**Software Requirement Specification(SRS)**

**Simple Merge Specification**

**Project Team**

**<4조>**

**20131961 김민조, 20132569 조성주, 20132862 김현빈**

**20134220 우정현, 20134695 김명규, 20162379 김성연**

**목차**

1. **소개(Introduction)**

1.1 목적(Purpose)

1.2 참조(Reference)

1.3 독자 대상(Intend Audience)

1. **전체 설명(Overall Description)**
   1. 제품 기능(Product Feature)
   2. 운영 환경(Operation Environment)
2. **유스케이스(Use Case)**
3. **시스템 시퀀스 다이어그램(System sequence diagrams)**
4. **외부 인터페이스 요구사항(External interface Requirement)**
   1. 사용자 인터페이스(User Interface)
   2. 소프트웨어 인터페이스(Software Interface)
5. **기능 이외의 다른 요구사항(Other Nonfunctional Requirement)**
   1. 성능 요구사항(Performance Requirement)
   2. 성능(Performance)
   3. 사용성(Usability)
6. **부록(Appendix)**
7. **소개 (Introduction)**
   1. **목적(Purpose)**

이 프로젝트의 목적은 파일을 비교하고 병합하는 것이 주요 기능인 Simple Merge을 구현하는 것이다.

* 1. **참조(Reference)**

WinMerge, WinDiff, BeyondCompare와 같은 유사한 현존하는 제품을 참조한다.

* 1. **독자 대상(Intend Audience)**

이 SRS 문서는 중앙대학교 ICT 소프트웨어공학 팀 프로젝트를 위해서 작성되었고, 팀 프로젝트에 참여한 팀원들과, 교수님 그리고 그 외 강의 수강생들이 독자 대상이다.

1. **전체 설명(Overall Description)**
   1. **제품 기능(Product Feature)**
2. **두개의 편집 패널과 버튼**

두개의 편집 패널이 메인 윈도우에 표시된다.

각 편집 패널이 메인 윈도우에 표시되며, 각 편집 패널 위에는 “불러오기”, “편집”, “저장” 버튼이 존재한다.

메인 윈도우에는 “비교”, “병합” 버튼이 존재한다.

1. **파일의 불러오기 / 편집 / 저장 기능**

불러오기: 시스템에서 파일 내용을 로드하고 해당 패널에 내용을 표시한다.

편집: 버튼을 누르면 해당 패널 내용을 편집상태가 되고, 버튼을 다시 누르면 편집상태가 끝난다.

저장: 해당 패널 내용을 시스템에 저장한다.

1. **두개의 파일을 비교하는 기능**

비교: 버튼을 누르면 라인 별로 지정된 글꼴과 색으로 두 파일을 비교한다.

비교 결과는 “diff” 프로그램에서 생성된 결과와 같다.

1. **두개의 파일을 병합하는 기능**

병합: 두 가지의 기능이 존재한다.

“CopytoRight”는 왼쪽 패널에서 선택된 블록을 오른쪽 패널에 표시된 파일로 복사한다.

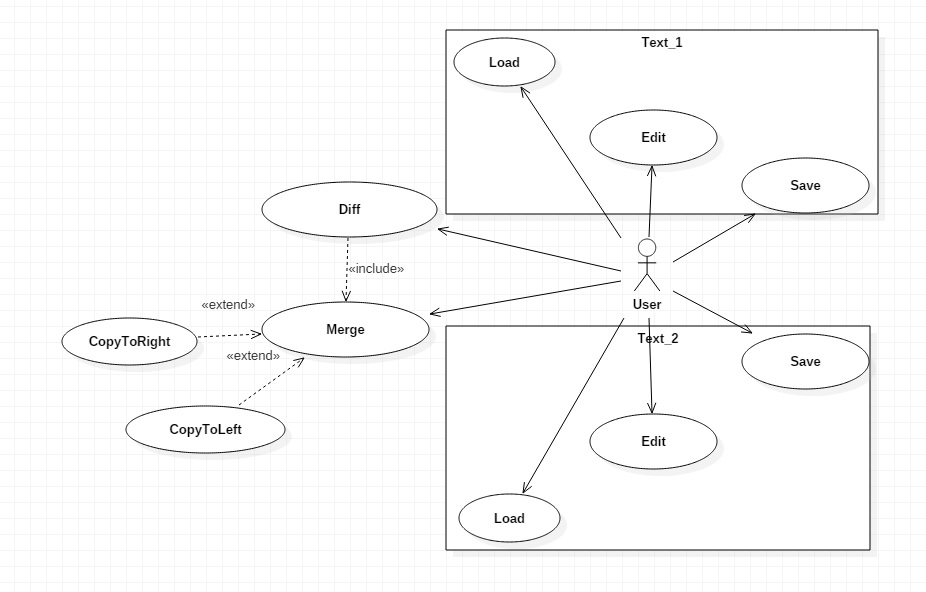
“CopytoRight”는 오른쪽 패널에서 선택된 블록을 왼쪽 패널에 표시된 파일이 복사된다.

* 1. **운영 환경(Operation Environment)**

Java Window form을 이용한 개발

1. **유스케이스(Use Case)**

**3-1. Use-Case Diagram**



**UC1 Load**

**UC2 Edit**

**UC3 Save**

**UC4 Diff**

**UC5 Merge**

**UC6 CopytoRight**

**UC7 CopytoLeft**

**3-2. Use-Case Flows**

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

* 1. **Preconditions:**

텍스트 파일이어야 하며, 편집 상태가 아니어야한다.

* 1. **Main Flow:**

프로그램을 실행했을 때 제일 먼저 활성화 되는 기능으로서, 프로그램이 실행되면 2개의 패널 위쪽에 로드 버튼이 활성화된다. 로드 버튼을 클릭시 파일 선택창이 나타나 사용자가 파일[E1]을 선택해 온다.

* 1. **Subflows:**

없음.

* 1. **Alternative Flows:**

[E1] 파일은 텍스트 파일이어야 한다. 만약 텍스트 파일이 아닌 경우, 아무것도 로드 되어 오지 않는다. 파일이 로드되어 있지 않은 패널도 편집이 가능하게 편집 버튼이 활성화 되도록 한다.

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

* 1. **Preconditions:**

패널에 파일이 로드된 [UC1] 상태여야 한다.

* 1. **Main Flow:**

편집 버튼을 누르면 파일의 내용을 편집한다. 각각의 패널은 편집 기능이 따로 적용되며, 한쪽이 편집된다고 다른 한쪽에 영향을 끼치지 않는다.

* 1. **Subflows:**

없음.

* 1. **Alternative Flows:**

좌우 패널에서 편집상태일 때 save, load, merge, diff가 모두 비활성화된다.

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

**3.1 Preconditions:**

편집 상태가 아니어야 한다.

**3.2 Main Flow:**

각각의 패널에 저장 버튼이 있으며, 각 패널의 저장 버튼을 누르면 그 해당 패널의 내용이 저장된다.

**3.3 Subflows:**

없음.

**3.4 Alternative Flows:**

없음.

**UC4 Flow of Events for the Diff Use Case**

**4.1 Preconditions:**

편집 상태가 아니어야 한다.

**4.2 Main Flow:**

비교 버튼을 클릭시 LCS(Longest Common Subsequence) 알고리즘을 사용하여 두 파일을 비교 분석하고, 그 차이점을 하이라이트로 표시된다.

**4.3 Subflows:**

비교 버튼을 누르면 비교된 내용이 지정된 글꼴과 배경으로 파일 내용 창에 표시된다.

**4.4 Alternative Flows:**

두 파일이 차이점이 없으면, 아무 변화가 없다.

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

**5.1 Preconditions:**

양쪽의 패널에 파일이 로드된 상태여야 하고[UC1], 비교가 실행된 상태여야 한다.[UC4]

**5.2 Main Flow:**

비교가 된 상태일 때, 각 패널의 차이가 하이라이트 표시되어 있는 상태이다. 병합 버튼은 2가지가 있으며, CopytoRight 버튼과 CopytoLeft 버튼으로 나뉜다. 차이를 나타내는 블록을 CopytoRight 버튼과 CopytoLeft 버튼을 통해서 한쪽의 차이점인 부분을 반대쪽 패널로 병합한다.

**5.3 Subflows:**

한 패널의 파일 내용을 반대쪽 패널로 덮어 씌운다.

* 1. **Alternative Flow:**

없음.

**UC6 Flow of Events for the CopytoRight Use Case**

**6.1 Preconditions:**

왼쪽 파일이 로드된 후[UC1], 비교된 상태가 되어야 한다.[UC4]

**6.2 Main Flow:**

CopytoRight 버튼을 누르면 왼쪽 패널에서 선택한 블록을 오른쪽 패널에 표시된 파일로 복사한다.

**6.3 Subflows:**

없음.

**6.4 Alternative Flow:**

없음.

**UC7 Flow of Events for the CopytoLeft Use Case**

**7.1 Precondtions:**

오른쪽 파일이 로드된 후[UC1], 비교된 상태가 되어야 한다.[UC4]

**7.2 Main Flow:**

CopytoLeft 버튼을 누르면 오른쪽 패널에서 선택한 블록을 왼쪽 패널에 표시된 파일로 복사한다.

**7.3 Subflows:**

없음.

**7.4 Alternative Flow:**

없음.

1. **시스템 시퀀스 다이어그램(System sequence diagrams)**

첨부파일 참조

1. **외부 인터페이스 요구사항(External Interface Requirement)**

**4.1 사용자 인터페이스(User Interface)**

프로그램은 사용자에게 keyboard, mouse, button event들을 허용해야만 한다.

**4.2 소프트웨어 인터페이스(Software Interface)**

Junit 및 EasyMock을 이용하여 프레임워크 테스트를 한다. MVC Architecture 기반으로 설계한다.

**5. 기능 이외의 다른 요구사항(Other Nonfunctional Requirement)**

**5.1 성능 요구사항(Performance Requirement)**

JAVA를 이용하기 때문에 최신버젼의 JDK 필요

**5.2 성능(Performance)**

Simple Merge 프로그램의 주요기능인 “diff”와 “merge”가 빠른 시간 내에, 신속하게 작동되도록 한다.

**5.3 사용성(Usability)**

사용자가 Simple Merge 프로그램을 잘 사용하기 위해서 텍스트가 잘 보여야 되며, 텍스트 파일들을 수정할 수 있어야 한다.

1. **부록(Appendix)**

**부록 A: 용어집(Glossary)**

|  |  |
| --- | --- |
| **용어** | **설명** |
| LCS(Longest Common Subsequence) | 최장 공통 부분 문자열로 불리며 두개의 텍스트 파일을 비교할 때 많이 쓰이는 알고리즘이다. |
| MVC | Model, View, Controller의 약자이며, model은 프로그램의 데이터와 정보를 나타내고, view는 화면(GUI)인 사용자 인터페이스를 나타내고, controller는 model과 view간의 상호작용하게 하는 역할을 나타냄으로써 소프트웨어 공학에서 사용되는 소프트웨어 디자인 패턴이다. |
| JUnit | 자바 언어에서 사용되는 유닛 테스트 프레임워크이다. |

**부록 B: 분석 모델(Analysis Model)**

Use Case diagram, class diagram 등과 같은 관련된 분석 모델을 설명한다.

**부록 C: 문제 목록(Issues List)**