

# Requirements Specification

BePro



소프트웨어공학개론 10조

2019313485 김지윤

2019312493 최윤채

2020312562 심윤보

2017312967 김주엽

2018313398 서정엽

2019314980 부윤종

2018311304 성보경

## Contents

1	Introduction.....	8
1.1	Purpose .....	8
1.2	Scope .....	8
1.3	Definitions, Acronyms, and Abbreviations .....	9
1.4	References.....	10
1.5	Overview.....	10
2	OVERALL DESCRIPTION .....	11
2.1	PRODUCT PERSPECTIVE.....	11
2.2	PRODUCT FUNCTIONS .....	13
2.2.1	MAIN PAGE .....	14
2.2.2	HOME PAGE.....	14
2.2.3	PROBLEM LIST PAGE.....	14
2.2.4	PROBLEM DESCRIPTION/SOULTION PAGE.....	15
2.2.5	SOLUTION RESULTS/AI ANALYSIS PAGE .....	15
2.2.6	LOGIN PAGE .....	16
2.2.7	MY PAGE (프로필 페이지).....	16
2.2.8	NOTICE PAGE.....	16
2.3	USER CLASSES AND CHARACTERISTICS .....	17
2.3.1	SYSTEM ADMINISTRATOR.....	17
2.3.2	USER.....	17
2.4	OPERATING ENVIRONMENT .....	18
2.5	DESIGN AND IMPLEMENTATION CONSTRAINTS .....	18
2.6	ASSUMPTIONS AND DEPENDENCIES .....	19
3	External Interface Requirements .....	19
3.1	User Interface .....	19

3.1.1	User Interface of Main Page .....	19
3.1.2	User Interface of Home Page .....	26
3.1.3	User Interface of Login Page .....	29
3.1.4	User Interface of Questions Page .....	30
3.1.5	User Interface of Result Page .....	37
3.1.6	User Interface of My Page .....	40
3.1.7	User Interface of Notice Page .....	43
3.1.8	User Interface of Problem Examiner Page .....	45
3.2	Hardware Interface.....	48
3.2.1	Hardware Interface of User Device .....	49
3.3	Software Interface .....	49
3.3.1	Software Interface of ChatGPT .....	49
3.3.2	Software Interface of Oracle Autonomous Database .....	51
3.4	Communication Interface.....	52
3.4.1	Communication Interface of Web Browser .....	52
4	SYSTEM REQUIREMENTS.....	53
4.1	FUNCTIONAL REQUIREMENTS .....	53
4.1.1	USE CASE .....	53
4.1.2	USE CASE DIAGRAM .....	63
4.1.3	DATA FLOW DIAGRAM.....	64
4.2	NON-FUNCTIONAL REQUIREMENTS.....	64
4.2.1	PRODUCT REQUIREMENT.....	64
4.2.1.1	USABILITY REQUIREMENT .....	64
4.2.1.2	PERFORMANCE REQUIREMENT .....	64
4.2.1.3	DEPENDABILITY REQUIREMENT .....	65
4.2.2	ORGANIZATIONAL REQUIREMENT .....	65
4.2.2.1	ENVIRONMENTAL REQUIREMENT.....	65

4.2.2.2	DEVELOPMENT REQUIREMENT .....	65
4.2.3	EXTERNAL REQUIREMENT .....	65
4.2.4	ETHICAL REQUIREMENT .....	65
4.2.5	SAFETY/SECURITY REQUIREMENT .....	66
4.3	ORGANIZING SYSTEM FLOW .....	66
4.3.1	CONTEXT MODEL .....	66
4.3.2	PROCESS MODEL .....	66
4.3.3	INTERACTION MODEL .....	67
4.3.4	BEHAVIOR MODEL .....	67
4.3.4.1	DATA FLOW DIAGRAM .....	67
4.3.4.2	SEQUENCE DIAGRAM .....	67
4.4	SYSTEM ARCHITECTURE .....	68
5	SUPPORTING INFORMATION .....	69
5.1	SOFTWARE REQUIREMENT SPECIFICATION .....	69
5.2	DOCUMENT HISTORY .....	69

## 그림 목차

Figure 1 Framework.....	17
Figure 2 Design of Main Page – Before Login.....	22
Figure 3 Design of Main Page – After Login.....	25
Figure 4 Design of Home Page.....	28
Figure 5 Design of Login Page.....	30
Figure 6 Design of Questions Page – Problem List.....	33
Figure 7 Design of Questions Page – Problem Description.....	36
Figure 8 Design of Result Page.....	39
Figure 9 Design of My Page.....	42
Figure 10 Design of Notice Page.....	45
Figure 11 Design of Problem Examiner Page.....	48
Figure 12 Use Case Diagram.....	63
figure 13 Data Flow Diagram.....	64
Figure 14 Context Model.....	66
Figure 15 Process Model.....	67
Figure 16 Sequence Diagram.....	68
Figure 17 System Architecture.....	68

## 표 목차

Table 1. Acronyms/Abbreviation .....	9
Table 2. Terms and definitions.....	9
Table 3 Table of Product Function.....	13
Table 4 User Interface of Main Page – Before Login.....	19
Table 5 Design of Main Page – After Login.....	23
Table 6 User Interface of Home Page .....	26
Table 7 User Interface of Login Page .....	29
Table 8 User Interface of Questions Page – Problem List.....	30
Table 9 User Interface of Questions Page – Problem Description .....	33
Table 10 User Interface of Review Page.....	37
Table 11 User Interface of My Page.....	40
Table 12 User Interface of Notice Page .....	43
Table 13 User Interface of Problem Examiner Page .....	45
Table 14 Hardware Interface of User Device .....	49
Table 15 Software Interface of ChatGPT .....	50
Table 16 Software Interface of Oracle Autonomous Database .....	51
Table 17 Communication Interface of Web Browser .....	52
Table 18 Functional Requirements Abstraction .....	53
Table 19 Use Case of Login .....	55
Table 20 Use Case of Learning Guideline.....	55
Table 21 Use Case of Problem History.....	56
Table 22 Use Case of My Class .....	56
Table 23 Use Case of Problem List.....	57
Table 24 Use Case of Problem Filter.....	57

Table 25 Use Case of Problem Description .....	58
Table 26 Use Case of Code Save.....	59
Table 27 Use Case of Code Execution .....	59
Table 28 Use Case of Code Submit.....	60
Table 29 Use Case of AI Code Analysis.....	60
Table 30 Use Case of Notice .....	61
Table 31 Use Case of Problem Create .....	62
Table 32 Use Case of GitHub Code Upload.....	62
Table 33 Document History .....	69

# 1 Introduction

## 1.1 Purpose

본 문서는 프로그래밍 학습자를 위한 "BePro" 서비스(이하 '본 서비스')를 제공하기 위한 소프트웨어 요구 명세서이다. 본 서비스는 2023년 1학기 성균관대학교 소프트웨어공학개론 강의 10조(이하 '개발팀')에 의해 고안되고 개발된다. 본 요구 명세서에서 소프트웨어 시스템의 요구 사항을 요약 및 분석하였으며, 이에 근거하여 본 서비스의 시스템을 개발한다.

본 문서는 개발팀 7명과 강의 조교님, 강의 교수님이 열람하는 것을 상정하여 작성되었다. 예외적으로, 소프트웨어공학개론 수강자 또한 학습 및 교육의 용도로 본 문서를 열람할 수 있다. 본 문서를 재 배포 및 수정하는 것은 자유이지만, 상업적 용도로 활용 시 개발팀의 허가를 얻어야 한다.

서비스 내에서 사용자가 제출한 코드를 채점하는 것으로부터 시작하여, 최종적으로 사용자가 본인이 제출한 코드에 대한 평가와 분석까지 가능할 수 있도록 서비스를 제공한다. 이 과정에서 웹 서버, 웹 인터페이스 및 사용자의 코드를 채점하는 시스템이 개발되며 세부사항은 이하 문서에 명시한다

본 문서의 목적은 프로그래밍 학습자를 위한 코딩 학습 및 교육 서비스인 "BePro"를 개발하고 제공함에 있어서 요구사항을 정리하고 명시하는 데 있다. "BePro"는 프로그래밍을 배우고자 하는 입문자 혹은 학습자가 다양한 난이도의 문제를 해결하고 적절한 피드백을 받는 과정을 통해 기초 프로그래밍이 가능하도록 수준을 향상시킬 수 있는 코딩 교육 웹서비스이다. 제공되는 서비스를 통해 개개인의 실력에 맞게 학습하고, 소프트웨어 엔지니어로서의 기초 역량을 기르는 것을 목표로 한다. 이를 위해 웹 서버, 사용자의 코드를 채점하는 기능 등이 개발되며 세부사항은 이하 문서에 명시한다

## 1.2 Scope

본 서비스는 프로그래밍 학습자에게 Python, C, C++, JAVA 언어를 사용한 프로그래밍 문제 해결 환경을 제공하기 위해 고안되었다. 학습이 목적인 본 서비스는 쉽고 편리하게 사용할 수 있도록 직관적인 UI/UX를 필요로 하며, 기능적 오류가 없도록 해야 한다. 또한 채점결과 오류와 잘못된 정보로 학습자의 학습을 방해하지 않도록 해야 한다. 이외에도 학습에 도움을 주기 위한 리뷰 게시판과 추가적인 팁 제공, 학습 진도 가이드 기능 등을 제공한다. 이를 구현하기 위해 front-end에서는 React 기반으로, back-end는 Django를 기반으로 구축하며 이와 관련된 상세



정보는 이하 문서에서 설명하도록 한다.

### 1.3 Definitions, Acronyms, and Abbreviations

이하의 표는 본 문서에서 활용된 약어의 정의를 나타낸다.

**Table 1. Acronyms/Abbreviation**

Acronyms/Abbreviation	Description
UI/ UX	User Interface/ User Experience
GUI	Graphic User Interface
DB	Database
URL	User Resource Locator
HTTP	Hyper Text Transfer Protocol
API	Application Programing Interface
OS	Operating System
HTML	Hyper Text Markup Language
CSS	Cascading Style Sheet
API	Application Programming Interface
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
AI	Artificial Intelligence
SQL	Structured Query Language

이하의 표는 본 문서에서 활용된 전문용어의 정의를 나타낸다.

**Table 2. Terms and definitions**

Term	Definition
Front-end	사용자가 볼 수 있는 화면(인터페이스) 또는 데이터 처리 시스템의 시작점.
Back-end	웹사이트나 웹 애플리케이션 또는 모바일 솔루션의 프로세스와 관련된 서버 측(Server-side)과 데이터베이스를 관

	리해 주는 기술.
React	user interface를 만들기 위한 JavaScript 기반의 라이브러리.
Django	웹사이트를 개발할 수 있도록 돕는 구성요소로 이루어진 파이썬 웹 프레임워크.
Git	파일의 변경사항을 추적하고, 사용자들 간의 작업을 조율하는데 사용되는 분산 버전 제어 시스템.
Oracle	미국 오라클(Oracle) 사의 관계형 데이터베이스 관리 시스템(RDBMS)의 이름.
ChatGPT	OpenAI에서 개발한 GPT(Generative Pre-trained Transformer) 아키텍처를 기반으로 한 언어 모델로, 대화식 환경에서 인간과 같은 텍스트 응답을 생성한다.
Host Server	클라이언트에 리소스, 서비스 또는 기능을 제공하는 서버로, 웹 사이트, 데이터베이스 또는 기타 애플리케이션을 호스팅한다.
Client	클라이언트-서버 아키텍처에서 서버로부터 서비스, 리소스 또는 기능을 요청하는 소프트웨어 애플리케이션 또는 하드웨어 장치.
OAuth	제3자 클라이언트에서 웹사이트나 애플리케이션의 사용자 정보에 대해 접근 권한을 위임받을 수 있는 표준 프로토콜.

## 1.4 References

- IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications, In IEEEExplore Digital Library  
<http://ieeexplore.ieee.org/Xplore/questhome.jsp>
- [https://github.com/skkuse/2022fall\\_41class\\_team2/blob/main/docs/Software\\_Requirements\\_Specification\\_Team2.pdf](https://github.com/skkuse/2022fall_41class_team2/blob/main/docs/Software_Requirements_Specification_Team2.pdf)

## 1.5 Overview

본 소프트웨어 요구사항 명세서는 네 부분 (Chapter 2~4)으로 구성되어 있다. 본 문서의

이하 내용 구성은 다음과 같다. Chapter 2에서는 제품의 관점과 기능, 예상 사용자 특징, 구동 환경 및 제약 조건에 대해 서술한다. Chapter 3에서는 서비스 기능을 위한 인터페이스에 대한 요구사항을 서술한다. Chapter 4에서는 기능 요구사항 이외의 요구 사항을 상세히 서술한다. 여기에는 제품 요구사항, 조직 요구사항, 사회적 요구사항 등을 포함된다.

## 2 OVERALL DESCRIPTION

### 2.1 PRODUCT PERSPECTIVE

현대 한국 사회에서는 디지털 환경과 인터넷 기술의 빠른 발전으로 인해 프로그래밍에 대한 관심이 크게 증가하고 있다. 이에 따라 프로그래밍 학습을 위한 다양한 학원, 웹사이트, 부트캠프 등의 교육 기관과 자료들이 생겨나면서 접근성이 개선되고 있다. 그러나 프로그래밍을 처음 시작할 때, 이론뿐만 아니라 문법, 개념, 로직 등 다양한 측면에서 새로운 정보를 받아들여야 하기 때문에 쉽지 않은 과정을 겪게 된다. 특히 변수와 데이터 타입 등 기초적인 개념을 이해하지 못하면 이후 학습에 어려움을 겪을 수 있고, 오류를 해결하는 과정에서도 많은 시간과 노력이 필요하게 된다.

그럼에도 불구하고 이러한 어려움을 극복하면 프로그래밍에 대한 흥미와 열정이 커지며, 다양한 기술과 프로젝트를 수행할 수 있는 능력을 갖추게 된다. 따라서 현대 사회에서는 프로그래밍에 대한 이해와 학습이 매우 중요한 역할을 하고 있으며, 이를 위한 다양한 학습 자원들이 증가하고 있다. 이에 따라 다양한 프로그래밍 교육 콘텐츠가 제공되고 있으며, 초등학생부터 성인, 심지어는 취업한 직장인들까지 모든 연령대에 걸쳐 접근할 수 있는 환경이 마련되어 있다.

하지만, 코딩을 처음 접하는 경우, 나이나 배경에 관계없이 어려움을 겪을 수 있다. 코드 작성 방법이나 문법적인 요소를 이해하는 것도 어렵겠지만, 자신이 생각한 로직을 코드로 옮기는 것이 또한 어려울 수 있다. 또한, 프로그래밍 과정에서 디버깅에 어려움을 겪는 경우가 흔하다. 예를 들어, 변수의 값이 원하는 대로 출력되지 않거나, 로직이 잘못되어 코드가 동작하지 않는 등의 상황이 발생할 수 있다. 이러한 디버깅 과정에서 막히면 많은 사람들이 흥미를 잃고 포기하게 된다.

따라서, 이러한 상황에서 AI 기술을 활용하여 사용자들에게 가시적으로 디버깅을 도와주는 웹사이트를 구축할 필요성이 있다. 이러한 기능을 할 수 있는 웹사이트인 BePro를 만드는 것이 팀 프로젝트의 목표이다. BePro는 프로그래밍을 학습할 수 있는 환경과 작성한 코드에 대한 AI 학습 도구를 제공하여 문제가 있는 부분을 시각적으로 쉽게 파악할 수 있는 서비스를 제공하는 것을 목표로 한다. 이를 통해 사용자들은 더욱

쉽게 프로그래밍을 접하고 경험할 수 있을 것이다. BePro와 같이 AI 기술을 활용한 학습 도구들은 더욱 쉽게 프로그래밍을 접할 수 있는 환경을 제공함으로써, 코딩을 새롭게 배우는 모든 사람들에게 코딩의 문턱을 낮춰줄 것을 기대한다.

## 2.2 PRODUCT FUNCTIONS

Table 3 Table of Product Function

Function Requirement Number	Name of Function	Function Description
F1	난이도별 문제 리스트 요청	User가 선택한 난이도에 대한 문제들 리스트를 요청
F2	문제 설명	문제의 조건과 상황에 대한 설명을 해주는 함수
F3	Code Editor 구현	User가 코드를 작성할 수 있는 code editor를 제공
F4	작성한 code 저장	Code 작성 중간에 User가 저장할 수 있는 기능
F5	작성한 code 실행	작성한 Code를 실행하기 위해 Testcase를 통해 Code 실행
F6	작성한 code 제출	작성한 Code를 제출
F7	실행 결과 확인	Testcase와 제출한 Code의 실행 결과 출력
F8	제출 코드 저장	제출한 Code를 파일로 만들어 저장
F9	Code 리뷰	ChatGPT API에 리뷰 요청하고 기록
F10	Code 주석 추가	ChatGPT API에 주석 요청하고 기록
F11	Code Refactoring	ChatGPT API에 Refactoring 요청하고 기록
F12	변경 코드 저장	F9, F10, F11을 통해 변경된 코드를 해당 파일에 덮어쓰워 저장
F13	git commit 요청	F8, F12에서 저장된 파일을 User의 GitHub에 commit하는 함수
F14	LogIn	User의 기록을 확인하고 GitHub과 연동하기 위해 LogIn

F15	User 정보 요청	User의 진행사항이나 개인 정보를 확인할 수 있도록 정보 요청
F16	개인정보 변경	User의 정보에서 변경사항을 수정하기 위한 함수

### 2.2.1 MAIN PAGE

User가 처음 "BePro"에 접속하면 'MAIN PAGE'에 도달한다. 가장 왼쪽 위에는 사이트 이름인 "BePro"가 명시되어 있고 오른쪽 위에는 "Login" 과 "Sign up"을 통해서 User가 Login하거나 새로 계정을 생성할 수 있도록 한다. Nav 영역에는 "Home", "Questions" 등의 여러 Page로 이동할 수 있도록 구성한다.

이후 Login을 한 이후의 MAIN PAGE는 "Login", "Sign Up"이 위치했던 곳에 "고객센터"와 User의 이름으로 버튼을 생성하여 User의 정보를 확인할 수 있는 MY PAGE로 향하는 버튼을 만든다. 또한, Nav영역 아래의 빈 공간에는 "BePro"에 대한 간단한 설명과 제공해주는 기능을 썸네일과 함께 구성한다.

### 2.2.2 HOME PAGE

Home Page에서는 User가 수강하고 있는 강의나 풀고 있던 문제를 나열해주어야 한다. 따라서, 가장 상단에는 수강 중인 강의를 표시해주고 간단하게 진행사항을 확인할 수 있도록 한다. 수강 중인 강의 밑에는 User가 풀고 있던 문제들에 다시 쉽게 접근이 가능하도록 문제의 ID, 제목, 관련된 알고리즘, 난이도를 함께 표시해주어야 한다. 또한, 자신이 수강하거나 풀고 있는 문제 외에 다른 강의에 대한 추천을 확인할 수 있도록 하는 것도 필요하다.

### 2.2.3 PROBLEM LIST PAGE

문제들은 ID, 제목, 관련 알고리즘, 난이도를 가지고 있도록 구성하며 User가 문제들을 난이도에 따라서 풀 수 있도록 난이도를 선택하면 해당 난이도의 전체 문제들을 나열해 주도록 구성한다. 만약, 전체 문제의 list들이 pagination으로 나뉘어져 있을 경우 해당 페이지별로 문제 리스트를 나열할 수 있도록 구성한다. 또한, "완료", "진행중", "미완료", "AI

추천"과 같이 문제들에 대한 상태를 알 수 있도록 구성한다.

문제들을 리스트로 나열할 때 작성해야 할 정보는 문제의 ID, 제목, 관련 알고리즘, 난이도를 확인할 수 있도록 제공한다. 또한, User가 자신이 풀 문제를 선택하면 PROBLEM DESCRIPTION/SOULUTION PAGE로 이동하게 된다.

#### 2.2.4 PROBLEM DESCRIPTION/SOULUTION PAGE

문제 상세/풀이 페이지는 해당 문제에 대한 자세한 설명과 Testcase를 제공한다. 이를 통해서 User가 자신이 풀어야 할 문제에 대한 자세한 정보를 확인할 수 있으며 소스코드를 작성하기 위한 Code Editor를 한 화면에 구성한다. Code Editor 상단에 자신이 어떠한 프로그래밍 언어를 선택할 것인지 드롭다운 리스트를 구성하여 선택할 수 있도록 한다. Code Editor를 위한 기능으로는 자신이 작성하고 있는 Code를 수시로 저장할 수 있도록 하는 기능이 필요하며 작성한 코드를 자신이 입력하고 싶은 Testcase와 함께 실행할 수 있도록 하는 기능과 코드를 제출할 수 있는 기능이 필요하다.

또한, 실행시킨 결과를 확인할 수 있도록 작성한 Code에 대한 출력 결과와 Test 결과를 보여줘야 한다. 또한, 제출한 Code를 파일로 저장하는 기능이 필요하며 이 기능을 통해서 User의 GitHub과 연동하여 commit할 수 있도록 하는 추가적인 기능을 구성한다. 마지막으로 제출 결과가 성공이라면 "PROBLEM LIST PAGE"로 이동할 수 있는 버튼과 "SOLUTION RESULTS/AI ANALYSIS PAGE"로 이동할 수 있는 버튼을 제공하여 다른 문제를 풀거나 해당 문제에 대한 자세한 분석을 할 수 있는 페이지를 선택할 수 있도록 한다.

#### 2.2.5 SOLUTION RESULTS/AI ANALYSIS PAGE

User가 문제를 푼 이후 자세한 분석을 필요로 하여 "SOLUTION RESULTS/AI ANALYSIS PAGE"로 이동을 선택한 경우에 도착하는 PAGE로 해당 PAGE에서는 Code Review 요청, 주석 추가 요청, Code Refactoring 요청과 같이 작성한 소스코드를 개선시킬 수 있는 요청들을 선택할 수 있도록 구성한다. 자신이 작성한 소스코드를 읽어와 보여주고 특정한 요청을 선택하면 해당 요청을 통해 얻은 결과를 한 화면에 보여줄 수 있도록 한다.

위와 같은 요청들을 처리하기 위해서 백엔드에서 해당 요청에 맞도록 GPT에 요청을 보내고 데이터베이스에 기록한다. GPT 응답 Table은 종류별로 구성하여 관리한다. 위처럼 변경된 Code를 원본 소스코드 파일과 구분하여 저장을 하는 기능이 필요하며 저장된 코드를 User의 GitHub에 commit할 수 있도록 요청하는 기능이 필요하다.

### 2.2.6 LOGIN PAGE

LOGIN PAGE에서는 User가 자신의 email과 password를 입력할 수 있도록 구성한다. Login에 실패하면, email혹은 password가 유효하지 않다는 팝업창을 띄우고, 입력창을 초기화한다.

Email과 password를 입력한 후 회원가입 button을 눌러 신규 유저의 정보를 데이터베이스에 저장한다. 이후 입력창을 초기화한다.

ID/비밀번호 찾기 button을 눌러 기존 유저의 정보를 가져와 password가 무엇인지 유저에게 알려준다. 버튼을 누르면 팝업창이 나와 연동된 email 계정을 통한 본인인증을 진행한다. 인증이 완료되면 팝업창을 닫고, 데이터베이스에서 email을 가지고 있는 유저의 정보를 가져와 password를 새로운 팝업창으로 알려준다.

### 2.2.7 MY PAGE (프로필 페이지)

Nav 영역 오른쪽 위에 위치한 My page 버튼을 통해서 들어올 수 있는 Page로 자신의 개인정보를 열람할 수 있도록 해야 하며 필요한 경우에는 수정도 가능하도록 구성한다. 또한, 자신의 GitHub 정보를 관리할 수 있어야 하며 풀었던 문제들을 확인할 수 있도록 해야 한다. 이때 어떠한 문제를 풀었는지 쉽게 확인할 수 있도록 ID, 제목, 관련 알고리즘, 난이도를 모두 확인할 수 있도록 관리해야 한다.

### 2.2.8 NOTICE PAGE

Nav영역 상단 우측에 존재하는 "고객센터" 버튼을 클릭하면 들어오는 Page로



“BePro”사이트에 관한 공지사항이나 FAQ란을 만들어 User의 사이트 편의성을 증가시키기 위한 Page이다. 따라서, 각 공지나 FAQ에 대해서 제목을 확인해 해당 문서를 열람할 수 있도록 한다.

## 2.3 USER CLASSES AND CHARACTERISTICS

### 2.3.1 SYSTEM ADMINISTRATOR

시스템 관리자는 BePro를 운영하고 관리하는 관리자로 해당 system의 유지/보수를 책임지는 사람을 의미한다. System을 유지/보수하기 위해서 system에 대한 전반적인 이해가 필요하며 특히 아래 그림 1과 같은 Framework와 그 사이 통신에 대한 이해가 충분하며 활용 역량을 가지고 있어야 한다. 또한, BePro에서 User가 사용할 수 있는 언어인 Python, C/C++, JAVA에 대한 기본적인 문법 지식을 가지고 있어야 한다.

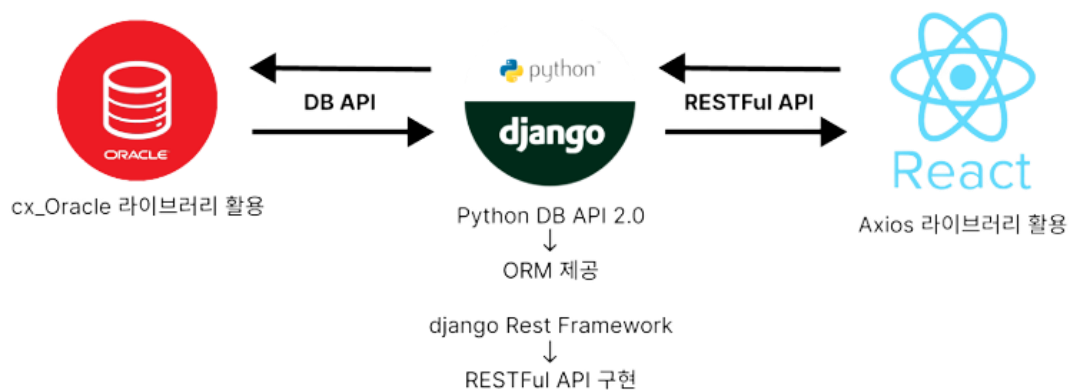


Figure 1 Framework

### 2.3.2 USER

“BePro”를 사용하는 User는 프로그래밍을 처음 접하여 배워보는 사람들이나 프로그래밍이 익숙하지 않은 사람을 대상으로 한다. 이러한 User는 자신이 학습할 프로그래밍 언어에 대한 아주 기초적인 지식은 필요로 하며 해당 문법지식을 가지고 여러 난이도의 문제들을 접하고 풀어내는 것을 목표로 해야 한다. 문제를 풀고 난 이후에는 자신의 소스코드에 대해서 리뷰해주고, 주석을 달아주고, 불필요한 부분을 Refactoring 해주는 기능을 통해서 프로그래밍에 대한 흥미를 가질 수 있다. User는 Chrome과 같은 모던 웹 브라우저를 통해 “BePro”에 접속하고 기본적인 GUI 및 Code Editor를 활용할 수 있는 능력이 필요하다.

## 2.4 OPERATING ENVIRONMENT

- ◆ Linux
- ◆ Mac OS
- ◆ Windows 10
- ◆ Windows 11

## 2.5 DESIGN AND IMPLEMENTATION CONSTRAINTS

본 시스템은 본 문서에 기재된 요구 사항을 바탕으로 고안되고 구현된다. 본 문서에 명시되지 않은 세부 조건들은 개발자의 재량에 의해 구현되고 설계될 수 있으며 아래의 제약 사항들을 준수하여 개발, 운영, 사용되어야 한다.

- 이미 널리 성능이 입증된 기술을 사용한다.
- 가능한 공개된 오픈소스를 사용한다.
- 시스템의 성능을 저하시키지 않는 방향으로 설계하고 구현한다.
- UI 설계 시 사용자의 입장에서 받아들이기 쉬운 방향으로 결정한다.
- 사용자에게 악영향을 끼칠 가능성이 있는 코드를 포함하지 않는다.
- 사용자가 본 시스템 외의 다른 제품을 설치할 필요 없이 시스템을 이용할 수 있도록 설계하고 구현한다.
- 서비스 이용자 수가 증가해도 시스템이 확장 가능해야 한다.
- 24시간 365일 서비스를 제공할 수 있어야 하며 시스템 장애 발생 시 빠른 복구가 보장되어야 한다.
- 사용자가 원활한 서비스 이용을 위해 빠른 응답 속도와 처리량이 보장되어야 한다.
- 시스템 유지보수를 용이하게 할 수 있도록 일관성, 가독성을 고려해야 한다.

## 2.6 ASSUMPTIONS AND DEPENDENCIES

본 문서의 시스템은 웹 기반으로 설계 및 구현되므로 사용자가 웹에 접속할 수 있는 환경 하에서 서비스를 사용할 것이라 가정한다. 따라서 웹의 불안정성으로 인해 야기되는 문제에 대해 고려하지 않는다. Chrome 브라우저의 시스템 요구사항과 2.4의 Operating Environment의 최소 Dependencies를 정리하면 아래와 같다.

1. 1GHz 이상 32비트(X86) 또는 64비트(X64) 프로세서 (Core i5 이상)
2. 1GB RAM(32비트) 또는 2GB RAM(64비트)
3. 16GB 사용 가능한 하드 디스크 공간(32비트) 또는 20GB(64비트)
4. WDDM 1.0 이상 드라이버와 DirectX 9 그래픽 디바이스

또한, Network 연결을 위한 NIC와 GUI와 소통하기 위한 마우스, 키보드 등의 HID가 필요하다. 위의 항목에서 기술한 조건을 만족하는 환경에서의 사용자 접근을 가정하고 그에 한해서 정상적 작동을 보장한다.

## 3 External Interface Requirements

### 3.1 User Interface

#### 3.1.1 User Interface of Main Page

Table 4 User Interface of Main Page – Before Login

이름	Main Page - Before Login
----	--------------------------

목적/설명	<p>메인페이지에 접속하였을 때, 사용자의 디바이스에서 로그인 되어있지 않은 상태일 경우 보이는 초기 화면이다.</p> <p>처음 들어온 고객들은 BePro 플랫폼의 주요 학습 서비스에 대해 안내받을 수 있으며, 각각의 서비스를 클릭해 더 자세한 설명을 확인할 수 있다.</p> <p>플랫폼을 이용하려는 고객들은 로그인 또는 회원가입을 통해 메인 페이지로 넘어갈 수 있다.</p>
입력 소스/ 출력대상	<p>클라이언트 / 호스트 서버</p> <p>호스트 서버 / 클라이언트</p>
범위/정확도/ 오차범위	페이지 전체 / - / -
단위	<p>웹 페이지</p> <p>Design: 1440 px * 1825 px</p> <p>Client Screen: 100 vw * 100 vh (클라이언트의 디바이스 크기) (좌우 맞춤, 스크롤)</p>
시간/속도	사용자 액션에 대한 서버간 통신 시간
기타 입출력관계	N/A
화면 형식과 구성	<p>A. Navigation Bar</p> <ul style="list-style-type: none"> <li>a. Logo - Navigation Bar 좌측에 로고가 표시된다. 로고를 클릭할 시 Landing Page 로 이동한다.</li> <li>b. Nav Links - 로고 우측에 Home 과 Questions 페이지로 이동할 수 있는 링크 버튼이 표시된다. 해당 링크를 클릭할 시, 로그인 화면으로 넘어간다.</li> <li>c. Login/Sign Up - Navigation Bar 우측에 Login 과 Sign Up 링크가 담긴 버튼이 표시된다. 각각을 클릭해 로그인 또는 회원가입 페이지로 이동할 수 있다.</li> </ul>

	<p>B. Body</p> <p>a. Hero Section</p> <ol style="list-style-type: none"> <li>Background - 코딩과 논리를 연상할 수 있는 기본적인 도형들과 기본 색상으로 간단한 배경 이미지를 구성한다.</li> <li>Title - 사용자들이 서비스에 가장 먼저 들어와 처음으로 볼 문구로, 서비스를 한 문장으로 요약해 표시한다.</li> <li>Description - 서비스에 대한 조금 더 자세한 설명을 표시한다.</li> <li>Sign Up - 설명을 읽은 후 고객을 끌어들이기 위해 바로 가입할 수 있는 버튼을 배치한다.</li> </ol> <p>b. Introduction Section</p> <ol style="list-style-type: none"> <li>Section Title - 주요 학습 서비스를 소개하는 섹션임을 드러낸다.</li> <li>Posts Square - 플랫폼의 주요 서비스들에 대해 카드 이미지 + 캡션의 형태로 정보가 제공되며, 더보기 버튼을 클릭해 자세한 설명을 확인할 수 있다.</li> </ol> <p>C. Footer</p> <ol style="list-style-type: none"> <li>Logo - Footer 좌측에 로고가 표시된다.</li> <li>Nav Links - Footer 중앙에 모바일 앱을 다운 받을 수 있는 링크 버튼과, BePro 커뮤니티, 그리고 BePro 팀 소개 페이지로 넘어갈 수 있는 링크 버튼이 표시된다.</li> </ol>
데이터 유형	이미지, 텍스트
명령 유형	버튼 클릭에 의한 페이지 변경
종료 메시지	N/A

# 미래형 AI 코딩 교육 플랫폼

Be Pro 설명

지금 가입하기

BePro 의 주요 학습 서비스

미래형 AI 코딩 교육 플랫폼 서비스 모음



나에게 맞는 강의 추천

[Learn More](#)



자동 코드 분석 및 리팩토링

[Learn More](#)



ChatGPT를 이용한 코딩 교육

[Learn More](#)

Figure 2 Design of Main Page – Before Login

**Table 5 Design of Main Page – After Login**

이름	Main Page - After Login
목적/설명	<p>메인 페이지에 접속하였을 때, 사용자의 디바이스에서 로그인 되어 있는 상태일 경우 보이는 초기 화면이다.</p> <p>사용자의 정보를 받아와 환영 문구를 띄워주고, BePro 플랫폼의 주요 학습 서비스에 대해 안내받을 수 있으며, 각각의 서비스를 클릭해 더 자세한 설명을 확인할 수 있다.</p> <p>플랫폼을 이용하려는 고객들은 상단 네비게이션 바를 이용해 Home, Questions, 고객센터, 마이페이지로 이동할 수 있다.</p>
입력 소스/ 출력대상	<p>클라이언트 / 호스트 서버</p> <p>호스트 서버 / 클라이언트</p>
범위/정확도/ 오차범위	<p>페이지 전체 / - / -</p>
단위	<p>웹 페이지</p> <p>Design: 1440 px * 1825 px</p> <p>Client Screen: 100 vw * 100 vh (클라이언트의 디바이스 크기) (좌우 맞춤, 스크롤)</p>
시간/속도	사용자 액션에 대한 서버간 통신 시간
기타 입출력관계	N/A
화면 형식과	<p>A. Navigation Bar</p> <p>a. Logo - Navigation Bar 좌측에 로고가 표시된다. 로고를</p>

구성	<p>클릭할 시 Landing Page 로 이동한다.</p> <p>b. Nav Links - 로고 우측에 Home 과 Questions 페이지로 이동할 수 있는 링크 버튼이 표시된다. 해당 링크를 클릭할 시, 로그인 화면으로 넘어간다.</p> <p>c. Notice - Navigation Bar 우측에 고객센터 링크가 담긴 버튼이 표시된다. 버튼을 클릭해 고객센터 페이지로 이동할 수 있다.</p> <p>d. Name - Notice 우측에 회원가입시 입력한 이름이 표시된다. 해당 버튼을 클릭 시, 드롭다운 메뉴가 생성되어, 마이페이지와 설정 페이지로 이동할 수 있는 링크가 나온다.</p> <p>B. Body</p> <p>a. Hero Section</p> <p>i. Background - 코딩과 논리를 연상할 수 있는 기본적인 도형들과 기본 색상으로 간단한 배경 이미지를 구성한다.</p> <p>ii. Title - 사용자들의 정보를 바탕으로 환영문구를 표시한다</p> <p>iii. Main Page - 해당 버튼을 클릭 시 메인 페이지로 이동할 수 있다.</p> <p>b. Introduction Section</p> <p>i. Section Title - 주요 학습 서비스를 소개하는 섹션임을 드러낸다.</p> <p>ii. Posts Square - 플랫폼의 주요 서비스들에 대해 카드 이미지 + 캡션의 형태로 정보가 제공되며, 더보기 버튼을 클릭해 자세한 설명을 확인할 수 있다.</p> <p>C. Footer</p> <p>a. Logo - Footer 좌측에 로고가 표시된다.</p> <p>b. Nav Links - Footer 중앙에 모바일 앱을 다운 받을 수 있는 링크 버튼과, BePro 커뮤니티, 그리고 BePro 팀 소개 페이지로 넘어갈 수 있는 링크 버튼이 표시된다.</p>
데이터 유형	이미지, 텍스트
명령 유형	버튼 클릭에 의한 페이지 변경



종료 메시지	N/A
--------	-----

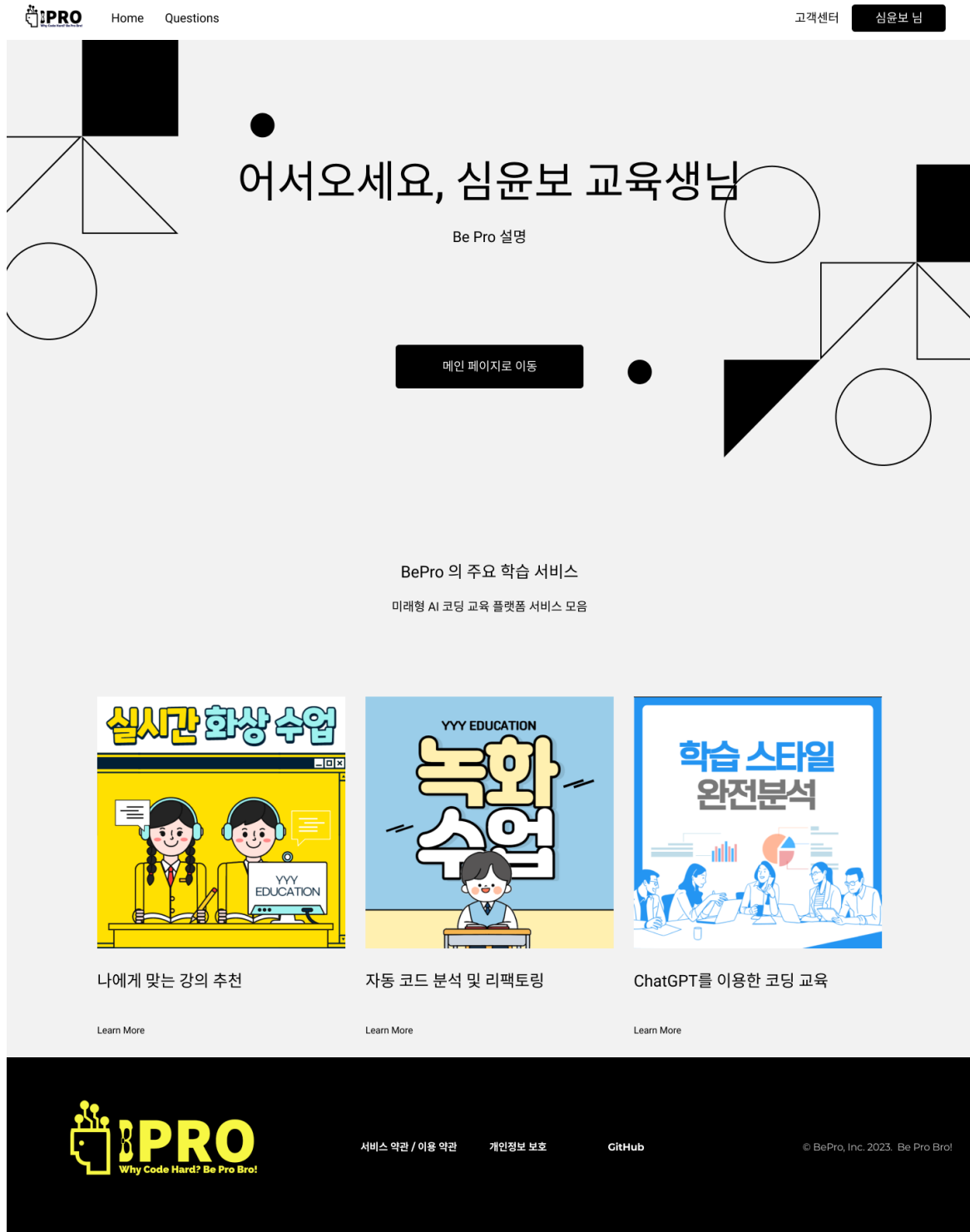


Figure 3 Design of Main Page – After Login

### 3.1.2 User Interface of Home Page

Table 6 User Interface of Home Page

이름	Home Page
목적/설명	<p>BePro 플랫폼의 기본 페이지이자, 사용자가 가장 많이 사용할 "진행 중인 강의", "풀고 있는 문제", "추천 강의"와 "추천 문제"를 보여주는 홈페이지이다.</p> <p>사용자는 이 페이지를 통해 BePro 서비스에서 하고 있던 작업 (강의/문제)로 바로 이동할 수 있으며, 본인이 푼 문제들과 수강한 강의, 그리고 배경 지식 및 사용자 정보를 이용해 강의와 문제들을 추천받을 수 있다.</p>
입력 소스/ 출력대상	<p>클라이언트 / 호스트 서버</p> <p>호스트 서버 / 클라이언트</p>
범위/정확도/ 오차범위	<p>페이지 전체 / - / -</p>
단위	<p>웹 페이지</p> <p>Design: 1440 px * 2191 px</p> <p>Client Screen: 100 vw * 100 vh (클라이언트의 디바이스 크기) (좌우 맞춤, 스크롤)</p>
시간/속도	사용자 액션에 대한 서버간 통신 시간
기타 입출력관계	N/A
화면 형식과 구성	<p>A. Navigation Bar</p> <p>B. Body</p> <p>a. Ongoing Lectures - 현재 수강하고 있는 강의 내역이 card</p>

	<p>형태로 표시한다. 강의명과 강사 이름이 텍스트 형태로 제공되고 각 강의의 수강을 완료한 정도가 보여진다. Card 를 클릭하면 해당 강의 링크로 이동한다..</p> <p>b. Ongoing Questions - 사용자가 시작했지만 아직 완료하지 못한 문제의 리스트를 표시한다. Question 을 클릭하면 해당 문제 풀이 페이지로 넘어간다.</p> <p>c. Recommendation - 사용자 정보, 강의 수강 내역, 풀이한 문제를 기반으로 추천하는 강의들을 card 형태로, 문제들을 bar 목록 형태로 제공한다.</p> <p>C. Footer</p>
데이터 유형	이미지, 텍스트
명령 유형	버튼 클릭에 의한 페이지 변경
종료 메시지	N/A

# Home

## 진행 중인 강의



### 기초 입출력

서정엽 선생님

8/10 완료



### 정렬 기초

김지윤 선생님

6/12 완료

## 풀고 있는 문제

# 1000	A+B	입출력	Level 1
# 1032	List Concatenation	자료구조	Level 2

## 추천 강의



### 자료구조개론

김주엽 선생님

★4.7/5

총 20강

Figure 4 Design of Home Page

### 3.1.3 User Interface of Login Page

Table 7 User Interface of Login Page

이름	Login Page
목적/설명	로그인을 하기 위한 페이지다.  로그인하지 않은 상태에서 서비스를 이용하려고 할 경우, 이 페이지를 거쳐 사용자를 인증해서 넘어가고, 그렇지 않은 경우, 로그인 후 Landing Page - After Login로 넘어간다.
입력 소스/ 출력대상	클라이언트 / 호스트 서버  호스트 서버 / 클라이언트
범위/정확도/ 오차범위	페이지 전체 / - / -
단위	웹 페이지  Design: 1440 px * 764 px Client Screen: 100 vw * 100 vh (클라이언트의 디바이스 크기) (좌우 맞춤, 스크롤)
시간/속도	사용자 액션에 대한 서버간 통신 시간
기타 입출력관계	N/A
화면 형식과 구성	A. Logo - 상단에 로고가 표시되며, 로고 클릭 시 메인 페이지로 넘어간다.  B. Github Account - 계정 연결을 안내하는 문구가 표시되고 버튼을 클릭하면 GitHub 계정으로 연동하여 로그인할 수 있게 연결된다.  C. Notice - 로그인에 문제를 겪는 유저들을 위해, FAQ를 볼 수 있는

	notice 페이지로 연결되는 고객센터 버튼이 표시된다.
데이터 유형	이미지, 텍스트
명령 유형	버튼 클릭에 의한 서버 연동 및 Authentication 또는 페이지 리다이렉션
종료 메시지	로그인을 성공적으로 완료 시, Landing Page로 이동된다.



**Figure 5 Design of Login Page**

### 3.1.4 User Interface of Questions Page

**Table 8 User Interface of Questions Page – Problem List**


이름	Questions Page - 문제 목록
----	------------------------

목적/설명	<p>사용자가 풀 수 있는 문제들의 목록이 나오는 페이지이다.</p> <p>전체 문제 목록이 표시되며, 이 중 완료(성공)한 문제, 풀고 있는 문제, 아직 풀지 않은 문제와 AI가 추천한 문제들로 나누어져, 각각 다른 색으로 칠해져 있고, 이 중 원하는 카테고리들을 선택해 정렬할 수 있다.</p> <p>찾고 있는 문제가 있을 시, 검색을 하면 검색에 적합한 문제들을 정렬하여 보여주며, 나온 문제들 중 하나를 선택하여 클릭하면 문제 풀이 페이지로 넘어간다.</p>
입력 소스/ 출력대상	<p>클라이언트 / 호스트 서버</p> <p>호스트 서버 / 클라이언트</p>
범위/정확도/ 오차범위	<p>페이지 전체 / - / -</p>
단위	<p>웹 페이지</p> <p>Design: 1440 px * 1698 px</p> <p>Client Screen: 100 vw * 100 vh (클라이언트의 디바이스 크기) (좌우 맞춤, 스크롤)</p>
시간/속도	<p>사용자 액션에 대한 서버간 통신 시간</p>
기타 입출력관계	<p>N/A</p>
화면 형식과 구성	<p>A. Navigation Bar</p> <p>B. Body</p> <p>a. Filtering Section</p> <p>i. Filter Buttons - 좌측 상단에 완료, 진행중, 미완료, AI 추천 버튼이 있어 해당 버튼을 클릭하면 조건에 맞게 문제들이 필터링된다.</p> <p>ii. Additional Filters - 우측 상단에 필터 버튼을 클릭하여 문제의 카테고리 또는 레벨에 대하여</p>

	<p>필터를 적용하여 정렬할 수 있다.</p> <p>iii. Search - 우측 상단에 검색창이 있어 글자를 입력한 후 우측의 돋보기 버튼을 누르면 원하는 문제를 검색할 수 있다.</p> <p>b. Problems Sections</p> <p>i. List - 조건에 맞는 문제 리스트를 표시한다. 초기화면은 모든 문제를 보여주며 문제의 번호, 이름, 분류, 난이도를 bar 형태로 표시한다.</p> <p>ii. Pagination - 조건에 맞는 문제의 개수에 따라 자동으로 pagination 이 진행되며, 현재페이지/전체페이지 형태로 표시된다.</p> <p>C. Footer</p>
데이터 유형	이미지, 텍스트
명령 유형	버튼 클릭에 의한 페이지 리다이렉션 또는 DB 검색 및 Information Retrieval
종료 메시지	N/A



## 문제 목록

완료	진행중	미완료	AI 추천	필터	검색	
# 1000	A+B	입출력	Level 1			
# 1001	A-B	입출력	Level 1			
# 1025	Fibonacci	알고리즘	Level 1			
# 1032	List Concatenation	자료구조	Level 2			

1/1

Figure 6 Design of Questions Page – Problem List

Table 9 User Interface of Questions Page – Problem Description

이름	Questions Page - 문제 풀이
----	------------------------

목적/설명	<p>사용자가 문제를 풀 때 나오는 페이지이다.</p> <p>현재 어떤 문제를 풀고 있는지 확인할 수 있으며, 문제에 대한 설명과, 문제 코드를 실행시켰을 때 Output을 한 눈에 볼 수 있다. 페이지 하단의 버튼들을 이용해 코드를 실행, 정지 시킬 수 있으며, 완성된 코드를 제출할 수 있다.</p>
입력 소스/ 출력대상	<p>클라이언트 / 호스트 서버</p> <p>호스트 서버 / 클라이언트</p>
범위/정확도/ 오차범위	페이지 전체 / - / -
단위	<p>웹 페이지</p> <p>Design: 1440 px * 1698 px</p> <p>Client Screen: 100 vw * 100 vh (클라이언트의 디바이스 크기) (좌우 맞춤, 스크롤)</p>
시간/속도	사용자 액션에 대한 서버간 통신 시간
기타 입출력관계	N/A
화면 형식과 구성	<p>A. Navigation Bar</p> <p>B. Body</p> <p>a. Description/Settings Section</p> <ol style="list-style-type: none"> <li>Question Info - 상단에 현재 풀고 있는 문제의 번호, 이름, 분류, 난이도를 bar 형태로 표시한다.</li> <li>Question Description - 해당 문제 내용이 표시된다.</li> <li>Language Selection - 프로그래밍 언어를 선택하기 위한 버튼. 마우스를 hover 하면 선택 가능한 프로그래밍 언어 종류가 dropdown menu 형식으로</li> </ol>

	<p>나타난다.</p> <p>b. Coding Section</p> <ul style="list-style-type: none"> <li>i. Coding Interface - 사용자가 문제에 대한 코드를 입력할 수 있는 입력창이 표시된다.</li> <li>ii. Input - 사용자가 작성한 코드에 테스트해보고 싶은 input 값을 입력할 수 있는 입력창이 있다.</li> <li>iii. Output - Input 에 입력한 testcase 를 실행한 결과값이 표시된다.</li> </ul> <p>c. Button Section</p> <ul style="list-style-type: none"> <li>i. Back - 버튼을 클릭하여 문제풀이 리스트 페이지로 돌아갈 수 있다.</li> <li>ii. Run - 작성한 코드를 실행할 수 있는 버튼이다.</li> <li>iii. Stop - 코드의 실행을 중단할 수 있다.</li> <li>iv. Submit - 해당 코드 채점이 요청된다.</li> </ul> <p>C. Footer</p>
데이터 유형	이미지, 텍스트
명령 유형	버튼 클릭에 의한 서버 통신 및 페이지 변경
종료 메시지	사용자가 작성한 코드 정답 여부

# 문제 풀이

# 1025

Fibonacci

알고리즘

Level 1

&lt;문제 설명&gt;

언어 선택

C

```
int fibonacci(int n) {  
    sca
```

Input

Output

Back

Run

Stop

Submit



서비스 약관 / 이용 약관

개인정보 보호

GitHub

© BePro, Inc. 2023. Be Pro Bro!

Figure 7 Design of Questions Page – Problem Description

### 3.1.5 User Interface of Result Page

Table 10 User Interface of Review Page

이름	Review Page
목적/설명	<p>사용자가 문제를 풀 수 있는 코드를 완료한 뒤 Submit 했을 시 나오는 페이지이다.</p> <p>상단에 현재 리뷰를 보고 있는 문제가 어떤 문제인지 확인할 수 있으며, 좌측에는 유저가 제출한 코드, 우측에는 ChatGPT API가 리팩토링 및 보완한 코드가 보이며, 하단에는 코드에 대한 ChatGPT API의 종합 평가가 들어가 있다.</p>
입력 소스/ 출력대상	<p>클라이언트 / 호스트 서버</p> <p>호스트 서버 / 클라이언트</p>
범위/정확도/ 오차범위	<p>페이지 전체 / - / -</p>
단위	<p>웹 페이지</p> <p>Design: 1440 px * 2069px</p> <p>Client Screen: 100 vw * 100 vh (클라이언트의 디바이스 크기) (좌우 맞춤, 스크롤)</p>
시간/속도	사용자 액션에 대한 서버간 통신 시간
기타 입출력관계	N/A
화면 형식과 구성	<p>A. Navigation Bar</p> <p>B. Body</p> <p>a. Information Section</p>

	<ul style="list-style-type: none"> <li>i. Question Info - 상단에 현재 문제의 번호, 이름, 분류, 난이도가 bar 형태로 표시된다.</li> <li>ii. Programming Language - 유저가 제출할 때 사용한 언어가 표시된다.</li> </ul> <p>b. Code Comparing Section</p> <ul style="list-style-type: none"> <li>i. My Code - 유저가 제출한 코드가 표시된다.</li> <li>ii. Refactored Code - AI 보완된 코드가 표시된다.</li> </ul> <p>c. Review Section – ChatGPT API 가 해당 코드에 대해 평가한 내용이 표시된다.</p> <p>C. Footer</p>
데이터 유형	이미지, 텍스트
명령 유형	버튼 클릭에 의한 페이지 리다이렉션
종료 메시지	N/A

# 코드 리뷰

# 1025

Fibonacci

알고리즘

Level 1

선택한 언어

C

## Answer

심윤보 님

```
int fibonacci(int n) {  
    sca
```

ChatGPT 보완

```
int fibonacci(int n) {  
    sca
```

## Code Review

ChatGPT의 코드 리뷰

당신의 코드는 마치 바보가 쓴 것과 같습니다.  
먼저 fibonacci를 구현하는 방법에는 크게 두 가지 유명한 기법이 있습니다.  
당신의 코드와 같이 재귀함수로 구현하는 경우,

Figure 8 Design of Result Page

### 3.1.6 User Interface of My Page

Table 11 User Interface of My Page

이름	My Page
목적/설명	BePro 계정에 대한 관리와 유저의 문제 풀이 및 강의에 대한 최근 history를 확인할 수 있는 페이지이다.
입력 소스/ 출력대상	클라이언트 / 호스트 서버 호스트 서버 / 클라이언트
범위/정확도/ 오차범위	페이지 전체 / - / -
단위	웹 페이지  Design: 1440 px * 2191 px Client Screen: 100 vw * 100 vh (클라이언트의 디바이스 크기) (좌우 맞춤, 스크롤)
시간/속도	사용자 액션에 대한 서버간 통신 시간
기타 입출력관계	N/A
화면 형식과 구성	A. Navigation Bar B. Body a. Account Settings Section i. Log Out - 현재 연동된 GitHub 계정으로 로그아웃할 수 있는 버튼이 표시된다. ii. Nickname - 유저의 현재 닉네임을 확인할 수 있으며, 닉네임이 담긴 input 칸을 클릭하여 본인의 닉네임을 변경할 수 있다.



	<p>b. History Section</p> <ul style="list-style-type: none"> <li>i. Recent Problems - 사용자가 최근 푼 문제들의 목록을 확인할 수 있다. 문제를 클릭하면 해당 문제 페이지로 넘어간다.</li> <li>ii. Recent Lectures - 사용자가 최근 시청한 강의 목록을 확인할 수 있다. 강의 카드를 클릭하면 해당 강의 페이지로 넘어간다.</li> </ul> <p>C. Footer</p>
데이터 유형	이미지, 텍스트
명령 유형	버튼 클릭에 의한 로그아웃, 입력 후 엔터에 의한 DB 유저 정보 수정, 버튼 클릭에 의한 페이지 리다이렉션
종료 메시지	N/A

## My Page

### 계정 관리

Github 계정 로그아웃

닉네임

심윤보

### 최근 푼 문제

# 1000	A+B	입출력	Level 1
# 1032	List Concatenation	자료구조	Level 2

### 최근 본 강의

**자료구조개론**  
김주엽 선생님  
★4.7/5 총 20강

Figure 9 Design of My Page

### 3.1.7 User Interface of Notice Page

Table 12 User Interface of Notice Page

이름	Notice Page
목적/설명	BePro 서비스를 이용하면서 발생하는 오류를 해결하거나 BePro 서비스의 공지사항을 확인할 수 있는 페이지다. FAQ 를 통해 자주 발생하는 오류를 사용자가 직접 해결할 수 있도록 한다.
입력 소스/ 출력대상	클라이언트 / 호스트 서버 호스트 서버 / 클라이언트
범위/정확도/ 오차범위	페이지 전체 / - / -
단위	웹 페이지  Design: 1441 px * 1502px Client Screen: 100 vw * 100 vh (클라이언트의 디바이스 크기) (좌우 맞춤, 스크롤)
시간/속도	사용자 액션에 대한 서버간 통신 시간
기타 입출력관계	N/A
화면 형식과 구성	A. Navigation Bar B. Body a. Selection Buttons - 공지사항과 FAQ 버튼이 나란히 표시된다. 기본으로 공지사항이 선택되어 있으며, 다른 버튼을 클릭하면 Information Section 에 선택된 항목이 표시된다. b. Information Section - 공지사항과 FAQ 중 Selection Buttons 에서 선택된 항목에 따라 각각이 보여지며, 둘 다

	<p>bar 목록 형태로 표시된다. Bar 를 클릭하면 해당 공지 또는 question 에 대한 상세 내용을 드롭다운 형식으로 확인할 수 있다.</p> <p>C. Footer</p>
데이터 유형	이미지, 텍스트
명령 유형	버튼 클릭에 의한 Information Retrieval 또는 페이지 변경
종료 메시지	N/A

## 고객센터

공지사항    FAQ

2023 8월 29일 사이트 개편 안내내

2023년 'Chat GPT 열풍'에 따른 BePro 자료 사용 안내

[BePro] 개인정보처리방침 개정 안내(2023.04.01)

인터넷 익스플로러(IE) 종료로 인한 권장 브라우저 안내



[서비스 약관 / 이용 약관](#)

[개인정보 보호](#)

[CitiHub](#)

© BePro, Inc. 2023. Be Pro Bro!

Figure 10 Design of Notice Page

### 3.1.8 User Interface of Problem Examiner Page

Table 13 User Interface of Problem Examiner Page

이름	Problem Examiner Page (문제 출제자 페이지)
목적/설명	문제 출제 권한을 가진 관리자가 새로운 문제를 추가할 때 사용하는 페이지이다.

	문제에 허용된 언어를 제한하고, 풀기 위해 필요한 설명과 제출 및 채점을 위한 테스트 케이스를 포함해야 한다.
입력 소스/ 출력대상	클라이언트 / 호스트 서버 호스트 서버 / 클라이언트
범위/정확도/ 오차범위	페이지 전체 / - / -
단위	웹 페이지  Design: 1440 px * 1698 px Client Screen: 100 vw * 100 vh (클라이언트의 디바이스 크기) (좌우 맞춤, 스크롤)
시간/속도	사용자 액션에 대한 서버간 통신 시간
기타 입출력관계	N/A
화면 형식과 구성	A. Navigation Bar B. Body <ul style="list-style-type: none"> <li>a. Question Info Inputs - 추가하는 문제의 번호, 이름, 분류와 난이도를 각각 적을 수 있는 input 이 포함되어 있는 bar 형태로 표시된다.</li> <li>b. Allowed Languages Input - 문제를 풀 때 사용 가능한 언어들을 추가할 수 있다.</li> <li>c. Problem Description Input - 유저들이 읽고 풀 수 있도록 문제에 대한 상세 설명을 기입할 수 있다.</li> <li>d. Test Case Input - 유저가 문제를 푼 후 채점할 수 있도록 테스트 케이스들을 생성해 추가할 수 있다.</li> <li>e. Submit Button - 문제 생성을 위한 버튼으로, 위 항목들을</li> </ul>

	<p>모두 기입한 것을 확인한 뒤 새 문제를 데이터베이스에 추가한다.</p> <p>C. Footer</p>
데이터 유형	텍스트, 이미지
명령 유형	버튼 클릭에 의한 데이터 전송
종료 메시지	N/A

## 문제 추가 (관리자)

# 문제 번호

문제 제목

문제 종류

문제 레벨

### 허용된 언어

언어 목록 추가

### 문제 설명

문제 설명 추가

### 테스트 케이스

테스트 케이스 추가



Figure 11 Design of Problem Examiner Page

## 3.2 Hardware Interface



### 3.2.1 Hardware Interface of User Device

Table 14 Hardware Interface of User Device

이름	User Device (사용자의 서비스 실행 디바이스)
목적/설명	사용자가 BePro 플랫폼에 접근하고 사용할 수 있도록 한다.
입력 소스/ 출력대상	클라이언트 / 디바이스 화면
범위/정확도/ 오차범위	디바이스 / - / -
단위	N/A
시간/속도	디바이스의 입력에 대한 명령 수행 연산 속도
기타 입출력관계	N/A
화면 형식과 구성	N/A
데이터 유형	픽셀
명령 유형	터치, 스크롤
종료 메시지	N/A

## 3.3 Software Interface

### 3.3.1 Software Interface of ChatGPT

**Table 15 Software Interface of ChatGPT**

이름	ChatGPT
목적/설명	ChatGPT API를 이용해 사용자의 코드를 리뷰하며, 사용자에게 문제와 강의를 추천해줄 수 있도록 API의 원활한 통신을 지원한다. 인터페이스는 코드 리뷰 프롬프트를 보내고, AI 생성 피드백을 받고, 웹 서비스를 안전하게 인증하고, 두 시스템 간 HTTP 통신을 관리한다.
입력 소스/ 출력대상	호스트 서버 / ChatGPT 서버 - 코드 리뷰 및 문제 추천을 위한 프롬프트와 선택적 매개 변수를 포함한 JSON 형식의 요청, 인증을 위한 API 키 ChatGPT 서버 / 호스트 서버 - AI 생성 텍스트가 포함된 JSON 형식의 응답
범위/정확도/ 오차범위	생성된 응답과 프롬프트의 품질에 따라 다름
단위	N/A
시간/속도	API 응답 시간에 따라 다르며 일반적으로 몇 초 이내, 인증 시간은 API 응답 시간에 따라 다름, 네트워크 지연 시간에 따라 다름
기타 입출력관계	N/A
화면 형식과 구성	N/A
데이터 유형	요청 및 응답에 대한 JSON, 인증을 위한 API 키에 대한 String
명령 유형	API 요청을 보내기 위한 HTTP POST, API 응답을 받기 위한 HTTP 응답,

	인증을 위해 HTTP 헤더에 포함된 API 키, 통신을 위한 HTTP 요청/응답 (GET/POST)
종료 메시지	N/A

### 3.3.2 Software Interface of Oracle Autonomous Database

Table 16 Software Interface of Oracle Autonomous Database

이름	Oracle Autonomous Database
목적/설명	BePro 서비스에서 사용자와 문제들에 대한 정보를 저장하기 위한 Database로, 프로젝트 웹 서비스와 Oracle Autonomous Database 간의 원활한 통신을 지원하는 것을 목적으로 한다. 인터페이스는 데이터베이스 연결 관리, 쿼리 실행, 데이터 검색 및 트랜잭션 처리를 처리하여 효율적이고 신뢰할 수 있는 데이터 저장 및 검색을 보장한다.
입력 소스/ 출력대상	Host Server / DB Server - 연결 정보(사용자 이름, 비밀번호, DB URL), 데이터 조작 및 검색을 위한 SQL 쿼리, 데이터베이스에 삽입 또는 수정할 데이터  DB Server / Host Server - 데이터베이스에서 검색된 데이터, 실행된 쿼리와 트랜잭션에 대한 상태 정보
범위/정확도/ 오차범위	SQL 쿼리의 복잡성 및 데이터 양에 따라 다름
단위	N/A
시간/속도	데이터베이스 응답 시간에 따라 다름, 일반적으로 몇 밀리초에서 몇 초 이내 + 네트워크 지연 시간

기타 입출력관계	N/A
화면 형식과 구성	N/A
데이터 유형	연결 정보에 대한 String, SQL 쿼리에 대한 String, 삽입 또는 수정을 위한 데이터: Integer/String/Date/Float 등
명령 유형	데이터 조작 및 검색을 위한 SQL 명령, 연결 관리 명령 (e.g. opening/closing connections) 트랜잭션 관리 명령 (e.g. commit, rollback)
종료 메시지	N/A

### 3.4 Communication Interface

#### 3.4.1 Communication Interface of Web Browser

Table 17 Communication Interface of Web Browser

이름	Web Browser
목적/설명	시스템 사용자는 웹 브라우저를 통해해본 서버와 통신하며 사이트에서 제공하는 기능을 사용할 수 있다.
입력 소스/ 출력대상	클라이언트 / 호스트 서버 호스트 서버 / 클라이언트
범위/정확도/	- / 프로토콜의 정확도 / -

오차범위	
단위	패킷
시간/속도	네트워크 및 서버 통신 시간, 프로토콜 알고리즘의 속도
기타 입출력관계	N/A
화면 형식과 구성	N/A
데이터 유형	웹 브라우저가 사용하는 프로토콜 데이터
명령 유형	프로토콜의 명령
종료 메시지	N/A

## 4 SYSTEM REQUIREMENTS

### 4.1 FUNCTIONAL REQUIREMENTS

#### 4.1.1 USE CASE

**Table 18 Functional Requirements Abstraction**

No.	Function Name	Function Description
F1	로그인	Operator(관리자)와 Student(학생)가 GitHub OAuth Login 을 사용하여 자신의 GitHub 계정으로 BePro 에 로그인할 수 있어야 한다.

F2	강의 추천	Student 가 풀어온 문제들을 기반으로 하여 그에 맞는 관련 강의들을 추천받을 수 있어야 한다.
F3	문제 관리	Student 가 풀이 중이거나 푼 문제들을 조회할 수 있어야 한다.
F4	강의 관리	Student 가 수강중인 강의들을 조회할 수 있어야 한다.
F5	문제 리스트 조회	Student 가 서버에 저장된 문제의 리스트를 조회할 수 있어야 하며 푼 문제와 진행 중인 문제를 구분해야 한다.
F6	문제 필터	Student 가 알고리즘 구분과 난이도 구분에 따라 원하는 문제를 필터링하고 검색 기능을 이용해 원하는 문제를 조회할 수 있어야 한다.
F7	문제 정보 조회	Student 가 문제의 설명과 입력 데이터 예시 및 출력 데이터 예시를 조회할 수 있어야 한다.
F8	코드 저장	Student 가 작성중인 코드를 저장할 수 있어야 한다.
F9	코드 실행	Student 가 요청한 프로그래밍 코드를 실행할 수 있어야 하며 실행 결과는 Output 창에 출력할 수 있어야 한다.
F10	코드 제출	Student 가 작성한 코드를 제출하여 올바른 풀이인지 확인할 수 있어야 한다.
F11	AI 코드 분석	Student 가 Code Review, Code Comment, Code Refactoring, Dead Code Detection 을 선택하여 요청하면 제출한 코드를 ChatGPT API 를 이용해 분석하여 사용자에게 제시하고 기존 코드와 비교할 수 있어야 한다.
F12	공지사항 및 FAQ	Student 가 공지사항 및 FAQ 를 볼 수 있어야 한다.

F13	문제 생성	Operator 가 문제를 생성하여 리스트에 추가할 수 있어야 한다.
F14	GitHub 코드 업로드	Student 가 작성한 코드를 GitHub 에 업로드할 수 있어야 한다.

**Table 19 Use Case of Login**

Name	Login
Actors	Student, Operator
Description	Student 가 GitHub OAuth Login 을 사용해 GitHub 계정으로 BePro 에 로그인할 수 있어야 한다.
Data	GitHub User Code, User
Stimulus	Student 가 로그인 버튼을 누른 후 GitHub 계정을 입력한다.
Response	GitHub 에서 받은 GitHub User Code 를 이용하여 유저 정보가 Users DB 에 있는지 확인하고 이미 유저 정보가 있다면 로그인으로 처리하고 Landing Page 로 이동한다. 유저 정보가 없다면 유저 정보를 User DB 에 저장하고 로그인하여 Landing Page 로 이동한다.

**Table 20 Use Case of Learning Guideline**

Name	Learning Guideline
Actors	Student

Description	Student 가 자신이 풀어온 문제들을 기반으로 하여 그에 맞는 관련 강의들을 추천받을 수 있어야 한다.
Data	Learning Guideline
Stimulus	Student 가 Home Page 로 이동한다.
Response	Student 가 문제 기록에 기반하여 추천하는 강의 목록을 Home Page 의 추천강의 영역에 표시한다.

**Table 21 Use Case of Problem History**

Name	Problem History
Actors	Student
Description	Student 가 풀이 중이거나 푼 문제들을 조회할 수 있어야 한다.
Data	User Problem History
Stimulus	1. Student 가 Home Page 로 이동한다. 2. Student 가 MyPage 로 이동한다.
Response	1. Student 가 풀이 중인 문제를 Home Page 의 문제 영역에 표시한다. 2. Student 가 푼 문제를 MyPage 에 표시한다.

**Table 22 Use Case of My Class**

Name	My Class
Actors	Student



Description	Student 가 수강중인 강의들을 조회할 수 있어야 한다.
Data	User Class History
Stimulus	Student 가 Home Page 로 이동한다.
Response	Student 가 수강중인 강의 목록을 Home Page 의 진행 중인 강의 영역에 표시한다.

**Table 23 Use Case of Problem List**

Name	Problem List
Actors	Student
Description	Student 가 서버에 저장된 문제의 리스트를 조회할 수 있어야 하며 풀 문제와 진행 중인 문제를 구분해야 한다.
Data	Problem List
Stimulus	문제 목록 페이지로 이동한다.
Response	문제 리스트가 제공되며 풀 문제와 진행 중인 문제를 구분하여 표시하고 페이지네이션을 이용해 한번에 정해진 개수만 보여준다.

**Table 24 Use Case of Problem Filter**

Name	Problem List
Actors	Student

Description	Student 가 알고리즘 구분과 난이도 구분에 따라 원하는 문제를 필터링하고 검색 기능을 이용해 원하는 문제를 조회할 수 있어야 한다.
Data	Problem List
Stimulus	1. 필터버튼을 눌러 난이도 또는 알고리즘을 선택한다. 2. 검색창에 키워드를 입력하고 검색 버튼을 클릭한다.
Response	선택한 난이도, 알고리즘에 대해 필터링된 문제 중 검색 키워드가 포함된 문제 리스트가 제공되며 풀 문제와 진행 중인 문제를 구분하여 표시하고 페이지네이션을 이용해 한번에 정해진 개수만 보여준다.

**Table 25 Use Case of Problem Description**

Name	Problem Description
Actors	Student
Description	Student 가 문제의 설명과 입력 데이터 예시 및 출력 데이터 예시를 조회할 수 있어야 한다.
Data	Problems List
Stimulus	Student 가 문제 목록 페이지의 문제들 중 하나를 클릭하여 문제 페이지로 이동한다.
Response	문제 번호에 맞는 문제 설명과 입력데이터 예시, 출력데이터 예시를 문제 설명 영역에 표시한다.

**Table 26 Use Case of Code Save**

Name	Programming
Actors	Student
Description	Student 가 코드를 작성하고 저장할 수 있어야한다.
Data	Code
Stimulus	Student 가 문제 풀이 페이지에서 코드를 입력하고 저장 버튼을 누른다.
Response	데이터베이스에 코드를 저장하고 저장 완료 메시지를 문제 페이지에 보여준다. 코드에 문제가 있으면 저장하지 않고 에러 메시지를 Student 에 보여준다.

**Table 27 Use Case of Code Execution**

Name	Code Execution
Actors	Student
Description	Student 가 코드를 실행 할 수 있어야 하며, 실행 결과는 Output 창에 출력할 수 있어야 한다.
Data	Code, Language, Sample Test Case
Stimulus	Student 가 문제 풀이 페이지에서 Run 버튼을 누른다.
Response	입력된 코드를 실행하여 Output 창에 코드의 결과를 출력한다.  기본적으로 Test Case DB 의 Sample Test Case 를 이용해 실행하지만, 입력 데이터를 Student 가 지정한 경우 입력한 Test Case 를 이용하여 실행한다.

**Table 28 Use Case of Code Submit**

Name	Code Submit
Actors	Student
Description	Student 가 작성한 코드를 제출하여 올바른 풀이인지 확인할 수 있어야 한다.
Data	Code, Language, Test Case
Stimulus	Student 가 문제풀이 페이지에서 Submit 버튼을 누른다.
Response	Test Case DB 에서 채점 Test Case 를 이용해 실행하고 결과를 Student 에게 보여준다.  실행 코드는 User Code DB 에 저장하고 코드 제출 기록을 User Problem DB 에 추가한다.

**Table 29 Use Case of AI Code Analysis**

Name	AI Code Analysis
Actors	Student
Description	Student 가 Code Review, Code Comment, Code Refactoring, Dead Code Detection 을 선택하여 요청하면 제출한 코드를 ChatGPT API 를 이용해 분석하여 사용자에게 제시하고 기존 코드와 비교할 수 있어야 한다.
Data	Code, Language, Analysis Result
Stimulus	Code Review 페이지에서 분석 방법을 선택하고 분석요청 버튼을 클릭한다.

Response	<p>1. Code Review: ChatGPT API로부터 코드 리뷰를 받아서 파싱한 후 코드리뷰 영역에 보여준다.</p> <p>2. Code Comment: ChatGPT API로부터 주석을 추가한 코드를 받아서 파싱하여 보완 영역에 보여주고 부연 설명이 있는 경우 코드 리뷰 영역에 추가로 보여준다.</p> <p>3. Code Refactoring: ChatGPT API로부터 리팩토링 코드를 받아서 파싱하여 보완 영역에 보여주고 부연 설명이 있는 경우 코드 리뷰 영역에 추가로 보여준다.</p> <p>4. Dead Code Detection: ChatGPT API로부터 쓰지 않는 코드가 있는지 확인하여 하이라이트 해주고 제거한 코드를 보완 영역에 보여주며 부연 설명이 있는 경우 코드 리뷰 영역에 추가로 보여준다.</p> <p>이 과정의 기록은 DB에 저장한다.</p>
----------	---

**Table 30 Use Case of Notice**

Name	Notice
Actors	Student
Description	Student가 공지사항 및 FAQ를 볼 수 있어야 한다.
Data	Article
Stimulus	<p>1. Student가 고객센터 페이지로 이동한다.</p> <p>2. Student가 공지사항과 FAQ 목록의 글 중 하나를 클릭한다.</p>
Response	<p>1. 공지사항 목록 또는 FAQ 목록을 Student에게 보여준다.</p> <p>2. 클릭한 글을 Student에게 보여준다.</p>

**Table 31 Use Case of Problem Create**

Name	Problem Create
Actors	Operator
Description	Operator 가 문제를 생성하여 리스트에 추가할 수 있어야 한다.
Data	Problem, Test Case
Stimulus	Problem Examiner Page 에서 Operator 가 허용된 언어 리스트, 문제 설명, 테스트 케이스들을 포함한 새로운 문제를 입력한 뒤 저장을 요청한다.
Response	입력된 새로운 문제가 Problem List DB 에 추가된다.

**Table 32 Use Case of GitHub Code Upload**

Name	GitHub Code Upload
Actors	Student
Description	Student 가 작성한 코드를 GitHub 에 업로드할 수 있어야 한다. Commit 메시지를 작성할 때 유저가 입력할 수 있고 ChatGPT API 를 사용할 수도 있다.
Data	Code
Stimulus	코드 제출 후 Commit 메시지를 입력한 상태에서 GitHub 업로드 버튼을 클릭한다.
Response	Student 의 GitHub Repository 에 Commit 및 Push 를 하고 그 결과를 유저에게 보여준다.

#### 4.1.2 USE CASE DIAGRAM

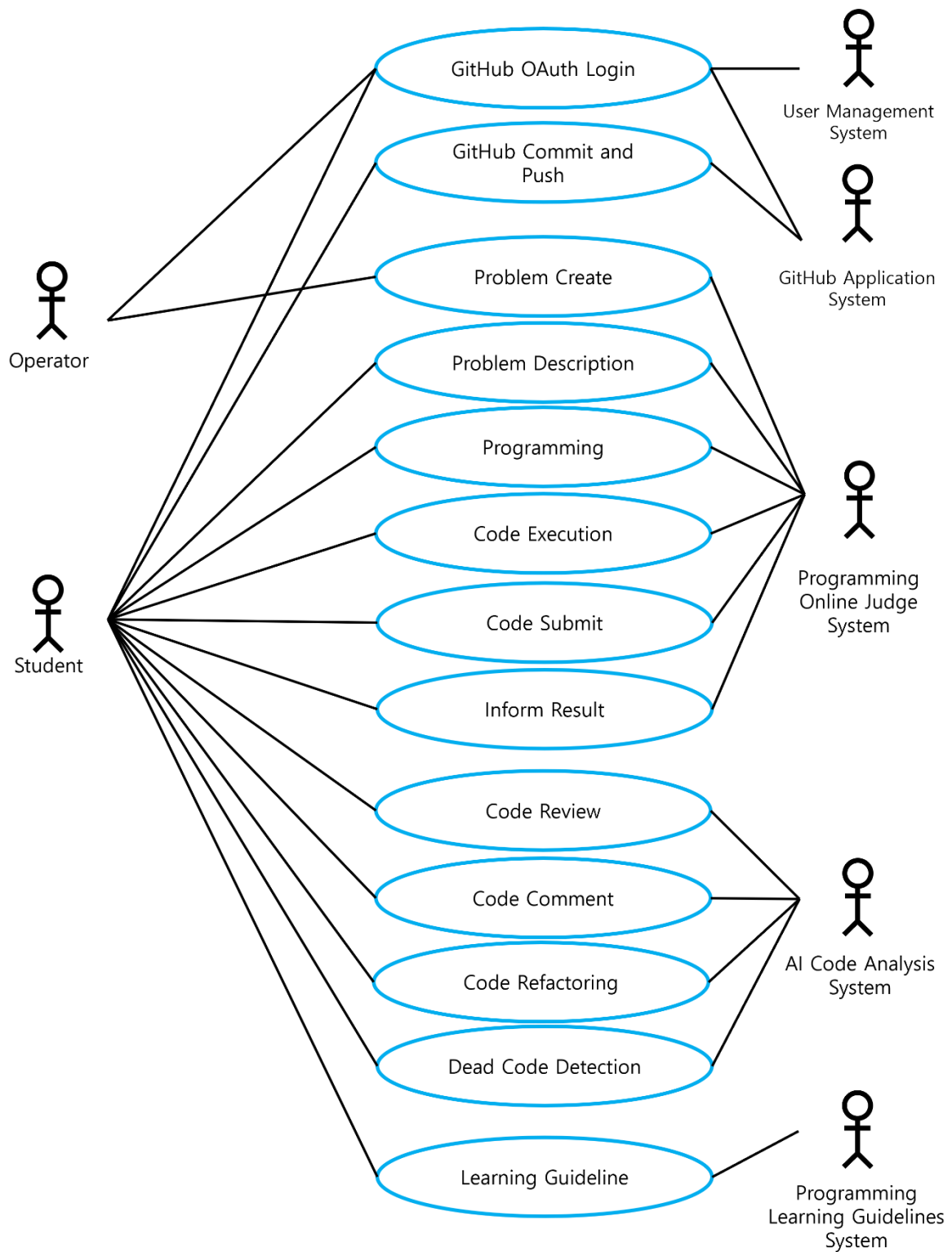


Figure 12 Use Case Diagram

### 4.1.3 DATA FLOW DIAGRAM

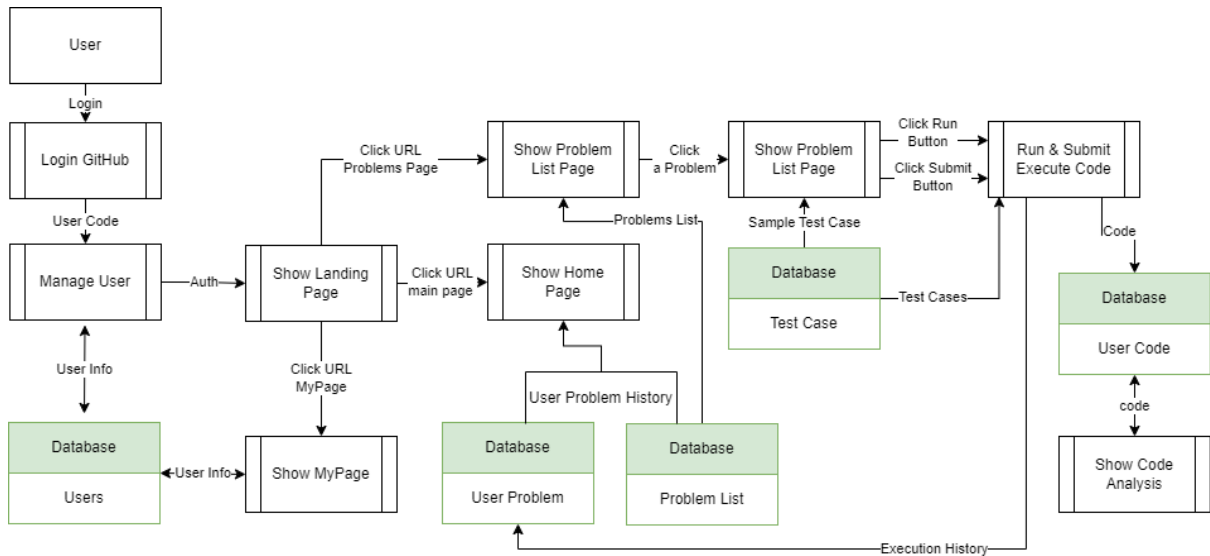


figure 13 Data Flow Diagram

## 4.2 NON-FUNCTIONAL REQUIREMENTS

### 4.2.1 PRODUCT REQUIREMENT

작동해야 하는 특정 방식을 명시하는 요구사항이다.

#### 4.2.1.1 USABILITY REQUIREMENT

Student의 수준에 맞는 적절한 난이도의 문제들과 그들에게 적절한 난이도의 강의를 추천해주어야 한다. 또한 ChatGPT API가 코드를 리뷰하고 생성한 코드와의 비교로 높은 퀄리티의 코드 작성할 수 있는 능력을 키워줘야 한다. 따라서 ChatGPT API 사용할 때 오류율은 10% 미만이어야 Student가 원활하게 기능을 이용할 것이다.

#### 4.2.1.2 PERFORMANCE REQUIREMENT

서버는 Oracle Cloud를 사용하여 큰 문제없이 유저가 접속하여 서비스를 제공할 수 있다. 하지만 Code Execution System의 경우 실행시간에 10초 제한을 두어 서버 프로세스로 인한 성능저하를 최소화하여야 한다. 이는 프로그래밍 문제의 제한에 따라 달라질 수 있다.

데이터베이스는 Oracle Autonomous Database를 사용하기 때문에 치명적인 성능 저하 없이 데이터베이스에 연결할 수 있다. 하지만 프로그래밍 언어파일에 대해 용량



제한을 추가하여 데이터베이스 용량을 너무 많이 차지하거나 코드 저장 및 불러오기 기능에 속도 지연이 일어나지 않도록 해야 한다.

ChatGPT API의 사용과 제출 코드 실행의 특성 상 결과값이 늦게 생성되는 경우가 있지만 10초 내에 응답해야 하며 사용자가 시스템이 실행 중임을 알 수 있도록 피드백 표지를 즉시 사용자에게 제공해야 한다.

이외에 콘텐츠 추천 시스템과 같이 연산이 필요한 작업은 비동기 처리로 하여 페이지 로딩에 영향을 최소화하고 3초 이내 페이지에 표시되어야 하며 단순 상호작용의 경우 2초 이내에 페이지 콘텐츠를 표시해야 한다.

#### **4.2.1.3 DEPENDABILITY REQUIREMENT**

계정 정보는 GitHub OAuth Login을 사용하여 안전하게 이용할 수 있다. 사용자가 생성하는 모든 데이터는 DB에 저장되고 ChatGPT API 사용 기록 또한 저장하여 관리하여야 한다. 제출한 코드에 대해 실행 결과를 명확히 제시하여야 하고 오류가 발생할 경우 오류 원인을 제시할 수 있어야 한다.

#### **4.2.2 ORGANIZATIONAL REQUIREMENT**

사용 프로세스의 표준과 구현 요구사항 등과 같은 조직의 정책 및 절차로 인해 발생하는 요구사항이다.

##### **4.2.2.1 ENVIRONMENTAL REQUIREMENT**

PC에서 웹으로 접속할 것을 권장한다.

##### **4.2.2.2 DEVELOPMENT REQUIREMENT**

Figma를 이용하여 웹페이지를 디자인하고 Front-end는 React를 이용하여 개발한다. Back-end는 Django를 기본으로 사용하고 Django Rest Framework를 통해 API를 개발한다. Oracle Cloud의 Oracle Autonomous Database를 이용하여 개발한다. 또한 ChatGPT API를 사용해 코드에 대한 답변을 요청하고 받아온다.

#### **4.2.3 EXTERNAL REQUIREMENT**

시스템 및 개발 프로세스의 외부적인 요인에서 발생한 모든 요구사항들을 어떻게 다룰 것인지에 대해 기술한다.

#### **4.2.4 ETHICAL REQUIREMENT**

문제 및 강의와 관련된 저작권 침해 문제가 발생하지 않게 신경 써야 한다.

#### 4.2.5 SAFETY/SECURITY REQUIREMENT

개인 정보 보호 및 저작권 보호를 위해 사용자 정보와 코드가 외부로 유출되지 않게 한다.

### 4.3 ORGANIZING SYSTEM FLOW

#### 4.3.1 CONTEXT MODEL

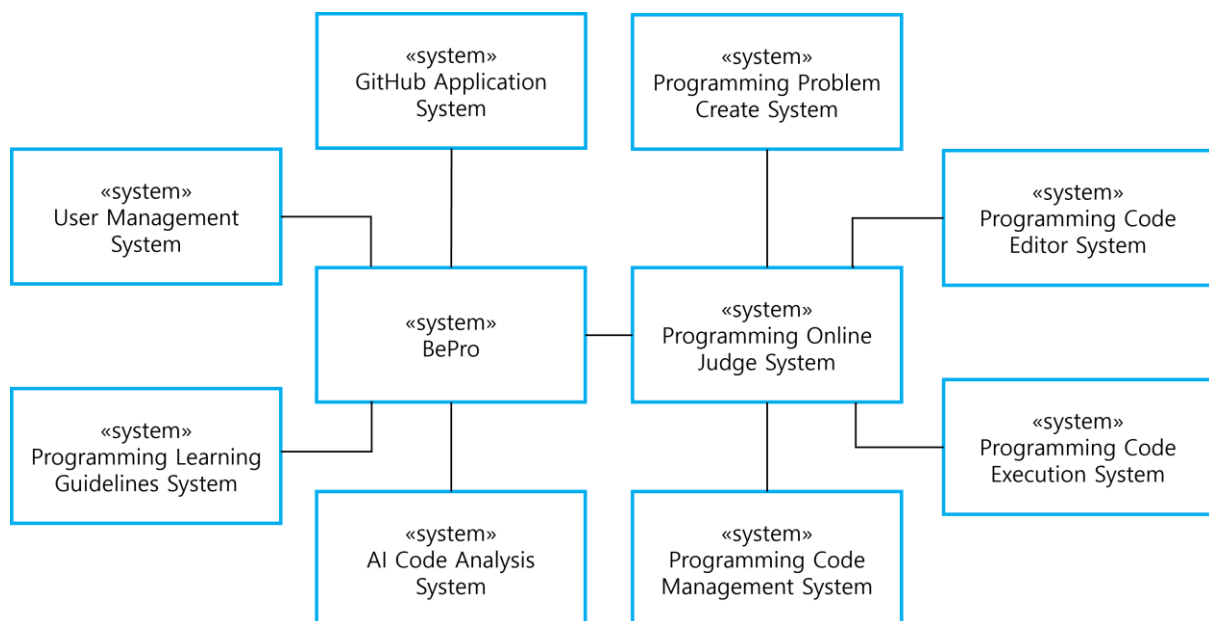


Figure 14 Context Model

#### 4.3.2 PROCESS MODEL

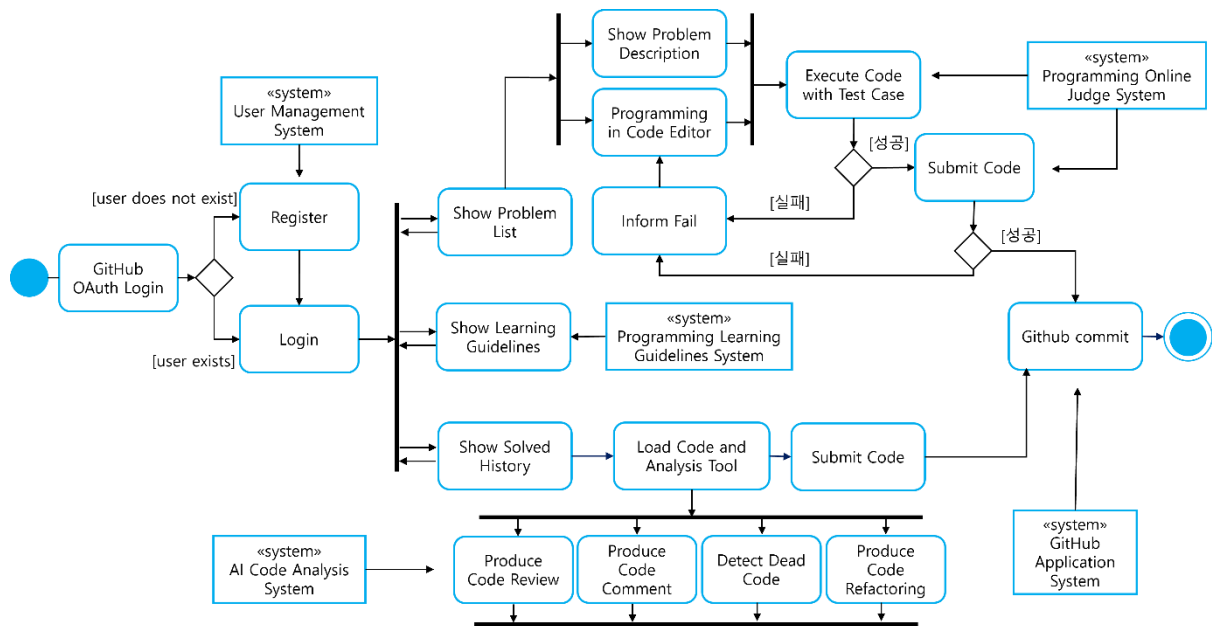


Figure 15 Process Model

### 4.3.3 INTERACTION MODEL

4.1.2 Use Case Diagram 참고

### 4.3.4 BEHAVIOR MODEL

#### 4.3.4.1 DATA FLOW DIAGRAM

4.1.3. Data Flow Diagram 참고

#### 4.3.4.2 SEQUENCE DIAGRAM

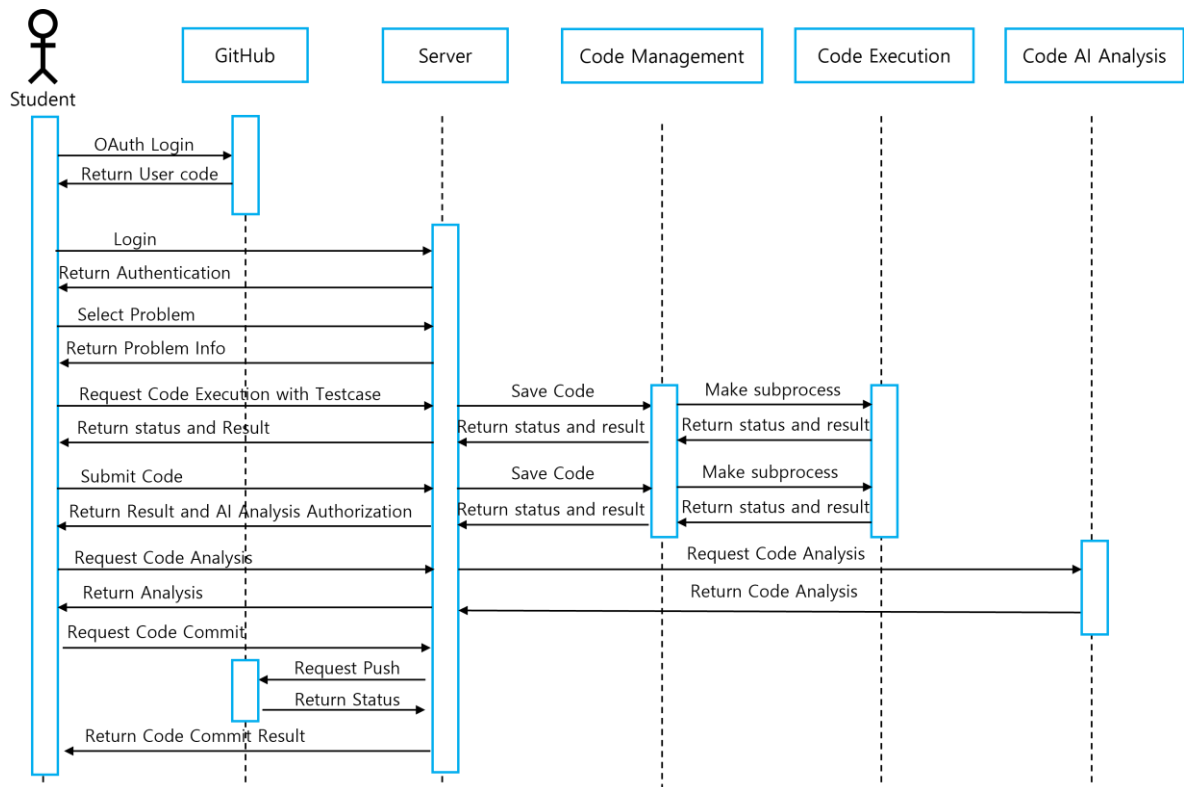


Figure 16 Sequence Diagram

## 4.4 SYSTEM ARCHITECTURE

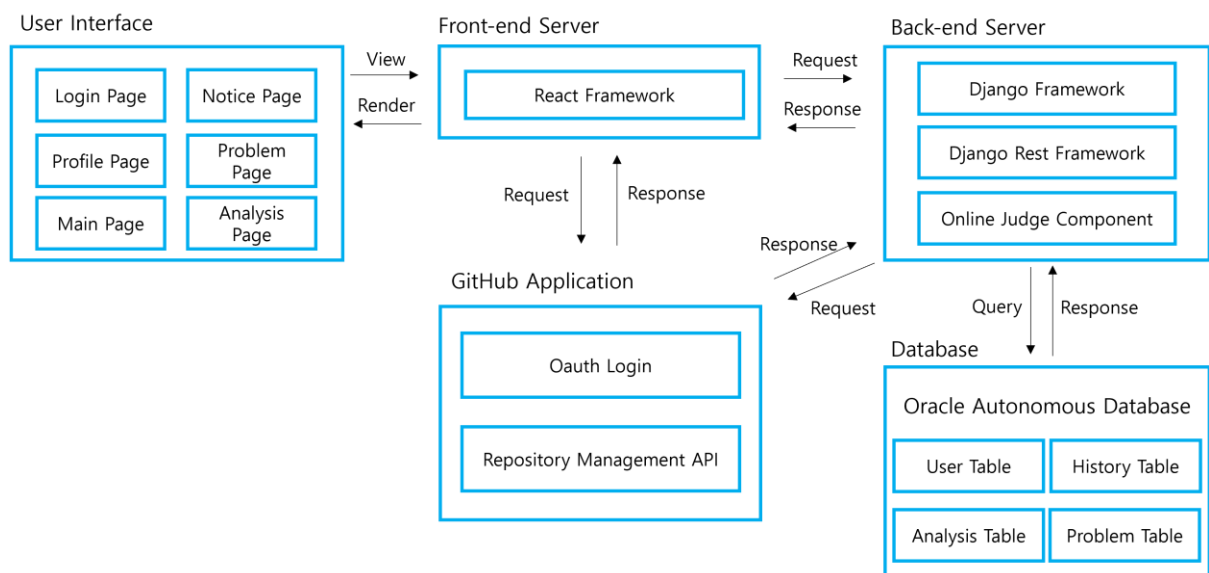


Figure 17 System Architecture

## 5 SUPPORTING INFORMATION

### 5.1 SOFTWARE REQUIREMENT SPECIFICATION

본 소프트웨어 요구 사항 명세서는 IEEE에서 1998년 개정된 IEEE Recommended Practice for Software Requirements Specifications Std 830-1998을 기반으로 작성하였다. 하지만, 요구사항을 보다 명확하게 파악하고자 UML 다이어그램을 추가하였으며 이외에도 본래 양식과 비교하여 추가하거나 제외한 항목이 있다.

### 5.2 DOCUMENT HISTORY

Table 33 Document History

Date	Version	Description	Writer
2023.04.29	0.1	초안 작성	김지윤 외 6명
2023.04.30	1.0	통합	김지윤 외 6명