Change request log

1 Team

BugHunters

Team Members: Sadaf Ghaffari, Prashant Thakur, Bibek Shrestha

2 Change Request

Allow the intersection of page ranges a document in the Merge Module of PDFsam

3 Concept Location

Step #	Description	Rationale
1	*We ran the system*	
2	*We used the Insta Search tool*	*To find the location we intend to change.*
3	*We searched for "Page Range" using the regular expression*	*Because we identified "page range" can lead us to where we want to change.*
4	*We tried the Merge module on PDFsam and change the pdf ranges on the options*	*To check and verify the page range works properly.*
5	*We inspected the class SelectionTableRawData, PageRangesColumn,ConversionUtils, and MergeSelectionPane. We went to this class using the keyword search Page Range.*	*We noticed that the these are the classes that PageRange keyword exists and plays a role.*
6	*We marked the class ConversionUtils as "located".*	*We confirmed this class had to be modified.*

Time spent (in minutes): 45

4 Impact Analysis

Step #	Description	Rationale
1	*We made a list of classes that include page range keyword and visited them to locate which parts needs to be changed. The classes in our lists that needed inspection were SelectionTableRawData, PageRangesColumn, ConversionUtils,and MergeSelectionPane*	*To track the classes that could be impacted by the change.*
2	*We inspected them and made sure the change was not going to lead large changes in the behavior of the system.*	

Time spent (in minutes): 30

5 Actualization

Step #	Description	Rationale
1	*We made modifications toPageRangeSet method in ConversionUtils to validate the page range in the GUI*	*This method was the main one for modifying the page range.*
2	*We ran the the test cases.*	*To make sure everything works.*

Time spent (in minutes): 30

6 Validation

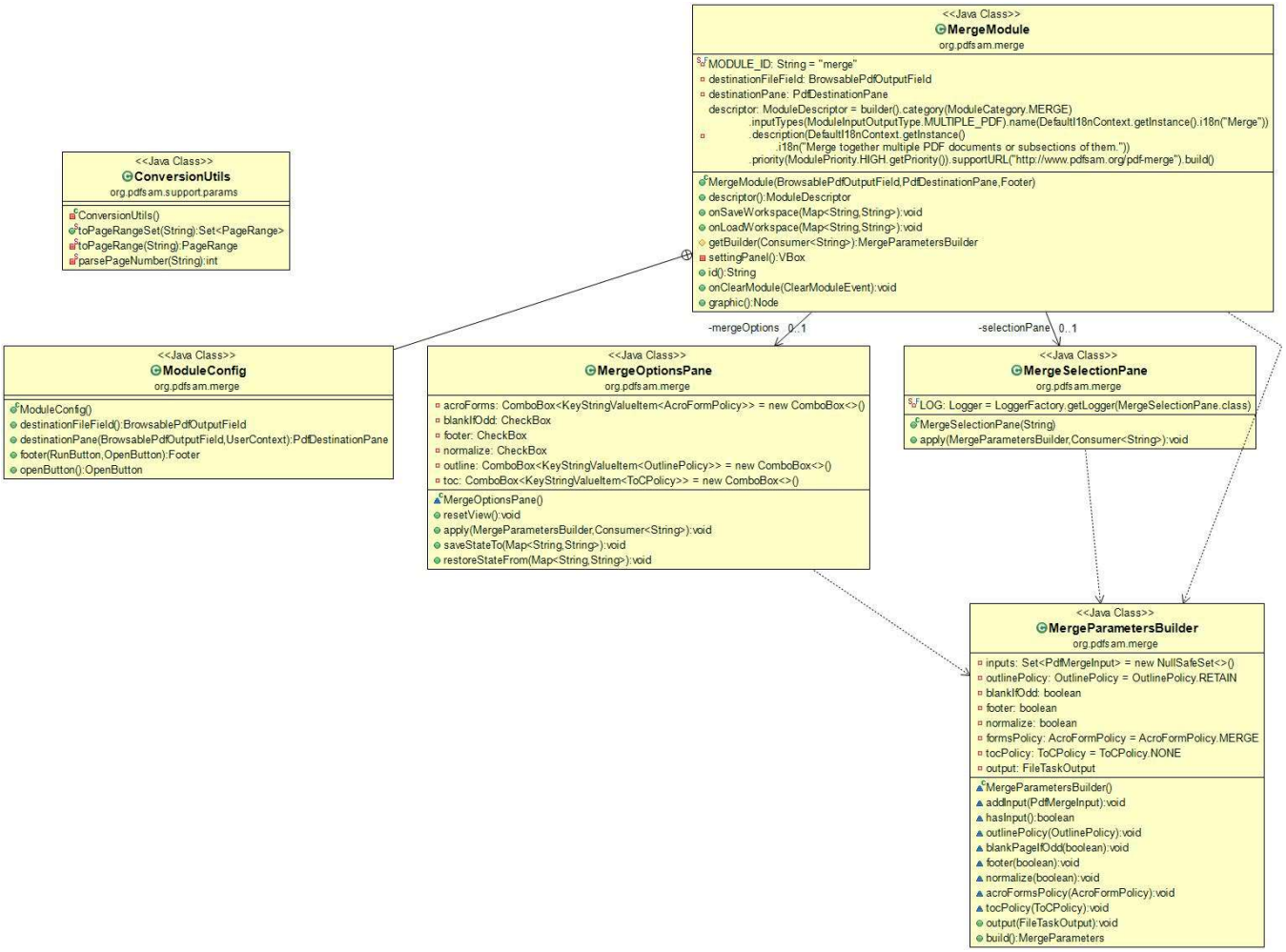
Step #	Description	Rationale
1	*Test case defined:* *Inputs: 1-20, 6-30* *Expected output:1-20, 21-30*	*This is the regular expected behavior.* *The test passed.*
2	*Test case defined:* *Inputs: 1-20, 6, 6-30* *Expected output: 1-20, 21-30*	*The test passed.*
3	*Test case defined:* *Inputs: 1-9, 10-40, 50, 30 -80* *Expected output: 1-9, 10-40, 41-80*	*The test passed.*
4	*Test case defined:* *Inputs: 1-3, 5-30, 6-9* *Expected output: 1-3, 5-30*	*The test passed.*

Time spent (in minutes): 15

7 Timing

Phase Name	Time (in minutes)
Concept location	45
Impact Analysis	30
Prefactoring	-
Actualization	30
Postfactoring	-
Verification	15
Total	120

8 UML sequence diagram for the visited classes



9 Conclusions

For this change, concept location was not too difficult, and its architecture and code are not very complicated, though there was no documentation for the code and comments were not provided. The ConversionUtils class was modified. The Concept location was done by text search utility like InstaSearch and grep. The impact analysis, actualization and validation were relatively easy. Testing was performed using multiple test cases to make sure the module works properly as requested.

Classes and methods changed:

- `pdfsam-core/src/main/java/org/pdfsam/support/params/ConversionUtils.java`
    - `static Set<PageRange>toPageRangeSet(String selection)`

Github Link: https://github.com/prashantkthakur/cs515-s19-a2-BugHunters-pdfsam401