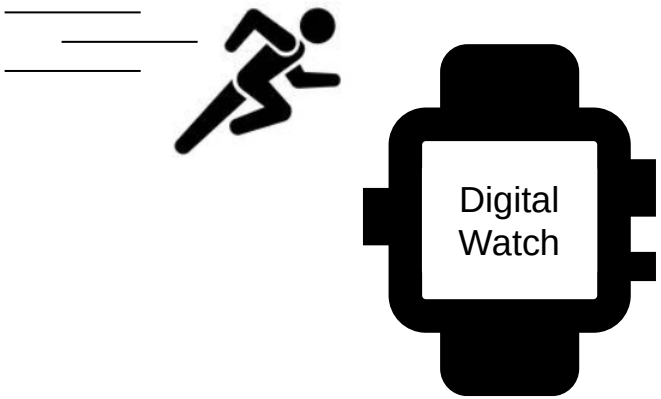


살빼 시계

Presentation 4: OOPT STAGE 2050/2060

Team2
박성호, 박동현, 천민수, 조윤직, 안찬우

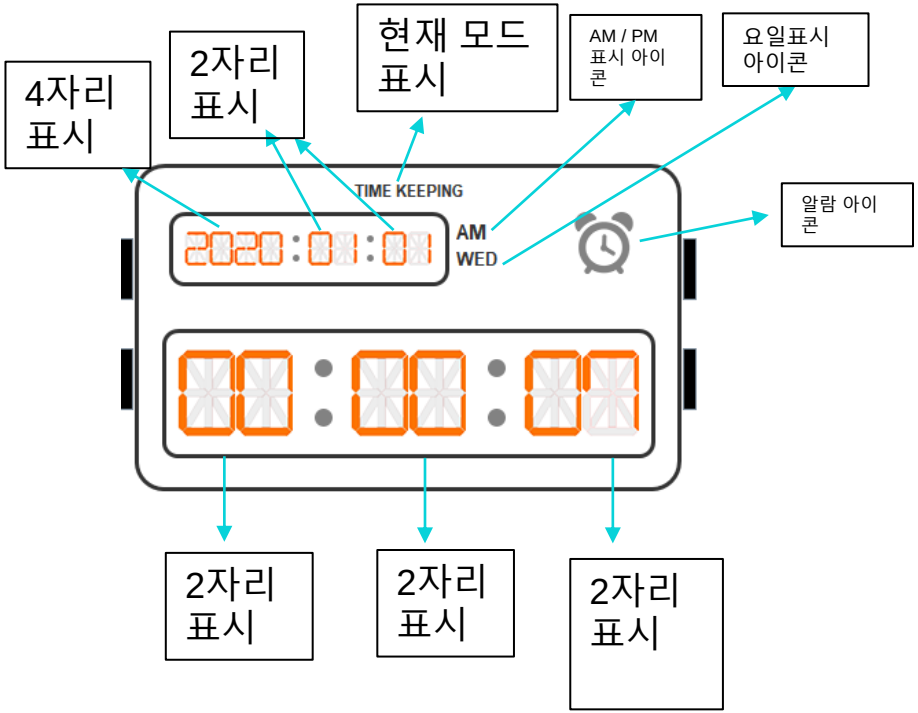
Index

: Construction Phase

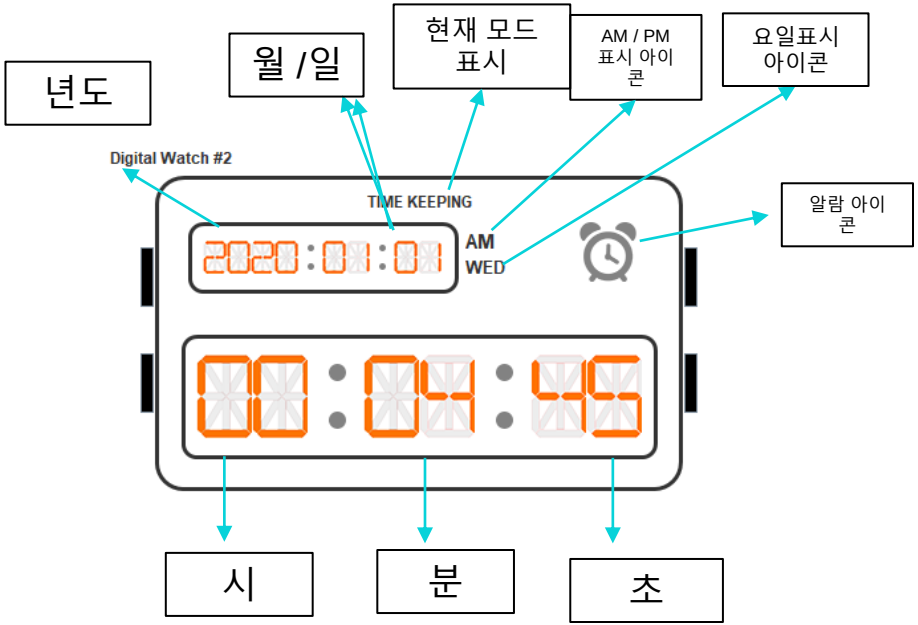
Design & Construct

1

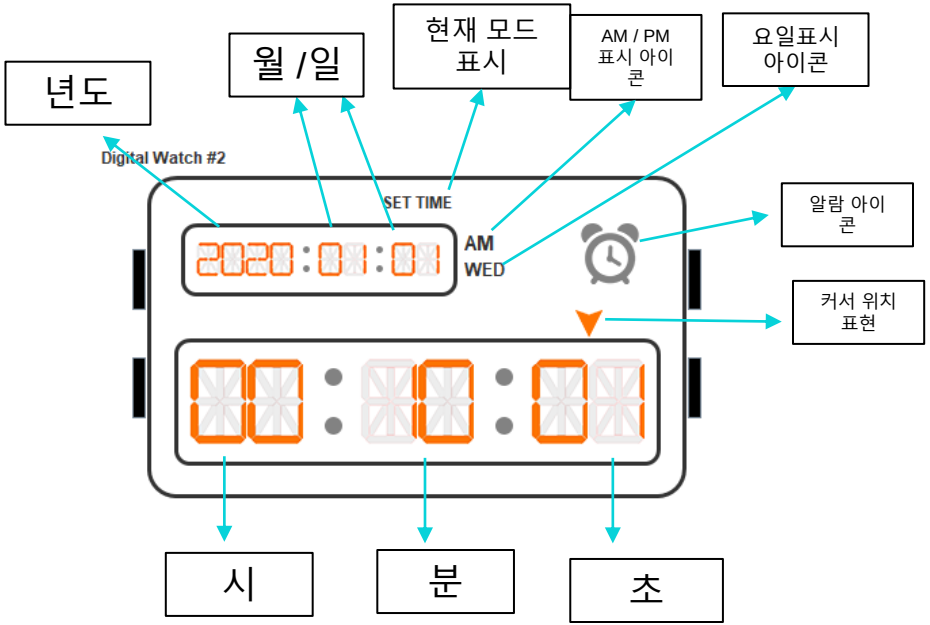
Define Real Use Case: Overall UI



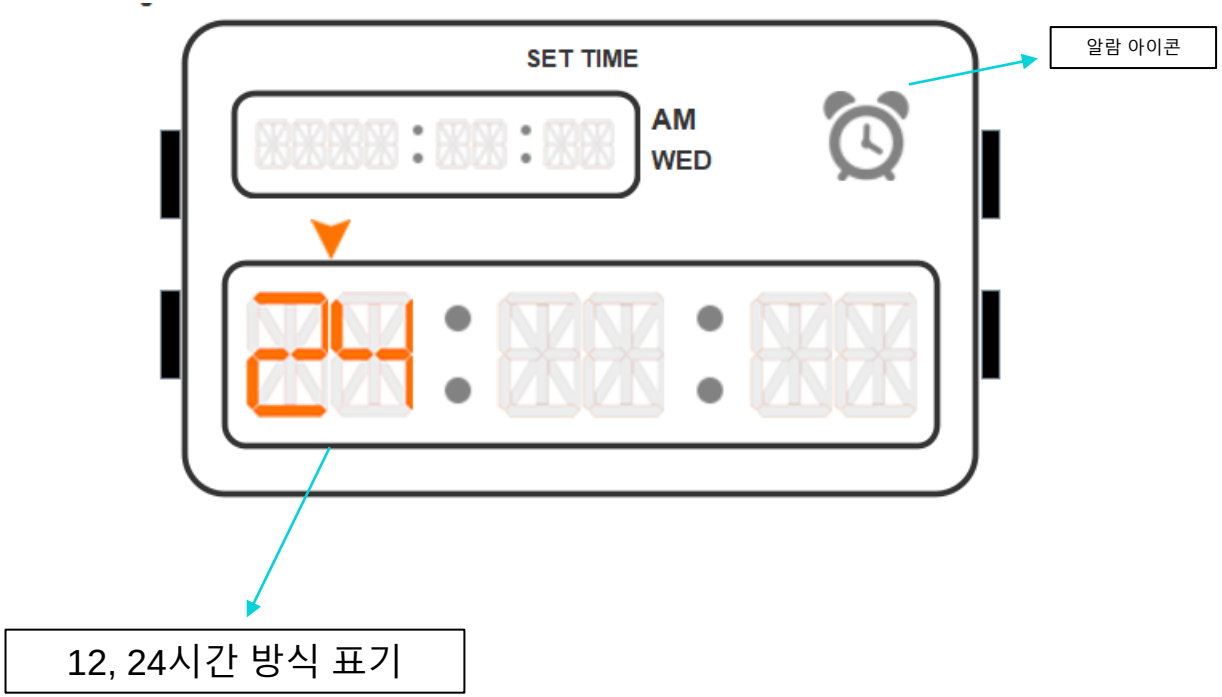
TimeKeeping



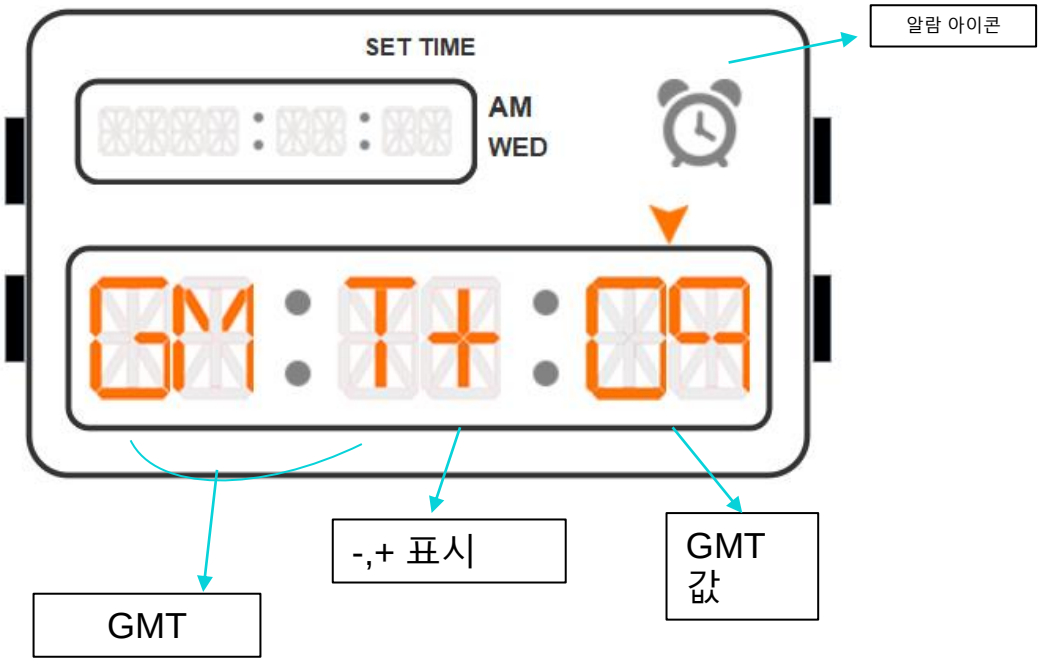
Set Time


년도
월 /일
현재 모드 표시
AM / PM 표시 아이콘
요일표시 아이콘
Digital Watch #2
SET TIME
AM
WED
알람 아이콘
커서 위치 표현
시
분
초

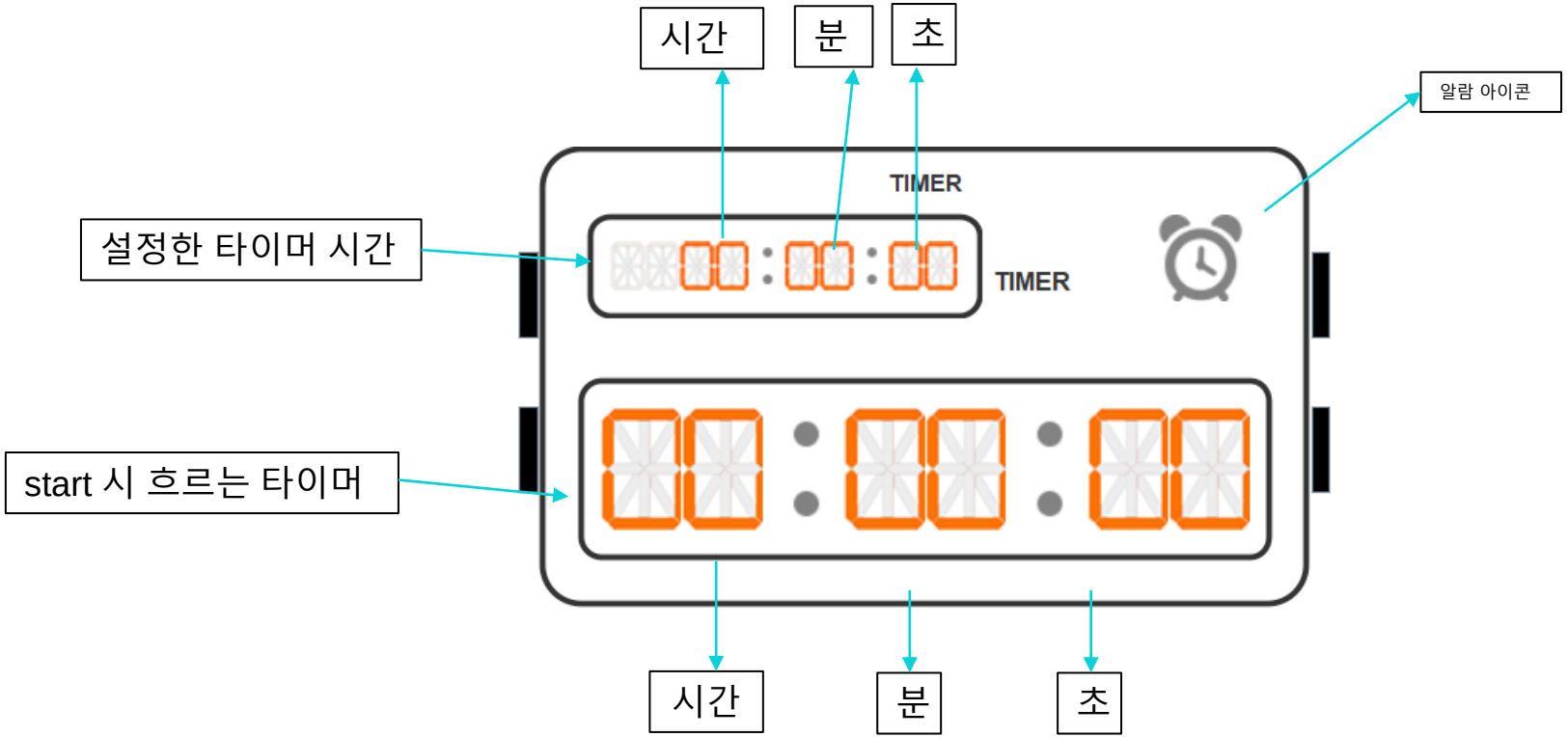
Set Time – 12/24H



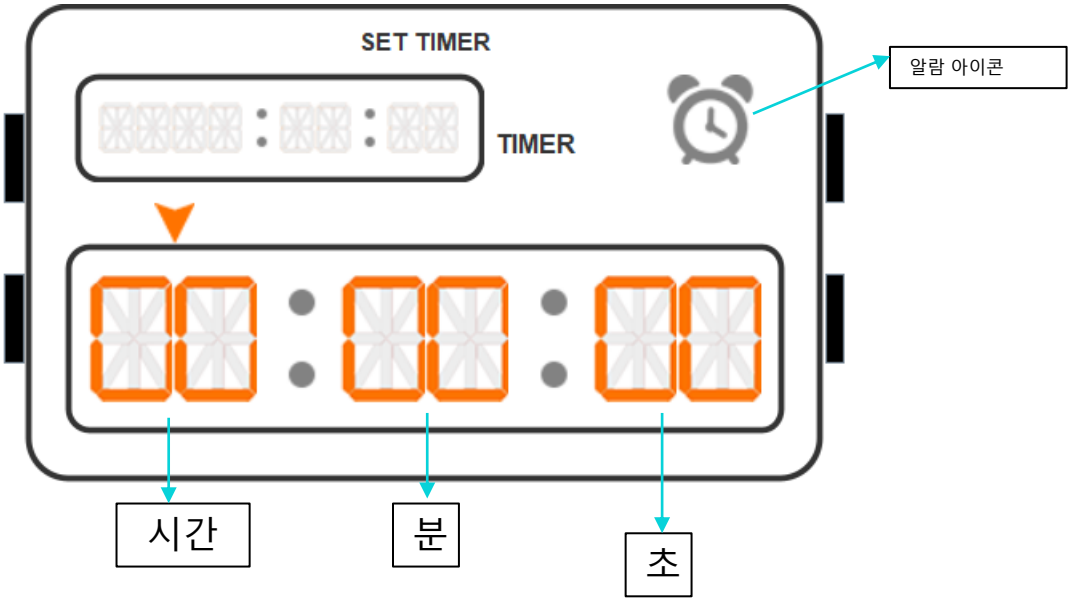
Set Time - GMT



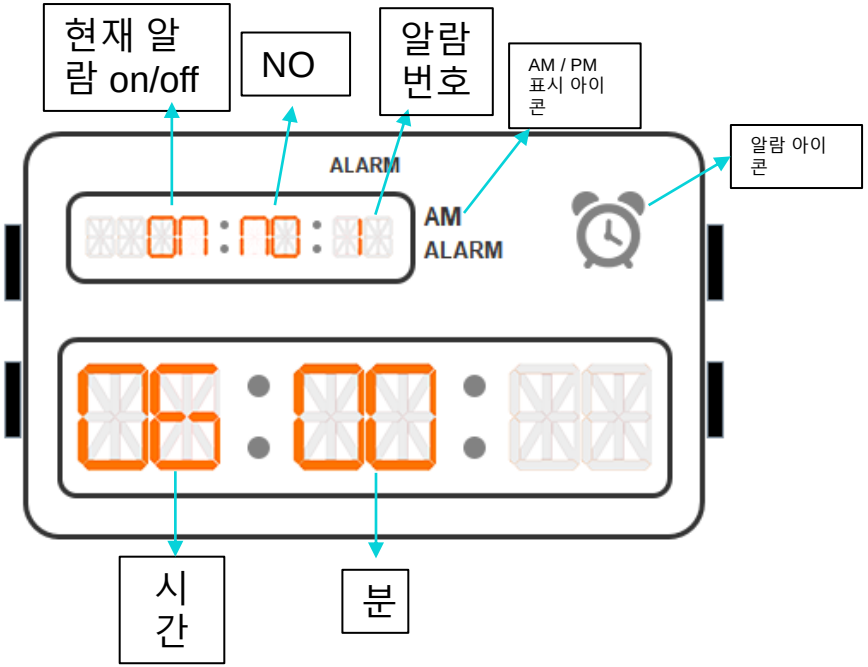
Show Timer



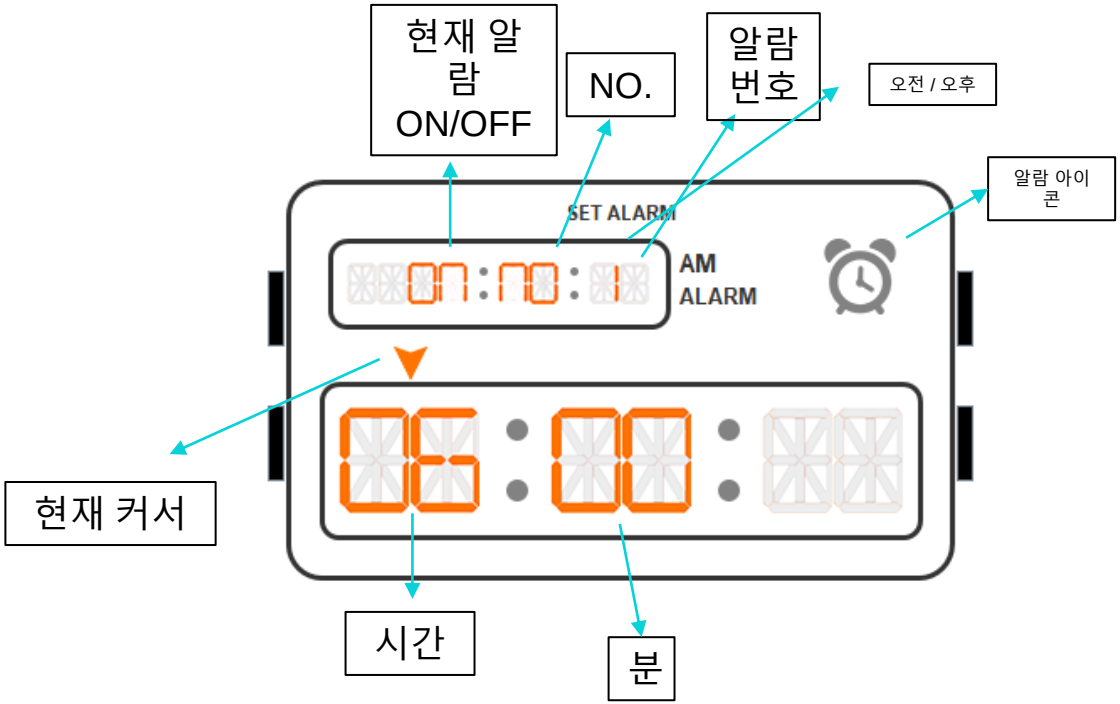
Set Timer



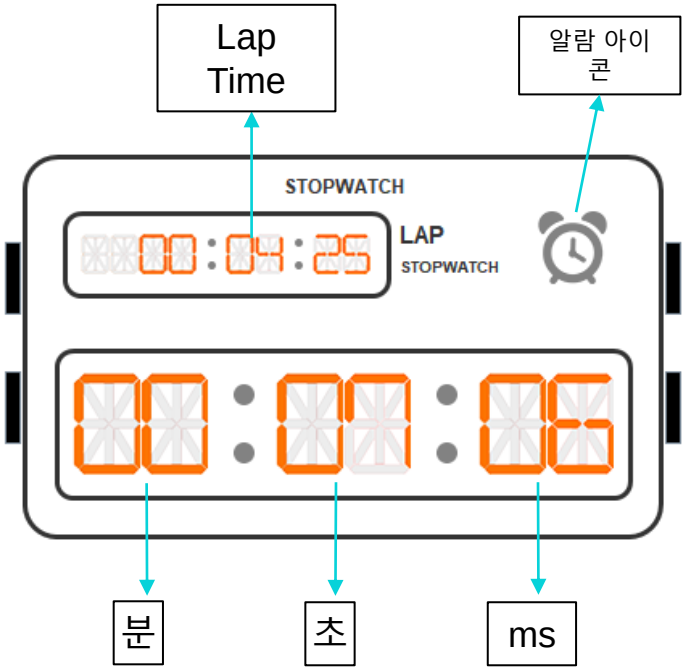
Show Alarm



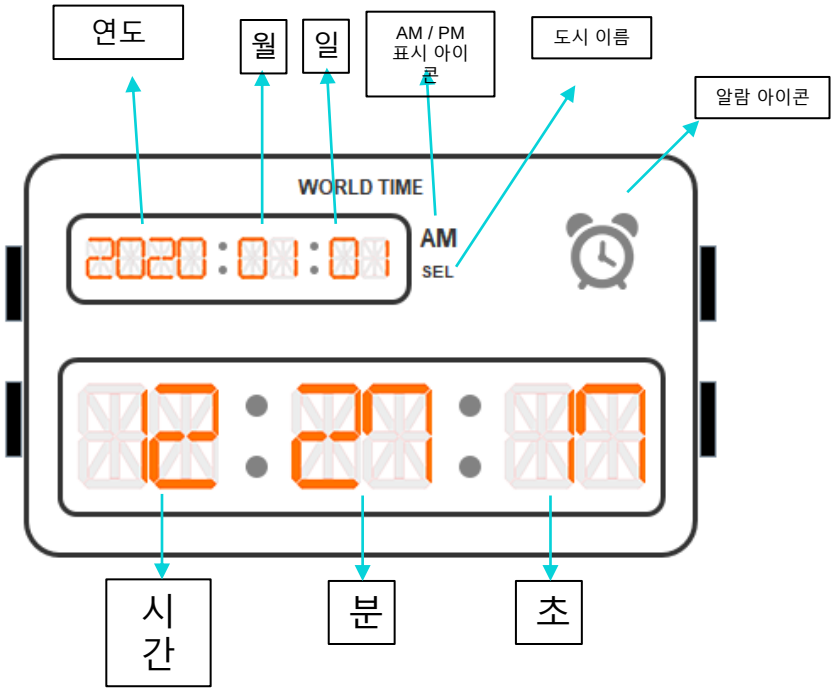
Set Alarm



Stopwatch

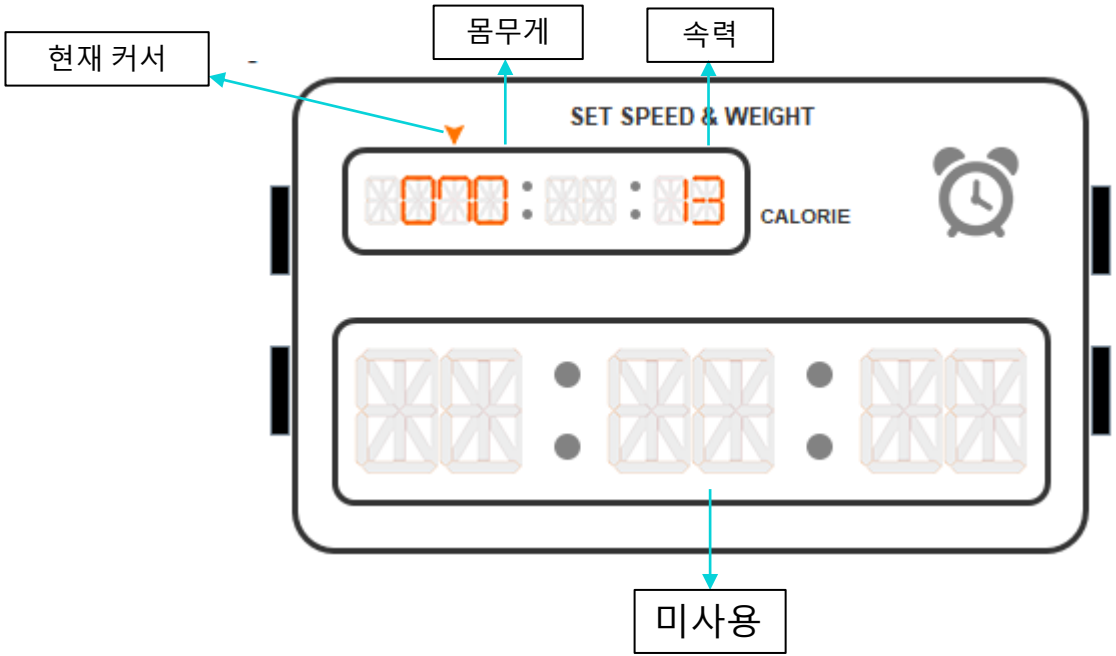


World Time


연도
월
일
AM / PM
표시 아이
콘
도시 이름
알람 아이콘
WORLD TIME
2020:01:01
AM
SEL
12:20:10
시
간
분
초

Calorie Check

Set Speed and Weight



Set Mode

ALRM : 알람

TIME : 시간

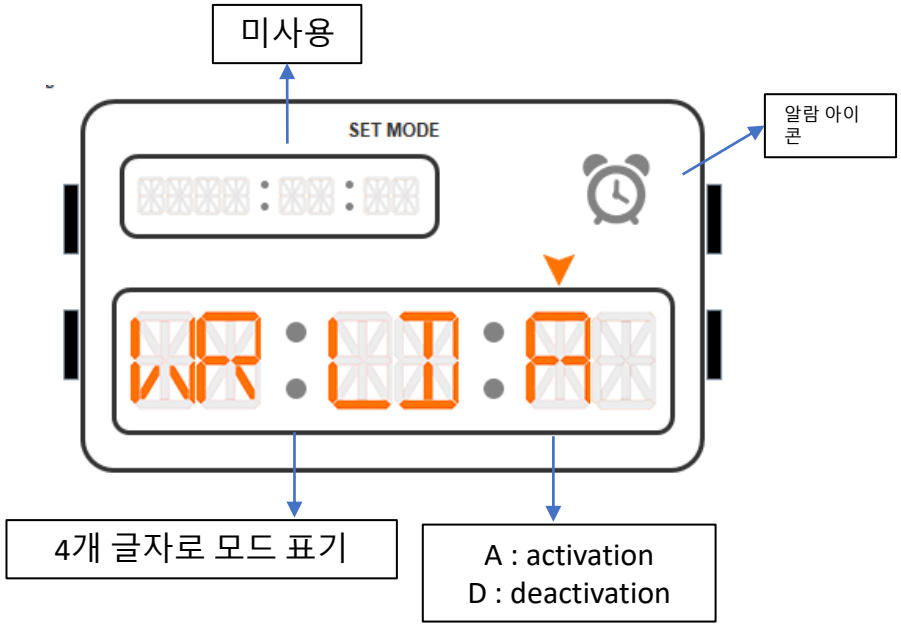
TMER : 타이머

CALO : 칼로리

STPW : 스탑워치

WRLD : 세계시간

A: activate D : deactivate



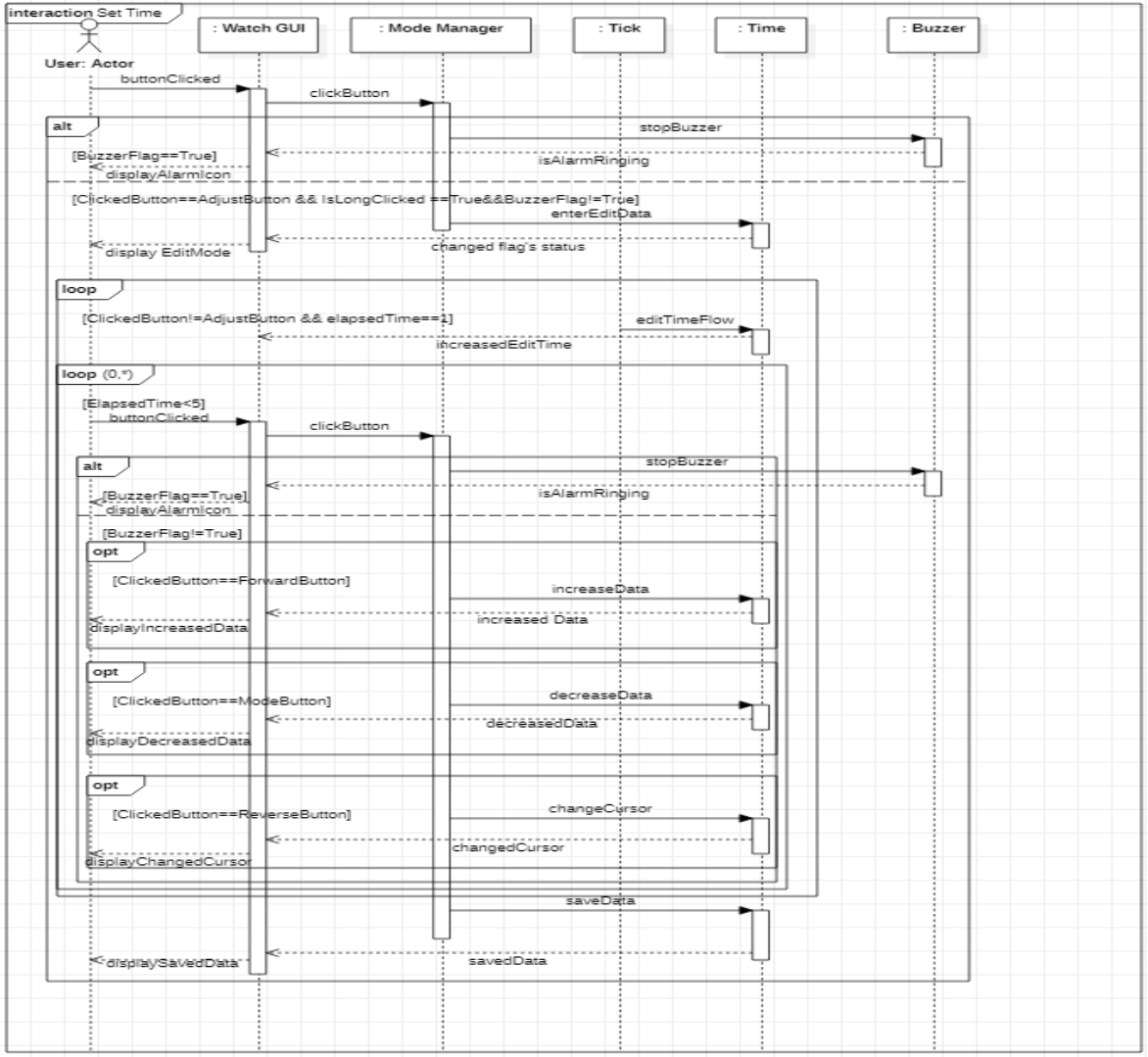
Implement Window

2

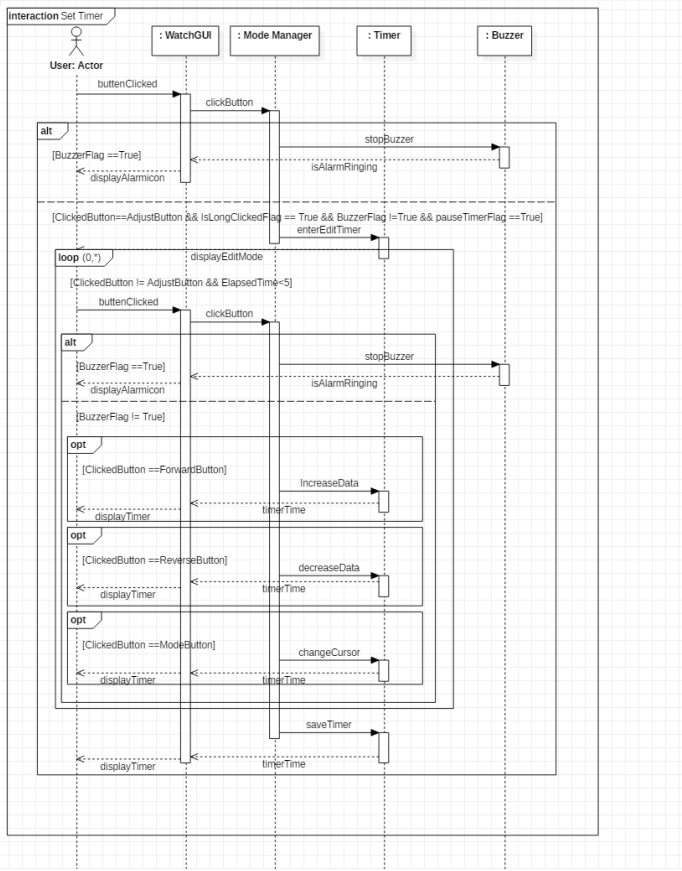
GUI Operation

Name	buttonClicked
Responsibilities	사용자가 시계의 어느 버튼을 짧게 누른다.
Type	GUI
Cross References	All reference
Notes	이 오퍼레이션들을 가지고 있는 버튼의 식별번호와 짧게 혹은 길게 눌렸음에 대한 여부를 ModeManager의 clickedButton을 통해 시스템에 전달
Pre-Conditions	N/A
Post-Conditions	N/A

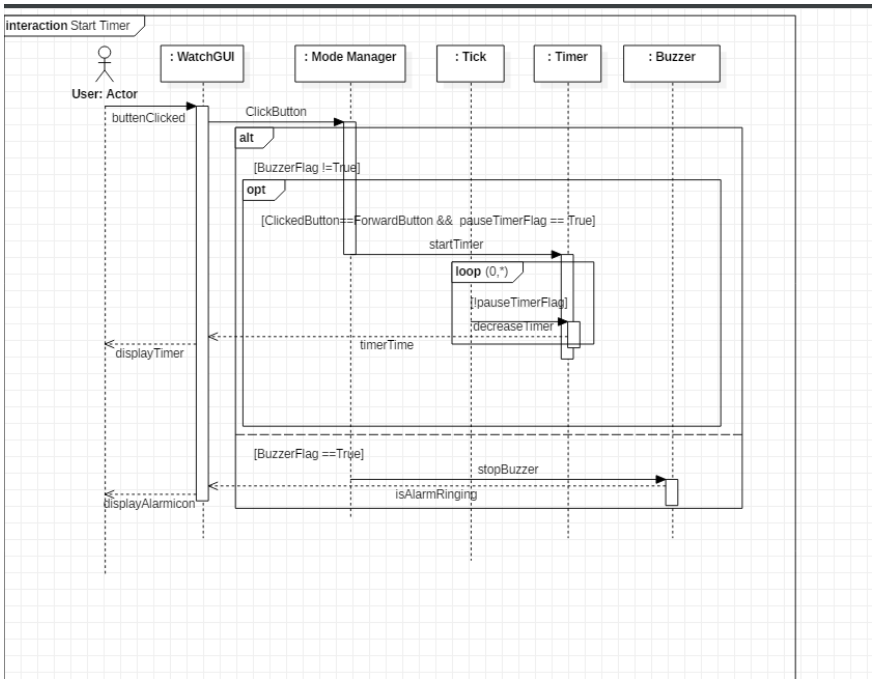
Set Time


interaction Set Time
User: Actor
: Watch GUI
: Mode Manager
: Tick
: Time
: Buzzer
buttonClicked
clickButton
alt
stopBuzzer
[BuzzerFlag==True]
isAlarmRinging
displayAlarmIcon
[ClickedButton==AdjustButton && IsLongClicked ==True&&BuzzerFlag!=True]
enterEditData
display EditMode
changed flag's status
loop
[ClickedButton!=AdjustButton && elapsedTime==1]
editTimeFlow
increasedEditTime
loop (0,*)
[ElapsedTime<5]
buttonClicked
clickButton
alt
stopBuzzer
[BuzzerFlag==True]
isAlarmRinging
displayAlarmIcon
[BuzzerFlag!=True]
opt
[ClickedButton==ForwardButton]
increaseData
displayIncreasedData
increased Data
opt
[ClickedButton==ModeButton]
decreaseData
displayDecreasedData
decreasedData
opt
[ClickedButton==ReverseButton]
changeCursor
displayChangedCursor
changedCursor
saveData
displaySavedData
savedData

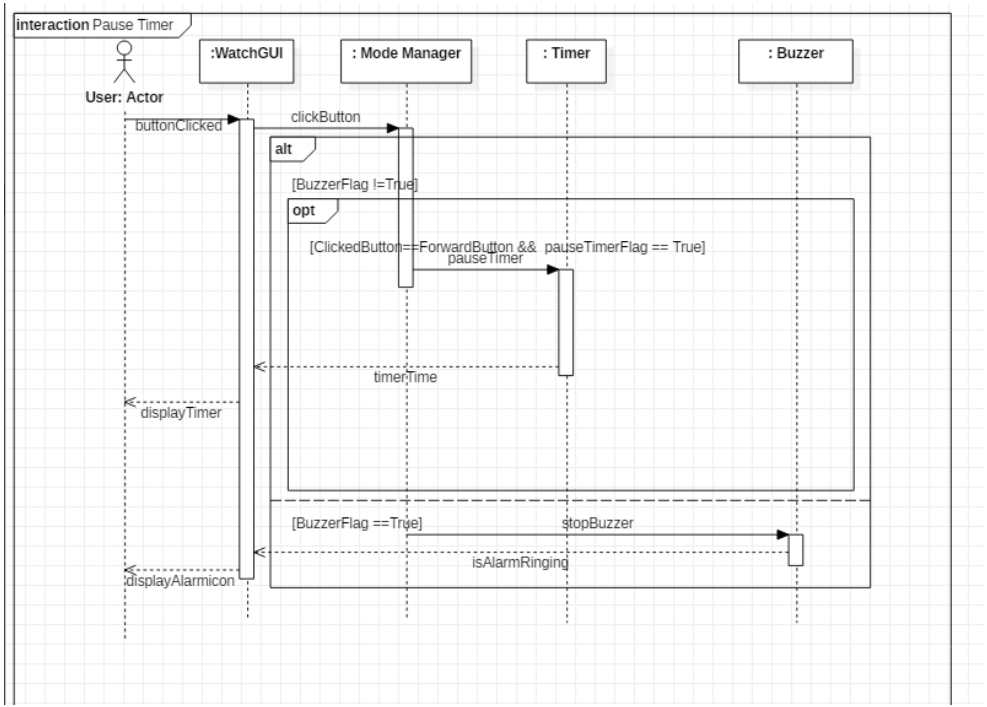
Set Timer


interaction Set Timer
User: Actor
: WatchGUI
: Mode Manager
: Timer
: Buzzer
buttenClicked
clickButton
alt
[BuzzerFlag ==True]
stopBuzzer
isAlarmRinging
displayAlarmicon
[ClickedButton==AdjustButton && IsLongClickedFlag == True && BuzzerFlag !=True && pauseTimerFlag ==True]
enterEditTimer
loop (0,*)
displayEditMode
[ClickedButton != AdjustButton && ElapsedTime<5]
buttenClicked
clickButton
alt
[BuzzerFlag ==True]
stopBuzzer
isAlarmRinging
displayAlarmicon
[BuzzerFlag != True]
opt
[ClickedButton ==ForwardButton]
increaseData
displayTimer
timerTime
opt
[ClickedButton ==ReverseButton]
decreaseData
displayTimer
timerTime
opt
[ClickedButton ==ModeButton]
changeCursor
displayTimer
timerTime
saveTimer
displayTimer
timerTime

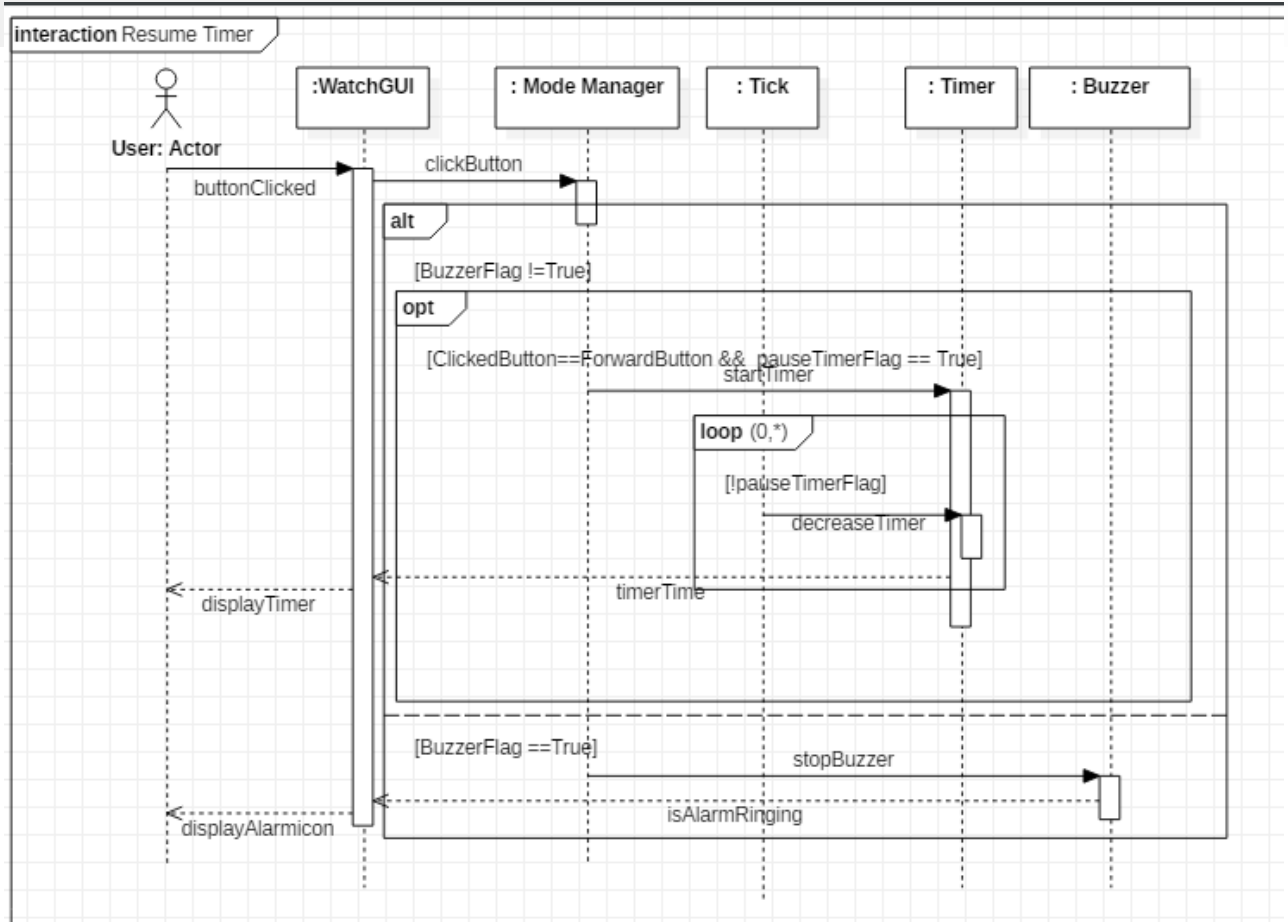
Start Timer


interaction Start Timer
: WatchGUI
: Mode Manager
: Tick
: Timer
: Buzzer
User: Actor
buttenClicked
ClickButton
alt
[BuzzerFlag !=True]
opt
[ClickedButton==ForwardButton && pauseTimerFlag == True]
startTimer
loop (0,*)
[!pauseTimerFlag]
decreaseTimer
timerTime
displayTimer
[BuzzerFlag ==Trje]
stopBuzzer
isAlarmRinging
displayAlarmicon

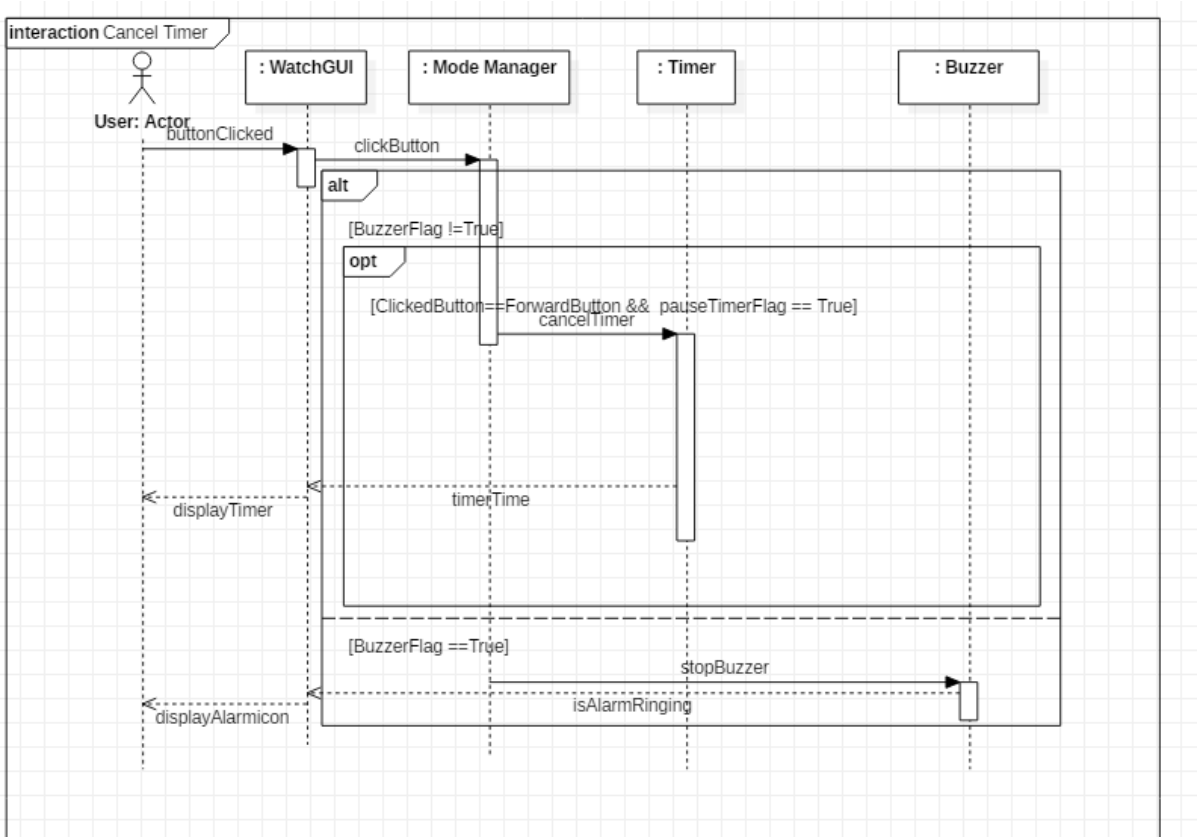
Pause Timer


interaction Pause Timer
User: Actor
:WatchGUI
: Mode Manager
: Timer
: Buzzer
buttonClicked
clickButton
alt
[BuzzerFlag !=True]
opt
[ClickedButton==ForwardButton && pauseTimerFlag == True]
pauseTimer
timerTime
displayTimer
[BuzzerFlag ==True]
stopBuzzer
isAlarmRinging
displayAlarmicon

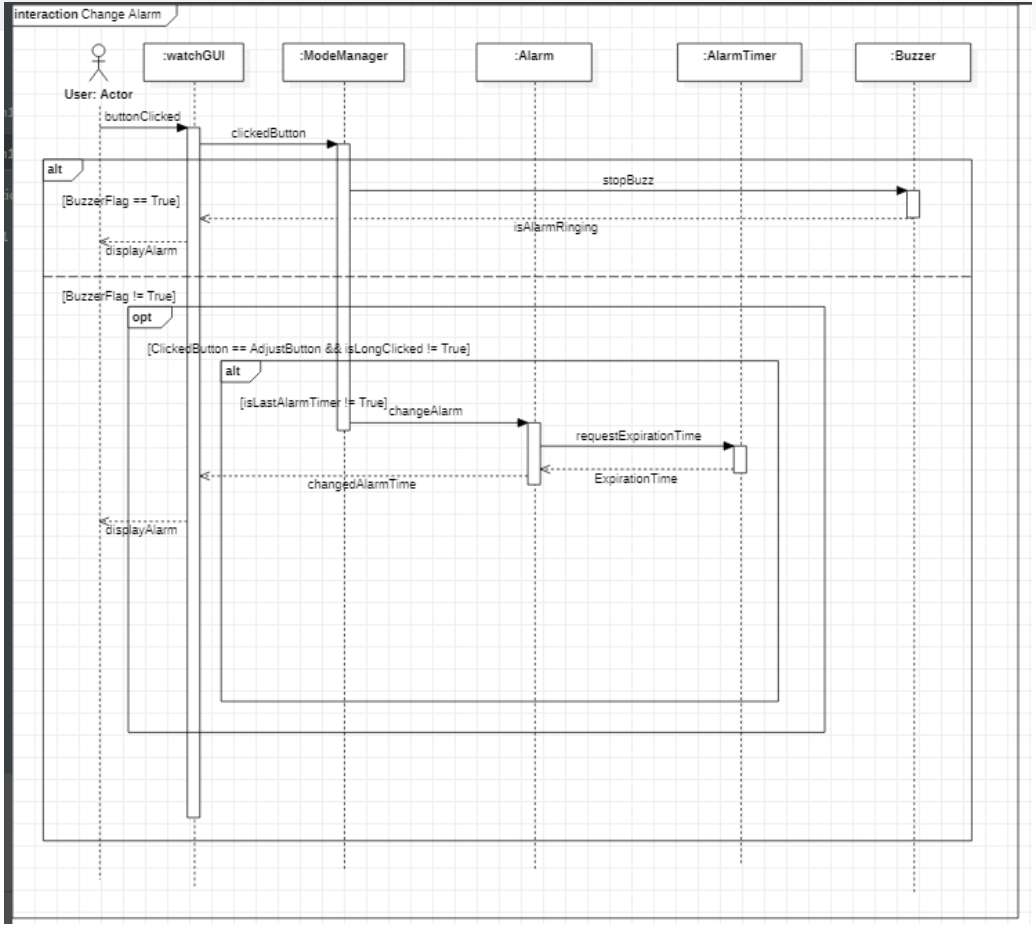
Resume Timer



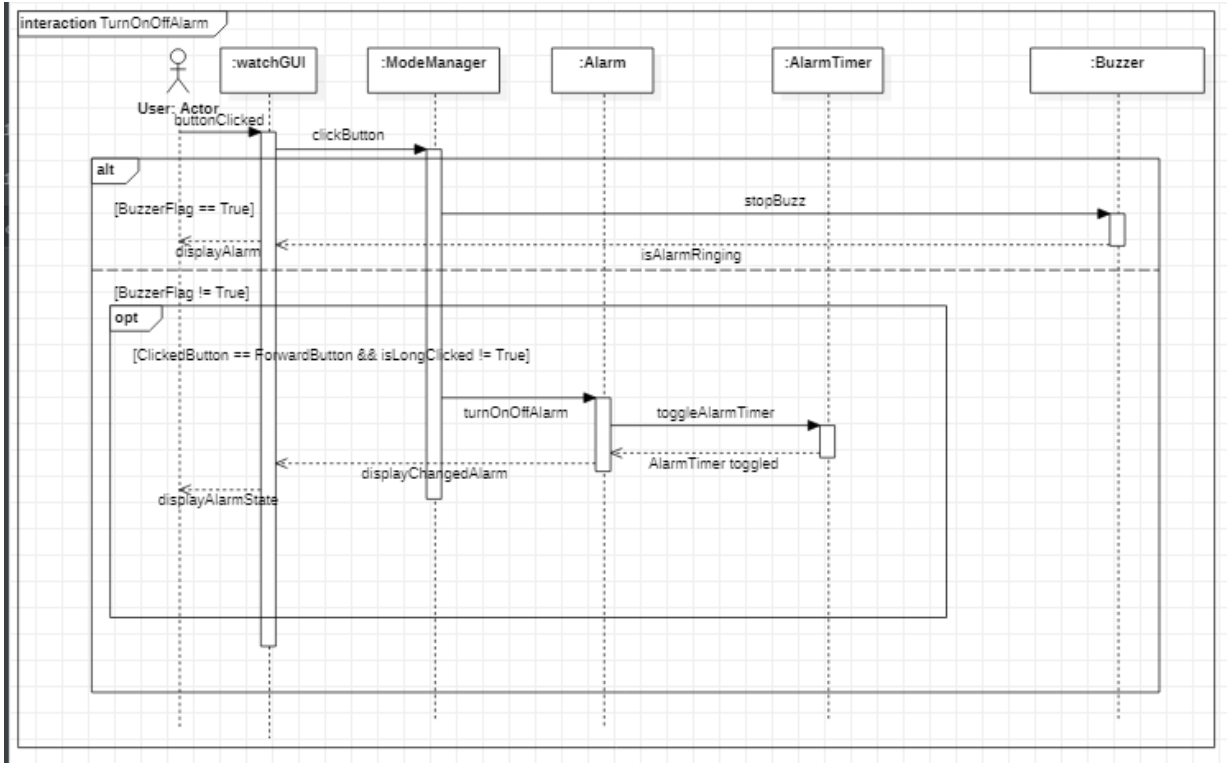
Cancel Timer



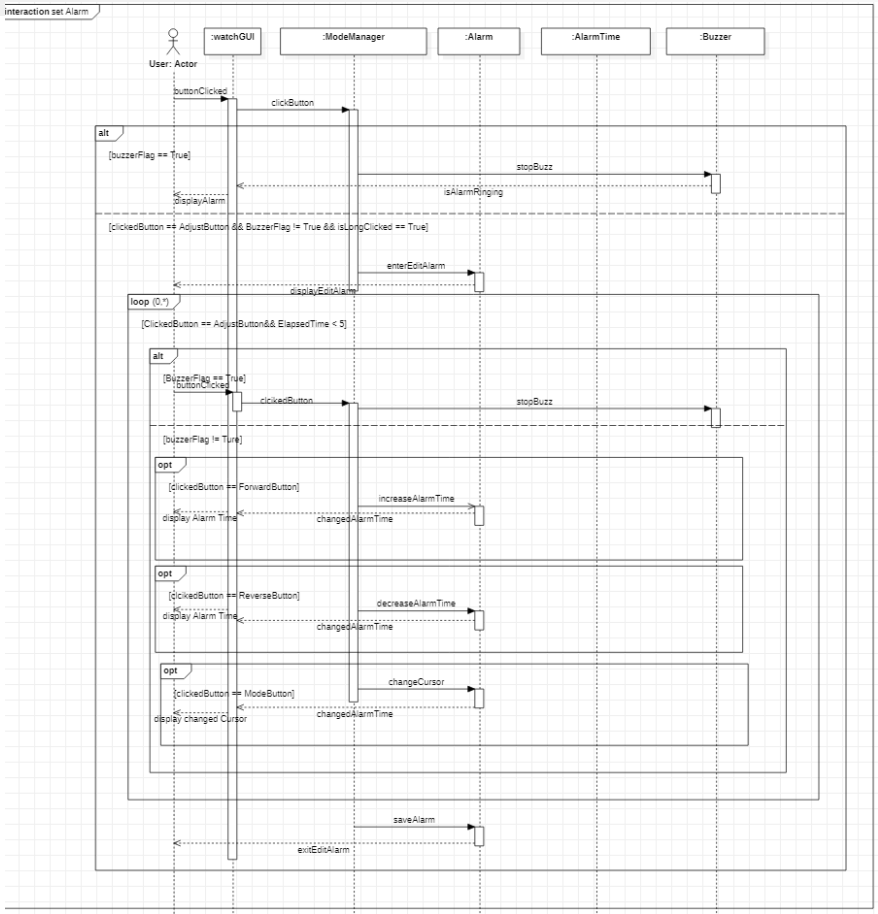
Change Alarm



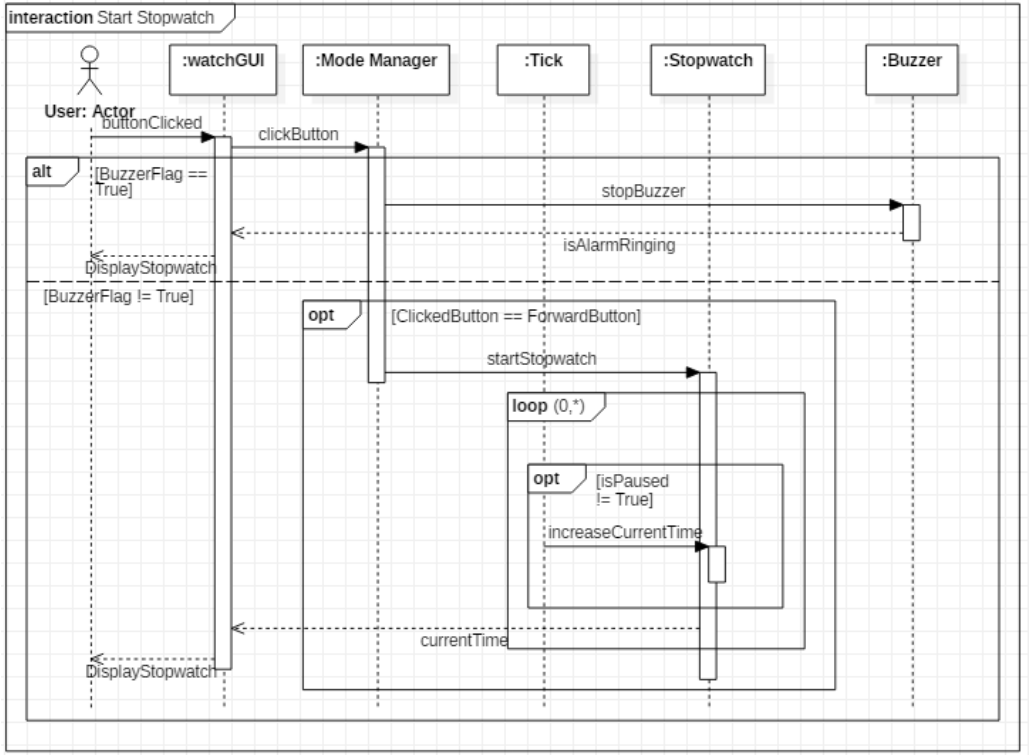
Turn On/Off Alarm



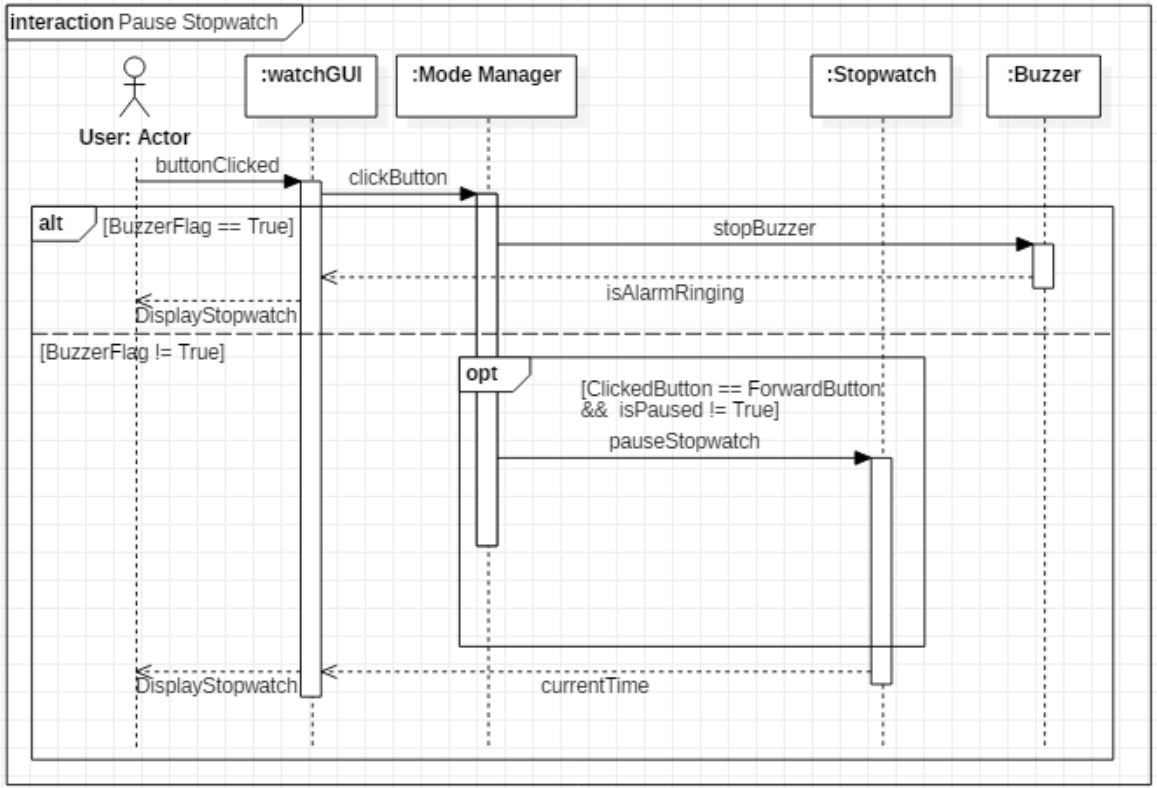
Set Alarm


interaction set Alarm
User: Actor
:watchGUI
:ModeManager
:Alarm
:AlarmTime
:Buzzer
buttonClicked
clickButton
alt
[buzzerFlag == True]
stopBuzz
isAlarmRinging
displayAlarm
[clickedButton == AdjustButton && BuzzerFlag != True && isLongClicked == True]
enterEditAlarm
displayEditAlarm
loop (0,*)
[ClickedButton == AdjustButton&& ElapsedTime < 5]
alt
[BuzzerFlag == True]
buttonClicked
clickedButton
stopBuzz
[buzzerFlag != Ture]
opt
[clickedButton == ForwardButton]
increaseAlarmTime
display Alarm Time
changedAlarmTime
opt
[clickedButton == ReverseButton]
decreaseAlarmTime
display Alarm Time
changedAlarmTime
opt
[clickedButton == ModeButton]
changeCursor
display changed Cursor
changedAlarmTime
saveAlarm
exitEditAlarm

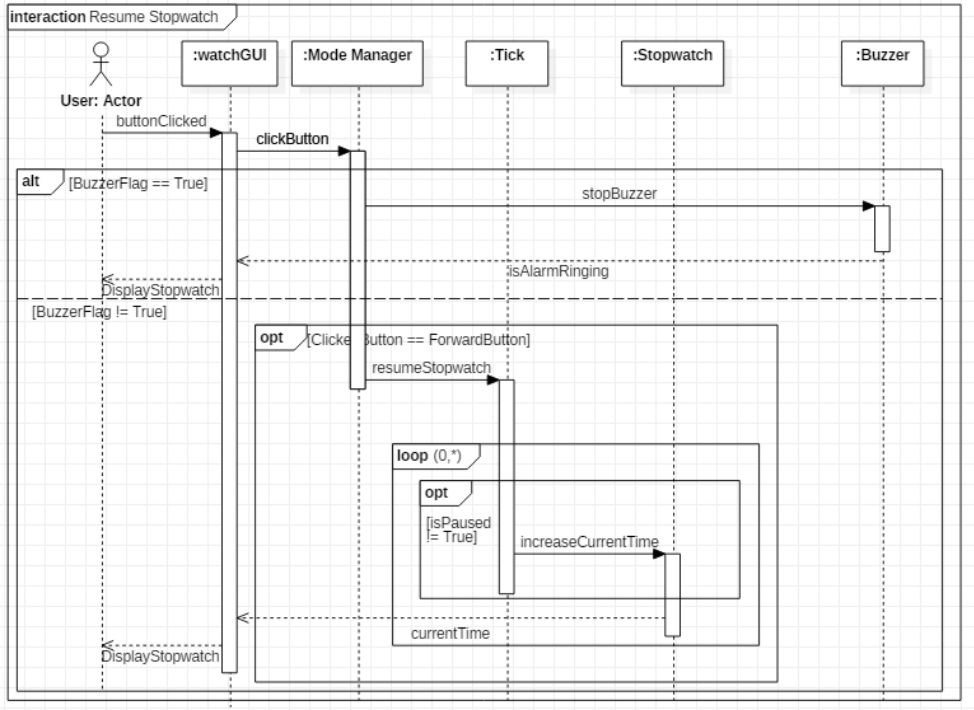
Start Stopwatch



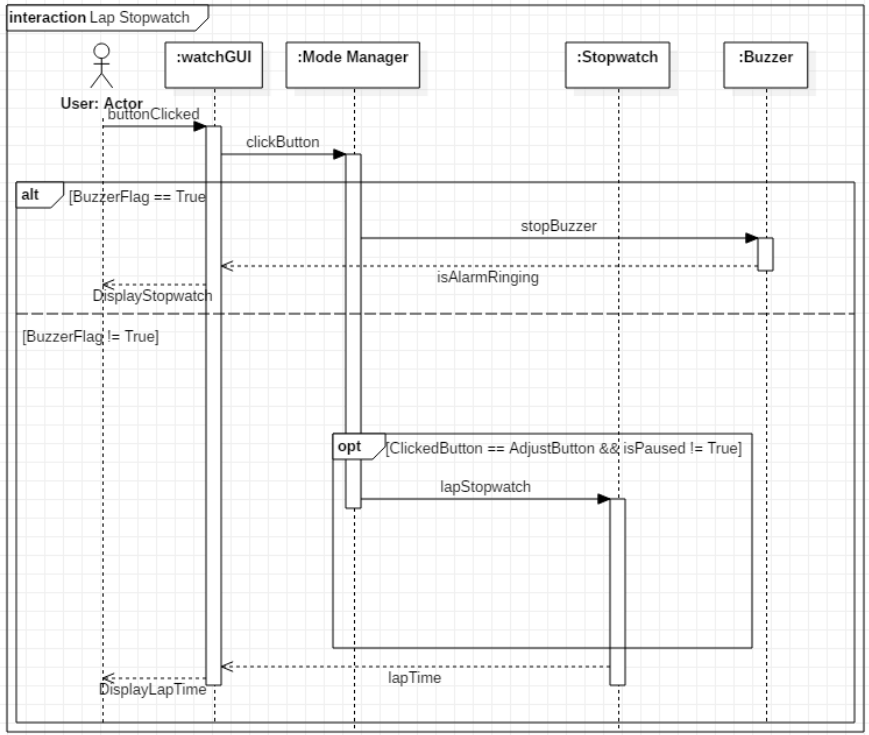
Pause Stopwatch



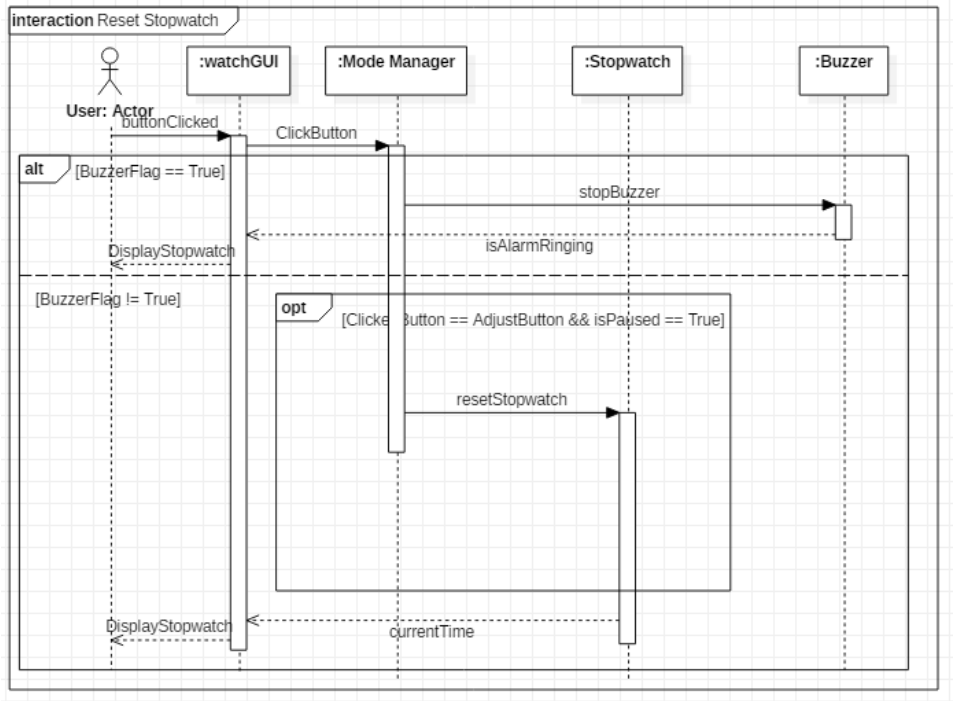
Resume Stopwatch



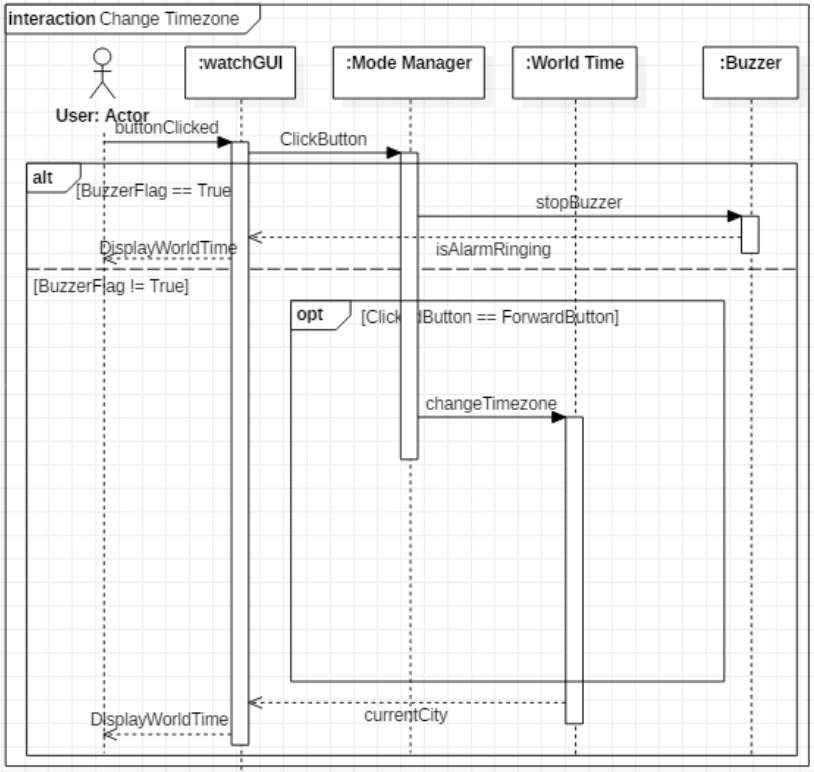
Lap Stopwatch



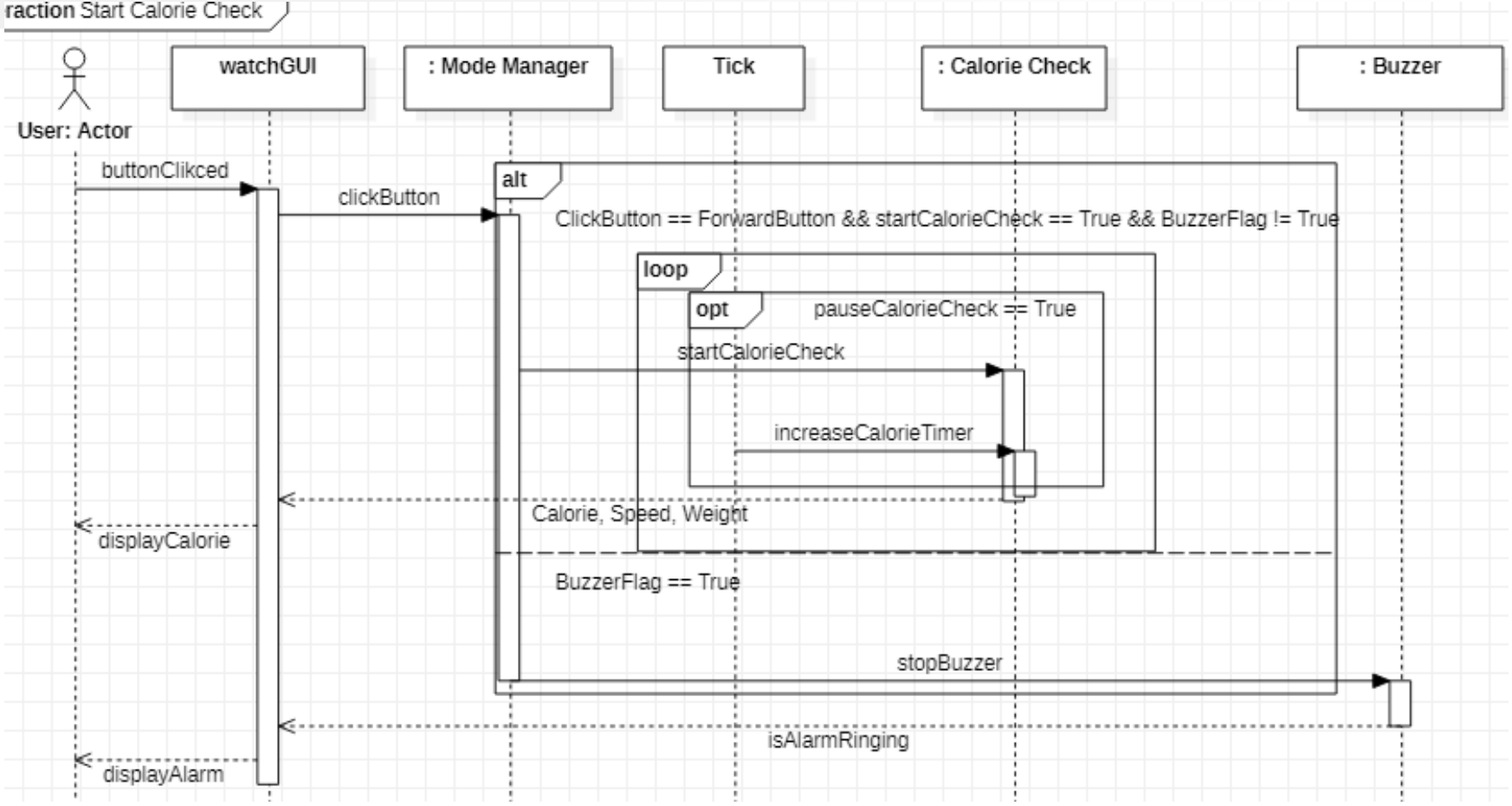
Reset Stopwatch



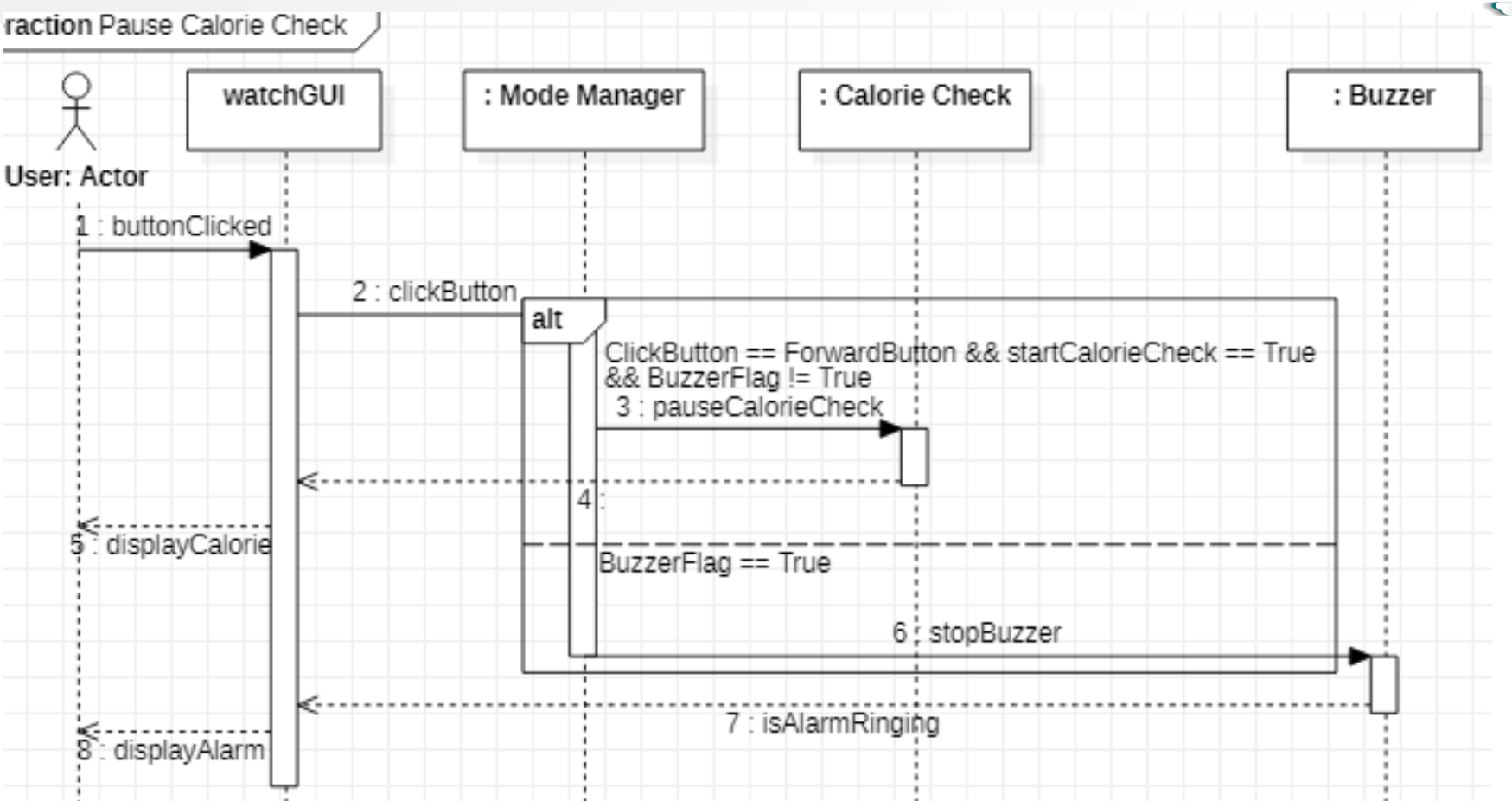
Change Timezone



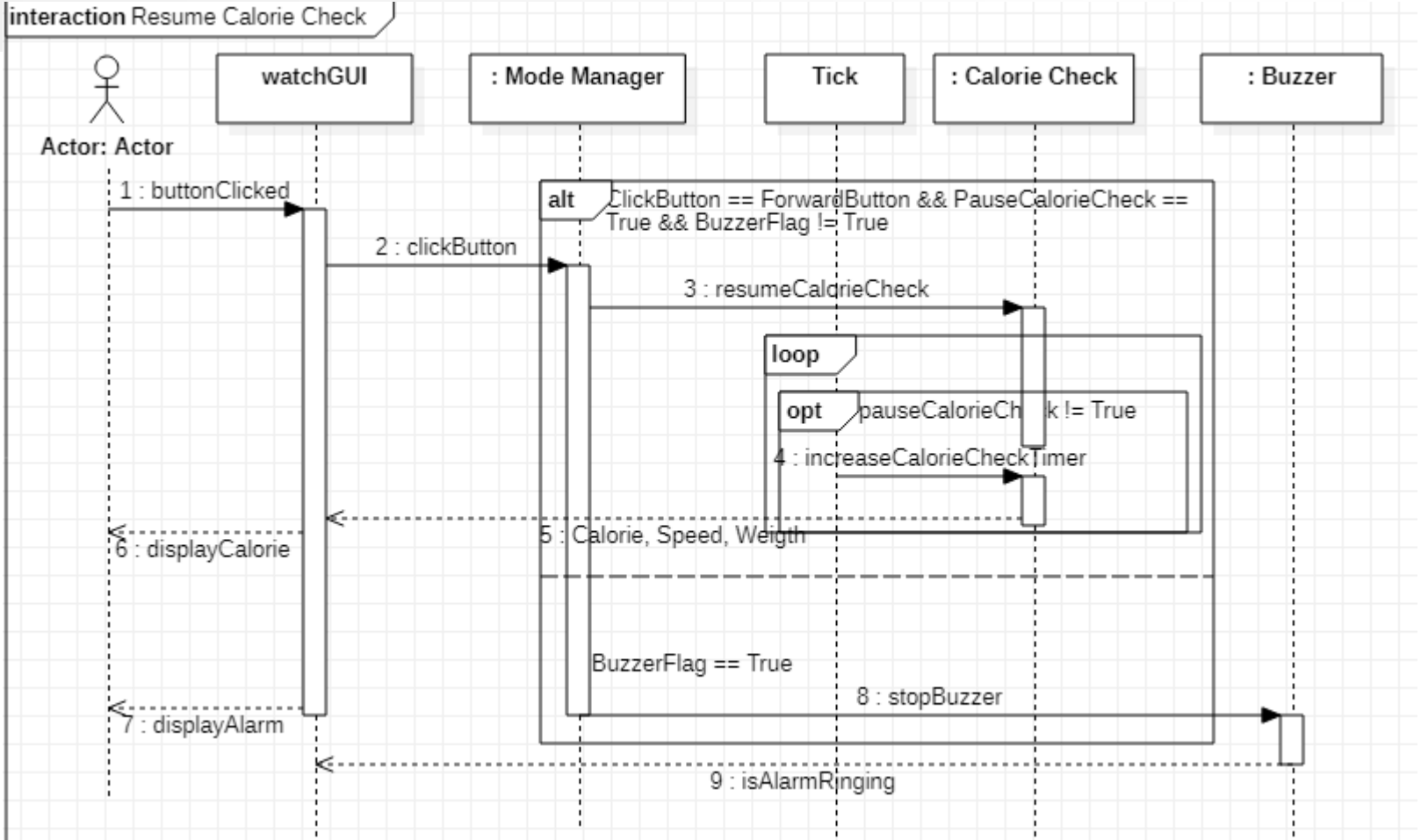
Start Calorie Check



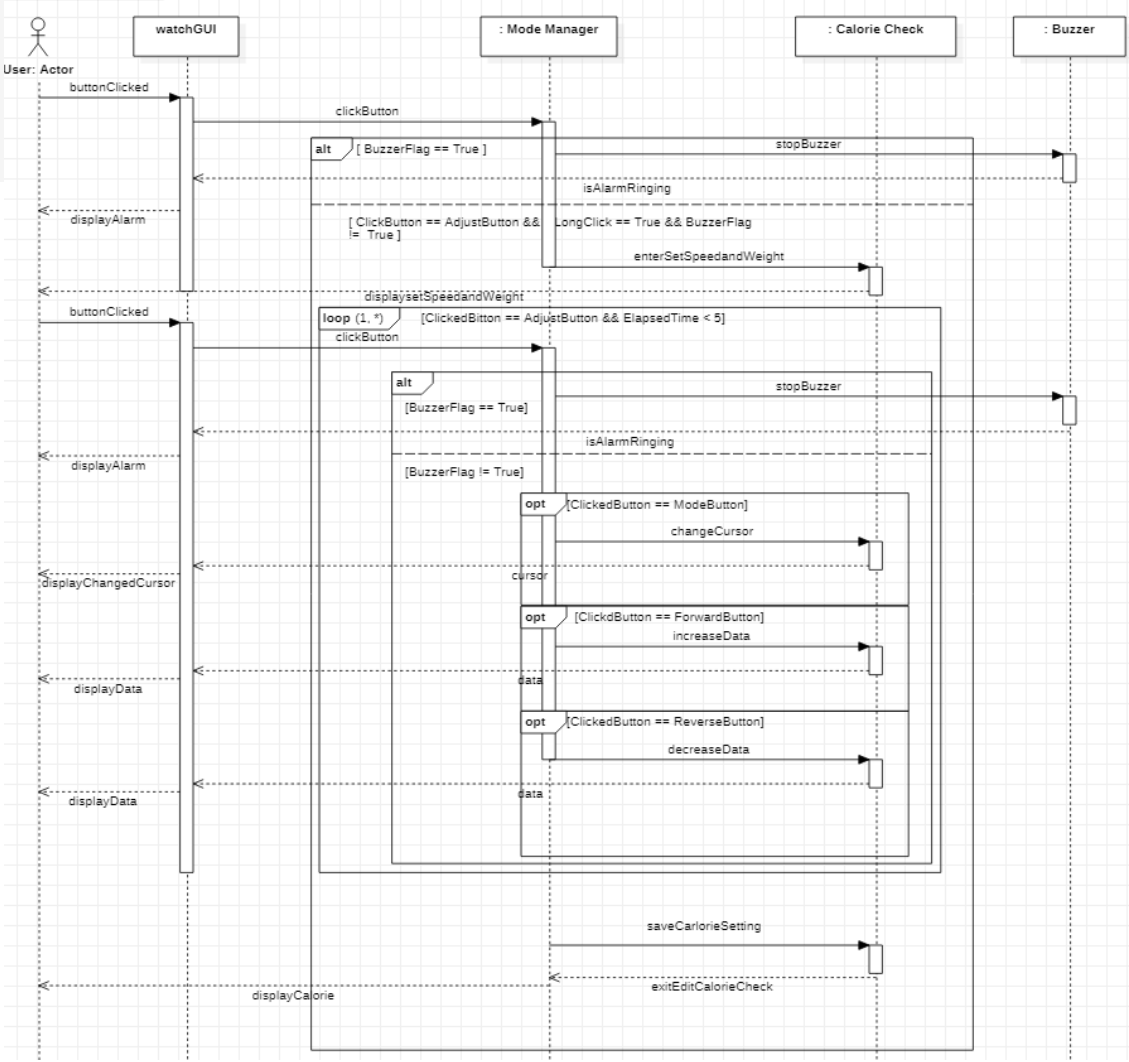
Pause Calorie Check



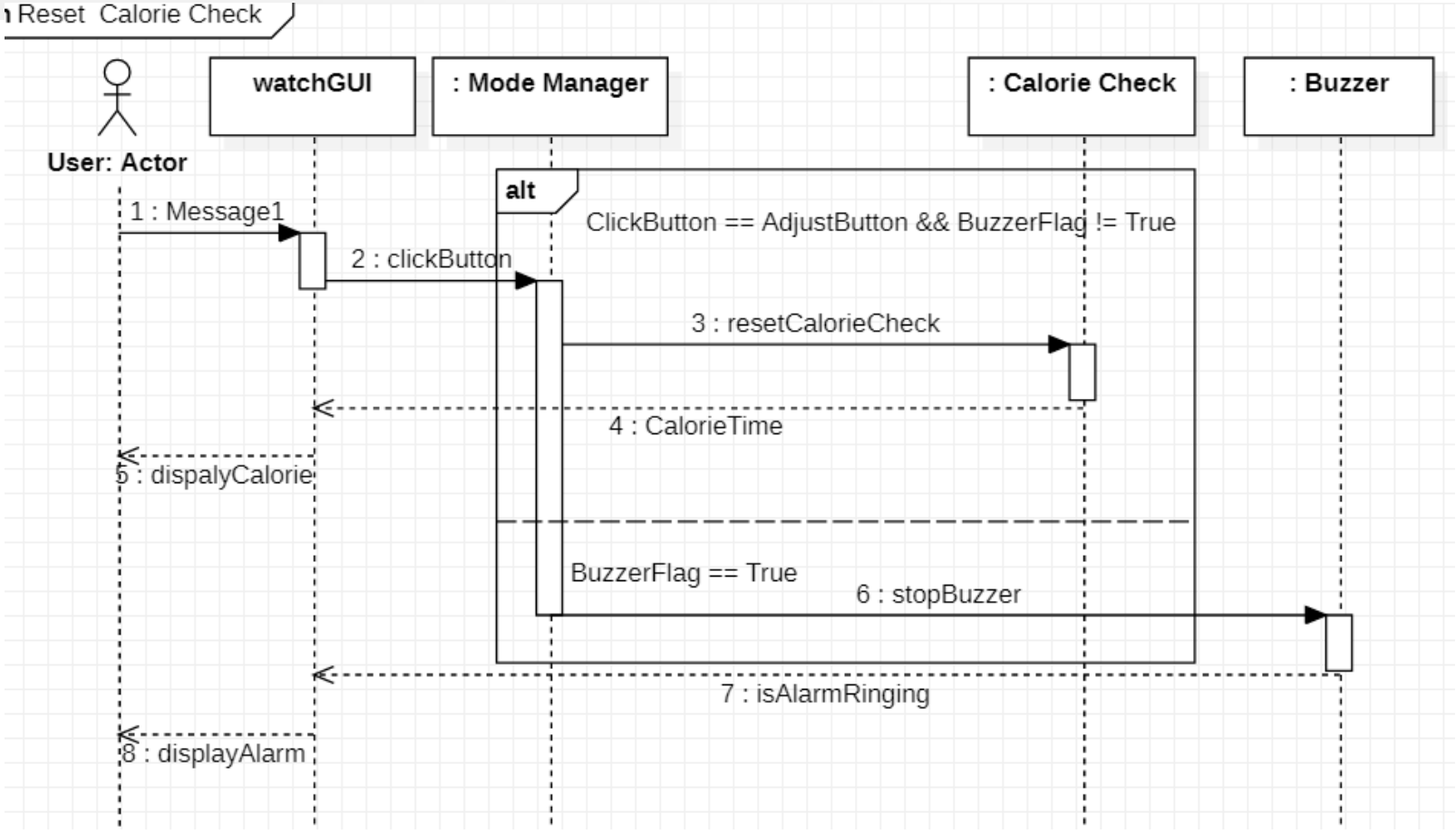
Resume Calorie Check



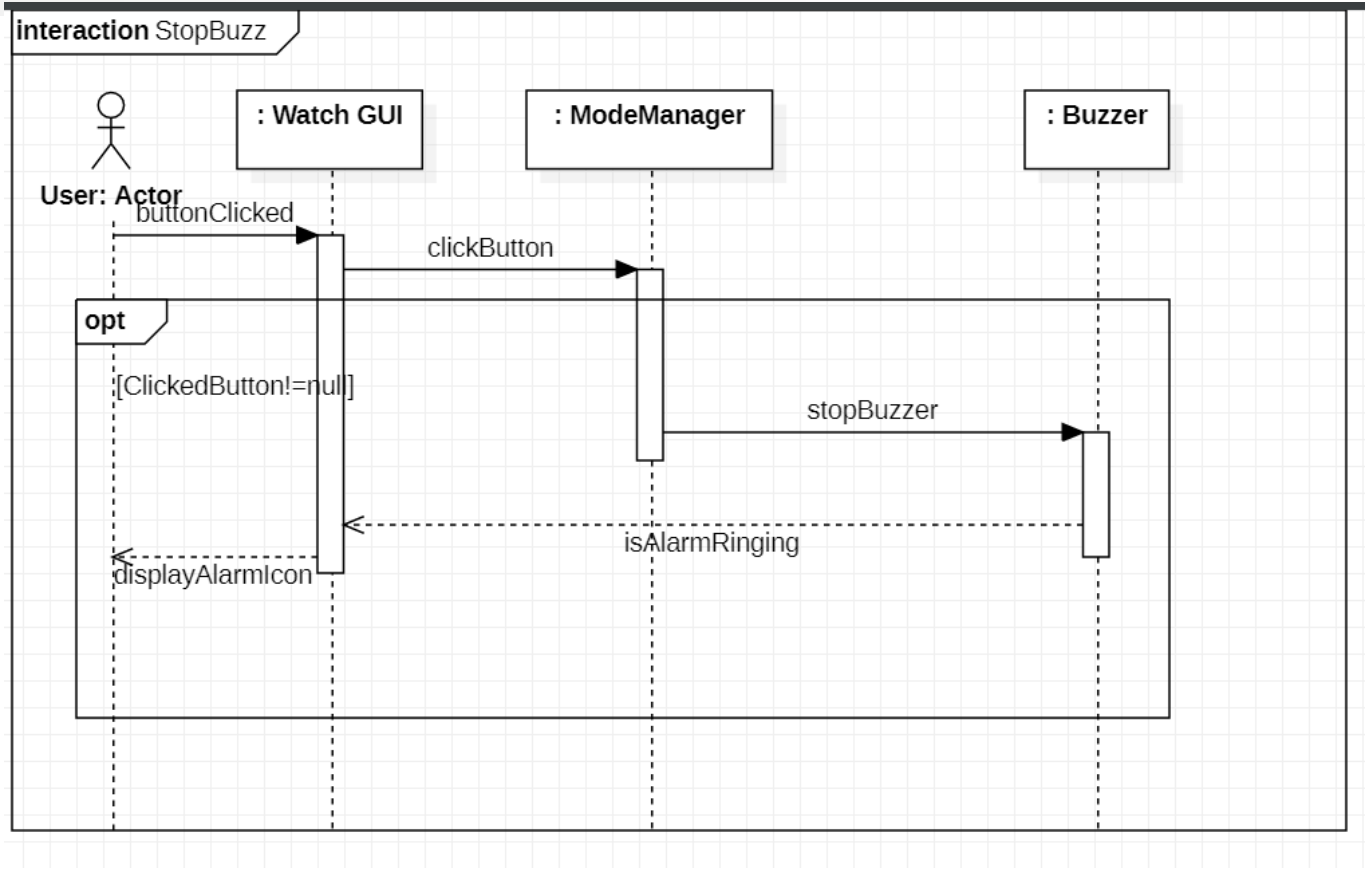
Set Speed and weight



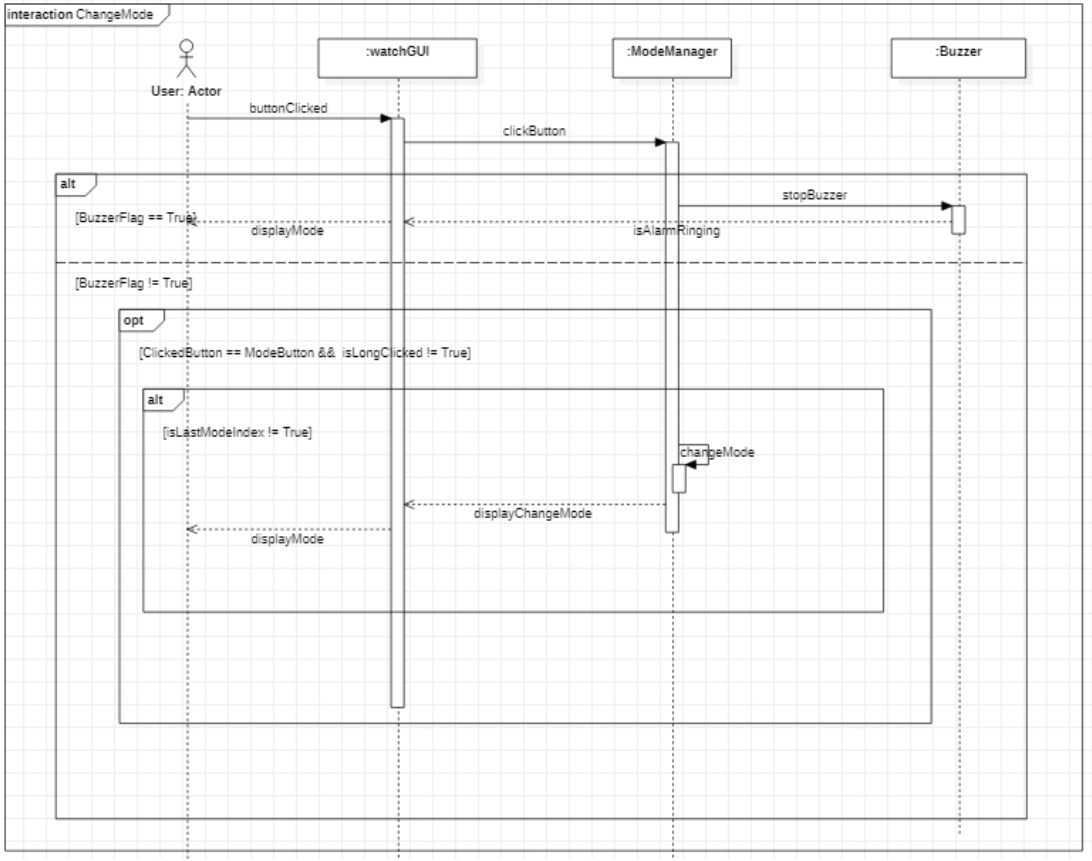
Reset Calorie Check



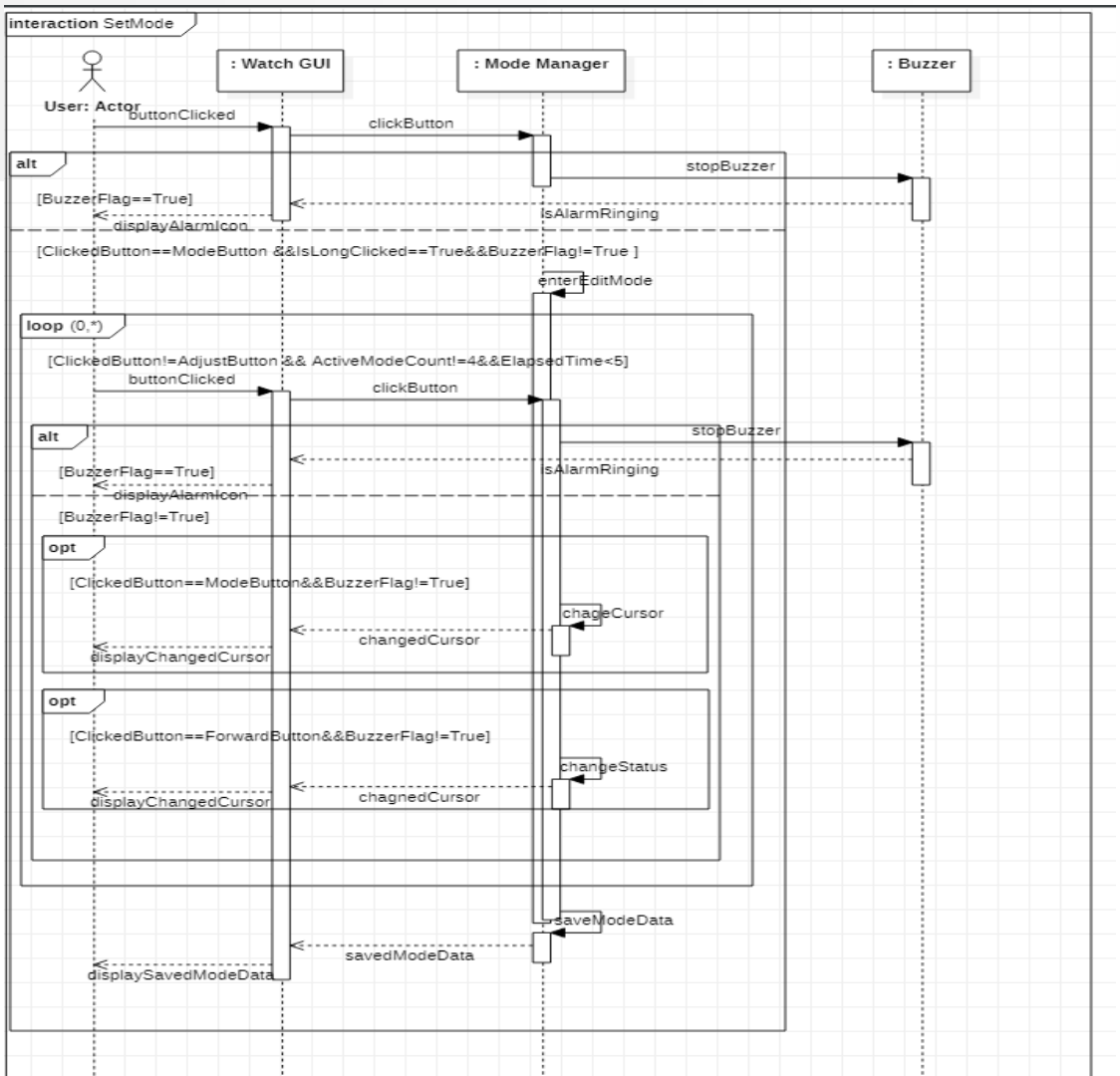
Stop Buzz


interaction StopBuzz
: Watch GUI
: ModeManager
: Buzzer
User: Actor
buttonClicked
clickButton
opt
[ClickedButton!=null]
stopBuzzer
isAlarmRinging
displayAlarmIcon

Change Mode


interaction ChangeMode
User: Actor
:watchGUI
:ModeManager
:Buzzer
buttonClicked
clickButton
alt
stopBuzzer
[BuzzerFlag == True]
displayMode
isAlarmRinging
[BuzzerFlag != True]
opt
[ClickedButton == ModeButton && isLongClicked != True]
alt
[isLastModeIndex != True]
changeMode
displayChangeMode
displayMode

Set Mode


interaction SetMode
User: Actor
: Watch GUI
: Mode Manager
: Buzzer
buttonClicked
clickButton
alt
stopBuzzer
[BuzzerFlag==True]
isAlarmRinging
displayAlarmIcon
[ClickedButton==ModeButton &&IsLongClicked==True&&BuzzerFlag!=True]
enterEditMode
loop (0,*)
[ClickedButton!=AdjustButton && ActiveModeCount!=4&&ElapsedTime<5]
buttonClicked
clickButton
alt
stopBuzzer
[BuzzerFlag==True]
isAlarmRinging
displayAlarmIcon
[BuzzerFlag!=True]
opt
[ClickedButton==ModeButton&&BuzzerFlag!=True]
chageCursor
changedCursor
displayChangedCursor
opt
[ClickedButton==ForwardButton&&BuzzerFlag!=True]
changeStatus
displayChangedCursor
chagnedCursor
saveModeData
savedModeData
displaySavedModeData

Write Test Code & Unit Testing 3

Time Test

```
@Test
public void timeflow() {
    Time t=new Time();
    for(int i=0;i<100;i++){...}
    int result=t.getCurrentTime().getSecond();
    assertEquals( expected: 1,result);
    LocalDateTime test=LocalDateTime.of( year: 9999, month: 12, dayOfMonth: 31, hour: 23, minute: 59, second: 59, nanoOfSecond: 990000000);
    t.setCurrentTime(test);
    for(int i=0;i<100;i++){
        t.timeflow();
    }
    //10000년 예외체크 0으로 돌아가는지.
    result=t.getCurrentTime().getYear();
    assertEquals( expected: 0,result);
}
```

```
public void changeCursor() {
    Time t= new Time();
    t.changeCursor();
    //한 번 바꿨을 때 cursor증가하는지
    int result=t.getCurrentCursor();
    assertEquals( expected: 1,result);
    t.changeCursor();
    t.changeCursor();
    t.changeCursor();
    t.changeCursor();
    t.changeCursor();
    t.changeCursor();
    t.changeCursor();

    //계속 change해서 끝에다다랐을 때 다시 처음으로 돌아오는지
    result=t.getCurrentCursor();
    assertEquals( expected: 0,result);
}
```

```
public void saveData() {
    Time t= new Time();
    LocalDateTime test=LocalDateTime.of( year: 7878, month: 1, dayOfMonth: 1, hour: 00, minute: 00, second: 00, nanoOfSecond: 0);
    t.setEditTime(test);
    t.saveData();
    int result=t.getCurrentTime().getYear();
    assertEquals( expected: 7878,result);
    assertEquals( expected: null,t.getEditTime());
}
}
```

Time Test

```java
@Test
public void increaseData() {
    Time t= new Time();
    t.enterEditData();

    t.increaseData();
    int result=t.getEditTime().getSecond();
    assertEquals( expected: 1,result);
    LocalDateTime test=LocalDateTime.of( year: 9999, month: 12, dayOfMonth: 31, hour: 23, minute: 59, second: 59, nanoOfSecond: 0);
    t.setEditTime(test);
    t.increaseData();
    //년도 10000이상일 경우 0이되는지
    result=t.getEditTime().getYear();
    assertEquals( expected: 0,result);

    //gmt
    t.setCurrentCursor(6);
    t.increaseData();
    result=t.getGMT();
    assertEquals( expected: 10,result);

    t.increaseData();
    t.increaseData();
    t.increaseData();
    t.increaseData();
    t.increaseData();
    result=t.getGMT();
    assertEquals( expected: -12,result);

    //format
    t.setCurrentCursor(7);
    t.increaseData();
    boolean re=t.getFormat();
    assertEquals( expected: false,re);
}
```

```java
@Test
public void decreaseData() {
    Time t= new Time();
    t.enterEditData();

    t.decreaseData();
    int result=t.getEditTime().getSecond();
    assertEquals( expected: 59,result);
    LocalDateTime test=LocalDateTime.of( year: 0, month: 1, dayOfMonth: 1, hour: 00, minute: 00, second: 00, nanoOfSecond: 0);
    t.setEditTime(test);
    t.decreaseData();
    //년도 10000이상일 경우 0이되는지
    result=t.getEditTime().getYear();
    assertEquals( expected: 9999,result);

    //gmt
    t.setCurrentCursor(6);
    t.decreaseData();
    result=t.getGMT();
    assertEquals( expected: 8,result);

    t.setGMT(-10);
    t.decreaseData();
    t.decreaseData();
    t.decreaseData();
    result=t.getGMT();
    assertEquals( expected: 14,result);

    //format
    t.setCurrentCursor(7);
    t.decreaseData();
    boolean re=t.getFormat();
    assertEquals( expected: false,re);

}
```

Time Test - passed

Test Results	18 ms
TimeTest	18 ms
timeflow()	13 ms
saveData()	2 ms
changeCursor()	1 ms
decreaseData()	1 ms
increaseData()	1 ms

Alarm Test

```
public void IsAlarmTimeCheck() {
    ModeManager man = new ModeManager();
    Buzzer buzzer = man.getBuzzer();

    Alarm alarm = ((Alarm)(man.getmodes()[1]));
    alarm.enterEditAlarm();
    alarm.increaseAlarmTime();
    alarm.changeCursor();
    alarm.increaseAlarmTime();
    alarm.saveAlarm();
    //알람을 01:01로 설정.
    Time time = ((Time)(man.getmodes()[0]));
    //시간을 00:01:01로 설정.
    time.setCurrentTime(LocalDateTime.of( year: 2020, month: 1, dayOfMonth: 1, hour: 1, minute: 1,  second: 0));

    alarm.isAlarmTimeCheck();
    //alarm을 활성화시키지 않아 buzzer가 울리지 않는 상태
    assertEquals( expected: false, buzzer.getBuzzerOn());
    assertFalse(buzzer.getIsAlarmRinging());

    //alarm을 활성화시킴
    alarm.turnOnOffAlarm();
    alarm.isAlarmTimeCheck();
    assertEquals( expected: true, buzzer.getBuzzerOn());
    assertTrue(buzzer.getIsAlarmRinging());

}
```

```
public void ChangeAlarm() {
    Buzzer buzzer = new Buzzer();
    Time time = new Time();

    Alarm alarm = new Alarm(buzzer, time);

    alarm.changeAlarm();
    alarm.changeAlarm();
    alarm.changeAlarm();
    alarm.changeAlarm();
    //alarm을 4번 바꾸면 0번째로 되돌아감.
    assertEquals( expected: 0, alarm.getCurrentAlarmIndex());
}

@Test
public void TurnOnOffAlarm() {
    Buzzer buzzer = new Buzzer();
    Time time = new Time();

    Alarm alarm = new Alarm(buzzer, time);

    //alarm을 deactivate->activate.
    alarm.changeAlarm();
    alarm.turnOnOffAlarm();
    boolean isActivated = alarm.getCurrentAlarmTimerObject().isActivatedTimer();

    assertEquals( expected: true, isActivated);

}
```

Alarm Test

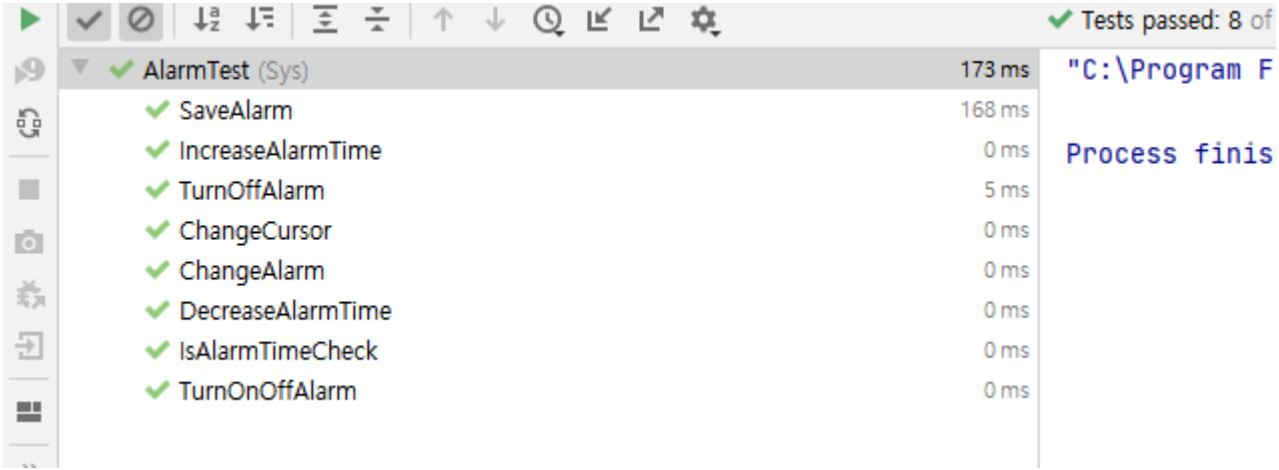
```java
public void IncreaseAlarmTime() {
    Buzzer buzzer = new Buzzer();
    Time time = new Time();
    int i;
    Alarm alarm = new Alarm(buzzer, time);

    alarm.enterEditAlarm();
    //hour를 25번 증가
    for(i=0; i < 25; i++)
        alarm.increaseAlarmTime();
    alarm.changeCursor();
    //minute를 70번 증가.
    for(i=0; i < 70; i++)
        alarm.increaseAlarmTime();
    LocalDateTime s = alarm.getCopyOfAlarmTimer();
    LocalTime t = s.toLocalTime();

    assertEquals( expected: 0,  t.compareTo(LocalTime.of( hour: 2, minute: 10, second: 0)));
}


@Test
public void DecreaseAlarmTime() {
    Buzzer buzzer = new Buzzer();
    Time time = new Time();

    Alarm alarm = new Alarm(buzzer, time);

    alarm.enterEditAlarm();
    alarm.decreaseAlarmTime();
    alarm.changeCursor();
    alarm.decreaseAlarmTime();

    LocalDateTime s = alarm.getCopyOfAlarmTimer();
    LocalTime t = s.toLocalTime();
    //00:00에서 hour를 1번 감소, minute를 1번 감소. -> 22:59분 이 출력되어야 정상.
    assertEquals( expected: 0,  t.compareTo(LocalTime.of( hour: 22, minute: 59, second: 0)));
}
```

```java
@Test
public void ChangeCursor() {
    Buzzer buzzer = new Buzzer();
    Time time = new Time();

    Alarm alarm = new Alarm(buzzer, time);
    alarm.changeCursor(); alarm.changeCursor(); alarm.changeCursor(); alarm.changeCursor();
    assertEquals( expected: true, alarm.isCursorHour());

}

@Test
public void SaveAlarm() {
    Buzzer buzzer = new Buzzer();
    Time time = new Time();

    Alarm alarm = new Alarm(buzzer, time);
    alarm.enterEditAlarm();
    alarm.increaseAlarmTime(); alarm.increaseAlarmTime(); alarm.increaseAlarmTime();
    alarm.changeCursor();
    alarm.increaseAlarmTime();
    alarm.increaseAlarmTime();
    LocalDateTime s = alarm.getCopyOfAlarmTimer();
    LocalTime t = s.toLocalTime();

    alarm.saveAlarm();
    //alarm의 시간이 저장되어 보고있던 alarm의 index에 저장이 되는지 test.
    assertEquals( expected: 0, alarm.getCopyOfAlarmTimer().toLocalTime().
            compareTo(alarm.getCurrentAlarmTimerObject().requestExpirationTime()));


}
```

Alarm Test

```
@Test
public void TurnOffAlarm(){
    ModeManager man = new ModeManager();
    Buzzer buzzer = man.getBuzzer();

    Alarm alarm = ((Alarm)(man.getmodes()[1]));
    alarm.enterEditAlarm();
    alarm.increaseAlarmTime();
    alarm.changeCursor();
    alarm.increaseAlarmTime();
    alarm.saveAlarm();
    Time time = ((Time)(man.getmodes()[0]));
    time.setCurrentTime(LocalDateTime.of( year: 2020, month: 1, dayOfMonth: 1, hour: 1, minute: 1));

    alarm.isAlarmTimeCheck();
    assertEquals( expected: false, buzzer.getBuzzerOn());
    assertFalse(buzzer.getIsAlarmRinging());

    alarm.turnOnOffAlarm();
    alarm.isAlarmTimeCheck();
    assertEquals( expected: true, buzzer.getBuzzerOn());
    //알람이 두번 울릴 경우 버저가 그대로 울리는지 test

}
```

Alarm Test- passed



Timer Test

```java
public class TimerTest {

    @Test
    public void enterEditTimer(){
        Buzzer buzzer = new Buzzer();
        Timer timer=new Timer(buzzer);

        timer.enterEditTimer();
        //enter Edit시 Timer 변수 체크
        assertEquals(timer.getpauseTimerFlag(), actual: true);
        assertEquals(timer.getsaveTimerFlag(), actual: false);
        assertEquals(timer.getCurrentCursor(), actual: 0);

    }

    @Test
    public void changeCursor(){
        Buzzer buzzer = new Buzzer();
        Timer timer=new Timer(buzzer);

        int tempCursor=timer.getCurrentCursor();
        tempCursor=(tempCursor+1)%3;
        timer.changeCursor();
        //changeCursor 후 현재 커서가 제대로 변경되었는지 check
        assertEquals(timer.getCurrentCursor(),tempCursor);

    }
```

```java
@Test
public void increaseData(){
    Buzzer buzzer = new Buzzer();
    Timer timer=new Timer(buzzer);

    timer.enterEditTimer();
    //시간test
    //현재 커서가 0(시간인지 확인)
    assertEquals(timer.getCurrentCursor(), actual: 0);

    //시간이 99시 일때 increase data 했을 때 0(day=1, hour=0) 되는지 확인
    timer.setTimerTime(LocalDateTime.of( year: 2000, month: 1, dayOfMonth: 5, hour: 3, minute: 0, second: 0));
    timer.increaseData();
    assertEquals(timer.getTimerTime().getDayOfMonth(), actual: 1);
    assertEquals(timer.getTimerTime().getHour(), actual: 0);

    //시간이 0일때 increasedata후 1더해지는지 확인
    int temp=timer.getTimerTime().getHour()+1;
    timer.increaseData();
    assertEquals(timer.getTimerTime().getHour(), temp);

    //분 테스트
    timer.changeCursor();
    //현재 커서가 1(분)인지 확인
    assertEquals(timer.getCurrentCursor(), actual: 1);

    //분이 59일때 increasedat 후 0이 되는 지 확인
    timer.setTimerTime(LocalDateTime.of( year: 2000, month: 1, dayOfMonth: 5, hour: 3, minute: 59, second: 0));
    timer.increaseData();
    assertEquals(timer.getTimerTime().getHour(), actual: 3);
    assertEquals(timer.getTimerTime().getMinute(), actual: 0);
    assertEquals(timer.getTimerTime().getSecond(), actual: 0);

    //분이 0일때 increasedata 후 1되는지 확인
    temp=timer.getTimerTime().getMinute()+1;
    timer.increaseData();
    assertEquals(timer.getTimerTime().getMinute(),temp);

    //초 테스트
    timer.changeCursor();
    //현재 커서가 2(초)인지 확인
    assertEquals(timer.getCurrentCursor(), actual: 2);

    //초가 59일때 increasedat 후 0이 되는 지 확인
    timer.setTimerTime(LocalDateTime.of( year: 2000, month: 1, dayOfMonth: 5, hour: 3, minute: 0, second: 59));
    timer.increaseData();
    assertEquals(timer.getTimerTime().getMinute(), actual: 0);
    assertEquals(timer.getTimerTime().getSecond(), actual: 0);

    //초가 0일때 increasedata 후 1되는지 확인
    temp=timer.getTimerTime().getSecond()+1;
    timer.increaseData();
    assertEquals(timer.getTimerTime().getSecond(),temp);
}
```

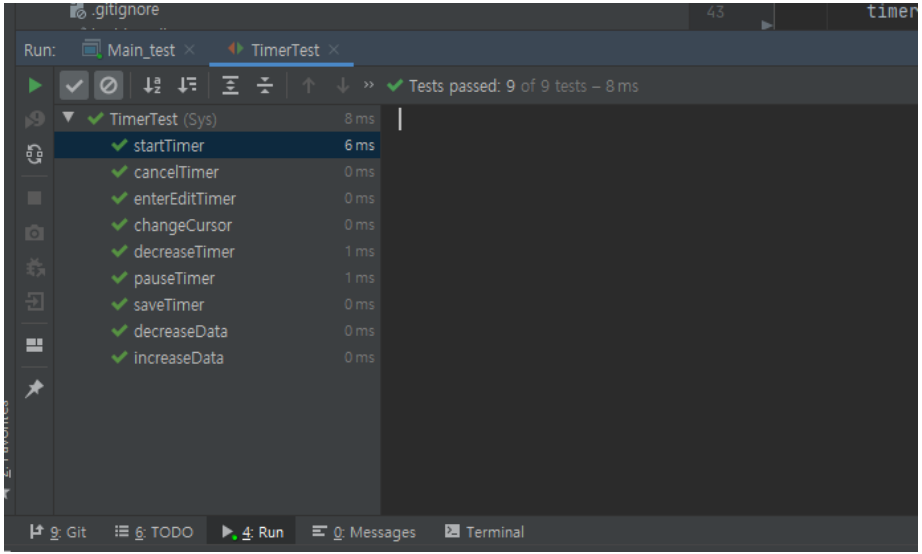
```java
@Test
public void decreaseData(){
    Buzzer buzzer = new Buzzer();
    Timer timer=new Timer(buzzer);

    timer.enterEditTimer();

    //시간test
    //현재 커서가 0(시간인지 확인)
    assertEquals(timer.getCurrentCursor(), actual: 0);

    //시간이 0시 일때 decreaseData 했을 때 99(day=5, hour=3) 되는지 확인
    timer.setTimerTime(LocalDateTime.of( year: 2000, month: 1, dayOfMonth: 1, hour: 0, minute: 0, second: 0));
    timer.decreaseData();
    assertEquals(timer.getTimerTime().getDayOfMonth(), actual: 5);
    assertEquals(timer.getTimerTime().getHour(), actual: 3);

    //시간이 99일때 decreaseData 후 1 감소되는지 확인
    int temp=timer.getTimerTime().getHour()-1;
    timer.decreaseData();
    assertEquals(timer.getTimerTime().getHour(), temp);

    //분 테스트
    timer.changeCursor();
    //현재 커서가 1(분)인지 확인
    assertEquals(timer.getCurrentCursor(), actual: 1);

    //분이 0일때 decreaseData 후 59이 되는 지 확인
    timer.setTimerTime(LocalDateTime.of( year: 2000, month: 1, dayOfMonth: 5, hour: 3, minute: 0, second: 0));
    timer.decreaseData();
    assertEquals(timer.getTimerTime().getHour(), actual: 3);
    assertEquals(timer.getTimerTime().getMinute(), actual: 59);
    assertEquals(timer.getTimerTime().getSecond(), actual: 0);

    //분이 59일때 decreaseData 후 -1되는지 확인
    temp=timer.getTimerTime().getMinute()-1;
    timer.decreaseData();
    assertEquals(timer.getTimerTime().getMinute(),temp);

    //초 테스트
    timer.changeCursor();
    //현재 커서가 2(초)인지 확인
    assertEquals(timer.getCurrentCursor(), actual: 2);

    //초가 0일때 decreaseData 후 59이 되는 지 확인
    timer.setTimerTime(LocalDateTime.of( year: 2000, month: 1, dayOfMonth: 5, hour: 3, minute: 0, second: 0));
    timer.decreaseData();
    assertEquals(timer.getTimerTime().getMinute(), actual: 0);
    assertEquals(timer.getTimerTime().getSecond(), actual: 59);

    //초가 59일때 decreaseData 후 -1되는지 확인
    temp=timer.getTimerTime().getSecond()-1;
    timer.decreaseData();
    assertEquals(timer.getTimerTime().getSecond(),temp);
}
```

Timer Test

```java
@Test
public void saveTimer() {
    Buzzer buzzer = new Buzzer();
    Timer timer = new Timer(buzzer);

    timer.setTimerTime(LocalDateTime.of( year: 2000, month: 1, dayOfMonth: 1, hour: 1, minute: 1, second: 1));
    timer.setSettingTimer(LocalDateTime.of( year: 2000, month: 3, dayOfMonth: 1, hour: 5, minute: 2, second: 1));

    LocalDateTime temp=timer.getTimerTime();

    timer.saveTimer();

    assertEquals(timer.getTimerTime(),temp);


}

@Test
public void decreaseTimer() {
    Buzzer buzzer = new Buzzer();
    Timer timer = new Timer(buzzer);

    timer.setTimerTime(LocalDateTime.of( year: 2000, month: 1, dayOfMonth: 1, hour: 0, minute: 0, second: 4));
    timer.startTimer();
    //timer가 시작되었는지 확인
    assertEquals(timer.getpauseTimerFlag(), actual: false);
    //10ms씩 100번 내려서 1초 감소 시키고 4->3 확인
    for(int i=0;i<100;i++)
        timer.decreaseTimer();
    assertEquals(timer.getTimerTime().getSecond(), actual: 3);
    //3초가 더흐르게 하고 buzzer가 on되었는지 확인
    for(int i=0;i<300;i++)
        timer.decreaseTimer();
    //buzzer on 확인
    assertEquals(buzzer.getBuzzerOn(), actual: true);
    //Timer멈췄는지 확인
    assertEquals(timer.getpauseTimerFlag(), actual: true);
    //Timer
    assertEquals(timer.getTimerTime(),LocalDateTime.of( year: 2000, month: 1, dayOfMonth: 1, hour: 0, minute: 0, second: 0));
}
```

```java
    @Test
public void startTimer(){
    Buzzer buzzer = new Buzzer();
    Timer timer = new Timer(buzzer);

    //처음에 정지상태 확인
    assertEquals(timer.getpauseTimerFlag(), actual: true);

    //리셋 상태일 때 startTimer후 시작이 되지않는지 확인
    timer.setTimerTime(LocalDateTime.of( year: 2000, month: 1, dayOfMonth: 1, hour: 0, minute: 0, second: 0));
    timer.startTimer();
    assertEquals(timer.getpauseTimerFlag(), actual: true);

    //리셋 상태가 아닐때 startTimer후 시작이 되는 지 확인
    timer.setTimerTime(LocalDateTime.of( year: 2000, month: 1, dayOfMonth: 1, hour: 4, minute: 0, second: 0));
    timer.startTimer();
    assertEquals(timer.getpauseTimerFlag(), actual: false);

}

@Test
public void pauseTimer(){
    Buzzer buzzer = new Buzzer();
    Timer timer = new Timer(buzzer);

    //리셋 상태가 아닐 때 startTimer 후 시작상태 확인
    timer.setTimerTime(LocalDateTime.of( year: 2000, month: 1, dayOfMonth: 1, hour: 4, minute: 0, second: 0));
    timer.startTimer();
    assertEquals(timer.getpauseTimerFlag(), actual: false);
    //pauseTimer 후 종료되었는지 확인
    timer.pauseTimer();
    assertEquals(timer.getpauseTimerFlag(), actual: true);

}

@Test
public void cancelTimer() {
    Buzzer buzzer = new Buzzer();
    Timer timer = new Timer(buzzer);

    timer.setTimerTime(LocalDateTime.of( year: 2000, month: 1, dayOfMonth: 1, hour: 1, minute: 1, second: 1));
    timer.setSettingTimer(LocalDateTime.of( year: 2000, month: 3, dayOfMonth: 1, hour: 5, minute: 2, second: 1));
    LocalDateTime temp=timer.getSettingTimer();

    //cancel Timer 후 settingTimer의 값이 timerTime에 저장되었는지 확인
    timer.cancelTimer();

    assertEquals(timer.getTimerTime(),temp);

}
```

Timer Test


.gitignore
43
timer
Run:
Main_test
TimerTest
Tests passed: 9 of 9 tests – 8 ms
TimerTest (Sys) 8 ms
startTimer 6 ms
cancelTimer 0 ms
enterEditTimer 0 ms
changeCursor 0 ms
decreaseTimer 1 ms
pauseTimer 1 ms
saveTimer 0 ms
decreaseData 0 ms
increaseData 0 ms
9: Git
6: TODO
4: Run
0: Messages
Terminal

Stopwatch Test

```
public void startStopwatch() {
    StopWatch stopWatch = new StopWatch();
    int i;
    stopWatch.startStopwatch();

    for(i=0; i < 10002; i++)
        stopWatch.increaseCurrentTime();
    assertEquals( expected: 0, stopWatch.getCurrentStopWatchTime().compareTo(LocalTime.of( hour: 0, minute: 1, second: 40, nanoOfSecond: 20000000)));

    assertEquals( expected: false, stopWatch.getIsPaused());

    //한계를 넘어서 stopwatch가 진행되었을 때 한계점에서 멈춘다.
    for(i=0; i < 60000000; i++)
        stopWatch.increaseCurrentTime();
    assertEquals( expected: 0, stopWatch.getCurrentStopWatchTime().compareTo(LocalTime.of( hour: 1, minute: 39, second: 59, nanoOfSecond: 990000000)));

    assertEquals( expected: true, stopWatch.getIsPaused());



}


@Test
public void resumeStopwatch() {
    StopWatch stopWatch = new StopWatch();
    stopWatch.startStopwatch(); stopWatch.pauseStopwatch();

    stopWatch.resumeStopwatch();
    assertFalse(stopWatch.getIsPaused());
}

@Test
public void pauseStopwatch() {
    StopWatch stopWatch = new StopWatch();
    stopWatch.startStopwatch();
    stopWatch.pauseStopwatch();
    assertTrue(stopWatch.getIsPaused());
}
```

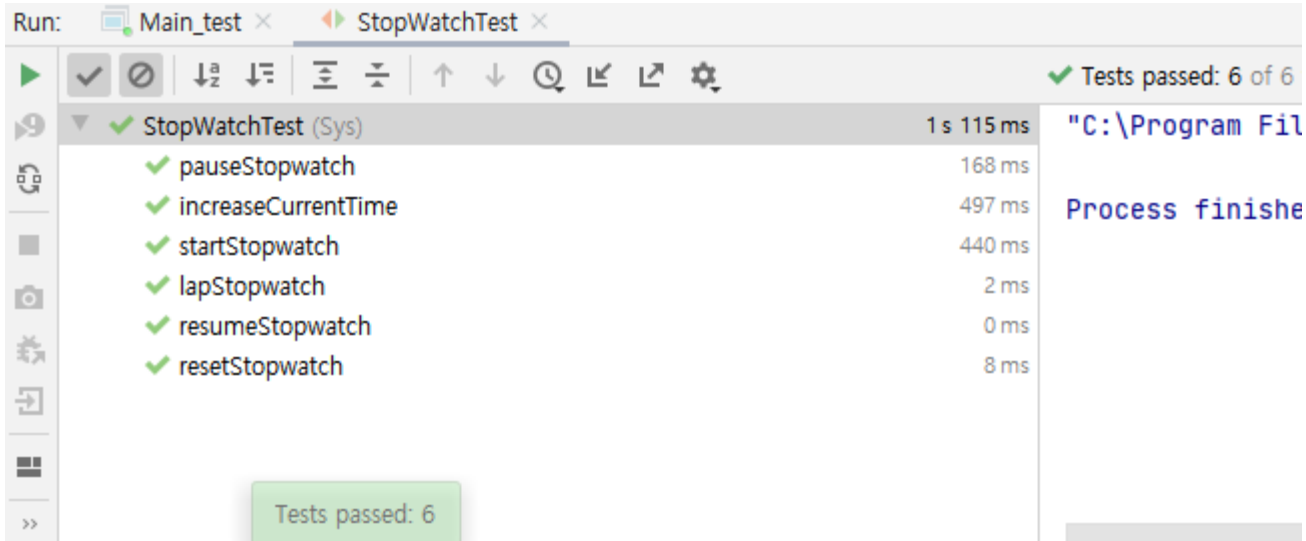
Stopwatch Test

```java
@Test
public void resetStopwatch() {
    StopWatch stopWatch = new StopWatch();
    int i;
    stopWatch.startStopwatch();
    for(i=0; i < 10002; i++)
        stopWatch.increaseCurrentTime();

    stopWatch.resetStopwatch();
    assertEquals( expected: 0, stopWatch.getCurrentStopWatchTime().compareTo(LocalTime.of( hour: 0, minute: 0, second: 0)));
}

@Test
public void lapStopwatch() {
    StopWatch stopWatch = new StopWatch();
    int i;
    stopWatch.startStopwatch();
    for(i=0; i < 10002; i++)
        stopWatch.increaseCurrentTime();
    stopWatch.lapStopwatch();
    for(i=0; i < 10002; i++)
        stopWatch.increaseCurrentTime();
    //lapTime의 저장.
    assertEquals( expected: 0,  stopWatch.getLapStopWatchTime().compareTo(LocalTime.of( hour: 0, minute: 1, second: 40, nanoOfSecond: 20000000)));
}
```

Stopwatch Test

```
@Test
public void increaseCurrentTime() {

    StopWatch stopWatch = new StopWatch();

    stopWatch.startStopwatch();
    int i;


    for(i=0; i < 6; i++)
        stopWatch.increaseCurrentTime();
    assertFalse(stopWatch.getIsPaused());

    for(i=0; i < 70000000; i++)
        stopWatch.increaseCurrentTime();
    assertEquals( expected: 0, stopWatch.getCurrentStopWatchTime().compareTo(LocalTime.of( hour: 1, minute: 39, second: 59, nanoOfSecond: 990000000)));

}
```

Stopwatch Test – passed



Calorie Check Test

```java
public void increaseData() {
    CalorieCheck calorieCheck = new CalorieCheck();

    //cursor의 기본값은 true
    //cursor = true, tempWeight 증가
    //정상적인 경우
    int tempWeight = calorieCheck.getWeight();
    calorieCheck.setTempWeight(tempWeight);
    calorieCheck.increaseData();
    assertTrue(calorieCheck.getCursor());
    assertEquals( expected: tempWeight+1, calorieCheck.getTempWeight());

    //tempWeight = 999이면 0으로 바꿔야 한다
    tempWeight = 999;
    calorieCheck.setTempWeight(tempWeight);
    calorieCheck.increaseData();
    assertEquals( expected: 0, calorieCheck.getTempWeight());


    //cursor = false, Speed 증가
    calorieCheck.setCursor(false);
    int tempSpeed = calorieCheck.getSpeed();
    calorieCheck.setTempSpeed(tempSpeed);
    calorieCheck.increaseData();
    assertFalse(calorieCheck.getCursor());
    assertEquals( expected: tempSpeed+1, calorieCheck.getTempSpeed());

    //tempSpeed = 99면 0으로 바꿔야 한다.
    tempSpeed = 99;
    calorieCheck.setTempSpeed(tempSpeed);
    calorieCheck.increaseData();
    assertEquals( expected: 0, calorieCheck.getTempSpeed());
}
```

```java
@Test
public void decreaseData() {
    CalorieCheck calorieCheck = new CalorieCheck();

    //cursor의 기본값은 true
    //cursor = true, tempWeight 감소
    //정상적인 경우
    int tempWeight = calorieCheck.getWeight();
    calorieCheck.setTempWeight(tempWeight);
    calorieCheck.decreaseData();
    assertTrue(calorieCheck.getCursor());
    assertEquals( expected: tempWeight-1, calorieCheck.getTempWeight());

    //tempWeight = 0이면 999으로 바꿔야 한다
    tempWeight = 0;
    calorieCheck.setTempWeight(tempWeight);
    calorieCheck.decreaseData();
    assertEquals( expected: 999, calorieCheck.getTempWeight());


    //cursor = false, Speed 증가
    calorieCheck.setCursor(false);
    int tempSpeed = calorieCheck.getSpeed();
    calorieCheck.setTempSpeed(tempSpeed);
    calorieCheck.decreaseData();
    assertFalse(calorieCheck.getCursor());
    assertEquals( expected: tempSpeed-1, calorieCheck.getTempSpeed());

    //tempSpeed = 0면 99으로 바꿔야 한다.
    tempSpeed = 0;
    calorieCheck.setTempSpeed(tempSpeed);
    calorieCheck.decreaseData();
    assertEquals( expected: 99, calorieCheck.getTempSpeed());
}
```

Calorie Check Test

```java
@Test
public void changeCursor() {
    CalorieCheck calorieCheck = new CalorieCheck();
    boolean cursor = calorieCheck.getCursor();
    calorieCheck.changeCursor();

    //커서를 바꿔보고 실제로 바뀌었는지 확인
    assertEquals(!cursor, calorieCheck.getCursor());
}
```

```java
@Test
public void enterSetSpeedandWeight() {
    CalorieCheck calorieCheck = new CalorieCheck();

    calorieCheck.enterSetSpeedandWeight();
    //실제 speed, weight가 tempSpeed, tempWeight에 들어갔는지 확인
    assertEquals(calorieCheck.getSpeed(), calorieCheck.getTempSpeed());
    assertEquals(calorieCheck.getWeight(), calorieCheck.getTempWeight());
}
```

```java
@Test
public void saveCalorieSetting() {
    CalorieCheck calorieCheck = new CalorieCheck();

    int expectedSpeed = 10;
    int expectedWeight = 66;
    //temp speed, weight 설정
    calorieCheck.setTempSpeed(expectedSpeed);
    calorieCheck.setTempWeight(expectedWeight);

    calorieCheck.saveCalorieSetting();

    //temp speed, weight가 실제 speed, weight에 저장됐는지 확인
    assertEquals(expectedSpeed, calorieCheck.getSpeed());
    assertEquals(expectedWeight, calorieCheck.getWeight());

    //cursor=true로 변했는지 확인
    assertTrue(calorieCheck.getCursor());
}
```

```java
@Test
public void startCalorieCheck() {
    CalorieCheck calorieCheck = new CalorieCheck();

    calorieCheck.startCalorieCheck();
    //flag 설정 제대로 됐는지 확인
    assertFalse(calorieCheck.getIsPause());
    assertTrue(calorieCheck.getIsStart());
}
```

Calorie Check Test

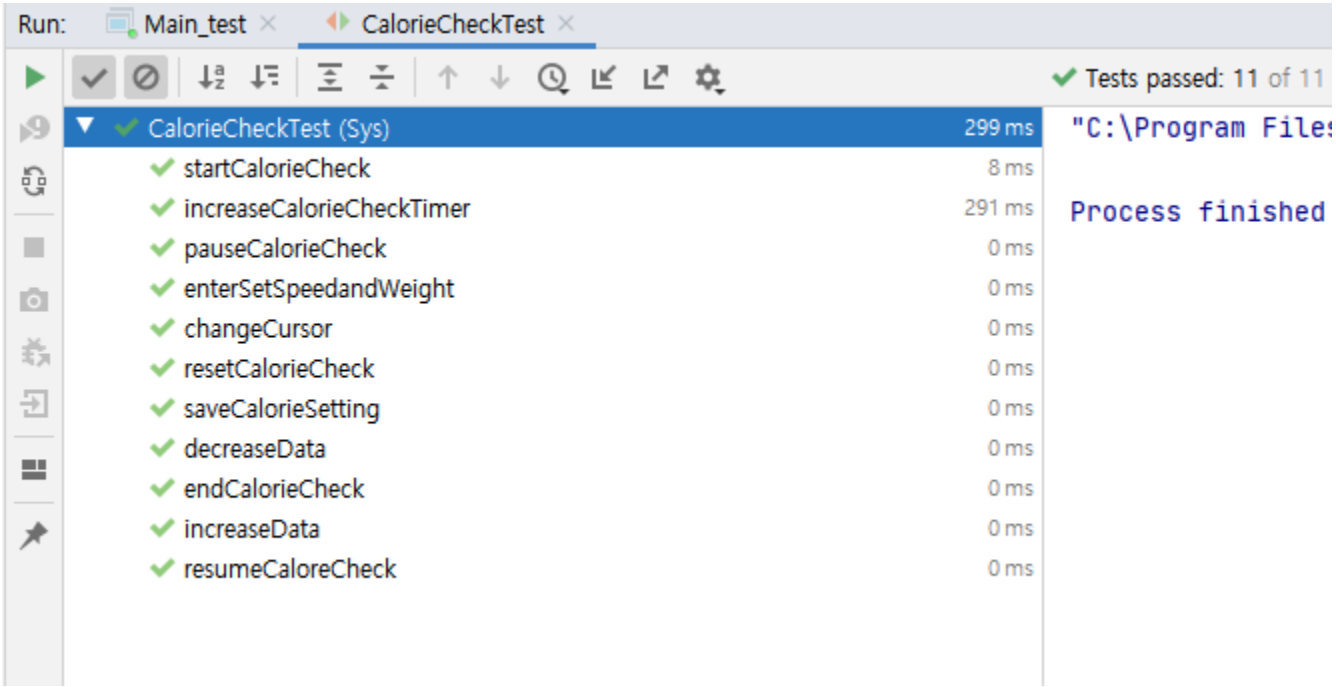
```java
@Test
public void resumeCaloreCheck() {
    CalorieCheck calorieCheck = new CalorieCheck();

    calorieCheck.resumeCaloreCheck();
    //flag 설정 제대로 됐는지 확인
    assertEquals( expected: false, calorieCheck.getIsPause());
}

@Test
public void pauseCalorieCheck() {
    CalorieCheck calorieCheck = new CalorieCheck();

    //flag 설정 제대로 됐는지 확인
    assertEquals( expected: true, calorieCheck.getIsPause());
}

@Test
public void endCalorieCheck() {
    CalorieCheck calorieCheck = new CalorieCheck();

    //flag 설정 제대로 됐는지 확인
    assertEquals( expected: true, calorieCheck.getIsPause());
    assertEquals( expected: false, calorieCheck.getIsStart());
}
```

```java
@Test
public void increaseCalorieCheckTimer() {
    CalorieCheck calorieCheck = new CalorieCheck();

    LocalTime time = calorieCheck.getCalorieTime();
    //calorieCheck.startCalorieCheck();
    calorieCheck.setStartFlag(true);
    calorieCheck.setPauseFlag(false);

    /*pause = false, start = true
    최초로 시작하고 0시 0분 0초에서 한번 증가시켜줬을 때를 가정*/
    calorieCheck.increaseCalorieCheckTimer();
    time = time.plusNanos(10000000);
    assertEquals(time, calorieCheck.getCalorieTime());

    double n = 23 * 3600 + 59 * 60 + 59;
    n = n*100;
    for(int i=0; i<n; i++){
        calorieCheck.increaseCalorieCheckTimer();
    }
    time = LocalTime.of( hour: 23,  minute: 59,  second: 59);

    //시간이 23시 59분 59초까지 늘어났는지 체크
    assertEquals(time, calorieCheck.getCalorieTime());
      assertTrue(time.equals(calorieCheck.getCalorieTime()));

    assertTrue(calorieCheck.getIsPause());
    assertFalse(calorieCheck.getIsStart());
}
```

Calorie Check Test

```java
@Test
public void resetCalorieCheck() {
    CalorieCheck calorieCheck = new CalorieCheck();

    calorieCheck.resetCalorieCheck();

    assertFalse(calorieCheck.getCursor());
    assertEquals( expected: 0, calorieCheck.getCalorie());
    assertEquals(LocalTime.of( hour: 0, minute: 0, second: 0, nanoOfSecond: 0)
        , calorieCheck.getCalorieTime());
}
```

Calorie Check Test - passed

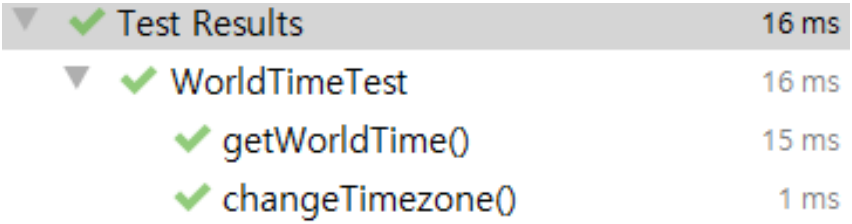


World Time Test

```java
@Test
public void changeTimezone() {
    WorldTime w= new WorldTime();
    w.changeTimezone();
    int result=w.getCurrentCity();
    assertEquals( expected: 1,result);


    w.changeTimezone();
    w.changeTimezone();
    w.changeTimezone();
    w.changeTimezone();
    w.changeTimezone();

    result=w.getCurrentCity();
    assertEquals( expected: 0,result);
}
```

```java
@Test
public void getWorldTime() {
    Time t=new Time();
    WorldTime w= new WorldTime();
    w.changeTimezone();

    LocalDateTime tm=t.getCurrentTime();//set gmt to paris
    int gmt=t.getGMT();
    int result=w.getWorldTime(tm,gmt).getHour();

    assertEquals( expected: 16,result);
}
}
```

World Time Test - passed

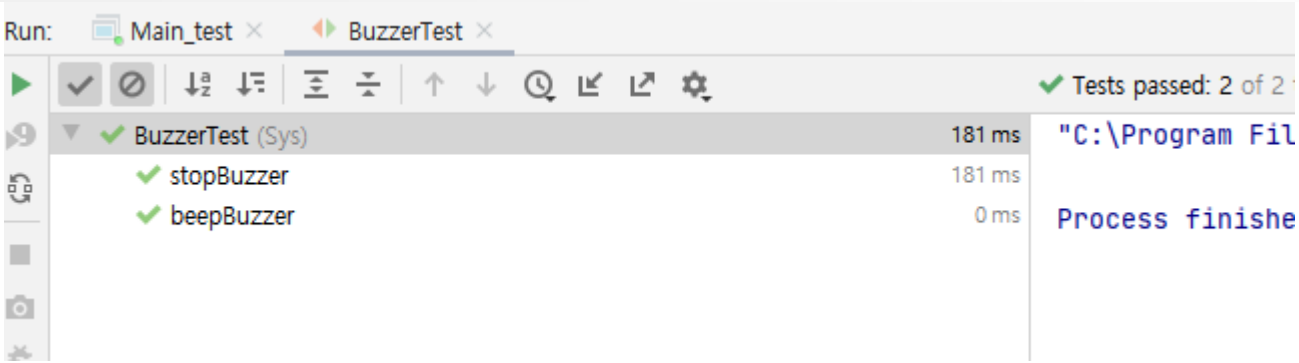


Buzzer test

```
public void beepBuzzer(){
    ModeManager man = new ModeManager();
    Buzzer buzzer = man.getBuzzer();

    Alarm alarm = ((Alarm)(man.getmodes()[1]));
    alarm.enterEditAlarm();

    alarm.increaseAlarmTime(); alarm.changeCursor(); alarm.increaseAlarmTime();
    alarm.saveAlarm(); alarm.turnOnOffAlarm();  //01:01로 1번 alarm 설정, 활성화

    Time time = ((Time)(man.getmodes()[0]));

    alarm.changeAlarm(); alarm.turnOnOffAlarm();    //00:00으로 2번 alarm 설정, 활성화

    time.setCurrentTime(LocalDateTime.of( year: 2020, month: 1, dayOfMonth: 1, hour: 0, minute: 0));
    alarm.isAlarmTimeCheck();
    //alarm이 울리고 있는 상태.
    time.setCurrentTime(LocalDateTime.of( year: 2020, month: 1, dayOfMonth: 1, hour: 1, minute: 1));

    assertEquals( expected: true, buzzer.getBuzzerOn());
    assertTrue(buzzer.getIsAlarmRinging());
    alarm.isAlarmTimeCheck();
    //버저가 울리고 있는 상태에서 알람을 꺼주는 operation을 test내에서 부르지 않았기때문에 flag는 계속 켜져있고
    // 계속 울리고 있는 상태.
    assertEquals( expected: true, buzzer.getBuzzerOn());

}
```

```
@Test
public void stopBuzzer() {
    ModeManager man = new ModeManager();
    Buzzer buzzer = man.getBuzzer();

    Alarm alarm = ((Alarm)(man.getmodes()[1]));
    alarm.enterEditAlarm();
    alarm.increaseAlarmTime();
    alarm.changeCursor();
    alarm.increaseAlarmTime();
    alarm.saveAlarm();
    alarm.turnOnOffAlarm();
    Time time = ((Time)(man.getmodes()[0]));
    time.setCurrentTime(LocalDateTime.of( year: 2020, month: 1, dayOfMonth: 1, hour: 1, minute: 1));
    alarm.isAlarmTimeCheck();
    time.setCurrentTime(LocalDateTime.of( year: 2020, month: 1, dayOfMonth: 1, hour: 1, minute: 1));
    alarm.isAlarmTimeCheck();

    //stopBuzzer를 하면 알람 아이콘이 켜져있었다면 꺼지고  buzzer가 꺼진다.
    assertTrue(buzzer.getBuzzerOn());
    buzzer.stopBuzzer();
    assertFalse(buzzer.getBuzzerOn());
    assertFalse(buzzer.getIsAlarmRinging());
}
```

Buzzer test - passed



Testing Traceability Analysis

4

Use Case	Match
Set Time	S16,S17,S18,S19,S20
Set Timer	S34,S39,S40,S41
Start Timer	S35
Pause Timer	S37
Resume Timer	S36
Cancel Timer	S38
Change Alarm	S11
Turn On/Off Alarm	S12
Set Alarm	S9,S10,S13,S14,S15
Start Stopwatch	S23
Pause Stopwatch	S24
Resume Stopwatch	S25
Lap Stopwatch	S27
Reset Stopwatch	S26
Change Timezone	S28
Start Calorie Check	S1
Pause Calorie Check	S2
Resume Calorie Check	S3
Set Speed and Weight	S4,S6,S7,S8
Reset Calorie Check	S5
Stop Buzz	S22
Change Mode	S33
SetMode	S29,S31,S30,S32

index	Operation in Sequence Diagram	Match
S1	startCalorieCheck()	I1
S2	pauseCalorieCheck()	I3
S3	resumeCalorieCheck()	I4
S4	enterSetSpeedandWeight()	I5,I9,I10,I11,I7
S5	resetCalorieCheck()	I8
S6	CalorieCheck.increaseData()	I9
S7	CalorieCheck.decreaseData()	I10
S8	CalorieCheck.changeCursor()	I11
S9	enterEditAlarm()	I28
S10	saveCurrentAlarm()	I32
S11	changeAlarm()	I33
S12	turnOnOffAlarm()	I34
S13	increaseAlarmTime()	I30
S14	decreaseAlarmTime()	I31
S15	Alarm.changeCursor()	I29
S16	saveData()	I22
S17	enterEditData()	I23
S18	Time.increaseData()	I24
S19	Time.decreaseData()	I25
S20	Time.changeCursor()	I26
S21	clickButton()	I27
S22	stopBuzzer()	I50
S23	startStopwatch()	I38
S24	pauseStopwatch()	I40
S25	resumeStopwatch()	I41
S26	resetStopwatch()	I42
S27	lapStopwatch()	I43
S28	changeTimezone()	I44
S29	enterEditMode()	I46
S30	saveModeData()	I49
S31	changeCursor()	I47
S32	changeStatus()	I48
S33	changeMode()	I45
S34	enterEditTimer	I12,I19,I20,I21,I13
S35	startTimer	I14
S36	resumeTimer	I16
S37	pauseTimer	I15
S38	cancelTimer	I18
S39	Timer.increaseData()	I19
S40	Timer.decreaseData()	I20
S41	Timer.changeCursor()	I21

index	Operation in Interaction Diagram	Match
I1	startCalorieCheck	M7
I2	increaseCalorieCheckTimer	M8
I3	pauseCalorieCheck	M9
I4	resumeCaloreCheck	M16
I5	enterSetSpeedandWeight	M10
I6	changeCursor	M13
I7	saveCalorieSetting	M14
I8	resetCalorieCheck	M15
I9	CalorieCheck.increaseData()	M11
I10	CalorieCheck.decreaseData()	M12
I11	CalorieCheck.changeCursor()	M13
I12	enterEditTimer	M38,M39,M40,M41,M42
I13	saveTimer	M42
I14	startTimer	M43
I15	pauseTimer	M45
I16	resumeTimer	M43
I17	decreaseTimer	M40
I18	cancelTimer	M46
I19	Timer.increaseData()	M39
I20	Timer.decreaseData()	M40
I21	Timer.changeCursor()	M41
I22	saveData()	M52
I23	enterEditData()	M49
I24	Time.increaseData()	M50
I25	Time.decreaseData()	M51
I26	Time.changeCursor()	M53
I27	clickButton	M2
I28	enterEditAlarm	M27,M29,M30,M31,M32
I29	Alarm.changeCursor	M29
I30	increaseAlarmTime	M30
I31	decreaseAlarmTime	M31
I32	saveAlarm	M32
I33	changeAlarm	M33
I34	turnOnOffAlarm	M34
I35	requestExpirationTime	M35
I36	toggleAlarmTimer	M35
I37	saveAlarmTime	M36
I38	startStopwatch	M19
I39	increaseCurrentTime	M20
I40	pauseStopwatch	M21
I41	resumeStopwatch	M24
I42	resetStopwatch	M22
I43	lapStopwatch	M23
I44	changeTimezone	M25
I45	changeMode	M1
I46	enterEditMode()	M3
I47	changeCursor()	M5
I48	changeStatus()	M5
I49	saveModeData()	M6
I50	stopBuzzer()	M47

	Method	Class
M1	changeMode	ModeManager
M2	clickButton	
M3	enterEditMode	
M4	changeCursor	
M5	changeStatus	
M6	saveModeData	
M7	startCalorieCheck	Calorie Check
M8	increaseCalorieCheckTime	
M9	pauseCalorieCheck	
M10	enterSetSpeedandWeight	
M11	increaseData	
M12	decreaseData	
M13	changeCursor	
M14	saveCalorieSetting	
M15	resetCalorieCheck	
M16	resumeCaloreCheck	
M17	endCalorieCheck	
M18	calculateCalorie	
M19	startStopwatch	Stopwatch
M20	increaseCurrentTime	
M21	pauseStopwatch	
M22	resetStopwatch	
M23	lapStopwatch	
M24	resumeStopwatch	
M25	changeTimezone	WorldTime
M26	getWorldTime	
M27	enterEditAlarm	Alarm
M28	isAlarmTimeCheck	
M29	changeCursor	
M30	increaseAlarmTime	
M31	decreaseAlarmTime	
M32	saveAlarm	
M33	changeAlarm	
M34	turnOnOffAlarm	
M35	toggleAlarmTimer	AlarmTimer
M36	saveAlarmTime	
M37	requestExpirationTime	
M38	enterEditTimer	Timer
M39	increaseData	
M40	decreaseData	
M41	changeCursor	
M42	saveTimer	
M43	startTimer	
M44	decreaseTimer	
M45	pauseTimer	
M46	cancelTimer	
M47	stopBuzzer	Buzzer
M48	changeCursor	Time
M49	enterEditData	
M50	increaseData	
M51	decreaseData	
M52	saveData	
M53	changeCursor	

System Test	Match
Show Time Test	R 1.1
Set Time Test	R 1.2
Show Timer Test	R 2.1
Set Timer Test	R 2.2
Start Timer Test	R 2.3
Pause Timer Test	R 2.4
Resume Timer Test	R 2.5
Cancel Timer Test	R 2.6
Show Alarm Test	R3.1
Change Alarm Test	R3.2
Turn On/Off Alarm Test	R3.3
Set Alarm Test	R3.4
Return to Default Screen Test	R10.1
Show Stopwatch Test	R4.1
Start Stopwatch Test	R4.2
Pause Stopwatch Test	R4.3
Resume Stopwatch Test	R4.4
Reset Stopwatch Test	R4.5
Lap Stopwatch Test	R4.6
Show World Time Test	R5.1
Change Timezone Test	R5.2
Show Calorie check Test	R6.1
Set Speed and Weight test	R6.2
Start Calorie Check test	R6.3
Pause Calorie Check test	R6.4
Resume Calorie Check test	R6.5
Reset Calorie Check test	R6.6
Beep Buzz test	R9.1
Stop Buzz test	R9.2
Change Mode test	R7.1
Set Mode test	R8.1

index	Unit Test
U1	clickButton()
U2	enterEditMode()
U3	changeCursor()
U4	changeStatus()
U5	saveModeData()
U6	changeMode()
U7	changeCursor()
U8	increaseData()
U9	decreaseData()
U10	saveCalorieSetting()
U11	enterSetSpeedandWeight()
U12	startCalorieCheck()
U13	resumeCaloreCheck()
U14	pauseCalorieCheck()
U15	endCalorieCheck()
U16	increaseCalorieCheckTimer()
U17	resetCalorieCheck()
U18	startStopwatch()
U19	resumeStopwatch()
U20	pauseStopwatch()
U21	resetStopwatch()
U22	lapStopwatch()
U23	increaseCurrentTime()
U24	getWorldTime
U25	changeTimeZone
U26	IsAlarmTimeCheck()
U27	ChangeAlarm()
U28	TurnOnOffAlarm()
U29	IncreaseAlarmTime()
U30	DecreaseAlarmTime()
U31	ChangeCursor()
U32	SaveAlarm()
U33	TurnOffAlarm()
U34	enterEditTimer()
U35	changeCursor()
U36	increaseData()
U37	decreaseData()
U38	saveTimer()
U39	decreaseTimer()
U40	startTimer()
U41	pauseTimer()
U42	cancelTimer()
U43	beepBuzzer
U44	stopBuzzer()
U46	changeCursor
U47	enterEditData
U48	increaseData
U49	decreaseData
U50	saveData

System Test	Match
Show Time Test	R 1.1
Set Time Test	R 1.2
Show Timer Test	R 2.1
Set Timer Test	R 2.2
Start Timer Test	R 2.3
Pause Timer Test	R 2.4
Resume Timer Test	R 2.5
Cancel Timer Test	R 2.6
Show Alarm Test	R3.1
Change Alarm Test	R3.2
Turn On/Off Alarm Test	R3.3
Set Alarm Test	R3.4
Return to Default Screen Test	R10.1
Show Stopwatch Test	R4.1
Start Stopwatch Test	R4.2
Pause Stopwatch Test	R4.3
Resume Stopwatch Test	R4.4
Reset Stopwatch Test	R4.5
Lap Stopwatch Test	R4.6
Show World Time Test	R5.1
Change Timezone Test	R5.2
Show Calorie check Test	R6.1
Set Speed and Weight test	R6.2
Start Calorie Check test	R6.3
Pause Calorie Check test	R6.4
Resume Calorie Check test	R6.5
Reset Calorie Check test	R6.6
Beep Buzz test	R9.1
Stop Buzz test	R9.2
Change Mode test	R7.1
Set Mode test	R8.1

index	Unit Test
U1	clickButton()
U2	enterEditMode()
U3	changeCursor()
U4	changeStatus()
U5	saveModeData()
U6	changeMode()
U7	changeCursor()
U8	increaseData()
U9	decreaseData()
U10	saveCalorieSetting()
U11	enterSetSpeedandWeight()
U12	startCalorieCheck()
U13	resumeCaloreCheck()
U14	pauseCalorieCheck()
U15	endCalorieCheck()
U16	increaseCalorieCheckTimer()
U17	resetCalorieCheck()
U18	startStopwatch()
U19	resumeStopwatch()
U20	pauseStopwatch()
U21	resetStopwatch()
U22	lapStopwatch()
U23	increaseCurrentTime()
U24	getWorldTime
U25	changeTimeZone
U26	IsAlarmTimeCheck()
U27	ChangeAlarm()
U28	TurnOnOffAlarm()
U29	IncreaseAlarmTime()
U30	DecreaseAlarmTime()
U31	ChangeCursor()
U32	SaveAlarm()
U33	TurnOffAlarm()
U34	enterEditTimer()
U35	changeCursor()
U36	increaseData()
U37	decreaseData()
U38	saveTimer()
U39	decreaseTimer()
U40	startTimer()
U41	pauseTimer()
U42	cancelTimer()
U43	beepBuzzer
U44	stopBuzzer()
U46	changeCursor
U47	enterEditData
U48	increaseData
U49	decreaseData
U50	saveData

System Test Cases

5

System Test Cases

Test	Test 항목	Description	Use Case	System Function	Result
1	Show Time Test	설정된 현재 시간이 시계에 설정된 AM/PM, 12/24시간 방식 등의 형식을 준수하여 시간을 출력하는지 test	1. Show time	R 1.1	Pass
2	Set Time Test	-User에게 모든 시간 형식에 대하여 입력 받는 기능을 Test -입력 후 test case와 대조하여 저장되었는지 Test	2.Set Time	R 1.2	Pass
3	Show Timer Test	-Timer mode로 잘 전환 되는지 test -현재 Timer 시간이 잘 표시되는지 test	3. Show Timer	R 2.1	Pass
4	Set Timer Test	-Timer를 설정하고 설정 값이 반영되는지 Test -Timer가 일시정지 된 상태에서만 작동하는지 Test.	4. Set Timer	R 2.2	Pass
5-1	Start Timer Test	-Timer를 시작하고 시간 값이 감소하는 것이 반영되는지 test	5. Start Timer	R 2.3	Pass
5-2	Start Timer Test	-Timer가 만료된 상태 or 등록되지 않은 상태일 때 시작 버튼을 눌러도 아무 동작 하지 않는지 test	5. Start Timer	R 2.3	Pass
6	Pause Timer Test	Timer가 동작 중인 상태에서 pause 버튼을 입력 받았을 때 Timer가 일시정지 하는지 test	6. Pause Timer	R 2.4	Pass
7	Resume Timer Test	Timer가 pause상태에서 resume 버튼을 입력 받았을 때 일시정지 되어 있던 Timer가 다시 재시작이 정상적으로 되는지 test	7.Resume Timer	R 2.5	Pass
8-1	Cancel Timer Test	Pause Timer상태에서 cancel 버튼을 입력 받았을 때 Timer가 초기화되는지 test	8. Cancel Timer	R 2.6	Pass
8-2	Cancel Timer Test	Timer 가 설정되어 있지 않을 때 버튼을 입력 받아도 아무런 동작을 하지 않는지 test	8. Cancel Timer	R 2.6	Pass
8-3	Cancel Timer Test	Timer 가 설정된 후 시작 하지 않은 상태에서 버튼을 입력 받아도 아무런 동작을 하지 않는지 test	8. Cancel Timer	R 2.6	Pass

System Test Cases

Test	Test항목	Description	Use Case	System Function	Result
9	Show Alarm Test	화면이 Alarm mode로 전환 되었을 때 User가 이전 Alarm Mode를 벗어나기 전 마지막으로 본 Alarm이 표시되는지 test	Show Alarm	R3.1	Pass
10	Change Alarm Test	다음 Alarm 화면을 보기위해 버튼을 입력 받았을 때, 현재 Alarm의 다음 Alarm이 화면에 표시되는지 test 현재 Display중인 Alarm이 순서 상 마지막 Alarm일 때 첫번째 Alarm을 마지막 Alarm 다음 Alarm으로 Display되는지 확인	Change Alarm	R3.2	Pass
11	Turn On/Off Alarm Test	Alarm을 키거나 끄기위해 버튼을 눌렀을 때, 현재 Display중인 Alarm이 켜져있다면, Alarm을 끄는지 확인 현재 Display중인 Alarm이 꺼져있다면, Alarm을 키는지 확인 Alarm을 끄거나 킨 내용이 Display에 반영되는지 확인	Turn On/Off Alarm	R3.3	Pass
12	Set Alarm Test	Alarm 시간을 수정하기 위해 버튼을 눌렀을 때, 현재 보고있는 알람 시간이 수정되는지 test 알람시간 수정버튼을 눌렀을 때 설정한 Alarm이 제대로 설정되었는지 test	Set Alarm	R3.4	Pass
13	Return to Default Screen Test	Set 모드에 있을 때 입력을 받지 않은 채로 일정 시간이 지나면 각 모드의 default 화면으로 전환되는지 test	Return to Default Screen	R10.1	Pass
14	Show Stopwatch Test	Stopwatch Mode가 잘 Display 되는 지 Test	Show Stopwatch	R4.1	Pass
15	Start Stopwatch Test	Stopwatch가 시작되지 않은 상태일 때 시작하기 위해서 버튼을 눌렀을 때 Stopwatch가 시작되어 countup되는지 test	Start Stopwatch	R4.2	Pass

System Test Cases

Test	Test항목	Description	Use Case	System Function	Result
16	Pause Stopwatch Test	Stopwatch가 동작 중인 상태에서 버튼을 입력 받았을 때, 시간 계산을 일시정지 하고, 일시정지된 시간이 올바르게 Display 되는지 Test	Pause Stopwatch	R4.3	Pass
17	Resume Stopwatch Test	Stopwatch가 멈춰 있는 상태에서 버튼을 입력 받았을 때, 일시정지 되어있던 Stopwatch가 정상적으로 재시작되는지 test	Resume Stopwatch	R4.4	Pass
18	Reset Stopwatch Test	Stopwatch가 일시정지 된 상태에서 버튼을 입력 받았을 때, Stopwatch가 초기화 되는 지 Test	Reset Stopwatch	R4.5	Pass
19	Lap Stopwatch	버튼을 입력 받았을 때, 화면에 Laptime을 올바르게 Display 하는지 test	Lap Stopwatch	R4.6	Pass
20	Show World Time Test	World Time Mode를 잘 Display하는 지 Test	Show World Time	R5.1	Pass
21	Change Timezone test	Timezone이 순차적으로 바뀌는지 Test	Change Timezone	R5.2	Pass
22	Show Calorie check test	Calorie Check Mode를 잘 Display 하는지 Test	Show Calorie Check	R6.1	Pass
23	Set Speed and Weight test	설정한 값이 Calorie Check에 올바르게 반영되는지 Test	Set Speed and Weight	R6.2	Pass
24	Start Calorie Check test	Calorie Check를 시작하고, 올바르게 Calorie를 계산하여 화면에 Display 하는지 Test	Start Calorie Check	R6.3	Pass
25	Pause Calorie Check test	Calorie Check 중인 상태에서 버튼을 입력 받았을 때 Calorie Check가 일시정지 되고, 계산중이었던 Calorie가 화면에 잘 Display 되는지 Test	Pause Calorie Check	R6.4	Pass
26	Resume Calorie Check test	Calorie Check를 일시정지한 상태에서 버튼을 입력받았을 때 Calorie Check가 재 시작되는지 Test	Resume Calorie Check	R6.5	Pass

System Test Cases

Test	Test항목	Description	Use case	System Function	Result
27	Reset Calorie Check test	Calorie Check를 일시정지한 상태에서, 버튼을 입력 받았을 때 Calorie Check가 초기화 되는지 Test	Reset Calorie Check	R6.6	Pass
28-1	Beep Buzz test	Alarm이 끝난 후 6가지 모드에서 Alarm Icon이 Display되고 Beep Buzz가 발생하는지 Test	Beep Buzz	R9.1	Pass
28-2	Beep Buzz test	Timer가 끝난 후 6가지 모드에서 Display되고 Beep Buzz가 발생하는지 Test	Beep Buzz	R9.1	Pass
29-1	Stop Buzz test	-Buzzer가 울리고 있고 Alarm Icon이 Display되고 있으면 어떤 버튼이 입력되든 Buzzer를 멈추고 Alarm Icon이 감춰지는지 Test.	Stop Buzz	R9.2	Pass
29-2	Stop Buzz test	-Buzzer가 울리고 있고 Alarm Icon이 Display되지 않았을 때 어떤 버튼이 입력되든 Buzzer가 멈춰지는지 Test.	Stop Buzz	R9.2	Pass
30	Change Mode test	모드가 순차적으로 바뀌는 지 Test	Change Mode	R7.1	Pass
31	Set Mode test	-Activate / Deactivate 설정 후 Activate 한 모드만 표시 되는지 Test	Set Mode	R8.1	Pass

시연

6

시연

Thank you