

# 마감 하루 전

## Category Partitioning Testing Tool

**Project Team**

**T1**

Date

**2017-03-30**

---

**Team Information**

201111334 김강욱

201211339 김진욱

201312243 김동권

201510411 이소영

## INDEX

**Activity 1001. Define Draft Plan**

**Activity 1002. Create Preliminary Investigation Report**

**Activity 1003. Define Requirements**

**Activity 1004. Record Terms in Glossary**

**Activity 1005. Implement Prototype**

**Activity 1006. Define Business Use Case**

**Activity 1007. Define Business Concept Model**

**Activity 1008. Define Draft System Architecture**

**Activity 1009. Define System Test Case**

**Activity 1010. Refine Plan**

## Activity 1001. Define Draft Plan

### 1. Motivation

점차 복잡하고 다양한 program 들이 개발됨에 따라 program의 spec 범위도 넓어지고 있다. '한정된 시간에서 꼭 필요한 Test 를 수행함으로써 가능한 적은 시간 내에 오류 없는 프로그램을 만드는 것' 에 대한 수요가 증가하고 있다.

이러한 추세에서 자연어로 구성 된 specification 문서는 tester와 기획자 간의 간격을 더 넓히며 시간의 효율성을 떨어트린다. Category Partitioning 은 Spec 명세를 독립적으로 테스트 할 수 있는 기능 단위로 분해하고 조건 별로 category를 설정함으로써 Tester 에게 효율적인 test case 조합을 제공한다. 본 Testing tool이 제공하는 효율성은 기업뿐 만 아니라 학교에서 진행되는 프로젝트에도 동일하게 적용되어야 하며 이에 따라 Category Partitioning Testing tool 을 개발하게 되었다.

### 2. Project Objectives

사용자가 정한 Category 별 Constraint에 대해 ideal한 test case를 조합하고, 이 결과를 사용자가 정한 우선순위에 따라 중요한 test case에 대한 정보도 제공하는 것을 목적으로 한다.

중요하다고 생각되는 Value 들에 대해 가중치를 줄 수 있고 test case 산출 시 이 가중치에 대한 계산 값을 함께 제공한다.

- 사용자가 입력 한 category 에 대해 test case를 조합한다.
- 팀원들이 각자 생각하는 중요한 test case 조합에 대해 명시한다.
- Admin 권한을 가진 팀장이 프로그램 내에서 팀원들에게 전달 받은 test case 조합에 대해 Value Set과 가중치에 대해 한번 더 검토한다.
- Test Case 와 함께 먼저 test가 수행되어야 하는 case 에 대한 정보를 제공받는다.

### 3. Project Scope

'마감 하루 전' 은 중요한 test case 를 먼저 test 할 수 있는 계산 된 결과 값을 제공함으로써 사용자로 하여금 효율적인 testing 이 가능하도록 한다. 프로그램 별로 중요하거나 오류 가능성이 많은 case가 존재하고, 한정된 시간 내에 오류 수정 및 program의 효율적 개발을 위해 이런 test가 먼저 수행되어야 한다는 점에 초점을 맞추었다.

### 4. Functional requirements

- Open File
- Analyze Set
- Apply to the Weight of the value
- Modify Weight
- Input Essential Test Case
- Show Missing List
- Check Constraints
- Compare with Test Case
- Indicate Essential Test Case
- Modify Constraint
- Make Test Case
- Rearrange by Weight Average
- Calculate Weight Average
- Make Final Test Case File

### 6. Non-Functional requirements

- 1분이내로 결과값이 나와야 한다.

**7. Resource Estimation**

- 1) Human Efforts (Man – Month: 4 – 3
- 2) Human Resource: 컴퓨터공학과 2명, 다전공 2명
- 3) Project Duration: 16주
- 4) Budget

	부분 계	비고
인건비	360	
활동비	12	
계	372	

**Activity 1002. Create Preliminary Investigation Report.**

**1. Alternative Solutions**

- 1) 기존에 있는 Category Partitioning Testing Tool을 사용한다.
- 2) 기존에 있는 Category Partitioning Testing Tool을 산다.
- 3) 타 소프트웨어 개발업체에 외주를 의뢰한다.

**2. Project Justification (Business Needs)**

- 1) Cost
  - ① 외주업체에 비해 개발 비용이 저렴하다.
- 2) Duration
  - ① 2017년 1학기 안에 완성해야 한다.
- 3) Risk
  - Plan Management, OOPT Skill, Programming Skill, UML Skill, 시험
- 4) Effect
  - ① 유지보수의 이점이 있다.
  - ② 빠른 시간 안에 테스트케이스를 조합할 수 있다.
  - ③ 프로그램을 독점적으로 이용할 수 있다.

**3. Risk Management**

Risk	Probability	Significance	Weight
Plan Management	5	5	25
OOPT Skill	4	10	40
Programing Skill	3	8	24
UML Skill	5	9	45
시험	10	10	100

**4. Risk Reduction Plan**

Risk	Reduction Plan
Plan Management	관련 사이트와 서적을 활용하여 도움을 얻는다.
OOPT Skill	수업강의 자료와 교수님, 조교님의 도움을 얻는다.
Programming Skill	Programing 관련 서적을 읽고 선배들의 도움을 얻는다.
UML Skill	UML관련 서적을 읽고 선배들의 도움을 얻는다.
시험	조원들끼리 서로 협력하여 서로 도움을 주고 받는다.

**5. Market Analysis**

1) 기존의 Category portioning Test Tool이 많지 않다. 또한 자연어로 명시된 programming specification 에 대한 요구가 점점 증가할 것이다. 한정된 시간 내에서 우선적으로 검증해야 하는 테스트 케이스가 필요할 것이다. 한정된 시간 내에 최대한 많은 오류를 잡고 출시해야 한다. 효율적으로 사용해야 한다.

**6. Other Managerial Issues**

1) 2017년 6월까지 개발을 완성하여 A+을 받아야 한다.

**Activity 1003. Define Requirements**

**1. Functional Requirements**

Function	Description
Open File	엑셀 파일 열기
Analyze Constraints	Value의 제한 조건(single, error 등 분석)
Apply to the Weight of the value	Value에 사용자가 입력한 가중치를 적용
Modify Weight	가중치 수정
Input Essential Test Case	중요 테스트 케이스 입력
Show Missing List	중요 테스트 케이스 중 프로그램에 의해 조합된 테스트 케이스에서 누락된 케이스를 보여줌
Check Constraints	중요 테스트 케이스가 조합된 테스트 케이스에 없다면 제약조건을 확인하고 그 원인 확인
Compare with Test Case	프로그램에 의해 조합 된 테스트 케이스와 사용자가 입력한 중요 테스트 케이스를 비교
Indicate Essential Test Case	조합 된 테스트 케이스 중 사용자가 입력 한 중요 테스트 케이스 표시
Modify Constraints	제한 조건 수정
Make Test Case	테스트 케이스 생성
Rearrange by Weight Average	가중치 평균에 따른 재정렬
Calculate Weight Average	테스트 케이스를 구성하는 Value들의가중치 평균 계산
Make Final Test Case File	최종 테스트 케이스 파일 생성

Ref.#	Function	Category
R 1.1	Open File	Evident
R 1.2	Analyze Constraints	Hidden
R 2.1	Apply to the Weight of the value	Hidden
R 3.1	Modify Weight	Hidden

R 4.1	Input Essential Test Case	Evident
R 4.2	Show Missing List	Evident
R 4.3	Check Constraints	Hidden
R 4.4	Compare with Test Case	Hidden
R 4.5	Indicate Essential Test Case	Hidden
R 5.1	Modify Constraints	Hidden
R 6.1	Make Test Case	Evident
R 6.2	Rearrange by Weight Average	Evident
R 6.3	Calculate Weight Average	Hidden
R 7.1	Make Final Test Case File	Evident

## 2. Operating Environments

- 1) OS: Windows 7, Windows 10
- 2) 개발언어: Java

## 3. Development Environments

- OS: Windows 7, Windows 10  
 IDE: Eclipse  
 개발 언어: Java

## Activity 1004. Record Terms in Glossary

Term	Description	Remarks
Open	열다	
File	파일	
Analyze	분석하다	
Input	입력한다	
Weight	가중치 값	
Assign	할당한다	
Value	가중치설정 후 계산된 값	
Modify	수정한다	
Essential Test Case	중요 테스트 케이스	
Show	보여준다	
Check	체크한다	
Constraints	Single, Error, Property 등의 제한 조건	
Compare	비교한다	
Test Case	Value 조합으로 만들어진 테스트케이스	
Indicate	표시한다	
Make	생성한다	
Test Case	기본 테스트 케이스	
Rearrange	재배열	
Average	평균	

### Activity 1005. Implement Prototype

Category1.xlsx

Index	Weight	Value1	Value1 Weight	Value1 Constraints	Value2
102	1	소스파일 매우 많이 추가	1	Single	
104	1	소스 파일이 아닌 파일 추가	1	Error	
110	1	아무것도 선택하지 않고 파일 삭제	1	Single	
100.200.300.310.320.330.340.350.400.410.420	0.7	소스 파일 1개 추가	0.5	property AD1	클론제킹 버튼 클릭
101.111.200.300.310.320.330.340.350.400.410.420	0.6	서로 다른 소스파일 2개 이상 추가	0.7		파일을 선택한 후 1개 삭제
101.112.200.300.310.320.330.340.350.400.410.420	0.6	서로 다른 소스파일 2개 이상 추가	0.7		
101.112.200.300.310.320.330.340.350.400.410.420	0.6	서로 다른 소스파일 2개 이상 추가	0.7		
101.112.200.300.310.320.330.340.350.400.410.420	0.6	서로 다른 소스파일 2개 이상 추가	0.7		
101.112.200.300.310.320.330.340.350.400.410.420	0.6	서로 다른 소스파일 2개 이상 추가	0.7		

Generate

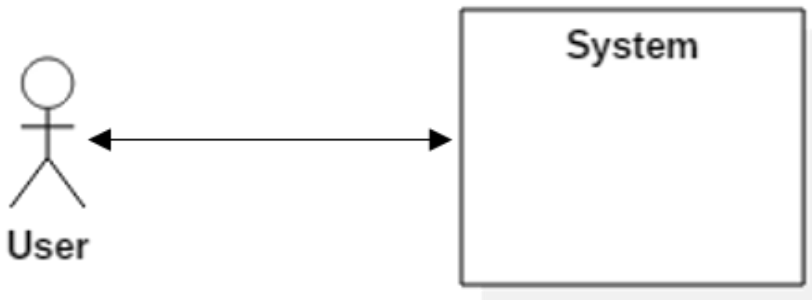
Essential Test Case

Index	Weight	Value1	Value1 Weight	Value1 Constraints	Value2
100.200.300.310.320.330.340.350.400.410.420	0.7	소스 파일 1개 추가	0.5	property AD1	
102.111.200.300.310.320.330.340.350.400.410.420	0.6	소스파일 매우 많이 추가	0.7	Single	파일을 선택한 후 1개 삭제
100.112.200.300.310.320.330.340.350.400.410.420	0.6	소스 파일 1개 추가	0.7	property AD1	파일을 삭제하지 않는다.

Input Essential Test Case

### Activity 1006. Define Business Use Case

#### 1. Define Business Use Case

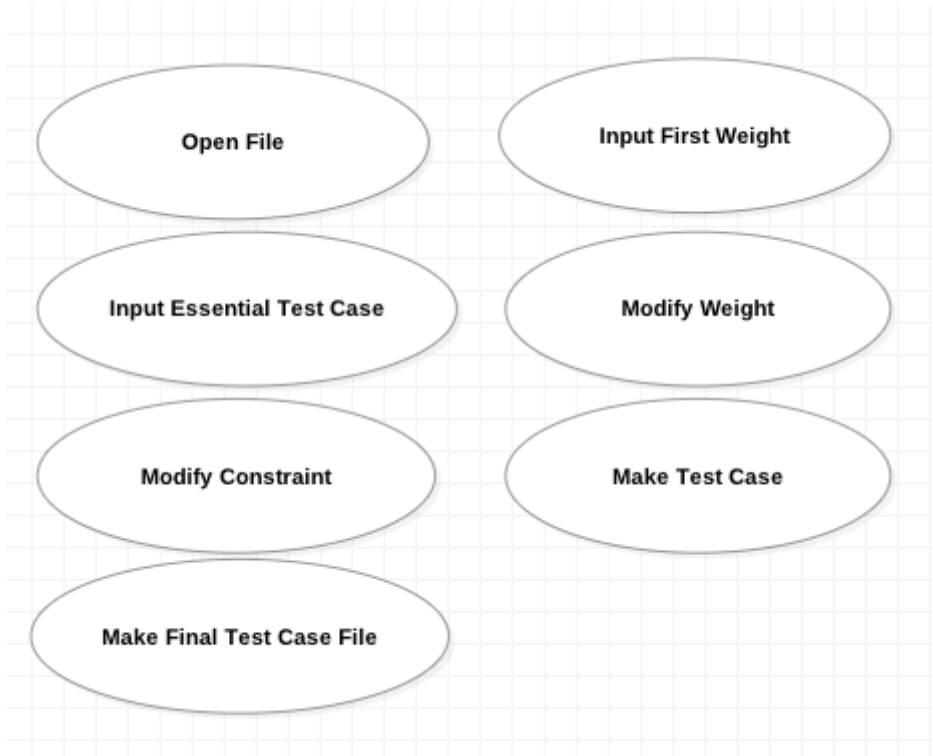


#### 2. Identify and Describe Actors

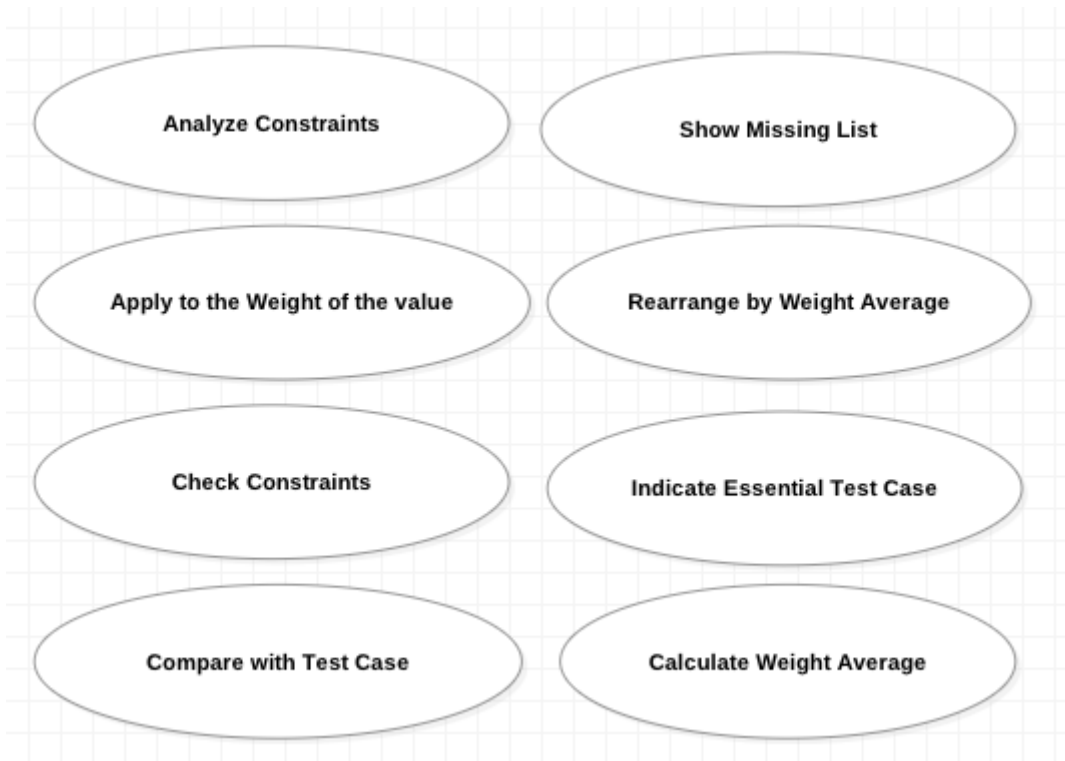
1) Actor

### 3. Identify Use-Case

#### 1) Actor based



#### 2) Event based





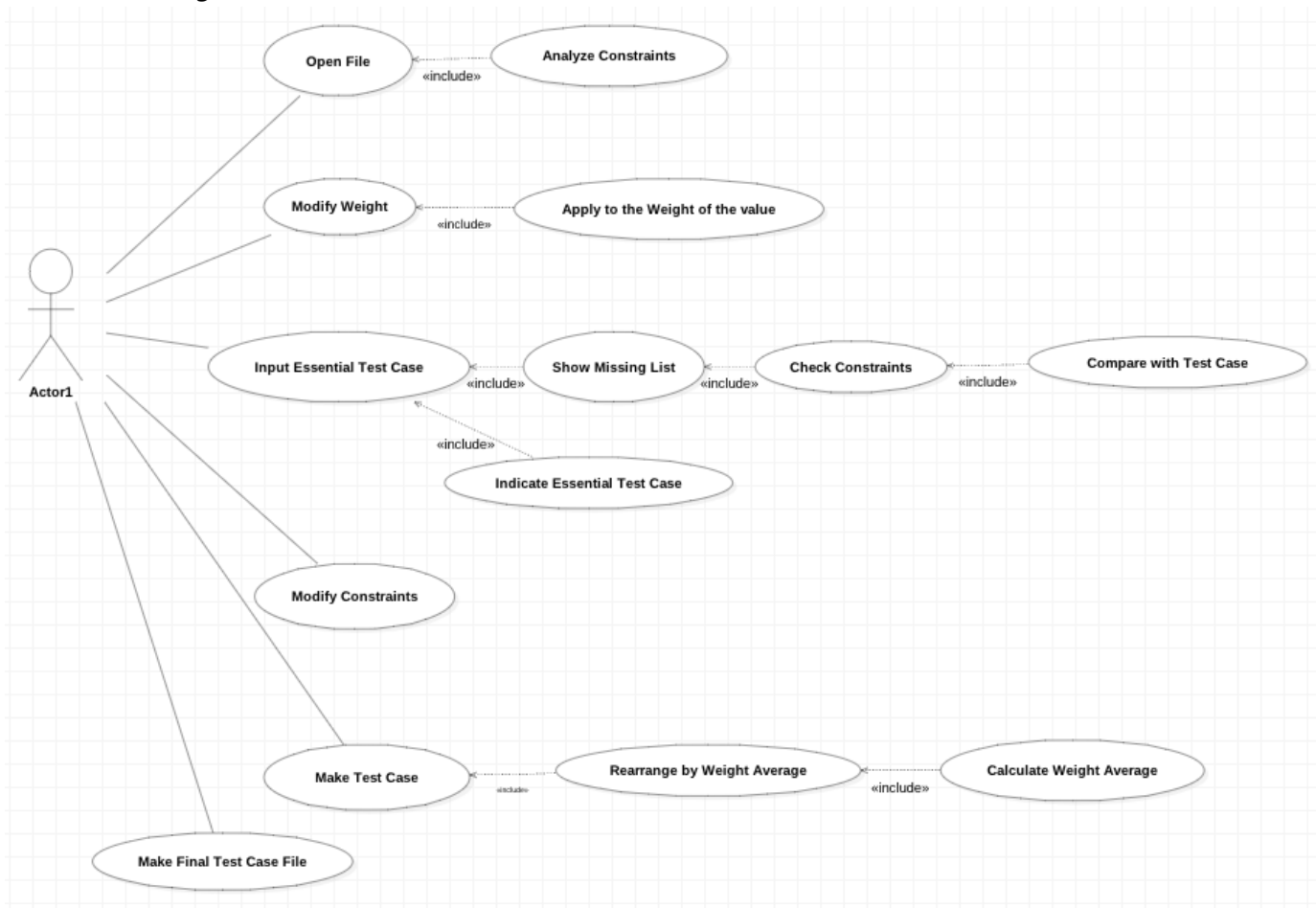
**4. Allocate system functions into Related Use-Case**

Ref.#	Function	Use-Case
R 1.1	Open File	Open File
R 1.2	Analyze Constraints	Analyze Constraints
R 2.2	Apply to the Weight of the value	Apply to the Weight of the value
R 3.1	Modify Weight	Modify Weight
R 4.1	Input Essential Test Case	Input Essential Test Case
R 4.2	Show Missing List	Show Missing List
R 4.3	Check Constraints	Check Constraints
R 4.4	Compare with Test Case	Compare with Test Case
R 4.5	Indicate Essential Test Case	Indicate Essential Test Case
R 5.1	Modify Constraints	Modify Constraints
R 6.1	Make Test Case	Make Test Case
R 6.3	Rearrange by Weight Average	Rearrange by Weight Average
R 6.4	Calculate Weight Average	Calculate Weight Average
R 7.1	Make Final Test Case File	Make Final Test Case File

**5. Category Use-Case**

Function	Use-Case
Open File	Primary
Analyze Constraints	Primary
Apply to the Weight of the value	Primary
Modify Weight	Primary
Input Essential Test Case	Primary
Show Missing List	Primary
Check Constraints	Primary
Compare with Test Case	Primary
Indicate Essential Test Case	Primary
Modify Constraints	Primary
Make Test Case	Primary
Rearrange by Weight Average	Primary
Calculate Weight Average	Primary
Make Final Test Case File	Primary

6. Draw a Use-Case diagram



7. Describe Use-Case

Use Case	1.Open File
Actors	User
Description	엑셀 파일을 불러온다.

Use Case	2. Make Test Case
Actors	System
Description	테스트 케이스를 생성한다.

Use Case	3. Rearrange by Weight Average
Actors	System
Description	테스트 케이스를 구성하는 Value들의 가중치 평균값을 구하여 재정렬한다.

Use Case	4. Input Essential Test Case
Actors	User
Description	중요 테스트 케이스를 입력한다.

Use Case	5. Indicate Essential Test Case
----------	---------------------------------

Actors	System
Description	Category Partitioning 에 의해 조합된 테스트 케이스 리스트에서 사용자가 입력한 중요한 테스트 케이스를 표시한다.

Use Case	6. Check Constraints
Actors	System
Description	조합된 테스트 케이스와 중요 테스트 케이스를 비교해 봤을 때 중요 테스트 케이스 누락의 원인을 분석한다. 제약 조건 설정이 잘못되어 있을 경우 User 에게 잘못 된 부분을 알려준다.

Use Case	7. Modify Weight
Actors	User
Description	Value에 할당 된 가중치를 수정한다.

Use Case	8. Modify Constraints
Actors	User
Description	제한 조건을 수정한다.

Use Case	9. Make Final Test Case File
Actors	User
Description	테스트 케이스 파일을 문서화 하여 생성한다.

Use Case	10. Analyze Constraints
Actors	System
Description	테스트 케이스 조합 규칙에 따라 Value들의 제한조건들을 분석한다

Use Case	11. Apply to the Weight of the value
Actors	System
Description	사용자가 입력 한 가중치를 Value에 할당하고 수정한 Value가 테스트케이스를 구성하는 Value일 경우 테스트케이스 가중치 평균 식에 들어가는 Value 가중치를 수정한다.

Use Case	12. Show Missing List
Actors	System
Description	사용자가 입력한 중요한 테스트 케이스 중Category Partitioning 에 의해 조합된 테스트 케이스 리스트에 없는 케이스를 표시한다.

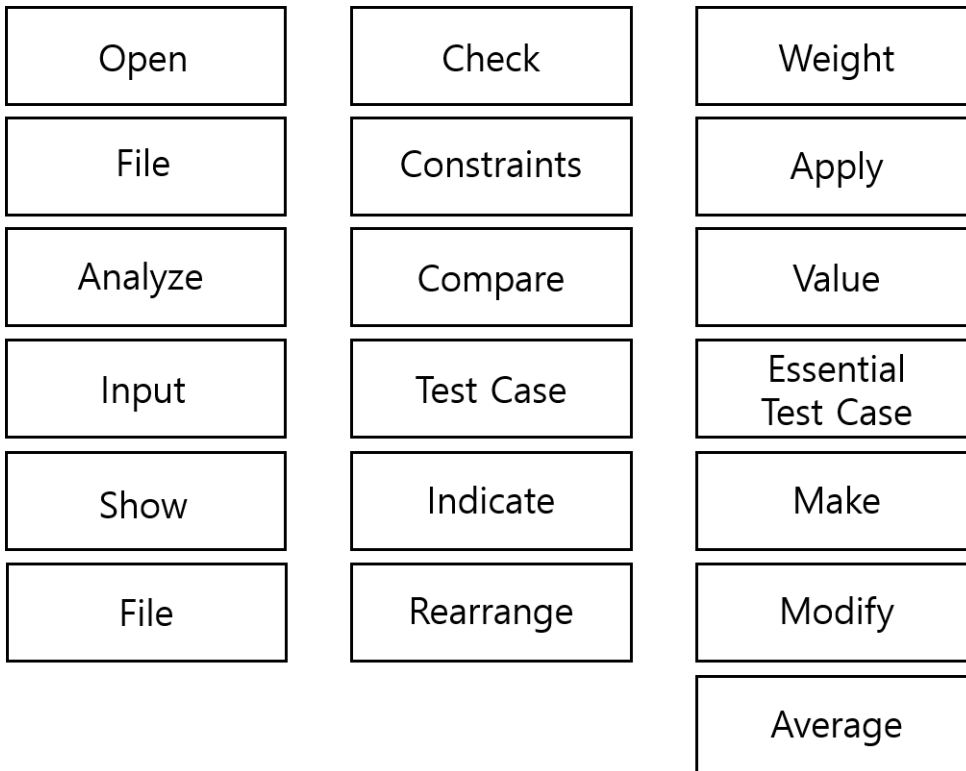
Use Case	13. Compare with Test Case
Actors	System
Description	테스트 케이스와 비교한다.

Use Case	14. Calculate Weight Average
Actors	System
Description	테스트 케이스를 구성하는 Value 들의 가중치 평균을 계산한다.

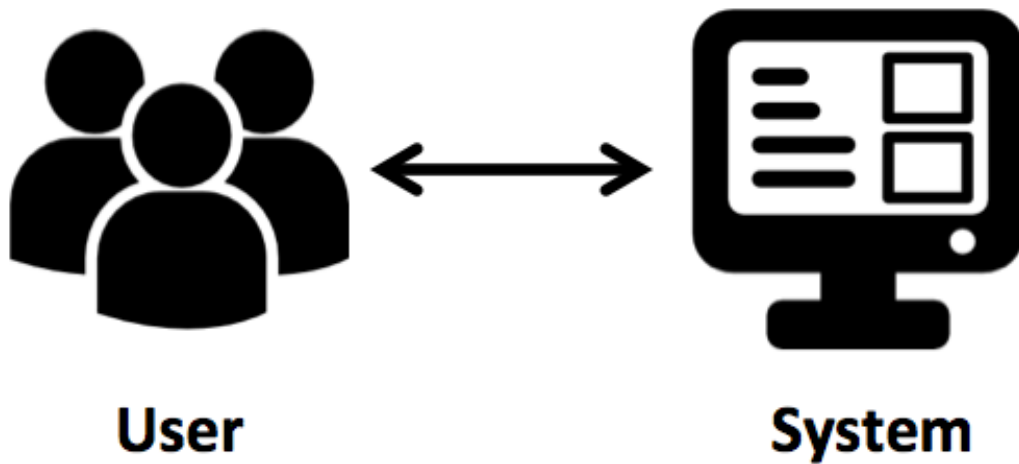
**8. Rank Use-Case**

Use-Case Number & Name	Rank
1. Open File	High
2. Analyze Constraints	High
3. Apply to the Weight of the value	High
4. Modify Weight	High
5. Input Essential Test Case	High
6. Show Missing List	High
7. Check Constraints	High
8. Compare with Test Case	High
9. Indicate Essential Test Case	High
10. Modify Constraints	High
11. Make Test Case	High
12. Rearrange by Weight Average	High
13. Calculate Weight Average	High
14. Make Final Test Case File	High

**Activity 1007. Define Business Concept Model**



**Activity 1008. Define Draft System Architecture**



**Activity 1009. Define System Test Case**

Test Number	Test 항목	Description	Use Case	System Function
1	파일 열기 시험	파일 열기 버튼을 눌러 엑셀 파일을 여는 기능 test	Open File	R 1.1
2	필수 테스트 케이스 입력 시험	입력 버튼을 눌러 필수 테스트 케이스를 입력하는 기능 test	Input Essential Test Case	R 2.1
3	테스트 케이스 생성 시험	조합 버튼을 눌러 테스트 케이스 생성 기능 test	Create Test Case	R 3.1
4	필수 테스트 케이스 표시 시험	필수 테스트 케이스를 따로 표시하는 기능 test	Indicate Essential Test Case	R 4.1
5	필수 테스트 케이스 확인 시험	테스트 케이스 생성 결과에 필수 테스트 케이스가 존재하는지 확인하는 test	Identify Essential Test Case	R 4.2
6	제한 조건 확인 시험	필수 테스트 케이스가 없을 경우 원인을 확인하는 기능 test	Check Constraints	R 4.3
7	제한 조건 수정 시험	제한 조건을 수정하는 기능 test	Modify Constraint	R 5.1
8	가중치 수정 시험	가중치를 수정하는 기능 test	Modify Weight	R 5.2
9	재정렬 시험	가중치를 수정한 후 재정렬하는 기능 test	Rearrange by Weight	R 6.1
10	최종 테스트 케이스 파일 생성 시험	최종적으로 생성된 테스트 케이스를 파일 출력 버튼을 눌러 파일로 내보내는 기능 test	Make final Test Case File	R 7.1

**Activity 1010. Refine Plan**

**1. Project Scope**

‘마감 하루 전’은 중요한 test case 를 먼저 test 할 수 있는 계산 된 결과 값을 제공함으로써 사용자로 하여금 효율적인 testing 이 가능하도록 한다. 프로그램 별로 중요하거나 오류 가능성이 많은 case가 존재하고, 한정된 시간 내에 오류 수정 및 program의 효율적 개발을 위해 이런 test가 먼저 수행되어야 한다는 점에 초점을 맞추었다.

**2. Project Objectives**

사용자가 정한 Category 별 Constraint에 대해 ideal한 test case를 조합하고, 이 결과를 사용자가 정한 우선순위에 따라 중요한 test case에 대한 정보도 제공하는 것을 목적으로 한다. 중요하다고 생각되는 Value 들에 대해 가중치를 줄 수 있고 test case 산출 시 이 가중치에 대한 계산 값을 함께 제공한다.

- 사용자가 입력 한 category 에 대해 test case를 조합한다.
- 팀원들이 각자 생각하는 중요한 test case 조합에 대해 명시한다.
- Admin 권한을 가진 팀장이 프로그램 내에서 팀원들에게 전달 받은 test case 조합에 대해 Value Set과 가중치에 대해 한 번 더 검토한다.
- Test Case 와 함께 먼저 test가 수행되어야 하는 case 에 대한 정보를 제공받는다.

### 3. Functional Requirements

- Open File
- Create Test Case
- Input Weight
- Rearrange by Weight
- Input Essential Test Case
- Identify Essential Test Case
- Indicate Essential Test Case
- Check Constraints
- Modify Weight
- Modify Constraint
- Make Final Test Case File

### 4. Performance Requirements

가중치에 따른 재정렬한 결과가 1분이내로 나와야 한다

### 5. Operating Environment

- 1) OS: Windows 7, Windows 10
- 2) IDE: Eclipse
- 3) 개발 언어: Java

### 6. Resources

- 1) Human Efforts (Man – Month): 4 – 3
- 2) Human Resource: 컴퓨터공학과 2명, 다전공 2명
- 3) Project Duration: 16주
- 4) Budget

	부분 계	비고
인건비	360	
활동비	12	
계	372	

7. Scheduling

Stage	Phase(00X0)/Activity(000X)	Schedule(Week)									
		1	2	3	4	5	6	7	8	9	10
1000. Plan & Elaboration	1001. Define Draft Plan	█									
	1002. Create Preliminary Investigation Report	█									
	1003. Define Requirements	█									
	1004. Record Terms in Glossary	█									
	1005. Implement Prototype	█	█								
	1006. Define Business Use Case		█								
	1007. Define Business Concept Model		█								
	1008. Define Draft System Architecture		█								
	1009. Define System Test Case		█								
	1010. Refine Plan		█	█							
2000. Build	2010. Revise Plan			█							
	2020. Synchronize Artifacts			█							
	2030. Analyze			█							
	2031. Define Essential Use Cases			█	█						
	2032. Refine Use Case Diagrams			█	█						
	2033. Define Domain Model			█	█						
	2034. Refine Glossary			█	█						
	2035. Define System Sequence Diagrams			█	█	█					
	2036. Define Operation Contracts			█	█	█					
	2037. Define State Diagrams			█	█	█					
	2038. Refine System Test Case			█	█	█					
	2039. Analyze (2030) Traceability Analysis			█	█	█					
	2040. Design			█	█	█	█				
	2041. Design Real Use Cases			█	█	█	█				
	2042. Define Reports, UI, and Storyboards			█	█	█	█				
	2043. Refine System Architecture			█	█	█	█				
	2044. Define Interaction Diagrams			█	█	█	█				
	2045. Define Design Class Diagrams			█	█	█	█				
	2046. Design Traceability Analysis			█	█	█	█				
	2047. Define Database Schema			█	█	█	█				
	2050. Construct			█	█	█	█	█			
	2051. Implement Class & Methods Definitions			█	█	█	█	█	█		
	2052. Implements Windows			█	█	█	█	█	█		
	2053. Implement Reports			█	█	█	█	█	█		
	2054. Implement DB Schema			█	█	█	█	█	█		
	2055. Write Unit Test Code			█	█	█	█	█	█		
	2060. Test			█	█	█	█	█	█		
2061. Unit Testing			█	█	█	█	█	█			
2062. Integration Testing			█	█	█	█	█	█			
2063. System Testing			█	█	█	█	█	█			
2064. Performance Testing			█	█	█	█	█	█			
2065. Acceptance Testing			█	█	█	█	█	█			
2066. Documentation Testing			█	█	█	█	█	█			
2067. Testing Traceability Analysis			█	█	█	█	█	█			