



Priority

Category Partitioning Testing Tool

FINAL Team 4 PRESENTATION

201311265 김상원
201311269 김제헌
201311297 이상명
201210194 김정환



Team4

Presentation Index

- 1. Stage Review**
 - Stage 1000 Review**
 - Stage 2030 Review**
 - Stage 2040 Review**
 - Stage 2050 & 2060 Review**
- 2. Feedback**
 - T1 feedback**
 - T2 feedback**
- 3. Epilogue**
- 4. Demonstration**



Stage Review

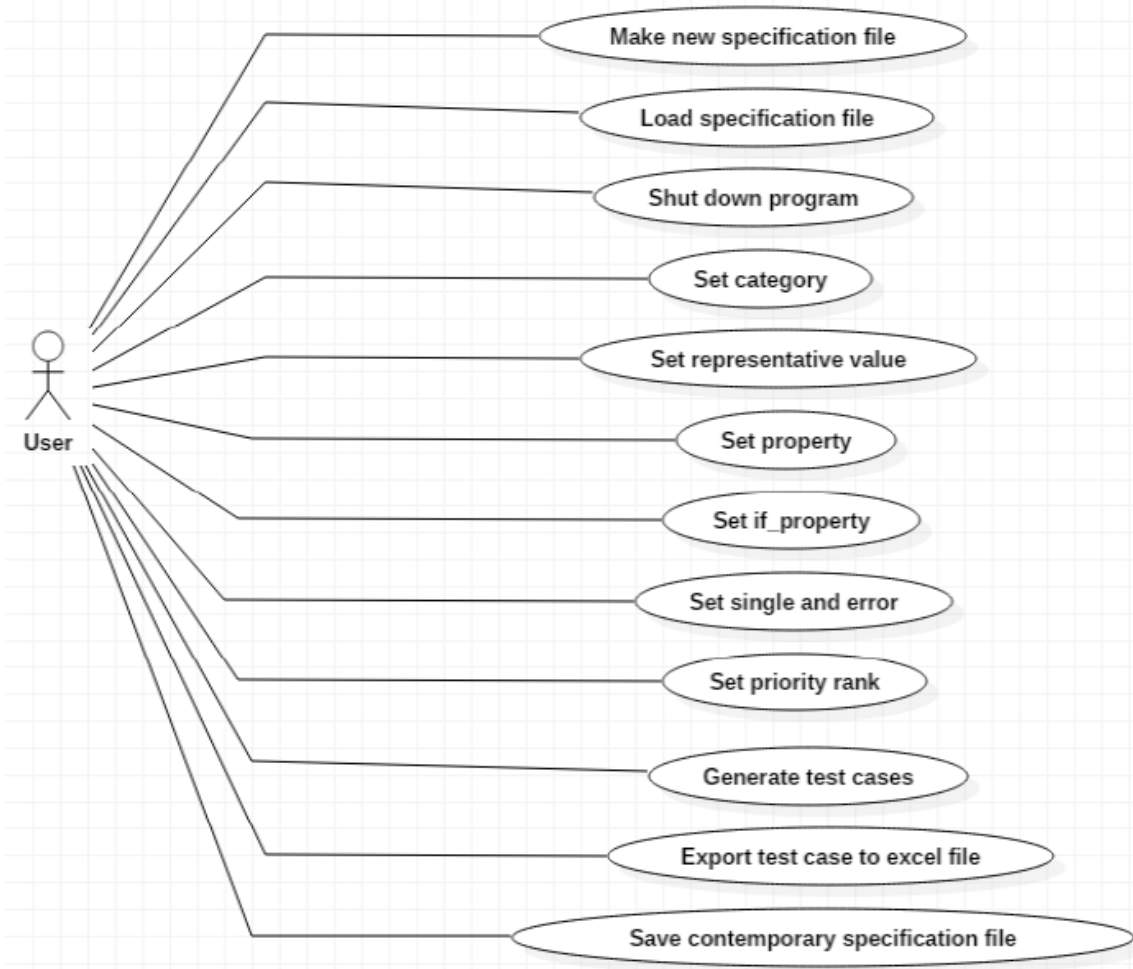
Stage 1000 Review



Activity 1003. Define Requirements 중 Functional Requirements

Function	Description
New specification file	새로운 명세파일을 만든다.
Load specification file	탐색기를 통해서 또는 최근 실행 목록에서 기존의 명세파일을 읽는다.
Shut down program	프로그램을 종료한다.
Set category	Category를 추가 또는 삭제, 변경한다.
Set representative values	representative value를 추가 또는 삭제, 변경한다.
Set property	Property constraints를 추가 또는 삭제, 변경한다.
Set if-property	If-property constraints를 추가 또는 삭제, 변경한다.
Set single and error	single, error constraints를 변경한다
Set priority rank	priority를 정한다.
Generate test cases	주어진 명세 정보로부터 Category Partitioning Test Case를 생성한다.
Export test case to excel file	생성된 Category Partitioning Test Case을 엑셀파일로 생성한다
Save contemporary specification file	지금까지 작성한 명세파일을 저장한다.

Activity 1007. Define Business Use Case 중 Draw a Use Case Diagram





Stage Review

Stage Review

Stage 2030 Review

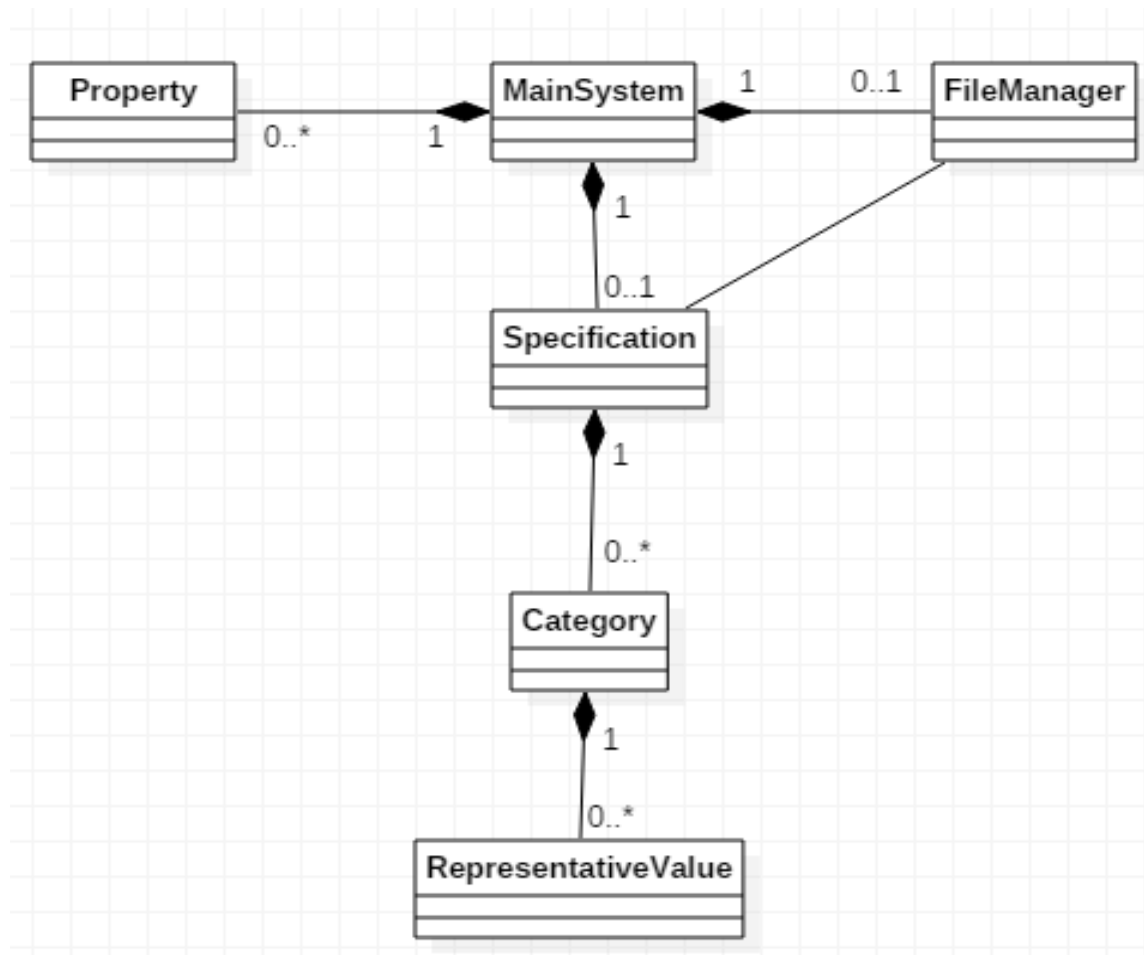
Activity 2031. Define Essential Use Case 중 일부

UseCase	Export test case to excel file
Actor	User
Purpose	Test cases를 엑셀파일로 저장한다.
Overview	Test case generate이 끝난 후, 생성할 엑셀파일의 이름을 받는다. 이후 생성된 test cases를 엑셀파일로 저장한다. 생성된 엑셀파일은 설정한 Priority 순으로 정렬된다.
Type	Primary
CrossReference	FuntionalRequirements:R4
Pre-Requisites	입력된 Specification이 test case를 만들기 위한 충분한 값이 입력되어서, Generate test cases가 실행되어야 한다.
Typical Courses of Events	(A):Actor, (S):System 1. (S) 'Generate test cases'가 실행된다. 2. (S) 생성할 엑셀파일의 이름을 받는 창이 뜬다. 3. (A) 생성할 엑셀파일의 이름을 입력한다. 4. (S) 생성된 test cases를 Priority 순으로 정렬한다. 5. (S) 입력 받은 엑셀파일의 이름으로 엑셀파일을 만든 후, Priority 순으로 정렬된 test cases를 저장한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A





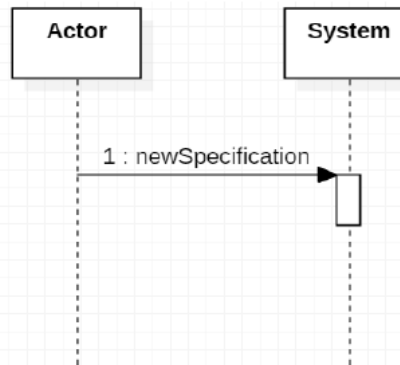
Activity 2033. Define Domain Model



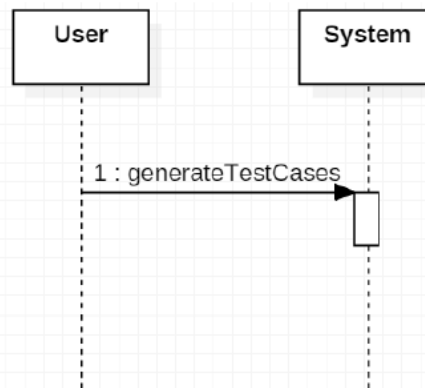


Activity 2035. Define System Sequence Diagrams 중 일부

- Use case : Make new specification file
1. 사용자가 초기화면에 있는 '새로 만들기' 버튼을 누른다.
 2. 시스템이 specification 파일 이름을 입력 받는 단계를 보여준다.
 3. 사용자가 새 파일 이름 항목란에 이름을 입력한다.
 4. 사용자가 '이름 결정' 버튼을 누른다.
 5. 시스템이 지정된 이름의 specification 파일을 만든다.
 6. 시스템이 생성된 specification 파일을 recent file list에 갱신한다.
 7. 시스템이 Set category page를 보여준다.



- Use case : Generate test cases
1. 사용자가 'Test case 생성 후 엑셀로 저장하기' 버튼을 누른다.
 2. 시스템이 입력된 specification을 통해 test case를 만든다.
 3. 시스템이 총 개수를 새로운 화면을 통해 사용자에게 알려준다.





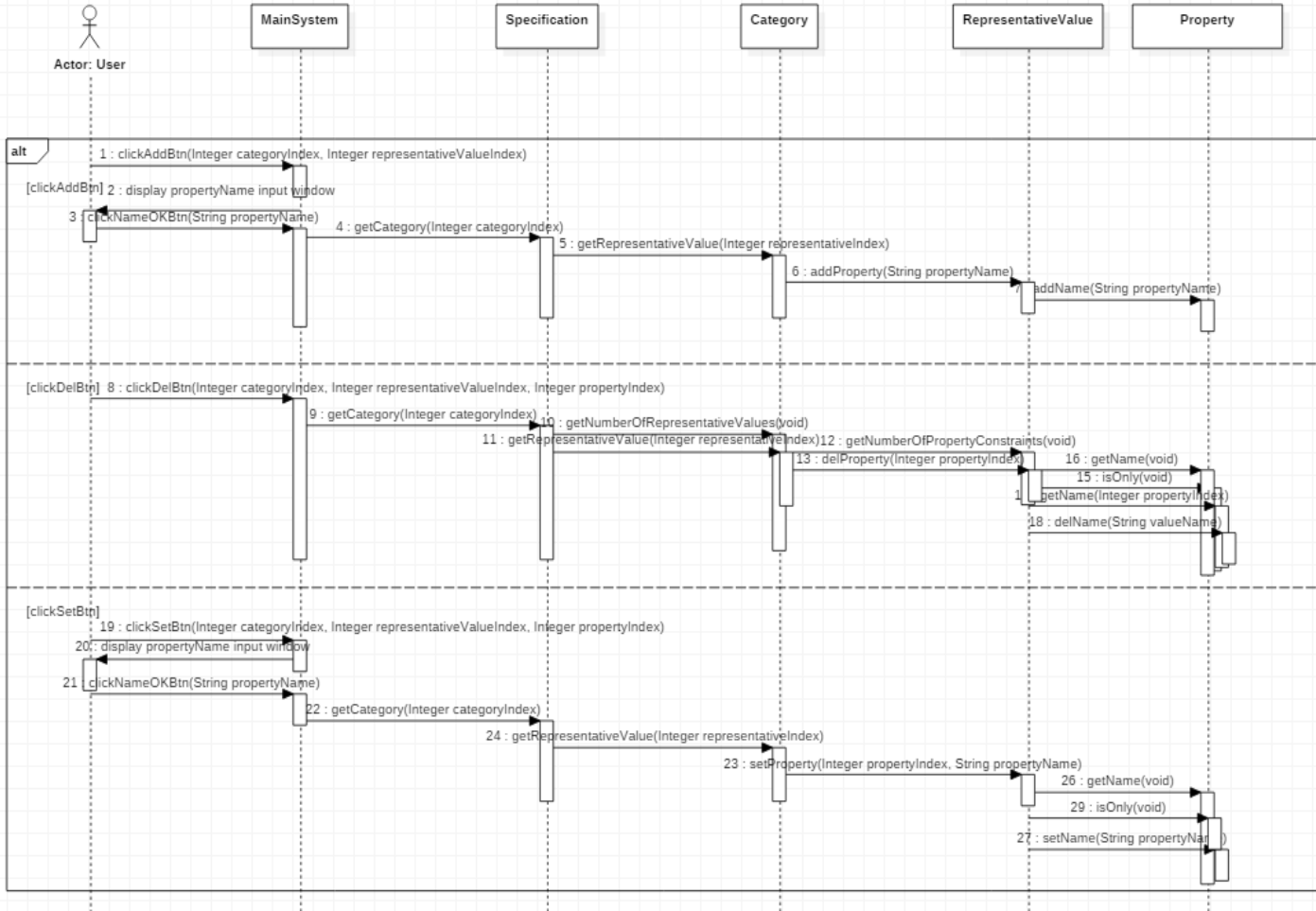
Stage Review

Stage Review

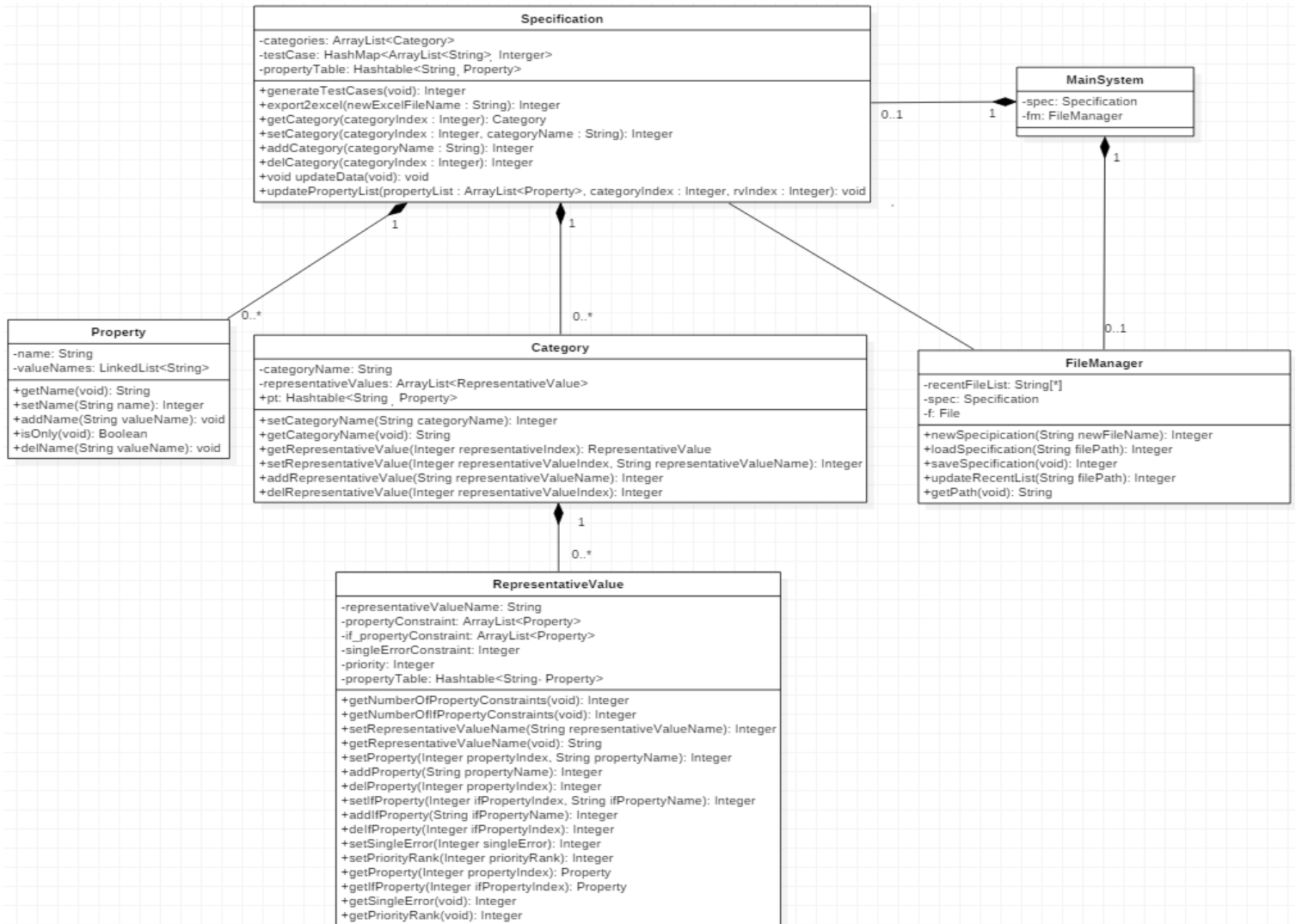
Stage 2040 Review

Activity 2044. Define Interaction Diagrams 중 일부

[setProperty]



Activity 2045. Define Design Class Diagrams





Stage Review

Stage 2050 & 2060 Review

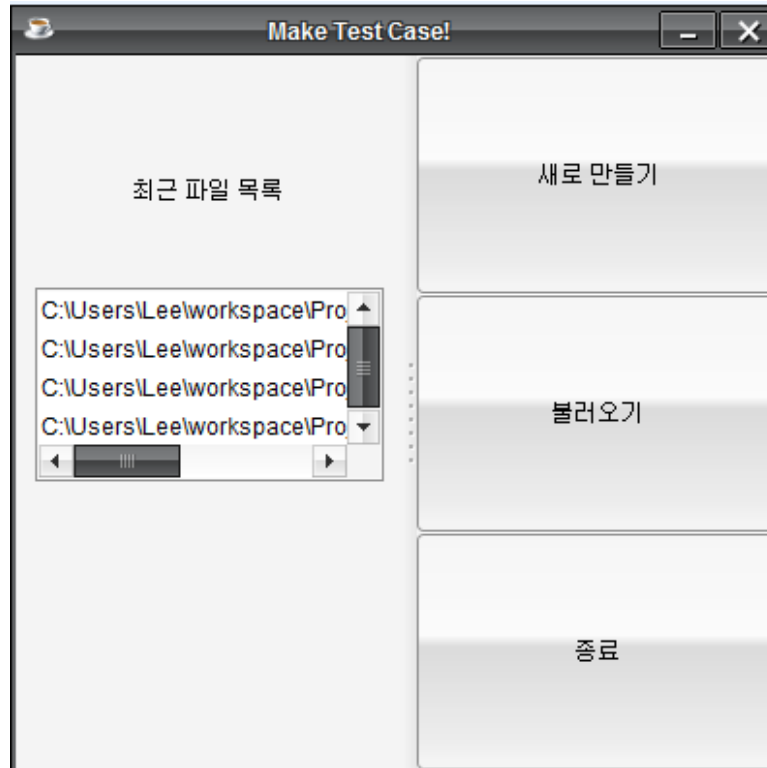
Stage 2050. Implementation

```

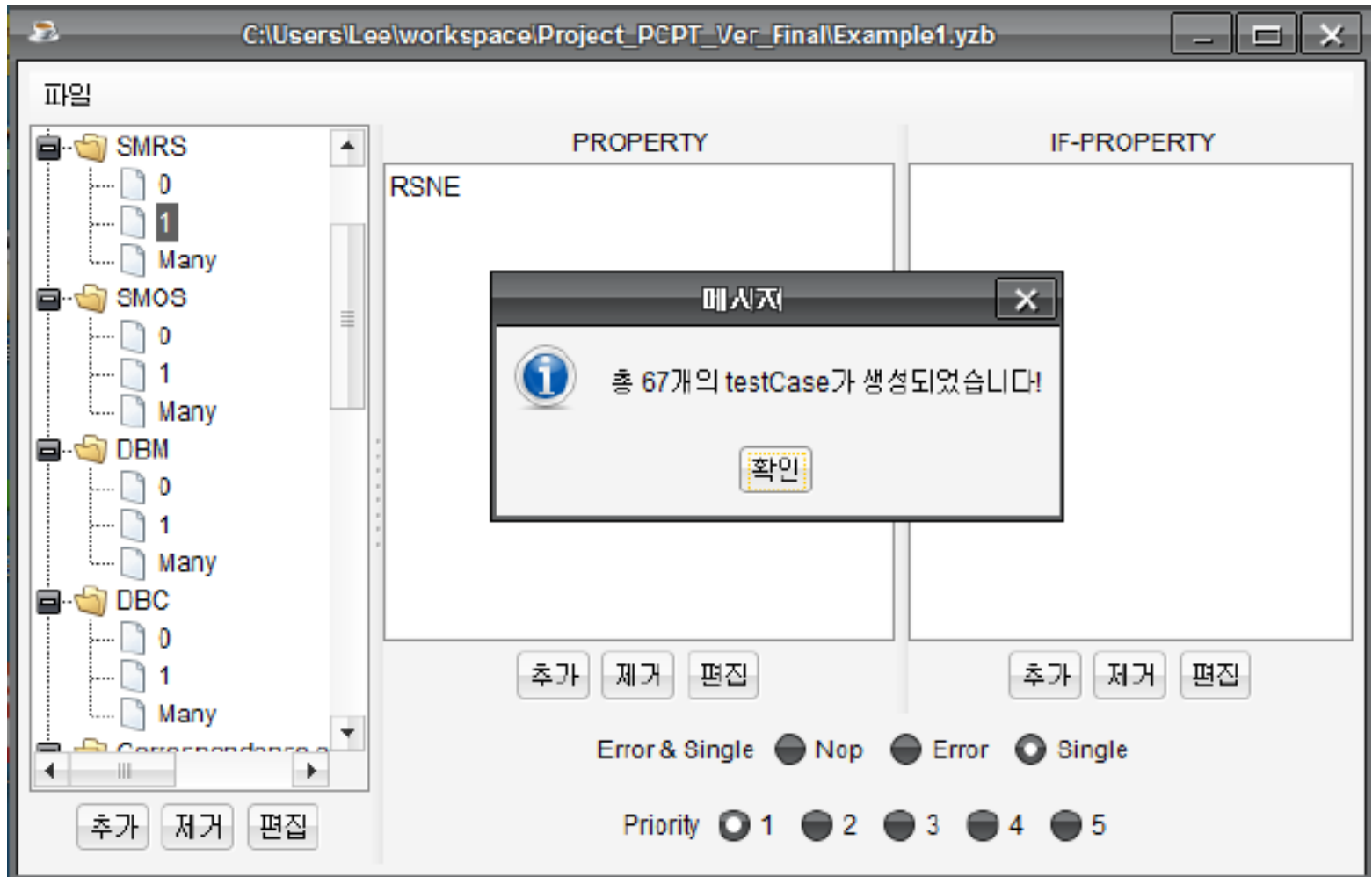
MainSystem.java Specification.java
1 package System;
2
3 import java.util.*;
13
14 public class Specification {
15     private ArrayList<Category> categories;
16     private HashMap<ArrayList<String>, Integer> testCase;
17     private Hashtable<String, Property> propertyTable = new Hashtable<String, Property>();
18
19     // Constructor
20 public Specification() {
21     categories = new ArrayList<Category>();
22 }
23
24 // number of categories
25 public int getNumberOfCategories() {
26     return categories.size();
27 }
28
29 // fill TestCase
30
31 public int generateTestCase() {
32     boolean isExist=false;
33     for(int i=0;i<getNumberOfCategories();i++){
34         if(getCategory(i).getNumberOfRepresentativeValues()!=0){
35             isExist=true;
36             break;
37         }
38     }
39     if(this.getNumberOfCategories()!=0 && isExist){
40         testCase = new HashMap<ArrayList<String>, Integer>();
41         ArrayList<Property> propertyList = new ArrayList<Property>();
42         ArrayList<RepresentativeValue> testCaseList = new ArrayList<RepresentativeValue>();
43         f(0, propertyList, testCaseList, testCase, 0);

```

Stage 2050. Implementation



Stage 2050. Implementation



Stage 2050. Implementation

Example1 (읽기 전용) [호환 모드] - Excel

Test Case No	Model num	SMRS	SMOS	DBM	DBC	Correspon#	of require	Required c#	of option	Optional c#	Priority	Score
1			0								1601	
2	Valid	Many	Many	Many	Many	Complete	0				1601	
3			1								1601	
4					0						1601	
5									>= 1 not in		1601	
6					1						1601	
7									Some defaults		1601	
8						Omitted slots					1601	
9	Malformed										1601	
10						Extra slots					1601	
11						Mismatched slots					1601	
12		0									1601	
13		1									1601	
14					0						1601	
15					1						1601	
16									>= 1 not in database		1601	
17	Not in database										1601	
18									Some defts		1601	
19	Valid	Many	Many	Many	Many	Complete	< number required slots				1601	
20	Valid	Many	Many	Many	Many	Complete	(= number (>= 1 incon = #SMOS	All valid			20	
21	Valid	Many	Many	Many	Many	Complete	(= number (>= 1 incon = #SMOS	>= 1 incon			20	
22	Valid	Many	Many	Many	Many	Complete	(= number (>= 1 incon < #SMOS	>= 1 incon			20	
23	Valid	Many	Many	Many	Many	Complete	(= number (>= 1 incon	0	>= 1 incon		20	
24	Valid	Many	Many	Many	Many	Complete	(= number (>= 1 incon	0	>= 1 incon		20	
25	Valid	Many	Many	Many	Many	Complete	(= number (>= 1 incon < #SMOS	All valid			20	
26	Valid	Many	Many	Many	Many	Complete	(= number (>= 1 incon	0	All valid		20	

Activity 2061. Unit Testing

The screenshot displays an IDE with two main windows. The left window shows the source code for `FileManagerTest.java`, and the right window shows the JUnit test runner output.

```
1 package test;
2
3 import static org.junit.Assert.*;
16
17 public class FileManagerTest {
18     private static Specification s;
19     private static FileManager f;
20     private static File file;
21
22     @BeforeClass
23     public static void beforeSetup() {
24         s = new Specification();
25         f = new FileManager(s);
26     }
27
28     @Test //생성 잘 되는지 테스트
29     public void testNewSpecification1() {
30         file = new File("newTest.yzb");
31         if(file.exists()){
32             file.delete();
33         }
34         assertEquals(0, f.newSpecification("newTest"));
35         assertTrue(file.exists());
36     }
37
38     @Test //test1에서 생성한 파일명과 같으니까 물어보는 장이 뜰 것이다. 예를 누르면 0, 아니오를 누르면 1이 반환되
39     public void testNewSpecification2() {
40         assertEquals(0, f.newSpecification(file.getAbsolutePath()).substr
41         assertEquals(1, f.newSpecification(file.getAbsolutePath()).substr
42     }
43
44     @Test //존재하는 파일은 잘 불러와야된다. 그러면 0을 리턴.
45     public void testLoadSpecification1() {
```

The right window shows the JUnit runner output:

Finished after 5.61 seconds

Runs: 9/9 Errors: 0 Failures: 0

test.FileManagerTest [Runner: JUnit 4] (5.523 s)

- testUpdateRecentList (0.000 s)
- testSaveSpecificationString1 (0.005 s)
- testSaveSpecificationString2 (1.760 s)
- testGetPath (0.000 s)
- testNewSpecification1 (0.013 s)
- testNewSpecification2 (2.947 s)
- testSaveSpecification (0.003 s)
- testLoadSpecification1 (0.009 s)
- testLoadSpecification2 (0.786 s)



Team 1

Feedback

Team 1

Feedback



T1's Analysis

[Sonar Qube]

```
filePath = new String(filePath+".yzb");
```

GUI.Explorer.doit(int, FileManager, Specification) invokes inefficient new String(String) constructor ...

Bug Major Open 할당되지 않음 30min effort

[Metrics]

Metric	Total	Mean	Std. Dev.	Maxim...	Resource causing Maximum	Method
> McCabe Cyclomatic Complexity (avg/max per method)		2.516	3.185	20	/T4_source/src/System/FileManager.java	loadSpecification
> Number of Parameters (avg/max per method)		0.731	0.985	5	/T4_source/src/System/Specification.java	f
> Nested Block Depth (avg/max per method)		1.946	1.498	8	/T4_source/src/System/FileManager.java	loadSpecification
> Afferent Couplino (avo/max per packageFrament)		3	3.559	8	/T4 source/src/Svstem	



T1's Analysis

[T4의 코드 중 일부]

```
for(int i=0;i<sub.length;i++){
    if(i==0){
        categories = sub[i].split("/");
        for(int j=0;j<categories.length;j++){
            spec.addCategory(categories[j]);
        }
    }
    else{
        values = sub[i].split("/");

        for(int j=0;j<values.length;j++){
            temp = values[j].split("\\\\");
            spec.getCategory(i-1).addRepresentativeValue(temp[0]);
            if(temp[1].length()!=0){
                properties = temp[1].split(",");
                for(int k=0;k<properties.length;k++){
                    spec.getCategory(i-1).getRepresentativeValue(j).addProperty(properties[k]);
                }
            }
            ifPropertybuff.append(i-1+" "+j+" "+temp[2]+" /");
            spec.getCategory(i-1).getRepresentativeValue(j).setSingleError(Integer.parseInt(temp[3]));
            spec.getCategory(i-1).getRepresentativeValue(j).setPriorityRank(Integer.parseInt(temp[4]));
        }
    }
}

if(ifPropertybuff.toString().length()!=0){
    ifPropertysemibuff = ifPropertybuff.toString().split("/");
    for(int i=0;i<ifPropertysemibuff.length;i++){
        ifPropertyqbuff = ifPropertysemibuff[i].split(":");
        int cNumber = Integer.parseInt(ifPropertyqbuff[0].substring(0, 1));
        int rNumber = Integer.parseInt(ifPropertyqbuff[0].substring(1));
        if(ifPropertyqbuff.length!=1){
            ifProperties = ifPropertyqbuff[1].split(",");
            for(int j=0;j<ifProperties.length;j++){
                spec.getCategory(cNumber).getRepresentativeValue(rNumber).addIfProperty(ifProperties[j]);
            }
        }
    }
}
```



Team2

Feedback

Team 2

Feedback



T2's Analysis

[JDepend]

항목 I에 의하면 System 패키지는 안정성 0.66으로 **변화에 안정적이지 않다.**

항목 A와 Ce에 의하면 GUI 패키지는 구체적이지만 이 패키지의 클래스들은 **1개의 패키지에만 종속하고 있기 때문에 이상적이지 않다.**

항목 AC에 의하면 **인터페이스나 추상클래스가 존재하지 않기 때문에** 확장이 어려워 보인다.

항목 Ce에 의하면 System 패키지의 클래스들은 4개의 **패키지에 의존**하고있다.



T2's Analysis

[FindBug]

FindBugs를 사용한 정적 분석 결과,

총 61개의 Bug를 발견하였으며,

이 중 Low 등급은 27개, Normal 등급은 23개, High 등급은 11개이다.

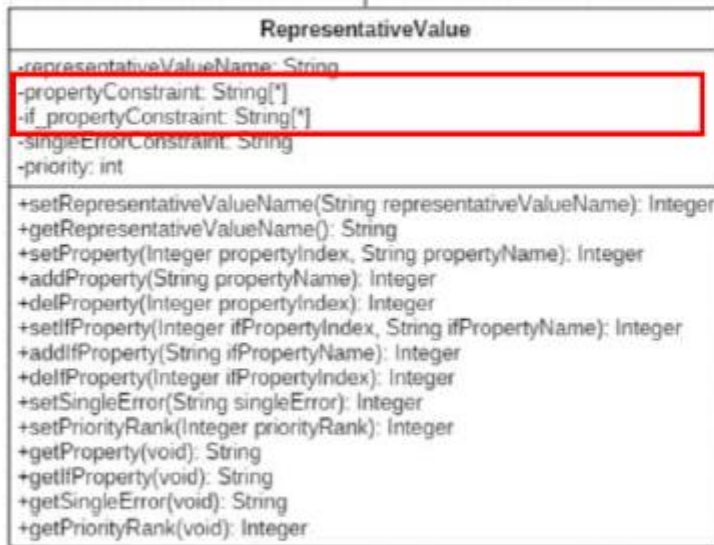
[CheckStyle]

T4의 프로젝트는 총 35개 종류의 CheckStyle을 위반하였으며,

위배된 코드의 총 개수는 6248개이다

T2's Analysis

[ETC]



Stage2040_T4.pdf
ClassDiagram



Eclipse Plugin ObjectAid
2017.5.25





T4

Epilogue

T4

Epilogue



T4

Demonstration

T4

Demonstration

Q & A





Thank you!