EMB00000aec676b

SRS(수정본)

|  |  |
| --- | --- |
| 과목 | 소프트웨어 공학 |
| 학과 | 컴퓨터공학부 |
| 팀 | 09 |
| 팀원 | 백인혁 20141266  백찬희 20142503  이동규 20140490  함지훈 20146512 |

**목차**

1. **Introduction**
2. **Use Cases**

: UC 1 ~ 9

1. **Misuse Cases**

: UC 10

1. **Nonfunctional Requirements**

: Usability, Reliability, Performance, Suportability, Constraints

1. **Requirements Dependency Traceability Table**
2. **Development and Target Platfroms**
3. **Project Glossary**

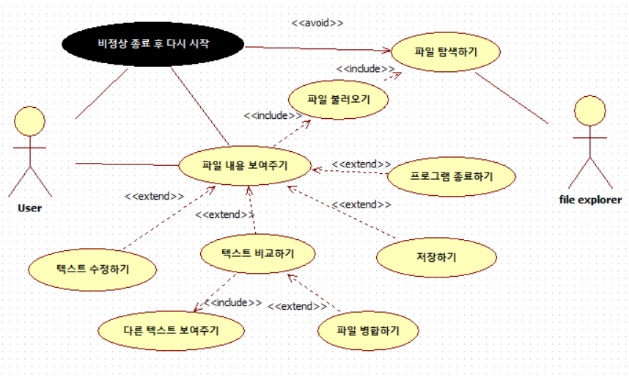
: View, Load, Compare, Merge, Save, Edit, Etc..

1. **Document Revision History**

**1. Introduction**

|  |  |
| --- | --- |
| UC1 | 파일 내용 보여주기 |
| UC2 | 파일 탐색하기 |
| UC3 | 파일 불러오기 |
| UC4 | 텍스트 수정하기 |
| UC5 | 텍스트 비교하기 |
| UC6 | 다른 텍스트 보여주기 |
| UC7 | 파일 병합하기 |
| UC8 | 저장하기 |
| UC9 | 프로그램 종료하기 |
| UC10 | 저장하지 않고 프로그램 강제 종료하기 |

Development of Simple Merge 프로젝트의 목표는 두 개의 다른 텍스트 파일을 비교하여 다른 부분을 찾아서 병합(수정) 할 수 있게 하는 것이다. 이를 위해서 프로젝트에서 텍스트 파일과 관련된 기능들을 제공할 것이다. 그 기능에는 ‘불러오기’, ‘저장하기’, ‘수정하기’, ‘비교하기’, ‘병합하기’ 등이 있다. 사용자가 비교하고자 하는 파일을 불러오면 두 개의 텍스트를 비교하여 서로 다른 부분을 글자색, 배경색, 글꼴(강조) 등으로 표시해줄 것이다. 사용자는 판단하여 둘 중 하나의 텍스트 파일로 덮어쓸 수 있다. 이 프로젝트는 JAVA 프로그램 언어를 사용하여 개발될 것이며 자세한 기능 및 시나리오는 다음 목차에서 설명할 예정이다.

****

**2. Use Cases**

**UC1 파일 내용 보여주기**

1. **Preconditions:**

파일 불러오기를 완료한 상태

1. **Main Flow:**

프로그램이 시작되면 비교할 파일을 불러오도록 유도한다.

비교할 파일을 불러오면 텍스트 뷰에 해당 파일의 내용을 보여준다.

1. **Subflows:**

None

1. **Alternative Flows:**

파일의 읽기 권한이 없을 시 권한이 없다는 Dialog를 띄운다.

**UC2 파일 탐색하기**

1. **Preconditions:**

사용자가 LOAD버튼을 눌러야 한다.

1. **Main Flow:**

사용자가 LOAD버튼을 눌러서 파일을 탐색해서 파일을 선택한다 [E1].

1. **Subflows:**

None

1. **Alternative Flows:**

[E1] 탐색해서 선택한 파일이 텍스트 파일이 아닌 경우에는 선택한 파일이 텍스트 파일이 아니라고 한 뒤에 불러오지 않는다.

**UC3 파일 불러오기**

1. **Preconditions:**

파일 탐색하기를 통해서 텍스트 파일을 탐색을 완료한 상태

1. **Main Flow:**

탐색해서 불러온 파일에 내용을 해당하는 패널에 불러온다 [S1][E1].

1. **Subflows:**

[S1] 패널에 파일을 불러올 때 줄 단위로 해서 불러온다.

1. **Alternative Flows:**

[E1] 패널에 탐색해서 불러온 파일 말고 먼저 불러온 파일이 있을 경우에는 먼저 불러왔던 파일을 저장 후 종료 할 것인지 저장 안하고 종료 할 것인지 불러오는 것을 그만 둘 것인지 물어본다.

**UC4 텍스트 수정하기**

1. **Preconditions:**

1) 파일을 불러와서 패널에 내용이 있어야 함

2) 사용자가 EDIT버튼을 눌러야 한다

1. **Main Flow:**

사용자는 EDIT버튼을 통해서 패널에 불러온 내용을 수정한다 [E1].

1. **Subflows:**

없음

1. **Alternative Flows:**

[E1] 패널에 불러온 내용이 없을 경우에는 수정할 수 없음

**UC5 텍스트 비교하기**

1. **Preconditions:**

1) 프로그램이 텍스트 파일에 대하여 읽기 권한이 있어야한다.

2) 사용자가 두 개의 텍스트 파일을 불러와서 등록 되어있어야 한다.

3) 사용자가 비교하기 버튼을 눌러야 한다.

1. **Main Flow:**

사용자가 비교를 희망하면 텍스트 파일을 등록 후 비교하기 버튼을 누른다.

용도에 따라서 한 줄 씩 비교할 수 있고 한 번에 비교할 수 있다.

사용자가 비교하기 버튼을 눌렀다면 두 텍스트의 내용을 String으로 한 줄 씩 받아온다.

두 개의 String을 비교하여 동일한지 확인한다. 만약 동일하지 않다면 다른 부분을 찾아서 저장한다.

1. **Subflows:**

[S1] 사용자가 버튼을 누를 때마다 텍스트를 한 줄 씩 읽으면서 비교한다.

[S2] 텍스트를 한 번에 읽어서 전부 비교한다.

1. **Alternative Flows:**

사용자가 비교를 요청했을 때 두 텍스트 파일의 내용이 동일하여 다른 부분을 찾지 못했을 때는 동일한 텍스트 파일이라는 것을 Dialog로 띄어서 알려준다.

**UC6 다른 텍스트 보여주기**

1. **Preconditions:**

1) 프로그램이 텍스트 파일에 대하여 읽기 권한이 있어야한다.

2) 사용자가 두 개의 텍스트 파일을 불러와서 등록 되어있어야 한다.

3) 사용자가 비교하기 버튼을 눌러야 한다.

1. **Main Flow:**

사용자가 텍스트 비교를 위하여 비교하기 버튼을 눌렀을 때 다른 부분이 있다면 비교하기에서 찾은 부분들을 글꼴의 서식을 바꾸어서 표시해준다.

이 때 한 쪽의 텍스트 뷰에만 표시해주는 것이 아닌 양쪽의 텍스트 뷰에 전부 표시를 해준다.

1. **Subflows:**

[S1] 텍스트의 다른 부분을 체크하면서 하나씩 순차적으로 보여준다.

[S2] 모든 비교를 끝내고 한번에 다른 부분을 보여준다.

1. **Alternative Flows:**

텍스트를 비교했을 때 다른 점이 없다면 텍스트의 글꼴을 변경하기 않는다.[UC5]

**UC7 파일 병합하기**

1. **Preconditions:**

1) 사용자가 두 개의 텍스트 파일을 불러와서 등록 되어있어야 한다.

2) 병합하기 전 미리 비교를 해야 한다.

3) 사용자로부터 병합하는 파일의 기준점을 입력 받아야 한다.(Left to Right or Right to Left)

1. **Main Flow:**

사용자가 두 개의 텍스트 파일을 불러왔을 때 왼쪽에서 오른쪽 또는 오른쪽에서 왼쪽의 텍스트 파일로 병합을 할 수 있어야한다. 버튼이 2개가 있는 데 'Copy to left' 또는 'Copy to right' 중 어느 버튼이 눌리는 지에 따라 복사되는 방향이 정해지며 수정이 된다.

사용자가 병합을 시도할 때 정말 ~에서 ~로 병합을 할 것인지 확인을 해주는 Dialog를 띄어준다. 이후 OK를 입력받았을 때 병합을 시도한다.

1. **Subflows:**

[S1] 비교하기로 구한 다른 부분을 하나씩 체크하며 병합할 것인지 확인하고 하나씩 병합한다.

[S2] 다른 모든 부분 또는 파일 전체를 한번에 병합한다.

1. **Alternative Flows:**

텍스트 파일 비교 결과 동일하다면 동일하여 병합하지 않는 다는 Dialog를 띄어준다.

**UC8 저장하기**

1. **Preconditions:**

‘파일 내용 보여주기’를 완료한 상태이다.

1. **Main Flow:**

사용자가 저장버튼을 누름으로써 현 상태의 텍스트 파일을 저장한다.

1. **Subflows:**

없음

1. **Alternative Flows:**

없음

**UC9 프로그램 종료하기**

1. **Preconditions:**

‘파일 내용 보여주기’를 완료한 상태이다.

1. **Main Flow:**

사용자가 종료버튼을 누름으로써 프로그램을 종료한다[E1 – E3].

1. **Subflows:**

없음

1. **Alternative Flows:**

[E1] 파일내용이 수정된 후 저장되어 있지 않다면, 저장하고 종료할 건지, 저장하지 않고 종료할 건지 물어보는 Dialog가 뜨게 되고 그중 Yes를 클릭한다. 그러면 파일은 저장되고, 프로그램은 종료된다.

[E2] 파일내용이 수정된 후 저장되어 있지 않다면, 저장하고 종료할 건지, 저장하지 않고 종료할 건지 물어보는 Dialog가 뜨게 되고 그중 No를 클릭한다. 그러면 파일은 저장되지 않고, 프로그램은 종료된다.

[E3] 파일내용이 수정된 후 저장되어 있지 않다면, 저장하고 종료할 건지, 저장하지 않고 종료할 건지 물어보는 Dialog가 뜨게 되고 그중 Cancel을 클릭한다. 그러면 프로그램은 종료되지 않고 종료버튼을 누르기 전 상태로 되돌아간다.

**3. Misuse Cases**

**UC10 저장하지 않고 프로그램 강제 종료하기**

1. **Preconditions:**

**1) 프로그램이 비정상적으로 종료되어 가장 최근 저장 이후 변경 내용을 저장하지 않고 종료된 상태이다**

**2). 가장 최근 저장파일이 항상 특정한 위치에 저장되고 있어야 한다.**

1. **Main Flow:**

**사용자가 파일을 불러오고 텍스트에 작업을 수행하고 있던 도중 비정상적으로 프로그램이 종료됨 이런 상황에서 프로그램을 다시 시작하게 되면 가장 최근 저장했던 작업내용을 보여준다.**

1. **Subflows:**

**없음**

1. **Alternative Flows:**

**가장 최근 저장했던 작업내용이 없을 시 새로운 파일을 불러오고 내용을 보여준다.**

**4. Nonfunctional Requirements**

**Usability**

**Reliability**

사용자가 주어진 기능이 아닌 잘못된 기능을 실행해서 시스템이 실행을 실패한 경우 경고 문고를 띄워서 실패했다고 알리고 빠르게 복구해서 시스템에 신뢰도를 높여야 한다.

**NR2.1 텍스트 파일이 아닌 잘못된 파일을 불러오는 경우**

텍스트 파일이 아닌 잘못된 파일을 불러오는 경우에는 텍스트파일이 아니라는 경고 문고를 띄우고 파일을 불러오지 않고 원상태로 돌아가야 함

**NR2.2 파일을 저장하지 않고 강제 종료를 하는 경우**

파일을 불러온 뒤에 수정을 하였으나 저장하지 않고 강제종료를 하는 경우 프로그램을 다시 실행 시 수정한 파일을 불러오도록 유도

**Performance**

프로그램이 진행하기 위해서는 텍스트 파일에 대한 읽기,쓰기 권한이 필요하다. 또한 동시에 수정과 저장을 하는 등 다중 작업을 시행해서는 안되기 때문에 한 번에 하나의 업무만 처리해야한다. 각 작업을 수행하는 데 있어서 긴 시간을 보내게 된다면 다중 작업을 지원하지 않기 때문에 사용자가 오래 기다려야한다. 따라서 가능한 모든 작업을 1초 내외로 처리하도록 한다.

**NR3.1 File I/O**

File I/O에는 텍스트 파일의 크기에 따라 시간의 차이가 많이 생긴다. Time Complexity가 O(n^2)을 넘지 않도록 하여 Runtime이 급격하기 증가하지 않도록 제한한다.

**NR3.2 텍스트 비교**

텍스트를 비교할 때도 마찬가지로 텍스트 파일만큼의 시간이 걸린다. 불러왔다면 전체를 비교하는 데 1초를 넘기지 않도록 구현한다.

**NR3.3 File Permission**

텍스트 파일을 불러오고 저장하는 데 알맞은 권한이 필요하다. 각 액션에 따라서 맞는 권한을 지정해줘야한다.

**Suportability**

**Constraints**

1. 코드는 JAVA Programming Language로, 구현환경은 Eclipse에서 구현할 예정.
2. 코드 테스트는 JUNIT환경에서 실시할 예정.

**5. Requirements Dependency Traceability Table**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | UC1 | UC2 | UC3 | UC4 | UC5 | UC6 | UC7 | UC8 | UC9 | UC10 | NR2.1 | NR2.2 | NR3.1 | NR3.2 | NR3.3 |
| UC1 |  |  | **X** |  |  |  |  |  |  |  |  |  |  |  |  |
| UC2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC3 |  | **X** |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC4 | **X** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC5 | **X** |  |  |  |  | **X** |  |  |  |  |  |  |  |  |  |
| UC6 | **X** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC7 | **X** |  |  |  |  | **X** |  |  |  |  |  |  |  |  |  |
| UC8 | **X** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC9 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| UC10 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| NR2.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| NR2.2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| NR3.1 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| NR3.2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| NR3.3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

**6. Development and Target Platfroms**

1. Windows(7, 8, 8.1, 10) Operating System

2. Intel & AMD processors

3. JAVA Eclipse IDE

4. GitHub

**7. Project Glossary**

1. view : 텍스트를 보여주는 패널/창을 의미한다.

2. Load : 텍스트 파일을 불러온다.

3. Compare : Load 된 2개의 텍스트 파일을 서로 비교한다.

4. Merge : Compare결과를 가지고 서로 다른 부분들을 어느 한쪽을 기준으로 서로 파일의 내용을 동일하게 만드는 작업을 의미한다.

\* 결과적으로 Merge를 실행하면 왼쪽과 오른쪽 파일내용이 같아져야 한다.

5. Save : 작업한 파일의 내용을 저장한다.

6. Edit : 텍스트 파일의 내용을 수정할 수 있게 된다. (추가/확인필요)

**8. Document Revision History**

|  |  |
| --- | --- |
| 버전 | 1.1 |
| 이름 | 소프트웨어 공학 9팀 |
| 날짜 | 2018년 5월 13일 |
| Change Description | Original creation of the SRS |