



Code with Star

Design Specification

Team 7

2018310561 유영석
2018311895 김정식
2020314754 오민재
2018312511 조예현
2018314737 김명수
2017310734 김수한

2023.05.19

CONTENTS

1. INTRODUCTION.....	7
1.1 OBJECTIVES	7
1.2 APPLIED DIAGRAMS	7
1.2.1 UML.....	7
1.2.1.1 STATE DIAGRAM	7
1.2.1.2 USE CASE DIAGRAM.....	8
1.2.1.3 SEQUENCE DIAGRAM	8
1.2.1.4 CLASS DIAGRAM	8
1.2.2 CONTEXT DIAGRAM.....	8
1.3 APPLIED TOOLS	9
1.3.1 VISUAL STUDIO CODE.....	9
1.3.2 FIGMA.....	9
1.4 SYSTEM OVERVIEW.....	9
2. SYSTEM ARCHITECTURE - OVERALL	10
2.1 OBJECTIVES	10
2.2 SYSTEM ORGANIZATION	10
2.2.1 CONTEXT DIAGRAM.....	11
2.2.2 SEQUENCE DIAGRAM	11
2.2.3 USE CASE DIAGRAM	12
3. SYSTEM ARCHITECTURE – FRONT END.....	12
3.1 OBJECTIVES	12
3.2 MAIN PAGE	12
3.2.1 ATTRIBUTES.....	13
3.2.2 METHODS.....	14
3.2.3 STATE DIAGRAM	16
3.3.1 ATTRIBUTES.....	17
3.3.2 METHODS.....	18
3.3.3 STATE DIAGRAM	19
3.4 CREATE LECTURE PAGE	20

3.4.1 ATTRIBUTES.....	20
3.4.2 METHODS.....	21
3.4.3 STATE DIAGRAM.....	22
3.5 MANAGE LECTURE PAGE	22
3.5.1 ATTRIBUTES.....	23
3.5.2 METHODS.....	24
3.5.3 STATE DIAGRAM.....	25
3.6.1 ATTRIBUTES.....	26
3.6.2 METHODS.....	27
3.6.3 STATE DIAGRAM.....	29
3.7. QUIZ PAGE	29
3.7.1 ATTRIBUTES.....	30
3.7.2 METHODS.....	30
3.7.3 STATE DIAGRAM.....	31
4. SYSTEM ARCHITECTURE – BACK END	31
4.1 OBJECTIVES	31
4.2 OVERALL ARCHITECTURE.....	32
4.3 SUB SYSTEMS.....	33
4.3.1 USER MANAGEMENT SYSTEM	33
4.3.2 COURSE MANAGEMENT SYSTEM	34
4.3.3 FEEDBACK SYSTEM	35
5. PROTOCOL	36
5.1 OBJECTIVES	36
5.2 HTTPS.....	36
5.3 JSON.....	36
5.4 INTERFACE.....	37
5.4.1 LECTURE LIST INQUIRY	37
5.4.2 LECTURE INFO INQUIRY	38
5.4.3 QUIZ LIST INQUIRY.....	39
5.4.4 QUIZ INFO INQUIRY.....	39
5.4.5 LECTURE CREATE	40
5.4.6 LECTURE INFO MODIFY	41

5.4.7 CHATBOT MESSAGE	42
5.4.8 CODE SEND & EXECUTE.....	42
6. DATABASE DESIGN	43
6.1 DJANGO MODEL	43
6.2 RELATIONAL SCHEMA	48
7. TESTING PLAN.....	49
7.1 OBJECTIVE	49
7.2 TESTING POLICY	49
7.2.1 DEVELOPMENT TESTING	49
7.2.2 RELEASE TESTING	49
7.2.3 USER TESTING.....	50
7.3 TEST CASE	50
7.3.1 LOGIN & SIGN UP	50
7.3.2 LECTURE SEARCH	51
7.3.3 VIEW LECTURE REGISTERED	51
7.3.4 LECTURE REGISTER.....	52
7.3.5 QUIZ PARTICIPATION	52
7.3.6 CHATBOT USAGE.....	53
7.3.7 LECTURE MANAGEMENT.....	53
8. DEVELOPMENT PLAN	54
8.1 OBJECTIVE	54
8.2 DEVELOPMENT TIMELINE	54

LIST OF FIGURES

Figure 1. System Organization.....	10
Figure 2. Context Diagram.....	11
Figure 3. Sequence Diagram.....	11
Figure 4. Use Case Diagram.....	12
Figure 5. Main Page State Diagram.....	16
Figure 6. User Page (Learner) State Diagram.....	19
Figure 7. User Page (Educator) State Diagram.....	20
Figure 8. Create Lecture Page State Diagram	22
Figure 9. Manage Lecture Page State Diagram	25
Figure 10. Learning Lecture Page State Diagram	29
Figure 11. Quiz Page State Diagram	31
Figure 12. Overall Architecture	32
Figure 13. User Sequence Diagram	33
Figure 14. Course Sequence Diagram.....	34
Figure 15. Feedback System Class Diagram.....	35
Figure 16. Feedback System Sequence Diagram	35
Figure 17. Relational Schema.....	48
Figure 18. Development Timeline	54

LIST OF TABLES

Table 1. Lecture List Inquiry Request Protocol.....	37
Table 2. Lecture List Inquiry Response Protocol	37
Table 3. Lecture Info Inquiry Request Protocol.....	38
Table 4. Lecture List Inquiry Response Protocol	38
Table 5. Quiz List Inquiry Request Protocol.....	39
Table 6. Quiz List Inquiry Response Protocol	39
Table 7. Quiz Info Inquiry Request Protocol.....	39

Table 8. Quiz Info Inquiry Response Protocol	40
Table 9. Lecture Create Request Protocol	40
Table 10. Lecture Create Response Protocol	41
Table 11. Lecture Info Modify Request Protocol	41
Table 12. Lecture Info Modify Response Protocol.....	42
Table 13. Chatbot Message Request Protocol	42
Table 14. Chatbot Message Response Protocol.....	42
Table 15. Code Send & Execute Request Protocol	43
Table 16. Code Send & Execute Response Protocol.....	43
Table 17. Course Model	43
Table 18. User Model.....	44
Table 19. Tag Model	44
Table 20. Chapter Model	44
Table 21. CourseRoom Model	45
Table 22. ChapterRoom Model.....	45
Table 23. ChapterChatData Model.....	45
Table 24. Feedback Model.....	46
Table 25. Quiz Model	46
Table 26. QuizRoom Model.....	46
Table 27. QuizChatData Model.....	46
Table 28. Analysis Model	47
Table 29. User Login	50
Table 30. Lecture Search Test	51
Table 31. View Lecture List.....	51
Table 32. Lecture Register Test.....	52
Table 33. Quiz Participation Test.....	52
Table 34. Chatbot Usage Test.....	53
Table 35. Lecture Management Test	53

1. INTRODUCTION

본 문서는 “Code with Star” (이하 ‘본 서비스’)에 대한 디자인 명세서이다. 본 문서는 본 서비스의 요구사항 명세서를 기반으로 서비스에 필요한 요소 및 요소들 간의 관계를 정의한다.

1.1 OBJECTIVES

이 장에서는 시스템 설계에 적용된 모델과 다이어그램 및 사용된 개발 도구를 소개한다.

1.2 APPLIED DIAGRAMS

시스템의 구성요소를 정의하고 전체적인 구조를 설계하고 표현하기 위해 몇 가지 diagram 을 사용했다.

1.2.1 UML

UML(Unified Modeling Language)는 소프트웨어 공학에서 사용되는 표준화된 범용 모델링 언어다. OMG(Object Management Group)에서 고안하고 관리하고 있으며 이를 사용하여 시스템의 동작 및 구조를 시각적으로 나타낼 수 있다.

1.2.1.1 STATE DIAGRAM

State diagram 은 시스템이 가질 수 있는 여러 state 들과 그들 간의 transition 을 나타낸다. 본 명세서에서는 시스템 사용자인 강의자와 수강자가 발생시킨 event 들에 대하여 시스템이 어떻게 반응하는지를 표현하기 위해 사용하였다.

1.2.1.2 USE CASE DIAGRAM

Use case diagram 은 actor 와 use case 에 대한 시스템의 상호작용을 나타낸다. Actor 는 시스템의 구성요소나 사용자 뿐만 아니라 외부의 다른 시스템도 포함된다. Use case 는 actor 가 시스템을 사용하는 시나리오다. Actor 와 use case 는 요구사항 명세서에서 정의되었으며 actor 는 사람의 모습으로, use case 는 타원의 형태로 표현한다.

1.2.1.3 SEQUENCE DIAGRAM

Sequence diagram 은 시스템을 이루는 여러 객체들 간의 상호작용을 그 순서와 함께 전체적으로 나타낸다. 상호작용은 객체 간 전달하는 데이터 및 영향을 주는 기능들을 중심으로 표현된다. 본 명세서에서는 시스템 내부에서 사용자가 전달 및 요청한 데이터가 서버와 스토리지, 그리고 ChatBot 을 연동한 외부 API 와 어떠한 방식으로 연결되는지를 표현하기 위해 사용하였다.

1.2.1.4 CLASS DIAGRAM

Class diagram 은 시스템을 구성하는 class 와 class 의 속성, 기능을 정의하고 class 간의 관계를 나타낸다. Class 는 시스템을 구성하는 하나의 객체로 class 의 이름, 속성, 기능(method) 정보를 포함하고 있다. Class 간의 상속, 위계 관계를 정의하여 각각의 class 가 전체 시스템을 어떻게 구성하는 지 표현한다.

1.2.2 CONTEXT DIAGRAM

Context diagram 은 시스템의 구성 요소들 및 시스템과 외부 entity 들 간의 관계를 간략하게 보여주는 다이어그램이다. 이 diagram 을 통해서는 시스템을 구성하는 하위 요소들을 구조적으로 파악하고 그들이 외부 entity 들과 어떠한 관계를 갖는지를 볼 수 있다.

1.3 APPLIED TOOLS

1.3.1 VISUAL STUDIO CODE

마이크로소프트에서 개발한 텍스트 에디터이다. 대부분의 OS 에서 사용할 수 있으며 다양한 확장 기능을 통해 여러 언어 지원, git 연동 등의 기능을 사용할 수 있다. 프론트엔드, 백엔드 개발에 사용되었다.

1.3.2 FIGMA

웹 페이지의 UI/UX 디자인을 고안하기 위해 사용된 도구이다. 웹 페이지 기반 서비스이며 팀원 간 동시 접근, 수정 및 공유가 가능하다.

1.4 SYSTEM OVERVIEW

본 서비스는 프로그래밍 지식을 학습하고자 하는 사용자 (이하 '학습자') 에게 양질의 AI 튜터를 제공하는 웹 어플리케이션이다. 또한 프로그래밍 지식을 전달하고자 하는 사용자 (이하 '교육자') 는 학습 데이터를 직접 업로드하여 튜터를 생성할 수 있다. 이와 같은 서비스를 제공하기 위해 React 기반의 프론트엔드와 Django 기반의 백엔드 서버를 구축한다.

2. SYSTEM ARCHITECTURE - OVERALL

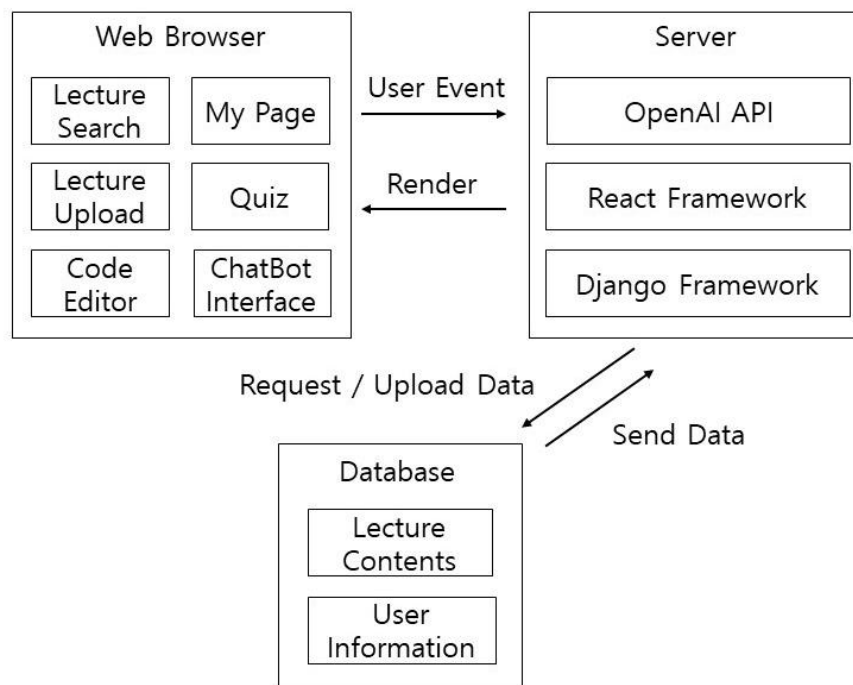
2.1 OBJECTIVES

이 장에서는 시스템의 전체적인 구조를 diagram 을 사용하여 설명한다.

2.2 SYSTEM ORGANIZATION

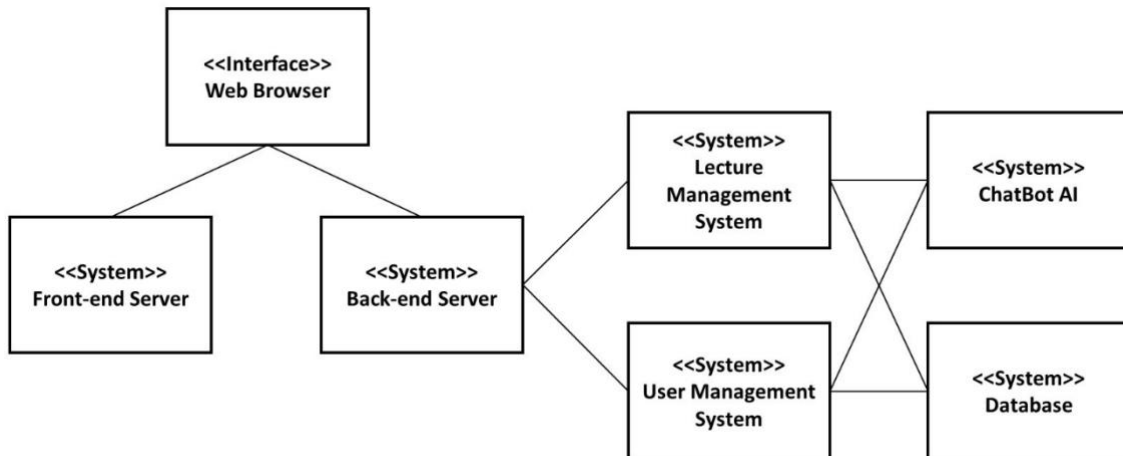
본 서비스는 Client-Server 모델을 적용하여 설계되었다. Frontend 는 client 가 상호작용할 수 있는 웹 페이지 기반의 어플리케이션으로 HTTP 프로토콜을 사용하여 backend 와 데이터를 주고 받는다. Frontend 어플리케이션은 react 프레임워크를 사용하여 개발했다. Backend 는 client 가 요청한 request 에 대해서 알맞은 service 를 client 에게 제공해주는 server 역할의 어플리케이션이다. Backend 어플리케이션은 경우에 따라 OpenAI API 를 사용하여 ChatGPT 의 서비스를 사용하거나 Firebase 기반의 database 와 통신하여 필요한 데이터를 요청, 처리할 수 있다. Backend 어플리케이션은 Django 를 사용하여 개발했다.

Figure 1. System Organization



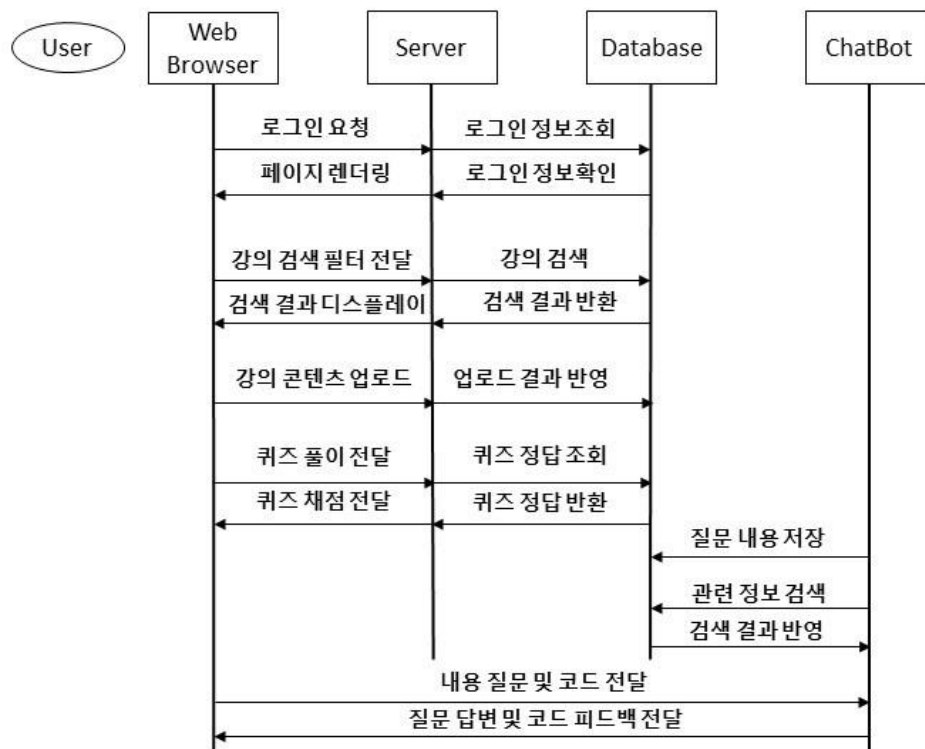
2.2.1 CONTEXT DIAGRAM

Figure 2. Context Diagram



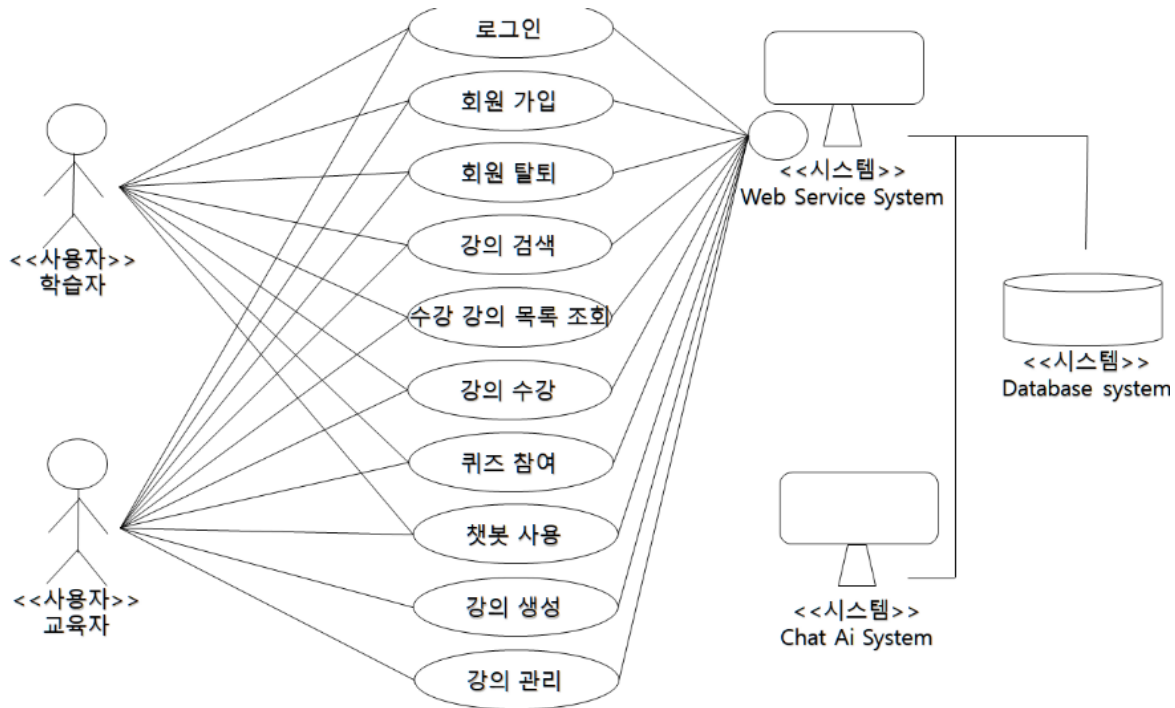
2.2.2 SEQUENCE DIAGRAM

Figure 3. Sequence Diagram



2.2.3 USE CASE DIAGRAM

Figure 4. Use Case Diagram



3. SYSTEM ARCHITECTURE – FRONT END

3.1 OBJECTIVES

이 장에서는 사용자와 상호작용할 수 있는 시스템의 웹 페이지 종류와 각 페이지의 기능, 목적을 정의하고 각 페이지를 구성하고 있는 세부 구성요소를 설명한다. 또한 각 페이지 간의 관계를 설명한다.

3.2 MAIN PAGE

본 시스템의 main page 이며 사용자가 정상적으로 로그인을 진행했음을 가정한다. 이 페이지에서는 키워드를 통해 시스템에 저장되어 있는 강의를 검색할 수 있고 오늘의 퀴즈, 수강 중인 강의 목록, 시스템에 저장되어 있는 강의 목록을 확인할 수 있다. 또한

로그아웃 버튼을 통해 로그아웃을 할 수 있고 마이페이지 버튼을 통해 마이페이지로 이동하여 회원의 정보를 확인할 수 있다.

3.2.1 ATTRIBUTES

Main page 를 구성하는 세부 구성요소와 그 속성은 다음과 같다.

(1) Header: Main page 의 header 부분으로, main page 로 이동할 수 있는 본 시스템의 로고, 강의 검색 bar, 로그아웃 버튼을 포함한다.

- Logo: main page 로 이동할 수 있는 로고 버튼이다.
- SearchInput: 사용자가 강의를 검색할 수 있는 검색 bar 다.
- SearchWord: 사용자가 SearchInput 에 입력한 키워드다.
- Search: 사용자가 입력한 키워드로 강의 검색을 수행하는 버튼이다.
- Logout: 사용자가 로그아웃 할 수 있는 버튼이다.

(2) NavBar: Main page 의 좌측에 세로 형태로 배치되어 있으며 회원의 이름, 계정, 프로필 사진 등의 간략한 정보와 마이페이지 버튼을 포함한다.

- UserName: 로그인한 사용자의 이름이다.
- userEmail: 로그인한 사용자의 이메일이다.
- UserImg: 로그인한 사용자의 프로필 사진이다.
- UserPage: 사용자의 마이페이지로 이동할 수 있는 버튼이다.

(3) QuizBanner: Main page 의 body 의 상단 부분으로 24 시간 마다 갱신되는 퀴즈의 정보를 출력하며 사용자는 이 banner 를 통해 오늘의 퀴즈에 참여할 수 있다.

- QuizInfo: 퀴즈의 정보이다.
- UserQuizStatus: 사용자에게 대한 해당 퀴즈의 정답 및 완료 여부 정보이다.
- QuizJoin: 사용자가 퀴즈에 참여할 수 있는 버튼이다.

(4) LectureList: 시스템에 저장된 강의 목록 일부를 출력하며 사용자가 특정 키워드를 통해 검색한 경우 검색한 키워드를 필터링하여 강의 목록을 출력한다.

- LectureId: 강의 고유 식별 번호이다.
- LectureName: 강의의 이름이다.
- LectureDescription: 강의에 대한 간략한 설명이다.
- LectureEducator: 강의를 제작한 교육자의 이름(Id)이다.
- LectureLanguage: 강의에서 사용되는 프로그래밍 언어 정보다.
- LectureImg: 강의의 대표 사진이다.
- LectureStart: 강의를 수강할 수 있는 버튼이다.

(5) UserLectureList: 사용자가 수강 중인 강의의 목록이다.

- LectureId: 강의 고유 식별 번호이다.
- LectureName: 강의의 이름이다.
- LectureProgress: 사용자의 해당 강의 진도율이다.
- LectureStart: 강의를 수강할 수 있는 버튼이다.

3.2.2 METHODS

Main page 를 구성하는 각 요소의 method 는 다음과 같다.

(1) Header

- clickLogo(): 사용자가 Logo 버튼을 눌렀을 때 main page 로 이동한다.
- getSearchWord(): 사용자가 search bar 에 입력한 단어를 반환한다.
- clickSearch(): 사용자가 search bar 의 검색 버튼을 눌렀을 때 search bar 에 입력한 단어를 반환하여 LectureList 의 method 인 getLectureList()의 인자로 전달한다.
- clickLogout(): 사용자가 로그아웃 버튼을 눌렀을 때 사용자의 session 을 삭제하고 로그인 페이지로 이동한다.

(2) NavBar

- getUsername(): 데이터베이스에서 사용자의 이름 정보를 가져온다.
- setUsername(): getUsername()으로 얻은 사용자의 이름 정보를 출력한다.
- getEmail(): 데이터베이스에서 사용자의 Email 정보를 가져온다.
- setUserEmail(): getEmail()으로 얻은 사용자의 Email 정보를 출력한다.
- clickUserPage(): 사용자가 마이페이지 버튼을 눌렀을 때 사용자의 마이페이지로 이동한다.

(3) QuizBanner

- getQuizInfo(): 데이터베이스에서 오늘의 퀴즈 정보를 가져온다.
- setQuizInfo(): getQuizInfo()에서 얻은 퀴즈 정보를 출력한다.
- setQuizStatus(): 퀴즈에 대한 사용자의 정답 및 완료 여부를 출력한다.
- clickQuizJoin(): 사용자가 퀴즈 참여 버튼을 눌렀을 때 퀴즈 참여 페이지로 이동한다.

(4) LectureList

- getLectureList([word]): 데이터베이스에서 강의 정보를 json 형식으로 가져온 후 해당 데이터를 setLectureList()로 전달한다.
- setLectureList(): 전달받은 json 형식의 데이터를 LectureName, LectureDescriptor, LectureEducator, LectureLanguage, LectureImg 로 parsing 하여 각각 setLectureName(), setLectureDescriptor(), setLectureEducator(), setLectureLanguage(), setLectureImg()로 전달한다.
- setLectureName(): 전달받은 LectureName 을 출력한다.
- setLectureDescriptor(): 전달받은 LectureDescriptor 을 출력한다.
- setLectureEducator(): 전달받은 LectureEducator 을 출력한다.
- setLectureLanguage(): 전달받은 LectureLanguage 을 출력한다.
- setLectureImg(): 전달받은 LectureImg 을 출력한다.
- clickLectureStart(): 해당 강의를 수강할 수 있는 페이지로 이동한다.

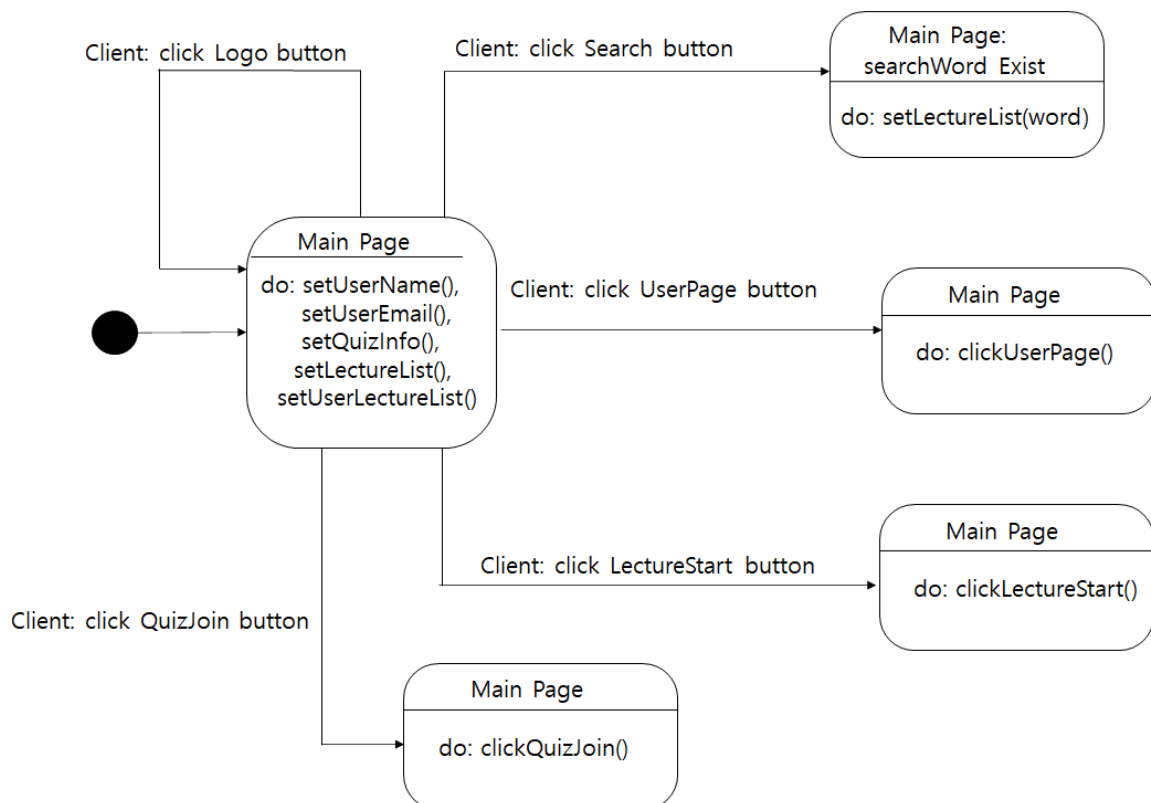
(5) UserLectureList

- getUserLectureList(): 데이터베이스에서 사용자가 수강 중인 강의 정보를 json 형식으로 가져온 후 해당 데이터를 setUserLectureList()로 전달한다.
- setUserLectureList(): 전달받은 json 형식의 데이터를 LectureName, LectureProgress 로 parsing 하여 각각 setUserLectureName(), setUserLectureProgress()로 전달한다.
- setUserLectureName(): 전달받은 LectureName 을 출력한다.
- setUserLectureProgress(): 전달받은 LectureProgress 를 출력한다.
- clickLectureStart(): 해당 강의를 수강할 수 있는 페이지로 이동한다.

3.2.3 STATE DIAGRAM

Main page 의 state diagram 은 다음과 같다.

Figure 5. Main Page State Diagram



3.3 USER PAGE

본 시스템의 사용자의 마이페이지(userpage, mypage)이며 사용자의 회원 종류(학습자, 교육자)에 따라 기능과 구성요소에 차이가 있다. 교육자의 user page 의 경우 교육자가 업로드한 자신의 강의를 관리할 수 있으며 새로운 강의를 제작하는 것이 가능하다.

3.3.1 ATTRIBUTES

Learner page 를 구성하는 세부 구성요소와 그 속성은 다음과 같다.

(1) Header: Main page 의 header[3.2.1-(1)]와 동일하다. (단, search bar 는 user page 에서 사용되지 않는다.)

(2) NavBar

- NavBar(학습자): Main page 의 NavBar[3.2.1-(2)]와 동일하다. (단, UserPage 버튼은 user page 에서 사용되지 않는다.)

- NavBar(교육자): 기본적으로 Main page 의 NavBar[3.2.1-(2)]와 동일하며 새로운 강의를 제작할 수 있는 페이지로 이동하는 버튼을 추가적으로 포함한다. (단, UserPage 버튼은 user page 에서 사용되지 않는다.)

- CreateLecture: 사용자(교육자)가 새로운 강의를 제작할 수 있는 페이지로 이동하는 버튼이다.

(3) UserLectureList

- UserLectureList(학습자): Main page 의 UserLectureList[3.2.1-(5)]와 동일하다.

- UserLectureList(교육자): 기본적으로 Main page 의 UserLectureList[3.2.1-(5)]와 동일하며 각 강의에 대해서 강의 정보를 수정할 수 페이지로 이동하는 버튼을

- 추가적으로 포함한다. - ManageLecture: 사용자(교육자)가 자신의 강의 정보를 수정할 수 있는 페이지로 이동하는 버튼이다.

(4) UserQuizList: 사용자가 본 서비스에서 참여한 퀴즈의 전체 목록을 정답 여부와 함께 출력하며 해당 퀴즈에 참여할 수 있는 버튼을 포함한다.

- QuizInfo: 퀴즈의 정보이다.
- UserQuizStatus: 사용자의 해당 퀴즈에 대한 정답 여부 정보이다.
- QuizJoin: 사용자가 퀴즈에 참여할 수 있는 버튼이다.

3.3.2 METHODS

(1) Header

Main page 의 header[3.2.2-(1)]와 동일하다. (단, getSearchWord()와 clickSearch()는 user page 에서 사용되지 않는다.)

(2) NavBar

- NavBar(학습자): Main page 의 NavBar [3.2.2-(2)]와 동일하다. (단, clickUserPage()는 user page 에서 사용되지 않는다.)
- NavBar(교육자): 기본적으로 Main page 의 NavBar[3.2.1-(2)]와 동일하며 새로운 강의를 제작할 수 있는 페이지로 이동하는 버튼을 추가적으로 포함한다. (단, clickUserPage()는 user page 에서 사용되지 않는다.)
- clickCreateLecture(): 사용자(교육자)가 새로운 강의를 제작할 수 있는 페이지로 이동한다.

(3) UserLectureList

- UserLectureList(학습자): Main page 의 UserLectureList [3.2.2-(5)]와 동일하다.
- UserLectureList(교육자): 기본적으로 Main page 의 UserLectureList[3.2.1-(5)]와 동일하며 각 강의에 대해서 강의관리를 할 수 페이지로 이동하는 버튼을 추가적으로 포함한다.

- clickManageLecture(): 사용자(교육자)가 자신의 강의 정보를 수정할 수 있는 페이지로 이동한다.

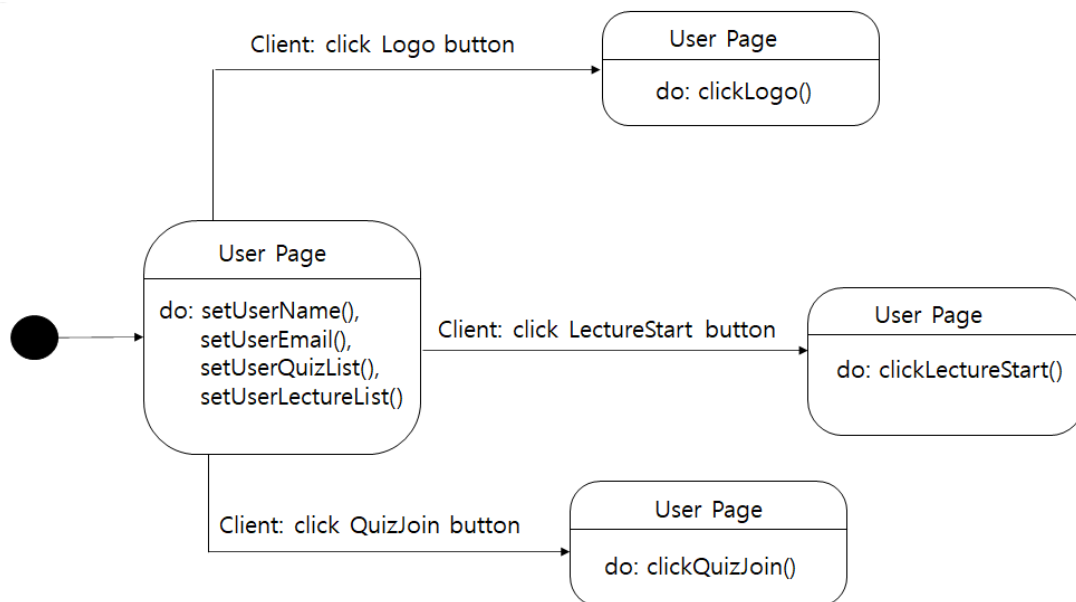
(4) UserQuizList

- getUserQuizList(): 데이터베이스에서 사용자가 참여한 모든 퀴즈의 목록과 그 정보를 json 형식으로 가져온 후 setUserQuizList()로 전달한다.
- setUserQuizList(): 전달받은 json 형식의 데이터를 각각의 퀴즈별로 QuizInfo, UserQuizStatus 로 parsing 하여 각각 setQuizInfo(), setQuizStatus()로 전달한다.
- setQuizInfo(): setUserQuizList()에서 얻은 퀴즈 정보를 출력한다.
- setQuizStatus(): setUserQuizList()에서 얻은 각각의 퀴즈에 대한 사용자의 정답여부를 출력한다.
- clickQuizJoin(): 사용자가 퀴즈 참여 버튼을 눌렀을 때 퀴즈 참여 페이지로 이동한다.

3.3.3 STATE DIAGRAM

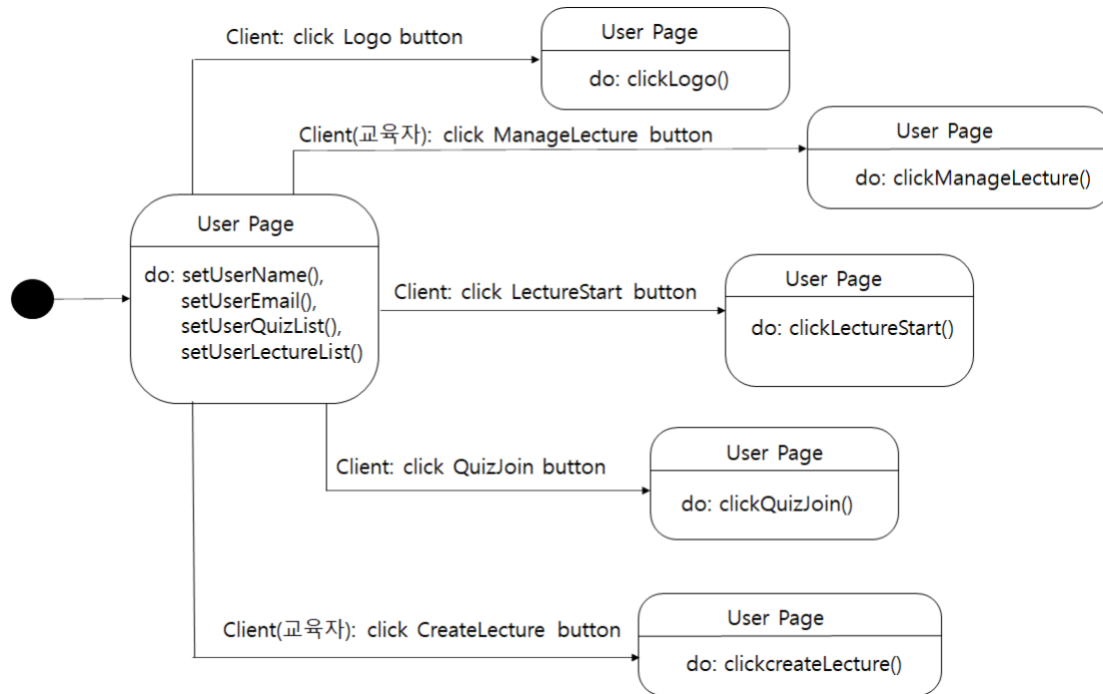
User page(학습자)의 state diagram 은 다음과 같다.

Figure 6. User Page (Learner) State Diagram



User page(교육자)의 state diagram 은 다음과 같다.

Figure 7. User Page (Educator) State Diagram



3.4 CREATE LECTURE PAGE

본 시스템의 create lecture page 이며 사용자(교육자)가 자신의 강의 콘텐츠를 제작하고 업로드할 수 있다. 강의명, 강의에 대한 간단한 소개글, 사용되는 프로그래밍 언어, 강의의 대표사진을 설정할 수 있다.

3.4.1 ATTRIBUTES

Create lecture page 를 구성하는 세부 구성요소와 그 속성은 다음과 같다.

(1) Header: Main page 의 header[3.2.1-(1)]와 동일하다. (단, search bar 는 create lecture page 에서 사용되지 않는다.)

(2) NavBar: Main page 의 NavBar [3.2.1-(2)]와 동일하다.

(3) CreateLecture: 강의에 대한 기본적인 정보로 강의를 제작할 수 있다.

- LectureName: 제작할 강의의 이름이다.
- LectureDescription: 제작할 강의에 대한 간단한 소개글이다.
- LectureLanguage: 제작할 강의에서 사용되는 프로그래밍 언어다.
- LectureImg: 제작할 강의의 대표사진이다.

(4) CheckCreateLecture: 사용자(교육자)가 강의 제작에 필요한 모든 정보를 입력했는지를 검사한 정보다.

3.4.2 METHODS

Create lecture page 를 구성하는 각 요소의 method 는 다음과 같다.

(1) Header: Main page 의 header[3.2.2-(1)]와 동일하다. (단, getSearchWord()와 clickSearch()는 user page 에서 사용되지 않는다.)

(2) NavBar: Main page 의 NavBar [3.2.2-(2)]와 동일하다. (단, clickUserPage()는 user page 에서 사용되지 않는다.)

(3) CreateLecture

- getLectureName(): 사용자(교육자)가 입력한 강의의 이름을 반환한다.
- getLectureDescription(): 사용자(교육자)가 입력한 강의에 대한 간단한 소개글을 반환한다.
- getLectureLanguage(): 사용자(교육자)가 입력한 강의에서 사용되는 프로그래밍 언어 정보를 반환한다.
- getLectureImg(): 사용자(교육자)가 설정한 강의의 대표사진을 반환한다.

- clickCreateLecture(): 사용자(교육자)가 입력한 강의의 정보를 데이터베이스 서버로 전송하여 강의를 제작한다.

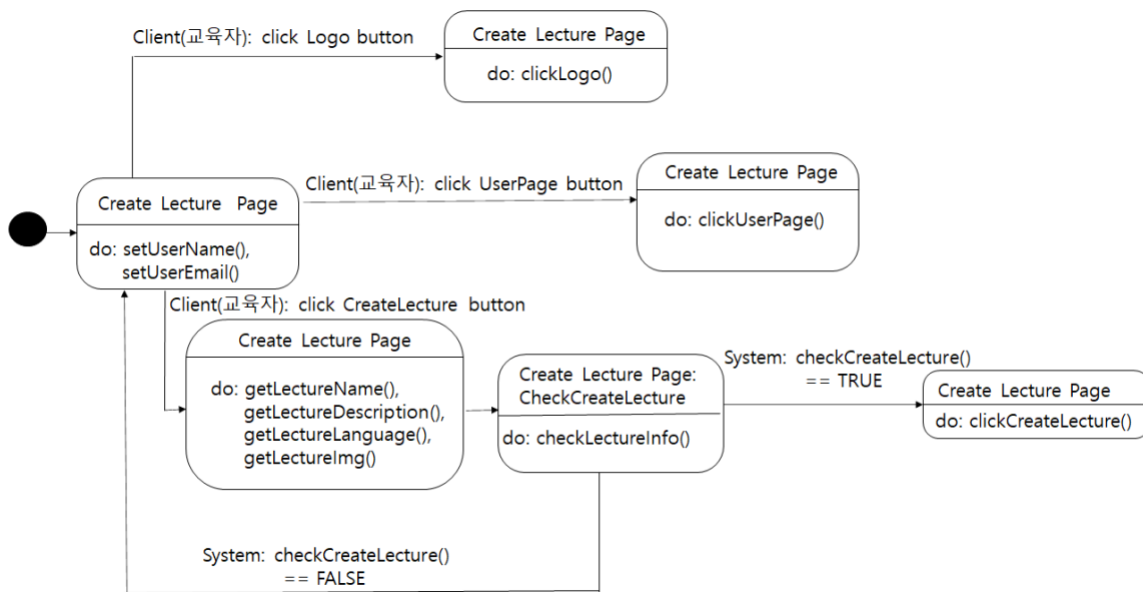
(4) CheckCreateLecture

- checkLectureInfo(): 사용자(교육자)가 강의 생성에 필요한 정보를 모두 입력했는지 검사하여 참, 거짓을 반환한다.

3.4.3 STATE DIAGRAM

Create lecture page 의 state diagram 은 다음과 같다.

Figure 8. Create Lecture Page State Diagram



3.5 MANAGE LECTURE PAGE

본 시스템의 manage lecture page 이며 사용자(교육자)가 자신이 제작한 강의에 콘텐츠를 업로드하고 강의 정보를 수정할 수 있는 기능을 포함한다. 사용자(교육자)는 강의 콘텐츠로 사진, 파일, 영상 등을 업로드 할 수 있으며 강의 차수를 구분하여

업로드할 수 있다. 각각의 강의 별로 chatGPT 의 서비스를 활용하여 강의에 대한 피드백을 받을 수 있다.

3.5.1 ATTRIBUTES

Manage lecture page 를 구성하는 세부 구성요소와 그 속성은 다음과 같다.

(1) Header: Main page 의 header[3.2.1-(1)]와 동일하다. (단, search bar 는 create lecture page 에서 사용되지 않는다.)

(2) NavBar: Main page 의 NavBar [3.2.1-(2)]와 동일하다.

(3) ManageLecture: 사용자(교육자)가 자신이 생성한 강의에 대한 기본 정보를 수정하고 강의 콘텐츠를 업로드 할 수 있다. 수정 가능한 정보로는 강의에 대한 간단한 소개글, 강의에 사용되는 언어, 강의의 대표사진이다.

- LectureName: 수정하고자하는 강의의 이름이다.
- LectureDescription: 수정하고자하는 강의에 대한 간단한 소개글이다.
- LectureLanguage: 수정하고자하는 강의에서 사용되는 프로그래밍 언어다.
- LectureImg: 수정하고자하는 강의의 대표사진이다.
- AddLectureElement: 강의의 차수를 추가하는 버튼이다.
- LectureElementTitle: 강의의 차수의 이름이다.
- LectureElementContents: 강의의 차수에 포함될 콘텐츠의 내용이다.
- DeleteLectureElement: 강의의 차수를 삭제하는 버튼이다.
- MangeLectureComplete: 강의 관리 및 수정을 완료하는 버튼이다.
- showFeedBack: chatBot 이 분석한 강의의 피드백을 출력하는 버튼이다.

(4) CheckManageLecture: 사용자(교육자)가 강의 수정에 필요한 모든 정보를 입력했는지를 검사한 정보다.

3.5.2 METHODS

(1) Header: Main page 의 header[3.2.2-(1)]와 동일하다. (단, getSearchWord()와 clickSearch()는 user page 에서 사용되지 않는다.)

(2) NavBar: Main page 의 NavBar [3.2.2-(2)]와 동일하다. (단, clickUserPage()는 user page 에서 사용되지 않는다.)

(3) ManageLecture

- setLectureName(): 사용자(교육자)가 수정하고자하는 강의의 이름을 출력한다.
- getLectureDescription(): 사용자(교육자)가 수정하고자하는 강의에 대한 간단한 소개글을 반환한다.
- setLectureDescription(): 사용자(교육자)가 새롭게 수정한 강의에 대한 간단한 소개글을 출력한다.
- getLectureLanguage(): 사용자(교육자)가 수정하고자하는 강의에서 사용되는 프로그래밍 언어 정보를 반환한다.
- setLectureLanguage(): 사용자(교육자)가 새롭게 수정한 강의에서 사용되는 프로그래밍 언어 정보를 출력한다.
- getLectureImg(): 사용자(교육자)가 설정한 수정하고자하는 대표사진을 반환한다.
- setLectureImg(): 사용자(교육자)가 설정한 새롭게 수정한 대표사진을 출력한다.
- clickAddLectureElement(): 사용자(교육자)가 강의의 새로운 차수를 추가할 수 있는 창을 열어준다.
- getLectureElementTitle(): 사용자(교육자)가 설정한 차수의 이름을 반환한다.
- setLectureElementTitle(): 사용자(교육자)가 입력한 차수의 이름을 출력한다.
- getLectureElementContents(): 사용자(교육자)가 업로드한 차수의 컨텐츠의 내용을 반환한다.
- setLectureElementContents(): 사용자(교육자)가 업로드한 차수의 컨텐츠의 내용을 출력한다.
- clickDeleteLectureElement(): 사용자(교육자)가 클릭한 강의의 차수를 삭제한다.

- clickMangeLectureComplete(): 사용자(교육자)가 수정한 강의의 정보를 데이터베이스로 전송한다.
- clickShowFeedBack(): 데이터베이스에 해당 강의에 대한 chatBot 의 feedback 정보를 요청한다. 이후 반환된 데이터를 setShowFeedBack()으로 전달한다.
- setShowFeedBack(): clickShowFeedBack()으로 전달받은 feedback 정보를 출력한다.

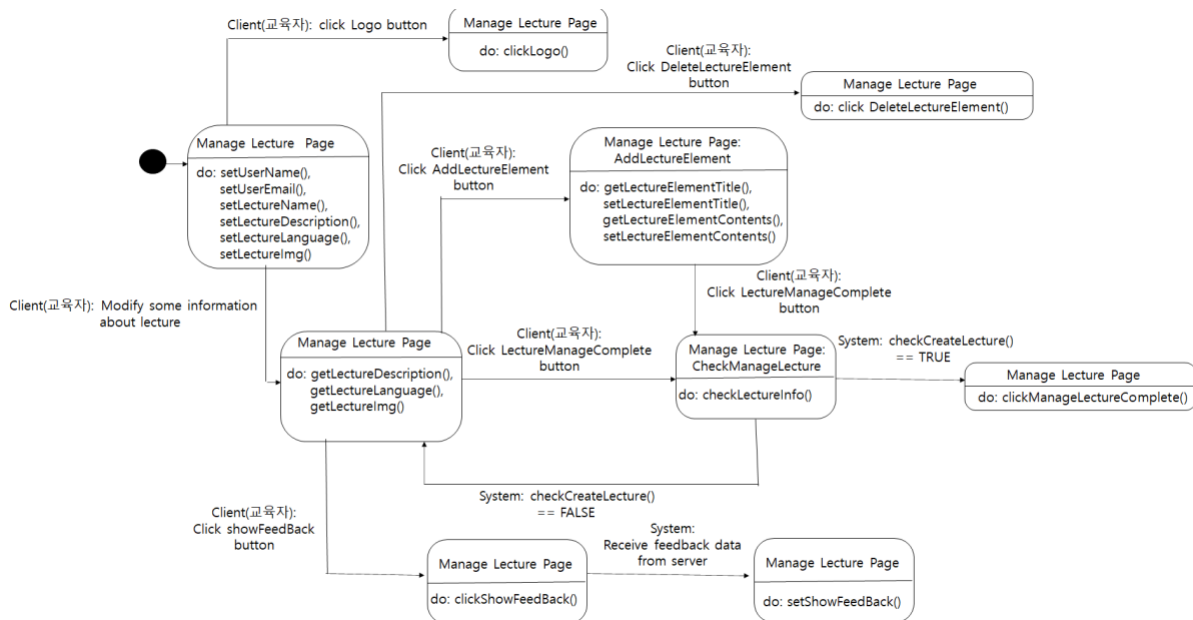
(4) CheckManageLecture

- checkLectureInfo(): 사용자(교육자)가 강의 수정에 필요한 정보를 모두 입력했는지 검사하여 참, 거짓을 반환한다.

3.5.3 STATE DIAGRAM

Mange lecture page 의 state diagram 은 다음과 같다.

Figure 9. Manage Lecture Page State Diagram



3.6 LEARNING PAGE

본 시스템의 learning page 로 사용자(학습자)가 강의를 수강할 때 접근되는 페이지다. 페이지 좌측에는 수강 중인 강의의 다른 차수로 이동할 수 있는 목차가 출력된다. 현재 수강 중인 강의의 이름과 진도율을 알 수 있다. 페이지의 중심에는 ChatBot 과 실시간으로 질의응답을 할 수 있는 채팅창과 코드를 입력하고 실행 및 제출할 수 있는 코드 에디터 창이 있다.

3.6.1 ATTRIBUTES

Learning page 를 구성하는 세부 구성요소와 그 속성은 다음과 같다.

(1) Header: Main page 의 header[3.2.1-(1)]와 동일하다. (단, search bar 는 learning page 에서 사용되지 않는다.)

(2) LectureIndex: 사용자(학습자)가 현재 수강 중인 강의의 목차 정보를 출력한다. 해당 목차를 클릭하여 해당 차수의 강의를 수강할 수 있다.

- LectureElementList: 사용자(학습자)가 현재 수강 중인 강의의 전체 차수 정보다.
- LectureElement: 사용자(학습자)가 현재 수강 중인 강의의 개별 차수 정보다.

(3) LectureHeader: 사용자(학습자)가 현재 수강 중인 강의의 간단한 정보를 출력한다. 출력하는 정보는 강의의 이름과 진도율이다.

- LectureName: 사용자(학습자)가 현재 수강 중인 강의의 이름이다.
- LectureProgress: 사용자(학습자)가 현재 수강 중인 강의의 진도율이다.

(4) ChatBot: 사용자(학습자)가 실시간으로 chatBot 과 질의응답을 할 수 있는 채팅 인터페이스 창이다.

- UserRequestMesg: 사용자(학습자)가 ChatBot 에게 전달한 메시지 정보다.

- UserMesgBox: 사용자(학습자)가 ChatBot 에게 전달한 메시지 정보가 출력될 공간이다.
- UserMesgInput: 사용자(학습자)가 ChatBot 에게 메시지를 입력할 수 있는 공간이다.
- ChatBotResponseMesg: 사용자(학습자)가 ChatBot 에게 전달한 메시지에 대한 답변 정보다.
- ChatBotMesgBox: ChatBot 의 답변 정보가 출력될 공간이다.
- UserSendMesg: 사용자(학습자)가 ChatBot 에게 메시지를 전달할 수 있는 버튼이다.

(5) CodeEditor: 사용자(학습자)가 코드를 입력하고 실행, 제출할 수 있는 코드 에디터 창이다.

- CodeEditorBox: 사용자(학습자)가 코드를 작성할 수 있는 에디터 공간이다.
- CodeSubmit: 사용자(학습자)가 코드를 제출할 수 있는 버튼이다.
- CodeRun: 사용자(학습자)가 작성한 코드에 대한 실행결과를 요청하는 버튼이다.

3.6.2 METHODS

Learning page 를 구성하는 각 요소의 method 는 다음과 같다.

(1) Header: Main page 의 header[3.2.2-(1)]와 동일하다. (단, getSearchWord()와 clickSearch()는 learning page 에서 사용되지 않는다.)

(2) LectureIndex

- setLectureElementList(): 사용자(학습자)가 현재 수강 중인 강의의 전체 차수 정보를 출력한다.
- clickLectureElement(): 사용자(학습자)가 현재 수강 중인 강의의 전체 차수 중 하나의 차수를 클릭하여 해당 차수의 강의로 이동한다.

(3) LectureHeader

- setLectureName(): 사용자(학습자)가 현재 수강 중인 강의의 이름을 출력한다.
- setLectureProgress(): 사용자(학습자)가 현재 수강 중인 강의의 진도율을 출력한다.

(4) ChatBot

- getUserMesgInput(): 사용자(학습자)가 UserMesgInput 에 작성한 메시지를 반환한다.
- clickUserSendMesg(): 사용자(학습자)가 UserMesgInput 에 작성한 메시지를 서버로 전달한다.
- setUserRequestMesg(): 사용자(학습자)가 서버로 전달한 메시지를 UserMesgBox 에 출력한다.
- receiveChatBotResponseMesg(): 서버로부터 ChatBot 의 답변을 받는다.
- setChatBotResponseMesg(): 서버로부터 받은 답변을 ChatBotMesgBox 에 출력한다.

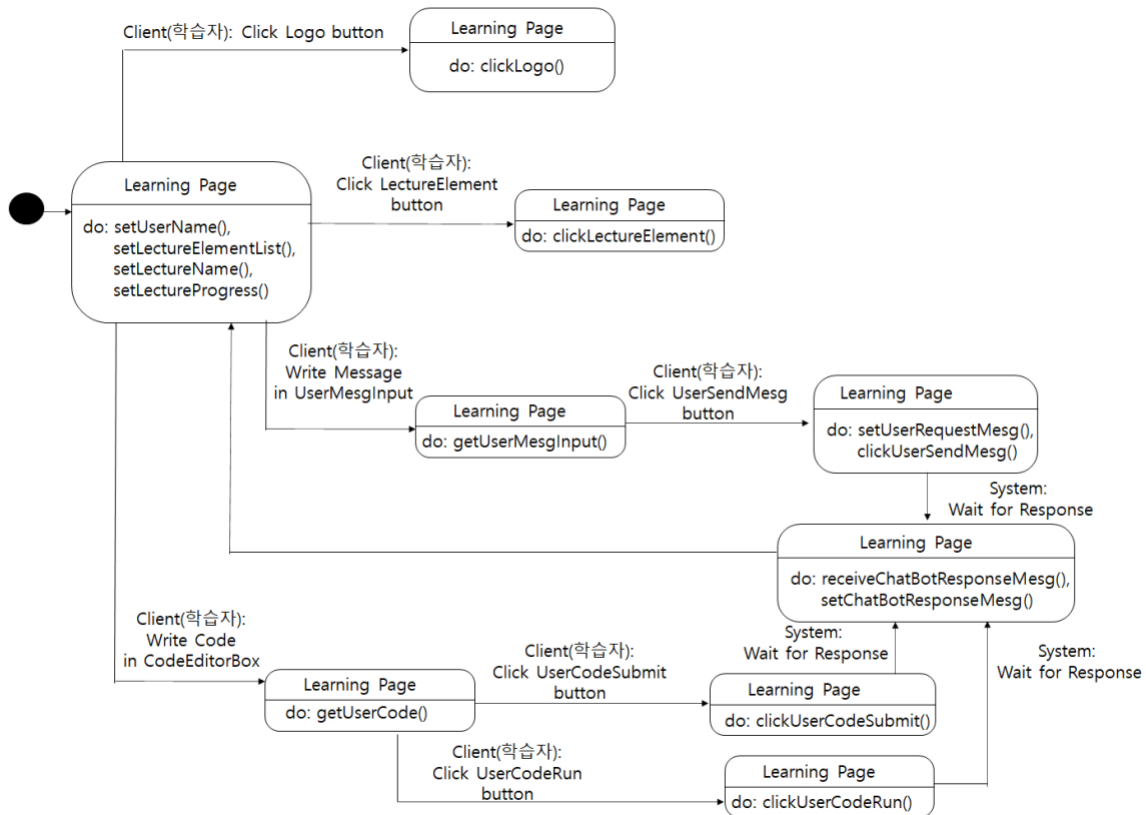
(5) CodeEditor

- getUserCode(): 사용자(학습자)가 작성한 코드를 반환한다.
- clickUserCodeSubmit(): 사용자(학습자)가 작성한 코드를 서버로 전달한다.
- clickUserCodeRun(): 사용자(학습자)가 작성한 코드를 서버로 전달한다.
- setUserCodeRunResult(): 사용자(학습자)의 코드 실행결과를 ChatBotMesgBox 에 출력한다.

3.6.3 STATE DIAGRAM

Learning page 의 state diagram 은 다음과 같다.

Figure 10. Learning Lecture Page State Diagram



3.7. QUIZ PAGE

본 시스템의 quiz page 로 사용자(학습자)가 main page 에서 퀴즈 참여 버튼을 통해 접근할 수 있는 페이지다. 페이지의 좌측 상단에는 퀴즈의 정보가 출력된다. 좌측 하단에는 ChatBot 에게 실시간으로 질의응답과 퀴즈에 대한 답 제출 및 피드백을 받을 수 있는 채팅창이 있다. 페이지의 우측에는 사용자(학습자)가 코드를 작성할 수 있는 코드 에디터 창이 있다.

3.7.1 ATTRIBUTES

Quiz page 를 구성하는 세부 구성요소와 그 속성은 다음과 같다.

(1) Header: Main page 의 Header[3.2.1-(1)]와 동일하다. (단, search bar 는 quiz page 에서 사용되지 않는다.)

(2) QuizHeader: 퀴즈의 정보가 출력된다.

- QuizInfo: 퀴즈의 정보이다.
- UserQuizStatus: 사용자에게 대한 해당 퀴즈의 정답 및 완료 여부 정보이다.

(3) ChatBot: Learning page 의 ChatBot[3.6.1-(4)]와 동일하다.

(4) CodeEditor: Learning page 의 CodeEditor[3.6.1-(5)]와 동일하다.

3.7.2 METHODS

Quiz page 를 구성하는 각 요소의 method 는 다음과 같다.

(1) Header: Main page 의 header[3.2.2-(1)]와 동일하다. (단, getSearchWord()와 clickSearch()는 quiz page 에서 사용되지 않는다.)

(2) QuizHeader

- getQuizInfo(): 데이터베이스에서 오늘의 퀴즈 정보를 가져온다.
- setQuizInfo(): getQuizInfo()에서 얻은 퀴즈 정보를 출력한다.
- setQuizStatus(): 퀴즈에 대한 사용자의 정답 및 완료 여부를 출력한다.

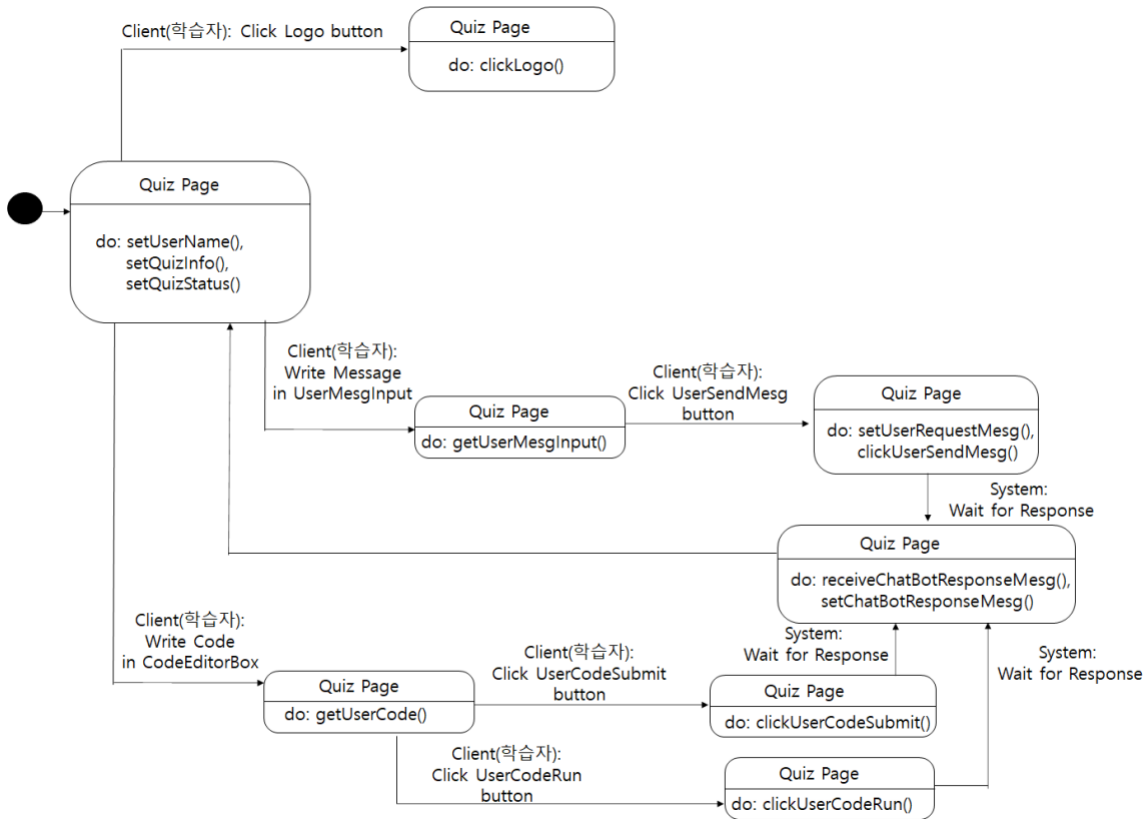
(3) ChatingAI: Learning page 의 ChatBot[3.6.2-(4)]와 동일하다.

(4) CodeEditor: Learning page 의 CodeEditor[3.6.2-(5)]와 동일하다.

3.7.3 STATE DIAGRAM

Quiz page 의 state diagram 은 다음과 같다.

Figure 11. Quiz Page State Diagram



4. SYSTEM ARCHITECTURE – BACK END

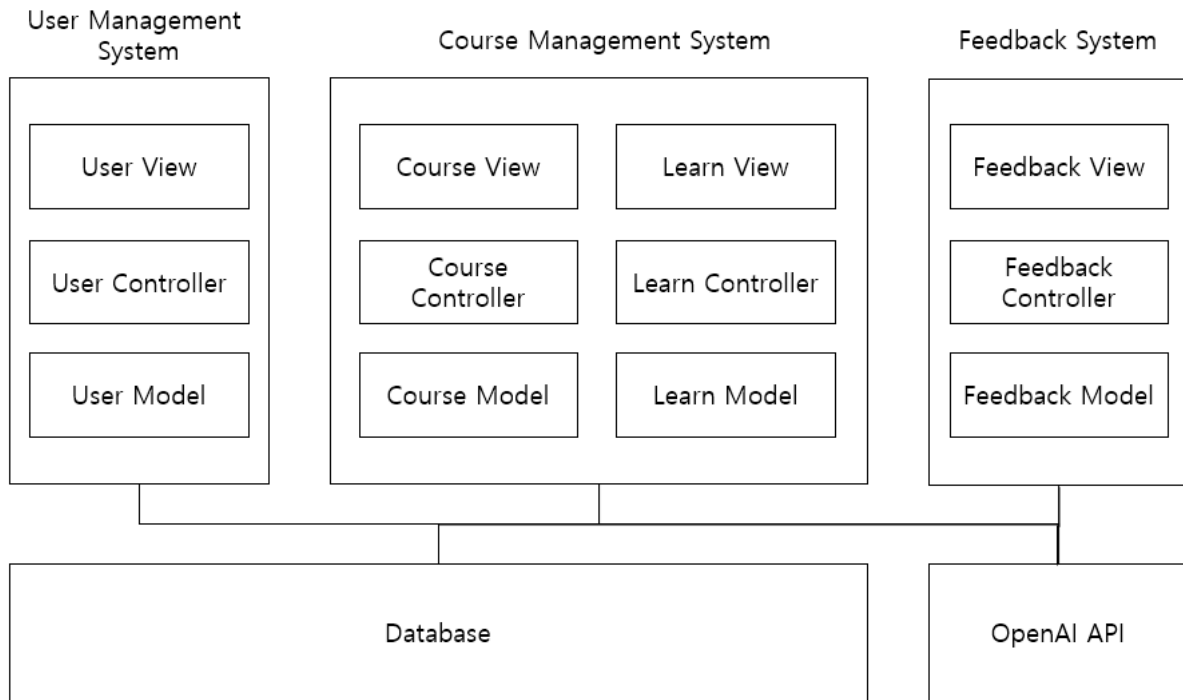
4.1 OBJECTIVES

이 장에서는 백엔드 서버의 세부 구조를 설명한다. 먼저 전반적인 구조 파악을 위해 백엔드를 이루는 하위 시스템들과 데이터베이스 간의 관계를 나타낸다. 그리고 각 하위

시스템에 대한 이해를 위해 각 컴포넌트를 구성하는 클래스와 그 동작 흐름을 나타낸다.

4.2 OVERALL ARCHITECTURE

Figure 12. Overall Architecture



Django Rest Framework 를 기반으로 구성된 Backend 는 User Management System 과 Course Management System, Feedback System 으로 나눌 수 있다. 이들은 각각 Database 와 OpenAI API 와 통신하며 동작한다. 각각의 Subsystem 은 MVC 디자인 패턴을 기반으로 구축되어 있다.

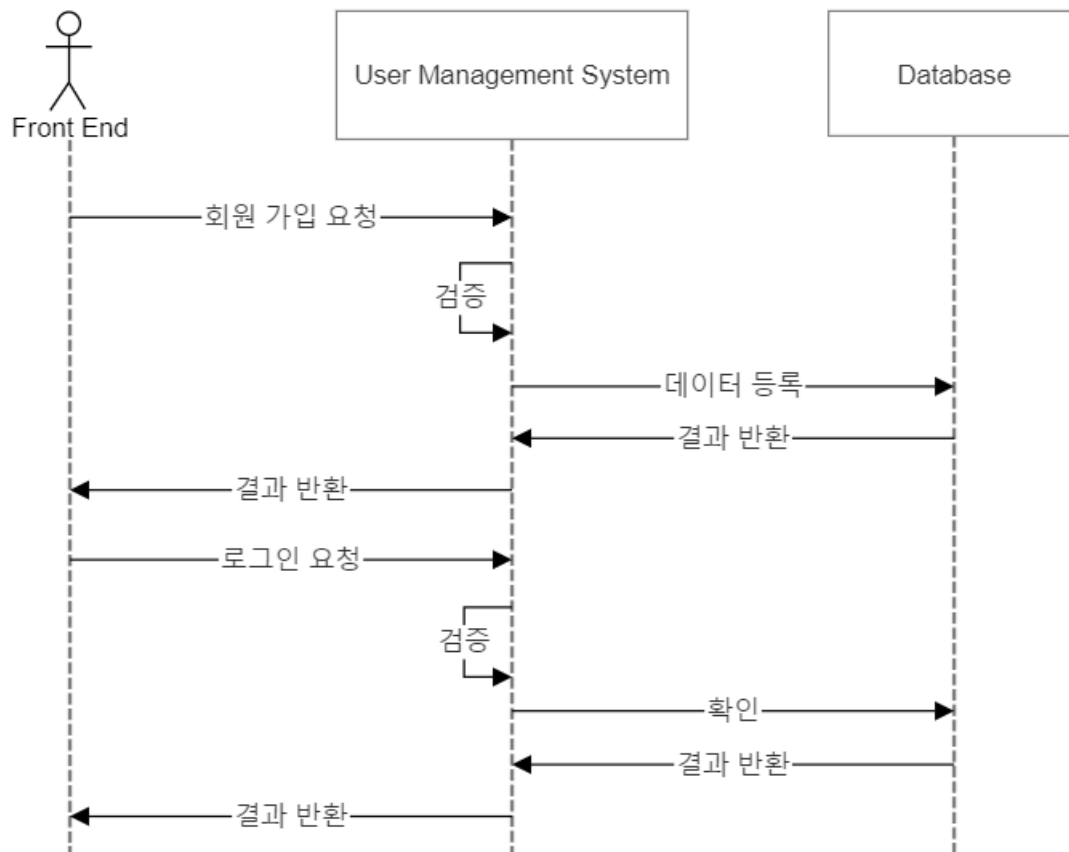
User Management System 은 사용자 회원가입, 인증 등 사용자 관리를 담당하는 시스템이다. Course Management System 은 교육자가 등록한 강의 정보를 관리하고 학습자의 학습 서비스를 제공하는 시스템이다. Feedback System 은 학습자의 강의 수강 내역을 바탕으로 퀴즈, 분석 등의 피드백 기능을 제공하는 시스템이다.

4.3 SUB SYSTEMS

4.3.1 USER MANAGEMENT SYSTEM

사용자 관리를 담당하는 User Management System 은 MVC 모델을 따라 View, Model, Controller 로 나뉜다. 구체적으로 회원 가입, 로그인, 로그아웃 등을 담당한다. Model 은 User 클래스를 통해 사용자의 정보를 관리하고, View 는 요청에 맞는 JSON 결과를 반환한다.

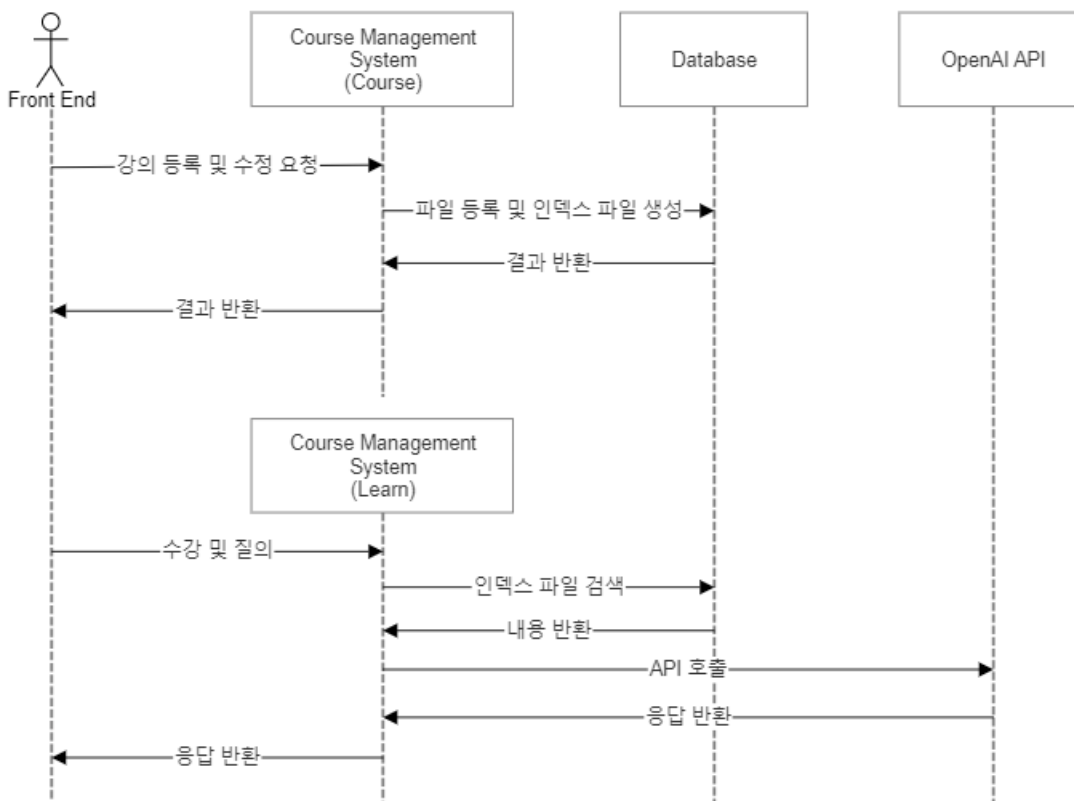
Figure 13. User Sequence Diagram



4.3.2 COURSE MANAGEMENT SYSTEM

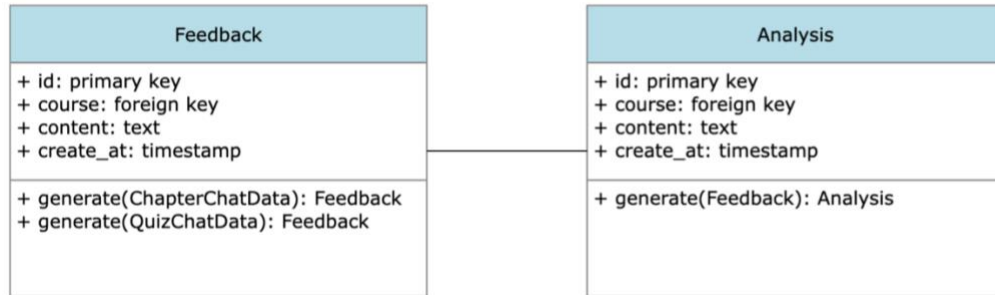
Course Management System 은 교육자가 등록한 강의 파일과 이를 기반으로 OpenAI API 를 통해 채팅 형태로 교육 서비스를 제공한다. 또한, 학습자 별 수강 이력을 저장하고 관리한다. 마찬가지로 MVC 패턴을 따르며, 크게 정적인 수강 콘텐츠 자체를 관리하는 Course 와 학습자와 상호작용하는 Learn, 2 개의 Component 로 나눌 수 있다. Course 는 교육자가 올린 교육 자료를 파일로 저장하고, Few Shot Learning 을 위한 인덱스 파일을 생성 및 관리한다. Learn 은 학습자가 강의를 수강할 때, 그 이력을 저장하고 사용자의 질의에 맞는 응답을 교육 자료 기반으로 OpenAI API 를 통해 생성하여 제공한다. 또한, Feedback System 에서 생성된 퀴즈의 수강과 이력의 관리도 담당한다.

Figure 14. Course Sequence Diagram



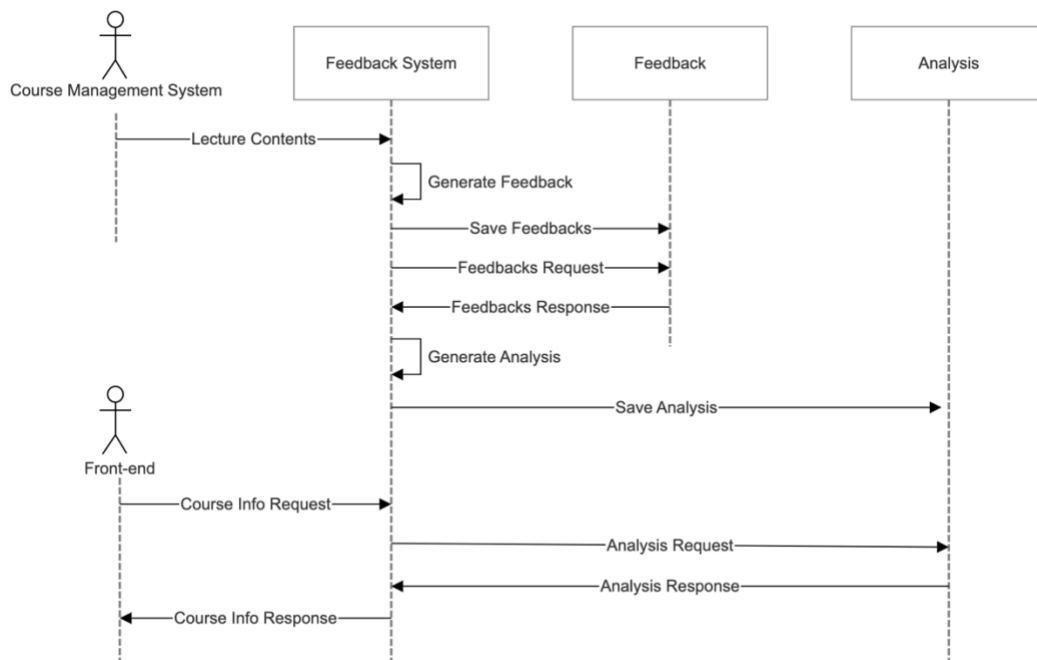
4.3.3 FEEDBACK SYSTEM

Figure 15. Feedback System Class Diagram



본 시스템에서 학습자가 강의를 수강할 때 하는 질문들과 학습 결과 등을 토대로 학습자가 어려워 하는 내용이 Feedback 객체에 저장된다. 이러한 피드백 정보를 토대로 ChatGPT 는 이를 분석한 결과를 도출해 Analysis 에 저장한다. 교육자는 이를 보고 학습의 목표와 방향 등을 설정하여 기존 교육 자료를 수정 및 새로운 교육 자료를 제작할 수 있다.

Figure 16. Feedback System Sequence Diagram



기본적으로 Course Management System 에서 각 강의에 해당하는 내용은 Feedback System 에 전달된다. 이를 바탕으로 Feedback System 에서는 ChatGPT 를 통해 해당 내용에서 학습자의 질문이나 어려워하는 내용을 추출하여 Feedback 객체에 저장한다. Feedback System 은 주기적으로 쌓인 Feedback 정보들을 Feedback 객체에 요청하여 가져온다. 이를 토대로 ChatGPT 를 통해 분석 정보를 생성하여 Analysis 객체에 저장한다. 교육자가 업로드한 강의에 대한 학습자 분석을 확인하고 싶을 때 이에 대한 요청을 프론트 엔드에서 보내면 Feedback System 은 Analysis 객체에 저장된 분석 정보를 제공한다.

5. PROTOCOL

5.1 OBJECTIVES

이 장에서는 본 시스템에서 server 와 client 의 데이터 통신에 사용되는 protocol 을 설명한다. 또한 각 interface 에 대한 요청 및 응답을 정의한다.

5.2 HTTPS

HTTPS(HyperText Transfer Protocol over Secure Socket Layer)은 웹 통신 protocol 인 HTTP internet protocol 에서 보안이 강화된 버전이다. 이를 통해 server 와 client 가 서로 요청과 응답을 주고받을 수 있다.

본 시스템에서는 기본 frontend client application 인 web 과 backend server 가 통신하기 위해 HTTPS protocol 을 사용했다.

5.3 JSON

JSON(JavaScript Object Notation)은 데이터 교환 포맷으로 속성-값의 쌍 또는 배열 등의 직렬화가 가능한 형식의 데이터 오브젝트를 전송하기 위해 사용된다. JSON 은

인간이 쉽게 읽을 수 있고 비교적으로 parsing 과 생성이 쉽고 자료의 형식에 큰 제한이 없다는 장점이 있다.

본 시스템에서는 server 와 client 가 서로 다양한 형식의 데이터를 교환하기 위해 JSON 을 사용했다.

5.4 INTERFACE

5.4.1 LECTURE LIST INQUIRY

-Request

Table 1. Lecture List Inquiry Request Protocol

Attribute	Description	
Method	GET	
URL	main.com/search	
Contents Type	text/application	
JSON	SearchWord	검색한 키워드
	Request Type	-메인 페이지 요청 -학습자 페이지 요청 -교육자 페이지 요청
	UserId	사용자 식별 번호

-Response

Table 2. Lecture List Inquiry Response Protocol

Attribute	Description
Success Code	200
Failure Code	400

JSON	LectureList	강의 식별 번호 배열
------	-------------	----------------

5.4.2 LECTURE INFO INQUIRY

-Request

Table 3. Lecture Info Inquiry Request Protocol

Attribute	Description	
Method	GET	
URL	main.com/lecture/<LectureId>	
Contents Type	multipart/form-data	
JSON	UserId	사용자 식별 번호
	LectureId	강의 식별 번호

-Response

Table 4. Lecture List Inquiry Response Protocol

Attribute	Description	
Success Code	200	
Failure Code	400	
JSON	LectureName	강의 이름
	LectureDescription	강의 설명
	LectureEducator	강의 제작자
	LectureLanguage	강의 프로그래밍 언어
	LectureImg	강의 대표 사진
	LectureProgress	강의 진도율(학습자)

5.4.3 QUIZ LIST INQUIRY

-Request

Table 5. Quiz List Inquiry Request Protocol

Attribute	Description	
Method	POST	
URL	main.com/userpage/quiz	
Contents Type	text/application	
JSON	UserId	사용자 식별 번호

-Response

Table 6. Quiz List Inquiry Response Protocol

Attribute	Description	
Success Code	200	
Failure Code	400	
JSON	UserQuizList	사용자의 퀴즈 식별 번호 목록
	QuizInfo	퀴즈 정보
	QuizStatus	사용자의 퀴즈 상태

5.4.4 QUIZ INFO INQUIRY

-Request

Table 7. Quiz Info Inquiry Request Protocol

Attribute	Description
Method	POST
URL	main.com/quiz
Contents Type	text/application

JSON	UserId	사용자 식별 번호
	QuizId	퀴즈 식별 번호

-Response

Table 8. Quiz Info Inquiry Response Protocol

Attribute	Description	
Success Code	200	
Failure Code	400	
JSON	QuizInfo	퀴즈 정보
	QuizStatus	사용자의 퀴즈 상태

5.4.5 LECTURE CREATE

-Request

Table 9. Lecture Create Request Protocol

Attribute	Description	
Method	POST	
URL	main.com/userpage/	
Contents Type	multipart/form-data	
JSON	UserId	사용자 식별 번호
	LectureName	강의 이름
	LectureDescription	강의 설명
	LectureLanguage	강의 프로그래밍 언어
	LectureImg	강의 대표 사진

-Response

Table 10. Lecture Create Response Protocol

Attribute	Description	
Success Code	200	
Failure Code	400	
JSON	CreateResult	강의 생성 성공 여부

5.4.6 LECTURE INFO MODIFY

-Request

Table 11. Lecture Info Modify Request Protocol

Attribute	Description	
Method	POST	
URL	main.com/userpage/<LectureId>	
Contents Type	multipart/form-data	
JSON	RequestType	-강의 차수 추가 -강의 차수 삭제 -강의 정보 수정
	UserId	사용자 식별 번호
	LectureName	강의 이름
	LectureDescription	강의 설명
	LectureLanguage	강의 프로그래밍 언어
	LectureImg	강의 대표 사진
	LectureElementName	강의 차수 이름
	LectureElementContents	강의 차수 콘텐츠 내용

-Response

Table 12. Lecture Info Modify Response Protocol

Attribute	Description	
Success Code	200	
Failure Code	400	
JSON	ManageResult	강의 수정 성공 여부

5.4.7 CHATBOT MESSAGE

-Request

Table 13. Chatbot Message Request Protocol

Attribute	Description	
Method	POST	
URL	main.com/lecture/chatbot	
Contents Type	text/application	
JSON	UserId	사용자 식별 번호
	UserRequestMesg	사용자 요청 메시지

-Response

Table 14. Chatbot Message Response Protocol

Attribute	Description	
Success Code	200	
Failure Code	400	
JSON	ChatBotResponseMesg	ChatBot 응답 메시지

5.4.8 CODE SEND & EXECUTE

-Request

Table 15. Code Send & Execute Request Protocol

Attribute	Description	
Method	POST	
URL	main.com/lecture/codeeditor	
Contents Type	text/application	
JSON	UserId	사용자 식별 번호
	RequestType	-코드 제출 -코드 실행
	UserCode	사용자 코드

-Response

Table 16. Code Send & Execute Response Protocol

Attribute	Description	
Success Code	200	
Failure Code	400	
JSON	ChatBotResponseMesg	ChatBot 응답 메시지

6. DATABASE DESIGN

6.1 DJANGO MODEL

Table 17. Course Model

Name	Course	
Description	강의의 정적인 정보를 저장하는 모델	
Field	Type	Description
id	PK	
title	CharField	강의의 제목

Code with Star - Design Specification

author	FK(User)	강의를 올린 사용자(교육자)
tag	FK(Tag)	강의의 주제
thumbnail	ImageField	강의 썸네일 이미지
bot_image	ImageField	채팅 AI 튜터 이미지
learner_count	PositiveIntegerField	수강자 수
created_at	DateTimeField	강의 생성 날짜
modified_at	DateTimeField	강의 수정 날짜

Table 18. User Model

Name	User	
Description	사용자의 정보를 저장하는 모델	
Field	Type	Description
id	PK	
username	CharField	사용자 아이디
password	CharField	사용자 비밀번호 (해시 값으로 저장)
email	CharField	사용자 이메일
educator	BooleanField	교육자 여부

Table 19. Tag Model

Name	Tag	
Description	강의의 주제 정보를 저장하는 모델	
Field	Type	Description
id	PK	
kind	CharField	주제 분류 ex. 언어, 프레임워크
name	CharField	주제 이름 ex. 파이썬, Django

Table 20. Chapter Model

Name	Chapter	
Description	각 강의의 목차에 해당하는 내용을 저장하는 모델	
Field	Type	Description
id	PK	
course	FK(Course)	챕터가 속한 강의

title	CharField	챕터의 제목
order	PositiveIntegerField	챕터 순서
intro	TextField	처음 진입 시 뜨는 소개말
content	FileField	교육자가 올린 강의 파일
index	FileField	Chat-GPT few shot learning 에 사용될 파일

Table 21. CourseRoom Model

Name	CourseRoom	
Description	학습자 별 강의 수강 이력을 저장하는 모델	
Field	Type	Description
id	PK	
course	FK(Course)	해당하는 강의
learner	FK(User)	해당하는 학습자
last_attempt	DateTimeField	최신 수강 날짜

Table 22. ChapterRoom Model

Name	ChapterRoom	
Description	학습자 별 챕터 수강 이력을 저장하는 모델	
Field	Type	Description
id	PK	
room	FK(CourseRoom)	해당하는 강의 방
last_attempt	DateTimeField	최신 수강 날짜

Table 23. ChapterChatData Model

Name	ChapterChatData	
Description	특정 챕터에서 AI 와 주고받은 대화 내용을 저장하는 모델	
Field	Type	Description
id	PK	
room	FK(ChapterRoom)	해당하는 챕터 방
data	TextField	채팅 내용
bot	BooleanField	AI 답변인지 여부
timestamp	DateTimeField	채팅 보낸 시간

Table 24. Feedback Model

Name	Feedback	
Description	강의 마다 학습자가 질문하거나 어려워하는 내용을 Chat-GPT 로 분석해서 따로 저장하는 모델	
Field	Type	Description
id	PK	
course	FK(Course)	해당하는 강의
content	TextField	분석 내용
created_at	DateTimeField	생성 날짜

Table 25. Quiz Model

Name	Quiz	
Description	Feedback 모델의 정보를 바탕으로 Chat-GPT 가 생성한 퀴즈	
Field	Type	Description
id	PK	
course	FK(Course)	해당하는 강의
problem	TextField	문제 내용
answer	TextField	정답
created_at	DateTimeField	생성 날짜

Table 26. QuizRoom Model

Name	QuizRoom	
Description	사용자 별 퀴즈 풀이 이력을 저장하는 모델	
Field	Type	Description
id	PK	
quiz	FK(Quiz)	해당하는 퀴즈
learner	FK(User)	해당하는 학습자
last_attempt	DateTimeField	최신 풀이 날짜

Table 27. QuizChatData Model

Name	QuizChatData
------	--------------

Description	사용자 별 퀴즈 풀이 대화 내용을 저장하는 모델	
Field	Type	Description
id	PK	
room	FK(QuizRoom)	해당하는 퀴즈 방
data	TextField	채팅 내용
bot	BooleanField	AI 답변인지 여부
timestamp	DateTimeField	채팅 보낸 시간

Table 28. Analysis Model

Name	Analysis	
Description	Feedback 모델에 저장된 정보를 바탕으로 Chat-GPT 가 작성한 강의에 대한 피드백	
Field	Type	Description
id	PK	
course	FK(Course)	해당하는 강의
content	TextField	분석 내용
created_at	DateTimeField	생성 날짜

6.2 RELATIONAL SCHEMA

Figure 17. Relational Schema



7. TESTING PLAN

7.1 OBJECTIVE

이 장에서는 요구사항 명세서에 기술된 시스템의 use case 들을 바탕으로 시스템이 올바르게 동작하는지를 확인한다. Testing 은 Development testing, Release testing, User testing 의 세 가지 관점에서 수행하며 이를 통하여 시스템의 잠재적인 오류 및 결함 사항들을 조기에 발견하여 시스템 배포 시 완성도와 안전성을 극대화하는 것을 목표로 한다.

7.2 TESTING POLICY

7.2.1 DEVELOPMENT TESTING

Development testing 은 시스템 개발 도중 발생할 수 있는 여러 위협 요소들과 비용을 줄이는 것을 목표로 한다. 이를 위해 시스템을 구성하는 여러 하위 시스템 및 컴포넌트들에 대하여 unit test 를 독립적으로 진행하고 이후 순차적인 통합을 통해 전체적인 시스템에 대한 테스트를 수행한다. Unit test 는 하위 시스템 또는 컴포넌트 단위에서 기능이 독립적으로 예상대로 작동하는 지를 확인하며 전체 시스템에 대한 테스트를 진행할 때에는 시스템의 성능, 보안, 안전성을 검증한다. 구체적으로는 정적 코드 분석, 데이터 흐름 분석, 피어 코드 리뷰 등의 방법을 사용한다.

7.2.2 RELEASE TESTING

Release testing 은 시스템의 새로운 버전이 출시될 때 그 완전성을 보장하고 배포 과정이 정상적으로 이루어질 수 있도록 하는 것을 목표로 한다. 이를 위해 기능 위주의 점검인 알파 테스트와 사용자의 피드백과 발견한 결함을 반영하는 베타 테스트를 수행하며 시스템이 예상대로 작동하고 사용자들의 요구사항을 만족시키는지 확인한다.

7.2.3 USER TESTING

User testing 은 시스템을 실제로 사용자가 사용하는 과정을 통해 명시된 요구사항들을 만족하는지를 점검하는 것을 목표로 한다. 사용자들은 이 테스트를 통해 시스템의 use case 들을 평가하고 이를 바탕으로 시스템이 배포될 수 있는지 여부를 판단한다.

7.3 TEST CASE

7.3.1 LOGIN & SIGN UP

Table 29. User Login

Element	Description
Test Name	User Login
Pre-Conditions	로그인 화면에서 사용자가 로그인 아이디 및 비밀번호를 제출하였거나 회원가입 옵션을 통해 새로운 로그인 아이디 및 비밀번호를 등록하였다.
Test Steps	<ol style="list-style-type: none"> 1. 사용자가 string data type 의 아이디 및 비밀번호를 제출하거나 회원가입에서 등록한다. 2. 로그인의 경우 서버는 아이디가 데이터베이스에 존재하는지 여부와 존재한다면 비밀번호가 해당 아이디와 등록된 비밀번호에 부합하는지를 확인한다. 회원가입의 경우 서버는 아이디와 비밀번호를 데이터베이스에 신규로 저장한다. 이 때 아이디는 기존에 존재하는 아이디와 다름을 먼저 확인하여야 한다. 3. 로그인의 경우 아이디가 존재하고 비밀번호가 올바름이 확인되었다면 메인 페이지로 넘어간다. 회원가입의 경우 아이디가 기존에 존재하지 않는다면 성공적으로 등록되었음을 알리고 존재하는 경우에는 아이디를 다시 입력할 것을 알린다.
Expected Results	올바른 사용자 아이디와 비밀번호 제출 시에만 메인 페이지로 넘어가며 신규로 가입된 회원 아이디가 기존과 다른 것을 보장한다.

7.3.2 LECTURE SEARCH

Table 30. Lecture Search Test

Element	Description
Test Name	Lecture Search
Pre-Conditions	사전에 강의 콘텐츠들이 준비되어 있으며 해당 강의들의 내용에 따라 키워드를 통해 그 종류가 분류되어 있다.
Test Steps	<ol style="list-style-type: none"> 1. 사용자가 강의 검색창에 키워드를 입력하고 검색 필터링 옵션을 선택하여 검색 버튼을 클릭한다. 2. 서버는 스토리지에 존재하는 강의 콘텐츠 중 해당 입력 키워드 및 검색 필터링 옵션에 부합하는 강의 목록을 찾아 그들의 강의 제목을 페이지에 렌더링하는 방식으로 사용자에게 반환한다.
Expected Results	사용자가 검색한 키워드 및 필터링 옵션에 알맞은 강의 콘텐츠 제목들이 디스플레이 된다.

7.3.3 VIEW LECTURE REGISTERED

Table 31. View Lecture List

Element	Description
Test Name	View Lecture List
Pre-Conditions	사용자가 수강하는 강의 목록에 대한 정보가 해당 사용자의 계정 정보와 함께 스토리지에 저장되어 있다.
Test Steps	<ol style="list-style-type: none"> 1. 사용자가 자신이 수강하는 강의 목록을 조회한다. 2. 서버는 데이터베이스에 저장된 사용자의 계정 정보와 이와 함께 저장되어 있는 연관된 강의 목록을 반환하여 페이지에 렌더링한다.
Expected Results	사용자가 수강 중인 강의 목록이 올바르게 디스플레이 된다.

7.3.4 LECTURE REGISTER

Table 32. Lecture Register Test

Element	Description
Test Name	Lecture Register
Pre-Conditions	사전에 강의 콘텐츠들이 스토리지에 업로드 되어 있다.
Test Steps	<ol style="list-style-type: none"> 1. 사용자가 강의를 선택하여 수강 신청한다. 2. 서버는 사용자의 계정 정보와 함께 수강 신청된 강의를 저장한다. 3. 서버는 수강하고자 하는 강의의 학습 페이지를 디스플레이 한다. 4. 사용자가 수강 강의 목록을 조회하면 추가로 신청된 해당 강의가 함께 결과로 나와야 한다.
Expected Results	사용자가 성공적으로 강의를 수강 신청 할 수 있다.

7.3.5 QUIZ PARTICIPATION

Table 33. Quiz Participation Test

Element	Description
Test Name	Quiz Participation
Pre-Conditions	사전에 강의 콘텐츠 자료가 스토리지에 저장되어 있다.
Test Steps	<ol style="list-style-type: none"> 1. 메인 페이지에 상단에 퀴즈 배너에는 24 시간을 주기로 바뀌는 퀴즈가 디스플레이 된다. (해당 퀴즈는 강의 콘텐츠를 기반으로 생성한다.) 2. 사용자가 퀴즈 배너를 클릭한다. 3. 서버는 퀴즈를 풀 수 있는 페이지를 디스플레이 한다. 4. 사용자가 퀴즈에 대한 답변을 제출한다. 5. OpenAI API 를 통하여 퀴즈에 대한 사용자의 답변을 평가하고, 해당 피드백 결과를 사용자에게 반환한다.

Expected Results	사용자는 성공적으로 24 시간 주기로 바뀌는 퀴즈에 참여하고 답변에 대한 피드백을 즉각적으로 받을 수 있다.
------------------	--

7.3.6 CHATBOT USAGE

Table 34. Chatbot Usage Test

Element	Description
Test Name	ChatBot Usage
Pre-Conditions	OpenAI API 토큰을 관리자가 보유하여 이를 통해 ChatBot 서비스를 제공할 수 있으며 강의 콘텐츠가 스토리지에 보관되어 있고 사용자의 질문 목록 역시 데이터베이스에 저장되어 있어 이를 ChatBot 이 참고하여 답변을 제공할 수 있다.
Test Steps	<ol style="list-style-type: none"> 1. 사용자가 강의 콘텐츠와 관련된 질문 또는 특정 코드를 ChatBot 에게 제출한다. 2. ChatBot 은 스토리지에서 강의 콘텐츠 정보를, 그리고 데이터베이스에서 과거 사용자의 질문 목록을 참고한 최종 질문 prompt 를 전달받아 해당 답변 및 피드백을 반환한다. 3. 서버는 사용자의 질문 목록을 사용자의 계정 정보와 함께 데이터베이스에 저장한다.
Expected Results	사용자는 강의 콘텐츠에 대한 질문을 ChatBot 에게 하고 그에 대한 답변을 받을 수 있으며 코드 피드백 역시 받을 수 있다.

7.3.7 LECTURE MANAGEMENT

Table 35. Lecture Management Test

Element	Description
Test Name	Lecture Management
Pre-Conditions	사용자의 구분은 강의를자이다.

Test Steps	1. 강의자가 새로운 강의 콘텐츠를 업로드 페이지에서 강의 제목과 함께 파일의 형태로 업로드하거나 기존 강의 콘텐츠를 수정 페이지에서 변경한다. 2. 서버는 등록된 새로운 강의 콘텐츠를 스토리지에 저장하거나 변경된 기존 강의 콘텐츠의 내용을 스토리지에 반영한다.
Expected Results	강의자는 성공적으로 새로운 강의 콘텐츠를 업로드하거나 기존 강의 콘텐츠의 내용을 수정 및 변경할 수 있다.

8. DEVELOPMENT PLAN

8.1 OBJECTIVE

이 장에서는 프로젝트의 개발 진행 단계에 대한 계획을 기술하고 개발 타임라인을 제시하며 현재 완성된 단계들을 명시한다.

8.2 DEVELOPMENT TIMELINE

Figure 18. Development Timeline

