

# Requirement Specification

소프트웨어 공학개론 5조



제출일	21.04.10	그룹	5조
과목	소프트웨어공학개론	담당교수	이은석 교수님
이름	김영현	학번	2016311033
이름	김진성	학번	2016311902
이름	안주현	학번	2014312840
이름	이규민	학번	2017314833
이름	이동현	학번	2017311710
이름	정지연	학번	2018311543

## 목차

<b>1. Preface .....</b>	<b>5</b>
<b>1.1 Objective .....</b>	<b>5</b>
<b>1.2 Readership .....</b>	<b>5</b>
<b>1.3 Document Structure .....</b>	<b>5</b>
1 Preface .....	5
2 Introduction .....	5
3. Glossary .....	5
4. User Requirement Definition .....	5
5. System Architecture .....	5
6. System Requirements Specification .....	6
7. System Model .....	6
8. System Evolution .....	6
9. Appendix .....	6
10. Index .....	6
<b>2. Introduction .....</b>	<b>6</b>
<b>2.1 Objective .....</b>	<b>6</b>
<b>2.2 Needs .....</b>	<b>6</b>
<b>2.3 Overview .....</b>	<b>8</b>
<b>3. Glossary .....</b>	<b>8</b>
<b>3.1 Objective .....</b>	<b>8</b>
<b>3.2 Term Definition .....</b>	<b>8</b>
<b>3.3 Acronyms and Abbreviations .....</b>	<b>9</b>
<b>4. User Requirement Definition .....</b>	<b>9</b>
<b>4.1 Objective .....</b>	<b>9</b>
<b>4.2 Functional Requirement .....</b>	<b>9</b>
4.2.1 회원가입 .....	9
4.2.2 로그인 .....	9
4.2.3 학습목차 .....	10
4.2.4 실습 .....	10
4.2.5 실습 결과 .....	10
4.2.6 사용자 로그 .....	10
4.2.7 사용자 프로필 .....	10
<b>4.3 Nonfunctional Requirement .....</b>	<b>10</b>
4.3.1 Product Requirement .....	10

4.3.2 Organization Requirement.....	11
4.3.3 External Requirement .....	11
<b>5. System Architecture .....</b>	<b>12</b>
5.1 Objective.....	12
5.2 System Architecture .....	12
5.3 Subsystem.....	13
5.3.1 Web Server System .....	13
5.3.2 Admin Server System .....	14
5.3.3 Grading Server System.....	15
<b>6. System Requirement Specification .....</b>	<b>16</b>
6.1 Objective.....	16
6.2 Functional Requirement.....	16
6.2.1 Table3 회원가입.....	17
6.2.2 Table4 로그인.....	17
6.2.3 Table5 학습목차.....	17
6.2.4 Table6 실습.....	18
6.2.5 Table7 실습 결과.....	18
6.2.6 Table8 사용자 로그.....	18
6.2.7 Table9 사용자 프로필.....	19
6.3 Nonfunctional Requirement.....	19
6.3.1 Product Requirement.....	19
6.3.2 Organization Requirements .....	20
6.3.3 External Requirements.....	20
6.4 Use Case.....	20
<b>7. System models .....</b>	<b>23</b>
7.1 Objective.....	23
7.2 Sequence Diagram .....	23
7.3 Structural Models .....	25
7.4 Behavioral Models.....	26
<b>8. System Requirement Evolution .....</b>	<b>27</b>
8.1 Objective.....	27
8.2 Limitation:.....	28
A. 문제의 부족 .....	28
B. Computational Resource 결여 .....	28
8.3 System Evolution .....	28

A. 학습 데이터 추가 .....	28
B. 개인 PC 자원 사용 .....	28
C. QnA 의 추가 .....	29
<b>9. Appendix.....</b>	<b>29</b>
9.1 Objective.....	29
9.2 Database:.....	29
<b>10. Index.....</b>	<b>29</b>
10.1 Objectives .....	29
10.2 Table Index.....	29
10.3 Diagram Index .....	30
10.4 Figure Index.....	30
<b>11. Reference.....</b>	<b>30</b>

# 1. Preface

## 1.1 Objective

Preface는 이 문서의 예상된 독자층과 문서의 전체적인 구조, 그리고 각 파트에서 서술할 내용들에 대해 소개한다.

## 1.2 Readership

해당 플랫폼은 Python 뿐만 아니라 프로그래밍에 대한 기초지식이 전혀 없는 비전공자를 대상으로, 쉽고 간단하게 익힐 수 있는 것을 목표로 한다. 또한 학습자의 흥미를 유발하기 위해 개/고양이의 사진 Classifier 개발을 최종 목적으로 두었다. 이는 어느 정도의 난이도가 있으면서도, 동물에 관심이 많은 비전공자가 친숙하게 접근할 수 있는 학습 방식이 될 것이다.

## 1.3 Document Structure

### 1 Preface

Preface는 이 문서의 예상된 독자층과 문서의 전체적인 구조, 그리고 각 파트에서 서술할 내용들에 대해 소개한다.

### 2 Introduction

Introduction에서는 본 시스템의 필요성과 시스템이 담아낼 기능들을 관련한 사회적 이슈에 근거하여 서술하고 소개한다.

### 3. Glossary

Glossary에서는 배경지식이 없는 독자층을 고려하여 본문에서 등장하는 용어들에 대해 설명하고 본문에서 사용되는 용어들을 정리하였다. Acronym들과 abbreviation에 대한 설명을 한다.

### 4. User Requirement Definition

User Requirement Definition에서는 본 시스템에 대한 기능적 요구사항과 비기능적 요구사항을 기술함으로써 시스템이 제공하는 서비스를 사용자의 관점에서 기술한다. 사용자의 이해를 돕기 위해 자연어로 기술한다.

### 5. System Architecture

System Architecture에서는 본 시스템의 아키텍처에 대해 상세하게 설명한다. 시스템을 구성하고 있는 각 컴포넌트들과 이들의 역할, 상호작용 등에 대해 기술하며 시각적인 자료를 이용하여 표현한다. 이 목차는 시스템의 전체적인 구조를 이해하는데 목적을 두고 있다.

## **6. System Requirements Specification**

System requirement Specification에서는 User Requirement Definition에서의 내용을 바탕으로 시스템을 구성하는 각 기능들의 기능적, 비기능적 요구사항을 상세히 서술한다. 각 기능들은 기능, 설명, 입력, 출력, 처리, 조건의 양식으로 명확하게 기술한다.

## **7. System Model**

System Requirement Model에서는 Sequence Diagram, Structural Models, Behavioral Models 과 같이 다양한 모델을 활용하여 시스템의 기능과 각 컴포넌트 간의 관계와 상호작용에 대해 설명한다. System Requirement Model 을 통해 서 시스템에 대한 이해도를 높일 수 있다.

## **8. System Evolution**

이번 장에서는 프로젝트의 한계점과 비즈니스 환경의 변화, 플랫폼의 변화, 사용자 요구 사항에 변화에 따라 어떻게 evolution고 한계점을 보완할 수 있을지 서술한다.

## **9. Appendix**

Appendix에서는 개발되는 시스템에서 사용할 구체적이고 자세한 정보를 제공한다.

## **10. Index**

Index에서는 본 문서에 사용된 표와 그림에 대한 인덱스를 나타낸다.

# **2. Introduction**

## **2.1 Objective**

Introduction은 본 시스템의 필요성과 시스템이 담아낼 기능들을 관련한 사회적 이슈에 근거하여 서술하고 소개한다.

## **2.2 Needs**

국내 디지털 교육 시장은 나날이 성장하고 있다. 2020년 194억달러, 2021년 227억 달러의 규모에 달하며 2025년에는 425억 달러로 성장할 전망이다. 정부에서는 2022년 디지털 뉴딜 예산으로 9조원을 투입하였고, 인천의 한 고등학교에서는 AI교육이 확대되었으며, 국비 지원 코딩 수강생이 1년새 4배나 증가하는 등 그 전망을 뒷받침하는 지표들이 나타나고 있다.

< '22년 한국판 뉴딜 재정투자 및 제도개선 계획 >

구분	투자(국비) (조원)				제도정비·규제개혁
	'20년	'21년	'22년	'20~'25년	
① D.N.A 생태계 강화	1.3	5.4	5.9	33.5	▶ 마이데이터 확산을 위한 개인정보보호법 개정 등
② 비대면 인프라 고도화	0.7	0.6	0.5	3.2	▶ 원격교육법 시행령 개정 등
③ 초연결 신산업 육성	-	0.5	0.8	2.6	▶ 블록체인 확산을 위한 개인정보보호법 시행령, 위치정보법 시행령 개정 등
④ SOC 디지털화	0.4	1.8	1.8	9.7	▶ 스마트 물류센터 인증 지원 및 수산물유통법 개정 등
계	2.5	8.3	9.0	49.0	

Figure1 2022년 디지털 뉴딜 예산

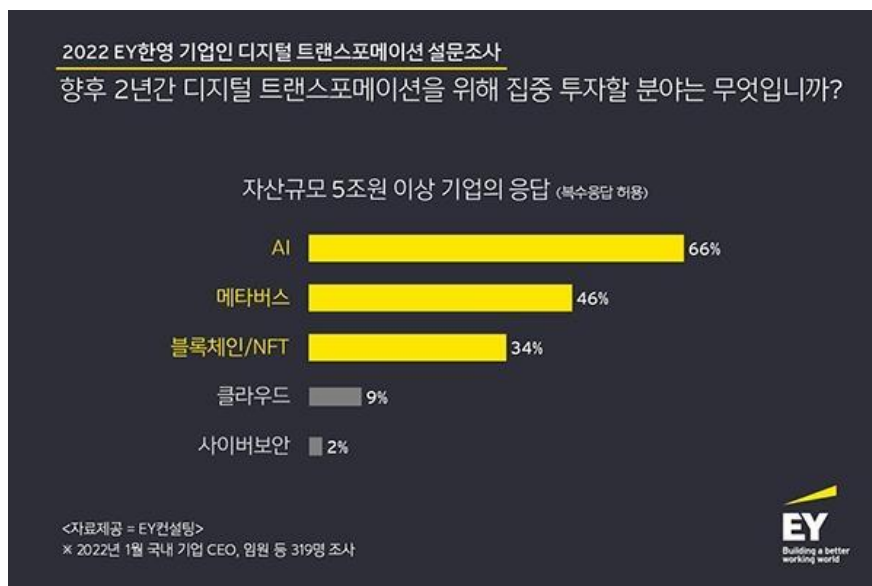


Figure2 2022 EY한영 기업인 디지털 트랜스포메이션 설문조사

하지만 코딩 교육에 있어, 기존의 교육방법으로는 해결할 수 없는 문제점이 존재한다. 첫째는 문제 해결에 대한 다수의 접근 방식이 존재한다는 점이다. 이 때문에 하나의 명확한 답안을 기준으로 채점하는 것이 불가능하다. 둘째는 코딩에 대한 지식 습득과 실습 환경이 동시에 제공되지 않는다는 점이다. 이 때문에 사용자가 독자적으로 실습을 진행해야 된다는 문제가 발생한다. 셋째는 학습자가 실습 결과에 대해 피드백을 받을 수 없다는 점이다. 이 때문에 학습자 본인이 제대로 코딩을 실행한 것인지 판단하기가 어렵다. 이에 본 글에서는 언급한 문제점들을 해소하여 새로운 코딩 학습 방법을 제시할 것이다.

## 2.3 Overview

이 항목에서는 대략적인 웹 플랫폼의 사용법에 대해 설명하겠다. 먼저 웹 사이트에 접속한 뒤 로그인하여 목차 페이지로 이동한다. 목차 페이지에서는 전체적인 학습 목차와 학습자 본인의 학습 진도와 기록을 확인할 수 있고, 해당하는 목차를 클릭함으로써 이전 강의를 다시 학습하거나, 해당하는 진도의 강의를 학습할 수 있다. 학습 페이지는 두 가지로 나뉘는데, 학습 내용을 제공하는 영역과 실제로 실습할 수 있는 영역이다. 학습자는 학습 내용을 바탕으로 실습 페이지에 직접 코드를 짜 넣어 실행할 수 있다. 실습 페이지 하단에는 오류와 로그, 결과를 출력하는 콘솔창도 있으며, 오류 발생시 그에 맞는 피드백을 제공한다.

앞에서 언급한 세가지 문제점을 해결한 코딩 학습 방법을 제시하고 그 플랫폼을 구축하는 것이 본 글의 목표이다. 구체적으로는 Python과 Classifier 개발을 위한 기초 지식을 공부할 수 있어야 하고, 동시에 단계적인 실습과 이 실습을 위한 환경을 제공해야 한다. 그리고 실습의 결과에 대해서 채점과 피드백 또한 주어져야 한다. 최종적으로는 단계적인 실습을 통해 학습자가 Deep-learning을 이해할 수 있도록 돕는, 웹 기반 학습 플랫폼을 구축하는 것이 되겠다.

## 3. Glossary

### 3.1 Objective

Glossary에서는 배경지식이 없는 독자층을 고려하여 본문에서 등장하는 용어들에 대해 설명하고 본문에서 사용되는 용어들을 정리하였다. Acronym들과 abbreviation에 대한 설명을 한다.

### 3.2 Term Definition

용어	정의
python	컴퓨터 프로그래밍 언어 중 일종으로, 인터프리터식, 객체지향식 언어이다. 문법이 간결하고, 라이브러리가 풍부하여 대학을 비롯한 여러 교육 기관, 연구 기관에서 사용된다.
indent	파이썬은 중괄호({})없이 들여쓰기를 활용하여 사용하는 영역을 지정한다.
syntax highlighting	소스 코드의 형식의 틀에 따라 여러가지 색과 글꼴로 다르게 보여주는 편집기의 기능이다.
개/고양이 Classifier	개 또는 고양이 사진을 입력으로 주었을때 어떤 항목인지 구별해내는 프로그램. 본 실습에서 최종적으로 다루게 되는 프로그램이다.
tensorflow	다양한 작업에 대해 데이터 흐름 프로그래밍을 위한 오픈소스 소프트웨어 라이브러리이다.
Keras	파이썬으로 작성된 오픈 소스 신경망 라이브러리로 tensorflow 위에서 수행할 수 있다.
log	사용자의 특정한 활동, 예를 들어 실습 내역 등에 대한 기록을 뜻한다.



실습 정확도	사용자가 제출한 코드가 통과한 test case의 비율을 뜻한다.
컴파일	사용자가 제출한 소스 코드를 컴퓨터에서 실행가능한 파일로 바꾸어주는 과정

**Table1 term definition**

### 3.3 Acronyms and Abbreviations

용어	정의
DB	Data base. 여러 사람이 공유하여 사용할 목적으로 체계화해 통합, 관리하는 데이터의 집합이다.
API	Application Programming Interface. 응용 프로그램에서 사용할 수 있도록 운영 체제나 프로그래밍 언어가 제공하는 기능을 제어할 수 있게 만든 인터페이스를 뜻한다.
UI/UX	User Interface/User Experience. 시각적으로 보여지는 소프트웨어의 디자인과 사용자에게 편리한 경험을 해주게끔 도와주는 디자인을 뜻한다.
PK	Parent Key. 관계형 데이터베이스에서 테이블의 기본키를 의미한다.
FK	Foreign Key. 관계형 데이터베이스에서 테이블이 참조하고 있는 다른 테이블의 기본키를 의미한다.

**Table2 acronyms and abbreviations**

## 4. User Requirement Definition

### 4.1 Objective

User Requirement Definition에서는 본 시스템에 대한 기능적 요구사항과 비기능적 요구사항을 기술함으로써 시스템이 제공하는 서비스를 사용자의 관점에서 기술한다. 사용자의 이해를 돕기 위해 자연어로 기술한다.

### 4.2 Functional Requirement

#### 4.2.1 회원가입

사용자는 서비스를 이용하기 위해 이메일, 비밀번호 등의 회원정보를 입력하여 회원가입을 진행해야 한다. 회원정보는 사용자를 식별하기 위한 것으로 시스템의 데이터베이스에 저장된다. 회원정보에는 이메일, 비밀번호, 이름이 포함된다.

#### 4.2.2 로그인

사용자는 서비스를 이용하기 위해서는 이메일과 비밀번호를 입력하여 로그인을 진행함으

로써 사용권한을 얻어야 한다. 로그인 후에 사용자가 기존에 학습했던 내용과 진행상황을 확인할 수 있다.

#### **4.2.3 학습목차**

사용자가 학습할 수 있는 콘텐츠가 리스트 형식으로 제시된다. 각 리스트에 나타나는 콘텐츠 제목을 클릭하면 해당 학습콘텐츠의 페이지로 이동되며 학습을 진행할 수 있게 된다. 학습내용은 시스템의 데이터베이스에 저장되며, 해당 페이지에서 실습을 동시 진행할 수 있다.

#### **4.2.4 실습**

사용자는 학습목차에서 클릭을 통해 연결된 학습콘텐츠 페이지에서 파이썬 프로그래밍에 대한 내용을 학습함과 동시에 시스템에서 제공하는 웹기반 파이썬 코드 에디터를 이용하여 학습한 내용에 대한 실습을 진행할 수 있다.

#### **4.2.5 실습 결과**

사용자는 학습한 내용에 대한 실습을 진행하고, 이에 대한 결과를 확인할 수 있다. 학습한 내용에 대한 실습을 진행하며 입력된 코드에 대한 결과를 작성된 코드 하단에 제시한다. 작성된 코드의 실행 결과를 확인함으로써 사용자는 디버깅을 진행하거나 학습한 내용에 대한 실습을 마무리할 수 있다.

#### **4.2.6 사용자 로그**

사용자는 사용자 프로필을 통해 최근에 학습했던 콘텐츠의 목록을 확인할 수 있으며, 이때 최신순으로 정렬된 최근 학습 목록의 각 항목의 클릭을 통해 학습 콘텐츠 페이지로 이동할 수 있다.

#### **4.2.7 사용자 프로필**

사용자는 로그인 이후 자신의 프로필을 확인함으로써 최근에 학습했던 콘텐츠의 목록을 확인할 수 있다. 또한 사용자가 진행한 실습 현황 및 제출한 코드의 정답 여부를 확인할 수 있다.

### **4.3 Nonfunctional Requirement**

#### **4.3.1 Product Requirement**

##### **A. usability**

사용자가 시스템을 쉽게 이해할 수 있도록 접근성이 좋고 직관적이며 사용하기 쉬운 user interface를 제공하여, 사용자가 시스템이 제공하는 서비스를 이용하기 쉽도록 제작되어야 한다. 학습내용과 실습에디터를 사용자가 동시에 볼 수 있도록 학습내용과 실습에

디터의 공간적인 분할이 이루어져야 한다. Python 에디터를 사용하여 실습을 진행하므로, 적당한 indent 간격을 지정하고 syntax highlighting을 지원하여 사용자 코드의 가독성을 높여야 한다.

#### B. efficiency

사용자 프로필에 사용자의 최근 학습목록, 진행한 실습 현황, 제출한 코드의 정답 및 오답 여부를 확인할 수 있도록 하여 사용자의 효율적인 학습을 지원해야 한다. 실습을 위해 에디터에 작성된 코드의 컴파일 시간을 최소화하여 사용자가 입력한 코드의 실행 결과를 위해 대기하는 시간을 최소화 해야한다. 학습 콘텐츠에 따른 단계적인 실습을 통해 사용자의 학습을 독려해야한다.

#### C. dependability

사용자가 회원 가입시 입력한 회원정보는 데이터 베이스에 왜곡없이 저장되어야 하며, 사용자가 올바른 회원정보(이메일, 비밀번호)를 입력했을 경우 로그인에 성공해야한다. 사용자의 학습 도중 생성된 모든 데이터는 왜곡 없이 시스템의 데이터베이스에 저장되어야 하며, 실습을 위해 제출된 코드에 대한 정확한 판별이 이루어져야 한다. 사용자가 제출한 실습 코드의 실행 결과를 사용자의 진도율과 오답률에 반영해야 한다.

#### D. security

사용자가 회원가입 및 로그인을 위해 가입한 사용자의 개인정보는 유출 및 왜곡 없이 데이터 베이스저장하고, 외부 접근으로부터 보호해야 한다. 각 학습내용에 따른 실습의 테스트케이스를 사용자로부터 보호하여 사용자가 열람할 수 없도록 해야한다. 관리자를 제외한 사용자, 혹은 외부인이 관리자 서버로 접속하여 사용자의 정보를 왜곡하거나 사용자의 정보에 접근하는 것을 방지해야 한다.

### 4.3.2 Organization Requirement

#### A. environmental

본 시스템은 web-based 서비스로, 학습의 효율성을 높이기 위해 PC 접속을 권장한다. 사용자는 인터넷에 연결된 상태에서 서비스를 제공받을 수 있으며, 크롬을 사용한 학습을 권장한다. Python 실습을 이용한 학습이 주를 이루는 서비스 환경이므로 키보드의 사용을 권장한다. 주 사용 환경은 PC를 전제로 하여, 본 시스템은 PC 사용자를 우선하여 개발한다.

#### B. operational

Front End는 NodeJs와 HTML, CSS3를 이용하여 개발한다. Back End는 MongoDB와 AWS를 이용하여 개발한다. 제공하는 학습 서비스는 python을 기반으로 이루어지며, 실습의 마지막 단계로 제시되는 개/고양이 분류기의 경우 Tensorflow의 Keras API를 기반으로 하는 학습을 제공한다.

#### C. development

사용자의 작성코드와 학습률, 오답률, 최근에 진행한 실습 목록은 사용자 데이터베이스에 저장되며, PC실습 환경의 사용자를 우선시하여 개발을 진행한다. 웹에서 사용자가 직접 python 코드를 작성하고 실행할 수 있는 환경을 제공한다.

### 4.3.3 External Requirement

#### A. regulatory

시스템이 제공하는 실습 코드의 출처를 명시하고 저작권을 확인해야 한다. 사용자의 정보 공개 범위는 사용자 본인과 관리자에 한정된다. 회원의 개인정보가 필요한 경우에는 사용자로부터 정보 제공 동의를 얻은 후 진행해야 한다. 사용자가 자신의 정보 삭제를 원할 경우 관리자 문의를 통해 삭제가 이루어질 수 있으나, 서비스 이용은 어려울 수 있다.

#### B. ethical

사용자에게 제공하는 실습의 내용은 사실을 바탕으로 작성되어야 한다. 본 시스템의 실습의 마지막 단계에서 개/고양이 분류기에 대한 학습 및 실습을 위해 Keras API를 이용하지만 이 과정에서 어느 특정 언어의 장점에 편중된 설명은 배제되어야 한다.

#### C. safety/security

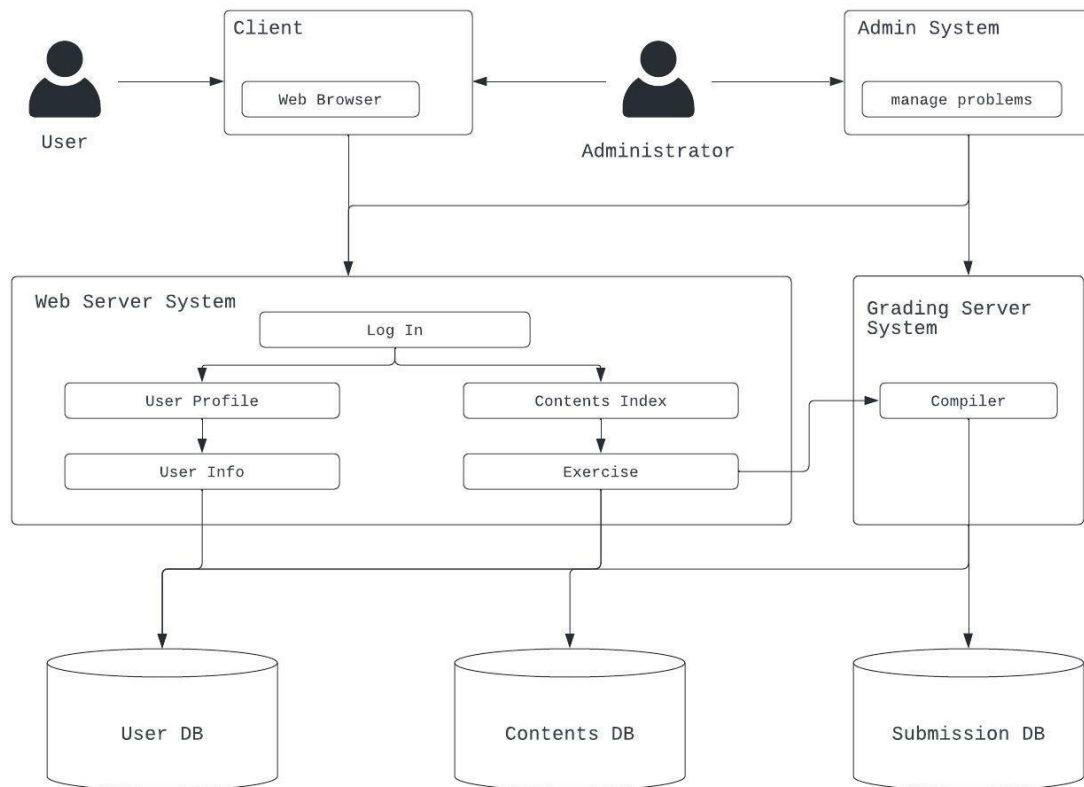
개인정보보호법에 따라 사용자의 개인정보를 보호하고 오용되지 않도록 관리해야 한다.

## 5. System Architecture

### 5.1 Objective

System Architecture에서는 본 시스템의 아키텍처에 대해 상세하게 설명한다. 시스템을 구성하고 있는 각 컴포넌트들과 이들의 역할, 상호작용 등에 대해 기술하며 시각적인 자료를 이용하여 표현한다. 이 목차는 시스템의 전체적인 구조를 이해하는데 목적을 두고 있다.

### 5.2 System Architecture

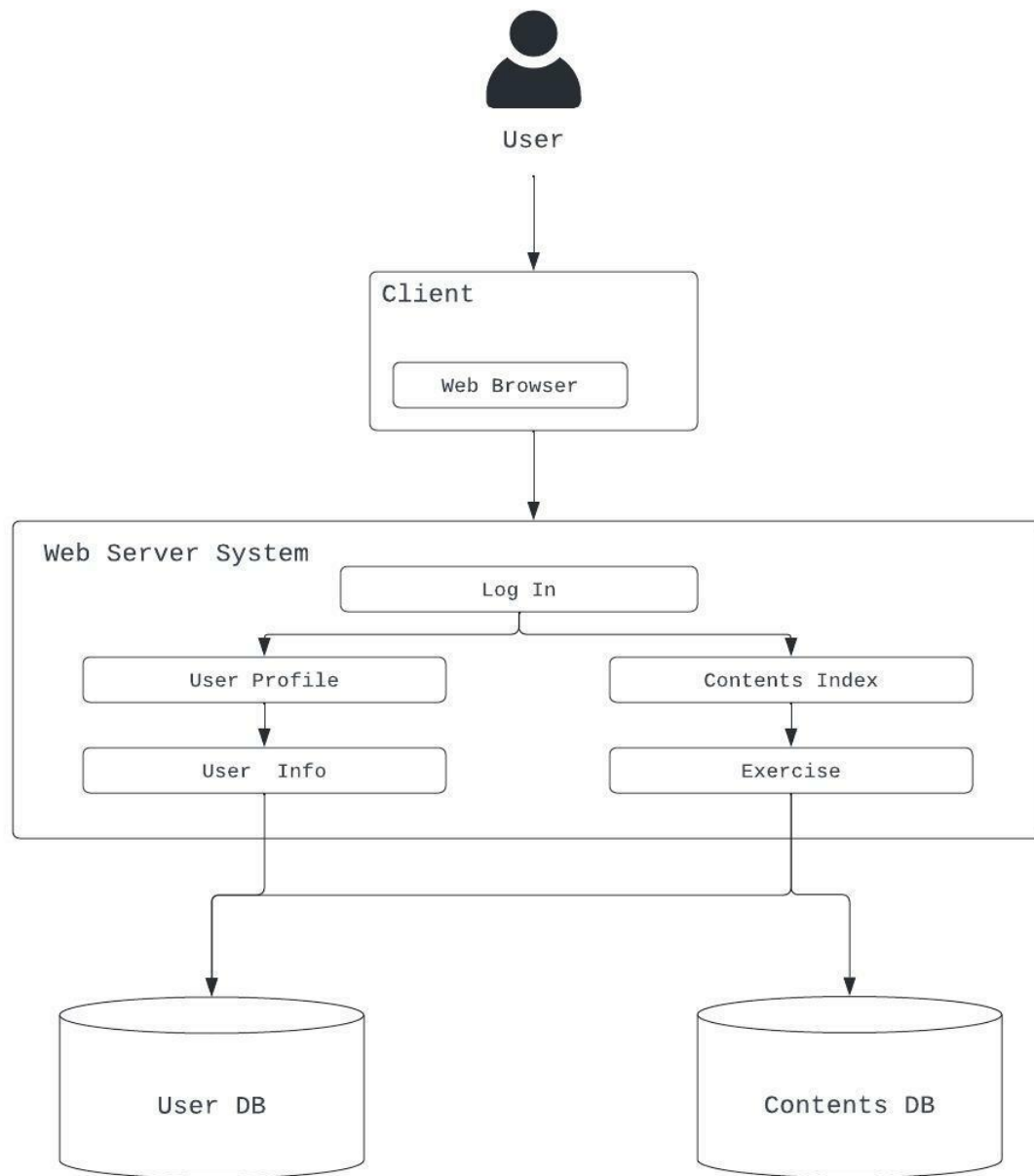


**Diagram 1 Overall System Architecture : 전체적인 시스템의 구조**

본 시스템은 Web-based 시스템으로 Web Browser을 이용하여 접속한다. Web Server System, Admin Server System, Grading Server System으로 구성된다. Web Server System은 Web의 모든 interface를 총괄한다. Admin Server System은 실습 컨텐츠에 대한 management를 담당한다. Grading Server System은 사용자의 실습 내용에 대한 데이터를 관리한다.

## 5.3 Subsystem

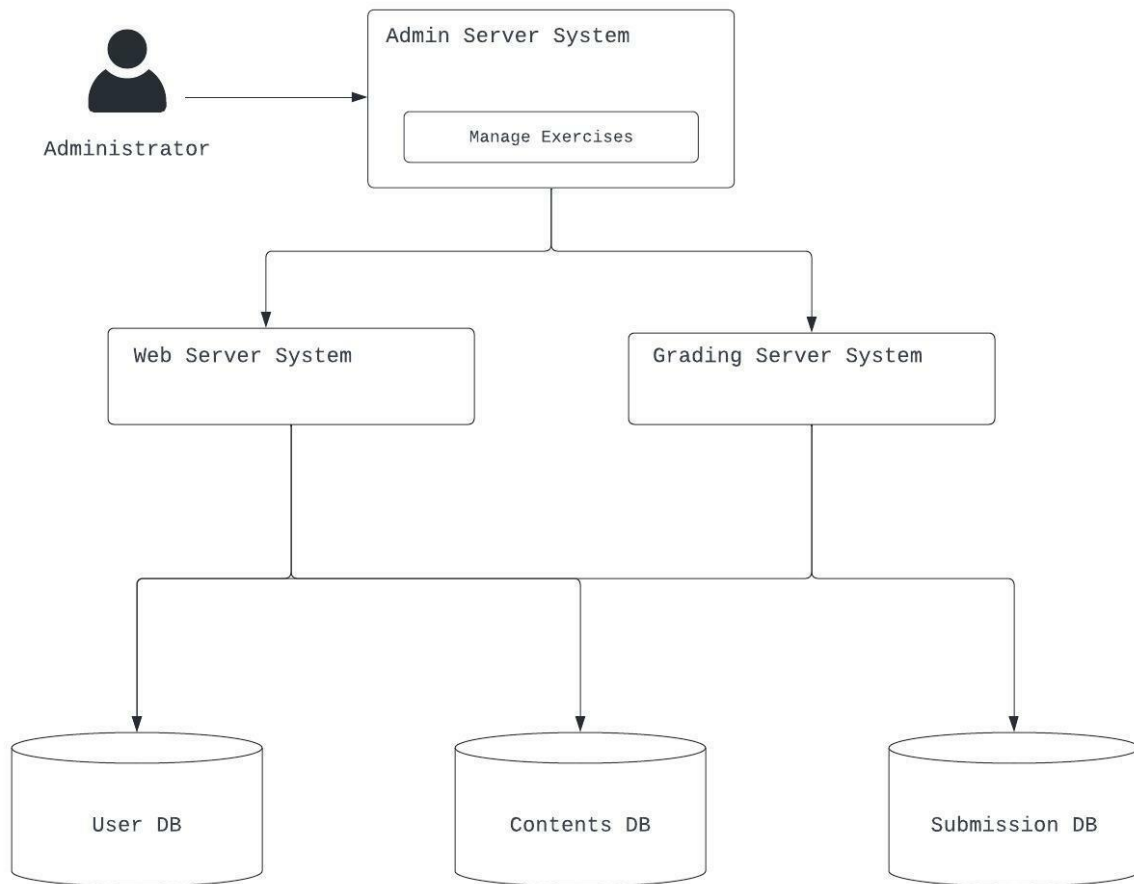
### 5.3.1 Web Server System



**Diagram 2 Web Server System Architecture : 웹 서버 시스템의 구조**

Web Server는 User에게 User Interface를 제공하며 Log-In interface, User Profile interface, User Info interface, Contents Index interface, Exercise interface를 가진다. 각 interface는 로그인, 사용자 프로필, 사용자 정보, 학습 컨텐츠 목차와 실습을 담당하며 각 페이지에서 발생한 event의 데이터는 해당 database에 저장한다.

### 5.3.2 Admin Server System



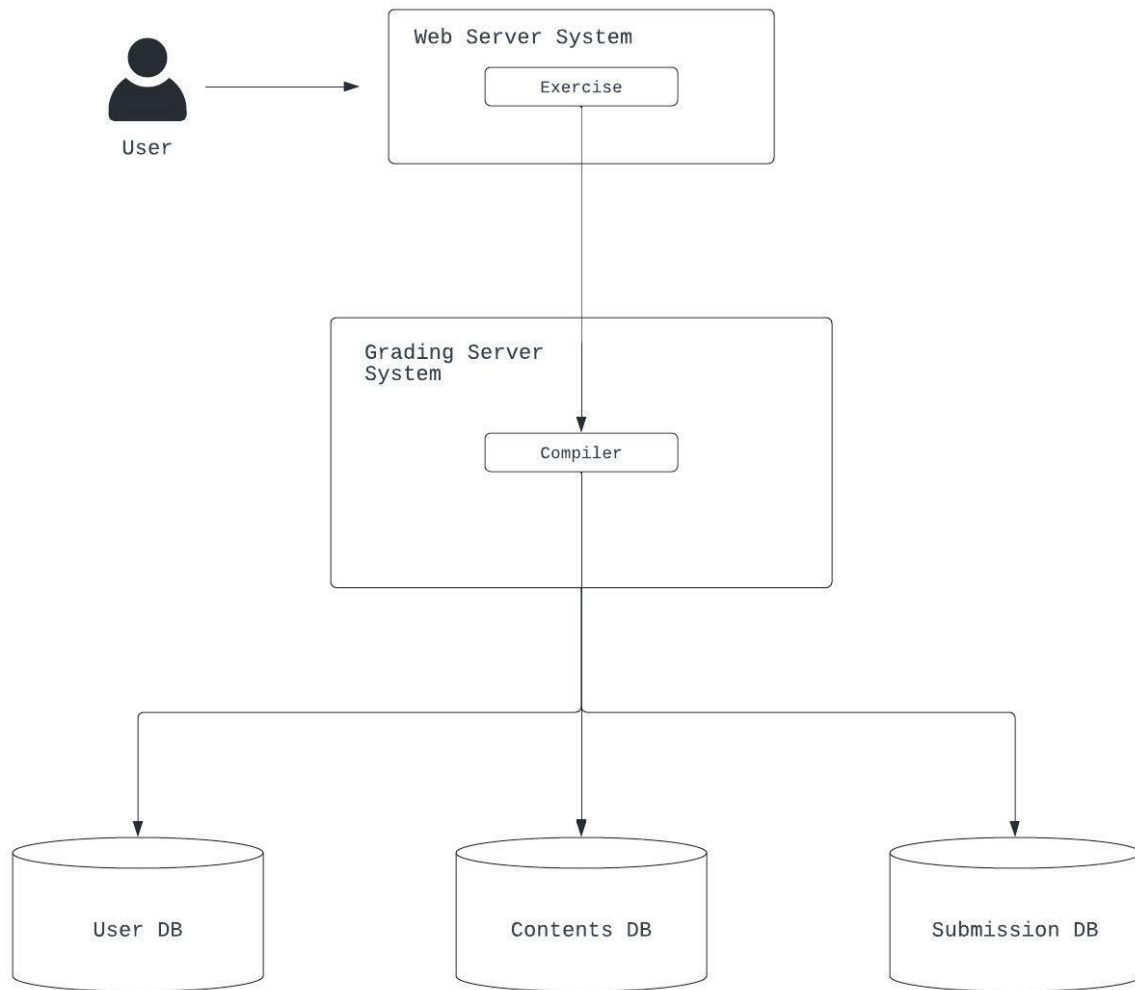
**Diagram 3 Admin Server System Architecture : 관리자 서버 시스템의 구조**

Admin Server system은 Web server와 Grading Server System과 상호운용하며 하나의 콘텐츠에 대한 추가/삭제/업데이트 및 수정을 담당한다. 관리자의 권한으로 접속하여 동작하며, 업데이트 된 콘텐츠의 내용에 대해 database에 접속하여 전체적인 실습 콘텐츠를 관리한다.

Web Server에서 추가할 실습이 존재하는 경우 관리자 권한으로 Admin Server에 접속한 뒤 실습을 추가한다. 그 후 Contents DB에 수정한 데이터를 저장하고, User DB에서 사용자의 학습 진도와 관련된 데이터를 수정한다.

Grading Server에 추가된 실습의 정답과 test case에 대한 데이터의 접근 경로를 생성하여 Grading Server와 Contents DB와의 데이터 유동 환경을 제공한다.

### 5.3.3 Grading Server System



**Diagram 4 Grading Server System Architecture : 채점 서버 시스템의 구조**

Grading Server System은 Web Server에서 제출한 사용자의 실습 코드에 대해 Compile을 진행하며 해당 코드의 정확도를 판단한다. Contents DB로부터 해당 실습에 대한 test case를 받아오며 사용자의 제출 코드에 대해 test를 진행하여 정답률을 계산한다. 이후 User DB와 Submission DB에 각각 사용자의 최근 실습 log와 제출한 코드를 저장한다.

## 6. System Requirement Specification

### 6.1 Objective

System requirement Specification에서는 User Requirement Definition에서의 내용을 바탕으로 시스템을 구성하는 각 기능들의 기능적, 비기능적 요구사항을 상세히 서술한다. 각 기능들은 기능, 설명, 입력, 출력, 처리, 조건의 양식으로 명확하게 기술한다.

### 6.2 Functional Requirement



### 6.2.1 Table3 회원가입

기능	회원가입
설명	회원 가입을 위해 조건에 맞는 사용자 정보를 입력받고, 데이터베이스에 입력받은 사용자 정보를 저장한다.
입력	이름, 이메일, 비밀번호
출력	이메일 중복 체크로 회원가입 중복 확인 입력 양식에 맞는지 확인 후, 조건 충족 여부 출력
처리	회원 가입 후 데이터베이스에 회원 정보 저장
조건	이메일은 문자열@도메인 형식이며, 중복이 되면 안된다. 비밀번호는 4~16자의 문자열, 특수문자, 숫자의 조합으로 구성 가능하다.

### 6.2.2 Table4 로그인

기능	로그인
설명	교육 서비스를 사용하기 위해 이메일과 비밀번호를 입력하며, 로그인이 진행된다면 사용권한을 얻는 기능이다. 로그인 후 자신의 학습 현황과 같은 내용을 확인할 수 있다.
입력	사용자의 이메일과 비밀번호를 입력한다.
출력	로그인에 성공했을 경우 학습 목차로 이동한다. 로그인에 실패했을 경우, 다시 시도하라는 메시지를 출력한다.
처리	데이터베이스에 저장되어 있는 이메일과 비밀번호와 입력된 정보가 일치하는지 확인하고 이를 통해 로그인 성공/실패를 결정한다.
조건	이메일은 문자열@도메인 형식이다.

### 6.2.3 Table5 학습목차

기능	학습 목차
설명	사용자가 학습할 수 있는 목차를 리스트형식으로 보여준다.
입력	화면에 표시되어 있는 목차를 클릭
출력	
처리	사용자가 클릭한 목차로 페이지를 이동한다.

조건	페이지에 표시된 목차만 클릭 가능하다.
----	-----------------------

#### 6.2.4 Table6 실습

기능	실습
설명	사용자가 저장되어 있는 실습 내용을 학습 할 수 있는 공간이며, 파이썬 코드 에디터를 이용하여 실습을 해볼 수 있는 기능이다.
입력	학습한 내용을 바탕으로 한 실습 코드 작성
출력	
처리	해당되는 목차에 맞는 실습내용을 사용자에게 표시하여 주고, 사용자가 실습 코드를 작성할 수 있는 창을 표시함.
조건	해당되는 목차에 맞는 실습 내용이 표시되며, 학습내용에 따른 실습만이 가능하다.

#### 6.2.5 Table7 실습 결과

기능	실습 결과
설명	사용자가 저장되어 있는 실습 내용을 학습 할 수 있는 공간이며, 직접 코드를 짜고 디버깅을 할 수 있는 기능이다.
입력	실습 코드 작성
출력	해당 코드를 실행한 뒤 디버깅 결과 출력
처리	사용자가 입력한 코드에 맞는 결과값을 출력하여 해당 코드가 정상적으로 작동하는지를 확인 할 수 있도록 한다.
조건	파이썬 코드로 디버깅이 진행되며, 다른 언어는 불가하다.

#### 6.2.6 Table8 사용자 로그

기능	사용자 로그
설명	사용자가 최근 학습했던 내용의 목록들을 확인할 수 있으며, 최신순으로 정렬된 항목들을 클릭함으로 해당하는 학습 목차로 이동을 할 수 있다.
입력	사용자의 학습 내역에 저장되어 있는 목차를 클릭
출력	해당되는 학습 목차로 이동

처리	해당 사용자가 이전에 학습했던 내용들을 데이터베이스에서 가져와 사용자가 클릭한 페이지로 이동한다.
조건	이전에 학습한 기록이 있어야 목록이 뜨며, 중간에 학습을 멈췄어도 기록이 되어있다.

### 6.2.7 Table9 사용자 프로필

기능	사용자 프로필
설명	로그인 이후 볼 수 있는 창이며, 최근에 학습했던 목록을 확인할 수 있다. 그리고 사용자가 진행한 실습 현황 및, 제출한 코드의 정답 여부 또한 확인할 수 있다.
입력	
출력	사용자의 학습 목록, 실습 현황, 코드의 정답여부를 출력
처리	해당 사용자의 기록된 내용을 데이터베이스에서 가져와 화면에 표시해준다.
조건	로그인이 되어 있어야 한다.

## 6.3 Nonfunctional Requirement

### 6.3.1 Product Requirement

#### A. Usability Requirement

접근성이 좋고 사용하기 쉬운 User Interface를 제공하며 시스템의 각 기능들을 찾기 쉽도록 제작이 되어야한다. 실습 내용과 사용자의 실습 코드의 공간적 분리가 명확해야한다. Python언어를 바탕으로 실습이 진행되므로 적당한 indent 간격을 지정하고 syntax highlighting을 제공하여 사용자 코드의 가독성을 높여야한다.

#### B. Efficiency Requirements

사용자 프로필에 사용자가 진행한 실습 현황, 제출한 코드의 정답 및 오답 여부, 최근에 진행한 실습 목록을 추가하여 효율적인 학습을 제공해야한다. 코드의 컴파일 시간을 최소화하여 사용자가 실습 결과를 기다리는 시간을 줄여야한다. 실습의 경우 주제에 따라 단계적인 학습을 제공하여 사용자의 원활한 진도를 제공해야한다.

#### C. Dependability Requirements

사용자의 학습 진행간 생성된 모든 데이터는 왜곡없이 데이터베이스에 저장되어야한다. 제출한 코드에 대한 정확한 판별이 이루어져야한다. 제출한 코드의 결과를 사용자의 진도율과 오답률에 반영해야 한다.

#### D. Security Requirements

사용자가 로그인에 사용한 모든 개인 정보는 유출이나 왜곡없이 데이터베이스에 저장되어야 한다. 외부의 불순한 의도의 접근에서 사용자의 개인정보를 보호해야 한다. 각 실습의 test case를 사용자로부터 보호해야 한다. 관리자 서버로 접속하여 사용자의 정보에 접근하거나 왜곡하는 것을 방지해야 한다.

### 6.3.2 Organization Requirements

#### A. Environmental Requirements

본 시스템은 web-based 서비스이고 높은 학습의 효율성을 위해 PC로 접속하는 것을 권장한다. 사용자는 인터넷에 연결된 상태에서 서비스를 제공받을 수 있다. 크롬을 사용하여 학습을 진행하는 것을 권장한다. 실습을 제공하는 환경이므로 키보드의 사용을 권장한다. 본 시스템은 PC 사용자를 우선하여 개발한다.

#### B. Operational Requirement

Front End의 경우 NodeJs, HTML, CSS3를, Back End의 경우 MongoDB, AWS를 이용하여 개발한다. 제공하는 학습 서비스는 python을 기반으로 이루어지며 실습의 마지막 단계로 개/고양이 분류기의 경우 Tensorflow의 Keras API를 기반으로 학습을 제공한다.

#### C. Development Requirement

사용자의 코드와 학습률, 오답률, 최근에 진행한 실습 목록은 사용자 데이터베이스에 저장된다. PC실습 환경의 사용자를 우선시하여 개발을 진행한다. 웹에서 직접 python코드를 적성하여 실행할 수 있는 환경을 제공한다.

### 6.3.3 External Requirements

#### A. Regulatory/Lesgilative Requirements

제공하는 실습 코드의 출처를 명시하고 저작권을 확인해야 한다. 회원의 개인 정보가 필요한 경우 정보 제공 동의를 획득한 후 진행해야 한다.

#### B. Ethical Requirements

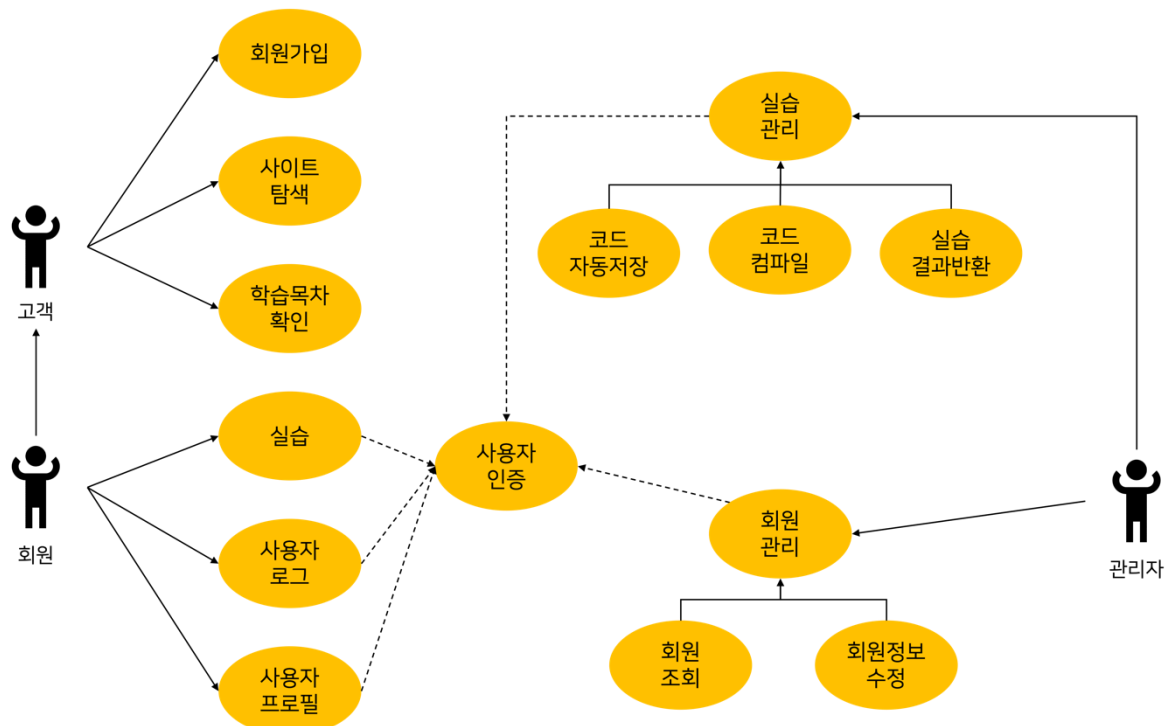
사용자에게 제공하는 실습의 내용이 사실을 바탕으로 작성되어야 한다. 본 시스템의 실습은 Keras API를 이용하지만 어느 특정 언어의 장점에 편중된 설명을 배제해야 한다.

#### C. Safety/Security Requirements

사용자의 개인 정보를 보호하고 오용되지 않도록 관리되어야 한다.

### 6.4 Use Case

#### A. Diagram 5 Use Case Model



## B. Tabular Description

6.4.1 Table10 회원가입

USE CASE	회원가입
ACTOR	User, User DB
DESCRIPTION	유저 구분을 위하여 회원 가입을 진행하고 입력 받은 사용자 정보를 토대로 데이터베이스에 사용자 정보를 저장한다.
STIMULUS	소셜 로그인 아니면 일반 로그인을 통해 사용자 정보를 입력하고 회원가입 완료 버튼을 누른다.
RESPONSE	입력한 정보를 사용자가 확인을 하고 확인 후 DB에 사용자 정보를 등록한다.
COMMENTS	

6.4.2 Table11 로그인

USE CASE	로그인
ACTOR	User, User DB
DESCRIPTION	사용자가 서비스 이용을 위해 이메일과 비밀번호를 입력하고 서버를 통해 검증된 사용자면 승인을 해준다.
STIMULUS	이메일과 비밀번호 입력 후 로그인 버튼을 누른다.
RESPONSE	로그인 정보와 DB에 저장된 사용자 정보가 대조하여 로그인 결과 반환한다.
COMMENTS	

6.4.3 Table12 학습목차

USE CASE	학습목차
ACTOR	User, Problem DB
DESCRIPTION	사용자가 웹 상에서 제공해주는 전체 문제 Overview를 보며 문제의 목표 및 목차를 보며 파악할 수 있다.
STIMULUS	사용자가 풀고자 하는 문제를 클릭한다.
RESPONSE	클린된 문제에 대한 정보와 학습 목차를 받게 된다.
COMMENTS	

6.4.4 Table13 실습

USE CASE	실습
ACTOR	User, Problem DB, Submission DB
DESCRIPTION	문제 순서에 맞춰서 사용자가 화면상에 코드를 입력한다. 입력한 코드를 가지고 온라인 Python 환경에서 Compile을 진행한다. Compile 및 실행 결과를 보여준다.
STIMULUS	온라인 IDE에서 코드를 작성하고 코드 제출 버튼을 누른다.
RESPONSE	제출된 코드를 기반으로 Compile 및 실행 결과를 알려준다.
COMMENTS	

6.4.5 Table14 사용자 로그

USE CASE	사용자 로그
ACTOR	User, User DB, Submission DB
DESCRIPTION	사용자가 문제풀이를 진행하며 작성한 코드 기록들을 자동으로 저장하며 각 로그를 기록한다.
STIMULUS	온라인 IDE 에서 코드를 작성한다.
RESPONSE	작성 후 나중에 들어왔을 때 사용자가 작성한 코드를 보여준다.
COMMENTS	

6.4.6 Table15 사용자 프로필

USE CASE	사용자 프로필
ACTOR	User, User DB, Submission DB
DESCRIPTION	로그인 된 사용자는 사용자 프로필을 통하여 회원정보를 볼 수 있고 회원정보도 수정할 수 있다. 또한, 학습 진행 상태도 알 수 있다.
STIMULUS	사용자가 로그인 후 프로필 버튼을 눌러 들어간다.
RESPONSE	사용자 프로필 화면을 제공해준다.
COMMENTS	

## 7. System models

### 7.1 Objective

System Requirement Model 에서는 Sequence Diagram, Structural Models, Behavioral Models 과 같이 다양한 모델을 활용하여 시스템의 기능과 각 컴포넌트 간의 관계와 상호작용에 대해 설명한다. System Requirement Model 을 통해 서 시스템에 대한 이해도를 높일 수 있다.

### 7.2 Sequence Diagram

Diagram 6 전체

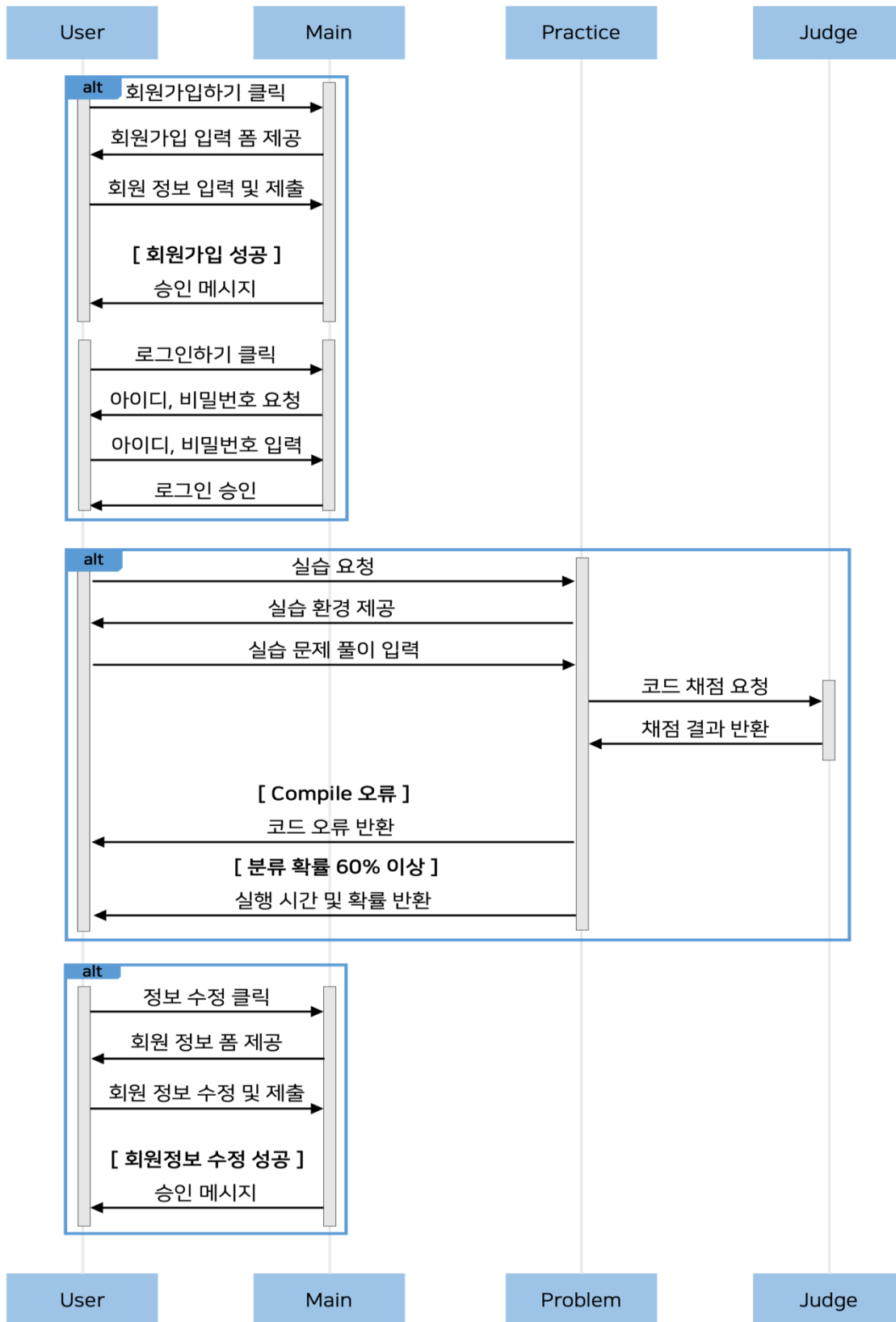


Diagram 7 회원가입



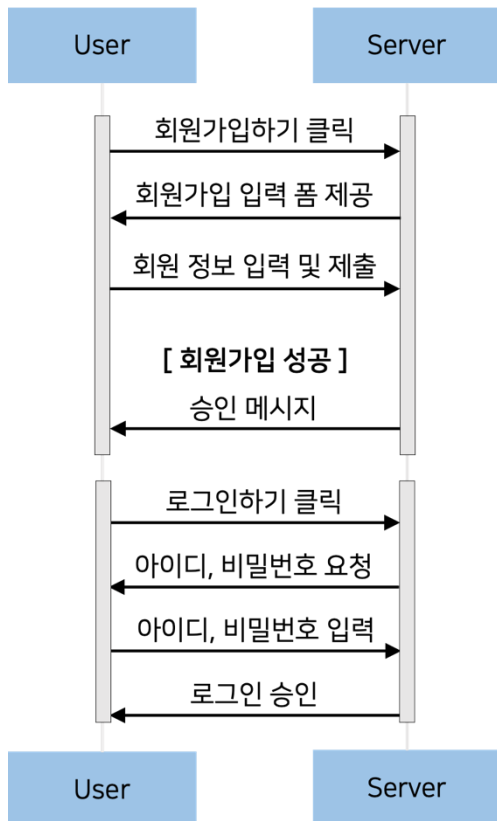
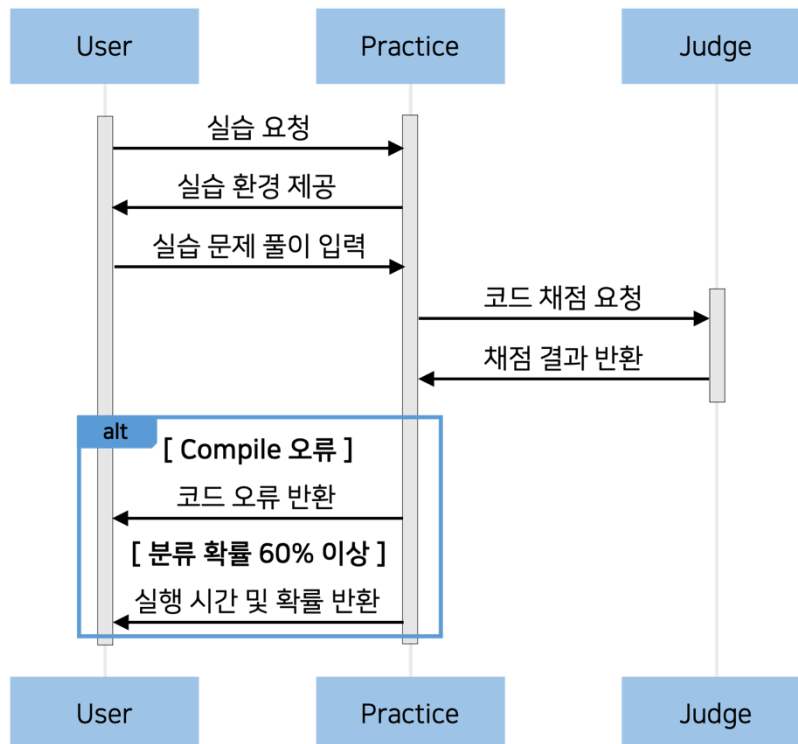
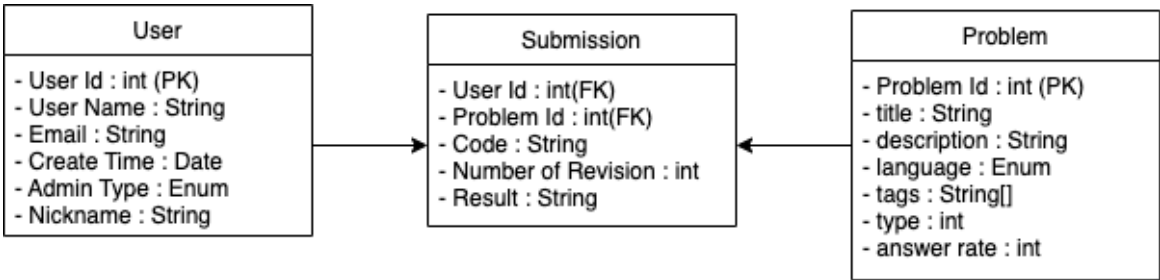


Diagram 8 실습



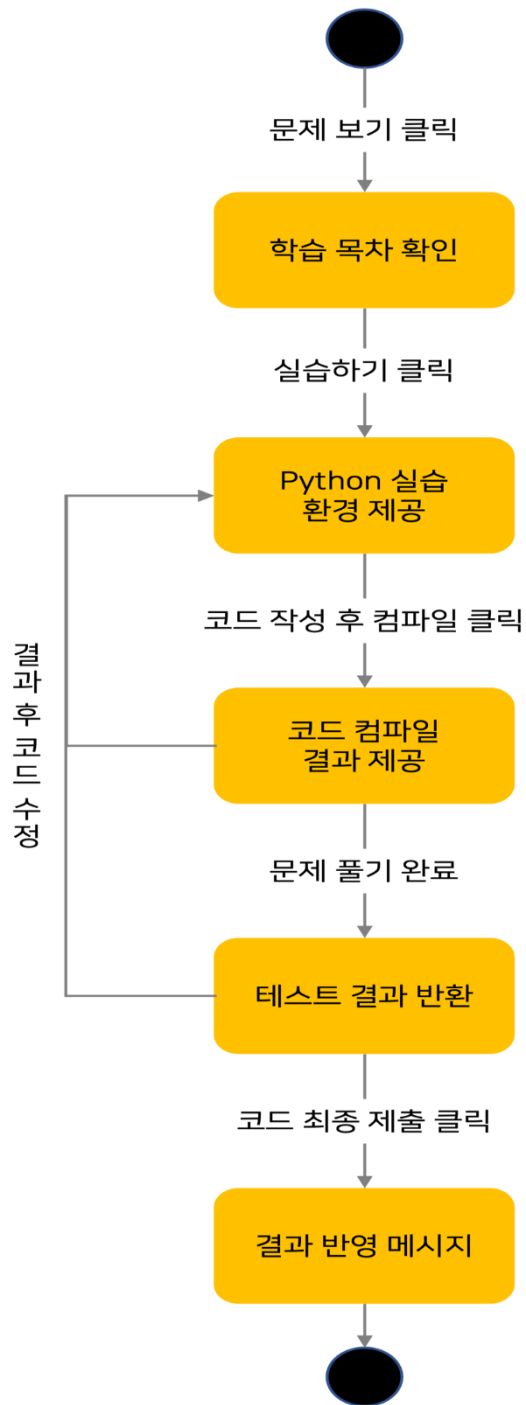
### 7.3 Structural Models

A. Diagram 9 Class Diagram



7.4 Behavioral Models

A. Diagram 10 Event Driven Diagram: 실습



## 8. System Requirement Evolution

### 8.1 Objective

이번 장에서는 프로젝트의 한계점과 비즈니스 환경의 변화, 플랫폼의 변화, 사용자 요구 사항에 변화에 따라 어떻게 evolution고 한계점을 보완할 수 있을지 서술한다.

## 8.2 Limitation:

### A. 문제의 부족

이 프로젝트는 대부분이 자체 제작한 content 혹은 타 플랫폼에서 가져온 content로 구성되어 있다. 자체 제작한 content의 문제로는 학습과정에서 학습한 내용을 확인하기 위한 문제의 수가 타 플랫폼에 비해서 적게 된다. 이로 인해서 학습자는 자신이 학습한 내용을 확인하는 과정이 부족하게 되고, 이로 인해서 학습된 내용에 대한 신뢰를 주지 못할 수도 있다.

타 플랫폼에서 가져온 content의 경우에는 자체 제작하는 문제가 아니므로 해당 문제에 어떤 문제점이 있는지 확인하기 어렵고, 이를 반드시 해당 프로젝트에서 학습할 필요가 없어서 타 플랫폼으로 사용자가 이동할 가능성이 존재한다.

### B. Computational Resource 결여

이번 프로젝트에서 사용자에게 학습시키는 내용은 딥러닝을 활용한 개/고양이 사진 Classifier이다. 딥러닝을 활용한 사진 Classifier들은 GPU, CPU 같은 자원들을 다른 프로그램에 비해서 많이 사용하고, 많은 양의 데이터 학습을 위해서는 높은 사양의 GPU, CPU를 요구한다. 그러나 이번 프로젝트에서는 고사양의 GPU, CPU를 확보할 수 없고, 이로 인해서 딥러닝 모델에 학습시킬 수 있는 내용의 양이 한정되게 된다. 만약 많은 양의 데이터를 학습시킬 경우, 상당한 시간이 걸리게 되고, 이는 사용자에게 불만족을 야기할 것이다. 그리고 이 Computational resource의 부족은 많은 수의 학습자가 접속해서 학습을 진행할 경우 더 크게 나타날 것이다.

## 8.3 System Evolution

### A. 학습 데이터 추가

사용자에게 학습시키는 내용은 라이브러리를 이용해 딥러닝 모델을 제작하고 학습시키는 것이다. 사용자는 이를 통해서 라이브러리 사용법과 라이브러리를 사용해 딥러닝 모델을 제작하고 학습시키는 것을 배울 수 있지만, 다른 머신 러닝에 대해서는 학습할 수가 없다. 이를 학습 내용을 추가해서, 다른 방법의 머신 러닝을 학습시킨다면, 딥러닝뿐만 아니라 인공지능 자체에 흥미를 느끼게 할 수 있고, 그와 관련된 지식을 습득하게 할 수 있다.

### B. 개인 PC 자원 사용

위에서도 작성한 것과 같이 현재 많은 학습자가 접속해서 딥러닝 같은 자원이 많이 들어가는 행동을 하면, 서버 전체에 부하가 올 수 있고, 이로 인해서 학습자가 불편함을 느낄 수 있다. 이를 해결하기 위해, 서버의 자원만 사용하는 것이 아니라, 학습자가 원할 경우 학습자의 PC의 자원을 이용해 모델을 학습시킬 수 있게 할 수 있다.

## C. QnA 의 추가

학습자들이 학습 데이터를 이용해 학습을 하더라도, 모든 것을 이해하는 것은 불가능 할 것이다. 이를 해결하기 위해서, 학습자가 질문한 내용을 대답해주는 QnA를 제공해야한다. 학습자가 자주 할 질문들을 미리 작성하는 것과, 학습자가 질문한 내용을 답해주는 QnA가 있으면 학습 효과를 증진시킬 수 있을 것이다.

## 9. Appendix

### 9.1 Objective

Appendix에서는 개발되는 시스템에서 사용할 구체적이고 자세한 정보를 제공한다.

### 9.2 Database:

데이터베이스의 이름은 **굵은 글씨체**, Attribute는 *기울임 꼴*, Constraint는 밑줄로 표시했다. **User DB**는 username을 필수적으로 가져야 하고, *password, email* 등을 추가적으로 가질 수 있다.

**Problem DB**는 id를 필수적으로 가져야 하고, *title, description, testcase, difficulty* 등을 추가적으로 가질 수 있다.

**Submission DB**는 id를 필수적으로 가져야 하고, *username, problem, code, result*등을 추가적으로 가질 수 있다.

## 10. Index

### 10.1 Objectives

Index에서는 본 문서에 사용된 표와 그림에 대한 인덱스를 나타낸다.

### 10.2 Table Index : 표에 대한 인덱스

Table1 term definition

Table2 acronyms and abbreviations

Table3 회원가입

Table4 로그인

Table5 학습목차

Table6 실습

Table7 실습 결과

Table8 사용자 로그

Table9 사용자 프로필

Table10 회원가입

Table11 로그인

Table12 학습목차

Table13 실습

Table14 사용자 로그

Table15 사용자 프로필

### 10.3 Diagram Index

Diagram 1 Overall System Architecutre: 전체적인 시스템의 구조

Diagram 2 Web Server System Architecture: 웹 서버 시스템의 구조

Diagram 3 Admin Server System Architecture: 관리자 서버 시스템의 구조

Diagram 4 Grading Server System Architecture: 채점 서버 시스템의 구조

Diagram 5 Use Case Model

Diagram 6 전체

Diagram 7 회원가입

Diagram 8 실습

Diagram 9 Class Diagram

Diagram 10 Event Driven Diagram

### 10.4 Figure Index

Figure1 2022년 디지털 뉴딜 예산

Figure2 2022 EY한영 기업인 디지털 트랜스포메이션 설문조사

## 11. Reference

김태진, “올해 디지털 뉴딜 예산 어디에 쓰이나”, ZDNet Korea, 2022.03.02,  
<https://zdnet.co.kr/view/?no=20220302165328>

장지용, “종로 이젠아카데미컴퓨터학원, 플랫폼 개발 기술 빅데이터 기반 AI 인공지능 국  
비교육으로 수준 높인다”, 한국강사신문, 2021.10.19,  
<http://www.lecturernews.com/news/articleView.html?idxno=78655>

이유연, “인천과학예술영재학교, 인공지능(AI) 관련 5종 교과서 개발·적용 미래사회 선도”,  
2022.03.17,  
<http://www.eduyonhap.com/news/view.php?no=62360#none;>