

영남대학교 학생 주차장 현황



Student No	22012204
Name	배재성
E-mail	jae089265@yu.ac.kr

[Revision history]

Revision date	Version #	Description	Author
2024.06.05	1.0.0	First Draft	배재성

= Contents =

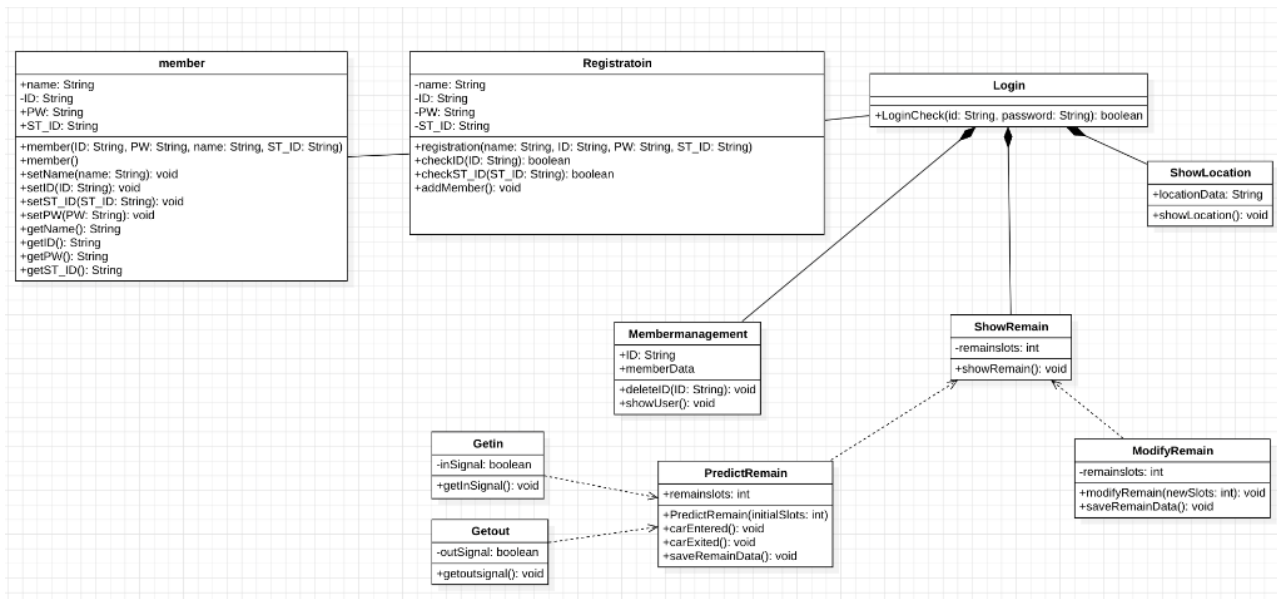
1. Introduction	
2. Class diagram	
3. Sequence diagram	
4. State machine diagram	
5. Implementation requirements	
6. Glossary	
7. References	

1. Introduction

작년부터 우리학교에 있는 무료주차장 중에 공대학생주차장과 동문학생주차장에 차단기가 생겼다. 외부인들이 학교축제기간이나 벚꽃기간에 무료주차장을 많이 이용하여 정작 학생들이 주차할 자리가 없었지는 경우가 많았는데 주차차단기가 생김으로써 학생증(RFID)을 가진 학생들만 주차를 할 수 있게 하였고 이로 인해 수업이나 공부를 하러 온 학생들이 차량주차에 어려움을 겪는 일이 줄어들었다. 하지만 그럼에도 불구하고 학생들 대다수가 수업이 많은 월요일이나 평일오후의 경우 학생들만으로도 주차자리가 없어서 학생들이 끝에서 회차를 하여 나오는 경우가 많다. 그래서 이러한 불편함을 해소하고자 차단기가 열리고 닫히는 신호를 받아와서 주차장에 자리가 얼마나 남았는지 알려줄 수 있도록 하기 위해서 이 프로그램을 구상하게 되었다.

아래는 시스템 개발의 세번째 단계인 Design에 관한 내용으로, 시스템 구현에 직접적
을 관여하는 모든 요소들의 윤곽을 확정하고 구체적으로 디자인하는 과정을 다룹니다.
본 문서의 모든 세부사항은 실제 구현 시 소스 코드에서의 용례와 완벽히 일치하는 것
을 목표로 합니다.

2. Class diagram



1. Member 클래스

1) Attributes:

name: String: 사용자 이름

ID: String: 사용자 ID

PW: String: 사용자 비밀번호

ST_ID: String: 사용자 학생 ID

2) Methods:

member(ID: String, PW: String, name: String, ST_ID: String): 생성자, 회원 정보를 초기화합니다.

member(): 기본 생성자

setName(name: String): void: 이름을 설정합니다.

setID(ID: String): void: ID를 설정합니다.

setST_ID(ST_ID: String): void: 학생 ID를 설정합니다.

setPW(PW: String): void: 비밀번호를 설정합니다.

getName(): String: 이름을 반환합니다.

getID(): String: ID를 반환합니다.

getPW(): String: 비밀번호를 반환합니다.

getST_ID(): String: 학생 ID를 반환합니다.

2. Registration 클래스

1) Attributes:

name: String: 사용자 이름

ID: String: 사용자 ID

PW: String: 사용자 비밀번호

ST_ID: String: 사용자 학생 ID

2) Methods:

registration(ID: String, PW: String, name: String, ST_ID: String): 생성자, 회원 정보를 초기화합니다.

checkID(ID: String): boolean: 입력한 ID가 이미 있는지 확인합니다.

checkST_ID(ST_ID: String): boolean: 입력한 학생 ID가 이미 있는지 확인합니다.

addMember(): void: 새로운 회원을 추가합니다.

3. Login 클래스

1) Methods:

LoginCheck(ID: String, PW: String): boolean: 회원의 ID와 비밀번호를 확인합니다.

4. MemberManagement 클래스

1) Attributes:

ID: String: 회원 ID

memberData: 회원 데이터

2) Methods:

deleteID(ID: String): void: 회원 ID를 삭제합니다.

showUser(): void: 모든 회원 정보를 출력합니다.

5. GetIn 클래스

1) Attributes:

inSignal: boolean: 차량이 들어오는 신호

2) Methods:

getinSignal(): void: 차량이 들어오는 신호를 설정합니다.

6. GetOut 클래스

1) Attributes:

outSignal: boolean: 차량이 나가는 신호

2) Methods:

getoutSignal(): void: 차량이 나가는 신호를 설정합니다.

7. PredictRemain 클래스

1) Attributes:

remainSlots: int: 남은 주차 공간

2) Methods:

PredictRemain(initialSlots: int): 생성자, 초기 주차 공간을 설정합니다.

carEntered(): void: 차량이 들어오면 남은 주차 공간을 줄입니다.

carExited(): void: 차량이 나가면 남은 주차 공간을 늘립니다.

saveRemainData(): void: 남은 주차 공간 정보를 저장합니다.

8. ModifyRemain 클래스

1) Attributes:

remainSlots: int: 남은 주차 공간

2) Methods:

modifyRemain(newSlots: int): void: 남은 주차 공간을 수정합니다.

saveRemainData(): void: 남은 주차 공간 정보를 저장합니다.

9. ShowRemain 클래스

1) Attributes:

remainSlots: int: 남은 주차 공간

2) Methods:

showRemain(): void: 남은 주차 공간을 보여줍니다.

10. ShowLocation 클래스

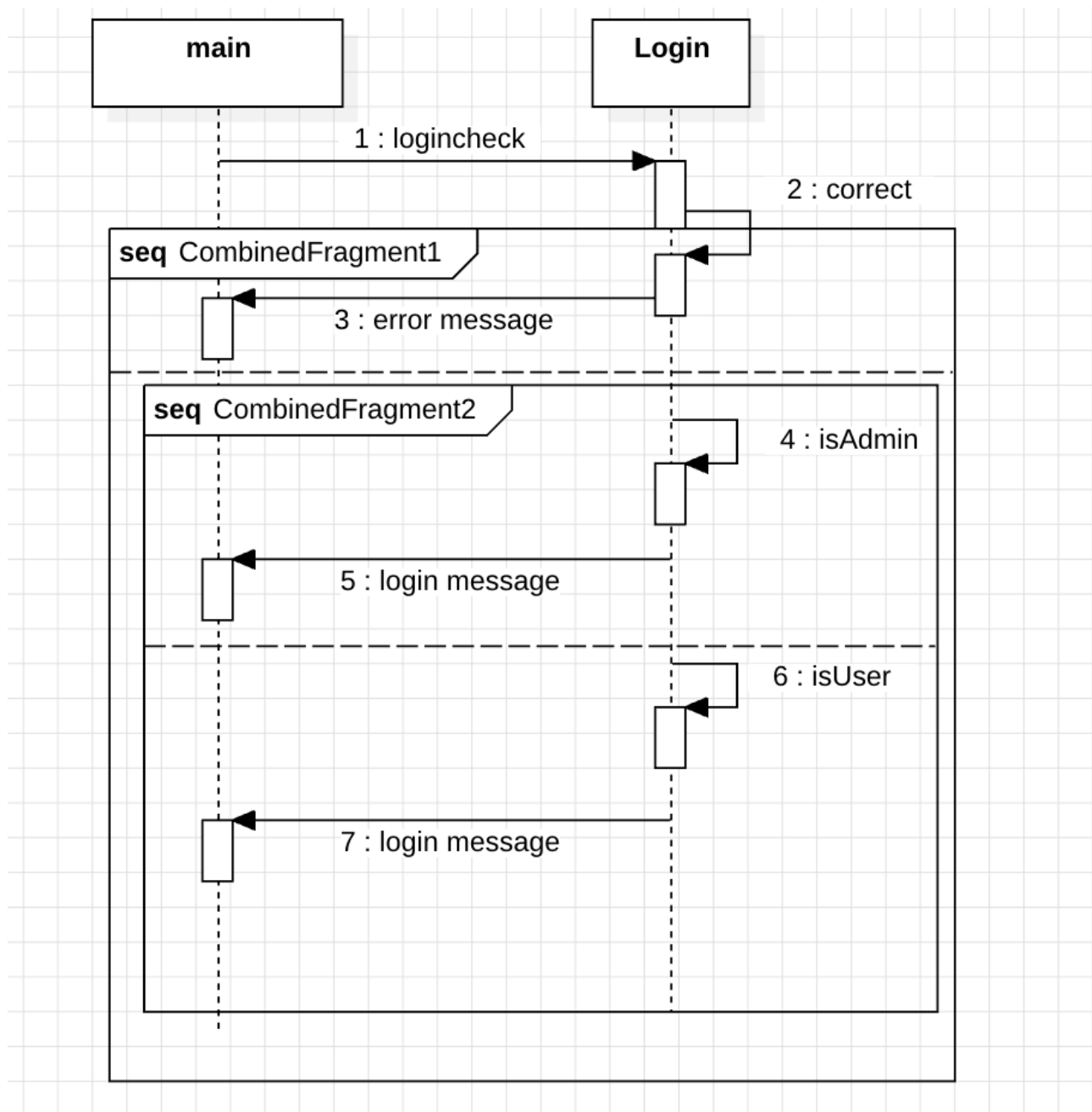
1) Attributes:

locatinData: String: 주차장 주소

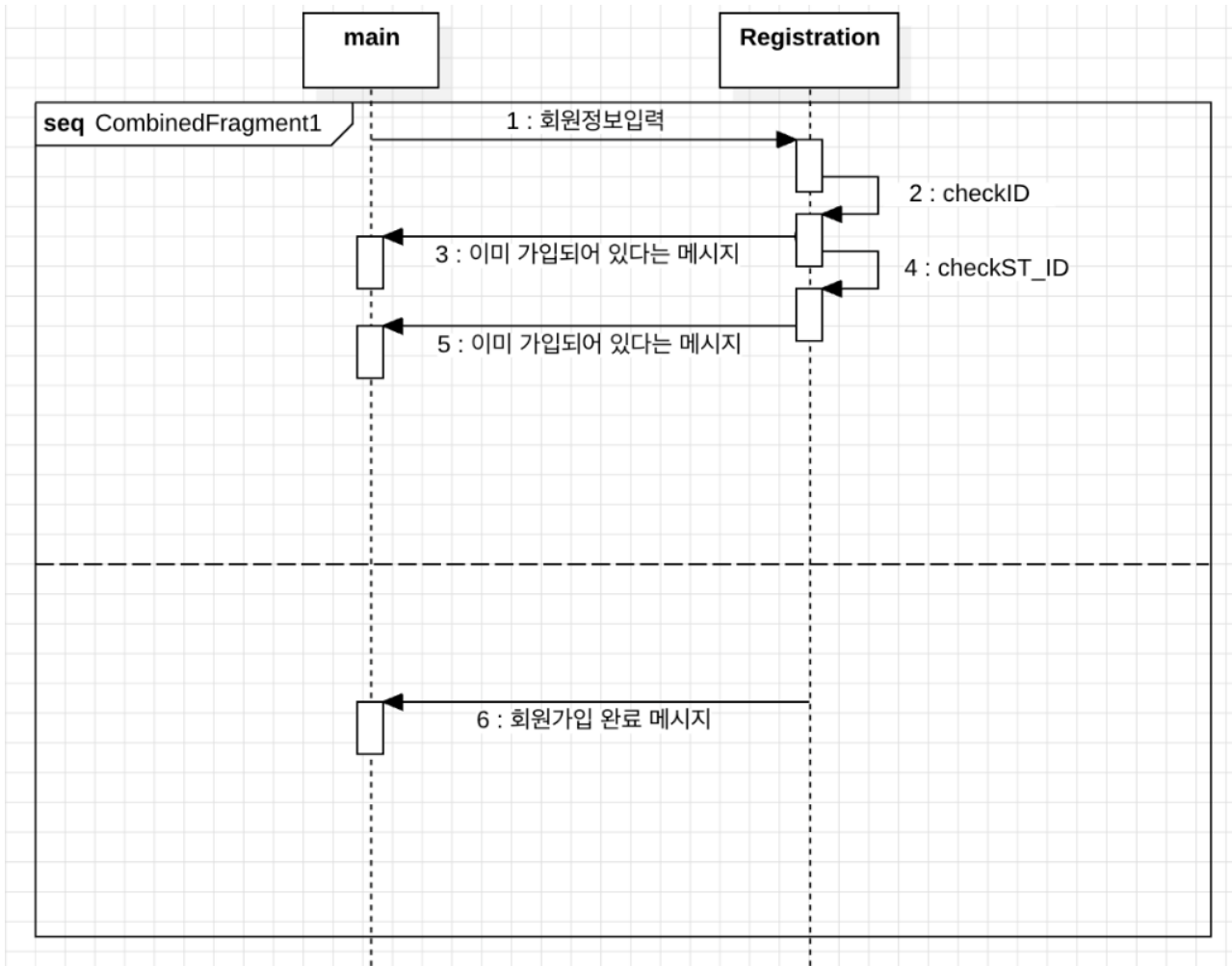
2) Methods:

showLocation(): void: 주차장의 위치를 보여줍니다.

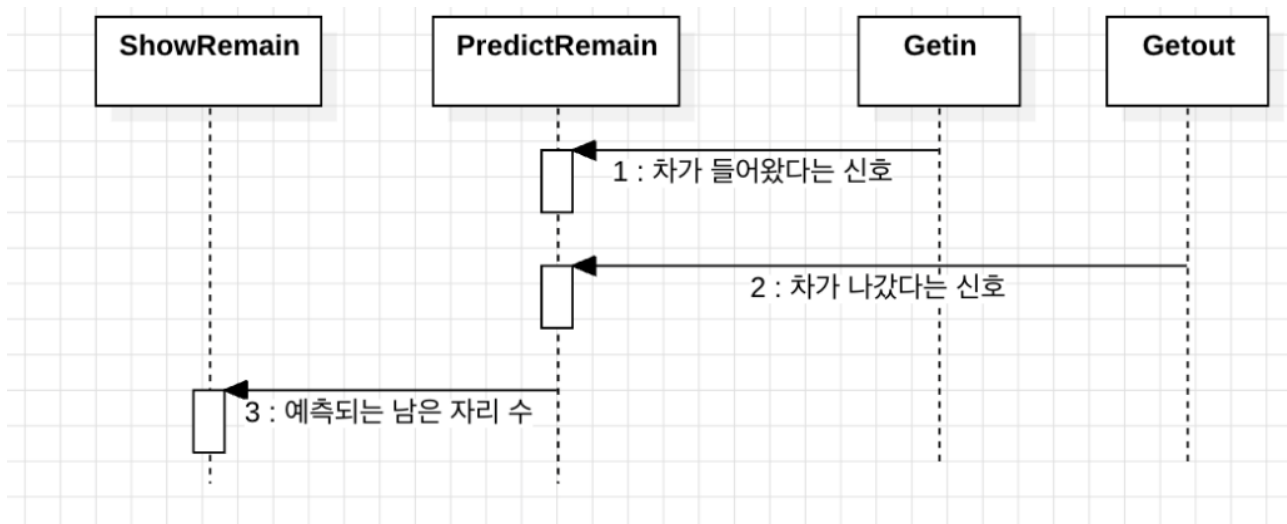
3. Sequence diagram



위 그림은 Login과정에 대한 Sequence Diagram이다. 먼저 로그인 정보가 맞는지 확인하고 틀렸다면 오류메세지를 반환하고 올바르다면 관리자인지 일반 유저인지 확인 후 로그인 메시지를 반환한다.

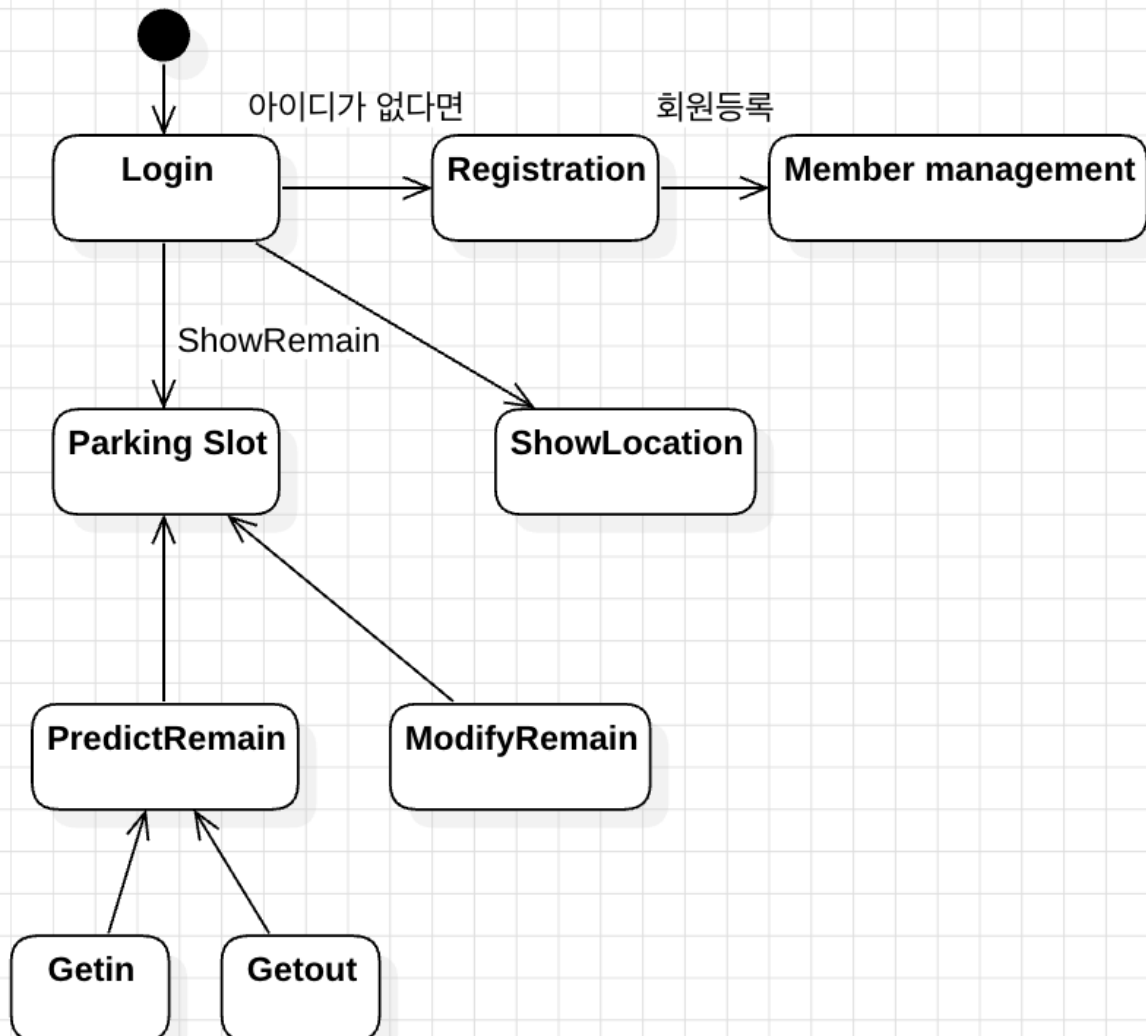


위는 회원가입의 Sequence Diagram이다. 회원정보를 입력받는데 ID와 ST_ID를 중복확인하여 중복되어 있다면 이미 가입되어 있다는 메시지를 반환하고 통과한다면 회원가입 완료 메시지를 반환한다.



위는 ShowRemain에서 남은 자리 수가 나오는 과정을 나타낸 Sequence diagram입니다. 실시간으로 정보를 받아서 현재 남아 있는 자리 수를 나타내기에 순서로 다이어그램을 그리는 것이 명확하지는 않지만 대략적으로 진행되는 틀을 보여드리고자 그렸습니다.

4. State machine diagram



위는 이 프로그램의 State machine diagram입니다. 로그인을 한뒤 ShowRemain을 통해 남은 주차장 자리수를 확인하거나 ShowLocation을 통해 주차장 위치를 확인할 수 있습니다. Getin과 Getout에서 실시간으로 차단기가 열리고 닫히는 신호를 받아서 PredictRemain을 통해 주차장에 남은 자리 수를 실시간으로 계산합니다.

5. Implementation requirements

1) H/W platform requirements

- (1) CPU : Intel PENTIUM IV 이상
- (2) RAM : 1GB 이상
- (3) HDD / SSD : 4GB 이상

2) S/W platform requirements

- (1) OS : Microsoft Window 10이상
- (2) Implemetaion Language : Python

6. Glossary

term	Description
Attribute	속성은 클래스에서 정의되는 데이터 변수로, 객체의 상태를 저장합니다. 속성은 클래스의 인스턴스마다 고유한 값을 가질 수 있습니다. 예를 들어, Member 클래스의 name, ID, PW, ST_ID는 속성입니다.
Class Diagram	클래스 다이어그램은 시스템의 클래스와 클래스 간의 관계를 시각적으로 표현한 다이어그램입니다. 클래스 다이어그램은 클래스의 속성, 메서드, 그리고 클래스 간의 상속, 연관, 의존 관계 등을 나타냅니다.
Sequence Diagram	시퀀스 다이어그램은 시스템의 객체 간에 주고받는 메시지를 시간 순서에 따라 시각적으로 표현한 다이어그램입니다. 이 다이어그램은 객체 간의 상호작용을 보여주며, 특정 기능을 수행하기 위해 객체들이 어떻게 협력하는지 설명합니다.

State Machine Diagram	상태 머신 다이어그램은 객체가 시간에 따라 겪는 상태 변화를 시각적으로 표현한 다이어그램입니다. 각 상태와 상태 간의 전이를 나타내며, 이벤트가 발생할 때 객체가 어떤 상태로 전이하는지를 보여줍니다.
Method	메서드는 클래스 내부에서 정의되는 함수로, 객체의 동작을 구현합니다. 메서드는 객체의 상태를 조작하거나 특정 기능을 수행합니다. 예를 들어, Member 클래스의 setName, getName 등은 메서드입니다.
Operation	오퍼레이션은 클래스나 인터페이스에서 정의되는 추상적인 기능을 나타내며, 객체가 수행할 수 있는 행동을 의미합니다. 연산은 메서드로 구현됩니다. 예를 들어, 클래스 다이어그램에서 getName()과 같은 연산은 실제 코드에서는 메서드로 구현됩니다.

7. References

- 1) 강의자료 : Structural Modeling II, Behavior Modeling I, II
- 2) Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994). Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley.
- 3) Fowler, M. (2003). UML Distilled: A Brief Guide to the Standard Object Modeling Language (3rd ed.). Addison-Wesley.