

Submission Worksheet

CLICK TO GRADE

<https://learn.ethereallab.app/assignment/IT114-002-S2024/it114-chatroom-milestone-3-2024/grade/am3485>

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

Submissions:

Submission Selection

1 Submission [active] 4/15/2024 9:41:04 AM

Instructions

^ COLLAPSE ^

Implement the Milestone 3 features from the project's proposal document: <https://docs.google.com/document/d/1ONmvEvel97GTFPGfVwwQC96xSsobbSbk56145Xi>
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 checkboxes are for your own tracking
#	Points	Details	

<input type="checkbox"/> #1	1	Connection Panel
<input type="checkbox"/> #2	1	User Details Panel
<input type="checkbox"/> #3	1	Chat Panel
<input type="checkbox"/> #4	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

SmallMediumLarge

Client

Rooms

Host:

127.0.0.1

Port:

3000

Next

Connection Panel

Checklist Items (0)

Client

Rooms

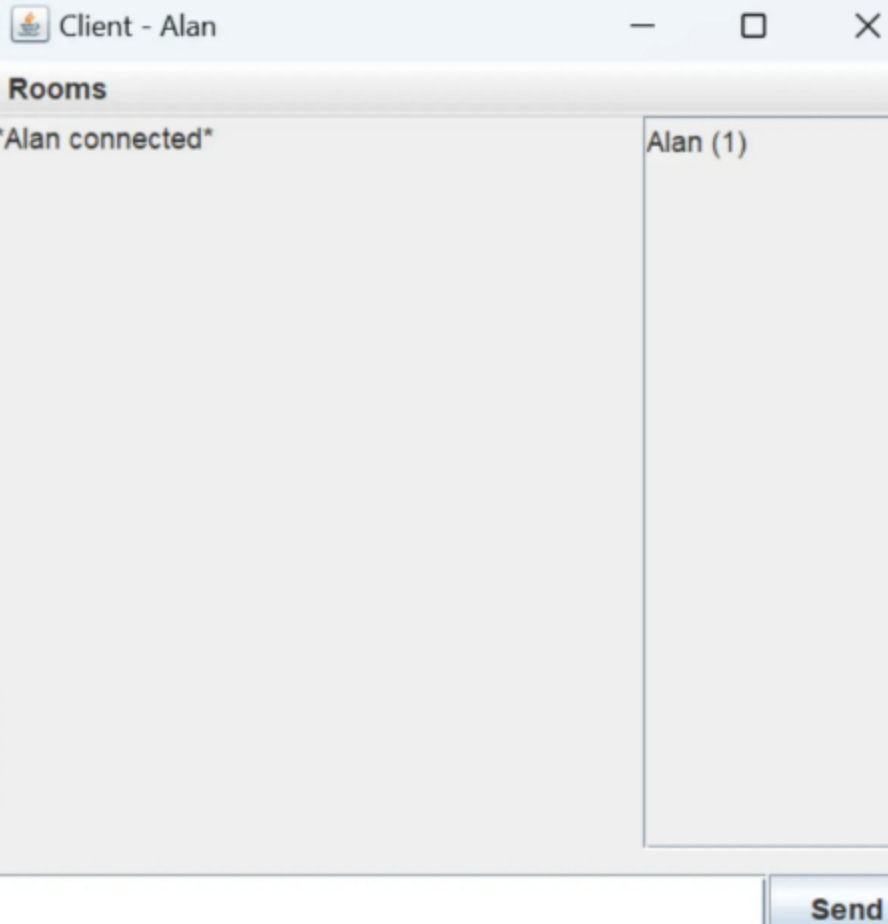
Username:

Previous

Connect

User Details Panel

Checklist Items (0)



Chat Panel

Checklist Items (0)



Formatting (2 pts.)

^COLLAPSE ^



Task #1 - Points: 1

Text: Screenshots demoing flip and roll commands

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Flip output in a different format than normal messages
<input type="checkbox"/> #2	1	Roll output in a different format than normal messages

<input type="checkbox"/> #2	1	Roll # output in a different format than normal messages
<input type="checkbox"/> #3	1	Roll #d# output in a different format than normal messages
<input type="checkbox"/> #4	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small Medium Large



Yellow is the flip output Blue is the #d# roll output Red is the # roll output

Checklist Items (0)

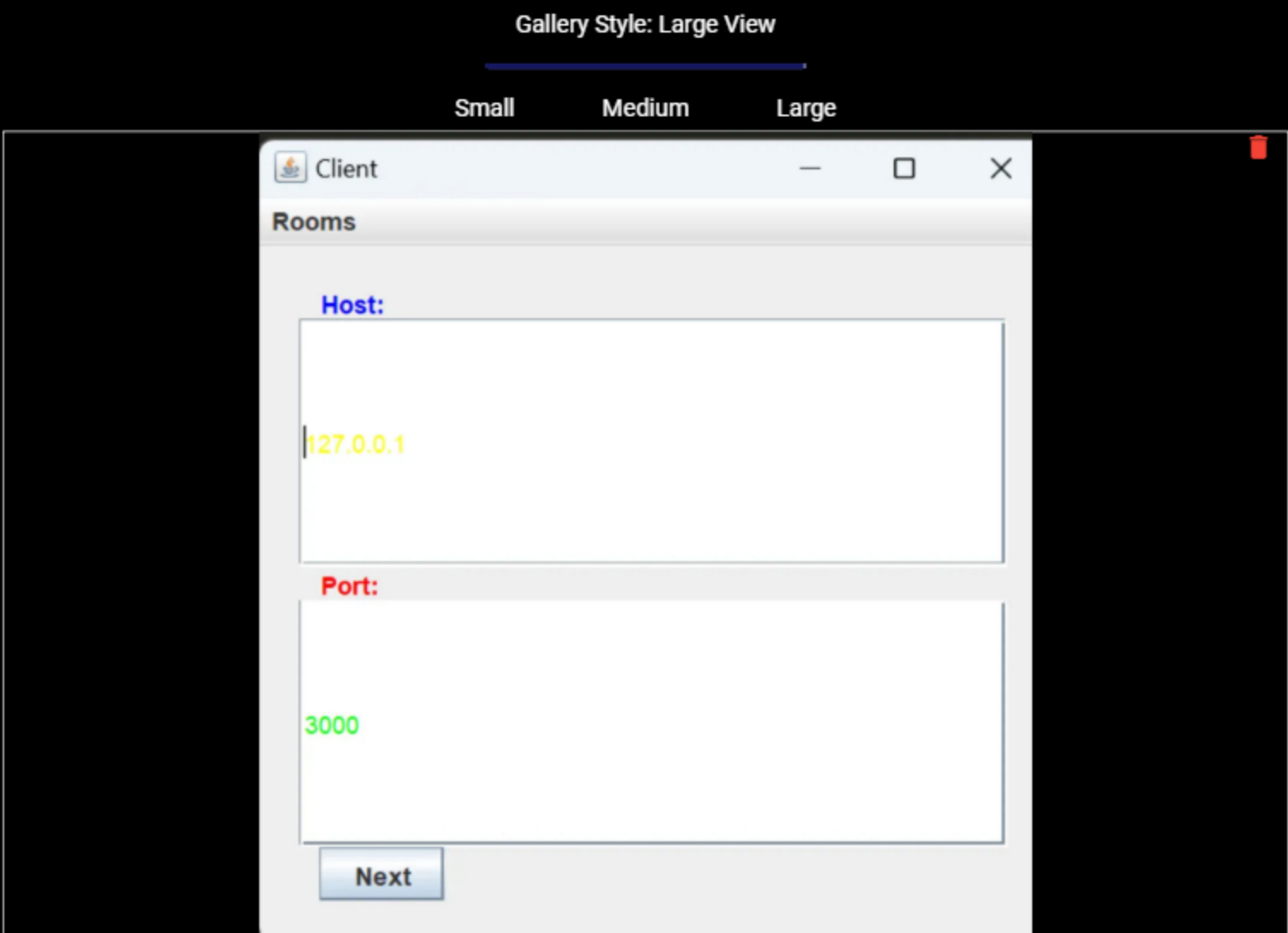
^COLLAPSE ^

Task #2 - Points: 1
Text: Screenshots demoing custom text formatting

Checklist			*The checkboxes are for your own tracking
#	Points	Details	
<input type="checkbox"/> #1	1	Custom text formatting for bold working (Part of the message should appear bold)	
<input type="checkbox"/> #2	1	Custom text formatting for italic working (Part of the message should appear italic)	
<input type="checkbox"/> #3	1	Custom text formatting for underline working (Part of the message should appear underline)	
<input checked="" type="checkbox"/> #4	1	Custom text formatting for red working (Part of the message should appear red)	
<input checked="" type="checkbox"/> #5	1	Custom text formatting for blue working (Part of the message should appear blue)	

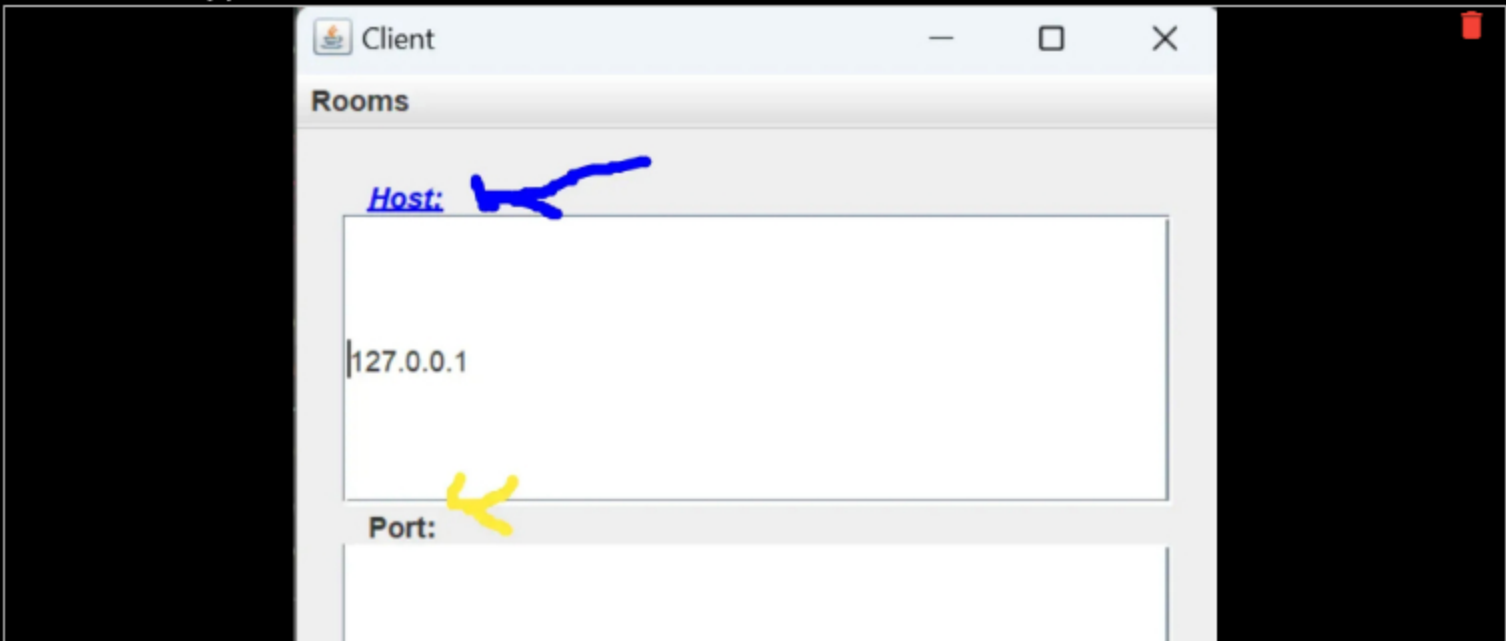
<input checked="" type="checkbox"/> #6	1	Custom text formatting for green working (Part of the message should appear green)
<input type="checkbox"/> #7	1	Custom text formatting for combined bold, italic, underline, and a color working (Part of the message should have all 4 formats applied at once)
<input checked="" type="checkbox"/> #8	1	Clearly caption screenshots

Task Screenshots:



Example of text using different colors

Checklist Items (0)



3000

Next

blue is example for italic bold underline and color all at once yellow is example for bold

Checklist Items (0)

Client

Rooms

Host:

127.0.0.1

Port:

3000

Next

red is underlined green is italics

Checklist Items (0)



^COLLAPSE ^

Task #3 - Points: 1

Text: Screenshot of the code solving the formatting display

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Show each relevant file this was done in (may be one or more)
<input type="checkbox"/> #2	1	Include ucid and date comment
<input type="checkbox"/> #3	1	Clearly caption screenshots

Task Screenshots:

Small

Medium

Large

```
case FLIP:
    try{
        FlipPayload fp = (FlipPayload) p;
        fp.toString();//am3485 4/1/2024
        send(fp);
        currentRoom.sendMessage(this, fp.lolString());//am3485 4/15/2024 You, 43 min
    }catch (Exception e){
        this.sendMessage(Constants.DEFAULT_CLIENT_ID,
            "You can only use the /turn command in a GameRoom and not the Lobby");
    }
    break;
case ROLL:
    try{
        RollPayload rp =(RollPayload) p;
        rp.toString();
        send(rp);//am3485 4/1/2024
        currentRoom.sendMessage(this, rp.lolString());//am3485 4/15/2024
    }catch (Exception e){
        this.sendMessage(Constants.DEFAULT_CLIENT_ID,
            "You can only use the /turn command in a GameRoom and not the Lobby");
    }
```

code added in server thread to make roll appear in chat documented in blue

Checklist Items (0)

```
//am3485 4/15/2024
public String lolString() {
    return String.format("Type[%s], Roll Prompt[%s], Result[%s] ", getPayloadType().toString(),
        getRollPrompt(), getResults());
}
```

You, 7 minutes ago • Uncommitted changes

code added in roll payload to make the string

Checklist Items (0)

```
//am3485 4/15/2024      You, 48 minutes ago • Uncommitted changes
public String lolString() {
    return String.format("Type[%s], Result[%s] ", getPayloadType().toString(),
        getResult());
}
```

code added in flip payload to make the string

Checklist Items (0)

```
public ChatPanel(ICardControls controls) {
    super(new BorderLayout(10, 10));
    JPanel wrapper = new JPanel();
    wrapper.setLayout(new BoxLayout(wrapper, BoxLayout.Y_AXIS));
    JPanel content = new JPanel();
    content.setLayout(new BoxLayout(content, BoxLayout.Y_AXIS));
    content.setAlignmentY(Component.BOTTOM_ALIGNMENT);

    // wraps a viewport to provide scroll capabilities
    JScrollPane scroll = new JScrollPane(content);
    scroll.setHorizontalScrollBarPolicy(ScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEVER);
    scroll.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED);
    scroll.setBorder(BorderFactory.createEmptyBorder());
    // no need to add content specifically because scroll wraps it
    wrapper.add(scroll);
    this.add(wrapper, BorderLayout.CENTER);

    JPanel input = new JPanel();
    input.setLayout(new BoxLayout(input, BoxLayout.X_AXIS));
    JTextField textValue = new JTextField();
    input.add(textValue);
    JButton button = new JButton("Send");
    JTextField diceValue = new JTextField();//am3485 4/15/2024
    //input.add(diceValue);
    JButton rbutton = new JButton("Roll");//am3485 4/15/2024
    JButton fbutton = new JButton("Flip");//am3485 4/15/2024
    // lets us submit with the enter key instead of just the button click
```


You, 2 days ago | 1 author (You)

```
textView.addKeyListener(new KeyListener() {
```

code added into chat panel for roll and flip function documented in blue

Checklist Items (0)

```
input.add(button); //am3485 4/15/2024
input.add(diceValue); //am3485 4/15/2024
fbutton.addActionListener((event) -> {
    try {
        if (true) {
            Client.INSTANCE.sendFlip();
            //am3485 4/15/2024
            // debugging
            logger.log(Level.FINEST, "Content: " + content.getSize());
            logger.log(Level.FINEST, "Parent: " + this.getSize());
        } <- #105-112 if (true)
    } catch (NullPointerException e) {
    } catch (IOException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
}); <- #103-118 fbutton.addActionListener
rbutton.addActionListener((event) -> {
    try {
        String rollText = diceValue.getText().trim();
        if (rollText.length() > 0) {
            Client.INSTANCE.sendRoll(rollText);
            diceValue.setText(""); // clear the original text
            //am3485 4/15/2024
            // debugging
            logger.log(Level.FINEST, "Content: " + content.getSize());
            logger.log(Level.FINEST, "Parent: " + this.getSize());
        } <- #122-130 if (rollText.length() > 0)
    } catch (NullPointerException e) {
    } catch (IOException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
}); <- #119-136 rbutton.addActionListener
```

code added into chat panel for roll and flip function documented in blue

Checklist Items (0)

```
//am3485 4/15/2024
```

```
input.add(rbutton);
```

```
input.add(fbutton);
```

```
userListPanel = new User
```


```
this.add(userListPanel,
```

code added into chat panel for roll and flip function documented in blue

Checklist Items (0)

```
public ConnectionPanel(ICardControls controls) {
    super(new BorderLayout(10, 10));


    JPanel content = new JPanel();
    content.setLayout(new BoxLayout(content, BoxLayout.Y_AXIS));
    // add host info
    JLabel hostLabel = new JLabel("Host:");
    hostLabel.setForeground(Color.blue); //am3485 4/15/24
    JTextField hostValue = new JTextField("127.0.0.1");
    JLabel hostError = new JLabel();
    hostValue.setForeground(Color.yellow); //am3485 4/15/24
    content.add(hostLabel);
    content.add(hostValue);
    content.add(hostError);
    // add port info
    JLabel portLabel = new JLabel("Port:");
    portLabel.setForeground(Color.red); //am3485 4/15/24
    JTextField portValue = new JTextField("3000");
    portValue.setForeground(Color.green); //am3485 4/15/24
    JLabel portError = new JLabel();
    You, 1 mi
```



Code for handling color changes in connection panel

Checklist Items (0)

```
JPanel content = new JPanel();
content.setLayout(new BoxLayout(content, BoxLayout.Y_AXIS));
// add host info
JLabel hostLabel = new JLabel("<HTML><U><I><B>Host:</B></I></U></HTML>"); //am3485 4/15/24
hostLabel.setForeground(Color.blue); // am3485 4/15/24
JTextField hostValue = new JTextField("127.0.0.1");
//JTextField hostValue = new JTextField("<HTML><U>127.0.0.1</U></HTML>");
JLabel hostError = new JLabel();
//hostValue.setForeground(Color.yellow); // am3485 4/15/24
content.add(hostLabel);
content.add(hostValue);
content.add(hostError);
// add port info
JLabel portLabel = new JLabel("<HTML><B>Port:</B></HTML>"); //am3485 4/15/24
//portLabel.setForeground(Color.red); // am3485 4/15/24
JTextField portValue = new JTextField("3000");
//JTextField portValue = new JTextField("<HTML><I>3000</I></HTML>");
//portValue.setForeground(Color.green); // am3485 4/15/24
```



green is code for bold yellow is code for italic bold underline and color at once

Checklist Items (0)

```
JPanel content = new JPanel();
content.setLayout(new BoxLayout(content, BoxLayout.Y_AXIS));
// add host info
JLabel hostLabel = new JLabel("<HTML><U>Host:</U></HTML>");//am3485 4/15/24
//hostLabel.setForeground(Color.blue); // am3485 4/15/24 You, 1 second ago
JTextField hostValue = new JTextField("127.0.0.1");
//JTextField hostValue = new JTextField("<HTML><U>127.0.0.1</U></HTML>");
JLabel hostError = new JLabel();
//hostValue.setForeground(Color.yellow); // am3485 4/15/24
content.add(hostLabel);
content.add(hostValue);
content.add(hostError);
// add port info
JLabel portLabel = new JLabel("<HTML><I>Port:</I></HTML>");//am3485 4/15/24
//portLabel.setForeground(Color.red);// am3485 4/15/24
JTextField portValue = new JTextField("3000");
//JTextField portValue = new JTextField("<HTML><I>3000</I></HTML>");
//portValue.setForeground(Color.green);// am3485 4/15/24
```

red is code for underlined green is code for italics

Checklist Items (0)

●

^COLLAPSE ^

Task #4 - Points: 1

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

Details:
Note each scenario

Response:

I used html code to change who the letters looked in the ui and then i used a method called setforeground to change the color of the text in the label

^COLLAPSE ^



^COLLAPSE ^

Task #1 - Points: 1

Text: Screenshots demoing private message

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Should have 3 clients in the same room
<input type="checkbox"/> #2	1	Demo a private message where only the sender and target see the message
<input type="checkbox"/> #3	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

Missing Caption



^COLLAPSE ^

Task #2 - Points: 1

Text: Screenshots of the related code

Checklist

*The checkboxes are for your own tracking

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

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

Missing Caption

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	
<input type="checkbox"/> #1	1	Include how the sender and receiver are handled	
<input type="checkbox"/> #2	1	Include how the username is used to get the proper id	

Response:

Missing Response



Mute/Unmute Users (3 pts.)

^COLLAPSE ^



Task #1 - Points: 1

Text: Screenshots demoing feature working

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Should have 3 clients in the same room
<input type="checkbox"/> #2	1	Demo mute preventing messages between the muter and the target
<input type="checkbox"/> #3	1	Demo mute also being accounted for with private messages
<input type="checkbox"/> #4	1	Demo unmute allowing the messages again from the target to the unmutter

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

Missing Caption

^COLLAPSE ^

Task #2 - Points: 1

Text: Screenshots of the related code

Checklist

*The checkboxes are for your own tracking

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

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

Missing Caption

^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
<input type="checkbox"/> #1	1	Explain how your mute list is handled
<input type="checkbox"/> #2	1	Explain how it's handled/processed in send message and private message

Response:

Missing Response



Misc (1 pt.)

^COLLAPSE ^



^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/alanpear/am3485-it114-002/pull/13>



^COLLAPSE ^

Task #2 - Points: 1

Text: Talk about any issues or learnings during this assignment

Response:

I have had various issues understanding how ClientUI and Client relate to each other and working things out so that roll and flip and private messages work I solved this using the videos recorded videos and looking up anything that wasn't covered in the notes.



^COLLAPSE ^

Task #3 - Points: 1

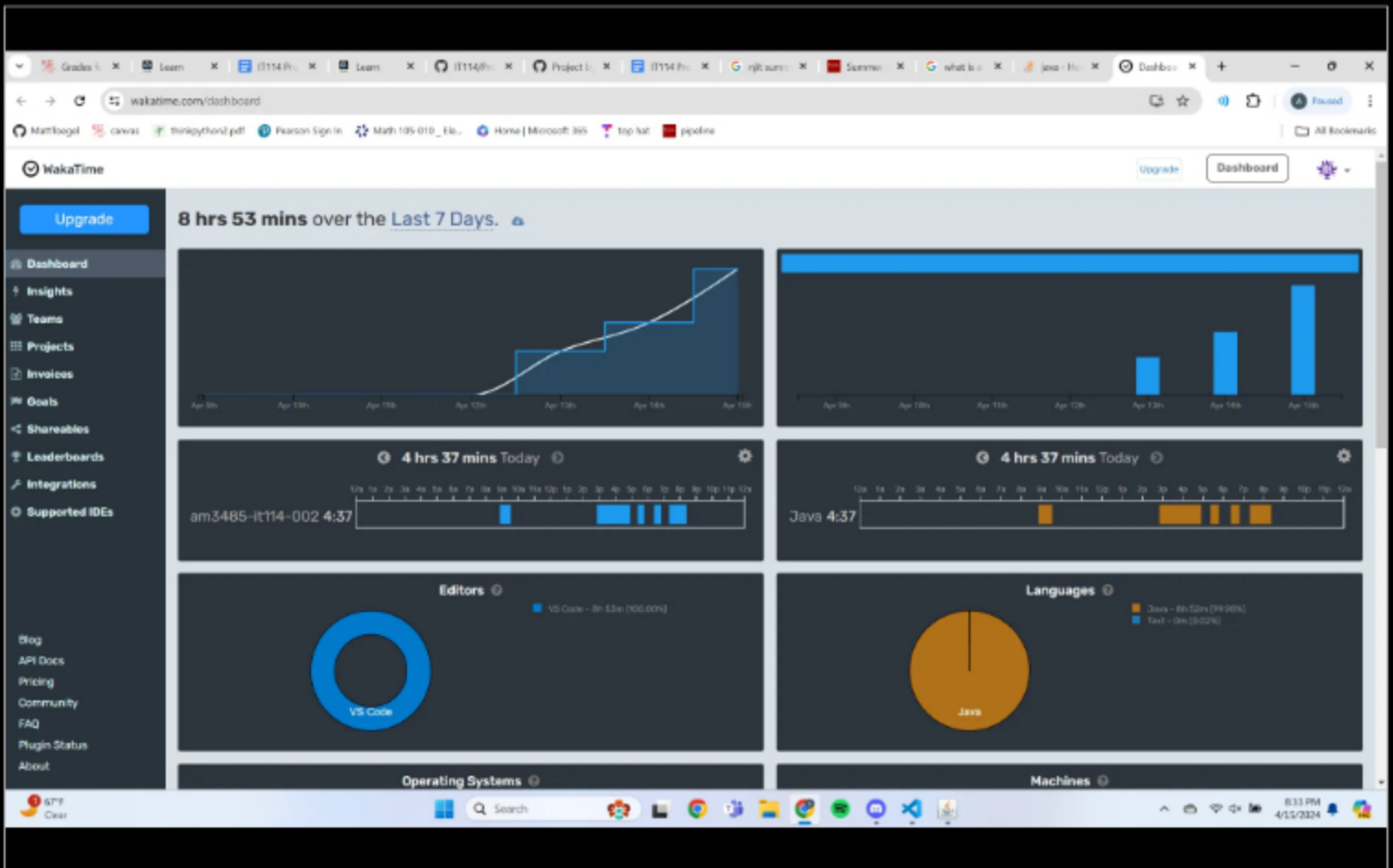
Text: WakaTime Screenshot

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



waka time screenshot

End of Assignment