Submission Worksheet

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

Submissions:

Submission Selection

1 Submission [active] 4/29/2024 8:20:46 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 ∧

●
∧COLLAPSE ∧

Task #1 - Points: 1

Text: Screenshots of the following

Checklist   *The checkboxes are for your own tracking
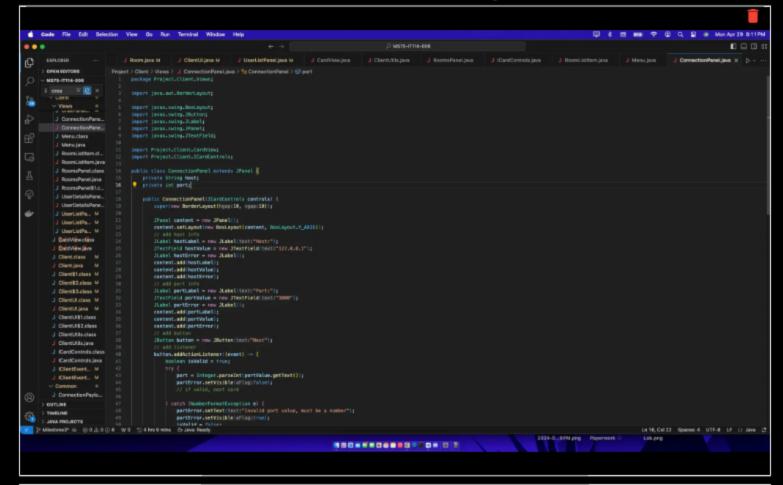
#	Points	Details

☐ #1	1		Connection Panel
☐ #2	1		User Details Panel
☐ #3	1		Chat Panel
☐ #4	1		Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small   Medium   Large

Connection Panel Part 1

Checklist Items (1)

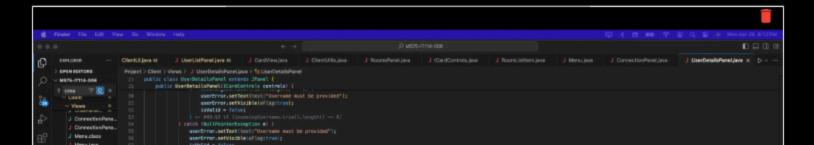#1 Connection Panel

Checklist Items (1)

#1 Connection Panel
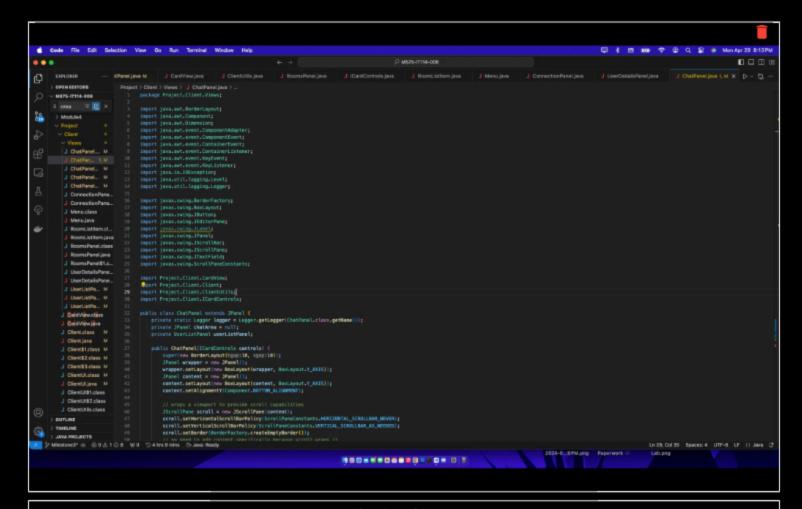

UserDetailsPanel Part 1

Checklist Items (1)

#2 User Details Panel

UserDetailsPanel Part 2
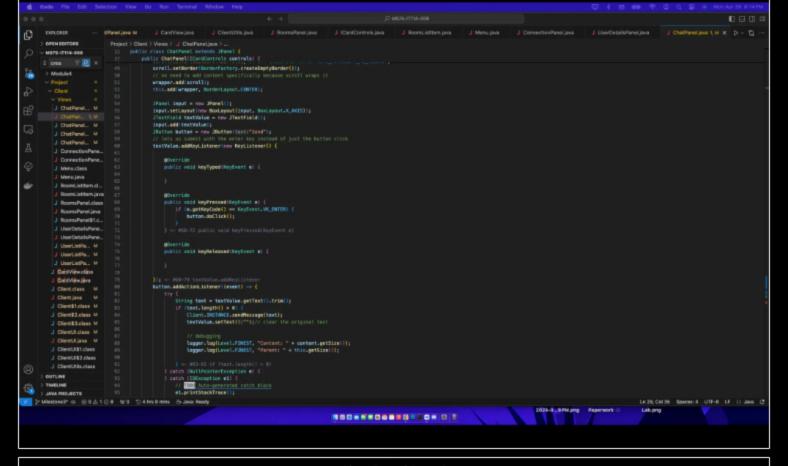
Checklist Items (1)
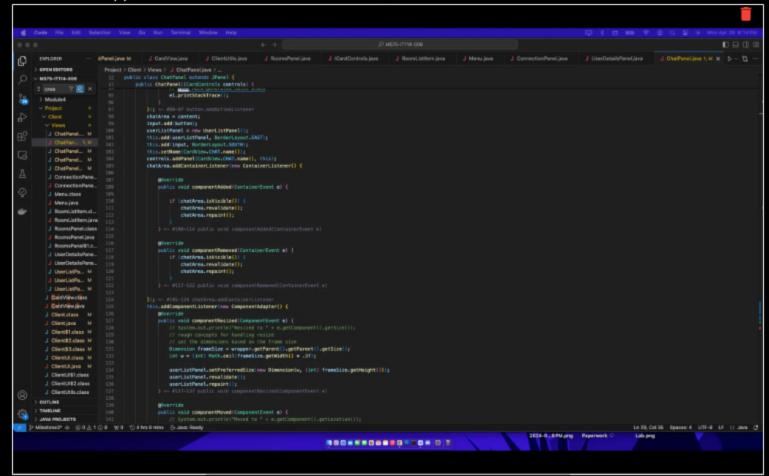
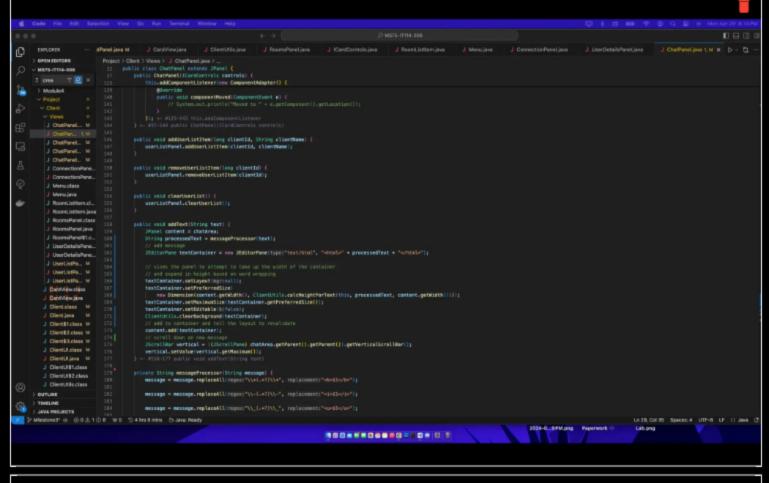### #2 User Details Panel

Chat Panel Part 1

Checklist Items (1)

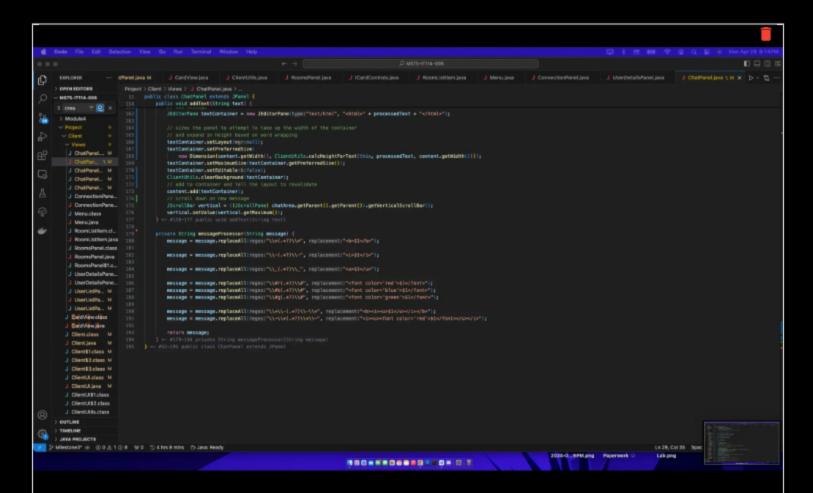### #3 Chat Panel

Chat Panel Part 2

Checklist Items (0)

Chat Panel Part 3
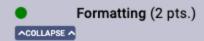
Checklist Items (0)

Checklist Items (1)

#3 Chat Panel

```java
public class ChatPanel extends JPanel {

    public ChatPanel(ICardControls controls) {
        this.addComponentListener(new ComponentAdapter() {
            @Override
            public void componentMoved(ComponentEvent e) {
                // System.out.println("Moved to " + e.getComponent().getLocation());
            }
        }); // <- #125-143 this.addComponentListener
    } // <- #37-144 public ChatPanel(ICardControls controls)

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

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

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

    public void addText(String text) {
        JPanel content = chatArea;
        String processedText = messageProcessor(text);
        // add message
        JEditorPane textContainer = new JEditorPane(type:"text/html", "<html>" + processedText + "</html>");

        // sizes the panel to attempt to take up the width of the container
        // and expand in height based on word wrapping
        textContainer.setLayout(mgr:null);
        textContainer.setPreferredSize(
            new Dimension(content.getWidth(), ClientUtils.calcHeightForText(this, processedText, content.getWidth())));
        textContainer.setMaximumSize(textContainer.getPreferredSize());
        textContainer.setEditable(b:false);
        ClientUtils.clearBackground(textContainer);
        // add to container and tell the layout to revalidate
        content.add(textContainer);
        // scroll down on new message
        JScrollBar vertical = ((JScrollPane) chatArea.getParent().getParent()).getVerticalScrollBar();
        vertical.setValue(vertical.getMaximum());
    } // <- #158-177 public void addText(String text)

    private String messageProcessor(String message) {
        message = message.replaceAll(regex:"\\*(.*?)\\*", replacement:"<b>$1</b>");

        message = message.replaceAll(regex:"\\-(.*?)\\-", replacement:"<i>$1</i>");

        message = message.replaceAll(regex:"\\_(.*?)\\_", replacement:"<u>$1</u>");
```

```java
public class ChatPanel extends JPanel {

    public void addText(String text) {

        JEditorPane textContainer = new JEditorPane(type:"text/html", "<html>" + processedText + "</html>");

        // sizes the panel to attempt to take up the width of the container
        // and expand in height based on word wrapping
        textContainer.setLayout(mgr:null);
        textContainer.setPreferredSize(
            new Dimension(content.getWidth(), ClientUtils.calcHeightForText(this, processedText, content.getWidth())));
        textContainer.setMaximumSize(textContainer.getPreferredSize());
        textContainer.setEditable(b:false);
        ClientUtils.clearBackground(textContainer);
        // add to container and tell the layout to revalidate
        content.add(textContainer);
        // scroll down on new message
        JScrollBar vertical = ((JScrollPane) chatArea.getParent().getParent()).getVerticalScrollBar();
        vertical.setValue(vertical.getMaximum());
    } // <- #158-177 public void addText(String text)

    private String messageProcessor(String message) {
        message = message.replaceAll(regex:"\\*(.*?)\\*", replacement:"<b>$1</b>");

        message = message.replaceAll(regex:"\\-(.*?)\\-", replacement:"<i>$1</i>");

        message = message.replaceAll(regex:"\\_(.*?)\\_", replacement:"<u>$1</u>");

        message = message.replaceAll(regex:"\\#r(.*?)\\#", replacement:"<font color='red'>$1</font>");
        message = message.replaceAll(regex:"\\#b(.*?)\\#", replacement:"<font color='blue'>$1</font>");
        message = message.replaceAll(regex:"\\#g(.*?)\\#", replacement:"<font color='green'>$1</font>");

        message = message.replaceAll(regex:"\\*\\-(.*?)\\-\\*", replacement:"<b><i><u>$1</u></i></b>");
        message = message.replaceAll(regex:"\\-\\*(.*?)\\*\\-", replacement:"<i><u><font color='red'>$1</font></u></i>");

        return message;
    } // <- #179-194 private String messageProcessor(String message)
} // <- #32-195 public class ChatPanel extends JPanel
```

Checklist Items (1)

#3 Chat Panel

● **Formatting (2 pts.)**
∧COLLAPSE ∧

●
∧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

Small   Medium   Large

I could not get the /flip and /roll methods to even work on Milestone3. On milestone 2 i wasnt able to get them to work either but after submitting the assignment I was able to, for some reason when i implemented all the Base UI code nflip/and /roll stopped working. I am assuming it is because of the new way to processClientCommands since we deprecated it in the Client.java class as apart of Milestone3 but I could not figure out how to get them to run.

Checklist Items (0)

🟢

COLLAPSE

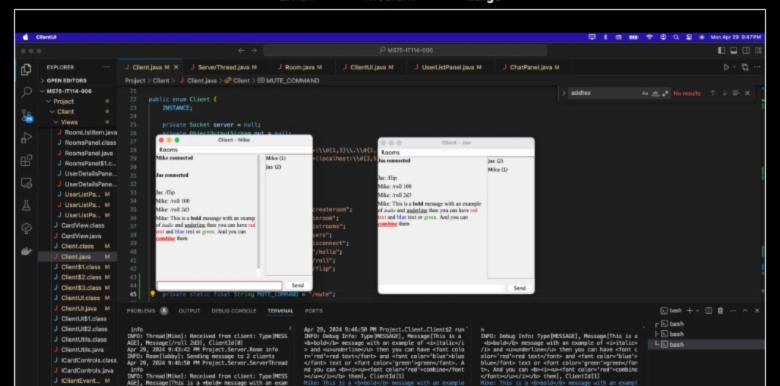## Task #2 - Points: 1

Text: Screenshots demoing custom text formatting

Checklist		*The checkboxes are for your own tracking

#	Points	Details
☐ #1	1	Custom text formatting for bold working (Part of the message should appear bold)
☐ #2	1	Custom text formatting for italic working (Part of the message should appear italic)
☐ #3	1	Custom text formatting for underline working (Part of the message should appear underline)
☐ #4	1	Custom text formatting for red working (Part of the message should appear red)
☐ #5	1	Custom text formatting for blue working (Part of the message should appear blue)
☐ #6	1	Custom text formatting for green working (Part of the message should appear green)
☐ #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)
☐ #8	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small   Medium   Large

Screenshot shows all aspects of the Custom text formatting working.

Checklist Items (0)
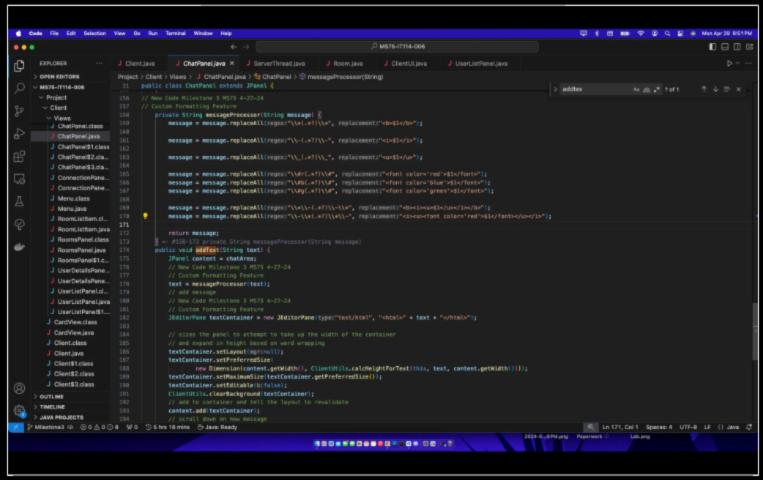
🟢

[COLLAPSE]

Task #3 - Points: 1

Text: Screenshot of the code solving the formatting display

Checklist
*The checkboxes are for your own tracking

#	Points	Details
☐ #1	1	Show each relevant file this was done in (may be one or more)
☐ #2	1	Include ucid and date comment
☐ #3	1	Clearly caption screenshots

Task Screenshots:

Gallery Style: Large View

Small   Medium   Large

The only new lines added were Lines 158 - 173, which was my messageProcessor() method that I recycled from the ServerThread.java class from milestone 2 and line 178. Line 182 was edited but stayed mostly the same as from the

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:

```
        text = messageProcessor(text);
```
the line of code above is creating a string and calling the messageProcessor() method I recycled from my Milestone2, and sends the string text in the addText() method into the messageProcessor() method.

The message processor as shown in the screenshot replaces special characters such as *-#_ with the correspondinf HTML tags.

```
        JEditorPane textContainer = new JEditorPane("text/html", "<html>" + text + "</html>");
```

The line of code above sends the text string into a textContainer and prints it as /html text rather than just /plain text as it was before.

● **Private Message with @ (2 pts.)**
^COLLAPSE ^

●
∨ EXPAND ∨
Task #1 - Points: 1

Text: Screenshots demoing private message
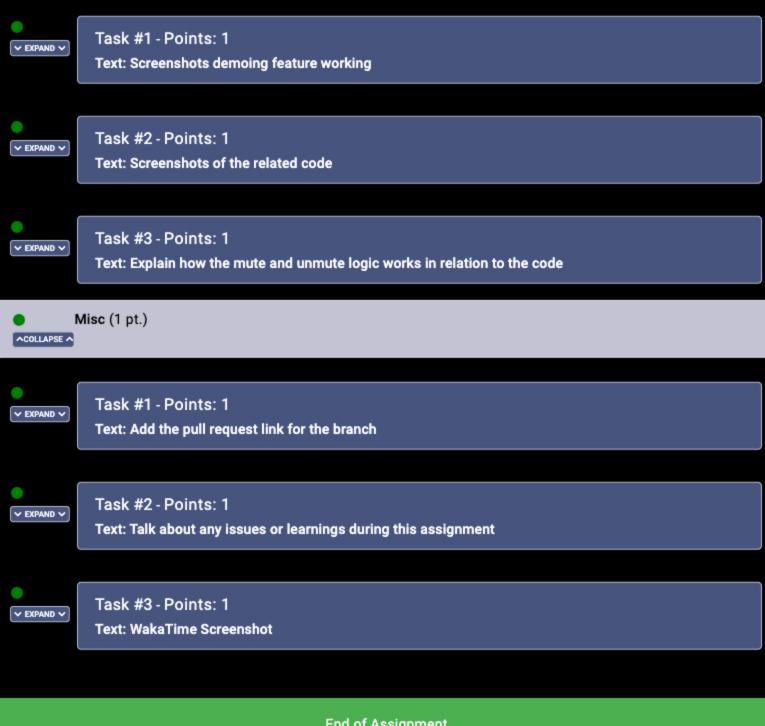
●
∨ EXPAND ∨
Task #2 - Points: 1

Text: Screenshots of the related code

●
∨ EXPAND ∨
Task #3 - Points: 1

Text: Explain how private message works related to the code above

● **Mute/Unmute Users (3 pts.)**
^COLLAPSE ^

Task #1 - Points: 1

Text: Screenshots demoing feature working

EXPAND

Task #2 - Points: 1

Text: Screenshots of the related code

EXPAND

Task #3 - Points: 1

Text: Explain how the mute and unmute logic works in relation to the code

EXPAND

Misc (1 pt.)

COLLAPSE

Task #1 - Points: 1

Text: Add the pull request link for the branch

EXPAND

Task #2 - Points: 1

Text: Talk about any issues or learnings during this assignment

EXPAND

Task #3 - Points: 1

Text: WakaTime Screenshot

EXPAND

End of Assignment