Software Modeling & Analysis

Category Partitioning test cases with User

OOPT stage 1000

Plan & Elaboration

Team No	Team5
과목	Software Modeling and Analysis
담당교수	JUNBEOM YOO Associate Professor / Ph.D
팀 구성원	201111383 전훈
	201311259 권오승
	201311292 유효상
	201511266 배율희
제출일자	2017-3-30

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 Draft System Architecture**

Activity 1007. **Define Business Use Case**

Activity 1009. **Define System Test Case**

Activity 1010. **Refine Plan**

Activity 1001. Define Draft Plan

1. Motivation

소프트웨어의 개발후 올바른 작동을 확인하고 오류를 탐색하기 위한 테스트가 필수적이다. 하지만 모든 경우를 테스트하기에는 시간적, 재정적 낭비가 심하게 된다. 이러한 경우엔, Category Partitioning Testing 테크닉을 이용하여 중복되는 테스트 케이스를 줄일 수 있다. 효율적으로 테스트 케이스의 수가 감소 했는지 확인 하는 것은 Category Partitioning 을 한 후 필수적이며, 사용자들이 보다 보기 쉽고 사용하기 쉬운 Category Partitioning Test Case 계산 Tool 이 필요하다. 그에 따라, Category Partitioning Testing Tool 을 개발하게 되었다

2. Project Objectives

사용자가 정의한 카테고리에 따른 Check configuration 을 입력 받아 해당 테스트 케이스의 개수를 알려주는 Category Partitioning Testing Tool 이다.

- Test case 계산후, 전체 case, error 혹은 single 값을 뺀 test case 등 사용자가 원하는 상황에 따른 case 계산 출력 제공

- 차트를 통해, Test case 를 비교 가능

3. Functional requirements

- Display Test List
- New Test
- Load Test
- Edit Test
- Delete Test
- Display Category List
- New Category
- Edit Category
- Delete Category
- Display Value Property
- Add Property
- Delete Property
- Display Value List
- Save Value
- Delete Value
- Save
- Calculate

- Single Check
- Error Check
- Property Check
- Multiply Values
- Exception Detection
- Show Chart

4. Non-functional requirements

- 쉽게 사용할 수 있는 UI

5. Resource Estimation

5.1 Human effort (M/M)

4M/4M

5.2 Human resource

4 명- 프로그래머 4 명

5.3 Duration

3 개월

5.4 Budget

T.F Wage : 30 * 4man * 3month = 360

Activity 1002. Create Preliminary Investigation Report

1. Alternative solution

1.1 시중에 판매하는 Category Partitioning Testing Tool 을 구매한다.

1.2 개발업체에 제작을 의뢰한다.

2. Project justification

1) Cost : 3,600,000
2) Duration : 3 Month
3) Risk : OOPT 에 대한 이해력 부족. Category Partitioning Testing 에 대한 지식 부족, 전공과목의 과제 폭탄, 학생회 활동, 동아리 활동, 연애
4) Effect : 프로그램 유지 보수가 수월하다. 사용방법이 간단하다. 비용이 저렴하다

3. Risk & Risk reduction plan

3.1 Risk Management

Risk	Probability	Significance	Weight
Un-Skilled OOPT	4	6`	17
Lack of knowledge Category Partitioning Testing	3	7	9
Assignments	6	3	10
Dating	2	4	6
Other School Activities	2	8	12

3.2 Risk Reduction Plan

Risk	Reduction Plan
Un-Skilled OOPT	교수님 혹은 지인에게 질문한다
Lack of knowledge Category Partitioning Testing	Category Partitioning Testing 에 대하여 공부한다
Assignments	동기들과 협동하여 과제를 해결한다, 또는 선택과 집중을 한다
Dating	헤어진다
Other School Activities	뒤로 미뤘다가 한번에 처리한다

4. Market Analysis

대규모의 소프트웨어 프로그램이 많아짐에 따라 테스트 케이스가 많아지고 있음. 따라서 테스트 케이스를 줄여주는 테스트 프로그램의 요구가 증가하고 있음

5. Managerial issue

- 2017 년 5 월까지 proto-type 개발이 완료되어야 한다.
- 2017 년 7 월까지 Final version 개발이 완료되어야 한다.

Activity 1003. Define Requirements

1. Functional Requirements

Function	Description
Display Test List	Test 목록을 보여준다
New Test	새로운 Test 를 생성한다
Load Test	Test 파일을 List 에 Load 한다
Edit Test	Test 파일의 이름을 수정한다
Delete Test	Test 파일을 List 에서 삭제한다
Display Category List	Category 목록을 보여준다
New Category	새로운 Category 를 생성한다
Edit Category	Category 의 정보를 수정한다
Delete Category	Category 를 삭제한다
Display Value Property	Value 의 Property 를 보여준다
Add Property	새로운 Property 를 생성한다
Delete Property	Property 를 삭제한다
Display Value List	Value 목록을 보여준다
Save Value	설정된 Value 를 저장한다
Delete Value	Value 를 삭제한다
Save	Test 의 모든 정보를 파일로 저장한다
Calculate	Test Case 를 계산한다
Single Check	Single 에 해당되는 경우를 확인한다
Error Check	Error 가 해당되는 경우를 확인한다
Property Check	Property 가 해당되는 경우를 확인한다
Multiply Values	Value 의 곱연산을 실행한다
Exception Detection	오류 발생을 탐색한다
Show Chart	테스트 케이스들을 차트 형식으로 보여준다.

2. Functional Requirements (Categorized Table)

Ref. #	Function	Category
R1.1	Display Test List	Hidden
R1.2	New Test	Evident
R1.3	Load Test	Evident
R1.4	Edit Test	Evident
R1.5	Delete Test	Evident
R2.1	Display Category List	Hidden
R2.2	New Category	Evident
R2.3	Edit Category	Evident
R2.4	Delete Category	Evident
R3.1	Display Value Property	Hidden
R3.2	Add Property	Evident
R3.3	Delete Property	Evident
R3.4	Display Value List	Hidden
R3.5	Save Value	Evident
R3.6	Delete Value	Evident
R4.1	Save	Evident
R5.1	Calculate	Evident
R5.2	Single Check	Hidden
R5.3	Error Check	Hidden
R5.4	Property Check	Hidden
R5.5	Multiply Values	Hidden
R5.6	Exception Detection	Hidden
R6.1	Show Chart	Hidden

3. Performance Requirements

- Test case 계산은 3 초를 넘기지 말아야한다

4. Operating Environment

Source	Case1	Case2	Case3
Operating Software	Win7 - 64bit	Win10 - 64bit	Mac OS - 64bit
CPU	i5-4569	i3-7100U	i5-5257U
RAM	16GB	8GB	8GB
Library	128GB SSD	256GB SSD	256GB SSD

Develop Environment: Eclipse NEON/NEON/MARS

5. Interface Requirement

- Test File, Category, Value 가 입력, 수정, 삭제가 가능하여 한다
- Test Case 의 모든 결과를 보여주되, 사용자의 요구에 따라 필터된 결과도 차트로 보여줄 수 있어야 한다
- Test Case 의 효율성을 보여주기 위해, 차트가 사용되어야 한다

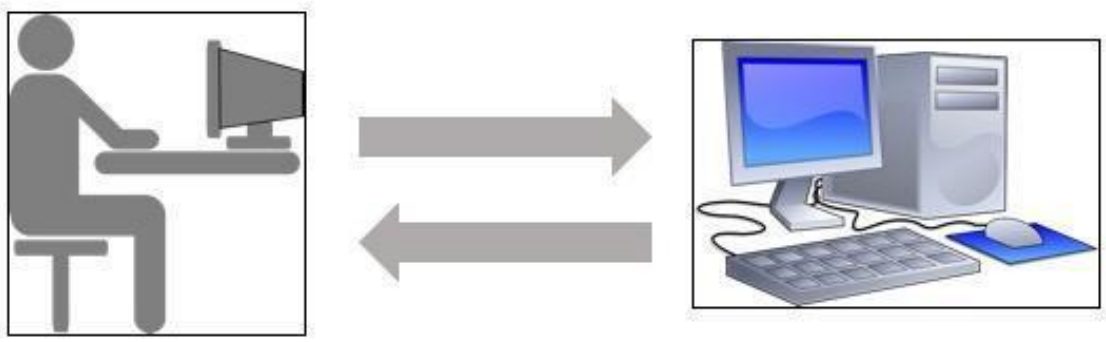
6. Other Requirements

N/A

Activity 1004. Record Terms in Glossary

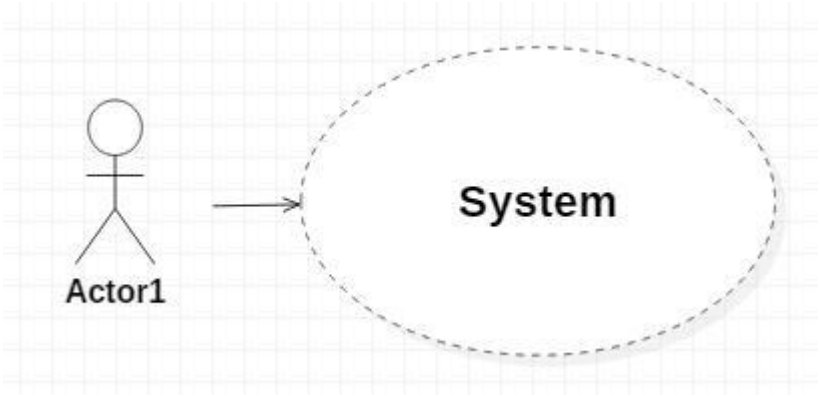
Term	Description	Remarks
User	Category Partitioning test 프로그램을 사용하여 test cases 를 만드는 객체, 판단하는 객체	
Value	Category 에서 선언된 Value 의 객체	
Category	Test File 에서 선언된 Category 의 객체	
Test File	Test case 의 실제 File	

Activity 1006. Define Draft System Architecture



Activity 1007. Define Business Use Case

1. Define system boundary

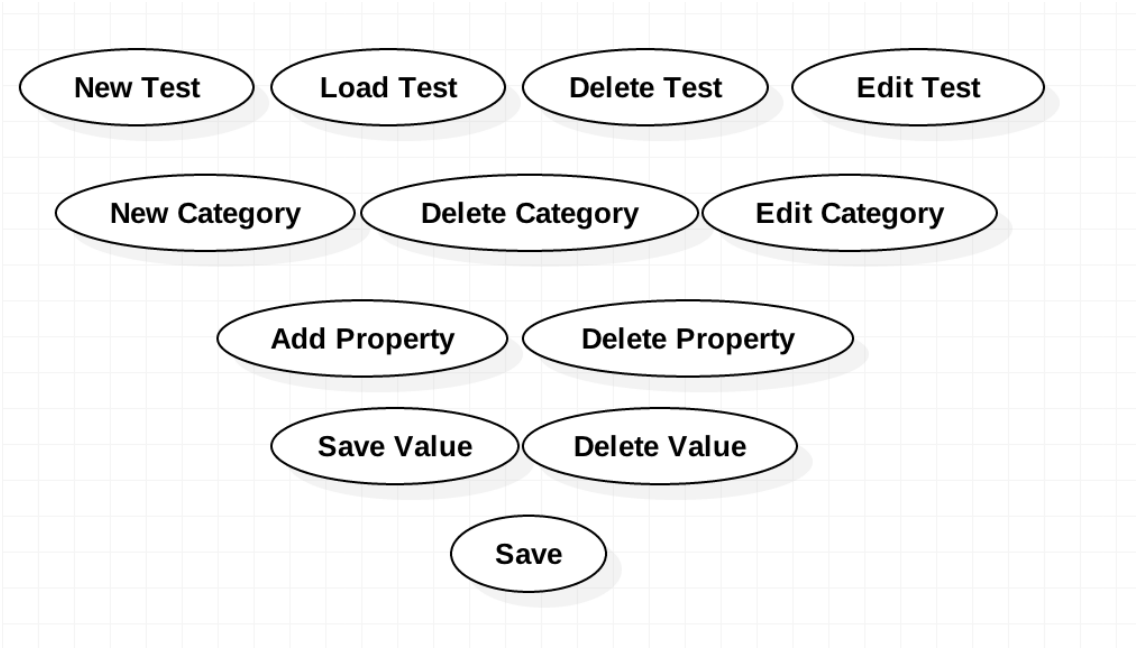


2. Identify and describe actors

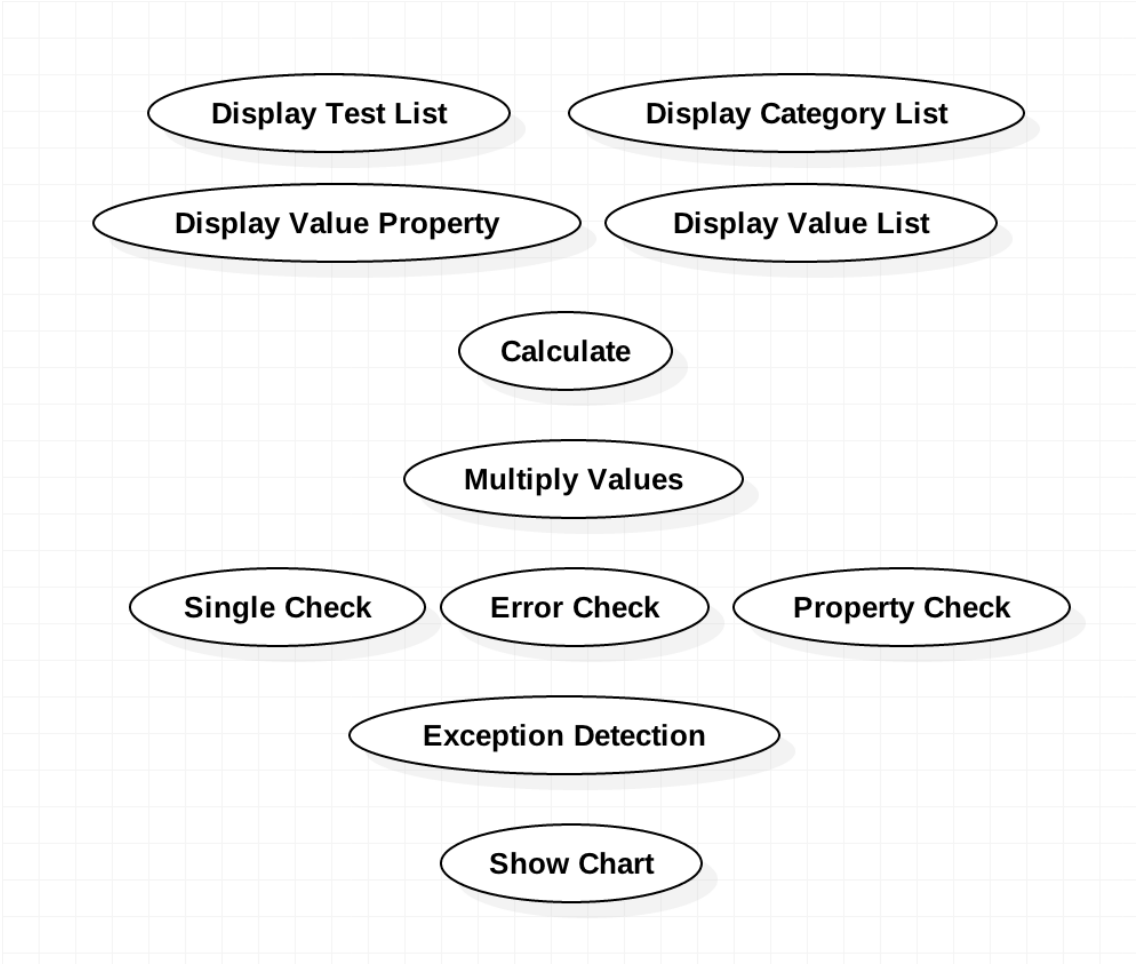
- User : Check configuration 을 입력하여 Category Partitioning Test 결과를 요구한다.

3. Identify use cases

3.1 Actor-Based



3.2 Event-Based



4. Allocate system functions into related use cases

Ref. #	Function	Use Case Number & Names	Category
R1.1	Display Test List	1. Display Test List	
R1.2	New Test	2. New Test	
R1.3	Load Test	3. Load Test	
R1.4	Edit Test	4. Edit Test	
R1.5	Delete Test	5. Delete Test	
R2.1	Display Category List	6. Display Category List	
R2.2	New Category	7. New Category	
R2.3	Edit Category	8. Edit Category	
R2.4	Delete Category	9. Delete Category	
R3.1	Display Value Property	10. Display Value Property	

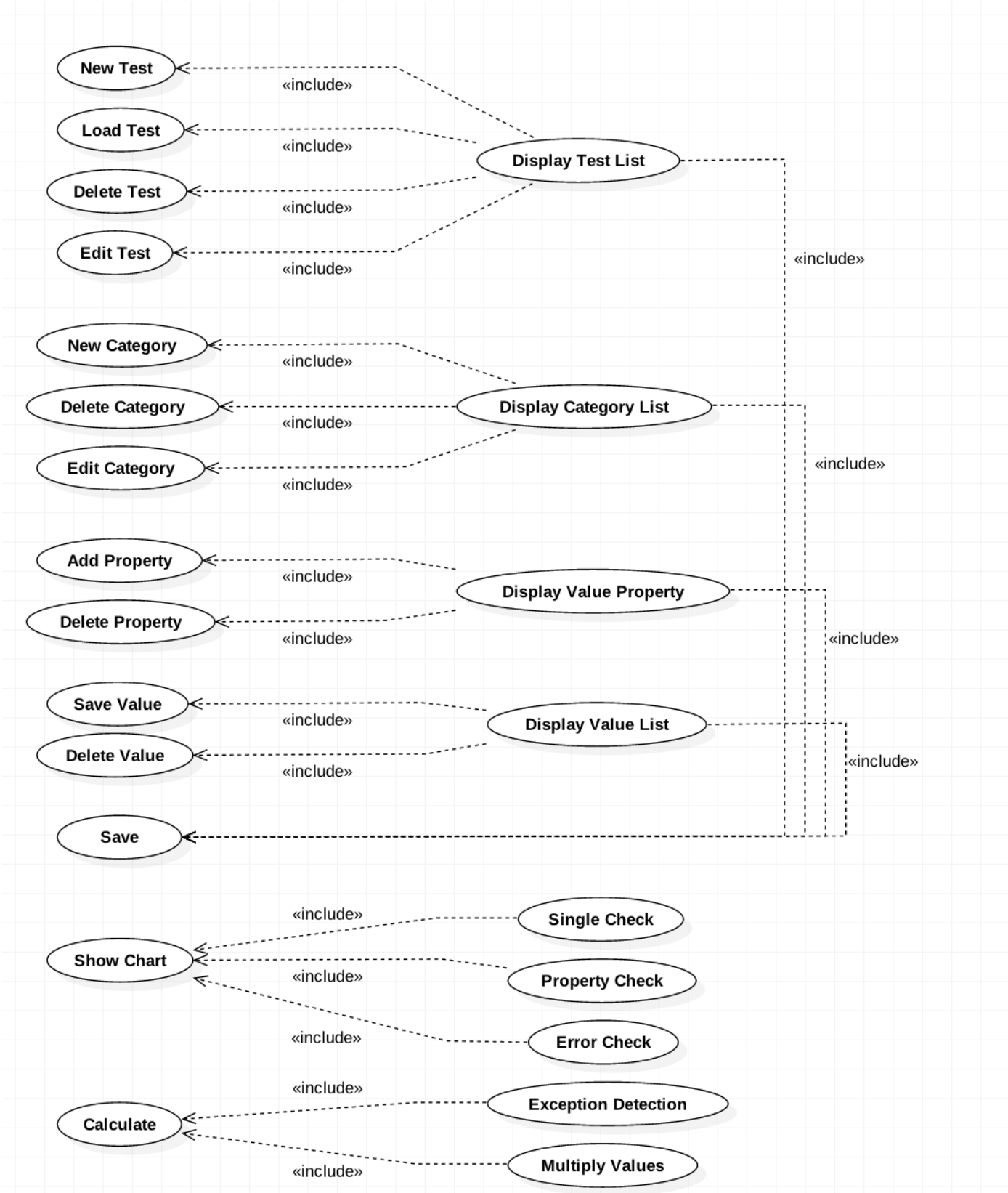
R3.2	Add Property	11. Add Property	
R3.3	Delete Property	12. Delete Property	
R3.4	Display Value List	13. Display Value List	
R3.5	Save Value	14. Save Value	
R3.6	Delete Value	15. Delete Value	
R4.1	Save	16. Save	
R5.1	Calculate	17. Calculate	
R5.2	Single Check	18. Single Check	
R5.3	Error Check	19. Error Check	
R5.4	Property Check	20. Property Check	
R5.5	Multiply Values	21. Multiply Values	
R5.6	Exception Detection	22. Exception Detection	
R6.1	Show Chart	23. Show Chart	

5. Categorize use cases

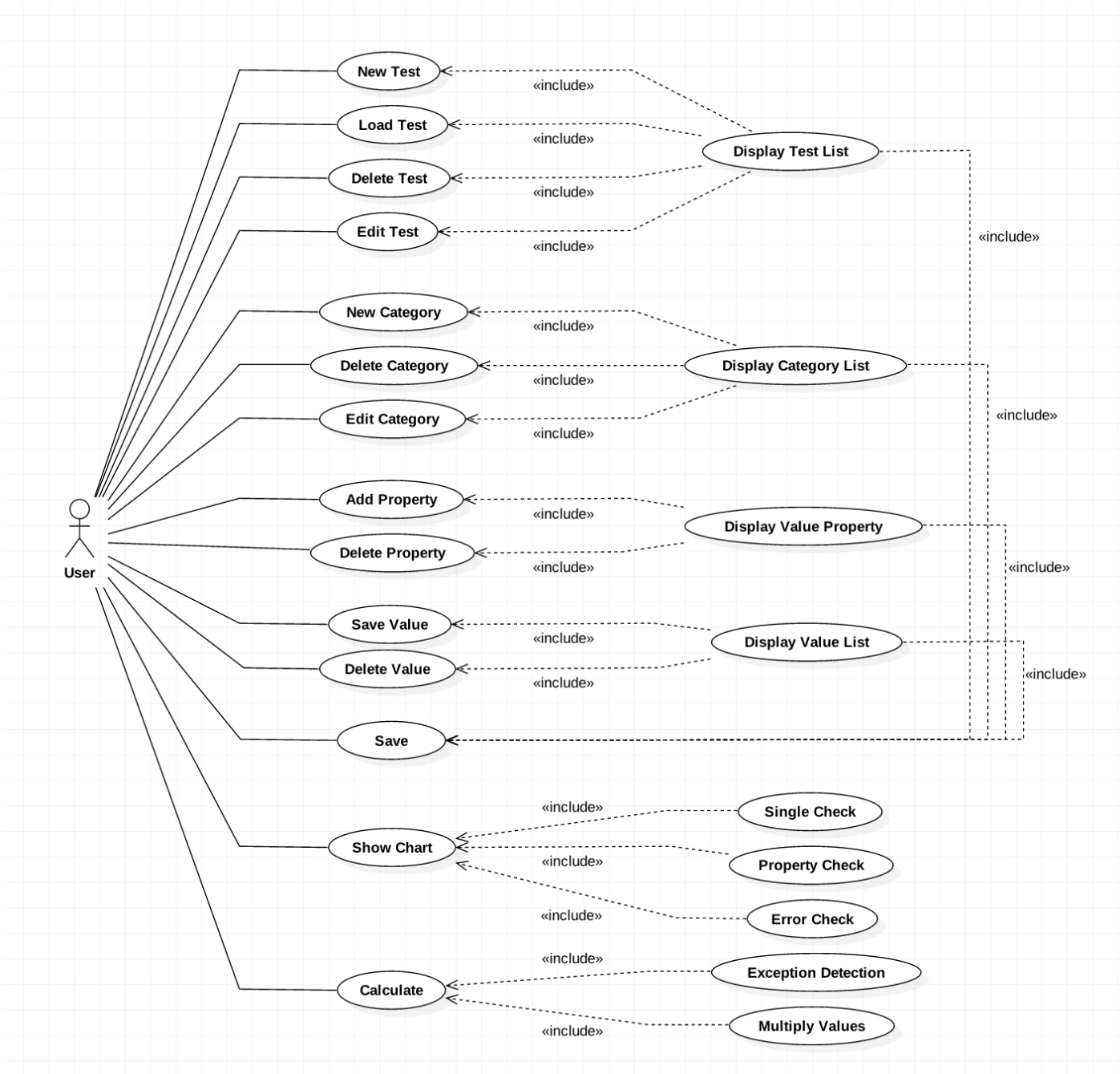
Ref. #	Function	Use Case Number & Names	Category
R1.1	Display Test List	1. Display Test List	Primary
R1.2	New Test	2. New Test	Primary
R1.3	Load Test	3. Load Test	Primary
R1.4	Edit Test	4. Edit Test	Primary
R1.5	Delete Test	5. Delete Test	Primary
R2.1	Display Category List	6. Display Category List	Primary
R2.2	New Category	7. New Category	Primary
R2.3	Edit Category	8. Edit Category	Primary
R2.4	Delete Category	9. Delete Category	Primary
R3.1	Display Value Property	10. Display Value Property	Primary
R3.2	Add Property	11. Add Property	Primary
R3.3	Delete Property	12. Delete Property	Primary
R3.4	Display Value List	13. Display Value List	Primary
R3.5	Save Value	14. Save Value	Primary
R3.6	Delete Value	15. Delete Value	Primary
R4.1	Save	16. Save	Primary

R5.1	Calculate	17. Calculate	Primary
R5.2	Single Check	18. Single Check	Primary
R5.3	Error Check	19. Error Check	Primary
R5.4	Property Check	20. Property Check	Primary
R5.5	Multiply Values	21. Multiply Values	Primary
R5.6	Exception Detection	22. Exception Detection	Primary
R6.1	Show Chart	23. Show Chart	Primary

6. Identify relationships between use cases



7. Draw a use case diagram



8. Describe use cases

Use Case	Display Test List
Actor	Event
Description	- Test 목록을 보여준다.

Use Case	New Test
Actor	User
Description	- 새로운 Test 를 추가한다. - 팝업 창에서 Test 이름을 입력한다.

Use Case	Load Test
Actor	User
Description	- 일정 파일 형식의 txt 파일을 불러온다. - 파일 탐색기를 이용한다.

Use Case	Edit Test
Actor	User
Description	- 팝업 창에서 Test 의 이름을 변경한다.

Use Case	Delete Test
Actor	User
Description	- Test 를 제거한다.

Use Case	Display Category List
Actor	Event
Description	- Category 목록을 보여준다.

Use Case	New Category
Actor	User
Description	- 새로운 Category 를 추가한다. - 팝업 창에서 Category 이름을 입력한다.

Use Case	Edit Category
Actor	User
Description	- 팝업 창에서 Category 의 이름을 변경한다.

Use Case	Delete Category
Actor	User
Description	- Category 를 제거한다.

Use Case	Display Value Property
Actor	Event
Description	- Value 의 Property 를 보여준다.

Use Case	Add Property
Actor	User
Description	- 새 Property 를 추가한다. - 팝업 창에서 property 의 이름, if 여부를 입력한다.

Use Case	Delete Property
Actor	User
Description	- Property 를 제거한다.

Use Case	Display Value List
Actor	User
Description	- Value 목록을 보여준다.

Use Case	Save Value
Actor	User
Description	- 설정된 Value 를 저장한다.

Use Case	Delete Value
Actor	User
Description	- Value 를 제거한다.

Use Case	Save
Actor	User
Description	- 현재 Test 상태를 txt 파일 형식으로 저장한다.

Use Case	Calculate
Actor	User
Description	- 테스트 케이스를 계산을 시작한다.

Use Case	Single Check
Actor	Event
Description	- [single] 항목을 확인한다.

Use Case	Error Check
Actor	Event
Description	- [error] 항목을 확인한다.

Use Case	Property Check
Actor	Event
Description	- property 조합을 확인한다.

Use Case	Multiply Values
Actor	Event
Description	- Value 의 곱연산을 실행한다.

Use Case	Exception Detection
Actor	Event
Description	- 오류 발생을 탐색한다.

Use Case	Show Chart
Actor	Event
Description	- 계산한 테스트 케이스들을 차트 형식으로 보여준다.

9. Rank use cases

Ref. #	Function	Use Case Number & Names	Category	Rank
R1.1	Display Test List	1. Display Test List	Primary	High
R1.2	New Test	2. New Test	Primary	High
R1.3	Load Test	3. Load Test	Primary	High
R1.4	Edit Test	4. Edit Test	Primary	High
R1.5	Delete Test	5. Delete Test	Primary	High
R2.1	Display Category List	6. Display Category List	Primary	High
R2.2	New Category	7. New Category	Primary	High
R2.3	Edit Category	8. Edit Category	Primary	High
R2.4	Delete Category	9. Delete Category	Primary	High
R3.1	Display Value Property	10. Display Value Property	Primary	High
R3.2	Add Property	11. Add Property	Primary	High
R3.3	Delete Property	12. Delete Property	Primary	High
R3.4	Display Value List	13. Display Value List	Primary	High
R3.5	Save Value	14. Save Value	Primary	High
R3.6	Delete Value	15. Delete Value	Primary	High
R4.1	Save	16. Save	Primary	High
R5.1	Calculate	17. Calculate	Primary	High
R5.2	Single Check	18. Single Check	Primary	High
R5.3	Error Check	19. Error Check	Primary	High
R5.4	Property Check	20. Property Check	Primary	High
R5.5	Multiply Values	21. Multiply Values	Primary	High
R5.6	Exception Detection	22. Exception Detection	Primary	High
R6.1	Show Chart	23. Show Chart	Primary	High

Activity 1009. Define System Test Case

Use-Case No.	Use-Case	Test Plan
1	Display Test List	Test 목록이 화면에 보여지는지 확인한다.
2	New Test	User 가 새로운 Test 를 Test List 에 추가되는지 확인한다.
3	Load Test	User 가 txt 파일 형식의 Test 를 불러와서 List 에 추가되는지 확인한다.
4	Edit Test	User 가 팝업 창에서 Test 의 이름을 Edit 되는지 확인한다.
5	Delete Test	User 가 Test List 에서 Test File 이 Delete 가 되는지 확인한다.
6	Display Category List	Category 목록이 화면에 보여지는지 확인한다.
7	New Category	User 가 새로운 Category 를 Category List 에 add 되는지 확인한다.
8	Edit Category	User 가 팝업 창에서 Category 의 이름을 Edit 되는지 확인한다.
9	Delete Category	User 가 Category List 에서 Category 를 Delete 되는지 확인한다.
10	Display Value Property	Value 의 Property 항목이 화면에 보여지는지 확인한다.
11	Add Property	User 가 새로운 Property 를 add 되는지 확인한다.
12	Delete Property	User 가 Property 를 Delete 되는지 확인한다.
13	Display Value List	Value 목록이 화면에 보여지는지 확인한다.
14	Save Value	User 가 설정된 Value 를 Save 되는지 확인한다.
15	Delete Value	User 가 입력된 Value 를 Delete 되는지 확인한다.
16	Save	User 가 Test 상태를 txt 파일 형식으로 Save 되는지 확인한다.
17	Calculate	User 가 Test cases 수를 새로운 팝업창으로 Display 되는지 확인한다.
18	Single Check	Single Constraint 항목이 Check 되는지 확인한다.
19	Error Check	Error Constraint 항목이 Check 되는지 확인한다.
20	Property Check	Property Constraint 항목 조합이 Check 되는지 확인한다.
21	Multiply Values	Test cases 수를 Multiply 되는지 확인한다.
22	Exception Check	예기치 못한 Error 에 대한 Exception Check 되는지 확인한다.
23	Show Chart	Test case 가 차트 형식으로 알맞게 변환되어 출력되는지 확인한다.

Activity 1010. Refine Plan

1. Project Scope

Category Partitioning Testing Technique 을 이용 Test Case 를 계산한다.

2. Project Objectives

전체적인 Test Case 를 계산하지만, 사용자의 요구에 따라 single, error, property 에 해당하는 case 를 차트를 이용해 볼 수 있는 Testing Tool 을 개발한다.

3. Functional Requirements

- Display Test List
- New Test
- Load Test
- Edit Test
- Delete Test
- Display Category List
- New Category
- Edit Category
- Delete Category
- Display Value Property
- Add Property
- Delete Property
- Display Value List
- Save Value
- Delete Value
- Save
- Calculate
- Single Check
- Error Check
- Property Check
- Multiply Values
- Exception Detection
- Show Chart

4. Performance Requirements

- Test case 계산은 3 초를 넘기지 말아야한다

5. Operating Environment

Source	Case1	Case2	Case3
Operating Software	Win7 - 64bit	Win10 - 64bit	Mac OS - 64bit
CPU	i5-4569	i3-7100U	i5-5257U
RAM	16GB	8GB	8GB
Library	128GB SSD	256GB SSD	256GB SSD

Develop Environment: Eclipse NEON/NEON/MARS

6. User Interface Requirements

- Test File, Category, Value 가 입력, 수정, 삭제가 가능하여 한다
- Test Case 의 모든 결과를 보여주되, 사용자의 요구에 따라 필터된 결과도 차트로 보여줄 수 있어야 한다
- Test Case 의 효율성을 보여주기 위해, 차트가 사용되어야 한다

7. Other Requirements

N/A

8. Resources

8.1 Human effort (M/M)

4M/4M

8.2 Human resource

4 명- 프로그래머 4 명

8.3 Duration

3 개월

8.4 Budget

T.F Wage : 30 * 4man * 3month = 360

9. Scheduling

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 Business Use Case		■	■							
	1007. Define Business Concept Model			■							
	1008. Define Draft System Architecture			■							
	1009. Define System Test Case			■							
	1010. Refine Plan			■	■						
2000. Build	2110. Revise Plan			■	■	■					
	2120. Synchronize Artifacts			■	■	■					
	2130. Analyze										
	2131. Define Essential Use Case					■					
	2132. Refine Use Case Diagram					■					
	2133. Refine Conceptual Model					■					
	2134. Refine Glossary					■					
	2135. Define System Sequence Diagram					■	■				
	2136. Define Operation Contracts					■	■				
	2137. Define State Diagrams					■	■				
	2140. Design										
	2141. Define Real Use Case						■				
	2142. Define Reports, UI and Story boards						■				
	2143. Refine System Architecture						■	■			
	2144. Define Interaction Diagram						■	■			
	2145. Define Design Class Diagrams							■			
	2150. Construct										
	2151. Implement Class & Interface Definition							■			
	2152. Implement Methods							■			
	2153. Implement Windows							■	■		
	2154. Implement Reports							■	■		
	2155. Write Test Code							■	■		
	2160. Test										
	2161. Unit Testing									■	
	2162. Integration Testing									■	■
	2163. System Testing										■
	2164. Performance Testing										■
	2165. Acceptance Testing										■
	2166. Documentation Testing										■