

Submission Worksheet

CLICK TO GRADE

<https://learn.ethereallab.app/assignment/IT114-006-S2024/it114-milestone-2-chatroom-2024/grade/bm47>

IT114-006-S2024 - [IT114] Milestone 2 Chatroom 2024

Submissions:

Submission Selection

1 Submission [active] 4/3/2024 7:23:33 PM

Instructions

^ COLLAPSE ^

Implement the Milestone 2 features from the project's proposal document:

<https://docs.google.com/document/d/1ONmvEvel97GTFPGfVwwQC96xSsobbSbk56145XizQG4/view>

Make sure you add your ucid/date as code comments where code changes are done

All code changes should reach the Milestone2 branch

Create a pull request from Milestone2 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: Milestone2

Tasks: 12 Points: 10.00



Demonstrate Usage of Payloads (2 pts.)

^ COLLAPSE ^



Task #1 - Points: 1

Text: Screenshots of your Payload class and subclasses and PayloadType

Checklist

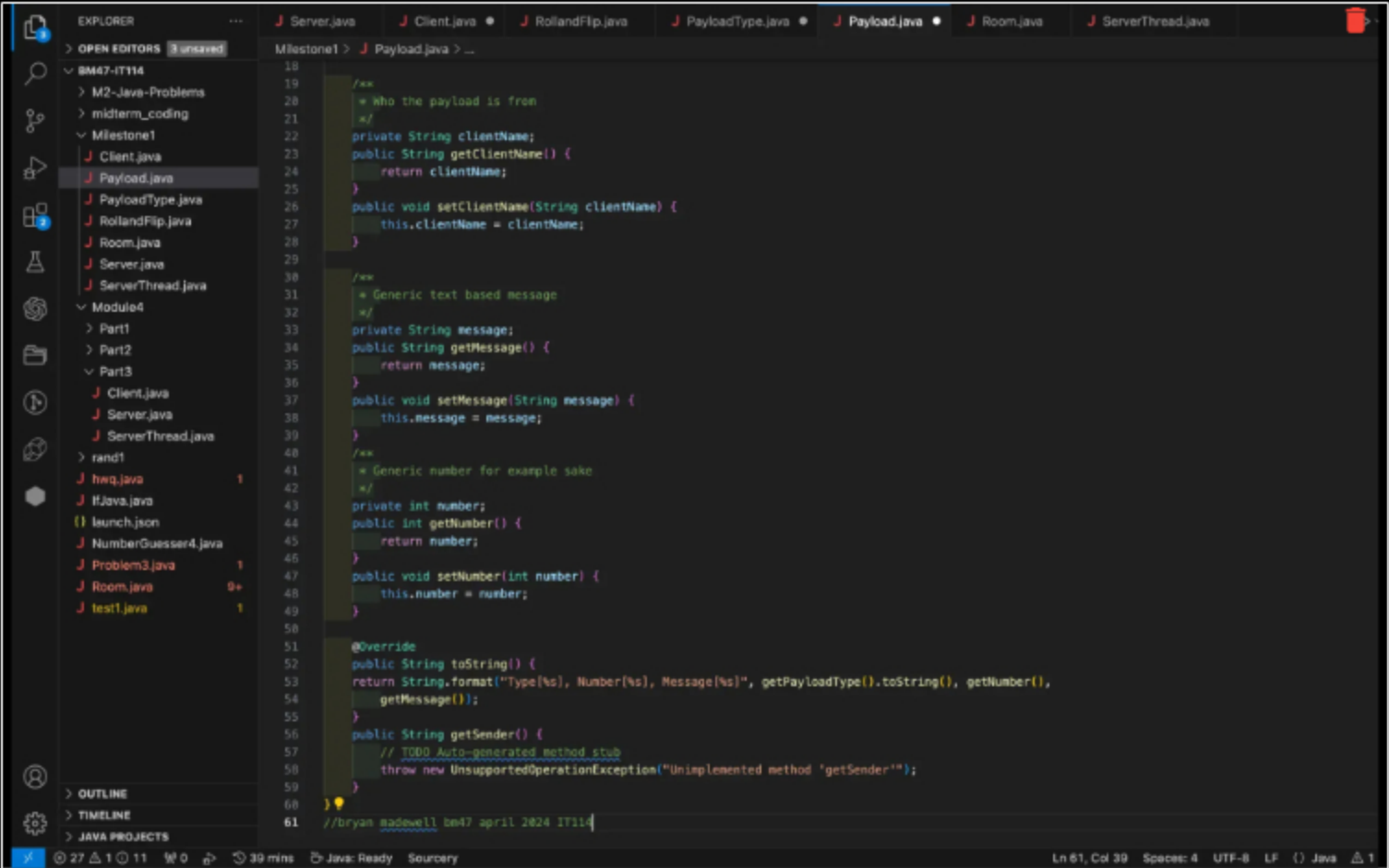
*The checkboxes are for your own tracking

#	Points	Details
#1	1	Payload, equivalent of RollPayload, and any others
#2	1	Screenshots should include ucid and date comment
#3	1	Each screenshot should be clearly captioned

Task Screenshots:

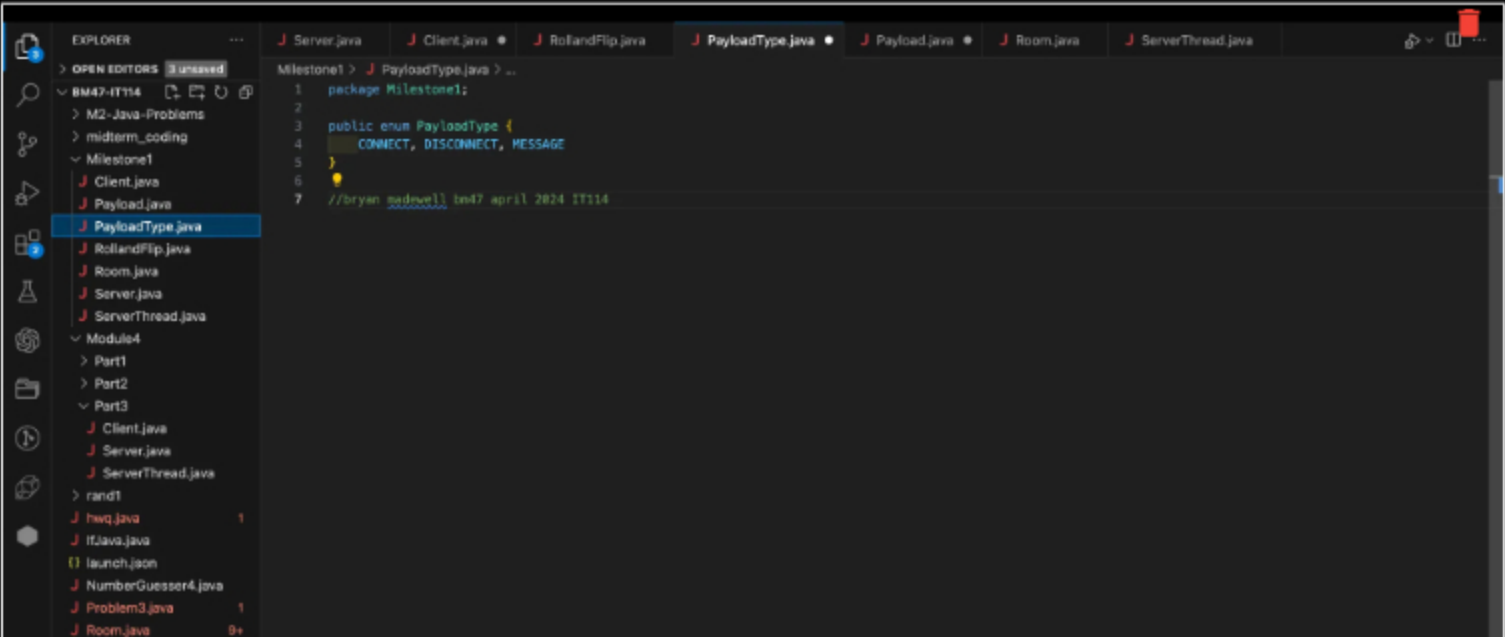
Gallery Style: Large View

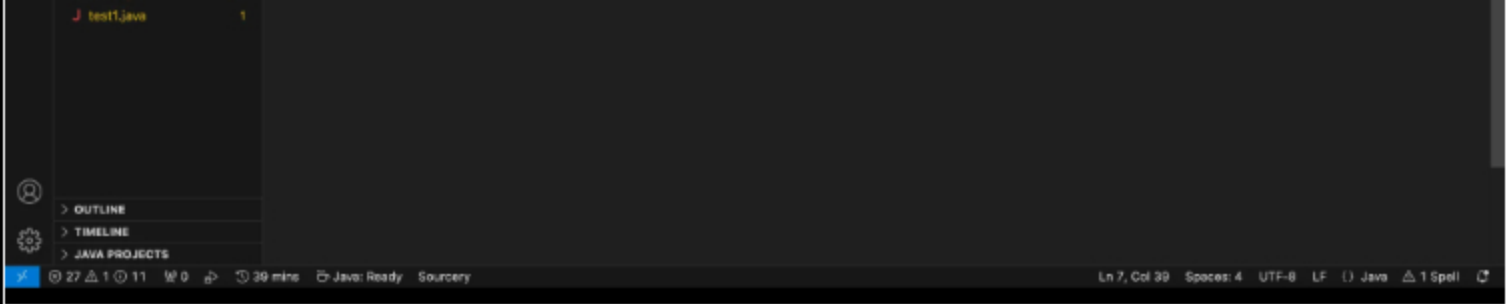
Small Medium Large



payload.java

Checklist Items (0)





payloadtype.java

Checklist Items (0)

Task #2 - Points: 1

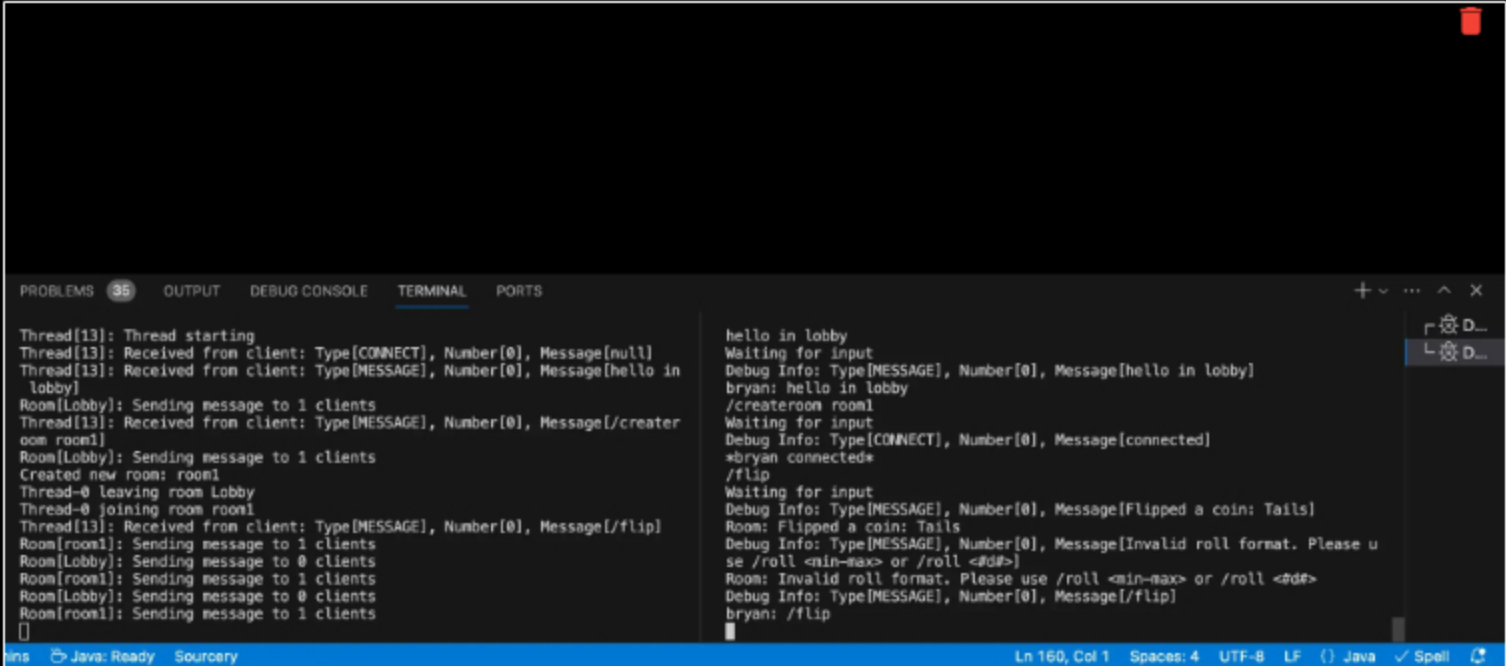
Text: Screenshots of the payloads being debugged/output to the terminal

Checklist			*The checkboxes are for your own tracking
#	Points	Details	
<input type="checkbox"/> #1	1	Demonstrate flip	
<input type="checkbox"/> #2	1	Demonstrate roll (both versions)	
<input type="checkbox"/> #3	1	Demonstrate formatted message along with any others	
<input type="checkbox"/> #4	1	Each screenshot should be clearly captioned	

Task Screenshots:

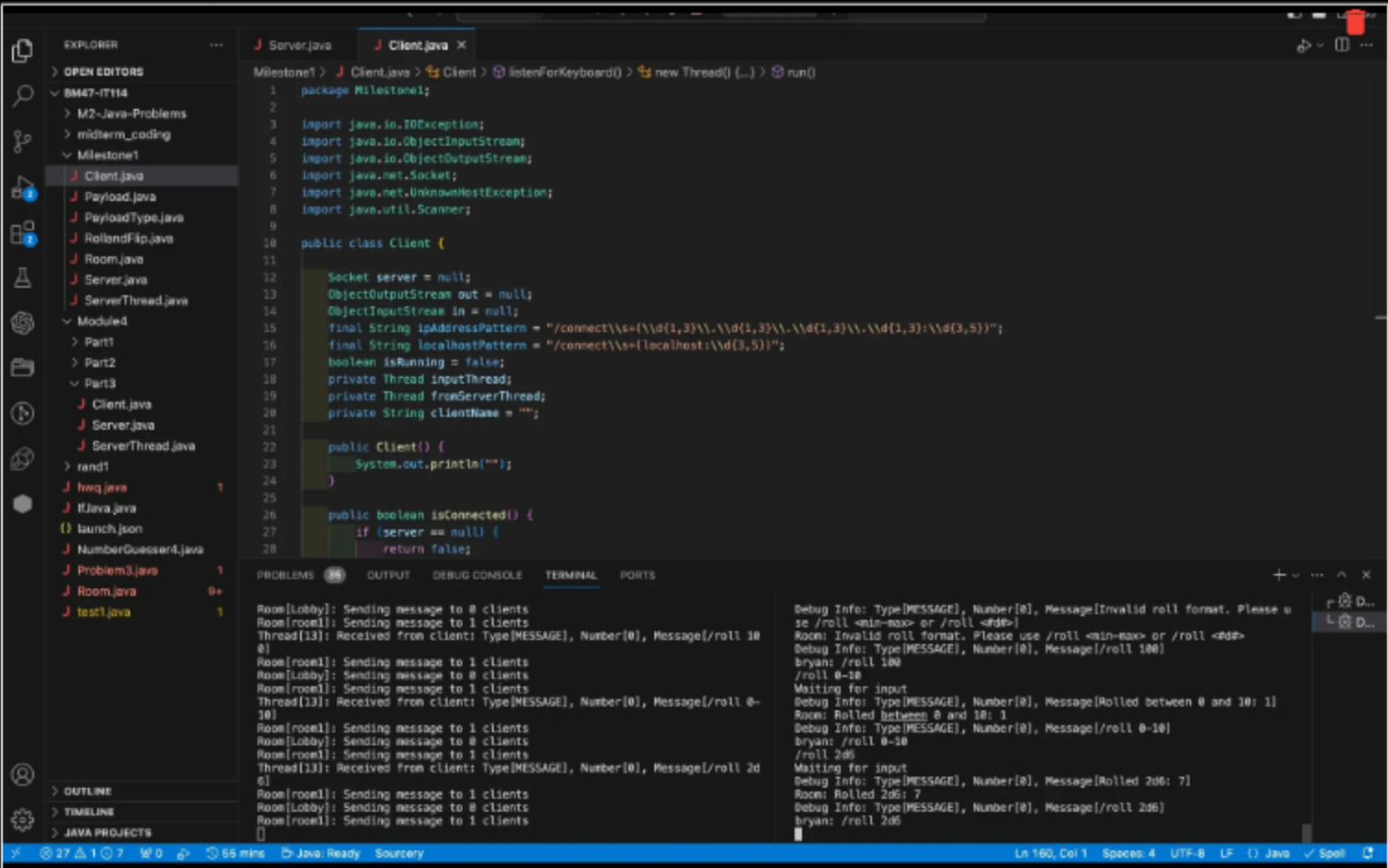
Gallery Style: Large View

Small Medium Large



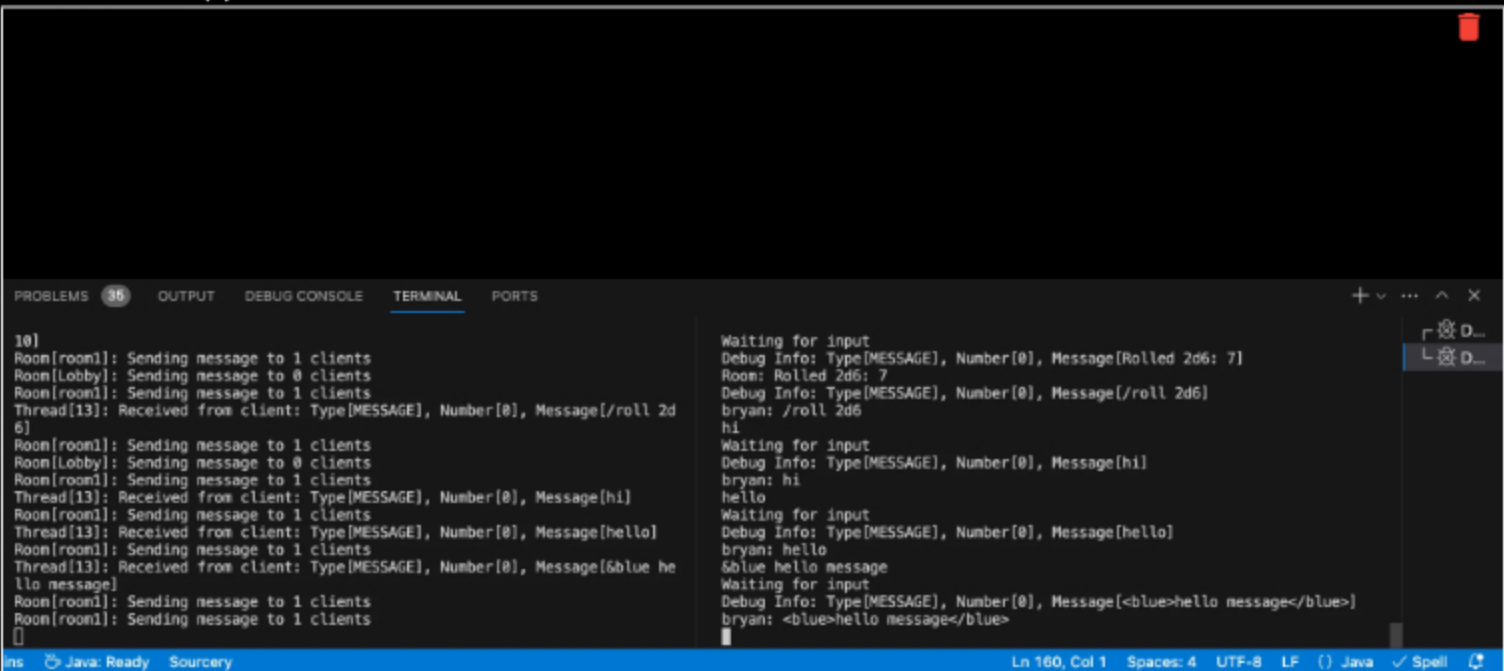
flip is shown working in the terminal. after executing the /flip command, the terminal says "debug Info: Type[MESSAGE], Number[0], Message[Flipped a coin: Tails] Room: Flipped a coin: Tails"

Checklist Items (0)



the terminal shows the /roll 0-10 and /roll 2d6 working properly after the commands are executed.

Checklist Items (0)



clearly shows the text formatting working. when i type with &blue, it shows that the text is wrapped in HTML.

Checklist Items (0)

Task #3 - Points: 1

Text: Explain the purpose of payloads and how your flip/roll payloads were made

Response:

Payloads allow you to add more information to a client/server request/response. This information is like metadata that gives the client code information that can be used to do things other than just display the message. The flip/roll payloads use a command and message portion to inform the client disconnected if it was just the message sent.

Demonstrate Roll Command (2 pts.)

Task #1 - Points: 1

Text: Screenshot of the following items

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input checked="" type="checkbox"/> #1	1	Client code that captures the command and converts it to a RollPayload (or equivalent) for both scenarios /roll # and /roll #d#
<input type="checkbox"/> #2	1	ServerThread code receiving the payload and passing it to the Room
<input checked="" type="checkbox"/> #3	1	Room handling the roll action correctly for both scenarios (/roll # and /roll #d#) including the message going back out to all clients
<input checked="" type="checkbox"/> #4	1	Code screenshots should include ucid and date comment
<input checked="" type="checkbox"/> #5	1	Each screenshot should be clearly captioned

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

```
J Server.java J Client.java • J Payload.java J RollandFlip.java J Room.java •
Milestone1 > J Client.java > Client > processMessage(Payload)
210         } finally {
211             close();
212             System.out.println("Stopped listening to server input");
213         }
214     }
215 }
```

shows payload information

The screenshot shows an IDE terminal window with two panes. The top bar of the terminal has tabs for PROBLEMS (40), OUTPUT, DEBUG CONSOLE, TERMINAL (selected), and PORTS. The left pane displays the application's output, showing messages from 'Room' and 'Thread' objects. The right pane displays the debug console, showing the same messages along with user input and debug information. The status bar at the bottom indicates 'hr 14 mins', 'Java: Ready', 'Sourcery', 'Ln 221, Col 72', 'Spaces: 4', 'UTF-8', 'LF', '() Java', and '1 Spell'.

```
Room[room1]: Sending message to 1 clients
Thread[13]: Received from client: Type[MESSAGE], Number[0], Message[/flip]
Room[room1]: Sending message to 1 clients
Room[Lobby]: Sending message to 0 clients
Room[room1]: Sending message to 1 clients
Room[Lobby]: Sending message to 0 clients
Room[room1]: Sending message to 1 clients
Thread[13]: Received from client: Type[MESSAGE], Number[0], Message[/roll 0-10]
Room[room1]: Sending message to 1 clients
Room[Lobby]: Sending message to 0 clients
Room[room1]: Sending message to 1 clients
Thread[13]: Received from client: Type[MESSAGE], Number[0], Message[/roll 2d 6]
Room[room1]: Sending message to 1 clients
Room[Lobby]: Sending message to 0 clients
Room[room1]: Sending message to 1 clients

Debug Info: Type[MESSAGE], Number[0], Message[Invalid roll format. Please use /roll <min-max> or /roll <#d#>]
Room: Invalid roll format. Please use /roll <min-max> or /roll <#d#>
Debug Info: Type[MESSAGE], Number[0], Message[/flip]
bryan: /flip
/roll 0-10
Waiting for input
Debug Info: Type[MESSAGE], Number[0], Message[Rolled between 0 and 10: 9]
Room: Rolled between 0 and 10: 9
Debug Info: Type[MESSAGE], Number[0], Message[/roll 0-10]
bryan: /roll 0-10
/roll 2d6
Waiting for input
Debug Info: Type[MESSAGE], Number[0], Message[Rolled 2d6: 7]
Room: Rolled 2d6: 7
Debug Info: Type[MESSAGE], Number[0], Message[/roll 2d6]
bryan: /roll 2d6
```

terminal showing both /roll 0-10 and /roll2d6 working correctly

Text: Explain the logic in how the two different roll formats are handled and how the message flows from the client, to the Room, and shared with all other users

Response:

The roll formats have their own handlers that wait for what command is coming in and checks based on that using the Payload format.

Demonstrate Flip Command (1 pt.)

^COLLAPSE ^

Task #1 - Points: 1

^COLLAPSE ^

Text: Screenshot of the following items

Checklist			*The checkboxes are for your own tracking
#	Points	Details	
<input checked="" type="checkbox"/> #1	1	Client code that captures the command and converts it to a payload	
<input type="checkbox"/> #2	1	ServerThread receiving the payload and passing it to the Room	
<input checked="" type="checkbox"/> #3	1	Room handling the flip action correctly	
<input checked="" type="checkbox"/> #4	1	Code screenshots should include ucid and date comment	
<input checked="" type="checkbox"/> #5	1	Each screenshot should be clearly captioned	

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

```
J Server.java J Client.java J Payload.java J RollandFlip.java J Room.java
Milestone1 > J Client.java > Client > processMessage(Payload)
218 //bryan madewell bm47 IT114 april 2024
219 private void processMessage(Payload p) {
220     switch (p.getPayloadType()) {
221         case CONNECT:// for now connect,disconnect are all the same
222         case DISCONNECT:
223             System.out.println(String.format("%s %s",
224                 p.getClientName(),
225                 p.getMessage()));
226             break;
227         case MESSAGE:
228             System.out.println(String.format("%s: %s",
229                 p.getClientName(),
230                 p.getMessage()));
231             break;
232         default:
233             break;
234     }
235 }
236
237
238 public void start() throws IOException {
239     listenForKeyboard();
240 }
241
242 private void close() {
243     try {
244         inputThread.interrupt();
```

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS + v ... ^ _
```

```
llo this is bold]
Room[room1]: Sending message to 1 clients
Room[room1]: Sending message to 1 clients
Thread[13]: Received from client: Type[MESSAGE], Number[0], Message[<b>hello &blue hello</b>
llo &blue hello]
Room[room1]: Sending message to 1 clients
Room[room1]: Sending message to 1 clients
Thread[13]: Received from client: Type[MESSAGE], Number[0], Message[<b>hello hi &underline hi &italics hi]
Room[room1]: Sending message to 1 clients
Room[room1]: Sending message to 1 clients
Thread[13]: Received from client: Type[MESSAGE], Number[0], Message[/flip]
Room[room1]: Sending message to 1 clients
Room[Lobby]: Sending message to 0 clients
Room[room1]: Sending message to 1 clients
Room[Lobby]: Sending message to 0 clients
Room[room1]: Sending message to 1 clients
```

```
Debug Info: Type[MESSAGE], Number[0], Message[<b>hello &blue hello</b>
>]
bryan: <b>hello &blue hello</b>
&bld hi &underline hi &italics hi
Waiting for input
Debug Info: Type[MESSAGE], Number[0], Message[<b>hi &underline hi &italics hi</b>
cs hi</b>-]
bryan: <b>hi &underline hi &italics hi</b>
/flip
Waiting for input
Debug Info: Type[MESSAGE], Number[0], Message[Flipped a coin: Tails]
Room: Flipped a coin: Tails
Debug Info: Type[MESSAGE], Number[0], Message[Invalid roll format. Please u
se /roll <min-max> or /roll <#d#>]
Room: Invalid roll format. Please use /roll <min-max> or /roll <#d#>
Debug Info: Type[MESSAGE], Number[0], Message[/flip]
bryan: /flip
```

terminal that shows flip working correctly

Text: Screenshot of Room how the following formatting is processed from a message

Details:

Note: this processing is server-side

Slash commands are not valid solutions for this and will receive 0 credit

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Room code processing for bold
<input type="checkbox"/> #2	1	Room code processing for italic
<input type="checkbox"/> #3	1	Room code processing for underline
<input type="checkbox"/> #4	1	Room code processing for color (at least R, G, B or support for hex codes)
<input type="checkbox"/> #5	1	Show each one working individually and one showing a combination of all of the formats and 1 color from the terminal
<input type="checkbox"/> #6	1	Must not rely on the user typing html characters, but the output can be html characters
<input type="checkbox"/> #7	1	Code screenshots should include uuid and date comment
<input type="checkbox"/> #8	1	Each screenshot should be clearly captioned

Task Screenshots:

Gallery Style: Large View

Small

Medium

Large

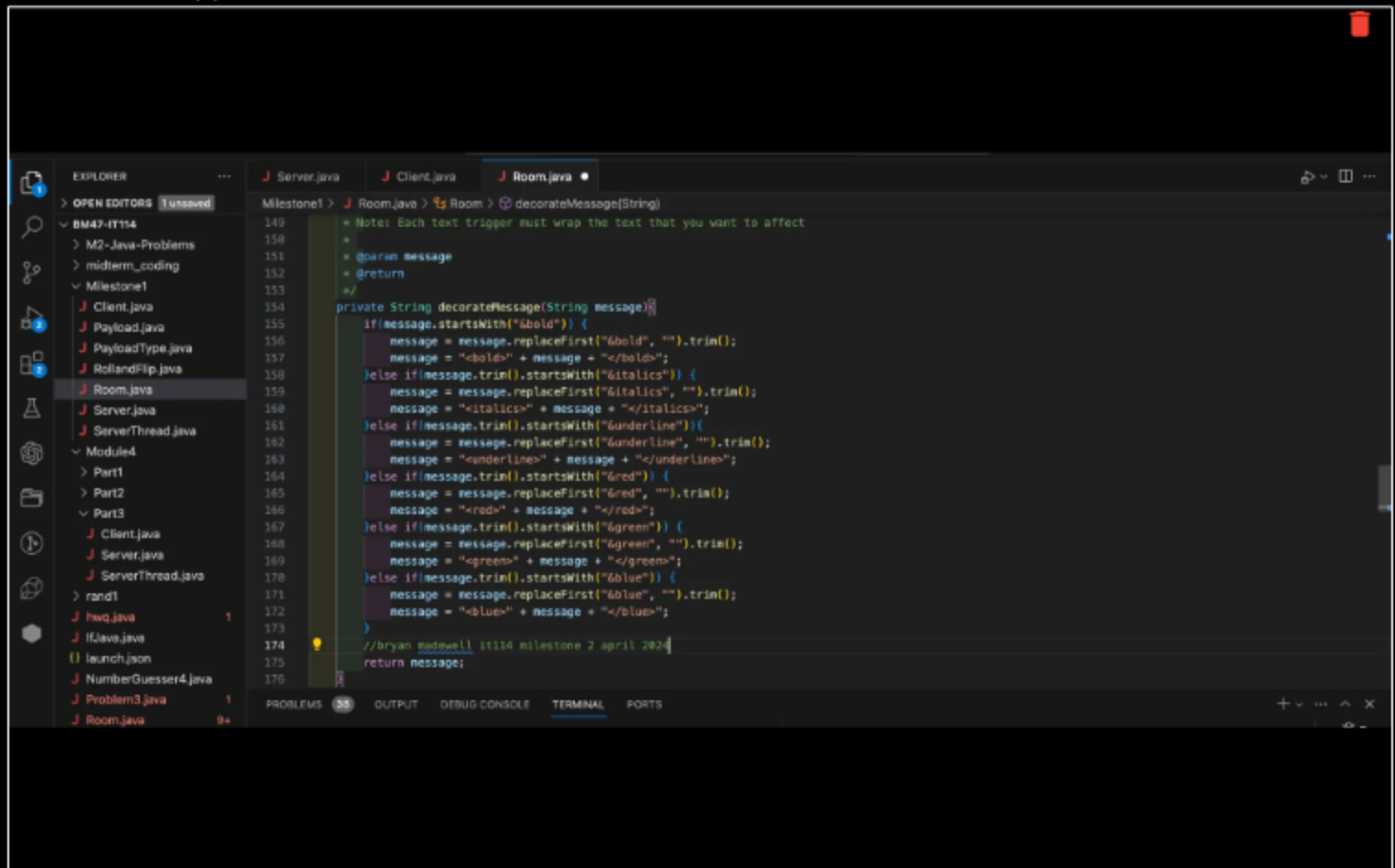
```
Thread[13]: Received from client: Type[MESSAGE], Number[0], Message[/roll 2d 6]
Room[room1]: Sending message to 1 clients
Room[Lobby]: Sending message to 0 clients
Room[room1]: Sending message to 1 clients
Thread[13]: Received from client: Type[MESSAGE], Number[0], Message[hi]
Room[room1]: Sending message to 1 clients
Thread[13]: Received from client: Type[MESSAGE], Number[0], Message[hello]
Room[room1]: Sending message to 1 clients
Thread[13]: Received from client: Type[MESSAGE], Number[0], Message[&blue hello message]
Room[room1]: Sending message to 1 clients
Room[room1]: Sending message to 1 clients
Thread[13]: Received from client: Type[MESSAGE], Number[0], Message[&bold hello this is bold]
Room[room1]: Sending message to 1 clients
Room[room1]: Sending message to 1 clients

hi
Waiting for input
Debug Info: Type[MESSAGE], Number[0], Message[hi]
bryan: hi
hello
Waiting for input
Debug Info: Type[MESSAGE], Number[0], Message[hello]
bryan: hello
&blue hello message
Waiting for input
Debug Info: Type[MESSAGE], Number[0], Message[<blue>hello message</blue>]
bryan: <blue>hello message</blue>
&bold hello this is bold
Waiting for input
Debug Info: Type[MESSAGE], Number[0], Message[<bold>hello this is bold</bold>]
bryan: <bold>hello this is bold</bold>
```

this shows the terminal output for when bold text is used

this shows the terminal output for when bold text is used.

Checklist Items (0)

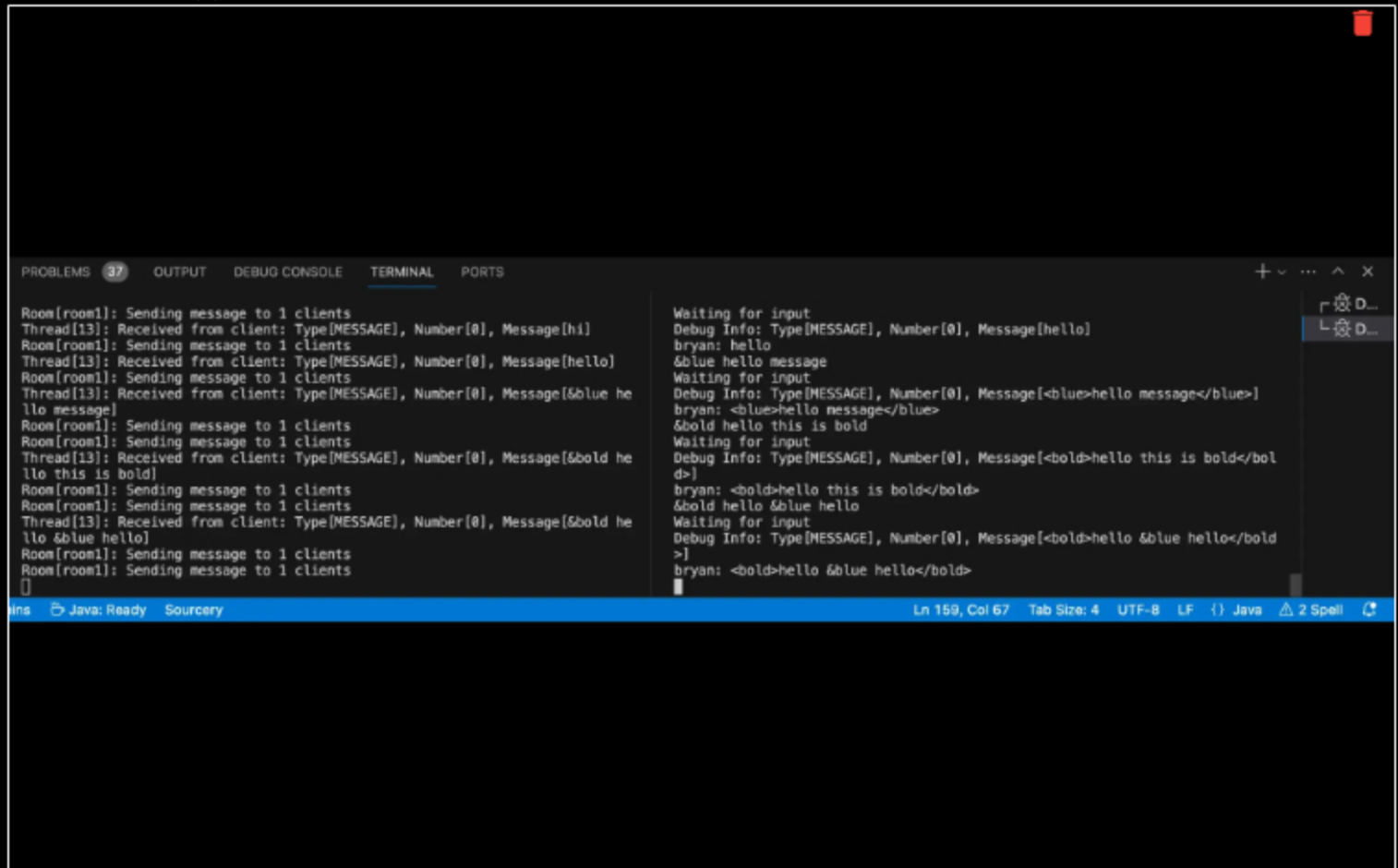


The screenshot shows an IDE with the Explorer view on the left and the Editor view on the right. The Explorer view shows a project structure with folders like 'Milestone1' and 'Room.java'. The Editor view shows the code for 'Room.java' with a method 'decorateMessage(String message)' that formats text using HTML tags. The code is as follows:

```
149 // Note: Each text trigger must wrap the text that you want to affect
150 //
151 // @param message
152 // @return
153 //
154 private String decorateMessage(String message) {
155     if (message.startsWith("<b>")) {
156         message = message.replaceFirst("<b>", "").trim();
157         message = "<b>" + message + "</b>";
158     } else if (message.trim().startsWith("<i>")) {
159         message = message.replaceFirst("<i>", "").trim();
160         message = "<i>" + message + "</i>";
161     } else if (message.trim().startsWith("<u>")) {
162         message = message.replaceFirst("<u>", "").trim();
163         message = "<u>" + message + "</u>";
164     } else if (message.trim().startsWith("<red>")) {
165         message = message.replaceFirst("<red>", "").trim();
166         message = "<red>" + message + "</red>";
167     } else if (message.trim().startsWith("<green>")) {
168         message = message.replaceFirst("<green>", "").trim();
169         message = "<green>" + message + "</green>";
170     } else if (message.trim().startsWith("<blue>")) {
171         message = message.replaceFirst("<blue>", "").trim();
172         message = "<blue>" + message + "</blue>";
173     }
174     //bryan badwell it114 milestone 2 april 2024
175     return message;
176 }
```

room.java code for bold, italics, underline, red, green, and blue.

Checklist Items (0)



The screenshot shows the IDE with the 'TERMINAL' view selected. The terminal output shows the execution of the code, including the formatting of text using HTML tags. The output is as follows:

```
Room[room1]: Sending message to 1 clients
Thread[13]: Received from client: Type[MESSAGE], Number[0], Message[hi]
Room[room1]: Sending message to 1 clients
Thread[13]: Received from client: Type[MESSAGE], Number[0], Message[hello]
Room[room1]: Sending message to 1 clients
Thread[13]: Received from client: Type[MESSAGE], Number[0], Message[<blue> he
llo message]
Room[room1]: Sending message to 1 clients
Room[room1]: Sending message to 1 clients
Thread[13]: Received from client: Type[MESSAGE], Number[0], Message[<b>old he
llo this is bold]
Room[room1]: Sending message to 1 clients
Room[room1]: Sending message to 1 clients
Thread[13]: Received from client: Type[MESSAGE], Number[0], Message[<b>old he
llo <blue> hello]
Room[room1]: Sending message to 1 clients
Room[room1]: Sending message to 1 clients
```

The output shows the text being formatted with HTML tags, such as old and <blue>hello. The terminal also shows the output of the code, including the formatting of text using HTML tags.

showing bold and blue being used in the same text message

Checklist Items (0)



^COLLAPSE ^

Task #2 - Points: 1

Text: Explain the following

Checklist

*The checkboxes are for your own tracking

#	Points	Details
<input type="checkbox"/> #1	1	Which special characters translate to the desired effect
<input type="checkbox"/> #2	1	How the logic works that converts the message to its final format

Response:

The special characters that translate to a special text effect is the & key. There is code that splits the message, and every time there is a & key, it checks for the word after it. Once it is either bold, italic, or underline, it will give it the desired effect using HTML tags.



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

Missing URL



^COLLAPSE ^

Task #2 - Points: 1

Text: Talk about any issues or learnings during this assignment

Response:

As the chatroom project gets more complicated, I begin running into more issues. I had some issues with payloads, getting the /flip and /roll commands to work, along with the chat effect like bold, italics, underline, and the colors to work properly. Eventually, I ended up figuring out some solutions that seem to work properly thus far. This milestone definitely helped me understand how the Java programming language works much more in depth than ever before.

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

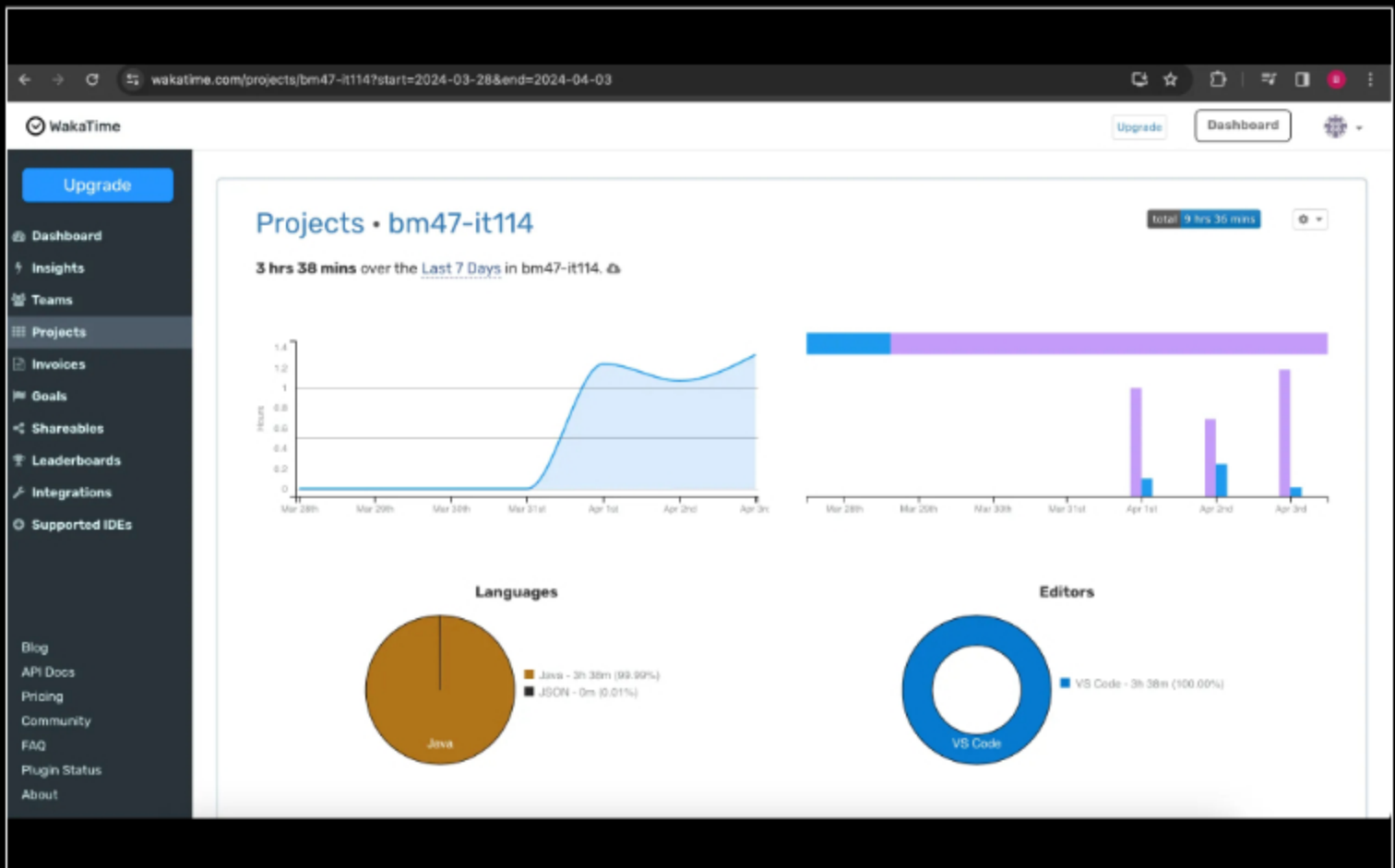
Task Screenshots:

Gallery Style: Large View

Small

Medium

Large



WakaTime shows 3hrs 38mins over the last 7 days.

End of Assignment