

# Check It Out - Digital Watch

## 2019 Software Modeling & Analysis 대응보고서

T6

201613856 소아이린

201711381 김소현

201711401 염혜지

201711420 임수연

201711428 조은지

# Contents

1. Requirements Correspondence Response
2. System Test Response

# 1. Requirements Correspondence Response

## A. Stage 1000 Planning

문제	Non-functional Requirement 의 일부 내용의 기준이 명확하지 않다.
대응	시계조작이 쉬워야 한다는 점을 버튼의 통일성으로 명시해주었다. 시계 ui가 간단해야한다는 점을 필수적인 정보만 화면에 보여줘야한다는 것으로 명시해주었다.

문제	Performance Requirement 의 일부 내용의 기준이 명확하지 않다.
대응	모든 입력에 대한 기기의 반응을 각 버튼에 대한 요청, 요청에 해당하는 기능으로 명시해 주었다.

문제	Test Case 의 목록선택시험의 내용이 Requirement의 내용과 일치하지 않는다.
대응	목록선택시험에서 '기능을 선택했을 때 개수가 4개 이하 인지 확인한다' 는 내용을 '기능을 선택할 때 4개의 기능을 선택한 후 기능조합을 완료할 수 있는지 확인한다' 로 고친다.

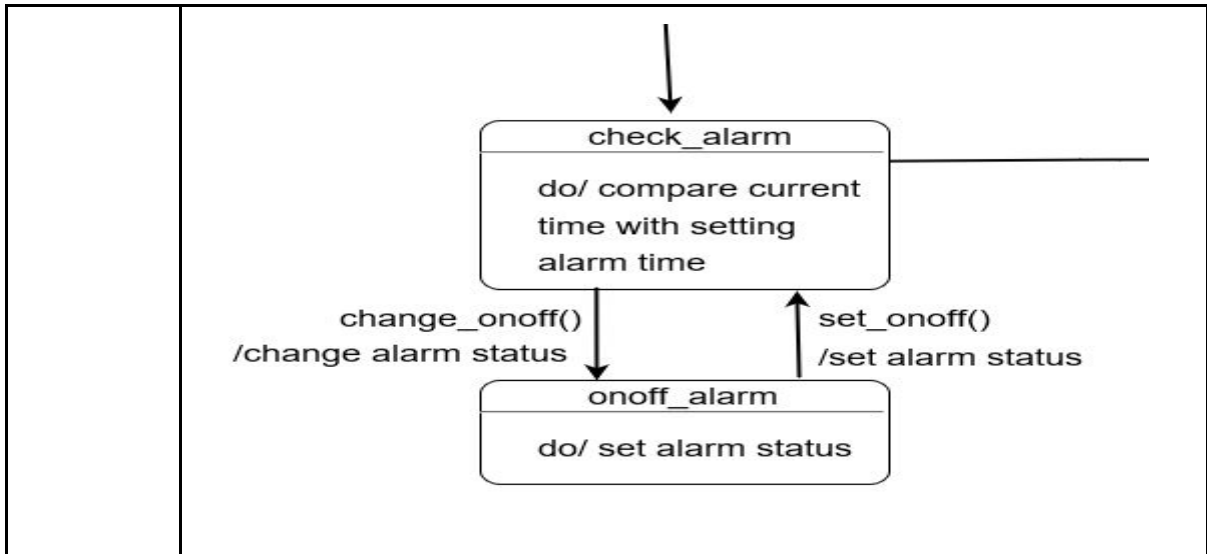
		기능을 선택할 때 4개의 기능을 선택한 후 기능조합을 완료할 수 있는지, 실행조합목록에 넣어졌는지 확인	8. <u>select_funcio</u> <u>n</u>	Evident	R.3.3.2
8	목록선택시험				

## B. Stage 2030 Planning

문제	시간보기 기능의 state diagram에서 2월의 경우가 누락되어 있다.
대응	2월의 경우를 추가해준다.
	<pre> stateDiagram-v2     state show_current_year {         entry / show_current_year     }     state set_date {         entry / show_current_date         exit / save Day     }     show_current_year --&gt; set_date : decide()     set_date --&gt; show_current_year : inc() / month = month / mod 12 + 1     set_date --&gt; set_date : inc() [month == 4, 6, 9, 11] / date = date / mod 30 + 1     set_date --&gt; set_date : inc() [other month] / date = date / mod 31 + 1     set_date --&gt; set_date : inc() [month == 2 &amp; year == leap_year] / date = date / mod 29 + 1     set_date --&gt; set_date : inc() [month == 2 &amp; year != leap_year] / date = date / mod 28 + 1     set_date --&gt; set_date : inc() [other month] / date = date / mod 31 + 1     set_date --&gt; set_date : decide()     </pre>

문제	시간보기 기능의 state diagram에서 hour을 mod 60으로 연산하는 것이 적절치 않다.
대응	hour의 대입값을 $hour = (hour+1)/24$ 로 수정해준다.
	<pre> stateDiagram-v2     state set_hour {         entry / show_current_hours     }     state set_minute     set_hour --&gt; set_minute : decide()     set_hour --&gt; set_hour : inc() / hour = (hour + 1) / mod 24     </pre>

문제	알람 기능의 state diagram에서 onoff_alarm에 대한 기능이 포함되어 있지 않다.
대응	onoff_alarm을 추가해준다.



문제	알람 기능의 state diagram에서 ring_buzzer() 함수 호출 시 alarm 에 대한 on/off 설정을 했는지 검사하는 부분이 미구현 되어있다.
대응	on/off 설정 검사를 추가해준다.

문제	스탑워치 기능의 state diagram에서 delete_record 기능이 적절치 않은 곳에 들어가 있다.
대응	delete_record() 기능을 show_record 에서가 아닌 record_stopwatch()에서 구현되는 기능으로 바꿔준다.

## C. Stage 2040 Planning

문제	Stage 2030에서의TimeKeeping 기능의 set_time Use Case와 Stage 2040에서의 내용이 일관성이 없다.																
대응	Stage 2030의 ‘현재 초는 0초로 설정한다’는 내용을 빼고 시간을 설정하는 단계에 초단위를 포함시킨다.																
	<table border="1"> <tr> <td data-bbox="395 1249 683 1285">Use Case</td> <td data-bbox="683 1249 1359 1285">set_time</td> </tr> <tr> <td data-bbox="395 1285 683 1321">Actor</td> <td data-bbox="683 1285 1359 1321">user/system</td> </tr> <tr> <td data-bbox="395 1321 683 1357">Purpose</td> <td data-bbox="683 1321 1359 1357">현재 날짜/시/분 을 설정한다.</td> </tr> <tr> <td data-bbox="395 1357 683 1438">Overview</td> <td data-bbox="683 1357 1359 1438">User는 날짜와 시간을 설정한다. 분 단위 까지의 설정을 마치면, 시스템은 User가 설정한 시간을 저장한다.</td> </tr> <tr> <td data-bbox="395 1438 683 1473">Type</td> <td data-bbox="683 1438 1359 1473">Evident</td> </tr> <tr> <td data-bbox="395 1473 683 1509">Cross Reference</td> <td data-bbox="683 1473 1359 1509">Functions : R.4.1 / Use Cases : look_time</td> </tr> <tr> <td data-bbox="395 1509 683 1545">Pre-Requisites</td> <td data-bbox="683 1509 1359 1545">현재 상태가 시간 보기 상태여야한다.</td> </tr> <tr> <td data-bbox="395 1545 683 1877">Typical Courses of Event</td> <td data-bbox="683 1545 1359 1877"> <div style="border: 1px solid black; padding: 5px;"> <p>(A): Actor, (S):System</p> <ol style="list-style-type: none"> <li>1.(A) User가 버튼을 통해 현재 시각(연도, 월, 일, 시, 분,초)선택을 요청한다.</li> <li>2.(S) 화면에 시각을 보여준다.</li> <li>3.(A) User가 버튼을 통해 현재 시각(연도,월,일,시,분,초)을 설정한다.</li> <li>4.(S) 현재 날짜에 해당하는 요일을 계산한다.</li> <li>5.(S) 설정한 시, 분, 초를 현재 시각으로 저장한다.</li> <li>6.(A) 시간 설정 종료를 요청한다.</li> <li>7.(S) look_time상태로 전환한다.</li> </ol> </div> </td> </tr> </table>	Use Case	set_time	Actor	user/system	Purpose	현재 날짜/시/분 을 설정한다.	Overview	User는 날짜와 시간을 설정한다. 분 단위 까지의 설정을 마치면, 시스템은 User가 설정한 시간을 저장한다.	Type	Evident	Cross Reference	Functions : R.4.1 / Use Cases : look_time	Pre-Requisites	현재 상태가 시간 보기 상태여야한다.	Typical Courses of Event	<div style="border: 1px solid black; padding: 5px;"> <p>(A): Actor, (S):System</p> <ol style="list-style-type: none"> <li>1.(A) User가 버튼을 통해 현재 시각(연도, 월, 일, 시, 분,초)선택을 요청한다.</li> <li>2.(S) 화면에 시각을 보여준다.</li> <li>3.(A) User가 버튼을 통해 현재 시각(연도,월,일,시,분,초)을 설정한다.</li> <li>4.(S) 현재 날짜에 해당하는 요일을 계산한다.</li> <li>5.(S) 설정한 시, 분, 초를 현재 시각으로 저장한다.</li> <li>6.(A) 시간 설정 종료를 요청한다.</li> <li>7.(S) look_time상태로 전환한다.</li> </ol> </div>
Use Case	set_time																
Actor	user/system																
Purpose	현재 날짜/시/분 을 설정한다.																
Overview	User는 날짜와 시간을 설정한다. 분 단위 까지의 설정을 마치면, 시스템은 User가 설정한 시간을 저장한다.																
Type	Evident																
Cross Reference	Functions : R.4.1 / Use Cases : look_time																
Pre-Requisites	현재 상태가 시간 보기 상태여야한다.																
Typical Courses of Event	<div style="border: 1px solid black; padding: 5px;"> <p>(A): Actor, (S):System</p> <ol style="list-style-type: none"> <li>1.(A) User가 버튼을 통해 현재 시각(연도, 월, 일, 시, 분,초)선택을 요청한다.</li> <li>2.(S) 화면에 시각을 보여준다.</li> <li>3.(A) User가 버튼을 통해 현재 시각(연도,월,일,시,분,초)을 설정한다.</li> <li>4.(S) 현재 날짜에 해당하는 요일을 계산한다.</li> <li>5.(S) 설정한 시, 분, 초를 현재 시각으로 저장한다.</li> <li>6.(A) 시간 설정 종료를 요청한다.</li> <li>7.(S) look_time상태로 전환한다.</li> </ol> </div>																

문제	이전 문서에 없던 database가 등장했다.
대응	2030단계에서 DB도메인을 추가해준다.
	<pre> classDiagram     class User     class Clock     class Buzzer {         +check_ring: Boolean     }     class Time {         +time: Time     }     class Timers {         +setting_time: Time     }     class Alarm {         +setting_time: Time         +repeat_alarm: Boolean     }     class Stopwatch {         +pass_time: Time     }     class D_plus_day {         +startDay: date         +goal: String     }     class Fitness {         +Calories: int         +time: Time         +exercise: String     }     class TimeKeeping {         +current_time: Time     }     class DB {         +sw_record: swRecord         +fit_record: firRecord     }      User "1" -- "1" Clock : Has a     Clock "1" -- "1" Buzzer : Has a     Clock "1" -- "1" Time : Has a     Buzzer "1" -- "1" Timers : Use a     Buzzer "1" -- "1" Alarm : Use a     Time "0..30" -- "0..10" Timers : Use a     Time "1" -- "1" Alarm : Use a     Time "1" -- "0..4" Stopwatch : Use a     Time "1" -- "1..6" D_plus_day : Use a     Time "1" -- "1" TimeKeeping : Use a     Time "1" -- "1" DB : refer to     Timers "1" -- "1" Alarm : refer to     Alarm "1" -- "1" Stopwatch : refer to     Alarm "1" -- "1" D_plus_day : refer to     Alarm "1" -- "1" Fitness : refer to     Stopwatch "1" -- "1" DB : refer to     D_plus_day "1" -- "1" Fitness : refer to     TimeKeeping "1" -- "1" Fitness : refer to     DB "1" -- "1" Fitness : Use     </pre>

문제	잘못된 형을 비교했다.
대응	status의 데이터타입을 String으로 바꾸고, 실제 구현에 반영했다.

문제	처음 나오는 용어를 정의하지 않고 사용했다.
대응	상태를 나타내는 변수명을 모두 status로 수정한다.  setSecond의 상태를 나타내는 status는 개념적으로만 사용이 되고, 실제구현에는 반영이 되지 않았기 때문에 조건을 삭제한다.  다른 유즈케이스도 마찬가지로 언급이 되어있지 않은 경우 삭제해준다.

## D. Stage 2050 Planning

문제	Method Description에 use case를 user case로 오타자를 낸 부분이 있다.
대응	user 을 use 로 오타자를 수정한다.

## 2. System Test Response

### 1. Timekeeping

문제	P10 : Timekeeping 기능에서 시간을 설정할 수 없다.
대응	시, 분, 초를 모두 마친 후 완료 버튼을 누르면 req_setTime()함수를 호출하도록 한다.



문제	C4 : TimeKeeping 년도 설정이 불가능하다. C5 : TimeKeeping 월 설정이 불가능하다. C5 : TimeKeeping 일 설정이 불가능하다.
대응	연,월,일의 설정을 마치면 req_setDate()함수를 호출하도록 한다.

문제	C7 : TimeKeeping에서 12-23시로 설정이 불가능하다.
대응	Timekeeping클래스에 값을 넘길 때, Calendar.HOUR(12시간 단위) 이 아닌 Calendar.HOUR_OF_DAY(24시간 단위)

### - Timekeeping 수정 코드

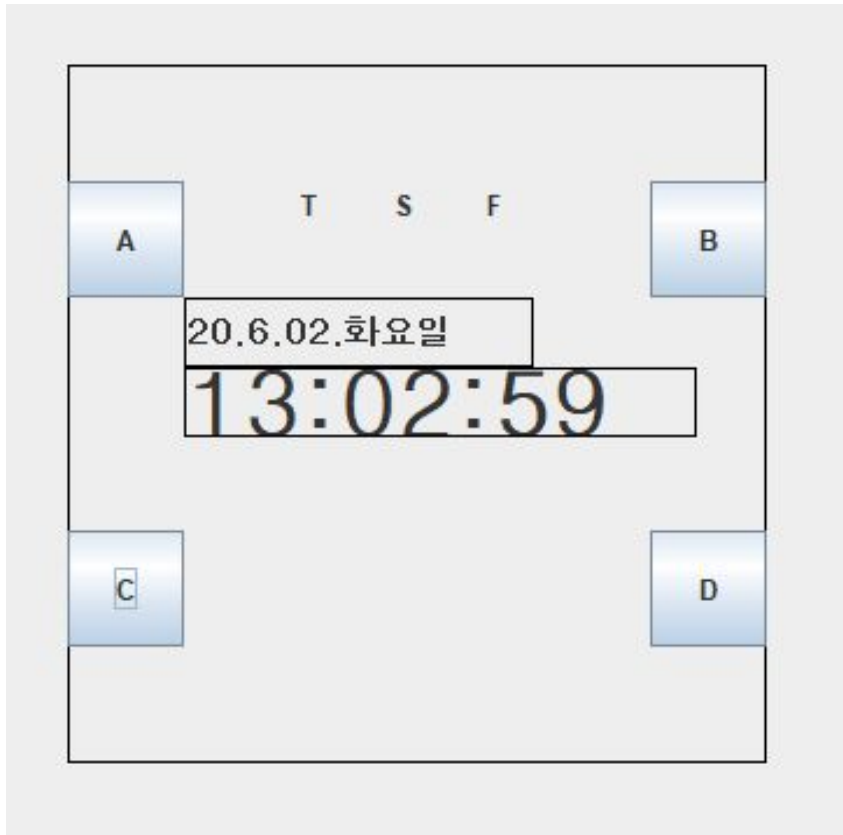
```

if(tk_status.equals("Setting") == true){
/이 때 버튼을 OK(Next)
    settingNum++;
    System.out.println(settingNum);
    if(settingNum==3)
    {
        base.controller.req_setDate( status: "timekeeping",calendar.get(Calendar.YEAR),calendar.get(Calendar.MONTH),calendar.get(Calendar.DAY_OF_MONTH));
        Timekeeping tk = InstManager.getInstance().getTimekeeping();
        System.out.println(tk.getDate());
    }
    else if(settingNum==6) {
        base.controller.req_setTime( status: "timekeeping",calendar.get(Calendar.HOUR_OF_DAY), calendar.get(Calendar.MINUTE));
        settingNum = 0;
        tk_status="Timekeeping";
        base.controller.req_continue( status: "timekeeping");
    }
}

```

버튼이 3번째 눌릴 때,(일 설정을 마친 후 ) req\_setDate()함수 호출, 초 단위의 시간까지 마친 후 req\_setTime()함수 호출 뒤 바로 시간보기 화면으로 넘어간다.

### - 수정 후 화면



연, 월, 일을 설정할 수 있음을 볼 수 있다.

## 2. Timer

문제	p11: 타이머 정지상태에서 시작할 수 없다.
대응	타이머 초기화면에서 시/분/초 설정 후, 일시정지 버튼을 눌러 countdown을 시작함으로써 타이머 기능을 실행할 수 있다.

문제	p12: 버저가 울리고, 타이머 정지상태에서 재시작할 수 없다.
대응	대응 중

문제	p19: 버저가 울리고 타이머 실행상태에서 다음 기능으로 이동할 수 없다.
대응	버저가 울리는 중에 mode버튼을 눌러 기능 전환이 가능하다.

문제	c10: 타이머 작동 후 시간이 만료되었을 경우 버저가 울리지 않는다.
대응	버저의 경로를 상대경로로 수정함으로써 타이머가 00:00:00에 도달하는 경우 버저가 울린다.

문제	c11: 타이머로 인해 버저가 울리고 있는 상태에서 버저를 멈출 수 있다.
대응	대응 중

문제	c12: 타이머 작동 시 초 단위로 값이 감소하지 않는다.
대응	TimerTask를 사용해서 타이머 작동 시 화면에서 초단위로 값이 감소시킨다.

문제	c13: 타이머 일시정지 상태에서 버저가 울리지 않아야한다.
대응	타이머 설정 후 일시정지버튼을 눌러 req_pause()를 호출한 결과, 설정한 시간만큼 기다려도 버저가 울리지 않는다.

문제	c14: 타이머 일시정지 상태에서 시간 값이 증가하지 않아야한다.
대응	타이머 일시정지버튼을 눌러 req_pause()를 호출한 결과, 시/분/초 값이 변하지 않는다.

문제	c15: 타이머 재시작 후 타이머가 만료되면 버저가 울린다.
대응	대응 중

문제	c16: 타이머 재시작 후 타이머로 인해 버저가 발생할 경우 버저를 멈출 수 있다.
대응	대응 중

문제	c17: 타이머 재시작 후 초 읽기가 진행되며, 임계치 초과 시 다음 단위 값이 증가한다.
대응	대응 중

문제	c18: 타이머 취소 후, 타이머로 인해 버저가 울리지 않아야 한다.
대응	타이머를 설정 후 reset버튼을 눌러 req_reset()을 호출한 결과, 설정시간만큼 시간이 지나도 버저는 울리지 않는다.

문제	c19: 타이머 취소 상태에서 시간값이 증가하지 않아야한다.
대응	reset버튼을 눌러 req_reset()을 호출한 결과, 시간값은 00:00:00으로 유지된다.

문제	c20: 타이머 다음 기능으로 이동 후, 다시 타이머 기능을 이용할 경우 이전 내용이 저장되어 있어야한다.
대응	타이머 일시정지 또는 취소 후, 다른 기능으로 mode전환을 했다가 다시 타이머로 돌아오면 시간값은 이전과 동일하다. (count 중 기능전환은 B1에서 대응완료)

문제	c21,22,23: 타이머 설정에서 시/분/초를 설정할 수 없다.
대응	up버튼, down버튼, ok버튼을 사용해서 타이머의 시/분/초를 설정할 수 있다.

문제	B1: 타이머에서 countdown 시작 후, 다른 모드로 변경 후 다시 돌아왔을 때 동작하는지 확인한다.
대응	countdown실행 중 mode전환 후 다시 타이머로 돌아와도 countdown은 여전히 실행 중이다.

문제	B7: 타이머의 버저 울리는 시간을 설정한 후 어떻게 동작하는지 확인한다.
대응	타이머에서 버저가 울리는 시간은 30초로 정해져있다.

### 3. Alarm

문제	B8 : Alarm의 부저 울리는 시간을 맞춘 후 어떻게 동작하는지 테스트
대응	대응중

문제	C25 : Alarm 일정 삭제가 정상적으로 이루어지지 않는다.
대응	req_deleteAlarm()을 호출하여, Alarm 일정을 삭제할 수 있다.

문제	C26 : Alarm 활성화가 이루어지지 않는다.
대응	req_onOff()를 호출하여, Alarm 활성화가 정상적으로 이루어지고, 이를 알람 리스트에서 확인할 수 있다. (dot 화면에 On 표시)

문제	C29 : Alarm 비활성화가 이루어지지 않는다.
대응	req_onOff()를 호출하여, Alarm 비활성화가 정상적으로 이루어지고, 이를

	알람 리스트에서 확인할 수 있다. (dot 화면에 Off 표시)
--	-------------------------------------

문제	C30 : Alarm 비활성화가 이루어지지 않는다.
대응	대응중

문제	C32 : 요일을 선택하지 않고 주기로 넘어갈 경우 다시 요일 설정으로 돌아갈 수 없다.
대응	요일을 하나도 선택하지 않은 경우, 주기 설정으로 넘어갈 수 없다. -> 요일을 하나도 선택하지 않은 경우, '일요일'에서 Next 버튼을 눌러도 주기 설정으로 넘어가지 않고, 다시 '월요일'로 돌아온다. -> 요일을 하나라도 선택하면, '일요일'에서 Next 버튼을 누를 시 주기 설정으로 넘어간다.

문제	C33 : 주기를 0으로 설정할 수 없다.
대응	알람 주기를 0, 5, 10, 15 중 하나로 설정할 수 있다.

문제	P1 : Alarm이 정지 상태에서, 알람 리스트를 확인할 수 없다.
대응	알람 정지 상사(OFF 상태)에서, 알람 리스트를 확인할 수 있다.

문제	P5 : Alarm 활성화 상태에서, 비활성화가 불가능하다.
대응	대응중

문제	P6 : Alarm이 활성화 상태에서, 알람 일정을 추가할 수 없다.
대응	알람 활성화 상태(ON)상태에서, 알람 일정을 추가할 수 있다.

문제	P20 : 버저가 울리고, Alarm 비활성화 상태에서, 알람을 활성화할 수 없다.
대응	대응중

문제	P21 : 버저가 울리고, Alarm 활성화 상태에서, 다음 기능으로 넘어갈 수 없다.
대응	대응중

#### 4. Stopwatch

문제	p4: 스톱워치 실행상태에서 일시정지를 할 수 없다.
대응	스톱워치 실행 후 일시정지 버튼을 눌러 countUp을 중지할 수 있다.

문제	p7: 버저가 울리고 스톱워치 실행 상태에서 기록을 확인할 수 없다
대응	스톱워치는 버저기능을 사용하지 않는다.

문제	p8: 버저가 울리고 스톱워치 정지상태에서 다음 기능으로 넘어갈 수 없다.
대응	스톱워치는 버저기능을 사용하지 않는다.

문제	p13: 스톱워치 실행상태에서 기록할 수 없다.
대응	스톱워치 실행 중, record버튼을 눌러 현재 시/분/초를 record list에 저장할 수 있다.(c37과 중복)

문제	p14: 스톱워치 정지상태에서 기능을 시작할 수 없다.
대응	스톱워치 초기화면 00:00:00에서 start버튼을 누르면 countUp이 진행된다.

문제	p15: 버저가 울리고 스톱워치 정지상태에서 초기화할 수 없다.
대응	스톱워치는 버저기능을 사용하지 않는다.

문제	c2: 스톱워치는 11개 이상의 기록을 저장할 수 없어야한다. (기능미구현)
대응	recordlist에서 기록이 10개를 초과하면 가장 오래된 기록을 삭제하여 저장된 총 기록의 개수가 10개를 초과할 수 없게 한다.

문제	c36: 스톱워치 실행 시 시간이 초단위로 증가하지 않는다.(기능미구현)
대응	Timertasking을 이용해서 countup 실행 시, 시간이 초단위로 증가한다.

문제	c37: 스톱워치 실행 중 시간을 기록할 수 있어야한다.(기능미구현)
대응	스톱워치가 countup되는 중에 record버튼을 누르면 req_record를 호출하여 현재 시/분/초를 기록할 수 있다. (p13과 중복)

문제	c38: 스톱워치를 일시정지하면 시간이 일정하게 유지되어야한다.(기능미구현)
----	--

대응	일시정지 버튼을 눌러 req_pause()를 호출하면 시간값이 변하지 않는다.
----	---

문제	c39: 스톱워치 재실행시, 시간이 초단위로 증가하지 않는다.
대응	일시정지/계속 버튼을 눌렀을 때, 시간값이 00:00:00이 아니라면 req_continue()를 호출해서 countup이 진행되도록 코드를 수정하였다.

문제	c40: 스톱워치 초기화 후, 시간이 일정하게 유지되어야한다. (기능미구현)
대응	reset버튼을 눌러 req_finish()를 호출하면 시간 값이 00:00:00으로 유지된다.

문제	c41: 스톱워치의 현재 정보를 저장한 후 다음 기능으로 넘어간 뒤, 다시 스톱워치로 돌아왔을 때 이전 정보가 저장되어 있어야한다. (기능미구현)
대응	스톱워치 일시정지 또는 기록화면에서 mode전환으로 다른 기능으로 넘어간 뒤 다시 스톱워치로 돌아왔을때 도트 화면과 세그먼트 화면은 이전과 동일하다.(count 중 기능 전환은 B2에서 대응)

문제	c42: 스톱워치의 기록리스트에서 다음 기록을 볼 수 있어야한다.(기능미구현)
대응	기록보기 화면에서 next버튼을 눌러 다음 기록을 볼 수 있다.

문제	c43: 스톱워치에서 기록을 하면 해당 기록들을 기록 리스트에서 볼 수 있어야 한다.(기능미구현)
대응	스톱워치 실행을 일시정지시키고 record버튼을 눌러 req_recordList()를 호출함으로써 이전 기록들을 볼 수 있다.

문제	c44: 스톱워치의 기록을 지울 수 있어야한다.
대응	스톱워치의 기록은 사용자가 지울 수 없으며, 단지 기록의 개수가 10개를 초과할때마다 가장 오래된 기록이 자동으로 삭제될 뿐이다.

문제	B2: 스톱워치에서 countup 시작 후, 다른 모드로 변경 후 다시 돌아왔을 때 동작하는지 확인한다.
대응	countup실행 후 mode전환을 하고 다시 스톱워치로 돌아와도 스톱워치 countup은 여전히 실행 중이다.

## 5. D+day

문제	B4 : D+Day 에서 년도 임계치를 설정했는지 검사.
대응	D+Day 에서 년도 임계치를 현재날짜로 설정 완료했다.

문제	B5 : D+Day 에서 년도 임계치를 설정했을 때, 정상적으로 동작하는지 검사한다.
대응	D+Day 년도 임계치(현재 날짜)를 설정했을 때, 정상적으로 동작한다.

문제	B9 : D+Day에서 일정을 설정해놓고 set Timekeeping을 통해 시간을 바꾸었을때 정상 계산하는지 검사
대응	대응중

문제	P22 : 버저가 울리고, DDay 활성화 상태에서, 리스트를 확인할 수 없다.
대응	대응중

## 6. Fitness

공통적으로 DB연결에 문제있음. 외부에서 접근이 불가능한 문제가 있다. 기능적 측면은 수정

문제	P16 : Fitness 측정 상태에서, 운동량 계산이 이루어지지 않는다.
대응	실시간으로 운동량이 계산되는 것이 보이도록 해준다.

문제	B10 : Fitness 에서 1분 단위 업데이트 직전에 운동 종목을 바꾸고 운동량 계산 검사
대응	운동실행 상태에서는 운동종목을 선택할 수 없도록 했다.

문제	C61 : Fitness 기능 일시정지 및 재실행이 간헐적으로 작동하지 않는다. -운동 정보 저장 문제
대응	일시정지를 했을 때 해당 운동시간과 칼로리량을 저장해두고, 재실행을 했을 때 이전 운동시간부터 카운트업을 할 수 있도록 한다.

문제	B6 : Fitness 에서 default값으로 했을 때, 작동하지 않는지 검사
----	--



대응	default값일 때는 운동시작 상태로 전환을 할 수 없도록 하였다.
----	--

문제	C63 : Fitness 측정 완료시 운동량이 저장되지 않는다.
----	-------------------------------------

대응	DB접속 문제 - 같은 날짜일 때는 해당날짜의 레코드에 값을 업데이트 하고, 날짜가 바뀌었을 때는 레코드를 추가할 수 있도록 했다.
----	---

문제	C62 : Fitness 측정 완료가 정상적으로 작동하지 않는다.
----	--------------------------------------

대응	finish버튼을 누르면 DB에 기록이 저장되고, 기록보기 화면으로 넘어갈 수 있도록 한다.
----	---

문제	C60 : Fitness 기능 일시정지 기능이 간헐적으로 작동하지 않는다.
----	---

대응	일시정지를 했을 때, 일시정지한 시각의 운동량을 화면에 보여줄 수 있도록 한다.
----	--

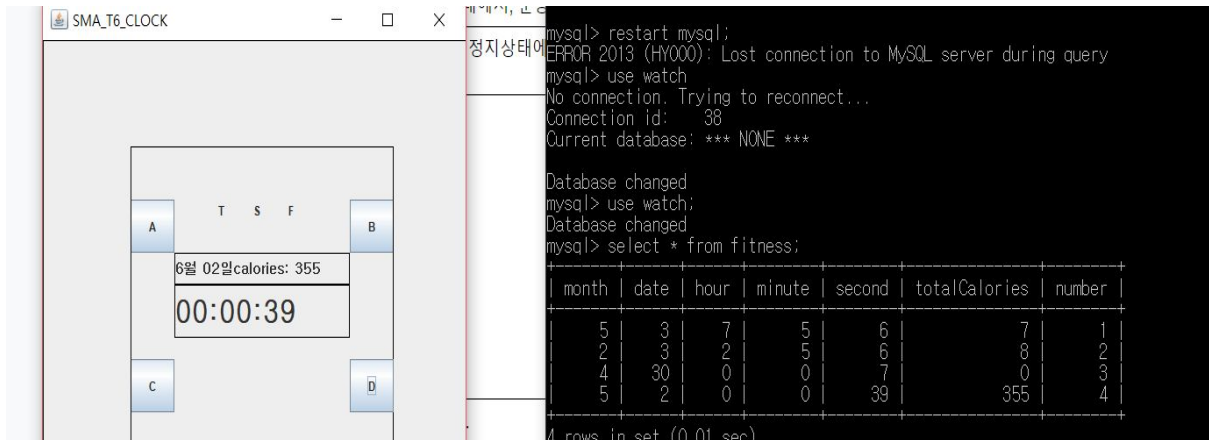
문제	P2 : Fitness 기능이 정지 상태에서, 운동량 측정을 종료할 수 없다.
----	---

대응	DB연결문제 - 시나리오대로 정지상태에서만 운동량측정을 종료할 수 있도록 수정한다.
----	--

문제	P23 : Fitness 측정 상태에서, 리스트를 확인할 수 없다.
----	---------------------------------------

대응	시나리오 상, 운동 측정 시에는 화면이 전환될 수 없다.(리스트 확인 못한다.)
----	--

-수정 화면



일시정지와 재실행을 했을 때 오류가 없도록 수정함.

운동을 끝냈을 때, 같은 날짜가 존재하면 해당레코드에 update하는 문제 수정.  
 (calendar클래스의 month가 0부터 시작하기 때문에 DB에 month에 값이 -1 되어 들어감. 실행의 문제는 없음)

운동량 측정이 잘 되는지 확인하기 위해 임시로 초 단위로 calculate\_calories()함수를 호출하였다.

## 7. SelectFunction

문제	C71 : 선택된 기능의 수가 6개이다.
대응	초깃값 으로 TimeKeeping, Timer, StopWatch 그리고 Fitness까지 총 4개가 선택되도록 한다.

문제	C72 : 기능 선택을 3가지로 하지 않아도 된다.
대응	대응중