

## Change request log

### Team

Specify the team members.

Brad Jakobitz

Nate Olson

### Change Request

PDFsam Change Request 1 (add buttons to allow documents to be moved to the top or bottom of the list)

### Concept Location

Step #	Description	Rationale
1	Ran the program to see what the "Alternate Mix" module looked like.	Wanted to see how the current format was and how difficult adding new buttons looked.
2	Searched the entire program for the text matching exactly "Move Up"	One of the buttons at the top of the Alternate Mix module is "Move Up" and we need to add a new button near that so it should be a good place to start looking.
3	Realized the place we found "Move Up" is for the right-click menu not for the buttons. So we spent some time looking at the right click menu.	It is the only place we found this text, so we figured the buttons might somehow be tied into a right-click or shortcut key.
4	Realized from exploring the right-click menu, there is already functionality that is implemented to move documents to the top and move documents to the bottom. Looks like all we will have to do is provide buttons to tie into the same functionality.	
5	Looking around the GUI more for clues about where to look for these in the code. Noticed a tooltip pops up for the "Move Up" button that says "Moves up selected documents" so we searched the source code for this exact String.	Even though we had discovered functionality for Moving documents to the top and bottom was most likely already there, we still had no indication of where in the source to look for this.
6	Found the above string in a single place (not counting internationalization). It is a tooltip for a "MoveUpButton" class which is a sibling to a "MoveDownButton" class. We believe this is where we will need to add new "MoveToTopButton" and "MoveToBottomButton" classes.	
7	We did an "Open Call Hierarchy" on the "MoveUpButton" class to see where it was being called from. It is only being called from the SelectionTableToolbar class constructor and only if the "canMove" flag is true.	Although we found out where we need to add the Classes for these new buttons, we still need to figure out how to get the buttons to appear on the GUI.
8	From examining the code, it looks like we can add the two new buttons as inner classes of SelectionTableToolbar (just like the current buttons) and then we should be able to call these at the same point in the SelectionTableToolbar constructor that the	

	MoveUpButton and MoveDownButton are being called (only if things are moveable).	
--	---	--

**Time spent (in minutes):** 21

## Impact Analysis

Step #	Description	Rationale
1	Anything calling the SelectionTableToolbar constructor is potentially impacted	This is the only class that we are changing. The new inner-classes being added are not public and will not be called from anywhere except for the Constructor of the SelectionTableToolbar.
2	Did an "Open Call Hierarchy" on the SelectionTableToolbar constructor.	This should give me a good indication of what needs to be examined.
3	The result is 2 tests (SelectionTableToolbarTest and SelectionTableToolbarWithoutMoveTest) and one other class (MultipleSelectionPane, more specifically, its constructor)	
4	Neither test will have to be changed (although an addition to the SelectionTableToolbarTest could be made to test the new functionality). The SelectionTableToolbarWithoutMoveTest will not even be impacted as it calls the SelectionTableToolbar constructor with canMove=false.	Just need to keep an eye on SelectionTableToolbarTest to ensure our changes do not break anything that currently works.
5	The MultipleSelectionPane class constructor calls the SelectionTableToolbar to set the top toolbar of its menu. This constructor should not have to be changed at all.	If this constructor is called with canMove=true, we want the "Move to Top" and "Move to Bottom" buttons to be shown.
6	This constructor is called by the AlternateMixSelectionPane, ExtractSelectionPange, MergeSelectionPane and RotateSelectionPane so it could impact more than just the AlternateMixSelectionPane change requested	Even though the Change Request only specifically asked for this functionality on the AlternateMixSelectionPane, this is the most logical place (lowest) to add the code. The fact that it may affect other modules should not be an issue as the overall feel of the application will remain consistent (The buttons will be the same for every module that allows moving files).

**Time spent (in minutes):** 6

## Prefactoring (optional)

Step #	Description	Rationale
1	At this point, we do not believe any prefactoring will be required. The	This is a good indication that the common functionality that we need for the

	MoveUpButton and MoveDownButton are already subclasses of the BaseMoveSelectedButton.	MoveToTopButton and MoveToBottomButton are likely already provided in the BaseMoveSelectedButton.
2	We also discovered that the "MoveType" required for the new buttons is already provided.	The BaseMoveSelectedButton requires a "MoveType" to be given, but because the right-click menu already allows moving to the top and the bottom, there is already a "MoveType" defined for both of these actions.

**Time spent (in minutes): 2**

## Actualization

Step #	Description	Rationale
1	Added the MoveToTopButton to the SelectionTableToolbar class. This involved setting the tooltip and the text of the button, but because we only know English, we made use of an already internationalized "Move to top" String and the tooltip of "Moves selected document to the top" will not be internationalized.	We needed this new class for the button to move things to the top. As for the internationalization, if we were actually checking this in, we would inform the proper team/people that there is a new string that needs to be internationalized or make note of it (or whatever that projects protocol called for).
2	Added the MoveToBottomButton to the SelectionTableToolbar class. We also made use of an existing string for the button text here, but had to create a new string for the tooltip, which was only added in English.	We needed this new class for the button to move things to the bottom. Same note about internationalization as above.
3	Added the MoveToTopButton and MoveToBottomButton to the SelectionTableToolbar constructor in the same location as the MoveUpButton and the MoveDownButton	We chose to add the MoveToTop and MoveToBottom buttons on the right-hand side of the existing buttons. If we were working with a marketing team or a UX designer, we would defer to them as to where these buttons should be (e.g., MoveToTop next to MoveUp instead)

**Time spent (in minutes): 8**

## Postfactoring (optional)

Step #	Description	Rationale
1	we did not do any post factoring	After adding the two new constructors and calling them in the same place as the other existing buttons, we believe we have met the criteria of the Change Request and everything is still laid out logically in the code.

**Time spent (in minutes): 1**

## Validation

Step #	Description	Rationale
1	Test case defined: <ul style="list-style-type: none"><li>- Launch app to see if new buttons are there</li></ul> Inputs: <ul style="list-style-type: none"><li>- None</li></ul> Expected output: <ul style="list-style-type: none"><li>- Appearance of new buttons</li></ul>	This is standard behavior as the buttons need to be there to use.  The test passed.
2	Test case defined: <ul style="list-style-type: none"><li>- Ensure if the top document is selected, the "Move to Top" button is disabled.</li></ul> Inputs: <ul style="list-style-type: none"><li>- A couple (<math>\geq 2</math>) documents being loaded and the top document being selected.</li></ul> Expected output: <ul style="list-style-type: none"><li>- The "Move to Top" button is disabled</li></ul>	This is normal behavior. If the top document is selected the "Move Up" button is disabled, so the "Move to Top" button should follow the same logic.  The test passed.
3	Test case defined: <ul style="list-style-type: none"><li>- Ensure if the bottom document is selected, the "Move to Bottom" button is disabled.</li></ul> Inputs: <ul style="list-style-type: none"><li>- A couple (<math>\geq 2</math>) documents being loaded and the bottom document being selected</li></ul> Expected output: <ul style="list-style-type: none"><li>- The "Move to Bottom" button is disabled</li></ul>	This is normal behavior. If the bottom document is selected the "Move Down" button is disabled, so the "Move to Bottom" button should follow the same logic.  This test passed.
4	Test case defined: <ul style="list-style-type: none"><li>- Ensure if multiple documents are selected both the "Move to Top" and "Move to Bottom" buttons are disabled.</li></ul> Inputs: <ul style="list-style-type: none"><li>- Multiple (<math>\geq 3</math>) documents being loaded and at least two, that are not at the top, being selected.</li></ul> Expected Output: <ul style="list-style-type: none"><li>- Both the "Move to Top" and "Move to Bottom" button should be disabled.</li></ul>	This is normal behavior. This is the same behavior observed with the right-click menu. Although one could argue whether you should be able to move multiple documents to the top (and keep their respected order), our goal is to keep the application consistent. If somebody later decides to allow this functionality, this can be done in a single place since we did not make exceptional behavior for the buttons.  This test passed.
5	Test case defined: <ul style="list-style-type: none"><li>- The top document can move all the way to the bottom while preserving the order of the other documents</li></ul> Inputs: <ul style="list-style-type: none"><li>- Exactly 2 documents, the top document being selected, and the "Move to Bottom" button being clicked.</li><li>- More than 2 documents, the top document being selected, and the "Move to Bottom" button being clicked</li></ul> Expected Output: <ul style="list-style-type: none"><li>- The top document, that was selected, should move to the bottom, while the order of the other documents should</li></ul>	This is normal behavior. The selected document should move to the bottom and the order of the unselected documents should remain unchanged.  This test passed.

	remain unchanged.	
<b>6</b>	<p>Test case defined:</p> <ul style="list-style-type: none"> <li>- The bottom document can move all the way to the top while preserving the order of the other documents</li> </ul> <p>Inputs:</p> <ul style="list-style-type: none"> <li>- Exactly 2 documents, the bottom document being selected, and the "Move to Top" button being clicked.</li> <li>- More than 2 documents, the bottom document being selected, and the "Move to Top" button being clicked</li> </ul> <p>Expected Output:</p> <ul style="list-style-type: none"> <li>- The bottom document, that was selected, should move to the top, while the order of the other documents should remain unchanged.</li> </ul>	<p>This is normal behavior. The selected document should move to the top and the order of the unselected documents should remain unchanged.</p> <p>This test passed.</p>
<b>7</b>	<p>Test case defined:</p> <ul style="list-style-type: none"> <li>- Made sure the middle documents can be moved to the top and the bottom without changing the order of the other documents involved.</li> </ul> <p>Inputs:</p> <ul style="list-style-type: none"> <li>- Exactly 3 documents, the middle document being selected, and the "Move to Bottom" button being clicked.</li> <li>- Exactly 3 documents, the middle document being selected, and the "Move to Top" button being clicked.</li> <li>- Greater than 3 documents, one of the middle documents being selected and the "Move to Bottom" button being clicked.</li> <li>- Greater than 3 documents, one of the middle documents being selected and the "Move to Top" button being clicked.</li> </ul> <p>Expected Output:</p> <ul style="list-style-type: none"> <li>- The document that was selected moves to the bottom or the top (depending on the button clicked) and the order of the other documents remains unchanged.</li> </ul>	<p>This is normal behavior, we expect a document to be moved to the bottom or top depending on which button is clicked and for this action to leave the other documents in an unchanged state.</p> <p>This test passed.</p>
<b>8</b>	<p>Test case defined:</p> <ul style="list-style-type: none"> <li>- Ensured the other modules impacted by the change looked and functioned in a similar matter</li> </ul> <p>Inputs:</p> <ul style="list-style-type: none"> <li>- ExtractSelectionPane</li> <li>- RotateSelectionPane</li> <li>- MergeSelectionPane</li> <li>- All panes with few documents loaded</li> </ul> <p>Expected Output:</p> <ul style="list-style-type: none"> <li>- Either:</li> <li>- A) The pane does not display the new buttons if it does not display the old "Move Up" and "Move Down" buttons</li> <li>- Or</li> <li>- B) The pane does display the new buttons and test cases 2-7 all pass with this module/pane</li> </ul>	<p>This is to ensure we did not break other modules by making a change in the AlternateMix module.</p> <p>The ExtractSelectionPane shows no move buttons – this module passed the test.</p> <p>The RotateSelectionPane shows no move buttons – this module passed the test.</p> <p>The MergeSelectionPane does show both the old and new move buttons and passed the tests outlined in steps 2-7. This module passed the test.</p>

9	<p>Test case defined:</p> <ul style="list-style-type: none"><li>- The Maven build that came with the PDFsam program that has already defined tests to ensure the program functions correctly.</li></ul> <p>Inputs:</p> <ul style="list-style-type: none"><li>- The PDFsam build script</li></ul> <p>Expected Output:</p> <ul style="list-style-type: none"><li>- The same results as when we ran this test immediately downloading PDFsam (290 tests run, 1 failure, 0 errors and 6 skipped – failure should be in PrefixFieldTest)</li></ul>	<p>This test passed as the results were exactly the same as they were when we originally downloaded the code.</p> <p>The single failure was in the PrefixFieldTest. The error shown below is what we get when we run these tests (on official PDFsam v3.3.5 branch or in our code)</p>
---	---	--

Time spent (in minutes): 20

```
[INFO] Results:
[INFO]
[ERROR] Failures:
[ERROR]   PrefixFieldTest.contextMenuReplacesText:67 expected:<[](BASENAME)> but was:<[PDFsam_](BASENAME)>
[INFO]
[ERROR] Tests run: 290, Failures: 1, Errors: 0, Skipped: 6
```

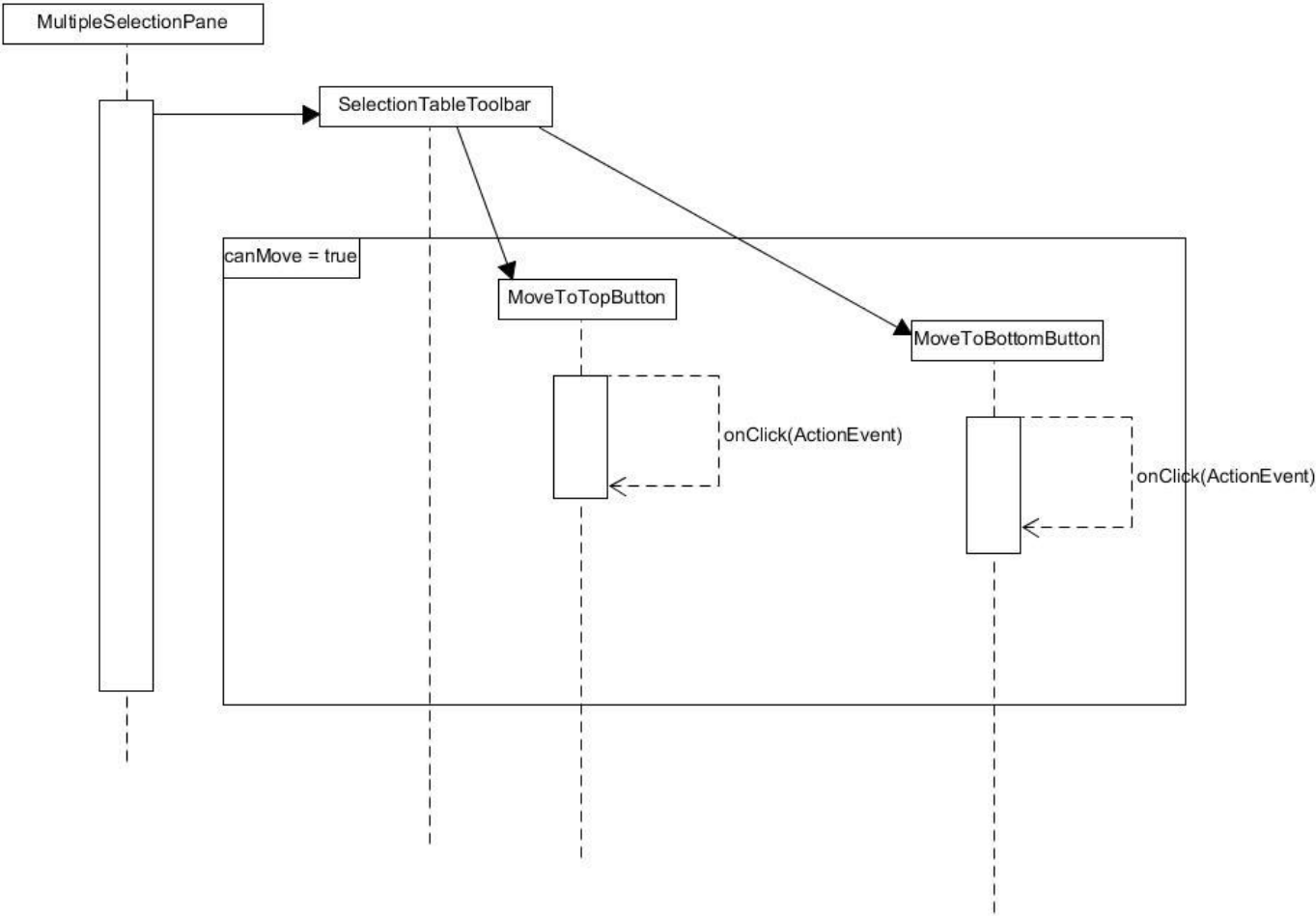
Timing

Summarize the time spent on each phase.

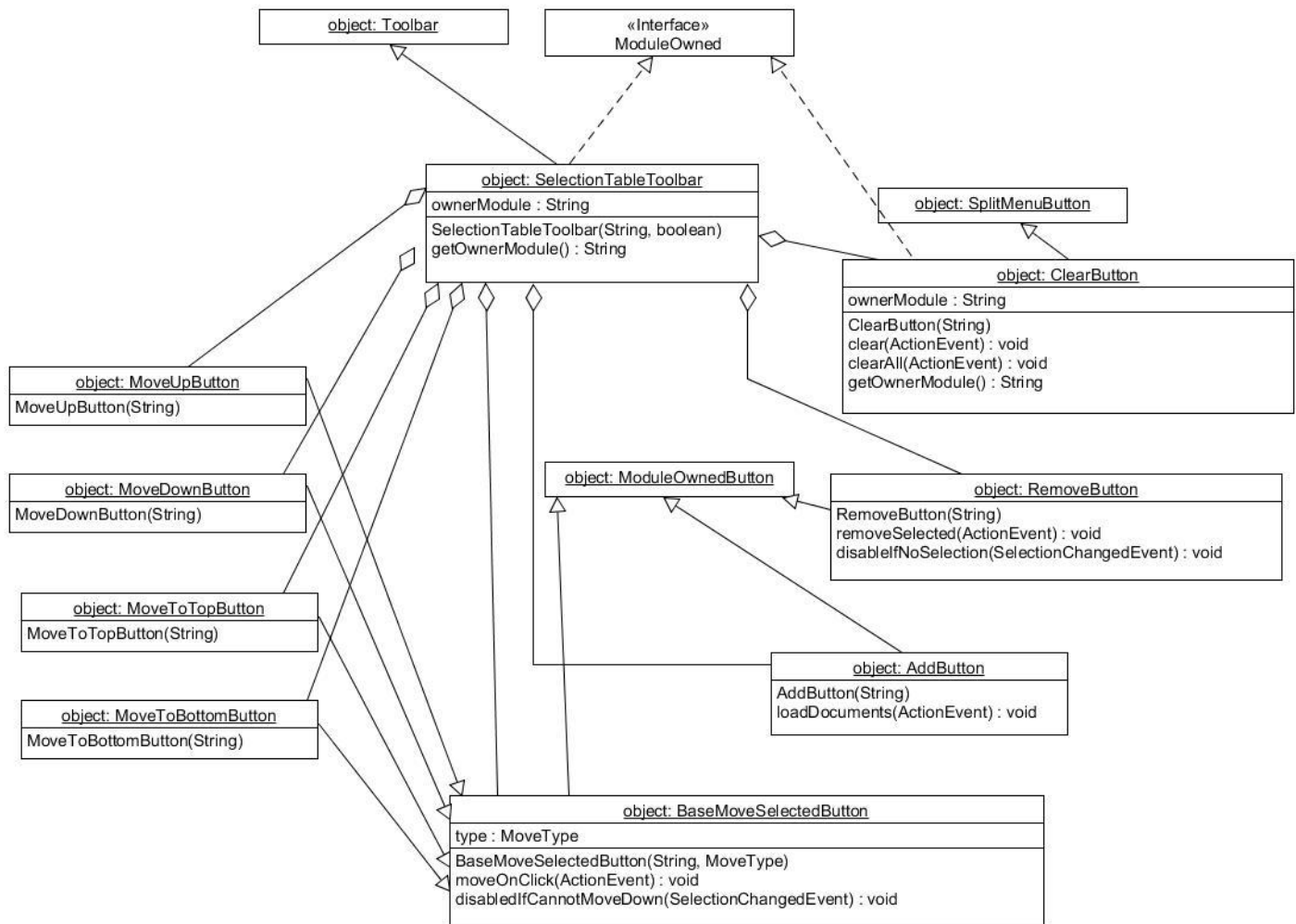
Phase Name	Time (in minutes)
Concept location	21
Impact Analysis	6
Prefactoring	2
Actualization	8
Postfactoring	1
Verification	20
Total	58

Reverse engineering

Sequence Diagram:



## Class Diagram:



## Conclusions

As expected with source code we are completely unfamiliar with, the most time consuming part of this change request was locating where the change needed to occur. This also happened to be a simple fix once it was located as there was already functionality built into the program to perform the actions required, we just needed to provide the buttons that linked to the existing actions. The only thing we were not capable of doing, that we would consider necessary, is providing internationalization of the new tooltips we added. Once the changes were finished, testing was also a pretty time consuming part, but this is expected from any change unless there is a very sophisticated, automated, set of regressions tests that do not require developer time.

Classes and methods changed:

- pdfsam-fx/src/main/java/org/pdfsam/ui/selection/multiple/SelectionTableToolbar.java
  - o Added new inner classes
    - MoveToTopButton
    - MoveToBottomButton
  - o SelectionTableToolbar constructor