**Simple-Merge Requirements Specification**

**Version 1.0**

**May 9, 2018**

**Project Team:**

Soyeon Kim

Sujin Kim

Eungyeong Kim

Yuseon Nam

Seungpyo Son

Shinyeong Yun

**Document Author(s):**

Sujin Kim

Yuseon Kim

**Customer Representative(s):**

Michael Jackson

1. **Introduction**

The goal of this project is to create a Java-Version of *SimpleMerge.* Its main functionality is to compare and merge the files.This program provides several features such as viewing, editing, and saving files. This document describes the requirements of this program.

1. **Use Cases**

UC1 Load File

UC2 Edit File

UC3 Save File

UC4 Compare Files

UC5 Merge Files

UC6 Back to View Mode

UC7 Traverse between Differences

**UC1 Flow of Events for the *Load File* Use Case**

* 1. **Preconditions:**

None.

* 1. **Main Flow:**

If the user presses the “Load”button[E1][E2][E3], the program will allow the user to choose a file in the file system, load the contents of the file, and displays the contents in the corresponding edit panel.

* 1. **Subflows:**

MNMNMNMNMNMNMNMWNMNWMWNMWNMWNMWNMWNMWNM

* 1. **Alternative Flows:**

[E1] 다른 프로그램에서 파일이 이미 열려있음 -> read only는 괜찮을듯 ?

[E2] 패널에 이미 파일 A가 열려 있고 수정을 했는데, Load를 하고자 하면 기존에 열려 있던 파일을 저장할 건지 묻는 Dialog가 나타난다.

[E3] 패널에 이미 파일 A가 열려 있고 수정이 안되어있고, 같은 패널에서 Load를 하고자 하면 기존 Content를 무시하고 파일을 Load한다.

~~[E4] 왼쪽 패널에 이미 파일 A가 열려 있는데, 오른쪽 패널에서 파일 A를 Load하고자 하면, 못하게 한다.~~

**UC2 Flow of Events for the *Edit File* Use Case**

**2.1 Preconditions:**

1. A file is loaded an the contents of if is displayed in the corresponding edit panel.

**2.2 Main Flow:**

If the user presses a “Edit” button, then the program allows the user to edit the strings shown in the edit panel.

**2.3 Subflows:**

MNWMWNMNWNWMNNMWWNMWNWMNWMWNMWNWMWNMW

**2.4 Alternative Flows:**

[E1] edit 이미 누른 상태에서 또 누름

[E2] 양쪽에 같은 파일이 열려 있고, 오른쪽에서는 수정 중인 상태에서 왼쪽 패널의 “Edit” 버튼을 누르면, 못하게 한다.

**UC3 Flow of Events for the *Save File* Use Case**

**3.1 Preconditions:**

1. A file is loaded and the contents of if is displayed in the corresponding edit panel.

**3.2 Main Flow:**

If the user presses a “Save” button, the the program saves the edited content into the file.

**3.3 Subflows:**

WNMWNMWNMWNMWNWMWMNWMNWMNWMNWMWNMWNMW

**3.4 Alternative Flows:**

None.

**UC4 Flow of Events for the *Compare Files* Use Case**

**4.1 Preconditions:**

1. Two files are loaded and the contents are displayed in the window.

2. Two files have not been edited since the last save.

3. Both panels are in view mode.

**4.2 Main Flow:**

If the user presses a “Compare” button, then the program displays the different lines with a colored font/background. The comparison is done line by line. The user is able to traverse the blocks indicating the differences.

**4.3 Subflows:**

WNMWNWMNWMWMNMWNWMNWMNWMWNMWNWWNMNWNM

**4.4 Alternative Flows:**

[E1]

[E2]

**UC5 Flow of Events for the *Merge Files* Use Case**

**5.1 Preconditions**

1. The program is in Compare Mode.

**5.2 Main Flow:**

If the user presses “Copy to Right” or “Copy to Left” button, the program copies the selected block in the corresponding panel to the file shown in the other panel.

**5.3 Subflows:**

WNMWNMWNWMNWMNWMWNWMMNWNWMNWMNWMNWMN

**5.4 Alternative Flows:**

[E1]

**UC6 Flow of Events for the *Back to View Mode* Use Case**

**6.1 Preconditions**

1. The program is in Edit Mode.

**6.2 Main Flow:**

If the user presses “View Mode” Button, the program protects the file from any modification.

**6.3 Subflows:**

WMWMWMWMMWMWMMWMWMWMMWMWMWMWMWMWM

**6.4 Alternative Flows:**

[E1] If the file has not been saved after the last edit, the program shows a dialog and asks the user to save the file. If the user rejects to do so, the program remains in Edit Mode.

**UC7 Flow of Events for the *Traverse between Differences* Use Case**

**7.1 Preconditions**

1. The program is in compare mode.

**7.2 Main Flow:**

If the user presses a “Next” or “Previous” button, the program scrolls to and shows the corresponding block of difference.

**7.3 Subflows:**

[S1] After the user selects a block to copy, the user may copy the specific block to the other file[UC5].

**7.4 Alternative Flows:**

None.

1. **Misuse Cases**
2. **Nonfunctional Requirements**

**NR1. Performance**

The system shall wait for all user inputs, and execute only the necessary functions given a user input to the system. All functions shall be completed quickly.

NR1.1 User response

The system shall respond to any user input within 0.01 seconds.

**NR2. Usability**

A user shall be able to determine quickly what user options they have to perform.

NR2.1 User options

A user shall only have access to functionality that is allowed to them at a given time.

NR2.2 User Interface

The system shall allow a user to interface with it through mouse events on buttons and drop down boxes and keyboard events on text fields. The amount of user keyboard input shall be minimized by the system other than editing files, or typing file names.

NR2.3 User Errors

The system shall catch improper input rom all text fields in the system.

**NR3. Reliability**

**NR4. Supportability**

**~~NR5. Syntax Highlighting~~** ~~여기 적는게 아닌거같아..ㅠㅠ~~

1. **Constraints**

All code development shall be done with the Java programming language.

All testing shall be done using JUnit and EasyMock. Therefore, GUI components should be designed by using MVC architecture pattern.

1. **Requirements Dependency Traceability Table**
2. Windows 10 Operating System
3. BLAHBLAHBLAH Processors
4. Eclipase IDE
5. **Development and Target Platforms**
6. **Project Glossary**

View Mode – the user can view the file, but is prohibited from editing the content.

Edit Mode – the user can edit the file.

Comparing Mode – state after pressing “compare” button. The user can merge selected blocks, however the user cannot edit the content manually.

1. **Document Revision History**

|  |  |
| --- | --- |
| Version | 1.0 |
| Name(s) | Sujin Kim and Yuseon Nam |
| Date | May 9, 2018 |
| Change Description | Original creation of the SRS. |