Name:

Daniyal Ghani
cms 53440

Text Highlight

1. **Change Request*:***
   Highlight lines based on type in hyper-search results
   • comment line match in green
   • source code match in red

2. **Concept Location:**

To find the concept that is responsible for hyper-searching text in jEdit project, we use the following systematic approach: First, we identify the search folder in jEdit project. There are usually 10-15 files dedicated to different text and query search within jEdit project. Next, we filter these files to find those that contain the hyper-search techniques and classes. Then, we use GREP to find the exact location within the code that is responsible for highlighting the search results. Finally, we find the "HighlightingTree" class, which houses the "highlightString" function. The "HighlightingString" function takes the text from the hyper-search bar, compares it to the existing code, and if a match is found, then it triggers text highlighting (usually in silver). We customize this behavior by changing the background color to red within this section. Finally, the conditions for handling comment and multiline comments are incorporated into the 'HighlightingTree' class, which is responsible for controlling the search and text matching conditions. This systematic approach ensures a high level of control & customization for text highlighting in jEdit project's hyper-search function.

Table 1. The list of all the classes visited during concept location.

#	File name	Tool used	Located?	Knowledge acquired
	HyperSearchResult.java	Ctrl+shift+N	org/gjt/sp/jedit/search/HyperSearchResults.java	Handle search
	HtmlUtilites.java	Ctrl+shift+N	org/gjt/sp/util/htmlUtilities.java	Text highlight

3. **Impact Analysis:**

Table 2. The list of all the classes visited during impact analysis.

#	Class name	Tool used	Impacted?	Knowledge acquired
	HighlightingTree	Manually	changed	

4. **Prefactoring (optional):**

5. **Description of the implementation/actualization:**

In order to complete this task, we need to add the following changes to the files: HyperSearchResults.JAVA To change the comment lines' background color to green, we add the following changes in the HyperSearchResult.JAVA file: To check if a string contains (++), change the background color to green. To change normal text to red, add the following code in the file: SearchTextBackgroundColors.JAVA

6. **Postfactoring (optional):**

7. **Testing:**

Manual testing by searching

8. **Sources:**

GPT

9. **Highlighted Source Code:**

HyperSearchResults code

```
@Override
public String convertValueToText(Object value, boolean selected,
        boolean expanded, boolean leaf, int row, boolean hasFocus)
{
    String s = super.convertValueToText(value, selected, expanded, leaf,
        row, hasFocus);
    if (s.contains("//") || s.contains("/") || s.contains("*")) {
        return "<html><span style='background-color:#00FF00;'>"+s+"</span></html>";
    }
```

Htmlutilities code

```
public static String highlightString(String s, String styleTag, List<Integer> ranges)
        {
                StringBuilder sb = new StringBuilder("<html><style>.highlightcolor { background-color:red; </style>");

                sb.append(styleTag);
                sb.append("}</style><body>");
                int lastIndex = 0;
                for (int i = 0; i < ranges.size(); i += 2)
                {
                        int rangeStart = ranges.get(i);
                        int rangeEnd = ranges.get(i + 1);
                        appendString2html(sb, s.substring(lastIndex, rangeStart));
                        sb.append("<span class=\"highlightcolor\">");
                        appendString2html(sb, s.substring(rangeStart, rangeEnd));
                        sb.append("</span>");
                        lastIndex = rangeEnd;
                }
```