Change request log

# Team

Jeremiah John

Steve O

# Change Request

Provide the id and description of the change request.

db3a5ca333591e57be298f11fd269cc64189e5f3

change request #1: update status bar to show word count

# Concept Location

Use the table below to describe each step you follow when performing concept location for this change request. In your description, include the following information when appropriate:

* IDE Features used (e.g., searching tool, dependency navigator, debugging, etc.)
* Queries used when searching
* System executions and input to the system
* Interactions with the system (e.g., pages visited)
* Classes visited
* The first class found to be changed (this is when concept location ends)

When there is a major decision/step in the process, include its rationale, i.e., why that decision/step was taken.

Make sure you time yourselves when going through this process and provide the total time spent below.

The following is an example of a concept location process for the change request "Color student schedule":

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | *We used IntelliJ’s search function to look for where the getCaretPosition() function was called* | *If the status bar shows the caret position, it has to get the caret position from somewhere* |
| 2 | *We searched the source code and the API for Jedit online to see how exactly the status bar* | *Only one file in statusbar folder used getCaretPosition(). It was for error handling* |
| 3 | *We found editpane.java and began to search for caret information there* | *We found info on carets when we searched through the folder containing editpane.java* |
| 4 | *Found caretInfo class and began looking for where it was used elsewhere* | *We selected this class because …* |
| 5 | *We found statusbar.java* | *We saw that the caret position was being sent to that file. Also, the comments led us to believe that this was how the status bar showed different info* |
| 6 | *We searched through jedit.java* | *The statusbar file was referencing caret info found in this file* |
| 7 | *We found updateCaretStatus() in Statusbar.java and marked it as “located”* | *This function was used to show the existing caret information formatted as is on the status bar, with parenthesis and appropriate spacing. This is where we should also include the word offset* |
| 8 |  |  |
| 9 |  |  |

**Time spent (in minutes):** 43

# Impact Analysis

Use the table below to describe each step you follow when performing impact analysis for this change request. Include as many details as possible, including why classes are visited or why they are discarded from the estimated impact set.

Do not take the impact analysis of your changes lightly. Remember that any small change in the code could lead to large changes in the behavior of the system. Follow the impact analysis process covered in the class. Describe in details how you followed this process in the change request log. Provide details on how and why you finished the impact analysis process.

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | *We compiled a list of the functions that called StatusBar and updateCaretStatus* | *To track the classes that could be impacted by the change.* |
| 2 | *We found that StatusBar, EditPane, and View were the only classes that called on updateCaretStatus* | *We realized this class had to be changed because the method render uses the StudenGraph to get the properties of the schedule. Then, the new property, i.e., color, has to be registered in the map of properties.* |
| 3 | *We concluded our changes would not drastically affect these classes* | *What were essentially doing was adding some more to a string that was returned by updateCaretStatus* |
| 4 |

**Time spent (in minutes):** 13

# 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 created two JCheckBox objects in StatusBarOptionPane: showTotalWords, showWordOffset* | *The checkbox class is what is used to set different options, like showing caret position, so we wanted to keep with the current structure* |
| 2 | *We created getWordPosition() and getTotalWords, and in Statusbar.java we created variables totalWords and wordPosition* | *This is in keeping with how the caret position and character count is retrieved and shown in the status bar* |
| 3 | *We added new if statements in updateCaretStatus to check if the user wants to show word count and current word position* | *updateCaretStatus is just a big If statement that shows what information to include in the status bar* |
| 4 | *We created unit tests for the new class and also performed functional testing. We also ran the existing test cases.* | *To make sure everything works.* |
| 5 | *...* |  |

**Time spent (in minutes):** 45

# Validation

Use the table below to describe any validation activity (e.g., testing, code inspections, etc.) you performed for this change request. Include the description of each test case, the result (pass/fail) and its rationale.

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | *Test case defined:enter the words “hello world!”*  *Inputs: “hello world!”*  *Expected output:...2/2* | *Simple test to see if it counts words* |
| 2 | *Test case defined: “the quick brown fox jumped over the lazy dog”, put cursor over fox”*  *Inputs: “the quick brown fox jumped over the lazy dog*  *Expected output:...4/9* | *Test to see if it can identify which word the caret is on* |
| 3 |  |  |
| 4 | *...* |  |

**Time spent (in minutes):** 15

# Timing

Summarize the time spent on each phase.

|  |  |
| --- | --- |
| Phase Name | Time (in minutes) |
| Concept location | 43 |
| Impact Analysis | 13 |
| Actualization | 45 |
| Verification | 15 |
| Total | 116 |

# 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.**

Diagram

Description automatically generatedDiagram

Description automatically generated

# Conclusions

Perform and analysis of the change requests and the change process. List the major challenges this change request posed.

* *We updated Statusbar.java TextArea.java, Statusbaroptionpane.java*
* *It was surpisingly difficult to make the necessary changes, because we first need to figure out how to count the number of words and which word the caret was on*
* *Then we had to store the variables in the corresponding files before we could modify the if statement in updateCaretStatus()*