

살빼 시계

Presentation 4: OOPT STAGE 2050/2060

Team2

박성호, 박동현, 천민수, 조윤직, 안찬우

Index

: Construction Phase

1

Design & Construct

2

Implement Window

3

Write Test Code & Unit Testing

4

Testing Traceability Analysis

5

System Test Cases

6

시연



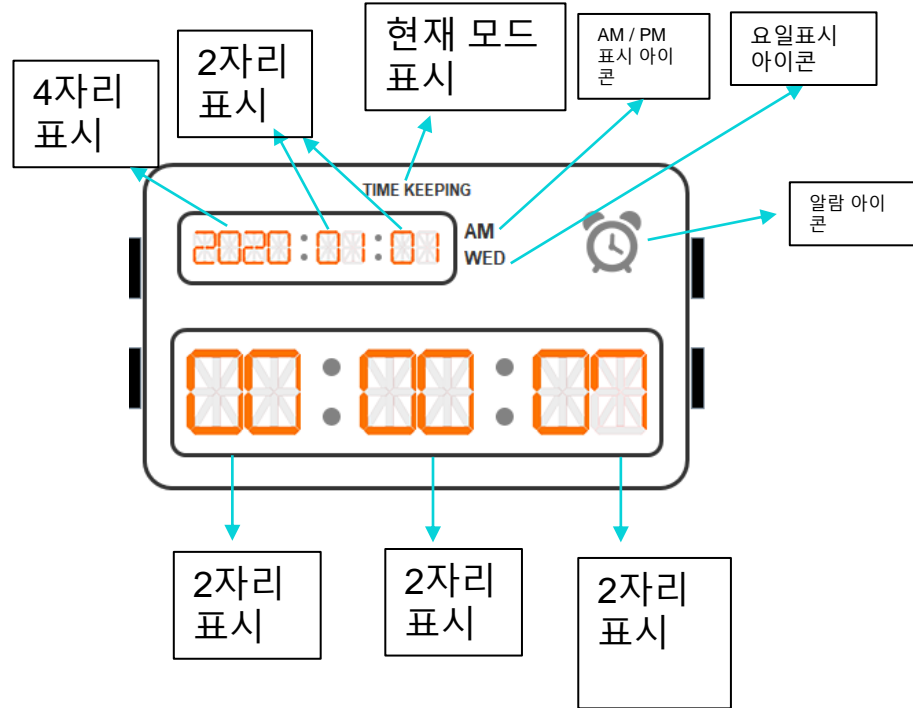
Design & Construct

1

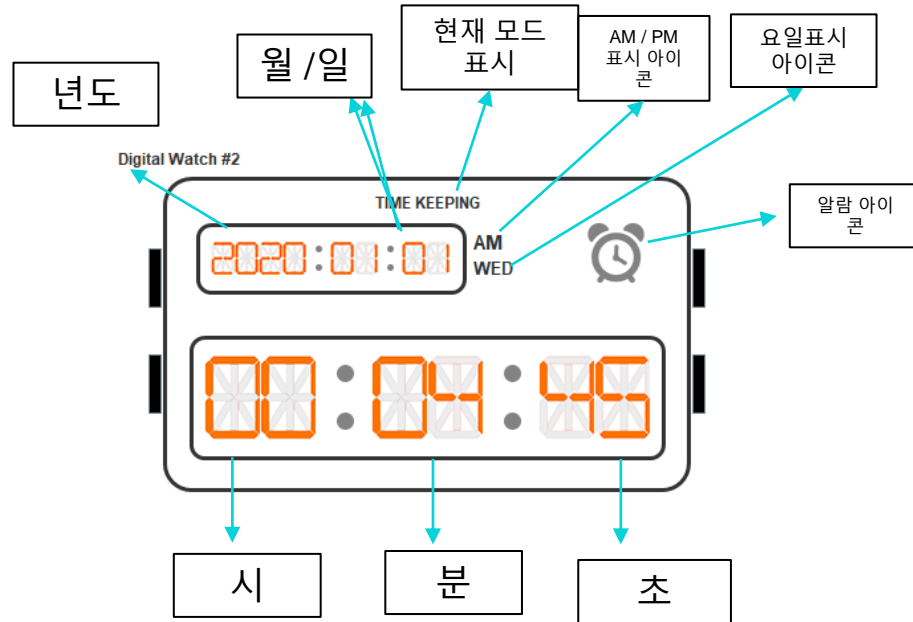




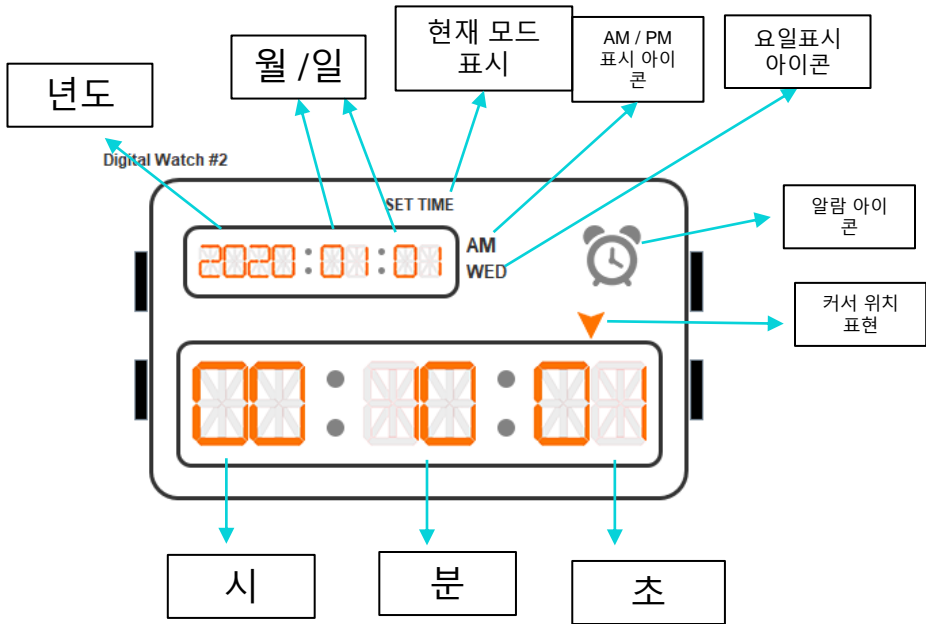
Define Real Use Case: Overall UI



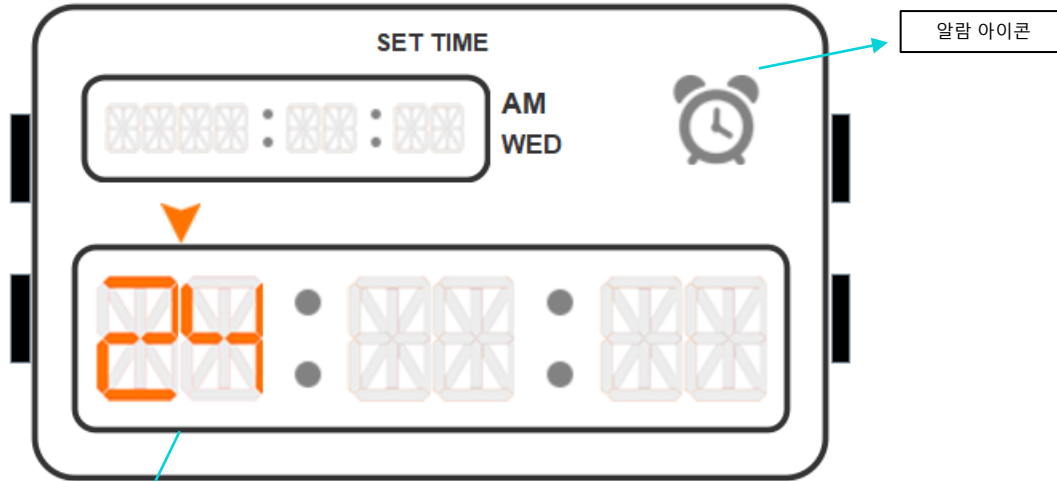
TimeKeeping



Set Time



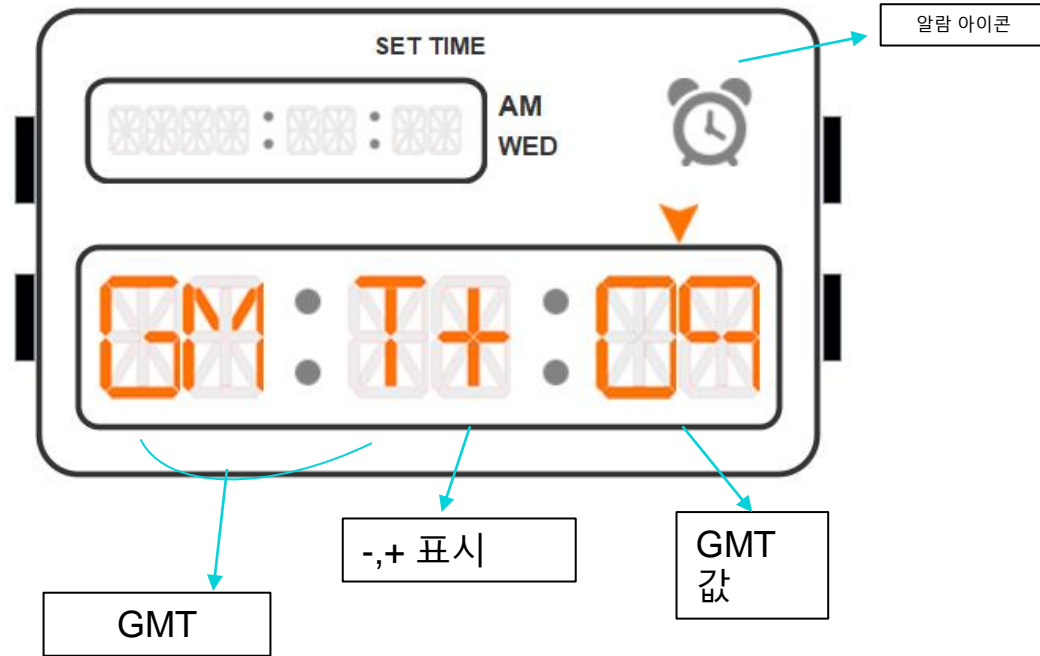
Set Time – 12/24H



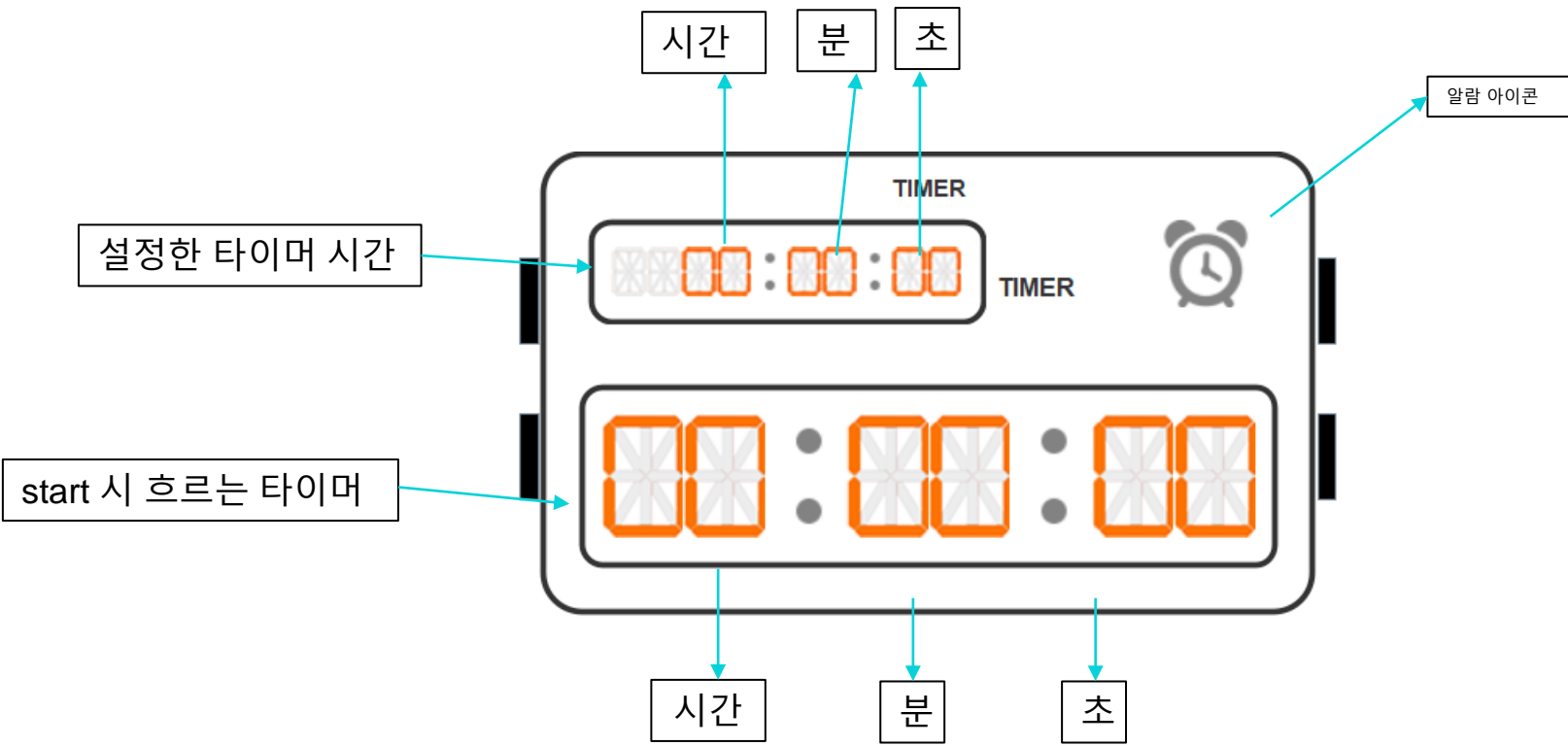
알람 아이콘

12, 24시간 방식 표기

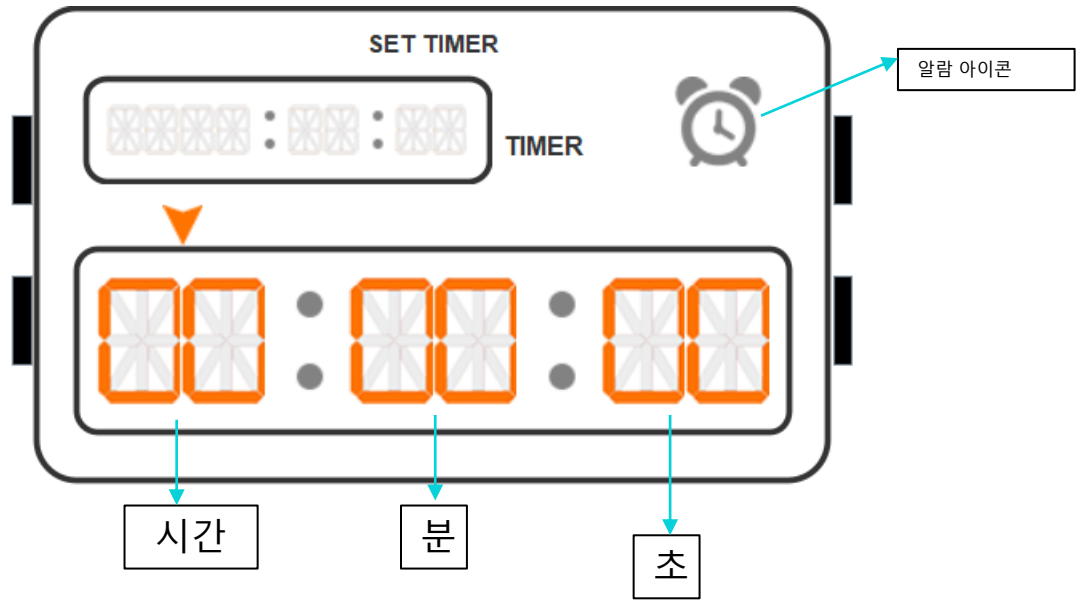
Set Time - GMT



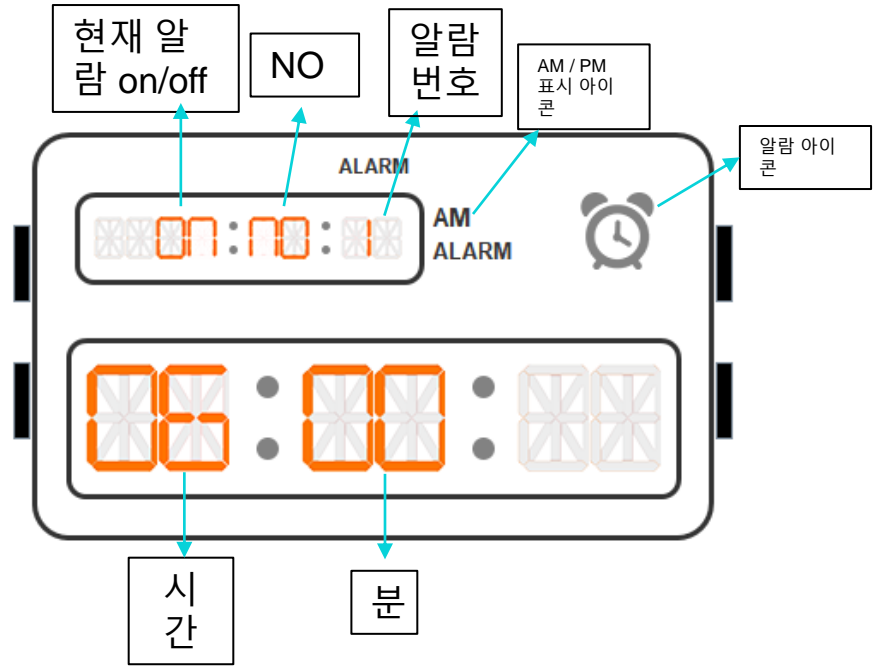
Show Timer



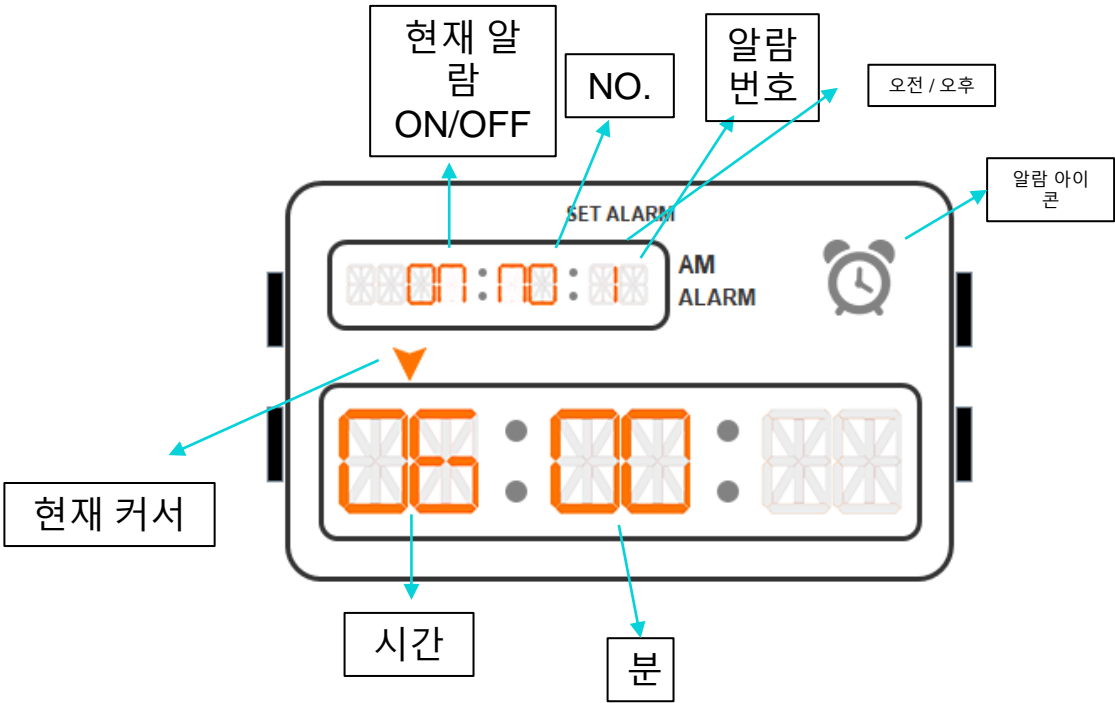
Set Timer



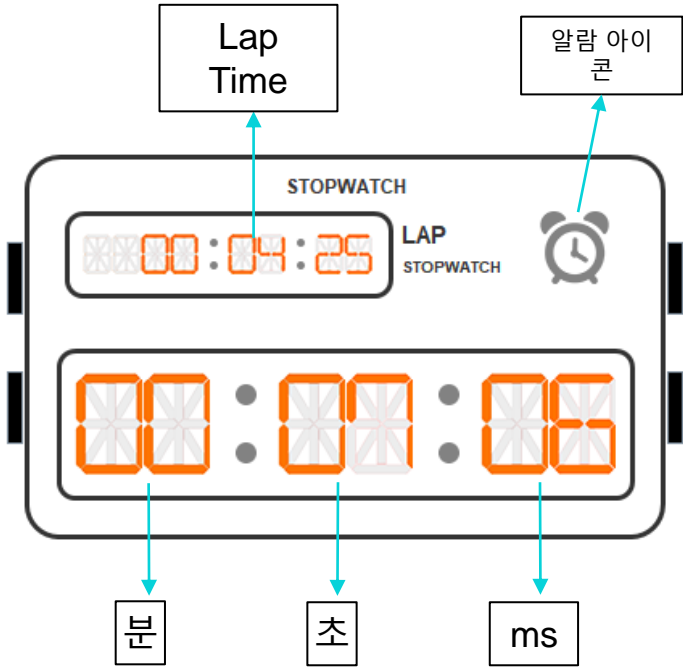
Show Alarm



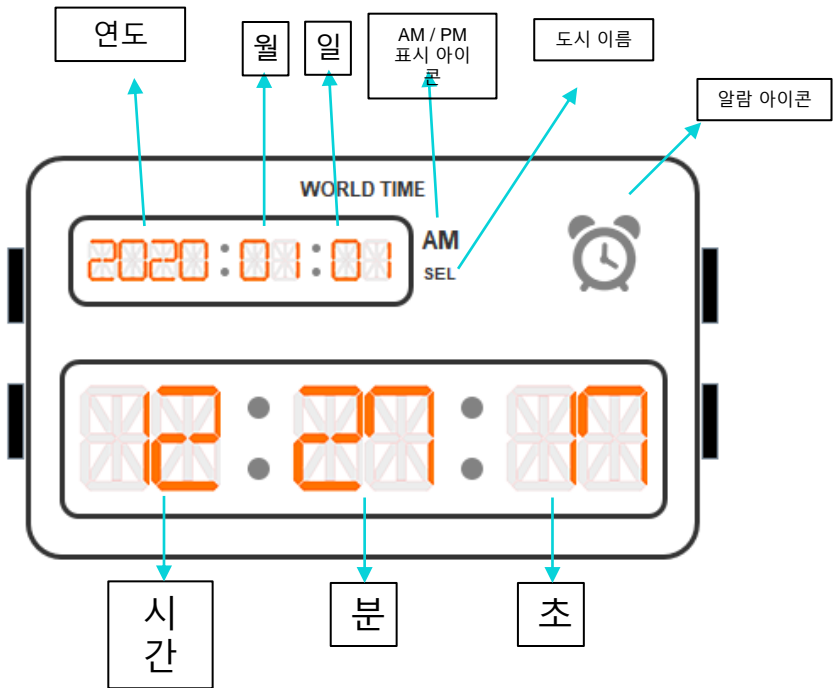
Set Alarm



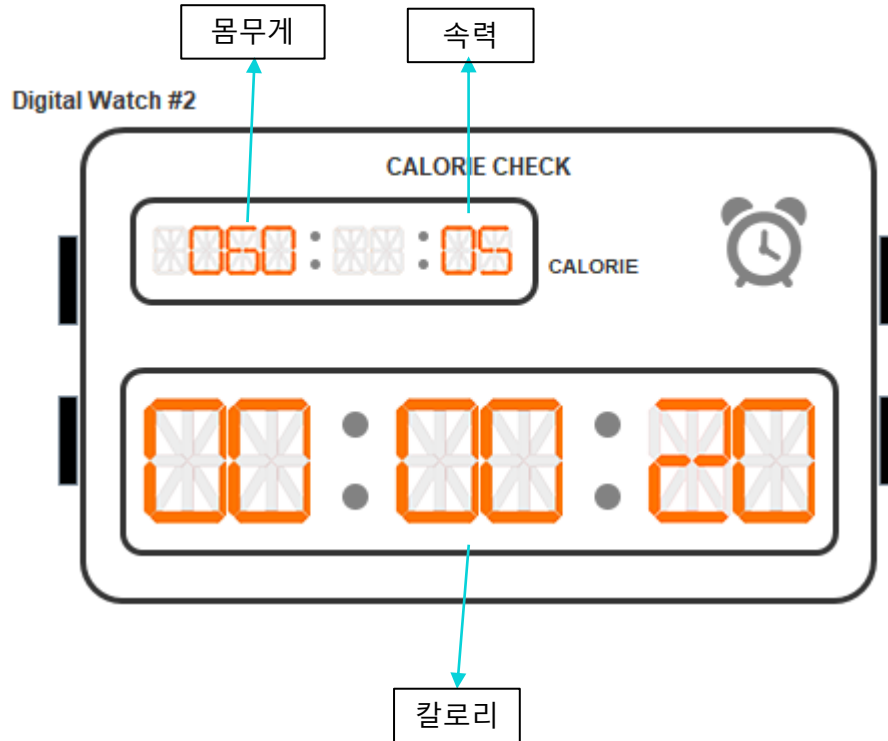
Stopwatch



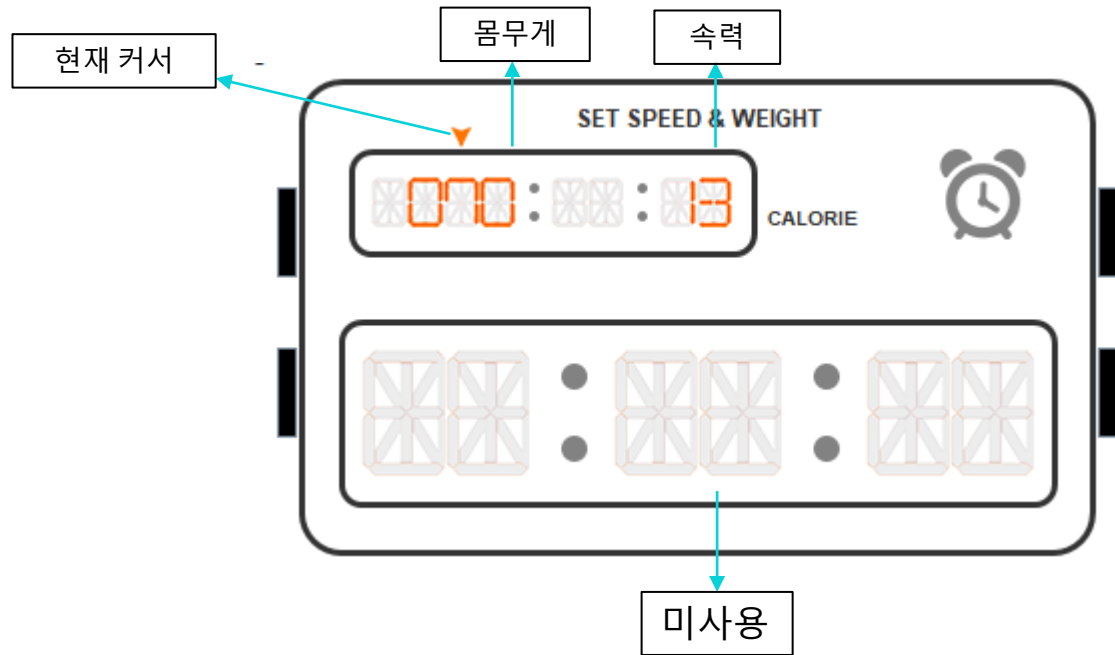
World Time



Calorie Check



Set Speed and Weight





Set Mode

ALRM : 알람

TIME : 시간

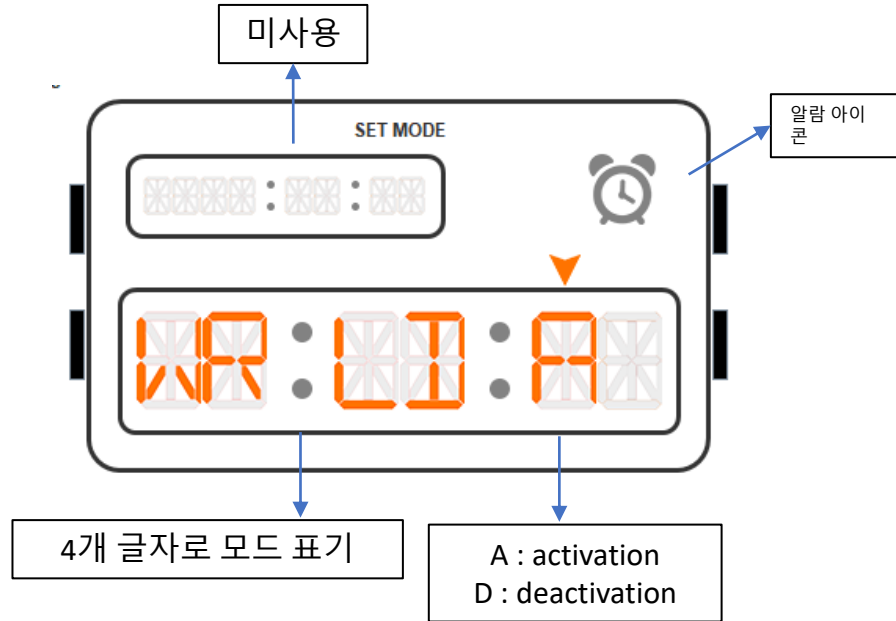
TMER : 타이머

CALO : 칼로리

STPW : 스탑워치

WRLD : 세계시간

A: activate D : deactivate



Implement Window

2

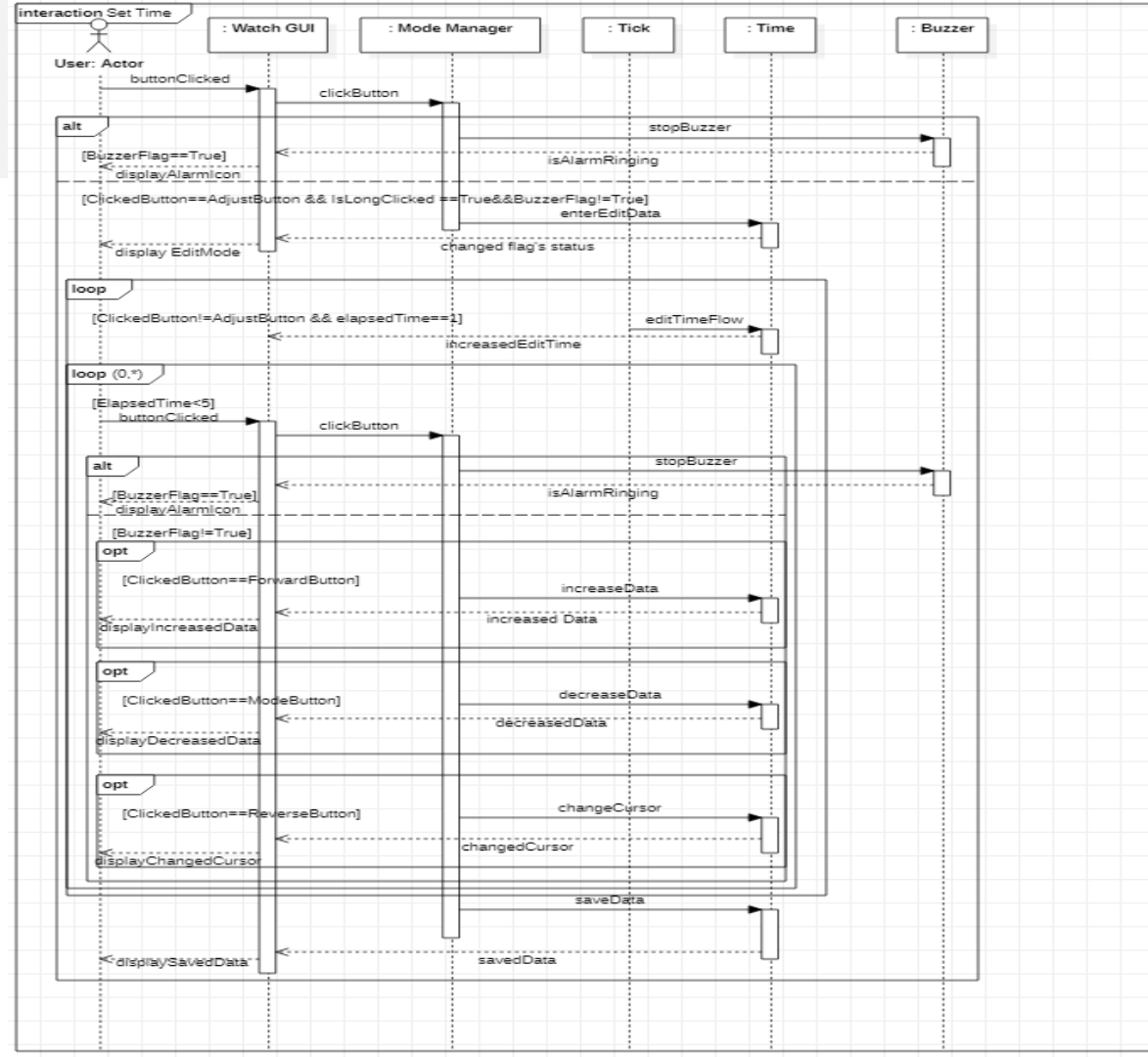


GUI Operation

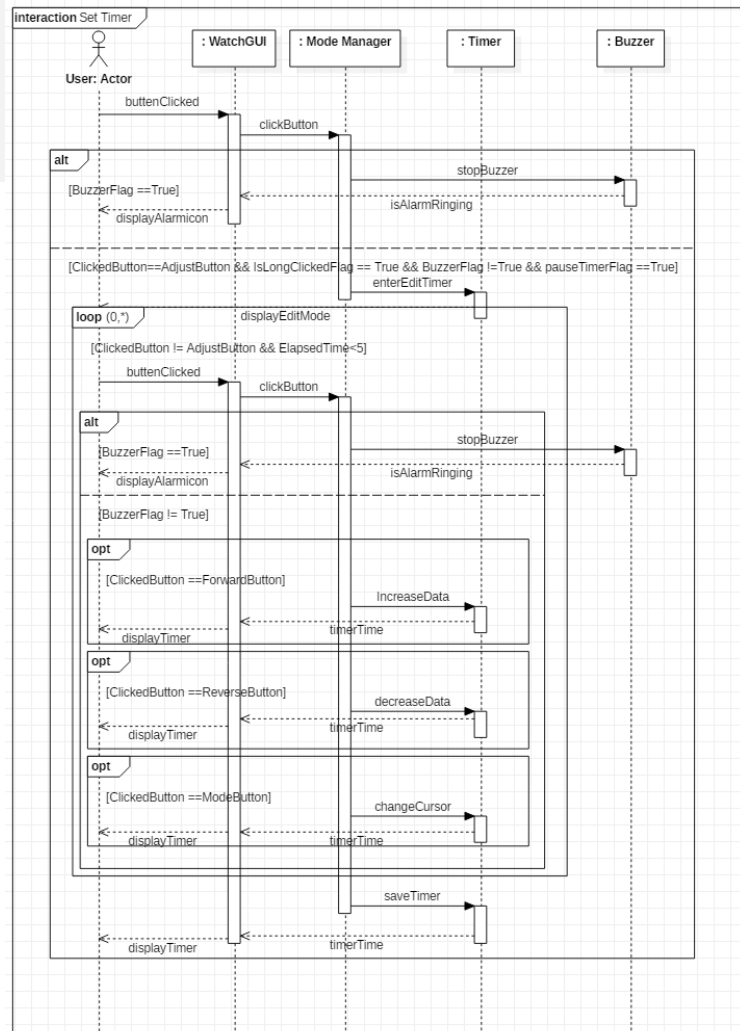


Name	buttonClicked
Responsibilities	사용자가 시계의 어느 버튼을 짧게 누른다.
Type	GUI
Cross References	All reference
Notes	이 오퍼레이션들을 가지고 있는 버튼의 식별번호와 짧게 혹은 길게 눌렀음에 대한 여부를 ModeManager의 clickedButton을 통해 시스템에 전달
Pre-Conditions	N/A
Post-Conditions	N/A

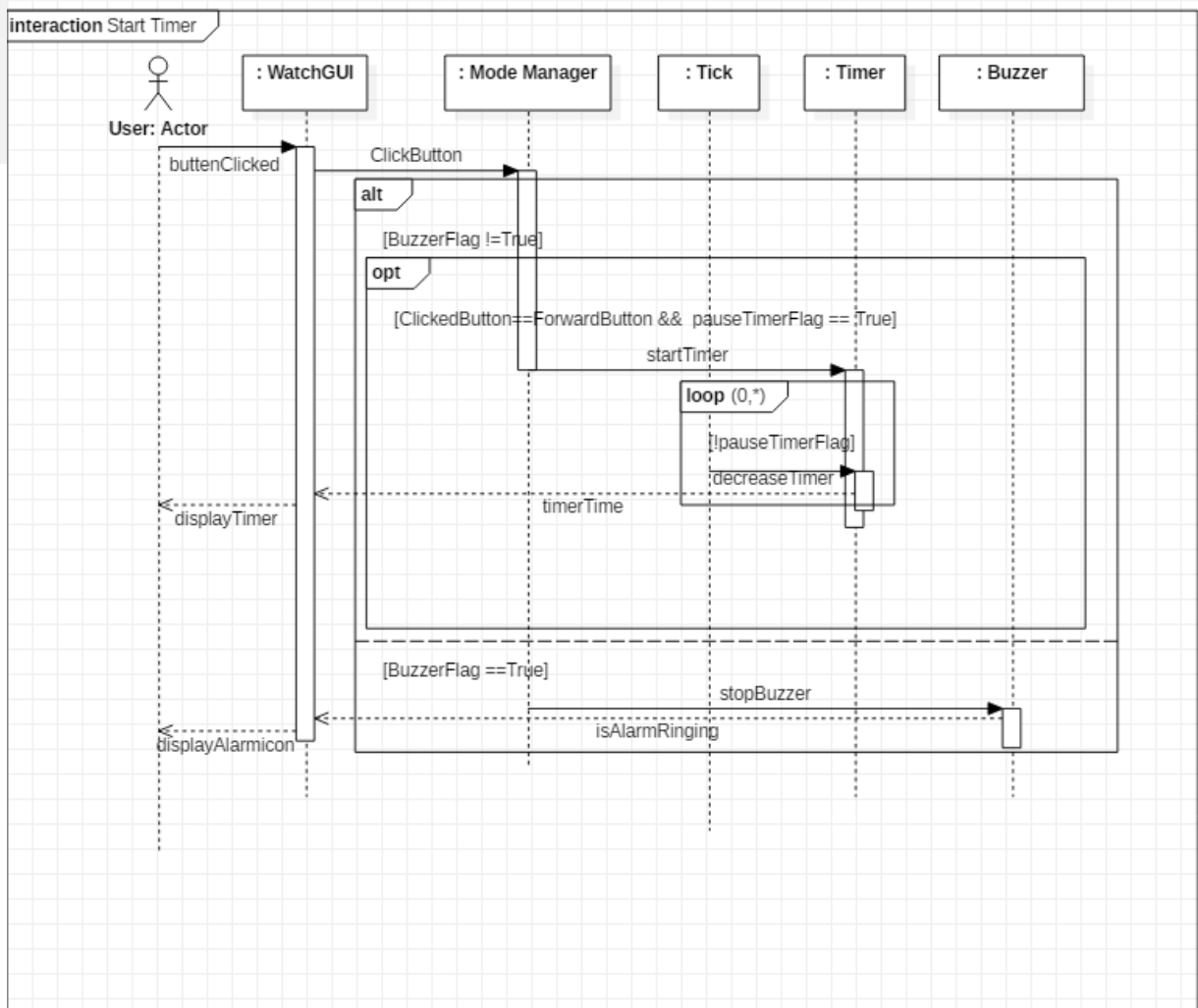
Set Time



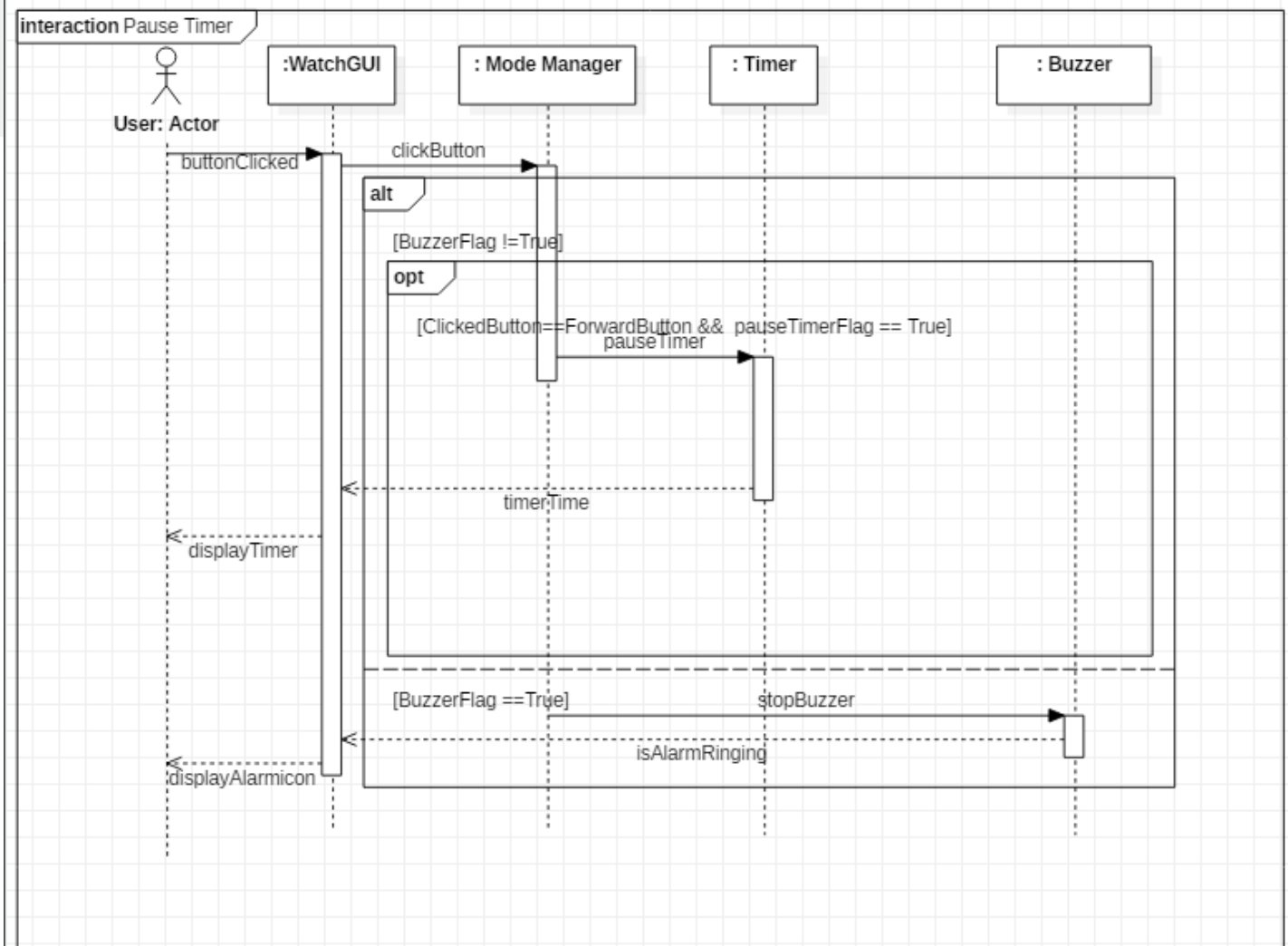
Set Timer



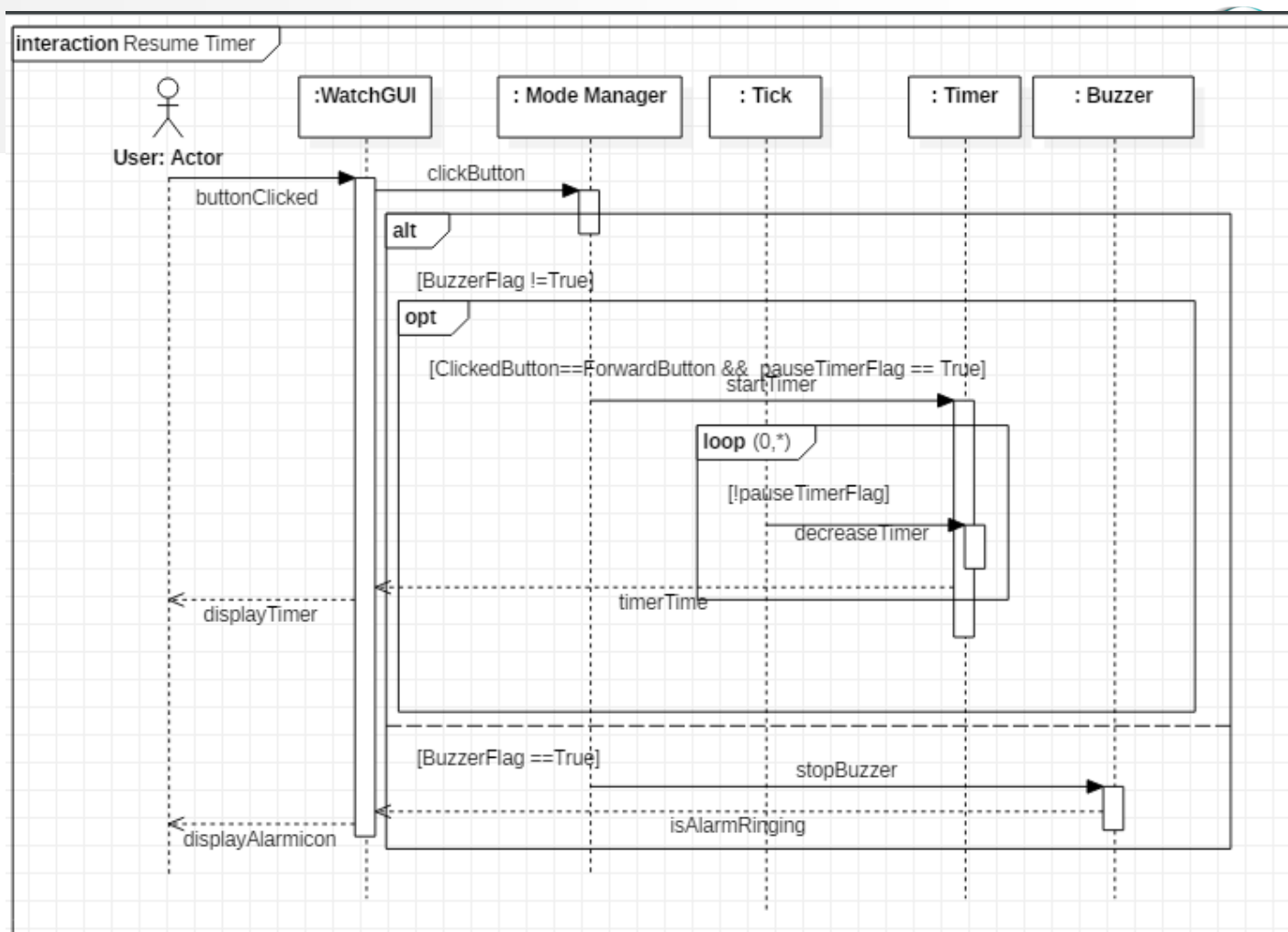
Start Timer



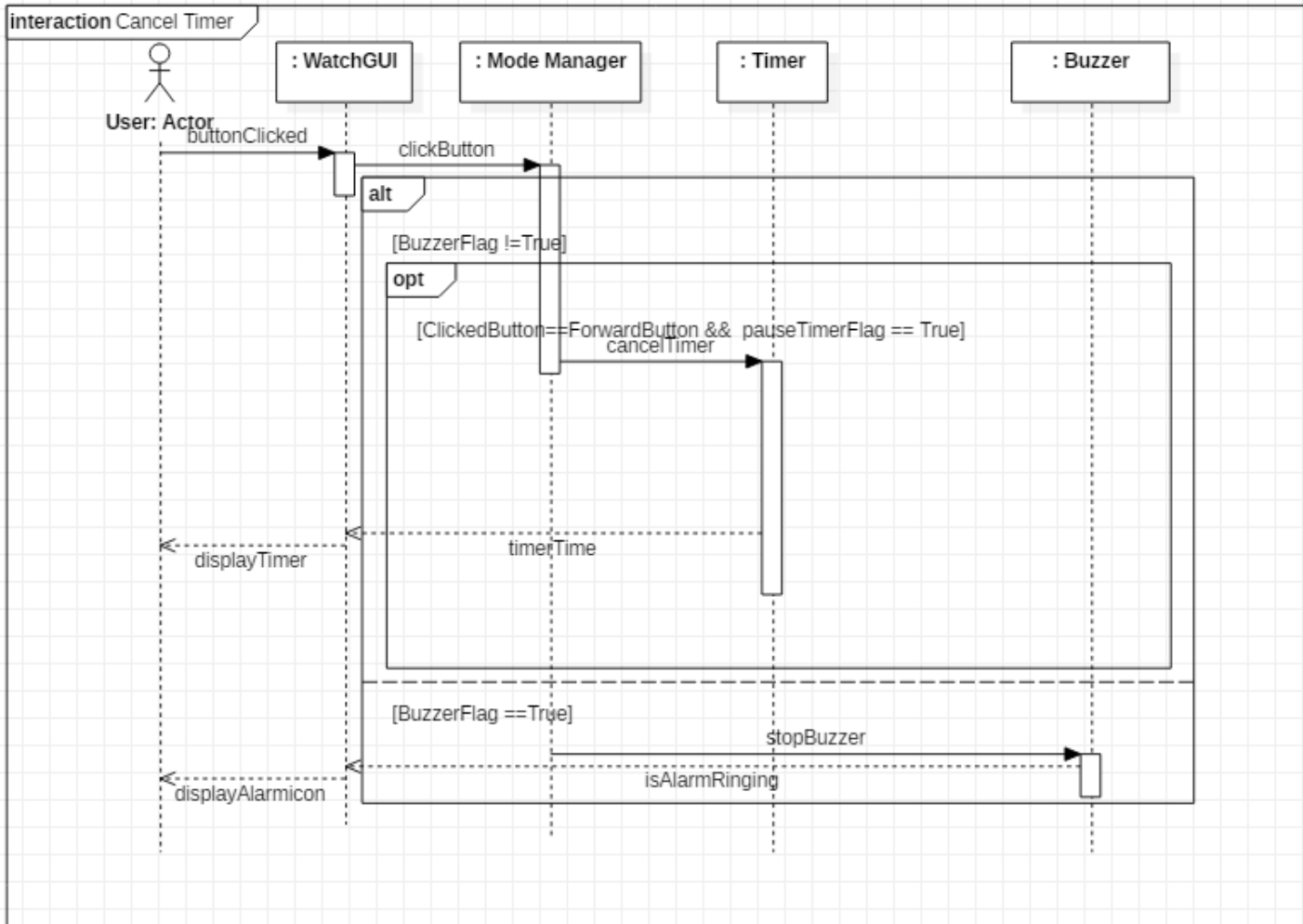
Pause Timer



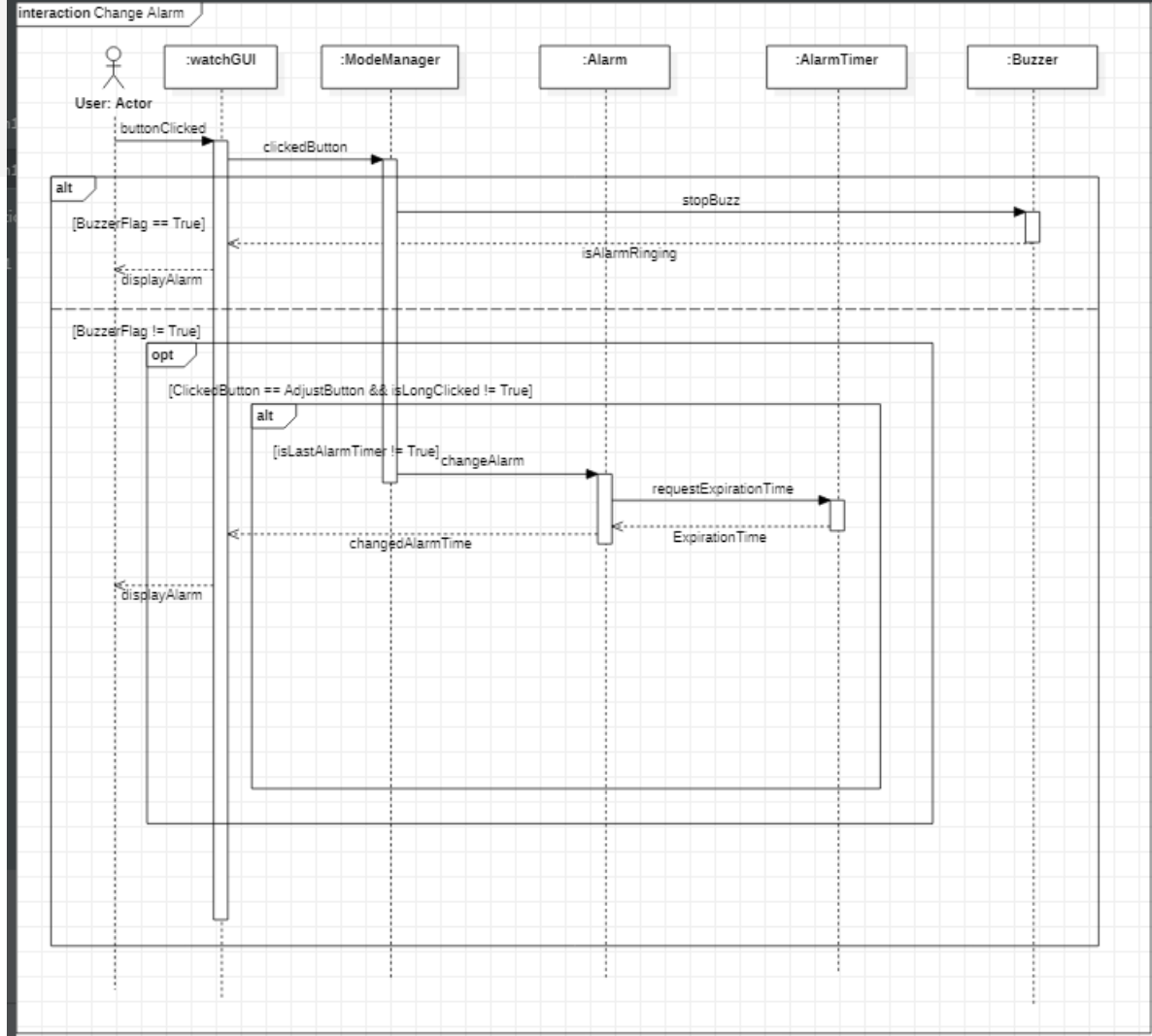
Resume Timer



Cancel Timer

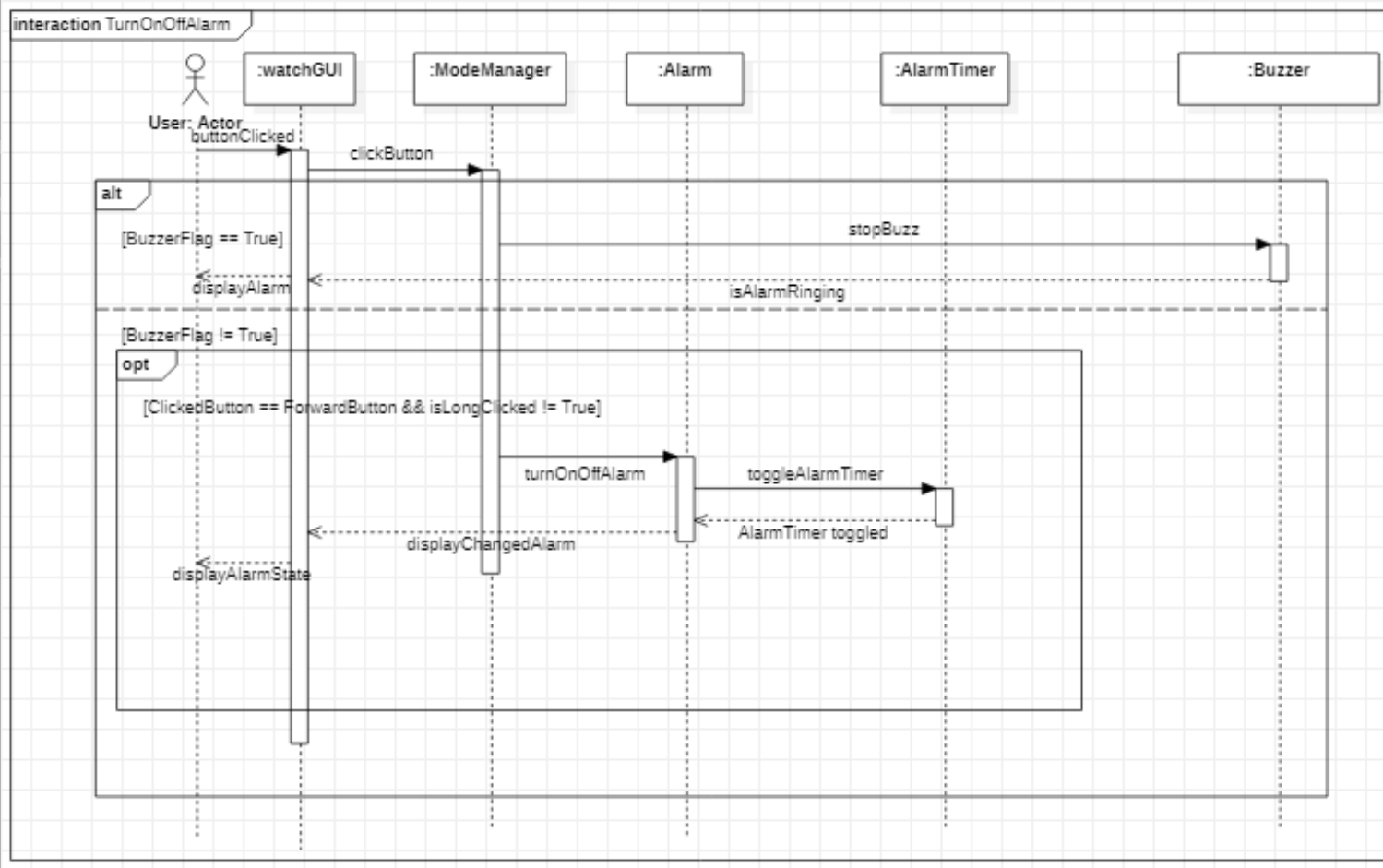


Change Alarm

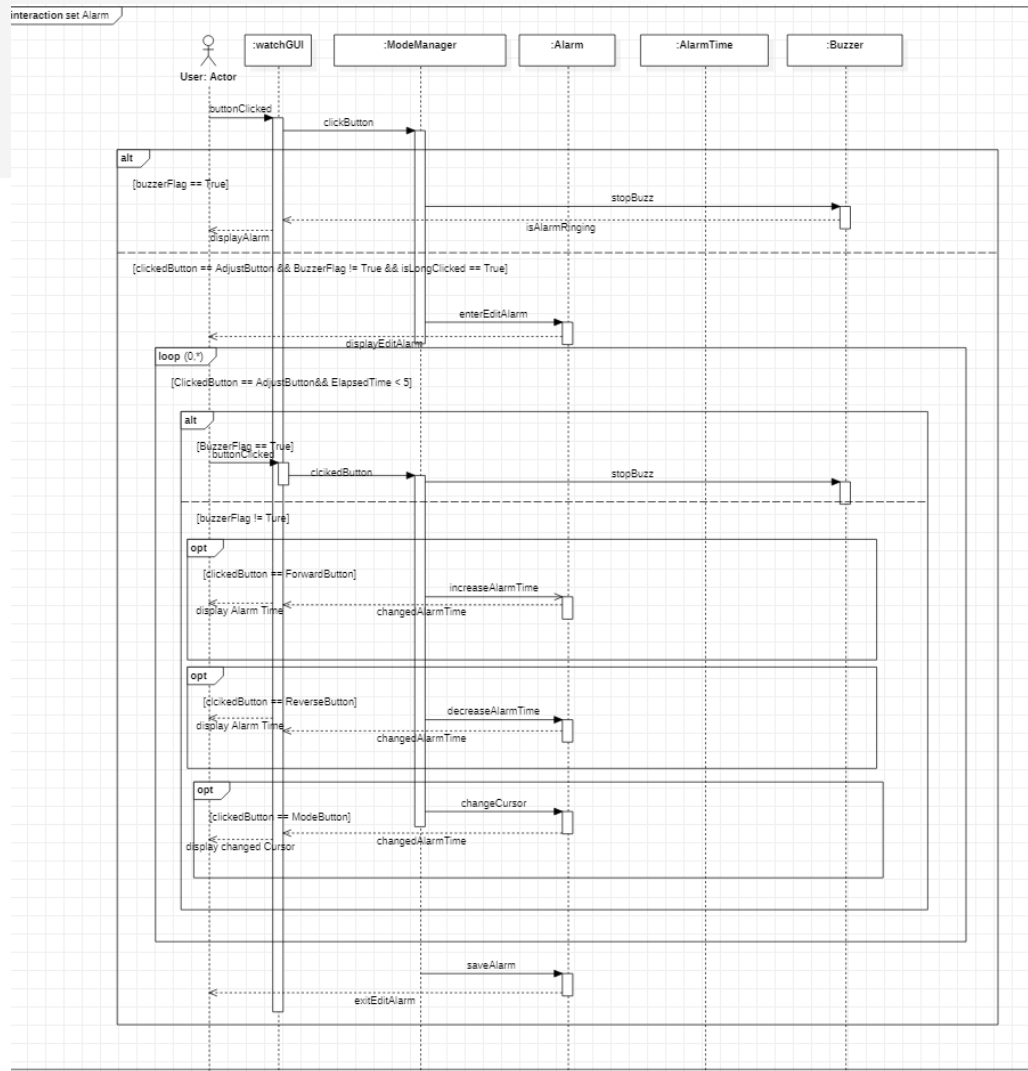




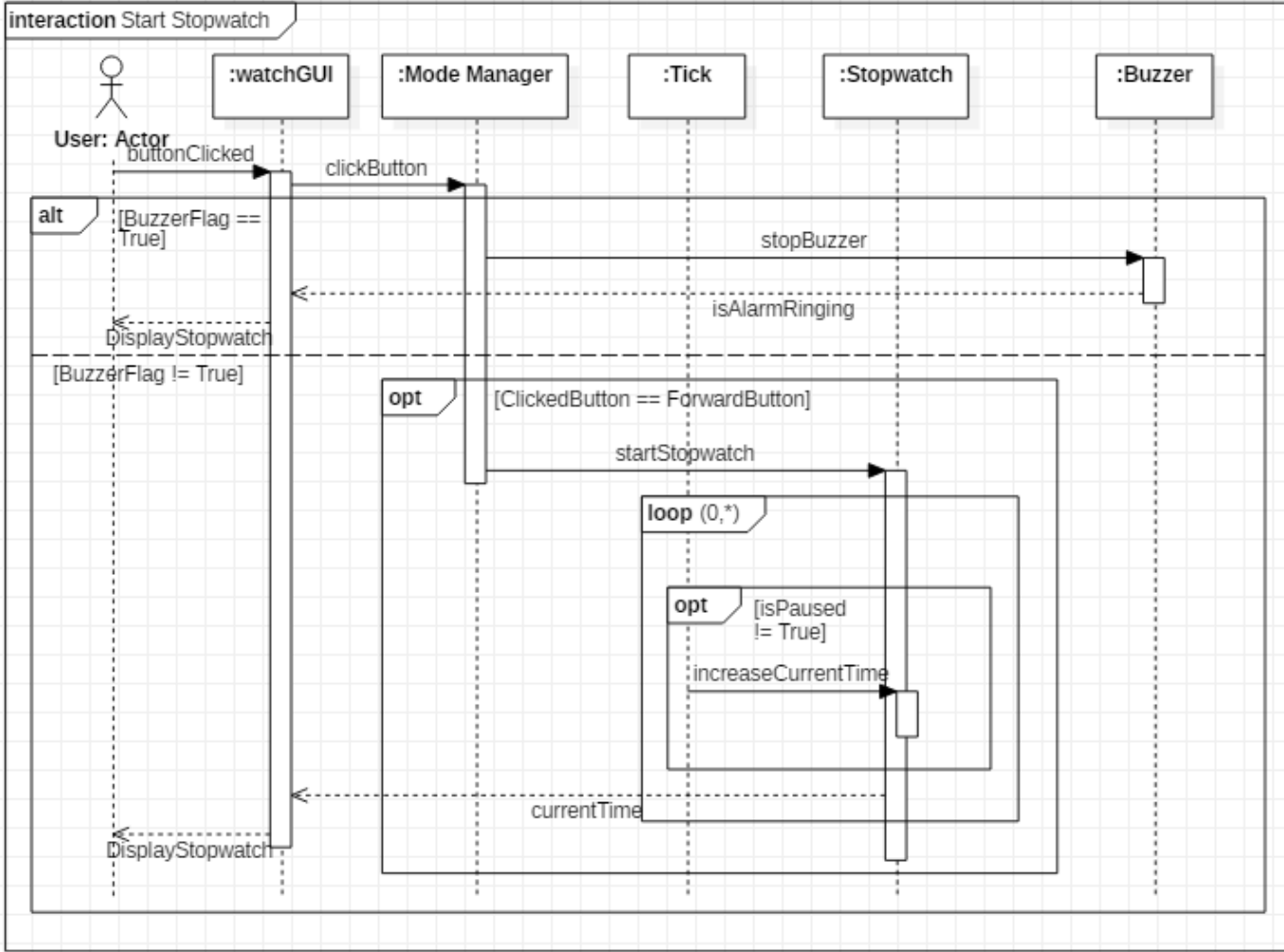
Turn On/Off Alarm



Set Alarm

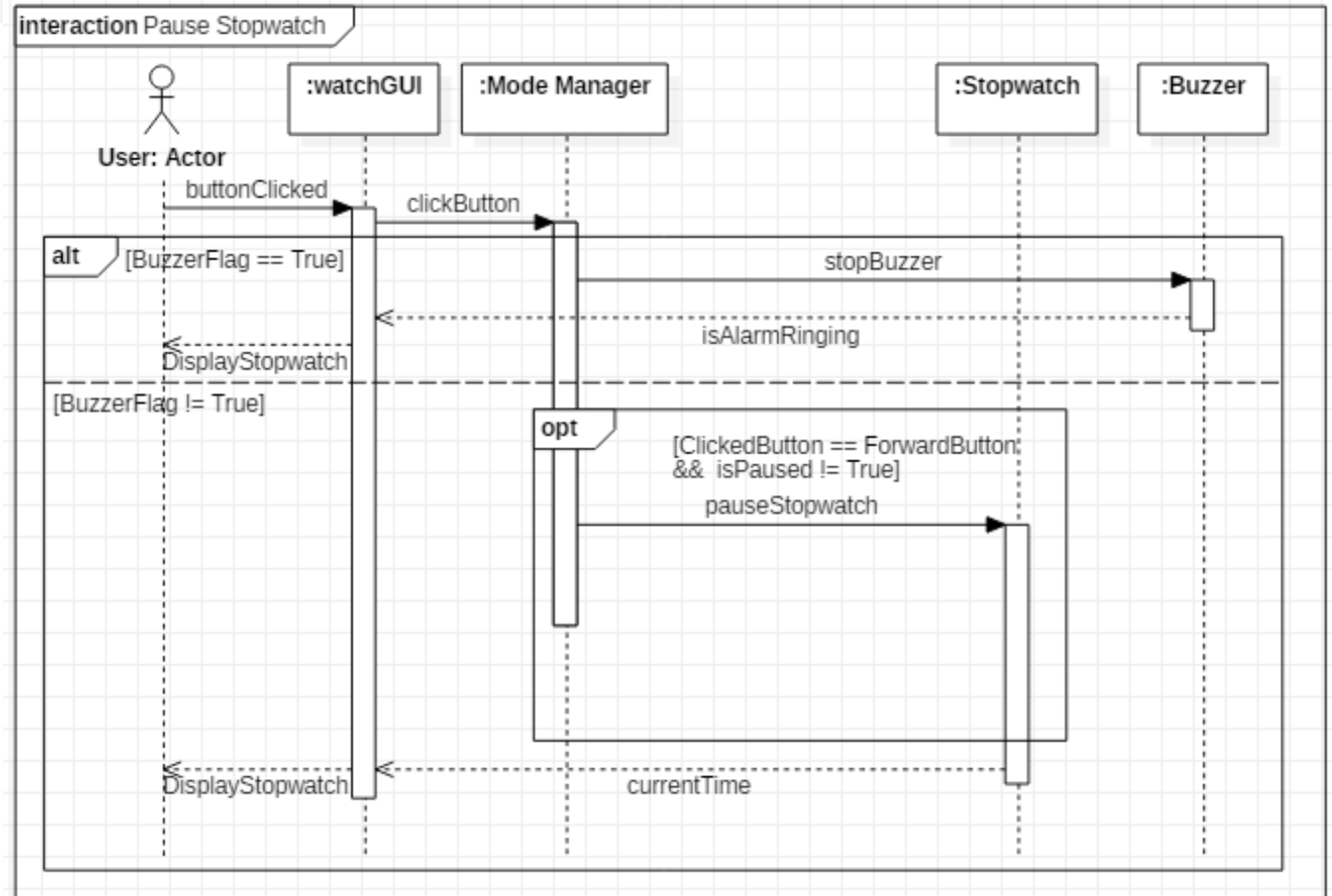


Start Stopwatch



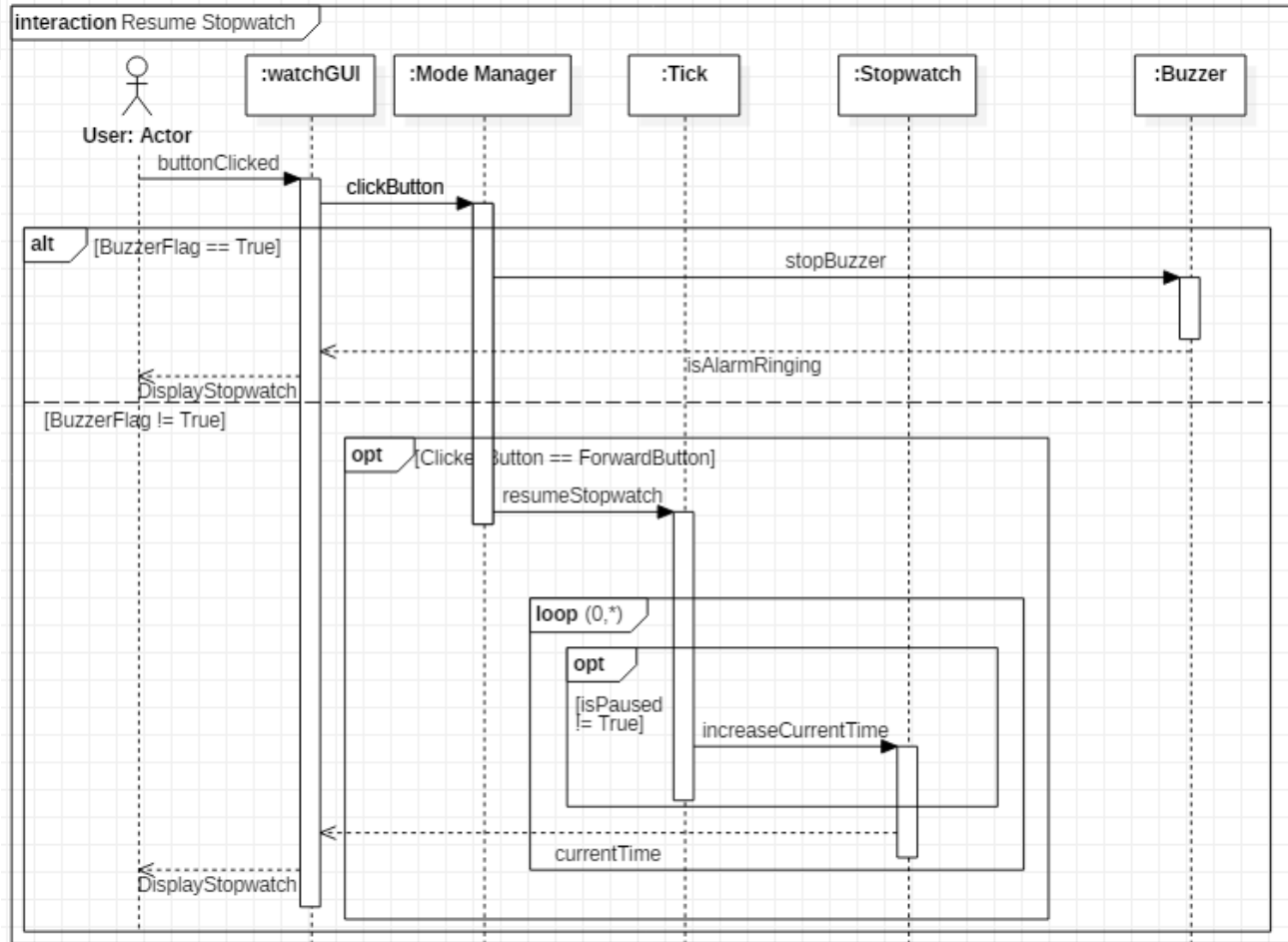


Pause Stopwatch



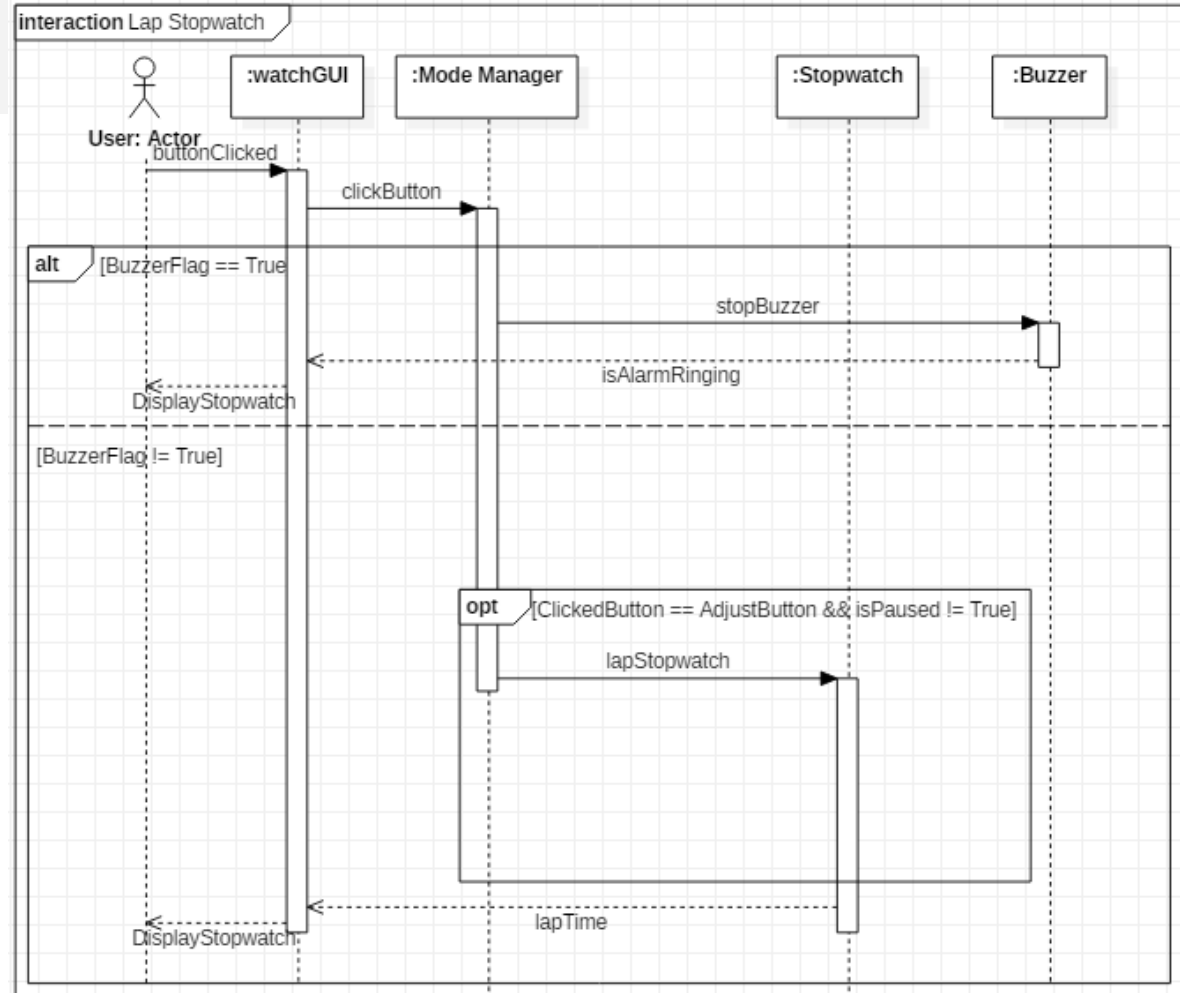


Resume Stopwatch



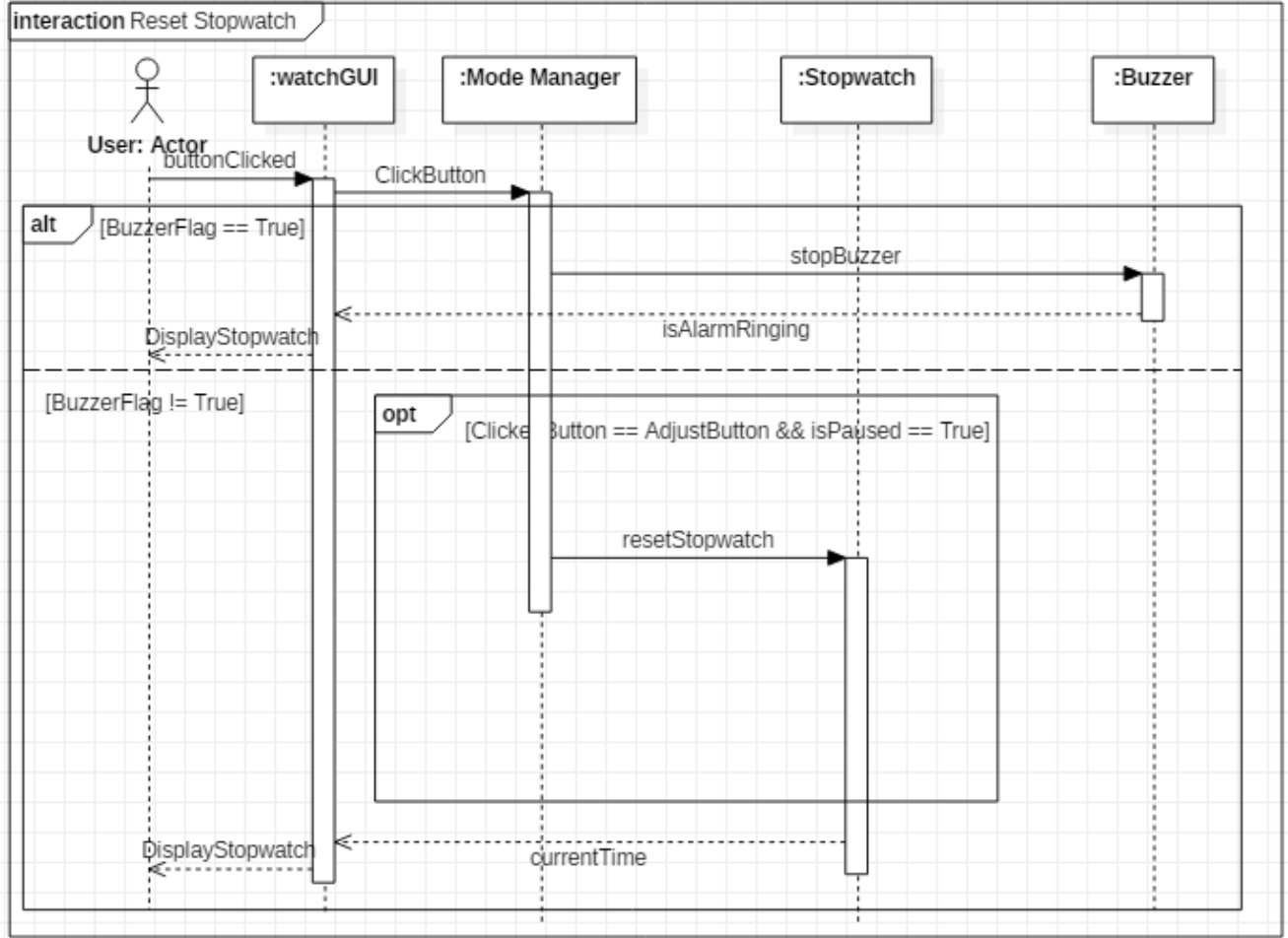


Lap Stopwatch

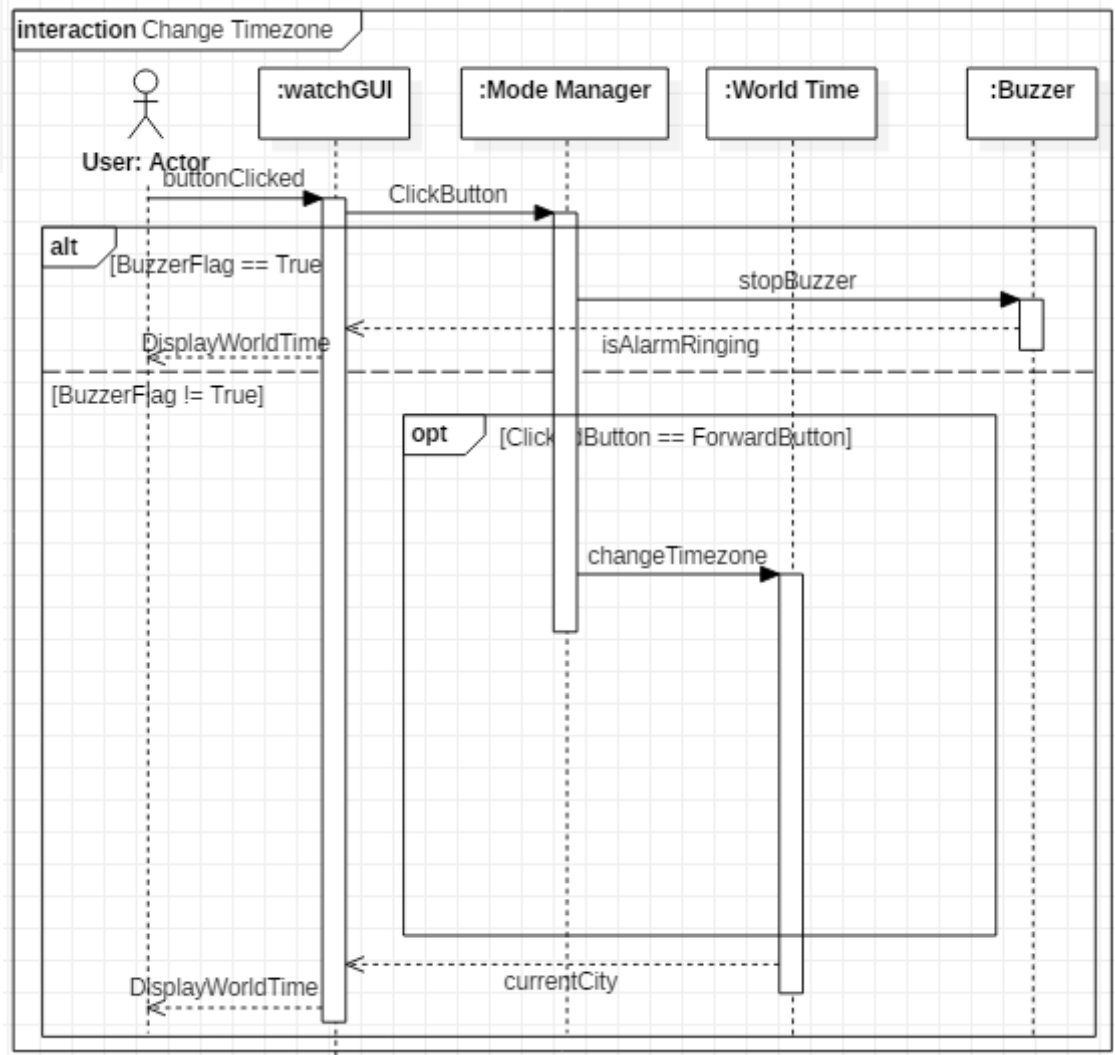




Reset Stopwatch



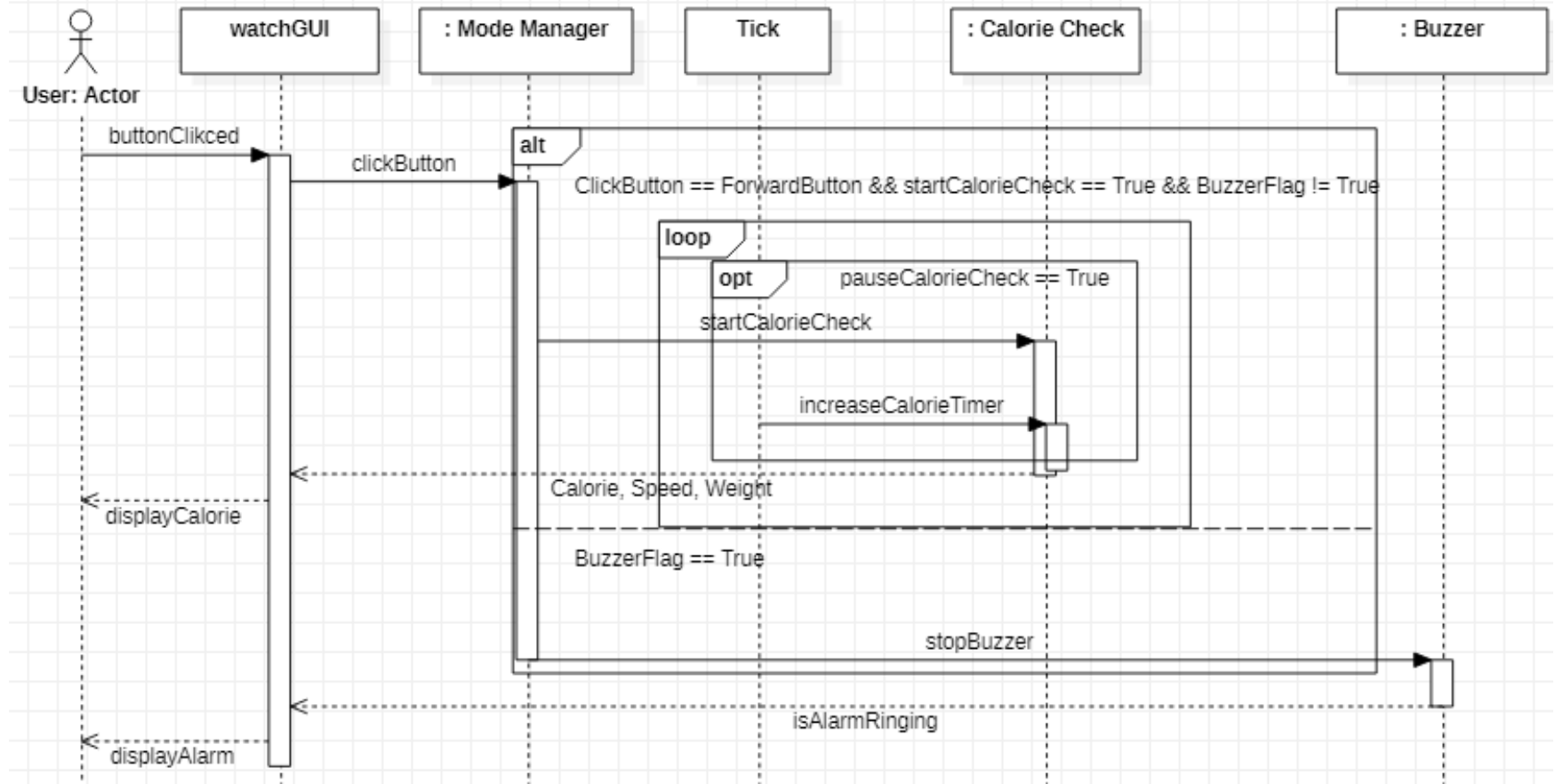
Change Timezone





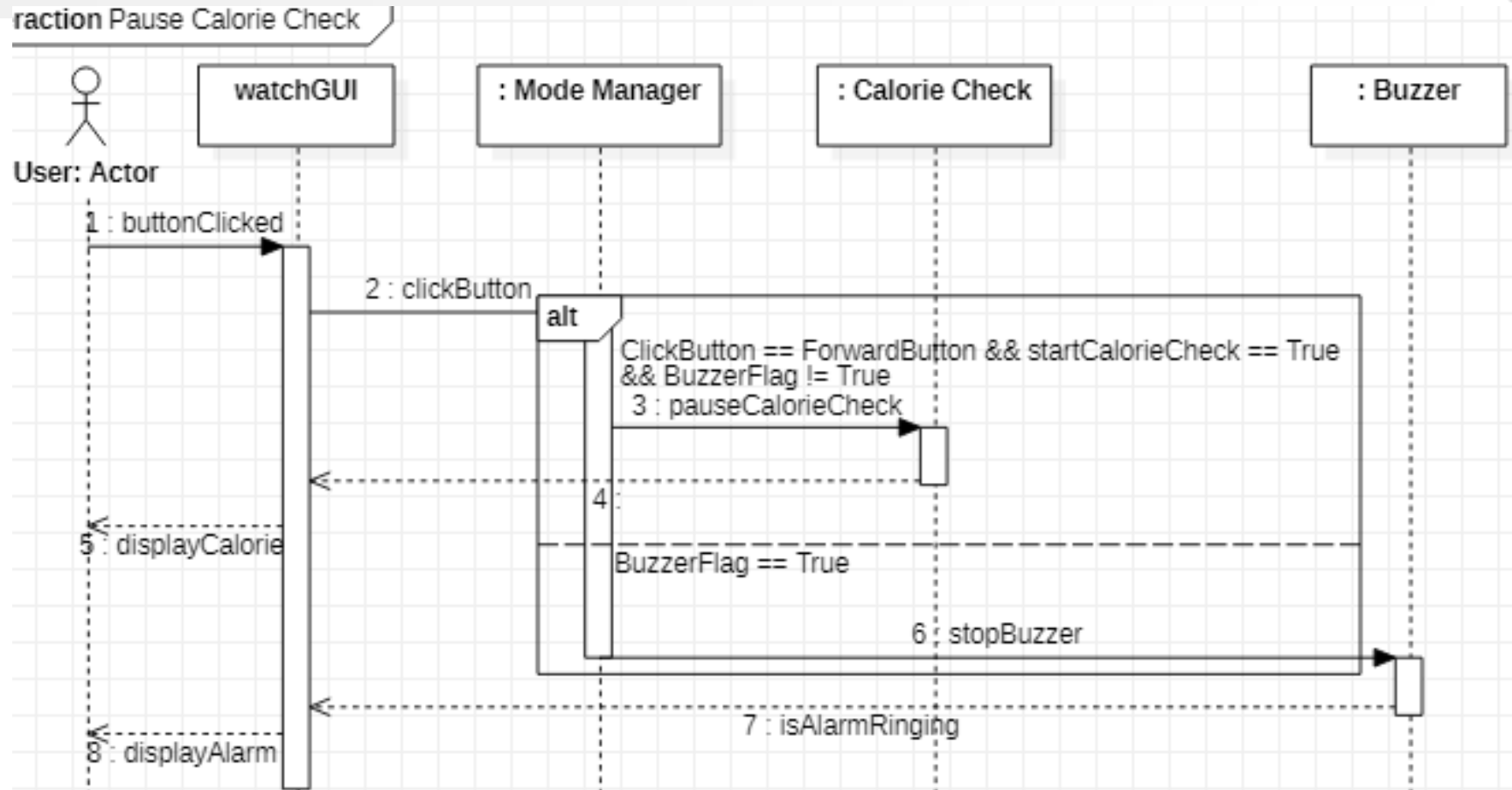
Start Calorie Check

reaction Start Calorie Check





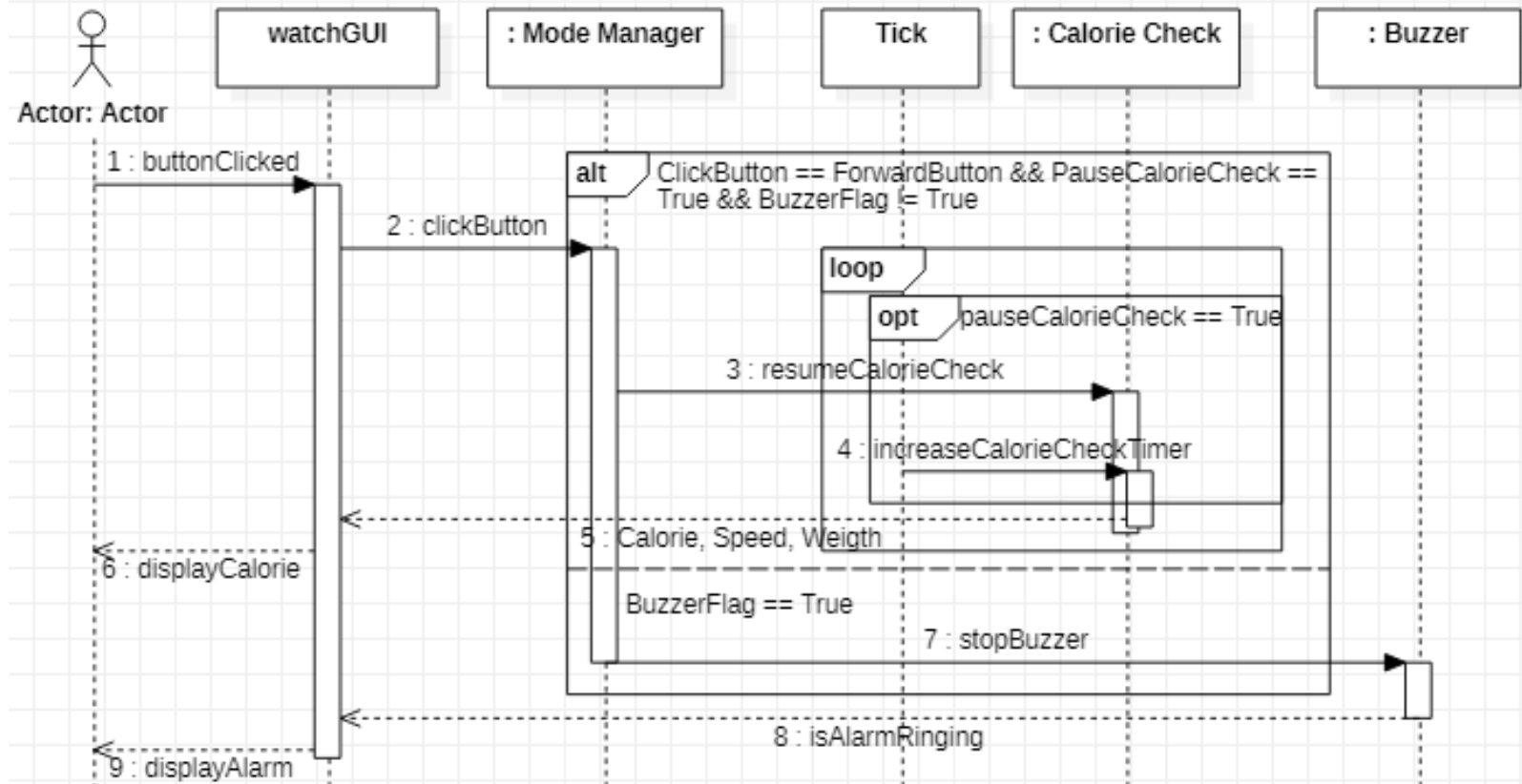
Pause Calorie Check



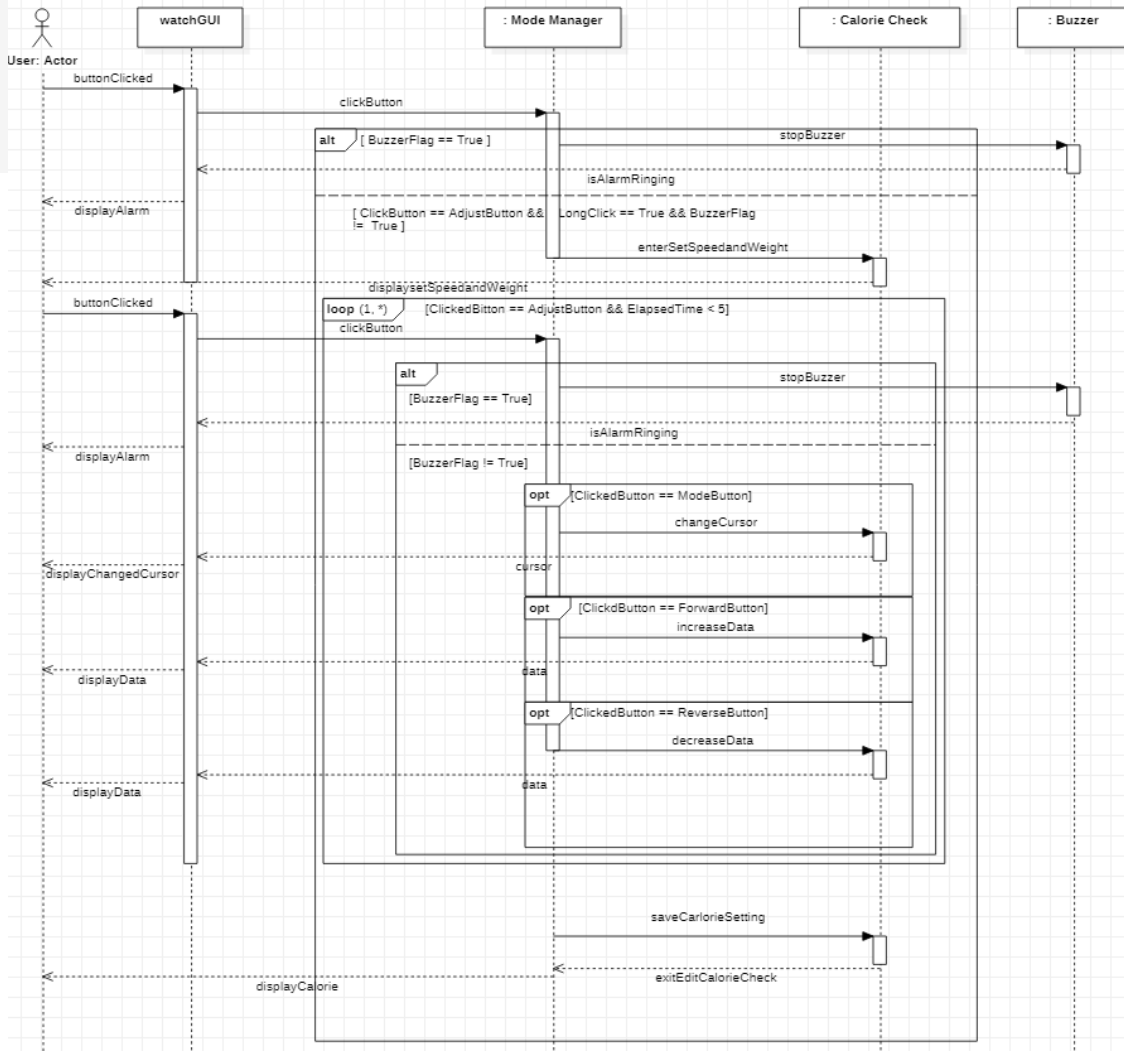


Resume Calorie Check

Interaction Resume Calorie Check



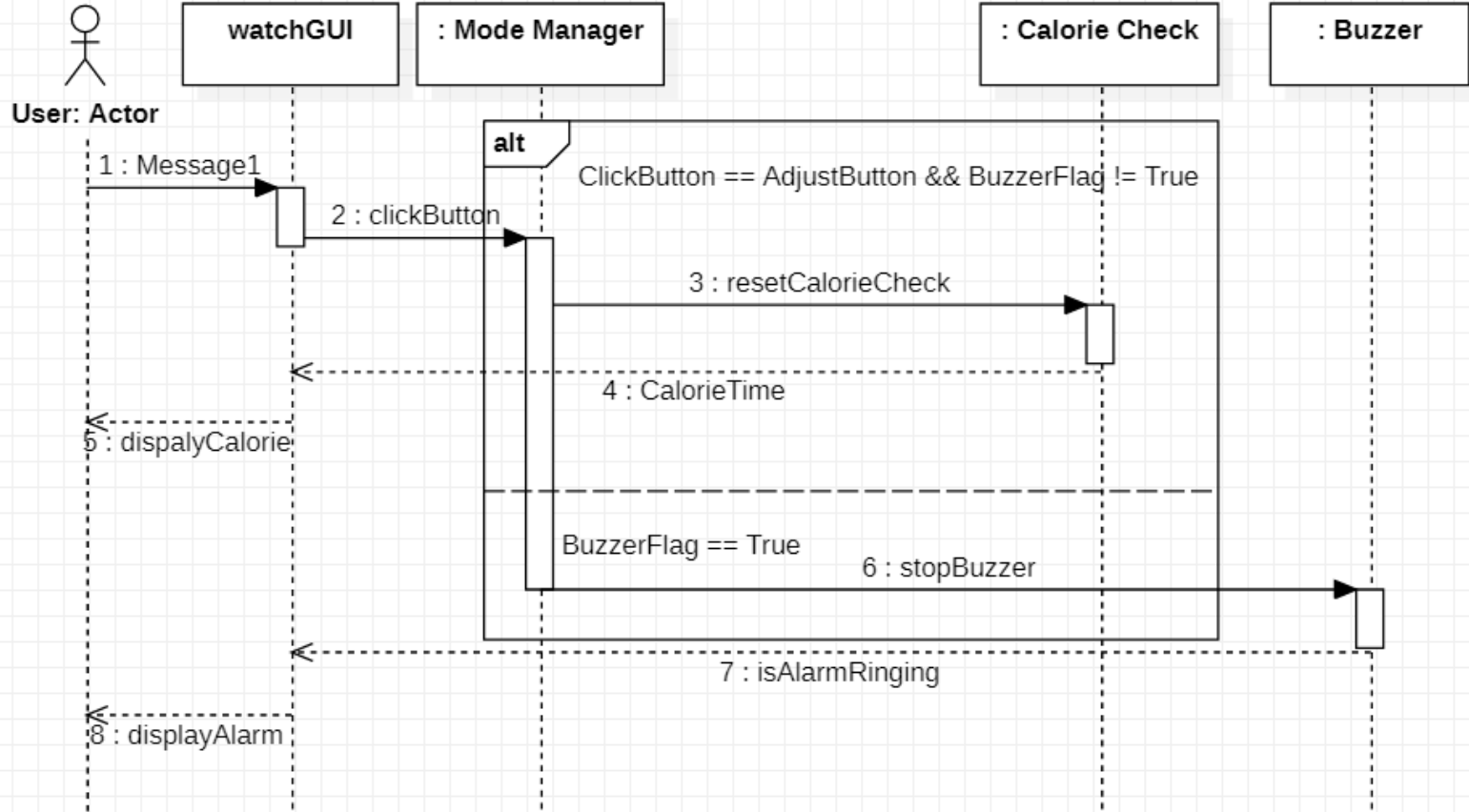
Set Speed and weight



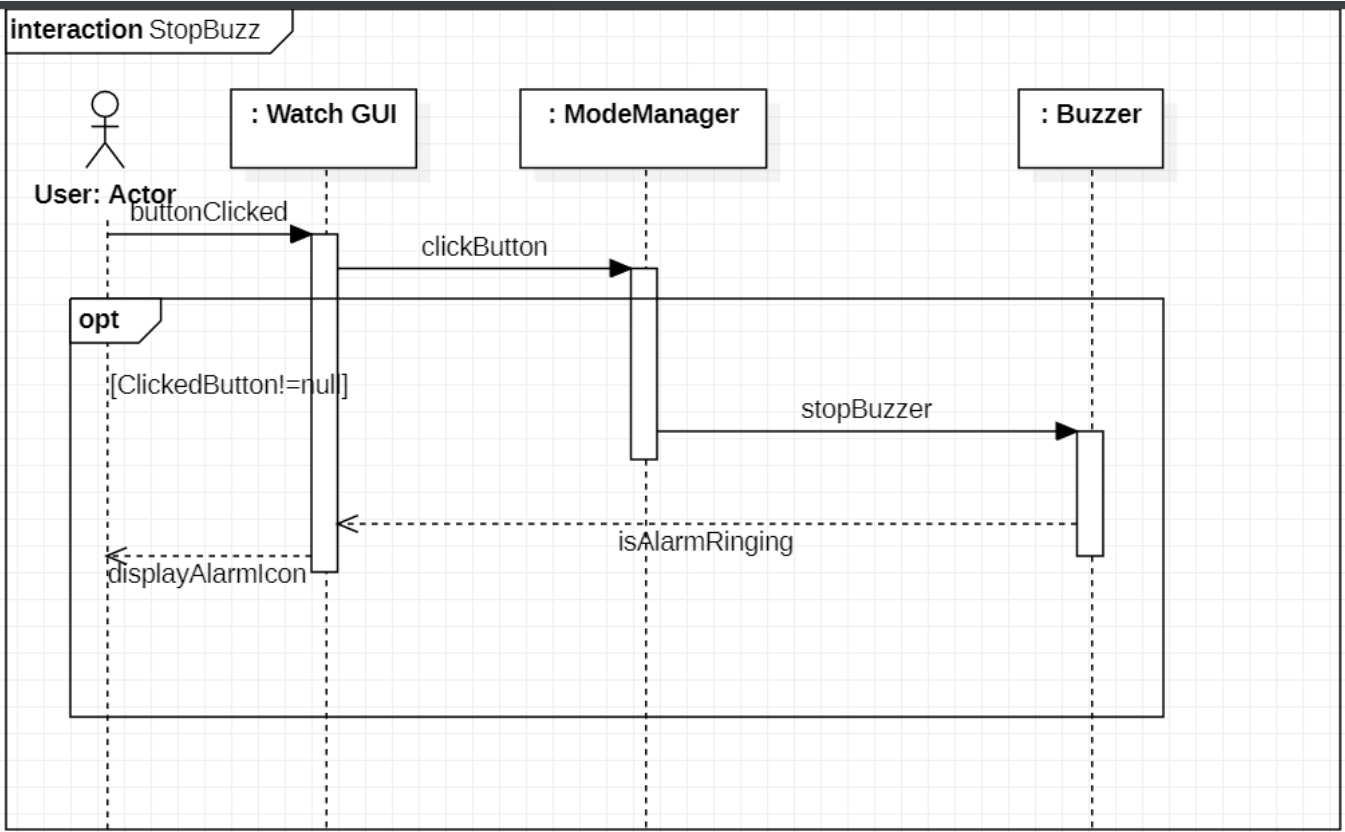


Reset Calorie Check

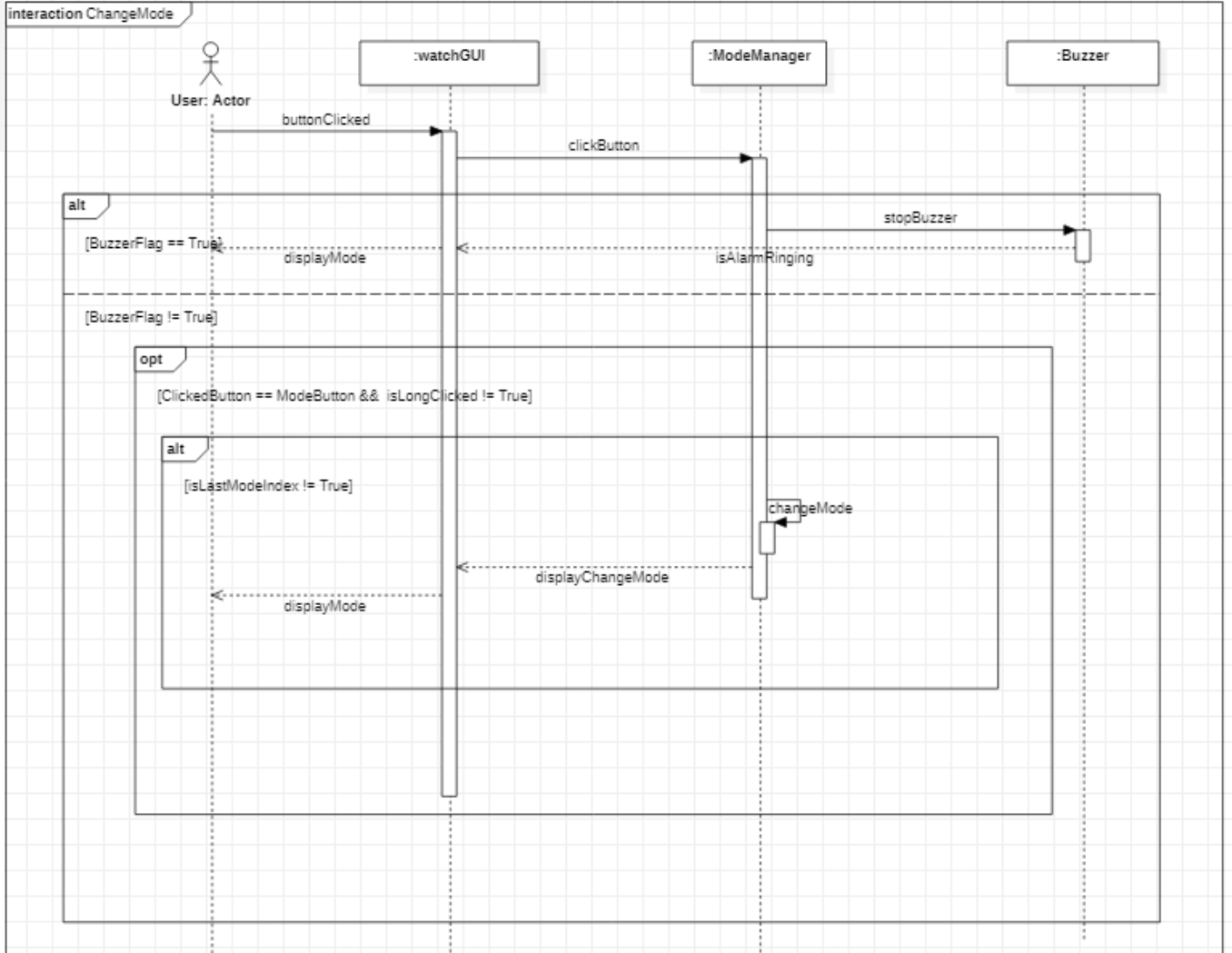
Reset Calorie Check



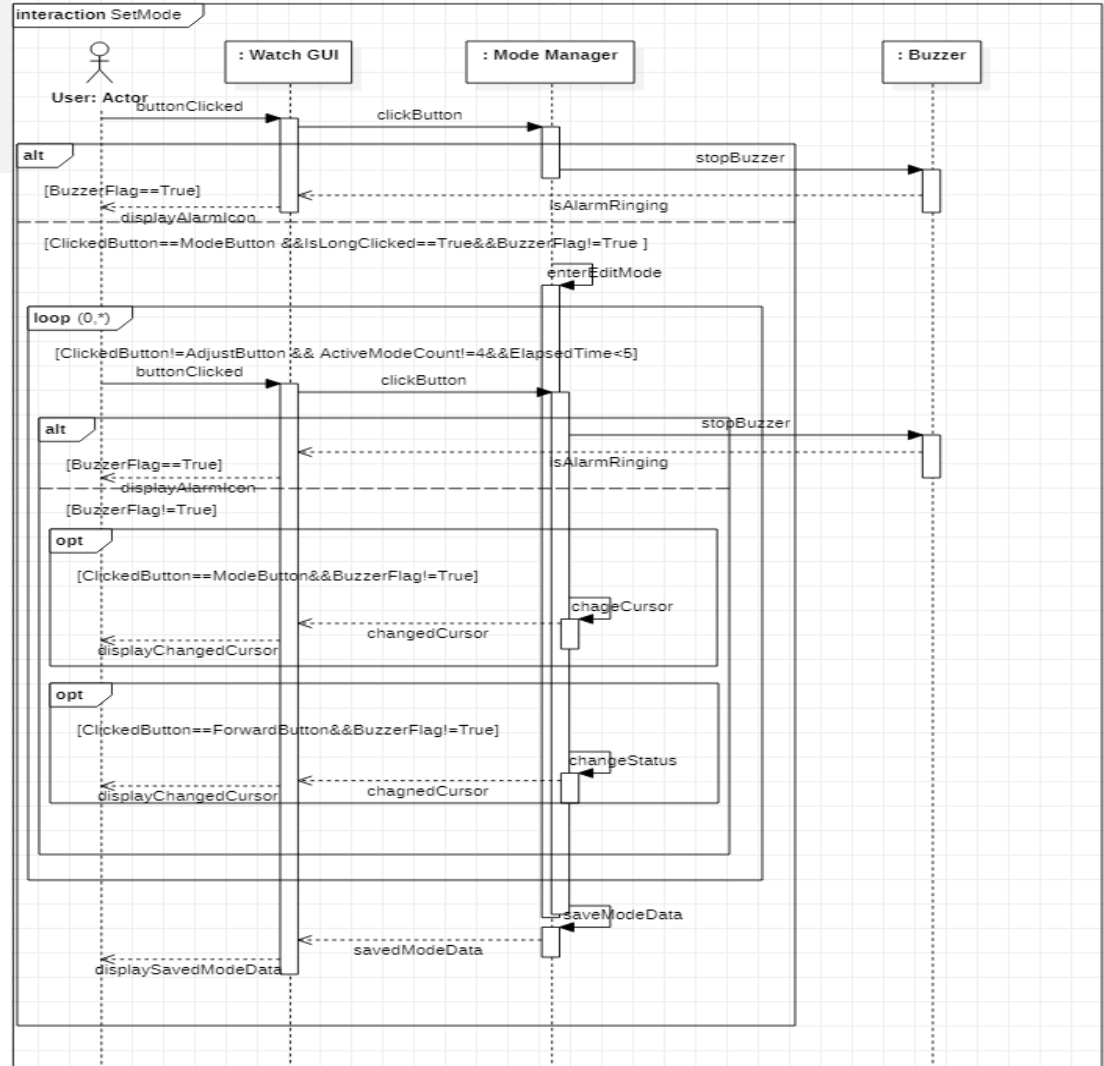
Stop Buzz



Change Mode



Set Mode



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 예의체크 0으로 돌아가는지
    result=t.getCurrentTime().getYear();
    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());
}
```

```
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();
    t.changeCursor();

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



Time Test

```
@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 10000 이상일 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);
}
```

```
@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 10000 이상일 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 활성화시킬
    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

```
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)));
}
```

```
@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(LocalDate.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



		✓ Tests passed: 8 of
▶	✓ AlarmTest (Sys)	173 ms
▶	✓ SaveAlarm	168 ms
▶	✓ IncreaseAlarmTime	0 ms
▶	✓ TurnOffAlarm	5 ms
▶	✓ ChangeCursor	0 ms
▶	✓ ChangeAlarm	0 ms
▶	✓ DecreaseAlarmTime	0 ms
▶	✓ IsAlarmTimeCheck	0 ms
▶	✓ TurnOnOffAlarm	0 ms

"C:\Program F
Process finis



Timer Test

```
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);  
    }  
}
```

```
@Test  
public void increaseData(){  
    Buzzer buzzer = new Buzzer();  
    Timer timer=new Timer(buzzer);  
  
    timer.enterEditTimer();  
    //시작test  
    //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.setTimeTime(LocalDateTime.of(year: 2000, month: 1, dayOfMonth: 5, hour: 3, minute: 0, second: 0));  
    timer.increaseData();  
    assertEquals(timer.getTimeTime().getDayOfMonth(), actual: 1);  
    assertEquals(timer.getTimeTime().getHour(), actual: 0);  
  
    //시간이 0일때 increasedata후 1단위되는지 확인  
    int temp=timer.getTimeTime().getHour()+1;  
    timer.increaseData();  
    assertEquals(timer.getTimeTime().getHour(), temp);  
  
    //본 테스트  
    timer.changeCursor();  
    //현재 커서가 1(분)인지 확인  
    assertEquals(timer.getCurrentCursor(), actual: 1);  
  
    //분이 59일때 increasedat 후 0이 되는 지 확인  
    timer.setTimeTime(LocalDateTime.of(year: 2000, month: 1, dayOfMonth: 5, hour: 3, minute: 59, second: 0));  
    timer.increaseData();  
    assertEquals(timer.getTimeTime().getHour(), actual: 3);  
    assertEquals(timer.getTimeTime().getMinute(), actual: 0);  
    assertEquals(timer.getTimeTime().getSecond(), actual: 0);  
  
    //분이 0일때 increasedata 후 1되는지 확인  
    temp=timer.getTimeTime().getMinute()+1;  
    timer.increaseData();  
    assertEquals(timer.getTimeTime().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.setTimeTime(LocalDateTime.of(year: 2000, month: 1, dayOfMonth: 5, hour: 3, minute: 0, second: 59));  
    timer.increaseData();  
    assertEquals(timer.getTimeTime().getMinute(), actual: 0);  
    assertEquals(timer.getTimeTime().getSecond(), actual: 0);  
  
    //초가 0일때 increasedata 후 1되는지 확인  
    temp=timer.getTimeTime().getSecond()+1;  
    timer.increaseData();  
    assertEquals(timer.getTimeTime().getSecond(), temp);  
}
```

```
@Test  
public void decreaseData(){  
    Buzzer buzzer = new Buzzer();  
    Timer timer=new Timer(buzzer);  
  
    timer.enterEditTimer();  
  
    //시작test  
    //현재 커서가 0(시간인지 확인)  
    assertEquals(timer.getCurrentCursor(), actual: 0);  
  
    //시간이 0시 일때 decreaseData 했을 때 99(day=5, hour=3) 되는지 확인  
    timer.setTimeTime(LocalDateTime.of(year: 2000, month: 1, dayOfMonth: 1, hour: 0, minute: 0, second: 0));  
    timer.decreaseData();  
    assertEquals(timer.getTimeTime().getDayOfMonth(), actual: 5);  
    assertEquals(timer.getTimeTime().getHour(), actual: 3);  
  
    //시간이 99일때 decreaseData 후 1 감소되는지 확인  
    int temp=timer.getTimeTime().getHour()-1;  
    timer.decreaseData();  
    assertEquals(timer.getTimeTime().getHour(), temp);  
  
    //본 테스트  
    timer.changeCursor();  
    //현재 커서가 1(분)인지 확인  
    assertEquals(timer.getCurrentCursor(), actual: 1);  
  
    //분이 0일때 decreaseData 후 59이 되는 지 확인  
    timer.setTimeTime(LocalDateTime.of(year: 2000, month: 1, dayOfMonth: 5, hour: 3, minute: 0, second: 0));  
    timer.decreaseData();  
    assertEquals(timer.getTimeTime().getHour(), actual: 3);  
    assertEquals(timer.getTimeTime().getMinute(), actual: 59);  
    assertEquals(timer.getTimeTime().getSecond(), actual: 0);  
  
    //분이 59일때 decreaseData 후 -1되는지 확인  
    temp=timer.getTimeTime().getMinute()-1;  
    timer.decreaseData();  
    assertEquals(timer.getTimeTime().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.setTimeTime(LocalDateTime.of(year: 2000, month: 1, dayOfMonth: 5, hour: 3, minute: 0, second: 0));  
    timer.decreaseData();  
    assertEquals(timer.getTimeTime().getMinute(), actual: 0);  
    assertEquals(timer.getTimeTime().getSecond(), actual: 59);  
  
    //초가 59일때 decreaseData 후 -1되는지 확인  
    temp=timer.getTimeTime().getSecond()-1;  
    timer.decreaseData();  
    assertEquals(timer.getTimeTime().getSecond(), temp);  
}
```

Timer Test



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

    timer.setTimerTime(LocalDate.of( year: 2000, month: 1, dayOfMonth: 1, hour: 1, minute: 1, second: 1));
    timer.setSettingTimer(LocalDate.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(LocalDate.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 더뎠르게 하고 buzzer가 on되었는지 확인
    for(int i=0; i<300; i++)
        timer.decreaseTimer();
    //buzzer on 확인
    assertEquals(buzzer.getBuzzerOn(), actual: true);
    //Timer일었는지 확인
    assertEquals(timer.getpauseTimerFlag(), actual: true);
    //Timer
    assertEquals(timer.getTimerTime(),LocalDate.of( year: 2000, month: 1, dayOfMonth: 1, hour: 0, minute: 0, second: 0));
}
```

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

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

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

    //리셋 상태가 아닐때 startTimer호출 시작이 되는 지 확인
    timer.setTimerTime(LocalDate.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 시전상태 확인
    timer.setTimerTime(LocalDate.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(LocalDate.of( year: 2000, month: 1, dayOfMonth: 1, hour: 1, minute: 1, second: 1));
    timer.setSettingTimer(LocalDate.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



The screenshot shows the VS Code test runner interface. At the top, it displays the file path `.gitignore` and the number of tests `43`. Below this, the `Run` tab is active, showing two test runs: `Main_test` and `TimerTest`. The `TimerTest` run is selected, and the results are displayed in a list:

Test Name	Duration
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

The status bar at the bottom indicates the current state: `9: Git`, `6: TODO`, `4: Run`, `0: Messages`, and `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: 2000000)));

    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: 99000000)));

    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

```
@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



The screenshot shows the Visual Studio Test Explorer interface. At the top, there are two tabs: 'Main_test' and 'StopWatchTest'. The 'StopWatchTest' tab is active, showing a toolbar with various icons and a status bar indicating 'Tests passed: 6 of 6'. Below the toolbar, a tree view shows the test results for 'StopWatchTest (Sys)'. The test suite is expanded, revealing six individual test methods, each with a green checkmark and a duration. To the right of the test results, the output window shows the path 'C:\Program Fil' and the text 'Process finishe'. At the bottom of the test results, a green box displays 'Tests passed: 6'.

Test Method	Duration
✓ pauseStopwatch	168 ms
✓ increaseCurrentTime	497 ms
✓ startStopwatch	440 ms
✓ lapStopwatch	2 ms
✓ resumeStopwatch	0 ms
✓ resetStopwatch	8 ms

Output: "C:\Program Fil
Process finishe

Tests passed: 6



Calorie Check Test

```
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 바뀌어야 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 바뀌어야 한다.
    tempSpeed = 99;
    calorieCheck.setTempSpeed(tempSpeed);
    calorieCheck.increaseData();
    assertEquals( expected: 0, calorieCheck.getTempSpeed());
}
```

```
@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 바뀌어야 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 바뀌어야 한다.
    tempSpeed = 0;
    calorieCheck.setTempSpeed(tempSpeed);
    calorieCheck.decreaseData();
    assertEquals( expected: 99, calorieCheck.getTempSpeed());
}
```



Calorie Check Test

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

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

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

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

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

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

Calorie Check Test

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

```
@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(100000000);
    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



```
@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

Run: Main_test x CalorieCheckTest x

Tests passed: 11 of 11

Test Name	Duration
CalorieCheckTest (Sys)	299 ms
startCalorieCheck	8 ms
increaseCalorieCheckTimer	291 ms
pauseCalorieCheck	0 ms
enterSetSpeedandWeight	0 ms
changeCursor	0 ms
resetCalorieCheck	0 ms
saveCalorieSetting	0 ms
decreaseData	0 ms
endCalorieCheck	0 ms
increaseData	0 ms
resumeCaloreCheck	0 ms

"C:\Program Files:
Process finished



World Time Test

```
@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);
}
```

```
@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



▼ ✓ Test Results	16 ms
▼ ✓ WorldTimeTest	16 ms
✓ getWorldTime()	15 ms
✓ changeTimezone()	1 ms



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



Run: Main_test x BuzzerTest x

Tests passed: 2 of 2

Test Name	Duration	Output
BuzzerTest (Sys)	181 ms	"C:\Program Fil
stopBuzzer	181 ms	
beepBuzzer	0 ms	Process finishe

Testing Traceability Analysis

4





Use Case	Match	index	Operation in Sequence Diagram	Match	index	Operation in Interaction Diagram	Match	Method	Class	System Test	Match	index	Unit Test
Set Time	S16,S17,S18,S19,S20	S1	startCalorieCheck()	I1	I1	startCalorieCheck	M7	M1	changeMode	Show Time Test	N/A	U1	clickButton()
Set Timer	S34,S39,S40,S41	S2	pauseCalorieCheck()	I3	I2	increaseCalorieCheckTimer	M8	M2	clickButton	Set Time Test	U46,U47,U48,U49,U50	U2	enterEditMode()
Start Timer	S35	S3	resumeCalorieCheck()	I4	I3	pauseCalorieCheck	M9	M3	enterEditMode	Show Timer Test	N/A	U3	changeCursor()
Pause Timer	S37	S4	enterSetSpeedandWeight()	I5,I9,I10,I11,I7	I4	resumeCalorieCheck	M16	M4	changeMode	Set Timer Test	U34,U35,U36,U37,U38,U3	U4	changeStatus()
Resume Timer	S36	S5	resetCalorieCheck()	I8	I5	enterSetSpeedandWeight	M10	M5	changeStatus	Start Timer Test	U40	U5	saveModeData()
Cancel Timer	S38	S6	CalorieCheck.increaseData()	I9	I6	changeCursor	M13	M6	saveModeData	Pause Timer Test	U41	U6	changeMode()
Change Alarm	S11	S7	CalorieCheck.decreaseData()	I10	I7	saveCalorieSetting	M14	M7	startCalorieCheck	Resume Timer Test	U40	U7	changeCursor()
Turn On/Off Alarm	S12	S8	CalorieCheck.changeCursor()	I11	I8	resetCalorieCheck	M15	M8	increaseCalorieCheckTime	Cancel Timer Test	U42	U8	increaseData()
Set Alarm	S9,S10,S13,S14,S15	S9	enterEditAlarm()	I28	I9	CalorieCheck.increaseData()	M11	M9	pauseCalorieCheck	Show Alarm Test	N/A	U9	decreaseData()
Start Stopwatch	S23	S10	saveCurrentAlarm()	I32	I10	CalorieCheck.decreaseData()	M12	M10	enterSetSpeedandWeight	Change Alarm Test	U27	U10	saveCalorieSetting()
Pause Stopwatch	S24	S11	changeAlarm()	I33	I11	CalorieCheck.changeCursor()	M13	M11	increaseData	Turn On/Off Alarm Test	U28	U11	enterSetSpeedandWeight()
Resume Stopwatch	S25	S12	turnOnOffAlarm()	I34	I12	enterEditTimer	M38,M39,M40,M41,M42	M12	decreaseData	Set Alarm Test	U29,U30,U31,U32	U12	startCalorieCheck()
Lap Stopwatch	S27	S13	increaseAlarmTime()	I30	I13	saveTimer	M42	M13	changeCursor	Return to Default Screen Test	U5	U13	resumeCalorieCheck()
Reset Stopwatch	S26	S14	decreaseAlarmTime()	I31	I14	startTimer	M43	M14	saveCalorieSetting	Show Stopwatch Test	N/A	U14	pauseCalorieCheck()
Change Timezone	S28	S15	Alarm.changeCursor()	I29	I15	pauseTimer	M45	M15	resetCalorieCheck	Start Stopwatch Test	U18	U15	endCalorieCheck()
Start Calorie Check	S1	S16	saveData()	I22	I16	resumeTimer	M43	M16	resumeCalorieCheck	Pause Stopwatch Test	U20	U16	increaseCalorieCheckTimer()
Pause Calorie Check	S2	S17	enterEditData()	I23	I17	decreaseTimer	M40	M17	endCalorieCheck	Resume Stopwatch Test	U19	U17	resetCalorieCheck()
Resume Calorie Check	S3	S18	Time.increaseData()	I24	I18	cancelTimer	M48	M18	calculateCalorie	Reset Stopwatch Test	U21	U18	startStopwatch()
Set Speed and Weight	S4,S6,S7,S8	S19	Time.decreaseData()	I25	I19	Timer.increaseData()	M39	M19	startStopwatch	Lap Stopwatch Test	U22	U19	resumeStopwatch()
Reset Calorie Check	S5	S20	Time.changeCursor()	I26	I20	Timer.decreaseData()	M40	M20	increaseCurrentTime	Show World Time Test	N/A	U20	pauseStopwatch()
Stop Buzz	S22	S21	clickButton()	I27	I21	Timer.changeCursor()	M41	M21	pauseStopwatch	Change Timezone Test	U25	U21	resetStopwatch()
Change Mode	S33	S22	stopBuzzer()	I50	I22	saveData()	M52	M22	resetStopwatch	Show Calorie Check Test	N/A	U22	lapStopwatch()
SetMode	S29,S31,S30,S32	S23	startStopwatch()	I38	I23	enterEditData()	M49	M23	lapStopwatch	Set Speed and Weight Test	U11,U7,U8,U9,U10	U23	increaseCurrentTime()
		S24	pauseStopwatch()	I40	I24	Time.increaseData()	M50	M24	resumeStopwatch	Start Calorie Check test	U12	U24	getWorldTime
		S25	resumeStopwatch()	I41	I25	Time.decreaseData()	M51	M25	changeTimezone	Pause Calorie Check test	U14	U25	changeTimeZone
		S26	resetStopwatch()	I42	I26	Time.changeCursor()	M53	M26	getWorldTime	Resume Calorie Check test	U13	U26	isAlarmTimeCheck()
		S27	lapStopwatch()	I43	I27	clickButton	M2	M27	enterEditAlarm	Reset Calorie Check test	U17	U27	ChangeAlarm()
		S28	changeTimezone()	I44	I28	enterEditAlarm	M27,M29,M30,M31,M32	M28	isAlarmTimeCheck	Beep Buzz test	U43	U28	TurnOnOffAlarm()
		S29	enterEditMode()	I46	I29	Alarm.changeCursor	M29	M29	changeCursor	Stop Buzz test	U44	U29	IncreaseAlarmTime()
		S30	saveModeData()	I49	I30	increaseAlarmTime	M30	M30	increaseAlarmTime	Change Mode test	U6	U30	DecreaseAlarmTime()
		S31	changeCursor()	I47	I31	decreaseAlarmTime	M31	M31	decreaseAlarmTime			U31	ChangeCursor()
		S32	changeStatus()	I48	I32	saveAlarm	M32	M32	saveAlarm			U32	SaveAlarm()
		S33	changeMode()	I45	I33	changeAlarm	M33	M33	changeAlarm			U33	TurnOffAlarm()
		S34	enterEditMode()	I12,I19,I20,I21,I13	I34	turnOnOffAlarm	M34	M34	turnOnOffAlarm			U34	enterEditTimer()
		S35	startTimer	I14	I35	requestExpirationTime	M35	M35	toggleAlarmTimer			U35	changeCursor()
		S36	resumeTimer	I16	I36	toggleAlarmTimer	M35	M36	saveAlarmTime			U36	increaseData()
		S37	pauseTimer	I15	I37	saveAlarmTime	M36	M37	requestExpirationTime			U37	decreaseData()
		S38	cancelTimer	I18	I38	startStopwatch	M19	M38	enterEditTimer			U38	saveTimer()
		S39	Timer.increaseData()	I19	I39	increaseCurrentTime	M20	M39	increaseData			U39	decreaseTimer()
		S40	Timer.decreaseData()	I20	I40	pauseStopwatch	M21	M40	decreaseData			U40	startTimer()
		S41	Timer.changeCursor()	I21	I41	resumeStopwatch	M24	M41	changeCursor			U41	pauseTimer()
					I42	resetStopwatch	M22	M42	saveTimer			U42	cancelTimer()
					I43	lapStopwatch	M23	M43	startTimer			U43	beepBuzzer
					I44	changeTimezone	M25	M44	decreaseTimer			U44	stopBuzzer()
					I45	changeMode	M1	M45	pauseTimer			U46	changeCursor
					I46	enterEditMode()	M3	M46	cancelTimer			U47	enterEditData
					I47	changeCursor()	M5	M47	stopBuzzer			U48	increaseData
					I48	changeStatus()	M5	M48	changeCursor			U49	decreaseData
					I49	saveModeData()	M6	M49	enterEditData			U50	saveData
					I50	stopBuzzer()	M47	M50	increaseData				
								M51	decreaseData				
								M52	saveData				
								M53	changeCursor				

System Test	Match
Show Time Test	N/A
Set Time Test	U46,U47,U48,U49,U50
Show Timer Test	N/A
Set Timer Test	U34,U35,U36,U37,U38,U3
Start Timer Test	U40
Pause Timer Test	U41
Resume Timer Test	U40
Cancel Timer Test	U42
Show Alarm Test	N/A
Change Alarm Test	U27
Turn On/Off Alarm Test	U28
Set Alarm Test	U29,U30,U31,U32
Return to Default Screen Test	U5
Show Stopwatch Test	N/A
Start Stopwatch Test	U18
Pause Stopwatch Test	U20
Resume Stopwatch Test	U19
Reset Stopwatch Test	U21
Lap Stopwatch Test	U22
Show World Time Test	N/A
Change Timezone Test	U25
Show Calorie check Test	N/A
Set Speed and Weight test	U11,U7,U8,U9,U10
Start Calorie Check test	U12
Pause Calorie Check test	U14
Resume Calorie Check test	U13
Reset Calorie Check test	U17
Beep Buzz test	U43
Stop Buzz test	U44
Change Mode test	U6
Set Mode test	U1,U2,U3,U4,U5

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 pause 상태에서 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 Timer 상태에서 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 켜는 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	설정된 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