

## YU info(임시 이름)

Student No.	21812137
Name	심 준 서
E-mail	ws7523@naver.com

## [ Revision history ]

Revision date	Version #	Description	Author
29/05/2024	0.4	Class diagram 작성	
06/06/2024	0.4	Class diagram 수정	
07/06/2024	1.0	1차 완성	

= Contents =

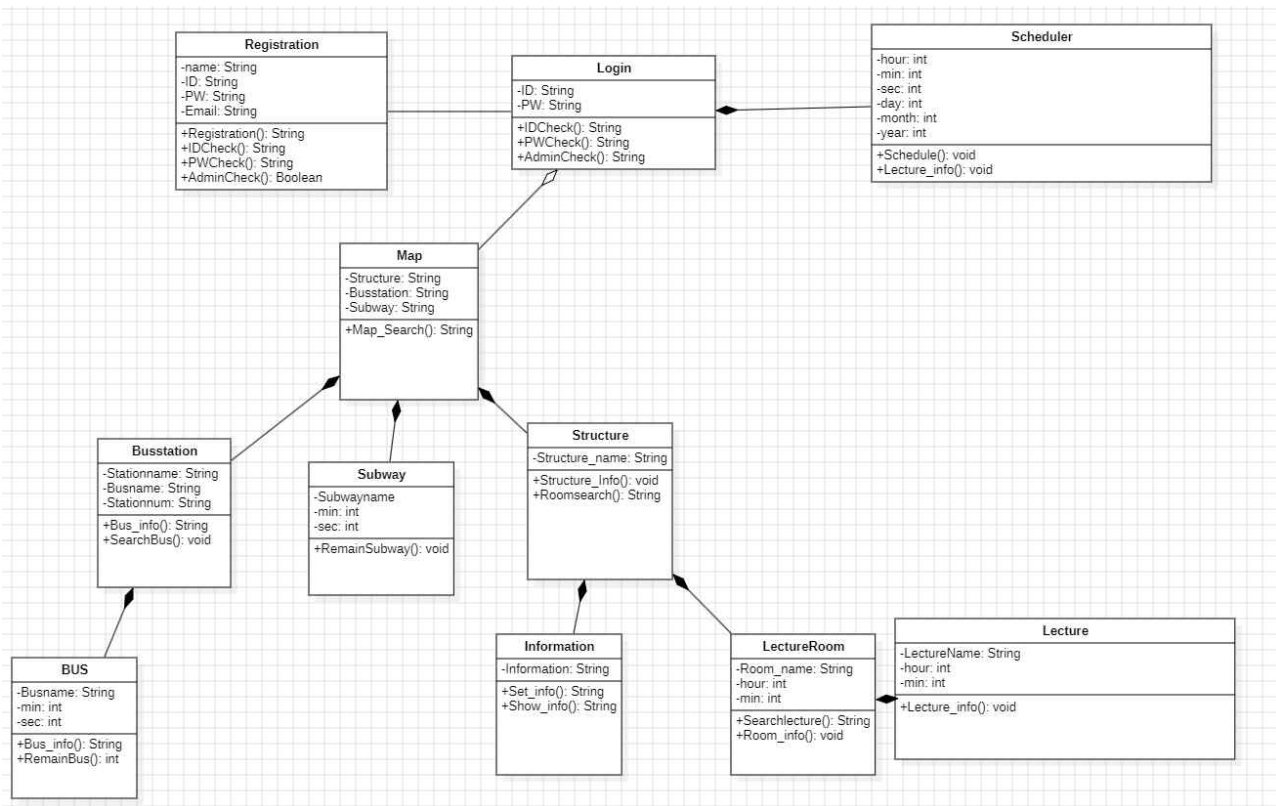
1. Introduction .....	
2. Class diagram .....	
3. Sequence diagram .....	
4. State machine diagram .....	
5. Implementation requirements .....	
6. Glossary .....	
7. References .....	

## 1. Introduction

영남대학교는 다른 대학교들과 비교하여 캠퍼스 크기가 넓은 속해있다. 그로 인하여 학교엔 길이 많고 건물도 많아 길을 잃기도 쉬운 편이다. 그러므로 자신의 강의가 여러 개의 건물에서 이루어진다면 머무를 장소도 찾기 힘들고 또 갔더니 아직 강의가 끝나지 않아 복도에서 기다리는 경우 또한 잦다. 그리고 강의가 끝난 이후 집으로 가는 학생들은 현재 건물로부터 지하철, 버스 정류장 등 귀가를 위해 타고 다니는 대중교통을 이용하기 위해 돌아가는 길 또한 오래 걸리는 경우가 많다. 그래서 보통 학생들은 건물과 길을 외워두고 강의실 또한 앞에 강의가 있었는지, 지하철 시간표 등을 외우고 다닌다. 이를 알아보려면 여러 개의 프로그램과 앱, 사이트 등으로 나뉘어 새 학기 새 건물에 강의를 듣게 되거나 복학후 학교를 다니게 되면 약간의 어려움이 동반한다. 이를 일부나마 해소하고자 영남대학교에 대한 정보를 하나로 뭉치자 생각하여 이 프로그램을 만들게 되었다. 이 프로그램을 통해 복학생이나 신입생들이 학교에 대한 여러 가지 잡다한 정보를 얻어 조금이라도 편하게 학교를 다니게 되었으면 좋겠다.

이번 보고서에서는 3번째 디자인 파트의 보고서로 class diagram과 sequence diagram을 통한 시스템의 구현과 기능에 대해서 확정하고 구체적으로 디자인할 계획이다. 이를 통해서 소스코드에서의 구현을 도울 것이다.

## 2. Class diagram



### 1) Registration

#### (1) Attribute

- name : String : 사용자의 이름입니다.
- ID : String : 사용자의 ID입니다.
- PW : String : 사용자의 비밀번호입니다.
- Email : String : 사용자의 이메일입니다.

#### (2) Methods

- Registration() String : ID, PW, 이름, 이메일을 확인한다.
- IDCheck() :String : ID가 중복인지 확인한다.
- PWCheck() :String : PW가 적합한지 확인한다.
- AdminCheck() :Boolean : 관리자인지 아닌지 확인한다.

## 2) Login

### (1) Attribute

- ID : String : 사용자의 ID입니다.
- PW : String : 사용자의 비밀번호입니다.

### (2) Methods

- IDCheck() : String : ID가 적합한지 확인한다.
- PWCheck() :String : PW가 적합한지 확인한다.
- AdminCheck() : Boolean : 관리자인지 아닌지 확인한다.

## 3) Scheduler

### (1) Attribute

- Hour : int : 시간
- Min : int : 분
- Sec : int : 초
- Day : int : 일
- Month : int : 달
- Year : int : 년

### (2) Methods

- Schedule() : void : 스케줄을 가져옵니다.
- Lecture\_info() : void : 강의 정보를 확인한다.

## 4) Map

### (1) Attribute

- Structure : String : 지도 내의 건물의 이름입니다.
- Busstation : String : 지도 내의 버스 정류장의 이름입니다.
- Subway : String : 지도 내의 지하철 이름입니다.

### (2) Methods

- Map\_Search() : String : 지도 내에서 검색을 합니다.

## 5) Busstation

### (1) Attribute

- Stationname : String : 버스 정류장의 이름입니다.
- Busname : String : 버스의 이름입니다.
- Stationnum : String : 버스 정류장 번호

### (2) Methods

- Bus\_info() : String : 버스의 정보를 보여준다.
- SearchBus() : String : 버스를 검색한다.

## 6) Subway

### (1) Attribute

- Subwayname : String : 지하철 역의 이름입니다.
- hour : int : 시간
- min : int : 분

### (2) Methods

- RemainSubway() : void : 지하철 도착까지 남은 시간

## 7) Structure

### (1) Attribute

- Structure\_name : String : 건물의 이름입니다.

### (2) Methods

- Room\_Search() : String : 강의실을 검색합니다.
- Stucture\_info() : void : 건물의 정보를 보여줍니다.

## 8) Information

### (1) Attribute

- Information : String : 건물의 정보를 나타냅니다.

### (2) Methods

- Set\_info() : String : 관리자가 정보를 입력 또는 제거합니다.
- Show\_info() : 건물의 정보에 대해 보여줍니다.

## 9) LectureRoom

### (1) Attribute

- Room\_name : String : 강의실의 이름입니다.
- Hour : int : 시간입니다.
- Min : int : 분입니다.

### (2) Methods

- Search\_Lecture() : String : 강의를 검색합니다.
- Room\_info() : void : 강의실에 대한 정보를 보여줍니다.

## 10) Lecture

### (1) Attribute

- LectureName : String : 강의의 이름입니다.
- Hour : int : 시간입니다.
- Min : int : 분입니다.

### (2) Methods

- Lecture\_info : 강의에 대한 정보입니다.

## 11) Bus

### (1) Attribute

- Busname : String : 버스의 이름(번호) 입니다.
- min : int : 남은 분을 나타낼 숫자입니다.
- sec : int : 남은 초를 나타낼 숫자입니다.

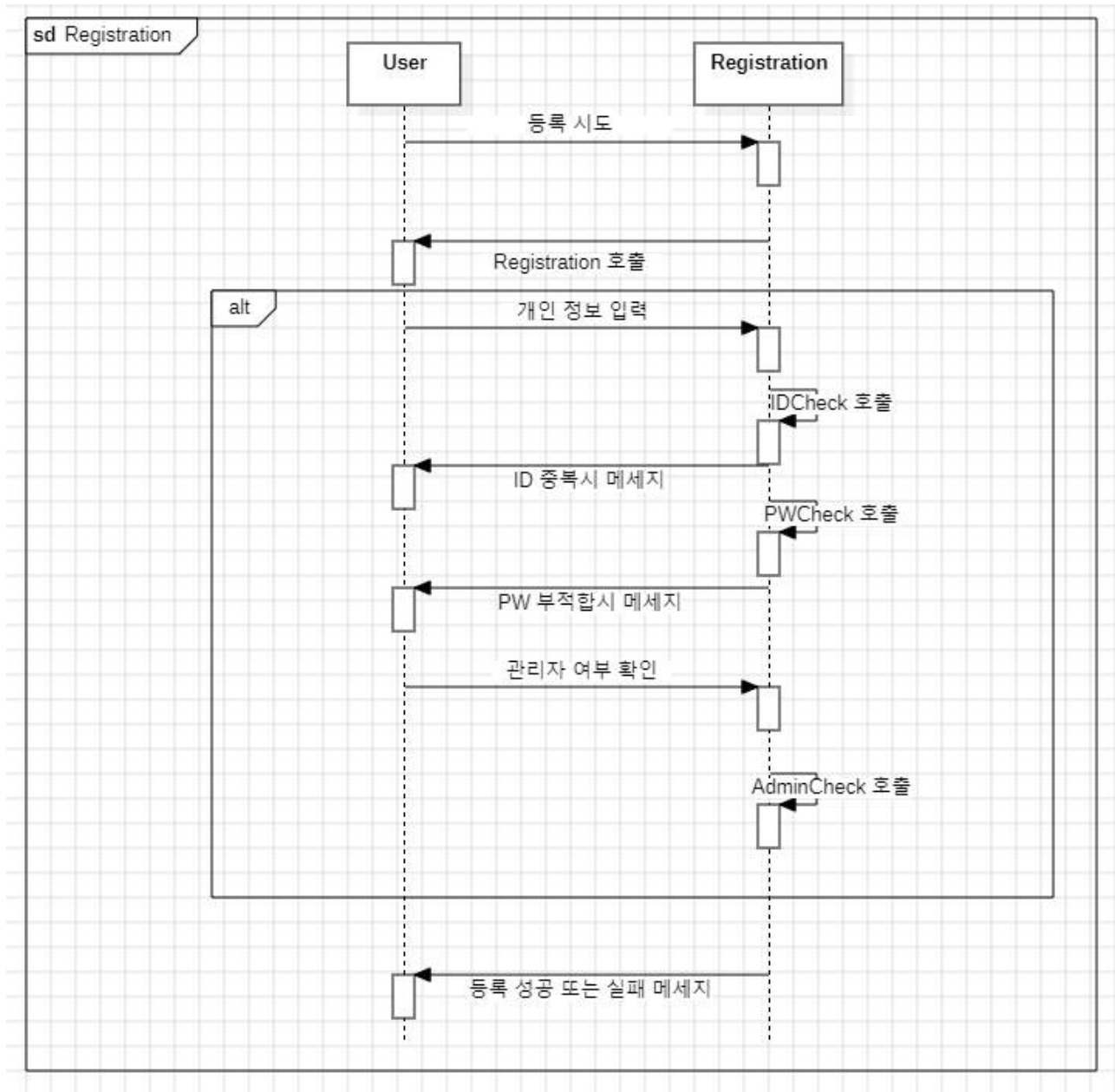
### (2) Methods

- Bus\_info() : String : 버스의 정보를 나타냅니다.
- Remain\_Bus() : int : 버스의 남은 시간을 나타냅니다.



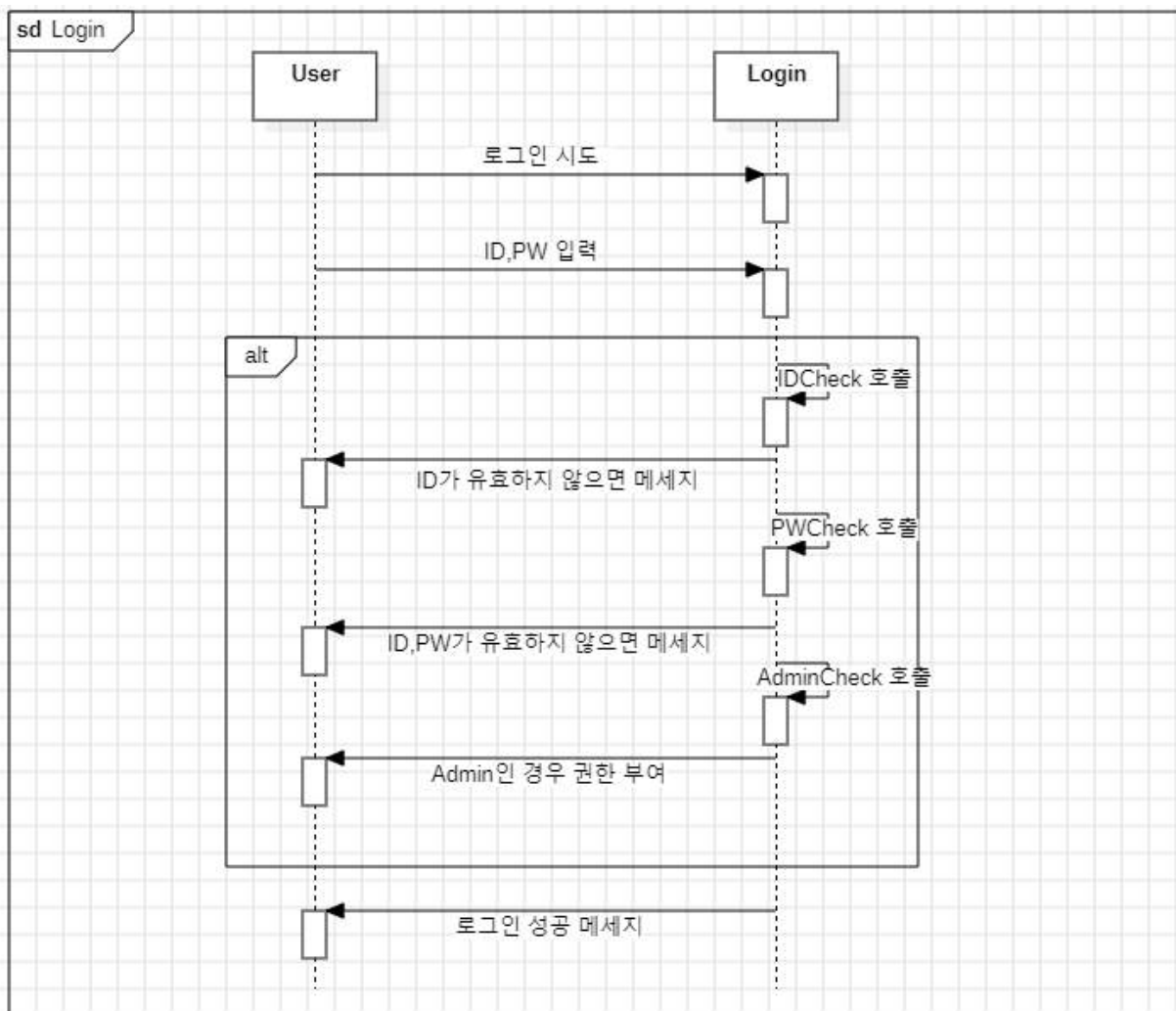
### 3. Sequence diagram

#### 1) Registration



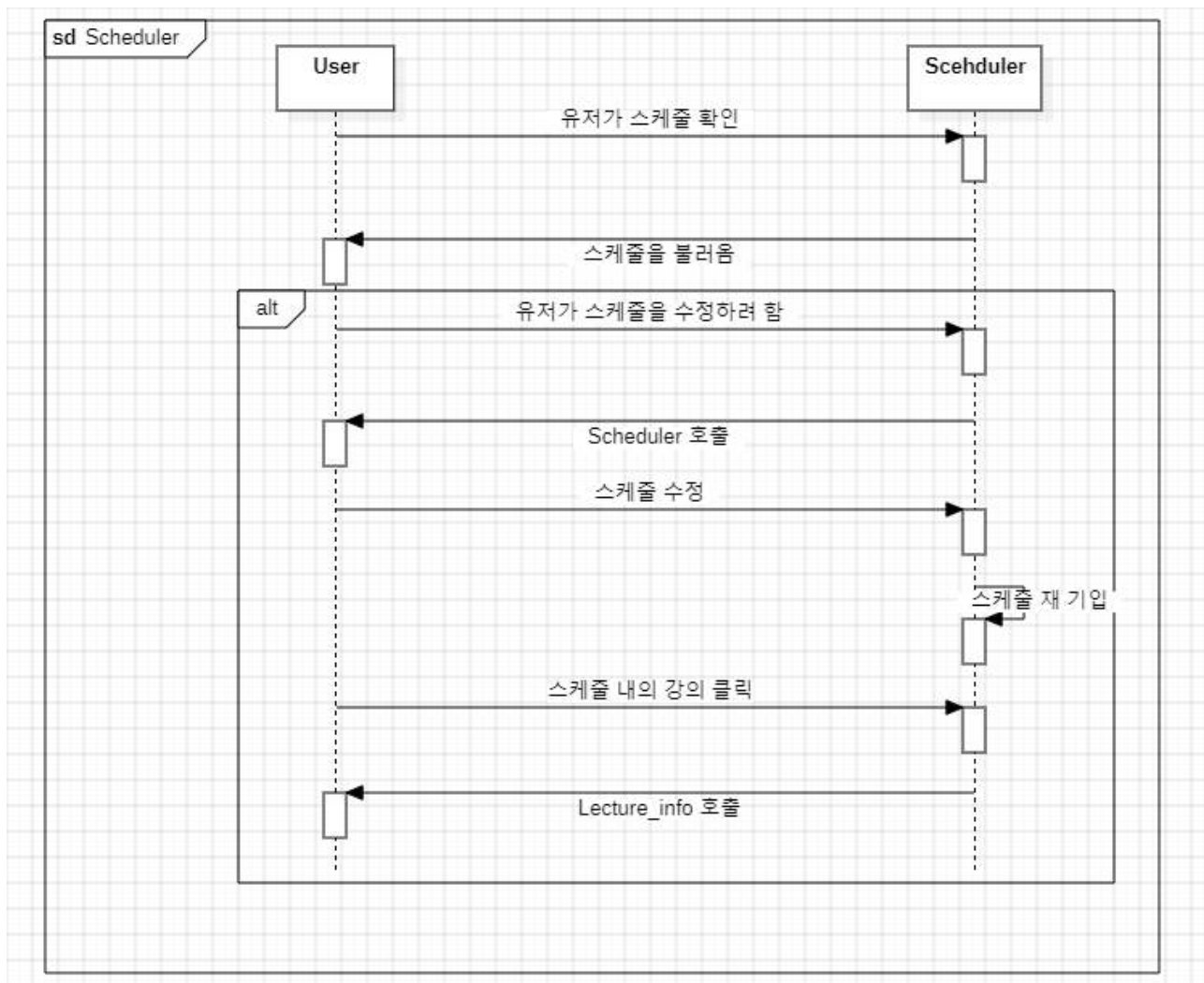
위 그림은 Registration의 기능을 수행할 경우를 표현한 Sequence Diagram이다. 사용자가 등록을 시도하면 Registration()을 호출하고 사용자가 개인 정보를 입력하면 내부에서 IDCheck() , PWCheck() , AdminCheck()를 통해서 가입 여부를 확인후 메시지를 보내준다.

## 2) Login



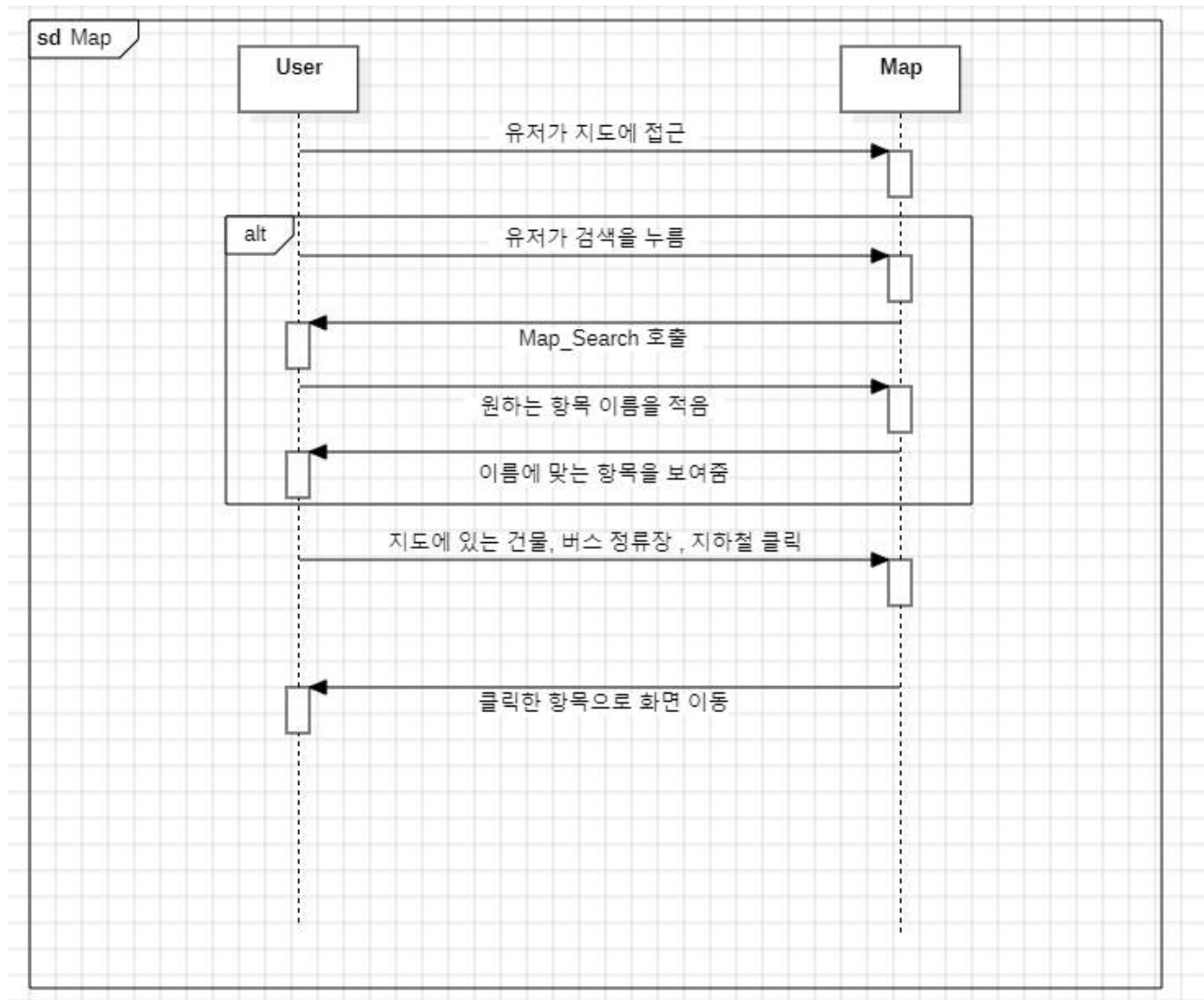
위 그림은 Login의 기능을 수행할 경우이다. 사용자가 로그인 시도후 ID, PW를 입력하며 내부에서 IDCheck() 호출하여 ID가 유효한지 체크 , 이후 PWCheck 를 호출하여 ID에 맞는 PW인지 체크한 뒤 부적합하면 메시지, 적합하면 AdminCheck를 통해 이 계정이 관리자의 계정인지 확인후 권한 부여한다. 마지막으로 로그인 성공 메시지를 보낸다.

### 3)Scheduler



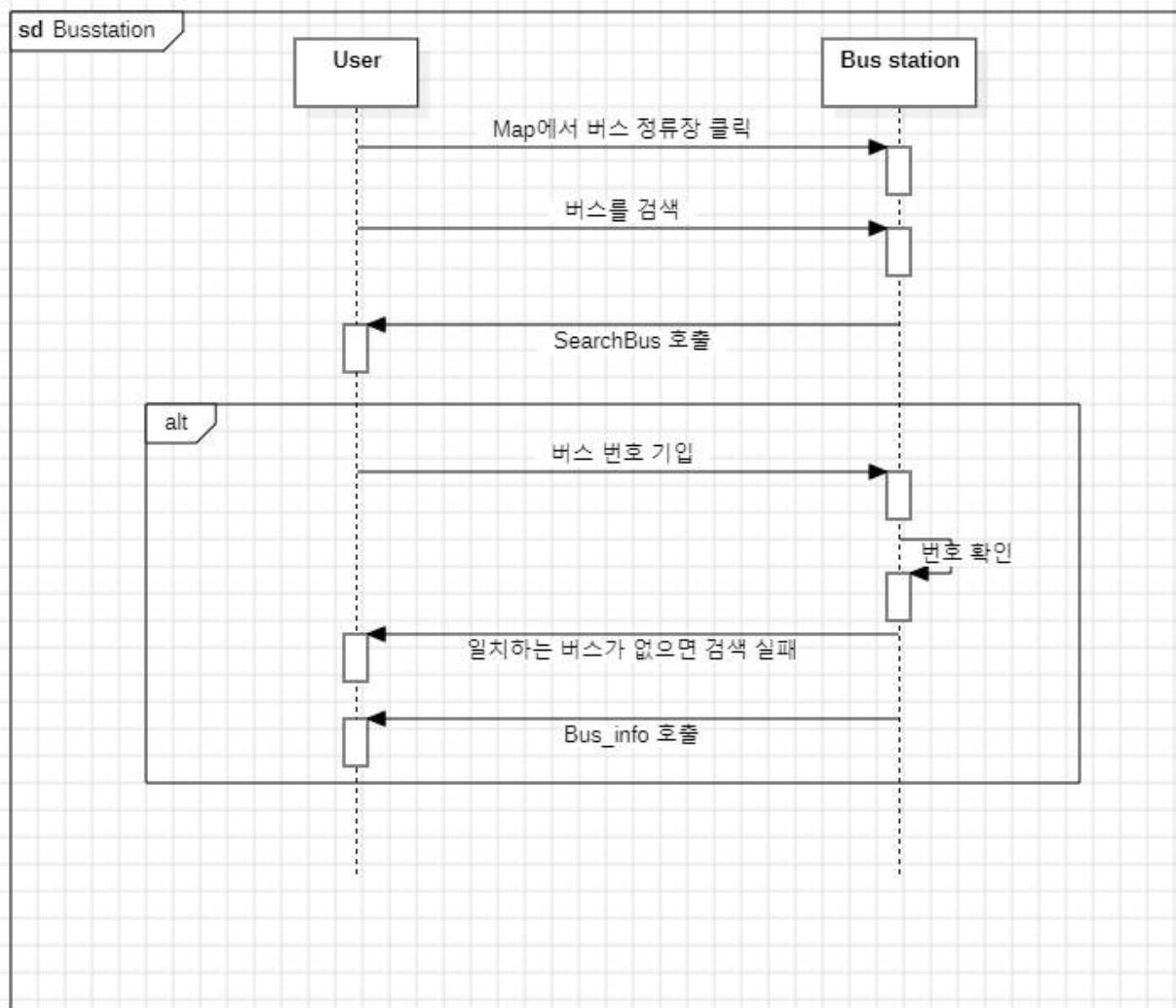
위 그림은 유저가 스케줄을 관리하기 위해 사용되는 기능을 수행할 경우이다. 유저에게 먼저 스케줄을 불러온 뒤 수정하려 하면 Scheduler() 를 호출해 거기에 유저가 스케줄을 수정하고 수정 완료시 재기입 한다. 또한 스케줄 내에 강의를 클릭하면 강의에 대한 정보를 불러와준다.

#### 4)Map



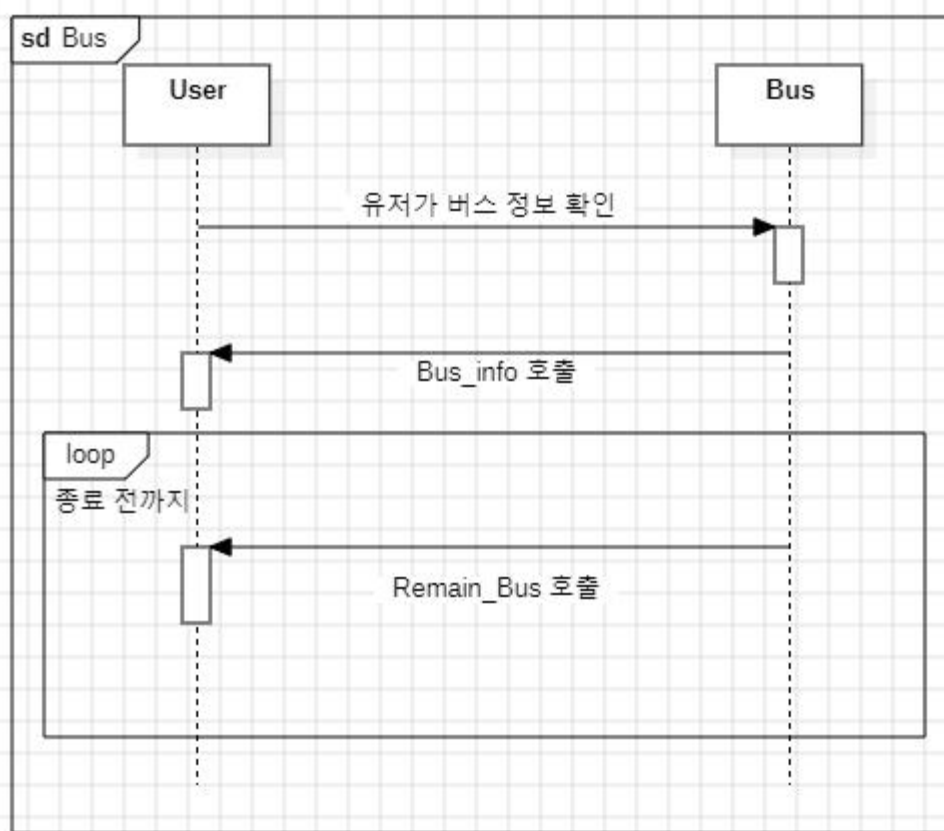
위 그림은 사용자가 지도에 접근했을 경우 기능을 수행하는 그림이다. 먼저 Map이 유저의 화면에 지도를 띄워준다. 지도에서 사용자가 검색을 누를 경우 Map\_Search()를 호출하고 유저는 거기에 원하는 이름을 적으면 그 이름에 맞는 항목을 보여준다. 또한 지도에 있는 건물 , 버스 정류장 , 지하철을 클릭하면 그 항목으로 화면을 이동한다.

## 5)Busstation



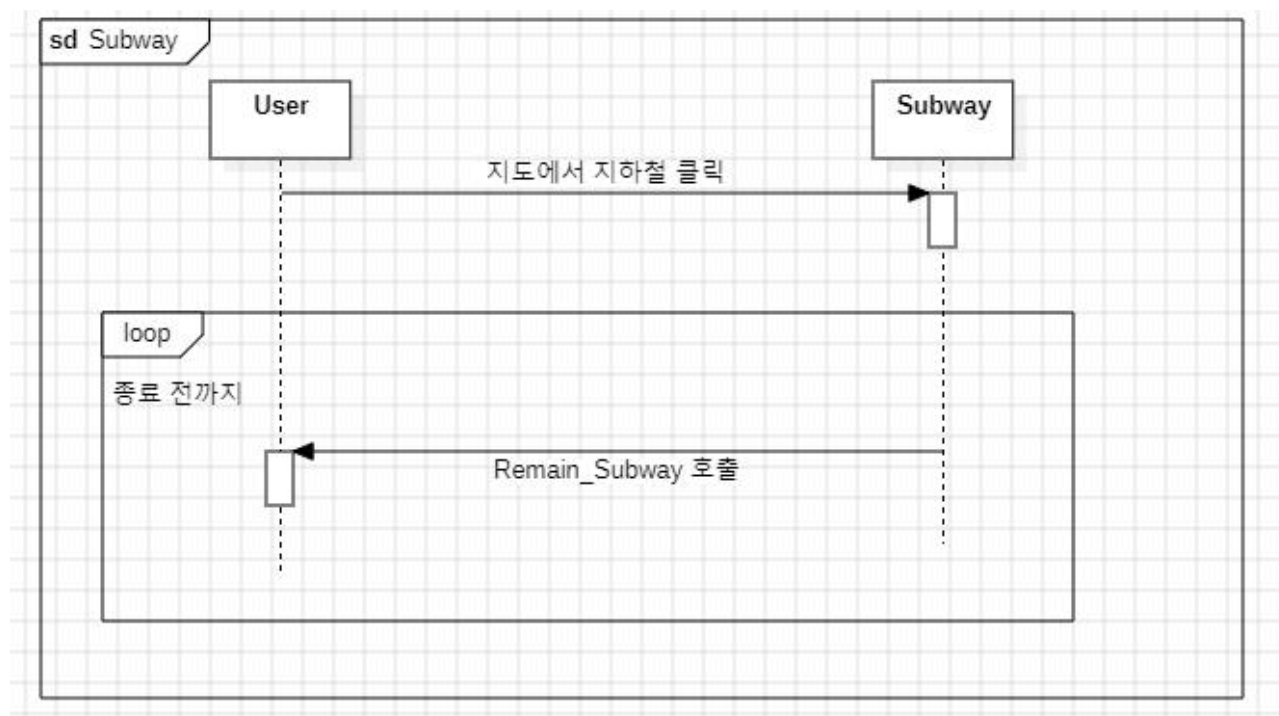
위 그림은 사용자가 지도에서 버스 정류장 항목을 누른 경우 기능을 수행하게 되는 경우이다. 먼저 사용자가 버스를 검색하면 SearchBus()를 호출해주는데 여기에 버스 번호를 기입하여 검색하거나 그냥 버스 목록을 보여준다. 검색할 경우 일치하는 버스가 없으면 검색 실패 메시지를 보여주고 있다면 Bus\_info()를 호출한다.

## 6)Bus



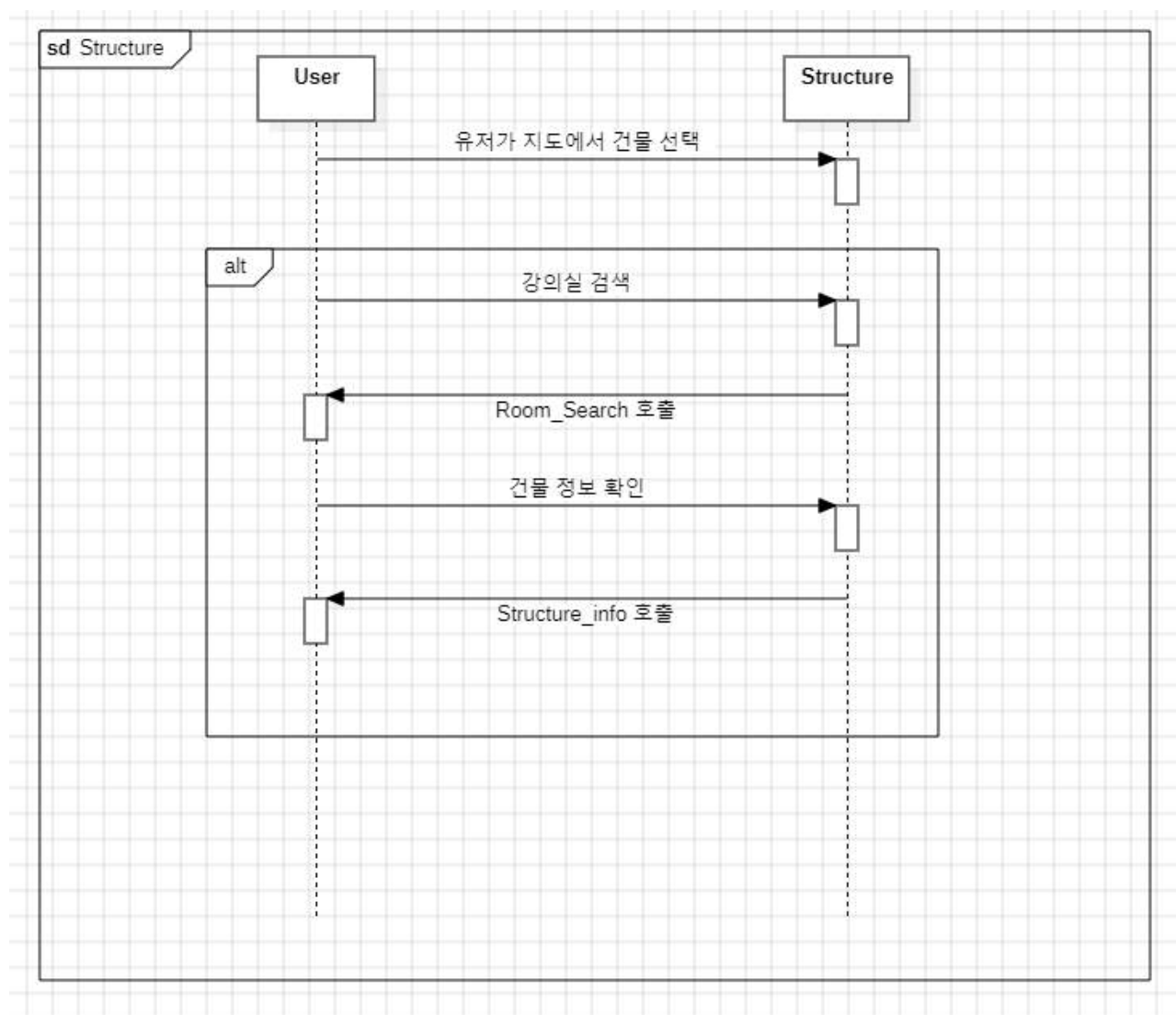
위 그림은 Busstation 에서 Bus\_info가 호출된 경우 기능을 수행한다. Bus\_info가 수행되면 유저가 종료하기 전까지 Remain\_Bus()를 호출하여 남은 시간을 보여준다.

## 7) Subway



위 그림은 지도에서 Subway 항목을 누른 경우 수행된다. 수행될 경우 그 지하철에서 열차 도착까지 남은 시간을 Remain\_Subway()를 통해 유저가 종료할 때 까지 호출한다.

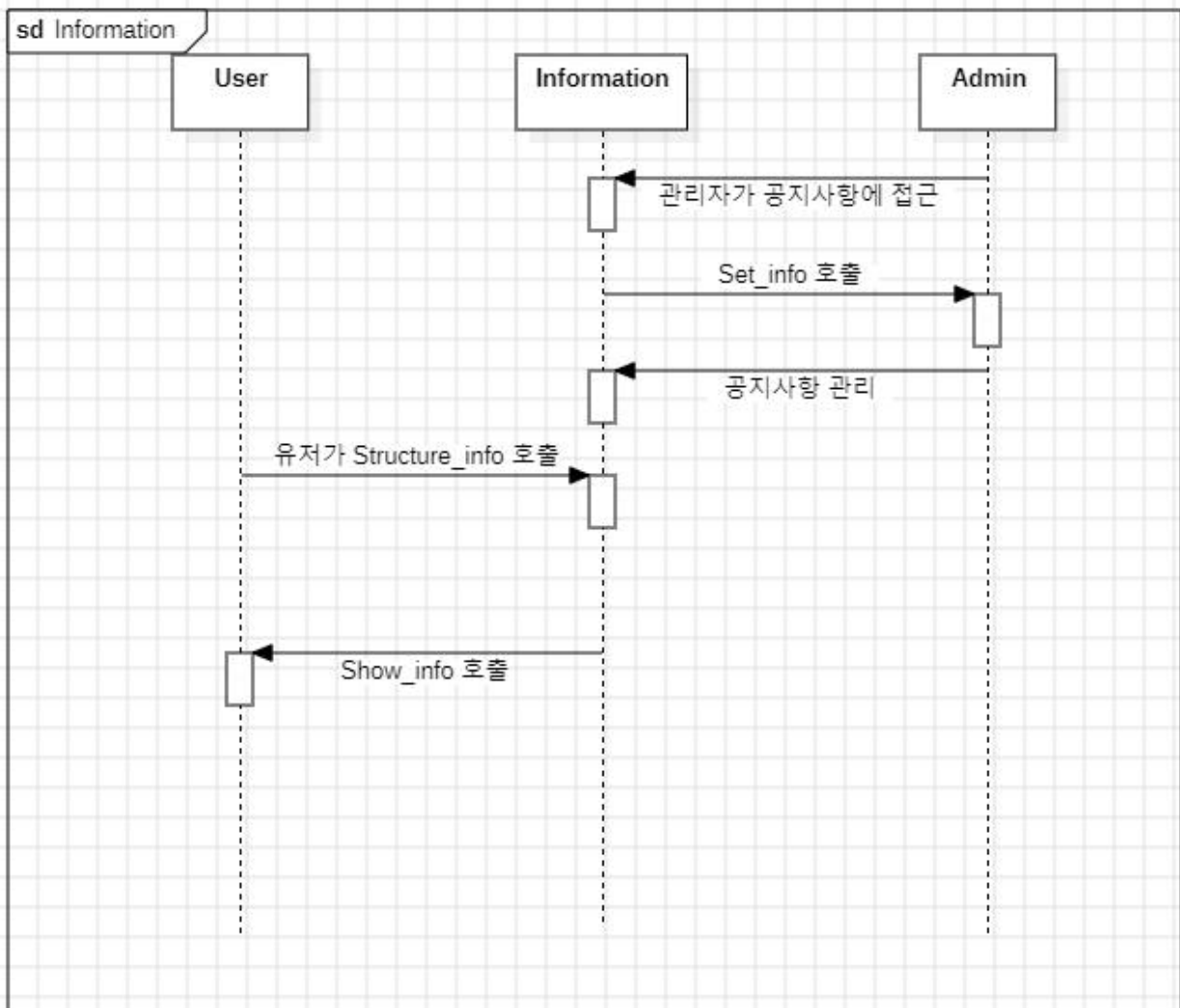
## 8)Structure



위 그림은 유저가 지도에서 건물을 클릭한 경우 수행되는 기능을 표현했다.  
 건물에서 강의실을 검색할 경우 Room\_Search()를 호출하고 건물 정보를 확인하려고  
 하면 Structure\_info를 호출해 준다.

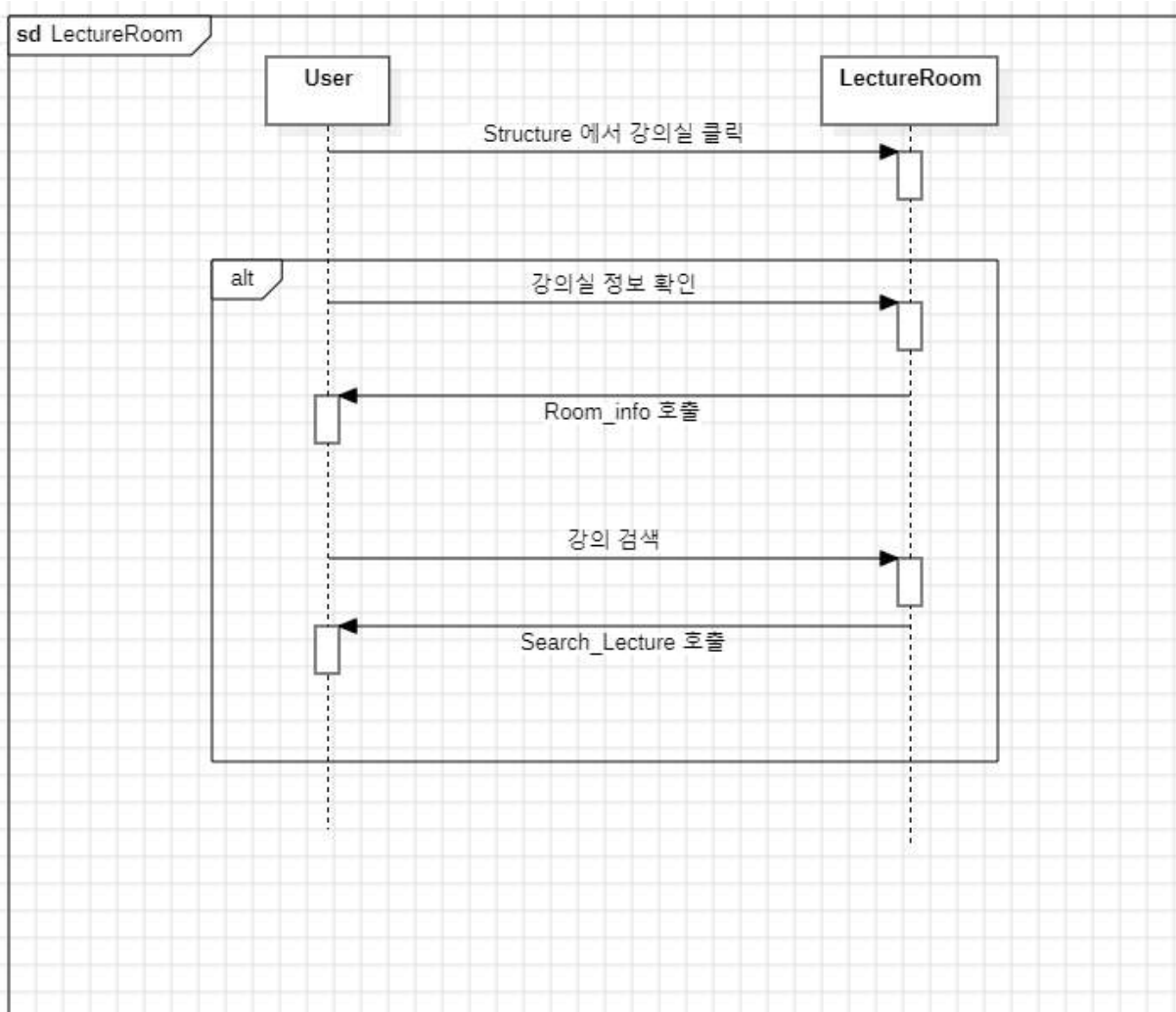


## 9)Information



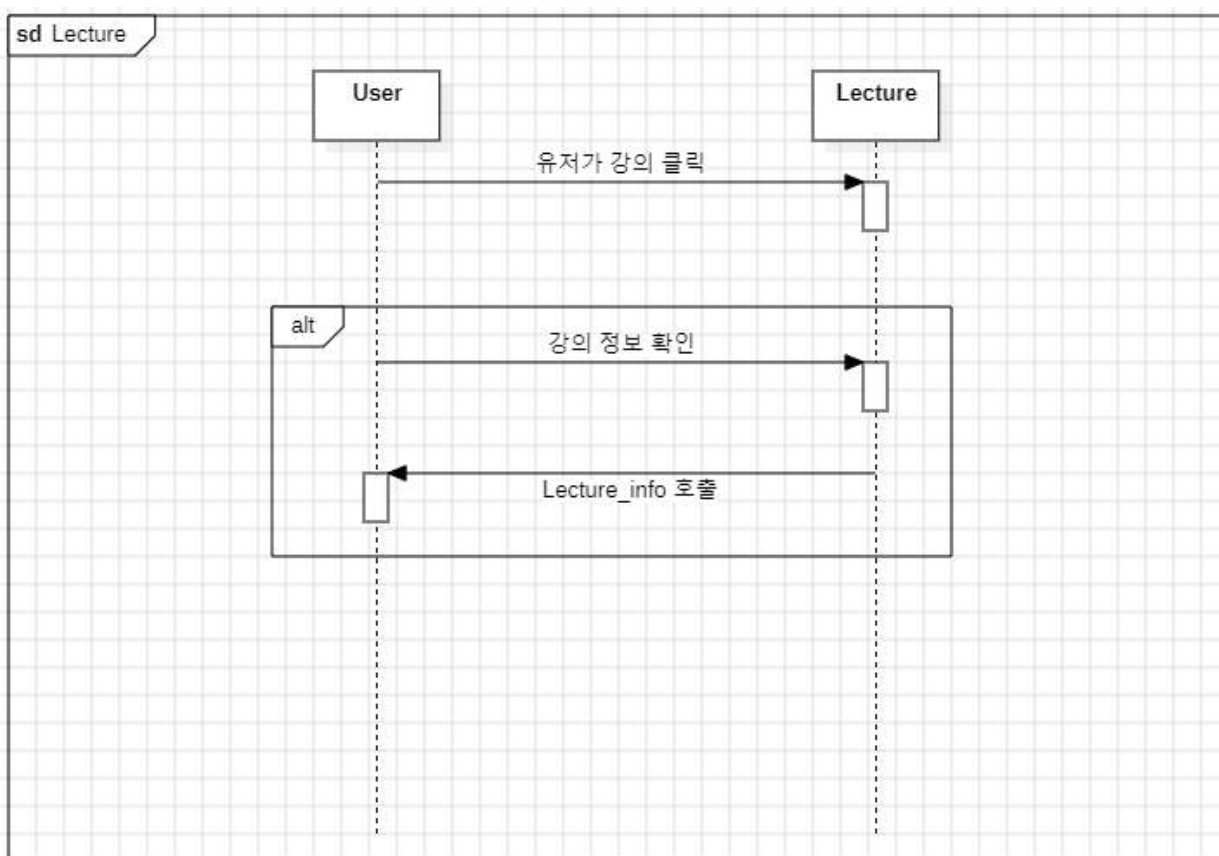
위는 유저가 Structure에서 Structure\_info()를 호출한 경우에 수행되는 기능을 표현한 그림이다. 우선 Admin이 먼저 공지사항에 접근해서 Set\_info를 통해 공지사항을 관리하여 정보를 추가 , 제거를 해둔 상태이고 그 이후 유저가 호출하였을때 Show\_info()를 다시 호출받아서 정보를 받는다.

## 10)LectureRoom



위 그림은 사용자가 Structure 에서 강의실을 클릭한 경우 수행되는 기능을 표현했다. 유저는 강의실 정보를 확인하기 위해 Room\_info를 호출하거나 강의 검색을 위해 Search\_Lecture를 호출 가능하다.

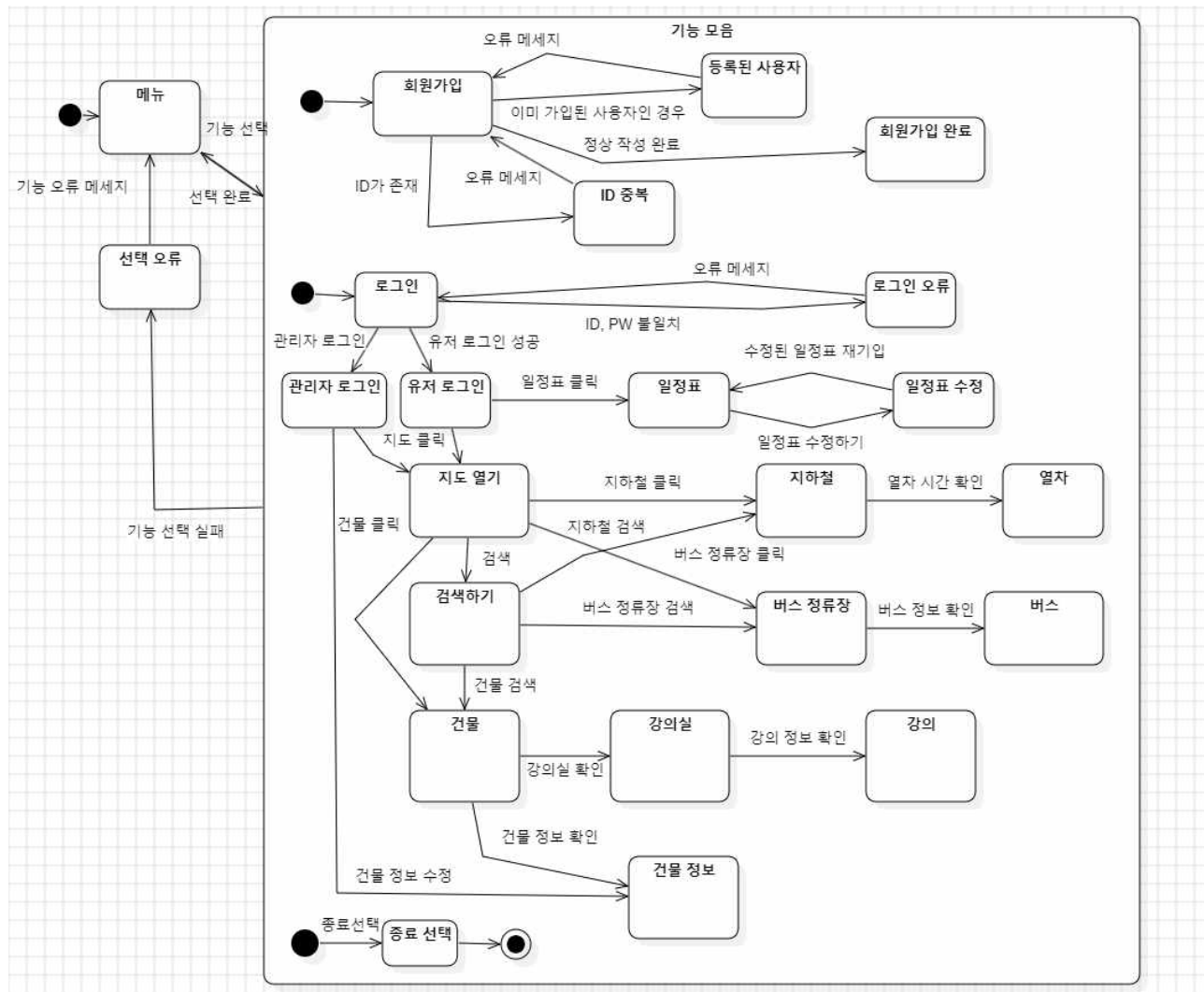
## 11)Lecture



위 그림은 유저가 강의를 클릭한 경우 기능을 수행을 표현했다.

유저가 강의 정보 확인을 원할시 Lecture\_info를 호출하여 강의에 대한 정보를 확인 가능하다.

## 4. State machine diagram



회원 가입후 로그인을 통해 유저는 대부분의 기능을 이용 가능하고 관리자는 실행 기능에선 건물 정보 수정만 추가로 기능을 가지고 있다.

둘 모두 지도를 통해 지하철, 버스 정류장, 건물에 접근이 가능하며 접근 이후로도 기능을 통해 정보를 얻는다.

## 5. Implementation requirements

인터넷이 연결되는 스마트폰  
용량은 2GB 이내로 예상

## 6. Glossary

Attribute : 멤버 변수를 가리킨다.

Method : 멤버 함수, 프로그래밍 언어에서 클래스나 함수가 될 예정

## 7. References

강의 자료 : Structural Modeling I,II , Behavior modeling I,II