



# T1 FINAL PRESENTATION

17.06.08 Final Presentation

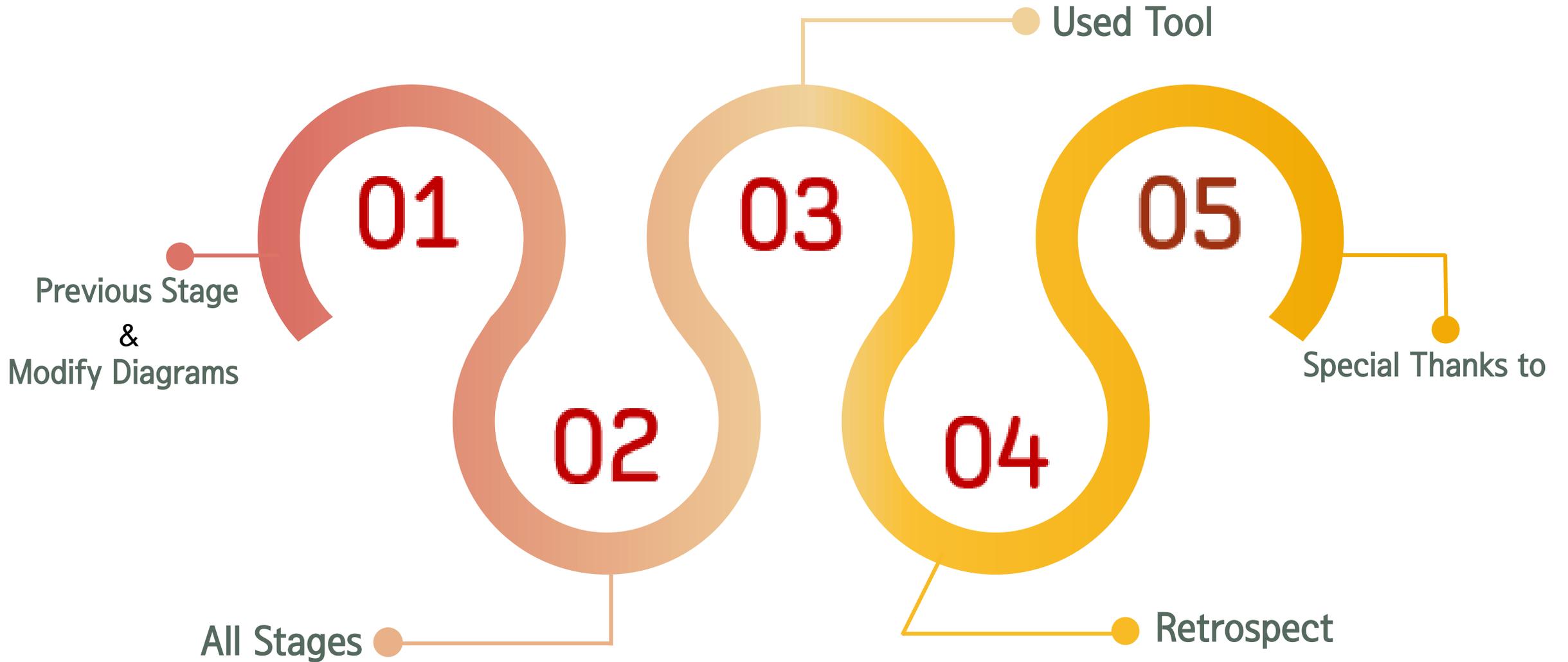
201111334 김강욱 / 201211339 김진욱(Leader)

201312243 김동권 / 201510411 이소영



# INDEX

---



# Implementation

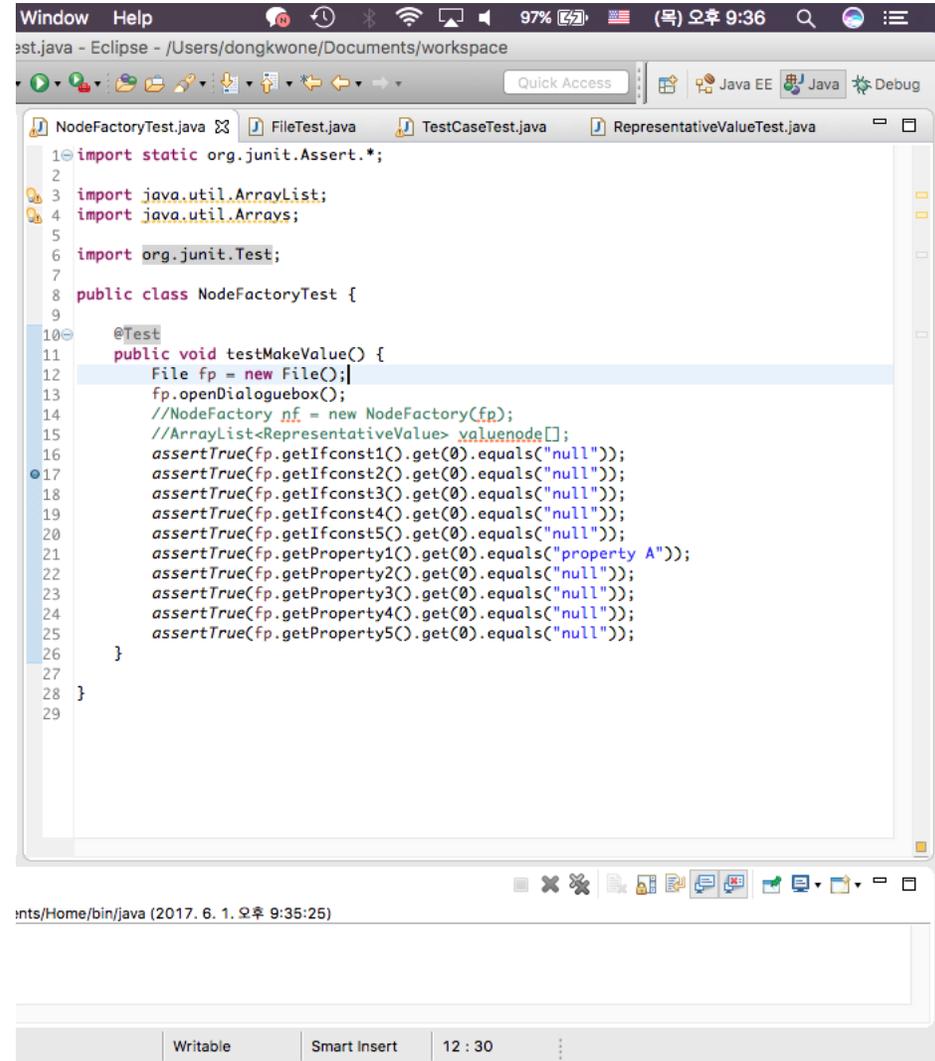
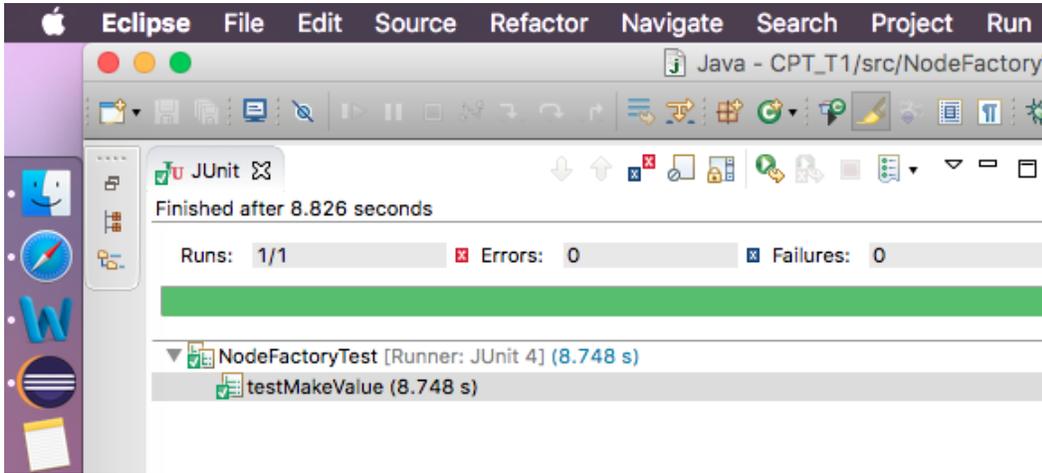
## Previous Stage\_GUI

```
*GUI_CPT.java
469
470 public void generatePrintButton() {
471     getGenerateBtn().addActionListener(new ActionListener() {
472
473         @Override
474         public void actionPerformed(ActionEvent generateEvent) {
475             System.out.println("event : " + generateEvent);
476             // TODO Auto-generated method stub
477             //ManagementSystem.reqMakeCase();
478             System.out.println("generate Btn Pressed");
479             if (generateflag == 0) {
480                 generateTable();
481                 addTestCaselist(manager.reqMakeCase());
482                 generateflag = 2;
483                 dm.fireTableDataChanged();
484             } else if (generateflag == 2) {
485                 System.out.println("그려져라 압!");
486                 for (int i = 0; i < table.getRowCount(); i++)
487                 {
488                     System.out.println("remove table");
489                     //table.removeAll();
490                     ((DefaultTableModel)table.getModel()).removeRow(i);
491                     //dm.removeRow(i);
492                 }
493                 addData(manager.reqMakeCase());
494             }
495             //table.repaint();

```

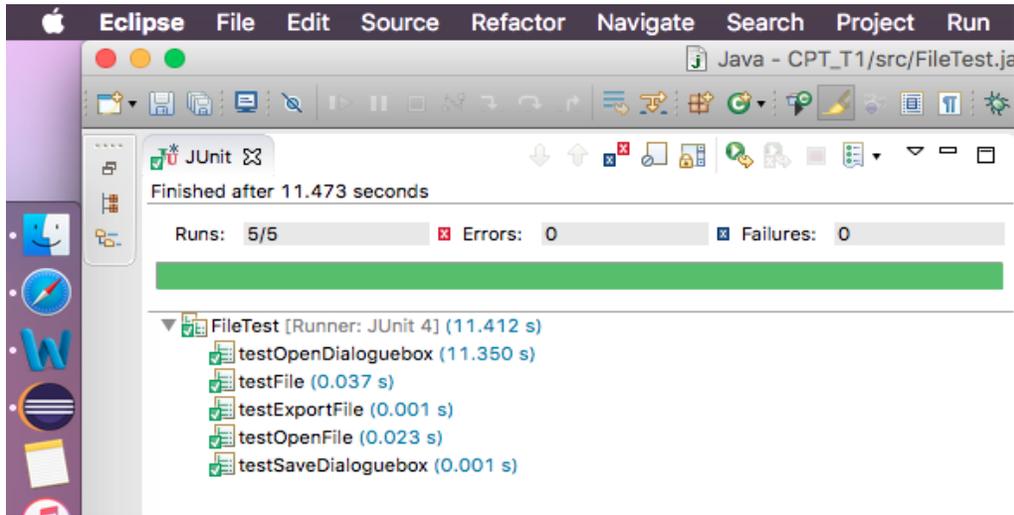
# Unit test

Previous Stage\_Make Value Test



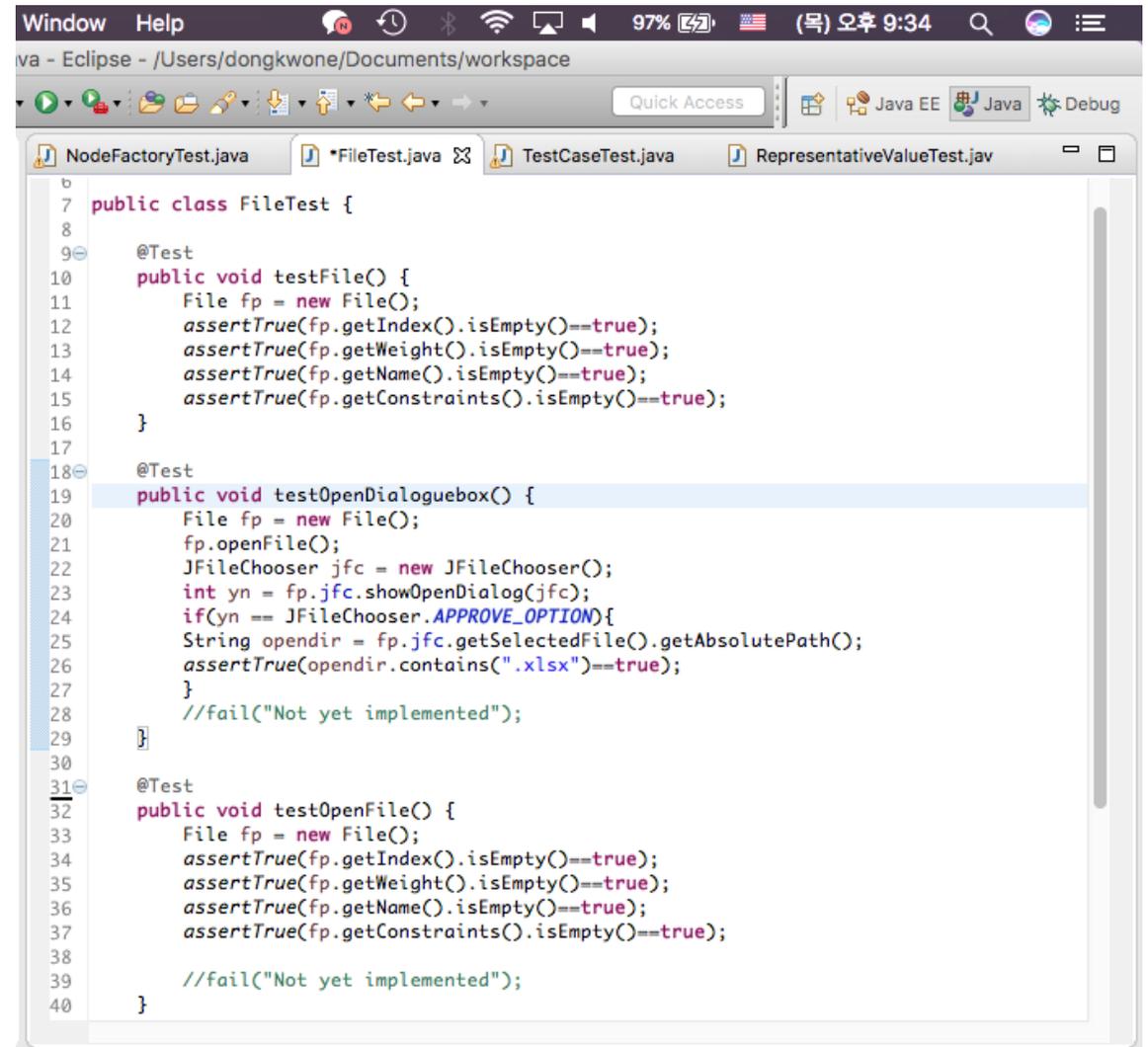
# Unit test

Previous Stage\_OpenDialogue Box test



The screenshot shows the Eclipse IDE interface. The top menu bar includes File, Edit, Source, Refactor, Navigate, Search, Project, and Run. The main editor area displays the file 'Java - CPT\_T1/src/FileTest.java'. Below the editor, the JUnit test runner window is visible, showing 'Finished after 11.473 seconds'. The summary indicates 'Runs: 5/5', 'Errors: 0', and 'Failures: 0'. A green progress bar is shown. The test results list includes:

- FileTest [Runner: JUnit 4] (11.412 s)
  - testOpenDialoguebox (11.350 s)
  - testFile (0.037 s)
  - testExportFile (0.001 s)
  - testOpenFile (0.023 s)
  - testSaveDialoguebox (0.001 s)

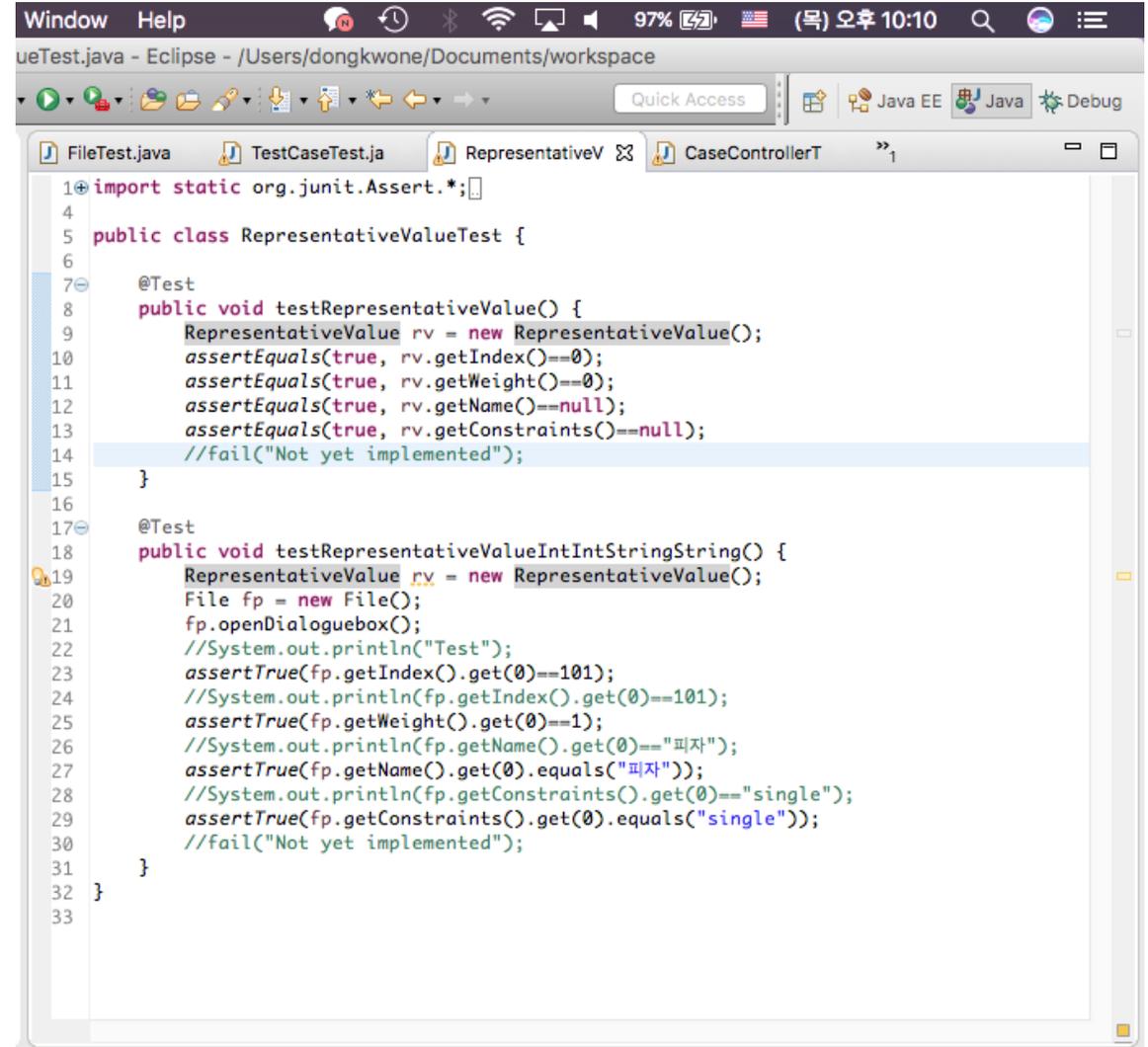
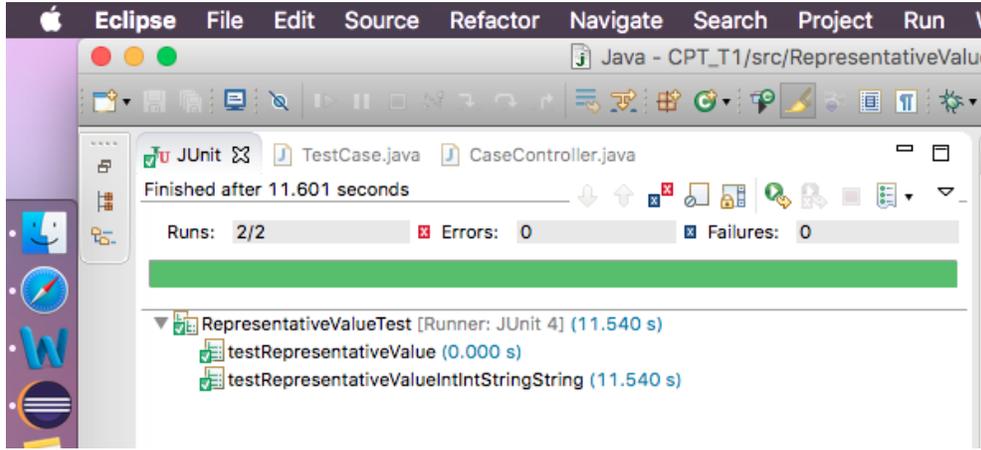


The screenshot shows the Eclipse IDE interface with the source code of 'FileTest.java' open. The code is as follows:

```
6
7 public class FileTest {
8
9     @Test
10    public void testFile() {
11        File fp = new File();
12        assertTrue(fp.getIndex().isEmpty()==true);
13        assertTrue(fp.getWeight().isEmpty()==true);
14        assertTrue(fp.getName().isEmpty()==true);
15        assertTrue(fp.getConstraints().isEmpty()==true);
16    }
17
18    @Test
19    public void testOpenDialoguebox() {
20        File fp = new File();
21        fp.openFile();
22        JFileChooser jfc = new JFileChooser();
23        int yn = fp.jfc.showOpenDialog(jfc);
24        if(yn == JFileChooser.APPROVE_OPTION){
25            String opendir = fp.jfc.getSelectedFile().getAbsolutePath();
26            assertTrue(opendir.contains(".xlsx")==true);
27        }
28        //fail("Not yet implemented");
29    }
30
31    @Test
32    public void testOpenFile() {
33        File fp = new File();
34        assertTrue(fp.getIndex().isEmpty()==true);
35        assertTrue(fp.getWeight().isEmpty()==true);
36        assertTrue(fp.getName().isEmpty()==true);
37        assertTrue(fp.getConstraints().isEmpty()==true);
38
39        //fail("Not yet implemented");
40    }
}
```

# Unit test

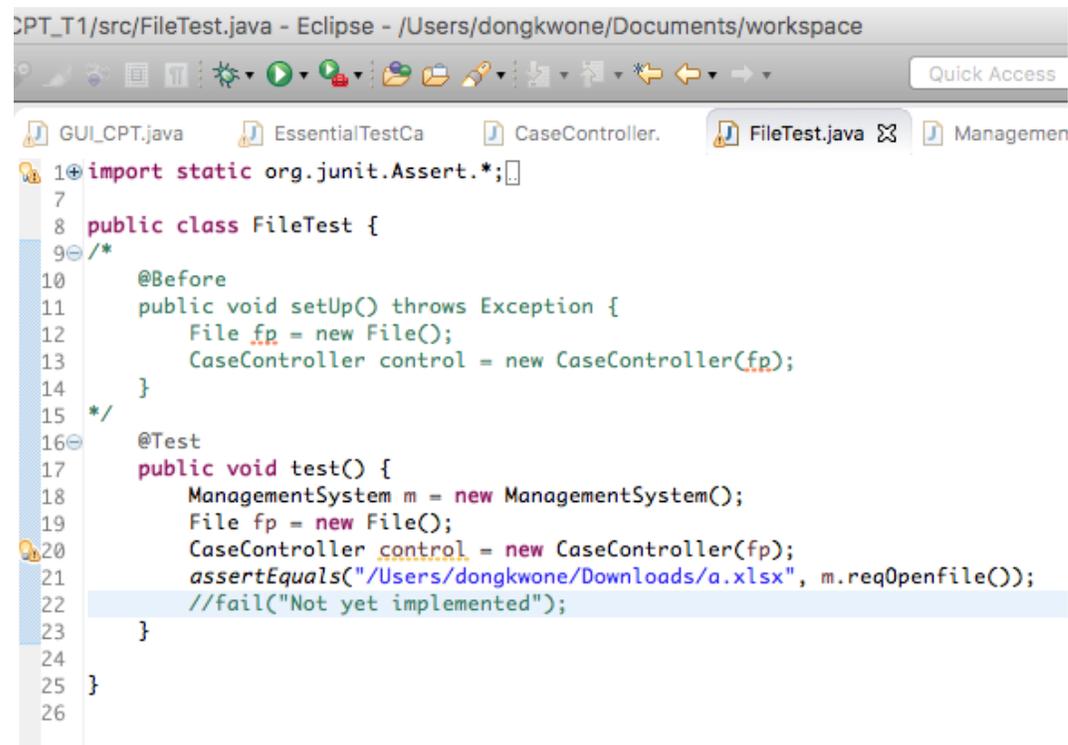
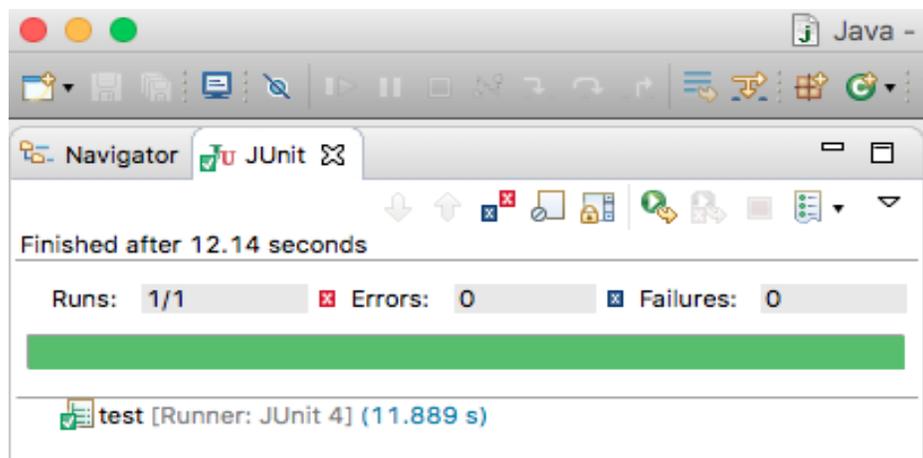
## Previous Stage\_Representative Value Test





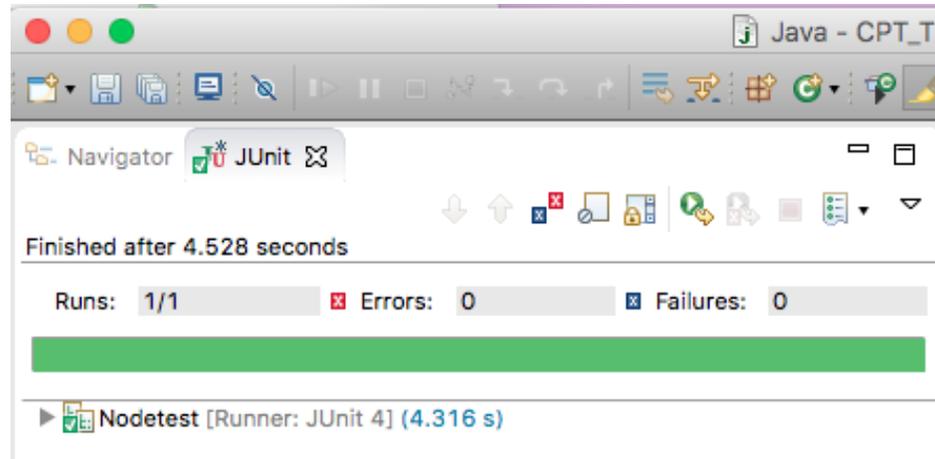
# Unit test

Previous Stage\_Open File Test



# Unit test

Previous Stage\_Value Node Test



```
NodeFactory.java File.java EssentialTestCa ManagementSyste
5 import org.junit.Before;
6 import org.junit.Test;
7
8 import junit.framework.Assert;
9
10 public class Nodetest {
11
12     @Before
13     public void setUp() throws Exception {
14
15
16
17     }
18
19     @Test
20     public void test() {
21         //makevalue node가 null이 아닌지 test
22         ArrayList<RepresentativeValue> valuenode[];
23         int categorynum;
24         int valuenum;
25         File fp = new File();
26         NodeFactory n = new NodeFactory(fp);
27         valuenode=new ArrayList[fp.getCategoryMax()+1];
28         for(int i=0;i<fp.getCategoryMax()+1;i++)
29         {
30             valuenode[i]=new ArrayList<RepresentativeValue>();
31         }
32         n.makeValue(fp);
33         Assert.assertNotNull(valuenode);
34         //fail("Not yet implemented");
35     }
}
```



# Reflect

Static Analysis\_반영



```
41     private int rows;
42     private int cells;
43     //private int file;
44     private XSSFRow row;
45     private XSSFCell cell;
```

```
1 import java.util.ArrayList;
2
3 public class EssentialTestCase {
4     private int existence;
5     private int causevalue;
6     private String[] split;
7     private ArrayList<RepresentativeValue> essentialtestcase = new ArrayList<RepresentativeValue>();
8
9     private ArrayList<Integer> ind = new ArrayList<Integer>();
10
11     //private ArrayList<String> eTestCase = new ArrayList<String>();
12
13     private ArrayList<String> property = new ArrayList<String>();
14     private int ifcheck=0;
15     private int ifcheck2=0;
16
17     private String index;
18     private float weight;
19
20     private String cause="존재";
21 }
```

## Unread Field 반영



# Reflect

Static Analysis\_반영



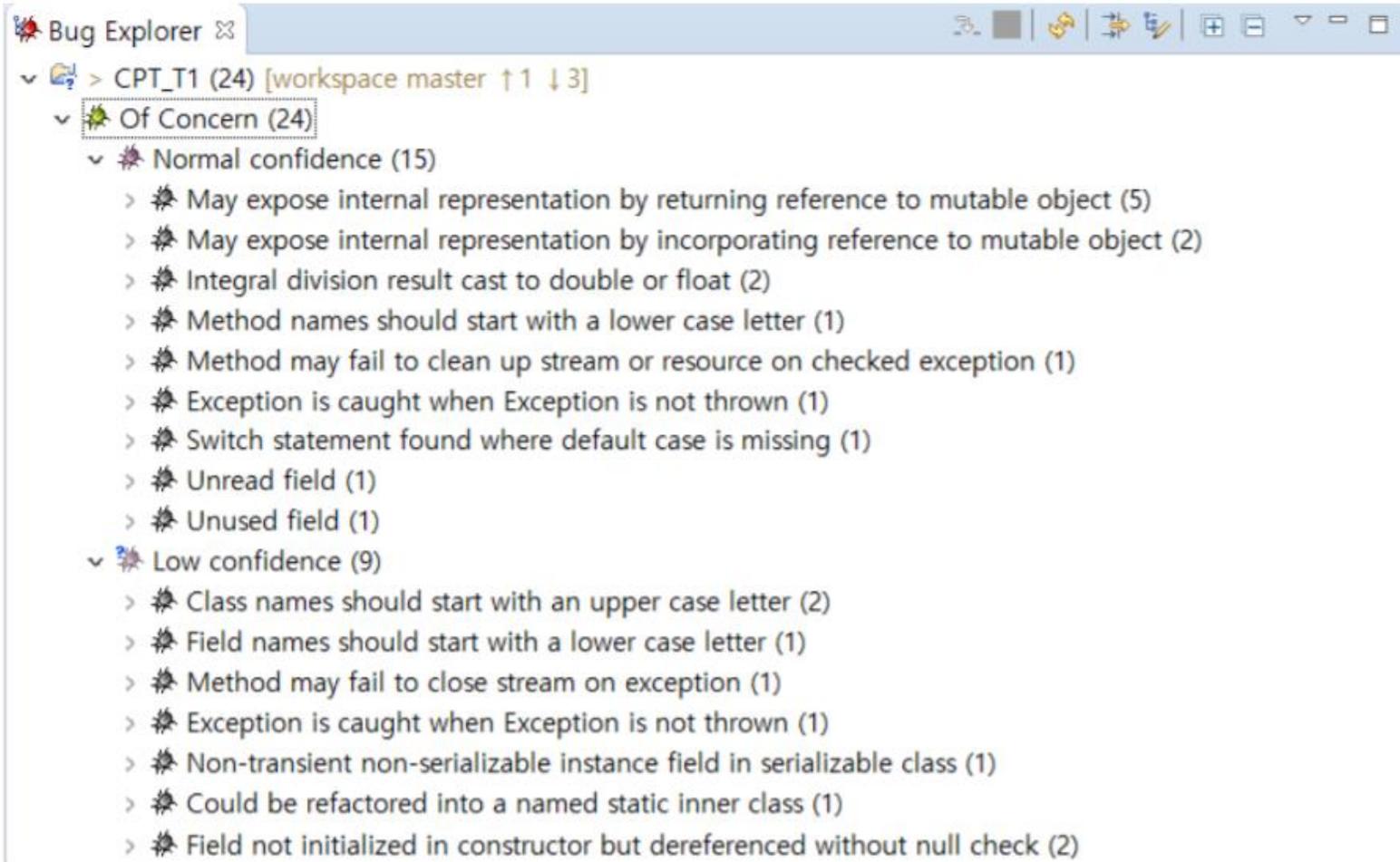
```
256 public float getWeight()  
257 {  
258     int sum=0;  
259     for(int i=0;i<al.size();i++)  
260     {  
261         sum+=al.get(i).getWeight();  
262     }  
263     if(al.size()>0)  
264     {  
265         weight=(float)sum/al.size();  
266     }  
267     else  
268     {  
269         weight=0;  
270     }  
271     return weight;  
272 }
```

## Float Cast 반영



# Reflect

## Static Analysis\_반영



Expose internal  
representation

-> 팀원 회의 결과 반영하지 않기로 함.

Exception 처리

-> e.printStackTrace() 를 쓰는  
방법 등으로 대체. 에러 근원지를 찾아  
단계별로 출력함으로써 구현과정에서  
예외 없이 돌아가도록 구현완료함.



# DEMO VIDEO

CPT\_T1 [www.Bandicam.co.kr](http://www.Bandicam.co.kr) - □ ×

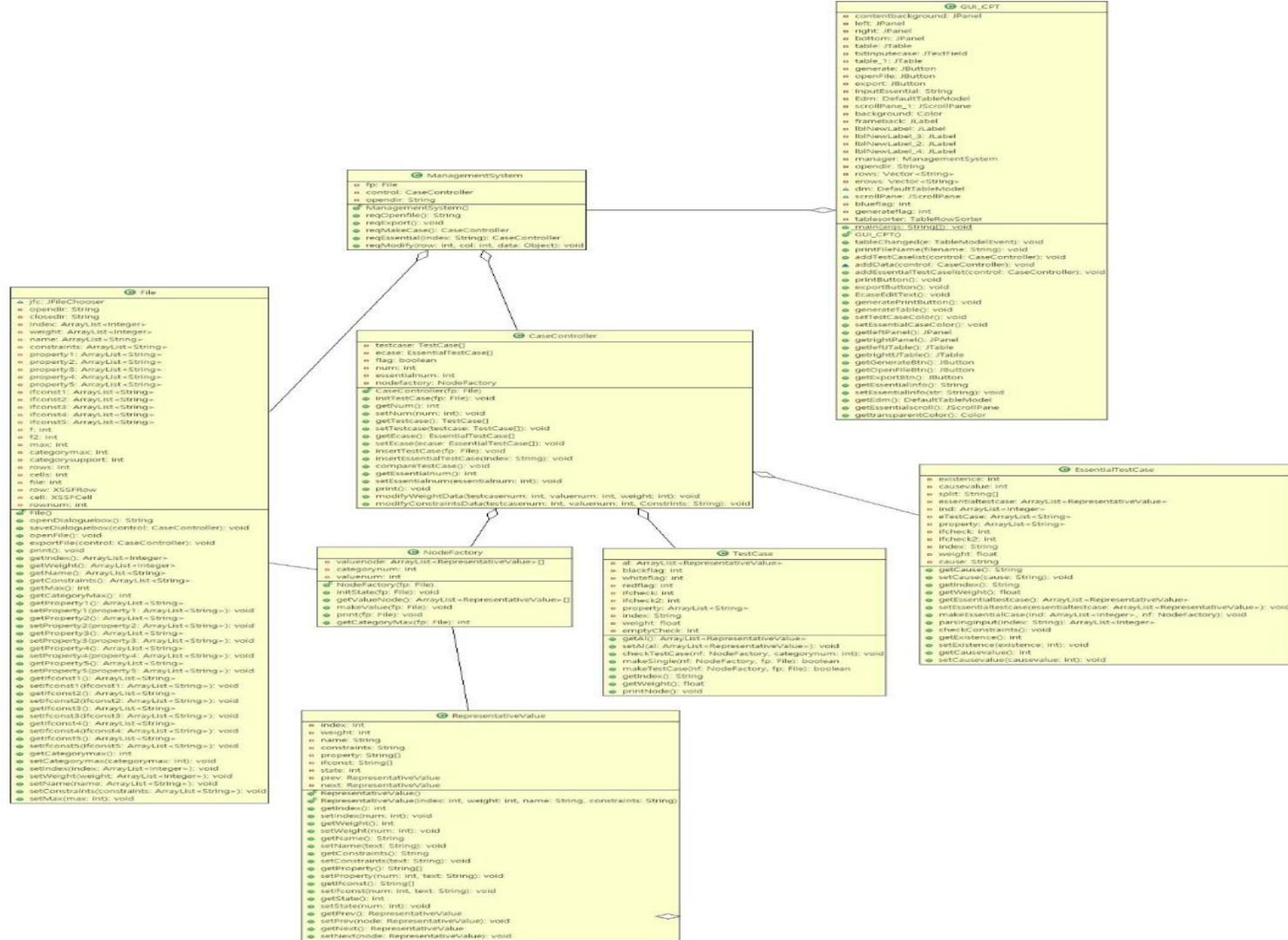
## Essential Test Case

--	--



# Modify Diagrams

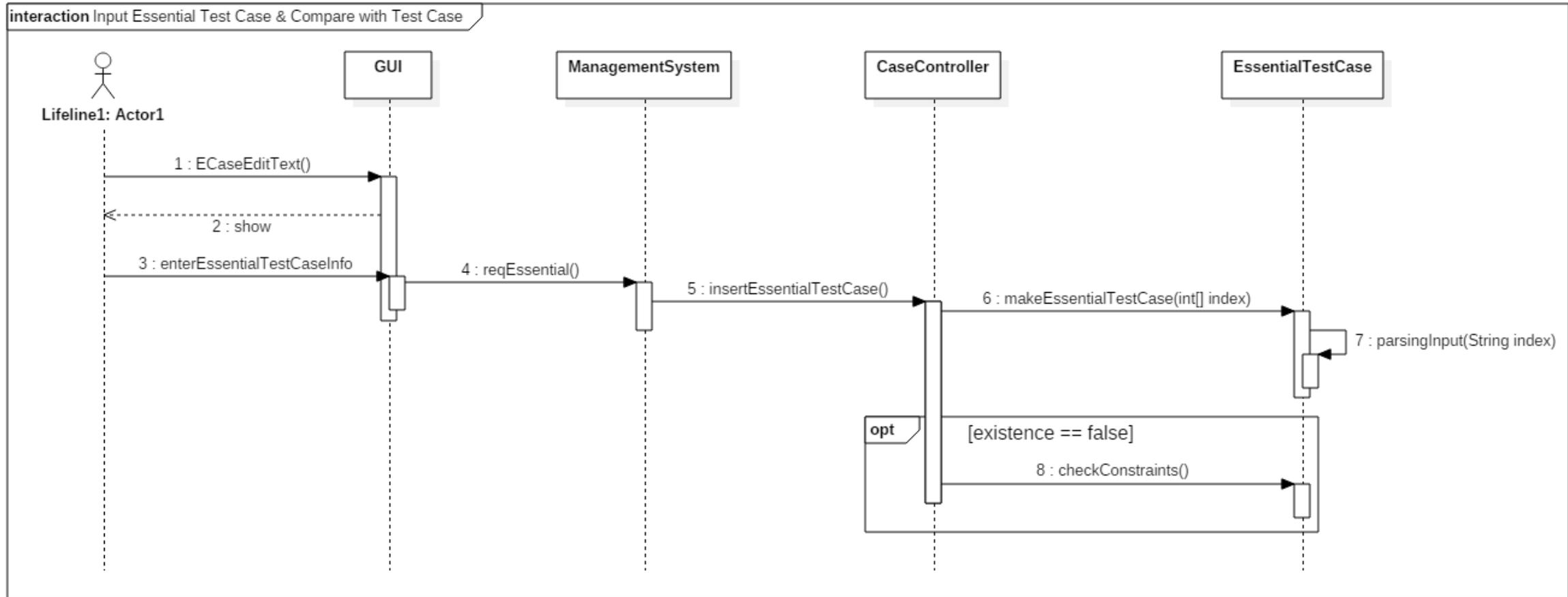
## Class Diagram v5





# Modify Diagrams

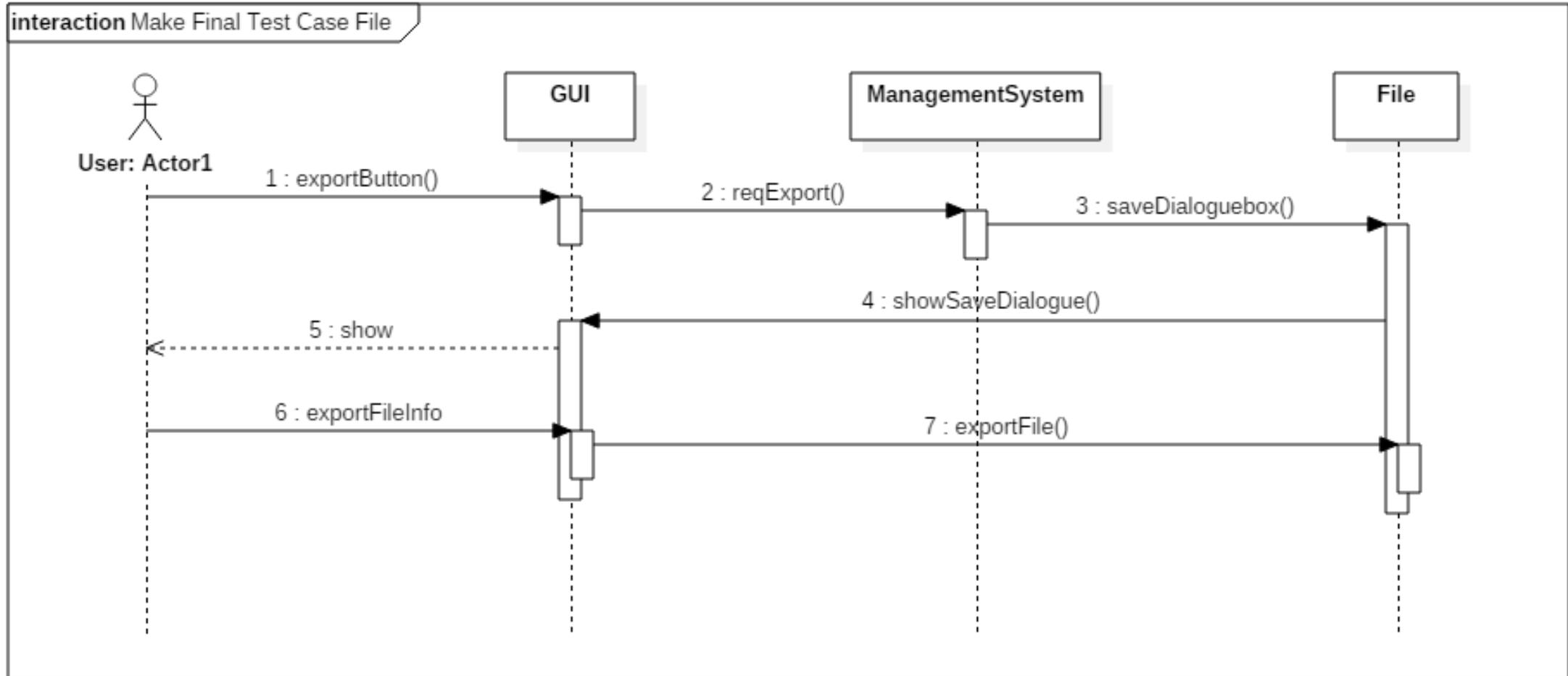
## Sequence Diagram\_1





# Modify Diagrams

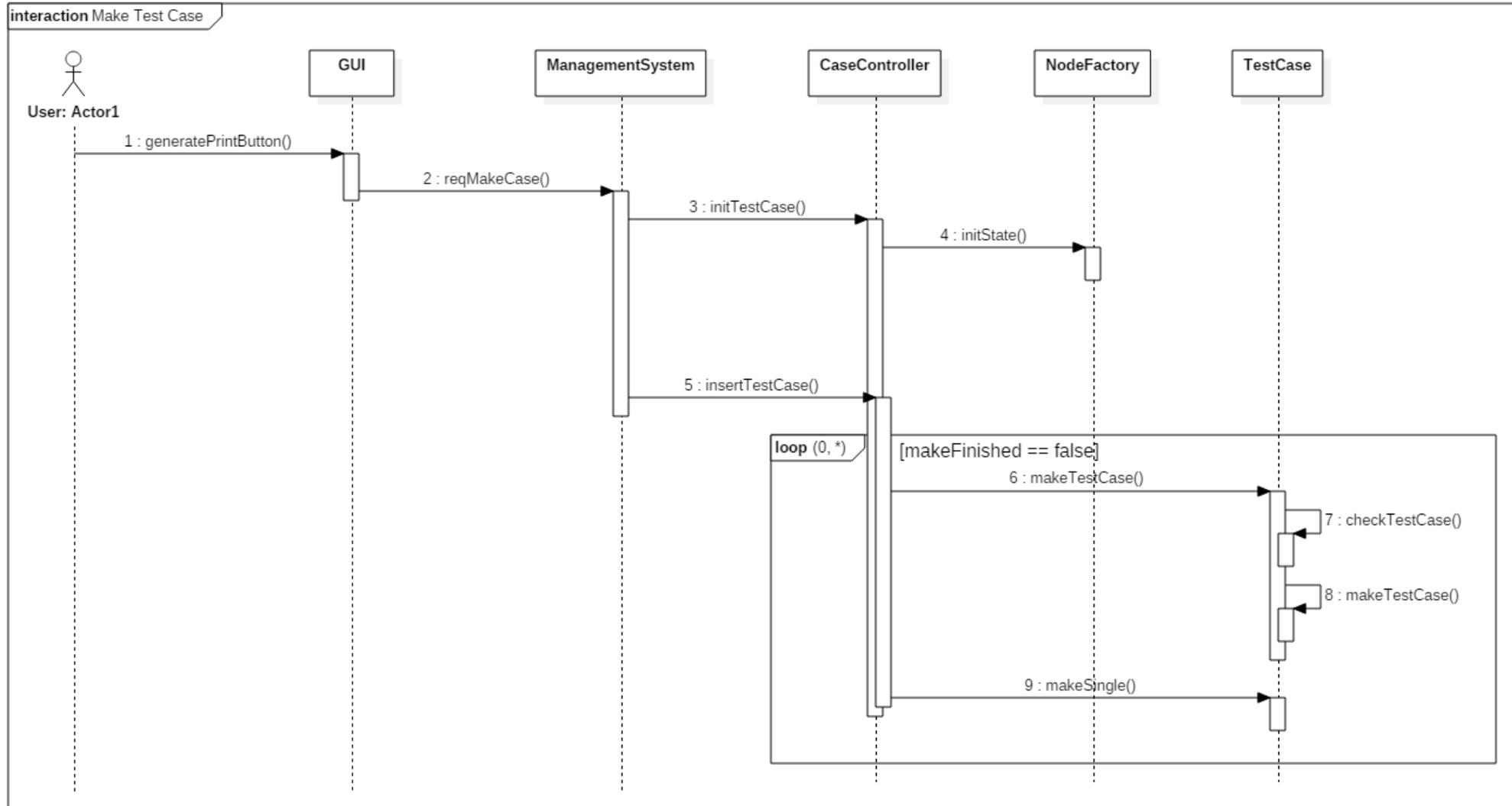
## Sequence Diagram\_2





# Modify Diagrams

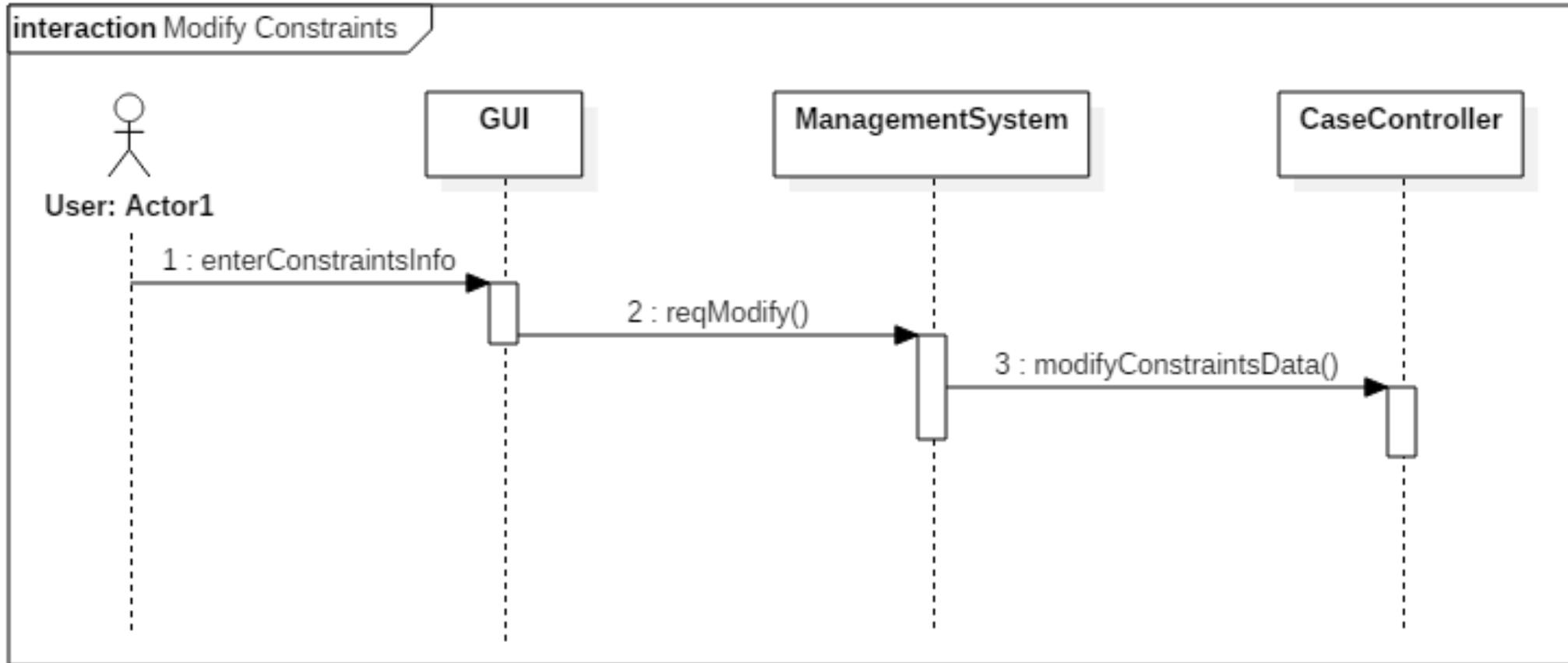
## Sequence Diagram\_3





# Modify Diagrams

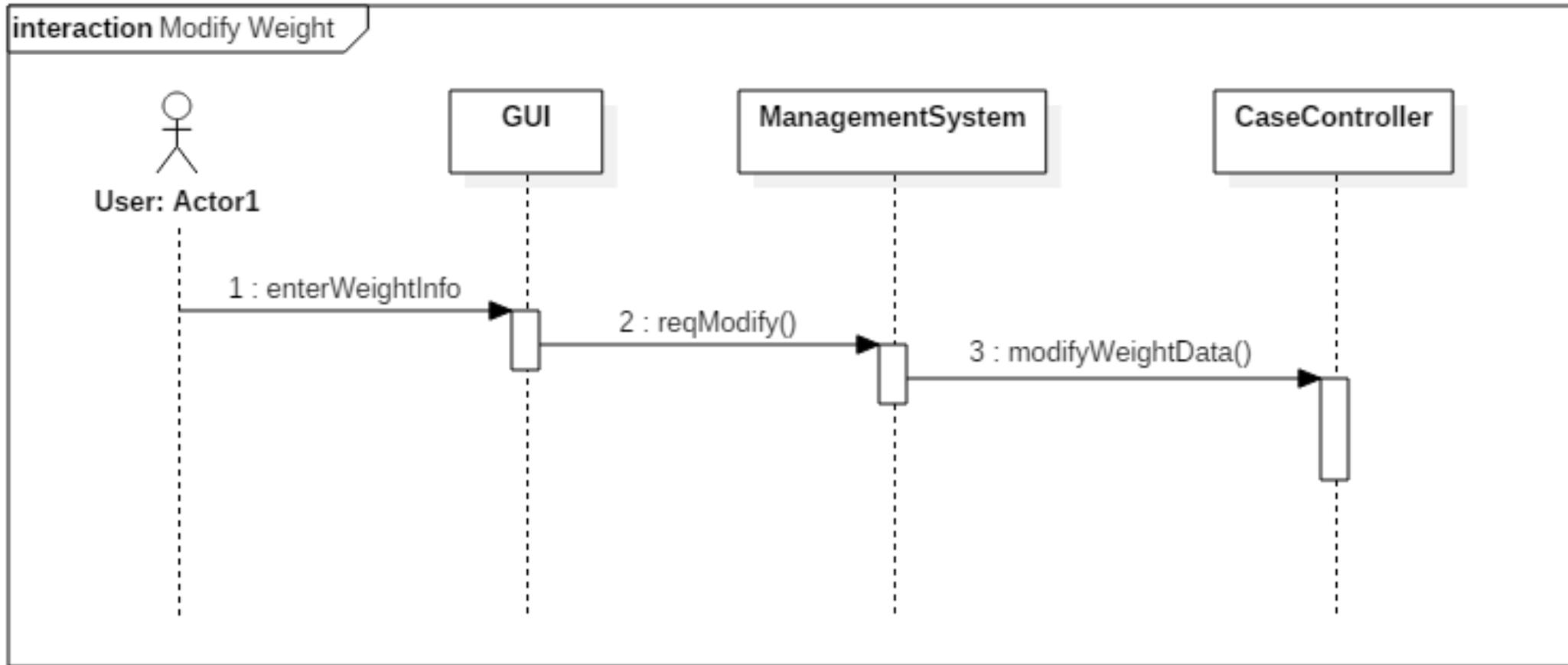
Sequence Diagram\_4





# Modify Diagrams

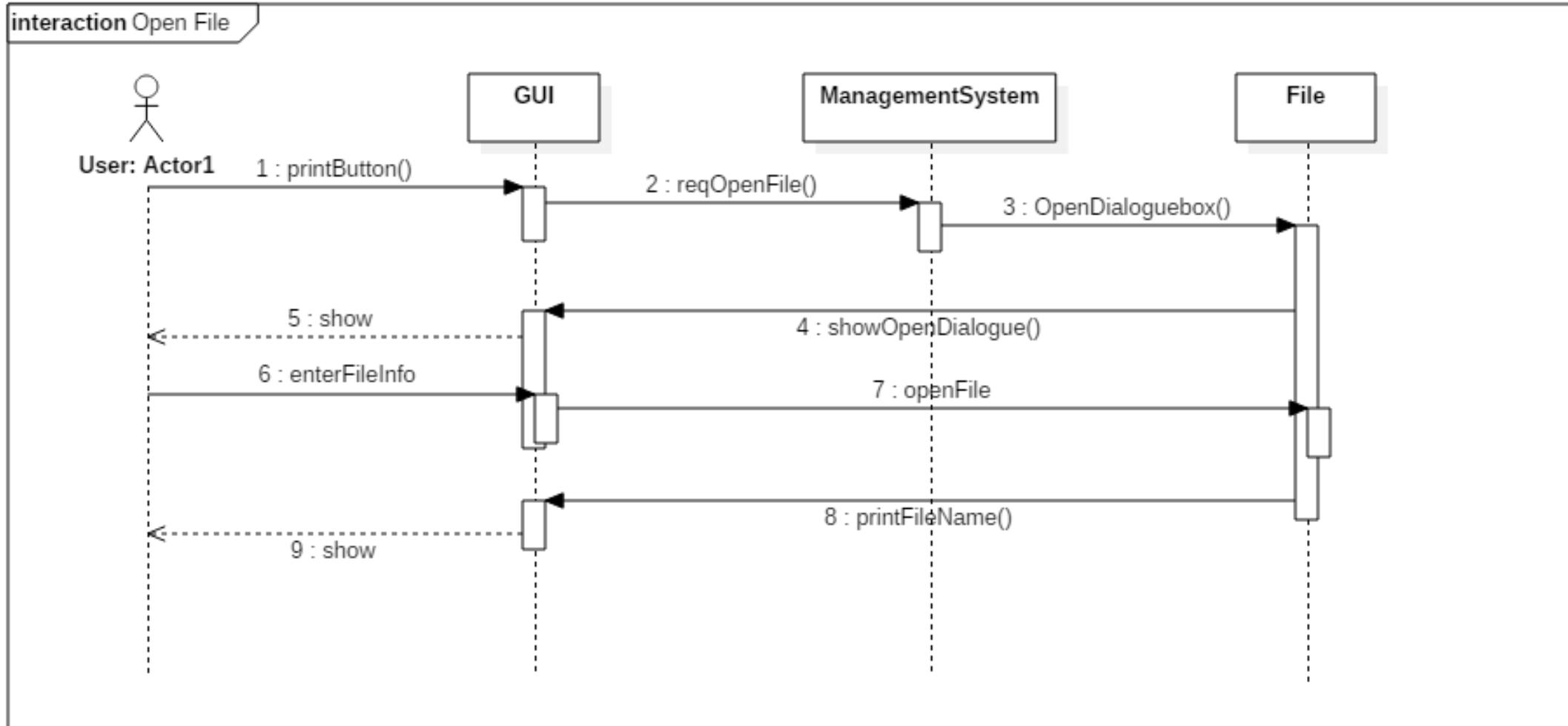
## Sequence Diagram\_5





# Modify Diagrams

## Sequence Diagram\_6



# 2

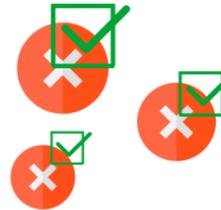
## Stage 1000 Plan & Elaboration



### 2 제품 설명 및 기대효과 제품 설명: 전체



한정된 시간에,



많은 오류 수정, 효율적 test

### 2 제품 설명 및 기대효과 제품 설명: 가중치 부여

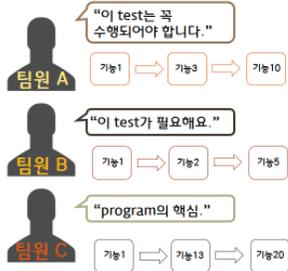


**Category List**  
Value 1 : Weight 1.5  
Value 2 : Weight 0.5  
Value 3 : Weight 1  
Value 4 : Weight 1.3  
⋮



→ 평균 Weight 높은 순서대로 정렬

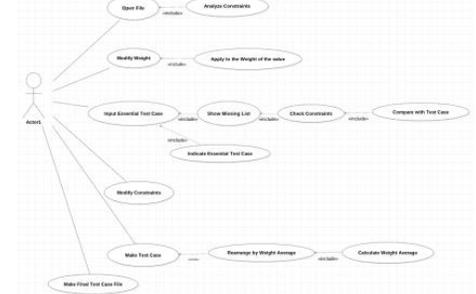
### 2 제품 설명 및 기대효과 제품 설명: 중요한 test case 제안



### 3 진행계획 Risk & Reduction Plan

	Plan	Java skill	OOPT Skill	UML Skill	Exam
Probability	5	4	3	5	10
Significance	5	10	8	9	10
Reduction Plan	Trello, 분담, 상호 압박 넘기	선배 찬스	교수님과 조교님께 도움을 요청	도움요청 + 선배찬스	상호 도움

### 4 시스템 구조 Use Case Diagram



# 2

## Stage 2030

Refine '000', Analyze

### 2nd UseCase Diagram

### 1st Domain Model

#### 1 Revise Plan Interview.

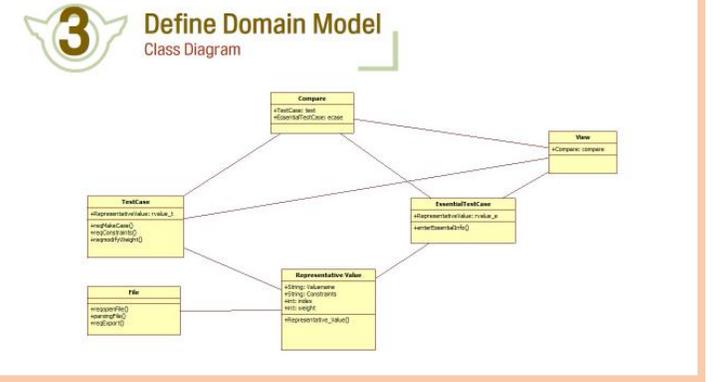
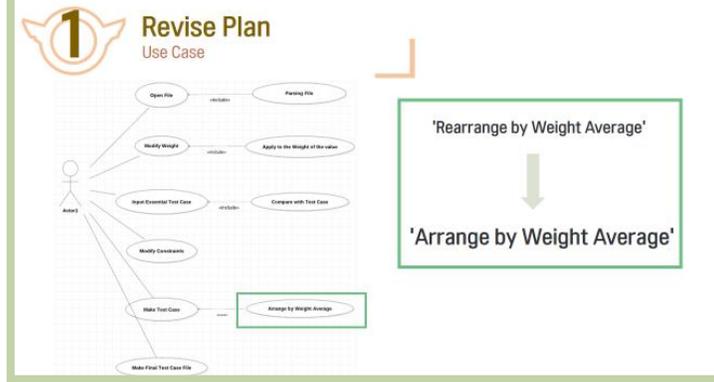


현업에서 쓰이는 test case 조합이 있는가?

- TDD(Test Driven Development) 를 사용
1. 내가 알고자 하는 결과를 생각(기능) 및 어떻게 테스트 할 것인가?
  2. 간단한 테스트를 작성(기능에 최적화 된)
  3. 테스트를 또 작성하는데 이번에는 Fail이 뜨도록 일부러 작성
  4. 마지막 테스트는 Pass가 나오도록 작성
  5. 결과를 보며, 종종 코드 혹은 Logic, 불분명한 코드, 리팩토링 가능한 코드를 찾아서 고치다.
  6. 마지막 모든 테스트 돌려서 Pass하도록 경리.
  7. 완료 되었다고 생각할 때까지 테스트 => GitHub(다른 사람 코드와 합침)=> 다시 테스트 케이스 돌림 => Deploy(서버에 올림) => Test=> Release



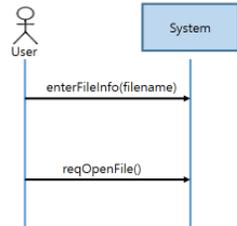
Jay  
Software Engineer



#### 5 Define System Sequence Diagram

Use Case : Open File

1. User가 파일에 대한 정보를 입력한다.
2. 파일을 불러올 것을 요청한다.
3. 요청한 파일을 불러온다.



#### 7 Refine System Test Case

Test Number	Test 항목	Description	Use Case	System Function
1 - 1	파일 열기 시험	파일 열기 버튼을 눌러 액셀 파일을 여는 기능 Test	Open File	R 1.1
1 - 2	파일 열기 시험	파일 열기 버튼을 눌러 확장자가 다른 파일을 여는 기능 Test	Open File	R 1.1
2	파일 파싱 시험	엑셀 파일을 읽어서 프로그램에 넣어 주는 기능 Test	Parsing File	R 1.2
3 - 1	가중치 적용 시험	사용자가 입력한 가중치를 적용하는 기능 Test	Apply to the Weight of the value	R 2.2
3 - 2	가중치 적용 시험	사용자가 가중치를 입력하지 않았을 때 기능 Test	Apply to the Weight of the value	R 2.2
4	가중치 수정 시험	가중치를 수정하는 기능 Test	Modify Weight	R 2.1
5 - 1	중요 Test Case 입력 시험	입력 버튼을 눌러 중요 Test Case 입력하는 기능 Test	Input Essential Test Case	R 3.1
5 - 2	중요 Test Case 입력 시험	휴무 하지 않는 Index를 추가 했을 때 중요 Test Case 입력 기능 Test	Input Essential Test Case	R 3.1

#### 7 Refine System Test Case

Test Number	Test 항목	Description	Use Case	System Function
6	Test Case 비교 시험	프로그램에 의해 조합된 Test Case와 사용자가 입력한 중요 Test Case를 비교하는 기능 Test	Compare with Test Case	R 3.2
7	제한 조건 수정 시험	제한 조건을 수정하는 기능 Test	Modify Constraint	R 4.1
8 - 1	Test Case 목록 생성 시험	Test Case 목록 생성 기능 Test	Make Test Case	R 5.1
8 - 2	Test Case 목록을 생성하는데 걸리는 시간 시험	Test Case가 조합되는 시간이 5초 미만인지 Test	Make Test Case (Nonfunctional Requirement)	R 5.1
9	가중치에 의한 정렬 시험	가중치에 대한 정렬 Test	Arrange by Weight	R 5.2
10	최종 Test Case 목록 파일 생성 시험	최종적으로 생성된 Test Case를 파일 출력 버튼을 눌러 파일로 내보내는 기능 Test	Make Final Test Case File	R 6.1

### 1st System Sequence Diagram

### System Test Case

# 2

## Stage 2040

Refine '000', Design

### 1 Revise Plan Project Scope



How to guarantee?

### 1 Revise Plan Functional Requirements

#### Version 2

Function	Description
Open File	작성 파일을 연다.
Parsing File	작성파일을 읽어 프로그램 내부에 저장.
Modify Weight	가중치 수정
Input Essential Test Case	중요 테스트 케이스 입력
Compare with Test Case	프로그램에 의해 호출 된 테스트 케이스와 사용자가 입력한 중요 테스트 케이스를 비교
Modify Constraints	제한 조건 수정
Make Test Case	테스트 케이스 생성
Arrange by Weight Average	가중치 평균에 따른 정렬
Make Final Test Case File	최종 테스트 케이스 파일 생성

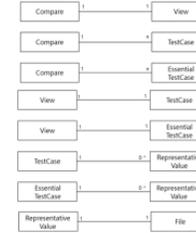
#### Version 3

Function	Description
Open File	작성 파일을 읽고 소스에 맞게 프로그램 내부에 저장
Modify Weight	가중치 수정
Input Essential Test Case	중요 테스트 케이스 입력
Compare with Test Case	프로그램에 의해 호출 된 테스트 케이스와 사용자가 입력한 중요 테스트 케이스를 비교
Modify Constraints	제한 조건 수정
Make Test Case	테스트 케이스 생성
Arrange by Weight Average	가중치 평균에 따른 정렬
Make Final Test Case File	최종 테스트 케이스 파일 생성

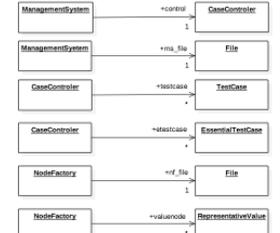
## 2nd Domain Model

### 1 Revise Plan Domain Model

#### Version 1



#### Version 2



## 3rd UseCase Diagram

### 1 Revise Plan Use Cases

#### Version 2



#### Version 3





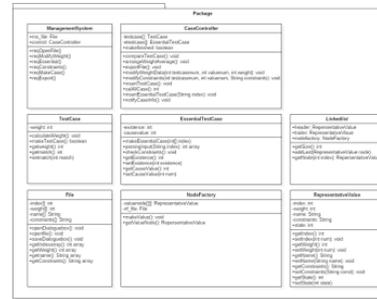
# Stage 2040 Design



## 3 State Chart Diagram

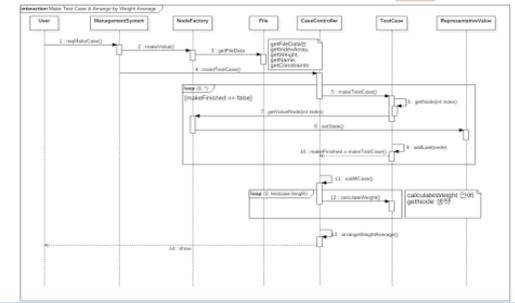


## 5 Refine System Architecture

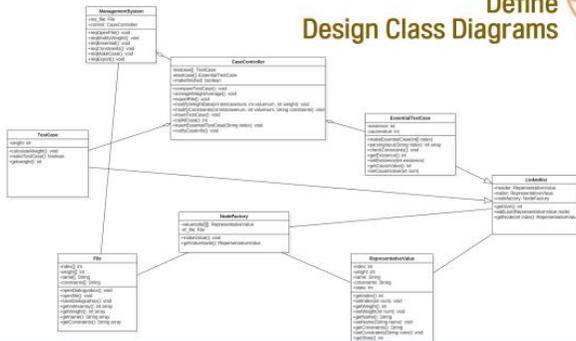


## 1st Sequence Diagram

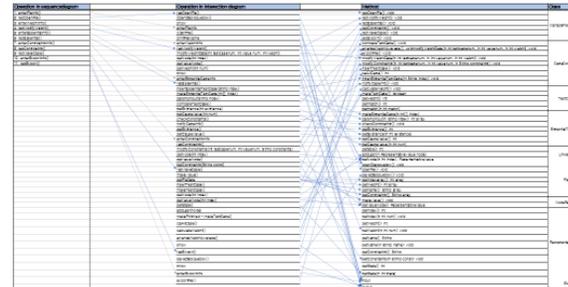
## 6 Define System Sequence Diagram Make Test Case & Arrange by Weight Average



## 7 Define Design Class Diagrams



## 8 Design Traceability Analysis



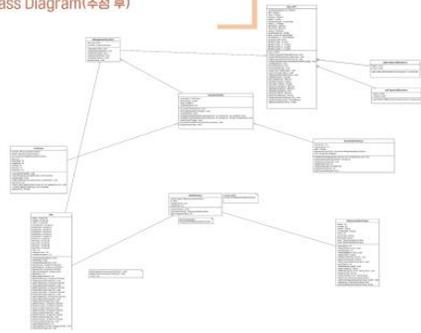
## 1st Class Diagram

# 2

## Stage 2050&2060

### Construct & Testing

#### 1 Modified Part Class Diagram(수정 후)



#### 3 Implement Windows



#### 4 Write Unit Test Code

```

import org.junit.Before;
import org.junit.Test;
import org.junit.Assert.*;

public class FileTest {

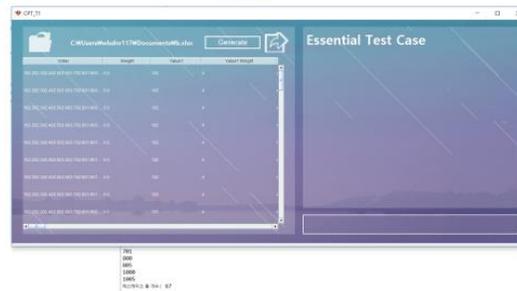
    #Before
    public void setUp() throws Exception {
        File file = new File();
        CaseController control = new CaseController(file);
    }

    #Test
    public void test() {
        ManagementSystem m = new ManagementSystem();
        File file = new File();
        CaseController control = new CaseController(file);
        assertEquals("file의 경우를 테스트하는 경우", m.read(file));
    }

    #Test
    public void test2() {
        assertEquals("file의 경우를 테스트하는 경우", m.read(file));
    }
}
  
```

#### 1 Unit Testing File Name test

#### 2 Testing



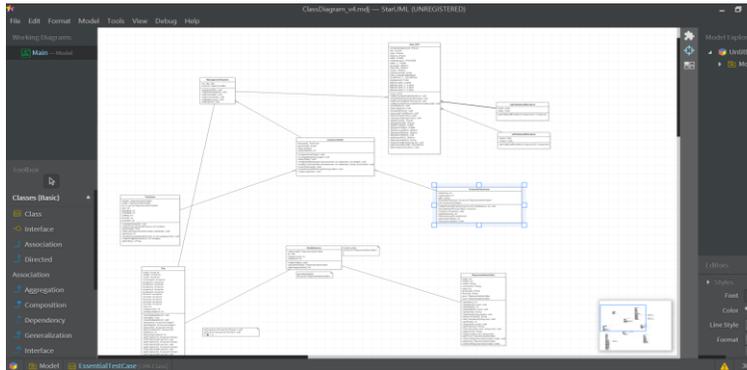
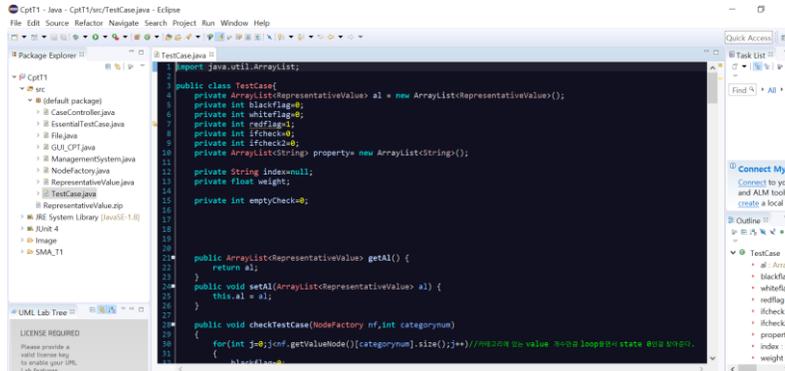
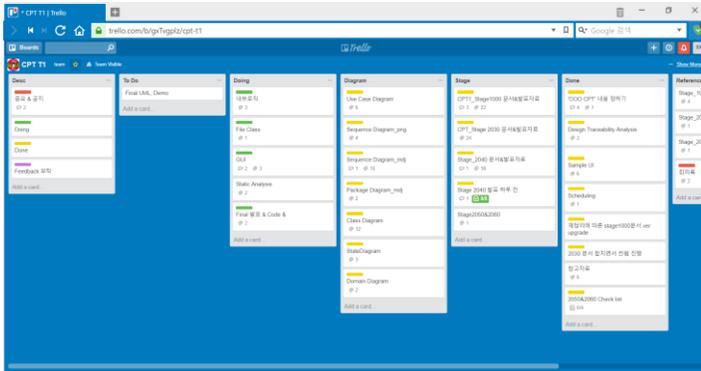
#### 2 System Testing

Test Number	Test 항목	Description	Use Case	System Function	P/F
1-1	파일 열기 시험	파일 열기 버튼을 눌러 파일 열 수 있는 Test	Open File	R 1.1	Pass
1-2	파일 열기 시험	파일 열기 버튼을 눌러 확장자 가 다른 파일을 여는 Test	Open File	R 1.1	Pass
3-1	가중치 수정 시험	가중치를 수정하는 Test	Modify Weight	R 2.1	Pass
3-2	가중치 적용 시험	사용자가 입력한 가중치를 적용하는 Test	Modify Weight	R 2.1	Fail
3-3	가중치 적용 시험	사용자의 가중치를 입력하지 않았을 때 Test	Modify Weight	R 2.1	Fail
4-1	중요 Test Case 입력 시험	입력 버튼을 눌러 중요 Test Case 입력하는 Test	Input Essential Test Case	R 3.1	Pass
4-2	중요 Test Case 입력 시험	무효하지 않는 Index를 추가했을 때 중요 Test Case 입력하는 Test	Input Essential Test Case	R 3.1	Fail
5	Test Case 비교 시험	프로그램에 의해 조합된 Test Case와 사용자가 입력한 중요 Test Case를 비교하는 Test	Compare with Test Case	R 3.2	Fail
6	제한 조건 수정 시험	제한 조건을 수정하는 Test	Modify Constraint	R 4.1	Fail
7-1	Test Case 목록 생성 시험	Test Case 목록 생성 Test	Make Test Case	R 5.1	Pass

# 3

## Used Tool

Trello, Eclipse, StarUML, SonarQube..



PROJECTS	비전	부트 러닝 수	BUGS	취약점	코드 품질	대체할 예정
T1	1	1,506	3	0	1,576	12/18
T3_localhost	1	2,550	5	0	2,588	2017/06/04
T3_localhost	0.2	1,888	7	0	1,943	2017/06/05



한 학기동안 SMA를 들으며..



소모를 마무리하며 ,어떤 느낌이신가요?



**KW**

소프트웨어 모델링 및 분석 강의를 통해 대략적으로나마 설계와 분석에 대해 알게 되었습니다. 구현 이전 설계의 중요성과 그 방법, 구현 후의 정적 분석까지 체계적인 시스템이 있다는걸 알게 되었고 그 효율성에 대해서도 느끼게 된 유익한 시간이었습니다.

**DK**

OOPT를 활용하여 만들면서 프로그램에 대해 쉽게 이해할 수 있고, 체계적으로 제작이 가능하다는 것에 놀랐고, 관리하는것이 수월하다고 느꼈다. 단계별로 문서화작업을 하면서 시간이 많이 들었지만, 프로젝트가 진행되면서 문서화작업이 용이하고 필수적이라는것을 느꼈다.



한 학기동안 SMA를 들으며..



소모를 마무리하며 ,어떤 느낌이신가요?



항상 친절하게 가르쳐주신 조교님과 교수님께 감사의 말을 올린다.



매 Stage 마다 열심히 고민하고 최선을 다했던 수업이었습니다.  
새로운 것을 많이 배웠고 같이 열심히 프로젝트를 진행한 팀원들에게 고맙습니다.



한 학기동안 SMA를 들으며..



문서는 얼마나 쓰셨나요?



- PartitioningTool\_CPT2030\_수정중v2\_7.docx
- PartitioningTool\_CPT2030\_수정중v2\_6.docx
- PartitioningTool\_CPT2030\_수정중v2\_5.docx
- PartitioningTool\_CPT1\_v2.2\_수정중\_오타수...
- PartitioningTool\_CPT1\_v1-8.docx
- [T1]Static Analysis.docx
- [T1]Static Analysis (1).docx
- [T1]Stage2050\_v2.docx
- [T1]Stage2050\_v1.docx
- [T1]Stage2050\_v1 (2).docx
- [T1]Stage2050\_v1 (1).docx
- [T1]Stage2040\_미추가.docx
- [T1]Stage2040\_sequence수정.docx
- [T1]Stage2040\_sequence수정 (1).docx
- [T1]Stage2030\_v2.docx

- [T1]Stage1000\_Schedul...
- [T1]Stage1000\_v1-8.do...
- [T1]Stage1000\_v2.docx
- [T1]Stage1000\_v2.pdf
- [T1]Stage1000\_v3.docx
- [T1]Stage1000\_v3.pdf
- KakaoTalk\_20170413\_1...
- KakaoTalk\_20170413\_1...
- stage1000\_대본.docx
- T1\_PPT\_stage1000\_Fin...
- T1\_Report\_stage1000\_...
- T1\_stage1000.pptx
- UseCaseDiagram\_v1.png
- ngrok-stable-windows-amd64
- sonarqube-5.6.6
- sonar-runner-2.4
- T1
- T3\_source
- T4\_source
- sonar-project.properties
- domainmodel
- [T1]Stage2030\_Report\_...
- [T1]Stage2030\_v1\_3.do...
- [T1]Stage2030\_v2.pdf
- [T1]Stage2030\_v2\_3.do...
- 0414 2030 presentatio...
- 2030 USE CASE.png
- 2030CLD.jpg
- 2030UseCase.png
- Analyze(2030)\_Traceabi...
- CPT2035\_시퀀스이미지...
- domain.png
- T1\_stage2030.pptx
- T1\_stage2030\_ppt.pdf

- SequenceDiagram\_mdj\_v3
- SequenceDiagramPng\_v2
- [T1]Stage2040\_PPT.pdf
- [T1]Stage2040\_Report.docx
- [T1]Stage2040\_Report.pdf
- [T1]Stage2040PPT.pptx
- 2040\_Define\_DK (1).docx
- 2040\_Define\_DK.docx
- 2040\_Define\_KW.docx
- 2040\_Define\_SY.docx
- 2040\_Define\_SY\_수정1.docx
- ClassDiagram\_v3\_1.mdj
- Operation List.docx
- Statechart\_v1\_1.mdj
- T1\_stage2040.pptx
- T1Design\_Traceability\_Analysi

- CptT1
- Image
- JUnit
- [T1]Stage2050&2060PPT.pdf
- [T1]Stage2050&2060Report.pdf
- [T1]Stage2050PPT.pptx
- ClassDiagram\_v4.mdj
- ClassDiagram\_v4.png
- CptT1.zip
- 기본.png
- [T1]Stage2050PPT.pptx
- [T1]Static Analysis.pdf
- [T1]Static Analysis.pptx
- [T1]Static Analysis\_REPORT.docx
- [T1]Static Analysis\_REPORT.pdf
- [T1]Static Analysis\_v2.pdf
- [T1]Static Analysis\_v2.pptx
- [T1]Static Analysis\_v3.pdf

많이... 썼습니다...

Special



Thanks To

유준범 교수님, 정세진 조교님.

김헌진 (성결대학교 컴퓨터공학, GUI)

윤지석(Grepp Inc., Interview)

Yellow Sketch 매니저님(장소제공)