG CONSOLE TERMINAL 日神・十、田自…へ r ⊡ java Thread[20]: Received from client: Type[CONNECT connected] Name set to Tim } □ jav], Number[0], Message[null] *Ryan connected* Waiting for input L 🖸 java /connect localhost:3000 Client connected waiting for next client Client connected Debug Info: Type[DISCONNECT], Number[0], Messa () Room.java U ge[disconnected]
null disconnected Thread[22]: Thread created Thread-2 leaving room Lobby
Thread-2 joining room Lobby
Thread[22]: Thread starting
Thread[22]: Received from client: Type[CONNECT () Server.class Debug Info: Type[CONNECT], Number[0], Message[Debug Info: Type[CONNECT], Number[θ], Messag connected] e[connected]
Tim connected () ServerThread.d., U Debug Info: Type[MESSAGE], Number[0], Message[() ServerThread... 1.U], Number[0], Message[null] Thread[22]: Received from client: Type[MESSAGE Waiting for input Debug Info: Type[MESSAGE], Number[0], Messag], Number[0], Message[yy] маѕд () Hellojava Room[Lobby]: Sending message to 2 clients Thread[20]: Received from client: Type[MESSAGE Tim: yy
Debug Info: Type[MESSAGE], Number[0], Messag Waiting for input Debug Info: Type[MESSAGE], Number[0], Message[> OUTLINE], Number[0], Message[wasg] Room[Lobby]: Sending message to 2 clients шаѕд] e[wasg] > TIMELINE Ryan: wasg Ryan: wasg > JAVA PROJECTS ™ Milestone1* 🌣 🔞 0 🛕 13 ① 11 💖 0 🔗 🕦 23 mins 🖰 Java: Ready

2 clients connected and are able to send messages to each other

Ln 1, Col 17 Tab Size: 4 UTF-8 CRLF () Java 4

Checklist Items (4)

- #1 At least two clients connected to the server
- #2 Client can send messages to the server
- #3 Server sends the message to all clients in the same room
- #4 Messages clearly show who the message is from (i.e., client name is clearly with the message)



```
( ) Payload.class
                                                                                                      private final static String CREATE ROOM = "createroom";
             () Payload java
                                                                               PROBLEMS (24) OUTPUT DEBUG CONSOLE TERMINAL PORTS
                                                                                                                                                                                                                                                                                                                                                                                                                                                   □ java + ∨ □ 自 …
                                                                              ], Number[0], Message[null]
Thread[22]: Received from client: Type[MESSAGE
            () PayloadType.java U
       | Thread[22]: Received from client: Type[MESSAGE] |
| Number[0], Message[yy] |
| Room[1.class | U | |
| Server.class | U |
| Server.java | U |
| ServerThreadch... | U |
| ServerThreadch... | U |
| ServerThreads... | U |
| Constitution | Constitution | Constitution |
| Constitution | Constitution | Constitution |
| Constitution | Constitution | Constitution |
| Number[0], Message[wasg] |
| Room[Lobby]: Sending message to 2 clients |
| Number[0], Message[/createroom njit] |
| Room[Lobby]: Sending message to 2 clients |
| Created new room: njit |
| Created new
                                                                                                                                                                                                                         Maiting for input
                                                                                                                                                                                                                                                                                                                                                                   Debug Info: Type[CONNECT], Number[0], Messag
                                                                                                                                                                                                                        Mailing for Impact
Debug Info: Type[MESSAGE], Number[0], Message[ e[connected] 
*Tim connected*
                                                                                                                                                                                                                       Masg]
                                                                                                                                                                                                                                                                                                                                                                   Waiting for input
Debug Info: Type[MESSAGE], Number[0], Messag
                                                                                                                                                                                                                          /createroom njit
                                                                                                                                                                                                                        Maiting for input
                                                                                                                                                                                                                         Debug Info: Type[CONNECT], Number[0], Message[
                                                                                                                                                                                                                                                                                                                                                                    e[yy]
                                                                                                                                                                                                                            *Ryan connected*
                                                                                                                                                                                                                                                                                                                                                                   Debug Info: Type[MESSAGE], Number[0], Messag
                                                                              Thread-0 leaving room Lobby
                                                                                                                                                                                                                                                                                                                                                                   e[wasg]
          () Hello,java
                                                                                                                                                                                                                       Thread-0 joining room njit
Thread[20]: Received from client: Type[MESSAGE
       > OUTLINE
                                                                             ], Number[0], Message[sds]
                                                                               Room[njit]: Sending message to 1 clients
       > TIMELINE
                                                                                                                                                                                                                         Ryan: sds
                                                                                                                                                                                                                                                                                                                                                                      *Ryan disconnected*
      > JAVA PROJECTS
*$ Milestone1* 中 ⊗ 0 点 13 ⊙ 11 № 0 ♪ ⊙ 23 mins ⑤ Java: Ready
                                                                                                                                                                                                                                                                                                                                                                                                           Ln 1. Col 17 Tab Size: 4 UTF-8 CRLF () Je
```

I made a new room called NJIT and joined it in the image, when I type a message it doesnt show for the other client.

Checklist Items (2)

#5 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 Clearly caption each image regarding what is being shown



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's side, their input is taken and passed through the process command function. Afterwards, the relevant data is sent out. In the server thread, the process message function handles three scenarios: connecting, disconnecting, and messaging. It receives input from the client and navigates through these different cases. Within the room class, the process command function manages input from both the client and the server thread. It employs a case structure to handle actions like creating, joining, and disconnecting from rooms. The resulting message is then sent back to the client. If clients are in the same room, they receive the same message, but not if they're in separate rooms.



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

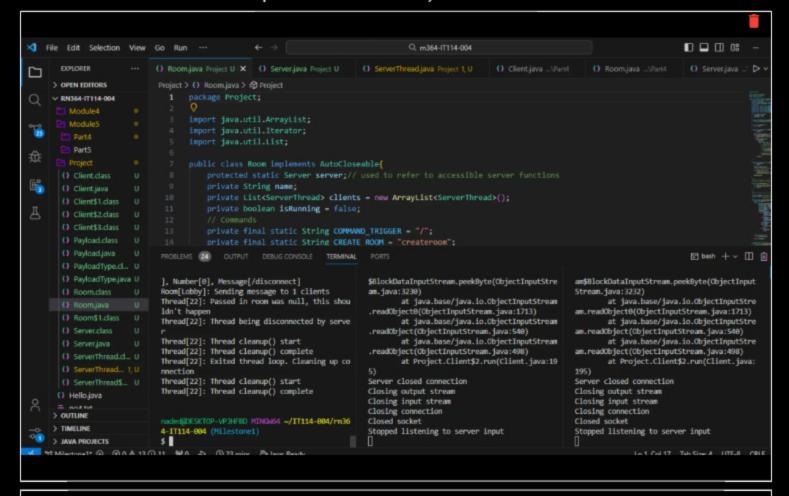
Task Screenshots:

Gallery Style: Large View

Small Medium Large x File Edit Selection View Go Run ··· Q m364-IT114-004 EXPLORER () ServerThread.java Project 1.U () Room.java ...\Part4 Project > () Room.java > 2 Project > OPEN EDITORS package Project; import java util List: 盘 protected static Server server;// used to refer to accessible server functions () Client.java private List<ServerThread> clients = new ArrayList<ServerThread>(); private boolean isRunning - false; PROBLEMS (24) OUTPUT DEBUG CONSOLE TERMINAL PORTS □ isvs 十~ □ 亩 Thread[20]: Received from client: Type[MESSAGE Tim: yy am\$BlockDataInputStream.peekByte(ObjectInput], Number[0], Message[sds] Room[njit]: Sending message to 1 clients Thread[22]: Received from client: Type[MESSAGE Stream.java:3232) Waiting for input at java.base/java.io.ObjectInputStre Debug Info: Type[MESSAGE], Number[0], Message[am.readObject0(ObjectInputStream.java:1713)], Number[0], Message[/disconnect] Room[Lobby]: Sending message to 1 clients at java.base/java.io.ObjectInputStre wasg] () Server.class U Ryan: wasg am.readObject(ObjectInputStream.java:540) () Server java Thread[22]: Passed in room was null, this show /createroom njit at java.base/java.io.ObjectInputStre Waiting for input am.readObject(ObjectInputStream.java:498) at Project.Client\$2.run(Client.java: Thread[22]: Thread being disconnected by serve Debug Info: Type[CONNECT], Number[0], Message[connected] Thread[22]: Thread cleanup() start *Ryan connected* Server closed connection Thread[22]: Thread cleanup() complete Thread[22]: Exited thread loop. Cleaning up co Closing output stream () Hello,java Waiting for input Closing input stream Debug Info: Type[MESSAGE], Number[0], Message[nnection Closing connection > OUTLINE Thread[22]: Thread cleanup() start Closed socket > TIMELINE Thread[22]: Thread cleanup() complete Stopped listening to server input > JAVA PROJECTS *\$ Milestone1* ♀ ⊗ 0 ≜ 13 ⊙ 11 ₩ 0 ♪ ⊙ 23 mins ♂ Java: Ready Ln 1, Col 17 Tab Size: 4 UTF-8 CRLF

Checklist Items (2)

- #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)
- #3 For each scenario, disconnected messages should be shown to the clients (should show a different person disconnected and should show the specific client disconnected)



Server is terminated

Checklist Items (1)

#2 Show the server terminating; Clients should be disconnected but still running and able to reconnect when the server is back online (demonstrate this)



Task #2 - Points: 1

Text: Explain the various Disconnect/termination scenarios

① 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
		Montion how (why the client program decen't areah when the conver

#2	1	disconnects/terminates.
#3	1	Mention how the server doesn't crash from the client(s) disconnecting

Response:

A client gets disconnected when the running status is false because there's no server thread running. To handle this, the client program uses a try-catch block to prevent crashing if the server isn't connected. Instead, it keeps running and informs the client that the server is disconnected. The server does something similar to check if a client is connected, also preventing the code from stopping if there's an issue.





Task #1 - Points: 1

Text: Add the pull request link for this branch

URL #1

https://github.com/ryann2n/rn364-IT114-004/pull/8



Task #2 - Points: 1

Text: Talk about any issues or learnings during this assignment



Few related sentences about the Project/sockets topics

Response:

It was confusing at first because I was used to just running the files with the play button on vscode, when I tried it in the terminal I was getting errors but thats just because I forgot to compile first.



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:

dalicity Style. Large view



End of Assignment