

Software Verification

- 1st System Test -



Team 3

201011320 김용현

201111360 손준익

201111384 정국빈

Index

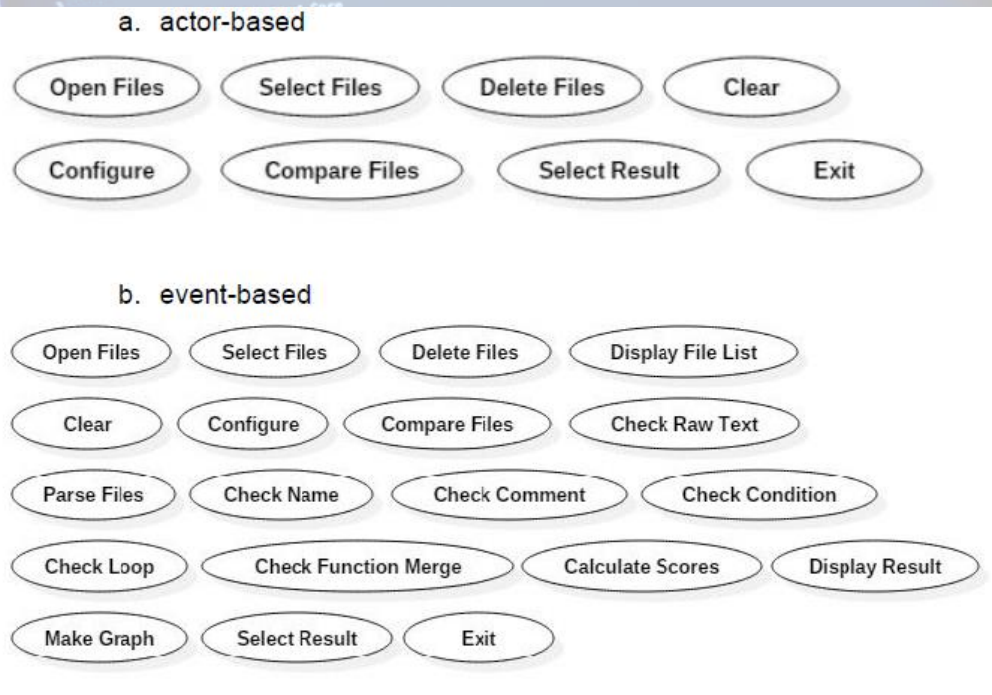
1. Specification Review
2. Category-partition Test
3. Pairwise Test
4. Brute Force Test
5. Overall

1. Specification Review



* Stage 1000 – 1007. Define Business Use Cases

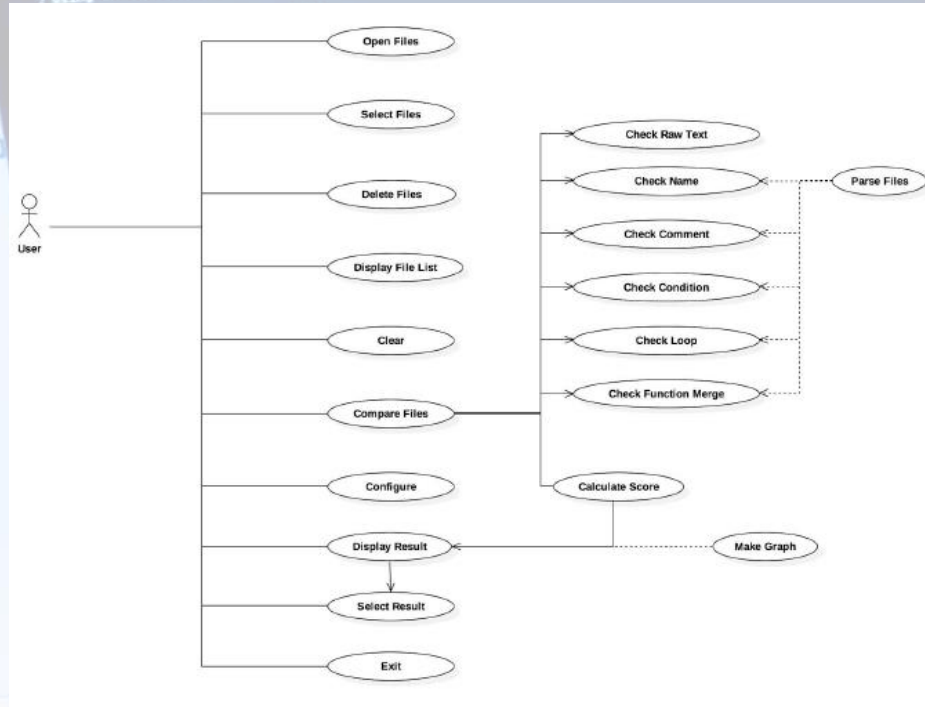
3. Identify Use Case



Actor-based Use Case가 Event-based에도 존재

* Stage 1000 – 1007. Define Business Use Cases

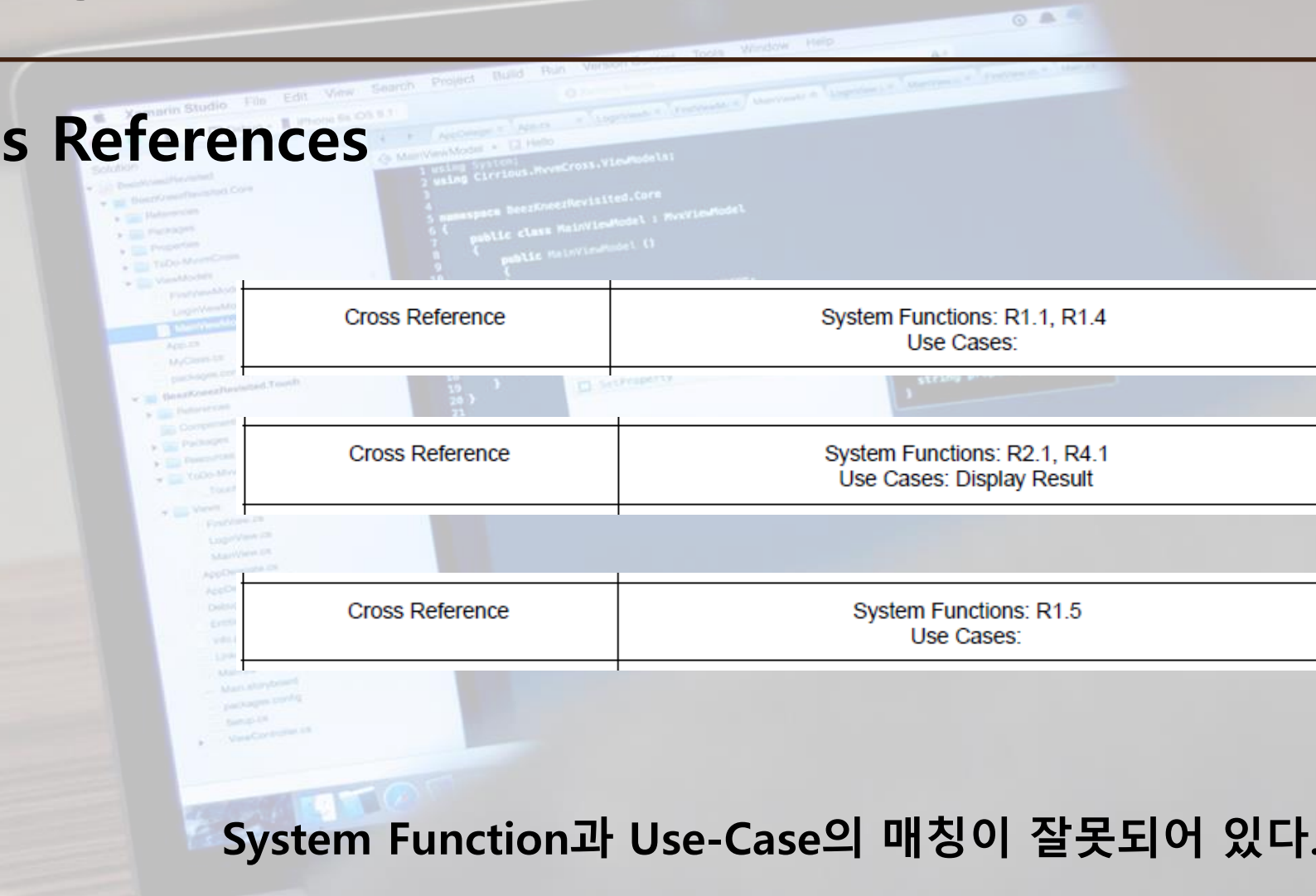
3. Identify Use Case



Event-based Use Case들이 User와 직접 연결되어 있다.
존재하지 않는 Check Function Merge 존재

* Stage 2030 – 2031. Define Essential Use Cases

Cross References



System Function과 Use-Case의 매칭이 잘못되어 있다.

* Stage 2030 – 2031. Define Essential Use Cases

2. Select Files

Use Case	Select Files
Actor	User
Purpose	User가 검사 대상 파일을 선택한다.
Overview	파일 목록에서 블록 지정을 하여 User가 검사 대상 파일을 선택한다.
Type	Secondary
Cross Reference	System Functions: R1.2 Use Cases:
Pre-Requisites	N/A
Typical Courses of Events	(A): Actor, (S): System 1. (A) 파일 목록에서 검사 대상 파일을 선택한다. 2. (S) 해당 파일을 블록 지정한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Pre-Requisites : Open Files

* Stage 2030 – 2031. Define Essential Use Cases

7. Compare

Use Case	Compare
Actor	User
Purpose	선택된 파일들을 1:1로 비교 분석한다.
Overview	선택된 파일들을 1:1로 비교 분석한다.
Type	Primary
Cross Reference	System Functions: R3.2, R3.3, R3.4, R3.5, R3.6, R3.7, R3.8, R4.1 Use Cases: Parse File, Check Raw Text, Check Name, Check Comment, Check Condition, Check Loop, Calculate Scores, Display Result
Pre-Requisites	N/A
Typical Courses of Events	(A): Actor, (S): System 1. (A) Compare를 클릭한다. 2. (S) 내부적으로 결과를 계산한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	E1. 검사 대상 파일이 없을 경우 에러를 띄운다.

Pre-Requisites : 2개 이상의 File이 Open or Select되어 있을 때

* Stage 2030 – 2031. Define Essential Use Cases

8. Check Raw Text, 9. Parse Files

Use Case	Check Raw Text
Actor	None
Purpose	두 파일이 텍스트 단위에서 얼마나 비슷한지 검사한다.
Overview	두 파일이 텍스트 단위에서 얼마나 비슷한지 검사한다.
Type	Primary
Cross Reference	System Functions: R3.1, R3.9, R4.1 Use Cases: Compare, Calculate Scores, Display Result
Pre-Requisites	N/A
Typical Courses of Events	(S): System 1. (S) 두 파일을 불러온다. 2. (S) 비교 검사한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	Parse File
Actor	None
Purpose	C 소스 파일을 파싱한다.
Overview	C 소스 파일을 파싱한다.
Type	Primary
Cross Reference	System Functions: R3.1, R3.4, R3.5, R3.6, R3.7 Use Cases: Compare, Check Name, Check Comment, Check Condition, Check Loop
Pre-Requisites	N/A
Typical Courses of Events	(S): System 1. (S) 소스 파일을 불러온다. 2. (S) 파싱한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Pre-Requisites : Compare

* Stage 2030 – 2035. Define System Sequence Diagrams

Display Result

Use Case	Display Result
Actor	None
Purpose	결과를 보여준다.
Overview	결과를 보여준다.
Type	Primary
Cross Reference	System Functions: R2.1, R3.9, R4.2, R4.3 Use Case: Configure, Calculate Scores, Make Graph, Select Result
Pre-Requisites	결과 계산이 마쳐져야 한다.
Typical Courses of Events	(S): System 1. (S) 결과 점수를 참조한다. 2. (S) Configure 정보를 참조한다. 3. (S) 조건에 맞는 결과를 보여준다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

Use Case	Name of Actor-Activated Event	System Operations
Open Files	Open Files	OpenFiles()
Select Files	Select Files	SelectFiles()
Delete Files	Delete Files	DeleteFiles()
Clear	Clear	Clear()
Configure	Configure	Configure()
Compare	Compare	CompareFiles()
Display Result	Display Result	DisplayResult()
Select Result	Select Result	SelectResult()
Exit	Exit	Exit()

Actor가 None인데 Actor-Activated Event에 존재

* Stage 2030 – 2036. Define Operation Contracts

Use Case	Name of Actor-Activated Event	System Operations
1.Open Files	1: Request opening files()	1.OpenFiles()
2.Select Files	1: Request selecting files()	2.SelectFiles()
3.Delete Files	1: Request deleting files()	3.Delete Files()
4.Display File List	N/A	N/A
5.Clear	1: Request clearing()	4.Clear()
6.Configure	1: Request configuring()	5.Configure()
7.Compare	1: Request comparing files()	6.CompareFiles()
8.Check Raw Text	N/A	N/A
9.Parse File	N/A	N/A
10.Check Name	N/A	N/A
11.Check Comment	N/A	N/A
12.Check Condition	N/A	N/A
13.Check Loop	N/A	N/A
14.Calculate Scores	N/A	N/A
15.Display Result	Request displaying result()	7.DisplayResult()
16.Make Graph	N/A	N/A
17.Select Result	Request Select Result	8.Select Result()
18.Exit	Request exiting()	9.Exit()

- Request 앞에 1:이 있다가 생략된다.

- Actor가 None인 Actor-Activated Event 존재

- System Operation들의 Cross References와 Pre-Condition들이 2031과 불일치한 항목 발견

* Stage 2030 – 2038. Refine System Test Case

1-5. File Delete Test

R1.3	Delete Files	<ul style="list-style-type: none"> - 선택한 파일을 삭제했을 때 파일 목록에서 없어졌는지 확인한다. - 다수의 파일을 한꺼번에 삭제 가능한지 확인한다.
------	--------------	--



1-5	File Delete Test	- 파일을 선택하지 않고 삭제하려 할 때 경고창이 잘 뜨는지 확인한다.	Delete Files	R1.3
1-6	File Delete Test	- 선택한 파일을 삭제할 때 파일 목록에서 없어지는지 확인한다.	Delete Files	R1.3
1-7	File Delete Test	- 다수의 파일을 한꺼번에 삭제할 수 있는지 확인한다.	Delete Files	R1.3

1009 단계의 Delete Files의 Case를 나누면서 새로운 Case 등장

* Stage 2030 – 2038. Refine System Test Case

4-2. Display Result Test

R4.1	Display Result	- 결과 목록 및 그래프가 잘 작성되는지 확인한다.
------	----------------	------------------------------



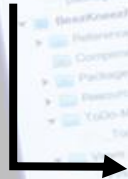
4-1	Display Result Test	- 결과 목록의 값과 그래프의 값이 일치하는지 확인한다.	Display Result	R4.1
4-2	Display Result Test	- 그래프가 표시되는지 확인한다.	Display Result	R4.1

1009 단계의 Display의 Case를 나누면서 새로운 Case 등장

* Stage 2030 – 2038. Refine System Test Case

4-3, 4-4. Make Graph Test

R4.2	Make Graph	- 그래프가 잘 작성되는지 확인한다.
------	------------	----------------------



4-3	Make Graph Test	- 측정된 값이 꺾은선 그래프로 잘 표시되는지 확인한다.	Make Graph	R4.2
4-4	Make Graph Test	- 측정된 값의 평균 값이 꺾은선 그래프로 잘 표시되는지 확인한다.	Make Graph	R4.2

1009 단계의 Make Graph의 Case를 나누면서 새로운 Case 등장

* Stage 2040 – 2041. Design Real Use Cases

15. Display Result

Use Case	Display Result
Actor	None



Use Case	Display Result
Actor	User or none

2031에 없던 Actor인 User 등장

* Stage 2040 – 2041. Design Real Use Cases

15. Select Result

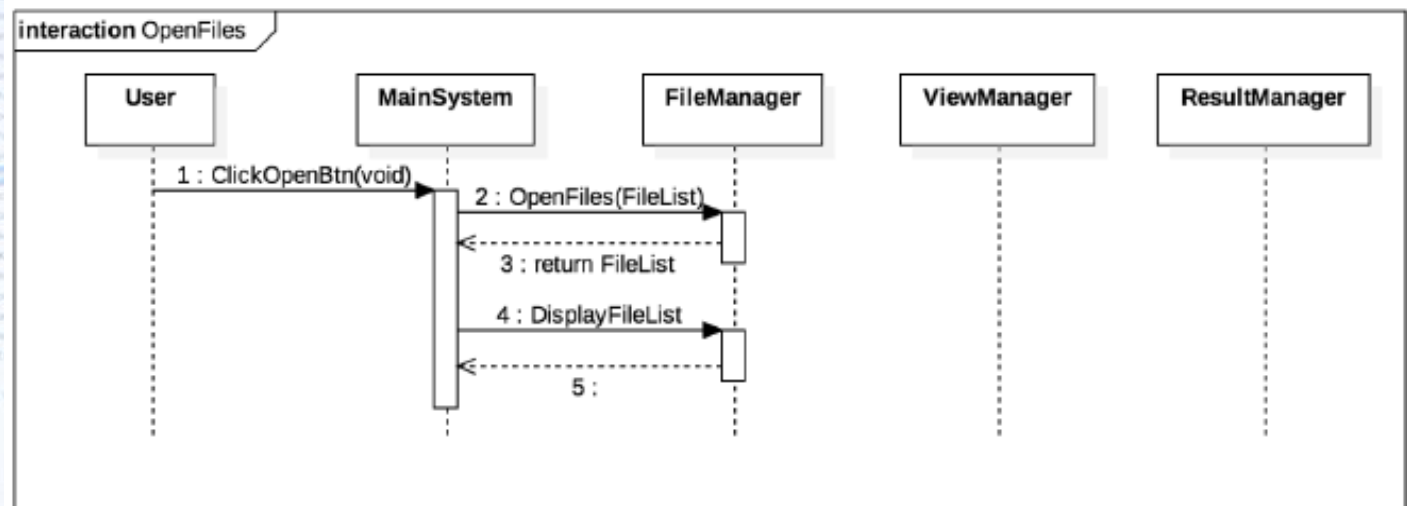
Pre-Requisites	N/A
----------------	-----



Pre-Requisites	결과 점수 계산이 모두 마쳐져야 한다.
----------------	-----------------------

2031과 Pre-Requisites 불일치

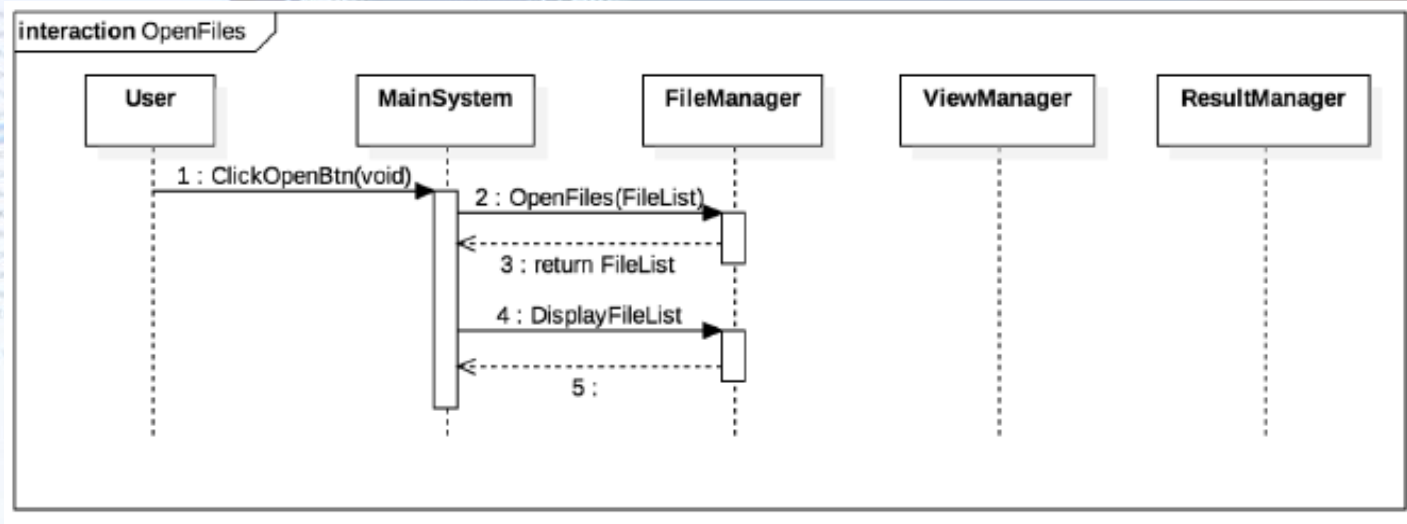
* Stage 2040 – 2044. Define Interaction Diagrams



2035에 존재하던 Display Result가 이번 단계에서 생략되어 있다.
Sequence Diagrams마다 사용하지 않는 Class들 생략 필요

* Stage 2040 – 2044. Define Interaction Diagrams

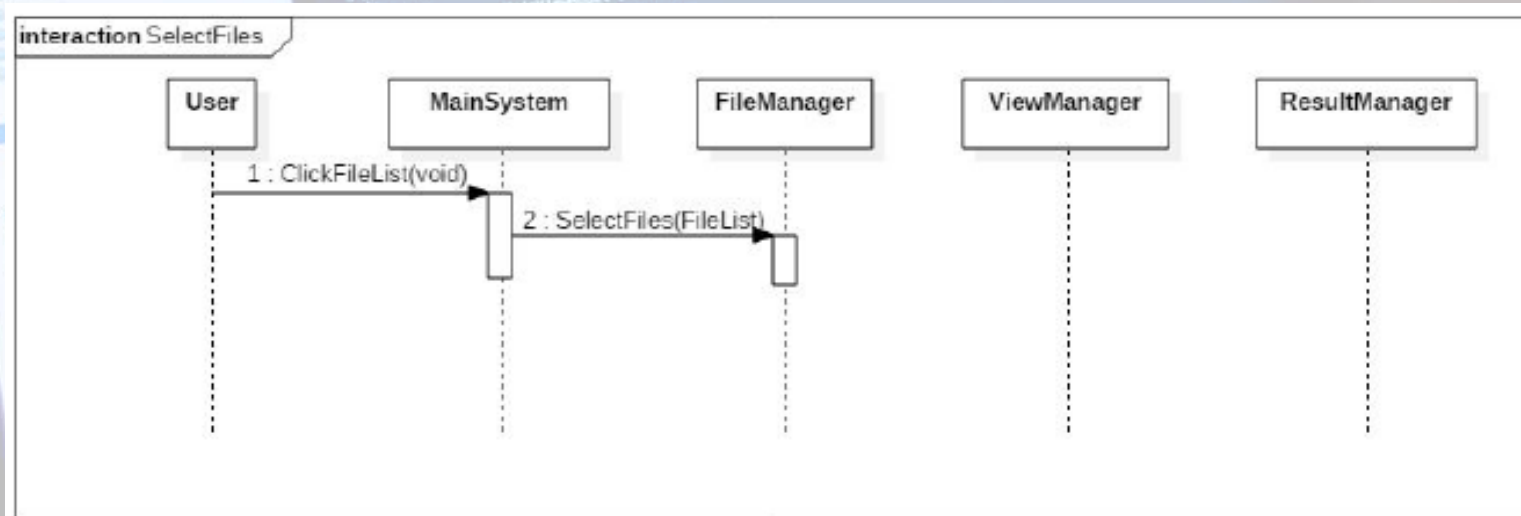
1. Open Files



Return값이 없는데 5번 화살표가 존재한다.

* Stage 2040 – 2044. Define Interaction Diagrams

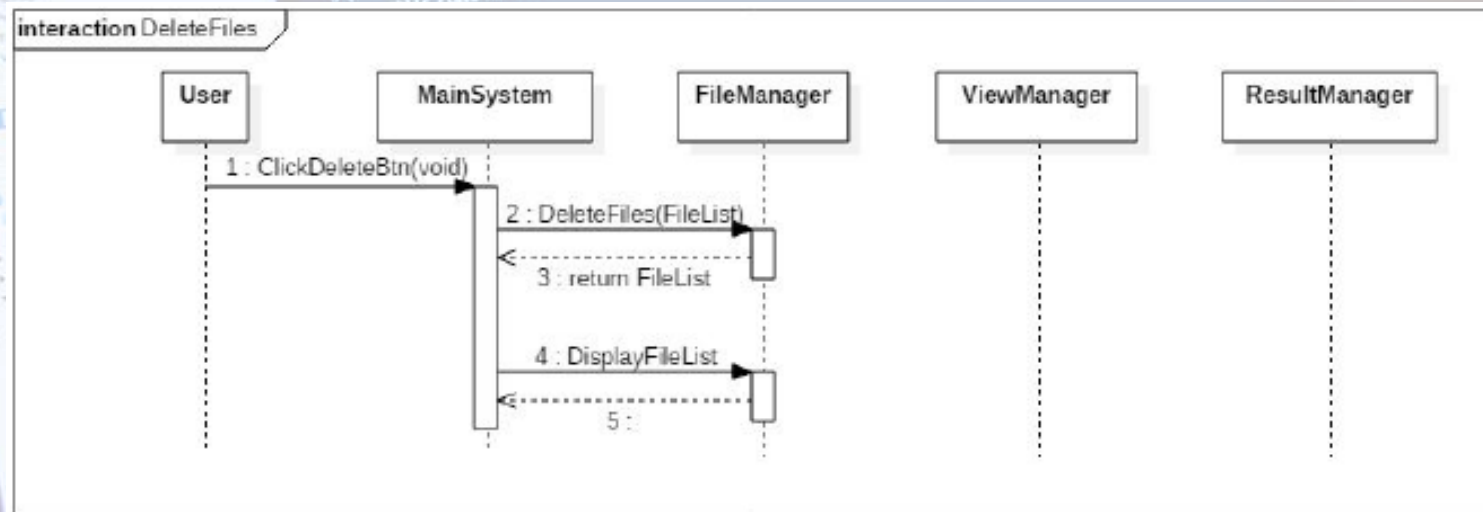
2. Select Files



MainSystem에 ClickFileList함수가 존재하지 않는다.

* Stage 2040 – 2044. Define Interaction Diagrams

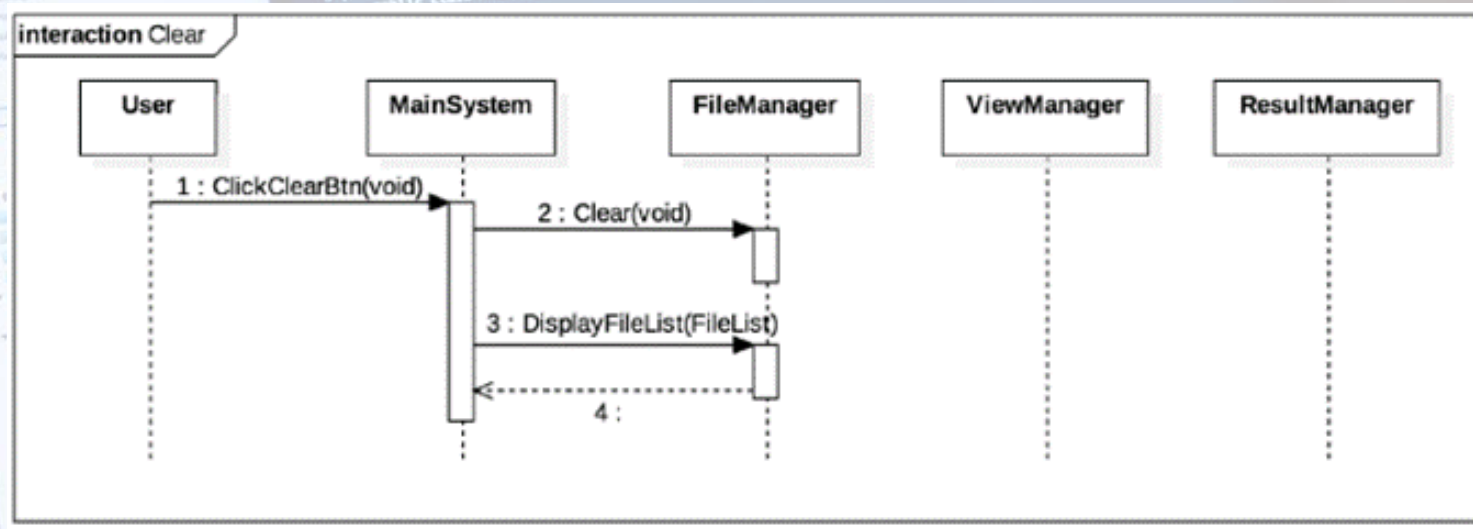
3. Delete Files



Return값이 없는데 5번 화살표가 존재한다.

* Stage 2040 – 2044. Define Interaction Diagrams

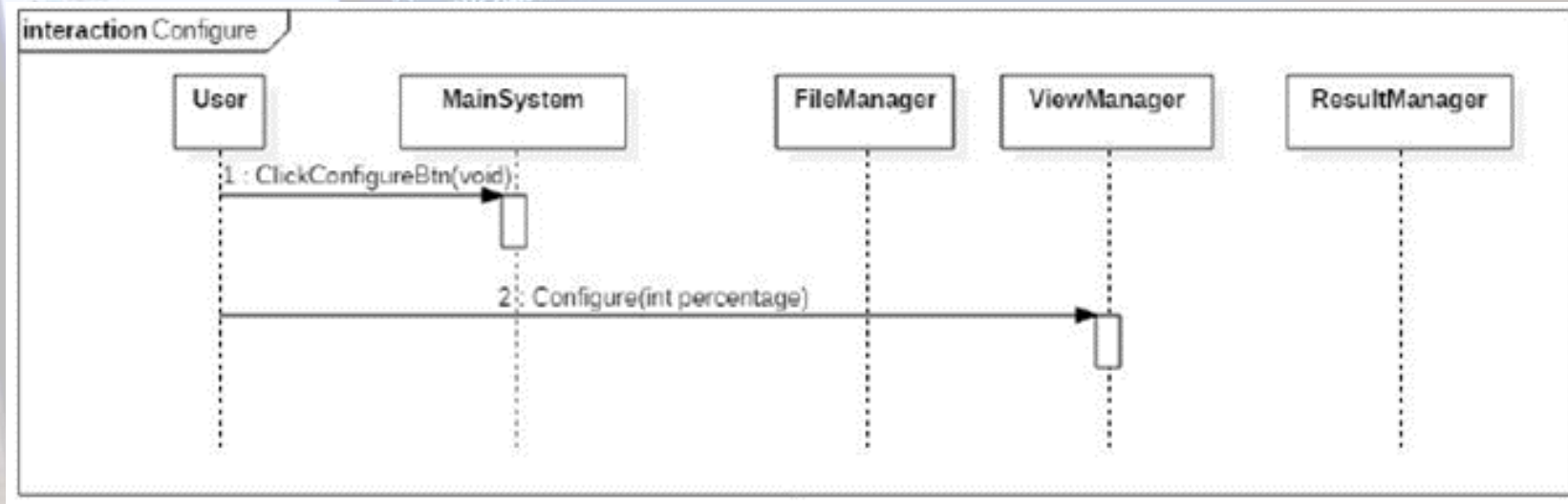
4. Clear



Return값이 없는데 4번 화살표가 존재한다.

* Stage 2040 – 2044. Define Interaction Diagrams

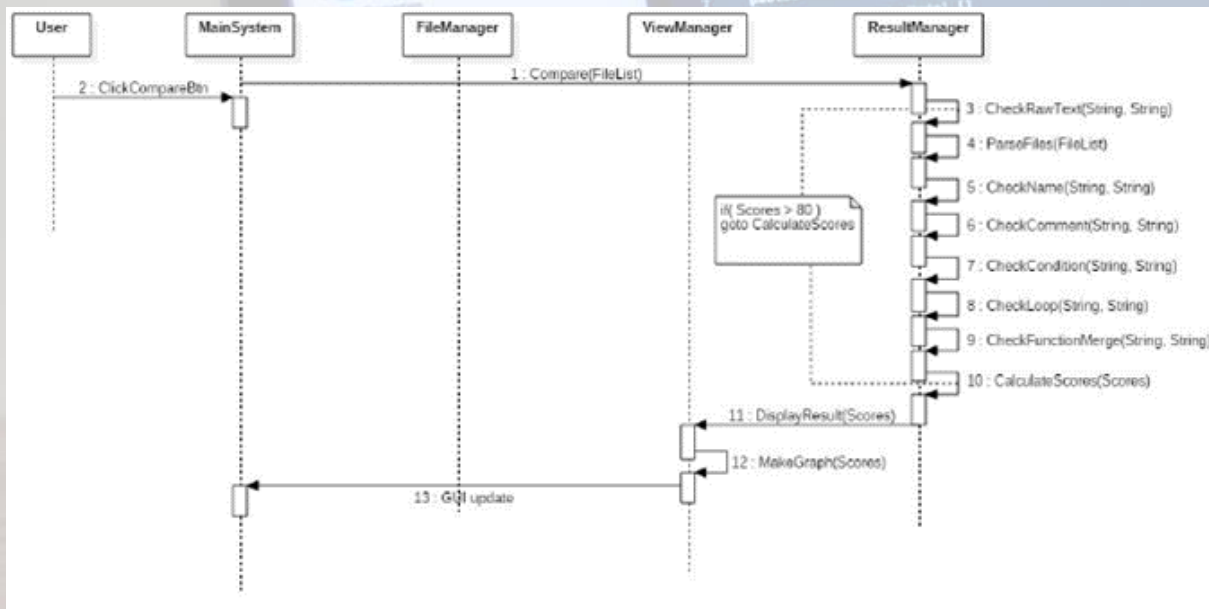
5. Configure



실제 동작 시 ClickConfigureBtn에서 showConfigureDialog을 호출하는데 Sequence Diagram에는 표시되어 있지 않다.

* Stage 2040 – 2044. Define Interaction Diagrams

6. Compare



- 1, 2번 순서가 바뀌어 있다.

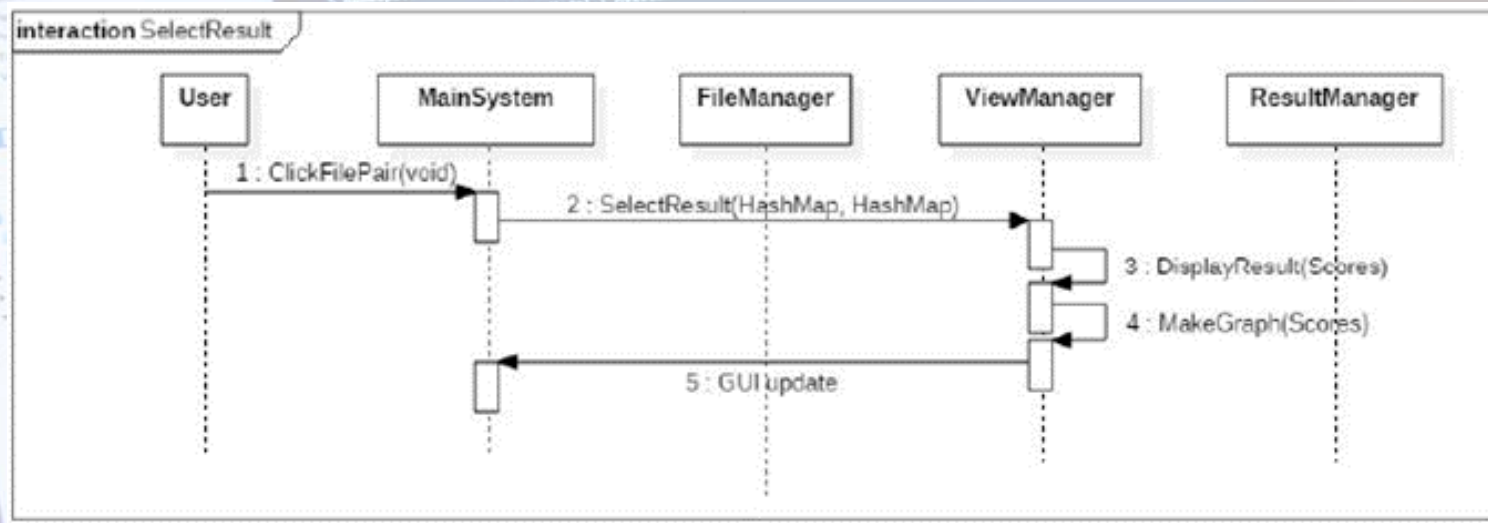
- 1. Compare의 생명선이
10. CalculateScores까지 내려와야 된다.

- 11. DisplayResult가 ViewManager에
존재하지 않고 MainSystem에 존재한다.

- 2. ClickComapreBtn의 생명선이
DisplayResult까지 내려와야 된다.

* Stage 2040 – 2044. Define Interaction Diagrams

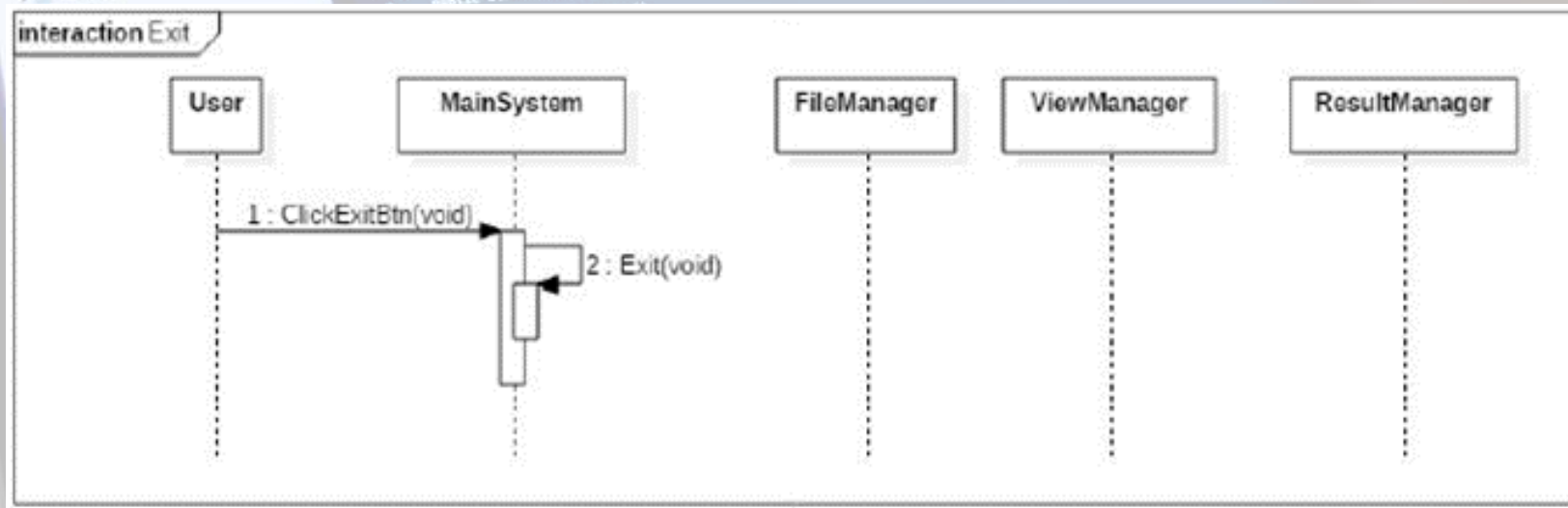
7. Select Result



MainSystem에 ClickFilePair함수가 존재하지 않는다.

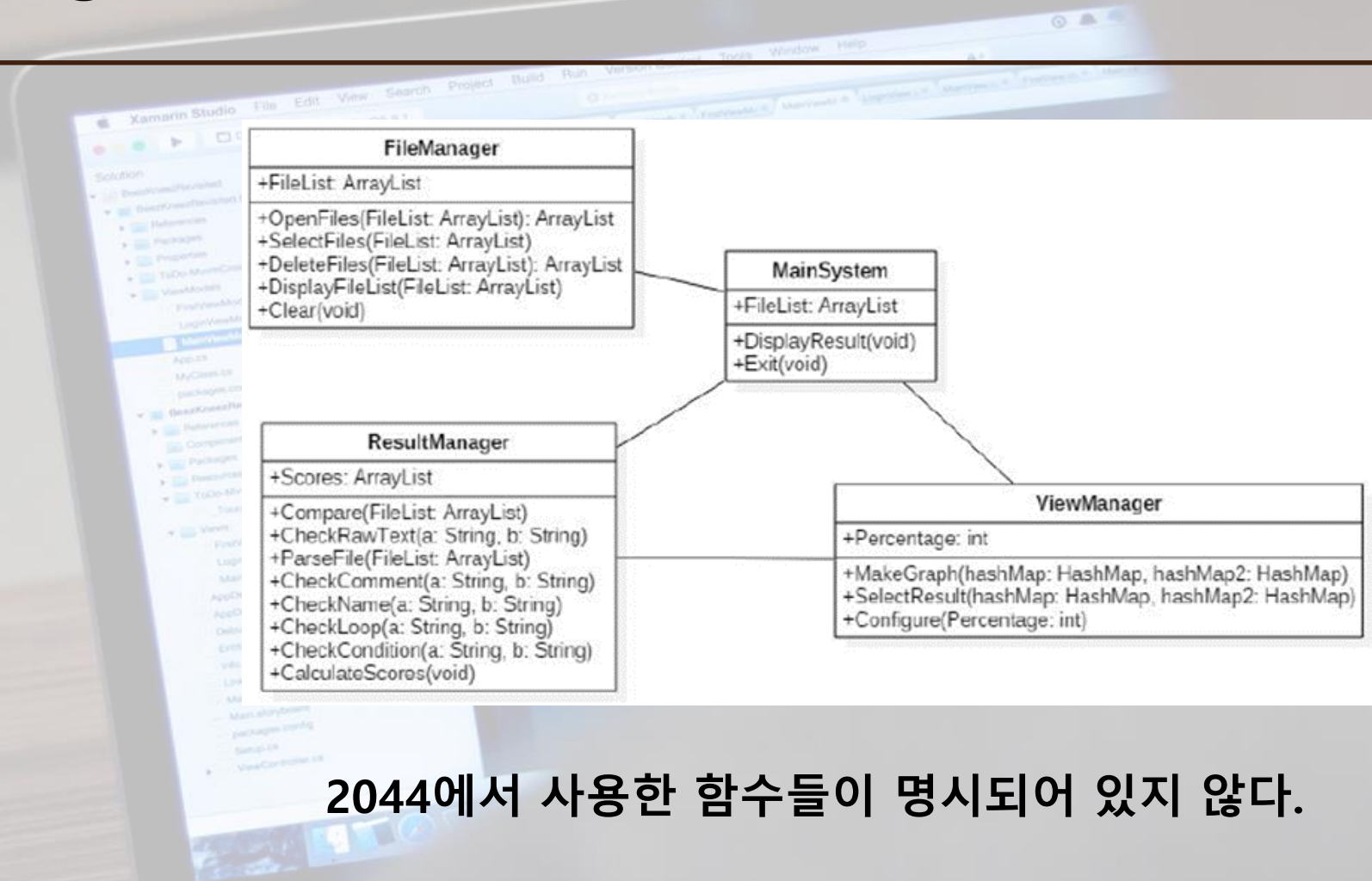
* Stage 2040 – 2044. Define Interaction Diagrams

8. Exit



1번과 2번 생명선이 같이 끝나야 된다.

* Stage 2040 – 2045. Define Design Class Diagrams



2044에서 사용한 함수들이 명시되어 있지 않다.

* Stage 2050 – 2051. Method Definitions

1. OpenFiles()

Type	Method
Name	OpenFiles
Purpose	User가 검사 대상 파일을 열 수 있도록 한다
Cross References	Functions: R1.1, R1.4 Use Cases
Input (Method)	FileList: ArrayList
Output (Method)	FileList: ArrayList
Abstract Operation (Method)	1. 인자로 받은 FileList를 FileManager.FileList에 추가하고, FileManger.FileList를 Return한다. 2. DisplayFileList에 FileList를 인자로 호출한다.
Exceptional Courses of Events	E1. 이미 FileList에 동일한 이름을 가진 파일이 있을 경우, 경고창을 띄운다.

Return 이후의 동작이 서술되어 있다.
실제 코드 상에는 DisplayFileList를 호출하는 구문 존재하지 않는다

* Stage 2050 – 2051. Method Definitions

2. SelectFiles()

Type	Method
Name	SelectFiles
Purpose	User가 검사 대상 파일을 선택한다.
Cross References	Functions: R1.2 Use Cases:
Input (Method)	FileList: ArrayList
Output (Method)	

실제 코드 상에서는 index값을 Return한다.

* Stage 2050 – 2051. Method Definitions

3. DeleteFiles()

Type	Method
Name	DeleteFiles
Purpose	검사 대상 파일을 파일 목록에서 삭제한다.
Cross References	Functions: R1.3 Use Cases:
Input (Method)	FileList: ArrayList
Output (Method)	
Abstract Operation (Method)	1. 인자로 받은 FileList를 FileManager.FileList에서 삭제하고, FileManger.FileList를 Return한다. 2. DisplayFileList에 FileList를 인자로 호출한다.
Exceptional Courses of Events	E1. 선택된 파일이 없을 경우 아무것도 삭제하지 않는다.

Output이 없는데 Return한다고 쓰여 있다.

* Stage 2050 – 2051. Method Definitions

7. Comapre()

Type	Method
Name	Compare
Purpose	선택된 파일들을 1:1로 비교 분석한다.
Cross References	Functions: R3.2, R3.3, R3.4, R3.5, R3.6, R3.7, R3.8, R4.1 Use Cases: Check Raw Text, Parse Files, Check Name, Check Comment, Check Condition, Check Loop, Calculate Scores, Display Result
Input (Method)	FileList: ArrayList
Output (Method)	
Abstract Operation (Method)	<ol style="list-style-type: none"> FileList의 파일들의 조합 A File, B File을 만들어 1:1로 비교를 한다. <ol style="list-style-type: none"> ResultManager.CheckRawText의 결과가 80을 초과하는 경우, CalculateScores로 진행한다. ResultManager.CheckRawText의 결과가 80 이하일 경우, ParseFile)를 진행한다. CheckName, CheckComment, CheckCondition, CheckLoop의 결과로 Scores를 업데이트 한다. ViewManager.DisplayResult와 ViewManager.MakeGraph를 진행하여 Result window의 GUI를 업데이트 한다.

ParseFile이 80이하일 경우가 아니라 맨 처음에 무조건 동작한다.
 DisplayResult가 ViewManager가 아니라 MainSystem에 존재한다.

* Stage 2050 – 2051. Method Definitions

8. CheckRawText()

Type	Method
Name	CheckRawText
Purpose	두 파일이 텍스트 단위에서 얼마나 비슷한지 검사한다.
Cross References	Functions: R3.1, R3.9, R4.1 Use Cases: Compare Files, Calculate Scores, Display Result
Input (Method)	a: String, b: String
Output (Method)	percentage: int
Abstract Operation (Method)	1. 두 파일을 input으로 받아서 텍스트 단위에서 정량적으로 얼마나 일치하는지 검사한다. 2. 검사 결과 수치를 Scores 배열에 저장한다.
Exceptional Courses of Events	N/A

Cross References : Functions 3.9가 존재하지 않는다.

* Stage 2050 – 2051. Method Definitions

18. Configure()

Type	Method
Name	Configure
Purpose	몇 퍼센트 이상의 결과를 보여줄 것인지 설정하도록 한다.
Cross References	Functions: R2.1, R4.1 Use Cases: Display Result
Input (Method)	Percentage: int
Output (Method)	N/A
Abstract Operation (Method)	<ol style="list-style-type: none"> 1. Configure window에서 0-100 사이의 정수를 받아 Apply를 받았을 경우 int Percentage를 업데이트 한다. 2. Display Result 시 업데이트 된 Percentage 이상의 결과만을 표시한다.
Exceptional Courses of Events	E1. 설정한 Filter range(10)가 0 미만이거나 100을 초과할 경우 에러를 띄운다.

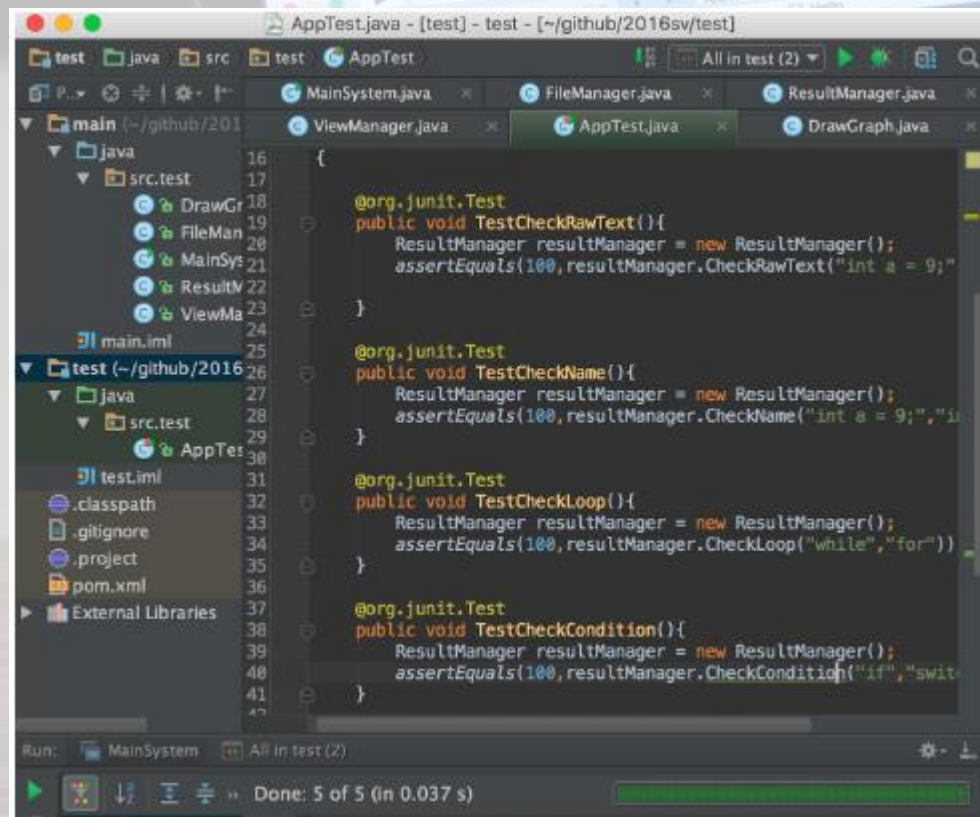
- 초기값 50 설명 누락

- 몇 퍼센트 이상 -> 몇 퍼센트 초과

- Abstract Operation : 2번 생략 가능
(DisplayResult에서 이미 서술)

- Exceptional Courses of Events :
에러를 띄우지 않고 기존 값으로 적용

* Stage 2060 – 2061. Unit Testing



The screenshot shows an IDE window titled 'AppTest.java - [test] - test - [~/github/2016sv/test]'. The left sidebar displays a project tree with folders 'test', 'java', and 'src.test'. The main editor area shows the following Java code:

```
16 {
17
18     @org.junit.Test
19     public void TestCheckRawText(){
20         ResultManager resultManager = new ResultManager();
21         assertEquals(100, resultManager.CheckRawText("int a = 9;"));
22     }
23
24
25     @org.junit.Test
26     public void TestCheckName(){
27         ResultManager resultManager = new ResultManager();
28         assertEquals(100, resultManager.CheckName("int a = 9;"));
29     }
30
31     @org.junit.Test
32     public void TestCheckLoop(){
33         ResultManager resultManager = new ResultManager();
34         assertEquals(100, resultManager.CheckLoop("while", "for"));
35     }
36
37     @org.junit.Test
38     public void TestCheckCondition(){
39         ResultManager resultManager = new ResultManager();
40         assertEquals(100, resultManager.CheckCondition("if", "switch"));
41     }
42 }
```

At the bottom, the 'Run' status bar indicates 'Done: 5 of 5 (in 0.037 s)'.

- Input값과 Output이 존재하는 함수들이 많은데 5개의 함수만 Testing했다.
- Test Case 자체가 적절하지 않다.
ex) 반복문 비교시 단순한 while, for 문자열 비교

* Stage 2060 – 2063. System Testing

1-1	File Open Test	Select한 파일이 하이라이팅이 잘 되는지 확인한다.	Open Files	R1.1	P
1-2	File Open Test	Open을 한 File이 List에 잘 뜨는지 확인한다.	Open Files	R1.1	P
1-3	File Select Test	Select한 파일이 하이라이팅이 잘 되는지 확인한다.	Select Files	R1.2	P

1-1과 1-3의 Description이 동일하다.

1-5	File Delete Test	파일을 선택하지 않고 삭제하려 할 때 경고창이 잘 뜨는지 확인한다.	Delete Files	R1.3	P
-----	-------------------------	---------------------------------------	--------------	------	---

경고창이 뜨지 않는다.

2-2	Configure Test	설정값이 음수이거나 100을 넘어갔을 때 경고창이 뜨는지 확인한다.	Configure	R2.1	P
-----	-----------------------	---------------------------------------	-----------	------	---

경고창이 뜨지 않는다.

2. Category-partition Test



* Testable Units

Group	Category
File	Open File 확장자
	Open File 수
	중복 파일 이름
	Select File 수
	Clear
	Delete
	Compare
Configure	Filter Range Value
Compare	Check Raw Text
	Check Name
	Check Comment
	Check Condition
	Check Loop
Result	Select Pair 수
	Display Pair 수

* Representative Values (1/4)

Category	Values	Number
Open File 확장자	.c	1000
	.etc	1001
Open File 수	0	1100
	1	1101
	n	1102
파일 이름 중복	이름이 중복되는 파일이 존재할 때	1200
	이름이 중복되는 파일이 존재하지 않을 때	1201
Select File 수	0	1300
	1	1301
	n	1302
Delete	Delete를 수행한다.	1400
Clear	파일 목록에 파일이 존재할 때	1401
	파일 목록에 파일이 존재하지 않을 때	1500
Compare	Compare를 수행한다.	1600

* Representative Values (2/4)

Category	Values	Number
Filter Range Value	0 미만	2000
	0	2001
	50	2002
	100	2003
	100초과	2004
	Malformed	2005
Check Raw Text	A의 단어 수 T_n , B의 단어 수 T_m , 같은 단어 수 T_n	3000
	A의 단어 수 T_n , B의 단어 수 T_m , 같은 단어 수 $T_n/2$	3001
	A의 단어 수 T_n , B의 단어 수 T_m , 같은 단어 수 0	3002
Check Name	A의 이름 수 N_n , B의 이름 수 N_m , 같은 이름 수 $N_n(N_n < N_m)$	3100
	A의 이름 수 N_n , B의 이름 수 N_m , 같은 이름 수 $N_n/2(N_n < N_m)$	3101
	A의 이름 수 N_n , B의 이름 수 N_m , 같은 이름 수 0	3102

* Representative Values (3/4)

Category	Values	Number
Check Comment	A의 총 주석('//') 수 Cn, B의 총 주석('//') 수 Cm, 같은 내용의 주석 수 Cn(Cn<Cm)	3200
	A의 총 주석('//') 수 Cn, B의 총 주석('//') 수 Cm, 같은 내용의 주석 수 Cn/2(Cn<Cm)	3201
	A의 총 주석('//') 수 Cn, B의 총 주석('//') 수 Cm, 같은 내용의 주석 수 0	3202
	A의 총 주석('/* */) 수 Cn, B의 총 주석('/* */) 수 Cm, 같은 내용의 주석 수 Cn(Cn<Cm)	3203
	A의 총 주석('/* */) 수 Cn, B의 총 주석('/* */) 수 Cm, 같은 내용의 주석 수 Cn/2(Cn<Cm)	3204
	A의 총 주석('/* */) 수 Cn, B의 총 주석('/* */) 수 Cm, 같은 내용의 주석 수 0	3205
	A의 총 주석('//') 수 Cn, B의 총 주석('/* */) 수 Cm, 같은 내용의 주석 수 Cn(Cn<Cm)	3206
	A의 총 주석('//') 수 Cn, B의 총 주석('/* */) 수 Cm, 같은 내용의 주석 수 Cn/2(Cn<Cm)	3207
	A의 총 주석('//') 수 Cn, B의 총 주석('/* */) 수 Cm, 같은 내용의 주석 수 0	3208
Check Condition	A의 총 if문 수 Bn, B의 총 if문 수 Bm, 같은 내용의 if문 수 Bn(Bn<Bm)	3300
	A의 총 if문 수 Bn, B의 총 if문 수 Bm, 같은 내용의 if문 수 Bn/2(Bn<Bm)	3301
	A의 총 if문 수 Bn, B의 총 if문 수 Bm, 같은 내용의 if문 수 0	3302
	A의 총 switch문 수 Bn, B의 총 switch문 수 Bm, 같은 내용의 switch문 수 Bn(Bn<Bm)	3303
	A의 총 switch문 수 Bn, B의 총 switch문 수 Bm, 같은 내용의 switch문 수 Bn/2(Bn<Bm)	3304
	A의 총 switch문 수 Bn, B의 총 switch문 수 Bm, 같은 내용의 switch문 수 0	3305
	A의 총 if문 수 Bn, B의 총 switch문(case 1개) 수 Bm, 같은 내용의 조건문 수 Bn(Bn<Bm)	3306
	A의 총 if문 수 Bn, B의 총 switch문(case 1개) 수 Bm, 같은 내용의 조건문 수 Bn/2(Bn<Bm)	3307
	A의 총 if문 수 Bn, B의 총 switch문(case 1개) 수 Bm, 같은 내용의 조건문 수 0	3308

* Representative Values (4/4)

Category	Values	Number
Check Loop	A의 총 whlie문 수 Ln, B의 총 while문 수 Lm, 같은 내용의 반복문 수 Ln(Ln<Lm)	3400
	A의 총 whlie문 수 Ln, B의 총 while문 수 Lm, 같은 내용의 반복문 수 Ln/2(Ln<Lm)	3401
	A의 총 whlie문 수 Ln, B의 총 while문 수 Lm, 같은 내용의 반복문 수 0	3402
	A의 총 for문 수 Ln, B의 총 for문 수 Lm, 같은 내용의 반복문 수 Ln(Ln<Lm)	3403
	A의 총 for문 수 Ln, B의 총 for문 수 Lm, 같은 내용의 반복문 수 Ln/2(Ln<Lm)	3404
	A의 총 for문 수 Ln, B의 총 for문 수 Lm, 같은 내용의 반복문 수 0	3405
	A의 총 while문 수 Ln, B의 총 for문 수 Lm, 같은 내용의 반복문 수 Ln(Ln<Lm)	3406
	A의 총 while문 수 Ln, B의 총 for문 수 Lm, 같은 내용의 반복문 수 Ln/2(Ln<Lm)	3407
Select pair 수	0	4000
	1	4001
	n	4002
Display pair 수	n	4100
Total Test Case 수	17,006,112	

* Error Constraints 적용

Category	Values
Open File 확장자	.etc
파일 이름 중복	이름이 중복되는 파일이 존재할 때
Filter Range Value	0 미만
	100초과
	Malformed
Select pair 수	0
	n
Error Constraints 적용 후 Test Case 수	17,006,112 -> 354,301

* Single Constraints 적용 (1/2)

Category	Values
Select File 수	0
	1
Clear	파일 목록에 파일이 존재할 때
	파일 목록에 파일이 존재하지 않을 때
Filter Range Value	50
	100
Check Raw Text	A의 단어 수 T_n , B의 단어 수 T_m , 같은 단어 수 T_n
	A의 단어 수 T_n , B의 단어 수 T_m , 같은 단어 수 $T_n/2$
	A의 단어 수 T_n , B의 단어 수 T_m , 같은 단어 수 0
Check Comment	A의 총 주석('//') 수 C_n , B의 총 주석('//') 수 C_m , 같은 내용의 주석 수 $C_n(C_n < C_m)$
	A의 총 주석('//') 수 C_n , B의 총 주석('//') 수 C_m , 같은 내용의 주석 수 $C_n/2(C_n < C_m)$
	A의 총 주석('//') 수 C_n , B의 총 주석('//') 수 C_m , 같은 내용의 주석 수 0
	A의 총 주석('/* */) 수 C_n , B의 총 주석('/* */) 수 C_m , 같은 내용의 주석 수 $C_n(C_n < C_m)$
	A의 총 주석('/* */) 수 C_n , B의 총 주석('/* */) 수 C_m , 같은 내용의 주석 수 $C_n/2(C_n < C_m)$
	A의 총 주석('/* */) 수 C_n , B의 총 주석('/* */) 수 C_m , 같은 내용의 주석 수 0
	A의 총 주석('//') 수 C_n , B의 총 주석('/* */) 수 C_m , 같은 내용의 주석 수 $C_n(C_n < C_m)$
	A의 총 주석('//') 수 C_n , B의 총 주석('/* */) 수 C_m , 같은 내용의 주석 수 $C_n/2(C_n < C_m)$
	A의 총 주석('//') 수 C_n , B의 총 주석('/* */) 수 C_m , 같은 내용의 주석 수 0

* Single Constraints 적용 (2/2)

Category	Values
Check Condition	A의 총 if문 수 B_n , B의 총 if문 수 B_m , 같은 내용의 if문 수 $B_n(B_n < B_m)$
	A의 총 if문 수 B_n , B의 총 if문 수 B_m , 같은 내용의 if문 수 $B_n/2(B_n < B_m)$
	A의 총 if문 수 B_n , B의 총 if문 수 B_m , 같은 내용의 if문 수 0
	A의 총 switch문 수 B_n , B의 총 switch문 수 B_m , 같은 내용의 switch문 수 $B_n(B_n < B_m)$
	A의 총 switch문 수 B_n , B의 총 switch문 수 B_m , 같은 내용의 switch문 수 $B_n/2(B_n < B_m)$
	A의 총 switch문 수 B_n , B의 총 switch문 수 B_m , 같은 내용의 switch문 수 0
	A의 총 if문 수 B_n , B의 총 switch문(case 1개) 수 B_m , 같은 내용의 조건문 수 $B_n(B_n < B_m)$
	A의 총 if문 수 B_n , B의 총 switch문(case 1개) 수 B_m , 같은 내용의 조건문 수 $B_n/2(B_n < B_m)$
	A의 총 if문 수 B_n , B의 총 switch문(case 1개) 수 B_m , 같은 내용의 조건문 수 0
Check Loop	A의 총 whlie문 수 L_n , B의 총 while문 수 L_m , 같은 내용의 반복문 수 $L_n(L_n < L_m)$
	A의 총 whlie문 수 L_n , B의 총 while문 수 L_m , 같은 내용의 반복문 수 $L_n/2(L_n < L_m)$
	A의 총 whlie문 수 L_n , B의 총 while문 수 L_m , 같은 내용의 반복문 수 0
	A의 총 for문 수 L_n , B의 총 for문 수 L_m , 같은 내용의 반복문 수 $L_n(L_n < L_m)$
	A의 총 for문 수 L_n , B의 총 for문 수 L_m , 같은 내용의 반복문 수 $L_n/2(L_n < L_m)$
	A의 총 for문 수 L_n , B의 총 for문 수 L_m , 같은 내용의 반복문 수 0
	A의 총 while문 수 L_n , B의 총 for문 수 L_m , 같은 내용의 반복문 수 $L_n(L_n < L_m)$
	A의 총 while문 수 L_n , B의 총 for문 수 L_m , 같은 내용의 반복문 수 $L_n/2(L_n < L_m)$
	A의 총 while문 수 L_n , B의 총 for문 수 L_m , 같은 내용의 반복문 수 0
Select Pair 수	1
Single Constraints 적용 후 Test Case 수	354,301 -> 240

* Property Constraints 적용

Category	Values	Property
Open File 수	0	[Property OZ]
	1	[Property OO]
	n	[Property ON]
파일 이름 중복	이름이 중복되는 파일이 존재할 때	if[OO] if[ON]
	이름이 중복되는 파일이 존재하지 않을 때	if[OO] if[ON]
Select File 수	0	[Property SZ]
	1	if[OO] [Property SO]
	n	if[ON] [Property SN]
Clear	파일 목록에 파일이 존재할 때	if[OO] if[ON]
	파일 목록에 파일이 존재하지 않을 때	if[OZ]
Delete button	선택된 파일이 없을 때	if[S0] if[SN]
Compare button	Compare를 수행한다.	if[SN]
Filter Range Value	0	[Property RVZ]
	50	[Property RVF]
	100	[Property RVH]
Display pair 수	n	if[RVZ] if[RVF] if[RVH]
Property Constraints 적용 후 Test Case 수	240 -> 62	

* Test Code Sample (반복문 비교)

```
int main(void){
    int a = 0;
    int b = 1;
    int c = 1;
    int d = 1;
    while (a < 10){
        a = a++;
        printf("test a\n");
    }
    while (b < 30){
        b = b + a;
        printf("test b\n");
    }
    c = 2;
    while (c < 10){
        printf("c = %d\n", c);
        c++;
    }
    d = 10;
    while (d > 1){
        printf("d = %d\n", d);
        d--;
    }
}
```

```
int main(void){
    int a = 0;
    int b = 1;
    int c = 1;
    int d = 1;
    printf("check my name");
    for (; a < 10; a++){
        printf("test a\n");
    }
    printf("kyakya");
    for (; b < 30; b = b + a){
        printf("test b\n");
    }
    printf("zzzzz");
    for (c = 2; c < 10; c++){
        printf("c = %d\n", c);
    }
    printf("kyoyko");
    for (d = 10; d > 1; d--){
        printf("d = %d\n", d);
    }
    printf("hoo");
}
```

While문을 For문으로 변경했을 때 Test Code
검사 시에 오차율 ±10% 적용

* Testing Result (1/3)

Test Case	Result
test-8:1001	Failed
test-9:1000.1100	Passed
test-10:1000.1101	Passed
test-11:1000.1102	Passed
test-12:1000.1101.1301	Passed
test-13:1000.1102.1302	Failed
test-14:1000.1101.1400	Passed
test-15:1000.1102.1400	Passed
test-19:2000	Failed
test-20:2001	Passed
test-21:2002	Passed
test-22:2003	Passed
test-23:2004	Failed
test-24:2005	Failed
test-25:1000.1102.1201.1302.1600.2001.3000.4001	Passed
test-26:1000.1102.1201.1302.1600.2001.3001.4001	Passed
test-27:1000.1102.1201.1302.1600.2001.3002.4001	Failed
test-28:1000.1102.1201.1302.1600.2001.3200.4001	Passed
test-29:1000.1102.1201.1302.1600.2001.3201.4001	Failed
test-30:1000.1102.1201.1302.1600.2001.3202.4001	Failed
test-31:1000.1102.1201.1302.1600.2001.3203.4001	Failed
test-32:1000.1102.1201.1302.1600.2001.3204.4001	Failed
test-33:1000.1102.1201.1302.1600.2001.3205.4001	Failed

* Testing Result (2/3)

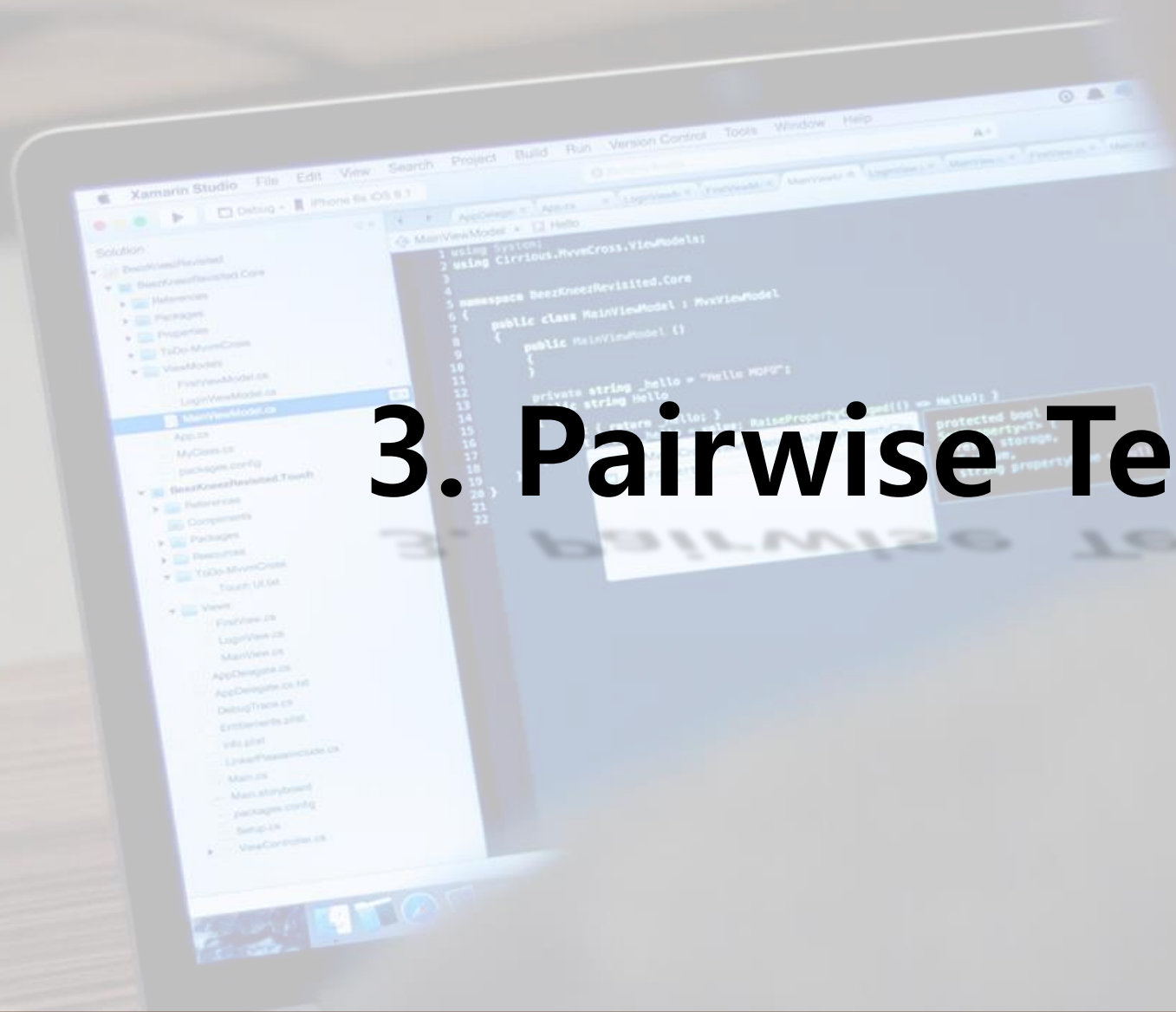
Test Case	Result
test-36:1000.1102.1201.1302.1600.2001.3300.4001	Failed
test-37:1000.1102.1201.1302.1600.2001.3301.4001	Failed
test-38:1000.1102.1201.1302.1600.2001.3302.4001	Failed
test-39:1000.1102.1201.1302.1600.2001.3303.4001	Failed
test-40:1000.1102.1201.1302.1600.2001.3304.4001	Failed
test-41:1000.1102.1201.1302.1600.2001.3305.4001	Failed
test-44:1000.1102.1201.1302.1600.2001.3400.4001	Failed
test-45:1000.1102.1201.1302.1600.2001.3401.4001	Failed
test-46:1000.1102.1201.1302.1600.2001.3402.4001	Failed
test-47:1000.1102.1201.1302.1600.2001.3403.4001	Failed
test-48:1000.1102.1201.1302.1600.2001.3404.4001	Failed
test-49:1000.1102.1201.1302.1600.2001.3405.4001	Failed
test-52:1000.1102.1201.1302.1600.2001.3100.4001	Failed
test-53:1000.1102.1201.1302.1600.2001.3101.4001	Failed
test-54:1000.1102.1201.1302.1600.2001.3102.4001	Failed
test-55:1000.1102.1201.1302.1600.2001.3206.4001	Failed
test-56:1000.1102.1201.1302.1600.2001.3207.4001	Failed

* Testing Result (3/3)

Test Case	Result
test-57:1000.1002.1201.1032.1600.2001.3208.4001	Failed
test-58:1000.1102.1201.1302.1600.2001.3306.4001	Failed
test-59:1000.1102.1201.1302.1600.2001.3307.4001	Failed
test-60:1000.1102.1201.1302.1600.2001.3308.4001	Failed
test-61:1000.1102.1201.1302.1600.2001.3406.4001	Failed
test-62:1000.1102.1201.1302.1600.2001.3407.4001	Failed
test-63:1000.1102.1201.1302.1600.2001.3408.4001	Failed
test-64:1000.1101.1200	Failed
test-65:1000.1102.1200	Failed
test-66:1000.1101.1201	Passed
test-67:1000.1102.1201	Passed
test-70:1102.1300	Passed
test-71:1102.1301	Failed
test-73:1300.1500	Passed
test-74:1301.1500	Passed
test-75:1302.1500	Passed
test-76:1300.1600	Failed
test-77:1301.1600	Failed
test-78:1302.1600	Failed
test-99:1000.1102.1201.1302.1600.2001.3000.4000	Failed
test-100:1000.1102.1201.1302.1600.2001.3000.4002	Failed
test-101:1000.1102.1201.1302.1600.2001.3000.4100	Passed

19/62 = 31% Pass

3. Pairwise Test



* Test Case 생성 (1/2)

PictMaster

v5.7.3 December 10, 2013

Item number		Item name		Date	
Sub-item number		Sub-item name		Creator	

Build

Edit

Settings

Parameters

value hierarchy

© IWATSU System & Software Co., Ltd. Licensed under the Open Software License version 2.1

open file extension	c, etc
open file number	0, 1, n
filename samesame	sse, ssne
select file number	0,1,n
clear	fle, flne
delete button	delete execute
compare button	compare execute
configure	lessthan0, 0, 50, 100, 100over, malformed
select pair number	0, 1, n
display pair number	n

PICT Tool을 사용 Pairwise Test Case 생성

* Test Case 생성 (2/2)

	A	B	C	D	G	H	I	J	K
1	open file e	open file n	filename s	select file	compare b	configure	select pair	display pair	number
2	etc	n	ssne		compar	100ove	n	n	
3	c	1	sse	n	compare e	100	1	n	
4	c	0	ssne		compare e	50	0	n	
5	etc	0	ssne		compare e	100	1	n	
6	etc	n	sse	n	compare e	0	0	n	
7	c	n	sse		compare e	100	n	n	
8	etc	0	ssne	n	compare e	malformec	n	n	
9	etc	1	sse		compare e	lessthan0	n	n	
10	c	1	ssne	n	compare e	malformec	0	n	
11	c	1	sse		compare e	50	n	n	
12	c	0	sse		compare e	lessthan0	0	n	
13	c	0	sse		compare e	100over	n	n	
14	etc	n	ssne		compare e	50	1	n	
15	etc	0	sse	n	compare e	50	0	n	
16	etc	0	sse		compare e	100	n	n	
17	c	0	sse		compare e	malformec	1	n	
18	etc	1	ssne		compare e	lessthan0	0	n	
19	etc	0	ssne	n	compare e	100over	1	n	
20	etc	1	sse		compare e	malformec	n	n	
21	c	1	ssne		compare e	50	1	n	
22	etc	1	sse		compare e	100over	1	n	
23	c	n	ssne		compare e	lessthan0	1	n	



	A	B	C	D	G	H	I	J	K
	open file e	open file n	filename s	select file	compare b	configure	select pair	display pair	number
	etc	n	ssne		compar	100ove	n	n	
	c	0	ssne		compare e	50	0	n	
	c	1	ssne		compare e	0	n	n	

Error Constraints를 이용 Test Case 범위 축소

* Testing Result (1/3)

Test Case	Result
test-102:1000.1102.1201.1300.1600.2002.3000.4001	Failed
test-103:1000.1102.1201.1300.1600.2002.3001.4001	Failed
test-104:1000.1102.1201.1300.1600.2002.3002.4001	Failed
test-105:1000.1102.1201.1300.1600.2002.3100.4001	Failed
test-106:1000.1102.1201.1300.1600.2002.3101.4001	Failed
test-107:1000.1102.1201.1300.1600.2002.3102.4001	Failed
test-108:1000.1102.1201.1300.1600.2002.3200.4001	Failed
test-109:1000.1102.1201.1300.1600.2002.3201.4001	Failed
test-110:1000.1102.1201.1300.1600.2002.3202.4001	Failed
test-111:1000.1102.1201.1300.1600.2002.3203.4001	Failed
test-112:1000.1102.1201.1300.1600.2002.3204.4001	Failed
test-113:1000.1102.1201.1300.1600.2002.3205.4001	Failed
test-114:1000.1102.1201.1300.1600.2002.3206.4001	Failed
test-115:1000.1102.1201.1300.1600.2002.3207.4001	Failed
test-116:1000.1102.1201.1300.1600.2002.3208.4001	Failed
test-117:1000.1102.1201.1300.1600.2002.3300.4001	Failed
test-118:1000.1102.1201.1300.1600.2002.3301.4001	Failed
test-119:1000.1102.1201.1300.1600.2002.3302.4001	Failed
test-120:1000.1102.1201.1300.1600.2002.3303.4001	Failed
test-121:1000.1102.1201.1300.1600.2002.3304.4001	Failed
test-122:1000.1102.1201.1300.1600.2002.3305.4001	Failed
test-123:1000.1102.1201.1300.1600.2002.3306.4001	Failed
test-124:1000.1102.1201.1300.1600.2002.3307.4001	Failed

* Testing Result (2/3)

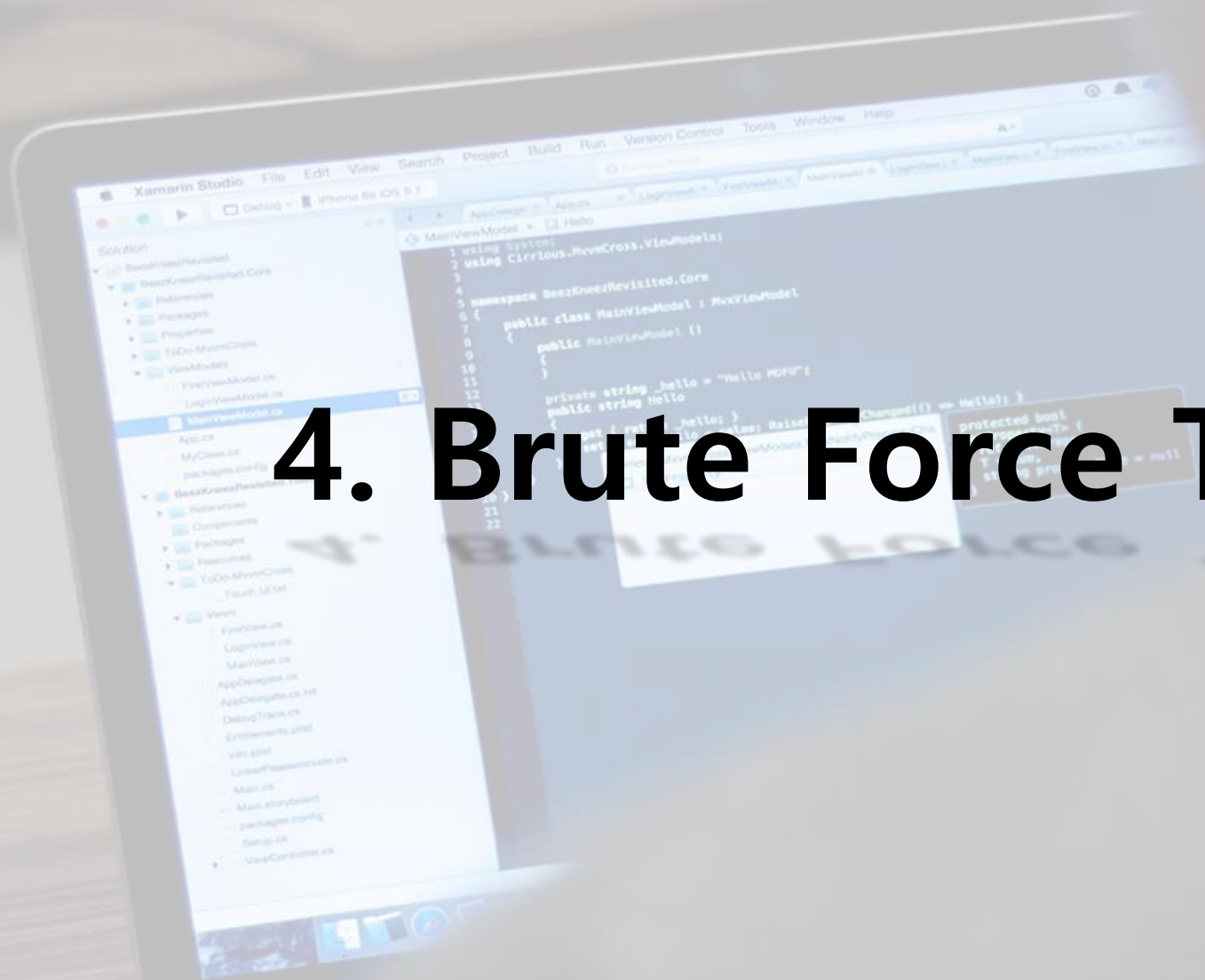
Test Case	Result
test-125:1000.1102.1201.1300.1600.2002.3308.4001	Failed
test-126:1000.1102.1201.1300.1600.2002.3400.4001	Failed
test-127:1000.1102.1201.1300.1600.2002.3401.4001	Failed
test-128:1000.1102.1201.1300.1600.2002.3402.4001	Failed
test-129:1000.1102.1201.1300.1600.2002.3403.4001	Failed
test-130:1000.1102.1201.1300.1600.2002.3404.4001	Failed
test-131:1000.1102.1201.1300.1600.2002.3405.4001	Failed
test-132:1000.1102.1201.1300.1600.2002.3406.4001	Failed
test-133:1000.1102.1201.1300.1600.2002.3407.4001	Failed
test-134:1000.1102.1201.1300.1600.2002.3408.4001	Failed
test-136:1000.1102.1201.1302.1600.2001.3000.4001	Passed
test-137:1000.1102.1201.1302.1600.2001.3001.4001	Passed
test-138:1000.1102.1201.1302.1600.2001.3002.4001	Failed
test-139:1000.1102.1201.1302.1600.2001.3200.4001	Passed
test-140:1000.1102.1201.1302.1600.2001.3201.4001	Failed
test-141:1000.1102.1201.1302.1600.2001.3202.4001	Failed
test-142:1000.1102.1201.1302.1600.2001.3203.4001	Failed
test-143:1000.1102.1201.1302.1600.2001.3204.4001	Failed
test-144:1000.1102.1201.1302.1600.2001.3205.4001	Failed
test-145:1000.1102.1201.1302.1600.2001.3206.4001	Failed
test-146:1000.1102.1201.1302.1600.2001.3207.4001	Failed
test-147:1000.1002.1201.1032.1600.2001.3208.4001	Failed
test-150:1000.1102.1201.1302.1600.2001.3100.4001	Failed

* Testing Result (3/3)

Test Case	Result
test-151:1000.1102.1201.1302.1600.2001.3101.4001	Failed
test-152:1000.1102.1201.1302.1600.2001.3102.4001	Failed
test-153:1000.1102.1201.1302.1600.2001.3300.4001	Failed
test-154:1000.1102.1201.1302.1600.2001.3301.4001	Failed
test-155:1000.1102.1201.1302.1600.2001.3302.4001	Failed
test-158:1000.1102.1201.1302.1600.2001.3305.4001	Failed
test-159:1000.1102.1201.1302.1600.2001.3306.4001	Failed
test-160:1000.1102.1201.1302.1600.2001.3307.4001	Failed
test-161:1000.1102.1201.1302.1600.2001.3308.4001	Failed
test-167:1000.1102.1201.1302.1600.2001.3303.4001	Failed
test-168:1000.1102.1201.1302.1600.2001.3304.4001	Failed
test-175:1000.1102.1201.1302.1600.2001.3400.4001	Failed
test-176:1000.1102.1201.1302.1600.2001.3401.4001	Failed
test-177:1000.1102.1201.1302.1600.2001.3402.4001	Failed
test-178:1000.1102.1201.1302.1600.2001.3403.4001	Failed
test-179:1000.1102.1201.1302.1600.2001.3404.4001	Failed
test-180:1000.1102.1201.1302.1600.2001.3405.4001	Failed
test-181:1000.1102.1201.1302.1600.2001.3406.4001	Failed
test-182:1000.1102.1201.1302.1600.2001.3407.4001	Failed
test-183:1000.1102.1201.1302.1600.2001.3408.4001	Failed
test-151:1000.1102.1201.1302.1600.2001.3101.4001	Failed
test-152:1000.1102.1201.1302.1600.2001.3102.4001	Failed
test-153:1000.1102.1201.1302.1600.2001.3300.4001	Failed

3/66 = 5% Pass

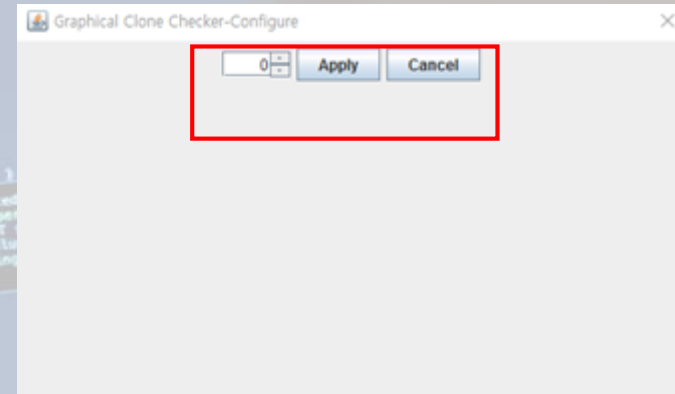
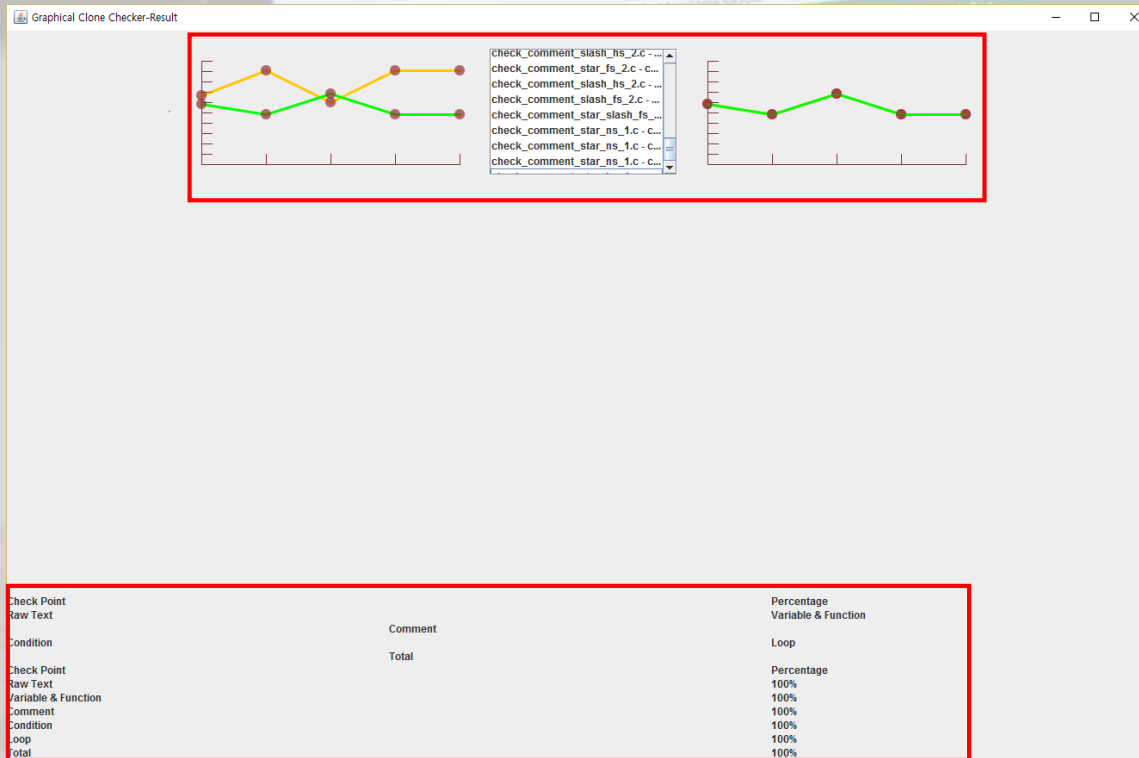
4. Brute Force Test



* Test Case Sample

6. User Interface Requirements

- 누구나 쉽게 사용할 수 있도록 최대한 단순한 UI를 제공한다.



Brute Force Test에서는 Non Functional Requirement을 Test

* Testing Result

Number	Test Case	Result
1	test-79:Filter range에 0-100 이외의 값을 작성 후 OK 버튼 입력Filter range에 숫자가 아닌 다른 값을 작성 후 OK 버튼 입력한다.	Failed
2	test-80:Filter range에 0-100 값을 작성 후 Cancel 버튼 입력	Failed
3	test-81:Sample Source로 Compare 한다.	Failed
4	test-82:누구나 쉽게 사용 할 수 있는 단순한 UI인지 확인한다.	Failed
5	test-83:조건문, 주석이 없고 같은 반복문만 존재하는 소스코드 2개를 Compare 한다.	Failed
6	test-84:반복문, 주석이 없고 같은 조건문만 존재하는 소스코드 2개를 Compare 한다.	Failed
7	test-85:조건문, 반복문이 없고 같은 주석만 존재하는 소스코드 2개를 Compare 한다.	Failed
8	test-86:Filter range 값을 100으로 설정하고 소스코드가 일치하는 파일 2개 compare	Failed
9	test-87:Filter range 값을 0으로 설정하고 소스코드가 없는 파일 2개를 Compare 한다.	Failed
10	test-88:Compare를 10번 실행시킨다.	Failed
11	test-89:Filter range에 int 범위는 벗어나는 값(4294967299)으로 설정 후 OK 버튼 입력한다.	Failed
12	test-90:Compare창과 Configure창 둘 다 연 상태에서 Exit 버튼 클릭	Passed
13	test-91:Compare 결과창, Configure 창의 왼쪽 위 모서리를 왼쪽 화면 바깥으로 보내고 꺾다가 다시 창을 연다.	Failed
14	test-92:File Open, Clear 버튼 입력, 이 과정을 10번 반복한다.	Passed
15	test-93:10개의 파일을 열고, 1개씩 delete 해서 10개의 파일 모두 지운다.	Passed
16	test-94:조건문, 주석, 반복문이 없고 같은 단어만 존재하는 소스코드 2개를 Compare 한다.	Failed
17	test-95:파일 목록에 파일이 아닌 흰 배경을 클릭	Failed
18	test-96:Filter range에 valid 값을 입력하고 Configure 창을 강제종료(Alt+F4)한다.	Failed
19	test-97:Configure 값을 변경하지 않고 파일을 n개 열고 Compare를 한다.	Failed
20	test-98:파일을 100개 열어본다.	Passed

4/20 = 20% Pass

* Failed Case Report

Number	Report
1	50a 로 Filter range값을 적용할 경우 Filter range가 50으로 적용이 된다.
2	Cancel 버튼을 눌렀는데 그 값으로 Filter range가 적용이 된다.
3	시간을 재봤는데 7분이 나온다. 교수님께서 정해준 5분 이내로 실행이 안된다.
4	파일명이 길어지면 결과 창에서 확인 할 수가 없다. 결과로 나온 파일 목록의 개수가 많아지면 확인할 수가 없다. 결과창을 확대하면 그에 따른 UI의 변화가 없다.
5	조건문, 주석이 없는데 Percentage가 존재한다. 없다면 0% 또는 100%가 나와야 한다고 생각을 한다.
6	조건문, 반복문이 없는데 Percentage가 존재한다. 없다면 0% 또는 100%가 나와야 한다고 생각을 한다.
7	반복문, 주석이 없는데 Percentage가 존재한다. 없다면 0% 또는 100%가 나와야 한다고 생각을 한다.
8	조건문, 주석, 반복문이 없는데 Percentage가 100%로 나온다. 또 Raw Text가 80% 가 넘을경우 조건문,주석,반복문이 Raw Text Percentage랑 똑같이 나온다.
9	복제된것으로 의심되는 소스가 없습니다. 라는 창이 뜬다. 문서에는 100이상인 것을 보여준다고 했는데 결과창이 안뜬다.
10	복제된것으로 의심되는 소스가 없습니다. 라는 창이 뜬다. 문서에는 Filter range 값 이상을 보여준다고 했는데 결과창이 안뜬다.
11	결과창 UI가 계속 변하고 나중에는 확인 할 수가 없다.
13	4294967299 값을 입력하면 Filter range가 3으로 설정된다.
16	창이 보이지 않는다. 창을 보려면 작업표시줄에 있는 프로그램을 우클릭 한다음에 이동을 누르고 키보드 방향키로 조작을 해야지 창이 모니터 화면 안에 보인다.
17	아무것도 선택이 되면 안되는데 맨 아래에 있는 파일이 선택이 된다.
18	강제종료하기 전에 입력한 값이 적용이 된다.
19	Configure 값이 80으로 기본 설정되어 있는데 total이 80 아래인 pair들이 결과창에 나온다.

5. Overall



* 1st System Test Result

Category-partition Test

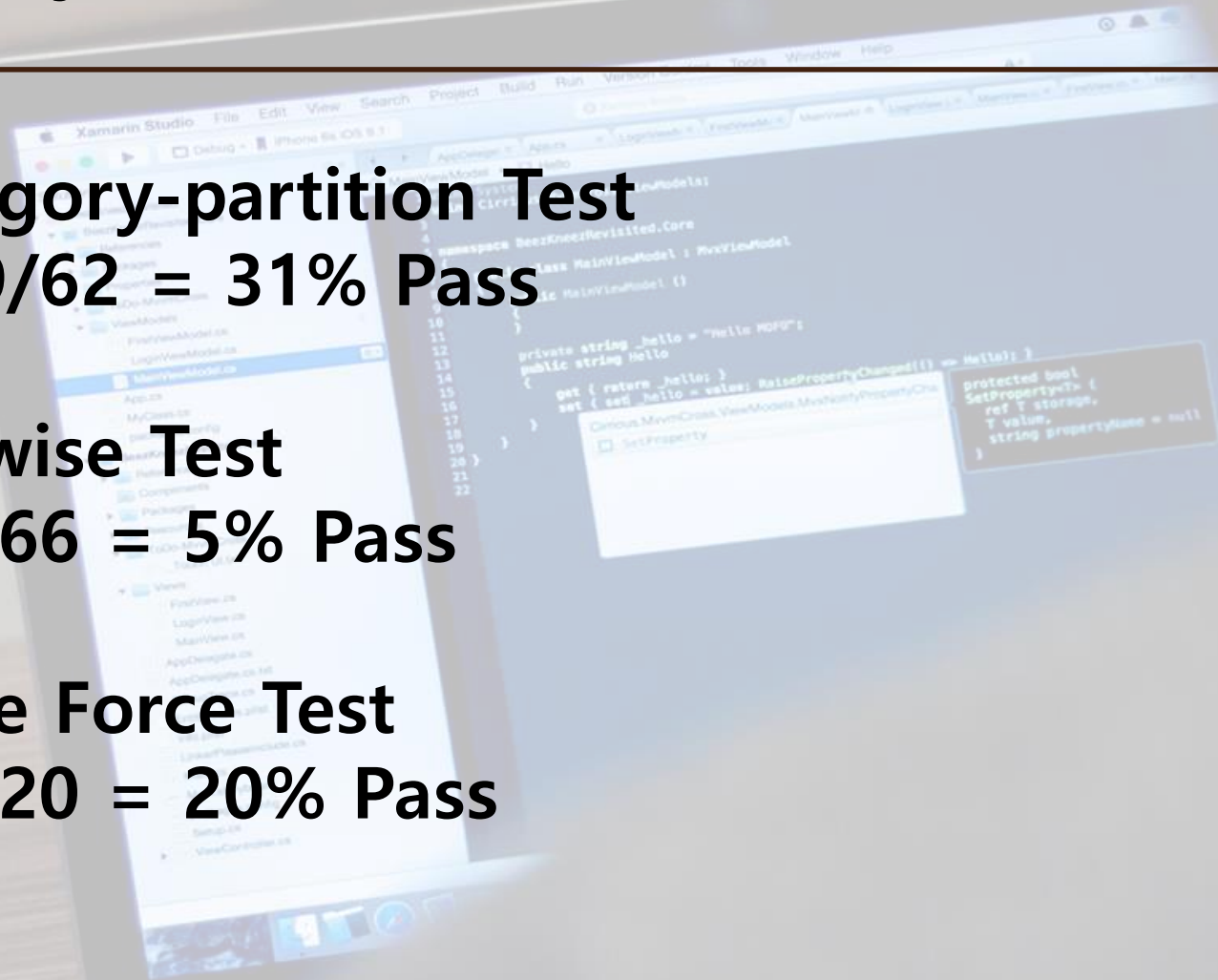
- 19/62 = 31% Pass

Pairwise Test

- 3/66 = 5% Pass

Brute Force Test

- 4/20 = 20% Pass



* Summary

- 문서들의 단계마다 일치하지 않는 부분들이 있다.
- 실제 구현된 코드와 문서간에 차이점이 존재한다.
- 사용한 알고리즘이 단순히 문자간의 거리를 측정하는 데 그치므로 반복문이나 조건문과 같이 코드의 구조를 검사해야 되는 Clone Checker라는 소프트웨어에 적합하지 않다.
- Compare를 2번 하면 GUI가 깨지는 것과 같이 예상하지 못한 Bug들이 존재한다.
- Performance와 User Interface와 같은 Non Functional Requirement를 만족시키지 못하였다.

* In CTIP – Testlink (1/2)

The screenshot displays the Testlink web interface. On the left, the 'Filters' panel is active, showing search criteria for Test Case ID (test-), Test Suite (G.C.C BrutalFoce Test suite), and other attributes. Below the filters is a tree view of test plans and suites. The main area on the right shows 'Test Results on Build version 1.0'. It includes a table for 'Last execution (any build)' and 'Last execution (current build)'. The current build execution is shown as 'Failed' on 05/25/2016. Below the table, there is a 'Summary' section with a message: 'Filter range에 숫자가 아닌 다른 값을 작성 후 OK 버튼 입력한다.' (Enter a value other than a number in the filter range and click the OK button). There are also buttons for 'Save execution' and 'Save and move to next'.

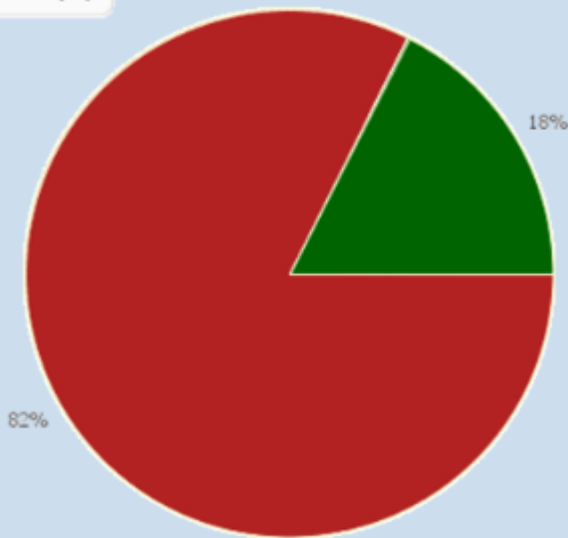
Date	Build	Tested by	Status	Exec (min)	Version	BUG mana
05/25/2016 07:38:39	version 1.0	admin	Failed		1	

Test Case를 Testlink를 이용해 관리

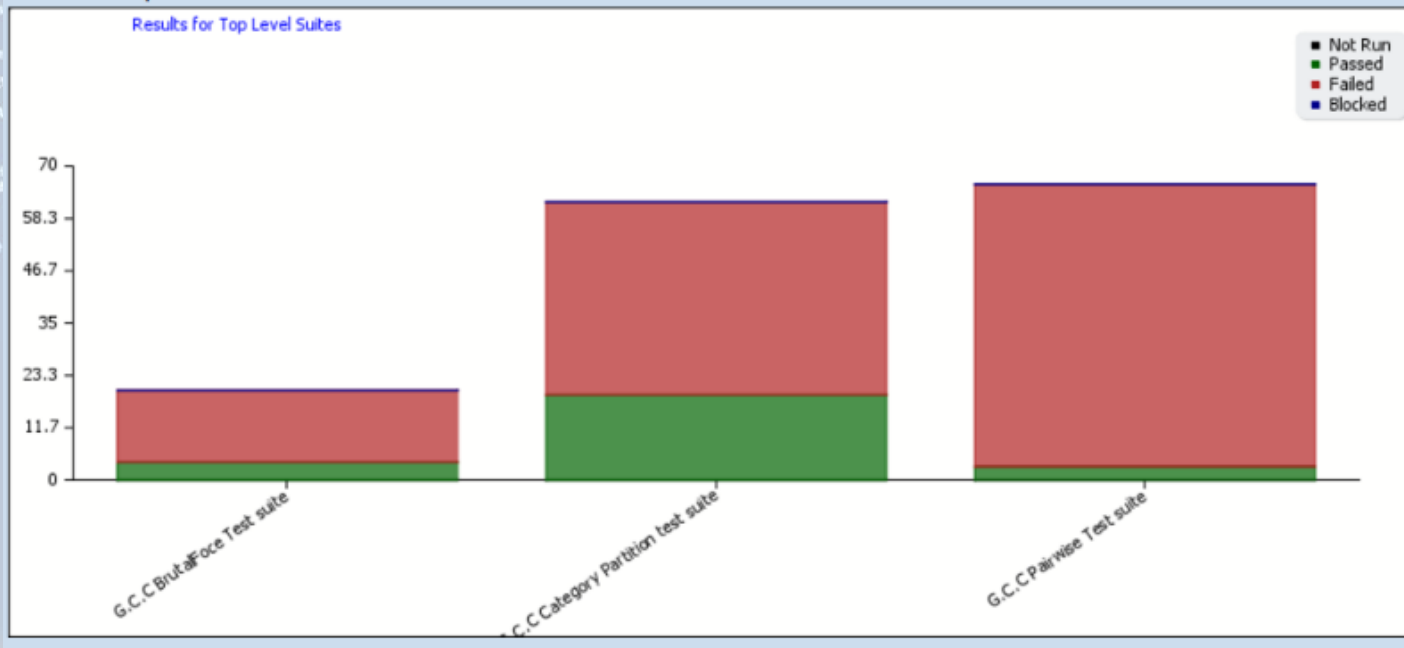
* In CTIP – Testlink (2/2)

Overall Metrics

Failed (122)
Passed (26)



Results for Top Level Suites



Test 결과를 Chart로 확인 가능

* In CTIP - Redmine

일감

▼ 검색조건
 상태 진행중 ▼ 검색조건 추가

▶ 옵션

✓ 적용 🔄 지우기 📄 저장

#	▼	유형	상태	우선순위	제목	담당자	변경
<input type="checkbox"/>	72	결함	신규	보통	[BrutalForce Test] Configure값을 변경하지 않은 상태에서 파일을 5개 열고 Compare 버튼을 누른다.		2016/05/25 17:52
<input type="checkbox"/>	71	결함	신규	보통	[BrutalForce Test] Filter range에 0-100 값을 입력하고 Configure 창을 강제 종료(Alt+F4)할 경우		2016/05/25 17:51
<input type="checkbox"/>	70	결함	신규	보통	[BrutalForce Test] 파일을 2개 열고 파일 목록에 파일이 아닌 회 배경을 클릭한다.		2016/05/25 17:51
<input type="checkbox"/>	68	결함	신규	보통	[BrutalForce Test] Compare 결과창과 Configure창의 왼쪽 위 모서리를 화면 바깥으로 보내고 꺾다가 다시 창을 킨다.		2016/05/25 17:48
<input type="checkbox"/>	67	결함	신규	보통	[BrutalForce Test] Filter range에 int 범위는 벗어나는 값(4294967299)으로 설정 후 OK 버튼 입력한다.		2016/05/25 17:47
<input type="checkbox"/>	66	결함	신규	보통	[BrutalForce Test] 같은 소스코드에 대해서 Compare 버튼을 10번 입력한다.		2016/05/25 17:47
<input type="checkbox"/>	65	결함	신규	보통	[BrutalForce Test] Filter range 값 0으로 설정하고 소스코드가 없는 파일 2개를 Compare한다.		2016/05/25 17:47
<input type="checkbox"/>	64	결함	신규	보통	[BrutalForce Test] Filter range값 100으로 조정하고 소스가 완전히 일치하는 파일 2개 Compare 한다.		2016/05/25 17:47
<input type="checkbox"/>	63	결함	신규	보통	[BrutalForce Test] 조건문, 주석, 반복문이 없고 같은 단어만 존재하는 소스코드 2개를 Compare한다.		2016/05/25 17:58
<input type="checkbox"/>	62	결함	신규	보통	[BrutalForce Test] 조건문, 반복문이 없고 같은 조건문만 존재하는 소스코드 2개를 Compare한다.		2016/05/25 17:58
<input type="checkbox"/>	61	결함	신규	보통	[BrutalForce Test] 반복문, 주석이 없고 같은 조건문만 존재하는 소스코드 2개를 Compare한다.		2016/05/25 17:46
<input type="checkbox"/>	60	결함	신규	보통	[BrutalForce Test] 조건문, 주석이 없고 같은 반복문만 존재하는 소스코드 2개를 Compare 한다.		2016/05/25 17:45
<input type="checkbox"/>	59	결함	신규	보통	[BrutalForce Test] 누구나 쉽게 사용할 수 있는 단순한 UI 제공하는지 확인한다.		2016/05/25 17:44
<input type="checkbox"/>	58	결함	신규	보통	[BrutalForce Test] Sample Source로 Compare를 한다.		2016/05/25 17:44
<input type="checkbox"/>	57	결함	신규	보통	[BrutalForce Test] Filter range에 0-100 값을 작성 후 Cancel 버튼 입력		2016/05/25 17:44
<input type="checkbox"/>	56	결함	신규	보통	[BrutalForce Test] Filter range에 숫자가 아닌 다른 값을 작성 후 OK 버튼 입력한다.		2016/05/25 17:43
<input type="checkbox"/>	55	결함	신규	보통	[Category Partition Test] 1000.1102.1201.1302.1600.2001.3000.4002		2016/05/25 16:28
<input type="checkbox"/>	54	결함	신규	보통	[Category Partition Test] 1000.1102.1201.1302.1600.2001.3000.4000		2016/05/25 16:27
<input type="checkbox"/>	53	결함	신규	보통	[Category Partition Test] 1302.1600		2016/05/25 16:21
<input type="checkbox"/>	52	결함	신규	보통	[Category Partition Test] 1301.1600		2016/05/25 16:21
<input type="checkbox"/>	51	결함	신규	보통	[Category Partition Test] 1300.1600		2016/05/25 16:21
<input type="checkbox"/>	50	결함	신규	보통	[Category Partition Test] 1102.1301		2016/05/25 16:20
<input type="checkbox"/>	49	결함	신규	보통	[Category Partition Test] 1000.1102.1200		2016/05/25 16:20
<input type="checkbox"/>	48	결함	신규	보통	[Category Partition Test] 1000.1101.1200		2016/05/25 16:20
<input type="checkbox"/>	47	결함	신규	보통	[Category Partition Test] 1000.1102.1201.1302.1600.2001.3408.4001		2016/05/25 16:19

Failed된 Test Case를 Redmine에 일감을 등록해 E-Mail로 소모팀에게 알림



Thank You!