Change request log

1 Team

Evolutionary Rams:

Nada Alalyani

Sanket Mehrotra

2 Change Request

ID	Description
#ps-2: merge module : allow intersecting ranges;	Allow the entry of overlapping ranges in the merge module, such that the pages in the overlap set are duplicated.

3 Concept Location

In the table below we describe each step we followed when performing concept location for this change request. In our description, we include the following information when appropriate and indicate is as follows in the below table:

3.1 ps#2: Allowing intersecting ranges in Merge module

Step #	Description	Rationale/Discovery
1	We searched for all functions in the project that related to key word "merge".	To list all functions and classes that could involve the location of the concept.

2	The following classes are found:
	`Merge Module (5 matches)`
	`Merge optionsPane (4 matches)`
	`Merge ParametersBuilder`
	`MergeSelectionPane (3 matches)`	

3	We checked `MergeParametersBuilder` class. we found `addInput(PdfMergeInput)` function called by `mergeParameters` which is one of the matches.	To see how how addInput function add input to inputs set.
4	Using break points in this class and other merge classes such as pageRange, Base MergeParameters classes, we trace the flow of adding inputs to Set<PdfMergeInput> inputs = **new** NullSafeSet<>();	

5	We noticed the change of variables through classes and methods. The variable `pageSelection` which composed of (start, end) of the ranges is firstly set in `AbstractList.class`	We cannot change in this file since its compiled class.
6	Moving through the flow of calls, we found that addInput function receive the page selection which has been set and add it to inputs = **new** NullSafeSet<>();	
7	Therefore, addInput function is the location where the inputs can be controlled.	

Time spent (in minutes): 130 mins

4 Impact Analysis

Step #	**Description**	**Rationale**
1	The change in addInput(PdfMergeInput input) might affect the following classes and methods in the next column.	Using call hierarchy we found all classes and methods that call addInput function. **class**: MergeSelectionPane.java • **method:** apply(MergeParametersBuilder builder, Consumer<String> onError) ○ Line of code: forEach(builder::addInput); called as parameter to add input to PdfMergeInputs set. These are test classes: **class** MergeParametersBuilderTest.java • **void** build() **class** MergeSelectionPaneTest.java • **void** converstionException() • **void** empty() • **void** emptyByZeroPagesSelected() • **void** emptyPageSelection() • **void** notEmptyPageSelection()
2	the call hierarchy of class: MergeSelectionPane is checked.	**class** MergeModule.java • **method:**MergeParametersBuilder getBuilder(Consumer<String>onError) - Apply () method is called from MergeParametersBuilder class as shown below: ○ **Private** MergeSelectionPane selectionPane = **new** MergeSelectionPane(**MODULE_ID**);

		▪ selectionPane.apply(builder, onError);
3	After listing the above classes, we see that some of the references are from test classes.	These references are not from actual functionalities, they will not significantly affect performance.
4	However, the call of addInput from actual function apply and getBuilder in MergeSelectionPane class and MergeModule respectively are not affected by the change.	There is no any applied change on ranges of selected pages. The changes applied on the way of merging pages based on the required range input. So, it is allowing duplicate of pages if there is an intersection between pages.

Time spent (in minutes): 30 minutes

5 Actualization

Step #	**Description**
1	Inside addInput Function, the array list of pageRange type is created and the set of pageRanges is copied to the new arrayList. ArrayList<PageRange> pglist = new ArrayList<PageRange>(set);
2	Object of PdfMergeInput is initiated to create new input PdfMergeInput tempInput = new PdfMergeInput(input.getSource());
3	The page ranges are added one by one to the new input tempInput.addPageRange(pglist.get(i));
4	*T*he new inputs added to Set<PdfMergeInput> inputs this.inputs.add(tempInput);

Time spent (in minutes): 5 hours

6 Validation

Step #	Description	Rationale
1	**Test case defined:** merge pdf pages as it is ordered in the entered range. **Inputs:** [1-5], [3-8] **Expected output:** the pages in this flow 1,2,3,4,5,3,4,5,6,7,8 should be merged.	We tested the this functionality if it is working as it is requested. This functionality is Merging pdf pages with duplicates of pages if there is an intersection between in the entered ranges. This test with the mentioned inputs and expected output is passed.

Time spent (in minutes): 15 minutes

7 Timing

Phase Name	Time (in minutes)
Concept location	90 mins
Impact Analysis	30 mins
Prefactoring	----
Actualization	60 mins
Postfactoring	-----
Verification	15 minutes
Total	525 mins 8 hours 45 minutes

8 Reverse engineering

- **Class Diagram**

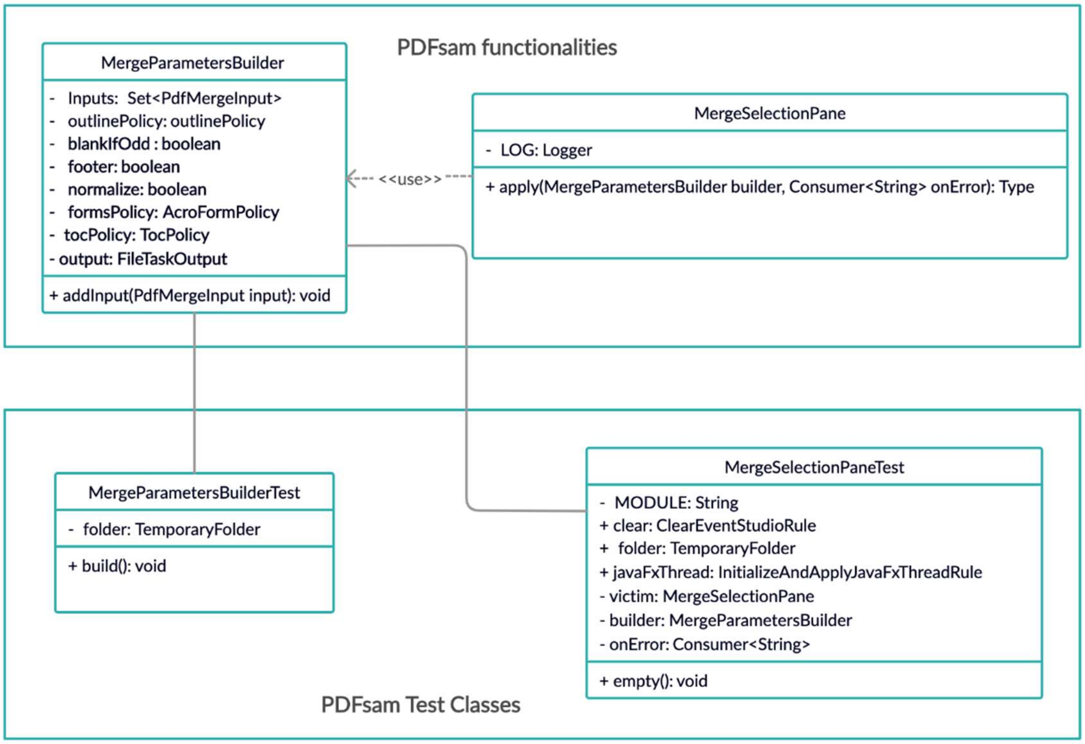
PDFsam functionalities
MergeParametersBuilder
- Inputs: Set<PdfMergeInput>
- outlinePolicy: outlinePolicy
- blankIfOdd : boolean
- footer: boolean
- normalize: boolean
- formsPolicy: AcroFormPolicy
- tocPolicy: TocPolicy
- output: FileTaskOutput
+ addInput(PdfMergeInput input): void
<<use>>
MergeSelectionPane
- LOG: Logger
+ apply(MergeParametersBuilder builder, Consumer<String> onError): Type
MergeParametersBuilderTest
- folder: TemporaryFolder
+ build(): void
MergeSelectionPaneTest
- MODULE: String
+ clear: ClearEventStudioRule
+ folder: TemporaryFolder
+ javaFxThread: InitializeAndApplyJavaFxThreadRule
- victim: MergeSelectionPane
- builder: MergeParametersBuilder
- onError: Consumer<String>
+ empty(): void
PDFsam Test Classes

Figure 1: Class diagram for ps#2

Sequence Diagram

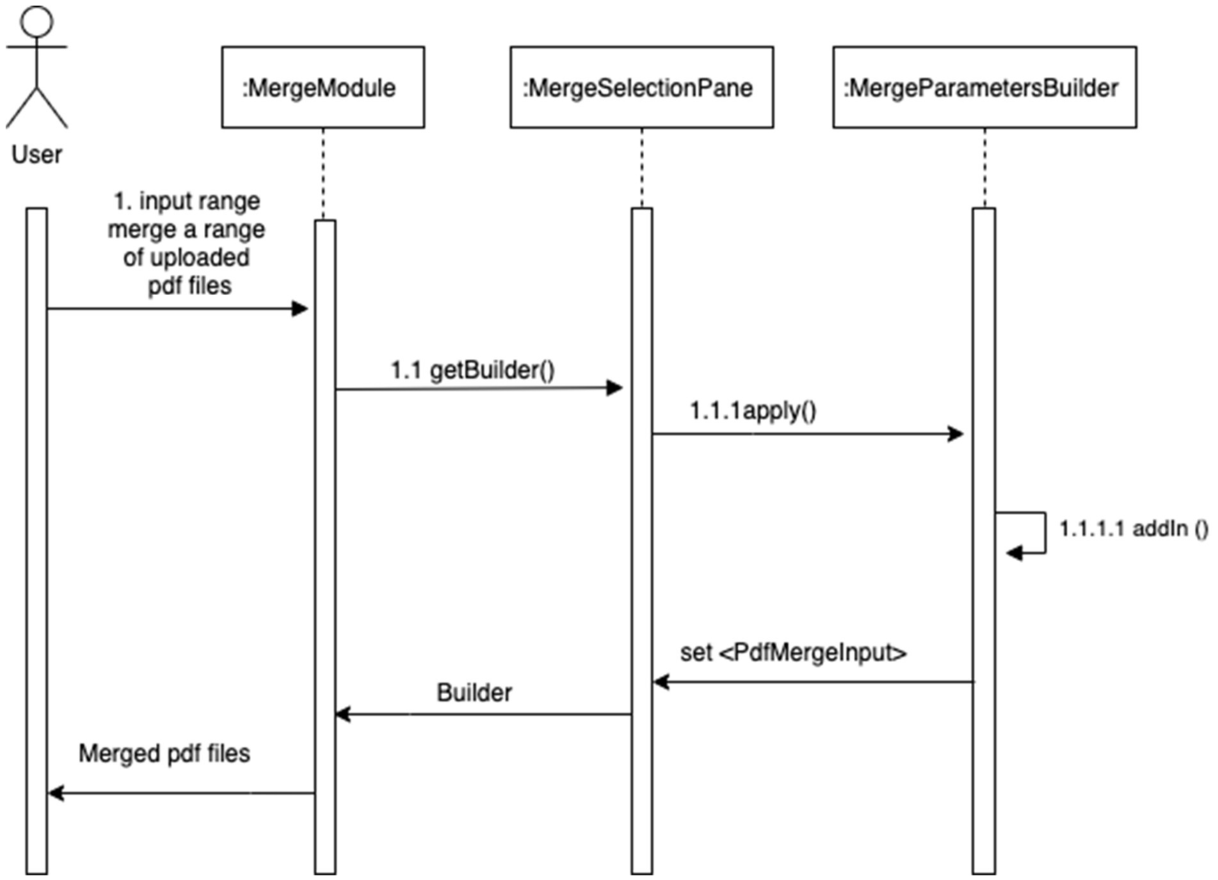

Figure 2: Sequence diagram for ps#2

9 Conclusions

For merge change request, there are interpretation challenge of the question. Specifically, what we interpret from the change request is avoiding the intersection among ranges and the output will be without duplication. However, the change request is intended to provide a pdf file that merge all pages that entered in the ranges even though there is a duplication of pages. There is not specific details the the change request that assist maintainers to interpret it by this way. Actualization step took a while when we started implementing the first interpretation of the request. Then, it took less than one hour for the actual interpretation that meant.

There is one method that change

- /pdfsam-merge/src/main/java/org/pdfsam/merge/MergeParametersBuilder.java
    - void addInput(PdfMergeInput input)