

언제 할지 알려주시게

OOPT Stage 2050, 2060

Project Team
T5

Date
2019-05-27

Team Information

201511243 김동연

201511262 박우진

201511284 이종빈

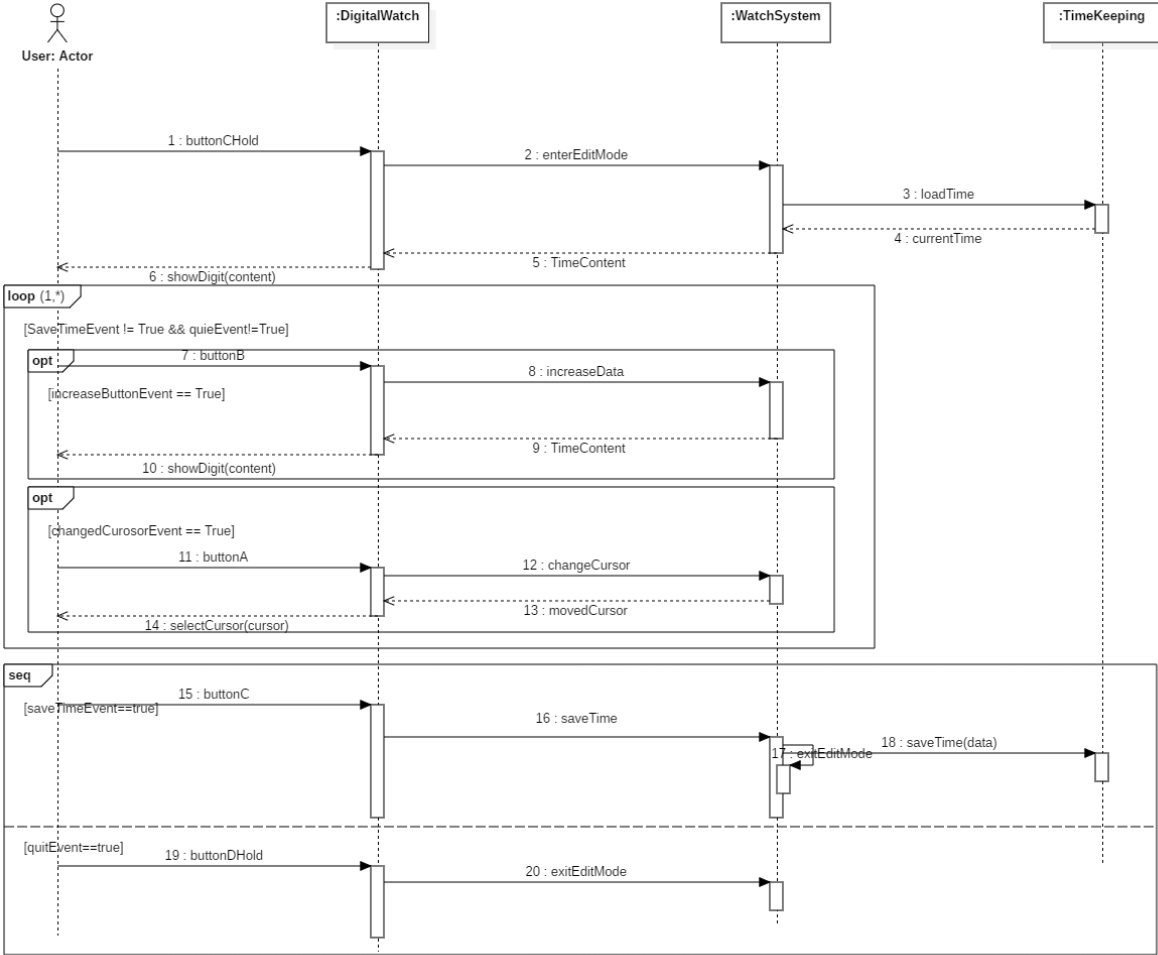
201511295 조범석

Activity 2052. Implement Windows	3
Activity 2055. Write Test Code	24
Activity 2061. Unit Testing	32
Activity 2063. System Testing	32
Activity 2067. Testing Traceability Analysis	34

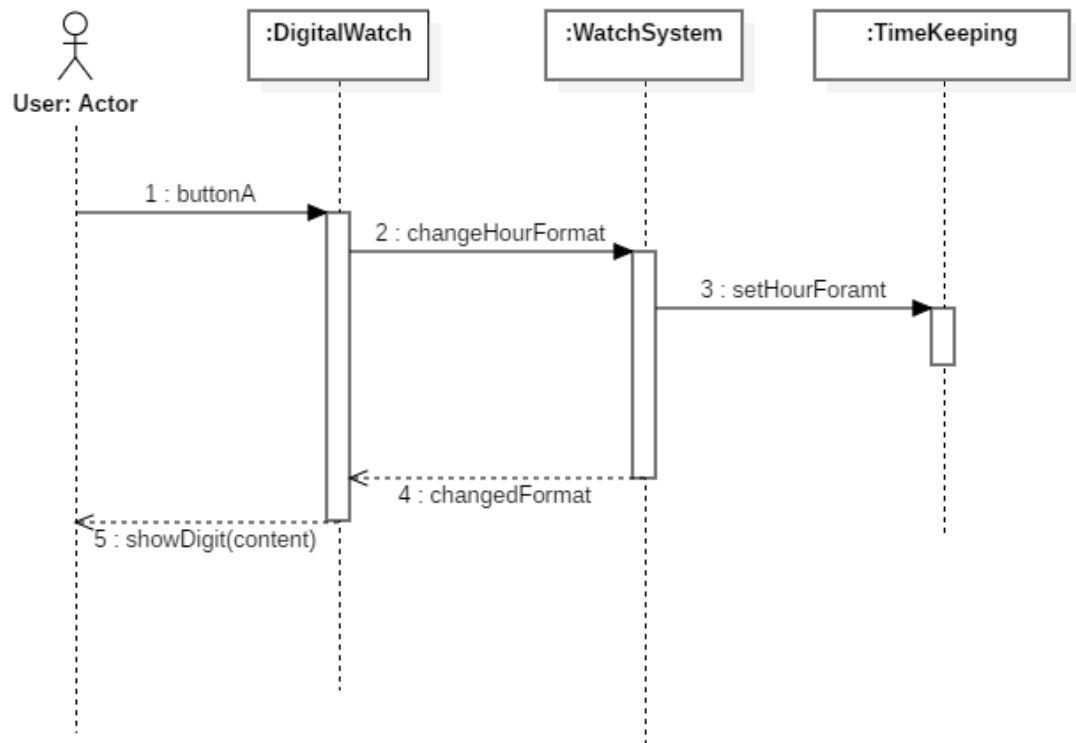
1. Activity 2052. Implement Windows

A. UI Layer, Application Logic Layer Mapping Interaction Diagram

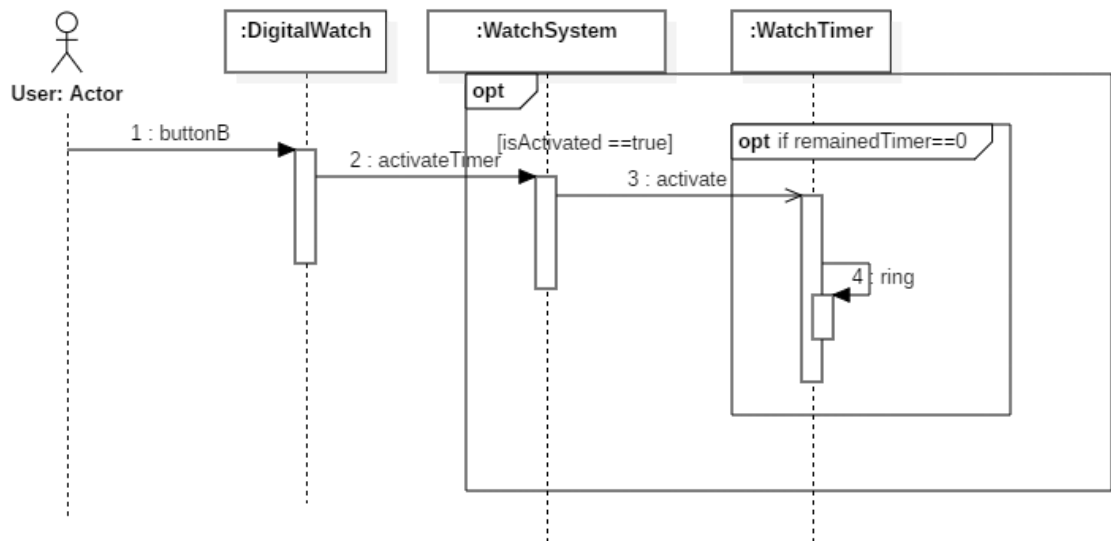
1. Set Time



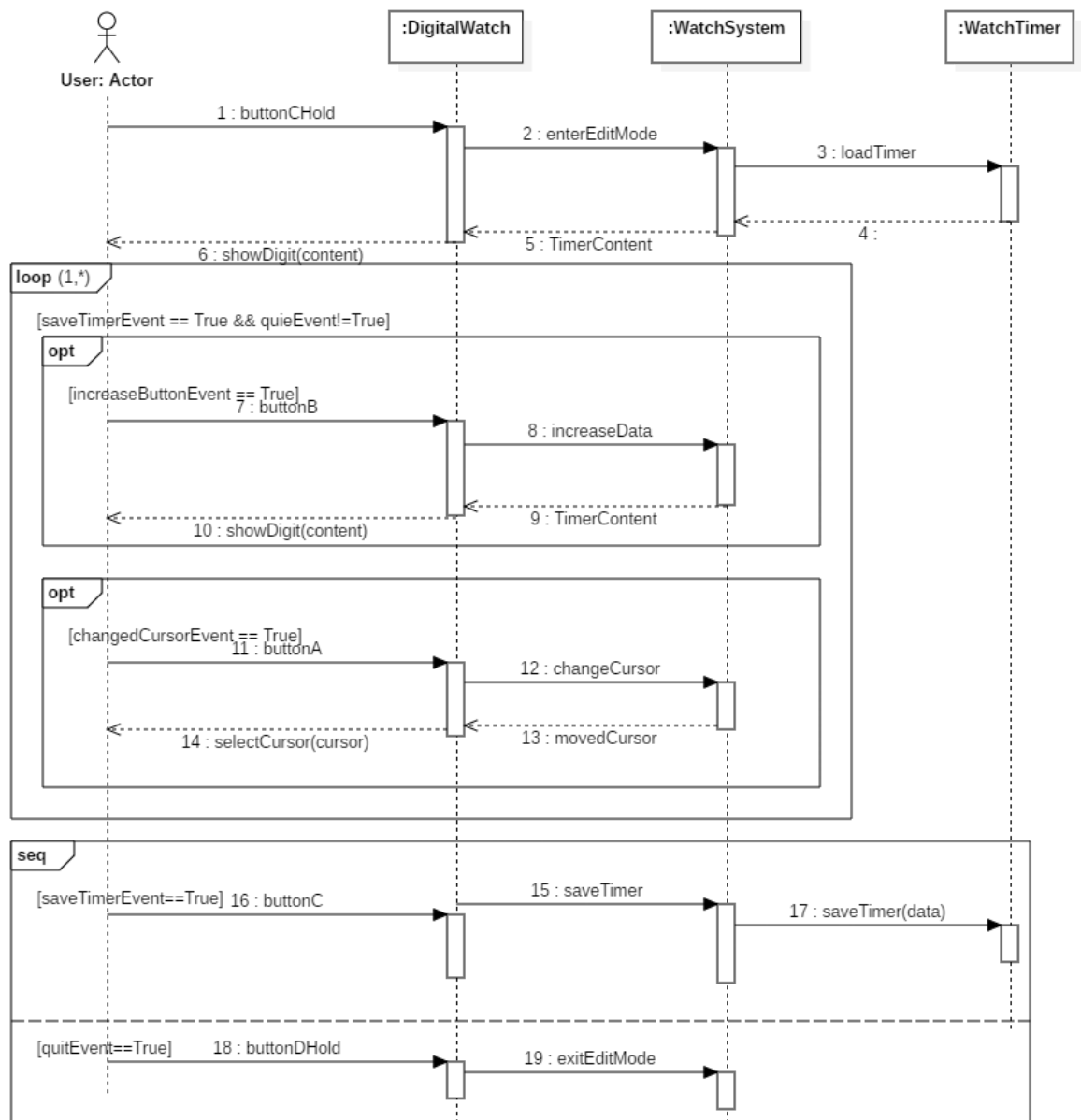
2. Set Hour Format



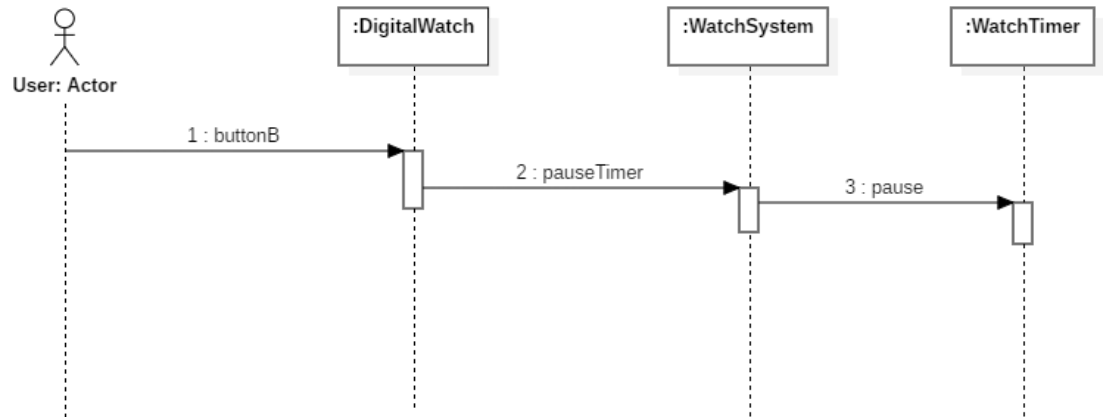
3. Activate Timer



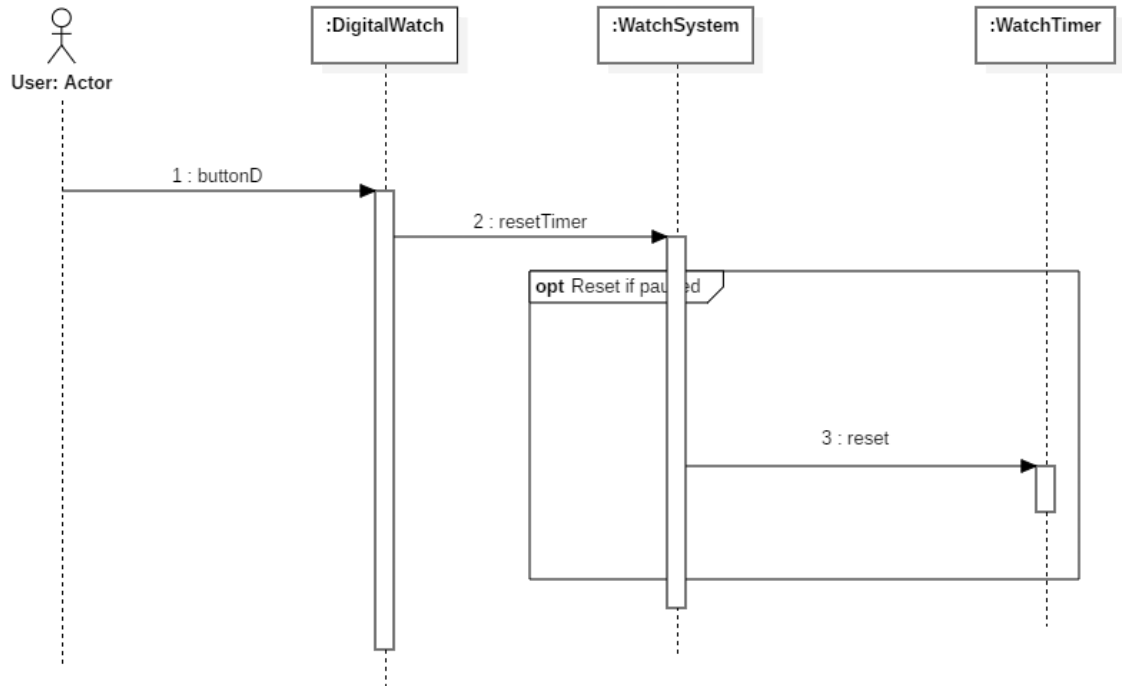
4. Set Timer



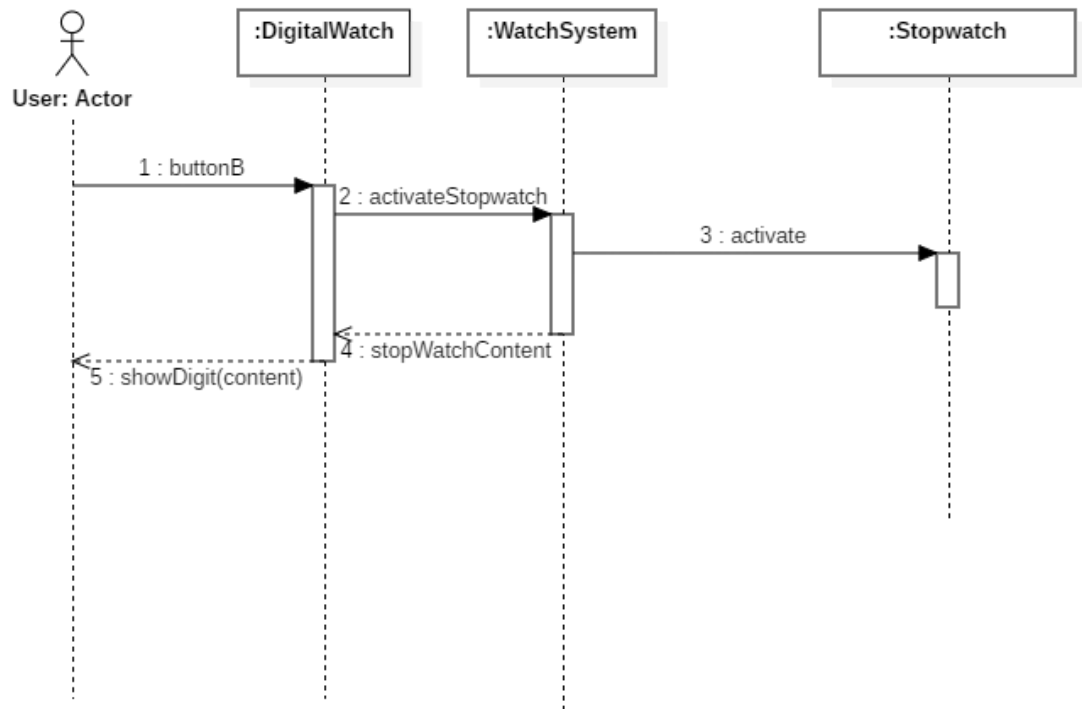
6. Pause Timer



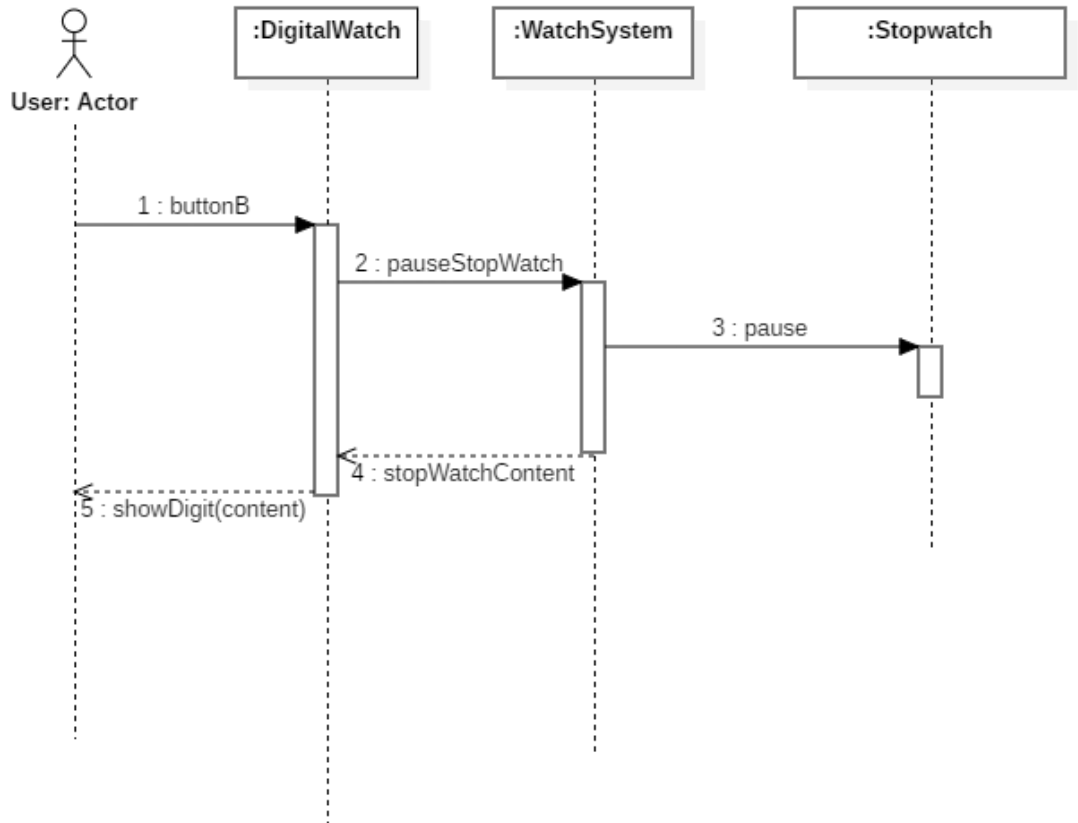
7. Reset Timer



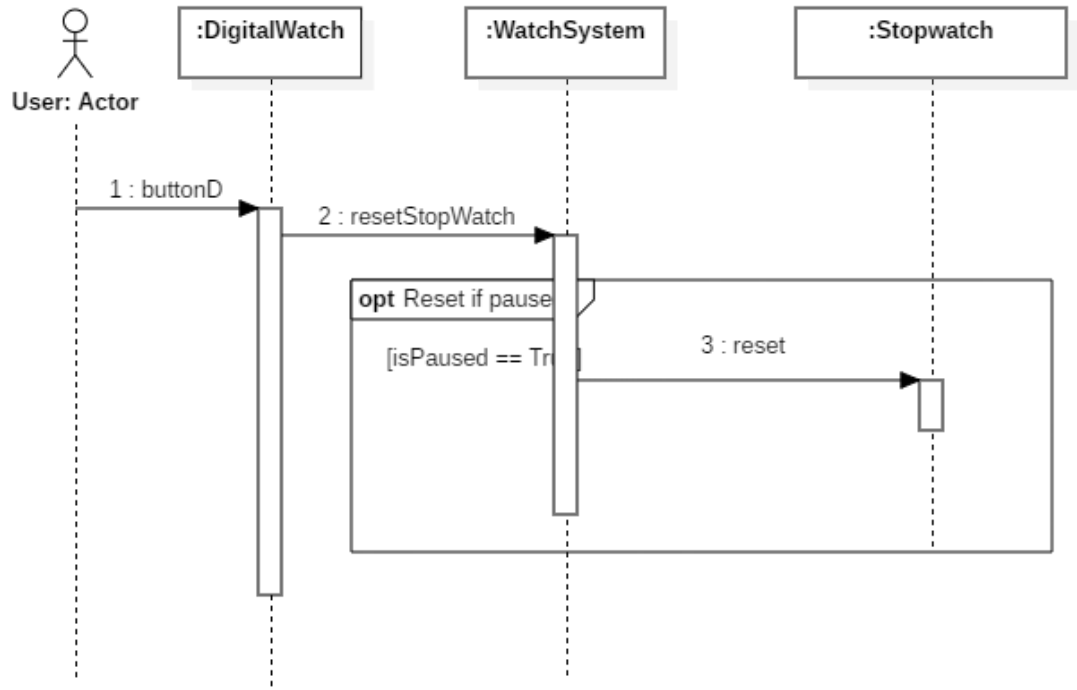
8. Activate Stopwatch



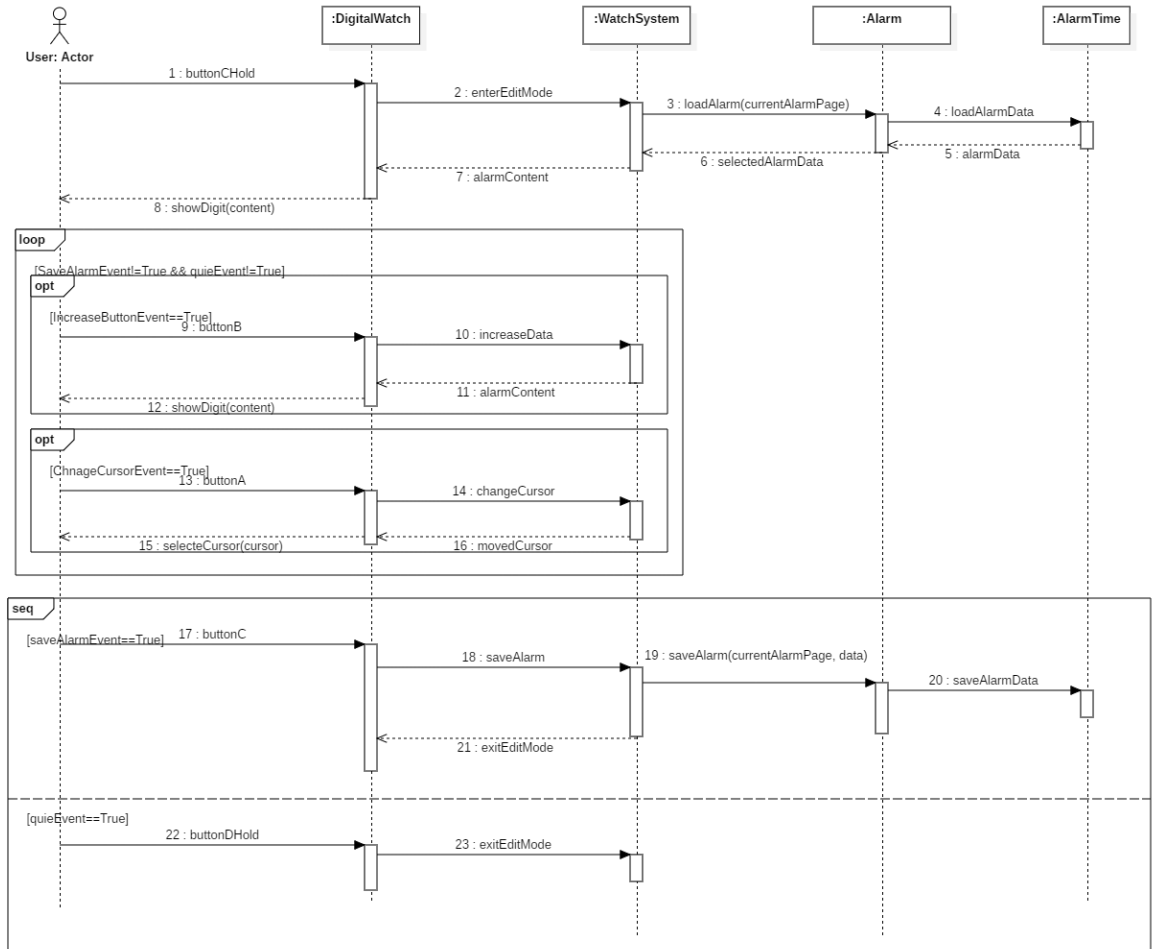
9. Pause Stopwatch



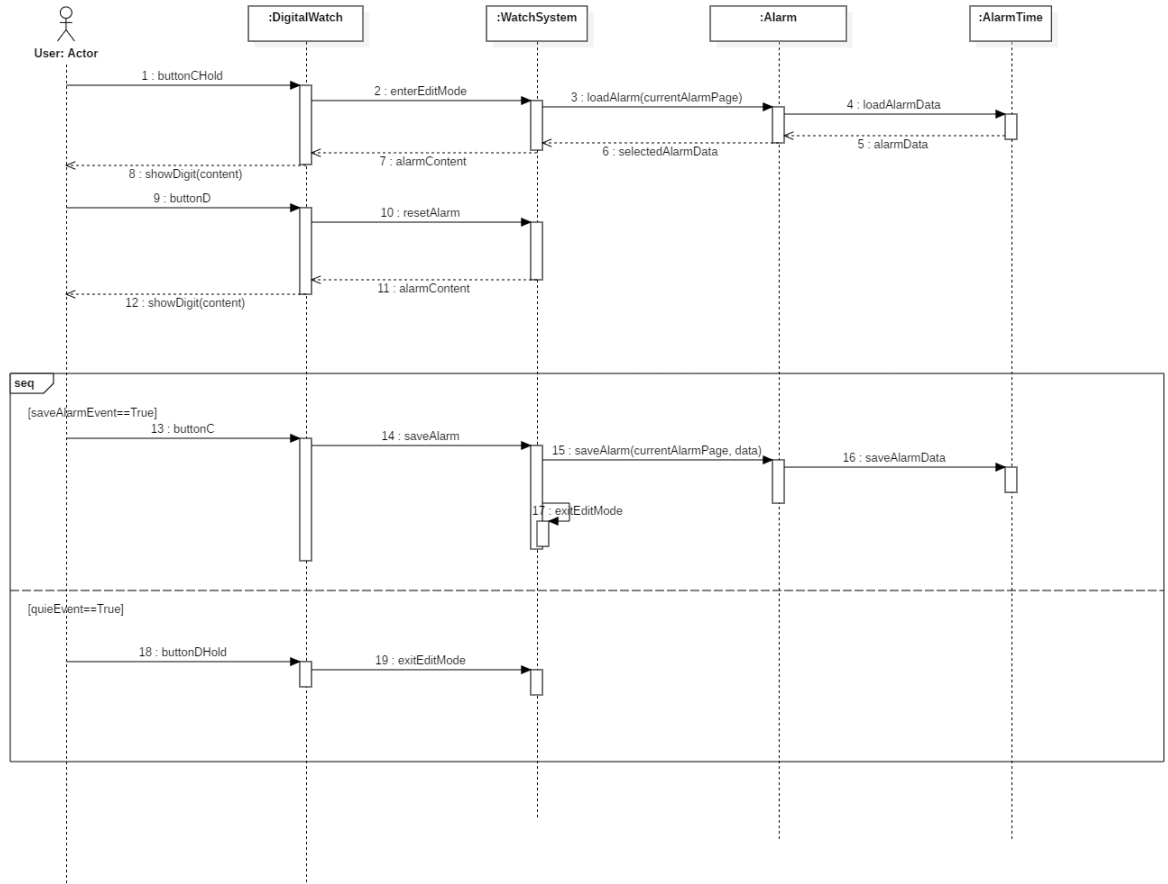
10. Reset Stopwatch



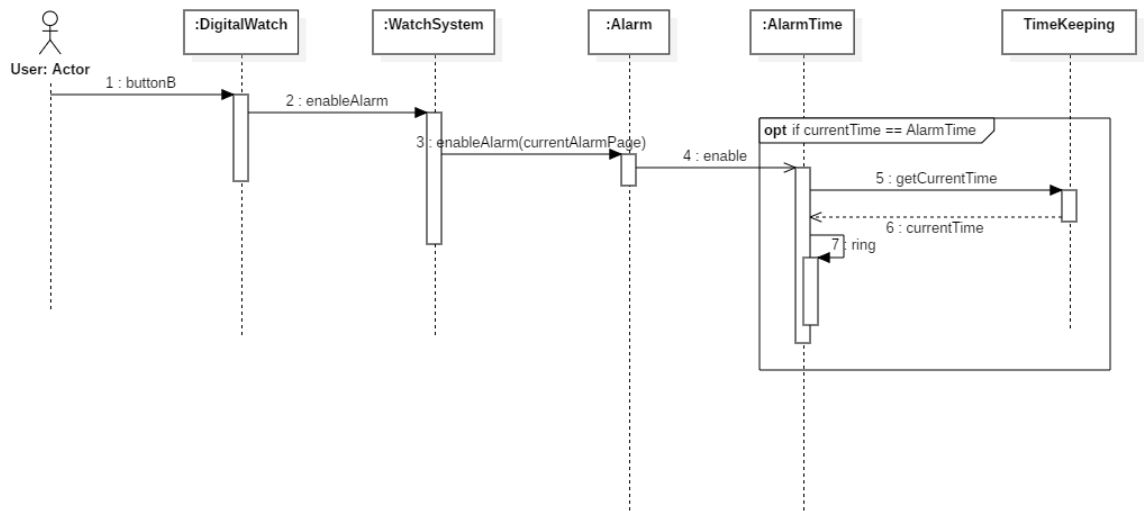
11. Set Alarm



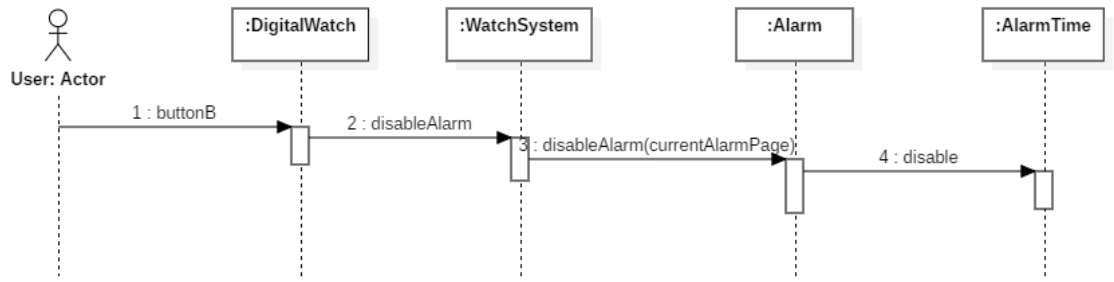
12. Reset Alarm



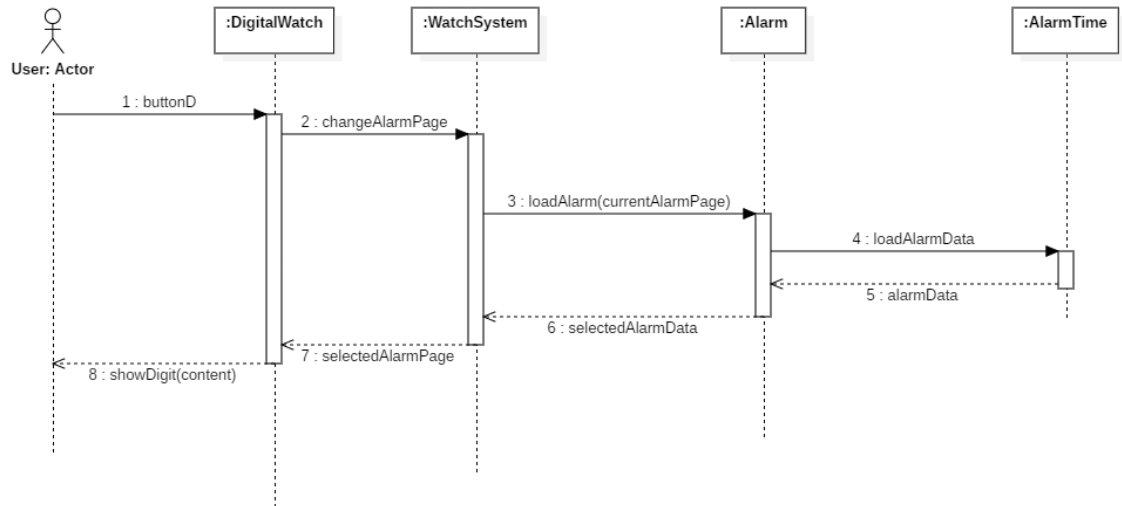
13. Enable Alarm



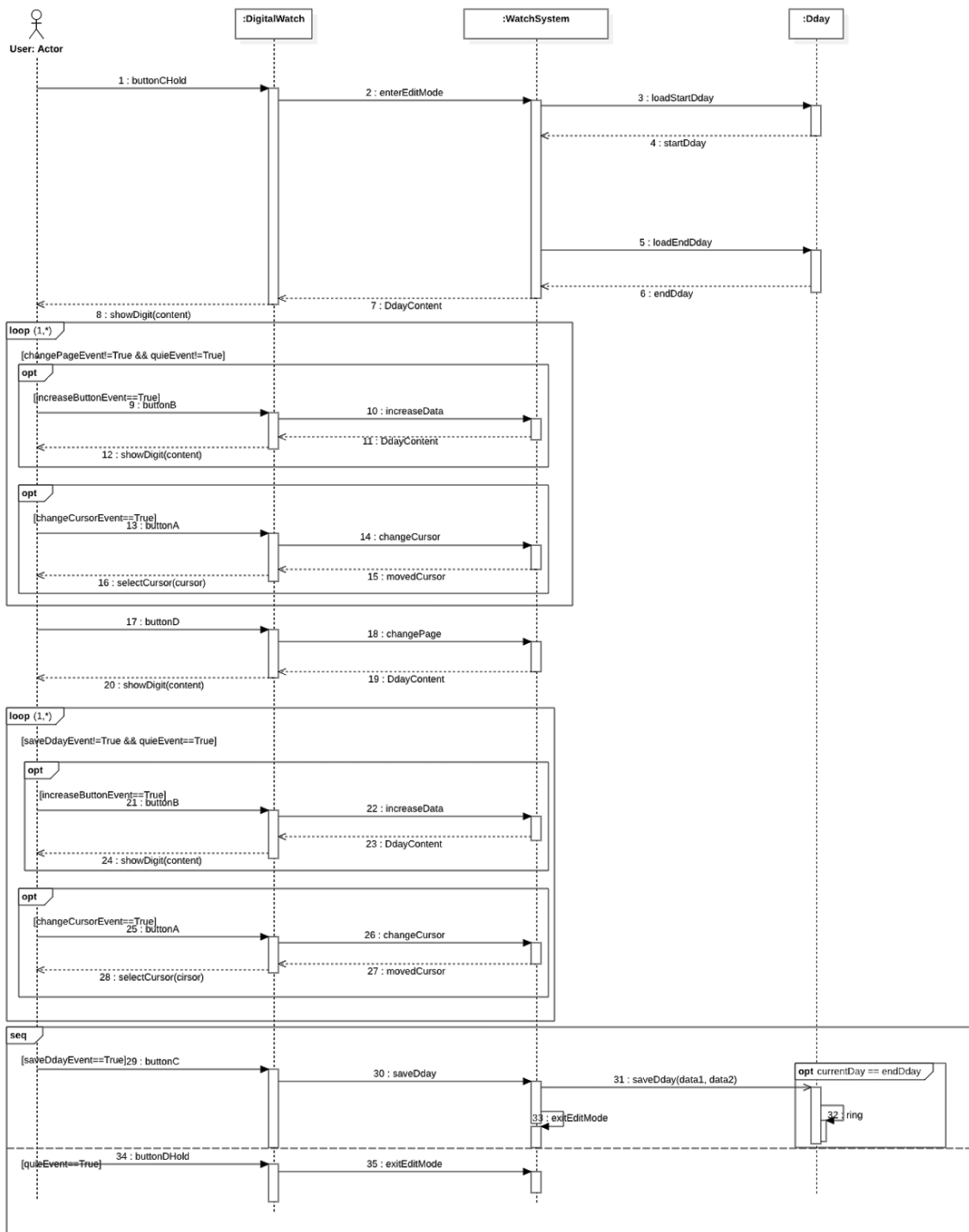
14. Disable Alarm



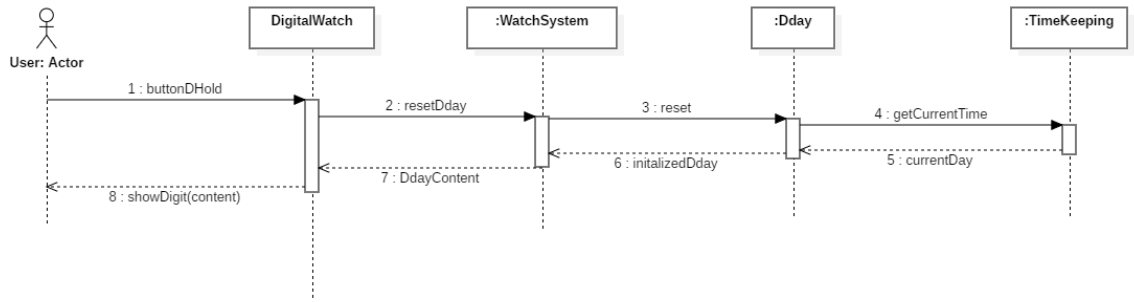
16. Change Alarm page



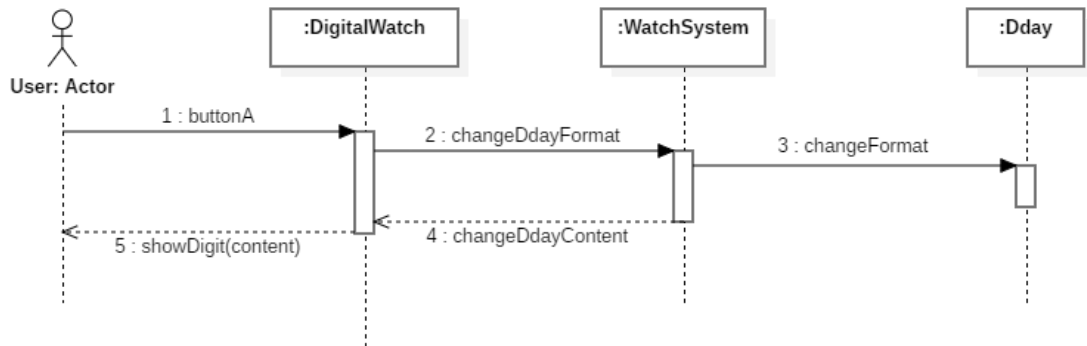
17. Set D-day



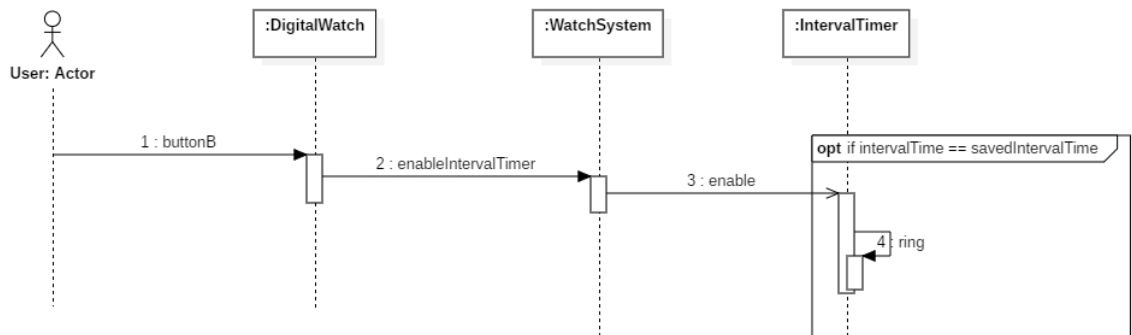
18. Reset D-day



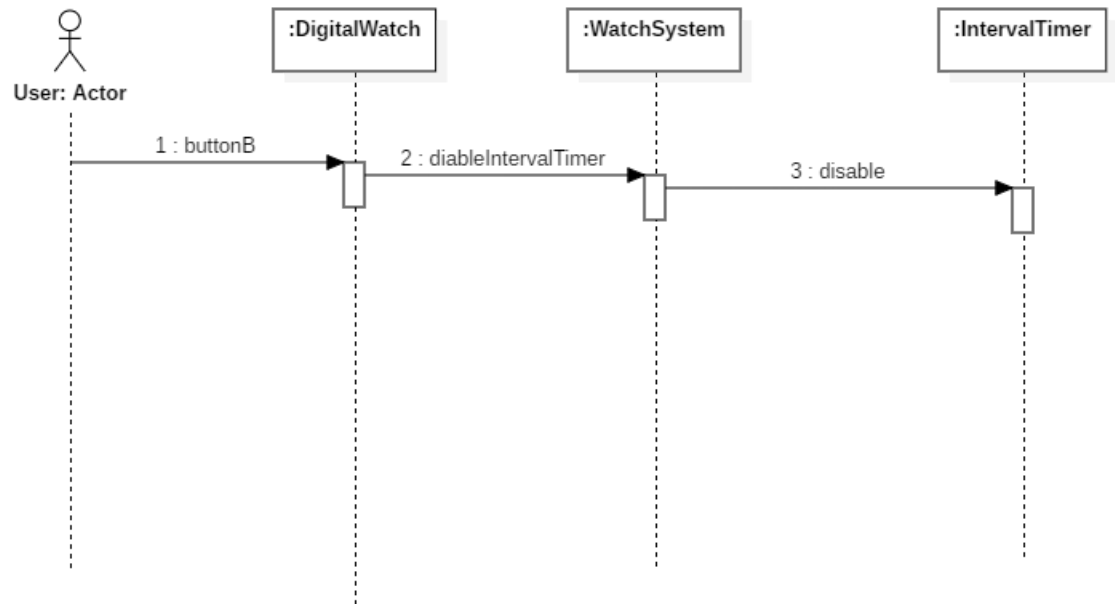
20. Set D-day Format



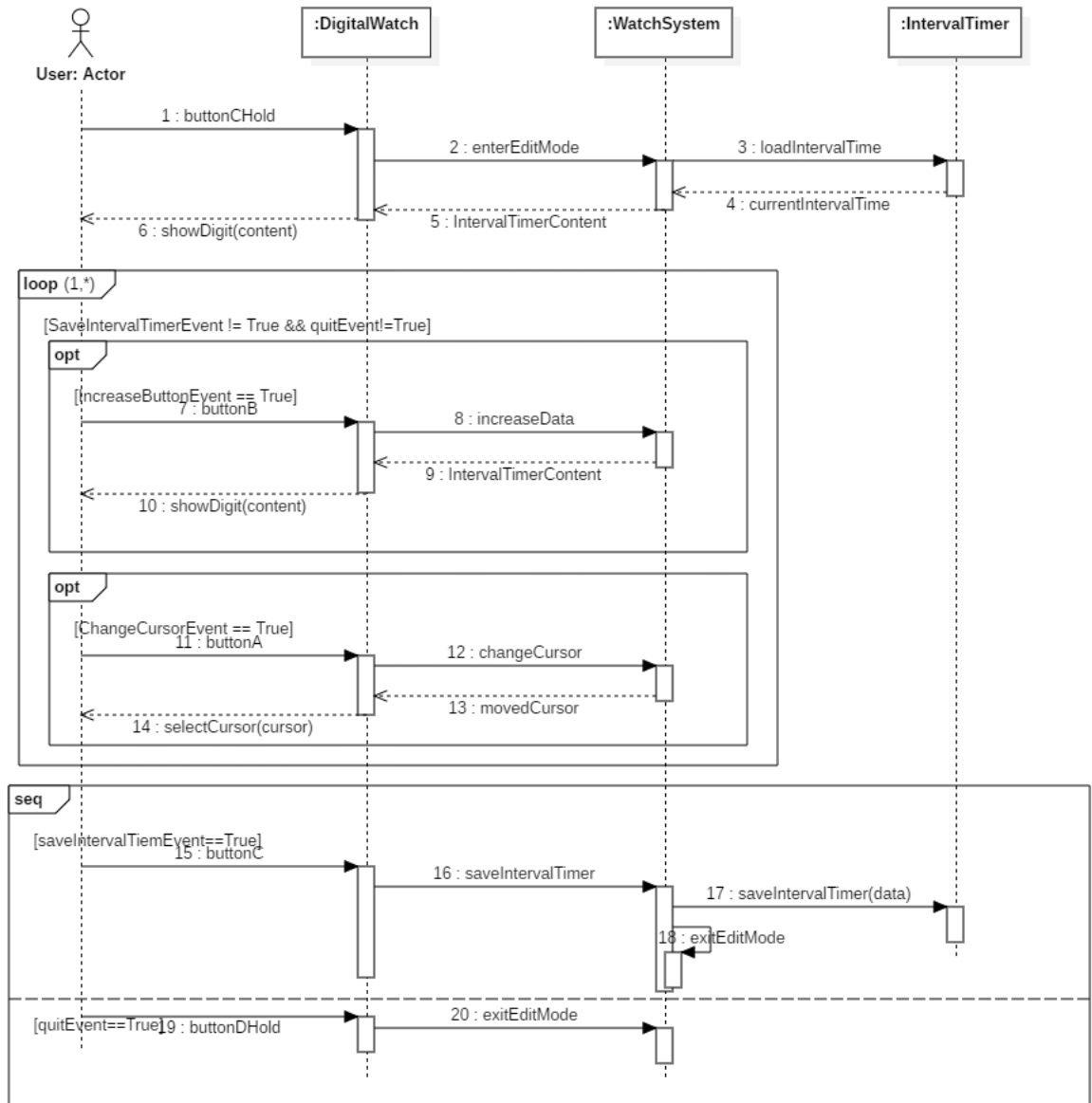
21. Enable Interval Timer



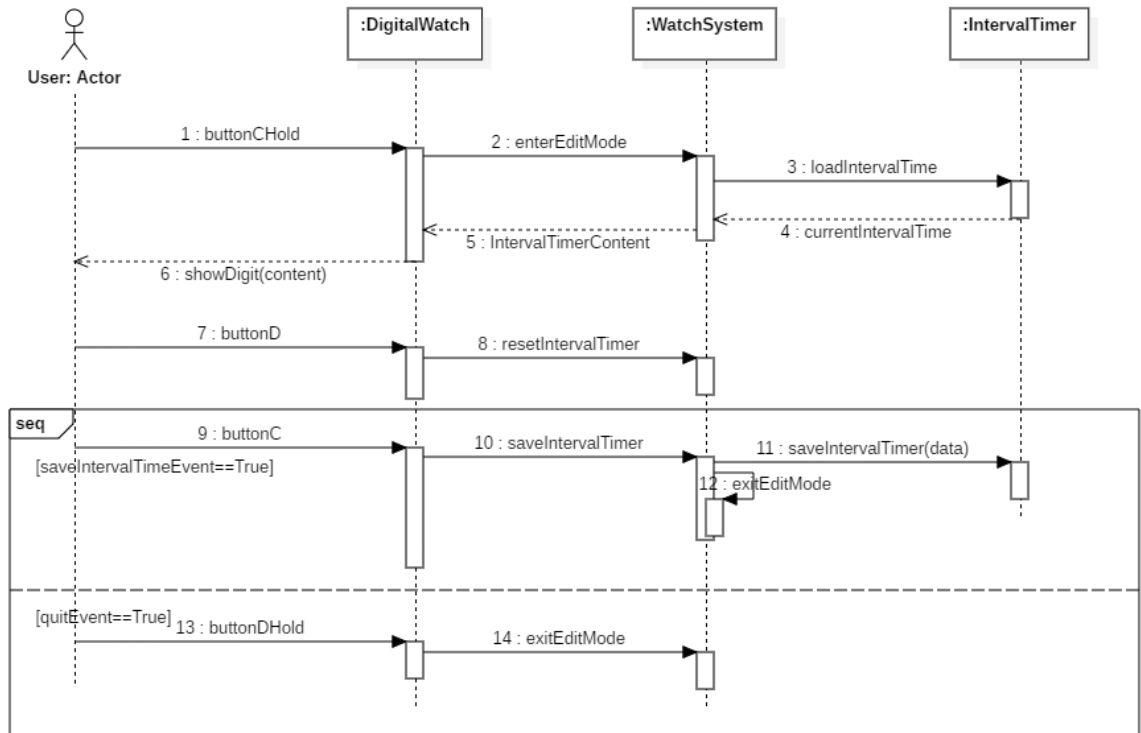
22. Disable Interval Timer



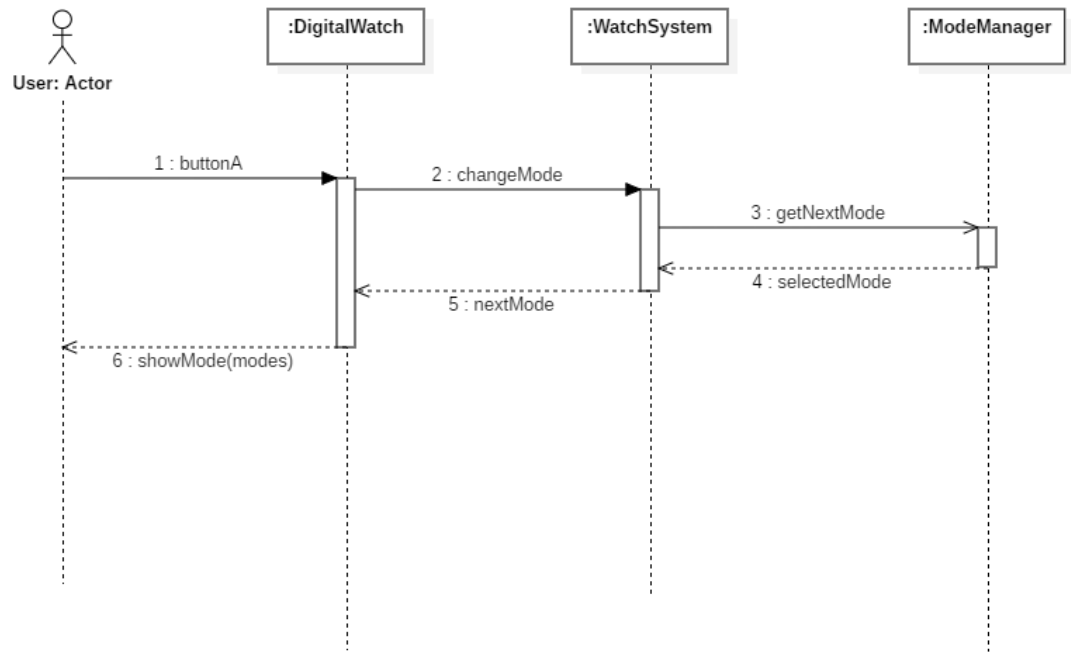
23. Set Interval Timer



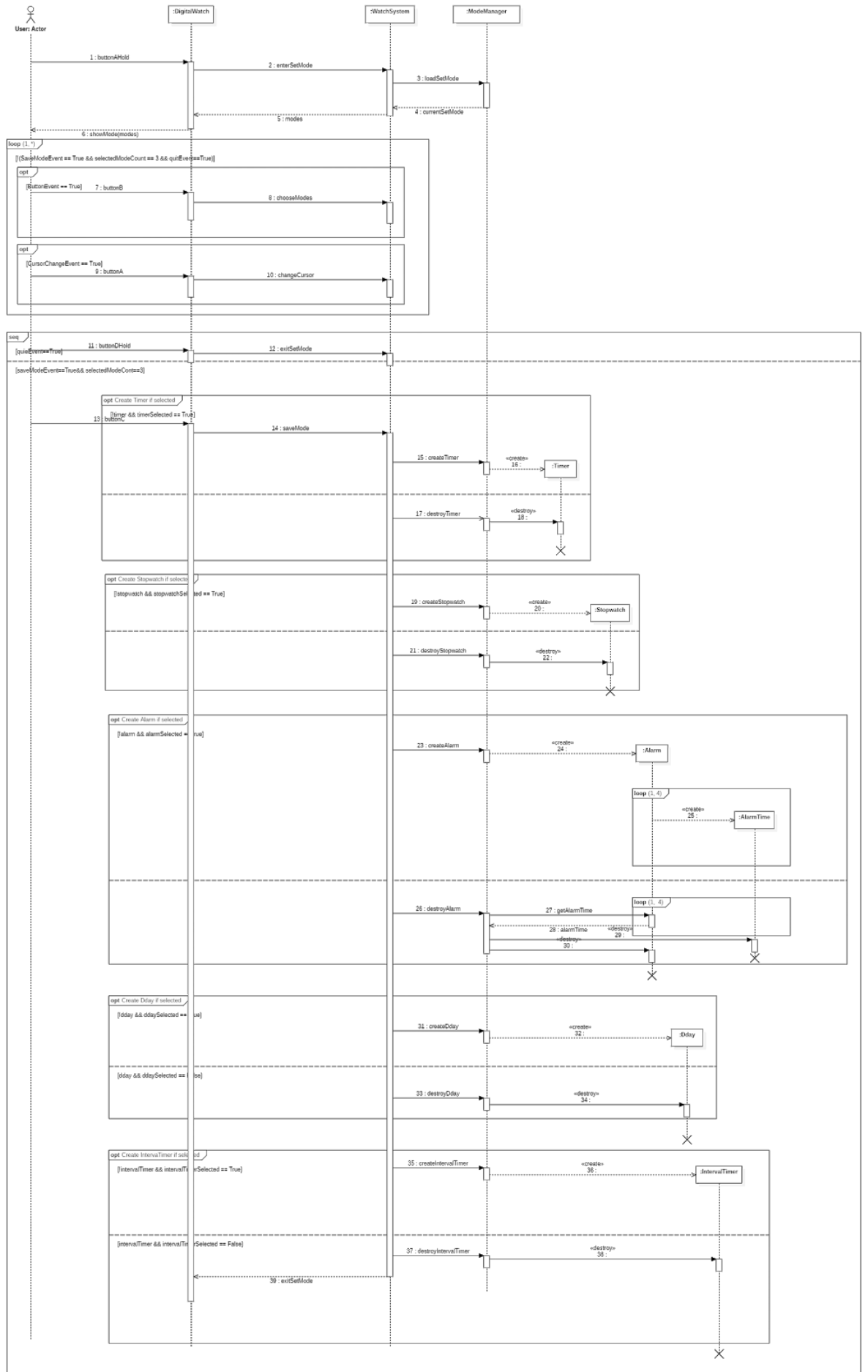
24. Reset Interval Timer



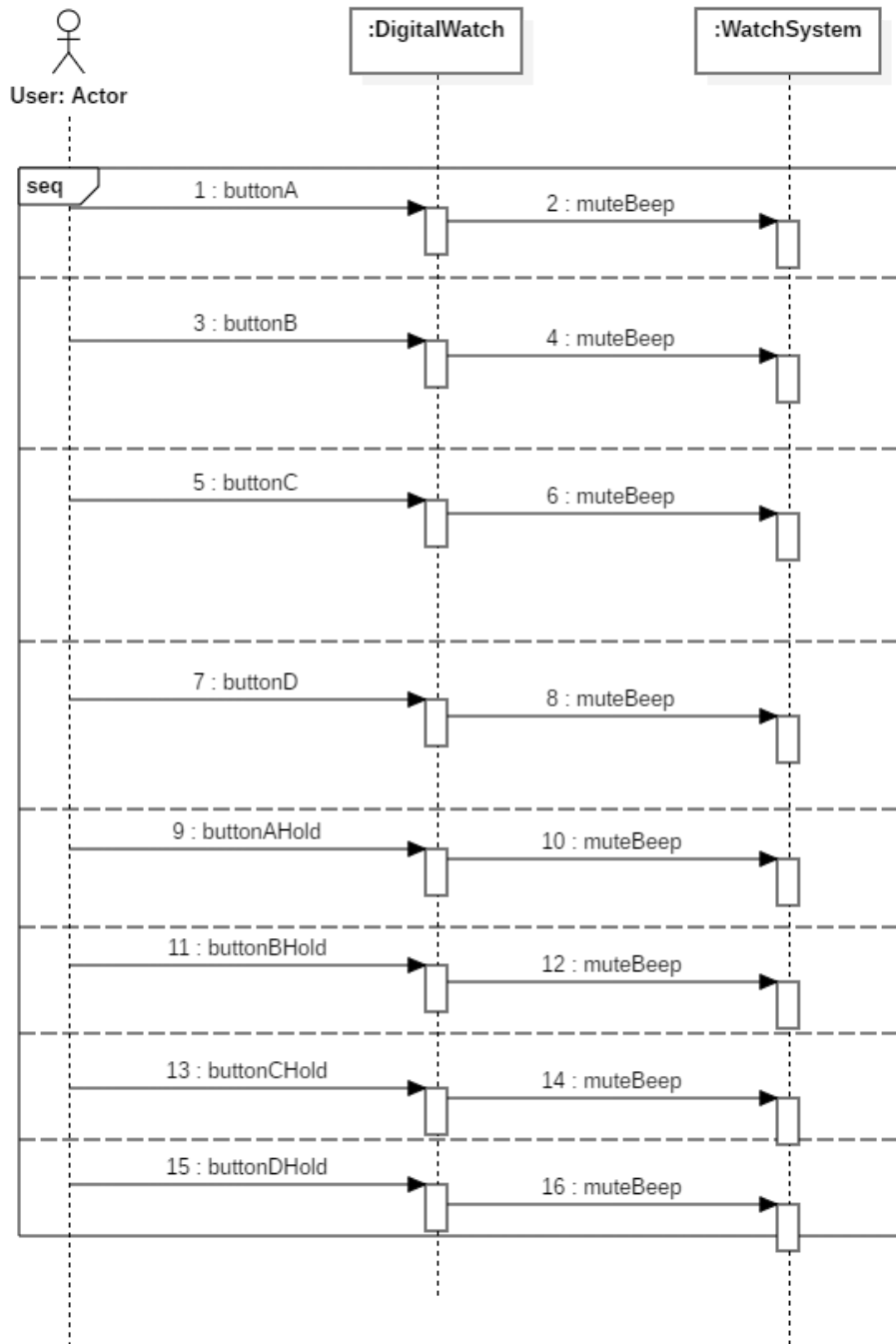
26. Change Mode



27. Set Mode



28. Mute Beep



B. Main GUI Operation Definition

Name	buttonA
Responsibility	사용자가 시계의 A버튼을 누른다.
Type	GUI
Cross Reference	R 1.1, R 1.2, R 2.2, R 4.1, R 5.1, R 5.4, R 6.3, R 7.2, R 7.3
Notes	현재 모드가 Time Keeping일 경우 시간 표시 형식을 변화시켜준다. 현재 모드가 D-day 모드일 경우 D-day 표시 형식을 변화시켜준다. 현재 모드가 WatchTimer, StopWatch, Alarm, IntervalTimer 모드 일 경우 동작하지 않는다. 현재 모드가 editMode일 경우 커서를 움직인다. 현재 모드가 setMode일 경우 커서를 움직인다. 알림이 울리고 있을 경우 mute시켜준다.
Pre-Conditions	N/A.
Post-Conditions	형식을 바꿔 각 모드에 맞는 표현을 보여주고, Edit 모드의 경우 커서를 이동해준다.

Name	buttonB
Responsibility	사용자가 시계의 B버튼을 누른다.
Type	GUI
Cross Reference	R 1.1, R 2.1, R 2.2, R 2.4, R 3.1, R 3.2, R 4.1, R 4.3, R 4.4, R 5.1, R 6.2, R 6.2, R 6.3, R 7.2, R 7.3
Notes	현재 모드가 editMode일 경우 현재 커서의 값을 증가시켜준다. 현재모드가 WatchTimer, Stopwatch일 경우 start/pause상태를 변화시켜준다. 현재모드가 Alarm, IntervalTimer 모드일 경우 enable/disable상태를 변화시켜준다. 현재모드가 setmode일 경우 모드 선택을 할 수 있다. 알림이 울리고 있을 경우 mute시켜준다.
Pre-Conditions	N/A
Post-Conditions	display모드의 경우 활성화 비활성화를 시켜줘야되고, Edit Mode 의 경우 증가된 값을 보여준다. SetMode의 경우 모드를 선택해준다.

Name	buttonC
Responsibility	사용자가 시계의 C버튼을 누른다.
Type	GUI
Cross Reference	R.1.1, R 2.2, R 4.1, R 4.2, R 5.1, R 6.3, R 6.4, R 7.1, R 7.2, R 7.3
Notes	현재 모드가 display mode인 경우 생성되어 있는 다음 모드로 넘어간다. 현재 모드가 editMode인 경우 edit를 멈추고 각 모드에 수정된 값들을 저장해준다. 현재 모드가 setMode인 경우 선택된 모드들을 저장해준다. 알림이 울리고 있을 경우 mute시켜준다.
Pre-Conditions	N/A
Post-Conditions	editMode의 경우 수정된 값들을 저장 후 displayMode로 돌아온다. setMode의 경우 선택된 모드들을 생성, 저장 후 TimeKeeping mode로 돌아온다.

Name	buttonD
Responsibility	사용자가 시계의 D버튼을 누른다.
Type	GUI
Cross Reference	R 2.5, R 3.3, R 4.2, R 4.6, R 5.1, R 6.4, R 7.3
Notes	현재 모드가 AlarmEditMode인 경우 현재 alarmPage의 알람을 reset시켜준다. 현재 모드가 D-day Edit Mode일 경우 현재 D-day Page를 변경시켜준다. 현재 모드가 IntervalTimer Edit모드인 경우 IntervalTime값을 reset시켜준다. 현재 모드가 WatchTimer, StopWatch인 경우 값을 reset시켜준다. 알림이 울리고 있을 경우 mute시켜준다.
Pre-Conditions	현재 모드가 WatchTimer, Stopwatch 모드일 경우 각 모드들은 pause 상태이어야 한다.
Post-Conditions	D-day Edit Mode의 경우 next Page의 저장된 값들을 보여줘야한다.

Name	buttonAHold
Responsibility	사용자가 시계의 A버튼을 1초 이상 누른다.
Type	GUI
Cross Reference	R 7.2, R 7.3
Notes	현재 모드가 displayMode일 경우 setMode로 이동한다. 알림이 울릴 경우 Mute시켜주고 해당 모드의 buttonAHold를 수행한다.
Pre-Conditions	N/A
Post-Conditions	N/A

Name	buttonBHold
Responsibility	사용자가 시계의 B버튼을 1초 이상 누른다.
Type	GUI
Cross Reference	R 7.3
Notes	알림이 울릴 경우 Mute시켜준다.
Pre-Conditions	N/A
Post-Conditions	N/A

Name	buttonCHold
Responsibility	사용자가 시계의 C버튼을 1초 이상 누른다.
Type	GUI
Cross Reference	R 1,1, R 2.2, R 4.1, R 4.2, R 5.1, R 6.3, R 6.4, R 7.3
Notes	현재 모드가 StopWatch를 제외한 displayMode일 경우 각 모드가 editMode로 변경된다. 알림이 울릴 경우 Mute시켜주고 해당 모드의 buttonCHold를 수행한다.
Pre-Conditions	N/A
Post-Conditions	N/A

Name	buttonDHold
Responsibility	사용자가 시계의 D버튼을 1초 이상 누른다.
Type	GUI
Cross Reference	R 1,1, R 2.2, R 4.1, R 4.2, R 5.1, R 5.2, R 6.3, R 6.4, R 7.2, R 7.3
Notes	각 모드의 edit Mode에서 수정된 값을 저장하지 않고 displayMode로 빠져나온다.

	현재 모드가 setMode일 경우 수정된 모드들을 저장하지 않고 setMode를 진입한 모드로 빠져나온다. 알림이 울릴 경우 Mute시켜주고 해당 모드의 buttonDHold를 수행한다.
Pre-Conditions	N/A
Post-Conditions	N/A

2. Activity 2055. Write Test Code

1) WatchSystem Test

```
@Test
void exitEditMode(){
    assertFalse(ws.getEdited());
    assertEquals(ws.getCurrentDayPage(), actual: 0);
}

@Test
void muteBeep() {
    ws.muteBeep();
    assertFalse(DigitalWatch.getInstance().getBell().isPlaying());
}

@Test
void enterSetMode() {
    ws.setCurrentMode(ws.getModeManager().getTimekeeping());
    ws.enterSetMode();
    assertTrue(ws.getIsSetMode());
    ws.exitSetMode();
    ws.setCurrentMode(ws.getModeManager().getStopwatch());
    ws.enterSetMode();
    assertTrue(ws.getIsSetMode());
}

@Test
void activateTimer() {
    ws.setCurrentMode(wt);

    LocalTime tmpTime = LocalTime.of( hour: 0, minute: 0, second: 0);
    LocalDateTime initDateTime = LocalDateTime.of(LocalDate.of( year: 2019, month: 5, dayOfMonth: 26), tmpTime);

    wt.setRemainedTimer(initDateTime);
    ws.activateTimer();
    assertFalse(wt.getActivated());
}
```

```
@Test
void enterEditMode() {
    ws.setCurrentMode(wt);
    wt.setActivated(true);
    assertFalse(ws.getEdited());
    LocalDateTime tmpDateTime = LocalDateTime.of(LocalDate.now(), LocalTime.of( hour: 0, minute: 0, second: 0));
    wt.setRemainedTimer(tmpDateTime);
    ws.enterEditMode();
    assertEquals(ws.getTempTime(), tmpDateTime);
    assertEquals(ws.getCurrentCursor(), actual: 4);
}

@Test
void increaseData() {
    LocalTime tmpTime = LocalTime.of( hour: 0, minute: 0, second: 0);
    LocalDateTime initDateTime = LocalDateTime.of(tk.getCurrentTime().toLocalDate(), tmpTime);
    ws.setCurrentCursor(0);
    ws.setCurrentDayPage(0);
    ws.setTempTime(initDateTime);
    ws.increaseData();
    assertEquals(initDateTime.plusYears(100), ws.getTempTime());
    ws.setCurrentDayPage(1);
    ws.setTempTime2(initDateTime);
    ws.increaseData();
    assertEquals(initDateTime.plusYears(100), ws.getTempTime2());
}

@Test
void changeCursor() {
    ws.setIsSetMode(false);
    ws.setCurrentMode(tk);
    ws.setCurrentCursor(0);
    ws.changeCursor();
    assertEquals(ws.getCurrentCursor(), actual: 1);
}
```

```
@Test
void resetStopwatch() {
    ws.setCurrentMode(sw);
    sw.setActivated(false);
    ws.resetStopwatch();
    assertEquals(sw.getCurrentStopwatch(), LocalTime.of( hour: 0, minute: 0, second: 0));
    assertEquals(sw.getActivated(), actual: false);
    assertEquals(sw.getCountDay(), actual: 0);
}

@Test
void chooseModes() {
    ws.setCurrentCursor(0);
    Boolean[] tmp = {true,true,false,false,true};
    ws.setSetMode(tmp);
    ws.chooseModes();
    assertFalse(ws.getSetMode()[0]);
}

@Test
void saveMode() {
    Boolean[] tmp = {true,true,false,false,true};
    ws.setSetMode(tmp);
    ws.saveMode();
    assertNotNull(ws.getModeManager().getWatchTimer());
    assertNotNull(ws.getModeManager().getStopwatch());
    assertNotNull(ws.getModeManager().getIntervalTimer());
}

@Test
void exitSetMode(){
    assertFalse(ws.getIsSetMode());
    assertEquals(ws.getCurrentModeCursor(), actual: 0);
    ws.setCurrentMode(ws.getModeManager().getWatchTimer());
    ws.enterSetMode();
    ws.exitSetMode();
    assertEquals(ws.getCurrentMode(), ws.getModeManager().getTimekeeping());
}
```

```
@Test
void saveTime() {
    LocalTime tmpTime = LocalTime.of( hour: 0, minute: 0, second: 0);
    LocalDateTime initDateTime = LocalDateTime.of(tk.getCurrentTime().toLocalDate(), tmpTime);
    ws.setTempTime(initDateTime);
    ws.setCurrentMode(tk);
    ws.saveTime();
    assertEquals(tk.getCurrentTime(), initDateTime);
}

@Test
void changeHourFormat() {
    ws.setCurrentMode(tk);
    ws.changeHourFormat();
    assertEquals(tk.getDisplayFormat(), actual: false);
}

@Test
void pauseTimer() {
    ws.setCurrentMode(wt);
    ws.pauseTimer();
    assertEquals(ws.getActivated(), actual: false);
}

@Test
void resetTimer() {
    LocalTime tmpTime = LocalTime.of( hour: 0, minute: 0, second: 0);
    LocalDateTime initDateTime = LocalDateTime.of(LocalDate.now(), tmpTime);
    ws.setCurrentMode(wt);
    wt.setActivated(false);
    ws.resetTimer();

    assertEquals(wt.getSavedTimer(), initDateTime);
    assertEquals(wt.getRemainedTimer(), initDateTime);
}
```



```

@Test
void saveTimer() {
    LocalTime tmpTime = LocalTime.of( hour: 0, minute: 0, second: 0);
    LocalDateTime initDateTime = LocalDateTime.of(LocalDate.now(), tmpTime);
    ws.setCurrentMode(wt);
    ws.setTempTime(initDateTime);
    ws.saveTimer();
    assertEquals(wt.getSavedTimer(), initDateTime);
    assertEquals(wt.getRemainedTimer(), initDateTime);
    assertEquals(wt.getActivated(), actual: false);
}

@Test
void enableIntervalTimer() {
    ws.setCurrentMode(it);

    LocalTime tmpTime = LocalTime.of( hour: 0, minute: 0, second: 0);
    LocalTime tmpTime2 = LocalTime.of( hour: 0, minute: 10, second: 0);
    LocalDateTime initDateTime = LocalDateTime.of(LocalDate.now(), tmpTime);
    LocalDateTime initDateTime2 = LocalDateTime.of(LocalDate.now(), tmpTime2);
    it.setSavedIntervalTimer(initDateTime2);
    it.setRemainedIntervalTimer(initDateTime);

    ws.enableIntervalTimer();

    assertTrue(it.getIsEnabled());
    assertEquals(it.getRemainedIntervalTimer(), it.getSavedIntervalTimer());
}

```

```

@Test
void changeAlarmPage() {
    ws.setCurrentMode(alarm);
    ws.setCurrentAlarmPage(3);
    ws.changeAlarmPage();
    assertEquals( expected: 0, ws.getCurrentAlarmPage());
}

@Test
void changePage() {
    ws.setCurrentDdayPage(0);
    ws.changePage();
    assertEquals(ws.getCurrentDdayPage(), actual: 1);
    ws.setCurrentDdayPage(1);
    ws.changePage();
    assertEquals(ws.getCurrentDdayPage(), actual: 0);
}

@Test
void saveDday() {
    ws.setCurrentMode(d);
    ws.setEdited(false);
    LocalDateTime tmpDateTime = d.getStartDday();
    ws.saveDday();
    assertEquals(d.getStartDday(), tmpDateTime);
    LocalTime tmpTime = LocalTime.of( hour: 0, minute: 0, second: 0);
    LocalDateTime initDateTime = LocalDateTime.of(LocalDate.of( year: 2019, month: 5, dayOfMonth: 26), tmpTime);
    LocalDateTime initDateTime2 = LocalDateTime.of(LocalDate.of( year: 2019, month: 5, dayOfMonth: 27), tmpTime);
    ws.setTempTime(initDateTime);
    ws.setTempTime2(initDateTime2);
    ws.setEdited(true);
    ws.saveDday();
    assertEquals(d.getStartDday(), initDateTime);
    assertEquals(d.getEndDday(), initDateTime2);
    assertEquals(d.getExistStartDday(), actual: true);
}

```

```

@Test
void saveAlarm() {
    ws.setCurrentMode(alarm);
    ws.setCurrentAlarmPage(0);
    LocalTime tmpTime = LocalTime.of( hour: 0, minute: 0, second: 0);
    LocalDateTime initDateTime = LocalDateTime.of(LocalDate.of( year: 2019, month: 5, dayOfMonth: 26), tmpTime);
    ws.setTempTime(initDateTime);
    ws.saveAlarm();
    assertEquals(alarm.getAlarmTime( currentAlarmpage: 0).getCurrentAlarm(), initDateTime);
}

@Test
void resetAlarm() {
    LocalDateTime tmpTime = LocalDateTime.of(LocalDate.now(), LocalTime.of( hour: 0, minute: 0, second: 0));
    ws.setCurrentMode(alarm);
    ws.setCurrentAlarmPage(0);
    ws.setTempTime(tmpTime);
    alarm.getAlarmTime( currentAlarmpage: 0).setCurrentAlarm(tmpTime);
    ws.resetAlarm();
    assertEquals(ws.getModeManager().getAlarm().getAlarmTime( currentAlarmpage: 0).getCurrentAlarm(), tmpTime);
}

@Test
void enableAlarm() {
    ws.setCurrentMode(alarm);
    ws.setCurrentAlarmPage(0);
    ws.enableAlarm();
    assertEquals(alarm.getAlarmTime( currentAlarmpage: 0).getEnabled(), actual: true);
}

@Test
void disableAlarm() {
    ws.setCurrentMode(alarm);
    ws.setCurrentAlarmPage(0);
    ws.disableAlarm();
    assertEquals(alarm.getAlarmTime( currentAlarmpage: 0).getEnabled(), actual: false);
}

```

```

@Test
void resetDday() {
    ws.setCurrentMode(d);

    LocalTime tmpTime = LocalTime.of( hour: 0, minute: 0, second: 0);
    LocalDateTime initDateTime = LocalDateTime.of(LocalDate.of( year: 2019, month: 5, dayOfMonth: 26), tmpTime);

    d.setEndDday(initDateTime);
    d.setCurrentDay(initDateTime);

    assertEquals(ws.resetDday(), d.getEndDday());
    assertFalse(d.getExistStartDday());
    assertEquals(d.getStartDday(), initDateTime);
    assertEquals(d.getEndDday(), initDateTime);
}

@Test
void changeDdayFormat() {
    ws.setCurrentMode(d);
    d.setExistStartDday(false);
    d.setCalDday(10);
    assertEquals(ws.changeDdayFormat(), d.getCalDday());
    assertEquals(d.getDisplayType(), actual: true);
}

@Test
void activateStopwatch() {
    ws.setCurrentMode(sw);
    ws.activateStopwatch();
    assertTrue(sw.getActivated());
}

@Test
void pauseStopwatch() {
    ws.setCurrentMode(sw);
    ws.pauseStopwatch();
    assertFalse(sw.getActivated());
}

```

2) TimeKeeping Test

```

@Test
void loadTime() {
    LocalTime initTime = LocalTime.of( hour: 0, minute: 0, second: 0);
    LocalDateTime initDateTime = LocalDateTime.of(LocalDate.now(), initTime);
    tk.setCurrentTime(initDateTime);
    assertEquals(tk.loadTime(), initDateTime);
}

@Test
void saveTime() {
    LocalTime tmpTime = LocalTime.of( hour: 0, minute: 0, second: 0);
    LocalDateTime initDateTime = LocalDateTime.of(LocalDate.now(), tmpTime);
    tk.saveTime(initDateTime);
    assertEquals(tk.getCurrentTime(), initDateTime);
}

@Test
void setHourformat() {
    tk.setHourformat();
    assertFalse(tk.getDisplayFormat());
}

```

3) WatchTimer Test

```
@Test
void activate() {
    LocalTime tmpTime = LocalTime.of( hour: 0, minute: 0, second: 0);
    LocalDateTime initDateTime = LocalDateTime.of(LocalDate.now(), tmpTime);
    wt.setRemainedTimer(initDateTime);
    wt.activate();
    assertFalse(wt.getActivated());
}

@Test
void pause() {
    wt.pause();
    assertEquals(wt.getActivated(), actual: false);
}

@Test
void reset() {
    LocalTime tmpTime = LocalTime.of( hour: 0, minute: 0, second: 0);
    LocalDateTime initDateTime = LocalDateTime.of(LocalDate.now(), tmpTime);
    wt.reset();
    assertEquals(wt.getRemainedTimer(), initDateTime);
    assertEquals(wt.getSavedTimer(), initDateTime);
}

@Test
void loadTimer() {
    LocalTime tmpTime = LocalTime.of( hour: 0, minute: 0, second: 0);
    LocalDateTime initDateTime = LocalDateTime.of(LocalDate.now(), tmpTime);
    wt.setRemainedTimer(initDateTime);
    LocalDateTime getRemainedTime = wt.getRemainedTimer();
    assertEquals(getRemainedTime, initDateTime);
}
```

```
@Test
void saveTimer() {
    LocalTime tmpTime = LocalTime.of( hour: 0, minute: 0, second: 0);
    LocalDateTime initDateTime = LocalDateTime.of(tk.getCurrentTime().toLocalDate(), tmpTime);
    wt.saveTimer(initDateTime);
    assertEquals(wt.getSavedTimer(), initDateTime);
    assertEquals(wt.getRemainedTimer(), initDateTime);
    assertFalse(wt.getActivated());
}

@Test
void ring() {
    wt.ring();
    assertTrue(DigitalWatch.getInstance().getBell().isPlaying());
}
```

4) Stopwatch Test

```
@Test
void activate() {
    sw.activate();
    assertTrue(sw.getActivated());
}

@Test
void pause() {
    sw.pause();
    assertFalse(sw.getActivated());
}

@Test
void reset() {
    sw.setActivated(false);
    sw.reset();
    assertEquals(sw.getActivated(), actual: false);
    assertEquals(sw.getCountDay(), actual: 0);
    assertEquals(sw.getCurrentStopwatch(), LocalTime.of( hour: 0, minute: 0, second: 0));
}

@Test
void loadStopWatch(){
    LocalTime tmpTime = LocalTime.of( hour: 0, minute: 0, second: 0);
    sw.setCurrentStopwatch(tmpTime);
    assertEquals(sw.loadStopWatch(), tmpTime);
}
```

5) Alarm Test

```
@Test
void loadAlarm() {
    LocalTime tmpTime = LocalTime.of( hour: 0, minute: 0, second: 0);
    alarm.getAlarmTime( currentAlarmPage: 0).setCurrentAlarm(LocalDateTime.of(LocalDate.now(), tmpTime));
    assertEquals(alarm.getAlarmTime( currentAlarmPage: 0).getCurrentAlarm(), LocalDateTime.of(LocalDate.now(), tmpTime));
}

@Test
void saveAlarm() {
    LocalDateTime ldt = LocalDateTime.now();
    alarm.saveAlarm( currentAlarmPage: 0, ldt);
    assertEquals(alarm.getAlarmTime( currentAlarmPage: 0).getCurrentAlarm(), ldt);
}

@Test
void enableAlarm() {
    alarm.enableAlarm( currentAlarmPage: 0);
    assertEquals(alarm.getAlarmTime( currentAlarmPage: 0).getEnabled(), actual: true);
}

@Test
void disableAlarm() {
    alarm.disableAlarm( currentAlarmPage: 0);
    assertEquals(alarm.getAlarmTime( currentAlarmPage: 0).getEnabled(), actual: false);
}
```

6) AlarmTime Test

```
@Test
void loadAlarmData() {
    artm.setCurrentAlarm(LocalDate.now());
    assertEquals(artm.getCurrentAlarm(), LocalDate.now());
}

@Test
void saveAlarmData() {
    LocalDateTime tmpDateTime = LocalDate.now();
    artm.saveAlarmData(tmpDateTime);
    assertEquals(artm.getCurrentAlarm(), tmpDateTime);
}

@Test
void enable() {
    artm.enable();
    assertTrue(artm.getEnabled());
}

@Test
void disable() {
    artm.disable();
    assertFalse(artm.getEnabled());
}

@Test
void ring(){
    artm.ring();
    assertFalse(artm.getEnabled());
}
```

7) Dday Test

```
@Test
void loadStartDday() {
    LocalTime initTime = LocalTime.of( hour: 0, minute: 0, second: 0);
    LocalDateTime initDateTime = LocalDateTime.of(tk.getCurrentTime().toLocalDate(), initTime);
    d.setStartDday(initDateTime);
    d.loadStartDday();
    assertEquals(d.getStartDday(), initDateTime);
}

@Test
void loadEndDday() {
    LocalTime initTime = LocalTime.of( hour: 0, minute: 0, second: 0);
    LocalDateTime initDateTime = LocalDateTime.of(tk.getCurrentTime().toLocalDate(), initTime);
    d.setEndDday(initDateTime);
    d.loadEndDday();
    assertEquals(d.getEndDday(), initDateTime);
}

@Test
void reset() {
    LocalDateTime initDateTime = LocalDateTime.of(tk.getCurrentTime().toLocalDate(), tk.getCurrentTime().toLocalTime());
    d.setCurrentDay(tk.getCurrentTime());
    d.reset();
    assertEquals(d.getStartDday(), initDateTime);
    assertEquals(d.getEndDday(), initDateTime);
    assertFalse(d.getExistStartDday());
    assertFalse(d.getExistEndDday());
    assertTrue(d.getDisplayType());
}
}
```

```
@Test
void changeFormat() {
    d.setDisplayType(true);
    d.setExistStartDday(false);
    d.changeFormat();
    assertTrue(d.getDisplayType());

    d.setDisplayType(true);
    d.setExistStartDday(true);
    d.changeFormat();
    assertFalse(d.getDisplayType());
}

@Test
void ring(){
    d.ring();
    assertTrue(DigitalWatch.getInstance().getBell().isPlaying());
}
}
```

8) IntervalTimer Test

```
@Test
void loadIntervalTimer() {
    LocalTime initTime = LocalTime.of( hour: 0, minute: 0, second: 0);
    LocalDate initDate = LocalDate.of( year: 0, month: 1, dayOfMonth: 1);
    LocalDateTime initDateTime = LocalDateTime.of( initDate, initTime);
    it.setRemainedIntervalTimer( initDateTime);
    assertEquals( it.loadIntervalTimer(), initDateTime);
}

@Test
void saveIntervalTimer() {
    LocalTime initTime = LocalTime.of( hour: 0, minute: 0, second: 0);
    LocalDate initDate = LocalDate.of( year: 0, month: 1, dayOfMonth: 1);
    LocalDateTime initDateTime = LocalDateTime.of( initDate, initTime);

    it.setEnabled( false);
    it.saveIntervalTimer( initDateTime);
    assertEquals( it.getSavedIntervalTimer(), initDateTime);
    assertEquals( it.getRemainedIntervalTimer(), initDateTime);
    assertEquals( it.getIteration(), actual: 0);
}

@Test
void ring() {
    it.ring();
    assertTrue( DigitalWatch.getInstance().getBell().isPlaying());
}
```

```
@Test
void enable() {
    LocalTime tmpTime = LocalTime.of( hour: 0, minute: 0, second: 0);
    LocalTime tmpTime2 = LocalTime.of( hour: 0, minute: 10, second: 0);
    LocalDateTime initDateTime = LocalDateTime.of( LocalDate.now(), tmpTime);
    LocalDateTime initDateTime2 = LocalDateTime.of( LocalDate.now(), tmpTime2);
    it.setSavedIntervalTimer( initDateTime2);
    it.enable();
    assertTrue( it.getIsEnabled());

    it.setRemainedIntervalTimer( initDateTime);
    it.enable();
    assertEquals( it.getRemainedIntervalTimer(), it.getSavedIntervalTimer());
}

@Test
void disable() {
    it.setEnabled( true);
    it.disable();
    assertEquals( it.getIsEnabled(), actual: false);
}

@Test
void loadIntervalTimer() {
    LocalTime initTime = LocalTime.of( hour: 0, minute: 0, second: 0);
    LocalDate initDate = LocalDate.of( year: 0, month: 1, dayOfMonth: 1);
    LocalDateTime initDateTime = LocalDateTime.of( initDate, initTime);
    it.setRemainedIntervalTimer( initDateTime);
    assertEquals( it.loadIntervalTimer(), initDateTime);
}
```

9) ModeManager Test

```
@Test
void createStopwatch() {
    Stopwatch stopwatch = mM.createStopwatch();
    assertNotNull(stopwatch);
}

@Test
void destroyStopwatch() {
    mM.destroyStopwatch();
    assertNull(mM.getStopwatch());
}

@Test
void createAlarm() {
    Alarm alarm = mM.createAlarm();
    assertNotNull(alarm);
}

@Test
void destroyAlarm() {
    mM.destroyAlarm();
    assertNull(mM.getAlarm());
}

@Test
void createDday() {
    Dday dday = mM.createDday();
    assertNotNull(dday);
}

@Test
void destroyDday() {
    mM.destroyDday();
    assertNull(mM.getDday());
}
```

```
@Test
void getCurrentMode() {
    mM.setCurrentMode(0);
    assertEquals(mM.getCurrentMode(), mM.getModes().get(0));
}

@Test
void getNextMode() {
    Boolean[] tmpSetMode = {true, true, true, false, false};
    ws.setSetMode(tmpSetMode);
    ws.saveMode();
    mM.setCurrentMode(0);
    Object nextMode = mM.getNextMode();
    assertEquals(nextMode, mM.getCurrentMode());
}

@Test
void loadSetMode() {
    Boolean[] tmpSetMode = {true, true, true, false, false};
    mM.setSetMode(tmpSetMode);
    assertEquals(mM.loadSetMode(), tmpSetMode);
}

@Test
void createTimer() {
    WatchTimer ctWatchTimer = mM.createTimer();
    assertNotNull(ctWatchTimer);
}

@Test
void destroyTimer() {
    mM.destroyTimer();
    assertNull(mM.getWatchTimer());
}
```

```
@Test
void createIntervalTimer() {
    IntervalTimer it = mM.createIntervalTimer();
    assertNotNull(it);
}

@Test
void destroyIntervalTimer() {
    mM.destroyIntervalTimer();
    assertNull(mM.getIntervaltimer());
}
```

3. Activity 2061. Unit Testing

```

▼ <default package> 1 s 697 ms
  ▼ AlarmTest 19 ms
    ✓ enableAlarm() 16 ms
    ✓ disableAlarm()
    ✓ saveAlarm() 2 ms
    ✓ loadAlarm() 1 ms
  ▼ WatchTimerTest 1 s 595 ms
    ✓ activate() 7 ms
    ✓ ring() 1 s 581 ms
    ✓ pause() 2 ms
    ✓ reset() 1 ms
    ✓ saveTimer() 2 ms
    ✓ loadTimer() 2 ms
  ▼ ModeManagerTest 16 ms
    ✓ destroyDday() 3 ms
    ✓ destroyAlarm()
    ✓ getNextMode() 1 ms
    ✓ createAlarm() 1 ms
    ✓ createTimer() 2 ms
    ✓ getCurerntMode() 1 ms
    ✓ createStopwatch()
    ✓ createIntervalTimer() 2 ms
    ✓ destoryTimer()
    ✓ createDday() 3 ms
    ✓ loadSetMode() 1 ms
    ✓ destroyStopwatch() 1 ms
    ✓ destroyIntervalTimer() 1 ms
  ▼ AlarmTimeTest 4 ms
    ✓ enable()
    ✓ ring() 1 ms
    ✓ loadAlarmData() 1 ms
    ✓ saveAlarmData() 1 ms
    ✓ disable() 1 ms
  ▼ StopwatchTest 2 ms
    ✓ activate() 1 ms
    ✓ loadStopWatch() 1 ms
    ✓ pause()
    ✓ reset()
  ▼ TimeKeepingTest
    ✓ saveTime()
    ✓ setHourFormat()
    ✓ loadTime()
  ▼ DdayTest 1 ms
    ✓ loadEndDday()
    ✓ loadStartDday()
    ✓ changeFormat()
    ✓ ring() 1 ms
    ✓ reset()
  ▼ WatchSystemTest 53 ms
    ✓ changeMode()
    ✓ changePage() 1 ms

```

```

  ✓ reset()
  ▼ WatchSystemTest 53 ms
    ✓ changeMode()
    ✓ changePage() 1 ms
    ✓ saveDday()
    ✓ saveMode()
    ✓ saveTime()
    ✓ exitEditMode()
    ✓ resetAlarm() 1 ms
    ✓ chooseModes() 1 ms
    ✓ resetTimer() 1 ms
    ✓ enableAlarm()
    ✓ enterEditMode()
    ✓ activateStopwatch()
    ✓ disableAlarm()
    ✓ changeAlarmPage()
    ✓ pauseStopwatch() 1 ms
    ✓ changeCursor()
    ✓ disableIntervalTimer()
    ✓ exitSetMode()
    ✓ pauseTimer() 1 ms
    ✓ saveAlarm() 3 ms
    ✓ saveTimer() 1 ms
    ✓ saveIntervalTimer() 1 ms
    ✓ changeHourFormat() 2 ms
    ✓ muteBeep() 31 ms
    ✓ changeDdayFormat() 4 ms
    ✓ enterSetMode() 1 ms
    ✓ resetIntervalTimer() 1 ms
    ✓ activateTimer() 1 ms
    ✓ enableIntervalTimer()
    ✓ resetStopwatch() 1 ms
    ✓ increaseData() 1 ms
    ✓ resetDday()
  ▼ IntervalTimerTest 7 ms
    ✓ enable() 1 ms
    ✓ loadIntervalTimer() 1 ms
    ✓ ring() 4 ms
    ✓ saveIntervalTimer() 1 ms
    ✓ disable()

```

4. Activity 2063. System Testing

Test Number	Test 항목	Description	Use Case	System Function	Pass
1-1	Set time test	현재 시간을 set time기능을 사용하여 시간이 저장되는 지 test	1. Set Time	R 1.1	P
1-2	Set time test	현재 시간을 set time기능을 사용해 변경하다가 저장을 하지 않고 나오는 지 test	1. Set Time	R 1.1	P
2	Format change test	12 hour format 시간을 24 hour format 시간으로 변경 Vice versa 하는 기능을 test	2. Set Hour Format	R 1.2	P

3-1	Activate timer test	Timer가 activate 되었을때 지정된 시간으로 부터 주기적으로 1초씩 discount 되는지 test	3. Activate Timer	R 2.1	P
3-2	Activate timer test	입력받은 Timer가 존재하지 않을 시, 작동하지 않는지 test	3. Activate Timer	R 2.1	P
4-1	Set Timer test	사용자가 Set Timer 기능을 사용해 timer 시간을 지정해 주었을때 제대로 반영되는지 test	4. Set Timer	R 2.2	P
4-2	Set Timer test	사용자가 Set Timer 기능을 사용해 변경하다가 저장을 하지 않고 나오는지 test	4. Set Timer	R 2.2	P
5	Notify finish Timer test	Timer에서 지정된 시간이 되었을때 BEEP로 알려주는지 test	5. Notify Finish Timer	R 2.3	P
6	Pause Timer test	Timer에서 버튼을 눌렀을때 Timer의 기능이 일시 정지되는지 test	6. Pause Timer	R 2.4	P
7-1	Reset Timer test	Timer에서 버튼을 눌렀을때 Timer의 시간이 0으로 초기화되는지 test	7. Reset Timer	R 2.5	P
7-2	Reset Timer test	Timer가 초기화되어 있는 경우 아무런 동작을 안하는지 test	7. Reset Timer	R 2.5	P
8	Activate Stopwatch test	Stopwatch가 Activate 되었을 때 0에서 부터 주기적으로 1초씩 Count 되는지 test	8. Activate Stopwatch	R 3.1	P
9	Pause Stopwatch test	Stopwatch에서 버튼을 눌렀을때 Stopwatch의 기능이 일시 정지되는지 test	9. Pause Stopwatch	R 3.2	P
10	Reset Stopwatch test	Stopwatch에서 버튼을 눌렀을때 Stopwatch의 시간이 0으로 초기화되는지 test	10. Reset Stopwatch	R 3.3	P
11-1	Set alarm test	set alarm기능을 사용하여 4개의 알람을 저장할 수 있는지 test	11. Set alarm	R 4.1	P
11-2	Set alarm test	set alarm 기능을 사용해 알람을 저장 중 나올 경우 저장이 안되는지 test	11. Set alarm	R 4.1	P
12-1	Reset alarm test	1~4개의 alarm 중 지정된 alarm이 제거되는지 test	12. Reset alarm	R 4.2	P
12-2	Reset alarm test	1~4개의 alarm 중 지정된 alarm을 reset 중 저장하지 않고 나오는지 test	12. Reset alarm	R 4.2	P
13	Enable alarm test	1~4개의 alarm 중 지정된 alarm이 비활성화에서 활성화되는지 test	13. Enable alarm	R 4.3	P
14	Disable alarm test	1~4개의 alarm 중 지정된 alarm이 활성화에서 비활성화되는지 test	14. Disable alarm	R 4.4	P
15	Notify alarm test	1~4개의 alarm이 지정된 시간에 순차적으로 잘 작동하는지 test	15. Notify alarm	R 4.5	P
16	Change alarm page	1~4개의 alarm 페이지를 잘 이동하는지 test	16. Change alarm page	R 4.6	P
17-1	Set D-day test	D-day를 입력받고 반영되었는지 test	17. Set D-day	R 5.1	P
17-2	Set D-day test	D-day를 입력받다가 빠져나왔을 때 저장이 안되어있는지 test	17. Set D-day	R 5.1	P
18	Reset D-day test	초기화 버튼을 눌렀을 때, 날짜가 초기화 되는지 test	18. Reset D-day	R 5.2	P

19	Notify D-day test	D-day 시간이 되었을 때, 해당 시간에 알람이 울리는지 test	19. Notify D-day	R 5.3	P
20-1	Set D-day Format test	D-day 표현 방식이 percentage에서 남은 일수로 잘 바뀌는지 test	20. Set D-day Format	R 5.4	P
20-2	Set D-day Format test	D-day 표현 방식이 남은 일수에서 percentage로 잘 바뀌는지 test	20. Set D-day Format	R 5.4	P
21	Enable Interval Timer test	Interval Timer을 활성화 시켰을 때, 제대로 동작하는 test	21. Enable Interval Timer	R 6.1	P
22	Disable Interval Timer test	Interval Timer을 비활성화 시켰을 때, 해당시간에 안 울리는지 test	22. Disable Interval Timer	R 6.2	P
23-1	Set Interval Timer test	Interval Timer을 사용자에게 입력받고 해당 시간이 반영되는지 test	23. Set Interval Time	R 6.3	P
23-2	Set Interval Timer test	Interval Timer을 사용자에게 입력받는 중 빠져나오면 저장이 안되어있는지 test	23. Set Interval Timer	R 6.3	P
24-1	Reset Interval Timer test	초기화 했을 때, 해당하는 Interval Timer 이 초기화 되는지 test	24. Reset Interval Timer	R 6.4	P
24-2	Reset Interval Timer test	초기화 하는 중, 빠져나왔을 때 저장이 안되어있는지 test	24. Reset Interval Timer	R 6.4	P
25	Notify Finish Interval Timer	반복되는 해당하는 시간에 알람이 울리는지 test	25. Notify Finish Interval Timer	R 6.5	P
26	Change Mode test	Mode 버튼을 눌렀을 때, 순차적으로 모드가 변경되는지 test	26. Change Mode	R 7.1	P
27-1	Set Mode test	6개의 기능중 4개의 기능을 골랐을 때, 해당하는 기능들이 반영되었는지 test	27. Set Mode	R 7.2	P
27-2	Set Mode test	4개의 기능보다 적게 골랐을 때 저장이 안되는지 test	27, Set Mode	R 7.2	P
27-3	Set Mode test	기능을 선택할 때 중간에 빠져나갈 경우 저장이 안되어있는지 test	27. Set Mode	R 7.2	P
28	Mute Beep test	버튼을 눌렀을 때, 알람이 종료되는지 test	28. Mute Beep	R 7.3	P

5. Activity 2067. Testing Traceability Analysis

Operation in sequence diagram	Operation in interaction diagram
enterEditMode	enterEditMode()
	loadTime()
	loadAlarm(currentAlarmPage)
	loadAlarmData()
	loadStartDay()
	loadEndDay()
	loadIntervalTime()
increaseData	increaseData()
changeCursor	changeCursor()
saveTime	saveTime()
	saveTime(data)
	exitEditMode()
changeHourFormat	changeHourFormat()
	setHourFormat()
activateTimer	activateTimer()
	activate()
	ring()
saveTimer	saveTimer()
	saveTimer(data)
	exitEditMode()
pauseTimer	pauseTimer()
	pause()
resetTimer	resetTimer()
	reset()
activateStopwatch	activateStopwatch()
	activate()
pauseStopwatch	pauseStopwatch()
	pause()
resetStopwatch	resetStopwatch()
	reset()
saveAlarm	saveAlarm()
	saveAlarm(currentAlarmPage, data)
	saveAlarmData()
	exitEditMode()
resetAlarm	resetAlarm()
enableAlarm	enableAlarm(currentAlarmPage)
	enable()
	getCurrentTime()
	ring()
disableAlarm	disableAlarm()
	disableAlarm(currentAlarmPage)
	disable()
changeAlarmPage	changeAlarmPage()
	loadAlarm(currentAlarmPage)
changePage	loadAlarmData()
saveDday	savePage()
	saveDday(data1, data2)
	ring()
resetDday	exitEditMode()
	resetDday()
	reset()
	getCurrentTime()
changeDdayFormat	changeDdayFormat()
	changeFormat()
enableIntervalTimer	enableIntervalTimer()
	enable()
	ring()
disableIntervalTimer	disableIntervalTimer()
	disable()
saveIntervalTimer	saveIntervalTimer()
	saveIntervalTimer(data)
	exitEditMode()
resetIntervalTimer	resetIntervalTimer()
changeMode	changeMode()
enterSetMode	enterSetMode()
	loadSetMode()
chooseModes	chooseModes()
saveMode	saveMode()
	createTimer()
	createStopwatch()
	createAlarm()
	createDday()
	createIntervalTimer()
muteBeep	muteBeep()
exitEditMode	exitEditMode()
exitSetMode	exitSetMode()

Method	Class	
enterEditMode(): Time	WatchSystem	
increaseData(): Time		
changeCursor(): int		
saveTime(): void		
changeHourFormat(): Time		
activateTimer(): void		
saveTimer(): void		
pauseTimer(): void		
resetTimer(): void		
activateStopwatch(): void		
pauseStopwatch(): void		
resetStopwatch(): void		
saveAlarm(): void		
resetAlarm(): Time		
enableAlarm(): void		
disableAlarm(): void		
changeAlarmPage(): AlarmTime		
changePage(): Time		
saveDday(): void		
resetDday(): Time		
changeDdayFormat(): int		
enableIntervalTimer(): void		
disableIntervalTimer(): void		
saveIntervalTimer(): void		
resetIntervalTimer(): void		
changeMode(): int		
enterSetMode(): int		
chooseModes(): void		
saveMode(): void		
muteBeep(): void		
exitEditMode(): void		
loadTime(): Time		Time Keeping
loadAlarm(): void		Timer
saveTime(data): void		
setHourFormat(): void		
activate(): void		
loadTimer(): Time		
saveTimer(data): void		
pause(): void		
reset(): void		
ring(): void		
activate(): void	StopWatch	
pause(): void		
reset(): void		
loadAlarm(currentAlarmPage): Time	Alarm	
saveAlarm(currentAlarmPage, data): void		
enableAlarm(currentAlarmPage): void		
disableAlarm(currentAlarmPage): void		
saveAlarmData(): void	AlarmTime	
enable(): void		
disable(): void		
ring(): void		
loadStartDay(): Time		
loadEndDay(): Time		
saveDday(data1, data2): void		
reset(): Time		
changeFormat(): int		
ring(): void		
enable(): void		
disable(): void		
loadIntervalTimer(): Time	IntervalTimer	
saveIntervalTimer(data): void		
ring(): void		
loadSetMode(): int		
toggleMode(): int		
createTimer(): Timer	ModeManager	
createStopwatch(): Stopwatch		
createAlarm(): Alarm		
createDday(): Dday		
createIntervalTimer(): IntervalTimer		

SystemTest
Set time test
Set time test
Format change test
Activate timer test
Activate timer test
Set Timer test
Set Timer test
Notify finish Timer test
Pause timer test
Reset timer test
Reset timer test
Reset timer test
Activate Stopwatch test
Pause Stopwatch test
Reset Stopwatch test
Set alarm test
Set alarm test
Reset alarm test
Reset alarm test
Enable alarm test
Disable alarm test
Notify alarm test
Change alarm page
Set D-day test
Set D-day test
Reset D-day test
Notify D-day test
Set D-day Format test
Set D-day Format test
Enable Interval Timer test
Disable Interval Timer test
Set Interval Timer test
Set Interval Timer test
Reset Interval Timer test
Reset Interval Timer test
Notify Finish Interval Timer
Change Mode test
Set Mode test
Set Mode test
Set Mode test
Mute Beep test