

**Simple Merge**

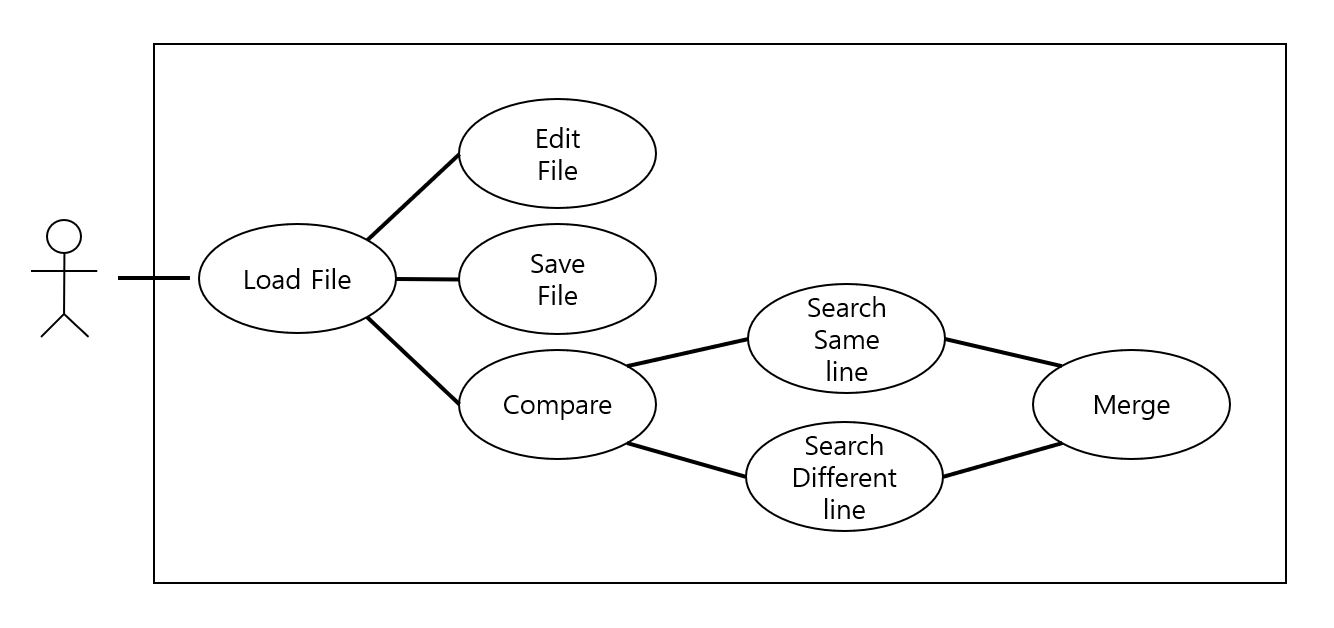
**Software Requirement Specification**

Team number **#5**  
Team member 김성재, 박준영, 이현재, 주현준, 전진우

**1. Introduction**

This system is for data comparison and merging text-like files. Users can compare codes between two different files, distinguishing similar code and merge, save, view and edit the files.

**2. Use Case Diagrams**

****

**3. Use Cases Descriptions**

1. **Load file.**

**1.1 Preconditions:**

At least two text files needed.

**1.2 Main Flow:**

Right after the program is started, the main window with two edit panels will be displayed. On each edit panel, when “Load” button is clicked, then the program should allow the user to choose a file in the file system.

**1.3 Subflows:**

None.

**1.4 Alternative Flows:**

None.

1. **Edit file.**

**2.1 Preconditions:**

None.

**2.2 Main Flow:**

When user presses a “Edit” button, the program allow the user to edit the content that shown in the edit panel **[E1]**.

**2.3 Subflows:**

None.

**2.4 Alternative Flows:**

**[E1]** When user clicked “Edit” again, the program disallow the user to edit the content.

1. **Save file.**

**3.1 Preconditions**

Load must be completed to execute Save.

Changes needed in loaded C code files.

**3.2 Main Flow**

After Load files, user can save changes to the file at anytime [S1].

**3.3 Subflows**

**[S1] ‘**Save as**’ :** Browse window / Save files at any directory and name that user sets.

**3.4 Alternative Flows**

Save cannot executed before file loaded

1. **Compare two files.**

**4.1 Predictions**

Load must be completed to execute Compare

**4.2 Main Flow**

This system provides Compare function that compare the files in both editing windows to highlight different parts. After user clicks Compare button, it shows different parts line by line with highlighted background color [S1] in each edit window.

**4.3 Subflows**

**[S1]** If the two parts are the same, it will be displayed in white background; If the two parts are different, it will be displayed in yellow background; If there is a empty line in the different part, it will be displayed in grey background.

**4.4 Alternative Flows**

Compare cannot executed before file loaded.

1. **Merge two files.**

**5.a Copy to Right**

**5.a.1 Preconditions**

After comparing two codes, it makes the lines get colored if the lines are different.

**5.a.2 Main Flow**

When the lines get colored, press “Copy to Right” button. the button works as copying the colored lines in the left panel to the colored lines in the right panel. when the user press the button one more time, the two colored lines in left and right make equal.

**5.a.3 Subflows**

None

**5.a.4 Alternative Flows**

there is nothing to happen if the two colored lines are equal.

**5.b Copy to Left**

**5.b.1 Preconditions**

After comparing two codes, it makes the lines get colored if the lines are different.

**5.b.2 Main Flow**

When the lines get colored, press “Copy to Left” button. the button works as copying the colored lines in the right panel to the colored lines in the left panel. when the user press the button one more time, the two colored lines in left and right make equal.

**5.b.3 Subflows**

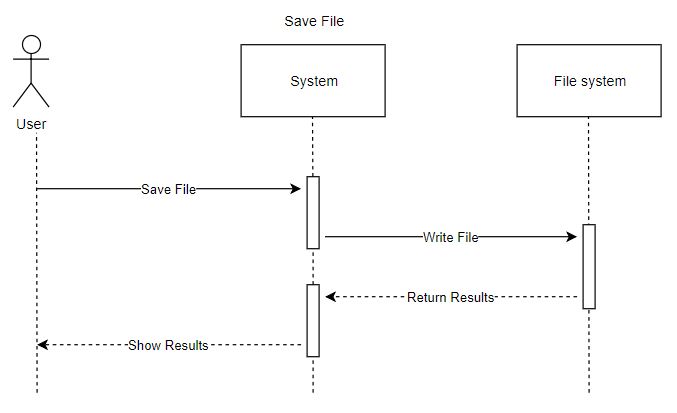
None

**5.b.4 Alternative Flows**

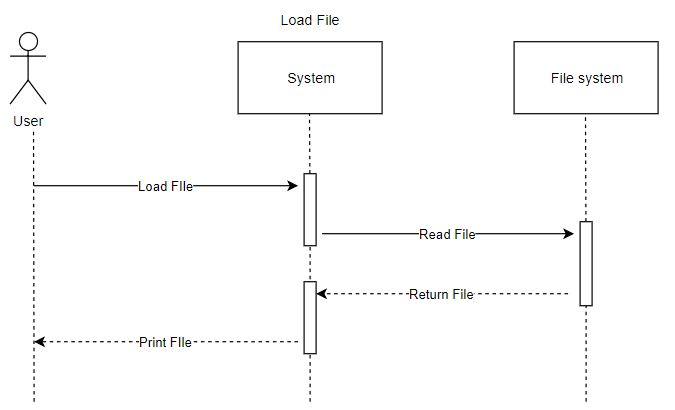
there is nothing to happen if the two colored lines are equal.

**4. System Sequence Diagrams**

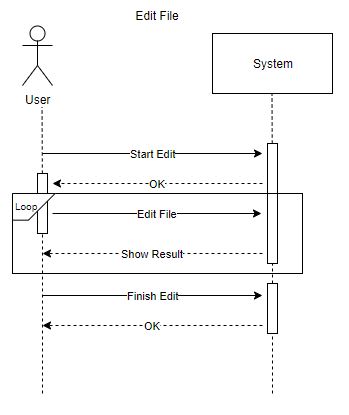
1. **Save File**

****

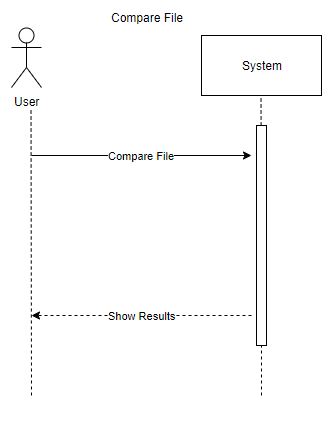
1. **Load File**

****

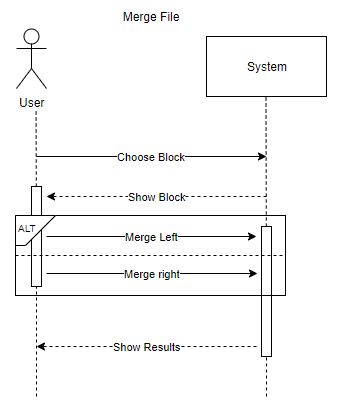
1. **Edit File**

****

1. **Compare File**

****

1. **Merge File**

****

**5. Non-functional requirements**

**NR1. Performance**

The System shall wait for a user inputs and execute after user gives input to the system. Also, it should perform as moderate as there isn’t any error that can cause by user’s activities.

**NR2. Constraints**

All code development shall be done with the Java language.

**NR2.1 Time Constraints**

This system must be implemented until June 9.

**NR3. Usability**

**NR3.1 User Interface**

Implement like the WinMerge(http://www.winmerge.org) interface.

**6. Requirement Dependency Traceability**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **UC1** | **UC2** | **UC3** | **UC4** | **UC5** | **NR1** | **NR2** | **NR3** |
| **UC1** |  |  |  |  |  |  |  |  |
| **UC2** | **o** |  |  |  |  |  |  |  |
| **UC3** | **o** | **o** |  |  | **o** |  |  |  |
| **UC4** | **o** |  |  |  |  |  |  |  |
| **UC5** | **o** |  |  | **o** |  |  |  |  |
| **NR1** |  |  |  |  |  |  |  |  |
| **NR2** |  |  |  |  |  |  |  |  |
| **NR3** |  |  |  |  |  |  |  |  |

**7. Development and Target Platforms**

1. Windows 10 Operating System.
2. Intel i5 processors.
3. Eclipse Oxygen IDE.

**8. Project Glossary**

**Merge** : Compare two text blocks from left to right (or from right to left)

**9. Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | 1.0 | 1.1 |  |
| Name(s) | J.Park, S.Kim, H.Lee, H.Joo,  J.Jeon | J.Park, S.Kim, H.Lee, H.Joo,  J.Jeon |  |
| Date | May, 8, 2018 | May,13,2018 |  |
| Change Description | Introduction, UseCases | System Sequence Diagram, Requirement Dependency Traceability |  |