Change request log

1 Team

SURAJ ESWARAN

2 Change Request

Change Request #ps3

Fixation of rotating the pdf with specific options and page ranges are been presented. That is, to consider both the page-range and the even/odd/all options

https://github.com/surajeswar22/cs515-801-s20-Eswaran-pdf

3 Concept Location

Step #	Description	Rationale
1	*I ran the system*	
2	*Initially, I worked on searching the methods or variables that consist of rotating pdf*	*Since, Page Range is the only source where it can lead to the actual fixation*
3	*I tried the rotate module on PDFSam on various pdf ranges*	*To check whether the following range works or not.*
4	*I inspected few classes like RotateModule, RotateOptionsPane, RotateParametersBuilder and SelectionTableRawData.*	
5	*I marked the class "RotateParameterBuilder" as "located".*	*I confirmed this class had to be modified.*

Time spent (in minutes): 60

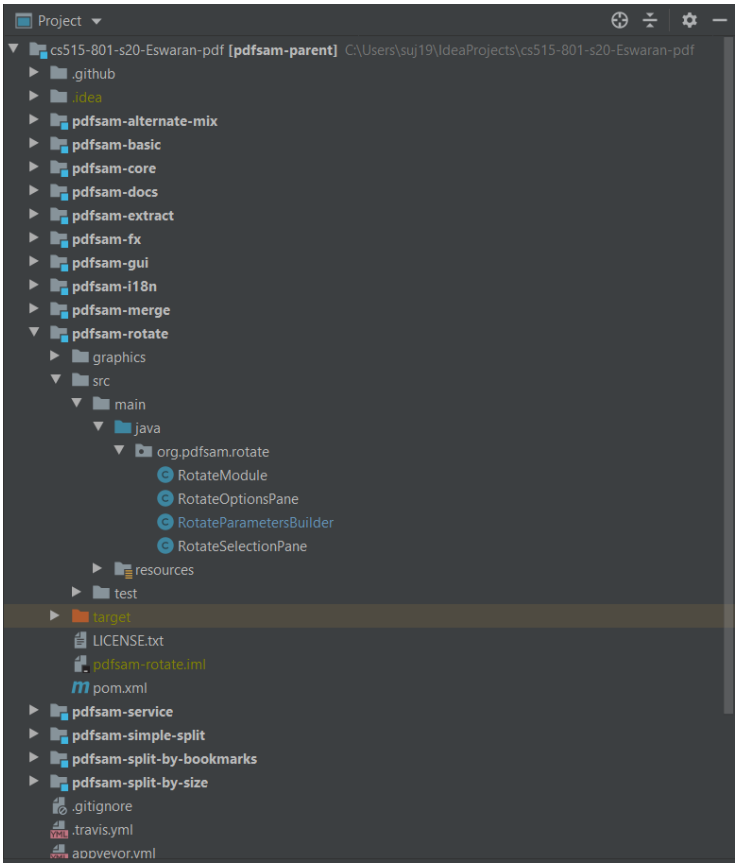

Figure 1: List of classes to be modified as per change request

4 Impact Analysis

Step #	Description	Rationale
1	*I found a list of classes that includes rotation and visited them to find out which part of the code must be changed. I inspected the class "RotateParameterBuilder". Such a class was marked as "to change" as well*	*To track the class that could be impacted by the change.*
2	*I made sure that change in the code must not lead to irregular behavior in the software*	

Time spent (in minutes): 65

5 Actualization

Step #	Description	Rationale
1	*I made modification in "RotateParameterBuilder" to validate rotation with specific ranges and options in it.*	.
2	*Finally, the test cases has been running*	*To make sure everything works.*

Time spent (in minutes): 90

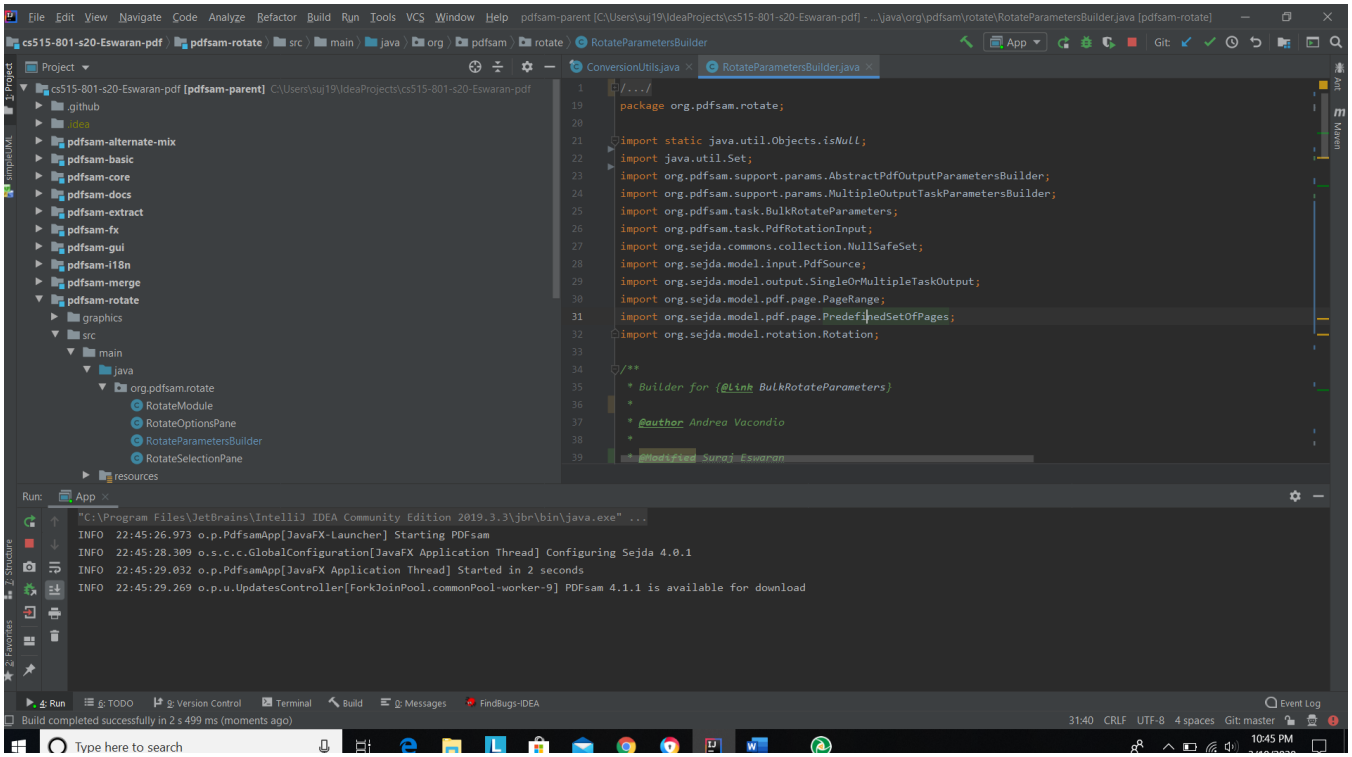

Figure 2: Running of the modified change

6 Validation

Step #	Description	Rationale
1	*Test case defined: EvenCase()* *Inputs: 1-10* *Expected output:2,4,6,8,10*	*This is the regular expected behavior.* *The test passed.*
2	*Test case defined: OddCase()* *Inputs:1-10* *Expected output: 1,3,5,7,9*	*This is the regular expected behavior.* *The test passed.*
3	*Test case defined: AllCase()* *Inputs:1-10* *Expected output: 1,2,3,4,5,6,7,8,9,10*	*This is the regular expected behavior.* *The test passed.*

Time spent (in minutes): 30

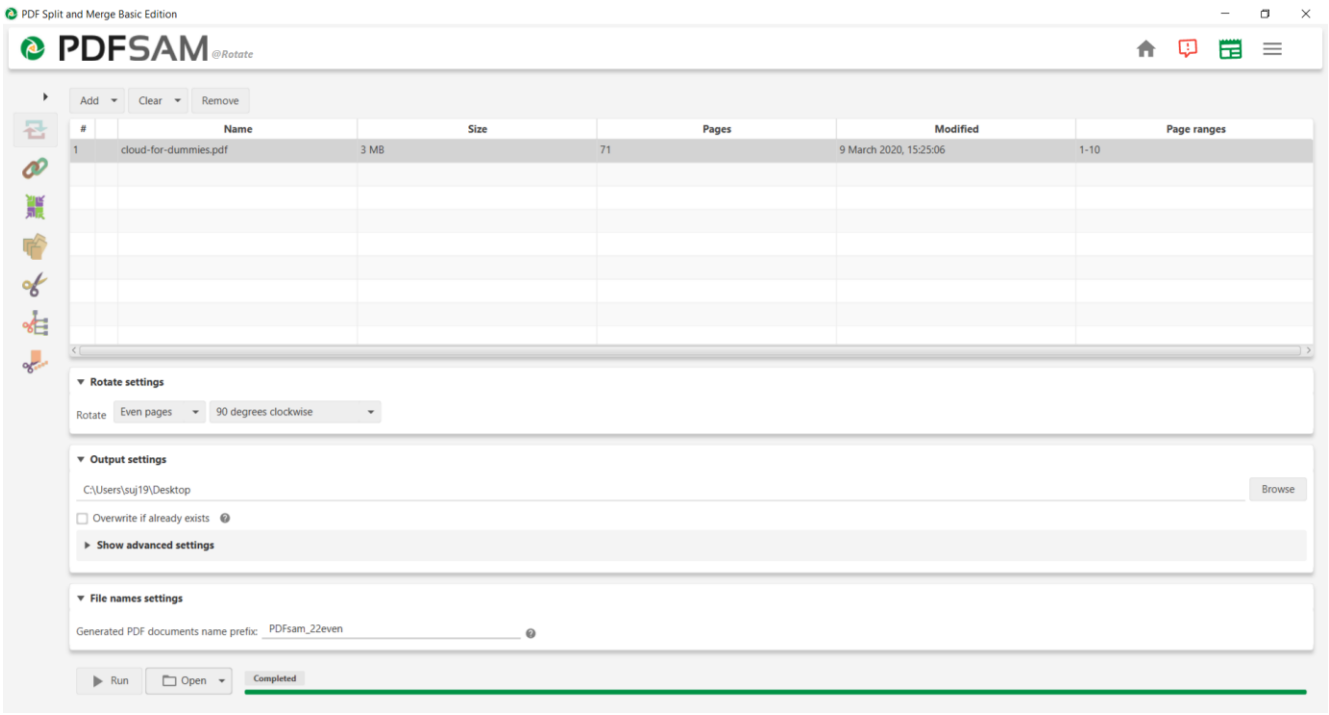

Figure 3: Output of the requested change

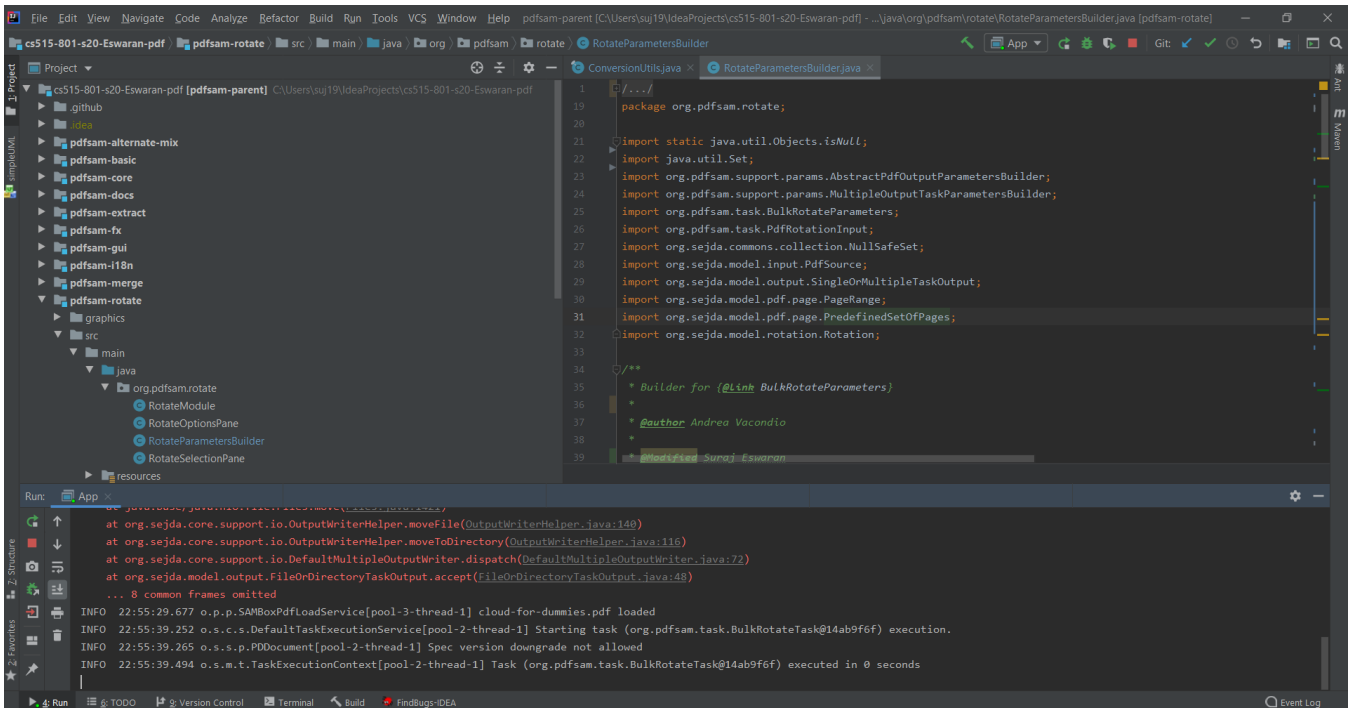

Figure 4: Status update of the Output of the modified change

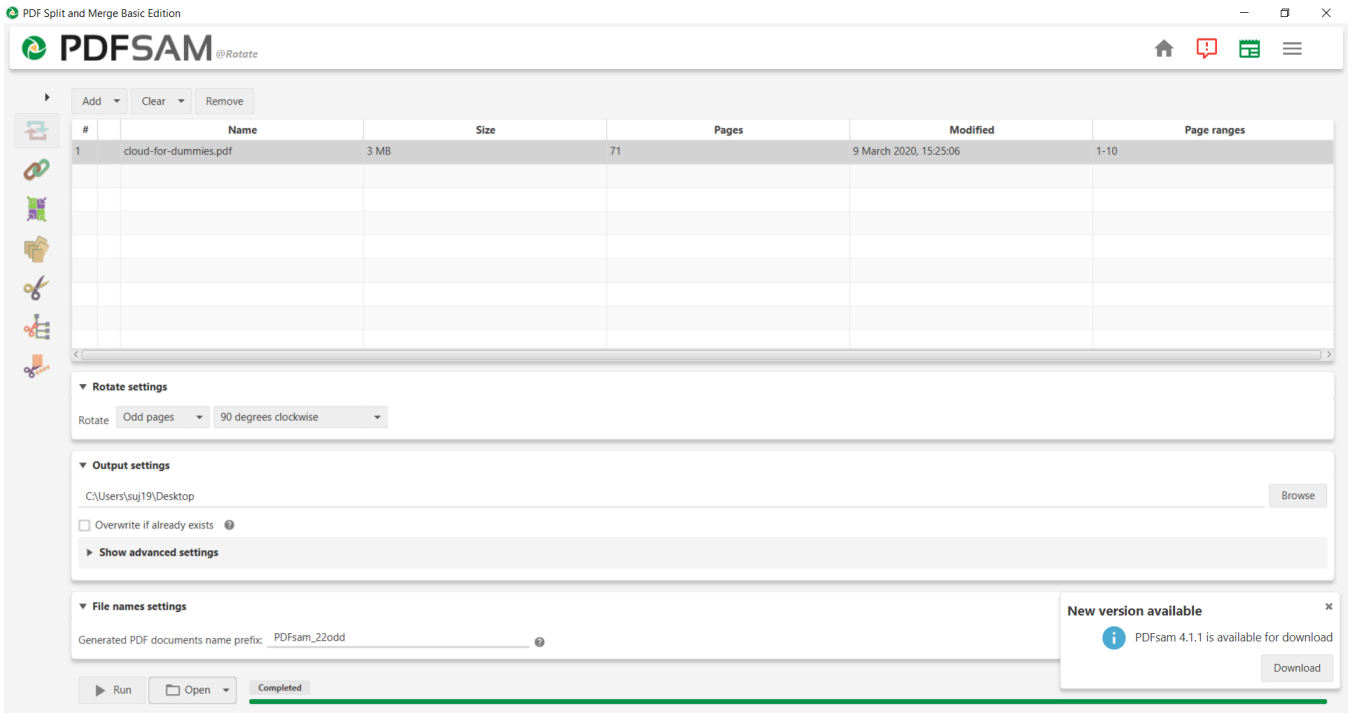

Figure 5: Output of the requested change with completion

7 Timing

Summarize the time spent on each phase.

Phase Name	Time (in minutes)
Concept location	60
Impact Analysis	65
Prefactoring	-
Actualization	90
Postfactoring	-
Verification	30
Total	245

8 Reverse engineering

UML is basically a diagrammatic representation for a software. With the help of diagrams, it is possible to better understand several flaws and errors in software.

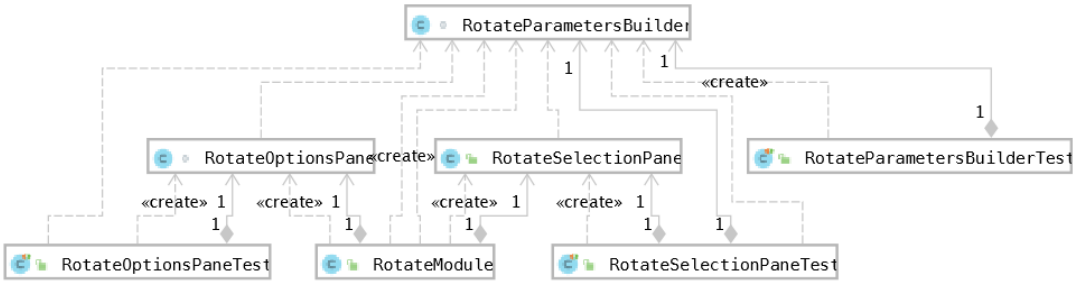

Figure 6: UML Diagram for #ps3

ACTORS	CUSTOMERS/USERS	The only actor of the system is the user where they deploy the feature or functionality in it.
RotateParametersBuilder		The function where the change has to be with options like even pages and odd with specific ranges.
RotateOptionsPane		The function which controls the option of even, odd and all pages
RotateSelectionPane		The function of selection of choices with ranges in it.

*Since the class diagram contains a list of variables and classes in the PdfSam_rotate that's the file is available in github. The tool used for generating UML is SimpleUML from Intellij.

9 Conclusions

For this change, concept location seems to be easy because there were not any parameters that determines odd and even pages, thus it was not complicated in finding it. Here, I have used Manual Testing where the functionality was used each time.

Classes and methods changed:

- *\pdfsam-rotate\src\main\java\org\pdfsam\rotate*
- *Void addinput()*

https://github.com/surajeswar22/cs515-801-s20-Eswaran-pdf