



전교생 Python 코딩 교육 시스템

소프트웨어 요구사항 명세서

2022.04.10.

Introduction to Software Engineering 41

TEAM 3

팀장 김지희

팀원 강신규

박상현

박상훈

엄지용

목차

1. Introduction.....	5
1.1. Purpose	5
1.2. Scope	6
1.3. Definitions, Acronyms, and Abbreviation	6
1.4. References	7
1.5. Overview	8
2. Overall Description.....	8
2.1. Product Perspective	8
2.1.1. System Interfaces	8
2.1.2. User Interfaces	9
2.1.3. Hardware Interfaces	10
2.1.4. Software Interfaces	10
2.1.5. Communications Interfaces	11
2.1.6. Memory Constraints	11
2.1.7. Operations	11
2.1.7.1. System administrator	11
2.1.7.2. User	11
2.2. Product Functions	12
2.2.1. Login & Register	12
2.2.2. Admin Page Access	13
2.2.3. Select Lecture Course	13
2.2.4. Select Lecture Class	13
2.2.5. Select Lecture Practice	14
2.2.6. Verifying User Code	14
2.3. User Characteristics	15
2.3.1. System Administrator	15
2.3.2. User	15
2.3.3. Vendor	15
2.4. Constraints	16
2.5. Assumptions and Dependencies	17
3. Specific Requirements	17
3.1. External Interface Requirements	17
3.1.1. User Interfaces	17
3.1.2. Hardware Interfaces	26

3.1.3. Software Interfaces	27
3.1.4. Communication Interfaces	28
3.2. Functional Requirements	30
3.2.1. Use Case	30
3.2.2. Use Case Diagram	33
3.2.3. Data Dictionary	33
3.2.4. Data Flow Diagram	34
3.3. Performance Requirements	34
3.3.1. Static numerical requirement	34
3.3.2. Dynamic numerical requirement	35
3.4. Logical Database Requirements	35
3.5. Design Constraints	36
3.6. Standards compliance	36
3.7. Software System Characteristics	36
3.7.1. Product Requirements	36
3.7.1.1. Usability Requirements	36
3.7.1.2. Performance Requirements	37
3.7.1.3. Security Requirements	37
3.7.2. Organizational Requirements	38
3.7.2.1. Environmental Requirements	38
3.7.2.2. Operational Requirement	38
3.7.3. External Requirements	38
3.7.3.1. Safety / Security Requirement	38
3.7.3.2. Regulatory Requirement	39
3.8. Organizing the Specific Requirements	39
3.8.1. Context Model	40
3.8.2. Process Model	40
3.8.3. Interaction Model	40
3.8.4. Behavior Model	41
3.8.4.1. Data Flow Diagram	41
3.8.4.2. Sequence Diagram	41
3.9. System Architecture	41
3.10. System Evolution	42
3.10.1. Limitation and Assumption	42
3.10.2. Evolutions of Hardware and Change of User Requirements	43
4. Supporting Information	43
4.1. Software Requirement Specification	43

4.2. Document History

44

그림 목차

[그림 1] Chrome 웹 브라우저 창 예시	18
[그림 2] 웹 브라우저 HTML 렌더링 예시	18
[그림 3] 로그인 및 등록 화면 예시	19
[그림 4] 로그인 오류 팝업 예시	20
[그림 5] 등록 화면 예시	21
[그림 6] 관리자 페이지 접근 예시	22
[그림 7] 관리자 페이지 예시	23
[그림 8] 교육 내용 열람 및 선택 예시	24
[그림 9] 교육자료 예시	25
[그림 10] 실습 환경 예시	26
[그림 11] 사용례 도식	33
[그림 12] SQLite Database 구조 및 상호작용	34
[그림 13] 맥락 모델	40
[그림 14] 프로세스 모델	40
[그림 15] 흐름 도식	41
[그림 16] 시스템 구조	42

표 목차

[표 1] 약어 정리	6
[표 2] 전문용어 정의	7
[표 3] 웹 서비스에 대한 사용자의 GUI 기반 접근	17
[표 4] 사용자 GUI (1) 로그인 및 등록 화면	19
[표 5] 사용자 GUI (2) 로그인 오류	20
[표 6] 사용자 GUI (3) 등록 오류	21
[표 7] 사용자 GUI (4) 관리자 페이지 접근	22
[표 8] 사용자 GUI (5) 관리자 페이지	23
[표 9] 사용자 GUI (6) 교육 선택	23
[표 10] 사용자 GUI (7) 교육 자료	24
[표 11] 사용자 GUI (8) 실습 환경	25
[표 12] 브라우저 사용에 필요한 하드웨어 인터페이스	26

[표 13]	서비스 서버 소프트웨어 인터페이스.....	27
[표 14]	서비스 사용자 소프트웨어 인터페이스.....	28
[표 15]	로그인 HTTP 요청 형태	28
[표 16]	계정 등록 HTTP 요청 형태	29
[표 17]	실습 코드 실행 HTTP 요청 형태	29
[표 18]	일반적인 페이지 렌더링 자원 HTTP 요청 형태	29
[표 19]	사용례 (1) 로그인	30
[표 20]	사용례 (2) 계정 등록	31
[표 21]	사용례 (3) 학습 자료 선택	31
[표 22]	사용례 (4) 관리자 페이지 접근	32
[표 23]	사용례 (5) 실습 코드 실행 요청	32
[표 24]	데이터베이스 Table 'User' 구조	33
[표 25]	데이터베이스 Table 'Class' 구조	33
[표 26]	데이터베이스 Table 'Complete_Log' 구조	34
[표 27]	문서 이력	44

1. Introduction

1.1. Purpose

본 문서는 "Python programming education for students" 서비스 (이하 '본 서비스') 제공을 위한 소프트웨어 요구 명세서이다. 본 서비스는 2022년 1학기 성균관대학교 소프트웨어공학개론 41분반 3팀(이하 '개발팀')에 의해 고안되고 개발된다. 본 요구 명세서에서 소프트웨어 시스템의 요구 사항을 요약 및 분석하였으며, 이에 근거하여 본 서비스의 시스템을 개발한다.

본 문서는 개발팀이 열람하는 것을 상정하여 작성되었다. 예외적으로, 교수, 조교를 포함한 성균관대학교 소프트웨어공학개론 수강자 또한 학습 및 교육의 용도로 본 문서를 열람할 수 있다. 본 문서를 재배포 및 수정하는 것은 자유이나, 상업적 용도로 활용 시 반드시 개발팀의 허가를 얻어야 한다.

본 문서는 본 서비스의 개요와 요구 명세를 제시하기 위해 작성되었다. 본 서비스는 웹 어플리케이션 기반으로 구현되어, Python 언어에 대한 교육 환경을 제공하기 위해 개발되었다. 서비스 내에서 Python 언어의 기초적인 문법에 대한 내용부터 시작하여, 최종적으로 사용자가 실시간 채팅 어플리케이션을 구현할 수 있도록 교육한다. 이 과정에서 웹 서버 및 사용자의 코드를 채점하는 시스템 등

이 개발되며 세부사항은 이하 문서에 명시한다.

1.2. Scope

본 서비스는 학생들에게 양질의 Python 언어 교육 환경을 제공하기 위해 고안되었다. 코드 작성 실력을 키우기 위한 매체 및 작성한 코드의 동작을 평가하는 기능을 제공한다. 또한 접근성이 뛰어난 웹 환경으로 제공하여 적용 가능한 대상을 늘린다. 이를 제공하기 위해 JavaScript 기반의 웹 서버를 구축하며, 사용자가 입력한 코드를 테스트하고 결과를 전달하기 위해 Docker 기술을 활용한다.

본 서비스를 통해 사용자는 Python의 기본지식을 익힐 뿐 아니라, 최종적으로 실시간 채팅 어플리케이션을 구현할 수 있다. 본 서비스의 공급자는 확장성을 고려한 설계를 바탕으로, 추후 다른 어플리케이션 구현 혹은 다른 언어에 대한 교육 플랫폼으로 활용할 수 있다.

1.3. Definitions, Acronyms, and Abbreviation

이하의 표는 본 문서에서 활용된 약어의 정의를 나타낸다.

[표 1] 약어 정리

용어	설명
RAM	Random Access memory
WDDM	Windows Display Driver Model
NIC	Network Interface Controller
GUI	Graphic User Interface
HTTP	HyperText Transfer Protocol
DNS	Domain Name Service
TCP	Transmission Control Protocol
SSH	Secure SHell
MD	MarkDown
HTML	HyperText Markup Language
CSS	Cascading Style Sheet

DOM	Document Object Model
URL	Uniform Resource Locator

이하의 표는 본 문서에서 활용된 전문용어의 정의를 나타낸다.

[표 2] 전문용어 정의

용어	설명
Docker	Linux Container Management Program
JavaScript	ECMA Script based JIT Interpreter Language
SQLite	Light-weight Database Management System
NodeJS	Network Application Development Platform
Express	NodeJS based Standard Web Framework
DirectX	Microsoft Based Multimedia Graphical API
OAuth	Industry-standard protocol for authorization

1.4. References

- IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications, In IEEEExplore Digital Library
<http://ieeexplore.ieee.org/Xplore/guesthome.jsp>
- Team 1. “Software Requirement Specification”. SKKU, Last Modified: Jun. 21, 2020.
https://github.com/skkuse/2020spring_01class_team1
- IETF RFC 2616, HyperText Transfer Protocol 1.1.
<https://datatracker.ietf.org/doc/html/rfc2616>
- ECMA Script Standard, ECMA-262, Jun 2021.
<https://www.ecma-international.org/publications-and-standards/standards/ecma-262/>
- HTML & CSS Standard, W3C.
<https://www.w3.org/standards/webdesign/htmlcss>
- Docker: Container Management Program
<https://www.docker.com/>

■ OAuth 2.0: Authorization and Authentication Protocol

<https://oauth.net/2/>

1.5. Overview

본 소프트웨어 요구 명세서의 이하 부분은 세 부분(Chapter 2.~4.)과 부록으로 구성되어 있다. Chapter 2.에서는 제품의 관점과 제품의 기능, 사용자 특징 및 제약조건에 대해 서술한다. 제품의 관점 항목은 하드웨어, 소프트웨어, 통신, 그리고 사용자의 인터페이스에 대해 서술한다. 제품의 기능 항목은 본 서비스가 제공하는 기능에 대해 서술한다. Chapter 3.에서는 세부적인 요구 명세가 기술된다. 각 인터페이스의 형태 및 변수들의 제약조건에 대해 서술한다. 또한 요구조건의 형태를 모델로서 나타낸다. Chapter 4.에서는 변경 이력 등 본 문서에 대한 부가적인 정보를 서술한다.

2. Overall Description

2.1. Product Perspective

본 제품은 Python 프로그래밍을 배우고자 하는 학생들을 위해 개발되었다. Python을 사용할 환경은 갖춰져 있지만 Python 프로그래밍에 대한 기술이 없는 사람들을 주 대상으로 한다. 교육 대상이 되는 Python 인터프리터 버전은 3.10.X 버전이다. 사용자는 서비스를 통해 Python 프로그래밍의 기초를 학습하고, 서비스에서 제공하는 예제를 풀이하면서 학습한 내용을 검증할 수 있다. 학습한 내용을 따라가면서 최종적으로 실시간 채팅을 구현할 수 있다. 본 제품이 가정하는 동시 사용자는 100명 이내이며 그 이상의 접근에 대해서는 거부하거나 대기를 요청한다.

2.1.1. System Interfaces

사용자는 Chrome 웹 브라우저를 이용해 본 서비스에 접근한다. 이 때 Chrome 웹 브라우저의 버전은 100.X 이상을 대상으로 한다. 웹 브라우저는 사용자에게 로그인 기능을 제공함과 동시에, 로그인하지 않은 계정의 서비스 이용을

제한한다. 이는 서버가 지나치게 많은 코드를 검사하거나 무차별적인 서비스 거부 공격(Denial of Service)을 당하는 상황을 방지하기 위함이다. 로그인에 사용되는 정보는 SQLite 데이터베이스를 통해 관리한다.

웹 서비스를 제공하기 위한 환경은 NodeJS 프레임워크를 활용하여 구현한다. NodeJS의 sqlite3 플러그인을 활용하여 본 서비스와 데이터베이스를 연결한다. SQLite는 NodeJS와의 연결이 용이하고, 또한 가벼운 구현으로 소규모 서비스를 제공하기 적합하다고 판단하여 선정했다. NodeJS 기반으로 웹 사이트를 작성하기 위해 Express 엔진을 활용한다

코드를 테스트하기 위한 환경은 Judge0 API와 Docker 프로그램을 이용한다. 사용자가 웹 서비스를 통해 전달한 내용을 Judge0 혹은 Docker의 샌드박스 환경에서 실험한다. 실험하는 바는 첫째, 코드가 에러를 내지 않고 동작하는지. 둘째, 코드가 의도한 동작을 수행하는 지이다. 두번째 항목을 테스트하기 위해서, 미리 코드에서 기대하는 동작들에 대한 테스트 케이스를 작성해 둔다. 두 항목의 결과 값을 웹에 표기하여 사용자에게 전달한다.

2.1.2. User Interfaces

사용자는 Chrome 웹 브라우저를 이용해 HTML, CSS, JavaScript로 작성된 내용을 읽고 상호작용할 수 있다. 웹 브라우저 기반의 구현이라 어떠한 환경이든 접근 가능하지만, 원활한 사용을 위해서는 데스크탑 PC와 모니터 해상도 1920 * 1080 이상을 권장한다. 사용자가 수행할 수 있는 주요 기능은 로그인(회원가입), 학습자료 열람, 코드 테스트의 세 가지이다. 또한 본 서비스는 이 기능들을 사용자에게 원활하게 제공하기 위한 웹 디자인적 요소들을 포함한다.

관리자는 테스트 케이스 및 새로운 학습 자료를 추가하기 위해서 직접 서비스가 제공되는 서버의 데이터를 수정해야 한다. 본 제품의 관리자는 프론트 엔드 수준의 사후 관리를 지원할 필요가 없으므로 별도 구현하지 않는다. 서비스 서버는 SSH 프로토콜 기반의 공개키 암호화를 통해 접근할 수 있으며 관리자는 NodeJS에서 사용하는 Express Template 및 데이터를 수정하여 사용자에게 제공되는 웹 서비스를 수정할 수 있다.

2.1.3. Hardware Interfaces

사용자의 측면에서, 웹 브라우저 외의 하드웨어 의존성은 존재하지 않는다. 본 제품은 Chrome 웹 브라우저를 Windows 환경에서 사용한다고 가정하고 개발한다. Chrome 브라우저의 시스템 요구사항은 Windows 7 이상의 운영체제, 및 일정 수준 이상의 프로세서이다. Windows 7의 최소사항과 조합하면 다음과 같다.

- ① 1GHz 이상 32비트(x86) 또는 64비트(x64) 프로세서 (Core I5 이상)
- ② 1GB RAM(32비트) 또는 2GB RAM(64비트)
- ③ 16GB 사용 가능한 하드 디스크 공간(32비트) 또는 20GB(64비트)
- ④ WDDM 1.0 이상 드라이버와 DirectX 9 그래픽 디바이스

또한 인터넷에 접근하기 위한 NIC 및 웹 브라우저의 GUI와 소통하기 위한 마우스 등의 HID를 포함한다.

관리자의 측면에서, 원활하게 NodeJS 서버와 Docker를 제공하기 위한 하드웨어 요구조건은 다음과 같다. 원활한 서버 운영을 위해 서술된 것보다 더 고성능의 환경을 갖추기를 권장한다.

- ① Ubuntu 20.04 운영체제 및 NIC를 제공하는 메인보드
- ② 4GB 이상의 RAM 환경
- ③ 25GB 이상의 하드 드라이브

2.1.4. Software Interfaces

상술했던 바와 같이 사용자는 Windows 7 이상의 운영체제에서 최신 안정화 Chrome 버전을 사용함을 기준으로 한다. Windows 10 기준 Chrome 100 버전 이상을 사용하는 환경을 가정한다. 또한 필수 사항은 아니지만 원활한 교육을 위해 웹 공간 뿐 아닌 각 사용자의 기기에서도 Python 인터프리터 버전 3.10.X 을 사용하는 것을 권장한다.

관리자는 Ubuntu 20.04 이상의 운영체제에서 NodeJS를 사용하여 웹 서버를 구

축한다. 웹 서버에서 사용자 정보를 다루기 위해 SQLite 데이터베이스를 활용한다. 사용자가 입력한 코드를 실행 및 검증하는 환경을 마련하기 위해 Docker 와 Judge0 API를 활용한다.

2.1.5. Communications Interfaces

웹 페이지는 HTTP 프로토콜을 기반으로 사용자의 정보를 서버로 전달하며, 서버 또한 사용자에게 노출될 정보를 HTTP 프로토콜을 기반으로 전달한다. TLS 1.2 이상을 지원하기 위한 DNS와 Root Certificate는 본 제품에 포함되어 있지 아니며 추후 관리자에 의해 추가될 수 있다.

2.1.6. Memory Constraints

서버의 메모리는 4GB RAM을 최소사양으로 한다. 그러나 안정적인 서비스 제공을 위해 16GB 혹은 32GB 이상의 메모리 공간을 권장한다. 본 서버가 지원하는 최대 인원인 100명에 대한 Docker Instance를 생성할 수 있어야 한다.

사용자의 메모리는 1GB RAM을 최소사양으로 한다. 그러나 안정적인 서비스 이용을 위해 Windows 10 이상의 운영체제 요구사항을 충족하고 4GB 이상의 메모리를 갖출 것을 권장한다. 이는 본 페이지가 제공하는 GUI 처리에 의한 성능 저하가 서비스 품질에 영향을 주지 않게 하기 위함이다.

2.1.7. Operations

2.1.7.1. System administrator

- 관리자 페이지
 - ✓ 관리자 계정일 경우 접근할 수 있다.
 - ✓ 관리자 간 공유할 내용 등을 기록할 때 활용한다.
 - ✓ 서버에 접근할 수 있는 SSH Key를 제공한다.

2.1.7.2. User

- 로그인
 - ✓ 사용자는 본인의 비밀번호 및 계정으로 서비스에 로그인할 수 있다.
- 등록

- ✓ 사용자는 비밀번호 및 계정을 서비스에 등록할 수 있다.
- ✓ 등록한 계정을 추후 로그인에 활용할 수 있다.
- ✓ OAuth 기반으로 타 서비스 계정을 이용해 로그인할 수 있다.
- 교육 대분류 선택
 - ✓ 사용자는 교육 및 코드 검증할 내용의 대분류를 선택할 수 있다.
 - ✓ 초기 버전 대분류는 ‘기초 지식’ , ‘실시간 통신’ 을 포함한다.
- 교육 소분류 선택
 - ✓ 대분류 하위의 소분류를 선택해 학습 자료를 열람할 수 있다.
- 실습하기
 - ✓ 각 학습 자료에 맞춰 실습 문제가 제공된다.
 - ✓ 실습할 수 없는 교육자료에 대해서는 실습이 제공되지 않을 수 있다.
 - ✓ 선택 시 코드를 입력할 수 있는 창이 등장한다.
- 검사하기
 - ✓ 사용자가 입력한 코드를 실행하고 검사한다.
 - ✓ 검사 결과 내용을 출력한다.

2.2. Product Functions

2.2.1. Login & Register

사용자가 웹 브라우저를 통해 서비스에 접근하면, 서비스는 사용자에게 로그인 혹은 계정 등록을 할 수 있는 인터페이스를 제공한다. 등록된 계정이 있을 경우 계정을 활용해 서비스를 이용할 수 있다. 등록된 계정이 없을 경우 신규 계정을 생성할 수 있다. OAuth 프로토콜을 기반으로 한 타 서비스의 계정 정보 공유 시스템도 제공할 수 있다. 등록 과정에서 사용하는 계정명과 비밀번호는 함께 Hash하여 서버의 데이터베이스에 저장한다. 로그인 시 해시값을 기반으로 성공 여부를 검증한다. 계정 정보가 잘못된 경우 재입력을 요청한다.

로그인하여 서비스에 접근할 경우 세션이 유지되며 세션의 수 및 세션의 유효성 여부를 이용해 서비스 품질을 향상할 수 있다. 본 제품이 지원하는 세션은 최대 100개이며 동일 세션에서 여러 코드 검증을 요청할 수 없다. 세션의 수를

초과한 요청에 대해서는 서버 상태에 대해 공지한 뒤 로그인을 허락하지 않는다.

2.2.2. Admin Page Access

로그인에 성공한 계정은 사용자 계정 및 관리자 계정으로 구분된다. 관리자 계정의 경우 일반적인 계정 등록을 통해 얻을 수 없으며 다른 관리자에 의해 권한을 부여 받아야한다. 공급자는 본 제품을 사용할 때, 최초의 관리자 계정과 SSH 키를 같이 제공받는다. 이 키를 활용하여 서버에 접근한 후 다른 관리자를 선정할 수 있다.

관리자 계정은 로그인 후 관리자 페이지로 접근할 수 있는 인터페이스를 제공한다. 관리자 페이지로 접근할 때에 계정의 관리자 여부를 다시 한번 검증한다. 관리자임이 확인된 경우 관리자 간 정보를 공유하기 위한 페이지 및 서버에 접근하기 위한 SSH 키 값을 제공받는다. 관리자는 서버에 직접 접근하여 다른 관리자를 선임하거나, 웹 서비스가 제공하는 내용들을 수정하거나, 서버 환경 자체를 수정할 수 있다.

2.2.3. Select Lecture Course

일반 사용자가 로그인에 성공한 경우, 교육 내용을 선택할 수 있는 인터페이스를 제공한다. 인터페이스는 대분류와 소분류로 구성되어 있으며 대분류로 수강할 교육 과정을 선택하고 소분류로 교육 차수를 선택하는 방식으로 구성된다. 수강 및 실습이 완료된 경우 계정 정보에 각 항목의 수강 여부를 기록한다. 이를 바탕으로 수강한 과목과 수강하지 않은 과목을 그래픽적으로 구분할 수 있다.

대분류의 이름을 선택하면 토글 형태로 소분류가 열리며, 소분류 내의 각 항목들을 클릭하여 교육 내용을 열람할 수 있다.

2.2.4. Select Lecture Class

수강할 교육 차수를 선택하면 교육 자료가 제공되며 교육 자료는 동영상, 그림, 글 등 다양한 멀티미디어를 기반으로 구성될 수 있다. 실습이 제공되는 교육의 경우 반드시 예시 코드가 포함되어야 하며 모든 교육 자료는 전공자에 의해 검토된 후 제공되어야 한다.

교육 자료는 MD 파일의 형태로 서버에 저장되고 이 파일을 해석하여 사용자에게 전달한다. MD 파일은 서버의 NodeJS 에서부터 markdown-viewer 플러그인을 기반으로 해석되고, 사용자에게 전달된다.

교육 자료는 1페이지 이상으로 구성되지 않는 것을 원칙으로 하며 부득이하게 구분해야 할 경우 교육 차수 자체를 구분하는 것을 권장한다. 이 경우 실습은 다음 차수에서 동시에 수행할 수 있다. 교육 자료 MD 파일을 제공하는 인터페이스 이후 ‘다음 차수’ 버튼이 활성화되어 사용자는 끊임 없이 학습을 이어 나갈 수 있다. 실습이 없는 경우 ‘완료’ 버튼을 눌러 해당 교육의 수강 여부를 계정에 기록할 수 있다.

2.2.5. Select Lecture Practice

실습이 제공되는 차수의 경우 ‘다음 차수’ 버튼 좌측에 ‘완료’ 대신 ‘실습하기’ 버튼이 활성화되며, 실습하기 버튼을 누를 경우 별도의 창에서 문제 및 IDE를 제공받는다. IDE는 NodeJS의 gh-highlight 등 문법 강조 플러그인을 활용하여 사용자에게 제공되는 HTML TextArea 컴포넌트의 형태이다. IDE의 우측에는 사용자가 실습을 위해 수행할 과제가 작성되어 있으며 사용자는 과제 내용을 만족하는 Python 코드를 IDE에 작성할 수 있다.

2.2.6. Verifying User Code

‘실행’ 버튼을 눌러 제공하는 코드를 서버가 테스트할 수 있으며 그 결과값은 IDE 하단에 구현된 Console 인터페이스를 통해 제출된다. 사용자가 제출한 코드가 정상적으로 동작하고 모든 테스트 케이스를 통과한 경우, 해당 사용자의 계정은 자동적으로 차수를 수강한 것으로 기록된다. 이후 ‘다음 차수’ 버튼이 활성화되며 해당 버튼을 눌러 사용자는 다음 실습으로 이어갈 수 있다.

서버는 사용자의 코드를 HTTP 형태로 수신하고 제공받은 코드를 일정 Timeout동안 수행한다. Timeout을 벗어난 경우 코드가 너무 오래 실행된다는 결과를 반환한다. 수행은 Judge0 및 Docker 환경을 기반으로 이루어질 수 있다.

2.3. User Characteristics

2.3.1. System Administrator

시스템 관리자는 본 서비스가 제공하는 Python 프로그래밍 교육자료와 실습 자료를 추가하고 서버 문제에 대응하는 사람을 뜻한다. 관리자는 SSH 키를 통해 서버에 직접적으로 자료를 추가하고 실습에 활용될 Test Case를 선정할 수 있다. 시스템 관리자는 서버 관리 및 NodeJS 프레임워크 기반 웹 프론트엔드 구축에 지식이 있는 사람이어야 한다. 즉 HTML, JavaScript, CSS 등의 컴포넌트를 적절히 활용하고 각 컴포넌트가 어떻게 서버와의 HTTP 통신에 사용되는지 알아야 한다. 또한 서버에 발생한 문제를 로그를 기반으로 파악하고 해결할 수 있어야 한다. 또한 시스템 관리자는 sqlite 데이터베이스 쿼리문에 대한 기본적인 지식을 바탕으로 다른 관리자를 설정하거나 사용자가 문의하는 오류 등을 해결할 수 있어야 한다.

관리 측면 외에도 제공되는 서비스에 대한 이해도가 필요하기 때문에 Windows 환경의 Python 프로그래밍 기술을 가진 사람을 관리자로 선임할 것을 권장한다.

2.3.2. User

사용자는 웹 서비스에 접근하여 교육 및 실습 환경을 이용할 사람을 뜻한다. 사용자는 한국어로 제공된 문서를 읽고 Python 프로그래밍을 학습할 수 있다. 본 서비스는 사용자가 Python 설치 등 기본적인 절차는 따로 교육하지 않으나 원활한 실습 및 학습 환경을 위해 환경 설정은 사용자가 수행할 역량이 있다고 가정한다. 사용자는 또한 Windows의 Chrome 브라우저를 활용하는 데에 지장이 없으며 각 GUI의 반응을 확인하고 새로운 상호작용을 시도할 수 있어야 한다. 즉, 본 서비스는 한글 문해력, 프로그래밍 언어 기초 지식, 그리고 웹 서비스 사용에 지장이 없는 사용자를 대상으로 한다.

2.3.3. Vendor

공급자는 시스템 관리자를 고용하고 본 서비스를 운영하는 비용을 제공하는

사람을 뜻한다. 공급자는 본 서비스를 제공하면서 홍보 혹은 사회 권익 증진 등 부가 가치를 창출할 수 있다. 공급자는 서버 관리 및 프로그래밍에 대한 지식이 전혀 필요하지 않으며 관련된 내용은 시스템 관리자에게 일임할 수 있다. 단, 원활한 소통 및 시스템 홍보 등을 위해 관련 지식을 가지고 있는 것을 권장한다. 최종적으로 공급자는 본 서비스를 통해 창출되는 모든 가치에 대한 권한을 가진다.

2.4. Constraints

본 시스템은 상기한 요구 조건과 후술할 세부 요소들을 바탕으로 고안되고 구현되어야 한다. 본 문서에 명시되지 않은 세부 조건들은 개발자의 재량에 의해 구현되고 설계될 수 있다. 그러나 그 과정에서 다음과 같은 원칙은 지켜져야 한다.

- ✓ 외부 API를 사용하거나 본 시스템에 내장할 경우에는 보안성 및 성능이 검증된 API를 사용할 것
- ✓ 불가피한 경우를 제외하고 모든 교육자료에는 실습이 포함될 것
- ✓ 본 제품이 가진 라이선스(Apache2 License) 와 호환되지 않는 라이선스를 사용하지 않을 것
- ✓ 본 제품의 전체적인 라이선스를 변경하지 않을 것
- ✓ 시스템 전체의 성능을 저하시키지 않을 수 있도록 인터페이스를 설계할 것
- ✓ 서버는 사용자에게서 주어지는 모든 입력값을 신뢰하지 않을 것
- ✓ 운영체제, 브라우저 등의 요구 조건과 상충하는 외부 자원을 활용하지 말 것
- ✓ 제품의 사용자는 브라우저 외 필수적으로 별도의 제품을 설치할 필요가 없을 것
- ✓ 사용자 비밀번호는 해시값을 통해 비식별화되어 저장되고 활용될 것
- ✓ 서비스 운영에 불필요한 사용자 정보를 요구하지 않을 것
- ✓ 현재 가용 사용자 수는 100명으로 설계되었으나, 하드웨어의 개선만으로 가용 인원을 늘릴 수 있도록 확장 가능한 설계를 할 것.

- ✓ 시스템 관리자에게 지나치게 많은 업무를 넘기지 말 것
- ✓ 시스템 관리자가 특정 기능을 할 것이 자명하다면 셸 스크립트 혹은 명령 줄 도구를 서버에 포함하여 배포할 것

2.5. Assumptions and Dependencies

본 문서에서 서술하는 제품은 사용자의 입장에서 요구 조건을 만족하는 환경에서 접근한다고 가정한다. 즉 운영체제, 브라우저, 하드웨어 한계점 등을 충족했다고 가정한다. 따라서 본 문서에서 상술한 요구조건을 만족하지 못한 환경에 대해서는 정상적인 동작을 보장할 수 없다.

또한 서버의 Ubuntu 환경과 NodeJS, Sqlite, Docker, Judge0등 외부 API가 서로 충돌하지 않는다고 가정한다. 이 때, 각 API 제품은 서로 충돌하지 않도록 설계되어 배포되었기 때문에 운영체제와의 호환성만을 고려한다. 각 제품의 버전은 Ubuntu 20.04 환경에서 제공하는 최신 버전에 대해 모든 Backward Compatibility를 제공해야 한다. 운영체제를 제외한 각 API의 최신 버전 기준은 2022년 3월을 기준으로 한다.

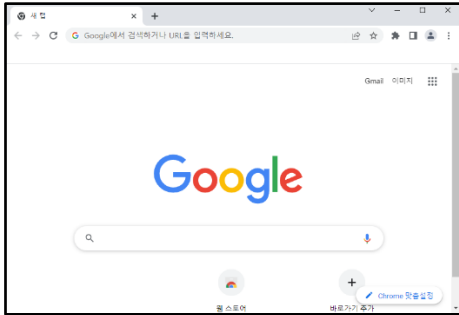
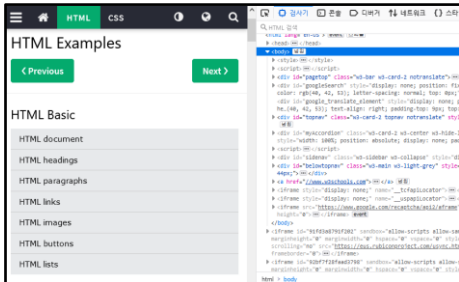
3. Specific Requirements

3.1. External Interface Requirements

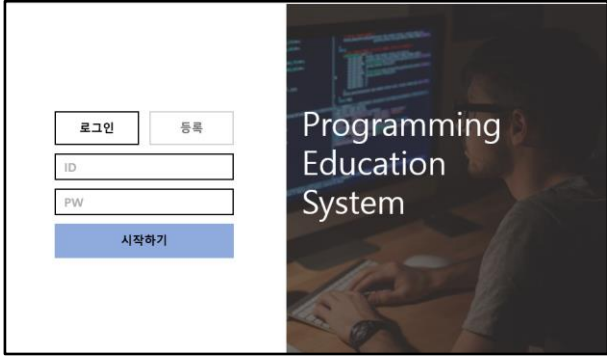
3.1.1. User Interfaces

[표 3] 웹 서비스에 대한 사용자의 GUI 기반 접근

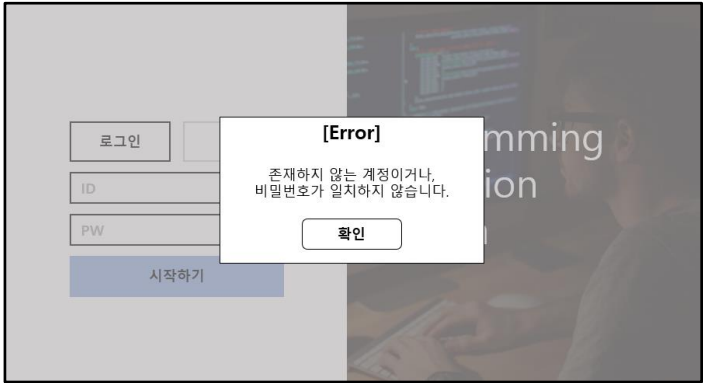
이름	웹 서비스에 대한 사용자의 GUI 기반 접근
목적 및 설명	사용자가 웹 서비스와 상호작용할 수 있는 가장 기본적인 인터페이스이다. 인터페이스는 브라우저의 기본 구현을 따르며 서버로부터 Html, CSS, JavaScript를 받아 사용자의 화면에 전달한다. 사용자는 브라우저 창을 통해 제공된 정보를 바탕으로 서비스에 요청을 보내거나 서비스가 제공하는 자료를 열람할 수 있다.
입출력 형태	마우스 클릭, 호버링, 키보드 입력 등 HID 입력 브라우저 JS 처리 및 HTML 링크 연산
범위/정확도/ 오차범위	W3C 표준에 의한 CSS 처리 EMCAScript 표준에 의한 JavaScript 처리

이름	웹 서비스에 대한 사용자의 GUI 기반 접근
	WHATWG 표준에 의한 Html 처리 *표준을 지키지 않는 기술에 대해서는 브라우저 종류와 버전에 따라 동작이 다를 수 있다.
단위	Html DOM 컴포넌트와의 상호작용 JavaScript 이벤트와의 상호작용
시간/ 속도	서버 및 통신망 의존성: 페이지 로드 네트워크 지연시간 사용자 컴퓨터 성능 의존성: 페이지 렌더링 지연시간
다른 입출력 간 연관성	브라우저 확장 기능에 의한 페이지 상태 변화 브라우저 자체 기능에 의한 페이지 차단 및 Redirecting
화면의 형태	사용자의 브라우저의 기본 설정 크롬 브라우저 기준 아래 그림과 같음 DOM 및 JS 표준에 맞추어 렌더링
창의 형태	브라우저의 기본 설정에 따름 크롬 브라우저 기준 아래 그림과 같음 사용자 확장기능 및 테마에 맞추어 변화 [그림 1] Chrome 웹 브라우저 창 예시 
데이터 형태	브라우저에서 인식하는 JS 이벤트 브라우저에서 인식하는 HTML 하이퍼링크 접근 [그림 2] 웹 브라우저 HTML 렌더링 예시 
종료 동작	브라우저 종료 혹은 페이지 이동

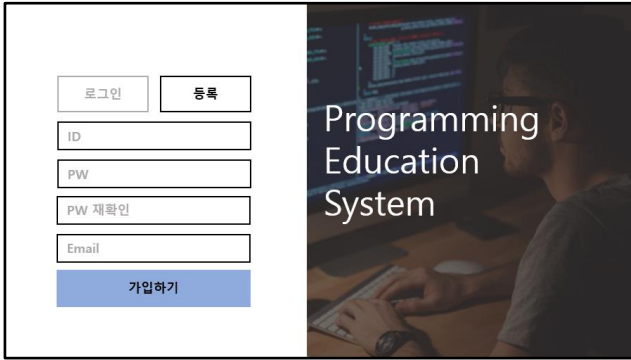
[표 4] 사용자 GUI (1) 로그인 및 등록 화면

이름	사용자 GUI 1. 로그인 및 등록 화면
목적 및 설명	사용자가 IP 주소 혹은 도메인 이름을 이용하여 본 서비스에 접근할 때, 사용자에게 처음 제공되는 화면이다. 본 화면에서는 사용자가 로그인 및 등록을 수행할 수 있는 환경을 제공한다.
입출력 형태	<p>사용자는 버튼 클릭으로 등록/로그인 탭을 선택할 수 있다. 이 동작에 따라 서버에 제출 가능한 HTTP POST 요청의 종류가 바뀐다. 본 인터페이스는 서버에 최초 접근할 때 수신한 Html, Css, JavaScript를 입력으로 형성된다.</p> <p>등록: 사용자 계정, 사용자 비밀번호, 사용자 이메일을 서버에 전송</p> <p>로그인: 사용자 계정, 사용자 비밀번호를 서버에 전송</p> <p>각 항목은 탭에 있는 '로그인' 혹은 '등록' 버튼을 눌러 전송할 수 있다.</p> <p>*이외의 동작은 상기 [표 3]의 웹 서비스 GUI에 상속됨</p>
범위/정확도/오차범위	표준을 지키지 않는 HTML 동작은 브라우저 버전에 따라 다를 수 있음 HTTP 패킷 전달은 TCP 기반 통신의 표준을 따른다.
단위	HTML DOM 요소와의 상호작용 (탭 전환 및 HTTP 요청 제출) HTTP POST 요청 통신
시간/ 속도	인터페이스 표기: 상기 [표 3]의 웹 서비스 GUI에 상속됨 HTTP POST 요청 처리: 서버 처리 시간에 상속됨
다른 입출력 간 연관성	*상기 [표 3]의 웹 서비스 GUI에 상속됨
화면의 형태	<p>아래의 화면과 같은 형태를 띤다.</p> <p>*세부적인 사항은 상기 [표 3]의 웹 서비스 GUI에 상속됨</p> <p>[그림 3] 로그인 및 등록 화면 예시</p> 
창의 형태	*상기 [표 3]의 웹 서비스 GUI에 상속됨
데이터 형태	계정, 이메일 외 비밀번호의 값은 문자열 붙이기 연산 후 해시하여 전송 전송 형태는 HTTP POST 표준을 따름.
종료 동작	서버 처리 결과에 의한 에러 팝업 서버 처리 결과에 의한 다음 동작 페이지

[표 5] 사용자 GUI (2) 로그인 오류

이름	사용자 GUI 2. 로그인 오류
목적 및 설명	상기 사용자 GUI 1. 로그인 및 등록 메인 화면에서 사용자가 서버에 요청한 로그인 HTTP POST의 처리 결과가 에러일 경우 발생한다.
입출력 형태	<p>서버로부터 HTTP Response를 수신하고, 해당 내용이 오류에 해당하는 경우 GUI 1의 JavaScript에 의해 팝업창이 발생하며, 팝업창의 내용은 수신한 POST Response의 Content에 의해 결정된다. 팝업창에 같이 표기되는 버튼을 클릭하여 창을 닫고 로그인 및 등록 화면 GUI로 돌아갈 수 있다.</p> <p>발생 가능한 오류의 종류는 다음과 같다.</p> <ul style="list-style-type: none"> ✓ 비밀번호가 일치하지 않음 ✓ 존재하지 않는 계정임 <p>*상기 두 에러는 서버 측에서는 서로 다른 접근이지만 보안성을 위해 동일한 에러로 표기함</p> <p>*이외의 동작은 상기 [표 3]의 웹 서비스 GUI에 상속됨</p>
범위/정확도/오차범위	*상기 [표 4]의 웹 서비스 GUI 로그인 및 등록 화면에 상속됨
단위	HTML DOM 요소와의 상호작용 (팝업 창 활성화, 버튼 클릭) HTTP POST 요청 통신 (HTTP Response 수신)
시간/ 속도	인터페이스 표기: 상기 [표 3]의 웹 서비스 GUI에 상속됨 HTTP POST 요청 처리: 서버 처리 시간에 상속됨
다른 입출력 간 연관성	*상기 [표 3]의 웹 서비스 GUI에 상속됨
화면의 형태	<p>아래의 화면과 같은 형태를 띤다.</p> <p>*세부적인 사항은 상기 [표 3]의 웹 서비스 GUI에 상속됨</p> <p>[그림 4] 로그인 오류 팝업 예시</p> 
창의 형태	*상기 [표 3]의 웹 서비스 GUI에 상속됨
데이터 형태	전송 형태는 HTTP POST Response 표준을 따름
종료 동작	버튼 클릭으로 인한 로그인 및 등록 화면 GUI로의 회귀

[표 6] 사용자 GUI (3) 등록 오류

이름	사용자 GUI 3. 등록 오류
목적 및 설명	상기 사용자 GUI 1. 로그인 및 등록 메인 화면에서 사용자가 서버에 요청한 등록 HTTP POST의 처리 결과가 에러일 경우 발생한다. 혹은 HTTP 요청을 보내지 않았더라도 비밀번호 및 비밀번호 확인란의 값이 일치하지 않는 경우 GUI 1의 JavaScript에 의해 본 GUI가 출력될 수 있다.
입출력 형태	서버로부터 HTTP Response를 수신하고, 해당 내용이 오류에 해당하는 경우 GUI 2의 JavaScript에 의해 팝업창이 발생하며, 팝업창의 내용은 수신한 POST Response의 Content에 의해 결정된다. 팝업창에 같이 표기되는 버튼을 클릭하여 창을 닫고 로그인 및 등록 화면 GUI로 돌아갈 수 있다. 발생 가능한 오류의 종류는 다음과 같다. <ul style="list-style-type: none"> ✓ 이미 존재하는 계정명 ✓ 이미 등록된 이메일 *이외의 동작은 상기 [표 3]의 웹 서비스 GUI에 상속됨
범위/정확도/오차범위	*상기 [표 4]의 웹 서비스 GUI 로그인 및 등록 화면과 동일함
단위	HTML DOM 요소와의 상호작용 (팝업 창 활성화, 버튼 클릭) HTTP POST 요청 통신 (HTTP Response 수신)
시간/ 속도	인터페이스 표기: 상기 [표 3]의 웹 서비스 GUI에 상속됨 HTTP POST 요청 처리: 서버 처리 시간에 상속됨
다른 입출력 간 연관성	*상기 [표 3]의 웹 서비스 GUI에 상속됨
화면의 형태	아래의 화면과 같은 형태를 띤다. *세부적인 사항은 상기 [표 3]의 웹 서비스 GUI에 상속됨 [그림 5] 등록 화면 예시  * 에러 화면은 GUI 2. 로그인 오류와 동일한 팝업 형태임
창의 형태	*상기 [표 3]의 웹 서비스 GUI에 상속됨
데이터 형태	전송 형태는 HTTP POST Response 표준을 따름
종료 동작	버튼 클릭으로 인한 로그인 및 등록 화면 GUI로의 회귀

[표 7] 사용자 GUI (4) 관리자 페이지 접근

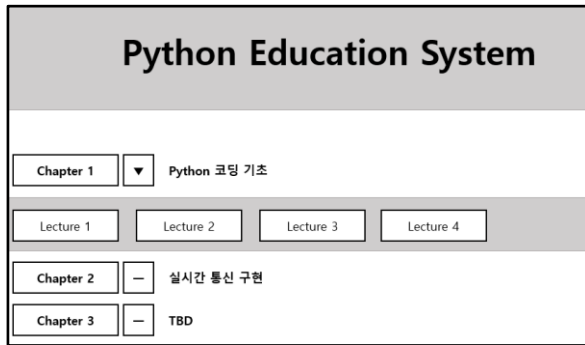
이름	사용자 GUI 4. 교육 내용 열람 - 관리자 페이지 접근
목적 및 설명	<p>상기 사용자 GUI 1. 로그인 및 등록 메인 화면에서 사용자가 서버에 요청한 등록 HTTP POST의 처리 결과가 정상일 경우 발생한다. 서버는 각 사용자에게 세션을 HTTP Cookie의 형태로 부여하고 계정과 세션을 결부하여 관리한다. 이 화면에서 사용자는 교육을 선택할 수 있다. 교육자료와의 상호작용은 GUI 6에서 서술된다.</p> <p>관리자 계정에 한해 이 화면에서 관리자 페이지를 열람하기 위한 요청 버튼을 확인할 수 있다.</p>
입출력 형태	<p>서버로부터 수신한 Html, Css, JavaScript에 근거하여 페이지를 렌더링한다. 계정이 관리자 계정일 경우 서버는 관리자 페이지 접근을 위한 GET 요청을 보낼 수 있는 버튼이 포함된 Html을 전송한다. 이외에 각 교육자료를 토글하여 교육자료를 선택하기 위한 Html이 표기된다.</p> <p>*관리자 계정이 아니면 관리자 페이지를 불러오는 GET 요청을 보내는 GUI가 없다.</p> <p>*관리자 계정이 아닌 Cookie를 가진 사람이 하는 GET요청은 반려된다.</p> <p>*이외의 동작은 상기 [표 3]의 웹 서비스 GUI에 상속됨</p>
범위/정확도/오차범위	*상기 [표 4]의 웹 서비스 GUI 로그인 및 등록 화면과 동일함
단위	HTML DOM 요소와의 상호작용 (팝업 창 활성화, 버튼 클릭) HTTP GET 요청 통신 (관리자 페이지 불러오기)
시간/ 속도	인터페이스 표기: 상기 [표 3]의 웹 서비스 GUI에 상속됨 HTTP GET 요청 처리: 서버 처리 시간에 상속됨
다른 입출력 간 연관성	*상기 [표 3]의 웹 서비스 GUI에 상속됨
화면의 형태	<p>아래의 화면과 같은 형태를 띤다.</p> <p>*세부적인 사항은 상기 [표 3]의 웹 서비스 GUI에 상속됨</p> <p>[그림 6] 관리자 페이지 접근 예시</p> <div data-bbox="448 1337 997 1628" data-label="Image"> </div>
창의 형태	*상기 [표 3]의 웹 서비스 GUI에 상속됨
데이터 형태	전송 형태는 HTTP GET Request 표준을 따름.
종료 동작	*상기 [표 3]의 웹 서비스 GUI에 상속됨

[표 8] 사용자 GUI (5) 관리자 페이지

이름	사용자 GUI 5. 관리자 페이지
목적 및 설명	상기 사용자 GUI 4. 교육 내용 열람 화면에서 사용자가 관리자 계정으로 접근, Html내 HTTP Get 요청을 전송하는 버튼을 통해 접근한 관리자 페이지이다. 관리자의 SSH 접근을 위한 공용키 및 관리자 간 공유하는 MD 파일을 읽을 수 있다.
입출력 형태	서버로부터 수신한 Html, Css, JavaScript에 근거하여 페이지를 렌더링한다. 계정이 관리자 계정이 아닐 경우 서버에서 HTTP Cookie 기반으로 세션을 구분하여 정보를 제공하지 않는다. 관리자 간 공유하는 MD 파일은 서버로부터 정해진 파일이 렌더링된다. SSH 공용키는 파일의 형태로 브라우저에서 다운로드할 수 있다.
범위/정확도/오차범위	*상기 [표 4]의 웹 서비스 GUI 로그인 및 등록 화면과 동일함
단위	HTML DOM 요소와의 상호작용 (공용키 다운로드)
시간/ 속도	인터페이스 표기: 상기 [표 3]의 웹 서비스 GUI에 상속됨 SSH 공용키 다운로드 처리: 사용자와 서버 간 통신상태에 상속됨
다른 입출력 간 연관성	*상기 [표 3]의 웹 서비스 GUI에 상속됨
화면의 형태	아래의 화면과 같은 형태를 띤다. *세부적인 사항은 상기 [표 3]의 웹 서비스 GUI에 상속됨 [그림 7] 관리자 페이지 예시 <div data-bbox="446 1115 1069 1426" data-label="Image"> </div>
창의 형태	*상기 [표 3]의 웹 서비스 GUI에 상속됨
데이터 형태	파일 전송 형태는 TCP 기반 암호화 통신을 따름
종료 동작	*상기 [표 3]의 웹 서비스 GUI에 상속됨

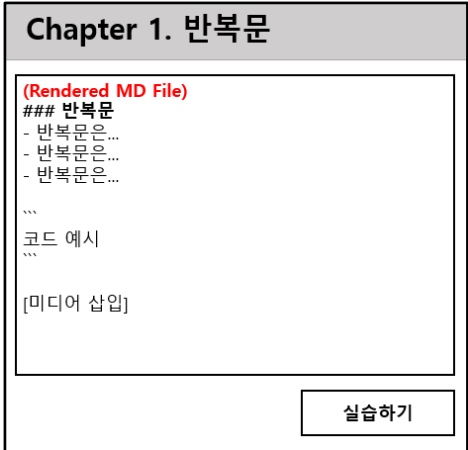
[표 9] 사용자 GUI (6) 교육 선택

이름	사용자 GUI 6. 교육 내용 열람 - 교육 선택
목적 및 설명	상기 사용자 GUI 4. 교육 내용 열람 화면에서 사용자가 수강을 원하는 교육을 선택하기 위한 인터페이스이다. 서버 내 존재하는 모든 강의를 표기하지만 대분류로 구분하여 토글된다. 토글된 대분류를 확장하고 특정 교육

이름	사용자 GUI 6. 교육 내용 열람 - 교육 선택
	자료를 선택하여 교육자료 페이지로 이동할 수 있다.
입출력 형태	서버로부터 수신한 Html, Css, JavaScript에 근거하여 페이지를 렌더링한다. 교육자료를 선택하여 페이지를 이동할 때에는 HTTP GET 요청을 사용한다. * 서버 내 계정 정보를 기반으로 이미 수강한 강의는 색상으로 구분된다.
범위/정확도/오차범위	*상기 [표 4]의 웹 서비스 GUI 로그인 및 등록 화면과 동일함
단위	HTML DOM 요소와의 상호작용 (대분류 토글) HTTP GET 요청 (교육자료 페이지 요청)
시간/ 속도	인터페이스 표기: 상기 [표 3]의 웹 서비스 GUI에 상속됨 교육자료 페이지 요청: 사용자와 서버 간 통신상태에 상속됨
다른 입출력 간 연관성	*상기 [표 3]의 웹 서비스 GUI에 상속됨
화면의 형태	아래의 화면과 같은 형태를 띤다. *세부적인 사항은 상기 [표 3]의 웹 서비스 GUI에 상속됨 [그림 8] 교육 내용 열람 및 선택 예시 
창의 형태	*상기 [표 3]의 웹 서비스 GUI에 상속됨
데이터 형태	HTTP GET 요청 표준을 따름
종료 동작	*상기 [표 3]의 웹 서비스 GUI에 상속됨

[표 10] 사용자 GUI (7) 교육 자료

이름	사용자 GUI 7. 교육 자료
목적 및 설명	상기 사용자 GUI 6. 교육 내용 열람 화면에서 사용자가 수강을 원하는 교육을 선택한 후 교육자료를 수강할 때의 인터페이스이다. 교육자료 하단에는 '실습하기' 혹은 '완료하기' 버튼이 존재한다. 각 버튼은 해당 페이지로 연결된다. 또한 '다음 차수 수강' 버튼이 있어 다음 자료로 바로 연결할 수 있다.
입출력 형태	서버로부터 수신한 Html, Css, JavaScript에 근거하여 페이지를 렌더링한다. 이 때, 각 교육자료는 MD파일 형태로 렌더링하고 사용자에게 전달된다. '실습하기' 버튼은 실습 페이지에 HTTP GET 요청을 전송한다. '완료하기' 버튼은 서버에 HTTP POST 요청을 전송하고 GUI 6으로

이름	사용자 GUI 7. 교육 자료
	Redirect(HTTP GET 요청)한다. '다음 차수 수강' 버튼은 다음 강의 항목을 HTTP GET 요청한다.
범위/정확도/ 오차범위	*상기 [표 4]의 웹 서비스 GUI 로그인 및 등록 화면과 동일함
단위	HTTP GET 요청 ('다음 차수 수강' 혹은 '실습하기') HTTP POST 요청 ('완료하기')
시간/ 속도	인터페이스 표기: 상기 [표 3]의 웹 서비스 GUI에 상속됨 실습 문제 및 다음 차수 요청: 사용자와 서버 간 통신상태에 상속됨 *서버 내 MD 파일 렌더링 시간 또한 요구됨
다른 입출력 간 연관성	*상기 [표 3]의 웹 서비스 GUI에 상속됨
화면의 형태	아래의 화면과 같은 형태를 띤다. *세부적인 사항은 상기 [표 3]의 웹 서비스 GUI에 상속됨 [그림 9] 교육자료 예시 
창의 형태	*상기 [표 3]의 웹 서비스 GUI에 상속됨
데이터 형태	HTTP GET 요청 표준을 따름 HTTP POST 요청 표준을 따름
종료 동작	*상기 [표 3]의 웹 서비스 GUI에 상속됨

[표 11] 사용자 GUI (8) 실습 환경

이름	사용자 GUI 8. 실습 환경
목적 및 설명	상기 사용자 GUI 7 교육 자료 화면에서 '실습하기' 버튼을 선택하여 접근 가능한 인터페이스이다. 사용자의 코드를 입력으로 받고, 해당 코드의 처리 결과를 반환하는 역할을 수행한다. 처리 결과는 페이지 내 콘솔 영역에 표기된다. 코드의 처리 요청을 보내는 '실행' 버튼이 존재한다.
입출력 형태	서버로부터 수신한 Html, Css, JavaScript에 근거하여 페이지를 렌더링한다. 이 때, 실습 문제는 MD 파일을 렌더링하여 전송한다. '실행' 버튼은 사용자의 코드를 서버에 HTTP POST 요청으로 전달한다.

이름	사용자 GUI 8. 실습 환경
	HTTP POST 응답을 근거로 에러 여부 및 실행 결과값을 콘솔에 표기한다. 모든 결과를 통과한 경우 팝업창을 통해 수강 완료를 알린다.
범위/정확도/ 오차범위	*상기 [표 4]의 웹 서비스 GUI 로그인 및 등록 화면과 동일함
단위	HTTP POST 요청 ('실행' 버튼 동작)
시간/ 속도	인터페이스 표기: 상기 [표 3]의 웹 서비스 GUI에 상속됨 코드 실행 요청: 사용자와 서버 간 통신 상태 및 서버 성능에 상속됨
다른 입출력 간 연관성	*상기 [표 3]의 웹 서비스 GUI에 상속됨
화면의 형태	아래의 화면과 같은 형태를 띤다. *세부적인 사항은 상기 [표 3]의 웹 서비스 GUI에 상속됨
창의 형태	<p>*상기 [표 3]의 웹 서비스 GUI에 상속됨</p> <p>[그림 10] 실습 환경 예시</p>
데이터 형태	HTTP POST 요청 표준을 따름
종료 동작	*상기 [표 3]의 웹 서비스 GUI에 상속됨

3.1.2. Hardware Interfaces

[표 12] 브라우저 사용에 필요한 하드웨어 인터페이스

이름	Chrome 브라우저에 사용 가능한 하드웨어 인터페이스
Graphic Driver	Graphic Processing Unit Specific Driver
HIDs Driver	Mouse / Keyboard Specific Driver
NIC Driver	Network Interface Chip Driver

* 본 서비스 이용을 위해 의존성이 있는 프로그램 외 별도의 요구사항은 없다.

3.1.3. Software Interfaces

[표 13] 서비스 서버 소프트웨어 인터페이스

이름	서버 구축에 필요한 소프트웨어 인터페이스	
목적 및 설명	사용자에게 교육자료를 제공하고, 로그인 및 등록을 처리하며 실습 과정에서 코드 수행을 위해 서버가 요청하는 소프트웨어 패키지 인터페이스이다.	
NodeJS 17.8.0~	설명	서비스 제공을 위한 웹 어플리케이션 엔진
	입력	서버 동작 구현을 위한 JavaScript 파일 및 Express 리소스
	출력	외부 접근 가능한 Html, Css, JavaScript 등 렌더링 자원
SQLite3 3.38.2~	설명	사용자 데이터 보관 및 처리를 위한 데이터베이스 엔진
	입력	사용자 데이터 처리를 위한 질의문
	출력	결과 반영 혹은 질의 응답
Judge0 Git Commit Hash 7dba...(2022 April)	설명	사용자의 실습 코드를 채점하기 위해 사용
	입력	사용자의 실습 코드
	출력	실습 코드 수행 결과
docker 20.10.11~	설명	사용자의 실습 코드를 실제로 실행하기 위해 사용
	입력	사용자의 실습 코드
	출력	실습 코드 수행 결과
npm: markdown-it 12.3.2~	설명	사용자에게 MD 파일 형태의 교육자료를 제공하기 위해 사용
	입력	MD파일 형식의 교육자료
	출력	Html, Css, JavaScript 등 브라우저 렌더링 자원
npm: highlights 3.1.6	설명	사용자가 입력한 코드의 문법을 강조하기 위해 사용
	입력	사용자가 입력한 코드
	출력	문법적 요소가 강조된 코드
SSH Demon (RFC 4254)	설명	서버 관리를 위해 서비스 관리자가 SSH 접근을 할 때 사용
	입력	서비스 관리자의 SSH Client를 통한 Stdin 입력
	출력	서비스 관리자의 접근에 대해 TTY 출력 전달

[표 14] 서비스 사용자 소프트웨어 인터페이스

이름	서비스 이용을 위한 사용자 소프트웨어 인터페이스	
목적 및 설명	사용자가 본 서비스를 이용하기 위한 필수적, 선택적 소프트웨어 인터페이스이다.	
브라우저 Chrome 100~	설명	사용자가 본 제품과 상호작용하기 위한 기본 환경
	입력	HID 기반 사용자 입력
	출력	Html, Css, JavaScript 렌더링 페이지
※Python 3.10.x~	설명	사용자가 학습 내용을 실습해보기 위한 환경
	입력	사용자 입력 코드
	출력	실행 결과 및 오류 보고
SSH Client Latest Ver.	설명	서비스 관리자가 서버 관리를 위해 사용하는 서버 접속 통신
	입력	서비스 서버 내 TTY Shell 로그인 및 입력
	출력	서비스 서버 내 TTY Shell 출력
※Windows Terminal Latest Ver.	설명	사용자가 Python, SSH를 사용하기 위한 Windows CLI
	입력	사용자 운영체제 내 TTY
	출력	명령 및 프로그램 실행 결과

* ※표기된 소프트웨어 인터페이스는 필수 요소가 아니다.

3.1.4. Communication Interfaces

[표 15] 로그인 HTTP 요청 형태

이름	로그인 HTTP 요청
용도	사용자가 서비스 이용을 위해 로그인 할 때 서버로 전송하는 요청
Method	POST
URL	main.com/login
Content Type	text/application
내용	사용자 계정 및 비밀번호 *비밀번호를 직접 전송하지 않고 계정명과 합친 해시값을 보낸다
서버 동작	Sqlite를 활용하여 로그인 가능 여부 판단 로그인 된 경우 관리자 여부를 판단하여 페이지 응답 로그인 이후 브라우저 쿠키를 활용하여 세션 관리
인코딩	Base64인코딩 및 전송

[표 16] 계정 등록 HTTP 요청 형태

이름	계정 등록 HTTP 요청
용도	사용자가 서비스 이용을 위해 계정 등록할 때 서버로 전송하는 요청
Method	POST
URL	main.com/register
Content Type	text/application
내용	사용자 계정 및 비밀번호, 이메일 *비밀번호 2차 확인 과정은 사용자 JavaScript 수준에서 이루어진다.
서버 동작	Sqlite를 활용하여 가입 가능 여부 판단 가입된 경우 자동 로그인 후 교육자료 페이지로 이동
인코딩	Base64인코딩 및 전송

[표 17] 실습 코드 실행 HTTP 요청 형태

이름	실습 코드 실행 HTTP 요청
용도	교육자료에 해당하는 실습 수행 시 사용자의 코드 동작 여부 검증
Method	POST
URL	main.com/theme*/practice
Content Type	text/application
내용	사용자가 작성한 코드 내용 일체 브라우저 쿠키(로그인 과정에서 생성된 세션값)
서버 동작	사용자의 코드를Docker 환경 혹은 Judge0 기반으로 실행 실행 결과 및 에러 발생 여부를 응답
인코딩	Base64인코딩 및 전송

※URL의 *은 Wildcard를 의미한다.

[표 18] 일반적인 페이지 렌더링 자원 HTTP 요청 형태

이름	계정 등록 HTTP 요청
용도	페이지 간 이동 및 자료 제공
Method	GET
URL	main.com/*

이름	계정 등록 HTTP 요청
Content Type	html, css, javascript
내용	페이지 렌더링을 위한 CSS, JavaScript, Html 등
서버 동작	사용자가 요청한 페이지에 해당하는 자료 전달 *MD 파일을 렌더링할 경우 NodeJS 플러그인 사용
인코딩	별도 인코딩 없음

※URL의 *은 Wildcard를 의미한다.

3.2. Functional Requirements

3.2.1. Use Case

[표 19] 사용례 (1) 로그인

이름	로그인
행위자(Actor)	사용자
설명	사용자는 사전에 등록한 자신의 계정으로 서비스를 이용할 수 있다. 자신의 계정으로 서비스를 활성화하기 위해선 웹 로그인 창에 관련 정보를 입력하고 POST 메시지를 전송해야 한다. 이후 서버는 해당 값을 검증하고 로그인 여부를 판단한다.
정상 흐름	①사용자가 이미 등록된 계정과 비밀번호를 입력한다. ②1에서 입력한 정보를 기반으로 비밀번호 해시값을 만든다. ③계정명과 비밀번호 해시값을 HTTP POST 메시지로 전송한다. ④서버는 수신한 데이터를 Sqlite3 데이터베이스 질의로 검증한다. ⑤-A.사용자의 계정에 문제가 없을 경우 학습 자료 선택 화면으로 Redirection이 수행된다. ⑤-B.사용자의 계정이나 서버에 문제가 있을 경우 팝업창을 통해 재시도를 요구한다.
전제 조건	서버에 100개 이하의 세션이 유지되어 충분한 공간이 있다. 사용자는 사전에 계정과 비밀번호를 등록했다.
사후 조건	로그인 성공 시 계정과 세션을 결부하기 위해 브라우저 Cookie를 수정한다.
가정	없음

[표 20] 사용례 (2) 계정 등록

이름	계정 등록
행위자(Actor)	사용자
설명	사용자는 서비스를 이용하기 위해 반드시 계정을 생성해야 한다. 각 계정은 이메일, 계정명, 비밀번호 3개의 정보로 생성 가능하며 비밀번호는 사용자 브라우저의 javascript를 이용하여 비밀번호 재입력 값과 비교한다. 실질적으로 계정을 생성하는 과정에서는 서버에 HTTP POST 요청을 전송하며 그 결과 혹은 에러 여부를 사용자에게 전달한다.
정상 흐름	①사용자가 생성을 원하는 계정명과 비밀번호, 이메일을 입력한다. ②비밀번호 재입력을 요구한다. 1번의 입력과 다를 경우 재시도를 요구한다. ③계정명, '계정명+비밀번호'의 해시값, 이메일 세 정보를 HTTP POST 요청으로 전송한다. ④서버는 수신한 데이터를 Sqlite3 데이터베이스 질의로 중복 검증한다. ⑤-A.사용자의 계정에 문제가 없을 경우 새로운 사용자를 추가한다. ⑤-B.사용자의 계정명이나 이메일이 중복되었을 경우 오류 메시지를 출력하고 재시도를 요구한다.
전제 조건	사용자는 본 서비스에 처음 접근한다. 동일한 이메일로 다른 계정이 존재하지 않는다.
사후 조건	계정 생성 시 수강한 과목 DB를 생성하고 초기화한다.
가정	없음

[표 21] 사용례 (3) 학습 자료 선택

이름	학습 자료 선택 및 수강
행위자(Actor)	사용자
설명	사용자는 학습자료 항목에서 본인이 어디까지 학습했는지 확인할 수 있다. 또한 각 항목을 선택하여 학습을 시작할 수 있다.
정상 흐름	①사용자는 학습 완료 체크된 항목을 확인하고 차후 학습할 내용을 선택한다. ②학습 페이지로 이동하며, MD파일의 학습자료는 서버에서 렌더링되고 제공된다. ③브라우저를 통해 사용자는 교육 자료를 읽고 공부한다.
전제 조건	사용자가 현재 어디까지 학습했는지 데이터베이스에서 확인할 수 있다.
사후 조건	완료 버튼 혹은 실습 버튼으로 수강 상태를 변경할 수 있다.
가정	없음

[표 22] 사용례 (4) 관리자 페이지 접근

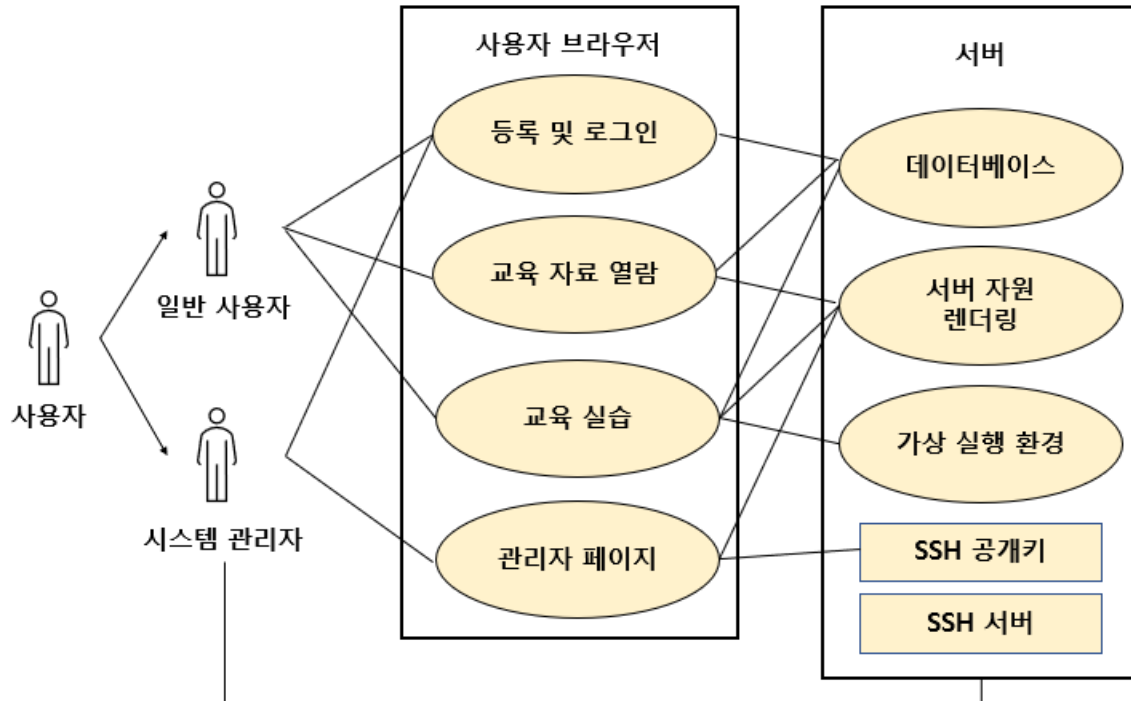
이름	관리자 페이지 접근
행위자(Actor)	사용자 (서비스 관리자)
설명	서비스 관리자는 관리자 간 공지사항의 공유를 위해, 또한 서버 접속 SSH 키를 얻기 위해 관리자 페이지에 접근할 수 있다.
정상 흐름	①서비스 관리자는 로그인 후 관리자 페이지 버튼을 통해 페이지를 불러온다. ②관리자 페이지 접근 요청이 발생했을 때 서버는 요청을 확인한다. ③요청 내용의 계정 및 세션이 관리자 권한인지 검증한다. ④관리자 페이지를 통해 관리자 간 공지사항을 열람한다. ⑤SSH 공개키 다운로드를 통해 관리자는 서버에 접근할 수 있다.
전제 조건	서비스 관리자는 서버를 관리하기 위해 서버에 접근해야 하지만 SSH 공개키가 없어 접근할 수 없다.
사후 조건	공개키는 외부에 유출되지 않도록 관리해야 한다.
가정	없음

[표 23] 사용례 (5) 실습 코드 실행 요청

이름	실습 코드 실행 요청
행위자(Actor)	사용자
설명	사용자는 학습 자료를 기반으로 배운 내용을 실험하기 위해, 서비스 관리자가 만든 실습을 진행할 수 있다. 실습에서 주어진 문제에 맞는 코드를 작성하여 서버에 실행을 요청하고 그 결과값을 받아볼 수 있다.
정상 흐름	①사용자는 학습 자료 수강 후 실습하기 버튼을 통해 실습 페이지를 불러온다. ②실습 내용에 있는 요구 조건을 잘 읽고 코드를 작성한다. ③작성한 코드는 HTTP POST 요청으로 서버에 전달된다. ④서버는 코드를 수행하여 에러 추적 및 결과 평가 후 사용자에게 전달한다. ⑤사용자는 결과물을 브라우저 내 가상 콘솔창을 통해 관찰함으로써 실제 환경인 것처럼 느낀다.
전제 조건	사용자는 학습 자료를 보고 실습 버튼을 눌러 본 페이지에 접근한다.
사후 조건	정상 실행 완료 시 해당 학습 자료의 수강 처리를 완료한다.
가정	없음

3.2.2. Use Case Diagram

[그림 11] 사용례 도식



3.2.3. Data Dictionary

[표 24] 데이터베이스 Table 'User' 구조

Field	Key	Constraint	Description
id	PK	Not Null	계정명
hashed_password		Not Null	비밀번호 해시값(비식별화)
email		Not Null	이메일 정보
Admin		Boolean	초기값 False

* 새로운 사용자가 등록될 때 본 테이블에 값을 추가해야 한다.

[표 25] 데이터베이스 Table 'Class' 구조

Field	Key	Constraint	Description
class_id	PK	Not Null	수업 번호
practice_exist		Boolean	실습 존재여부

* 시스템 관리자가 수업을 추가할 때 본 테이블에 값을 추가해야 한다.

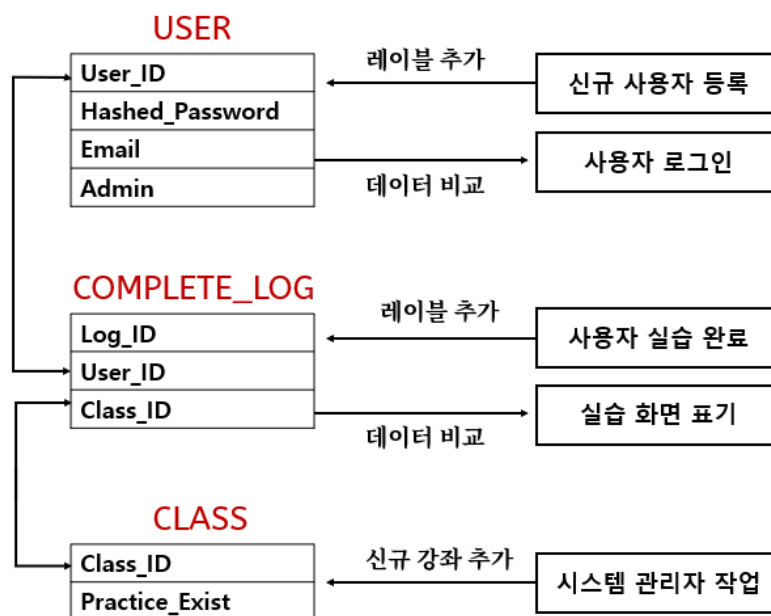
[표 26] 데이터베이스 Table ‘Complete_Log’ 구조

Field	Key	Constraint	Description
id	PK	Not Null	수강 번호
user_id		Not Null	수강자 식별자
class_id		Not Null	수업 식별자

* 임의의 사용자가 임의의 수업을 완료했을 때 본 테이블에 값이 추가된다.

3.2.4. Data Flow Diagram

[그림 12] SQLite Database 구조 및 상호작용



3.3. Performance Requirements

다음 항목들은 본 서비스가 제공해야 할 성능의 요구조건을 나타낸다.

3.3.1. Static numerical requirement

- ✓ 본 시스템은 사용자가 복수의 코드 검증(실습 실행)을 수행하는 것을 제공하지 않는다. 각각의 사용자는 세션 및 쿠키를 통해 구분된다.
- ✓ 사용자의 코드 검증은 특별한 실습 조건이 아닌 한 5초의 실행 제한시간을 가지며 그 시간을 벗어나면 시간 초과 실행으로 전달한다.

- ✓ 본 시스템은 하드웨어 요구조건을 만족하는 한, 브라우저 및 응용 프로그램에서 연결 종료라고 판단하는 TTL 이전에 정보를 전달해야 한다.

3.3.2. Dynamic numerical requirement

- ✓ 본 시스템은 복수의 관리자가 관리할 수 있으며, SSH 공개키를 통한 셀 획득으로 동시 작업이 가능하다.
- ✓ 본 시스템은 최대 100명의 동시 사용자를 가정하고 있으며, 그 이상의 인원 에 대해서는 최적의 품질을 제공하기 어렵다.
- ✓ 상기 실행시간 5초에 근거하여, 각 실습의 검사 시간은 테스트 케이스의 수 X 5 초 안에 종료되어야 한다.
- ✓ 페이지 이동 등 데이터베이스에서 자료를 찾지 않고, 데이터베이스를 수정하지 않는 동작은 5초 안에 수행되어야 한다.
- ✓ MD 해석 등의 연산을 요구하는 경우 서버로부터 페이지를 전달받을 때 10초 이내에 수행되어야 한다.
- ✓ 데이터베이스에서 자료를 탐색하는 과정을 동반하는 각 기능은 10초 이내에 수행되어야 한다.

3.4. Logical Database Requirements

본 시스템은 SQLite 데이터베이스를 사용한다. 서버와의 연동을 위해 NodeJS 의 SQLite3 플러그인을 사용한다. 사용자 정보, 강의 정보를 데이터베이스에 저장한다. 사용자가 추가될 때마다 사용자 테이블에 새로운 레이블이 생성된다. 강의가 생성될 때마다 강의 테이블에 새로운 레이블이 생성된다. 학습 여부 기록 테이블에서는 임의의 사용자가 임의의 강의를 마칠 때마다 그 순서쌍을 레이블의 형태로 저장하며, 추후 이 레이블을 참고하여 사용자가 이미 수강한 강좌 정보를 가져올 수 있다. 이를 위해 SQLite는 JOIN 쿼리 및 동시다발적인 쿼리에 대한 동작을 수행할 수 있어야 한다. SQLite 데이터베이스는 본 시스템을 위해 필요한 모든 기능을 큰 성능 부하 없이 제공한다.

3.5. Design Constraints

본 시스템은 Apache2 라이선스를 따른다. NodeJS 기반의 웹 서비스를 제공한다. 서버는 사용자에게 교육 자료를 제공한다. 사용자가 입력한 코드를 서버가 수행하고 결과를 반환할 수 있다. 본 시스템은 NodeJS, Docker, Judge0, SQLite를 기반 설계되었다.

3.6. Standards compliance

시스템 내의 모든 JavaScript는 ECMAScript 표준을 따른다. NodeJS 구현 및 사용자에게 전달되는 JavaScript 양 측에 해당되는 내용이다. CSS는 W3C의 표준을 따른다. 사용자에게 전달되는 Html의 경우 NodeJS의 템플릿 엔진 Express를 활용한다. Express 표준과 HTML5 표준이 상충할 경우 Chrome 브라우저에서 해석하는 것을 기준으로 한다. 동일한 경우 HTML5의 내용을 따른다.

서비스 가동 중 발생하는 모든 네트워크 통신은 TCP, HTTP 프로토콜 표준을 따르며 각 항목은 RFC 793, RFC 2616에 명시된 바를 따른다.

3.7. Software System Characteristics

소프트웨어 시스템 특성은 비 기능적 요구조건으로부터 발생한다. 이 항목에서는 본 시스템의 비 기능적 요구조건을 소개한다. 비 기능적 요구조건은 제품 요구조건(Product Requirements), 조직 요구조건(Organization Requirements), 외부 요구조건(External Requirements) 3개로 구분한다.

3.7.1. Product Requirements

제품 요구조건 항목은 본 시스템의 동작 중에 지켜져야 하는 요구조건들을 뜻한다. 본 시스템은 운영 중 다음과 같은 요구조건을 충족해야 한다.

3.7.1.1. Usability Requirements

본 시스템은 사용자에게 사용성 좋은 시스템을 제공해야 한다. 접근성이 좋은 브라우저 기반의 웹 어플리케이션으로 개발되었기 때문에, 사용자가 본 시스템을 사용할 때 별도의 설치 과정 및 요구 조건을 가지지 않는 것은 본 시스템의

주요 요구조건이다. 사용자가 본 시스템을 이용하기 위해 브라우저 이외의 도구를 사용하게 되는 것을 지양한다. 또한 본 시스템에서 제공하는 모든 교육자료는 관련 전문지식이 충분하지 않은 사람일지라도 이해할 수 있도록 설계되어 있다. 만일 충분한 배경지식이 있어야만 이해할 수 있는 내용이라면, 시스템 관리자는 배경 지식에 해당하는 교육 자료 또한 적절히 구성하고 사용자가 외부 서비스의 도움 없이 제공되는 내용을 이해할 수 있도록 설계해야 한다.

3.7.1.2. Performance Requirements

본 시스템은 세션과 계정 기반으로 사용자를 구분할 수 있다. 사용자가 서버에 코드 검증을 요청하는 형태임을 고려하여 본 서비스는 서로 다른 100명의 사용자 이상의 접근을 지양한다. 만일 더 큰 규모로 서비스를 확장할 경우 서버는 하드웨어, 소프트웨어 측면에서 개선이 필요할 수 있다. 또한 상대적으로 과부하의 대상이 되기 쉬운 웹 서비스 특성 상, 본 시스템은 한 사람이 동시에 여러 코드 검증을 시도하는 것을 허락하지 않는다.

상기한 요구 조건이 충족된다면, 본 시스템은 제공되는 모든 서비스를 브라우저 환경에서 무리 없이 사용할 수 있는 수준까지의 속도를 보장한다. 각각의 페이지 이동마다 5초, 자료 요청 및 서버 자료 수정 등에 대해서는 10초 이상의 지연이 발생하지 않도록 설계한다.

3.7.1.3. Security Requirements

본 서비스 이용 도중, 사용자는 서버에 임의의 데이터를 보낼 수 있다. 그 과정은 TCP 프로토콜을 기반으로 충분히 안전하지만, MITM 중간자 공격을 고려하여 사용자 데이터는 비식별화하여 전송한다. 사용자의 정보 보호 이외에도, 웹 서비스는 각 사용자에게 의해 데이터 유출 공격의 대상이 될 수 있다. 이러한 상황을 방지하기 위해 서버는 사용자가 전송하는 어떠한 HTTP 요청의 값도 신뢰하지 않고 검증한다. 또한 관리자 권한이 필요한 모든 동작에 대하여 세션 및 쿠키 기반으로 올바른 접근을 수행하고 있는지 확인한다. 최종적으로, 실습을 위해 사용자가 서버에 전송하는 Python 코드는 Sandbox 환경에서 실행한다. 이 환경은 Judge0 API 혹은 Docker 프로그램을 이용해 자연스럽게 구축할 수 있다.

3.7.2. Organizational Requirements

조직 요구조건은 사용자 집단 및 개발자 집단의 규제 및 절차에 의해 생성된 요구 조건을 뜻한다.

3.7.2.1. Environmental Requirements

- ✓ 본 시스템의 최종 라이선스인 Apache2 라이선스에 위배되거나 충돌하지 않는 한, 개발자는 어떠한 오픈 소스 API든 활용할 수 있다. 단, 이 경우 본 요구사항 명세서를 적절히 수정하여야 한다.
- ✓ 개발자는 시스템 관리자가 데이터베이스, Python, NodeJS 및 Express 엔진에 대한 기본적인 지식을 가지고 있다고 가정할 수 있다. 즉, 본 서비스의 시스템 관리자는 SSH 접근 후에는 별도의 시스템 관리 어플리케이션 없이 서버를 수정할 수 있다.

3.7.2.2. Operational Requirement

- ✓ 본 서비스는 본 문서에서 정해진 범주 내에서 서비스 이용에 장애가 발생하지 않아야 한다.
- ✓ 사용자는 본 서비스를 통해 최종적으로 ‘실시간 채팅’ 어플리케이션을 개발할 수 있을 정도의 교육을 제공받아야 한다. 본 서비스를 출시할 때 시스템 관리자가 별도로 추가하지 않더라도 실시간 채팅 어플리케이션 개발에 대한 기본적인 내용 교육과 실습은 포함되어 있어야 한다.

3.7.3. External Requirements

본 항목에서는 시스템의 운영과 개발에 있어 외부적으로 발생할 수 있는 요구조건을 서술한다.

3.7.3.1. Safety / Security Requirement

- ✓ 이 시스템은 각 개인정보가 충분히 비식별화되어, 서버에게 전송될 때 중간자 공격으로부터 원천적으로 자유롭게 설계되었다.
- ✓ 비식별화 과정에서 사용자의 비밀번호와 계정을 같이 해시했기 때문에, 레

인보우 공격으로부터 안전하다.

- ✓ 이 시스템은 세션 및 브라우저 쿠키를 바탕으로 사용자를 구분한다. 브라우저 쿠키는 외부로 유출되지 않아야 한다.
- ✓ 사용자가 서버로 보내는 모든 코드는 실행할 때 완전히 격리된 환경에서 수행되어야 한다.

3.7.3.2. Regulatory Requirement

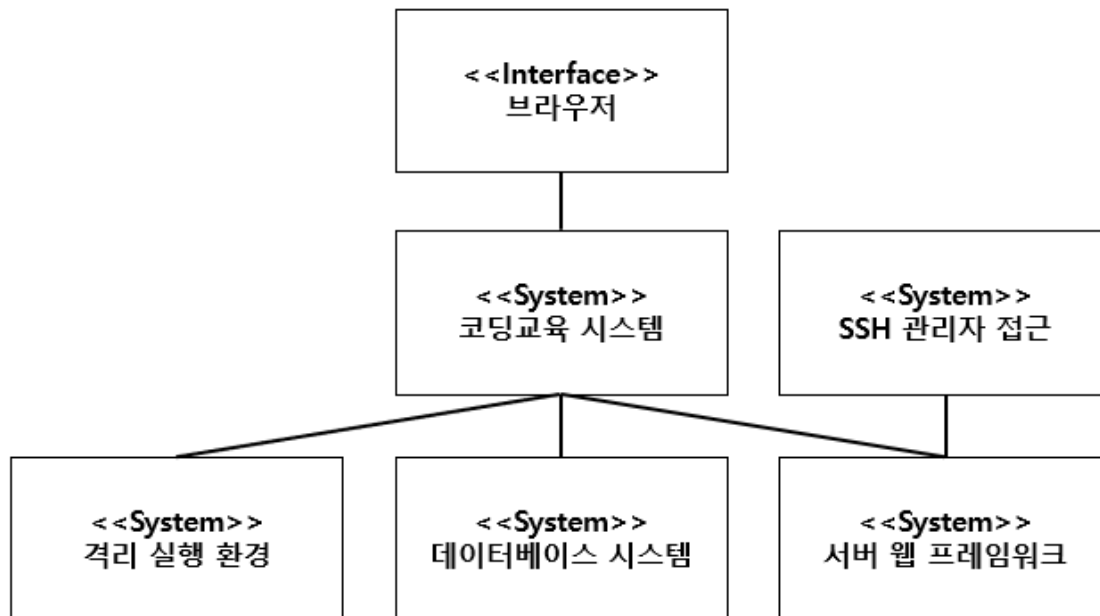
- ✓ 본 시스템에서 이용하는 Docker, NodeJS, Judge0 등 외부 API의 라이선스를 철저히 지켜야 한다. 만일 라이선스가 충돌할 경우 대체를 찾아야 한다.
- ✓ 로그인 서비스를 제공하기 위해 OAuth 프로토콜을 활용할 경우, OAuth의 공급자가 되는 사이트의 약관에 따라 서비스를 구현해야 한다.

3.8. Organizing the Specific Requirements

본 항목에서, 우리는 시스템 모델을 Unified Modeling Language(UML)에 기반한 가시적인 형태와 표를 기반으로 나타내고자 한다. 이러한 모델은 시스템, 서브 시스템, 시스템 내의 각 요소와 운영 환경 등 세부적인 요구사항의 형태를 명확하게 나타낼 수 있다.

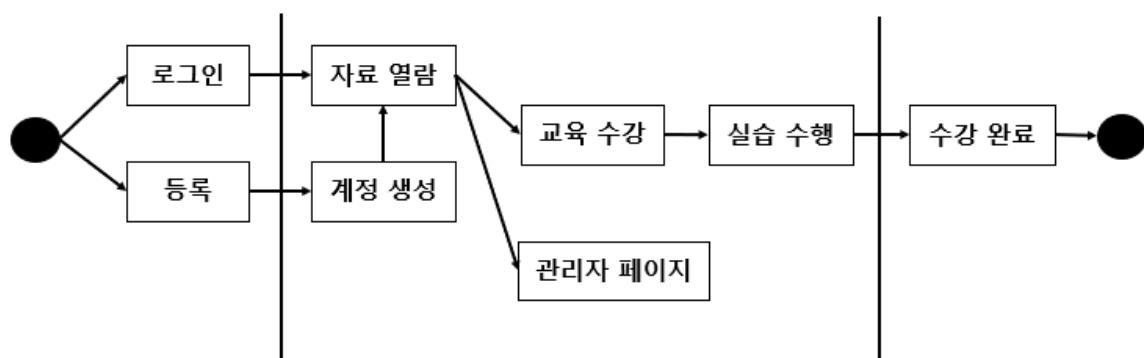
3.8.1. Context Model

[그림 13] 맥락 모델



3.8.2. Process Model

[그림 14] 프로세스 모델



3.8.3. Interaction Model

3.2.2 항목의 User Case Diagram 형태로 표현되었다.

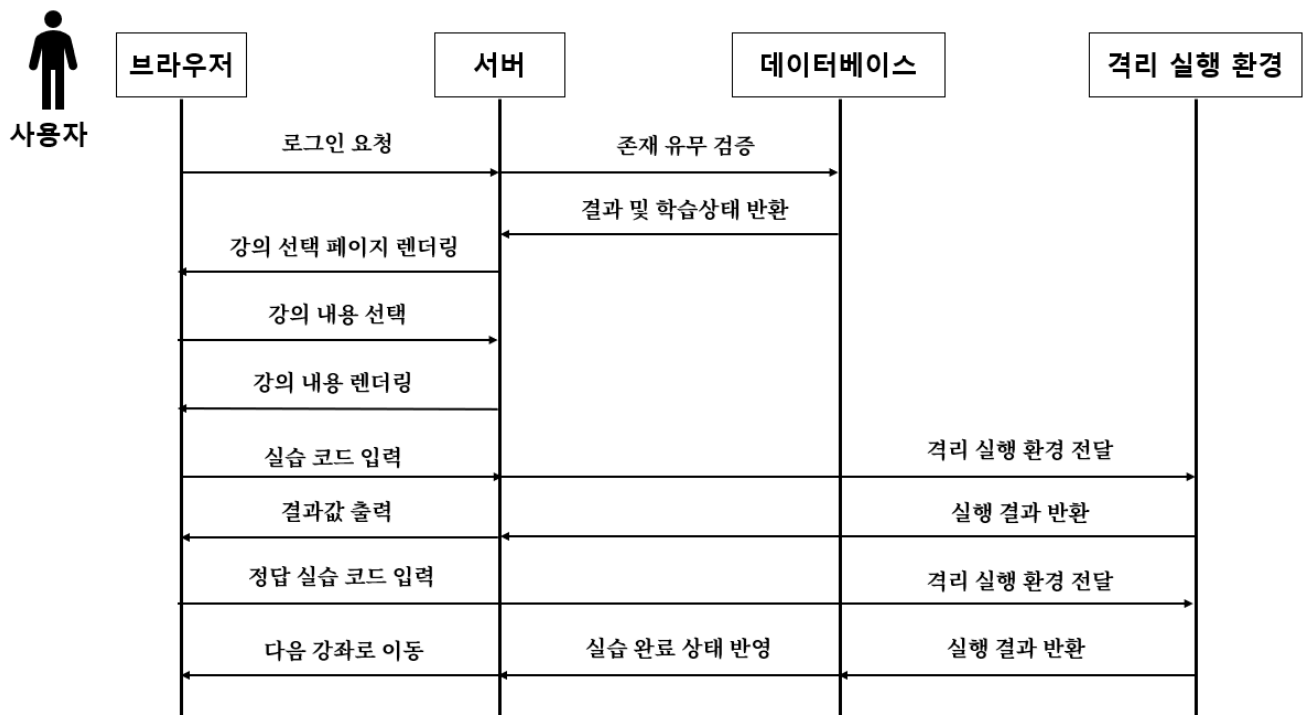
3.8.4. Behavior Model

3.8.4.1. Data Flow Diagram

3.2.4 항목에 Data Flow Diagram 형태로 표현되었다.

3.8.4.2. Sequence Diagram

[그림 15] 흐름 도식

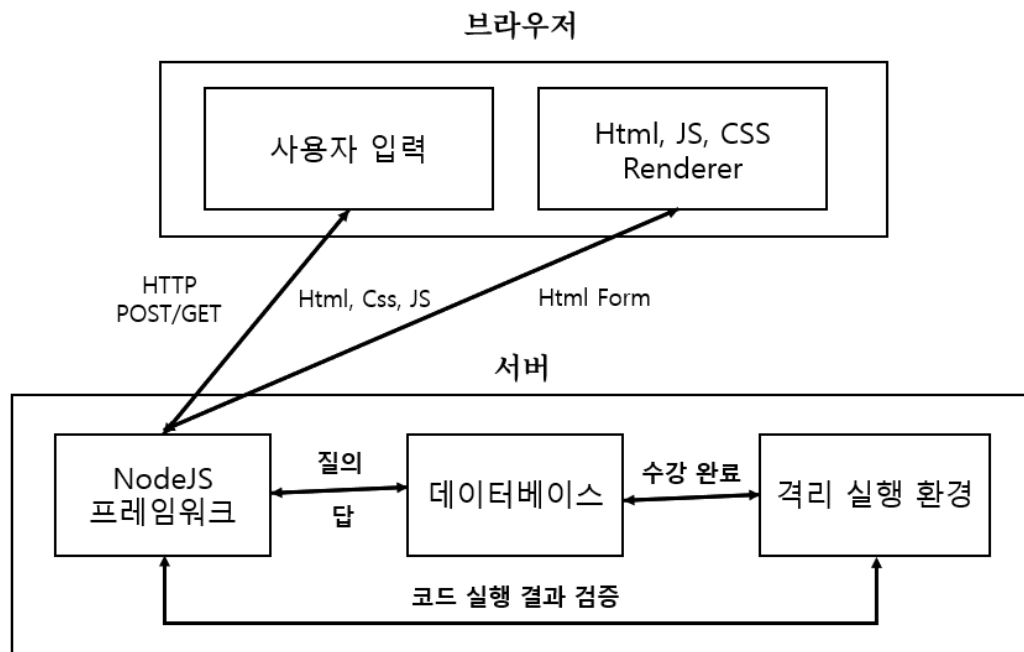


[그림 15] Sequence Diagram

3.9. System Architecture

본 항목은 시스템 구조를 고수준의 추상화된 도식 형태로 나타낸다. 시스템을 구성하는 각 요소 (사용자 브라우저, 서버)간 존재하는 기능과 통신에 대해 가장 근본적인 형태가 나타난다.

[그림 16] 시스템 구조



3.10. System Evolution

본 항목에서 우리는 시스템이 추후 하드웨어 개선 및 소프트웨어 개선, 혹은 새로운 요구사항이 발생할 때 대처하기 위한 기본적인 입장을 서술한다. 이 항목은 추후 본 소프트웨어의 방향성을 서술하는 것이기 때문에 시스템 디자이너는 이를 고려하여 시스템을 설계하여야 한다.

3.10.1. Limitation and Assumption

본 서비스는 브라우저 기반으로 사용자가 Python 프로그래밍을 연습할 수 있는 환경을 제공한다. 본 제품을 출시하는 시점에는 ‘기초 교육’ 및 ‘실시간 통신 구현’ 2가지의 대분류밖에 없지만, 추후 시스템 관리자와의 협의를 통해 다양한 대분류가 추가될 수 있다.

본 서비스는 사용자가 격리된 실행 환경 즉 Sandbox를 벗어나는 것이 불가능하다는 전제로 보안성을 설계했다. 따라서 Docker 혹은 Judge0의 격리 실행 환경을 위협하는 취약점이 발생할 경우 최대한 빠르게 업데이트를 수행하고 서버의

접근 기록을 점검하여야 한다.

3.10.2. Evolutions of Hardware and Change of User Requirements

본 서비스는 동시 접속자의 수를 100명으로 제한하고 한 사람당 수행할 수 있는 코드 검증의 수를 제한하였다. 추후 하드웨어 및 소프트웨어가 개선되더라도 한 사람당 동시에 두 개 이상의 코드 검증을 허가하지는 않는다. 단 동시 접속자의 수가 늘어날 경우, 하나의 서버에 접근하는 네트워크 통신에 병목 현상이 생길 수 있다.

이를 해결하기 위해 로그인 시점에서 Stun 서버 혹은 Relay 서버를 구축하여 사용자 부하를 조정해주는 작업이 필요할 수 있다. 따라서 일정 규모 이상 사용자 수용량을 늘리고자 한다면 요구 자원이 급증하게 된다. 관리자 및 설계자는 이 시점을 적절히 파악하여 비용 및 서비스 품질을 개선해야 한다.

사용자 요청 사항이 발생할 경우, 만일 새로운 기능을 개발하는 것이 아니라 단순히 내용물의 추가라면 시스템 관리자 선에서 해당 내용을 추가할 수 있다. 새로운 기능을 개발해야 하는 경우, 본 요구사항 명세서에 있는 내용들을 기반으로 새로운 API를 추가하거나 새로운 웹 서비스 페이지를 설계할 수 있다. 단 각 설계 단계에 있어 기존 요구사항 명세와 통일된 양식을 가져야 하며, 작성 후 본 문서 또한 수정되어야 한다.

4. Supporting Information

4.1. Software Requirement Specification

본 소프트웨어 요구 명세서는 IEEE 권장사항에 맞추어 작성되었다. 본 서비스에 적합한 요구 조건을 작성하기 위해 본래의 양식에서 일부 추가되거나 제외된 바 있다. (IEEE Recommended Practice for Software Requirements Specifications, IEEE-Std-830)

4.2. Document History

[표 27] 문서 이력

Date	Version	Description	Writer
2022.04.10.	1.0	팀원 회의 후 최초 버전 발행	Jihoi Kim, et al.