2019 Software Modeling And Analysis

Check it out

– stage 2050&2060 –

Team 6.
201613856 소아이린
201711381 김소현
201711401 염혜지
201711420 임수연
201711428 조은지

01 발표 contents

Digital Watch Manual

Layered Architecture

Methods Description

Implement Class

JUnit Testing Plan

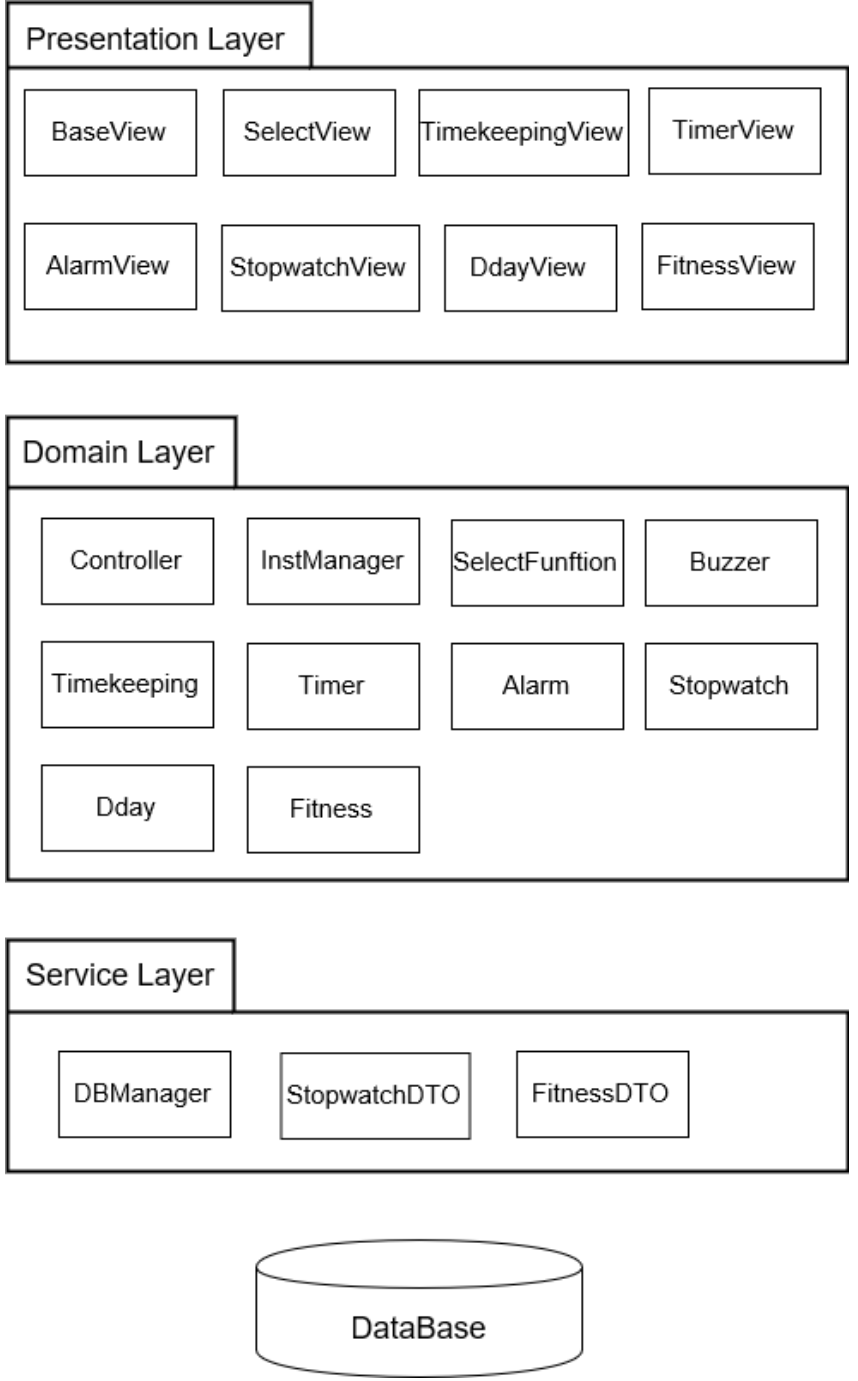
Write Unit Test Code

02 Digital Watch Manual

Clock-2. TimeKeeping 화면(초기화면)

Mode버튼을 통해 다음 기능을 사용 할 수 있다.
Setting버튼을 통해서 시간을 원하는 시간으로 설정할 수 있다.
None버튼은 기능이 없는 버튼으로써 버튼을 눌러도 아무 기능이 작동하지 않는다.

03 Layered Architecture



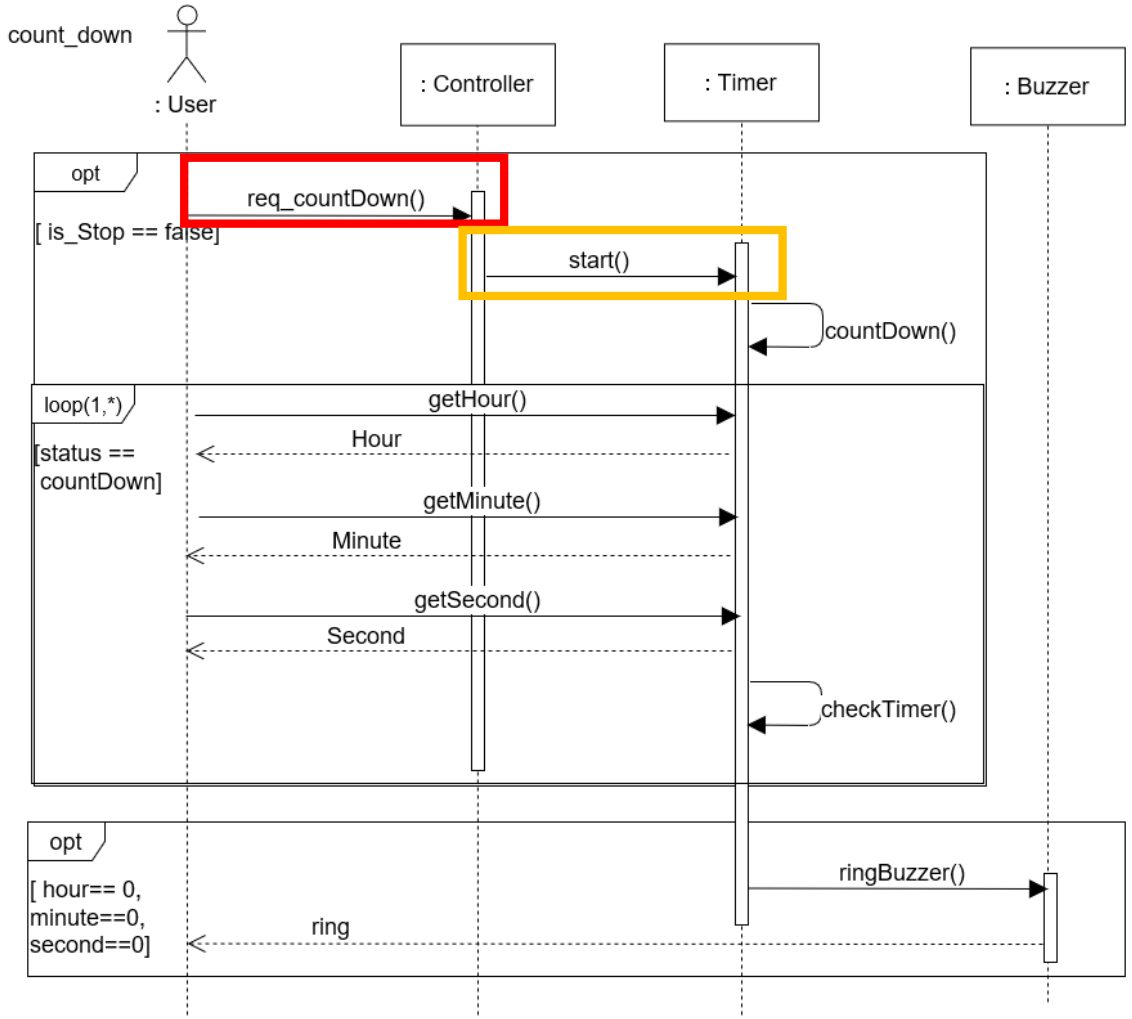
04 Methods Description

checkAlarm Method

Type	Method
Name	check Alarm
Purpose	현재시간이 알람시간과 일치하는지 체크하고 일치한다면 버저를 울린다.
Cross Reference	System function : 6.3 User cases : check_alarm
Input	
Output	
Abstract operation	설정된 요일, 시간, 분을 현재 요일, 시간 분과 비교하고 값이 같다면 설정된 주기 간격으로 버저를 3번 울린다.
Exceptional Course of Event	N/A

05 Implement Class

Count_Down Use Case



Controller Class

```
public void req_countDown(){
        timer.setIs_stop(false);
        timer.start();

    }
```

Timer Class

```
public void run(){
        //checkTimer()은 countDown()안에 존재한다.
        countDown();
    }
```

05 Implement Class – Count Down Use Case

Timer Class

```
1  //usecase: count_down
2     synchronized public void countDown() {
3         outerLoop:
4         while (true) {
5             if(is_stop == false) {
6                 this.second--;
7                 if (this.second == 0 && this.minute != 0) {
8                     this.second = 60;
9                     this.minute--;
10                } else if (this.second == 0 && this.minute == 0 && this.hour != 0) {
11                    this.minute = 60;
12                    this.hour--;
13                } else {
14
15                    check = checkTimer();
16                    if (check == true) {
17                        buzzer.setIs_stop(false);
18                        buzzer.ringBuzzer();
19                    }
20
21                }
22                try {
23                    Thread.sleep(1000);
24                } catch (InterruptedException e) {
25                    // TODO Auto-generated catch block
26                    e.printStackTrace();
27                }
28            }
29            //is_stop == true
30            else{
31                synchronized (this) {
32                    try {
33                        //System.out.println("wait");
34                        this.wait();
35                    } catch (InterruptedException e) {
36                        // TODO Auto-generated catch block
37                        e.printStackTrace();
38                    }
```

```
//usecase: count_down
public boolean checkTimer(){
    if(this.hour==0 && this.minute==0 && this.second==0){
        return true;
    }
    else
        return false;
}
```

Buzzer Class

```
public void ringBuzzer() {
        while(time<30){
            if(is_stop == false) {
                beep();
                this.time++;
                try {
                    Thread.sleep(1000);
                } catch (InterruptedException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
                }

            }
            //is_stop == true
            else{
                break;
            }
        }
    }
```

06

JUnit Testing Plan

Test	Test항목	Description	Use Case Number & Names	Ref.#
1	버저작동시험	버저가 울려야 할 때 작동이 잘 되는지 확인	1. ring_buzzer	R.1.1.1
2	버저중지시험	버저가 울린 뒤 입력을 받았을 때 중지가 잘 되는지 확인	2. stop_buzzer	R.1.1.2
3	시간측정확인시험	초 단위로 시간이 잘 측정되고 있는지 확인	3. count_up	R.1.2
4	기능전환시험	알맞은 기능으로 전환되었는지 확인	4. change_function	R.2.1
5	첫화면확인시험	첫번째 화면에서만 조합변경기능이 가능한지 확인	5. check_first_display	R.3.1
6	초기기능설정확인시험	모든 기능이 기본상태일때만 조합변경기능이 가능한지 확인	6. check_default_display	R.3.2
7	목록화면확인시험	기능 목록이 화면에 나오는지 확인	7. look_function	R.3.3.1
8	목록선택시험	기능을 선택했을 때 개수가 4개 이하 인지 확인하고, 실행조합목록에 넣어졌는지 확인	8. select_function	R.3.3.2
9	시간 화면 확인 시험	화면에 날짜와 시간이 나오는지 확인	9. look_time	R.4.1
10	시간 설정 확인 시험	날짜와 시간이 설정한 대로 변경되는지 확인	10. set_time	R.4.2
11	타이머설정시험	타이머가 설정되는지 확인	11. set_timer	R.5.1
12	카운트다운시험	타이머 시작 시에 카운트다운이 되는지 확인	12. count_down	R.5.2
13	타이머일시정지시험	입력을 받았을 때 카운트다운이 멈추는지 확인	13. pause_timer	R.5.3
14	타이머 취소기능 시험	카운트다운이 멈추고, 시간이 00시00분00초로 잘 초기화 되었는지 확인	14. cancel_timer	R.5.4
15	알람화면확인시험	알람 목록이 화면에 잘 보이는지 확인	15. look_alarm	R.6.1
16	알람설정확인시험	알람의 요일/시간/반복/주기 설정이 저장되었는지 확인	16. set_alarm	R.6.2
17	알람체크확인시험	현재 요일, 시각과 설정한 알람 요일, 시각을 잘 비교하고 있는지 확인	17. check_alarm	R.6.3
18	알람실행여부시험	알람이 제대로 끄고 켜지는지 확인	18. onoff_alarm	R.6.4
19	알람삭제확인시험	알람 목록이 제대로 삭제되는지 확인	19. delete_alarm	R.6.5
20	시간기록확인시험	기록입력이 들어왔을 때의 시각이 목록에 잘 저장되었는지 확인	20. record_stopwatch	R.7.1.1
21	스톱워치 일시중지 확인 시험	스톱워치의 카운트업이 멈추는지 확인	21. pause_stopwatch	R.7.2.1
22	기록화면확인시험	목록에서 기록한 시간들을 잘 보여주는지 확인	22. look_record	R.7.2.2
23	스톱워치 초기화 확인 시험	카운트업을 멈추고 0초로 초기화한 후, 목록의 시간 기록들을 모두 지웠는지 확인	23. reset_stopswatch	R.7.2.3
24	날짜선택시험	D+day의 날짜를 화면에 띄워서 선택할 수 있는지 확인	24. select_date	R.8.1
25	목표 선택화면 확인 시험	화면에 6가지 목표목록을 띄워서 선택할 수 있는지	25. select_goal	R.8.2

06 JUnit Testing Plan

		확인		
26	디데이 갱신 시험	하루가 지날 때마다 D+day값을 갱신하는지 확인	26. update_Dday	R.8.3
27	디데이 화면 확인 시험	목표의 종류와 D+day 값을 화면에 보여주는지 확인, D+day 값을 오름차순으로 정렬하여 화면에 보여주는지 확인	27. look_Dday	R.8.4
28	디데이목록제거시험	D+day 목록에서 하나씩 목표를 삭제할 수 있는지 확인	28. delete_Dday	R.8.5
29	운동량 화면 확인 시험	화면에 하루치 총 소모 칼로리량과 운동 시간 을 보여주는지 확인	29. look_exercise	R.9.1
30	운동선택시험	3개의 유산소 운동 종목 중 하나를 선택할 수 있는지 확인	30. select_exercise	R.9.2
31	칼로리계산시험	총 칼로리 소모량(해당 운동의 1분당 소모 칼로리량 * 운동시간)을 알맞게 계산하는지 확인, 화면에 실시간으로 보여주는 지 확인	31.calculate_calories	R.9.3
32	소모 칼로리 갱신 시험	하루 총 소모 칼로리를 갱신하는지 확인	32. update_calories	R.9.4
33	운동 일시정지 시험	운동 시간 측정을 일시정지하는지 확인	33. pause_exercise	R.9.5.1
34	운동 완료 확인 시험	운동 시간 측정을 완료하는지 확인	34. finish_exercise	R.9.5.2

07 Write Unit Test Code

Timekeeping Test Code

```java
@BeforeAll
public static void makeInstance(){
    try {
        junitTest = new Timekeeping();
    } catch (Exception var1) {
        System.out.println("error");
    }
}


@Test
void getDate() throws Exception {
    try {
        Assertions.assertEquals(26, junitTest.getDate());
    } catch (Exception var2) {
        System.out.println("error: expect값이 현재 '일'과 동일하지 않습니다");
    }
}
```

```java
@Test
void getDayNum()  throws Exception {
    try {
        Assertions.assertEquals(1, junitTest.getDayNum());
    } catch (Exception var2) {
        System.out.println("error: expect값이 현재 '요일'과 동일하지 않습니다.");
    }
}

@Test
void getIs_stop() throws Exception{
    try{
        junitTest.getIs_stop();
    }catch (Exception var2){
        System.out.println("error");
    }
}
```

07 Write Unit Test Code

```java
@Test
void setIs_stop() throws Exception {
    try{
        junitTest.setIs_stop(false);
    }catch (Exception var2){
        System.out.println("error");
    }
}

@Ignore
void countUp()  throws Exception {
    try {
        junitTest.countUp();
    } catch (Exception var2) {
        System.out.println("error: 테스트가 무한히 진행되므로 Ignore처리해줍니다");
    }
}
```

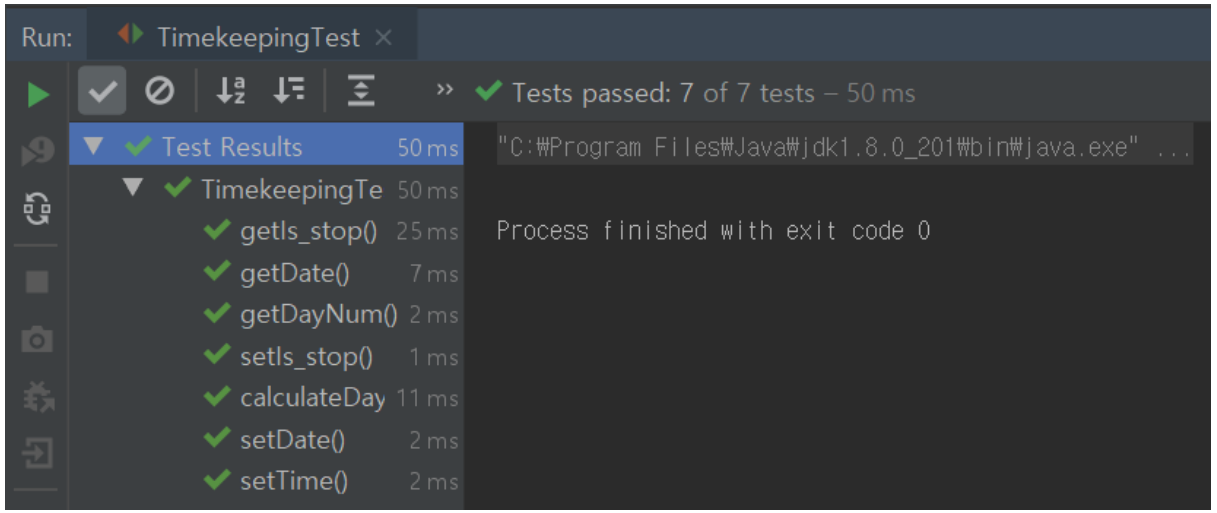
```java
@Test
void setDate() throws Exception {
    try {
        junitTest.setDate(2020, 3, 11);
    } catch (Exception var2) {
        System.out.println("error");
    }
}

@Test
void setTime() throws Exception {
    try {
        junitTest.setTime(6, 30, 30);
    } catch (Exception var2) {
        System.out.println("error");
    }
}
```

```java
@Test
void calculateDay() throws Exception {
    try {
        junitTest.calculateDay(2019, 5, 26);
    } catch (Exception var2) {
        System.out.println("error");
    }
}
```

07 Write Unit Test Code

Timekeeping Test 결과



Thank you.