Human ICT Software Engineering

**Final Report**

Simple Merge

20132682 김현빈

20132569 조성주

20134695 김명규

20134220 우정현

20131961 김민조

20162379 김성연

**목차**

1. **Program Description**

1) 개요

2) 주요 기능

1. **Development Environment**
2. Development OS
3. Development Tool
4. **Design & Implementation**
5. Development Process Model
6. MVC Pattern
7. Model Design
8. View Design
9. Observer Design
10. Util Design
11. Class Diagram
12. **Object-Oriented Design Principles**
13. OOP Principles
14. SOLID Principles
15. **Test**
16. Functional Unit Test Cases
17. GUI Test Case
18. **Execution Screen & Explanation**
19. **Project Management**
20. Program Description
21. 개요

이 프로젝트의 목표는 텍스트 파일 2개를 불러와서 비교하고 서로 다른 부분들을 줄 단위로 구분 해주며, 줄 단위로 복사 할 수 있으며, 불러온 텍스트 파일들을 창에서 편집할 수 있으며, 저장까지 할 수 있는 ‘Simple Merge’ 프로그램을 개발 하는 것이다.

1. 주요 기능

**i. load**

텍스트 파일을 불러오는 기능이다..

**ii. edit**

불러온 텍스트를 편집하거나 빈 창에서 텍스트를 작성 할 수 있는 기능이다.

**iii. diff**

두 텍스트 파일들을 줄 단위로 서로 다른 부분들을 색칠해서 표시해주는 기능이다.

**iv. copy**

‘diff’를 사용하여 나타낸 서로 다른 부분이 있는 줄을 위에서부터 차례대로 왼쪽텍스트를 오른쪽에 붙여넣기 하던가, 오른쪽텍스트를 왼쪽에 붙여넣기 하는 기능이다.

**v. save**

편집된 텍스트 파일을 저장하는 기능이다.

1. Development Environment
2. Development OS

프로젝트를 개발 할 때 사용 했던 운영체제는 window7, window10을 중점적으로 사용하였다

1. Development Tool

JAVA 구현하였고 이에 사용된 버전은 1.8.0\_144 버전이다. GUI는 JAVA swing/awt를 사용하였고, ~~Junit, EasyMock으로 Unit 테스트를 했다~~.

팀 프로젝트의 효율을 높이기 Git을 사용하였고 Git HUB를 이용하여 팀원들과 프로젝트 공유를 효율적으로 하였다.

본 문서는 Microsoft Word를 이용하여 작성하였다. SRS와 기타 문서들은 md를 이용하여 gitub에서 작성하였다.

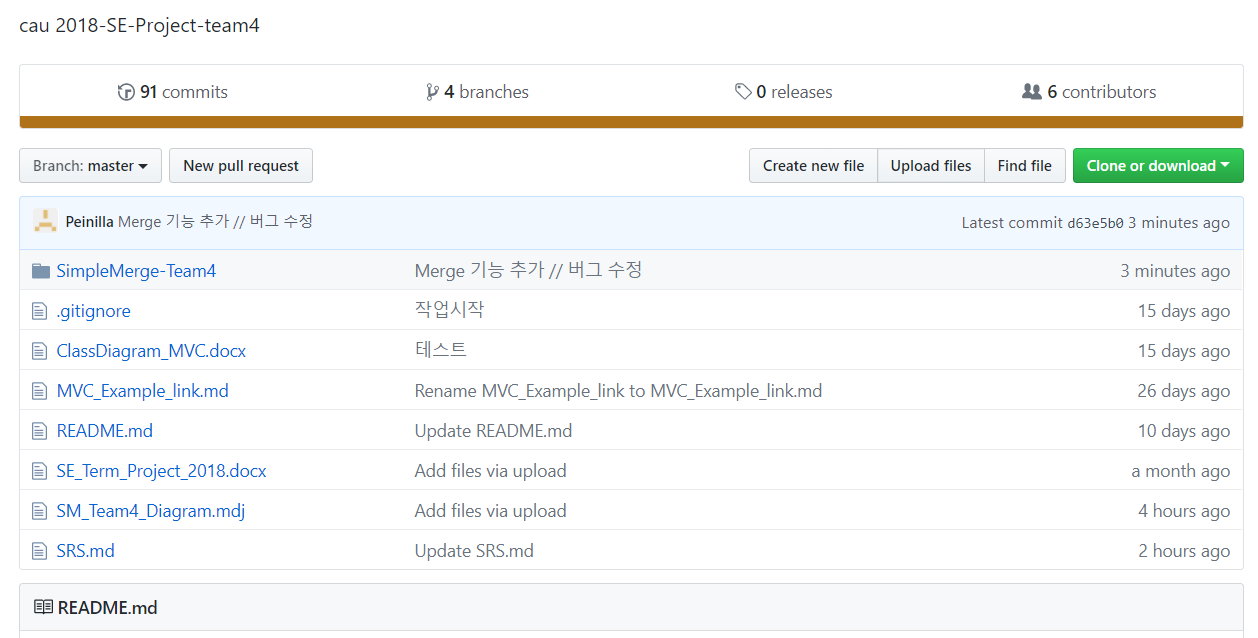


그림 1. Github

1. Design & Implementation
2. Development Process Model

우리 팀은 목표로 하는 프로그램의 기능과 요구사항이 과제로서 명확하기에, 프로그램의 변화가 필요 없을 것으로 생각하여 폭포수모델(Waterfall Model)을 사용했다. 폭포수모델의 방식대로 요구사항 분석, 소프트웨어 설계, 소프트웨어 구현, 소프트웨어 시험, 소프트웨어 유지보수의 단계로 진행하였다.

먼저 요구사항 분석 단계에서는 우리가 프로그래밍해야 하는 프로그램의 주요기능들을 분석했다. 분석한 요구사항들을 알아보기 쉽게 문서화해서 SRS를 작성했다.

소프트웨어 설계 단계에서는 MVC 아키텍처 패턴에 맞춰 프로그램을 설계했다.

소프트웨어 구현 단계에서는 ………..

소프트웨어 시험 단계에서는 다양한 방법으로 프로그램의 테스트를 해보면서 이상한 버그가 없는지 확인을 하였다.

소프트웨어 유지보수 단계는 우리 프로그램의 미숙한 부분들을 고쳐나가는 것으로 하겠다.

1. MVC Pattern

MVC 패턴은 응용 프로그램을 세 가지의 구성요소인 모델, 뷰, 컨트롤러로 나누는 디자인 패턴이다.

모델은 모델의 상태에 변화가 있을 때 컨트롤러와 뷰에 이를 통보한다. 이와 같은 통보를 통해서 뷰는 최신의 결과를 보여줄 수 있고, 컨트롤러는 모델의 변화에 따른 적용 가능한 명령을 추가, 제거, 수정할 수 있다.

뷰는 사용자가 볼 결과물을 생성하기 위해 모델로부터 정보를 얻어 온다.

컨트롤러는 모델에 명령을 보냄으로써 모델의 상태를 변경할 수 있다.

mvc그림

……………………..

1. Model Design
2. View Design
3. Observaer Design
4. Util Design
5. Class Diagram
6. Object-Oriented Design Principles
7. OOP Principles
8. SOLID Principles
9. Test
10. Functional Unit Test Cases
11. GUI Test Case
12. Execution Screen & Explantion
13. Project Management