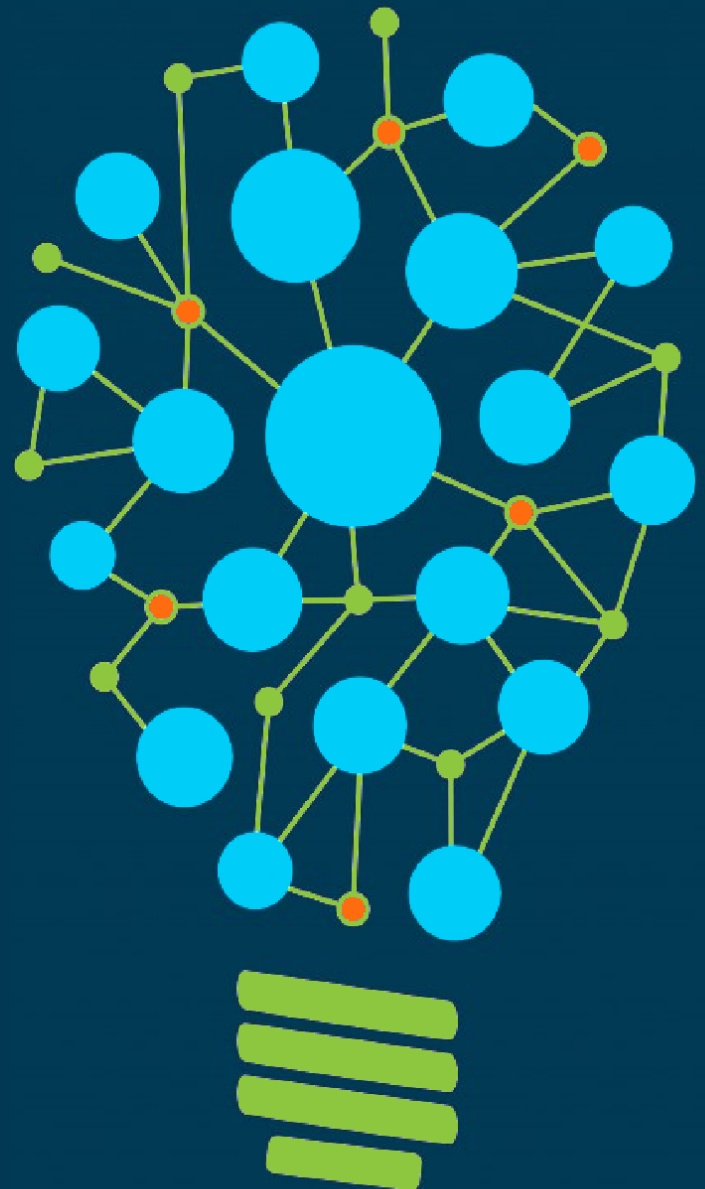


Software Requirements Specification

DOG/CAT MACHINE LEARNING ONLINE CLASS

SWE3002-41: Prof. 이은석

김기윤, 강영호, 강동준, 박상민, 이선종, 조민구



Software Requirements Specification

DOG/CAT MACHINE LEARNING ONLINE CLASS

by

김기윤, 강영호, 강동준, 박상민, 이선종,
조민구

TEAM 1

Instructor:	이은석
Teaching Assistant:	김기철, 김영경, 김진영, 허진석
Document Date:	09 April, 2022
Faculty:	SungKyunKwan University

Contents

1	Introduction	1
1.1	Purpose	1
1.2	Scope	1
1.3	Definitions, Acronyms, and Abbreviation	1
1.4	References	1
1.5	Overview	2
2	Overall Description	3
2.1	Product Perspective	3
2.1.1	System Interfaces	3
2.1.2	User Interfaces	3
2.1.3	Hardware Interfaces	3
2.1.4	Software Interfaces	3
2.1.5	Memory Constraints	3
2.1.6	Operations	4
2.1.7	Hardware Interfaces	4
2.2	Product Functions	4
2.2.1	로그인	4
2.2.2	이론 학습(코드)	4
2.2.3	이론학습(선지)	4
2.2.4	코딩실습	4
2.2.5	최종 Classifier 코드 작성	5
2.2.6	관리자 페이지	5
2.3	User Characteristics	5
2.3.1	System Administrator	5
2.3.2	User 데이터 분석능력이 필요한 전교생	5
2.4	Constraints	5
2.5	Assumptions and Dependencies	5
3	Specific Requirements	6
3.1	External Interface Requirements	6
3.1.1	User Interface	6
3.1.2	Hardware Interface	8
3.1.3	Software Interface	9
3.1.4	Communication Interface	9
3.2	Functional Requirements	10
3.2.1	Use Case	10
3.2.2	Use Case Diagram	13
3.2.3	Data Dictionary	13
3.2.4	Data Flow Diagram	14
3.3	Performance Requirements	14
3.3.1	Static Numerical Requirement	14
3.3.2	Dynamic Numerical Requirement	15
3.4	Logical Database Requirements	15
3.5	Design Constraints	15
3.5.1	Physical design constraints	15
3.5.2	Standards compliance	15
3.6	Software System Characteristics	15
3.6.1	Product Requirements	15
3.6.2	Organizational Requirements	16

3.6.3	External Requirements	16
3.7	Organizing the Specific Requirements	16
3.7.1	Context Model	16
3.7.2	Process Model	17
3.7.3	Interaction Model	17
3.8	System Architecture	17
3.9	System Evolution	17
3.9.1	Limitation and Assumption	17
3.9.2	Evolution of Hardware and Change of User Requirements	18
4	Supporting Information	19
4.1	Software Requirements Specification	19
4.2	Document History	19

List of Figures

3.1	Use Case Diagram	13
3.2	Data Flow Diagram	14
3.3	Context Model	16
3.4	Overall Process Model	17
3.5	System Architecture	17

List of Tables

1.1	Table of acronyms and abbreviations	1
1.2	Table of terms and definitions	1
3.1	User interface of input processing	6
3.2	User interface of main page	7
3.3	User interface of register	8
3.4	Hardware interface of applicable device for the system	8
3.5	Software interface of applicable device for the system	9
3.6	Communication interface of applicable device for the system	9
3.7	Use Case of Register	10
3.8	Use Case of Log-in/out	11
3.9	Use Case of Learning (Lesson & Quiz)	12
3.10	Use Case of Coding	12
3.11	Account	13
3.12	Problem	14

1

Introduction

1.1. Purpose

이 문서는 전교생을 대상으로 Python을 통해 Cat and Dog Classifier 구현해보는 것을 도와주는 프로그램의 requirements를 명확히 하고, 추 후 유지보수하는 과정에서 참고할 수 있도록 제작되었다.

1.2. Scope

머신러닝과 파이썬에 대한 이해 + (pytorch를 이용한 CNN을 이용해 Cat and Dog 뿐만 아니라 다른 data에도 적용하여 ex) MNIST data 또한 classification할 수 있다.)

1.3. Definitions, Acronyms, and Abbreviation

<i>Acronyms & Abbreviations</i>	<i>Explanation</i>
CNN	Convolutional Neural Network

Table 1.1: Table of acronyms and abbreviations

<i>Terms</i>	<i>Definitions</i>
Machine Learning	데이터를 분석하고 해당 데이터를 통해 학습한 후 정보를 바탕으로 결정을 내리기 위해 학습한 내용을 적용하는 알고리즘
Deep Learning	머신 러닝의 한 분야로서, 여러 Layer를 통해 사람의 뇌와 같은 neural network로 빅데이터를 학습하는 알고리즘
Optimization Method	딥러닝에서 Loss function을 최소화하는 기법
Classification	기존에 존재하는 데이터의 category 관계를 파악하고, 새롭게 관측된 데이터의 category를 스스로 판별하는 것
Judge	User가 제출한 답이 정답과 일치하는지 판단

Table 1.2: Table of terms and definitions

1.4. References

IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications, In IEE-EXplore Digital Library

<http://ieeexplore.ieee.org/Xplore/guesthome.jsp>

Pytorch

<https://pytorch.org/docs/stable/index.html>

2020 Spring 41 class team1

https://github.com/skkuse/2020spring_41class_team1

1.5. Overview

본 소프트웨어 요구사항 명세서는 네 챕터로 구성되어 있다. 첫번째 챕터에서는 Team1에서 제시하는 **Cat and Dog Classifier** 교육실습 프로그램 서비스와 그 목적에 대한 소개 및 본 문서에서 사용하는 약어 및 용어에 대한 설명을 하고 있다. 두번째 챕터에서는 시스템 인터페이스 및 기능, 다른 시스템과의 상호작용을 포함한 **product perspective**에 대한 전반적인 설명을 제공한다. 세번째 챕터에서는 외부 인터페이스, 기능 등을 포함한 자세한 요구사항 명세화를 진행한다. 네번째 챕터는 본 명세서가 어떤 기준이나 문서를 참고하였는지 알려준다. 팀의 모든 멤버는 본 문서 작성 및 프로그램을 설계하고 구현하는데 기여하였다.

2

Overall Description

2.1. Product Perspective

딥 러닝을 활용한 개-고양이 사진 classifier를 학습 목표로 하는 code learning platform 시스템은 다수의 성균관대학교 학생을 대상으로 python programming에 대한 선행 지식이 없더라도 개-고양이 사진 classifier 프로그램을 만들어가는 과정을 진행하며 학습할 수 있도록 하기 위해 설계되었다. Python의 기초부터 개-고양이 사진 classifier 구축까지 진행하며 학생들은 python의 기능을 공부하고 퀴즈와 프로그래밍 실습을 통해 자신이 얼마나 수업 내용을 이해했는지 확인할 수 있다. 또한 교수는 퀴즈와 프로그래밍 실습의 결과가 저장된 학생 데이터베이스를 바탕으로 학생들의 수업 진행 현황을 확인할 수 있고 학점을 부여하는 데에 참고 자료로 활용할 수 있다. 또한 교수는 algorithm judge 시스템을 활용하여 학생들에게 python 프로그래밍 문제를 제시하고 그에 따른 학생들의 결과를 얻을 수 있다.

2.1.1. System Interfaces

학생들은 python 및 개-고양이 사진 classifier 제작 학습 사이트를 통해 이론 학습을 마친 후 주어진 문제를 해결한다. 퀴즈의 답 및 문제풀이 코드의 정보를 평문으로 시스템의 학생 데이터베이스에 저장한다. 또한 algorithm judge 시스템을 통해 학생이 작성한 python 코드를 실행하여 문제 데이터베이스와 같은 출력 값을 가지는지 비교하여 정답 여부를 학생 데이터베이스에 전달한다. 코드의 제출 결과들은 학생과 교수 모두 데이터베이스를 통해 정답인지 오답인지 코드에 오류가 있는지 열람할 수 있다.

2.1.2. User Interfaces

인터페이스는 컴퓨터 화면을 통하여 제공한다. 학생들에게 python 학습 사이트를 컴퓨터를 통해 접속할 수 있도록 하며 좌측에 강의 내역의 section별로 목차를 제공하여 원하는 내용에 쉽게 접근할 수 있도록 한다. 각 section에서는 강의 영상 및 강의 내용을 시각적으로 제공하고 각 강의의 내용을 바탕으로 코딩을 실습할 수 있도록 제시한다. 또한 학생과 교수 모두 학생이 만든 코드의 내용과 그 결과를 저장 및 확인할 수 있다.

2.1.3. Hardware Interfaces

본 시스템은 웹 어플리케이션으로 컴퓨터를 통해 사용할 수 있다. 이 때 요구되는 사양은 1.0GHz 이상의 싱글 프로세서 데스크탑 및 노트북이다.

2.1.4. Software Interfaces

본 시스템은 적어도 Windows 7 이상의 OS에서 동작할 수 있으며 Windows 10 및 Windows 11의 환경을 타겟으로 하고 있다. 또한 Chrome 버전 100.0, Microsoft Edge 버전 100.0, Firefox 버전 99 이상의 웹 브라우저를 사용함을 권장한다. 그리고 학습을 정상적으로 진행하기 위해 학생 및 교수의 웹 브라우저에서 javascript 실행을 허용해야 한다.

2.1.5. Memory Constraints

본 시스템은 적어도 2GB의 RAM을 가진 컴퓨터에서 접속하는 것을 권장한다.

2.1.6. Operations

System Administrator

- 문제 제출: 문제 데이터베이스에 문제를 저장하여 학생들에게 해결해 보도록 한다. 학생들이 제출한 답안의 결과 또한 확인할 수 있다.

2.1.7. Hardware Interfaces

- 로그인: 학생은 자신의 학번을 ID로 사용하여 로그인할 수 있다.
- 회원가입: 학생은 자신의 학번을 ID로 사용하여 회원가입을 할 수 있다.
- 강의 내용 학습: 저장된 강의를 강의 웹사이트를 통하여 열람할 수 있다. 또한 강의 내용에 관련된 python 코딩 실습을 직접 스크립트를 만들며 쉽게 이해할 수 있다.
- 문제 해결 및 답안 제출: 문제 데이터베이스에 저장된 문제의 답안을 웹사이트 내에 코드로 입력할 수 있다. 이때 제출한 답안이 맞거나 틀렸는지 또는 오류가 있는지 알 수 있다.
- Product Perspective

system interfaces: 학습자가 이론 학습 후 코드 정보와 문제풀이 정보를 평문으로 서버로 전송한다. 그 후 사용자 코드를 실행하여 서버 DB와 비교 후 답안 확인 후 사용자 웹클라이언트에게 돌려준다.

user interfaces: 학습자가 웹 사이트를 이용 → 교육이 코딩을 필요로 하는 만큼 컴퓨터 활용을 recommend

hardware interfaces: 컴퓨터를 통해 사용할 수 있음. 요구 되는 사항은 https://github.com/skkuse/2021spring_41class_team9/blob/main/docs/RS_TEAM9.pdf 참조

software interfaces: Window 7 이상 OS, Window 10, 11 환경을 타겟. 브라우저 버전 정보 크롬/EDGE/파이어폭스/사파리, enable javascript in browser

communications interfaces: memory interface

memory constraints: 크롬 최소사양

operations: system administrator and user

2.2. Product Functions

2.2.1. 로그인

서비스가 각 user의 학습 진행 현황을 알고 user가 사이트를 다시 방문하였을 때, 진행해야 하는 곳을 알려주기 위해 필요한 기능이다. (ex) 회원가입 버튼을 통해 아이디와 비밀번호를 입력하고, 다음에 사이트를 방문했을 때 가입한 아이디와 비밀번호를 입력하여 본인이 학습하던 부분부터 다시 시작할 수 있게 하여야 한다.

2.2.2. 이론 학습(코드)

user가 Cat and Dog Classifier를 만들기 전에 기본이 되는 Machine Learning의 이론과 Tensor에 대한 이해, Tensor를 기반으로 만들어지는 CNN에 대한 이해가 선행되어야 한다. 기본적인 개념을 소개함과 동시에 개념을 코드로 직접 작성할 수 있도록 똑같은 코드를 제공하고, 이후에 개념을 이해해야 살짝 응용하여 풀 수 있는 문제를 보여주고 user가 문제를 풀고 답을 출력하게 하여 출력된 값과 답을 비교해 개념의 이해를 도우는 방식으로 진행된다.

2.2.3. 이론 학습(선지)

user가 Linear Regression이나 gradient descent algorithm과 같은 알고리즘을 잘 이해할 수 있도록 문제를 선지 형태로 내어서 맞추게 하고 선택한 답에 따라 피드백을 주는 형태로 교육을 진행한다. relu, softmax와 같은 machine learning에 필요한 함수의 경우에도 특정값을 이 함수들에 넣으면 어떤 결과가 나오는지 계산을 하도록 유도하고 정답을 객관식으로 맞추게 한다.

2.2.4. 코딩실습

user에게 먼저 CNN에 대해 알려주고, CNN모형을 python code를 통해 직접 구현해보고 CNN모형을 시각화해서 볼 수 있도록 한다. Optimization method의 다양한 종류들에 대해 알려주고, 각 method의 쓰임을 직접 코드를 구현해보면서 간략하게 알게 해준다.

2.2.5. 최종 Classifier 코드 작성

지금까지 교육하면서 user가 작성했던 코드들을 기반으로 전체 Classifier Code를 각각 Data Loading, CNN Model, Optimizer, Train, Test로 구분화해서 보여주고. (Discussion 필요) 직접 학습을 시켜보고 Test해서 모델의 Accuracy가 얼마나 되는지 볼 수 있게 해주면서 학습을 마친다.

2.2.6. 관리자 페이지

이 페이지에서 관리자가 각 단계별 문제와 정답을 설정할 수 있다.

2.3. User Characteristics

2.3.1. System Administrator

웹 서비스에 충분한 지식을 가진 사람이 System administrator 역할을 할 수 있다. 컴퓨터 공학을 전공하였거나, 그에 준하는 웹 개발 관련 지식을 가진 사람이 추후 발생하는 문제를 다룰 수 있다. Teaching - flow 관련하여서는 python을 기반으로 하여, CNN 모델을 딥러닝시켜본 경험자와 pytorch에 기반한 지식을 가져야 더 심화적인 문제, 기본 문제들을 추가할 수 있다.

2.3.2. User 데이터 분석능력이 필요한 전교생

웹 서비스에 충분한 지식을 가진 사람이 System administrator 역할을 할 수 있다. 컴퓨터 공학을 전공하였거나, 그에 준하는 웹 개발 관련 지식을 가진 사람이 추후 발생하는 문제를 다룰 수 있다. Teaching - flow 관련하여서는 python을 기반으로 하여, CNN 모델을 딥러닝시켜본 경험자와 pytorch에 기반한 지식을 가져야 더 심화적인 문제, 기본 문제들을 추가할 수 있다.

2.4. Constraints

새로 개발된 Dog/Cat Machine Learning 플랫폼은 이 양식에 작성된 대로 설계 및 개발될 것임.. 다음 목록은 학습 플랫폼에서 고려해야 할 제약 조건임..

- 코딩을 위한 화면 크기의 제한으로 인해 휴대폰이 아닌 컴퓨터를 사용.
- 학생들은 자바스크립트가 활성화된 인터넷 브라우저를 사용해야 함.
- 개발은 Windows 10 환경을 대상으로 함.
- 코딩에 익숙하지 않은 사용자도 플랫폼을 쉽고 직관적으로 학습할 수 있어야 합니다.
- 시스템 개발 및 유지 관리 비용을 고려해야 합니다.
- 시스템의 미래 확장성 및 가용성 고려해야 함.

2.5. Assumptions and Dependencies

플랫폼의 궁극적인 목표는 머신러닝을 코딩하는 방법을 가르치는 것이기 때문에 컴퓨터처럼 코딩하기 쉬운 환경에서 수업을 배운다고 가정합니다. 이러한 가정 환경으로 인해 양식에 언급된 최소 사양을 충족하지 않는 장치에서는 시스템이 원활하게 작동하지 않을 수 있습니다.

3

Specific Requirements

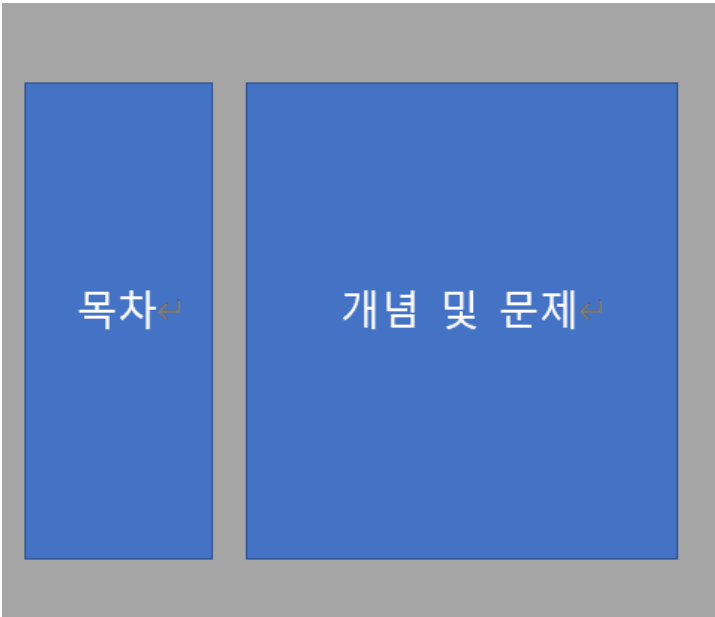
3.1. External Interface Requirements

3.1.1. User Interface

이름	마우스 및 키보드를 통한 입력 처리
목적/내용	시스템 사용자가 키보드 및 마우스의 입력을 통해 시스템에 명령 전달
입력 주체/출력 목적지	사용자/Windows 기반의 컴퓨터 기기
범위/정확도/허용 오차	<ul style="list-style-type: none"> • 범위: 화면에서의 버튼의 개수에 따른 입력 범위. • 정확도: 유저의 마우스 및 키보드 입력에 따른 정확도. • 허용 오차: 해당 없음
단위	버튼 클릭/키보드 입력
시간/속도	비정기적인 사용자의 입력/즉각적인 사용자 명령 수행
타 입출력과의 관계	입력 내용에 따라 클라이언트에서 처리 또는 서버로 명령 요청
화면 형식 및 구성	해당 없음
윈도우 형식 및 구성	해당 없음
데이터 형식 및 구성	Python 코드, Text, Int 형의 코드 값
명령 형식	각 코드 값에 따른 명령 매핑
종료 메시지	해당 없음

Table 3.1: User interface of input processing

이름	모니터를 통한 메인 화면 출력
목적/내용	시스템 사용자에게 제공하는 인터페이스
입력 주제/출력 목적지	클라이언트/사용자
범위/정확도/허용 오차	<div><ul style="list-style-type: none">• 범위: 화면에서의 버튼의 개수에 따른 입력 범위.• 정확도: 유저의 마우스 및 키보드 입력에 따른 정확도.• 허용 오차: 해당 없음</div>
단위	화면
시간/속도	사용자의 입력에 따른 화면 전환
타 입출력과와의 관계	사용자의 입력을 위한 인터페이스로서 출력 후 사용자의 입력 대기
화면 형식 및 구성	



- 개발할 웹사이트는 좌측의 목차와 우측의 개념 및 문제의 형태로 구성
- 목차를 보고 학생들은 순차적으로 학습을 진행해 나갈 수 있음
- 개념을 기본적으로 설명해준 후 그 밑에 고안한 문제를 풀고 다음 chapter로 넘어가는 진행

윈도우 형식 및 구성	<div><ul style="list-style-type: none">• Linear Layout 형식으로 목차 및 개념 구성• Linear Layout 형식으로 문제 구성• Python code를 작성하는 IDE 창</div>
데이터 형식 및 구성	이미지, 텍스트
명령 형식	해당 없음
종료 메시지	해당 없음

Table 3.2: User interface of main page

이름	사용자 등록 인터페이스
목적/내용	시스템에 사용자의 정보를 등록
입력 주체/출력 목적지	사용자/서버
범위/정확도/허용 오차	해당 없음
단위	화면
시간/속도	해당 없음
타 입출력과의 관계	해당 없음
화면 형식 및 구성	<ul style="list-style-type: none"> • 새로운 계정에 대한 정보 입력을 위한 슬롯 • 계정 정보 등록을 위한 버튼
윈도우 형식 및 구성	해당 없음
데이터 형식 및 구성	Query
명령 형식	등록 버튼에 따른 명령 매핑
종료 메시지	“가입되었습니다”

Table 3.3: User interface of register

3.1.2. Hardware Interface

이름	시스템에서 사용 가능한 디바이스
목적/내용	키보드, 마우스를 사용한 사용자의 입력
입력 주체/출력 목적지	사용자/서버
범위/정확도/허용 오차	해당 없음
단위	해당 없음
시간/속도	사용자의 입력/문제 풀이에 해당하는 처리
타 입출력과의 관계	해당 없음
화면 형식 및 구성	해당 없음
윈도우 형식 및 구성	해당 없음
데이터 형식 및 구성	해당 없음
명령 형식	코드
종료 메시지	해당 없음

Table 3.4: Hardware interface of applicable device for the system

3.1.3. Software Interface

이름	웹 사이트
목적/내용	화면 출력
입력 주체/출력 목적지	해당 없음
범위/정확도/허용 오차	Chrome, Edge, Firefox, Safari와 같은 웹 브라우저에서 사용 가능
단위	해당 없음
시간/속도	새로 고침에 따른 즉각적인 처리
타 입출력과의 관계	해당 없음
화면 형식 및 구성	웹 브라우저를 통한 웹사이트 출력
윈도우 형식 및 구성	해당 없음
데이터 형식 및 구성	해당 없음
명령 형식	해당 없음
종료 메시지	해당 없음

Table 3.5: Software interface of applicable device for the system

3.1.4. Communication Interface

이름	호스트 서버 - 클라이언트
목적/내용	각 클라이언트에서 호스트 서버에 접속을 요청하고, 사용자가 입력한 정답 및 코드를 호스트 서버에서 각 클라이언트에게 전달받고 이에 해당하는 결과를 제공
입력 주체/출력 목적지	클라이언트와 호스트서버
범위/정확도/허용 오차	해당 없음
단위	패킷
시간/속도	최소 10Mbps 이상
타 입출력과의 관계	해당 없음
데이터 형식	<ul style="list-style-type: none"> • Struct를 이용한 명령 코드 (문제 채점 요청, 코드 실행 등) • Struct를 이용한 계정 로그인 데이터
명령 형식	Send() 콜에 의한 통신
종료 메시지	Close() 콜에 의한 소켓 종료

Table 3.6: Communication interface of applicable device for the system

3.2. Functional Requirements

3.2.1. Use Case

<i>Use case name</i>	<i>Register</i>
Actor	등록하지 않은 사용자 (학생)
Description	미등록 사용자가 시스템의 모든 기능을 이용하기 위해 회원으로 등록을 시도하는 과정이다.
Normal Course	<ol style="list-style-type: none"> 모든 사용자는 웹사이트 접속 후 하단에 로그인 페이지가 나타난다. 등록 후 사용자에게만 서비스가 제공되는 것을 확인한 후, 미등록 사용자는 로그인 페이지에서 등록 버튼을 클릭한다. 사용자가 페이지를 등록하도록 등록 페이지로 이동한다. 사용자는 등록 양식에 따라 추가 정보를 입력해야 한다. 필요한 정보는 다음을 포함한다. <ol style="list-style-type: none"> ID (이메일 주소) Password 학번 시스템은 이메일 주소가 올바른지 확인하고 암호를 찾는 상황을 준비하기 위해 지정된 전자 메일 주소로 확인 코드를 보낸다. 양식을 작성한 후 사용자가 등록되고 양식 끝에 있는 등록 버튼을 클릭하면 로그인 페이지로 돌아간다.
Precondition	<ul style="list-style-type: none"> 사용자가 아직 시스템에 등록되지 않아야 한다. 사용자가 올바른 정보를 입력해야 한다. 동일한 이메일 주소를 다른 사용자의 이메일 주소와 중복해서는 안 된다. 잘못된 입력이 있는 경우, 시스템은 이메일 주소 및 비밀번호의 형식을 확인해야 한다.
Post Condition	보안을 위해 비밀번호를 암호화하고 사용자 데이터베이스에 저장한다.
Assumptions	해당 없음

Table 3.7: Use Case of Register

Use case name		Log-in/out
Actor	등록하지 사용자	
Description	<ul style="list-style-type: none"> 로그인은 시스템의 등록된 사용자가 서비스를 사용하기 위해 시스템에 들어 가려고 하는 프로세스이다. 로그아웃은 로그인한 사용자가 시스템에서 나가려고 할 때의 프로세스이다. 	
Normal Course	<ul style="list-style-type: none"> 로그인 <ul style="list-style-type: none"> 시스템에 이미 회원으로 등록한 사용자가 시스템에서 서비스를 사용하려고 한다. 등록을 위해 설정한 전자 메일 주소와 암호를 입력한다. 정보가 올바르면, 시스템은 사용자가 시스템에 접속할 수 있도록 하며, 이제 사용자는 시스템에서 제공하는 모든 서비스를 이용할 수 있다. 로그아웃 <ul style="list-style-type: none"> 시스템을 종료하려면 'logout' 버튼을 클릭한다. 사용자가 로그아웃하지 않고 응용 프로그램을 닫은 경우 시스템이 해당 사용자에게 대한 세션을 임의로 닫는다. 	
Precondition	<ul style="list-style-type: none"> 로그인: 사용자가 시스템에 이미 등록되어 있어야 한다. 로그아웃: 사용자가 로그인 상태여야 한다. 	
Post Condition	사용자가 온라인 상태여야 한다.	
Assumptions	해당 없음	

Table 3.8: Use Case of Log-in/out

<i>Use case name</i>	<i>Learning (lesson & quiz)</i>
Actor	등록된 사용자
Description	학습 과정은 회원으로 등록된 사용자가 시스템에서 제공하는 cat/dog classifier 에 대한 내용 및 머신 러닝에 필요한 기초 지식을 웹사이트를 통해 습득하고 이를 퀴즈를 통해 확인하는 프로세스이다.
Normal Course	<ul style="list-style-type: none"> • 모든 사용자는 로그인 후 학습 페이지가 나타난다. • 자신의 저장되어 있는 학습 진도를 확인한 후, 알맞은 이후 학습 페이지로 이동한다. • 새롭게 나타난 창에 있는 설명 및 이론을 읽으며 학습한다. • 학습을 마친 후 객관식 및 주관식으로 되어있는 퀴즈 탭을 눌러 퀴즈 창으로 이동한다. • 퀴즈를 푼다. • 제출 버튼을 누른 뒤 채점 결과를 확인한다.
Precondition	사용자가 로그인 상태여야 한다.
Post Condition	채점 결과와 현재 학습 진도는 새롭게 학습자의 정보와 함께 데이터베이스에 업데이트된다.
Assumptions	해당 없음

Table 3.9: Use Case of Learning (Lesson & Quiz)

<i>Use case name</i>	<i>Coding</i>
Actor	등록된 사용자
Description	코딩 과정은 회원으로 등록된 사용자가 시스템에서 제공하는 cat/dog classifier 에 대한 내용 및 머신 러닝에 필요한 기초 지식을 웹사이트를 통해 모두 습득하고 직접 이를 위한 코드를 실습해보는 프로세스이다.
Normal Course	<ul style="list-style-type: none"> • 모든 사용자는 로그인 후 학습 페이지가 나타난다. • 코드를 실습할 수 있는 코드 실습 페이지로 이동한다. • 코드를 작성할 때의 유의사항 및 skeleton 코드를 확인한 후 이에 알맞은 코드를 작성한다. • 제출 버튼을 누른 뒤 결과를 확인한다.
Precondition	사용자가 로그인 상태여야 한다.
Post Condition	해당 파이썬 코드를 실행한 후 나오는 결과를 사용자에게 출력한다.
Assumptions	해당 없음

Table 3.10: Use Case of Coding

3.2.2. Use Case Diagram

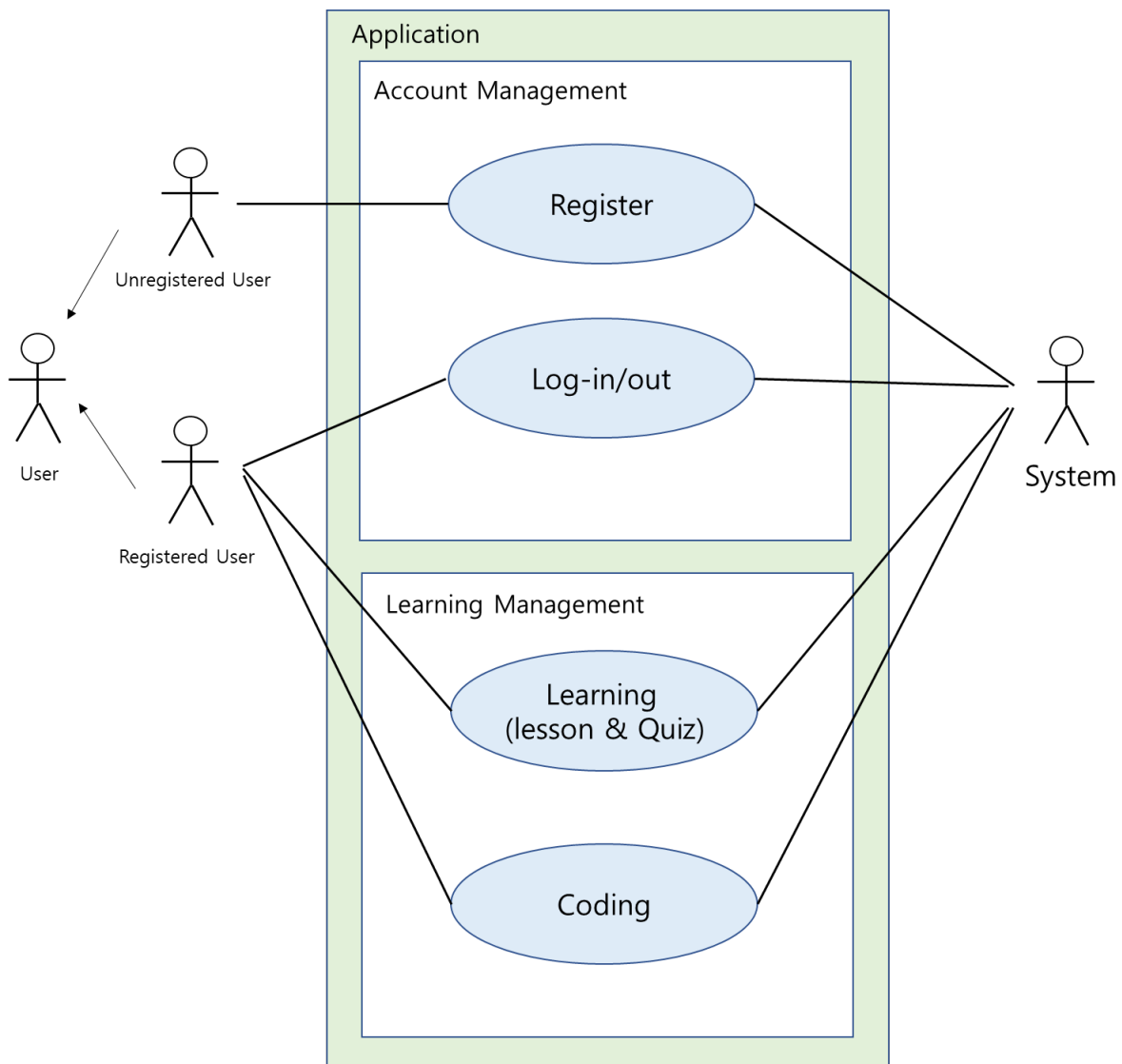


Figure 3.1: Use Case Diagram

3.2.3. Data Dictionary

Account			
Field	Key	Constraint	Description
id	PK	Not Null	User id
password		Not Null	User password
name		Not Null	User's name
progress		Not Null	User's learning progress
solved problem path			Problem's answer that user solved(File path)

Table 3.11: Account

두 테이블간 의존성이 없기 때문에 E-R Diagram은 생략한다.

Problem			
Field	Key	Constraint	Description
number	PK	Not Null	problem number
context		Not Null	problem context
test case path		Not Null	Test case to validate code (File path)
progress		Not Null	Answer expected (File path)

Table 3.12: Problem

3.2.4. Data Flow Diagram

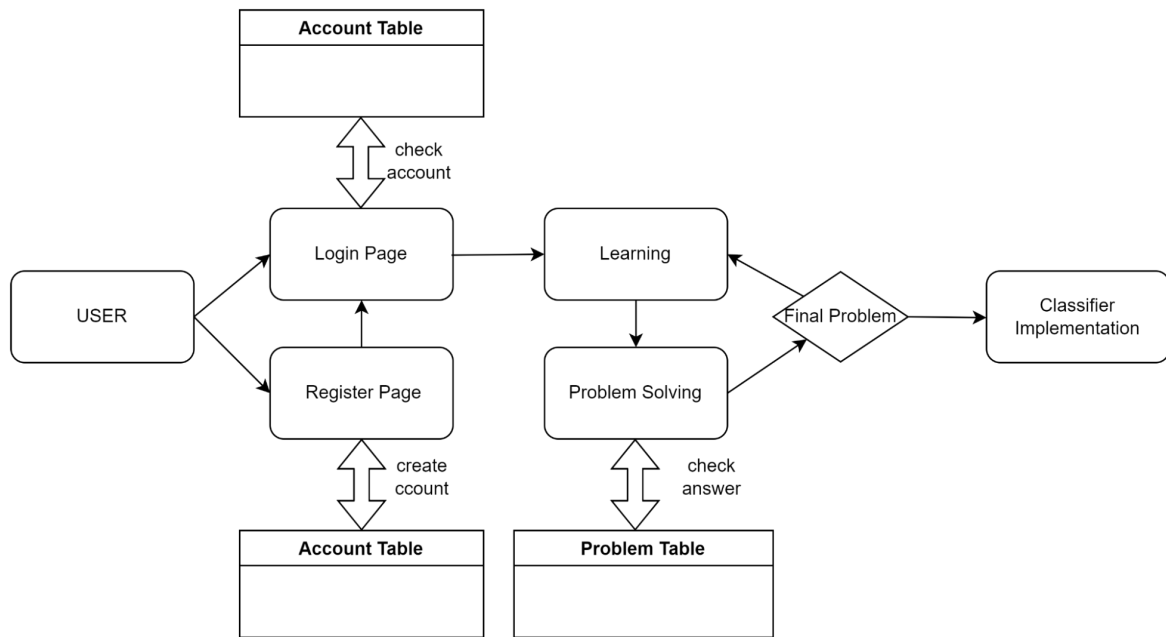


Figure 3.2: Data Flow Diagram

user - id(ID) / pw / 진행상황 / Code / 정답률 / problem - 문제 번호(ID), 답, test case 파일의 경로

3.3. Performance Requirements

아래는 본 시스템의 성능 요구사항에 관한 것이다. 예측에 기반한 내용이며 실제 구현시에 달라질 수 있다.

3.3.1. Static Numerical Requirement

- 시스템은 여러 명의 사용자가 동시에 이론을 학습하고 코드를 작성할 수 있도록 지원한다. 해당 사용자는 관리자와 학생이어야 한다.
- 시스템은 1.5GHz 이상의 싱글 프로세서, 2GB 이상의 메모리 환경의 데스크탑 윈도우 및 1.0GHz의 싱글 프로세서에서 부드럽게 동작해야 한다.
- 1Mbps 이상의 인터넷 연결 속도 환경을 요구한다. 운영체제는 윈도우 7 이상이며, 웹 어플리케이션을 지원하는 환경에서 정상적으로 동작한다.

3.3.2. Dynamic Numerical Requirement

- 시스템은 동시에 최소한 200명의 사용자 접속을 유지할 수 있어야 한다. 각 웹페이지는 동시에 최소한 200명의 사용자가 접속하여도 부드럽게 동작하여야 한다. 그리고 10000명까지의 사용자의 계정 정보를 관리할 수 있어야 한다.
- 사용자가 머신러닝 코드를 실행하고 결과를 확인하는데 걸리는 시간은 5분 이내로 완료되어야 한다. 계산된 결과는 5초 이내로 데이터베이스에 저장되고 이를 나타낼 수 있어야 한다.
- 사용자가 다음 목차로 이동할 시 해당 화면이 5초 이내로 표시되어야 한다.
- 사용자의 회원가입 요청을 5초 이내로 완료해야 한다. 새로운 회원의 계정 정보는 5초 이내로 데이터베이스에 저장되어야 한다.
- 사용자의 로그인/로그아웃 요청을 5초 이내로 완료해야 한다.
- 사용자의 문제 채점 결과를 5초 이내로 데이터베이스에 저장하고 이를 사용자에게 알려주어야 한다.

3.4. Logical Database Requirements

시스템은 Django에 기본적으로 설치되어 있는 SQLite 데이터베이스 서비스를 이용하여 데이터를 관리한다. 해당 데이터베이스를 통하여 시스템 사용자들의 계정 정보를 저장한다. 또한 각 사용자들의 학습 진행상황, 직접 짠 코드, 정답률 등의 교육내용을 데이터베이스에 저장한다. 사용자는 원한다면 자신이 풀었던 문제와 정답률, 자신이 직접 짠 코드 등을 확인할 수 있어야 한다.

3.5. Design Constraints

3.5.1. Physical design constraints

시스템의 목적은 이용자의 Python 코딩 교육이므로 데스크탑 또는 노트북 기기를 이용할 것을 권장하며, 모바일 기기 환경은 권장하지 않는다. 시스템은 SQLite 데이터베이스를 이용하여 필요한 데이터를 해당 데이터베이스에 저장할 수 있어야 한다.

3.5.2. Standards compliance

시스템은 웹 어플리케이션으로 Python Django를 이용해 개발되며, Django 코딩 표준(Python style) 및 HTML 표준을 따른다. 변수 이름은 lower 카멜 케이스를 따르며, 함수와 데이터베이스는 파스칼 케이스를 따른다.

3.6. Software System Characteristics

아래는 본 시스템의 몇 가지 비기능적 요구사항에 관련된 내용이다. 비기능적 요구사항은 제품 요구사항, 조직상의 요구사항, 외부적인 요구사항으로 나누어 기술하였다.

3.6.1. Product Requirements

제품 요구사항에서는 본 시스템의 실행 시간 중에 시스템이 어떻게 작동해야 하는지에 대해 설명한다. 본 시스템은 아래의 요구사항을 만족해야 한다.

Usability Requirements

본 시스템은 비전문가를 대상으로 하는 Python 교육 시스템으로 기초적인 Python 문법만을 알고 있는 사용자라고 하더라도 시스템에서 제공하는 교육 커리큘럼을 따라갈 수 있어야 한다. Python을 접해본 적이 있는 사용자라면 그들이 Python을 사용했던 것과 동일하거나 그와 비슷한 코딩 환경을 제공할 수 있어야 한다.

Performance Requirements

각각의 교육과정은 비전문가가 이해하고 따라갈 수 있는 수준이어야 하므로, 코드의 길이가 짧아야 하고 각 코드의 실행시간이 5초를 넘어서는 안 된다. 또한 마찬가지로 이유로 본 시스템은 사용자로 하여금 아주 정교한 프로그램이 아닌 아주 간단한 dog and cat classifier를 작성하도록 유도하며, 완성된 프로그램이 model training과 test를 실행함에 있어 5분을 넘어가지 않는 것을 목표로 한다.

Security Requirements

사용자들은 시스템을 이용하기 전에 올바른 절차를 통해 인증을 받고, 시스템에 접속하기 위한 계정을 생성해야 한다. 시스템 관리자를 제외한 일반 사용자는 시스템 관리자만큼의 권한을 가질 수 없고, 사용자

자신과 관련된 제한된 데이터만을 볼 수 있으며, 시스템 데이터베이스에 직접적으로 접근할 수 없어야 한다.

3.6.2. Organizational Requirements

조직상의 요구사항은 사용자 및 개발자가 속한 조직의 정책과 절차에 의해 발생하는 요구사항들에 관한 설명이다.

Environmental Requirements

데스크탑 또는 노트북 웹 브라우저 환경에서 접속을 권장한다.

Operational Requirements

본 시스템의 사용자들은 각 사용자가 소속된 대학교의 이메일 주소를 통해서 본인을 인증받는다. 시스템은 사용자가 원하는 때 언제라도 자유롭게 접속하여 수업을 진행할 수 있다.

3.6.3. External Requirements

외부적인 요구사항에서는 시스템의 외부적 요인으로 인해서 발생하는 다양한 요구사항들에 관해서 설명한다.

Safety / Security Requirements

시스템은 사용자들의 그 어떤 개인 정보도 외부의 시스템이 접근할 수 없도록 해야 한다.

Regulatory Requirements

본 시스템은 오로지 사용자의 교육상의 데이터 관리와 보안을 목적으로 사용자 정보를 필요로 함을 밝히고, 이에 필요한 최소한의 정보만을 얻을 수 있도록 한다. 법에 따라 사용자의 개인 정보를 침해해서는 안 된다. 시스템은 국제 개인 정보 보호 표준에 따라 개발되어야 한다.

3.7. Organizing the Specific Requirements

이 구간에서는 Unified Modeling Language(UML) 및 표 형식 기반의 그래픽 표기법을 사용하여 시스템 모델을 설명한다. 시스템 모델은 시스템, 서브 시스템 간의 관계를 설명한다

3.7.1. Context Model

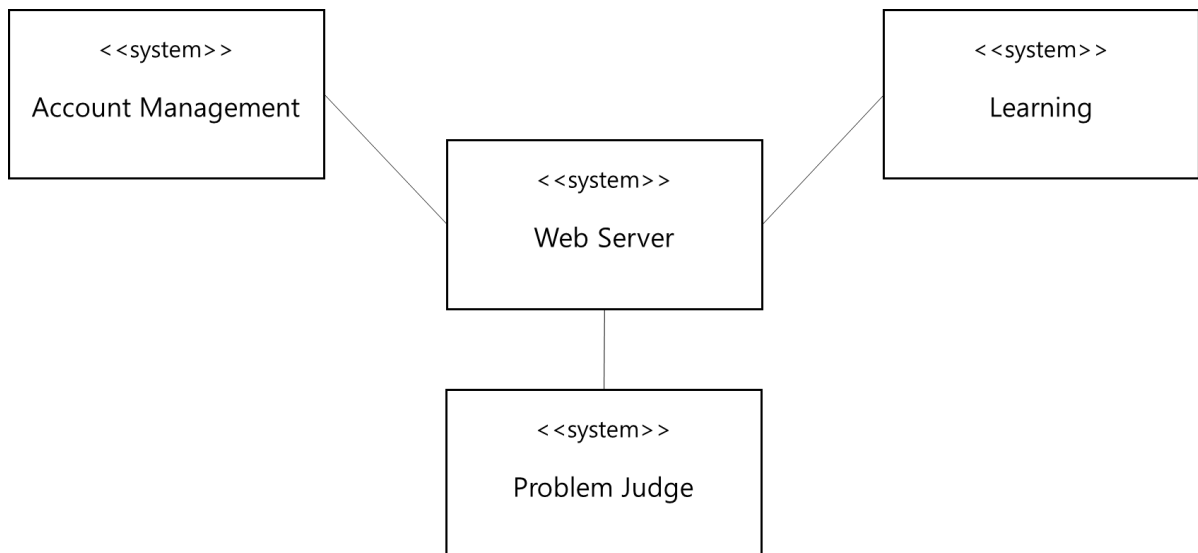


Figure 3.3: Context Model

3.7.2. Process Model

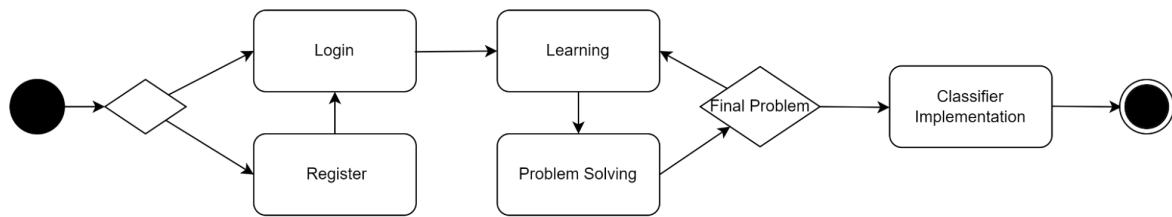


Figure 3.4: Overall Process Model

3.7.3. Interaction Model

Use Case Diagram 참조

3.8. System Architecture

이 절에서는 개발될 시스템 아키텍처에 대한 high-level 개요를 제시한다. 이는 각각의 서브 시스템과 그 구성요소를 밝히고 서브 시스템 간의 통신이 어떻게 이루어지는 지 명시한다.

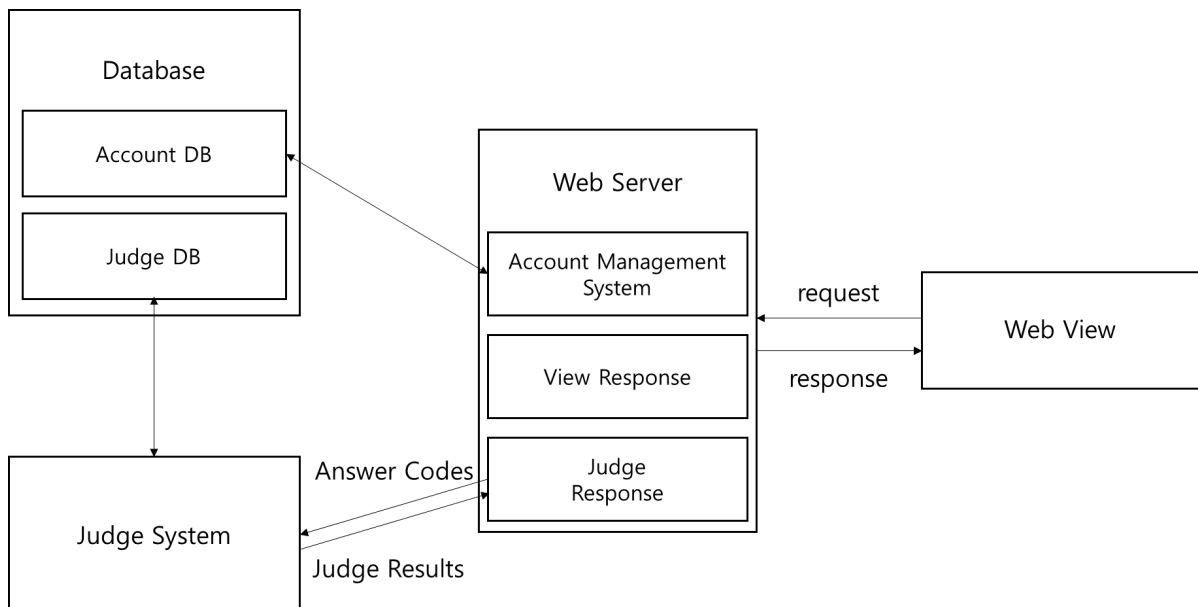


Figure 3.5: System Architecture

3.9. System Evolution

이 절에서는 시스템의 기초가 되는 기본 가정과 하드웨어 진화, 사용자 요구 변화 등으로 인한 모든 예상할 수 있는 변화를 설명한다. 또한 System Evolution 은 시스템 설계자가 시스템의 향후 변경을 제약하는 문제가 생길 수 있는 설계의 결정을 피하는데 도움이 될 수 있으며 유용하게 사용될 것이다.

3.9.1. Limitation and Assumption

본 시스템의 주요 목적은 학습자가 딥러닝에 대해 정확히 이해하는 것이 가장 중요하다. 여기서 가장 중요한 시스템은 학습자가 딥러닝 코드를 학습/작성 후 평가받는 시스템이다. 직접 학습자가 코드를 작성해야하는 특성 상 서비스 환경을 모바일 환경보다는 데스크탑 환경에 집중한다. 웹의 특성상 모바일 환경으로도 서비스를 제공할 수 있지만 주 대상은 데스크탑을 기준으로 잡는다.

3.9.2. Evolution of Hardware and Change of User Requirements

본 프로젝트의 큰 장점은 서비스 하나로 완결성이 있다는 점이다. 머신러닝 이론에 대한 교육이기 때문에 학습하는 방식이 변할 가능성은 있어도 학습내용이 변할 가능성은 현저히 낮다. 또한 외부환경에 의해 하드웨어나 소프트웨어 프레임워크 등이 변할 가능성 또한 낮다. 결국 사용자의 요구사항이 변경하는 사항은 사용성과 학습효율에 관한 내용으로 국한되고 본 시스템 또한 그 점에 초점을 맞춰 발전해 갈 것이다.

Supporting Information

4.1. Software Requirements Specification

소프트웨어 요구사항 명세서 IEEE 권장사항 (IEEE Recommend Practice for Software Requirements Specifications, IEEE-Std-830)에 따라 작성되었다

4.2. Document History

Document History			
Date	Version	Description	Writer
2022-04-09	V1.00	Product Perspective	김기윤
2022-04-09	V1.00	Product Functions and User Characteristics	강영호
2022-04-09	V1.00	Constraints, assumptions and dependencies	조민구
2022-04-09	V1.00	External Interface Requirements, and Functional & Performance Requirements	강동준
2022-04-09	V1.00	Logical Database Requirements, Design Constraints, and Software System Characteristics	이선종
2022-04-09	V1.00	Organizing the Specific Requirements, System Architecture, System Evolution	박상민
2022-04-09	V1.00	Latex Formatting	조민구