



Code with Star

Requirement Specification

Team 7

2018310561 유영석

2018311895 김정식

2020314754 오민재

2018312511 조예현

2018314737 김명수

2017310734 김수한

2023.04.28

CONTENTS

1. INTRODUCTION.....	6
1.1 PURPOSE	6
1.2 SCOPE.....	6
1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS	7
1.4 REFERENCES	8
1.5 OVERVIEW.....	9
2. OVERALL DESCRIPTION	9
2-1. PRODUCT PERSPECTIVE	10
2-2. PRODUCT FUNCTIONS	10
2-2-1. USER AUTHENTICATION	10
2-2-2. COURSE	11
2-2-3. QUIZZ.....	11
2-2-1. COURSE UPLOAD & MANAGEMENT	11
2-3. USER CLASSES AND CHARACTERISTICS	12
2-3-1. SYSTEM ADMINISTRATOR.....	12
2-3-2. EDUCATOR.....	12
2-3-3. LEARNER	12
2-4. OPERATING ENVIRONMENT.....	12
2-5. DESIGN AND IMPLEMENTATION CONSTRAINTS.....	13
2-6. ASSUMPTION AND DEPENDENCIES	13
3. EXTERNAL INTERFACE REQUIREMENTS	14
3.1 USER INTERFACE.....	14
3.2 HARDWARE INTERFACE	32
3.3 SOFTWARE INTERFACE.....	33
3.4 COMMUNICATION INTERFACE	34
4. SYSTEM REQUIREMENTS	40

4.1. FUNCTIONAL REQUIREMENTS.....	40
4.1.1. USE CASE.....	40
4.1.2. USE CASE DIAGRAM	46
4.1.3. DATA FLOW DIAGRAM	47
4.2. NON FUNCTIONAL REQUIREMENTS	48
4.2.1. PRODUCT REQUIREMENT	48
4.2.2. ORGANIZATIONAL REQUIREMENT	49
4.2.3. EXTERNAL REQUIREMENT	49
4.3. ORGANIZING SYSTEM FLOW	50
4.3.1. CONTEXT MODEL.....	50
4.3.2. PROCESS MODEL	50
4.3.3. INTERACTION MODEL.....	51
4.3.4. BEHAVIOR MODEL.....	51
4.4. SYSTEM ARCHITECTURE.....	52

LIST OF FIGURES

Figure 1. Design of Login Page	15
Figure 2. Design of Sign Up Page.....	16
Figure 3. Design of Main Page	18
Figure 4. Design of Course Page.....	20
Figure 5. Design of My Page.....	21
Figure 6. Design of Quiz Page (Execute)	23
Figure 7. Design of Quiz Page (Submit)	24
Figure 8. Design of Uploading Lectures (Initial State)	26
Figure 9. Design of Modal Window for Adding Chapters	27
Figure 10. Design of Uploading Lectures (Added Chapters)	27
Figure 11. Design of Modal Window for Adding Sections.....	28
Figure 12. Design of Uploading Lectures (Added Sections)	28
Figure 13. Design of Course List	30

Figure 14. Design of Course List (After Feedback)	30
Figure 15. Design of Course Modification.....	32
Figure 16. Use Case Diagram.....	46
Figure 17. Data Flow Label	47
Figure 18. User Management Data Flow Diagram	47
Figure 19. Educator Data Flow Diagram	47
Figure 20. Learner Data Flow Diagram	48
Figure 21. Context Model	50
Figure 22. Process Model.....	50
Figure 23. Sequence Diagram.....	51
Figure 24. System Architecture	52

LIST OF TABLES

Table 1. List of Terms and Definitions.....	7
Table 2. List of Acronyms and Abbreviations	8
Table 3. User Interface of Login Page.....	14
Table 4. User Interface of Sign Up Page	15
Table 5. User Interface of Main Page.....	17
Table 6. User Interface of Course Page	18
Table 7. User Interface of My Page	20
Table 8. User Interface of Quiz Page (Execute)	22
Table 9. User Interface of Quiz Page (Submit)	23
Table 10. User Interface of Uploading Lectures.....	25
Table 11. User Interface of Course Syllabus	29
Table 12. User Interface of Course Modification	31
Table 13. Hardware Interface of Client	32
Table 14. Hardware Interface of Server.....	32
Table 15. Software Interface of Client.....	33
Table 16. Software Interface of Server	33

Table 17. HTTP Request Format for Login	34
Table 18. HTTP Request Format for Sign Up.....	34
Table 19. HTTP Request Format for Withdrawal	35
Table 20. HTTP Request Format for Course Search	35
Table 21. HTTP Request Format for Course List Lookup	36
Table 22. HTTP Request Format for Course.....	36
Table 23. HTTP Request Format for Quiz.....	37
Table 24. HTTP Request Format for Chatbot.....	37
Table 25. HTTP Request Format for My Page	38
Table 26. HTTP Request Format for Uploading Lectures	38
Table 27. HTTP Request Format for Lecture Management.....	39
Table 28. Function Use Case	40
Table 29. Use Case (1) Login.....	41
Table 30. Use Case (2) Sign Up	41
Table 31. Use Case (3) Withdrawal	42
Table 32. Use Case (4) Course Search.....	43
Table 33. Use Case (5) Course Registration	43
Table 34. Use Case (6) Course	44
Table 35. Use Case (7) Quiz	44
Table 36. Use Case (8) Chatbot	44
Table 37. Use Case (9) Course Upload.....	45
Table 38. Use Case (10) Course Management	45

1. INTRODUCTION

1.1 PURPOSE

본 문서는 "Code with Star" (이하 '본 서비스') 서비스를 개발하기 위해 작성된 소프트웨어 요구 명세서이다. 본 서비스는 2023 년도 1 학기 성균관대학교 소프트웨어공학개론 41 분반 7 팀 (이하 '개발팀')에 의해 고안된 서비스이다. 본 문서에서는 해당 서비스를 제공하는 소프트웨어 시스템의 요구 사항들을 요구 및 분석하였으며, 개발팀은 이에 기재된 내용에 기반하여 본 시스템을 개발한다.

본 문서는 기본적으로 개발팀이 열람하는 것을 목적으로 작성되었다. 예외적으로 교수, 조교를 포함한 성균관대학교 소프트웨어공학개론 41 분반 수강자 전원은 평가 및 참조의 용도로 본 문서를 열람할 수 있다. 본 문서를 학습 및 교육의 목적으로 열람 및 배포하는 것은 자유이나, 상업적 용도로 활용할 시 반드시 개발팀의 허가가 있어야 한다.

본 문서의 목적은 본 서비스를 개발하기 위한 요구사항들을 명세화 하는데 있다. 본 서비스는 웹 어플리케이션 기반으로 구현되어 온라인 프로그래밍 학습 플랫폼을 제작하기 위해 개발되었다. 해당 시스템의 개요와 이를 구성하는 요소인 클라이언트, 서버, 데이터베이스의 사용과 그 관계에 대한 요구 사항을 본 문서에 명시한다.

1.2 SCOPE

본 서비스인 "Code with Star" 는 프로그래밍 지식을 학습하고자 하는 사용자 (이하 '학습자') 에게 양질의 AI 튜터를 제공하는 웹 어플리케이션이다. 학습자는 강의 콘텐츠를 제공받을 뿐만 아니라 질문을 하고 실습 및 퀴즈에 대한 피드백을 받는 등 튜터와 상호작용 하며 프로그래밍 지식을 쌓을 수 있다. 뿐만 아니라 프로그래밍 지식을 전달하고자 하는 사용자 (이하 '교육자') 는 해당 튜터를 생성하기 위한 학습 데이터를 직접 업로드 할 수 있다. 이와 같은 서비스를 제공하기 위해 React 기반의 프론트엔드와 Django 기반의 백엔드 서버를 구축한다.

본 서비스를 통해 학습자는 프로그래밍 능력을 기를 수 있으며, 교육자는 본인이 가지고 있는 지식을 AI 튜터를 통해 보다 활용도 있게 제공할 수 있다. 시스템 엔지니어는 확장성을 고려한 설계를 바탕으로, 추후 본 서비스는 프로그래밍 뿐만 아니라 다양한 분야의 지식을 지원하는 교육 플랫폼으로 확장될 수 있다.

1.3 DEFINITIONS, ACRONYMS, AND ABBREVIATIONS

아래의 표는 본 문서에서 사용된 용어와 그에 대한 설명을 나타낸다.

Table 1. List of Terms and Definitions

Term	Definition
Web Browser	인터넷을 통해 웹 페이지를 검색하고 보여주는 소프트웨어이다. 대표적인 예시로 Google 사의 Chrome 이 있다.
Frontend	사용자가 직접 상호 작용하는 클라이언트의 인터페이스 부분을 의미한다.
Backend	사용자가 보지 못하는 영역인 서버나 데이터베이스 등을 의미한다.
React	User Interface 를 만들기 위한 Javascript 기반의 오픈소스 라이브러리이다.
Django	백엔드 서버를 구축하기 위한 Python 기반의 오픈소스 웹 프레임워크이다.
Git	오픈소스 분산형 버전 제어 시스템으로, 소스코드의 버전 관리와 협업 작업을 지원한다.
OAuth	웹 및 모바일 어플리케이션에서 사용자 인증 및 권한 부여를 처리하기 위한 프로토콜이다.
SQLite	오픈소스 관계형 데이터베이스 관리 시스템(RDBMS) 이다.
Firebase Storage	Firebase 의 클라우드 스토리지 서비스로, 웹 어플리케이션에서 파일을 저장하고, 다운로드 및 공유할 수 있다.

아래의 표는 본 문서에서 사용된 약어와 그에 대한 풀이를 나타낸다.

Table 2. List of Acronyms and Abbreviations

Acronym, Abbreviation	Description
UI / UX	User Interface / User Experience
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
JS	Javascript
JWT	Json Web Token
DB	Database
SQL	Structured Query Language
URL	Uniform Resource Locator
VCS	Version Control System
CPU	Central Processing Unit
RAM	Random Access Memory

1.4 REFERENCES

- IEEE, "830-1998 - IEEE Recommended Practice for Software Requirements Specifications".
<https://ieeexplore.ieee.org/document/720574>
- Team 6, "Team6_Requirement Specification".
https://github.com/skkuse/2022fall_41class_team6
- Facebook, "React".

<https://react.dev/>

- Django Software Foundation, "Django: The web framework for perfectionists with deadlines".

<https://www.djangoproject.com/>

- IETF, "The OAuth 2.0 Authorization Framework"

<https://oauth.net/>

- Google, "Firebase"

<https://firebase.google.com/?hl=ko>

1.5 OVERVIEW

본 문서의 이하 부분은 세 부분 (Chapter 2.~4.) 과 부록으로 구성되어 있다. Chapter 2.에서는 제품의 관점과 제품의 기능, 사용자 특징, 운영 환경 및 제약조건에 등 시스템에 대한 전체적인 설명을 기술한다. Chapter 3.에서는 사용자 인터페이스를 중심으로 시스템과 관련된 여러 인터페이스들을 명시한다. Chapter 4.에서는 시스템의 기능적, 비기능적인 요구사항을 기술한다. 그에 더해 시스템의 흐름을 모델의 형태로 나타내며 시스템 구조를 명시한다. 마지막으로 부록에는 본 문서의 이력 등 추가적인 정보를 기술한다.

2. OVERALL DESCRIPTION

2.1. PRODUCT PERSPECTIVE

다양한 소프트웨어가 산업의 중심으로 떠오르고, 컴퓨팅 사고가 대두되면서 프로그래밍을 공부하려는 사람이 늘어나고 있다. 그만큼 다양한 프로그래밍 교육 서비스들이 등장하고 있으며, 대부분 동영상 강좌 혹은 텍스트 형태의 학습 콘텐츠나, 알고리즘 문제풀이를 제공하고 있다. 온라인 사전 제작 형태의 교육 서비스는 언제든지 다시 볼 수 있다는 장점이 있지만, 상호 피드백이 어렵고 동기 부여가 잘 되지 않아 완강률이 낮다는 단점이 있다.

본 제품은 ChatGPT를 활용한 채팅 인터페이스 기반 프로그래밍 교육 웹서비스로써, 사전 제작 교육 콘텐츠의 한계를 극복하여 사용자에게 색다른 교육/학습 경험을 제공하는 것을 목표로 한다. 사용자는 크게 교육자와 학습자로 나뉘며, 교육자는 본 제품을 통해 학습 콘텐츠를 제작하고 개선하는 사용자를 말하고, 학습자는 교육자가 생산한 콘텐츠를 본 제품의 채팅 인터페이스를 통해 학습하는 사용자를 말한다. 교육자는 콘텐츠 제작 인터페이스를 통해 손쉽게 목차별로 학습 콘텐츠를 제작할 수 있다. 또한, 학습자가 수강한 이력을 바탕으로 콘텐츠의 개선점 등을 레포트로 받아볼 수 있다. 학습자는 채팅 인터페이스와 목차를 통해 교육 콘텐츠를 보고 실습 코딩을 진행하며 실시간으로 질문과 피드백을 받을 수 있다. 이를 통해 교육자는 학습 콘텐츠 제작의 부담을 덜고 학습자는 1:1 과외를 받는 듯한 몰입감 있는 학습 경험을 가질 수 있다.

2.2. PRODUCT FUNCTIONS

2.2.1. USER AUTHENTICATION

사용자는 사용자를 유일하게 식별하는 'ID'와 사용자를 인증하는 '비밀번호'를 통해 로그인하여 서비스를 이용할 자격을 얻을 수 있다. 가입 및 로그인을 진행하지 않은

사용자는 서비스를 이용할 수 없으며, 가입 및 로그인을 요구하는 페이지만을 볼 수 있다. 또한, 가입 진행 시 사용자는 교육자 혹은 학습자 중 하나의 역할을 선택할 수 있다.

2.2.2. COURSE

모든 사용자는 서비스에 등록된 모든 학습 콘텐츠에 접근하고 이용할 수 있다. 별도의 수강신청은 필요하지 않으며 모든 진행사항은 자동으로 저장된다. 학습 페이지는 목차, Q&A 목록, 채팅 인터페이스, 코드 에디터로 구성된다. 사용자는 목차를 통해 특정 주제로 건너뛰거나 복습할 수 있으며, 목차를 선택할 때마다 해당 목차의 진행상황에 맞게 채팅 인터페이스와 코드 에디터가 구성된다. Q&A 목록은 채팅 인터페이스를 통해 수행한 문답의 리스트 형태로 제공되며 자신이 궁금하고 헛갈려 했던 내용을 빠르게 다시 확인할 수 있도록 보조한다. 채팅 인터페이스를 통해 학습 콘텐츠가 제공되며, 순차적으로 진행할 수도 있고, 이전 내용을 더 자세히 설명하기를 요구할 수 있다. 코드 에디터는 실습 코드를 제출하고 평가 받을 수 있는 공간으로써, 채팅 인터페이스의 맥락과 연동되어 사용자의 학습을 돕는다. 모든 진행사항은 목차 별 혹은 전체 초기화가 가능하다.

2.2.3. QUIZZ

모든 사용자는 자신이 현재 수강하고 있는 혹은 수강 완료한 학습 콘텐츠에 대해서 퀴즈를 제공받을 수 있다. 퀴즈는 해당 콘텐츠를 수강한 사용자들이 어려워하거나, 많이 질문한 내용을 토대로 자동으로 (혹은 교육자에 의해) 생성되며, 사용자는 해당 퀴즈를 풀어 봄으로써 자신의 학습을 점검할 수 있다.

2.2.4. COURSE UPLOAD & MANAGEMENT

모든 교육자는 자신의 학습 콘텐츠를 업로드하고, 수정할 수 있다. 학습 콘텐츠를 업로드할 때, 교육자는 목차별로 나눠서 업로드해야 하며, 직접 작성하거나 파일로 올릴 수 있다. 학습 콘텐츠의 수정 역시 같은 방법으로 가능하다. 또한, 일정 이상의 사용자가

해당 콘텐츠를 수강하면, 사용자들이 어려워하거나, 질문을 많이 한 부분에 대한 레포트를 제공받을 수 있다. 교육자는 해당 내용을 토대로 콘텐츠를 개선할 수 있다.

2.3. USER CLASSES AND CHARACTERISTICS

2.3.1. SYSTEM ADMINISTRATOR

시스템 관리자는 본 시스템의 구조를 이해하고 관리할 수 있는 역량을 가진 사람을 말한다. 주로 웹 상의 인터페이스보다는 CUI 혹은 IDE 등을 통해 데이터베이스, 코드 등을 보고 관리한다. 시스템에 문제가 생기면 원인을 파악하고 해결하는 역할을 수행한다.

2.3.2. EDUCATOR

교육자는 본 서비스의 사용자 중 콘텐츠를 업로드하고 관리할 수 있는 권한을 가진 사람을 말한다. 교육자는 본 시스템을 통해 양질의 콘텐츠를 업로드하고 자동으로 생성되는 레포트를 바탕으로 이를 더 개선할 수 있다. 그 외에 학습자가 수행할 수 있는 모든 동작을 수행할 수 있다. 주로 강사나 교사, 교수 혹은 시니어 개발자 등의 전문적인 지식을 갖춘 사람들로 구성된다.

2.3.3. LEARNER

학습자는 학습 콘텐츠를 수강하고 퀴즈를 푸는 등 프로그래밍 학습을 목적으로 본 서비스를 이용하는 사용자를 말한다. 이들은 채팅 인터페이스를 통해 동적으로 상호작용하면서 몰입하며 학습을 수행할 수 있다. 주로 중학생 이상의 프로그래밍 입문을 희망하는 사람으로 구성된다.

2.4. OPERATING ENVIRONMENT

본 제품은 웹서비스로써, 웹 브라우저를 설치 및 실행 가능한 모든 환경에서 사용할 수 있다.

- Mac OS
- Linux
- Windows 10
- Windows 11
- Android
- IOS

2.5. DESIGN AND IMPLEMENTATION CONSTRAINTS

본 시스템의 다음의 제약사항을 준수하여 설계 및 개발된다.

- 사용에 제약이 없는 permissive license 를 가진 오픈소스를 주로 이용한다.
- 사용자에게 익숙하고 사용하기 쉬운 인터페이스를 구축한다.
- 사용자의 편의성을 크게 해치지 않는 선에서 성능을 최대화하는 방향으로 설계한다.
- 사용자에게 불필요한 정보를 요구하지 않는다.
- 실행되지 않을 코드를 작성하지 않는다.

2.6. ASSUMPTION AND DEPENDENCIES

본 서비스는 크로미움 기반의 웹 브라우저(ex. 크롬, 엣지)에서 가장 잘 작동하며 그 외의 웹 브라우저에서도 실행할 수 있으나, 약간의 불안정이 존재할 수 있다.

3. EXTERNAL INTERFACE REQUIREMENTS

3.1 USER INTERFACE

Table 3. User Interface of Login Page

이름	로그인 페이지
목적 및 설명	실제 IP 주소 또는 도메인 이름을 통해 본 시스템에 접근했을 때 사용자가 최초로 접하는 화면이다. 사용자는 해당 화면에서 아이디와 비밀번호를 입력하고 로그인 버튼을 눌러 로그인에 성공하면 본 시스템의 메인 페이지로 redirect 된다.
입력소스, 출력대상	클라이언트, 호스트 서버 호스트 서버, 클라이언트
범위, 정확도, 오차 범위	페이지 전체, HTML 및 데이터 처리 정확도,
구성 단위	웹 페이지
시간 및 속도	사용자 로그인 버튼을 눌렀을 때 클라이언트와 서버 간 통신 속도
기타 입출력 관계	N/A
화면 형식과 구성	1. 좌측에는 사이트 이름과 로고를 확인할 수 있다. 2. 우측에는 학생인지 강사인지 선택할 수 있는 탭과 아이디 및 비밀번호를 입력할 수 있는 창이 있다. 3. 로그인 버튼을 눌러 로그인에 성공하면 메인 페이지로 redirect 된다. 4. 아직 계정이 없는 사용자의 경우 회원가입 텍스트 하이퍼링크를 눌러 회원가입 페이지로 이동한다.

데이터 유형	텍스트, 이미지, 위젯
명령 유형	버튼 및 하이퍼링크 클릭을 통한 코드 실행
종료 메시지	N/A

Figure 1. Design of Login Page

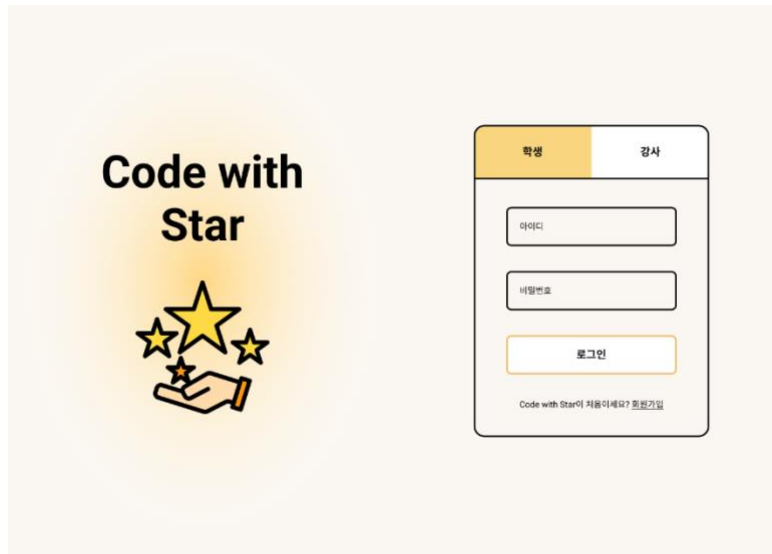


Table 4. User Interface of Sign Up Page

이름	회원이입 페이지
목적 및 설명	로그인 페이지에서 계정이 없는 사용자가 회원가입 하이퍼링크를 눌러 접속할 수 있는 페이지이다. 사용자는 해당 화면에서 아이디, 이메일 계정, 비밀번호, 비밀번호 확인을 입력해 회원가입을 진행할 수 있으며, 회원가입 성공 시 자동으로 로그인 페이지로 redirect 된다.
입력소스, 출력대상	클라이언트, 호스트 서버 호스트 서버, 클라이언트
범위, 정확도,	페이지 전체, HTML 및 데이터 처리 정확도,

오차 범위	
구성 단위	웹 페이지
시간 및 속도	사용자가 회원가입 버튼을 눌렀을 때 클라이언트와 서버 간 통신 속도
기타 입출력 관계	N/A
화면 형식과 구성	<ol style="list-style-type: none"> 1. 학생/강사를 선택할 수 있는 탭과 아이디, 이메일 계정, 두번의 비밀번호를 입력할 수 있는 창이 있다. 2. 회원가입 버튼을 눌러 회원가입에 성공하면 로그인 페이지로 redirect 된다. 3. 이미 계정이 존재하는 사용자의 경우 로그인 텍스트 하이퍼링크를 눌러 로그인 페이지로 이동한다.
데이터 유형	텍스트, 위젯
명령 유형	버튼 및 하이퍼링크 클릭을 통한 코드 실행
종료 메시지	N/A

Figure 2. Design of Sign Up Page

The image shows a sign-up page design. At the top, there are two tabs: '학생' (Student) and '강사' (Teacher). The '학생' tab is selected. Below the tabs, there are four input fields: '아이디' (ID), '이메일' (Email), '비밀번호' (Password), and '비밀번호 확인' (Confirm Password). Below these fields is a '회원가입' (Sign Up) button. At the bottom, there is a link that says '이미 계정이 있으세요? 로그인' (Already have an account? Login).

Table 5. User Interface of Main Page

이름	메인 페이지
목적 및 설명	사용자가 로그인 후에 볼 수 있는 초기 상태의 메인 페이지 이다. 사용자는 전체 강의 리스트 조회 및 여러 세부 기능들을 이용할 수 있다.
입력소스, 출력대상	클라이언트, 시스템 서버 시스템 서버, 클라이언트
범위, 정확도, 오차 범위	페이지 전체 사용자 데이터 처리의 정확도
구성 단위	웹 페이지
시간 및 속도	클라이언트 - 서버 간 통신 속도
기타 입출력 관계	N/A
화면 형식과 구성	<p>1. Header Section : 가장 왼쪽에는 해당 사이트 로고가 위치해있고, 중앙에 위치한 검색바를 통해서 사용자가 원하는 강의를 검색할 수 있다. 또한 사용자는 가장 오른쪽에 위치한 '로그아웃'을 통해 로그아웃이 가능하다.</p> <p>2. 왼쪽 Navigator : 상단에 사용자의 프로필 사진과 계정에 대한 정보가 표시되고, '마이페이지'로 이동할 수 있는 버튼이 있다. 만약 사용자가 '교육자'일 경우, '강의 관리 페이지'로 이동할 수 있는 버튼이 추가로 표시된다. 하단에는 프로그래밍 언어 별로 진행되는 강의들을 볼 수 있도록 언어 선택 메뉴가 표시된다. (파이썬, 자바, C)</p>

	3. 중앙 Contents : '오늘의 퀴즈'를 풀 수 있는 배너가 가장 먼저 위치해있으며, 그 다음으로는 해당 시스템 내 전체 강의 리스트가 나열된다.
데이터 유형	이미지, 텍스트
명령 유형	버튼 클릭을 통한 페이지 이동, 사용자 입력을 통한 검색 기능
종료 메시지	N/A

Figure 3. Design of Main Page

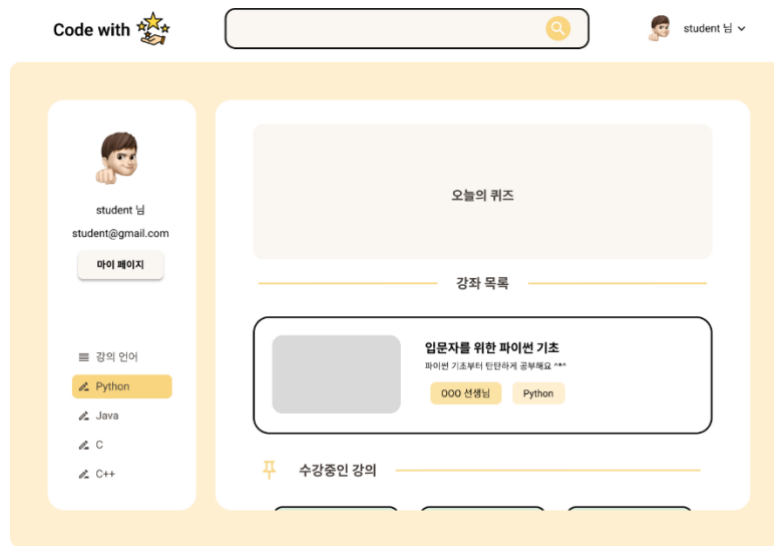


Table 6. User Interface of Course Page

이름	강의수강 페이지
목적 및 설명	<p>사용자가 강의를 수강할 수 있는 페이지이다.</p> <p>사용자는 채팅 interface 를 통해 교육자가 올린 강의 콘텐츠를 마치 교육자와 1:1 과외를 하는 것처럼 학습할 수 있으며 강의 내용과 관련해 궁금한 점을 AI 튜터에게 질문할 수 있다. 또한 AI 튜터가 제공하는 퀴즈 및 예제들을 풀어볼 수 있다.</p>

입력소스, 출력대상	클라이언트, 호스트 서버, 호스트 서버, 클라이언트
범위, 정확도, 오차 범위	페이지 전체, HTML 및 데이터 처리 정확도,
구성 단위	웹 페이지
시간 및 속도	사용자가 전송 및 제출 버튼을 눌렀을 때 클라이언트와 서버 간 통신 속도
기타 입출력 관계	N/A
화면 형식과 구성	<ol style="list-style-type: none"> 1. 상단의 Header 섹션은 메인 페이지와 동일하다. 2. 왼쪽의 Navigator 섹션에는 해당 강의에 대한 목차와 사용자가 AI 튜터에게 질문했던 내용들에 대한 리스트를 확인할 수 있다. 3. 오른쪽 섹션에는 강의 콘텐츠 관련 내용들이 표시된다. 4. 오른쪽 섹션의 상단에는 강의명과 사용자의 진도율이 표시된다. 이때 + 버튼을 누르면 교육자가 작성한 강의에 대한 소개 글을 확인할 수 있다. 5. 오른쪽 섹션의 왼쪽에는 AI 튜터와 소통할 수 있는 채팅 interface가 구현이 되어 있으며 오른쪽에는 AI튜터가 제공하는 퀴즈 및 예제들에 대한 code-based 답변을 작성할 수 있는 Text Box가 존재한다. 제출 버튼을 눌러 답변을 제출하면 정답 여부를 표시한다.
데이터 유형	텍스트, 이미지, 위젯
명령 유형	버튼 및 하이퍼링크 클릭을 통한 코드 실행
종료 메시지	N/A

Figure 4. Design of Course Page

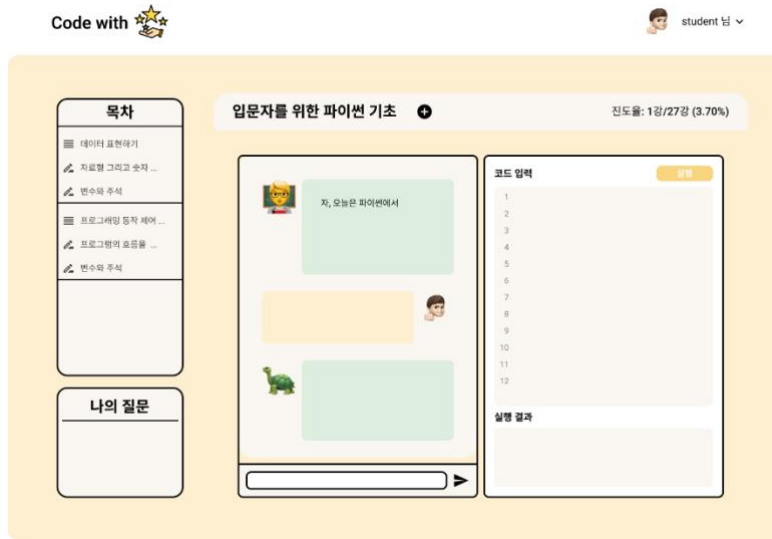


Table 7. User Interface of My Page

이름	마이 페이지
목적 및 설명	사용자는 자신이 수강 중인 강의 현황을 확인하고, 사용자 진도에 맞는 퀴즈를 제공 받을 수 있다.
입력소스, 출력대상	클라이언트, 호스트 서버 호스트 서버, 클라이언트
범위, 정확도, 오차 범위	페이지 전체, HTML 및 데이터 처리 정확도,
구성 단위	웹 페이지
시간 및 속도	사용자가 전송 및 제출 버튼을 눌렀을 때 클라이언트와 서버 간 통신 속도
기타 입출력 관계	AI 를 이용한 사용자 맞춤 퀴즈 생성 및 출력

화면 형식과 구성	<p>1. 상단 Header 와 왼쪽 Naviagtor Section 은 메인 페이지와 동일하다.</p> <p>2. 중앙 Content Section : 상단에는 사용자가 수강 중인 강의 현황을 확인 할 수 있다. 수강 중인 강의의 간략한 정보와 진도율이 표시된다. 해당 강의를 클릭하면 자신이 수강했던 부분부터 공부를 시작할 수 있다. 하단에는 사용자가 학습한 부분들에 대해서 생성된 퀴즈들이 제공된다.</p>
데이터 유형	텍스트, 이미지, 위젯
명령 유형	버튼 클릭을 통한 페이지 이동
종료 메시지	N/A

Figure 5. Design of My Page



Table 8. User Interface of Quiz Page (Execute)

이름	퀴즈 페이지(실행)
목적 및 설명	사용자는 해당 사이트에서 제공하는 퀴즈들을 자유롭게 풀어볼 수 있다.
입력소스, 출력대상	클라이언트, 시스템 서버 시스템 서버, 클라이언트
범위, 정확도, 오차 범위	사용자 입력 코드 각 언어 interpreter 와 사용자 코드에 따른 정확도 각 언어 interpreter 와 사용자 코드에 따른 오차
구성 단위	웹 페이지
시간 및 속도	사용자가 '실행' 버튼을 눌렀을 때 실행 코드의 복잡도 및 클라이언트 - 서버 간 통신 속도
기타 입출력 관계	해당 코드를 실행시키는 interpreter 결과를 사용자에게 표시 해당 interpreter 가 실행할 수 없거나 코드에 오류가 발생할 경우 오류 결과를 사용자에게 표시
화면 형식과 구성	1. 왼쪽 상단 배너에는 퀴즈에 대한 소개와 언어 선택 드롭다운이 표시 된다. 2. 왼쪽 하단에는 강의 수강 페이지의 레이아웃과 동일한 AI 와의 채팅 인터페이스로 구성되어 있고, 퀴즈 문제를 제공한다. 3. 사용자는 오른쪽에 위치한 코드 에디터를 활용하여 문제를 해결할 수 있다. 상단에 위치한 '실행' 버튼을 통해 사용자가 작성한 코드를 실행하면, 하단 실행 결과 창에 코드 결과가 출력된다.
데이터 유형	텍스트
명령 유형	버튼 클릭을 통한 코드 실행
종료 메시지	N/A

Figure 6. Design of Quiz Page (Execute)

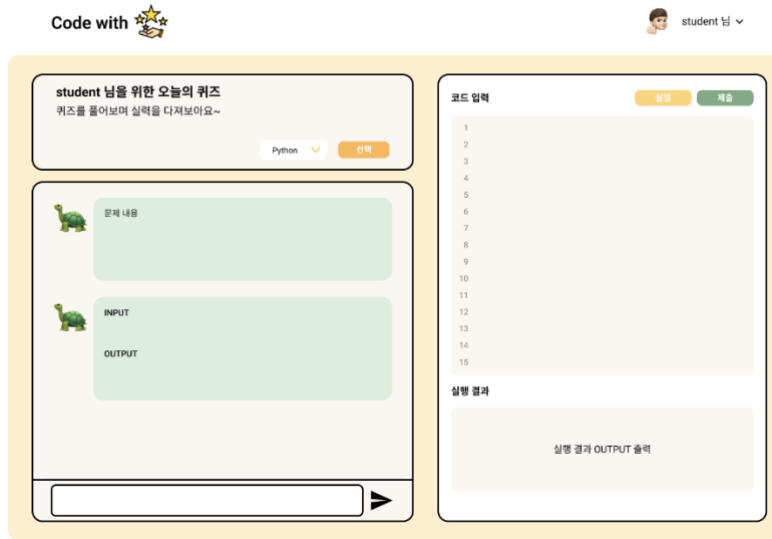


Table 9. User Interface of Quiz Page (Submit)

이름	퀴즈 페이지(제출)
목적 및 설명	사용자는 해당 사이트에서 제공하는 퀴즈를 풀어보고 제출하여 채점을 받을 수 있다.
입력소스, 출력대상	클라이언트, 시스템 서버 시스템 서버, 클라이언트
범위, 정확도, 오차 범위	사용자 제출 코드 각 언어 interpreter와 사용자 코드에 따른 정확도 각 언어 interpreter와 사용자 코드에 따른 오차
구성 단위	웹 페이지
시간 및 속도	사용자가 '제출' 버튼을 눌렀을 때 실행 코드의 복잡도 및 클라이언트 - 서버 간 통신 속도
기타 입출력 관계	해당 코드를 실행시키는 interpreter 결과를 사용자에게 표시 해당 interpreter가 실행할 수 없거나 코드에 오류가 발생할 경우 오류 결과를 사용자에게 표시

화면 형식과 구성	<ol style="list-style-type: none"> 1. 왼쪽 상단 배너에는 퀴즈에 대한 소개와 언어 선택 드롭다운이 표시 된다. 2. 왼쪽 하단에는 강의 수강 페이지의 레이아웃과 동일한 AI와의 채팅 인터페이스로 구성되어 있고, 퀴즈 문제를 제공한다. 3. 사용자는 오른쪽에 위치한 코드 에디터를 이용하여 문제에 대한 답을 작성할 수 있다. 상단의 '제출' 버튼을 누르면 사용자가 작성한 코드를 제출하고, 코드 제출 결과는 왼쪽 채팅 인터페이스를 통해 알 수 있다.
데이터 유형	텍스트
명령 유형	버튼 클릭을 통한 코드 실행
종료 메시지	N/A

Figure 7. Design of Quiz Page (Submit)

Table 10. User Interface of Uploading Lectures

이름	강의생성 페이지
목적 및 설명	사용자가 새로운 강의를 생성할 수 있는 페이지이다. 사용자는 강의명을 입력하고 간단한 소개를 작성해 콘텐츠를 구성하여 학습자에게 제공할 강의를 생성할 수 있다.
입력소스, 출력대상	클라이언트, 호스트 서버 호스트 서버, 클라이언트
범위, 정확도, 오차 범위	페이지 전체, HTML 및 데이터 처리 정확도,
구성 단위	웹 페이지
시간 및 속도	사용자가 선택 및 제출 버튼을 눌렀을 때 클라이언트와 서버 간 통신 속도
기타 입출력 관계	N/A
화면 형식과 구성	<ol style="list-style-type: none"> 1. 상단의 Header에는 메인 페이지와 동일하다. 2. 왼쪽의 Navigator 섹션에서는 사용자의 프로필 이미지, 아이디, 이메일 주소를 확인할 수 있으며 누르면 마이페이지로 접속할 수 있는 마이페이지 버튼이 있다. 3. 오른쪽 섹션의 상단에는 취소 버튼과 저장 버튼이 존재한다. 사용자는 취소 버튼을 누르면 강의 생성하는 것을 중단하고 강의목록 페이지로 돌아갈 수 있다. 저장 버튼을 눌러 저장하기가 완료 되면 강의목록 페이지로 redirect 된다. 4. 사용자는 오른쪽 섹션에서 강의명을 입력하고 강의가 어떤 내용에 관한 것인지 선택할 수 있으며 강의에 대해 간단한 소개 글을 작성할 수 있다. 5. 사용자가 대단원 추가 버튼을 누르면 대단원 추가 모달창이 나타나는데 여기에는 대단원 제목을 작성할 수 있는 창이 있다. 사용자는 취소 버튼을 눌러 대단원을 추가하는 것을 멈출 수 있으

	<p>며 저장 버튼을 눌러 대단원을 추가할 수 있다. 대단원이 추가가 되면 작성한 내용이 강의목록에 업데이트 된다. 삭제 버튼을 통해 이를 삭제할 수 있다.</p> <p>6. 각 대단원마다 소단원 추가 버튼이 존재한다. 사용자가 소단원 추가 버튼을 누르면 소단원 제목과 콘텐츠를 작성할 수 있는 모달창이 나타난다. 취소 버튼을 눌러 소단원을 추가하는 것을 멈출 수 있으며 저장 버튼을 눌러 소단원을 추가할 수 있다. 추가된 소단원은 강의목록에 업데이트 되며 삭제 버튼을 통해 이를 삭제할 수 있다.</p>
데이터 유형	텍스트, 이미지, 위젯
명령 유형	버튼 클릭을 통한 API 호출 및 코드 실행
종료 메시지	N/A

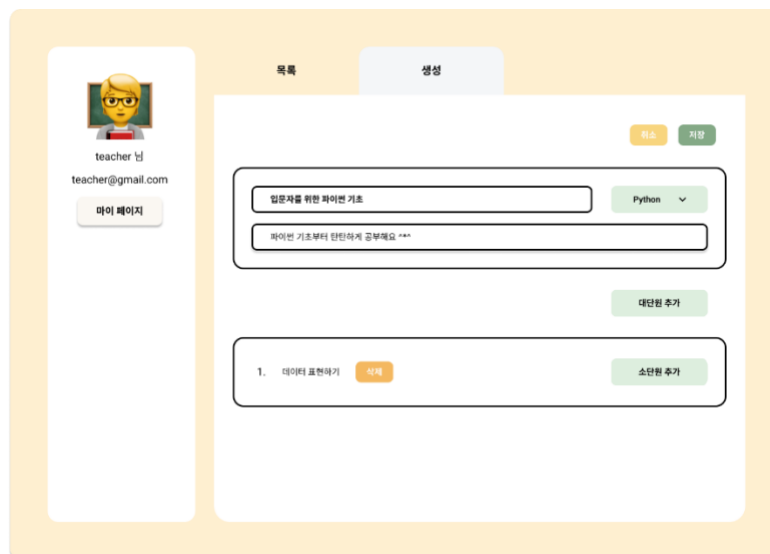
Figure 8. Design of Uploading Lectures (Initial State)

Code with Star - Requirement Specification

Figure 9. Design of Modal Window for Adding Chapters



Figure 10. Design of Uploading Lectures (Added Chapters)



Code with Star - Requirement Specification

Figure 11. Design of Modal Window for Adding Sections



Figure 12. Design of Uploading Lectures (Added Sections)

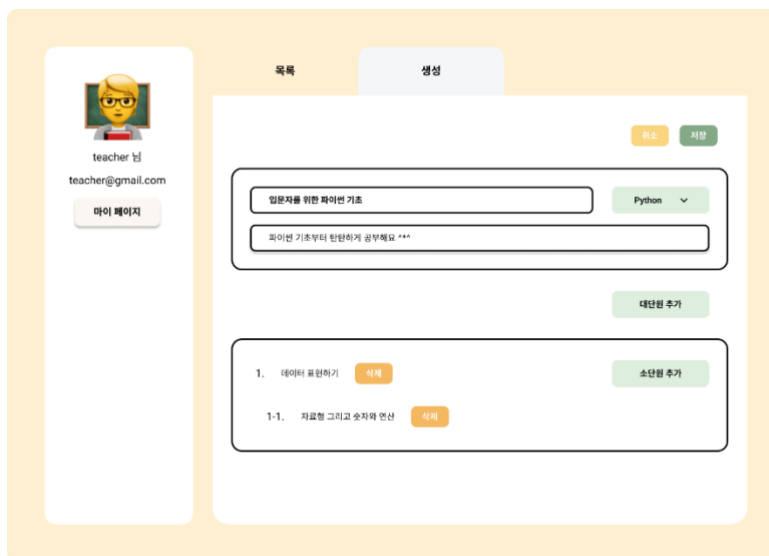


Table 11. User Interface of Course Syllabus

이름	강의목록 페이지
목적 및 설명	사용자가 생성한 강의들을 목록을 확인할 수 있는 페이지이다. 사용자는 수정 및 삭제 버튼을 눌러 강의를 수정하거나 삭제할 수 있으며 AI가 제공하는 강의에 대한 피드백을 확인할 수 있다.
입력소스, 출력대상	클라이언트, 호스트 서버 호스트 서버, 클라이언트
범위, 정확도, 오차 범위	페이지 전체, HTML 및 데이터 처리 정확도,
구성 단위	웹 페이지
시간 및 속도	사용자가 수정 또는 삭제 버튼을 눌렀을 때 클라이언트와 서버 간 통신 속도
기타 입출력 관계	N/A
화면 형식과 구성	<ol style="list-style-type: none"> 1. 상단의 Header 섹션은 메인 페이지와 동일하다. 2. 왼쪽의 Navigator 섹션은 강의생성 페이지와 동일하다. 3. 오른쪽 섹션의 상단에는 목록 탭과 생성 탭이 있다. 생성 탭을 누르면 강의를 추가할 수 있는 강의생성 페이지로 이동할 수 있다. 4. 목록 탭에서는 사용자가 생성한 강의들의 목록을 확인할 수 있다. 각 강의마다 수정, 삭제, 피드백 버튼이 있다. 사용자는 수정 버튼을 눌러 강의를 수정할 수 있는 페이지로 이동할 수 있으며, 삭제 버튼을 통해 강의를 삭제할 수 있다. 피드백 버튼을 누르면 해당 강의에 대한 피드백을 확인할 수 있다. 피드백에는 학습자들이 어떠한 어려움을 겪고 있는지와 같은 내용들이 포함된다.
데이터 유형	텍스트, 이미지, 위젯
명령 유형	버튼 클릭을 통한 코드 실행

Code with Star - Requirement Specification

종료 메시지	N/A
-----------	-----

Figure 13. Design of Course List

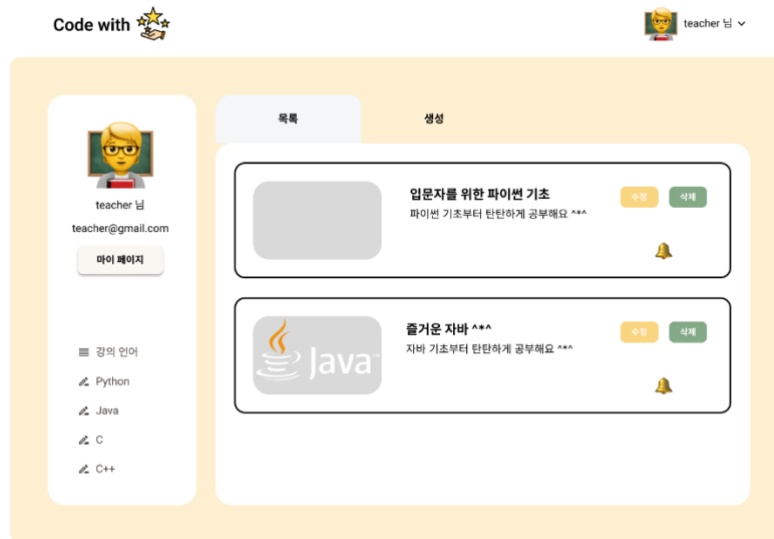


Figure 14. Design of Course List (After Feedback)

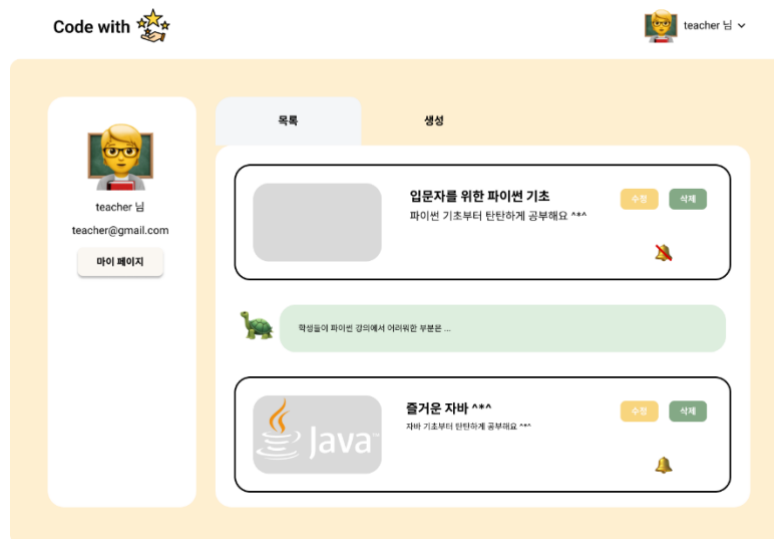


Table 12. User Interface of Course Modification

이름	강의수정 페이지
목적 및 설명	사용자가 강의의 콘텐츠를 수정할 수 있는 페이지이다. 사용자는 추가, 수정, 삭제 버튼을 눌러 각각 콘텐츠를 추가하거나 수정 또는 삭제할 수 있다. 저장 버튼을 눌러 콘텐츠 수정에 성공하면 강의 목록 페이지로 redirect 된다.
입력소스, 출력대상	클라이언트, 호스트 서버 호스트 서버, 클라이언트
범위, 정확도, 오차 범위	페이지 전체, HTML 및 데이터 처리 정확도,
구성 단위	웹 페이지
시간 및 속도	사용자가 각종 버튼을 눌렀을 때 클라이언트와 서버 간 통신 속도
기타 입출력 관계	N/A
화면 형식과 구성	1. 왼쪽의 Navigator 섹션은 강의생성 페이지와 동일하다. 2. 오른쪽 섹션의 상단에는 취소 버튼과 저장 버튼이 존재한다. 사용자는 취소 버튼을 누르면 강의 수정하기를 취소하고 강의목록 페이지로 돌아갈 수 있다. 저장 버튼을 눌러 저장하기가 완료되면 강의목록 페이지로 redirect 된다. 3. 오른쪽 섹션에는 사용자는 강의 제목, 언어, 소개 내용을 수정할 수 있다. 4. 오른쪽 섹션의 하단은 대단원 및 소단원 제목을 수정할 수 있다는 점 외에 강의 생성 페이지와 동일하다.
데이터 유형	텍스트, 이미지, 위젯
명령 유형	버튼 클릭을 통한 코드 실행

종료 메시지	N/A
-----------	-----

Figure 15. Design of Course Modification

3.2 HARDWARE INTERFACE

Table 13. Hardware Interface of Client

이름	Web Browser
목적 및 설명	HTTP 를 통한 네트워크 통신이 가능한 웹 브라우저와, 기본 입출력 (모니터, 키보드, 마우스)이 가능한 기기를 통해 해당 시스템을 이용할 수 있다.

Table 14. Hardware Interface of Server

이름	AWS Cloud Server
목적 및 설명	아마존의 클라우드 컴퓨팅 서비스를 이용하여 서버를 구축한다.

구성	Amazon EC2 인스턴스를 구동하여 Frontend 서버와 Backend 서버를 동작 시킨다. Backend 서버에는 데이터베이스 SQLite 가 내장되어 있다.
----	--

3.3 SOFTWARE INTERFACE

Table 15. Software Interface of Client

이름	사용자 소프트웨어 인터페이스	
Chrome Browser	목적 및 설명	사용자가 해당 서비스를 이용하기 위한 기본 환경
	입력	마우스, 키보드 등의 입력 장치로 데이터 입력
	출력	HTML, CSS, JS 렌더링 페이지
Python	목적 및 설명	사용자가 해당 서비스가 제공하는 코드 채점 기능을 사용하기 위한 환경
	입력	사용자의 입력 코드
	출력	실행 결과 및 오류 출력

Table 16. Software Interface of Server

이름	서버 소프트웨어 인터페이스	
React	목적 및 설명	서비스 UI/UX 제공을 위한 front-end 프레임 워크
	입력	서비스 UI/UX 기능 구현을 위한 코드 파일, 사용자 입력
	출력	HTML, CSS, JS 렌더링 페이지
Django	목적 및 설명	서비스 기능 동작을 위한 Back-end 프레임 워크
	입력	서비스 기능 구현을 위한 코드 파일, 사용자 입력 데이터
	출력	JSON 형식의 REST 형식

Open AI (ChatGPT)	목적 및 설명	사용자의 질문에 대한 답변 및 코드, 학습 콘텐츠에 대한 피드백 기능을 위한 API
	입력	사용자 입력 데이터
	출력	입력 데이터에 대한 결과값

3.4 COMMUNICATION INTERFACE

Table 17. HTTP Request Format for Login

이름	로그인 HTTP 요청
용도	사용자가 로그인을 위해 자신의 정보를 서버로 전송
Method	POST
URL	main.com/login
Content Type	text/application
내용	사용자의 계정 및 비밀번호
서버 동작	입력받은 사용자의 계정과 비밀번호를 데이터베이스에 저장된 데이터와 비교한 후 로그인 성공/실패 여부 판단하여 페이지 응답
인코딩	Base64 인코딩 및 전송

Table 18. HTTP Request Format for Sign Up

이름	회원가입 HTTP 요청
용도	사용자가 회원가입을 위해 자신의 정보를 서버로 전송
Method	POST
URL	main.com/signup
Content Type	text/application
내용	사용자 계정, 비밀번호, 이메일, 회원 유형(학습자, 교육자)

서버 동작	입력받은 사용자의 계정과 이메일을 데이터베이스에 저장된 데이터와 비교한 후 회원가입의 성공/실패 여부 판단하여 페이지 응답
인코딩	Base64 인코딩 및 전송

Table 19. HTTP Request Format for Withdrawal

이름	회원탈퇴 HTTP 요청
용도	사용자가 회원탈퇴를 요청
Method	POST
URL	main.com/expireuser
Content Type	text/application
내용	사용자 계정 정보
서버 동작	사용자의 계정과 관련된 모든 데이터를 데이터베이스에서 삭제한 후 회원탈퇴 성공 페이지 응답
인코딩	Base64 인코딩 및 전송

Table 20. HTTP Request Format for Course Search

이름	강의검색 HTTP 요청
용도	사용자가 수강하고 싶은 강의를 단어와 필터링을 통해 검색하여 해당하는 강의 목록을 출력
Method	POST
URL	main.com/search
Content Type	text/application
내용	사용자가 입력한 단어와 설정한 필터링 값
서버 동작	사용자가 입력한 단어와 설정한 필터링에 해당하는 강의를 데이터베이스에서 탐색 후 강의 목록 출력
인코딩	Base64 인코딩 및 전송

Table 21. HTTP Request Format for Course List Lookup

이름	수강 강의 목록 조회 HTTP 요청
용도	사용자가 수강했던 강의를 다시 수강하거나 이어서 수강하기 위해 수강 중이던 강의 목록을 요청
Method	GET
URL	main.com/userpage
Content Type	multipart/form-data
내용	사용자가 수강했던 강의 정보(강의 내용, 강의 진도)
서버 동작	사용자의 정보를 토대로 사용자가 수강 중인 강의를 데이터베이스에서 탐색하여 목록으로 출력
인코딩	Base64 인코딩 및 전송

Table 22. HTTP Request Format for Course

이름	강의수강 HTTP 요청
용도	사용자가 강의 수강을 위해 강의를 선택
Method	POST
URL	main.com/learner/lectureid
Content Type	multipart/form-data
내용	사용자가 선택한 강의 정보
서버 동작	사용자가 선택한 강의에 포함된 콘텐츠를 데이터베이스에서 탐색하여 페이지 응답 사용자의 강의 수강 기록을 데이터베이스에 기록
인코딩	Base64 인코딩 및 전송

Table 23. HTTP Request Format for Quiz

이름	퀴즈참여 HTTP 요청
용도	사용자가 퀴즈에 참여하여 퀴즈를 해결
Method	GET, POST
URL	main.com/dailyquiz
Content Type	multipart/form-data
내용	사용자의 정보, 오늘의 퀴즈 정보
서버 동작	24 시간을 주기로 바뀌는 퀴즈 정보를 데이터베이스에서 탐색하여 페이지 응답 사용자가 제출한 퀴즈의 답안을 데이터베이스에 저장된 퀴즈의 정답과 비교하여 정답/오답 판별 후 페이지 응답
인코딩	Base64 인코딩 및 전송

Table 24. HTTP Request Format for Chatbot

이름	챗봇 사용 HTTP 요청
용도	사용자가 질의 응답 및 code 전송을 위해 챗봇을 사용
Method	POST
URL	main.com/chatAI (별도의 배너 사용)
Content Type	text/application
내용	사용자가 질의한 텍스트 및 code
서버 동작	사용자가 요청한 질의에 대해서 openAI API 를 사용하여 chatGPT 로 redirection 을 하고 chatGPT 의 응답을 사용자에게 전달
인코딩	Base64 인코딩 및 전송

Table 25. HTTP Request Format for My Page

이름	마이 페이지 HTTP 요청
용도	사용자가 자신의 정보를 조회, 수정
Method	GET, POST
URL	main.com/userpage
Content Type	text/application
내용	사용자의 회원 정보(계정, 비밀번호, 이메일)
서버 동작	사용자의 현재 회원 정보를 출력 사용자가 현재 회원 정보를 수정 요청할 경우 새로운 회원 정보에 대해서 데이터베이스에서 탐색 후 수정 가능/불가능 여부 판단 후 페이지 응답
인코딩	Base64 인코딩 및 전송

Table 26. HTTP Request Format for Uploading Lectures

이름	강의생성 HTTP 요청
용도	교육자가 자신의 강의를 제작 및 업로드
Method	POST
URL	main.com/
Content Type	multipart/form-data
내용	교육자가 업로드하는 강의 콘텐츠(영상, 코드, 텍스트, 파일)
서버 동작	교육자가 제작하고 업로드한 강의를 데이터베이스에 저장
인코딩	Base64 인코딩 및 전송

Table 27. HTTP Request Format for Lecture Management

이름	강의관리 HTTP 요청
용도	교육자가 자신이 업로드한 강의 정보를 수정하거나 수강 중인 학생을 관리하기 위해 요청
Method	GET, POST
URL	main.com/educator/lectureid
Content Type	multipart/form-data
내용	수정한 강의 정보, 해당 강의를 수강중인 학생 정보(진도, 성취도)
서버 동작	교육자가 수정한 강의 정보를 데이터베이스에 update 교육자가 자신의 강의를 수강 중인 학생들의 데이터(진도, 성취도)를 출력
인코딩	Base64 인코딩 및 전송

4. SYSTEM REQUIREMENTS

4.1. FUNCTIONAL REQUIREMENTS

4.1.1. USE CASE

Table 28. Function Use Case

Function No.	Name of Function	Function Description
F1	로그인	사용자가 회원가입한 정보를 사용하여 시스템에 로그인 할 수 있다.
F2	회원 가입	사용자가 기존에 등록되지 않은 개인정보를 사용하여 시스템에 등록할 수 있다.
F3	회원 탈퇴	사용자가 시스템에 등록된 자신의 개인정보를 삭제할 수 있다.
F4	강의 검색	사용자가 단어와 필터링을 통해 강의를 검색할 수 있다.
F5	수강 강의 목록 조회	사용자가 수강 중인 강의의 목록을 조회할 수 있다.
F6	강의 수강	사용자가 강의를 선택하여 수강할 수 있다.
F7	퀴즈 참여	사용자가 24 시간을 주기로 바뀌는 퀴즈에 참여할 수 있다.
F8	챗봇 사용	사용자가 챗봇을 통해 질의응답을 할 수 있다.
F9	강의 생성	사용자(교육자)가 강의를 생성하고 업로드할 수 있다.
F10	강의 관리	사용자(교육자)가 생성한 강의를 관리할 수 있다.

Table 29. Use Case (1) Login

Name	로그인
Actors	사용자(교육자, 학습자)
Description	사용자가 회원가입한 정보를 사용하여 시스템에 로그인 할 수 있다.
Data	사용자의 아이디, 비밀번호
Stimulus	사용자가 로그인 버튼을 누른다.
Response	로그인에 실패하면 상황에 따른 알림 창을 띄운다. 로그인에 성공하면 메인 페이지로 이동한다.
Comments	(1). 입력한 사용자의 아이디가 데이터베이스에 존재하는 지 검사한다. (2). (1)이 참일 경우(존재할 경우) 입력한 사용자의 비밀번호가 데이터베이스에 존재하는 비밀번호와 일치하는 지 검사한다. -(1)이 거짓일 경우 존재하지 않는 회원 정보라고 알림 창을 띄워준다. -(2)가 거짓일 경우 비밀번호가 일치하지 않는다고 알림 창의 띄워준다. -(2)가 참일 경우 로그인에 성공하여 메인 페이지로 이동한다.

Table 30. Use Case (2) Sign Up

Name	회원 가입
Actors	사용자(교육자, 학습자)
Description	사용자가 기존에 등록되지 않은 개인정보를 사용하여 시스템에 등록할 수 있다.

Data	사용자의 아이디, 비밀번호, 이메일, 회원유형(교육자/학습자)
Stimulus	사용자가 회원가입 버튼을 누른다.
Response	회원 가입에 실패하면 상황에 따른 알림 창을 띄운다. 회원 가입에 성공하면 메인 페이지로 이동하여 로그인을 유도한다.
Comments	(1). 입력한 사용자의 아이디가 데이터베이스에 존재하는 지 검사한다. (2). (1)이 거짓일 경우 나머지 정보가 모두 알맞게 채워 졌는지 검사한다. -(1)이 참일 경우(존재할 경우) 이미 존재하는 회원이라고 알림 창을 띄운다 -(2)이 거짓일 경우 나머지 정보가 모두 알맞게 채워지지 않았다고 알림 창의 띄워준다. -(2)가 참일 경우 해당 정보를 데이터베이스에 추가하고 메인 페이지로 이동한 뒤 로그인을 유도한다.

Table 31. Use Case (3) Withdrawal

Name	회원 탈퇴
Actors	사용자(교육자, 학습자)
Description	사용자가 시스템에 등록된 자신의 개인정보를 삭제할 수 있다.
Data	사용자의 아이디, 비밀번호, 이메일, 회원유형(교육자/학습자)
Stimulus	사용자가 회원탈퇴 버튼을 누른다.

Response	시스템에서 사용자와 관련된 모든 정보를 삭제한다.
----------	-----------------------------

Table 32. Use Case (4) Course Search

Name	강의 검색
Actors	사용자(교육자, 학습자)
Description	사용자가 단어와 필터링을 통해 강의를 검색할 수 있다.
Data	사용자가 입력한 키워드, 설정한 필터링, 데이터베이스에 저장된 강의 목록
Stimulus	사용자가 키워드와 필터링을 통해서 검색한다.
Response	사용자가 입력한 키워드와 설정한 필터링으로 데이터베이스 상에 존재하는 강의 목록을 출력한다.

Table 33. Use Case (5) Course Registration

Name	수강 강의 목록 조회
Actors	사용자(교육자, 학습자)
Description	사용자가 수강 중인 강의 목록을 출력해준다.
Data	사용자의 정보, 사용자가 수강 중인 강의 정보, 사용자의 수강 강의 진도
Stimulus	사용자가 마이페이지 버튼을 누른다.
Response	사용자가 수강 중인 강의 목록과 각 강의의 진도를 표시해준다.

Table 34. Use Case (6) Course

Name	강의 수강
Actors	사용자(교육자, 학습자)
Description	사용자가 강의를 선택하여 수강할 수 있다.
Data	사용자의 정보, 수강하고자 하는 강의 정보
Stimulus	사용자가 자신의 수강 중인 강의 목록에서 수강하고자 하는 강의를 클릭한다.
Response	수강하고자 하는 강의 학습 페이지를 출력한다.

Table 35. Use Case (7) Quiz

Name	퀴즈 참여
Actors	사용자(교육자, 학습자)
Description	사용자가 24 시간을 주기로 바뀌는 퀴즈에 참여할 수 있다.
Data	사용자의 정보, 24 시간 마다 바뀌는 퀴즈 정보
Stimulus	사용자가 상단의 퀴즈 배너를 클릭한다.
Response	퀴즈를 풀 수 있는 페이지를 출력한다.

Table 36. Use Case (8) Chatbot

Name	챗봇 사용
Actors	사용자(교육자, 학습자)
Description	사용자가 챗봇을 통해 질의응답을 할 수 있다.
Data	사용자의 정보, 사용자가 챗봇에게 질의한 텍스트 및 code

Stimulus	챗봇에게 질의할 수 있는 텍스트 박스에 질의할 내용 및 code 를 입력한 뒤 버튼을 클릭한다.
Response	사용자가 질의한 내용에 대해 chatGPT 의 답변을 출력한다.
Comments	

Table 37. Use Case (9) Course Upload

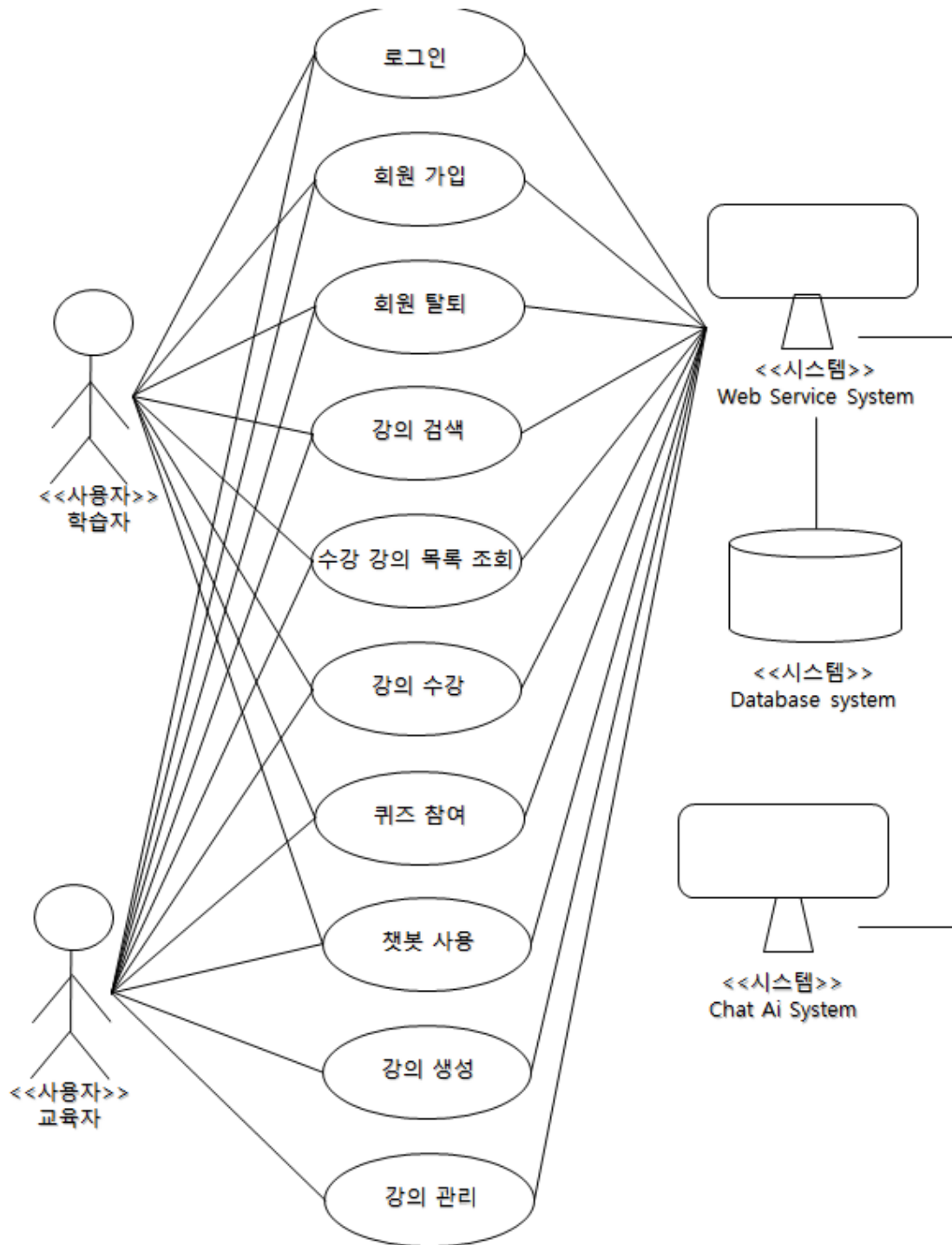
Name	강의 생성
Actors	사용자(교육자)
Description	사용자(교육자)가 강의를 생성하고 업로드할 수 있다.
Data	사용자(교육자) 정보, 강의 구성 데이터
Stimulus	사용자(교육자)가 강의를 직접 구성한 뒤 강의 업로드 버튼을 클릭한다.
Response	사용자(교육자)가 제작한 강의정보가 데이터베이스에 등록된다.

Table 38. Use Case (10) Course Management

Name	강의 관리
Actors	사용자(교육자)
Description	사용자(교육자)가 생성한 강의를 관리할 수 있다.
Data	사용자(교육자) 정보, 강의 정보
Stimulus	사용자(교육자)가 자신이 제작한 강의 리스트에서 관리하고자 하는 강의를 클릭한다.
Response	사용자(교육자)가 해당 강의 정보를 수정할 수 있는 페이지를 출력한다.

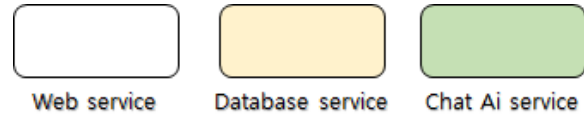
4.1.2. USE CASE DIAGRAM

Figure 16. Use Case Diagram



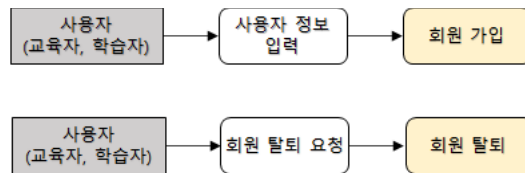
4.1.3. DATA FLOW DIAGRAM

Figure 17. Data Flow Label



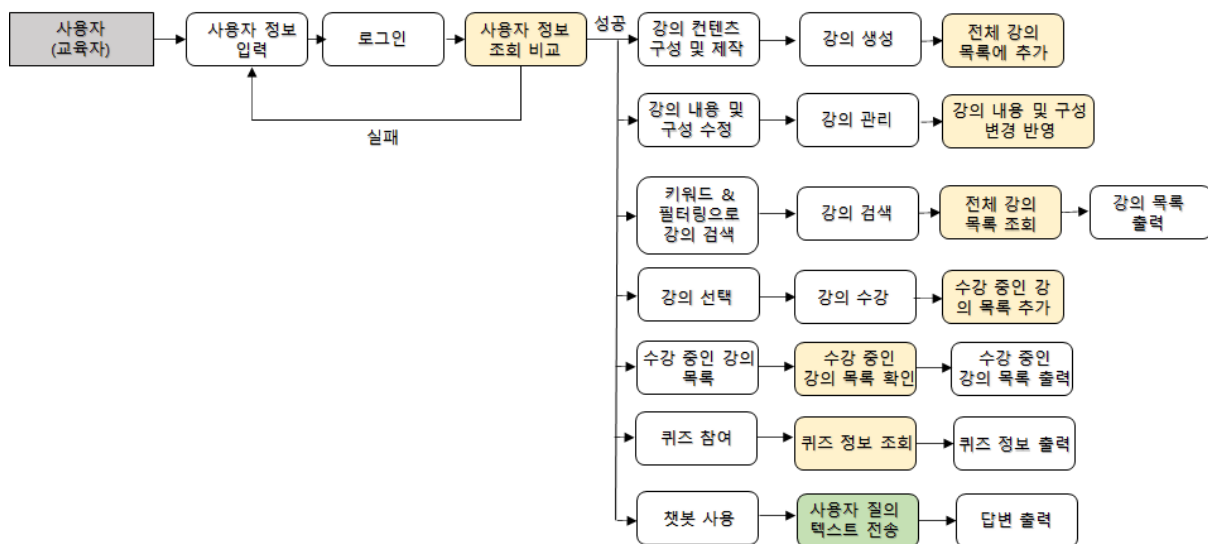
a. 회원 관리

Figure 18. User Management Data Flow Diagram



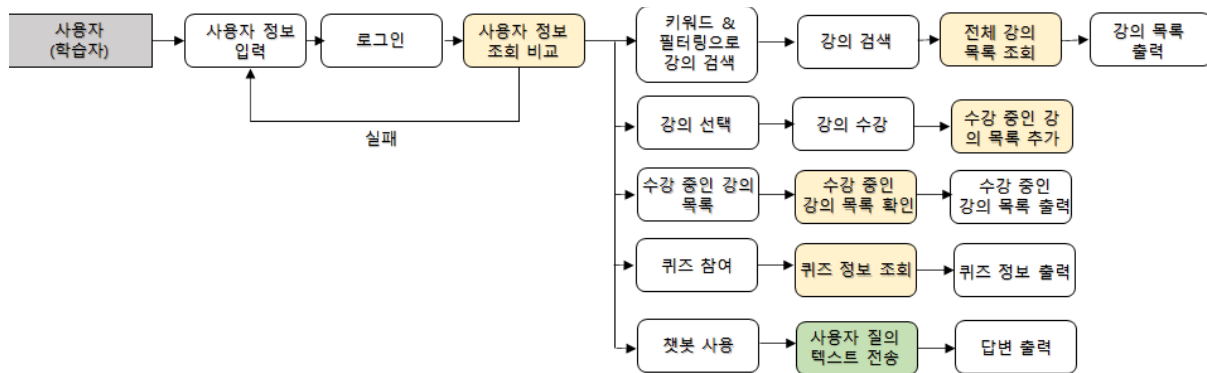
b. 교육자

Figure 19. Educator Data Flow Diagram



c. 학습자

Figure 20. Learner Data Flow Diagram



4.2. NON FUNCTIONAL REQUIREMENTS

4.2.1. PRODUCT REQUIREMENT

본 시스템은 사용자에게 서비스를 제공하면서 다음과 같은 요구조건을 충족해야 한다.

A. USABILITY REQUIREMENT

본 시스템을 사용하는 사용자(학습자, 교육자)는 별도의 훈련없이 시스템의 주요 기능들을 유용하게 사용할 수 있어야 한다. 챗봇 기능은 어떤 상황이나 특정 페이지에 제약을 받지 않고 언제나 사용가능 하도록 UI 를 배치해야 한다.

B. PERFORMANCE REQUIREMENT

- 본 시스템에서 사용자와 Database 간 데이터 교환은 1 초 이내에 완료되어야 한다.

- 본 시스템에서 사용자가 질의한 내용에 대해 챗봇은 5 초 이내에 답변을 출력해야 한다.

C. DEPENDABILITY REQUIREMENT

- 본 시스템에서 사용자가 제공한 사용자 관련 데이터는 모두 안전하게 Database 에 저장되어야 한다.
- 본 시스템에서 학습자가 또 다른 학습자의 개인 데이터를 임의로 열람하지 못하도록 해야 한다.

4.2.2. ORGANIZATIONAL REQUIREMENT

A. ENVIRONMENTAL REQUIREMENT

PC 를 통해 웹 페이지에 접속하는 것을 권장한다.

B. DEVELOPMENT REQUIREMENT

Front End 의 경우 React framework 를 사용하며 디자인은 Figma 로 제작한다. Back End 의 경우 Django framework 를 사용한다.

AI 챗봇의 경우 openAI 의 API 를 활용하여 chatGPT 의 기능을 활용한다.

Database 와 storage 의 경우 Firebase 를 사용한다.

4.2.3. EXTERNAL REQUIREMENT

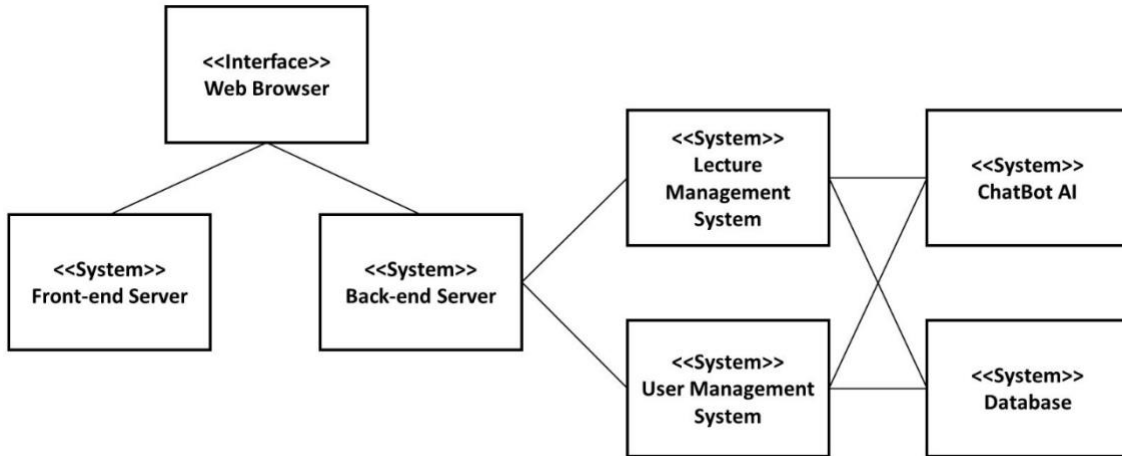
A. ETHICAL/SAFETY/SECURITY REQUIREMENT

- 사용자가 설정한 비밀번호는 안전하게 hash 하여 Database 에 저장한다.
- 학습자가 다른 학습자의 개인 정보를 열람할 수는 없도록 한다.
- 교육자는 학습자가 개인정보이용동의 후에 학습자의 질의응답, 학습 상황, 코드 제출물 등을 열람할 수 있다.

4.3. ORGANIZING SYSTEM FLOW

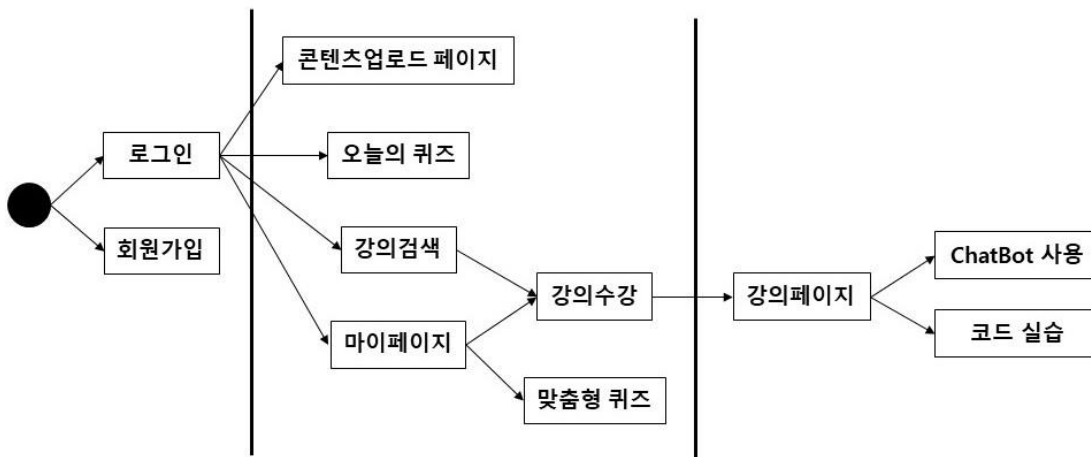
4.3.1. CONTEXT MODEL

Figure 21. Context Model



4.3.2. PROCESS MODEL

Figure 22. Process Model



4.3.3. INTERACTION MODEL

4.1.2. Use Case Diagram 참고

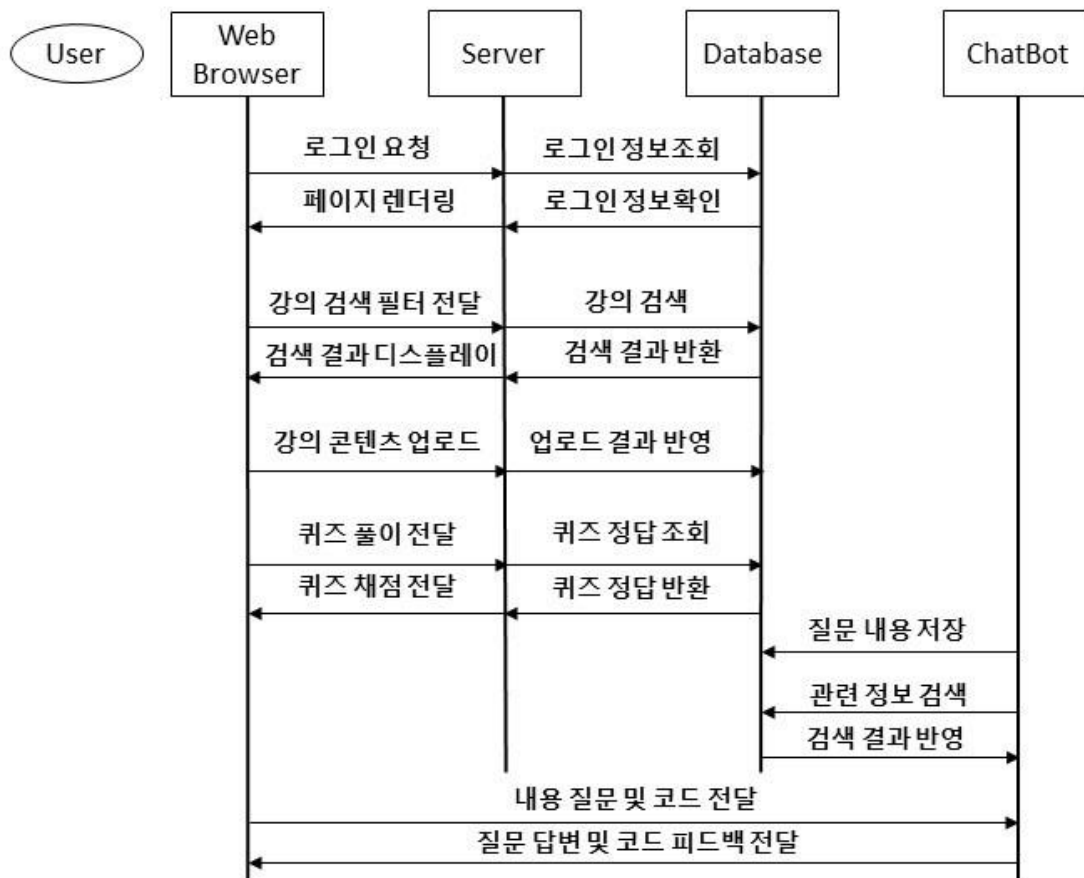
4.3.4. BEHAVIOR MODEL

A. DATA FLOW DIAGRAM

4.1.3. Data Flow Diagram 참고

B. SEQUENCE DIAGRAM

Figure 23. Sequence Diagram



4.4. SYSTEM ARCHITECTURE

Figure 24. System Architecture

