OOPT 2050: Implement

Team6: Advanture Company

Advanture Digital Watch

팀장 정주원

팀원 강현우 김나연 송승현 이동현

0. Demo video

1. Activity 2152. Implement Windows

2. Activity 2155. Write Test Code

3. Activity 2161. Unit Testing

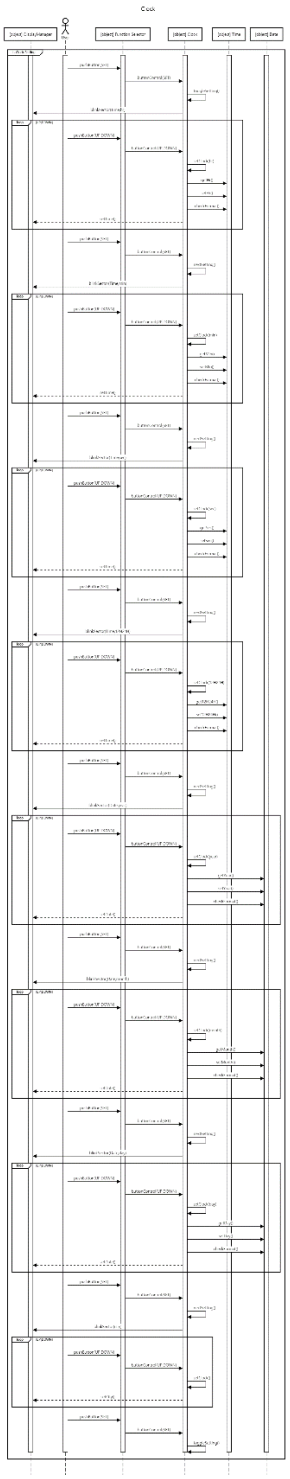
4. Activity 2163. System Testing

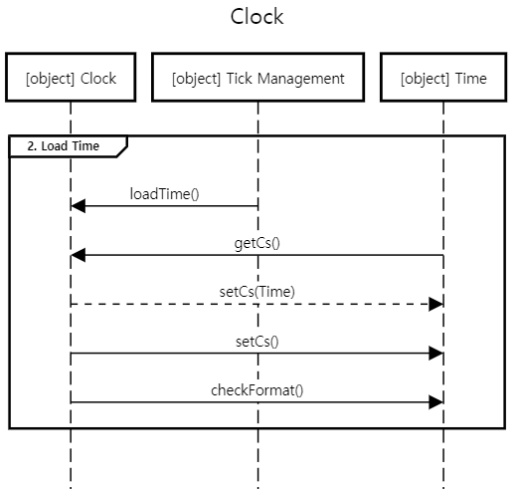
5. Activity 2167. Testing Traceability Analysis

0. Demo

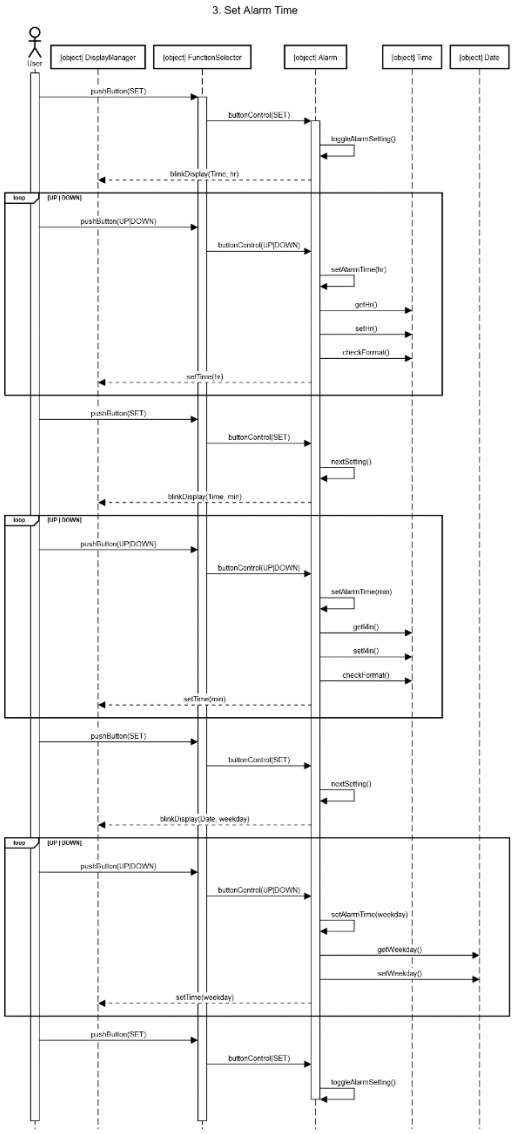
. Activity 2152. Implement Windows

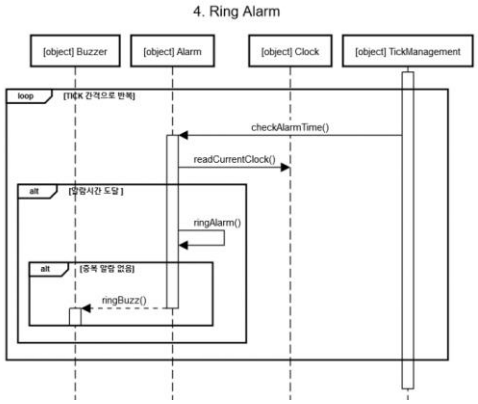
1) Clock

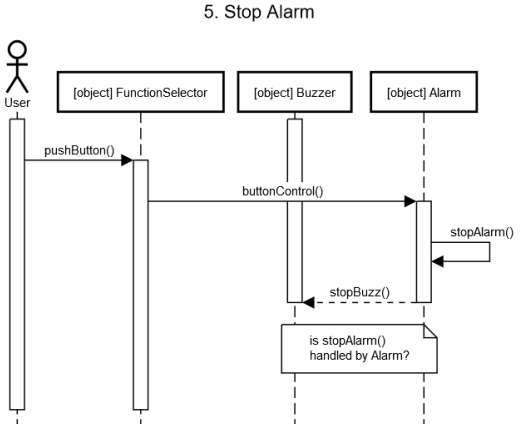


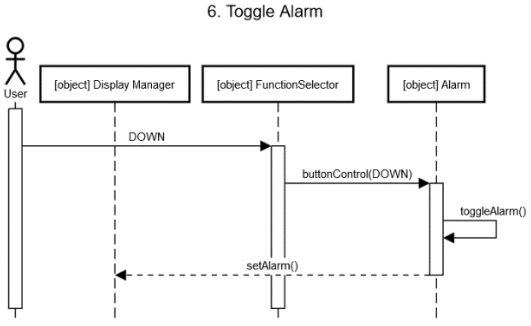


2) Alarm

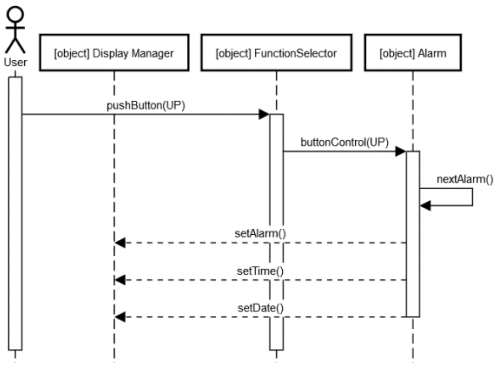

3. Set Alarm Time
User
[object] DisplayManager
[object] FunctionSelector
[object] Alarm
[object] Time
[object] Date
pushButton(SET)
buttonControl(SET)
toggleAlarmSetting()
blinkDisplay(Time, hr)
loop
[UP | DOWN]
pushButton(UP|DOWN)
buttonControl(UP|DOWN)
setAlarmTime(hr)
getHr()
setHr()
checkFormat()
setTime(hr)
pushButton(SET)
buttonControl(SET)
nextSetting()
blinkDisplay(Time, min)
loop
[UP | DOWN]
pushButton(UP|DOWN)
buttonControl(UP|DOWN)
setAlarmTime(min)
getMin()
setMin()
checkFormat()
setTime(min)
pushButton(SET)
buttonControl(SET)
nextSetting()
blinkDisplay(Date, weekday)
loop
[UP | DOWN]
pushButton(UP|DOWN)
buttonControl(UP|DOWN)
setAlarmTime(weekday)
getWeekday()
setWeekday()
setTime(weekday)
pushButton(SET)
buttonControl(SET)
toggleAlarmSetting()


4. Ring Alarm
[object] Buzzer
[object] Alarm
[object] Clock
[object] TickManagement
loop
[TICK 간격으로 반복]
checkAlarmTime()
readCurrentClock()
alt
[알람시간 도달]
ringAlarm()
alt
[중복 알람 없음]
ringBuzz()


5. Stop Alarm
User
[object] FunctionSelector
[object] Buzzer
[object] Alarm
pushButton()
buttonControl()
stopAlarm()
stopBuzz()
is stopAlarm()
handled by Alarm?

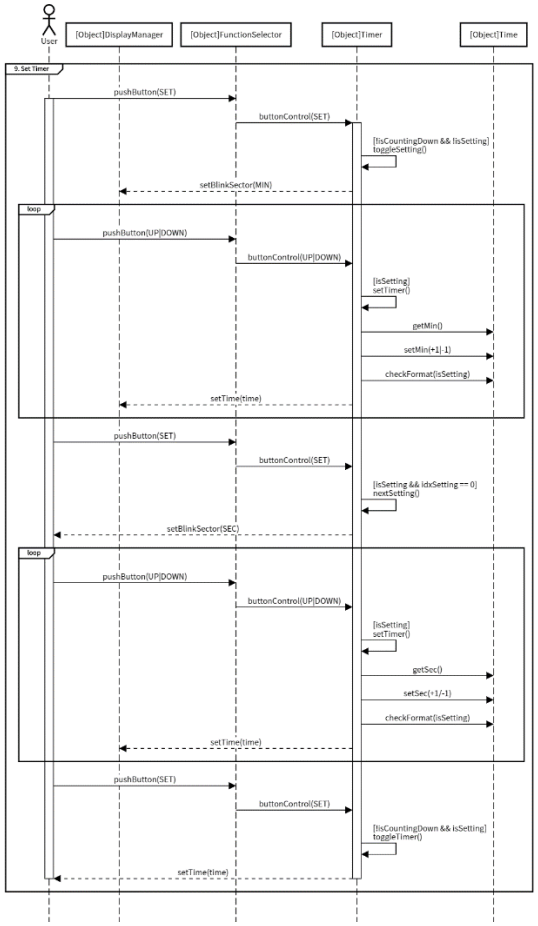

6. Toggle Alarm
User
[object] Display Manager
[object] FunctionSelector
[object] Alarm
DOWN
buttonControl(DOWN)
toggleAlarm()
setAlarm()

7. Next Alarm

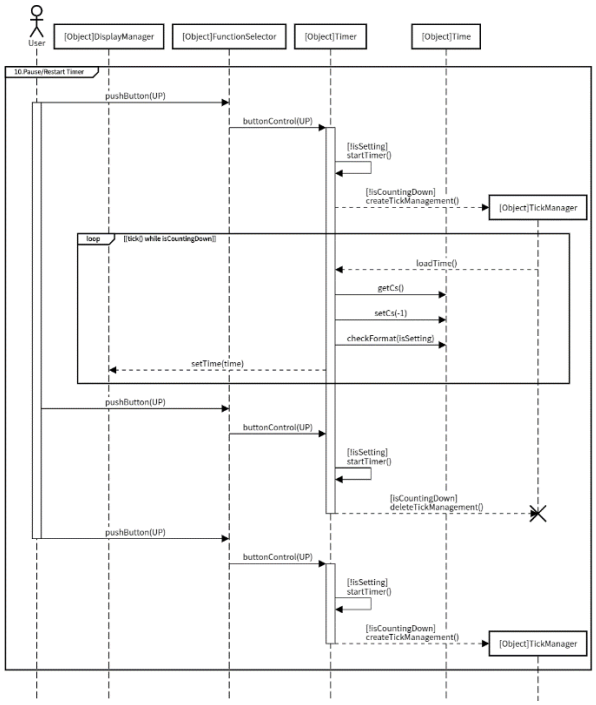


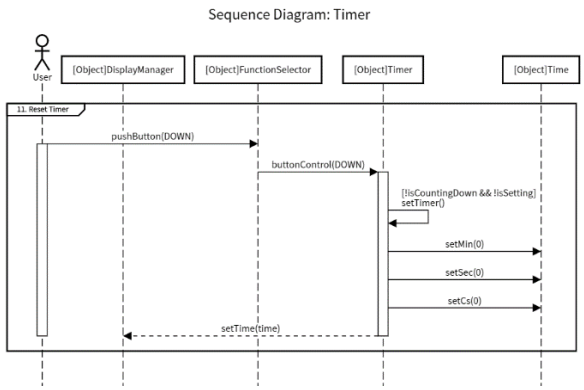
3) Timer

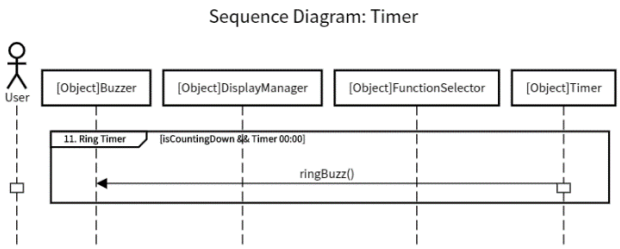
Sequence Diagram: Timer



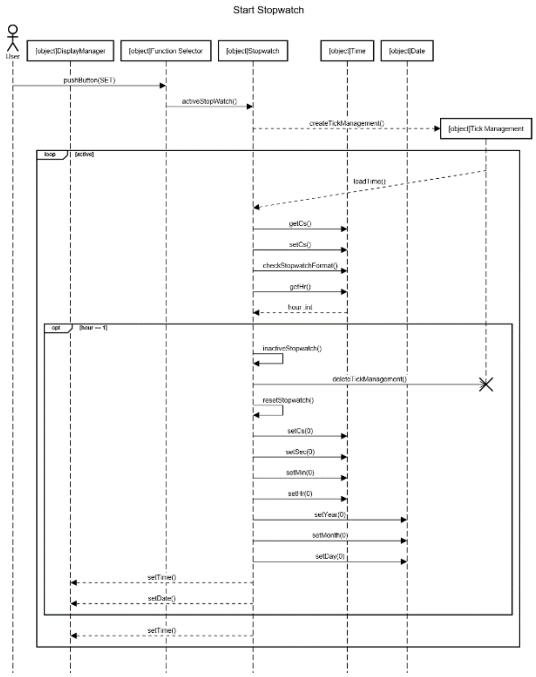
Sequence Diagram: Timer

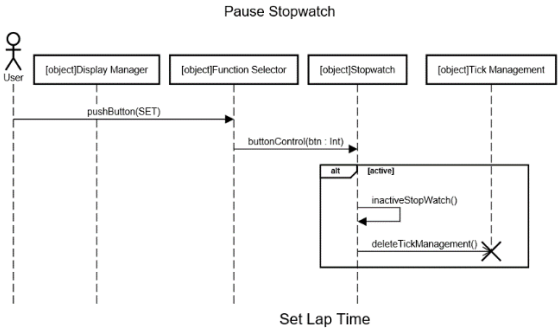


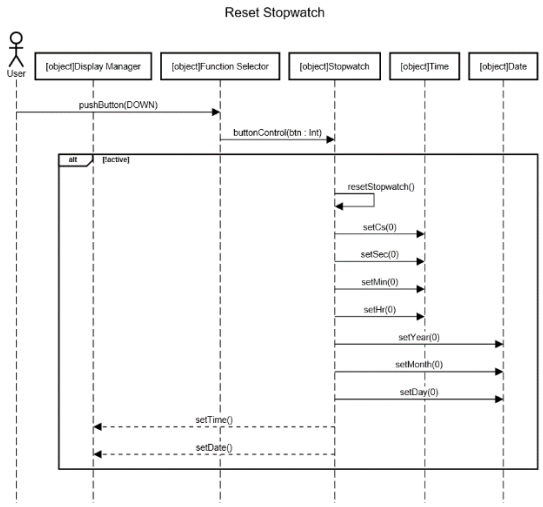

Sequence Diagram: Timer
User
[Object]DisplayManager
[Object]FunctionSelector
[Object]Timer
[Object]Time
11. Reset Timer
pushButton(DOWN)
buttonControl(DOWN)
[!isCountingDown && !isSetting]
setTimer()
setMin(0)
setSec(0)
setCs(0)
setTime(time)

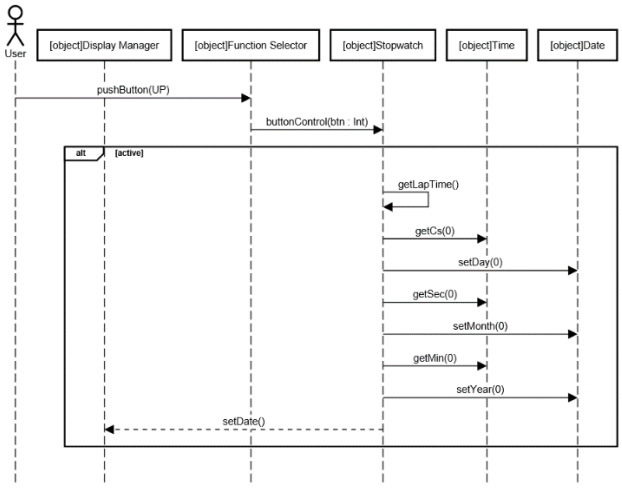

Sequence Diagram: Timer
User
[Object]Buzzer
[Object]DisplayManager
[Object]FunctionSelector
[Object]Timer
11. Ring Timer
[isCountingDown && Timer 00:00]
ringBuzz()

4) Stopwatch

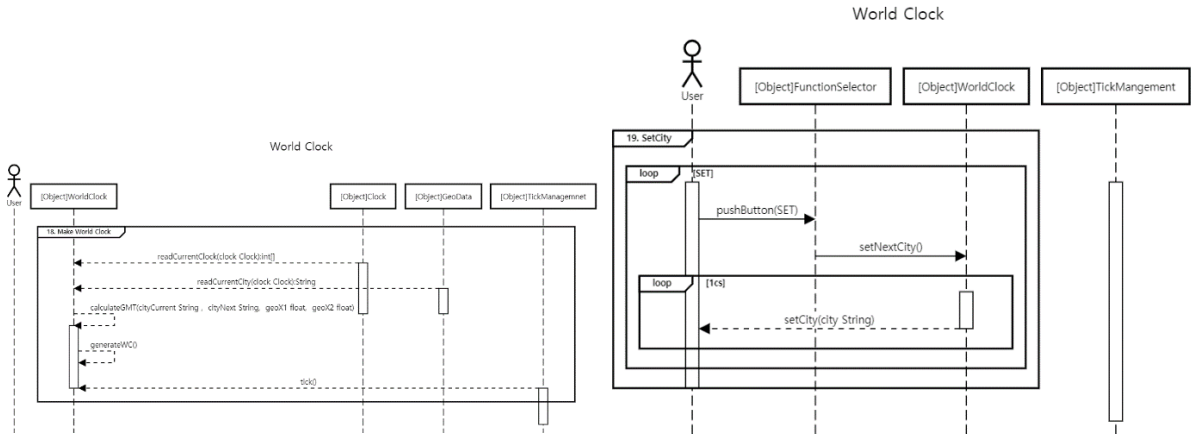

Start Stopwatch
User
[object]DisplayManager
[object]Function Selector
[object]Stopwatch
[object]Time
[object]Date
pushButton(SET)
activeStopWatch()
createTickManagement()
[object]Tick Management
loop
[active]
loadTime()
getCs()
setCs()
checkStopwatchFormat()
getHr()
hour_int
opt
[hour == 1]
inactiveStopwatch()
deleteTickManagement()
resetStopwatch()
setCs(0)
setSec(0)
setMin(0)
setHr(0)
setYear(0)
setMonth(0)
setDay(0)
setTime()
setDate()
setTime()


Pause Stopwatch
User
[object]Display Manager
[object]Function Selector
[object]Stopwatch
[object]Tick Management
pushButton(SET)
buttonControl(btn : Int)
alt
[active]
inactiveStopWatch()
deleteTickManagement()

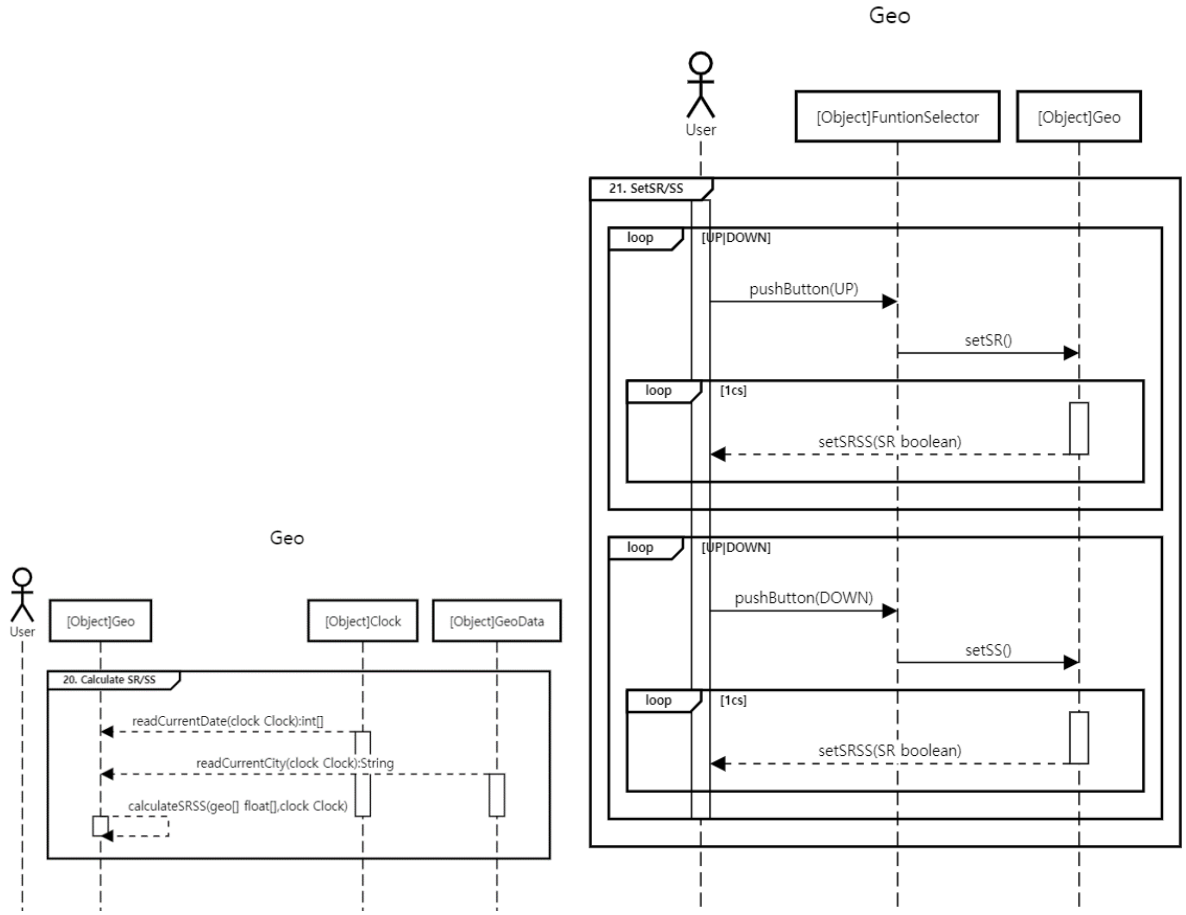

Reset Stopwatch
User
[object]Display Manager
[object]Function Selector
[object]Stopwatch
[object]Time
[object]Date
pushButton(DOWN)
buttonControl(btn : Int)
alt
[active]
resetStopwatch()
setCs(0)
setSec(0)
setMin(0)
setHr(0)
setYear(0)
setMonth(0)
setDay(0)
setTime()
setDate()


Set Lap Time
User
[object]Display Manager
[object]Function Selector
[object]Stopwatch
[object]Time
[object]Date
pushButton(UP)
buttonControl(btn : Int)
alt
[active]
getLapTime()
getCs(0)
setDay(0)
getSec(0)
setMonth(0)
getMin(0)
setYear(0)
setDate()

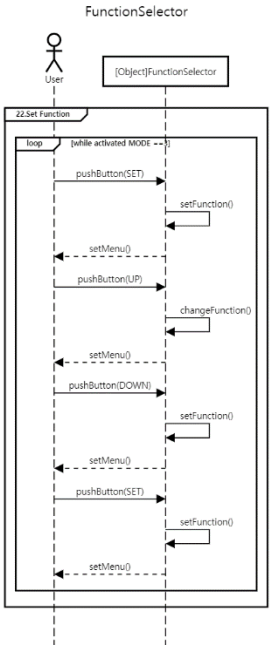
5) WorldTime

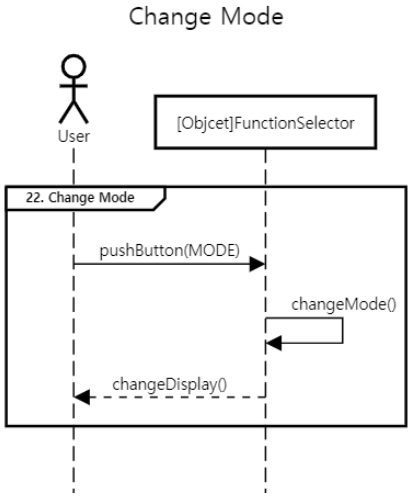


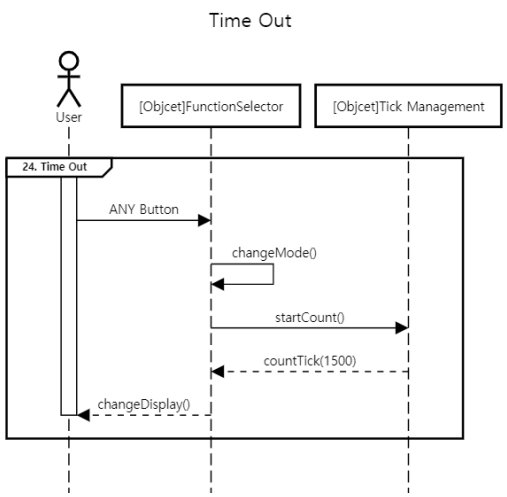
6) Geo

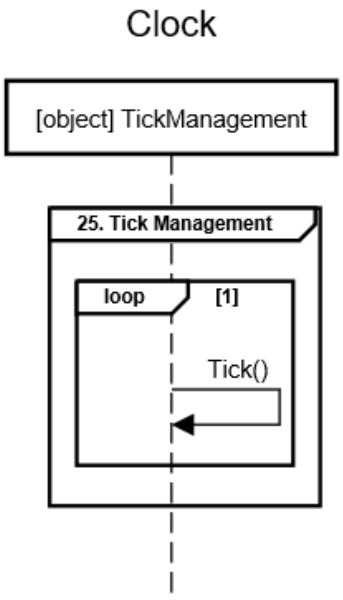


7) Function Selector









2. Activity 2155. Write Test Code

1) Clock

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

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

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

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

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

2) Alarm

```
@Test
public void toggleSettingTest() {
    assertNotEquals( unexpected: -1, alarm.buttonControl(Main.SET));
}
```

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

```
@Test
public void setAlarmTimeTest() {
    alarm.buttonControl(Main.SET); // enter setting mode
    assertEquals( expected: 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)
    alarm.checkAlarmTime();
    assertTrue(fs.getBuzzer().isRing());

    buzzerThread.interrupt();
}
```

```java
@Test
public void toggleAlarmTest() {
    assertNotEquals(alarm.buttonControl(Main.DOWN), alarm.buttonControl(Main.DOWN));
}

@Test
public void nextAlarmTest() {
    assertEquals( expected: (alarm.buttonControl(Main.UP)+1)%4, (alarm.buttonControl(Main.UP)));
}
```

3) Timer

```java
@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());
}
```

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

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

```java
@Test
public void testStartTimer(){
    //start timer
    timer.buttonControl(Main.SET);
    timer.buttonControl(Main.UP);
    timer.buttonControl(Main.SET);
    timer.buttonControl(Main.SET);
    assertEquals( expected: 1, timer.getTime().getMin());
    timer.buttonControl(Main.UP);
    assertNotEquals( unexpected: 0, timer.getTime().getSec());
    assertNotEquals( unexpected: 1, timer.getTime().getMin());
    //pause timer
    int temp;
    timer.buttonControl(Main.UP);
    temp = timer.getTime().getTotalCs();
    assertEquals(temp, timer.getTime().getTotalCs());
}
```

```java
@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
    fs.pushButton(Main.UP); //set timer sec 1
    fs.pushButton(Main.SET); //mode: stopWatch
    fs.pushButton(Main.SET); //mode: clock
    fs.pushButton(Main.SET); //mode: timer
    assertFalse(timer.isSetting());
    assertEquals( expected: 1, timer.getTime().getSec());
}
```

```java
@Test
public void testRingTimer() {
    timer.buttonControl(Main.SET);
    timer.buttonControl(Main.SET);
    timer.buttonControl(Main.UP);
    timer.buttonControl(Main.SET);
    timer.buttonControl(Main.UP);
    while(timer.getTime().getTotalCs() > 0){
        System.out.println("running");
    }
    assertTrue(fs.getBuzzer().isRing());
}
```

4) Stopwatch

```
public class StopwatchTest {
    FunctionSelector fs = new FunctionSelector();
    public final StopWatch stopwatch = new StopWatch(fs.getDM());

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

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

```
    @Test
    public void loadTimeTest() throws InterruptedException {
        //check time running
        assertEquals( expected: 0, stopwatch.getTime().getCs());
        stopwatch.createTickmanagement();
        assertNotEquals( unexpected: 0, stopwatch.loadTime().getCs());
        //check time over
        stopwatch.setTime();
        Thread.sleep( millis: 20);
        assertEquals( expected: 0, stopwatch.getTime().getHr());
        assertEquals( expected: 0, stopwatch.getTime().getCs());
        assertEquals( expected: 0, stopwatch.getTime().getSec());
        assertEquals( expected: 0, stopwatch.getTime().getMin());
        assertTrue(stopwatch.activeStopwatch());

    }

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

```
public void inactiveStopWatchTest() throws InterruptedException {
    stopwatch.activeStopwatch();
    Thread.sleep( millis: 20);
    assertEquals( expected: 1, stopwatch.getTime().getCs());
    assertFalse(stopwatch.inactiveStopWatch());
    Thread.sleep( millis: 20);

    assertEquals( expected: 2,stopwatch.getTime().getCs());
    assertNull(stopwatch.getTick());
}

@Test
public void getLapTimeTest() throws InterruptedException {
    //positive
    stopwatch.activeStopwatch();
    Thread.sleep( millis: 30);
    stopwatch.getLapTime();
    assertEquals( expected: 2,stopwatch.getTime().getCs());
    assertEquals( expected: 2,stopwatch.getLaptime().getDay());
    Thread.sleep( millis: 30);
    stopwatch.getLapTime();
    assertEquals( expected: 5,stopwatch.getTime().getCs());
    assertEquals( expected: 5,stopwatch.getLaptime().getDay());

}
```

5) WorldClock

```
@Test
public void setCityTest() {
    worldClock.buttonControl(Main.SET); // enter setting mode
    assertEquals( expected: (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( expected: (worldClock.loadTime().getCs()+1)%100, worldClock.loadTime().getCs()); }
```

6) Geo

```
class GeoTest {

    FunctionSelector fs = new FunctionSelector();
    Geo geoTest = new Geo(fs.getDM(),fs.getClock());

    @Test
    void isSSSR() {

        geoTest.buttonControl(Main.UP);
        assertEquals( expected: true,geoTest.isSSSR());

        geoTest.buttonControl(Main.DOWN);
        assertEquals( expected: false,geoTest.isSSSR());

    }
```

7) FunctionSelector

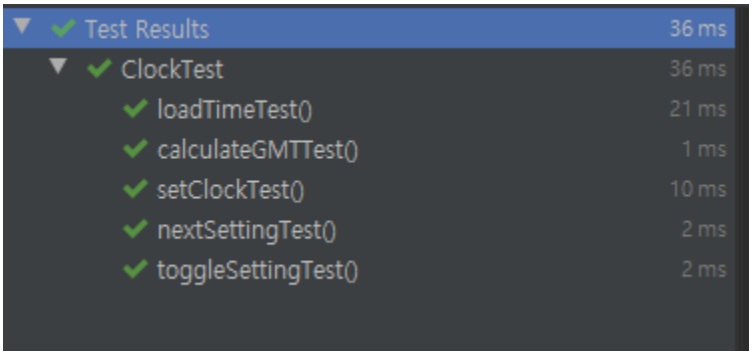
```
@Test
public void testEnterSetting(){
    fs.pushButton(Main.SET);
    assertTrue(fs.isSetting());
}

@Test
public void testCreateFunction(){
    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( expected: 3, fs.getTempNumOfModes());
    fs.pushButton(Main.SET);
    assertFalse(fs.isSetting());
    assertNull(fs.getTimer());
    assertNotNull(fs.getWorldClock());
}
```

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

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

@Test
public void testStopBuzzer(){
    fs.getBuzzer().ringBuzzer();
    fs.pushButton(Main.MODE);
    assertFalse(fs.getBuzzer().isRing());
}
```

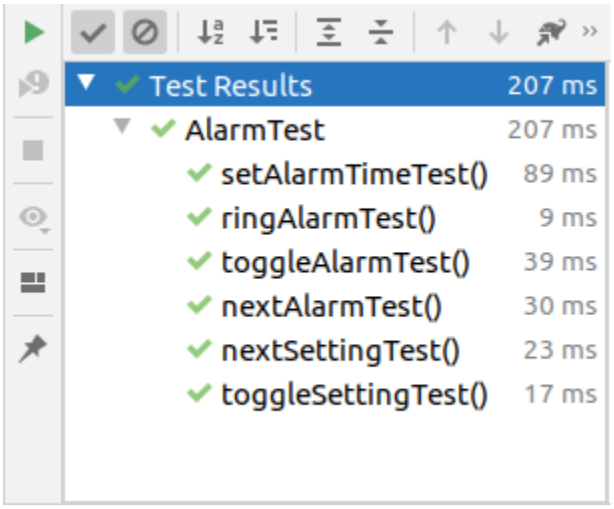
```java
public void testChangeMode(){
    int temp = fs.getCurrentMode();
    fs.pushButton(Main.MODE);
    assertNotEquals(temp, fs.getCurrentMode());
    fs.pushButton(Main.MODE);
    fs.pushButton(Main.MODE);
    fs.pushButton(Main.MODE);
    assertEquals(temp, fs.getCurrentMode());
}

@Test
public void testTimeout() {
    fs.pushButton(Main.MODE);
    while(fs.getTickCount() < 15000){

    }
    assertEquals( expected: 0, fs.getCurrentMode());
}
```

3. Activity 2161. Unit Testing

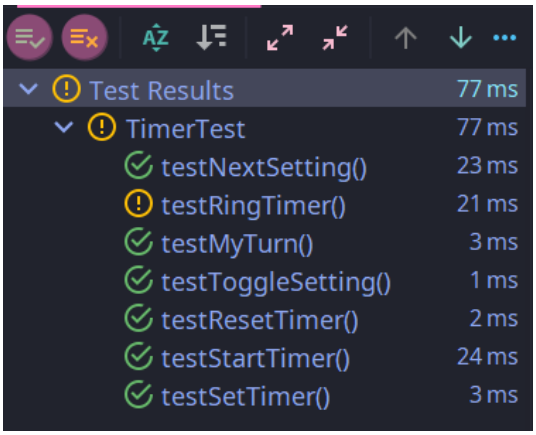
1) Clock



2) Alarm

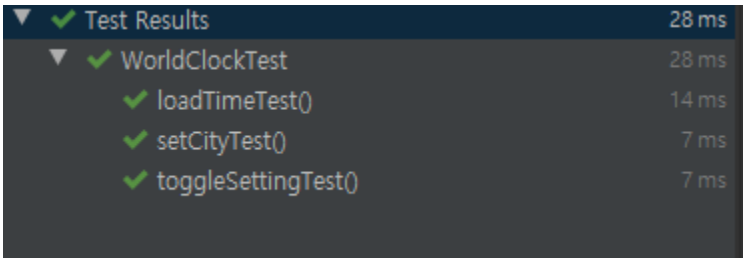


3) Timer

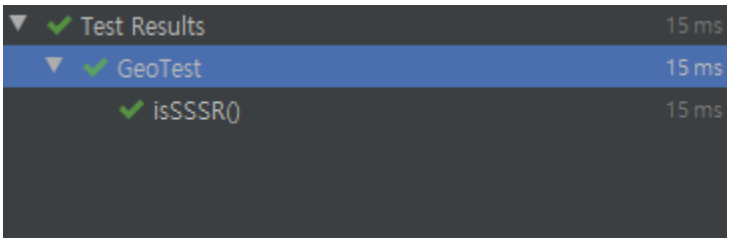


4) Stopwatch

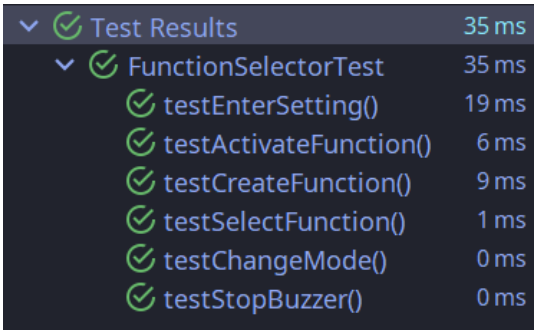
5) WorldClock

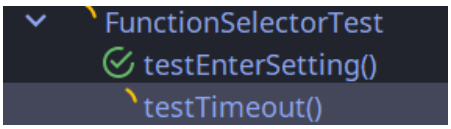


6) Geo



7) FunctionSelector





4. Activity 2163. System 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	P
1-2	setClockTest	UP 버튼이 눌러졌을 때 잘 작동하는지 확인한다.	1. Clock Setting	R 1.1	P
1-3	nextSettingTest	세팅할 때 다음 세팅으로 잘 넘어가는지 확인한다.	1. Clock Setting	R 1.1	P
1-4	toggleSettingTest	세팅일 때 으로 넘어가는지 확인한다.	1. Clock Setting	R 1.1	P
2	loadTimeTest	지금 현재 시간이 잘 확인되는지 확인한다.	2. Load Time	R 1.2	P
3-1	toggleSettingTest	설정 모드로 진입되는지 확인한다.	3. Set Alarm Time	R 2.1	P
3-2	nextSettingTest	설정 모드에서 설정할 단위를 넘길 수 있는지 확인한다.	3. Set Alarm Time	R 2.1	P
3-3	setAlarmTimeTest	버튼 입력에 따 설정할 단위의 값을 1 씩 변경할 수 있는지 확인한다.	3. Set Alarm Time	R 2.1	P

4	ringAlarmTest	활성화된 알람과 현재 시간이 같을 때 buzzer 가 활성화되는지 확인한다.	4. Ring Alarm	R 2.2	P
5	toggleAlarmTest	알람을 켜고 끌 수 있는지 확인한다.	5. Toggle Alarm	R 2.3	P
6	nextAlarmTest	버튼 입력에 따라 현재 알람에서 다음 알람으로 제대로 넘어가는지 확인한다.	6. Next Alarm	R 2.4	P
7	testToggleSetting	설정모드로 진입하는지 확인한다.	7. Set Timer	R 3.1	P
7	testNextSetting	분 설정에서 초 설정으로 넘어가는지 확인한다.	7. Set Timer	R 3.1	P
7	testSetTimer	분과 초가 up/down 버튼에 따라 설정값이 증가/감소 하는지 확인한다.	7. Set Timer	R 3.1	P
8	testStartTimer	타이머 시간 설정후 thread 가 돌며 카운트다운되는지 확인한다.	8. Pause/Restart Timer	R 3.2	P
9	testResetTimer	타이머 시간 설정후 초기화 시 시간이 0 으로 설정되는지 확인한다.	9. Reset Timer	R 3.3	P

10	testRingTimer	타이머의 카운트 다운이 끝난후 buzzer 의 알림이 울리는지 확인한다.	10. Ring Timer	R 3.4	F
11-1	activeStopwatchTest	스톱워치 상태를 활성화 하였는지 확인한다.	11. Start Stopwatch	R 4.1	P
11-2	createTickmanagementTest	Tickmanagement 를 잘 생성하였는지 확인한다.	11. Start Stopwatch	R 4.1	P
11-3	loadTimeTest	cs 단위로 시간이 1 씩 증가하는 것을 확인한다.	11. Start Stopwatch	R 4.1	P
12-1	deleteTickManagementTest	Tickmanagement 를 삭제하는 지 확인하고, 시간 증가가 멈추는 지 확인한다.	12. Pause Stopwatch	R 4.2	P
12-2	inactiveStopWatchTest	스톱워치 상태를 비활성화 하였는지 확인한다.	12. Pause Stopwatch	R 4.2	P
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	P

14	getLapTimeTest	돌아가고 있는 스톱워치의 타임과 랩타임의 시간이 일치하는 지 확인한다.	14. Set Lap Time	R 4.4	P
15-1	setCityTest	도시가 설정한 다음도시로 잘 넘어가는지 확인한다.	15. Set City	R 5.1	P
15-2	toggleSettingTest	세팅을 켜고 끌수 있는지 확인한다.	15. Set City	R 5.1	P
16	loadTimeTest	현재 시간과 일치하는지 확인한다.	16. Make World Clock	R 5.2	P
17	isSSSR	일출 일몰 시간을 잘 계산했는지 확인한다.	17. Calculate SR/SS	R 6.1	P
18	isSSSR	일출 일몰 시간을 잘 세팅했는지 확인한다	18. Set SR/SS	R 6.2	P
19	testEnterSetting	메뉴 설정모드에 진입했는지 확인한다.	19. Set Function	R 7.1	P
19	testCreateFunction	메뉴 설정 완료후 설정모드를 빠져나오면 모드가 잘 생성되는지 확인한다.	19. Set Function	R 7.1	P
19	testSelectFunction	메뉴설정모드에서 up 버튼으로 모드 간 이동이	19. Set Function	R 7.1	P

		이루어지는지 확인한다.			
19	testActivateFunction	메뉴설정모드에서 down 버튼으로 모드 활성화/비활성화 기능이 잘 작동하는지 확인한다.	19. Set Function	R 7.1	P
20	testChangeMode	아무 화면에서 모드 버튼을 누르면 화면이 순서대로 잘 전환되는지 확인한다.	20. Change Mode	R 7.2	P
21	testTimeout	clock 모드를 제외한 모드에서 15 초이상 버튼동작이 없을 시 clock 모드로 화면이 전환되는지 확인한다.	21. Time Out	R 7.3	F
22	testStopBuzzer	알람/타이머의 buzzer 가 울리고 있을 때 아무 버튼 동작으로 알림이 중지되는지 확인한다.	22. Stop Buzzer	R 7.4	P
23			23. Tick Management	R 7.5	P

5. Activity 2167. Testing Traceability Analysis

Operation in sequence diagram
setTime()
setDate()
setCity()
setTimeExpression()
setAlarmTime()
ringAlarm()
stopAlarm()
toggetAlarm()
setTimer()
controlTimer()
startTimer()
pauseTimer()
ResetTimer()
ringTimer()
startStopwatch()
pauseStopwatch()
resetStopWatch()
getLapTime()
setCity()
serSRSS()
setFunction()
changeFunction()
selectFunction()
changeMode()

Operation in Interaction diagram
toggleSetting()
buttonControl()
blinkSector(Time)
serClock()
checkFormat()
setHr()
setTime()
nextSetting()
setMin()
setSec()
setH1224()
setYear()
setMonth()
setDay()
setDate()
setClock()
setCity()
loadTime()
getCs()
setCs()
CheckFormat()
buttonControl()
toggleAlarmSetting()
blinkDisplay(Time)
setAlarmTime()
nextSetting()
getMin()
setMin()
getWeekday()
setWeekday()
checkAlarmTime()
readCurrentClock()
ringAlarm()
ringBuzz()
buttonControl()
toggleSetting()
setBlinkSector()
setTimer()
getMin()
setMin()
getSec()
setSec()
checkFormat()
startTimer()
creatTickManagement()
loadTime()
getCs()
setCs()
checkFormat()

Method	Class
toggleSetting()	Clock
readCurrentClock()	
readCurrentCity()	
readCurrentCityIndex()	
readCurrentDate()	
buttonControl(int btn)	
calculateGMT(int cityCurrent, int cityNext)	
setClock(boolean up)	
nextSetting()	
toggleSetting()	
loadTime()	
loadTime()	
myTurn()	
toggleSetting()	Alarm
nextSetting()	
setAlarmTime(boolean up)	
ringAlarm(int index)	
toggleAlarm()	
nextAlarm()	
checkAlarmTime()	
myTurn()	
buttonControl(int btn)	
isSetting()	Timer
getIdxSetting()	
getTime()	
myTurn()	
setTimer(int btn)	
nextSetting()	
toggleSetting()	
startTimer()	
loadTime()	
createTimerTickManagement()	
deleteTimerTickManagement()	
ringBuzz()	
buttonControl(int btn)	

getGeo()	StopWatch
createTickmanagement()	
loadTime()	
deleteTickManagement()	
resetTime()	
inactiveStopWatch()	
getLapTime()	
buttonControl(int btn)	
setCity(boolean up)	WorldClock
toggleSetting()	
loadTime()	
myTurn()	
buttonControl(int btn)	
setSSSR(int btn)	Geo
calculateSRSS(boolean isSSSR)	
myTurn()	
buttonControl(int btn)	
isSetting()	Function Selector
getIdxSetting()	
getTempNumOfModes()	
getTickCount()	
getCurrentMode()	
myTurn()	
getTimer()	
getClock()	
getStopWatch()	
getAlarm()	
getWorldClock()	
getBuzzer()	
getGeo()	
getDM()	
timeout()	
createFunction(boolean flag, int index)	
enterSetting()	
selectFunction()	
activateFunction()	
stopBuzzer()	
changeMode()	
buttonControl(int btn)	
pushButton(int btn)	

tick()	TickManagement
run()	
getBlinkSector()	DisplayManager
setBlinkSector	
setDate(Date date)	
setTime(Time time)	
setCity(String city)	
setAMPM(boolean isitAM)	
set1224H(boolean isit12H)	
setMenu(int menu)	
setAlarmList(boolean[] alarmiList)	
isRing()	Buzzer
ringBuzzer()	
stopBuzzer()	
checkBuzzerTime()	
run()	
checkFormat(boolean isSetting)	Time
calculateWeekday()	Date
checkFormat(boolean isSetting)	
setYear(int x)	
setMonth(int x)	
setDay(int x)	
setWeekday(int x)	
getYear()	
getMonth()	
getDay()	
getWeekday()	

System Test
calculateGMTTest
setClockTest
nextSettingTest
toggleSettingTest
loadTimeTest
toggleSettingTest
nextSettingTest
setAlarmTimeTest
ringAlarmTest
toggleAlarmTest
nextAlarmTest
testToggleSetting
testNextSetting
testSetTimer
testStartTimer
testMyTurn
testRingTimer
activeStopwatchTest
createTickmanagementTest
loadTimeTest
deleteTickManagementTest
inactiveStopWatchTest
getLapTimeTest
setCityTest
toggleSettingTest
loadTimeTest
isSSSRTest
testEnterSetting
testCreateFunction
testSelectFunction
testActivateFunction
testStopBuzzer
testChangeMode
testTimeout