

43-Check It Out - Digital Watch

**2019 Software Modeling & Analysis
OOPT Stage 2040**

T6

201613856 소아이린

201711381 김소현

201711401 염혜지

201711420 임수연

201711428 조은지

Contents

1. Activity 2141. Design Real Use Cases
2. Activity 2142. Define Reports,UI,and Storyboards
3. Activity 2143. Refine System Architecture
4. Activity 2144. Define Interaction Diagram
5. Activity 2145. Define Design Class Diagram
6. Activity 2146. Analyze Traceability Analysis

1. Activity 2141. Design Real Use Cases

- Common function

Use Case	ring_buzzer
Actor	system
Purpose	버저를 울린다.
Overview	특정 조건이 충족되면 시스템은 버저를 울린다.
Type	Hidden
Cross Reference	Functions: R.1.1.2, R.5.2, R.6.3 Use Cases: stop_buzzer, count_down, check_alarm
Pre-Requisites	타이머에서 카운트다운이 0시 0분 0초에 도달해야 한다. 알람의 설정 요일/설정 시간이 현재 요일/현재 시간과 일치해야한다.
UI Widgets	none
Typical Courses of Events	(A):Actor, (S):System 타이머 1.(S) 타이머 기능에서, 카운트다운을 통해 시간이 0시 0분 0초에 도달하면 버저를 울린다. 알람 (A):Actor, (S):System 1.(S) 알람 기능에서, 설정 요일/설정 시간이 현재 요일/현재 시간과 일치하면 버저를 울린다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A
Use Case	stop_buzzer
Actor	user, system
Purpose	울리는 버저를 멈춘다.
Overview	사용자가 버튼을 이용해서 버저를 멈춘다.
Type	Evident
Cross Reference	Functions: R.1.1.1 Use Cases: ring_buzzer
Pre-Requisites	버저가 울려야 한다.

UI Widgets	clock-20
	(A):Actor, (S):System (User가 중지 버튼을 누른 경우) 1.(A) User가 중지 버튼을 누른다. 2.(S) 버저를 멈춘다.
Typical Courses of Events	(User가 중지 버튼을 누르지 않은 경우) 1.(A) User가 중지 버튼을 누르지 않는다. 2.(S) 버저가 울린 지 30초 뒤에, 버저를 멈춘다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A
Use Case	count_up
Actor	user/system
Purpose	시작 시간부터 초 단위로 카운트업한다.
Overview	사용자는 카운트업 되는 시간을 실시간으로 화면을 통해 본다. 시스템은 시간보기 기능, 스톱워치 기능, 운동시간 측정 기능에서 초 단위로 시간을 카운트업한다.
Type	Evident
Cross Reference	Functions : R.4.1, R.7.2.1, R.9.1, R.9.5.1 / Use Cases : show_time, pause_stopwatch, show_exercise, pause_exercise
Pre-Requisites	시간보기, 스톱워치, 운동량 측정 기능에서만 실행된다.
UI Widgets	clock-2,clock-3,clock-18
	(A):Actor, (S):System 시간보기 기능 1.(A) User가 시간보기 기능을 사용한다. 2.(S) 현재 시간을 가져와 세그먼트 화면에 표시한다. 3.(S) 1초 단위로 현재 시간을 카운트업 해서, 현재 시간을 갱신한다. 4.(S) 갱신한 현재 시간을 세그먼트 화면에 보여준다.
Typical Courses of Event	스탑워치, 운동량 측정 기능 1. (A) User가 모든 설정을 끝낸 뒤, 실행 버튼을 누른다. 2. (S) countUp 화면으로 전환 하고, 카운트 업 할 시작시간을 가져온다. 3. (S) 1초 단위로 시작시간을 카운트업 해서, 시간을 갱신한다. 4. (S) 갱신한 시간을 세그먼트 화면에 보여준다.
Alternative Courses of Event	N/A

Exceptional Courses of Event	N/A
Use Case	change_fuction
Actor	user/system
Purpose	시계의 4가지 기능 중, 다음 순서의 기능으로 바꿔준다.
Overview	User는 기능을 변경한다. 시스템은 User가 사용할 기능을 다음 순서의 기능으로 변경해준다.
Type	Evident
Cross Reference	Functions :R4.1, R.5.1, R.6.1, R.1.3,R.8.4, R.9.1 / Use Cases : look_time, set_timer, look_alarm, count_up, look_Dday, look_exercise
Pre-Requisites	N/A
UI Widgets	clock-1...clock-20
Typical Courses of Event	(A):Actor, (S):System 1.(A) User가 Mode 버튼을 누른다. 2-1.(S) 현재 기능을 그대로 실행하면서, 다음 실행해야하는 기능으로 전환한다. (이전에 저장된 상태 그대로 가져와서 바꾼다.) 2-2.(S) User가 4번째 기능에서 또 Mode 버튼을 누르면, 다시 처음 기능으로 바꿔준다.
Alternative Courses of Event	N/A
Exceptional Courses of Event	N/A
Use Case	check_first_display
Actor	system
Purpose	사용자가 조합변경 기능을 요청할 때, 현재 기능이 시간보기 기능인지 확인한다.
Overview	사용자가 조합변경 기능을 요청할 때, 시스템은 현재 기능이 시간보기 기능인지를 확인한다. 시간보기 기능일 때만, 조합변경기능을 사용할 수 있도록 한다.
Type	Hidden
Cross Reference	Functions : R.3.2 / Use Cases : check_default_display
Pre-Requisites	조합변경기능 요청이 들어와야 실행된다.
UI Widgets	none

Typical Courses of Event	(A):Actor, (S):System 1.(S) 시스템은 현재화면이 시간보기 화면인지 확인한다. 2-1. (S) 현재화면이 시간보기 화면이라면 check_default_display 함수를 실행한다. 2-2. (S) 현재화면이 시간보기 화면이 아니라면 check_default_display 함수를 실행하지 않는다.
Alternative Courses of Event	N/A
Exceptional Courses of Event	N/A
Use Case	check_default_display
Actor	system
Purpose	사용자가 조합변경기능을 요청할 때, 현재 실행되는 4가지 기능들이 모두 기본 상태인지 확인한다.
Overview	시스템은 현재 실행되는 4가지 기능들이 모두 기본 상태인지 확인한다. 모든 기능들이 기본 상태라면, 조합변경기능을 사용할 수 있도록 한다.
Type	Hidden
Cross Reference	Functions : R.1.2, R.3.3.1,R.4.2, R.5.2, R.6.2, R.8.1, R.8.2, R.9.2 / Use Cases : count_up, look_function, set_time, count_down, set_alarm, select_date, select_goal, select_exercise
Pre-Requisites	현재 화면이 시간보기 화면이어야 실행된다.
UI Widgets	none
Typical Courses of Event	(A):Actor, (S):System 1.(S) 4가지 기능이 모두 초기 상태인지 확인한다. 2-1.(S) 4가지 기능 모두 초기 상태라면, look_function을 실행한다. 2-2.(S) 4가지 기능 중 하나라도 초기 상태가 아니라면, look_function을 실행하지 않는다.
Alternative Courses of Event	N/A
Exceptional Courses of Event	N/A
Use Case	look_function
Actor	user
Purpose	화면에서 시간보기 기능을 제외한 5가지 기능 목록을 하나씩 본다.

Overview	사용자는 버튼을 통해 시간보기 기능을 제외한 5가지 기능 목록을 화면에서 하나씩 본다. 5번째 기능을 본 후 다시 첫번째 기능을 본다.
Type	Evident
Cross Reference	Functions : R.3.3.2 / Use Cases : select_function
Pre-Requisites	모든 기능이 기본 상태여야 한다.
UI Widgets	clock-1
Typical Courses of Event	(A):Actor, (S):System 1. (A) User가 기능 목록 중 하나를 본다. 2. (A) User가 다음 버튼을 누른다. 3.(S) 시스템은 다음 순서에 맞는 기능 목록을 화면에 제공한다.
Alternative Courses of Event	N/A
Exceptional Courses of Event	N/A
Use Case	select_function
Actor	user/system
Purpose	사용할 3가지 기능을 선택하여 실행목록에 넣는다.
Overview	사용자는 기능을 하나씩 선택한다. 시스템은 사용자가 선택한 기능의 개수를 센다. 3개까지만 선택되도록 한다. 시스템은 선택된 기능을 실행목록에 넣는다.
Type	Evident
Cross Reference	Functions :R.3.1,R.3.2 / Use Cases : look_function,check_first_display
Pre-Requisites	현재 화면이 기능목록을 보여주는 화면이어야 한다. 실행 목록에 시간보기 기능은 항상 넣어져 있어야 한다.
UI Widgets	clock-1
Typical Courses of Event	(A):Actor, (S):System 1. (A) User가 첫번째 기능을 확인한다. 2. (A) User가 Next 버튼을 눌러 다음 기능을 확인한다. 3-1. (A) User가 OK버튼을 눌러 조합에 추가하기를 원하는 기능을 선택한다. 3-2. (A) User가 선택을 취소하기를 원하는 기능은 다시 OK버튼을 눌러 선택을 취소한다. 4. (S) 선택된 기능을 실행확인배열에 true로. 선택을 취소한 기능은 false로 설정한다. 5. (S) 선택된 기능을 LCD 화면에 표시한다. 6. (A) User가 Finish 버튼을 눌러 선택을 완료한다. 7-1. (S) 선택된 기능의 개수가 3개일 경우 시간보기 화면으로 전환된다. 7-2. (S)선택된 기능의 개수가 3개가 아닐 경우 화면전환이 되지 않는다.

Alternative Courses of Event	N/A
Exceptional Courses of Event	E4 : 선택하지 않은 기능은 선택 취소를 할 수 없다.

- TimeKeeping function

Use Case	look_time
Actor	user
Purpose	날짜와 시간을 본다.
Overview	User는 현재 날짜와 시간을 화면에서 본다.
Type	Evident
Cross Reference	Functions : R.1.2, R.4.2,R.6.3, R.8.3,R.9.4/ Use Cases : count_up, set_time, check_alarm,update_Dday,update_calories
Pre-Requisites	N/A
UI Widgets	clock-2
Typical Courses of Event	(A): Actor, (S):System 1.(A) User가 시간보기 기능을 사용한다. 2.(S) 실시간으로 현재시간을 화면에 제공한다.
Alternative Courses of Event	N/A
Exceptional Courses of Event	N/A
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.
UI Widgets	clock-3
	<p>(A): Actor, (S):System</p> <p>1.(A) User가 시간설정 버튼을 누른다. 2.(S) 시간 설정 화면으로 전환한다..</p> <p>3.(A) User가 Up,Down버튼을 통해 '연도'를 변경한다. 4.(S) 변경되는 '연도'를 보여준다. 5.(A) User가 OK버튼을 누른다. 6.(S) '월'을 설정할 수 있도록 넘어간다.</p> <p>7.(A) User가 Up,Down버튼을 통해 '월'을 변경한다. 8.(S) 변경되는 '월'을 보여준다. 9.(A) User가 OK버튼을 누른다. 10.(S) '일'을 설정할 수 있도록 넘어간다.</p> <p>11.(A) User가 Up,Down버튼을 통해 '일'을 변경한다. 12.(S) 변경되는 '일'을 보여준다. 13.(A) User가 OK버튼을 누른다. 14.(S) 해당 '일'를 저장하고, 객체에 값을 넘겨주어 현재 연,월,일을 변경한다. 15.(S) 설정 날짜에 맞게 요일을 계산하여 저장한다.</p> <p>16.(A) User가 Up,Down버튼을 통해 '시'를 변경한다. 17.(S) 변경되는 '시'를 보여준다. 18.(A) User가 OK버튼을 누른다. 19.(S) '분'을 설정할 수 있도록 넘어간다.</p> <p>20.(A) User가 Up,Down버튼을 통해 '분'을 변경한다. 21.(S) 변경되는 '분'을 보여준다. 22.(A) User가 OK버튼을 누른다. 23.(S) '분'을 설정할 수 있도록 넘어간다.</p> <p>24.(A) User가 Up,Down버튼을 통해 '초'를 변경한다. 25.(S) 변경되는 '초'를 보여준다. 26.(A) User가 OK버튼을 누른다. 27.(S) 해당 '초'를 저장하고, 객체에 값을 넘겨주어 현재 시,분,초를 변경한다. 28.(S) 시간보기화면으로 전환한다.</p>
Typical Courses of Event	
Alternative Courses of Event	N/A
Exceptional Courses of Event	N/A

- Timer function

Use Case	set_timer
Actor	user, system
Purpose	타이머의 시간을 설정한다.
Overview	사용자가 버튼을 이용해 타이머 시간을 설정한다. 시간이 설정되면 카운트다운이 시작된다.
Type	Evident
Cross Reference	Functions: R.5.2 Use Cases: count_down
Pre-Requisites	시계의 4가지 기능 중 타이머 기능이 포함되어야 한다. 타이머는 초기상태여야한다.
UI Widgets	clock-4
Typical Courses of Events	(A):Actor, (S):System 1.(A) User가 Up,Down버튼을 통해 '시'를 변경한다. 2.(S) 변경되는 '시'를 보여준다. 3.(A) User가 OK버튼을 누른다. 4.(S) '분'을 설정할 수 있도록 넘어간다. 5.(A) User가 Up,Down버튼을 통해 '분'을 변경한다. 6.(S) 변경되는 '분'을 보여준다. 7.(A) User가 OK버튼을 누른다. 8.(S) '초'를 설정할 수 있도록 넘어간다. 9.(A) User가 Up,Down버튼을 통해 '초'를 변경한다. 10.(S) 변경되는 '초'를 보여준다. 11.(A) User가 OK버튼을 누른다. 12.(S) 해당 '초'를 저장하고, 객체에 값을 넘겨주어 시,분,초를 설정한다. 13.(S)count_down을 실행한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A
Use Case	count_down
Actor	user, system
Purpose	타이머에 설정된 시간으로부터 카운트다운을 한다.
Overview	시스템은 설정된 시간에서 0초가 될 때까지 카운트다운한다. 0초가 되면 카운트다운을 멈추고 버저를 울린다. 사용자는 울리는 버저를 멈춘다.
Type	Evident
Cross Reference	Functions: R.1.1.1, R.1.1.2, R.5.1, R.5.3, R.5.4 Use Cases: ring_buzzer, stop_buzzer, set_timer, pause_timer,

	cancel_timer
Pre-Requisites	타이머의 시간이 설정되어 있어야 한다.
UI Widgets	clock-5
Typical Courses of Events	(A):Actor, (S):System 1.(A) User가 타이머 설정 화면에서 초 단위의 시간까지 설정을 마친 후 OK버튼을 누른다. 2.(S) 타이머 설정 시간을 최종적으로 저장하고 화면을 전환한다. 3.(S) 설정된 시간에서부터 초 단위로 카운트다운한다. 4.(S) 카운트다운되는 시간을 화면에 표시한다. 5.(S) 카운트다운이 0초에 도달하면 타이머를 종료하고 시스템에 버저를 울리는 신호를 보낸다. 6.(S) 버저를 울린다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A
Use Case	pause_timer
Actor	user, system
Purpose	타이머를 일시정지 또는 재실행한다
Overview	사용자는 일시정지 버튼을 누른다. 일시정지 입력이 들어오면 시스템은 타이머의 카운트다운을 멈춘다. 일시정지상태에서 일시정지 입력이 들어오면 타이머를 재실행한다.
Type	Evident
Cross Reference	Functions: R.5.2, R.5.4 Use Cases: count_down, cancel_timer
Pre-Requisites	타이머가 카운트다운을 하는 중이어야 한다. 또는 타이머가 일시정지 상태이어야 한다.
UI Widgets	clock-5
Typical Courses of Events	(A):Actor, (S):System 1.(A) User가 Pause버튼을 누른다. 2.(S) 타이머의 카운트다운을 중지한다. 3.(S) 중지된 타이머의 시간,분,초를 저장해서 화면에 보여준다. 4.(A) User가 일시정지 상태에서 Pause버튼을 누른다. 5.(S) 타이머에 저장된 값을 시작으로 카운트다운을 재실행한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A
Use Case	cancel_timer

Actor	user
Purpose	실행 중인 타이머를 취소한다.
Overview	사용자는 타이머를 취소한다. 시스템은 카운트다운 실행을 멈추고 설정시간을 0으로 초기화한다.
Type	Evident
Cross Reference	Functions: R.5.1, R.5.2 Use Cases: set_timer. count_down
Pre-Requisites	타이머는 실행 중이거나 일시정지 상태여야 한다.
UI Widgets	clock-5
Typical Courses of Events	(A):Actor, (S):System 1.(A) User는 Reset 버튼을 누른다. 2.(S) 카운트다운을 멈춘다. 3.(S) 설정 시간을 0초로 초기화하고 화면을 타이머 설정화면으로 전환한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A

- Alarm function

Use Case	look_alarm
Actor	user
Purpose	사용자가 목록에 저장된 알람을 확인한다.
Overview	사용자는 알람 목록을 확인한다. 버튼을 이용해서 그 다음 알람 목록을 확인한다. 알람 목록은 최대 4개까지 볼 수 있다.
Cross Reference	Functions: R.6.2, R.6.4, R.6.5 Use Cases: set_alarm, onoff_alarm, delete_alarm
Type	Evident
Pre-Requisites	N/A
UI Widgets	clock-6
Typical Courses of Events	(A):Actor, (S):System 1.(A) User가 첫번째 알람을 확인한다. 2.(A) User가 Next 버튼을 누른다. 3-1.(S) 순차적으로 객체배열에 저장된 알람을 세그먼트 화면에 표시한다. 해당 알람이 설정된 요일을 도트화면에 밀줄로 표시한다. 3-2.(S) 배열의 마지막 인덱스에 저장된 알람을 보여준 경우, 배열의 첫번째 인덱스에 저장된 알람을 보여준다. 3-3.(S) 배열이 비어있으면 빈 화면을 보여준다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A
Use Case	set_alarm
Actor	user
Purpose	알람을 설정한다
Overview	사용자는 버튼을 이용해 알람의 요일/주기/시간/분을 설정한다.
Type	Evident
Cross Reference	Functions: R.6.1, R.6.3 Use Cases: look_alarm, check_alarm
Pre-Requisites	알람 목록의 개수가 3개 이하여야 한다.
UI Widgets	clock-8

Typical Courses of Events	(A):Actor, (S):System 1.(A) User가 알람 목록화면에서 Add alarm버튼을 통해 알람 추가를 요청한다. 2.(S) 도트화면에는 요일과 주기를, 세그먼트 화면에는 00:00을 보여준다. 3.(A) User가 Next,Prev,OK버튼을 통해 알람시간(요일,시간,주기)을 설정한다. 4.(A) User가 세그먼트 화면의 초 단위에서 OK버튼을 눌러 알람 설정 종료요청을 한다. 5.(S) 알람의 설정 요일,시간,주기를 저장한다 6.(S) 알람 목록화면으로 전환한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A
Use Case	check_alarm
Actor	system
Purpose	설정된 알람과 현재 시각을 비교한다.
Overview	시스템은 설정된 알람의 요일과 시간 값을 가져온다. 현재의 요일과 시간 값을 가져온다. 현재 요일/시간이 설정 요일/시간과 같으면 버저를 울린다. 설정된 알람의 주기에 맞춰 버저를 울린다.
Type	Hidden
Cross Reference	Functions: R.1.1.1, R.4.1, R.6.2, R.6.4 Use Cases: ring_buzzer, look_time, set_alarm, onoff_alarm
Pre-Requisites	시계의 4가지 기능 중 알람 기능이 포함되어 있어야 한다. 설정된 알람이 존재해야 한다.
UI Widgets	none
Typical Courses of Events	(A):Actor, (S):System 1.(S) 설정된 알람의 요일/시간/주기설정을 가져온다. 2.(S) 두 요일과 시간이 매분마다 같은지 확인한다. 3.(S) 시간이 같으면, 알람의 주기설정을 확인한다. 4-1.(S) 주기 설정이 되어 있지 않으면, ring_buzzer을 한번 실행한다. 4-2.(S) 주기 설정이 되어 있으면, 설정된 주기 간격으로 ring_buzzer을 3번 실행한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A
Use Case	delete_alarm
Actor	user
Purpose	버튼을 이용해 알람을 삭제한다.

Overview	버튼을 이용해 목록에 저장된 알람 하나를 삭제한다.
Type	Evident
Cross Reference	Functions: R.6.1 Use Cases: look_alarm
Pre-Requisites	현재 화면이 알람 목록 보기 화면이어야 한다. 목록에 저장된 알람이 최소 1개 이상이어야 한다.
UI Widgets	clock-7
Typical Courses of Events	(A):Actor, (S):System 1.(A) User는 알람 목록 보기 화면에서 Setting버튼을 누른다. 2.(A) User는 Next버튼을 통해 도트화면에서 Del를 선택한다. 3.(A) User는 OK버튼을 눌러 알람 설정을 저장한다. 4.(S) 시스템은 알람목록리스트에서 해당 알람이 저장된 인덱스를 삭제한다. 5.(S) 다음 인덱스에 저장된 알람을 화면에 표시한다.
Alternative Courses of Events	N/A
Exceptional Courses of Events	N/A
Use Case	onoff_alarm
Actor	user
Purpose	알람의 실행여부를 설정한다.
Overview	사용자는 버튼을 통해 해당 알람을 비활성화한다. 사용자는 버튼을 통해 비활성화된 알람을 다시 활성화한다.
Type	Evident
Cross Reference	Functions: R.6.1, R.6.3 Use Cases: look_alarm, check_alarm
Pre-Requisites	목록에 저장된 알람이 적어도 1개 이상이어야 한다.
UI Widgets	clock-7
Typical Courses of Events	(A):Actor, (S):System 1.(A) User는 알람 목록 보기 화면에서 Setting버튼을 누른다. 2.(A) User는 Next버튼을 통해서 도트화면에서 알람의 on을 선택할 수 있다. 3.(A) User는 OK버튼을 누른다. 4.(S) 현재 알람을 on 상태로 저장한다.
Alternative Courses of Events	2.(A) User는 Next버튼을 통해 도트화면에서 알람 off를 선택할 수 있다. 3.(A) User는 OK버튼을 누른다. 4.(S) 현재 알람을 off 상태로 저장한다.
Exceptional Courses of Events	N/A

- Stopwatch function

Use Case	record_stopwatch
Actor	user/system
Purpose	실행중인 스톱워치의 현재 시간을 기록한다.
Overview	사용자는 실행중인 스톱워치 화면에서 스톱워치 시간을 기록할 수 있다. 가장 최근에 기록한 시간을 실행중인 스톱워치 화면에서 본다.
Type	Evident
Cross Reference	Functions : R.1.2, R.7.2.2, R.7.2.3 / Use Cases : count_up, look_record, reset_stopwatch
Pre-Requisites	스톱워치가 실행 중이어야 한다. (스톱워치의 시간이 CountUp 중이어야 한다.)
UI Widgets	Clock-9
Typical Courses of Event	(A):Actor, (S):System 1.(A) User가 실행중인 스톱워치 화면을 본다. 2.(A) User가 스톱워치 시간 기록 버튼을 누른다. 3.(S) DB에 저장된 스톱워치 시간 기록 개수가 10개 이하인지 확인한다. 4-1.(S) 10개 이하이면, 기존 시간 기록을 삭제하지 않는다. 4-2.(S) 10개 보다 많으면, 가장 오래된 시간 기록을 삭제한다. 5.(S) User가 시간 기록 버튼을 눌렀을 때의 스톱워치 시간을 DB에 추가한다. 6.(S) 가장 최근에 저장한 스톱워치 시간 기록을 도트 화면에 제공한다.
Alternative Courses of Event	N/A
Exceptional Courses of Event	N/A
Use Case	pause_stopwatch
Actor	user/system
Purpose	스톱워치의 실행을 일시정지한다.
Overview	사용자는 실행중인 스톱워치를 일시정지한다. 시스템은 스톱워치의 카운트업을 멈춘다. 시스템은 멈춘 스톱워치의 시간을 화면에 보여준다. 사용자는 화면을 통해 멈춘 스톱워치의 시간을 본다.
Type	Evident
Cross Reference	Functions : R.1.2, R.7.2.2, R.7.2.3 / Use Cases : count_up, look_record, reset_stopwatch
Pre-Requisites	스톱워치가 실행 중이어야 한다. (스톱워치의 시간이 카운트 업 중이어야 한다.)

UI Widgets	Clock-9
Typical Courses of Event	(A): Actor, (S):System 1.(A) User가 실행중인 스톱워치 화면을 본다. 2.(A) User가 스톱워치 Pause 버튼을 누른다. 3.(S) 스톱워치의 카운트업을 멈춘다. 4.(S) 멈췄을 때의 스톱워치 시간을 세그먼트 화면에 보여준다. 5.(A) User가 일시정지 상태에서 Pause 버튼을 한 번 더누른다. (스톱워치 재실행을 요청한다.) 6.(S)일시정지했던 시간부터 스톱워치를 재실행한다.
Alternative Courses of Event	N/A
Exceptional Courses of Event	N/A
Use Case	look_record
Actor	user/system
Purpose	DB에 저장한 모든 스톱워치 시간 기록들을 하나씩 본다.
Overview	사용자는 스톱워치 일시 정지 상태에서 기록 보기를 실행한다. 사용자는 처음에 가장 최근의 시간 기록을 본다. 버튼을 통해서 이전 시간 기록을 볼 수 있다. 볼 수 있는 시간 기록은 가장 최근 시간 기록부터 시작해서 최대 10개까지이다. 사용자가 기록 보기 화면에서 다시 한번 기록 보기를 실행하면, 스톱워치 일시정지상태로 되돌아간다.
Type	Evident
Cross Reference	Functions : R.7.1.1, R.7.1.2, R.7.2.1, R.7.2.3 / Use cases : record_stopwatch, delete_record, pause_stopwatch, reset_stopwatch
Pre-Requisites	스톱워치가 일시 정지 상태여야 한다.
UI Widgets	Clock-10, Clock-19
Typical Courses of Event	(A):Actor, (S):System 1. (A) User가 일시 정지 상태인 스톱워치 화면을 본다. 2. (A) User가 Record 버튼을 누른다. 3-1.(S) 스톱워치 기록배열에 데이터가 1개 이상 존재하면, 가장 최근의 시간 기록을 도트 화면에 제공한다. 3-2.(S) 스톱워치 기록배열에 데이터가 1개 이상 존재하지 않으면, 도트 화면에 빈 화면을 제공한다. 3.(A) User가 Next 버튼을 누른다. 4-1.(S) 이전에 저장한 시간 기록을 도트 화면에 제공한다. 4-2.(S) 이전에 저장한 시간 기록이 더이상 없으면, 다시 가장 최근의 시간 기록을 도트 화면에 제공한다. 5.(A) User가 스톱워치 기록 목록화면에서, 다시한번 Record 버튼을 누른다. 6.(S) 일시 정지 상태의 화면으로 되돌아간다.

Alternative Courses of Event	N/A
Exceptional Courses of Event	N/A
Use Case	reset_stopwatch
Actor	user/system
Purpose	일시정지 상태인 스톱워치의 시간을 0초로 초기화하고, 모든 시간 기록들을 삭제한다.
Overview	사용자는 스톱워치 일시정지 상태에서 초기화를 실행한다. 시스템은 스톱워치의 시간을 0초로 초기화한다. 시스템은 DB에 저장한 스톱워치의 모든 시간 기록을 삭제한다.
Type	Evident
Cross Reference	Functions : R.1.2,R.7.2.1, R.7.2.2 / Use Cases :count_up, pause_stopwatch, look_record
Pre-Requisites	스톱워치가 일시 정지 상태여야 한다.
UI Widgets	Clock-9
Typical Courses of Event	1.(A) User가 Reset 버튼을 누른다. 2.(S) 스톱워치의 시간을 0초로 초기화한다. 3.(S) DB에 저장한 스톱워치의 모든 시간 기록을 삭제한다. 4.(S) 초기화된 스톱워치 시간을 화면에 보여준다.
Alternative Courses of Event	N/A
Exceptional Courses of Event	N/A

- D+Day function

Use Case	select_date
Actor	user/system
Purpose	User가 선택한 D+day의 시작 날짜를 저장한다.
Overview	사용자는 D+day의 시작 날짜를 선택한다. 이전 날짜 또는 현재 날짜만 시작 날짜로 선택할 수 있고, 미래의 날짜는 선택할 수 없다. 시스템은 사용자가 선택한 시작날짜를 저장한다.
Type	Evident
Cross Reference	Functions : R.4.1, R.8.3 / Use Cases : look_time, update_Dday
Pre-Requisites	D+day의 목록이 5개 이하여야 한다. (D+day는 최대 6개까지 설정할 수 있다.) D+day 목록보기 상태여야 한다.
UI Widgets	Clock-13
Typical Courses of Event	(A):Actor, (S):System 1.(A) User가 D+Day 추가버튼을 누른다. 2.(S) D+Day 추가 화면으로 전환한다. 3.(A) User가 Up,Down버튼을 통해 '연도'를 변경한다. 4.(S) 변경되는 '연도'를 보여준다. 5.(A) User가 OK버튼을 누른다. 6.(S) 해당 '연도'를 저장하고, '월'을 설정할 수 있도록 넘어간다. 7.(A) User가 Up,Down버튼을 통해 '월'을 변경한다. 8.(S) 변경되는 '월'을 보여준다. 9.(A) User가 OK버튼을 누른다. 10.(S) 해당 '월'을 저장하고, '일'을 설정할 수 있도록 넘어간다. 11.(A) User가 Up,Down버튼을 통해 '일'을 변경한다. 12.(S) 변경되는 '일'을 보여준다. 13.(A) User가 OK버튼을 누른다. 14.(S) 해당 '일'를 저장한다. 15.(S) User가 선택한 날짜가 이전 날짜 또는 현재 날짜이면, D+day 값(day count값)을 계산한다.
Alternative Courses of Event	N/A
Exceptional Courses of Event	E15 : User가 선택한 날짜가 미래 날짜이면, User가 연도, 월, 일을 다시 입력하도록 한다.
Use Case	select_goal

Actor	user/system
Purpose	D+day의 목표를 선택해서 저장한다.
Overview	시스템은 화면에 6가지 목표 목록을 제공한다. 사용자는 6가지 목표 목록 중에 하나를 선택한다. 시스템은 사용자가 선택한 목표를 저장한다.
Type	Evident
Cross Reference	Functions : R.8.4 / Use cases : look_Dday
Pre-Requisites	D+day 시작 날짜 설정을 완료해야 한다.
UI Widgets	Clock-13
Overview	(A):Actor, (S):System 1.(A) User가 Up, Down버튼을 누른다. 2. (S) 시스템이 제공하는 6가지의 목표 목록을 차례대로 보여준다. 3.(A) User가 OK버튼을 누른다. 4.(S) 해당 목표를 저장한다. 5.(S) D+day 값(day count 값)을 하루마다 update해준다. (update_Dday를 실행한다.)
Alternative Courses of Event	N/A
Exceptional Courses of Event	N/A
Use Case	update_Dday
Actor	system
Purpose	사용자가 선택한 시작 날짜로부터 현재 날짜까지의 D+day 값을 계산하고, 하루가 지날 때마다 D+day 값을 갱신한다. 시스템은 사용자가 선택한 시작 날짜로부터 현재 날짜까지의 D+day 값을 계산한다. 하루가 지날 때마다 D+day값을 1씩 증가시켜서 갱신한다.
Type	Hidden
Cross Reference	Functions : R.4.1, R.8.1, R.8.4, R.8.5 / Use cases : look_time, select_date, look_Dday, delete_Dday
Pre-Requisites	D+day목록이 존재해야 한다.
UI Widgets	Clock-11

Typical Courses of Event	(A):Actor, (S):System 1.(S) 현재 날짜 데이터를 가져온다. 2.(S) User가 선택한 시작 날짜 데이터를 가져온다. 3.(S) User가 선택한 시작 날짜로부터 현재 날짜까지의 D+day 값(day count 값)을 계산한다. 4.(S) D+day값을 저장한다. 5.(S) 하루가 지날 때마다 D+day값(day count 값)을 1씩 증가시켜서 갱신한다. (각각의 객체마다 개별적으로 실행이 된다.)
Alternative Courses of Event	N/A
Exceptional Courses of Event	N/A
Use Case	look_Dday
Actor	user/system
Purpose	User가 저장된 모든 D+day 목록(목표와 이에 해당하는 D+day값)을 본다.
Overview	사용자는 D+day 초기 화면에서 가장 최근의 D+day 목록(목표와 이에 해당하는 D+day값)을 본다. 다음 버튼을 통해서 다음 D+day 목록을 하나씩 볼 수 있다. 볼 수 있는 목록 개수는 최대 6개까지이다.
Type	Evident
Cross Reference	Functions : R.8.2, R.8.3, R.8.5 / Use cases : select_goal, update_Dday, delete_Dday
Pre-Requisites	D+day 목록이 1개 이상 존재해야 한다.
UI Widgets	Clock-11
Typical Courses of Event	(A):Actor, (S):System 1.(A) User가 D+Day 목록 보기를 실행한다. 2-1.(S) D+day 객체 배열에서 목표와 D+Day값을 가져와 화면에 보여준다. 2-2.(S) 저장된 D+day 객체 배열이 존재하지 않으면, 아무 목록도 보여주지 않는다. 3.(A) User가 Next 버튼을 누른다. 4-1.(S) 다음 D+day 목록을 화면에 제공한다. 4-2.(S) 다음 D+day 목록이 더이상 없으면, 다시 첫번째 D+day 목록을 화면에 제공한다.
Alternative Courses of Event	N/A
Exceptional Courses of Event	N/A
Use Case	delete_Dday

Actor	user/system
Purpose	저장된 D+day 목록 중 하나를 삭제한다.
Overview	사용자는 D+day 목록 보기 화면에서 D+day 목록 중 하나를 삭제한다. 삭제한 목록은 더이상 목록 보기 화면에서 볼 수 없다.
Type	Evident
Cross Reference	Functions : R.8.4 / Use cases : look_Dday
Pre-Requisites	저장된 D+day 목록이 1개 이상 존재해야 한다.
UI Widgets	Clock-12
Typical Courses of Event	(A):Actor, (S):System 1.(A) User가 특정 D+day 목록에서 Setting 버튼을 누른다. 2.(S) Setting 화면으로 전환한다. 3.(A) User가 Delete 버튼을 누른다. 4.(S) D+Day 객체 배열에서 해당 객체를 삭제한다. 5.(S) D+Day 목록 보기 화면으로 전환한다. (다음 D+Day 목록부터 보여준다. 삭제한 D+Day 목록은 보여주지 않는다.)
Alternative Courses of Event	N/A
Exceptional Courses of Event	N/A

- Fitness function

Use Case	look_exercise
Actor	user
Purpose	하루 총 소모 칼로리와 운동시간을 본다.
Overview	User는 버튼을 통해서 30일 동안의 하루 총 소모 칼로리와 운동시간을 하나씩 본다.
Type	Evident
Cross Reference	Functions : R.9.4 / Use cases : update_calories
Pre-Requisites	Fitness실행을 한 적이 없거나 운동시간측정을 끝낸 상태여야한다.(초기화면을 볼 수 있는 상태여야 한다.)
UI Widgets	clock-14
Typical Courses of Event	(A):Actor, (S):System 1.(A) User가 버튼을 눌러 화면을 전환한다. 2.(S) DB에 저장되어 있는 운동 기록을 보여준다. 2.(A) User가 Next 버튼을 누른다. 3-1.(S) 이전날짜의 운동기록을 보여준다. 3-2.(S) 더이상 이전 운동 기록이 없는 경우, 다시 현재 운동기록을 보여준다.
Alternative Courses of Event	N/A
Exceptional Courses of Event	E1: 운동 기록이 존재 하지 않는 경우에는 이전 버튼을 실행할 수 없다.
Use Case	select_exercise
Actor	user
Purpose	시작할 운동종목을 선택한다.
Overview	User는 운동목록을 보고 , 목록 중 하나를 선택할 수 있다. 운동목록은 총 3가지 이다.
Type	Evident
Cross Reference	Functions : R.9.3 / Use Cases : calculate_exercise
Pre-Requisites	현재 상태가 look_exercise 상태여야 한다.
UI Widgets	clock-15

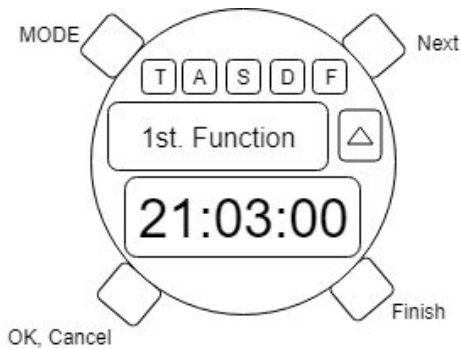
Typical Courses of Event	<p>1.(A) User가 Add_Exercise버튼을 통해 운동추가를 요청한다.</p> <p>2.(S) 운동 추가(운동 종목 설정)화면으로 전환한다.</p> <p>3.(S) 시스템에서 제공하는 3가지의 운동종목(Cycling, Walking,Running)을 하나씩 보여준다.</p> <p>3-1.(S) User가 마지막 운동 목록에서 또 한번 Next버튼을 누르면 다시 Cycling목록을 보여준다.</p> <p>4.(A) User가 Next버튼을 눌러서 하나씩 넘긴다.</p> <p>5.(A) User가 OK(Start)버튼을 눌러 운동 종목을 선택한다.</p> <p>6.(S) 해당 운동종목을 저장하고,해당운동의 1분 당 칼로리 값을 가져온다.</p> <p>7.(S) 운동시작화면으로 전환한다.</p>
Alternative Courses of Event	N/A
Exceptional Courses of Event	N/A
Use Case	calculate_calories
Actor	user/system
Purpose	실시간으로 1분 당 소모 칼로리 값과 운동한 시간을 곱하여, 총 소모 칼로리량을 계산한다.
Overview	User는 실시간으로 현재 총 소모 칼로리량을 볼 수 있다. 시스템은 해당 운동의 분 당 소모칼로리값과 운동한 시간을 곱하여 현재 총 소모 칼로리를 계산한다.
Type	Evident
Cross Reference	Functions : R.1.3, R.9.5.1, R.9.5.2 / Use Cases : count_up, pause_calories, finish_calories
Pre-Requisites	운동선택을 완료한 상태여야 한다.
UI Widgets	clock-16
Typical Courses of Event	<p>1.(A) User가 운동 종목 화면에서 OK버튼을 누른다.</p> <p>2.(S) 운동시작 화면으로 전환을 한다.</p> <p>3.(S) 운동시간을 카운트 업 한다.</p> <p>4.(S) 1분마다 (운동시간*해당 운동의 1분 당 칼로리)를 계산하여 총 소모 칼로리 양을 갱신한다.</p> <p>5.(S) 총 소모 칼로리 양과 운동 시간을 화면에 실시간으로 보여준다.</p>
Alternative Courses of Event	N/A
Exceptional Courses of Event	N/A
Use Case	update_calories
Actor	system
Purpose	하루 총 소모 칼로리를 갱신하고 기록을 저장한다.

Overview	<p>시스템은 저장된 하루 총 소모 칼로리양에 현재 소모 칼로리양을 더해서 업데이트 해준다.</p> <p>저장된 하루 총 소모 칼로리가 0이었다면 현재 운동기록 리스트의 개수가 30개 미만인지 확인한다.</p> <p>30개 미만이라면 바로 목록에 추가 해준다.</p> <p>만약 운동 기록 리스트의 개수가 이미 30개라면 가장 오래된 목록을 지우고 오늘의 하루 총 소모 칼로리양을 새로 넣어준다.</p> <p>실행을 끝내고 초기화면으로 돌아간다.</p>
Type	Hidden
Cross Reference	Functions : R.9.5.2, R.9.1 / Use cases : finish_exercise, look_exercise
Pre-Requisites	운동 측정을 완료한 상태여야 한다. 완료한 운동의 총 소모 칼로리량을 가지고 있어야 한다.
Typical Courses of Event	<ol style="list-style-type: none"> 1.(S) 총 소모 칼로리량을 가져온다. 2.(S) 이전 기록 중에 현재 날짜와 같은 기록이 있는지 확인한다. 3.(S) 만약 같은 기록이 있다면 시간, 소모칼로리를 갱신하여 다시 저장한다. 4.-1.(S) 같은 기록이 없다면, 현재 운동량 목록이 30개 미만이라면 갱신한 값을 운동기록 리스트에 바로 추가해준다. 4-2. (S) 현재 운동량 목록이 30개라면 가장 오래된 운동기록을 리스트에서 삭제하고 오늘의 운동기록을 리스트에 넣어준다. 5.(S) Fitness 목록 화면으로 전환한다.
Alternative Courses of Event	N/A
Exceptional Courses of Event	N/A
Use Case	finish_exercise
Actor	user/system
Purpose	운동시간측정을 완료한다.
Overview	시스템은 일시정지상태에서 운동시간측정을 완료한다. 시스템은 카운트업을 종료한다. 운동측정을 완전히 끝낸 후, update_calories를 실행한다.
Type	Evident
Cross Reference	Functions : R.9.4 / Use cases : update_calories
Pre-Requisites	현재상태가 일시정지 상태여야 한다.
UI Widgets	clock-14
Typical Courses of Event	<ol style="list-style-type: none"> 1.(A): User가 Finish버튼을 누른다. 2.(S): count_up 실행을 멈춘다. 3.(S): 현재 운동시간과 소모 칼로리양을 저장하고 update_calories를 실행한다. 4.(S): Fitness 목록 화면으로 전환한다.
Alternative Courses of Event	N/A

Exceptional Courses of Event	N/A
Use Case	pause_exercise
Actor	user/system
Purpose	운동 시간 측정을 일시정지한다.
Overview	User는 운동 시간 측정을 일시정지한다. 시스템은 카운트업을 멈추고, 일시정지 시각과 현재 소모 칼로리량을 보여준다. 일시정지를 한 상태에서 다시 한번 일시정지 버튼을 누르면, 카운트업을 다시 실행한다.
Type	Evident
Cross Reference	Functions : R.9.5.2 / Use cases : finish_exercise
Pre-Requisites	현재 상태가 운동시간 측정이 실행 중인 상태여야 한다.
UI Widgets	clock-16
Typical Courses of Event	(A):Actor, (S):System 1.(A): User가 Pause버튼을 누른다. 2.(S): count_up을 멈춘다. 3.(S): 현재 소모 칼로리량과 운동시간을 저장한다. 4.(S): 화면에 현재까지의 소모칼로리량과 운동시간을 보여준다. 5.(A): User가 Pause 버튼을 한번 더 누른다. 6.(S): 현재 운동시간 부터 다시 카운트업을 하고 화면에 보여준다.
Alternative Courses of Event	N/A
Exceptional Courses of Event	N/A

2. Activity 2142. Define Reports,UI,and Storyboards

Clock-1. Select Function 화면



MODE를 클릭하면 시간보기 화면으로 돌아간다.

Next 버튼을 누르면 다음 기능을 도트화면에 보여준다.

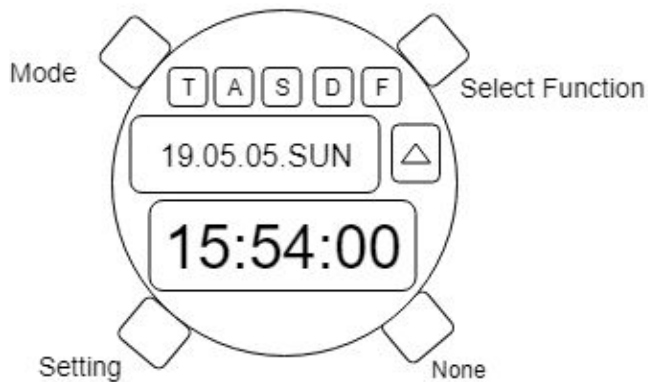
OK 버튼을 누르면 현재 기능이 조합될 기능으로 선택된다. 선택된 기능은 LCD 화면에 표시된다. 한번 더 누르면 취소된다.

Finish 버튼을 누르면 현재까지 선택된 기능의 개수를 확인한다.

현재까지 선택된 기능의 개수가 3개라면 선택된 기능들을 활성화시키고 시간보기 화면으로 전환된다.

3개가 아니라면 기능조합화면에 머무른다.

Clock-2. TimeKeeping 화면(초기화면)

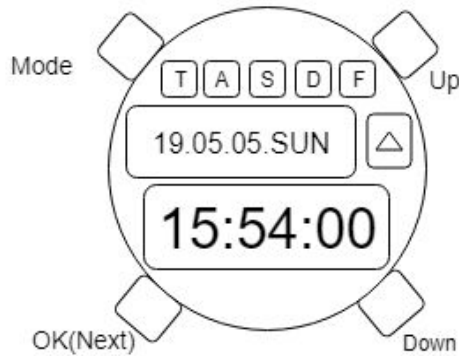


Mode버튼을 통해 다음 기능을 사용 할 수 있다.

Setting버튼을 통해서 시간을 원하는 시간으로 설정할 수 있다.

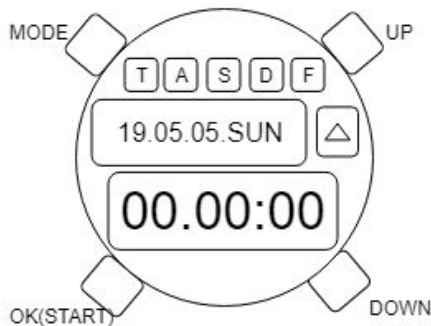
None버튼은 기능이 없는 버튼으로써 버튼을 눌러도 아무 기능이 작동하지 않는다.

Clock-3. TimeKeeping Setting(시간 설정)화면



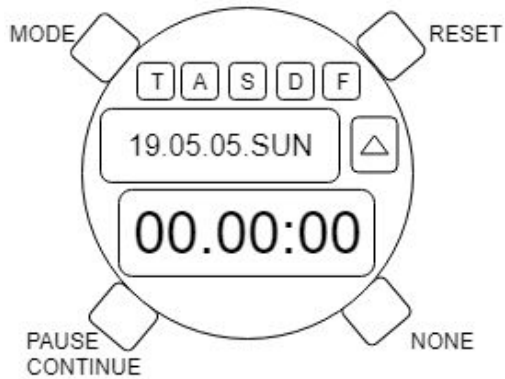
Mode 버튼을 통해서 다음 기능을 사용할 수 있다.
이 때, 시간은 설정하기 전의 시간으로 설정되고 실행이 된다.
그 후 다시 TimeKeeping으로 돌아온다면 설정화면으로 돌아온다.
Up과 Down버튼을 통해서 시간을 선택할 수 있다.
OK버튼을 통해서 연,월,일,시,분,초 의 설정이 가능하며 다음 시간선택으로 넘어간다.
이 때는 시간이 흐르지 않고 고정된 시각을 제공한다.
마지막 초 설정을 끝낸 뒤 OK를 누르면 1-1.TimeKeeping으로 돌아간다.

Clock-4. Timer 시간 설정 화면



MODE를 클릭하면 다른 기능으로 전환이 가능하다.
UP,DOWN을 클릭해서 타이머의 시/분/초를 선택한다.
OK를 클릭해서 해당 단위의 시간을 저장하고 다음 단위를 설정한다.
초 단위까지의 설정을 끝나치고 OK를 클릭하면 타이머가 실행된다.

Clock-5. Timer 실행 화면



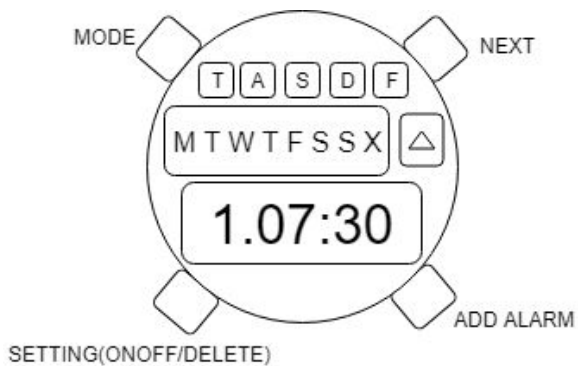
MODE를 클릭하면 다른 기능으로 전환이 가능하다.

PAUSE를 클릭하면 실행중이던 타이머를 일시정지한다.

PAUSE상태에서 PAUSE를 한 번 더 누르면 타이머가 재실행된다.

RESET을 클릭해서 타이머의 실행을 취소하고 초기화면으로 돌아간다.

Clock-6. Alarm 목록 화면



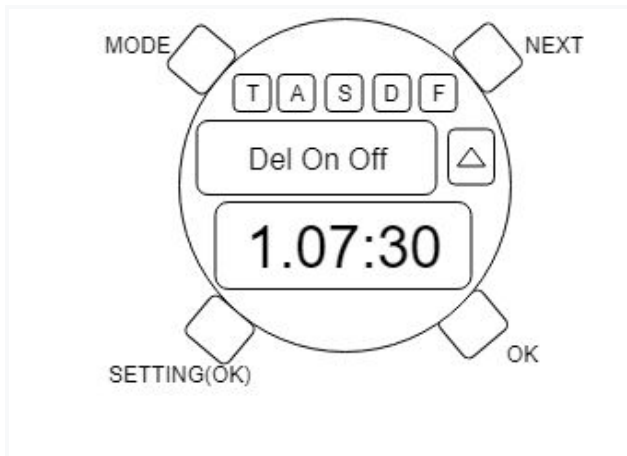
MODE를 클릭하면 다른 기능으로 전환이 가능하다.

NEXT를 클릭해서 현재 저장된 알람을 순서대로 볼 수 있다.

SETTING을 클릭하면 알람 ON-OFF 및 DELETE화면으로 전환된다.

ADD ALARM을 클릭하면 현재 알람을 수정하거나 새로운 알람을 추가할 수 있다.

Clock-7. Alarm 설정화면(ONOFF/DELETE)

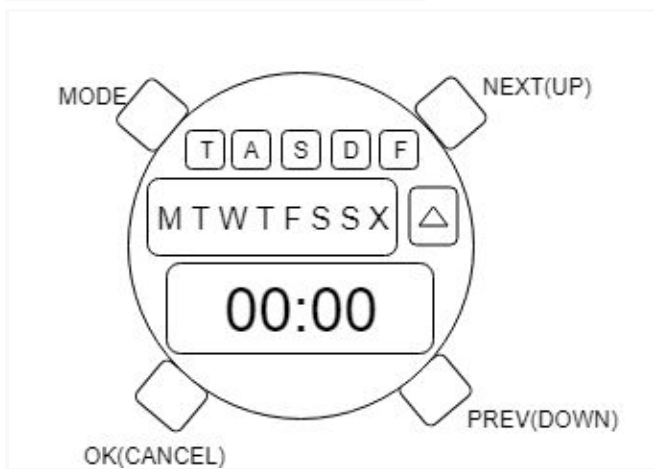


MODE를 클릭해서 다른 기능으로 전환이 가능하다.

NEXT를 클릭해서 알람을 ON 또는 OFF 또는 삭제를 지정 할 수 있다.

OK를 클릭하면 현재 상태를 저장하고 알람 목록화면으로 돌아간다.

Clock-8. Alarm 추가화면



MODE를 클릭하면 다른 기능으로 전환이 가능하다.

요일의 경우, NEXT와 PREV를 통해서 원하는 요일로 이동할 수 있고,

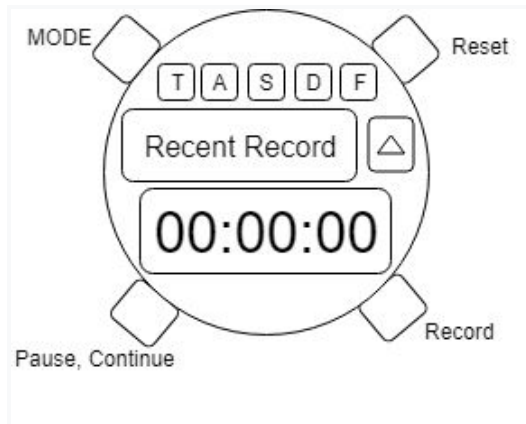
OK버튼을 통해서 알람을 설정할 수 있다.

이 때, OK을 눌렀던 요일을 한 번 더 누르게 되면 CANCEL을 할 수 있다.

마찬가지로 NEXT,PREV를 클릭해서 주기, 알람의 시/분 을 각각 선택할 수 있다.

OK를 클릭하면 현재 저장된 값을 기반으로 알람의 설정값을 변경한다..

Clock-9. Stopwatch 실행 화면



MODE를 클릭하면 다른 기능으로 전환이 가능하다.

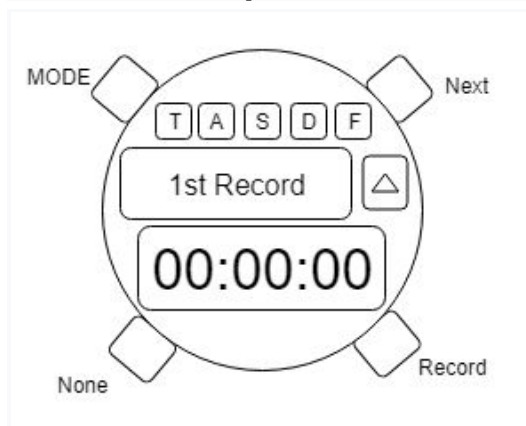
Pause 버튼을 통해 일시정지를 할 수 있고, 다시 한 번 누르면 Continue 기능이 실행된다.

Reset 버튼을 누르면 현재 측정중이던 시간이 00:00:00으로 초기화된다.

Record 버튼을 누르면 현재 시간을 기록한다. 가장 최근에 측정한 시간이 화면에 나타난다.

일시정지 상태일 때 Record 버튼을 누르면 기록 목록화면으로 넘어간다.

Clock-10. Stopwatch 기록 보기 화면

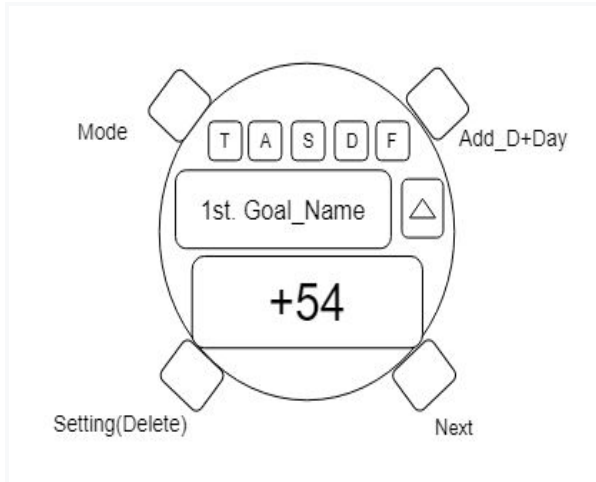


MODE를 클릭하면 다른 기능으로 전환이 가능하다.

Next를 누르면 다음 기록을 보여준다. 마지막 기록을 보여주고 나면 처음 기록을 보여준다.

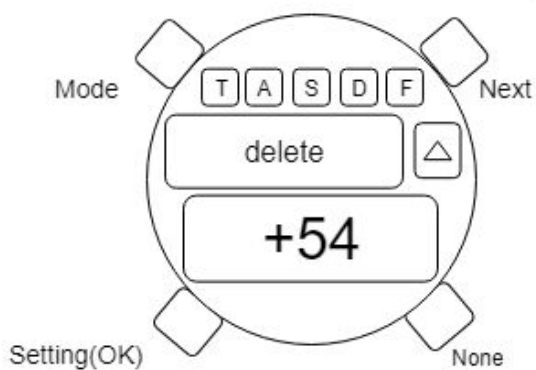
Record 버튼을 누르면 일시정지 상태로 되돌아간다.

Clock-11. D+Day 목록 보기 화면



D+day기능의 초기화면은 목록화면이다.
목록이 없다면 빈 화면을 보여준다.
목록이 있다면 목표내용과 경과날짜를 보여준다.
Next버튼을 통해서 다음 목록들을 볼 수 있다.
Setting버튼을 통해 목록을 지울 수 있다.

Clock-12. D+Day Setting(삭제) 화면



Next버튼을 통해서 삭제를 결정할 수 있다.
OK버튼을 통해서 삭제 또는 변경없음을 실행한다.
그 후 목록보기 화면으로 돌아간다.
삭제를 했다면 그 다음 목록을 삭제를 안했다면 해당목록부터 보여준다.

Clock-13. D+Day 목록 추가 화면

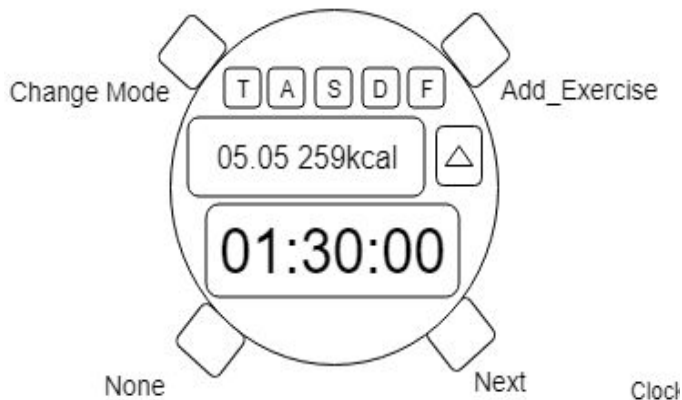


Up,Down버튼을 통해서 시스템이 제공하는 목표목록을 볼 수 있고, 날짜를 변경할 수 있다.

연도선택으로 넘어가고, 날짜까지의 설정이 끝나 OK버튼을 누르면 목표를 선택할 수 있고 OK버튼을 눌러서 설정을 끝낸다.

해당 D+Day가 추가가되고 목록화면으로 돌아간다.

Clock-14. Fitness 목록 화면



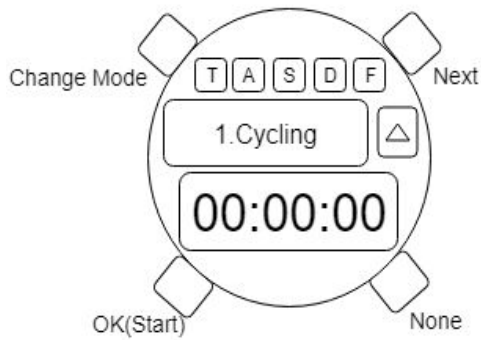
Fitness기능의 초기화면이다.

첫화면은 오늘의 운동량을 보여준다.

이전에 운동기록이 있다면,Next버튼을 통해서 이전 날짜의 운동량과 운동시간을 볼 수 있다.

Add_Exercise 버튼을 통해서 오늘의 운동을 추가 할 수 있다.

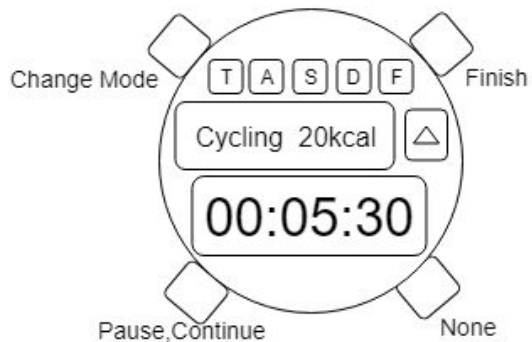
Clock-15. Fitness 운동 추가 화면(종목 선택)



운동 종목을 선택하는 화면

Next를 통해서 시스템이 제공하는 3가지 운동종목 중 하나를 선택할 수 있다.
OK버튼을 통해 결정을 하면 바로 운동시간 측정이 실행된다.

Clock-16. Fitness 운동 실행 화면



운동시간 측정이 되면서 소모되는 칼로리량을 화면에 함께 실시간으로 보여준다.
Pause버튼을 통해 일시정지를 할 수 있고, Pause버튼을 한 번 더 누르게 되면
Continue기능이 실행된다.
Finish버튼을 통해서 운동을 종료할 수 있다.

Clock-17. Stop Buzzer

stop_buzzer



4개의 버튼 중 임의의 것을 눌러서 버저를 멈출 수 있다.
화면은 전환되지 않고 현재 화면을 그대로 보여준다.

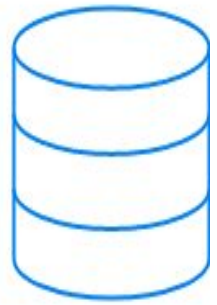
3. Activity 2143. Refine System Architecture



(User)



(Clock)



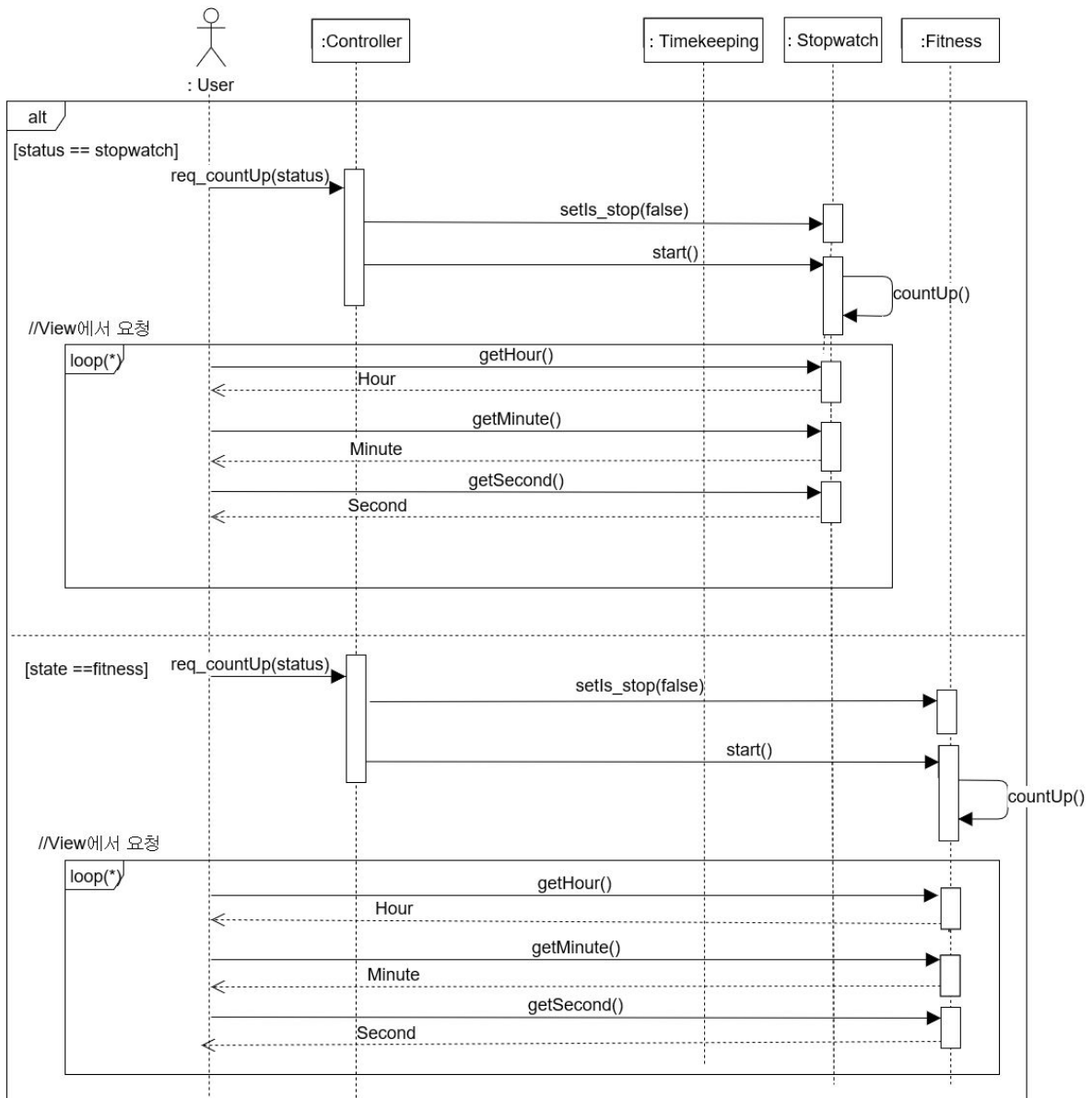
(DataBase)

4. Activity 2144. Define Interaction Diagram

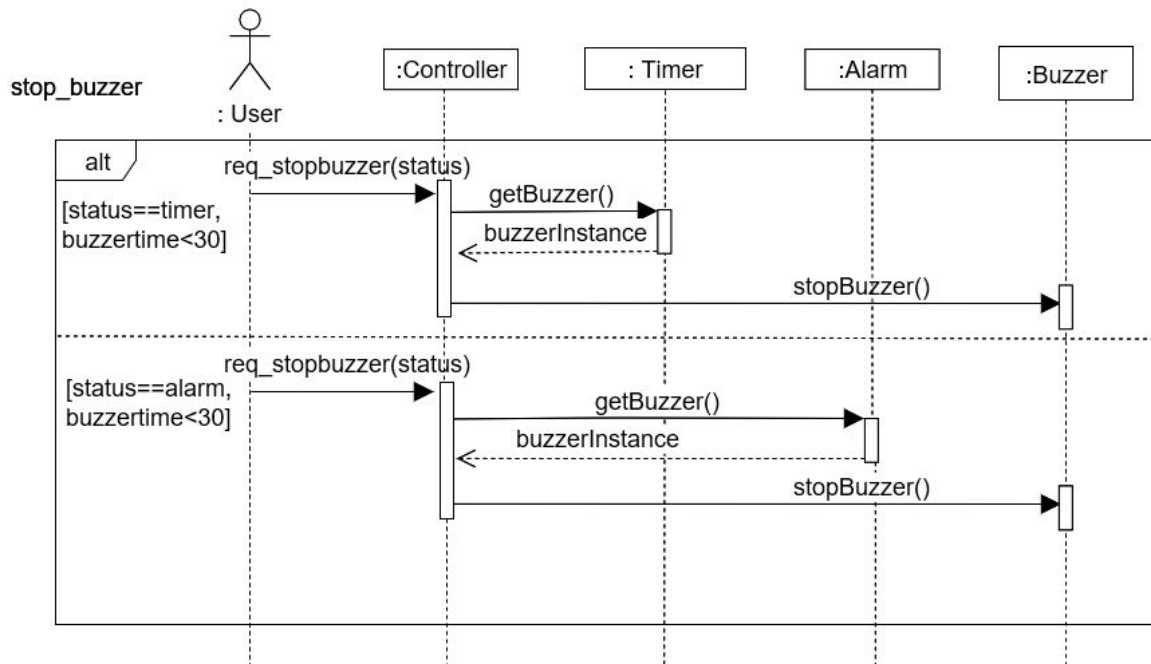
4-1) Common function

4-1-1) Use case : count_up

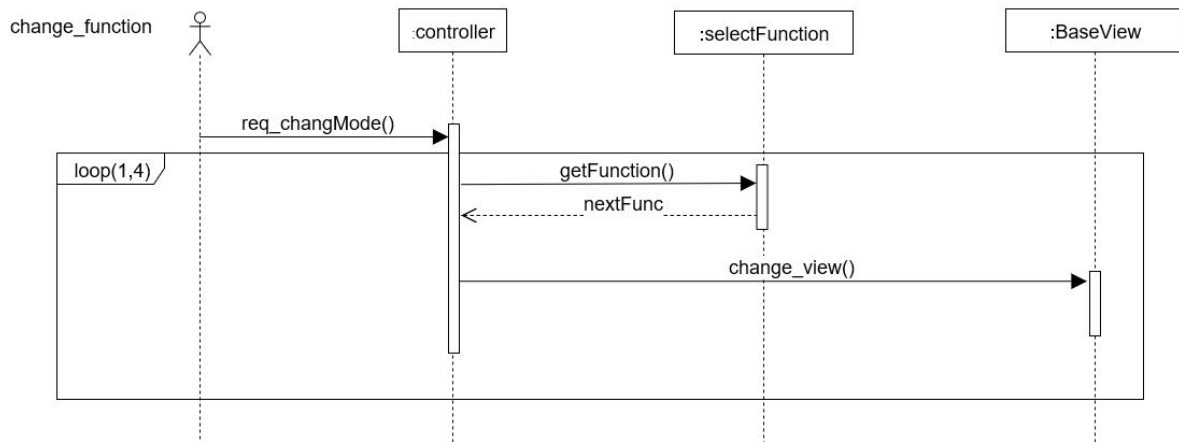
count_up



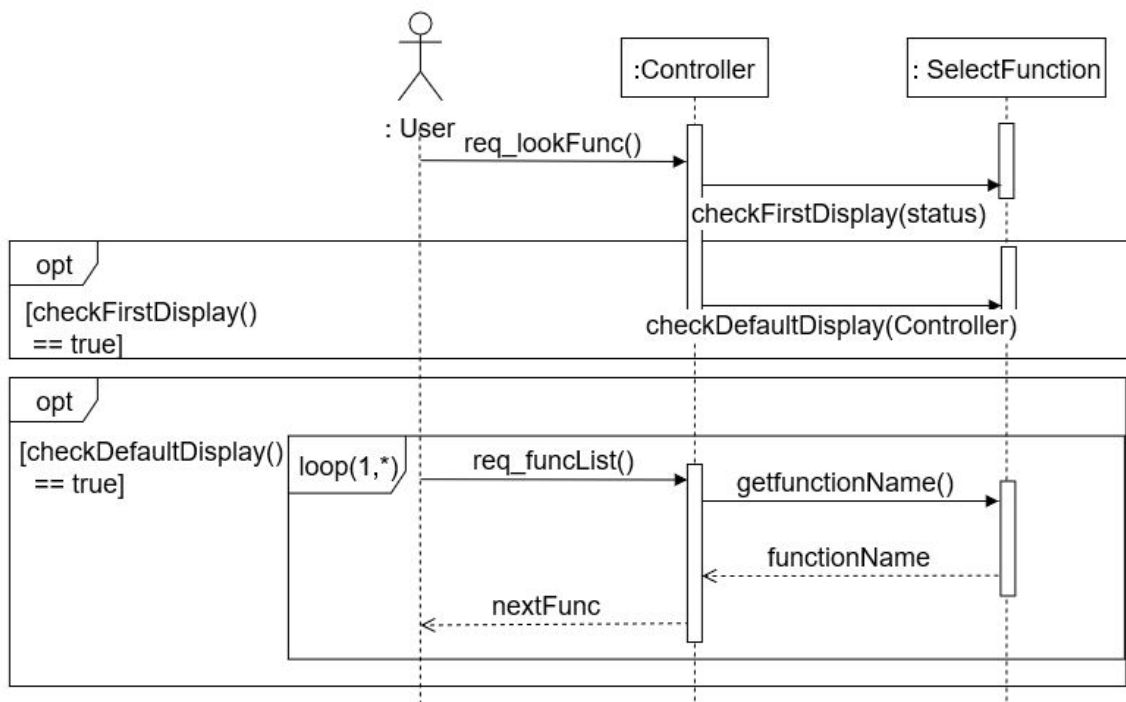
4-1-2) Use case : stop_buzzer



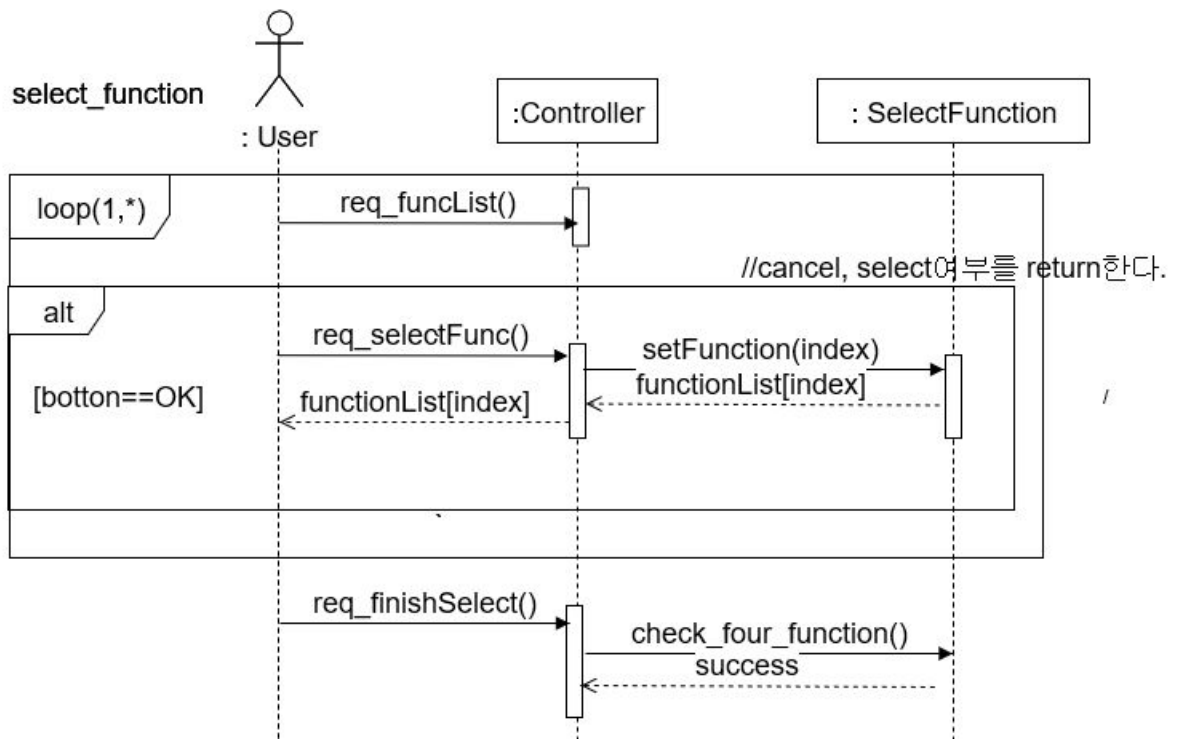
4-1-3) Use case : change_function



4-1-4) Use case : look_function

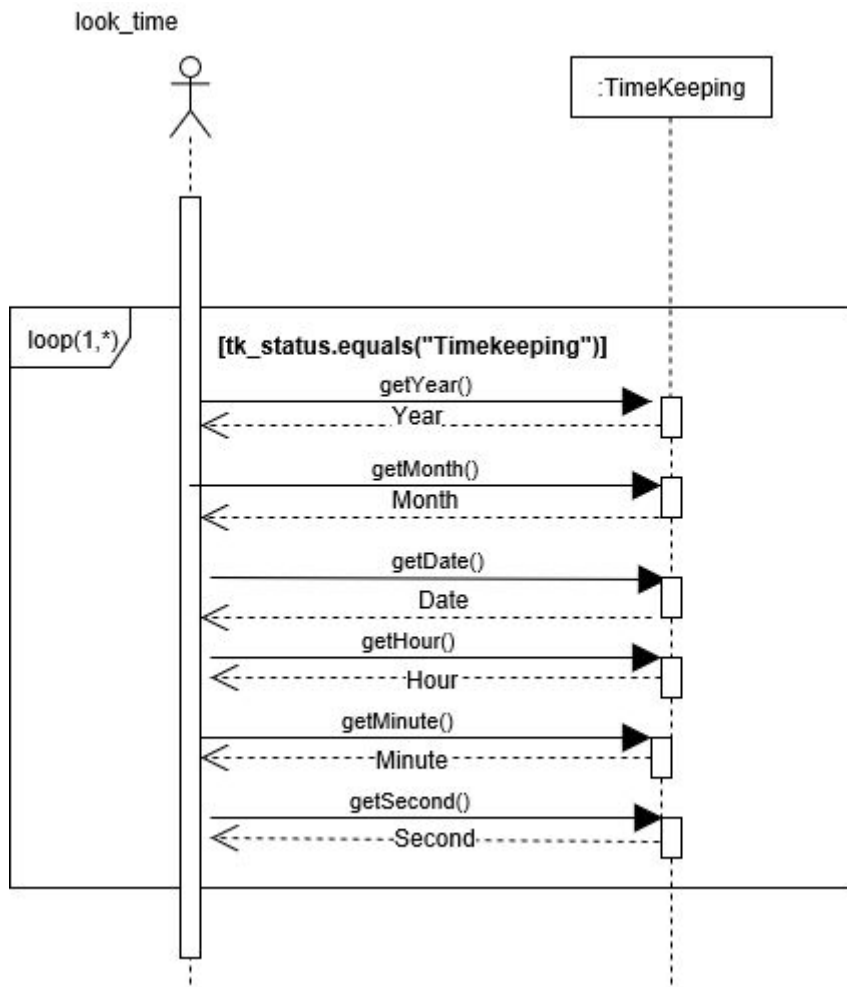


4-1-5) Use case : select_function

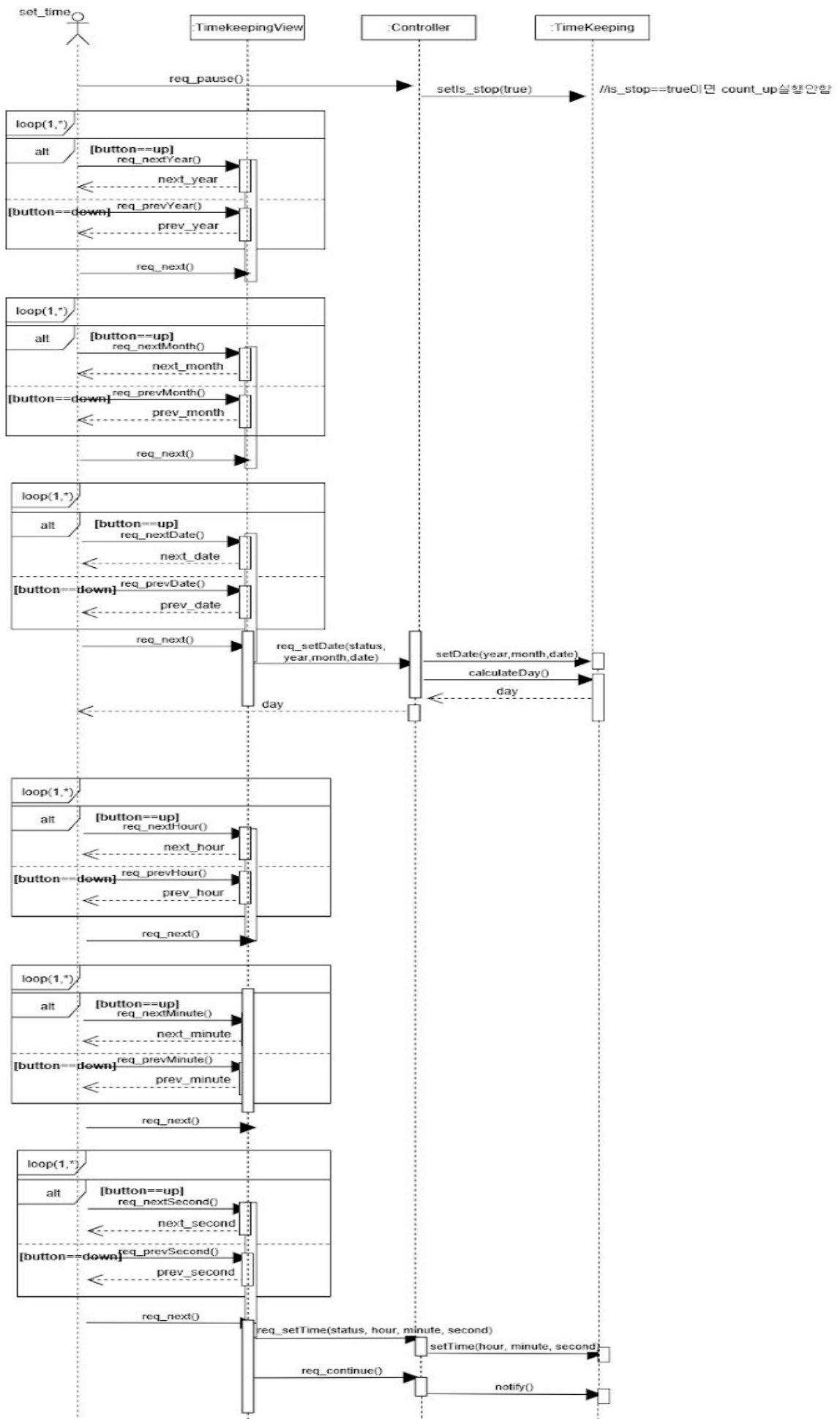


4-2) Timekeeping function

4-2-1) Use case : look_time

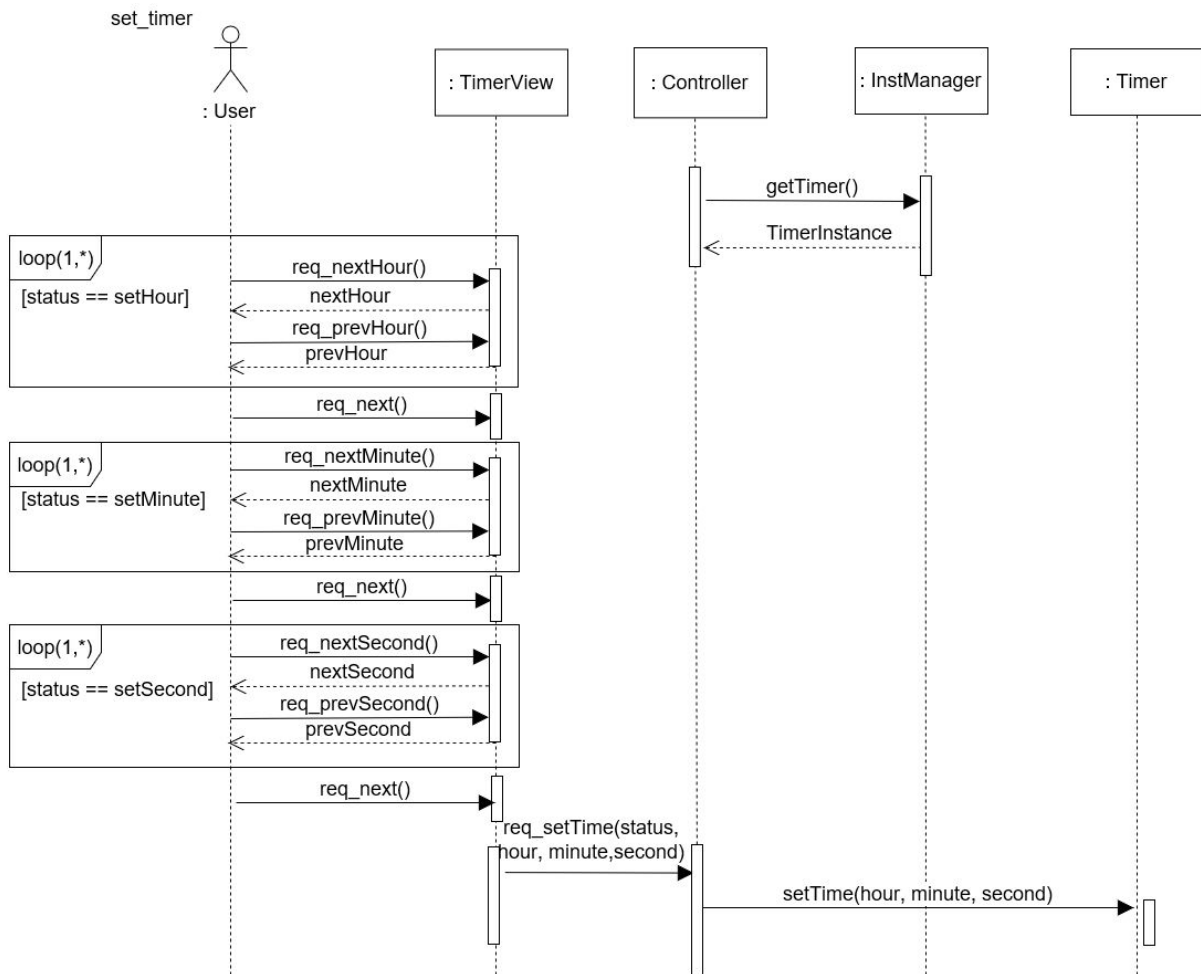


4-2-2) Use case : set_time

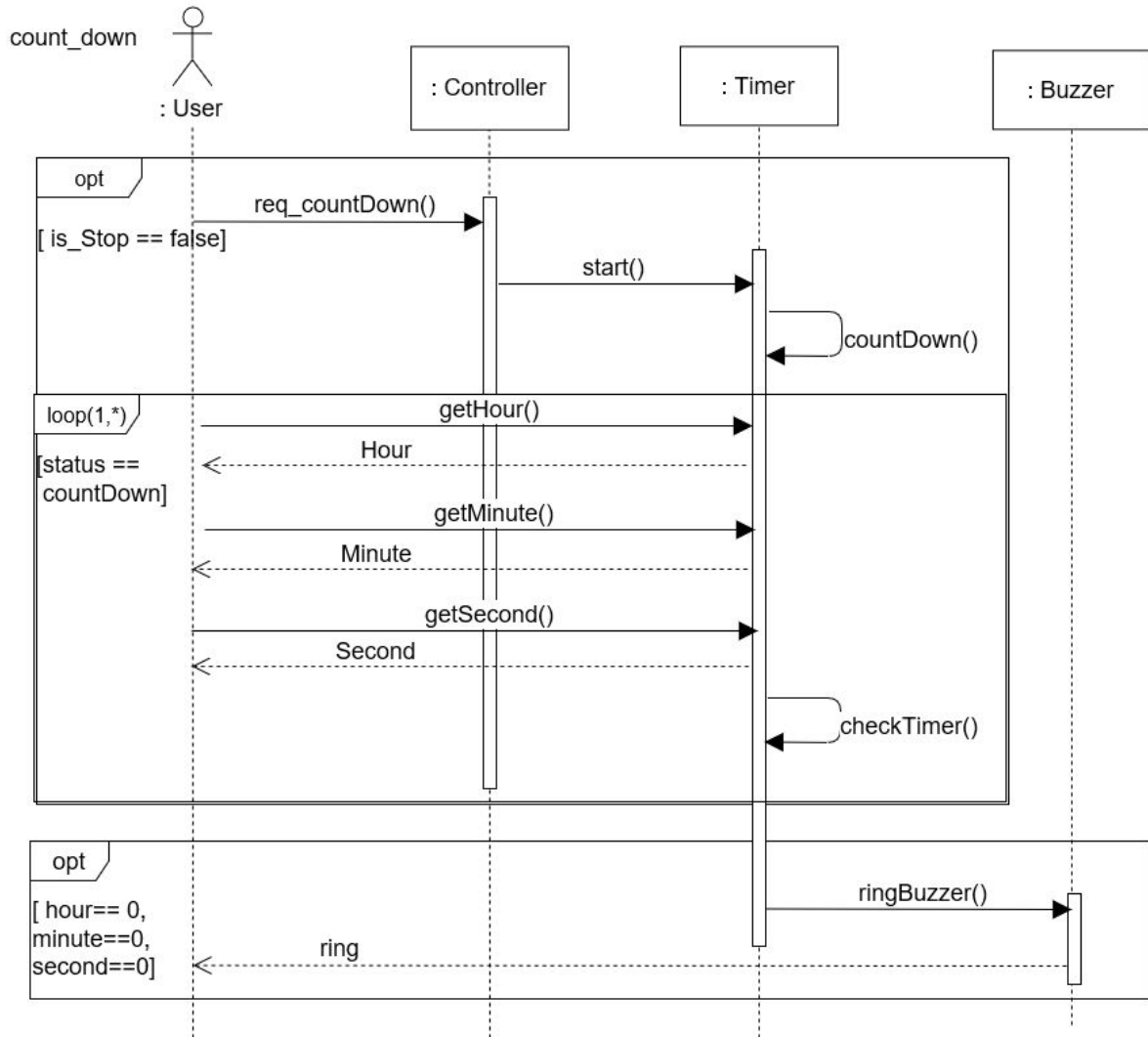


4-3) Timer function

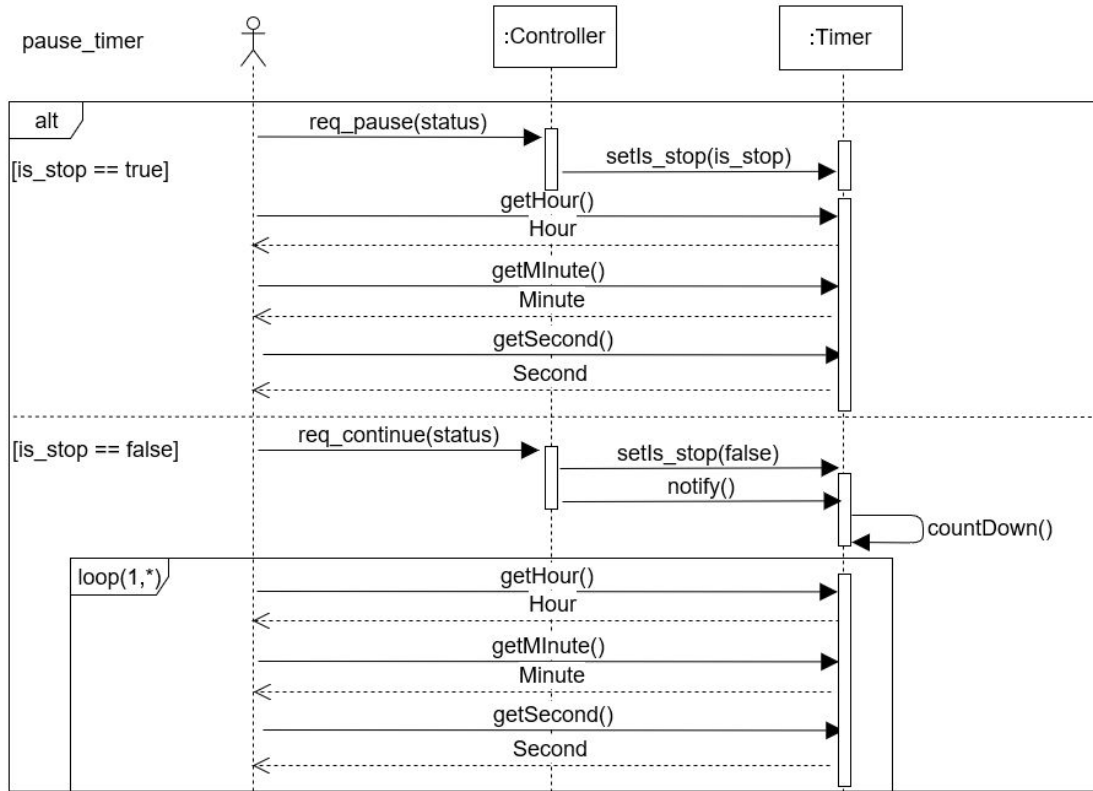
4-3-1) Use case : set_timer



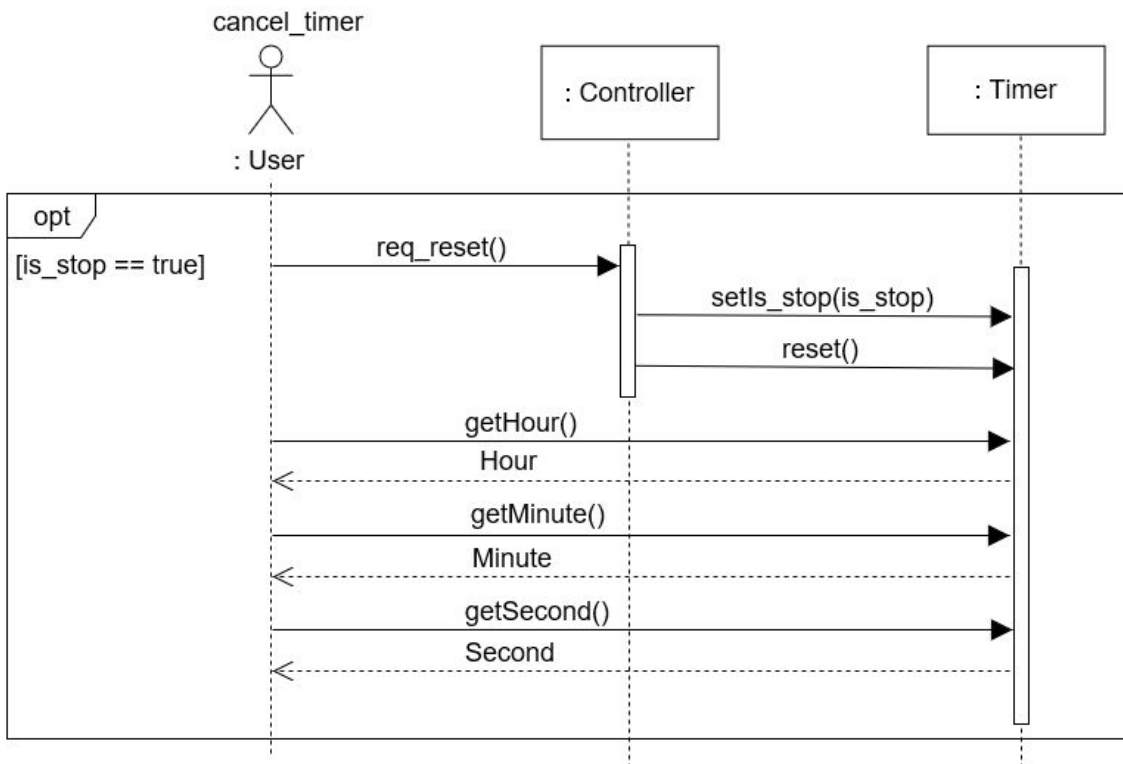
4-3-2) Use case : count_down



4-3-3) Use case : pause_timer

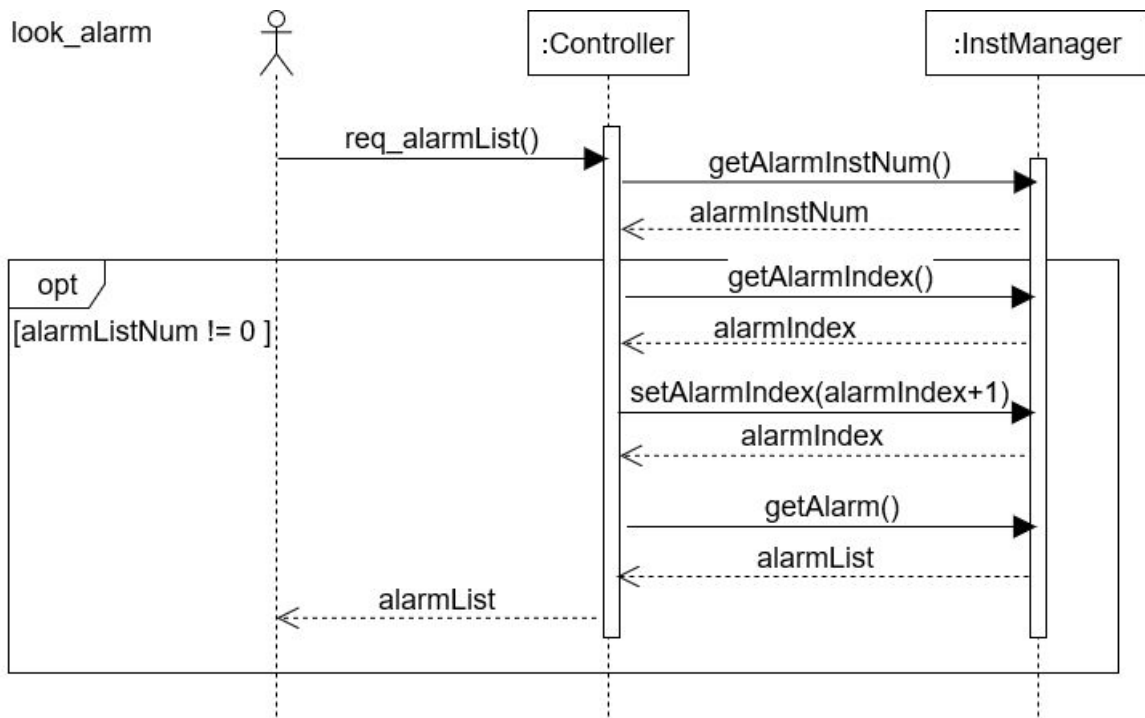


4-3-4) Use case : cancel_timer

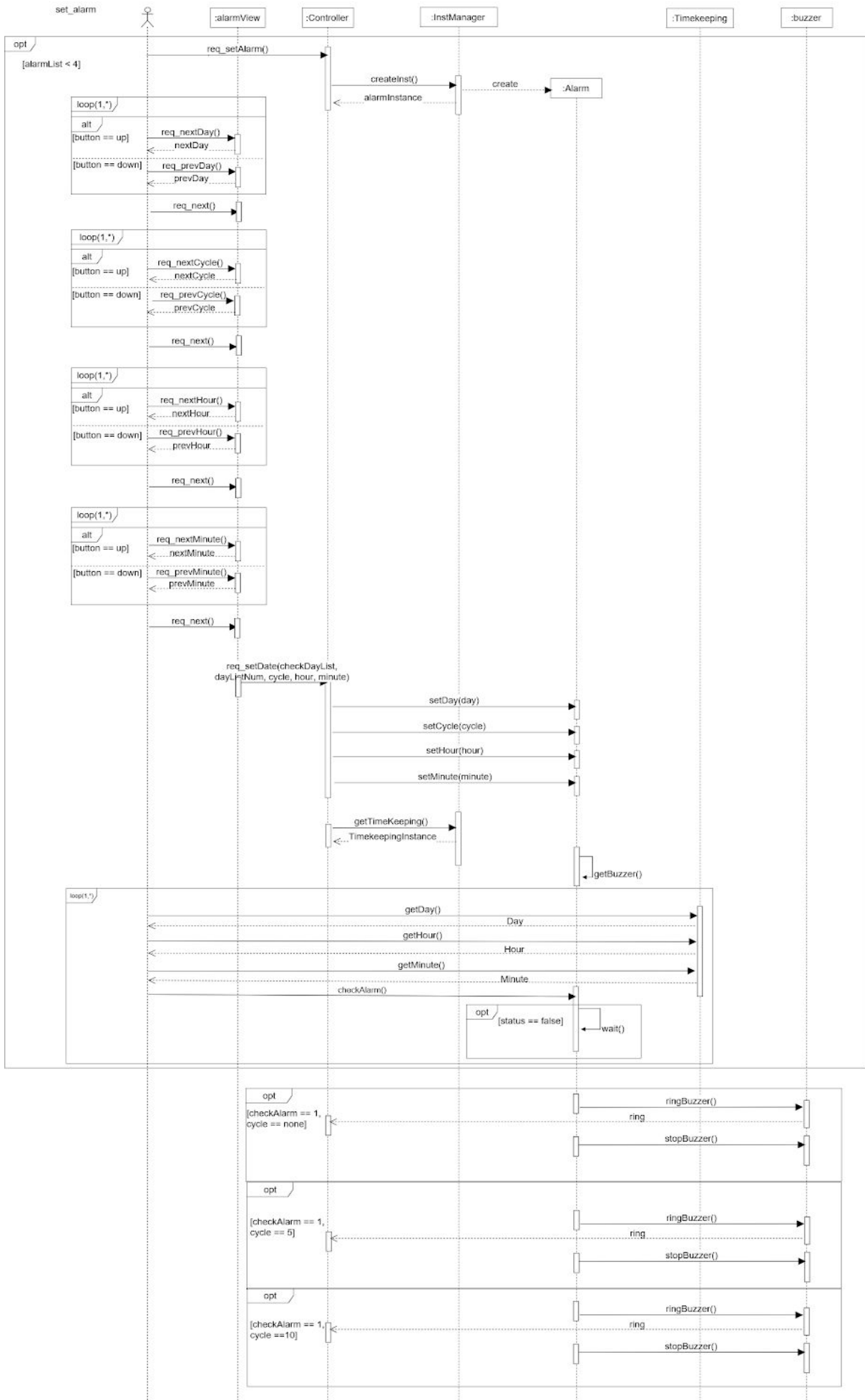


4-4) Alarm function

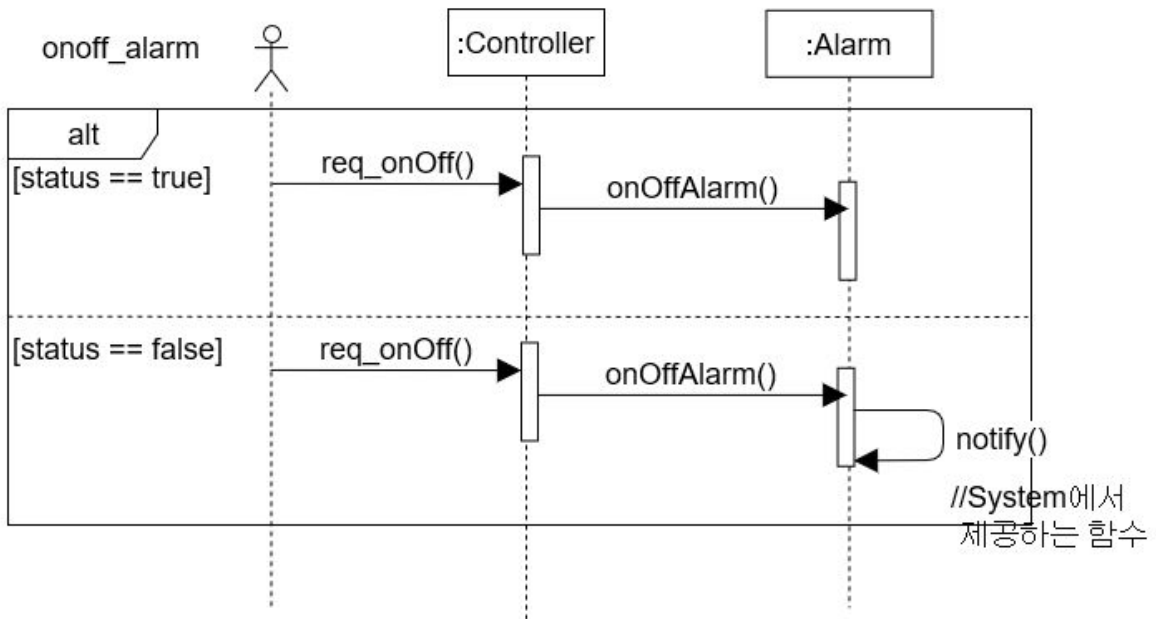
4-4-1) Use case : look_alarm



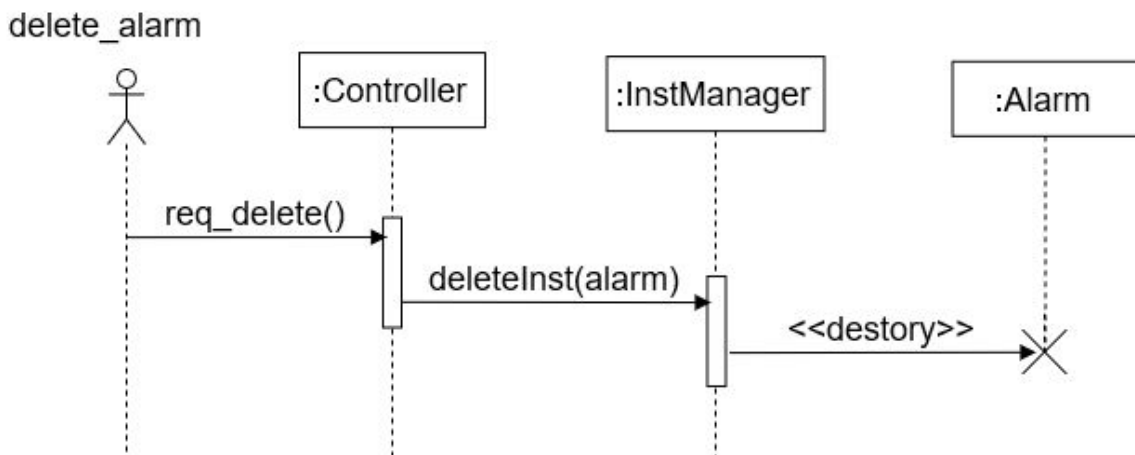
4-4-2) Use case : set_alarm



4-4-3) Use case : onoff_alarm

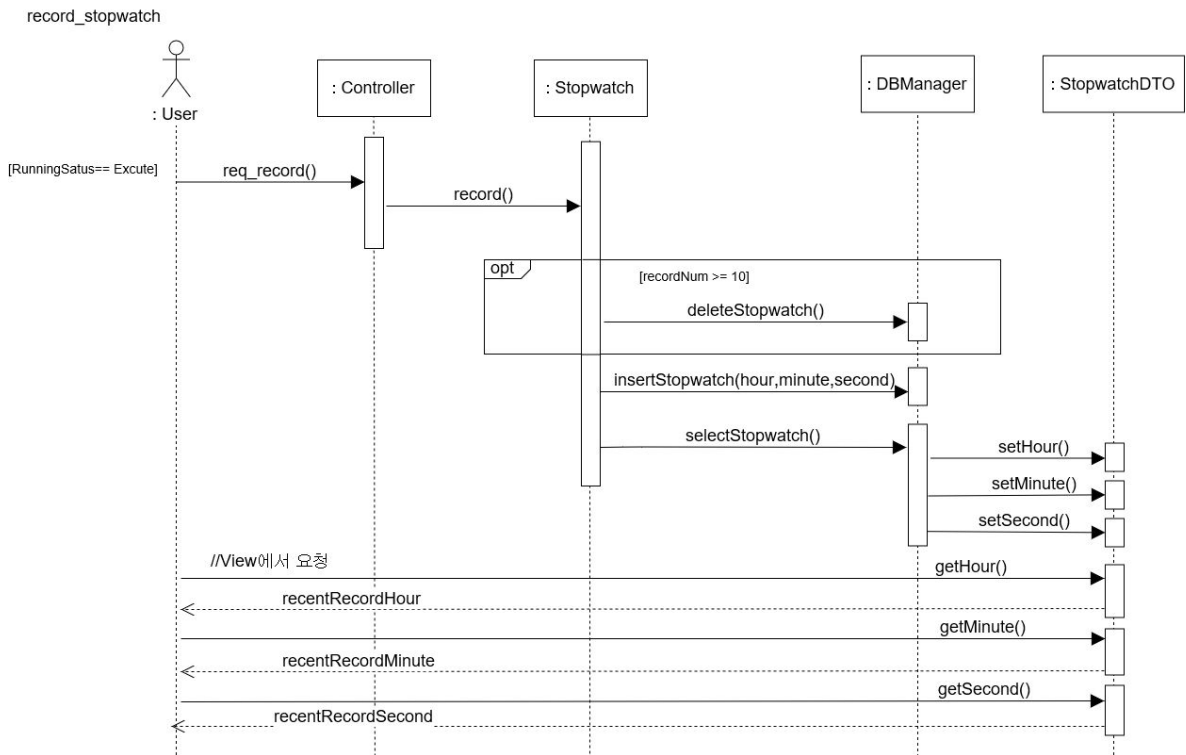


4-4-4) Use case : delete_alarm

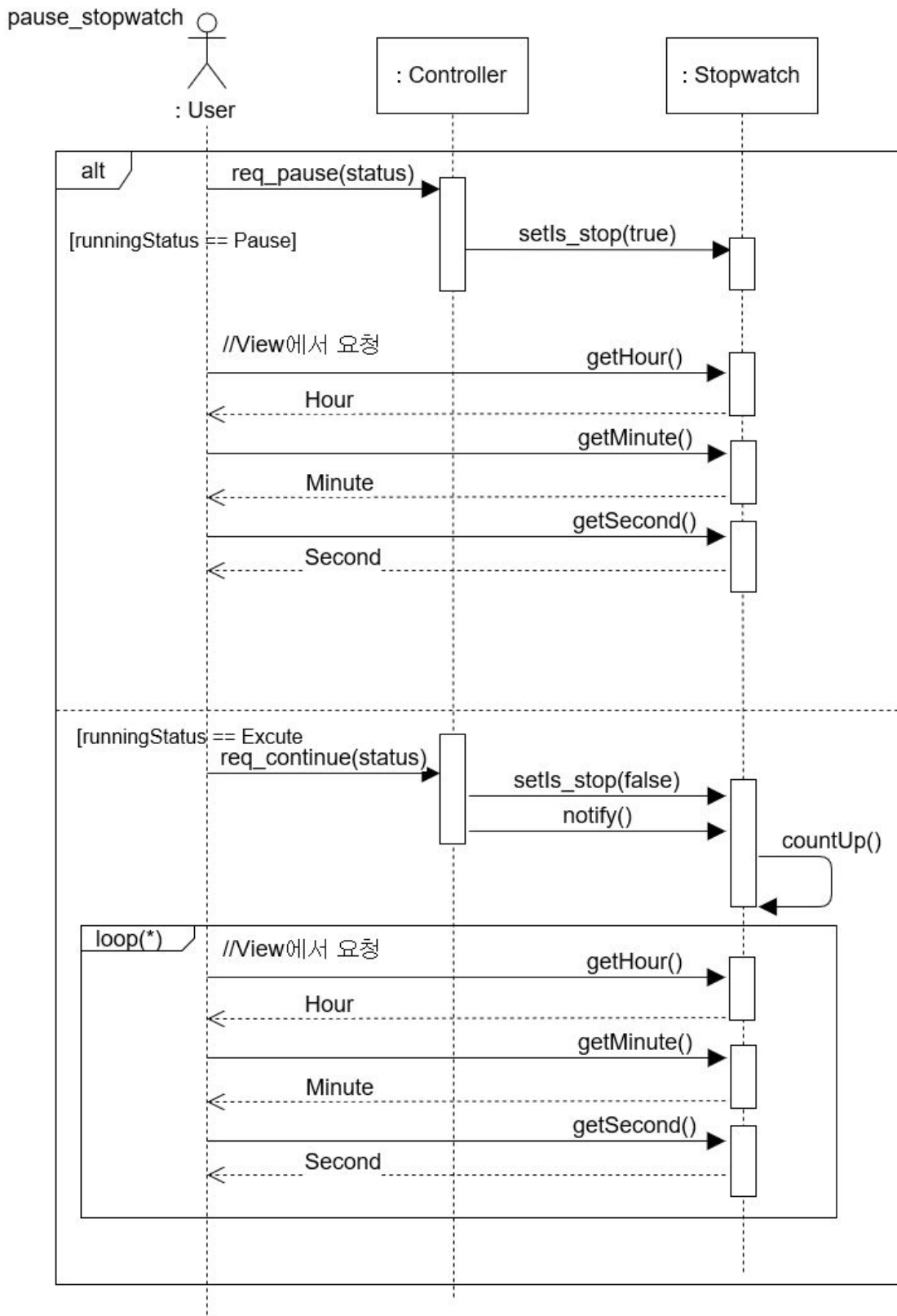


4-5) Stopwatch function

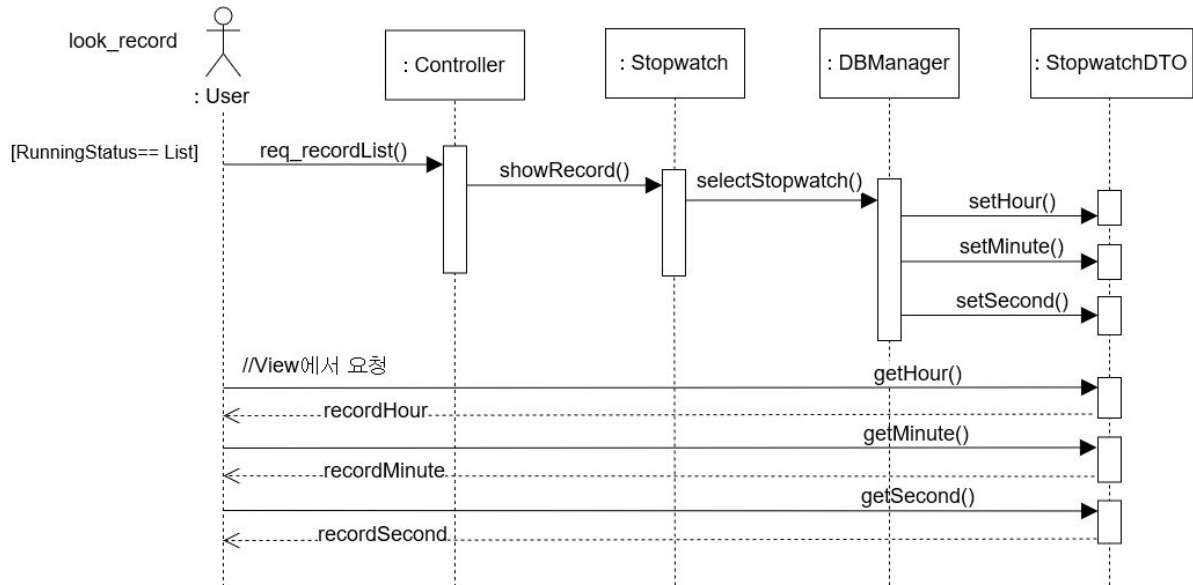
4-5-1) Use case : record_stopwatch



4-5-2) Use case : pause_stopwatch

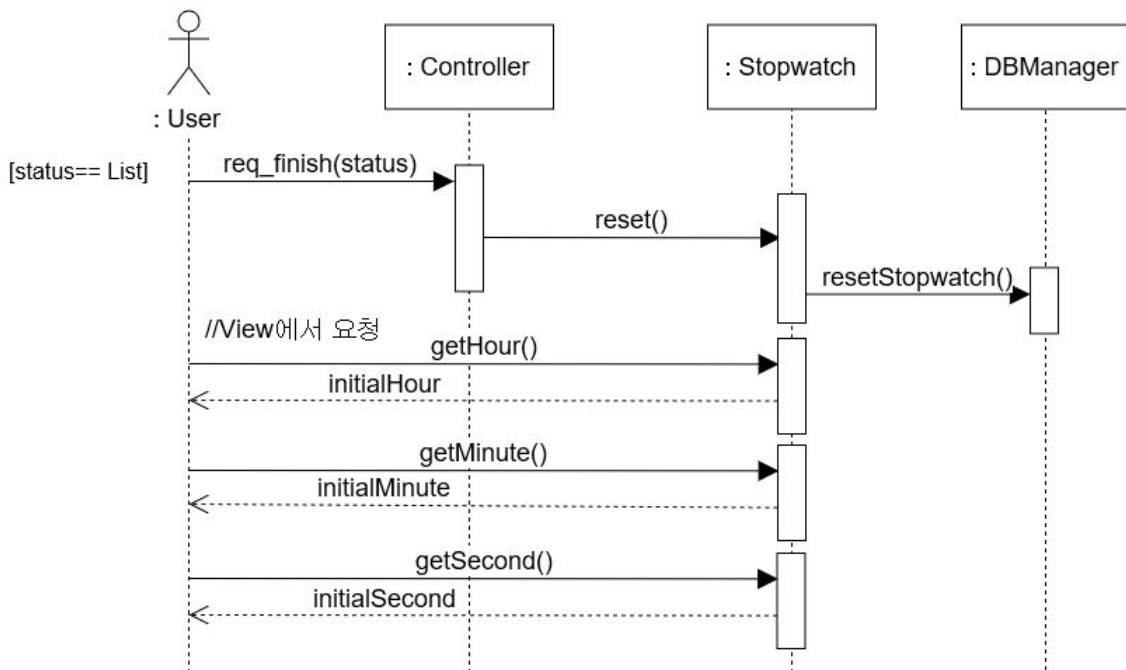


4-5-3) Use case : look_record



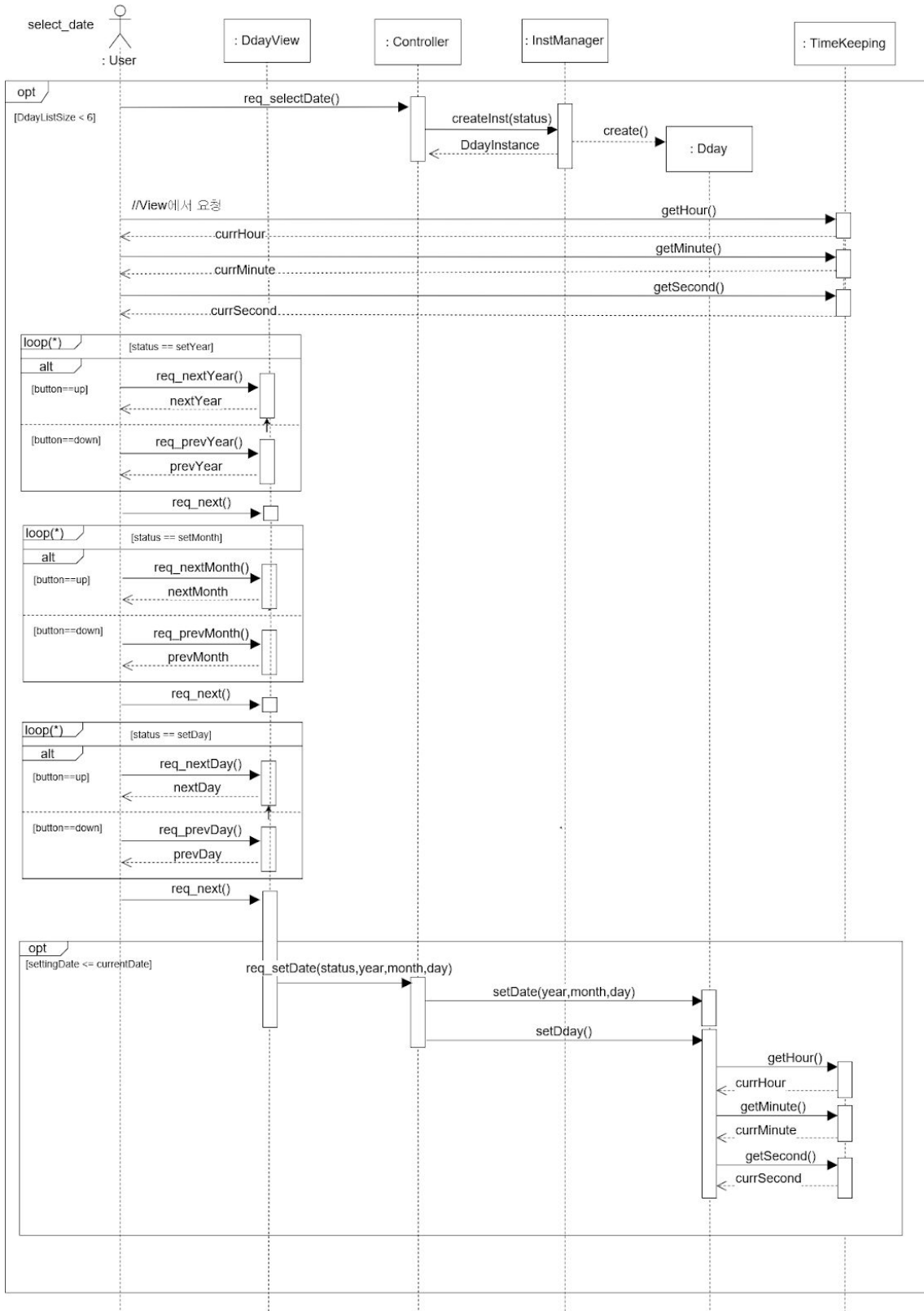
4-5-4) Use case : reset_stopwatch

reset_stopwatch



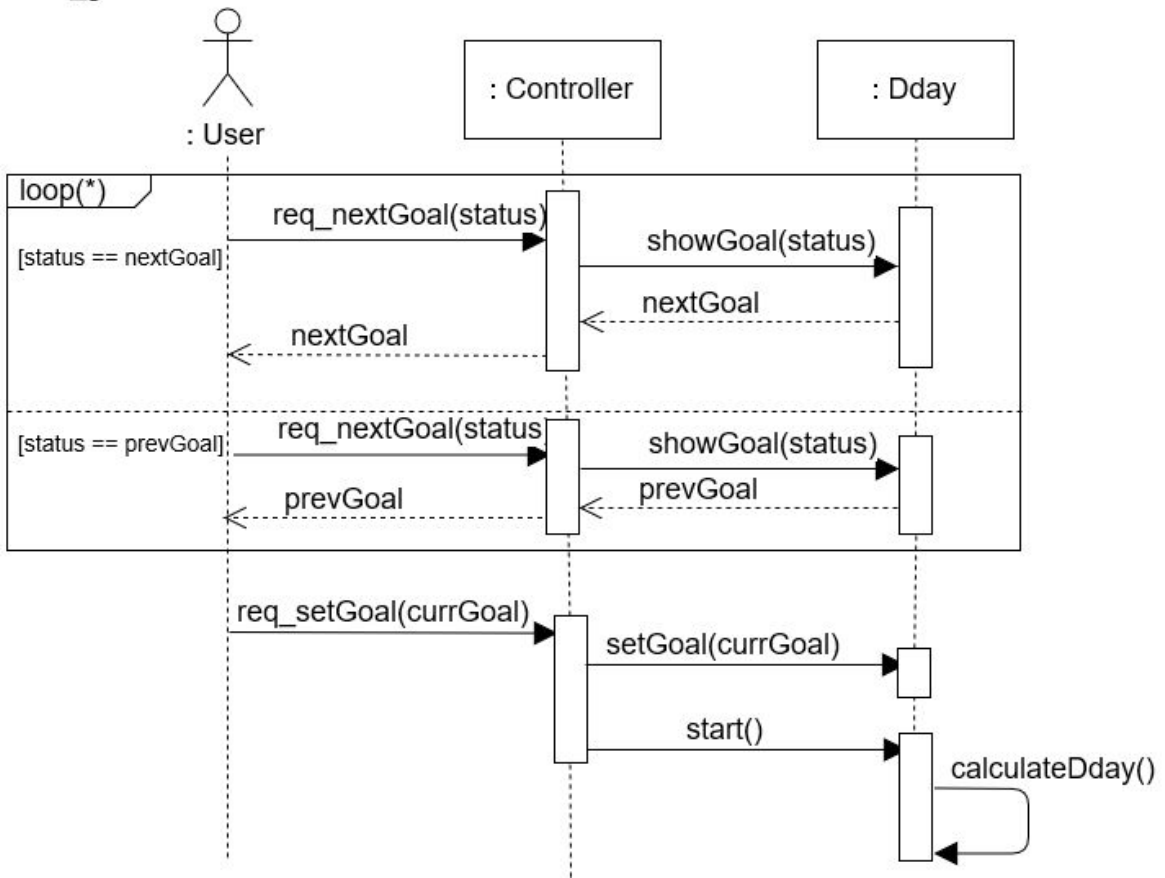
4-6) D+day function

4-6-1) Use case : select_date



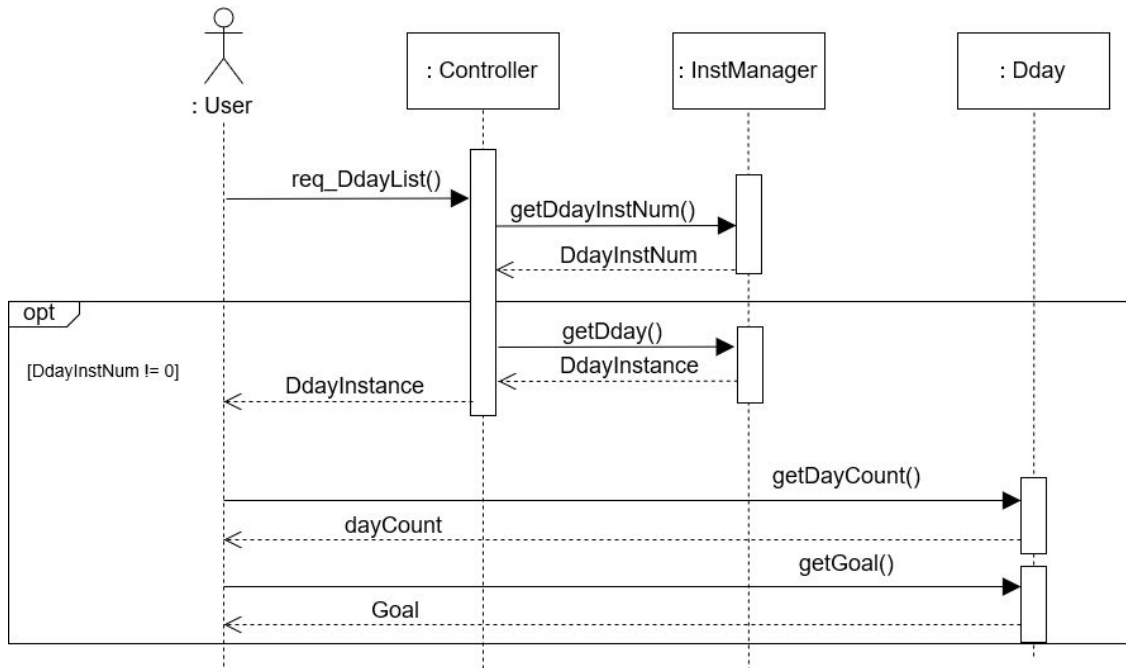
4-6-2) Use case : select_goal

select_goal

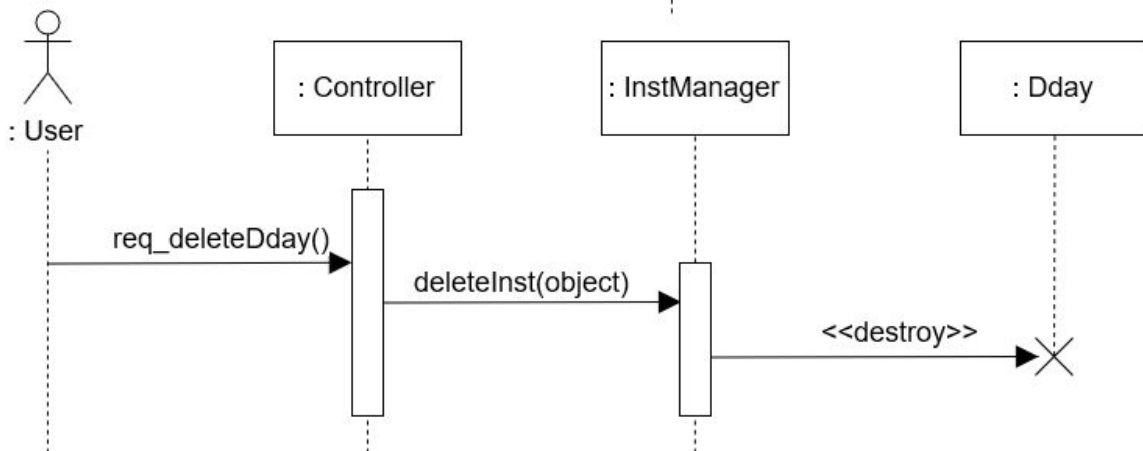


4-6-3) Use case : look_Dday

look_Dday

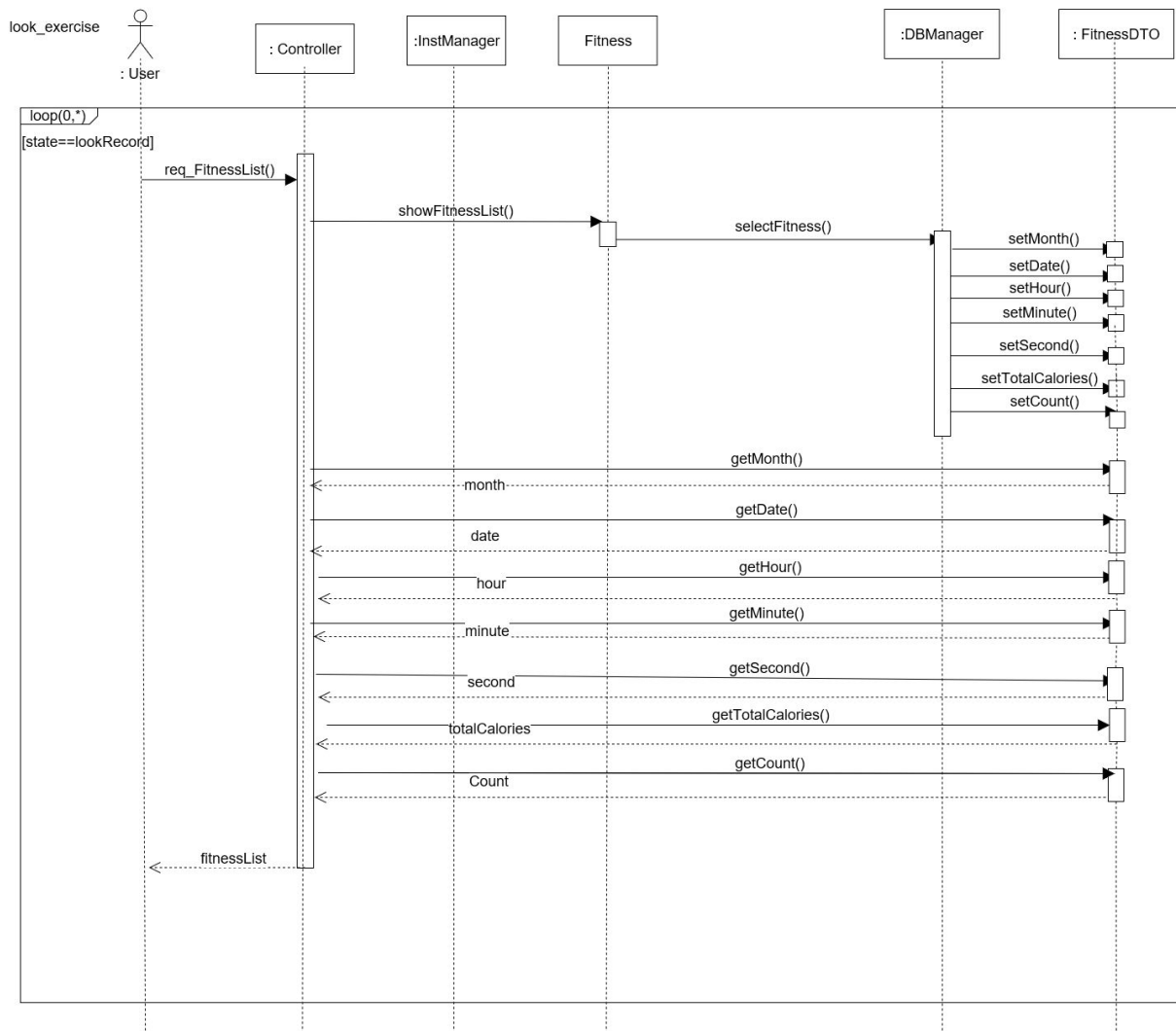


4-6-4) Use case : delete_Dday

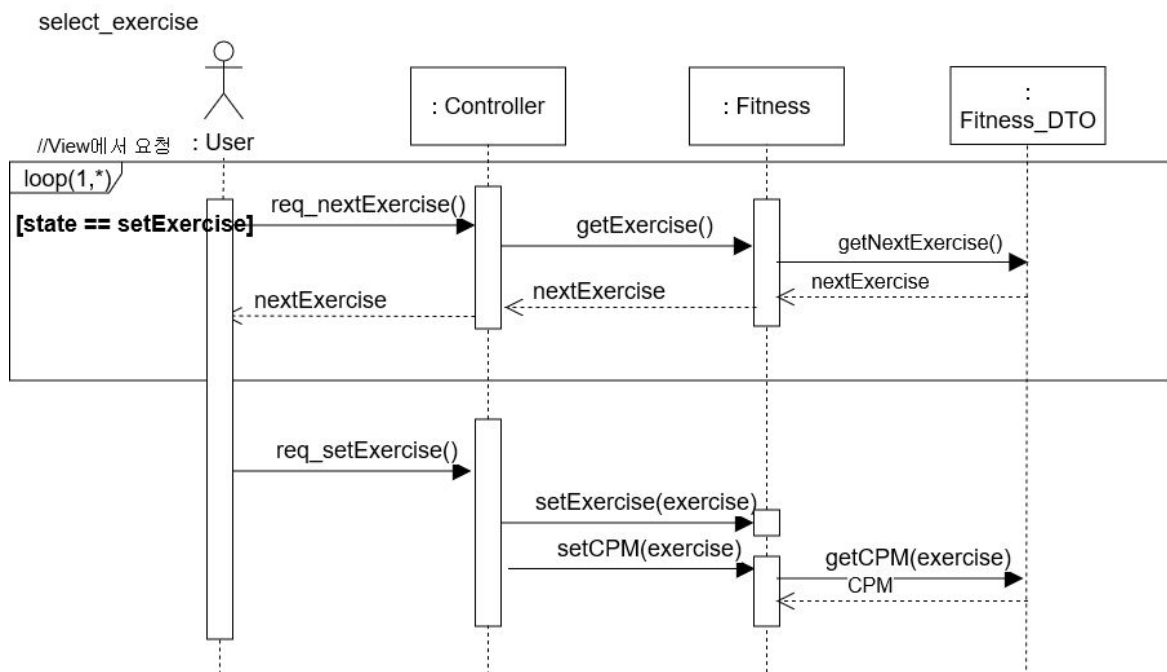


4-7) Fitness function

4-7-1) Use case : look_exercise

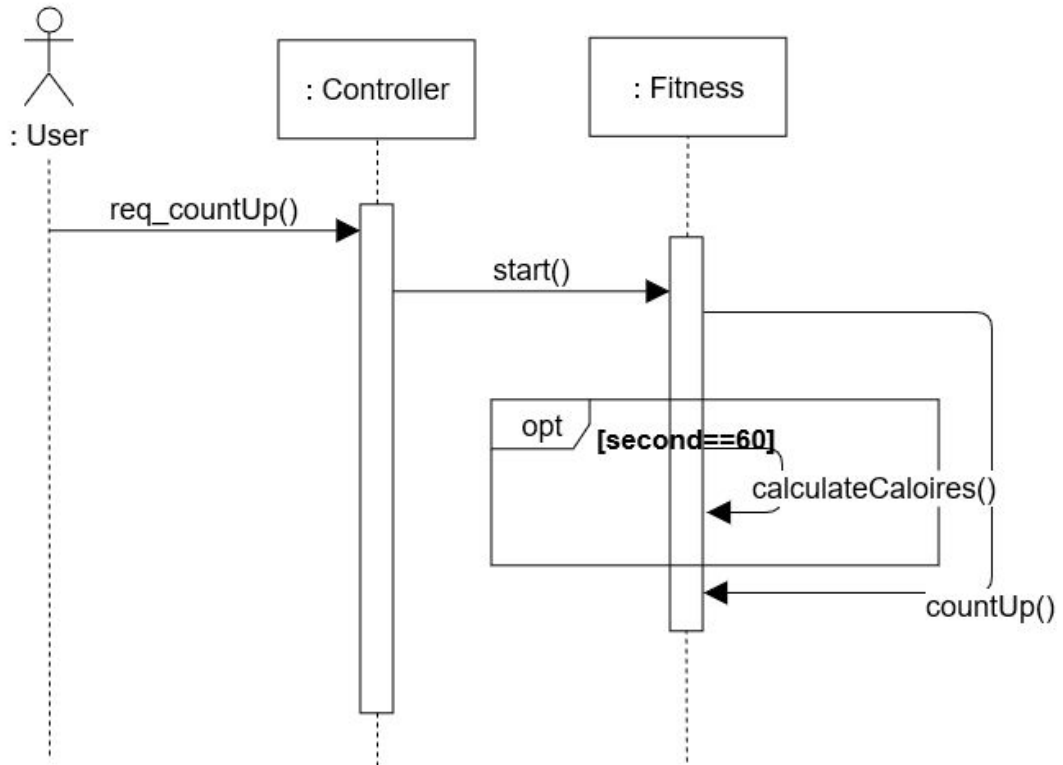


4-7-2) Use case : select_exercise



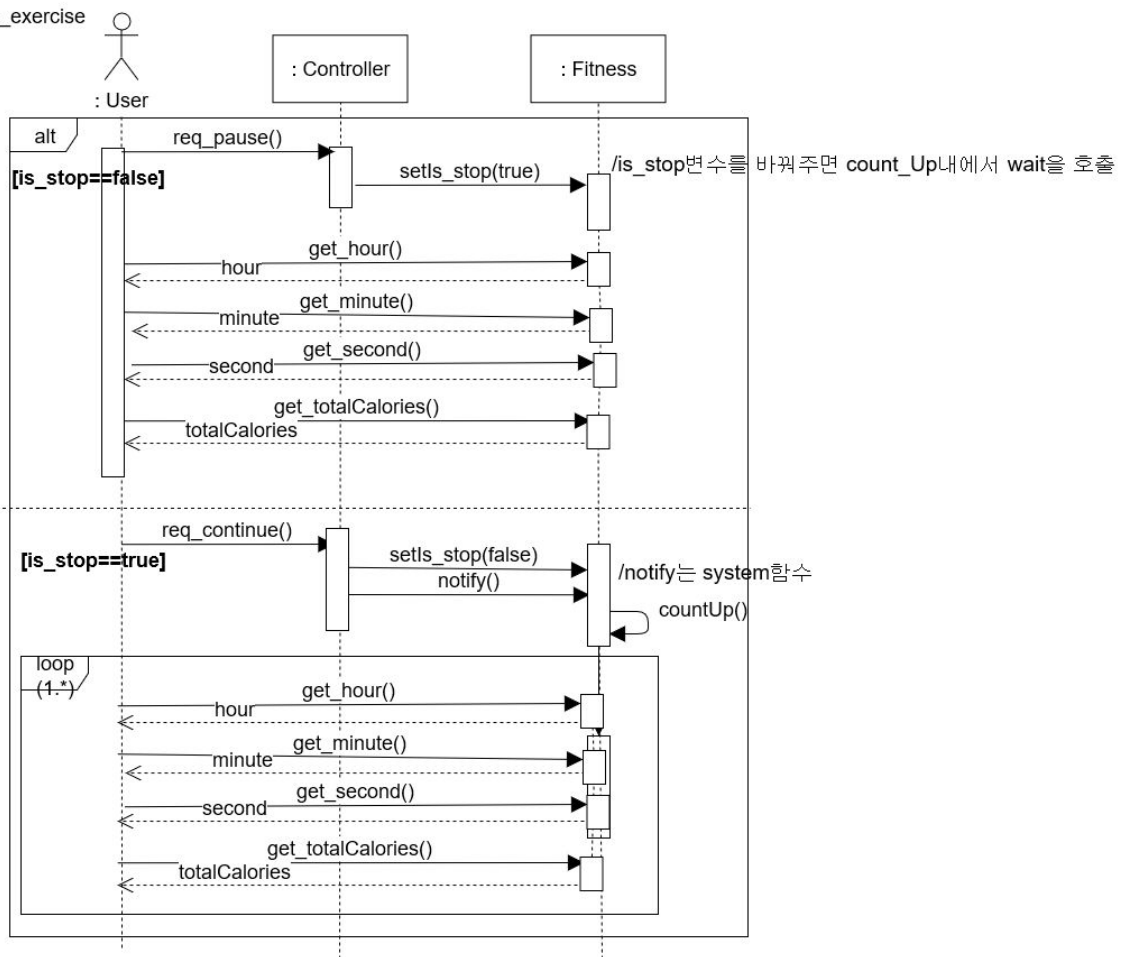
4-7-3) Use case : calculate_calories

calculate_calories

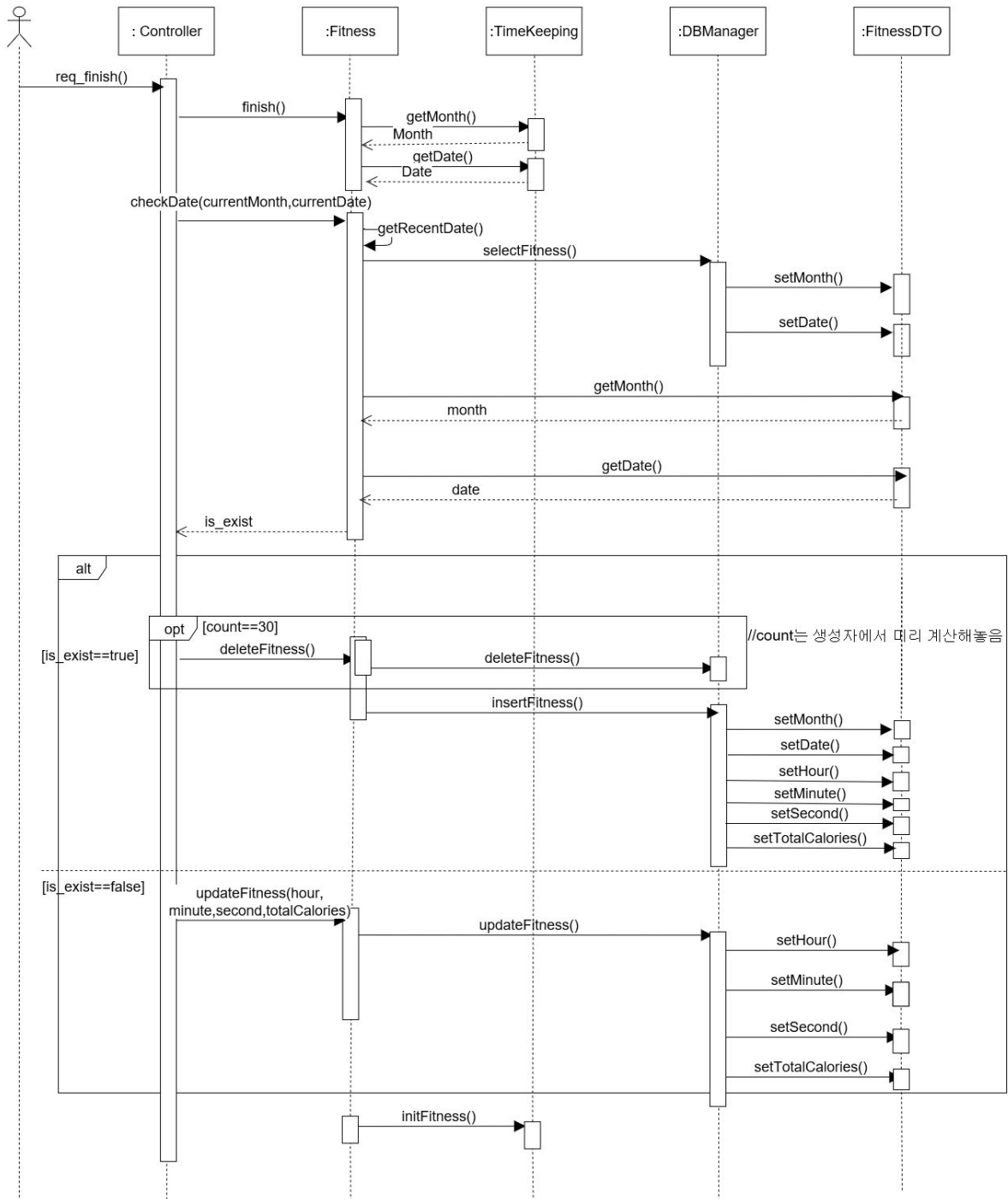


4-7-4) Use case : pause_exercise

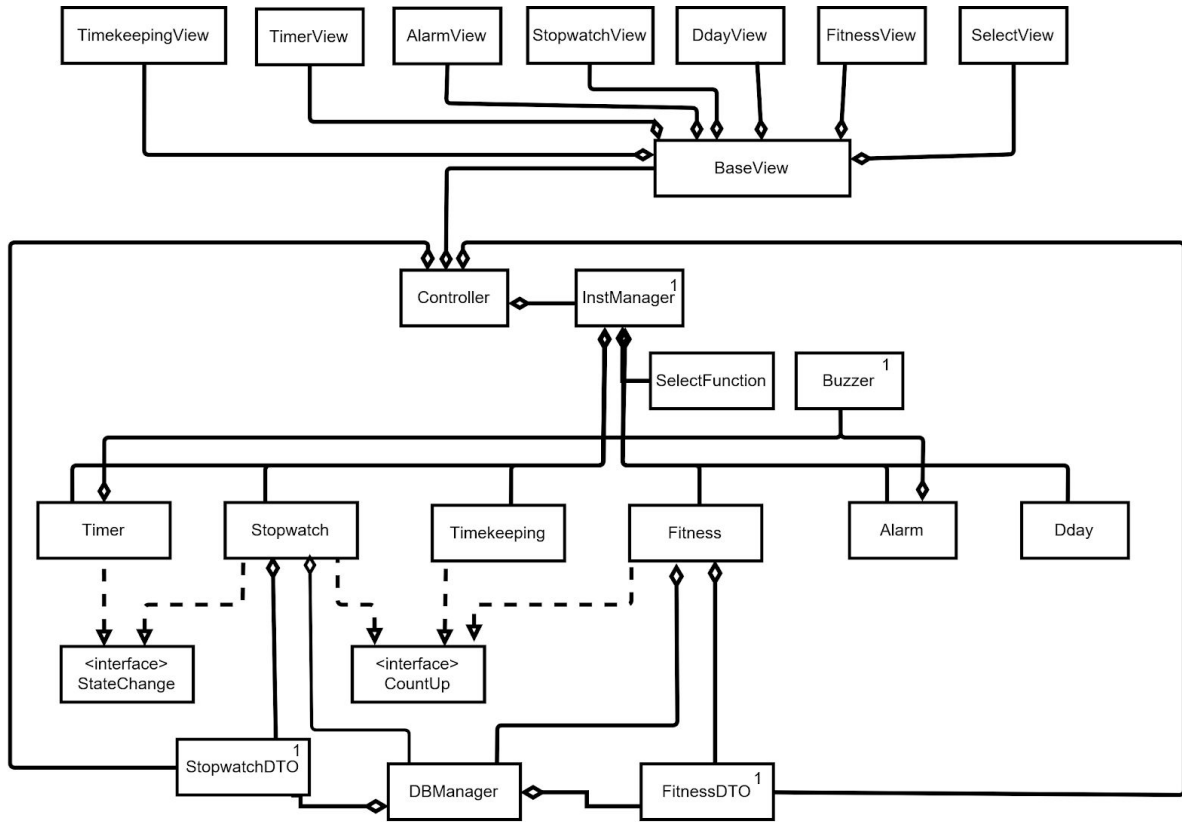
pause_exercise

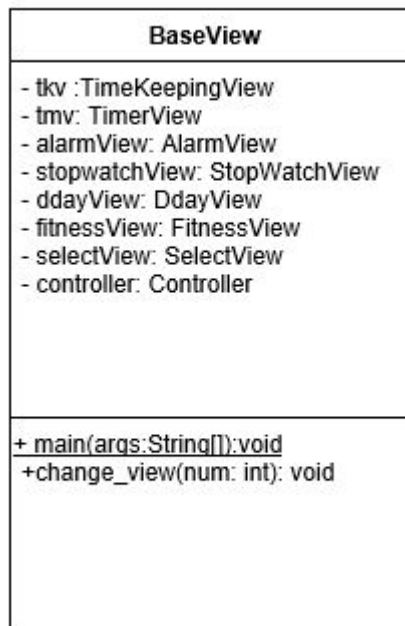


4-7-5) Use case : finish_exercise

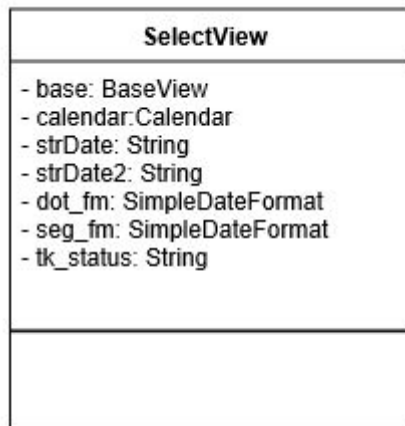


5. Activity 2145. Define Design Class Diagram





5-2) SelectView Class



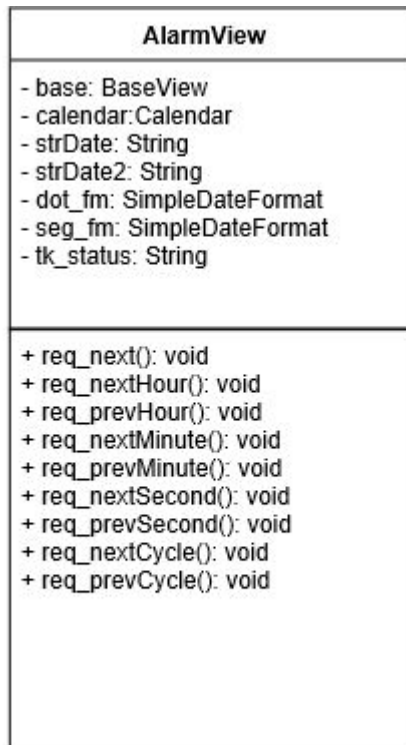
5-3) TimekeepingView Class

TimeKeepingView
- base: BaseView - calendar: Calendar - strDate: String - strDate2: String - dot_fm: SimpleDateFormat - seg_fm: SimpleDateFormat - tk_status: String
+ req_next(): void + req_nextYear(): void + req_prevYear(): void + req_nextMonth(): void + req_prevMonth(): void + req_nextDate(): void + req_prevDate(): void + req_nextHour(): void + req_prevHour(): void + req_nextMinute(): void + req_prevMinute(): void + req_nextSecond(): void + req_prevSecond(): void

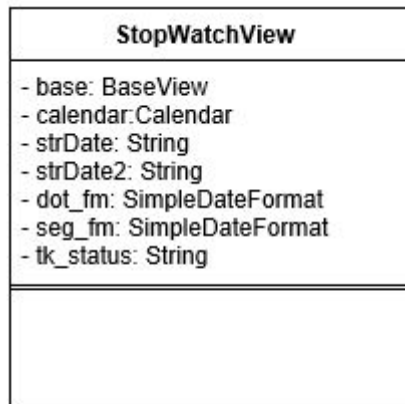
5-4) TimerView Class

TimerView
- base: BaseView - calendar: Calendar - strDate: String - strDate2: String - dot_fm: SimpleDateFormat - seg_fm: SimpleDateFormat - tk_status: String
+ req_next(): void + req_nextHour(): void + req_prevHour(): void + req_nextMinute(): void + req_prevMinute(): void + req_nextSecond(): void + req_prevSecond(): void

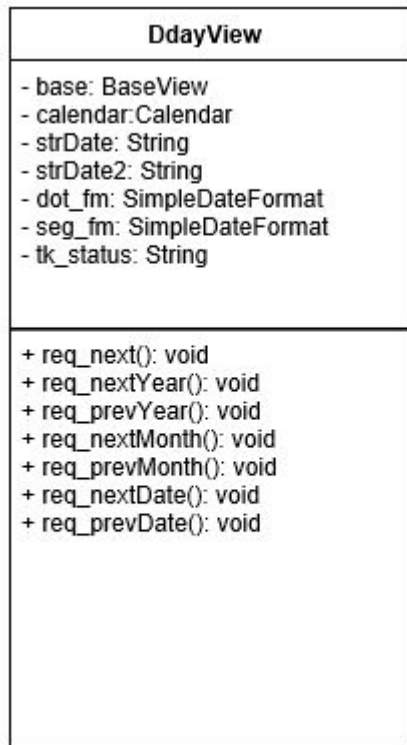
5-5) AlarmView Class



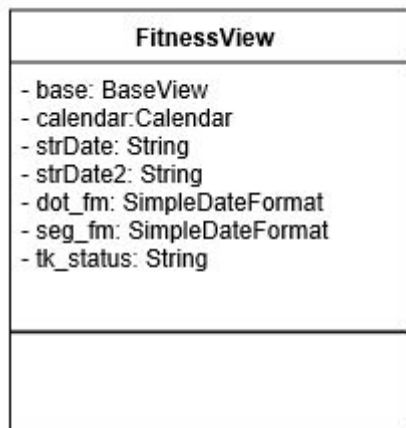
5-6) StopwatchView Class



5-7) DdayView Class



5-8) FitnessView Class



5-9) Controller Class

Controller

- instmanager: InstManager
- timekeeping: Timekeeping
- alarm: Alarm
- stopwatch: Stopwatch
- dDay: Dday
- fitness: Fitness
- selectFunc: SelectFunction
- stDTO: stopwatchDTO
- fitDTO: fitnessDTO

- dDayIndex: int
- alarmIndex: int
- alarmInstNum: int
- dDayInstNum: int
- currentFunc: int

+ req_countUp(): void
+ req_stopBuzzer(): void
+ req_changeMode(): int
+ req_lockFunc(): boolean
+ req_funcList(): void
+ req_selectFunc(index: int): boolean
+ req_finishSelect(): boolean

+ req_setDate(year: int, month: int, date: int): void
+ req_setTime(status: String, hour: int, minute: int, second: int): void

+ req_setTimer(): void
+ req_countDown(): void
+ req_pause(status: String): void
+ req_continue(): void
+ req_reset(): void

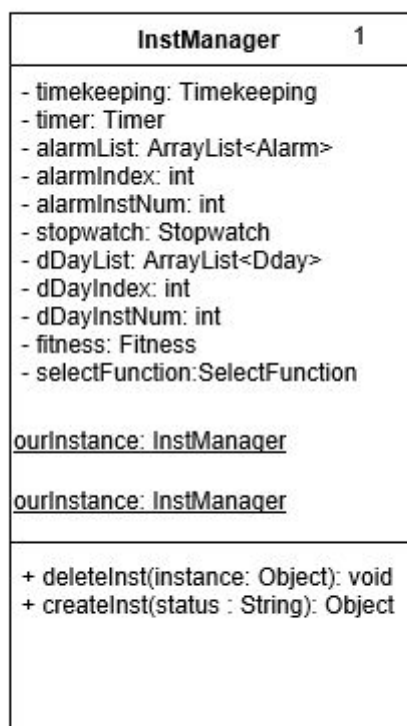
+ req_alarmList(): Alarm
+ req_setAlarm(): void
+ req_setDate(day: int, cycle: String, hour: int, minute: int): void
+ req_onOff(): boolean
+ req_delete(): void

+ req_record(): void
+ req_recordList(): void
+ req_finish(): void

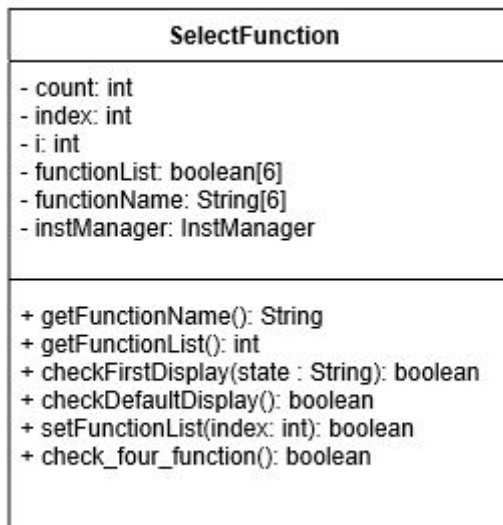
+ req_selectDate(): Object
+ req_nextGoal(status : String): String
+ req_setGoal(): void
+ req_DdayList(): void
+ req_deleteDday(): void

+ req_fitnessList():void
+ req_nextExercise(): String
+ req_setExercise(exercise : String): void

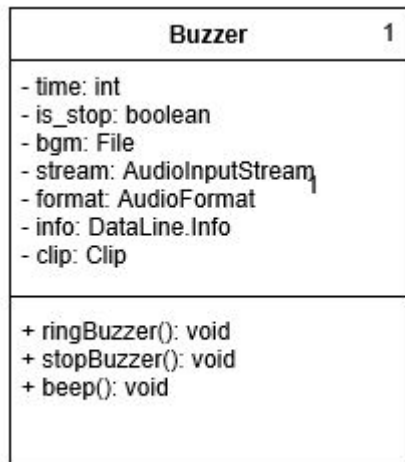
5-10) InstManager Class



5-11) SelectFunction Class



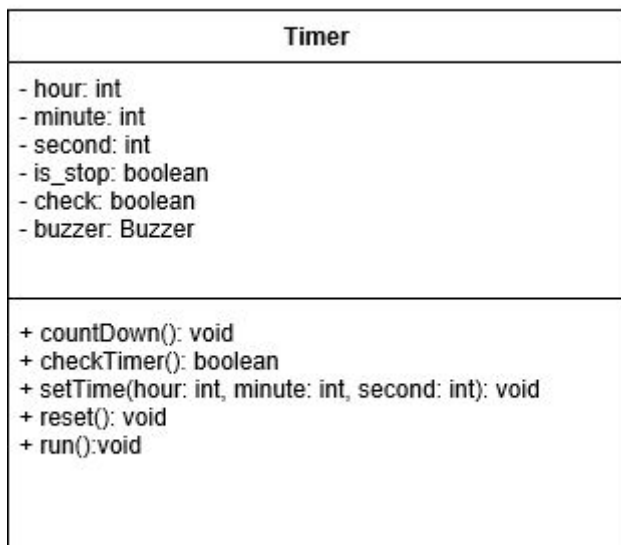
5-12) Buzzer Class



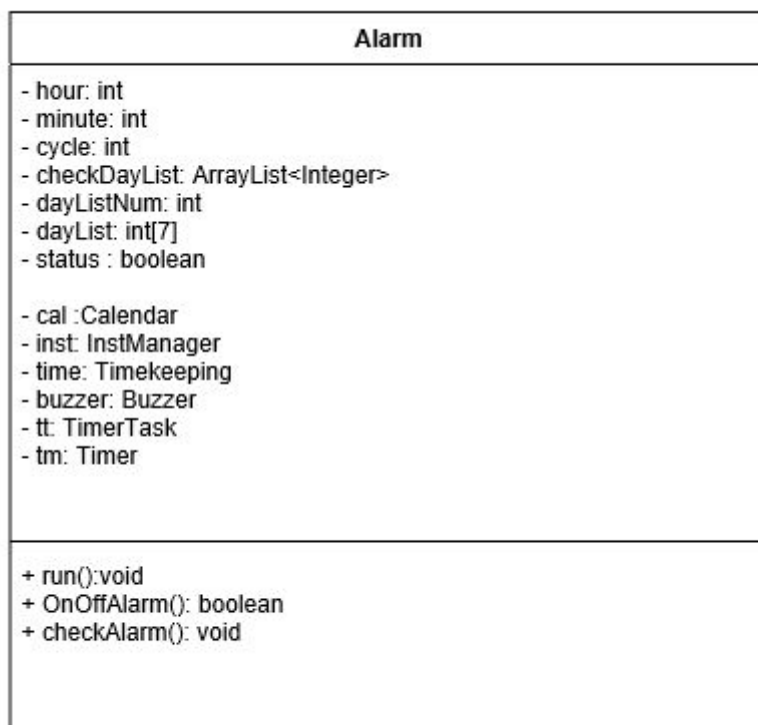
5-13) Timekeeping Class

Timekeeping
- year: int - month: int - date: int - hour: int - minute: int - second: int - dayNum: int - is_stop: int; - dayName:String[7]
+ setDate(year: int, month: int, date: int): void + setTime(hour: int, minute: int, second: int): void + countUp(): void + calculateDay(year: int, month: int, date: int): String

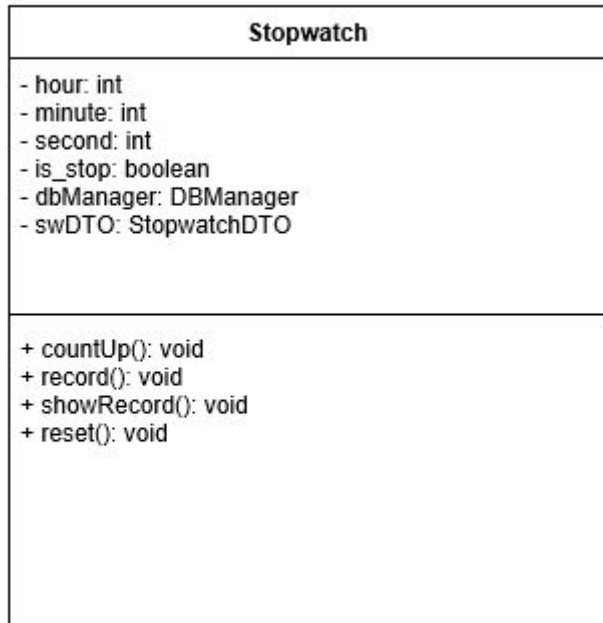
5-14) Timer Class



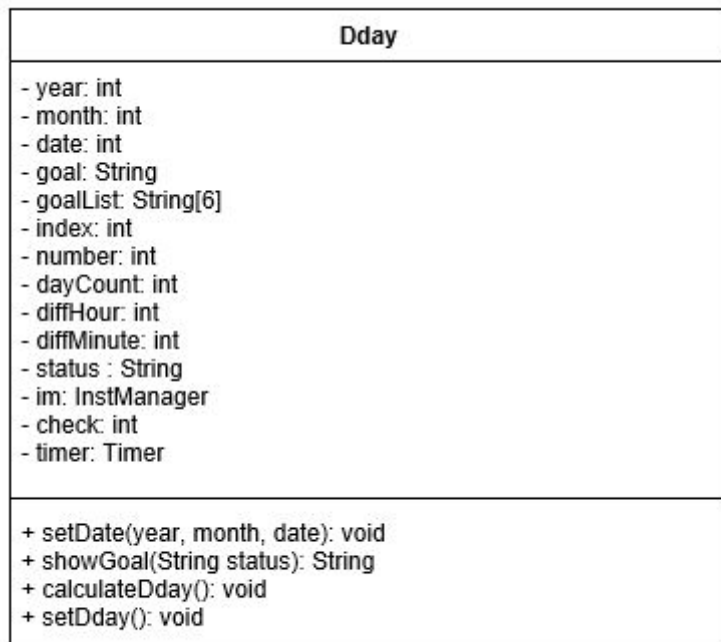
5-15) Alarm Class



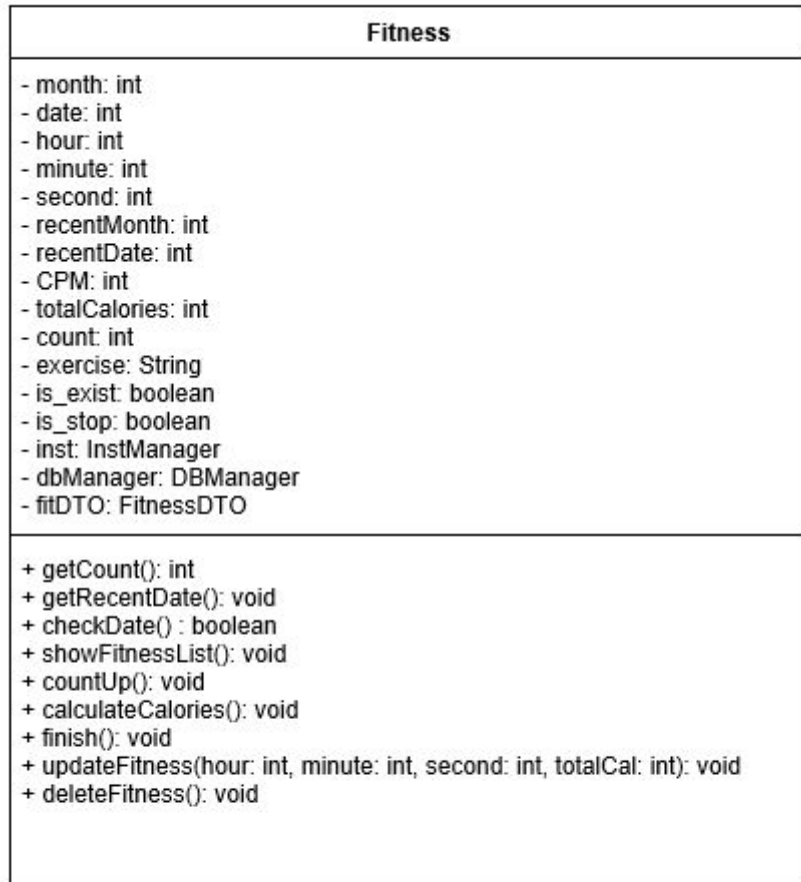
5-16) Stopwatch Class



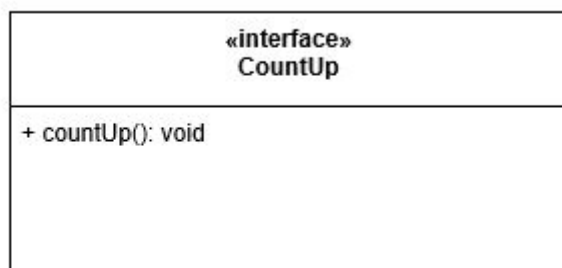
5-17) Dday Class



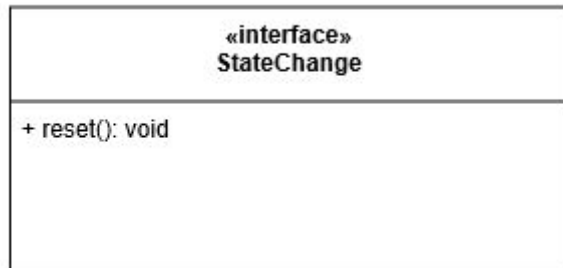
5-18) Fitness Class



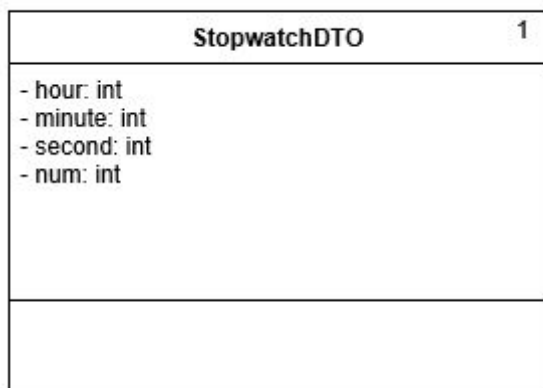
5-19) CountUp Interface



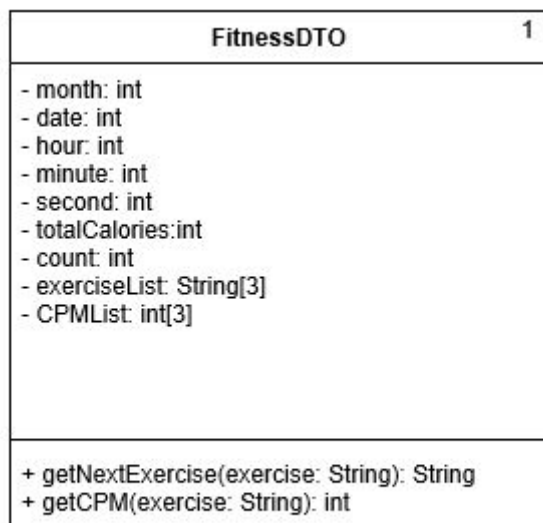
5-20) StateChange Interface



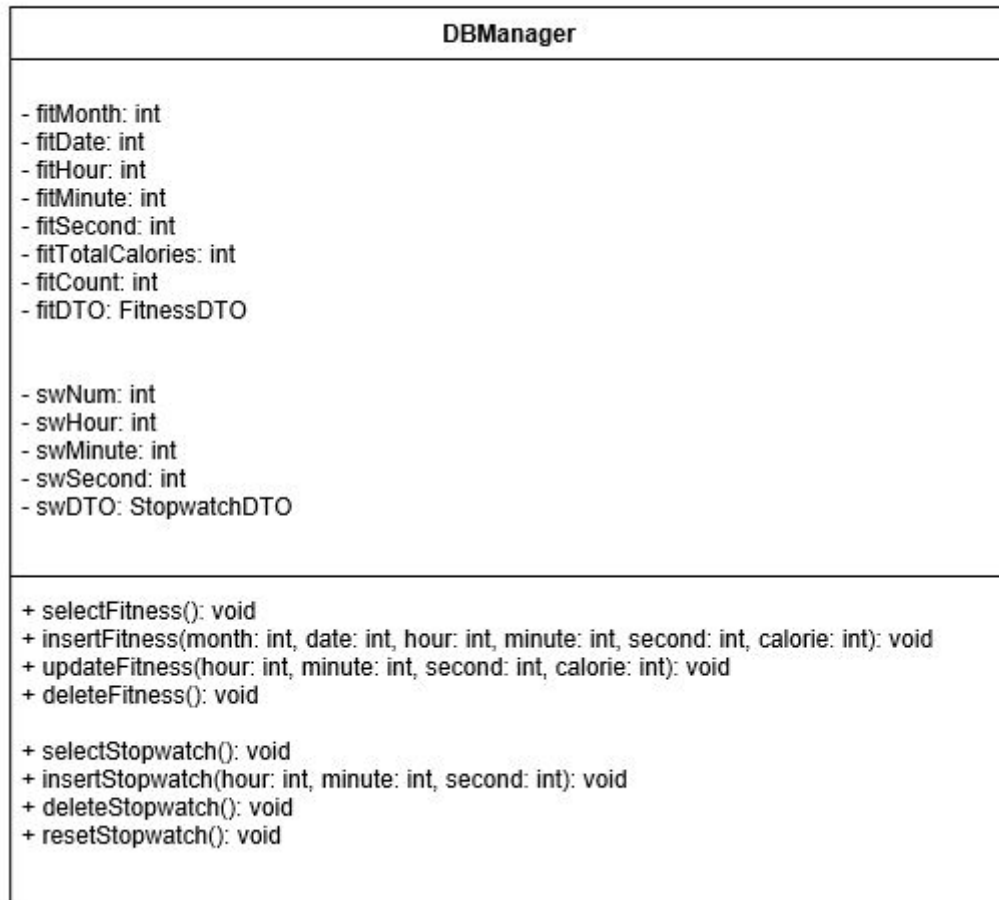
5-21) StopwatchDTO Class



5-22) FitnessDTO Class



5-23) DBManager Class



6. Activity 2146. Analyze Traceability Analysis

	Operation in Sequence diagram		Operation in Interaction diagram		Method
1	request_stopBuzzer()	5-4,6-1	req_funcList()	5-4,6-1	req_funcList()
2	request_countUp()	6- 2,8-1	req_selectFunc()	6- 2,8-1	req_selectFunc()
3	request_nextFunction()	6- 3,8-2	setFunction(index)	6- 3,8-2	setFunction(index)
4	request_funcList()	9-1	req_finishSelect()	9-1	req_finishSelect()
5	change_func()	7-2	check_four_function()	7-2	check_four_function()
6	set_func()	5-1	req_lookFunc()	5-1	req_lookFunc()
7	cancel_func()	5-2	checkFirstDisplay(status)	5-2	checkFirstDisplay(status)
8	exit_selectfunc()	5-3	checkDefaultDisplay(Controller)	5-3	checkDefaultDisplay(Controller)
9	request_currentTime()	5-5	getFuncionName()	5-5	getFuncionName()
10	change_currentTime()	9-2	check_four_function()	9-2	check_four_function()
11	set_currentTime()	4-1	req_changeMode()	4-1	req_changeMode()
12	exit_setTime()	4-2	getFunction()	4-2	getFunction()
13	chage_timerTime()	3-3	change_view()	3-3	change_view()
14	set_timerTime()	46-1,2-1	req_countUp(status)	46-1,2-1	req_countUp(status)
15	request_countDown()	16-2,17-2,47-2,48-2,28-2,29-2,32-2,2-2	setIs_stop()	16-2,17-2,47-2,48-2,28-2,29-2,32-2,2-2	setIs_stop()
16	pause_timer()	15-6,46-2,2-7	start()	15-6,46-2,2-7	start()
17	continue_timer()	46-3,29-3,32-3,2-6	countUp()	46-3,29-3,32-3,2-6	countUp()
18	cancel_timer()	9-4,16-3,17-3,18-3,42-3,2-3,47-3,48-3,28-3,2	getHour()	9-4,16-3,17-3,18-3,42-3,2-3,47-3,48-3,28-3,2	getHour()

		9-4,34-4,3 2-42-3		9-4,34-4 ,32-42-3	
1 9	exit_setTimer()	9-5,16-4,1 7-4,18-4,4 2-4,47-4,4 8-4,28-4,2 9-5,34-5,3 2-5,2-4	getMinute()	9-5,16-4 ,17-4,18 -4,42-4, 47-4,48- 4,28-4,2 9-5,34-5 ,32-5,2- 4	getMinute()
2 0	request_alarmList()	9-6,16-5,1 7-5,18-5,4 2-5,47-5,4 8-5,28-5,2 9-6,34-6,3 2-6,2-5	getSecond()	9-6,16-5 ,17-5,18 -5,42-5, 47-5,48- 5,28-5,2 9-6,34-6 ,32-6,2- 5	getSecond()
2 1	exit_setAlarmList()	24-4,1-1	req_stopbuzzer(status)	24-4,1-1	req_stopbuzzer(stat us)
2 2	change_alarmTime(2)	24-5,1-2	getBuzzer()	24-5,1-2	getBuzzer()
2 3	set_alarmTime()	24-6,1-3	stopBuzzer()	24-6,1-3	stopBuzzer()
2 4	exit_setAlarm()	9-1	getYear()	9-1	getYear()
2 5	request_onoff()	9-2, 42-1,49-8	getMonth()	9-2, 42-1,49- 8	getMonth()
2 6	delete_alarmList()	9-3,42-2,4 9-9	getDate()	9-3,42-2 ,49-9	getDate()
2 7	record_stopwatch()	10-1, 16-1,47-1, 28-1	req_pause()	10-1, 16-1,47- 1,28-1	req_pause()
2 8	pause_stopwatch()	10-2,13-11 ,33-2	req_nextYear()	10-2,13- 11,33-2	req_nextYear()
2 9	continue_stopwatch ()	10-3,13-2, 33-3	req_prevYear()	10-3,13- 2,33-3	req_prevYear()
3 0	request_stopwatchL ist()	10-4,13-3, 33-4	req_nextMonth()	10-4,13- 3,33-4	req_nextMonth()
3 1	reset_stopwatch()	10-5,13-4, 33-5	req_prevMonth()	10-5,13- 4,33-5	req_prevMonth()
3 2	continue_stopwatch ()				

3	change_DdayTime()	10-6, 13-5,33-6	req_nextDate()	10-6, 13-5,33-6	req_nextDate()
3	set_DdayTime()	10-7,13-6, 33-7	req_prevDate()	10-7,13-6,33-7	req_prevDate()
3	exit_selectDday()	10-8,13-7, 33-8	req_next()	10-8,13-7,33-8	req_next()
3	change_goal()	11-1,34-2	req_setDate(status, year,month,date)	11-1,34-2	req_setDate(status, year,month,date)
3	set_goal()				
3	exit_setGoal()	11-2,34-3, 35-1	setDate(year,month, date)	11-2,34-3,35-1	setDate(year,month, date)
3	request_DdayList()	12-1	calculateDay()	12-1	calculateDay()
4	delete_DdayList()		req_nextHour()		req_nextHour()
4	request_exerciseResultList()		req_prevHour()		req_prevHour()
4	exit_lookExercise()		req_nextMinute()		req_nextMinute()
4	request_exerciseList()		req_prevMinute()		req_prevMinute()
4	set_exercise()		req_nextSecond()		req_nextSecond()
4	exit_setExercise()		req_prevSecond()		req_prevSecond()
4	start_exercise()	14-1	req_setTime(status, hour,minute,second)	14-1	req_setTime(status, hour,minute,second)
4	pause_exercise()	19-1	setTime(hour,minute, second)	19-1	setTime(hour,minute ,second)
4	continue_exercise()	17-1,48-2, 29-1,32-1	req_continue()	17-1,48-2,29-1,32-1	req_continue()
4	finish_exercise()	15-5	ringBuzzer()	15-5	ringBuzzer()
		15-1	req_countDown()	15-1	req_countDown()
		15-2	countDown()	15-2	countDown()
		15-3	checkTlmer()	15-3	checkTlmer()
		15-4	ringBuzzer()	15-4	ringBuzzer()
		18-1	req_reset()	18-1	req_reset()
		18-2	reset()	18-2	reset()

		20-1	req_alarmList()		20-1	req_alarmList()
		20-2	getAlarmInstNum()		20-2	getAlarmInstNum()
		20-3	getAlarmIndex()		20-3	getAlarmIndex()
		20-4	setAlarmIndex()		20-4	setAlarmIndex()
		21-1	getAlarm()		21-1	getAlarm()
		22-1	req_setAlarm()		22-1	req_setAlarm()
		22-2	createInst(status)		22-2	createInst(status)
		22-3	req_nextDay()		22-3	req_nextDay()
		22-4	req_prevDay()		22-4	req_prevDay()
		22-5	req_nextCycle()		22-5	req_nextCycle()
		22-6	req_prevCycle()		22-6	req_prevCycle()
		23-1	req_setDate(checkDayList,dayListNum,cycle,hour,minute)		23-1	req_setDate(checkDayList,dayListNum,cycle,hour,minute)
		23-2	setDay(day)		23-2	setDay(day)
		23-3	setCycle(cycle)		23-3	setCycle(cycle)
		23-4	setHour(hour)		23-4	setHour(hour)
		23-5	setMinute(minute)		23-5	setMinute(minute)
		24-1	getTimeKeeping()		24-1	getTimeKeeping()
		24-2	getDay()		24-2	getDay()
		24-3	checkAlarm()		24-3	checkAlarm()
		25-1	req_onOff()		25-1	req_onOff()
		25-2	onOffAlarm()		25-2	onOffAlarm()
		26-1	req_delete()		26-1	req_delete()
		26-2,40-2	deleteInst(object)		26-2,40-2	deleteInst(object)
		27-1	req_record()		27-1	req_record()
		27-2	record()		27-2	record()
		27-3	deleteStopWatch()		27-3	deleteStopWatch()
		27-4	insertStopwatch(hour,minute,second)		27-4	insertStopwatch(hour,minute,second)
		27-5,30-3	selectStopwatch()		27-5,30-3	selectStopwatch()
		27-6	setSecond()		27-6	setSecond()
		30-1	req_recordList()		30-1	req_recordList()
		30-2	showRecord()		30-2	showRecord()
		49-1,31-2	req_finish(status)		49-1,31-2	req_finish(status)

		31-1	resetStopwatch()		31-1	resetStopwatch()
		33-1	req_selectDate()		33-1	req_selectDate()
		34-1	setDday()		34-1	setDday()
		36-1	req_nextGoal(status)		36-1	req_nextGoal(status)
		36-2	showGoal(status)		36-2	showGoal(status)
		37-1	req_setGoal(currGoal)		37-1	req_setGoal(currGoal)
		37-2	setGoal(currGoal)		37-2	setGoal(currGoal)
		38-1	calculateDay()		38-1	calculateDay()
		39-1	req_DdayList()		39-1	req_DdayList()
		39-2	getDdayInstNum()		39-2	getDdayInstNum()
		39-3	getDday()		39-3	getDday()
		39-4	getDayCount()		39-4	getDayCount()
		39-5	getGoal()		39-5	getGoal()
		40-1	req_deleteDday()		40-1	req_deleteDday()
		43-1	req_nextExercise()		43-1	req_nextExercise()
		43-2	getExercise()		43-2	getExercise()
		43-3	getNextExercise()		43-3	getNextExercise()
		44-1,45-1	req_setExercise()		44-1,45-1	req_setExercise()
		44-2,45-2	setExercise(exercise)		44-2,45-2	setExercise(exercise)
		44-3,45-3	setCPM(exercise)		44-3,45-3	setCPM(exercise)
		44-4,45-4	getCPM(exercise)		44-4,45-4	getCPM(exercise)
		46-4	calculateCalories()		46-4	calculateCalories()
		49-2	checkDate(currentMonth, currentDate)		49-2	checkDate(currentMonth, currentDate)
		49-3	getRecentDate()		49-3	getRecentDate()
		41-3	selectFitness()		41-3	selectFitness()
		41-4,49-10	setMonth()		41-4,49-10	setMonth()
		49-4	deleteFitness()		49-4	deleteFitness()
		49-5	insertFitness()		49-5	insertFitness()
		49-6	updateFitness(hour, minute, second, totalCalories)		49-6	updateFitness(hour, minute, second, totalCalories)
		49-7	initFitness()		49-7	initFitness()

		41-1	req_FitnessList()		41-1	req_FitnessList()
		41-2	showFitnessList()		41-2	showFitnessList()
		,49-11	setDate()		,49-11	setDate()
		41-6,49-12 ,27-7,30-4	setHour()		41-6,49-12,27-7, 30-4	setHour()
		41-7,49-13 ,27-8,30-5	setMinute()		41-7,49-13,27-8, 30-5	setMinute()
		41-8,49-14 ,30-6	setSecond()		41-8,49-14,30-6	setSecond()
		41-9,47-6, 48-6	setTotalCalories()		41-9,47-6,48-6	setTotalCalories()
		41-10	setCount()		41-10	setCount()
		42-6	getTotalCalories()		42-6	getTotalCalories()
		42-7	getCount()		42-7	getCount()
		49-2	finish()		49-2	finish()