



코딩 테스트 웹사이트

소프트웨어 요구사항 명세서

2022.10.30.

소프트웨어공학개론 41

TEAM 8

팀장 박건우

팀원 김동희

서의태

윤재한

정찬

목차

1. Introduction.....	5
1.1. Purpose	5
1.2. Scope	5
1.3. Definitions, Acronyms, and Abbreviation	6
1.4. Reference	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	9
2.1.4. Software Interfaces	10
2.1.5. Communications Interfaces	10
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. Select Assignments	12
2.2.3. Solve Assignment	13
2.2.4. Save Code	13
2.2.5. Verifying Test Case	13
2.2.6. Execute & Verifying User Code	14
2.2.7. Submit User Code & Analysis	14
2.3. User Characteristics	14
2.3.1. Administrator	14
2.3.2. User	15
2.3.3. Ministry of Education	15
2.4. Constraints	15
2.5. Assumptions and Dependencies	16
3. Specific Requirements	17
3.1. External Interface Requirements	17
3.1.1. User Interfaces	17

3.1.2. Hardware Interfaces	23
3.1.3. Software Interfaces	24
3.1.4. Communication Interfaces	26
3.2. Functional Requirements	28
3.2.1. Use Case	28
3.2.2. Use Case Diagram	31
3.2.3. Data Dictionary	31
3.2.4. Data Flow Diagram	33
3.3. Performance Requirements	33
3.3.1. Static numerical requirement	33
3.3.2. Dynamic numerical requirement	33
3.4. Logical Database Requirements	34
3.5. Design Constraints	34
3.6. Standards compliance	34
3.7. Software System Characteristics	35
3.7.1. Product Requirements	35
3.7.1.1. Usability Requirements	35
3.7.1.2. Performance Requirements	35
3.7.1.3. Security Requirements	36
3.7.2. Organizational Requirements	36
3.7.2.1. Environmental Requirements	36
3.7.2.2. Operational Requirement	37
3.7.3. External Requirements	37
3.7.3.1. Safety / Security Requirement	37
3.7.3.2. Regulatory Requirement	37
3.8. Organizing the Specific Requirements	38
3.8.1. Context Model	38
3.8.2. Process Model	39
3.8.3. Interaction Model	39
3.8.4. Behavior Model	39
3.8.4.1. Data Flow Diagram	39
3.8.4.2. Sequence Diagram	40
3.9. System Architecture	40
3.10. System Evolution	41
3.10.1. Limitation and Assumption	41
3.10.2. Evolutions of Hardware and Change of User Requirements	42
4. Supporting Information	43

4.1.	Software Requirement Specification	43
4.2.	Document History	43

그림 목차

[그림 1]	Chrome 웹 브라우저 창 예시.....	17
[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]	문제 풀이 페이지 예시.....	23
[그림 8]	사용례 도식.....	31
[그림 9]	데이터 흐름 도식.....	33
[그림 10]	맥락 모델.....	38
[그림 11]	프로세스 모델.....	39
[그림 12]	Sequence Diagram.....	40
[그림 13]	시스템 구조.....	41

표 목차

[표 1]	약어정리.....	6
[표 2]	약어정리.....	7
[표 3]	웹 서비스에 대한 사용자의 GUI 기반 접근.....	17
[표 4]	사용자 GUI (1) 로그인 및 등록 화면.....	18
[표 5]	사용자 GUI (2) 로그인 오류.....	19
[표 6]	사용자 GUI (3) 등록 오류.....	20
[표 7]	사용자 GUI (4) 관리자 페이지.....	21
[표 8]	사용자 GUI (5) 문제 풀이 페이지.....	22
[표 9]	브라우저 사용에 필요한 하드웨어 인터페이스.....	23
[표 10]	서비스 서버 소프트웨어 인터페이스.....	24
[표 11]	서비스 사용자 소프트웨어 인터페이스.....	25
[표 12]	로그인 HTTP 요청 형태.....	26
[표 13]	계정 등록 HTTP 요청 형태.....	26
[표 14]	코드 실행 HTTP 요청 형태.....	27
[표 15]	코드 채점 HTTP 요청 형태.....	27

[표 16] 코드 제출 HTTP 요청 형태	27
[표 17] 일반적인 페이지 렌더링 자원 HTTP 요청 형태	28
[표 18] 사용례 (1) 계정 등록	28
[표 19] 사용례 (2) 로그인	29
[표 20] 사용례 (3) 문제 선택, 풀이, 제출	29
[표 21] 사용례 (4) 관리자 페이지	30
[표 22] 데이터베이스 'User' 테이블의 구조	31
[표 23] 데이터베이스 'Status' 테이블의 구조	31
[표 24] 데이터베이스 'Save Status' 테이블의 구조	32
[표 25] 데이터베이스 'Problem' 테이블의 구조	32
[표 26] 데이터베이스 'Problem Log' 테이블의 구조	32
[표 27] 문서 이력	43

1. Introduction

1.1. Purpose

해당 문서는 “Python Coding Test” 서비스 (이하 ‘본 서비스’)를 제공하기 위한 소프트웨어 요구사항 명세서이다. 본 서비스는 2022년 2학기 성균관대학교 소프트웨어공학개론 41분반 8조 (이하 ‘개발팀’)에 의해 개발된다. 해당 문서는 소프트웨어 시스템의 요구 사항을 분석, 정리했으며 이에 근거하여 본 서비스의 시스템을 개발하고 서비스한다.

해당 문서는 개발팀을 주요 독자로 작성되었다. 또한 교수, 조교 및 성균관대학교 소프트웨어공학개론 수강자도 본 문서를 열람할 수 있다.

본 서비스는 웹 어플리케이션을 기반으로 사용자가 Python 언어로 코딩 테스트 문제를 풀이할 수 있는 환경을 제공한다. 서비스 안에서 코드를 작성하여 제출하고, 그 결과와 문제 풀이에 도움이 되는 추천 콘텐츠를 제공받는다. 이를 위해 웹 서버, 코드 채점 시스템 및 추천 시스템 등이 개발되어야 하며 세부 사항은 아래 문서에 명시한다.

1.2. Scope

본 서비스는 Python 코딩 테스트 환경을 제공하기 위해 고안되었다. 코딩 테스트 문제와 작성한 코드의 동작, 효율, 가독성을 채점하고 문제 해결에 도움이 되는 영상, 웹 문서 등의 콘텐츠를 자동으로 추천하는 기능을 제공한다. 또한 계

정이 필요한 웹 서비스 형태로 제공하여 작성 중인 코드를 저장하고 이후 이어 작성할 수 있도록 한다. 이를 제공하기 위해 JavaScript, Python 기반의 웹 서버를 구축한다. 사용자는 코딩 테스트 문제를 풀어보며 코딩 테스트 환경에 익숙해질 수 있고 추천 콘텐츠를 통해 필요한 지식을 얻을 수도 있다.

1.3. Definitions, Acronyms, and Abbreviation

이하의 표는 본 문서에서 활용된 약어의 정의를 나타낸다.

[표 1] 약어 정의

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
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
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
Django	High-level Python web Framework
React	JavaScript Framework for building user interfaces
Docker	Linux Container Management Program

1.4. Reference

- IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications, In IEEEExplore Digital Library
<http://ieeexplore.ieee.org/Xplore/guesthome.jsp>
- Team 8. “Software Requirement Specification”. SKKU, Last Modified: Oct. 30, 2022.
https://github.com/skkuse/2022fall_41class_team8
- 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>

1.5. Overview

본 소프트웨어 요구사항 명세서의 이하 부분은 세 개의 Chapter로 구성되어 있다. Chapter 2.에서는 전반적인 서비스에 걸쳐 서비스의 특징과 기능, 사용자의 특징, 제약사항에 대해 서술한다. Chapter 3에서는 세부적인 요구사항 명세를 기술한다. 각 인터페이스의 요구사항 및 기능적인 요구사항, 시스템의 특징을 서술하고 그 형태들을 모델로 표현한다. Chapter 4.에서는 수정 이력 등과 같은 부가적인 정보를 서술한다.

2. Overall Description

2.1. Product Perspective

본 제품은 Python으로 자신의 프로그래밍 실력을 확인하거나 단련하고자 하는 학생들을 위해 개발되었다. Python 언어의 프로그래밍을 공부 중이지만 실제로 문제를 풀 때, 어떤 알고리즘을 적용해야 하는지 알고 싶은 학생들을 주요한 대상으로 한다. 사용자는 이 제품을 통해 기한이 정해진 여러 가지 과제들을 풀면서 자신의 프로그래밍 능력을 확인하거나 향상시킬 수 있다.

2.1.1. System Interfaces

사용자는 웹 브라우저를 이용해 서비스에 접근한다. 또한 사용자의 로그인 기능과 등록 기능은 Django 프레임워크의 기본 기능을 활용해서 구현한다. 또한 Django 기본 내장 데이터 베이스인 SQLite를 활용해서 로그인에 사용되는 정보를 관리한다. 기존에는 Mongo를 사용할 계획이었지만 Django와 충돌하는 경우가 발생했기에 SQLite로 변경한다.

웹 서비스를 제공하기 위하여 Django를 활용하여 구현한다. Django와 React를

활용해서 서비스와 데이터 베이스를 연결한다. 또한 웹페이지의 디자인에 figma를 사용한다. 코딩 테스트를 할 경우에 사용자의 과제에 대한 채점이 많을 것으로 예상되기에 변화가 잦을 경우에 사용하는 React를 채용했다.

코드를 채점하기 위해서 unittest library를 사용한다. Unittest library를 통해 테스트 케이스를 얼마만큼 통과했는지 확인하고 실패한 테스트 케이스가 있으면 오답으로 표시한다. 전부 통과하면 copydetect, multimetric, pylama등을 통해 표절과 효율, 가독성에 대해 채점한다. openAI codex를 통해 코드를 설명한다. diff2HTML으로 정답 코드와 어디가 다른지 표시한다.

2.1.2. User Interfaces

사용자는 임의의 웹 브라우저를 이용해 Python, JavaScript로 작성된 내용물에 접속해서 기능을 쓸 수 있다. 사용자는 로그인, 회원 등록, 과제 선택, 과제 테스트, 과제 제출 등의 주요 기능을 수행할 수 있다. 또한 웹 디자인을 통해 이 기능들을 사용자가 편리하게 사용할 수 있도록 한다.

관리자는 사용자가 코드를 제출하거나 테스트 케이스를 시험했을 때, 테스트 케이스를 통과했는지 여부를 서버의 데이터에 넣어야 한다. 또한 서버에서 바로 채점 프로세스를 실행하기 때문에 다른 가상 환경의 설정은 필요하지 않다. 서버의 보안을 위해서는 SSL/HTTP 기반의 protocol을 활용해 구현한다.

2.1.3. Hardware Interfaces

사용자는 웹 브라우저를 사용할 수 있는 하드웨어를 준비할 필요가 있다. 또한 본 서비스는 IE에 대해서 보안을 신뢰 있게 제공할 수 없으므로 IE 이외의 웹 브라우저가 설치된 하드웨어가 필요하다. 또한 본 서비스는 기본적으로 Chrome을 기준으로 만들어 지므로 Chrome의 시스템 요구사항을 충족해야 한다. Chrome의 시스템 요구사항은 다음과 같다.

- Windows 7, Windows 8, Windows 8.1, Windows 10 이상 버전
- SSE3 을 지원하는 Intel Pentium 4 프로세서 이상 버전

또한 인터넷에 접근하기 위한 NIC 및 웹 브라우저의 GUI와 소통하기 위한

마우스 등의 HID를 포함한다.

관리자의 경우 Django를 이용해 서버를 구축하게 되므로 이하의 조건을 갖추어야 한다.

1. Windows 7 이상 버전(Windows 10이 권장됨)
2. Intel core i3-2340UE 이상의 CPU(i5-4400E이상이 권장됨)
3. 4GB 이상의 RAM(8GB가 권장됨)
4. AMD Radeon R5 M230 이상의 GPU(AMD Radeon R7 A10-7850K이상이 권장됨)
5. 3GB 이상의 HDD

관리자가 React를 사용해서 데이터베이스와 서버를 연결하기 위해서는 이하의 조건을 갖춘 하드웨어가 필요하다.

- Windows 7, Windows 8, Windows 8.1, Windows 10 이상 버전
- JS 를 사용가능한 웹 브라우저를 설치함

2.1.4. Software Interfaces

본 서비스는 Windows 7 이상의 운영체제에서 Chrome을 사용하는 것을 기준으로 한다. 또한 서버에서 Python 인터프리터 3.10.x를 사용하기 때문에 시용자에게도 Python 3.10.x를 사용하는 것을 권장한다.

관리자는 Windows 7 이상의 운영체제에서 Django를 사용해서 서버를 만든다. 사용자의 정보를 처리하기 위해 Django의 기본 데이터베이스 중 하나인 SQLite를 사용한다. 또한 코드의 채점은 서버에서 바로 채점 프로세스를 실행해서 이루어진다.

2.1.5. Communications Interfaces

웹 페이지는 SSL/HTTP 프로토콜을 기반으로 통신한다. 또한 필요하다면 Django의 암호화 라이브러리를 활용하여 서로 간의 통신을 더욱 기밀성이 높도록

할 수 있다.

2.1.6. Memory Constraints

서버의 메모리는 4GB RAM을 최소사양으로 한다. 그러나 권장되는 메모리는 8GB 이상의 RAM이다. 또한 서버는 최소한 2명 이상의 접속을 유지할 수 있어야 한다.

사용자는 Windows 7 이상의 운영체제와 1GB 이상의 RAM을 최소 사양으로 한다. 그러나 서버와 같이 Windows 10 이상의 운영체제와 4GB 이상의 RAM이 권장된다.

2.1.7. Operations

2.1.7.1. System administrator

- 서버
 - ✓ 관리자일 경우 접근한다.
 - ✓ 유저의 정보를 데이터 베이스에 기록한다.
 - ✓ 직접 채점 프로세스를 실행한다.

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.
- 과제 선택
 - ✓ 사용자는 어떤 내용의 과제를 풀지 선택할 수 있다.
 - ✓ 과제의 종류는 Python 기반의 프로그래밍 기술을 기초로 한다.
 - ✓ 선택된 과제에 대한 제약, 상세 등을 표시한다.
 - ✓ 과제에 대한 코드를 작성할 수 있는 창을 표시한다.
- 코드 작성

- ✓ 선택한 과제에 대해 Python으로 과제의 조건에 맞도록 코드를 작성할 수 있다.
- ✓ 코드 작성 중 잘못된 부분을 수정할 수 있다.
- 저장하기
 - ✓ 제한된 횟수 내에서 작성 중인 코드를 저장할 수 있다.
 - ✓ 한 번 저장된 코드는 사용자가 서비스를 종료하고 다시 접속해도 그대로 남아 있다.
- 테스트 케이스 검증
 - ✓ 사용자가 입력한 코드에 대해서 open 테스트 케이스에 대해 검사한다.
 - ✓ 검사 결과 내용을 출력한다.
- 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. 계정을 생성할 경우 ID, PW, email이 필요하다. 등록은 Django의 DB 접속 기능을 활용하여 구현한다. 로그인을 할 경우에는 브라우저가 보내온 Session key를 DB의 내용과 비교하여 DB에 그 key의 내용이 있을 경우에 성공한다. 만약 DB에 session key가 없을 경우에는 로그인이 실패한다.

2.2.2. Select Assignments

로그인 후에 계정은 사용자(User)와 관리자(Admin)로 나뉜다. 관리자 계정을 사용하기 위해서는 관리자 권한을 얻어야 한다. 로그인에 성공한 후에 사용자는 과제를 선택할 수 있다. 또한 과제를 선택하면 과제에 대한 설명, 제약사항, hint 등을 표시하고 과제를 풀 수 있는 창을 띄운다.

관리자는 과제들을 서버에서 직접 생성해서 웹 페이지에 등록한다. 그리고

사용자가 과제를 선택하면 브라우저에서 과제의 번호를 받아 DB에 접속해서 그 번호에 해당되는 과제문을 표시한다. 또한 관리자는 새로운 과제를 등록하거나 존재하는 과제를 삭제할 수도 있다.

과제들은 Python 3.10.X 인터프리터를 기준으로 선정되며 기본적인 코딩 문제부터 심화된 알고리즘까지의 다양한 난이도를 다룬다.

2.2.3. Solve Assignment

User 사용자가 과제를 선택한 후에 코드를 적을 수 있는 인터페이스를 제공한다. 이 인터페이스는 코드를 쓸 수 있는 곳, 저장 기능, open 테스트 케이스의 검증, 제출하기 라는 항목들로 구성되어 있다. 또한 과제에 대해서 종료 시간을 표시하는 인터페이스가 있다. 유저에게 풀이 중, 정답, 오답이라는 상태를 제공해서 자신의 과제 진척 상황을 알 수 있다.

2.2.4. Save Code

유저가 코드를 작성하다 저장하기 인터페이스를 클릭했을 때, 작성 중인 코드를 저장해서 서버의 DB에 등록한다. 로그의 ID를 primary key로 삼고 유저ID와 과제 번호를 foreign key로 삼는다. 그리고 과제의 status를 풀이 중, 맞음, 틀림 세가지 상태 중 하나로 선택해서 등록한다. Save state를 따로 작성하여 유저의 저장 횟수를 검사한다.

유저가 다른 과제를 풀다가 다시 저장한 과제로 돌아왔을 경우나 브라우저에서 나갔다가 다시 로그인 했을 경우에는 로그의 ID를 통해 DB에 접속해서 유저 ID, 과제 번호를 primary key로 삼는 DB에 접속하여 저장된 코드를 표시한다.

2.2.5. Verifying Test Case

유저가 테스트케이스 시험하기를 눌렀을 경우에 open testcase들을 공개하고 유저가 작성한 코드에 대해 어떤 결과가 출력되었는지 표시한다. 이러한 testcase들은 관리자가 서버의 DB에 사전에 등록해 놓아야 한다. 사용자는 이 기능을 통해 과제를 제출하기에 앞서 자신의 코드가 어느 정도로 정확한 것인지 대략적으로 알 수 있다.

2.2.6. Execute & Verifying User Code

실행하기 버튼을 눌러 코드를 실행한다. 만약 에러가 날 경우에 출력값을 제공하고 어느 줄에서 에러가 났는지 표시하고 그 라인을 강조한다. 또한 에러 메시지를 제공한다.

채점하기 버튼을 눌러 유저가 생성한 코드를 모든 테스트 케이스에 대해서 검증한다. Open 테스트 케이스의 fail의 경우에는 테스트케이스의 상세가 제공되지만 close 테스트 케이스의 경우에는 유저는 상세한 사항을 볼 수 없다. 일정한 비율의 테스트 케이스를 통과하면 pass, 그 이외는 fail을 출력한다.

서버는 사용자가 제출한 코드를 수신하고 이 코드에 대해 테스트 케이스의 검증을 실행한다.

2.2.7. Submit User Code & Analysis

제출하기 버튼을 눌러 유저가 생성한 코드를 모든 테스트 케이스에 대해서 검증한다. 그 후 점수를 매긴다. 제출은 최대 n번까지 실행 가능하다. 또한 과거 n번의 코드를 불러올 수 있다. Code Diff를 통하여 채점하고 표절검사, 기능, 효율성, 가독성의 각각의 분야에서 채점한다. Open AI Codex API를 활용하여 코드에 대해 설명하고 과제에 관한 관련자료들을 표시한다.

2.3. User Characteristics

2.3.1. Administrator

관리자는 본 서비스가 제공하는 Python 코딩 웹사이트에 과제들을 추가하고 서버를 관리하는 사람을 의미한다. 관리자는 서버에 과제를 추가하고 그와 관련된 테스트 케이스들을 등록할 수 있다. 또한 DB를 통해 자료를 검색해 과제와 관련된 추천 시스템을 사용할 수 있어야 한다. 따라서 관리자는 Django, React, SQLite에 능숙한 사람이어야 한다. 또한 서버 자체를 관리하기 위해 서버 관리에 대한 지식을 갖추어야 한다.

그 외에 본 서비스가 Python을 기반으로 동작하기에 Python에 정통한 사람이어야 한다.

2.3.2. User

유저는 본 서비스에 접근하여 자신의 Python 프로그래밍 기술을 향상시키거나 확인할 사람을 의미한다. 유저는 자신이 선택한 한국어로 작성된 과제를 통해 그와 관련된 프로그래밍 기술을 익힐 수 있고 그 알고리즘에 따라 Python 프로그램을 직접 만들어 볼 수 있다. 유저는 Windows와 웹 브라우저의 사용을 할 수 있으며 본 서비스의 각 기능을 원활히 사용할 수 있어야 한다. 또한 사용자가 Python을 설치할 필요는 없으나 기본적인 Python의 기본적인 지식은 있어야 한다. 다시 말해 본 서비스를 사용하기 위해 유저는 한국어 능력, Python의 기초, 웹 브라우저 사용을 기본적으로 숙지해야 한다.

2.3.3. Ministry of Education

교육부는 본 서비스의 제공에 대해 유저들에게 어떤 과제를 제공하는지에 대해 심사한다. 만약 본 서비스가 제공하는 과제가 사회의 미풍양속상 용납할 수 없는 것이라면 교육부는 본 제품의 시장유통을 거부할 수 있다. 교육부는 서버나 프로그래밍에 대한 지식이 필요하지 않으며 이러한 내용은 관리자가 모든 권한을 가진다. 그러나 교육부는 현행법령에 대해 상세히 숙지하여 본 서비스의 평가시 오판을 내리지 않도록 해야 한다.

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
- ✓ 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 입력 값을 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 해싱을 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.

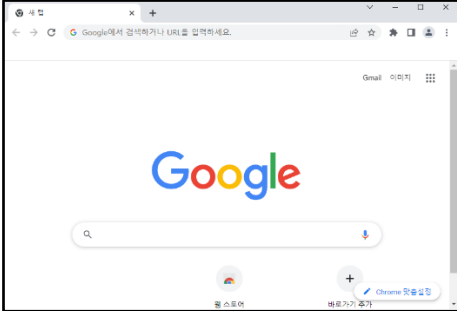
또한 서버의 Ubuntu 환경과 NodeJS 등 외부 API가 서로 충돌하지 않는다고 가정한다. 이 때, 각 API 제품은 서로 충돌하지 않도록 설계되어 배포되었기 때문에 운영체제와의 호환성만을 고려한다. 각 제품의 버전은 Ubuntu 20.04 환경에서 제공하는 최신 버전에 대해 모든 Backward Compatibility를 제공해야 한다. 운영체제를 제외한 각 API의 최신 버전 기준은 2022년 10월 30일을 기준으로 한다.

3. Specific Requirements

3.1. External Interface Requirements

3.1.1. User Interfaces

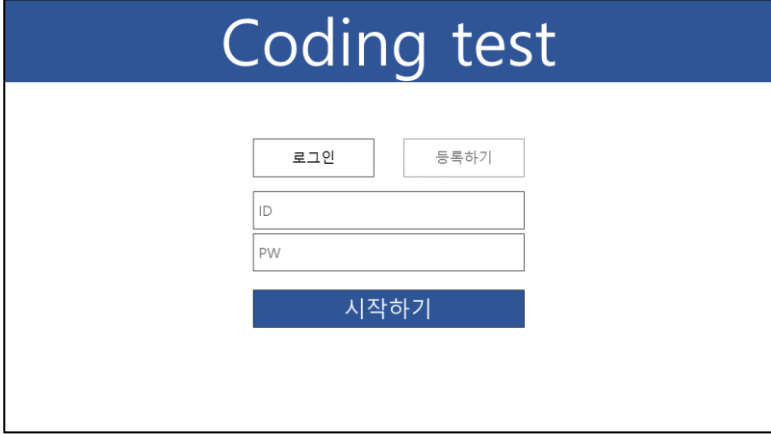
[표 3] 웹 서비스에 대한 사용자의 GUI 기반 접근

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 ECMAScript 표준에 의한 JavaScript 처리 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

이름	웹 서비스에 대한 사용자의 GUI 기반 접근
	<p>[그림 2] 웹 브라우저 HTML 렌더링 예시</p> 
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 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	아래의 화면과 같은 형태를 띤다. *세부적인 사항은 상기 [표 3]의 웹 서비스 GUI에 상속됨

이름	사용자 GUI 1. 로그인 및 등록 화면
	<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

이름	사용자 GUI 2. 로그인 오류
화면의 형태	<p>아래의 화면과 같은 형태를 띤다. *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

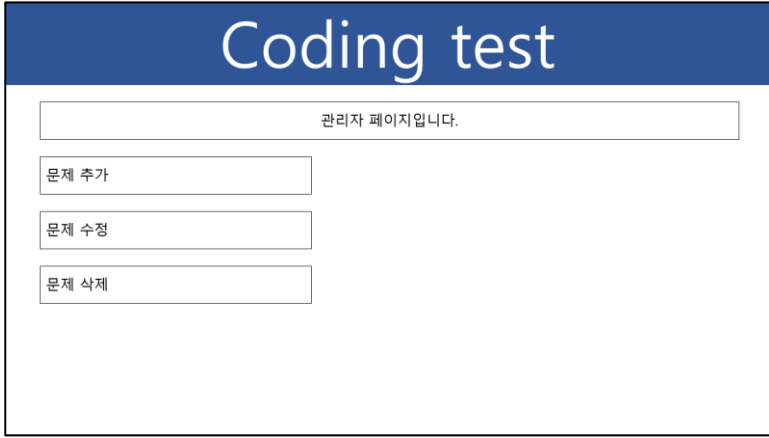
[표 6] 사용자 GUI (3) 등록 오류

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	<p>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.</p> <p>발생 가능한 오류의 종류는 다음과 같다.</p> <ul style="list-style-type: none"> ✓ 이미 존재하는 계정명 ✓ 이미 등록된 이메일 <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 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

이름	사용자 GUI 3. 등록 오류
화면의 형태	<p>아래의 화면과 같은 형태를 띤다. *세부적인 사항은 상기 [표 3]의 웹 서비스 GUI에 상속됨</p> <p>[그림 5] 등록 화면 예시</p>  <p>* 에러 화면은 GUI 2. 로그인 오류와 동일한 팝업 형태임</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

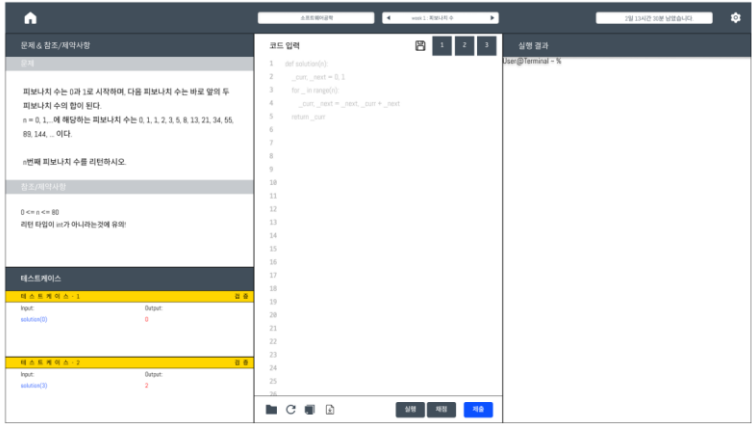
[표 7] 사용자 GUI (4) 관리자 페이지

이름	사용자 GUI 4. 관리자 페이지
목적 및 설명	상기 사용자 GUI 1. 로그인 및 등록 메인 화면에서 사용자가 서버에 요청한 등록 HTTP POST의 처리 결과가 정상이고 해당 계정이 관리자 계정이었을 경우 발생한다. 이 화면에서는 메인 페이지에 나타날 문제를 추가하고 수정하고 삭제하는 기능이 있다.
입출력 형태	<p>서버로부터 수신한 HTML, CSS, JavaScript에 근거하여 페이지를 렌더링한다. 관리자 페이지를 의미함과 동시에 문제 추가, 수정, 삭제 기능을 수행하는 각각의 버튼이 나타난다.</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. *세부적인 사항은 상기 [표 3]의 웹 서비스 GUI에 상속됨</p>

이름	사용자 GUI 4. 관리자 페이지
	<p>[그림 6] 관리자 페이지 접근 예시</p> 
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) 문제 풀이 페이지

이름	사용자 GUI 5. 문제 풀이 페이지
목적 및 설명	<p>상기 사용자 GUI 1. 로그인 및 등록 화면에서 관리자가 아닌 계정으로 로그인에 성공했을 때 나타나는 페이지이다. 사용자가 등록된 문제를 풀고 그 결과를 확인할 수 있는 페이지이다. 총 6가지 섹션으로 나눌 수 있으며 각각은 헤드 섹션, 문제 설명 섹션, 테스트 케이스 섹션, 코드 에디터 섹션, 각종 기능 세션, 결과 섹션이라고 칭한다.</p> <p>헤드 섹션에서는 홈버튼, 강의명, 문제 이동 버튼, 마감일, 설정 버튼이 존재한다.</p> <p>문제 설명 섹션에는 선택된 문제의 설명과 참조/제약사항이 나타난다.</p> <p>테스트 케이스 섹션에는 공개된 테스트 케이스의 내용과 해당 케이스를 사용자가 작성한 코드가 통과 여부를 알리는 상태가 나타난다.</p> <p>코드 에디터 섹션에서는 현재 시점까지 작성한 코드를 저장하는 버튼이 존재하며 최대 3번 저장할 수 있으며 코드를 작성할 수 있는 영역이 존재한다.</p> <p>각종 기능 세션에는 파일 불러오기, 코드 초기화, 코드 복사, 코드 다운로드, 실행, 채점, 제출 버튼이 있다. 실행은 코드를 컴파일하고 실행하는 것이고 채점은 문제의 모든 테스트 케이스에 대한 채점을 진행하고 알려준다. 이때 모든 테스트 케이스는 공개 테스트 케이스와 비공개 테스트 케이스를 합친 것을 의미한다. 제출 버튼은 코드의 표절률, 채점 결과, 효율 점수, 가독성 점수를 표시한다.</p>
입출력 형태	서버로부터 수신한 HTML, CSS, JavaScript에 근거하여 페이지를 렌더링한다. 문제를 선택할 수 있는 화살표와 문제의 내용, 코드를 작성할 수 있는 영역, 작성된 코드를 제출했을 때 결과가 나타나는 영역이 있다.

이름	사용자 GUI 5. 문제 풀이 페이지
범위/정확도/ 오차범위	*상기 [표 4]의 웹 서비스 GUI 로그인 및 등록 화면과 동일함
단위	문제 이동 버튼, 코드 저장 버튼, 파일 불러오기 버튼, 코드 초기화 버튼, 파일 다운로드 버튼, 실행 버튼, 채점 버튼, 제출 버튼
시간/ 속도	인터페이스 표기: 상기 [표 3]의 웹 서비스 GUI에 상속됨
다른 입출력 간 연관성	*상기 [표 3]의 웹 서비스 GUI에 상속됨
화면의 형태	<p>아래의 화면과 같은 형태를 띤다. *세부적인 사항은 상기 [표 3]의 웹 서비스 GUI에 상속됨</p> <p>[그림 7] 문제 풀이 페이지 예시</p> 
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

3.1.2. Hardware Interfaces

[표 9] 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 Card Driver

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

3.1.3. Software Interfaces

[표 10] 서비스 서버 소프트웨어 인터페이스

이름	서버 구축에 필요한 소프트웨어 인터페이스	
목적 및 설명	사용자에게 로그인 및 등록을 처리하며 코딩 테스트 문제 제공, 자동 채점, 결과 분석 그리고 문제와 관련된 자료를 제공하는 전 과정을 처리하기 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
Django 4.1.2	설명	Python 기반 웹 프레임워크
	입력	시스템을 연결시키는 Python 파일
	출력	웹 어플리케이션
React 18.2.0	설명	유저 인터페이스 설계를 위한 JavaScript 라이브러리
	입력	JavaScript 코드
	출력	외부 접근 가능한 HTML, CSS, JavaScript 등 렌더링 자원
Unittest	설명	테스트 케이스 채점을 수행하는 라이브러리
	입력	사용자의 제출 코드
	출력	테스트 케이스 채점 결과
copydetect 0.4.2	설명	사용자의 제출 코드의 표절도를 계산하는 라이브러리
	입력	사용자의 제출 코드와 정답 코드
	출력	제출 코드의 표절도(0~1 사이의 실수)
multimetric 1.3.0	설명	사용자의 제출 코드의 효율성을 채점하는 라이브러리
	입력	사용자의 제출 코드
	출력	4가지 효율성 점수
pylama 8.4.1	설명	사용자의 제출 코드의 가독성을 채점하는 라이브러리
	입력	사용자의 제출 코드
	출력	5가지 가독성 점수

이름	서버 구축에 필요한 소프트웨어 인터페이스	
openAI codex	설명	AI가 사용자의 제출 코드를 설명해주는 라이브러리
	입력	사용자의 제출 코드
	출력	AI가 생성한 코드 설명
diff2HTML 3.4.20	설명	사용자의 제출 코드와 정답 코드의 차이를 구하는 라이브러리
	입력	사용자의 제출 코드와 정답 코드
	출력	두 코드의 diff에 대한 HTML 코드
docker 20.10.11~	설명	사용자의 실습 코드를 실제로 실행하기 위해 사용
	입력	사용자의 실습 코드
	출력	실습 코드 수행 결과
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 11] 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	설명	사용자가 Python, SSH를 사용하기 위한 Windows CLI
	입력	사용자 운영체제 내 TTY

이름	서비스 이용을 위한 사용자 소프트웨어 인터페이스	
Latest Ver.	출력	명령 및 프로그램 실행 결과

* ※표기된 소프트웨어 인터페이스는 필수 요소가 아니다.

3.1.4. Communication Interfaces

[표 12] 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 13] 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 메인 페이지로 이동
인코딩	Base64인코딩 및 전송

[표 14] 코드 실행 HTTP 요청 형태

이름	코드 실행 HTTP 요청
용도	사용자가 제출한 코드의 실행 결과를 요청
Method	POST
URL	main.com/problem*/run
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 15] 코드 채점 HTTP 요청 형태

이름	코드 실행 HTTP 요청
용도	사용자가 제출한 코드의 채점 결과를 요청
Method	POST
URL	main.com/problem*/check
Content Type	text/application
내용	사용자가 작성한 코드 내용 일체 브라우저 쿠키(로그인 과정에서 생성된 세션값)
서버 동작	사용자의 코드를Docker 환경에서 실행 실행 결과 및 에러 발생 여부를 응답
인코딩	Base64인코딩 및 전송

[표 16] 코드 제출 HTTP 요청 형태

이름	코드 실행 HTTP 요청
용도	사용자가 제출한 코드의 제출 결과를 요청
Method	POST
URL	main.com/problem*/submit
Content Type	text/application
내용	사용자가 작성한 코드 내용 일체 브라우저 쿠키(로그인 과정에서 생성된 세션값)
서버 동작	사용자의 코드를Docker 환경에서 실행 실행 결과 및 에러 발생 여부를 응답

이름	코드 실행 HTTP 요청
인코딩	Base64인코딩 및 전송

[표 17] 일반적인 페이지 렌더링 자원 HTTP 요청 형태

이름	계정 등록 HTTP 요청
용도	페이지 간 이동 및 자료 제공
Method	GET
URL	main.com/*
Content Type	HTML, CSS, JavaScript
내용	페이지 렌더링을 위한 CSS, JavaScript, HTML 등
서버 동작	사용자가 요청한 페이지에 해당하는 자료 전달
인코딩	별도 인코딩 없음

※URL의 *은 Wildcard를 의미한다.

3.2. Functional Requirements

3.2.1. Use Case

[표 18] 사용례 (1) 계정 등록

이름	계정 등록 (Registration)
행위자(Actor)	사용자 (User)
설명	본 서비스 이용하기 전에 사용자는 계정을 만들어야한다. 계정은 email, user ID, password 정보를 받아 생성할 수 있다.
정상 흐름	<ol style="list-style-type: none"> ① 사용자가 생성을 원하는 email, password, user ID 입력한다. ② password 재입력 요구한다. 이때 브라우저 JavaScript의 localStorage에 저장된 1번에 입력한 password와 비교해서 다를 경우 재입력 요청한다. ③ user ID, email을 HTTP POST 요청 보낸 후 SQLite 데이터베이스에 중복 검증한다. ④ 중복 없을 경우 새로운 계정이 정상적으로 생성되며 데이터베이스에도 추가된다. ⑤ User ID나 email 중복되었을 경우 다른 ID, email을 사용하라는 오류 메시지를 출력한다.

이름	계정 등록 (Registration)
전제 조건	사용자는 본 서비스 처음 접근한다. 동일한 email로 전에 등록한 계정이 없다.
사후 조건	계정 생성 시 user정보가 DB에 추가된다.
가정	없음

[표 19] 사용례 (2) 로그인

이름	로그인 (Login)
행위자(Actor)	사용자 (User)
설명	사용자는 전 단계에서 만든 계정으로 로그인 후 서비스를 이용할 수 있다. 웹사이트 로그인 창에 정보를 입력한 후, 서버에서 검증한다. 이후로 로그인 성공 여부를 User에 알려준다.
정상 흐름	<ul style="list-style-type: none"> ① 사용자는 위에서 등록한 User ID, password 입력한다. ② User ID, password로 해시값을 만들어 HTTP Post 요청 통해 데이터베이스에 있는 해시값이랑 비교한다. ③ 데이터베이스에 있는 값이랑 동일한 경우 성공적으로 로그인하여 문제 선택 페이지로 redirection 된다. ④ 동일하지 않을 경우 로그인 재시도 alert 메시지를 보낸다.
전제 조건	사용자는 사전에 등록한 계정이 있다.
사후 조건	로그인 성공 시 Cookie 수정해서 세션과 계정을 결부한다.
가정	없음

[표 20] 사용례 (3) 문제 선택, 풀이, 제출

이름	문제 선택, 풀이, 제출 (Problem Selection, Solving, Submission)
행위자(Actor)	사용자 (User)
설명	사용자는 여러 개의 문제 중에 하나를 선택하여 문제풀이를 진행한다.
정상 흐름	<ul style="list-style-type: none"> ① 사용자는 문제 선택 Main Page에서 풀고 싶은 문제를 선택한다. 이때 Main Page에 있는 문제들은 난이도별로 분리되어 있다. ② 문제 선택하면 코드 작성할 수 있는 In-browser code editor 페이지로 redirection받아서 문제 풀기 시작할 수 있다.

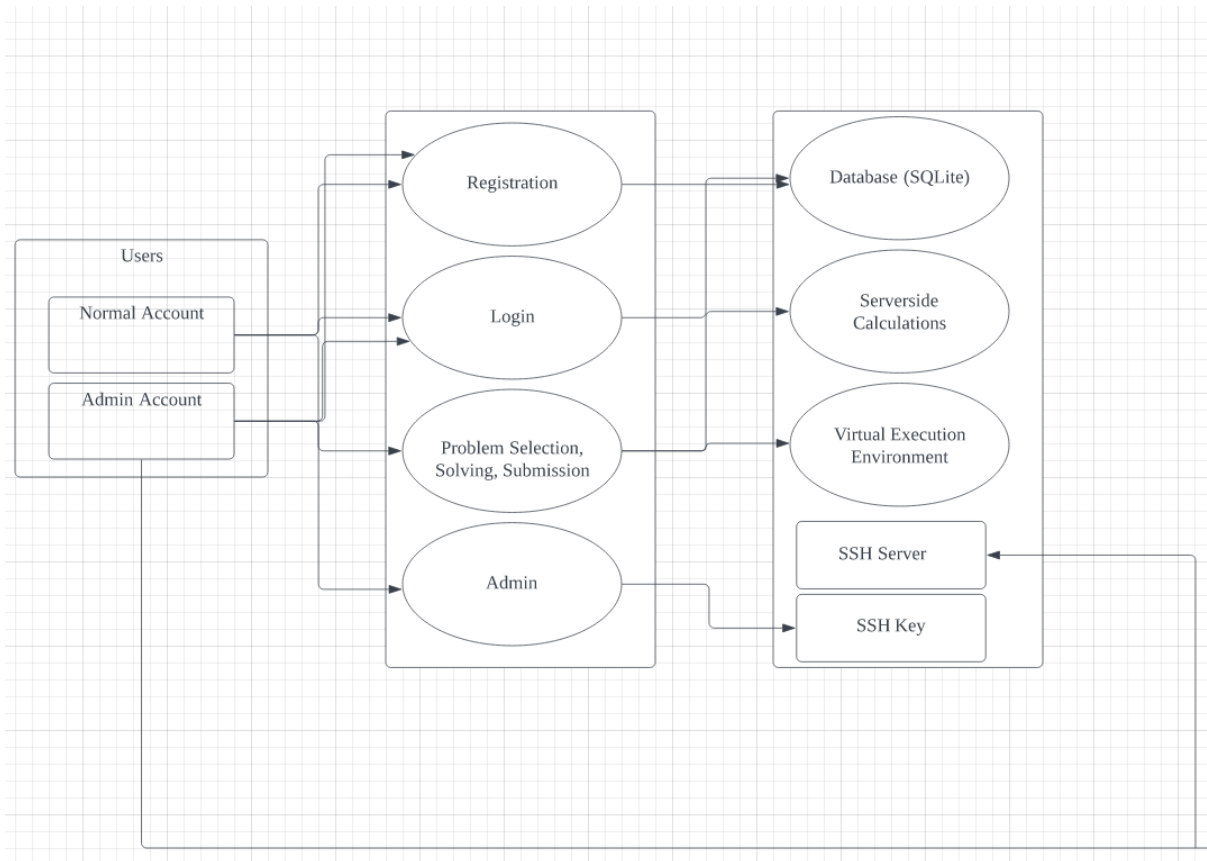
이름	문제 선택, 풀이, 제출 (Problem Selection, Solving, Submission)
	③ 문제의 요구 조건에 맞춰서 코드를 작성한다. ④ 작성한 코드를 최종 제출하면 서버에 전달된다. ⑤ 서버에서 코드를 수행하여 코드의 정확도, 효율성을 평가해서 이에 대한 피드백을 사용자에게 전달한다.
전제 조건	사용자가 성공적으로 로그인한 후 문제를 골랐다.
사후 조건	코드 제출 버튼 누르면 서버에서 채점이 진행되며 사용자에게 채점결과를 보여준다.
가정	없음

[표 21] 사용례 (4) 관리자 페이지

이름	관리자 페이지
행위자(Actor)	서비스 관리자 (Admin)
설명	서비스 관리자는 웹사이트의 여러가지 관리하는 작업을 위해 특별한 Admin 권한이 부여된다. 이때 서비스 관리자는 Registration 단계로 통해서 생성되는 계정이 아니라, 웹사이트, 혹은 다른 관리자의 확인 후 별도로 만들어주는 계정이다. 그러나 서버에 접근하려면 SSH key를 로그인 후 얻어야 한다.
정상 흐름	① 서비스 관리자가 관리자 계정으로 로그인 한다. ② 관리자들 만 접근할 수 있는 페이지 따로 있으며, 여기서 관리자 공지사항을 확인하거나 SSH key를 다운로드 할 수 있다.
전제 조건	서비스 관리자 계정 성공적으로 로그인 후, SSH 키 다운로드해서 서버를 접근할 수 있다.
사후 조건	SSH key 외부에 유출되지 않도록 관리해야 한다.
가정	없음

3.2.2. Use Case Diagram

[그림 8] 사용례 도식



3.2.3. Data Dictionary

[표 22] 데이터베이스 'User' 테이블의 구조

Field	Key	Constraint	Description
id	PK	Not Null	User ID
h_password		Not Null	Password Hash Value
email		Not Null	Email
admin		Boolean	Initialize False

* 새로운 사용자가 등록될 때 본 테이블에 값을 추가해야 한다.

[표 23] 데이터베이스 'Status' 테이블의 구조

Field	Key	Constraint	Description
status_ID	PK	Not Null	Completion status for Problem
incomplete		Boolean	
correct		Boolean	
incorrect		Boolean	

[표 24] 데이터베이스 'Save Status' 테이블의 구조

Field	Key	Constraint	Description
save_ID	PK	Not Null	ID of the problem and save state
save_content		Not Null	Content of the problem at the time of save

[표 25] 데이터베이스 'Problem' 테이블의 구조

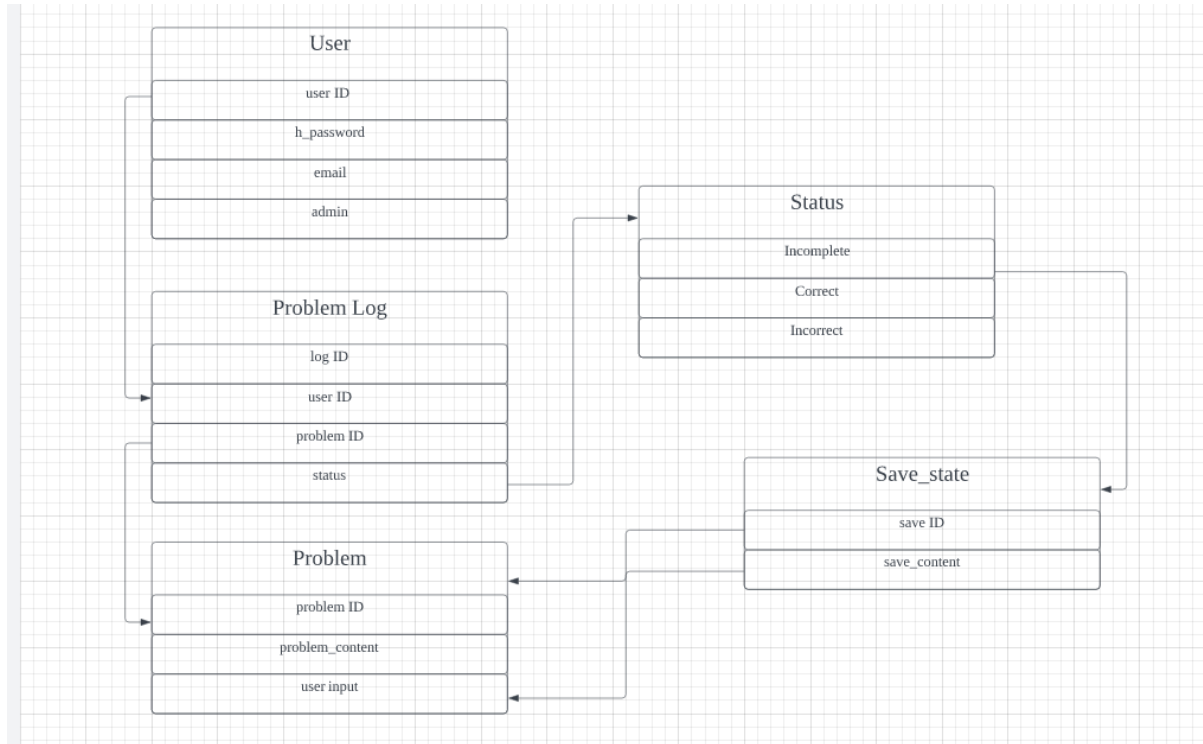
Field	Key	Constraint	Description
problem_id	PK	Not Null	Problem ID
problem_content		Not Null	Problem Content
user_input		Not Null	User input

[표 26] 데이터베이스 'Problem Log' 테이블의 구조

Field	Key	Constraint	Description
log_id	PK	Not Null	Log ID
user_id	FK	Not Null	user ID
problem_id	FK	Not Null	problem ID
status		Not Null	status of the question

3.2.4. Data Flow Diagram

[그림 9] 데이터 흐름 도식



3.3. Performance Requirements

다음 항목들은 본 서비스가 제공해야 할 성능의 요구조건을 나타낸다.

3.3.1. Static numerical requirement

- ✓ 사용자의 코드 검증 조건 중 하나는 X초의 실행 제한시간을 가지며 그 시간을 벗어나면 시간 초과 실행 Alert 사용자에게 보낸다.

3.3.2. Dynamic numerical requirement

- ✓ 본 시스템은 여러 명의 관리자가 관리할 수 있는 SSH 공개키를 이용하여 동시 작업이 가능하다.
- ✓ 데이터베이스 접근, 탐색 및 수정하지 않는 동작들은 3초 이내로 수행되는 걸로 목표로 잡는다. 경우에 따라 이 조건을 만족하기 힘들 경우, 5초 이내로 동작을 수행하는 것을 목표로 잡는다.

- ✓ 사용자 save state을 불러오는 동작은 5초 이내에 수행되어야 한다.
- ✓ 코드 채점 및 구체적인 피드백(효율성, 정확도)에 관련된 동작들은 (즉 서버 및 데이터베이스 접근, 탐색, 수정 동작) 10초 이내 수행되어야 하며, 사용자에게 관련 데이터는 15초내로 전송되어야 한다.

3.4. Logical Database Requirements

본 시스템은 SQLite 데이터베이스를 사용한다. 사용자 정보, 강의 정보, 문제 풀이 진행 상태, 각 문제의 save state을 데이터베이스에 저장한다. 사용자가 추가될 때마다 사용자 테이블에 새로운 레이블이 생성되며 문제도 똑같이 문제 테이블에 추가된다. 사용자는 문제 진행상태를 수동으로 save 버튼 통해 저장할 수 있으며, 그때까지 작성한 코드가 데이터베이스에 저장된다. 추후에 사용자가 다음 세션때 로그인해서 같은 문제를 풀려고 할때 저장한 데이터를 불러올 수 있다. 새 세션 만들지 않아도, 사용자가 저장한 save state을 복구할 기능이 추가될 예정이다.

3.5. Design Constraints

JavaScript기반의 웹 서비스를 제공하며 디자인 관리는 서버는 사용자에게 Problem solving 및 Coding Test를 위한 문제, 참조/제약사항, 테스트 케이스를 제공한다. 사용자가 입력한 코드를 서버가 채점하고 결과를 반환할 수 있으며 그 결과에는 테스트케이스에 대한 검증, 숨겨진 테스트 케이스에 대한 채점 결과, 관련 자료 추천, 포절 검사, 효율 채점 등을 포함한다. 웹 프레임워크로 Django, React, SQLite를 사용한다.

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) 3 개로 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 인터페이스는 코딩테스트와 관련한 경험이 충분하지 않은 사람일지라도 직관적으로 이해하기 쉽고 사용하기 쉽도록 설계되어야 한다. 헤드 섹션, 문제 설명 섹션, 테스트 케이스 섹션, 각종 기능 버튼 섹션, 그리고 결과 섹션 총 6개의 섹션의 가시성을 각각 명확히 하고 각 섹션이 무슨 기능을 하는지 직관적으로 이해하기 쉽도록 현재 문제 풀이에 범용적으로 많이 사용되고 있는 백준(<https://www.acmicpc.net>), Codecademy(<https://www.codecademy.com>), 프로그래머스(<https://programmers.co.kr>), 온코드(<https://www.oncoder.com/>)와 유사한 인터페이스를 설계한다. 시스템 관리자는 사용자가 코딩테스트가 처음일지라도 문제를 읽고 코드를 작성한 후 사용자가 필요한 채점 결과들을 손 쉽게 볼 수 있도록 해야 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. 정확한 Performance와 동시 접속 client 수의 타협점

은 추후에 정하지만, 만일 더 큰 규모로 서비스를 확장할 경우 서버는 하드웨어, 소프트웨어 측면에서 개선이 필요할 수 있다. 특히 채점서버를 확장해야 할 필요성이 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. 웹 페이지에서 특정 버튼을 눌렀을 때 Back-End까지 도달하는데 5초, Back-End에 저장된 자료를 요청하거나 수정하는 등의 명령에 대해서는 10초 이상의 지연이 발생하지 않도록 설계한다.

3.7.1.3. Security Requirements

본 서비스 이용을 위해 사용자는 로그인을 해야하고 유저의 ID와 비밀번호를 포함하여 각 유저의 작성된 코드가 서버에 저장될 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, 코딩 테스트를 위해 사용자가 서버에 전송하는 Python 코드는 Sandbox 환경에서 실행한다. 이 환경은 Docker 프로그램을 이용해 쉽게 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 코드의 작성 및 채점 뿐만 아니라 표절 검사, 기능 채점, 효율 채점, 가독성 채점등의 기능의 존재를 알고 사용할 수 있어야한다.

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 해싱을 거쳤기 때문에, 레인보우 공격으로부터 안전하다.
- ✓ 이 시스템은 세션 및 브라우저 쿠키를 바탕으로 사용자를 구분한다. 브라우저 쿠키는 외부로 유출되지 않아야 한다.
- ✓ 사용자가 서버로 보내는 모든 코드는 실행할 때 Docker등의 완전히 격리된 환경에서 수행되어야 한다.

3.7.3.2. Regulatory Requirement

- ✓ 본 시스템에서 이용하는 Docker, NodeJS, Django, React 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.

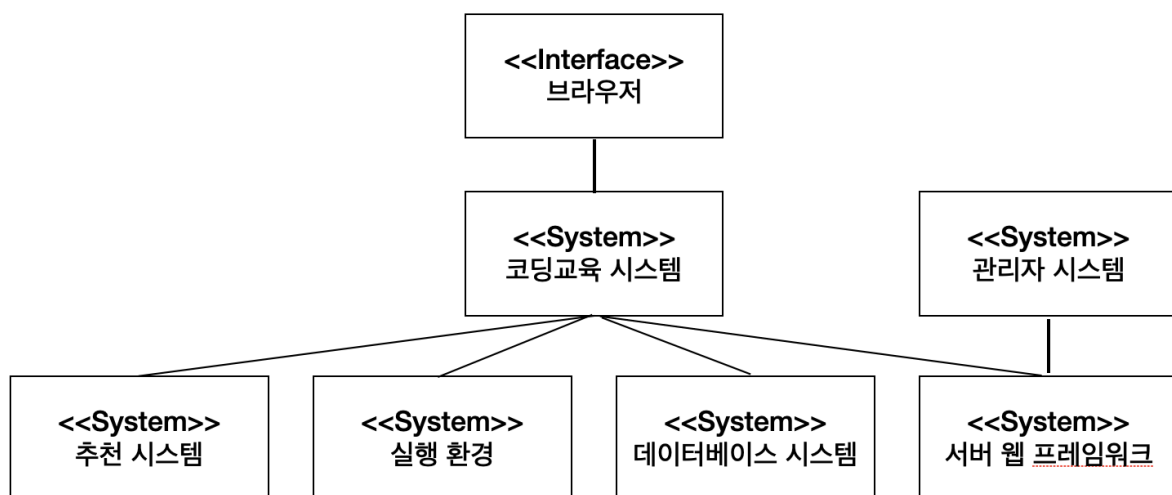
3.8. Organizing the Specific Requirements

우리는 시스템을 Unified Modeling Language(UML)을 이용하여 시각적인 도표로 나타내고자 한다. 시스템과 시스템 내부의 서브 시스템, 각 요소와 외부 환경 등 요구사항을 명확하게 표현할 수 있을 것이다.

이러한 모델은 시스템, 서브 시스템, 시스템 내의 각 요소와 운영 환경 등 세부적인 요구사항의 형태를 명확하게 나타낼 수 있다.

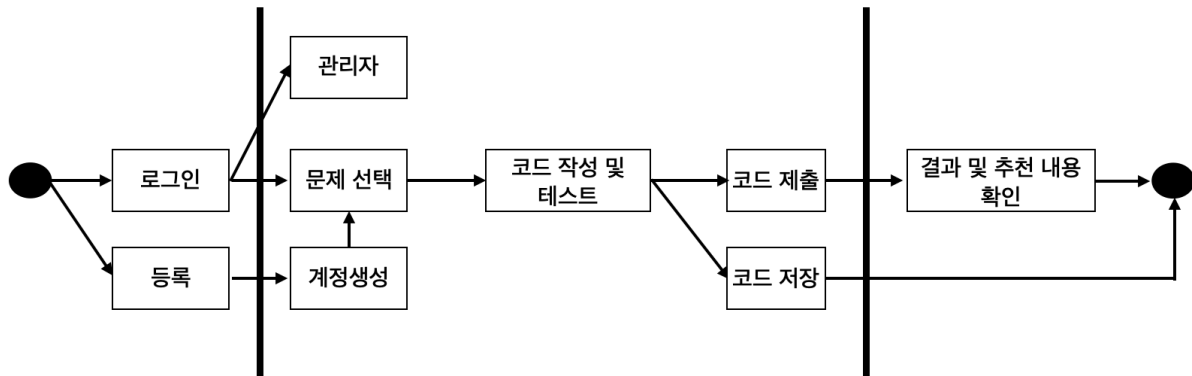
3.8.1. Context Model

[그림 10] 맥락 모델



3.8.2. Process Model

[그림 11] 프로세스 모델



3.8.3. Interaction Model

3.2.2 항목의 User Case Diagram 형태로 표현되었다.

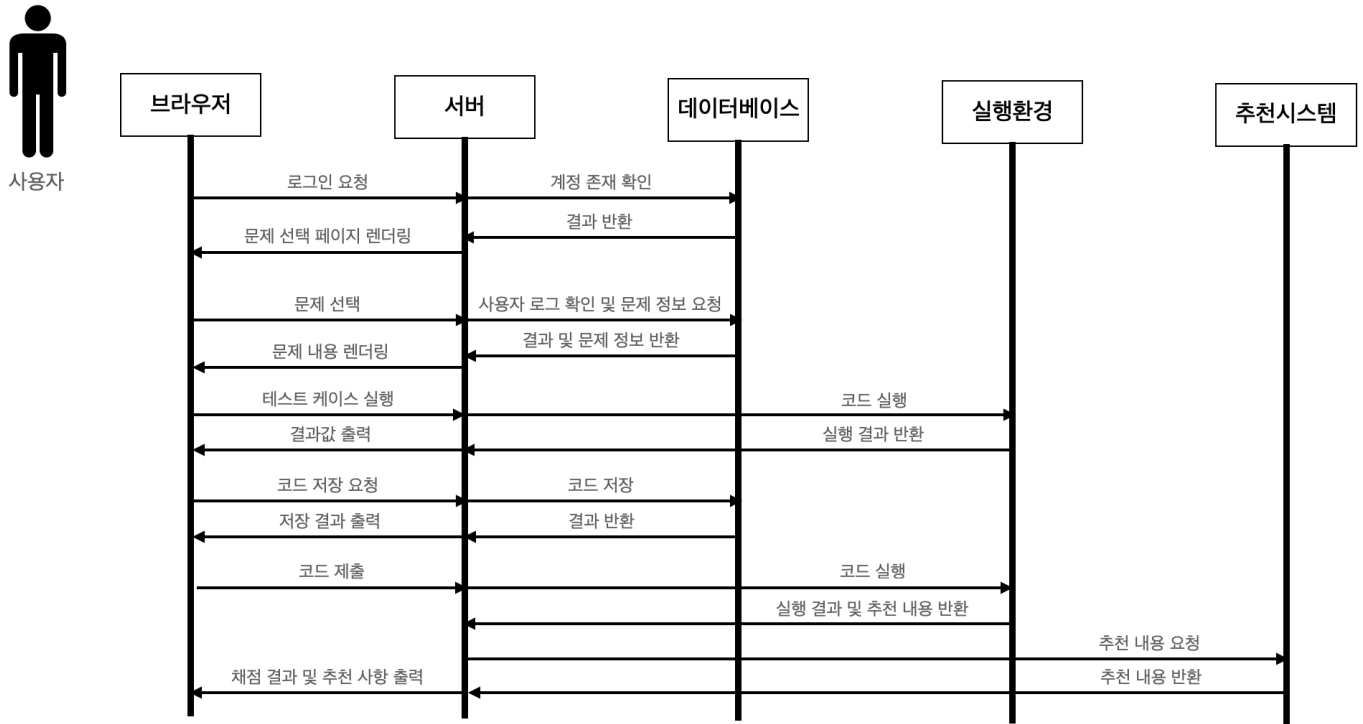
3.8.4. Behavior Model

3.8.4.1. Data Flow Diagram

3.2.4 항목에 Data Flow Diagram 형태로 표현되었다.

3.8.4.2. Sequence Diagram

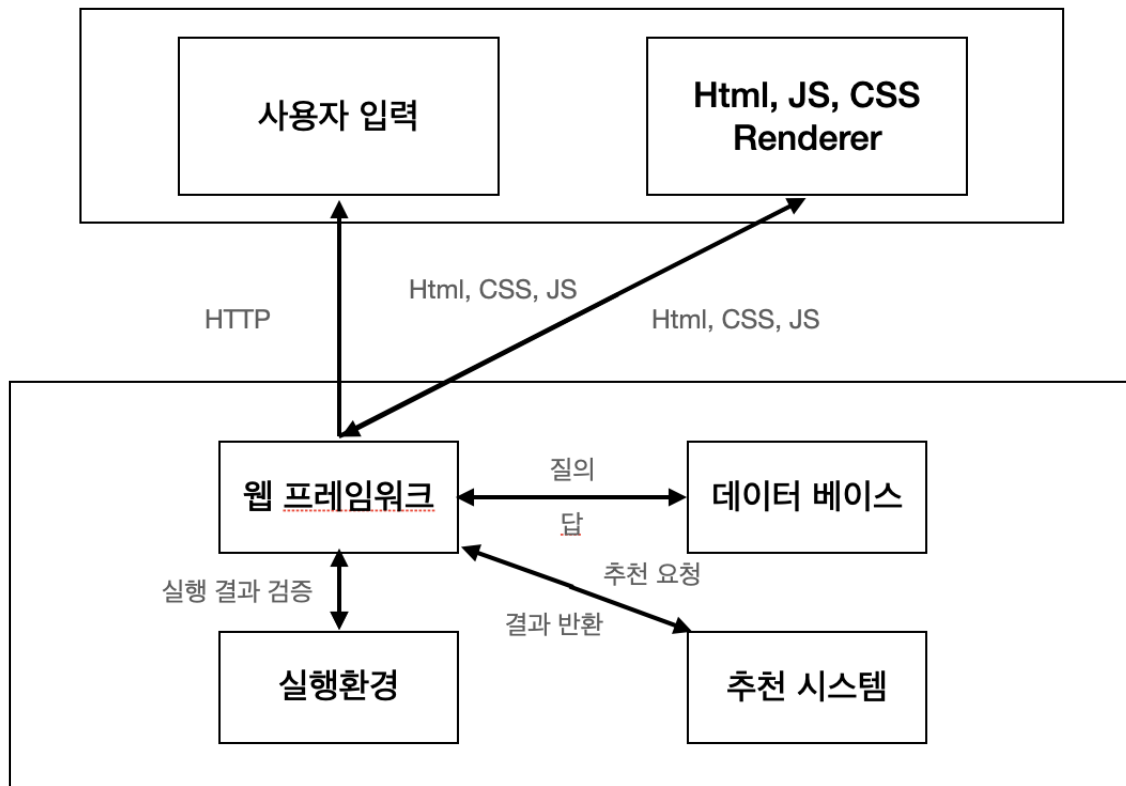
[그림 12] Sequence Diagram



3.9. System Architecture

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

[그림 13] 시스템 구조



3.10. System Evolution

본 항목은 시스템 출시 후 기술 동향의 변화, 새로운 요구사항의 발생 및 수정 필요성의 발생 시의 대처에 대한 입장을 서술한다. 이 항목은 추후 본 시스템의 방향성을 서술하는 것이기 때문에 시스템을 수정하게 될 경우 이를 고려하여야 한다.

3.10.1. Limitation and Assumption

본 서비스는 브라우저 환경에서 Python 프로그래밍 문제에 대한 답안을 작성하고, 그 결과와 연관된 추천 자료를 보고 학습할 수 있는 환경을 제공한다. 출시 시점에는 ‘코드 작성 및 채점’, ‘문제에 대한 추가자료 추천’ 및 ‘실시간 통신 구현의 대분류’를 가진다. 추후 협의를 통해 새로운 대분류를 추가할 수 있다.

본 서비스는 사용자가 제출한 Python 코드가 심각한 문제를 야기할 가능성은 고려하지 않았다. 끝나지 않는 반복문, 과도한 메모리 사용과 접근 권한과 관련된 문제 등에 대한 방지책만 고려할 것이며, 이후 실행 환경을 격리할 필요성이 발생할 경우 빠르게 대응하여 수정하여야 한다.

본 서비스는 계정 당 서버에 저장할 수 있는 코드의 수를 제한하였으며 한 계정에서 동시에 다수의 코드 검증을 실행하는 것은 허용하지 않는다. 추후 하드웨어 환경이 개선될 경우 저장 가능한 코드의 상한이 증가할 수 있다.

3.10.2. Evolutions of Hardware and Change of User Requirements

3.10.1 항목에서 기술한 바와 같이 사용자의 제출 코드 실행 시 추가적인 보안이 필요하다고 판단될 경우, docker와 같은 방법으로 실행환경을 격리할 수 있다. 이 경우 격리된 실행 환경과 기존 시스템 사이에 통신이 원활한지 유의해야 한다.

본 서비스는 예상 동시 접속자의 수는 100명 이하로 가정하였다. 동시 접속자의 수가 일정 수준 이상으로 증가할 경우, 병목 현상으로 인해 서비스의 제공이 원활하지 않을 수 있다.

이를 해결하기 위해 서버를 추가하여 로그인 시점에서 각 서버의 부하를 조절하거나, 서버 하드웨어의 변경이 필요할 수 있다. 이 경우 큰 추가비용이 발생하기 때문에 관리자 및 설계자는 이 시점을 적절히 파악하여야 한다.

기능의 수정, 변경이 아니라 문제 추가와 같이 단순 콘텐츠의 수정, 추가라면 시스템 관리자 기능을 통하여 수정, 추가할 수 있다. 기능을 수정, 추가하는 경우 본 요구사항 명세서의 내용들을 기반으로 새롭게 설계할 수 있다. 단 기존 요구사항 명세와 동일한 양식을 가져야 하며, 본 문서 또한 수정되어야 한다.

4. Supporting Information

4.1. Software Requirement Specification

해당 요구사항 명세서는 IEEE 권장사항에 맞추어 작성되었다. 이후 본 서비스의 개발에 가장 적합한 형태로 요구 사항 조건들을 작성하기 위해 본래 양식에서 일부 수정되었다. (IEEE Recommended Practice for Software Requirements Specifications, IEEE-Std-830)

4.2. Document History

[표 27] 문서 이력

Date	Version	Description	Writer
2022.10.25.	1.0	회의 후 최초 작성	박건우 외 4명