

Change request log #1

1. Team

Derik Flores and Vedang Patel

2. Change Request

Change Request #1: modify the status bar to track the word offset of the caret from the beginning of the file, as well as the number of total words in the file.

3. Concept Location

Step #	Description	Rationale
1	We compiled the program	
2	We took a look at the classes and class structure and looked at some README.txt	To familiarize ourselves with the class structure and naming system, we looked at subfolders and classes
3	We opened the folder named GUI	The tracker was a GUI element, so we looked in the GUI subfolder
4	Took a look in SelectionLengthWidget.java	We selected this class because it was named selection length, and upon inspection seemed to update the number that the caret is located at.
5	SelectionLengthWidget.java was incorrect, but found another java file called StatusBar.Java	After realizing SelectionLength was not the right file, we looked into StatusBar since that's where the caret data is displayed
6	We found a function called UpdateCaretStatus	The function was aptly named so we looked into it.
7	We found a variable called buf	A variable called buf seemed to control the display function. We tested its output by replacing it with "Hello", and it output "Hello", and confirmed that the function had to be modified.
8	We marked the function UpdateCaretStatus as "located"	Confirmed that the function had to be modified
9		

Time spent (in minutes): 61

4. Impact Analysis

Step #	Description	Rationale
1	We made a list of functions that called updateCaretStatus()	To track the classes that could be impacted by the change.
2	We inspected the class function propertiesChanged	This class called updateCaretStatus, but did not handle the printing of updateCaretStatus to the status bar, since this was the only function that called updateCaretStatus, no functions other than updateCaretStatus needed to be changed.

3		
4		

Time spent (in minutes): 10

5. Actualization

Use the table below to describe each step you followed when changing the code. Include as many details as possible, including why classes/methods were modified, added, removed, renamed, etc.

Step #	Description	Rationale
1	We added onto the updateCaretStatus method	Since the tracking of the total words and current word played well into the function's purposes, we decided that we did not need to build a new class or function for our changes. The buf variable was in charge of storing text to output, so we only needed to modify that.
2	Added Total Word Tracking	To implement total word tracking, we used existing variables and adapted them. In this case, the existing textArea variable could return each line independently of each other. By counting each line and then each word separated by a space in the line, we can count how many words there are in total. Then, we simply append this to our buf variable.
3	Added Current Word Tracking	To implement current word tracking, we have different actions depending on how many words there are on the line, and track the caret's column position with the "dot" variable. We compare the column position with the length of each word to determine the word on the line we're on, then we add the number of words on previous lines to obtain the current word
4	Tested the word counter	Verified the code by testing the word counter

Time spent (in minutes): 126

6. Validation

Step #	Description	Rationale
1	Test case defined: Check each word position on line 1 Inputs: 13 total words, 4 words on the first line Expected output: 1 for word 1, 2 for word 2, etc	The test passed, numbering each word on line 1 correctly, and also numbering the total words correctly
2	Test case defined: Check each word position on a line that only has 1 word Inputs: 13 total words, 1 word on a given line Expected output: The proper word position	The test passed, tracking the current and total words correctly

3	Test case defined: Test current word count with empty lines Inputs: 13 total words, 0 word on a given line Expected output: The proper total word count and position	The test failed, and counted empty lines towards current and total word count.
4		

Time spent (in minutes): 20

7. Timing

Summarize the time spent on each phase.

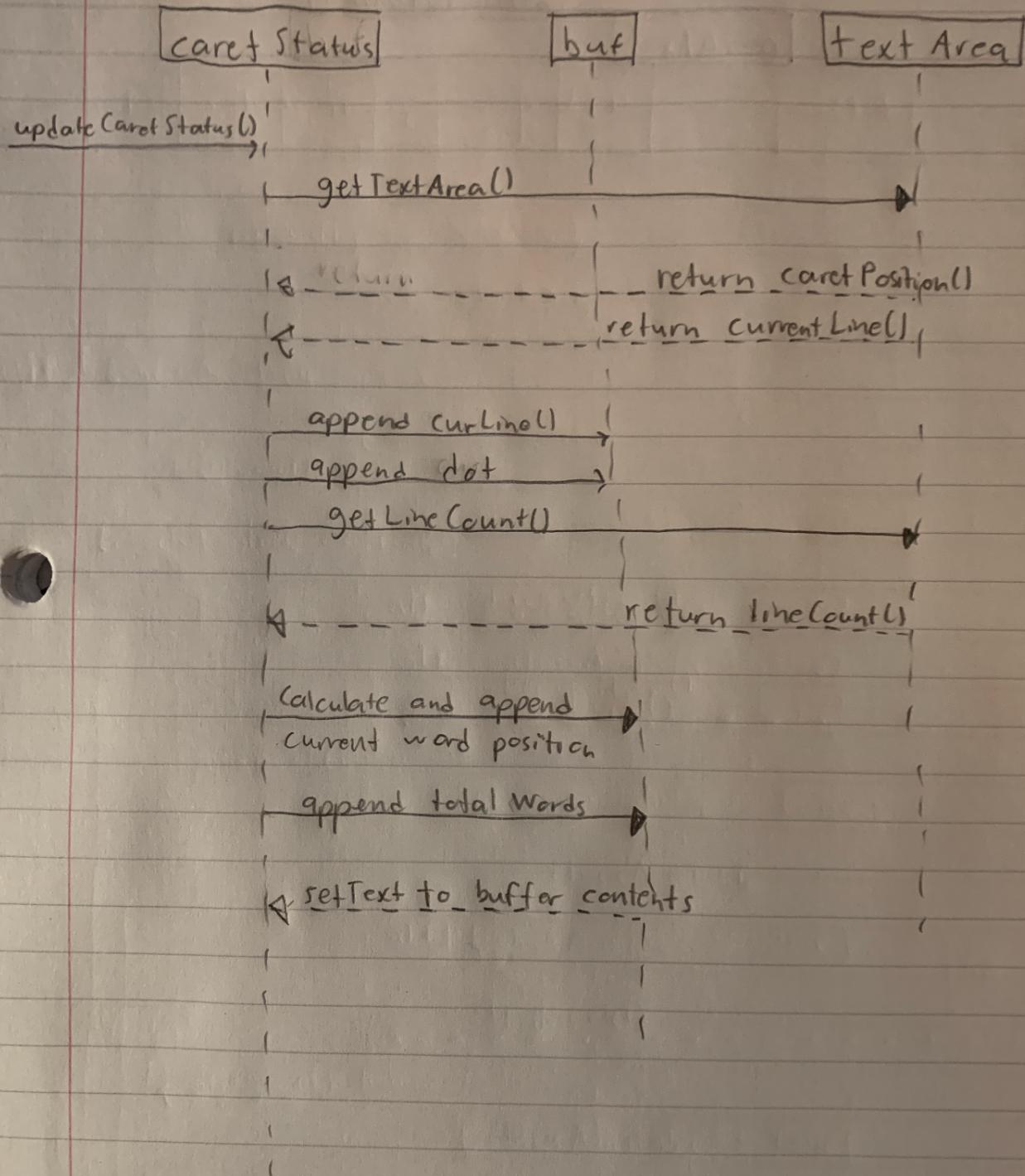
Phase Name	Time (in minutes)
Concept location	61
Impact Analysis	10
Actualization	126
Verification	20
Total	217

8. Reverse engineering

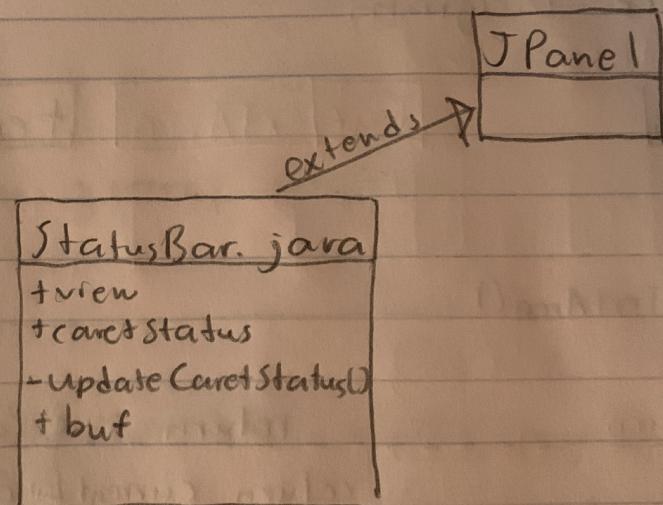
Create a UML sequence diagram (or more if needed) corresponding to the main object interactions affected by your change.

Create a partial UML class diagram of the classes visited while navigating through the code. Include the associations between classes (e.g., inheritance, aggregations, compositions, etc.), as well as the important fields and methods of each class that you learn about. The diagram may have disconnected components. Use the UML tool of your preference. When a significant fact about a class or method is learned, indicate it via annotations on the diagram. **For each change request, start with the diagram produced in the previous change request. For the first, you will start from scratch.**

UML Sequence Diagram



UML Class diagram



9. Conclusions

For this change, concept location was a little tedious because we had to familiarize ourselves with the program's structure, but when we got more familiar with the directories, it became easier. Another challenge before we could start concept location was setting up the project using ant. IntelliJ made this easier due to having ant previously installed, and automatically setting ant up for us. Impact analysis was simple, as the changes only affected one method in one class that handled output, as we were mostly reading data rather than writing to any pre-existing system. The toughest part of impact analysis was finding which named variables stored which kind of data, but since the output was formatted using character literals, it helped us identify which part of code to look at. The actualization portion was a lot of fun to do, piecing together variables that helped identify text, tracked the lines, and tracked the caret.

Classes and methods changed:

- org/gjt/sp/jedit/gui/statusbar/StatusBar.java
 - void updateCaretStatus()