

# OOPT Stage 1000

## <Plan and Elaboration>

### Version 1.0

#### Project Team

T7 Team

Date

2017-03-30

---

#### Team Information

201414134 오세욱

201414136 임현유

201211375 임동현

201211387 하헌규

## Table of Contents

1	Activity 1001. Define Draft Plan .....	3
2	Activity 1002. Create Preliminary Investigation Report .....	4
3	Activity 1003. Define Requirements.....	5
4	Activity 1004. Record Terms in Glossary .....	8
5	Activity 1005. Implement Prototype.....	9
6	Activity 1006. Define System Architecture .....	9
7	Activity 1007. Define Business Use Case.....	10
8	Activity 1008 Define Business Concept Model.....	17
9	Activity 1009 Define System Test Case.....	18
10	Activity 1010 Refine Plan .....	19

## 1 Activity 1001. Define Draft Plan

### 1.1 Motivation

Category Partition Tool은 Test Case를 계산해주는 도구이다. Test Case를 생성할 때 Tester의 능력과 경험에 따라 Category, Value, Constraint를 정하고, 그에 따라 Test Case의 Volume이 정해진다. 하지만 이러한 방식으로 Test Case의 Volume을 줄인다고 해도 Test Case의 Volume이 여전히 클 수 있다. 이 때 Tester가 자신이 먼저 확인하고 싶은 Value에 우선 순위를 부여하여 전체 Test Case를 계산하는 동안 자신이 선택한 Value의 Test Case를 먼저 확인하여 시간을 절약 할 수 있다.

### 1.2 Project Objectives

Value에 우선 순위를 부여하여 그 결과값을 먼저 출력하고, 확인할 동안 나머지 전체 결과값을 출력하여 시간을 절약할 수 있는 Category Partition Tool를 목표로 한다.

### 1.3 Functional Requirements

- A. Category 추가
- B. Category 수정
- C. Category 삭제
- D. Value 추가
- E. Value 수정
- F. Value 삭제
- G. Constraint 추가
- H. Constraint 수정
- I. Constraint 삭제
- J. Value 우선 순위 선택
- K. Test Case 계산
- L. 우선 순위 계산
- M. 진행 상황 계산

- N. 출력
- O. 우선 순위 출력
- P. 진행 상황 출력
- Q. File 출력
- R. 초기화
- S. 프로그램 종료

#### **1.4 Non-Functional Requirements**

- A. 조작하기 쉽고 익숙한 인터페이스
- B. 우선 순위 계산이 전체 계산보다 빠르게 출력

#### **1.5 Resource Estimation**

- A. Human Resource : 4 명
- B. Project Duration : 3 개월(12 주)
- C. Human Effort(Man-Month) : 12
- D. Cost : 인건비

#### **1.6 Other Information**

- A. Future Version

계산 결과를 출력한 후에 Category, Value, Constraint를 변경하면 그 결과값 내에서 계산하여 Performance와 유연성을 향상시킨다.

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

### **2.1 Alternative Solutions**

- A. 기존에 있는 Category Partition Tool 을 사용한다.
- B. 개발 경험이 있는 사람에게 도움을 요청한다.
- C. 직접 Test Case 를 작성한다.

## 2.2 Project Justification

- A. Cost : 인건비
- B. Duration : 3 개월
- C. Risk : Java 에 대한 숙련도 부족, UML 숙련도 부족, 시험 기간, 고양이가 아픴, 아르바이트
- D. Effect : 시중의 Tool 보다 비용적인 측면에서 더 저렴하다. 직접 계산하는 것 보다 시간적인 면에서 더 빠르다.

## 2.3 Risk Management

Risk	Probability(5)	Significance	Weight
Java 숙련도 부족	4	5	20
UML 숙련도 부족	5	3	15
고양이가 아픴	2	4	8
시험 기간	5	5	25
아르바이트	2	3	6

## 2.4 Risk Reduction Plan

Risk	Way of Reduction
Java 숙련도 부족	이전에 사용한 자바 교재와 Google을 활용한다.
UML 숙련도 부족	UML Manual과 수업 ppt를 활용한다.
고양이가 아픴	병원을 꾸준히 데려가고, 평소에 잘해준다.
시험 기간	평소에 미리 공부한다.
아르바이트	그만 둔다.

## 2.5 Market Analysis

다른 테스트케이스 제네레이터와 차별화 되는 기능과 프리웨어라는 이점을 가짐으로써 시장성을 확보한다.

## 2.6 Other Managerial issues

이 프로젝트는 2017년 6월 7일까지 완성되어야 한다.

## 3 Activity 1003. Define Requirements

### 3.1 Functional Requirements

Ref	Function	Description
R1.1	Category 추가	Category 객체를 생성한다.
R1.2	Category 수정	Category를 수정한다.
R1.3	Category 삭제	Category 객체를 삭제한다.
R1.4	Value 추가	Value 객체를 생성한다.
R1.5	Value 수정	Value를 수정한다.
R1.6	Value 삭제	Value 객체를 삭제한다.
R1.7	Constraint 추가	Constraint를 추가한다.
R1.8	Constraint 수정	Constraint를 수정한다.
R1.9	Constraint 삭제	Constraint를 삭제한다.
R1.10	Value 우선 순위 선택	Value에 우선 순위를 부여한다.
R2	Test Case 계산	전체 Test Case를 계산한다.
R2.1	우선 순위 계산	우선 순위가 있는 Value에 관련된 Test Case를 먼저 계산하고 그 Test Case가 전체 Test Case중 가지는 비중을 계산한다.
R2.2	진행 상황 계산	전체 Test Case중 얼마나 진행 했는지 계산한다.
R2.3	출력	결과값들을 출력한다.
R2.3.1	우선 순위 출력	우선 순위 Test Case를 화면에 출력한다.
R2.3.2	진행 상황 출력	진행 상황을 화면에 출력한다.
R2.3.3	File 출력	전체 Test Case를 File로 출력한다.
R3.1	초기화	모든 입력과 출력 화면을 초기화한다.
R4.1	프로그램 종료	프로그램을 종료한다.

Ref	Function	Category
R1.1	Category 추가	Evident
R1.2	Category 수정	Evident
R1.3	Category 삭제	Evident
R1.4	Value 추가	Evident
R1.5	Value 수정	Evident
R1.6	Value 삭제	Evident
R1.7	Constraint 추가	Evident
R1.8	Constraint 수정	Evident
R1.9	Constraint 삭제	Evident
R1.10	Value 우선 순위 선택	Evident

R2	Test Case 계산	Evident
R2.1	우선 순위 계산	Hidden
R2.2	진행 상황 계산	Hidden
R2.3	출력	Hidden
R2.3.1	우선 순위 출력	Hidden
R2.3.2	진행 상황 출력	Hidden
R2.3.3	File 출력	Hidden
R3.1	초기화	Evident
R4.1	프로그램 종료	Evident

### 3.2 Operating Environments

Microsoft Windows 7, 10

### 3.3 Develop Environments

- A. OS : Microsoft Windows 10
- B. CPU : Intel
- C. IDE : Eclipse
- D. Language : Java
- E. UML : StarUML

### 3.4 Interface Requirements

- A. 메인 화면
  - i. Category 추가 버튼  
Category 우측에 위치하여 클릭 시 아래 쪽에 Category 를 추가한다.
  - ii. Category 삭제 버튼  
Category 우측에 위치 클릭 시 Category 를 삭제한다.
  - iii. Value 추가 버튼  
Value 우측에 위치하여 클릭 시 아래 쪽에 Value 를 추가한다.
  - iv. Value 삭제 버튼

Value 우측에 위치하여 클릭 시 Value 를 삭제한다.

v. Constraint 추가 버튼

Constraint 우측에 위치하여 클릭 시 Constraint 를 추가한다.

vi. Constraint 삭제 버튼

Constraint 우측에 위치하여 클릭 시 Constraint 를 삭제한다.

vii. Value 체크 박스

Value 좌측에 위치하여 체크 시 우선 순위에 추가한다.

viii. Calculate 버튼

화면 중앙에 위치하여 클릭 시 Test Case 를 계산한다.

ix. Clear 버튼

화면 중앙에 위치하여 클릭 시 모든 입력과 출력 화면을 초기화한다.

B. 결과 출력 화면

i. Test Case 출력 화면

화면 우측에 위치하여 우선 순위 Test Case 값을 출력한다.

ii. 진행 상황 출력 화면

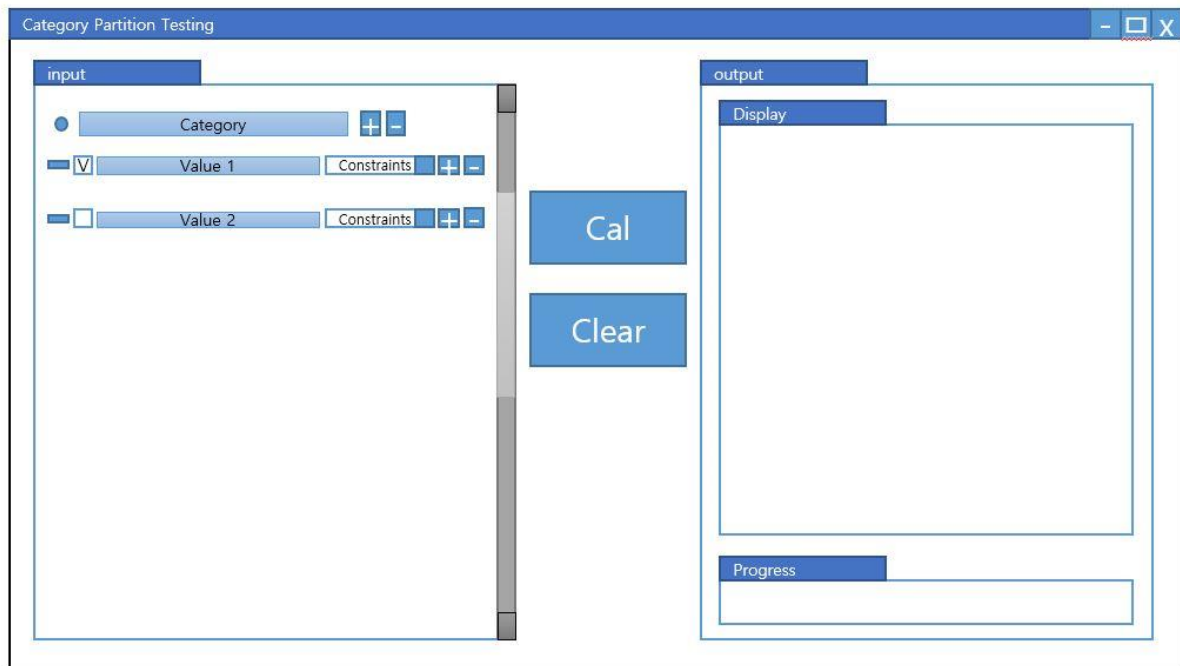
화면 우측 아래에 위치하여 현재 진행 상황을 출력한다.

#### 4 Activity 1004. Record Terms in Glossary

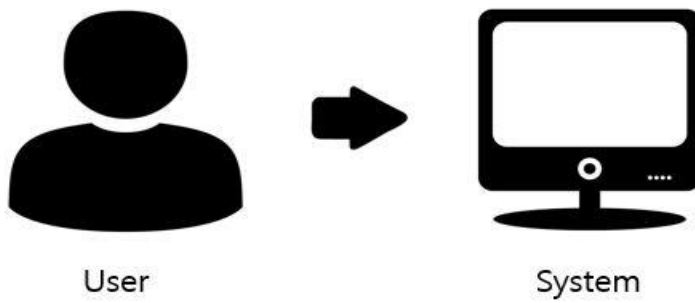
Term	Description	Remarks
Category	Test Case를 생성하는데 사용되는 항목	
Value	한 Category안에서 선택 가능한 값	
Constraints	Value에 조건을 부여하여 Test Case의 Volume을 줄임	



### 5 Activity 1005. Implement Prototype

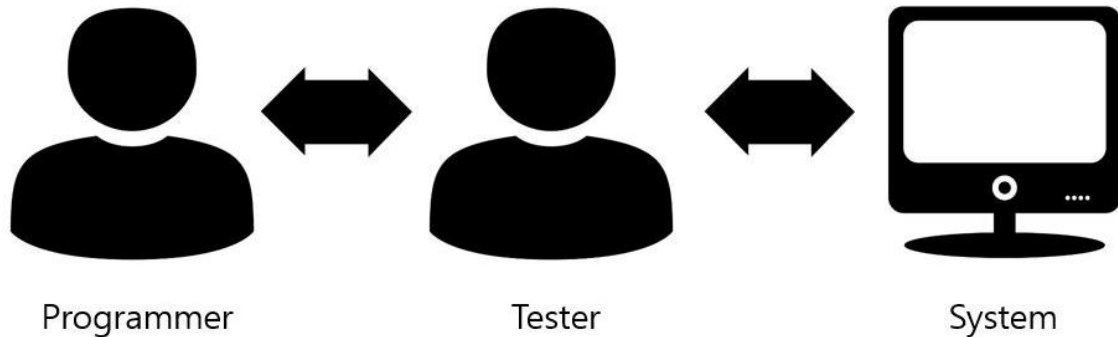


### 6 Activity 1006. Define System Architecture



## 7 Activity 1007. Define Business Use Case

### 7.1 Define System Boundary



### 7.2 Identify and Describe Actors

#### A. Actor(User)

Category, Value, Constraint를 추가/삭제하고, 우선 순위를 부여하여 Test Case를 생성하는 대상

### 7.3 Identify Use-Case

#### 7.3.1 Actor-Based



### 7.3.2 Event-Based



### 7.4 Allocate System Functions into Related Use-Cases

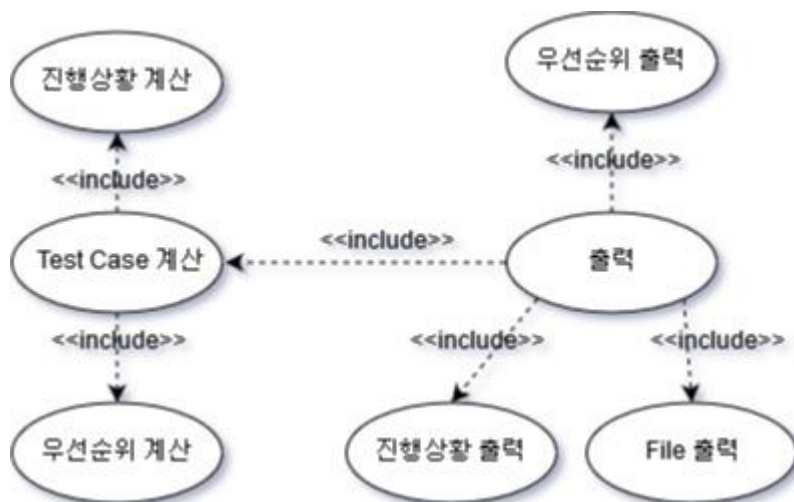
Ref	Function	Use-Case Number & Name
R1.1	Category 추가	1.Category 추가
R1.2	Category 수정	2.Category 수정
R1.3	Category 삭제	3.Category 삭제
R1.4	Value 추가	4.Value 추가
R1.5	Value 수정	5.Value 수정
R1.6	Value 삭제	6.Value 삭제
R1.7	Constraint 추가	7.Constraint 추가
R1.8	Constraint 수정	8.Constraint 수정
R1.9	Constraint 삭제	9.Constraint 삭제
R1.10	Value 우선 순위 선택	10.Value 우선 순위 선택
R2	Test Case 계산	11.Test Case 계산
R2.1	우선 순위 계산	12.우선 순위 계산
R2.2	진행 상황 계산	13.진행 상황 계산
R2.3	출력	14.출력
R2.3.1	우선 순위 출력	15.우선 순위 출력
R2.3.2	진행 상황 출력	16.진행 상황 출력
R2.3.3	File 출력	17.File 출력
R3.1	초기화	18.초기화
R4.1	프로그램 종료	19.프로그램 종료

### 7.5 Categorize Use-Cases

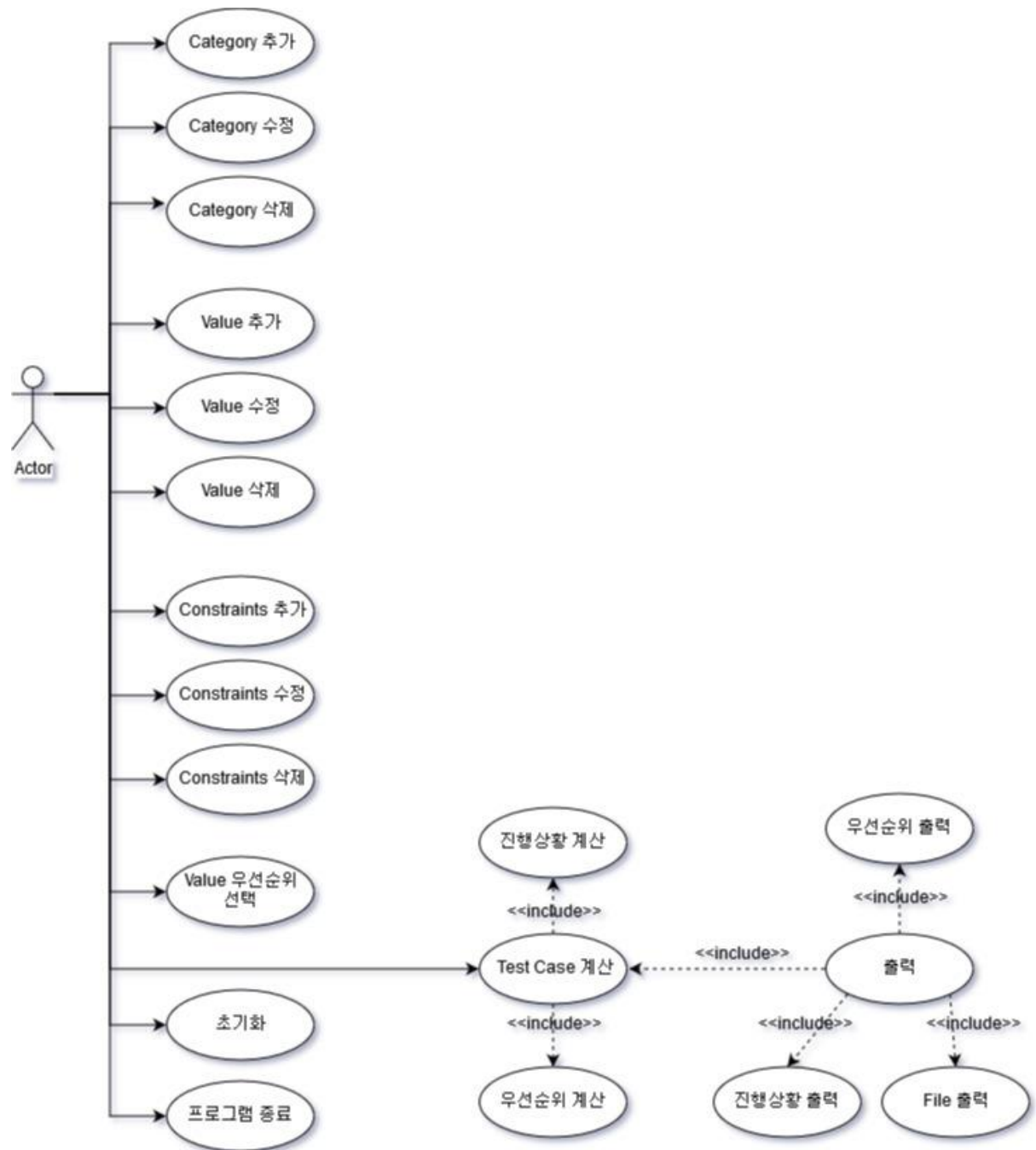
Use-Case Number & Name	Category
1.Category 추가	Primary
2.Category 수정	Primary
3.Category 삭제	Primary
4.Value 추가	Primary

5.Value 수정	Primary
6.Value 삭제	Primary
7.Constraint 추가	Primary
8.Constraint 수정	Primary
9.Constraint 삭제	Primary
10.Value 우선 순위 선택	Primary
11.Test Case 계산	Primary
12.우선 순위 계산	Primary
13.진행 상황 계산	Primary
14.출력	Primary
15.우선 순위 출력	Primary
16.진행 상황 출력	Primary
17.File 출력	Primary
18.초기화	Primary
19.프로그램 종료	Primary

## 7.6 Identify Relationships between Use-Cases



### 7.7 Draw a Use-Case Diagram



### 7.8 Describe Use-Cases

Use-Case Name	Description
1. Category 추가	Category 객체를 생성한다.
<b>Actor</b>	
User	

Use-Case Name	Description
2. Category 수정	Category를 수정한다.
<b>Actor</b>	
User	

Use-Case Name	Description
3. Category 삭제	Category 객체를 삭제한다.
<b>Actor</b>	
User	

Use-Case Name	Description
4. Value 추가	Value 객체를 생성한다.
<b>Actor</b>	
User	

Use-Case Name	Description
5. Value 수정	Value를 수정한다.
<b>Actor</b>	
User	

Use-Case Name	Description
6.Value 삭제	Value 객체를 삭제한다.
<b>Actor</b>	
User	

Use-Case Name	Description
7.Constraint 추가	Constraint를 추가한다.
<b>Actor</b>	
User	

Use-Case Name	Description
8. Constraint 수정	Constraint를 수정한다..
<b>Actor</b>	
User	

Use-Case Name	Description
9. Constraint 삭제	Constraint를 삭제한다.
<b>Actor</b>	
User	

Use-Case Name	Description
10. Value 우선 순위 선택	Value에 우선 순위를 부여한다.
<b>Actor</b>	
User	

Use-Case Name	Description
11.Test Case 계산	Test Case 계산 버튼을 클릭해서 전체 Test Case의 계산을 시작한다.
<b>Actor</b>	
User	

Use-Case Name	Description
12.우선 순위 계산	우선 순위가 있는 Value에 관련된 Test Case를 먼저 계산하고 그 Test Case가 전체 Test Case중 가지는 비중을 계산한다.
<b>Actor</b>	
System	

Use-Case Name	Description
13.진행 상황 계산	전체 Test Case중 얼마큼 진행 했는지 '%'로 출력한다.
<b>Actor</b>	
System	

Use-Case Name	Description
14.출력	결과값들을 출력한다.
<b>Actor</b>	
System	

Use-Case Name	Description
15.우선 순위 출력	우선 순위 Test Case를 화면에 출력한다.
<b>Actor</b>	
System	

Use-Case Name	Description
16.진행 상황 출력	진행 상황을 화면에 출력한다.
<b>Actor</b>	
System	

Use-Case Name	Description
17.File 출력	전체 Test Case를 File로 출력한다.
<b>Actor</b>	
System	

Use-Case Name	Description
18.초기화	모든 입력과 출력 화면을 초기화한다.
<b>Actor</b>	
User	

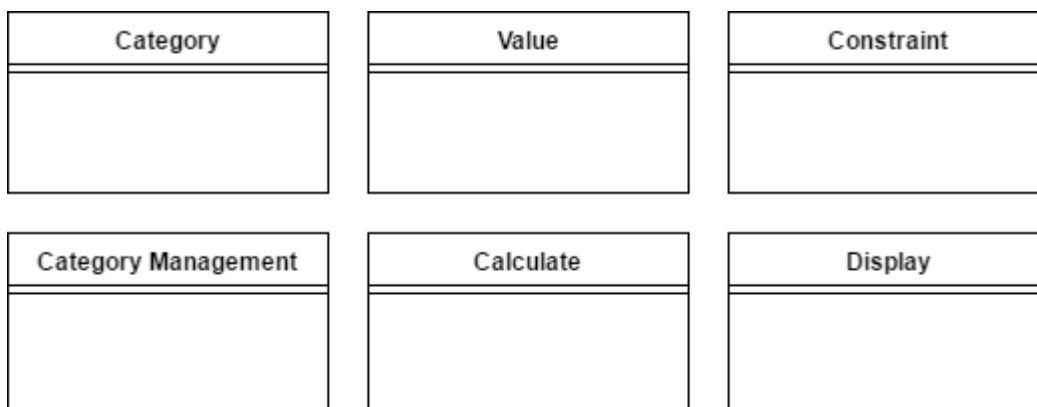
Use-Case Name	Description
19.프로그램 종료	프로그램을 종료한다.
<b>Actor</b>	
User	



## 7.9 Rank Use-Cases

Use-Case Number & Name	Rank
1.Category 추가	High
2.Category 수정	High
3.Category 삭제	High
4.Value 추가	High
5.Value 수정	High
6.Value 삭제	High
7.Constraint 추가	High
8.Constraint 수정	High
9.Constraint 삭제	High
10.Value 우선 순위 선택	High
11.Test Case 계산	High
12.우선 순위 계산	High
13.진행 상황 계산	High
14.출력	High
15.우선 순위 출력	High
16.진행 상황 출력	High
17.File 출력	High
18.초기화	High
19.프로그램 종료	High

## 8 Activity 1008 Define Business Concept Model



## 9 Activity 1009 Define System Test Case

### 9.1 Functional Requirements Test Case

Ref	Function	Test Case
R1.1	Category 추가	Category 추가 버튼을 눌렀을 때 Category 객체가 생성되는지 확인한다.
R1.2	Category 수정	Category가 수정되는지 확인한다.
R1.3	Category 삭제	Category 삭제 버튼을 눌렀을 때 Category 객체가 삭제되는지 확인한다.
R1.4	Value 추가	Value 추가 버튼을 눌렀을 때 Value 객체가 생성 되는지 확인한다.
R1.5	Value 수정	Value가 수정되는지 확인한다.
R1.6	Value 삭제	Value 삭제 버튼을 눌렀을 때 Value 객체가 생성 되는지 확인한다.
R1.7	Constraint 추가	Constraint 추가 버튼을 눌렀을 때 Constraint 객체가 생성 되는지 확인한다.
R1.8	Constraint 수정	Constraint가 수정되는지 확인한다.
R1.9	Constraint 삭제	Constraint 추가 버튼을 눌렀을 때 Constraint 객체가 생성 되는지 확인한다.
R1.10	Value 우선 순위 선택	Value 박스를 체크하고 우선 순위가 부여되는지 확인한다.
R2	Test Case Calculate	우선 순위 Test Case, 전체 Test Case, 진행 상황의 결과값을 예상 결과값과 비교한다.
R2.1	우선 순위 Calculate	우선순위 Constraint를 변경하면서 예상한 결과값과 계산 후 얻어지는 결과값을 비교한다.
R2.2	진행 상황 Calculate	진행 상황이 출력 되는지 확인한다.
R2.3	출력	결과 값이 출력화면에 나오는지 확인한다.
R2.3.1	우선 순위 출력	우선 순위로 선택한 Test Case가 출력되는지 확인한다.
R2.3.2	진행 상황 출력	진행 상황이 %로 출력되는지 확인한다.
R2.3.3	File 출력	결과값을 저장한 파일이 출력되는지 확인한다.
R3.1	초기화	모든 값과 출력 화면이 초기화 되는지 확인한다.
R4.1	프로그램 종료	프로그램이 종료 되는지 확인한다.

## 10 Activity 1010 Refine Plan

### 10.1 Project Scope

**테스터가 지정한 값을 중심으로 빠르게 결과값을 보여줄 수 있는 Test Case Generator**

여러 개의 테스트 케이스들을 한꺼번에 계산 하는 데는 큰 부담이 걸리고, 속도도 느려질 수 밖에 없다. 이러한 문제를 해결하기 위해 당장 필요한 테스트 케이스들을 우선적으로 받아 볼 수 있는 프로그램을 개발한다.

### 10.2 Project Objective

원하는 테스트 케이스를 우선적으로 확인할 수 있게 하여, 총 테스트에 걸리는 시간을 줄이는데 도움을 주는 프로그램을 목표로 한다.

### 10.3 Functional Requirements

- A. Category 추가
- B. Category 수정
- C. Category 삭제
- D. Value 추가
- E. Value 수정
- F. Value 삭제
- G. Constraint 추가
- H. Constraint 수정
- I. Constraint 삭제
- J. Value 우선 순위 선택
- K. Test Case 계산
- L. 우선 순위 계산
- M. 진행 상황 계산
- N. 출력

- O. 우선 순위 출력
- P. 진행 상황 출력
- Q. File 출력
- R. 초기화
- S. 프로그램 종료

#### **10.4 Non-Functional Requirements**

- A. 조작하기 쉽고 익숙한 인터페이스
- B. 우선 순위 계산이 전체 계산보다 빠르게 출력

#### **10.5 Operating Environment**

OS : Windows 7,10

IDE : Eclipse

Language : Java

#### **10.6 Resources**

- A. Human Resource : 4 명
- A. Project Duration : 3 개월(12 주)
- A. Human Effort(Man-Month) : 12
- A. Cost : 인건비

### 10.7 Scheduling

Stage	Phase(00x0)/Activity(000x)	Schedule(Week)													
		1	2	3	4	5	6	7	8	9	10				
1000. Plan & Elabo rate	1001. Define Draft Plan	█													
	1002. Create Preliminary Investigation Report														
	1003. Define Requirements	█													
	1004. Record Terms in Glossary	█	█												
	1005. Implement Prototype	█	█												
	1006. Define System Architecture	█	█												
	1007. Define Use Cases	█	█												
	1008. Define Draft Conceptual Model	█	█												
	1009. Define Draft System Architecture	█	█												
	1010. 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											█	█			