

골라보시게

OOPT 2nd Cycle : Revision

Team #1

201411273 박재범

201411275 박진호

201411311 장원영

201311313 정인원

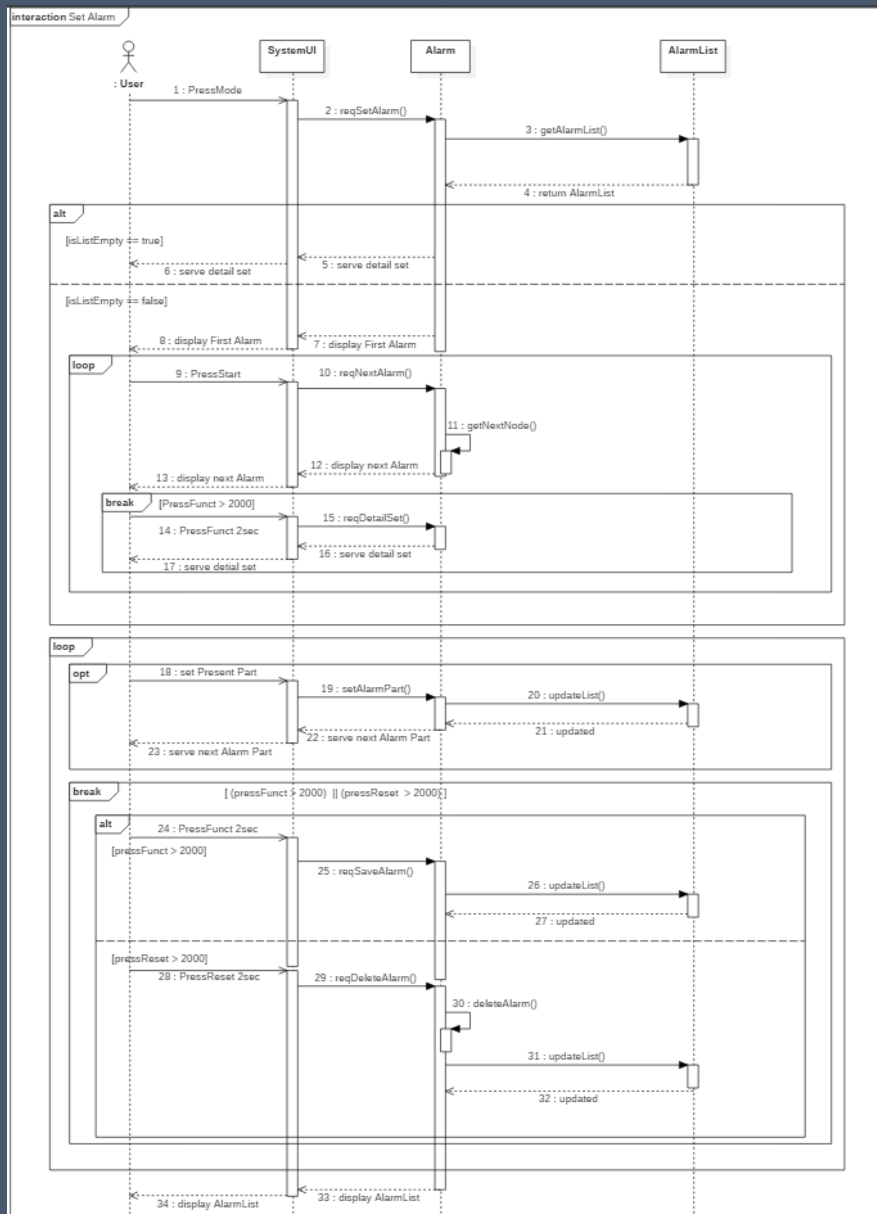
Contents

- 01 Incomplete Part Completion
- 02 Specification Revision
- 03 Testing Revision

1

Incomplete Part Completion

1. Incomplete Part Completion



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

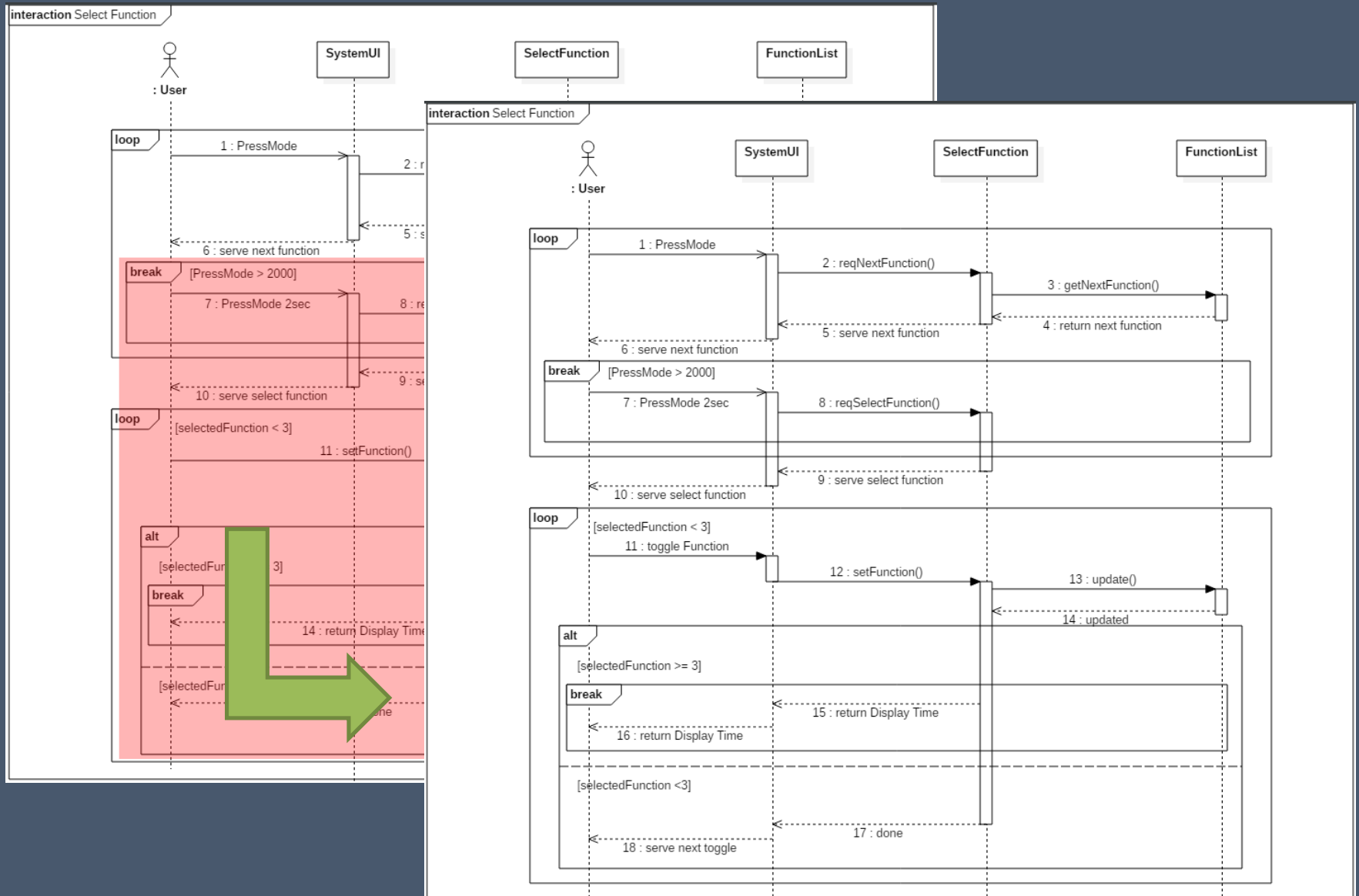
Name	8. display First Alarm
Responsibilities	알람 리스트의 첫번째 요소를 출력한다.
Type	GUI
Cross Reference	R4.1, R4.2
Note	N/A
Pre-Conditions	알람 리스트에 데이터가 존재해야 한다.
Post-Conditions	N/A

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. PressFunc 2sec
Responsibilities	Func 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

1. Incomplete Part Completion



1. Incomplete Part Completion

Test #	Test 항목	Description	Use Case	Sys. Func.
9-2	랩, 스플릿 확인 시험 (35 Times)	- 정지 상태에서 Funct 버튼을 2초간 누르면 Lap, Split 확인으로 넘어가는지 Test - 7-1에서 35번의 저장 요청에서 먼저 저장한 30개의 데이터만 저장되어 있는지 Test - 제일 마지막 Lap, Split이 출력된 후 다시 맨 처음 Lap, Split이 출력되는지 Test - Lap, Split 데이터가 없으면 기능이 제대로 <u>블록되는지</u> Test	9. Check Lap & Split	R3.4
10-1	스톱워치 초기화 시험	- 정지 상태에서 Reset 버튼을 누르면 스톱워치가 초기화 되는지 Test - 다시 시작 후 바로 정지한 뒤 이전에 <u>저장되어있던</u> Lap, Split이 삭제되었는지 Test	10. Reset Stopwatch	R3.5
11-1	알람 설정 시험 (세부 설정)	- 17:00에 월, 수, 금에 반복되는 알람을 On 하도록 설정 - 리스트에서 설정한 알람에 대한 정보를 확인 - 생성한 알람이 <u>삭제</u> 되는지 Test - 알람 기능을 껐다 켤 시 알람 정보가 삭제되지 않는지 Test	11. Set Alarm	R4.1
11-2	알람 설정 시험 (목록 설정)	- 17:00부터 1분 간격으로 화, 목에 반복되는 알람을 Off 상태로 10개 설정 - 리스트에서 설정한 알람들에 대한 정보를 확인 - 10개를 초과하여 새로운 알람을 생성할 수 없는지 확인	11. Set Alarm	R4.1
12-1	알람 버저 시험	- 17:00에 울리도록 알람에 저장된 시간이 되면 버저가 1분 동안 울리는지 Test	12. Beep Alarm Buzzer	R4.2
13-1	알람 버저 중지 시험 (Start)	- 알람 버저가 울릴 때 Start 버튼을 누르면 멈추는지 각각 Test	13. Stop Alarm Buzzer	R4.3
13-2	알람 버저 중지 시험 (Reset)	- 알람 버저가 울릴 때 Reset 버튼을 누르면 멈추는지 각각 Test	13. Stop Alarm Buzzer	R4.3
14-1	<u>스누즈</u> 시험 (Mode)	- 알람 버저가 울릴 때 Mode 버튼을 누르면 5분 뒤 Snooze가 울리는지 각각 Test	14. Set Snooze	R4.4
14-2	<u>스누즈</u> 시험 (Funct)	- 알람 버저가 울릴 때 Funct 버튼을 누르면 5분 뒤 Snooze가 울리는지 각각 Test	14. Set Snooze	R4.4

1. Incomplete Part Completion

```
AlarmTest.java
1 import org.junit.Test;
2
3 import static org.junit.Assert.*;
4
5 public class AlarmTest {
6     @Test
7     public void reqSetAlarm() {
8         View v = new View();
9         AlarmList al = new AlarmList(v);
10        Buzzer b = new Buzzer(v,al);
11        Alarm alm = new Alarm(v,al,b);
12
13        alm.alarmIndex = 1;
14        alm.reqSetAlarm();
15        assertEquals(0, alm.alarmIndex);
16    }
17
18    @Test
19    public void reqDetailSet() {
20        View v = new View();
21        AlarmList al = new AlarmList(v);
22        Buzzer b = new Buzzer(v,al);
23        Alarm alm = new Alarm(v,al,b);
24
25        alm.reqDetailSet();
26        assertEquals(0, alm.sec);
27        assertEquals(0, alm.sec10);
28        assertEquals(0, alm.min);
29        assertEquals(0, alm.min10);
30        assertEquals(0, alm.hour);
31        assertEquals(0, alm.hour10);
32    }
33
34    @Test
35    public void setAlarmPart() {
36        View v = new View();
37        AlarmList al = new AlarmList(v);
38        Buzzer b = new Buzzer(v,al);
39        Alarm alm = new Alarm(v,al,b);
40
41        alm.part=1;
42        alm.sec = 8;
43        alm.setAlarmPart(0);
44        assertEquals(9, alm.sec);
45        alm.sec = 9;
46        alm.setAlarmPart(0);
47        assertEquals(9,alm.sec);
48    }
49
50    @Test
51    public void reqSaveAlarm() {
52        View v = new View();
53        AlarmList al = new AlarmList(v);
54        Buzzer b = new Buzzer(v,al);
55        Alarm alm = new Alarm(v,al,b);
56        alm.reqSaveAlarm();
57        assertEquals(0,alm.sec);
58        assertEquals(0,alm.sec10);
59        assertEquals(0,alm.min);
60        assertEquals(0,alm.min10);
61        assertEquals(0,alm.hour);
62    }
63}
```

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

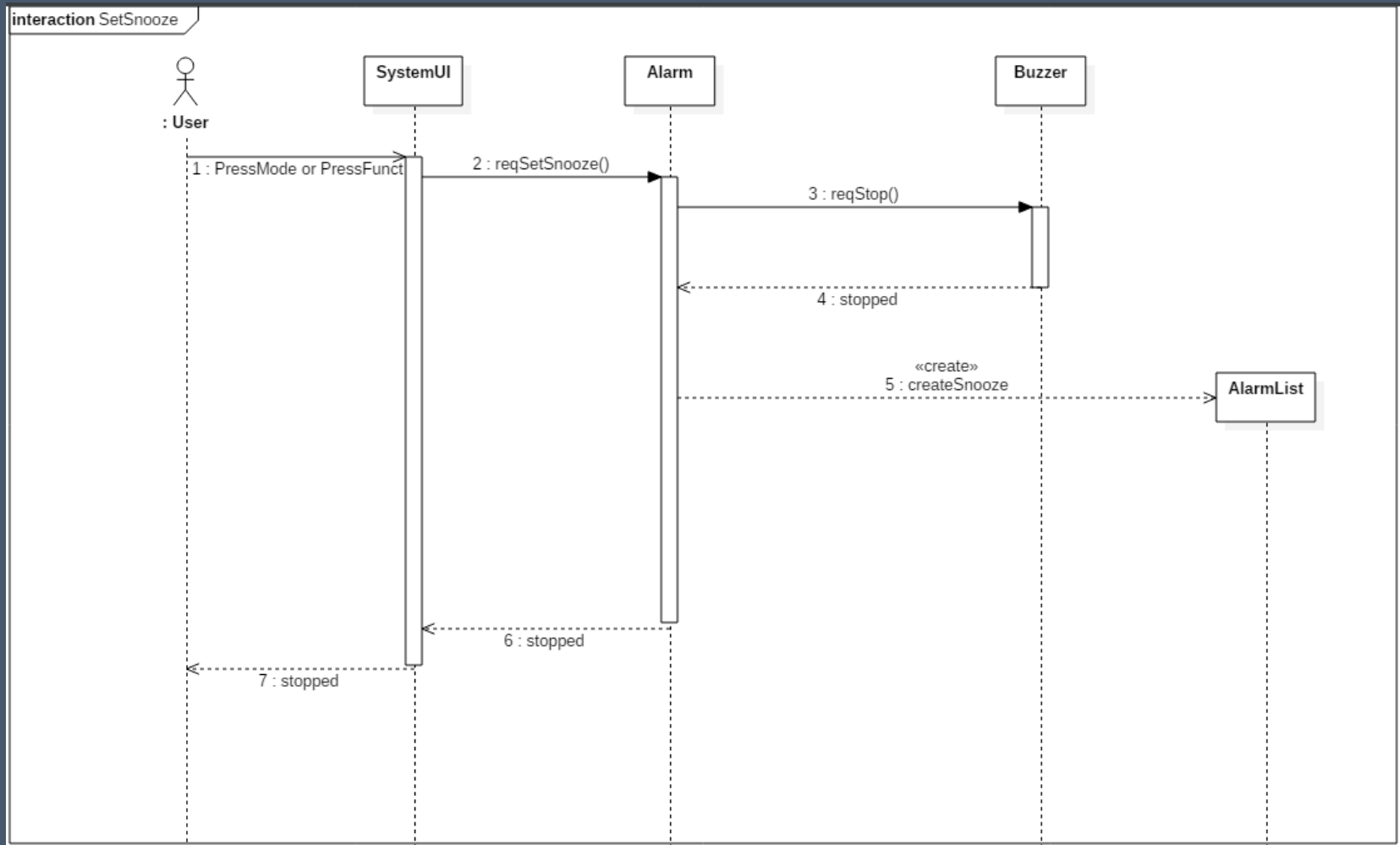
passed 76 ms

2

Specification Revision

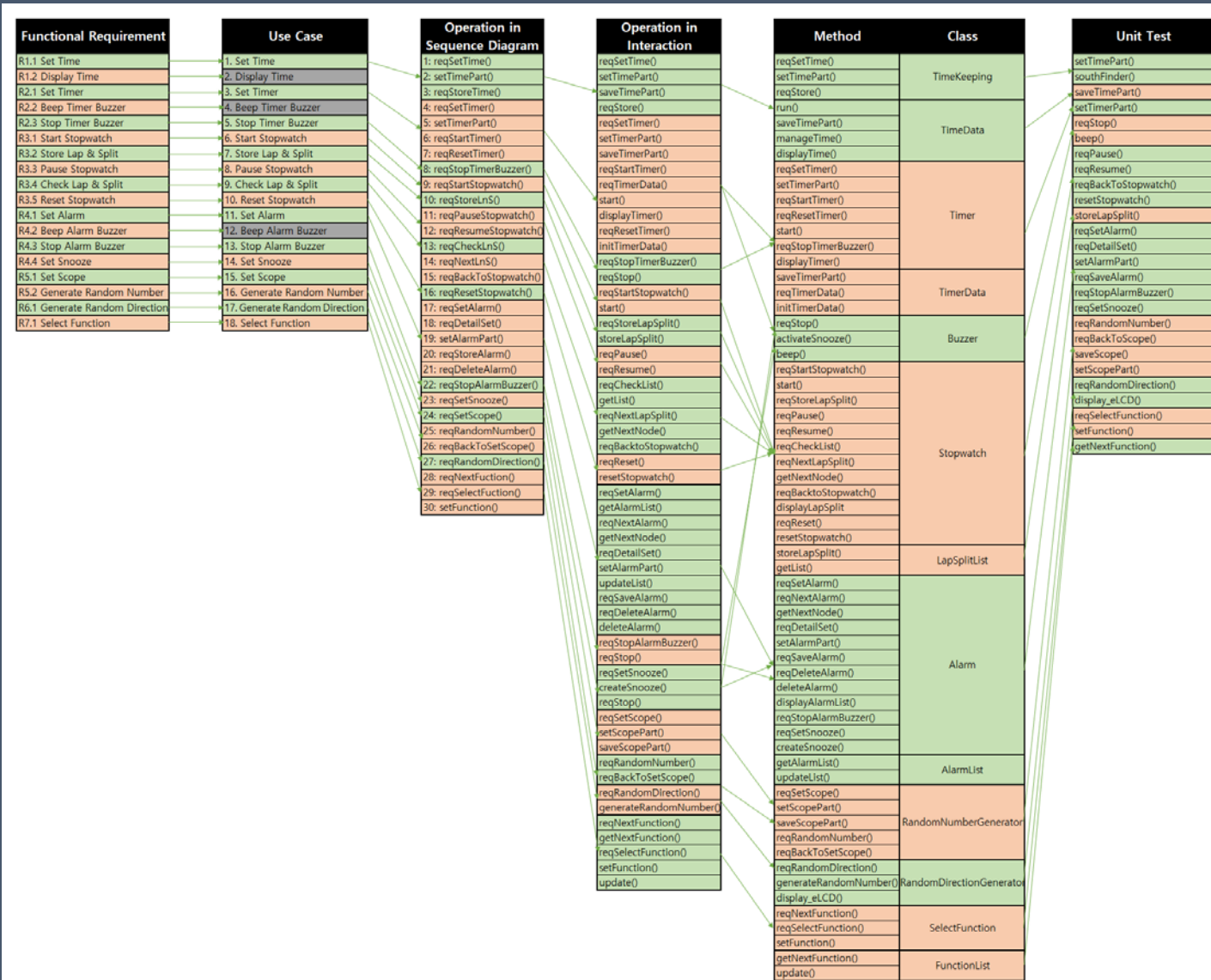
2. Specification Revision

- Set Snooze 작동 방식 변경(Method 추가)



2. Specification Revision

- Set Snooze 작동 방식 변경(Method 추가)



2. Specification Revision

- Buzzer OFF를 위한 조건 변경

Use Case	Set Snooze
Actor	User
Purpose	알람 Snooze를 설정한다.
Overview	User가 Mode나 Funct 버튼을 눌러 버저를 즉시 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 Funct 버튼을 누른다. 2. (S) : 버저를 즉시 멈추고 5분 후에 다시 울리도록 임시 알람을 설정한다.
Alternative Courses Of Events	N/A
Exceptional Courses Of Events	N/A



Use Case	Set Snooze
Actor	User
Purpose	알람 Snooze를 설정한다.
Overview	User가 Alarm 기능을 선택한 상태에서 Funct 버튼을 눌러 버저를 즉시 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 Funct 버튼을 누른다. 2. (S) : 버저를 즉시 멈추고 5분 후에 다시 울리도록 임시 알람을 설정한다.
Alternative Courses Of Events	N/A
Exceptional Courses Of Events	N/A

2. Specification Revision

1. OOPT Stage 1000 & 2030

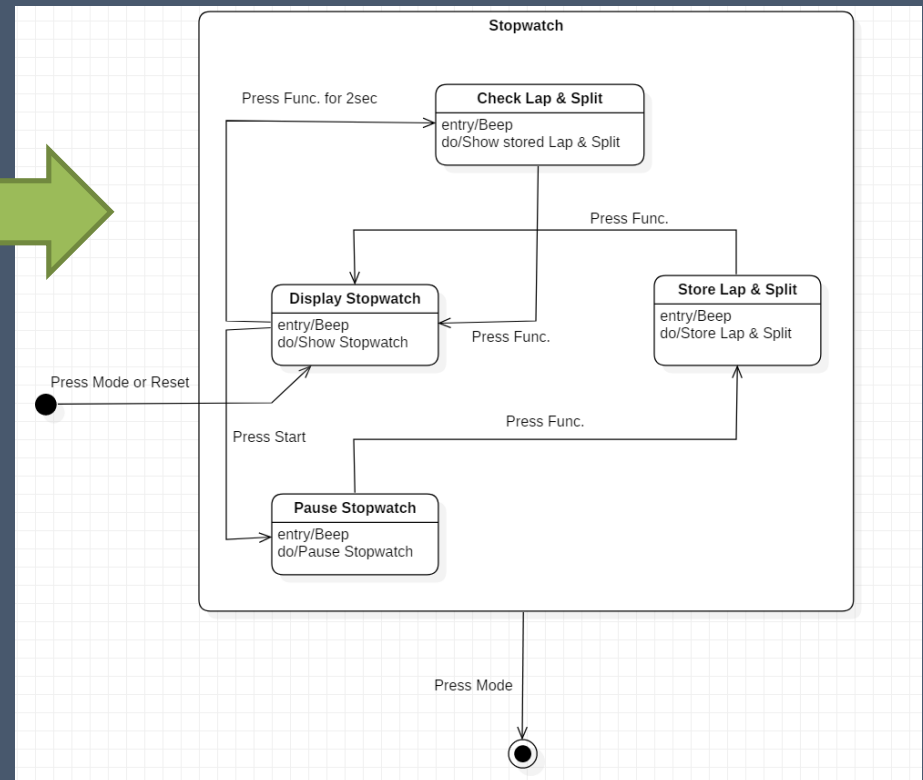
✓ 오타 수정 및 모호한 부분 구체적인 설명 추가 요망

✓ State Diagram의 Flow가 앞선 설명과 불일치

Use Case	Start Stopwatch	Use Case	Store Lap & Split
Actor	User	Actor	User
Purpose	스톱워치를 작동시킨다.	Purpose	Stopwatch의 Lap과 Split을 저장한다.
Overview	User가 Stopwatch를 작동시킨다.	Overview	Stopwatch가 시작된 상태에서 Func키를 눌러 Lap과 Split을 저장한다.
Type	Evident	Type	Evident
Cross Reference	Functional Requirements : R.3.2	Cross Reference	Functional Requirements : R.3.2

Use Case	Pause Stopwatch	Use Case	Check Lap & Split
Actor	User	Actor	User
Purpose	Stopwatch를 일시 정지하거나 및 재개한다.	Purpose	현재까지 저장된 Lap과 Split 시간을 확인한다.
Overview	작동 중인 Stopwatch를 Start 버튼을 눌러 정지시킨다. 정지 상태에서 다시 Start 버튼을 누르면 Stopwatch가 ON 상태에 들어간다.	Overview	Func 버튼을 2초간 눌러 저장된 Lap, Split 데이터를 확인할 수 있다.
Type	Evident	Type	Evident
Cross Reference	Functional Requirements : R.3.3	Cross Reference	Functional Requirements : R.3.4
Pre-Requisites	Select Function에 Stopwatch가 ON 상태에 있어야 한다.	Pre-Requisites	Select Function에 Stopwatch가 ON 상태에 있어야 한다.

Use Case	Reset Stopwatch	Use Case	Display Stopwatch
Actor	User	Actor	User
Purpose	Stopwatch의 데이터를 초기화한다.	Purpose	현재까지 저장된 Lap과 Split 시간을 확인한다.
Overview	Reset 버튼을 눌러 시간과 저장된 Lap, Split 데이터를 초기화한다.	Overview	Func 버튼을 2초간 누르면 저장된 Lap, Split 데이터를 확인할 수 있다.
Type	Evident	Type	Evident
Cross Reference	Functional Requirements : R.3.5	Cross Reference	Functional Requirements : R.3.5
Pre-Requisites	Stopwatch가 일시정지 상태에 있어야 한다.	Pre-Requisites	Stopwatch가 일시정지 상태에 있어야 한다.
Typical Courses Of Events	(A) : Actor, (S) : System 1. (A) : Reset 버튼을 누른다. 2. (S) : 시간을 0으로 초기화하고 Lap & Split 데이터도 모두 초기화한다.	Typical Courses Of Events	(A) : Actor, (S) : System 1. (A) : Display 버튼을 누른다. 2. (S) : 현재까지 저장된 Lap과 Split 데이터를 화면에 표시한다.
Alternative Courses Of Events	N/A	Alternative Courses Of Events	N/A
Exceptional Courses Of Events	E2. Stopwatch가 작동 중에는 Reset 버튼을 눌러도 초기화하지 않는다.	Exceptional Courses Of Events	E2. Stopwatch가 작동 중에는 Display 버튼을 눌러도 데이터를 확인할 수 없다.



2. Specification Revision

2. OOPT Stage 2040

☑ 기존에 없던 설명이 생겼음

→ 기능적 추가가 아닌 UI Design 단계에서의 Detail 설명



2. Specification Revision

3. OOPT Stage 2050 & 2060

☑ 도표 누락 및 중복이 발생

4. Conclusion

☑ 오타가 발생하지 않도록 주의

☑ 구체적인 설명 추가가 필요한 부분이 있다

☑ 예전 보고서에 일부 업데이트되지 않은 부분이 있다

→ 놓쳤던 부분에 대한 전반적인 검토 및 수정 완료

☐ 19-04-09 Software Modeling 1000 보고서(V4).hwp

☐ 19-05-25 Software Modeling 2030 보고서(V5).hwp

☐ 19-05-25 Software Modeling 2040 보고서(V3).hwp

☐ 19-05-28 Software Modeling 2050 & 2060 보고서(V2).hwp

3

Testing Revision

3. Testing Revision

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**)

3. Testing Revision

Fail #1. 시간 설정 중 Mode를 넘겨서 돌아올 경우
Display가 모두 0으로 초기화됨

→ 시간 설정 중에는 Mode 버튼을 Block 처리

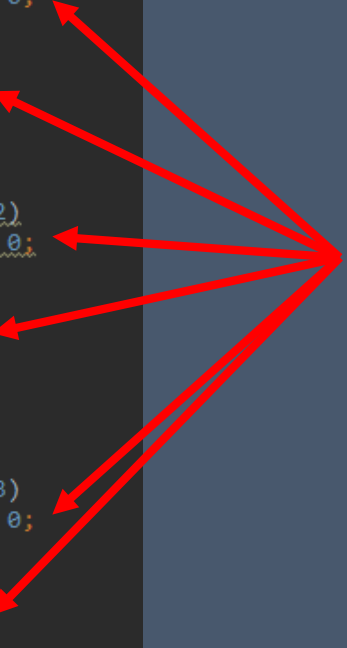
```
@Override
public void mouseReleased(MouseEvent arg0) {
    // TODO Auto-generated method stub
    timeClicked = new Date().getTime() - pressedTime.getTime();
    if(timeClicked >= 2000) {
        if(selectfunction.functionlist.timekeeping.time.onoff == 0)
            return;
        if(selectfunction.setting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

3. Testing Revision

Fail #2. 시간 설정 중 시 10의 자리가 최대치일 때
값을 상승시키면 0이 아니라 1로 초기화됨
→ 초기화값을 0으로 수정

```
case 6://hour10
saveValue++;
switch(time.format) {
case 0:
    if(saveValue > 0) {
        if(time.hour > 2)
            saveValue = 0;
    }
    if(saveValue > 1)
        saveValue = 0;
    break;
case 1:
    if(saveValue > 0) {
        if(time.hour > 2)
            saveValue = 0;
    }
    if(saveValue > 1)
        saveValue = 0;
    break;
case 2:
    if(saveValue > 1) {
        if(time.hour > 3)
            saveValue = 0;
    }
    if(saveValue > 2)
        saveValue = 0;
    break;
```

1 → 0



3. Testing Revision

Fail #3. 시간 설정 중 마지막 Part에서 Funct 버튼을
눌러도 첫번째 Part 설정으로 돌아가지 않음
→ Part 변수가 최대치를 넘어서면 0으로 초기화

```
try {  
    if(view.checkfunc( funcNum: 1) == 0) {  
        return;  
    }  
    if(tk.time.onoff == 1) {  
        return;  
    }  
    tk.reqStoreTime();  
    tk.saveValue = 0;  
    tk.part++;  
    if(tk.part == 10) } ←  
        tk.part = 0;  
} catch (RuntimeException e) {}  
}
```

3. Testing Revision

Fail #4. 타이머 설정 중 마지막 Part에서 Funct 버튼을 눌러도 첫번째 Part 설정으로 돌아가지 않음
→ Part 변수가 최대치를 넘어서면 1로 초기화

```
public void actionPerformed(ActionEvent startlistener) {  
    try {  
        if(view.checkfunc( funcNum: 2) == 0) {  
            return;  
        }  
        if(timer.part == 0)  
            return;  
        if(timer.buzzer.on == true && timer.buzzer.funcType == 1) {  
            timer.buzzer.reqStop(timer.funcNum);  
            return;  
        }  
        timer.part++;  
        timer.saveValue = 0;  
        if(timer.part == 7) }  
            timer.part = 1;  
    } catch (RuntimeException e) {}  
};
```



3. Testing Revision

Fail #5. 타이머가 작동 중일 때 Funct 버튼을 누르고
Start 버튼을 누르면 시간이 변경됨
→ 타이머가 작동중일 때 Mode 버튼을 제외하고
모두 Block 처리

```
public void mouseReleased(MouseEvent arg0) {  
    // TODO Auto-generated method stub  
    timeClicked = new Date().getTime() - pressedTime.getTime();  
    if(timeClicked >= 2000) {  
        if(view.checkfunc( funcNum: 2) == 0) {  
            return;  
        }  
        try {  
            timer.reqStartTimer();  
        } catch (RuntimeException e) {}  
    }  
    else {  
        try {  
            if(view.checkfunc( funcNum: 2) == 0) {  
                return;  
            }  
            if(timer.buzzer.on == true && timer.buzzer.funcType == 1) {  
                timer.buzzer.reqStop(timer.funcNum);  
                return;  
            }  
            if(view.checkfunc( funcNum: 2) == 0) {  
                return;  
            }  
            if(timer.part == 0) {  
                return;  
            }  
            timer.setTimerPart(0);  
        } catch (RuntimeException e) {}  
    }  
}
```

```
ActionListener flistener = new ActionListener() {  
    public void actionPerformed(ActionEvent startlistener) {  
        try {  
            if(view.checkfunc( funcNum: 2) == 0) {  
                return;  
            }  
            if(timer.part == 0) {  
                return;  
            }  
            if(timer.buzzer.on == true && timer.buzzer.funcType == 1) {  
                timer.buzzer.reqStop(timer.funcNum);  
                return;  
            }  
            timer.part++;  
            timer.saveValue = 0;  
            if(timer.part == 7) {  
                timer.part = 1;  
            }  
        } catch (RuntimeException e) {}  
    }  
};
```

```
ActionListener flistener = new ActionListener() {  
    public void actionPerformed(ActionEvent startlistener) {  
        try {  
            if(view.checkfunc( funcNum: 2) == 0) {  
                return;  
            }  
            if(timer.buzzer.on == true && timer.buzzer.funcType == 1) {  
                timer.buzzer.reqStop(timer.funcNum);  
                return;  
            }  
            if(timer.part == 0) {  
                return;  
            }  
            timer.setTimerPart(1);  
        } catch (RuntimeException e) {}  
    }  
};
```

3. Testing Revision

Fail #6. Lap & Split 확인 중 Reset 버튼을 누르면
Display에 불필요한 Lap GUI가 잔존함

→ Lap & Split 확인 중에는 Reset 버튼 Block 처리

```
ActionListener rlistener = new ActionListener() {  
    public void actionPerformed(ActionEvent startlistener) {  
        try {  
            if(view.checkfunc( funcNum: 3) == 0) {  
                return;  
            }  
            if(sw.paused == 3) } ←  
                return;  
            sw.resetStopwatch();  
            view.sec.setText("0");  
            view.sec10.setText("0");  
            view.min.setText("0");  
            view.min10.setText("0");  
            view.hour.setText("0");  
            view.hour10.setText("0");  
        } catch (RuntimeException e) {}  
    }  
};
```

— Thank You —

Q&A