

# OOPT 2050: Implement

Team6: Adventure Company  
Adventure Digital Watch

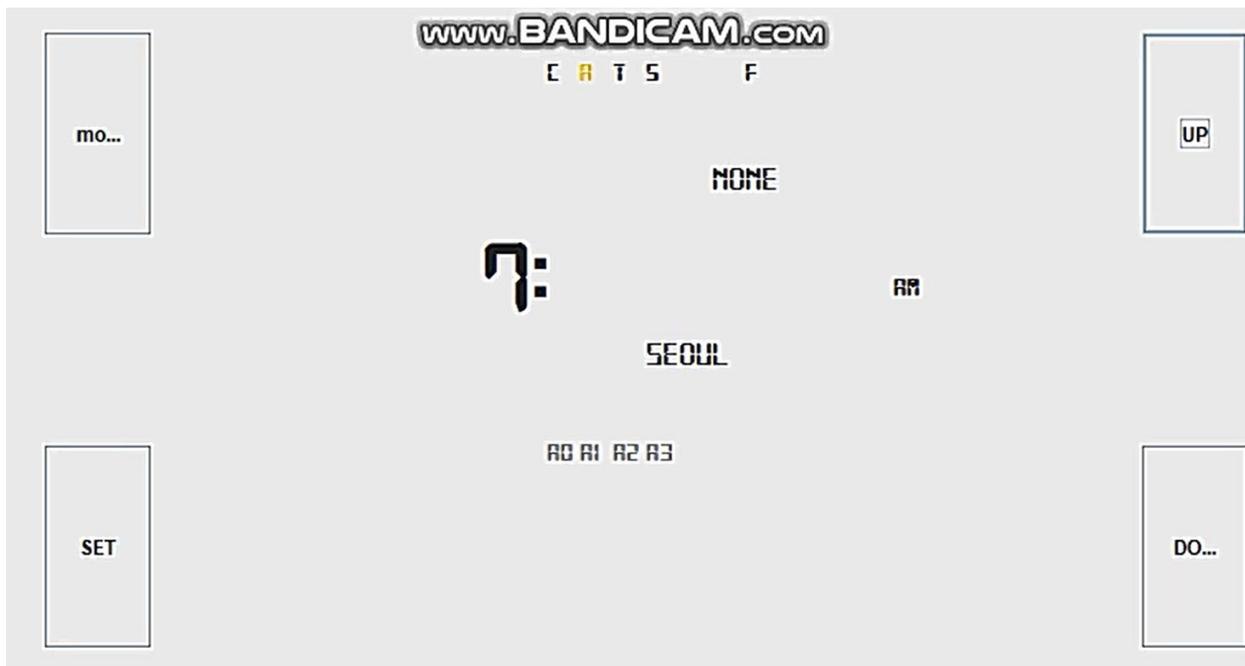
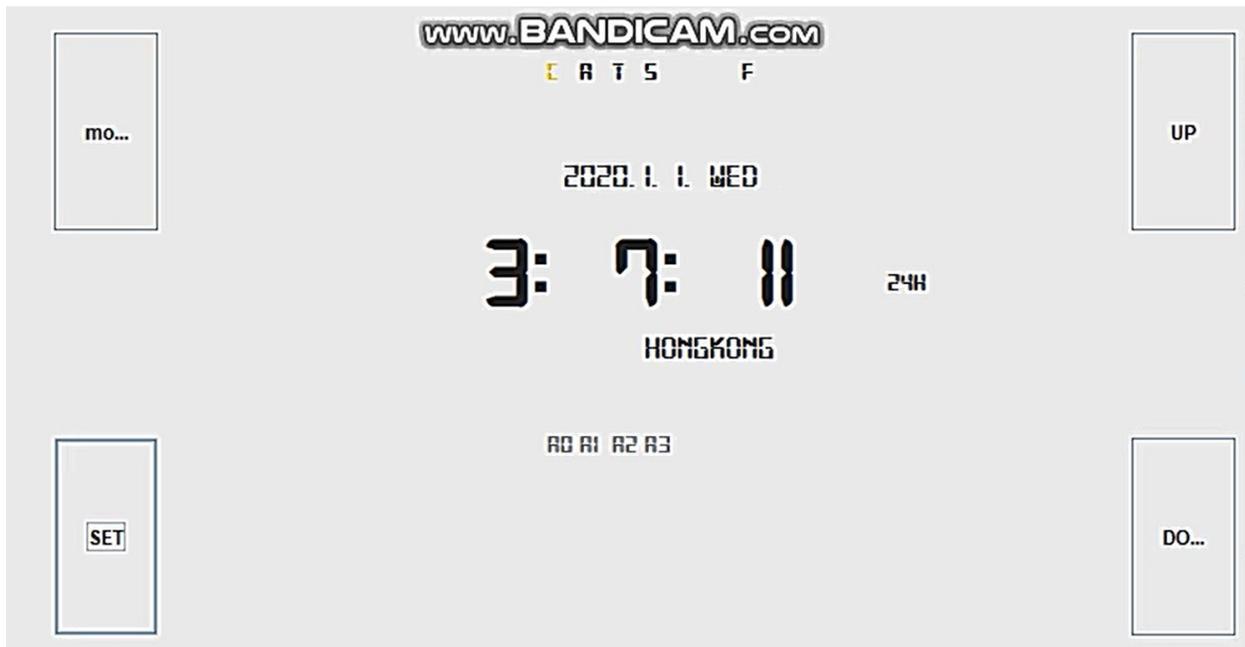
팀장 정주원  
팀원 강현우 송승현 이동현

# CONTENTS

- 0. Demo..... 4
- 1. Activity 2152. Implement Windows..... 6
  - 1) Clock..... 6
  - 2) Alarm..... 8
  - 3) Timer ..... 11
  - 4) Stopwatch ..... 14
  - 5) WorldTime ..... 18
  - 6) Geo..... 19
  - 7) Function Selector ..... 20
- 2. Activity 2155. Write Test Code ..... 22
  - 1) Clock..... 22
  - 2) Alarm..... 23
  - 3) Timer ..... 25
  - 4) Stopwatch ..... 28
  - 5) WorldClock..... 31
  - 6) Geo..... 32
  - 7) FunctionSelector ..... 33
- 3. Activity 2161. Unit Testing ..... 36
  - 1) Clock..... 41
  - 2) Alarm..... 41
  - 3) Timer ..... 41
  - 4) Stopwatch ..... 42
  - 5) WorldClock..... 42
  - 6) Geo..... 42
  - 7) FunctionSelector ..... 43
- 4. Activity 2163. System Testing ..... 44

5. Activity 2167. Testing Traceability Analysis ..... 49

## 0. Demo



www.BANDICAM.com

E R T S F

mo...

UP

0. 3. 44.

0: 3: 56

SET

DO...

R0 R1 R2 R3

www.BANDICAM.com

E R T S W G F

mo...

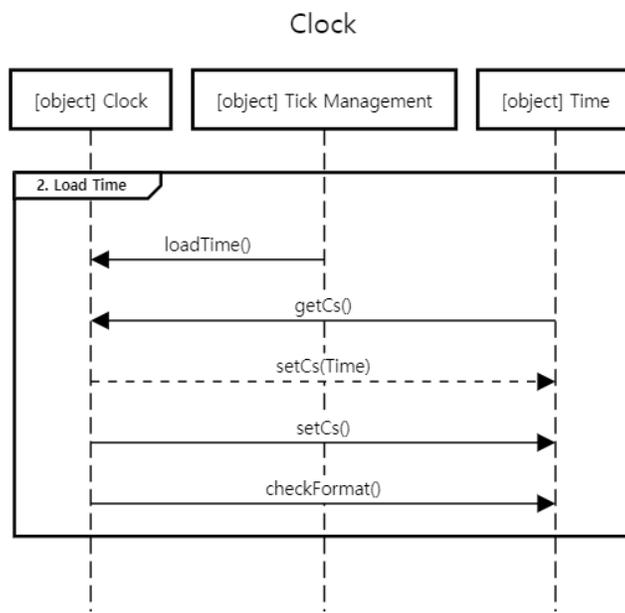
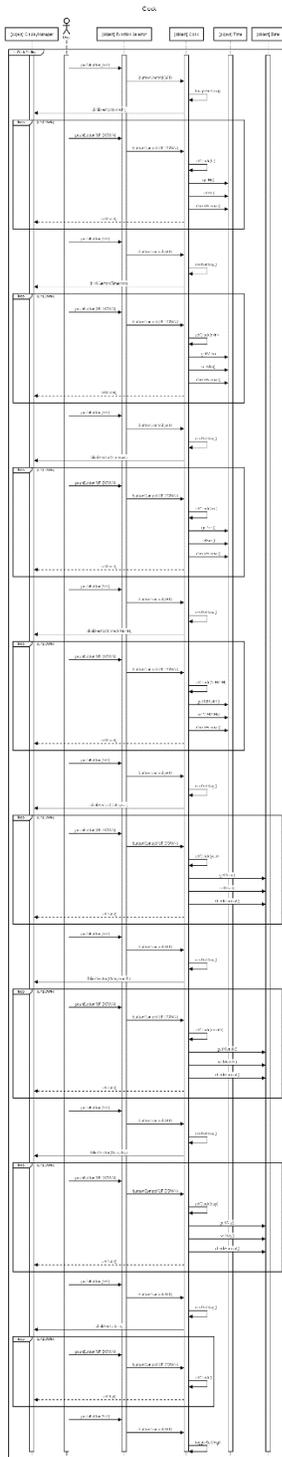
UP

SET

DO...

# 1. Activity 2152. Implement Windows

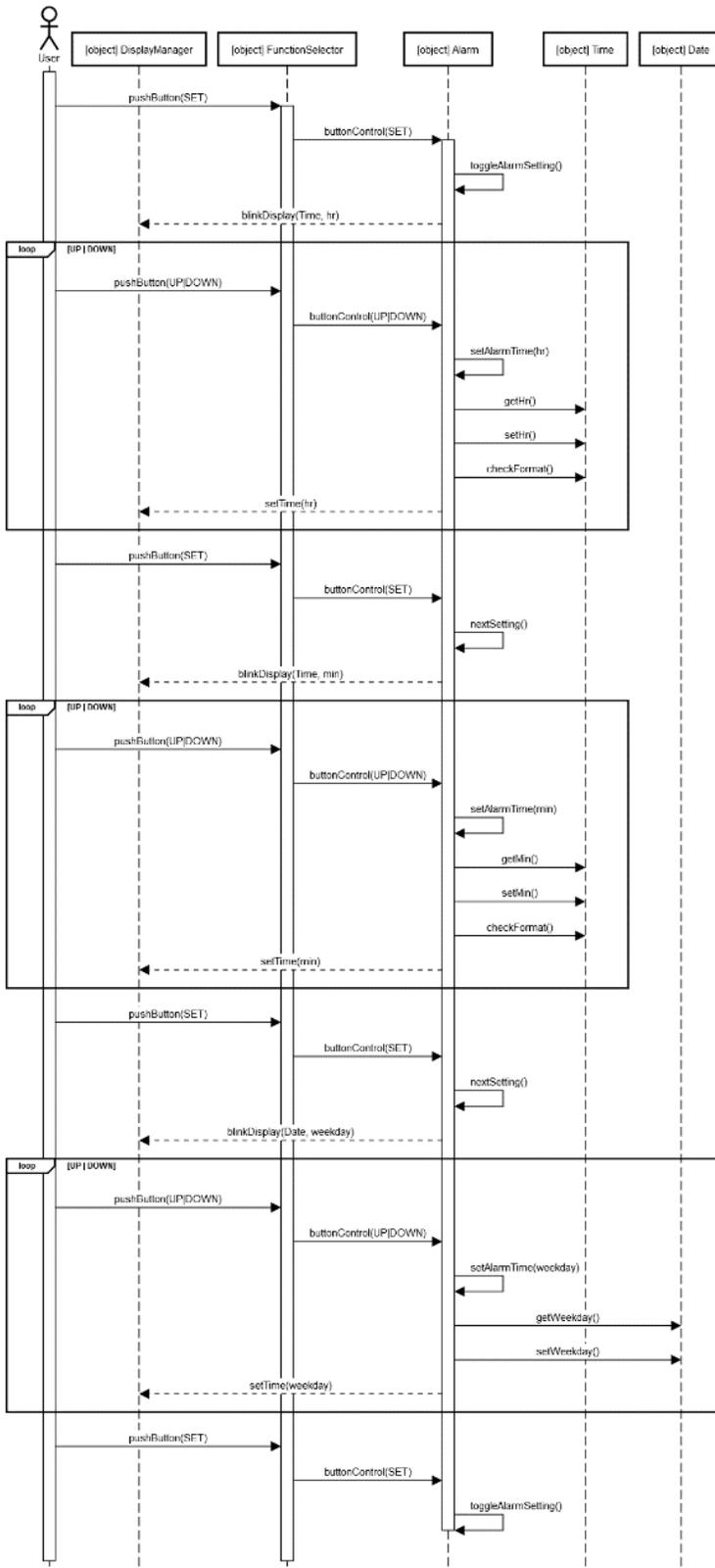
## 1) Clock



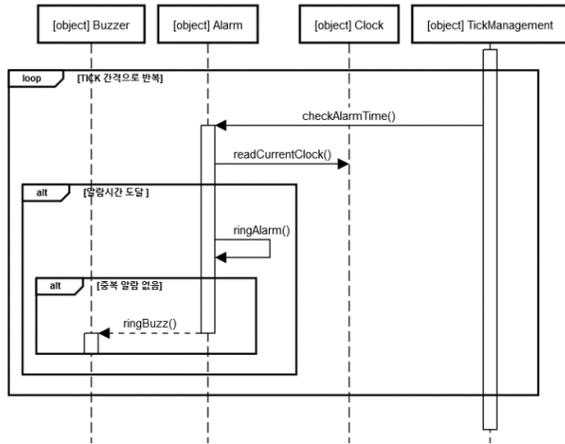


## 2) Alarm

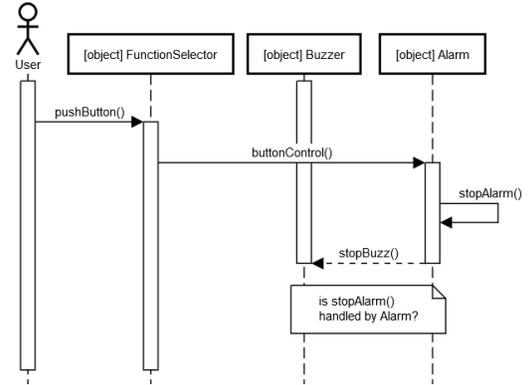
### 3. Set Alarm Time



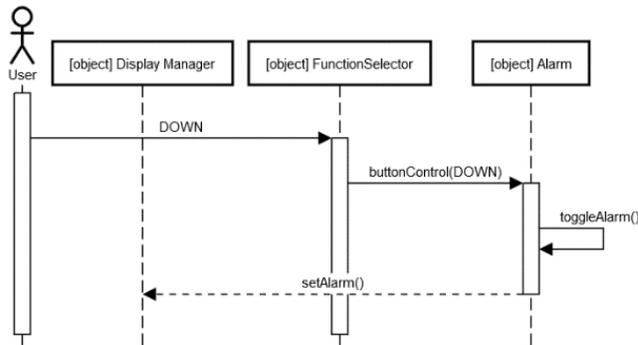
#### 4. Ring Alarm



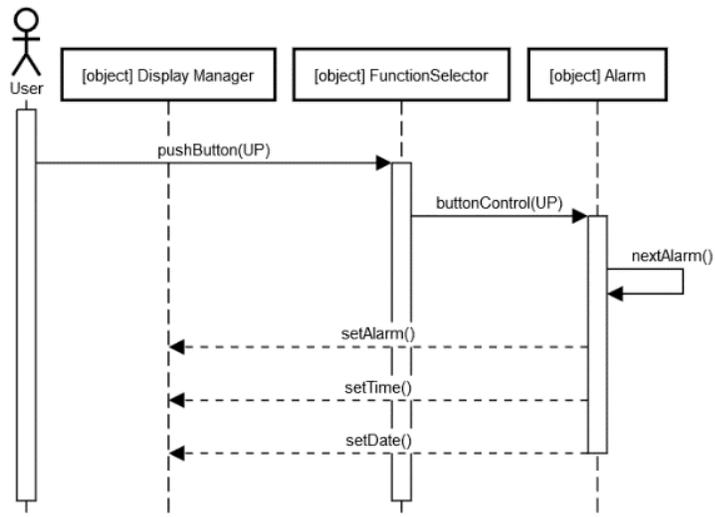
#### 5. Stop Alarm



#### 6. Toggle Alarm

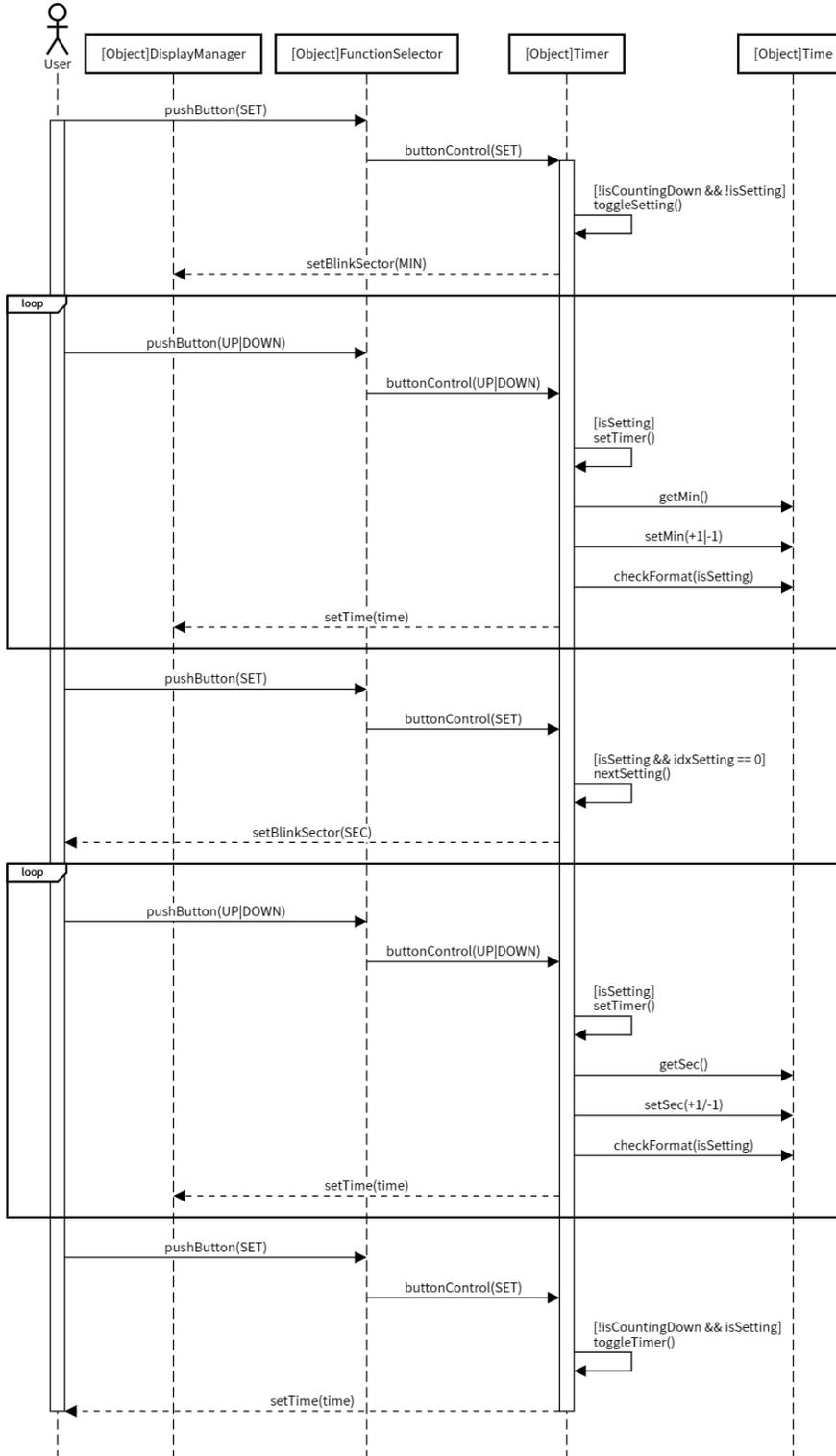


#### 7. Next Alarm

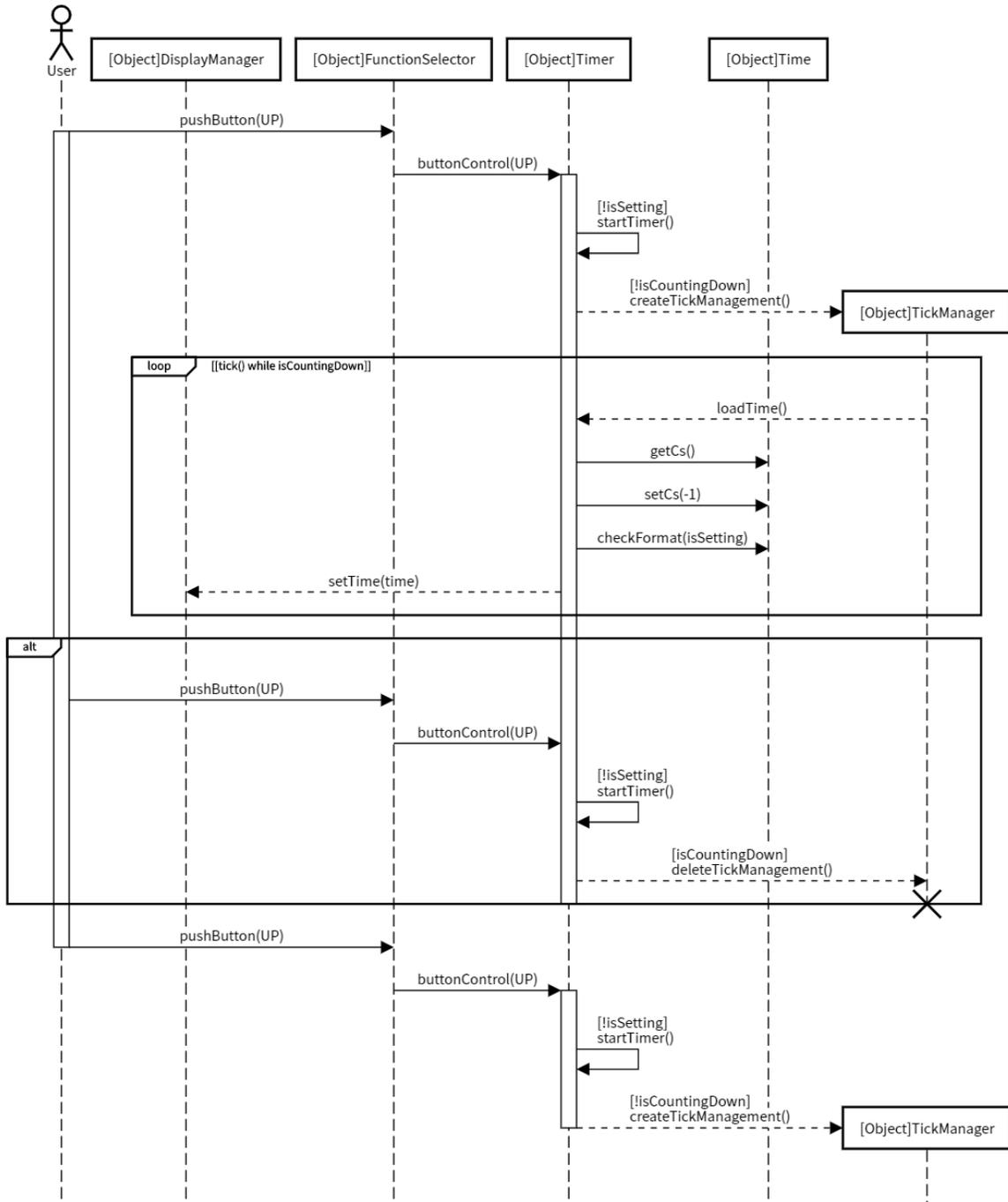


### 3) Timer

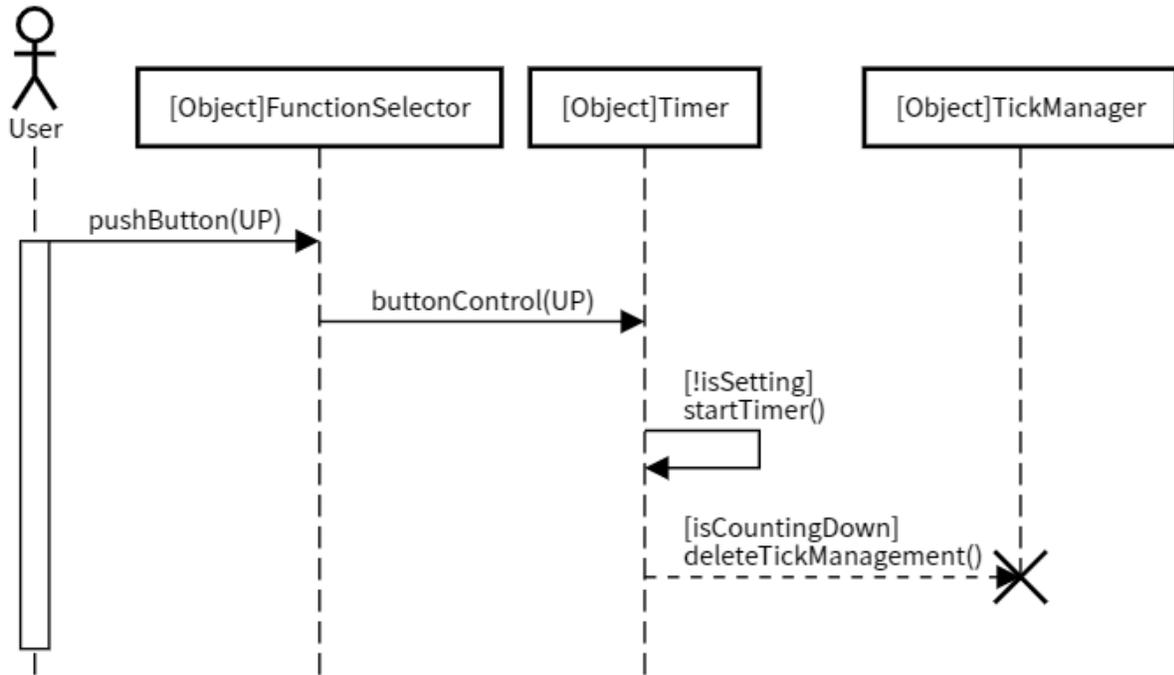
Timer: Set Timer



# Timer: Start/Restart Timer

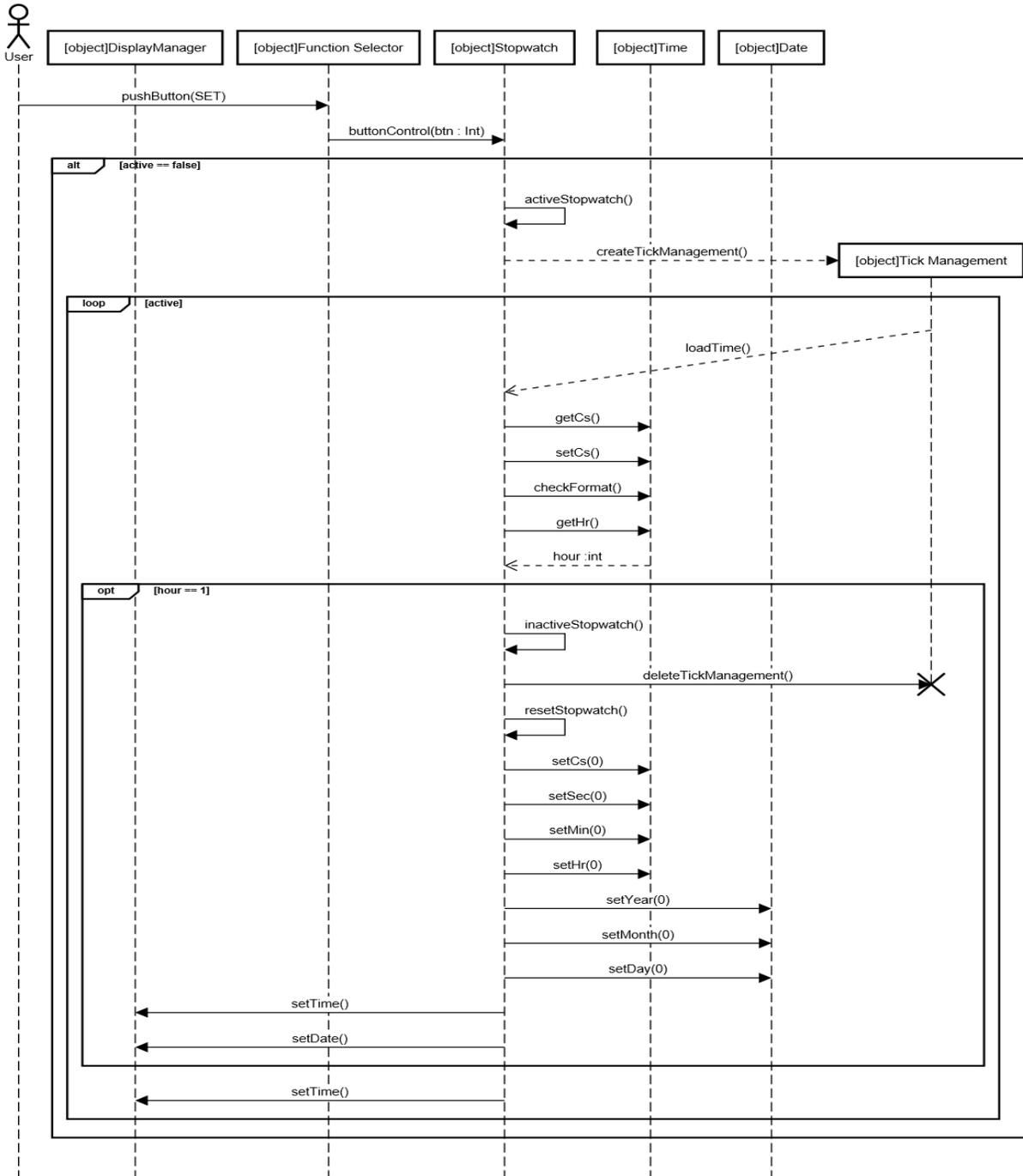


# Timer: Pause Timer

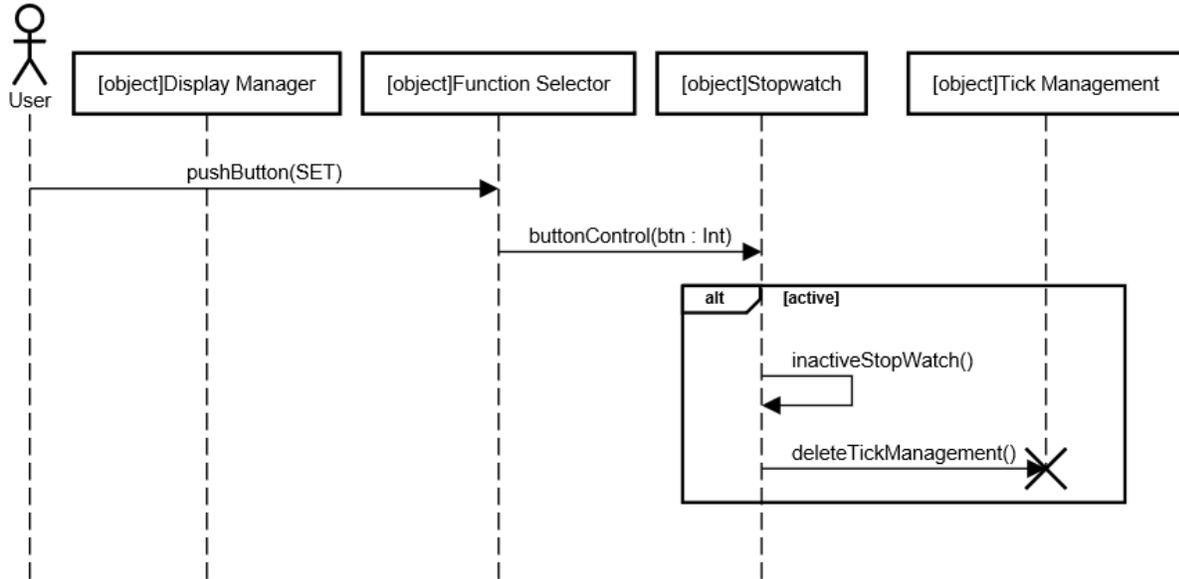


## 4) Stopwatch

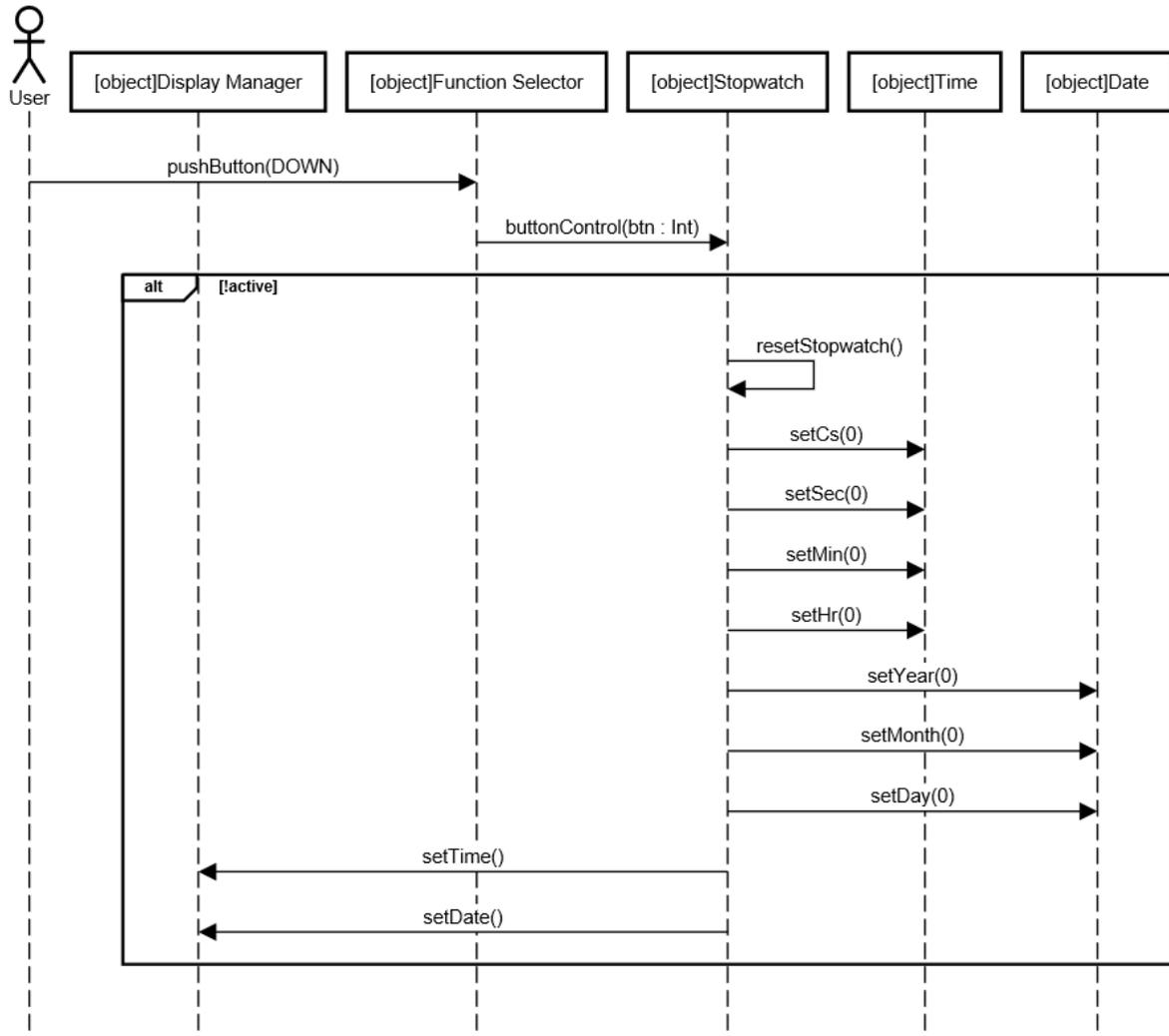
### Start Stopwatch



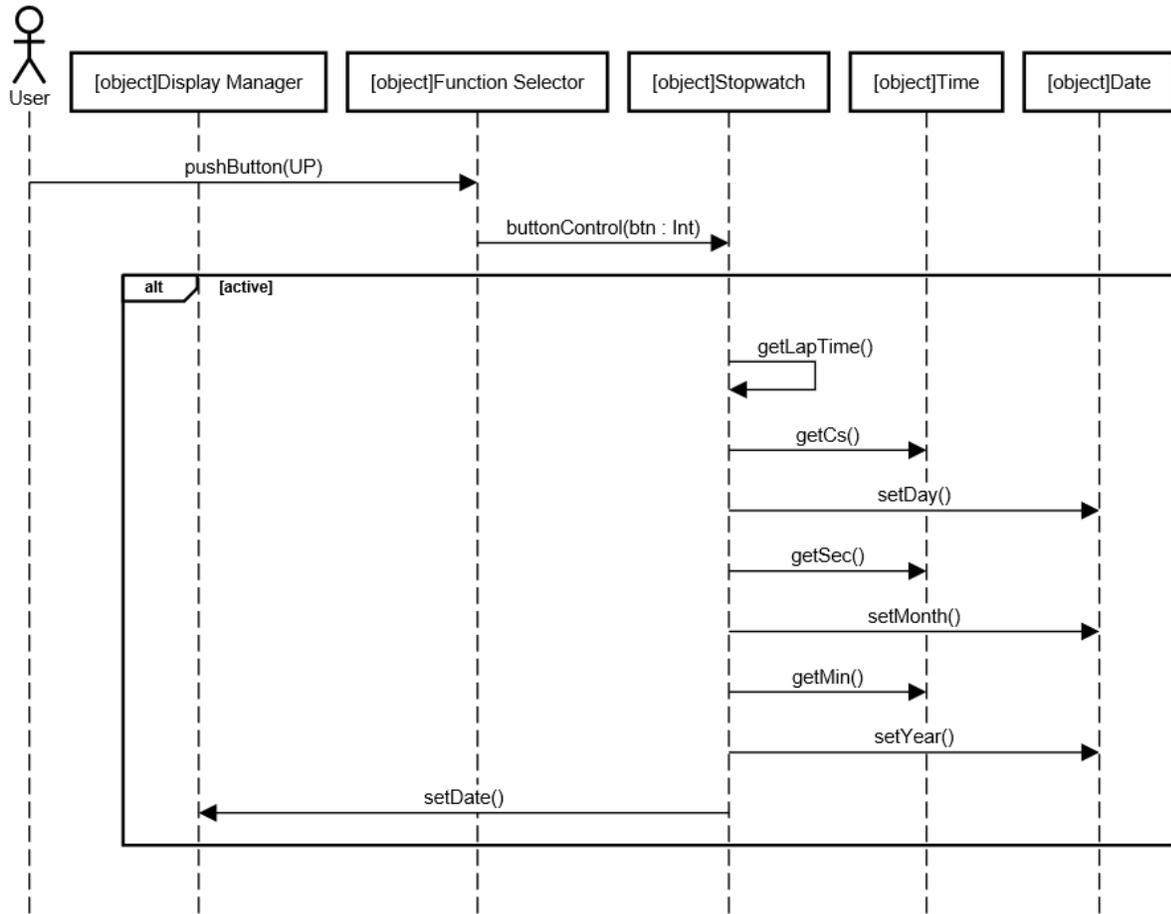
# Pause Stopwatch



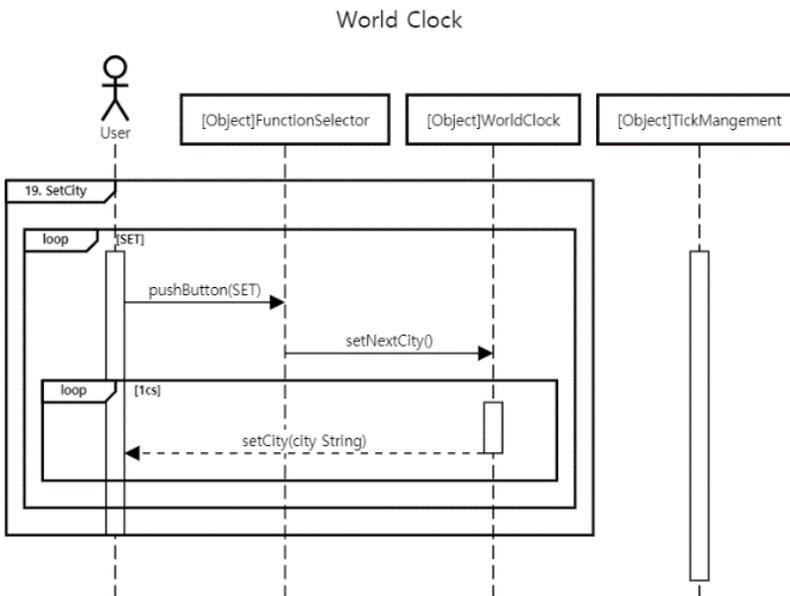
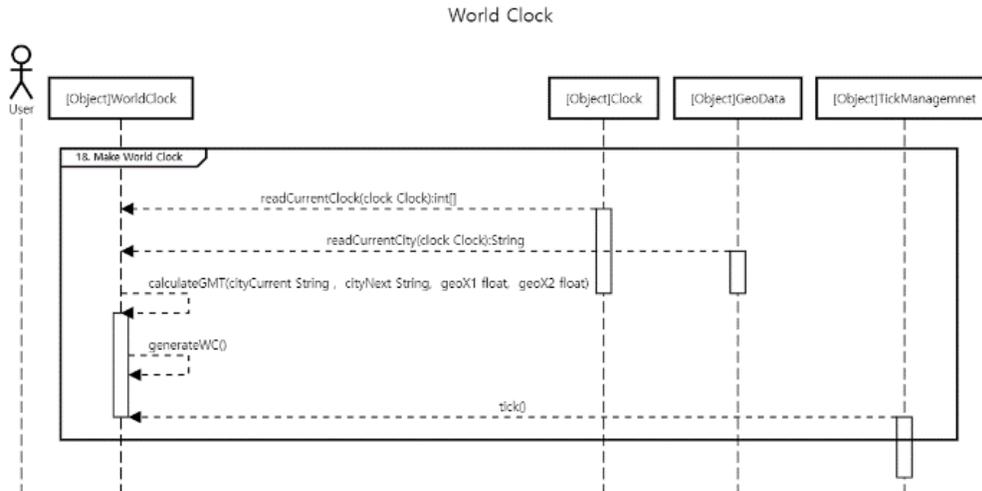
# Reset Stopwatch



# Set Lap Time

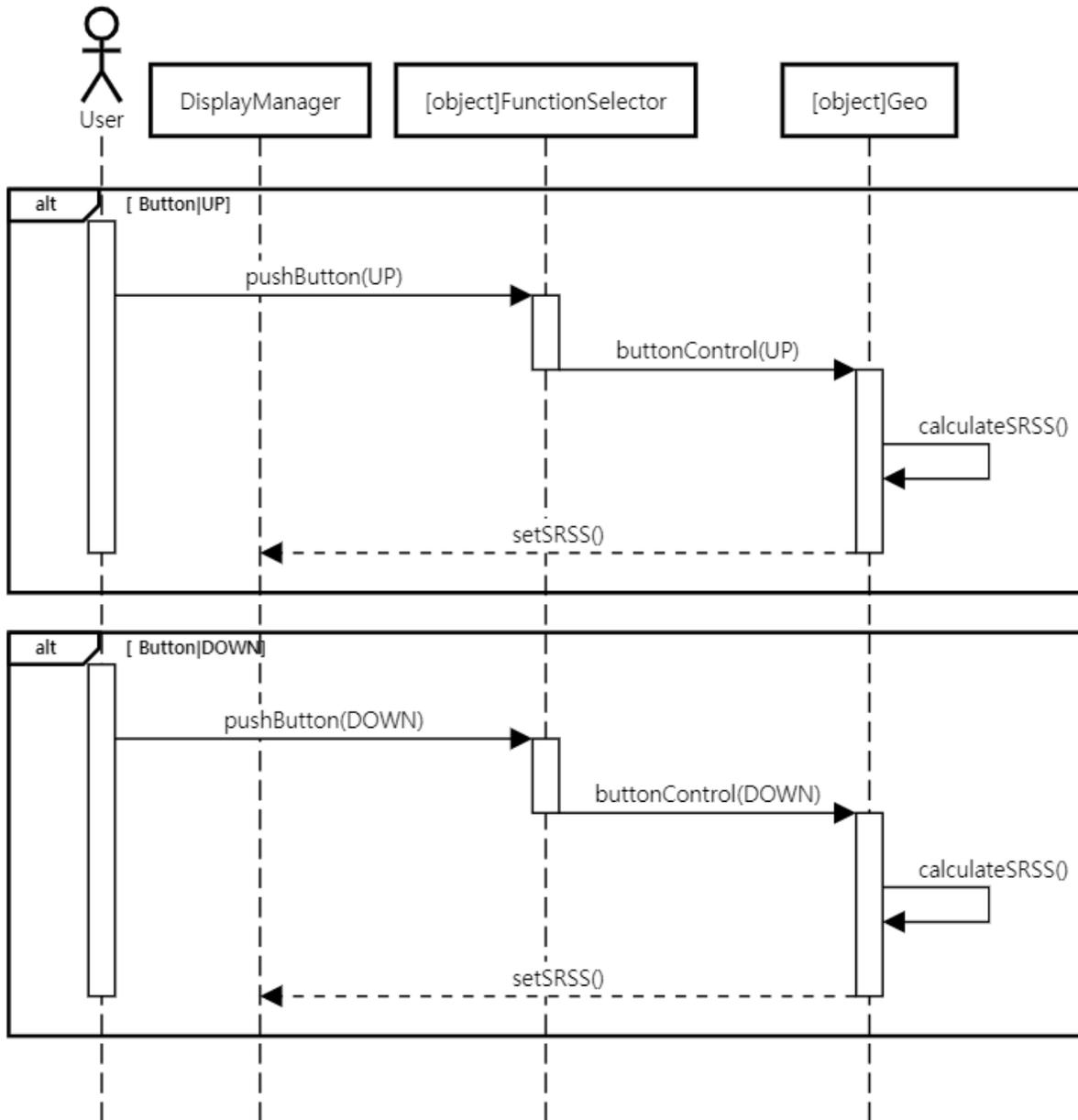


## 5) WorldTime



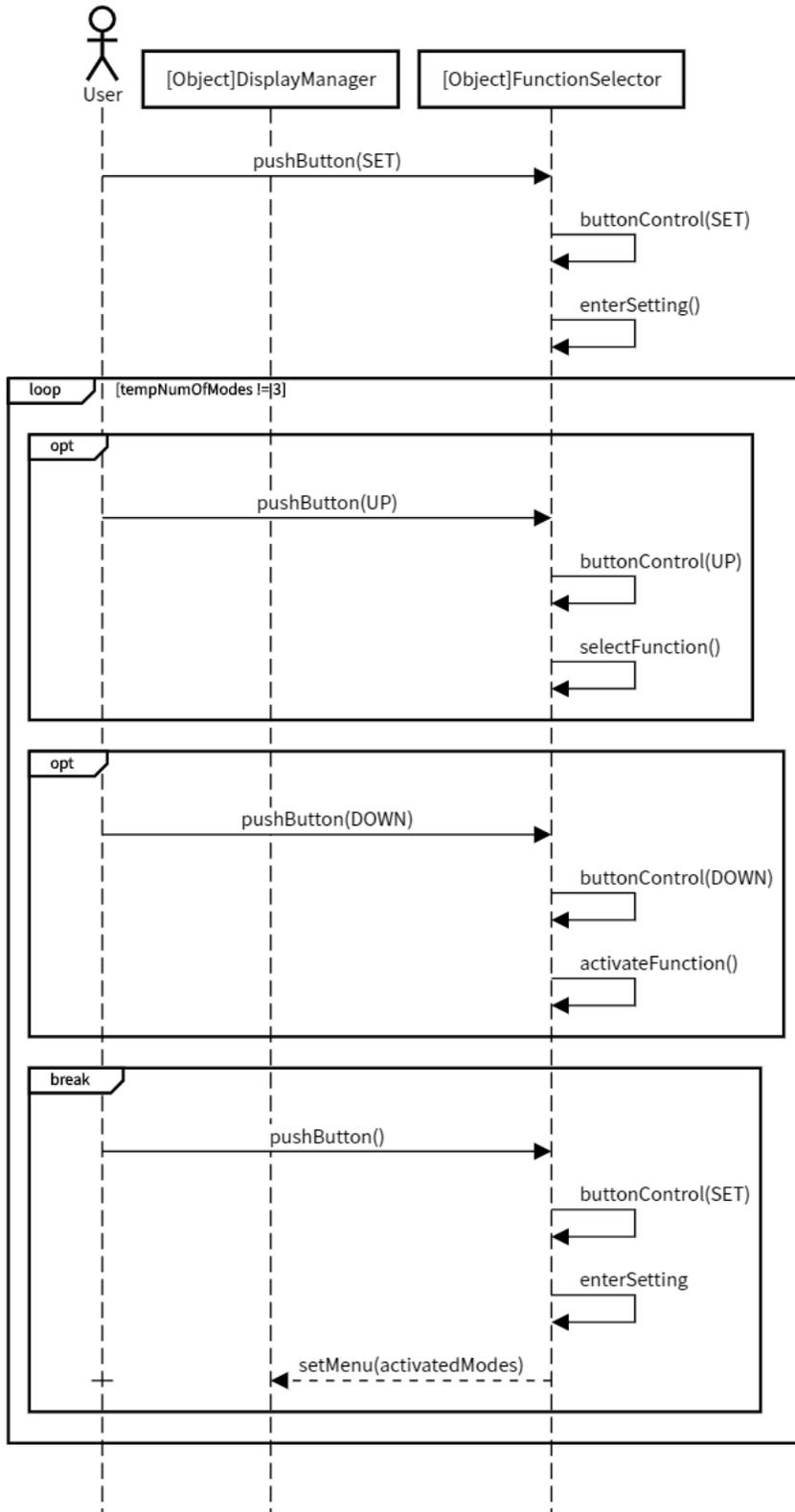
6) Geo

Geo\_setSRSS

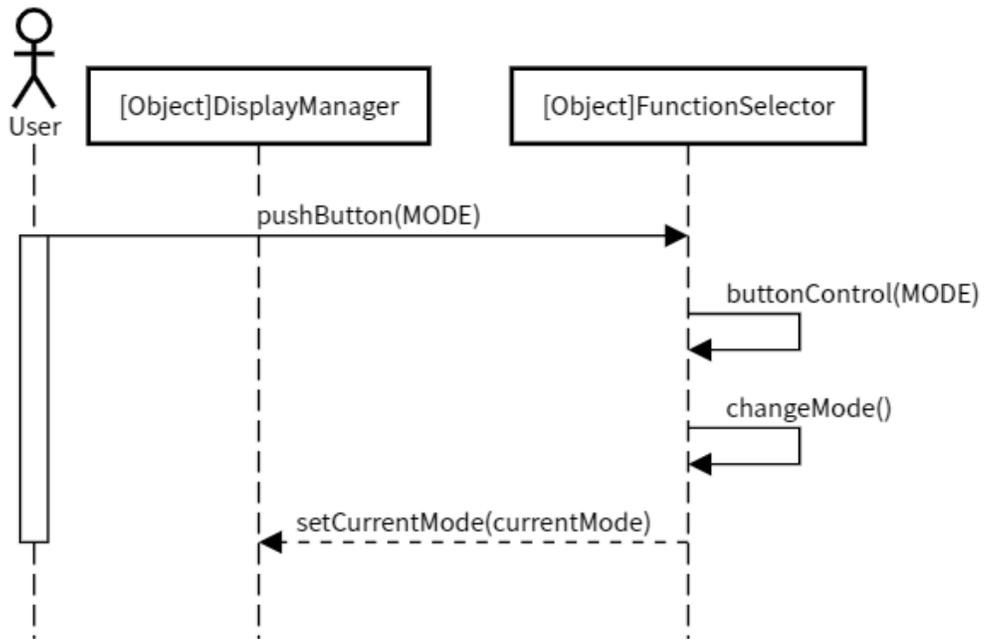


## 7) Function Selector

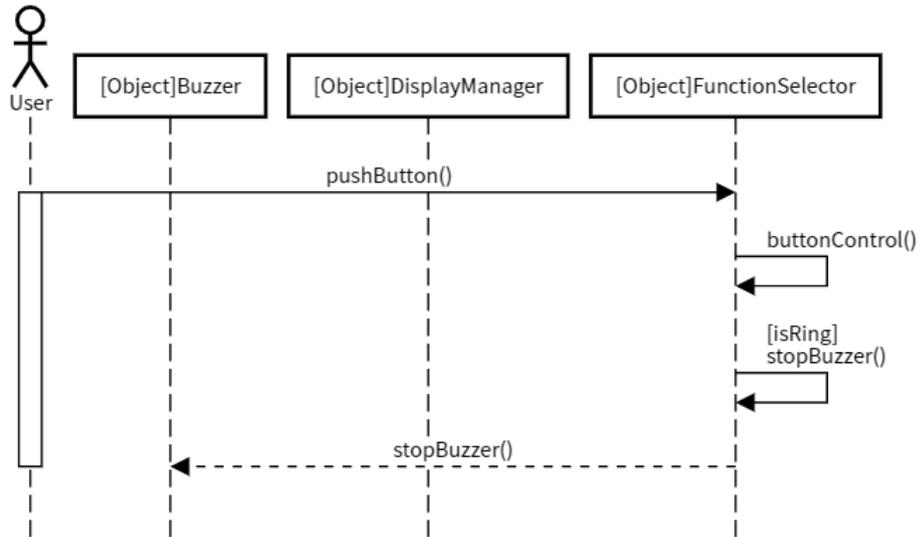
### Function Selector: Set Function



## Function Selector: Change Mode



## Function Selector: Stop Buzzer



## 2. Activity 2155. Write Test Code

### 1) Clock

```
@Test
public void calculateGMTTest() {
    int cityCurrent = 0;
    int cityNext = 1;
    assertEquals(exClock.calculateGMT(cityCurrent, cityNext), exClock.readCityGMT()[cityCurrent] -
exClock.readCityGMT()[cityNext]);
}

@Test
public void setClockTest() {
    exClock.buttonControl(Main.SET);
    assertEquals(exClock.buttonControl(Main.UP) + 1, exClock.buttonControl(Main.UP));
}

@Test
public void nextSettingTest() {
    assertEquals((exClock.buttonControl(Main.SET) + 1) % 8, exClock.buttonControl(Main.SET));
}

@Test
public void toggleSettingTest() {
    assertEquals(-1, exClock.buttonControl(Main.SET));
}

@Test
public void loadTimeTest(){
    assertEquals(exClock.readCurrentClock(),exClock.loadTime());
}
```

## 2) Alarm

```

@Test
public void toggleSettingTest() {
    assertEquals(-1, alarm.buttonControl(Main.SET));
}

@Test
public void nextSettingTest() {
    assertEquals((alarm.buttonControl(Main.SET)+1)%3, alarm.buttonControl(Main.SET));
}

@Test
public void setAlarmTimeTest() {
    alarm.buttonControl(Main.SET); // enter setting mode
    assertEquals(alarm.buttonControl(Main.UP)+1, alarm.buttonControl(Main.UP));
}

@Test
public void ringAlarmTest() {
    buzzerThread.start();

    assertFalse(fs.getBuzzer().isRing());
    alarm.buttonControl(Main.DOWN); // activate alarm (00:00, no repeat); current time = (2020.1.1 00:00)
    tickThread.start();
    alarm.checkAlarmTime();
    assertTrue(fs.getBuzzer().isRing());

    try {
        Thread.sleep(1000);
    } catch (Exception ignore){}

    fs.pushButton(Main.MODE);
    assertFalse(fs.getBuzzer().isRing());

    buzzerThread.interrupt();
    tickThread.interrupt();
}

@Test

```

### 3) Timer

```
@Test
public void testToggleSetting(){
    //enter setting
    timer.buttonControl(Main.SET);
    assertTrue(timer.isSetting());
    //exit setting
    timer.buttonControl(Main.SET);
    timer.buttonControl(Main.SET);
    assertFalse(timer.isSetting());
}
```

```
@Test
public void testNextSetting(){
    //min setting
    timer.buttonControl(Main.SET);
    assertEquals(0, timer.getIdxSetting());
    //sec setting
    timer.buttonControl(Main.SET);
    assertEquals(1, timer.getIdxSetting());
}
```

```
@Test
public void testSetTimer(){
    int temp;
    //increasing min
    timer.buttonControl(Main.SET);
    temp = timer.getTime().getMin();
    timer.buttonControl(Main.UP);
    assertEquals(temp + 1, timer.getTime().getMin());
    //decreasing min
    temp = timer.getTime().getMin();
    timer.buttonControl(Main.DOWN);
    assertEquals(temp - 1, timer.getTime().getMin());
}
```

```

@Test
public void testStartTimer() throws InterruptedException{
    //start timer
    timer.buttonControl(Main.SET);
    timer.buttonControl(Main.UP);
    timer.buttonControl(Main.SET);
    timer.buttonControl(Main.SET);
    assertEquals(1, timer.getTime().getMin());
    timer.buttonControl(Main.UP);
    Thread.sleep(5000);
    assertEquals(0, timer.getTime().getSec());
    assertEquals(1, timer.getTime().getMin());
    //pause timer
    int temp;
    timer.buttonControl(Main.UP);
    temp = timer.getTime().getTotalCs();
    assertEquals(temp, timer.getTime().getTotalCs());
}

```

```

@Test
public void testResetTimer(){
    timer.buttonControl(Main.SET);
    timer.buttonControl(Main.UP);
    timer.buttonControl(Main.SET);
    timer.buttonControl(Main.SET);
    timer.buttonControl(Main.DOWN);
    assertEquals(0, timer.getTime().getTotalCs());
}

```

```

@Test
public void testMyTurn() {
    //mode: clock at first
    fs.pushButton(Main.MODE); //mode: alarm
    fs.pushButton(Main.MODE); //mode: timer
    fs.pushButton(Main.SET); //enter timer min setting
    fs.pushButton(Main.SET); //enter timer sec setting
    System.out.println(timer.getTime().getMin() + ":" + timer.getTime().getSec());
    fs.pushButton(Main.UP); //set timer sec 1
    System.out.println(timer.getTime().getMin() + ":" + timer.getTime().getSec());

    fs.pushButton(Main.MODE); //mode: stopWatch
    fs.pushButton(Main.MODE); //mode: fs
    fs.pushButton(Main.MODE); //mode: clock
    fs.pushButton(Main.MODE); //mode: timer
    assertFalse(timer.isSetting());
}

```

```
@Test
public void testRingTimer() throws Exception {
    buzzerThread.start();
    assertFalse(fs.getBuzzer().isRing());
    timer.buttonControl(Main.SET);
    timer.buttonControl(Main.SET);
    timer.buttonControl(Main.UP);
    timer.buttonControl(Main.SET);
    timer.buttonControl(Main.UP);
    Thread.sleep(5000);
    assertTrue(fs.getBuzzer().isRing());
    buzzerThread.interrupt();
}
```

#### 4) Stopwatch

```
@Test
public void activeStopwatchTest() {
    assertTrue(stopwatch.activeStopwatch());
}

@Test
public void createTickmanagementTest() {
    assertNotNull(stopwatch.createTickmanagement());
}

@Test
public void loadTimeTest() throws InterruptedException {
    //check time running
    assertEquals(0, stopwatch.getTime().getCs());
    stopwatch.createTickmanagement();
    assertEquals(0, stopwatch.loadTime().getCs());
    //check time over
    stopwatch.setTime();
    Thread.sleep(200);
    assertEquals(0, stopwatch.getTime().getHr());
    assertEquals(0, stopwatch.getTime().getCs());
    assertEquals(0, stopwatch.getTime().getSec());
    assertEquals(0, stopwatch.getTime().getMin());
    assertTrue(stopwatch.activeStopwatch());
}
```

```
@Test
public void deleteTickManagementTest() {
    //TODO check whether time is counting
    stopwatch.createTickmanagement();
    assertNull(stopwatch.deleteTickManagement());
}
```

```
@Test
public void inactiveStopWatchTest() throws InterruptedException {
    stopwatch.activeStopwatch();
    Thread.sleep(200);
    assertEquals(0, stopwatch.getTime().getCs());
    assertFalse(stopwatch.inactiveStopWatch());
    assertEquals(3, stopwatch.getTime().getCs());
    assertNull(stopwatch.getTick());
}
```

```
@Test
public void getLapTimeTest() throws InterruptedException {
    //positive
    stopwatch.activeStopwatch();
}
```

## 5) WorldClock

```
@Test
public void setCityTest() {
    worldClock.buttonControl(Main.SET); // enter setting mode
    assertEquals((worldClock.buttonControl(Main.UP)+1)%3, worldClock.buttonControl(Main.UP));
}

@Test
public void toggleSettingTest() {
    assertNotEquals(worldClock.buttonControl(Main.SET), worldClock.buttonControl(Main.SET));
}

@Test
public void loadTimeTest() {
    assertEquals((worldClock.loadTime().getCs()+1)%100, worldClock.loadTime().getCs());
}
```

## 6) Geo

```
@Test
public void getSunriseTime() {
    Geo geoTest = new Geo(dm,clock);
    double testLST = geoTest.getSunriseTime(2020,1,1,37.34,126.5988,9,0);
    assertEquals(399.8331638950076,testLST);
}

@Test
public void getSunsetTime() {
    Geo geoTest = new Geo(dm,clock);
    double testLST = geoTest.getSunsetTime(2020,1,1,37.34,126.5988,9,0);
    assertEquals(979.6307011066772,testLST);
}

@Test
public void setSSSR(){
    Geo geoTest = new Geo(dm,clock);
    geoTest.buttonControl(UI.UP);
    assertEquals(true,geoTest.isSSSR());
    geoTest.buttonControl(UI.DOWN);
    assertEquals(false,geoTest.isSSSR());
}
```

## 7) FunctionSelector

```

@Test
public void testCreateFunction(){
    fs.pushButton(Main.MODE);
    fs.pushButton(Main.MODE);
    fs.pushButton(Main.MODE);
    fs.pushButton(Main.MODE);
    fs.pushButton(Main.SET);
    fs.pushButton(Main.UP);
    fs.pushButton(Main.DOWN);
    fs.pushButton(Main.UP);
    fs.pushButton(Main.UP);
    fs.pushButton(Main.DOWN);
    assertEquals(3, fs.getTempNumOfModes());
    fs.pushButton(Main.SET);
    assertFalse(fs.isSetting());
    assertNull(fs.getTimer());
    assertNotNull(fs.getWorldClock());
}

```

```

@Test
public void testSelectFunction(){
    fs.pushButton(Main.MODE);
    fs.pushButton(Main.MODE);
    fs.pushButton(Main.MODE);
    fs.pushButton(Main.MODE);
    fs.pushButton(Main.SET);
    int temp = fs.getIdxSetting();
    fs.pushButton(Main.UP);
    assertNotEquals(temp, fs.getIdxSetting());
}

```

```

@Test
public void testActivateFunction(){
    fs.pushButton(Main.MODE);
    fs.pushButton(Main.MODE);
    fs.pushButton(Main.MODE);
    fs.pushButton(Main.MODE);
    fs.pushButton(Main.SET);
    fs.pushButton(Main.DOWN);
    assertNotEquals(3, fs.getTempNumOfModes());
}

```

```

@Test
public void testStopButton(){

```

```
@Test
public void testChangeMode(){
    int temp = fs.getCurrentMode();
    fs.pushButton(Main.MODE);
    assertEquals(temp, fs.getCurrentMode());
    fs.pushButton(Main.MODE);
    fs.pushButton(Main.MODE);
    fs.pushButton(Main.MODE);
    fs.pushButton(Main.MODE);
    assertEquals(temp, fs.getCurrentMode());
}
```

```
@Test
public void testTimeout() throws Exception {
    tickThread.start();
    fs.pushButton(Main.MODE);
    assertEquals(1, fs.getCurrentMode());
    Thread.sleep(20000);
    assertEquals(0, fs.getCurrentMode());
    tickThread.interrupt();
}
```

### 3. Activity 2161. Unit Testing

Test Number	Test 항목	Description	Use Case	System Function	Pass
1-1	calculateGMTTest	현재도시와 다음도시와 GMT 시간차이를 계산한다.	1. Clock Setting	R 1.1	Pass
1-2	setClockTest	UP 버튼이 눌러졌을 때 잘 작동하는지 확인한다.	1. Clock Setting	R 1.1	Pass
1-3	nextSettingTest	세팅할 때 다음 세팅으로 잘 넘어가는지 확인한다.	1. Clock Setting	R 1.1	Pass
1-4	toggleSettingTest	세팅일 때 으로 넘어가는지 확인한다.	1. Clock Setting	R 1.1	Pass
2	loadTimeTest	지금 현재 시간이 잘 확인되는지 확인한다.	1. Clock Setting	R 1.2	Pass
3-1	toggleSettingTest	설정 모드로 진입되는지 확인한다.	3. Set Alarm Time	R 2.1	Pass
3-2	nextSettingTest	설정 모드에서 설정할 단위를 넘길 수 있는지 확인한다.	3. Set Alarm Time	R 2.1	Pass
3-3	setAlarmTimeTest	버튼 입력에 따 설정할 단위의 값을 1 씩 변경할 수 있는지 확인한다.	3. Set Alarm Time	R 2.1	Pass
4	ringAlarmTest	활성화된 알람과 현재 시간이 같을 때 buzzer 가 활성화되는지 확인하고 아무 버튼을 눌러	4. Ring Alarm	R 2.2	Pass

		buzzer 가 중지되는지 확인한다.			
5	toggleAlarmTest	알람을 켜고 끌 수 있는지 확인한다.	5. Toggle Alarm	R 2.3	Pass
6	nextAlarmTest	버튼 입력에 따라 현재 알람에서 다음 알람으로 제대로 넘어가는지 확인한다.	6. Next Alarm	R 2.4	Pass
7-1	testToggleSetting	설정모드로 진입하는지 확인한다.	7. Set Timer	R 3.1	Pass
7-2	testNextSetting	분 설정에서 초 설정으로 넘어가는지 확인한다.	7. Set Timer	R 3.1	Pass
7-3	testSetTimer	분과 초가 up/down 버튼에 따라 설정값이 증가/감소 하는지 확인한다.	7. Set Timer	R 3.1	Pass
8	testStartTimer	타이머 시간 설정후 thread 가 돌며 카운트다운되는지 확인한다.	8. Start/Restart Timer	R 3.2	Pass
9	testResetTimer	타이머 시간 설정후 초기화 시 시간이 0 으로 설정되는지 확인한다.	9. Reset Timer	R 3.3	Pass
10	testRingTimer	타이머의 카운트 다운이 끝난후 buzzer 의 알림이 울리는지 확인한다.	10. Ring Timer	R 3.4	Pass
11-1	activeStopwatchTest	스톱워치 상태를 활성화 하였는지 확인한다.	11. Start Stopwatch	R 4.1	Pass

11-2	createTickmanagementTest	Tickmanagement 를 잘 생성하였는지 확인한다.	11. Start Stopwatch	R 4.1	Pass
11-3	loadTimeTest	cs 단위로 시간이 1 씩 증가하는 것을 확인한다.	11. Start Stopwatch	R 4.1	Pass
12-1	deleteTickManagementTest	Tickmanagement 를 삭제하는 지 확인하고, 시간 증가가 멈추는지 확인한다.	12. Pause Stopwatch	R 4.2	Pass
12-2	inactiveStopWatchTest	스톱워치 상태를 비활성화 하였는지 확인한다.	12. Pause Stopwatch	R 4.2	Pass
13	loadTimeTest	1.리셋이 잘 되는 지 확인한다.(값이 모두 0) 2. 1 시간이 지났을 때 자동적으로 Reset Time 을 해주는 지 확인한다.	13. Reset Stopwatch	R 4.3	Pass
14	getLapTimeTest	돌아가고 있는 스톱워치의 타임과 랩타임의 시간이 일치하는 지 확인한다.	14. Set Lap Time	R 4.4	Pass
15-1	setCityTest	도시가 설정한 다음도시로 잘 넘어가는지 확인한다.	16. Set City	R 5.1	Pass
15-2	toggleSettingTest	세팅을 켜고 끌수 있는지 확인한다.	16. Set City	R 5.1	Pass
16	loadTimeTest	현재 시간과 일치하는지 확인한다.	15. Make World Clock	R 5.2	Pass
17	getSunriseTime()	같은 조건(2020,1,1,SEOUL	17. Calculate SR/SS	R 6.1	Pass

		)일 때 getSunsetTime()의 결과값이 일출시간 계산 공식의 기대 값과 같은지 확인			
18	getSunsetTime()	같은 조건(2020,1,1,SEOUL )일 때 getSunsetTime()의 결과값이 일출시간 계산 공식의 기대 값과 같은지 확인	17. Calculate SR/SS	R 6.1	Pass
19	setSSSR()	Geo 모드에서 buttonControl(UP)이 실행되었을 때 isSSSR 이 true 가 됨을 확인 buttonControl(DOW N)이 실행되었을 때 isSSSR 이 true 가 됨을 확인	18.set SR/SS	R.6.2	Pass
20-1	testEnterSetting	메뉴 설정모드에 진입했는지 확인한다.	19. Set Function	R 7.1	Pass
20-2	testCreateFunction	메뉴 설정 완료후 설정모드를 빠져나오면 모드가 잘 생성되는지 확인한다.	19. Set Function	R 7.1	Pass
20-3	testSelectFunction	메뉴설정모드에서 up 버튼으로 모드 간 이동이 이루어지는지 확인한다.	19. Set Function	R 7.1	Pass
20-4	testActivateFunction	메뉴설정모드에서 down 버튼으로 모드 활성화/비활성화 기능이 잘	19. Set Function	R 7.1	Pass

		작동하는지 확인한다.			
21	testChangeMode	아무 화면에서 모드 버튼을 누르면 화면이 순서대로 잘 전환되는지 확인한다.	20. Change Mode	R 7.2	Pass
22	testTimeout	clock 모드를 제외한 모드에서 15 초이상 버튼동작이 없을 시 clock 모드로 화면이 전환되는지 확인한다.	21. Time Out	R 7.3	Pass
23	testStopBuzzer	알람/타이머의 buzzer 가 울리고 있을 때 아무 버튼 동작으로 알림이 중지되는지 확인한다.	22. Stop Buzzer	R 7.4	Pass

### 1) Clock

▼	✔ Test Results	51 ms
▼	✔ ClockTest	51 ms
	✔ loadTimeTest()	33 ms
	✔ calculateGMTTest()	1 ms
	✔ setClockTest()	12 ms
	✔ nextSettingTest()	2 ms
	✔ toggleSettingTest()	3 ms

### 2) Alarm

▼	✔ Test Results	88 ms
▼	✔ AlarmTest	88 ms
	✔ setAlarmTimeTest()	49 ms
	✔ ringAlarmTest()	7 ms
	✔ toggleAlarmTest()	11 ms
	✔ nextAlarmTest()	15 ms
	✔ nextSettingTest()	3 ms
	✔ toggleSettingTest()	3 ms

### 3) Timer

▼	✔ Test Results	10 s 61 ms
▼	✔ TimerTest	10 s 61 ms
	✔ testNextSetting()	33 ms
	✔ testRingTimer()	5 s 13 ms
	✔ testMyTurn()	4 ms
	✔ testToggleSetting()	2 ms
	✔ testResetTimer()	2 ms
	✔ testStartTimer()	5 s 6 ms
	✔ testSetTimer()	1 ms

#### 4) Stopwatch

▼	✓	Test Results	489 ms
▼	✓	StopwatchTest	489 ms
	✓	deleteTickManagementTe	15 ms
	✓	getLapTimeTest()	64 ms
	✓	createTickmanagementTest	2 ms
	✓	activeStopwatchTest()	2 ms
	✓	loadTimeTest()	205 ms
	✓	inactiveStopWatchTest()	201 ms

#### 5) WorldClock

▼	✓	Test Results	50 ms
▼	✓	WorldClockTest	50 ms
	✓	loadTimeTest()	27 ms
	✓	setCityTest()	15 ms
	✓	toggleSettingTest()	8 ms

#### 6) Geo

▼	✓	Test Results	17 ms
▼	✓	GeoTest	17 ms
	✓	getSunsetTime()	17 ms
	✓	getSunriseTime()	0 ms
	✓	setSSSR()	0 ms

## 7) FunctionSelector

▼	✔ Test Results	20 s 53 ms
▼	✔ FunctionSelectorTest	20 s 53 ms
	✔ testEnterSetting()	30 ms
	✔ testTimeout()	20 s 6 ms
	✔ testActivateFunction()	5 ms
	✔ testCreateFunction()	8 ms
	✔ testSelectFunction()	2 ms
	✔ testChangeMode()	1 ms
	✔ testStopBuzzer()	1 ms

#### 4. Activity 2163. System Testing

Test Number	Test 항목	Description	Use Case	System Function	Pass
1	Clock Setting Test	시간 표시 화면에서 셋 버튼을 누르게 되면 시계 설정을 변경할 수 있다. 처음 셋 버튼을 누르게 되면 시간을 변경할 수 있으며 셋 버튼을 누를 때 마다 분, 초, 년, 월, 일, 도시, 시간표현 방식(12/24H) 순으로 설정을 변경할 수 있다. 시간표현방식을 설정한 이후에 셋 버튼을 한번 더 누르게 되면 설정이 종료되고 변경된 시간으로 시계가 동작하는 것을 확인한다.	1.Clock Setting	R 1.1	Pass
2	Clock Setting Test	현재 시간 /년 일 입력(년 월 일 시 분초 오버 없음) - 요일 조건 달성 시 요일 자동설정	1.Clock Setting	R 1.1	Pass
3	Clock Setting Test	초기값을 9999 년 12 월 31 일 오후 12 시 59 분 59 초로 입력 후 Tick Management 증가 - > 0000 년 1 월 1 일 오전 00:00:00 으로 초기화 한다.	1.Clock Setting	R 1.1	Pass
4	Clock Setting Test	정해진 시간 값 오버 입력 (ex 25 시간, 13 월)시도 데이터 엔드 값(혹은 음수) 도달 시 시작점 반환한다.	1.Clock Setting	R 1.1	Pass
5	Clock Setting Test	시간 설정 도중 MODE Button 을 누르면 설정 모드를 벗어나고, 마지막 설정 시점까지의 변경사항이 적용된다.	1.Clock Setting	R 1.1	Pass
6	Set Alarm Time Test	SET Button 을 통해 Alarm Setting Mode 에 진입하며 SET Button 을 누를 때마다 시, 분, 반복을 설정할 수 있고 한 번 더 누르면 Alarm Setting Mode 에서 벗어난다.	2. Set Alarm Time	R 2.1	Pass

		각각의 Setting Mode 에서는 UP/DOWN Button 을 통해 값을 변경한다.			
7	Set Alarm Time Test	알람을 설정할 때 시가 0~23 시 범위를 넘어가거나 분이 0~59 분 범위를 넘어가면 단위를 올림 또는 내림 하지 않고 반대쪽 끝으로 이동한다.	2. Set Alarm Time	R 2.1	Pass
8	Set Alarm Time Test	요일 설정에서 UP, DOWN Button 을 통해 반복 설정을 하며, 그 어떤 요일도 선택되지 않았을 경우 반복을 하지 않는다.	2. Set Alarm Time	R 2.1	Pass
9	Set Alarm Time Test	Alarm 설정 도중 MODE Button 을 누르면 현재까지 설정한 알람 시간이 반영된 채 다음 모드로 넘어간다.	2. Set Alarm Time	R 2.1	Pass
10	Ring Alarm Test	요일 반복 여부와 상관없이 Alarm 시간을 설정하고 알람을 켜지 않으면 해당 시간에 도달해도 작동하지 않는다.	3. Ring Alarm	R 2.2	Pass
11	Ring Alarm Test	A1 ~ A4 중 활성화된 Alarm 의 시간이 도래했을 때 반복 설정이 되어 있지 않다면 Beep 을 하고 해당 Alarm 을 비활성화한다.	3. Ring Alarm	R 2.2	Pass
12	Ring Alarm Test	A1 ~ A4 중 활성화된 Alarm 의 시간이 도래했을 때 반복 설정이 되어 있으며 현재 요일과 일치한다면 Beep 을 하고 해당 Alarm 을 비활성화 하지 않는다.	3. Ring Alarm	R 2.2	Pass
13	Ring Alarm Test	A1 ~ A4 중 활성화된 Alarm 의 시간이 도래했을 때 반복 설정이 되어 있으며 현재 요일과 일치하지 않는다면 Beep 을 하지 않고 해당 알람을 비활성화 하지도 않는다.	3. Ring Alarm	R 2.2	Pass
14	Ring Alarm Test	A1 ~ A4 중 활성화된 Alarm 의 시간이 도래했을 때 울려야 할	3. Ring Alarm	R 2.2	Pass

		Alarm 이 둘 이상 존재한다면 하나의 Alarm 에 대해서만 Beep 을 하고 중복되는 모든 Alarm 에 대하여 반복 설정이 되어 있다면 그대로 유지하고 그렇지 않다면 비활성화 한다..			
15	Toggle Alarm Test	DOWN Button 을 눌러 Alarm 을 활성화하거나 비활성화 하고 모드를 변경했을 때 활성화된 Alarm 의 indicator 가 유지된다.	4. Toggle Alarm	R 2.3	Pass
16	Next Alarm Test	다음 Alarm 을 화면에 띄우는 UP Button 을 누를 때마다 A1 ~ A4 를 한 칸씩 원형 순회한다.	5. Next Alarm	R 2.4	Pass
17	Set Timer Test	타이머 시간 설정을 0 분 0 초에서 1 분 감소시켜 59 분 0 초가 되는지 확인한다. 마찬가지로 0 분에서 1 초 감소시켜 59 분 59 초가 되는지 확인한다.	Set Timer	R 3.1	Pass
18	Set Timer Test	타이머 시간을 59 분 59 초로 설정 후 MODE Button 으로 메뉴를 순회하여 다시 타이머로 돌아왔을 때 설정 시간이 그대로 있는지 확인한다.	Set Timer	R 3.1	Pass
19	Start / Restart Timer Test	타이머 시간을 2 분으로 설정 후 시간이 줄어들고 00:00 에서 멈추고 버저가 울리는 것을 확인한다.	Start / Restart Timer	R 3.2	Pass
20	Start / Restart Timer Test	타이머 시간을 2 분으로 맞춰 둔 후 MODE Button 을 눌러 다른 화면에 있을 때도 버저가 울리는지 확인한다.	Start / Restart Timer	R 3.2	Pass
21	Start / Restart Timer Test	00 :00 상태에서 UP Button 을 눌러도 타이머가 작동하지 않는 것을 확인한다.	Start / Restart Timer	R 3.2	Pass
22	Start / Restart Timer Test	타이머 시간을 2 분으로 설정 후 1 분에서 UP Button 을 눌러 일시정지 후, 다시 UP Button 을	Start / Restart Timer	R 3.2	Pass

		눌러 카운트 다운이 재시작 되는지 확인한다.			
23	Pause Timer Test	타이머 시간을 2 분으로 설정 후 1 분에서 UP Button 을 눌러 카운트 다운이 일시정지 되는지 확인한다.	Pause Timer	R 3.3	Pass
24	Reset Timer Test	타이머를 2 분으로 설정 후 1 분에 일시 정지 후 DOWN Button 을 눌러 잘 초기화 되는지 확인한다.	Reset Timer	R 3.4	Pass
25	Reset Timer Test	타이머를 2 분으로 설정 후 동작하여 카운트 다운 도중 DOWN Button 이 작동하지 않아 초기화 되지 않는 것을 확인한다.	Reset Timer	R 3.4	Pass
26	Start Stopwatch Test	스톱워치가 동작하지 않을 때, SET Button 을 눌러 cs 단위로 시간이 증가하는 것을 확인한다.	11. Start Stopwatch	R 4.1	Pass
27	Start Stopwatch Test	스톱워치가 동작할 때 MODE Button 을 눌렀을 스톱워치가 동작됨을 확인한다.	11. Start Stopwatch	R 4.1	Pass
28	Start Stopwatch Test	스톱워치의 시간이 59:59:99 를 넘어가면 스톱워치가 멈추고 stop time 과 lap time 이 초기화 되는 것을 확인한다.	11. Start Stopwatch	R 4.1	Pass
29	Start Stopwatch Test	스톱워치를 동작할 때, 15 초가 흘러도 Time Out 이 수행되지 않음을 확인한다.	11. Start Stopwatch	R 4.1	Pass
30	Pause Stopwatch Test	스톱워치가 동작할 때, SET Button 을 누르면 동작을 정지하는 것을 확인한다.	12. Pause Stopwatch	R 4.2	Pass
31	Reset Stopwatch Test	스톱워치가 동작하지 않을 때, DOWN Button 을 누르면 stop time 과 lap time 이 초기화 되는 것을 확인한다.	13. Reset Stopwatch	R 4.3	Pass
32	Reset Stopwatch Test	스톱워치가 동작 중에는 reset 이 수행되지 않음을 확인한다.	13. Reset Stopwatch	R 4.3	Pass

33	Set Lap Time Test	스톱워치가 동작할 때, 1 분, 2 분, 3 분마다 UP Button 을 눌러 lap time 이 잘 기록되는지 확인한다.	14. Set Lap Time	R 4.4	Pass
34	Set Lap Time Test	스톱워치가 동작하지 않을 때, Set Lap Time 이 수행되지 않음을 확인한다.	14. Set Lap Time	R 4.4	Pass
35	Calculate WorldTime Test	설정된 도시의 GMT 를 기준으로 시간을 계산하고, 날짜가 변경될 경우 날짜도 반영한다.	15. Make World Clock	R 5.1	Pass
36	Set City Test	SET Button 을 눌러 정해진 도시명의 리스트를 순환할 수 있는지 확인한다.	16. Set City	R 5.2	Pass
37	Set SR/SS Test	SET Button 을 눌러 도시 시간을 순차적으로 변경할 수 있음을 확인한다. 또한 도시가 변경되었을 경우 일출 시간이 먼저 표시됨을 확인한다.	18. Set SR/SS	R 6.2	Pass
38	Set SR/SS Test	UP 버튼을 눌렀을 때, SR 로 설정되는지 확인한다. DOWN 버튼을 눌렀을 때 SS 로 설정되는지 확인한다.	18. Set SR/SS	R 6.2	Pass
39	Calculate SR/SS Test	SR 혹은 SS 설정하였을 때, SR 에서 일출시간이, SS 에서 일몰시간이 표시되는지 확인한다. 양쪽모두 Clock 의 Date,City 와 Date City 가 같은지 확인한다.	17. Calculate SR/SS	R 6.1	Pass
40	Set Function Test	메뉴 설정으로 들어가면 Clock 을 제외한 다른 메뉴를 각각 활성화/비활성화 시킬 수 있는지 확인한다.	19. Set Function	R 7.1	Pass
41	Set Function Test	C A T S 활성화 상태에서 A 를 비활성화 하고 W 를 활성화하여 설정을 빠져나오고 결과가 잘 반영되는지 확인한다.	19. Set Function	R 7.1	Pass
42	Set Function Test	C A T S 활성화 상태에서 A 를 비활성화하고 설정 버튼을 눌렀을	19. Set Function	R 7.1	Pass

		때 설정을 빠져 나올 수 없는지 확인한다.			
43	Set Function Test	메뉴 설정 상태에서 MODE Button 을 눌렀을 때 설정 중이던 사항이 반영되지 않는 것을 확인한다.	19. Set Function	R 7.1	Pass
44	Change Mode Test	시계의 각 상태에서 MODE Button 을 눌렀을 때 모드가 정해진 순서대로 화면이 전환되는 지 확인하다. 타이머에서 알람 화면으로 전환되었을 때, 항상 첫번째 알람 화면이 나타남을 확인한다	20. Change Mode	R 7.2	Pass
45	Time Out Test	Clock 을 제외한 모드에 15 초 이상 아무 버튼 작동 없이 있으면 time out 되어 clock 모드로 변경된다.	21. Time Out	R 7.3	Pass
46	Stop Buzzer Test	알람 시간이 되어 Buzzer 가 울리고 있을 때, 아무 버튼을 눌러 버저음이 멈추는 것을 확인한다.	22. Stop Buzzer	R 7.4	Pass
47	Stop Buzzer Test	타이머 시간이 되어 Buzzer 가 울리고 있을 때, 아무 버튼을 눌러 버저음이 멈추는 것을 확인한다.	22. Stop Buzzer	R 7.4	Pass
48	Tick Management Test	Time Keeping 또는 Stopwatch 카운트 다운 도중 100ms 마다 sec 에, 60s 마다 min 에, 60min 마다 hr 에, 24hr 마다 day 에, 28~31 마다 month 에, 12month 마다 year 에 1 값이 추가되어 자릿수를 올림하는 것을 확인한다. Timer 의 카운트 다운 경우는 반대로 자릿수 내림되는 것을 확인한다.	23. Tick Management	R 8.1	Pass

## 5. Activity 2167. Testing Traceability Analysis

use case	operation
1.Clock Setting	1-17,66,67,60,23
2.Set Alarm Time	21-29,68,67,66,5-6,9-11,19-20
3. Ring Alarm	hidden
4. Toggle Alarm	21,66,69
5. Next Alarm	21,68,69,7,15
6. Set Timer	34-37,68,67,66,5-6,9-11,19-20,,7,15
7.Start/Restart Timer	38-41,19-20,5,7,15,66
8. Pause Timer	38,41,66
9.Reset Timer	9,10,20,42,7,66
10. Ring Timer	hidden
11. Start Stopwatch	66,43-48,50,19-20,5,7,15
12.Pause Stopwatch	46,47,43,66
13.Reset Stopwatch	43,66,48,20,,9,10,6,12-14,7,15
14.Set Lap Time	43,66,49,,19,27,70,12,14,17
15. Make World	hidden
16. Set City	51,52,71,53,54,17,66
17. Calculate SR/SS	hidden
18. Set SR/SS	56,57,66
19. Set Funcion	65,66,60,59,72
20.Change Mode	64-66,73
21.Time Out	hidden
22.Stop Buzzer	33,66,63
23. Tick	hidden

num	Operation in Interaction diagram
1	toggleSetting()
2	buttonControl()
3	blinkSector(Time)
4	setClock()
5	checkFormat()
6	setHr()
7	setTime()
8	nextSetting()
9	setMin()
10	setSec()
11	setH1224()
12	setYear()
13	setMonth()
14	setDay()
15	setDate()
16	setClock()
17	setCity()
18	loadTime()
19	getCs()
20	setCs()
21	buttonControl()
22	toggleAlarmSetting()
23	setBlinkSector()
24	toggleSetting()
25	setAlarmTime()
26	nextSetting()
27	getMin()
28	getWeekday()
29	setWeekday()
30	checkAlarmTime()
31	readCurrentClock()
32	ringAlarm()
33	stopBuzzer()

34	buttonControl()
35	toggleSetting()
36	setBlinkSector()
37	setTimer()
38	startTimer()
39	createTimerTickManagement()
40	loadTime()
41	deleteTimerTickManagement()
42	setTimer()
43	buttonControl()
44	activeStopwatch()
45	createTickManagement()
46	inactiveStopwatch()
47	deleteTickManagement()
48	resetStopwatch()
49	getLapTime()
50	loadTime()
51	readCurrentDate()
52	readCurrentCity()
53	setNextCity()
54	calculateGMT()
55	setCity()
56	calculateSRSS()
57	setSSSR()
58	timeout()
59	selectFunction()
60	enterSetting
61	selectFunction()
62	activateFunction()
63	stopBuzzer()
64	changeMode()
65	buttonControl()
66	pushButton()
67	getHr()
68	nextAlarm()
69	setCurrentTime()
70	getSec()
71	readCurrentTime()
72	setMenu()
73	setCurrentMode()



Unit Test
calculateMTRate
setClockTest
nextSettingTest
toggleSettingTest
loadTimeTest
toggleSettingTest
nextSettingTest
setAlarmTimeTest
ringAlarmTest
toggleAlarmTest
nextAlarmTest
testToggleSetting
testNextSetting
testSetTimer
testStartTimer
testResetTimer
testRingTimer
activeStopwatchTest
createTickmanagementTest
loadTimeTest
deleteTickManagementTest
inactiveStopWatchTest
loadTimeTest
getLapTimeTest
setCityTest
toggleSettingTest
loadTimeTest
getSunriseTime()
getSunsetTime()
setSSR()
testEnterSetting
testCreateFunction
testSelectFunction
testActivateFunction
testChangeMode
testTimeout
testStopBuzzer