

# Design Specification

Edupy

소프트웨어공학개론 Team4



제출일	22.05.00s.	그룹	Team4 <b>Edupy</b>
과목	소프트웨어공학개론	담당교수	이은석 교수님
이름	양이삭	학번	2016312117
이름	김창헌	학번	2015313546
이름	손영훈	학번	2020311323
이름	송예진	학번	2019310847
이름	정동진	학번	2017314786
이름	최가은	학번	2019315437

## 목차

<b>1. Preface</b>	<b>7</b>
1.1 Objective	7
1.2 Readership	7
1.3 Document Struecutre	7
1.3.1 Preface	7
1.3.2 Introduction	7
1.3.3 System Architecture	7
1.3.4 User Management System	7
1.3.5 Curriculum System	7
1.3.6 Python & Web Editor System	7
1.3.7 Board System	7
1.3.8 Database Design	8
1.3.9 Testing Plan	8
1.3.10 Development Environment	8
1.3.11 Development Plan	8
1.3.12 Index	8
1.3.13 References	8
1.4 Version of the Document	8
1.4.1 Version Format	8
1.4.2 Version Management Policy	8
1.4.3 Version Update History	9
<b>2. Introduction</b>	<b>10</b>
2.1 Objective	10
2.2 Applied Diagram	10
2.2.1 UML	00
2.2.2 Class Diagram	10
2.2.3 State Diagram	11
2.2.4 Sequence Diagram	11

2.2.5 ER Diagram .....	12
<b>2.3 Applied Tool .....</b>	<b>12</b>
2.3.1 JavaScript .....	12
2.3.2 HTML .....	13
2.3.3 CSS .....	13
2.3.4 MariaDB .....	13
2.3.5 Draw.io .....	14
<b>2.4 Project Scope .....</b>	<b>14</b>
2.4.1 User Management System .....	14
2.4.2 Curriculum System .....	14
2.4.3 Python & Web Editor System .....	15
2.4.4 Board System .....	15
<b>3. System Architecture .....</b>	<b>16</b>
3.1 Objective .....	16
3.2 System Organization .....	16
3.2.1 User Management System .....	17
3.2.2 Curriculum System .....	17
3.2.3 Python & Web Editor System .....	18
3.2.4 Board System .....	18
<b>4. User Management System .....</b>	<b>19</b>
4.1 Objective .....	19
4.2 Class Diagram .....	19
A. Account .....	19
A.1. Attributes .....	19
A.2. Method .....	19
B. DB_Handler .....	20
B.1. Attributes .....	20
B.2. Method .....	20
4.3 Sequence Diagram .....	20

4.4 State Diagram.....	20
<b>5. Curriculum System.....</b>	<b>21</b>
5.1 Objective.....	21
5.2 Class Diagram.....	21
A. Problem.....	21
A.1. Attributes.....	21
A.2. Method.....	21
B. Judge.....	21
B.1. Attributes.....	21
B.2. Method.....	21
5.3 Sequence Diagram.....	22
5.4 State Diagram.....	23
<b>6. Python &amp; Web Editor System.....</b>	<b>24</b>
6.1 Objective.....	24
6.2 Class Diagram.....	24
6.3 Sequence Diagram.....	24
6.4 State Diagram.....	25
<b>7. Board System.....</b>	<b>25</b>
7.1 Objective.....	25
A. Board.....	26
A.1. Attributes.....	26
A.2. Method.....	26
B. DB_Handler.....	26
B.1. Attributes.....	26
B.2. Method.....	26
7.2 Class Diagram.....	27
7.3 Sequence Diagram.....	27
7.4 State Diagram.....	27
<b>8. Database Design.....</b>	<b>28</b>

8.1 Objective	28
8.2 ER Diagram	28
8.3 Rational Schema	28
<b>9. Testing Plan</b>	<b>32</b>
9.1 Objective	32
9.2 Testing Policy	32
9.2.1 Development Testing	32
9.2.2 Release Testing	33
9.2.3 User Testing	33
9.3 Test Case	33
9.3.1 User Management System	33
9.3.2 Curriculum System	34
9.3.3 Python & Web Editor System	36
9.3.4 Board System	37
<b>10. Development Environment</b>	<b>38</b>
10.1 Objective	38
10.2 Bootstrap	38
10.3 MariaDB	39
10.4 GitHub	39
<b>11. Development Plan</b>	<b>39</b>
11.1 Objective	39
11.2 Gantt Chart	39
<b>12. Index</b>	<b>40</b>
12.1 Objective	40
12.2 Figure Index	40
12.3 Diagram Index	40
12.4 Table Index	41
<b>13. Reference</b>	<b>42</b>

13.1 Objective.....	42
13.2 Reference.....	42

## 1. Preface

### 1.1 Objective

Preface에서는 이 문서의 예상된 독자와 문서의 전체적인 구조, 각 파트에서 서술하는 세부사항들에 대해 설명한다.

### 1.2 Readership

본 문서에서는 독자를 Edupy를 개발 및 유지보수하는 소프트웨어 엔지니어, 시스템 구조를 설계하는 시스템 아키텍처, 고객 기술 지원을 위한 팀 모두 본 문서의 독자로 설정한다.

### 1.3 Document Structure

#### 1.3.1 Preface

Preface에서는 이 문서의 예상된 독자와 문서의 전체적인 구조, 각 파트에서 서술하는 세부사항들에 대해 설명한다.

#### 1.3.2 Introduction

Introduction에서는 시스템의 설계에 사용한 UML(Unified Modeling Language)과 데이터베이스 모델의 다양한 다이어그램 작성을 위해 사용한 개발 툴을 소개한다.

#### 1.3.3 System Architecture

System Architecture에서는 타겟 시스템에 대한 높은 수준의 개요와 시스템 기능이 전체적인 분포를 서술한다. 전체 시스템의 구조는 Block Diagram을 통해 도식화하였다. 서버 시스템의 관계와 실제 배포 형태는 Block Diagram, Deployment Diagram을 사용하여 표현하였다.

#### 1.3.4 User Management System

이 절에서는 Class Diagram, Sequence Diagram, State Diagram을 통해 User Management System의 구조와 사용자 및 데이터베이스와의 상호작용을 설계한다.

#### 1.3.5 Curriculum System

Lecture System은 사용자가 수강하는 강의와 사용자가 수행하는 실력 테스트와 연습 문제 등과 같은 커리큘럼에 관련된 시스템이다. 이에 대한 Class Diagram, Sequence Diagram, State Diagram을 통해 Curriculum System의 구조를 표현하고 상세 내용을 기술한다.

#### 1.3.6 Python & Web Editor System

Python & Web Editor System은 강의 내 코드를 연습하고 연습 문제를 풀기 위한 코드 작성을 제공하는 서비스로 사용자와 관련한다. 이 절에서는 Class Diagram, Sequence Diagram, State Diagram을 통해 Python & Web Editor System의 구조와 사용자 및 데이터베이스와의 상호 작용을 설계한다.

#### 1.3.7 Board System

Board System은 서비스 사용자들의 게시판 내 글 작성, 코멘트 작성, 관리자 공지글 작성 등과 같

은 게시판 서비스로 사용자와 관련한다. 이 절에서는 Class Diagram, Sequence Diagram, State Diagram을 통해 Board System의 구조와 사용자 및 데이터베이스와의 상호 작용을 설계한다.

### 1.3.8 Database Design

Database Design에서는 요구사항 명세서에서 기술하였던 요구사항을 기반으로 데이터베이스를 설계하고 이에 대해 기술한다. 요구사항에 기반한 데이터베이스의 ER Diagram을 설명하고, 이에 대한 Relation Schema를 기술한다. Normalization 과정을 통하여 데이터베이스 간에 존재할 수 있는 중복을 방지한다.

### 1.3.9 Testing Plan

Testing Plan에서는 요구사항 명세서에서 기술한, 사용자 및 관리자 시나리오를 바탕으로 한 전체 시스템이 구성한대로 잘 실행되는지 확인한다. 또한, 발생할 수 있는 오류를 사전에 발견하기 위해 진행한다. 관련 테스트를 수행하기 위한 테스트 정책을 기술하고, 이를 설계한다. 각 Testing Plan에서는 Testing Policy와 Test Case를 설명한다.

### 1.3.10 Development Environment

Development Environment에서는 실제 시스템 개발을 위해 필요한 개발 환경과 코딩 규칙에 대해 기술한다. 개발 과정에서의 버전 관리 도구를 설명한다. 프로그래밍 과정에서 기반이 되는 규칙들에 대해 서술한다.

### 1.3.11 Development Plan

Development Plan에서는 프로젝트 개발 일정을 기술한다. Gantt Chart로 전체적인 개발 순서와 시기를 알아보기 쉽게 표현하고 현재까지의 개발 현황을 설명한다.

### 1.3.12 Index

Index에서는 문서의 인덱스들을 정리한다. 다이어그램 인덱스 및 기능 인덱스가 포함된다.

### 1.3.13 References

문서 작성에 참고한 참고문헌 목록을 기술한다.

## 1.4 Version of the document

### 1.4.1 Version Format

버전 번호는 major.minor[maintenance] 형태로 표현하며, 본 문서의 버전은 0.1부터 시작한다.

### 1.4.2 Version Management Policy

본 명세서가 수정할 때마다 버전을 업데이트한다. 다만 변경 간의 간격이 1시간 이내일 때 버전 번호를 추가적으로 업데이트하는 대신, 하나의 업데이트로 통합하여 간주한다. 새로운 부분이 추가되거나 문제의 구성이 과거 문서에 비해 커다란 변화가 있는 경우 major number를 변경한다. 이미 완성되거나 작성된 부분에 대한 변경일 경우, minor number를 변경한다. 이미 작성된 부분에 대하여 오타를 수정하거나 문서의 구조를 단순히 변경하는 경우 maintenance number를 변경한다.



### 1.4.3 Version Update History

버전이 업데이트된 기록을 기술한다.

0.0	2022-05-08	Preface, Introduction 작성
1.0	2022-05-09	System Architecture, Testing Plan, Development Environment, Development Plan 작성
2.0	2022-05-12	Curriculum System 작성
3.0	2022-05-13	Board System 작성
4.0	2022-05-14	User Management System, Python & Web Editor System, Database System, Testing Plan, Development Environment, Development Plan, Index, Reference 작성

## 2. Introduction

### 2.1 Objective

Introduction에서는 본 시스템의 설계에 사용한 UML(Unified Modeling Language)과 데이터베이스 모델의 다양한 다이어그램 작성을 위해 사용한 개발 툴을 소개한다.

### 2.2 Applied Diagram

#### 2.2.1 UML



Figure 1 Logo of UML

UML(Unified Modeling Language, 통합 모델링 언어)는 소프트웨어 공학에서 사용하는 표준화된 범용 모델링 언어로, 소프트웨어 집약 시스템의 시각적 모델을 만들기 위한 도안 표기법을 포함한다.

UML의 관리는 처음 고안한 OMG(Object Management Group)에서 진행하며 소프트웨어 산업 내 시각화의 실질적인 표준 역할을 하고 있다. 개발자의 의사소통을 원활하게 하고, stakeholder들에게 시스템을 공유하여 의견을 얻는 데 도움을 준다. UML을 기반으로 UML Diagram이 존재하며, 현재 13개 종류의 Diagram들이 존재한다.

EduPy에서 사용할 UML Diagram은 Class Diagram, Sequence Diagram, State Diagram 3가지와 데이터베이스를 표현할 ER Diagram을 활용할 예정이다.

#### 2.2.2 Class Diagram

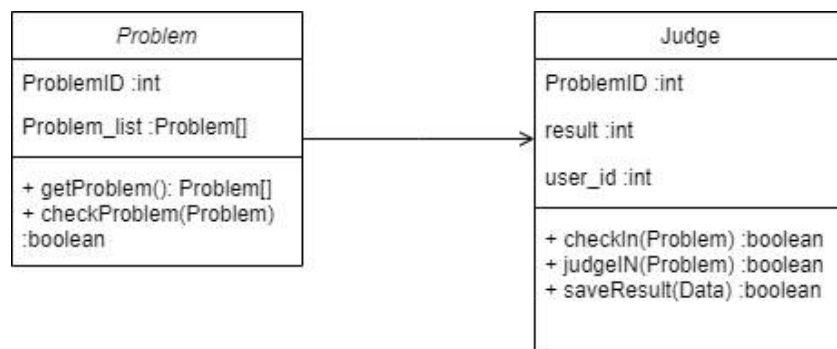


Figure 2 Example of Class Diagram

Class Diagram은 거의 모든 객체 지향 설계에서 사용하는 기본적인 Diagram이다. 시스템의 정적인 구조를 설명하며, 시스템의 논리적 구조 또한 표현한다. 하나의 클래스는 직사각형의 모양으로, 위에서부터 순서대로 클래스 이름(Name), 속성(Attribute), 함수(Method, Operation)가 포함된다. 각 클래스

스 간 상속뿐 아니라 다양한 관계를 표현할 수 있으며, Implementation 단계에서 직접적인 도움이 된다.

### 2.2.3 State Diagram

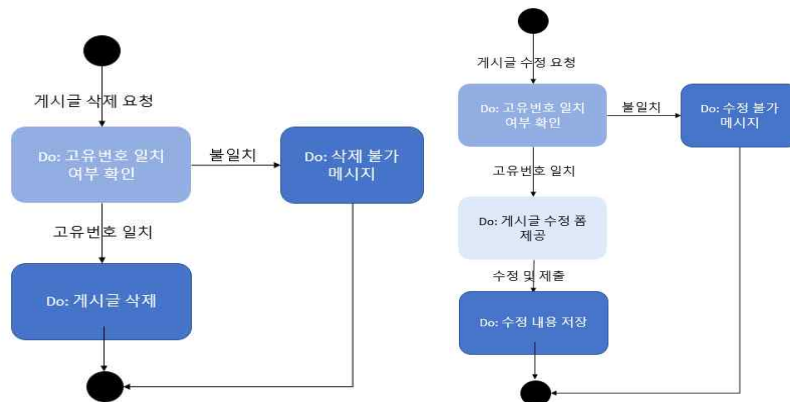


Figure 3 Examples of State Diagram

State Diagram은 시스템의 상태를 표현하거나 특정 순간의 시스템의 부분을 기술하는 다이어그램으로, 시스템을 Behavioral 관점에서 표현한다. 외부 자극의 변화나 시간에 따라 각 object 혹은 class의 동적인 행동과 상태를 기술한다.

각 다이어그램은 일반적으로 단일 class의 객체를 표현하고, 시스템을 통해 해당 객체의 여러 상태를 추적할 수 있다.

### 2.2.4 Sequence Diagram

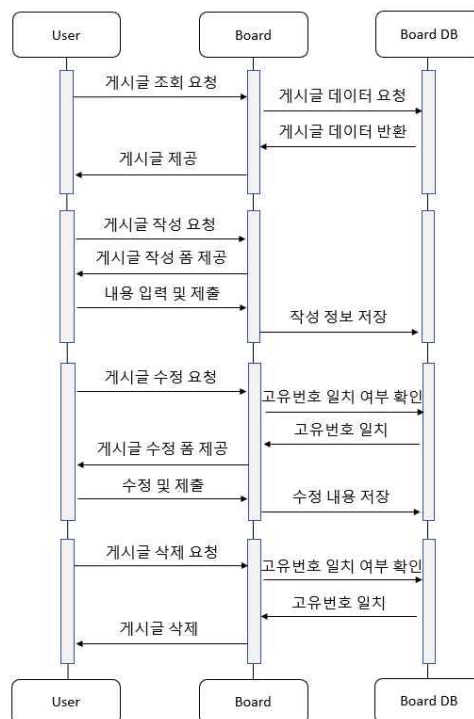


Figure 4 Example of Sequence Diagram

Sequence Diagram은 시간 흐름에 따라 시스템, object, class 간의 상호 작용을 기술한다. Scenario와 관련된 object와 class를 설명하고, scenario의 functionality가 진행될 때 객체 간 교환하는 메시지의 결과들을 기술한다. 해당 다이어그램은 개발 과정에서 시스템의 논리적 구조 내에서 use case들과 관련된다.

Sequence Diagram은 하나의 sub-system은 논리적으로 관련성이 있는 object들을 그룹으로 묶어 이들의 상호 작용의 순서를 보여주는데, object들은 상단을 가로질러 수평으로 배열되고, 시간은 수직으로 표시되어 위에서 아래로 읽을 수 있게 한다. 상호 작용은 레이블링된 화살표로 나타내며, object의 lifeline의 얇은 직사각형은 object가 시스템에서 controlling object인 시간을 나타낸다.

## 2.2.5 ER Diagram

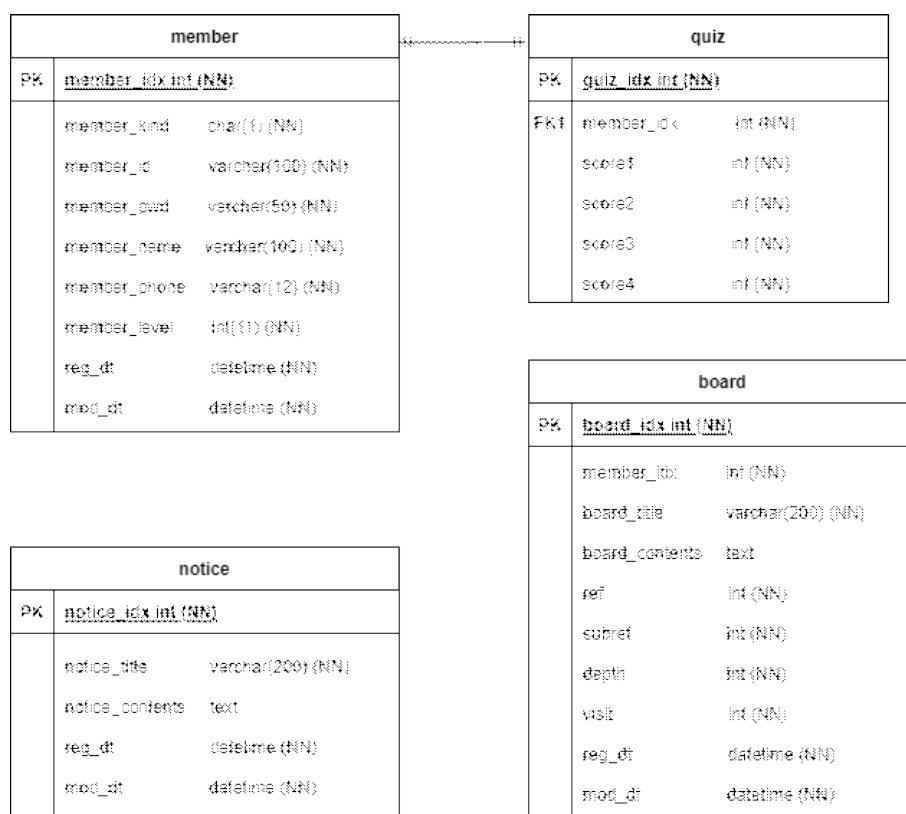


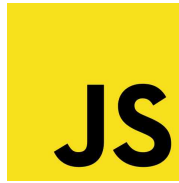
Figure 5 Example of ER Diagram

ER Diagram은 현실 세계에 존재하는 데이터와 그들 간의 관계를 사람이 이해할 수 있는 형태로 명확하게 표현하기 위해서 가장 널리 사용되고 있는 모델로 개념적 모델링의 대표적인 도구이다.

1:1, 1:N, N:M 등의 관계 타입을 표현하며, Crow-feet 표기법으로 개체 타입들을 연결하여 실선 위에 관계의 의미나 제약 조건 등을 표현한다.

## 2.3 Applied Tool

### 2.3.1 JavaScript



**Figure 6 Logo of JavaScript**

Front-end 개발에 사용한 JavaScript는 객체 기반의 스크립트 프로그래밍 언어로, 웹 브라우저 내에서 주로 사용하며 다른 응용 프로그램의 내장 객체에도 접근할 수 있는 기능을 가지고 있다.

### 2.3.2 HTML



**Figure 7 Logo of HTML**

HTML은 웹 페이지를 위한 지배적인 마크업 언어로, 제목, 단락, 목록 등과 같은 본문을 위한 구조적인 의미뿐 아니라 링크, 인용, 그 밖의 항목으로 구조적 문서를 만들 수 있는 방법을 제공한다. HTML은 JavaScript와 함께 Front-end 개발에 사용하였다.

### 2.3.3 CSS



**Figure 8 Logo of CSS**

CSS는 마크업 언어가 실제 표시되는 방법을 기술하는 스타일언어로, HTML과 XHTML에 주로 쓰인다. W3C의 표준이며, 레이아웃과 스타일을 정의할 때의 자유도가 높기 때문에 Edupy에서 HTML과 JavaScript와 함께 프론트엔드에 사용되었다.

### 2.3.4 MariaDB



**Figure 9 Logo of MariaDB**

백엔드에 사용한 MariaDB는 오픈소스의 관계형 데이터베이스 관리 시스템으로, MySQL과 동일한

소스 코드를 기반으로 하며 GPL v2 라이선스를 따른다.

### 2.3.5 Draw.io



Figure 10 Logo of Draw.io

Draw.io는 브라우저 기반의 무료 다이어그램 작성 도구로, UML diagram을 비롯하여 많은 다이어그램을 그릴 수 있는 기본 틀을 제공한다.

## 2.4 Project Scope

### 2.4.1 Overview

Edupy 시스템은 다중 사용자 간의 실시간 채팅 프로그램과 챗봇 시스템이 널리 퍼진 현재, 그 기능의 구현을 위한 멀티 스레드 소켓 기반의 채팅 프로그램을 파이썬으로 구현하기 위한 커리큘럼을 제공하고, 이와 함께 파이썬 기초 교육을 목표로 하는 학습자와 학습자들 간의 커뮤니티를 활성화하기 위한 시스템이다. Edupy 시스템은 크게 User Management System, Curriculum System, Python & Web Editor System, Board System의 세 가지로 나뉠 수 있다.

### 2.4.2 User Management System

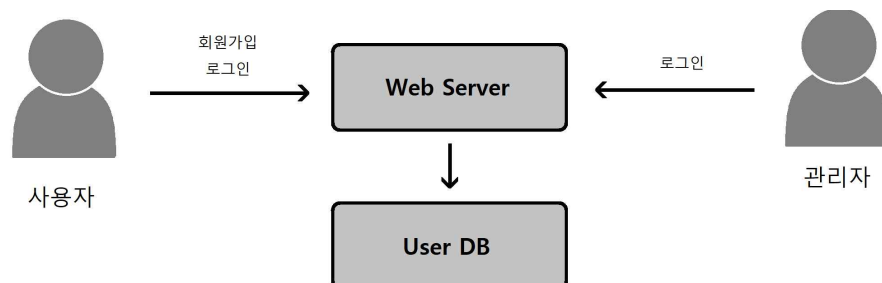


Diagram 1 Product Scope: User Management System

User Management System이다. User Management System은 사용자의 회원 정보를 관리하며 회원가입, 프로필 확인, 로그인/로그아웃과 같은 sub-system으로 나뉜다. 사용자는 이때 Web Server를 통해 User DB와 상호 작용한다.

### 2.4.3 Curriculum System

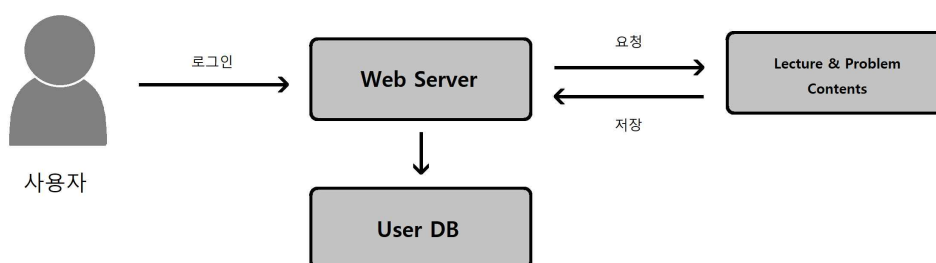
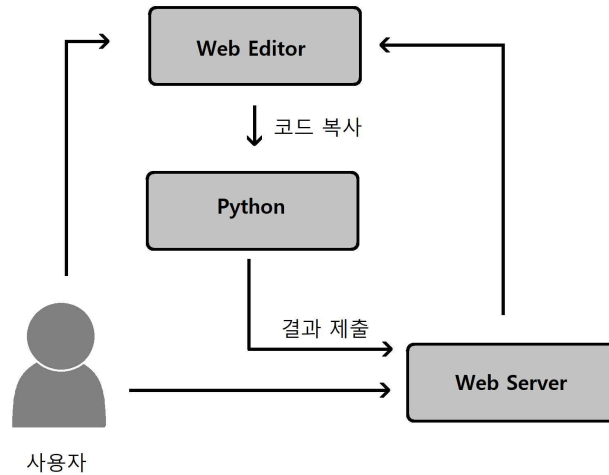


Diagram 2 Product Scope: Curriculum System

Curriculum System이다. Curriculum System은 사용자가 학습하는 강의 내용과 풀어야 하는 문제들로 구성된 Contents들을 포함하는 system으로, 실력 확인 문제 풀기, 이론 학습하기, 연습 문제 풀기, 채점하기와 같은 sub-system으로 나뉜다. 사용자와 Curriculum System은 이때 Web Server를 통해 User DB와 상호 작용하고, Web Server가 요청하고 User DB에 저장한다.

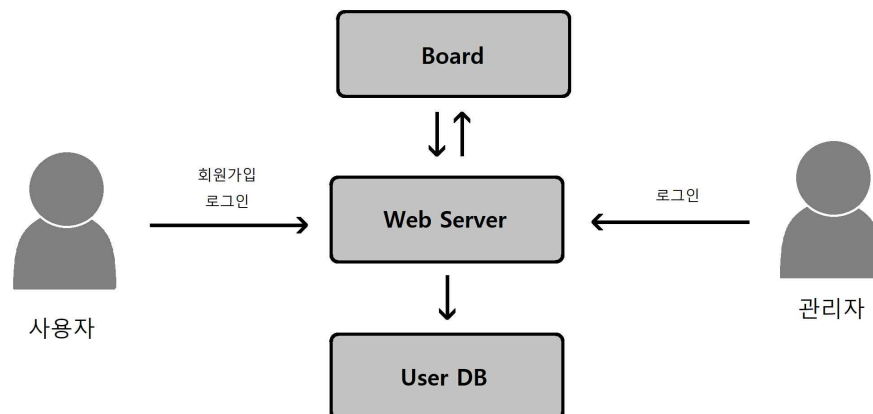
#### 2.4.4 Python & Web Editor System



**Diagram 3 Product Scope: Python & Web Editor System**

Python & Web Editor System이다. Python & Web Editor System은 사용자가 웹 브라우저 환경에서 코드 형식의 연습 문제를 작성하고 제출할 때 사용되는 시스템으로, 이때 Python은 사용자가 웹 브라우저로 Edupy를 실행하는 환경의 단말기에 설치되어 있어야 한다. 사용자가 Web Editor를 통해 작성한 코드는 사용자의 단말기에 설치된 Python으로 복사되어 실행되고, 실행된 결과는 Web Server로 제출된다. 이 시스템은 코드 작성, 코드 복사, 결과 제출과 같은 sub-system으로 나뉘며, Curriculum System과 상호작용한다.

#### 2.4.5 Board System



**Diagram 4 Product Scope: Board System**

Board System이다. Board System은 시스템의 게시판 역할을 하는 Board를 통해 사용자가 게시글을

조회, 작성, 수정하거나 관리자가 공지를 작성, 수정하는 시스템으로, 글 작성하기, 공지 작성하기, 답글 작성하기와 같은 Sub-system으로 나뉘며, DB handler와 상호 작용한다.

### 3. System Architecture

#### 3.1 Objective

System Architecture에서는 타겟 시스템에 대한 높은 수준의 개요와 시스템 기능의 전체적인 분포를 서술한다. 전체 시스템의 구조와 sub-system의 관계는 Block Diagram을 통해 도식화하였다.

#### 3.2 System Organization

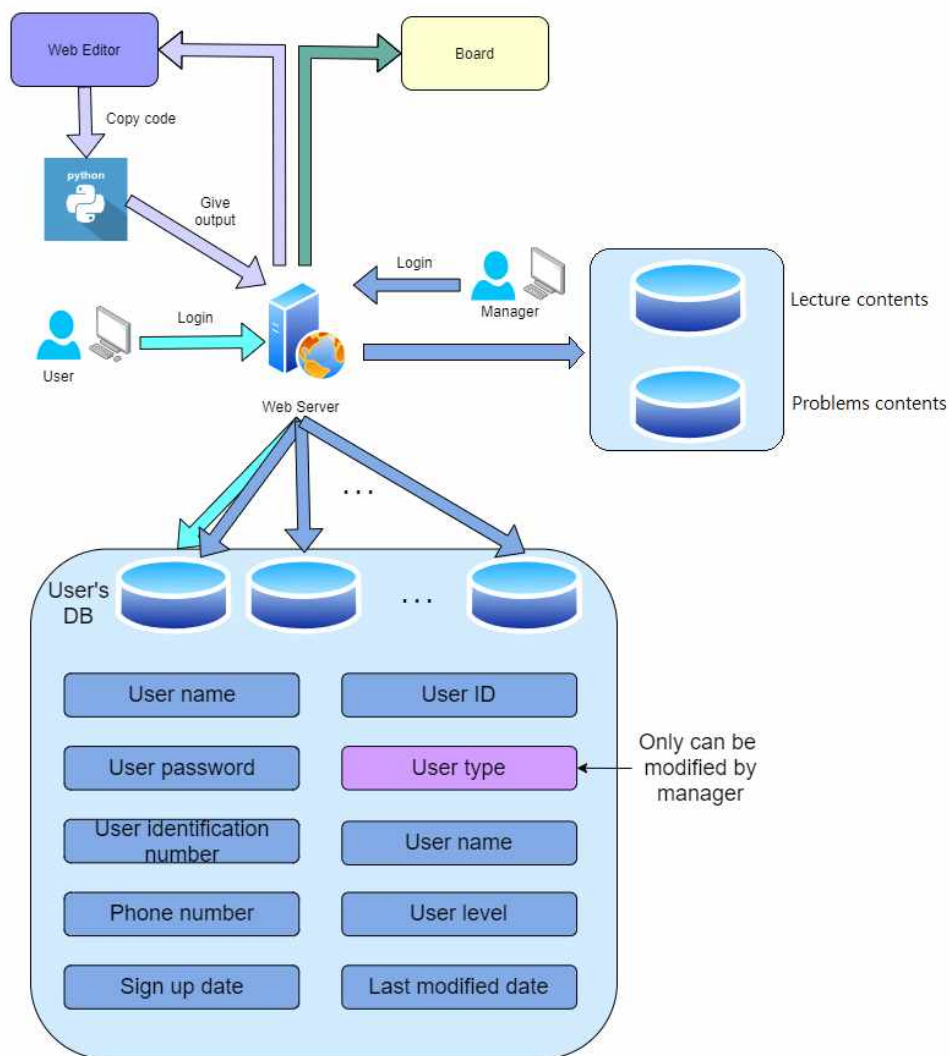
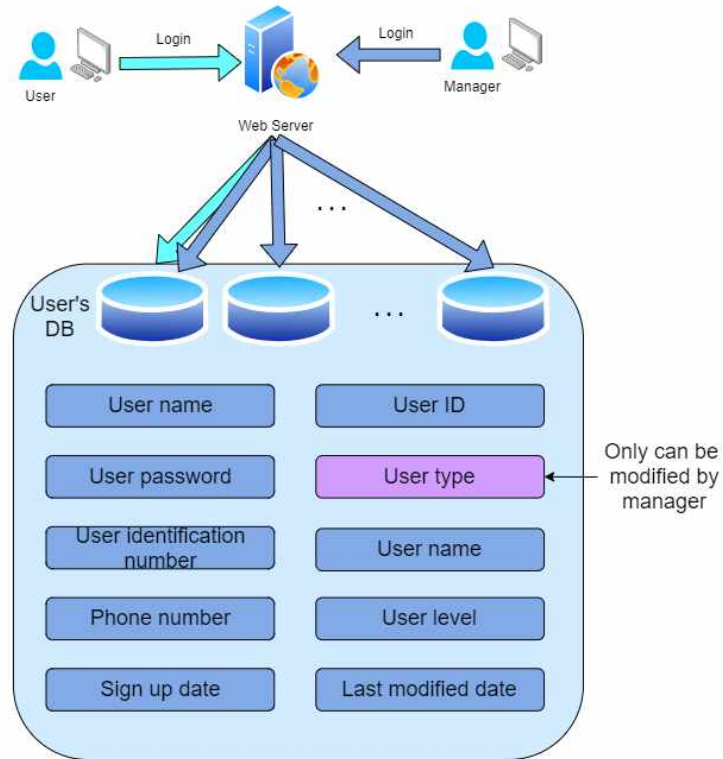


Diagram 5 System Architecture: Edupy 전체

Edupy의 전체 시스템 구조는 Web Application 구조로, 웹 브라우저로 접속하여 크게 Web Server, Web Editor System, Board System, User, Manager 시스템으로 구성된다.



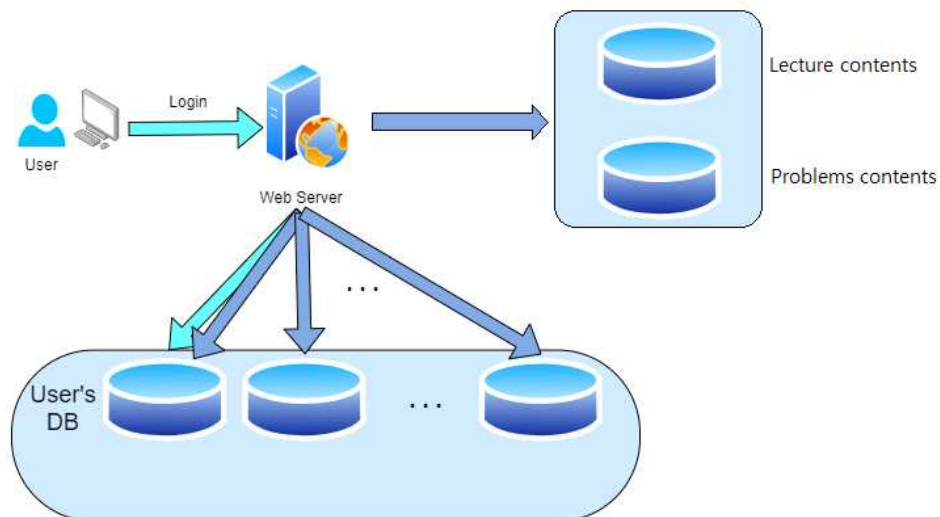
### 3.2.1 User Management System



**Diagram 6 System Architecture: User Management System**

User Management System은 사용자의 회원 가입, 로그인, 로그아웃 등 전반적인 사용자 관련 요구 사항을 담당하는 시스템이다.

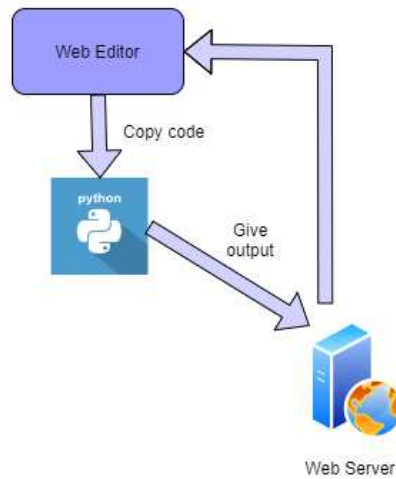
### 3.2.2 Curriculum System



**Diagram 7 System Architecture: Curriculum System**

Curriculum System은 사용자가 학습하는 강의와 문제(실력 확인 문제, 연습 문제 등) 등의 전반적인 학습과 관련된 시스템이다.

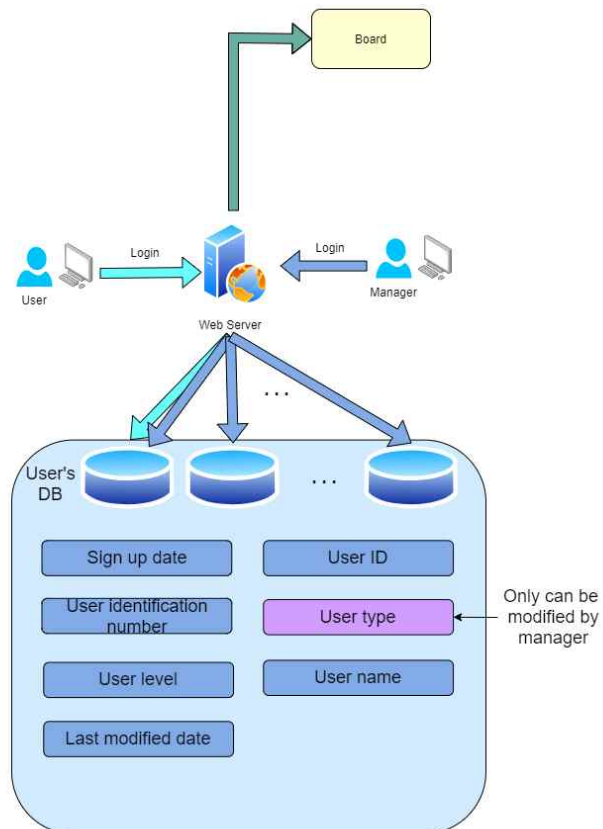
### 3.2.3 Python & Web Editor System



**Diagram 8 System Architecture: Python & Web Editor System**

Python & Web Editor System은 사용자가 웹 브라우저 환경에서 코드 형식의 연습 문제를 작성하고 제출할 때 사용되는 시스템이다.

### 3.2.4 Board System



**Diagram 9 System Architecture: Board System**

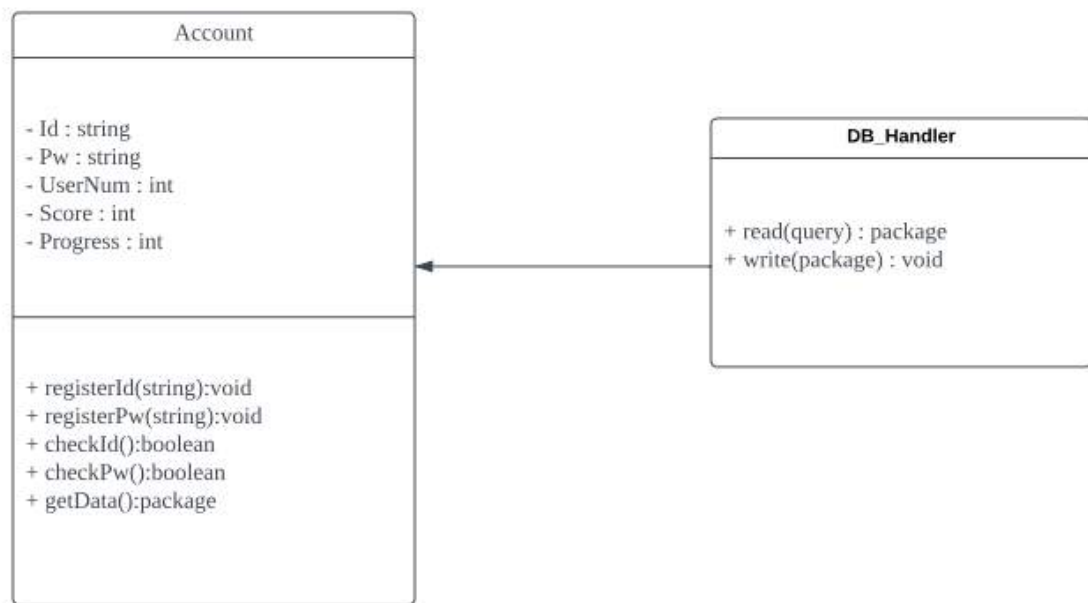
Board System은 Edupy 시스템의 게시판 역할을 하는 Board를 통해 사용자가 게시글을 조회, 작성, 수정하거나 관리자가 공지를 작성, 수정하는 시스템이다.

## 4. User Management System

### 4.1 Objective

User Management System은 사용자들의 회원 가입과 로그인에 관련된 서비스이다. 이 절에서는 Class Diagram, Sequence Diagram, State Diagram을 통해 Board System의 구조와 사용자 및 데이터베이스와의 상호 작용을 설계한다.

### 4.2 Class Diagram



**Diagram 10 User Management System: Class Diagram**

#### A. Account

##### A.1 Attributes

Id : 사용자의 Id

Pw : 사용자의 Pw

UserNum : 사용자의 고유 식별 번호

Score : 사용자의 점수

Progress : 사용자의 진도

##### A.2 Methods

registerId : 회원 가입을 통해 Id를 DB에 등록한다.

registerPw : 회원 가입을 통해 Pw를 DB에 등록한다

checkId : 로그인을 시도할 때 DB에 등록된 Id와 같은 지 확인한다.

CheckPw : 로그인을 시도할 때 DB에 등록된 Pw와 같은 지 확인한다.

getData : 해당 데이터 베이스로부터 유저의 정보를 가져온다.

## B. DB\_Handler

### B.1 Attributes

해당 사항 없음

### B.2 Methods

Package read(query): 해당 데이터베이스에서 원하는 데이터 읽어오기

void write(package): 해당 데이터베이스에 게시글 데이터 저장하기

## 4.3 Sequence Diagram

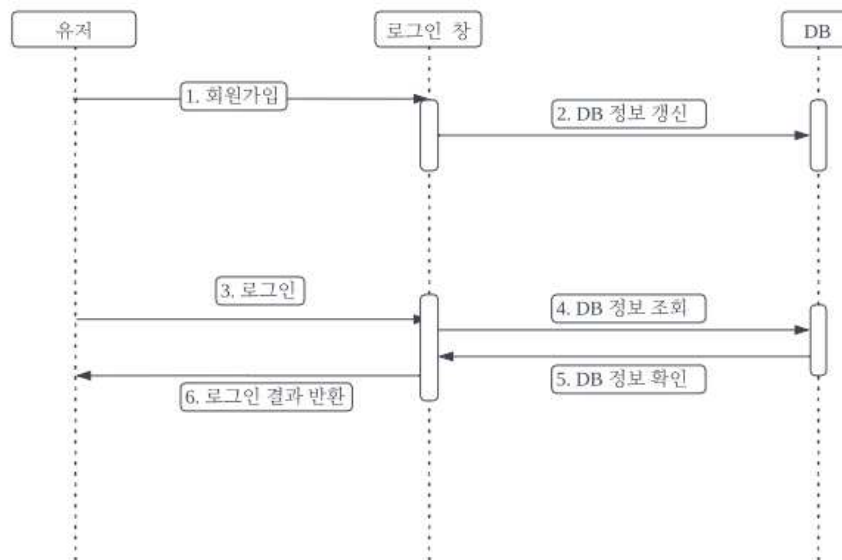


Diagram 11 User Management System: Sequence Diagram

## 4.4 State Diagram

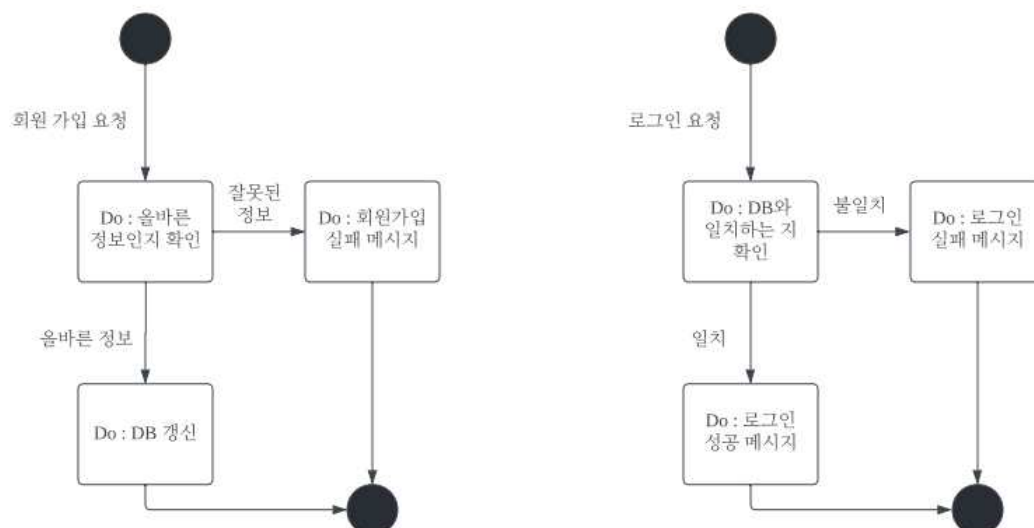


Diagram 12 User Management System: State Diagram

## 5. Curriculum System

### 5.1 Objective

Curriculum system은 사용자가 커리큘럼의 내용을 알고 있을 때 해당 내용을 넘어갈 수 있는 기회를 제공하는 실력 테스트와 각 커리큘럼에서 학습하게 될 내용과 다음 학습을 시작하기 위한 문제, 사용자의 실력을 테스트하는 문제를 관리한다. sub-system으로는 문제를 다루는 problem과 정답을 확인하는 judge가 존재한다. 이 내용을 class diagram과 sequence diagram, state diagram을 통해 상세 내용을 기술한다.

### 5.2 Class Diagram

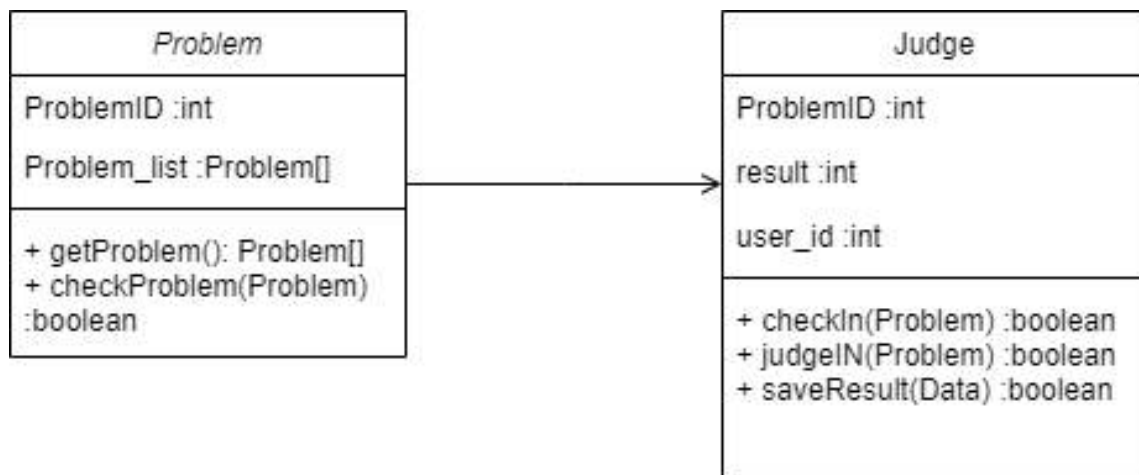


Diagram 13 Curriculum System: Class Diagram

#### A. Problem

##### A.1. Attributes

- +Problem\_list:Problem[]: 문제 배열
- +ProblemID:int: 문제 식별번호

##### A.2 Method

- +getProblem():Problem[]: 문제를 호출한다.
- +checkProblem(Problem):Boolean: 문제를 채점한다.

#### B. Judge

##### B.1. Attribute

- +ProblemID:int: 문제 식별번호
- +result:int: 채점결과
- +user\_id:int: 사용자 식별번호

##### B.2. Method

- +checkIn(Problem):Boolean: 문제 여부 판단
- +judgeIn(Problem):Boolean: 문제 채점
- +saveResult(Data):Boolean: 제출정보 submission DB에 저장

### 5.3 Sequence Diagram

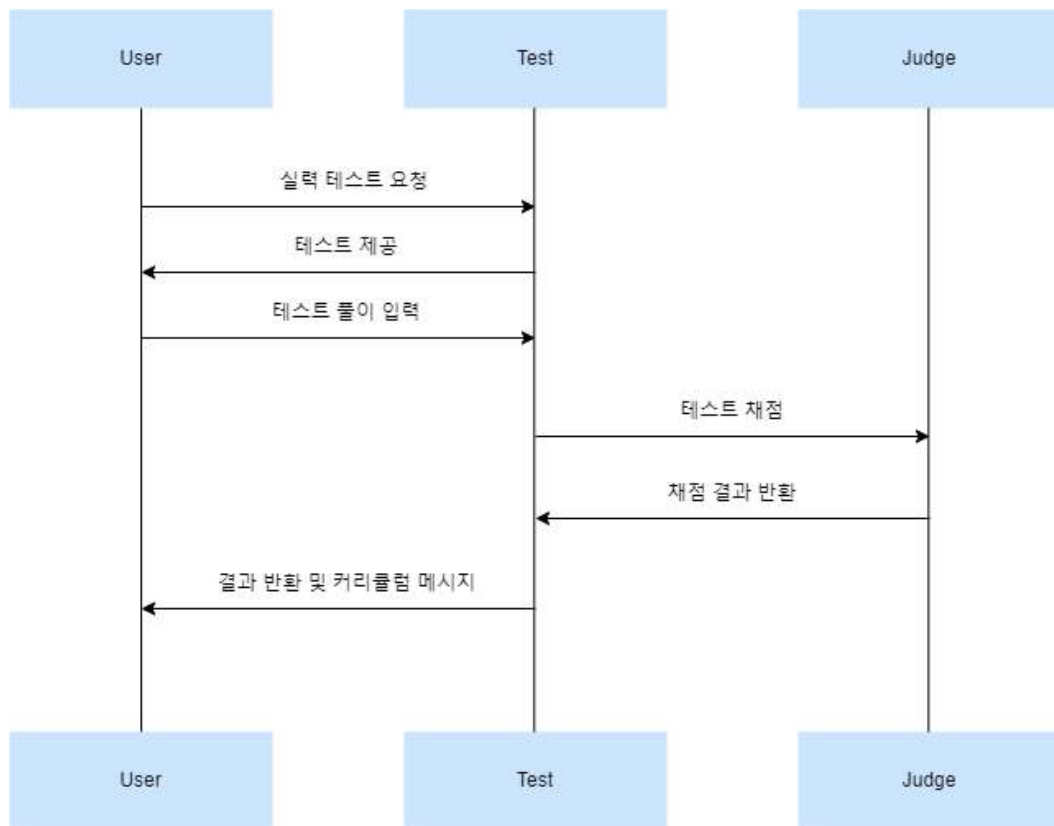


Diagram 14 Curriculum System: Sequence Diagram: 실력 테스트

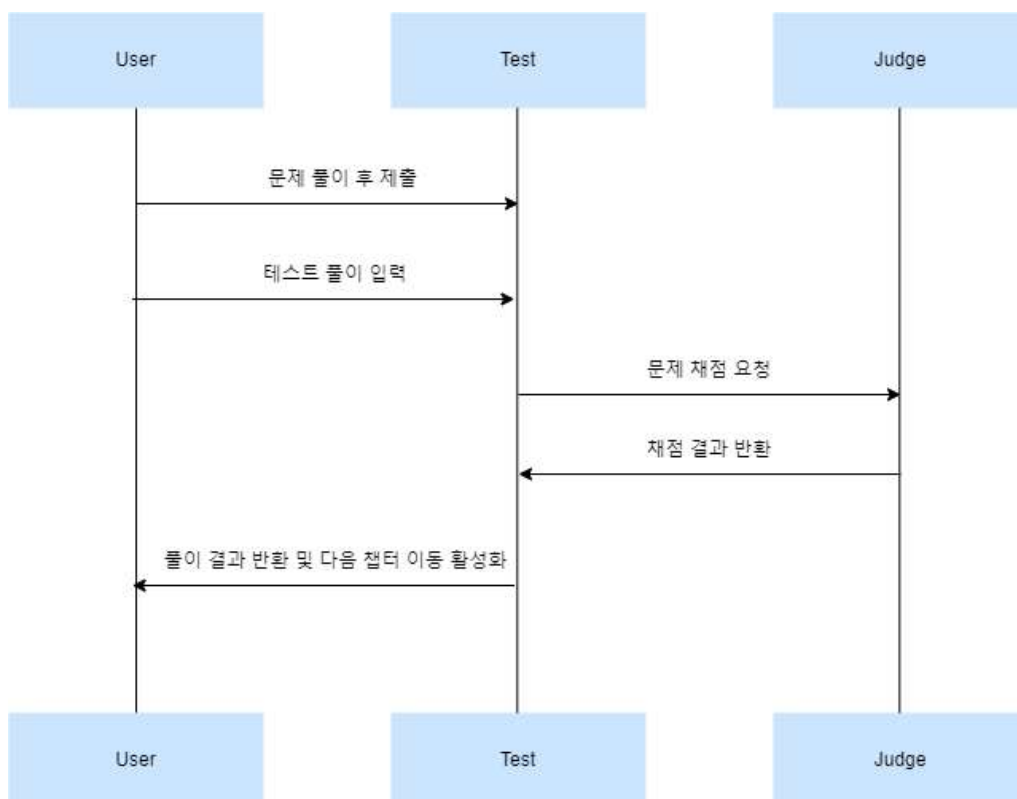


Diagram 15 Curriculum System: Sequence Diagram: 학습 테스트

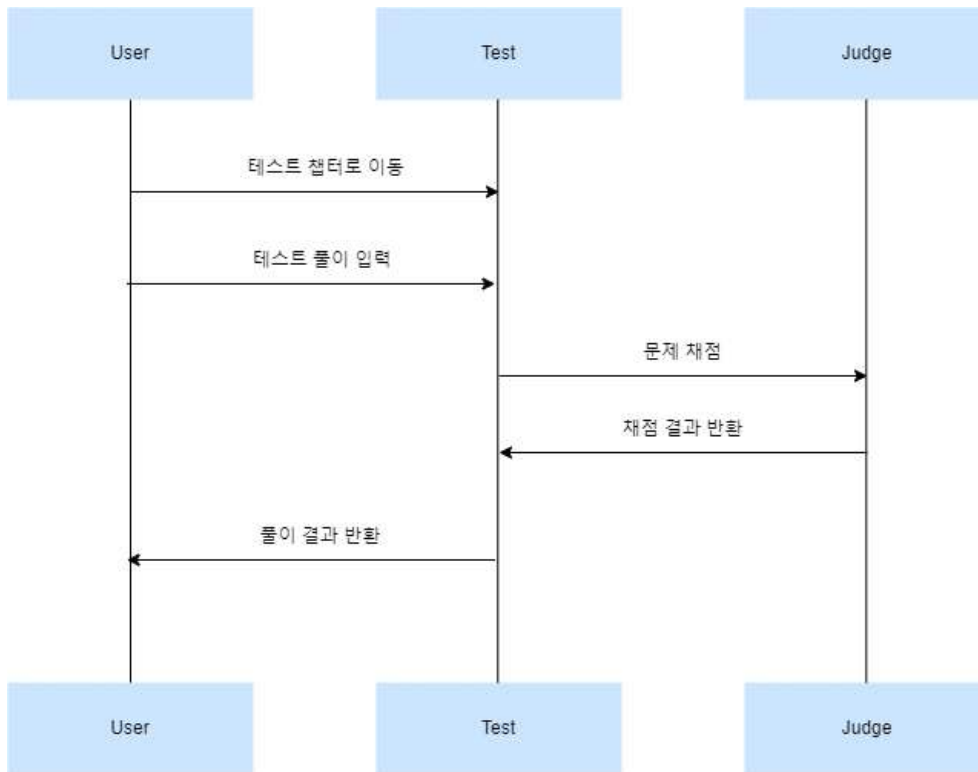


Diagram 16 Curriculum System: Sequence Diagram: 통합 테스트

#### 5.4 State Diagram

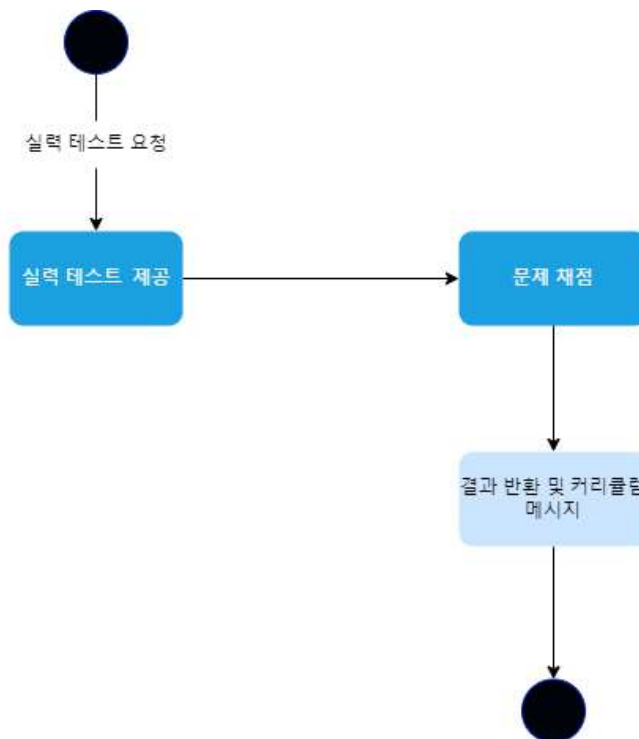


Diagram 17 Curriculum System: State Diagram: 실력테스트

## 6. Python & Web Editor System

### 6.1 Objective

Python & Web Editor System은 사용자가 실습을 통한 Python 교육을 진행하도록 도와준다. Skeleton code가 주어질 수도 있고 주어지지 않을 수도 있다. 사용자는 문제를 읽고 그에 맞는 코드를 web editor를 통해 작성한다. 이 절에서는 Class Diagram, Sequence Diagram, State Diagram을 통해 Python & Web Editor System의 구조와 사용자 및 데이터베이스와의 상호 작용을 설계한다.

### 6.2 Class Diagram

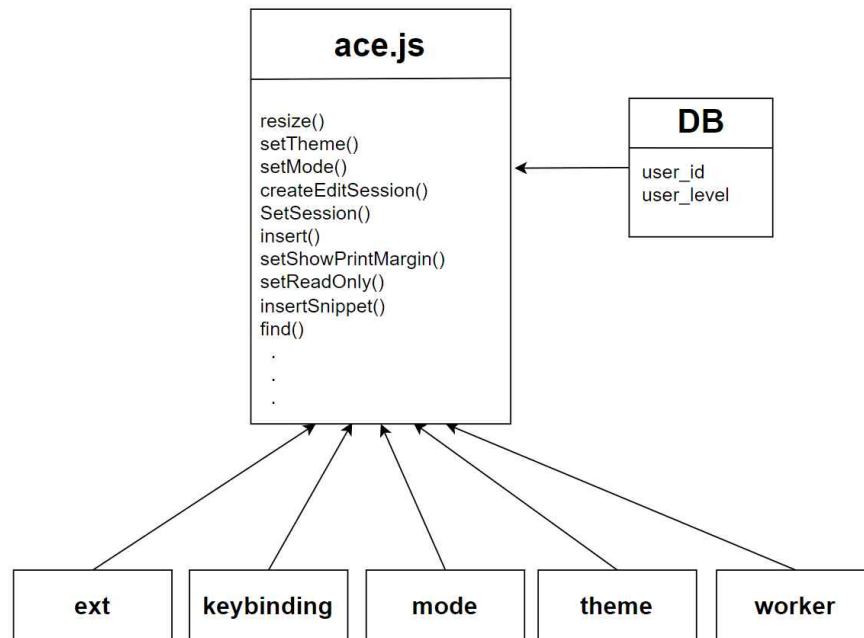


Diagram 18 Python & Web Editor System: Class Diagram

### 6.3 Sequence Diagram

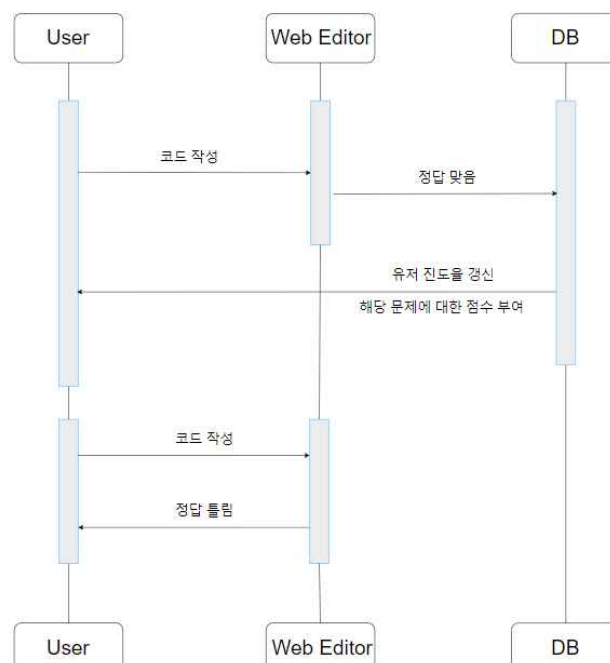


Diagram 19 Python & Web Editor System: Sequence Diagram



## 6.4 State Diagram

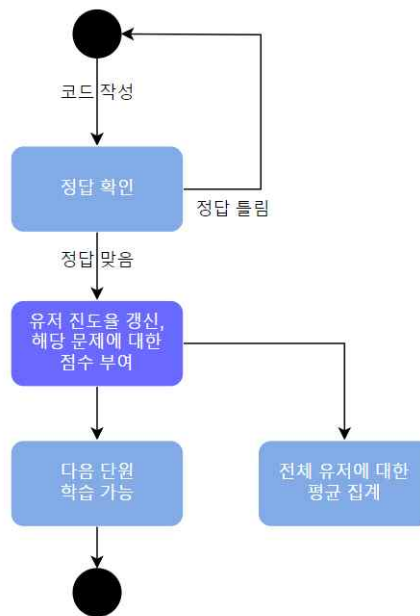


Diagram 20 Python & Web Editor System: State Diagram

## 7. Board System

### 7.1 Objective

Board System은 서비스 사용자들의 게시판 내 글 작성, 댓글 작성, 관리자 공지글 작성 등과 같은 게시판 서비스로 사용자와 관련한다. 이 절에서는 Class Diagram, Sequence Diagram, State Diagram을 통해 Board System의 구조와 사용자 및 데이터베이스와의 상호 작용을 설계한다.

### 7.2 Class Diagram

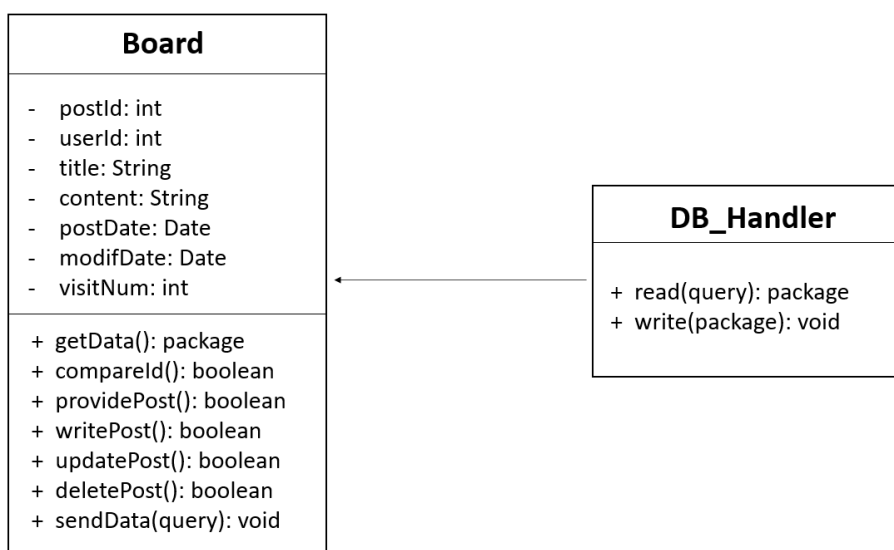


Diagram 21 Board System: Class Diagram

## A. Board

### A.1 Attributes

postId: 게시글의 고유 식별 번호

userId: 게시글 작성자의 고유 식별 번호

title: 게시글 제목

content: 게시글 내용

postDate: 게시글 작성 날짜

modifDate: 게시글 수정 날짜

visitNum: 게시글 방문자 수

### A.2 Methods

getData():Package: 해당 데이터베이스에서 게시글 데이터 가져오기.

compareId(int 사용자\_고유번호):boolean: 사용자의 고유 번호와 게시글 작성자의 고유 번호가 일치하는지 확인.

providePost():boolean 게시글 보여주기

addPost():boolean 게시글 추가하기

updatePost():boolean 게시글 수정하기

deletePost():boolean 게시글 삭제하기

sendData(query):void 해당 데이터베이스에 게시글 데이터 전송하기

## B. DB\_Handler

### B.1 Attributes

해당 사항 없음

### B.2 Methods

Package read(query): 해당 데이터베이스에서 원하는 데이터 읽어오기

void write(package): 해당 데이터베이스에 게시글 데이터 저장하기

### 7.3 Sequence Diagram

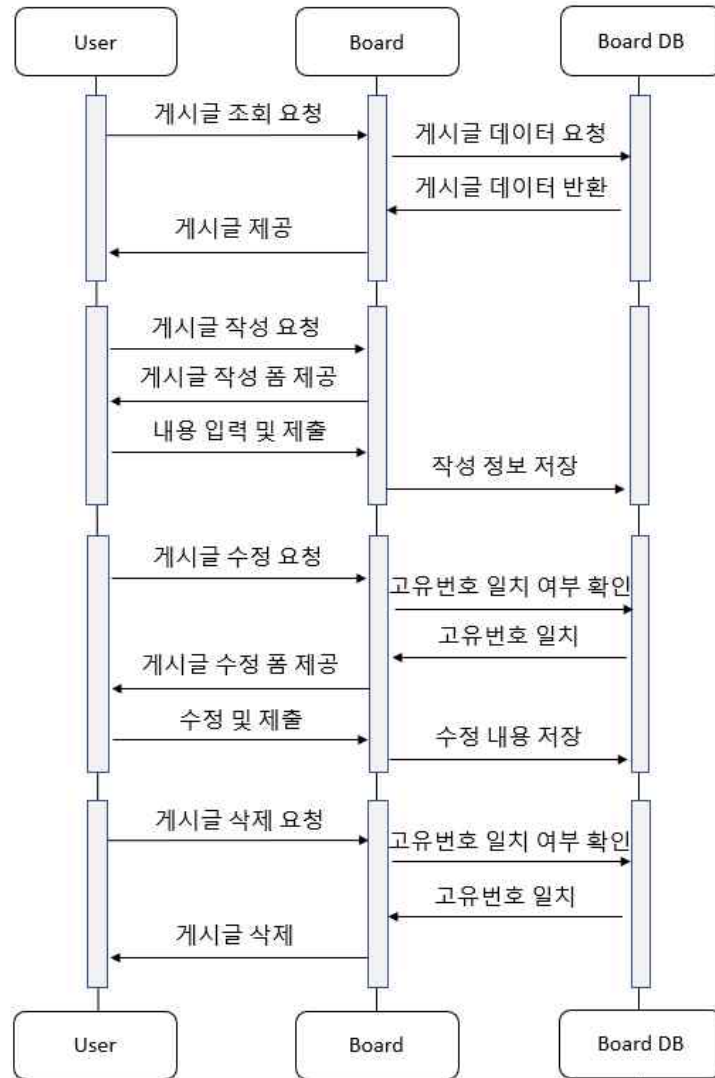


Diagram 22 Board System: Sequence Diagram

### 7.4 State Diagram

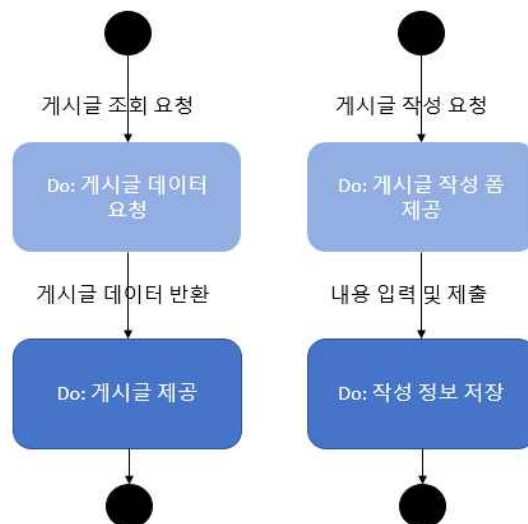


Diagram 23 Board System: State Diagram: 게시글 조회 및 작성



Diagram 24 Board System: State Diagram: 게시글 수정

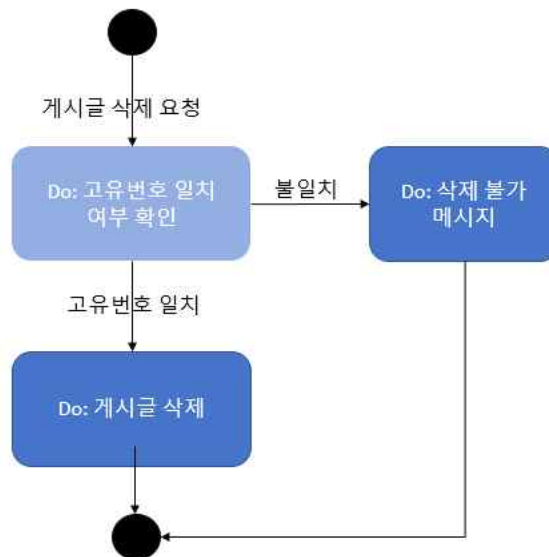


Diagram 25 Board System: State Diagram: 게시글 삭제

## 8. Database Design

### 8.1 Objective

요구사항 명세서를 기반으로 데이터베이스를 설계한다. 데이터베이스 내의 객체-데이터 관계를 표현하는 ER Diagram을 제시하고, Relational Schema를 도식화하여 서술한다. 또한 실제 테이블을 작

성할 수 있는 SQL의 DDL 쿼리를 작성한다.

## 8.2 ER Diagram

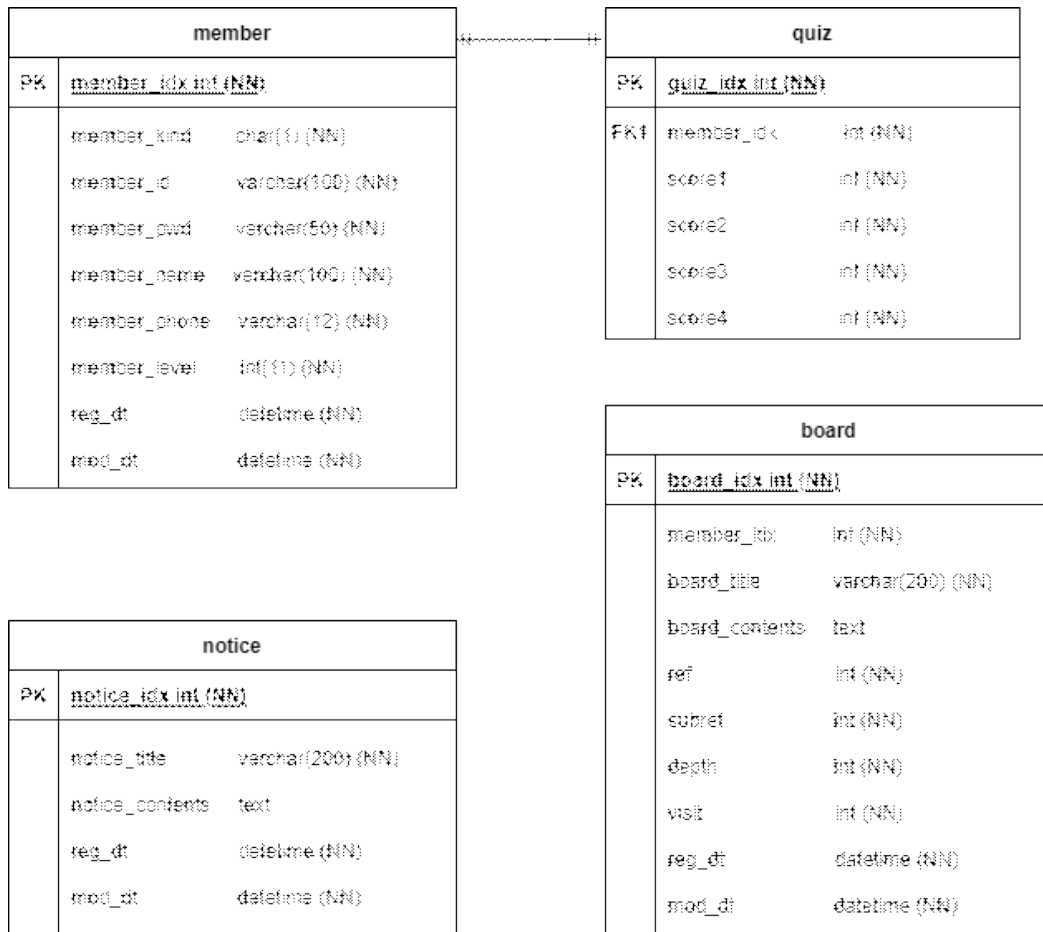


Diagram 26 Database System: ER Diagram

## 8.3 Rational Schema

### A. member

테이블 이름		member						
테이블 설명		사용자가 회원 가입시 입력한 정보들을 저장한다.						
PRIMARY KEY		member_idx						
FOREIGN KEY								
INDEX								
NO	PK	AI	FK	NULL	컬럼 이름	TYPE	설명	참조 테이블
1	Y	Y		N	member_idx	INT	유저 고유 번호	
2				N	member_kind	CHAR(1)	교수자 또는 학습자	
3				N	member_id	VARCHAR(100)	로그인 아이디	
4				N	member_pwd	VARCHAR(50)	로그인 비밀번호	
5				N	member_name	VARCHAR(100)	사용자 이름	
6				N	member_phone	VARCHAR(12)	사용자 전화번호	
7				N	member_level	INT(11)	학습자 본인의 진도	
8				N	reg_dt	DATETIME	최초 가입일	
9				N	mod_dt	DATETIME	회원정보 마지막 수정일	

CREATE TABLE member

```
(
  member_idx int NOT NULL AUTO_INCREMENT,
  member_kind char(1) DEFAULT 'C' NOT NULL,
  member_id varchar(100) DEFAULT " NOT NULL,
  member_pwd varchar(50) DEFAULT " NOT NULL,
  member_name varchar(100) DEFAULT " NOT NULL,
  member_phone varchar(12) DEFAULT " NOT NULL,
  member_level int(11) DEFAULT 1 NOT NULL,
  reg_dt datetime DEFAULT '2021-12-19' NOT NULL,
  mod_dt datetime DEFAULT '2021-12-19' NOT NULL,
  PRIMARY KEY (member_idx)
);
```

B. board

테이블 이름				board				
테이블 설명				자유게시판의 게시글 정보들을 저장한다.				
PRIMARY KEY				board_idx				
FOREIGN KEY								
INDEX								
NO	PK	AI	FK	NULL	컬럼 이름	TYPE	설명	참조 테이블
1	Y	Y		N	board_idx	INT	게시글 고유 번호	
2				N	member_idx	INT	게시글 작성자 번호	
3				N	board_title	VARCHAR(200)	게시글 제목	
4					board_contents	TEXT	게시글 내용	
5				N	ref	INT	게시글 그룹 번호	
6				N	subref	INT	그룹내의 순서	
7				N	depth	INT	게시글의 깊이	
8				N	visit	INT	게시글 조회 수	
9				N	reg_dt	DATETIME	게시글 최초 등록일	
10				N	mod_dt	DATETIME	게시글 마지막 수정일	

CREATE TABLE board

```
(
  board_idx int NOT NULL,
  member_idx int DEFAULT 0 NOT NULL,
  board_type varchar(10) DEFAULT 'free' NOT NULL,
  board_title varchar(200) DEFAULT " NOT NULL,
```

```

board_contents text,
file varchar(100) DEFAULT " NOT NULL,
ref int DEFAULT 0 NOT NULL,
subref int DEFAULT 0 NOT NULL,
depth int DEFAULT 0 NOT NULL,
visit int DEFAULT 1 NOT NULL,
reg_dt datetime DEFAULT '1900-01-01' NOT NULL,
mod_dt datetime DEFAULT '1900-01-01' NOT NULL,
PRIMARY KEY (board_idx)

```

);

#### C. notice

테이블 이름	notice							
테이블 설명	공지사항 게시판의 공지사항 정보들을 저장한다.							
PRIMARY KEY	notice_idx							
FOREIGN KEY								
INDEX								
NO	PK	AI	FK	NULL	컬럼 이름	TYPE	설명	참조 테이블
1	Y	Y		N	notice_idx	INT	공지사항 고유 번호	
2				N	notice_title	VARCHAR(200)	공지사항 제목	
3					notice_contents	TEXT	공지사항 내용	
4				N	reg_dt	DATETIME	최초 등록일	
5				N	mod_dt	DATETIME	마지막 수정일	

#### CREATE TABLE notice

```

(
notice_idx int NOT NULL AUTO_INCREMENT,
notice_title varchar(200) DEFAULT " NOT NULL,
notice_contents text,
reg_dt datetime DEFAULT '1900-01-01' NOT NULL,
mod_dt datetime DEFAULT '1900-01-01' NOT NULL,
PRIMARY KEY (notice_idx)
);

```

#### D. quiz

테이블 이름	quiz
테이블 설명	학습자의 퀴즈 점수를 저장한다.
PRIMARY KEY	quiz_idx
FOREIGN KEY	member_idx

INDEX								
NO	PK	AI	FK	NULL	컬럼 이름	TYPE	설명	참조 테이블
1	Y	Y		N	quiz_idx	INT	데이터 고유 번호	
2			Y	N	member_idx	INT	학습자 고유 번호	member
3				N	score1	INT	퀴즈1 점수	
4				N	score2	INT	퀴즈2 점수	
5				N	score3	INT	퀴즈3 점수	
6				N	score4	INT	퀴즈4 점수	

**CREATE TABLE quiz**

```
(
  quiz_idx int NOT NULL AUTO_INCREMENT,
  member_idx int NOT NULL,
  score1 int DEFAULT 0 NOT NULL,
  score2 int DEFAULT 0 NOT NULL,
  score3 int DEFAULT 0 NOT NULL,
  score4 int DEFAULT 0 NOT NULL,
  PRIMARY KEY (quiz_idx)
);
```

**ALTER TABLE quiz**

```
ADD FOREIGN KEY (member_idx)
REFERENCES member (member_idx)
ON UPDATE RESTRICT
ON DELETE RESTRICT
```

## 9. Testing Plan

### 9.1 Objective

Testing Plan에서는 Edupy가 요구사항 명세서에서 기술한 요구대로 만들어졌는지, 결함은 없는지 등을 테스트하여 확인한다. 관련 테스트를 수행하기 위한 Testing Policy를 기술하고 이를 설계하며, 각 Testing Plan에서는 Testing Policy와 Test Case를 설명한다.

### 9.2 Testing Policy

본 서비스의 Testing은 Development Testing, Release Testing, User Testing 세 단계로 구성된다. 그 중 Development Testing은 Component Testing, System Testing, Acceptance Testing을 순차적으로 진행한다. 다음은 각각의 testing의 정의와 과제들을 서술한다.

#### 9.2.1 Development Testing



Development Testing은 소프트웨어의 개발 도중에 발생할 수 있는 위험과 시간 및 비용을 줄이기 위해 광범위한 결함 예방 및 탐지 전략의 동기화된 적용을 수반하는 단계이다. 이 단계에서는 소프트웨어 개발에 대한 조직의 기대치에 따라 Static Code Analysis, Data Flow Analysis, Metrics Analysis, Peer Code Reviews, Unit Testing, Code Coverage Analysis, Traceability 및 기타 소프트웨어 검증 관행이 포함될 수 있다.

### 9.2.2 Release Testing

Release Testing은 소프트웨어 출시 준비자(개발자)가 사용자에게 준비되었다는 확신을 팀에 제공하는 코딩 관행 및 테스트 전략으로, Release Testing은 소프트웨어 출시에서 오류와 버그를 찾아 제거하여 사용자에게 출시할 수 있도록 하는 것을 목표로 한다.

### 9.2.3 User Testing

개발된 소프트웨어가 사용자의 실제 사용 환경에서 의도와 요구대로 동작하는지 확인하기 위한 테스트 단계이다. 실제 사용자들을 통해 개발된 소프트웨어가 직관적인지, 사용하기 쉬운지 등을 평가하고, 만들어진 소프트웨어를 바로 출시할 것인지 결정한다.

## 9.3 Test Case

### 9.3.1 User Management System

#### 9.3.1.1 회원 가입하기

구성요소	설명
테스트 이름 (Test Title)	회원 가입하기 테스트
사전 조건 (Precondition)	- 이메일은 '아이디 문자열 @ 도메인 문자열' 형식이며, 중복되어서는 안된다. - 비밀번호는 4~16자의 문자열 형식이다.
수행 절차 (Test Step)	1. 이메일 형식 체크 2. 이메일 중복 체크 3. 비밀번호 형식 체크
기대되는 결과 (Expected Result)	- 회원 가입 후 User DB에 회원 정보 저장 - 로그인 화면으로 전환

**Table 1 Test case: 회원 가입하기**

#### 9.3.1.2 로그인하기

구성요소	설명
테스트 이름 (Test Title)	로그인하기 테스트
사전 조건 (Precondition)	- 이메일은 '아이디 문자열 @ 도메인 문자열' 형식이며, 중복되어서는 안된다. - 비밀번호는 4~16자의 문자열 형식이다. - 여러번 로그인 시도할 경우 최초 한 번의 시도만 서버에 요청한다.

수행 절차 (Test Step)	1. 이메일 형식 체크 2. 비밀번호 형식 체크 3. 이메일 존재 여부 체크 4. 이메일과 비밀번호 일치 여부 체크
기대되는 결과 (Expected Result)	- 사용자 로그인 상태로 기록, 메인 화면으로 전환

**Table 2 Test case: 로그인하기**

#### 9.3.1.3 비밀번호 변경하기

구성요소	설명
테스트 이름 (Test Title)	비밀번호 변경하기 테스트
사전 조건 (Precondition)	사용자는 로그인한 상태여야 한다.
수행 절차 (Test Step)	1. 사용자가 마이페이지 버튼을 누른다. 2. 사용자가 비밀번호 변경 버튼을 누른다. 3. 새로운 비밀번호를 입력할 화면이 나온다. 4. 새로운 비밀번호를 입력한다. 5. 비밀번호 확인 칸에 다시 입력한다. 6. 비밀번호 변경 확인 버튼을 누른다.
기대되는 결과 (Expected Result)	- 새로운 비밀번호를 User DB에 저장한다.

**Table 3 Test case: 비밀번호 변경하기**

#### 9.3.1.4 커리큘럼 진행 확인하기

구성요소	설명
테스트 이름 (Test Title)	커리큘럼 진행 확인하기 테스트
사전 조건 (Precondition)	사용자는 로그인한 상태여야 한다.
수행 절차 (Test Step)	1. 사용자가 마이페이지 버튼을 누른다. 2. 사용자가 커리큘럼 진행도를 확인한다.
기대되는 결과 (Expected Result)	- 사용자가 자신이 진행한 커리큘럼의 진행도를 확인한다.

**Table 4 Test case: 커리큘럼 진행 확인하기**

### 9.3.2 Curriculum System

### 9.3.2.1 이론 학습하기

구성요소	설명
테스트 이름 (Test Title)	이론 학습하기 테스트
사전 조건 (Precondition)	사용자는 로그인한 상태여야 한다.
수행 절차 (Test Step)	<ol style="list-style-type: none"> <li>1. 사용자가 원하는 커리큘럼 버튼을 누른다.</li> <li>2. 사용자가 실력 테스트 문제를 푼다.</li> <li>3. 사용자가 제출하기 버튼을 클릭한다.</li> <li>4. 사용자가 결과를 확인한다.</li> </ol>
기대되는 결과 (Expected Result)	- 실력 테스트를 통과하면 원하는 커리큘럼을 학습한다.

**Table 5 Test case: 이론 학습하기**

### 9.3.2.2 실력 테스트 문제 풀기

구성요소	설명
테스트 이름 (Test Title)	실력 테스트 문제 풀기 테스트
사전 조건 (Precondition)	사용자는 로그인한 상태여야 한다.
수행 절차 (Test Step)	<ol style="list-style-type: none"> <li>1. 사용자가 원하는 커리큘럼 버튼을 누른다.</li> <li>2. 사용자가 누른 커리큘럼에 해당하는 실력 테스트를 제공한다.</li> <li>3. 사용자가 실력 테스트를 푼다.</li> <li>4. 사용자가 제출하기 버튼을 클릭한다.</li> <li>5. 사용자가 제출한 테스트를 채점한다.</li> <li>6. 결과를 반환한다.</li> <li>7. 사용자가 결과를 확인한다.</li> </ol>
기대되는 결과 (Expected Result)	- 실력 테스트 결과를 반환하여 커리큘럼 메시지를 출력한다.

**Table 6 Test case: 실력 테스트 문제 풀기**

### 9.3.2.3 학습 테스트 문제 풀기

구성요소	설명
테스트 이름 (Test Title)	학습 테스트 문제 풀기 테스트
사전 조건 (Precondition)	<ul style="list-style-type: none"> <li>- 사용자는 로그인한 상태여야 한다.</li> <li>- 사용자는 이론을 학습한 상태여야 한다.</li> </ul>

수행 절차 (Test Step)	<ol style="list-style-type: none"> <li>1. 사용자가 학습한 커리큘럼에서 문제를 제공한다.</li> <li>2. 사용자가 테스트 풀이를 입력한다.</li> <li>3. 문제 채점을 요청한다.</li> <li>4. 채점 결과를 반환한다.</li> <li>5. 풀이 결과를 반환한다.</li> </ol>
기대되는 결과 (Expected Result)	- 풀이 결과를 반환하고 통과하면 다음 챕터 이동을 활성화한다.

**Table 7 Test case: 학습 테스트 문제 만들기**

#### 9.3.2.4 통합 테스트 문제 풀기

구성요소	설명
테스트 이름 (Test Title)	통합 테스트 문제 풀기 테스트
사전 조건 (Precondition)	<ul style="list-style-type: none"> <li>- 사용자는 로그인한 상태여야 한다.</li> <li>- 사용자는 이론의 대단위 챕터를 학습한 상태여야 한다.</li> </ul>
수행 절차 (Test Step)	<ol style="list-style-type: none"> <li>1. 사용자가 테스트 챕터로 이동한다.</li> <li>2. 사용자가 테스트 풀이를 입력한다.</li> <li>3. 문제 채점을 요청한다.</li> <li>4. 채점 결과를 반환한다.</li> <li>5. 풀이 결과를 반환한다.</li> </ol>
기대되는 결과 (Expected Result)	- 풀이 결과를 반환하여 사용자가 확인할 수 있게 한다.

**Table 8 Test case: 통합 테스트 문제 풀기**

### 9.3.3 Python & Web Editor System

#### 9.3.3.1 코드 작성 및 제출하기

구성요소	설명
테스트 이름 (Test Title)	코드 작성 및 제출하기 테스트
사전 조건 (Precondition)	사용자는 로그인한 상태여야 한다.
수행 절차 (Test Step)	<ol style="list-style-type: none"> <li>1. 사용자가 코드를 작성한다.</li> <li>2. 사용자가 작성한 코드를 확인한다.</li> </ol>
기대되는 결과 (Expected Result)	<ul style="list-style-type: none"> <li>- 정답이 맞으면 사용자의 진도율(커리큘럼 진행도)을 갱신하여 해당 문제에 대한 점수를 부여한다.</li> <li>- 정답이 틀리면 갱신하지 않는다.</li> </ul>

**Table 9 Test case: 코드 작성 및 제출하기**

### 9.3.4 Board System

#### 9.3.4.1 게시물 조회하기

구성요소	설명
테스트 이름 (Test Title)	게시글 조회하기 테스트
사전 조건 (Precondition)	사용자는 로그인한 상태여야 한다.
수행 절차 (Test Step)	1. 사용자가 게시물 조회를 요청한다. 2. 게시물 데이터를 요청하여 반환한다.
기대되는 결과 (Expected Result)	- 게시글을 제공한다.

**Table 10 Test case: 게시물 조회하기**

#### 9.3.4.2 게시물 작성하기

구성요소	설명
테스트 이름 (Test Title)	게시글 작성하기 테스트
사전 조건 (Precondition)	사용자는 로그인한 상태여야 한다.
수행 절차 (Test Step)	1. 사용자가 게시물 작성을 요청한다. 2. 게시물 작성 폼을 제공한다. 3. 사용자가 게시글의 내용을 입력하고 제출한다. 4. 작성 정보를 저장한다.
기대되는 결과 (Expected Result)	사용자가 작성한 게시글을 보여준다.

**Table 11 Test case: 게시물 작성하기**

#### 9.3.4.3 게시물 수정하기

구성요소	설명
테스트 이름 (Test Title)	게시글 수정하기 테스트
사전 조건 (Precondition)	- 사용자는 로그인한 상태여야 한다. - 사용자는 최소 한 개 이상의 게시글을 작성한 상태여야 한다.
수행 절차 (Test Step)	1. 사용자가 게시글의 수정을 요청한다. 2. 사용자의 고유번호가 일치하는지 확인한다.

	3. 고유번호가 일치하면 게시글 수정 폼을 제공한다. 4. 수정한 게시글을 제출한다. 5. 수정 내용을 저장한다.
기대되는 결과 (Expected Result)	사용자가 수정한 게시글을 보여준다.

**Table 12 Test case: 게시글 수정하기**

#### 9.3.4.4 게시글 삭제하기

구성요소	설명
테스트 이름 (Test Title)	게시글 삭제하기 테스트
사전 조건 (Precondition)	- 사용자는 로그인한 상태여야 한다. - 사용자는 최소 한 개 이상의 게시글을 작성한 상태여야 한다.
수행 절차 (Test Step)	1. 사용자가 게시글 삭제를 요청한다. 2. 사용자의 고유번호가 일치하는지 확인한다. 3. 고유번호가 일치하면 게시글을 삭제한다.
기대되는 결과 (Expected Result)	사용자의 게시글이 삭제되었다는 메시지를 출력한다.

**Table 13 Test case: 게시글 삭제하기**

## 10. Development Environment

### 10.1 Objective

Development Environment에서는 실제 시스템 개발을 위해 필요한 개발 환경과 코딩 규칙에 대해 기술한다. 개발 과정에서의 버전 관리 도구를 설명하고 프로그래밍 과정에서 기반이 되는 규칙들에 대해 서술한다.

### 10.2 Bootstrap



**Figure 11 Logo of Bootstrap**

Bootstrap은 웹사이트를 쉽게 만들 수 있게 도와주는 HTML, CSS, JS 프레임워크이다. 하나의 CSS로 휴대폰, 태블릿, 데스크탑까지 다양한 기기에서 작동한다. 다양한 기능을 제공하여 사용자가 쉽게 웹

사이트를 제작, 유지, 보수할 수 있도록 도와준다.

### 10.3 MariaDB



Figure 12 Logo of MariaDB

MariaDB는 오픈 소스의 관계형 데이터베이스 관리 시스템(RDBMS)이다. MySQL과 동일한 소스 코드를 기반으로 하며, GPL v2 라이선스를 따른다.

### 10.4 Github



Figure 13 Logo of GitHub

GitHub는 분산 버전 관리 툴인 깃 저장소 호스팅을 지원하는 웹 서비스이다. 전세계에서 오픈소스 프로젝트 관리를 위해 가장 많이 사용되는 서비스 중 하나로, 사용자에게 무료로 계정과 저장소를 제공하고 부분적으로 비공개 프로젝트도 무료로 진행할 수 있기 때문에, 한국을 포함한 전세계 IT 업계에서는 GitHub 계정이 취업 시 일종의 포트폴리오로 사용되기도 한다.

## 11. Development Plan

### 11.1 Objective

Development Plan에서는 프로젝트 개발 일정을 기술한다. Gantt Chart로 전체적인 개발 순서와 시기를 알아보기 쉽게 표현하고 현재까지의 개발 현황을 설명한다.

### 11.2 Gantt Chart

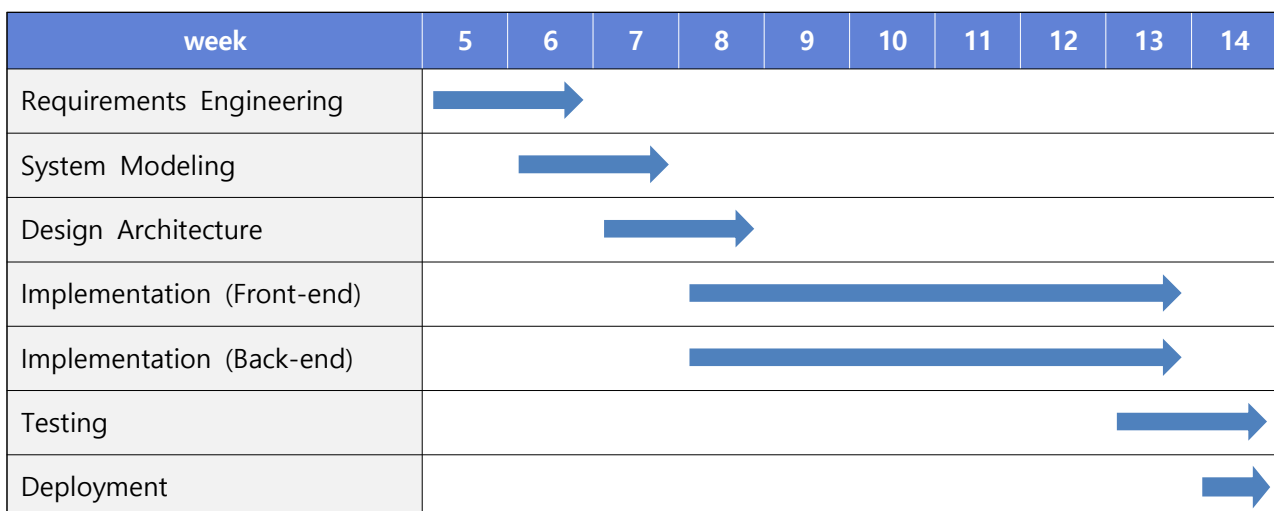


Diagram 27 Gantt Chart

현재까지의 개발 상황은 위와 같다. 파이썬의 기초부터 커리큘럼의 최종 목표인 실시간 채팅 기능 구현과 게시판 커뮤니티 기능의 구현을 위한 서비스와 필요 기능들을 정의하였다. 또한 사용자의 요구사항을 구체적으로 명세화하였고, 그 요구사항을 Functional, Non-functional로 나누어 분석하고, 동시에 System Modeling을 다양한 Diagram으로 표현하였다.

## 12. Index

### 12.1 Objective

Index에서는 본 문서에 사용된 용어와 다이어그램, 기능에 대한 인덱스를 나타낸다.

### 12.2 Figure Index

Figure 1 Logo of UML.....	10
Figure 2 Example of Class Diagram.....	10
Figure 3 Examples of State Diagram.....	11
Figure 4 Example of Sequence Diagram.....	11
Figure 5 Example of ER Diagram.....	12
Figure 6 Logo of JavaScript.....	13
Figure 7 Logo of HTML.....	13
Figure 8 Logo of CSS.....	13
Figure 9 Logo of MariaDB.....	13
Figure 10 Logo of Draw.io.....	14
Figure 11 Logo of Bootstrap.....	38
Figure 12 Logo of MariaDB.....	39
Figure 13 Logo of GitHub.....	39

### 12.3 Diagram Index

Diagram 1 Product Scope: User Management System.....	14
Diagram 2 Product Scope: Curriculum System.....	14
Diagram 3 Product Scope: Python & Web Editor System.....	15
Diagram 4 Product Scope: Board System.....	15
Diagram 5 System Architecture: Edupy 전체.....	16
Diagram 6 System Architecture: User Management System.....	17
Diagram 7 System Architecture: Curriculum System.....	17
Diagram 8 System Architecture: Python & Web Editor System.....	18
Diagram 9 System Architecture: Board System.....	18
Diagram 10 Curriculum System: Class Diagram.....	19



Diagram 11 Curriculum System: Sequence Diagram	20
Diagram 12 Curriculum System: State Diagram	20
Diagram 13 Curriculum System: Class Diagram	21
Diagram 14 Curriculum System: Sequence Diagram: 실력 테스트	22
Diagram 15 Curriculum System: Sequence Diagram: 학습 테스트	22
Diagram 16 Curriculum System: Sequence Diagram: 통합 테스트	23
Diagram 17 Curriculum System: State Diagram: 실력테스트	23
Diagram 18 Python & Web Editor System: Class Diagram	24
Diagram 19 Python & Web Editor System: Sequence Diagram	24
Diagram 20 Python & Web Editor System: State Diagram	24
Diagram 21 Board System: Class Diagram	25
Diagram 22 Board System: Sequence Diagram	27
Diagram 23 Board System: State Diagram: 게시글 조회 및 작성	27
Diagram 24 Board System: State Diagram: 게시글 수정	28
Diagram 25 Board System: State Diagram: 게시글 삭제	28
Diagram 26 Database System: ER Diagram	29
Diagram 27 Gantt Chart	39

#### 12.4 Table Index

Table 1 Test case: 회원 가입하기	33
Table 2 Test case: 로그인하기	33
Table 3 Test case: 비밀번호 변경하기	34
Table 4 Test case: 커리큘럼 진행 확인하기	34
Table 5 Test case: 이론 학습하기	35
Table 6 Test case: 실력 테스트 문제 풀기	35
Table 7 Test case: 학습 테스트 문제 만들기	35
Table 8 Test case: 통합 테스트 문제 풀기	36
Table 9 Test case: 코드 작성 및 제출하기	36
Table 10 Test case: 게시글 조회하기	37
Table 11 Test case: 게시글 작성하기	37
Table 12 Test case: 게시글 수정하기	37
Table 13 Test case: 게시글 삭제하기	38

## 13. Reference

### 13.1 Objectives

문서 작성에 참고한 참고문헌 목록을 기술한다.

### 13.2 Reference

- [https://ko.wikipedia.org/wiki/%ED%86%B5%ED%95%A9\\_%EB%AA%A8%EB%8D%B8%EB%A7%81\\_%EC%96%B8%EC%96%B4](https://ko.wikipedia.org/wiki/%ED%86%B5%ED%95%A9_%EB%AA%A8%EB%8D%B8%EB%A7%81_%EC%96%B8%EC%96%B4)
- <https://m.blog.naver.com/gongtong/150135598792>
- <https://ko.wikipedia.org/wiki/%EC%9E%90%EB%B0%94%EC%8A%A4%ED%81%AC%EB%A6%BD%ED%8A%B8>
- <https://ko.wikipedia.org/wiki/HTML>
- <https://ko.wikipedia.org/wiki/CSS>
- <https://ko.wikipedia.org/wiki/MariaDB>
- [https://en.wikipedia.org/wiki/Development\\_testing](https://en.wikipedia.org/wiki/Development_testing)
- <https://launchdarkly.com/blog/get-a-detailed-explanation-of-release-testing-several/#:~:text=Release%20testing%20refers%20to%20coding,can%20be%20released%20to%20users.>