Change request log

# Team

Kyle Bolin – Driver

John McGlumphy - Navigator

# Change Request

Change Request #3

# Concept Location

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | *We ran the system* |  |
| 2 | *We interacted with the system: after logging in we entered the schedule screen.* | *To get familiar with some of the features of the system, and identify the screens or graphical elements we had to change.* |
| 3 | *Search for where the options in view* | *To find where the options were located* |
| 4 | *Found where the gui buttons are located (actions.xml)* | *To help create the button* |
| 5 | *Narrowed down the TextArea.java* |  |

**Time spent (in minutes):** 100

# Impact Analysis

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | *Inspected all classes which were related to the classes we found in concept* | *To see what other classes would be affected* |
| 2 | *No other classes found would be affected* |  |

**Time spent (in minutes):** 40

# Actualization

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | *We created two new menu items in jedit\_gut.props* | *Was requested* |
| 2 | *Added a method in TextArea.java called toggleBars()* |  |
| 3 | *Added two actions in the actions.xml and textarea.actions.xml* | *Links the method to the gui* |
| 4 | *Added English localization to jedit\_en.props* | *So English would appear on button* |

**Time spent (in minutes):** 120

# Validation

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | *Manual testing* | *See if it works* |

**Time spent (in minutes):** 10

# Timing

Summarize the time spent on each phase.

|  |  |
| --- | --- |
| Phase Name | Time (in minutes) |
| Concept location | 100 |
| Impact Analysis | 40 |
| Actualization | 120 |
| Verification | 10 |
| Total | 270 |

# 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 generated

Diagram, engineering drawing

Description automatically generated

# Conclusions

*For this change, the hardest part was concept location. Finding the code that needed to be modified was the hardest part because we had no familiarity of the code, so learning it took a bit. Actualization was the easiest because the code already existed, we just needed to add it to RecentDirectoriesProvider.java .*

*Classes and methods changed:*

* *org/gjt/sp/jedit/textarea/TextArea.java*
* *void toggleBars()*
* *org/gjt/sp/jedit/textarea/textarea.actions.xml*
* *org/gjt/sp/jedit/actions.xml*
* *org/gjt/sp/jedit/jedit\_gui.props*
* *org/jedit/localization/jedit\_en.props*