



Online Coding Test Web Platform

2022.10.30

소프트웨어공학개론
TEAM 7

Team Leader	김호진
Team Member	김양선
Team Member	변영민
Team Member	이주빈
Team Member	임세아
Team Member	전윤태

CONTENTS

1. INTRODUCTION	7
1.1. PURPOSE	7
1.2. SCOPE.....	7
1.3. DEFINITION, ACRONYMS, AND ABBREVIATION.....	7
1.4. REFERENCES	8
1.5. OVERVIEW.....	8
2. OVERALL DESCRIPTION.....	9
2.1. PRODUCT PERSPECTIVE	9
2.2. PRODUCT FUNCTIONS	9
2.2.1. HOME PAGE	11
2.2.2. LOGIN/START	11
2.2.3. LECTURE/ASSIGNMENT SELECT	11
2.2.4. SOLVING PAGE	11
2.2.4.1. HEAD SECTION	11
2.2.4.2. PROBLEM SECTION.....	11
2.2.4.3. TEST CASE SECTION	12
2.2.4.4. CODE EDITOR SECTION	12
2.2.4.5. FUNCTION SECTION.....	12
2.2.4.6. TERMINAL SECTION.....	12
2.3. USER CLASSES AND CHARACTERISTICS.....	13
2.3.1. SYSTEM ADMINISTRATOR	13
2.3.2 USER	13
2.3.3. LECTURER.....	13
2.3.4. TEACHING ASSISTANT	13
2.4. OPERATING ENVIRONMENT	14
2.5. DESIGN AND IMPLEMENTATION CONSTRAINTS.....	14

2.6. USER DOCUMENTATION	14
2.7. ASSUMPTIONS AND DEPENDENCIES	14
3. EXTERNAL INTERFACE REQUIREMENTS.....	15
3.1. USER INTERFACE.....	15
3.2. HARDWARE INTERFACES	27
3.3. SOFTWARE INTERFACES.....	27
3.4. COMMUNICATION INTERFACES	28
4. SYSTEM REQUIREMENTS	32
4.1. FUNCTIONAL REQUIREMENTS	32
4.1.1. USE CASE	32
4.1.2. USE CASE DIAGRAM	41
4.1.3. DATA FLOW DIAGRAM	42
4.2. NON-FUNCTIONAL REQUIREMENTS.....	42
4.2.1. PRODUCT REQUIREMENT	42
4.2.1.1. USABILITY REQUIREMENT	42
4.2.1.2. PERFORMANCE REQUIREMENT	42
4.2.1.3. SPACE REQUIREMENT	43
4.2.1.4. SEQRITY REQUIREMENT.....	43
4.2.2. ORGANIZATIONAL REQUIREMENT.....	43
4.2.2.1. ENVIRONMENTAL REQUIREMENT	43
4.2.2.2. OPERATIONAL REQUIREMENT.....	43
4.2.3. EXTERNAL REQUIREMENT	43
4.2.3.1. REGULATORY REQUIRMENTS	43
4.2.3.2. ETHICAL REQUIREMENTS	44
4.2.3.3. SAFETY/SECURITY REQUIREMENTS	44
4.3. ORGANIZING SYSTEM FLOW	44
4.3.1. CONTEXT MODEL.....	44
4.3.2. PROCESS MODEL	45

4.3.3. INTERACTION MODEL	45
4.3.4. BEHAVIOR MODEL	45
4.3.4.2. SEQUENCE DIAGRAM	46
4.4. SYSTEM ARCHITECTURE	47
5. SUPPORTING INFROMATION	48
5.1. SOFTWARE REQUIREMENT SPECIFICATION	48
5.2. DOCUMENT HISTORY	48

LIST OF FIGURES

Figure 1 Design of Home page	16
Figure 2 Design of Home page - error	16
Figure 3 Select lecture page	17
Figure 4 Test page - Head section	19
Figure 5 Test page - Problem section.....	20
Figure 6 Test page - Testcase section	22
Figure 7 Test page - Code editor section.....	24
Figure 8 Test page - Result section (terminal).....	26
Figure 9 Use Case Diagram	41
Figure 10 Data Flow Diagram	42
Figure 11 Context Model.....	44
Figure 12 Process Model	45
Figure 13 Sequence Diagram.....	46
Figure 14 System Architecture	47

LIST OF TABLES

Table 1 Table of Acronyms and Abbreviation	7
Table 2 Table of terms and definitions	8
Table 3 Table of Product Functions	9
Table 4 User Interface of Home page.....	15
Table 5 User Interface of Select lecture page.....	17
Table 6 User Interface of Test page - Head section.....	18
Table 7 User Interfaces of Test page - Problem section	20
Table 8 User Interfaces of Test page - Testcase section	21
Table 9 User Interfaces of Test page - Code editor section	23
Table 10 User Interfaces of Test page - Result section	25
Table 11 Hardware Interface of Client.....	27
Table 12 Hardware Interface of Server.....	27
Table 13 Software Interface of Server	27
Table 14 Software Interface of Client	28
Table 15 Login HTTP request interface	28
Table 16 Problem information HTTP request interface	28
Table 17 Class list HTTP request interface	29
Table 18 Class enroll HTTP request interface.....	29
Table 19 Class information HTTP request interface	29
Table 20 Code submit HTTP request interface.....	30
Table 21 Code submission result HTTP request interface.....	30
Table 22 Code execute HTTP request interface	30
Table 23 Code grade HTTP request interface.....	30
Table 24 Code store HTTP request interface	31
Table 25 Code retrieve HTTP request interface	31
Table 26 Table of Product Functions.....	32
Table 27 Use Case of Sign up	34
Table 28 Use Case of Login.....	34
Table 29 Use Case of Home Button.....	34
Table 30 Use Case of Lecture and Assignment information.....	34
Table 31 Use Case of Deadline	35
Table 32 Use Case of Problem Explanation	35
Table 33 Use Case of Testcase.....	35
Table 34 Use Case of Code Editor	36
Table 35 Use Case of Save in Real-time	36
Table 36 Use Case of Copy Button	37
Table 37 Use Case of File Button	37
Table 38 Use Case of Initialize Button.....	37
Table 39 Use Case of Save Button	38
Table 40 Use Case of Run Button	38
Table 41 Use Case of Scoring Button	38
Table 42 Use Case of Submission Button	39
Table 43 Use Case of Code Diff.....	39
Table 44 Use Case of Code Result.....	39
Table 45 Use Case of Code Explanation	40
Table 46 Use Case of Recommend comments	40
Table 47 Document history	48

1. INTRODUCTION

1.1. PURPOSE

본 문서는 성균관대학교에 재학 중인 학생들에게 온라인 코딩 테스트 서비스 프로그램을 제공하기 위해 작성된 요구사항 명세서이다. 본 문서에는 서비스를 운영하는데 필요한 요구사항이 정리되어 있으며, 해당 내용에 기반하여 서비스가 구성되고 개발된다. 모든 서비스는 2022년 2학기 성균관대학교 소프트웨어공학개론의 41분반 TEAM 7에 의해서 디자인되고 운영된다.

본 문서는 TEAM 7에 속한 모든 팀원이 열람하는 것을 상정하여 작성되었다. 이 밖에도 소프트웨어공학개론의 교수 및 조교가 평가를 위해 본 문서를 열람할 수 있으며, 추후 소프트웨어공학개론을 수강하는 학생들이 학습의 용도로 본 문서를 참고할 수 있다. 본 문서의 재배포 및 수정은 교육적인 목적에 한해서만 허용한다.

본 문서는 온라인 코딩 테스트 서비스 프로그램의 개요와 여러 요구사항들을 제시하기 위해서 작성되었다. 웹 어플리케이션 기반으로 서비스가 구현되며, 성균관대학교의 프로그래밍 관련 강의를 수강 중인 학생들이 Python 언어 기반의 코딩 테스트를 진행할 수 있도록 서비스를 제공한다. 해당 과정에서 사용자의 코드를 기능, 효율, 가독성면을 종합적으로 고려하여 평가하는 채점 시스템 등이 개발되며 이에 대한 세부사항은 이하 문서에서 명시한다.

1.2. SCOPE

본 서비스는 성균관대학교 학생들에게 웹 기반의 코딩 테스트 사이트를 제공하기 위해서 고안되었다. 작성 코드의 저장이 가능하게 하여 서비스 사용자의 편리성을 향상시키고, 제출된 코드를 다양한 측면에서 평가하여 학생들의 문제 해결 및 프로그래밍 실력이 증진될 수 있도록 돕는다. 또한, 접근성이 뛰어난 웹 환경을 제공하여 가능한 많은 학생이 별다른 조건 없이도 서비스를 이용할 수 있도록 한다. 사용자는 본 서비스를 통해서 자신의 학습 수준을 파악할 수 있으며, 운영자는 확장성을 고려한 설계를 바탕으로 웹 외의 다양한 어플리케이션을 구현하고 Python 이외의 다른 프로그래밍 언어로 서비스 범위를 넓힐 수 있다.

1.3. DEFINITION, ACRONYMS, AND ABBREVIATION

Table 1 Table of Acronyms and Abbreviation

Acronyms/Abbreviation	Description
URL	Uniform Resource Locator

HTTP	HyperText Transfer Protocol
API	Application Programming Interface
AWS	Amazon Web Services
DRF	Django REST Framework

Table 2 Table of terms and definitions

Term	Definition
Registered	로그인이 되어있는 사용자를 의미
UnRegistered	로그인이 되어 있지 않는 사용자를 의미
Host	개발 시스템을 의미
Client	개발 시스템을 사용하는 성균관대학교 학생을 의미
Front-end	웹 페이지를 구성하고 서버와 통신하는 부분을 의미
Back-end	서버와 데이터베이스를 포함하는 부분을 의미

1.4. REFERENCES

- IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements
- Ian Sommerville, Software Engineering (10th Edition)
- Team 2, "Software Requirement Specification", Last Modified: May 2, 2022
https://github.com/skkuse/2022spring_41class_team2

1.5. OVERVIEW

요구사항 명세서는 크게 다섯 부분(Chapter 1~5)으로 구성되어 있다. Chapter 1에서는 본 서비스의 필요성과 기능에 대해 간략하게 서술하고 문서의 예상 독자층을 정의한다. Chapter 2에서는 제품의 관점과 기능, 운영 환경 및 제약사항과 같이 시스템에 대한 전반적인 설명을 제공한다. Chapter 3에서는 여러 인터페이스를 비롯한 시스템 기능 및 다른 시스템과의 상호 작용을 설명한다. 이를 위해서 사용자 인터페이스와 하드웨어 인터페이스, 프로그램 서비스 요구사항 등을 내용에 포함하고, 클라우드 통신망에 관한 사항을 기재한다. Chapter 4에서는 기능적인 요구사항, 비기능적 요구사항, 시스템 플로우와 같은 시스템 요구사항에 대해 상세하게 명시한다. 마지막으로, Chapter 5에서는 문서의 history를 기술한다.

2. OVERALL DESCRIPTION

2.1. PRODUCT PERSPECTIVE

코딩 테스트는 개발자로 취업을 하기 위해 필수적으로 거쳐야 하는 관문이다. 많은 회사들이 학력, 나이, 스펙보다는 순수한 실력과 현실적인 역량을 확인하기 위해 코딩테스트를 통해 업무 수행 능력을 평가하고 있다. 따라서 이를 대비하기 위해 다양한 코딩 연습 사이트들이 현존하고 있지만, 수많은 문제 중에 필요한 것을 골라 학습하는 것은 쉬운 일이 아니다. 따라서 성균관대 학생에게 적합한 새로운 코딩 사이트를 제작하고자 한다. 실제로 학생들이 수강하는 과목에서 요구하는 과제를 해당 사이트에서 직접 수행, 제출, 채점함은 물론 관련 정보를 획득할 수 있도록 이 교육 웹 사이트를 제작함으로써 성균관대 학생들이 편리하게 코딩 연습을 행할 수 있도록 도와주는 것이 본 프로젝트의 목표이다.

2.2. PRODUCT FUNCTIONS

Table 3 Table of Product Functions

Function Requirement No.	Name of Function	Function Description
F1	회원가입	사용자가 서비스를 이용하기 위해 회원가입을 할 수 있어야 한다.
F2	로그인	사용자가 서비스에 로그인 할 수 있어야 한다.
F3	홈 화면 돌아가기	처음 시작 화면으로 돌아갈 수 있어야 한다.
F4	강의 및 과제 선택	사용자가 자신이 수강하고 있는 강의와 해당 과제를 선택할 수 있어야 한다.
F5	마감기한 확인	과제 제출 마감기간까지 남은 시간을 확인할 수 있어야 한다.
F6	문제설명 및 참조/제약사항	과제에 대한 설명과 관련된 제약사항이 명시되어 있어야 한다.
F7	테스트케이스 검증/실행	출력에 따라 pass/fail 결과를 제공하고, fail인 경우 학생 코드의 결과를 제공해야한다.
F8	코드 에디터 구현	사용자가 코드 에디터를 통해 코드를 작성할 수 있어야 한다.
F9	실시간 코드 저장	사용자가 실시간으로 작성한 코드를 저장하여 브라우저를 재실행해도 이전에 작성된 코드가 존재해야 한다.

F10	코드 복사	작성 중인 코드를 복사할 수 있어야 한다.
F11	파일 불러오기	파일로 저장된 코드를 불러올 수 있어야 한다.
F12	코드 초기화	Skeleton 코드로 초기화 할 수 있어야 한다.
F13	코드 중간 저장	사용자가 코드를 작성하는 도중에 저장할 수 있어야 한다.
F14	코드 실행	터미널을 통해 코드를 실행하고, 실행 결과가 에러가 아닐 경우 출력 값을 제공해야 한다. 에러가 발생한 경우에는 위치를 확인하여 라인 하이라이트로 강조하고, 에러 메시지를 제공해야 한다.
F15	코드 채점	공개/비공개 테스트케이스를 모두 검증하여 코드를 채점한다. 각 테스트케이스의 pass/fail 여부를 제공하고 오픈 테스트케이스에서 fail 했다면 해당 정보를 제공해야 한다. 히든 테스트케이스에서 fail한 경우에는 제공하지 않는다.
F16	코드 제출	과제에 대해 완성된 코드를 제출하며, 최대 3번까지 제출이 가능해야 한다.
F17	코드 분석 (Code Diff)	제출된 코드를 diff2html library를 사용하여 교수자 코드와 비교하고, 결과를 제공한다. 이때 입력창을 확대하면 코드와 결과가 좌우로 표기되도록 해야 한다.
F18	코드 제출 결과	copydetect library를 이용하여 제출 코드의 표절 검사를 진행하고, multimetric, pylama library 등을 이용하여 코드에 대한 기능, 효율성, 가독성에 대한 점수를 각각 표기해야 한다.
F19	코드 설명	Open Codex API를 이용하여 코드에 대한 설명이 이루어져야 한다.
F20	관련 자료 추천	학생이 해결한 문제와 관련된 자료(문제, 강의, 개념 사이트)를 추천해줘야 한다.

2.2.1. HOME PAGE

처음 웹 사이트에 들어가면 홈 페이지가 나타난다. 페이지 내 로그인 버튼을 누르면 로그인 URL로 넘어가게 된다.

2.2.2. LOGIN/START

로그인을 할 수 있는 항목이 나타난다.

2.2.3. LECTURE/ASSIGNMENT SELECT

로그인을 완료하면 다시 홈 페이지로 돌아오게 된다. 로그아웃 버튼이 우측 상단에 생성된다. 각 강의 세부 항목에서 사용자는 해당 강의의 과제와 과제 마감기한을 확인할 수 있다.

2.2.4. SOLVING PAGE

로그인을 완료한 이후 나타나는 홈페이지에서 강의와 과제를 선택할 수 있다. 특정 강의의 과제를 선택하면, 선택한 과제를 수행할 수 있는 학습 페이지가 나타난다. 해당 학습을 통해 학생은 과제를 풀어보고, 제출하고, 채점 받으면서 수강과목에 대한 과제를 쉽게 진행할 수 있다.

2.2.4.1. HEAD SECTION

상단에는 헤드 섹션이 존재한다. 헤드 섹션의 좌측에는 홈페이지로 돌아가는 홈 버튼이 있고, 가운데에는 과목명과 수행 중인 과제명이, 우측에는 제출마감까지 남은 기간과 설정 버튼이 있다.

2.2.4.2. PROBLEM SECTION

좌측 상단에는 문제와 참조/제약사항을 표기하는 문제 설명 섹션이 있다. 선택한 과제에 해당하는 문제를 표기하고, 해당 문제를 풀 때 고려해야 할 참조사항 및 제약사항을 표기한다.

2.2.4.3. TEST CASE SECTION

좌측 하단에는 테스트 케이스가 표기되는 섹션이 있다. 공개 테스트 케이스를 표기하고, 해당 케이스에서 고려해야하는 input과 도출되는 output을 표기한다. 히든 테스트 케이스는 채점을 위해 내부에서만 사용되고, 여기서 직접적으로 표기하지는 않는다. 각 테스트케이스 별로 존재하는 검증 버튼을 클릭하면 output 여부에 따라 pass/fail 결과를 제공하고, fail의 경우에는 학생 코드의 output 결과를 함께 제공한다.

2.2.4.4. CODE EDITOR SECTION

가운데에는 코드 입력이 가능한 코드 에디터 섹션이 존재한다. 상단에는 저장 버튼과 이전에 저장한 내용을 불러올 수 있는 버튼이 있고, 중간에는 코드를 직접 입력할 수 있게 되어있으며, 각 라인 별 번호가 좌측에 표기된다.

2.2.4.5. FUNCTION SECTION

가운데 하단에는 각종 기능에 대한 버튼 섹션이 존재한다. 좌측에는 파일에 작성된 코드를 불러올 수 있는 버튼, skeleton 코드로 초기화할 수 있는 버튼, 작성중인 코드를 복사할 수 있는 버튼, 코드를 파일로 다운로드 가능한 버튼이 존재하며, 우측에는 실행 버튼, 채점 버튼, 제출버튼이 차례로 위치한다.

2.2.4.6. TERMINAL SECTION

우측에는 결과 섹션(터미널)이 나타난다. FUNCTION SECTION의 출력 버튼을 눌렀을 경우에 작동하며, 실행 결과가 에러가 아닌 경우 출력 값을 제공하고 실행 결과가 에러일 경우 에러 메시지에 따른 위치를 확인하여 해당 라인을 하이라이트 표기한다. 채점 결과는 점수, pass/fail 여부, Code diff 결과(Over Score 도표, 기능 점수, 효율 점수, 가독성 점수, 표절도), 관련 자료(문제, 영상, 학습자료) 등을 포함하여 제공한다.

2.3. USER CLASSES AND CHARACTERISTICS

2.3.1. SYSTEM ADMINISTRATOR

시스템 관리자는 웹 페이지에 대한 이해를 가지고 있어야 한다. 웹 사이트 상에 코딩 작성이 요구되는 과제들의 수행에 필요한 기능들이 무엇이 있고, 학생들이 이를 어떻게 사용하는지에 대한 전반적인 흐름을 파악할 수 있어야 하며, 웹 페이지가 프론트 엔드와 백엔드를 통해 어떻게 작용하고, 서버와의 연결을 통해 어떤 방식으로 과제를 가져오는지 전반적인 이해가 필요하다. 더 나아가 Code diff와 OpenAI Codex API가 어떤 방식으로 작동하는지에 대한 지식이 있다면 좋을 것이다. 그렇기 때문에 소프트웨어 전공자가 적합할 것이며, 사용자의 불만 사항을 처리할 수 있어야 한다.

2.3.2. USER

프로그램 사용자는 소프트웨어 전공 과목을 수강하는 성균관대학교 학생이 될 것이다. 일반적으로 학생들은 직접 코드 작업용 프로그램을 설치하거나, 구글 코랩 등 인터넷 사이트를 통해 코딩을 진행하기 때문에 과제를 확인하고 수행하는데 있어서 불편을 겪는다. 그렇기 때문에 본 프로그램을 활용한다면 과제를 보다 편리하게 확인하고 수행할 수 있을 것이다.

2.3.3. LECTURER

교수자 및 과제 출제자는 학생들이 해결해야 할 과제와 해당과제의 채점에 사용할 모범 답안을 제작하는 사람일 것이다. 따라서 관련된 지식이 뛰어나고 온전한 문제를 제작할 수 있는 사람이 적합할 것이다.

2.3.4. TEACHING ASSISTANT

사용자가 제출한 과제의 점수를 기록하고, 경우에 따라 학생들의 질문사항에 답할 수 있는 조교가 될 것이다. 해당 강의의 커리큘럼을 잘 알고 있으며 그에 대한 지식도 갖춘 소프트웨어 전공자가 적합할 것이다.

2.4. OPERATING ENVIRONMENT

- Window 7
- Window 8
- Window 10
- Mac OS
- Linux

2.5. DESIGN AND IMPLEMENTATION CONSTRAINTS

- 공개된 타 사이트를 참고하여 작동 구조 확인
- 별도의 라이선스 혹은 비용을 지불해야 하는 소프트웨어의 사용을 자제
- 성균관대학교 학생의 학습에 적합하도록 아이캠퍼스 계정과 연계
- 사용자가 편안하게 작업을 진행할 수 있도록 피로도를 줄이는 디자인 설계
- 코드의 최적화를 통해 유지보수 편의성 획득
- 직관적인 코드 변수명으로 차후 변경사항 발생시 대응 유연화

2.6. USER DOCUMENTATION

사용자는 홈페이지에 있는 도움말을 통해 프로그램 이용에 도움을 받을 수 있다. 도움말 페이지에 들어가면 각 기능에 대한 간단한 설명이 표시되어 있어 사용자가 더 쉽게 인터페이스에 접근할 수 있다.

2.7. ASSUMPTIONS AND DEPENDENCIES

본 명세서에 명시된 시스템은 웹을 통해 운용되고 제공되는 것을 전제한다. 따라서 이 시스템의 유저는 laptop 혹은 desktop을 사용하여 인터넷에 접속할 수 있어야 한다.

3. EXTERNAL INTERFACE REQUIREMENTS

3.1. USER INTERFACE

Table 4 User Interface of Home page

이름	홈페이지
목적/설명	홈페이지에 진입하면 사용자는 '로그인' 버튼을 클릭하여 아이디와 비밀번호를 통해 로그인 할 수 있다. 회원가입이 되어있지 않은 경우 사이트 내부에서 회원가입이 가능하다.
입력 소스/ 출력 대상	클라이언트/호스트서버 호스트서버/클라이언트
범위/ 정확도/ 오차 범위	N/A
구성 단위	화면
시간/속도	사용자가 해당 항목을 클릭했을 때/클라이언트와 서버간 통신 속도
기타 입출력관계	1. 사용자 계정 정보의 유효함을 체크한 결과값을 받는다. 2. 계정 정보가 유효하지 않을 경우 오류 메시지를 출력한다. 3. 회원가입 시 데이터베이스에 사용자 정보 추가한다.
화면 형식과 구성	1. '로그인' 클릭 시 계정 정보가 유효한 경우 강의 선택 화면으로 이동한다. 2. '회원가입' 클릭 시 가입 화면으로 이동한다.
데이터 유형	텍스트
명령 유형	'로그인' 버튼과 '회원가입' 버튼에 해당하는 명령
종료 메시지	회원가입 시 '회원가입 완료'

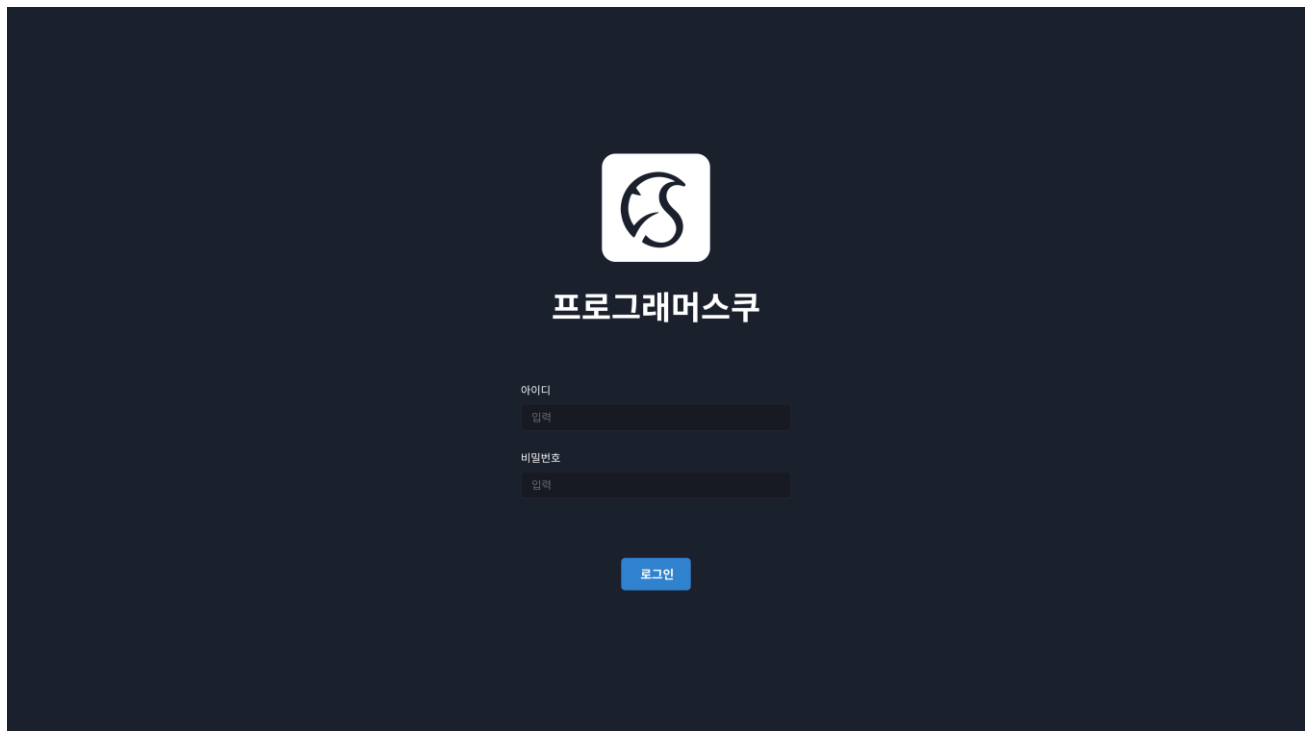


Figure 1 Design of Home page

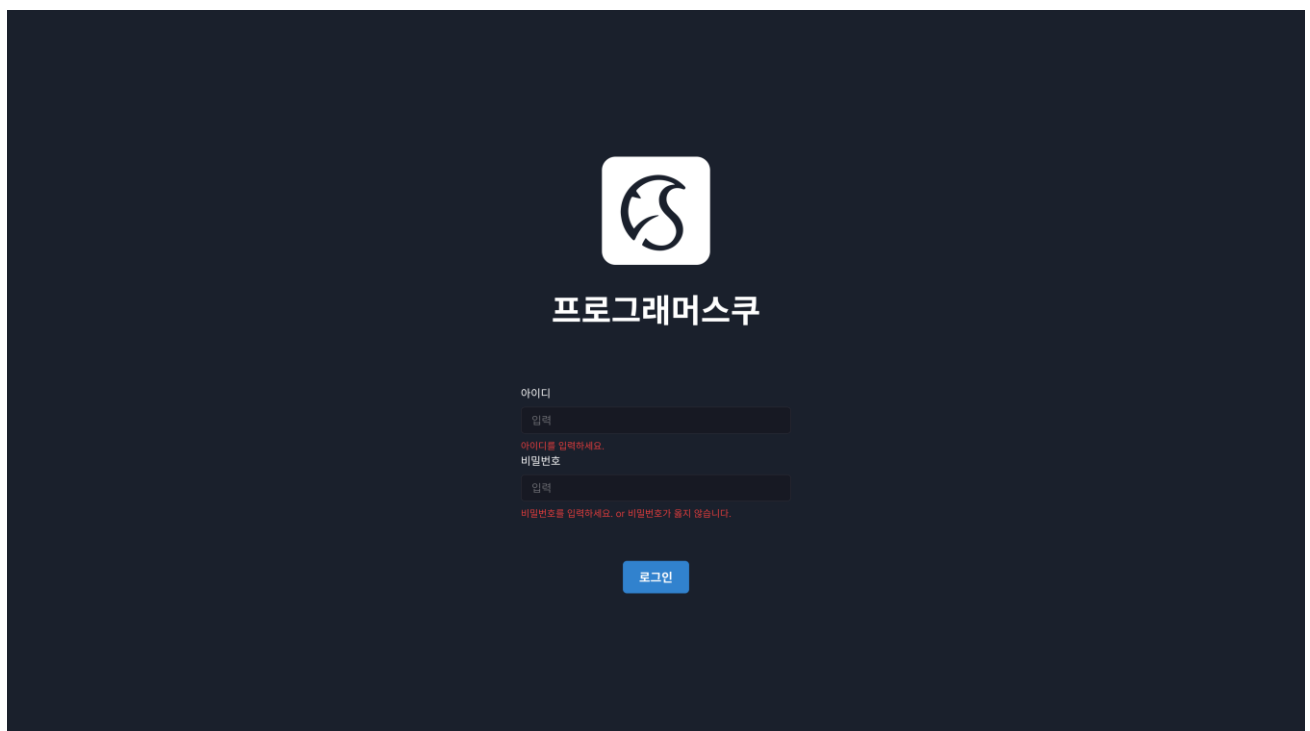


Figure 2 Design of Home page - error

Table 5 User Interface of select lecture page

이름	강의 선택 페이지
목적/설명	사용자는 강의 목록 중에서 원하는 강의를 선택하여 '시험 응시하기' 버튼을 통해 선택한 강의의 테스트 페이지로 진입할 수 있다. 취소 버튼을 통해 시험 응시를 취소할 수 있다.
입력 소스/ 출력 대상	클라이언트/호스트서버 호스트서버/클라이언트
범위/ 정확도/ 오차 범위	강의의 총 개수만큼 강의 목록에 표시
구성 단위	화면
시간/속도	사용자가 해당 항목을 클릭했을 때/클라이언트와 서버간 통신 속도
기타 입출력과 관계	N/A
화면 형식과 구성	<ol style="list-style-type: none"> 1. Select 컴포넌트인 '강의 선택'을 클릭했을 때 시험 응시가 가능한 강의 목록을 볼 수 있다. 2. '취소' 버튼 클릭 시 시험 응시가 취소된다. 3. '시험 응시하기' 버튼 클릭 시 테스트 페이지로 이동한다.
데이터 유형	텍스트
명령 유형	'강의 선택', '시험 응시하기', '취소' 버튼에 해당하는 명령
종료 메시지	N/A

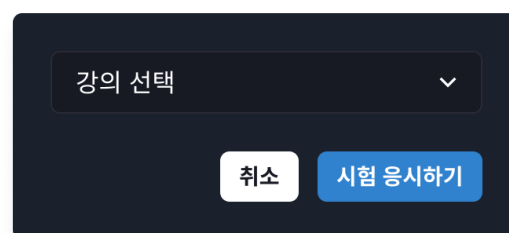


Figure 3 Select lecture page

Table 6 User Interface of Test page - Head section

이름	테스트 페이지 - 헤드 섹션
목적/설명	헤드 섹션을 통해 사용자가 선택한 강의, 과제명, 사용자의 이름, 과제 마감일을 볼 수 있으며, 시험 종료가 가능하고, 코드 에디터에 대한 설정을 할 수 있다. 로고 버튼 클릭 시 홈페이지로 이동할 수 있다.
입력 소스/ 출력 대상	클라이언트/호스트서버 호스트서버/클라이언트
범위/ 정확도/ 오차 범위	N/A
구성 단위	화면
시간/속도	N/A
기타 입출력과 관계	남은 제출 기간을 사용자에게 실시간으로 표시한다.
화면 형식과 구성	<ol style="list-style-type: none"> 1. 시험 페이지의 상단에 위치하는 섹션이다. 2. 가장 왼쪽에 성균관 대학교의 로고 이미지가 표시된다. 클릭 시 홈페이지로 이동한다. 3. 시험을 수행중인 강의, 과제명이 breadcrumb으로 표시된다. 4. 과제명 클릭 시 강의에 해당하는 과제 목록을 볼 수 있으며 다른 과제로 이동할 수 있다. 5. 사용자의 이름을 확인할 수 있다. 6. 과제의 남은 제출 기간을 확인할 수 있으며, 일/시간/분/초 단위로 제공된다. 7. 시험 종료 버튼을 통해 시험을 종료할 수 있다. 8. 설정 버튼을 통해 배경 색을 변경할 수 있다. (단, 기능은 추후 구현한다.)
데이터 유형	텍스트, 이미지
명령 유형	다른 과제로 이동, 시험 종료버튼, 로고 클릭에 해당하는 명령
종료 메시지	N/A

The screenshot displays the '문제 설명' (Problem Description) section of an online coding test platform. The problem is about calculating the sum of the first N Fibonacci numbers. The description includes the problem statement, a formula for the sum, a list of input/output examples, and a hint. The '테스트케이스' (Test Cases) section shows three cases: Case 1 (PASS), Case 2 (FAIL), and Case 3 (PASS). The interface also features a top navigation bar with the user's name '소프트웨어공학' and a dropdown menu for 'week 1: 피보나치 수'. A right sidebar shows the test progress, including a timer for '2월 13시간 30분 12초' and a '시행 종료' (End Test) button. The main coding area is currently empty, showing only the problem description and test cases.

문제 설명

피보나치 수는 0과 1로 시작한다. 0번째 피보나치 수는 0이고, 1번째 피보나치 수는 1이다. 그 다음 2번째 부터는 바로 앞 두 피보나치 수의 합이 된다.

이를 식으로 써보면 $FN = FN-1 + FN-2$ ($N \geq 2$)가 된다.

$N=17$ 일때 까지 피보나치 수를 써보면 다음과 같다.

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597

N 이 주어졌을 때, N 번째 피보나치 수를 구하는 프로그램을 작성하시오.

참조 / 제약 사항

첫째 줄에 N 이 주어진다.

N 은 45보다 작거나 같은 자연수이다.

테스트케이스

테스트케이스 1 PASS 완료

입력: 4
출력: 3

테스트케이스 2 FAIL 완료

입력: 5
출력: 5
내 출력: 6

테스트케이스 3 완료

입력: 6
출력: 8

1. 2022-10-06 22:22:07.362 저장 10

실행 재검 제출 13

22:24:08 >> 프로세스가 시작되었습니다.
22:24:09 >> 처리중...

Figure 4 Test page – Head section

Table 7 User Interfaces of Test page – problem section

이름	테스트 페이지 – 문제 섹션
목적/설명	사용자는 문제 설명과 참조/제약사항을 확인할 수 있다.
입력 소스/ 출력 대상	클라이언트/호스트서버 호스트서버/클라이언트
범위/ 정확도/ 오차 범위	N/A
구성 단위	화면
시간/속도	N/A
기타 입출력과 관계	N/A
화면 형식과 구성	<ol style="list-style-type: none"> 1. 테스트 페이지의 좌측 상단에 위치하는 섹션이다. 2. 문제 설명 부분에는 사용자가 해결해야 할 문제에 대한 설명이 주어진다. 3. 참조/제약사항 부분에는 문제에 대한 참조/제약사항이 주어진다.
데이터 유형	텍스트
명령 유형	N/A
종료 메시지	N/A

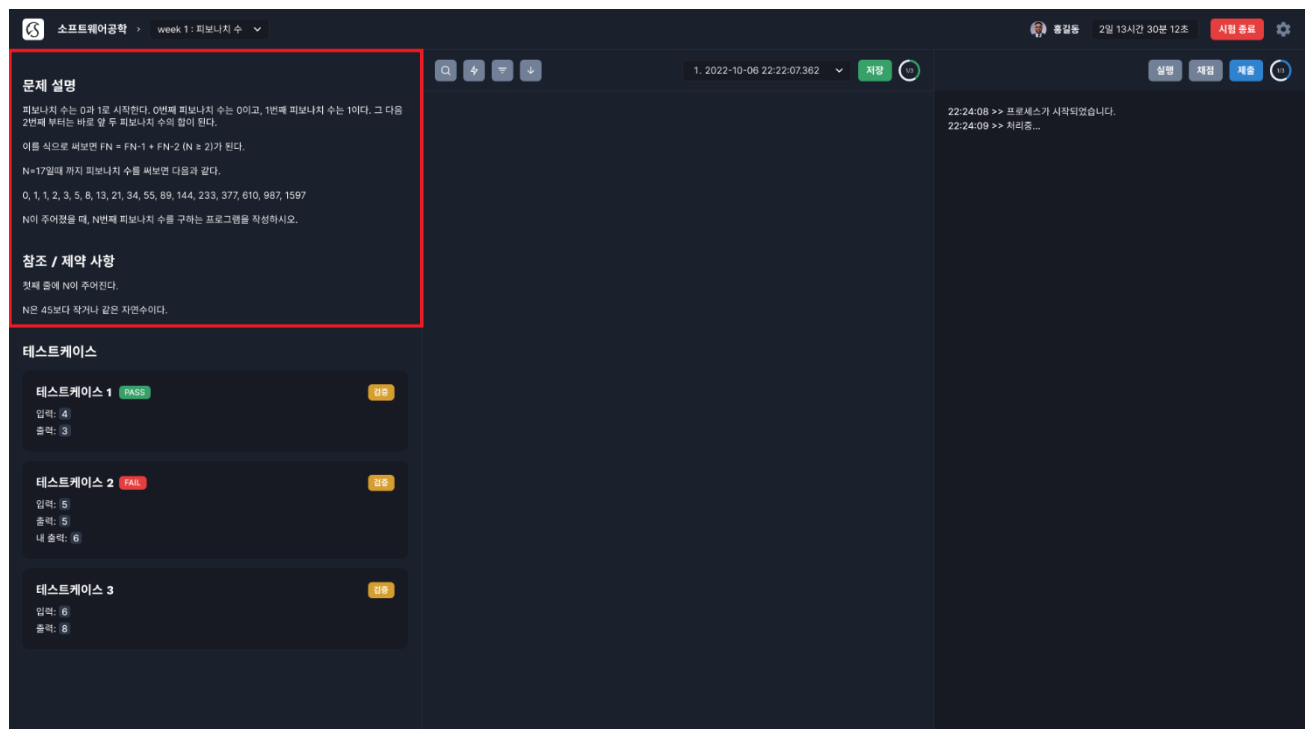


Figure 5 Test page – Problem section

Table 8 User Interfaces of Test page – test case section

이름	시험 페이지 – 테스트 케이스 섹션
목적/설명	사용자가 작성한 코드를 기반으로 테스트 케이스의 검증을 해볼 수 있다. '검증'버튼을 클릭하여 검증을 할 수 있으며, 테스트 케이스는 공개와 비공개로 구분된다. 사용자에게는 공개 테스트 케이스만 보이도록 한다.
입력 소스/ 출력 대상	클라이언트/호스트서버 호스트서버/클라이언트
범위/ 정확도/ 오차 범위	테스트 케이스의 개수/ 파이썬 인터프리터와 사용자의 코드에 따른 정확도/ 파이썬 인터프리터와 사용자의 코드에 따른 오차
구성 단위	화면
시간/속도	사용자가 검증 버튼을 눌렀을 때/ 실행 코드의 복잡도 및 클라이언트와 서버간 통신 속도
기타 입출력과 관계	1. 테스트 케이스의 입력/출력 값 표시 2. 검증 클릭 시 테스트 케이스의 pass/fail 여부를 사용자에게 표시 3. fail일 경우 사용자 코드의 출력 결과도 표시
화면 형식과 구성	1. 테스트 케이스의 개수만큼 테스트 케이스를 표시한다. 2. 사용자에게 각 테스트 케이스의 입력 값과 올바른 출력 값을 보여준다. 3. 각 테스트 케이스에는 '검증' 버튼이 있으며, 클릭 시 사용자의 코드가 테스트 케이스의 입력 값을 input으로 하여 실행된다. 이후 테스트 케이스의 pass/fail여부를 사용자에게 표시한다. 4. pass일 경우, 테스트 케이스의 이름 우측에 PASS를 출력한다. 5. fail일 경우, 테스트 케이스의 이름 우측에 FAIL을 출력한다. 6. fail일 경우, 사용자의 코드의 출력 결과를 올바른 출력 값 하단에 표시한다.
데이터 유형	텍스트
명령 유형	'검증' 버튼에 해당하는 명령
종료 메시지	PASS 또는 FAIL 표시

문제 설명

피보나치 수의 0과 1로 시작한다. 0번째 피보나치 수는 0이고, 1번째 피보나치 수는 1이다. 그 다음 2번째 부터는 바로 앞 두 피보나치 수의 합이 된다.

이름 식으로 써보면 $FN = FN-1 + FN-2$ ($N \geq 2$)가 된다.

$N=17$ 번째 까지 피보나치 수를 써보면 다음과 같다.

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597

N 이 주어졌을 때, N 번째 피보나치 수를 구하는 프로그램을 작성하시오.

참조 / 제약 사항

첫째 줄에 N 이 주어진다.

N 은 45보다 작거나 같은 자연수이다.

테스트케이스

테스트케이스	결과	입력	출력
테스트케이스 1	PASS	4	3
테스트케이스 2	FAIL	5	5
테스트케이스 3	PASS	6	8

22:24:08 >> 프로세스가 시작되었습니다.
22:24:09 >> 처리중...

Figure 6 Test page – Test case section

Table 9 User Interfaces of Test page – code editor section

이름	시험 페이지 – 코드 에디터 섹션
목적/설명	사용자가 좌측 문제에 대한 코드를 작성한다. 외부 파이썬 파일을 불러올 수 있고, 코드를 개별 저장할 수 있으며, skeleton code 로 돌아갈 수 있다. 작성한 코드를 별도의 파일로 저장할 수 있다. 코드는 파이썬을 사용하고, 가독성을 돕는 시각화 기능을 제공한다.
입력 소스/ 출력 대상	클라이언트/호스트서버 호스트서버/클라이언트
범위/ 정확도/ 오차 범위	저장한 코드의 수/ N/A / N/A
구성 단위	화면
시간/속도	사용자가 각 버튼을 눌렀을 때 / 클라이언트와 서버 간 통신 속도
기타 입출력과 관계	<ol style="list-style-type: none"> 1. 로컬에 존재하는 파이썬 파일 목록 표시 및 파일 선택 시 해당 파일 내용 출력 2. 현재 작성한 코드를 skeleton code 로 초기화 3. 현재 코드를 파이썬 파일로 추출 4. 현재 작성한 코드의 내용 저장 / 저장된 코드 불러오기 5. 사용자 로그인 정보를 바탕으로 해당 사용자 데이터베이스에 업데이트
화면 형식과 구성	<ol style="list-style-type: none"> 1. 코드 에디터 상단의 네비게이션 바에는 코드 관련 버튼이 있다. <ol style="list-style-type: none"> 1-2) '불러오기' 버튼은 로컬에 존재하는 파이썬 파일 목록을 보여주고 사용자가 선택한 파일을 불러와 에디터에 그 내용을 출력한다. 1-3) '초기화' 버튼은 작성된 코드를 초기 skeleton code로 되돌린다. 1-4) '내려받기' 버튼은 작성된 코드를 별도의 파이썬 파일로 로컬에 저장한다. 1-5) '저장' 버튼은 현재까지 작성한 코드의 내용을 별도로 저장한다. 최대 3개의 코드를 저장할 수 있고, 현재 저장된 코드의 개수를 우측 circular progress로 보여준다. 2. 'code editor' 은 파이썬 코드의 라인 넘버와 줄 맞춤, 키워드 별 색깔 구분을 제공한다.
데이터 유형	텍스트
명령 유형	'불러오기', '저장', '초기화', '파일 다운로드', '저장된 코드 불러오기' 버튼에 해당하는 명령
종료 메시지	N/A

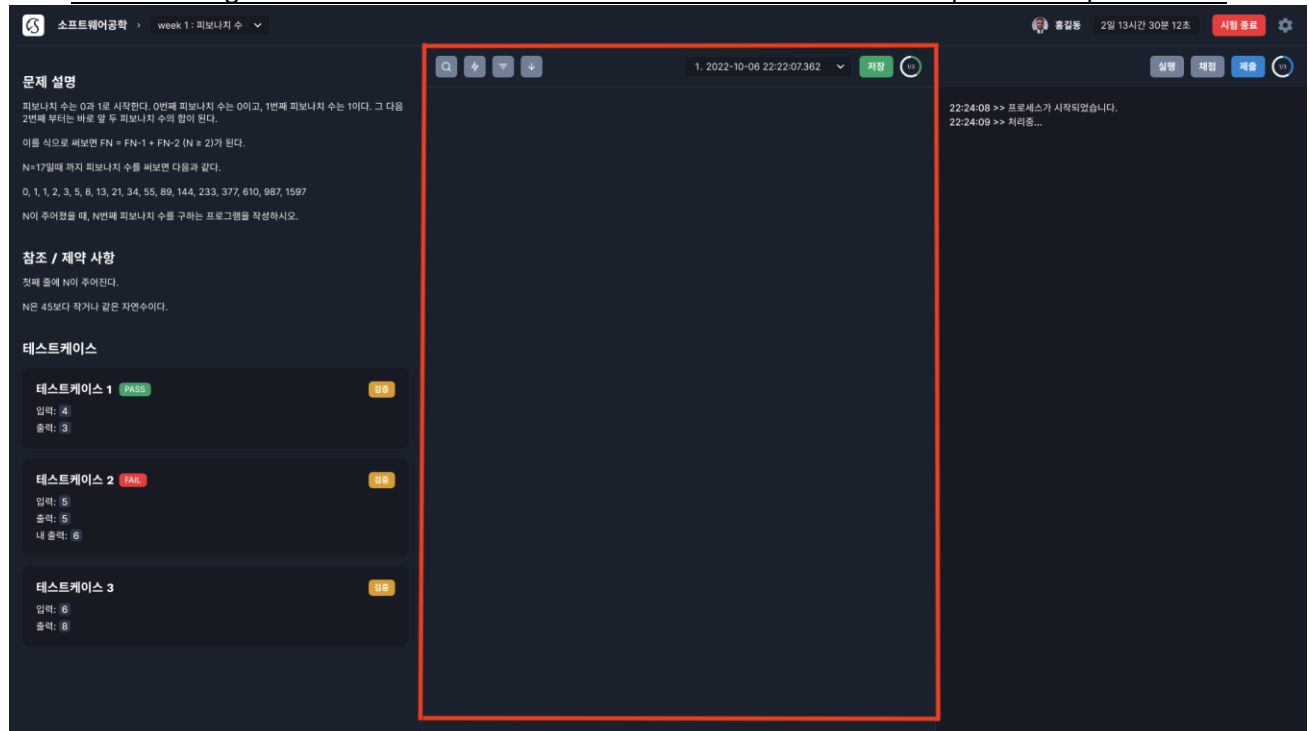


Figure 7 Test page – Code editor section

Table 10 User Interfaces of Test page – result section (terminal)

이름	시험 페이지 – 결과 출력 섹션
목적/설명	사용자가 작성한 코드를 실행하거나 제출하여 채점을 진행한 결과를 출력한다. 사용자의 코드를 실행하거나 제공하는 테스트 케이스를 통해 채점한 결과를 서버로부터 받아 이를 출력한다. 제출은 최대 3번 할 수 있으며 제출하게 될 경우 사용자의 코드의 가독성, 효율, 기능, 표절을 검토하여 분석한 결과를 서버로부터 받아 이를 출력한다.
입력 소스/ 출력 대상	클라이언트/호스트서버 호스트서버/클라이언트
범위/ 정확도/ 오차 범위	제출한 사용자의 코드 / 파이썬 인터프리터와 사용자의 코드에 따른 정확도 / 파이썬 인터프리터와 사용자의 크롤링 실습 코드에 따른 오차
구성 단위	화면
시간/속도	사용자가 각 버튼을 눌렀을 때/ 실행 코드의 복잡도 및 클라이언트와 서버간 통신 속도
기타 입출력과 관계	<ol style="list-style-type: none"> 1. 사용자가 작성한 코드를 실행한 결과 출력 2. 사용자가 작성한 코드를 공개/비공개 테스트 케이스로 채점한 결과 출력 3. 사용자가 작성한 코드를 공개/비공개 테스트 케이스로 채점한 결과 및 표절, 가독성, 효율, 기능 면에서 분석한 결과를 출력
화면 형식과 구성	<ol style="list-style-type: none"> 1. 결과 출력 터미널 상단의 네비게이션 바는 코드의 실행 관련 버튼이 있다. <ol style="list-style-type: none"> 1-2) '실행' 버튼은 사용자가 작성한 코드를 실행한 결과를 터미널에 출력한다. 1-3) '제출' 버튼은 사용자가 작성한 코드를 공개/비공개 테스트 케이스를 통해 채점하여 Pass/Fail을 출력하고, Fail일 경우 사용자의 코드가 출력하는 결과를 보여준다. 1-4) '제출' 버튼은 사용자가 작성한 코드를 공개/비공개 테스트 케이스로 채점하고 표절, 가독성, 효율, 기능 면에서 분석하여 결과를 종합적인 결과를 출력한다. 제출은 최대 3번 가능하다.
데이터 유형	텍스트
명령 유형	'실행' 버튼, '채점' 버튼, '제출' 버튼 각각에 해당하는 명령
종료 메시지	'해당 문제가 제출되었습니다.'

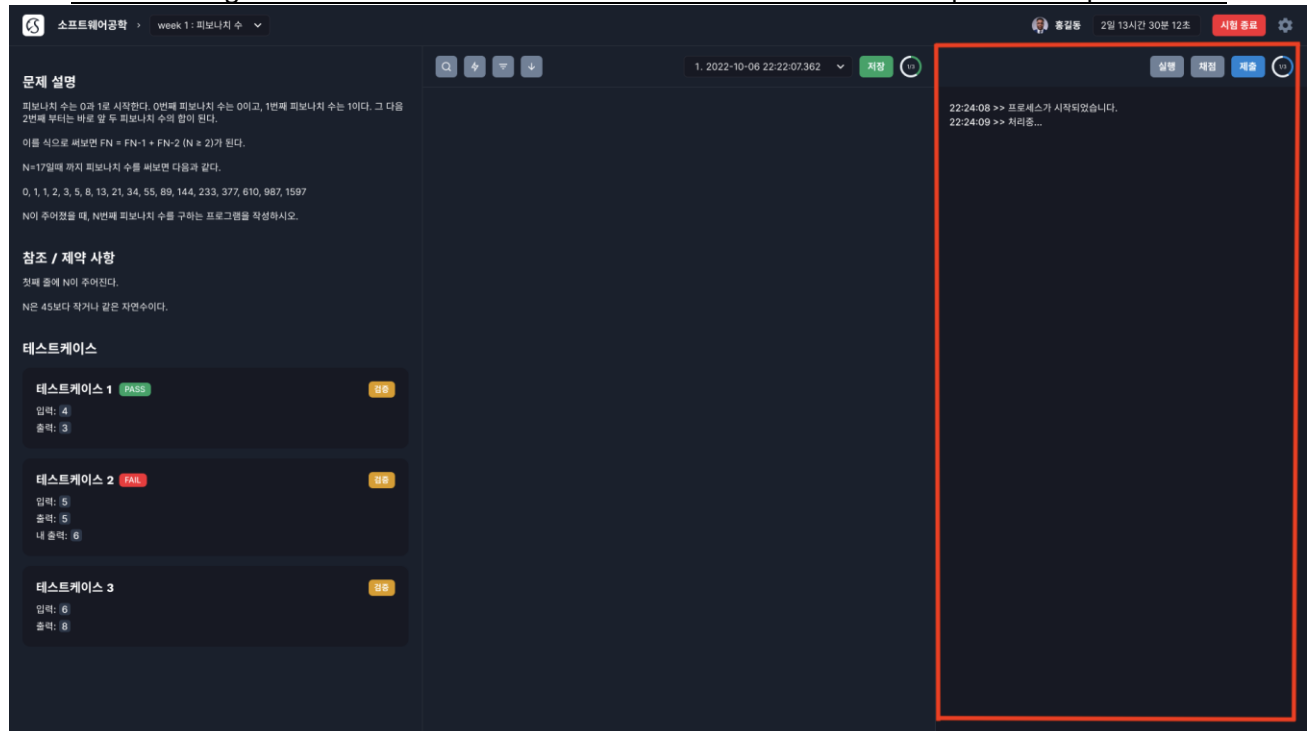


Figure 8 Test page – Result section (terminal)

3.2. HARDWARE INTERFACES

Table 11 Hardware Interface of Client

이름	클라이언트 - 사용자가 이 시스템에 접근할 수 있는 기기
목적/설명	네트워크 통신이 가능하고, 기본 입출력 장치가 가능한 기기로 사용자는 웹 브라우저를 통해 이 시스템을 이용할 수 있다.

Table 12 Hardware Interface of Server

이름	서버 - AWS 플랫폼 이용
목적/설명	클라우드 컴퓨팅 서비스 이용하여 서버를 구축한다.
구성	프론트엔드 서버와 백엔드 서버는 AWS EC2 인스턴스 1대에 배포되고 서로 연결되어 있다. 해당 인스턴스에는 데이터베이스가 설치되어 있고 백엔드 서버와 연결되어 있다.
구성 단위	Amazon EC2
스펙	T2.micro - vCPU:1, 메모리:1GiB, 기준 성능/vCPU:20%, 획득한 CPU 크레딧/시간:6

3.3. SOFTWARE INTERFACES

Table 13 Software Interface of Server

이름	서버 구축에 필요한 소프트웨어 인터페이스	
목적 및 설명	사용자에게 코딩 테스트 환경과 문제를 제공하고, 로그인 기능 및 코드 실행과 채점을 처리하기 위해 서버가 사용하는 소프트웨어 패키지 인터페이스이다.	
Python3 3.9.0~	설명	서버 애플리케이션을 구동하기 위한 런타임 엔진
	입력	백엔드 서버 동작을 구현한 Python 코드 파일 및 Django/DRF 리소스
	출력	외부 접근 가능한 JSON 형식의 데이터
NodeJS 16.1.0~	설명	프론트엔드 애플리케이션을 구동하기 위한 런타임 엔진
	입력	프론트엔드 서버 동작을 구현한 JavaScript 코드 파일 및 React 라이브러리 리소스
	출력	외부 접근 가능한 HTML, CSS, JavaScript 파일 및 데이터
SQLite DB 3.39.0~	설명	사용자 데이터를 보관 및 처리하기 위한 데이터베이스 엔진
	입력	사용자 데이터 처리를 위한 질의문 (SQL 형식)
	출력	결과 반영 혹은 질의 응답
unittest	설명	사용자의 코드를 채점하기 위해 사용하는 단위 테스트 프레임워크
	입력	사용자의 코드
	출력	코드 실행 출력 결과와 테스트 통과 여부
copydetect 0.4.2~	설명	사용자 코드의 표절 검사를 위한 라이브러리
	입력	사용자의 코드
	출력	표절 검사 결과 리포트 파일 (HTML 형식)
multimetric	설명	사용자 코드의 효율성을 측정하기 위한 라이브러리

1.3.0~	입력	사용자의 코드
	출력	코드의 효율성 측정 결과 출력 (JSON 형식)
pylama 8.4.0~	설명	사용자 코드의 가독성을 측정하기 위한 라이브러리
	입력	사용자의 코드
	출력	코드 가독성 검사 결과 출력 (이슈 목록)
openAI codex API	설명	사용자 코드에 대한 추가적인 설명을 생성하기 위한 외부 API
	입력	사용자의 코드
	출력	사용자의 코드에 대한 설명문 (영어)

Table 14 Software Interface of Client

이름	서비스 이용을 위한 클라이언트 소프트웨어 인터페이스	
목적 및 설명	사용자가 서비스를 이용하기 위한 소프트웨어 인터페이스다.	
웹 브라우저 (Chorme 권장)	설명	사용자가 서비스와 상호작용하기 위한 기본 환경
	입력	HID 기반 사용자 입력
	출력	HTML, CSS, JavaScript 기반의 렌더링된 웹페이지
Python 3.9.0~ (optional)	설명	사용자가 코드 테스트를 응시할 때 로컬 환경에서 실행해보기 위한 환경
	입력	사용자 코드
	출력	실행 결과 출력
SSH Client	설명	서비스 관리자가 서버에 접속하고 관리하기 위해 사용하는 프로그램
	입력	서비스 서버 내 TTY Shell 로그인 및 입력
	출력	서비스 서버 내 Shell 출력

3.4. COMMUNICATION INTERFACES

Table 15 Login HTTP request interface

이름	로그인 HTTP 요청
용도	사용자가 서비스를 이용하기 위해 등록된 계정에 로그인할 때 전송하는 요청
Method	POST
URL	/login
Content Type	application/json
내용	사용자 학번 및 비밀번호
서버 동작	DB에서 학번으로 사용자를 검색 비밀번호를 해싱해서 찾은 사용자의 비밀번호 값과 비교해서 검증 성공시 사용자 정보를 반환 + 브라우저 쿠키(인증 정보) 설정 헤더

Table 16 Problem information HTTP request interface

이름	문제 정보 HTTP 요청
용도	사용자가 선택한 문제에 대한 정보를 얻기 위한 요청

Method	GET
URL	/problems/:id
Content Type	application/json
내용	문제 id 브라우저 쿠키(인증 정보)
서버 동작	DB에서 요청한 id의 문제를 검색 찾은 문제에서 히든 테스트케이스, 관련 자료, 정답 코드를 필터링 필터링된 내용 + 저장된 코드 목록 + 제출물id 목록 반환

Table 17 Class list HTTP request interface

이름	강의 목록 HTTP 요청
용도	사용자가 등록할 수 있는 강의 목록 얻기 위한 요청
Method	GET
URL	/classes
Content Type	application/json
내용	브라우저 쿠키(인증 정보)
서버 동작	DB에서 강의 목록을 검색 찾은 강의 목록에서 마감 날짜가 지난 강의 필터링 강의의 id, 이름 목록을 반환

Table 18 Class enroll HTTP request interface

이름	강의 수강 HTTP 요청
용도	사용자가 선택한 강의를 등록하고 시험을 시작하는 요청
Method	GET
URL	/classes/:id/enroll
Content Type	N/A
내용	강의 id 브라우저 쿠키(인증 정보)
서버 동작	DB에서 id로 강의 검색 찾은 강의의 수강 가능 여부 검사 (마감 날짜 검사) 수강 가능하면 수강 처리 (기록 없을 시 생성)

Table 19 Class information HTTP request interface

이름	강의 정보 HTTP 요청
용도	사용자가 선택한 강의에 대한 정보들을 얻기 위한 요청
Method	GET
URL	/classes/:id
Content Type	application/json
내용	강의 id 브라우저 쿠키(인증 정보)
서버 동작	DB에서 id로 강의 검색 찾은 강의의 수강 가능 여부 검사 (수강 기록 유무 검사)

	강의의 모든 정보 + 연결된 문제 id, 이름 목록 반환
--	---------------------------------

Table 20 Code submit HTTP request interface

이름	코드 제출 HTTP 요청
용도	사용자의 코드를 제출하기 위한 요청
Method	POST
URL	/submissions?problem_id=:id
Content Type	application/json
내용	사용자 코드 브라우저 쿠키(인증 정보)
서버 동작	DB에서 id로 문제 검색 찾은 문제에 대한 제출 가능 여부 검사 (최대 제출 횟수 검사) 제출 가능할 시 제출물 기록 생성 제출된 코드 검증 및 분석 처리 생성된 제출물 기록 id 반환

Table 21 Code submission result HTTP request interface

이름	코드 제출 결과 HTTP 요청
용도	사용자의 제출물에 대한 결과를 얻기 위한 요청
Method	GET
URL	/submissions/:id
Content Type	application/json
내용	제출물 id 브라우저 쿠키(인증 정보)
서버 동작	DB에서 id로 제출물 검색 제출물의 상태 검사 (분석 완료 여부) 연결된 강의의 마감 여부 검사 제출 결과 정보 반환

Table 22 Code execute HTTP request interface

이름	코드 실행 HTTP 요청
용도	사용자 코드를 실행하기 위한 요청
Method	POST
URL	/execute
Content Type	application/json
내용	사용자 코드 브라우저 쿠키(인증 정보)
서버 동작	제출된 코드를 실행 출력 결과를 반환

Table 23 Code grade HTTP request interface

이름	코드 채점 HTTP 요청
----	---------------

용도	사용자 코드를 채점하기 위한 요청
Method	POST
URL	/grade/:id
Content Type	application/json
내용	사용자 코드 테스트케이스 id 브라우저 쿠키(인증 정보)
서버 동작	제출된 코드를 실행 출력 결과를 id에 맞는 테스트케이스에 대해 검증 출력 결과와 검증 결과 반환

Table 24 Code store HTTP request interface

이름	코드 저장 HTTP 요청
용도	사용자 코드를 서버에 저장하기 위한 요청
Method	POST
URL	/storages/:order
Content Type	N/A
내용	저장소 번호 브라우저 쿠키(인증 정보)
서버 동작	사용자의 코드를 순서에 맞는 저장소에 저장 (덮어쓰기)

Table 25 Code retrieve HTTP request interface

이름	코드 불러오기 HTTP 요청
용도	서버에 저장된 사용자의 코드를 얻기 위한 요청
Method	GET
URL	/storages/:order
Content Type	application/json
내용	저장소 번호 브라우저 쿠키(인증 정보)
서버 동작	순서에 맞는 저장소에서 코드를 검색 찾은 코드를 반환

4. SYSTEM REQUIREMENTS

4.1. FUNCTIONAL REQUIREMENTS

4.1.1. USE CASE

Table 26 Table of Product Functions

Function Requirement No.	Name of Function	Function Description
F1	회원가입	사용자가 서비스를 이용하기 위해 회원가입을 할 수 있어야 한다.
F2	로그인	사용자가 서비스에 로그인 할 수 있어야 한다.
F3	홈 화면 돌아가기	처음 시작 화면으로 돌아갈 수 있어야 한다.
F4	강의 및 과제 선택	사용자가 자신이 수강하고 있는 강의와 해당 과제를 선택할 수 있어야 한다.
F5	마감시한 확인	과제 제출 마감기간까지 남은 시간을 확인할 수 있어야 한다.
F6	문제설명 및 참조/제약사항	과제에 대한 설명과 관련된 제약사항이 명시되어 있어야 한다.
F7	테스트케이스 검증/실행	출력에 따라 pass/fail 결과를 제공하고, fail인 경우 학생 코드의 결과를 제공해야한다.
F8	코드 에디터 구현	사용자가 코드 에디터를 통해 코드를 작성할 수 있어야 한다.
F9	실시간 코드 저장	사용자가 실시간으로 작성한 코드를 저장하여 브라우저를 재실행해도 이전에 작성된 코드가 존재해야 한다.
F10	코드 복사	작성 중인 코드를 복사할 수 있어야 한다.
F11	파일 불러오기	파일로 저장된 코드를 불러올 수 있어야 한다.
F12	코드 초기화	Skeleton 코드로 초기화 할 수 있어야 한다.
F13	코드 중간 저장	사용자가 코드를 작성하는 도중에 저장할 수 있어야 한다.
F14	코드 실행	터미널을 통해 코드를 실행하고, 실행 결과가 에러가 아닐 경우 출력 값을 제공해야 한다. 에러가 발생한 경우에는

Online Coding Test Web Platform

Requirement Specification

		위치를 확인하여 라인 하이라이트로 강조하고, 에러 메시지를 제공해야 한다.
F15	코드 채점	공개/비공개 테스트케이스를 모두 검증하여 코드를 채점한다. 각 테스트케이스의 pass/fail 여부를 제공하고 오픈 테스트케이스에서 fail 했다면 해당 정보를 제공해야 한다. 히든 테스트케이스에서 fail한 경우에는 제공하지 않는다.
F16	코드 제출	과제에 대해 완성된 코드를 제출하며, 최대 3번까지 제출이 가능해야 한다.
F17	코드 분석 (Code Diff)	제출된 코드를 diff2html library를 사용하여 교수자 코드와 비교하고, 결과를 제공한다. 이때 입력창을 확대하면 코드와 결과가 좌우로 표기되도록 해야 한다.
F18	코드 제출 결과	copydetect library를 이용하여 제출 코드의 표절 검사를 진행하고, multimetric, pylama library 등을 이용하여 코드에 대한 기능, 효율성, 가독성에 대한 점수를 각각 표기해야 한다.
F19	코드 설명	Open Codex API를 이용하여 코드에 대한 설명이 이루어져야 한다.
F20	관련 자료 추천	학생이 해결한 문제와 관련된 자료(문제, 강의, 개념 사이트)를 추천해줘야 한다.

Table 27 Use Case of Sign up

이름	Sign up
사용자	UNREGISTERED
설명	사용자가 서비스를 이용하기 위해 회원가입을 할 수 있어야 한다. 회원가입 정보는 데이터베이스에 저장하여 추후 로그인하는데 문제가 발생하지 않도록 해야 한다.
정상 프로세스	사용자가 회원가입을 할 수 있다.
사전 조건	서버가 정상적으로 동작한다. 서버와 데이터베이스가 정상적으로 연결되어 있다.
사후 조건	데이터베이스에 회원가입 정보가 반영된다.
가정	사용자의 인터넷 환경이 원활하다

Table 28 Use Case of Login

이름	Login
사용자	REGISTERED
설명	사용자가 서비스에 로그인 할 수 있어야 한다. 시스템은 로그인을 검증하기 위해서 데이터베이스에서 저장된 사용자 정보를 가져온다.
정상 프로세스	사용자가 로그인을 할 수 있다.
사전 조건	서버가 정상적으로 동작한다. 서버와 데이터베이스가 정상적으로 연결되어 있다.
사후 조건	X
가정	사용자의 인터넷 환경이 원활하다. 사용자의 회원 정보가 데이터베이스에 반영되어 있다.

Table 29 Use Case of Home Button

이름	Return to the home page
사용자	REGISTERED
설명	사용자는 로그인을 진행한 이후 나타나는 화면에서 홈 버튼을 통해 처음 시작화면으로 돌아갈 수 있어야 한다.
정상 프로세스	사용자가 홈 버튼을 누르면 시작 화면으로 돌아간다.
사전 조건	서버가 정상적으로 동작한다. 사용자가 로그인을 한 상태이다.
사후 조건	X
가정	사용자의 인터넷 환경이 원활하다.

Table 30 Use Case of Lecture and Assignment Information

이름	Get lecture and assignment information
사용자	REGISTERED

설명	사용자가 현재 진행 중인 강의와 과제에 대한 정보를 불러올 수 있어야 한다. 사용자는 로그인 이후 나타나는 페이지에서 해당 내용을 볼 수 있으며, 특정 강의에 대한 과제를 선택하여 문제를 확인할 수 있다.
정상 프로세스	사용자는 강의 목록과 그에 해당하는 과제를 볼 수 있다.
사전 조건	서버가 정상적으로 동작한다. 사용자가 로그인을 한 상태이다. 서버와 데이터베이스가 정상적으로 연결되어 있다.
사후 조건	X
가정	사용자의 인터넷 환경이 원활하다. 강의와 과제에 대한 정보가 데이터베이스에 반영되어 있다.

Table 31 Use Case of Deadline

이름	Check the assignment deadline
사용자	REGISTERED
설명	사용자가 수행할 과제를 선택하고 나면, 시스템은 해당 과제의 마감기한을 사용자에게 알려줘야 한다.
정상 프로세스	사용자가 문제 페이지의 우측 상단을 통해 과제 마감 기한을 확인할 수 있다.
사전 조건	서버가 정상적으로 동작한다. 사용자가 로그인을 한 상태이다. 서버와 데이터베이스가 정상적으로 연결되어 있다.
사후 조건	과제 마감 기한이 지나면, 과제 수행이 불가능하게 된다.
가정	사용자의 인터넷 환경이 원활하다. 과제 마감기한이 데이터 베이스에 반영되어 있다.

Table 32 Use Case of Problem Explanation

이름	Identify problem description and reference/restriction
사용자	REGISTERED
설명	문제에 대한 설명과 해당 문제를 해결하기 위해서 고려해야 하는 참조 및 제약사항을 제공해야 한다. 사용자는 특정 강의에 대한 과제를 선택한 이후 나타나는 페이지의 문제 설명 섹션에서 이를 확인할 수 있다.
정상 프로세스	사용자가 문제 설명 섹션을 통해 문제에 대한 내용을 확인할 수 있다.
사전 조건	서버가 정상적으로 동작한다. 사용자가 로그인을 한 상태이다. 서버와 데이터베이스가 정상적으로 연결되어 있다.
사후 조건	X
가정	사용자의 인터넷 환경이 원활하다. 문제에 대한 설명과 참조 및 제약사항은 데이터베이스에 사전 기재되어 있다.

Table 33 Use Case of Testcase

이름	Testcase validation
사용자	REGISTERED

설명	코드 작성 페이지의 좌측 하단에 테스트 케이스가 표기되는 섹션이 존재한다. 해당 섹션에서 공개 테스트 케이스를 확인할 수 있으며, 각 케이스마다 고려해야하는 input과 도출되는 output이 표기된다. 히든 테스트 케이스는 채점을 위해서 내부적으로만 사용되고, 표기되지 않는다. 테스트케이스의 검증 버튼을 클릭하면 output 여부에 따른 pass/fail 결과를 제공하고, fail의 경우 학생 코드의 output 결과를 함께 제공한다.
정상 프로세스	사용자는 공개 테스트케이스를 확인할 수 있고, 검증 버튼을 통해 작성한 코드가 테스트를 통과하는지 확인해볼 수 있다.
사전 조건	서버가 정상적으로 동작한다. 사용자가 로그인을 한 상태이다. 서버와 데이터베이스가 정상적으로 연결되어 있다.
사후 조건	X
가정	사용자의 인터넷 환경이 원활하다. 문제의 테스트케이스는 데이터베이스에 사전 저장되어 있다.

Table 34 Use Case of Code Editor

이름	Edit the code
사용자	REGISTERED
설명	사용자가 코드를 작성할 수 있어야 한다. 시스템은 사용자가 코드를 입력할 수 있도록 코드 에디터를 제공해야 하고, 작성된 코드를 터미널에서 실행하여 실행 결과를 보여줘야 한다.
정상 프로세스	사용자가 코드를 작성하고 해당 코드를 터미널 상에서 실행시킬 수 있다.
사전 조건	서버가 정상적으로 동작한다. 사용자가 로그인을 한 상태이다.
사후 조건	X
가정	사용자의 인터넷 환경이 원활하다.

Table 35 Use Case of Save in Real-time

이름	Save the real-time code
사용자	REGISTERED
설명	예기치 못한 사건으로 인해 웹사이트가 갑작스럽게 종료되어도 브라우저를 재실행하면 사용자가 작성 중이던 코드가 저장되어 있어야 한다. 이를 위해서 서버는 사용자의 코드 작성 로그를 실시간으로 저장하여 사용자가 작성한 코드를 복원할 수 있도록 한다.
정상 프로세스	브라우저가 갑작스럽게 종료되어도, 재실행시 작성 중인 코드가 존재한다.
사전 조건	사용자는 로그인하고 있어야한다. 서버가 정상 동작한다.
사후 조건	X
가정	사용자의 인터넷 환경이 원활하다.

Table 36 Use Case of Copy Button

이름	Copy the code
사용자	REGISTERED
설명	사용자가 작성 중인 코드를 복사할 수 있어야 한다..
정상 프로세스	사용자가 복사 버튼을 누르면, 작성 중이던 코드가 복사된다.
사전 조건	서버가 정상적으로 동작한다. 사용자가 로그인을 한 상태이다.
사후 조건	X
가정	사용자의 인터넷 환경이 원활하다.

Table 37 Use Case of File Button

이름	Load Code Files
사용자	REGISTERED
설명	사용자가 파일에 작성된 코드를 불러올 수 있어야 한다. 시스템은 사용자가 사전에 저장한 코드를 데이터 베이스에 보관하여 언제든지 파일을 불러올 수 있도록 한다.
정상 프로세스	사용자가 파일 불러오기 버튼을 눌러 원하는 코드 파일을 불러오고, 해당 내용이 코드 에디터에 반영된다.
사전 조건	서버가 정상적으로 동작한다. 사용자가 로그인을 한 상태이다. 서버와 데이터베이스가 정상적으로 연결되어 있다.
사후 조건	X
가정	사용자의 인터넷 환경이 원활하다.

Table 38 Use Case of Initialize Button

이름	Return to Skeleton Code
사용자	REGISTERED
설명	사용자가 언제든지 작성 중이던 코드를 skeleton code로 초기화 할 수 있어야 한다. 이를 위해서 시스템은 문제의 skeleton code를 데이터베이스에 보관하고 있어야 한다.
정상 프로세스	사용자가 초기화 버튼을 눌러 skeleton 코드로 초기화한다.
사전 조건	서버가 정상적으로 동작한다. 사용자가 로그인을 한 상태이다. 서버와 데이터베이스가 정상적으로 연결되어 있다.
사후 조건	X
가정	사용자의 인터넷 환경이 원활하다. 문제의 skeleton code가 데이터 베이스에 반영되어 있다

Table 39 Use Case of Save Button

이름	Save code
----	-----------

사용자	REGISTERED
설명	사용자가 작성 중인 코드를 최대 3번 저장할 수 있어야 한다. 시스템은 저장 코드를 데이터 베이스에 반영하여 코드를 언제든지 불러올 수 있도록 한다.
정상 프로세스	사용자가 저장 버튼을 누르면, 작성 중이던 코드가 저장된다.
사전 조건	서버가 정상적으로 동작한다. 사용자가 로그인을 한 상태이다. 서버와 데이터베이스가 정상적으로 연결되어 있다.
사후 조건	데이터베이스 내부에 사용자가 저장한 코드 내용이 기록된다.
가정	사용자의 인터넷 환경이 원활하다.

Table 40 Use Case of Run Button

이름	Run the code
사용자	REGISTERED
설명	사용자는 터미널을 통해 코드를 실행해볼 수 있어야 한다. 여러 번 실행 가능하며, 실행 결과 에러가 발생하지 않았을 경우 코드 출력값이 제공된다. 만약 에러가 발생했을 때는 에러 메시지에 따른 위치를 확인하여 해당 라인을 하이라이트 표기하고 에러 메시지를 제공한다.
정상 프로세스	사용자가 실행 버튼을 누르면 작성한 코드가 실행되고, 터미널 상에 결과가 나타난다.
사전 조건	서버가 정상적으로 동작한다. 사용자가 로그인을 한 상태이다.
사후 조건	X
가정	사용자의 인터넷 환경이 원활하다.

Table 41 Use Case of Scoring Button

이름	Score the code
사용자	REGISTERED
설명	사용자는 작성한 코드를 채점할 수 있어야 한다. 채점은 공개, 비공개 테스트케이스를 모두 검증하여 진행한다. 각 테스트케이스에 대한 pass/fail 여부를 제공하고, 오픈 테스트케이스에서 fail한 경우라면 해당 정보를 제공해야 한다. 다만 히든 테스트케이스가 fail 했을 경우에는 테스트케이스 정보를 제공하지 않아야 한다.
정상 프로세스	사용자가 채점 버튼을 누르면 모든 테스트케이스에 대한 pass/fail 여부와 코드 점수가 결과창에 반영된다.
사전 조건	서버가 정상적으로 동작한다. 사용자가 로그인을 한 상태이다. 서버와 데이터베이스가 정상적으로 연결되어 있다.
사후 조건	X
가정	사용자의 인터넷 환경이 원활하다. 문제의 테스트케이스는 데이터베이스에 사전 반영되어 있다.

Table 42 Use Case of Submission Button

이름	Submit the code
사용자	REGISTERED
설명	사용자는 최대 3번까지 제출이 가능해야 한다. 이를 위해 시스템은 문제당 제출 횟수를 기록하여 제한된 횟수 이상의 제출이 발생하지 않도록 하고, 제출된 코드를 저장해 언제든지 해당 코드를 불러올 수 있도록 한다.
정상 프로세스	사용자가 제출 버튼을 누르면 작성 코드가 제출된다.
사전 조건	서버가 정상적으로 동작한다. 사용자가 로그인을 한 상태이다. 서버와 데이터베이스가 정상적으로 연결되어 있다.
사후 조건	사용자의 제출 횟수와 제출된 코드 내용이 데이터베이스 상에 반영된다.
가정	사용자의 인터넷 환경이 원활하다.

Table 43 Use Case of Code Diff

이름	Compare the code
사용자	REGISTERED
설명	제출된 코드에 대하여 diff2html library를 사용해서 데이터베이스 상에 존재하는 교수자 코드와 비교할 수 있어야 한다. 이때 입력창을 확대하면 코드와 결과가 좌우로 표기되도록 해야 한다.
정상 프로세스	제출 버튼을 누른 이후, 제출 코드와 교수자 코드를 비교한 결과가 나타난다.
사전 조건	서버가 정상적으로 동작한다. 사용자가 로그인을 한 상태이다. 서버와 데이터베이스가 정상적으로 연결되어 있다.
사후 조건	X
가정	사용자의 인터넷 환경이 원활하다. 교수자 코드는 데이터베이스에 사전 반영되어 있다.

Table 44 Use Case of Code Result

이름	Result the code
사용자	REGISTERED
설명	제출한 코드에 대해 copydetect library를 이용하여 표절 검사를 진행하고, multimetric, pylama library 등을 이용하여 코드에 대한 기능, 효율성, 가독성에 대한 점수를 각각 표기해야 한다.
정상 프로세스	사용자가 코드를 제출하고 나면, 제출 결과 section에서 표절율을 비롯하여 기능, 효율, 가독성에 대한 각각의 점수와 종합 점수가 나타난다.
사전 조건	서버가 정상적으로 동작한다. 사용자가 로그인을 한 상태이다. 서버와 데이터베이스가 정상적으로 연결되어 있다.
사후 조건	X
가정	제출 코드에 대한 평가 정보가 데이터베이스에 반영된다.

Table 45 Use Case of Code Explanation

이름	Explain the code
사용자	REGISTERED
설명	Open Codex API를 이용하여 제출 코드에 대한 설명이 이루어져야 한다.
정상 프로세스	사용자가 코드를 제출하고 나면, 코드 설명 section에서 설명이 이루어진다.
사전 조건	서버가 정상적으로 동작한다. 사용자가 로그인을 한 상태이다.
사후 조건	X
가정	사용자의 인터넷 환경이 원활하다.

Table 46 Use Case of Recommend Contents

이름	Recommend the related contents
사용자	REGISTERED
설명	사용자가 제출한 문제와 관련된 문제, 영상, 학습 자료를 추천해줘야 한다. 이를 위해서 시스템은 각 문제에 대한 추천 내용을 데이터베이스 상에 저장하고 있어야 한다.
정상 프로세스	사용자가 코드를 제출하고 나면, 관련 자료 section에 추천 내용이 나타난다.
사전 조건	서버가 정상적으로 동작한다. 사용자가 로그인을 한 상태이다. 서버와 데이터베이스가 정상적으로 연결되어 있다.
사후 조건	X
가정	사용자의 인터넷 환경이 원활하다. 문제에 대한 추천자료는 데이터베이스에 사전 반영되어 있다.

4.1.2. USE CASE DIAGRAM



Figure 9 Use Case Diagram

4.1.3. DATA FLOW DIAGRAM

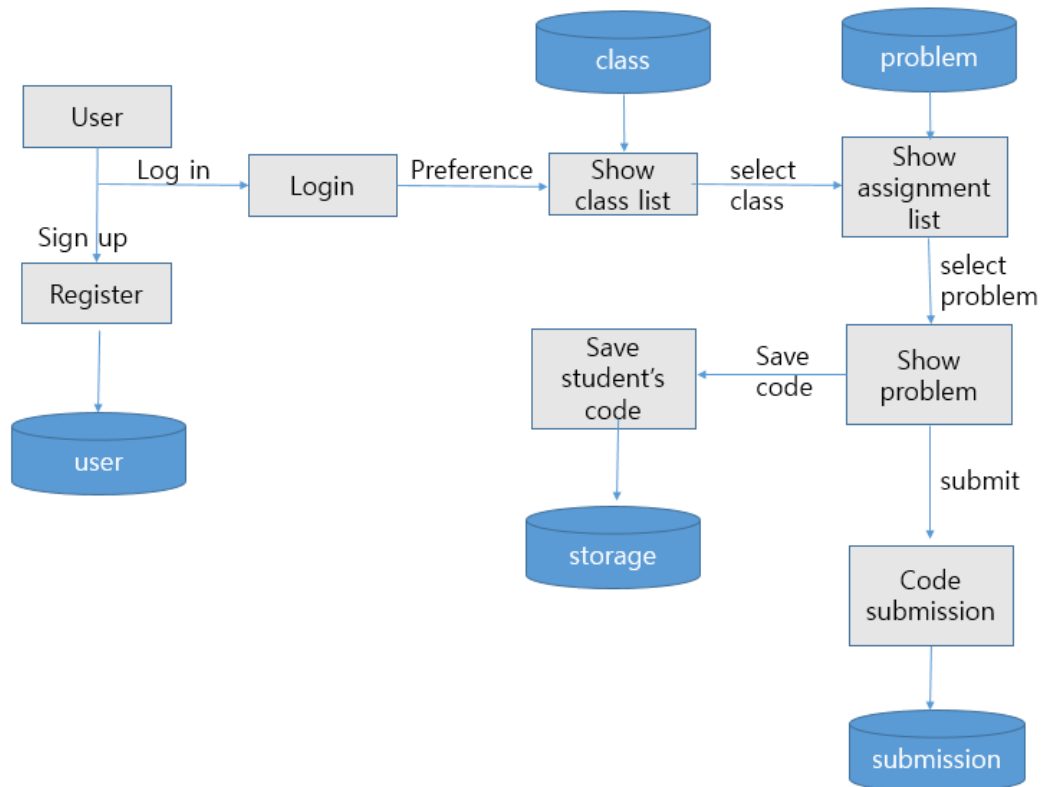


Figure 10. Data Flow Diagram

4.2. NON-FUNCTIONAL REQUIREMENTS

시스템이 제공하는 서비스나 시스템 속성 및 제약 조건과 같은 비기능적 부분에 대해 설명한다.

4.2.1. PRODUCT REQUIREMENT

특정한 방식으로 작동해야 하는 것을 명시하는 요구 사항이다.

4.2.1.1. USABILITY REQUIREMENT

시스템은 전문적이지 않은 사용자가 사용하기 쉬워야 한다. 사용자의 오류를 최소화하도록 시스템 레벨에서 barrier 장치를 마련해 구성되어야 한다. 전문적인 용어의 사용을 최소화해야 하고, 이해하기 쉽게 디자인하여야 한다. 각 사용자는 별도의 매뉴얼 없이 모든 시스템 기능을 직관적으로 사용할 수 있도록 간단 명료하게 구성하여야 한다. 로그인 페이지와 code-editor 페이지에서 사용자의 오류율은 10%를 초과하지 않아야 한다.

4.2.1.2. PERFORMANCE REQUIREMENT

시간당 3,000 명의 사용자를 지원하는 해당 시스템은 chrome 데스크톱 브라우저에서 3 초 이하의 응답 시간을 제공해야 하며, 점수 평가의 기능을 수행하다 보니 안정적이고 신뢰성 있는 시스템이 되어야 할 것이다.

4.2.1.3. SPACE REQUIREMENT

시스템은 지정된 시간에 원하는 서비스를 제공해야 한다. 시스템은 오류를 최소화하면서 서비스를 제공해야 한다. 결과는 사용자가 예상한 대로 신뢰할 수 있어야 한다.

4.2.1.4. SECURITY REQUIREMENT

코드 에디터 프로그램을 사용하려면 사용자의 계정을 인증하고 해당 강좌를 등록해야 한다. 권한이 없는 사용자가 시스템 관리자로 액세스할 수 없고 시스템을 사용할 수 없도록 해야 한다. 또한 권한이 없는 사용자는 시스템 사용자로서 사용자의 개인 정보, ID 및 암호와 같은 기밀 정보에 액세스할 수 없다. 데이터베이스는 HIPPA 요구사항을 충족해야 한다.

4.2.2. ORGANIZATIONAL REQUIREMENT

사용하는 프로세스 표준, 구현 요구사항 등과 같이 조직의 정책 및 절차의 결과로 발생하는 요구사항이다.

4.2.2.1. ENVIRONMENTAL REQUIREMENT

시스템 사용자는 자신의 학번 및 이메일을 사용하여 자신의 신원을 확인해야 한다. 사용자인 교육 대상자는 해당 시스템을 데스크탑 또는 노트북을 이용하여 사용하기를 권장한다.

4.2.2.2. OPERATIONAL REQUIREMENT

호환성이 높은 시스템은 일반적으로 다른 응용 프로그램이 장치에서 실행될 때 잘 작동한다. Windows 7, Windows 8, Windows 8.1, Windows 10 이상이고, Intel Pentium 4 프로세서 또는 SSE3 를 지원하는 그 이상 버전으로 Chrome 브라우저를 이용한다.

4.2.3. EXTERNAL REQUIREMENT

시스템 및 개발 프로세스의 외부 요인에서 파생된 모든 요구 사항을 다룹니다.

4.2.3.1. REGULATORY REQUIREMENTS

본 시스템은 사용자의 로그인 정보를 구글의 OAuth 2.0 을 이용하여 진행한다. OAuth 의 경우 일일 토큰 부여 한도는 10,000 회 이다. 이 이상 진행될 경우 구글에 상향 요청을 보내야한다.

4.2.3.2. ETHICAL REQUIREMENTS

사용자간 답안 공유 및 타인의 결과물을 그대로 쓰지 못하도록 표절 검사를 수행함으로써 해당 문제를 방지한다.

4.2.3.3. SAFETY/SECURITY REQUIREMENTS

개인정보보호법에 따라 사용자의 개인정보를 외부/내부적으로 안전하게 저장하고, 보호해야한다. 시스템이 제공하는 서비스나 시스템 속성 및 제약 조건과 같은 비기능적 부분에 대해 설명한다.

4.3. ORGANIZING SYSTEM FLOW

4.3.1. CONTEXT MODEL

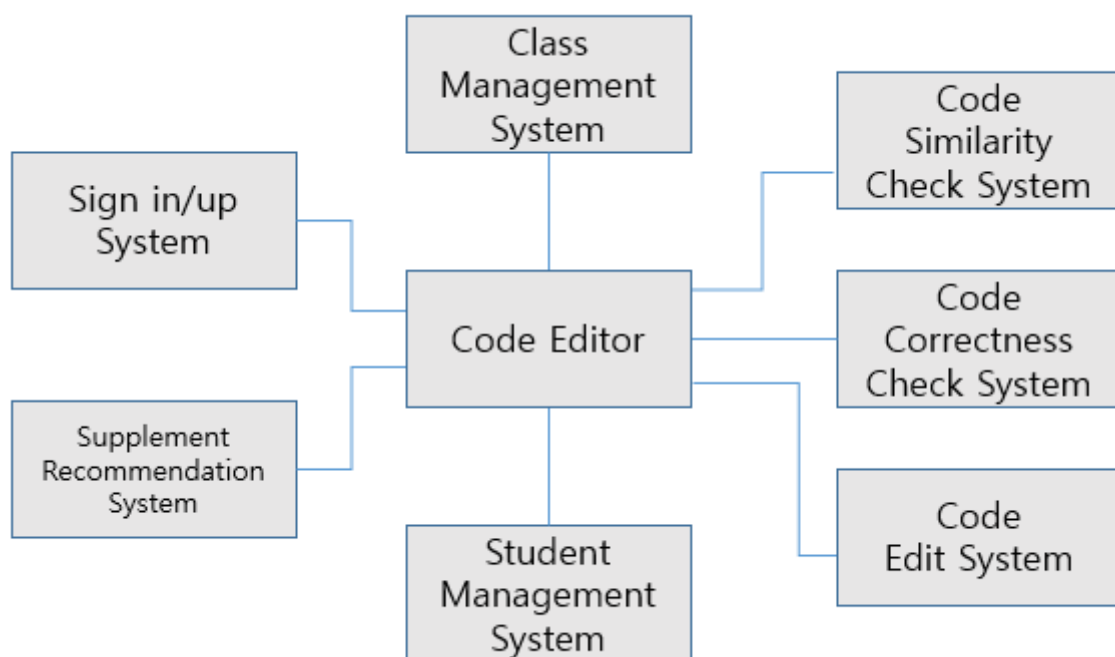


Figure 11 Context Model

4.3.2. PROCESS MODEL

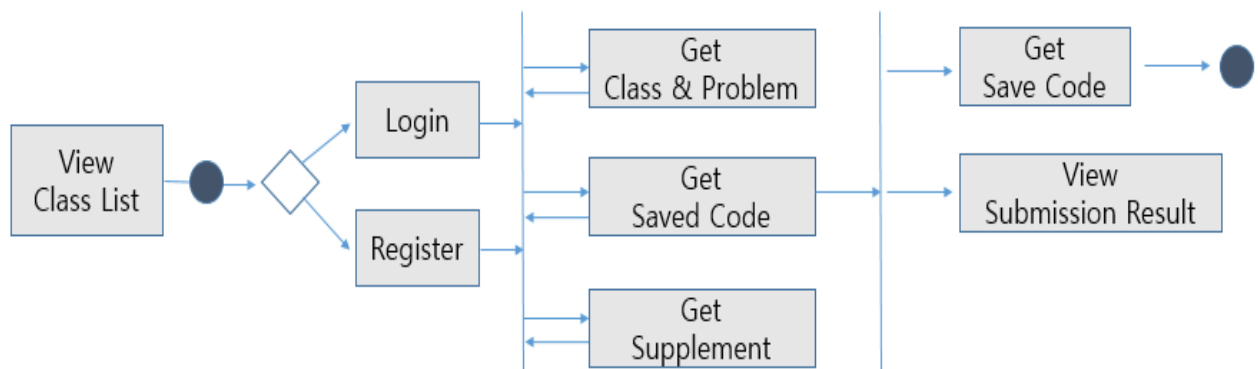


Figure 12 Process Model

4.3.3. INTERACTION MODEL

4.1.2. Use Case Diagram 참고

4.3.4. BEHAVIOR MODEL

4.3.4.1. DATA FLOW DIAGRAM

4.1.3. data flow diagram 참고

4.3.4.2. SEQUENCE DIAGRAM

메인 시스템 설계 상태

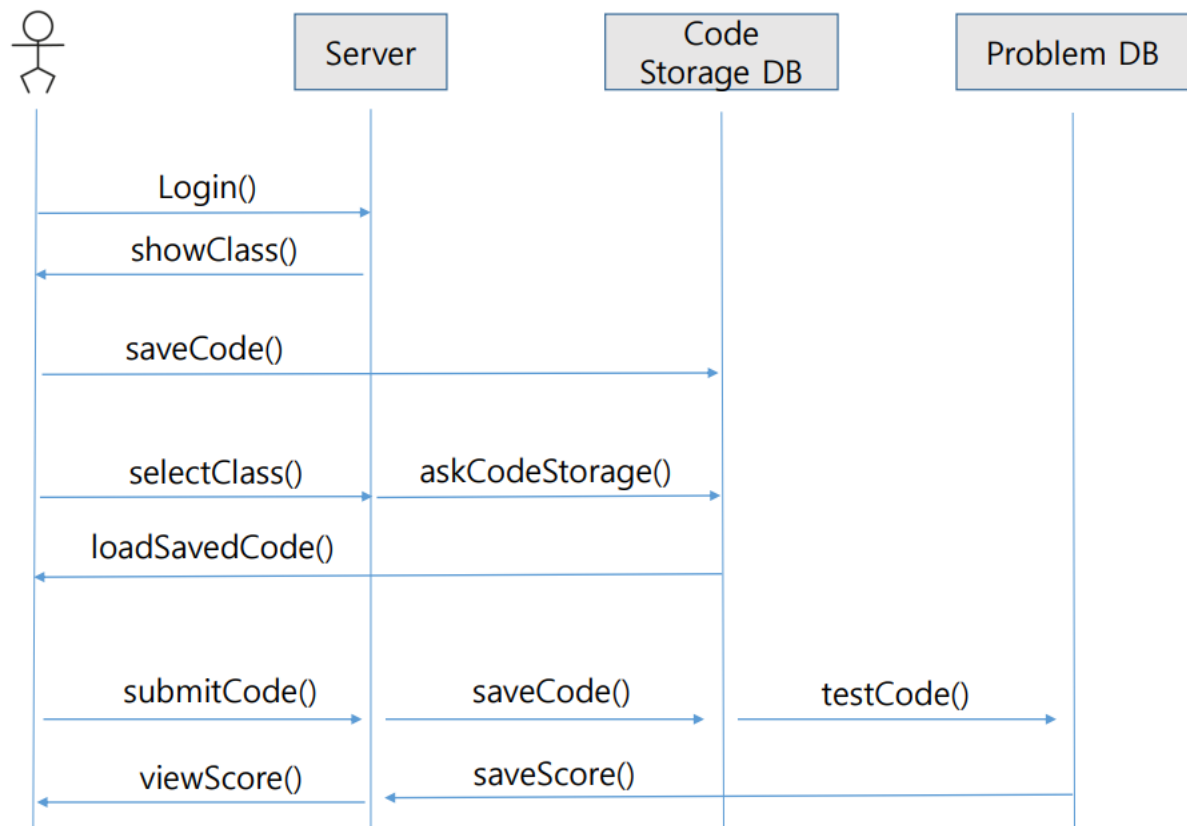


Figure 13 Sequence Diagram

4.4. SYSTEM ARCHITECTURE

이 섹션에서는 예상되는 시스템 아키텍처에 대한 높은 수준의 개요를 제시하고 시스템을 구성하는 각 요소간 존재하는 기능과 통신에 대해 가장 근본적인 형태가 나타난다.

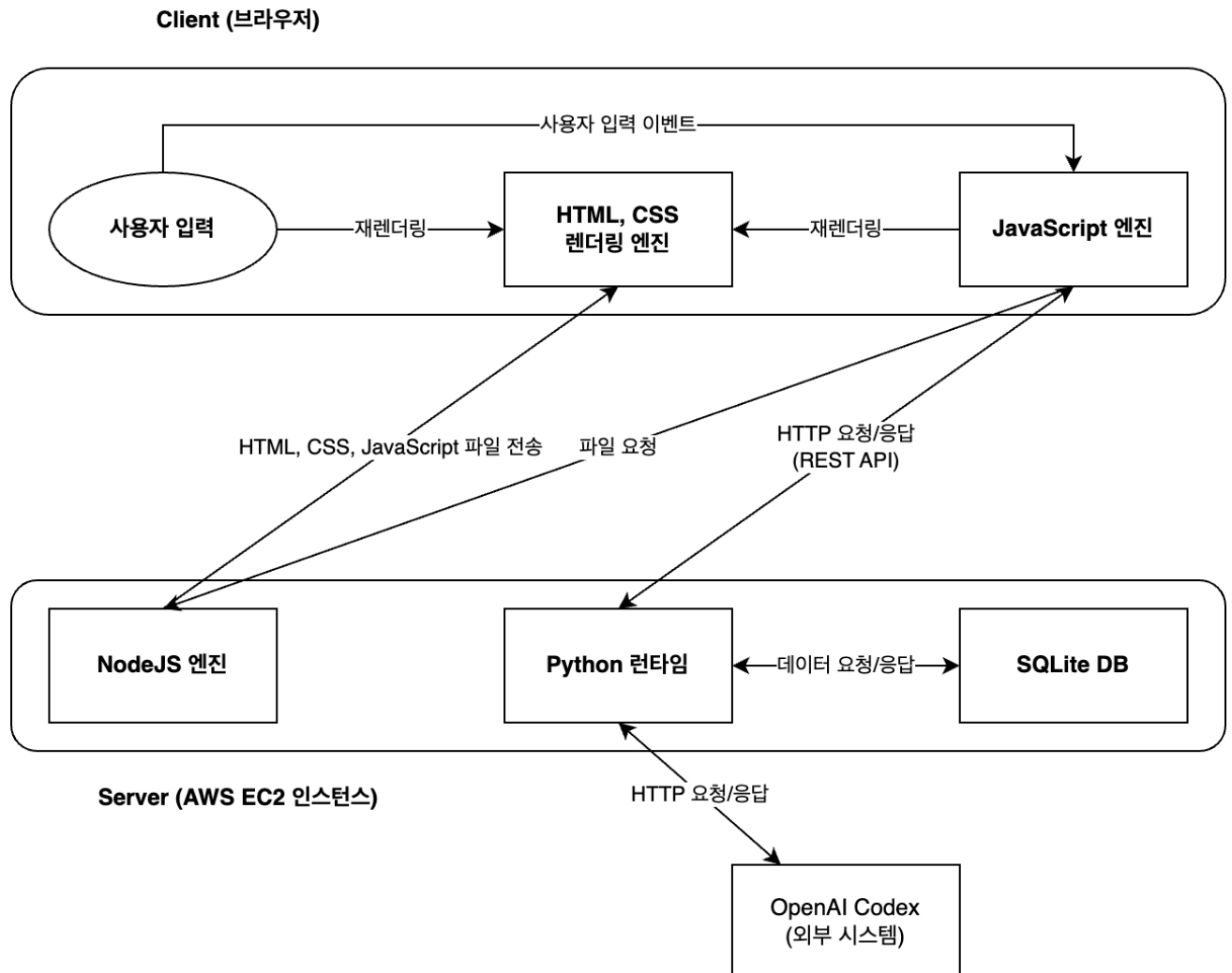


Figure 14 System Architectur

5. SUPPORTING INFROMATION

5.1. SOFTWARE REQUIREMENT SPECIFICATION

소프트웨어 요구사항 명세서는 IEEE Recommended Practice for Software Requirements Specifications 문서를 기준으로 작성되었다.

5.2. DOCUMENT HISTORY

Table 47 Document history

Date	version	Description	Writer
2022/10/22	0.1	1	김호진
2022/10/27	0.1	2	이주빈
2022/10/28	0.1	3.1 (home page ~ testcase section)	김양선
2022/10/29	0.1	4.1.3 ~ 4.3	전윤태
2022/10/29	0.1	3.2 ~ 3.4	변영민
2022/10/29	0.1	4.4	변영민
2022/10/29	0.1	4.1.1 ~ 4.1.2	김호진
2022/10/29	0.1	3.1 (code editor ~ Result section)	임세아
2022/10/30	1	통합	Team 7