



# 골라보시게

## OOPT STAGE 2050 & 2060

### Team #1

201411273 박재범  
201411275 박진호  
201411311 장원영  
201311313 정인원

# Contents

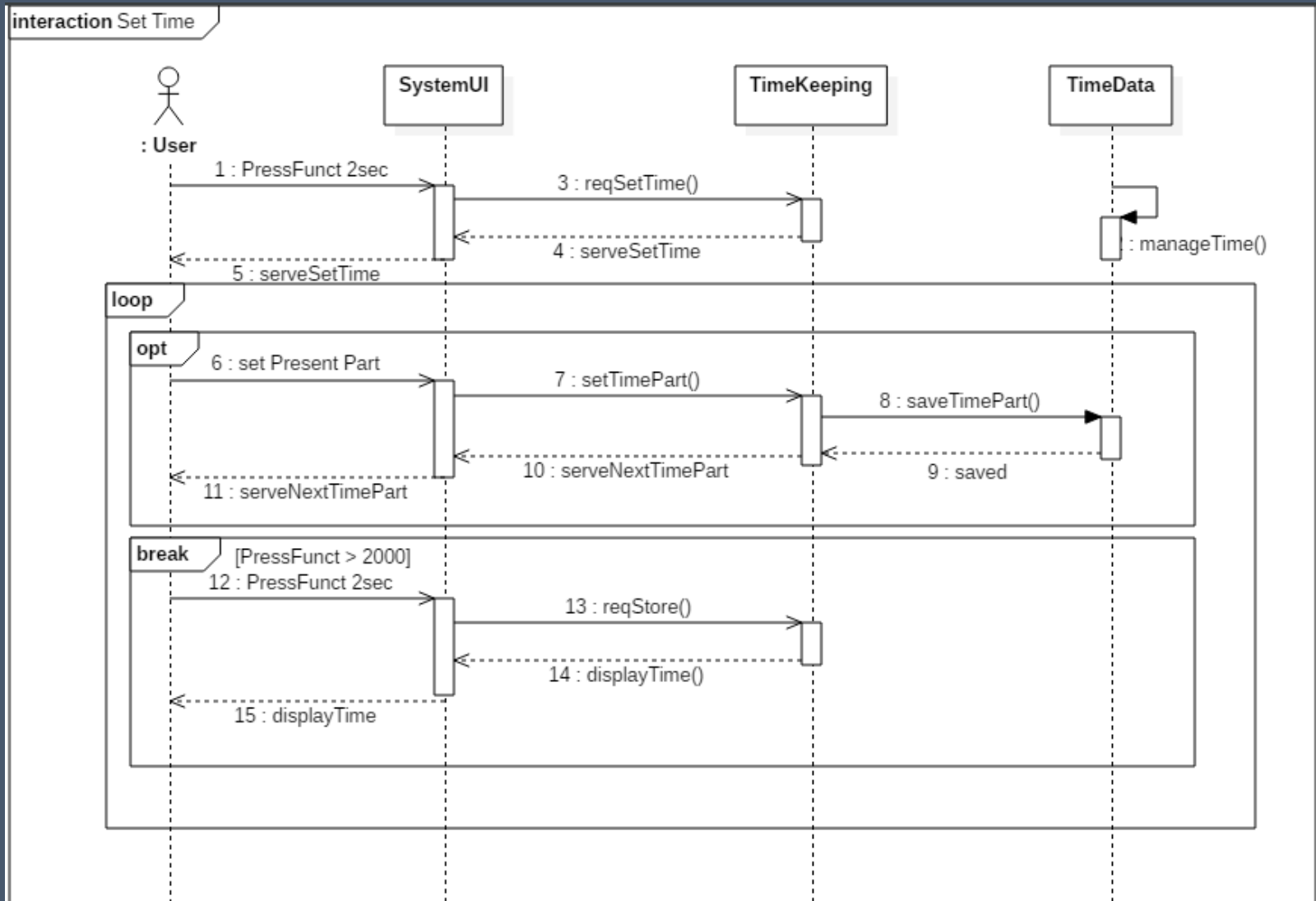
- 01 Implements Windows
- 02 Write Unit Test Code
- 03 Unit Testing
- 04 System Testing
- 05 Testing Traceability Analysis

**1**

**Implements Windows**

# 1. Implements Windows

## 1. Set Time



# 1. Implements Windows

## 1. Set Time

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

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

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

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

# 1. Implements Windows

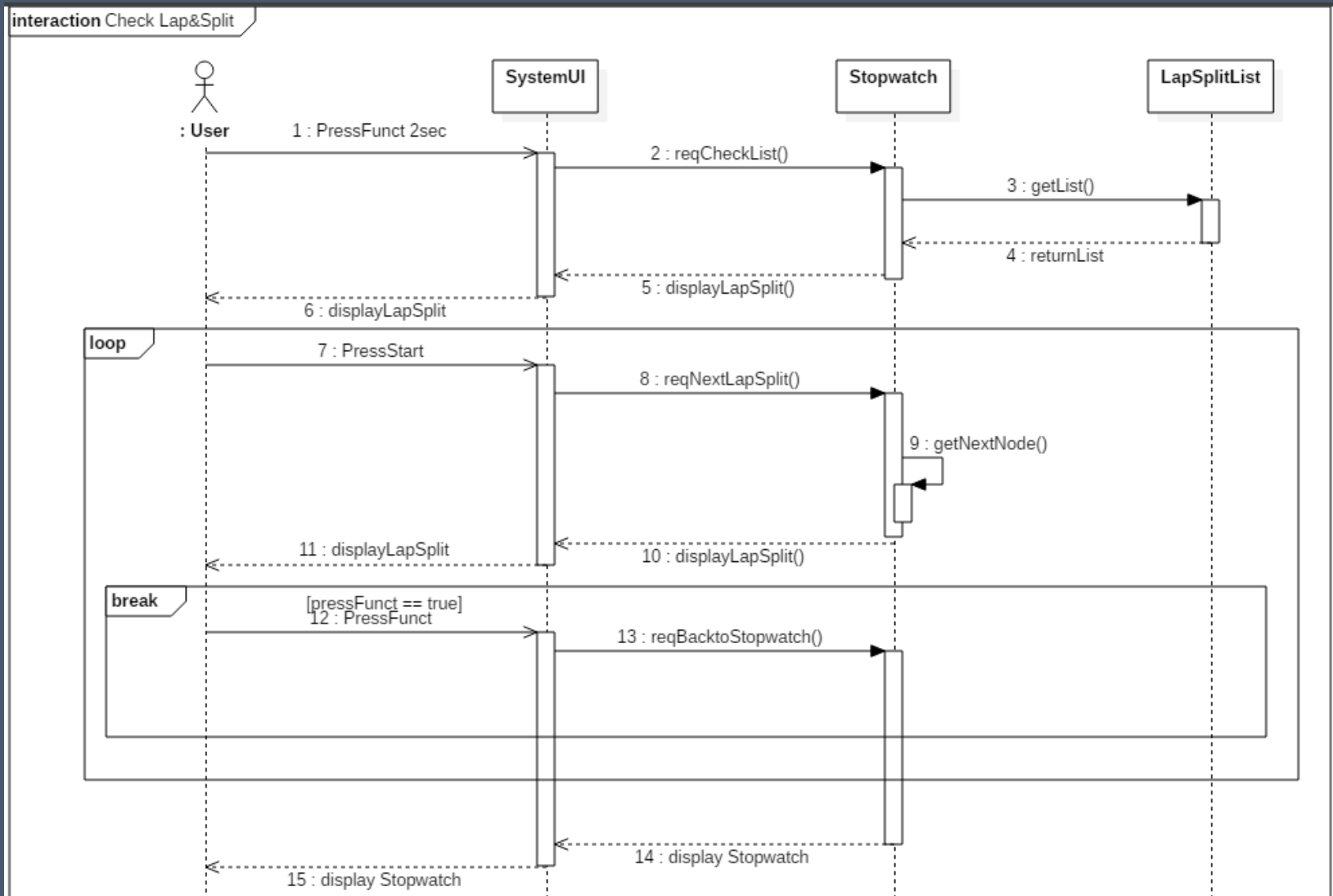
## 1. Set Time

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

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

# 1. Implements Windows

## 7. Check Lap & Split



# 1. Implements Windows

## 7. Check Lap & Split

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

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

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

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



# 1. Implements Windows

## 7. Check Lap & Split

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

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

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

**2**

**Write Unit Test Code**

## 2. Write Unit Test Code

### 1. TimeKeepingTest

```
TimeKeepingTest.java x
1  import org.junit.Test;
2
3  import static org.junit.Assert.*;
4
5  public class TimeKeepingTest {
6
7      @Test
8      public void setTimePart() {
9          View v = new View();
10         TimeKeeping tk = new TimeKeeping(v);
11         tk.part = 6; // hour10
12         tk.time.format= 2; // time format 24h
13         tk.time.hour = 5;
14         tk.saveValue = 2;
15         tk.setTimePart(0);
16
17         assertEquals( expected: 0, tk.saveValue);
18     }
19
20     @Test
21     public void southFinder() {
22         View v = new View();
23         TimeKeeping tk = new TimeKeeping(v);
24         tk.time.format =0;
25         tk.southFinder( min: "0", min10: "3", hour: "2", hour10: "1");
26
27         assertEquals( expected: 1,tk.eLCD[1]);
28         assertEquals( expected: 1,tk.eLCD[2]);
29     }
30 }
```

# 2. Write Unit Test Code

## 2. TimerTest

```
TimerTest.java x
1  import org.junit.Test;
2
3  import static org.junit.Assert.*;
4
5  public class TimerTest {
6
7      @Test
8      public void setTimerPart() {
9          View v = new View();
10         Buzzer b = new Buzzer(v);
11         Timer tm = new Timer(v,b);
12
13         tm.part = 1; // sec
14         tm.saveValue = 5;
15
16         tm.setTimerPart(0);
17
18         assertEquals(tm.timer.sec , tm.saveValue);
19     }
20 }
21
```

## 2. Write Unit Test Code

### 3. LapSplitListTest

```
LapSplitListTest.java x
1  import org.junit.Test;
2
3  import java.util.Vector;
4
5  import static org.junit.Assert.*;
6
7  public class LapSplitListTest {
8
9      @Test
10     public void storeLapSplit() {
11
12         LapSplitList ls = new LapSplitList();
13
14         Vector v = new Vector();
15         v.add(1);
16         v.add(2);
17         v.add(3);
18         ls.storeLapSplit(v);
19
20         assertEquals("expected: 1, ls.lns.elementAt( index: 0));", 1, ls.lns.elementAt( index: 0));
21         assertEquals("expected: 2, ls.lns.elementAt( index: 1));", 2, ls.lns.elementAt( index: 1));
22         assertEquals("expected: 3, ls.lns.elementAt( index: 2)); // LAP", 3, ls.lns.elementAt( index: 2));
23         assertEquals("expected: 1, ls.lns.elementAt( index: 3));", 1, ls.lns.elementAt( index: 3));
24         assertEquals("expected: 2, ls.lns.elementAt( index: 4));", 2, ls.lns.elementAt( index: 4));
25         assertEquals("expected: 3, ls.lns.elementAt( index: 5)); // SPLIT", 3, ls.lns.elementAt( index: 5));
26     }
27 }
```

## 2. Write Unit Test Code

### 4. RandomNumberGeneratorTest

```
RandomNumberGeneratorTest.java x ✓
1  import org.junit.Test;
2
3  import static org.junit.Assert.*;
4
5  public class RandomNumberGeneratorTest {
6
7      @Test
8      public void reqRandomNumber() {
9
10         View v = new View();
11         RandomNumberGenerator rng = new RandomNumberGenerator(v);
12         int result = rng.reqRandomNumber( scope: 100);
13         int flag;
14
15         if(result>=0 && result<100)
16             flag = 1;
17         else
18             flag =0;
19
20         assertEquals(flag, actual: 1);
21     }
22 }
```

## 2. Write Unit Test Code

### 5. RandomDirectionGeneratorTest

```
RandomDirectionGeneratorTest.java x
1  import org.junit.Test;
2
3  import java.util.Random;
4
5  import static org.junit.Assert.*;
6
7  public class RandomDirectionGeneratorTest {
8
9      @Test
10     public void reqRandomDirection() {
11         View v = new View();
12         RandomDirectionGenerator rdg = new RandomDirectionGenerator(v);
13         rdg.reqRandomDirection();
14
15         int flag = 0;
16         if(rdg.result >= 0 && rdg.result <=60)
17             flag = 1;
18
19         assertEquals("expected: 1, flag);
20
21     }
22 }
23 }
```

**3**

## **Unit Testing**



# 3. Unit Testing

<b>TimeKeepingTest: 2 total, 2 passed</b>		612 ms
		<a href="#">Collapse</a>   <a href="#">Expand</a>
 TimeKeepingTest		612 ms
 southFinder	passed	225 ms
 setTimePart	passed	387 ms

<b>TimerTest: 1 total, 1 passed</b>		226 ms
		<a href="#">Collapse</a>   <a href="#">Expand</a>
 TimerTest		226 ms
 setTimerPart	passed	226 ms

<b>LapSplitListTest: 1 total, 1 passed</b>		5 ms
		<a href="#">Collapse</a>   <a href="#">Expand</a>
 LapSplitListTest		5 ms
 storeLapSplit	passed	5 ms

# 3. Unit Testing

**RandomNumberGeneratorTest: 1 total, 1 passed** 221 ms

[Collapse](#) | [Expand](#)

RandomNumberGeneratorTest

221 ms

reqRandomNumber

passed

221 ms

**RandomDirectionGeneratorTest: 1 total, 1 passed** 206 ms

[Collapse](#) | [Expand](#)

RandomDirectionGeneratorTest

206 ms

reqRandomDirection

passed

206 ms

**4**

## **System Testing**

# 4. 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 0이 되면 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 불량되는지 Test	9. Check Lap & Split	R3.4

# 4. System Testing

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 불량되는지 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

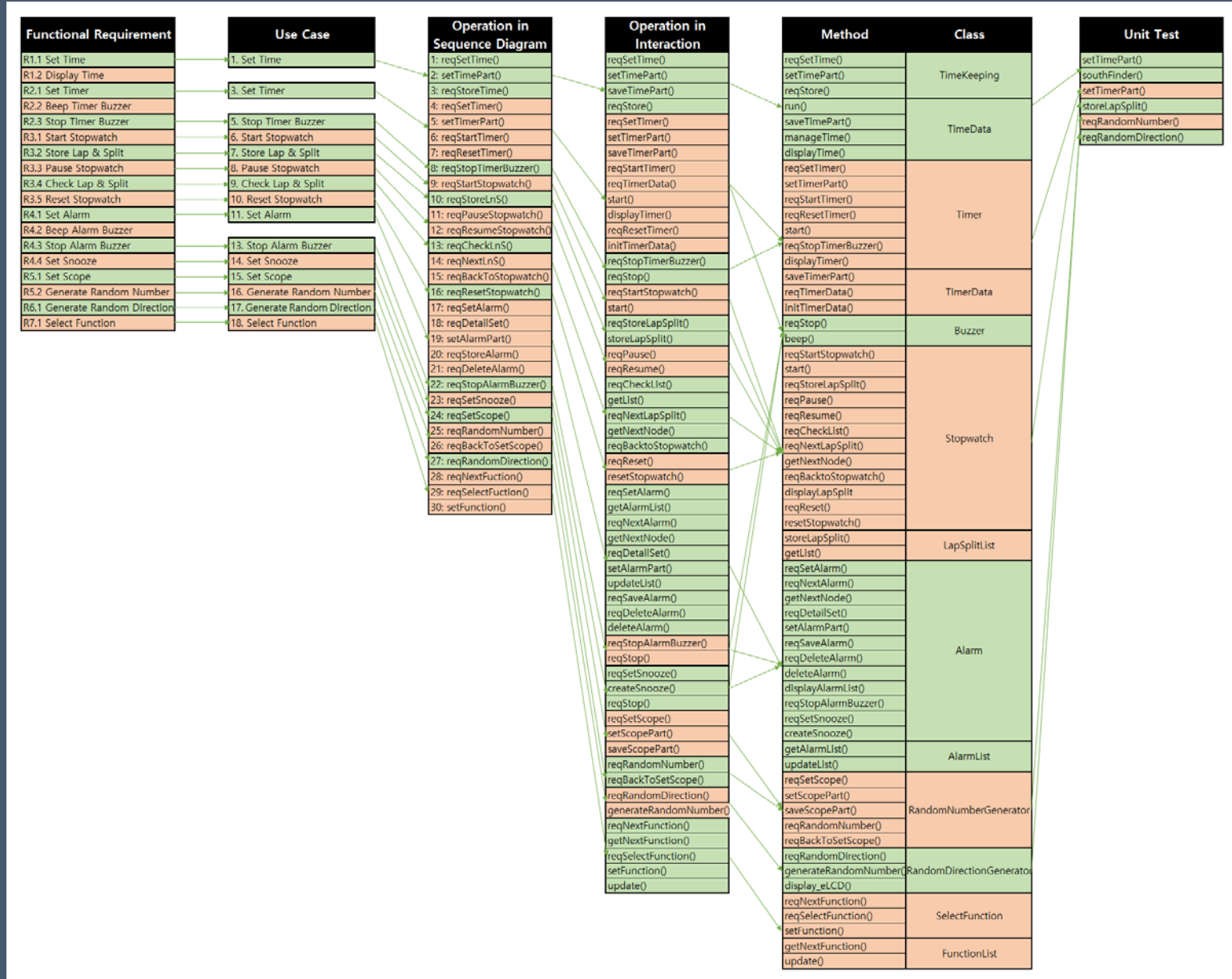
# 4. System Testing

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 On하면 자동으로 설정을 저장하고 TimeKeeping으로 전환되는지 Test	18. Select Function	R7.1

**5**

## **Testing Traceability Analysis**

# 5. Testing Traceability Analysis





— Thank You —

Q&A