Software Modeling & Analysis

낚시하시계

(Fishing Digital Watch)

OOPT Stage 2050 & 2060

Team 8

201714170 조해성
201714168 유호원
201310507 정희찬
201613575 루카이
201712845 류한길

Index

Activity 2051. Implement Class & Methods Definitions

1) Class Definition

1.1 SystemUI

Type	Class
Name	SystemUI
Purpose	User가 Digitalwatch를 사용할 수 있게 해주는 클래스
Overview	N/A
Cross Reference	Funtions : R0, R1, R2, R3, R4, R5, R6 Use Cases : “showTime”, “adjustTime”, “showTimer”, “setTimer”, “startTimer”, “pauseTimer”, “resetTimer”, “buzzTimer”, “showAlarm”, “nextAlarm”, “addAlarm”, “deleteAlarm”, “buzzAlarm”, “stopAlarm”, “showStopwatch”, “startStopwatch”, “recordStopwatch”, “pauseStopwatch”, “resetStopwatch”, “showTide”, “nextTide”, “showMoonphase”, “modeSelect”, “nextMode”
Exceptional Course of Events	N/A

1.2 CreateMode

Type	Class
Name	CreateMode
Purpose	System이 모드를 생성하는 클래스
Overview	N/A
Cross Reference	Funtions :R6 Use Cases : “modeSelect”
Exceptional Course of Events	N/A

1.3 DeleteMode

Type	Class

Name	DeleteMode
Purpose	System이 모드를 삭제하는 클래스
Overview	N/A
Cross Reference	Funtions :R6 Use Cases : “modeSelect”
Exceptional Course of Events	N/A

1.4 ModeSelector

Type	Class
Name	ModeSelector
Purpose	User가 모드를 선택하도록 해주는 클래스
Overview	N/A
Cross Reference	Funtions :R6 Use Cases : “modeSelect”
Exceptional Course of Events	N/A

1.5 TimeKeeping

Type	Class
Name	TimeKeeping
Purpose	System이 현재시간을 보여주게 하는 모드
Overview	N/A
Cross Reference	Funtions :R0 Use Cases : “showTime”, “adjustTime”
Exceptional Course of Events	N/A

1.6 TimeDB

Type	Class
Name	TimeDB

Purpose	System이 현재시간을 저장하게 만드는 모드
Overview	N/A
Cross Reference	Funtions :R0 Use Cases : “showTime”, “adjustTime”
Exceptional Course of Events	N/A

1.7 Alarm

Type	Class
Name	Alarm
Purpose	User가 Alarm을 사용할수 있게 해주는 클래스
Overview	N/A
Cross Reference	Funtions : R2 Use Cases : “showAlarm”, “nextAlarm”, “addAlarm”, “deleteAlarm”, “buzzAlarm”, “stopAlarm”
Exceptional Course of Events	N/A

1.8 Timer

Type	Class
Name	Timer
Purpose	User가 Timer를 사용할 수있게 해주는 클래스
Overview	N/A
Cross Reference	Functions : R1 Use Cases : “showTimer”, “setTimer”, “startTimer”, “pauseTimer”, “resetTimer”, “buzzTimer”
Exceptional Course of Events	N/A

1.9 Stopwatch

Type	Class
Name	Stopwatch

Purpose	User가 스탑워치를 사용할수 있게 해주는 클래스
Overview	N/A
Cross Reference	Functions :R3 Use Cases : “showStopwatch”, “startStopwatch”, “recordStopwatch”, “pauseStopwatch”, “resetStopwatch”
Exceptional Course of Events	N/A

1.10 Tide

Type	Class
Name	Tide
Purpose	User가 Tide를 확인할수 있게 해주는 클래스
Overview	N/A
Cross Reference	Functions : R4 Use Cases : “showTide”, “nextTide”, “calculateTide”
Exceptional Course of Events	N/A

1.11 Moonphase

Type	Class
Name	Moonphase
Purpose	User가 Moonphase를 확인할수있게 해주는 클래스
Overview	N/A
Cross Reference	Functions : R5 Use Cases : “showMoonphase”, “calculateMoonphase”
Exceptional Course of Events	N/A

2) Method Definition

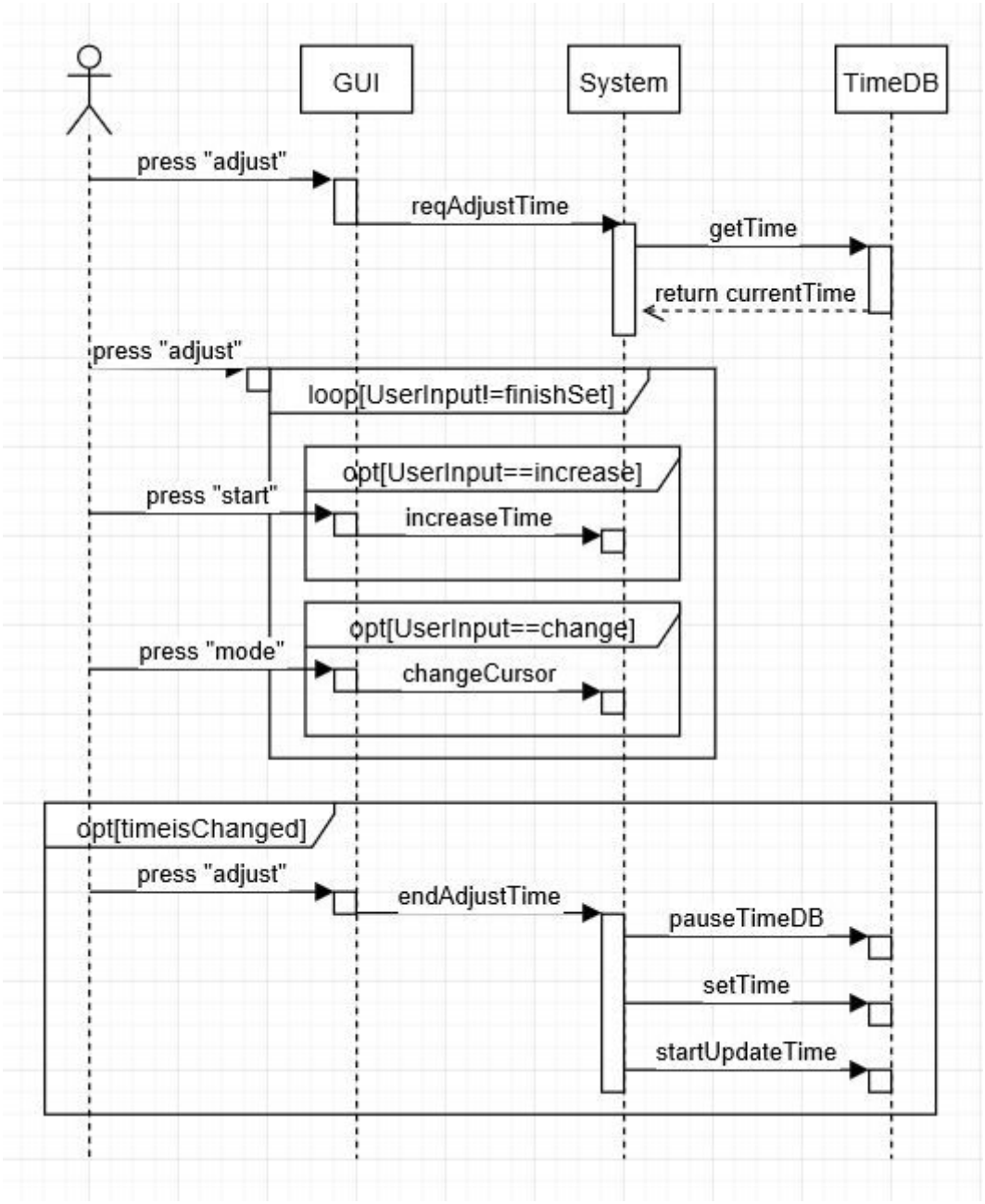
1.1.1

Type	Method

Name	calculateTide
Purpose	현재시간을 음력으로 바꿔 밀물썰물의 차이를 계산한다
Cross Reference	Functions:R6
Input(Method)	string : currtime
Output(Method)	string : tidegraphic
Abstract Operation(Method)	N/A
Exceptional Course of Events	N/A

Type	Method
Name	calaculateMoonphase
Purpose	현재시간에 맞는 달모양을 계산한다
Cross Reference	Functions:R5
Input(Method)	string : currtime
Output(Method)	string : moongraphic
Abstract Operation(Method)	N/A
Exceptional Course of Events	N/A

Activity 2052. Implement Windows



adjust time

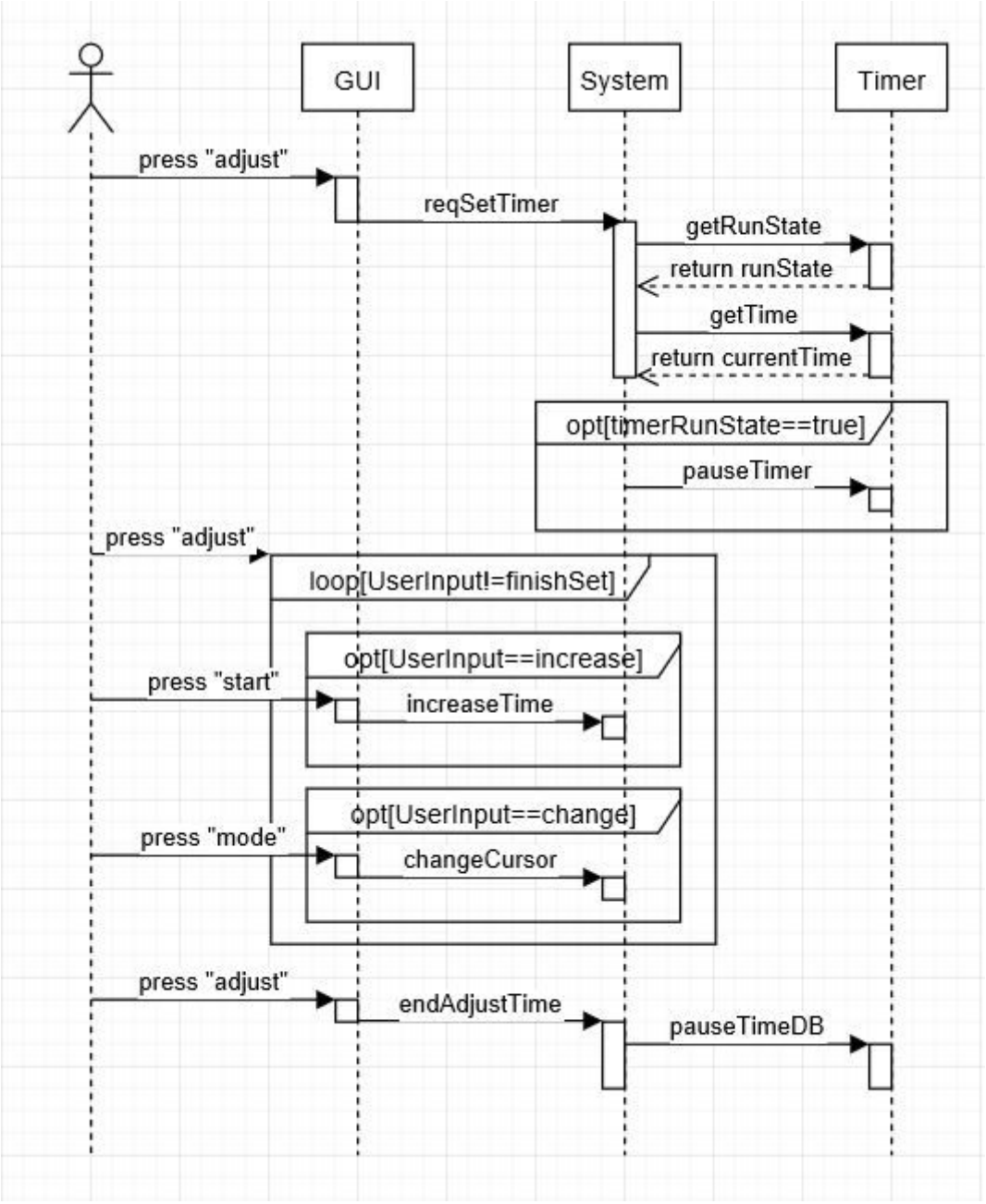
Name	adjustTime
Responsibilities	TimeKeeping모드의 “adjust”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.0.1
Notes	adjustTime으로 진입한다
Pre-conditions	TimeKeeping모드여야 한다

Post-Conditions	N/A

Name	increaseTime
Responsibilities	adjustTime에서 “start”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.0.1
Notes	사용자가 원하는 만큼 시간을 증가시킨다
Pre-conditions	adjustTime을 진입해야 한다
Post-Conditions	ChangeCurser로 진행할수 있다

Name	ChangeCurser
Responsibilities	adjustTime에서 “mode”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.0.1
Notes	다음 커서로 바꾼다
Pre-conditions	adjustTime을 진입해야 한다
Post-Conditions	커서가 가리키는 시간을 증가시킬수 있다

Name	endAdjustTime
Responsibilities	adjustTime에서 “Adjust”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.0.1
Notes	설정한 시간을 적용시킨다
Pre-conditions	adjustTime을 진입해야 한다
Post-Conditions	N/A

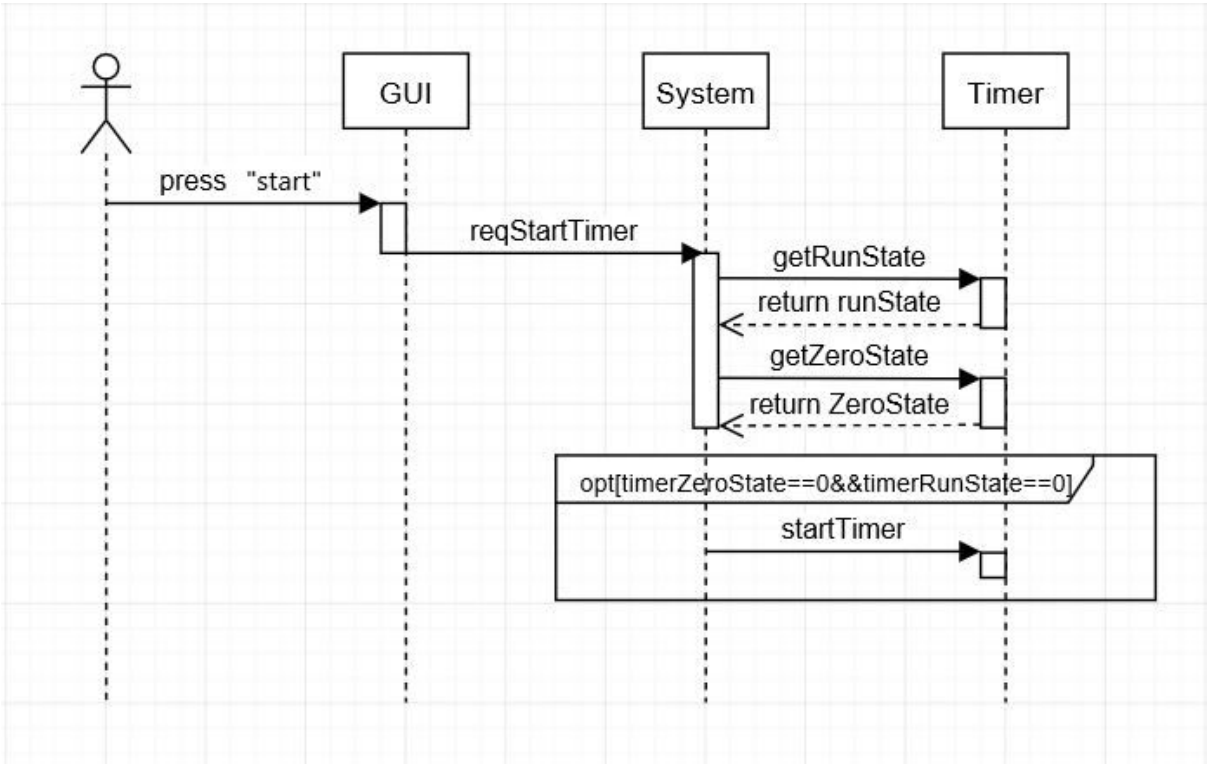


Name	reqsetTimer
Responsibilities	Timer모드의 “adjust”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.1.1
Notes	adjustTimer로 진입한다
Pre-conditions	TimerState가 0이어야 한다
Post-Conditions	N/A

Name	increaseTimerTime
Responsibilities	adjustTimer에서 “start”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.0.1
Notes	사용자가 원하는 만큼 시간을 증가시킨다
Pre-conditions	adjustTimer을 진입해야 한다
Post-Conditions	ChangeCurser로 진행할수 있다

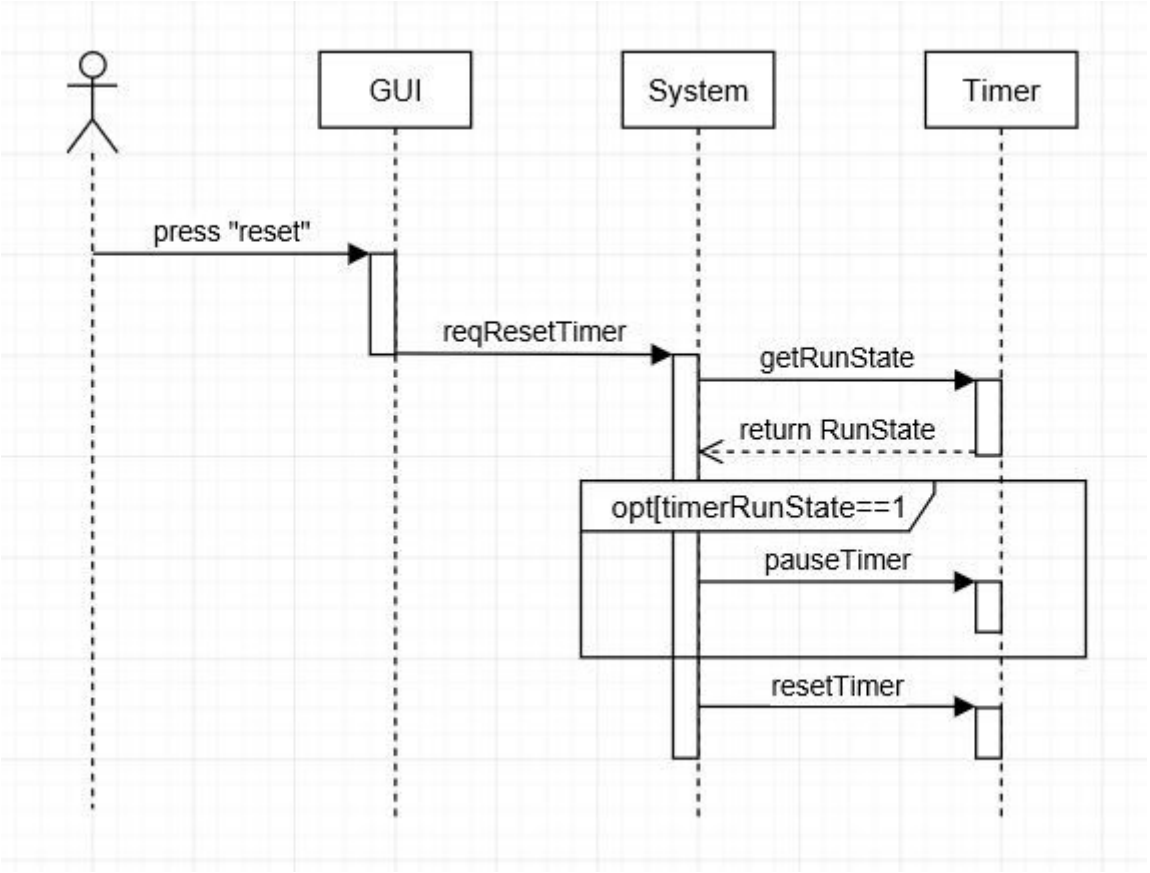
Name	ChangeCurser
Responsibilities	adjustTimer에서 “mode”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.1.1
Notes	다음 커서로 바꾼다
Pre-conditions	adjustTime을 진입해야 한다
Post-Conditions	현재 시간을 증가시킬수 있다

Name	endsetTimer
Responsibilities	Timer모드의 “adjust”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.1.1
Notes	adjustTimer에서 Timer모드로 돌아간다
Pre-conditions	Timer모드여야 한다 TimerState가 0이어야 한다
Post-Conditions	타이머를 설정한 경우 타이머가 저장된다



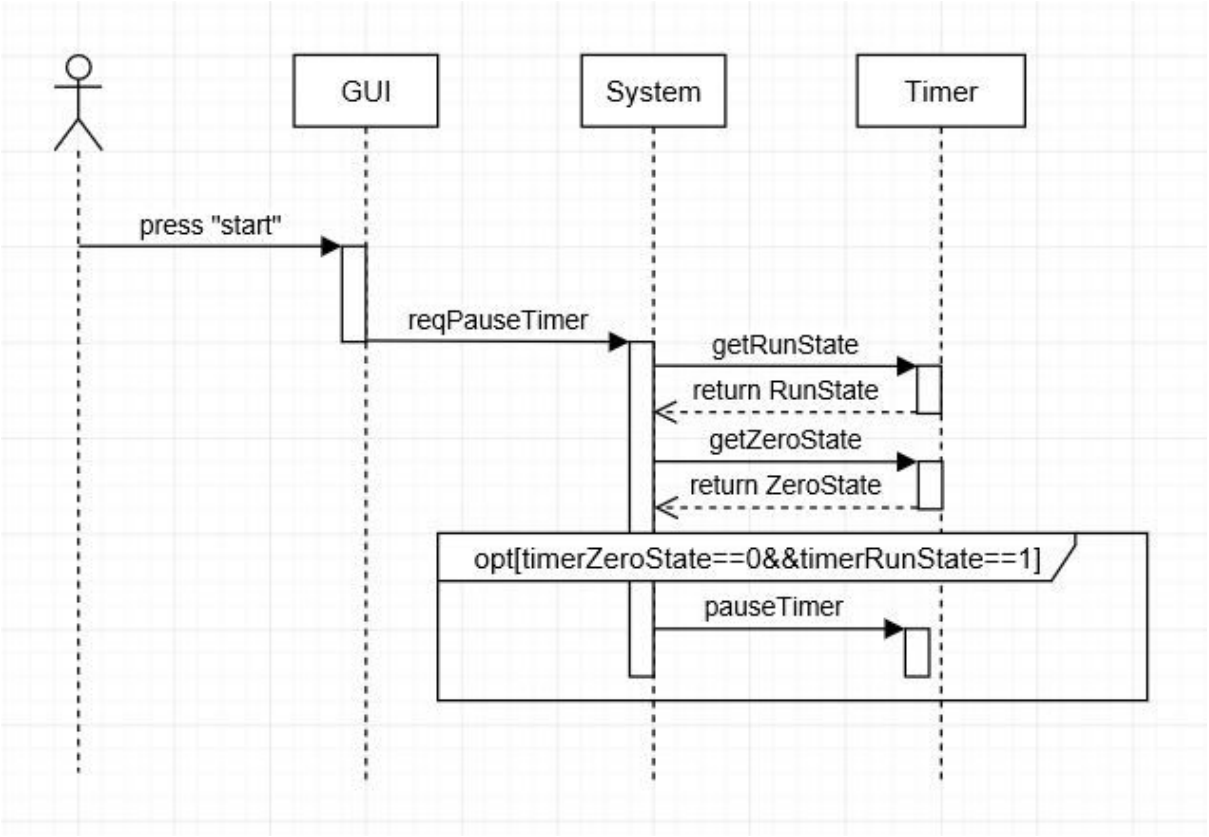
startTimer

Name	reqstartTimer
Responsibilities	Timer를 설정하고 “start”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.1.2
Notes	타이머를 시작한다
Pre-conditions	TimerState가 0이어야 하고 타이머가 설정되어야 한다
Post-Conditions	타이머를 멈출수 있다



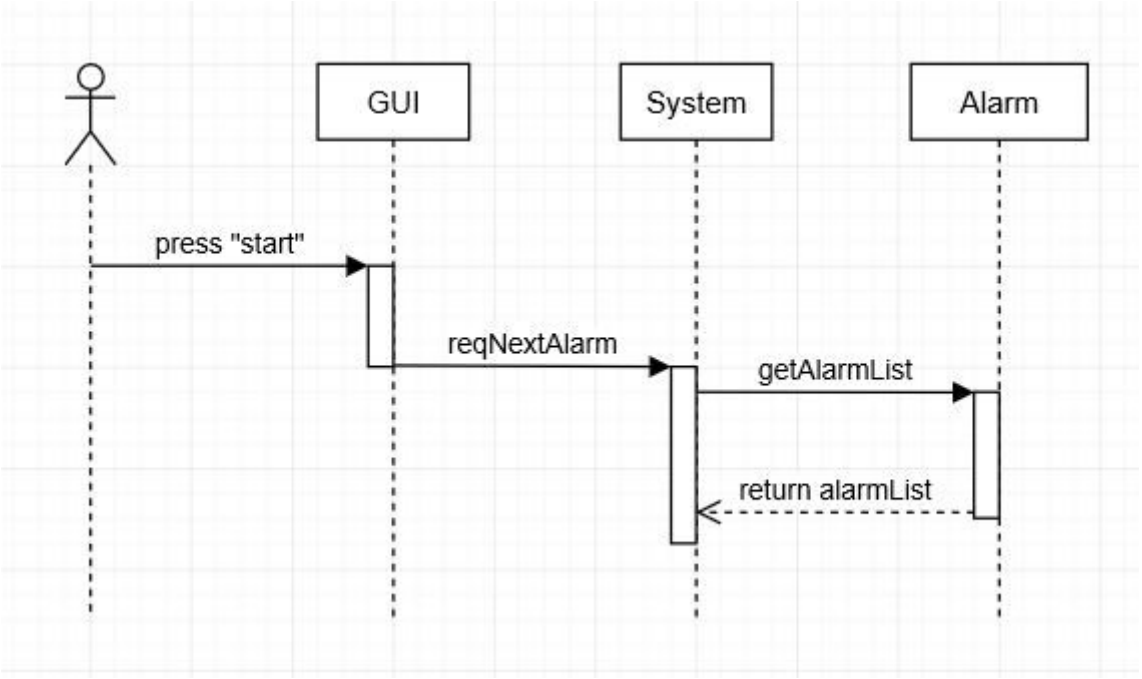
resetTimer

Name	reqResetTimer
Responsibilities	Timer모드의 “reset”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.0.1
Notes	타이머가 0초로 리셋된다
Pre-conditions	TimerState가 1이어야 한다
Post-Conditions	N/A



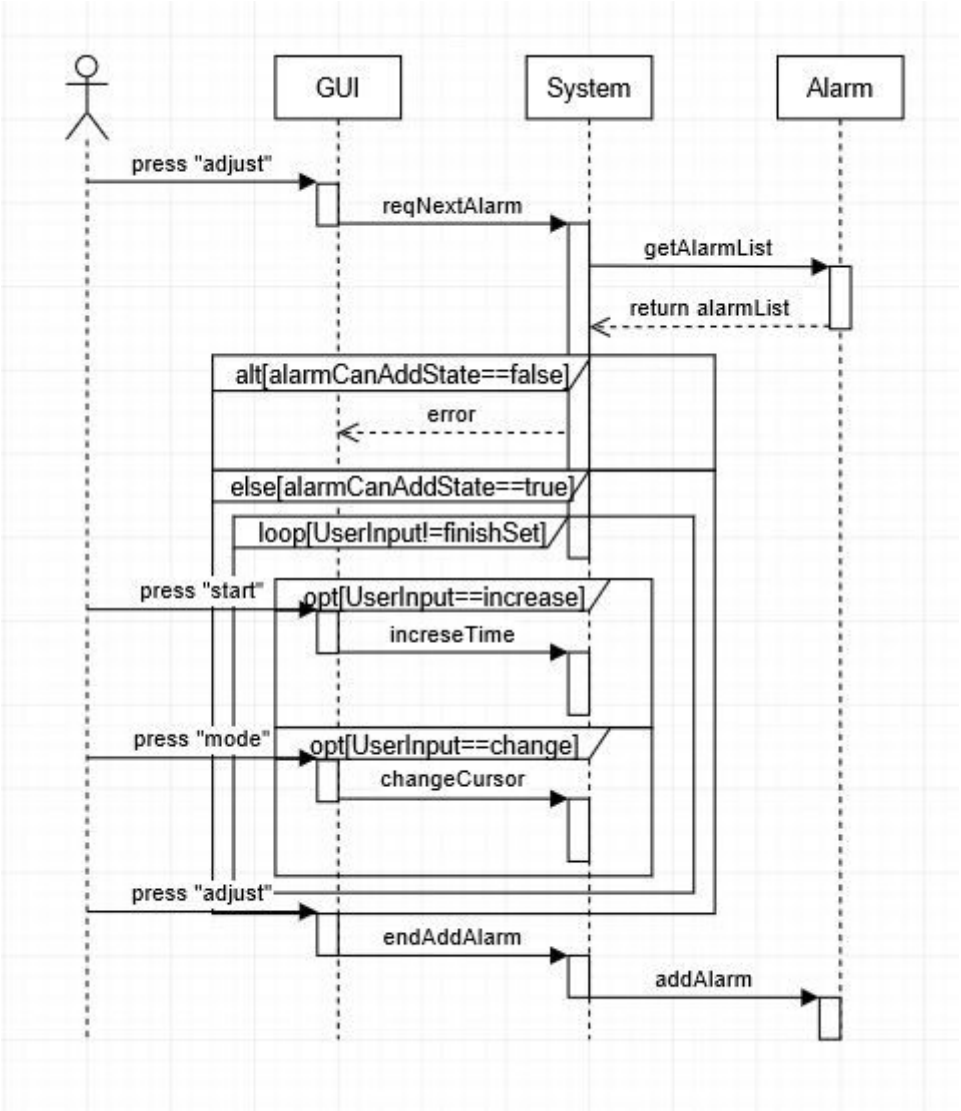
pauseTimer

Name	reqPauseTimer
Responsibilities	Timer모드의 “start”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.1.3
Notes	타이머가 흐르다가 일시정지된다
Pre-conditions	runState가 1이어야 한다
Post-Conditions	N/A



next alarm

Name	reqNextAlarm
Responsibilities	Alarm모드의 “start”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.2.1
Notes	alarmlist의 다음 index값으로 설정해서다음 알람이 표시가 된다
Pre-conditions	N/A
Post-Conditions	N/A



add alarm

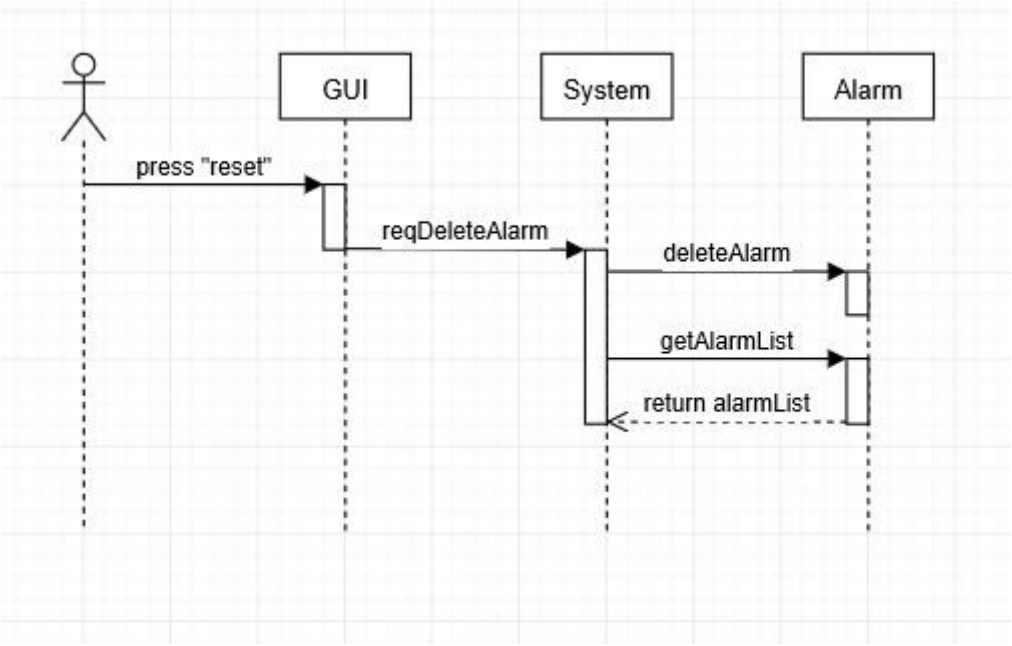
Name	reqAddAlarm
Responsibilities	Alarm모드의 “adjust”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.2.2
Notes	addAlarm에 진입할수 있다
Pre-conditions	이전에 설정된 알람이 4개 미만이어야 한다
Post-Conditions	알람 시간 설정을 할수 있다

Name	increaseTime

Responsibilities	addAlarm에서 “start”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.2.2
Notes	사용자가 원하는 만큼 시간을 증가시킨다
Pre-conditions	addAlarm으로 진입한다
Post-Conditions	ChangeCurser로 진행할수 있다

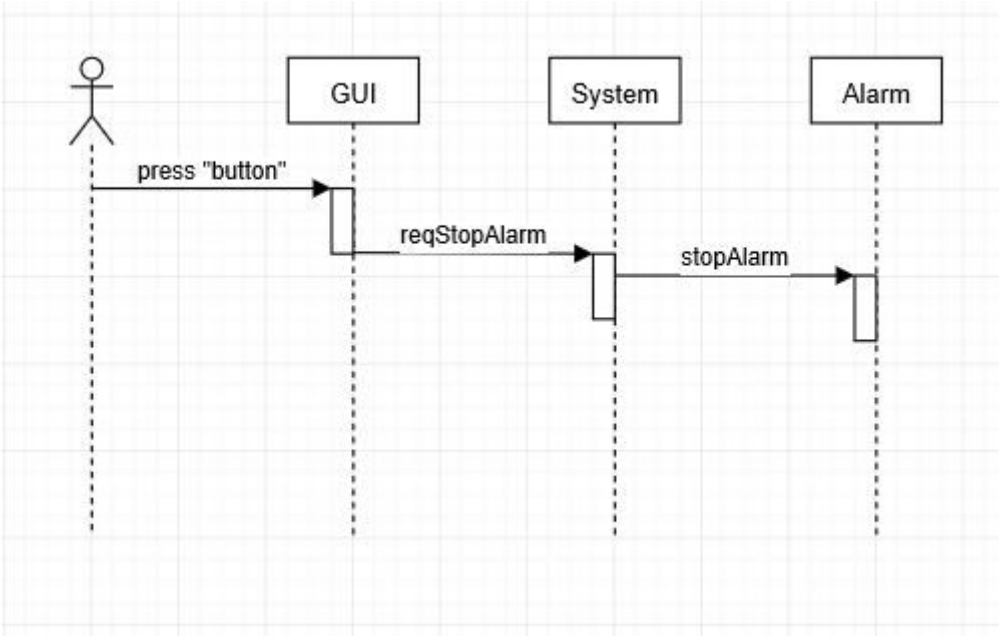
Name	ChangeCurser
Responsibilities	adjustTimer에서 “mode”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.2.2
Notes	다음 커서로 바꾼다
Pre-conditions	addAlarm을 진입해야 한다
Post-Conditions	현재 시간을 증가시킬수 있다

Name	endAddAlarm
Responsibilities	addAlarm모드의 “adjust”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.2.2
Notes	설정한 알람을 적용시킨다
Pre-conditions	N/A
Post-Conditions	알람모드로 돌아간다



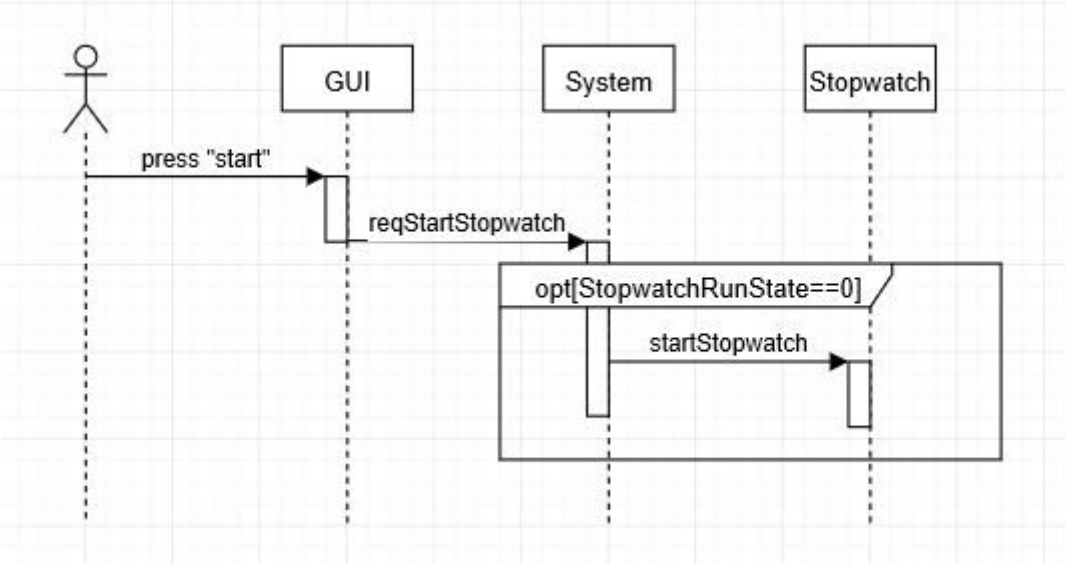
delete alarm

Name	reqDeleteAlarm
Responsibilities	Alarm모드의 “reset”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.2.3
Notes	현재알람을 지운다
Pre-conditions	알람이 설정되어 있어야 한다
Post-Conditions	N/A



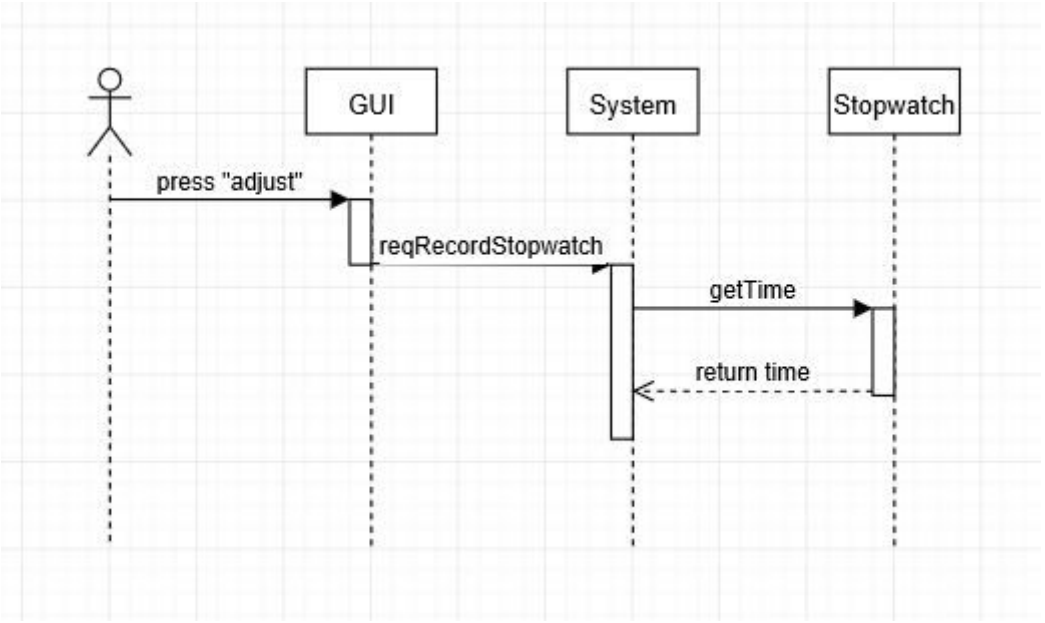
stop alarm

Name	reqStopAlarm
Responsibilities	어떤 모드이든 아무 버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.2.5
Notes	알람이 울리는 경우 알람 울리는걸 종료한다
Pre-conditions	alarmState가 1이어야 한다
Post-Conditions	N/A



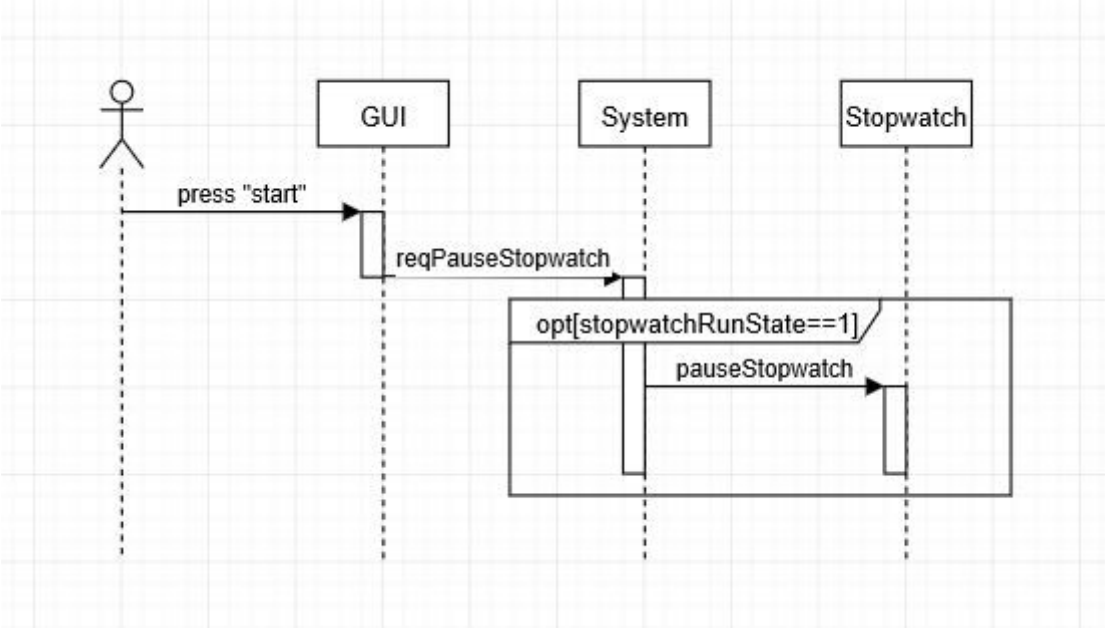
start stopwatch

Name	reqStartStopwatch
Responsibilities	Stopwatch모드의 “start”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.3.1
Notes	스탑워치가 시작된다
Pre-conditions	runState가 0이어야 한다
Post-Conditions	N/A



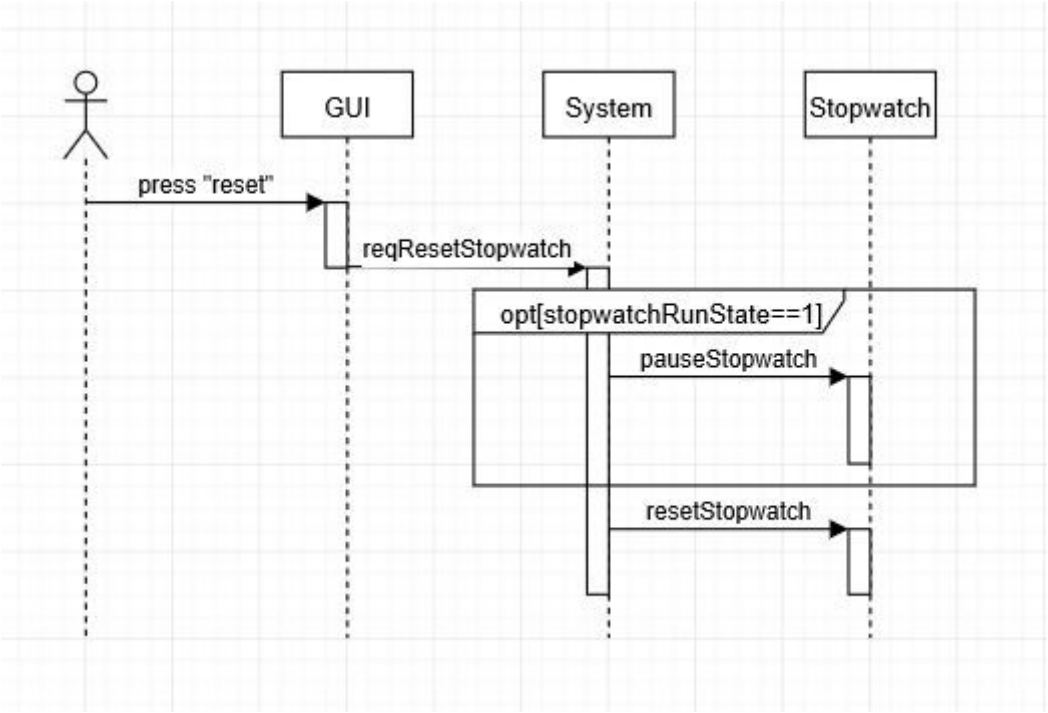
record stopwatch

Name	reqRecordStopwatch
Responsibilities	Stopwatch모드의 “adjust”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.3.1
Notes	현재 스탑워치 시간을 저장한다
Pre-conditions	runState가 1이어야 한다
Post-Conditions	N/A



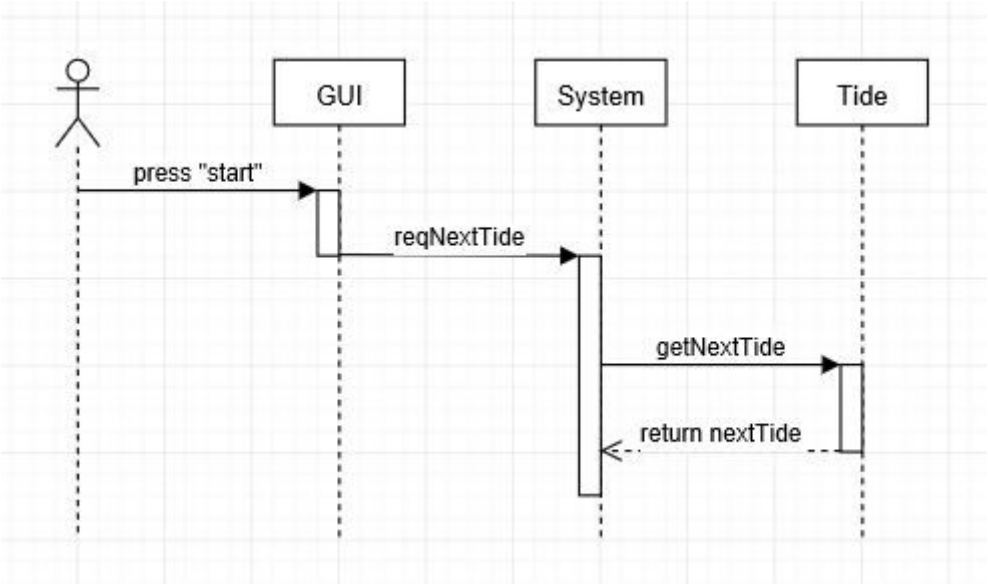
pause stopwatch

Name	reqPauseStopwatch
Responsibilities	Stopwatch모드의 “start”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.3.3
Notes	스탑워치를 잠시 멈춘다
Pre-conditions	runState가 1이어야 한다
Post-Conditions	N/A



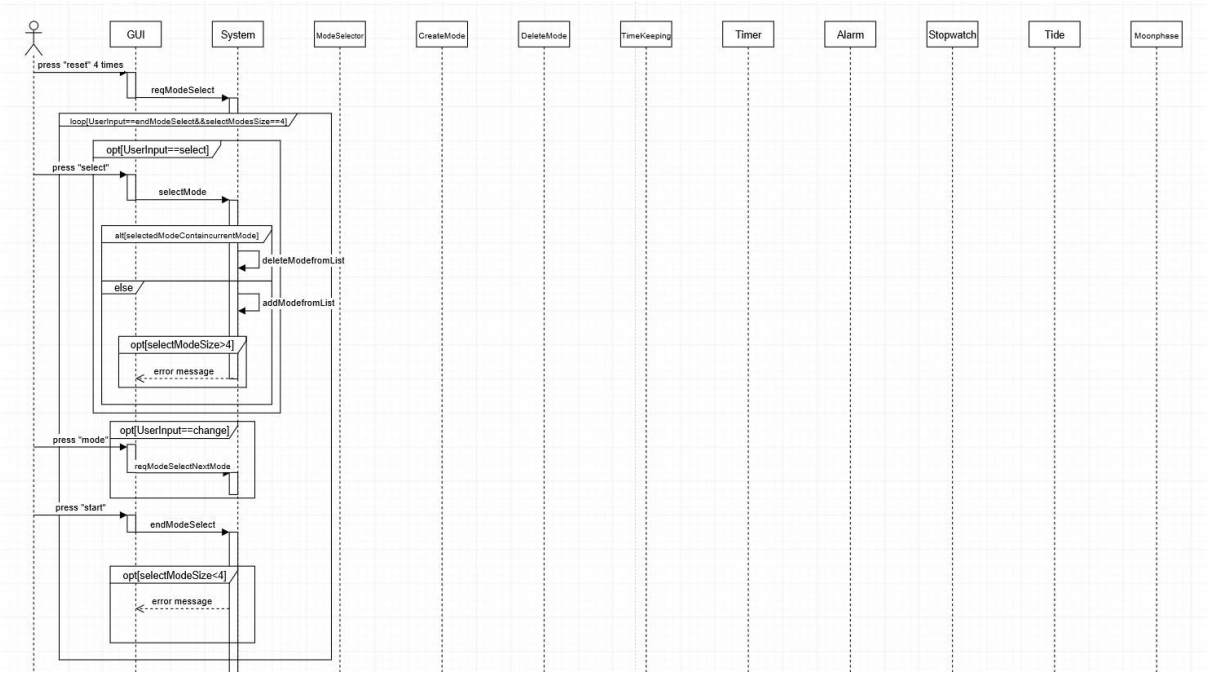
reset stopwatch

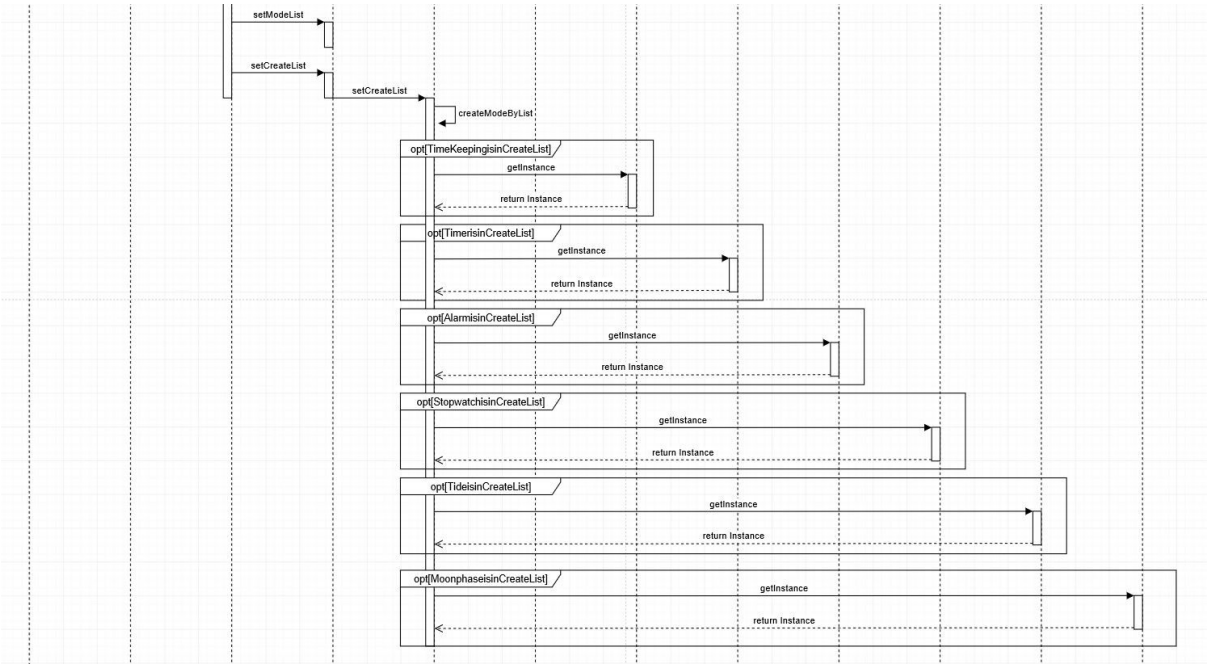
Name	reqResetStopwatch
Responsibilities	Stopwatch모드의 “reset”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.3.4
Notes	타이머가 0초로 리셋된다
Pre-conditions	Timer모드여야 한다 TimerState가 0이어야 한다
Post-Conditions	N/A

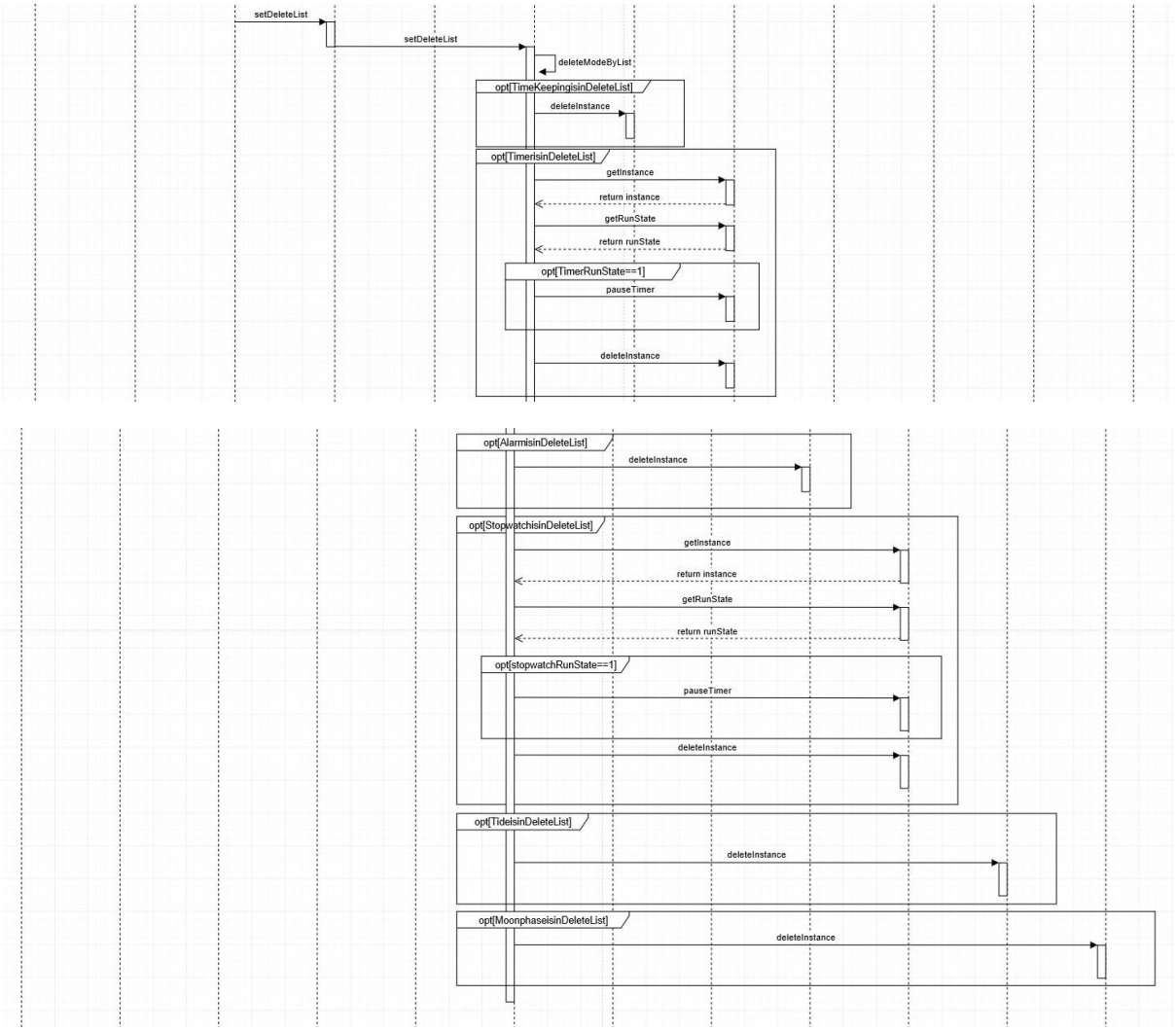


next tide

Name	reqNextStopwatch
Responsibilities	Tide모드의 “start”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.4.1
Notes	동해 서해 남해중 다음 바다의 조수를 보여준다
Pre-conditions	N/A
Post-Conditions	N/A


GUI
System
ModeSelector
CreateMode
DeleteMode
TimeKeeping
Timer
Alarm
Stopwatch
Tide
Moonphase
press "reset" 4 times
reqModeSelect
loop[UserInput==endModeSelect&&selectModesSize==4]
opt[UserInput==select]
press "select"
selectMode
alt[selectedModeContaincurrentMode]
deleteModefromList
else
addModefromList
opt[selectModeSize>4]
error message
opt[UserInput==change]
press "mode"
reqModeSelectNextMode
press "start"
endModeSelect
opt[selectModeSize<4]
error message


setModeList
setCreateList
setCreateList
createModeByList
opt[TimeKeepingisinCreateList]
getInstance
return Instance
opt[TimerisinCreateList]
getInstance
return Instance
opt[AlarmisinCreateList]
getInstance
return Instance
opt[StopwatchisinCreateList]
getInstance
return Instance
opt[TideisinCreateList]
getInstance
return Instance
opt[MoonphaseisinCreateList]
getInstance
return Instance



modeselect

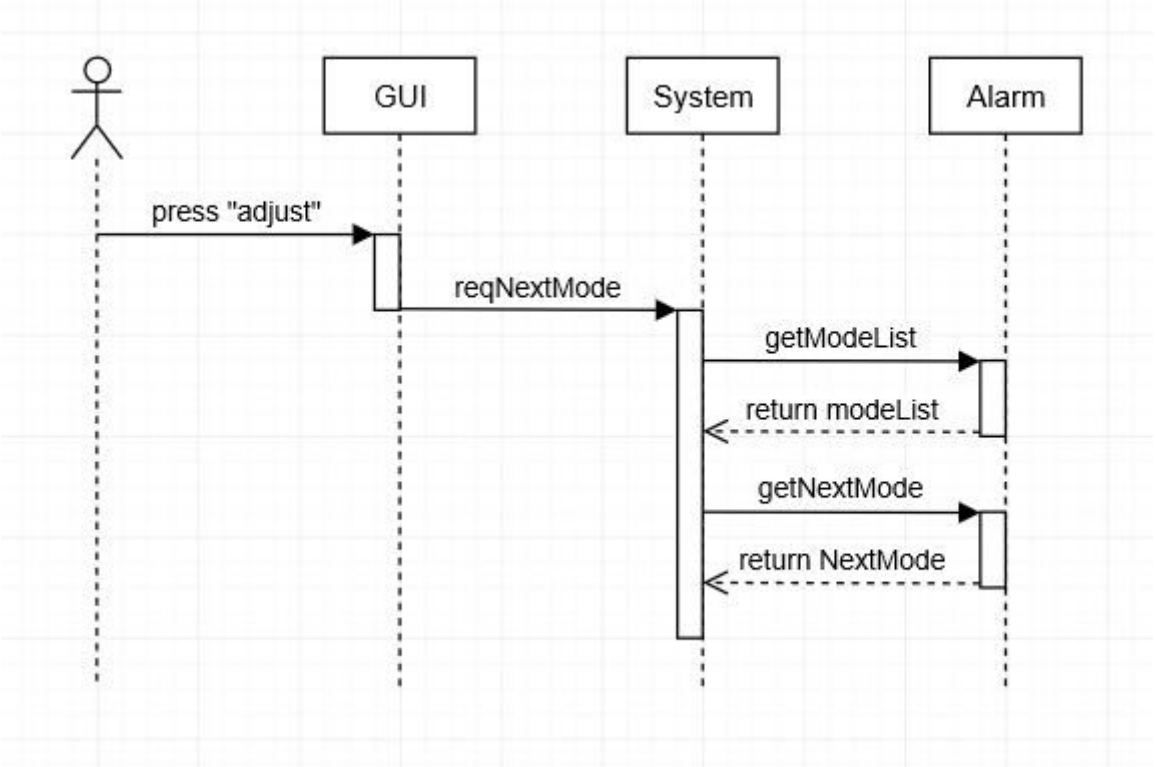
Name	reqModeSelect
Responsibilities	어떤 모드에서든지 “Reset”버튼을 4회 누른다
Type	GUI
Cross References	Funtional Requirements : R.6.0
Notes	모드셀렉트로 접근한다
Pre-conditions	Timer모드여야 한다 TimerState가 0이어야 한다
Post-Conditions	N/A

Name	SelectMode

Responsibilities	ModeSelect모드의 “start”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.6.0
Notes	사용하고 싶은 모드를 선택한다
Pre-conditions	N/A
Post-Conditions	4가지 모드 초과시 에러메세지를 출력한다

Name	reqModeSelectNextMode
Responsibilities	ModeSelect모드의 “mode”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.6.0
Notes	현재 모드의 다음모드를 선택할수 있게 해준다
Pre-conditions	N/A
Post-Conditions	N/A

Name	endModeSelect
Responsibilities	ModeSelect모드의 “start”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.6.0
Notes	선택한 모드를 적용시킨다
Pre-conditions	4가지 미만의 모드를 선택한경우 접근할수 없다
Post-Conditions	N/A



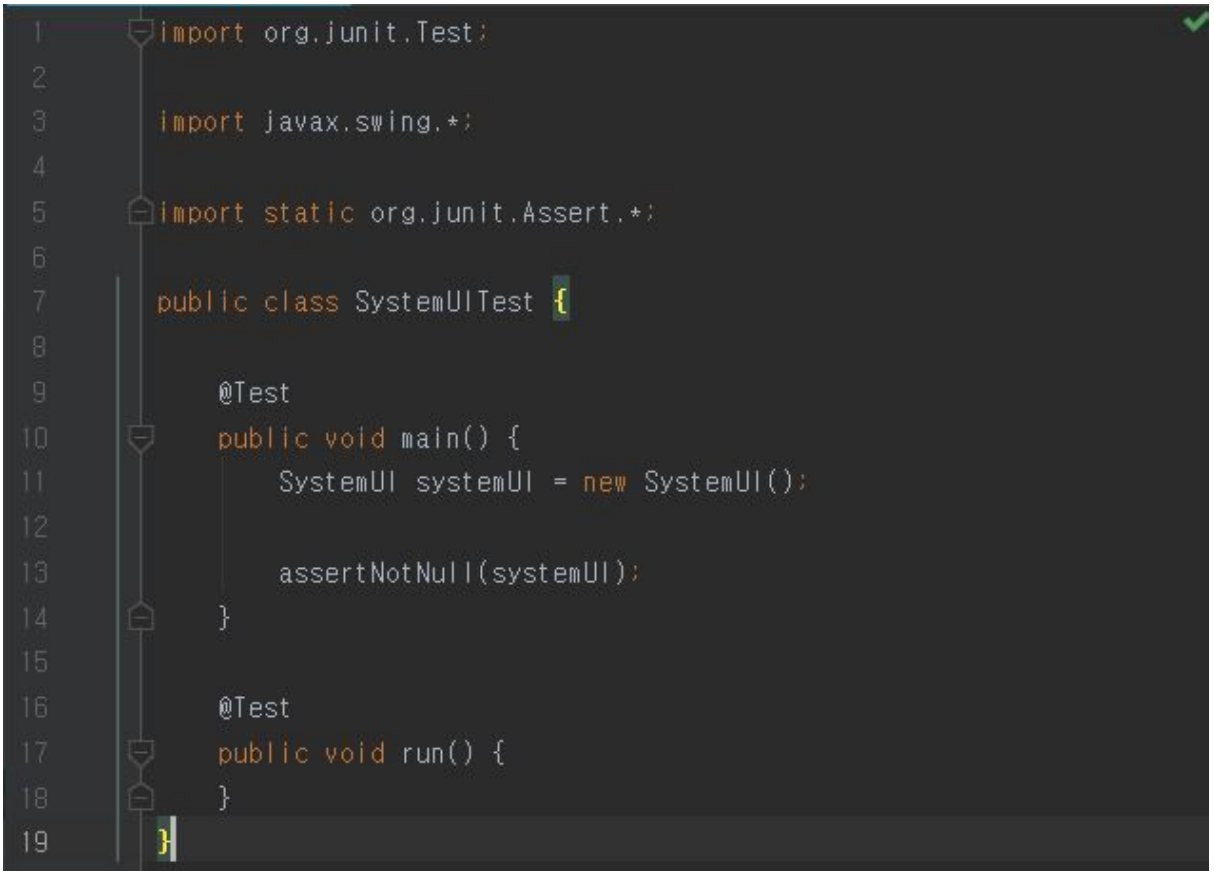
Name	reqNextMode
Responsibilities	“adjust”버튼을 누른다
Type	GUI
Cross References	Funtional Requirements : R.6.0
Notes	현재 모드의 다음 모드를 적용한다
Pre-conditions	N/A
Post-Conditions	N/A

Activity 2053. Implement Reports

OOPT Stage 1000, 2030, 2040, 2050 보고서를 통해 검토

Activity 2055. Write Unit Test Code

a. SystemUI



```java
import org.junit.Test;

import javax.swing.*;

import static org.junit.Assert.*;

public class SystemUITest {

    @Test
    public void main() {
        SystemUI systemUI = new SystemUI();

        assertNotNull(systemUI);
    }

    @Test
    public void run() {
    }
}
```

b. CreateMode

```java
import org.junit.Test;

import java.util.ArrayList;

import static org.junit.Assert.*;

public class CreateModeTest {

    @Test
    public void setCreateList() {

        CreateMode createMode = new CreateMode();
        ArrayList<String> arrayList=new ArrayList<>();
        arrayList.add("Stopwatch");
        arrayList.add("Alarm");

        Stopwatch stopwatch=Stopwatch.getInstance();
        Alarm alarm=Alarm.getInstance();
        Stopwatch.deleteInstance();
        Alarm.deleteInstance();

        createMode.setCreateList(arrayList);

        Stopwatch stopwatch1=Stopwatch.getInstance();
        Alarm alarm1= Alarm.getInstance();

        assertNotEquals(stopwatch, stopwatch1);
        assertNotEquals(alarm, alarm1);
    }
}
```

c. DeleteMode

```java
import org.junit.Test;

import java.util.ArrayList;

import static org.junit.Assert.*;

public class DeleteModeTest {

    @Test
    public void setDeleteList() {

        DeleteMode deleteMode = new DeleteMode();
        ArrayList<String> arrayList=new ArrayList<>();
        arrayList.add("Stopwatch");
        arrayList.add("Alarm");

        Stopwatch stopwatch=Stopwatch.getInstance();
        Alarm alarm=Alarm.getInstance();

        deleteMode.setDeleteList(arrayList);

        Stopwatch stopwatch1 =Stopwatch.getInstance();
        Alarm alarm1=Alarm.getInstance();

        assertNotEquals(stopwatch,stopwatch1);
        assertNotEquals(alarm, alarm1);
    }
}
```

d. ModeSelector

```java
public class ModeSelectorTest {


    @Test
    public void setSettingModeList() {
        ModeSelector modeSelector= new ModeSelector();

        ArrayList<String> settingModeList= new ArrayList<String>();
        settingModeList.add("TimeKeeping");
        settingModeList.add("Timer");
        settingModeList.add("Tide");
        settingModeList.add("Stopwatch");

        modeSelector.setSettingModeList(settingModeList);
        assertEquals(modeSelector.getModeList(), settingModeList);
    }
```

```java
29          @Test
30          public void getNextMode() {
31              ModeSelector modeSelector = new ModeSelector("TimeKeeping", "Timer", "Alarm", "Stopwatch");
32
33              String mode=modeSelector.getNextMode("TimeKeeping");
34              assertEquals(mode, "Timer");
35              mode=modeSelector.getNextMode("Timer");
36              assertEquals(mode, "Alarm");
37              mode=modeSelector.getNextMode("Alarm");
38              assertEquals(mode, "Stopwatch");
39              mode=modeSelector.getNextMode("Stopwatch");
40              assertEquals(mode, "TimeKeeping");
41          }
42
43          @Test
44          public void getDefaultNextMode() {
45              ModeSelector modeSelector=new ModeSelector();
46
47              String nextMode= modeSelector.getDefaultNextMode("Timer");
48              assertEquals(nextMode, "Alarm");
49              nextMode=modeSelector.getDefaultNextMode("Alarm");
50              assertEquals(nextMode, "Stopwatch");
51              nextMode=modeSelector.getDefaultNextMode("Stopwatch");
52              assertEquals(nextMode, "Tide");
53              nextMode=modeSelector.getDefaultNextMode("Tide");
54              assertEquals(nextMode, "Moonphase");
55              nextMode=modeSelector.getDefaultNextMode("Moonphase");
56              assertEquals(nextMode, "TimeKeeping");
57              nextMode=modeSelector.getDefaultNextMode("TimeKeeping");
58              assertEquals(nextMode, "Timer");
59          }
60
```

```
60
61          @Test
62          public void setCreateList() {
63
64          }
65
66          @Test
67          public void setDeleteList() {
68          }
69      }
```

e. TimeKeeping

```
1   import org.junit.Test;
2
3   import static org.junit.Assert.*;
4
5   public class TimeKeepingTest {
6
7       @Test
8       public void getInstance() {
9           TimeKeeping timeKeeping = TimeKeeping.getInstance();
10
11          TimeKeeping.deleteInstance();
12
13          TimeKeeping timeKeeping1 = TimeKeeping.getInstance();
14
15          assertNotEquals(timeKeeping, timeKeeping1);
16      }
17
18      @Test
19      public void deleteInstance() {
20          TimeKeeping timeKeeping = TimeKeeping.getInstance();
21
22          TimeKeeping.deleteInstance();
23
24          TimeKeeping timeKeeping1 = TimeKeeping.getInstance();
25
26          assertNotEquals(timeKeeping, timeKeeping1);
27      }
28
29      @Test
30      public void setTime() {
31          TimeKeeping timeKeeping= TimeKeeping.getInstance();
32
33          timeKeeping.setTime("2015 3 30 9 15 20");
34          String tmp = timeKeeping.getTime();
```

```
35
36          assertEquals(tmp, "2015 3 30 9 15 0");
37      }
38
39      @Test
40      public void getTime() {
41          TimeKeeping timeKeeping= TimeKeeping.getInstance();
42
43          timeKeeping.setTime("2015 3 30 9 15 20");
44          String tmp = timeKeeping.getTime();
45
46          assertEquals(tmp, "2015 3 30 9 15 0");
47      }
48  }
```

f. TimeDB

```java
import org.junit.Test;

import java.util.HashMap;

import static org.junit.Assert.*;

public class TimeDBTest {

    @Test
    public void getInstance() {
        TimeDB timeDB = TimeDB.getInstance();

        assertNotNull(timeDB);
    }

    @Test
    public void setTime() {
        TimeDB timeDB = TimeDB.getInstance();

        String time = "2015 11 20 2 15";
        timeDB.setTime(time);
        String tmp=timeDB.getTime();

        time= time+ " 0";
        assertEquals(tmp, time);
    }
```

```java
    @Test
    public void setMonthMap() {
        TimeDB timeDB = TimeDB.getInstance();

        timeDB.setMonthMap(2019);

        HashMap<Integer, Integer> tmpMap = new HashMap<>();

        tmpMap=timeDB.getMonthMap();

        tmpMap.get(2);
        assertEquals(28+"", tmpMap.get(2)+"");
    }

    @Test
    public void getTime() {
        TimeDB timeDB= TimeDB.getInstance();

        String time="2015 3 30 9 30";
        timeDB.setTime("2015 3 30 09 30");

        time=time+" 0";
        assertEquals(time, timeDB.getTime());
    }

    @Test
    public void getMonthMap() {
        TimeDB timeDB=TimeDB.getInstance();

        timeDB.setMonthMap(2019);
        HashMap<Integer, Integer> tmpMap = new HashMap<>();

        tmpMap=timeDB.getMonthMap();
        assertEquals(tmpMap.get(2)+"", 28+"");
```

```
61          assertEquals(tmpMap.get(2)+"", 28+"");
62      }
63
64      @Test
65      public void updateTime() {
66          TimeDB timeDB=TimeDB.getInstance();
67
68          String time = "2015 3 30 9 30 10";
69          timeDB.setTime(time);
70
71          timeDB.updateTime();
72          String tmp=timeDB.getTime();
73
74          String upTime="2015 3 30 9 30 0";
75          assertEquals(upTime, tmp);
76      }
77
78      @Test
79      public void startUpdateTime() {
80      }
81
82      @Test
83      public void pauseTimeDB() {
84      }
85
86      @Test
87      public void run() {
88      }
89  }
```

g. Alarm

```
public class AlarmTest {

    @Test
    public void getInstance() {
        Alarm alarm = Alarm.getInstance();

        assertNotNull(alarm);
    }

    @Test
    public void deleteInstance() {
        Alarm alarm=Alarm.getInstance();

        alarm.deleteInstance();

        Alarm alarm1=Alarm.getInstance();

        assertNotEquals(alarm, alarm1);
    }

    @Test
    public void addAlarm() {
        Alarm alarm = Alarm.getInstance();

        alarm.addAlarm( alarmHour: 1,  alarmMinute: 2,  index: 3);

        assertEquals(alarm.getAlarmList().get(3),  actual: "1 2 0");
    }
```

```java
@Test
public void deleteAlarm() {
    Alarm alarm = Alarm.getInstance();

    alarm.addAlarm( alarmHour: 1,  alarmMinute: 2,  index: 3);
    alarm.deleteAlarm( index: 3);

    assertEquals(alarm.getAlarmList().get(3),  actual: null);

}

@Test
public void stopAlarm(){
    Alarm alarm = Alarm.getInstance();

    alarm.stopAlarm();

    assertEquals(alarm.stopAlarm(),  actual: false);
}
```

i. Stopwatch

```java
import javafx.scene.paint.Stop;
import org.junit.Test;

import static org.junit.Assert.*;

public class StopwatchTest {

    @Test
    public void getInstance() {
        Stopwatch stopwatch=Stopwatch.getInstance();

        Stopwatch.deleteInstance();

        Stopwatch stopwatch1=Stopwatch.getInstance();

        assertNotEquals(stopwatch, stopwatch1);
    }

    @Test
    public void deleteInstance() {
        Stopwatch stopwatch=Stopwatch.getInstance();

        Stopwatch.deleteInstance();

        Stopwatch stopwatch1=Stopwatch.getInstance();

        assertNotEquals(stopwatch, stopwatch1);
    }

    @Test
    public void getRunState() {
        Stopwatch stopwatch =Stopwatch.getInstance();

        assertEquals(0, stopwatch.getRunState());
```

```
    }

    @Test
    public void getZeroSate() {
        Stopwatch stopwatch=Stopwatch.getInstance();

        assertEquals(0, stopwatch.getZeroSate() );
    }

    @Test
    public void getTime() {
        Stopwatch stopwatch=Stopwatch.getInstance();

        stopwatch.setStopwatch("11 1 2 3");

        String tmp=stopwatch.getTime();

        assertEquals("11 1 2 3", tmp);
    }

    @Test
    public void setStopwatch() {
        Stopwatch stopwatch=Stopwatch.getInstance();

        stopwatch.setStopwatch("11 1 2 3");

        String tmp=stopwatch.getTime();

        assertEquals("11 1 2 3", tmp);
    }

```

```
    @Test
    public void startStopwatch() {
        Stopwatch stopwatch=Stopwatch.getInstance();

        stopwatch.startStopwatch();
        int tmp=stopwatch.getRunState();

        assertEquals(tmp, 1);
    }

    @Test
    public void recordStopwatch() {
        Stopwatch stopwatch = Stopwatch.getInstance();

        stopwatch.setStopwatch("10 2 12 23");

        String tmp=stopwatch.recordStopwatch();

        assertEquals(tmp, "10 2 12 23");
    }

    @Test
    public void pauseStopwatch() {
        Stopwatch stopwatch=Stopwatch.getInstance();
        stopwatch.startStopwatch();
        stopwatch.pauseStopwatch();

        int runState = stopwatch.getRunState();

        assertEquals(runState, 0);
    }

```

```
    @Test
    public void resetStopwatch() {
        Stopwatch stopwatch=Stopwatch.getInstance();

        stopwatch.startStopwatch();
        stopwatch.pauseStopwatch();
        stopwatch.resetStopwatch();

        assertEquals(1, stopwatch.getZeroSate());
    }

    @Test
    public void updateTime() {
        Stopwatch stopwatch=Stopwatch.getInstance();

        stopwatch.setStopwatch("11 11 11 1");
        stopwatch.updateTime();
        String tmp=stopwatch.recordStopwatch();

        assertEquals(tmp, "11 11 11 2");
    }

}
```

j. Moonphase

```
import org.junit.Test;

import java.text.ParseException;
import java.text.SimpleDateFormat;
import java.util.Date;

import static org.junit.Assert.*;

public class MoonphaseTest {

    @Test
    public void getInstance() {
        Moonphase moonphase=Moonphase.getInstance();

        Moonphase.deleteInstance();

        Moonphase moonphase1=Moonphase.getInstance();

        assertNotEquals(moonphase, moonphase1);
    }

    @Test
    public void deleteInstance() {
        Moonphase moonphase=Moonphase.getInstance();

        Moonphase.deleteInstance();

        Moonphase moonphase1=Moonphase.getInstance();

        assertNotEquals(moonphase, moonphase1);
    }
```

```
32
33      @Test
34      public void showMoonphase() {
35      }
36
37      @Test
38      private void calculateMoonphase() {
39      }
40  }
```

k. Tide

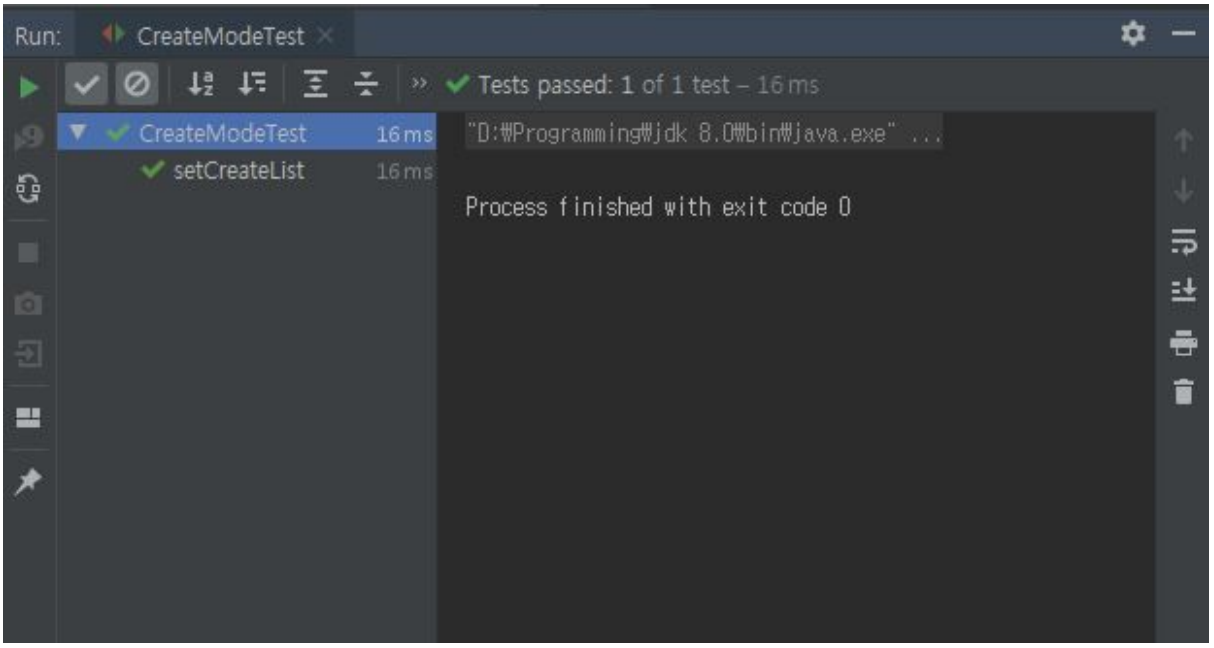
```
1   import org.junit.Test;
2
3   import static org.junit.Assert.*;
4
5   public class TideTest {
6
7       @Test
8       public void getInstance() {
9           Tide tide = Tide.getInstance();
10
11          Tide.deleteInstance();
12
13          Tide tide1=Tide.getInstance();
14
15          assertNotEquals(tide, tide1);
16      }
17
18      @Test
19      public void deleteInstance() {
20          Tide tide = Tide.getInstance();
21
22          Tide.deleteInstance();
23
24          Tide tide1=Tide.getInstance();
25
26          assertNotEquals(tide, tide1);
27      }
28
29      @Test
30      public void showTide() {
31      }
32
```

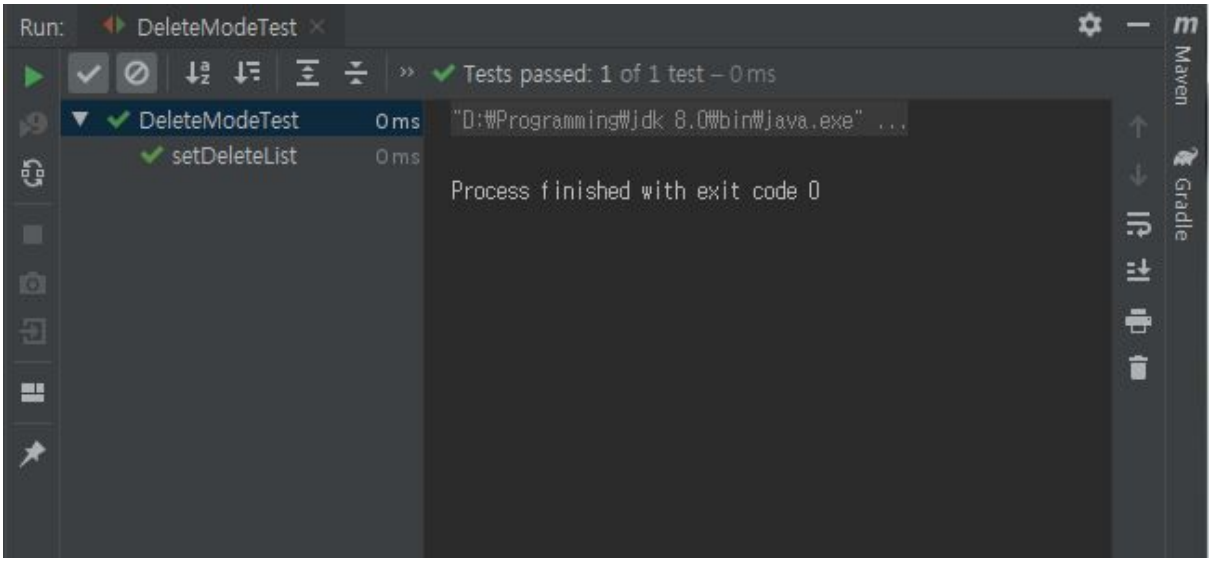
```
33      @Test
34      public void getNextTide() {
35      }
36
37      @Test
38      public void getTideList() {
39      }
40  }
```

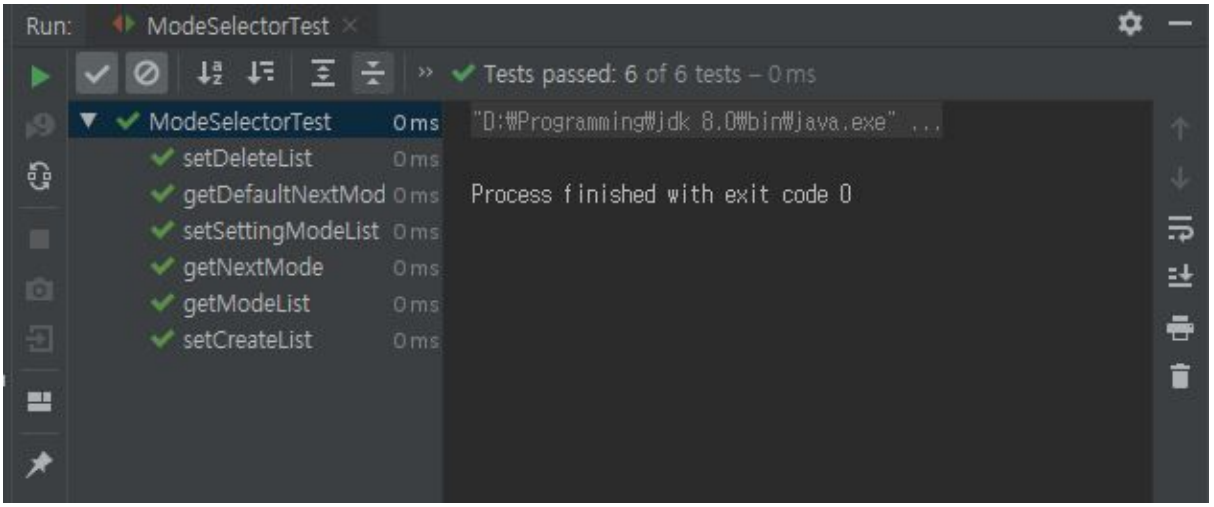
I. Timer

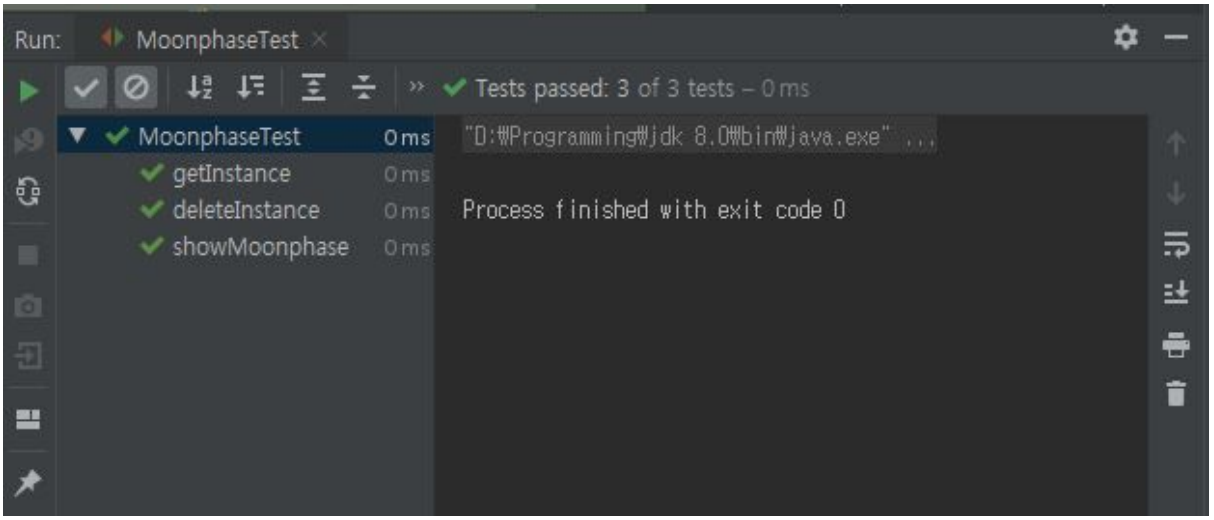
```java
import org.junit.Test;

import static org.junit.Assert.*;

public class TimerTest {

    @Test
    public void getInstance() {
        Timer timer=Timer.getInstance();
        Timer.deleteInstance();

        Timer timer1=Timer.getInstance();
        assertNotEquals(timer, timer1);
    }

    @Test
    public void deleteInstance() {
        Timer timer=Timer.getInstance();
        Timer.deleteInstance();

        Timer timer1=Timer.getInstance();
        assertNotEquals(timer, timer1);
    }

    @Test
    public void getRunState() {
    }

    @Test
    public void getZeroState() {
        Timer timer=Timer.getInstance();
        assertEquals( expected: 0, timer.getZeroState());
    }

    @Test
    public void getTime() {
        Timer timer=Timer.getInstance();
        timer.setTimer("11 9 5");
        String tmp=timer.getTime();
        assertEquals( expected: "11 9 5", tmp);
    }

    @Test
    public void pauseTimer() {
        Timer timer=Timer.getInstance();
        timer.startTimer();
        timer.pauseTimer();

        int runState = timer.getRunState();

        assertEquals(runState,  actual: 0);
    }

    @Test
    public void setTimer() {
        Timer timer=Timer.getInstance();
        timer.setTimer("11 9 5");
        String tmp=timer.getTime();
        assertEquals( expected: "11 9 5", tmp);
    }

    @Test
    public void startTimer() {
        Timer timer=Timer.getInstance();

        timer.startTimer();
        int tmp=timer.getRunState();

        assertEquals(tmp,  actual: 1);
    }

    @Test
    public void resetTimer() {
        Timer timer=Timer.getInstance();
        timer.startTimer();
        timer.pauseTimer();
        timer.resetTimer();
        assertEquals( expected: 1, timer.getZeroState());
    }

    @Test
    public void buzzTimer() {
    }

    @Test
    public void updateTime() {
        Timer timer=Timer.getInstance();

        timer.setTimer("11 11 11");
        for(int i=0;i<10;i++)
            timer.updateTime();
        String tmp=timer.getTime();

        assertEquals(tmp,  actual: "11 11 10");
    }
}
```

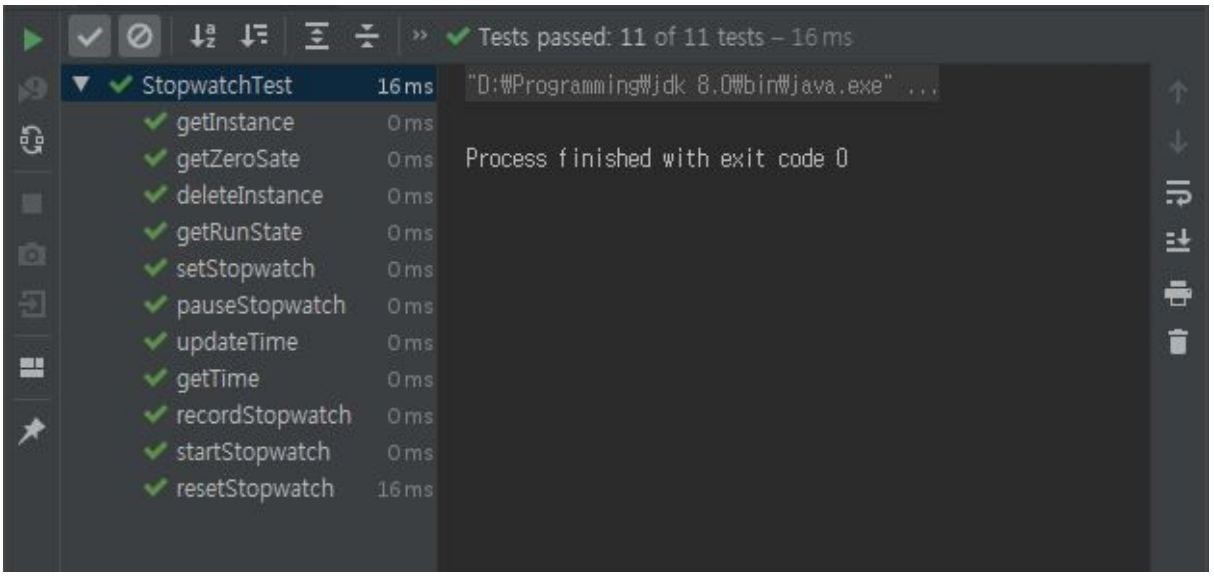
Activity 2061. Unit Testing

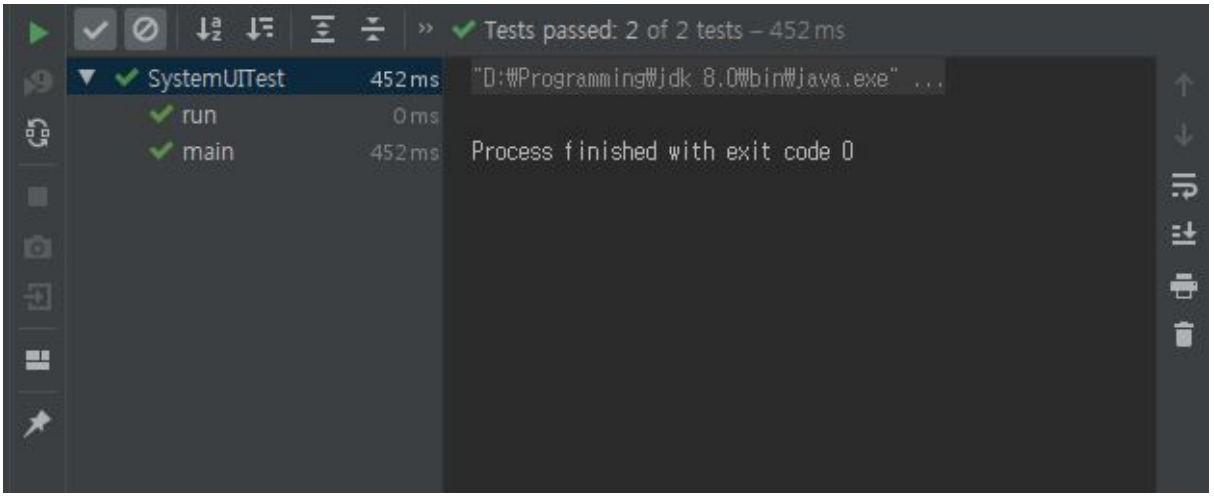


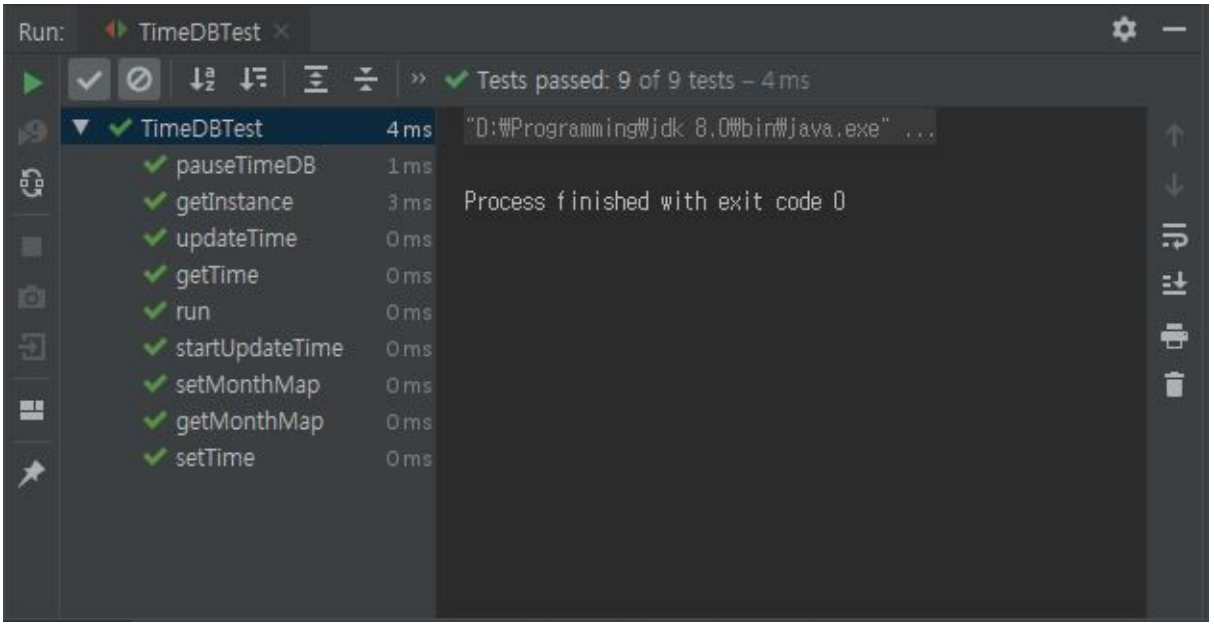


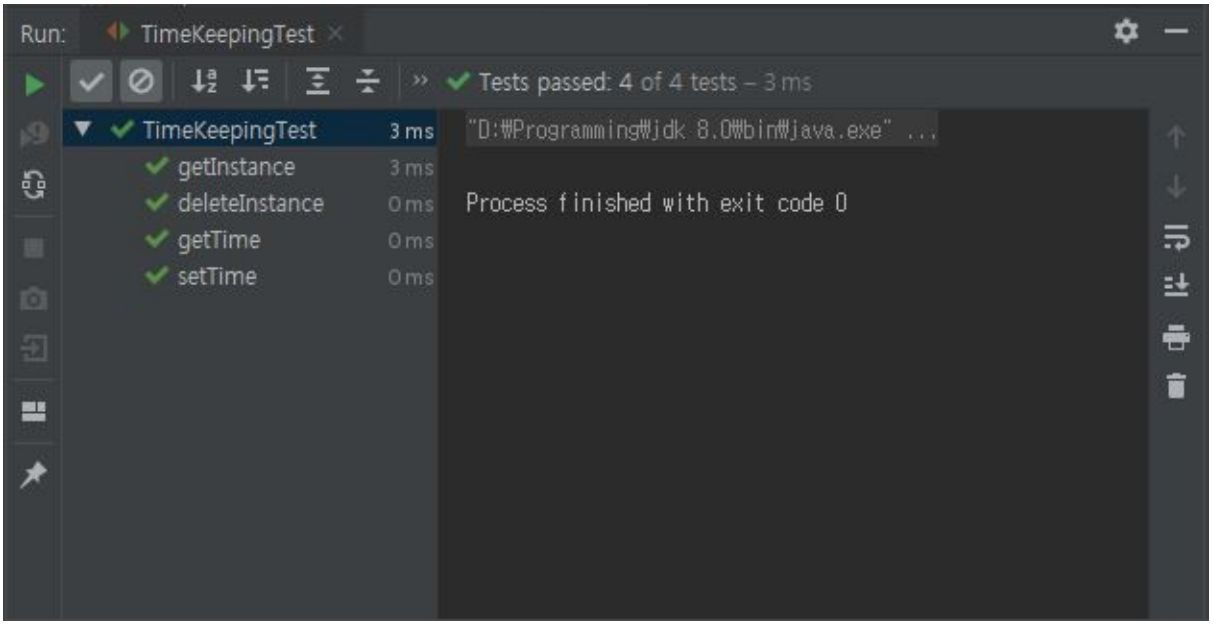

Run: ModeSelectorTest
Tests passed: 6 of 6 tests – 0 ms
ModeSelectorTest 0 ms
setDeleteList 0 ms
getDefaultNextMod 0 ms
setSettingModeList 0 ms
getNextMode 0 ms
getModeList 0 ms
setCreateList 0 ms
"D:₩Programming₩jdk 8.0₩bin₩java.exe" ...
Process finished with exit code 0

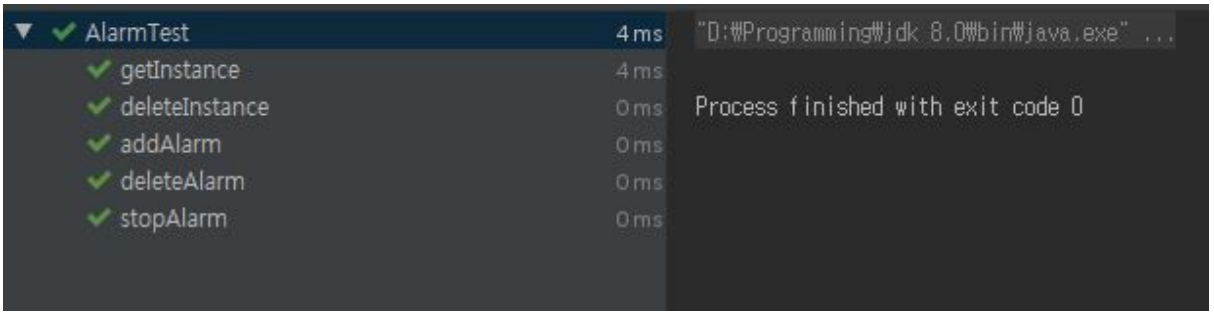

Run: MoonphaseTest
Tests passed: 3 of 3 tests – 0 ms
MoonphaseTest 0 ms
getInstance 0 ms
deleteInstance 0 ms
showMoonphase 0 ms
"D:₩Programming₩jdk 8.0₩bin₩java.exe" ...
Process finished with exit code 0


Tests passed: 11 of 11 tests – 16 ms
StopwatchTest 16 ms
getInstance 0 ms
getZeroSate 0 ms
deleteInstance 0 ms
getRunState 0 ms
setStopwatch 0 ms
pauseStopwatch 0 ms
updateTime 0 ms
getTime 0 ms
recordStopwatch 0 ms
startStopwatch 0 ms
resetStopwatch 16 ms
"D:₩Programming₩jdk 8.0₩bin₩java.exe" ...
Process finished with exit code 0


Tests passed: 2 of 2 tests – 452 ms
SystemUITest 452 ms
run 0 ms
main 452 ms
"D:₩Programming₩jdk 8.0₩bin₩java.exe" ...
Process finished with exit code 0


Run:
TimeDBTest
Tests passed: 9 of 9 tests – 4 ms
TimeDBTest 4 ms
pauseTimeDB 1 ms
getInstance 3 ms
updateTime 0 ms
getTime 0 ms
run 0 ms
startUpdateTime 0 ms
setMonthMap 0 ms
getMonthMap 0 ms
setTime 0 ms
"D:₩Programming₩jdk 8.0₩bin₩java.exe" ...
Process finished with exit code 0


Run:
TimeKeepingTest
Tests passed: 4 of 4 tests – 3 ms
TimeKeepingTest 3 ms
getInstance 3 ms
deleteInstance 0 ms
getTime 0 ms
setTime 0 ms
"D:₩Programming₩jdk 8.0₩bin₩java.exe" ...
Process finished with exit code 0


AlarmTest 4 ms
getInstance 4 ms
deleteInstance 0 ms
addAlarm 0 ms
deleteAlarm 0 ms
stopAlarm 0 ms
"D:₩Programming₩jdk 8.0₩bin₩java.exe" ...
Process finished with exit code 0


Run:
TimerTest
Tests passed: 11 of 11 tests – 44 ms
TimerTest 44 ms
resetTimer 39 ms
getInstance 0 ms
startTimer 1 ms
deleteInstance 1 ms
getRunState 0 ms
updateTime 0 ms
"C:₩Program Files₩Java₩jdk1.8.0_211₩bin₩java.exe" ...
Process finished with exit code 0
9: Version Control
Terminal
Build
0: Messages
4: Run
6: TODO

Activity 2063. System Testing

Test Number	Test 항목	Description	Use Case	System Function	P/F
1	getInstance	인스턴스를 통해 객체를 가져오는지 확인	getInstance	R6.0	p
2	deleteInstance	인스턴스를 통해 가져온 객체를 삭제하는지 확인	deleteInstance	R6.0	p
3	getTime Test	시간이 정상적으로 1초마다 업데이트 되는지 확인한다.	getTime	R0.0	p
4	setTime Test	adjustTime으로 새롭게 시간을 설정한 후 설정한 시간이 잘 업데이트되는지 확인한다.	setTime	R0.1	p
5	showAlarm Test	화면에 알람목록의 첫번째 알람이 정상적으로 출력되는지 확인한다	showAlarm	R2.1	p
6	nextAlarm Test	버튼 입력에 따라 현재 알람에서 다음알람으로 제대로 넘어가는지 확인한다	nextAlarm	R2.2	p
7	addAlarm Test	알람을 설정 가능할경우에만 알람을 새롭게 설정하고 알람목록에 알람이 정상적으로 갱신되는지 확인한다	addAlarm	R2.3	p
8	deleteAlarm Test	현재 지우려고 하는 알람이 정상적으로 지워지는지 확인하고 알람이 4개 설정되어 있을때 알람이 삭제하는 것만 가능한지 확인한다	deleteAlarm	R2.4	p
9	buzzAlarm Test	알람이 설정된 시간이 되었을 때에 맞춰 잘 울리는지 확인하고 바뀐 알람상태를 잘 반환해주는지 확인한다	buzzAlarm	R2.5	p

10	StopAlarm Test	알람이 울릴 때 어떤 모드에서든지 버튼 입력이 있으면 울리던 알람이 즉각적으로 잘 멈추는지 확인한다.	stopAlarm	R2.6	p
11	getZerostate	기능이 실행되면 해당 기능의 runstate가 0으로 바뀌게 만드는지 확인한다	pauseTimer buzzAlarm pauseStopwatch	R1.3 R2.4 R3.3	p
12	getRunstate	기능이 실행되면 해당기능의 runstate가 으로 바뀌게 만드는지 확인한다	startTimer' stopAlarm startStopwatch	R1.2 R2.5 R3.1	p
13	getTime	스탑워치로 보낸 시간이 똑같이 출력되는지 확인한다	showStopwatch	R3.0	p
14	updateTime	갱신된 시간이 스탑워치에도 잘 적용되는지 확인한다	showStopwatch	R3.0	p
15	set Stopwatch Test	스톱워치의 현재 시간이 정상적으로 보이는지 확인한다	set Stopwatch	R3.1	p
16	start Stopwatch Test	버튼 입력에 따라 스톱워치가 정상적으로 시간을 갱신하는지 확인한다	start Stopwatch	R3.2	p
17	record Stopwatch Test	스톱워치가 동작 중일 때 버튼을 입력하면 정확하게 버튼을 입력한 때에 나타났던 시간이 잘 기록되는지 확인한다.	record Stopwatch	R3.3	p
18	pause Stopwatch Test	스톱워치가 동작 중일 때 버튼을 입력하면 정확하게 버튼을 입력한 때에 잘 멈추는지 확인한다.	pause Stopwatch	R3.4	p
19	reset Stopwatch Test	스톱워치 모드에서 스톱워치의 동작 중 여부에 관계없이 버튼을 입력하면 스톱워치가 정상적으로 초기화되는지 확인한다.	reset Stopsatch	R3.5	p
20	show Moonphas	날짜에 맞게 달의 위상이 정확하게 나타나는지	show Moonphas	R5.1	p

	e Test	확인한다.	e		
21	Moonphase Test	현재 날짜를 음력으로 바꾸고 음력날짜에 맞는 달 모양의 그래픽으로 바꿔주는지 확인한다	calculate Moonpahse	R5.2	p
22	modeSelect Test	모드들을 선택할때 정상적으로 체크/체크해제가 되는지 확인한다 모드들을 정상적으로 4개를 선택해야 modeSelect를 빠져나갈 수 있는지 확인한다	modeSelect	R6.1	p
23	setDeleteList	설정한 모드가 담긴 리스트가 삭제되는지 확인한다	modeSelect	R6.0	p
24	getDefault NextMod	ModeSelect화면에 모든 6개의 모드가 나오는지 확인한다	modeSelect	R6.0	p
25	setSetting ModeList	ModeDelect에서 설정한 모드와 ModeList와 같은지 확인	modeSelect	R6.0	p
26	getNextMode	ModeList의 다음모드가 잘 적용되는지 확인	modeSelect	R6.0	p
27	getModeList	ModeList가 정상적으로 불러와지는지 확인한다	modeSelect	R6.0	p
28	setCreateList	ModeList가 정상적으로 설정되었는지 확인한다	modeSelect	R6.0	p
29	pauseTime DB	시간을 조정하는 동안 시간 갱신을 잠시 멈추는 기능이 정상적으로 작동하는지 확인한다	adjustTime	R0.1	p
30	getTime	TimeDB에 설정한 시간이 설정하려던 시간과 일치하는지 확인	adjustTime	R0.1	p
31	updateTime	사용자가 설정한 시간이 잘 갱신이 되었는지 확인한다	adjustTime	R0.1	p
32	startUpdateTime	시간갱신이 잘 시작는지 확인한다	showTime	R0.0	p
33	setMonthMap	Monthmap이 잘 구성이 되는지 확인한다	showTime	R0.0	p

34	getMonthMap	Monthmap이 잘 받아와지는지확인한다	showTime	R0.0	p
35	setTime	설정한 날짜를 연월일 나누어 정확히 저장하는지 확인한다	adjustTime	R0.1	p

2063. Testing Traceability Analysis