

# Pillendar

-Design document-



Student No.	21912114
Name	구자현
E-mail	jahyungoo6979@gmail.com

## [ Revision history ]

Revision date	Version #	Description	Author
06/03/2023	1.0	First Draft	구자현

= Contents =

1. Introduction .....	4
2. Class diagram .....	4
3. Sequence diagram .....	9
4. State machine diagram .....	16
5. Implementation requirements .....	17
6. Glossary .....	17
7. References .....	17

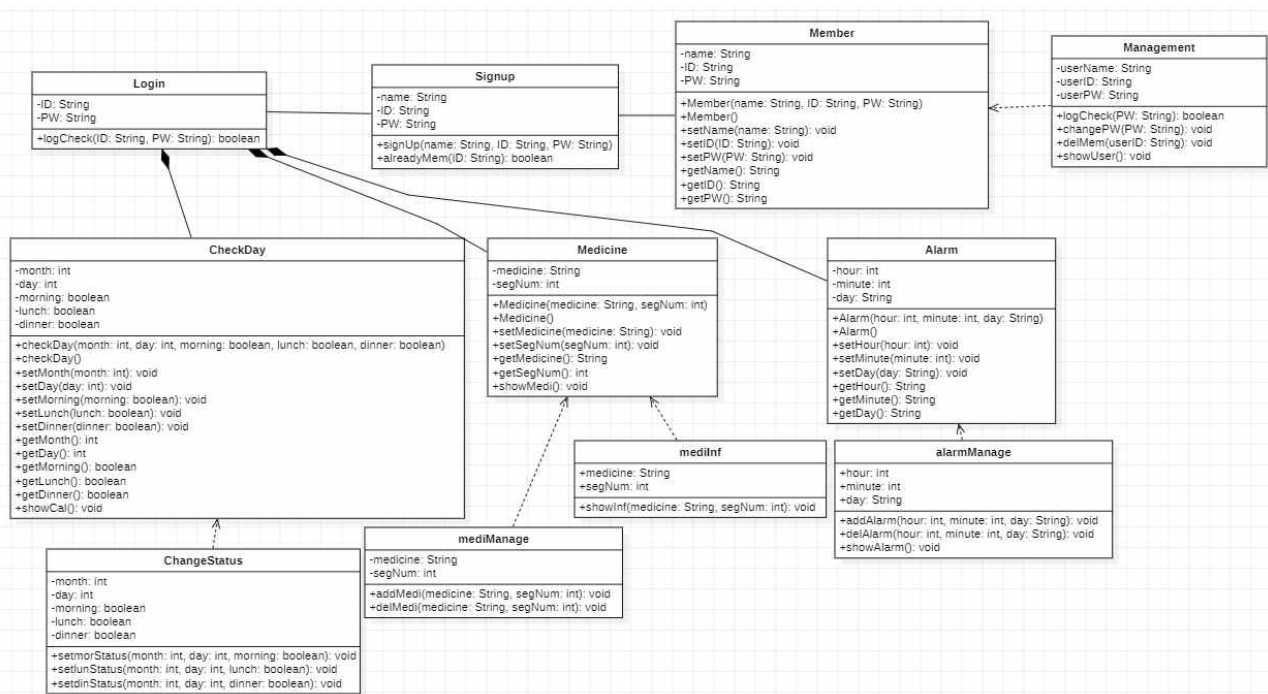
## 1. Introduction

### 1) Summary

현대사회에서 ‘덤벨 이코노미’라는 단어가 생길 정도로 건강, 운동에 대한 관심이 커졌다. 따라서, 많은 사람들이 비타민, 보충제를 찾거나 헬스를 통한 몸 관리를 시작하는 것을 볼 수 있다. 관리를 하는 만큼 최대한의 효율을 낼 수 있다면 정말 좋을 것이다. 하지만, 적절한 시기에 복용을 하지 못한다거나 잘못된 정보를 통한 복용으로 인해 효과를 제대로 보지 못하는 경우가 많다. 이런 경우에 시간, 돈만 낭비하게 될 가능성이 크다. 이러한 상황에 도움을 주는 것이 ‘Pillendar’이다. ‘Pillendar’는 복용하는 비타민, 약의 정보를 제공하고 설정해놓은 시간에 알람을 울려주면서 제시간에 복용을 할 수 있게 돕는다. 꾸준히 복용을 하는 것이 중요한데 ‘Pillendar’가 효과를 제대로 보게 도와줄 것이다.

이 문서는 ‘Pillendar’를 개발하기 위한 단계 중 세 번째 단계인 Design을 다룬 문서이다. Class Diagram, Sequence Diagram, State Machine Diagram을 통해 ‘Pillendar’의 전반적인 모습을 한눈에 볼 수 있기 때문에 이 시스템을 쉽게 이해할 수 있을 것이다.

## 2. Class diagram



Class Name	Explanation
Login	<p>로그인을 할 때 아이디와 비밀번호를 검사하는 클래스이다.</p> <p><b><u>Attributes</u></b></p> <p>-ID: String : 로그인할 때 입력받는 ID -PW: String : 로그인할 때 입력받는 비밀번호</p> <p><b><u>Methods</u></b></p> <p>+logCheck(ID: String, PW: String): boolean : 입력한 ID와 비밀번호가 등록된 회원의 정보와 맞는지 확인한다.</p>
SignUp	<p>회원가입을 할 때 사용하는 클래스이다.</p> <p><b><u>Attributes</u></b></p> <p>-name: String : 사용자의 이름 -ID: String : 사용할 ID -PW: String : 사용할 비밀번호</p> <p><b><u>Methods</u></b></p> <p>+signUp(name: String, ID: String, PW: String): 입력받은 이름, ID, 비밀번호로 회원가입을 한다. +alreadyMem(ID: String): boolean : 이미 가입한 회원인지 확인한다.</p>
Member	<p>회원 정보가 저장되는 클래스이다.</p> <p><b><u>Attributes</u></b></p> <p>-name: String : 사용자의 이름 -ID: String : 사용자의 ID -PW: String : 사용자의 비밀번호</p> <p><b><u>Methods</u></b></p> <p>+Member(name: String, ID: String, PW: String) : 생성자 +Member() : 생성자 +setName(name: String): void : 이름을 name으로 설정 +setID(ID: String): void : ID를 ID로 설정 +setPW(PW: String): void : 비밀번호를 PW로 설정 +getName(): String : 이름 얻어옴 +getID(): String : ID 얻어옴 +getPW(): String : PW 얻어옴</p>

Management	<p>회원들의 정보를 관리할 수 있는 클래스이다.</p> <p><b><u>Attributes</u></b></p> <ul style="list-style-type: none"> <li>-userName: String : 회원의 이름</li> <li>-userID: String : 회원의 ID</li> <li>-userPW: String : 회원의 비밀번호</li> </ul> <p><b><u>Methods</u></b></p> <ul style="list-style-type: none"> <li>+logCheck(PW: String): boolean : 관리자의 비밀번호가 맞는지 확인</li> <li>+changePW(PW: String): void : 회원의 비밀번호 변경</li> <li>+delMem(userID: String): void : 회원 정보 삭제</li> <li>+showUser(): void : 회원 목록 출력</li> </ul>
CheckDay	<p>회원이 약 혹은 비타민을 복용한 시기를 해당 날짜에 표시하는 클래스이다.</p> <p><b><u>Attributes</u></b></p> <ul style="list-style-type: none"> <li>-month: int : 달력의 월</li> <li>-day: int : 달력의 일</li> <li>-morning: boolean : 아침에 복용하였는지 표시를 위한 변수</li> <li>-lunch: boolean : 점심에 복용하였는지 표시를 위한 변수</li> <li>-dinner: boolean : 저녁에 복용하였는지 표시를 위한 변수</li> </ul> <p><b><u>Methods</u></b></p> <ul style="list-style-type: none"> <li>+checkDay(month: int, day: int, morning: boolean, lunch: boolean, dinner: boolean) : 생성자</li> <li>+checkDay() : 생성자</li> <li>+setMonth(month: int): void : 월을 month로 설정</li> <li>+setDay(day: int): void : 일을 day로 설정</li> <li>+setMorning(morning: boolean): void : 아침 표시 설정</li> <li>+setLunch(lunch: boolean): void : 점심 표시 설정</li> <li>+setDinner(dinner: boolean): void : 저녁 표시 설정</li> <li>+getMonth(): int : 월 얻어옴</li> <li>+getDay(): int : 일 얻어옴</li> <li>+getMorning(): boolean : 아침 표시 정보 얻어옴</li> <li>+getLunch(): boolean : 점심 표시 정보 얻어옴</li> <li>+getDinner(): boolean : 저녁 표시 정보 얻어옴</li> </ul>

	+showCal(): void : 달력 얻어옴 해당 날짜의 아침, 저녁, 저녁에 복용을 하였는지 각각 표시하거나 지우는 클래스이다.
ChangeStatus	<p><b><u>Attributes</u></b></p> -month: int : 달력의 월 -day: int : 달력의 일 -morning: boolean : 아침에 복용하였는지 표시를 위한 변수 -lunch: boolean : 점심에 복용하였는지 표시를 위한 변수 -dinner: boolean : 저녁에 복용하였는지 표시를 위한 변수
	<p><b><u>Methods</u></b></p> +setmorStatus(month: int, day: int, morning: boolean): void : 아침에 복용하였다고 표시한 것을 지우거나 표시한다. +setlunStatus(month: int, day: int, lunch: boolean): void : 점심에 복용하였다고 표시한 것을 지우거나 표시한다. +setdinStatus(month: int, day: int, dinner: boolean): void : 저녁에 복용하였다고 표시한 것을 지우거나 표시한다.
Medicine	복용 중인 약 혹은 비타민의 정보를 저장하는 클래스이다.
	<p><b><u>Attributes</u></b></p> -medicine: String : 복용 중인 약 혹은 비타민의 정보 -segNum: int : 복용 중인 것이 약인지 비타민인지 구분
	<p><b><u>Methods</u></b></p> +Medicine(medicine: String, segNum: int) : 생성자 +Medicine() : 생성자 +setMedicine(medicine: String): void : 복용 중인 것의 이름을 medicine으로 설정 +setSegNum(segNum: int): void : 복용 중인 것이 약이라면 1, 비타민이라면 0으로 설정 +getMedicine(): String : 복용 중인 것의 이름 불러옴 +getSegNum(): int : 복용 중인 것의 구분번호 불러옴 +showMedi(): void : 복용 중인 것들의 목록 출력
mediManage	복용 중인 약 혹은 비타민을 추가하거나 삭제하는 클래스이다.
	<p><b><u>Attributes</u></b></p> -medicine: String : 추가 삭제하려는 것의 이름

	-segNum: int : 추가 삭제하려는 것의 구분번호 <b><u>Methods</u></b> +addMedi(medicine: String, segNum: int): void : 약 혹은 비타민을 추가한다. +delMedi(medicine: String, segNum: int): void : 약 혹은 비타민을 삭제한다.
mediInf	복용 중인 약 혹은 비타민의 정보를 얻는 클래스이다. <b><u>Attributes</u></b> -medicine: String : 약 혹은 비타민의 이름 -segNum: int : 구분번호 <b><u>Methods</u></b> +showInf(medicine: String, segNum: int): void : 선택한 약 혹은 비타민의 정보를 보여준다.
Alarm	원하는 시간, 요일에 맞게 설정한 알람 목록을 저장하는 클래스이다. <b><u>Attributes</u></b> -hour: int : 시 -minute: int : 분 -day: String : 요일 <b><u>Methods</u></b> +Alarm(hour: int, minute: int, day: String) : 생성자 +Alarm() : 생성자 +setHour(hour: int): void : 시를 hour로 설정 +setMinute(minute: int): void : 분을 minute로 설정 +setDay(day: String): void : 요일을 day로 설정 +getHour(): String : 시를 불러옴 +getMinute(): String : 분을 불러옴 +getDay(): String : 요일을 불러옴
alarmManage	설정한 알람들을 관리하는 클래스이다. <b><u>Attributes</u></b> -hour: int : 시 -minute: int : 분 -day: String : 요일



### Methods

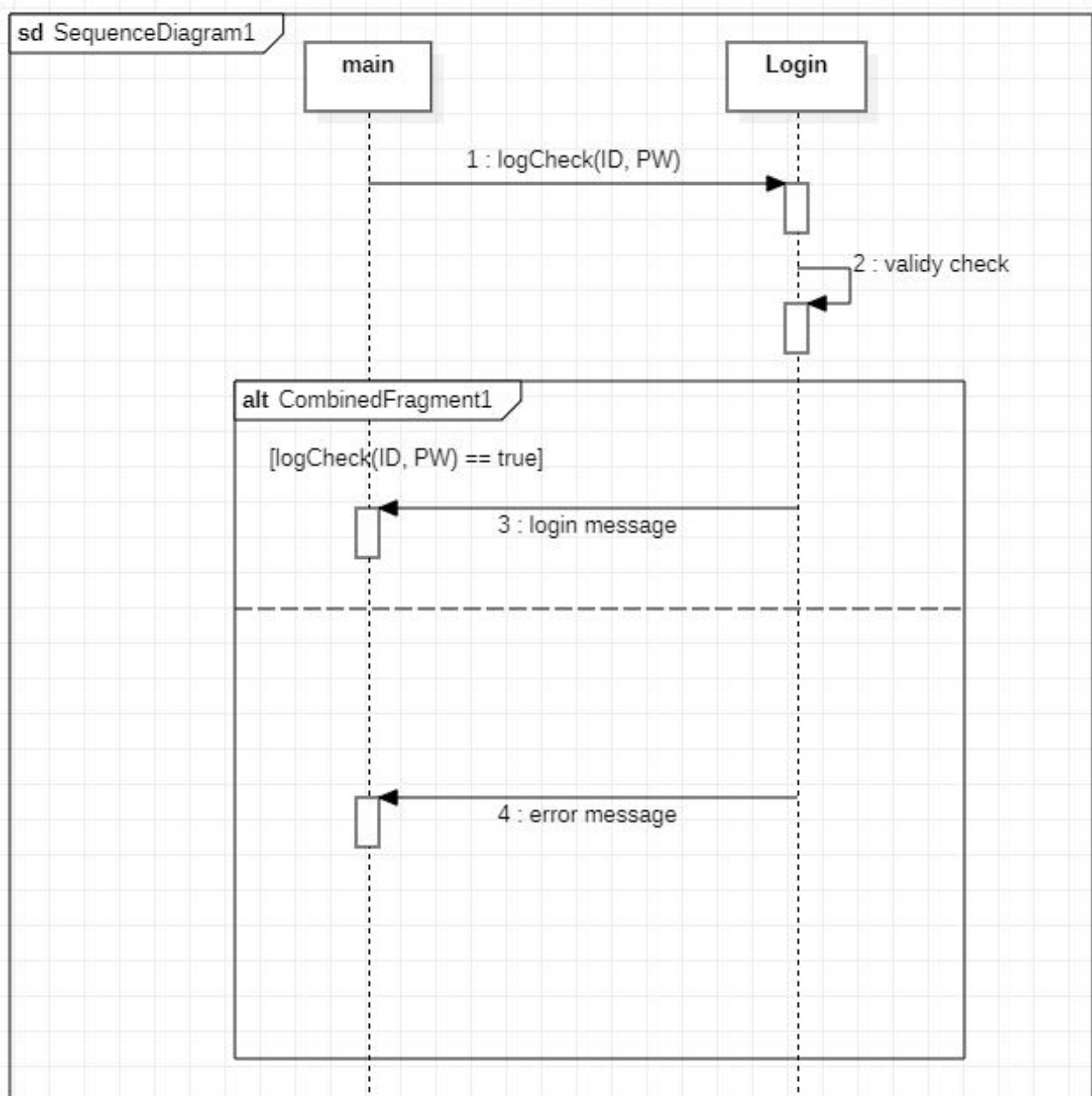
+addAlarm(hour: int, minute: int, day: String): void : 원하는 시간, 요일에 알람을 추가한다.

+delAlarm(hour: int, minute: int, day: String): void : 원하는 알람을 삭제한다.

+showAlarm(): void : 설정한 알람들을 출력

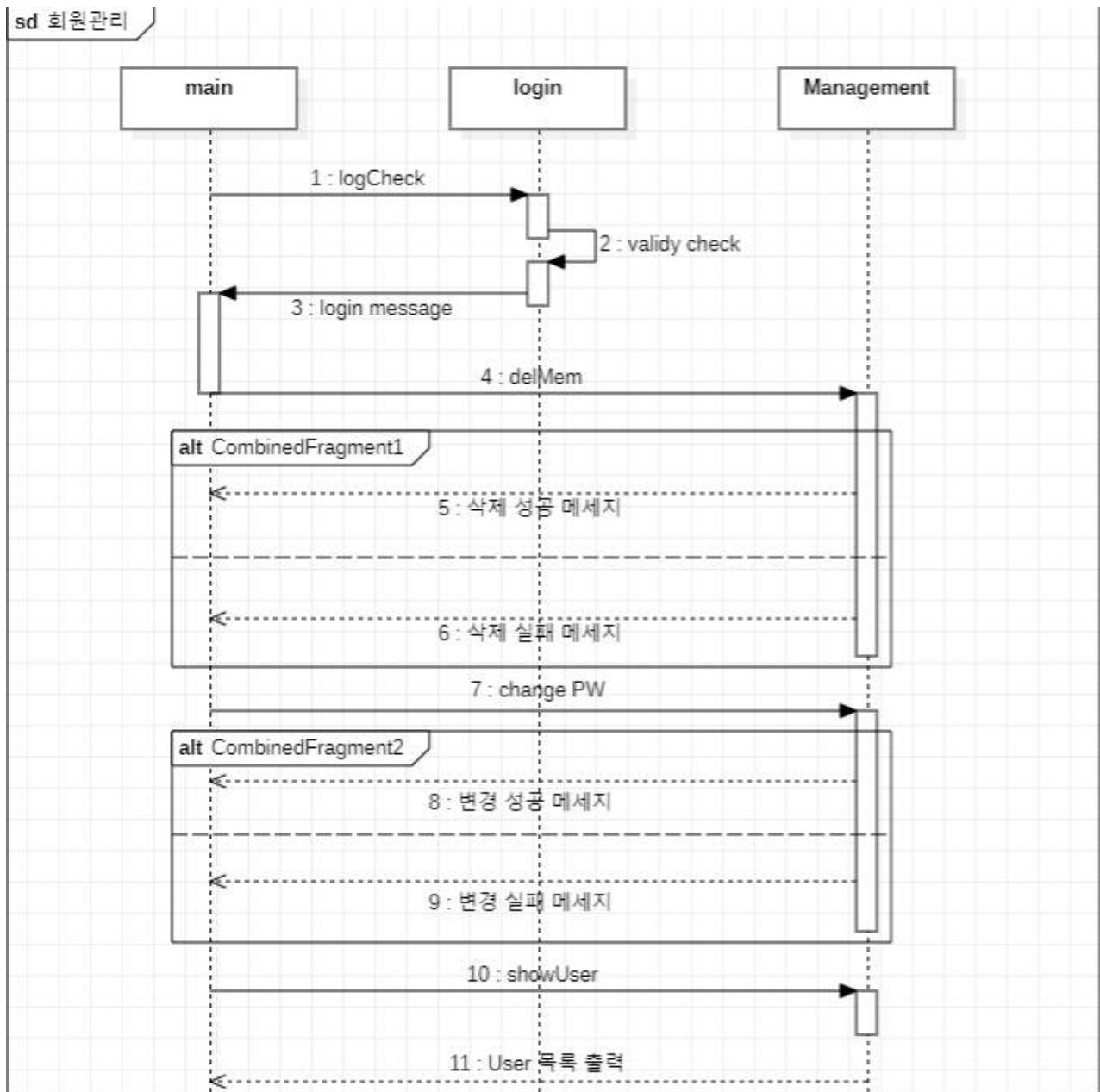
## 3. Sequence diagram

### 1) 로그인



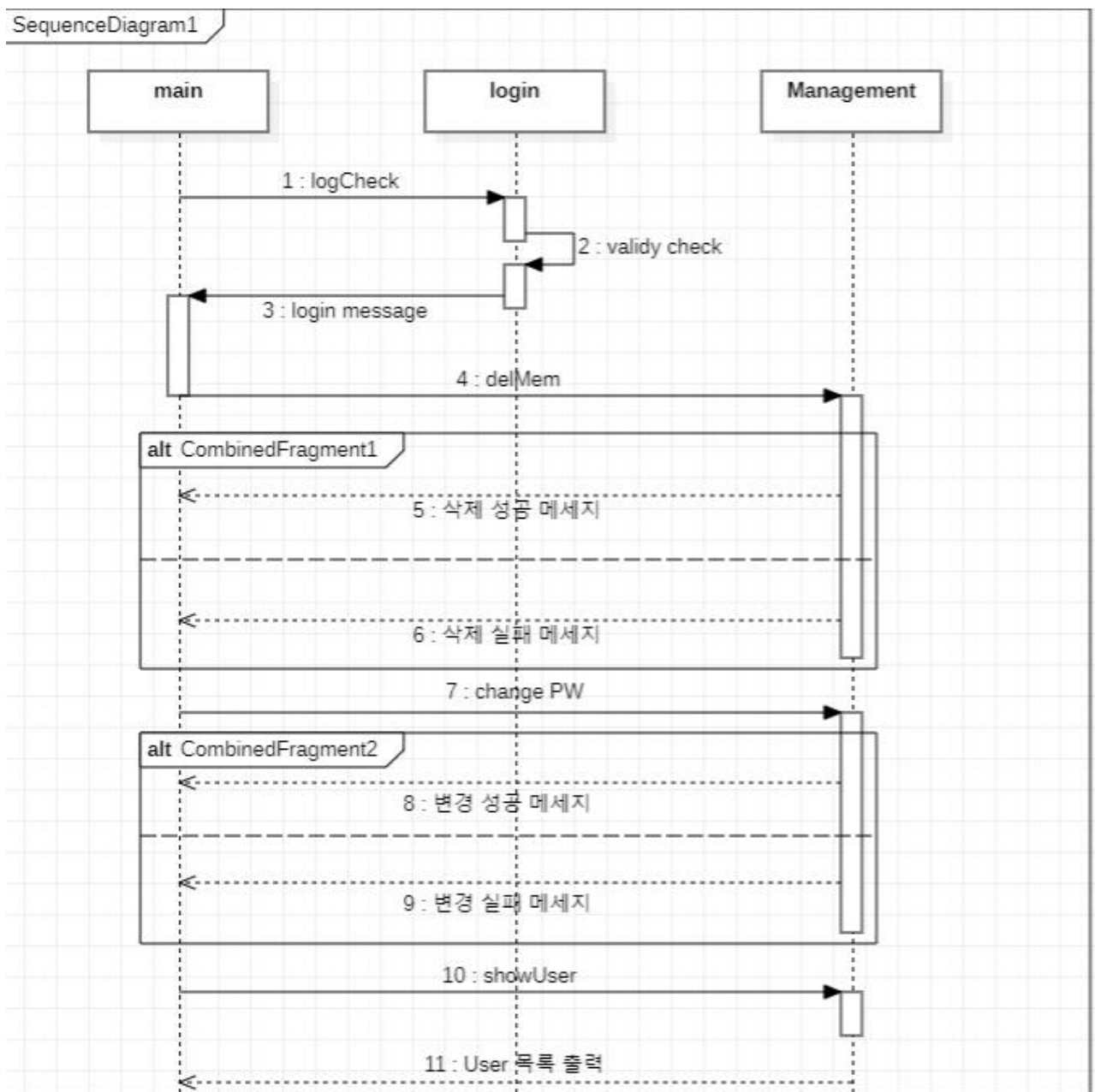
위 그림은 로그인 기능을 수행할 때의 Sequence Diagram이다. 아이디, 패스워드를 입력해서 logCheck 메소드를 부르고 입력한 아이디와 비밀번호가 등록된 회원정보와 맞는지 판단하여 잘못되었으면 메시지를 출력하고 일치한다면 true 값을 return 한다.

## 2) 회원가입



위 그림은 회원가입 기능을 수행할 때의 Sequence Diagram이다. 회원 정보를 입력하면 중복된 아이디가 있는지 확인하고 없다면 성공적으로 회원가입 기능을 수행한다.

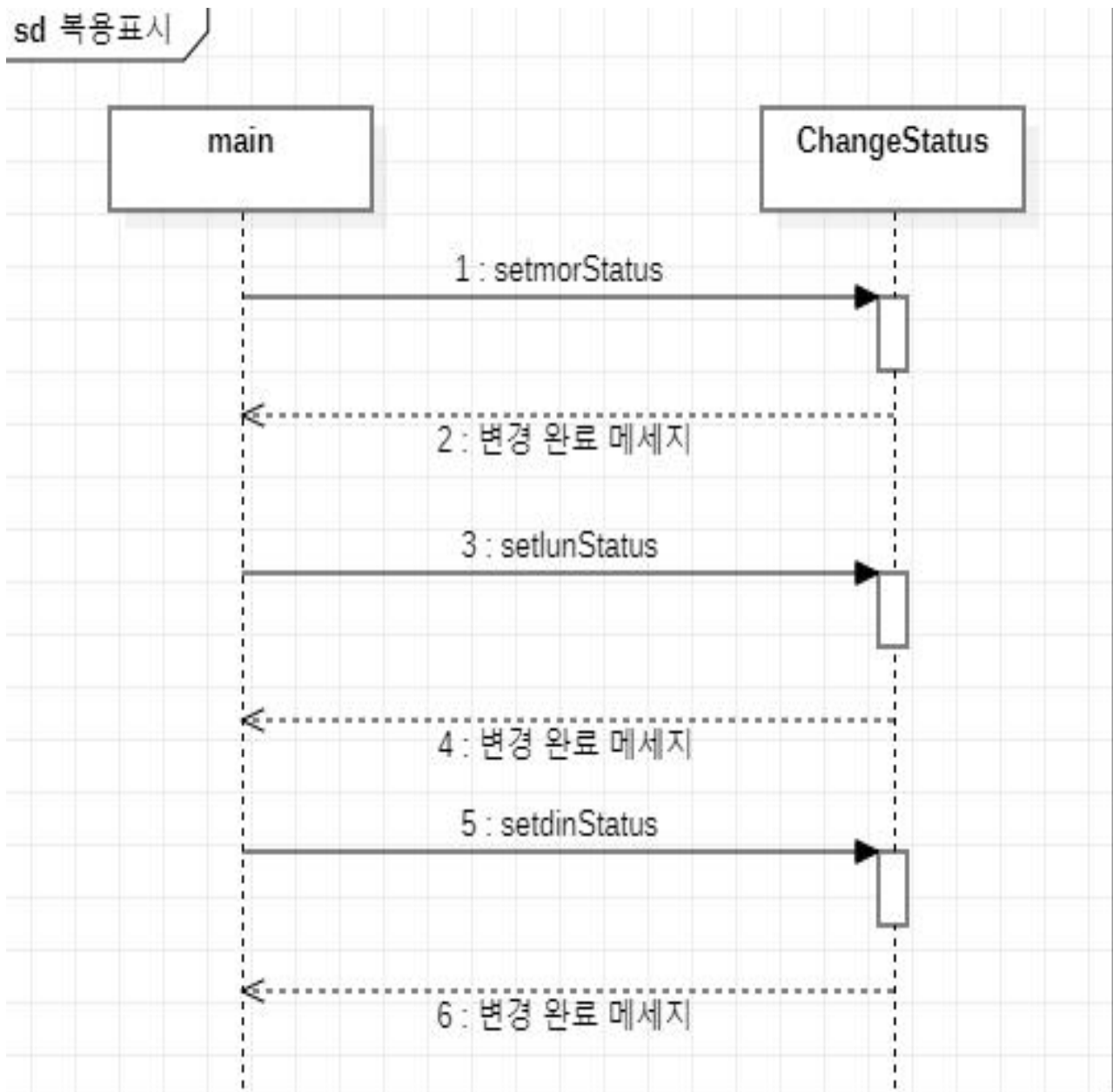
### 3) 회원관리



회원관리 기능을 수행할 때의 Sequence Diagram이다. 관리자 비밀번호와 일치하게 되면 관리자 신분으로 회원들의 정보를 관리할 수 있게 된다. 회원정보를 삭제하는 delMem, 회원의 비밀번호를 변경하는 changePW 기능을 수행 후 결과에 따라 메시지를 출력한다. showUser를 실행하면 모든 회원들의 정보를 출력한다.

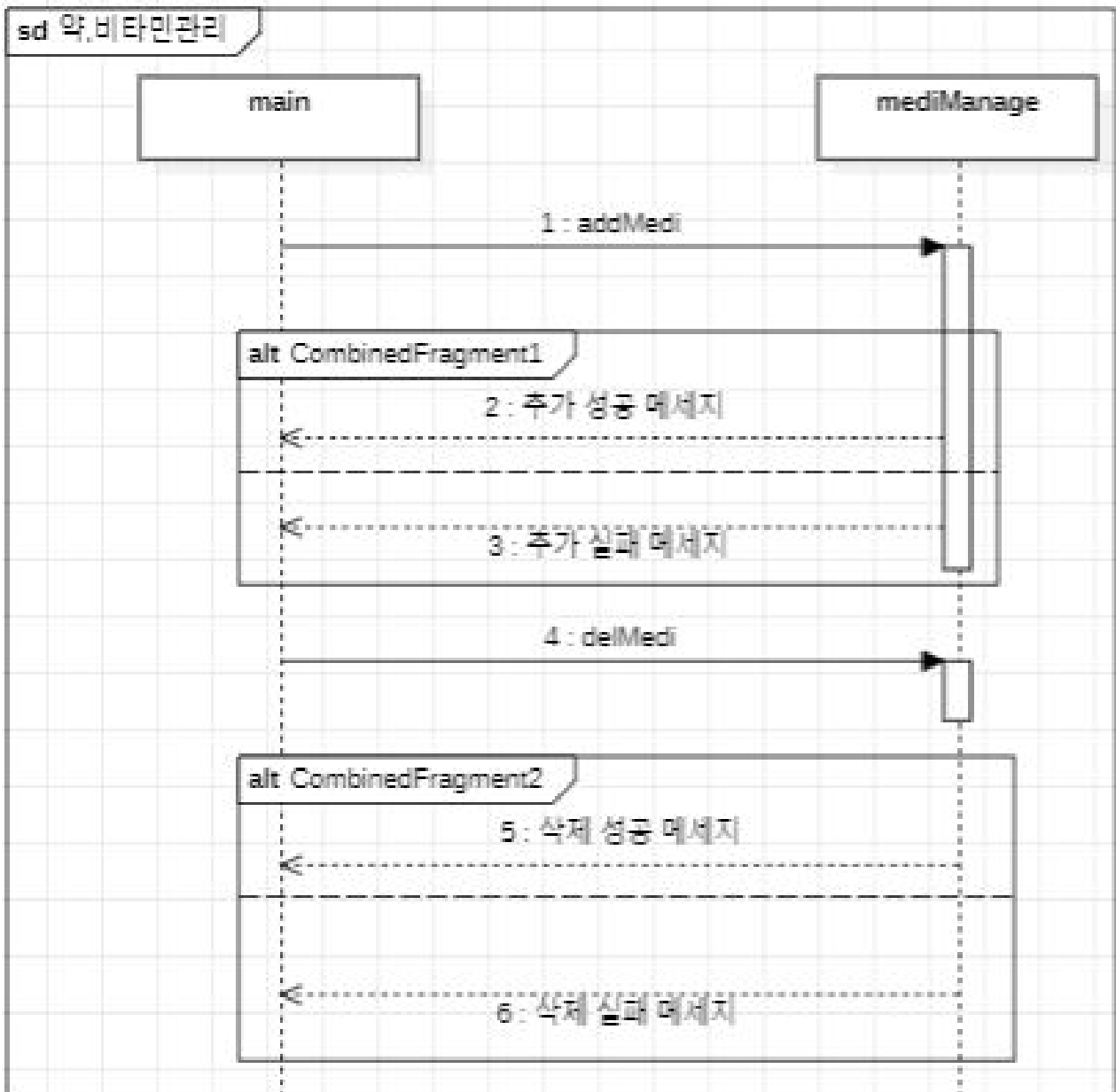
#### 4) 복용 표시

sd 복용표시



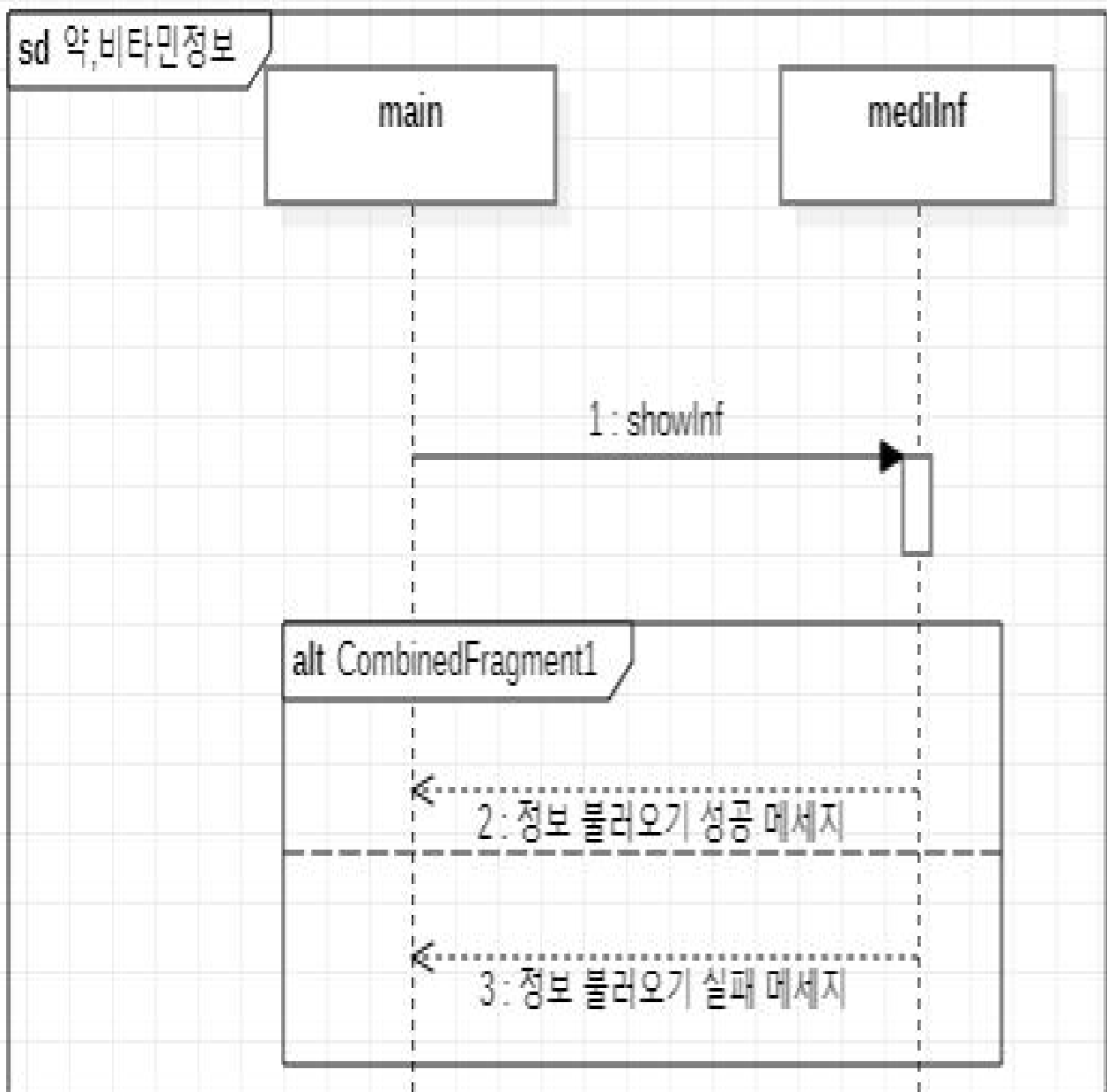
복용 완료 표시를 할 때의 Sequence Diagram이다. 아침에 복용 표시 및 표시 삭제를 하는 `setmorStatus`, 점심에 복용했다는 표시 및 표시 삭제를 하는 `setlunStatus`, 저녁에 복용했다는 표시 및 표시 삭제를 하는 `setdinStatus`를 각각 실행하고 변경 완료 메시지를 출력한다.

## 5) 약, 비타민 관리



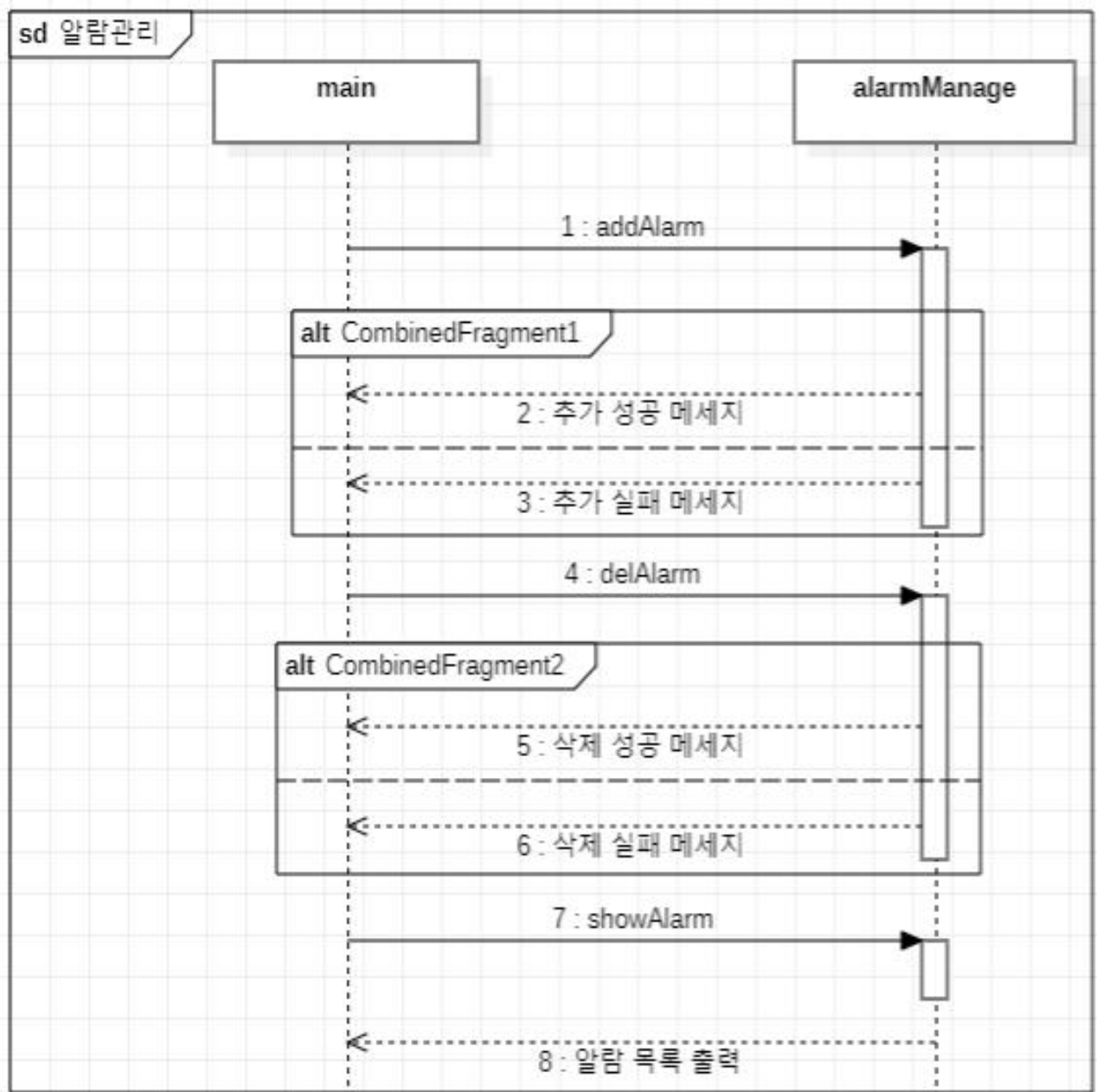
복용중인 약, 비타민 관리를 할 때의 Sequence Diagram이다. main에서 추가를 원하는 약, 비타민의 이름과 매개변수를 addMedi에 입력 후 결과를 메시지로 출력한다. 이때, 약은 매개변수 1, 비타민은 0으로 가정하고 추가를 해서 둘을 구분할 수 있게 한다. 삭제 기능을 하는 delMedi도 addMedi와 같은 순서로 실행된다.

## 6) 약, 비타민 정보



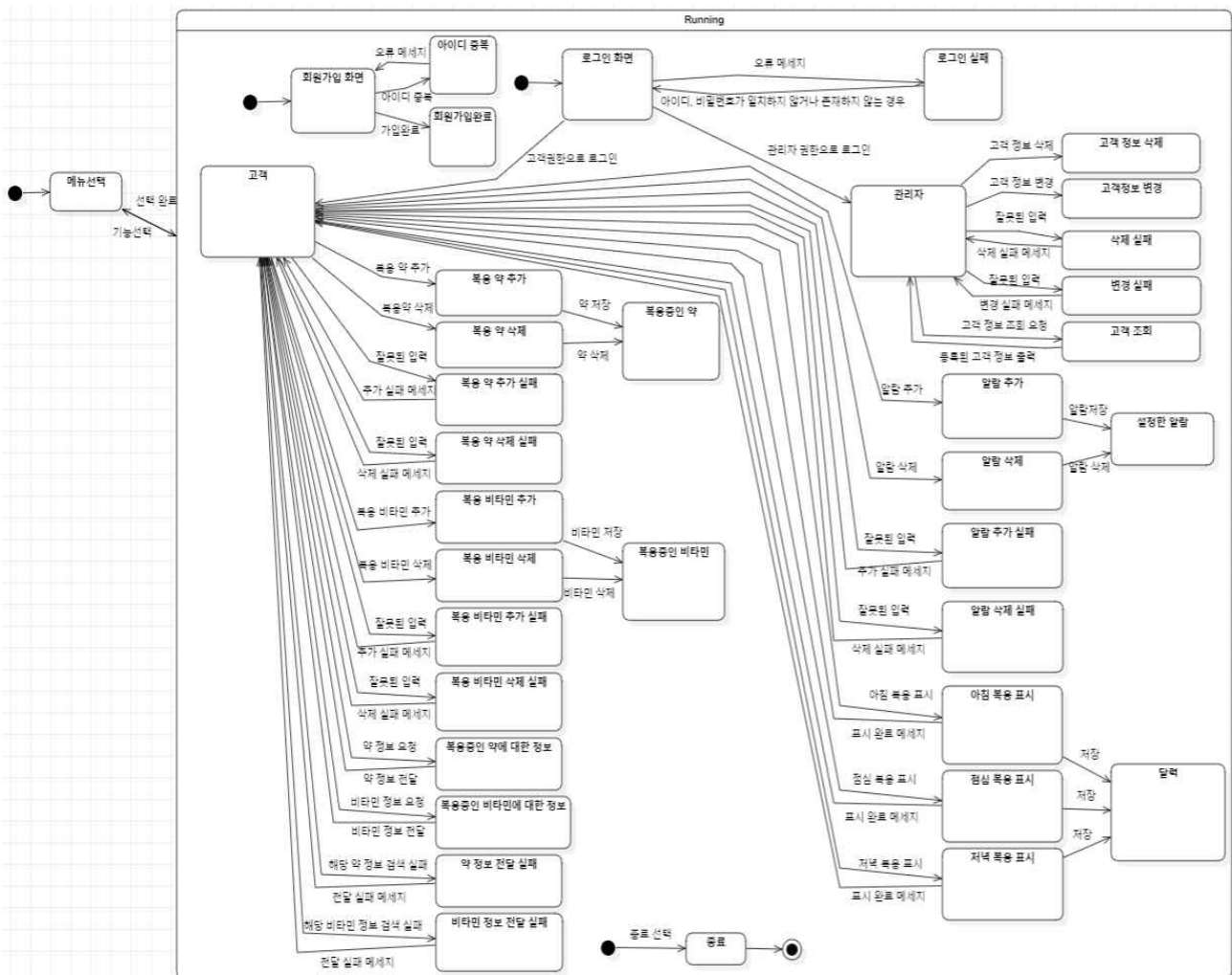
약 정보를 얻을 때의 Sequence Diagram이다. 복용 중인 약, 비타민을 복용할 때 알면 좋은 정보들을 얻기 위해 showInf를 실행하고 결과를 메시지로 출력한다.

## 7) 알람 관리



알람 관리를 할 때의 Sequence Diagram이다. 원하는 시간, 요일을 입력하고 addAlarm을 실행하면 알람을 설정할 수 있다. 삭제하려는 알람의 시간, 요일을 입력하고 delAlarm을 실행하면 삭제를 할 수 있다. 삭제할 알람이 없으면 삭제 실패 메시지가 출력 될 것이다. showAlarm을 실행하면 설정한 전체 알람 목록을 볼 수 있다.

#### 4. State machine diagram



회원 등록이 되어있다면 아이디와 비밀번호를 입력해 로그인하고 등록이 되어있지 않다면 회원가입을 한다. 고객 권한으로 로그인하면 복용 약, 비타민 관리, 복용 중 약, 비타민의 정보, 복용 표시, 알람 관리를 할 수 있다. 관리자 권한으로 로그인하면 고객 정보 변경, 삭제를 할 수 있다. 각 기능이 제대로 실행되지 않았다면 실패 메시지를 출력해서 사용자가 알 수 있게 한다.



## 5. Implementation requirements

### 1) H/W requirements

- (1) CPU : Intel Pentium IV 이상
- (2) RAM : 1GB 이상
- (3) HDD/ SSD : 5GB 이상

### 2) S/W requirements

- (1) OS: Microsoft Window 10 이상
- (2) Implementation Language: JAVA

## 6. Glossary

Terms	Description
Attribute	Class의 멤버 변수를 가리킨다.
Method	특정 기능을 정의한 코드들의 집합이다.
Class Diagram	클래스 내부 구성 요소, 클래스 간의 관계를 도식화하여 시스템의 구조를 보여주는 정적 구조 다이어그램이다.
Sequence Diagram	특정 행동이 객체와 어떻게 상호작용을 하는지 순서를 표현한 다이어그램이다.
State Machine Diagram	객체의 상태, 상태변화를 도식화하여 표현한 다이어그램이다.

## 7. References

- ① 아이콘 - <a href="https://www.flaticon.com/kr/free-icons/" title="캡슐 아이콘"> 캡슐 아이콘 제작자: Icongeek26 - Flaticon</a>