

# 언제 할지 알려주시게

OOPT 3<sup>rd</sup> Cycle

Team 5

201511243 김동언  
201511262 박우진  
201511284 이종빈  
201511295 조범석

# Content

## 1. Specification Review

1. Stage 2040
2. Stage 2050

## 2. Brute Force Testing

## 3. Static Analysis

1. Bug
2. Critical

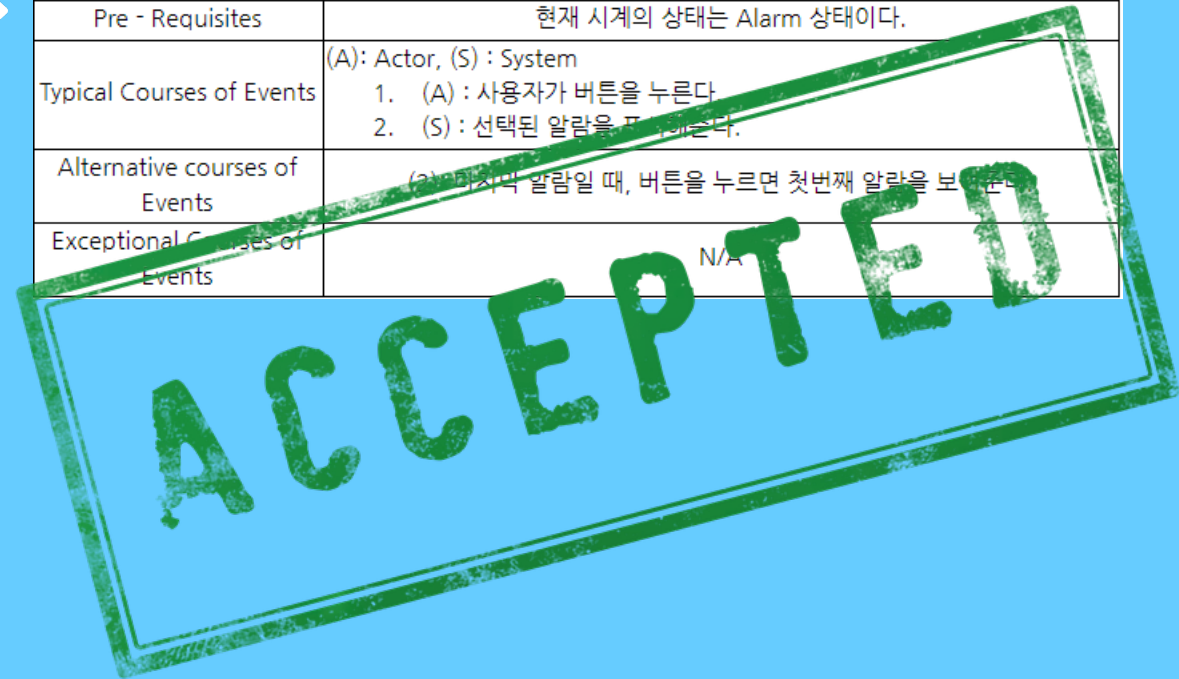
## 4. 소감

# OOPT Stage 2030\_Activity 2031

Use Case	Change alarm page
Actor	User
Purpose	설정하고자 하는 알람을 선택한다.
Overview	버튼을 통해 알람 page를 변경해준다.
Type	Evident
Cross Reference	Functions: R 4.6 Use Cases: 16
Pre - Requisites	현재 시계의 상태는 Alarm 상태이다.
Typical Courses of Events	(A): Actor, (S) : System 1. (A) : 버튼을 눌러 4개의 알람중 특정 알람을 선택한다. 2. (S) : 선택된 알람을 표시해준다.
Alternative courses of Events	(1): 마지막 알람 일 때, D버튼을 누르면 첫번째 알람을 보여준다.
Exceptional Courses of Events	N/A



Use Case	Change alarm page
Actor	User
Purpose	설정하고자 하는 알람을 선택한다.
Overview	버튼을 통해 알람 page를 변경해준다.
Type	Evident
Cross Reference	Functions: R 4.6 Use Cases: 16
Pre - Requisites	현재 시계의 상태는 Alarm 상태이다.
Typical Courses of Events	(A): Actor, (S) : System 1. (A) : 사용자가 버튼을 누른다. 2. (S) : 선택된 알람을 표시해준다.
Alternative courses of Events	(1): 마지막 알람일 때, 버튼을 누르면 첫번째 알람을 보여준다.
Exceptional Courses of Events	N/A



# OOPT Stage 2030\_Activity 2031

(2) (A) : 시작날짜를 입력하지 않고 다음 버튼을 누른다. (8)로 이동한다  
(6) (S) : 사용자가 커서를 바꾸면 (2)로 이동한다.  
(12) (S) : 사용자가 커서를 바꾸면 (8)로 이동한다.  
(14) (S) : 입력받은 종료날짜를 저장한다.  
(2) : 첫번째 년도가 99일때 증가시키면 19로 변화한다.  
(10) : 첫번째 년도가 99일때 증가시키면 19로 변화한다.  
(4)~(5) : 사용자가 '월'을 바꿀 시 '일'의 변화값이의 범위가 1일부터 '월'의 최대일로 재설정된다.  
(9)~(10) : 사용자가 '월'을 바꿀 시 '일'의 변화값이의 범위가 1일부터 '월'의 최대일로 재설정된다.



(2) (A) : 시작날짜를 입력하지 않고 다음 버튼을 누른다. (10)으로 이동한다  
->(18) (S) : 입력받은 종료날짜를 저장한다.  
(8) (S) : 사용자가 커서를 바꾸면 (2)로 이동한다.  
(16) (S) : 사용자가 커서를 바꾸면 (10)으로 이동한다.  
(2) : 첫번째 년도가 99일때 증가시키면 19로 변화한다.  
(10) : 첫번째 년도가 99일때 증가시키면 19로 변화한다.  
(5)~(6) : 사용자가 '월'을 바꿀 시 '일'의 변화값이의 범위가 1일부터 '월'의 최대일로 재설정된다.  
(13)~(14) : 사용자가 '월'을 바꿀 시 '일'의 변화값이의 범위가 1일부터 '월'의 최대일로 재설정된다.

ACCEPTED

# OOPT Stage 2040\_Activity 2041

(2)~(9), :

1. D버튼을 눌러 종료날짜 페이지로 이동한다.

(9), :

1. B버튼을 누르면 (2)으로 이동한다.

(2)~(18), :

1. C버튼을 누르면 (20)으로 이동한다.

(18):

1. A버튼을 누르면 (11)로 이동한다.

(11)~(18), :

1. D버튼을 눌러 (2)로 이동한다.

2. C버튼을 누르면 (20)으로 이동한다.

(2): 첫번째 년도가 99일때 증가시키면 19로 변화한다.

(12): 첫번째 년도가 99일때 증가시키면 19로 변화한다.

(6)~(7): 사용자가 '월'을 바꿀 시 '일'의 변화값의 범위가 1일부터 '월'의 최대일로 재설정된다.

(15)~(16): 사용자가 '월'을 바꿀 시 '일'의 변화값의 범위가 1일부터 '월'의 최대일로 재설정된다.



(2)~(9), :

1. D버튼을 눌러 종료날짜 페이지로 이동한다.

(9), :

1. A버튼을 누르면 (2)으로 이동한다.

(2)~(18) :

1. C버튼을 누르면 (20)으로 이동한다.

(18):

1. A버튼을 누르면 (11)로 이동한다.

(11)~(18) :

1. D버튼을 눌러 (2)로 이동한다.

(2): 첫번째 년도가 99일때 증가시키면 19로 변화한다.

(12): 첫번째 년도가 99일때 증가시키면 19로 변화한다.

(6)~(7): 사용자가 '월'을 바꿀 시 '일'의 변화값의 범위가 1일부터 '월'의 최대일로 재설정된다.

(15)~(16): 사용자가 '월'을 바꿀 시 '일'의 변화값의 범위가 1일부터 '월'의 최대일로 재설정된다.

ACCEPTED

# Brute Force Testing

Testing Num	Result	Description	Test Output
1-1	Pass	'년도'의 값을 99보다 크게 설정한다	연도 1 뿐만 아니라 모든 값이 이상한 값으로 바뀌고 원하는 시간 변경이 불가능해진다. (이후 선택된 값이 아닌 다른 값이 같이 바뀜)
1-2	Fail	모드 설정에서 9999년 12월 31일로 설정 후, 하루가 지나면 각 숫자가 값들이 한 칸씩 밀린다.	시계의 각 숫자 값들이 한 칸씩 밀려나 정상적인 시간 값을 보여주지 못함.
2-1	Pass	타이머 설정 후 모드 변경하여 타이머를 제외하면 알람이 울리는지 확인한다.	타이머 30초 설정 후 모드 변경하여 타이머를 제외해도 30초 후에 알람이 울림
2-2	Pass	타이머가 0이 되어 알람이 울리고 멈추게 한 뒤 모든 값(시분초)이 0으로 초기화 되는지 확인한다.	타이머가 0이 되어 알람이 울리고, 알람을 멈추면 설정한 타이머 시간으로 다시 채워진다는 내용이 스펙에 나와 있지 않음.
3-1	Pass	수정모드에서 D버튼을 누르면 enabled된 알람이 disabled 상태로 변경되는지 확인한다.	수정모드에서 D버튼을 눌러도 enabled된 알람이 disabled 상태로 바뀌지 않는다. (00시00분00초 상태로 En이 활성화됨)
3-2	Pass	수정모드 접근 후 D버튼을 눌러 0으로 리셋하고 C버튼으로 저장하면 알람이 리셋 되는지 확인한다.	C버튼을 눌러 수정모드 접근 후 D버튼을 누르면 0으로 리셋 된다. (스펙과 다름)
4-1	Pass	'년도 1'값을 99이상으로 설정한다.	연도 1 뿐만아니라 모든 값이 이상한 값으로 바뀌고 원하는 D-day 변경이 불가능해진다. (이후 선택된 값이 아닌 다른 값이 같이 바뀜)
4-2	Pass	종료일을 시작일보다 이전으로 설정한다.	종료일이 시작일 보다 이전으로 선택 가능하며 음수 값으로 표시된다.
4-3	Pass	(종료일 - 시작일)이 9999보다 크도록 설정한다.	4자리 이상(9999이상)의 d-day는 표현이 안된다. (앞의 자리가 제외되는듯함)
4-4	Fail	시작 날짜에서 종료 날짜로 변경 시 커서가 년도 1로 위치한다.	커서가 현재 상태에 머물러있음.
4-5	Pass	시작일과 종료일이 같고, 날짜를 오늘보다 이후로 설정한 경우 d-day와 % 표기가 잘 되는지 확인한다.	d-day에서는 잔여일이 잘 나타나지만 % 에서는 'done'으로 표시된다.
4-6	Pass	종료날짜만 있으면 Remain-day 방식으로만 보여 주는지 확인한다.	종료날짜만 설정 불가능 하다.
5-1	Pass	타이머 설정 후 모드변경 하여 타이머를 제외하면 알람이 울리는지 확인한다.	타이머 30초 설정 후 모드변경하여 타이머를 제외해도 30초 후에 알람이 울린다.
5-2	Pass	수정모드 접근 후 D버튼을 눌러 0으로 리셋하고 C버튼으로 저장하면 알람이 리셋 되는지 확인한다.	C버튼을 눌러 수정모드 접근 후 D버튼을 누르면 0으로 리셋 된다. (스펙과 다름)
6-1	Pass	모드 변경 시 Beep 소리가 나는지 확인한다.	Beep 소리가 나지 않는다.



Testing Num	A/R	Description
1-2	Accepted	시간이 흘러 10000년이 되면 1970년 1월 1일로 날짜를 변경함
4-4	Accepted	D-day page를 변경하면 커서가 "년도1"로 위치하도록 변경됨.

# Static Analysis (Bug)

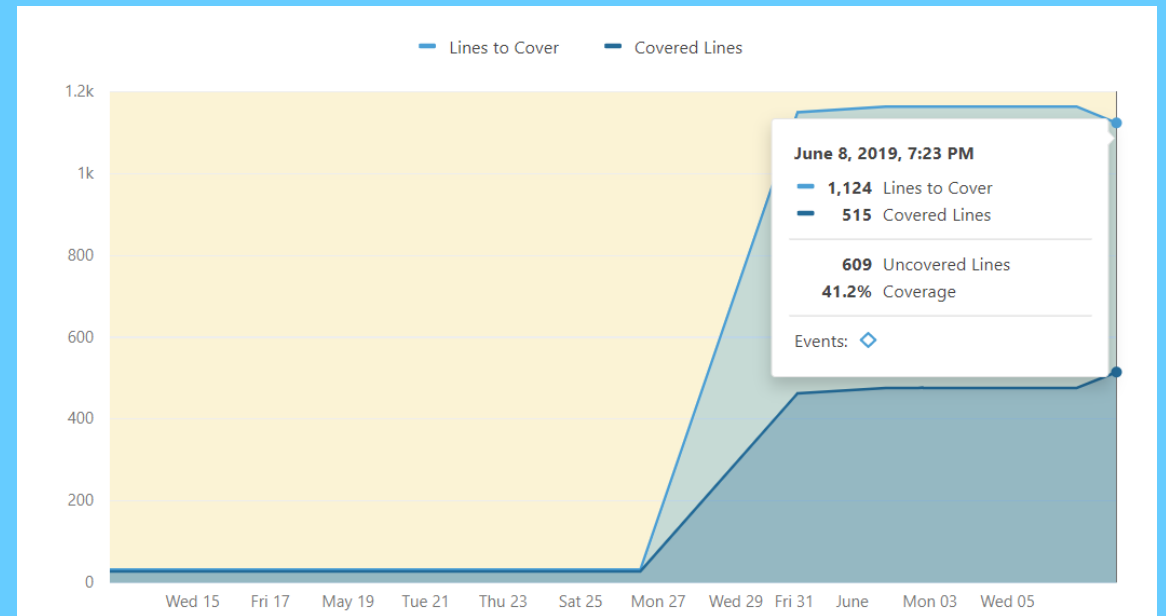
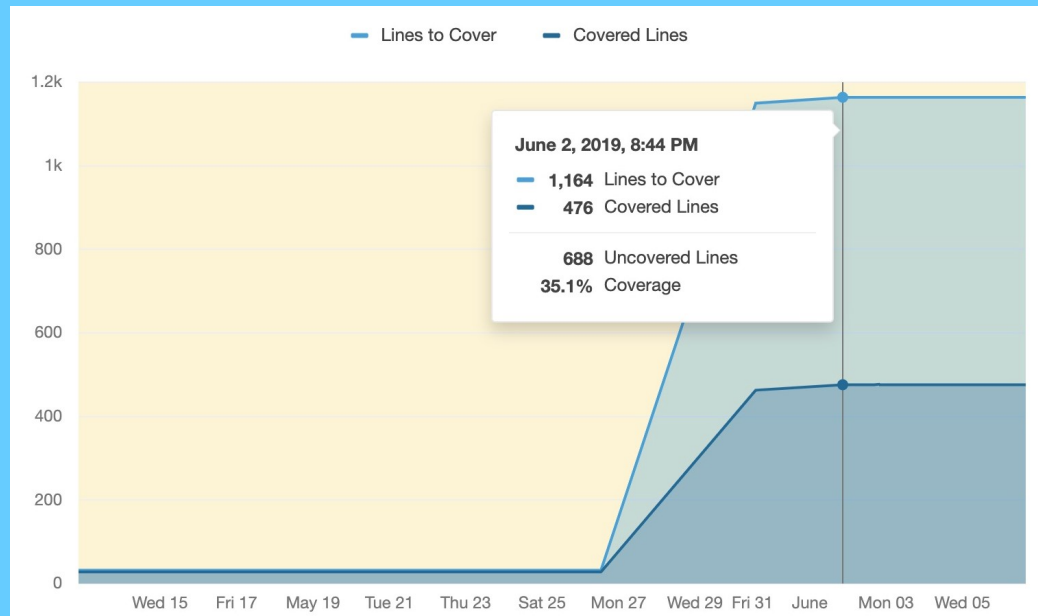
Issue & num	A/R	Title	Description
Bug 01	Accept	Bug 01 : Unread field: GUI.DigitalWatch\$Bell.isBeeping	Boolean -> boolean으로 수정
Bug 02	Accept	Bug 02 : Call the method Thread.start() to execute the content of the run() method in a dedicated thread.	run() -> start() 으로 수정.
Bug 03	Accept	Bug 03 : GUI.DigitalWatch\$Bell.play() explicitly invokes run on a thread (did you mean to start it instead?)	run() -> start() 으로 수정.
Bug 04	Accept	Bug 04 : Unread field: Logic.AlarmTime.m_timer	Timer m_timer 삭제
Bug 05	Accept	Bug 05 : Unread field: Logic.Dday.m_timer	Timer m_timer 삭제
Bug 06	Accept	Bug 06 : Unread field: Logic.IntervalTimer.m_timer	Timer m_timer 삭제
Bug 07	Accept	Bug 07 : Unread field: Logic.StopWatch.m_timer	Timer m_timer 삭제
Bug 08	Accept	Bug 08 : Unread field: Logic.TimeKeeping.m_timer	Timer m_timer 삭제
Bug 09	Accept	Bug 09 : Unread field: Logic.WatchSystem.m_timer	Timer m_timer 삭제
Bug 10	Reject	Bug 10 : Logic.WatchSystem.digitIdeal(Object) concatenates strings using + in a loop	루프가 최대 7번만 밖에 안하기 때문에 굳이 stringbuffer를 사용할 이유가 없음 -> 성능에 크게 영향을 미치지 않음
Bug 11	Accept	Bug 11 : Equality tests should not be made with floating point values.	calDday == 0 -> ==0f 로 수정
Bug 12	Accept	Bug 12 : Self assignment of field WatchTimer.m_timer in new Logic.WatchTimer()	Timer m_timer 삭제
Bug 13	Accept	Bug 13 : Uninitialized read of m_timer in new Logic.WatchTimer()	Timer m_timer 삭제

# Static Analysis (Critical)

Issue & num	A/R	Title	Description
Critical 01	Reject	Critical 01 : Refactor this method to reduce its Cognitive Complexity from 17 to the 15 allowed.	GUI Layer부분이 Thick하므로 복잡성이 높을 수 밖에 없음
Critical 02	Reject	Critical 02 : Refactor this code to not nest more than 3 if/for/while/switch/try statements.	소프트웨어 설계 과정에서 시스템 오퍼레이션이 많게 설계되었기때문에 gui의 영역이 Thick해질 수 밖에 없다. 또한 해당 Branch가 기능구현에 필요하기 때문에 해당 Issue를 Reject 하였다.
Critical 03	Reject	Critical 03 : Refactor this code to not nest more than 3 if/for/while/switch/try statements.	소프트웨어 설계 과정에서 시스템 오퍼레이션이 많게 설계되었기때문에 gui의 영역이 Thick해질 수 밖에 없다. 또한 해당 Branch가 기능구현에 필요하기 때문에 해당 Issue를 Reject 하였다.
Critical 04	Accept	Critical 04 : Avoid empty if statements	System.out.println("none"); 으로 작업 없음을 대략 명시함
Critical 05	Reject	Critical 05 : Refactor this method to reduce its Cognitive Complexity from 53 to the 15 allowed.	해당기능에 필요한 메소드이므로 복잡도를 낮추기엔 어렵다.
Critical 06	Reject	Critical 06 : The Cyclomatic Complexity of this method "buttonB" is 13 which is greater than 10 authorized.	GUI와 관련된 메서드이기 때문에 두꺼운 메서드가 되었음
Critical 07	Reject	Critical 07 : Refactor this code to not nest more than 3 if/for/while/switch/try statements.	소프트웨어 설계 과정에서 시스템 오퍼레이션이 많게 설계되었기때문에 gui의 영역이 Thick해질 수 밖에 없다. 또한 해당 Branch가 기능구현에 필요하기 때문에 해당 Issue를 Reject 하였다.
Critical 08	Reject	Critical 08 : Refactor this code to not nest more than 3 if/for/while/switch/try statements.	소프트웨어 설계 과정에서 시스템 오퍼레이션이 많게 설계되었기때문에 gui의 영역이 Thick해질 수 밖에 없다. 또한 해당 Branch가 기능구현에 필요하기 때문에 해당 Issue를 Reject 하였다.



# Code Coverage



# 소감

OOPT 2050, 2060
OOPT 2040
OOPT 2030
OOPT 1000
OOPT 3rd Cycle
OOPT 2nd Cycle

Report (18)

UseCase
StateChart
Sequence Diagram
InteractionDiagram
GUI포함 Interaction
Domain
ClassDiagram

Diagram

Timer.png
TimeKeeping.png
Stopwatch.png
Overall.png
ModeChange.png
IntervalTimer.png
Dday.png
Alarm.png
시계 UI .pptx
시계 UI .pptx
6_IntervalTimer.PNG
5_Dday.PNG
4_Alarm.PNG
3_Stopwatch.PNG
2_Timer.PNG
1_TimeKeeping.PNG



Stage 1000  
V4



Stage 2030  
V6



Stage 2040  
V4



Stage  
2050/2060  
V2



2<sup>nd</sup> Cycle  
V1



3<sup>rd</sup> Cycle  
V1





“참 힘들고... 긴 시간이었지...”

**Project Plan**

**Use Case** (Brief to Fully Dress)

**Diagram**

(Interaction to Class Diagram)

**Implementation**

**Unit Test & Static Analysis**

(with 소검)