

2. SDS (Software Design Specification)

YUVID

백신 접종 관리 시스템



21621943	서완
21720903	조태식
21720913	박주홍
21813622	주정준
21910733	김소희
21913671	박효상

[Revision history]

Revision date	Version #	Description	Author
11-03-2021	0.01	First document	

= Contents =

1. Introduction	4
2. Use case analysis	5
3. Class diagram	9
4. Sequence diagram	39
5. State machine diagram.....	55
6. User interface prototype	57
7. Implementation requirements	67

1. Introduction

YUVID는 백신접종예약프로그램으로 현재 COVID-19의 예약이 가능하며, 부가적으로 COVID-19의 증상, 접종률 및 이미 접종을 완료한 사람들의 후기등 다양한 정보를 얻을 수 있다.

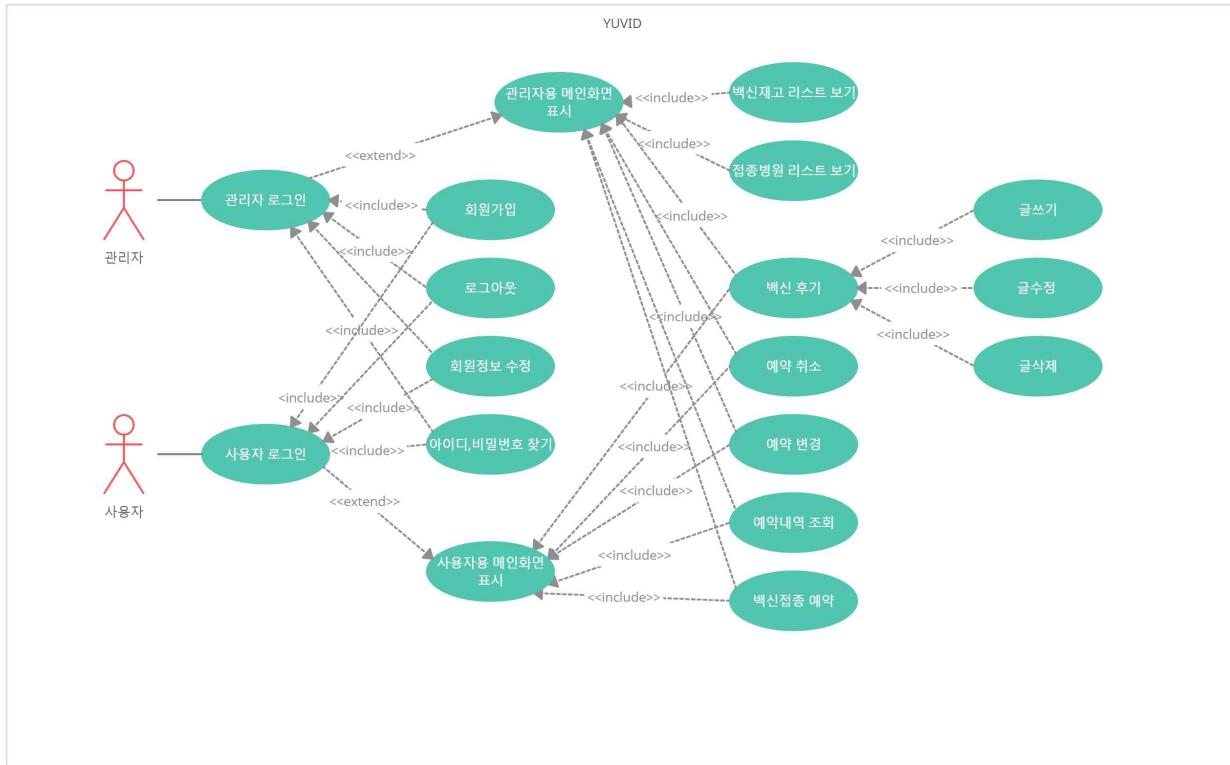
신종 바이러스의 위험성이 대두된 지금 빠르고 체계적인 접종을 통한 의료 인프라 구축을 위해 본 프로그램을 개발하게 되었다.

추후 COVID-19가 아닌 다른 바이러스의 유행이 발생했을 때에도 기존 시스템을 응용하여 백신 접종 예약을 빠르고 체계적으로 할 수 있다.

본 문서는 실제 구현에 관한 세부적 사항을 명시하고, 구체적으로 시스템의 구성에 대해 설명한다. 이를 위해 다음의 내용들이 본 문서에서 기술되어진다.

- Class Diagram
 - Class Diagram과, Class Diagram 들의 attribute, method 들의 상세 기술
- Sequence Diagram
 - Sequence Diagram 과, 각각의 Sequence Diagram 에 대한 상세 기술
- State Machine Diagram
 - State Machine Diagram과, State Machine Diagram 에 대한 상세 기술

2. Use case description



Use case #1 : 로그인	
GENERAL CHARACTERISTICS	
Summary	YUVID를 이용하기 위한 회원인증 기능
Scope	YUVID
Level	User level
Author	
Last Update	
Status	Analysis
Primary Actor	anyone
Preconditions	YUVID에 회원가입을 한 상태
Trigger	로그인하지 않은 사용자가 로그인 창에서 ID, PW를 입력한 후 로그인 버튼을 클릭할 때
Success Post Condition	로그인이 정상적으로 이루어지고 메인화면으로 이동한다.
Failed Post Condition	로그인이 성공적으로 되지 않는다.

MAIN SUCCESS SCENARIO	
Step	Action
S	사용자가 YUVID를 실행한다.
1	사용자가 YUVID의 메인화면에서 아이디와 비밀번호를 입력 후 로그인 버튼을 클릭한다.
2	시스템은 사용자가 회원가입이 되어있는지 DB에서 확인한다.
3	로그인이 성공하면 해당 Use case를 종료한다.

EXTENSION SCENARIOS	
Step	Branching Action
2	<p>2a. 아이디나 비밀번호가 잘못되었을 경우</p> <p>...2a1. 아이디나 비밀번호가 잘못되었다는 경고메시지를 보여준다.</p> <p>...2a2. 아이디와 비밀번호를 입력하는 단계로 돌아간다.</p> <p>2b. 아이디 입력 칸에 아이디가 입력되지 않은 경우</p> <p>...2b1. 아이디를 입력해달라는 경고 메시지를 보여준다.</p> <p>...2b2. 아이디를 입력하는 단계로 돌아간다.</p> <p>2c. 비밀번호 입력 칸에 비밀번호가 입력되지 않은 경우</p> <p>...2c1. 비밀번호를 입력해달라는 경고 메시지를 보여준다.</p> <p>...2c2. 비밀번호를 입력하는 단계로 돌아간다.</p>

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	회원당 하루에 평균 2번
<Concurrency>	
Due Date	

Use case #2 : 회원가입	
GENERAL CHARACTERISTICS	
Summary	YUVID를 이용하기 위한 회원가입
Scope	YUVID
Level	User level
Author	
Last Update	
Status	Analysis
Primary Actor	anyone
Preconditions	YUVID에 정상적으로 접속한 상태
Trigger	회원가입이 되지 않은 사용자가 로그인 창에서 회원가입 버튼을 클릭할 때
Success Post Condition	회원 정보가 DB에 저장되고 로그인 창으로 이동한다.
Failed Post Condition	회원 정보가 DB에 정상적으로 저장되지 않는다.

MAIN SUCCESS SCENARIO	
Step	Action
S	사용자가 YUVID를 실행한다.
1	사용자가 YUVID의 메인화면에서 회원가입 버튼을 클릭한다.
2	시스템은 회원가입 창으로 전환한다.
3	사용자는 ID, PW, 이름, 전화번호 등 필요한 정보를 작성한다.
4	시스템은 사용자가 회원가입 버튼을 클릭 시 모든 입력칸에 내용이 작성되었는지 확인한다.
5	시스템은 작성된 회원 정보를 DB에 저장한다.

EXTENSION SCENARIOS	
Step	Branching Action
3	<p>3a. 아이디 입력 시 DB에 저장되어 있는 아이디를 입력한 경우 ...3a1. 이미 존재하는 아이디라는 경고 메시지를 띠운다. ...3a2. 입력하던 회원 정보 입력 창으로 돌아간다.</p> <p>3b. 아이디 입력 칸에 한글을 입력한 경우 ...3b1. 아이디는 영문, 숫자만 입력 가능하다는 메시지를 띠운다. ...3b2. 입력하던 회원 정보 입력 창으로 돌아간다.</p> <p>3c. 비밀번호 입력 칸과 비밀번호 확인 칸에 입력한 비밀번호가 일치하지 않을 경우 ...3c1. 비밀번호 확인 칸에 입력한 비밀번호가 잘못되었다는 메시지를 띠운다. ...3c2. 입력하던 회원 정보 입력 창으로 돌아간다.</p> <p>3d. 회원 정보 입력 칸에 입력하지 않은 칸이 존재할 경우 ...3d1. 모든 항목을 채워달라는 메시지를 띠운다. ...3d2. 입력하던 회원 정보 입력 창으로 돌아간다.</p>

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	회원당 하루에 평균 2번
<Concurrency>	
Due Date	

Use case #3 : 아이디 찾기

GENERAL CHARACTERISTICS

Summary	사용자가 아이디를 분실하였을 경우 일부 개인 정보를 이용하여 아이디를 찾을 수 있다.
Scope	YUVID
Level	User level
Author	
Last Update	
Status	Analysis
Primary Actor	anyone
Preconditions	로그인하기 위한 ID를 분실한 상태
Trigger	로그인하지 못한 사용자가 ‘아이디 찾기’ 버튼을 클릭했을 때
Success Post Condition	사용자가 본인의 ID를 성공적으로 찾는다.
Failed Post Condition	사용자가 본인의 ID를 찾지 못한다.

MAIN SUCCESS SCENARIO

Step	Action
S	사용자가 YUVID를 실행한다.
1	사용자가 로그인 화면에서 ‘아이디 찾기’ 버튼을 클릭한다.
2	본인 식별을 위해 이름, 전화번호를 입력한다.
3	시스템은 이름, 전화번호가 정상적으로 입력되었는지 확인한다.
4	DB를 확인해 사용자가 입력한 정보와 동일한 정보가 있는지 확인한다.
5	사용자의 ID를 출력하고, 해당 Use case를 종료한다.

EXTENSION SCENARIOS

Step	Branching Action
3	3a. 회원 정보 입력 칸에 입력하지 않은 칸이 존재할 경우 ...3a1. 이름과 전화번호를 모두 입력해달라는 경고메시지를 보여준다. ...3a2. 아이디, 이름과 전화번호를 입력하는 단계로 돌아간다.
4	4a. 입력된 정보가 DB에 존재하지 않을 경우 ...4a1. 가입된 사용자 정보가 아니라는 경고메시지를 보여준다. ...4a2. 아이디, 이름과 전화번호를 입력하는 단계로 돌아간다.

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	회원당 하루에 평균 2번
<Concurrency>	
Due Date	

Use case #4 : 비밀번호 찾기	
GENERAL CHARACTERISTICS	
Summary	사용자가 비밀번호를 분실하였을 경우 일부 개인정보를 이용하여 비밀번호를 재설정할 수 있다.
Scope	YUVID
Level	User level
Author	
Last Update	
Status	Analysis
Primary Actor	anyone
Preconditions	로그인하기 위한 비밀번호를 분실한 상태
Trigger	로그인하지 못한 사용자가 ‘비밀번호 찾기’ 버튼을 클릭했을 때
Success Post Condition	사용자가 본인의 비밀번호를 성공적으로 재설정한다.
Failed Post Condition	사용자가 본인의 비밀번호를 찾지 못한다.

MAIN SUCCESS SCENARIO	
Step	Action
S	사용자가 YUVID를 실행한다.
1	사용자가 로그인 화면에서 ‘비밀번호 찾기’ 버튼을 클릭한다.
2	본인 식별을 위해 아이디, 이름, 전화번호를 입력한다.
3	시스템은 아이디에 대응하는 이름, 전화번호가 정상적으로 입력되었는지 확인한다.
4	DB를 확인해 사용자가 입력한 정보와 동일한 정보가 있는지 확인한다.
5	사용자의 비밀번호를 재설정하고, 해당 Use case를 종료한다.

EXTENSION SCENARIOS	
Step	Branching Action
3	3a. 회원 정보 입력 칸에 입력하지 않은 칸이 존재할 경우 ...3a1. 빈칸 없이 정보를 입력해달라는 경고메시지를 보여준다. ...3a2. 아이디, 이름과 전화번호를 입력하는 단계로 돌아간다.
4	4a. 입력된 정보가 DB에 존재하지 않을 경우 ...4a1. 가입된 사용자 정보가 아니라는 경고메시지를 보여준다. ...4a2. 아이디, 이름과 전화번호를 입력하는 단계로 돌아간다.

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	회원당 하루에 평균 2번
<Concurrency>	
Due Date	

Use case #5 : 회원정보 수정	
GENERAL CHARACTERISTICS	
Summary	사용자나 관리자는 아이디에 등록되어있는 전화번호나 비밀번호를 바꿀 수 있다.
Scope	YUVID
Level	User level
Author	
Last Update	
Status	Analysis
Primary Actor	anyone
Preconditions	YUVID에 로그인한 상태
Trigger	로그인 후 메인화면에서 ‘회원정보 수정’ 버튼을 클릭했을 때
Success Post Condition	사용자가 자신의 회원정보를 성공적으로 수정한다.
Failed Post Condition	사용자가 자신의 회원정보를 수정하지 못한다.

MAIN SUCCESS SCENARIO	
Step	Action
S	사용자가 YUVID에 로그인한다.
1	사용자가 메인화면에서 ‘회원정보 수정’ 버튼을 클릭한다.
2	사용자가 본인확인용 비밀번호 입력칸에 비밀번호를 기입한다.
3	사용자는 수정하고 싶은 정보를 입력한다.
4	사용자의 회원정보가 성공적으로 수정되면 Use case를 종료한다

EXTENSION SCENARIOS	
Step	Branching Action
2	2a1. 비밀번호 입력칸에 입력된 값이 없을 때 ...2a2. 비밀번호를 다시 입력해달라는 메시지를 출력한다. 2b1. 입력된 비밀번호가 사용자의 비밀번호와 다를 때 ...2b2. 잘못된 비밀번호라는 메시지를 출력한다.
	3a1. 비밀번호 재설정 시 비밀번호 확인창의 값이 틀릴 때 ...3a2. 새 비밀번호 입력칸과 확인칸의 비밀번호가 다르다는 에러 메시지를 출력한다. 3b1. 입력되지 않은 텍스트 박스가 있을 때 ...3b2. 기입하지 않은 정보가 있다는 에러 메시지를 출력한다.
RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	회원당 하루에 평균 2번
<Concurrency>	
Due Date	

Use case #6 : 백신 접종 예약	
GENERAL CHARACTERISTICS	
Summary	사용자가 접종하고자 하는 백신 종류와 접종 병원, 날짜에 맞게 예약할 수 있다.
Scope	YUVID
Level	User level
Author	
Last Update	
Status	Analysis
Primary Actor	anyone
Preconditions	YUVID에 로그인한 상태
Trigger	메인화면에서 ‘백신 접종 예약’ 버튼을 클릭했을 때
Success Post Condition	사용자가 백신 접종 예약을 성공적으로 한다.
Failed Post Condition	사용자가 예약하지 못한다.

MAIN SUCCESS SCENARIO	
Step	Action
S	사용자가 YUVID에 로그인한다.
1	사용자가 메인화면에서 ‘백신 접종예약’ 버튼을 클릭한다.
2	사용자는 ‘지역 선택 리스트’에서 접종 가능한 병원을 찾고, 접종하고자 하는 시간과 백신을 선택한다.
3	예약이 성공적으로 완료되면 해당 Use case를 종료한다.

EXTENSION SCENARIOS	
Step	Branching Action
2	<p>2a. 사용자가 접종하고자 하는 시간이나 백신을 선택하지 않고 ‘예약’ 버튼을 클릭한다.</p> <p>...2a1. 시간 혹은 백신을 선택해달라는 경고메시지를 출력한다.</p>

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	회원당 하루에 평균 2번
<Concurrency>	
Due Date	

Use case #7 : 예약 내역 조회	
GENERAL CHARACTERISTICS	
Summary	사용자가 접종 예약한 내역을 조회할 수 있다
Scope	YUVID
Level	User level
Author	
Last Update	
Status	Analysis
Primary Actor	anyone
Preconditions	백신 예약을 완료하고, YUVID에 로그인한 상태
Trigger	메인화면에서 ‘예약 내역 조회’ 버튼을 클릭했을 때
Success Post Condition	사용자가 자신의 예약 내역을 성공적으로 조회한다.
Failed Post Condition	사용자의 예약 내역이 출력되지 않는다.

MAIN SUCCESS SCENARIO	
Step	Action
S	사용자가 YUVID에 로그인한다.
1	사용자가 메인화면에서 ‘예약 내역 조회’ 버튼을 클릭한다.
2	DB에서 사용자의 예약 정보를 불러온다.
3	사용자에게 접종 일자, 시간, 접종병원 등을 표시해주고 해당 Use case를 종료한다.

EXTENSION SCENARIOS	
Step	Branching Action
2	2a. 사용자의 예약 정보가 확인되지 않을 때 ...2a1. 접종 예약을 하지 않았다는 메시지를 출력한다. ...2a2. 백신접종 예약 페이지로 이동한다.

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	회원당 하루에 평균 1번
<Concurrency>	
Due Date	

Use case #8 : 예약 변경	
GENERAL CHARACTERISTICS	
Summary	사용자가 접종 예약한 내역을 변경할 수 있다.
Scope	YUVID
Level	User level
Author	
Last Update	
Status	Analysis
Primary Actor	anyone
Preconditions	YUVID에 로그인한 상태
Trigger	메인화면에서 ‘예약 변경’ 버튼을 클릭했을 때
Success Post Condition	사용자가 자신의 예약 일정을 성공적으로 변경한다.
Failed Post Condition	사용자가 예약 일정을 변경하지 못한다.

MAIN SUCCESS SCENARIO	
Step	Action
S	사용자가 YUVID에 로그인한다.
1	사용자가 메인화면에서 ‘예약 변경’ 버튼을 클릭한다.
2	사용자가 본인확인을 위해 비밀번호를 입력한다.
3	사용자가 지역 리스트에서 병원과 접종시간을 선택한다.
4	사용자의 예약이 성공적으로 변경되면 해당 Use case를 종료한다.

EXTENSION SCENARIOS	
Step	Branching Action
2	<p>2a. 사용자의 예약이 확인되지 않을 때</p> <p>...2a2. 접종 예약을 하지 않았다는 메시지를 출력한다.</p> <p>...2a3. ‘백신 접종 예약’ 페이지로 이동한다.</p> <p>3b. 비밀번호 입력칸에 입력이 되지 않았을 때</p> <p>...3b1. 비밀번호를 다시 입력해달라는 메시지를 출력한다.</p>

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	회원당 하루에 평균 1번
<Concurrency>	
Due Date	

Use case #9 : 예약 취소	
GENERAL CHARACTERISTICS	
Summary	사용자가 접종 예약한 내역을 취소할 수 있다.
Scope	YUVID
Level	User level
Author	
Last Update	
Status	Analysis
Primary Actor	anyone
Preconditions	YUVID에 로그인하였고, 백신을 예약한 상태
Trigger	메인화면에서 ‘예약 취소’ 버튼을 클릭했을 때
Success Post Condition	사용자가 자신의 백신 접종 예약을 성공적으로 취소한다.
Failed Post Condition	사용자가 자신의 백신 접종 예약을 취소하지 못한다.

MAIN SUCCESS SCENARIO	
Step	Action
S	사용자가 YUVID에 로그인한다.
1	사용자가 메인화면에서 ‘예약 취소’ 버튼을 클릭한다.
2	사용자가 본인확인을 위한 비밀번호 입력 칸을 입력한다.
3	비밀번호 입력 후 나오는 예약 취소 버튼과 확인 메시지를 확인한다.
4	사용자의 예약이 성공적으로 취소되면 해당 Use case를 종료한다.

EXTENSION SCENARIOS	
Step	Branching Action
2	<p>2a. 사용자의 예약이 확인되지 않을 때</p> <p>...2a2. 접종 예약을 하지 않았다는 메시지를 출력한다.</p> <p>...2a3. ‘백신 접종 예약’ 페이지로 이동한다.</p> <p>3b. 비밀번호 입력 칸에 입력이 되지 않았을 때</p> <p>...3b1. 비밀번호를 다시 입력해달라는 메시지를 출력한다.</p>

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	회원당 하루에 평균 1번
<Concurrency>	
Due Date	

Use case #10 : 접종 병원 리스트	
GENERAL CHARACTERISTICS	
Summary	관리자가 사용자가 예약할 수 있는 병원을 리스트에 관리한다.
Scope	YUVID
Level	User level
Author	
Last Update	
Status	Analysis
Primary Actor	Administrator
Preconditions	관리자가 YUVID에 로그인한다.
Trigger	관리자로 로그인 후 메인화면에서 ‘병원 리스트 수정’ 버튼을 클릭했을 때
Success Post Condition	관리자가 접종병원을 성공적으로 추가한다.

Failed Post Condition		관리자가 접종병원을 추가하지 못한다.		
MAIN SUCCESS SCENARIO				
Step	Action			
S	관리자가 YUVID에 로그인한다.			
1	관리자가 메인화면에서 ‘접종병원 리스트’ 버튼을 클릭한다.			
2	관리자는 지역을 선택하고 해당 지역의 접종병원을 확인하고 Use case를 종료한다.			
EXTENSION SCENARIOS				
Step	Branching Action			
2	2a1. 지역을 선택하지 않았을 때 ...2a2. 지역을 선택하지 않았다는 에러 메시지를 출력한다. 2b1. 병원명 입력칸에 입력을 하지 않았을 경우 ...2b2. 병원명을 입력해달라는 에러 메시지를 출력한다.			
RELATED INFORMATION				
Performance	≤ 2 seconds			
Frequency	회원당 하루에 평균 1번			
<Concurrency>				
Due Date				
Use case #11 : 백신 재고 리스트				
GENERAL CHARACTERISTICS				
Summary	관리자가 사용자가 예약할 수 있도록 백신 재고를 관리한다.			
Scope	YUVID			
Level	User level			
Author				
Last Update				
Status	Analysis			
Primary Actor	Administrator			
Preconditions	관리자가 YUVID에 로그인한다.			
Trigger	관리자로 로그인 후 메인화면에서 ‘백신 재고리스트 수정’ 버튼을 클릭했을 때			
Success Post Condition	관리자가 백신 재고 리스트를 성공적으로 확인한다.			
Failed Post Condition	관리자가 백신 재고 리스트를 확인하지 못한다.			

MAIN SUCCESS SCENARIO	
Step	Action
S	관리자가 YUVID에 로그인한다.
1	관리자가 메인화면에서 ‘백신 재고 리스트’ 버튼을 클릭한다.
2	관리자는 병원을 선택하고 해당 병원의 접종병원을 확인하고 Use case를 종료한다.

EXTENSION SCENARIOS	
Step	Branching Action
2	2a1. 병원을 선택하지 않았을 때 ...2a2. 병원을 선택하지 않았다는 에러 메시지를 출력한다. 2b1. 병원명 입력칸에 입력을 하지 않았을 경우 ...2b2. 병원명을 입력해달라는 에러 메시지 창을 출력한다.

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	회원당 하루에 평균 1번
<Concurrency>	
Due Date	

Use case #12 : 백신 종류 리스트	
GENERAL CHARACTERISTICS	
Summary	관리자가 백신 재고를 추가할 때 이용할 백신 종류를 관리한다.
Scope	YUVID
Level	User level
Author	
Last Update	
Status	Analysis
Primary Actor	Administrator
Preconditions	관리자가 YUVID에 로그인한다.
Trigger	백신 재고 리스트 수정 창에서 ‘백신 종류 추가’ 버튼을 클릭했을 때
Success Post Condition	관리자가 백신 종류를 성공적으로 추가한다.
Failed Post Condition	관리자가 백신 종류를 추가하는데 실패한다.

MAIN SUCCESS SCENARIO	
Step	Action
S	관리자가 YUVID에 로그인한다.
1	관리자가 메인화면에서 ‘백신 재고리스트 수정’ 버튼을 클릭한다.
2	백신 종류 추가 버튼을 클릭한다.
2	관리자는 백신 종류의 이름을 입력하고 추가한 뒤 정상적으로 추가가 완료되면 Use case를 종료한다.

EXTENSION SCENARIOS	
Step	Branching Action
2	2a1. 백신 종류 이름에 숫자나 영어를 입력했을 경우 ...2a2. 한글만 입력 가능하다는 메시지를 띠운다.

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	회원당 하루에 평균 1번
<Concurrency>	
Due Date	

Use case #13 : 백신 후기 게시판	
GENERAL CHARACTERISTICS	
Summary	백신 종류에 따른 부작용들 및 다양한 정보를 서로 공유 할 수 있는 게시판이다.
Scope	YUVID
Level	User level
Author	
Last Update	
Status	Analysis
Primary Actor	anyone
Preconditions	사용자가 YUVID에 로그인한다.
Trigger	메인화면에서 ‘게시판’ 버튼을 클릭했을 때
Success Post Condition	사용자가 다른 사용자들이 기록한 백신 후기를 확인할 수 있다.
Failed Post Condition	사용자가 백신에 대한 후기를 확인하지 못한다.

MAIN SUCCESS SCENARIO	
Step	Action
1	사용자가 메인화면에서 ‘게시판’ 버튼을 클릭한다.
2	사용자는 다른 사용자가 작성한 글들을 확인할 수 있다.
3	해당 Use case를 종료한다.

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	회원당 하루에 평균 2번
<Concurrency>	
Due Date	

Use case #14 : 백신 후기 글쓰기	
GENERAL CHARACTERISTICS	
Summary	백신 후기에 대한 글을 게시한다.
Scope	YUVID
Level	User level
Author	
Last Update	
Status	Analysis
Primary Actor	anyone
Preconditions	사용자가 YUVID에 로그인한다.
Trigger	게시판에서 ‘글작성’ 버튼을 클릭했을 때
Success Post Condition	사용자가 백신 후기 게시판에 글을 성공적으로 작성한다.
Failed Post Condition	사용자가 글쓰기에 실패한다.

MAIN SUCCESS SCENARIO	
Step	Action
S	사용자가 YUVID에 로그인한다.
1	사용자가 메인화면에서 ‘게시판’ 버튼을 클릭하여 게시판에 접속한다.
2	사용자가 ‘글작성’ 버튼을 클릭한다.
3	사용자가 글작성에 성공했을 때 Use case를 종료한다.

EXTENSION SCENARIOS	
Step	Branching Action
2	2a1. 로그인 되어있지 않을 때 ‘글쓰기’ 버튼을 눌렀을 때 ...2a2. 먼저 로그인 해달라는 에러 메시지를 출력한다. ...2a3. 로그인 페이지로 이동한다.
	2b1. 글쓰기 페이지 내에서 작성 칸이 비어있을 때 ...2b2. 작성되지 않은 내용이 있다는 에러 메시지를 출력한다.

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	회원당 하루에 평균 2번
<Concurrency>	
Due Date	

Use case #15 : 백신 후기 글 삭제	
GENERAL CHARACTERISTICS	
Summary	백신 후기 게시판에 게시한 글을 삭제한다.
Scope	YUVID
Level	User level
Author	
Last Update	
Status	Analysis
Primary Actor	anyone
Preconditions	사용자가 YUVID에 로그인한다.
Trigger	게시물 창에서 ‘글삭제’ 버튼을 클릭했을 때
Success Post Condition	사용자가 작성한 글을 삭제하는 데 성공한다.
Failed Post Condition	사용자가 글삭제에 실패한다.

MAIN SUCCESS SCENARIO	
Step	Action
S	사용자가 YUVID에 로그인한다.
1	사용자가 메인화면에서 ‘게시판’ 버튼을 클릭하여 게시판에 접속한다.
2	사용자가 게시판에 있는 글 중 삭제하고 싶은 글을 클릭한다.
3	사용자가 게시글 창에서 글 삭제 버튼을 클릭한다.
4	사용자가 글 삭제에 성공했을 때 Use case를 종료한다.

EXTENSION SCENARIOS	
Step	Branching Action
3	<p>3a1. 로그인되어 있지 않거나, 글 작성자가 아닌 사용자가 글삭제 버튼을 눌렀을 때</p> <p>...3a2. 글 작성자만 글을 삭제할 수 있다는 메시지를 출력한다.</p>

RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	회원당 하루에 평균 2번
<Concurrency>	
Due Date	

Use case #16 : 백신 후기 게시글 검색	
GENERAL CHARACTERISTICS	
Summary	백신 후기 게시판에 게시된 글 중 검색하고 싶은 키워드로 글을 검색한다.
Scope	YUVID
Level	User level
Author	
Last Update	
Status	Analysis
Primary Actor	anyone
Preconditions	사용자가 YUVID에 로그인한다.
Trigger	게시판에서 검색창에 검색어 입력후 검색 버튼을 클릭했을 때
Success Post Condition	사용자가 찾고자 하는 글을 DB에서 찾아 리스트로 보여준다.
Failed Post Condition	사용자가 찾고자 하는 글을 검색하는 데 실패한다.

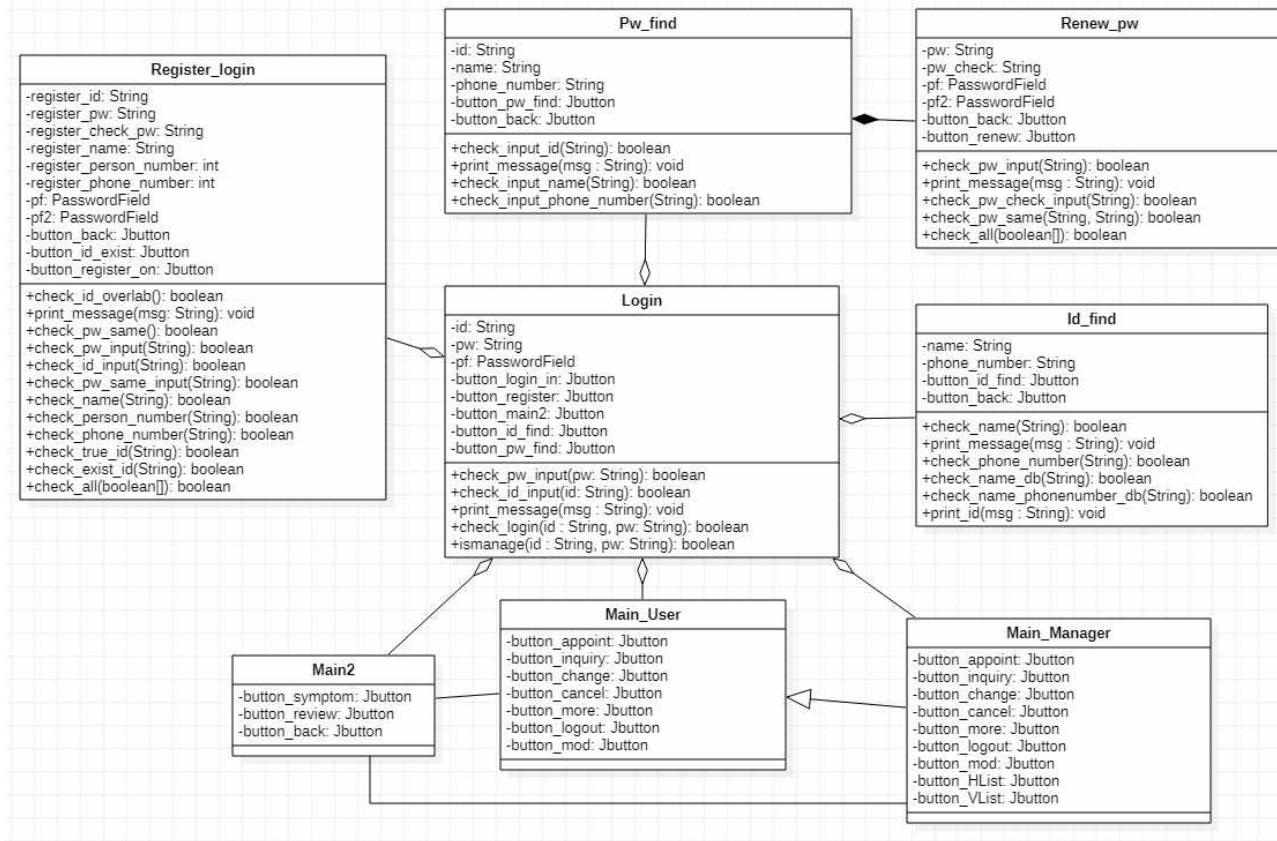
MAIN SUCCESS SCENARIO	
Step	Action
1	사용자가 메인화면에서 ‘게시판’ 버튼을 클릭하여 게시판에 접속한다.
2	사용자가 찾고자 하는 검색어 입력 후 검색 버튼을 클릭한다.
3	사용자가 글 검색에 성공했을 때 Use case를 종료한다.

EXTENSION SCENARIOS	
Step	Branching Action
2	<p>2a1. 검색어 조건에 해당하는 글이 존재하지 않을 때 ...2a2. 해당하는 게시글이 없다는 에러 메시지를 출력한다.</p> <p>2b1. 검색어 칸이 비어있을 때 …2b2. 검색어를 입력해달라는 에러 메시지를 출력한다.</p>

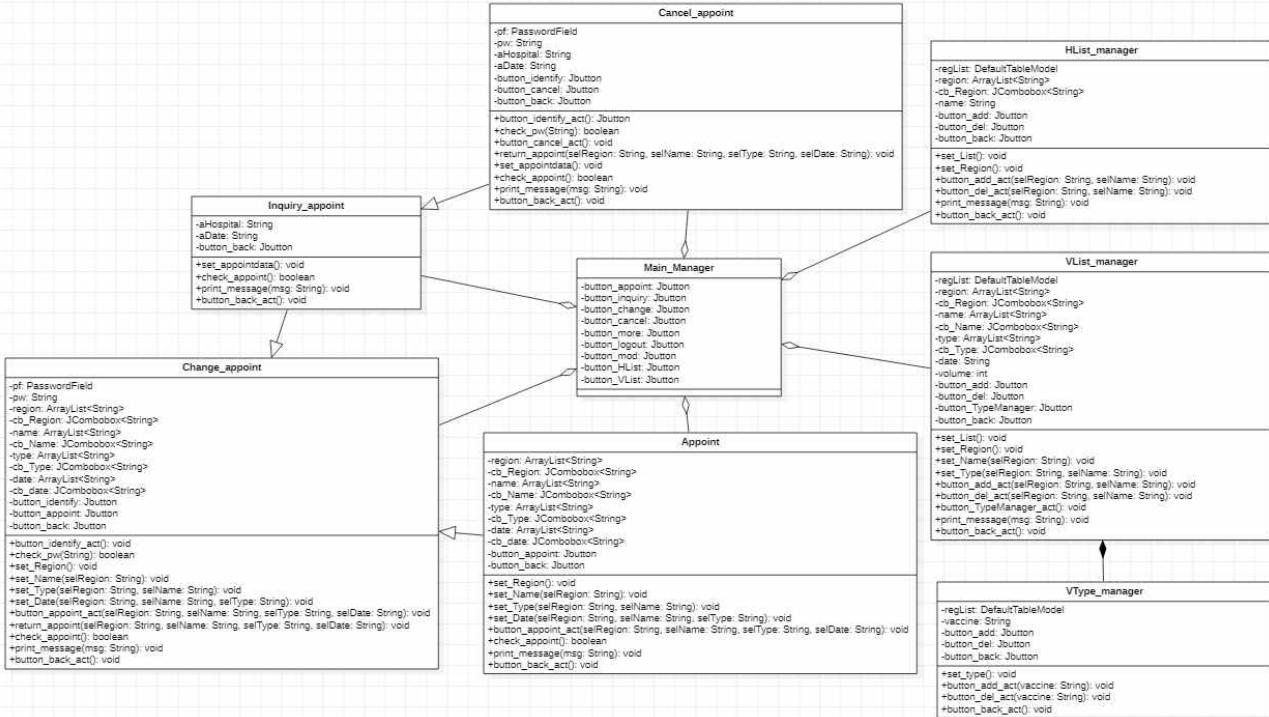
RELATED INFORMATION	
Performance	≤ 2 seconds
Frequency	회원당 하루에 평균 2번
<Concurrency>	
Due Date	

3. Class diagram

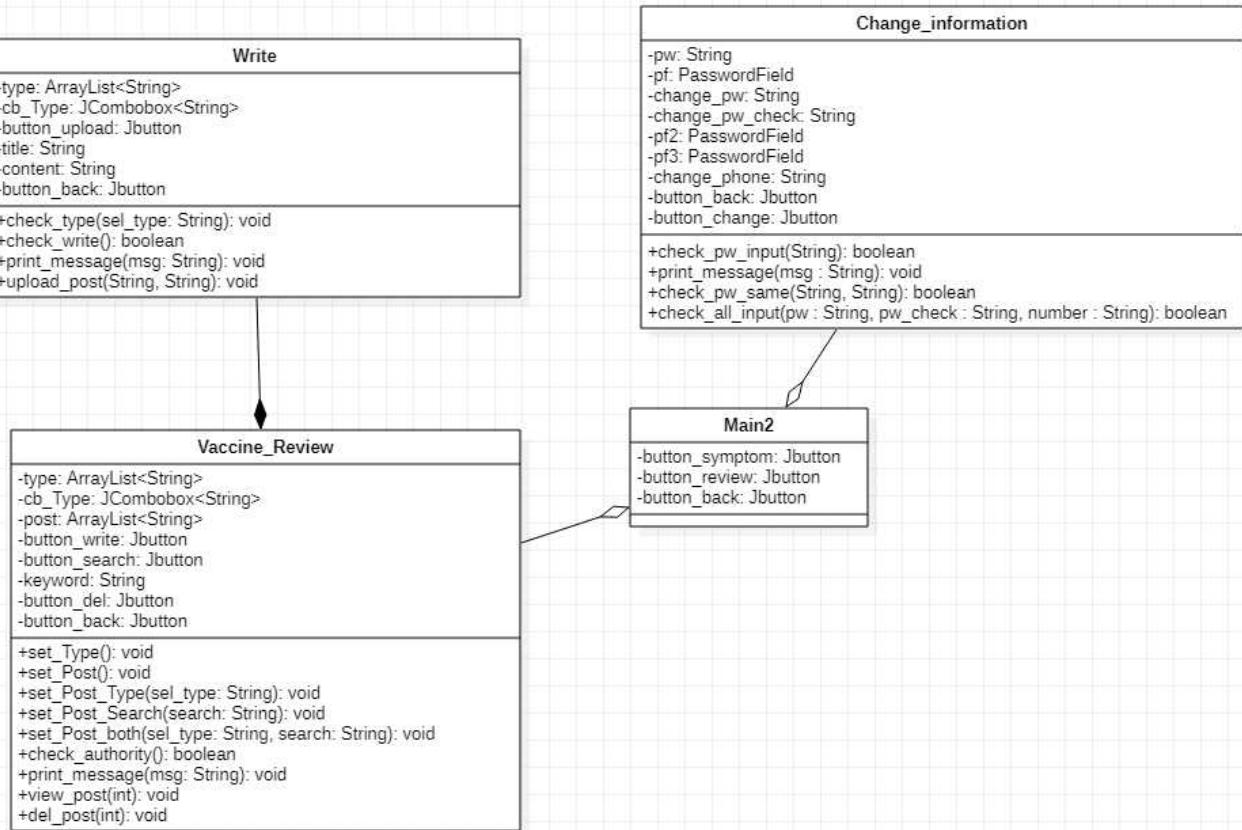
클래스 다이어그램을 크게 3부분으로 나누었다.



로그인 화면에 대한 클래스 다이어그램이다.



메인화면1의 클래스 다이어그램으로 관리자의 사용권한 내에서 일반 사용자의 기능을 모두 사용할 수 있으므로, 관리자의 메인화면만을 원본에 첨부한다.



메인화면2의 클래스 다이어그램이다.

Class #1 : cn

Attribute	userno : int	사용자 회원 번호
	ismanage : boolean	관리자 확인
	condition : boolean	상태 확인
	conn : Connection	DB 연결을 위한 커넥션
	stmt : Statement	SQL문 처리
	pstmt : PreparedStatement	SQL문 처리
	rset : ResultSet	select문 처리

Class #2 : Login_Panel

Attribute	win : Test	패널 전환
	id_tf : JTextField	아이디를 입력받는 텍스트 필드
	pw_pf : JPasswordField	비밀번호를 입력받는 패스워드 필드
	label1 : JLabel	백신 예약 프로그램 라벨
	label2 : JLabel	YUVID 라벨
	label3 : JLabel	ID 라벨
	label4 : JLabel	PW 라벨
	reg_btn : JButton	회원가입 버튼
	find_id_btn : JButton	ID 찾기 버튼
	find_pw_btn : JButton	PW 찾기 버튼
	login_btn : JButton	로그인 버튼
	commu_btn : JButton	게시판 버튼
	Login_Panel(Test)	생성자
Method	check_id_input(String) : boolean	사용자가 아이디를 입력했는지 확인하는 함수
	check_pw_input(String) : boolean	사용자가 비밀번호를 입력했는지 확인하는 함수
	check_login(String, String) : boolean	사용자가 입력한 ID와 PW가 DB에 저장되어있는지 체크하는 함수
	clear_field() : void	텍스트 박스 초기화
	actionPerformed(ActionEvent) : void	액션 이벤트 처리

Class #3 : SignUp_Panel

Attribute	checked_id : String	아이디 중복확인
	win : Test	패널 전환
	label1 : JLabel	ID 라벨
	label2 : JLabel	PW 라벨
	label3 : JLabel	PW 확인 라벨
	label4 : JLabel	이름 라벨

Method	label5 : JLabel	전화번호 라벨
	id_tf : JTextField	아이디를 입력받는 텍스트 필드
	name_tf : JTextField	이름을 입력받는 텍스트 필드
	phone_tf : JTextField	전화번호를 입력받는 텍스트 필드
	pw_pf : JPasswordField	비밀번호를 입력받는 패스워드 필드
	pw_confirm_pf : JPasswordField	비밀번호 재확인을 위해 비밀번호를 한번 더 입력받는 패스워드 필드
	id_check_btn : JButton	아이디 중복확인 버튼
	ok_btn : JButton	가입 버튼
	back_btn : JButton	취소 버튼
	SignUp_Panel(Test)	생성자
Method	clear_field() : void	텍스트 필드 초기화
	insert_id(String, String, String, String)	사용자가 입력한 회원정보를 DB에 저장하는 함수
	actionPerformed(ActionEvent e) : void	액션 이벤트 처리

Class #4 : Find_ID_Panel		
Attribute	name_tf : JTextField	이름을 입력받는 텍스트 필드
	phone_tf : JTextField	전화번호를 입력받는 텍스트 필드
	win : Test	패널 전환
	label1 : JLabel	이름 라벨
	label2 : JLabel	전화번호 라벨
	find_btn : JButton	찾기 버튼
	back_btn : JButton	취소 버튼
Method	Find_ID_Panel(Test)	생성자
	clear_field() : void	텍스트 필드 초기화
	actionPerformed(ActionEvent e) : void	액션 이벤트 처리

Class #5 : Find_PW_Panel		
Attribute	id_tf : JTextField	아이디를 입력받는 텍스트 필드
	name_tf : JTextField	이름을 입력받는 텍스트 필드
	phone_tf : JTextField	전화번호를 입력받는 텍스트 필드
	win : Test	패널 전환
	label1 : JLabel	아이디 라벨
	label2 : JLabel	이름 라벨
	label3 : JLabel	전화번호 라벨
	find_btn : JButton	찾기 버튼
	back_btn : JButton	취소 버튼

Method	Find_PW_Panel(Test)	생성자
	clear_field() : void	텍스트 필드 초기화
	actionPerformed(ActionEvent) : void	액션 이벤트 처리

Class #6 : Confirm_PW_Panel		
Attribute	win : Test	패널 전환
	pw_tf : JTextField	비밀번호를 입력받는 패스워드 필드
	label1 : JLabel	현재 비밀번호 라벨
	verity_btn : JButton	확인 버튼
	back_btn : JButton	취소 버튼
Method	Confirm_PW_Panel(Test)	생성자
	clear_field() : void	텍스트 필드 초기화
	actionPerformed(ActionEvent) : void	액션 이벤트 처리

Class #7 : MainPanel		
Attribute	win : Test	패널 전환
	user_mod_btn : JButton	회원정보 수정 버튼
	back_btn : JButton	로그아웃 버튼
	appoint_btn : JButton	접종 예약 버튼
	inquiry_btn : JButton	접종 조회 버튼
	cancel_btn : JButton	접종 취소 버튼
	hospital_mod_btn : JButton	병원 리스트 수정 버튼
	vaccine_mod_btn : JButton	백신 재고리스트 수정 버튼
	commu_btn : JButton	게시판 버튼
Method	MainPanel(Test)	생성자
	menu_manage() : void	관리자임을 판별하고 관리자에게만 보여야 하는 버튼을 처리하는 함수
	actionPerformed(ActionEvent) : void	액션 이벤트 처리

Class #8 : User_Modify_Panel		
Attribute	win : Test	패널 전환
	pw_pf : JPasswordField	비밀번호를 입력 받는 패스워드 필드
	pw_confirm_pf : JTextField	비밀번호 재확인을 위해 비밀번호를 한

Method	JPasswordField	번 더 입력받는 패스워드 필드
	phone_tf : JTextField	전화번호를 입력받는 텍스트 필드
	label1 : JLabel	새 비밀번호 라벨
	label2 : JLabel	새 비밀번호 확인 라벨
	label3 : JLabel	새 전화번호 입력 라벨
	modify_btn : JButton	변경 버튼
	back_btn : JButton	취소 버튼
	User_Modify_Panel(Test)	생성자

Class #9 : JPanel08		
Attribute	win : Test	패널 전환
	name_cb : JComboBox<String>	지역 선택을 위한 콤보박스
	hospital_cb : JComboBox<String>	병원 선택을 위한 콤보박스
	vaccine_cb : JComboBox<String>	백신 종류 선택을 위한 콤보박스
	idx : ArrayList<Integer>	접종 예약 정보를 DB에 저장하기 위해 사용자가 선택한 정보를 저장할 ArrayList
	appoint_btn : JButton	예약 버튼
	Back_btn : JButton	취소 버튼
	JPanel08(Test)	생성자
Method	update_Hlist(String) : void	접종 예약 정보에 따라 병원리스트를 수정하는 함수
	update_Vlsit(int) : void	접종 예약 정보에 따라 백신재고리스트를 수정하는 함수
	clear_field() : void	텍스트 필드 초기화
	actionPerformed(ActionEvent e) : void	액션 이벤트 처리

Class #10 : JPanel09

Attribute	win : Test	패널 전환
	sql1 : String	sql문을 저장하는 String 변수
	sql2 : String	sql문을 저장하는 String 변수
	sql3 : String	sql문을 저장하는 String 변수
	label1 : JLabel	라벨
	label2 : JLabel	라벨
	label3 : JLabel	라벨
	label4 : JLabel	라벨
	label5 : JLabel	경고 문구 라벨
	back_btn : JButton	예약 취소 버튼
Method	confirm_btn : JButton	확인 버튼
	back2_btn : JButton	돌아가기 버튼
	JPanel09(Test)	생성자
Method	is_inquiry() : void	사용자 상태 확인 함수
	actionPerformed(ActionEvent event) : void	액션 이벤트 처리

Class #11 : JPanel12

Attribute	win : Test	패널 전환
	locate_arr : ArrayList<String>	지역 리스트
	label1 : JLabel	병원이름 입력 라벨
	label2 : JLabel	전화번호 입력 라벨
	back_btn : JButton	뒤로 버튼
	add_btn : JButton	추가 버튼
	del_btn : JButton	삭제 버튼
Method	JPanel12(Test)	생성자
	actionPerformed(ActionEvent event) : void	액션 이벤트 처리

Class #12 : JPanel13

Attribute	win : Test	패널 전환
	volume_tf : JTextField	백신 재고를 입력받을 텍스트 필드
	input_time : JTextField	시간을 입력받을 텍스트 필드
	name_cb : JComboBox<String>	지역 리스트
	hospital_cb : JComboBox<String>	병원 리스트

	vaccine_cb : JComboBox<String>	백신 재고 리스트
	lblLb1 : JLabel	시간 라벨
	lblLb2 : JLabel	재고 라벨
	add_btn : JButton	추가 버튼
	del_btn : JButton	삭제 버튼
	back_btn : JButton	뒤로가기 버튼
	addVacType : JButton	백신종류추가 버튼
Method	JPanel13(Test)	생성자
	actionPerformed(ActionEvent) : void	액션 이벤트 처리

Class #13 : JPanel14		
Attribute	win : Test	패널 전환
	vaccine_tf : JTextField	백신 이름을 입력받을 텍스트 필드
	label1 : JLabel	백신이름 라벨
	add_btn : JButton	추가 버튼
	del_btn : JButton	삭제 버튼
	back_btn : JButton	뒤로가기 버튼
Method	JPanel14(Test)	생성자
	clear_field() : void	텍스트 필드 초기화
	actionPerformed(ActionEvent) : void	액션 이벤트 처리

Class #14 : Community_Panel		
Attribute	win : Test	패널 전환
	colNames : String[]	게시물 리스트 테이블 생성 시 열 이름 배열
	table : JTable	게시물 리스트
	table2 : JTable	게시물 리스트2
	searchString : JTextField	검색 시 검색어 입력 받을 텍스트 필드
	scrollPane : JScrollPane	스크롤 기능을 위한 스크롤페이지
	search_cb : JComboBox<String>	검색 조건 콤보박스
	search_btn : JButton	검색 버튼
	write_btn : JButton	글작성 버튼
	refresh_btn : JButton	새로고침 버튼
	back_btn : JButton	나가기 버튼

Method	Community_Panel(Test)	생성자
	table_refresh() : void	게시물 리스트를 새로고침 하는 함수
	get_userid(int) : String	userinfo를 이용해 DB에서 유저의 아이디를 가져오는 함수
	insert(String, String, String) : void	사용자가 입력한 게시글을 DB에 저장하는 함수
	delete(CommunityVO) : void	사용자가 삭제하고자 하는 게시글을 DB에서 삭제하는 함수
	select() : List<CommunityVO>	현재 게시된 게시글을 DB에서 불러와 리스트로 만들어주는 함수
	search(String, String) : void	사용자가 검색하고자 하는 조건과 검색어에 따라 게시글을 검색해주는 함수
	mouseClicked(MouseEvent) : void	마우스 액션 이벤트 처리
	actionPerformed(ActionEvent) : void	액션 이벤트 처리

Class #15 : CommunityVO		
Attribute	num : int	게시글 번호
	vname : String	백신 이름
	title : String	게시글 제목
	content : String	게시글 내용
	writer : String	게시글 작성자

Class #16 : Post_Search		
Attribute	table : JTable	게시글 리스트 테이블
	Post_Search(List<CommunityVO>)	생성자
Method	actionPerformed(ActionEvent) : void	액션 이벤트 처리

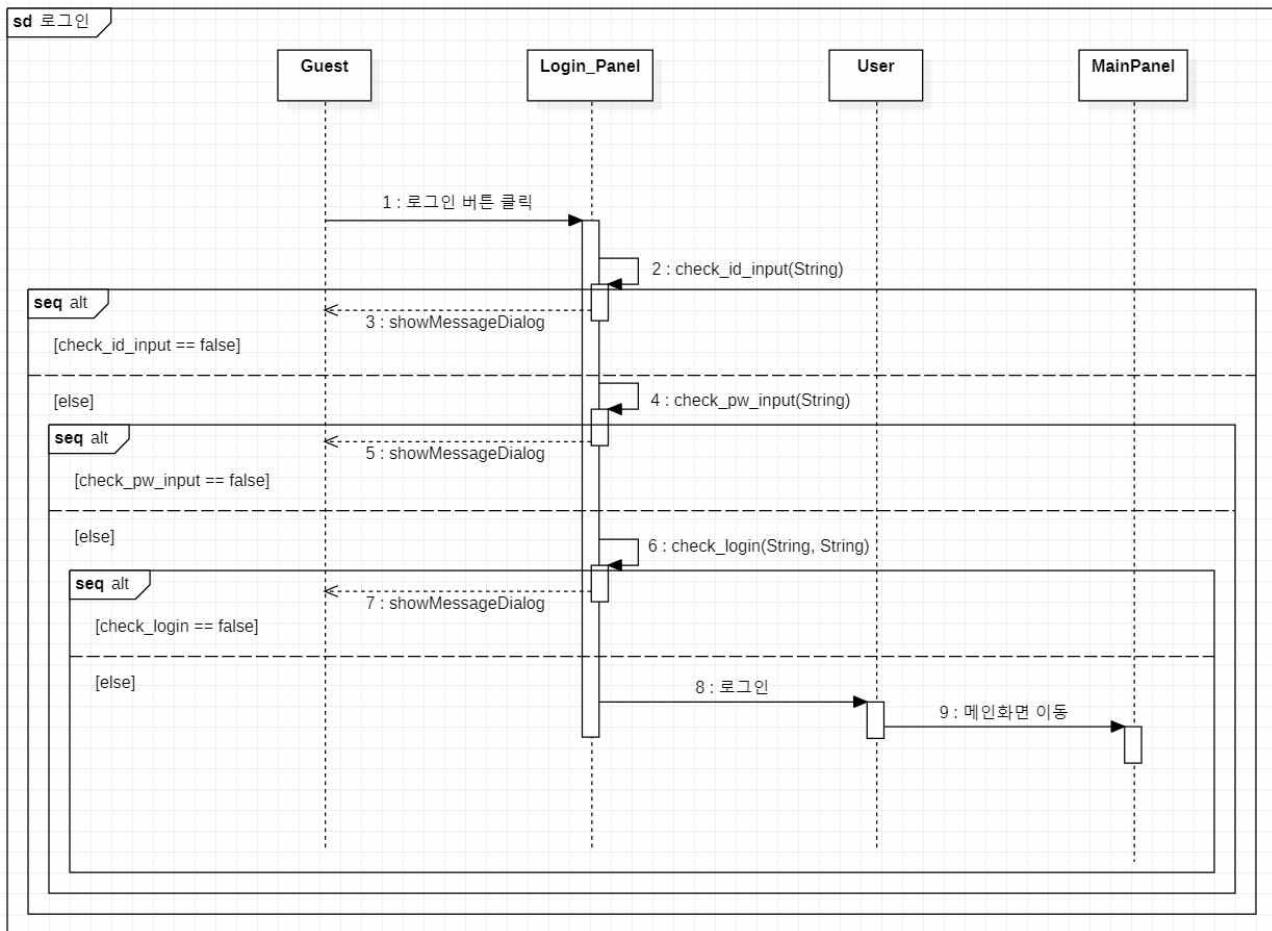
Class #17 : Post_View		
Method	Post_View(CommunityVO)	생성자
	actionPerformed(ActionEvent) : void	액션 이벤트 처리

Class #18 : Post_Add

Attribute	title_tf : JTextField	게시글 제목을 입력 받을 텍스트 필드
	content_ta : JTextArea	게시글 내용을 입력 받을 텍스트 필드
	vnameLabel : JLabel	백신 종류 라벨
	titleLabel : JLabel	제목 라벨
	contentLabel : JLabel	내용 라벨
	btnWrite : JButton	작성완료 버튼
Method	Post_Add()	생성자
	actionPerformed(ActionEvent event) : void	액션 이벤트 처리

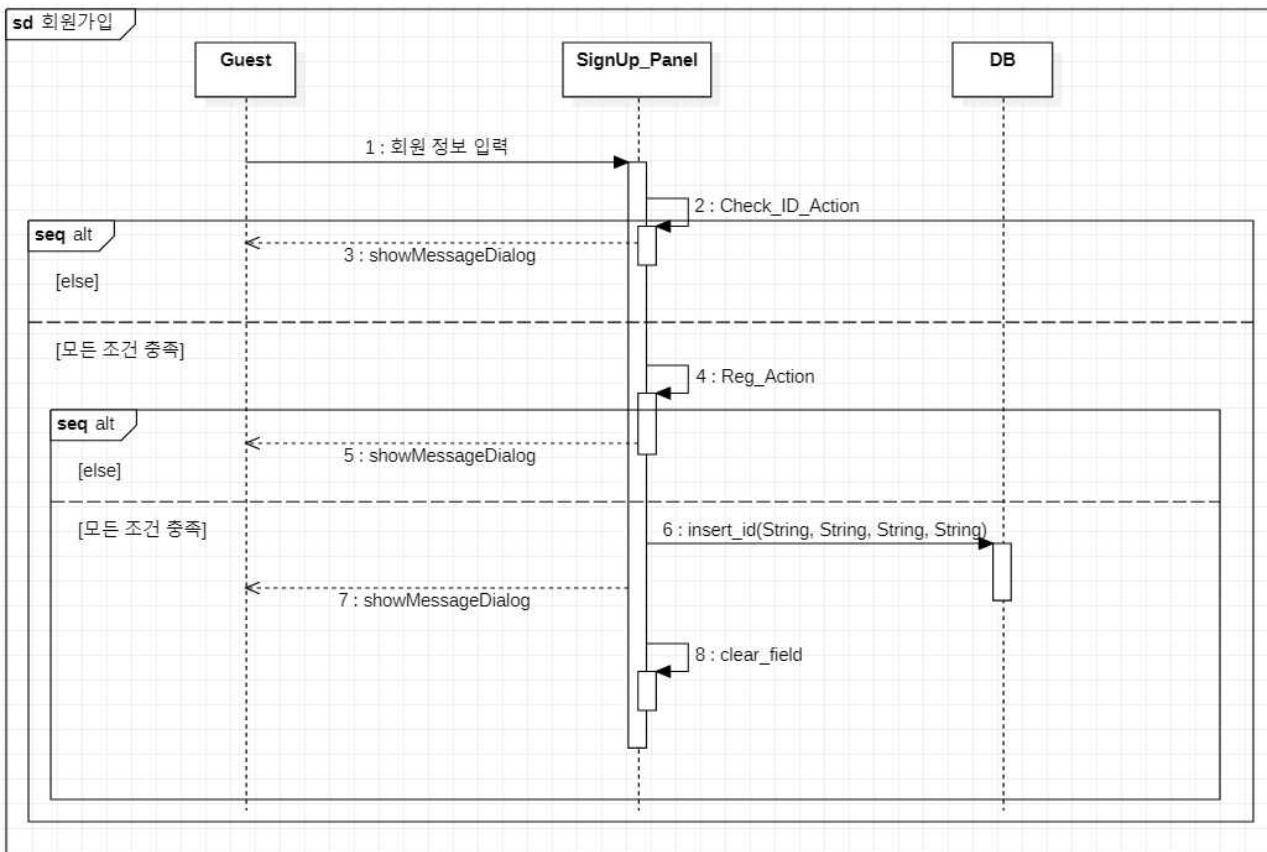
4. Sequence diagram

1) 로그인



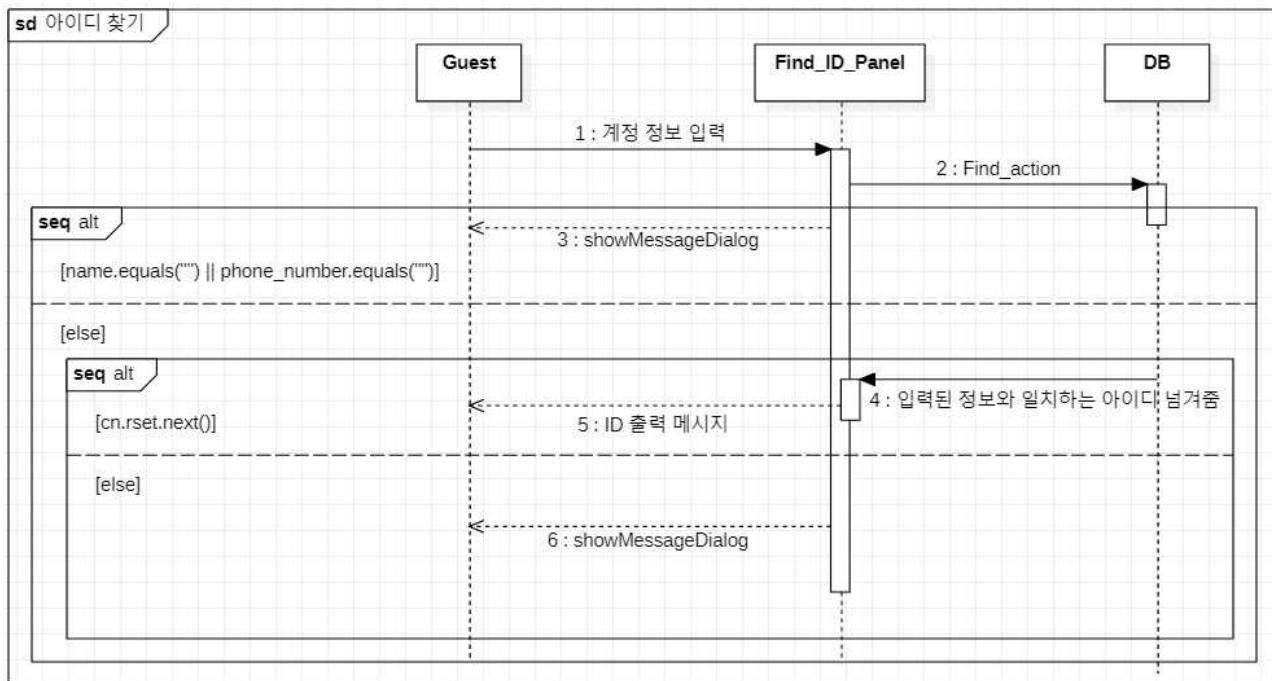
사용자가 로그인하는 것을 나타낸 Sequence Diagram이다. 사용자 또는 관리자가 로그인 버튼을 눌러 로그인을 시도한다. 먼저 check_id_input 함수로 아이디 입력 칸을 입력했는지 확인하고, 입력하지 않았으면 아이디를 입력해달라는 메시지를 띠운다. 그다음 check_pw_input 함수로 비밀번호 입력 칸을 입력했는지 확인하고, 입력하지 않았으면 비밀번호를 입력해달라는 메시지를 띠운다. 마지막으로 check_login 함수로 입력한 아이디와 비밀번호가 DB에 저장되어있는지 확인하고 저장된 아이디와 비밀번호가 아니면 존재하지 않는 아이디라는 메시지를 띠우고 저장된 아이디와 비밀번호이면 로그인에 성공하고 메인화면으로 이동한다.

2) 회원가입



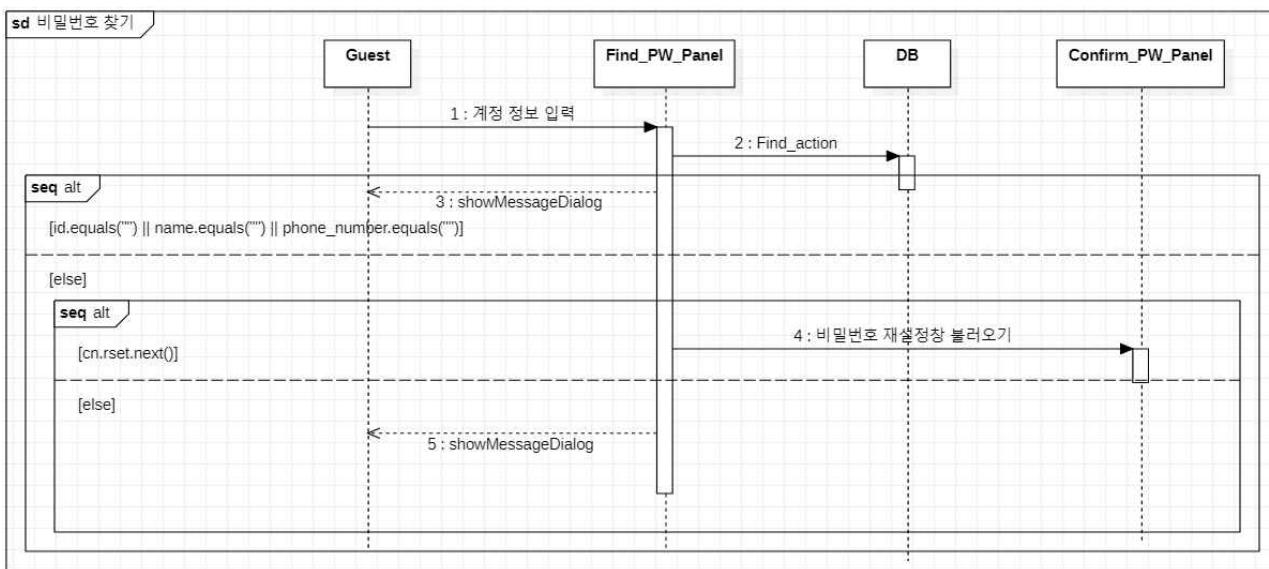
사용자가 회원가입 하는 것을 나타낸 Sequence Diagram이다. 로그인 되지 않은 사용자가 로그인 화면에서 회원가입 버튼을 누르면 회원가입을 위해 필요한 정보들을 입력하도록 한다. 각 정보들이 제대로 입력되어 있는지, 비어있는 칸은 없는지, 아이디가 중복되지는 않는지 확인하고 제대로 작성되지 않은 정보가 있을 경우, 에러 메시지를 출력하도록 한다. 모든 정보가 제대로 입력되었다면 DB에 사용자가 입력한 정보를 저장하고 가입이 완료되었다는 메시지를 띠운다.

3) 아이디 찾기



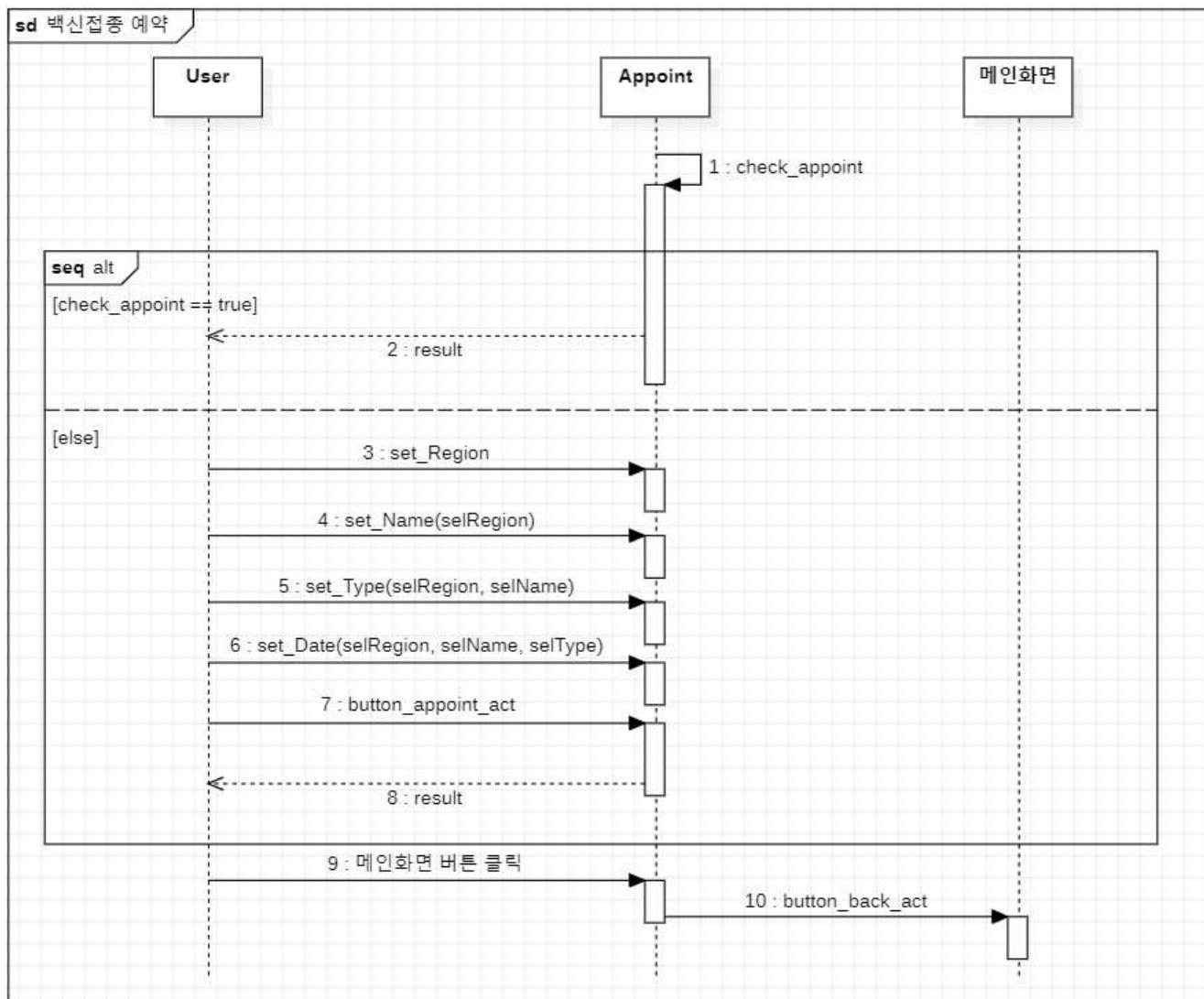
사용자가 ID를 찾는 것을 나타낸 Sequence Diagram이다. 사용자가 아이디 찾기 창에서 이름과 전화번호를 입력하고 찾기 버튼을 누르면 find_action을 통해 먼저 이름과 전화번호 입력 칸 둘 다 입력되어 있는지 검사하고, 빈칸이 있으면 경고메시지를 출력한다. 입력 칸이 모두 입력되어 있다면 DB에 연결하여 일치하는 계정정보가 있는지 확인하고 있으면 아이디를 메시지창을 통해 출력하고, 없으면 존재하는 아이디가 없다는 메시지창을 띄운다.

4) 비밀번호 찾기



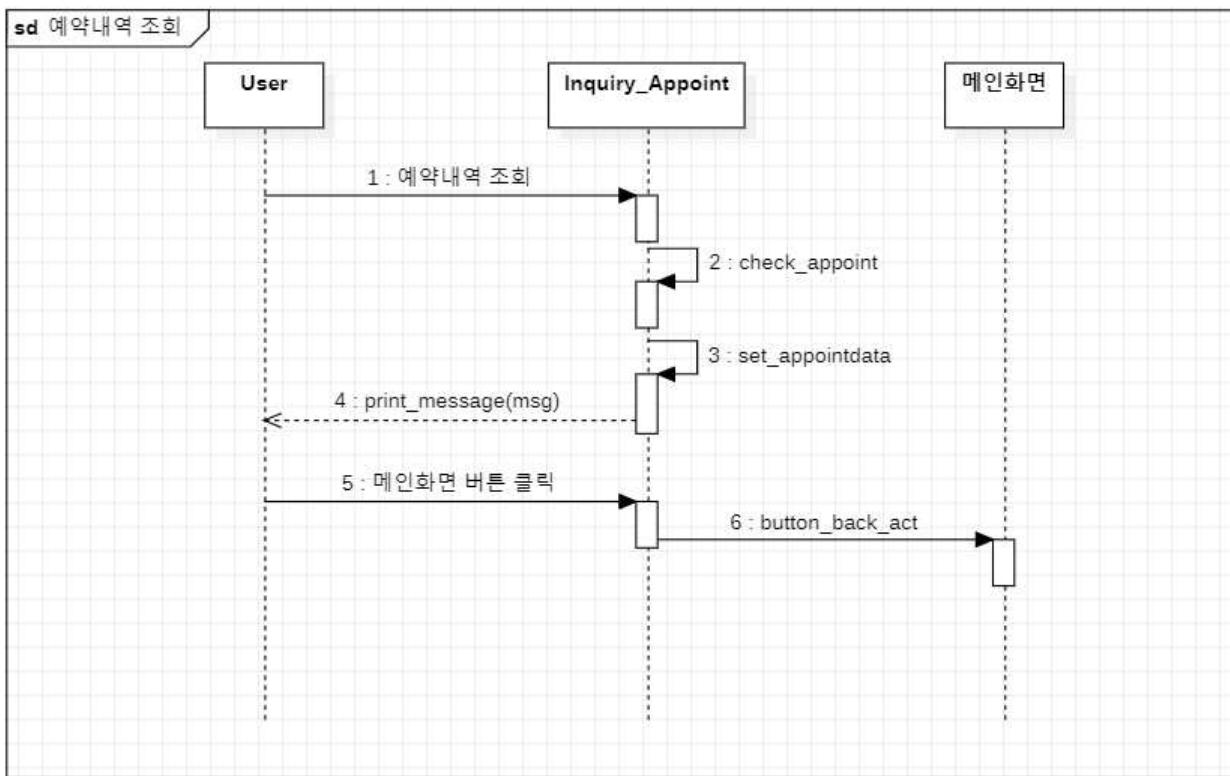
사용자가 비밀번호를 찾는 것을 나타낸 Sequence Diagram이다. 사용자가 비밀번호 찾기 창에서 아이디와 이름, 전화번호를 입력하고 찾기 버튼을 누르면 find_action을 통해 먼저 아이디와 이름, 전화번호 입력 칸 셋 다 입력되어 있는지 검사하고, 빈칸이 하나라도 있으면 경고메시지를 출력한다. 입력 칸이 모두 입력되어 있다면 DB에 연결하여 일치하는 계정정보가 있는지 확인하고 있으면 비밀번호 재설정 창으로 전환하고, 없으면 입력하신 정보와 일치하는 계정이 없다는 메시지창을 띠운다.

5) 백신접종 예약



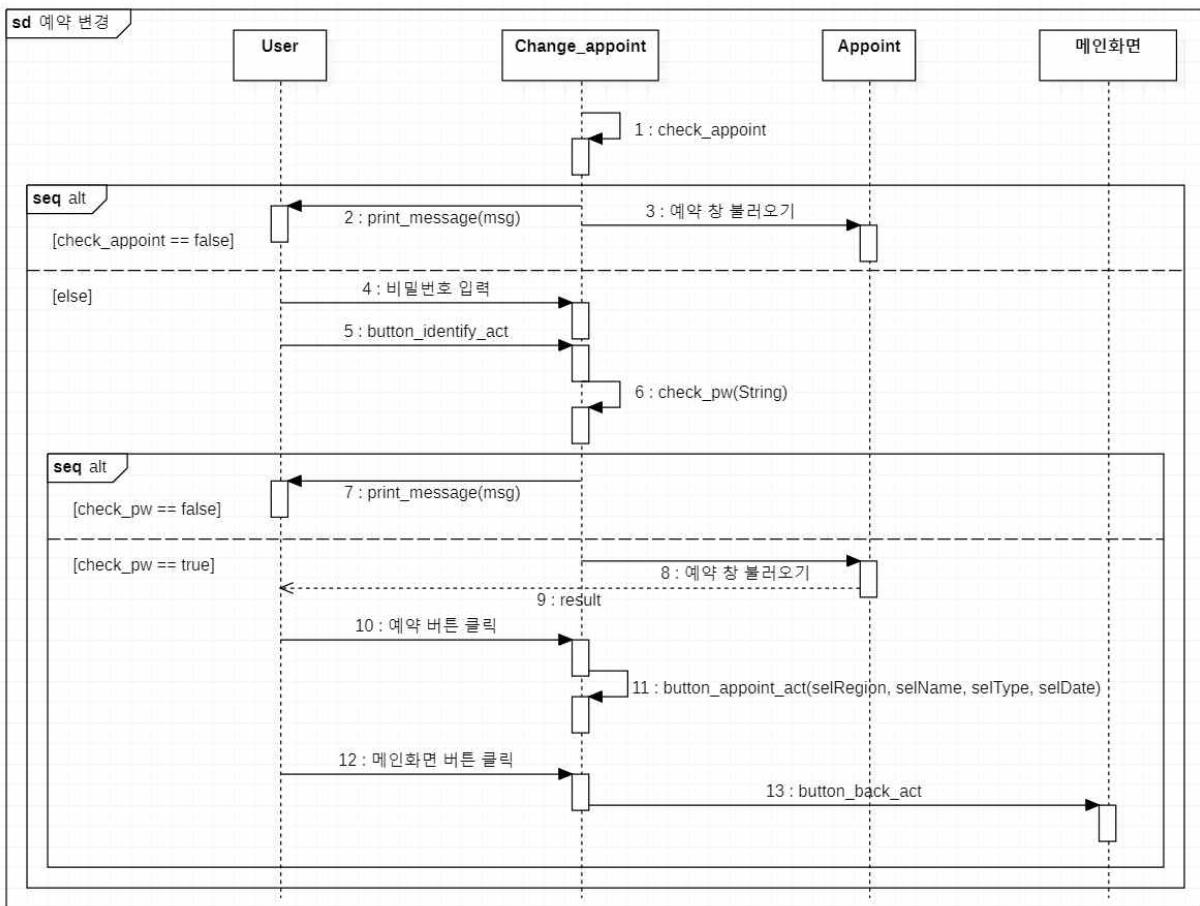
사용자가 백신접종 예약을 하는 것을 나타낸 Sequence Diagram이다. 사용자가 백신접종 예약 버튼을 누르면 우선 check_appoint 메소드로 사용자의 예약 내역이 있는지 여부를 확인한다. check_appoint의 boolean 값이 true면 예약 내역이 이미 있으므로 예약되어 있다는 메시지를 출력해준다. boolean 값이 false라면 새롭게 예약을 하기 위한 절차를 시작한다. 사용자가 접종을 원하는 지역과 병원, 백신종류, 접종일자를 set_Region, set_Name, set_Type, set_Date 메소드를 통해 설정한다. 사용자가 모두 선택하고 나서 백신 예약 버튼을 누르면 예약이 완료되었다는 메시지를 출력한다. 사용자가 메인화면 버튼을 클릭하면 button_back_act() 메소드를 실행시켜 메인화면으로 돌아간다.

6) 예약내역 조회



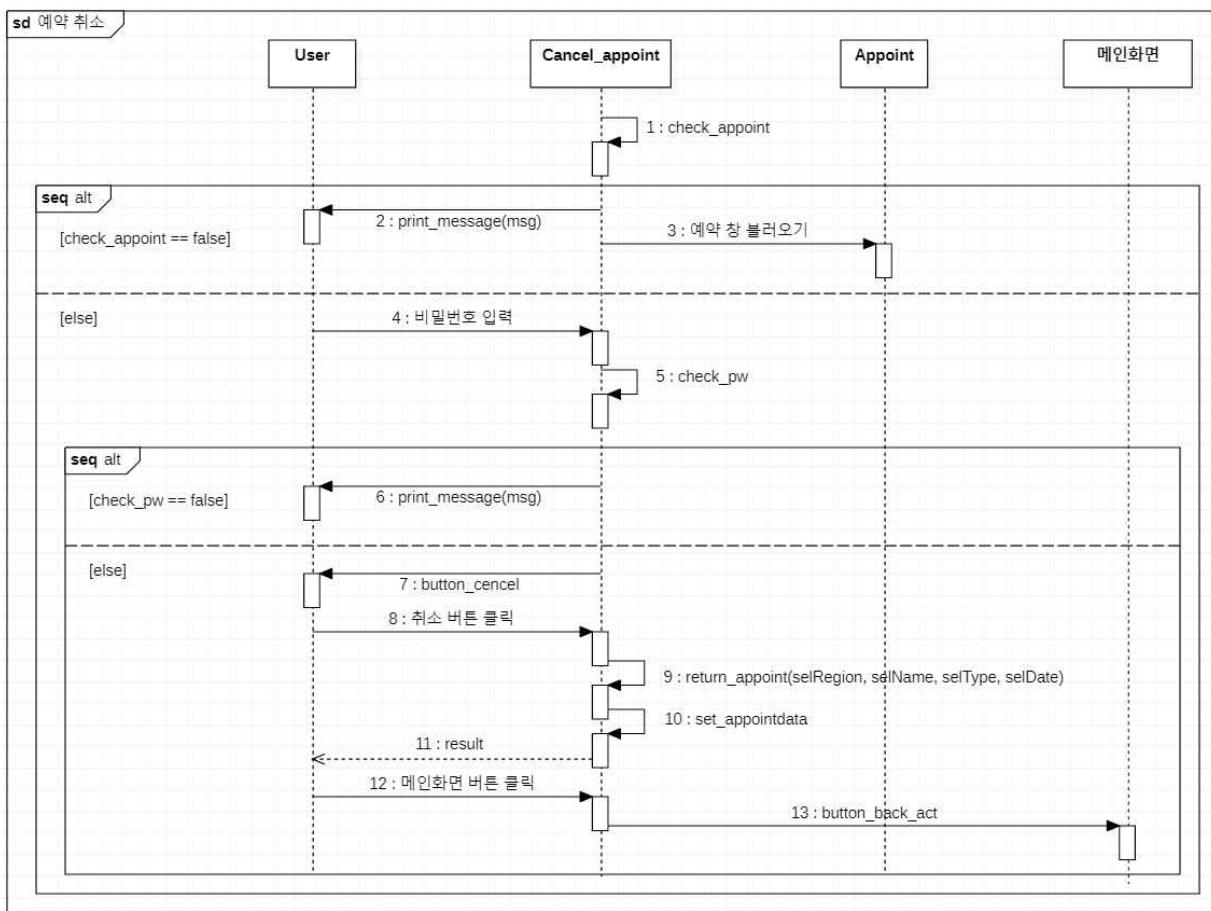
사용자가 백신접종 예약 내역을 조회하는 것을 나타낸 Sequence Diagram이다. 사용자가 예약내역 조회 버튼을 누르면 check_appoint() 메소드가 사용자의 예약 여부를 판별한다. 사용자의 예약 정보가 있을 시, set_appointdata() 메소드로 사용자의 예약된 접종 병원과 접종일자 정보를 설정하고 print_message() 메소드에서 완료되었음을 메시지로 출력한다. 사용자가 메인화면 버튼을 클릭하면 button_back_act() 메소드를 실행시켜 메인화면으로 돌아간다.

7) 예약 변경



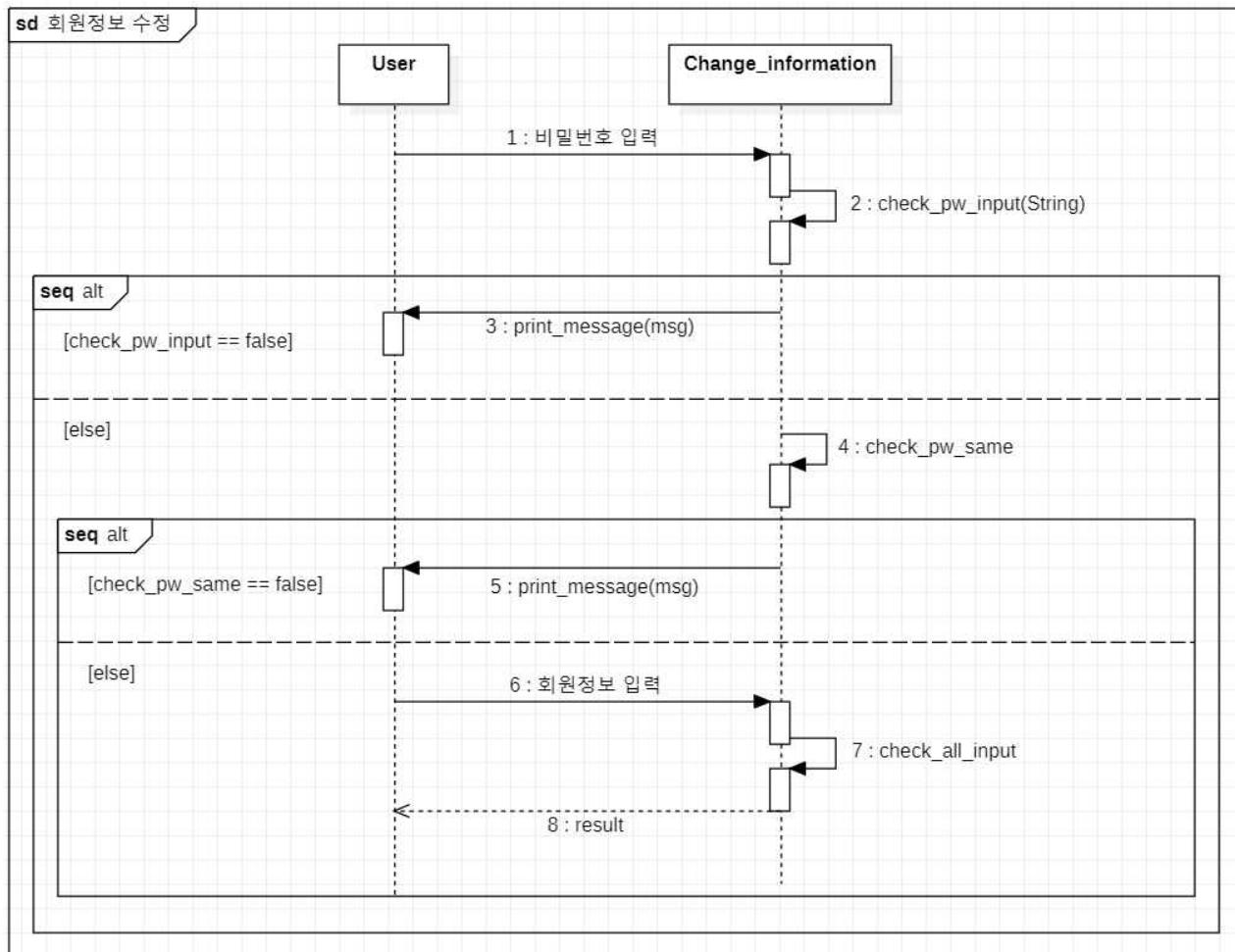
사용자가 예약을 변경하는 것을 나타낸 Sequence Diagram이다. 사용자가 예약 변경 버튼을 누르면 `check_appoint` 메소드를 이용해 먼저 사용자의 예약 여부를 확인한다. 이 메소드에서 반환되는 boolean 값이 false라면 예약이 되지 않은 상태이므로 예약이 되지 않은 상태라는 메시지 창을 띄우고 예약 창을 불러온다. boolean 값이 true라면 예약이 된 상태이므로 비밀번호 입력칸을 띄우고 비밀번호 입력 후 일치 여부 확인 버튼을 누르면 `check_pw()` 메소드를 이용해 입력한 비밀번호와 회원의 비밀번호가 일치하는지 확인한다. 그 후 차례로 `set_Region`, `set_Name`, `set_Type`, `set_Date` 메소드를 실행시켜 지역, 병원, 이름, 백신 종류를 선택해 예약을 변경한다. 예약 변경 후 메인화면 버튼 클릭시 메인화면으로 돌아간다.

8) 예약 취소



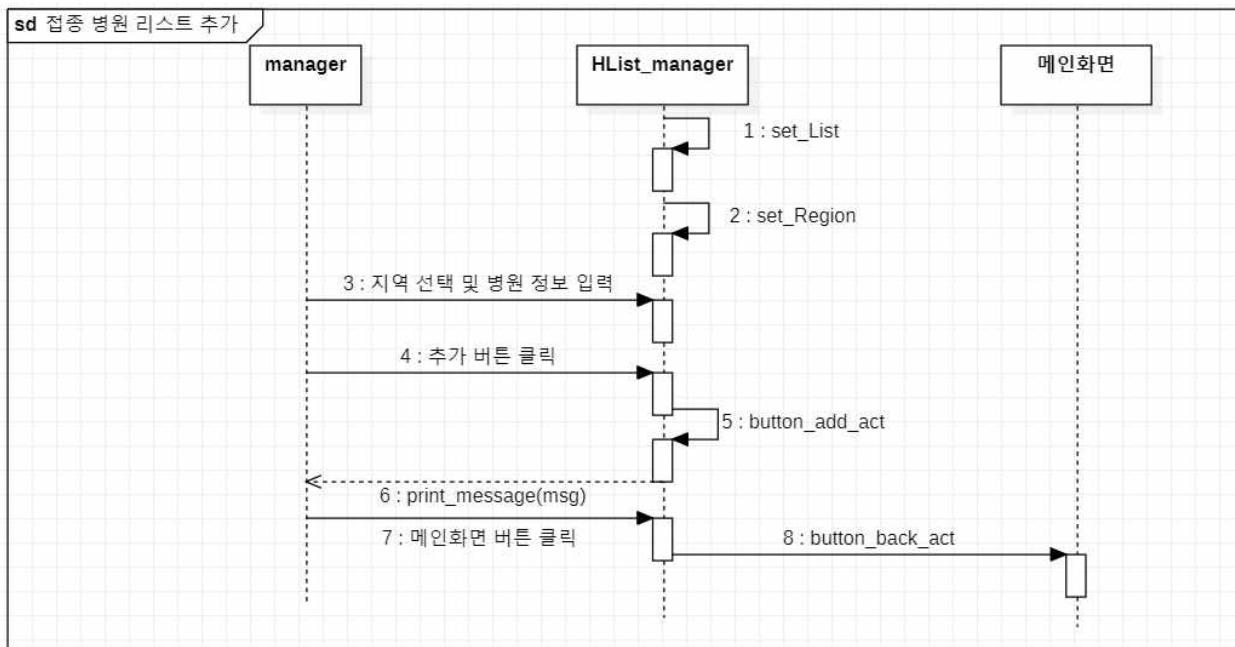
사용자가 예약을 취소하는 것을 나타낸 Sequence Diagram이다. 사용자가 예약 취소 버튼을 누르면 check_appoint 메소드를 이용해 먼저 사용자의 예약 여부를 확인한다. 이 메소드에서 반환되는 boolean 값이 false라면 예약이 되지 않은 상태이므로 예약이 되지 않은 상태라는 메시지 창을 띄우고 예약 창을 불러온다. boolean 값이 true라면 비밀번호 입력칸을 띄우고 사용자가 비밀번호를 입력한 후 일치 여부 확인 버튼을 누르면 check_pw() 메소드를 이용해 입력한 비밀번호와 회원의 비밀번호가 일치하는지 확인한다. 비밀번호가 일치하면 정말 예약을 취소할 것인지 재확인 창을 띄우고 사용자가 확인을 누르면 return_appoint() 메소드로 취소한 백신을 다시 백신리스트에 추가하고 set_appointdata() 메소드로 병원 정보와 날짜 정보를 재설정한다. 재설정후 사용자에게 취소되었다는 메시지 창을 띄운다. 그 후 사용자가 메인화면 버튼을 클릭하면 button_back_act() 메소드를 실행시켜 메인화면으로 돌아간다.

9) 회원정보 수정



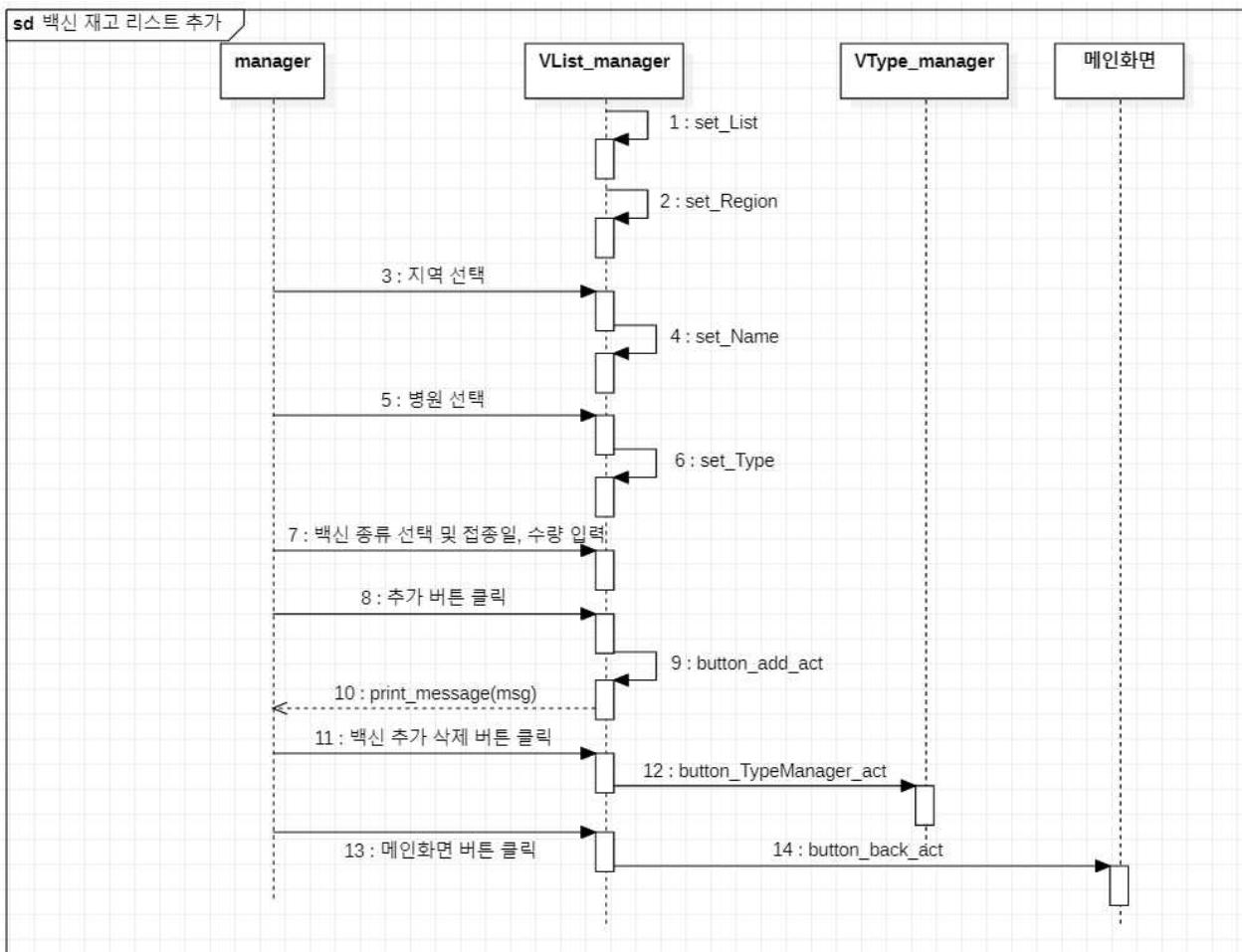
사용자가 회원정보를 수정하는 것을 나타낸 Sequence Diagram이다. 사용자가 회원정보 수정 버튼을 누르면 비밀번호 입력 창이 뜬다. 비밀번호를 입력하면 check_pw_input() 메소드로 비밀번호 입력 칸에 입력 값이 있는지 확인하고, 반환된 boolean 값이 false이면 입력 값이 없다는 것이므로 print_message() 메소드를 통해 에러메세지를 출력한다. true이면 입력 값이 있다는 것이므로 check_pw_same 메소드를 실행해 입력한 비밀번호와 회원의 비밀번호가 일치하는지 확인한다. 반환된 boolean 값이 false 이면 일치하지 않는다는 것이므로 print_message() 메소드를 통해 에러메시지를 출력하고 true 이면 일치한다는 것이므로 회원 정보 입력 창을 띄운다. 사용자가 회원정보를 입력 한 후 check_all_input() 메소드를 실행시켜 최소 1개 이상의 입력칸이 채워져 있는지 확인한 후 false이면 수정할 정보가 없다는 메시지를 출력하고, true이면 수정이 완료되었다는 메시지를 출력한다.

10) 접종 병원 리스트 추가



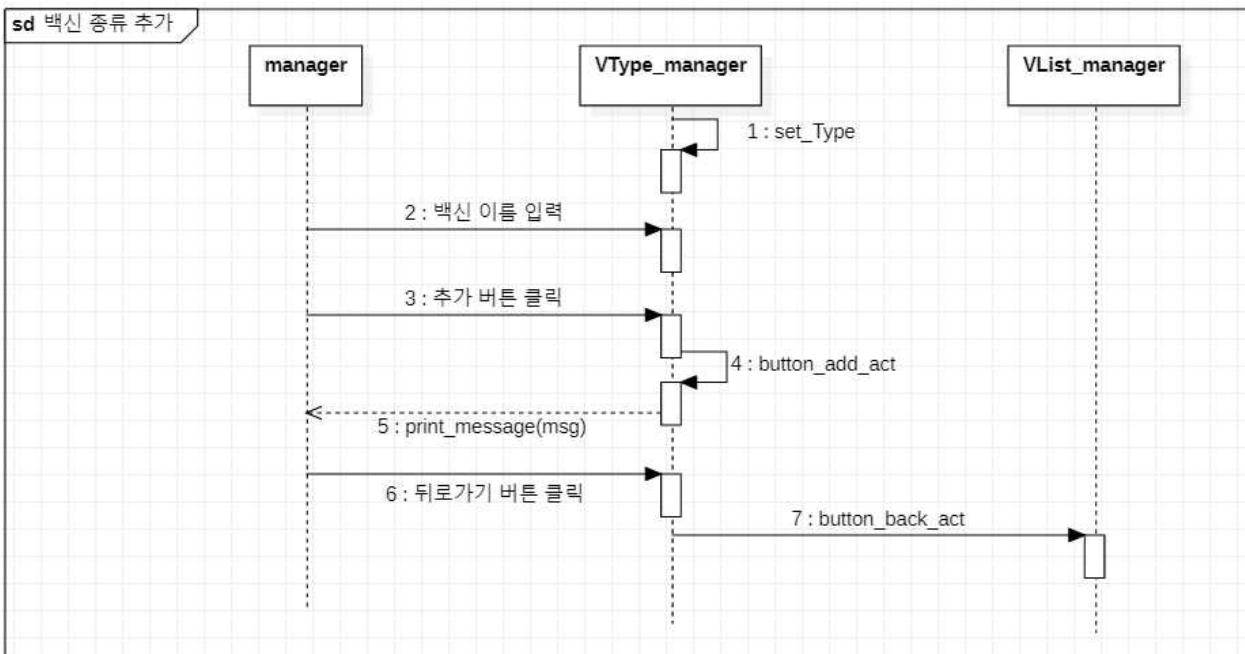
관리자가 접종 병원 리스트를 추가하는 것을 나타낸 Sequence Diagram이다. 먼저 접종병원 리스트를 누르면 set_List, set_Region 메소드로 현재 등록된 모든 지역, 병원 리스트를 보여준다. 지역 선택 및 병원 정보 입력후 추가 버튼을 누르면 button_add_act 메소드가 실행되고 print_message 메소드를 통해 추가가 완료되었다는 메시지를 출력하며, 접종 병원 리스트 삭제 역시 동일하게 진행된다. 메인화면 버튼을 누르면 메인화면으로 돌아간다.

11) 백신 재고 리스트 추가



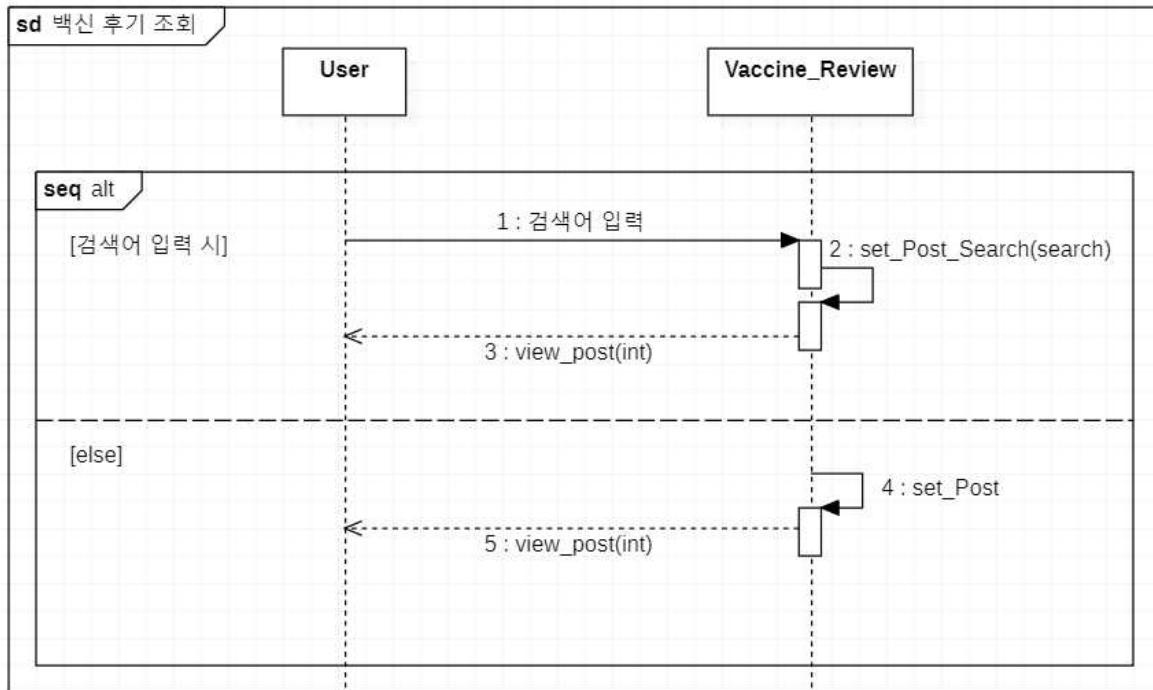
관리자가 병원에 백신 재고를 리스트에 추가하는 것을 나타낸 Sequence Diagram이다. 먼저 백신 재고 리스트를 누르면 set_List, set_Region 메소드로 현재 등록된 모든 지역, 병원, 백신, 시간, 수량 리스트를 보여준다. 지역 선택시 set_Name 메소드로 해당 지역의 병원을 보여주고 병원이름 선택시 set_Type으로 등록 가능한 백신의 종류를 보여준다. 백신 접종일 및 수량 입력후 추가 버튼을 누르면 button_add_act 메소드가 실행되고 print_message 메소드를 통해 추가가 완료되었다는 메시지를 출력하며, 백신 재고 리스트 삭제 역시 동일하게 진행된다. 백신 종류 추가/삭제 버튼을 누르면 백신 종류 추가/삭제 화면으로, 메인화면 버튼을 누르면 메인화면으로 돌아간다.

12) 백신 종류 추가/삭제



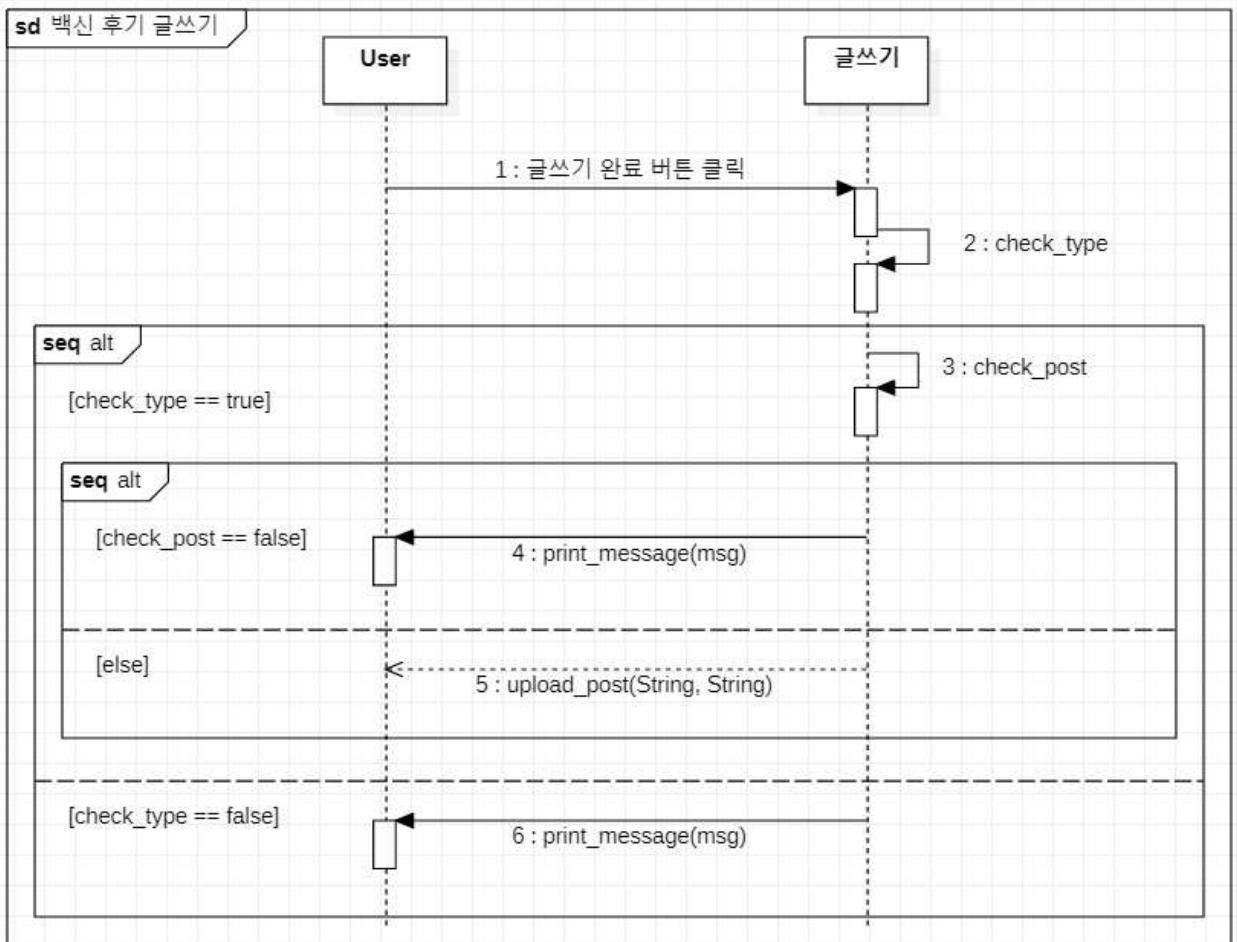
관리자가 백신 종류를 추가하는 것을 나타낸 Sequence Diagram이다. 먼저 접종병원 리스트를 누르면 set_Type 메소드로 현재 등록된 모든 백신 리스트를 보여준다. 백신 이름 입력후 추가 버튼을 누르면 button_add_act 메소드가 실행되고 print_message 메소드를 통해 추가가 완료되었다는 메시지를 출력하며, 백신 종류 삭제 역시 동일하게 진행된다. 메인화면 버튼을 누르면 메인화면으로 돌아간다.

13) 백신 후기 게시판



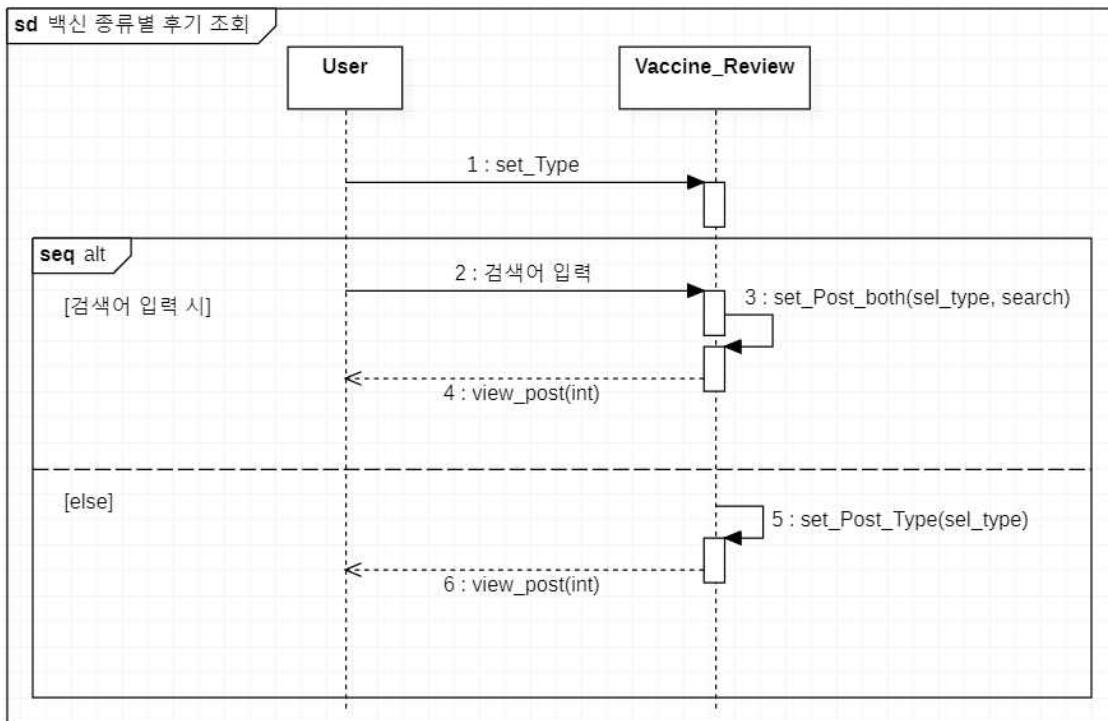
사용자가 게시판에 게시된 백신 후기 글을 조회하는 것을 나타낸 Sequence Diagram이다. 검색어 입력 시 사용자가 검색어를 입력하면 set_Post_Search 메소드를 실행시켜 검색어가 포함된 게시글을 검색하여 post를 설정하고, 설정된 post를 view_post 메소드로 보여준다. 검색어를 입력하지 않고 글 조회 시 set_post로 post를 설정하고, view_post로 후기 글을 보여준다.

13) 백신 후기 글쓰기



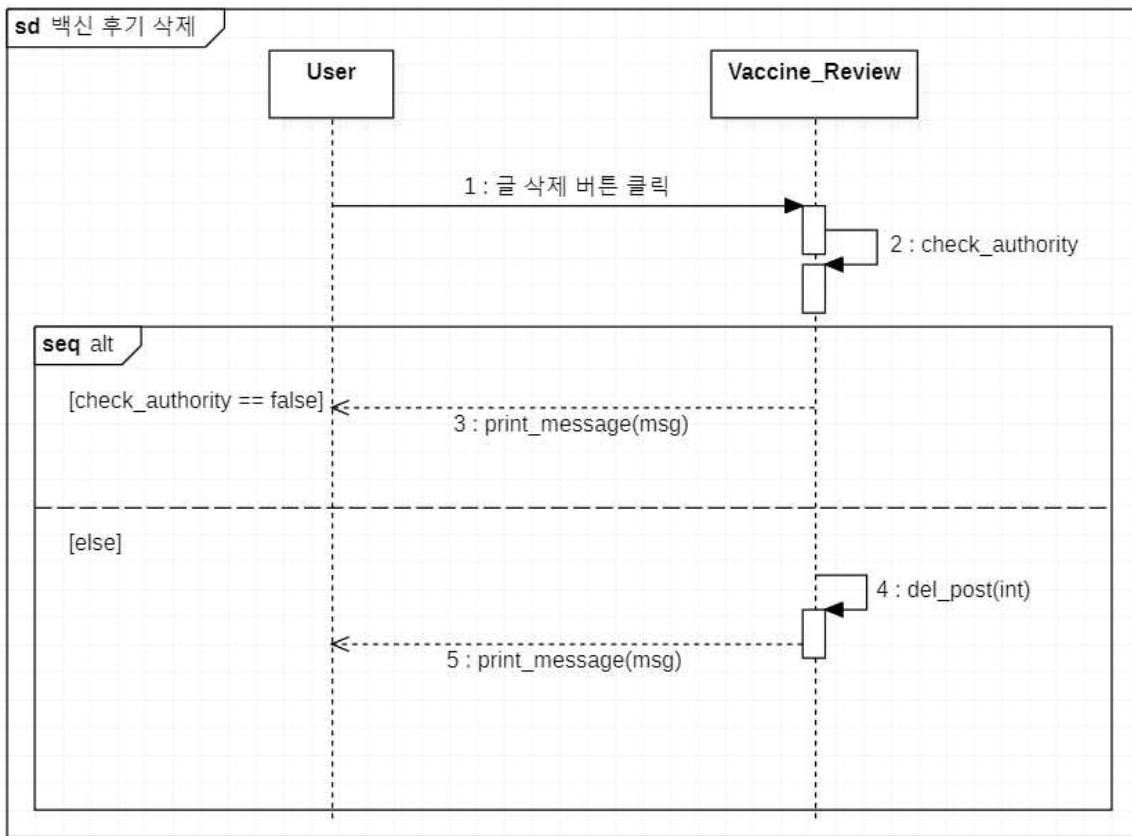
사용자가 게시판에 백신 후기 글쓰는 것을 나타낸 Sequence Diagram이다. 글쓰기 창에 글을 쓰고 글쓰기 완료 버튼을 누르면 check_type 메소드로 백신 종류를 선택했는지 확인한다. 반환된 값이 false이면 입력된 값이 없다는 것이므로 내용이 없다는 경고창을 띠우고, 반환된 값이 true일 때 check_post 메소드로 글쓰기 창에 내용이 있는지 확인하여, true라면 upload_post함수를 이용해 글을 업로드한다.

15) 백신 후기 게시글 검색



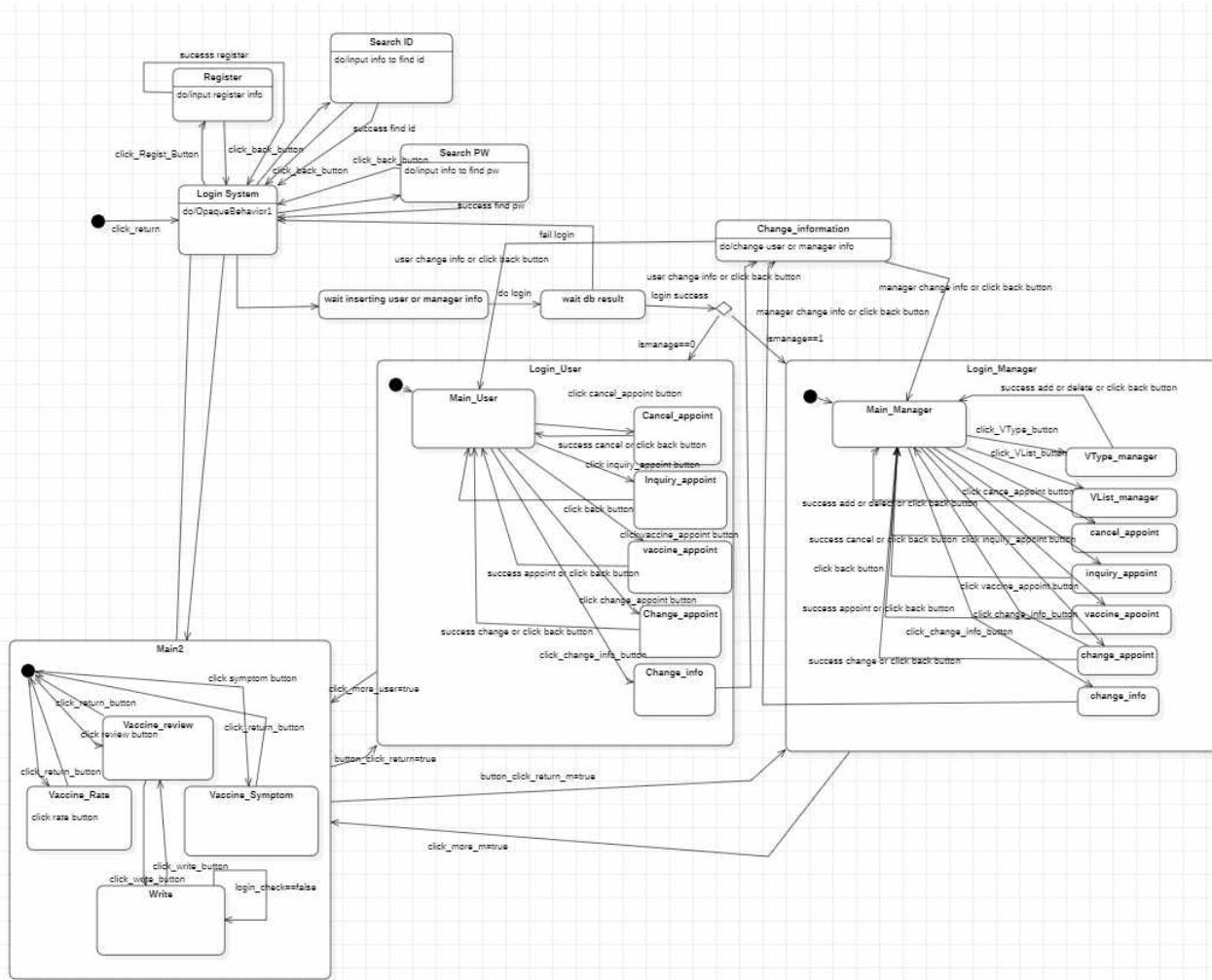
사용자가 게시판에 게시된 백신 후기 글을 조회하는 것을 나타낸 Sequence Diagram이다. 먼저 사용자가 set_type으로 백신 종류를 선택한다. 검색어를 입력했을 때는 set_Post_both로 post를 설정해 입력한 검색어가 포함된 선택한 백신 종류 후기글만 view_post 메소드를 통해 보여주고, 검색어를 입력하지 않았을 때는 set_Post_type 메소드로 post를 설정하고 선택된 백신 종류의 후기글만 view_post 메소드를 통해 보여준다.

16) 백신 후기 삭제



사용자가 게시판에 게시된 백신 후기 글을 삭제하는 것을 나타낸 Sequence Diagram이다. 먼저 사용자가 글 삭제 버튼을 클릭하면, check_authority 메소드로 사용자가 글을 삭제할 권한이 있는지 검사 후 boolean 값을 반환한다. 반환된 값이 false이면 글을 삭제할 권한이 없다는 뜻이므로 print_message 메소드로 글 삭제 권한이 없다는 메시지를 보낸다. true이면 글을 삭제할 권한이 있다는 뜻이므로 del_post 메소드로 글을 삭제하고, print_message 메소드로 글 삭제가 완료되었다는 메시지를 보낸다.

5. State machine diagram



프로그램을 정상적으로 실행하게 되면 로그인 시스템이라는 곳에서 시작하게 된다. 여기서 크게 회원가입, 메인화면2로 이동하는 기능을 수행할 수 있다. 먼저 회원가입을 하기 위해서는 로그인 기능을 수행할 수 있는 정보들을 입력하여 기능을 수행한다. 사용자와 관리자, 두 가지의 로그인 기능 중 하나로 기능을 수행하게 되면 각각의 초기 메인화면으로 들어선다. 만약 사용자가 로그인에 실패하였거나 로그인을 하지 않은 상태일 경우, ID 찾기 또는 PW 찾기 기능을 이용하여 로그인에 필요한 정보를 찾을 수 있다. 메인화면에서 사용자는 정보변경, 백신 예약 취소, 백신 예약 조회, 백신 접종예약, 백신 접종예약 변경, 메인화면2 이동 기능을 할 수 있고, 관리자는 여기에 추가로 백신 재고의 추가/삭제, 백신 병원 추가/삭제 기능을 선택할 수 있다.

정보변경은 관리자나 사용자로 로그인이 되어있을 때 사용할 수 있고 또한 사용자나 관리자로부터 개인 관련된 정보를 받기에 DB에 연결이 되어야만 한다. 백신 예약 취소, 조회, 변경 기능은 백신 접종예약이 되어있어야만 사용할 수 있다. 백신 접종예약

기능은 병원과 백신 DB에 연결이 되어있어야 하고, 정상적으로 작동이 되면 병원과 백신 종류, 접종 시간을 선택할 수 있다. 백신 재고와 백신 병원을 추가/삭제할 때도 역시 DB에 연결이 되어있어야 사용할 수 있다.

메인화면2에서는 다른 사람들의 백신 후기를 보거나 후기 작성 및 후기 삭제가 가능하다. 여기서 후기 삭제는 본인의 글만 삭제할 수 있다. 후기를 볼 때 원하는 후기를 검색하여 볼 수 있다. 그리고 후기를 등록할 때는 내용이 있어야만 작성할 수 있다.

사용자/관리자는 로그아웃을 통해 프로그램의 접속을 종료할 수 있다.

6. User interface prototype

1) 로그인 화면

백신 예약 프로그램

YUVID

ID	<input type="text" value="abcdxyz"/>	<input type="button" value="로그인"/>
PW	<input type="password" value="●●●●●●●●"/>	

[회원가입](#) | [아이디찾기](#) | [비밀번호찾기](#)

COVID-19 알아보기

2) 회원가입 및 정보수정 화면

ID**abcdxyz****중복 확인****PW****●●●●●●●****PW 확인****●●●●●●●****이름****홍길동****주민등록번호****3010101234567****전화번호****01012345678****가입****취소**

회원가입 화면의 프로토타입으로, 아이디 작성 부분을 지우고 회원정보수정의 화면으로 사용가능하다.

3) 아이디 찾기 화면

이름

홍길동

전화번호

01012345678

찾기

취소

4) 비밀번호 찾기 화면

ID 이름 전화번호

5) 비밀번호 변경 화면

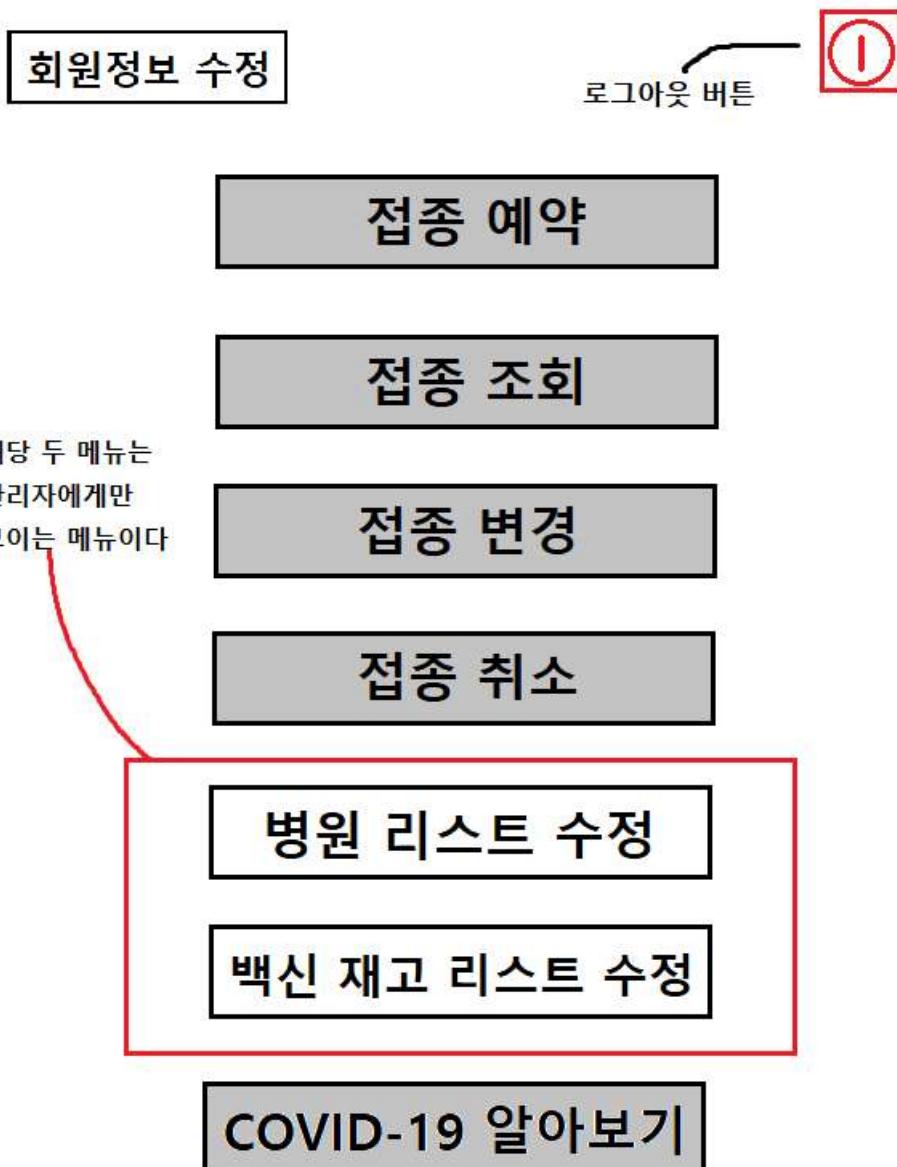
새 PW [●●●●●●●●]

새 PW 확인 [●●●●●●●●]

[변경]

[취소]

6) 메인화면1



7. 백신 예약 화면

대구 

영대병원 

화이자 | MM-DD / hh:mm 

예약

취소

8) 예약 확인 화면

홍길동님의 접종일은

영대병원

YYYY-MM-DD

hh:mm 입니다

확인

백신 예약을 확인하는 화면의 프로토타입으로, 아래의 확인 버튼 대신 ‘예약 취소’, ‘뒤로 가기’ 버튼으로 바꾸어 예약 취소화면으로 사용 가능하다.

9) 백신 후기 게시판 화면

뒤로

글쓰기

게시글

게시글

•
•
•
•
•

게시글

백신 이름 

검색어

검색

10) 백신 후기 글쓰기 화면

백신 이름

검색어

후기 내용

[게시](#)

[뒤로](#)

7. Implementation requirements

YUVID를 사용하기 위한 요구사항은 다음과 같다.

1. H/W requirements

- ◎ CPU: Intel i3+
- ◎ Memory: 1GB+
- ◎ Storage: DB 100G+ / Client 500mb+
- ◎ Network: Connected

2. S/W requirements

- ◎ OS: Windows 7+
- ◎ Implementation Language : eclipse(Java)
- ◎ DB : Oracle Database, JDK 8+

3. Nonfunctional requirement

◎ Modularity

시스템의 여러 부분은 비슷한 기능을 공유하고 있다. 이러한 부분들은 재사용성을 높여서 최적화 및 유지보수에서 효율을 높인다.

◎ Accuracy

시스템은 DB에서 데이터를 받아 사용한다. 따라서 DB에서 항상 사용자가 원하고 정확한 값을 반환해야 한다.