

Requirement Specification

소프트웨어공학개론 4조

제출일	23.04.30	그룹	4조
과목	소프트웨어공학개론	담당교수	이은석 교수
이름	김영우	학번	2017314333
이름	김준형	학번	2018312707
이름	박충현	학번	2017310127
이름	송지은	학번	2018315379
이름	조성현	학번	2020311119
이름	조현준	학번	2017314593



목차

1	<i>Preface</i>	5
1.1	Objective	5
1.2	Readership	5
1.3	Document Structure	5
A.	Preface	5
B.	Introduction	5
C.	Glossary	6
D.	User Requirement Definition	6
E.	System Architecture	6
F.	System Requirement Specification	6
G.	System Model	6
H.	System Evolution	6
I.	Appendix	7
J.	Index	7
2	<i>Introduction</i>	7
2.1	Objective	7
2.2	Needs & Background	7
2.3	System Overview	9
2.4	Expectations	10
A.	학생	10
B.	강사	10
3	<i>Glossary</i>	10
3.1	Objective	10
3.2	Term Definition	10
3.3	Acronyms and Abbreviations	11
4	<i>User Requirement Definition</i>	11

4.1	Objective	11
4.2	Functional Requirement	11
A.	회원가입	11
B.	로그인	12
C.	문제 추천	12
D.	코드 작성 및 제출	12
E.	코드 피드백 확인	12
F.	풀 문제 확인	12
G.	학생들이 작성한 코드 확인	13
H.	ChatGPT를 사용한 학생 코드 분석 및 시각화	13
I.	피드백 작성과 제출	13
4.3	Nonfunctional Requirement	13
A.	Product Requirement	13
B.	Organization Requirement	14
C.	External Requirement	15
5	System Architecture	15
5.1	Objective	15
5.2	System Architecture	17
5.3	Sub system	18
A.	사용자 시스템	17
B.	문제 추천 시스템	18
C.	문제 풀이 시스템	19
D.	코드 분석 시스템	21
E.	학생 관리 시스템	21
6	System Requirement Specification	22
6.1	Objective	22
6.2	Functional Requirement	22
A.	회원가입	23
B.	로그인	23

C. 문제 추천	23
D. 코드 작성 및 제출	24
E. 코드 피드백 확인	25
F. 푼 문제 확인	25
G. 학생들이 작성한 코드 확인	26
H. ChatGPT를 사용한 학생 코드 분석 및 시각화	26
I. 피드백 작성과 제출	26
6.3 Nonfunctional Requirement	27
A. Product Requirement	27
B. Organization Requirement	28
C. External Requirement	28
6.4 Scenario	29
A. 회원가입	29
B. 로그인	30
C. 문제추천	30
D. 코드 작성 및 제출	31
E. 코드 피드백 확인	32
F. 푼 문제 확인	32
G. 학생들이 작성한 코드 확인	33
H. ChatGPT를 활용한 학생 코드 분석 및 시각화	34
I. 피드백 작성과 제출	34
7 System Model	35
7.1 Objective	35
7.2 Context Models	35
A. Context Model	36
B. Process Model	36
7.3 Interaction Models	39
A. Use Case Model	39
B. Tabular Description	41
C. Sequence Diagram	43

7.4	Structural Models	46
A.	Class Diagram	46
7.5	Behavioral Models	47
A.	Event-driven Diagram	47
8	<i>System Requirement Evolution</i>	50
8.1	Objective	50
8.2	Assumption and Limitation	50
A.	서비스 완전 자동화 문제	51
B.	오픈소스 코딩 문제의 유한성	51
8.3	Evolutions for Hardware and Software	51
8.4	Evolutions for Teacher	51
8.5	Evolutions for Student	52
9	<i>Appendix</i>	52
9.1	Objectives	52
9.2	Database Requirements	52
9.3	Database Detail	53
10	<i>Index</i>	53
10.1	Objectives	53
10.2	Table Index	53
10.3	Figure Index	54
10.4	Diagram Index	54
11	<i>Reference</i>	55

1 Preface

1.1 Objective

이 장에서는 본 문서의 예상된 독자, 문서의 구조, 각 단원에서 전달하고자하는 사항에 대해 보다 세부적으로 설명한다.

1.2 Readership

A. User Requirement Readership

시스템이 제공하는 서비스와 운영 제약 조건에 대해 서술하며 사용자가 쉽게 이해하기 위해 자연어, 도표 등을 활용하여 작성한다. 주된 독자로는 **Client manager**과 **contractor manager**를 상정한다.

B. System Requirement Readership

시스템의 기능, 서비스 및 운영 제약 조건에 대한 자세한 설명을 서술한다. 계약의 일부가 될 수 있으므로 최대한 상세하게 기술해야 한다. 시스템 개발 과정에 참여하는 개발자, **architects**, **system**을 계약하는 **client** 등이 주된 독자가 된다.

1.3 Document Structure

A. Preface

본 문서의 예상된 독자, 문서의 구조, 각 단원에서 전달하고자하는 사항에 대해 보다 세부적으로 설명한다.

B. Introduction

본 문서 시스템의 필요성과 시스템의 기능을 간략히 소개한다. 다른 시스템과 어떻게 작동하는지, 시스템이 조직의 비즈니스 또는 전략적 목표에 어떻게 적합하는지 설명한다.

C. Glossary

본 문서에서 사용하는 기술 용어들을 정의한다. 다양한 배경을 가진 독자들이 본 문서를 읽는데 어려움이 없도록 가능한 모든 용어들을 상세하게 정의한다.

D. User Requirement Definition

사용자에게 제공되는 서비스를 설명한다. 시스템에 대한 기능적 요구사항 및 비기능적 요구사항을 기술하며 사용자에게 이해를 돕기 위해 자연어, 그림, 도표 등을 활용한다.

E. System Architecture

본 시스템 구조 개요를 높은 수준으로 서술한다. 이는 시스템 모듈들에 기능이 분포되어 있는지 보여주며 재사용되는 요소는 강조되어 표현한다. 개발자가 시스템의 전체적인 구조를 이해하는 데에 도움을 준다.

F. System Requirement Specification

기능적 요구사항 및 비기능적 요구사항에 대해 상세히 서술한다. 필요한 경우, 비기능적 요구사항에 대한 세부적 정보를 추가할 수 있다.

G. System Model

시스템 구성 요소, 시스템 및 시스템 환경 간의 관계를 설명한다.

H. System Evolution

시스템이 기반으로 하는 가정과 하드웨어 진화, 사용자 요구사항의 변화 등으로 인한 예상 변화에 대해 설명한다. 시스템 디자이너는 이를 통해 시스템의 미래 변경

사항을 제한할 수 있는 디자인 결정을 피할 수 있다.

I. Appendix

개발되고 있는 어플리케이션과 관련된 세부 정보를 제공한다.

J. Index

본 문서에 사용된 그림, 다이어그램, 기능 등의 인덱스가 포함된다.

2 Introduction

2.1 Objective

본 문서 시스템의 필요성과 시스템의 기능을 간략히 소개한다. 다른 시스템과 어떻게 작동하는지, 시스템이 조직의 비즈니스 또는 전략적 목표에 어떻게 적합하는지 설명한다.

2.2 Needs & Background

현재 대한민국에서 가장 대표적으로 이용되는 프로그래밍 학습 사이트는 백준과 프로그래머스이다. 이 두 사이트는 알고리즘 및 프로그래밍 학습을 위한 다양한 문제를 제공하고 있으며, 사용자들이 자신의 코드를 제출하면 자동으로 채점해주는 기능을 제공한다. 하지만 이러한 사이트들도 여전히 한계점을 가지고 있다. 대부분 단순한 채점만 지원하고 각 사이트에서 제공하는 학습 커리큘럼에 한정되어 있다.


```
def fibonacci(n):  
    if n <= 1:  
        return n  
    else:  
        return fibonacci(n-1) + fibonacci(n-2)
```

Codemy는 보완된 기능을 기반으로 코딩 교육 서비스를 개발할 계획이다. ChatGPT를 활용하여 학생이 작성한 코드를 상세하게 분석하고, 분석 결과를 기반으로 강사가 정확한 피드백을 제공한다. 이를 통해 학생의 개선점을 파악하고, 보다 효과적인 학습을 도모할 수 있다. 또한, 학생이 푼 알고리즘 문제들을 바탕으로 해당 학생의 수준을 파악하고, 적절한 난이도의 새로운 문제를 추천하는 기능을 개발할 예정이다.

2.3 System Overview

교육 효율을 높이기 위해 개발된 학원 코딩 교육 서비스이다. 이 서비스를 사용하는 학원 강사들은 학생들이 코드를 제출한 후 ChatGPT에 입력하여 1차 피드백을 받게 된다. 강사는 받은 1차 피드백을 검토한 후 추가적인 피드백을 남겨 학생에게 전달한다. 이를 통해 각 학생의 능력을 시각화하고 데이터 분석을 통해 학생들의 강점과 약점을 파악하여 보완할 수 있게 된다. 또한 ChatGPT는 학생들이 감당할 수 있는 수준의 문제를 추천하여 학생들이 문제를 풀고 학습할 수 있는 데이터 기반 서비스도 제공한다. 해당 프로젝트에서 구현하고자 하는 사항은 다음과 같다.

- a. **Efficiency:** 코드의 효율성. 코드가 돌아가는 속도를 빨라지게 하거나, 사용되는 메모리를 줄이는 등의 개선점을 보여준다. 더 나은 알고리즘을 사용하거나 반복되는 계산을 줄이거나 더 나은 자료구조를 사용하는 등의 방법으로 시간복잡도와 공간복잡도를 더 효율적으로 만들어준다.
- b. **Readability:** 코드의 가독성. 다른 개발자들이 읽기 쉽게해서 유지보수 하기에 쉽게 만든다. 변수와 함수이름을 정해주거나 필요한 부분에 주석을 달거나 하는 등의 개선점을 알려준다.
- c. **Correctness:** 코드가 입력받은 원래 코드와 같은 결과를 출력하는지 확인한다. 특히 케이스들과 에러부분을 같이 수정하는지 확인한다.
- d. **Scalability:** 입력값의 크기가 달라졌을때 코드가 마찬가지로 효율적인지 확인한다. 연산을 병렬화하고 작업을 더 잘게 쪼개는 등의 방법이 있다.
- e. **Modularity:** 코드를 여러번 쓰일 수 있는 작은 부분들로 나눠서 중복되는 코드들을 줄이고 코드의 크기를 작게 만들 수 있다.

- f. **Security:** 코드를 더 안전하게 만든다. 코드삽입 공격을 막기 위해 입력값이 적절한 형식인지 확인한다. 민감한 정보들이 제대로 암호화 되어 저장되는지 확인한다.

2.4 Expectations

A. 학생

학생은 더욱 다양한 문제를 접하며, 자신의 실력을 더욱 발전시킬 수 있을 것이다. 또한, 학생의 학습 상황을 시각화하여 제공함으로써, 학생 스스로 자신의 학습 상황을 파악하고 개선해 나갈 수 있도록 도울 수 있다. 이를 통해 학생은 학습에 대한 불안감을 덜어내며, 더욱 적극적인 학습 태도를 유지할 수 있을 것으로 기대된다.

B. 강사

강사는 학생에게 전달할 1차 피드백을 받고 학생들의 수준을 더욱 편리하게 시각화해서 확인할 수 있다. 이를 통해 각 학생의 장단점을 파악하여 실력을 향상하는 데에 도움을 줄 수 있다. 또한, 각 학생에게 내줄 과제도 추천받으며 수업 커리큘럼의 퀄리티를 높일 수 있다.

3 Glossary

3.1 Objective

본 문서에서 사용하는 기술 용어들을 정의한다. 다양한 배경을 가진 독자들이 본 문서를 읽는데 어려움이 없도록 가능한 모든 용어들을 상세하게 정의한다.

3.2 Term Definition

용어	정의
학생(Student)	본 서비스를 이용하여 학습하는 사람을 말한다.
강사(Teacher)	본 서비스를 이용하는 학생들을 관리하는 사람을 말한다.

문제	학생이 풀어야 하는 알고리즘 문제를 의미한다.
추천 문제	학생이 푼 문제들을 기반으로 ChatGPT가 추천해주는 문제를 의미한다.
Codemy	본 서비스를 제공하는 시스템의 이름이다.
피드백	강사들이 학생들이 푼 문제의 코드에 대해 ChatGPT의 의견을 받고, 이를 수정하여 학생에게 코드에 대한 의견을 남기는 것을 의미한다.
ChatGPT	학생들이 푼 문제들을 기반으로 문제를 추천하고, 코드에 대한 피드백을 제공하는 AI이다.
시각화	코드에 대한 지표를 다양한 형태의 차트로 보여주는 것을 의미한다.

Table 1 용어 정의

3.3 Acronyms and Abbreviations

용어	정의
URL	Uniform Resource Locator. 웹 문서의 각종 서비스를 제공하는 서버들에 있는 파일의 위치를 표시하는 표준을 말한다.
DB	Data Base. 여러 사람에 의해 공유되어 사용될 목적으로 통합하여 관리되는 데이터의 집합을 말한다.

Table 2 약어

4 User Requirement Definition

4.1 Objective

사용자에게 제공되는 서비스를 설명한다. 시스템에 대한 기능적 요구사항 및 비기능적 요구사항을 기술하며 사용자에게 이해를 돕기 위해 자연어, 그림, 도표 등을 활용한다.

4.2 Functional Requirement

A. 회원가입

사용자는 서비스를 이용하기 위해 개인 정보를 입력하고 회원가입을 해야 한다. 회원 정보는 각 사용자를 식별할 수 있는 값을 포함하고 있어야 한다. 입력받는 개인 정보에는 이메일, 비밀번호, 이름, 직업(학생 또는 강사), 백준 아이디가 있다.

B. 로그인

사용자가 서비스 이용을 위해 이메일과 비밀번호를 입력하여 학생 또는 강사 권한을 얻는다. 계정 권한에 따라 다른 서비스를 제공받는다. 학생 권한의 경우 추천 문제 받기, 코드 피드백 받기, 푼 문제에 대한 요약 기능을 사용할 수 있다. 강사 권한의 경우 학생이 제출한 문제 피드백 하기, ChatGPT를 통한 학생 코드 분석 도구 기능을 사용할 수 있다.

C. 문제 추천

추천 받는 학생이 지금까지 푼 문제와 받은 피드백을 바탕으로 새롭게 풀 문제를 제공해준다. 학생은 각 문제에 대한 요약된 정보(난이도, 사용 알고리즘)를 확인하여 선호에 따라 풀 문제를 선택할 수 있다.

D. 코드 작성 및 제출

해당 서비스에서는 코드 실행, 채점 기능을 제공하지 않는다. 따라서 문제와 채점을 제공해주는 백준의 하위 서비스로서 사용자에게 피드백을 제공해주는 기능을 제공한다. 사용자는 자신의 IDE를 사용하여 문제에 대한 코드를 작성하고 백준에 제출하여 채점한 후 제출한 코드를 이 서비스에 제출하면 코드에 대한 피드백을 제공한다.

E. 코드 피드백 확인

학생이 제출한 코드를 강사가 확인하고, 남긴 피드백을 사용자가 확인할 수 있다. 피드백에는 코드에 대한 정량적인 수치(시간 복잡도, 공간 복잡도, 가독성 등)와 강사의 의견(코드에 대한 전체적인 평가, 앞으로 학습하면 좋을 것 같은 방향)이 포함되어 있다.

F. 푼 문제 확인

학생이 지금까지 푼 문제에 대한 요약을 확인할 수 있다. 단순한 문제 나열을 넘어서 푼 문제와 제출한 코드를 바탕으로 통계를 만들고 이를 시각화해준다. 여기에는 문제들의 알고리즘 분포, 학생이 작성한 코드들의 강점과 약점 등이 포함되어있다.

G. 학생들이 작성한 코드 확인

강사는 학생이 작성한 코드를 확인할 수 있다. 코드는 학생이 제출한 문제 번호에 매치되어 제공되며 학생이 작성하는 코드의 특성을 파악하기 위해 해당 학생이 제출한 모든 코드를 한 번에 보기 위한 UI가 제공된다.

H. ChatGPT를 사용한 학생 코드 분석 및 시각화

학생이 제출한 코드는 ChatGPT의 여러 단계의 쿼리를 거쳐 정량적, 정성적인 분석을 받는다. 해당 분석 내용을 강사가 해석하기에는 시간이 오래 걸리므로 파싱을 통해 필요한 정보만을 정리하여 보기 편한 차트로 강사에게 보여준다.

I. 피드백 작성과 제출

ChatGPT를 사용한 분석 시각화로 빠르게 학생 코드를 파악할 수 있다. 하지만 ChatGPT의 한계로 부분적인 오류가 있을 수 있기 때문에 강사는 분석 내용을 부분 수정하고 정리하여 학생에게 피드백을 작성해야한다. 작성을 완료한 피드백은 학생에게 전송되며 학생은 이를 확인하고 학습에 적용한다.

4.3 Nonfunctional Requirement

A. Product Requirement

A.1 Usability Requirement (UI)

학생 페이지의 UI는 자신의 부족한 부분이 어떤 점이고 개선하기 위한 솔루션을 한 눈에 보기 편하게 제시해야한다. 강사 페이지의 UI는 ChatGPT로부터 분석된 강점 및 약점을 한

눈에 이해할 수 있게 시각화 해야하며 강사가 피드백을 가능한 빠르고 정확하게 작성할 수 있도록 해야한다.

A.2 Efficiency Requirement

ChatGPT로 보내는 쿼리는 가능한 적어야하며 문제 추천 페이지와 같이 페이지를 열 때마다 쿼리를 보낼 필요가 없는 경우 DB에 미리 쿼리 결과를 저장하여 빠른 결과를 보여주어야 한다. 코드에 대한 피드백은 병렬적인 ChatGPT 쿼리를 통해 가능한 빠른 시간 내에 사용자에게 보여져야한다.

A.3 Dependability Requirement

사용자에 대한 정보가 오류 없이 저장되어야 한다. 적절한 이메일과 비밀번호를 입력했을 때 정상적으로 로그인 되어야 한다. 학생이 제출한 코드 원본이 오류 없이 저장되어야 하며 강사에게 보내져야 한다. ChatGPT로 보내는 쿼리는 코드를 평가하기에 가장 정확도가 높은 방식으로 이루어져야하며 강사에게는 왜곡 없이 코드 분석 결과를 보여주어야 한다. 강사는 학생을 객관적으로 평가할 수 있는 환경에서 피드백을 작성하며 강사의 피드백 의도가 학생에게 그대로 전달되어야 한다.

A.4 Security Requirement

사용자의 개인정보는 암호화되어 보관해야 한다. 사용자의 비밀번호는 해당 시스템에서도 확인할 수 없어야 하며 사용자를 식별하는데만 사용되어야 한다. 외부의 요청으로 사용자의 정보가 외부로 유출되는 일은 없어야 한다.

B. Organization Requirement

B.1. Environmental Requirement

Codemy는 온라인 웹서비스이다. PC의 Chrome 브라우저에 최적화되어 있으며 모바일 서비스는 지원하지 않는다.

B.2. Operational Requirement

React와 Spring을 이용해 개발한다. 개발 언어는 JavaScript와 Java이며 Solved.ac API와 ChatGPT API를 사용하여 사용자, 문제에 대한 정보와 코드에 대한 피드백을 제공받는다.

B.3. development Requirement

학생이 추천받은 문제와 제출한 코드 받은 피드백은 DB에 저장된다. ChatGPT와 강사가 평가한 피드백은 제출한 코드에 매칭되어 저장된다. 코드 제출하고 피드백을 시각화하는 서비스의 특성에 의해 모바일 페이지는 제공하지 않으며 PC에 최적화되어 개발된다.

C. External Requirement

C.1. Regulatory Requirement / Legislative Requirement

학생의 정보는 강사에게 부분 공유될 수 있지만 비밀번호의 경우 사용자 본인만 알 수 있다. 사용자가 자신의 계정을 삭제하기 원하는 경우 삭제할 수 있으나 동일한 unique key를 사용한 계정을 새로 생성할 수 없다.

C.2. Ethical Requirement

Codemy는 어떤 사용자도 사용할 수 있는 서비스이다.

C.3. safety/security requirement

개인정보보호법에 따라 고객 개인정보를 안전하게 보호하여야 한다.

5 System Architecture

5.1 Objective

본 시스템 구조 개요를 높은 수준으로 서술한다. 이는 시스템 모듈들에 기능이

분포되어 있는지 보여주며 재사용되는 요소는 강조되어 표현한다. 개발자가 시스템의 전체적인 구조를 이해하는 데에 도움을 준다.

5.2 System Architecture

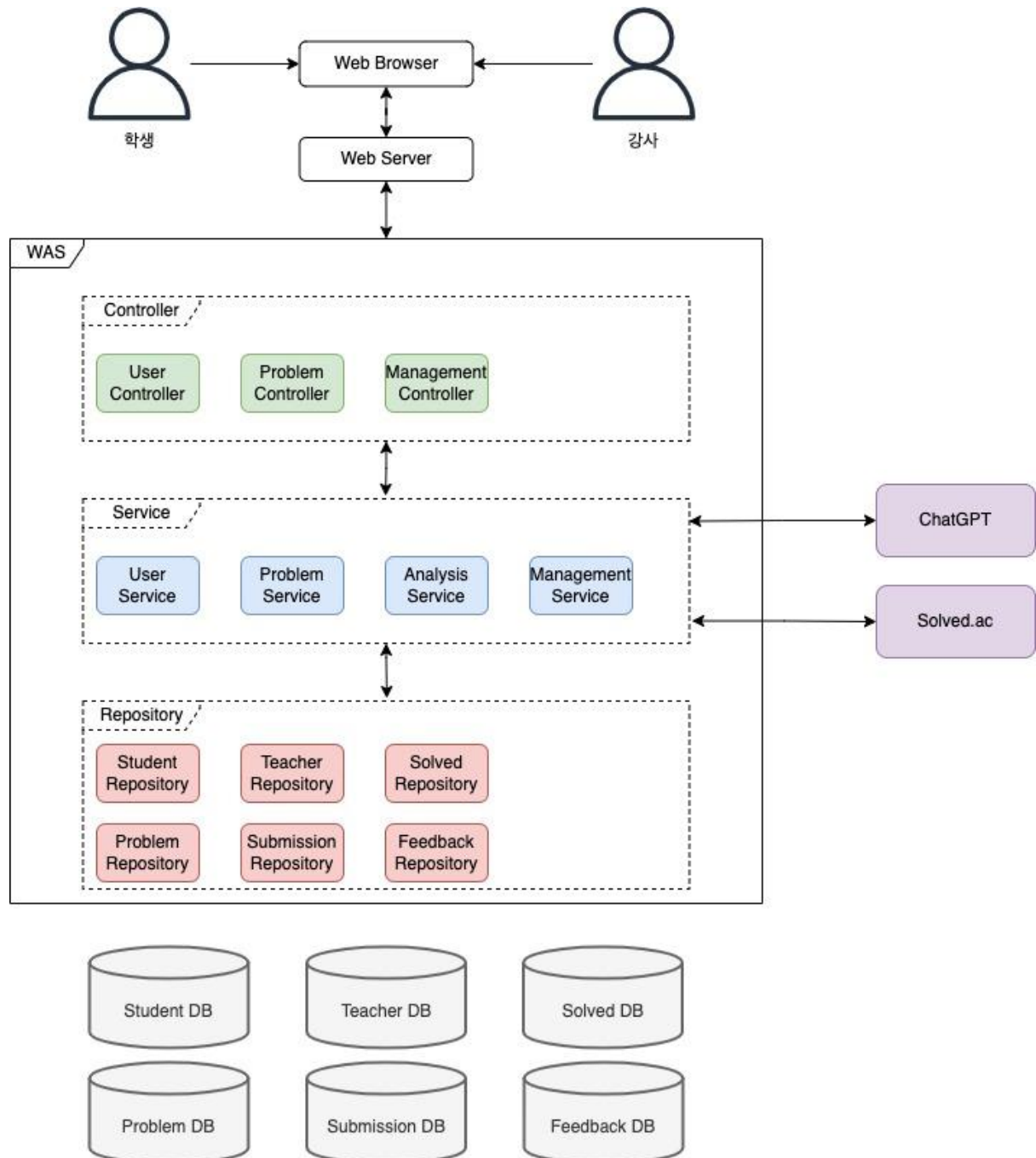


Diagram 1 System Architecture: 전체 시스템 구조

전체 시스템 구조는 Web Server와 WAS 및 DB로 구성된 구조이다. 사용자는 Web Browser로 접속하여 시스템을 이용할 수 있다. WAS는 Spring Boot Application으로, 사용자가 요청한 HTTP 요청을 받아들이고, 이를 처리하여 응답을 반환하는 **Controller**, 비즈니스 로직을 구현하는 **Service**, 데이터베이스와의 연동을 처리하는 **Repository**로 구성되어 있다.

5.3 Sub system

A. 사용자 시스템

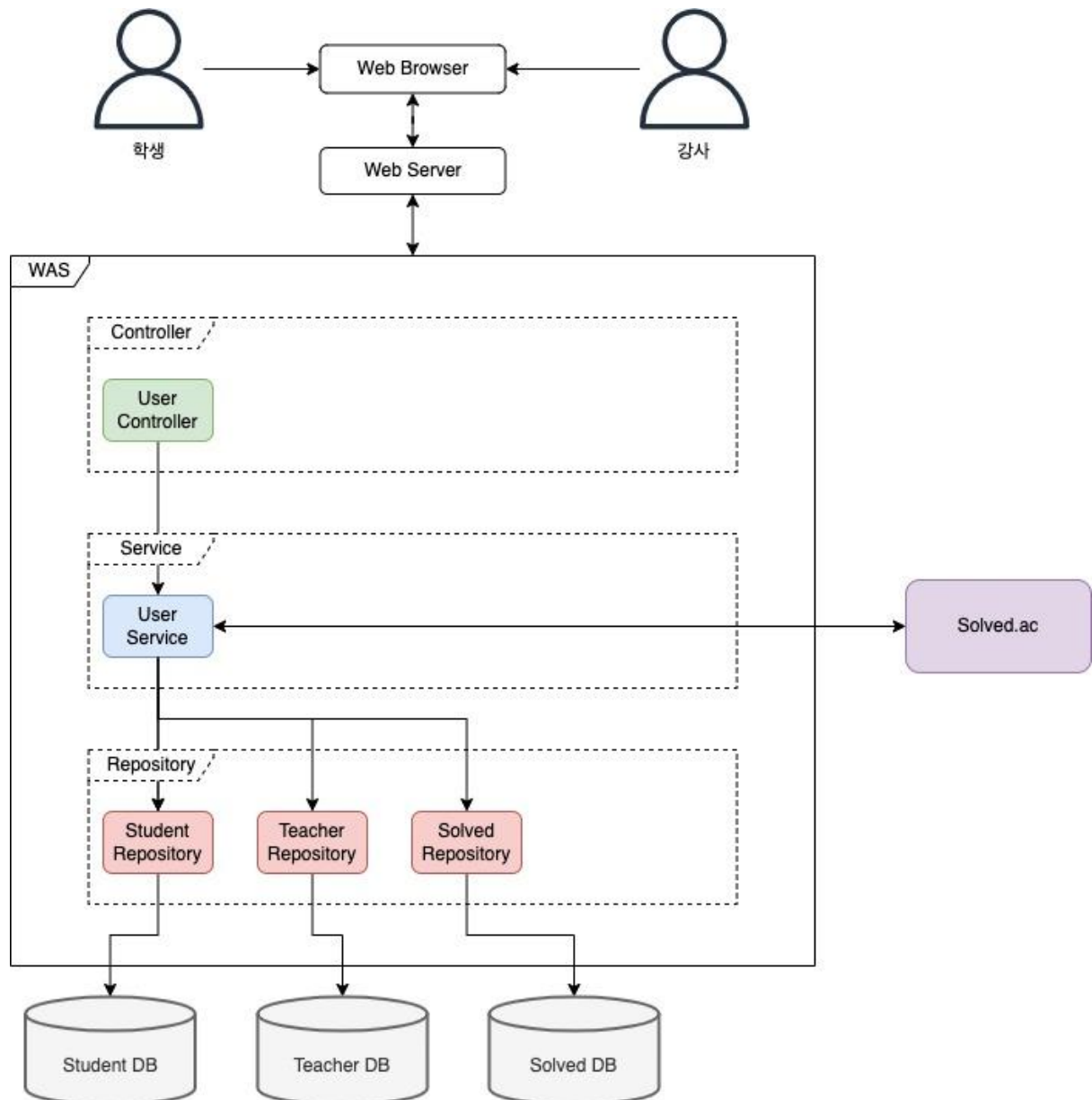


Diagram 2 System Architecture: 사용자 시스템

사용자 시스템은 학생 및 강사의 회원가입 및 로그인과 같은 전반적인 사용자 관련 요구사항을 담당하는 시스템이다. 학생 회원가입 시, 백준 ID를 입력 받아 지금까지 풀었던 문제들을 조회하고 저장한다.

B. 문제 추천 시스템

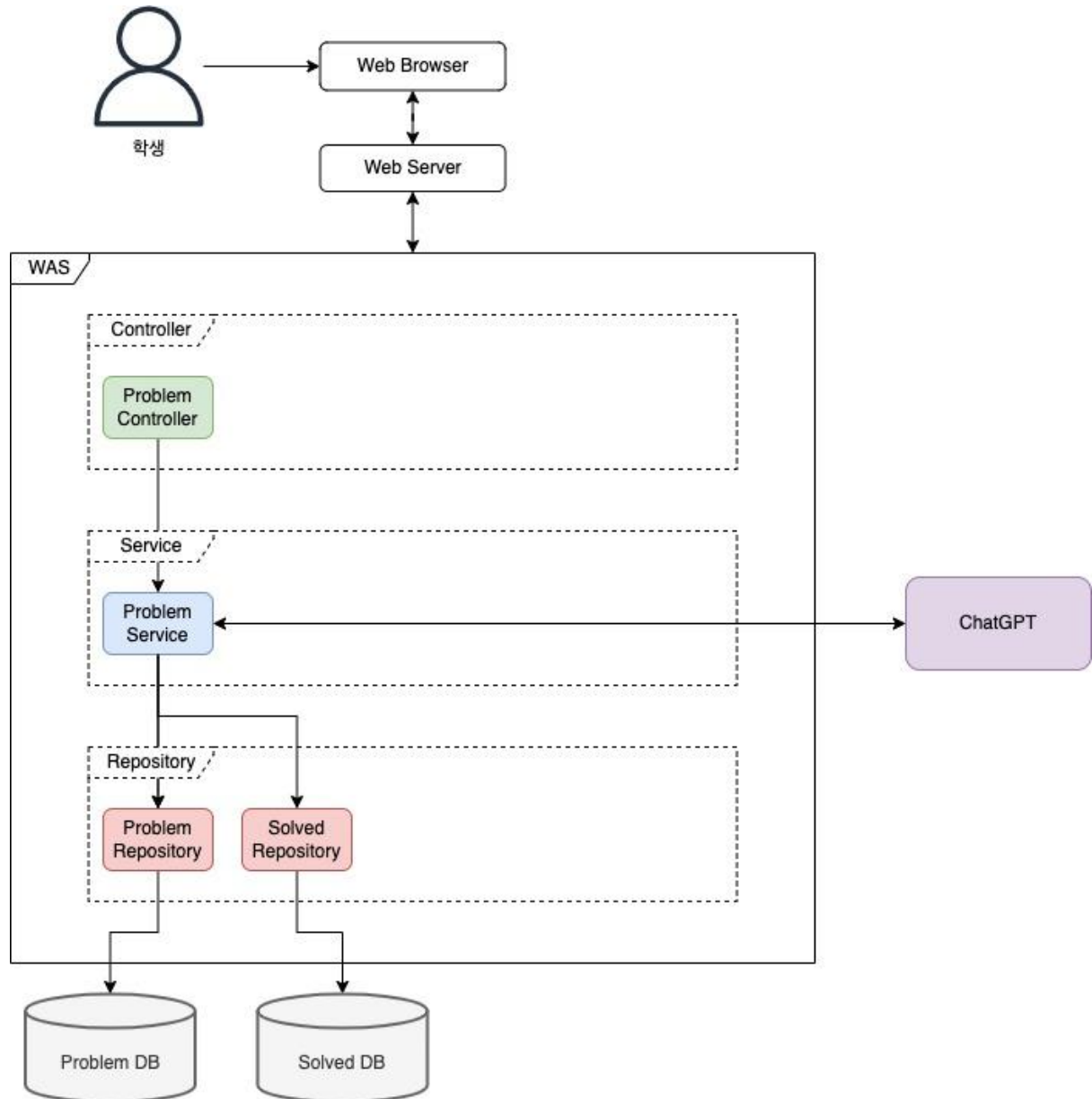


Diagram 3 System Architecture: 문제 추천 시스템

문제 추천 시스템은 학생이 메인 페이지에 접속했을 때 지금까지 풀었던 문제들을 기반으로 ChatGPT를 이용하여 새로운 백준 문제들을 추천해주는 시스템이다.

C. 문제 풀이 시스템

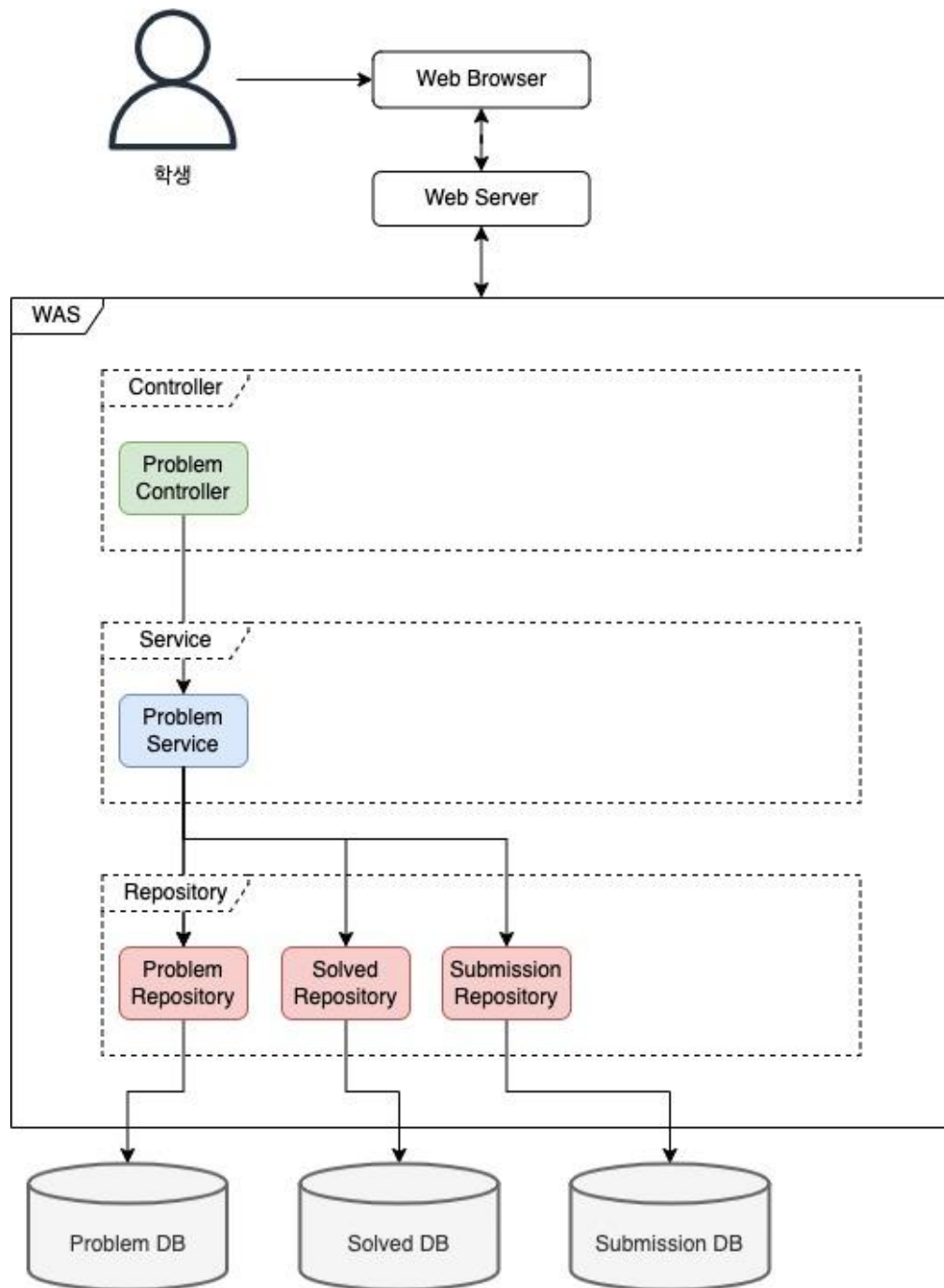


Diagram 4 System Architecture: 문제 풀이 시스템

문제 풀이 시스템은 “문제에 대한 학생의 코드 제출” 요구사항을 처리하는 시스템이다. 코드 제출 이후 제출 코드는 **Submission DB**에 저장되고, 학생이 푼 문제는 **Solved DB**에 추가된다. 또한, 분석은 제출한 코드를 바탕으로 코드 분석 시스템을 통해 이루어진다.

D. 코드 분석 시스템

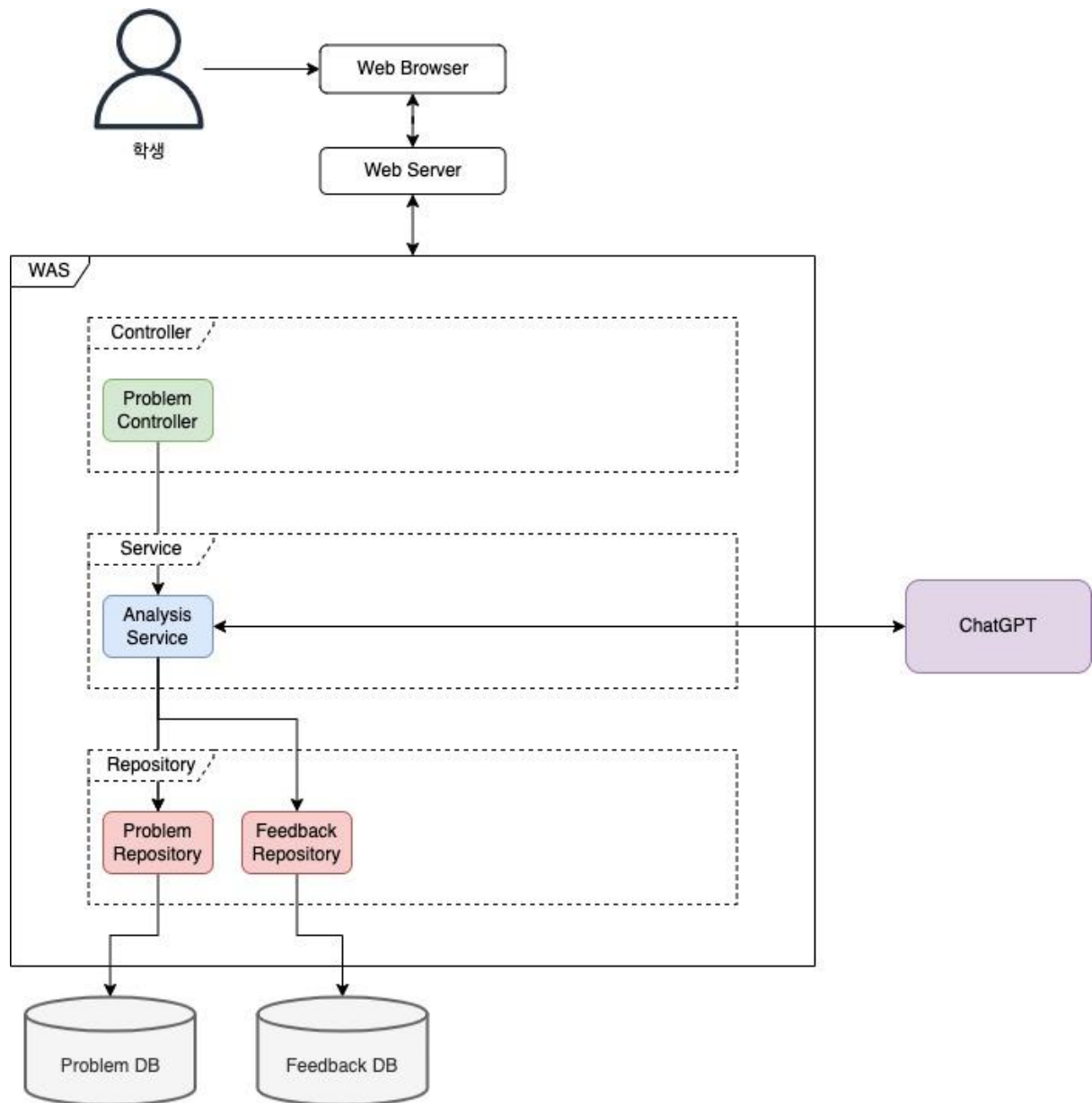


Diagram 5 System Architecture: 코드 분석 시스템

코드 분석 시스템은 제출 코드에 대한 분석을 처리하는 시스템이다. ChatGPT를 통해 제출 코드에 대한 **efficiency, readability, correctness, scalability, modularity, security**에 대한 점수 및 코드에 대한 전체적인 피드백을 얻는다.

E. 학생 관리 시스템

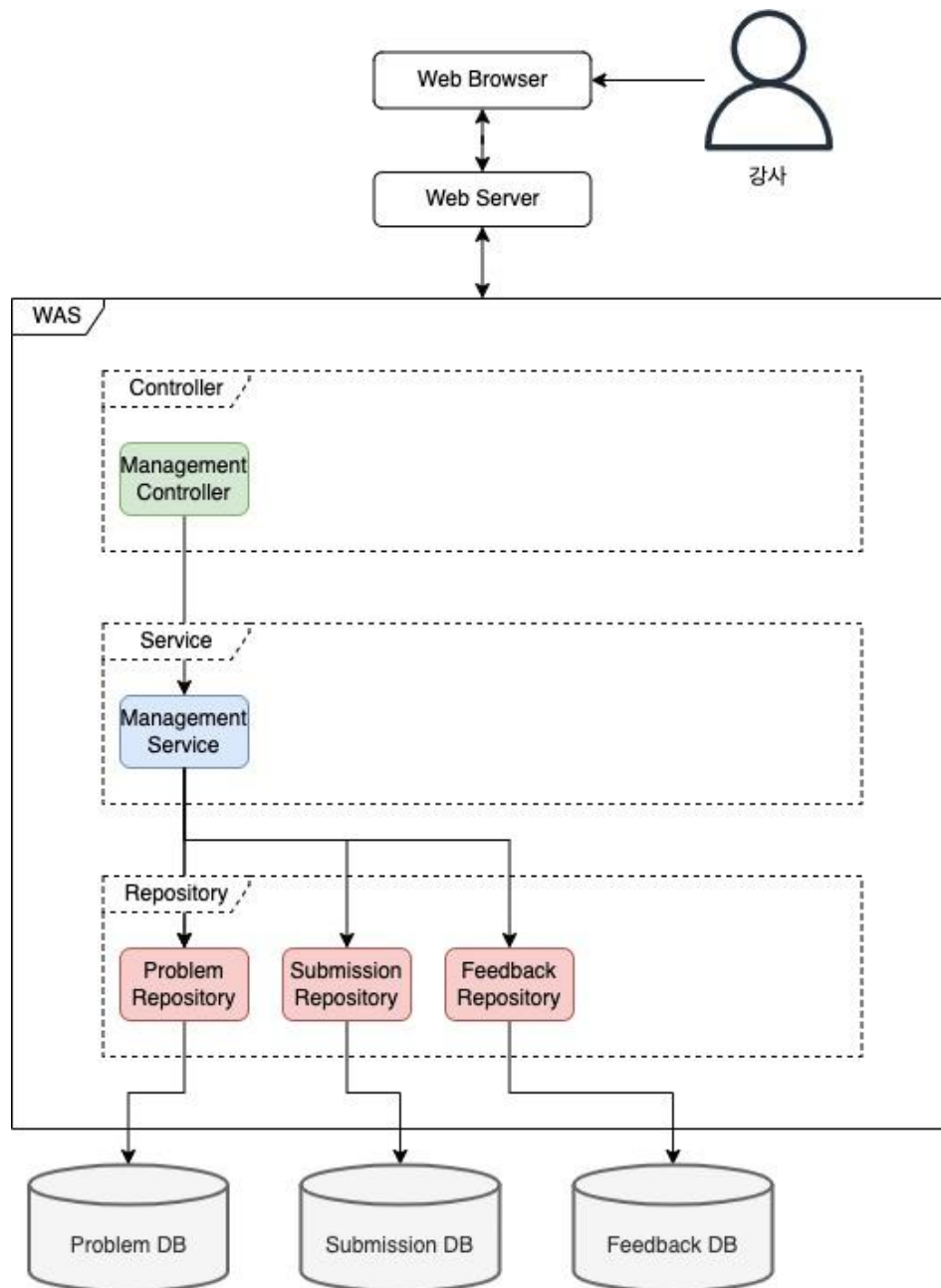


Diagram 6 System Architecture: 학생 관리 시스템

학생 관리 시스템은 강사가 학생들을 관리하는 시스템이다. 이를 통해 강사는 학생들이 제출한 코드를 확인하고, ChatGPT의 피드백을 확인할 수 있으며, 이를 수정하여 학생에게 피드백을 전달할 수 있다.

6 System Requirement Specification

6.1 Objective

기능적 요구사항 및 비기능적 요구사항에 대해 상세히 서술한다. 필요한 경우, 비기능적 요구사항에 대한 세부적 정보를 추가할 수 있다.

6.2 Functional Requirement

A. 회원가입

항목	설명
기능	회원가입
설명	사용자 정보를 입력받은 다음, 데이터베이스에 저장한다.
입력	이메일, 비밀번호, 이름, 직업(학생/강사), 직업이, 학생일 경우 담당 강사, 백준 아이디
출력	이메일 중복 체크 후 중복 여부에 따른 결과 메시지 출력 외부 알고리즘 서비스 계정 정보 존재 여부 확인 후 결과 메시지 출력 입력 양식을 모두 작성했는지 확인 후, 조건 충족 여부 출력 회원가입 결과 출력
처리	회원 가입을 완료한 뒤 데이터베이스에 회원 정보를 저장
조건	이메일: '아이디 문자열 @ 도메인 문자열' 형식이며, 중복되어서는 안된다. 비밀번호는 8~16자의 문자열 형식이다.

Table 3 Functional Requirement: 회원가입

B. 로그인

항목	설명
기능	로그인
설명	사용자가 서비스를 이용하기 위해 이메일과 비밀번호를 입력하면 학생 혹은 강사 권한을 얻을 수 있다. 학생의 경우 추천 문제 받기, 코드 피드백 받기, 푼 문제에 대한 요약 등의 권한을 받는다. 강사 권한의 경우 학생이 제출한 문제 피드백 하기, ChatGPT를 통한 학생 코드 분석 기능의 권한을 얻을 수 있다. 로그인 없이는 서비스를 이용할 수 없다.
입력	사용자의 이메일과 비밀번호를 입력한다.
출력	<p>데이터 베이스에 해당 이메일과 비밀번호가 일치한다면 로그인에 성공한다. 성공하면 메인 화면으로 redirect한다.</p> <p>이메일이 틀리다면 해당 이메일이 존재하지 않는다는 메시지를 출력하고 사용자 정보 입력창은 초기화된다.</p> <p>이메일이 데이터베이스에 있지만 비밀번호가 일치하지 않는다면 잘 못된 비밀번호라는 메시지를 출력하고 사용자 정보 입력창을 초기화한다.</p>
처리	<p>이메일 형식과 비밀번호 형식을 지켰는지 확인하고, 잘못된 입력일 경우 경고 문구를 출력한다.</p> <p>형식을 지켰을 경우, 로그인을 시도한다.</p> <p>데이터베이스에 저장된 회원 정보와 대조해 로그인에 성공 여부를 확인한다.</p>
조건	<p>이메일은 '아이디 문자열 @ 도메인 문자열' 형식이다.</p> <p>비밀번호는 4~12자의 문자열 형식이다.</p> <p>여러번 로그인 시도할 경우, 최초 한 번의 시도만 서버에 요청한다.</p>

Table 4 Functional Requirement: 로그인

C. 문제 추천

항목	설명
----	----

기능	문제 추천
설명	지금까지 푼 문제들을 바탕으로, 혹은 기본 추천 문제들을 학생에게 추천
입력	지금까지 풀었던 문제 숫자
출력	ChatGPT를 통한 다음 추천 문제
처리	ChatGPT api를 통해 지금까지 푼 문제를 바탕으로 다음 문제를 추천
조건	1. 학생이 로그인 되어 있어야 한다. 2. 문제가 없을 경우 기본 문제를 추천한다. 3. 학생에게 보이는 추천 문제는 총 3개가 유지 되도록 한다.

Table 5 Functional Requirement: 문제 풀기

D. 코드 작성 및 제출

항목	설명
기능	제출 된 코드에 대해 ChatGPT가 개선점 추천
설명	ChatGPT는 학생이 입력한 코드를 바탕으로 추천피드백을 한다.
입력	코드와 그에 대한 ChatGPT의 피드백
출력	강사의 코멘트가 추가된 학생의 코드
처리	ChatGPT의 피드백 된 코드를 바탕으로 강사는 코멘트를 작성한다. 작성된 코멘트는 학생에게 전송된다.
조건	1. 제출한 코드가 ChatGPT 에게 피드백을 받은 상태여야 한다. 2. 인터넷이 연결된 상태여야 한다.

Table 6 Functional Requirement: 코드 작성 및 제출

E. 코드 피드백 확인

항목	설명
기능	코드 피드백 확인
설명	ChatGPT가 생성한 코드에 더해 강사의 코멘트가 추가된 피드백을 학생이 확인할 수 있다.
입력	코드의 unique key
출력	강사가 작성한 코드에 대한 피드백
처리	강사가 제출한 피드백을 DB에 저장해두고 학생의 피드백 확인 쿼리가 발생했을 때 피드백을 보낸다.
조건	1. 학생이 로그인 되어 있어야 한다. 2. 학생이 제출한 문제에 대한 피드백이 완료 되어야 한다.

Table 7 Functional Requirement: 코드 피드백 확인

F. 푼 문제 확인

항목	설명
기능	푼 문제 확인
설명	학생이 지금까지 풀었던 문제에 대한 코멘트를 확인할 수 있다. 앞으로 풀어야 할 문제들이 어떤 것인지 확인할 수 있다. 단순한 문제 나열이 아닌 코드에 대한 정략적인 수치들이 포함된 내용들을 나타낸다.
입력	마이페이지 클릭
출력	마이페이지 화면
처리	마이페이지 클릭하면 학생이 풀었던 문제들을 나열하고 그 문제들을 보여준다.
조건	1. 학생이 로그인 되어있어야 한다. 2. 문제 풀이 기록이 있어야 한다.

Table 8 Functional Requirement: 푼 문제 확인

G. 학생들이 작성한 코드 확인

항목	설명
기능	학생들이 작성한 코드 확인
설명	강사가 학생들이 제출한 코드를 확인할 수 있다
입력	학생들의 코드 클릭
출력	학생들의 코드
처리	강사가 학생들의 코드를 클릭하면 코드가 펼쳐지고 확인할 수 있다.
조건	<ol style="list-style-type: none"> 1. 강사가 로그인 되어 있어야 한다. 2. 학생이 제출한 코드가 존재해야 한다.

Table 9 Functional Requirement: 학생들이 작성한 코드 확인

H. ChatGPT를 사용한 학생 코드 분석 및 시각화

항목	설명
기능	ChatGPT를 사용한 학생 코드 분석 및 시각화
설명	학생들의 코드에 대한 ChatGPT의 정량적, 정성적인 분석을 시각화 해서 보여준다.
입력	학생들의 코드
출력	ChatGPT의 분석이 추가된 학생들의 코드
처리	ChatGPT로부터 받은 피드백을 파싱하여 정리된 데이터로 변환한 후 이를 차트 혹은 웹페이지 요소로 변환하여 보여준다.
조건	<ol style="list-style-type: none"> 1. 학생의 문제가 제출 되어있어야한다. 2. 인터넷과 ChatGPT에 연결이 되어있어야 한다.

Table 10 Functional Requirement: ChatGPT를 사용한 학생 코드 분석 및 시각화

I. 피드백 작성과 제출

항목	설명
기능	피드백 작성과 제출
설명	강사가 ChatGPT의 분석이 추가된 학생의 코드를 받고 피드백을 작성하고 제출한다.
입력	시각화된 코드에 대한 피드백
출력	강사의 피드백이 포함된 학생의 코드

처리	강사가 시각화 자료를 보고 학생 코드를 분석한 후 개선해야할 점을 피드백으로 작성한다.
조건	<ol style="list-style-type: none"> 1. 강사가 로그인 되어있어야 한다. 2. 학생의 코드에 대한 ChatGPT의 분석이 되어있어야 한다.

Table 11 Functional Requirement: 피드백 작성과 제출

6.3 Nonfunctional Requirement

A. Product Requirement

A.1. Usability Requirement (UI)

단순하고 한 눈에 볼 수 있는 UI를 갖추어야 한다. 제출된 코드에 대한 피드백과 코멘트들은 코드와 배경색과 다른 색으로 구분되어야한다. 피드백 종류 별로 (eg 시간복잡도나 공간복잡도 관련 피드백으로는 빨간색, 가독성 관련 피드백은 파란색) 다른 색으로 한다.

A.2 Efficiency Requirement

제출한 문제가 강사에게 제출 되는 것이 3sec를 넘어서는 안된다. 강사가 받은 코드들에 대한 ChatGPT의 피드백이 5sec를 넘어서는 안된다. ChatGPT의 피드백은 최대 1000자로 제한한다.

A.3 Dependability Requirement

회원가입할 때 입력한 사용자의 데이터가 데이터 베이스에 저장되는 과정에서 변질되지 않아야 한다. 올바른 사용자 정보를 입력했을 경우 로그인을 성공할 수 있어야 한다. 문제 풀이 후 제출된 코드는 손상되지 않아야 하고 코드에 대한 ChatGPT 피드백이 정확히 강사에게 추천 되어야 한다.

A.4 Security Requirement

사용자의 개인정보는 암호화되어 보관해야 한다. 사용자의 이름과 이메일은 강사에게는

공개 되지만, 비밀 번호는 공개되지 않는다. 비밀번호는 입력시 *로 입력되어 노출되지 않게 한다.

B. Organization Requirement

B.1 Environmental Requirement

해당 서비스는 온라인 웹서비스이며, **PC**로 이용할 수 있다. 사용자는 인터넷이 연결된 상태에서 사용하여야 한다. 크롬, 익스플로러에서 사용할 수 있다. 사용자의 사용 환경에 관계 없이 이용에 불편이 없도록 하는 것이 목표이다.

B.2 Operational Requirement

HTML, CSS, JavaScript, React를 사용해 프론트엔드를 개발한다.

ChatGPT API, Spring, Django를 사용해 백엔드를 개발한다.

B.3 Development Requirement

사용자의 문제 풀이 데이터와 사용자 정보는 데이터베이스에 저장된다. 여러 페이지를 이동하는 동안 정보가 기억될 수 있다. 서비스 특성상 **PC** 사용자를 우선순위를 두고 개발하나, 모바일 환경에서도 이용할 수 있도록 한다. **Django** 프레임워크를 통해 호환성을 높인다.

C. External Requirement

C.1 Regulatory Requirement / Legislative Requirement

- (저작권을 확인해야 한다. 회원 정보를 외부에 제공할 경우 정보 제공 동의를 얻어야 한다.)
- 사용자 정보의 공개 범위는 사용자 본인과 강사까지이며, 비밀번호의 경우 사용자 본인만 알 수 있다.

학생 정보를 외부에 제공 하려면 학생 본인에게 정보 제공 동의를 받아야 한다.

사용자가 회원탈퇴를 원하는 경우, 강사에게 문의하여 삭제될 수 있으나 다시 회원가입을 원할 경우 과거의 기록은 복구할 수 없다.

C.2 Ethical Requirement

본 서비스는 이용 연령 제한이 없다.

C.3 Safety/Security Requirement

- 개인정보보호법에 따라 고객 개인정보를 안전하게 보관하고 보호하여야 한다.

6.4 Scenario

A. 회원가입

A.1 Initial Assumption

사용자는 적어도 한 개 이상의 이메일 계정을 가지고 있어야 한다.

A.2 Normal Flow of Events

사용자는 로그인 되지 않은 상태에서 메인 페이지에 접근할 경우, 로그인을 요청받는다.

로그인 화면에서 회원 가입 버튼을 클릭해 회원 가입 화면으로 이동한다.

회원 가입 화면에서 이메일, 비밀번호, 이름, 직업(학생/강사)를 입력한다.

회원 가입에 필요한 정보들을 입력한 다음 회원가입을 요청한다.

A.3 What can go wrong

중복된 이메일이 있을 경우, 해당 이메일로는 가입하지 못한다.

이메일 형식을 지키지 못한 경우, 올바른 형식을 입력하도록 유도한다.

비밀번호 형식을 지키지 못한 경우, 올바른 형식을 입력하도록 유도한다.

백준, **hackerrank** 외의 외부 알고리즘 서비스를 입력하려 할 경우, 다시 입력하도록 한다.

외부 알고리즘 서비스에 등록되지 않은 계정을 입력 하려 할 경우, 다시 입력하도록 한다.

A.4 System State on Completion

모든 정보가 올바르게 입력되고 서버에서 검증작업을 완료한 후에 사용자 데이터베이스에 저장된다.

B. 로그인

B..1 Initial Assumption

사용자는 데이터 베이스에 존재된 회원 정보가 한 개 이상 존재하여야 한다.

B.2 Normal Flow of Events

사용자는 로그인 되지 않은 상태에서 메인 페이지에 접근할 경우, 로그인을 요청받는다.

로그인 화면에서 이메일, 비밀번호를 입력한다.

입력 후 로그인 버튼을 클릭해 로그인을 요청한다.

B.3 What can go wrong

이메일 형식을 지키지 못한 경우, 올바른 형식을 입력하도록 유도한다.

비밀번호 형식을 지키지 못한 경우, 올바른 형식을 입력하도록 유도한다.

이메일이 존재하지 않을 경우, 존재하지 않는 이메일이라는 메시지를 출력한다.

이메일은 존재하지만 비밀번호가 틀릴 경우, 비밀번호가 틀렸다는 메시지를 출력한다.

B.4 System State on Completion

모든 정보가 올바르게 입력되고 요청을 보내면, 서버에서 사용자 데이터베이스의 정보와 사용자가 요청한 계정 정보를 대조해, 로그인 상태로 저장한다. 사용자에게 **access token**을 발급해 서비스를 이용할 수 있도록 한다.

C. 문제 추천

C.1 Initial Assumption

학생은 로그인을 한 상태여야 한다. 학생이 풀었던 문제의 기록이 있다.

C.2 Normal Flow of Events

학생이 로그인 된 상태에서 풀었던 문제들의 번호를 입력한다. ChatGPT를 통해 다음 문제를 추천한다.

C.3 What can go wrong

추천해준 문제가 이미 있는 초기 문제와 겹칠 경우, 다른 문제를 추천해준다.

C.4 System State on Completion

추천된 문제가 문제 목록에 추가된다.

학생이 제출한 문제에 대한 ChatGPT의 피드백이 강사에게 전달된다.

D. 코드 작성 및 제출

D.1 Initial Assumption

학생이 백준을 통해 푼 문제가 존재한다.

D.2 Normal Flow of Events

학생이 백준에서 푼 문제들을 제출 칸에 입력하고 제출 버튼을 누른다.

D.3 What can go wrong

컴파일이 안되거나 실제로 실행이 안되는 등의 문제를 제출할 경우 다시 백준을 통해 검증받는다.

D.4 System State on Completion

학생이 제출한 코드는 강사에게 전달된다.

E. 코드 피드백 확인

E.1 Initial Assumption

학생이 **Codemy**를 통해 코드를 제출한다.

E.2 Normal Flow of Events

학생이 **Codemy**에 본인의 문제에 대한 풀이 코드를 제출한다. 해당 코드는 **ChatGPT API**를 통해 분석되어지고 강사의 코멘트 등이 추가되어 있다. 학생이 자신이 풀었던 문제 목록을 클릭하면 해당 피드백들을 확인할 수 있다.

E.3 What can go wrong

학생이 **Codemy**에 입력한 코드들이 없을 경우 입력된 코드가 없다는 알림이 나타난다.

E.4 System State on Completion

학생이 제출한 코드의 피드백을 확인하고 원래 페이지로 돌아온다.

.

F. 푼 문제 확인

F.1 Initial Assumption

학생이 문제에 대한 코드를 제출한다.

F.2 Normal Flow of Events

학생은 문제에 대한 코드를 제출한다.

이후 풀었던 문제들을 확인할 수 있는 페이지에 접속하면 풀었던 문제들을 확인할 수 있다.

F.3 What can go wrong

학생이 풀었던 문제가 없다면 풀었던 문제가 존재하지 않는다.

F.4 System State on Completion

학생이 풀었던 문제들의 리스트를 확인할 수 있다.

G. 학생들이 작성한 코드 확인

G.1 Initial Assumption

학생들이 풀었던 문제들에 대한 코드를 제출한다.

G.2 Normal Flow of Events

학생들이 풀었던 문제들을 제출한다. 강사는 학생들의 목록중에 문제를 클릭하고 해당 코드를 확인할 수 있다.

G.3 What can go wrong

학생들이 제출한 코드가 없을 경우 제출물이 없다는 알림이 나타난다.

G.4 System State on Completion

학생들의 제출물을 확인하고 원래 페이지로 돌아온다.

H. ChatGPT를 활용한 학생 코드 분석 및 시각화

H.1 Initial Assumption

학생이 코드를 제출한다.

H.2 Normal Flow of Events

학생이 코드를 제출한다. ChatGPT API를 통해 해당 코드에 대한 정량적인, 정성적인 분석이 이루어진다. 이루어진 분석을 바탕으로 시각적인 결과를 보여준다.

H.3 What can go wrong

분석된 코드의 문제가 있다면, 강사 피드백 단계에서 수정된다.

H.4 System State on Completion

분석이 끝난 코드가 강사에게 보내진다.

I. 피드백 작성과 제출

I.1 Initial Assumption

학생이 제출한 코드가 ChatGPT의 분석을 거쳐 강사에게 전달 되었다.

I.2 Normal Flow of Events

학생이 코드를 제출한다. ChatGPT의 분석을 거쳐 강사에게 전달된다. 강사는 ChatGPT의 분석을 기반으로 피드백을 추가한다.

I.3 What can go wrong

ChatGPT의 분석을 그대로 입력할 경우 피드백에 차이가 없다는 경고 메시지를 출력한다..

I.4 System State on Completion

학생의 풀었던 문제 목록에 강사의 피드백이 추가된 코드가 업데이트 된다.

7 System Model

7.1 Objective

시스템 구성 요소, 시스템 및 시스템 환경 간의 관계를 설명한다.

7.2 Context Models

A. Context Model

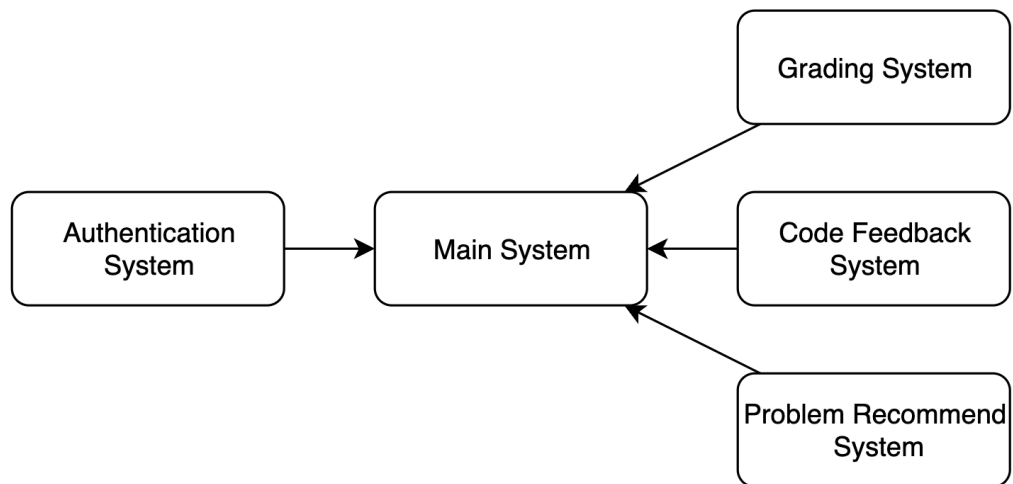


Diagram 7 Context Model

B. Process Model

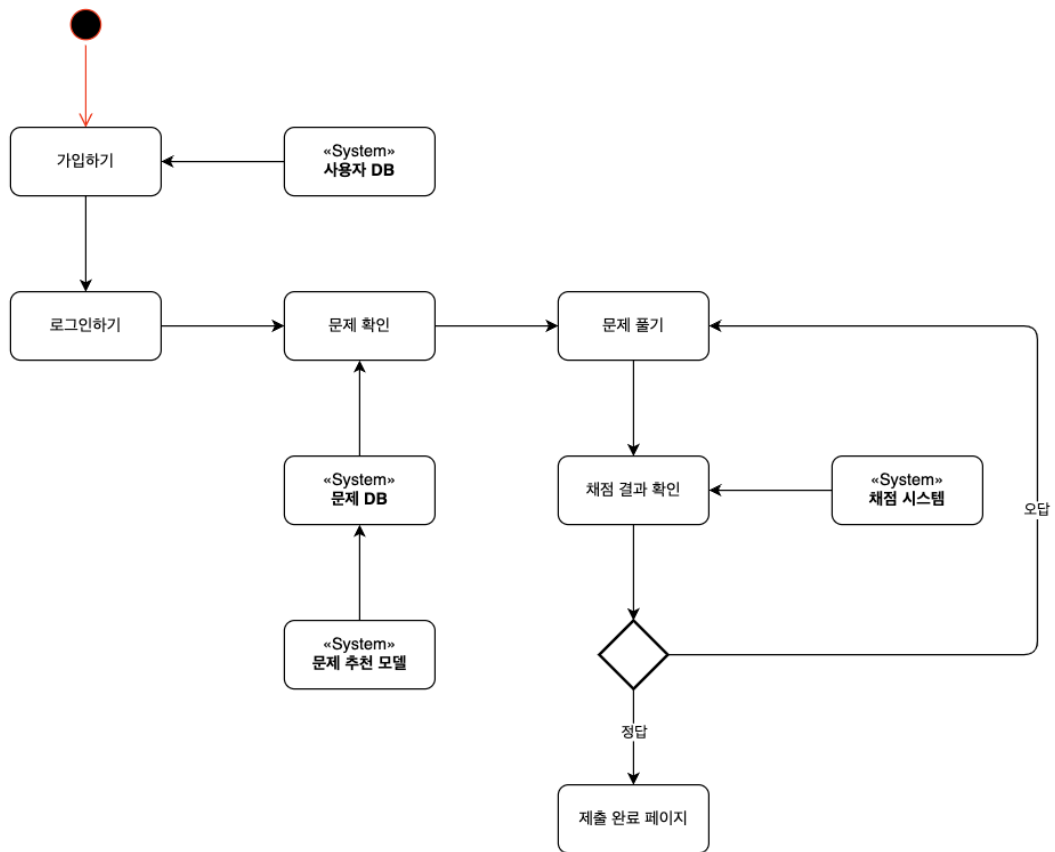


Diagram 8 Process Model - 문제 풀기

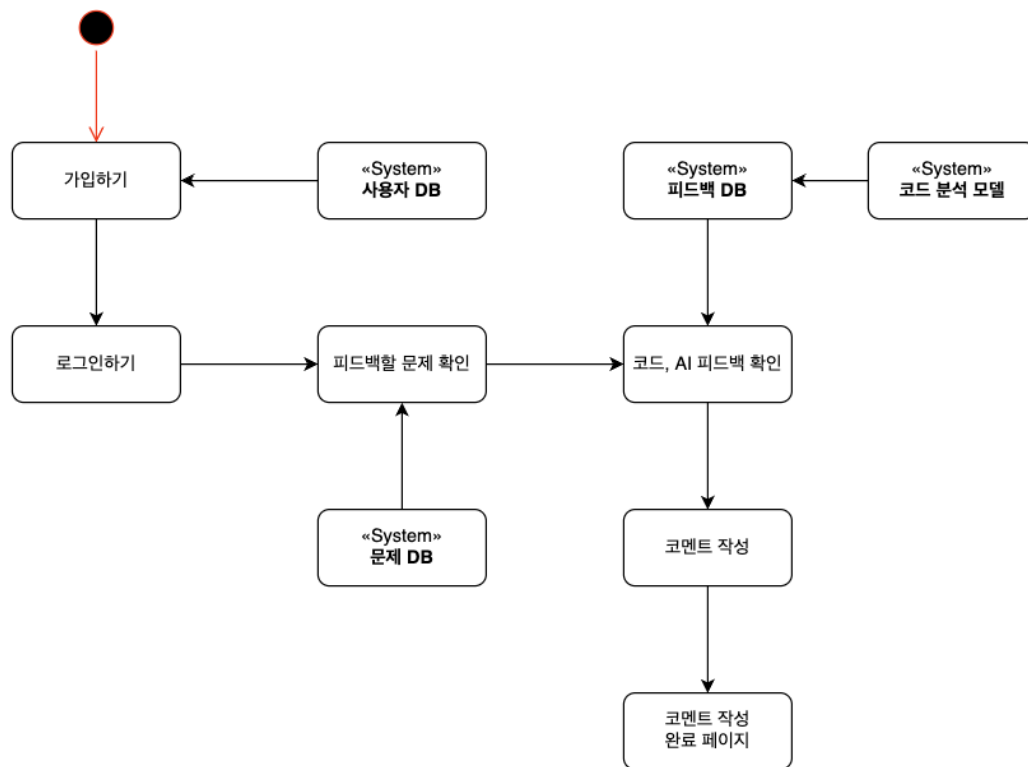


Diagram 9 Process Model - 코멘트 작성

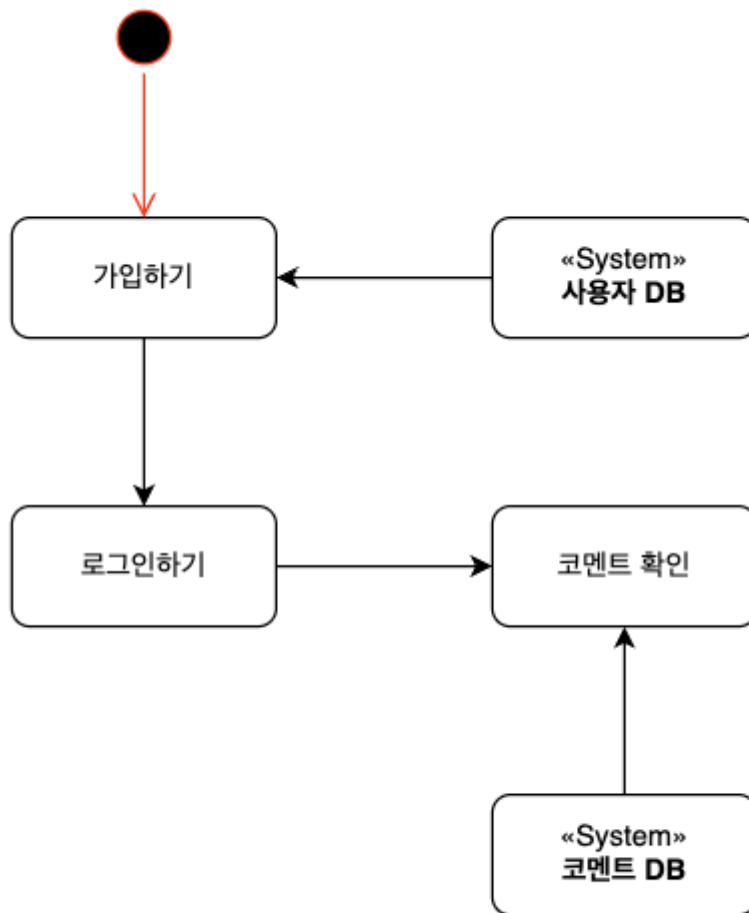


Diagram 10 Process Model - 코멘트 확인

7.3 Interaction Models

A. Use Case Model

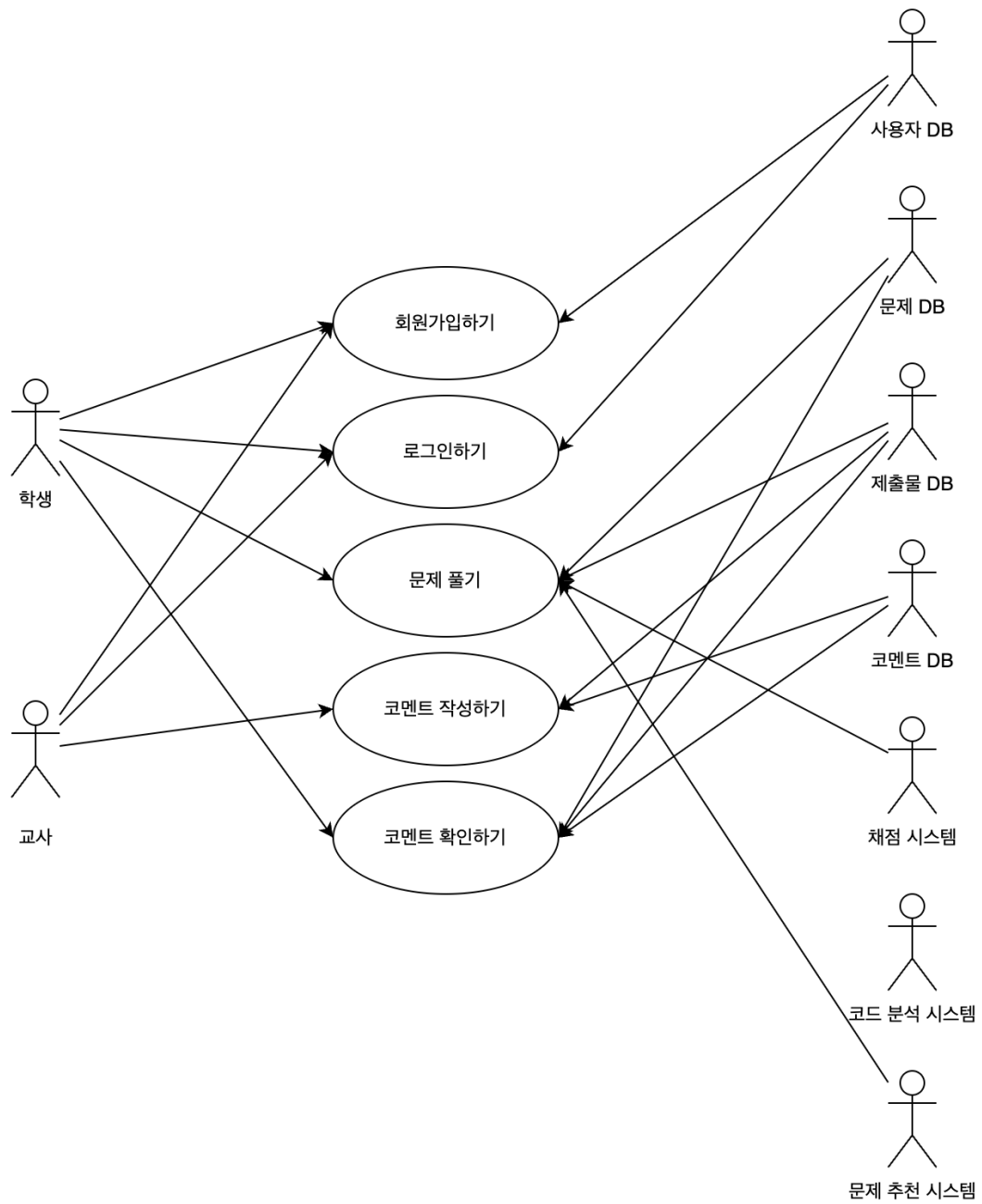


Diagram 11 Use Case Model

B. Tabular Description

B.1 회원가입하기

USE CASE	회원가입하기
ACTOR	학생, 교사, 사용자 DB
DESCRIPTION	회원 가입을 위해 사용자 정보를 입력 받고, 데이터베이스에 사용자 정보를 저장한다.
STIMULUS	사용자가 사용자 정보를 입력하고 회원가입 버튼을 누른다.
RESPONSE	사용자의 회원가입 정보를 검증한 뒤 DB에 사용자 정보를 등록한다.
COMMENTS	

Table 12 Tabular Description: 회원가입하기

B.2 로그인하기

USE CASE	로그인하기
ACTOR	학생, 교사, 사용자 DB
DESCRIPTION	사용자가 서비스 이용을 위해 이메일과 비밀번호를 입력하고 사용 권한을 얻는다.
STIMULUS	사용자가 이메일과 비밀번호를 입력하고 로그인 버튼을 누른다.
RESPONSE	사용자의 로그인 정보와 DB에 저장된 사용자 정보를 대조하여 로그인 결과를 반환한다.
COMMENTS	

Table 13 Tabular Description: 로그인하기

B.3 문제 풀기

USE CASE	문제 풀기
ACTOR	학생, 문제 DB, 제출물 DB, 문제 추천 시스템, 채점 시스템
DESCRIPTION	학생이 추천 문제 중 하나를 골라서 풀면 코드를 채점한다. 만약

	추천 문제가 없다면 추천 시스템을 통해서 새로운 문제를 추가한다. 학생의 코드가 정답인 경우, 학생의 코드가 제출물 DB에 저장된다. 오답인 경우 다시 풀도록 요청한다.
STIMULUS	학생이 추천 문제 중 하나를 골라서 코드를 작성해서 제출한다.
RESPONSE	학생이 제출한 코드를 채점하여 정답인 경우 정답 메시지와 함께 제출물을 저장한다. 오답인 경우 다시 풀도록 요청한다.
COMMENTS	

Table 14 Tabular Description: 문제 풀기

B.4 코멘트 작성하기

USE CASE	코멘트 작성하기
ACTOR	교사, 제출물 DB, 코멘트 DB
DESCRIPTION	교사가 학생의 제출물과 코드 분석 시스템의 피드백을 바탕으로 코멘트를 작성한다.
STIMULUS	교사는 학생의 제출물과 피드백을 바탕으로 코멘트를 작성한다.
RESPONSE	교사가 작성한 코멘트가 저장되고, 교사에게는 코멘트가 추가된 제출물에 대한 화면이 보여진다.
COMMENTS	

Table 15 Tabular Description: 코멘트 작성하기

B.5 코멘트 확인하기

USE CASE	코멘트 확인하기
ACTOR	학생, 문제 DB, 제출물 DB, 코멘트 DB
DESCRIPTION	학생이 자신의 제출물에 교사가 작성한 코멘트를 확인한다.
STIMULUS	학생이 자신의 제출물을 조회한다.
RESPONSE	교사가 학생에게 코멘트를 작성한 경우, 제출물에 교사의 코멘트가 함께 표시된다.
COMMENTS	

Table 16 Tabular Description: 코멘트 확인하기

C. Sequence Diagram

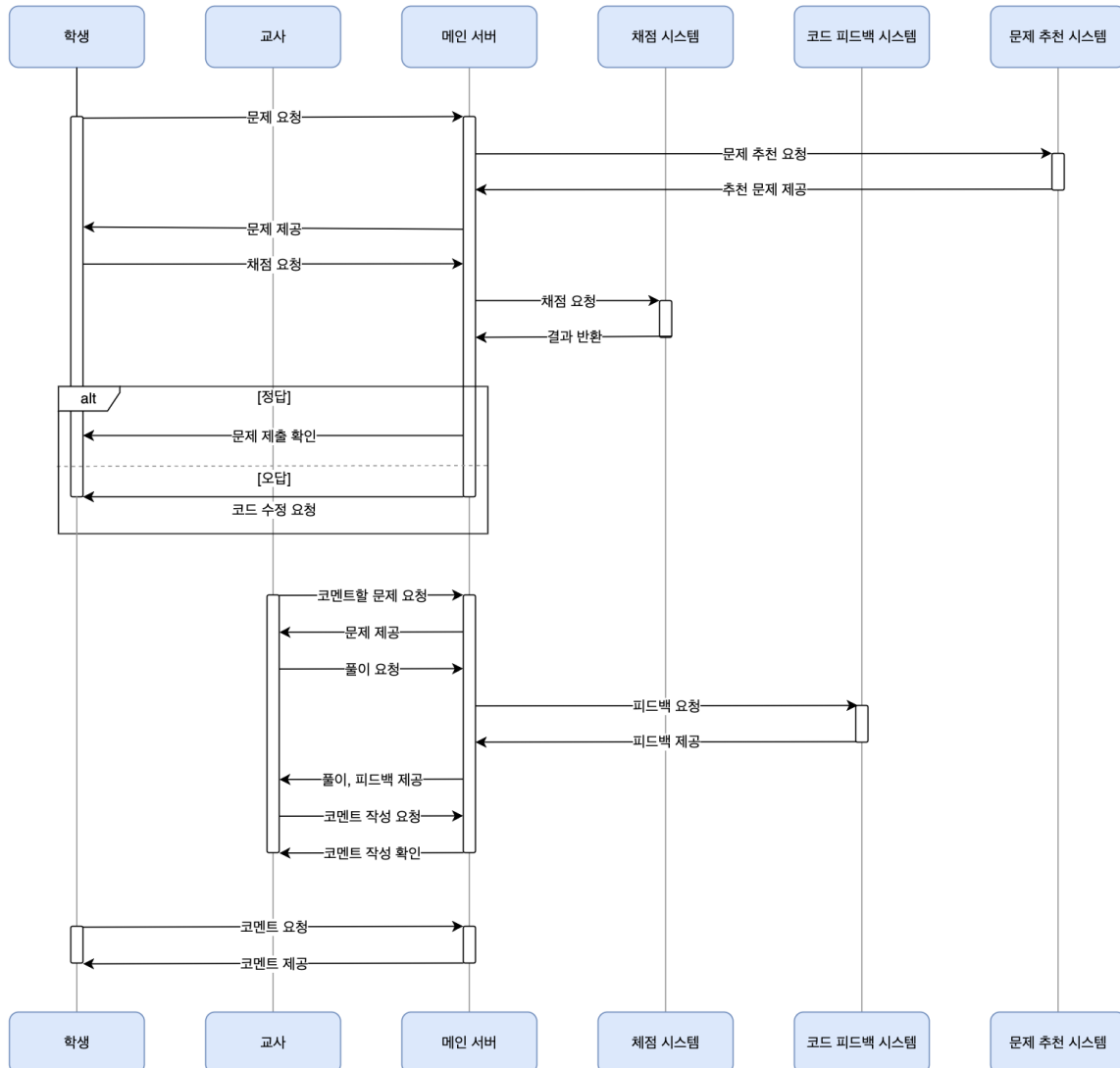


Diagram 12 Sequence Diagram: 전체

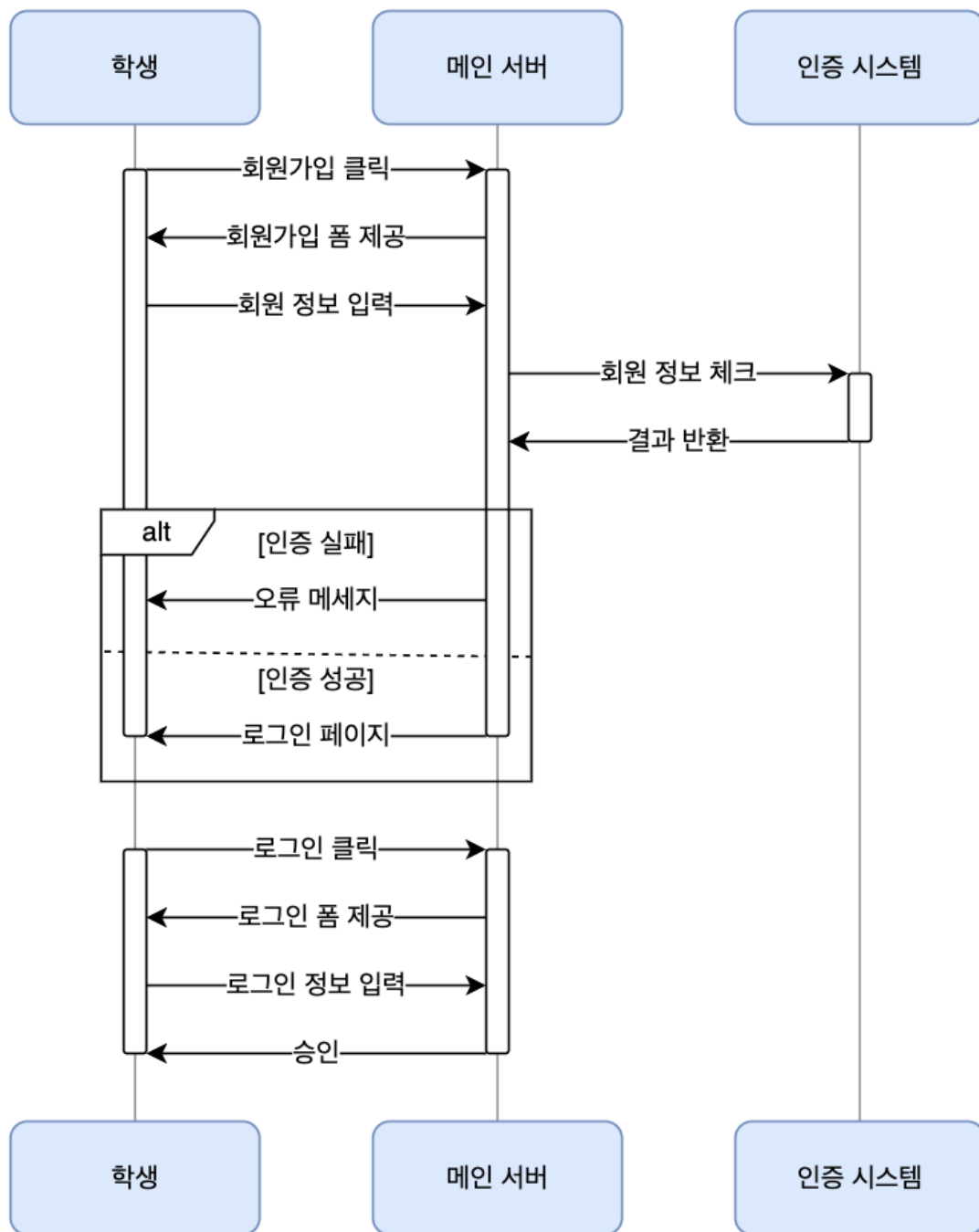


Diagram 13 Sequence Diagram: 회원가입

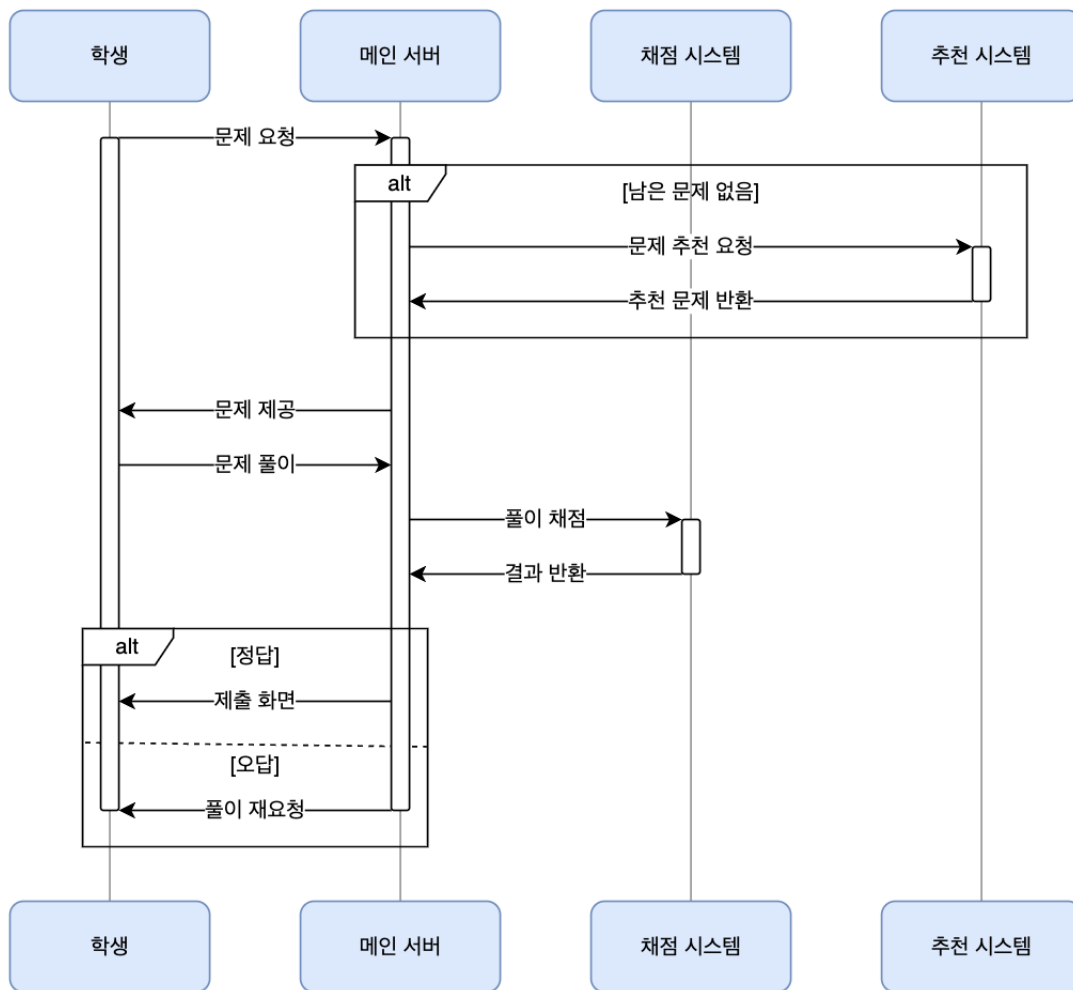


Diagram 14 Sequence Diagram: 문제 풀이

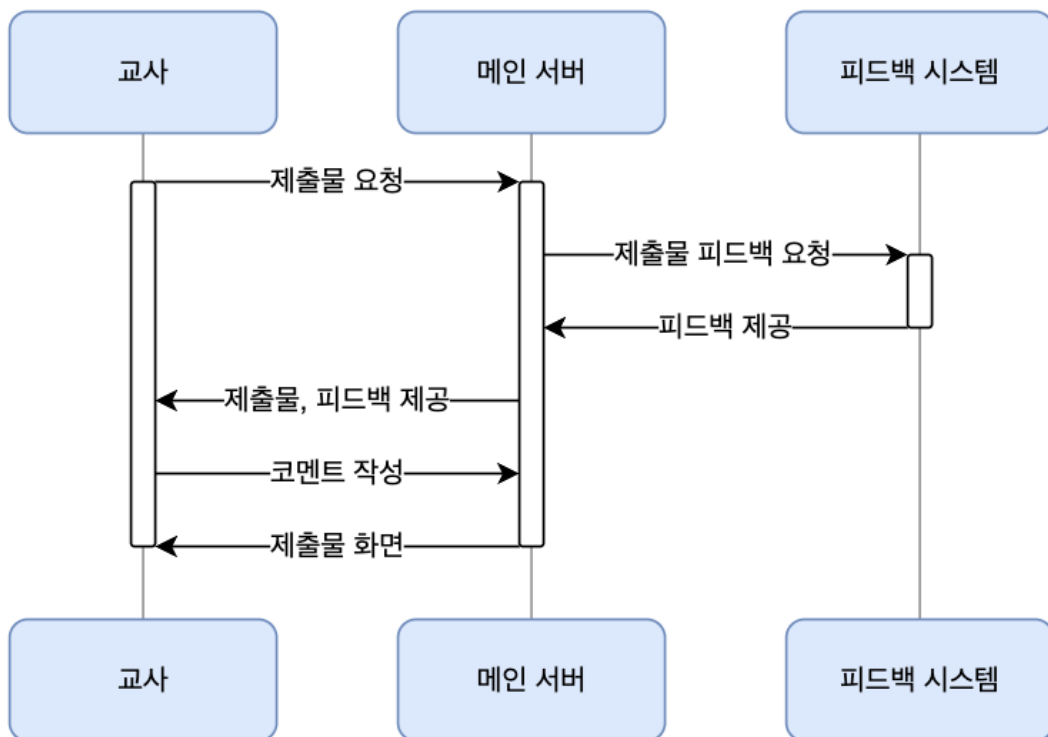


Diagram 15 Sequence Diagram: 피드백

7.4 Structural Models

A. Class Diagram

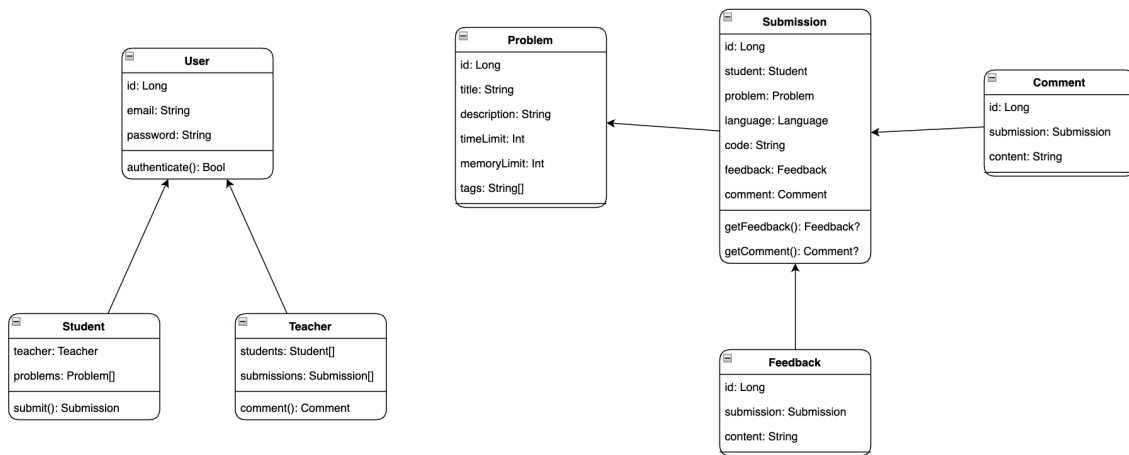


Diagram 16 Class Diagram

7.5 Behavioral Models

A. Event-driven Diagram

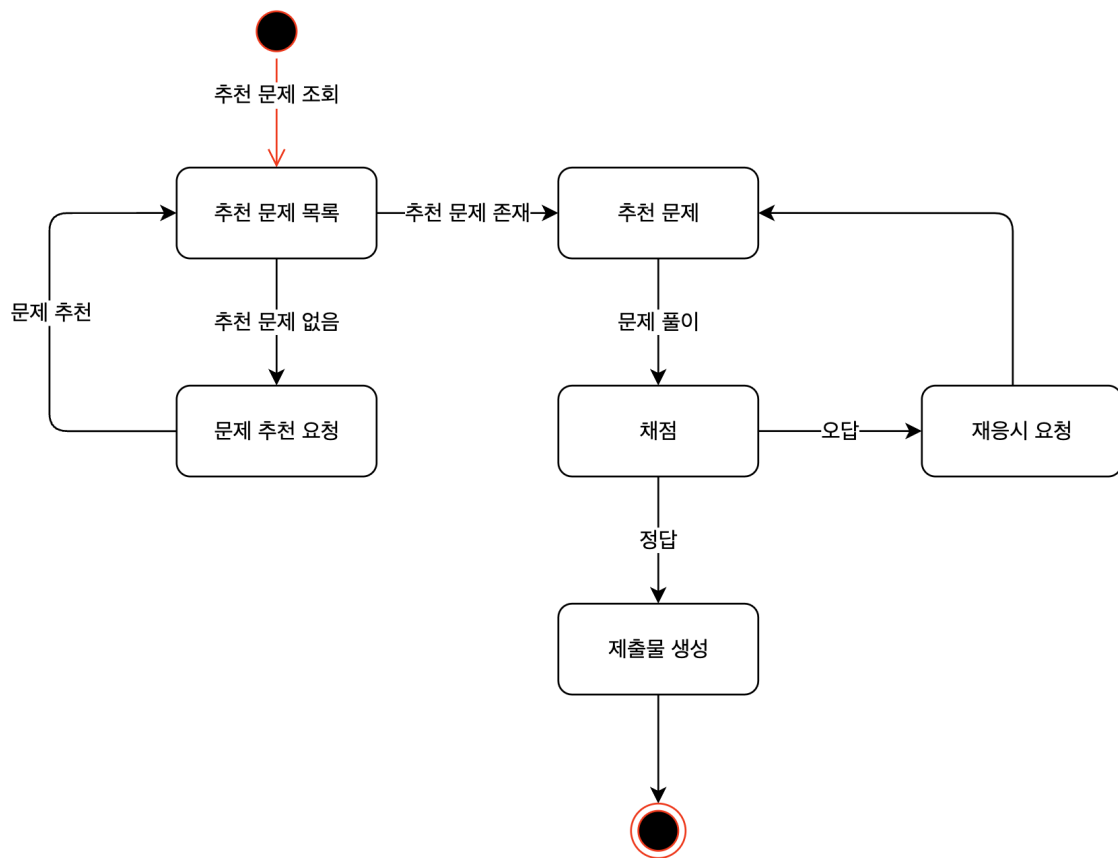


Diagram 17 Event Driven Diagram: 문제 풀이

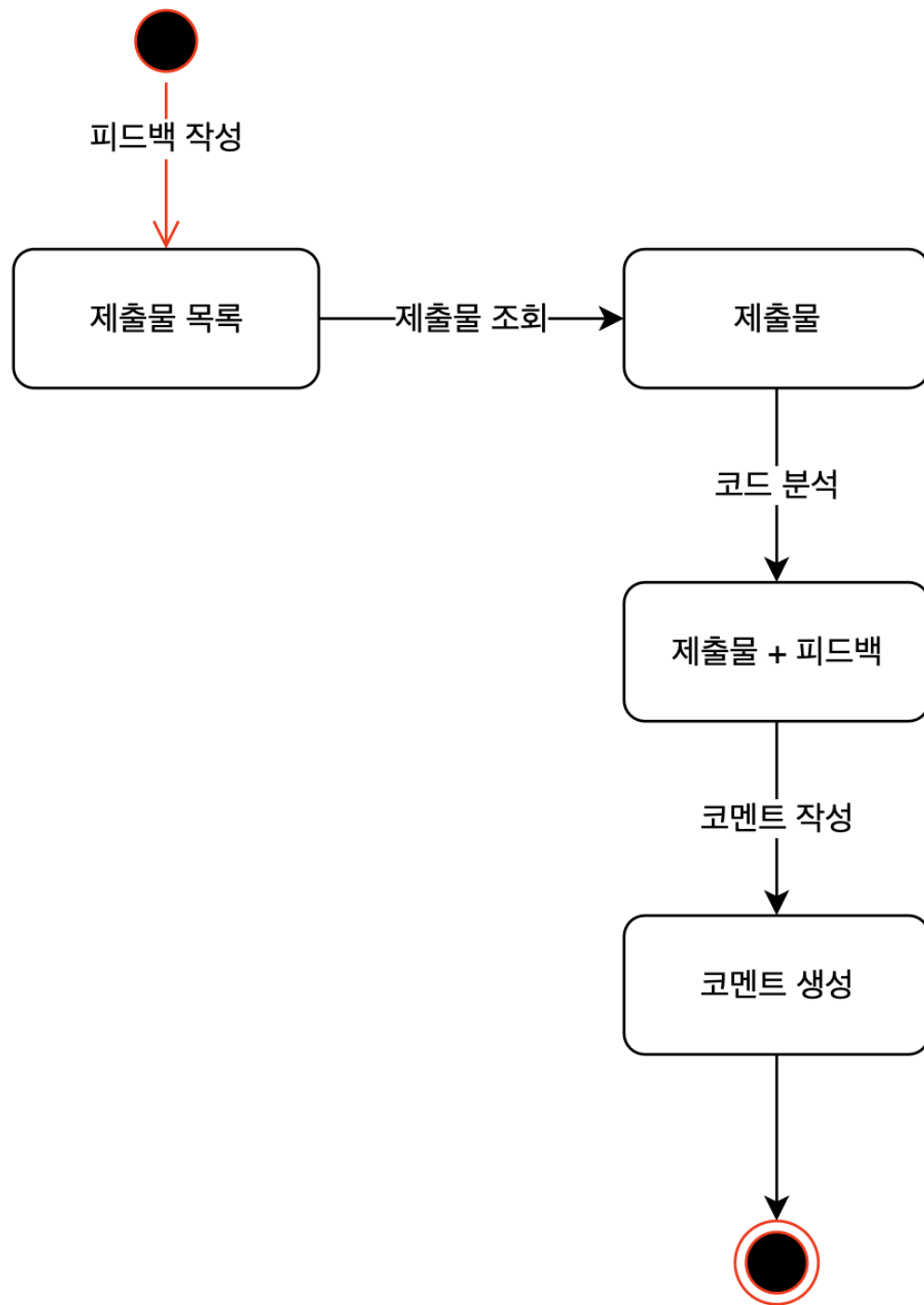


Diagram 18 Event Driven Diagram: 코멘트 작성

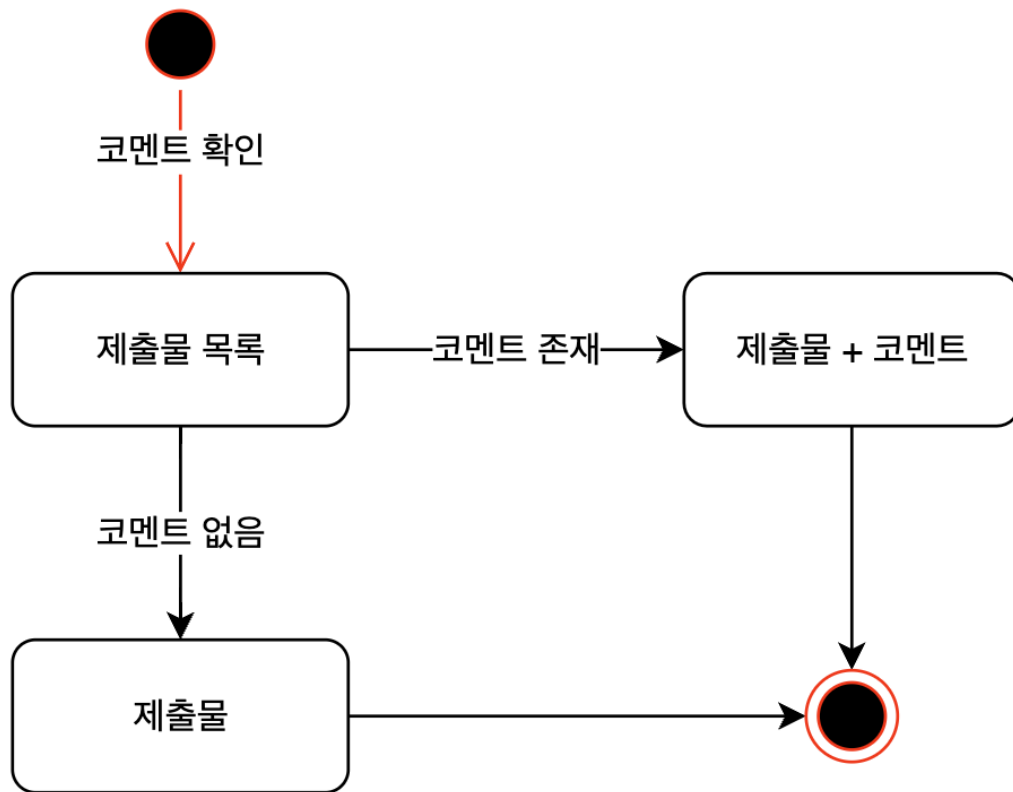


Diagram 19 Event Driven Diagram: 코멘트 확인

8 System Requirement Evolution

8.1 Objective

시스템이 기반으로 하는 가정과 하드웨어 진화, 사용자 요구사항의 변화 등으로 인한 예상 변화에 대해 설명한다. 시스템 디자이너는 이를 통해 시스템의 미래 변경 사항을 제한할 수 있는 디자인 결정을 피할 수 있다.

8.2 Assumption and Limitation

A. 서비스 완전 자동화 문제

Codemy는 학생이 제출한 코드에 대한 분석 및 피드백을 하는 과정에서, 1차적으로

ChatGPT가 분석을 수행한 다음, 분석 결과에 오류나 문제가 있는지 2차적으로 교강사가 판단하게 된다. 사전 테스트 과정에서 ChatGPT의 코드 분석 성능을 확인해본 결과, 간혹 중복된 문제를 추천해 주거나, 코드의 의도를 파악하지 못하는 경우가 발생하였으며, 학생들이 이로 인해 잘못된 지식을 학습하는 것을 방지하기 위하여, 2단계의 검증 과정을 거쳐서 최종 피드백을 산출하도록 시스템을 설계하였다.

하지만, 이로 인해서 학생은 자신의 코드의 문제점을 즉각적으로 피드백 받지 못하고, 교강사가 열람하기까지 기다려야 하는 불편함을 감수해야 한다. 코드에 대한 피드백의 정확도를 얻는 대신 자동화된 플랫폼 사용을 통한 편리함은 포기한 셈이다. ChatGPT의 분석 성능을 더욱 개선하거나, 미리 준비한 답안을 학습하는 등의 방법으로 서비스를 완전 자동화하여 사용자의 편리성을 증대시킬 수 있을 것이다.

B. 오픈소스 코딩 문제의 유한성

Codemy는 오픈소스 코딩 문제에 한하여 학생에게 문제를 추천해준다. 그러나 백준과 같은 검증된 웹사이트의 PS(Problem-Solving) 문제수는 유한하고, 기업에서 테스트에 사용하는 문제 목록은 대부분 외부로 유출되는 것을 엄격히 규제하기 때문에 사용하기 어렵다. 이러한 상황에서, 특정 유형의 문제를 반복적으로 추천해준다 보면 문제수가 부족할 수 있다.

또한, ChatGPT API가 추천해줄 수 있는 코딩 문제 자료는 모델이 학습한 데이터셋 상에 존재해야 하기 때문에, 교강사가 새로운 문제를 직접 만들어서 추천 문제 목록에 추가하는 것은 어려우며, 이에 해당하는 기능을 따로 구현해 주어야 한다.

8.3 Evolutions for Hardware and Software

Codemy의 웹페이지는 HTML, CSS, JavaScript를 기반으로 작성되었으며, 서비스 개발에 주어진 기간이 짧고, 다수의 팀원이 사용할 수 있는 언어를 사용하는 것이 효율적이기 때문에 추가적인 프레임워크는 사용하지 않았다. 하지만, 코딩 교육 플랫폼의 페이지 수가 증가할 수록 소프트웨어의 규모도 커질 것이며, 페이지 단위가 아닌 세부 파트로 웹페이지 구성 코드를 나누어 개발하는 것이 편리할 것이다. 또한 모바일 어플리케이션으로 교육 플랫폼을 이용하고자 하는 요구 사항이 생길 수 있기 때문에, 장기적으로 서비스를 유지

보수하기 위해서는 적절한 웹/앱 프레임워크를 이용해 코드를 재정리하는 과정이 필요할 수 있다.

8.4 Evolutions for Teacher

코딩 교육 플랫폼을 사용하는 학생들이 같은 학원이나 집단에 소속된 경우에, 학생들 간에 문제 정답을 공유하여 숙제를 완료하는 문제가 발생할 수도 있다. 따라서, 학생들 간에 중복된 정답이 존재하는지를 자동으로 검사해주는 표절도 검사 시스템을 서비스에 추가할 필요가 있다. 이 서비스를 통해서 교강사는 코드의 내용이 학생들 간에 겹치는지 여부를 신경쓰지 않고, 오로지 각 학생의 코드에 대한 ChatGPT 피드백이 적절히 이루어졌는지만 파악하면 된다.

8.5 Evolutions for Student

현재 코딩 교육 시장은 공신력 있는 SW 교육 자격증이 따로 존재하지 않고, 검증되지 않은 사설 자격증이 통용되고 있다는 문제점이 존재한다. 이러한 문제점을 해결하기 위해서 SW 교육과 관련한 ‘국가 공인 자격증’이 신설되어야 한다는 의견도 존재하지만, 당장은 코딩 교강사 인원의 교육 역량을 명확히 검증하기 어려운 것이 현실이다. 따라서 학생들이 실력있는 교강사로부터 교육을 받고 있는지에 대해 선 검증하는 절차를 플랫폼에 요구할 수도 있다. 이 경우, 본 교육 플랫폼에 교강사가 가입하는 과정에서 신뢰할 만한 기관에서 취득한 교육 자격증을 필수로 기재하도록 의무화 하는 등의 방법으로 서비스를 개선할 수 있다.

9 Appendix

9.1 Objectives

개발되고 있는 어플리케이션과 관련된 세부 정보를 제공한다.

9.2 Database Requirements

본 시스템에서 사용하는 데이터베이스의 요구사항을 서술한다. 각 요구사항에 *Attribute*는

이태릭체, **Entity type**은 굵은 글씨체, **Relationship type**은 굵은 이태릭체, Constraint는 밑줄로 표시한다.

본 시스템에서 사용하는 DB(데이터베이스)는 **Student DB, Teacher DB, Solved DB, Problem DB, Submission DB, Feedback DB**이다. 다음에서 설명하는 내용들은 해당되는 요구사항에 맞는 형태로 데이터베이스 테이블에 저장되어야 한다.

Student DB는 id를 필수적으로 가져야 하고, *email* 등을 추가적으로 가질 수 있다.

Teacher DB는 id를 필수적으로 가져야 하고, 추가적으로 *problem, user_id, username, code, result*를 attribute로 가질 수 있다.

Solved DB는 id를 필수적으로 가져야 하고, *title, description, input, output, test_case_id, difficulty*를 attribute로 가질 수 있다.

Problem DB는 id를 필수적으로 가져야 하고, *title, title, description, timeLimit, memoryLimit, tags*를 attribute로서 가질 수 있다.

Submission DB는 id를 필수적으로 가져야 하고, *problem_id, student_id, language, code, feedback_id, comment_id*를 attribute로서 가져야 한다.

Feedback DB는 id를 필수적으로 가져야 하고, *submission_id, content*를 attribute로서 가져야 한다.

Comment DB는 id를 필수적으로 가져야 하고, *submission_id, content*를 attribute로서 가져야 한다.

9.3 Database Detail

TODO

Student DB, Teacher DB는 Codemy에 사용자가 가입할 때 입력되며, **Problem DB, Submission DB**에 참조된다. **Submission DB**는 학생이 문제에 대한 코드를 제출하면 제출 결과를 저장한다. **Problem DB**는 학생이 회원가입할 때 지금까지 풀었던 문제들을 저장하는데 사용된다. **Feedback DB**는 학생이 문제를 풀었을 때 ChatGPT가 피드백 작성을

완료하면 피드백 내용이 저장된다. **Comment DB**는 ChatGPT의 피드백을 바탕으로 강사의 피드백을 추가하면 피드백 내용이 저장된다.

10 Index

10.1 Objectives

본 문서에 사용된 그림, 다이어그램, 기능 등의 인덱스가 포함된다.

10.2 Table Index

Table 1 용어 정의	11
Table 2 약어	11
Table 3 Functional Requirement: 회원가입	23
Table 4 Functional Requirement: 로그인	24
Table 5 Functional Requirement: 문제 풀기	25
Table 6 Functional Requirement: 코드 작성 및 제출	25
Table 7 Functional Requirement: 코드 피드백 확인	26
Table 8 Functional Requirement: 푼 문제 확인	26
Table 9 Functional Requirement: 학생들이 작성한 코드 확인	27
Table 10 Functional Requirement: ChatGPT를 사용한 학생 코드 분석 및 시각화	27
Table 11 Functional Requirement: 피드백 작성과 제출	27
Table 12 Tabular Description: 회원가입하기	42
Table 13 Tabular Description: 로그인하기	42
Table 14 Tabular Description: 문제 풀기	42
Table 15 Tabular Description: 코멘트 작성하기	43
	55

10.3 Figure Index

Figure 1 코드 분석 및 개선점 제공 (개선 전)	8
Figure 2 코드 분석 및 개선점 제공 (개선 후)	8
Figure 3 문제 추천	8

10.4 Diagram Index

Diagram 1 System Architecture: 전체 시스템 구조	16
Diagram 2 System Architecture: 사용자 시스템	18
Diagram 3 System Architecture: 문제 추천 시스템	19
Diagram 4 System Architecture: 문제 풀이 시스템	20
Diagram 5 System Architecture: 코드 분석 시스템	21
Diagram 6 System Architecture: 학생 관리 시스템	22
Diagram 7 Context Model	37
Diagram 8 Process Model - 문제 풀기	38
Diagram 9 Process Model - 코멘트 작성	39
Diagram 10 Process Model - 코멘트 확인	40
Diagram 11 Use Case Model	41
Diagram 12 Sequence Diagram: 전체	44
Diagram 13 Sequence Diagram: 회원가입	45

Diagram 14 Sequence Diagram: 문제 풀이	46
Diagram 15 Sequence Diagram: 피드백	47
Diagram 16 Class Diagram	48
Diagram 17 Event Driven Diagram: 문제 풀이	49
Diagram 18 Event Driven Diagram: 코멘트 작성	50
Diagram 19 Event Driven Diagram: 코멘트 확인	51

11 Reference