

# Submission Worksheet

CLICK TO GRADE

<https://learn.ethereallab.app/assignment/IT114-006-S2024/it114-chatroom-milestone-3-2024/grade/sb57>

IT114-006-S2024 - [IT114] Chatroom Milestone 3 2024

## Submissions:

Submission Selection

1 Submission [active] 5/1/2024 8:15:06 PM

## 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



```

@Override
public void componentAdded(ContainerEvent e) {
    if (userListArea.isVisible()) {
        userListArea.revalidate();
        userListArea.repaint();
    }
}

@Override
public void componentRemoved(ContainerEvent e) {
    if (userListArea.isVisible()) {
        userListArea.revalidate();
        userListArea.repaint();
    }
}

}

protected void addNewUserToList(long clientId, String clientName) {
    logger.log(Level.DEBUG, "Adding user to list: " + clientName);
    JPanel content = userListArea;
    logger.log(Level.DEBUG, "User-List: " + content.getSize());
    JEditorPane textContainer = new JEditorPane("text/plain", clientName);
    textContainer.setName(clientId + "");
    textContainer.setLayout(GridLayout());
    textContainer.setPreferredSize(
        new Dimension(content.getWidth(), (clientId * 1.5).intValue()));
    textContainer.setOpaque(true);
    textContainer.setEditable(false);
    ClientInfo.clearBackground(textContainer);
    content.add(textContainer);
}

protected void removeUserFromList(long clientId) {
    logger.log(Level.DEBUG, "Removing user list item for id: " + clientId);
    Component[] cs = userListArea.getComponents();
    for (Component c : cs) {
        if (c.getName().equals(clientId + "")) {
            userListArea.remove(c);
            break;
        }
    }
}

protected void clearUserList() {
    Component[] cs = userListArea.getComponents();
    for (Component c : cs) {
        userListArea.remove(c);
    }
}
}

```

Screenshot of UserList

## Checklist Items (2)

### #2 User Details Panel

### #4 Clearly caption screenshots

```

public class ChatPanel extends JPanel {
    private static Logger logger = Logger.getLogger(ChatPanel.class.getName());
    private JPanel chatArea = null;
    private UserListPanel userListPanel;
    public ChatPanel(JComponent controls) {
        super(new BorderLayout());
        JPanel wrapper = new JPanel();
        wrapper.setLayout(new BorderLayout());
        JPanel content = new JPanel();
        content.setLayout(new BorderLayout());
        content.setAlignmentX(Component.BOTTOM_ALIGNMENT);
        JScrollPane scroll = new JScrollPane(content);
        scroll.setHorizontalScrollBarPolicy(ScrollPaneConstants.HORIZONTAL_SCROLLBAR_NEVER);
        scroll.setVerticalScrollBarPolicy(ScrollPaneConstants.VERTICAL_SCROLLBAR_AS_NEEDED);
        wrapper.add(scroll);
        this.add(wrapper, BorderLayout.CENTER);

        JPanel input = new JPanel();
        input.setLayout(new BorderLayout());
        JPasswordField testValue = new JPasswordField();
        input.add(testValue);
        JButton button = new JButton("Send");
        testValue.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                //Streak Boss
                //v057
                //April 15, 2020
            }
        });
        button.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                if (e.getKeyCode() == KeyEvent.VK_ENTER) {
                    button.doClick();
                }
            }
        });
        button.addActionListener(event -> {
            try {
                String text = testValue.getText().trim();
                if (text.length() > 0) {
                    ClientInfo.sendMessage(text);
                    testValue.setText(""); // clear the original text
                    logger.log(Level.FINEST, "Content: " + content.getSize());
                    logger.log(Level.FINEST, "Parent: " + this.getSize());
                }
            } catch (NullPointerException e) {
            } catch (IOException e) {
            }
            // TODO Auto-generated catch block
            e.printStackTrace();
        });
        chatArea = content;
        input.add(button);
        userListPanel = new UserListPanel(controls);
        this.add(userListPanel, BorderLayout.EAST);
        this.add(input, BorderLayout.NORTH);
        this.setName("Chat.DMF.name");
        controls.addPanel(Card.Chat.name(), this);
        chatArea.addComponentListener(new ComponentListener() {
            @Override
            public void componentAdded(ContainerEvent e) {
                if (chatArea.isVisible()) {
                    chatArea.revalidate();
                    chatArea.repaint();
                }
            }
        });
    }
}

```

Screenshot of ChatPanel part 1

## Checklist Items (2)

### #3 Chat Panel

## #4 Clearly caption screenshots

```

    }
    chatArea = content;
    input.add(button);
    userListPanel = new UserListPanel(controls);
    this.add(userListPanel, BorderLayout.EAST);
    this.add(input, BorderLayout.SOUTH);
    this.setName(Card.Chat.name());
    controls.addPanel(Card.Chat.name(), this);
    chatArea.addContainerListener(new ContainerListener() {
        @Override
        public void componentAdded(ContainerEvent e) {
            if (chatArea.isVisible()) {
                chatArea.revalidate();
                chatArea.repaint();
            }
        }

        @Override
        public void componentRemoved(ContainerEvent e) {
            //Shreya Boss
            //KMS
            //April 15, 2024
            if (chatArea.isVisible()) {
                chatArea.revalidate();
                chatArea.repaint();
            }
        }
    });
    this.addComponentListener(new ComponentAdapter() {
        @Override
        public void componentResized(ComponentEvent e) {
            Dimension frameSize = wrapper.getParent().getSize();
            int w = (int) Math.ceil(frameSize.getWidth() * .34);
            userListPanel.setPreferredSize(new Dimension(w, (int) frameSize.getHeight()));
            userListPanel.revalidate();
            userListPanel.repaint();
        }

        @Override
        public void componentMoved(ComponentEvent e) {
        }
    });

    public void addNewListItem(long clientId, String clientName) {
        userListPanel.addNewListItem(clientId, clientName);
    }

    public void removeNewListItem(long clientId) {
        userListPanel.removeNewListItem(clientId);
    }

    public void clearNewList() {
        userListPanel.clearNewList();
    }

    public void addText(String text) {
        JPanel content = chatArea;
        JScrollPane textContainer = new JScrollPane(text, text);
        textContainer.setLayout(new BorderLayout());
        textContainer.setPreferredSize(
            new Dimension(content.getWidth(), ClientUtils.calculateHeightForText(this, text, content.getWidth())));
        textContainer.setMaximumSize(textContainer.getPreferredSize());
        textContainer.setEditable(false);
        ClientUtils.clearBackground(textContainer);
        content.add(textContainer);
        JScrollBar vertical = ((JScrollPane) chatArea.getParent().getParent()).getVerticalScrollBar();
        vertical.setValue(vertical.getMaximum());
    }
}

```

Screenshot of ChatPanel part 2

## Checklist Items (2)

## #3 Chat Panel

## #4 Clearly caption screenshots

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
#1	1	Flip output in a different format than normal messages
#2	1	Roll # output in a different format than normal messages

#3	1	Roll #d# output in a different format than normal messages
#4	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

SmallMediumLarge

Project.Server.ServerThread info  
INFO: Thread[Shreya]: Received from client: Type[MESSAGE], Message[/roll 6], ClientId[0], ClientName[null]  
May 01, 2024 4:30:09 PM Project.Server.Room sendMessage  
INFO: Sending message to 2 clients  
May 01, 2024 4:30:09 PM Project.Server.Room sendMessage  
INFO: Sending message to 2 clients  
May 01, 2024 4:32:32 PM Project.Server.ServerThread info  
INFO: Thread[Shreya]: Received from client: Type[MESSAGE], Message[/flip], ClientId[0], ClientName[null]  
May 01, 2024 4:32:32 PM Project.Server.Room sendMessage  
INFO: Sending message to 2 clients  
May 01, 2024 4:32:32 PM Project.Server.Room sendMessage  
INFO: Sending message to 2 clients  
May 01, 2024 4:32:36 PM Project.Server.ServerThread info  
INFO: Thread[Bud]: Received from client: Type[MESSAGE], Message[/flip], ClientId[0], ClientName[null]  
May 01, 2024 4:32:36 PM Project.Server.Room sendMessage  
INFO: Sending message to 2 clients  
May 01, 2024 4:32:36 PM Project.Server.Room sendMessage  
INFO: Sending message to 2 clients  
[]

Id[2], ClientName[null]Bud: Rolled a 1 out of 1 /roll 6  
May 01, 2024 4:30:09 PM Project.Client.Client\$1 run  
INFO: Waiting for inputMay 01, 2024 4:30:09 PM Project.Client.Client\$2 run  
INFO: Debug Info: Type[MESSAGE], Message[Rolled a 3 out of 6], ClientId[1], ClientName[null]Shreya: Rolled a 3 out of 6 /flip  
May 01, 2024 4:32:32 PM Project.Client.Client\$1 run  
INFO: Waiting for inputMay 01, 2024 4:32:32 PM Project.Client.Client\$2 run  
INFO: Debug Info: Type[MESSAGE], Message[Flipped a coin and got tails], ClientId[1], ClientName[null]  
Shreya: Flipped a coin and got tails  
May 01, 2024 4:32:36 PM Project.Client.Client\$2 run  
INFO: Debug Info: Type[MESSAGE], Message[Flipped a coin and got heads], ClientId[2], ClientName[null]  
Bud: Flipped a coin and got heads  
[]

INFO: Debug Info: Type[MESSAGE], Message[Rolled a 1 out of 1], ClientId[2], ClientName[null]  
Bud: Rolled a 1 out of 1  
May 01, 2024 4:30:09 PM Project.Client.Client\$2 run  
INFO: Debug Info: Type[MESSAGE], Message[Rolled a 3 out of 6], ClientId[1], ClientName[null]  
Shreya: Rolled a 3 out of 6  
May 01, 2024 4:32:32 PM Project.Client.Client\$2 run  
INFO: Debug Info: Type[MESSAGE], Message[Flipped a coin and got tails], ClientId[1], ClientName[null]  
Shreya: Flipped a coin and got tails /flip  
May 01, 2024 4:32:36 PM Project.Client.Client\$1 run  
INFO: Waiting for input  
May 01, 2024 4:32:36 PM Project.Client.Client\$2 run  
INFO: Debug Info: Type[MESSAGE], Message[Flipped a coin and got heads], ClientId[2], ClientName[null]  
Bud: Flipped a coin and got heads  
[]

Shows flip output

Checklist Items (2)

- #1 Flip output in a different format than normal messages
- #4 Clearly caption screenshots

INFO: Starting queue manager  
May 01, 2024 4:29:16 PM Project.Server.Server start  
INFO: waiting for next client  
May 01, 2024 4:29:22 PM Project.Server.Server start  
INFO: waiting for next client  
May 01, 2024 4:29:22 PM Project.Server.Server start  
INFO: Client connected  
May 01, 2024 4:29:22 PM Project.Server.ServerThread info  
INFO: Thread[null]: Thread created  
May 01, 2024 4:29:22 PM Project.Server.ServerThread info  
INFO: Thread[null]: Thread starting  
May 01, 2024 4:29:22 PM Project.Server.ServerThread info  
INFO: Thread[null]: Received from client: Type[CONNECT], Message[null], ClientId[0], ClientName[Shreya], Client name Shreya  
May 01, 2024 4:29:22 PM Project.Server.Server joinRoom  
INFO: Thread-1 joining room lobby  
May 01, 2024 4:29:26 PM Project.Server.Server start  
INFO: waiting for next client  
May 01, 2024 4:29:26 PM Project.Server.Server start  
INFO: Client connected  
May 01, 2024 4:29:26 PM Project.Server.ServerThread info  
INFO: Thread[null]: Thread created  
May 01, 2024 4:29:26 PM Project.Server.ServerThread info  
INFO: Thread[null]: Thread starting

INFO: Client connected  
May 01, 2024 4:29:22 PM Project.Client.Client\$1 run  
INFO: Waiting for input  
May 01, 2024 4:29:22 PM Project.Client.Client\$2 run  
INFO: Debug Info: Type[CLIENT\_ID], Message[null], ClientId[1], ClientName[Shreya], Client name Shreya  
May 01, 2024 4:29:22 PM Project.Client.Client processPayload  
INFO: My Client Id is 1  
May 01, 2024 4:29:22 PM Project.Client.Client\$2 run  
INFO: Debug Info: Type[JOIN\_ROOM], Message[lobby], ClientId[0], ClientName[null]  
May 01, 2024 4:29:22 PM Project.Client.Client\$2 run  
INFO: Debug Info: Type[CONNECT], Message[connected], ClientId[1], ClientName[Shreya], Client name Shreya  
May 01, 2024 4:29:22 PM Project.Client.Client processPayload  
INFO: \*Shreya connected\*  
May 01, 2024 4:29:26 PM Project.Client.Client\$2 run  
INFO: Debug Info: Type[CONNECT], Message[connected], ClientId[2], ClientName[Bud], Client name Bud  
May 01, 2024 4:29:26 PM Project.Client.Client processPayload  
INFO: \*Bud connected\*  
/roll

INFO: Closing input stream  
May 01, 2024 4:29:04 PM Project.Client.Client close  
INFO: Closing connection  
May 01, 2024 4:29:04 PM Project.Client.Client close  
SEVERE: Closed socket  
May 01, 2024 4:29:04 PM Project.Client.Client\$2 run  
INFO: Stopped listening to server input  
/connect localhost:3000  
May 01, 2024 4:29:26 PM Project.Client.Client connect  
INFO: Client connected  
May 01, 2024 4:29:26 PM Project.Client.Client\$1 run  
INFO: Waiting for input  
May 01, 2024 4:29:26 PM Project.Client.Client\$2 run  
INFO: Debug Info: Type[CLIENT\_ID], Message[null], ClientId[2], ClientName[Bud], Client name Bud  
May 01, 2024 4:29:26 PM Project.Client.Client processPayload  
INFO: My Client Id is 2  
May 01, 2024 4:29:26 PM Project.Client.Client\$2 run  
INFO: Debug Info: Type[JOIN\_ROOM], Message[lobby], ClientId[0], ClientName[null]  
May 01, 2024 4:29:26 PM Project.Client.Client\$2 run  
INFO: Debug Info: Type[CONNECT], Message[connected], ClientId[2], ClientName[Bud], Client name Bud  
May 01, 2024 4:29:26 PM Project.Client.Client processPayload



```
May 01, 2024 4:29:26 PM Project.Server.ServerThread info
INFO: Thread[null]: Received from client: Type[CONNECT],
Message[null], ClientId[0], ClientName[Bud], Client name
Bud
May 01, 2024 4:29:26 PM Project.Server.Server joinRoom
INFO: Thread-2 joining room lobby
May 01, 2024 4:29:32 PM Project.Server.ServerThread info
INFO: Thread[Shreya]: Received from client: Type[MESSAGE],
Message[/roll], ClientId[0], ClientName[null]
May 01, 2024 4:29:32 PM Project.Server.Room sendMessage
INFO: Sending message to 2 clients
May 01, 2024 4:29:32 PM Project.Server.Room sendMessage
INFO: Sending message to 2 clients
May 01, 2024 4:29:43 PM Project.Server.ServerThread info
INFO: Thread[Shreya]: Received from client: Type[MESSAGE],
Message[/roll 2d6], ClientId[0], ClientName[null]

May 01, 2024 4:29:43 PM Project.Server.Room sendMessage
INFO: Sending message to 2 clients
Project.Server.ServerThread info
INFO: Thread[Shreya]: Received from client: Type[MESSAGE],
Message[/roll 6], ClientId[0], ClientName[null]
May 01, 2024 4:30:09 PM Project.Server.Room sendMessage
INFO: Sending message to 2 clients
May 01, 2024 4:30:09 PM Project.Server.Room sendMessage
INFO: Sending message to 2 clients
[]
```

```
May 01, 2024 4:29:32 PM Project.Client.Client$1 run
INFO: Waiting for input
May 01, 2024 4:29:32 PM Project.Client.Client$2 run
INFO: Debug Info: Type[MESSAGE], Message[Invalid roll com
mand. Please specify the roll parameters, e.g., '/roll 2d
6' or '/roll 100'.], ClientId[1], ClientName[null]
Shreya: Invalid roll command. Please specify the roll par
ameters, e.g., '/roll 2d6' or '/roll 100'.
/roll 2d6
Id[2], ClientName[null]Bud: Rolled a 1 out of 1
/roll 6
May 01, 2024 4:30:09 PM Project.Client.Client$1 run
INFO: Waiting for inputMay 01, 2024 4:30:09 PM Project.Cl
ient.Client$2 run
INFO: Debug Info: Type[MESSAGE], Message[Rolled a 3 out o
f 6], ClientId[1], ClientName[null]Shreya: Rolled a 3 out
of 6
[]
```

```
load
INFO: *Bud connected*
May 01, 2024 4:29:26 PM Project.Client.Client$2 run
INFO: Debug Info: Type[SYNC_CLIENT], Message[null], Clie
ntId[1], ClientName[Shreya], Client name Shreya
May 01, 2024 4:29:32 PM Project.Client.Client$2 run
INFO: Debug Info: Type[MESSAGE], Message[Invalid roll co
mmand. Please specify the roll parameters, e.g., '/roll
2d6' or '/roll 100'.], ClientId[1], ClientName[null]
INFO: Debug Info: Type[MESSAGE], Message[Rolled a 1 out
of 1], ClientId[2], ClientName[null]
Bud: Rolled a 1 out of 1
May 01, 2024 4:30:09 PM Project.Client.Client$2 run
INFO: Debug Info: Type[MESSAGE], Message[Rolled a 3 out
of 6], ClientId[1], ClientName[null]
Shreya: Rolled a 3 out of 6
[]
```

Shows roll output for # and for #d#

Checklist Items (3)

#2 Roll # output in a different format than normal messages

#3 Roll #d# output in a different format than normal messages

#4 Clearly caption screenshots

●

⌵ 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 type="checkbox"/> #4	1	Custom text formatting for red working (Part of the message should appear red)	
<input type="checkbox"/> #5	1	Custom text formatting for blue working (Part of the message should appear blue)	
<input 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 type="checkbox"/> #8	1	Clearly caption screenshots	

Task Screenshots:

SmallMediumLarge

Project.Server.Server joinRoom  
INFO: Thread-1 joining room test  
May 01, 2024 5:13:48 PM Project.Server.ServerThread info  
INFO: Thread[Shreya]: Received from client: Type[MESSAGE]  
, Message[hi], ClientId[0], ClientName[null]  
May 01, 2024 5:13:48 PM Project.Server.Room sendMessage  
INFO: Sending message to 1 clients  
May 01, 2024 5:15:21 PM Project.Server.ServerThread info  
INFO: Thread[Shreya]: Received from client: Type[MESSAGE]  
, Message[\*hi\*], ClientId[0], ClientName[null]  
May 01, 2024 5:15:21 PM Project.Server.Room sendMessage  
INFO: Sending message to 1 clients  
May 01, 2024 5:15:55 PM Project.Server.ServerThread info  
INFO: Thread[Shreya]: Received from client: Type[MESSAGE]  
, Message[\*hi\*], ClientId[0], ClientName[null]  
May 01, 2024 5:15:55 PM Project.Server.Room sendMessage  
INFO: Sending message to 1 clients  
May 01, 2024 5:16:03 PM Project.Server.ServerThread info  
INFO: Thread[Bud]: Received from client: Type[MESSAGE], M  
essage[\_hello\_], ClientId[0], ClientName[null]  
May 01, 2024 5:16:03 PM Project.Server.Room sendMessage  
INFO: Sending message to 1 clients  
May 01, 2024 5:16:50 PM Project.Server.ServerThread info  
INFO: Thread[Shreya]: Received from client: Type[MESSAGE]  
, Message[-hi-], ClientId[0], ClientName[null]  
May 01, 2024 5:16:50 PM Project.Server.Room sendMessage  
INFO: Sending message to 1 clients  
█

oad  
INFO: \*Shreya connected\*  
May 01, 2024 5:13:26 PM Project.Client.Client\$2 run  
INFO: Debug Info: Type[CONNECT], Message[connected], Clie  
ntId[2], ClientName[Bud], Client name Bud  
May 01, 2024 5:13:26 PM Project.Client.Client processPayl  
CONNECT], Message[connected], ClientId[1], ClientName[Shr  
eya], Client name Shreya  
May 01, 2024 5:13:44 PM Project.Client.Client processPayl  
oad  
INFO: \*Shreya connected\*  
hi  
May 01, 2024 5:13:48 PM Project.Client.Client\$1 run  
INFO: Waiting for inputMay 01, 2024 5:13:48 PM Project.Cl  
ient.Client\$2 run  
INFO: Debug Info: Type[MESSAGE], Message[hi], ClientId[1]  
, ClientName[null]  
Shreya: hi  
\*hi\*  
May 01, 2024 5:15:21 PM Project.Client.Client\$1 run  
INFO: Waiting for input  
May 01, 2024 5:15:21 PM Project.Client.Client\$2 run  
INFO: Debug Info: Type[MESSAGE], Message[\*hi\*], ClientId[1]  
, ClientName[null]  
Shreya: \*hi\*  
\*hi\*  
May 01, 2024 5:15:55 PM Project.Client.Client\$1 run  
INFO: Waiting for input  
May 01, 2024 5:15:55 PM Project.Client.Client\$2 run  
INFO: Debug Info: Type[MESSAGE], Message[hi], ClientId[1]  
, ClientName[null]  
Shreya: hi  
~hi~  
May 01, 2024 5:16:50 PM Project.Client.Client\$1 run  
INFO: Waiting for input  
May 01, 2024 5:16:50 PM Project.Client.Client\$2 run  
INFO: Debug Info: Type[MESSAGE], Message[hi], ClientId[1]  
, ClientName[null]  
Shreya: hi  
█

May 01, 2024 5:12:49 PM Project.Client.Client\$1 run  
INFO: Listening for input  
May 01, 2024 5:12:49 PM Project.Client.Client\$1 run  
INFO: Waiting for input  
/name Bud  
May 01, 2024 5:13:21 PM Project.Client.Client \$sName  
INFO: Name set to Bud  
May 01, 2024 5:13:21 PM Project.Client.Client\$1 run  
INFO: Waiting for input  
/connect localhost:3000  
May 01, 2024 5:13:26 PM Project.Client.Client connect  
INFO: Client connected  
May 01, 2024 5:13:26 PM Project.Client.Client\$1 run  
INFO: Waiting for input  
May 01, 2024 5:13:26 PM Project.Client.Client\$2 run  
INFO: Debug Info: Type[CLIENT\_ID], Message[null], Client  
Id[2], ClientName[Bud], Client name Bud  
Bud: hi  
May 01, 2024 5:13:38 PM Project.Client.Client\$2 run  
INFO: Debug Info: Type[MESSAGE], Message[hello], ClientI  
d[1], ClientName[null]  
Shreya: hello  
May 01, 2024 5:13:44 PM Project.Client.Client\$2 run  
INFO: Debug Info: Type[DISCONNECT], Message[disconnected  
, ClientId[1], ClientName[Shreya], Client name ShreyaMa  
y 01, 2024 5:13:44 PM Project.Client.Client processPaylo  
ad  
INFO: \*Shreya disconnected\*  
\_hello  
May 01, 2024 5:16:03 PM Project.Client.Client\$1 run  
INFO: Waiting for input  
May 01, 2024 5:16:03 PM Project.Client.Client\$2 run  
INFO: Debug Info: Type[MESSAGE], Message[hello], ClientI  
d[2], ClientName[null]  
Bud: hello  
█

Shows examples of all formatting

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:

SmallMediumLarge

```

    }
    //Shreya Bose
    //sb56
    //April 15, 2024

private String formatMessage(String message) {
    message = message.replaceAll(regex:"\\*([\\^\\*]+)\\*", TextFX.colorize(text:"$1", TextFX.Color.RED)); // Bold as red
    message = message.replaceAll(regex:"_([\\^_]+)_", TextFX.colorize(text:"$1", TextFX.Color.GREEN)); // Italics as green
    message = message.replaceAll(regex:"~([\\^~]+)~", TextFX.colorize(text:"$1", TextFX.Color.BLUE)); // Underline as blue
    return message;
}

```

Screenshot showing formatting

#### Checklist Items (3)

#1 Show each relevant file this was done in (may be one or more)

#2 Include ucid and date comment

#3 Clearly caption screenshots



^ COLLAPSE ^

#### Task #4 - Points: 1

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

#### Details:

Note each scenario

Response:

The formatting was handled in Room



Private Message with @ (2 pts.)

^ COLLAPSE ^



Task #1 - Points: 1

Text: Screenshots demoing private message

Checklist		*The checkboxes are for your own tracking
#	Points	Details
#1	1	Should have 3 clients in the same room
#2	1	Demo a private message where only the sender and target see the message
#3	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

SmallMediumLarge

<div>DINFO: Thread-1 leaving room lobby</div> <div>May 01, 2024 8:34:12 PM Project.Server.Server joinRoom</div> <div>DINFO: Thread-1 joining room Private</div> <div>May 01, 2024 8:34:26 PM Project.Server.ServerThread Info</div> <div>DINFO: Thread[Bad]: Received from client: Type[JOIN_ROOM], Message[Private], ClientId[0], ClientName[null]</div> <div>May 01, 2024 8:34:26 PM Project.Server.Server joinRoom</div> <div>DINFO: Thread-2 leaving room lobby</div> <div>May 01, 2024 8:34:26 PM Project.Server.Server joinRoom</div> <div>DINFO: Thread-2 joining room Private</div> <div>May 01, 2024 8:34:31 PM Project.Server.ServerThread Info</div> <div>DINFO: Thread[Shreya]: Received from client: Type[MESSAGE], Message[hi], ClientId[0], ClientName[null]</div> <div>May 01, 2024 8:34:31 PM Project.Server.Room sendMessage</div> <div>DINFO: Sending message to 2 clients</div> <div>May 01, 2024 8:34:39 PM Project.Server.ServerThread Info</div> <div>DINFO: Thread[Bad]: Received from client: Type[MESSAGE], Message[hello], ClientId[0], ClientName[null]</div> <div>May 01, 2024 8:34:39 PM Project.Server.Room sendMessage</div> <div>DINFO: Sending message to 2 clients</div> <div></div>	<div>DINFO: Waiting for input</div> <div>May 01, 2024 8:34:12 PM Project.Client.Client\$2 run</div> <div>DINFO: Debug Info: Type[JOIN_ROOM], Message[Private], ClientId[0], ClientName[null]</div> <div>May 01, 2024 8:34:12 PM Project.Client.Client\$2 run</div> <div>DINFO: Debug Info: Type[CONNECT], Message[connected], ClientId[1], ClientName[Shreya], Client name Shreya</div> <div>May 01, 2024 8:34:12 PM Project.Client.Client processPayload</div> <div>DINFO: "Shreya connected"</div> <div>May 01, 2024 8:34:26 PM Project.Client.Client\$2 run</div> <div>DINFO: Debug Info: Type[CONNECT], Message[connected], ClientId[2], ClientName[Bad], Client name Bad</div> <div>May 01, 2024 8:34:26 PM Project.Client.Client processPayload</div> <div>DINFO: "Bad connected"</div> <div>hi</div> <div>May 01, 2024 8:34:31 PM Project.Client.Client\$1 run</div> <div>DINFO: Waiting for input</div> <div>May 01, 2024 8:34:31 PM Project.Client.Client\$2 run</div> <div>DINFO: Debug Info: Type[MESSAGE], Message[hi], ClientId[1], ClientName[null]</div> <div>Shreya: hi</div> <div>May 01, 2024 8:34:39 PM Project.Client.Client\$2 run</div> <div>DINFO: Debug Info: Type[MESSAGE], Message[hello], ClientId[2], ClientName[null]</div> <div>Bad: hello</div> <div></div>	<div>/JoinRoom Private</div> <div>May 01, 2024 8:34:26 PM Project.Client.Client\$1 run</div> <div>DINFO: Waiting for input</div> <div>May 01, 2024 8:34:26 PM Project.Client.Client\$2 run</div> <div>DINFO: Debug Info: Type[JOIN_ROOM], Message[Private], ClientId[0], ClientName[null]</div> <div>May 01, 2024 8:34:26 PM Project.Client.Client\$2 run</div> <div>DINFO: Debug Info: Type[CONNECT], Message[connected], ClientId[2], ClientName[Bad], Client name Bad</div> <div>May 01, 2024 8:34:26 PM Project.Client.Client processPayload</div> <div>DINFO: "Bad connected"</div> <div>May 01, 2024 8:34:26 PM Project.Client.Client\$2 run</div> <div>DINFO: Debug Info: Type[SWK_CLIENT], Message[null], ClientId[1], ClientName[Shreya], Client name Shreya</div> <div>May 01, 2024 8:34:31 PM Project.Client.Client\$2 run</div> <div>DINFO: Debug Info: Type[MESSAGE], Message[hi], ClientId[1], ClientName[null]</div> <div>Shreya: hi</div> <div>hello</div> <div>May 01, 2024 8:34:39 PM Project.Client.Client\$1 run</div> <div>DINFO: Waiting for input</div> <div>May 01, 2024 8:34:39 PM Project.Client.Client\$2 run</div> <div>DINFO: Debug Info: Type[MESSAGE], Message[hello], ClientId[2], ClientName[null]</div> <div>Bad: hello</div> <div></div>	<div>DINFO: Debug Info: Type[CONNECT], Message[connected], ClientId[3], ClientName[Random], Client name Random</div> <div>May 01, 2024 8:33:23 PM Project.Client.Client processPayload</div> <div>DINFO: "Random connected"</div> <div>May 01, 2024 8:33:23 PM Project.Client.Client\$2 run</div> <div>DINFO: Debug Info: Type[SWK_CLIENT], Message[null], ClientId[1], ClientName[Shreya], Client name Shreya</div> <div>May 01, 2024 8:33:23 PM Project.Client.Client\$2 run</div> <div>DINFO: Debug Info: Type[SWK_CLIENT], Message[null], ClientId[2], ClientName[Bad], Client name Bad</div> <div>May 01, 2024 8:34:12 PM Project.Client.Client\$2 run</div> <div>DINFO: Debug Info: Type[DISCONNECT], Message[disconnected], ClientId[1], ClientName[Shreya], Client name Shreya</div> <div>May 01, 2024 8:34:12 PM Project.Client.Client processPayload</div> <div>DINFO: "Shreya disconnected"</div> <div>May 01, 2024 8:34:26 PM Project.Client.Client\$2 run</div> <div>DINFO: Debug Info: Type[DISCONNECT], Message[disconnected], ClientId[2], ClientName[Bad], Client name Bad</div> <div>May 01, 2024 8:34:26 PM Project.Client.Client processPayload</div> <div>DINFO: "Bad disconnected"</div> <div></div>
---	--	---	--

Screenshot of Demo

Checklist Items (3)

- #1 Should have 3 clients in the same room
- #2 Demo a private message where only the sender and target see the message
- #3 Clearly caption screenshots



^ 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

Screenshot of

## Checklist Items (0)



^ COLLAPSE ^

### 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:

The sender only sends to the room they are in, which is handled in Room.java.



## Mute/Unmute Users (3 pts.)

^ COLLAPSE ^



^ COLLAPSE ^

### Task #1 - Points: 1

Text: Screenshots demoing feature working

#### Checklist

\*The checkboxes are for your own tracking

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

```
DBUG: Thread-1 leaving room lobby
May 01, 2024 8:34:12 PM Project.Server.Server joinRoom
DBUG: Thread-1 joining room Private
May 01, 2024 8:34:28 PM Project.Server.ServerThread Info
DBUG: Thread[Bad]: Received from client: Type[JOIN_ROOM], Message[Private], ClientId[0], ClientName[null]
May 01, 2024 8:34:26 PM Project.Server.Server joinRoom
DBUG: Thread-2 leaving room lobby
May 01, 2024 8:34:26 PM Project.Server.Server joinRoom
DBUG: Thread-2 joining room Private
May 01, 2024 8:34:31 PM Project.Server.ServerThread Info
DBUG: Thread[Shreya]: Received from client: Type[MESSAGE], Message[hi], ClientId[0], ClientName[null]
May 01, 2024 8:34:31 PM Project.Server.Room sendMessage
DBUG: Sending message to 2 clients
May 01, 2024 8:34:39 PM Project.Server.ServerThread Info
DBUG: Thread[Bad]: Received from client: Type[MESSAGE], Message[hello], ClientId[0], ClientName[null]
May 01, 2024 8:34:39 PM Project.Server.Room sendMessage
DBUG: Sending message to 2 clients
May 01, 2024 8:34:12 PM Project.Client.Client$2 run
DBUG: Debug Info: Type[JOIN_ROOM], Message[Private], ClientId[0], ClientName[null]
May 01, 2024 8:34:12 PM Project.Client.Client$2 run
DBUG: Debug Info: Type[CONNECT], Message[connected], ClientId[1], ClientName[Shreya], Client name Shreya
May 01, 2024 8:34:12 PM Project.Client.Client processPayload
DBUG: "Shreya connected"
May 01, 2024 8:34:26 PM Project.Client.Client$2 run
DBUG: Debug Info: Type[CONNECT], Message[connected], ClientId[2], ClientName[Bad], Client name Bad
May 01, 2024 8:34:26 PM Project.Client.Client processPayload
DBUG: "Bad connected"
hi
May 01, 2024 8:34:31 PM Project.Client.Client$1 run
DBUG: Waiting for input
May 01, 2024 8:34:31 PM Project.Client.Client$2 run
DBUG: Debug Info: Type[MESSAGE], Message[hi], ClientId[1], ClientName[null]
Shreya: hi
May 01, 2024 8:34:39 PM Project.Client.Client$2 run
DBUG: Debug Info: Type[MESSAGE], Message[hello], ClientId[2], ClientName[null]
Bad: hello
May 01, 2024 8:34:26 PM Project.Client.Client$1 run
DBUG: Waiting for input
May 01, 2024 8:34:26 PM Project.Client.Client$2 run
DBUG: Debug Info: Type[JOIN_ROOM], Message[Private], ClientId[0], ClientName[null]
May 01, 2024 8:34:26 PM Project.Client.Client$2 run
DBUG: Debug Info: Type[CONNECT], Message[connected], ClientId[0], ClientName[null]
May 01, 2024 8:34:26 PM Project.Client.Client processPayload
DBUG: "Shreya connected"
May 01, 2024 8:33:23 PM Project.Client.Client$2 run
DBUG: Debug Info: Type[SWC_CLIENT], Message[null], ClientId[1], ClientName[Shreya], Client name Shreya
May 01, 2024 8:33:23 PM Project.Client.Client$2 run
DBUG: Debug Info: Type[SWC_CLIENT], Message[null], ClientId[2], ClientName[Bad], Client name Bad
May 01, 2024 8:34:12 PM Project.Client.Client$2 run
DBUG: Debug Info: Type[DISCONNECT], Message[disconnected], ClientId[1], ClientName[Shreya], Client name Shreya
May 01, 2024 8:34:12 PM Project.Client.Client processPayload
DBUG: "Shreya disconnected"
May 01, 2024 8:34:26 PM Project.Client.Client$2 run
DBUG: Debug Info: Type[DISCONNECT], Message[disconnected], ClientId[2], ClientName[Bad], Client name Bad
May 01, 2024 8:34:26 PM Project.Client.Client processPayload
DBUG: "Bad disconnected"
```

Screenshot of 3 clients

Checklist Items (4)

- #1 Should have 3 clients in the same room
- #2 Demo mute preventing messages between the muter and the target
- #3 Demo mute also being accounted for with private messages
- #4 Demo unmute allowing the messages again from the target to the unmutter

^ COLLAPSE ^

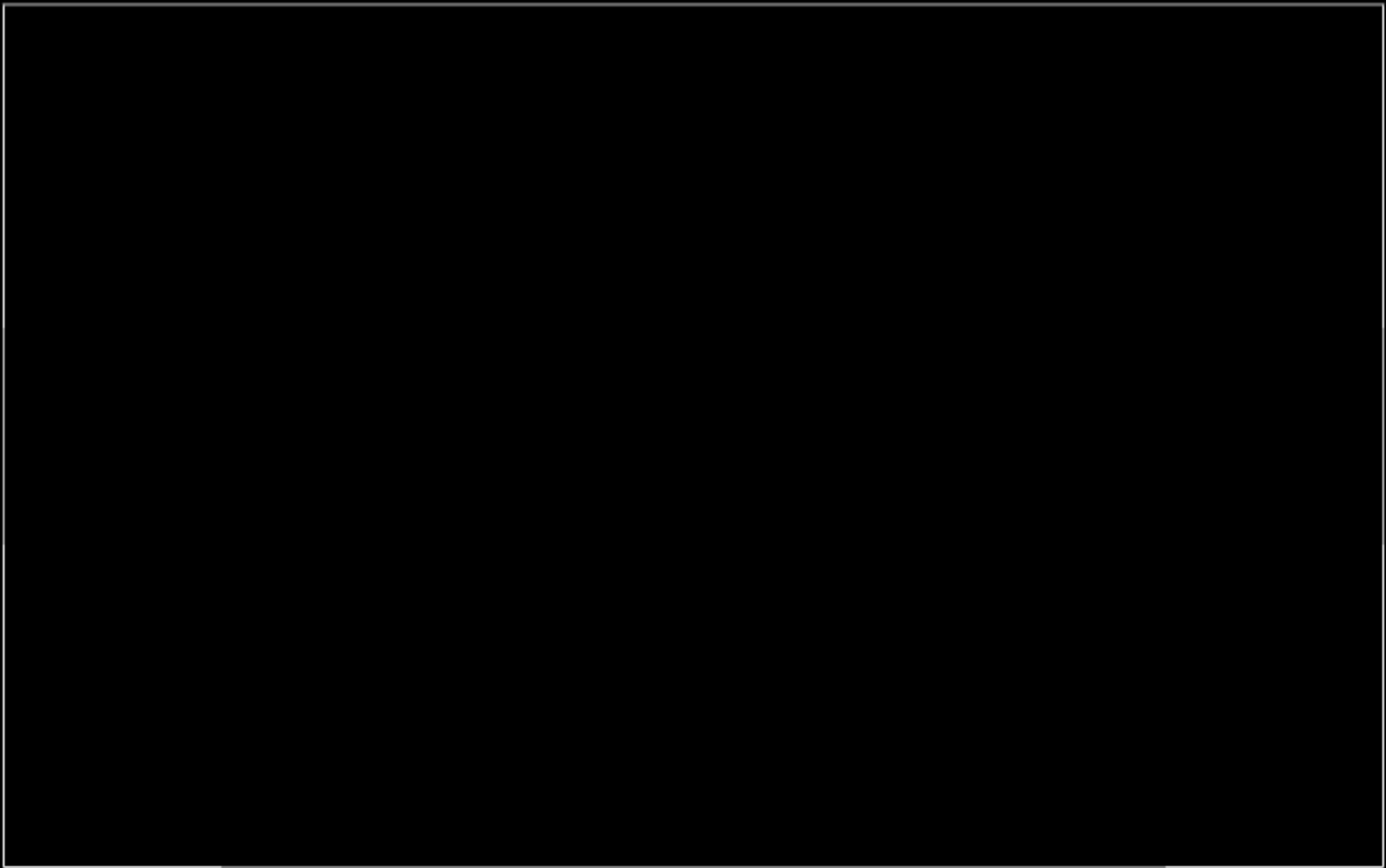
Task #2 - Points: 1  
Text: Screenshots of the related code

Checklist		*The checkboxes are for your own tracking
#	Points	Details
<div><div></div><div>#1</div></div>	1	ServerThread should have a list of who they muted
<div><div></div><div>#2</div></div>	1	ServerThread should expose and add, remove, and is muted check to room
<div><div></div><div>#3</div></div>	1	Room should handle the mute list when receiving the appropriate payloads
<div><div></div><div>#4</div></div>	1	Room should check the mute list during send message and private messages
<div><div></div><div>#5</div></div>	1	Include ucid and date comment
<div><div></div><div>#6</div></div>	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

SmallMediumLarge



Did not have mute feature.

Checklist Items (0)

^ 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:

Did not handle a mute list.



^ 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/sb57-shreya/sb57-IT114-006/compare/Milestone3?expand=1>



^ COLLAPSE ^

### Task #2 - Points: 1

Text: Talk about any issues or learnings during this assignment

Response:

This Milestone was extremely difficult for me but I am grasping a slight understanding.



^ 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.

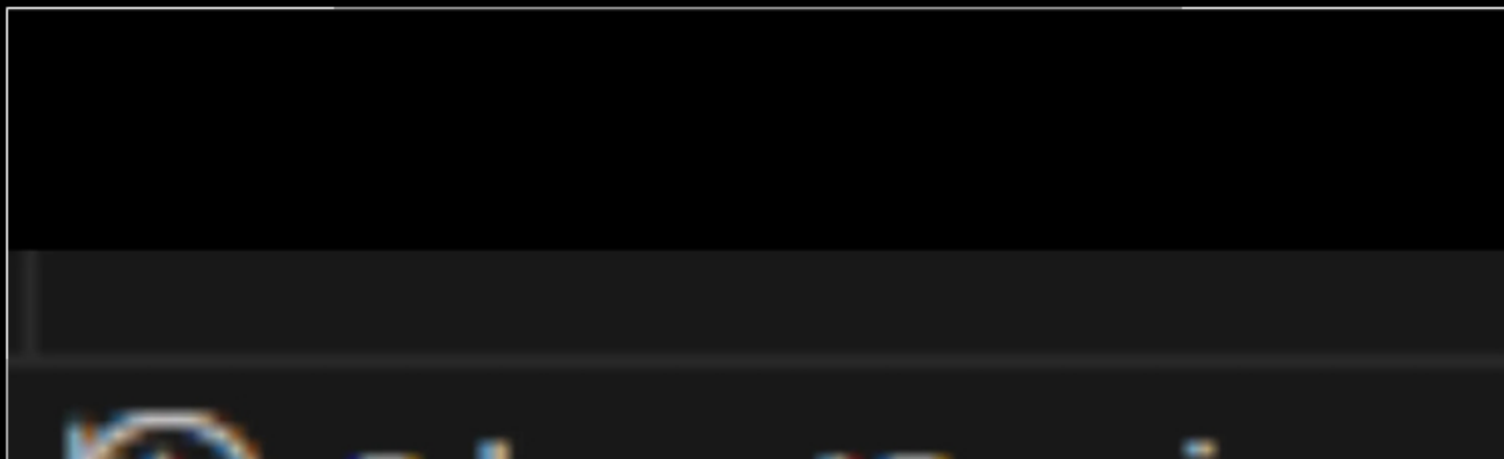
Task Screenshots:

Gallery Style: Large View

Small

Medium

Large



5 6 hrs 42 mins

Screenshot of WakaTime

End of Assignment