2019 Software Modeling & Analysis
OOPT 2nd Cycle
【 Revision 】

골라보 시계

Team #1
201411273 박재범
201411275 박진호
201411311 장원영
201311313 정인원

Contents

1. Incomplete Part Completion

1. Alarm 관련 기능 구현 완료

1) Alarm Class 구현

reqSetAlarm(), reqNextAlarm(), getNextNode(), reqDetailSet()
setAlarmPart(), reqSaveAlarm(), reqDeleteAlarm(), deleteAlarm()
displayAlarmList(), reqStopAlarmBuzzer(), reqSetSnooze(),
createSnooze()

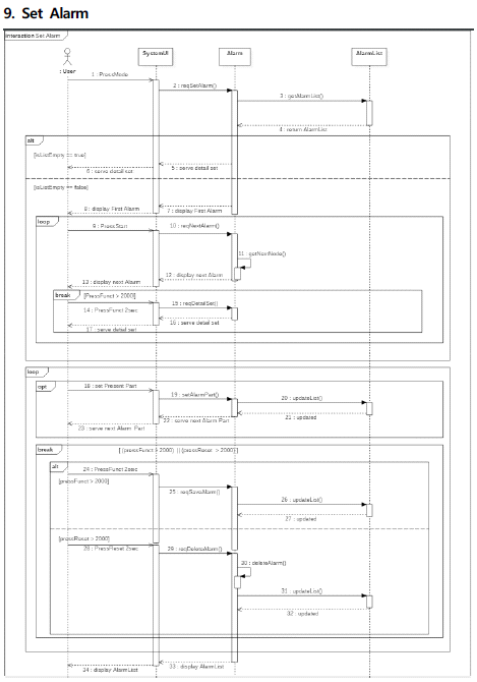
2) AlarmList Class 구현

getAlarmList(), updateList()

3) AlarmUI Class 구현

4) OOPT 2051 Implements Windows에 Alarm 관련 항목 추가

Use Case : Set Alarm, Stop Alarm Buzzer, Set Snooze



Name	1. PressMode
Responsibilities	Mode 버튼을 누른다.
Type	GUI
Cross Reference	R4.1, R4.2
Note	N/A
Pre-Conditions	Alarm 기능을 선택한 상태여야 한다.
Post-Conditions	N/A

Name	6. serve detail set
Responsibilities	선택한 알람의 세부 설정 기능을 제공한다.
Type	GUI
Cross Reference	R4.1, R4.2
Note	N/A
Pre-Conditions	알람 리스트가 비어있는 상태여야 한다.
Post-Conditions	N/A

Name	8. display First Alarm
Responsibilities	알람 리스트의 첫번째 요소를 출력한다.
Type	GUI
Cross Reference	R4.1, R4.2
Note	N/A
Pre-Conditions	알람 리스트에 데이터가 존재해야 한다.
Post-Conditions	N/A

Name	9. PressStart
Responsibilities	Start 버튼을 누른다.
Type	GUI
Cross Reference	R4.1, R4.2
Note	N/A
Pre-Conditions	알람 리스트에 데이터가 존재해야 한다.
Post-Conditions	N/A

Name	13. display next Alarm
Responsibilities	리스트의 다음 알람 요소를 출력한다.
Type	GUI
Cross Reference	R4.1, R4.2
Note	N/A
Pre-Conditions	알람 리스트에 데이터가 존재해야 한다.
Post-Conditions	N/A

Name	14. PressFunct 2sec
Responsibilities	Funct 버튼을 2초간 누른다.
Type	GUI
Cross Reference	R4.1, R4.2
Note	N/A
Pre-Conditions	알람 리스트에 데이터가 존재해야 한다.
Post-Conditions	N/A

2. Select Function 중 기능 On/Off Choice 부분 구현 완료

1) SelectFunction Class 수정

reqNextFunction(), reqSelectFunction(), setFunction()

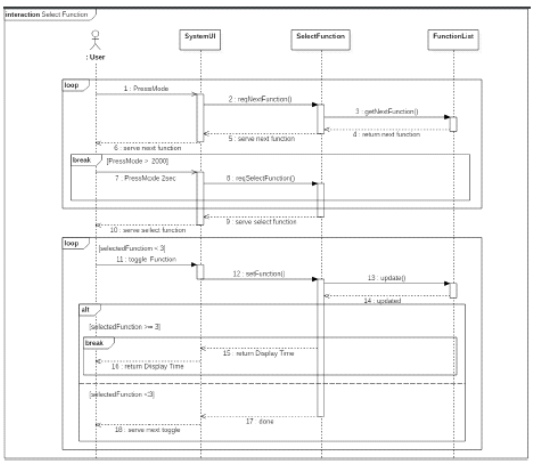
2) FunctionList Class 수정

getNextFunction(), update()

3) SelectFunctionUI Class 수정

4) OOPT 2051 Implements Windows에 Select Function 내용 추가

15. Select Function



Name	7. PressMode 2sec
Responsibilities	Mode 버튼을 2초간 누른다.
Type	GUI
Cross Reference	R7.1
Note	N/A
Pre-Conditions	N/A
Post-Conditions	N/A

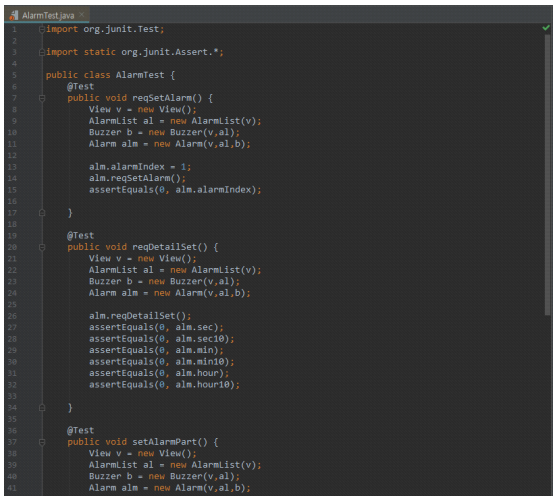
Name	10. serve select function
Responsibilities	기능 선택을 제공한다.
Type	GUI
Cross Reference	R7.1
Note	N/A
Pre-Conditions	N/A
Post-Conditions	N/A

Name	11. toggle Function
Responsibilities	Start, Reset 버튼으로 현재 기능을 On/Off 설정하고 Funct 버튼으로 확인한다.
Type	GUI
Cross Reference	R7.1
Note	N/A
Pre-Conditions	N/A
Post-Conditions	N/A

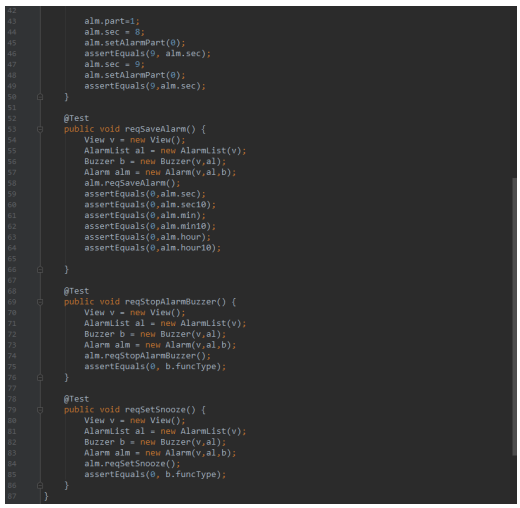
Name	16. return Display Time
Responsibilities	TimeKeeping의 시간 출력 화면으로 돌아간다.
Type	GUI
Cross Reference	R7.1
Note	내부적으로 설정을 저장한다.
Pre-Conditions	기능 3개가 On 되어야 한다.
Post-Conditions	N/A

Name	18. serve next toggle
Responsibilities	다음 기능의 toggle을 제공한다.
Type	GUI
Cross Reference	R7.1
Note	N/A
Pre-Conditions	기능이 3개 미만 On 되어야 한다.
Post-Conditions	N/A

3. Unit Test Code 추가 작성 및 테스팅 진행



```
import org.junit.Test;

import static org.junit.Assert.*;

public class AlarmTest {
    @Test
    public void reqSetAlarm() {
        View v = new View();
        AlarmList al = new AlarmList(v);
        Buzzer b = new Buzzer(v,al);
        Alarm alm = new Alarm(v,al,b);

        alm.alarmIndex = 1;
        alm.reqSetAlarm();
        assertEquals(0, alm.alarmIndex);

    }

    @Test
    public void reqDetailSet() {
        View v = new View();
        AlarmList al = new AlarmList(v);
        Buzzer b = new Buzzer(v,al);
        Alarm alm = new Alarm(v,al,b);

        alm.reqDetailSet();
        assertEquals(0, alm.sec);
        assertEquals(0, alm.sec10);
        assertEquals(0, alm.min);
        assertEquals(0, alm.min10);
        assertEquals(0, alm.hour);
        assertEquals(0, alm.hour10);

    }

    @Test
    public void setAlarmPart() {
        View v = new View();
        AlarmList al = new AlarmList(v);
        Buzzer b = new Buzzer(v,al);
        Alarm alm = new Alarm(v,al,b);
```



```
        alm.part=1;
        alm.sec = 0;
        alm.setAlarmPart(0);
        assertEquals(0, alm.sec);
        alm.sec = 9;
        alm.setAlarmPart(0);
        assertEquals(9,alm.sec);
    }

    @Test
    public void reqSaveAlarm() {
        View v = new View();
        AlarmList al = new AlarmList(v);
        Buzzer b = new Buzzer(v,al);
        Alarm alm = new Alarm(v,al,b);
        alm.reqSaveAlarm();
        assertEquals(0,alm.sec);
        assertEquals(0,alm.sec10);
        assertEquals(0,alm.min);
        assertEquals(0,alm.min10);
        assertEquals(0,alm.hour);
        assertEquals(0,alm.hour10);

    }

    @Test
    public void reqStopAlarmBuzzer() {
        View v = new View();
        AlarmList al = new AlarmList(v);
        Buzzer b = new Buzzer(v,al);
        Alarm alm = new Alarm(v,al,b);
        alm.reqStopAlarmBuzzer();
        assertEquals(0, b.funcType);
    }

    @Test
    public void reqSetSnooze() {
        View v = new View();
        AlarmList al = new AlarmList(v);
        Buzzer b = new Buzzer(v,al);
        Alarm alm = new Alarm(v,al,b);
        alm.reqSetSnooze();
        assertEquals(0, b.funcType);
    }
}
```

AlarmTest: 6 total, 6 passed 343 ms

Collapse | Expand

AlarmTest		343 ms
reqSaveAlarm	passed	213 ms
reqSetSnooze	passed	54 ms
reqDetailSet	passed	21 ms
setAlarmPart	passed	20 ms
reqSetAlarm	passed	15 ms
reqStopAlarmBuzzer	passed	20 ms

SelectFunctionTest: 2 total, 2 passed 305 ms

Collapse | Expand

SelectFunctionTest		305 ms
setFunction	passed	229 ms
reqSelectFunction	passed	76 ms

4. 진행하지 못한 System Test 진행

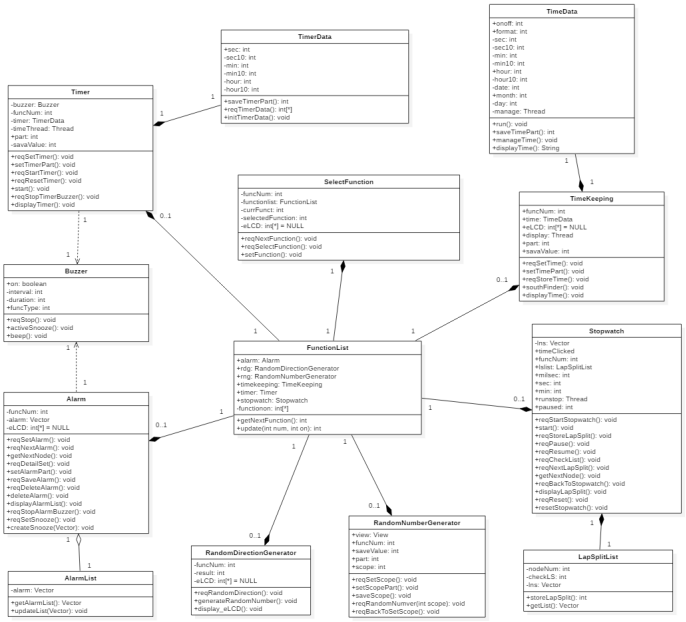
Test #	Test 항목	Description	Use Case	Sys. Func.
1-1	시간 설정 시험 (24h)	- 시간을 24h 포맷 23:59로 설정 - 입력 후 저장되었는지 Test	1. Set Time	R1.1
1-2	시간 설정 시험 (12h)	- 시간을 01:59 AM으로 설정 - 입력 후 저장되었는지 Test	1. Set Time	R1.1
2-1	시간 출력 시험 (24h)	- 24h 포맷 15:59부터 1분간 LCD에 잘 출력되는지 Test - South Finder가 정확한 방향을 가리키고 있는지 Test	2. Display Time	R1.2
2-2	시간 출력 시험 (12h)	- 12h 포맷 15:59 PM부터 1분간 LCD에 잘 출력되는지 Test - South Finder가 정확한 방향을 가리키고 있는지 Test	2. Display Time	R1.2
3-1	타이머 설정 시험	- Timer를 1분 30초로 설정 - 타이머가 설정한 시간부터 잘 작동하는지 Test	3. Set Timer	R2.1
4-1	타이머 버저 시험	- Timer가 0이되면 5초동안 버저를 울리는지 Test	4. Beep Timer Buzzer	R2.2
5-1	타이머 버저 중지 시험	- Timer의 버저가 울릴 때 User가 버튼을 누르면 정지하는지 Test - Timer가 초기화 되는지 Test	5. Stop Timer Buzzer	R2.3
6-1	스톱워치 시작 시험	- Start 버튼을 누를 시 Stopwatch가 제대로 시작하는지 Test	6. Start Stopwatch	R3.1
7-1	랩, 스플릿 시험 (7 Times)	- 진행 중 7번의 Data 저장 요청이 정상적으로 수행 가능한지 Test	7. Store Lap & Split	R3.2
7-2	랩, 스플릿 시험 (35 Times)	- 진행 중 35번의 Data 저장 요청이 정상적으로 수행 가능한지 Test	7. Store Lap & Split	R3.2
8-1	스톱워치 중지 시험	- 진행 중 Start 버튼을 누르면 정지하는지 Test - 정지 상태에서 다시 Start 버튼을 누르면 재개하는지 Test	8. Pause Stopwatch	R3.3
9-1	랩, 스플릿 확인 시험 (7 Times)	- 정지 상태에서 Funct 버튼을 2초간 누르면 Lap, Split 확인으로 넘어가는지 Test - 7-1에서 저장한 7개의 데이터가 저장되어 있는지 Test - 제일 마지막 Lap, Split이 출력된 후 다시 맨 처음 Lap, Split이 출력되는지 Test - Lap, Split 데이터가 없으면 기능이 제대로 블락되는지 Test	9. Check Lap & Split	R3.4

Test #	Test 항목	Description	Use Case	Sys. Func.
15-1	난수 범위 설정 시험	- 범위를 1, 22, 333, 4444, 55555, 999999까지 각각 설정 가능한지 Test	15. Set Scope	R5.1
16-1	난수 생성 시험 (6 Each Times)	- Start 버튼을 2초간 눌러 15-1에서 설정한 범위에 대해서 각각 정상적으로 난수가 생성되는지 Test	16. Generate Random Number	R5.2
16-2	난수 생성 시험 (Repeat)	- 15-1에서 범위를 6으로 설정하고 하나의 범위에 대해서 총 10번 반복하여 난수가 잘 생성되는지 Test	16. Generate Random Number	R5.2
17-1	임의 방향 생성 시험	- Start 버튼을 누르면 60개의 LCD에 랜덤하게 방향이 표시되는지 10번 반복하여 Test	17. Generate Random Direction	R6.1
18-1	기능 선택 시험 (Next Funct)	- Mode 버튼을 눌러 다음 기능으로 정상적으로 넘어가는지 Test	18. Select Function	R7.1
18-2	기능 선택 시험 (Cycle)	- 마지막 기능에서 Mode 버튼을 눌러 다시 처음 기능으로 정상적으로 돌아가는지 Test	18. Select Function	R7.1
18-3	기능 선택 시험 (Setting)	- Mode 버튼을 2초간 눌러 설정 화면으로 넘어가는지 Test - 3개의 기능을 On하고 설정을 저장하면 TimeKeeping으로 전환되는지 Test	18. Select Function	R7.1

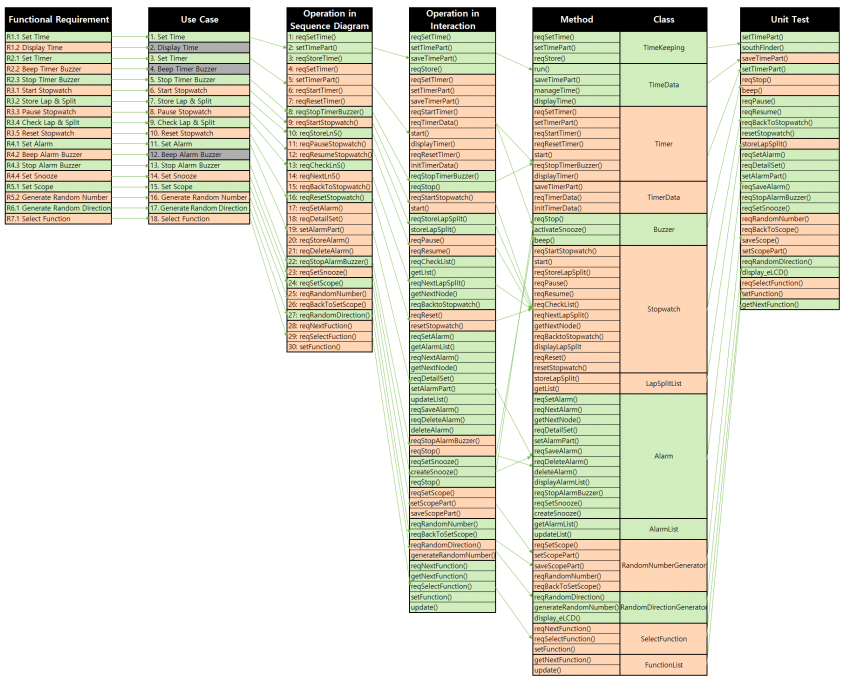
5. 변경된 부분을 반영해 전반적인 보고서 수정

OOTP Stage 1000, 2030, 2040, 2050, 2060

- Class Diagram



- Traceability Analysis



- 이외 기타 세부 사항 변경 완료

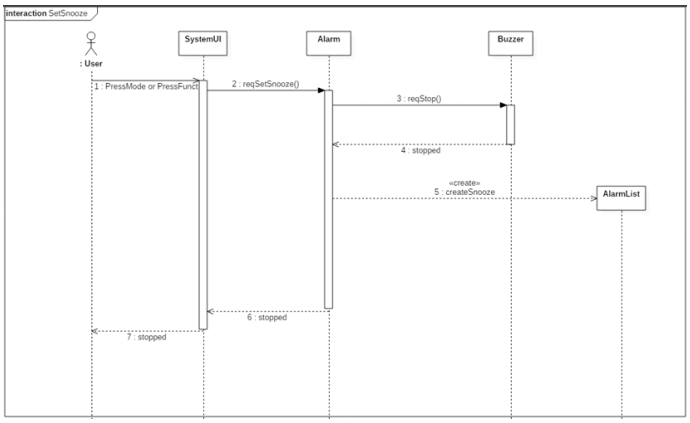
2. Specification Revision

1. 자체적인 변경 사항

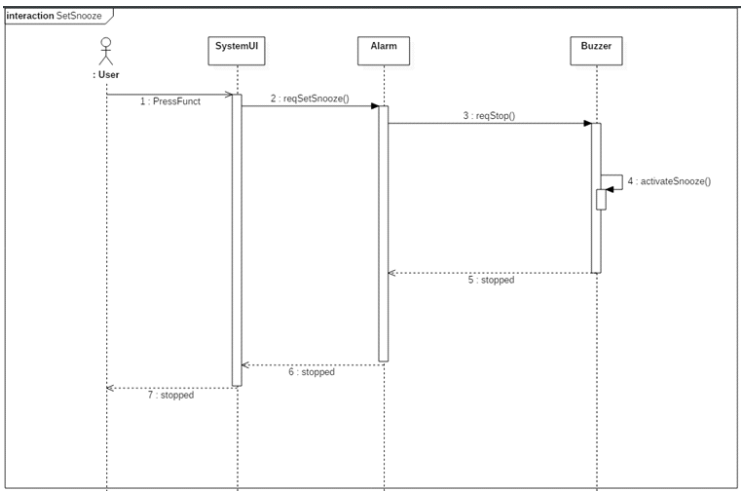
디자인대로 구현하는 데 어려움이 있거나 테스팅 과정에서 개선이 필요하다고 생각한 부분들을 변경

1) Set Snooze 작동 방식 변경

기존 : AlarmList 객체를 임시로 생성하여 스누즈 등록



변경 : Buzzer Class에 activateSnooze() 메소드를 추가해 Buzzer Class 내부적으로 스누즈 관리



2) Buzzer OFF를 위한 조건 변경

Use Case : Stop Timer Buzzer, Stop Alarm Buzzer, Set Snooze

기존 : 어떤 기능에서도 Buzzer를 멈출 수 있다.

Use Case	Set Snooze
Actor	User
Purpose	알람 Snooze를 설정한다.
Overview	User가 **Mode나 Funct버튼**을 눌러 버저를즉시 멈추고 5분 후에 다시 울리게 한다.
Type	Evident
Cross Reference	Functional Requirements ; R.4.4 R.4.2
Pre-Requisites	Alarm의 Buzzer가 울리는 상태여야 한다. **Select Function에서 Alarm이 ON 상태여야 한다.**
Typical Courses Of Events	(A) : Actor, (S) : System 1. (A) : **Mode나 Funct버튼**을누른다. 2. (S) : 버저를즉시 멈추고 5분 후에 다시 울리도록 임시 알람을 설정한다.
Alternative Courses Of Events	N/A
Exceptional Courses Of Events	N/A

변경 : 해당 기능을 선택한 상태에서만 Buzzer를 멈출 수 있다.

ex) Stop Alarm Buzzer와 Set Snooze는 Alarm 기능에서만 수행 가능

Use Case	Set Snooze
Actor	User
Purpose	알람 Snooze를 설정한다.
Overview	User가 **Alarm 기능을 선택한 상태에서** **Funct버튼**을눌러 버저를즉시 멈추고 5분 후에 다시 울리게 한다.
Type	Evident
Cross Reference	Functional Requirements ; R.4.4 R.4.2
Pre-Requisites	Alarm의 Buzzer가 울리는 상태여야 한다. **Alarm 기능을 선택한 상태여야 한다.**
Typical Courses Of Events	(A) : Actor, (S) : System 1. (A) : **Alarm 기능을 선택한 상태에서** **Funct버튼**을 누른다. 2. (S) : 버저를즉시 멈추고 5분 후에 다시 울리도록 임시 알람을 설정한다.
Alternative Courses Of Events	N/A
Exceptional Courses Of Events	N/A

2. Specification Revision

1 Stage 1000

1.1 Stage 1001 (page3) : 2. Objectives
➢ "범용성이 높은 난수 생성 기능, 360 랜덤 방위 출력 기능"
→ 구체화가 필요하다
대응 : 아이디어의 개요를 설명하는 초반부로, 구체적인 디자인 설계 단계가 아니었기 때문에 현상 유지

1.2 Stage 1001 (page4) : 3. Functional Requirements
➢ Random Number Generator
→ 난수에 대한 범위가 언급되어 있지 않음
대응 : 범위에 대한 세부적인 설명 추가

1.3 Stage 1004 (page12) :
7. Describe use cases 2. Display Time
- southfinder : 이름의 합당한지의 여부 (혼동을 준다)
➔ 시침과 12시의 중간이 남향인지의 여부
- 시침이 6시를 가리킨다면 어느 쪽을 가리키는지 여부 -> 구체화
대응 : 남향을 알려주는 것이 목적이므로 SouthFunder라는 명칭은 합당, 알고리즘을 숙지하고 있으면 문제가 없으므로 현상 유지

12. Beep Alarm Buzzer -> 1분동안 버저가 울린 뒤의 프로세스 구체화
대응 : '1분 동안' 버저가 울린다는 표현에서 이미 버저가 1분 울린 뒤 꺼진다는 의미가 내포되어 있으므로 현상 유지

1.4 Stage 1006 (page 18) :

2. Define System Test Plan and Mapping With System Function

12. 알람버저 시험 : "1분 울리는지 test" 이후의 프로세스 구체화

대응 : 이전 사항과 동일

17. 임의 방향 생성 시험 : "See.d" -> 오타 수정 요망

대응 : 오타 수정 완료

2 Stage 2030

2.1 Stage 2031 (page 3) : Set Time

➢ Typical Courses Of Events의 7번의 내용 수정 요망

➔'설정을 마치면' 이 아니라 '모든 과정에서' 가 합당해 보임

대응 : 설정을 마치면 → 설정 중 언제든지 로 표현 변경

2.2 Stage 2031 (page 5) : Stop Timer Buzzer

"Mode 버튼을 제외한 버튼을 누르면 stop" -> 예외처리를 추가해야 한다

대응 : Buzzer는 기능들과 별개의 쓰레드로 동작하므로 별도의 예외처리 필요 없음, 현상 유지

2.3 Stage 2031 (page 9) : Set Alarm

- 알람기능에 대한 설명이 부족하다

➢ 알람 데이터가 있을때, 기존 알람의 세부설정이 아닌 새로운 알람을 추가시 어떤 버튼을 눌러야 하는지 설명이 없다

대응 : Funct 버튼을 2초간 누르면 기존 알람을 수정하는 방식에서 Funct 버튼을 2초간 누르면 새로운 알람을 추가하는 방식으로 알고리즘 변경(보고서 및 다이어그램 반영 완료)

- Exception E10. : 명확하지 않은 설명

➢ "알람기능을 껐다켜도 정보가 삭제되지 않는다"가 명확

- "최대 10까지 메모리에 저장할 수 있어야한다" : 내용추가 요망

대응 : 명확한 표현으로 수정, 10개까지 저장할 수 있다는 내용 일부 누락되어 추가 완료

2.4 Stage 2031 (page 10) : Beep Alarm Buzzer

➢ E1 : Stage 1000에는 없었던 내용임 -> 추가 요망

대응 : 이미 1000 단계에 존재하던 내용이므로 현상 유지

2.5 Stage 2031 (page 13) : Generate Random Direction

- Typical Courses Of Events의 과정을 구체화할 필요가 있어 보임

대응 : 이미 충분히 구체적으로 설명되어 있으므로 현상 유지

2.6 Stage 2032 (page 15) :

2. Assign Class Name into Concepts & Draw a Conceptual Class Diagram

- Time Keeping 과 Time의 구별이 필요해 보임

(이름이 비슷하여 혼동이 오므로 구체화된 이름이 요구된다)

대응 : Time과 Time Keeping은 전체적인 이전 흐름을 통해서 차이를 명확히 구별할 수 있음, 현상 유지

2.7 Stage 2032 (page 16 ~ 18) :

➢ 4번의 "Stopwatch has Time"이 3번에서는 언급되고 있지 않음

대응 : 연결관계 추가 완료

2.8 Stage 2034 (page 35) :

➢ Use Case : Generate Rand. Direct.에서 "Gecerator" 오타 -> 수정 요망

대응 : 오타 수정 완료

2.9 Stage 2034 (page 36)

➢ Use Case : Select Function에서 5번에 순서 순환에 대한 명시가 없다

대응 : 순환에 대한 세부 설명 추가

2.10 Stage 2035 (page 38) : 2. Operation Contracts

➢ Page38~47 : 도표 Type 부분 "syetem" 오타 -> 수정 요망

대응 : 오타 수정 완료

➢ 2. setTimePart() -> 내용 "AM/PM 표시" 추가 요망

대응 : 내용 추가 완료

2.11 Stage 2035 (page 46) : 2. Operation Contracts

➢ Post-Conditions의 설명에서 '범위 내'를 '사용자가 설정한 범위 내' 로 구체화 해야할 필요가 있다

대응 : 설정한 범위 내로 표현 구체화

2.12 Stage 2036 (page 51) : 4. Stopwatch

➢ 전체적인 다이어그램의 Flow가 앞서 설명한 내용과 불일치함

대응 : 여러 Use Case를 Fucntion 단위의 State Diagram으로 나타내는 데 제한이 있고 최대한 반영하려고 하였음, 현상 유지

2.13 Stage 2037 (page 57) :
17. 임의 방향 생성 시험 : "See.d" -> 오타 수정 요망
대응 : 오타 수정 완료

3 Stage 2040

3.1 Stage 2042 (page 16) : 1. Time Keeping
➢ "알람 indicator"는 모든 기능에서 작동함"
➔ 기존에 없던 새로운 기능이 생김 : 업데이트가 필요함
대응 : 기능적 추가가 아닌 UI적인 Detail로, GUI Design 단계에서 추가된 내용임

3.2 Stage 2042 (page 17) : 3. Stopwatch
➢ Lap & Split 확인 시, 이 둘을 동시에 표시하고 있다.
➔ 기존에서 "1번 Lap -> 1번 Split -> 2번...."식의 설명 때문에 따로 표현한다는 듯하는 혼동을 주었음
대응 : 순서대로 표현하도록 제대로 구현되어 있으므로 현상 유지

3.3 Stage 2042 (page 18) : 4. Alarm
➢ "6시 방향 LCD가 on/off 의미" : 기존에 없던 내용 -> 업데이트 요망
대응 : 기능적 추가가 아닌 UI적인 Detail로, GUI Design 단계에서 추가된 내용임

3.4 Stage 2042 (page 21) : 7. Select Function
➢ "6시 방향 LCD가 on/off 의미"라는 설명을 추가하는게 필요하다고 봄
대응 : 세부적인 설명 추가

4 Stage 2050 & 2060

4.1 Stage 2051 (page 5) : 2. Set Timer
➢ 10. nextTimePart의 도표가 누락됨 -> 추가해야 함
대응 : 누락된 도표 추가

4.2 Stage 2051 (page 13) : 7. Check Lap & Split
➢ 7. PressStart의 도표가 중복됨(page12에 이미 있음)
대응 : 중복되는 도표 삭제

4.3 Stage 2051 (page 18) : 14. Generate Random Direction
➢ "1. display random direction" : 오타 -> "5. ~"으로 수정 요망
대응 : 오타 수정 완료

5 Conclusion

➢ Stage 2050 & 2060 같은 최근에 작성한 문서의 변경사항이 Stage 1000, 2030, 2040 같은 예전에 작성한 문서에 업데이트하지 않아서 혼돈을 주었고, 일관성이 지켜지지 않았다.
➢ 전체적으로 구체적인 기준으로 기능 및 조건이 정의 되었지만, 몇몇의 경우에는 구체화가 필요해 보인다.
➢ 오타 주의
대응 : 모든 단계 보고서에 대한 전반적인 검토 및 수정을 통해 일관성을 보전하였고 구체적인 설명 추가와 오타 수정 진행

3. Testing Revision

1. Brute Force Test Result

Ref.	UseCase	detail	결과
R1.1	Set Time	set time에 12h, 12시 4분 입력 후 상태저장	T
		set time 중 mode를 여러 번 누르고 돌아왔을 때 set time 정보 초기화	F
		Set time 중 10의 자리가 최대치일 때 값을 상승시키면 0이 아닌 1로 초기화 됨	F
		funct 버튼을 여러 번 눌렀을 때 처음으로 돌아가지 않음	F
R1.2	Display Time	모든 시간 상태가 화면에 정상적으로 출력	T
		Southfinder에 1시, 12시 사이를 가리키는 지	T
R2.1	Set Timer	1시 2분 22초 입력 후 타이머 실행	T
		funct 버튼을 여러 번 눌렀을 때 처음으로 돌아가지 않음	F
		타이머가 작동 중일 때 Funct 버튼을 누르고 Start 버튼을 누르면 시간이 변경됨	F
R2.2	Beep Timer Buzzer	Timer가 0이되면 5초동안 버저를 울리는 지	T
R2.3	Stop timer Buzzer	Timer의 버저가 울릴 때 User가 mode버튼을 눌렀을 때 버저가 정지하는 지	T
		Timer가 0으로 초기화 되는지	T
R3.1	Stop Stopwatch	Stopwatch에서 start버튼 누르기	T
		입력 후 성공적으로 저장 되었는 지	T
R3.2	Store Lap & Split	버튼을 누를 때 마다 Lap, Split이 저장되는지	T
		최대 30개까지 저장이 되는지	T
		30개가 넘어가면 저장이 안되는 지	T
		확인 중 reset을 눌렀을 때 Lap 확인 gui가 계속 보임	F
R3.3	Pause Stowatch	User에게 스톱워치 일시정지를 입력받으면 정지하는 지	T
		User에게 스톱워치 재개를 요청받으면 재개하는지	T
R3.4	Check Lap & Split	User에게 버튼을 입력받아 저장된 Lap, Split이 출력되는지	T
		제일 마지막 Lap, Split이 출력된 후 다시 맨 처음 Lap, Split이 출력되는지	T
		Lap, Split 데이터가 없으면 기능이 제대로 블락되는지	T
R3.5	Reset Stopwatch	User에게 버튼을 입력받아 실행중인 스톱워치가 초기화 되는지	T
		저장되어있던 Lap, Split이 삭제되는지	T
R4.1	Set Alarm	User에게 버튼을 입력받아 알람이 저장되는지	T
		최대 10까지 저장이 되며 10개가 넘어가면 저장이 안되는지	T
		설정한 알람이 삭제 되는지	T

		알람 기능을 껐다 켤 시 알람 정보가 삭제되지 않는지	T
R4.2	Beep Alarm Buzzer	알람에 저장된 시간이 되면 버저가 1분 동안 울리는지	T
R4.3	Stop Alarm Buzzer	알람 버저가 울릴 때 User가 중지 요청 시 버저가 중지되는지	T
R4.4	Set Snooze	알람 버저가 울릴 때 User가 Snooze 요청 시 5분 후 다시 버저가 울리는지	T
R5.1	Set Scope	User의 버튼 입력에 따라 1~999,999가 입력 되는지	T
R5.2	Generate Random Number	현재 시각을 Seed로 하여 완전히 Random한 1~999,999 사이의 난수가 생성되는지	T
		생성된 난수가 화면에 출력되는지	T
R6.1	Generate Random Direction	현재 시각을 See.d로 하여 완전히 Random한 1~60 사이의 난수가 생성되는지	T
		생성된 난수에 해당하는 LCD가 출력되는지	T
R7.1	Select Function	User가 원하는 기능을 순서대로 선택할 수 있는지	T
		User가 원하는 기능을 설정하고 제대로 적용되는지	T

2. Partition Test Result

	case	result
1	12h	T
2	24h	T
3	min : 0	T
4	min : 60 >=	T
5	hour : 0	T
6	sec : 0	T
7	sec : 60 >=	T
8	split : 0	T
9	split : 10 >=	T
10	random : 0	T
11	random : 999999 >	T
12	1.0.1.1.0.0.3.0.0.0.0.0.0.0.)	T
13	1.0.1.2.0.0.3.0.0.0.0.0.0.0.)	T
14	1.0.2.1.0.0.3.0.0.0.0.0.0.0.)	T
15	1.0.2.2.0.0.3.0.0.0.0.0.0.0.)	F
16	1.0.3.1.0.0.3.0.0.0.0.0.0.0.)	F
17	1.0.3.2.0.0.3.0.0.0.0.0.0.0.)	T
18	1.0.4.1.0.0.3.0.0.0.0.0.0.0.)	T
19	1.0.4.2.0.0.3.0.0.0.0.0.0.0.)	T
20	2.0.1.2.0.2.2.2.1.1.1.0.0.0.)	T
21	2.0.1.2.0.2.3.2.1.1.1.0.0.0.	T
22	2.0.2.2.0.2.2.2.1.1.1.0.0.0.)	T
23	2.0.2.2.0.2.3.2.1.1.1.0.0.0.)	T
24	2.0.3.2.0.2.2.2.1.1.1.0.0.0.)	T
25	2.0.3.2.0.2.3.2.1.1.1.0.0.0.)	T
26	2.0.4.2.0.2.2.2.1.1.1.0.0.0.)	T
27	2.0.4.2.0.2.3.2.1.1.1.0.0.0.)	T
28	3.0.1.2.0.2.2.2.0.0.0.0.0.0.)	T
29	3.0.1.2.0.2.3.2.0.0.0.0.0.0.)	T
30	3.0.2.2.0.2.2.2.0.0.0.0.0.0.	T
31	3.0.2.2.0.2.3.2.0.0.0.0.0.0.)	T
32	3.0.3.2.0.2.2.2.0.0.0.0.0.0.)	T
33	3.0.3.2.0.2.3.2.0.0.0.0.0.0.)	F
34	3.0.4.2.0.2.2.2.0.0.0.0.0.0.)	T
35	3.0.4.2.0.2.3.2.0.0.0.0.0.0.	F
36	4.1.1.1.0.0.3.0.0.0.0.0.0.0.)	T
37	4.1.1.2.0.0.3.0.0.0.0.0.0.0.)	T
38	4.1.2.1.0.0.3.0.0.0.0.0.0.0.)	T
39	4.1.2.2.0.0.3.0.0.0.0.0.0.0.)	T
40	4.1.3.1.0.0.3.0.0.0.0.0.0.0.)	T
41	4.1.3.2.0.0.3.0.0.0.0.0.0.0.)	T
42	4.1.4.1.0.0.3.0.0.0.0.0.0.0.)	F
43	4.1.4.2.0.0.3.0.0.0.0.0.0.0.)	T
44	4.2.1.1.0.0.3.0.0.0.0.2.0.0.	F
45	4.2.1.2.0.0.3.0.0.0.0.2.0.0.)	F
46	4.2.2.1.0.0.3.0.0.0.0.2.0.0.)	T
47	4.2.2.2.0.0.3.0.0.0.0.2.0.0.)	T
48	4.2.3.1.0.0.3.0.0.0.0.2.0.0.	T

49	4.2.3.2.0.0.3.0.0.0.0.2.0.0.)	T
50	4.2.4.1.0.0.3.0.0.0.0.2.0.0.)	T
51	4.2.4.2.0.0.3.0.0.0.0.2.0.0.)	T
52	5.0.1.1.0.0.3.0.0.0.0.0.0.0.	T
53	5.0.1.2.0.0.3.0.0.0.0.0.0.0.)	T
54	5.0.2.1.0.0.3.0.0.0.0.0.0.0.)	T
55	5.0.2.2.0.0.3.0.0.0.0.0.0.0.)	T
56	5.0.3.1.0.0.3.0.0.0.0.0.0.0.)	T
57	5.0.3.2.0.0.3.0.0.0.0.0.0.0.)	T
58	5.0.4.1.0.0.3.0.0.0.0.0.0.0.)	T
59	5.0.4.2.0.0.3.0.0.0.0.0.0.0.)	T
60	6.0.1.1.0.0.3.0.0.0.0.0.2.0.)	F
61	6.0.1.2.0.0.3.0.0.0.0.0.2.0.)	T
62	6.0.2.1.0.0.3.0.0.0.0.0.2.0.)	F
63	6.0.2.2.0.0.3.0.0.0.0.0.2.0.)	F
64	6.0.3.1.0.0.3.0.0.0.0.0.2.0.)	T
65	6.0.3.2.0.0.3.0.0.0.0.0.2.0.)	T
66	6.0.4.1.0.0.3.0.0.0.0.0.2.0.)	T
67	6.0.4.2.0.0.3.0.0.0.0.0.2.0.)	T
68	7.0.1.2.0.0.3.0.0.0.0.0.0.0.)	T
69	7.0.2.2.0.0.3.0.0.0.0.0.0.0.)	T
70	7.0.3.2.0.0.3.0.0.0.0.0.0.0.)	T
71	7.0.4.2.0.0.3.0.0.0.0.0.0.0.)	T
72	8.0.1.1.0.0.3.0.0.0.0.0.0.1.)	T
73	8.0.1.1.0.0.3.0.0.0.0.0.0.2.	T
74	8.0.1.2.0.0.3.0.0.0.0.0.0.1	F
75	8.0.1.2.0.0.3.0.0.0.0.0.0.2.)	T
76	8.0.2.1.0.0.3.0.0.0.0.0.0.1.)	T
77	8.0.2.1.0.0.3.0.0.0.0.0.0.2.)	F
78	8.0.2.2.0.0.3.0.0.0.0.0.0.1.)	T
79	8.0.2.2.0.0.3.0.0.0.0.0.0.2.)	T
80	8.0.3.1.0.0.3.0.0.0.0.0.0.1.)	T
81	8.0.3.1.0.0.3.0.0.0.0.0.0.2.)	T
82	8.0.3.2.0.0.3.0.0.0.0.0.0.1.)	T
83	8.0.3.2.0.0.3.0.0.0.0.0.0.2.)	T
84	8.0.4.1.0.0.3.0.0.0.0.0.0.1.)	T
85	8.0.4.1.0.0.3.0.0.0.0.0.0.2.)	T
86	8.0.4.2.0.0.3.0.0.0.0.0.0.1.)	T
87	8.0.4.2.0.0.3.0.0.0.0.0.0.2.)	T
88	9.0.1.2.0.0.3.0.0.0.0.0.0.0.)	T
89	9.0.2.2.0.0.3.0.0.0.0.0.0.0.)	T
90	= 9.0.3.2.0.0.3.0.0.0.0.0.0.0.)	T
91	9.0.4.2.0.0.3.0.0.0.0.0.0.0.)	T
92	10.0.1.2.0.0.3.0.0.0.0.0.0.0.)	T
93	10.0.2.2.0.0.3.0.0.0.0.0.0.0.	T
94	10.0.3.2.0.0.3.0.0.0.0.0.0.0.)	T
95	10.0.4.2.0.0.3.0.0.0.0.0.0.0.)	T

3. Pairwise Test Result

	case	result
1	1.0.1.1.0.0.3.0.0.0.0.0.0.0.)	T
2	1.0.1.2.0.0.3.0.0.0.0.0.0.0.)	T
3	1.0.2.1.0.0.3.0.0.0.0.0.0.0.)	T
4	1.0.2.2.0.0.3.0.0.0.0.0.0.0.)	F
5	1.0.3.1.0.0.3.0.0.0.0.0.0.0.)	F
6	1.0.3.2.0.0.3.0.0.0.0.0.0.0.)	T
7	1.0.4.1.0.0.3.0.0.0.0.0.0.0.)	T
8	1.0.4.2.0.0.3.0.0.0.0.0.0.0.)	T
9	2.0.1.2.0.2.2.2.1.1.1.0.0.0.)	T
10	2.0.1.2.0.2.3.2.1.1.1.0.0.0.	T
11	2.0.2.2.0.2.2.2.1.1.1.0.0.0.)	T
12	2.0.2.2.0.2.3.2.1.1.1.0.0.0.)	T
13	2.0.3.2.0.2.2.2.1.1.1.0.0.0.)	T
14	2.0.3.2.0.2.3.2.1.1.1.0.0.0.)	T
15	2.0.4.2.0.2.2.2.1.1.1.0.0.0.)	T
16	2.0.4.2.0.2.3.2.1.1.1.0.0.0.)	T
17	3.0.1.2.0.2.2.2.0.0.0.0.0.0.)	T
18	3.0.1.2.0.2.3.2.0.0.0.0.0.0.)	T
19	3.0.2.2.0.2.2.2.0.0.0.0.0.0.	T
20	3.0.2.2.0.2.3.2.0.0.0.0.0.0.)	T
21	3.0.3.2.0.2.2.2.0.0.0.0.0.0.)	T
22	3.0.3.2.0.2.3.2.0.0.0.0.0.0.)	F
23	3.0.4.2.0.2.2.2.0.0.0.0.0.0.)	T
24	3.0.4.2.0.2.3.2.0.0.0.0.0.0.	F
25	4.1.1.1.0.0.3.0.0.0.0.0.0.0.)	T
26	4.1.1.2.0.0.3.0.0.0.0.0.0.0.)	T
27	4.1.2.1.0.0.3.0.0.0.0.0.0.0.)	T
28	4.1.2.2.0.0.3.0.0.0.0.0.0.0.)	T
29	4.1.3.1.0.0.3.0.0.0.0.0.0.0.)	T
30	4.1.3.2.0.0.3.0.0.0.0.0.0.0.)	T
31	4.1.4.1.0.0.3.0.0.0.0.0.0.0.)	F
32	4.1.4.2.0.0.3.0.0.0.0.0.0.0.)	T
33	4.2.1.1.0.0.3.0.0.0.0.2.0.0.	F
34	4.2.1.2.0.0.3.0.0.0.0.2.0.0.)	F
35	4.2.2.1.0.0.3.0.0.0.0.2.0.0.)	T
36	4.2.2.2.0.0.3.0.0.0.0.2.0.0.)	T
37	4.2.3.1.0.0.3.0.0.0.0.2.0.0.	T
38	4.2.3.2.0.0.3.0.0.0.0.2.0.0.)	T
39	4.2.4.1.0.0.3.0.0.0.0.2.0.0.)	T
40	4.2.4.2.0.0.3.0.0.0.0.2.0.0.)	T
41	5.0.1.1.0.0.3.0.0.0.0.0.0.0.	T
42	5.0.1.2.0.0.3.0.0.0.0.0.0.0.)	T
43	5.0.2.1.0.0.3.0.0.0.0.0.0.0.)	T
44	5.0.2.2.0.0.3.0.0.0.0.0.0.0.)	T
45	5.0.3.1.0.0.3.0.0.0.0.0.0.0.)	T
46	5.0.3.2.0.0.3.0.0.0.0.0.0.0.)	T
47	5.0.4.1.0.0.3.0.0.0.0.0.0.0.)	T
48	5.0.4.2.0.0.3.0.0.0.0.0.0.0.)	T

49	6.0.1.1.0.0.3.0.0.0.0.0.2.0.)	T
50	6.0.1.2.0.0.3.0.0.0.0.0.2.0.)	T
51	6.0.2.1.0.0.3.0.0.0.0.0.2.0.)	T
52	6.0.2.2.0.0.3.0.0.0.0.0.2.0.)	T
53	6.0.3.1.0.0.3.0.0.0.0.0.2.0.)	T
54	6.0.3.2.0.0.3.0.0.0.0.0.2.0.)	T
55	6.0.4.1.0.0.3.0.0.0.0.0.2.0.)	T
56	6.0.4.2.0.0.3.0.0.0.0.0.2.0.)	T
57	7.0.1.2.0.0.3.0.0.0.0.0.0.0.)	T
58	7.0.2.2.0.0.3.0.0.0.0.0.0.0.)	T
59	7.0.3.2.0.0.3.0.0.0.0.0.0.0.)	T
60	7.0.4.2.0.0.3.0.0.0.0.0.0.0.)	T
61	8.0.1.1.0.0.3.0.0.0.0.0.0.1.)	T
62	8.0.1.1.0.0.3.0.0.0.0.0.0.2.	T
63	8.0.1.2.0.0.3.0.0.0.0.0.0.1	T
64	8.0.1.2.0.0.3.0.0.0.0.0.0.2.)	T

4. Test Pass Rate

- Brute Force Test

→ Pass Rate : 85.7% (6 in 42 Test Cases was Failed)

- Category Partitioning Test

→ Pass Rate : 87.4% (12 in 95 Test Cases was Failed)

- Pairwise Combination Test

→ Pass Rate : 89.1% (7 in 64 Test Cases was Failed)

5. Conclusion

Fail 처리된 항목들은 모두 Brute Force Test와 관련된 사항으로 그에 해당하는 6개의 Fail Case에 대해서 조치가 필요

5. Fail Management

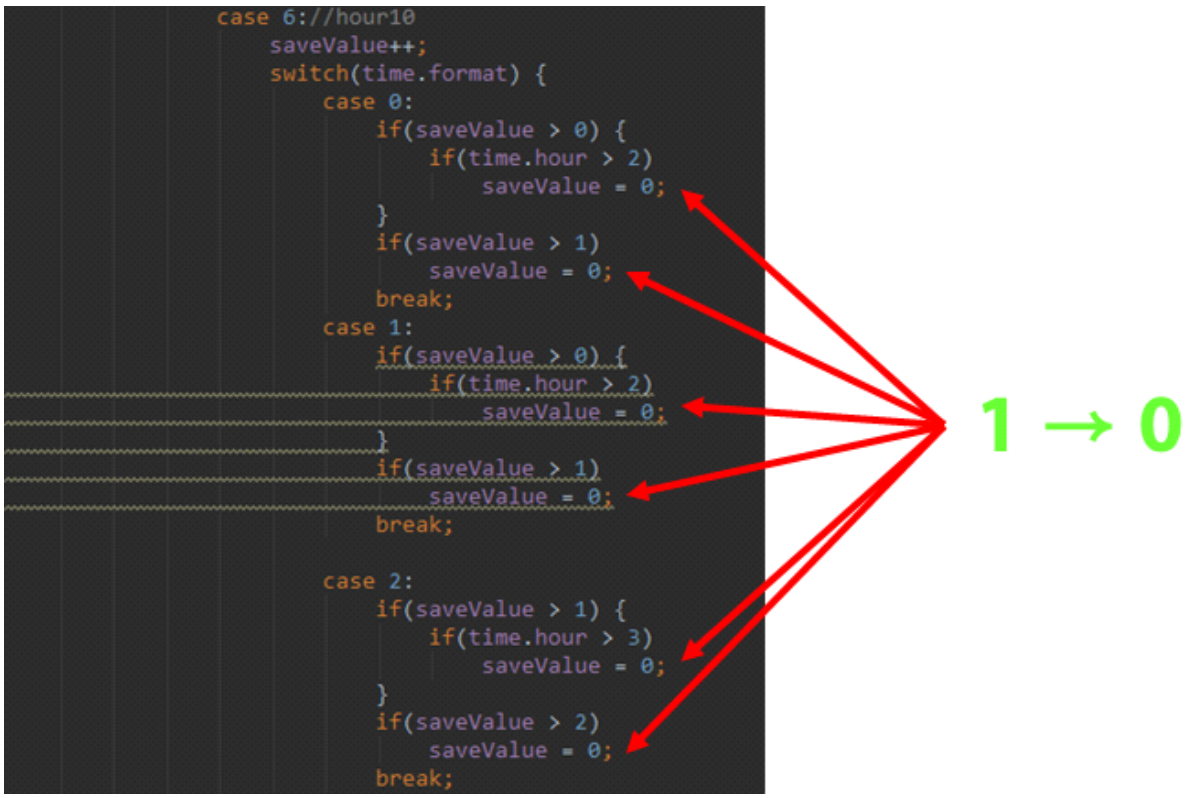
1) 시간 설정 중 Mode를 넘겨서 돌아올 경우 Display가 모두 0으로 초기화됨

→ 시간 설정 중에는 Mode 버튼을 사용할 수 없도록 Block 처리

```
    @Override
    public void mouseReleased(MouseEvent arg0) {
        // TODO Auto-generated method stub
        timeClicked = new Date().getTime() - pressedTime.getTime();
        if(timeClicked >= 2000) {
            if(selectfunction.functionlist.timekeeping.time.onoff == 0)
                return;
            if(selectfunction.setting == 1)
                return;
            view.currfunc = 7;
            selectfunction.reqSelectFunction();
        }
        else {
            if(selectfunction.functionlist.timekeeping.time.onoff == 0)
                return;
            if(selectfunction.setting == 1)
                return;
            selectfunction.reqNextFunction();
        }
    }
};
```

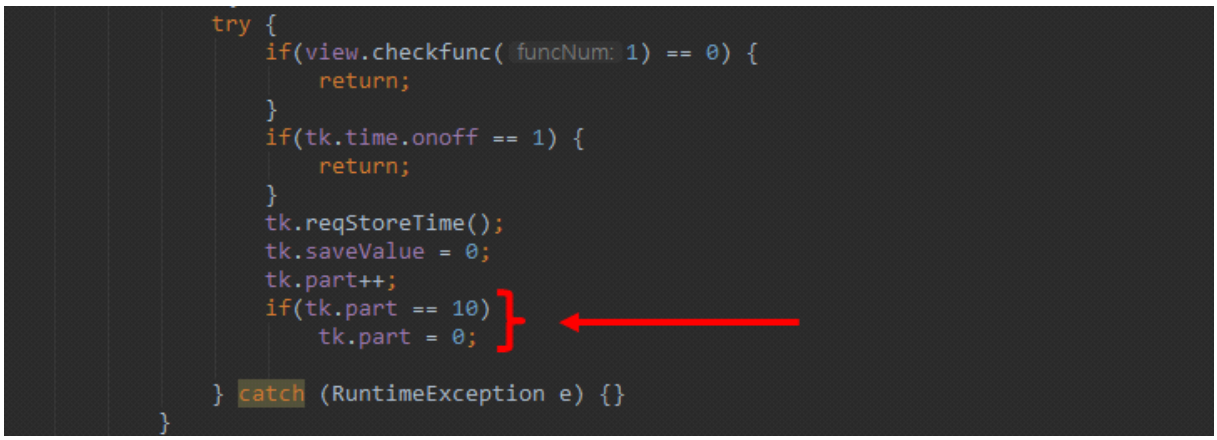
2) 시간 설정 중 시 10의 자리가 최대치일 때 값을 상승시키면 0이 아니라 1로 초기화됨

→ 시간 설정 중에는 Mode 버튼을 사용할 수 없도록 Block 처리



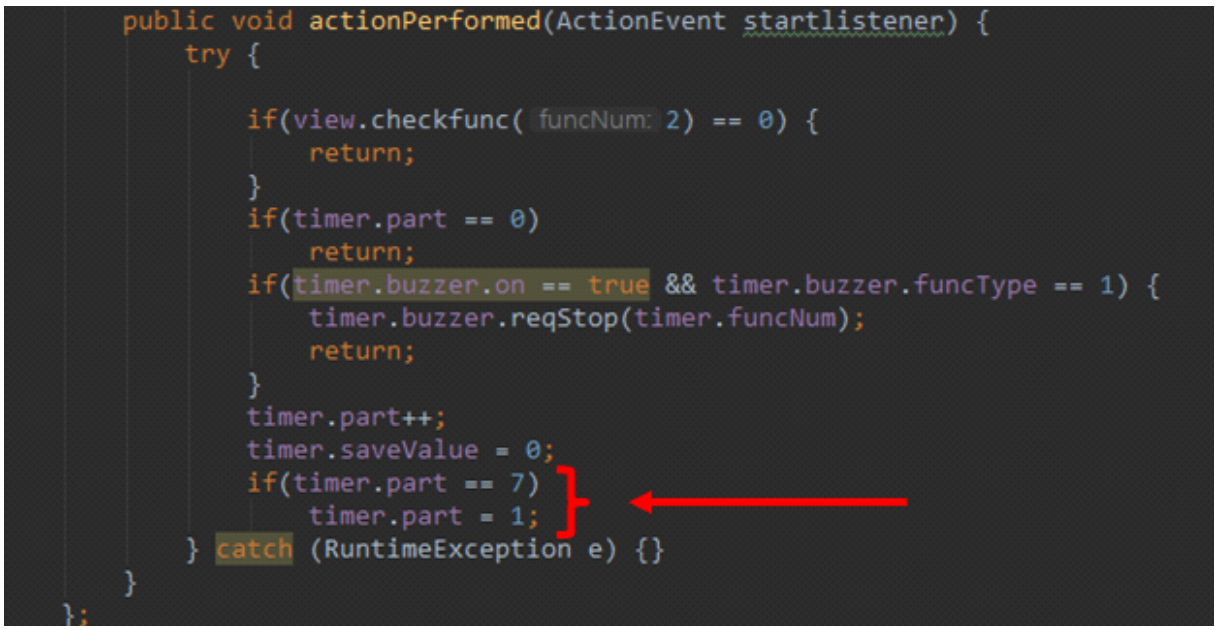
3) 시간 설정 중 마지막 Part에서 Funct 버튼을 눌러도 첫번째 Part 설정으로 돌아가지 않음

→ Part 변수가 최대치를 넘어서면 0으로 초기화



4) 타이머 설정 중 마지막 Part에서 Funct 버튼을 눌러도 첫번째 Part 설정으로 돌아가지 않음

→ Part 변수가 최대치를 넘어서면 1로 초기화



5) 타이머가 작동 중일 때 Funct 버튼을 누르고 Start 버튼을 누르면 시간이 변경됨

→ 타이머가 작동중일 때 Mode 버튼을 제외하고 모두 Block 처리

6) Lap & Split 확인 중 Reset 버튼을 누르면 Display에 불필요한 Lap GUI가 잔존함

→ Lap & Split 확인 중에는 Reset 버튼 Block 처리