

# System Test Report for Team 5

## 2nd

Project Team  
Software Verification Team 3

Latest update on:  
2019-06-06

---

### Team Information

컴퓨터공학과	201311276	박형민
컴퓨터공학과	201311287	엄현식
컴퓨터공학과	201311318	최정헌
컴퓨터공학과	201311320	한예훈

# Table of Contents

<b>1</b>	<b>Specification Review</b>	<b>3</b>
1.1	Rejected Proposal Report	3
1.1.1	Stage 1000 Planning	3
1.1.2	Stage 2030 Analysis	3
1.1.3	Stage 2040 Design	9
1.2	Specification Review (2nd)	11
1.2.1	Stage 2030 Analysis	11
1.2.2	Stage 2040 Design	12
<b>2</b>	<b>System Testing</b>	<b>13</b>
2.1	Category-Partition Testing Report	13
2.1.1	Categorize	13
2.1.2	Description	18
2.1.3	Property & Constraints	23
2.1.4	Text File for TSL-Generator	27
2.1.5	Test Result	30
2.2	Pairwise Testing Report	32
2.2.1	Test Result	32
2.3	Brute Force Testing Report	33
2.3.1	Testing Result	33
2.3.2	Failed Case Report	34
<b>3</b>	<b>Static Analysis</b>	<b>35</b>
3.1	SonarQube Rule	35
3.1.1	Rule of Findbugs	35
3.1.2	Rule of PMD	35
3.1.3	Rule of CheckStyle	36
3.1.4	Code Coverage	36
3.2	SonarQube Result	36
3.2.1	Dashboard	36
3.2.2	Bug	37
3.2.3	Vulnerability	37
3.2.4	Code Smell	38
3.2.5	Code Coverage	39
3.3	SonarQube Usage	40
3.4	Conclusion	42
<b>4</b>	<b>Overall</b>	<b>43</b>
4.1	System Test Result	43
4.2	Static Analysis Result	43
4.3	Summary	44

# 1 Specification Review

## 1.1 Rejected Proposal Report

### 1.1.1 Stage 1000 Planning

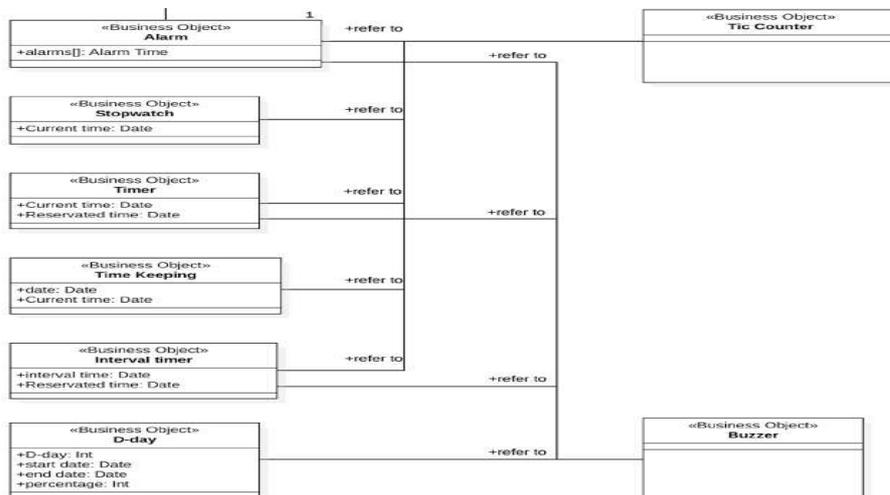
Activity 1001. Define Draft Plan

#### 4. Non-Functional Requirements

- 모드 전환 간에 latency이 없어야한다.  
⇒ Latency의 기준이 명확하지 않다
  - GUI상 디자인이 깔끔해야한다.  
⇒ 기준이 명확하지 않고, 불편한 부분이 있다(ex. 해가 밝을 때가 AM인지 PM인지 명확히 구분할 수가 없다)
- ⇒ SMA : 개발에 기능적으로 필요한 사항이 아닌 개발 도중 지향해야하는 것들로 명시해놓은 것  
⇒ SMA 팀의 Reject 의견에 동의함

### 1.1.2 Stage 2030 Analysis

Activity 2033. Define Domain Model

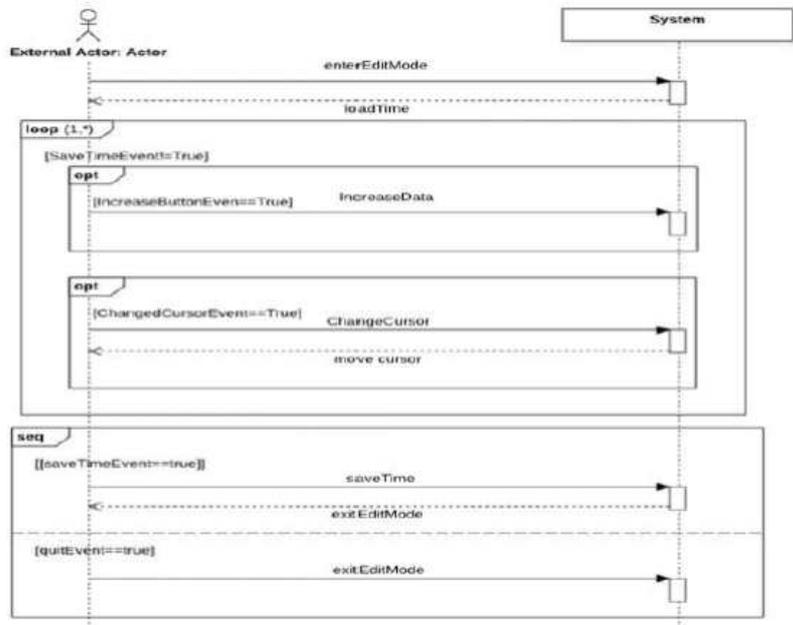


- ⇒ Tic Counter에 해당하는 게 초 단위라면, Alarm과 Interval Timer는 Tic Counter와 Refer to 관계가 아니게 됨

- ⇒ SMA : Alarm과 Interval Timer에서 개별적으로 Tic counter을 가진 게 아니라, 하나의 Tic counter에서 상대적인 시간을 계산하여 반영
- ⇒ SMA 팀의 Reject 의견에 동의함

Activity 2035. Define Domain Model

1. Set Time

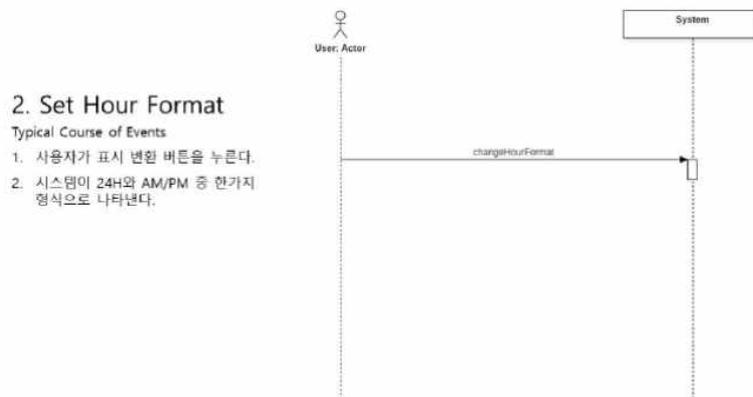


- ⇒ quitEvent == True 일 때 User에서 System으로 exitEditMode을 요청하면, System에서도 changeMode와 같이 화면을 바꿔주는 기능이 분리되어 구현되어야 함
- ⇒ IncreaseData을 System에서 받으면 바뀐 시간에 대해 'loadTime'을 Response 해주어야함
- ⇒ SMA : 시스템의 Response에 해당하는 기능은 개념적인 성격이 크기 때문에 'Interaction Diagram'에서 Don't care 사항임
- ⇒ SMA 팀의 Reject 의견에 동의함

- ⇒ Loop의 조건이 [SaveTimeEvent != True]가 아닌, [SaveTimeEvent != True && quitEvent != True]가 되어야함

⇒ 성공적으로 반영되었음

## 2. Set Hour Format



- ⇒ System은 User에게 신호를 받으면 수행결과에 대한 변경된 화면을 다시 User에게 제공해주어야 하는데, 이러한 부분이 생략되어 있음
- ⇒ 위의 오류는 이후 Activity 2035에 있는 모든 단일 화살표 그림들에 똑같이 발견됨

⇒ **SMA : 시스템의 Response에 해당하는 기능은 개념적인 성격이 크기 때문에 'Interaction Diagram'에서 Don't care 사항임**

⇒ **SMA 팀의 Reject 의견에 동의함**

- ⇒ Activity 1003에서는 '12H/24H'로 형식을 정의했는데, 여기서는 AM/PM 방식이라고 하여 일관성이 깨짐

⇒ 성공적으로 반영되었음

## 7. Reset Timer



- ⇒ System에서 Reset된 Timer의 값을 User에게 보여주는 화살표가 생략되어 있음
- ⇒ 이는 '10. Reset Stopwatch', '18. Reset D-Day', '24. Reset Interval Timer' 에서도 오류임

- ⇒ **SMA : 시스템의 Response에 해당하는 기능은 개념적인 성격이 크기 때문에 'Interaction Diagram'에서 Don't care 사항임**
  - ⇒ **SMA 팀의 Reject 의견에 동의함**

## 11. Set Alarm

### Typical Course of Events

1. 사용자가 수정모드 버튼을 누른다.
2. 사용자가 커서가 맞춰진 알람의 '시' 를 설정한다.
3. 사용자가 버튼을 눌러 커서를 '분'으로 바꾼다.
4. 사용자가 커서가 맞춰진 알람의 '분' 을 설정한다.
5. 사용자가 저장 버튼을 누른다.
6. 시스템이 알람을 입력된 시간으로 변경한다.

### Alternative Courses of Event

(2), (3): 적용버튼을 눌러 해당 알람 시간을 입력 받은 시간으로 적용한다.

**E2. 버튼을 눌러 저장하지 않고 알람모드로 돌아간다.**

- ⇒ p9에서 언급하였듯, Activity 2031에 있는 Alternative Courses of Events나 Exceptional Courses of Events가 현재 Activity에서 누락되어 있는 곳이 많은데, 현재 'Set Alarm'의 경우에는 Exceptional Courses of Events가 추가되어 있어서 일관성마저 깨져버림

- ⇒ **SMA : Exceptional courses of Event는 상위 boundary가 아닌 현재 boundary에서 표현할 수 있는 것만 표시하여 몇 가지가 누락되어 보일 수 있음**

- ⇒ 해당 Reject 의견에 동의하며, 동시에 SMA 팀에서 전체적인 문서 수정을 통해 해당사항을 모두 일관되게 반영해주었음

### 13. Enable Alarm



- ⇒ enableAlarm 요청 시 System에서 화면에 알람 상태를 Enable 상태로 표시하는 화살표가 추가되어야 함
- ⇒ 이는 '20. Set D-Day Format', '21. Enable Interval Timer'에서도 같은 오류임

- ⇒ **SMA : 시스템의 Response에 해당하는 기능은 개념적인 성격이 크기 때문에 'Interaction Diagram'에서 Don't care 사항임**
- ⇒ **SMA 팀의 Reject 의견에 동의함**

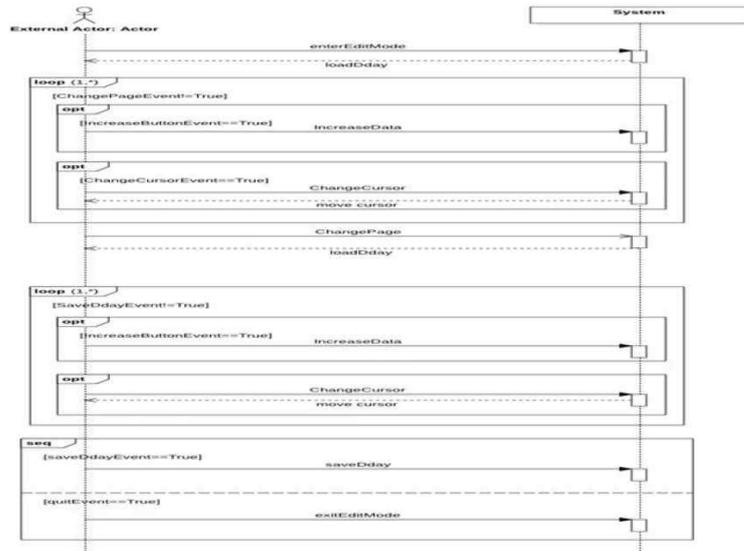
### 14. Disable Alarm



- ⇒ disableAlarm 요청 시 System에서 화면에 알람 상태를 disable 상태로 표시하는 화살표가 추가되어야 함
- ⇒ 이는 '22. Disable Interval Timer'에서도 같은 오류임

- ⇒ **SMA : 시스템의 Response에 해당하는 기능은 개념적인 성격이 크기 때문에 'Interaction Diagram'에서 Don't care 사항임**
- ⇒ **SMA 팀의 Reject 의견에 동의함**

## 17. Set D-Day



- ⇒ IncreaseData 후에, System에서 바뀐 데이터 값을 다시 User쪽에 전달하는 화살표가 필요함
- ⇒ saveDday 혹은 exitEditMode 전송 시 System에서 화면을 전환해주는 화살표가 추가되어야 함
- ⇒ **SMA : 시스템의 Response에 해당하는 기능은 개념적인 성격이 크기 때문에 'Interaction Diagram'에서 Don't care 사항임**
  - ⇒ **SMA 팀의 Reject 의견에 동의함**
- ⇒ 시작날짜, 종료날짜가 다르도록 디자인을 했기 때문에 첫 loadDday와 중간의 loadDday가 이름이 같으면 안 됨
- ⇒ 모든 loop 안의 모든 조건에서 '&& quitEvent != True'가 추가되어야 함
- ⇒ **성공적으로 반영되었음**

## Activity 2038. Refine System Test Case

28	Mute Beep test	버튼을 눌렀을 때, 알람이 종료되는지 test	28. Mute Beep	R 7.3
----	----------------	---------------------------	---------------	-------

- ⇒ Activity 2040부터 나오는 Hold 기능이 추가됨에 따라

해당 기능도 점검하는 Test Case를 추가

- ⇒ SMA : 버튼에 관한 부분은 Project의 도메인이 아님
  - ⇒ 해당 Reject 의견에 동의하며, 동시에 SMA 팀에서 해당사항을 반영해줌

### 1.1.3 Stage 2040 Design

Activity 2041. Define Essential Use Cases

[Use Case 17]

Alternative courses of Events	(2)~(9). : 1. D버튼을 눌러 종료날짜 페이지로 이동한다. (11)~(18). : 1. D버튼을 눌러 시작날짜 페이지로 이동한다. 2. C버튼을 누르면 (19)으로 이동한다.
Exceptional Courses of Events	(2)~(9): D버튼을 길게 누르면 저장되지 않고 D-day 모드로 돌아간다.

- ⇒ (2)~(9) 구간에서 C버튼을 누르면 바로 (19)로 가는지에 대한 여부가 누락됨
- ⇒ Exceptional Courses of Events에서 (2)~(9)뿐만 아니라 (11)~(18)에서도 D를 길게 누를 경우를 추가해야함
- ⇒ Activity 2031과 비교하여 (9)와 (18)에서 A버튼을 누르면 다시 커서가 년도1로 이동하는 기능이 누락됨
- ⇒ '2. C버튼을 누르면 (19)으로 이동한다.' -> '2. C버튼을 누르면 (20)으로 이동한다.' 로 수정 되어야함

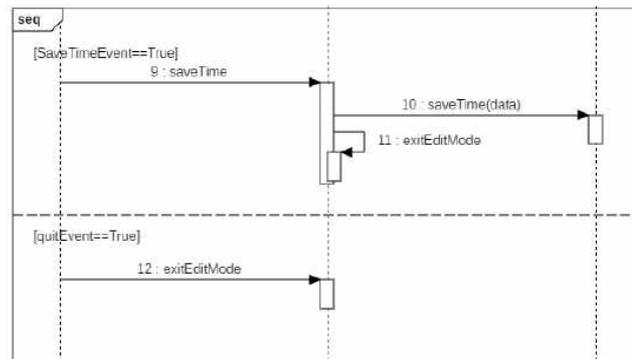
⇒ 성공적으로 반영되었음

- ⇒ Alternative Courses of Events에서 (2)와 (11)과 같은 System쪽에서의 기능은 포함될 필요가 없음
- ⇒ SMA : 시스템의 Response에 해당하는 기능은 개념적인 성격이 크기 때문에 'Interaction Diagram'에서 Don't care 사항임
  - ⇒ SMA 팀의 Reject 의견에 동의함

## Activity 2044. Define Essential Use Cases

### 1. Set Time

- ⇒ exitEditMode 후에 사용자에게 showTime 해주는 화살표가 누락되어 있음
- ⇒ 이는 '4. Set Tlmer', '11. Set Alarm', '12. Reset Alarm', '17. Set D-day', '23. Set Interval Timer', '24. Reset

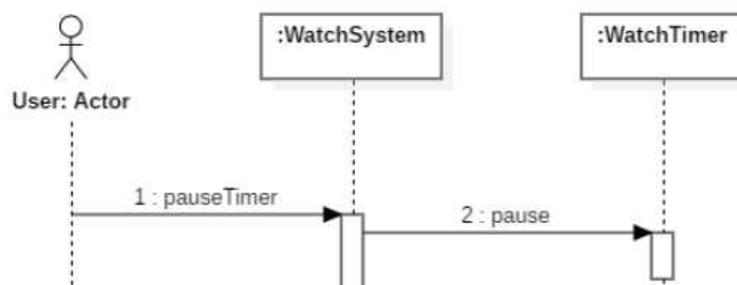


Interval Timer', '27. Set Mode' 에서도 같은 오류

- ⇒ **SMA : 시스템의 Response에 해당하는 기능은 개념적인 성격이 크기 때문에, 굳이 표시를 안 해줘도 Use Case에서 Interaction을 충분히 표시하였다고 판단하여 넣지 않음**

⇒ **SMA 팀의 Reject 의견에 동의함**

### 6. Pause Timer



- ⇒ 앞서 언급되었던 조건인 '타이머가 활성화 상태일 때'의 조건 상자가 누락되어있음
- ⇒ '7. Reset Timer' 에서처럼 'isPaused == false'와 같은 조건이 필요

⇒ 이는 '8. Activate Stopwatch', '9. Pause Stopwatch', '13. Enable Alarm', '14. Disable Alarm', '21.Enable Interval Time', '22.Disable Interval Time' 에서도 같은 오류

⇒ **SMA : 활성화 상태일 때와 같은 Guard는 User가 UI적인 부분에서 판단한다고 생각. 따라서 Interaction diagram에서 표시하지 않음**

⇒ **SMA 팀의 Reject 의견에 동의함**

## 1.2 Specification Review 2<sup>nd</sup>

### 1.2.1 Stage 2030 Analysis

Activity 2031. Define Essential Use Cases

[Use Case 16]

(1): 마지막 알람 일 때, D버튼을 누르면 첫번째 알람을 보여준다.

⇒ 다른 Stage 2030들과는 다르게, Stage 2040부터 나타나는 'D 버튼'과 같은 특정 버튼이 언급되어 있어 일관성을 해침

[Use Case 17]

(2) (A) : 시작날짜를 입력하지 않고 다음 버튼을 누른다. (8)로 이동한다

(6) (S) : 사용자가 커서를 바꾸면 (2)로 이동한다.

(12) (S) : 사용자가 커서를 바꾸면 (8)로 이동한다.

(14) (S) : 입력받은 종료날짜를 저장한다.

(2): 첫번째 년도가 99일때 증가시키면 19로 변화한다.

(10): 첫번째 년도가 99일때 증가시키면 19로 변화한다.

(4)~(5): 사용자가 '월'을 바꿀 시 '일'의 변화값이의 범위가 1일부터 '월'의 최대일로 재설정된다.

(9)~(10): 사용자가 '월'을 바꿀 시 '일'의 변화값이의 범위가 1일부터 '월'의 최대일로 재설정된다.

(2)~(12): 버튼을 눌러 저장하지 않고 D-day 모드로 돌아간다.

⇒ Typical Courses of Events와 비교해봤을 때, Alternative Courses of Events와의 숫자 값들이 일치하지 않음

## 1.2.2 Stage 2040 Design

Activity 2041. Design Real Use Cases

(2)~(9), :
1. D버튼을 눌러 종료날짜 페이지로 이동한다.
(9), :
1. B버튼을 누르면 (2)으로 이동한다.
(2)~(18), :
1. C버튼을 누르면 (20)으로 이동한다.
(18):
1. A버튼을 누르면 (11)로 이동한다.
(11)~(18), :
1. D버튼을 눌러 (2로 이동한다).
2. C버튼을 누르면 (20)으로 이동한다.
(2): 첫번째 년도가 99일때 증가시키면 19로 변화한다.
(12): 첫번째 년도가 99일때 증가시키면 19로 변화한다.
(6)~(7): 사용자가 '월'을 바꿀 시 '일'의 변화값의 범위가 1일부터 '월'의 최대일로 재설정된다.
(15)~(16): 사용자가 '월'을 바꿀 시 '일'의 변화값의 범위가 1일부터 '월'의 최대일로 재설정된다.

⇒ (9)에서 B버튼을 누르는 것이 아닌 A버튼을 눌러야 함  
⇒ 치명적인 오류는 아니나, (2)~(18)에서 'C버튼을 누르면 (20)으로 이동한다.'를 언급하였기에, (11)~(18)에서 '2번' 내용을 다시 제시할 필요가 없다고 판단됨

## 2 System Testing

### 2.1 Category-partition Testing Report

#### 2.1.1 Categorize

Group	Category	Values	#
Functions	Selected mode	timekeeping	1011
		set timekeeping	1012
		timer	1013
		set timer	1014
		alarm	1015
		set alarm	1016
		stopwatch	1017
		record stopwatch	1018
		dday	1019
		set dday	1020
		interval	1021
		set interval	1022
Actions	Timekeeping	convert time format	2011
		go to next function	2012
	Set Timekeeping	set time	2021
	Timer	active timer	2031
		pause timer	2032

		restart timer	2033
		go to next function	2034
	Set Timer	set timer	2041
	Stopwatch	active stopwatch	2051
		pause stopwatch	2052
		reset stopwatch	2053
		go to next function	2054
	Alarm	see next alarm	2061
		reset alarm	2062
		enable alarm	2063
		disable alarm	2064
		go to next function	2065
	Set Alarm	set alarm	2071
	Dday	reset dday	2081
		change dday format	2082
		go to next function	2083
	Set Dday	set dday	2091
	Interval	reset interval	2101
		enable interval	2102
		disable interval	2103

		go to next function	2104
	Set Interval	set interval	2111
Check	Timekeeping	check valid time	3011
		check counting real time	3012
		check converting format	3013
	Set Timekeeping	year value up one cycle	3021
		month value up one cycle	3022
		day value up one cycle	3023
		hour value up one cycle	3024
		minute value up one cycle	3025
	Timer	ring buzzer	3031
		stop buzzer	3032
		no ring buzzer	3033
		check counting down	3034
		check no counting down	3035
	Set Timer	hour value up one cycle	3041
		minute value up one cycle	3042
		second value up one cycle	3043
	Stopwatch	check counting up	3051

		check no counting up	3052
	Alarm	check valid alarm list	3061
		ring buzzer	3062
		stop buzzer	3063
		no ring buzzer	3064
	Set alarm	day value up one cycle	3071
		period value up one cycle	3072
		hour value up one cycle	3073
		minute value up one cycle	3074
	Dday	check valid dday list	3081
		check value up after one day	3082
		check value up after one month	3083
		check value up after one year	3084
	Set Dday	year value up one cycle	3091
		month value up one cycle	3092
		day value up one cycle	3093
	Interval	ring buzzer	3101
		stop buzzer	3102
		no ring buzzer	3103
	Set Interval	hour value up one cycle	3111

		<b>minute value up one cycle</b>	<b>3112</b>
	<b>Next</b>	<b>check save premode info</b>	<b>3121</b>

## 2.1.2 Description

Values	Description
timekeeping	시간 표시가 올바른지 확인한다.
set timekeeping	시간 설정이 올바르게 작동하는지 확인한다.
timer	타이머 기능이 올바르게 작동하는지 확인한다.
set timer	타이머 설정이 올바르게 작동하는지 확인한다.
alarm	알람 기능이 올바르게 작동하는지 확인한다.
set alarm	알람 설정 기능이 올바르게 작동하는지 확인한다.
stopwatch	스톱워치 기능이 올바르게 작동하는지 확인한다.
record stopwatch	스톱워치 기록이 올바르게 작동하는지 확인한다.
dday	D+Day 기능이 올바르게 작동하는지 확인한다.
set dday	D+Day 설정이 올바르게 작동하는지 확인한다.
interval	인터벌 기능이 올바르게 작동하는지 확인한다.
set interval	인터벌 설정 기능이 올바르게 작동하는지 확인한다.
convert time format	시간 출력 형식 변경이 올바르게 작동하는지 확인한다.
go to next function	다음 기능으로 이동 시 결과를 확인한다.
set time	시간 설정 기능이 올바르게 작동하는지 확인한다.
active timer	타이머 작동 기능이 올바르게 작동하는지 확인한다.
pause timer	타이머 중지 기능이 올바르게 작동하는지 확인한다.
reset timer	타이머 리셋 기능이 올바르게 작동하는지 확인한다.

go to next function	다음 기능으로 이동 시 결과를 확인한다.
set timer	타이머 설정 기능이 올바르게 작동하는지 확인한다.
see next alarm	다음 알람 화면이 정상적으로 표시되는지 확인한다.
reset alarm	알람 리셋 기능이 올바르게 작동하는지 확인한다.
enable alarm	알람을 활성화 했을 경우 올바르게 작동하는지 확인한다.
disable alarm	알람을 비활성화 했을 경우 올바르게 작동하는지 확인한다.
go to next function	다음 기능으로 이동 시 결과를 확인한다.
set alarm	알람 설정 기능이 올바르게 작동하는지 확인한다.
active stopwatch	스톱워치 시작 기능이 올바르게 작동하는지 확인한다.
pause stopwatch	스톱워치 정지 기능이 올바르게 작동하는지 확인한다.
reset stopwatch	스톱워치 리셋 기능이 올바르게 작동하는지 확인한다.
go to next function	다음 기능으로 이동 시 결과를 확인한다.
reset dday	D+Day 일정을 리셋한 후 결과를 확인한다.
change dday format	D+Day 형식을 변환한 후 결과를 확인한다.
go to next function	다음 기능으로 이동 시 결과를 확인한다.
set dday	D+Day 값을 설정하는 기능이 올바르게 작동하는지 확인한다.
reset interval	인터벌 리셋 기능이 올바르게 작동하는지 확인한다.
enable interval	인터벌 기능을 활성화 했을 경우 올바르게 작동하는지 확인한다.
disable interval	인터벌 기능을 비활성화 했을 경우 올바르게 작동하는지 확인한다.
go to next function	다음 기능으로 이동 시 결과를 확인한다.

set interval	인터벌 설정 기능이 올바르게 작동하는지 확인한다.
check valid time	시간보기 기능에서 유효한 시간인지 검사한다.
check counting real time	시간보기 기능에서 실시간으로 표현되는지 검사한다.
check converting format	시간보기 기능에서 시간 포맷 변경 시 올바른 시간 표시를 검사한다.
check save premode info	시간보기 다음 기능으로 넘어갈 때, 정보가 저장되는지 검사한다.
year value up one cycle	시간 설정 시 년도 임계치를 검사한다.
month value up one cycle	시간 설정 시 월 임계치를 검사한다.
day value up one cycle	시간 설정 시 일 임계치를 검사한다.
hour value up one cycle	시간 설정 시 시간 임계치를 검사한다.
minute value up one cycle	시간 설정 시 분 임계치를 검사한다.
ring buzzer	타이머 작동 시, 버저가 울리는지 검사한다.
stop buzzer	타이머 작동 시, 버저가 멈추는지 검사한다.
check counting down	타이머 작동 시, 초읽기가 되는지 검사한다.
no ring buzzer	타이머 일시정지 시, 버저가 울리지 않는지 검사한다.
check no counting down	타이머 일시정지 시, 초읽기가 멈추는지 검사한다.
no ring buzzer	타이머 초기화 시, 버저가 울리지 않는지 검사한다.
check no counting down	타이머 초기화 시, 버저가 울리지 않는지 검사한다.
check save premode info	타이머 다음 기능으로 넘어갈 때에, 정보가 저장되는지 검사한다.
hour value up one cycle	타이머 설정 시 시간 임계치를 검사한다.
minute value up one cycle	타이머 설정 시 분 임계치를 검사한다.

second value up one cycle	타이머 설정 시 초 임계치를 검사한다.
check valid alarm list	알람 리스트가 유효한지 검사한다
check valid alarm list	알람 초기화 시, 리스트가 유효한지 검사한다.
check valid alarm list	알람 활성화 시, 버저가 울리는지 검사한다.
ring buzzer	알람 활성화 시, 버저가 울리는지 검사한다.
stop buzzer	알람 활성화 시, 버저가 멈추는지 검사한다.
check valid alarm list	알람 비활성화 시, 리스트가 유효한지 검사한다.
no ring buzzer	알람 비활성화 시, 버저가 울리지 않는지 검사한다.
check save premode info	알람 다음 기능으로 넘어갈 때에, 정보가 저장되는지 검사한다.
day value up one cycle	알람 설정 시 일 임계치를 검사한다.
period value up one cycle	알람 설정 시 주기 임계치를 검사한다.
hour value up one cycle	알람 설정 시 시간 임계치를 검사한다.
minute value up one cycle	알람 설정 시 분 임계치를 검사한다.
check counting up	스톱워치 작동 시, 값이 증가하는지 검사한다.
check no counting up	스톱워치 일시정지 시, 값 증가가 멈추는지 검사한다.
check no counting up	스톱워치 초기화 시, 값 증가가 멈추는지 검사한다.
check save premode info	스톱워치 다음 기능으로 넘어갈 때, 정보가 저장되는지 검사한다.
check valid dday list	D+Day 초기화 후, 리스트가 유효한지 검사한다.
check value up after one day	D+Day 초기화 후, 1일이 지난 뒤 값이 유효한지 검사한다.
check value up after one month	D+Day 초기화 후, 1달이 지난 뒤 값이 유효한지 검사한다.

check value up after one year	D+Day 초기화 후, 1년이 지난 뒤 값이 유효한지 검사한다.
check valid dday list	D+Day 형식 변환 후, 리스트가 유효한지 검사한다.
check value up after one day	D+Day 형식 변환 후, 1일이 지난 뒤 값이 유효한지 검사한다.
check value up after one month	D+Day 형식 변환 후, 1달이 지난 뒤 값이 유효한지 검사한다.
check value up after one year	D+Day 형식 변환 후, 1년이 지난 뒤 값이 유효한지 검사한다.
check save premode info	D+Day 다음 기능으로 넘어갈 때, 정보가 저장되는지 검사한다.
year value up one cycle	D+Day 설정 시, 년도 임계치를 검사한다.
month value up one cycle	D+Day 설정 시, 월 임계치를 검사한다.
day value up one cycle	D+Day 설정 시, 일 임계치를 검사한다.
ring buzzer	Interval 활성화 시, 버저가 울리는지 검사한다.
stop buzzer	Interval 활성화 시, 버저가 멈추는지 검사한다.
no ring buzzer	Interval 비활성화 시, 버저가 울리지 않는지 검사한다.
check save premode info	Interval 다음 기능으로 넘어갈 때, 정보가 저장되는지 검사한다.
hour value up one cycle	Interval 설정 시, 시간 임계치를 검사한다.
minute value up one cycle	Interval 설정 시, 분 임계치를 검사한다.

### 2.1.3 Property & Constraints

Category	Values	Property / Constraints
Function mode	timekeeping	[property Timekeeping]
	set timekeeping	[property TimekeepingSet]
	timer	[property Timer]
	set timer	[property TimerSet]
	alarm	[property Alarm]
	set alarm	[property AlarmSet]
	stopwatch	[property Stopwatch]
	record stopwatch	[property StopwatchRecord]
	dday	[property Dday]
	set dday	[property DdaySet]
	interval	[property Interval]
	set interval	[property IntervalSet]
Timekeeping	convert time format	[if (Timekeeping)] [property FormatChange]
	go to next function	[if (Timekeeping)] [property Next]
Timekeeping Set	set time	[if (TimekeepingSet)]
Timer	active timer	[if (Timer)] [property TimerOn]
	pause timer	[if (Timer)] [property TimerOff]
	reset timer	[if (Timer)] [property TimerOff]

	<b>go to next function</b>	<b>[if (Timer)] [property Next]</b>
<b>TimerSet</b>	<b>set timer</b>	<b>[if (TimerSet)]</b>
<b>Stopwatch</b>	<b>active stopwatch</b>	<b>[if (Stopwatch)] [property StopwatchOn]</b>
	<b>pause stopwatch</b>	<b>[if (Stopwatch)] [property StopwatchOff]</b>
	<b>reset stopwatch</b>	<b>[if (Stopwatch)] [property StopwatchOff]</b>
	<b>go to next function</b>	<b>[if (Stopwatch)] [property Next]</b>
<b>Alarm</b>	<b>see next alarm</b>	<b>[if (Alarm)]</b>
	<b>reset alarm</b>	<b>[if (Alarm)]</b>
	<b>enable alarm</b>	<b>[if (Alarm)] [property AlarmOn]</b>
	<b>disable alarm</b>	<b>[if (Alarm)] [property AlarmOff]</b>
	<b>go to next function</b>	<b>[if (Alarm)] [property Next]</b>
<b>AlarmSet</b>	<b>set alarm</b>	<b>[if (AlarmSet)]</b>
<b>Dday</b>	<b>reset dday</b>	<b>[if (Dday)]</b>
	<b>change dday format</b>	<b>[if (Dday)]</b>
	<b>go to next function</b>	<b>[if (Dday)] [property Next]</b>
<b>DdaySet</b>	<b>set dday</b>	<b>[if (DdaySet)]</b>
<b>Interval</b>	<b>reset interval</b>	<b>[if (Interval)]</b>
	<b>enable interval</b>	<b>[if (Interval)] [property IntervalOn]</b>
	<b>disable interval</b>	<b>[if (Interval)] [property IntervalOff]</b>
	<b>go to next function</b>	<b>[if (Interval)] [property Next]</b>

<b>IntervalSet</b>	set interval	[if (IntervalSet)]
<b>Timekeeping</b>	check valid time	[if (Timekeeping && !Next)]
	check counting real time	[if (Timekeeping && !Next)]
	check converting format	[if (Timekeeping && !Next && FormatChange)]
<b>Timekeeping Set</b>	year value up one cycle	[if (TimekeepingSet)]
	month value up one cycle	[if (TimekeepingSet)]
	day value up one cycle	[if (TimekeepingSet)]
	hour value up one cycle	[if (TimekeepingSet)]
	minute value up one cycle	[if (TimekeepingSet)]
<b>Timer</b>	ring buzzer	[if (Timer && TimerOn && !Next)]
	stop buzzer	[if (Timer && TimerOn && !Next)]
	no ring buzzer	[if (Timer && TimerOff && !Next)]
	check counting down	[if (Timer && TimerOn && !Next)]
	check no counting down	[if (Timer && TimerOff && !Next)]
<b>TimerSet</b>	hour value up one cycle	[if (TimerSet)]
	minute value up one cycle	[if (TimerSet)]
	second value up one cycle	[if (TimerSet)]
<b>Stopwatch</b>	check counting up	[if (Stopwatch && StopwatchOn && !Next)]
	check no counting up	[if (Stopwatch && StopwatchOff && !Next)]
<b>Alarm</b>	check valid alarm list	[if (Alarm && !Next)]

	ring buzzer	[if (Alarm && AlarmOn && !Next)]
	stop buzzer	[if (Alarm && AlarmOn && !Next)]
	no ring buzzer	[if (Alarm && AlarmOff && !Next)]
AlarmSet	day value up one cycle	[if (AlarmSet)]
	period value up one cycle	[if (AlarmSet)]
	hour value up one cycle	[if (AlarmSet)]
	minute value up one cycle	[if (AlarmSet)]
DDay	check valid dday list	[if (Dday && !Next)]
	check value up after one day	[if (Dday && !Next)]
	check value up after one month	[if (Dday && !Next)]
	check value up after one year	[if (Dday && !Next)]
DDaySet	year value up one cycle	[if (DdaySet)]
	month value up one cycle	[if (DdaySet)]
	day value up one cycle	[if (DdaySet)]
Interval	ring buzzer	[if (Interval && IntervalOn && !Next)]
	stop buzzer	[if (Interval && IntervalOn && !Next)]
	no ring buzzer	[if (Interval && IntervalOff && !Next)]
IntervalSet	hour value up one cycle	[if (IntervalSet)]
	minute value up one cycle	[if (IntervalSet)]
Next	check save pre mode info	[if (Next)]

## 2.1.4 Text file for TSL-Generator

```
1 Function Mode:
2   #Timekeeping:
3     timekeeping.           [property Timekeeping]
4     set timekeeping.      [property TimekeepingSet]
5   #Timer:
6     timer.                 [property Timer]
7     set timer.            [property TimerSet]
8   #Alarm:
9     alarm.                 [property Alarm]
10    set alarm.             [property AlarmSet]
11  #Stopwatch:
12    stopwatch.             [property Stopwatch]
13    record stopwatch.      [property StopwatchRecord]
14  #DDay:
15    dday.                  [property Dday]
16    set dday.              [property DdaySet]
17  #Interval:
18    interval.              [property Interval]
19    set interval.          [property IntervalSet]
20
21 Do:
22  #Timekeeping:
23    convert time format.   [if (Timekeeping)] [property FormatChange]
24    go to next function.   [if (Timekeeping)] [property Next]
25  #TimekeepingSet:
26    set time.              [if (TimekeepingSet)]
27
28  #Timer:
29    active timer.          [if (Timer)] [property TimerOn]
30    pause timer.           [if (Timer)] [property TimerOff]
31    reset timer.           [if (Timer)] [property TimerOff]
32    go to next function.   [if (Timer)] [property Next]
33  #TimerSet:
34    set timer.             [if (TimerSet)]
35
36  #Stopwatch:
37    active stopwatch.      [if (Stopwatch)] [property StopwatchOn]
38    pause stopwatch.       [if (Stopwatch)] [property StopwatchOff]
39    reset stopwatch.       [if (Stopwatch)] [property StopwatchOff]
40    go to next function.   [if (Stopwatch)] [property Next]
41
42  #Alarm:
43    see next alarm.        [if (Alarm)]
44    reset alarm.           [if (Alarm)]
45    enable alarm.          [if (Alarm)] [property AlarmOn]
46    disable alarm.         [if (Alarm)] [property AlarmOff]
47    go to next function.   [if (Alarm)] [property Next]
48  #AlarmSet:
```

```

49         set alarm.                                [if (AlarmSet)]
50
51     #Dday:
52         reset dday.                                [if (Dday)]
53         change dday format.                        [if (Dday)]
54         go to next function.                       [if (Dday)] [property Next]
55     #DdaySet:
56         set dday.                                  [if (DdaySet)]
57
58     #Interval:
59         reset interval.                            [if (Interval)]
60         enable interval.                           [if (Interval)] [property IntervalOn]
61         disable interval.                          [if (Interval)] [property IntervalOff]
62         go to next function.                       [if (Interval)] [property Next]
63     #IntervalSet:
64         set interval.                               [if (IntervalSet)]
65
66     Check Screen:
67     #Timekeeping:
68         check valid time.                          [if (Timekeeping && !Next)]
69         check counting real time.                  [if (Timekeeping && !Next)]
70         check converting format.                   [if (Timekeeping && !Next && FormatChange)]
71     #TimekeepingSet:
72         year value up one cycle.                   [if (TimekeepingSet)]
73         month value up one cycle.                  [if (TimekeepingSet)]
74         day value up one cycle.                    [if (TimekeepingSet)]
75         hour value up one cycle.                   [if (TimekeepingSet)]
76         minute value up one cycle.                 [if (TimekeepingSet)]
77
78     #Timer:
79         ring buzzer.                                [if (Timer && TimerOn && !Next)]
80         stop buzzer.                               [if (Timer && TimerOn && !Next)]
81         no ring buzzer.                            [if (Timer && TimerOff && !Next)]
82         check counting down.                       [if (Timer && TimerOn && !Next)]
83         check no counting down.                    [if (Timer && TimerOff && !Next)]
84     #TimerSet:
85         hour value up one cycle.                    [if (TimerSet)]
86         minute value up one cycle.                 [if (TimerSet)]
87         second value up one cycle.                 [if (TimerSet)]
88
89     #Stopwatch:
90         check counting up.                          [if (Stopwatch && StopwatchOn && !Next)]
91         check no counting up.                      [if (Stopwatch && StopwatchOff && !Next)]
92
93     #Alarm:
94         check valid alarm list.                    [if (Alarm && !Next)]
95         ring buzzer.                               [if (Alarm && AlarmOn && !Next)]
96         stop buzzer.                               [if (Alarm && AlarmOn && !Next)]
97         no ring buzzer.                           [if (Alarm && AlarmOff && !Next)]
98
99     #AlarmSet:
100         day value up one cycle.                    [if (AlarmSet)]
101         period value up one cycle.                 [if (AlarmSet)]
102         hour value up one cycle.                   [if (AlarmSet)]
103         minute value up one cycle.                 [if (AlarmSet)]

```

```

103
104 #DDay:
105     check valid dday list.                [if (Dday && !Next)]
106     check value up after one day.         [if (Dday && !Next)]
107     check value up after one month.       [if (Dday && !Next)]
108     check value up after one year.        [if (Dday && !Next)]
109 #DDaySet:
110     year value up one cycle.              [if (DdaySet)]
111     month value up one cycle.            [if (DdaySet)]
112     day value up one cycle.              [if (DdaySet)]
113
114 #Interval:
115     ring buzzer.                          [if (Interval && IntervalOn && !Next)]
116     stop buzzer.                          [if (Interval && IntervalOn && !Next)]
117     no ring buzzer.                       [if (Interval && IntervalOff && !Next)]
118 #IntervalSet:
119     hour value up one cycle.              [if (IntervalSet)]
120     minute value up one cycle.           [if (IntervalSet)]
121
122 #Next:
123     check beep sound.                     [if (Next)]
124     check save pre mode info.            [if (Next)]

```

## 2.1.5 Test Result

TestCase	Result	Description
Test01 – 1011.2011.3011	Pass	
Test02 – 1011.2011.3012	Pass	
Test03 – 1011.2011.3013	Pass	
Test04 – 1011.2012.3121	Pass	
Test05 – 1012.2021.3021	Pass	
Test06 – 1012.2021.3022	Pass	
Test07 – 1012.2021.3023	Pass	
Test08 – 1012.2021.3024	Pass	
Test09 – 1012.2021.3025	Pass	
Test10 – 1013.2031.3031	Pass	
Test11 – 1013.2031.3032	Pass	
Test12 – 1013.2031.3034	Pass	
Test13 – 1013.2032.3033	Pass	
Test14 – 1013.2032.3035	Pass	
Test15 – 1013.2033.3033	Pass	
Test16 – 1013.2033.3035	Pass	
Test17 – 1013.2034.3121	Pass	
Test18 – 1014.2041.3041	Pass	
Test19 – 1014.2041.3042	Pass	
Test20 – 1014.2041.3043	Pass	
Test21 – 1015.2061.3061	Pass	
Test22 – 1015.2062.3061	Pass	
Test23 – 1015.2063.3061	Pass	
Test24 – 1015.2063.3062	Pass	
Test25 – 1015.2063.3063	Pass	
Test26 – 1015.2064.3061	Pass	
Test27 – 1015.2064.3064	Pass	
Test28 – 1015.2065.3121	Pass	
Test29 – 1016.2071.3071	Pass	
Test30 – 1016.2071.3072	Pass	
Test31 – 1016.2071.3073	Pass	
Test32 – 1016.2071.3074	Pass	
Test33 – 1017.2051.3051	Pass	
Test34 – 1017.2052.3052	Pass	
Test35 – 1017.2053.3052	Pass	
Test36 – 1017.2054.3121	Pass	
Test37 – 1018	Pass	
Test38 – 1019.2081.3081	Pass	
Test39 – 1019.2081.3082	Pass	
Test40 – 1019.2081.3083	Pass	
Test41 – 1019.2081.3084	Pass	
Test42 – 1019.2082.3081	Pass	
Test43 – 1019.2082.3082	Pass	
Test44 – 1019.2082.3083	Pass	
Test45 – 1019.2082.3084	Pass	

Test46 – 1019.2083.3121	Pass	
Test47 – 1020.2091.3091	Pass	
Test48 – 1020.2091.3092	Pass	
Test49 – 1020.2091.3093	Pass	
Test50 – 1021.2101	Pass	
Test51 – 1021.2102.3101	Pass	
Test52 – 1021.2102.3102	Pass	
Test53 – 1021.2103.3103	Pass	
Test54 – 1021.2104.3121	Pass	
Test55 – 1022.2111.3111	Pass	
Test56 – 1022.2111.3112	Pass	
전체 Test Case 56개	Pass: 56개 Fail: 0개	Pass Percentage – 100%

## 2.2 Pairwise Testing Report

### 2.2.1 Test Result

No	Selected Function	Running	Buzzer	Action	Combo Countdown	RESULT pass/fail
1	TimeKeeping	running	ringing	timekeeping	142	PASS
2	Timer	stoped	ringing	pause	130	PASS
3	Alarm	stoped	stoped	disable	114	PASS
4	Stopwatch	running	ringing	active	109	PASS
5	DDay	running	stoped	set	99	PASS
6	Interval	stoped	ringing	enable	94	PASS
7	Stopwatch	running	stoped	pause	90	PASS
8	Interval	running	stoped	enable	82	PASS
9	TimeKeeping	running	stoped	convertFormat	76	PASS
10	Timer	stoped	stoped	active	73	PASS
11	TimeKeeping	running	stoped	setTime	65	PASS
12	Alarm	stoped	stoped	list	59	PASS
13	DDay	running	stoped	change	55	PASS
14	Timer	running	stoped	nextFunction	53	PASS
15	Alarm	running	stoped	enable	51	PASS
16	Interval	running	stoped	disable	46	PASS
17	Interval	running	stoped	nextFunction	44	PASS
		TC NUM	PASS	FAIL	PERCENTAGE	
		17	17	0	100.00%	

⇒ 사용한 Tool : Testcover

## 2.3 Brute Force Testing Result

### 2.3.1 Testing Result

Test	Num	Description	P/F
Timekeeping	1-1	'년도1'의 값을 99보다 크게 설정한다.	Pass
	1-2	모드 설정에서 9999년 12월 31일로 설정 후, 하루가 지나면 각 숫자 값들이 한 칸씩 밀린다.	Fail
WatchTimer	2-1	타이머 설정 후 모드 변경하여 타이머를 제외하면 알람이 울리는지 확인한다.	Pass
	2-2	타이머가 0이 되어 알람이 울리고 멈추게 한 뒤 모든 값(시분초)이 기존 Timer 값으로 초기화 되는지 확인한다.	Pass
Alarm	3-1	수정모드에서 D버튼을 누르면 enabled된 알람이 disabled 상태로 변경되는지 확인한다.	Pass
	3-2	수정모드 접근 후 D버튼을 눌러 0으로 리셋하고 C버튼으로 저장하면 알람이 리셋 되는지 확인한다.	Pass
D-Day	4-1	'년도 1'값을 99이상으로 설정한다.	Pass
	4-2	종료일을 시작일보다 이전으로 설정한다.	Pass
	4-3	(종료일 - 시작일)이 9999보다 크도록 설정한다.	Pass
	4-4	시작 날짜에서 종료 날짜로 변경 시 커서가 년도1로 위치한다.	Fail
	4-5	시작일과 종료일이 같고, 날짜를 오늘보다 이후로 설정한 경우 d-day와 % 표기가 잘 되는지	Pass

		확인한다.	
IntervalTimer	5-1	타이머 설정 후 모드변경 하여 타이머를 제외하면 알람이 울리는지 확인한다.	Pass
	5-2	수정모드 접근 후 D버튼을 눌러 0으로 리셋하고 C버튼으로 저장하면 알람이 리셋 되는지 확인한다.	Pass

Pass 성공률 :  $11/13 * 100 = 84.6(\%)$

### 2.3.2 Fail Case Report

Test	Num	Expected Output	Test Output
Timekeeping	1-2	9999년 12월 31일에서 하루가 지나면, 따로 지정해놓은 날짜로 변경된 후 정상적으로 시간이 흐를 것이다.	시계의 각 숫자 값들이 한 칸씩 밀려나 정상적인 시간 값을 보여주지 못함
D-Day	4-4	수정 모드에서 D버튼을 누르면 스펙에서 명시한 대로 년도1로 커서가 옮겨질 것이다.	커서가 현재 상태에 머물러있음

## 3 Static Analysis

### 3.1 SonarQube Rule

#### 3.1.1 Rule of Findbugs

Ref	Category
1	Bad Practice
2	Correctness
3	Experimental
4	Internationalization
5	Malicious code vulnerability
6	Multithreaded correctness
7	Performance
8	Security
9	Dodgy code

⇒ Findbugs에서 정의한 규칙이 위배될 경우 대부분 "Bug"로 판별된다.

#### 3.1.2 Rule of PMD

Ref	Category	Description
1	Best Practice	Rules which enforce generally accepted best practices.
2	Code Style	Rules which enforce a specific coding style.
3	Design	Rules that help you discover design issues.
4	Documentation	Rules that are related to code documentation.
5	Error Prone	Rules to detect constructs that are either broken, extremely confusing or prone to runtime errors.
6	Multithreading	Rules that flag issues when dealing with multiple threads of execution.
7	Performance	Rules that flag suboptimal code.
8	Security	Rules that flag potential security flaws.
9	Additional rulesets	추가적인 규칙 정의

### 3.1.3 Rule of CheckStyle

Ref	Category
1	Google's java Style
2	Sus's java Style

### 3.1.4 Code Coverage

SMA 팀에서 작성한 junit Test를 기반으로 Code coverage 계산.

Jacoco Test Report를 활용하여 Sonarqube에서 해당 결과 xml 파일을 분석.

Java	<code>sonar.jacoco.reportPaths</code>	Path to JaCoCo reports in binary format. <b>Supported only for Java.</b>
Sonarqube Test Coverage docs ( <a href="https://docs.sonarqube.org/latest/analysis/coverage/">https://docs.sonarqube.org/latest/analysis/coverage/</a> )		
Java	<code>sonar.junit.reportPaths</code>	Comma-delimited list of paths to Surefire XML-format reports.
Sonarqube Test Execution docs ( <a href="https://docs.sonarqube.org/latest/analysis/coverage/">https://docs.sonarqube.org/latest/analysis/coverage/</a> )		

- ⇒ CTIP 환경에서 Gradle을 활용해 빌드를 진행할 때에, Junit Test를 진행한다.
- ⇒ Jacoco 플러그인을 활용하여, 빌드 단계에서 Junit Test 결과를 분석한다.
- ⇒ 분석 결과를 xml 파일로 저장한다.
- ⇒ SonarQube Scanner는 해당 xml 파일을 활용한다.

## 3.2 SonarQube Result

### 3.2.1 Dashboard



### 3.2.2 Bug

Ref	plug-in	tag	rule	count
1	findbugs	performance	Method needlessly boxes a boolean constant	64
2			Unread field	7
3			Method appears to call the same method on the same object redundantly	2
4			Unused field	2
5			Method allocations a java.awt.Graphics object without disposing it	1
6			Method concatenates strings using + in a loop	1
7			Method creates array using constants	1
8			Should be a static inner class	1
9		correctness	Class defines fields that are used only as locals	11
10			Method accesses list or array with constant index	10
11			Method explicitly sets the color of a Component	9
12			Constructor makes call to non-final method	5
13			Class has a circular dependency with other classes	1
14			Potential memory bloat in static field	1
15			Self assignment of field	1
16			Uninitialized read of field in constructor	1
17			Window sets size manually, and doesn't use pack	1
18		multi-threading	Multi-threading - Invokes run on a thread (did you mean to start it instead?)	1
19	sonar	multi-threading	"Thread.run()" should not be called directly	1
20		based-on-misra	Floating point numbers should not be tested for equality	1

### 3.2.3 Vulnerability

Ref	plug-in	tag	rule	count
1	pmd	security	Method returns internal array	4
2			Array is stored directly	2
3	findbugs	cwe, owasp-a4, wasc	Potential Path Traversal (file read)	1
4	sonar sonar	cwe	Class variable fields should not have public accessibility	5
5		cert, cwe, unpredictable	Mutable members should not be stored or returned directly	3
6		cert, cwe, unpredictable	Mutable fields should not be "public static"	1
7		cert, cwe	"public static" fields should be constant	1
8		no tags	Member variable visibility should be specified	1

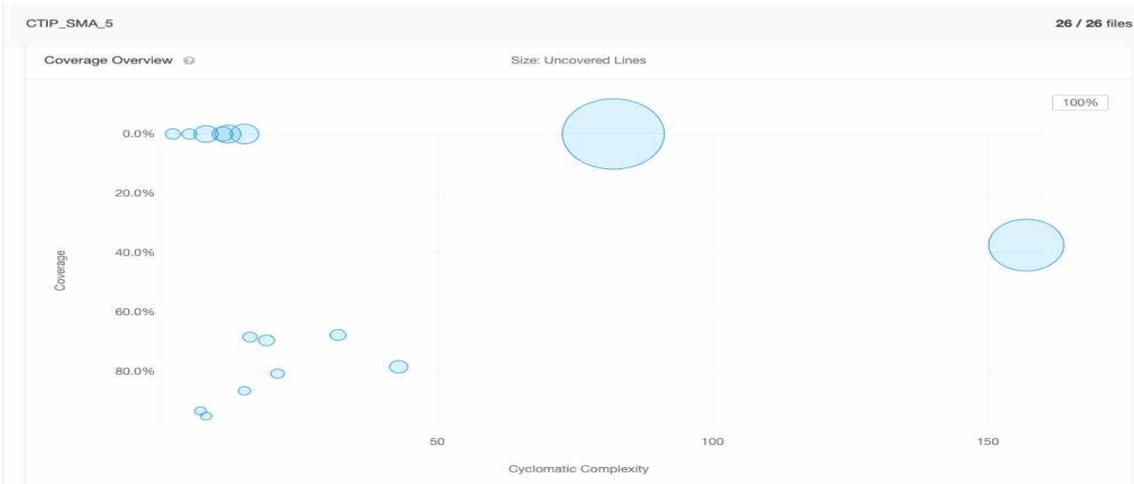
### 3.2.4 Code Smell

The screenshot shows a code quality tool interface. On the left, a list of code smells is displayed, with '(Java) Visibility Modifier' selected. The center panel shows a 'Tag' filter menu with a search bar and a list of tags such as 'convention', 'style', 'tests', etc. On the right, a list of tags is shown with their respective counts, including 'design' (6), 'comment' (5), 'confusing' (5), etc.

Ref	plug-in	tag	rule	count
1	pmd	multithreading	Non Thread Safe Singleton	2
2			Do Not Use Threads	1
3	sonar	multi-threading	"this" should not be exposed from constructors	7
4	sonar	junit	Literal boolean values should not be used in assertions	13
5	checkstyle	error-handling	Illegal Catch	1

- ⇒ Code Smell은 심각한 에러나 결함보다는, 문제로 이어질 수 있는 패턴들로 인하여 발견된다.
- ⇒ 많은 False Alarm을 포함하고 있다.

## 3.2.5 Code Coverage



Overview

Overall	
Coverage	95.0%
Lines to Cover	18
Uncovered Lines	0
Line Coverage	100%
Conditions to Cover	2
Uncovered Conditions	1
Condition Coverage	50.0%
Tests	
Errors	0
Failures	0
Skipped	0
Success	100%

TimeKeeping.java

Overall	
Coverage	35.1%
Lines to Cover	1,164
Uncovered Lines	688
Line Coverage	40.9%
Conditions to Cover	416
Uncovered Conditions	338
Condition Coverage	18.8%
Tests	
Unit Tests	72
Errors	0
Failures	0
Skipped	5
Success	100%

CTIP\_SMA\_5

```

38 kkom...
39 mois...
40 admi...
41 kkom...
42

```

Fully covered by tests.

Covered

```

43 public void setHourformat() {
44
45 admi...
46 kkom...
47 mois...
48 kkom...

```

Partially covered by tests (1 of 2 conditions).

Not covered

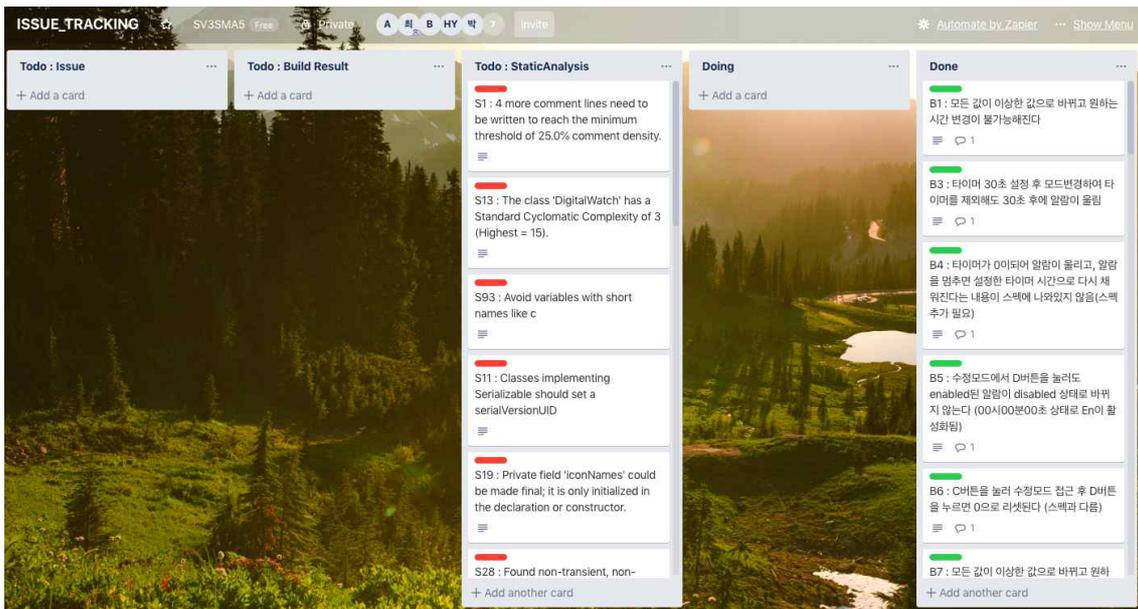
### 3.3 SonarQube Usage

#### ✓ Slack을 통한 Notification 확인



#### ✓ Trello Issue Card 확인

⇒ CTIP 환경에서 심각도를 기준으로 (Blocked / Critical / Major) 이슈를 자동 등록



## ✓ Sonarqube를 통해 Issue 확인

The screenshot shows the Sonarqube interface for project CTIP\_SMA\_5. The 'Issues' tab is active, displaying a list of 1,423 issues. The left sidebar shows filters for Type (Bug, Vulnerability, Code Smell) and Severity (Blocker, Critical, Major, Minor, Info). The main area shows a list of issues, including:

- Add a 'package-info.java' file to document the 'src/main/java/Logic' package** (Code Smell, Minor, Open, 20min effort)
- Add a 'package-info.java' file to document the 'src/main/java/GUI' package** (Code Smell, Minor, Open, 20min effort)
- Add or update the header of this file.** (Code Smell, Blocker, Open, 5min effort)
- 4 more comment lines need to be written to reach the minimum threshold of 25.0% comment density.** (Code Smell, Major, Open, 8min effort)
- Missing an import control file.** (Code Smell, Blocker, Open, Not assigned)
- Name 'GUI' must match pattern '^[a-z]+(\.[a-zA-Z\_][a-zA-Z0-9\_]\*)\*\$'.** (Code Smell, Major, Open, 20min effort)

## ✓ 해당 소스 코드로 이동

The screenshot shows the source code for `src/main/java/GUI/DigitalWatch.java` with several issues highlighted. The issues are:

- Method new GUI.DigitalWatch() Method explicitly sets the color of a Component** (Bug, Major, Open, 14 days ago, L44)
- Method new GUI.DigitalWatch() Method explicitly sets the color of a Component** (Bug, Major, Open, 17 days ago, L50)
- Window GUI.DigitalWatch sets size manually, and doesn't use pack** (Bug, Major, Open, 17 days ago, L58)

The code snippet shows the following lines:

```

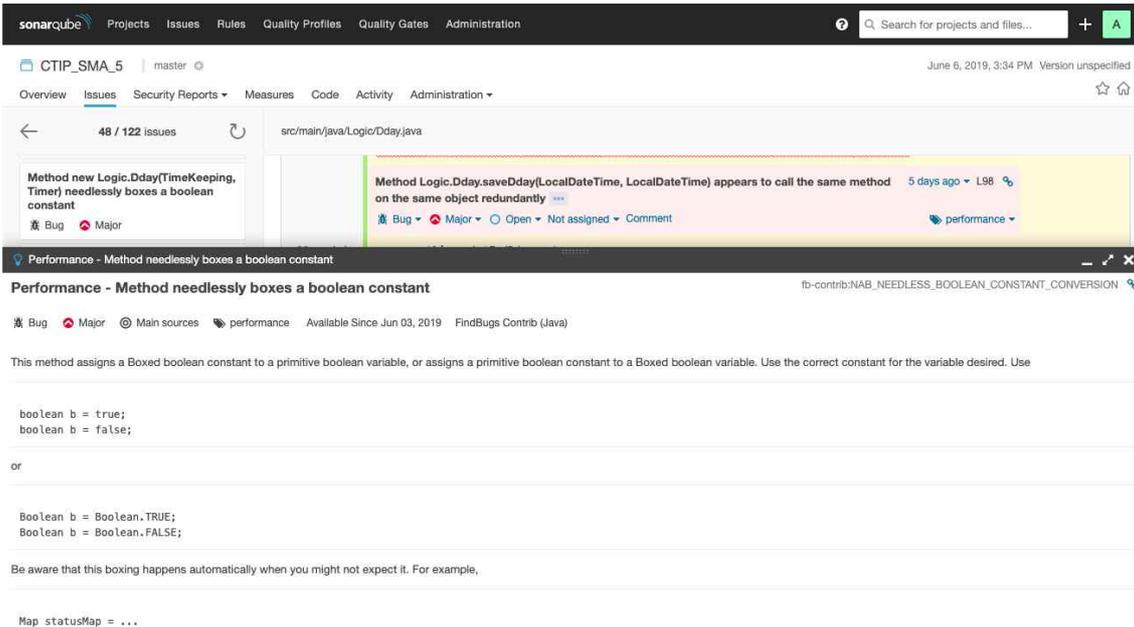
39 IconImages.getImage("segment/color1/" + Character.toString(digitFileName[1]));
40 IconImages.getImage("segment/color2/" + Character.toString(digitFileName[1]));
41 }
42 kkom... int[] btnX = {24, 646, 24, 646};
43 int[] btnY = {100, 100, 310, 310};
44 admI... this.setBackground(Color.black);

45 kkom... buttons = new WatchButton[4];
46 for(int btnIndex = 0; btnIndex < 4; btnIndex++){
47 WatchButton btn = new WatchButton();
48 btn.setSize(30, 60);
49 btn.setLocation(btnX[btnIndex], btnY[btnIndex]);
50 btn.setBackground(Color.red);

51 btn.setVisible(true);
52 btn.setShape((btnIndex % 2) == 0);
53 btn.setHorizontalAlignment(JLabel.CENTER);
54 this.add(btn);
55 buttons[btnIndex] = btn;
56 }
57 this.setTitle("Digital Watch");
58 this.setSize(700, 500);

```

- ✓ 위배 규칙 및 패턴을 확인한다.



- ✓ 코드 수정 - 이슈 해결
- ✓ 적절하지 않은 이슈에 대해서는 검증팀과 의논 - Slack 활용
- ✓ 각 이슈 처리 과정은 Trello를 통해 실시간 공유

### 3.4 Conclusion

- ✓ 미처 생각하지 못한 오류 혹은 결함을 미리 예방함으로써, 안정적인 소프트웨어를 개발할 수 있다.
- ✓ 지속적인 Static Analysis를 통해서, 개발자들은 로직 구현에 더욱 집중할 수 있을 것이다.
- ✓ 프로젝트 목적이나 환경에 맞게, 규칙 및 패턴을 정의하여 활용하는 것이 필요하다.

## 4 Overall

### 4.1 System Test Result

⇒ **Category-Partition Test : 56/56 \* 100 = 100% PASS**

- 1<sup>st</sup> Cycle에서 발견된 오류 수정 및 문서 변경들을 통하여 100%라는 성공률을 기록하게 되었습니다. 기존에도 성공률이 높았기 때문에, 약간의 코드 수정만으로도 쉽게 100% 성공률에 도달할 수 있었던 것 같습니다.

⇒ **Pairwise Test : 17/17 \* 100 = 100% PASS**

- 1<sup>st</sup> Cycle에서와 마찬가지로, 100%라는 완벽한 성공률이 나왔습니다. 이는 'Testcover'라는 Web Tool의 특성이 가미되었을 수도 있지만, 전체적으로 코드가 잘 구현되었기 때문이라고 생각합니다.

⇒ **Brute Force Test : 11/13 \* 8 = 84.6% PASS**

- 1<sup>st</sup> Cycle에서 진행했던 15개의 Brute Force Test 중에서, SA팀에서 철회한 테스트 1개와 SMA팀에서 구현 상세를 수정한 4개를 제외한 나머지 11개의 Test를 모두 성공하였습니다. 이외에도 추가로 발견한 2개의 Test를 합쳐 총 84.6%의 매우 높은 성공률을 기록하였습니다.

### 4.2 Static Analysis Result

#### 4.2.1 Issues

- Sonar way, Findbugs, PMD, CheckStyle 규칙 및 패턴을 통해 분석한 결과, 다양한 이슈들을 발견할 수 있었습니다. 대부분 하나의 규칙에 대해서 다수의 코드에서 위배되었고, GUI와 Multi-Threading에서 많은 위배가 발견되었습니다.

#### 4.2.2 Code Coverage

- Jenkins 빌드 환경에서 I/O 관련 테스트를 지원하지 않아 몇 Test Case들이 Disabled 상태로 진행되었습니다. 그 외의 코드에 대해서도 Coverage 수치가 낮게 측정되었습니다.

### 4.3 Summary

- ⇒ 1<sup>st</sup> Cycle에서 리뷰 했던 '전체적인 문서의 일관성'에 대하여 수정 작업이 잘 진행되었다고 판단되며, 그로 인해 문서의 일관성 및 통일성이 매우 높아졌음
  
- ⇒ 어떤 Case들에 대해서는, 따로 제시하지 않았던 부분에 대해서도 꼼꼼하게 수정 작업이 이루어진 것을 확인할 수 있었음
  
- ⇒ System Test의 경우, 코드 구성이 체계적으로 잘 구성되어 전체적으로 높은 성공률을 보이고 있음. Category, Pairwise는 100%라는 완벽한 테스트 결과가 나왔으며 Brute Force에서 발견한 몇몇 오류만 수정한다면 완성도 높은 프로그램이 예상됨
  
- ⇒ Static Analysis에서는, 그 결과로 **Code Refactoring** 작업과 **Junit Test case Refactoring** 작업이 요구됨