Change request log

# Team

CSRAMS - Anurag Kumar and Ahmar Aftab

# Change Request

PDFsam Change Request #1: The Alternate Mix and Merge modules of PDFsam provide buttons to move a document up or down in their respective lists of files. However, these buttons are not ideal when the list of documents is long, and the user wants to move one of them to the top or the bottom. You are requested to add two new buttons in these modules, to allow the user to move a selected document to the top and bottom of the list.

*Change Owner:* Implemented and documented by Anurag Kumar

# Concept Location

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | *We build the PDFsam* |  |
| 2 | *We ran the system and add some files to the merge option.* | *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 the keywords “move”, “move up/down”, “button”, “toolbar”, “merge” using InstaSearch.* | *Because we were looking for class or methods or variables names which are related to the “Move Up” and “Move Down” buttons which are already there in the software.* |
| 4 | *We checked all the results one by one in decreasing order of the frequency of the keyword until we located the target class.* |  |
| 5 | *After going through the class MergeSelectionPane, SelectionTable, MoveType, Selection, ModuleOwnedButton, SelectionTableToolbar, etc. we finally located the target class named SelectionTableToolbar.* | *The class MoveType already had the functions for moving the files to both the top and bottom available and the class SelectionTableToolbar was implementing the “Move Up” and “Move Down” buttons. The sub-class MoveUpButton and MoveDownButton used ownerModule and class MoveType to implement the above methods.* |
| 6 | *We inspected the class SelectionTableToolbar.* | *We found that the constructor of the class SelectionTableToolbar creates new objects of the class MoveButtonUp and MoveButtonDown which are responsible for moving the files up or down in the list of files and thus was relevant to our change request.* |
| 7 | *We also inspected MoveType.java* | *To get a better understanding of the Move Up/Down was implemented, we went through the file and also found other move types available to us.* |
| 8 | *We marked the class SelectionTableToolbar as “located”.* | *We confirmed this class had to be modified.* |

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

# Impact Analysis

Since new features are being added without changing any of the available functionality, therefore it had minimal or no impact on the related systems. However, we did the following impact analysis:

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | *We looked for dependencies of the class SelectionTableToolbar using JRipples.* | *To track the classes that could be impacted by the change.* |
| 2 | *Three dependent classes found namely SelectionTableToolbarWithoutMoveTest, SelectionTableToolbarTest, and MultipleSelectionPane* | *The first two dependencies are related to the automated testing and as it was not working in our case, so we skipped them. The third class just creates the object of the SelectionTableToolbar, therefore there was no actual change in that either. Thus, we just checked the class itself for any ripple effects but did not find any impact.* |

**Time spent (in minutes):** 25

# Actualization

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | *We created two more classes named MoveTopButton and MoveBottomButton which extends the class BaseMoveSelectionButton and we also defined a method with the same name which implements the function* | *We used this metods to create the buttons, add inline text, add description of the function, add context to the button and call the appropriate MoveType method.* |
| 2 | *We created two more objects MoveTopButton and MoveBottomButton in the method SelectionTableToolbar* | *We realized that first we need create two more buttons to add in the toolbar panel to move the files either to the top or to the bottom of the list.* |

**Time spent (in minutes):** 90

# Validation

|  |  |  |
| --- | --- | --- |
| Step # | Description | Rationale |
| 1 | *Test Case 1: drag and drop*  *“output.txt”*  *“input.txt”*  *“sample.txt”*  *“demo.txt”*  *Inputs: “demo.txt” Move Top*  *Expected outputs:*  *“demo.txt”*  *“output.txt”*  *“input.txt”*  *“sample.txt”* | *This is the regular expected behavior.*  *The test passed.* |
| 2 | *Test Case 2:*  *“output.txt”*  *“input.txt”*  *“sample.txt”*  *“demo.txt”*  *Inputs: “output.txt” Move Bottom*  *Expected outputs:*  *“input.txt”*  *“sample.txt”*  *“demo.txt”*  *“output.txt”* | *This is the regular expected behavior.*  *The test passed.* |

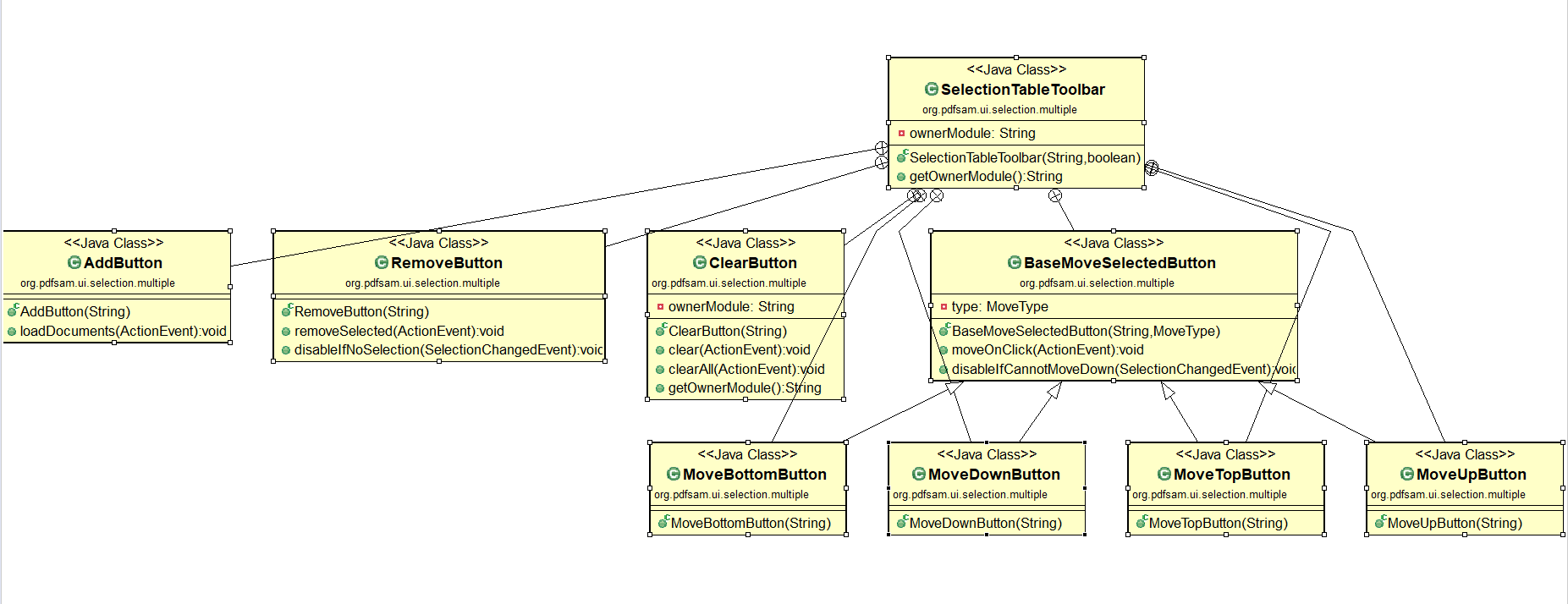
**Time spent (in minutes):** 30

# Timing

Summarize the time spent on each phase.

|  |  |
| --- | --- |
| Phase Name | Time (in minutes) |
| Concept location | 120 |
| Impact Analysis | 25 |
| Prefactoring | - |
| Actualization | 90 |
| Postfactoring | - |
| Verification | 30 |
| Total | 265 |

# Reverse engineering



A partial UML class diagram of the classes visited while navigating through the code.

# Conclusions

*This change request was relatively easy and took less time in general compared to the other change requests. Even though this is software is bigger than jEdit but its architecture and code are not complicated. As the target class was interacting with a lot classes, interfaces and abstract classes, thus we had to go through all of them one by one to get an understanding of which one is actually the target class and what methods and members are inherited from other classes. Once the concept location was completed, the impact analysis took considerable time as we had go through all the dependencies to find out which of these could get affected by our changes. After that we implemented the two required classes for the desired functions, created their objects and added them to the table toolbar. We manually tested the software as we were unable to build the software with automated testing. The tools that we used for this project are Eclipse IDE, InstaSearch, JRipples, Git, and ObjectAid.*

*Classes and methods changed:*

* *pdfsam-parent/pdfsam-fx/src/main/java/org/pdfsam/ui/selection/multiple/SelectionTableToolbar.java/SelectionTableToolbar*
  + *public SelectionTableToolbar(String ownerModule, boolean canMove)*
  + *static class MoveTopButton*
  + *static class MoveBottomButton*