Change request log

1 Team

Evolutionary Rams:
Nada Alalyani
Sanket Mehrotra

2 Change Request

ID	Description
#je-3: Hypersearch modification;	allow more than one match per line.

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 je#3: Fixing the HyperSearch Match highlight function

Step #	Description	Rationale
1	We ran the project and applied the functionality	To understand the fault that should be changed.
2	We navigated the JEdit project, org/gjt/sp/Jedit	To find the package that could include the concept location
3	We navigated "search package", which is in this path: org/gjt/sp/Jedit.	To find all classes related to search function
4	To dig into the code, we used break points and debugging features in Eclipse IDE.	
5	We checked 3 classes, HyperSearchRequest, HyperSearchResults, and HyperSearchResult.	To try finding the concept location at these classes. However, it was not located.
6	We checked searchMatcher (Parent class), patternSearchMatcher and BoyerMooreSearchMatcher (children classes).	To try finding the concept location at these classes. However, it was not located.

7	We debugged BoyerMooreSearchMatcher class, specifically, nextMatch function.	To ensure that nextMatch function returns all matches in the same line. We found that it correctly returns them, so it does not have any problem.
8	Then, we debugged dohypersearch function, which is in HyperSearchRequest class.	Because it calls NextMatch, which we had already checked it, it might have the problem in displaying the matches.
9	The overloading method "dohypersearch" in the same class HyperSearchRequest was debugged, and it returns the number of occurrences correctly.	Because the overloading method "dohypersearch" called the previous one.
11	Checking for highlighting concepts in the whole project and we found 1102 matches.	To facilitate reaching to classes related to highlighting, however, there were any bugs related to those functions.
12	However, we found function convertValueToText in HyperSearchResults class sending results (matches) to HtmlUtilities.highlightString, which might include the bug.	After we debug the code and trace the matches in this function, we found the bug in this function.

Time spent (in minutes): 200 mins

4 Impact Analysis

Step #	Description	Rationale
1	The change is in the following location might affect the following classes and methods that call it. **class** HyperSearch Results • String convertValueToText(Object value, **boolean** selected,**boolean** expanded, **boolean** leaf, **int** row, **boolean** hasFocus) This function is inherited from JTree class. The methods that call this function are in the next column.	By inspecting the code, we found that convertValueToText calls the super class Jtree and implement **class** JTree.**class** • **Method:** String convertValueToText(Object value, **boolean** selected, **Boolean** expanded, **boolean** leaf, **int** row, **Boolean** hasFocus)
2	we can conclude that the change of convertValueToText() does not impact the performance of any functions in project since it is not called by other modules.	

Time spent (in minutes): 60 minutes

5 Actualization

Step #	Description	Rationale

1	There is a piece of code is commented. This code is
	finally {
	m = null;
	}
	
	m is the offsets of matches.
		To justify our change, the work of this function should be clarified:
		1. It starts with finding the matches in the beginning of the line
		2. Enters the while loop to move through the line and return the matches in the variable 'm'
		3. However, inside the loop there is finally block that assigns 'm' to null.
		4. Because of that, the while loop is ended since its condition m does not equal null.
		5. The function then starts from the beginning of the next line and so on. That is why the 'finally' block is commented.

Time spent (in minutes): 20 mins

6 Validation

Step #	Description	Rationale
1	Test case defined: finding all matches in the text Inputs: search for 'th' Text: The reasearch is conducted on the lab is more expensive than ... Expected output: "The reasearch is conducted on the lab is more expensive than " These substrings should be highlighted.	We tested the function by entering those text and all these substrings highlighted correctly. Test Passed

Time spent (in minutes): 10 mins

7 Timing

Summarize the time spent on each phase.

Phase Name	**Time (in minutes)**
Concept location	200 mins
Impact Analysis	60 mins
Prefactoring	----
Actualization	20 mins
Postfactoring	----
Verification	10 mins
Total	290 mins

8 Reverse engineering

Class Diagram

This class diagram is added to the previous change request's diagram. **Note:** The child class (HyperSearchResults) has the change but there is not classes that call it. However, there are other .class modules calls the method in the super class (JTree).

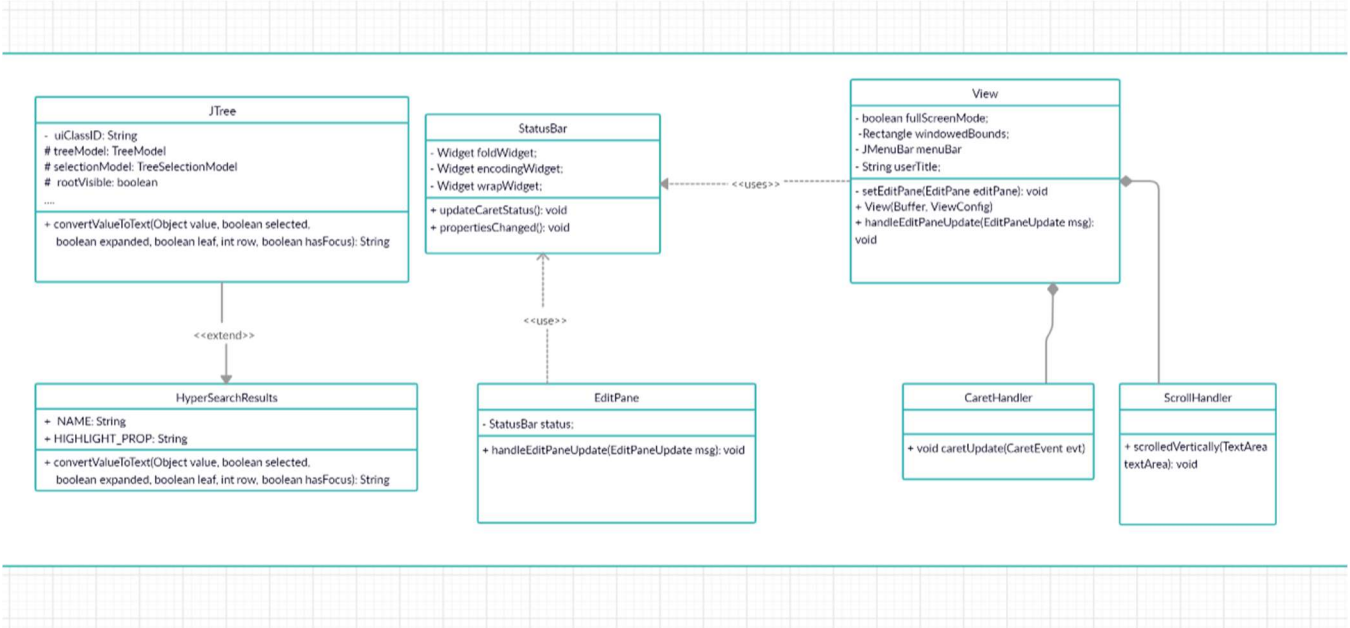

Figure 1: Class diagram for je#1 and je#3

Sequence diagram

This the behavior of the function that has the change. No other classes call this function.

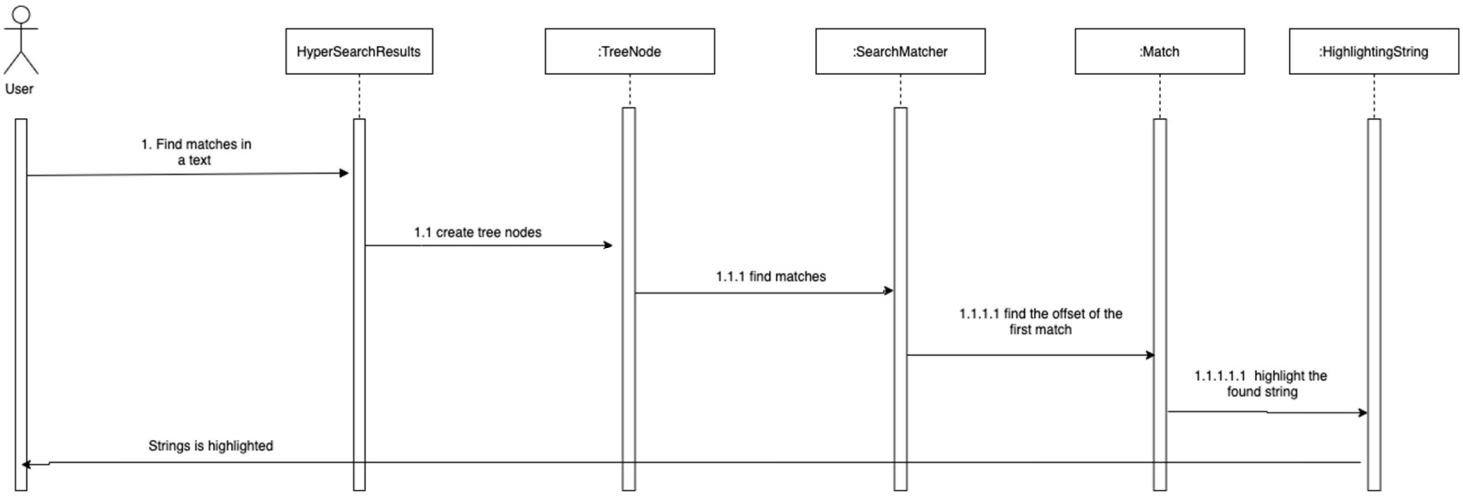

Figure 2: Sequence diagram for je#3

9 Conclusions

For this change, concept location was difficult since there are many functions that should be checked and traced to find the bug. Additionally, we started with Jedit functions and we still had difficulty in understanding the code, using debugging tools, and finding the bugs. However, the change was extremely easy since we only remove one block that prevent the correct output to be shown.

Classes and methods changed:

- Jedit/org/gjt/sp/jedit/search/HyperSearchResults.java
    - convertValueToText(Object value, **boolean** selected,**boolean** expanded, **boolean** leaf, **int** row, **boolean** hasFocus)