

Design Specification

Coding & Ethics Games with AI

로고, 상징, 폰트, 그래픽이(가) 표시된 사진

자동 생성된 설명

2023 May 21

소프트웨어공학개론 8조

2017310811오승재  
2017312355서민석  
2018310812조예성  
2018312567조명하  
2018314951백일웅  
2020314978강현서

CONTENTS

[1. Preface 6](#_Toc135530266)

[1.1. Objective 6](#_Toc135530267)

[1.2. Readership 6](#_Toc135530268)

[1.3. Document Structure 6](#_Toc135530269)

[1.3.1. Preface 6](#_Toc135530270)

[1.3.2. Introduction 6](#_Toc135530271)

[1.3.3. System Architecture 6](#_Toc135530272)

[1.3.4. User Management System 6](#_Toc135530273)

[1.3.5. Ethics Quiz System 7](#_Toc135530274)

[1.3.6. AI Competition coding game System 7](#_Toc135530275)

[1.3.7. Protocol Design 7](#_Toc135530276)

[1.3.8. Database Design 7](#_Toc135530277)

[1.3.9. Testing Plan 7](#_Toc135530278)

[1.3.10. Development Environment 7](#_Toc135530279)

[1.3.11. Development Plan 7](#_Toc135530280)

[1.3.12. Index 8](#_Toc135530281)

[1.3.13. References 8](#_Toc135530282)

[1.4. Version of the Document 8](#_Toc135530283)

[2. Introduction 9](#_Toc135530284)

[2.1. Objective 9](#_Toc135530285)

[2.2. Applied Diagram 9](#_Toc135530286)

[2.2.1. UML 9](#_Toc135530287)

[2.2.2. Use-case Diagram 10](#_Toc135530288)

[2.2.3. Class Diagram 11](#_Toc135530289)

[2.2.4. State Diagram 12](#_Toc135530290)

[2.2.5. Sequence Diagram 13](#_Toc135530291)

[2.2.6. E-R Diagram 14](#_Toc135530292)

[2.3. Applied Tool 15](#_Toc135530293)

[2.3.1. Visual Studio Code 15](#_Toc135530294)

[2.3.2. draw.io 15](#_Toc135530295)

[2.3.3. Smartdraw 16](#_Toc135530296)

[2.4. Project Scope 16](#_Toc135530297)

[2.4.1. Overview 16](#_Toc135530298)

[2.4.2. User Management System 16](#_Toc135530299)

[2.4.3. Ethics Quiz System 16](#_Toc135530300)

[2.4.4. AI Competition coding game System 17](#_Toc135530301)

[3. System Architecture 18](#_Toc135530302)

[3.1. Objective 18](#_Toc135530303)

[3.2. System Organization 18](#_Toc135530304)

[3.2.1. Overall System Architecture 18](#_Toc135530305)

[3.2.2. User Management System 19](#_Toc135530306)

[3.2.3. Problem Selection System 19](#_Toc135530307)

[3.2.4. Ethics Quiz System 20](#_Toc135530308)

[3.2.5. AI Competition coding game System 20](#_Toc135530309)

[4. User Management System 21](#_Toc135530310)

[4.1. Objective 21](#_Toc135530311)

[4.2. Class Diagram 21](#_Toc135530312)

[4.2.1. DB Handler 21](#_Toc135530313)

[4.2.2. Account 22](#_Toc135530314)

[4.3. Sequence Diagram 23](#_Toc135530315)

[4.3.1. Register and Login 23](#_Toc135530316)

[4.4. State Diagram 24](#_Toc135530317)

[4.4.1. Register 24](#_Toc135530318)

[4.4.2. Login 24](#_Toc135530319)

[5. Ethics Quiz System 25](#_Toc135530320)

[5.1. Objective 25](#_Toc135530321)

[5.2. Class Diagram 25](#_Toc135530322)

[5.3. Sequence Diagram 27](#_Toc135530323)

[5.4. State Diagram 28](#_Toc135530324)

[5.4.1. User 28](#_Toc135530325)

[5.4.2. Website 28](#_Toc135530326)

[6. AI Competition Coding Game System 30](#_Toc135530327)

[6.1. Objective 30](#_Toc135530328)

[6.2. Class Diagram 30](#_Toc135530329)

[6.3. Sequence Diagram 32](#_Toc135530330)

[6.4. State Diagram 33](#_Toc135530331)

[7. Protocol Design 36](#_Toc135530332)

[7.1. Objective 36](#_Toc135530333)

[7.2. Json 36](#_Toc135530334)

[7.3. Protocol Description 36](#_Toc135530335)

[7.3.1. Overview 36](#_Toc135530336)

[7.3.2. Login Protocol 37](#_Toc135530337)

[7.3.3. Register Protocol 37](#_Toc135530338)

[7.3.4. Select Problem Type Protocol 38](#_Toc135530339)

[7.3.5. Select Challenge Problem Protocol 38](#_Toc135530340)

[7.3.6. Challenge Problem Submission Protocol 38](#_Toc135530341)

[7.3.7. Select Ethical Problem Protocol 39](#_Toc135530342)

[7.3.8. Ethical Problem Submission Protocol 39](#_Toc135530343)

[7.3.9. Create Problem Protocol 40](#_Toc135530344)

[8. Database Design 41](#_Toc135530345)

[8.1. Objective 41](#_Toc135530346)

[8.2. E-R Diagram 41](#_Toc135530347)

[8.3. Relation Schema 46](#_Toc135530348)

[9. Testing Plan 47](#_Toc135530349)

[9.1. Objective 47](#_Toc135530350)

[9.2. Testing Policy 47](#_Toc135530351)

[9.2.1. Developing Testing 47](#_Toc135530352)

[9.2.2. Release Testing 47](#_Toc135530353)

[9.2.3. User Testing 47](#_Toc135530354)

[9.3. Test Case 48](#_Toc135530355)

[9.3.1. User Management System 48](#_Toc135530356)

[9.3.2. Ethics Quiz System 51](#_Toc135530357)

[9.3.2.1 윤리 퀴즈 문제 리스트 확인 51](#_Toc135530358)

[9.3.3. AI Competition coding game System 53](#_Toc135530359)

[10. Development Environment 56](#_Toc135530360)

[10.1. Objective 56](#_Toc135530361)

[10.2. Django 56](#_Toc135530362)

[10.3. React 57](#_Toc135530363)

[10.4. GitHub 58](#_Toc135530364)

[11. Development Plan 59](#_Toc135530365)

[11.1. Objective 59](#_Toc135530366)

[11.2. Gantt Chart 59](#_Toc135530367)

[12. Index 60](#_Toc135530368)

[12.1. Objective 60](#_Toc135530369)

[12.2. Figure Index 60](#_Toc135530370)

[12.3. Diagram Index 60](#_Toc135530371)

[12.4. Table Index 62](#_Toc135530372)

[13. References 63](#_Toc135530373)

[13.1. Objective 63](#_Toc135530374)

[13.2. References 63](#_Toc135530375)

# Preface

## Objective

Preface에서는 본 문서의 독자를 정의하고, 문서의 전체적인 구조 및 각 항목의 내용에 대해 간략히 기술한다. 문서의 구조 소개 시에는 각 항목의 목적과 개요를 명시한다.

## Readership

본 문서는 독자를 코딩 및 윤리 게임을 개발 및 유지 보수하는 소프트웨어 엔지니어, 시스템 구조를 설계하는 시스템 아키텍처와 고객 시술지원을 위한 팀으로 설정한다.

## Document Structure

### Preface

본 문서의 독자 정의 및 문서의 전반적인 구조와 각 항목의 내용을 간략히 소개한다.

### Introduction

시스템의 설계에 사용한 UML과 데이터베이스 모델의 다양한 다이어그램 작성을 위해 사용한 개발 툴을 소개한다.

### System Architecture

개발되어지는 시스템에 대한 상세한 개요와 시스템기능의 전체적 분포에 대해 기술한다. 전체 시스템의 구조는 Block Diagram을 이용하여 표현한다.

### User Management System

사용자 관리 시스템은 사용자 정보 데이터베이스와 연동되며, 회원가입, 로그인, 사용자의 해결한 문제 관리를 담당한다. 회원가입과 로그인의 두개의 sub-system으로 구성된다. 이 절에서는 User Management System의 구조를 모식화 하고 시스템의 사용자 및 데이터베이스와의 상호작용을 설계한다.

### Ethics Quiz System

윤리 교육을 위한 퀴즈 system이다. 이 system은 database에 저장되어 있는 윤리 퀴즈 문제와 각 문제에 대한 사용자의 진행상황을 불러온다. Sub-system은 Website, Database, Problem, ChatGPT의 4개이다. 이 윤리 퀴즈 system에 대해 Class diagram, Sequence diagram, State diagram을 통해 윤리 퀴즈 system의 구조를 표현하고 상세 내용을 기술한다.

### AI Competition coding game System

AI 경쟁 코딩 게임 system의 설계에 대해 설명한다. 이 System은 database에 저장되어 있는 난이도별 코딩 문제와 각 문제에 대한 User의 진행상황을 불러온다. Sub-system은 Website, Database, Problem, ChatGPT의 4 개이다. AI 경쟁 코딩 게임 system에 대해 Class diagram, Sequence diagram, State diagram을 통해 AI 경쟁 코딩 게임 system의 구조를 표현하고 상세 내용을 기술한다.

### Protocol Design

Sub-System들의 상호작용에서 필수적으로 준수해야 하는 프로토콜에 대해 기술한다. 각 프로토콜의 메시지 형식, 용도, 의미를 설명한다.

### Database Design

E-R 다이어그램을 통해 개체 관계와 개체 속성에 대해 설명하고, 릴레이션 스키마와 데이터 정의, 접근양식 등 데이터베이스 설계의 전반적인 내용을 구체적으로 기술한다.

### Testing Plan

개발되는 시스템의 테스트 정책 및 수행을 위한 설계에 대해 기술합니다. 각각의 테스트 계획에서는 테스트 정책 및 테스트케이스를 설명한다.

### Development Environment

실제 시스템 개발을 위해 필요한 개발 환경과 코딩 규칙에 대해 기술한다. 이외에 개발과정에서의 버전관리 도구 및 프로그래밍 과정에서의 기반이 되는 규칙에 대해 기술한다.

### Development Plan

시스템 개발 일정을 기술한다. Gantt Chart를 사용하여 전체적 개발 순서 및 시기를 나타낸다.

### Index

그림, 표, 다이어그램의 인덱스이다.

### References

문서 작성시 참고한 참고문헌의 목록이다.

## Version of the Document

|  |  |  |
| --- | --- | --- |
| Version | Modified Date | Explanation |
| 0.1 | 2023.05.15 | 문서의 목차 작성 |
| 1.0 | 2023.05.16 | 문서의 1, 2, 10, 11, 12, 13초안 작성 |
| 2.0 | 2023.05.18 | 문서의 4, 5, 7, 절 초안 작성 |
| 3.0 | 2023.05.19 | 문서의 3, 8, 9절 초안 작성 |
| 3.1 | 2023.05.20 | 문서의 전체 서식 통일 |
| 3.2 | 2023.05.21 | 1절 요약 내용 일부 수정 및 참고문헌 수정 |

Table Document Version History

# Introduction

## Objective

이 절에서는 시스템의 설계에 사용한 UML(Unified Modeling Language)과 데이터베이스 모델의 다양한 다이어그램 작성을 위해 사용한 개발 도구들을 소개한다

## Applied Diagram

### UML

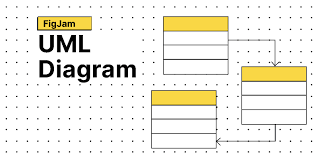


Figure Concept of UML

UML(Unified Modeling Language, 통합 모델링 언어) 은 요구분석, 시스템 설계, 시스템 구현 등의 시스템 개발 과정에서 개발자 간 의사소통을 원활하게 이루어지게 하기 위하여 표준화한 모델링 언어이다. 시스템의 시각적인 모델을 표현하며, 객체 지향 소프트웨어 시스템을 개발할 때, 산출물을 가시화, 명세화, 문서화할 때 사용한다.

UML의 관리는 처음 고안한 OMG(Object Management Group)에서 진행하며 소프트웨어 산업 내 시각화의 실질적인 표준 역할을 하고 있다. UML을 기반으로 UML Diagram이 존재하며, 현재 13개의 종류의 Diagram이 존재한다.

본 문서에서 시스템 표현을 위해 사용할 UML Diagram은 Use Case Diagram, Class Diagram, Sequence Diagram, State Diagram 4가지와, 데이터베이스를 표현할 ER Diagram이다.

### Use-case Diagram

텍스트, 도표, 스케치, 그림이(가) 표시된 사진

자동 생성된 설명

Figure Example of Use-Case Diagram

Use-Case 다이어그램은 사용자를 포함하는 Actor와 Use Case간의 관계를 모식화 하여 나타낸 사용자-시스템간 상호작용의 표현이다. Use-Case 다이어그램은 각기 다른 종류의 시스템 사용자와 각기 다른 Use-Case를 식별할 수 있으며 다른 유형의 다이어그램을 수반하기도 한다. Use-Case는 원 또는 타원으로 표현한다.

### Class Diagram

텍스트, 스크린샷, 도표, 평행이(가) 표시된 사진

자동 생성된 설명

Figure Example of Class Diagram

Class Diagram 시스템의 정적인 구조를 설명하고 시스템의 논리적 구조를 표현하는 다이어그램이다. 하나의 클래스는 직사각형의 모양이다. 위에서부터 순서대로 클래스 이름, 속성(attributes), 그리고 함수(method, operation)이 포함된다. 각 클래스 간 상속뿐 아니라 다양한 관계를 표현할 수 있다.

### State Diagram

도표이(가) 표시된 사진

자동 생성된 설명

Figure Example of State Diagram

State Diagram은 시스템의 상태를 표현하거나 특정 순간의 시스템의 부분을 기술하는 다이어그램으로 Behavioral Diagram의 하나이다. 외부 자극의 변화나 시간에 따라 각 객체 혹은 클래스의 동적인 행동과 상태를 기술한다.

각 다이어그램은 일반적으로 단일 클래스의 객체를 표현하고 시스템을 통해 해당 객체의 여러 상태를 추적할 수 있다.

### Sequence Diagram

텍스트, 도표, 평행, 라인이(가) 표시된 사진

자동 생성된 설명

Figure Example of Sequence Diagram

Sequence Diagram은 시간 흐름에 따라, 시스템, 객체, 그리고 클래스 간의 상호작용을 기술한다. 시나리오와 관련된 객체와 클래스를 설명하고 시나리오의 functionality가 진행될 때 객체간 교환하는 메시지의 결과들을 기술한다. Diagram은 개발 과정에서 시스템의 논리적 구조 내에서 use case들과 관련된다.

Sequence Diagram은 lifeline이라고 불리는 평행의 수직 직선 내에, 동시에 진행되는 다양한 프로세스나 객체를 표현한다. 가로로 표현되는 화살표는 각 객체간 교환하는 메시지를 표현한다.

### E-R Diagram

텍스트, 스크린샷, 폰트, 도표이(가) 표시된 사진

자동 생성된 설명

Figure Example of E-R Diagram

E-R(Entity-Relationship) 다이어그램은 Entity-Relationship Model을 표현하는 것으로 데이터베이스를 개체, 속성, 관계성을 표현하는 방식을 통해 모델링하는 방식이다. 데이터베이스 설계 시에 주로 사용되는 다이어그램이다.

## Applied Tool

### Visual Studio Code

폰트, 그래픽, 로고, 스크린샷이(가) 표시된 사진

자동 생성된 설명

Figure VS Code Logo

Django Framework및 React사용 및 연동을 위해 Visual Studio Code를 사용한다.

Visual Studio Code는 MS에서 개발한 소스코드 편집 도구이다.

### draw.io

그래픽, 폰트, 디자인, 타이포그래피이(가) 표시된 사진

자동 생성된 설명

Figure draw.io Logo

draw.io는 브라우저 기반의 무료 diagramming 도구이다. UML diagram을 비롯하여 많은 diagram을 그릴 수 있는 기본 틀을 제공한다.

### Smartdraw



Figure smartdraw Logo

SmartDraw는 순서도에서 평면도에 이르기까지 70개 이상의 다양한 다이어그램 유형에 대한 템플릿을 제공한다. Microsoft Office​​, Atlassian's Confluence 및 Jira 등에 다이어그램을 쉽게 추가할 수 있고Google Drive, OneDrive, Dropbox 등에 파일을 저장하고 공유하는 것이 간단하다.

## Project Scope

### Overview

본 시스템은 소프트웨어 교육의 필요성의 증진에 힘입어 해당 교육에 초심자가 코딩에 흥미를 지속적으로 가질 수 있도록 유도하는 시스템이다. 본 시스템은 추가적으로 소프트웨어 학습자들에게 자칫하면 고려되지 않을 수 있는 항목인 윤리 교육을 실시하여 균형 잡힌 학습이 되도록 한다. 본 시스템은 User Management System, Ethics Quiz System, 시스템 C 로 구성된다.

### User Management System

User Management System은 사용자의 회원정보를 관리한다. 회원가입 및 로그인의 두개의 Sub-System으로 구성된다. 사용자는 회원가입, 로그인 기능을 통해 User DB와 상호작용한다

### Ethics Quiz System

Ethics Quiz System은 database에 저장되어 있는 윤리 퀴즈 문제와 각 문제에 대한 User의 진행상황을 불러온다. Website, Database, Problem, ChatGPT와 같은 4개의 Sub-System으로 구성된다.

### AI Competition coding game System

AI Competition System은 database에 저장되어 있는 난이도별 코딩 문제와 각 문제에 대한 User의 진행상황을 불러온다. Website, Database, Problem, ChatGPT와 같은 4 개의 Sub-System으로 구성된다.

# System Architecture

## Objective

개발되어지는 시스템에 대한 상세한 개요와 시스템기능의 전체적 분포에 대해 기술한다. 전체 시스템의 구조는 Block Diagram을 이용하여 표현한다.

## System Organization

### Overall System Architecture

텍스트, 스크린샷, 도표, 직사각형이(가) 표시된 사진

자동 생성된 설명

Diagram Overall System Architecture

### User Management System

텍스트, 스크린샷, 도표, 평행이(가) 표시된 사진

자동 생성된 설명

Diagram User Management System

### Problem Selection System

텍스트, 스크린샷, 도표, 평행이(가) 표시된 사진

자동 생성된 설명

Diagram Problem Selection System

### Ethics Quiz System

텍스트, 스크린샷, 도표, 평면도이(가) 표시된 사진

자동 생성된 설명

Diagram Ethics Quiz System

### AI Competition coding game System

텍스트, 스크린샷, 도표, 직사각형이(가) 표시된 사진

자동 생성된 설명

Diagram AI Competition Coding Game System

# User Management System

## Objective

전체 시스템의의 회원 가입과 로그인 기능을 이용 시에 시스템에서 발생하는 데이터를 처리하는 사용자 관리 시스템의 설계에 대해 설명한다. Class Diagram, Sequence Diagram, State Diagram 등 다양한 도식을 활용하여 User Management System의 구조를 표현하고 서술한다.

## Class Diagram

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

Diagram User Management System: Class Diagram

### DB Handler

A.1 Attributes

해당 사항 없음

A.2 Methods

+ read(query): 데이터베이스에서 원하는 데이터를 읽어온다.

+ write(userInfo): 데이터베이스에 해당 데이터를 저장한다.

### Account

B.1 Attributes

+ username : 해당 계정의 사용자 이름

+ email : 해당 계정의 ID

+ password : 해당 계정의 비밀번호

+ solvedCodingProblems : 해당 계정의 사용자가 푼 코딩 문제 ID 배열

+ solvedEthicsProblems : 해당 계정의 사용자가 푼 윤리 문제 ID 배열

B.2 Methods

+ getData() : 데이터베이스에서 원하는 데이터를 얻어온다.

+ writeData(userInfo) : 데이터베이스에 해당 데이터를 보낸다.

## Sequence Diagram

### Register and Login

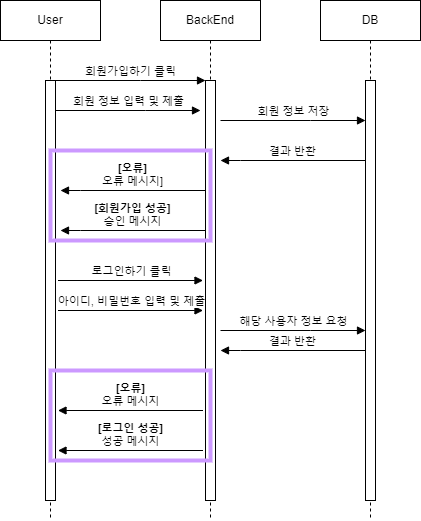


Diagram User Management System: Sequence Diagram

## State Diagram

### Register

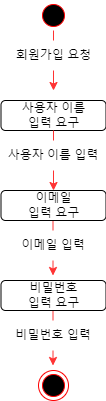


Diagram User Management System: State Diagram 회원가입

### Login

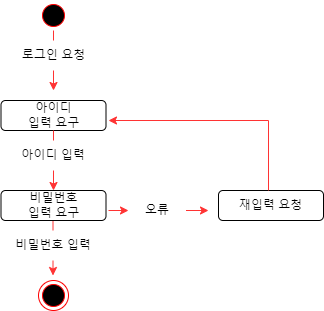


Diagram User Management System: State Diagram 로그인

# Ethics Quiz System

## Objective

윤리 교육을 위한 퀴즈 system이 있다. 이 system은 database에 저장되어 있는 윤리 퀴즈 문제와 각 문제에 대한 User의 진행상황을 불러온다. Sub-system은 Website, Database, Problem, ChatGPT 4개로 구성된다. 이 system에 대해 Class diagram, Sequence diagram, State diagram을 통해 본system의 구조를 표현하고 상세 내용을 기술한다.

## Class Diagram

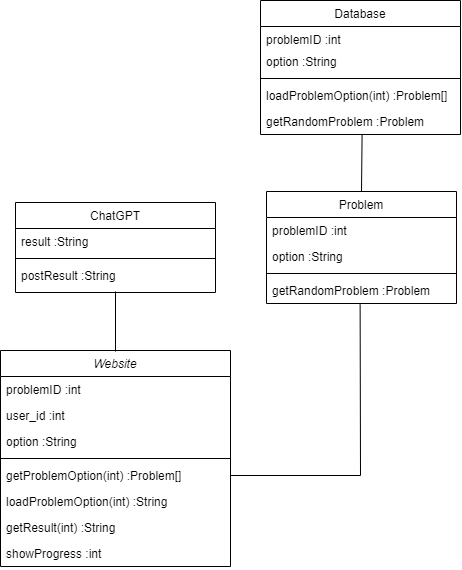


Diagram Ethic Quiz System: Class Diagram

1. ChatGPT

A.1. Attributes

-result :String 🡪 ChatGPT가 generate한 사용자의 선택에 대한 결과

A.2. Method

+postResult :String 🡪 result를 Website에 보낸다.

1. Website

B.1. Attributes

-problemID :int 🡪 여러 problem 중 선택된 problem의 ID값

-user\_id :int 🡪 User의 ID

-option :String 🡪 문제에 대한 선택지

B.2. Method

+getProblemOption(int) :Problem[] 🡪 문제들과 그에 대한 선택지 set를 get한다.

+loadProblemOption(int) :String 🡪 문제들과 그에 대한 선택지 set를 load한다.

+getResult(int) :String 🡪 ChatGPT가 generate한 사용자의 선택에 대한 결과를 받는다.

+showProgress :int 🡪 사용자별 진행률을 변경한다.

1. Problem

C.1. Attributes

-problemID :int 🡪 문제 상황

-option :String 🡪 문제 상황에 대한 선택지

C.2. Method

+getRandomProblem :Problem 🡪 여러 문제들 중 하나의 랜덤한 문제를 get한다.

1. Database

D.1. Attributes

-problemID :int 🡪 여러 problem 중 선택된 problem의 ID값

-option :String 🡪 문제 상황에 대한 선택지

D.2. Method

+loadProblemOption(int) :Problem[] 🡪 여러 문제들 중 하나의 랜덤한 문제를 load한다.

+getRandomProblem :Problem 🡪 여러 문제들 중 하나의 랜덤한 문제를 뽑아온다.

## Sequence Diagram

텍스트, 도표, 스크린샷, 평행이(가) 표시된 사진

자동 생성된 설명

Diagram Ethic Quiz System: Sequence Diagram

## State Diagram

### User

텍스트, 스크린샷, 도표, 라인이(가) 표시된 사진

자동 생성된 설명

Diagram Ethic Quiz System: State Diagram (User)

### Website

텍스트, 도표, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Diagram Ethic Quiz System: State Diagram (Website)

* + 1. Database

텍스트, 스크린샷, 도표, 원이(가) 표시된 사진

자동 생성된 설명

Diagram Ethic Quiz System: State Diagram (Database)

* + 1. Problem

텍스트, 스크린샷, 도표, 원이(가) 표시된 사진

자동 생성된 설명

Diagram Ethic Quiz System: State Diagram (Problem)

* + 1. ChatGPT

텍스트, 스크린샷, 도표, 원이(가) 표시된 사진

자동 생성된 설명

Diagram Ethic Quiz System: State Diagram (ChatGPT)

# AI Competition Coding Game System

## Objective

AI 경쟁 코딩 게임 system의 설계에 대해 설명한다. 이 System은 database에 저장되어 있는 난이도별 코딩 문제와 각 문제에 대한 User의 진행상황을 불러온다. Sub-system은 Website, Database, Problem, ChatGPT의 4 개이다. AI 경쟁 코딩 게임 system에 대해 Class diagram, Sequence diagram, State diagram을 통해 AI 경쟁 코딩 게임 system의 구조를 표현하고 상세 내용을 기술한다.

## Class Diagram

텍스트, 스크린샷, 도표, 폰트이(가) 표시된 사진

자동 생성된 설명

Diagram AI Competition Coding Game System: Class Diagram

1. ChatGPT

A.1. Attributes

-result :String 🡪 사용자의 코드로 돌려본 결과

-analysis: String 🡪 사용자의 코드의 첨삭 결과

A.2. Method

+postResult(int) : int 🡪 result를 Website로 보낸다.

+postAnalysis(string) :string 🡪 첨삭 결과를 Website로 보낸다.

1. Website

B.1. Attributes

-problemID :int 🡪 여러 problem 중 선택된 problem의 ID값

-user\_ id :int 🡪 User의 ID

-problem\_difficulty :int 🡪 문제의 난이도

B.2. Method

+selectProblem\_difficulty(int) :problem[] 🡪 선택된 난이도의 문제들만 보여준다.

+selectProblem(int) 🡪 :string 선택한 문제를 load한다.

+getResult(int) 🡪 :int 코드에 대한 점수를 ChatGPT로부터 받는다.

+getAnalysis(string) :string 🡪 코드에 대한 첨삭을 ChatGPT로부터 받는다.

+is\_hide : boolean 🡪 AI 코드를 숨길지를 결정한다.

+showProgress(int) : int 🡪 사용자의 진행도를 보여준다.

1. Problem

C.1. Attributes

-problemID : int 🡪 코딩 문제

-AI\_answer : string 🡪 AI 답안

C.2. Method

+getProblem(int) :string 🡪 선택한 문제를 받아온다.

1. Database

D.1. Attributes

-problemID : int 🡪 문제의 ID값

-problem\_case : case 🡪 문제의 test case

D.2. Method

+loadProblem(int) : string 🡪 선택한 문제를 load 한다.

+post\_case(int) : case 🡪 문제의 test case를 보낸다.

## Sequence Diagram

텍스트, 스크린샷, 도표, 번호이(가) 표시된 사진

자동 생성된 설명

Diagram AI Competition Coding Game System: Sequence Diagram

## State Diagram

* + 1. User

텍스트, 스크린샷, 도표, 폰트이(가) 표시된 사진

자동 생성된 설명

Diagram AI Competition Coding Game System: State Diagram (User)

* + 1. Website

텍스트, 도표, 스크린샷, 폰트이(가) 표시된 사진

자동 생성된 설명

Diagram AI Competition Coding Game System: State Diagram (Website)

* + 1. Database

텍스트, 스크린샷, 도표, 폰트이(가) 표시된 사진

자동 생성된 설명

Diagram AI Competition Coding Game System: State Diagram (Database)

* + 1. Problem

텍스트, 스크린샷, 도표, 원이(가) 표시된 사진

자동 생성된 설명

Diagram AI Competition Coding Game System: State Diagram (Problem)

6.4.5. ChatGPT

텍스트, 스크린샷, 도표, 폰트이(가) 표시된 사진

자동 생성된 설명

Diagram AI Competition Coding Game System: State Diagram (ChatGPT)

# Protocol Design

## Objective

Protocol Design 챕터는 서비스 내의 시스템 간 통신을 위한 프로토콜에 대해 다룬다. 데이터 형식으로는 XML과 JSON이 주로 사용되며, 웹과 서버 간의 데이터 포맷은 Django 템플릿을 이용하고, API에서는 JSON을 사용한다. 프론트엔드 시스템과 백엔드 시스템의 동작 방식과 상호작용, 그리고 서버와 클라이언트 간의 통신 프로토콜과 인터페이스를 설명한다. 또한, 서브시스템 간 상호작용에서 필수적으로 준수해야 하는 프로토콜과 메시지의 형식, 용도, 의미도 다룬다.

## Json

JSON은 개방형 표준 포맷으로, 데이터를 속성-값 쌍 또는 키-값 쌍으로 전달하기 위해 사용됩니다. 이는 주로 비동기 AJAX나 XML을 대체하는 데이터 포맷으로 알려져 있으며, 웹 애플리케이션과 웹 서버 간의 통신에서 널리 사용됩니다. JSON은 사람이 읽을 수 있는 형태의 파일 형식이며, 다양한 자료의 타입을 담을 수 있습니다. 우리 시스템에서는 유저 인증에 JWT를 사용하고, 프론트엔드 서버와 백엔드 서버 간의 상호작용에는 REST API를 통해 JSON을 사용합니다. JSON은 객체 문법으로 구조화된 데이터를 표현하며, 인터넷 데이터 교환에서 널리 활용됩니다.

## Protocol Description

### Overview

Protocol Description에서는 Client와 Server 간의 요청-응답 메시지를 각각의 구현 기능에 따라 구분하여 설명한다. 이 절에서는 표의 캡션을 생략하고 설명한다. 우리 시스템에서는 HyperText Transfer Protocol인 HTTP 프로토콜을 사용하여 서버와 클라이언트가 통신한다. REST API를 활용하여 주소와 HTTP 요청 메소드를 통해 서버로부터 동작을 요청할 예정이다.

### Login Protocol

|  |  |  |  |
| --- | --- | --- | --- |
| GET | /user/login | | |
| 사용자가 로그인할 때 요청하는 주소 | | | |
| 분류 | 항목명 | 설명 및 제약사항 | 예시 |
| Parameter | Id | 사용자 아이디 |  |
| password | 비밀번호 |  |
| Response | Message | “성공”, “실패” |  |
| Status code | 상태 코드 |  |
| Status Code | 200 | 로그인 성공 |  |
| 400 | 로그인 실패 |  |

Table Login Protocol

### Register Protocol

|  |  |  |  |
| --- | --- | --- | --- |
| POST | /user | | |
| 사용자가 회원가입을 요청할 때 사용하는 주소 | | | |
| 분류 | 항목명 | 설명 및 제약사항 | 예시 |
| Parameter | Id | 사용자 아이디 |  |
| Password | 비밀번호 |  |
| Password\_confirmation | 비밀번호 확인, 위 비밀번호와 동일 해야함 |  |
| Response | Message | “성공”, “실패: 사유” |  |
| Status code |  |  |
| Status Code | 201 | 회원가입 성공 |  |
| 400 | 비밀번호 확인이 비밀번호와 동일하지 않음, |  |
| 409 | 사용자 id 중복일 경우 |  |

Table Register Protocol

### Select Problem Type Protocol

|  |  |  |  |
| --- | --- | --- | --- |
| GET | /problems | | |
| 문제 유형을 선택하여 문제 유형을 요청하는 주소 | | | |
| 분류 | 항목명 | 설명 및 제약사항 | 예시 |
| Parameter | select | 유형 선택 |  |
| Response | Problem\_list | 문제 리스트 |  |
| Status code |  |  |
| Status Code | 200 | 문제 유형 요청 성공 |  |
| 400 | Type 파라미터가 숫자형이 아님 |  |

Table Select Problem Type Protocol

### Select Challenge Problem Protocol

|  |  |  |  |
| --- | --- | --- | --- |
| GET | /problems/challenge | | |
| 사용자가 AI와 대결할 문제를 요청하는 주소 | | | |
| 분류 | 항목명 | 설명 및 제약사항 | 예시 |
| Parameter | Id | 문제 ID |  |
| Response | Title | 문제 제목 |  |
| Type | 문제 유형 |  |
| Data | 문제 데이터 |  |
| Status code | 상태 코드 |  |
| Status Code | 200 | 문제 요청 성공 |  |
| 400 | Id 파라미터가 숫자형이 아님 |  |
| 404 | 해당 문제 존재하지 않음 |  |

Table Select Challenge Problem Protocol

### Challenge Problem Submission Protocol

|  |  |  |  |
| --- | --- | --- | --- |
| GET | /problems/challenge/submission | | |
| 사용자가 문제의 정답을 제출하여 채점과 피드백을 요청하는 주소 | | | |
| 분류 | 항목명 | 설명 및 제약사항 | 예시 |
| Parameter | answer | 정답 + 코드 리스트 |  |
| Response | Redirect\_url | 제출 결과 보여주는 URL 반환 |  |
| Data | 코드 피드백 |  |
| Status code | 상태 코드 |  |
| Status Code | 200 | 문제 제출 성공 |  |
| 400 | 코드와 정답이 형식에 맞지 않음 |  |

Table Challenge Problem Submission Protocol

### Select Ethical Problem Protocol

|  |  |  |  |
| --- | --- | --- | --- |
| GET | /problems/ethical | | |
| 사용자가 윤리 문제를 요청하는 주소 | | | |
| 분류 | 항목명 | 설명 및 제약사항 | 예시 |
| Parameter | Id | 문제 ID |  |
| Response | Title | 문제 제목 |  |
| Type | 문제 유형 |  |
| Data | 문제 데이터 |  |
| Status code | 상태 코드 |  |
| Status Code | 200 | 문제 요청 성공 |  |
| 400 | Id 파라미터가 숫자형이 아님 |  |
| 404 | 해당 문제 존재하지 않음 |  |

Table Select Ethical Problem Protocol

### Ethical Problem Submission Protocol

|  |  |  |  |
| --- | --- | --- | --- |
| GET | /problems/ethical/submission | | |
| 사용자가 문제의 선택을 제출하여 결과를 요청하는 주소 | | | |
| 분류 | 항목명 | 설명 및 제약사항 | 예시 |
| Parameter | choice | 선택 |  |
| Response | Redirect\_url | 제출 결과 보여주는 URL 반환 |  |
| Data | 선택에 따른 결과 |  |
| Status code | 상태 코드 |  |
| Status Code | 200 | 문제 제출 성공 |  |
| 400 | 코드와 정답이 형식에 맞지 않음 |  |

Table Ethical Problem Submission Protocol

### Create Problem Protocol

|  |  |  |  |
| --- | --- | --- | --- |
| POST | /problems/challenge | | |
| 관리자가 AI와 대결할 문제를 추가하는 주소 | | | |
| 분류 | 항목명 | 설명 및 제약사항 | 예시 |
| Parameter | pid | 문제 ID |  |
| Type | 문제 유형 |  |
| data | 문제 데이터 |  |
| Response | Message | “성공”, “실패” |  |
| Status code | 상태 코드 |  |
| Status Code | 200 | 문제 추가 성공 |  |
| 400 | 파라미터가 제대로 채워지지 않음 |  |

Table Create Problem Protocol

# Database Design

## Objective

이 장에서는 요구사항 명세서에서 작성한 데이터베이스 요구사항에 대하여 실질적인 데이터베이스 설계를 기술한다. ER Diagram을 통해 개체 간 관계를 보여주고, 개체 속성의 역할을 명시하고, 실제 개발에 대하여 Relational Schema, 데이터 정의와 접근 양식에 대하여 기술한다.

## E-R Diagram

테이블이 되는 모든 개체/관계는 PK로 등록 번호를 가진다.

1. Entities

A-1. User

도표, 원, 텍스트, 라인이(가) 표시된 사진

자동 생성된 설명

Diagram E-R Diagram: User

A-2. Coding Problems

텍스트, 도표, 라인, 원이(가) 표시된 사진

자동 생성된 설명

Diagram E-R Diagram: Coding Problems

PID는 DB에서 자동으로 생성해 주는 int 값이며, 이 값으로 각 문제를 식별한다. Contents는 코딩 문제의 내용을 의미하며, string으로 저장한다. Answer는 사용자가 힌트 보기를 클릭했을 때 보여줄 AI가 작성한 답안이며, string으로 저장한다. Testcase는 사용자가 제출한 답안을 채점할 때 사용할 값으로, 입출력 쌍으로 구성되어 있으며 string으로 저장한다. 모두 null을 허용하지 않는다.

A-3 Ethics Problems

텍스트, 도표, 라인, 화이트이(가) 표시된 사진

자동 생성된 설명

Diagram E-R Diagram: Ethics Problems

PID는 DB에서 자동으로 생성해 주는 int 값이며, 이 값으로 각 문제를 식별한다. Contents는 윤리 문제의 내용을 의미하며, string으로 저장한다. Options는 사용자가 선택할 수 있는 선택쌍으로, string으로 저장한다.

A-4 User Coding Submission

스케치, 그림, 도표, 라인이(가) 표시된 사진

자동 생성된 설명

Diagram E-R Diagram: User Coding Submission

SID는 DB에서 자동으로 생성해 주는 int 값이며, 이는 User Coding Submission table의 기본키이다. UID는 각 유저를 구분하고 PID는 각 Coding 문제를 구분한다. 유저가 마지막으로 제출한 코드는 submission에 string으로 저장되고 AI 첨삭 문구가 response에 저장된다. 유저가 이 문제를 맞혔는지에 대한 여부는 isSolved에 저장된다.

A-5 User Ethics Submission

스케치, 도표, 그림, 라인이(가) 표시된 사진

자동 생성된 설명

Diagram E-R Diagram: User Ethics Submission

기본적으로 SID, UID, PID에 관한 설명은 A-4와 동일하다. 하지만 coding이 아닌 윤리 문제 제출 테이블이므로 submission은 두 가지 선택 중 유저가 선택한 것을 저장하며 response는 이 선택에 대한 결과를 나타낸다. 또한 유저가 이 문제를 풀었는지에 대한 여부가 isSolved에 저장된다.

1. Relations

B-1. User - Coding Problem

스크린샷, 원, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

Diagram E-R Diagram: User - Coding Problem

사용자는 코드 문제를 리스트에서 선택할 수 있으며, 제한시간 및 답안의 흐림 설정을 통해 난이도를 자체적으로 조절할 수 있다.

B-2. User - Ethics Problem

스크린샷, 폰트, 원, 라인이(가) 표시된 사진

자동 생성된 설명

Diagram E-R Diagram: User - Ethics Problem

사용자는 윤리 문제를 리스트에서 선택할 수 있으며, Contents를 읽고 Options을 읽고 선택할 수 있다.

B-3 Coding Problem - Coding Submission

도표, 스케치, 그림, 원이(가) 표시된 사진

자동 생성된 설명

Diagram E-R Diagram: Coding Problem - Coding Submission

코딩 문제는 답안이 제출될 수 있고 한 문제당 여러 답안이 제출되므로 1 : N의 관계를 가진다. 이때 답이 맞으면 IsSolved는 참 값으로 저장된다. Coding Submission의 UID 값으로 해당 문제는 어떤 유저가 풀었는지 알 수 있다.

B-4 Ethics Problem – Ethics Submission

스크린샷, 원, 폰트, 텍스트이(가) 표시된 사진

자동 생성된 설명

Diagram E-R Diagram: Ethics Problem - Ethics Submission

윤리문제의 선택된 옵션은 제출되고 제출 시스템의 응답을 수신한다. 이때 IsSolved는 참 값으로 설정된다. Ethics Submission의 PID와 UID는 각각 User과 Ethics Problem을 참조하는 외래 키이다.

## Relation Schema

텍스트, 스크린샷, 폰트, 도표이(가) 표시된 사진

자동 생성된 설명

Diagram E-R Diagram: Ethics Problem - Ethics Submission

User, CodingProblem, EthicsProblem이 3가지 메인 테이블이고 UserCoding Submission과 UserEthicsSubmission은 각 유저의 각 문제별 submission 및 response 정보를 담고 있다. Coding문제에서 문제별 Testcase와 Answer는 중복되기 때문에 독립적인 Coding\_TestcaseAnswer table을 추가하였다.

# Testing Plan

## Objective

요구사항 명세서에서 기술한 사용자 및 관리자 Scenario를 바탕으로 전체 시스템이 구상대로 실행되는지 확인한다. 또한, 발생할 수 있는 오류를 사전에 발견한다. 테스트를 수행하기 위한 테스팅 정책을 기술하고, 이를 설계한다. 각 testing plan에서는 testing policy와 test case를 설명한다.

## Testing Policy

### Developing Testing

시스템 개발 과정에서의 Developing Testing은 소프트웨어의 신뢰성과 품질을 위해 중요하다. Component Testing, System Testing, Acceptance Testing의 세 가지 단계로 구성된다. Component Testing에서는 개별 컴포넌트를 독립적으로 테스트하고, System Testing에서는 개별 컴포넌트들을 통합한 전체 시스템을 테스트한다. 마지막으로 Acceptance Testing에서는 실제 고객 데이터를 이용하여 시스템이 고객의 요구사항을 충족시키는지 확인한다.

### Release Testing

시스템의 새로운 버전 출시를 위해 수행되는 과정으로, 배포 과정이 정상적으로 이루어지는지 검증한다. 이를 위해 일반적으로 알파 테스트와 베타 테스트를 포함한 여러 단계로 구성된다. 알파 테스트는 개발 조직 내에서 진행되며, 베타 테스트는 제한된 사용자에게 공개되어 진행된다.

### User Testing

실제 사용 환경에서 시스템을 사용자들에게 테스트하는 단계이다. 사용자들을 통해 시스템의 직관성과 사용 용이성을 평가하며, 시스템 출시 여부를 결정한다. 예상 사용자가 시스템을 사용하는 과정을 시뮬레이션하고, 시스템이 모든 요구사항을 충족하는지 검증한다. 이를 통해 시스템의 베타 버전을 평가하고, 실제 환경에 배포할지 여부를 결정한다.

## Test Case

### User Management System

9.3.1.1 회원가입

|  |  |
| --- | --- |
| **구성요소** | **설명** |
| 테스트 이름  (Test Title) | 회원가입 테스트 |
| 사전 조건 (Precondition) | 1. 이메일: ' 아이디 문자열 @ 도메인 문자열 ' 형식이며, 중복되어서는 안된다. 2. 비밀번호는 8~16자의 문자열 형식이다. |
| 수행 절차 (Test Step) | 1. 이메일 형식 체크 2. 이메일 중복 체크 3. 비밀번호 형식 체크 4. 모든 양식 작성했는지 체크 |
| 기대되는 결과 (Expected Result) | 회원 가입 후 데이터베이스에 회원 정보 저장  로그인 화면으로 전환 |

Table Test Case: 회원가입

9.3.1.2 로그인

|  |  |
| --- | --- |
| **구성요소** | **설명** |
| 테스트 이름  (Test Title) | 로그인 테스트 |
| 사전 조건 (Precondition) | 1. 이메일은 ' 아이디 문자열 @ 도메인 문자열 ' 형식이다. 2. 비밀번호는 4~12자의 문자열 형식이다. 3. 여러번 로그인 시도할 경우, 최초 한 번의 시도만 서버에 요청한다. |
| 수행 절차 (Test Step) | 1. 이메일 형식 체크 2. 비밀번호 형식 체크 3. 모든 양식 기입했는지 체크 4. 이메일 존재여부 체크 5. 이메일과 비밀번호 일치 여부 체크 |
| 기대되는 결과 (Expected Result) | 사용자 로그인 상태로 기록, 메인 화면으로 전환 |

Table Test Case: 로그인

9.3.1.3 마이페이지 비밀번호 변경

|  |  |
| --- | --- |
| **구성요소** | **설명** |
| 테스트 이름  (Test Title) | 마이페이지 비밀번호 변경 테스트 |
| 사전 조건  (Precondition) | 사용자는 로그인한 상태여야 한다. |
| 수행 절차  (Test Step) | 1. 사용자가 마이페이지 버튼을 누른다. 2. 사용자가 비밀번호 변경 버튼을 누른다. 3. 새로운 비밀번호를 입력할 화면이 뜬다 4. 새로운 비밀번호를 입력한다. 5. 비밀번호 확인 칸에 다시 입력한다. 6. 비밀번호 변경 확인 버튼을 누른다. |
| 기대되는 결과  (Expected Result) | 새로운 비밀번호를 사용자DB에 저장한다. |

Table Test Case: 마이페이지 비밀번호 변경

### Ethics Quiz System

### 9.3.2.1 윤리 퀴즈 문제 리스트 확인

|  |  |
| --- | --- |
| **구성요소** | **설명** |
| 테스트 이름  (Test Title) | 윤리 퀴즈 문제 리스트 확인 테스트 |
| 사전 조건  (Precondition) | 1. 사용자는 로그인한 상태여야 한다. 2. 사용자는 전 화면에서 윤리 퀴즈와 AI코딩대결 중 윤리 퀴즈를 선택해야 한다. |
| 수행 절차  (Test Step) | 1. 사용자가 전 화면에서 윤리 퀴즈 버튼을 누른다. 2. 사용자가 윤리 퀴즈 문제 리스트를 본다. 3. 각 문제에는 그 문제에 대한 사용자의 문제 풀이 현황이 표시된다. |
| 기대되는 결과  (Expected Result) | 사용자는 각 문제에 대한 자신의 풀이 상황과 함께 모든 문제 리스트를 확인한다. |

Table Test Case: 윤리 퀴즈 문제 리스트 확인

9.3.2.2 윤리 퀴즈 문제 풀기

|  |  |
| --- | --- |
| **구성요소** | **설명** |
| 테스트 이름  (Test Title) | 윤리 퀴즈 문제 풀기 테스트 |
| 사전 조건  (Precondition) | 사용자는 로그인한 상태여야 한다. |
| 수행 절차  (Test Step) | 1. 사용자가 문제 리스트에서 문제를 클릭한다. 2. 사용자는 문제 상황을 확인한다. 3. 사용자는 주어진 두 가지 선택지 중 하나를 선택한다. 4. 사용자가 선택한 선택지에 대해 ChatGPT의 결과를 받아온다. 5. 사용자는 결과 화면을 확인한다. |
| 기대되는 결과  (Expected Result) | 사용자가 선택한 선택지에 대한 ChatGPT의 결과 |

Table Test Case: 윤리 퀴즈 문제 풀기

### AI Competition coding game System

9.3.3.1 문제 선택하기

|  |  |
| --- | --- |
| **구성요소** | **설명** |
| 테스트 이름  (Test Title) | AI Competition coding game 선택하는 테스트 |
| 사전 조건  (Precondition) | 1. 사용자는 로그인한 상태여야 한다. 2. 사용자는 AI Competition coding game을 선택한 상태여야 한다. |
| 수행 절차  (Test Step) | 1. 문제 리스트를 볼 수 있게 한다. 2. 난이도 초급, 중급, 고급을 선택할 수 있게 한다. 3. 사용자는 난이도를 하나 선택한다. 4. 선택한 난이도에 맞는 문제가 뜬다. |
| 기대되는 결과  (Expected Result) | 문제를 풀 수 있는 화면으로 표시되고, 사용자가 선택한 문제가 표시된다. |

Table 문제 선택하기

9.3.3.2 AI Competition coding game문제 풀기

|  |  |
| --- | --- |
| **구성요소** | **설명** |
| 테스트 이름  (Test Title) | AI Competition coding game 문제 풀기 테스트 |
| 사전 조건  (Precondition) | 1. 사용자는 로그인한 상태여야 한다. 2. 사용자는 AI Competition coding game을 선택한 상태여야 한다. 3. 사용자는 문제를 선택한 상태여야 한다. |
| 수행 절차  (Test Step) | 1. 사용자는 문제 설명을 확인한다. 2. 사용자는 문제의 정답 코드를 작성한다. 3. AI의 진행속도에 맞춰 코드가 풀리는 걸 확인한다. 4. 사용자는 제출하기 버튼을 클릭한다. 5. 사용자는 결과화면을 확인한다. |
| 기대되는 결과  (Expected Result) | 사용자의 내부 문제 채점 결과  사용자의 코드에 대한 피드백  (성공, 실패, 승리, 패배, 에러) |

Table AI Competition coding game문제 풀기

9.3.3.3 AI Competition coding game 문제 관리하기

|  |  |
| --- | --- |
| **구성요소** | **설명** |
| 테스트 이름  (Test Title) | AI Competition coding game 문제 관리하기 테스트 |
| 사전 조건  (Precondition) | 관리자로 로그인된 상태여야한다. |
| 수행 절차  (Test Step) | 1. 관리자는 문제 추가하기 버튼을 누른다. 2. 관리자는 문제 ID를 입력한다. (0~9999999) 3. 관리자는 문제 이름을 입력한다. 4. 관리자는 문제 설명을 입력한다. 5. 관리자는 입력 설명을 입력한다. 6. 관리자는 출력 설명을 입력한다. 7. 관리자는 입출력 케이스를 업로드한다. 8. 관리자는 추가하기 버튼을 누른다. |
| 기대되는 결과  (Expected Result) | 시스템 메시지 (성공, 에러) |

Table AI Competition coding game 문제 관리하기

# Development Environment

## Objective

이 절에서는 실제 시스템 개발을 위해 필요한 개발환경 및 코딩 규칙에 대해 기술한다. 또한 개발과정에서의 버전 관리도구 및 프로그래밍 과정에서 기반이 되는 규칙에 대해 서술한다.

## Django



Figure Django Logo

Django는 python으로 만들어진 무료 오픈소스 웹 애플리케이션 프레임워크(web application framework) 로 쉽고 빠르게 웹사이트를 개발할 수 있도록 한다. 모델-뷰-컨트롤러(MVC) 패턴을 따르고 있다. Django는 유연하고 빠른 확장성을 제공해 CMS부터 SNS까지 다양한 플랫폼에 구축·활용되고 있다.

## React

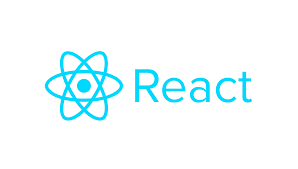


Figure React Logo

React는 메타에서 개발한 오픈 소스 자바스크립트 라이브러리로 프론트 엔드 개발자들이 자주 사용하는 라이브러리이다. SPA을 전제로 하고 있으며, Dirty checking과 Virtual DOM을 활용하여 업데이트해야 하는 DOM 요소를 찾아서 해당 부분만 업데이트하는 방법으로 재 렌더링이 잦은 동적 모던 웹에서 높은 퍼포먼스를 보여줄 수 있도록 한다. 모듈형 개발이기 때문에 높은 생산성을 보여주는 라이브러리이다. 다른 프레임워크와 함께 사용할 수 있는 장점이 있다.

## GitHub



Figure GitHub Logo

Git(버전 관리 시스템 중 하나)을 사용하는 프로젝트 버전제어 및 공동 작업을 위한 코드 호스팅 플랫폼이다. 전 세계에서 오픈소스 프로젝트 관리를 위해 가장 많이 사용되는 서비스로 사용자에게 무료로 계정과 저장소를 제공하고, 부분적으로 비공개 프로젝트도 무료로 진행할 수 있다.

# Development Plan

## Objective

본 항목에서는 프로젝트 개발 일정에 대해 기술한다. Gantt Chart를 이용해 전반적인 개발 순서 및 시기에 대해 표현한다.

## Gantt Chart

텍스트, 스크린샷, 번호, 폰트이(가) 표시된 사진

자동 생성된 설명

Figure Gantt Chart

현재까지 사용자 요구사항 구체적 명세화 단계까지 진행되었다. 기능적, 비기능적 요구사항을 분석하고 시스템 모델링의 다이어그램 표현까지 진행하였다.

# Index

## Objective

본 문서의 인덱스를 정리하여 표현한다.

## Figure Index

[Figure 1 Concept of UML 9](#_Toc135530376)

[Figure 2 Example of Use-Case Diagram 10](#_Toc135530377)

[Figure 3 Example of Class Diagram 11](#_Toc135530378)

[Figure 4 Example of State Diagram 12](#_Toc135530379)

[Figure 5 Example of Sequence Diagram 13](#_Toc135530380)

[Figure 6 Example of E-R Diagram 14](#_Toc135530381)

[Figure 7 VS Code Logo 15](#_Toc135530382)

[Figure 8 draw.io Logo 15](#_Toc135530383)

[Figure 9 smartdraw Logo 16](#_Toc135530384)

[Figure 10 Django Logo 56](#_Toc135530385)

[Figure 11 React Logo 57](#_Toc135530386)

[Figure 12 GitHub Logo 58](#_Toc135530387)

[Figure 13 Gantt Chart 59](#_Toc135530388)

## Diagram Index

[Diagram 1 Overall System Architecture 18](#_Toc135530389)

[Diagram 2 User Management System 19](#_Toc135530390)

[Diagram 3 Problem Selection System 19](#_Toc135530391)

[Diagram 4 Ethics Quiz System 20](#_Toc135530392)

[Diagram 5 AI Competition Coding Game System 20](#_Toc135530393)

[Diagram 6 User Management System: Class Diagram 21](#_Toc135530394)

[Diagram 7 User Management System: Sequence Diagram 23](#_Toc135530395)

[Diagram 8 User Management System: State Diagram 회원가입 24](#_Toc135530396)

[Diagram 9 User Management System: State Diagram 로그인 24](#_Toc135530397)

[Diagram 10 Ethic Quiz System: Class Diagram 25](#_Toc135530398)

[Diagram 11 Ethic Quiz System: Sequence Diagram 27](#_Toc135530399)

[Diagram 12 Ethic Quiz System: State Diagram (User) 28](#_Toc135530400)

[Diagram 13 Ethic Quiz System: State Diagram (Website) 28](#_Toc135530401)

[Diagram 14 Ethic Quiz System: State Diagram (Database) 29](#_Toc135530402)

[Diagram 15 Ethic Quiz System: State Diagram (Problem) 29](#_Toc135530403)

[Diagram 16 Ethic Quiz System: State Diagram (ChatGPT) 29](#_Toc135530404)

[Diagram 17 AI Competition Coding Game System: Class Diagram 30](#_Toc135530405)

[Diagram 18 AI Competition Coding Game System: Sequence Diagram 32](#_Toc135530406)

[Diagram 19 AI Competition Coding Game System: State Diagram (User) 33](#_Toc135530407)

[Diagram 20 AI Competition Coding Game System: State Diagram (Website) 33](#_Toc135530408)

[Diagram 21 AI Competition Coding Game System: State Diagram (Database) 34](#_Toc135530409)

[Diagram 22 AI Competition Coding Game System: State Diagram (Problem) 34](#_Toc135530410)

[Diagram 23 AI Competition Coding Game System: State Diagram (ChatGPT) 35](#_Toc135530411)

[Diagram 24 E-R Diagram: User 41](#_Toc135530412)

[Diagram 25 E-R Diagram: Coding Problems 42](#_Toc135530413)

[Diagram 26 E-R Diagram: Ethics Problems 42](#_Toc135530414)

[Diagram 27 E-R Diagram: User Coding Submission 43](#_Toc135530415)

[Diagram 28 E-R Diagram: User Ethics Submission 43](#_Toc135530416)

[Diagram 29 E-R Diagram: User - Coding Problem 44](#_Toc135530417)

[Diagram 30 E-R Diagram: User - Ethics Problem 44](#_Toc135530418)

[Diagram 31 E-R Diagram: Coding Problem - Coding Submission 45](#_Toc135530419)

[Diagram 32 E-R Diagram: Ethics Problem - Ethics Submission 45](#_Toc135530420)

[Diagram 33 E-R Diagram: Ethics Problem - Ethics Submission 46](#_Toc135530421)

## Table Index

[Table 1 Document Version History 8](#_Toc135530422)

[Table 2 Login Protocol 37](#_Toc135530423)

[Table 3 Register Protocol 37](#_Toc135530424)

[Table 4 Select Problem Type Protocol 38](#_Toc135530425)

[Table 5 Select Challenge Problem Protocol 38](#_Toc135530426)

[Table 6 Challenge Problem Submission Protocol 38](#_Toc135530427)

[Table 7 Select Ethical Problem Protocol 39](#_Toc135530428)

[Table 8 Ethical Problem Submission Protocol 39](#_Toc135530429)

[Table 9 Create Problem Protocol 40](#_Toc135530430)

[Table 10 Test Case: 회원가입 48](#_Toc135530431)

[Table 11 Test Case: 로그인 49](#_Toc135530432)

[Table 12 Test Case: 마이페이지 비밀번호 변경 50](#_Toc135530433)

[Table 13 Test Case: 윤리 퀴즈 문제 리스트 확인 51](#_Toc135530434)

[Table 14 Test Case: 윤리 퀴즈 문제 풀기 52](#_Toc135530435)

[Table 15 문제 선택하기 53](#_Toc135530436)

[Table 16 AI Competition coding game문제 풀기 54](#_Toc135530437)

[Table 17 AI Competition coding game 문제 관리하기 55](#_Toc135530438)

# References

## Objective

본 문서 작성에 활용된 참고문헌에 대해 기술한다.

## References

- draw.io

<https://app.diagrams.net/> (2023.05.17)

- 깃허브 블로그

<https://github.blog/> (2023.05.17)

- 스마트 드로우

<https://www.mindonmap.com/ko/blog/smartdraw-review/> (2023.05.19)

-위키피디아 개체-관계 모델 (2023.05.19)

<https://ko.wikipedia.org/wiki/%EA%B0%9C%EC%B2%B4-%EA%B4%80%EA%B3%84_%EB%AA%A8%EB%8D%B8>

- Django 시작하기 (2023.05.19)

https://docs.djangoproject.com/ko/4.2/intro/