Change request log

1. Concept Location

Step #	Description	Rationale
1	We compiled the code and ran the system.	
2	We interacted with system : creating files, exploring different functions.	To get familiar with some of the features of the system, and identify the screens or graphical elements we had to change.
3	We searched for “Caret Position” term using the search tool of IDE across the whole codebase.	Because we identified that we will require value of caret position to correctly make the change.
4	From total of 73 results across 26 files, we selected StatusBar class.	We selected this class because while looking into the results, one of the comments in this class mentioned about displaying the caret position.
5	We inspected the class StatusBar, and searched for term “caret” using find tool to find matches inside the file.	We noticed one comment which mentioned that updateCaretStatus() method updates the status bar with information about the caret position, line number, etc.
6	We inspected the updateCaretStatus() method of StatusBar class.	To understand more about the method’s working.
7	We debugged this method by putting debugger point at updateCaretStatus() method.	To decide whether this is the Scorrect method or not which updates the StatusBar with correct caret position, and we eventually find out this is the method responsible to update the caret position using debugger tool of IDE.
9	We marked the class StatusBar as "located".	We confirmed this class had to be modified.

Time spent (in minutes): 41

Classes and methods inspected:

- \org\gjt\sp\jedit\gui\StatusBar.java
  - void propertiesChanged()
  - void updateCaretStatus()

2. Impact Analysis

Step #	Description	Rationale
1	We made a list of classes can be impacted by the change in StatusBar Class, to find out the list of classes found out the exact matches with term “StatusBar”.	To track the classes that could be impacted by the change. Searched the term “StatusBar” with case matching to get the idea, which classes are getting the instances of this class.
2	We inspected the class EditPane. Such class was marked as "Inspected and unchanged" as well.	We realized this class is not required to make any changes, as this class is just calling the method updateCaretStatus() on buffer change, and we have only made changes which will visually impact the statusbar.
3	We inspected the class View. This class marked as "Inspected and unchanged" as well.	This class is having one method named as getStatus() which is returning the instance of statusbar, so visited other classes as well which were making calls to this method, but

		because of the visual change there was no impact.
4	We inspected HoverSetStatusMouseHandler class. This class marked as "Inspected and unchanged" as well.	We understood from the comment that this class is used for setting status with help text on hover which does not posses any relation with Caret Position.

Time spent (in minutes): 37

Classes and methods inspected:

- \org\gjt\sp\jedit\gui\StatusBar.java
  - void propertiesChanged()
  - void updateCaretStatus()
- \org\gjt\sp\jedit\gui\statusbar\HoverSetStatusMouseHandler.java
  - void mouseEntered()
  - void cleanupStatusBar()
- \org\gjt\sp\jedit\View.java
  - StatusBar getStatus()
- \org\gjt\sp\jedit\EditPane.java
  - void handleBufferUpdate()
  - void statusChanged()
  - void bracketSelected()
  - void narrowActive()

3. Actualization

Step #	Description	Rationale
1	We created the helper method countWords to to split the string into the words using the regex.	We realized to calculate the caret position we need the string that is ahead of the caret position in order to calculate the exact caret position as well as total words in the file.
2	We created two variables for wordOffset and totalWords.	We have used the buffer and getText method which was already available to get the text. For the text before the caret, we have passed the caret position to get that string only and for the content of whole document we have passed the buffer length to get the whole string and using the newly created helper method we have retrieved the two lengths and stored them into variables.
3	We have added the new properties view.status.show-word-offset and view.status.show-total-words.	We need to add the new properties view.status.show-word-offset and view.status.show-total-words to the jEdit properties file to control whether these new pieces of information are displayed or not
4	We have formatted the output in required format.	Upon checking the condition to display the information, we have framed the output in the required format by wrapping it around the brackets and separating them by the back slash
5	We performed functional testing and manual testing.	To make sure everything works.

Time spent (in minutes): 35

Inspected:

- \org\gjt\sp\jedit\gui\statusbar\HoverSetStatusMouseHandler.java

  - void mouseEntered()
  - void cleanupStatusBar()
- \org\gjt\sp\jedit\View.java
  - StatusBar getStatus()
- \org\gjt\sp\jedit\EditPane.java
  - void handleBufferUpdate()
  - void statusChanged()
  - void bracketSelected()
  - void narrowActive()

Inspected and Changed:

- \org\gjt\sp\jedit\gui\StatusBar.java
  - void propertiesChanged()
  - void updateCaretStatus()

4. Validation

Step #	Description	Rationale
1	Test case defined: The Correct value should be displayed as per required. Expected output: The correct output was there.	This is the regular expected behavior. The test passed.
2	Test case defined: Correct value should be displayed of the caret position at the very start of the file. Expected output: Correct output was there for the very start position of caret.	This is the regular expected behavior. The test passed.
3	Test case defined: Correct value should be displayed of the caret position at the very end of the file. Expected output: Correct output was there for the very end position of caret.	This is the regular expected behavior. The test passed.
4	Test case defined: Correct value should be displayed for all the files. Expected output: Correct output was there for every file.	This is the regular expected behavior. The test passed.
5	Test case defined: Count of word should not be there if we put the caret before the first character of the word. Expected output: Correct count was there as expected.	This is the regular expected behavior. The test passed.
6	Test case defined: Count of word should be there if we move the caret after the first character of the word. Expected output: Correct count was there as expected.	This is the regular expected behavior. The test passed.

Time spent (in minutes): 20

5. Summary of the change request

Phase	Time (minutes)	No. of classes inspected	No. of classes changed	No. of methods inspected	No. of methods changes
Concept location	41	1	0	2	0
Impact Analysis	37	4	0	9	0
Actualization	35	4	1	2	2
Verification	20	4	0	9	0
Total	133	13	1	22	2

6. Conclusions

The concept location for this change was the easiest part as the file for the status bar was separately created. The impact analysis was also easy as there were very few files that were being affected by this change. The Actualization was not as easy as expected because there was no previous existing method which can be used directly to serve the purpose of the change requested, we altogether have to create new method to fulfill the requirement to the change, also we have to format the output in the requested format which was not hard after we have all the required data for the display. The Verification process for this change was also a time-consuming process as we must correctly display the value and it should be correct for all the files. Overall, this change was not as easy as it appeared at first but was not too difficult.