Change request log

1. Team

Supraj - Driver

Veshwaj - Navigator

2. Change Request

Change request #4: implementing an option to clear recent directories

3. Concept Location

Use the table below to describe each step you follow when performing concept location for this change request. In your description, include the following information when appropriate:

- IDE Features used (e.g., searching tool, dependency navigator, debugging, etc.)
- Queries used when searching
- System executions and input to the system
- Interactions with the system (e.g., pages visited)
- Classes visited
- The first class found to be changed (this is when concept location ends)

When there is a major decision/step in the process, include its rationale, i.e., why that decision/step was taken.

Make sure you time yourselves when going through this process and provide the total time spent below.

The following is an example of a concept location process for the change request "Color student schedule":

Step #	Description	Rationale
1	We ran the system.	
2	Looked at the RecentFilesProvider.java class from change request #2 and saw code that deletes recent files.	The instructions mentioned that recent directories would need to be deleted in the same way that recent files are deleted. So we looked at the RecentFilesProvider.java class to see how the recent files get deleted.
3	Searched for RecentDirectoriesProvider.java using the navigate feature and found a class with the name.	Because knew that a RecentFilesProvider.java class exists from the previous change request, we looked for one that related to directories.

Time spent (in minutes): 10

4. Impact Analysis

Use the table below to describe each step you follow when performing impact analysis for this change request. Include as many details as possible, including why classes are visited or why they are discarded from the estimated impact set.

Do not take the impact analysis of your changes lightly. Remember that any small change in the code could lead to large changes in the behavior of the system. Follow the impact analysis process covered in the class. Describe in details how you followed this process in the change request log. Provide details on how and why you finished the impact analysis process.

Step #	Description	Rationale
1	We made a list of the methods in RecentDirectoriesProvider.java. The only two methods were updateEveryTime() and update().	Since we'll be making changes to this class, we need to check where the methods in the class are called.
2	By right-clicking each method, we see that they are each only called once in EnhancedMenu.init().	We figured that looking at where these methods were called would help determine whether changing either method would impact other parts of the project.
3	We determined that making changes to updateEveryTime() could potentially affect how EnhancedMenu.init() would run.	updateEveryTime() is a boolean method called within a conditional expression in EnhancedMenu.init(). This means that parts of EnhancedMenu.init could either run or not run depending on what this method returns.
4	We determined that making changes to update() would not impact how EnhancedMenu.init runs.	

Time spent (in minutes): 15

5. Actualization

Use the table below to describe each step you followed when changing the code. Include as many details as possible, including why classes/methods were modified, added, removed, renamed, etc.

Step #	Description	Rationale
1	We copied and pasted the code from the RecentFilesProvider.java class that deleted the recent files into the RecentDirectoryProvider.java class	We noticed that the RecentFilesProvider.java class was fairly similar to that of the RecentDirectoryProvider.java class so we thought that the code should work
2	We then changed the label from "clear-recent-files.label" to "clear-recent-directories.label"	Because this code is in the RecentDirectoryProvider.java class, changes need to be made to the label
3	We searched for the "clear-recent-files" label using the navigate function to find where we would need to create a new label and found that labels were added in a file called "jedit_en.props".	We thought that labels for both files and directories would be stored in the same file.
4	We added a new label called "clear-recent-directories.label=Clear Recent Directories"under the section of this file reserved for the Recent Directories Menu.	The label needs to exist in this file in order for the code from step 2 to function like it needs to.
5	The menuItem.addActionListener(e -> BufferHistory.clear()) was changed to menuItem.addActionListener(e -> model.removeAllElements());	The code for now deletes all of the recent directories since there was no BufferHistory.clear() method that would have worked

Time spent (in minutes): 45

6. Validation

Use the table below to describe any validation activity (e.g., testing, code inspections, etc.) you performed for this change request. Include the description of each test case, the result (pass/fail) and its rationale.

Step #	Description	Rationale
1	We ran jEdit and opened a directory.	
2	We clicked on “Clear Recent Directories” under the Utilities tab of the main menu.	We expected that clicking on this would clear any recent directories from the Recent Directories tab, including the current directory. The test passed.

Time spent (in minutes): 10

7. Timing

Summarize the time spent on each phase.

Phase Name	Time (in minutes)
Concept location	10
Impact Analysis	15
Actualization	45
Verification	10
Total	80

8. Reverse engineering

Create a UML sequence diagram (or more if needed) corresponding to the main object interactions affected by your change.

Create a partial UML class diagram of the classes visited while navigating through the code. Include the associations between classes (e.g., inheritance, aggregations, compositions, etc.), as well as the important fields and methods of each class that you learn about. The diagram may have disconnected components. Use the UML tool of your preference. When a significant fact about a class or method is learned, indicate it via annotations on the diagram. **For each change request, start with the diagram produced in the previous change request. For the first, you will start from scratch.**

9. Conclusions

For this change, locating where the change needed to be made was fairly straightforward. We knew from change request #2 that a RecentFilesProvider class existed, and the instructions themselves even referenced how the recent directories needed to be deleted in the same manner that the recent files needed to as well. The actualization part was a little more time-consuming but was still straightforward nonetheless. We tested the change by running the code and checking to see if a “delete recent directories” tab existed and clicked it to see if it worked.

Classes and methods changed:

- org/jedit/localization/jedit_en.props
    - Created a label for Clear Recent Directories
- org/gjt/sp/jedit/menu/RecentDirectoriesProvider.java
    - Copied, pasted, and altered deletion code from RecentFilesProvider.java class