



Web Shooter

Software Requirement Specification

2024.11.10.

Introduction to Software Engineering 41

TEAM 1

| | |
|-------------|-----|
| Team Leader | 이유진 |
| Team Member | 권서진 |
| Team Member | 김문수 |
| Team Member | 김현진 |
| Team Member | 박경일 |

CONTENTS

| | |
|--|-----------|
| 1. Introduction..... | 6 |
| 1.1. Purpose..... | 6 |
| 1.2. Scope..... | 6 |
| 1.3. Definitions, Acronyms, and Abbreviation | 6 |
| 1.4. References | 7 |
| 1.5. Overview | 7 |
| 2. Overall Description | 9 |
| 2.1. Product Perspective..... | 9 |
| 2.1.1. System Interfaces | 9 |
| 2.1.2. User Interfaces | 9 |
| 2.1.3. Hardware Interfaces | 9 |
| 2.1.4. Software Interfaces..... | 10 |
| 2.1.5. Memory Constraints..... | 10 |
| 2.1.6. Operations | 10 |
| 2.2. Product Functions | 11 |
| 2.2.1. 회원가입 및 로그인..... | 11 |
| 2.2.2. 실습 문제 생성 | 11 |
| 2.2.3. 실시간 실습 환경 | 11 |
| 2.2.4. 채점 및 피드백 시스템..... | 11 |
| 2.2.5. 실습 문제 등록..... | 11 |
| 2.2.6. 학습 현황 대시보드..... | 11 |
| 2.2.7. 시스템 관리자 페이지 | 12 |
| 2.3. User Classes and Characteristics | 12 |
| 2.3.1. System Administrator..... | 12 |
| 2.3.2. User | 12 |
| 2.4. Design and Implementation Constraints..... | 12 |
| 2.5. Assumptions and Dependencies | 13 |
| 3. Specific Requirements..... | 14 |
| 3.1. External Interface Requirements..... | 14 |
| 3.1.1. User Interfaces | 14 |
| 3.1.2. Hardware Interface..... | 17 |
| 3.1.3. Software Interface | 17 |
| 3.1.4. Communication Interface..... | 18 |
| 3.2. Function requirement | 19 |
| 3.2.1. Use case..... | 19 |
| 3.2.1.1. User | 19 |
| 3.2.1.2. System | 24 |
| 3.2.2. Use Case Diagram..... | 27 |
| 3.2.3. Data Dictionary | 28 |
| 3.2.4. Data Flow Diagram..... | 31 |

| | | |
|-------------|--|-----------|
| 3.3. | Performance Requirements..... | 31 |
| 3.3.1. | Static Numerical Requirements..... | 31 |
| 3.3.2. | Dynamic Numerical Requirements..... | 32 |
| 3.4. | Logical Database Requirements..... | 34 |
| 3.5. | Design Constraints | 35 |
| 3.5.1. | Physical Design Constraints..... | 35 |
| 3.5.2. | Standards Compliance..... | 35 |
| 3.6. | Software System Attributes | 36 |
| 3.6.1. | Reliability | 36 |
| 3.6.2. | Availability | 36 |
| 3.6.3. | Security | 36 |
| 3.6.4. | Maintainability | 36 |
| 3.6.5. | Portability | 37 |
| 3.7. | Organizing the Specific Requirements | 37 |
| 3.7.1. | Context Model..... | 37 |
| 3.7.2. | Process Model | 38 |
| 3.7.3. | Interaction Model | 38 |
| 3.8. | System Architecture | 38 |
| 3.9. | System Evolution | 39 |
| 3.9.1. | Limitation and Assumption..... | 39 |
| 3.9.2. | Evolution of Hardware and Change of User Requirements..... | 39 |
| 4. | <i>Appendix.....</i> | 40 |
| 4.1. | W3C HTML & CSS Standards..... | 40 |
| 4.2. | Software Requirements Specification..... | 40 |
| 4.3. | Document History | 40 |

LIST OF FIGURES

| | |
|---|----|
| [Figure 1] Use Case Diagram of System | 27 |
| [Figure 2] Use Case Diagram of Admin | 27 |
| [Figure 3] Use Case Diagram of Dashboard..... | 27 |
| [Figure 4] Entity Relationship Diagram | 30 |
| [Figure 5] Data Flow Diagram..... | 31 |
| [Figure 6] Context model..... | 37 |
| [Figure 7] Process model | 38 |
| [Figure 8] System Architecture | 38 |

LIST OF TABLES

| | |
|--|----|
| [Table 1] Table of acronyms and abbreviations | 6 |
| [Table 2] Table of terms and definitions | 6 |
| [Table 3] User Interface of input processing..... | 14 |
| [Table 4] User Interface of main page | 15 |
| [Table 5] User Interface of register | 16 |
| [Table 6] Hardware Interface of applicable device for the system | 17 |
| [Table 7] Software Interface of applicable device for the system..... | 17 |
| [Table 8] Communication Interface of applicable device for the system | 18 |
| [Table 9] Communication Interface of backend server and ChatGPT prompt | 18 |
| [Table 10] Use case of Register | 19 |
| [Table 11] Use case of Find password | 20 |
| [Table 12] Use case of Login/Logout | 21 |
| [Table 13] Use case of Submitting the problem solution..... | 22 |
| [Table 14] Use case of sharing problem..... | 22 |
| [Table 15] Use case of Q&A with the chatbot | 23 |
| [Table 16] Use case of Problem provision | 24 |
| [Table 17] Use case of User problem judgment..... | 25 |
| [Table 18] Use case of Real-time preview | 26 |
| [Table 19] Use case of API testing tool..... | 26 |
| [Table 20] User | 28 |
| [Table 21] Profile | 28 |
| [Table 22] Problem | 29 |
| [Table 23] Submission | 29 |
| [Table 24] Comment | 29 |
| [Table 25] Dashboard..... | 30 |
| [Table 26] Problem statistics..... | 30 |
| [Table 27] Document History | 40 |

1. Introduction

1.1. Purpose

이 문서는 웹 개발 학습을 희망하는 모든 사람들을 대상으로 웹 개발 실습 환경을 제공하는 프로그램의 요구사항을 명확히 하고, 추후 유지보수하는 과정에서 참고할 수 있도록 작성되었다.

1.2. Scope

WebShooter는 프론트엔드부터 백엔드까지 웹 전반에 걸친 실습환경을 제공하는 것을 목표로 하며 문제 출제, 문제 풀이 환경 제공, 채점 및 피드백 제공, 실습 문제 공유, 유저 학습 현황 관리 기능을 포함한다.

1.3. Definitions, Acronyms, and Abbreviation

[Table 1] Table of acronyms and abbreviations

| Acronyms& Abbreviations | Explanation |
|-------------------------|--|
| API | Application Programming Interface (응용 프로그램 인터페이스). |
| AWS | Amazon Web Services. WebShooter 시스템의 클라우드 인프라 제공 업체. |
| RDS | Relational Database Service. AWS에서 제공하는 관리형 데이터베이스 서비스. |
| MQ | Message Queue. 작업을 비동기적으로 처리할 때 사용되는 큐 시스템. |
| AGI | Artificial General Intelligence. 인간 수준의 일반 지능을 지닌 인공지능. |

[Table 2] Table of terms and definitions

| Terms | Definitions |
|----------------|---|
| LLM | 대규모 데이터를 통해 학습된 언어 모델로, 다양한 언어 이해와 생성 작업에 사용된다. |
| Load Balancing | 트래픽을 여러 서버에 분산하여 시스템 성능을 최적화하는 기술 |

| Terms | Definitions |
|---------------------|---|
| Sandbox Environment | 사용자가 제출한 코드를 안전하게 실행할 수 있도록 격리된 환경. |
| Hot Reloading | 코드 변경 사항을 실시간으로 반영하여 사용자에게 빠르게 업데이트 하는 기능. |
| LLM | 대규모 데이터를 통해 학습된 언어 모델로, 다양한 언어 이해와 생성 작업에 사용된다. |
| Load Balancing | 트래픽을 여러 서버에 분산하여 시스템 성능을 최적화하는 기술 |

1.4. References

- AWS Documentation : <https://docs.aws.amazon.com/>
- OpenAI API Documentation : <https://platform.openai.com/docs/overview>

1.5. Overview

이 SRS (Software Requirements Specification) 문서는 WebShooter 시스템의 요구사항을 체계적으로 정의하고 설명한다. 전체 문서는 다음과 같은 주요 섹션들로 구성되어 있다.

Section 1 : Introduction

WebShooter 프로그램의 개요와 목적을 설명한다. 이 문서의 목적은 웹 개발 학습을 희망하는 사람들에게 웹 개발 실습 환경을 제공하기 위한 WebShooter의 요구사항을 명확히 하고, 유지보수 시 참고할 수 있도록 하는 것이다. WebShooter는 프론트엔드부터 백엔드까지 실습 환경을 제공하며, 주요 기능으로는 문제 출제, 풀이 환경 제공, 채점 및 피드백, 문제 공유, 유저 학습 현황 관리 기능을 포함한다. 주요 용어와 약어를 정의하여 독자가 문서를 이해하는 데 도움이 되도록 하며, 참고 자료로 AWS와 OpenAI API 문서를 제공한다.

Section 2 : Overall Description

WebShooter의 전반적인 개요와 목표를 설명한다. WebShooter는 웹 개발 학습을 지원하며, AWS와 Vercel을 기반으로 서버, 데이터베이스, 실시간 실습 환경을 제공한다. 주요 사용자 클래스(일반 사용자와 관리자)와 그 특성을 설명하고, 시스템이 운영되기 위한 하드웨어, 소프트웨어 제약 사항을 정의한다. 또한, 사용 환경과 외부 서비스에 대한 의존성을 기술하여 시스템 성능에 영향을 줄 수 있는 요인들을 제시한다.

Section 3 : Requirements

WebShooter 시스템의 성능, 데이터베이스, 설계 제약, 품질 속성, 그리고 사용자 및 시스템 간의 상호작용을 포함한 구체적인 요구사항을 다룬다. 성능 요구사항에서는 동시 접속 사용자 수와 응답 시간을 포함한 목표를 설정하고, 데이터베이스 요구사항에서는 사용자 정보와 문제 기록을 안전하게 관리하는 방안을 설명한

다. 설계 제약은 시스템이 준수해야 할 표준과 하드웨어 요구사항을 정의하며, 소프트웨어 시스템 속성에서는 신뢰성, 보안, 유지보수성 등 시스템의 품질 요건을 제시한다.

Section 4 : Appendix

RS 문서를 이해하는 데 도움이 되는 추가 정보를 제공한다. 참고할 수 있는 자료를 포함하여, 독자가 문서의 내용을 더 잘 이해할 수 있도록 지원한다.

2. Overall Description

2.1. Product Perspective

WebShooter는 웹 개발 학습을 희망하는 모든 사람을 대상으로 하며, 프론트엔드부터 백엔드까지 웹 전반에 걸친 실습을 제공하는 것을 목표로 한다. 이 시스템은 사용자가 회원가입 및 로그인 후 자신이 원하는 유형과 난이도에 맞는 학습 문제를 생성하여 실시간 실습 환경에서 웹 개발을 단계별로 학습할 수 있도록 설계되었다. 사용자는 실습을 통해 웹 개발의 기초부터 고급 문제 해결 능력까지 점진적으로 향상시킬 수 있으며, 학습 진척도는 데이터베이스에 저장된 문제 풀이 기록을 통해 관리된다.

2.1.1. System Interfaces

백엔드 서버와 채점 서버 AWS에, 프론트 서버가 Vercel에 배포된 클러스터 구조로 운영된다. 백엔드 서버는 GPT API를 활용하여 문제 생성과 피드백 제공, AI 챗봇 응답을 처리하며, AWS의 데이터베이스 서버와 채점 서버와 통신하여 사용자 요청을 실시간으로 처리한다.

채점 서버는 AWS 내부 네트워크를 통해 백엔드 서버로부터 전달받은 코드를 샌드박스 환경에서 채점한 후, 결과를 데이터베이스에 저장한다. 데이터베이스 서버는 사용자 정보, 문제 풀이 기록, 학습 히스토리를 저장하며, 개인화된 학습 환경을 제공하기 위한 데이터 관리에 사용된다.

Vercel에 배포된 프론트 서버는 사용자와 직접 상호작용하는 인터페이스를 제공하며, 백엔드 서버와의 HTTPS 통신을 통해 사용자 요청과 실시간 학습 데이터를 처리한다. 프론트 서버와 백엔드 서버는 외부 인터넷망을 통해 연결된다.

2.1.2. User Interfaces

랩탑과 PC 사용자는 키보드와 마우스를 통해 시스템의 모든 기능에 접근할 수 있다. 웹 브라우저 인터페이스를 통해 회원가입 및 로그인, 사용자 맞춤 문제 생성, 실시간 실습 환경 이용, 채점 및 피드백 확인, 실습 문제 공유 게시판 사용, 대시보드에서 학습 이력 확인 등의 기능을 제공한다. 각 기능은 직관적으로 배치되어 있으며, 관리자는 사용자와 공유된 문제들을 통합 관리할 수 있는 인터페이스를 통해 원활하게 시스템을 운영할 수 있다.

2.1.3. Hardware Interfaces

서버 인프라는 AWS와 Vercel을 통해 구성된다. 채점 서버와 백엔드 서버는 각각 2대의 인스턴스를 사용하며, 각 인스턴스는 2 vCPU와 4GB RAM을 갖추고 있다. 데이터베이스 서버는 AWS RDS의 인스턴스를 사용하며, 동일하게 2 vCPU와 4GB RAM을 제공한다. 프론트 서버는 Vercel에 배포되어 사용자 인터페이스를 제공한다.

사용자 디바이스의 최소 사양은 1.0GHz 이상의 프로세서와 2GB 이상의 RAM을 권장한다.

2.1.4. Software Interfaces

백엔드 서버와 채점 서버는 AWS EC2 인스턴스의 Ubuntu 20.04 LTS 운영체제 환경에 배포된다. 데이터베이스 서버는 AWS RDS에서 운영되며 이미지 파일은 AWS S3에 저장된다. 문제 생성, 피드백 제공에 사용되는 LLM은 GPT API를 사용한다.

프론트 서버는 Vercel에 배포되며, 클라이언트는 Chrome, Edge, Firefox와 같은 최신 웹 브라우저 환경에서 이용하는 것을 권장한다.

2.1.5. Memory Constraints

채점 서버와 백엔드 서버는 각각 4GB RAM을 갖춘 2대의 인스턴스로 구성된다. 데이터베이스 서버는 4GB RAM을 제공하는 AWS RDS 인스턴스를 사용하며, 프론트 서버는 Vercel의 서버리스 환경에서 자동으로 메모리 확장이 이루어진다.

채점 서버 간에는 MQ(Messaging Queue)를 통해 작업을 분배하여, 병목 현상이 발생하지 않도록 관리한다. 백엔드 서버 간에는 로드 밸런싱을 적용하여 각 서버에 균등하게 트래픽을 분산시키고, 메모리 사용의 효율성을 극대화한다.

2.1.6. Operations

관리자 페이지는 시스템 내 사용자 관리와 문제 관리 기능을 제공한다. 사용자는 관리자 페이지에서 사용자 계정을 생성, 수정, 삭제할 수 있다. 또한 문제 관리 기능을 통해 등록된 문제를 수정하거나 부적절한 댓글과 리뷰를 처리하여, 시스템의 적절한 운영과 문제 데이터의 품질을 유지한다.

서버 관리 작업에는 채점 서버와 백엔드 서버의 상태 모니터링과 데이터베이스 유지보수가 포함된다. 서버 관리는 AWS CloudWatch를 통해 CPU와 메모리 사용량, 서버 로그를 실시간으로 확인하여 시스템이 정상 작동하는지 점검하고, 이상이 발생할 경우 즉각 조치한다. 데이터베이스는 주기적인 백업과 복구, 최적화를 통해 데이터 안전성과 성능을 보장하며, 필요시 불필요한 데이터를 정리하여 효율성을 유지한다.

LLM 신뢰성 관리에는 Prompt Generator를 사용하여 문제 생성 프롬프트의 품질을 지속적으로 점검하고, 개선이 필요한 경우 즉각 수정한다. 이를 통해 문제 생성 프롬프트의 신뢰도를 유지하고 플랫폼의 품질을 높이는 절차를 마련한다.

2.2. Product Functions

2.2.1. 회원가입 및 로그인

사용자가 개인 계정을 생성하고 로그인하여 자신만의 학습 기록을 관리할 수 있도록 한다. 회원가입과 로그인은 이메일과 비밀번호를 통해 이루어진다. 비밀번호 찾기 및 재설정 기능을 제공하며, 로그인 상태에서 사용자는 자신이 생성한 문제와 자신의 활동을 확인할 수 있다.

2.2.2. 실습 문제 생성

사용자가 관심 있는 주제와 난이도를 선택하면, LLM을 통해 그에 맞는 문제를 자동으로 생성한다. 사용자의 주제 선택과 난이도 설정을 반영하여 맞춤형 실습 문제를 제공하며, 이를 통해 사용자가 원하는 학습 목표에 집중할 수 있도록 한다.

2.2.3. 실시간 실습 환경

사용자가 실시간으로 코드를 작성하고 결과를 확인할 수 있는 웹 개발 실습 환경을 제공한다. 실습 환경에는 코드 에디터와 스켈레톤 코드가 포함된다. 프론트엔드 실습의 경우 작성한 코드를 핫 리로딩으로 미리볼 수 있으며, 백엔드 실습의 경우 API 테스트 도구를 통해 API 요청과 응답을 확인할 수 있다.

각 문제에 대한 설명과 요구사항 안내를 제공하여 사용자가 명확하게 목표를 이해하고 접근할 수 있도록 한다. 또한, AI 챗봇과 Q&A 형식을 통해 키워드에 대한 설명을 받거나 문제 해결에 필요한 힌트를 얻을 수 있다.

2.2.4. 채점 및 피드백 시스템

사용자가 제출한 코드에 대해 자동으로 채점하고, 채점 결과를 바탕으로 피드백을 제공한다. 채점 기준으로 유닛 테스트 코드 실행 여부와 스크린샷 유사도 측정을 사용하여 코드의 기능과 UI를 검증한다. 이러한 결과를 바탕으로 코드의 개선점이나 최적화 방안을 피드백으로 제공한다.

2.2.5. 실습 문제 등록

사용자가 문제를 해결하면 이를 등록하여 다른 사용자들이 동일한 문제를 풀어볼 수 있다. 다른 사용자가 문제를 해결했을 경우, 댓글을 통해 의견을 남기거나 다른 사람들이 남긴 댓글을 확인할 수 있으며, 서로의 학습 경험을 공유할 수 있다. 또한, 다른 사용자의 제출 내역을 확인하여 각 코드 라인별로 리뷰를 제공할 수 있어 보다 심층적인 피드백과 논의가 가능하다.

2.2.6. 학습 현황 대시보드

사용자가 해결한 문제들을 한눈에 확인하고 학습 성과를 파악할 수 있도록 대시보드를 제공한다. 이 대시보드는 사용자가 푼 문제의 난이도와 유형 정보를 그래

프로 시각화하여, 학습 패턴을 직관적으로 확인할 수 있게 한다

2.2.7. 시스템 관리자 페이지

플랫폼의 운영과 사용자 관리를 위한 시스템 관리자 페이지를 제공한다. 관리자는 사용자 정보를 조회하고 계정을 관리하며, 게시판에 공유된 문제를 확인하고 필요시 수정 또는 삭제를 통해 문제 콘텐츠를 관리한다. 또한, 사용자가 남긴 피드백과 게시판 내용을 모니터링하여 적절히 대응하고, 사용자 수, 문제 풀이 통계, 학습 참여율 등을 시각화하여 플랫폼 운영 현황을 파악할 수 있다.

2.3. User Classes and Characteristics

2.3.1. System Administrator

시스템 관리자는 WebShooter 플랫폼의 운영을 책임지며, 사용자 계정 관리와 문제 및 댓글 관리에 초점을 둔다. 관리자 페이지를 통해 사용자 계정을 생성, 수정, 삭제할 수 있다. 또한, 게시판에 공유된 문제의 수정과 부적절한 댓글이나 리뷰를 처리하는 역할을 수행한다. 이를 통해 학습 환경이 적절히 유지되도록 관리하며, 사용자가 게시판에서 상호작용할 때 규정된 기준을 준수할 수 있도록 돕는다.

2.3.2. User

클라이언트 사용자는 웹 개발 학습을 희망하는 학생, 초보 웹 개발자, 일반 사용자로 구성된다. 이들은 HTML, CSS, JavaScript와 같은 웹 프로그래밍의 기본 언어를 이해하지만, 전문적인 개발자가 아닐 수 있다. 주로 Chrome, Edge, Firefox 등의 최신 웹 브라우저를 사용하여 데스크톱이나 노트북에서 WebShooter 플랫폼에 접속하며, 플랫폼 내 다양한 학습 기능을 활용한다. 사용자는 실시간으로 문제를 풀고 코드 작성 및 실행을 하며, AI 챗봇을 통해 질문하고 답변을 얻으며, 다른 사용자와 코드 및 피드백을 주고받는다.

또한, 이들은 학습 진척 상황을 대시보드를 통해 시각적으로 확인하며, 필요에 따라 관리자에게 문의하거나 피드백을 요청할 수 있다. 클라이언트 사용자는 직관적인 인터페이스를 통해 학습 중 발생하는 문제를 쉽게 해결하고, 문제 풀이에 대한 피드백을 즉시 확인하며, 다른 사용자와 상호작용하면서 학습 경험을 강화하길 기대한다.

2.4. Design and Implementation Constraints

WebShooter 플랫폼은 다음과 같은 제약을 고려하여 설계 및 개발된다.

- Chrome, Edge, Firefox의 최신 버전을 지원하도록 설계되며, 구형 브라우저는 지원하지 않는다.

- 사용자는 JavaScript가 활성화된 환경에서만 플랫폼에 접근할 수 있으며, 최소 2GB RAM과 1.0GHz 이상의 프로세서를 갖춘 데스크톱 또는 노트북을 사용하는 것을 권장한다.
- 널리 사용되고 성능이 입증된 기술과 프레임워크를 사용하며, 최신 버전의 안정적인 소프트웨어만을 채택한다.
- 시스템 개발 및 유지 관리 시 비용을 고려하여, 가능한 한 오픈 소스 소프트웨어를 사용하고 라이선스 비용이나 저작권료가 필요한 기술은 피한다. 단, 시스템의 필수 기능을 구현하기 위해 유일한 선택지가 있을 경우에는 이를 예외로 둔다.
- 사용자가 증가함에 따라 서버 자원을 확장할 수 있도록 설계된다.
- 소프트웨어 개발 시 추후 유지보수와 확장 가능성을 고려하여 충분한 주석을 추가한다.

2.5. Assumptions and Dependencies

사용자는 Chrome, Edge, Firefox 등 최신 웹 브라우저에서 접속한다고 가정한다. 구형 브라우저나 비표준 브라우저는 지원하지 않는다.

서버 인프라와 데이터베이스를 AWS에서 운영하며, 프론트엔드는 Vercel에 배포된다. AWS와 Vercel 서비스의 유지보수, 성능, 보안 정책에 따라 시스템의 성능과 가용성이 영향을 받을 수 있다.

3. Specific Requirements

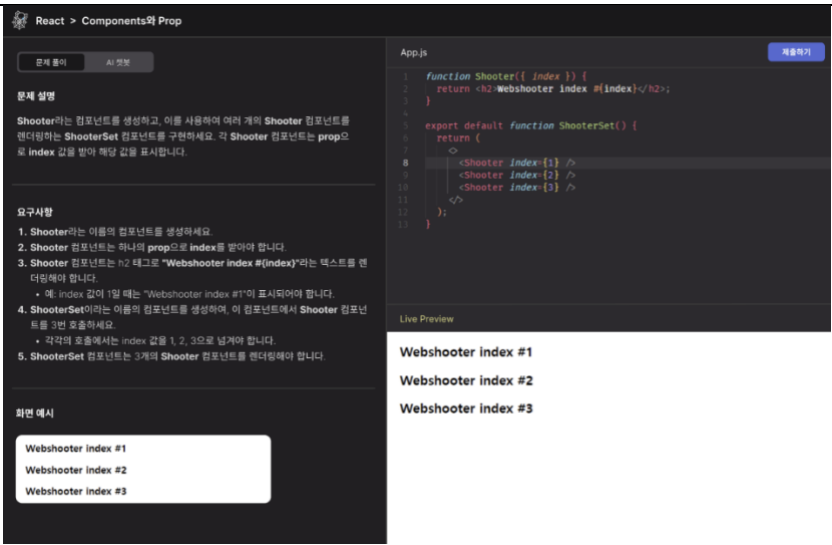
3.1. External Interface Requirements

3.1.1. User Interfaces

[Table 3] User Interface of input processing

| 이름 | 마우스 및 키보드를 통한 입력 처리 |
|-------------------|--|
| 목적/내용 | 목적/내용 : 시스템 사용자가 키보드 및 마우스의 입력을 통해 실습 환경 및 UI 내 기능(e.g., 코드 실행, 파일 저장 등)을 제어하고, 코드를 입력하거나 실행 등을 수행할 수 있도록 함. |
| 입력 주체/입력 소스/출력 대상 | 입력 주체 : 사용자 입력 소스 : 키보드(코드 입력, 단축키 활용, 텍스트 및 명령 입력), 마우스(메뉴 선택, 드래그 앤 드롭, 실행 버튼 등 UI 요소 클릭) 출력 대상 : 웹 편집기 및 관련 UI 화면에 입력 반영 |
| 유효 범위/정확도/허용 오차 | 유효 범위 : 화면에서의 버튼 개수에 따른 입력 범위 정확도 : 모든 키 입력 정확히 반영, 시스템 단축키와의 충돌 최소화, 지정된 단축키 동작에 대해 100% 정확도 보장, 클릭, 더블 클릭, 드래그 앤 드롭 동작의 정확한 인식. 허용 오차 : 드래그 시 위치 오차 5px 이내 허용, 입력 지연 최대 100ms까지 허용 |
| 측정 단위 | ms (반응 시간), px (해상도) |
| 시간/속도 | 모든 키보드/마우스 입력이 발생한 순간에 즉시 UI에 반영되도록 처리 유지(100ms 이내), 동시실행되는 다른 작업(e.g., 코드 컴파일)보다 우선적인 반응성 유지 |
| 타 입출력과의 관계 | 키보드/마우스 입력 내용에 따라 클라이언트에서 처리하여 편집 상태를 실시간으로 반영하거나, 서버에서 처리하도록 명령 요청 |
| 화면 형식 및 구성 | - |
| 윈도우 형식 및 구성 | - |
| 데이터 형식 및 구성 | JavaScript / Python / Java 코드, Text / Int형의 코드 값 |
| 명령 형식 | 각 코드에 따른 명령 매핑 |
| 종료 메시지 | - |

[Table 4] User Interface of main page

| 이름 | 모니터를 통한 메인 화면 출력 |
|-----------------|---|
| 목적/내용 | 목적/내용 : 시스템 사용자가 웹프로그래밍 실습 환경의 UI를 시각적으로 확인하고, 코드 편집, 실행 결과, 로그, 설정 등의 상황을 직관적으로 확인할 수 있도록 함. |
| 입력 소스/출력 대상 | 시스템 내부 데이터, 서버 데이터 / 시스템 사용자 모니터 |
| 유효 범위/정확도/허용 오차 | 유효 범위 : 화면에서의 버튼 개수에 따른 입력 범위 정확도 : 최소 1920x1080 해상도에서 UI가 깨짐없이 표시 보장, 해상도별 반응형 디자인 적용, 다양한 해상도에 맞게 최적화, 해상도 별로 폰트 크기 및 UI 구성 요소의 위치의 일관성을 유지 허용 오차 : 레이아웃 요소 간의 간격 오차 5px 이내, |
| 측정 단위 | ms (반응 시간), px (해상도) |
| 시간/속도 | 사용자 입력 및 데이터 변경 사항이 발생한 후, 화면에 즉각적인 피드백 반영(100ms 이내), 실시간 데이터가 아닌 경우 1초마다 주기적인 갱신 |
| 타 입출력과의 관계 | 키보드와 마우스 입력은 메인 화면에서의 출력에 즉시 반영되며, 실습 환경에서의 타이핑, 실행 요청 등에서 시각적 피드백 제공, 서버와의 데이터 동기화가 필요한 경우 자동으로 UI 반영되도록 함, 사용자가 실행한 작업의 상태나 오류 메시지는 화면 상단 또는 하단에 실시간으로 출력하여 사용자 피드백 제공 |
| 화면 형식 및 구성 |  <p>좌측에는 문제 설명 및 문제 요구사항, 화면 예시를 제시하고 우측에는 IDE 창에 작성한 코드에 따른 Live Preview를 제공함. (React-Runner 등 이용)</p> |

| | |
|-------------|--|
| 윈도우 형식 및 구성 | Linear Layout 형식으로 좌측엔 문제 설명 / 요구사항 / 화면 예시, 우측엔 IDE 창 / Live Preview 로 구성 |
| 데이터 형식 및 구성 | 이미지, 텍스트 |
| 명령 형식 | - |
| 종료 메시지 | - |

[Table 5] User Interface of register

| 이름 | 사용자 등록 인터페이스 |
|-----------------|--|
| 목적/내용 | 시스템에 사용자의 정보를 등록하여 오류 없이 사용자 확인이 가능하도록 하고 서비스 개인정보를 사용자 별로 저장, 관리함 |
| 입력 주체/출력 대상 | 사용자 / 서버 |
| 유효 범위/정확도/허용 오차 | - |
| 측정 단위 | - |
| 시간/속도 | - |
| 타 입출력과의 관계 | 이메일, 비밀번호 등 각 필드의 입력값에 대해 유효성 검사를 하며, 형식에 오류가 있을 경우 해당 필드 옆에 오류 메시지가 출력됨 |
| 화면 형식 및 구성 | 새 계정에 대한 정보 입력을 위한 슬롯 및 가입 버튼 |
| 윈도우 형식 및 구성 | - |
| 데이터 형식 및 구성 | Query |
| 명령 형식 | 등록 버튼에 따른 명령 매핑 |
| 종료 메시지 | “가입이 완료되었습니다.” |

3.1.2. Hardware Interface

[Table 6] Hardware Interface of applicable device for the system

| 이름 | 시스템에서 사용 가능한 디바이스 |
|-----------------|--------------------------|
| 목적/내용 | 키보드, 마우스를 사용한 사용자의 입력 |
| 입력 주체/출력 대상 | 사용자 / 서버 |
| 유효 범위/정확도/허용 오차 | - |
| 측정 단위 | Mbps (전송 속도), ms (지연 시간) |
| 시간/속도 | 사용자의 입력 / 문제 풀이에 대한 처리 |
| 타 입출력과의 관계 | 키보드/마우스 입력은 실시간으로 반응 |
| 화면 형식 및 구성 | - |
| 윈도우 형식 및 구성 | - |
| 데이터 형식 및 구성 | - |
| 명령 형식 | 코드 |
| 종료 메시지 | - |

3.1.3. Software Interface

[Table 7] Software Interface of applicable device for the system

| 이름 | 시스템 애플리케이션 |
|-----------------|-------------------------------|
| 목적/내용 | 사용자의 입력 및 서버에서 보낸 데이터들의 화면 출력 |
| 입력 주체/출력 대상 | - |
| 유효 범위/정확도/허용 오차 | - |
| 측정 단위 | ms (반응 시간) |
| 시간/속도 | 실시간 또는 사용자 요청 시 실행 |
| 타 입출력과의 관계 | - |
| 화면 형식 및 구성 | 애플리케이션에서 출력 |
| 윈도우 형식 및 구성 | - |
| 데이터 형식 및 구성 | - |
| 명령 형식 | - |
| 종료 메시지 | - |

3.1.4. Communication Interface

[Table 8] Communication Interface of applicable device for the system

| 이름 | 호스트 서버-클라이언트 |
|-----------------|--|
| 목적/내용 | 각 클라이언트에서 호스트 서버에 접속을 요청하고, 사용자가 입력한 코드와 그 출력화면을 호스트 서버에서 각 클라이언트에게 전달받고 정답 확인을 하여 결과 전달 |
| 입력 주체/출력 대상 | 클라이언트 / 호스트 서버 |
| 유효 범위/정확도/허용 오차 | - |
| 측정 단위 | Mbps (전송 속도), 패킷 (데이터 전송 단위) |
| 시간/속도 | 최소 10Mbps 이상 |
| 타 입출력과의 관계 | - |
| 화면 형식 및 구성 | - |
| 윈도우 형식 및 구성 | - |
| 데이터 형식 및 구성 | Struct를 이용한 명령 코드 및 계정 로그인 데이터 |
| 명령 형식 | Send() 콜에 의한 통신 |
| 종료 메시지 | Close() 콜에 의한 소켓 종료 |

[Table 9] Communication Interface of backend server and ChatGPT prompt

| 이름 | 백엔드 서버-GPT |
|-----------------|--|
| 목적/내용 | ChatGPT 프롬프트를 통해 자동으로 문제를 생성하고, 생성된 문제를 백엔드 서버로 전송하여 데이터베이스에서 관리할 수 있도록 함 |
| 입력 주체/출력 대상 | 백엔드 서버(원 요청자는 사용자) / ChatGPT 프롬프트 |
| 유효 범위/정확도/허용 오차 | 정확도 : 생성된 문제는 JSON 형식으로 백엔드에 전달되며, 문제 내용, 선택지, 정답, 난이도 등의 필수 요소가 포함되어야 함, 문제 데이터의 필드 구조를 미리 정의하여 필드가 누락되지 않도록 함 |
| 측정 단위 | Mbps (전송 속도), 패킷 (데이터 전송 단위) |
| 시간/속도 | 백엔드 서버로 문제 데이터를 전송하는 과정이 500ms 이내의 응답 속도로 이뤄질수 있도록 함. |
| 타 입출력과의 관계 | 프롬프트가 ChatGPT에 전달되어 문제 생성 요청을 수행하며, 생성된 문제는 프론트엔드에서 구조를 검증한 후 백엔드에 전송, 백엔드 서버는 문제 데이터가 유효성을 통과한 후 데이터베이스에 저장하며, 이후 검색 및 수정 요청이 가능하도록 관리, 문제 생성 실패 시 프론 |

| | |
|-------------|---|
| | 트엔드에 에러 메시지를 반환하고 재시도 옵션을 제공 |
| 화면 형식 및 구성 | - |
| 윈도우 형식 및 구성 | - |
| 데이터 형식 및 구성 | JSON 형식 |
| 명령 형식 | JSON 형식으로 command, parameters 형태로 작업 명령 |
| 종료 메시지 | JSON 형식으로 status, message, result 형태의 응답으로 세션 종료 |

3.2. Function requirement

3.2.1. Use case

3.2.1.1. User

[Table 10] Use case of Register

| Use case name | Register |
|---------------|--|
| Actor | 미등록 사용자 |
| Description | 미등록 사용자가 시스템의 모든 기능을 이용하기 위해서 회원 등록을 시도하는 과정이다. |
| Normal Course | <ol style="list-style-type: none"> 모든 사용자는 웹사이트 접속 후 오른쪽 상단에 있는 로그인, 회원가입 창에 접근할 수 있다. 웹사이트의 메인에서 등록 후의 사용자에게만 서비스가 제공되는 것을 확인한 후, 미등록 사용자는 회원가입 창에 접근한다. 사용자는 등록 양식에 따라 추가 정보를 입력해야한다. 필요한 정보는 다음을 포함한다. <ol style="list-style-type: none"> ID(이메일주소) Username Password 관심언어 관심분야 시스템은 이메일 주소가 올바른지 확인하고, 지정된 전자 메일 주소로 확인 코드를 보낸다. 작성된 양식대로 사용자가 등록되고, 양식 끝에 있는 회원 가입 버튼을 클릭하면 등록이 되며, 해당 정보로 로그인되어 메인페이지로 리다이렉트된다. |

| Use case name | Register |
|----------------|--|
| Precondition | <ul style="list-style-type: none"> • 사용자가 시스템에 등록되기 이전이어야한다. • 사용자가 올바른 정보를 입력해야한다. <ul style="list-style-type: none"> ○ 이메일의 형식을 따라야한다. aaa@abc.com ○ 확인 코드를 체크할 수 있는 유효한 이메일이어야한다. ○ 다른 사용자의 이메일 주소와 중복되지 않아야한다. ○ 다른 사용자의 Username과 중복되지 않아야한다. • 잘못된 입력이 있는 경우, 시스템은 이메일 주소 및 비밀번호의 형식을 확인할 수 있도록 한다 <ul style="list-style-type: none"> ○ 어떠한 형식을 지켜야하는지 안내한다. |
| Post Condition | 보안을 위해 비밀번호를 암호화하고, 데이터베이스에 저장한다. |
| Assumptions | N/A |

[Table 11] Use case of Find password

| Use case name | Find password |
|----------------|---|
| Actor | 등록한 사용자 |
| Description | 등록된 사용자가 암호를 잊어버린 경우 암호를 찾고 재설정하는 과정이다. |
| Normal Course | <ol style="list-style-type: none"> 1. 사용자는 로그인 페이지에서 비밀번호 찾기 창에 접근할 수 있다. 2. 자신의 이메일을 입력한 후 비밀번호 찾기 창을 누르면, 이메일로 인증 코드를 전송받을 수 있다. 3. 이메일에서 인증번호를 확인한 후, 원래 비밀번호를 찾던 창으로 되돌아가 해당 인증번호를 입력한다. 4. 인증번호가 일치하면 새 비밀번호를 바로 설정할 수 있다. 5. 새 비밀번호 설정이 완료되면, 로그인창으로 이동한다. |
| Precondition | <ul style="list-style-type: none"> • 데이터베이스에 해당 사용자 정보가 존재해야한다. • 해당 사용자 정보가 존재한다. |
| Post Condition | 데이터베이스에 해당 사용자 정보에 맞게 새로운 비밀번호를 업데이트한다. |
| Assumptions | N/A |

[Table 12] Use case of Login/Logout

| Use case name | Login/Logout |
|----------------|---|
| Actor | 등록한 사용자 |
| Description | <ul style="list-style-type: none"> 로그인은 시스템에 등록된 사용자가 서비스를 사용하기 위해 시스템에 들어가려는 프로세스이다. 로그아웃은 로그인한 사용자가 시스템에서 나가려고 하는 프로세스이다 |
| Normal Course | <ul style="list-style-type: none"> 로그인 <ul style="list-style-type: none"> 시스템에 이미 회원으로 등록된 사용자가 시스템에서 서비스를 이용하려한다. 로그인을 위해 등록된 이메일과 암호를 입력한다. 입력된 정보에 따라 다른 안내메시지를 받는다. <ul style="list-style-type: none"> 정보가 올바를 때 <ul style="list-style-type: none"> 시스템에 접속이 가능하며, 시스템에서 제공하는 서비스를 이용할 수 있다. 정보가 올바르지 않을 때 <ul style="list-style-type: none"> 시스템에 접속이 불가능하며, ID와 암호를 확인하라는 안내 문구를 확인한다. 등록된 정보가 없을 때 <ul style="list-style-type: none"> 해당 정보와 일치하는 정보가 없음을 안내하는 문구를 확인한다. 로그아웃 <ul style="list-style-type: none"> 시스템을 나가기 위해 Logout 버튼을 사용한다. <ul style="list-style-type: none"> 사용자가 Logout하지않고 해당 웹 창을 닫은 경우에도 세션이 유지되며 해당 웹 사이트를 다시 방문했을 때 로그인 상태를 유지하도록한다. |
| Precondition | <ul style="list-style-type: none"> 로그인 : USER 데이터베이스에 해당 사용자 정보가 존재해야한다. 로그아웃 : 사용자가 로그인 상태여야한다. <ul style="list-style-type: none"> 세션 유지를 위한 기술이 필요하다. |
| Post Condition | 사용자가 온라인 상태여야한다. |
| Assumptions | N/A |

[Table 13] Use case of Submitting the problem solution

| Use case name | Submitting the problem solution |
|----------------|--|
| Actor | 등록된 사용자 |
| Description | 사용자가 특정 문제에 대한 코드를 작성하여 제출하는 과정 |
| Normal Course | <ol style="list-style-type: none"> 1. 사용자는 생성된 문제를 보고 문제 풀이 버튼을 통해 코드 작성 화면에 접근한다. 2. 문제에 대한 코드를 작성하고 제출 버튼을 클릭한다. 3. 시스템은 사용자의 코드를 Judge 서버로 전송하여 채점 프로세스를 시작한다. 4. 채점 결과가 나오면, 사용자에게 풀이 성공 여부 또는 오류 메시지를 제공한다. |
| Precondition | <ul style="list-style-type: none"> • 사용자는 문제 풀이 권한이 있어야 한다. • 문제에 대한 정답이 명확해야 한다. |
| Post Condition | <ul style="list-style-type: none"> • 사용자의 제출 코드와 채점 결과를 데이터베이스에 저장한다. <ul style="list-style-type: none"> ◦ 제출된 코드 중 성공으로 판단된 코드는 실습 문제 게시판에 공유할 수 있다. |
| Assumptions | N/A |

[Table 14] Use case of Sharing problem

| Use case name | Sharing problem |
|---------------|--|
| Actor | 등록된 사용자 |
| Description | 사용자가 특정 문제에 대한 정답 풀이를 다른 사용자와 공유하는 과정이다. |
| Normal Course | <ol style="list-style-type: none"> 1. 사용자는 자신이 제출한 문제 풀이 중 성공으로 판단된 풀이를 공유하기 위해 실습 문제 등록 버튼을 클릭한다. 2. 실습 문제 게시판에서는 여러 유저들이 공유한 문제 및 코드를 확인할 수 있다. 3. 실습 문제 게시판에 공유된 코드에는 line by line 코멘트 작성이 가능하다. |
| Precondition | <ul style="list-style-type: none"> • 실습 문제 게시판에 게시된 문제에 대한 정보, 난이도, 키워드 등을 |

| Use case name | Sharing problem |
|----------------|---|
| | <p>게시판에서 확인할 수 있다.</p> <ul style="list-style-type: none"> 공유할 수 있는 코드는 정답으로 인정된 코드로 제한된다 |
| Post Condition | 실습 문제 게시판에서 공유된 코드는 다른 유저들이 열람 및 풀이가 가능하다. |
| Assumptions | N/A |

[Table 15] Use case of Q&A with the chatbot

| Use case name | Q&A with the chatbot |
|----------------|--|
| Actor | 사용자 |
| Description | 사용자가 챗봇을 통해 문제 풀이 관련 질의응답을 요청하거나 키워드 기반 추천 질문, LLM 개념 설명을 받을 수 있는 과정이다. |
| Normal Course | <ol style="list-style-type: none"> 1. 사용자는 문제풀이환경에 제공되어있는 챗봇 화면에 접근하여 질문을 입력한다. 2. 챗봇은 키워드 기반의 추천 질문을 제공하거나, 관련 개념 설명을 제공한다. 3. 사용자는 챗봇의 답변을 통해 추가적인 정보를 얻는다. |
| Precondition | <ul style="list-style-type: none"> 챗봇은 LLM을 통해 관련 설명을 한다 챗봇이 제공한 답변에는 출처가 함께 표시된다 |
| Post Condition | 사용자는 필요한 정보를 얻거나, 추천 질문을 확인할 수 있다. |
| Assumptions | N/A |

3.2.1.2. System

[Table 16] Use case of Problem provision

| Use case name | Problem provision |
|----------------|---|
| Actor | 시스템(AI) |
| Description | 시스템이 LLM 을 사용해 문제를 사용자한테 제공하는 과정이다. |
| Normal Course | <ol style="list-style-type: none"> 1. 시스템은 사용자로부터 난이도와 유형을 입력받아 LLM 을 사용하여 문제를 생성한다. 2. 사용자는 생성된 문제를 보고 문제를 풀 것인지 결정할 수 있다. |
| Precondition | <ul style="list-style-type: none"> • LLM이 제공하는 문제 형식은 다음과 같은 항목을 포함해야한다. <ul style="list-style-type: none"> ◦ 문제 제목, 문제 설명, 요구사항, 화면예시, 채점 기준, 난이도, 키워드 • 문제에 대한 제출 내역이 없는 상태에서 뒤로 가거나 창을 닫으면 해당 문제는 저장하지 않는다. |
| Post Condition | <ul style="list-style-type: none"> • 제출내역이 있는 경우 <ul style="list-style-type: none"> ◦ 해당 문제를 데이터베이스에 저장한다. • 제출내역이 없는 경우 <ul style="list-style-type: none"> ◦ 뒤로가거나, 창을 닫으면 해당 문제를 저장하지 않는다는 경고 문구를 제공한다. |
| Assumptions | N/A |

[Table 17] Use case of User problem judgment

| Use case name | User problem judgment |
|----------------|--|
| Actor | 시스템(Judge Server) |
| Description | 시스템이 사용자가 제출한 문제 풀이 코드를 Judge 서버로 전송하여 채점하는 과정이다. |
| Normal Course | <ol style="list-style-type: none"> 1. 시스템은 사용자가 제출한 문제 풀이 코드를 Judge server로 전송한다. 2. Judge server는 문제에 대한 테스트케이스와 비교하여 채점한다. 3. 채점결과가 시스템으로 돌아오면, 이를 사용자에게 제공 후 데이터베이스에 저장한다. |
| Precondition | Judge server가 정상적으로 작동하여 채점이 가능해야한다. |
| Post Condition | <ul style="list-style-type: none"> • 정답시 <ul style="list-style-type: none"> ○ 제출결과는 submission db에 저장되고, 사용자에게 정답인지를 알려준다. ○ 해당 문제에 정답 여부를 저장한다. • 오답시 <ul style="list-style-type: none"> ○ 제출결과는 submission db에 저장되고, 해당 오답에 대한 통과되지 못한 테스트케이스, LLM을 통한 피드백을 간단하게 알려준다. ○ 해당 문제에 오답 여부를 저장한다. |
| Assumptions | N/A |

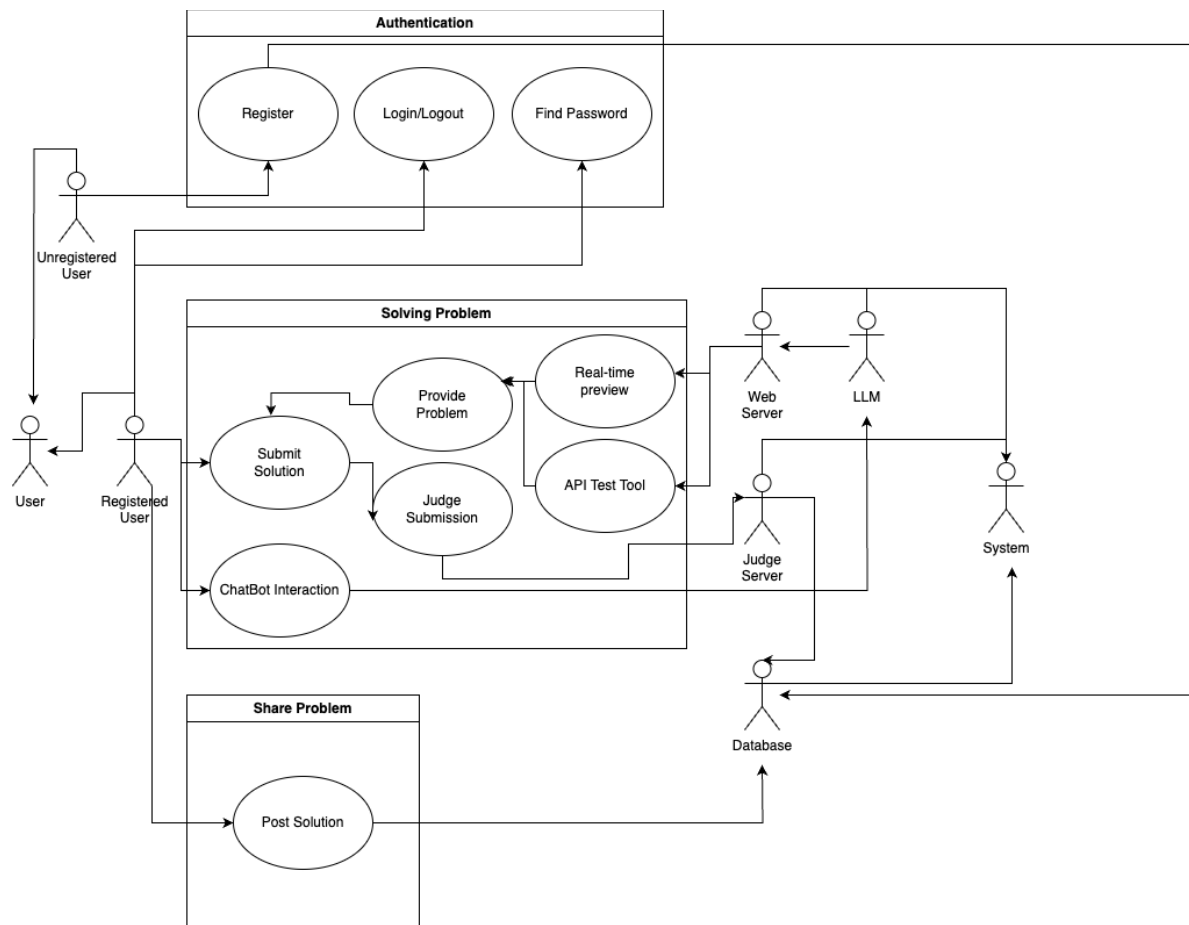
[Table 18] Use case of Real-time preview

| Use case name | Real-time preview |
|----------------|---|
| Actor | 시스템(Web) |
| Description | 사용자가 문제를 풀 때, 해당 문제 코드에 대한 결과를 실시간 미리보기로 제공하는 과정이다. (프론트) |
| Normal Course | <ol style="list-style-type: none"> 1. 사용자가 코드를 작성하는 동안 시스템이 입력된 내용을 미리보기 창에 반영한다. 2. 사용자는 미리보기를 통해 실시간으로 코드 작동을 예측할 수 있다. |
| Precondition | 실시간 미리보기 기능이 활성화되어있어야한다 |
| Post Condition | 코드를 수정하는 즉시 미리보기가 업데이트되어야한다 |
| Assumptions | N/A |

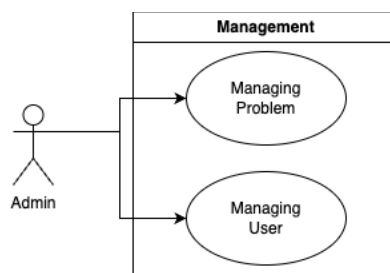
[Table 19] Use case of API testing tool

| Use case name | API testing tool |
|----------------|---|
| Actor | 시스템(Web) |
| Description | 사용자가 문제를 풀 때, 해당 코드를 실행해 결과를 확인하는 과정이다. (백엔드) |
| Normal Course | <ol style="list-style-type: none"> 1. 사용자가 백엔드 코드를 작성한다 2. 해당 백엔드 코드의 API를 테스트해볼 수 있다. 3. 사용자는 실행된 결과를 보고, 코드를 수정할 수 있다. |
| Precondition | <ul style="list-style-type: none"> • API 테스트를 제공할 수 있어야한다. • 입력, 출력이 존재해야하고, 이를 제어하는 함수 로직이 필수로 존재해야한다. |
| Post Condition | 해당 결과를 콘솔에서 확인할 수 있다. (프론트의 실시간 미리보기 창과 같은 위치) |
| Assumptions | N/A |

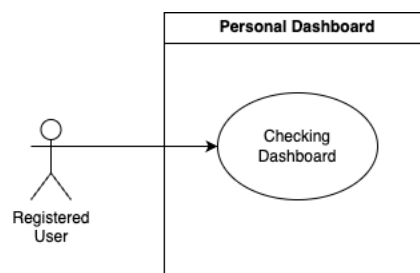
3.2.2. Use Case Diagram



[Figure 1] Use Case Diagram of System



[Figure 2] Use Case Diagram of Admin



[Figure 3] Use Case Diagram of Dashboard

3.2.3. Data Dictionary

[Table 20] User

| User | | | |
|-------------------|-----|------------------|--------------|
| Field | Key | Constraint | Description |
| user_id | PK | Not Null, Unique | 사용자 아이디 |
| email | | Not Null, Unique | 사용자 아이디(이메일) |
| username | | Not Null, Unique | 사용자 이름 |
| password | | Not Null | 사용자 암호 |
| interest_language | | Not Null | 사용자 관심 언어 |
| interest_field | | Not Null | 사용자 관심 분야 |
| role | | | 일반/ 관리자 구분 |
| profile_id | FK | | 프로필 사진 |

[Table 21] Profile

| Profile | | | |
|--------------|-----|------------|-------------|
| Field | Key | Constraint | Description |
| profile_id | PK | Not Null | 프로필 사진 아이디 |
| original_img | | Not Null | 원본 파일명 |
| uploaded_img | | Not Null | S3 업로드명 |

[Table 22] Problem

| Problem | | | |
|---------------------|-----|------------|-------------|
| Field | Key | Constraint | Description |
| problem_id | PK | Not Null | 문제 아이디 |
| problem_title | | Not Null | 문제 제목 |
| problem_context | | Not Null | 문제 설명 |
| problem_requirement | | Not Null | 문제 요구사항 |
| problem_example | | Not Null | 문제 화면 예시 |
| problem_constraint | | Not Null | 문제 채점 기준 |
| problem_level | | Not Null | 문제 난이도(1~5) |
| problem_keyword | | Not Null | 문제 키워드 |
| problem_correct | | Not Null | 문제 정답 여부 |

[Table 23] Submission

| Submission | | | |
|-----------------|-----|------------|-------------|
| Field | Key | Constraint | Description |
| submission_id | PK | Not Null | 문제 제출 아이디 |
| submission_date | | Not Null | 문제 제출 날짜 |
| result | | Not Null | 문제 제출 결과 |
| username | FK | Not Null | 문제를 제출한 유저 |
| problem_id | FK | Not Null | 제출된 문제의 아이디 |

[Table 24] Comment

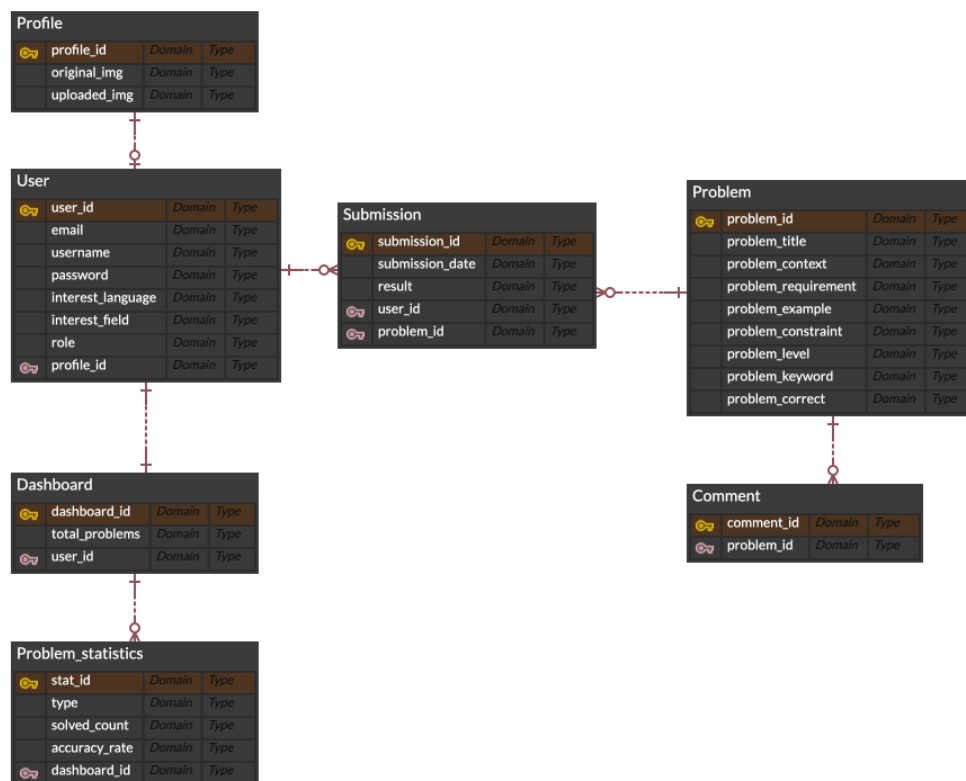
| Comment | | | |
|------------|-----|------------|-------------|
| Field | Key | Constraint | Description |
| comment_id | PK | Not Null | 문제 댓글 |
| problem_id | FK | Not Null | 해당 문제의 아이디 |

[Table 25] Dashboard

| Dashboard | | | |
|----------------|-----|------------|-------------|
| Field | Key | Constraint | Description |
| dashboard_id | PK | Not Null | 대시보드 ID |
| total_problems | | Not Null | 생성한 전체 문제 수 |

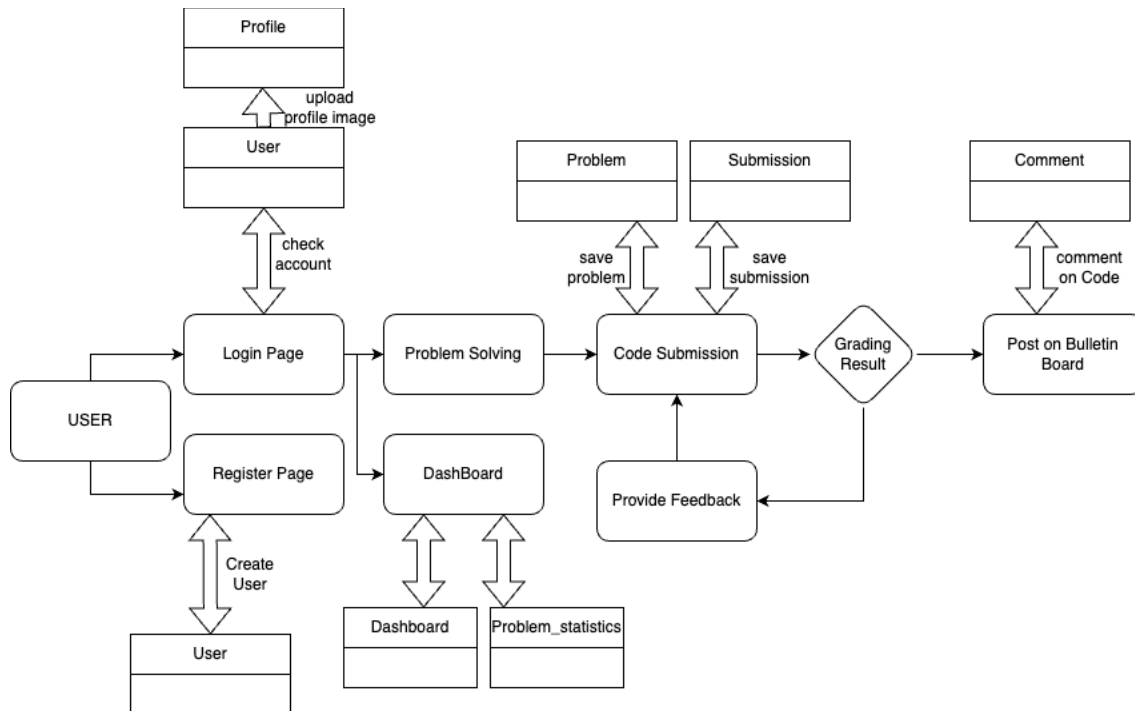
[Table 26] Problem statistics

| Problem_statistics | | | |
|--------------------|-----|------------|---------------|
| Field | Key | Constraint | Description |
| stat_id | PK | Not Null | 대시보드 ID |
| type | | Not Null | 문제 유형 |
| solved_count | | Not Null | 푼 문제 수 |
| accuracy_rate | | Not Null | 정답률 (백분율 값) |
| dashboard_id | FK | Not Null | dashboard와 연결 |



[Figure 4] Entity Relationship Diagram

3.2.4. Data Flow Diagram



[Figure 5] Data Flow Diagram

3.3. Performance Requirements

아래는 본 시스템의 성능 요구사항에 관한 것이다. 예측에 기반한 내용이며 실제 구현시에 달라질 수 있다.

3.3.1. Static Numerical Requirements

- 동시 접속 사용자 수: WebShooter 시스템은 최소 200명의 동시 접속 사용자를 지원할 수 있어야 한다. 이를 위해 AWS의 로드 밸런싱 기능을 활용하여 트래픽을 분산 처리함으로써, 학습 환경이 안정적으로 유지될 수 있어야 한다.
- 서버 사양: 백엔드 서버와 채점 서버는 AWS의 t2.medium 인스턴스를 사용하며, 각각 2 vCPU와 4GB RAM을 갖추어야 한다. 이 서버 사양은 사용자가 요청하는 학습 문제 생성 및 피드백 처리, AI 챗봇 응답을 실시간으로 처리할 수 있어야 한다. 데이터베이스 서버는 AWS RDS의 db.t3.medium 인스턴스를 사용하여 사용자 정보와 문제 풀이 기록, 학습 진척도 기록을 안정적으로 관리할 수 있어야 한다.

- **프론트 서버의 분산 운영:** Vercel에 배포된 프론트 서버는 사용자 요청에 따라 메모리를 자동으로 확장하여 사용자와의 상호작용을 실시간으로 지원할 수 있어야 한다. 이를 통해 사용자가 실시간 실습 환경에서 원활히 학습할 수 있도록 보장해야 한다.
- **사용자 디바이스 최소 사양:** WebShooter를 사용하는 사용자는 최소 1.0GHz 이상의 프로세서와 2GB 이상의 RAM을 갖춘 데스크톱이나 노트북을 사용해야 하며, Chrome, Edge, Firefox와 같은 최신 브라우저가 설치된 환경에서 원활히 작동해야 한다.
- **인터넷 연결 속도:** 시스템의 모든 기능이 원활히 동작하기 위해, 사용자는 최소 1Mbps 이상의 인터넷 연결을 확보해야 한다. 이는 실시간 문제 생성, 채점 및 피드백 제공에서 발생할 수 있는 지연을 최소화하기 위한 요구 사항이다.

3.3.2. Dynamic Numerical Requirements

- **인터넷 연결 속도:** 시스템의 모든 기능이 원활히 동작하기 위해, 사용자는 최소 1Mbps 이상의 인터넷 연결을 확보해야 한다. 이는 실시간 문제 생성, 채점 및 피드백 제공에서 발생할 수 있는 지연을 최소화하기 위한 요구 사항이다.
- **페이지 로딩 시간:** 사용자가 새로운 학습 콘텐츠나 실습 문제로 이동할 때 페이지 로딩 시간은 3초 이내여야 한다. 프론트 서버는 Vercel에 배포되어 글로벌 사용자에게 신속히 콘텐츠를 제공할 수 있어야 하며, 이를 통해 빠른 페이지 로딩이 보장되어야 한다.
- **문제 생성 및 피드백 제공 시간:** 사용자가 학습 문제 생성을 요청할 때, GPT API를 활용한 문제 생성 과정은 2분 이내에 완료되어야 한다. 문제 생성 시, 백엔드 서버는 사용자 요청을 신속히 처리하고 생성된 문제를 데이터베이스에 즉시 기록하여 사용자가 지체 없이 실습을 진행할 수 있도록 해야 한다.

- **채점 시스템 응답 시간:** 사용자가 제출한 코드에 대한 채점은 ****5초 이내에 사용자에게 전달되기 시작해야 한다. 이를 위해 채점 서버는 코드 실행을 위한 샌드박스 환경을 제공하고, 채점 결과를 데이터베이스에 실시간으로 저장할 수 있어야 한다. 여러 사용자의 채점 요청이 동시에 발생할 경우 메시지 큐(MQ)를 통해 작업을 분산 처리하여, 서버의 병목 현상을 방지해야 한다.
- **피드백 제공 시간:** 사용자가 제출한 코드에 대한 피드백은 5초 이내에 제공되어야 한다. 채점 결과와 연계된 피드백 시스템은 코드의 기능 검증과 UI 일치도를 기반으로 최적화 피드백을 제공하여 사용자가 코드의 개선 사항을 신속히 파악할 수 있도록 해야 한다.
- **회원가입 및 로그인 처리 시간:** 새로운 사용자가 회원가입을 하거나 기존 사용자가 로그인할 때, 모든 요청은 5초 이내에 완료되어야 한다. 이 과정에서 사용자 정보는 데이터베이스에 즉시 저장되며, 로그인 후에는 학습 기록과 사용자 설정이 로드되어야 한다.
- **학습 데이터 저장 및 로드 시간:** 사용자가 학습 진척 상황을 저장하거나 불러오는 작업은 5초 이내에 완료되어야 하며, 데이터베이스에 저장된 학습 기록은 대시보드에서 실시간으로 반영되어야 한다. 이를 통해 사용자는 자신의 학습 진행 상황을 쉽게 확인할 수 있어야 한다.
- **실시간 실습 환경 응답성:** 실습 환경은 사용자가 코드를 작성하고 즉시 실행 결과를 확인할 수 있도록 설계되어야 한다. 프론트엔드의 핫 리로딩 기능과 백엔드 서버와의 실시간 API 통신을 통해 프론트엔드 실습 환경의 응답 시간은 최대 5초 이내에 유지되어야 하며, 이를 통해 사용자는 실시간으로 학습 피드백을 받을 수 있어야 한다.
- **프롬프트 엔지니어링 시간:** LLM이 생성한 문제의 신뢰성을 평가하기 위해, 프롬프트 실행으로 얻은 문제와 테스트 케이스, 예시 답안을 생성한 후 서버에서 이를 풀어보는 과정을 통해 신뢰도를 확인해야 한다. 이 평가 프로세스는 최대 1분 이내에 완료되어야 하며, 문제를 통과하지 못할 경

우 대체 프롬프트로 재평가를 진행한다.

3.4. Logical Database Requirements

- WebShooter 시스템은 AWS RDS의 MySQL 데이터베이스를 통해 데이터를 관리한다. 이 데이터베이스는 사용자 계정 정보, 학습 문제 및 풀이 기록, 실시간 학습 데이터, 학습 진척도 및 히스토리를 저장하고 관리하는데 사용된다. 각 사용자의 계정 정보에는 고유 ID, 이메일, 암호화된 비밀번호 등이 포함되며, 이는 사용자의 인증과 학습 이력 조회를 위해 안전하게 저장된다.
- 데이터베이스는 사용자가 생성한 문제, 채점 결과, 정답률, 풀이 기록을 효율적으로 관리한다. 사용자가 특정 문제를 풀고 제출한 코드와 채점 결과는 각각 기록되고, 채점 서버로부터 전달받은 피드백을 통해 학습 진척 상황을 추적할 수 있도록 한다. 사용자는 자신이 해결한 문제, 제출한 코드, 정답률을 확인하고 개선할 수 있어야 하며, 이러한 기록은 대시보드에서 시각적으로 제공되어 사용자가 자신의 학습 패턴을 직관적으로 파악할 수 있도록 해야 한다.
- 데이터베이스는 사용자 접근 권한을 체계적으로 관리하여, 사용자는 자신의 데이터에만 접근할 수 있으며, 관리자만이 모든 사용자 정보와 학습 기록을 조회 및 관리할 수 있어야 한다.
- 데이터 무결성을 위해 모든 데이터는 참조 무결성과 정합성을 유지하도록 설계되어야 하며, 사용자와 문제 풀이 데이터의 중복을 방지하기 위한 유일성을 확보해야 한다. 트랜잭션 중 오류가 발생할 경우에는 롤백 처리가 가능하도록 하여 데이터 일관성을 보장해야 한다.

3.5. Design Constraints

3.5.1. Physical Design Constraints

- WebShooter 시스템은 웹 개발 학습을 위한 교육 목적의 플랫폼이므로 사용자는 데스크톱 또는 노트북을 통해 학습하는 것을 권장한다. 시스템의 기능과 실시간 실습 환경을 고려할 때, PC나 랩탑 환경에서 안정적이고 최적화된 학습 경험을 제공할 수 있도록 설계되어 있으며, 모바일 기기에서는 학습 경험이 제한될 수 있다. 서버와 데이터베이스는 AWS와 Vercel을 기반으로 운영되며, 데이터 저장과 관리를 위해 AWS RDS의 MySQL 데이터베이스를 사용하여 필요한 학습 데이터를 안전하게 관리할 수 있어야 한다.

3.5.2. Standards Compliance

- WebShooter 시스템은 백엔드 개발에 Java Spring Boot와 MySQL을, 프론트엔드에는 Next.js와 React.js를 사용하여 구현된다. Java와 JavaScript 코드는 각각의 표준 스타일 가이드를 준수하여 유지보수성과 일관성을 확보해야 한다. Java 코드는 Java Naming Conventions에 따라 작성하며, 변수는 lowerCamelCase, 클래스는 PascalCase를 사용한다. JavaScript 코드 역시 JavaScript Standard Style에 기반하여 작성되며, 변수명은 lowerCamelCase, 컴포넌트 이름은 PascalCase를 따른다. HTML과 CSS는 W3C 웹 표준을 준수하여 웹 호환성을 보장하고, 사용자 인터페이스는 접근성을 고려하여 설계된다.

3.6. Software System Attributes

3.6.1. Reliability

- WebShooter 시스템은 사용자가 학습 문제를 생성하고 실시간 실습 환경에서 코드를 실행하며, 피드백을 제공받는 과정에서 안정적인 작동을 보장해야 한다. 주요 기능의 실패를 방지하기 위해 AWS 인프라에서 백엔드와 채점 서버의 성능을 모니터링하고, 오류 발생 시 즉각적인 복구 절차를 마련해야 한다. 모든 사용자 데이터와 학습 기록은 정기적으로 백업되어 데이터 손실 위험을 최소화한다.

3.6.2. Availability

- AWS CloudWatch를 통해 서버 상태를 지속적으로 모니터링하고, 필요 시 자동으로 자원을 확장할 수 있도록 로드 밸런싱 및 자동 스케일링 기능을 적용한다.
- 시스템 장애 발생 시, 자동 복구 프로세스를 통해 5분 이내에 시스템이 복구되도록 한다.

3.6.3. Security

- 사용자 인증 및 데이터 접근은 HTTPS 통신을 통해 안전하게 보호되며, 비밀번호와 민감한 정보는 암호화하여 저장됩니다.
- 사용자별로 접근 권한을 구분하여 관리하고, 관리자 계정과 일반 사용자 계정의 권한을 분리합니다.

3.6.4. Maintainability

- 시스템의 유지 보수성을 높이기 위해 코드 모듈화를 철저히 진행하여 각각의 기능을 분리하고 인터페이스를 명확하게 정의한다. Spring Boot와 Next.js 프레임워크의 최신 버전을 사용하여 코드의 가독성과 확장성을

개선하며, 개발 및 유지 보수 과정에서 충분한 주석과 문서화를 통해 향후 코드 변경 및 추가에 대한 용이성을 보장한다.

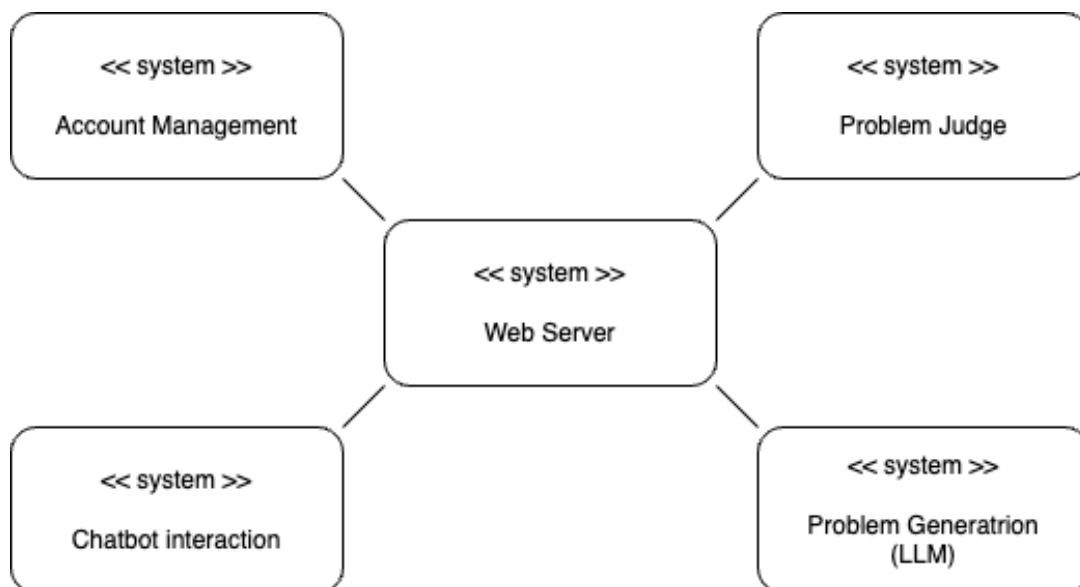
3.6.5. Portability

- WebShooter 시스템은 웹 기반으로 Chrome, Edge, Firefox 등의 최신 웹 브라우저에서 접근할 수 있으며, Windows 운영체제에 종속된다. Vercel 과 AWS 인프라를 활용하여 배포되므로, 다른 클라우드 서비스로의 이전이 용이하며, 소프트웨어 구조는 이식성을 고려하여 설계한다.

3.7. Organizing the Specific Requirements

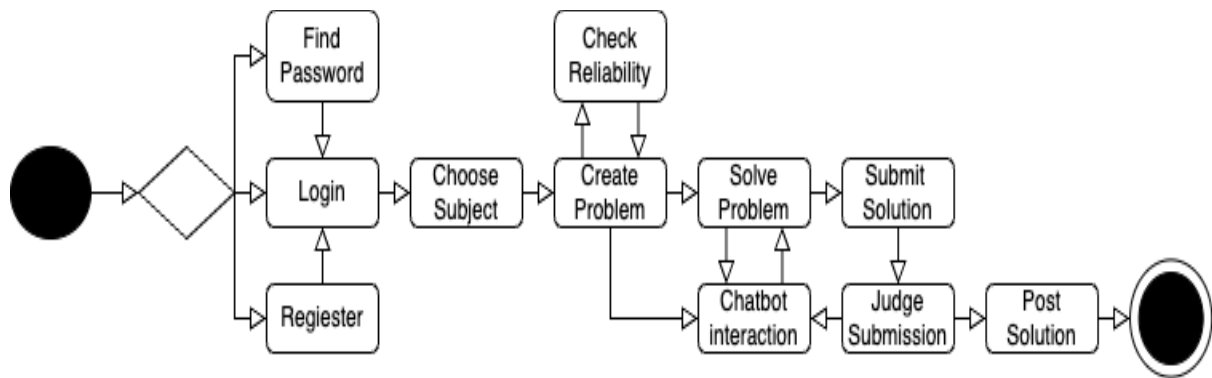
이 구간에서는 Unified Modeling Language(UML) 및 표 형식 기반의 그래픽 표기법을 사용하여 시스템 모델을 설명한다. 시스템 모델은 시스템, 서브 시스템 간의 관계를 설명한다.

3.7.1. Context Model



[Figure 6] Context model

3.7.2. Process Model

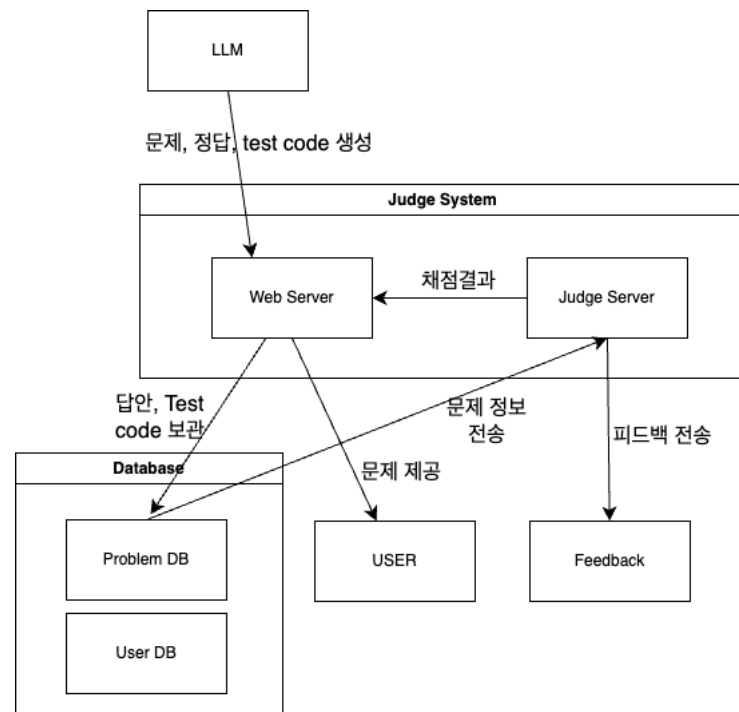


[Figure 7] Process model

3.7.3. Interaction Model

Use Case Diagram 참조(3.2.2)

3.8. System Architecture



[Figure 8] System Architecture

3.9. System Evolution

3.9.1. Limitation and Assumption

현재 시스템은 급격한 사용자 증가 시 성능 저하가 발생할 수 있다. 또한, 문제 생성을 위해 사용하는 LLM의 성능과 신뢰도에 따라 생성되는 문제 품질에 제한이 생길 수 있다. 또, 시스템은 AWS와 Vercel 같은 클라우드 제공업체의 서비스에 의존하므로, 해당 서비스의 정책 변경 또는 사용 제한에 따라 시스템 운영에 영향을 받을 수 있다.

시스템은 일정 수준의 평균적인 트래픽을 유지할 것으로 가정하고 있으며, 이 수준을 초과하는 과도한 트래픽이 발생하지 않을 것으로 예상된다. 또한, 클라이언트는 Chrome, Edge, Firefox 등 최신 웹 브라우저를 사용할 것을 가정하여 구형 브라우저 호환성은 고려하지 않았다. LLM의 GPT API와 AWS CloudWatch 등의 외부 API가 지속적으로 안정적으로 제공될 것이라는 전제 하에 시스템이 운영된다.

3.9.2. Evolution of Hardware and Change of User Requirements

사용자가 증가할 경우, 백엔드 서버와 채점 서버는 현재보다 높은 사양으로 업그레이드가 필요하다. 현재 각 서버는 2 vCPU와 4GB RAM을 갖추고 있지만, 향후 확장을 위해 더 많은 CPU 코어와 RAM을 제공하는 인스턴스로 전환하여 처리 능력을 강화할 수 있다. 또한, 트래픽이 크게 증가할 경우, 오토스케일링을 적용하여 사용자 수에 맞춰 서버 인스턴스 수를 자동으로 조절함으로써 서비스 안정성을 유지할 수 있다. 데이터베이스 역시 트랜잭션 양이 늘어날 가능성을 고려하여 읽기 전용 복제본 또는 캐싱을 추가해 응답 시간을 줄이고 데이터 처리 성능을 향상시킬 수 있다.

더 나아가 사용자들이 보다 높은 수준의 문제와 피드백을 기대할 경우, 성능이 우수한 최신 LLM 모델을 도입할 필요가 있다. 현재 사용 중인 GPT API가 더 고도화된 버전으로 업데이트 되면 이를 반영하고 이와 함께, 더 빠른 응답을 위해 LLM 모델을 프록시 서버로 분리하여 처리하거나, LLM 응답 속도를 최적화하기 위한 별도의 캐싱 계층을 두어 성능과 품질을 지속적으로 향상시킬 수 있다.

4. Appendix

4.1. W3C HTML & CSS Standards

HTML 및 CSS 는 W3C 웹 표준을 준수하여 작성되며, 웹 호환성과 접근성을 보장한다.

W3C HTML Standard : <https://www.w3.org/standards/>

4.2. Software Requirements Specification

소프트웨어 요구사항 명세서 IEEE 권장사항 (IEEE Recommend Practice for Software Requirements Specifications, IEEE-Std-830)에 따라 작성되었다.

4.3. Document History

[Table 27] Document History

| Date | Version | Description | Writer |
|------------|---------|-------------------------------------|--------|
| 2024/11/10 | 1.0 | Addition of 1, 4 | 박경일 |
| 2024/11/10 | 1.0 | Addition of 2, 3.9 | 권서진 |
| 2024/11/10 | 1.0 | Addition of 3.1 | 김현진 |
| 2024/11/10 | 1.0 | Addition of 3.2, 3.8 | 이유진 |
| 2024/11/10 | 1.0 | Addition of 3.3, 3.4, 3.5, 3.6, 3.7 | 김문수 |