

Software Requirements Specification

for

DevNavi: Web Coding Platform



2018314692 권민성
2018312018 서주원
2017313987 안서현
2016314729 안정민
2019312497 이장엽
2019314961 한수현

소프트웨어공학개론 3조

2023. 04. 30.

CONTENTS

1. INTRODUCTION	6
1.1. PURPOSE	6
1.2. SCOPE	6
1.3. DEFINITION, ACRONYMS AND ABBREVIATION	7
1.4. REFERENCES	8
1.5. OVERVIEW	9
2. OVERALL DESCRIPTION	10
2.2. PAGE DESCRIPTION	10
2.2.1. Main Page	10
2.2.2. Problem Selection Page	11
2.2.3. Problem Solving (PS) Page	11
2.2.4. PS Feedback Page	11
2.2.5. Coding Test Page	11
2.2.6. Test Result Page	12
2.3. USER CLASSES AND CHARACTERISTICS	13
2.3.1. System Administrator	13
2.3.2. User	13
2.4. OPERATING ENVIRONMENT	13
2.5. DESIGN AND IMPLEMENTATION CONSTRAINTS	13
2.6. ASSUMPTIONS AND DEPENDENCIES	14
3. EXTERNAL INTERFACE REQUIREMENTS	15
3.1. USER INTERFACE	15
3.2. HARDWARE INTERFACE	25
3.3. SOFTWARE INTERFACE	25
3.4. COMMUNICATION INTERFACE	26

4.	SYSTEM REQUIREMENTS.....	28
4.1.	FUNCTIONAL REQUIREMENTS.....	28
4.1.1.	Use Cases	28
4.1.2.	Use Case Diagram	37
4.1.3.	Data Flow Diagram	38
4.2.	NON-FUNCTIONAL REQUIREMENTS.....	39
4.2.1.	Product Requirements	39
4.2.2.	Organizational Requirements	40
4.2.3.	External Requirements	41
4.3.	ORGANIZING SYSTEM FLOW.....	42
4.3.1.	Context Model	42
4.3.2.	Process Model	42
4.3.3.	Interaction Model	43
4.3.4.	Behavior Model	43
4.4.	SYSTEM ARCHITECTURE.....	44
5.	ADDITIONAL INFORMATIONS.....	45
5.1.	SOFTWARE REQUIREMENT SPECIFICATION	45
5.2.	DOCUMENT HISTORY	45

Figures

Figure 1. Design of Main Page	17
Figure 2. Design of Problem List Page	19
Figure 3. Design of Problem Solving Page.....	21
Figure 4. User Interface of Test Result Page	23
Figure 5. Design of Interview Page.....	24
Figure 6. Use Case Diagram.....	37
Figure 7. Data Flow Diagram.....	38
Figure 8. Context Model.....	42
Figure 9. Process Model.....	42
Figure 10. Sequence Diagram	43
Figure 11. System Architecture.....	44

Tables

Table 1. Acronyms/Abbreviation	7
Table 2. Terms and Definitions	8
Table 3. User Interface of Landing Page	15
Table 4. User Interface of Main Page.....	16
Table 5. User Interface of Coding Test Setting Modal	17
Table 6. User Interface of Problem List Page.....	18
Table 7. User Interface of Problem Solving Page.....	20
Table 8. User Interface of Result Page	22
Table 9. User Interface of Interview Page.....	23

Table 10. Hardware Interface for User.....	25
Table 11. Software Interface of DB.....	25
Table 12. Software Interface of AI.....	26
Table 13. Communication Interface of Web Browser.....	26
Table 14. Functional Requirements Abstraction.....	28
Table 15. Use Case of Login.....	29
Table 16. Use Case of Problem List.....	30
Table 17. Use Case of Code Reset.....	30
Table 18. Use Case of Select Programming Language.....	31
Table 19. Use Case of Code Test.....	31
Table 20. Use Case of Code Submit.....	32
Table 21. Use Case of Starting Coding Test.....	32
Table 22. Use Case of Select Problem Number.....	33
Table 23. Use Case of Terminate Coding Test.....	33
Table 24. Use Case of Test Result.....	34
Table 25. Use Case of Test Feedback.....	34
Table 26. Use Case of Test Analysis.....	35
Table 27. Use Case of Select Problems.....	35
Table 28. Use Case of Interview.....	36
Table 29. Use Case of Interview Result.....	36
Table 30. Document History.....	45

1. INTRODUCTION

1.1. Purpose

본 문서는 “DevNavi: Web Coding Platform (이하 DevNavi)” 서비스를 위한 소프트웨어 요구사항 명세서이다. “DevNavi” 서비스는 성균관대학교 2023학년도 1학기 “소프트웨어공학개론” 교과목의 수강생 중 Team 3의 구성원에 의해 설계되고 구현되며, 본 문서에서는 해당 서비스를 위한 요구사항을 정리하고 분석한다. Team 3의 구성원은 본 문서의 내용에 따라 “DevNavi” 서비스를 개발한다.

본 문서의 주요 독자는 Team 3의 구성원이며, 추가로 본 문서의 프로젝트가 진행되는 교과목 “소프트웨어공학개론”의 담당 교수자, 조교진을 독자로 상정한다. 이외에도 해당 교과목의 다른 수강생이나 다른 학기의 수강생 등을 대상으로 교육적 목적이나 비상업적 목적으로 본 문서가 열람되거나 사용되는 것을 허용한다.

본 문서의 목적은 “DevNavi” 시스템을 개발하고 제공하기 위한 요구사항을 정리하고 명시하는 것이다. “DevNavi” 시스템은 사용자에게 양질의 코딩 학습 경험을 제공하기 위해 고안된 코딩 연습 웹 시스템이다. 해당 시스템의 개요를 비롯하여, 시스템의 구성 요소인 UI/UX, 서버, 데이터베이스 사용에 관한 요구사항을 본 문서에서 정리하여 앞으로의 프로젝트 진행 방향의 근거로 삼을 수 있도록 한다.

1.2. Scope

“DevNavi” 시스템은 다양한 난이도의 코딩 문제를 제공함과 동시에 IT 기업의 신입사원 채용 과정에서 지원자 실력을 평가하는 코딩 테스트와 면접 시스템을 제공하여 개발 직무를 희망하는 취업 준비자들의 취업 준비 과정에 도움을 주는 서비스이다. 본 시스템은 사용자가 작성한 코드 및 면접 답변에 대해 AI를 활용한 피드백을 사용자에게 제공함으로써 사용자의 개발자로서의 역량을 성장시킨다.

사용자가 본 시스템을 효과적으로 활용할 수 있도록, 본 시스템은 처음 사용하는 사용자도 적응하기 쉬운 UI/UX를 제공해야 한다. 또한, 시스템 전반을 사용함에 있어 사용자가 불편을 느낄 수

있는 기능적 오류가 없어야 한다. 사용자의 성장에 이바지할 수 있는 적절한 코딩 문제와 코딩 테스트 시스템, 면접 질문과 함께 알맞은 피드백이 제공될 수 있도록 한다.

본 시스템은 JavaScript 기반의 프론트엔드와 백엔드 서버를 구축하여 필요한 기능을 제공하고자 하였다. 전체 기능에 대한 상세는 이하 문서에서 설명하도록 한다.

1.3. Definition, Acronyms and Abbreviation

아래의 표는 본 문서에서 사용된 약어와 그 설명을 나타낸다.

Table 1. Acronyms/Abbreviation

Acronyms/Abbreviation	Description
UI / UX	User Interface / User Experience
DB	Database
OS	Operating System
GUI	Graphical User Interface
URL	User Resource Locator
HTTP	HyperText Transfer Protocol
API	Application Programming Interface

아래의 표는 본 문서에서 사용한 용어와 그 정의를 나타낸다.

Table 2. Terms and Definitions

Term	Definition
Front-end	웹사이트 또는 앱의 사용자 인터페이스(UI)를 구축하는 영역.
Back-end	웹사이트나 앱의 로직, 데이터베이스, 서버 구성 등을 다루는 영역.
GitHub	분산 버전 관리 시스템인 깃(Git)을 사용하여 소스 코드를 호스팅하는 온라인코드 저장소.
React	Facebook에서 개발한 사용자 인터페이스(UI) 라이브러리.
NestJS	TypeScript를 기반으로 하는 프레임워크.
MySQL	오픈 소스 관계형 데이터베이스 관리 시스템 (RDBMS).
OAuth	Open Authorization의 약어로, 인증 및 권한 부여를 위한 프로토콜.
chatGPT	오픈 AI에서 개발한 대화형 인공지능 모델.
Problem Solving	알고리즘과 자료구조를 활용하여 다양한 문제를 해결하는 과정으로, 본 시스템에서는 편의상 PS로도 표현한다.
Coding Test	개발자의 역량을 평가하기 위한 시험.

1.4. References

- "IEEE Guide for Developing System Requirements Specifications". in IEEE Std 1233-1996 , vol., no., pp.1-30, 22 Dec. 1996, doi: 10.1109/IEEESTD.1996.81000.
- Team 2. "CodingCAT: 온라인 코딩 테스트 플랫폼". SKKU, Last Modified: Dec. 12, 2022
https://github.com/skkuse/2022fall_41class_team2
- Team 6. "Web Coding Test". SKKU, Last Modified: Dec. 12, 2022
https://github.com/skkuse/2022spring_41class_team6
- Team 10. "온라인 저지 플랫폼". SKKU, Last Modified: Dec. 12, 2022
https://github.com/skkuse/2022spring_41class_team10

1.5. Overview

본 문서의 이하 내용은 다음과 같이 구성되어 있다. 두 번째 장에서는 제품 관점, 페이지, 운영 환경 등에 관한 내용을 정리하며 시스템에 대한 전체적인 내용을 기록한다. 이후의 장에서는 요구사항을 명시한다. 세 번째 장에서는 외적인 인터페이스에 대한 요구사항을 나타내고, 네 번째 장에서는 기능적·비기능적 요구사항, 시스템 구조 등과 같은 시스템 요구사항을 나타낸다.

2. OVERALL DESCRIPTION

2.1. Product Perspective

오늘날 프로그래머의 진출 분야가 크게 확대되면서 다양한 기업이나 분야에서 필요한 개발자 인재를 찾기 위해 코딩 테스트를 시행하기도 한다. 이러한 테스트들은 응시자가 기본적인 소프트웨어 코딩 역량을 갖추었는지는 물론, 지원한 분야에 적합한 성향과 지식을 갖추고 있는지를 파악하는 데에 중요한 요소가 된다.

하지만, 코딩 테스트 과제를 수행하면서 많은 어려움이 따르기도 한다. 코드 작성 과제를 수행하는 사람은 코드가 길어질 때 자신이 작성한 코드가 올바르게 작성되었는지 확인하는 데에 어려움을 겪을 수 있다. 과제에서 요구하는 내용에 부합하는 코드인지를 확인하려면 단순한 문법적 오류뿐만 아니라, 테스트케이스 통과 여부, 시간복잡도와 같은 코드 효율, 협업 환경을 고려한 가독성 등 다양한 측면에서 코드를 파악해볼 필요가 있다. 아울러, 단순히 코드를 작성하는 것을 넘어 직면한 문제를 어떤 방향으로 해결할 수 있을지 이론과 논리에 근거한 해답을 제시하는 것 또한 개발자의 중요한 능력이다. 따라서, 정성적인 문제 해결 방식에 관해서도 피드백 학습이 이루어질 필요성이 제기된다.

따라서 본 프로젝트에서는 단순히 기존의 코딩 교육 플랫폼처럼 코드가 정확한 출력을 내는지만 확인하는 것이 아니라, AI를 활용하여 다양한 방면에서 학습자의 코드를 분석하며 피드백해주고, 나아가 실제 면접과 유사한 형태의 질의응답 과정을 통해 학습자의 이해도를 테스트하는 시스템을 제공하고자 한다. 이 시스템이 제공하는 서비스를 통해 학습자는 자신이 작성한 코드가 어떻게 하면 더 좋아질지 알 수 있고, 질문과 답변을 통해 자신이 작성한 코드를 잘 이해하고 있는지를 확인하여 더욱 세련된 개발자로 성장할 수 있을 것으로 기대된다.

2.2. Page Description

2.2.1. Main Page

메인 페이지는 학습자가 처음 시스템에 접속하여 로그인을 한 후에 도착하는 페이지다. 메인 페이지의 중앙에는 큰 버튼을 두 개 두는데, 여기서 좌측 버튼은 문제풀이(PS) 페이지로, 우측 버튼은

코딩 테스트 페이지로 이동하는 버튼이다. 코딩 테스트 페이지로 이동하면 테스트의 설정을 조절할 수 있는 모달창이 먼저 뜬 후, 확인이 완료되면 코딩 테스트 페이지로 이동하게 된다.

페이지 상단에는 사이트의 간단한 소개가 있고, 상단 좌측에는 각종 설정을 위한 프로필 버튼이 있다.

2.2.2. Problem Selection Page

메인 페이지에서 PS를 선택했을 때 이동할 수 있는, 문제를 선택할 수 있는 페이지다. 여기서는 DB에 있는 문제들의 리스트를 볼 수 있고, 상단의 검색창을 이용하거나 난이도나 태그에 따라 필터를 걸어서 원하는 문제를 찾을 수 있다.

2.2.3. Problem Solving (PS) Page

문제 선택 페이지에서 선택한 문제를 푸는 페이지다. 본 페이지는 크게 두 부분으로 나누어지게 되는데, 사용자는 좌측에서 문제 설명을 확인하고 우측에서 코드를 작성하거나 실행하게 된다. 페이지 우측의 상단에는 코드를 작성할 수 있는 코드 에디터가 있고, 우측 하단에는 실행 및 채점과 관련된 로그창이 있다. 로그창의 내용은 사용자가 선택하는 하단 버튼에 따라 달라진다. 사용자는 하단의 버튼을 이용하여 언어 변경, 리셋, 코드 실행, 코드 제출을 수행할 수 있다. 또한, 피드백 페이지로 이동할 수 있는 버튼이 있는데, 이 버튼은 처음에는 비활성화되어있다가 문제를 풀었을 때에 한해 활성화된다.

2.2.4. PS Feedback Page

문제 풀이가 끝난 후 피드백을 받는 페이지다. 좌측에는 학습자가 작성한 코드가 표시되고, 우측에는 피드백 내용이 표시된다. 사용자는 상단의 탭을 활용하여 가독성이나 복잡도와 같은 피드백 카테고리를 설정할 수 있다. 선택된 카테고리에 해당하는 피드백을 시스템이 AI에 요청하고, 그 답변이 본 페이지에 표시된다.

2.2.5. Coding Test Page

메인 페이지에서 코딩 테스트를 선택하면 이동하는 페이지다. 단, 본 페이지로 이동하기 전 모

달창에서 문제의 난이도나 문제의 개수를 설정하고 제한 시간을 확인하는 과정이 있다.

기본적으로는 Problem Solving Page와 유사하지만, 좌측에 다른 문제로 이동할 수 있는 부분이 따로 있고, 테스트 종료 전에는 채점 결과를 볼 수 없다는 점에서 차이가 있다. 각 문제에 대한 답을 제출하는 것과는 상관없이, 제한 시간이 초과되거나 사용자가 우측 상단의 시험 종료 버튼을 눌렀을 경우에만 시험이 종료된다. 시험이 종료되면 결과 페이지로 이동한다.

2.2.6. Test Result Page

코딩 테스트가 완전히 종료된 후에 도착하는 페이지다. 사용자는 좌측의 내비게이션 메뉴를 통해 전체 결과, 문제별 피드백, 면접 연습을 선택할 수 있다.

전체 결과는 본 페이지에 진입하면 기본으로 선택되는 메뉴이며, 결과 확인 탭과 피드백 탭으로 나누어져 있다. 결과 확인 탭을 누르면 각 문제별 점수와 소요 시간이 그래프로 표시된다. 그리고 피드백 탭을 누르면 학습자가 테스트에서 작성한 코드의 전체적인 경향성과 문제점을 AI가 분석하여 알려준다.

문제별 피드백은 기본적으로 PS Feedback Page와 유사하게 이루어지지만, 마찬가지로 내비게이션 메뉴가 좌측에 추가로 존재하며 이를 통해 피드백을 볼 문제를 선택할 수 있다.

면접 연습 메뉴를 선택하면 질의응답 형식의 모의 면접을 볼 수 있다. 사용자가 시작 버튼을 누르면 면접이 시작되며, 면접이 끝나기 전까지는 다른 메뉴(전체 결과, 문제별 피드백)로 이동할 수 없다. 모의 면접에서는 시스템이 학습자에게 코드 내용에 관해 질문하고, 그 질문에 대해 학습자가 답변하고, 시스템이 그 답변을 평가하는 세 단계가 반복되게 된다. 학습자에게는 답변을 작성하는 데에 3분의 제한시간이 주어진다.

학습자에게 주어지는 질문에는 크게 두 가지 유형이 있다. 먼저, 컴퓨터 과학에 대한 전반적인 지식에 대한 질문이 주어질 수 있다. 이 유형의 질문은 시스템에 사전에 저장되어야 하며, AI는 이 유형의 질문에 대해 작성된 답변을 평가할 때에만 이용하게 된다. 다음으로, 테스트에서 푼 문제 혹은 학습자가 작성한 코드를 학습자가 제대로 이해하고 있는지 확인하는 질문이 주어질 수 있다. 상황에 따라 다양한 질문이 주어져야 하므로, AI는 이 유형의 질문을 작성할 때와 답변을 평가할 때에 모두 사용되어야 한다.

2.3. User Classes and Characteristics

2.3.1. System Administrator

‘시스템 관리자’는 이 시스템을 운영, 관리하고 시스템의 오류를 확인하는 사람을 의미한다. 시스템 관리자는 오류나 버그가 발생할 때에만 시스템과 상호작용한다. 따라서 시스템과 관련한 지식을 모두 보유해야 한다.

2.3.2. User

사용자는 이 시스템을 통해 코딩 테스트에 대한 피드백을 제공받고 코딩 테스트에 대한 면접 시뮬레이션을 경험할 수 있다. 사용자는 Chrome과 같은 모던 웹 브라우저를 통해 이 시스템에 접속하고, 자신이 사용할 수 있는 언어로 코드 작성 과제를 수행하고 이를 기반으로 AI와의 가상 면접을 진행한다.

2.4. Operating Environment

- Linux
- Mac OS
- Windows 11
- Windows 10
- Windows 8
- Windows 7

2.5. Design and Implementation Constraints

본 시스템은 아래의 기재된 제한사항들을 토대로 개발된다. 해당 항목에 포함되지 않는 내용은 개발 상황과 환경에 맞추어 설계된다.

- 가능한 범위 내에서 최대한 오픈소스를 활용한다.
- 오픈소스 활용 시, 해당 소스의 신뢰도를 검토한다.
- 서비스 운영에 필수적인 유저 정보만 활용한다.

- 변경된 요구사항을 반영할 수 있도록 유지보수성을 갖춘 시스템을 개발한다.
- 사용자를 고려한 사용자 친화적인 인터페이스를 구축한다.

2.6. Assumptions and Dependencies

본 문서의 시스템은 웹 기반의 시스템이다. 따라서 사용자가 웹 브라우저를 통해 정상적으로 접속하여 서비스를 사용한다고 가정한다. 웹 브라우저 중에서는 Chrome 브라우저에 가장 최적화된 서비스를 제공한다.

위의 '2.4. Operating Environment' 항목에서의 환경을 만족하는 사용자의 접근을 기본 전제로 하고 이에 한해 정상 동작을 보장한다.

3. EXTERNAL INTERFACE REQUIREMENTS

3.1. User Interface

Table 3. User Interface of Landing Page

이름	랜딩 페이지
목적/설명	사용자가 처음으로 도메인에 접속하면 볼 수 있는 초기 페이지이다. 사용자는 중앙에 있는 GitHub OAuth 버튼을 클릭하여 GitHub 인증 페이지로 리다이렉트된다. 사용자가 GitHub 계정으로 인증을 완료하면 메인 페이지로 리다이렉트되고 인증이 실패하면 랜딩 페이지로 리다이렉트된다.
입력 소스/ 출력 대상	입력: 사용자의 GitHub 계정 인증 정보 출력: 메인 페이지
범위/정확도 /오차범위	올바른 GitHub 계정
시간/속도	웹 브라우저와 Front 사이의 통신 시간 Front-end 서버와 Back-end 서버 사이의 통신 시간 Back-end 서버와 GitHub Authentication 서버 사이의 통신 시간
기타 입·출력관계	사용자의 인증 실패 혹은 인증 취소에 따라 랜딩 페이지로 리다이렉트되는 경우
화면 형식과 구성	중앙 상단부의 서비스 로고 중앙 하단부의 GitHub 로그인 버튼
데이터 유형	N/A
명령 유형	버튼 클릭에 의한 코드 실행 사용자의 추가적인 GitHub 인증 단계
종료 메시지	N/A

Table 4. User Interface of Main Page

이름	메인 페이지
목적/설명	사용자가 로그인 시 전환되는 메인 페이지이다. Problem List 버튼을 클릭하면 Problem List 페이지로 전환된다. Coding Test 시작 버튼을 클릭하여 Coding Test Setting 모달창을 열 수 있다.
입력 소스/ 출력 대상	입력: N/A 출력: Problem List Page, Coding Test Setting Modal
범위/정확도 /오차범위	N/A
시간/속도	웹 브라우저와 Front-end 서버 사이의 통신 시간
기타 입·출력관계	N/A
화면 형식과 구성	중앙부 상단의 서비스 관련 텍스트 중앙부 좌측의 Problem List 버튼 중앙부 우측의 Coding Test 버튼
데이터 유형	N/A
명령 유형	버튼 클릭에 의한 코드 실행
종료 메시지	N/A



Figure 1. Design of Main Page

Table 5. User Interface of Coding Test Setting Modal


이름	Coding Test Setting Modal
목적/설명	코딩 테스트를 시작하기 전 시험 관련 설정을 하는 모달이다. 문제들의 평균 레벨과 문제 수를 선택할 수 있다. 문제 레벨 탭 클릭 시 난이도 중 하나를 선택할 수 있으며 해당 난이도의 ± 1 범위의 문제들이 코딩 테스트 문제로 출제된다. 제한 시간은 문제 레벨과 수에 따라 정해지며 임의로 수정될 수 없다. 코딩 테스트 시작 버튼을 클릭하여 코딩 테스트를 시작할 수 있다.
입력 소스/출력 대상	입력: Problem Level, Problem Number 출력: Coding Test Page
범위/정확도/오차범위	난이도: 1 ~ 10 문제 수: 1 ~ 10 키 입력이 아닌 드롭다운 메뉴로 설정되므로 오차가 발생하지 않는다.
단위	N/A
시간/속도	웹 브라우저와 Front-end 서버 사이의 통신 시간

기타 입출력관계	N/A
화면 형식과 구성	<p>모달 창의 영역은 세로 정렬로 4가지 영역이 존재한다.</p> <ul style="list-style-type: none"> • 문제 레벨 선택 메뉴 • 문제 수 선택 메뉴 • 설정된 문제 레벨과 수에 따른 코딩 테스트 제한 시간(수정 불가) • 코딩 테스트 시작 버튼
데이터 유형	난이도 및 문제 수 (정수형)
명령 유형	버튼 클릭에 의한 코드 실행
종료 메시지	N/A

Table 6. User Interface of Problem List Page

이름	Problem List Page
목적/설명	<p>문제 리스트를 보여주는 페이지이다.</p> <p>난이도나 태그를 포함한 필터와 검색어의 조건에 맞는 문제가 표시된다.</p> <p>난이도를 설정할 수 있고 태그를 추가하거나 제거할 수 있다. 검색어를 통해 문제 검색을 할 수 있다.</p> <p>난이도 탭을 클릭하면 드롭다운 메뉴가 켜지고, 검색하고자 하는 난이도를 복수 선택할 수 있다. 확인 시 해당 난이도 조건을 기반으로 문제 리스트가 갱신된다.</p> <p>태그 추가 버튼을 사용하여 존재하는 알고리즘 태그 중 하나를 선택하고 필터에 추가할 수 있다. 태그 중 하나를 선택하면 삭제할 수 있는 'X' 표시가 생성된다. 태그의 추가나 제거가 발생하면 문제 리스트가 갱신된다.</p> <p>문제 리스트 중 특정 문제를 클릭 시 Problem Solving Page로 전환된다.</p>
입력 소스/ 출력 대상	<p>입력: 난이도, 태그, 검색어</p> <p>출력: 조건에 맞는 문제 리스트</p>
범위/정확도 /오차범위	<p>난이도: 1 ~ 10</p> <p>알고리즘 태그: 빈 문자열 제외</p> <p>검색어: 빈 문자열 제외</p>

시간/속도	웹 브라우저와 Front-end 서버 사이의 통신 시간 Front-end와 Back-end 서버 사이의 통신 시간 Back-end 서버와 DB 사이의 통신 시간
기타 입·출력관계	검색어에 빈 문자열이라면 검색 버튼이 비활성화
화면 형식과 구성	최상단에 검색을 할수있는 검색바가 위치한다. 검색어를 입력 후 검색을 하면 문제 리스트가 갱신된다. 바로 아래에는 난이도, 태그 등의 필터를 설정할 수 있는 바가 있다. 필터의 왼쪽에는 난이도 탭, 오른쪽에는 태그 설정 탭이 존재한다. 필터 밑에는 검색어와, 필터 정보에 해당하는 문제 리스트가 보여진다.
데이터 유형	검색어, 알고리즘 태그 (문자열) 문제 난이도 (정수형)
명령 유형	버튼 클릭에 의한 코드 실행 검색어 입력
종료 메시지	N/A

 코딩 연습 모드>

난이도 ▼
태그

문제 선택하기

문제 번호	문제 제목	난이도
1	쿼리의 모음의 개수	Lv.5
2	중력 작용	Lv.5
3	RPG와 쿼리	Lv.4
4	안티세포	Lv.3
5	문자열의 아름다움	Lv.3
6	사라지는 발판	Lv.3

Figure 2. Design of Problem List Page

Table 7. User Interface of Problem Solving Page

이름	Problem Solving Page
목적/설명	<p>사용자가 문제를 풀 수 있는 페이지이다.</p> <p>code editor를 통해 solution에 대한 code를 작성할 수 있다.</p> <p>작성한 코드를 기반으로 sample test cases를 test하거나 실제 제출을 통해 모든 test cases를 채점을 받을 수 있다. 코드 제출 버튼을 클릭시 모든 test case를 채점하고 결과를 시스템 출력 영역에 표시한다.</p> <p>테스트와 채점 결과를 확인할 수 있는 시스템 출력 영역이 있다.</p> <p>채점을 받을 programming language를 선택할 수 있다. 언어 선택 버튼을 클릭하여 지원하는 언어 중 하나를 선택할 수 있다. 클릭 시 confirm 창이 보여지며 확인을 클릭 시 해당 언어의 default code로 초기화 된다.</p> <p>code를 리셋하여 해당 언어의 default code로 초기화할 수 있다. 코드 리셋 버튼 클릭시 confirm 창이 보여지며 확인을 클릭 시 해당 언어의 default code로 초기화 된다.</p> <p>피드백 버튼 클릭시 사용자의 code에 대한 AI의 피드백을 확인하는 피드백 페이지로 전환된다. 피드백 버튼은 문제를 맞추기 전까지는 비활성화 되어있으며, 모든 test case를 맞출 시 활성화된다. 피드백 버튼을 클릭하면 피드백 페이지로 전환된다.</p>
입력 소스/ 출력 대상	<p>입력: 코드</p> <p>출력: 테스트 및 채점 결과</p>
범위/정확도 /오차범위	N/A
시간/속도	<p>웹 브라우저와 Front-end 서버 사이의 통신 시간</p> <p>Front-end와 Back-end 서버 사이의 통신 시간</p> <p>코드 채점 시간</p>
기타 입·출력관계	N/A

화면 형식과 구성	<p>화면은 크게 3개의 영역으로 나누어진다.</p> <p>좌측 영역은 문제에 대한 설명을 보여준다. 최상단 라벨에는 문제 제목이 있으며 아래에 문제 설명이 있다.</p> <p>우측 상단은 사용자가 입력할 수 있는 코드 에디터 영역이다.</p> <p>우측 하단은 시스템 출력의 영역으로 테스트 및 채점에 대한 결과를 보여준다. 바로 아래에는 버튼 메뉴들이 존재하며 왼쪽부터 각각 언어 선택 버튼, 코드 리셋 버튼, 테스트 실행 버튼, 코드 제출 버튼, 피드백 버튼이 있다.</p>
데이터 유형	Text
명령 유형	<p>버튼 클릭에 의한 코드 실행</p> <p>유저의 코드 입력</p>
종료 메시지	N/A

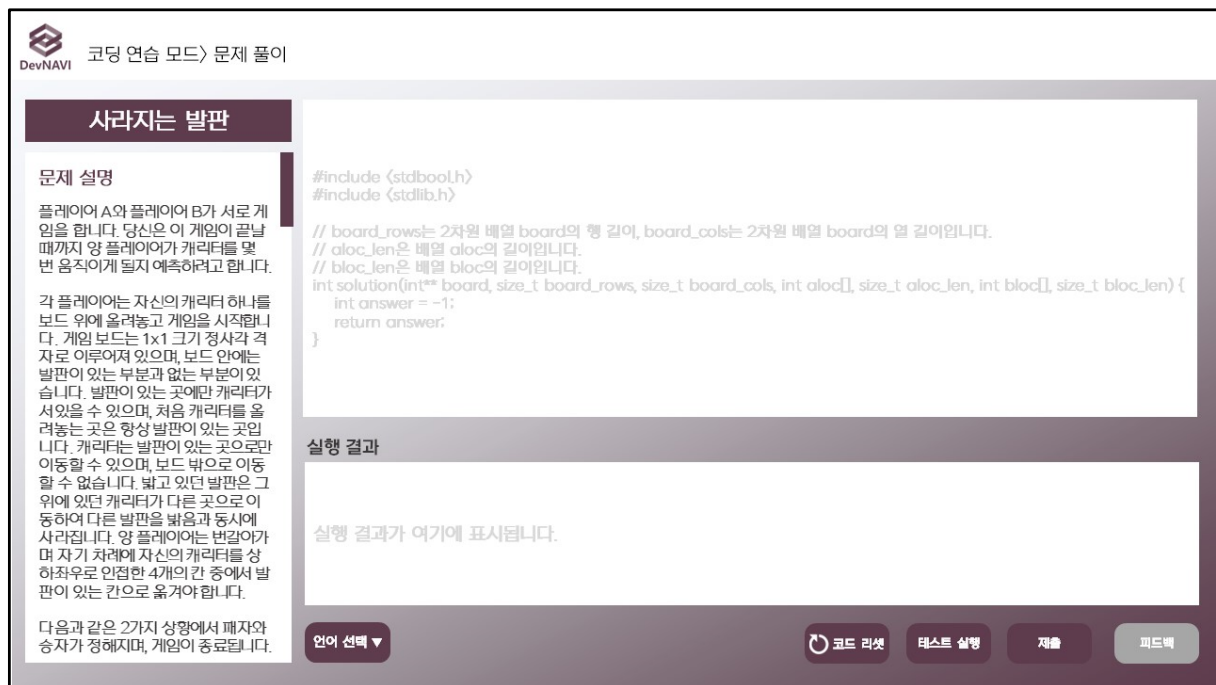


Figure 3. Design of Problem Solving Page

Table 8. User Interface of Result Page

이름	Test Result Page
목적/설명	<p>사용자의 코드를 채점한 결과를 보여주는 페이지이다.</p> <p>default 상태는 사용자의 점수와 측정된 시험 시간을 표시한다.</p> <p>화면 좌측에는 “전체 결과” 버튼, 테스트의 각 문항 번호가 적힌 버튼들과 “면접” 버튼이 존재한다.</p> <p>“전체 결과” 버튼을 클릭하면 default 상태로 이동한다.</p> <p>번호가 적힌 버튼을 클릭하면 클릭한 번호 문항의 상세 결과가 화면 중앙에 나타납니다. 상세 결과에는 사용자가 작성한 코드와 AI가 개선한 코드가 나타납니다. 그리고 가독성, 시간복잡도 버튼을 클릭하여 AI가 제공하는 피드백을 확인할 수 있다.</p> <p>면접 버튼 클릭 시 면접 시뮬레이션 페이지로 이동한다.</p> <p>화면 상단에는 “분석” 버튼과 “피드백” 버튼이 존재한다. 두 버튼은 클릭 시에 새로운 화면이 팝업 형식으로 나타난다. “분석” 버튼을 클릭하면 문제별 점수와 시간 그래프가 나타난다. “피드백” 버튼을 클릭하면 AI가 사용자의 코딩 성향을 분석하여 코드의 전체적인 경향성을 알려주고 전체적인 문제점을 보여준다.</p>
입력 소스/ 출력 대상	<p>입력: 문제 번호</p> <p>출력: 시험 결과 (API의 응답)</p>
범위/정확도 /오차범위	N/A
시간/속도	API의 응답 시간
기타 입·출력관계	N/A
화면 형식과 구성	<p>상단에 “가독성”, “시간복잡도”, “개선된 코드” 탭이 존재하고 좌측에는 “전체 결과”, 문제 번호, “분석”, “피드백” 버튼이 존재한다.</p> <p>중앙은 좌우 두 부분으로 나누어지고 왼쪽에는 사용자 코드가 나타나고 오른쪽에는 가독성, 시간복잡도, 개선된 코드 등 선택한 탭의 결과가 나타난다.</p> <p>좌측의 문제 번호를 선택하면 해당 문항의 사용자 코드가 중앙 왼쪽에 나타나고 가독성이 default로 오른쪽에 나타난다. 또한 분석, 피드백 버튼을 클릭하면 팝업 형식의 새로운 창이 나타난다. 이 창이 열리거나 닫히는 경우 기존 페이지는 그대로 유지된다.</p> <p>우하단에는 집 모양의 버튼이 존재하는데, 이 버튼으로 메인 페이지로 이동할 수 있다.</p>

데이터 유형	텍스트, 그래프
명령 유형	버튼 클릭에 의한 이벤트 핸들
종료 메시지	N/A

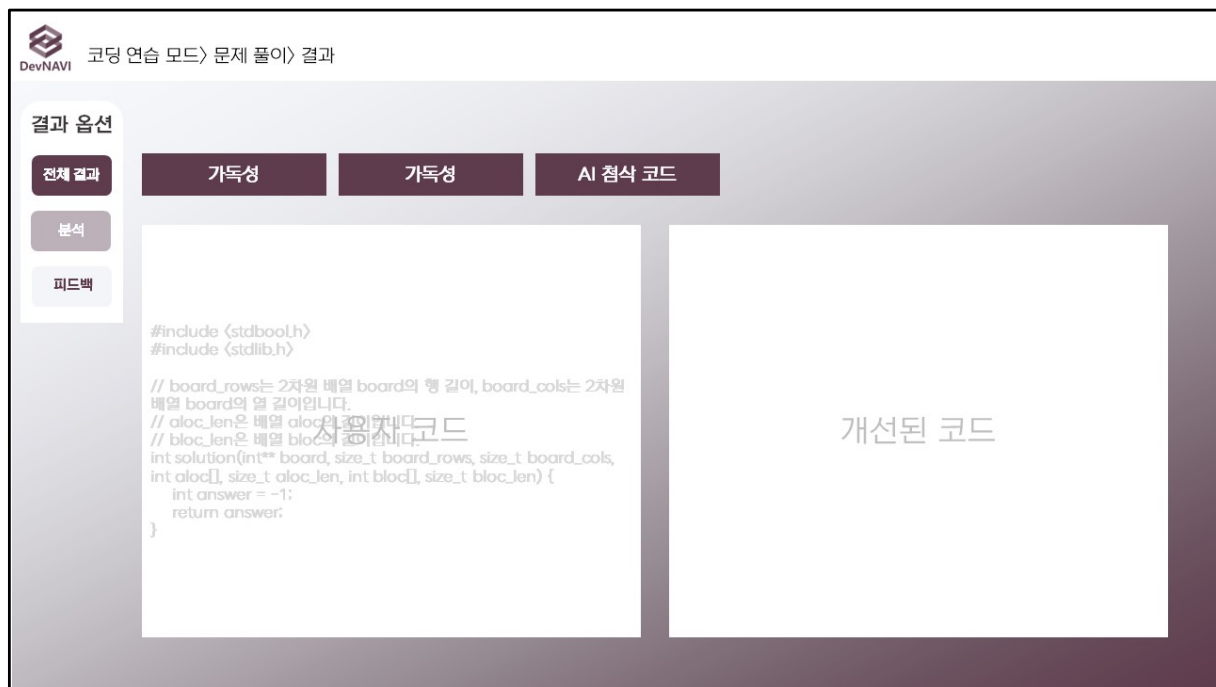


Figure 4. User Interface of Test Result Page

Table 9. User Interface of Interview Page

이름	Interview Page
목적/설명	<p>사용자가 AI 를 활용한 가상 면접 시뮬레이션을 진행하는 페이지이다.</p> <p>이 때 면접은 사용자가 진행했던 코딩 테스트에 기반하여 문제가 생성되기도 하고 컴퓨터 과학과 관련된 기본 지식들에 대한 질문도 생성이 된다.</p> <p>사용자는 제한 시간내에 주어진 문제에 응답을 하여 제출을 해야한다. 이미 응답했거나 제한 시간이 지난 문제는 읽거나 수정할 수 없다.</p> <p>AI 가 사용자의 응답을 채점하여 결과를 도출한다.</p>
입력 소스/ 출력 대상	<p>입력: 사용자의 응답</p> <p>출력: 면접 결과 (API 의 응답)</p>

범위/정확도 /오차범위	N/A
시간/속도	API의 응답 시간
기타 입·출력관계	N/A
화면 형식과 구성	화면은 좌측의 진행중인 문항 번호들이 존재한다. 중앙은 상단과 하단으로 나눌 수 있는데, 상단은 AI 가 제시하는 문제가 나타나고 하단에는 사용자가 답안을 작성한다. 우하단의 제출 버튼을 누르면 답안이 제출된다. 우상단에는 남은 제한 시간이 표시되고 면접을 종료할 수 있는 버튼인 “면접 종료” 버튼도 존재한다.
데이터 유형	텍스트
명령 유형	버튼 클릭에 의한 실행
종료 메시지	N/A

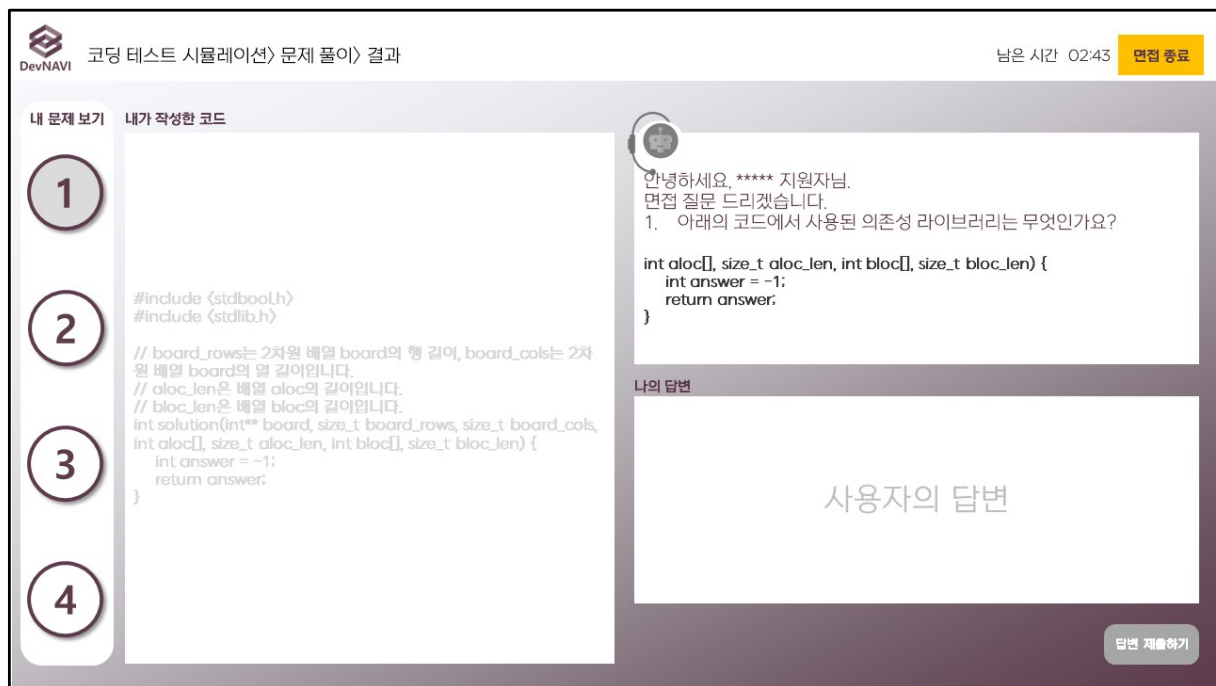


Figure 5. Design of Interview Page

3.2. Hardware Interface

Table 10. Hardware Interface for User

이름	사용자가 시스템을 실행하는 기기
목적/설명	사용자가 시스템의 웹페이지에 접근하고 해당 시스템을 사용할 수 있도록 한다.

3.3. Software Interface

Table 11. Software Interface of DB

이름	MySQL
목적/설명	사용자 및 문제 정보를 입출력하기 위한 쿼리 데이터
입력 소스/ 출력 대상	입력: 백엔드 서버 출력: 데이터베이스
범위/정확도	MySQL의 정확도
구성 단위	쿼리
시간/속도	즉시 반응
기타 입·출력 관계	서버의 모든 입출력 데이터와 연관
화면 형식과 구성	N/A
데이터 유형	MySQL 쿼리
명령 유형	MySQL 명령어
종료 메시지	N/A

Table 12. Software Interface of AI

이름	ChatGPT
목적/설명	학습자의 피드백을 위한 AI의 답변을 얻기 위한 API 요청
입력 소스/ 출력 대상	입력: 백엔드 서버 출력: ChatGPT
범위/정확도	ChatGPT의 정확도
구성 단위	텍스트(문자열)
시간/속도	ChatGPT의 반응속도
기타 입·출력 관계	출력 텍스트가 학습자의 결과 평가/피드백에 사용됨
화면 형식과 구성	N/A
데이터 유형	텍스트
명령 유형	텍스트 전송
종료 메시지	N/A

3.4. Communication Interface

Table 13. Communication Interface of Web Browser

이름	웹 브라우저
목적/설명	시스템 사용자는 웹 브라우저를 통해 본 서버와 통신하며 사이트에서 제공하는 기능을 사용할 수 있다.
입력 소스/ 출력 대상	입력: 사용자의 하드웨어 출력: 프론트엔드 서버
범위/ 정확도	해당 브라우저가 사용하는 프로토콜의 정확도

구성 단위	패킷
시간/속도	네트워크와 해당 프로토콜 알고리즘의 속도
기타 입·출력 관계	사용자와 서버 간 모든 데이터 입출력
화면 형식과 구성	사용자가 사용하는 웹 브라우저의 화면
데이터 유형	웹 브라우저가 사용하는 프로토콜 데이터
명령 유형	해당 프로토콜의 명령
종료 메시지	N/A

4. SYSTEM REQUIREMENTS

4.1. Functional Requirements

4.1.1. Use Cases

Table 14. Functional Requirements Abstraction

Functional Requirement No.	Name of Function	Function Description
F1	로그인	사용자가 GitHub 계정으로 서비스에 로그인을 할 수 있어야 한다.
F2	문제 리스트 확인	사용자는 필터 및 검색을 기반으로 문제 리스트를 확인할 수 있어야 한다.
F3	코드 수정	사용자는 코드 에디터에서 코드를 수정할 수 있어야 한다.
F4	코드 언어 선택	사용자는 프로그래밍 언어를 선택할 수 있어야 한다.
F5	코드 리셋	사용자가 코드를 초기값으로 초기화할 수 있어야 한다.
F6	코드 테스트	사용자는 기본으로 주어진 테스트 케이스를 테스트를 할 수 있어야 한다.
F7	코드 제출	사용자는 숨겨진 테스트 케이스들을 채점 받을 수 있어야 한다.
F8	코딩 테스트 시작	사용자는 코딩 테스트에 대한 설정을 하고 테스트 시작을 할 수 있어야 한다.
F9	코딩 테스트 문제 선택	사용자는 코딩 테스트를 치는 동안 문제를 선택할 수 있어야 한다.
F10	코딩 테스트 종료	사용자는 시간이 끝나지 않아도 시험을 종료할 수 있다.

F11	코딩 테스트 결과 확인	사용자는 코딩 테스트를 마친 후 결과 화면을 볼 수 있어야 한다.
F12	문항 별 결과 확인	사용자는 코딩 테스트의 각 문항 별 결과를 확인할 수 있어야 한다.
F13	결과 분석 확인	사용자는 코딩 테스트의 결과를 분석한 화면을 확인할 수 있어야 한다.
F14	피드백 확인	사용자는 코드에 대한 각 항목별 피드백을 확인할 수 있어야 한다.
F15	면접 진행	사용자는 면접 화면으로 이동하여 질문을 확인할 수 있어야 한다.
F16	면접 채점	사용자는 면접에 대한 응답을 채점한 결과를 확인할 수 있어야 한다.

Table 15. Use Case of Login

Name	Login
Actors	유저
Description	사용자는 GitHub 계정을 통해 서비스에 로그인할 수 있다.
Input Data	None
Output Data	None
Stimulus	사용자가 GitHub 인증 버튼을 클릭한다.
Response	GitHub 인증 페이지로 redirect되고 인증 후 메인 페이지로 진입할 수 있다.
Comments	

Table 16. Use Case of Problem List

Name	Problem List
Actors	유저
Description	사용자가 필터와 검색을 기반으로 문제 목록을 확인한다.
Input Data	tags, search_word
Output Data	prblem_list
Stimulus	검색어를 입력하고 검색을 클릭한다. 필터를 추가하거나 제거한다.
Response	검색어와 필터를 종합하여 해당하는 문제 리스트를 불러온다.
Comments	

Table 17. Use Case of Code Reset

Name	Code Reset
Actors	학생
Description	사용자가 작성한 코드를 기본 코드로 초기화한다.
Input Data	None
Output Data	None
Stimulus	사용자가 코드 리셋 버튼을 클릭한다.
Response	confirm box가 열리고 확인 시 언어에 따른 기본 코드로 초기화된다.
Comments	

Table 18. Use Case of Select Programming Language

Name	Select Programming Language
Actors	유저
Description	사용자는 에디터의 코드 언어를 변경할 수 있다.
Input Data	None
Output Data	default_code, language_info
Stimulus	사용자가 프로그래밍 언어 탭을 클릭하고 원하는 언어를 클릭한다.
Response	클릭 시 confirm box가 뜨고 확인을 누르면 선택한 언어의 default code로 초기화 된다.
Comments	

Table 19. Use Case of Code Test

Name	Code Test
Actors	유저
Description	사용자가 작성한 코드를 테스트 케이스를 기반으로 테스트한다.
Input Data	user_code, testcase
Output Data	test_result
Stimulus	사용자가 코드 테스트 버튼을 클릭한다.
Response	각 테스트 케이스에 대한 결과가 출력된다.
Comments	

Table 20. Use Case of Code Submit

Name	Code Submit
Actors	유저
Description	사용자가 코드를 제출하여 채점을 받는다.
Input Data	user_code
Output Data	user_code
Stimulus	사용자가 코드 제출 버튼을 클릭한다.
Response	제출 확인 confirm box의 확인을 클릭하면 피드백 페이지로 넘어간다.
Comments	

Table 21. Use Case of Starting Coding Test

Name	Starting Coding Test
Actors	유저
Description	사용자가 코딩 테스트에 관련된 설정을 하고 테스트를 시작한다.
Input Data	problem_rank, problem_number
Output Data	coding_test_problems
Stimulus	사용자가 문제의 난이도와 문제 수를 설정하고 코딩 테스트 시작 버튼을 클릭한다.
Response	코딩 테스트 페이지로 redirect된다.
Comments	

Table 22. Use Case of Select Problem Number

Name	Select Problem Number
Actors	유저
Description	사용자가 문제의 번호를 클릭하여 해당 문제를 확인한다.
Input Data	None
Output Data	problem_info, temp_saved_code
Stimulus	사용자가 문제 번호를 클릭한다.
Response	선택한 문항의 정보와 해당 문제에 임시로 저장된 유저의 code를 불러온다.
Comments	

Table 23. Use Case of Terminate Coding Test

Name	Terminate Coding Test
Actors	유저
Description	사용자가 시간이 남은 상황에서 코딩 테스트를 종료한다.
Input Data	None
Output Data	None
Stimulus	사용자가 코딩 테스트 종료 버튼을 클릭한다.
Response	종료 후 로딩 페이지로 redirect된다.
Comments	

Table 24. Use Case of Test Result

Name	Test Result
Actors	User
Description	사용자가 시험을 종료한 뒤 결과를 확인한다.
Input Data	None
Output Data	coding_test_result
Stimulus	사용자가 시험 종료를 하거나 시험 시간이 종료된다. 출력된 결과 화면에서 “전체 결과” 버튼을 클릭한다.
Response	결과 화면을 출력한다.
Comments	결과 데이터를 로드하는 시간이 필요하다.

Table 25. Use Case of Test Feedback

Name	Test Feedback
Actors	User
Description	사용자의 시험 결과에 대한 피드백을 제공한다. 사용자가 작성한 코드를 분석하여 코드에 대한 문제와 개선 방향이 제공된다.
Input Data	None
Output Data	test_feedback
Stimulus	사용자가 결과 화면에서 피드백 버튼을 클릭한다.
Response	피드백 화면을 출력한다.
Comments	

Table 26. Use Case of Test Analysis

Name	Test analysis
Actors	User
Description	사용자의 시험 결과에 대한 분석을 제공한다. 문제 별 점수 그래프와 시간 그래프가 제공된다.
Input Data	None
Output Data	test_analysis
Stimulus	사용자가 시험 결과 화면에서 분석 버튼을 클릭한다.
Response	분석 화면을 출력한다.
Comments	

Table 27. Use Case of Select Problems

Name	Select Problems
Actors	User
Description	코딩 테스트의 각 문항 별 결과를 제공한다.
Input Data	problem_num
Output Data	selected_problem_result
Stimulus	사용자가 문항의 번호가 적힌 버튼을 클릭한다.
Response	문항별 결과를 출력한다.
Comments	

Table 28. Use Case of Interview

Name	Interview
Actors	User
Description	사용자가 코딩 테스트 이후 면접을 진행한다. 한 질문당 제한 시간이 존재한다.
Input Data	test_set
Output Data	interview_questions
Stimulus	사용자가 코딩 테스트 결과 화면에서 면접 버튼을 클릭한다.
Response	사용자에게 제공될 질문을 생성하여 화면에 출력한다.
Comments	1. 출제된 문제 관련 질문 생성 2. CS 관련 기본 지식 질문 (고정 라이브러리에서 추출)

Table 29. Use Case of Interview Result

Name	Interview Result
Actors	User
Description	사용자가 제출한 면접 질문에 대한 응답을 채점하고 올바른 정답을 출력한다.
Input Data	user_interview_response
Output Data	interview_result
Stimulus	사용자가 면접 질문에 대한 응답을 완료한다. 질문에 응답할 수 있는 제한 시간이 초과된다.
Response	사용자의 응답을 정답과 비교하고 틀렸다면 정답을 출력한다.
Comments	

4.1.2. Use Case Diagram

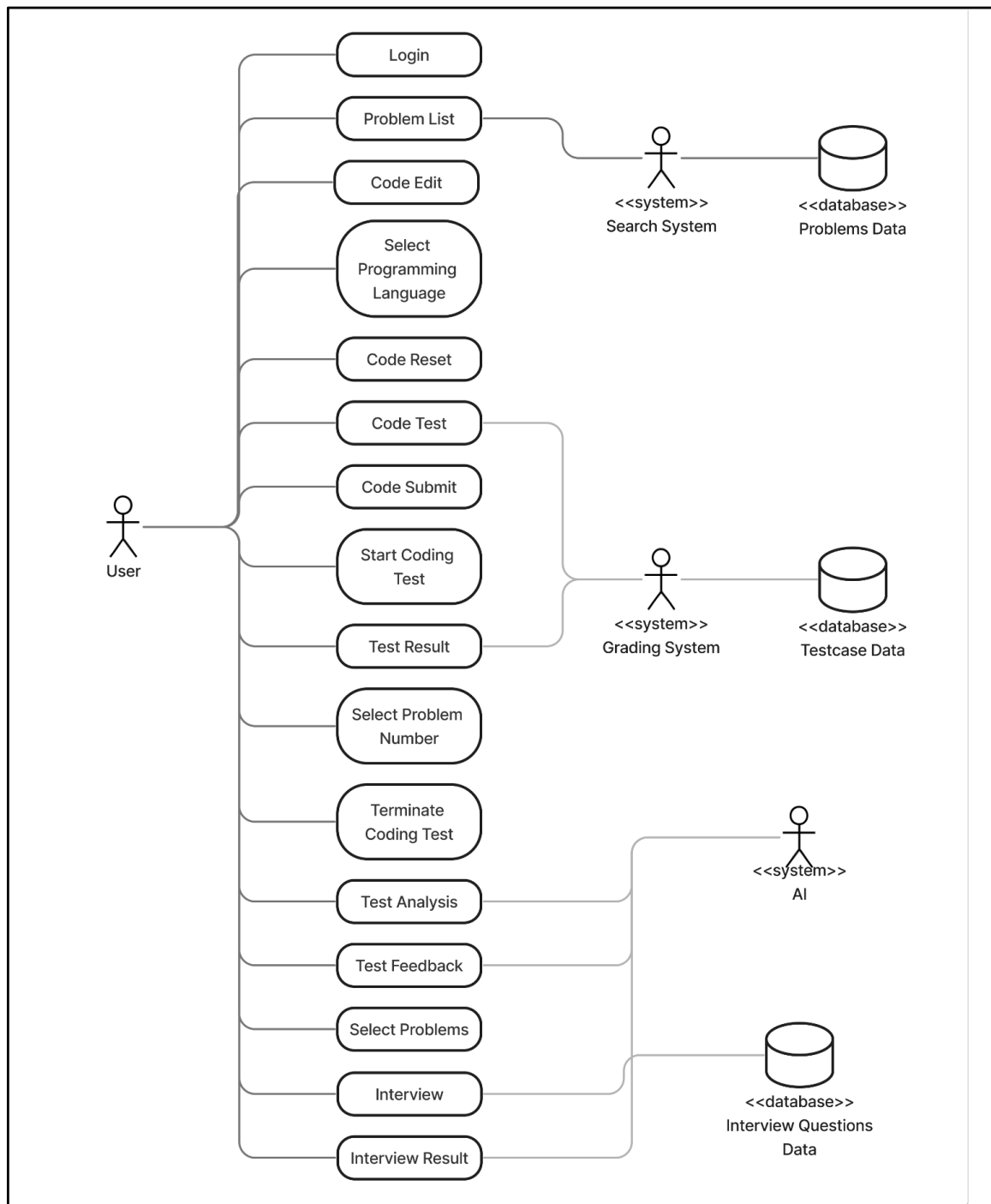


Figure 6. Use Case Diagram

4.1.3. Data Flow Diagram

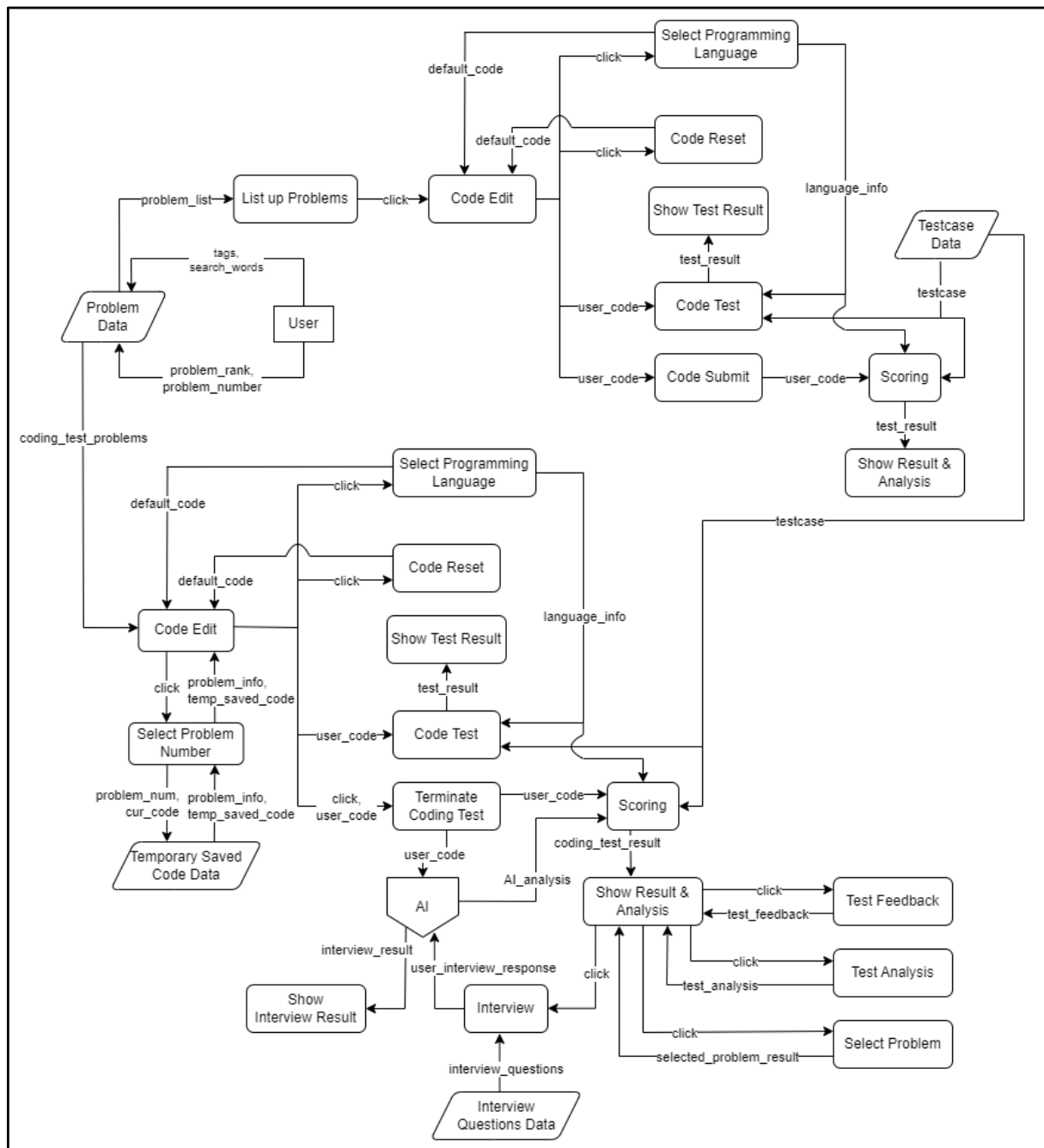


Figure 7. Data Flow Diagram

4.2. Non-functional Requirements

시스템이 제공하는 서비스나 시스템 속성 및 제약 조건을 정의한다.

4.2.1. Product Requirements

특정한 방식으로 작동해야 하는 것을 명시하는 요구사항이다.

4.2.1.1. Usability Requirements

본 시스템은 사용자가 Problem Solving 연습을 하거나 Coding Test를 할 환경을 웹 애플리케이션 기반으로 제공한다. 사용자는 현대의 웹 브라우저를 통해 시스템을 사용하는 것이 요구되며, Chrome 브라우저 사용이 권장된다. 본 시스템은 브라우저 이외의 다른 요구사항을 사용자에게 요구하지 않아야 한다.

본 시스템을 이용하는 사용자는 문제 해결 과정이나 결과에 직접적으로 영향을 미치는 코딩 능력에 따라 시스템의 기능을 제한적으로 이용하게 될 수 있다. 그러나 코딩 능력이 아닌 다른 요인에 따라 시스템의 기능을 이용하는 데에 제한이 발생하지는 않아야 하고, 사용자는 큰 불편사항 없이 간단한 매뉴얼만으로도 시스템의 모든 기능을 수행하면서 Problem Solving과 Coding Test 수행에 집중할 수 있어야 한다.

4.2.1.2. Performance Requirements

본 서비스는 다중 사용자의 이용을 허용하며, 각 사용자는 OAuth2.0 기반으로의 로그인 세션으로 구분된다.

- 본 시스템은 최대 100인의 동시 이용을 가정하며, 이를 초과하는 인원에는 최적의 성능을 제공하는 데에 문제가 발생할 수 있다.
- 네트워크 연결과 서버의 상태가 정상적이라면 모든 서비스 이용에는 일반적으로 10초 이상의 지연이 발생하지 않아야 한다.
- 사용자의 코드 검증에는 일반적으로 테스트케이스당 5초의 실행 제한시간이 주어지며, 초과 시 Timeout으로 처리한다.

- 데이터베이스에서 자료를 탐색하지 않거나 데이터베이스의 수정이 이루어지지 않는 간단한 동작에는 5초 이상의 지연이 발생하지 않아야 한다.

4.2.1.3. Dependability Requirements

- 사용자의 동작에 따라 생성된 데이터는 데이터베이스에 저장되고 보존되어야 한다.
- 코드를 비롯하여 사용자가 입력한 데이터는 요구되는 프로세스에 따라 처리되어야 한다.
- 정해진 프로세스에 따라 처리된 결과는 사용자에게 시각적으로 전달되어야 한다.

4.2.2. Organizational Requirements

사용하는 프로세스 표준이나 구현 요구사항 등과 같이 조직의 정책이나 방침 혹은 절차상의 결과로 발생하는 요구사항이다.

4.2.2.1. Environmental Requirements

본 시스템은 필요한 서비스를 Web 기술에 기반하여 제공하며, 본 시스템의 사용자가 PC 환경의 웹 브라우저로 본 시스템에 접속한다는 가정하에 올바른 작동을 보장한다. 이외의 환경에서의 작동은 올바르지 않을 수 있다.

사용자와의 직접적인 상호작용이 발생하는 Front-end 서버의 개발에는 Node.js, HTML, CSS에 기반한 React 라이브러리가 사용된다.

Back-end 서버의 개발에는 NestJS framework가 사용되며, 데이터베이스의 관리에는 MySQL을 사용한다.

개발자는 라이선스 문제나 다른 API와의 충돌 문제가 없을 것으로 판단된다면 개발 과정에서 오픈소스 소프트웨어를 적극적으로 시스템에 포함할 수 있다. 단, 본 문서가 작성된 이후에 시스템에 새롭게 포함되는 내용은 Git에 기록되거나 추후 본 문서에 반영되어야 한다.

4.2.2.2. Operational Requirements

본 시스템은 본 문서에서 다루어지는 요구사항 명세의 범위 내에서는 서비스 이용에 장애가 발생하지 않아야 한다.

본 시스템의 사용자는 이 시스템을 통해 소프트웨어 엔지니어로서의 문제해결 능력을 비롯한 코딩 테스트 수행 능력을 함양할 수 있어야 한다.

4.2.3. External Requirements

시스템 및 개발 프로세스의 외부 요인에서 파생된 모든 요구사항을 다룬다.

4.2.3.1. Regulatory Requirements

본 시스템에서는 주로 React, NestJS, MySQL 등의 외부 API가 사용되며, 이들을 포함한 모든 외부 API들의 라이선스는 철저히 검토되어야 한다. 각 API나 각 API의 라이선스 간에 충돌이 발생하는 경우는 시스템의 개발 과정에서 반드시 해결되어야 한다.

사용자에게 제공되는 코딩 테스트 등의 문제를 비롯한 모든 콘텐츠는 저작권법을 준수해야 한다.

4.2.3.2. Safety/Security Requirements

본 시스템은 개인정보보호법과 GitHub의 이용약관에 근거하여 사용자의 개인정보의 처리 및 보호를 수행한다. 본 시스템은 사용자의 개인정보 처리 과정에서 사용자의 자유와 권리를 침해하지 않아야 한다.

본 시스템은 OAuth 프로토콜을 따르며, 클라이언트에서 서버로의 정보 전송이 이루어질 때 사용자의 개인정보는 비식별화되어야 한다.

본 시스템의 사용자는 OAuth2.0 기반으로의 로그인 세션으로 구분되며, 본 시스템은 사용자 인증 수단으로 Access Token 방식을 사용하여 비밀번호 유출 등의 문제가 발생할 위험을 차단한다.

4.2.3.3. Ethical Requirements

본 시스템의 사용자는 동등한 학습 기회를 제공받아야 하며, 사용자의 테스트 내용을 평가하는 기준은 공정성이 보장되어야 한다.

본 시스템의 개발자와 관리자는 제공하는 콘텐츠가 사회적 질서나 윤리관을 위배하지 않도록 신중히 검토 절차를 수행하여야 한다.

4.3. Organizing System Flow

4.3.1. Context Model

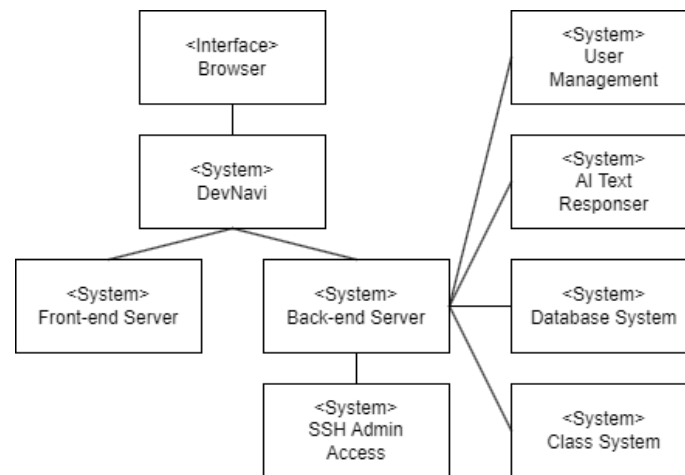


Figure 8. Context Model

4.3.2. Process Model

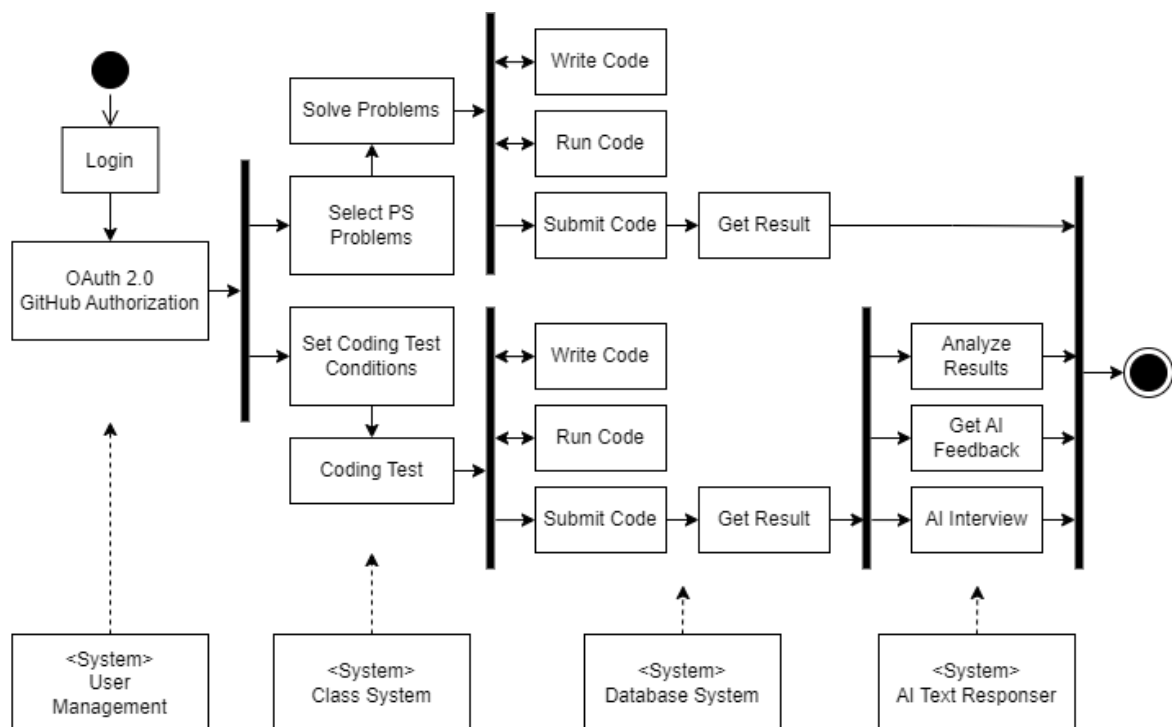


Figure 9. Process Model

4.3.3. Interaction Model

본 내용은 '4.1.2. Use Case Diagram'을 참고한다.

4.3.4. Behavior Model

4.3.4.1. Data Flow Diagram

본 내용은 '4.1.3. Data Flow Diagram'을 참고한다.

4.3.4.2. Sequence Diagram

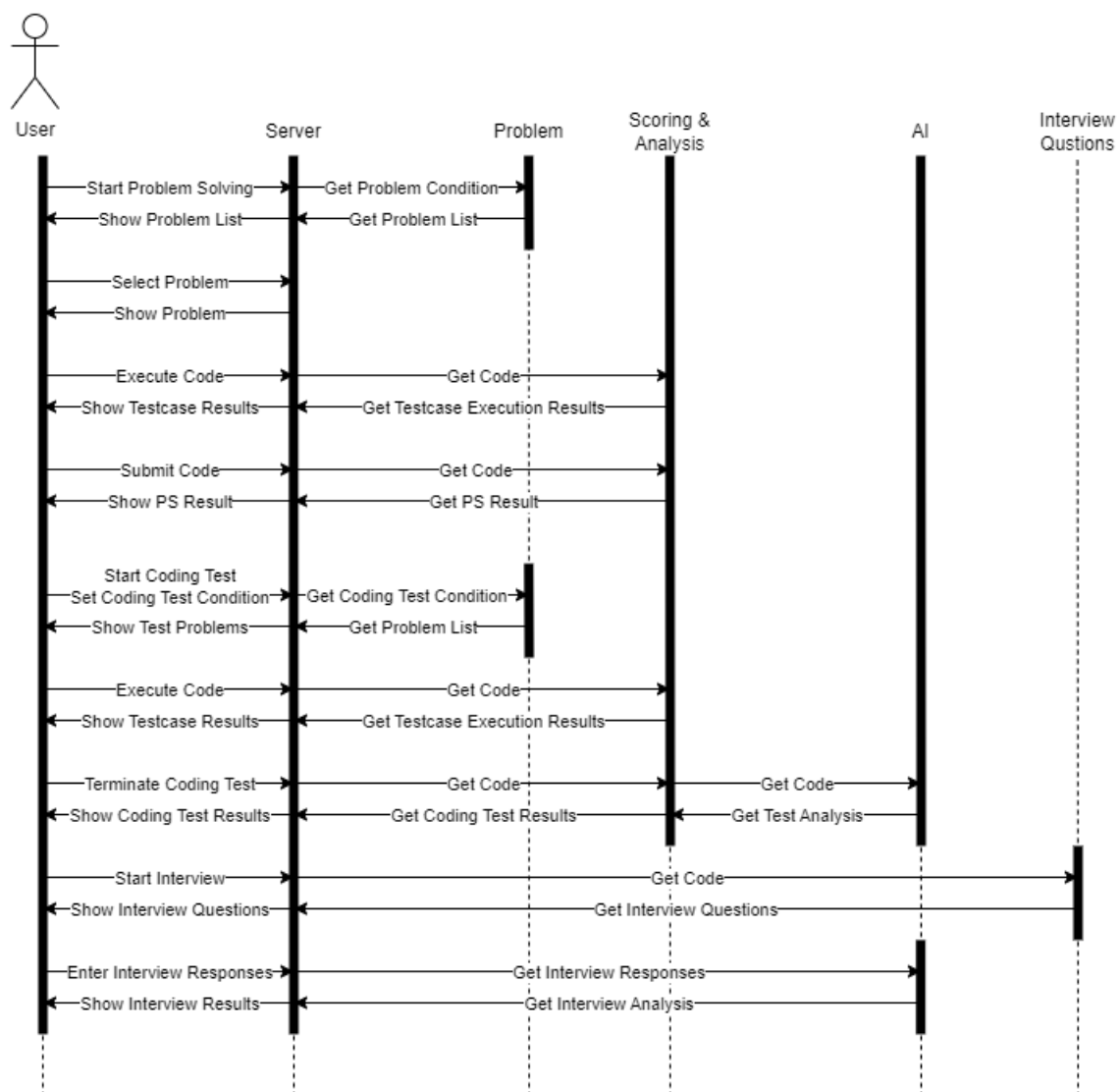


Figure 10. Sequence Diagram

4.4. System Architecture

이 섹션에서는 예상되는 시스템 아키텍처에 대한 개요를 제시하고 시스템 모듈 전반의 기능 분포를 보인다.

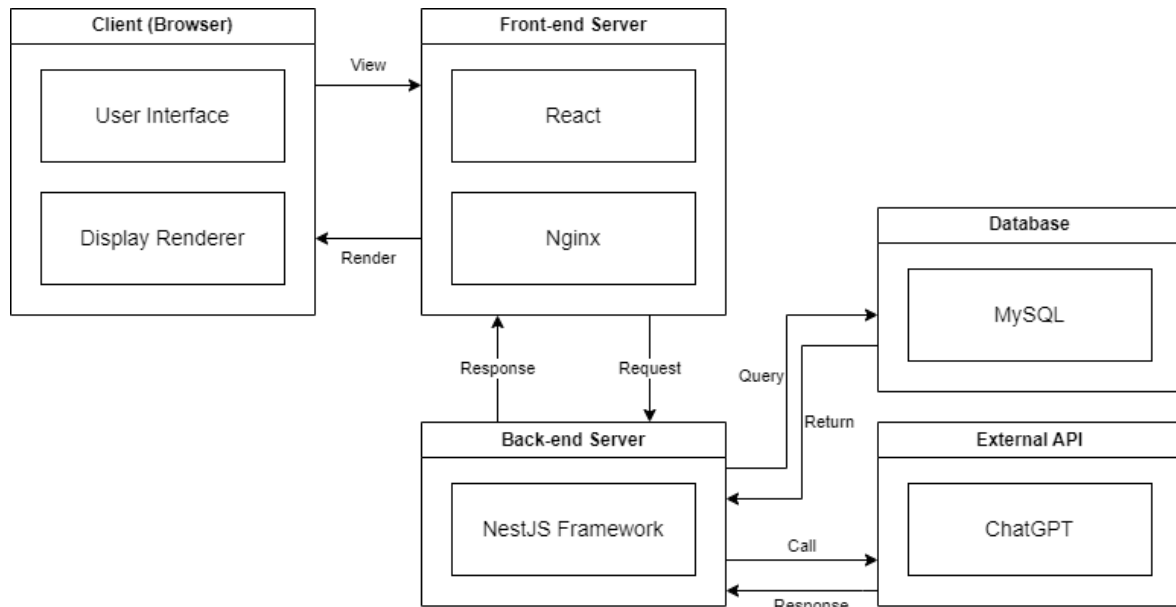


Figure 11. System Architecture

5. ADDITIONAL INFORMATIONS

5.1. Software Requirement Specification

본 문서는 IEEE 양식("IEEE Guide for Developing System Requirements Specifications")에 따라 작성되었다, 단, 본 시스템 개발 상황과 규모를 고려하여 본래의 양식에서 변형된 부분이 있다.

5.2. Document History

Table 31. Document History

Date	Version	Description	Written by
2023.04.23.	0.1	문서 초안 작성	권민성, 서주원, 안서현, 안정민, 이장엽, 한수현
2023.04.30.	1.0	최초 발행	권민성, 서주원, 안서현, 안정민, 이장엽, 한수현