Graphical Clone Checker

T3. 김태현. 고수창. 이상규.

Contents

01 Project Introduction

Motivation
Alternative Solutions
Project Objectives

Motivation

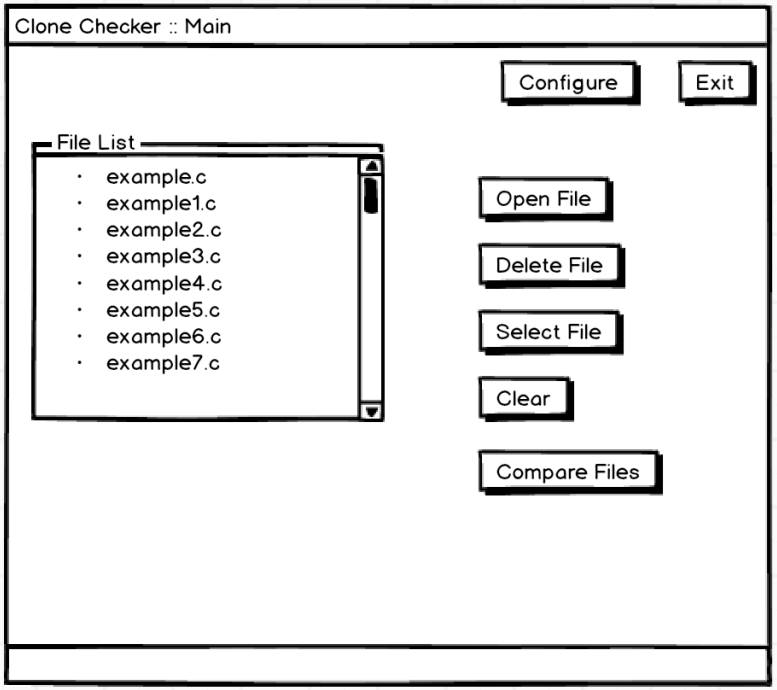
클론을 찾기 위해 사람이 코드를 일일이 다 비교해서
검사하는 일은 불가능

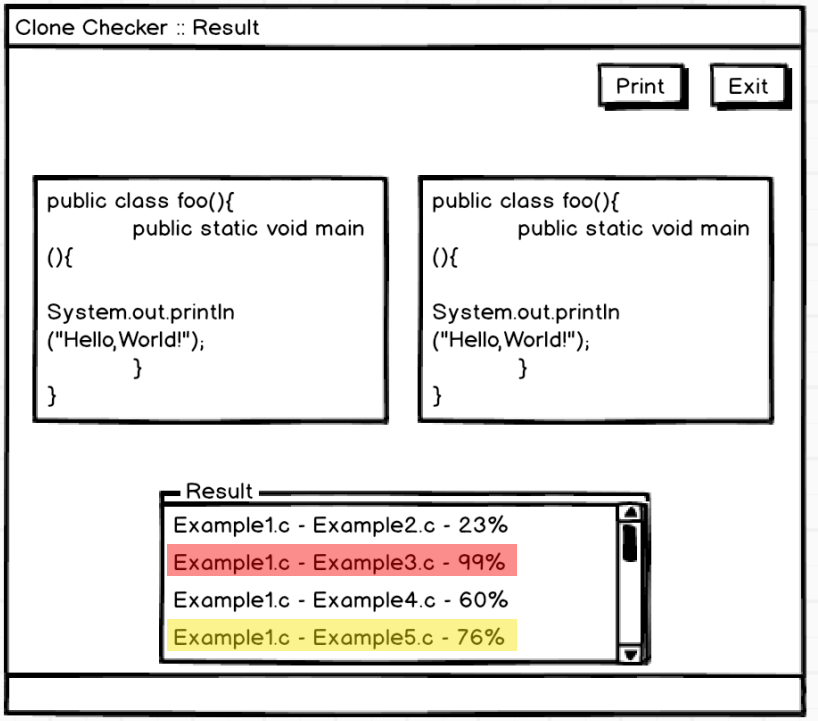
Alternative Solutions

- 기존의 Clone Checker
  - 한 번에 다수의 파일을 검사하지 못 함
  - Source code보다는 논문 등 문서에 쓰임
- 외주 업체에 개발 의뢰
- 수동으로 검사

Project Objectives

- 분산된 다수의 파일을 1:n 비교 가능
- 비교 결과를 정량적으로 판단하여 Cheating 여부 표시
- 도출되는 결과들을 그래프와 원색을 이용하여 사용자가 이해하기 쉽게 표시





02 Requirements

Functional Requirements
Non-functional Requirements

Functional Requirements

Function	Description
Open Files	비교하고자 하는 파일 열기
Select Files	열려있는 파일 선택하기
Delete Files	선택한 파일들 삭제하기
Display File List	열려있는 파일 목록 보여주기
Clear	파일 목록 전체 제거하기
Configure	비교 결과 몇 퍼센트 이상의 결과를 보여줄지, 오름차순, 내림차순으로 보여줄지 등 설정
Compare Files	선택한 파일들 비교하기
Check Raw Text	텍스트 단위의 단순 비교
Parse Files	소스 파일 파싱하기
Check Name	변수명, 함수면 변경 검사하기
Check Comment	주석의 내용 검사하기
Check Condition	if - switch 변경 검사하기
Check Loop	for - while 변경 검사하기
Check Function Merge	함수 병합 검사하기
Display Result	결과 보여주기
Calculate Scores	검사 결과 점수 계산하기
Exit	프로그램 종료하기

Non-functional Requirements

- 결과를 표시할 때 강조하기 위해 원색을 사용한다.
- 결과를 Percentage로 보여준다.
- 결과를 그래프로 표시한다.

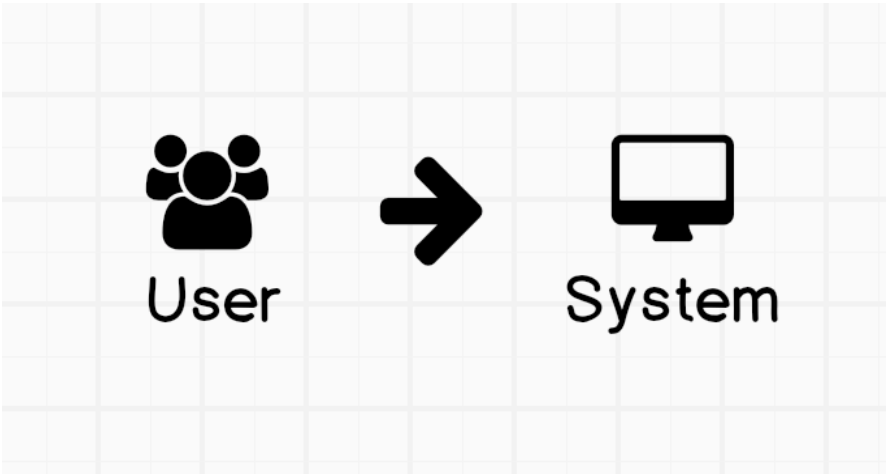
03 Architecture

System Architecture

Use Case

System Architecture

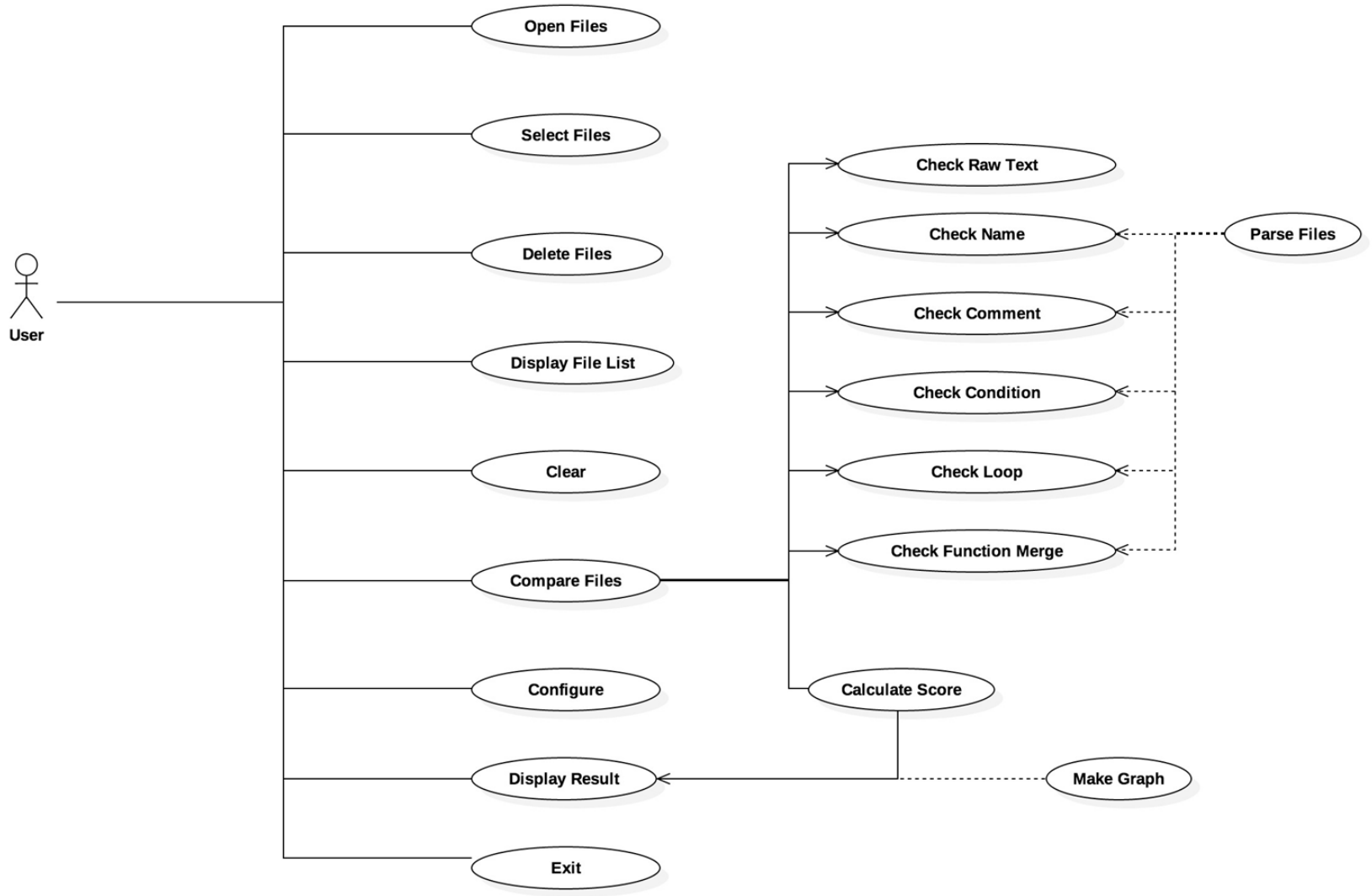
Stand-Alone (1 Tier)



- User (사용자)

과제의 표절 여부를 검사하려는 교수님 및 조교

Use Case


User
Open Files
Select Files
Delete Files
Display File List
Clear
Compare Files
Configure
Display Result
Exit
Check Raw Text
Check Name
Check Comment
Check Condition
Check Loop
Check Function Merge
Calculate Score
Parse Files
Make Graph

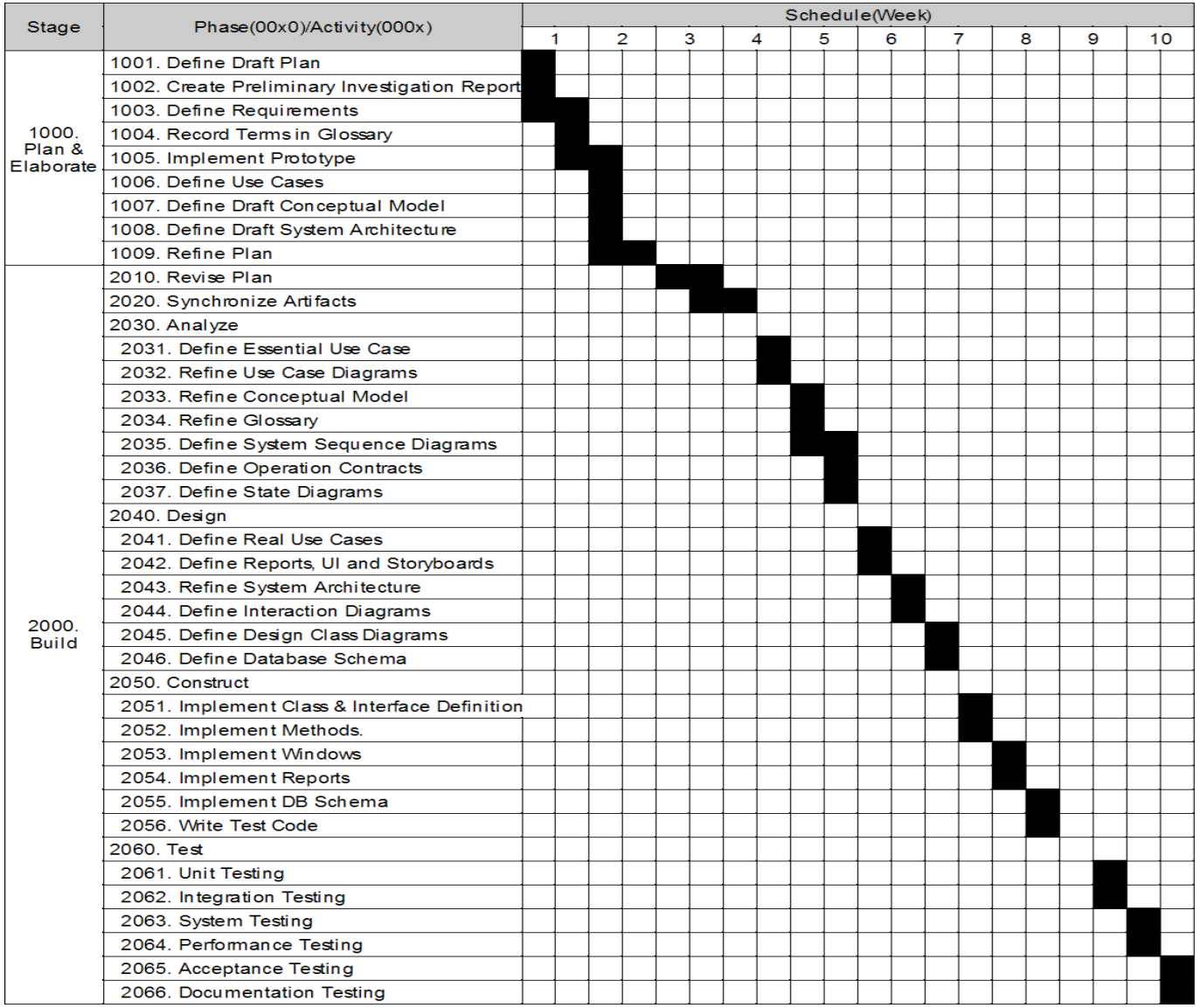
04 Detailed Plan

Risk Management & Reduction Schedule

Risk Management & Reduction

Risk	Reduction Plan
OOPT Skill	이전 프로젝트 자료 참조 및 교수님과 조교님께 자문을 구한다.
Programming Skill	Programming 관련 서적 및 Official API Document를 참조하여 공부한다.
Plan Management	일정이 밀리지 않도록 주기적으로 팀원들간에 피드백을 하는 시간을 갖고 최대한 일정에 맞추려고 노력한다.
직장	야근을 최대한 멀리한다.
연애	사랑 싸움을 최대한 멀리하여 감정적인 소모를 방지한다.
건강	평소에 주기적인 운동과 규칙적인 생활을 한다. 아플 때는 바로 병원에 가도록 한다.

Schedule


Stage
Phase(00x0)/Activity(000x)
Schedule(Week)
1
2
3
4
5
6
7
8
9
10
1000. Plan & Elaborate
1001. Define Draft Plan
1002. Create Preliminary Investigation Report
1003. Define Requirements
1004. Record Terms in Glossary
1005. Implement Prototype
1006. Define Use Cases
1007. Define Draft Conceptual Model
1008. Define Draft System Architecture
1009. Refine Plan
2000. Build
2010. Revise Plan
2020. Synchronize Artifacts
2030. Analyze
2031. Define Essential Use Case
2032. Refine Use Case Diagrams
2033. Refine Conceptual Model
2034. Refine Glossary
2035. Define System Sequence Diagrams
2036. Define Operation Contracts
2037. Define State Diagrams
2040. Design
2041. Define Real Use Cases
2042. Define Reports, UI and Storyboards
2043. Refine System Architecture
2044. Define Interaction Diagrams
2045. Define Design Class Diagrams
2046. Define Database Schema
2050. Construct
2051. Implement Class & Interface Definition
2052. Implement Methods.
2053. Implement Windows
2054. Implement Reports
2055. Implement DB Schema
2056. Write Test Code
2060. Test
2061. Unit Testing
2062. Integration Testing
2063. System Testing
2064. Performance Testing
2065. Acceptance Testing
2066. Documentation Testing

Q & A