

Change request log

___Team

Max Roselius

___Change Request

#1. This change request seeks to add a new button to the Alternate Mix and Merge modules that allows the user to move a selected document to either the top or bottom of the display list.

___Concept Location

Step #	Description	Rationale
1	<i>Ran the Alternate Mix and Merge features.</i>	<i>Visualize where the current buttons are, and get an idea for what to search for in the code (button names)</i>
2	<i>Performed a regex search in Eclipse with terms "Move Up" and "Move Down"</i>	<i>This is the text on the buttons, so it should lead to a starting point as to which class needs to be modified to add new buttons. This will also give an idea to which data structure will be modified to perform the actions.</i>
3	<i>Found class SelectionTable, which oddly contains code for buttons move to top and move to bottom. Need to find why they do not show.</i>	<i>Behavior for move to top and move to bottom works with shortcut keys, but no buttons exist.</i>
4	<i>Search workspace for instantiations of SelectionTable object</i>	<i>Want to see how the class is initialized and used</i>
5	<i>Found nothing useful with SelectionTable instantiation, searched workspace text for term UP</i>	<i>Noticed that buttons have move types associated with them, searching for exact matches of UP should lead to locations where buttons are created</i>
6	<i>Examined class SelectionTableToolbar,</i>	<i>Found class definitions for Move Up and Move Down buttons, this looks like spot where new buttons can be added</i>
7	<i>Mark SelectionTableToolbar as located.</i>	

Time spent (in minutes): 35

___Impact Analysis

Step #	Description	Rationale
1	<i>Checked for methods called and classes initialized by SelectionTableToolbar</i>	<i>To track the classes that could be impacted by the change.</i>
2	<i>Test functionality of move to top and move to bottom</i>	<i>To ensure the functionality works, and all I need to do is add buttons to the toolbar</i>

Time spent (in minutes): 10

___Actualization

Step #	Description	Rationale
1	<i>Created two new static classes in SelectionTableToolbar (one for move to top, one for move to bottom)</i>	<i>These classes are identical to the other defined buttons (move up and move down), except with different names and different functionality.</i>
2	<i>Modify SelectionTableToolbar's constructor to add the newly defined classes to its items</i>	<i>Need to make the buttons show</i>
3	<i>Rebuild the project and run it</i>	<i>Make sure the buttons show as expected</i>

Time spent (in minutes): 15

___Validation

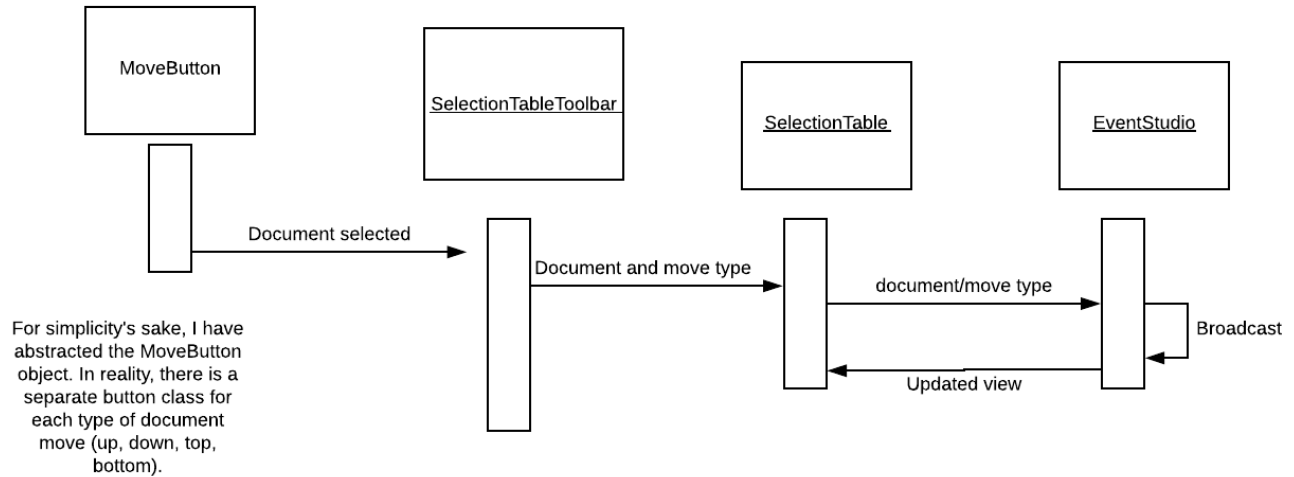
Step #	Description	Rationale
1	<i>Test case defined: Add 3 pdf documents in merge module, move bottom to top, top to bottom, middle to top, middle to bottom.</i>	<i>First test the buttons in merge module to make sure the changes were correct. Test passed</i>
2	<i>Test case defined: Same as above test, but for Alternate Mix module</i>	<i>Perform same test for Alternate Mix module. Test passed</i>
3	<i>For both modules, add 5 pdf documents to list, ensure that move up, move down function correctly</i>	<i>Ensure that the other buttons were not affected by the changes. Test passed</i>
4	<i>For both modules, add 5 pdf documents and move documents from top to bottom, bottom to top, middle to top, middle to bottom.</i>	<i>A more thorough test involving more documents. Ensures that document order shifts correctly. Test passed.</i>
5	<i>Pushed changes to GitHub</i>	<i>Only push changes once they work</i>

Time spent (in minutes): 10

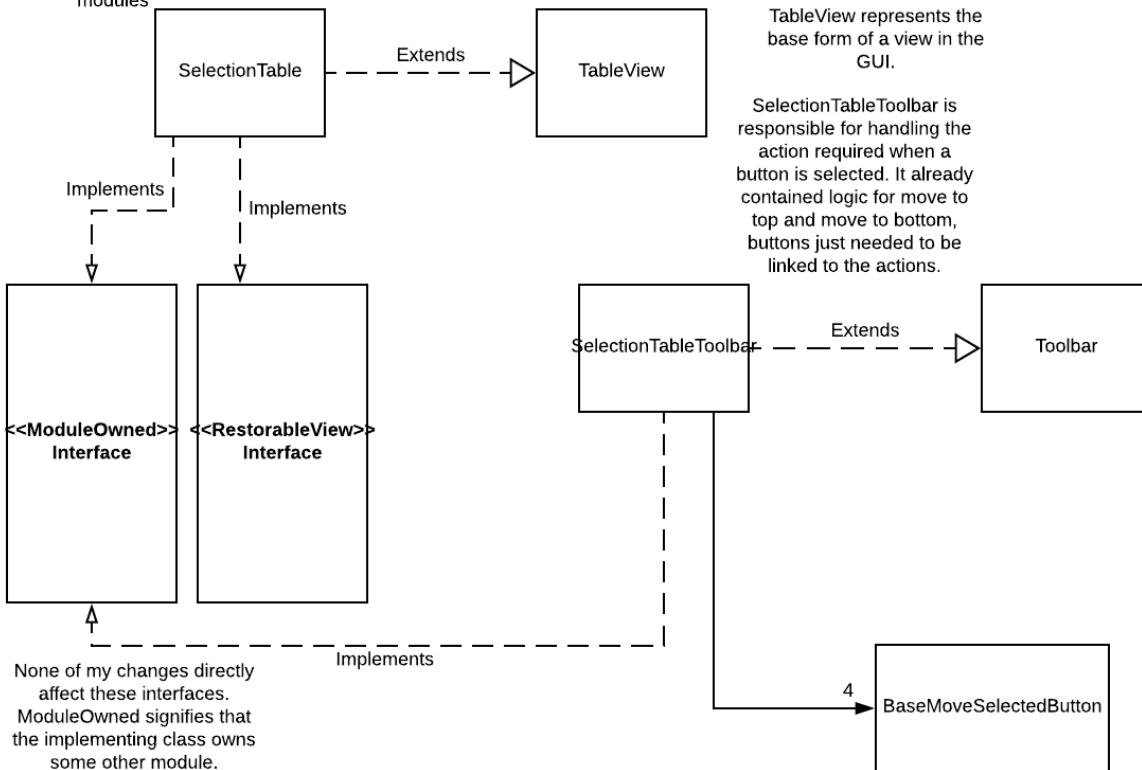
___Timing

Summarize the time spent on each phase.

Phase Name	Time (in minutes)
Concept location	35
Impact Analysis	10
Prefactoring	NA
Actualization	15
Postfactoring	NA
Verification	10
Total	70



SelectionTable is a specialized version of TableView, used by the Merge and Alternate Mix modules



TableView represents the base form of a view in the GUI.

SelectionTableToolbar is responsible for handling the action required when a button is selected. It already contained logic for move to top and move to bottom, buttons just needed to be linked to the actions.

None of my changes directly affect these interfaces. ModuleOwned signifies that the implementing class owns some other module. Implementing RestorableView requires a module view to maintain state when switching between other modules.

BaseMoveSelectedButton is the base class for the buttons on the toolbar. Already included were buttons for move up/down, so creating move to bottom/top buttons was rather simple.

—Conclusions

For this change, concept location was much easier than the first change, mostly because I had gained more familiarity with the system. Searching was relatively painless with the search terms being obvious to me. The change itself was also very easy, thanks to there being functionality for Move to Top and Move to Bottom already built, only the buttons needed to be added. Additionally, there were already existing buttons so I had a blueprint for how to add the new buttons in the correct location.

Classes and methods changed:

- *org/pdfsam/ui/selection/multiple/SelectionTableToolbar*
 - added two new classes: *MoveTopButton*, *MoveBottomButton*
 - changed constructor to build two new buttons and add them to its items list