2019 Software Modeling & Analysis
OOPT Stage 2050 & 2060.
【 Implementation & Unit Test 】

골라보 시계

Team #1

201411273 박재범

201411275 박진호

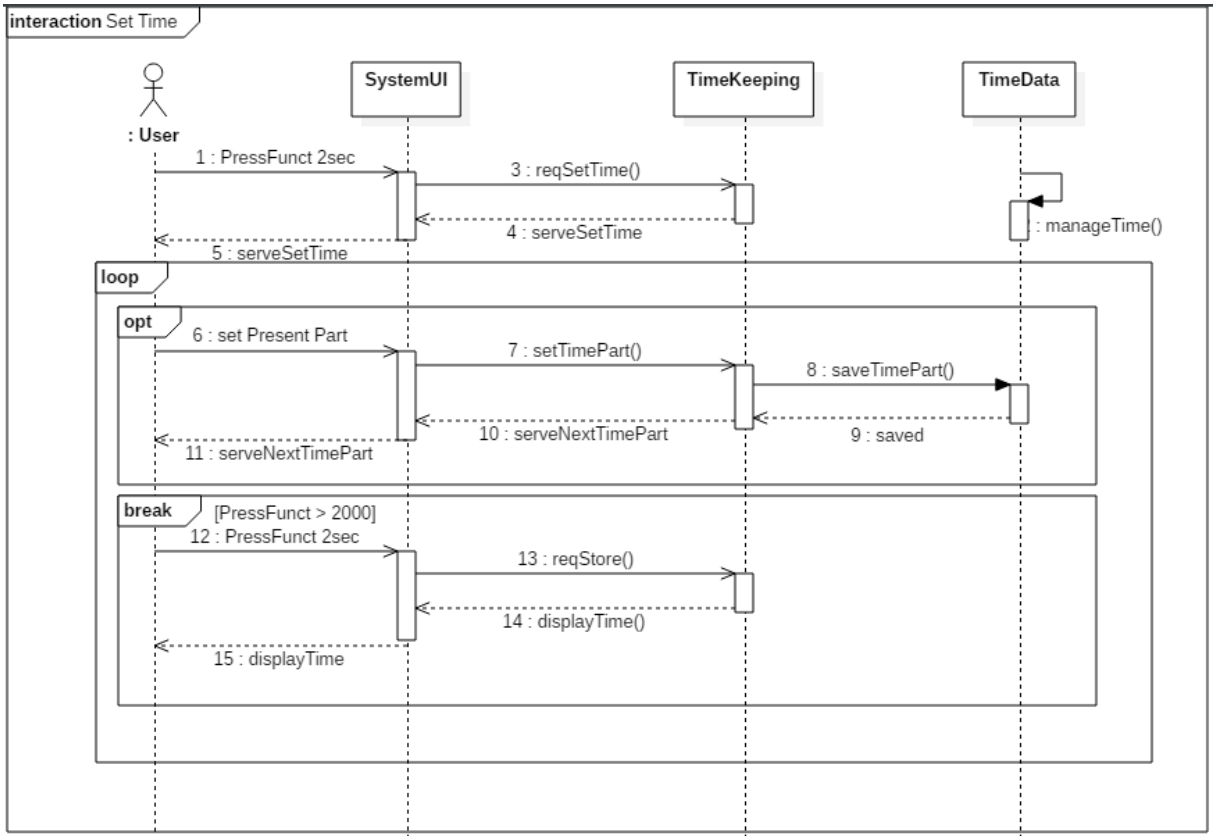
201411311 장원영

201311313 정인원

Contents

2051. Implements Windows

1. Set Time



Name	1. PressFunct 2sec
Responsibilities	Funct 버튼을 2초간 누른다.
Type	GUI
Cross Reference	R1.1, R1.2
Note	N/A
Pre-Conditions	TimeKeeping 기능을 선택한 상태여야 한다.
Post-Conditions	N/A

Name	5. serveSetTime
Responsibilities	시간 세부 설정 기능을 제공한다.
Type	GUI
Cross Reference	R1.1, R1.2
Note	N/A
Pre-Conditions	TimeKeeping 기능을 선택한 상태여야 한다.
Post-Conditions	N/A

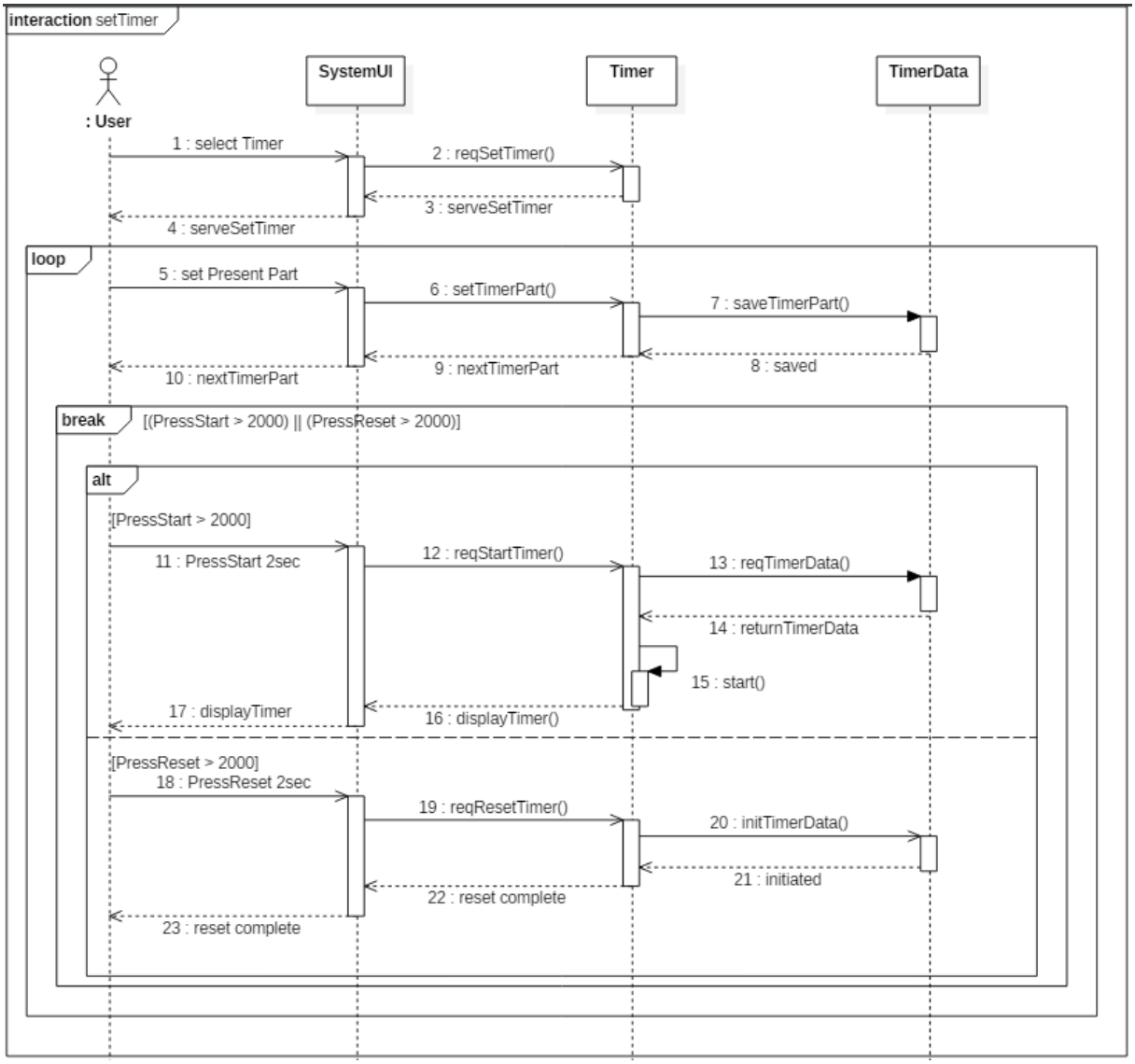
Name	6. set Present Part
Responsibilities	현재 설정중인 자릿수에 대한 설정값을 입력한다.
Type	GUI
Cross Reference	R1.1, R1.2
Note	N/A
Pre-Conditions	TimeKeeping 기능을 선택한 상태여야 한다.
Post-Conditions	N/A

Name	11. serveNextTimePart
Responsibilities	다음 자릿수를 설정할 수 있게 넘어간다.
Type	GUI
Cross Reference	R1.1, R1.2
Note	N/A
Pre-Conditions	자릿수 설정을 마쳐야 한다.
Post-Conditions	N/A

Name	12. PressFunct 2sec
Responsibilities	Funct 버튼을 2초간 누른다.
Type	GUI
Cross Reference	R1.1, R1.2
Note	N/A
Pre-Conditions	TimeKeeping 기능을 선택한 상태여야 한다.
Post-Conditions	설정 저장을 요청한다.

Name	15. displayTime
Responsibilities	설정된 시간을 출력한다.
Type	GUI
Cross Reference	R1.1, R1.2
Note	N/A
Pre-Conditions	N/A
Post-Conditions	TimeKeeping 초기 상태로 돌아간다.

2. Set Timer



Name	1. select Timer
Responsibilities	Mode 버튼을 눌러 Timer 기능을 선택한다.
Type	GUI
Cross Reference	R2.1, R2.2
Note	N/A
Pre-Conditions	N/A
Post-Conditions	N/A

Name	4. serveSetTimer
Responsibilities	Timer 시간 설정을 제공한다
Type	GUI
Cross Reference	R2.1, R2.2
Note	N/A
Pre-Conditions	N/A
Post-Conditions	N/A

Name	5. set Present Part
Responsibilities	현재 설정중인 자릿수에 대한 설정값을 입력한다.
Type	GUI
Cross Reference	R2.1, R2.2
Note	N/A
Pre-Conditions	N/A
Post-Conditions	N/A

Name	10. nextTimerPart
Responsibilities	다음 자릿수를 설정할 수 있게 넘어간다.
Type	GUI
Cross Reference	R2.1, R2.2
Note	N/A
Pre-Conditions	N/A
Post-Conditions	N/A

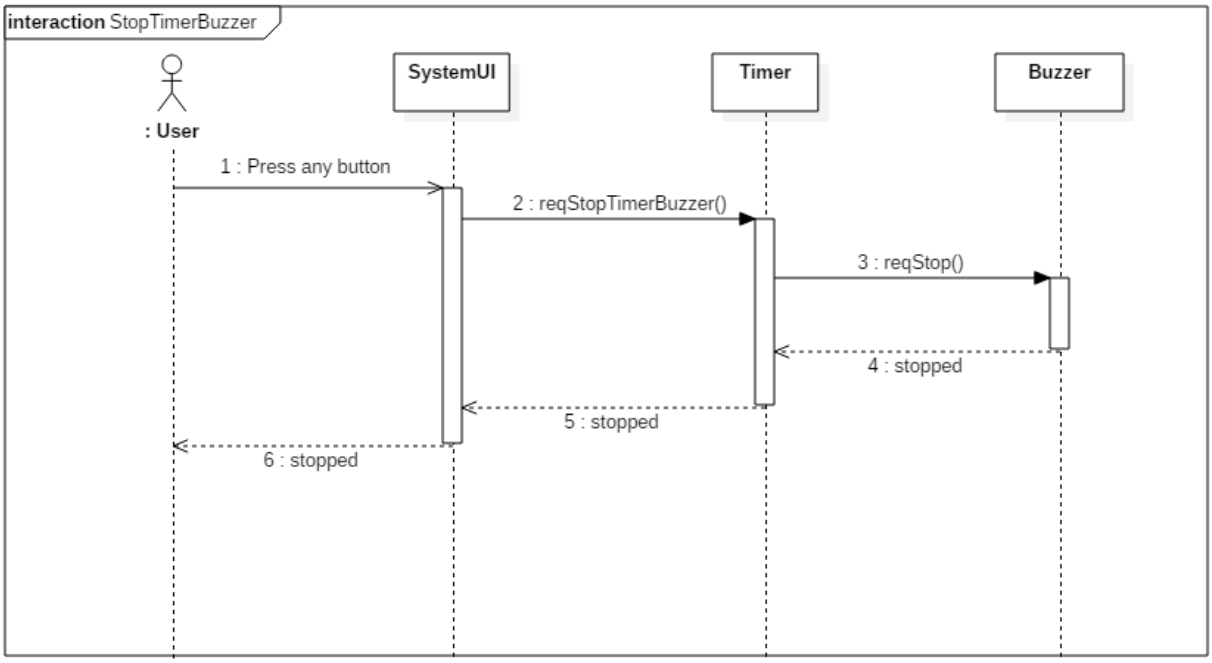
Name	11. PressStart 2sec
Responsibilities	Start 버튼을 2초간 누른다.
Type	GUI
Cross Reference	R2.1, R2.2
Note	N/A
Pre-Conditions	N/A
Post-Conditions	설정한 시간으로부터 타이머를 시작하게 된다.

Name	17. displayTimer
Responsibilities	타이머를 시작하는 것을 출력한다.
Type	GUI
Cross Reference	R2.1, R2.2
Note	N/A
Pre-Conditions	N/A
Post-Conditions	타이머가 진행된다.

Name	18. PressReset 2sec
Responsibilities	Reset 버튼을 2초 누른다.
Type	GUI
Cross Reference	R2.1, R2.2
Note	N/A
Pre-Conditions	N/A
Post-Conditions	타이머 설정을 초기화한다.

Name	12. reset complete
Responsibilities	Reset을 완료하고 초기 화면을 출력한다.
Type	GUI
Cross Reference	R2.1, R2.2
Note	N/A
Pre-Conditions	N/A
Post-Conditions	초기 타이머 화면으로 돌아간다.

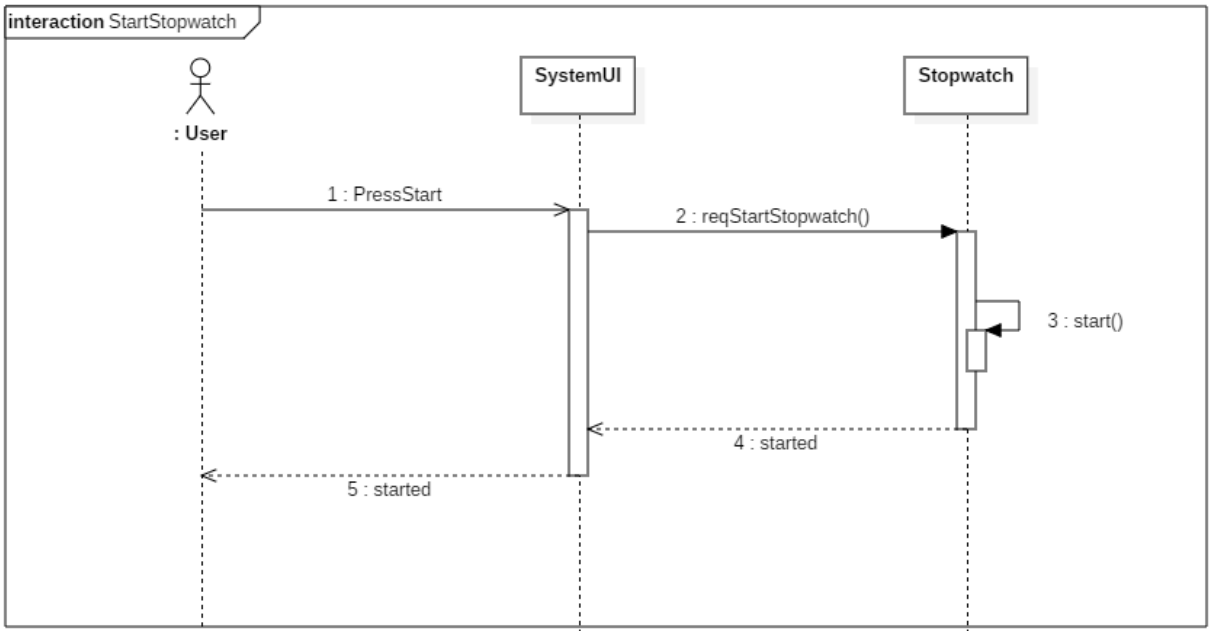
3. Stop Timer Buzzer



Name	1. Press any button
Responsibilities	Mode를 제외한 아무 버튼이나 누른다.
Type	GUI
Cross Reference	R2.2, R2.3
Note	N/A
Pre-Conditions	Timer의 버저가 울리고 있어야 한다.
Post-Conditions	N/A

Name	6. stopped
Responsibilities	타이머의 버저를 중지한다.
Type	GUI
Cross Reference	R2.2, R2.3
Note	N/A
Pre-Conditions	Timer의 버저가 울리고 있어야 한다.
Post-Conditions	N/A

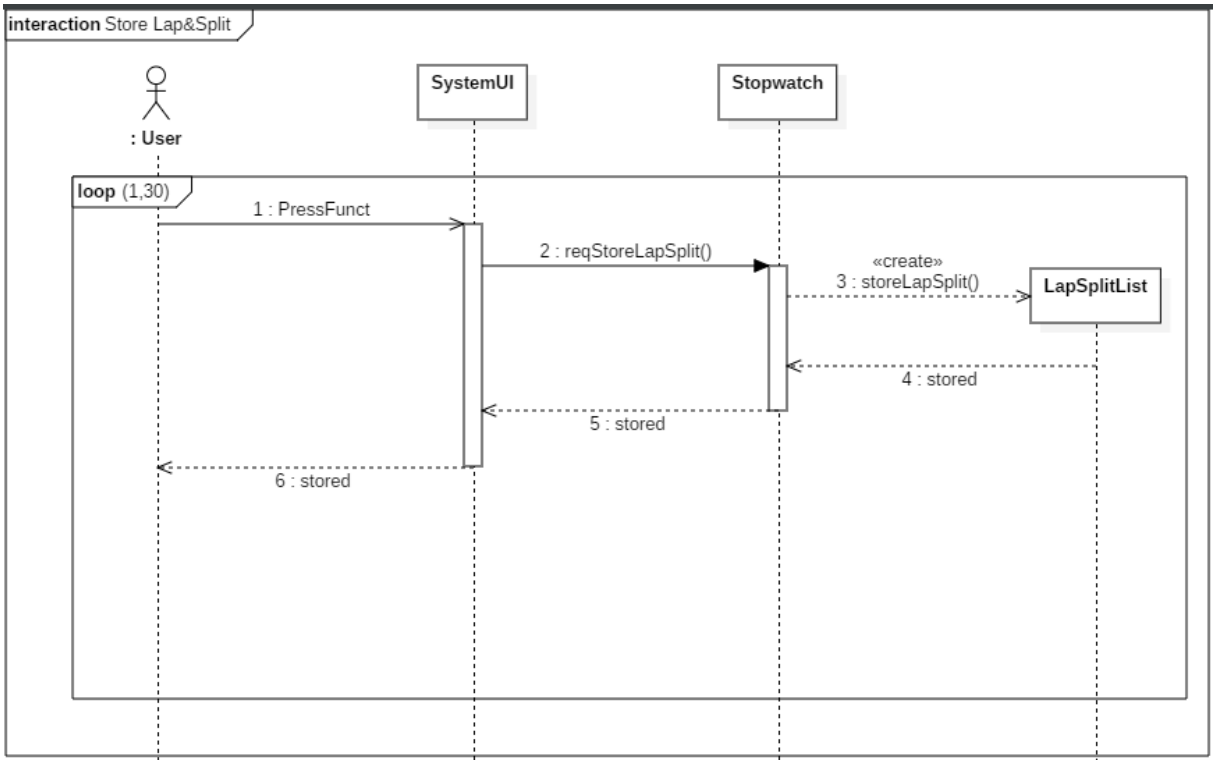
4. Start Stopwatch



Name	1. PressStart
Responsibilities	Start 버튼을 누른다.
Type	GUI
Cross Reference	R3.1
Note	N/A
Pre-Conditions	Stopwatch 기능이 선택되어 있어야 한다.
Post-Conditions	N/A

Name	5. started
Responsibilities	스톱워치를 시작한다.
Type	GUI
Cross Reference	R3.1
Note	N/A
Pre-Conditions	Stopwatch 기능이 선택되어 있어야 한다.
Post-Conditions	N/A

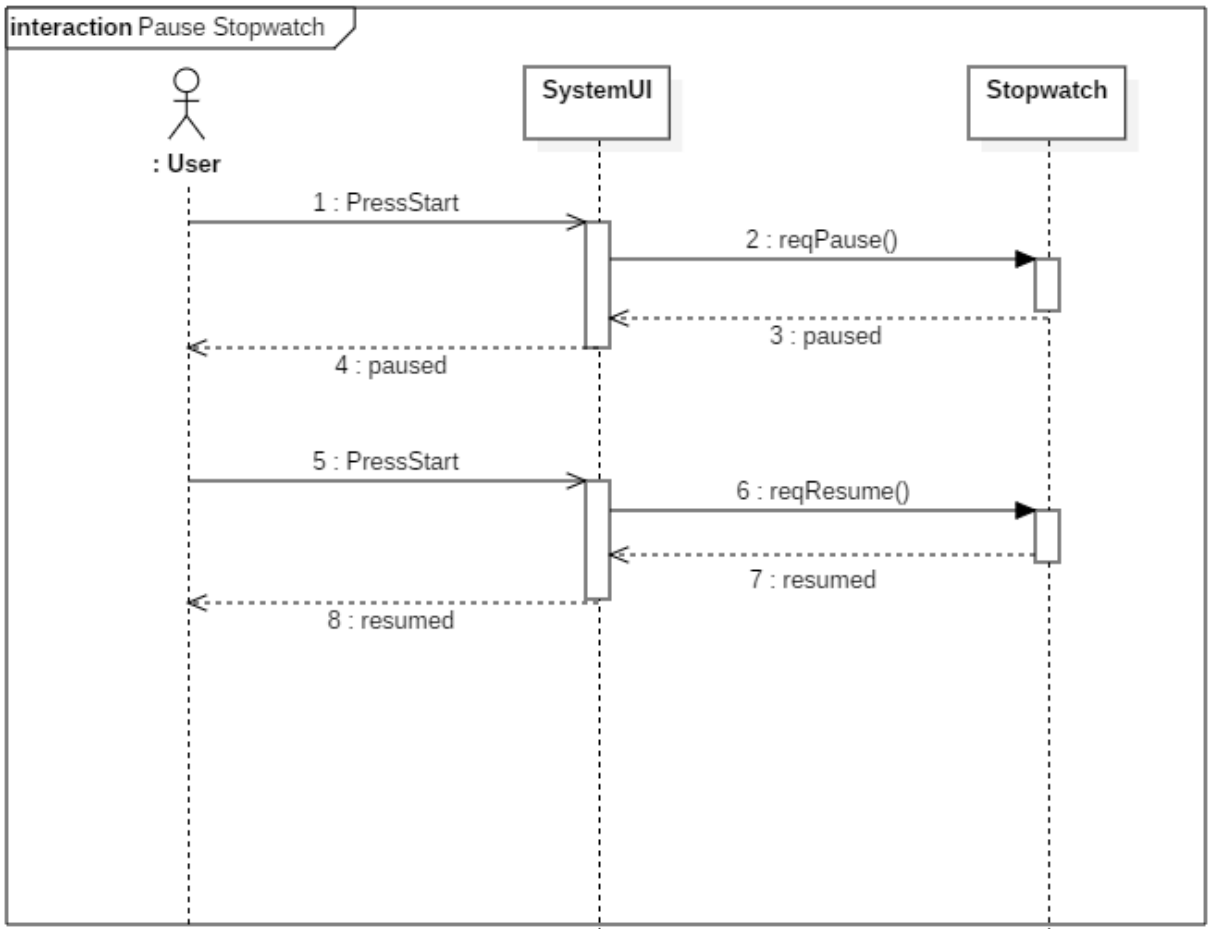
5. Store Lap & Split



Name	1. PressFunct
Responsibilities	Funct 버튼을 누른다.
Type	GUI
Cross Reference	R3.2
Note	N/A
Pre-Conditions	스톱워치가 진행중인 상태여야 한다.
Post-Conditions	N/A

Name	6. stored
Responsibilities	저장 후 진행 중인 스톱워치로 복귀한다.
Type	GUI
Cross Reference	R3.2
Note	N/A
Pre-Conditions	스톱워치가 진행중인 상태여야 한다.
Post-Conditions	N/A

6. Pause Stopwatch



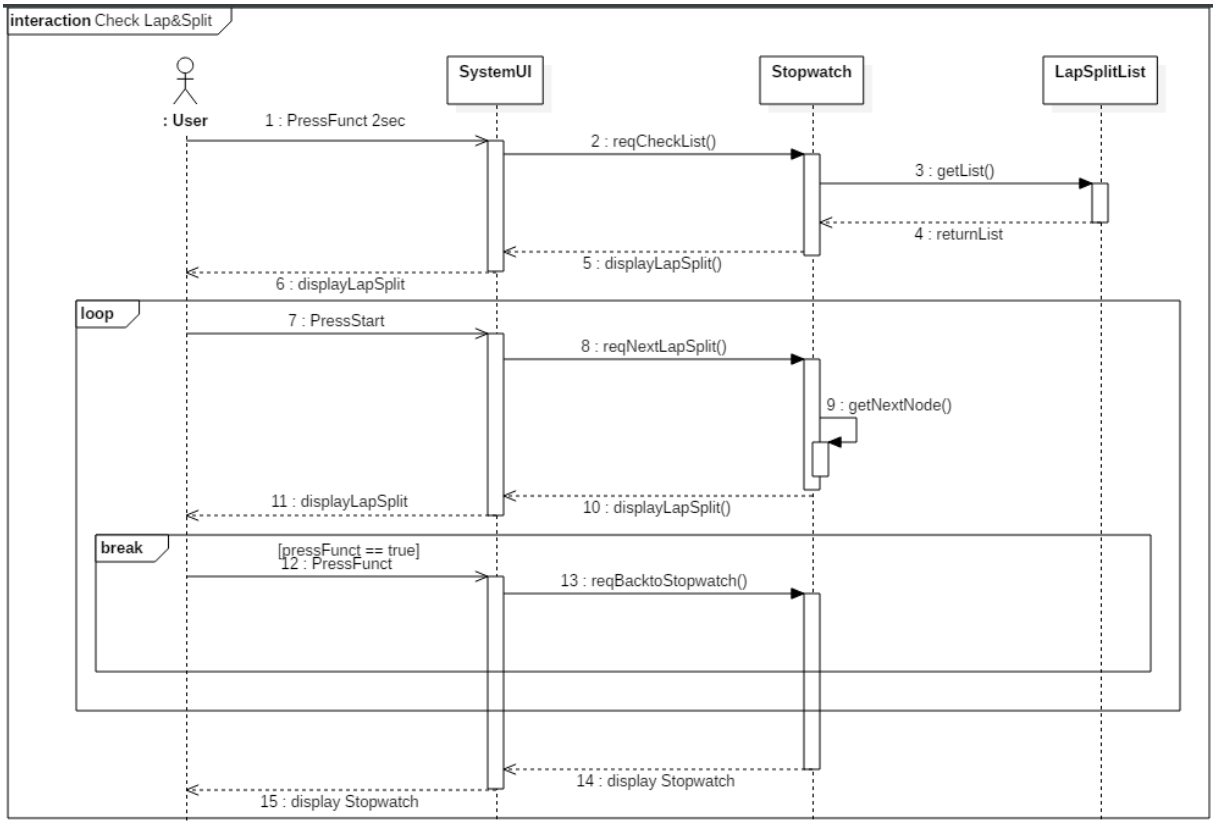
Name	1. PressStart
Responsibilities	Start 버튼을 누른다.
Type	GUI
Cross Reference	R3.3
Note	N/A
Pre-Conditions	스톱워치가 진행중인 상태여야 한다.
Post-Conditions	N/A

Name	4. paused
Responsibilities	스톱워치를 일시정지 시킨다.
Type	GUI
Cross Reference	R3.3
Note	N/A
Pre-Conditions	스톱워치가 진행중인 상태여야 한다.
Post-Conditions	N/A

Name	5. PressStart
Responsibilities	Start 버튼을 누른다.
Type	GUI
Cross Reference	R3.3
Note	N/A
Pre-Conditions	스톱워치가 일시정지된 상태여야 한다.
Post-Conditions	N/A

Name	8. resumed
Responsibilities	스톱워치를 재개시킨다.
Type	GUI
Cross Reference	R3.3
Note	N/A
Pre-Conditions	스톱워치가 일시정지된 상태여야 한다.
Post-Conditions	N/A

7. Check Lap & Split



Name	1. PressFunct 2sec
Responsibilities	Funct 버튼을 2초간 누른다.
Type	GUI
Cross Reference	R3.4
Note	N/A
Pre-Conditions	스톱워치가 일시정지 상태여야 한다.
Post-Conditions	N/A

Name	6. displayLapSplit
Responsibilities	LapSplitList의 정보를 순서에 맞게 출력한다.
Type	GUI
Cross Reference	R3.4
Note	N/A
Pre-Conditions	스톱워치가 일시정지 상태여야 한다.
Post-Conditions	N/A

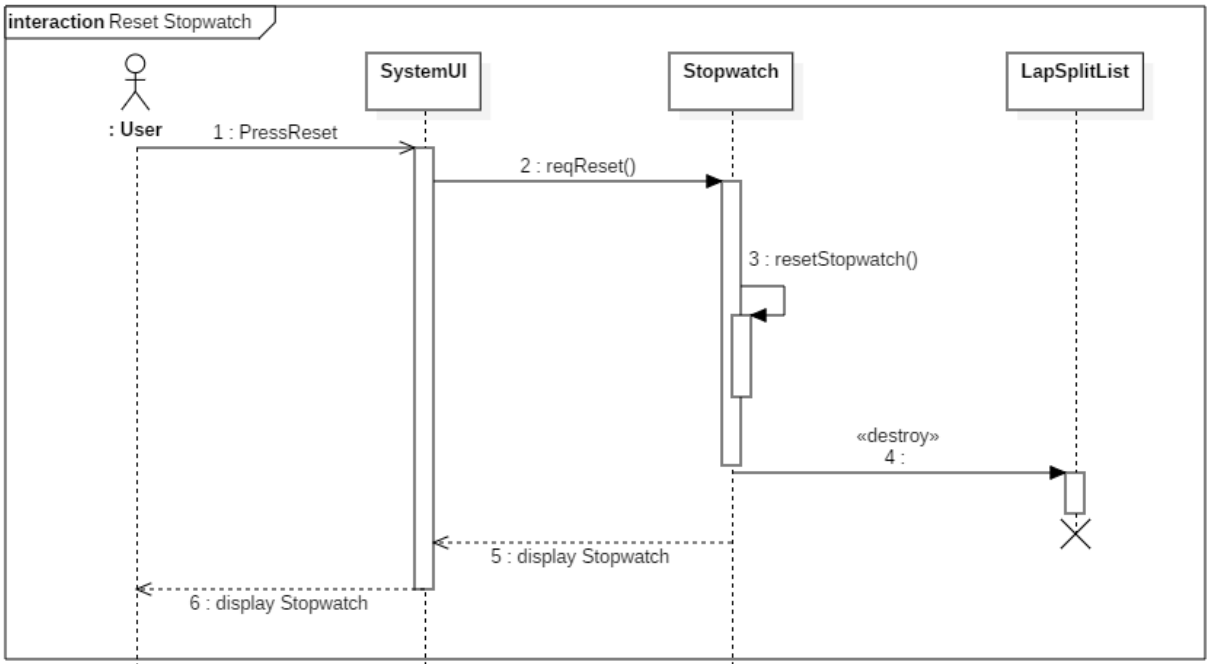
Name	7. PressStart
Responsibilities	Start 버튼을 누른다.
Type	GUI
Cross Reference	R3.4
Note	N/A
Pre-Conditions	Lap, Split 데이터를 열람중인 상태여야 한다.
Post-Conditions	N/A

Name	11. displayLapSplit
Responsibilities	LapSplitList의 정보를 순서에 맞게 출력한다.
Type	GUI
Cross Reference	R3.4
Note	N/A
Pre-Conditions	Lap, Split 데이터를 열람중인 상태여야 한다.
Post-Conditions	N/A

Name	12. PressFunct
Responsibilities	Funct 버튼을 누른다.
Type	GUI
Cross Reference	R3.4
Note	N/A
Pre-Conditions	Lap, Split 데이터를 열람중인 상태여야 한다.
Post-Conditions	열람을 종료한다.

Name	15. display Stopwatch
Responsibilities	멈춰 있는 스톱워치 화면으로 다시 돌아간다.
Type	GUI
Cross Reference	R3.4
Note	N/A
Pre-Conditions	Lap, Split 데이터를 열람중인 상태여야 한다.
Post-Conditions	N/A

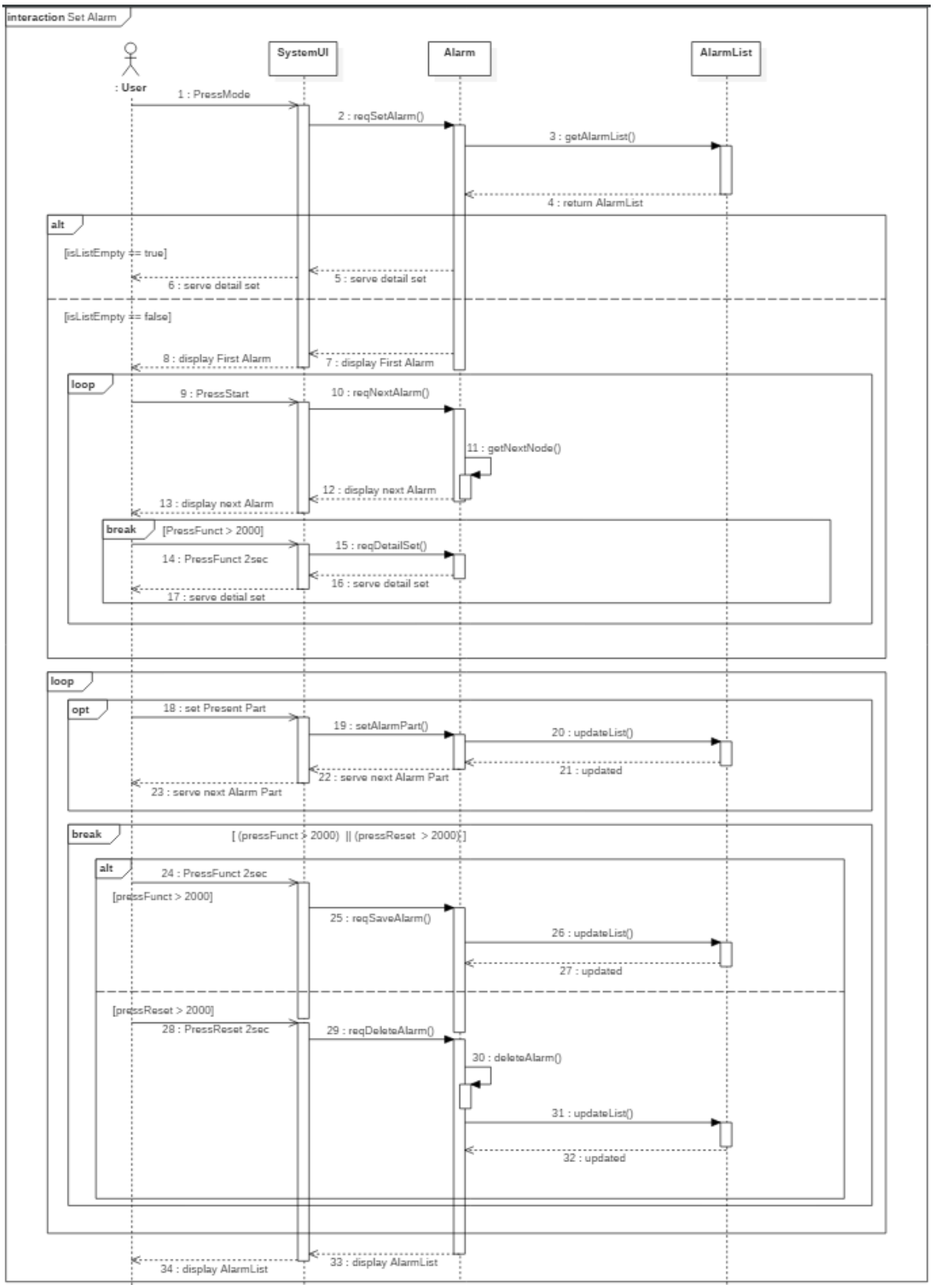
8. Reset Stopwatch



Name	1. PressReset
Responsibilities	Reset 버튼을 누른다.
Type	GUI
Cross Reference	R3.5
Note	N/A
Pre-Conditions	스톱워치가 일시정지된 상태여야 한다.
Post-Conditions	N/A

Name	6. display Stopwatch
Responsibilities	Reset 버튼을 누른다.
Type	GUI
Cross Reference	R3.5
Note	스톱워치를 초기화한 뒤 초기 화면으로 돌아가는 작업을 한다.
Pre-Conditions	스톱워치가 일시정지된 상태여야 한다.
Post-Conditions	N/A

9. Set Alarm


interaction Set Alarm
: User
SystemUI
Alarm
AlarmList
1 : PressMode
2 : reqSetAlarm()
3 : getAlarmList()
4 : return AlarmList
alt
[isListEmpty == true]
6 : serve detail set
5 : serve detail set
[isListEmpty == false]
8 : display First Alarm
7 : display First Alarm
loop
9 : PressStart
10 : reqNextAlarm()
11 : getNextNode()
12 : display next Alarm
13 : display next Alarm
break
[PressFunct > 2000]
14 : PressFunct 2sec
15 : reqDetailSet()
16 : serve detail set
17 : serve detial set
loop
opt
18 : set Present Part
19 : setAlarmPart()
20 : updateList()
21 : updated
22 : serve next Alarm Part
23 : serve next Alarm Part
break
[(pressFunct > 2000) || (pressReset > 2000)]
alt
24 : PressFunct 2sec
[pressFunct > 2000]
25 : reqSaveAlarm()
26 : updateList()
27 : updated
[pressReset > 2000]
28 : PressReset 2sec
29 : reqDeleteAlarm()
30 : deleteAlarm()
31 : updateList()
32 : updated
34 : display AlarmList
33 : display AlarmList

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

Name	17. serve detail set
Responsibilities	선택한 알람의 세부 설정 기능을 제공한다.
Type	GUI
Cross Reference	R4.1, R4.2
Note	N/A
Pre-Conditions	알람 리스트에 데이터가 존재해야 한다.
Post-Conditions	N/A

Name	18. set Present Part
Responsibilities	선택한 알람의 한 부분에 대한 설정을 한다.
Type	GUI
Cross Reference	R4.1, R4.2
Note	N/A
Pre-Conditions	알람 리스트에 데이터가 존재해야 한다.
Post-Conditions	N/A

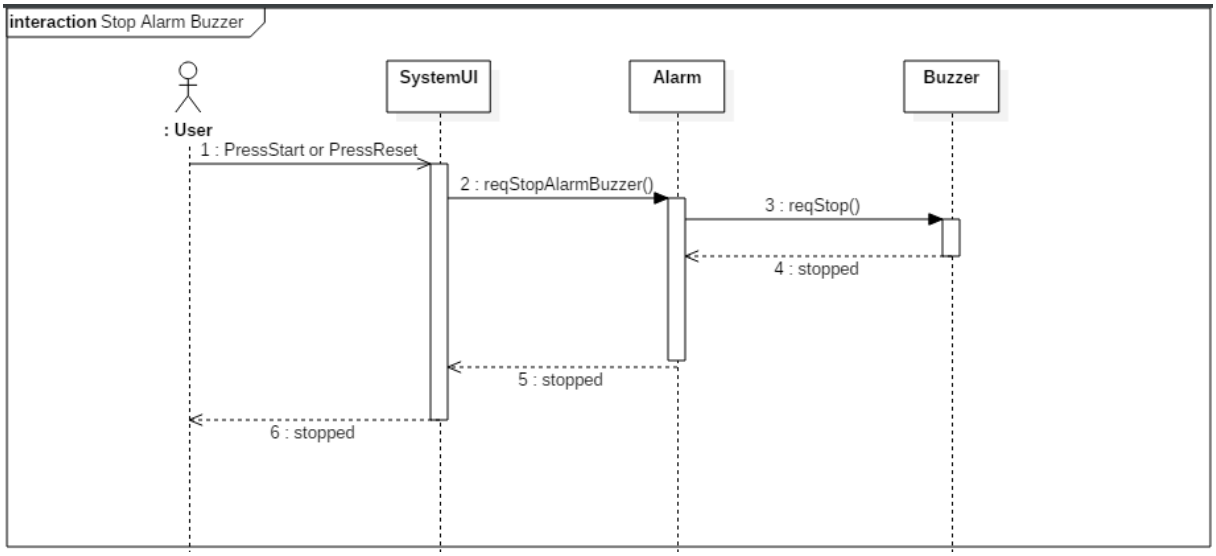
Name	23. serve next Alarm Part
Responsibilities	선택한 알람의 다음 부분에 대한 설정을 제공한다.
Type	GUI
Cross Reference	R4.1, R4.2
Note	N/A
Pre-Conditions	알람 리스트에 데이터가 존재해야 한다.
Post-Conditions	N/A

Name	24. PressFunct 2sec
Responsibilities	Funct 버튼을 2초간 누른다.
Type	GUI
Cross Reference	R4.1, R4.2
Note	N/A
Pre-Conditions	N/A
Post-Conditions	알람 설정이 저장된다.

Name	28. PressReset 2sec
Responsibilities	Reset 버튼을 2초간 누른다.
Type	GUI
Cross Reference	R4.1, R4.2
Note	N/A
Pre-Conditions	N/A
Post-Conditions	현재 알람을 리스트에서 삭제한다.

Name	34. display AlarmList
Responsibilities	알람 리스트 열람 화면으로 돌아간다.
Type	GUI
Cross Reference	R4.1, R4.2
Note	N/A
Pre-Conditions	N/A
Post-Conditions	N/A

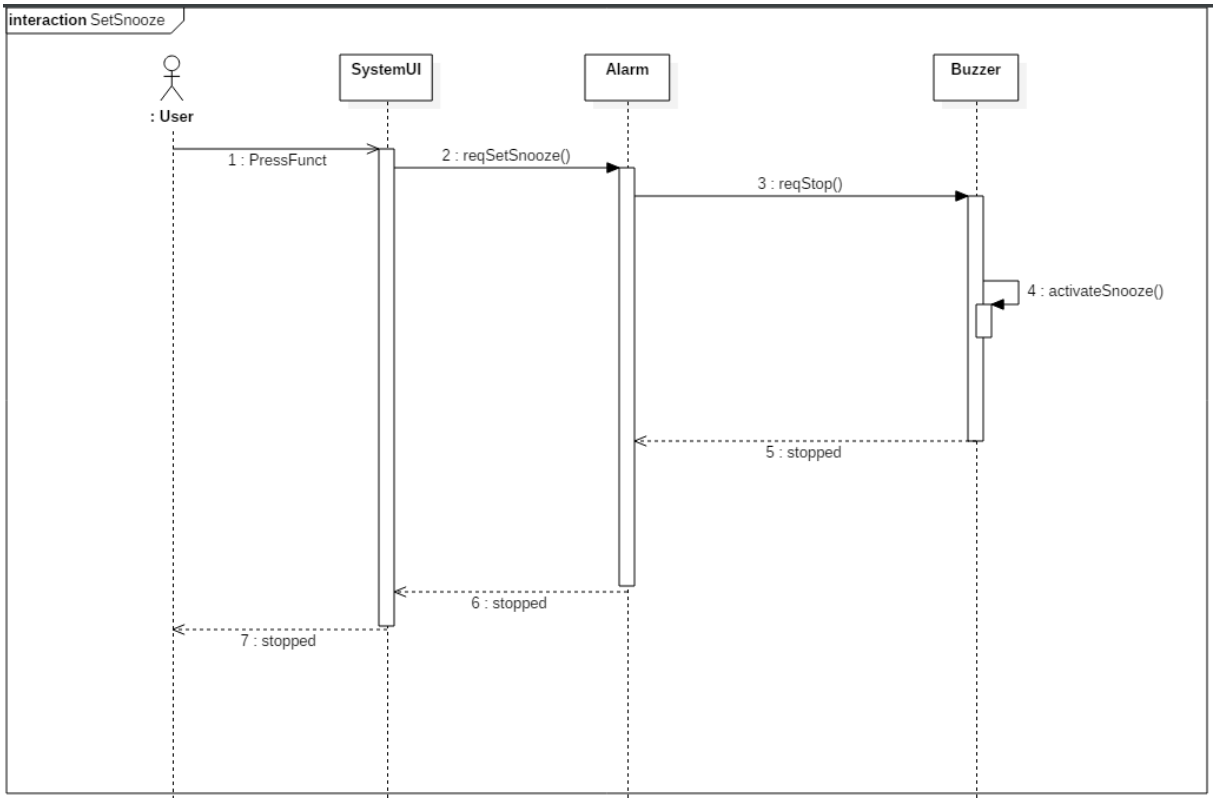
10. Stop Alarm Buzzer



Name	1. PressStart or PressReset
Responsibilities	Start 또는 Reset 버튼을 누른다.
Type	GUI
Cross Reference	R4.2, R4.3
Note	N/A
Pre-Conditions	알람의 Buzzer가 울리고 있는 상태여야 한다.
Post-Conditions	N/A

Name	6. stopped
Responsibilities	알람의 Buzzer를 즉시 정지시킨다.
Type	GUI
Cross Reference	R4.2, R4.3
Note	N/A
Pre-Conditions	알람의 Buzzer가 울리고 있는 상태여야 한다.
Post-Conditions	N/A

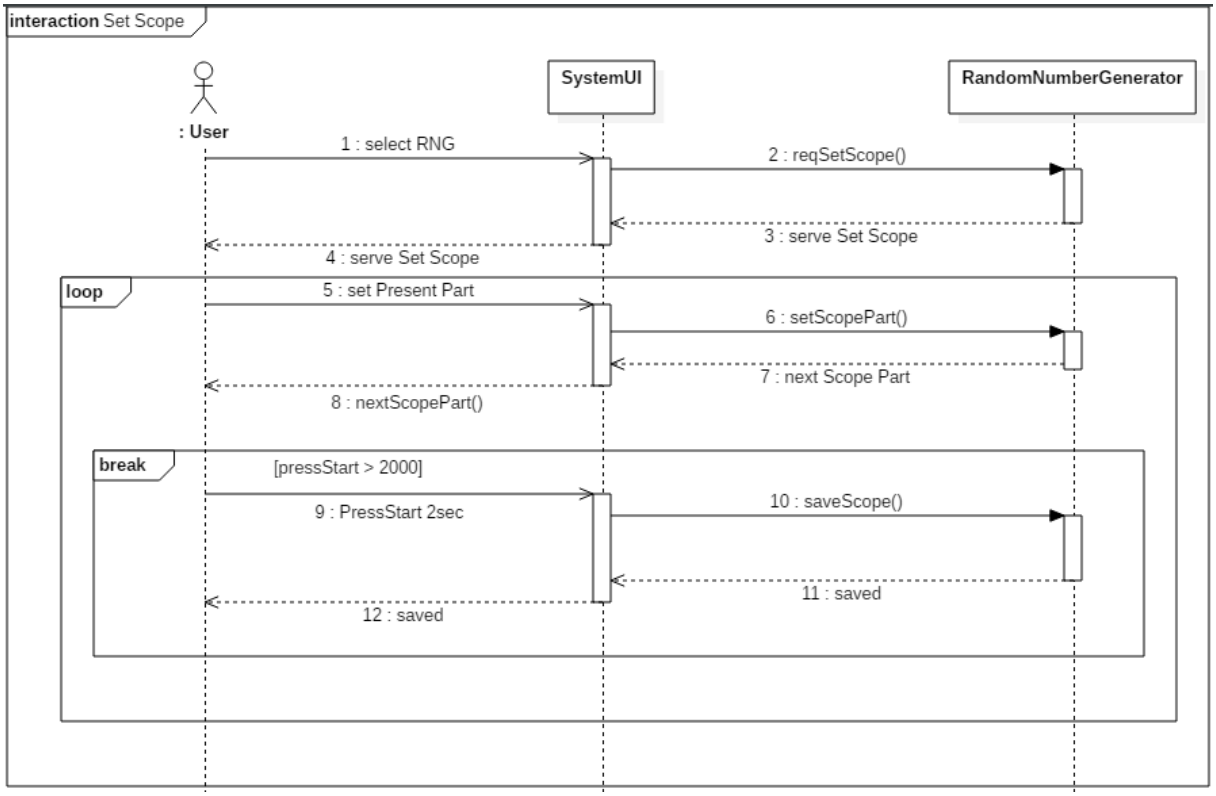
11. Set Snooze



Name	1. PressFunct
Responsibilities	Funct 버튼을 누른다
Type	GUI
Cross Reference	R4.2, R4.4
Note	N/A
Pre-Conditions	알람의 Buzzer가 울리고 있는 상태여야 한다.
Post-Conditions	N/A

Name	7. stopped
Responsibilities	알람의 Buzzer를 즉시 정지시킨다.
Type	GUI
Cross Reference	R4.2, R4.4
Note	내부적으로 현재 시각으로부터 5분 뒤에 임시 알람을 생성한다.
Pre-Conditions	알람의 버저가 울리는 상태여야 한다.
Post-Conditions	N/A

12. Set Scope



Name	1. select RNG
Responsibilities	Mode 버튼을 눌러 RandomNumberGenerator 기능을 선택한다.
Type	GUI
Cross Reference	R5.1, R5.2
Note	N/A
Pre-Conditions	N/A
Post-Conditions	N/A

Name	4. serve Set Scope
Responsibilities	현재 설정중인 자릿수에 대한 설정값을 입력한다.
Type	GUI
Cross Reference	R5.1, R5.2
Note	N/A
Pre-Conditions	RandomNumberGenerator 기능을 선택한 상태여야 한다.
Post-Conditions	N/A

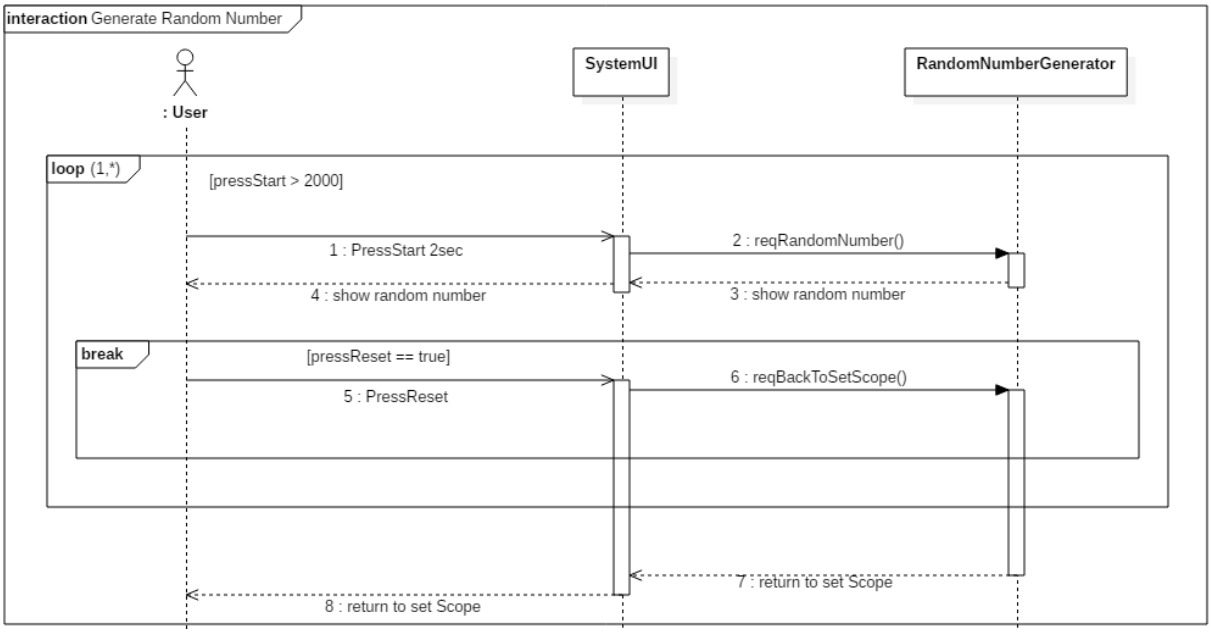
Name	5. set Present Part
Responsibilities	현재
Type	GUI
Cross Reference	R5.1, R5.2
Note	N/A
Pre-Conditions	RandomNumberGenerator 기능을 선택한 상태여야 한다.
Post-Conditions	N/A

Name	8. nextScopePart
Responsibilities	다음 자릿수를 설정할 수 있게 넘어간다.
Type	GUI
Cross Reference	R5.1, R5.2
Note	N/A
Pre-Conditions	RandomNumberGenerator 기능을 선택한 상태여야 한다.
Post-Conditions	N/A

Name	9. PressStart 2sec
Responsibilities	Start 버튼을 2초간 누른다.
Type	GUI
Cross Reference	R5.1, R5.2
Note	N/A
Pre-Conditions	RandomNumberGenerator 기능을 선택한 상태여야 한다.
Post-Conditions	N/A

Name	12. saved
Responsibilities	난수 범위 설정을 저장했음을 알린다.
Type	GUI
Cross Reference	R5.1, R5.2
Note	N/A
Pre-Conditions	RandomNumberGenerator 기능을 선택한 상태여야 한다.
Post-Conditions	설정된 범위 안에서 난수를 생성한다.

13. Generate Random Number



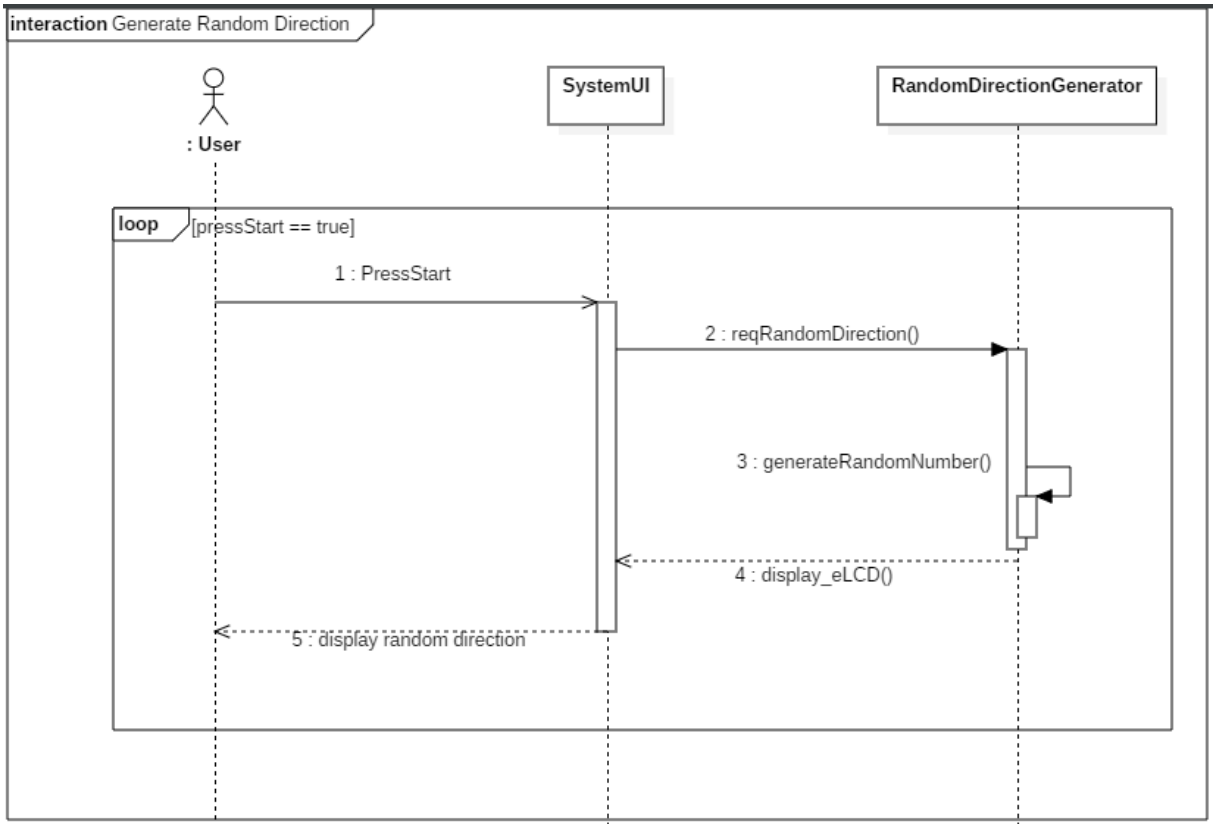
Name	1. PressStart 2sec
Responsibilities	Start 버튼을 2초간 누른다.
Type	GUI
Cross Reference	R5.1, R5.2
Note	N/A
Pre-Conditions	Set Scope를 진행중이어야 한다.
Post-Conditions	N/A

Name	4. show random number
Responsibilities	설정한 범위 내의 난수를 생성해 출력한다.
Type	GUI
Cross Reference	R5.1, R5.2
Note	N/A
Pre-Conditions	Set Scope를 진행중이어야 한다.
Post-Conditions	N/A

Name	5. PressReset
Responsibilities	Reset 버튼을 누른다.
Type	GUI
Cross Reference	R5.1, R5.2
Note	N/A
Pre-Conditions	N/A
Post-Conditions	N/A

Name	8. return to set Scope
Responsibilities	설정을 초기화하고 Set Scope로 다시 돌아간다.
Type	GUI
Cross Reference	R5.1, R5.2
Note	N/A
Pre-Conditions	N/A
Post-Conditions	Set Scope로 돌아간다.

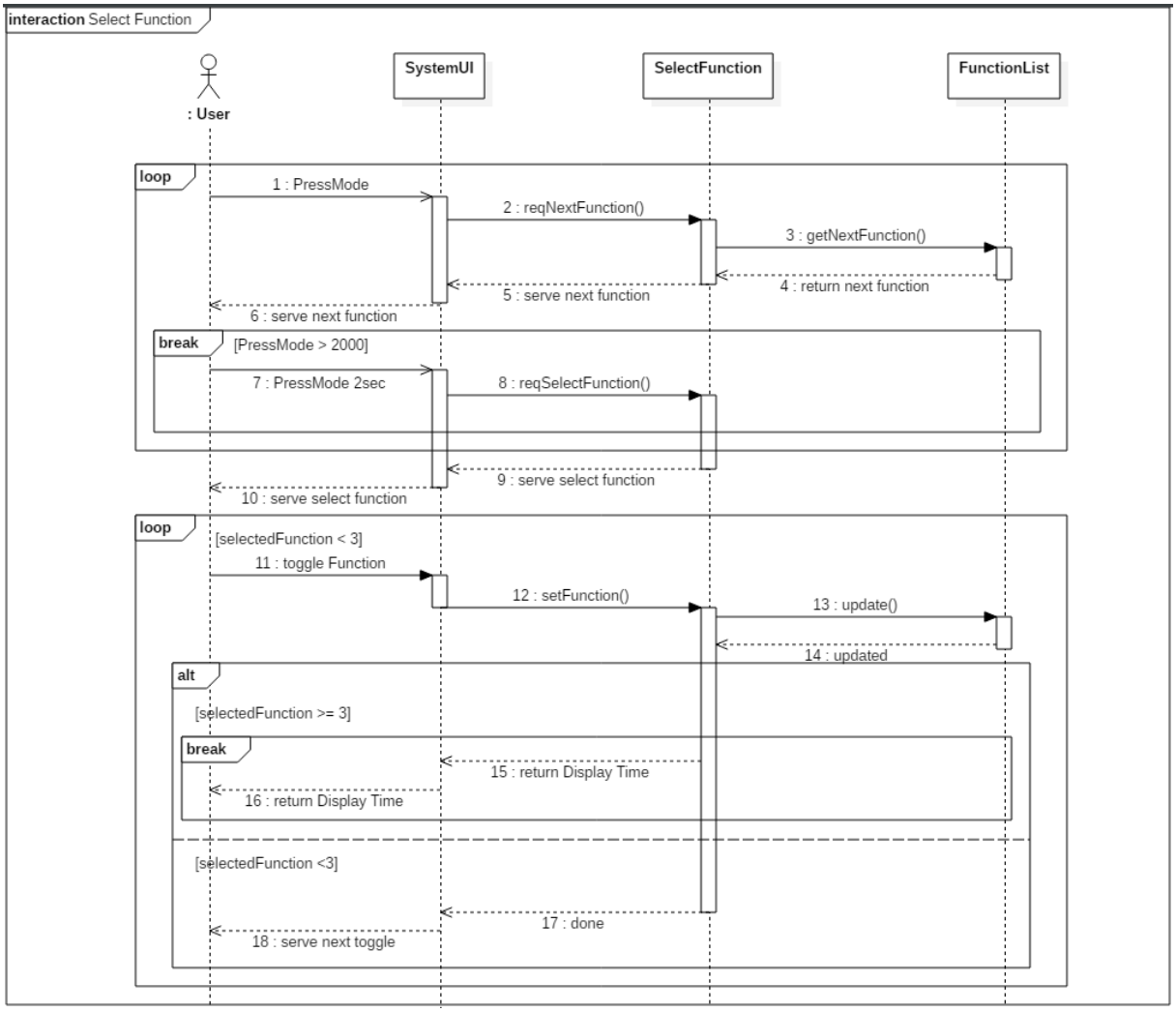
14. Generate Random Direction



Name	1. PressStart
Responsibilities	Start 버튼을 누른다.
Type	GUI
Cross Reference	R6.1
Note	N/A
Pre-Conditions	RandomDirectionGenerator 기능이 선택된 상태여야 한다.
Post-Conditions	N/A

Name	5. display random direction
Responsibilities	테두리의 LCD에 랜덤한 위치를 출력한다.
Type	GUI
Cross Reference	R6.1
Note	N/A
Pre-Conditions	RandomDirectionGenerator 기능이 선택된 상태여야 한다.
Post-Conditions	N/A

15. Select Function



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

Name	6. serve next function
Responsibilities	다음 기능으로 넘어간다.
Type	GUI
Cross Reference	R7.1
Note	현재가 마지막 기능이라면 다시 첫번째 기능으로 돌아간다.
Pre-Conditions	N/A
Post-Conditions	N/A

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

2052. Write Unit Test Code

1. TimeKeepingTest

TimeKeepingTest.java

```java
import org.junit.Test;

import static org.junit.Assert.*;

public class TimeKeepingTest {

    @Test
    public void setTimePart() {
        View v = new View();
        TimeKeeping tk = new TimeKeeping(v);
        tk.part = 6; // hour10
        tk.time.format= 2; // time format 24h
        tk.time.hour = 5;
        tk.saveValue = 2;
        tk.setTimePart(0);

        assertEquals( expected: 0, tk.saveValue);
    }

    @Test
    public void southFinder() {
        View v = new View();
        TimeKeeping tk = new TimeKeeping(v);
        tk.time.format =0;
        tk.southFinder( min: "0", min10: "3", hour: "2", hour10: "1");

        assertEquals( expected: 1,tk.eLCD[1]);
        assertEquals( expected: 1,tk.eLCD[2]);
    }
}
```

2. TimeDataTest

TimeDataTest.java

```java
import org.junit.Test;

import static org.junit.Assert.*;

public class TimeDataTest {

    @Test
    public void saveTimePart() {
        View v = new View();
        AlarmList al = new AlarmList(v);
        Buzzer b = new Buzzer(v,al);
        TimeData td = new TimeData(b);

        //part = 5 (hour), value = 2
        td.saveTimePart(5,2);
        assertEquals(2, td.hour);
    }

}
```

3. TimerTest

TimerTest.java

```java
import org.junit.Test;

import static org.junit.Assert.*;

public class TimerTest {

    @Test
    public void setTimerPart() {
        View v = new View();
        Buzzer b = new Buzzer(v);
        Timer tm = new Timer(v,b);

        tm.part = 1; // sec
        tm.saveValue = 5;

        tm.setTimerPart(0);

        assertEquals(tm.timer.sec , tm.saveValue);

    }
}
```

4. BuzzerTest

BuzzerTest.java

```java
import org.junit.Test;

import static org.junit.Assert.*;

public class BuzzerTest {

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

        b.reqStop(1);
        assertEquals(0, b.funcType);
    }

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

        b.on = true;
        b.beep(1); // Timer beep
        assertEquals(1, b.funcType);
    }

}
```

5. StopwatchTest

StopwatchTest.java

```java
import org.junit.Test;

import static org.junit.Assert.*;

public class StopwatchTest {

    @Test
    public void reqPause() {
        View v = new View();
        Stopwatch s = new Stopwatch(v);

        assertEquals(1,s.paused);
    }

    @Test
    public void reqResume() {
        View v = new View();
        Stopwatch s = new Stopwatch(v);

        assertEquals(1,s.paused);
    }

    @Test
    public void reqBackToStopwatch() {
        View v = new View();
        Stopwatch s = new Stopwatch(v);

        assertEquals(1,s.paused);
    }

    @Test
    public void resetStopwatch() {
        View v = new View();
        Stopwatch s = new Stopwatch(v);

        assertEquals(0, s.sec);
        assertEquals(0, s.milsec);
        assertEquals(0,s.min);

    }

}
```

6. LapSplitListTest

LapSplitListTest.java

```java
import org.junit.Test;

import java.util.Vector;

import static org.junit.Assert.*;

public class LapSplitListTest {

    @Test
    public void storeLapSplit() {

        LapSplitList ls = new LapSplitList();

        Vector v = new Vector();
        v.add(1);
        v.add(2);
        v.add(3);
        ls.storeLapSplit(v);

        assertEquals(1, ls.lns.elementAt(0));
        assertEquals(2, ls.lns.elementAt(1));
        assertEquals(3, ls.lns.elementAt(2)); // LAP
        assertEquals(1, ls.lns.elementAt(3));
        assertEquals(2, ls.lns.elementAt(4));
        assertEquals(3, ls.lns.elementAt(5)); // SPLIT
    }
}
```

7. AlarmTest

AlarmTest.java

```java
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);
```

```java

        alm.part=1;
        alm.sec = 8;
        alm.setAlarmPart(0);
        assertEquals(9, 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);
    }
}
```

8. RandomNumberGeneratorTest

RandomNumberGeneratorTest.java

```java
import org.junit.Test;
import static org.junit.Assert.*;
public class RandomNumberGeneratorTest {

    @Test
    public void reqRandomNumber() {

        View v = new View();
        RandomNumberGenerator rng = new RandomNumberGenerator(v);
        int result = rng.reqRandomNumber(100);
        int flag;
        if(result>=0 && result<100)
            flag = 1;
        else
            flag =0;
        assertEquals(flag,1);
    }

    @Test
    public void reqBackToSetScope()
    {
        View v = new View();
        RandomNumberGenerator rng = new RandomNumberGenerator(v);
        rng.reqBackToSetScope();
        assertEquals(1,rng.part);
    }

    @Test
    public void saveScope()
    {
        View v = new View();
        RandomNumberGenerator rng = new RandomNumberGenerator(v);
        rng.part = 2;
        rng.saveScope();
        assertEquals(0, rng.saveValue);
    }

    @Test
    public void setScopePart()
    {
        View v = new View();
        RandomNumberGenerator rng = new RandomNumberGenerator(v);
        rng.setScopePart(0);
        rng.part = 2; // sec10
        rng.saveValue = 8;

        assertEquals(8,rng.saveValue);
    }
}
```

9. RandomDirectionGeneratorTest

RandomDirectionGeneratorTest.java

```java
import org.junit.Test;

import java.util.Random;

import static org.junit.Assert.*;

public class RandomDirectionGeneratorTest {

    @Test
    public void reqRandomDirection() {
        View v = new View();
        RandomDirectionGenerator rdg = new RandomDirectionGenerator(v);
        rdg.reqRandomDirection();

        int flag =0;
        if(rdg.result >= 0 && rdg.result <=60)
            flag =1;

        assertEquals(1, flag);


    }

    @Test
    public void display_eLCD() {
        View v = new View();
        RandomDirectionGenerator rdg = new RandomDirectionGenerator(v);

        rdg.result = 10;
        rdg.display_eLCD();
        assertEquals(1, rdg.eLCD[rdg.result-1]);
    }
}
```

10. SelectFunctionTest

SelectFunctionTest.java

```java
import org.junit.Test;

import static org.junit.Assert.*;

public class SelectFunctionTest {

    @Test
    public void reqSelectFunction() {
        View v = new View();
        AlarmList al = new AlarmList(v);
        Buzzer b = new Buzzer(v,al);
        Alarm a = new Alarm(v,al,b);
        RandomDirectionGenerator rdg = new RandomDirectionGenerator(v);
        RandomNumberGenerator rng = new RandomNumberGenerator(v);
        Stopwatch s = new Stopwatch(v);
        Timer t = new Timer(v,b);
        TimeKeeping tk = new TimeKeeping(v,b);
        FunctionList fl = new FunctionList(v,a,rdg,rng,s,t,tk);
        SelectFunction sf = new SelectFunction(v,fl);

        sf.reqSelectFunction();
        assertEquals(1, sf.setting);
    }

    @Test
    public void setFunction() {
        View v = new View();
        AlarmList al = new AlarmList(v);
        Buzzer b = new Buzzer(v,al);
        Alarm a = new Alarm(v,al,b);
        RandomDirectionGenerator rdg = new RandomDirectionGenerator(v);
        RandomNumberGenerator rng = new RandomNumberGenerator(v);
        Stopwatch s = new Stopwatch(v);
        Timer t = new Timer(v,b);
        TimeKeeping tk = new TimeKeeping(v,b);
        FunctionList fl = new FunctionList(v,a,rdg,rng,s,t,tk);
        SelectFunction sf = new SelectFunction(v,fl);

        sf.part = 3;
        sf.setFunction(1);
        assertEquals(0, sf.showArray[1]);
    }
}
```

11. FunctionListTest

FunctionListTest.java

```java
import org.junit.Test;

import static org.junit.Assert.*;

public class FunctionListTest {

    @Test
    public void getNextFunction() {
        View v = new View();
        AlarmList al = new AlarmList(v);
        Buzzer b = new Buzzer(v,al);
        Alarm a = new Alarm(v,al,b);
        RandomDirectionGenerator rdg = new RandomDirectionGenerator(v);
        RandomNumberGenerator rng = new RandomNumberGenerator(v);
        Stopwatch s = new Stopwatch(v);
        Timer t = new Timer(v,b);
        TimeKeeping tk = new TimeKeeping(v,b);
        FunctionList fl = new FunctionList(v,a,rdg,rng,s,t,tk);

        assertEquals(3, fl.getNextFunction(2));
    }
}
```

2061. Unit Testing

TimeKeepingTest: 2 total, 2 passed — 612 ms

Collapse | Expand

TimeKeepingTest		612 ms
southFinder	passed	225 ms
setTimePart	passed	387 ms

TimeDataTest: 1 total, 1 passed — 219 ms

Collapse | Expand

TimeDataTest		219 ms
saveTimePart	passed	219 ms

TimerTest: 1 total, 1 passed — 226 ms

Collapse | Expand

TimerTest		226 ms
setTimerPart	passed	226 ms

BuzzerTest: 2 total, 2 passed — 296 ms

Collapse | Expand

BuzzerTest		296 ms
beep	passed	232 ms
reqStop	passed	64 ms

StopwatchTest: 4 total, 4 passed — 290 ms

Collapse | Expand

StopwatchTest		290 ms
reqBackToStopwatch	passed	208 ms
reqPause	passed	51 ms
reqResume	passed	15 ms
resetStopwatch	passed	16 ms

LapSplitListTest: 1 total, 1 passed — 5 ms

Collapse | Expand

LapSplitListTest		5 ms
storeLapSplit	passed	5 ms

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

RandomNumberGeneratorTest: 4 total, 4 passed 313 ms

Collapse | Expand

RandomNumberGeneratorTest		313 ms
setScopePart	passed	219 ms
saveScope	passed	20 ms
reqBackToSetScope	passed	50 ms
reqRandomNumber	passed	24 ms

RandomDirectionGeneratorTest: 2 total, 2 passed 267 ms

Collapse | Expand

RandomDirectionGeneratorTest		267 ms
reqRandomDirection	passed	209 ms
display_eLCD	passed	58 ms

SelectFunctionTest: 2 total, 2 passed 305 ms

Collapse | Expand

SelectFunctionTest		305 ms
setFunction	passed	229 ms
reqSelectFunction	passed	76 ms

FunctionListTest: 1 total, 1 passed 219 ms

Collapse | Expand

FunctionListTest		219 ms
getNextFunction	passed	219 ms

2062. System Testing

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.
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 블락되는지 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 저장되어있던 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 삭제 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	스누즈 시험 (Mode)	- 알람 버저가 울릴 때 Mode 버튼을 누르면 5분 뒤 Snooze가 울리는지 각각 Test	14. Set Snooze	R4.4
14-2	스누즈 시험 (Funct)	- 알람 버저가 울릴 때 Funct 버튼을 누르면 5분 뒤 Snooze가 울리는지 각각 Test	14. Set Snooze	R4.4

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

2063. Testing Traceability Analysis