Human ICT Software Engineering

**Final Report**

Simple Merge

20132682 김현빈

20132569 조성주

20134695 김명규

20134220 우정현

20131961 김민조

20162379 김성연

**목차**

1. **Project Description**
2. Summary
3. Development Environment

* Development OS
* Development Tool

1. **Analysis**
2. **Design & Implementation**
3. MVC Pattern
4. Model Design
5. View Design

* Main Frame
* Text Frame

1. Controller Design

* Diff Algorithm

1. Class Diagram
2. **Program UI and Features**
3. **Test**
4. Functional Unit Test Cases
5. GUI Test Case
6. **Execution Screen & Explanation**
7. **Project Management**
8. Program Description
9. Summary

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. Development Environment

- Development OS

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

- 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.

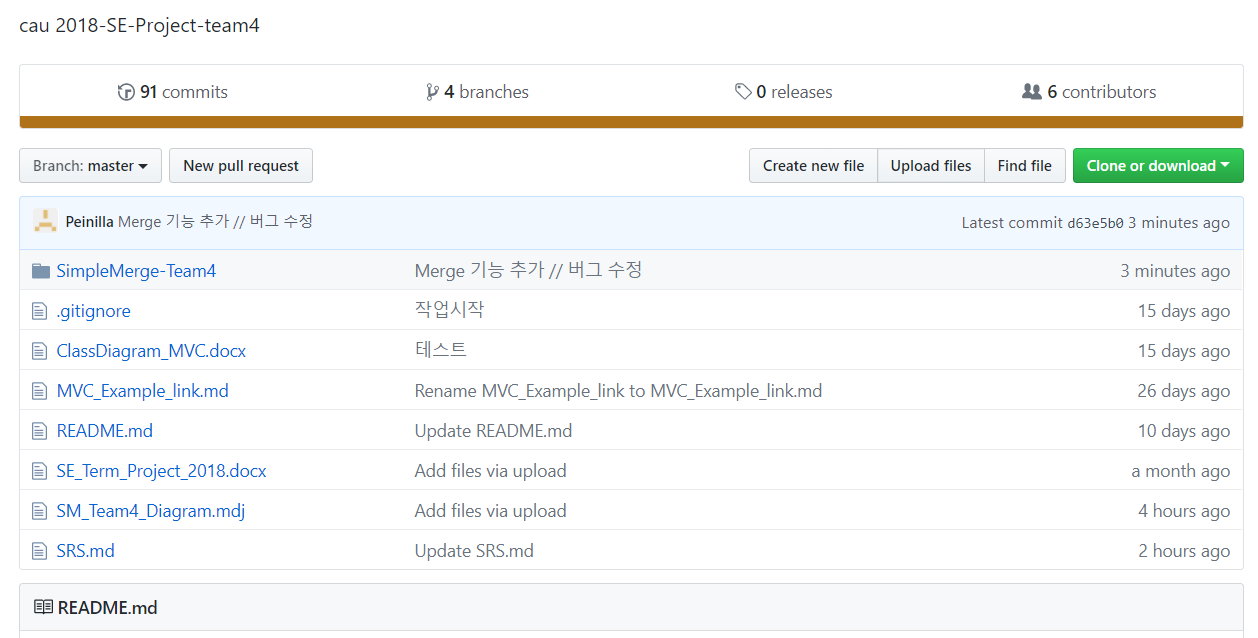


그림 1. Github

1. Analysis

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 설계 한걸 기초로 하여 코딩을 하였다.

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

소프트웨어 유지보수 단계는 우리 프로그램의 미숙한 부분들을 고쳐나가는 것으로 하겠다.

**#구현하여야 하는 기능들**

**i. load**

텍스트 파일을 불러오는 기능이다.

Eidt 상태에서는 버튼이 비활성화 되어야한다.

**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.

같은 줄은 색을 안 칠하고, 서로 다른 줄은 분홍색, 추가된 빈 줄들은 회색으로 칠할것이다.

**iv. copy**

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

Diff가 활성화 된 상태가 아니면 비활성화 되어야 한다.

Edit 상태에서도 비활성화 되어야 한다.

**v. save**

편집된 텍스트 파일을 저장하는 기능이다.

Edit 상태에서 비활성화 되어야 한다.

1. Design & Implementation

@ SM\_Model

MVC 패턴 중 모델에 속하는 클래스로, 처음 실행창이 켜졌을 때 완전 빈 텍스트 창이 뜨는걸 막기 위해서 스트링을 추가한 상태로 간다. 실행창에서 텍스트파일을 읽어오는 기능과 저장하는 기능을 구현하면서 리스트마다 줄 번호를 주는 기능까지 있는 클래스이다.

줄 번호를 주는 이유는 diff 상태에서 merge를 구현할 때 LCS 방법을 이용하기 편하기 위해 구현하였다.

@ SM\_Controller\_LCS

LCS 방법 중 Longest Common Subsequence 방법으로 구현한 diff 클래스이다.

텍스트 파일을 받아오는 프로그램을 구현하는 것이라 string을 받아와서 줄 마다 리스트로 배열을 맞춰주었다. 맞춰진 배열들을 SM\_Model 클래스에서 구현한 줄 번호를 통해서 diff를 하는데 Longest Common Subsequence 방식으로 구분하다 보니 빈칸이 생기기도 하게된다. 생기는 빈칸은 2로 구분하고 같은 줄은 1로 구분, 서로 다른 줄들은 0으로 구분하였다.

@ SM\_Controller

MVC 패턴 중 컨트롤러를 담당하는 클래스이다.

SM\_Model과 SM\_Controller\_LCS에서 구현된 것들을 ActionEvent로 받아 들여서

@ Test

main클래스이다.

@ View\_mainFrame

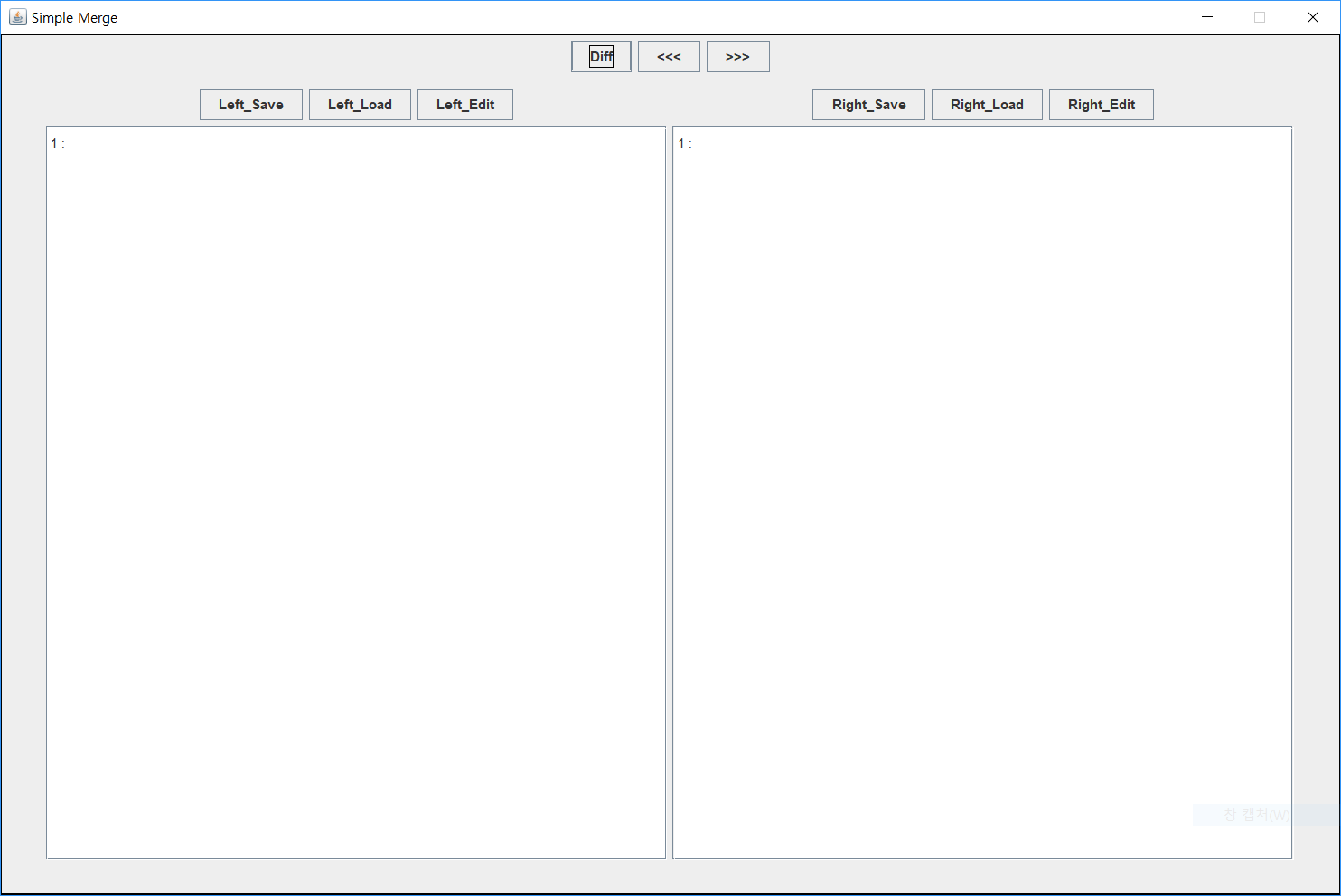
MVB패턴 중 뷰에 속하는 클래스로 Gui의 가장 큰 틀을 만드는 클래스이다. Gui는 자바의 JPanel을 사용하여 만들었다. 클래스안에서 구현하는 버튼은 merge\_L(<<<), merge\_R(>>>), diff가 있다.

@ View\_textFrame

MVC패턴 중 뷰에 속하는 클래스로 mainFrame 안에 왼쪽, 오른쪽 text창을 구현하는 클래스이다.

구현하는 버튼은 Left\_Save, Left\_Load, Left\_Edit, Right\_Save, Right\_Load, Right\_Edit이 있다.

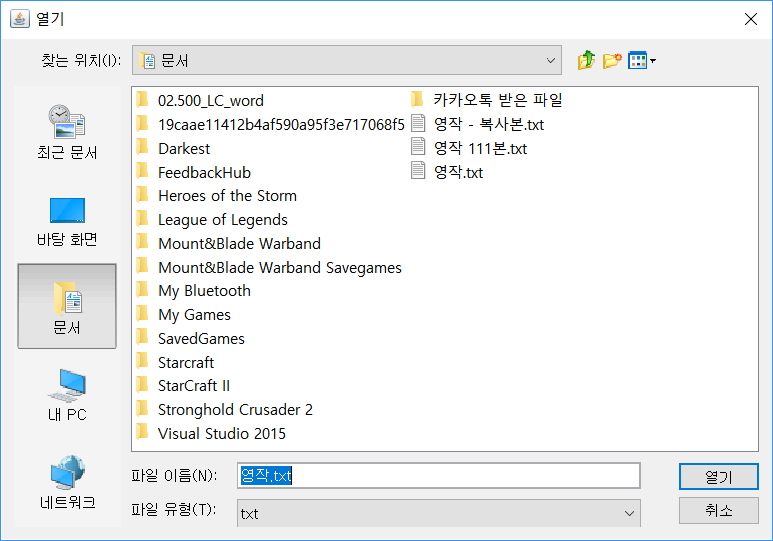
1. Program UI and Features



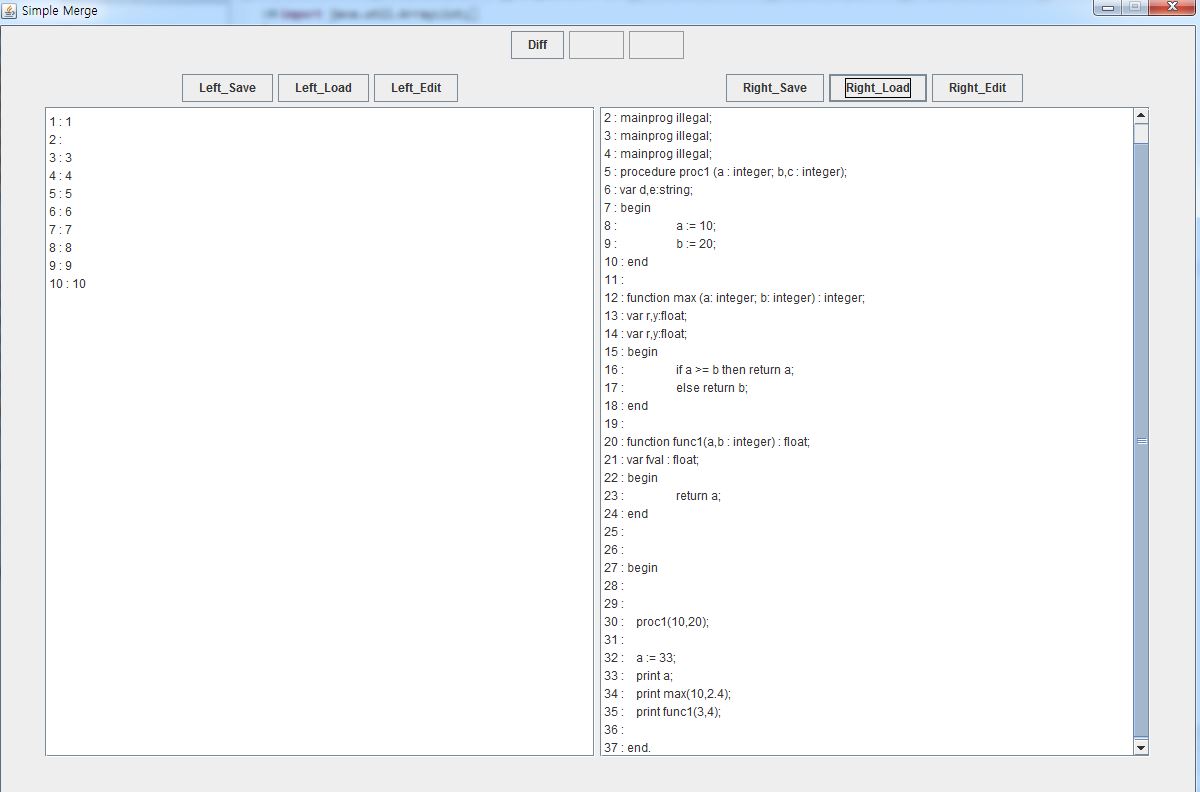
첫 실행 시 위와 같은 화면이 나타난다.

각 좌우 패널의 load버튼을 눌러 파일을 불러오거나 edit을 눌러 패널마다 수정을 할 수 있다.

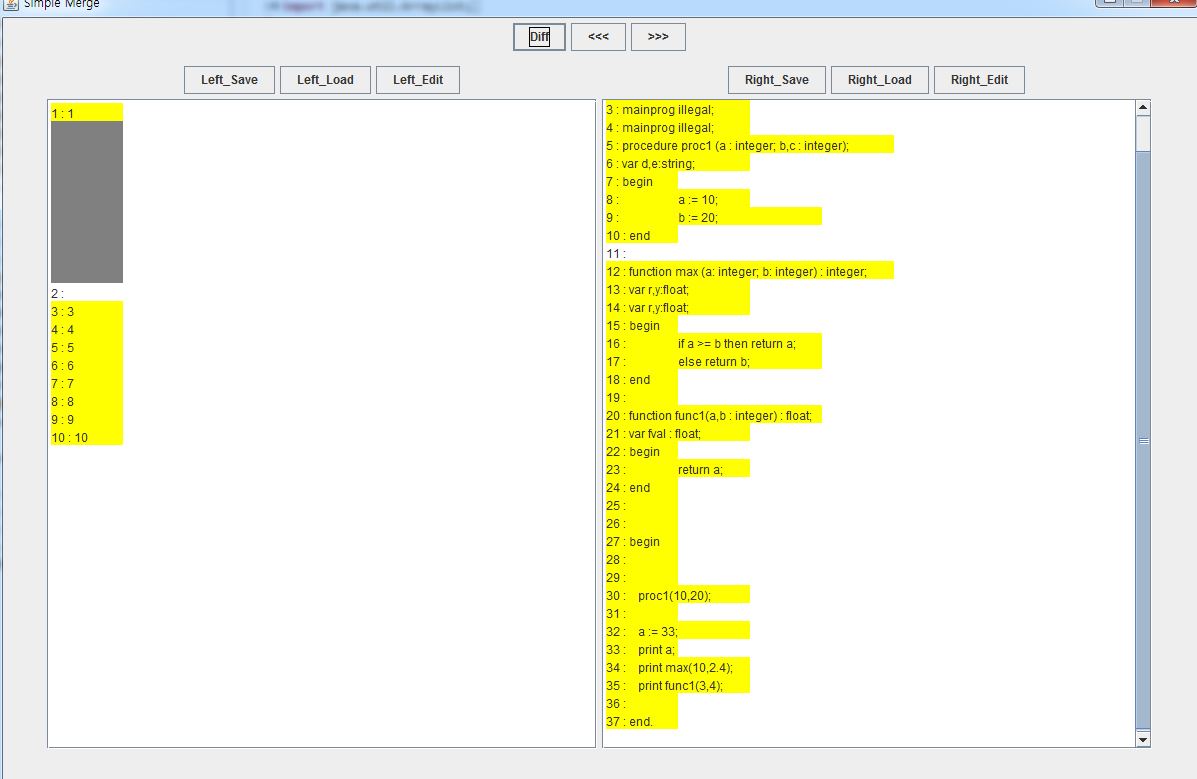
diff버튼을 실행하지 않으면 우측의 <<, >> merge버튼이 비활성화 된다.

Save 혹은 load 버튼을 클릭할 시 위와 같은 창이 나타난다.

이를 통해 파일을 불러오거나 저장할 수 있으며, 불러온 파일이 없는 채로 edit을 하여도 새 파일을 만들도록 하였다.

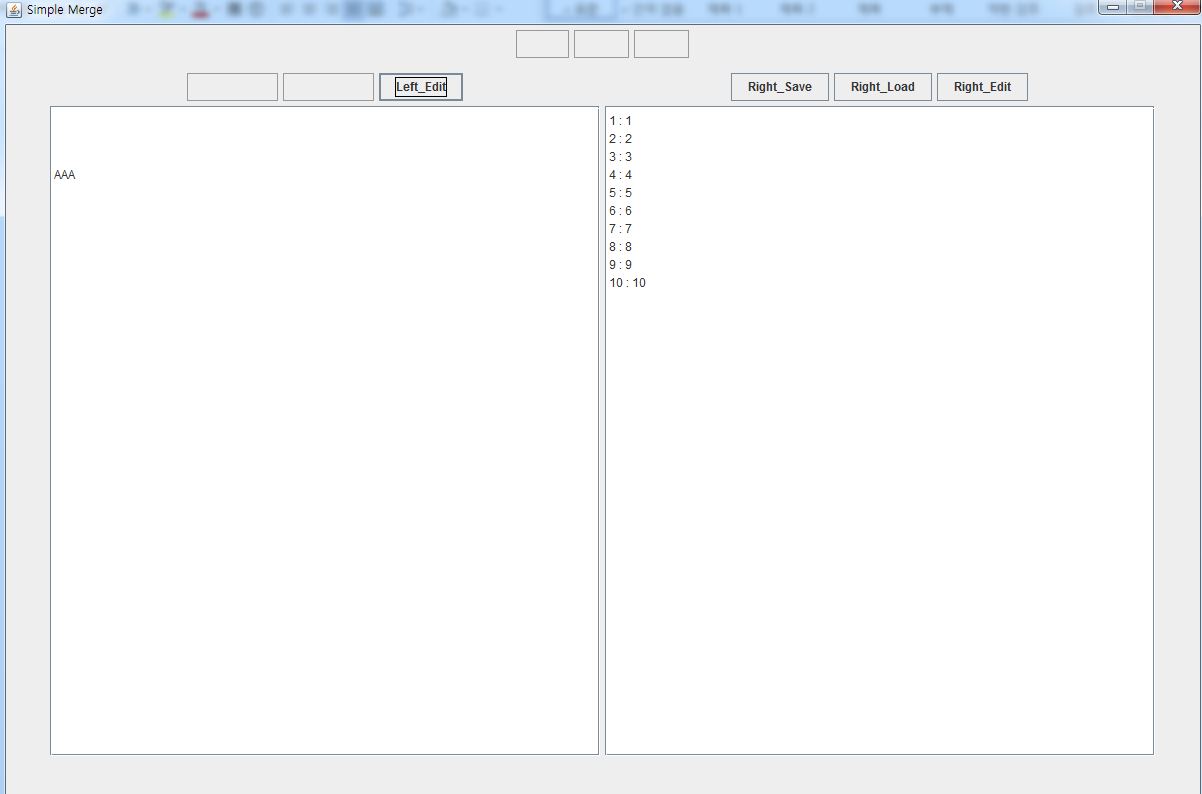
텍스트파일을 불러왔을 때의 화면

편집 및 비교의 용이를 위해 각 줄마다 줄 번호를 좌측에 표시하도록 하였다.



diff버튼을 실행했을 때의 화면

서로 내용이 다른 부분의 경우 노란색으로 표시되며, 반대쪽과 비교해서 없는 부분이 있을 경우 검은색으로 표시하도록 했다.



좌측 edit버튼을 눌렀을 때의 화면

Edit이 활성화되면 줄 번호가 사라지며 save, load, diff, merge버튼이 모두 비활성화 된다.

Edit버튼을 한번 더 누르면 edit 상태에서 빠져나온다.

1. Test
2. Functional Unit Test Cases
3. GUI Test Case
4. Execution Screen & Explantion
5. Project Management