

System Test Report for Team 6

2nd

Project Team
Software Verification Team 1

Latest update on:
2019-05-26

Team Information

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

Table of Contents

1	Specification Review	3
1.1	Rejected Proposal Report	3
1.1.1	Stage 1000 Planning	3
1.1.2	Stage 2030 Analysis	4
1.1.3	Stage 2040 Design	6
1.1.4	Stage 2050 Planning	8
2	System Testing	9
2.1	Category-Partition Testing Report	9
2.1.1	Categorize	9
2.1.2	Description	14
2.1.3	Property & Constraints	19
2.1.4	Text File for TSL-Generator	24
2.1.5	Test Result	27
2.2	Pairwise Testing Report	29
2.2.1	Test Result	29
2.3	Brute Force Testing Report	30
2.3.1	Testing Result	30
2.3.2	Failed Case Report	31
3	Static Analysis	32
3.1	SonarQube Rule	32
3.2	SonarQube Result	33
3.3	SonarQube Usage	37
3.4	Conclution	39
4	Overall	40
4.1	System Test Result	40
4.2	Static Analysis Result	40
4.3	Summary	41

1 Specification Review

1.1 Rejected Proposal Report

1.1.1 Stage 1000 Planning

Activity 1001. Define Draft Plan

4. Non-Functional Requirements

- 시계에서 각 기능 별 버튼들의 조작은 통일성이 있어야 한다.
- 사용자가 쉽게 이해할 수 있도록 시계의 UI가 복잡하지 않고 실행가능한 기능, 시/분/초 등 꼭 필요한 정보들만 보여줘야한다.

⇒ 검증팀의 의견이 반영됨.

Activity 1003. Define Requirements

4. Performance Requirements

각 버튼에 의한 요청에 대해 시계는 즉각적으로 요청에 해당하는 기능을 실행한다.

⇒ 여전히 기준이 명확하지는 않지만 Critical 하지 않음.

Activity 1009. Define System Test Case

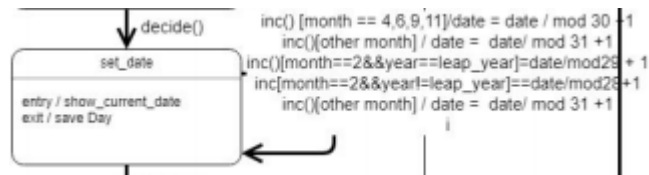
8	목록선택시험	기능을 선택할 때 4개의 기능을 선택한 후 기능조합을 완료할 수 있는지, 실행조합목록에 넣어졌는지 확인	8. select_function	Evident	R.3.3.2
---	--------	---	--------------------	---------	---------

⇒ 검증팀의 의견이 반영됨.

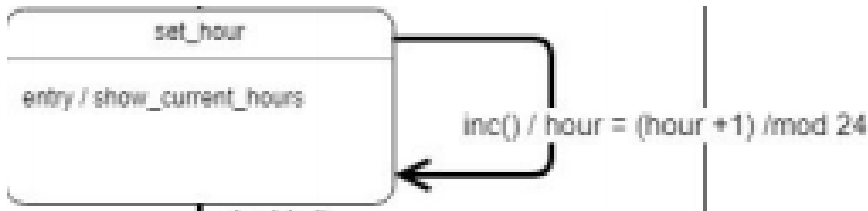
1.1.2 Stage 2030 Planning

Activity 2037. Define State Diagram

1. Current Time

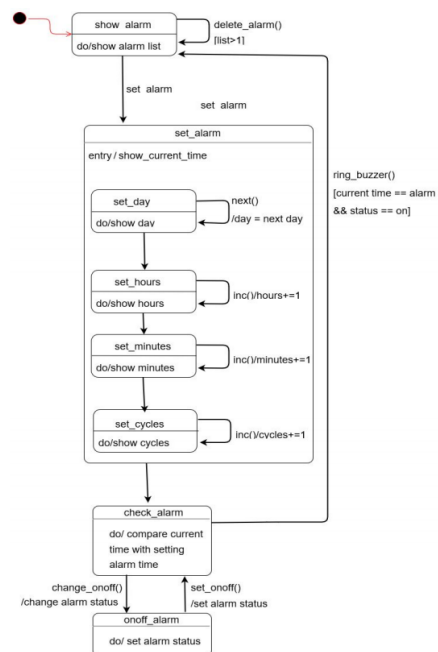


⇒ 검증팀의 의견이 반영됨.



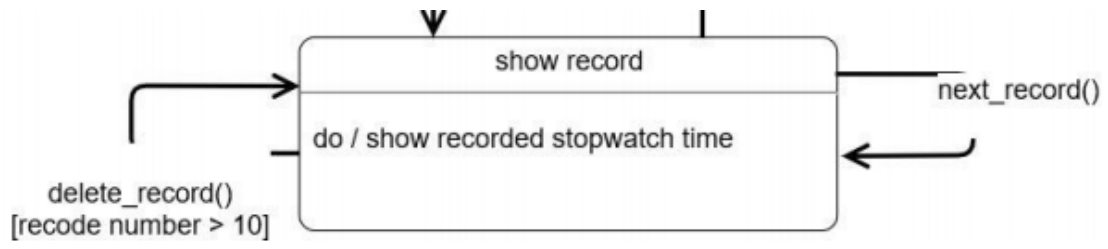
⇒ 검증팀의 의견이 반영됨.

3. Alarm



⇒ 검증팀의 의견이 반영됨

4. Stopwatch



⇒ 수정이 안 됨.

1.1.3 Stage 2040 Planning

Activity 2141. Design Real Use Cases

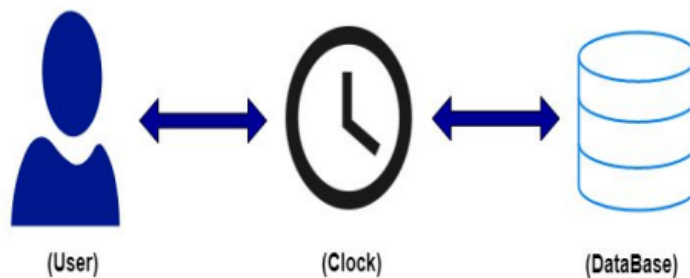
- TimeKeeping function

- Stage 2030

Use Case	set_time
Actor	user/system
Purpose	현재 날짜/시/분 을 설정한다.
Overview	User는 날짜와 시간을 설정한다. 분 단위 까지의 설정을 마치면, 시스템은 User가 설정한 시간을 저장한다.
Type	Evident
Cross Reference	Functions : R.4.1 / Use Cases : look_time
Pre-Requisites	현재 상태가 시간 보기 상태여야한다.
Typical Courses of Event	<div style="border: 1px solid gray; padding: 5px;"> (A): Actor, (S):System 1.(A) User가 버튼을 통해 현재 시각(연도, 월, 일, 시, 분,초)선택을 요청한다. 2.(S) 화면에 시각을 보여준다. 3.(A) User가 버튼을 통해 현재 시각(연도, 월, 일, 시, 분, 초)을 설정한다. 4.(S) 현재 날짜에 해당하는 요일을 계산한다. 5.(S) 설정한 시, 분, 초를 현재 시각으로 저장한다. 6.(A) 시간 설정 종료를 요청한다. 7.(S) look_time상태로 전환한다. </div>

⇒ 검증팀의 의견이 반영 됨.

Activity 2143. Refine System Architecture



⇒ Database를 Stage 2030에 추가 함.

┌ Activity 2144. Define Interaction Diagram

└ Activity 2145. Define Design Class Diagram

1. [state == timekeeping]

⇒ status로 용어 통일

1.1.4 Stage 2050 Planning

3. Methods Description

Use cases : look_time, set_time

⇒ Use case 로 수정 됨.

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
		fitness	1021
		set fitness	1022
Actions	Timekeeping	go to next function	2011
	Set Timekeeping	set time	2021
	Timer	count down	2031
		pause timer	2032

		restart timer	2033
		cancel timer	2034
		go to next function	2035
	Set Timer	set timer	2041
	Alarm	see alarm	2051
		delete alarm	2052
		turn on alarm	2053
		turn off alarm	2054
		go to next function	2055
	Set Alarm	set one alarm	2061
		set five alarm	2062
	Stopwatch	start stopwatch	2071
		record a time	2072
		record eleven time	2073
		pause stopwatch	2074
		restart stopwatch	2075
		reset stopwatch	2076
		go to next function	2077
	Record Stopwatch	get next record	2081
		get previous record	2082

		delete a record	2083
	Dday	delete dday item	2091
		add dday item	2092
		go to next function	2093
	Set Dday	set dday	2101
		add seven dday item	2102
	Fitness	run fitness	2111
		update info	2112
		pause fitness	2113
		finish fitness	2114
		go to next function	2115
	Set Fitness	select cycling	2121
		select walking	2122
		select running	2123
Check	Timekeeping	check valid time	3011
		check counting real time	3012
	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

		ring buzzer	3031
		stop buzzer	3032
	Timer	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
	Alarm	check valid alarm list	3051
		ring buzzer	3052
		stop buzzer	3053
		no ring buzzer	3054
	Set Alarm	day value up one cycle	3061
		period value up one cycle	3062
		hour value up one cycle	3063
		minute value up one cycle	3064
	Stopwatch	check counting up	3071
		check no counting up	3072
	Record Stopwatch	check valid record list	3081
	Dday	check valid dday list	3091

		check value up after one day	3092
		check value up after one month	3093
		check value up after one year	3094
	Set Dday	year value up one cycle	3101
		month value up one cycle	3102
		day value up one cycle	3103
	Fitness	check calculating info	3111
		check no calculating info	3112
		check save other info	3113
	Set Fitness	check valid exercise list	3121
		check valid cycling list	3122
		check valid walking list	3123
		check valid running list	3124
	Next	check valid premode list	3131
Select Function	Set 3 function	set 3 Function	4011
	Set not 3 function	set [0-2] [4-6] function	4012

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.
fitness	운동량 측정 기능이 올바르게 작동하는지 확인한다.
set fitness	운동 설정 기능이 올바르게 작동하는지 확인한다.
go to next function	다음 기능으로 이동 시 결과를 확인한다.
set time	시간 설정 결과를 확인한다.
count down	타이머 기능에서 타이머가 작동하는 것을 확인한다.
pause timer	타이머 중지 결과를 확인한다.
restart timer	타이머 재시작 결과를 확인한다.
cancel timer	타이머 취소 결과를 확인한다.

go to next function	다음 기능으로 이동 시 결과를 확인한다.
set timer	타이머 설정 시 결과를 확인한다.
see alarm	알람이 올바르게 표시되는지 확인한다.
delete alarm	알람 제거가 올바르게 작동하는지 확인한다.
turn on alarm	알람을 활성화했을 경우 올바르게 작동하는지 확인한다.
turn off alarm	알람을 비활성화했을 경우 올바르게 작동하는지 확인한다.
go to next function	다음 기능으로 이동 시 결과를 확인한다.
set one alarm	알람을 하나 만들었을 경우 결과를 확인한다.
set five alarm	알람을 다섯 개 만들었을 경우 결과를 확인한다.
start stopwatch	스톱워치 시작했을 경우 결과를 확인한다.
record a time	스톱워치 기록 후 결과를 확인한다.
record eleven time	스톱워치 기록이 11번 수행했을 때 결과를 확인한다.
pause stopwatch	스톱워치 일시정지 후 결과를 확인한다.
restart stopwatch	스톱워치 재시작 후 결과를 확인한다.
reset stopwatch	스톱워치 리셋 후 결과를 확인한다.
go to next function	다음 기능으로 이동 시 결과를 확인한다.
get next record	스톱워치 다음 기록을 확인한다.
get previous record	스톱워치 이전 기록을 확인한다.
delete a record	스톱워치 기록을 제거한 후 결과를 확인한다.
delete dday item	D+Day 일정을 제거한 후 결과를 확인한다.

add dday item	D+Day 일정을 추가한 후 결과를 확인한다.
go to next function	다음 기능으로 이동 시 결과를 확인한다.
set dday	D+Day 일정을 설정한 후 결과를 확인한다.
add seven dday item	D+Day 일정을 7개 추가한 후 결과를 확인한다.
run fitness	운동량 측정을 시작한 후 결과를 확인한다.
update info	운동량 측정이 실시간으로 업데이트되는 결과를 확인한다.
pause fitness	운동량 측정을 일시정지 후 결과를 확인한다.
finish fitness	운동량 측정을 마친 뒤 결과를 확인한다.
go to next function	다음 기능으로 이동 시 결과를 확인한다.
select cyding	운동 종목을 자전거로 설정한 후 결과를 확인한다.
select walking	운동 종목을 걷기로 설정한 후 결과를 확인한다.
select running	운동 종목을 달리기로 설정한 후 결과를 확인한다.
check valid time	유효한 시간 값인지 검사한다.
check counting real time	실시간으로 시간표시가 이루어지는지 검사한다.
year value up one cyde	‘년도’ 값의 임계값을 검사한다.
month value up one cyde	‘월’ 값의 임계값을 검사한다.
day value up one cyde	‘일’ 값의 임계값을 검사한다.
hour value up one cycle	‘시간’ 값의 임계값을 검사한다.
minute value up one cycle	‘분’ 값의 임계값을 검사한다.
ring buzzer	버저가 울리는지 검사한다.

stop buzzer	버저가 멈추는지 검사한다.
no ring buzzer	버저가 울리지 않는지 검사한다.
check counting down	초읽기가 작동하는지 검사한다.
check no counting down	초읽기가 작동하지 않는지 검사한다.
hour value up one cycle	‘시간’ 값의 임계값을 검사한다.
minute value up one cycle	‘분’ 값의 임계값을 검사한다.
second value up one cycle	‘초’ 값의 임계값을 검사한다.
check valid alarm list	알람 리스트가 유효한지 검사한다.
ring buzzer	버저가 울리는지 검사한다.
stop buzzer	버저가 멈추는지 검사한다.
no ring buzzer	버저가 울리지 않는지 검사한다.
day value up one cycle	‘일’ 값의 임계값을 검사한다.
period value up one cycle	‘주기’ 값의 임계값을 검사한다.
hour value up one cycle	‘시간’ 값의 임계값을 검사한다.
minute value up one cycle	‘분’ 값의 임계값을 검사한다.
check counting up	스톱워치 초읽기 작동을 검사한다.
check no counting up	스톱워치 초읽기 비작동을 검사한다.
check valid record list	스톱워치 기록 리스트가 유효한지 검사한다.
check valid dday list	D+Day 리스트가 유효한지 검사한다.
check value up after one day	1일 이후 D+Day 표시가 유효한지 검사한다.

check value up after one month	1개월 이후 D+Day 표시가 유효한지 검사한다.
check value up after one year	1년 이후 D+Day 표시가 유효한지 검사한다.
year value up one cyde	'년도' 값의 임계값을 검사한다.
month value up one cyde	'월' 값의 임계값을 검사한다.
day value up one cyde	'일' 값의 임계값을 검사한다.
check calculating info	실시간 운동량 계산 작동을 검사한다.
check no calculating info	실시간 운동량 계산 비작동을 검사한다.
check save other info	운동 종목 변경시 이전 운동량 정보가 저장되는지 검사한다.
check valid exercise list	운동 종목 리스트가 유효한지 검사한다.
check valid cycling info	자전거 운동량 리스트가 유효한지 검사한다.
check valid walking info	걷기 운동량 리스트가 유효한지 검사한다.
check valid running info	달리기 운동량 리스트가 유효한지 검사한다.
check valid premode info	이전 모드의 값이 유효한지 검사한다.

2.1.3 Property & Constraints

Category	Values	Property / Constraints
Selected 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]
	fitness	[property Fitness]
	set fitness	[property FitnessSet]
Timekeeping	go to next function	[if (Timekeeping)] [property Next]
Set Timekeeping	set time	[if (TimekeepingSet)]
Timer	count down	[if (Timer)] [property TimerOn]
	pause timer	[if (Timer)] [property TimerOff]
	restart timer	[if (Timer)] [property TimerOn]
	cancel timer	[if (Timer)] [property TimerOff]

	go to next function	[if (Timer)] [property Next]
Set Timer	set timer	[if (TimerSet)]
Alarm	see alarm	[if (Alarm)]
	delete alarm	[if (Alarm)]
	turn on alarm	[if (Alarm)] [property AlarmOn]
	turn off alarm	[if (Alarm)] [property AlarmOff]
	go to next function	[if (Alarm)] [property Next]
Set Alarm	set one alarm	[if (AlarmSet)]
	set five alarm	[error] [if (AlarmSet)]
Stopwatch	start stopwatch	[if (Stopwatch)] [property StopwatchOn]
	record a time	[if (Stopwatch)]
	record eleven time	[error] [if (Stopwatch)]
	pause stopwatch	[if (Stopwatch)] [property StopwatchOff]
	restart stopwatch	[if (Stopwatch)] [property StopwatchOn]
	reset stopwatch	[if (Stopwatch)] [property StopwatchOff]
	go to next function	[if (Stopwatch)] [property Next]
Record Stopwatch	get next record	[if (StopwatchRecord)]
	get previous record	[if (StopwatchRecord)]
	delete a record	[if (StopwatchRecord)]
Dday	delete dday item	[if (Dday)]

	add dday item	[if (Dday)]
	go to next function	[if (Dday)] [property Next]
Set Dday	set dday	[if (DdaySet)]
	add seven dday item	[error] [if (DdaySet)]
Fitness	run fitness	[if (Fitness)] [property FitnessOn]
	update info	[if (Fitness)]
	pause fitness	[if (Fitness)] [property FitnessOff]
	finish fitness	[if (Fitness)] [property FitnessOff]
	go to next function	[if (Fitness)] [property Next]
Set Fitness	select cycling	[if (FitnessSet)] [property Cycling]
	select walking	[if (FitnessSet)] [property Walking]
	select running	[if (FitnessSet)] [property Running]
Timekeeping	check valid time	[if (Timekeeping && !Next)]
	check counting real time	[if (Timekeeping && !Next)]
Set Timekeeping	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)]
Timer	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)]
Set Timer	hour value up one cycle	[if (TimerSet)]
	minute value up one cycle	[if (TimerSet)]
	second value up one cycle	[if (TimerSet)]
Alarm	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)]
Set Alarm	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)]
Stopwatch	check counting up	[if (Stopwatch && StopwatchOn && !Next)]
	check no counting up	[if (Stopwatch && StopwatchOff && !Next)]
Record Stopwatch	check valid record list	[if (StopwatchRecord)]
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)]
Set Dday	year value up one cycle	[if (DdaySet)]
	month value up one cycle	[if (DdaySet)]
	day value up one cycle	[if (DdaySet)]
Fitness	check calculating info	[if (Fitness && FitnessOn && !Next)]
	check no calculating info	[if (Fitness && FitnessOff && !Next)]
	check save other info	[if (Fitness && !Next)]
Set Fitness	check valid exercise list	[if (FitnessSet)]
	check valid cyding list	[if (FitnessSet && Cycling)]
	check valid walking list	[if (FitnessSet && Walking)]
	check valid running list	[if (FitnessSet && Running)]
Next	check valid pre mode list	[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  #Fitness:
18    fitness.                  [property Fitness]
19    set fitness.              [property FitnessSet]
20
21 Do:
22   #Timekeeping:
23     go to next function.       [if (Timekeeping)] [property Next]
24   #TimekeepingSet:
25     set time.                 [if (TimekeepingSet)]
26
27   #Timer:
28     count down.               [if (Timer)] [property TimerOn]
29     pause timer.              [if (Timer)] [property TimerOff]
30     restart timer.            [if (Timer)] [property TimerOn]
31     cancel timer.             [if (Timer)] [property TimerOff]
32     go to next function.      [if (Timer)] [property Next]
33   #TimerSet:
34     set timer.                [if (TimerSet)]
35
36   #Alarm:
37     see alarm.                [if (Alarm)]
38     delete alarm.             [if (Alarm)]
39     turn on alarm.            [if (Alarm)] [property AlarmOn]
40     turn off alarm.           [if (Alarm)] [property AlarmOff]
41     go to next function.      [if (Alarm)] [property Next]
42   #AlarmSet:
43     set one alarm.            [if (AlarmSet)]
44     set five alarm.           [error] [if (AlarmSet)]
45
46   #Stopwatch:
47     start stopwatch.          [if (Stopwatch)] [property StopwatchOn]
```



```

48     record a time.                [if (Stopwatch)]
49     record eleven times.          [error] [if (Stopwatch)]
50     pause stopwatch.              [if (Stopwatch)] [property StopwatchOff]
51     restart stopwatch.            [if (Stopwatch)] [property StopwatchOn]
52     reset stopwatch.              [if (Stopwatch)] [property StopwatchOff]
53     go to next function.          [if (Stopwatch)] [property Next]
54 #StopwatchRecord:
55     get next record.              [if (StopwatchRecord)]
56     get previous record.          [if (StopwatchRecord)]
57     delete a record.              [if (StopwatchRecord)]
58
59 #Dday:
60     delete dday item.             [if (Dday)]
61     go to next function.          [if (Dday)] [property Next]
62 #DdaySet:
63     add dday item.                [if (DdaySet)]
64     add seven dday item.          [error] [if (DdaySet)]
65
66 #Fitness:
67     run fitness.                  [if (Fitness)] [property FitnessOn]
68     update info.                  [if (Fitness)]
69     pause fitness.                [if (Fitness)] [property FitnessOff]
70     finish fitness.               [if (Fitness)] [property FitnessOff]
71     go to next function.          [if (Fitness)] [property Next]
72 #FitnessSet:
73     select cycling.               [if (FitnessSet)] [property Cycling]
74     select walking.               [if (FitnessSet)] [property Walking]
75     select running.               [if (FitnessSet)] [property Running]
76
77 Check Screen:
78 #Timekeeping:
79     check valid time.              [if (Timekeeping && !Next)]
80     check counting real time.      [if (Timekeeping && !Next)]
81 #TimekeepingSet:
82     year value up one cycle.       [if (TimekeepingSet)]
83     month value up one cycle.      [if (TimekeepingSet)]
84     day value up one cycle.        [if (TimekeepingSet)]
85     hour value up one cycle.       [if (TimekeepingSet)]
86     minute value up one cycle.     [if (TimekeepingSet)]
87
88 #Timer:
89     ring buzzer.                   [if (Timer && TimerOn && !Next)]
90     stop buzzer.                   [if (Timer && TimerOn && !Next)]
91     no ring buzzer.                [if (Timer && TimerOff && !Next)]
92
93     check counting down.            [if (Timer && TimerOn && !Next)]
94     check no counting down.        [if (Timer && TimerOff && !Next)]
95 #TimerSet:
96     hour value up one cycle.       [if (TimerSet)]
97     minute value up one cycle.     [if (TimerSet)]
98     second value up one cycle.     [if (TimerSet)]

```

```

99
100 #Alarm:
101     check valid alarm list.           [if (Alarm && !Next)]
102     ring buzzer.                     [if (Alarm && AlarmOn && !Next)]
103     stop buzzer.                     [if (Alarm && AlarmOn && !Next)]
104     no ring buzzer.                  [if (Alarm && AlarmOff && !Next)]
105 #AlarmSet:
106     day value up one cycle.          [if (AlarmSet)]
107     period value up one cycle.       [if (AlarmSet)]
108     hour value up one cycle.         [if (AlarmSet)]
109     minute value up one cycle.       [if (AlarmSet)]
110
111 #Stopwatch:
112     check counting up.               [if (Stopwatch && StopwatchOn && !Next)]
113     check no counting up.           [if (Stopwatch && StopwatchOff && !Next)]
114 #StopwatchRecord:
115     check valid record list.        [if (StopwatchRecord)]
116
117 #DDay:
118     check valid dday list.           [if (Dday && !Next)]
119     check value up after one day.    [if (Dday && !Next)]
120     check value up after one month.  [if (Dday && !Next)]
121     check value up after one year.   [if (Dday && !Next)]
122 #DDaySet:
123     year value up one cycle.         [if (DdaySet)]
124     month value up one cycle.        [if (DdaySet)]
125     day value up one cycle.          [if (DdaySet)]
126
127 #Fitness:
128     check calculating info.          [if (Fitness && FitnessOn && !Next)]
129     check no calculating info.       [if (Fitness && FitnessOff && !Next)]
130     check save other info.          [if (Fitness && !Next)]
131 #FitnessSet:
132     check valid exercise list.       [if (FitnessSet)]
133     check valid cycling info.        [if (FitnessSet && Cycling)]
134     check valid walking info.        [if (FitnessSet && Walking)]
135     check valid running info.        [if (FitnessSet && Running)]
136
137 #Next:
138     check save pre mode info.        [if (Next)]
139

```

2.1.5 Test Result

TestCase	Result	Description
Test01 - 2062	Pass	
Test02 - 2073	Pass	
Test03 - 2102	Pass	
Test04 - 1011.2011.3131	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	Fail	Timer 작동 안 됨.
Test13 - 1013.2032.3033	Fail	Timer 작동 안 됨.
Test14 - 1013.2032.3035	Fail	Timer 작동 안 됨.
Test15 - 1013.2033.3043	Fail	Timer 작동 안 됨.
Test16 - 1013.2033.3032	Fail	Timer 작동 안 됨.
Test17 - 1013.2033.3034	Fail	Timer 작동 안 됨.
Test18 - 1013.2034.3033	Fail	Timer 작동 안 됨.
Test19 - 1013.2034.3035	Fail	Timer 작동 안 됨.
Test20 - 1013.2035.3131	Fail	Timer 작동 안 됨.
Test21 - 1014.2041.3041	Fail	Timer 작동 안 됨.
Test22 - 1014.2041.3042	Pass	
Test23 - 1014.2041.3043	Pass	
Test24 - 1015.2051.3051	Pass	
Test25 - 1015.2052.3051	Pass	
Test26 - 1015.2053.3051	Pass	
Test27 - 1015.2053.3052	Pass	
Test28 - 1015.2053.3053	Pass	
Test29 - 1015.2054.3051	Fail	초기 상태에서는 buzzer가 울리나, 한 번 알람을 끄거나 off 후 on을 한 뒤 buzzer가 울리지 않음.
Test30 - 1015.2054.3054	Pass	
Test31 - 1015.2055.3131	Pass	
Test32 - 1016.2061.3063	Pass	
Test33 - 1016.2061.3062	Pass	
Test34 - 1016.2061.3063	Pass	
Test35 - 1016.2061.3064	Pass	
Test36 - 1017.2071.3071	Pass	
Test37 - 1017.2072.0	Pass	
Test38 - 1017.2074.3072	Pass	
Test39 - 1017.2075.3071	Pass	
Test40 - 1017.2076.3072	Pass	
Test41 - 1017.2077.3131	Pass	
Test42 - 1018.2081.3081	Pass	
Test43 - 1018.2082.3081	Pass	
Test44 - 1018.2083.3081	Pass	

Test45 - 1019.2091.3091	Pass	
Test46 - 1019.2091.3092	Pass	
Test47 - 1019.2091.3093	Pass	
Test48 - 1019.2091.3094	Pass	
Test49 - 1019.2092.3091	Pass	
Test50 - 1019.2092.3092	Pass	
Test51 - 1019.2092.3093	Pass	
Test52 - 1019.2092.3094	Pass	
Test53 - 1019.2093.3131	Pass	
Test54 - 1020.2101.3101	Pass	
Test55 - 1020.2101.3102	Pass	
Test56 - 1020.2101.3103	Pass	
Test57 - 1021.2111.3111	Pass	
Test58 - 1021.2111.3113	Pass	
Test59 - 1021.2112.3113	Pass	
Test60 - 1021.2113.3112	Pass	
Test61 - 1021.2113.3113	Pass	
Test62 - 1021.2114.3112	Pass	
Test63 - 1021.2114.3113	Pass	
Test64 - 1021.2115.3131	Pass	
Test65 - 1022.2121.3121	Pass	
Test66 - 1022.2121.3122	Pass	
Test67 - 1022.2122.3121	Pass	
Test68 - 1022.2122.3123	Pass	
Test69 - 1022.2123.3121	Pass	
Test70 - 1022.2123.3124	Pass	
Test71 - 4011	Pass	
Test72 - 4012	Pass	
전체 Test Case 72개	Pass : 61개	Pass Percentage 84.7%

2.2 Pairwise Testing Report

2.2.1 Test Result

1	Alarm	stoped	stoped	list	194	PASS
5	Alarm	running	stoped	off	140	PASS
6	Alarm	running	stoped	add	137	PASS
20	Alarm	stoped	ringing	on	87	PASS
21	Alarm	running	ringing	nextFunction	86	PASS
22	DDay	running	ringing	list	84	PASS
23	DDay	running	stoped	nextFunction	83	PASS
2	Fitness	stoped	stoped	finish	169	PASS
9	Fitness	running	ringing	select	119	PASS
16	Fitness	running	stoped	update	95	PASS
17	Fitness	stoped	stoped	run	93	PASS
18	Fitness	stoped	stoped	select	89	PASS
24	Fitness	running	stoped	pause	82	PASS
25	Fitness	running	stoped	list	81	PASS
26	Fitness	running	stoped	nextFunction	80	PASS
4	Stopwatch	running	stoped	pause	153	PASS
7	Stopwatch	running	ringing	list	131	PASS
8	Stopwatch	stoped	ringing	nextFunction	128	PASS
13	Stopwatch	running	stoped	record	103	PASS
14	Stopwatch	stoped	stoped	start	101	PASS
15	Stopwatch	stoped	ringing	reset	99	PASS
3	TimeKeeping	running	stoped	nextFunction	161	PASS
10	TimeKeeping	running	stoped	setTime	117	PASS
11	Timer	stoped	stoped	on	113	FAIL
12	Timer	stoped	ringing	restart	111	FAIL
19	Timer	running	ringing	nextFunction	88	FAIL
		TC NUM	PASS	FAIL	PERCENTAGE	
		26	22	3	84.62%	

⇒ Testcover이라는 웹툴을 사용하여 Pairwise Test를 진행하였습니다.

2.3 Brute Force Testing Report

2.3.1 Testing Result

Test	#	Description	P/F
Action	2-1	Timer/Stopwatch에서 count 해놓고 다른 모드 변경 후 다시 갔을 때 Timer의 동작 하는지 검사	Fail
	2-2	Calculate_calories에서 23:59에서 24:00 넘어간 후 운동을 끝냈을 때 해당 날짜로 저장을 하는지 검사	Pass
	2-3	D+day에서 년도의 임계치를 설정했는지 검사	Pass
	2-4	Fitness에서 Default 값으로 했을 때 작동하지 않는지 검사	Pass
Check	3-1	Timer와 Alarm(주기 설정)의 부저 울리는 시간을 맞춘 후 어떻게 동작하는지 Test	Fail
	3-2	D+day에서 하나 설정 해놓고 setTimeKeeping을 통해 시간 바꿨을 때 D+day 계산이 정상적으로 바뀌는지 검사	Fail
	3-3	Fitness에서 1분 단위 업데이트 직전에 운동 종목을 바꿨을 때 Calories 계산을 제대로 하는지 검사	Fail

2.3.2 Failed Case Report

Test	#	Description
Action	2-1	Timer가 동작 안함.
Check	3-1	Timer가 동작 안함.
	3-2	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;">1번째 목표 : save money</div> <div style="border: 1px solid black; padding: 5px;">1번째 D+day 값: +-1</div>

		미래의 날짜에 대해 -1이 됨.				
	3-3	<p>1분 단위가 되기 전에 운동을 finish하면 값이 업데이트 되지 않음. (running을 30초씩 2번 해도 0kcal)</p> <table border="1" data-bbox="497 405 901 566"> <tr> <td data-bbox="497 405 901 465">6월 10일calories: 50</td> </tr> <tr> <td data-bbox="497 465 901 566">00:07:17</td> </tr> </table> <table border="1" data-bbox="497 589 901 750"> <tr> <td data-bbox="497 589 901 649">6월 10일calories: 50</td> </tr> <tr> <td data-bbox="497 649 901 750">00:08:32</td> </tr> </table>	6월 10일calories: 50	00:07:17	6월 10일calories: 50	00:08:32
6월 10일calories: 50						
00:07:17						
6월 10일calories: 50						
00:08:32						

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	Sun's java Style

3.1.4 Code Coverage

소모팀에서 작성한 **junit Test**를 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. Supported only for Java.
------	---------------------------------------	--

Sonarqube Test Coverage docs (<https://docs.sonarqube.org/latest/analysis/coverage/>)

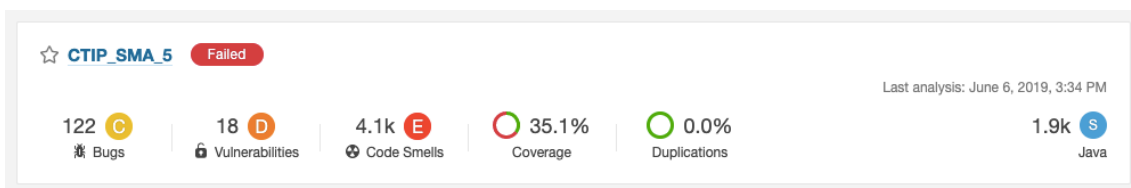
Java	<code>sonar.junit.reportPaths</code>	Comma-delimited list of paths to Surefire XML-format reports.
------	--------------------------------------	---

Sonarqube Test Execution docs (<https://docs.sonarqube.org/latest/analysis/coverage/>)

- ✓ 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 xml파일을 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 allocates 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		multi-threading	Window sets size manually, and doesn't use pack	1
18			Inconsistent synchronization	9
19			Method calls wait, notify or notifyAll on a Thread instance	8
20			Unconditional wait	4
21			Method calls Thread.sleep() with a lock held	3
22			Unread field	6
23		performance	Method invokes inefficient Number constructor; use static valueOf instead	3
24			Unused field	3
		Method passes constant String of length 1 to character overridden method	2	
25	sonar	cwe, error-handling, multi-threading	"InterruptedException" should not be ignored	10
26		multi-threading	Methods "wait(...)", "notify()" and "notifyAll()" should not be called on Thread instances	8
27		cert, multi-threading, performance	"wait(...)" should be used instead of "Thread.sleep(...)" when a lock is held	6
28		cert, multi-threading,	"notifyAll" should be used	4
29		deadlock, multi-threading	"wait" should not be called when multiple locks are held	3
30		cert	Loops should not be infinite	3
31		based-on-misra, cert, cwe, overflow, sans-top25-risky	Math operands should be cast before assignment	2
32		cert, cwe, denial-of-service, leak	Resources should be dosed	2
33	pmd	no tags	Close Resource	2

3.2.3 Vulnerability

Ref	plug-in	tag	rule	count
1	pmd	security	Method returns internal array	2
2			Array is stored directly	1
3	findbugs	cwe, injection, owasp-a1, wasc	Potential JDBC Injection	15
4	sonar	cwe	Class variable fields should not have public accessibility	12
5		cert, cwe, unpredictable	Mutable members should not be stored or returned directly	5
6		cwe, error-handling, owasp-a3	Throwable.printStackTrace(...) should not be called	31
7		no tags	Member variable visibility should be specified	52

3.2.4 Code Smell

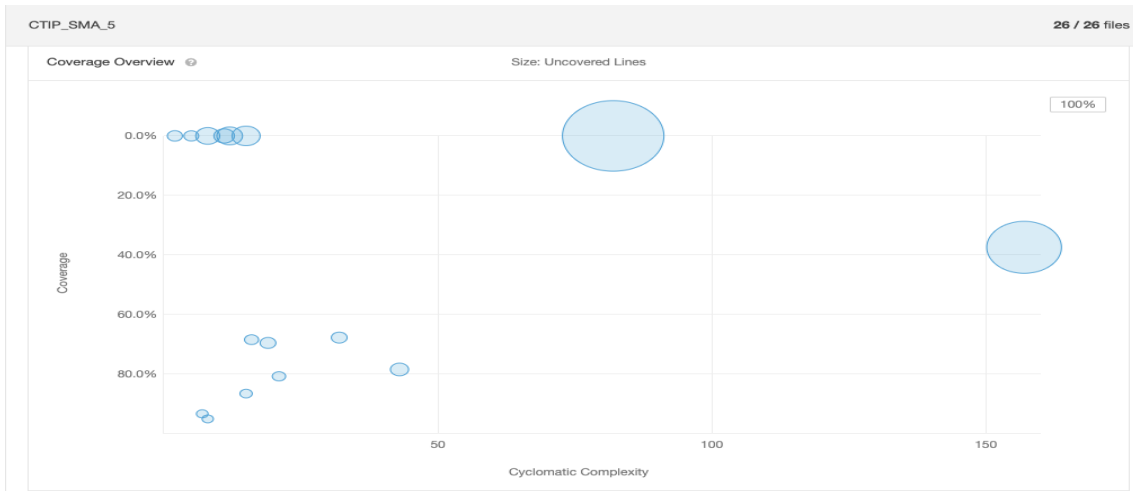
The screenshot displays three panels of code smell analysis results:

- Left Panel:** Lists 20 code smells with their counts. The most frequent is "(Java) Call Super In Constructor" with 13 occurrences. Other notable items include "(Java) Unused Private Field" (13), "(Java) Unused *private* fields should b..." (13), and "(Java) Methods should not have too ma..." (10).
- Middle Panel:** Lists 17 tags with their counts. The most frequent is "java8" with 57 occurrences, followed by "finding" (46) and "performance" (44).
- Right Panel:** Shows a "Tag" filter interface with a search box and a list of 17 tags. The most frequent tag is "convention" with 1.6k occurrences, followed by "style" (1k) and "brain-overload" (691).

Ref	plug-in	tag	rule	count
1	pmd	multithreading	Non Thread Safe Singleton	2
2			Do Not Use Threads	6
3	sonar	multi-threading	Instance methods should not write to "static" fields	6
4	sonar	junit	Literal boolean values should not be used in assertions	13
5	checkstyle	error-handling	Illegal Catch	19
...				

- ✓ 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...

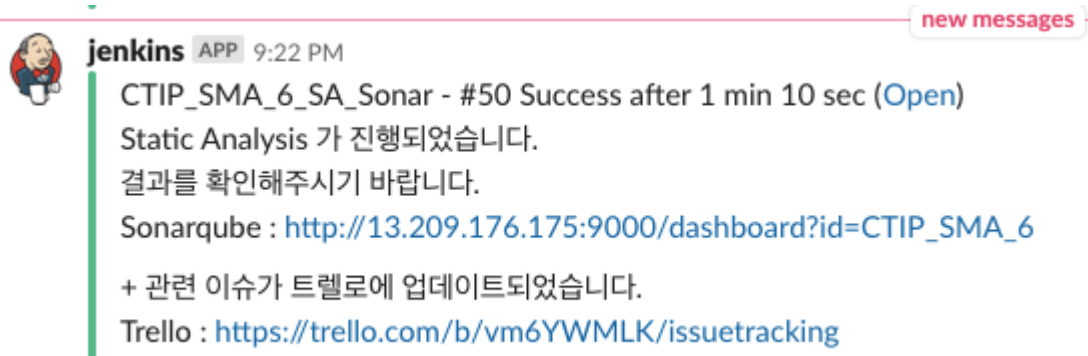
```

Partially covered by tests (1 of 2 conditions).

Not covered

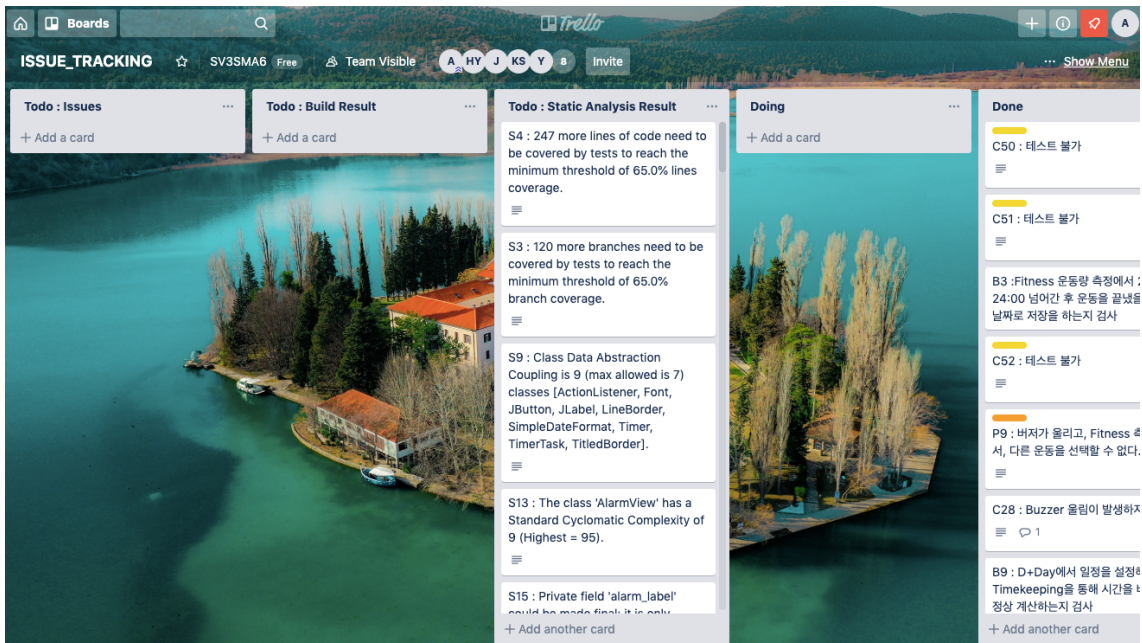
3.3 SonarQube Usage

- ✓ Slack을 통한 Notification 확인



- ✓ Trello Issue Card 확인

- CTIP환경에서 심각도를 기준으로 (Blocked / Critical / Major) 이슈를 자동 등록



✓ Sonarqube를 통해 Issue 확인

SonarQube interface showing 7 issues for file AlarmView.java. The issues include:

- Add a 'package-info.java' file to document the 'src/main/java/watch' package (Code Smell, Minor)
- Add or update the header of this file. (Code Smell, Blocker)
- 20 duplicated blocks of code must be removed. (Code Smell, Major)
- 153 more comment lines need to be written to reach the minimum threshold of 25.0% comment density. (Code Smell, Major)
- 120 more branches need to be covered by tests to reach the minimum threshold of 65.0% branch coverage. (Code Smell, Major)
- 247 more lines of code need to be covered by tests to reach the minimum threshold of 65.0% lines coverage. (Code Smell, Major)

✓ 해당 소스 코드로 이동

SonarQube interface showing 8 issues for file AlarmView.java. The issues include:

- Assign this magic number 500 to a well-named constant, and use the constant instead. (Code Smell, Major)
- Assign this magic number 100 to a well-named constant, and use the constant instead. (Code Smell, Major)
- Assign this magic number 300 to a well-named constant, and use the constant instead. (Code Smell, Major)
- This block of commented-out lines of code should be removed. (Code Smell, Major)

```

86
87
88     tmp = new JLabel();
89     tmp.setBorder(new TitledBorder(new LineBorder(Color.BLACK)));
90     tmp.setBounds(100, 100, 300, 300);
91
92
93
94     LCD1 = new JLabel();
95     LCD1.setBounds(200, 150, 20, 20);
    
```

- ✓ 위배 규칙 및 패턴을 확인한다.



Noncompliant Code Example

```
public static void doSomething() {
    for(int i = 0; i < 4; i++){
        // Noncompliant, 4 is a magic number
        ...
    }
}
```

Compliant Solution

```
public static final int NUMBER_OF_CYCLES = 4;
public static void doSomething() {
    for(int i = 0; i < NUMBER_OF_CYCLES ; i++){
        ...
    }
}
```

- ✓ 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

4.2.1 Category-Partition Test

많은 기능이 구현되었고 예외처리나 세부적인 것들까지 Spec대로 잘 구현했습니다. 타이머의 기능이 정상적으로 동작하지 않은 것만 구현이 된다면 더 좋은 시스템 테스트 결과를 기대 할 수 있습니다.

4.2.2 Pairwise Test

Pairwise Test의 경우, 첫 번째 보고서의 말대로 기능이 구현되면서 자연스럽게 좋은 결과가 나왔습니다. 타이머의 기능만 구현한다면 100% 성공률이 기대됩니다.

4.2.3 Brute Force Test

검증팀에서 정상적으로 동작한다고 판단한 기능들에 대해 특수한 요구에 예외처리나 Specification에 명시한 것처럼 동작을 하는지 Test 한 것이기 때문에 비교적 실패율이 높았습니다.

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, GUI, 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. 문서의 일관성이 더 좋아졌습니다.
2. 문서와 실제 구현된 코드 간 차이가 많이 줄었습니다.
3. 아직 기본 기능이 구현되지 않은 부분이 있습니다.
4. UI와 Manual이 많이 달라 Usability는 여전히 떨어집니다.
5. Static Analysis 결과, **Code Refactoring** 작업이 요구됩니다.
6. **Junit Test Case Refactoring** 작업이 요구됩니다.