



2022-2학기 소프트웨어공학개론 4조

신정환, 한희근, 김성학, 김세란, 김영석, 지준현

Requirement Specification

Contents

1. introduction

- 1.1 purpose
- 1.2 scope
- 1.3 definition, acronyms, abbreviation
- 1.4 references
- 1.5 overview

2. overall description

- 2.1 product perspective
- 2.2 product functions
- 2.3 user classes and characteristics
- 2.4 operating environment
- 2.5 design and implementation constraints
- 2.6 assumptions and dependencies

3. external interface requirement

- 3.1 user interface
- 3.2 hardware interface
- 3.3 software interface
- 3.4 communication interface

4. system requirement

- 4.1. Functional requirements
- 4.2. Non functional requirements
 - 4.2.1 product requirement
 - 4.2.2 organizational requirement
 - 4.2.3 external requirement

4.3. Organizing system flow

- 4.3.1 context model
- 4.3.2 process model

4.4. System Architecture

5. supporting information

- 5.1. software requirement specification
- 5.2. Document history

1. INTRODUCTION

1.1. Purpose

본 문서는 코딩 테스트 웹 서비스를 제공하기 위해 필요한 요구사항을 정리한다. 본 서비스는 **2022년 2학기** 성균관대학교 소프트웨어공학개론의 **Team4**에 의해서 디자인되고 운용된다.

본 문서에는 서비스를 위한 요구사항이 분석, 제시 되어있으며 본 문서에 명시되어 내용에 의거하여 서비스가 구성되고 제공된다.

본 문서는 개발팀이 열람하는 것을 상정하여 작성되었다. 추가적으로, 교수, 조교를 포함한 성균관대학교 소프트웨어공학개론 수강생 또한 학습 및 교육의 용도로 본 문서를 열람할 수 있다.

본 문서는 본 서비스의 개요와 요구사항의 명세를 위해 작성되었다. 본 서비스는 웹 어플리케이션 형태로 구현되어, 여러 기업과 단체에서 각자의 목적에 맞게 코딩 테스트에 응시할 것을 요구하는 현 요구사항에 맞춰 코딩 테스트에 응시하는 사용자가 더욱 편리한 환경에서 문제없이 코딩 테스트를 응시할 수 있도록 하고 문제를 출제하는 출제자가 문제와 테스트 케이스 등 테스트에 필요한 데이터를 추가할 수 있도록 하는 것을 목적으로 코딩 테스트 웹에 필요한 디자인, 프론트 엔드와 백 엔드, 서버와 데이터베이스의 사용을 위한 요구사항을 정리하였다.

1.2. Scope

본 서비스는 코딩 테스트 관련 서비스를 제공한다. 본 서비스는 코딩 테스트를 출제하고자 하는 집단이 웹 페이지에 코딩테스트를 출제하고 테스트 케이스를 추가할 수 있도록 하고 코딩 테스트의 응시자가 게시된 코딩 테스트를 웹에서 코딩 테스트 문제를 열람하고 문제에 대한 답안 코드를 작성, 수정, 저장 및 제출이 가능하도록 한다.

본 서비스를 통해 코딩 테스트 출제자는 별도의 웹 페이지를 구성하지 않고 코딩 테스트 문제와 관련 데이터를 게시할 수 있으며 코딩 테스트 응시자는 코딩테스트를 응시하고 응시하는 과정에서 필요한 편리한 서비스들을 제공받을 수 있다.

서비스를 이용하는 이용자가 불편함을 느끼지 않도록 UI를 제공하고 코딩 테스트 서비스인 만큼 동시 접속자가 많을 것을 예상하여 부하에 대응할 수 있으며 저장되는 데이터에 대한 보안을 철저히 하도록 설계하는 것을 목표로 한다.

1.3. Definition, Acronyms, and Abbreviation

[표 1] 전문용어 및 약어 정리

Acronyms/ Abbreviation	Description
UI	User Interface
Back-end	서버와 데이터베이스를 운영하는 부분을 의미
Front-end	웹 페이지를 구성하고 서버와 통신을 하는 부분을 의미
Django	python 언어를 기반으로 제작된 오픈소스 웹 프레임워크
React	자바스크립트 라이브러리로써 싱글 페이지 애플리케이션의 UI를 생성하는데 집중한 라이브러리
AWS	Amazon Web Services는 Amazon에서 제공하는 클라우드 서비스
Lambda	AWS에서 제공하는 서버리스 컴퓨팅 플랫폼

AWS API Gateway	규모와 관계없이 REST 및 WebSocket API를 생성, 게시, 유지, 모니터링 및 보호하기 위한 AWS 서비스
openAI Codex API	openAI 연구소에서 개발한 자연어처리 인공지능 모델
HTML	Hyper Text Markup Language 약어로 HyperText 기능을 가진 문서를 만드는 언어
HyperText	하이퍼 링크를 통해 사용자가 한 문서에서 다른 문서로 이동할 수 있도록 하는 텍스트
Javascript	웹 페이지에서 복잡한 기능을 구현할 수 있도록 하는 스크립팅 언어 또는 프로그래밍 언어
TCP/ UDP	IP를 사용하는 네트워크 내에서 컴퓨터간 메시지들이 교환될 때 사용하는 프로토콜
REST API	자원을 이름(자원의 표현)으로 구분하여 해당 자원의 상태(정보)를 주고 받는 모든 것을 의미

1.4. References

- IEEE Software Requirements Specification Template
- IEEE Recommended Practice for Software Requirements Specifications
- https://docs.aws.amazon.com/ko_kr/apigateway/latest/developerguide/welcome.html

1.5. Overview

본 소프트웨어 요구사항 명세서는 다섯 개의 장으로 구성되어 있다. 이어지는 두번째 장에서는 개발하고자 하는 제품에 대한 전반적인 설명을 제공한다.

제품의 관점, 기능, 운영환경, 디자인과 구현, 운영에서의 제약사항, 가정 등을 포함한다. 세번째 장에서는 사용자 인터페이스, 하드웨어 인터페이스,

소프트웨어 인터페이스 등 여러 인터페이스에 대한 내용을 포함한다.

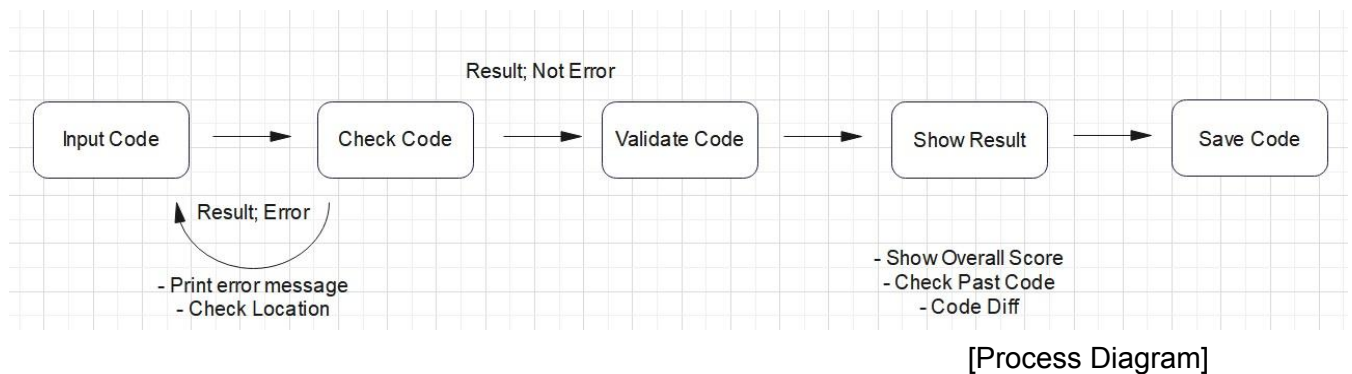
시스템의 기능 및 외부 시스템과의 상호작용 또한 포함한다. 네번째 장에서는 기능적 요구사항, 비기능적 요구사항에 대해 명시하며 조직적 시스템 흐름에 대해서도 명시한다. 마지막 장에서는 프로젝트를 진행하면서 발생하는 변경 이력 및 본 문서에 대한 추가적인 정보를 기록한다.

2. OVERALL DESCRIPTION

2.1. Product Perspective

해당 웹페이지는 학습자의 코드 개발에 도움이 되고자 고안되었다. 작성한 코드에 대해 다양한 측면에서 점수를 측정해주고, 개선 방향을 제시해주며 문제에 관련된 교육자료를 추천해줌으로써 학생의 개발에 도움을 줄 수 있다.

2.2. Product Functions



2.3. User Classes and Characteristics

- 학습자

학생은 문제를 검색하고 문제에 대한 자신의 정답을 제출해 오류 여부를 확인한다. 자신의 제출한 코드와 정답 코드를 비교하고 기능성, 효율성, 가독성 점수를 제공해 어떤 식으로 개선할 수 있을지 알아야 한다. 결과에 따라 참고 자료를 제공해 추가 학습을 권장한다

- 학습 문제 Manager

페이지에 학습 목표에 대한 적절한 학습 문제를 입력하고 추가하거나 수정할 수 있어 변경사항을 서버에 저장할 수 있도록 해야 한다.

- 웹 페이지 Manager

로그인, 페이지 넘김, 실행 결과 출력 오류 등 페이지 실행 도중에 발생할 수 있는 다양한 문제를 처리할 수 있어야 한다. 또한 사용자 **access** 관리하고 **UI** 유지, 보완에 대해서도 담당한다.

- 교수자

학습자는 어떤 코드가 적절한지 자신의 코드가 왜 효율이 떨어지는 지 이유를 알고 보다 나은 코드를 작성할 수 있도록 도움을 준다. 프로그래밍 작업을 통해 수업에 대한 이해를 용이하게 해줄 수 있다.

2.4. Operating Environment

1) Web Browser

- Chrome

2) Windows Platform

- Windows, Linux, macOS

2.5. Design and Implementation Constraints

- Django, React, sqlite를 사용하여 웹 서비스를 개발한다.

- 웹 페이지 디자인은 Figma를 통해 관리한다.

- 코드는 <https://jeonghakhur.gitbooks.io/frontend-coding-convention/content/> 에서 사용할 것을 권장한다.

2.6. Assumptions and Dependencies

웹 페이지는 데스크탑을 기준으로 개발된 웹 페이지이므로 여러 Windows에서 사용되며 사용자가 스마트폰에서 접속하고자 할 때 문제가 생길 수 있다.

테스트 케이스의 개수나 다양성에 따라 코드의 성능에 대한 평가가 달라질 수 있다.

사용자가 문제를 푸는 도중 페이지를 나가거나 코드를 입력하지 않고 제출을 누를 경우 결과가 저장되지 않거나 에러가 발생할 수 있다.


3. EXTERNAL INTERFACE REQUIREMENTS


3.1. User Interface

[표 2] 문제 목록 페이지

이름	문제 목록 (로그인상태)
목적 / 설명	로그인 전엔 문제에 접근할 수 없다. 로그인 후 문제를 클릭하여 해당 페이지로 진입할 수 있다. 난이도, 최신, 제출, 정답률, 혹은 무작위 순서로 정렬할 수 있다.
입력소스 / 출력 대상	클라이언트/호스트서버 호스트서버/클라이언트
범위 / 정확도 / 오차 범위	문제 리스트 / 정렬 순서에 맞는 6개의 문제 표시/ 오차 없음
구성 단위	화면
시간 / 속도	사용자가 해당 항목을 클릭했을 때 / 클라이언트와 서버 간 통신 속도
기타 입출력관계	N/A

화면 형식과 구성	1. 좌상단의 홈버튼은 이 페이지로 로드된다. 새로고침과 동일하다. 2. 헤더의 우측에 로그인 정보가 나타난다. 3. 헤더에는 문제 페이지(현재 페이지), 랭킹 페이지, 마이 페이지로 이동할 수 있는 버튼이 있다. 4. 바디 내부에 검색, 정렬기준, 문제 리스트가 표시된다. 5. 문제 버튼을 누르면 해당 문제 페이지로 이동한다. 6. 하단에는 페이지 번호가 나타난다.
데이터 유형	텍스트
명령 유형	홈 이동, 문제 페이지 이동, 랭킹 페이지 이동, 로그아웃, 검색, 정렬, 문제 선택, 페이지 이동
종료 메시지	N/A


[Problems](#)
[Ranking](#)
[My page](#)


2016311072
Logout

정렬 기준

난이도
최신
제출 수
정답률
랜덤

Fibonacci Numbers
난이도: 쉬움 정답률: 30% 제출 수: 10

Make A Star
난이도: 쉬움 정답률: 30% 제출 수: 10

Shortest Path
난이도: 쉬움 정답률: 30% 제출 수: 10

문제 4
난이도: 쉬움 정답률: 30% 제출 수: 10

문제 5
난이도: 쉬움 정답률: 30% 제출 수: 10


문제 6
난이도: 쉬움 정답률: 30% 제출 수: 10

< 1 2 3 4 5 6 7 8 9 10 >

[그림 1] 문제 목록 페이지

[표 3] 내 정보 페이지

이름	내 정보
목적 / 설명	사용자의 통계 정보를 확인할 수 있는 페이지다.
입력소스 / 출력 대상	클라이언트/호스트서버 호스트서버/클라이언트
범위 / 정확도 / 오차 범위	문제 리스트, 통계 정보 / 모든 문제의 점수와 색 표시, 사용자 통계 정보 제공/ 오차 없음
구성 단위	화면
시간 / 속도	사용자가 내 정보 페이지에 접근했을 때 / 클라이언트와 서버 간 통신 속도
기타 입출력관계	N/A
화면 형식과 구성	1. 헤더 구성은 문제 목록과 동일하다. 2. 바디 왼쪽에는 프로필과 함께 제출, 맞은 문제, 정답, 오답, 시간 초과, 메모리 초과, 런타임 에러, 컴파일 에러 횟수가 제공된다. 3. 바디에는 문제 리스트가 제공되며 100점인 문제는 초록색, 다른 문제는 파란 색으로 표시된다.
데이터 유형	텍스트
명령 유형	홈 이동, 문제 페이지 이동, 랭킹 페이지 이동, 로그아웃
종료 메시지	N/A


[Problems](#)
[Ranking](#)
[My page](#)


2016311072
[Logout](#)



2016311072

제출 수

4931

맞은 문제

1667

맞았습니다

2172

틀렸습니다

1778

시간 초과

471

메모리 초과

198

런타임 에러

244

컴파일 에러

46

Fibonacci Numbers

100 / 100

Make A Star

40 / 100

Shortest Path

0 / 100

Fibonacci Numbers

100 / 100

Make A Star

40 / 100

Shortest Path

0 / 100

Fibonacci Numbers

100 / 100

Make A Star

40 / 100

Shortest Path

0 / 100

Fibonacci Numbers

100 / 100

Make A Star

40 / 100

Shortest Path

0 / 100

Fibonacci Numbers

100 / 100

Make A Star

40 / 100

Shortest Path

0 / 100

Fibonacci Numbers

100 / 100

Make A Star

40 / 100

Shortest Path

0 / 100

[그림 2] 내 정보 이미지

[표 4] 문제 페이지

이름	문제 페이지
목적 / 설명	<p>사용자는 좌측 문제 설명을 읽고 적절한 정답 코드를 작성한다.</p> <p>사용자는 상단의 실행버튼을 눌러 자신의 코드를 실행할 수 있다.</p> <p>사용자는 오류가 발생한 부분, 오류 코드를 확인하고 테스트 결과, 채점결과, 제출결과를 확인하여 실습할 수 있다.</p>
입력소스 / 출력 대상	<p>클라이언트 / 호스트서버</p> <p>호스트서버 / 클라이언트</p>

범위 / 정확도 / 오차 범위	해당 실습 페이지의 사용자 실습 코드 / 파이썬 인터프리터와 사용자 실습 코드에 따른 정확도 / 파이썬 인터프리터와 사용자 실습 코드에 따른 오차
구성 단위	화면
시간 / 속도	사용자가 실행 버튼을 눌렀을 때 / 실행 코드의 복잡도 및 클라이언트와 서버간 통신 속도
기타 입출력관계	1. 해당 코드를 실행하는 파이썬 인터프리터의 결과를 사용자에게 표시 2. 테스트 케이스에 대한 사용자 코드의 출력 표시 3. 사용자의 코드를 pc에 저장 / pc의 코드를 화면에 불러오기 4. 사용자의 코드를 서버에 저장 / 서버의 코드를 화면에 불러오기 5. openAI Codex API를 이용한 코드 설명

<p>화면 형식과 구성</p>	<ol style="list-style-type: none"> 1. 페이지 헤더엔 홈 버튼, 이전/다음 버튼, 현재 위치, 설정 버튼이 존재한다. <ol style="list-style-type: none"> 1-1) 홈 버튼을 누르면 문제 목록 페이지로 돌아간다. 1-2) 이전, 다음 버튼을 누르면 이전, 다음 문제 페이지로 이동한다. 2. 바디 왼쪽에는 문제 설명과 테스트 케이스가 표시된다. 이 구역의 크기는 드래그로 조절할 수 있다. <ol style="list-style-type: none"> 2-1) 문제 설명은 내용이 길면 스크롤 바가 생긴다. 2-2) 테스트 케이스는 검증 버튼을 눌러 사용자 코드에 대한 출력 결과를 확인할 수 있으며 pass / fail 여부가 표시된다. 3. 바디 우측 상단에는 save(1), load, div, reset, copy, save(2), run scoring, submit 버튼이 있다. <ol style="list-style-type: none"> 3-1) save(1) 버튼을 누르면 사용자의 코드를 pc에 저장하는 탐색기 창이 열린다. 3-2) load 버튼을 누르면 탐색기 창이 열려 불러올 코드를 선택할 수 있다. 3-3) div를 누르면 code diff결과를 이분할로 나타낼 수 있다. 이 기능은 submit 후에만 사용 가능하다. 3-4) reset을 누르면 사용자의 코드가 스켈레톤 코드로 초기화된다. 3-5) copy를 누르면 사용자의 코드를 클립보드에 복사한다. 3-6) save(2)를 누르면 사용자의 코드를 서버에 저장한다. 3-7) load(2)를 누르면 사용자의 코드를 서버에서 불러온다. 3-8) run을 누르면 사용자의 코드를 실행한다. 3-9) scoring을 누르면 사용자의 코드를 채점한다. 채점 결과는 하단에 나타난다. 3-10) submit을 누르면 사용자의 코드를 제출한다. 제출 결과는 하단에 나타난다.
------------------	---

	<p>4. 바디 우측 중앙에는 코드를 작성할 수 있는 부분이 존재한다. 가독성을 위해 적절히 텍스트에 색이 입혀진다. 이 구역의 크기는 드래그로 조절할 수 있다.</p> <p>5. 바디 우측 하단에는 여러 결과가 각 탭에 나타난다. 이 구역의 크기는 드래그로 조절할 수 있다.</p> <p>5-1) terminal 탭엔 코드 실행결과가 표시된다.</p> <p>5-2) score 탭엔 코드 채점 결과가 표시된다. scoring 버튼을 누른 직후에만 활성화된다.</p> <p>5-3) result 탭엔 코드 제출 결과가 표시된다. submit 버튼을 누른 직후에만 활성화된다.</p> <p>5-4) explanation 탭엔 코드 제출 결과에 맞는 설명이 표시된다. submit 버튼을 누른 직후에만 활성화된다.</p> <p>5-5) related 탭엔 코드 제출 결과에 맞는 설명 자료가 나타난다. submit 버튼을 누른 직후에만 활성화된다.</p>
데이터 유형	텍스트, 이미지
명령 유형	save(1), load(1), div, reset, copy, save(2), load(2), run, scoring, submit 에 해당하는 명령
종료 메시지	<p>"저장되었습니다."</p> <p>"채점이 완료되었습니다."</p> <p>"제출되었습니다."</p>



Fibonacci Numbers

2일 12시간 30분



Statement

Time Limit: 1.0 seconds
Memory Limit: 512 MB

A Fibonacci sequence is calculated by adding the previous two members of the sequence, with the first two members being both 1.

$f(1) = 1, f(2) = 1, f(n > 2) = f(n - 1) + f(n - 2)$

Your task is to take a number as input, and print that fibonacci number.

```
def fib(n):  
    if n == 0:  
        return 0  
    elif n == 1 or n == 2:  
        return 1  
    else:  
        return fib(n - 1) + fib(n - 2)  
  
n = input()  
print(fib(n))
```

terminal score result explanation related

Example

Input 1

100

Output 1

354224848179261915075

```
100  
354224848179261915075
```

[그림 3] 문제 페이지



Fibonacci Numbers

2일 12시간 30분



Statement

Time Limit: 1.0 seconds
Memory Limit: 512 MB

A Fibonacci sequence is calculated by adding the previous two members of the sequence, with the first two members being both 1.

$f(1) = 1, f(2) = 1, f(n > 2) = f(n - 1) + f(n - 2)$

Your task is to take a number as input, and print that fibonacci number.

```
def fib(n):  
    if n == 0:  
        return 0  
    elif n == 1 or n == 2:  
        return 1  
    else:  
        return fib(n - 1) + fib(n - 2)  
  
n = input()  
print(fib(n))
```

terminal score result explanation related

Example

Input 1

100

Output 1

354224848179261915075

```
Score: 60  
Test Case #1: Pass  
Test Case #2: Fail  
Input: function(3)  
Answer: 77  
Your Output: -71  
Hidden Test Cases: Pass
```

[그림 4] 문제 페이지 확장

Fibonacci Numbers

2일 12시간 30분

Statement

Time Limit: 1.0 seconds
Memory Limit: 512 MB

A Fibonacci sequence is calculated by adding the previous two members of the sequence, with the first two members being both 1.

 $f(1) = 1, f(2) = 1, f(n > 2) = f(n - 1) + f(n - 2)$

Your task is to take a number as input, and print that fibonacci number.

save

load

div

reset

copy

save

load

run

scoring

submit

def fib(n):

if n == 0:

return 0

terminal

score

result

explanation

related

Overall Score

functional score

testcase #1: pass

testcase #2: fail

hidden testcase #1: pass

hidden testcase #2: fail

hidden testcase #3: pass

efficiency score

Line of Codes: 20

Resevation Words: 20

Data Flow Complexity: 30

Control Flow Complexity: 30

readability score

mypy: 20

pylint: 18

eradicate: 20

radon: 20

pycodestyle: 15

Example

Input 1

100

Output 1

354224848179261915075

[그림 5] 문제 페이지 - 제출결과

terminal

score

result

explanation

related

Explain code using AI [Beta]

Note: You can use this 10 times per day. Many results might not be accurate as this is in Beta.

The code is doing the following:

* 1. Initialize _curr and _next to 0 and 1 respectively.
* 2. For each iteration, assign the value of _next to _curr and add _next to _curr.
* 3. Return _curr after n iterations.

[그림 6] 문제 페이지 - 코드설명



Statement

Time Limit: 1.0 seconds
Memory Limit: 512 MB

A Fibonacci sequence is calculated by adding the previous two members of the sequence, with the first two members being both 1.

$f(1) = 1, f(2) = 1, f(n > 2) = f(n - 1) + f(n - 2)$

Your task is to take a number as input, and print that fibonacci number.

[save](#) [load](#)[div](#)[reset](#)[copy](#)[save](#)[load](#)[run](#)[scoring](#)[submit](#)

```
def fib(n):  
    if n == 0:  
        return 0
```

[terminal](#) [score](#) [result](#) [explanation](#) [related](#)

관련 영상

Fibonacci Sequence

34, 55, 89, 144, 233, 377

Mathematics - Fibonacci Sequence and the Golden Ratio

8:39 440KB - 2GB

26 The Organic Chemistry Tutor

This mathematics video tutorial provides a basic introduction into the Fibonacci sequence and the golden ratio. It explains how to ...

Fibonacci Sequence | The Fibonacci Sequence | The Fibonacci Sequence and the Golden Ratio | What Is ...

추천 문제

- [Fibonacci Sequence](#)
- [Recursive Function](#)

학습 자료

- [Fibonacci number - Wikipedia](#)

Example

Input 1

100

Output 1

354224848179261915075

[그림 7] 문제 페이지 - 관련자료

3.2. Hardware Interfaces

[표 5] Client Hardware Interface

Name	Client Hardware
Usage	사용자가 서비스를 이용하기 위해 이용하는 하드웨어
Components	최소한 사용자가 코드를 입력할 수 있는 장치와 서비스의 출력을 볼 수 있는 장치
Specs	일반적으로 PC로 예상되며 최소한 RAM 4기가, 여유 저장 공간 1GB, 2.4Ghz network adapter

[표 6] Server Hardware Interface

Name	Server Hardware (AWS 이용 예정)
Usage	사용자의 여러 요청에 대한 처리를 하기 위한 서버
Components	Amazon API Gateway 와 AWS Lambda 등으로 구성
Specs	AWS plan에 의존

3.3. Software Interfaces

[표 7] Client OS Software Interface

Name	Client OS
Usage	Client Hardware의 OS, 사용자에게서 입력을 받아 서버에게 전달하고 서버의 반응을 사용자에게 제공
Input Source	사용자의 마우스와 키보드
Input Content	사용자의 코드는 데이터 형태의 입력이며 사용자의 마우스 움직임/클릭은 이벤트 형태의 입력
Output Destination	사용자의 모니터
Output Content	서비스의 UI 및 사용자의 코드에 대한 서버의 채점 결과
Protocol/Languages	HTML, Javascript
Data type	Plain text
Supported OS	No constraint

[표 8] Database Software Interface

Name	Database (AWS 이용 예정)
Usage	사용자들의 계정 정보와 각 문제에 대해 사용자들의 과거 코드, 그에 대한 채점 결과 등 모든 정보 저장

Input Source	서버
Input Content	서버에서 사용자로부터 입력 받은 코드, 계정 정보
Output Destination	서버
Output Content	서버가 사용자 별로 제공해야 되는 과거 코드나 채점 결과
Protocol/Languages	sqlite
Data type	sqlite 쿼리
Supported OS	linux

[표 9] Server OS Software Interface

Name	Server OS
Usage	Server Hardware에서의 OS, 사용자에게 받은 정보를 처리 후 데이터베이스에 필요한 정보를 저장하며 사용자에게 처리 결과 제공
Input Source	사용자, 데이터 베이스
Input Content	코드, 이벤트, 계정 정보
Output Destination	사용자, 데이터 베이스
Output Content	서비스 UI, 과거 코드, 채점결과
Protocol/Languages	Mysql, HTML, Javascript

Data type	mysql 쿼리, HTML 문서
Supported OS	Linux

[표 10] API Software Interface

Name	코딩 에디터, 채점, 테스트 케이스 실행 API
Usage	서비스 개발 시 존재하는 오픈 소스를 사용해 적은 노력으로 높은 퀄리티의 서비스 제공
Input Source	사용자, 데이터베이스
Input Content	코드, 테스트 케이스
Output Destination	사용자
Output Content	서비스 UI, 테스트케이스 실행 결과, 채점결과
Protocol/Languages	Python
Data type	Plaintext
Supported OS	No constraint

3.4. Communication Interface

[표 11] Browser Communication Interface

Name	브라우저 어플리케이션
Usage	물리적으로 분리되어 있는 Client 와 Server 간의 통신을 가능하게 함
Input Source	사용자, 서버
Input Content	서버, 사용자가 상호작용하는 모든 데이터
Output Destination	사용자, 서버
Output Content	서버, 사용자가 상호작용하는 모든 데이터
Protocol/Languages	TCP/UDP
Data type	HTML 문서
Supported OS	No constraint
Speed	Client의 네트워크 환경에 의존

4. SYSTEM REQUIREMENTS

4에서는 시스템 기능 서비스와 운영적 제약사항을 기능적, 비기능적 요구사항으로 나누어 기술한다.

4.1. Functional requirements

시스템이 제공하는 기능은 크게 5가지로, 회원가입, 로그인, 문제 선택, 문제 별 코드 작성, 코드 실행 및 채점이 있다. 시스템의 각 기능에 대해 기능, 설명, 입력, 출력, 처리, 조건을 표 양식으로 표현한다.

기능1. 회원가입

기능설명	사용자 식별에 필요한 데이터를 사용자에게 입력 받아 데이터베이스에 저장한다.		
입력	이름, 메일, 비밀번호	처리	조건 충족 여부 확인 및 충족 시 정보 저장
출력	조건 충족 여부 출력	조건	각 필드의 형식을 지켜야 함.

기능2. 로그인

기능설명	사용자 데이터를 불러오기 위해 필요한 사용자 식별 과정이다.		
입력	메일, 비밀번호	처리	데이터베이스에 입력된 데이터와 일치하는 데이터가 있는 지 확인하여 로그인 성공 여부를 결정한다.
출력	로그인 성공 여부 출력	조건	각 필드의 형식을 지켜야 함.

기능3. 문제 선택

기능설명	출제자가 데이터베이스에 입력한 문제 목록을 사용자의 각 문제 현황과 함께 리스트 형식으로 보고, 사용자가 문제를 선택한다.		
입력	사용자 식별 번호	처리	데이터베이스에 있는 문제에 대한 데이터와, 각 문제에 대한 사용자의 현황 데이터를 가져오는 쿼리를 수행한다.
출력	문제들과 문제 별 사용자의 문제 현황	조건	사용자 식별 번호가 유효해야함.

기능4. 문제 별 코드 작성

기능설명	사용자가 선택한 문제에 대하여 코드를 작성 및 저장하여 문제풀이를 수행할 수 있도록 한다.		
입력	사용자 식별 번호, 문제 식별 번호, 사용자가 입력한 코드	처리	사용자가 선택한 문제에 대한 코드를 저장한다.
출력	사용자에 대한 문제 현황, 문제 정보	조건	사용자 식별 번호와 문제 식별 번호가 유효 해야함.

기능5. 코드 실행 및 채점

기능설명	사용자가 작성한 코드를 실행시키고, 상태를 반환한다.		
입력	사용자 식별 번호, 문제 식별 번호, 사용자가 입력한 코드	처리	코드를 실행시켜 테스트 케이스와같은 입력에 대해 출력 값이 같은 지 비교하고, 이 결과를 반환 받아 사용자가 선택한 문제에 문제 현황 데이터를 업데이트 한다.
출력	실행결과 에러 여부 및 에러 상세 메세지, 테스트케이스별 통과 여부	조건	시스템이 지원하는 언어와 문법으로 작성된 코드여야 함.

4.2. Non functional requirements

4.2.1, 4.2.2, 4.2.3에서 non-functional requirement의 요소로서 product, organizational, external requirement를 각각 기술한다.

4.2.1 product requirement

A. usability requirement

- 회원가입과 로그인을 통해 사용자를 식별하므로 사용자 데이터를 이용하여 맞춤형 화면을 제공해야 한다.
- 코드 작성 기능에 대하여 제공하는 세부 기능에 대해 아이콘 등을 활용하여 직관적인 UI를 제공한다.
- 입력 받은 코드에 대하여 인덴트와 같은 문법 오류를 줄일 수 있도록 하이라이팅 기능을 제공한다.

B. efficiency requirement

- 한 화면에 보이는 데이터의 양을 조절하여 데이터를 로딩 하는 데 걸리는 지연시간을 줄인다.

C. dependability requirement

- 사용자가 코드를 저장하고, 이를 토대로 문제 현황을 업데이트하는 과정에서 데이터의 일관성이 깨지지 않도록 한다.

D. security requirement

- 회원가입 및 활동 내역에 대한 사용자의 정보가 유출되지 않도록 한다.
- 외부로부터 관리자 서버를 보호한다.

4.2.2 organizational requirement

A. environment requirement

- 핵심이 되는 기능인 코드 작성 기능의 경우 모바일로 코드를 입력하는 경우가 많기 때문에 모바일 버전 보다는 PC 웹 브라우저 해상도에 맞는 UI만 제공하도록 한다.
- 코드를 실행하는 디바이스는 사용자의 디바이스가 아니기 때문에 사용자의 디바이스는 인터넷 연결이 필요하다.
- 사용자는 인터넷 연결을 위한 브라우저 및 코드를 입력하는 데 필요한 키보드를 갖추어야 한다.

B. operational requirement

- 프론트엔드는 React.js, HTML, CSS, 백엔드는 Django(Python), REST API, 데이터베이스는 SQLite을 이용하여 개발한다. 배포 및 데이터베이스 서버는 AWS 서비스를 활용하여 구축한다.
- 사용자로부터 입력 받는 기능의 경우 모나코 라이브러리를 이용한다.
- 사용자의 코드를 실행 및 채점하는 기능의 경우 unittest, copydetect, multimetric, pylama, openAPICodex, diff2html과 같은 라이브러리를 이용한다.

C. development requirement

- 사용자에게 대해 각 문제의 학습률, 오답률과 같은 문제 현황 데이터를 계산 후 데이터베이스에 저장한다.
- 사용자에게 인터넷에 연결된 웹 브라우저에서 별도의 툴 없이 코드를 작성 및 테스트할 수 있는 환경을 제공한다.

4.2.3 external requirement

A. regulatory requirement

- 데이터베이스에 저장된 코드의 경우 출처자 정보 등의 출처 명시하고 저작권을 보장한다.
- 회원 가입 시 입력 받은 개인정보 활용시에는 개인 정보 활용 동의를 받아야만 한다.

B. ethical requirement

- 사용자가 입력한 코드에 대해 부정확한 채점 결과로 사용자에게 혼돈을 주지 않도록 해야 한다.

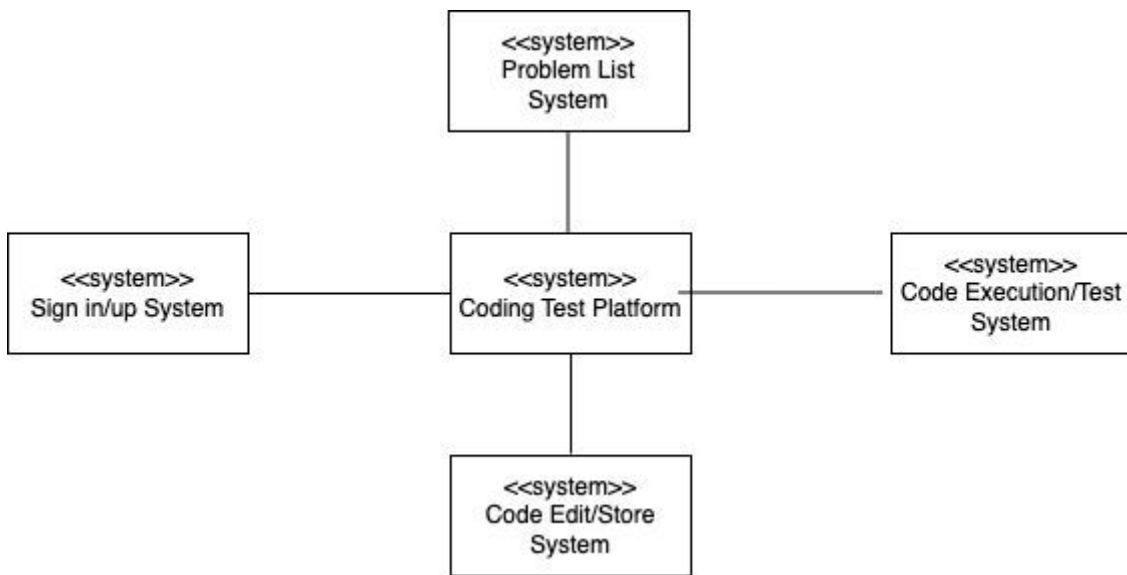
- 사용자의 문제 풀이 현황 등 동의하지 않은 정보를 함부로 공개하지 않도록 한다.

C. safety/security requirement

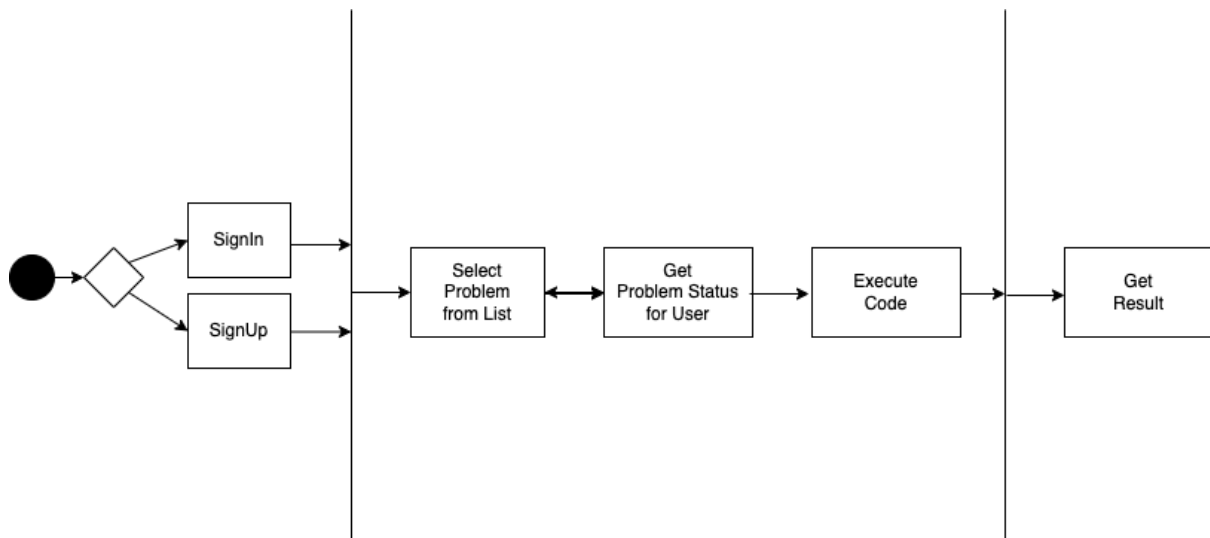
- 회원 가입 및 서비스 이용 도중 생성된 사용자 데이터를 외부로부터 보호해야 한다.

4.3. Organizing system flow

4.3.1 context model



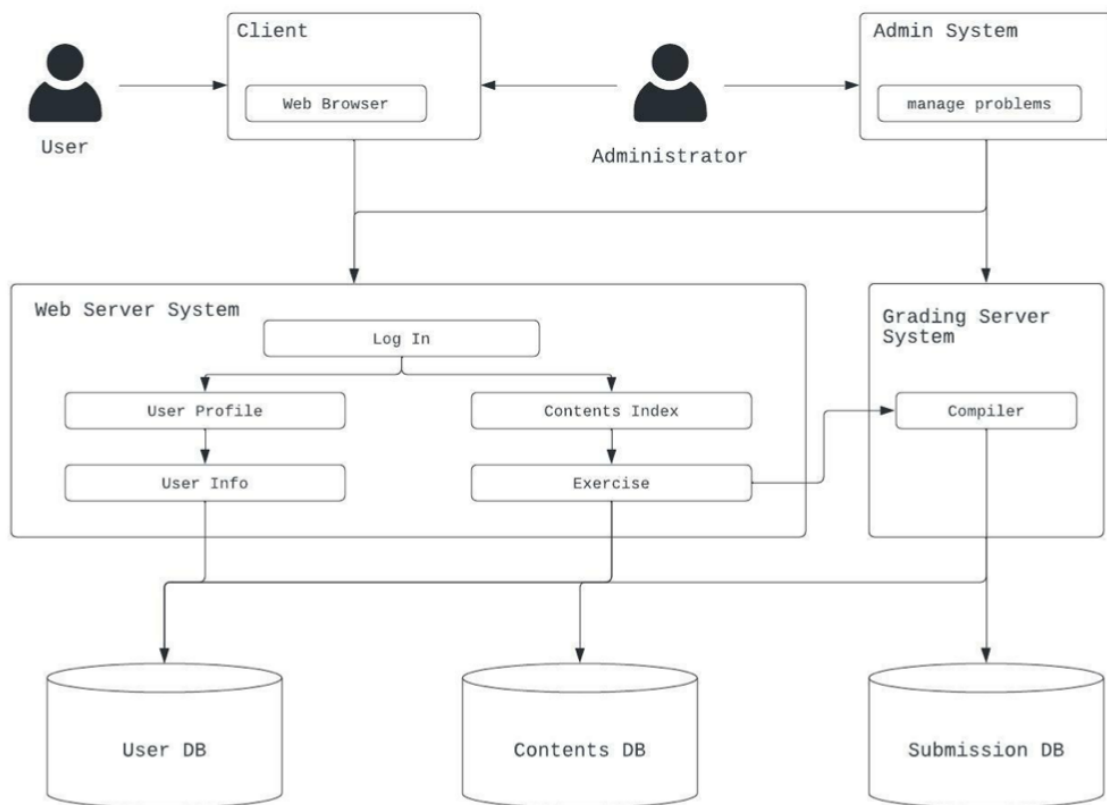
4.3.2 process model



4.4. System Architecture

시스템의 구성 요소와, 이들의 역할과 상호작용을 기술하여 전체적인 구조를 이해하도록 한다.

전체 시스템은 크게 사용자, 시스템 관리자, 서버, 데이터베이스의 구성요소로 이루어져 있다. 사용자는 웹 브라우저를 이용하여 웹 서버에 접속한 후 서비스를 이용한다. 관리자는 웹 서버에 서비스에 필요한 데이터를 관리한다. 웹 서버는 로그인, 회원가입, 코드작성 등의 기능을 제공하며, 코드 실행 및 테스트 케이스 수행에 대한 채점 기능의 경우 별도의 서버를 활용할 수 있다. 데이터베이스는 정보의 종류에 따라 유저, 문제, 문제 현황 로 데이터베이스를 분류하여 저장할 수 있다.



5. SUPPORTING INFORMATION

5.1 software requirement specification

본 소프트웨어 명세서는 IEEE Recommended Practice for Software Requirements Specifications 문서를 기준으로 작성되었다. 본 서비스에 맞는 요구 조건을 작성하기 위해 원본 양식에서 일부 추가되거나 제외된 바 있다.

5.2. Document history

[표 17] 문서 이력

Date	Version	Description	writer
2022/10/26	0.1	1	지준현
2022/10/26	0.1	2	김성학
2022/10/26	0.1	3.1	한희근, 신정환
2022/10/26	0.1	3.2~3.4	김영석
2022/10/26	0.1	4	김세란
2022/10/26	0.1	5	지준현