

Software Requirements Specification

LLM-based Coding Education Project

by

이정혁, 김남현, 김태영, 이상현, 한준호

TEAM 6

Instructor

이은석

Teaching Assistant

김영경, 김진영, 최동욱, 허진석

Document Date

08 November, 2024

Faculty

SungKyunKwan University

Contents

- 1. Introduction 7
 - 1.1. Purpose..... 7
 - 1.2. Scope 7
 - 1.3. Definitions, Acronyms, and Abbreviation 7
 - 1.4. References..... 8
 - 1.5. Overview..... 8
- 2. Overall Description 9
 - 2.1.1. System Interfaces..... 9
 - 2.1.2. Hardware Interfaces..... 9
 - 2.1.3. User Interfaces 9
 - 2.1.4. Software Interfaces 10
 - 2.1.5 Memory Constraints 10
 - 2.1.6 Operations - System manager 10
 - 2.1.7 Hardware Interfaces..... 10
 - 2.2 Product Functions 11
 - 2.2.1 로그인 11
 - 2.2.2. 코딩 실습 시 LLM 서비스 제공 11
 - 2.2.3. 코딩 실습 시 강의 제공..... 11
 - 2.2.4. 정답 확인 11
 - 2.2.5. 커리큘럼 생성 11
 - 2.2.6. 포트폴리오 생성 11
 - 2.3 User Characteristic 11
 - 2.3.1. System Administrator 11
 - 2.3.2. User 12
 - 2.4 Constraints..... 12

2.5. Assumptions and Dependencies	12
3. Specific Requirements.....	13
3.1. External Interface Requirements	13
3.1.1. User Interface	13
3.1.2 Hardware Interface.....	17
3.1.3 Software Interface	17
3.1.4 Communication Interface.....	18
3.2 Functional Requirements	19
3.2.1 Use Case	19
3.2.2 Use Case Diagram	23
3.2.3 Data Dictionary	23
3.2.4 Data Flow Diagram	26
3.3 Performance Requirements	26
3.3.1 Static Numerical Requirements.....	26
3.3.2 Dynamic Numerical Requirements.....	26
3.4. Logical database requirements.....	27
3.5. Design constraints.....	27
3.5.1. Physical design constraints	27
3.5.2. Standards compliance	27
3.6 Software System Characteristics.....	28
3.6.1. Reliability	28
3.6.2. Availability	28
3.6.3 Security.....	28
3.6.4 Maintainability.....	28
3.6.5 Portability	28
3.7. Organizing the specific requirements.....	29
3.7.1. Context Model	29

3.7.2. Process Model.....	29
3.8 System Architecture	30
3.9 System Evolution	30

List of Figures

Figure 1: Use Case Diagram	23
Figure 2: Data Flow Diagram.....	26
Figure 3: Context Model.....	29
Figure 4: Process Model	29
Figure 5: System Architecture	30

List of Tables

Table 1: Table of acronyms and abbreviations	7
Table 2: Table of terms and definitions	7
Table 3: User Interface of Input Processing	13
Table 4: User Interface of Register	14
Table 5: User Interface of Main page.....	15
Table 6: User Interface of Learning page	16
Table 7: Hardware Interface.....	17
Table 8: Software Interface.....	17
Table 9: Communication Interface.....	18
Table 10: Use Case of Register.....	19
Table 11: Use Case of Log-in / out.....	20
Table 12: Use Case of Keyword Selection	21
Table 13: Use Case of Curriculum Learning	21
Table 14: Use Case of LLM Q&A Interaction	22
Table 15: Data Dictionary of Account	23
Table 16: Data Dictionary of Curriculum.....	24

Table 17: Data Dictionary of Module 24

Table 18: Data Dictionary of Curriculum Progress 24

Table 19: Data Dictionary of Training Log 25

Table 20: Data Dictionary of User Code 25

Introduction

1.1. Purpose

본 문서는 LLM을 이용해 코딩 학습에 도움을 주는 프로젝트 CodeSunbi의 다양한 측면의 requirements에 관련된 정보와, CodeSunbi의 간단한 system 구조를 담아, 추후에 유지 보수하는 과정에서 참고할 수 있도록 제작되었다.

1.2. Scope

본 프로젝트는 코딩 교육에 LLM을 접합해 사용자에게 서비스를 제공하는 웹 코딩 플랫폼이다. LLM을 이용해 사용자 맞춤의 코딩 교육 및 코딩 진행에 적합한 피드백을 받을 수 있다.

1.3. Definitions, Acronyms, and Abbreviation

Acronyms & Abbreviations	Explanation
LLM	Lange Language Model

Table 1: Table of acronyms and abbreviations

Terms	Definitions
Langchain	자연어 처리와 인공지능 모델을 연결해 사용자와 대화형 AI 애플리케이션을 만들기 위한 프레임 워크
Machine Learning	데이터를 통해 패턴을 학습하여 미래 예측이나 분류 작업을 수행하는 알고리즘 기반의 기술
Deep Learning	인공신경망을 활용하여 복잡하 패턴을 학습하는 머신러닝의 하위 분야로, 주로 대량의 데이터와 강력한 연산이 필요한 문제에 사용됨
Coding	컴퓨터가 이해할 수 있는 명령어를 작성하여 원하는 기능을 구현하고 소프트웨어를 개발하는 과정

Table 2: Table of terms and definitions

1.4. References

Langchain: <https://python.langchain.com/docs/introduction/>

IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications,
In IEEEExplore Digital Library <http://ieeexplore.ieee.org/Xplore/guesthome.jsp>

1.5. Overview

본 요구사항 명세서는 총 4개의 챕터로 나누어져 있다. 첫 번째 챕터는 프로젝트 CodeSunbi의 소개와 그 목적을 설명하고 프로젝트와 관련된 여러 용어들에 대한 설명을 제공하고 있다. 두 번째 챕터는 프로젝트의 전반적인 Description을 제공한다. 여기에는 제품의 개발 동기 및 목적, 소프트웨어의 주요 기능 등을 포함하고 있다. 세 번째 챕터는 프로젝트의 requirements에 관한 부분을 정리하였다. 여기에는 software의 여러 특징에 관한 정보와 보안, 가용성 등에 관한 정보도 포함시켰다. 마지막 챕터는 본 명세서의 references를 포함하였다. 팀의 모든 멤버는 본 문서 작성 및 system을 디자인하는데 기여하였다

2

Overall Description

2.1. Product Perspective

자연어 처리를 활용하여 코딩 beginner들에게 흥미를 높여주는 CodeSunbi는 기존 코딩 및 프로그래밍에 대한 사전 지식이 없거나, 혹은 적은 사람들이 프로그래밍을 공부할 때 사용자에게 적합한 프로젝트를 생성하여 흥미를 잃지 않고 공부할 수 있게 도와주도록 설계되었다. 키워드 선택을 통해 맞춤형 프로젝트를 생성함으로써 전체적인 개발 과정에서 재미와 ownership을 느끼도록 하였고, 이 과정에서 단계별 학습을 통해 관련 지식들을 순차적으로 습득할 수 있게 하였다. 각 과정에서 핵심 내용과 관련된 추가 강의 수강 및 자연어 처리를 활용한 Chat GPT와 같은 기능을 도입하여 사용자들의 자유로운 질문에 대한 유연한 답변 또한 얻을 수 있다. 이후 완성된 프로젝트를 각 개인이 가시화된 결과로 볼 수 있으며 이는 추후 포트폴리오 제작 등에 유용하게 쓰일 것이다.

2.1.1. System Interfaces

사용자들은 키워드 선택을 통해 자신의 흥미 및 관심도가 반영된 개인 프로젝트를 생성하게 된다. 이후 순차적으로 나타나는 단계에 의해 프로젝트를 완성하게 되며, 이 과정에서 강의 및 자연어 처리를 이용한 AI와 질의응답을 할 수 있다. 이는 각 사용자들의 정보가 캐시 메모리에 저장되어 이전 대답과의 관계를 유추하여 시스템이 좀 더 유동적이고 정확하게 답변을 할 수 있게 할 것이다. 각 사용자 별 진도와 강의 수강 과정은 데이터베이스에 저장되어 이후 학습을 이어 할 때 유용하게 사용된다.

2.1.2. Hardware Interfaces

CodeSunbi는 웹 기반으로 작동한다. 프로젝트 생성 중에는 게임 개발, 앱 개발, 웹 개발 등이 포함되기 때문에 권장되는 사양은 언리얼 엔진이 권장하는 사양인 윈도우 기준 Windows 10 64비트, Intel 또는 AMD 쿼드코어, 2.5GHz 이상, 8GB RAM, DirectX 11 또는 12 호환 그래픽카드이다.

2.1.3. User Interfaces

사용자들은 CodeSunbi에 첫 가입을 하게 되면, 프로그래밍과 관련된 여러 키워드 중 하나씩 선택하는 과정을 가진다. 키워드 선택의 각 단계는 이전 선택 단계의 결과를 반영하여 유동적으로 나타나게 되며, 이를 통해 사용자는 조금 더 자신의 취향과 맞는 결과가 반영되는 것을 볼 수 있다. 프로젝트가 생성이 되고 학습을 시작하게 되면, 순차적 step이 생성되고 좌측에는 목차 및 step의 순서, 중앙부에는 실제 코딩을 하는 space가, 우측에는 AI에 자연어를 입력할 수 있는, 질의응답 및 코딩에 도움을 주는 도우미 창이 있다. 강의의 경우에는 작은 팝업 창 형태로 삽입

되어 클릭 시 수강할 수 있게 한다. 이는 이해가 되지 않는 내용을 시각적, 청각적으로 이해하는데 있어 큰 도움이 될 것이다. 최종적으로 완성된 프로젝트를 본인이 확인하고, 이후 수정 및 개선을 하는 과정에서 애정과 흥미를 붙일 수 있고 이는 훌륭한 개발자로 성장하는 것에 있어서 큰 도움이 될 것이다.

2.1.4. Software Interfaces

엔리얼 엔진이 가동되기 위한 요구 사양인 64비트 윈도우 10이상이 권장되며, 인터넷을 통해 본 system에 접근할 수 있기 때문에 Microsoft edge 버전 79이상을 사용해야 하며, 일반적으로는 95이상의 가장 최신 버전의 사용을 권장한다. Chrome의 경우 100.0이상이 권장된다.

2.1.5 Memory Constraints

엔리얼 엔진의 조건 때문에 게임 개발을 희망하는 사용자의 경우 8GB RAM이상이 요구된다. Backend 및 Docker과 같은 가상 환경을 이용하는 개발을 희망하는 사용자의 경우에는 16GB RAM이상이 권장된다.

2.1.6 Operations - System manager

채 접속률을 통해 시스템에 대한 사용자의 만족도를 알 수 있다. 또한, 프로젝트 진행 중 메모리에 저장된 AI와의 대화 내용의 빈도 및 내용을 분석하여 AI가 주어진 자연어에 대해 완벽히 이해하고 알맞은 대답을 하는 지 유추할 수 있고, 이를 통해 AI의 기능을 개선할 수 있다. 마지막으로, 만들어진 프로젝트에 대한 수정 및 개선 통계를 확인하여 본 시스템을 통해 개발한 프로젝트에 대한 사용자의 만족도를 확인할 수 있다.

2.1.7 Hardware Interfaces

- 로그인 : 사용자들은 자신의 ID, Password로 로그인 할 수 있다.
- 회원가입 : 자신의 고유 ID, Password로 회원가입 할 수 있다.
- 키워드 선택 : 첫 회원가입 후, 사용자들은 키워드 선택을 통해 자신에게 알맞은 프로젝트를 생성하게 된다. 이는 각 개인 데이터베이스에 저장된다.
- 프로젝트 제작 및 학습 : 저장된 강의를 팝업창을 이용하여 수강할 수 있다. 또한 이를 기반으로 직접 단계별 코딩을 통해 프로젝트를 제작할 수 있다. 우측 AI를 활용하여 학습 중 질문에 대한 답 혹은 개발 과정 중 도움을 쉽게 받을 수 있다.
- Product perspective
- System interfaces : User이 자연어 처리를 이용하는 AI를 이용하여 본인에게 맞는 커리큘럼 생성 및 학습을 통해 프로젝트 완성, AI에 입력한 input 및 output, 학습자별 진도율과 코딩 내용은 데이터베이스에 저장되어 이어지는 학습을 용이하게 할 수 있음.
- User interfaces : User가 웹을 통해 본 system을 이용, 실력 향상, 흥미 추구 등 원하는 목표에 도달하기까지 사용. 프로젝트 완성을 recommended.
- Hardware interfaces : Intel 또는 AMD 쿼드코어, 2.5GHz 이상, 8GB RAM, DirectX 11 또는 12 호환 그래픽카드
- Software interfaces : 64bit Windows 10이상, 크롬 100.0 및 Microsoft Edge 95이상의 최신 버전
- Communications interfaces : memory interfaces, software interfaces, hardware interfaces.
- Memory constraints : 8GB(게임), 16GB(Docker)
- Operation : system manager, users

2.2 Product Functions

2.2.1 로그인

현재 코딩 교육 서비스를 받는 user가 누구인지를 확인하는 작업이다. User가 이전에 진행했던 교육의 커리큘럼과 교육의 순서를 읽어 다음에 진행해야 될 교육이 어떤 것인지 확인한다. 로그인 과정은 사용자가 아이디와 비밀번호를 입력하면 시스템은 그것을 Database과 비교해 일치하는 계정을 찾고 계정의 정보를 읽어 들인다. 만약 계정 정보가 없다면 사용자에게 알람 메시지를 보낸다.

2.2.2. 코딩 실습 시 LLM 서비스 제공

코딩 교육을 진행할 때 사용자가 LLM 서비스를 이용해 도움을 받고자 한다. 이때 LLM 모델은 사용자가 현재까지 작성한 코드를 읽고, 정답인 코드와 비교한 후에 사용자가 다음으로 작성해야 될 코드의 일부분을 알려준다. 사용자는 이것을 읽고, 본인이 현재 코딩을 진행 할 때 막혀 있는 부분을 파악하고, 다음 스텝에 대한 힌트를 얻어서 사용자 스스로 개발에 대한 생각을 할 수 있게 유도한다.

2.2.3. 코딩 실습 시 강의 제공

코딩 교육을 진행할 때에 사용자는 현재 본인이 진행하고 있는 부분에 대한 해설 자료나 강의 자료를 요구할 수 있다. 사용자가 강의 자료를 요구할 시 LLM 모델은 현재 사용자가 진행하고 있는 코딩의 주제와 사용자가 현재까지 작성한 코드를 분석해서 현재 가장 필요한 개념이 무엇인지를 파악해 해당 주제와 관련된 강의 자료를 제공한다.

2.2.4. 정답 확인

사용자가 제공된 코딩 문제를 다 풀었을 때 작성된 코드가 맞는지 확인해야 된다. 만약 사용자의 코드가 정확하다면 시스템은 코드가 정답인 것을 사용자에게 알려주고, 만약 정답이 아니라면 그 사실을 사용자에게 알려준다. 만약 사용자가 틀린 코드의 이유를 묻는다면 LLM 모델은 정답 코드의 힌트를 알려주어 사용자가 스스로 틀린 부분을 파악할 수 있도록 유도한다.

2.2.5. 커리큘럼 생성

사용자가 본인이 진행하고 싶은 코딩 분야를 설정한 후, 이것을 시스템이 읽어서 사용자에게 맞는 코딩 교육 커리큘럼을 생성한다. 그 후 이 커리큘럼에 맞추어서 사용자의 코딩교육을 진행한다.

2.2.6. 포트폴리오 생성

사용자가 현재까지 진행한 코딩 교육을 분석해서 이것을 하나의 포트폴리오로 만들어 사용자에게 제공한다. 사용자는 포트폴리오를 읽고 본인의 교육 진행 상황과 방향을 파악할 수 있다.

2.3 User Characteristic

2.3.1. System Administrator

웹 기반 서비스이기 때문에, 이에 대한 기본적인 지식을 가지고 있어야 한다. 또한, 웹 인터페이스 뿐만 아니라 서버와의 연동을 통해 User별 데이터 저장을 관리하고 분석해야 하기 때문에 웹

과 backend에 능통한 사람이 본 서비스를 전반적으로 관리해야 한다.

User에게 도움을 주고 프로젝트를 완성하는 데에 큰 도움이 될 AI와 관련된 사항의 경우 자연어 처리를 Python 혹은 Java 등으로 경험해 본 적이 있는 개발자가 관련 문제를 담당해야 한다. 담당자는 AI모델의 답변 개선 뿐만 아니라 관련 키워드를 통한 커리큘럼 추천 기능, 커리큘럼의 단계별 생성을 분석하여 User들의 만족도 및 retention을 더 높이도록 해야 한다.

2.3.2. User

CodeSunbi의 경우 코딩을 도와주는 도우미의 역할을 하기 때문에, 본 시스템의 주된 사용자는 초보 혹은 그에 준하는 사람들이 될 것이다. 코딩, 즉 프로그래밍의 경우 양이 방대하고 실제 개발 과정에서 생각지도 못하는 어려움, 다른 개발 지식들과 연계가 필요한 경우가 많다. 또한 어려움에 부딪혔을 때 필요한 정보들이 산발적으로 분포되어 있기 때문에 이러한 정보를 찾고 종합하는 과정이 어렵다. 이러한 특성들로 인해 공부 중 포기율이 높은 편이다. 따라서 본 시스템은 이러한 문제점을 보완하여 User들이 흥미를 잃지 않고 프로그래밍을 공부하도록 도와줄 것이다.

2.4 Constraints

- 코딩을 직접 해야 하기 때문에 모바일이 아닌 PC를 이용해야 한다.
- 사용자들은 커리큘럼에 맞게 자바스크립트, 언리얼 엔진, python등이 사용 가능한 PC 환경을 이용해야 한다.
- 개발은 Windows 10 환경을 대상으로 하고, 최소 요구 사양을 만족해야 한다.
- 코딩 초보자들을 위한 시스템이므로 각 기능에 대한 접근성과 가시성이 좋아야 한다.
- 서버와의 연동이 필요하기 때문에 데이터 관리 및 수정, 변경 시 데이터 보존이 중요하다.
- 사용자들의 개인 정보 및 자료를 관리하기 때문에 보안성에 특히 신경 써야 하고, 취약점을 끊임없이 보완해야 한다.
- 인터넷 기반 시스템이기 때문에 자바 스크립트가 구동되는 인터넷 환경이 지원되어야 한다.

2.5. Assumptions and Dependencies

본 시스템은 사용자 별 키워드에 따라 만들어지는 커리큘럼이 달라지기 때문에 사용자의 컴퓨터 하드웨어 환경과 소프트웨어 환경이 만들어지는 커리큘럼이 요구하는 최소 사양을 만족해야 한다.

Specific Requirements

3.1. External Interface Requirements

3.1.1. User Interface

이름	마우스 및 키보드를 통한 입력 처리
목적 / 내용	등록하지 않은 사용자
입력 주체 / 출력 목적지	사용자 / Windows나 Mac 기반의 컴퓨터 기기
범위 / 정확도 / 허용 오차	1) 범위 : 화면에서의 버튼의 개수에 따른 입력 범위. 2) 정확도 : 유저의 마우스 및 입력에 따른 정확도. 3) 허용 오차 : 해당 없음
단위	버튼 클릭 / 키보드 입력
시간 / 속도	비 정기적인 사용자의 입력 / 즉각적인 사용자 명령 수행
타 입출력과의 관계	입력 내용에 따라 클라이언트에서 처리 또는 서버로 명령 요청
화면 형식 및 구성	해당 없음
윈도우 형식 및 구성	해당 없음
데이터 형식 및 구성	Python 코드, Text, Int 형의 코드 값
명령 형식	각 코드 값에 따른 명령 매핑
종료 메시지	해당 없음

Table 3: User Interface of Input Processing

이름	사용자 등록 인터페이스
목적 / 내용	시스템에 사용자의 정보를 등록
입력 주체 / 출력 목적지	사용자 / 서버
범위 / 정확도 / 허용 오차	해당 없음
단위	화면
시간 / 속도	해당 없음
타 입출력과의 관계	해당 없음
화면 형식 및 구성	1) 새로운 계정에 대한 기본적인 정보 입력을 위한 슬롯이 존재 2) 이메일 정보 확인을 위한 버튼 및 인증 버튼 3) 계정 정보 등록을 위한 버튼
윈도우 형식 및 구성	해당 없음
데이터 형식 및 구성	Query, Text
명령 형식	해당 없음
종료 메시지	“회원가입이 완료 되었습니다.”

Table 4: User Interface of Register

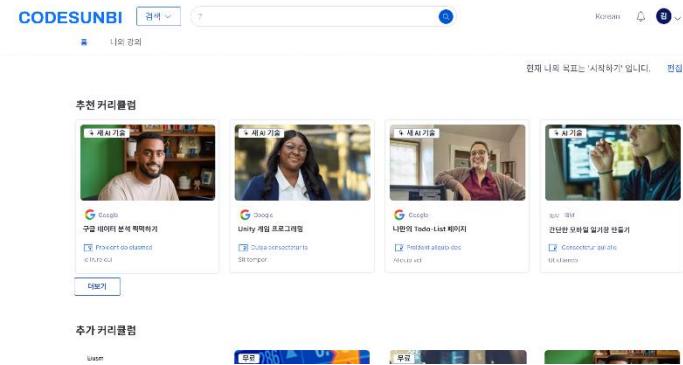
이름	모니터를 통한 메인 화면 출력
목적 / 내용	시스템 사용자에게 제공하는 인터페이스
입력 주체 / 출력 목적지	클라이언트 / 사용자
범위 / 정확도 / 허용 오차	1) 범위 : 화면에서의 버튼의 개수에 따른 입력 범위. 2) 정확도 : 유저의 마우스 및 입력에 따른 정확도. 3) 허용 오차 : 해당 없음
단위	화면
시간 / 속도	사용자의 입력(클릭)에 따른 화면 전환 (2초 이내)
타 입출력과 관계	해당 없음
화면 형식 및 구성	 <p>1) 개발할 웹 사이트는 위와 같은 형태로 구성 2) 메인 페이지에서 시스템이 추천하는 커리큘럼을 확인할 수 있음</p>
윈도우 형식 및 구성	<ul style="list-style-type: none"> - Grid Layout을 활용하여 추천 커리큘럼 및 추가 커리큘럼을 보여줌 - 화면 상단은 Navbar를 두어 검색 및 기본적인 회원정보 페이지로 넘어갈 수 있는 UI를 구성
데이터 형식 및 구성	강의 커리큘럼 (DB에서 가져옴), 예시 이미지, Text
명령 형식	해당 없음
종료 메시지	해당 없음

Table 5: User Interface of Main page

이름	모니터를 통한 학습 화면 출력
목적 / 내용	1) 목적 : 사용자가 선택한 커리큘럼을 학습하고, 챗터 별 학습 콘텐츠를 통해 실습을 수행할 수 있도록 지원합니다. 2) 내용 : 사용자는 "Course" 목록에서 챗터를 선택하여 각 챗터의 학습 내용을 확인하고, 코드 작성 및 실습 예제를 수행합니다. 학습 중 필요한 경우 "ChatGPT" 창에서 추가 자료를 확인하거나 질문을 통해 보조 학습 자료를 얻을 수 있습니다.
입력 주체 / 출력 목적지	클라이언트 / 사용자
범위 / 정확도 / 허용 오차	1) 범위 : 학습자는 각 챗터에 포함된 자료와 예제를 순차적으로 학습 2) 정확도 : 학습 자료 및 보조 자료는 정확하고 신뢰할 수 있는 정보로 제공 3) 허용 오차 : 학습자의 코드 작성 오류에 대해서는 즉각적인 피드백이 제공되지 않으나, 참고 자료를 통해 자가 학습이 가능하도록 구성
단위	화면
시간 / 속도	페이지 내 각 섹션의 로딩 시간은 3초 이내로 제한 ChatGPT 창에서 추가 자료를 검색하는 경우, 평균 응답 속도는 2초 이내를 목표로 합니다
타 입출력과의 관계	사용자가 챗터를 선택하면 해당 챗터에 맞는 코드와 설명이 중앙 코드 창에 표시됩니다
화면 형식 및 구성	1) 좌측에 Course 목록이 나열되어 있고, 중앙에는 코드 작성 창, 우측에는 ChatGPT 보조 자료 창과 채팅 입력 창이 배치되어 있습니다.
윈도우 형식 및 구성	- 단일 페이지 레이아웃을 사용하며, 각 섹션은 같은 페이지 내에서 스크롤 없이 구분됩니다. 추가 자료는 ChatGPT 창을 통해 바로 열람할 수 있습니다.
데이터 형식 및 구성	챗터 명과 설명은 텍스트 형식으로 제공되며, 코드 예제는 구문 강조가 적용된 Python 코드 형식으로 표시됩니다. 보조 자료는 텍스트 및 이미지 링크 형식으로 구성됩니다.
명령 형식	사용자는 챗터 목록에서 챗터를 클릭하여 챗터 내용을 불러옵니다. ChatGPT 창에서 질문을 입력하고 "Send" 버튼을 클릭하여 질문을 보냅니다.
종료 메시지	사용자가 모든 챗터를 완료했을 때 "학습을 완료하셨습니다. 수고하셨습니다!"라는 메시지가 표시됩니다.

Table 6: User Interface of Learning page

3.1.2 Hardware Interface

이름	시스템 하드웨어 디바이스
목적 / 내용	키보드, 마우스를 이용한 입력
입력 주체 / 출력 목적지	사용자 / 서버
범위 / 정확도 / 허용 오차	해당 없음
단위	해당 없음
시간 / 속도	사용자의 입력 / LLM이 처리하는 속도
타 입출력과의 관계	해당 없음.
화면 형식 및 구성	화면의 중앙에 위치
윈도우 형식 및 구성	해당 없음
데이터 형식 및 구성	자연어
명령 형식	자연어 및 코드
종료 메시지	해당 없음

Table 7: Hardware Interface

3.1.3 Software Interface

이름	LLM과의 연동
목적 / 내용	결과 출력 / LLM을 통해 자연어 처리 후 외부 resource에서 관련 답을 찾아 가공 후 출력
입력 주체 / 출력 목적지	사용자의 자연어 / 서버 및 화면
범위 / 정확도 / 허용 오차	Chrome, Edge 등의 웹 브라우저 및 LangChain과 external source의 버전/ 해당 없음 /Chrome 100.0이상, Edge 85이상, LangChain은 최신 버전 유지
단위	token
시간 / 속도	관련 코드 요청 시간/ 4s 강의 요청 시간 / 4s 코딩 커리큘럼 제공 / 8s 코드 정답 유무 판별 / 4s
타 입출력과의 관계	LangChain을 통해 External source와 연동 / 주어진 자료들을 가공하여 화면에 출력
화면 형식 및 구성	웹사이트 내에 출력, 화면의 우측에 존재.
윈도우 형식 및 구성	해당 없음
데이터 형식 및 구성	JSON, Python
명령 형식	자연어
종료 메시지	해당 없음, Max token 설정

Table 8: Software Interface

3.1.4 Communication Interface

이름	호스트 - 클라이언트
목적 / 내용	서버와 사용자 간의 학습 상태 최신화 및 데이터화/서버에 접속 요청, 과거 학습 기록 및 데이터 불러오기
입력 주체 / 출력 목적지	사용자 / 서버
범위 / 정확도 / 허용 오차	WAN / 중간 수준의 정확도 / 중간 수준의 허용 오차
단위	JSON
시간 / 속도	로그인 / 아웃 요청 시간 / 5s 회원가입 요청 처리 / 4s 1Mbps 이상의 인터넷
타 입출력과의 관계	코드 입력 시 외부 소스를 통한 답변 / 서버에 저장되어 있는 데이터 사용
화면 형식 및 구성	화면 우측
윈도우 형식 및 구성	해당 없음
데이터 형식 및 구성	패킷과 log
명령 형식	Post()
종료 메시지	Close()

Table 9: Communication Interface

3.2 Functional Requirements

3.2.1 Use Case

<i>Use case name</i>	<i>Register</i>
Actor	등록하지 않은 사용자
Description	1) 미등록 사용자가 시스템의 대부분의 기능을 이용하기 위해 회원으로 등록을 시도하는 과정이다. 2) 회원가입은 등록과 동일한 의미이다.
Normal Course	1. 모든 사용자는 웹 사이트 접속 시 로그인 및 회원가입을 할 수 있는 초기 페이지가 나타난다. 2. 회원가입을 한 사용자에게만 서비스가 되는 것을 확인한 후, 미등록 사용자는 로그인 페이지에서 등록 버튼을 클릭한다. 3. 사용자가 페이지를 등록하도록 등록 페이지로 이동한다. 4. 사용자는 회원 가입 양식에 따라 추가적인 정보를 입력해야 한다. 필요한 정보는 다음을 포함한다. <ul style="list-style-type: none"> a. ID (Email 주소) b. Password c. 이름 d. 나이 e. 성별 f. 전화번호 5. 시스템은 해당 이메일 주소와 비밀번호가 적합한 지 확인을 한 다음 만약 비밀번호를 찾게 되는 상황이 발생할 수 있기에 메일 주소를 인증하는 코드를 통하여 인증한다. 6. 양식대로 모두 작성한 다음 완료를 클릭하면 사용자가 등록되고 로그인 페이지로 돌아가게 된다.
Precondition	<ul style="list-style-type: none"> • 사용자가 아직 시스템에 등록되어 있지 않아야 한다. • 사용자가 올바른 정보를 입력해야 한다. • 동일한 이메일 주소를 다른 사용자의 이메일 주소와 중복해서는 안 된다. • 잘못된 입력이 있는 경우, 시스템은 이를 확인하여 경고 메세지나 알림을 주어 사용자에게 알려주어야 한다. • 사용자가 필수 입력사항(ID, PW)를 제외하고 나머지 정보는 선택 사항으로 입력해야 한다.
Post Condition	보안을 위하여 비밀번호를 암호화를 진행한 다음 DB에 해당 정보를 저장한다.
Assumptions	해당 없음.

Table 10: Use Case of Register

<i>Use case name</i>	<i>Log-in / out</i>
Actor	등록한 사용자
Description	<p>1) 로그인은 시스템의 등록된 사용자가 서비스를 사용하기 위하여 시스템에 접속하려고 하는 프로세스이다.</p> <p>2) 로그아웃은 로그인 한 사용자가 시스템에서 나가려고 할 때의 프로세스이다.</p> <p>3) 회원가입은 등록과 동일한 의미이다.</p>
Normal Course	<p>* 로그인</p> <ul style="list-style-type: none"> - 시스템에 이미 회원으로 등록한 사용자가 시스템에서 서비스를 사용하려고 한다. - 회원가입을 하는 과정에서 설정한 Email 주소와 비밀번호를 입력한다. - 만약 입력한 정보가 올바르다면 시스템은 사용자가 시스템에 접속할 수 있도록 해야 하며, 이제 사용자는 시스템의 메인 페이지로 이동하게 되어 시스템에서 제공하는 서비스를 이용할 수 있다. <p>* 로그아웃</p> <ul style="list-style-type: none"> - 시스템을 종료하려면 우측 상단의 본인의 성으로 되어 있는 아이콘을 클릭하여 나온 여러 아이템들 중 Logout 버튼을 클릭한다. 사용자가 만약 로그아웃을 하지 않고 웹 페이지를 닫는 경우에는 시스템의 임의로 사용자의 세션을 닫을 수 있다.
Precondition	<p>* 로그인 : 사용자가 시스템에 이미 회원가입한 상태여야 한다.</p> <p>* 로그아웃 : 사용자가 로그인 한 상태여야 한다.</p>
Post Condition	사용자가 온라인 상태여야 한다. (웹 페이지에 접속해 있는 상태)
Assumptions	해당 없음.

Table 11: Use Case of Log-in / out

<i>Use case name</i>	<i>Keyword Selection</i>
Actor	최초 등록한 사용자 혹은 새로운 커리큘럼을 원하는 사용자
Description	1) 키워드는 사용자가 본 서비스에서 관심 있어 하는 즉 배우고 싶어 하는 것들을 단순화 해 놓은 단어를 의미한다. 2) 사용자는 자신이 배우고자 하는 것, 즉, 만들고 싶은 프로그램이나 배우고 싶은 언어 등의 키워드를 선택한다. 3) 시스템이 사용자를 위한 맞춤 교육 과정을 추천하기 위한 프로세스이다.
Normal Course	1. 모든 사용자는 최초 로그인 시 키워드를 선택할 수 있는 키워드 선택 페이지가 나타난다. 2. 사용자는 키워드를 선택한다. 2.1. 사용자는 시스템이 제시한 3가지 키워드 중 자신이 선호하는 키워드를 하나 선택한다. 2.2 키워드와 연관된 키워드가 지속적으로 3개씩 나와 결론적으로 총 4개의 키워드를 선택할 수 있는 프로세스를 거친다. 3. 선택 완료 버튼을 누른다. 4. 시스템의 메인 페이지인 교육 과정을 선택할 수 있는 페이지로 이동하게 된다.
Precondition	1) 사용자가 로그인을 한 상태여야 한다. 2) 사용자가 최초로 시스템에 접속했을 때 진행한다.
Post Condition	1) 사용자가 학습을 진행할 수 있는 상태여야 한다. 2) 사용자에게 맞춤인 강의 커리큘럼이 메인 페이지에 floating 되어 있어야 한다. 3) 사용자가 원하는 커리큘럼이 완성되지 않을 경우, 사용자는 직접 커리큘럼을 만들 수 있다.
Assumptions	해당 없음.

Table 12: Use Case of Keyword Selection

<i>Use case name</i>	<i>Curriculum Learning</i>
Actor	커리큘럼 생성을 완료한 사용자
Description	1) 학습 과정(Curriculum)은 회원으로 등록된 사용자가 선택한 키워드를 바탕으로 시스템에서 사용자에게 맞춤으로 제공하는 교육 과정을 말한다. 2) 커리큘럼 러닝은 회원으로 등록된 사용자가 시스템에서 제공한 교육 과정을 학습하는 과정을 말한다.
Normal Course	1. 사용자는 웹 페이지의 메인 페이지에서 자신이 듣고자 하는 커리큘럼을 선택한다. 2. 시스템은 사용자의 기존 학습 정보를 불러온 뒤 알맞은 학습 페이지로 이동시킨다. 3. 정해져 있는 모든 챕터 학습을 완료할 경우 학습이 완료된다. 4. 학습 중간에 종료한다면 해당 학습 정보가 DB에 저장된다.
Precondition	1) 사용자가 로그인을 한 상태여야 한다. 2) 사용자가 Keyword Selection을 마친 상태여야 한다.
Post Condition	1) 커리큘럼의 특정 챕터 완료 표시. 2) 커리큘럼 내 진도가 표시 되어야 한다. (학습률)
Assumptions	해당 없음.

Table 13: Use Case of Curriculum Learning

<i>Use case name</i>	<i>LLM Q&A Interaction</i>
Actor	등록한 사용자
Description	1) 학습 과정에서 모르는 것이나 질문 사항 등이 생길 경우 학습 페이지 내부에서 시스템에서 제공하는 LLM을 활용하여 질문을 주고 받는 것을 말한다.
Normal Course	1. 사용자는 학습 페이지에서 학습을 진행하는 와중에 질문이 생긴다. 2. 학습 페이지 우측에 존재하는 LLM을 활용하여 질문을 한다. 2.1. 사용자는 학습을 진행하다 모르는 것을 LLM에게 질문할 수 있다. 2.2. 사용자가 LLM에게 예시 코드 등과 같은 정답 코드를 요청할 수 있다. 2.3. LLM은 사용자의 강의 진행률이 미비할 때 사용자를 위한 강의를 추천해줄 수 있다. 3. 시스템은 해당 질문에 맞는 답변을 사용자에게 제공한다.
Precondition	1) 사용자가 로그인을 한 상태여야 한다. 2) 사용자가 학습(Curriculum Learning)을 진행하는 중이어야 한다.
Post Condition	사용자에게 질문에 대응되는 답변 제시
Assumptions	해당 없음.

Table 14: Use Case of LLM Q&A Interaction

3.2.2 Use Case Diagram

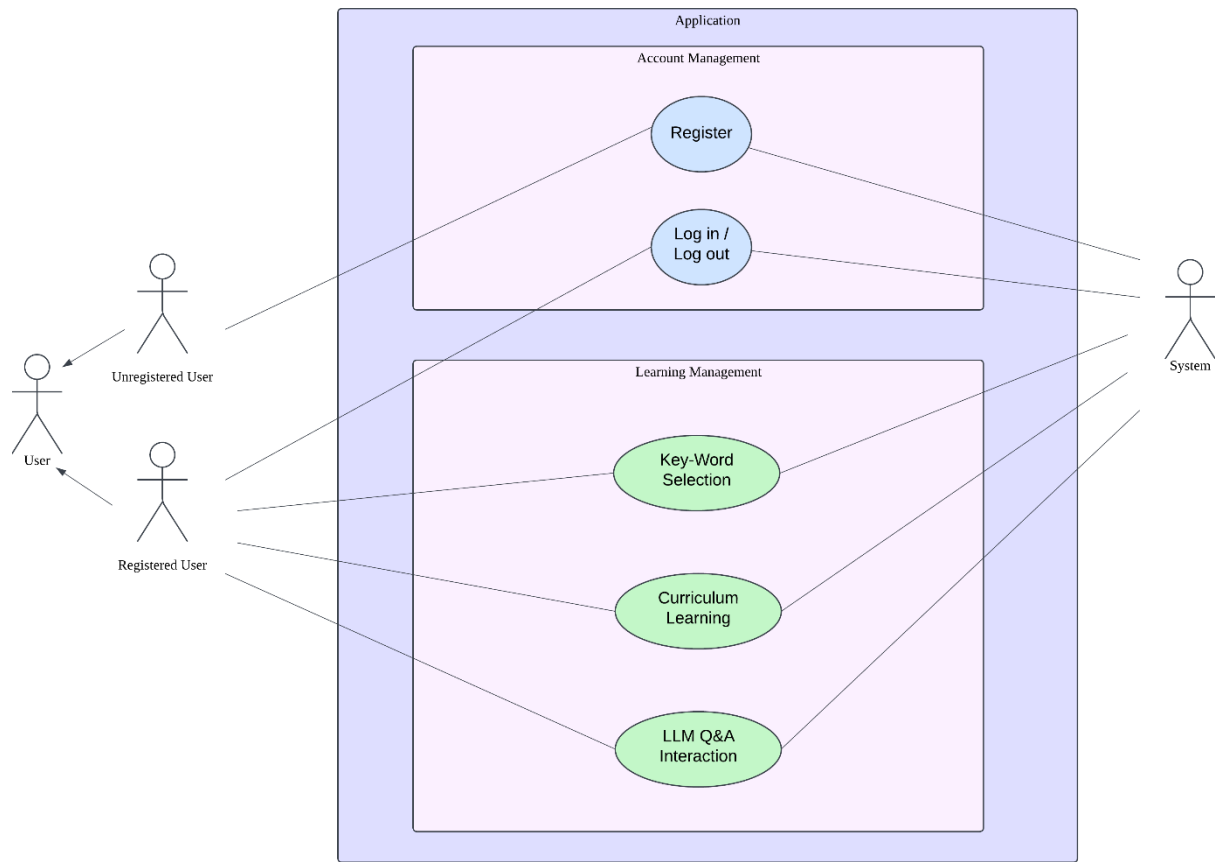


Figure 1: Use Case Diagram

3.2.3 Data Dictionary

Account				
Field	Key	Constraint	Description	
id	PK	Not Null	User id	
password		Not Null	User password	
name		Not Null	User's name	
age		Not Null	User's age	
gender		Not Null	User's gender	
phone		Not Null	User's phone number	
keyword			사용자가 선택한 키워드	
goal			사용자의 학습 목표	

Table 15: Data Dictionary of Account

Curriculum			
Field	Key	Constraint	Description
curriculum_id	PK	Not Null	커리큘럼 아이디(고유 식별자)
curriculum_name		Not Null	커리큘럼 이름
curriculum_type			커리큘럼의 유형 (ex: game programming, web programming)
description			커리큘럼에 대한 설명

Table 16: Data Dictionary of Curriculum

Module			
Field	Key	Constraint	Description
module_id	PK	Not Null	모듈 고유 식별자
curriculum_id	FK	Not Null	해당 모듈이 속한 커리큘럼
module_name		Not Null	모듈 이름(예: Chapter 1: Make Character Class)
module_type			모듈 유형 (예 : Introduction, Chapter)
description			모듈 설명
order_index		Not Null	모듈 순서를 정렬하기 위한 인덱스

Table 17: Data Dictionary of Module

Curriculum Progress			
Field	Key	Constraint	Description
id	FK	Not Null	User id
curriculum_id	FK	Not Null	Key Word of this curriculum
completed_modules		Not Null	User's learning progress
total_modules			전체 모듈 수
completion_percentage		(default 0)	커리큘럼 진행률 (%)
Primary Key (user_id, curriculum_id)	PK		복합 기본 키 (하나의 유저가 여러 개의 커리큘럼을 수강할 수 있기에)

Table 18: Data Dictionary of Curriculum Progress

Training Log			
Field	Key	Constraint	Description
log_id	PK	Not Null	학습 로그 ID
user_id	FK	Not Null	사용자 ID
module_id	FK	Not Null	학습한 모듈의 ID
date		Not Null	학습 날짜
notes			학습 중 사용자가 작성한 메모
duration_hours			학습 시간 (단위 : 시간)
LLM_comment			사용자의 학습을 평가한 LLM comment

Table 19: Data Dictionary of Training Log

User Code			
Field	Key	Constraint	Description
code_id	PK	Not Null	학습 로그 ID
user_id	FK	Not Null	사용자 ID
module_id	FK	Not Null	학습한 모듈의 ID
code			사용자가 작성한 코드
created_at		DEFAULT (현재시각)	코드 작성 일자

Table 20: Data Dictionary of User Code

3.2.4 Data Flow Diagram

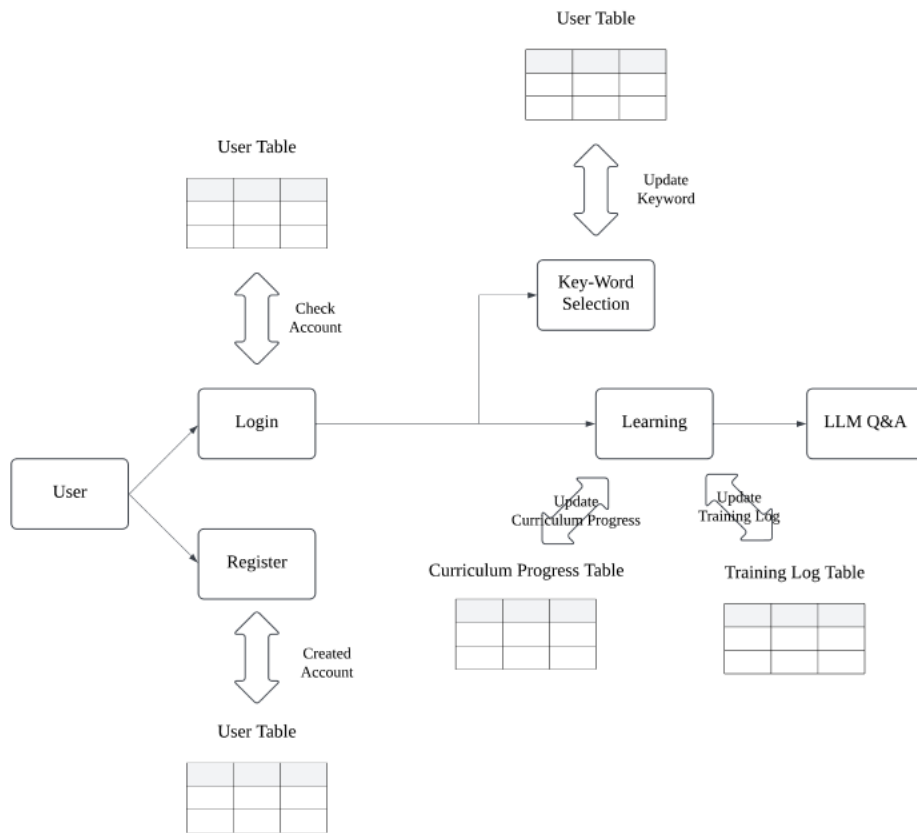


Figure 2: Data Flow Diagram

3.3 Performance Requirements

아래는 본 시스템의 성능의 요구사항에 관한 것이다. 성능의 요구사항은 정적인 것과 동적인 것으로 나누어 분석하였다.

3.3.1 Static Numerical Requirements

- 본 시스템은 여러 명의 사용자가 동시에 접속해서 코딩 교육에 대한 서비스를 받는 것을 지원한다. 사용자는 본 시스템에 계정을 가지고 있어야 한다.
- 본 시스템은 웹 기반의 서비스이며, Chrome 기반의 웹 브라우저 기반으로 개발되었다.
- 1Mbps 이상의 인터넷 연결 속도를 요구한다. 만약 해당 인터넷 연결 환경에서 접속하지 않을 시 제공되는 서비스가 정상적으로 작동하지 않을 수 있다.

3.3.2 Dynamic Numerical Requirements

- 시스템은 동시에 최소 100 명의 사용자 접속을 유지할 수 있어야 된다. 또한 사용자의 계정을 10,000 명까지 관리할 수 있어야 된다.
- 사용자가 코딩 진행 시 LLM 으로 코드의 일부분에 대한 도움을 요청했을 때 4 초 안에 LLM 이 정답 코드의 일부분을 출력해야 된다.

- 사용자가 코딩 진행 시 코드 관련된 강의 영상 등을 요청 시, LLM은 사용자가 진행 중인 코딩 교육에 맞는 강의 영상을 4초 내에 완료해 제공해야 한다.
- 사용자의 로그인/로그아웃 요청을 5초 내에 완료해야 된다.
- 사용자의 코딩 커리큘럼에 대한 정보를 8초 내에 구성하여 사용자에게 제공하여야 한다.
- 사용자의 회원가입 요청을 4초 내에 완료해야 된다. 만약 아이디나 비밀번호 등의 요구사항이 invalid 할 시 사용자에게 2초 내에 관련된 적절한 정보를 제공하여야 된다.
- 사용자가 코드를 제출할 시에 정답 유무를 8초 내에 제공하여야 된다.

3.4. Logical database requirements

시스템은 사용자의 학습 로그를 관리할 수 있어야 한다. 사용자 별로 학습 로그를 관리하기 위해서 ID, GOAL, CURRICULUM, PROGRESS, TRAINING LOG, LLM COMMENT 등의 정보를 담을 수 있는 JSON 형식의 데이터를 사용한다. 그러나 JSON 형식의 데이터는 일반적인 관계형 데이터베이스에서 다룰 수 없는 형식이기 때문에 이를 지원하기 위해 NoSQL 문서 지향 데이터베이스인 MongoDB를 사용하여 관리한다. 해당 데이터베이스를 통하여 시스템 사용자들의 계정 정보를 저장하고, User ID를 Key 값으로 사용하여 학습 로그를 불러올 수 있어야 한다. 새로운 진행 상황이 생길 때마다 지속적으로 POST 요청을 통해 서버에 저장할 수 있어야 하며, 사용자의 PROGRESS 및 TRAINING LOG를 LLM에게 prompting하여 LLM으로 하여금 이번 학습에 대한 피드백, 즉 LLM COMMENT를 생성하도록 한 후 학습 로그에 덧붙여 서버에 저장한다. 각 사용자가 이전 학습 내용을 복습하거나 확인하기 위해 요청한다면, 애플리케이션이나 서버가 사용자의 요청을 SQL 쿼리로 변환하고 데이터베이스에서 해당 키를 기반으로 사용자의 이전 학습 기록을 가져와 제공할 수 있어야 한다.

3.5. Design constraints

3.5.1. Physical design constraints

시스템의 목적은 이용자의 LLM을 보조 도구로 한 코딩 교육이므로 데스크탑 또는 노트북 기기를 이용할 것을 권장하며, 모바일 기기 환경은 권장하지 않는다. 시스템은 MongoDB 데이터베이스를 이용하여 필요한 데이터를 해당 데이터베이스에 저장할 수 있어야 한다. 또한 시스템은 최소한의 지연 시간으로 빠른 데이터 검색과 저장을 지원해야 하며, 서버는 다수의 동시 사용자 요청을 처리할 수 있는 성능을 제공해야 한다. MongoDB 데이터베이스는 주기적인 백업과 비상 시 복구 기능을 지원하여 데이터 유실방지 및 안정성을 보장해야 한다. 시스템은 향후 이용자 증가에 대비해 수평적 및 수직적 확장이 가능하도록 설계되어야 한다.

3.5.2. Standards compliance

시스템은 웹 어플리케이션으로 Python Django를 이용해 개발되며, Django 코딩 표준(Python style) 및 HTML 표준을 따른다. 변수 이름은 lower 카멜 케이스를 따르며, 함수와 데이터베이스는 파스칼 케이스를 따른다. 데이터 교환 포맷은 JSON 형식을 사용해야 한다. 코드베이스는 Git 버전 관리 시스템을 통해 관리되며, 명확한 commit 메시지와 적절한 분기 전략(Git Flow)을 사용하여 개발 프로세스의 일관성과 가시성을 유지해야 한다.

3.6 Software System Characteristics

3.6.1. Reliability

- 여러 환경에서 하나의 계정으로 로그인 했을 때, 기존 로그인 되어 있던 계정은 로그아웃이 되어야 한다. 또한 회원가입의 경우 4초 내로 처리할 수 있어야 한다.
- 사용자의 코딩 커리큘럼에 대한 정보를 8초 내에 구성하여 사용자에게 제공하여야 한다.
- LLM은 사용자의 요청이나 질문을 4초안에 처리하여 답변하여야 한다.
- LLM은 사용자에게 강의를 제공할 때 4초 내에 진행해야 하고, 오류가 없어야 한다. 또한 강의 내용은 주어진 커리큘럼과 관련된 내용이어야 한다.
- 사용자가 코드를 제출할 때 8초안에 정답 유무를 체크하여 알려줄 수 있어야 한다.
- 최종적으로 만들어진 프로젝트에 오류가 없어야 한다.

3.6.2. Availability

- 서버는 24시간 가동되어야 하며, 사용자가 물리는 기간이나 시간대를 파악하여 트래픽 관리를 용이하게 하도록 해야 한다.
- 시스템은 동시에 최소 100명의 사용자 접속을 유지할 수 있어야 된다. 또한 사용자의 계정을 10000명까지 관리할 수 있어야 된다.

3.6.3 Security

- User은 각 개인의 ID와 Password를 이용해야 본 시스템에 접근할 수 있다.
- User은 자신의 ID, Password를 찾을 때 검증된 이메일과 연동하여 사용자 정보를 획득할 수 있어야 한다.
- 사용자의 개인 정보가 LLM에 입력되었을 때, 이를 암호화하여 유출을 막아야 한다.
- 서버에 저장된 데이터의 경우 Manager만 접근이 가능하도록 한다. 또한, 방화벽 등으로 2차 보안을 강화한다.
- 사용자 데이터는 암호화해서 저장한다.

3.6.4 Maintainability

- 시스템은 모듈화를 통해 일반화하여 이후 수정 및 개선이 필요할 때 관련 class나 component들의 연결성을 해치지 않아야 한다.
- 각 기능의 변경 혹은 개선 사항이 있을 경우 이를 수정하고 문서에 기록하여 이후 관련된 추가 문제가 발생할 때 참고할 수 있도록 해야 한다.
- 코드 복잡도를 줄이고 가독성을 높이기 위해 refactoring을 일정 기간마다 진행하고 pair programming을 통해 점검한다.
- 다양한 사이즈의 화면을 가진 PC에서 모두 정상적으로 작동되고 보여져야 하기 때문에 이를 고려하여 유동적으로 배치될 수 있는 interface를 구성한다.

3.6.5 Portability

- 웹 기반 서비스이기 때문에 타 웹, 혹은 기존의 웹의 업데이트에 맞춰 사항들을 변경해야 할 수 있다. 따라서 platform 의존성과 관련된 부분을 따로 관리한다.
- dependency가 높은 코드들과 그렇지 않은 코드들을 구별하고, 이를 각각 관리하여 이후 발생할 수 있는 호스트 시스템/운영 체제의 변경에 대응하기 쉽게 한다.

3.7. Organizing the specific requirements

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

3.7.1. Context Model

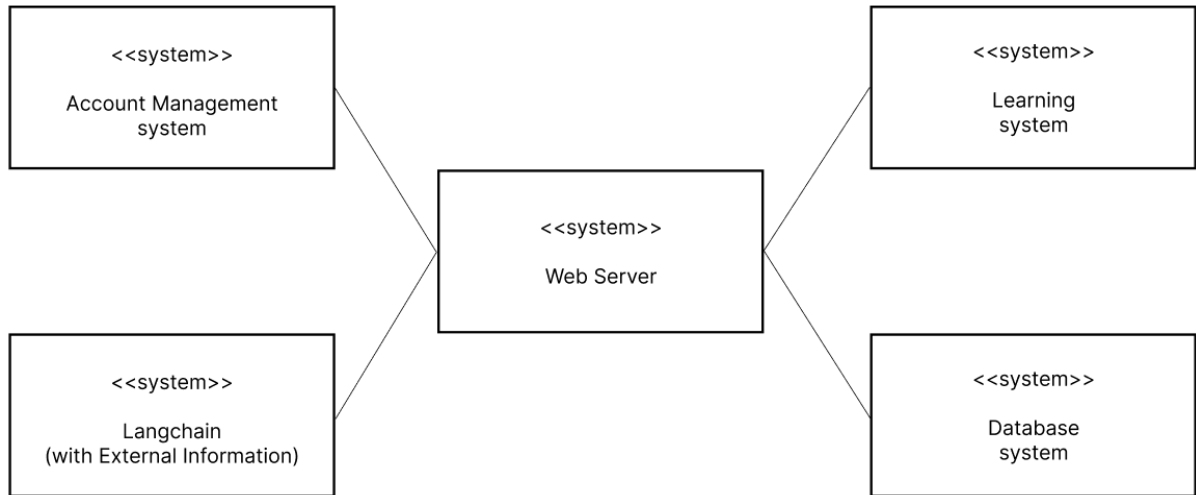


Figure 3: Context Model

3.7.2. Process Model

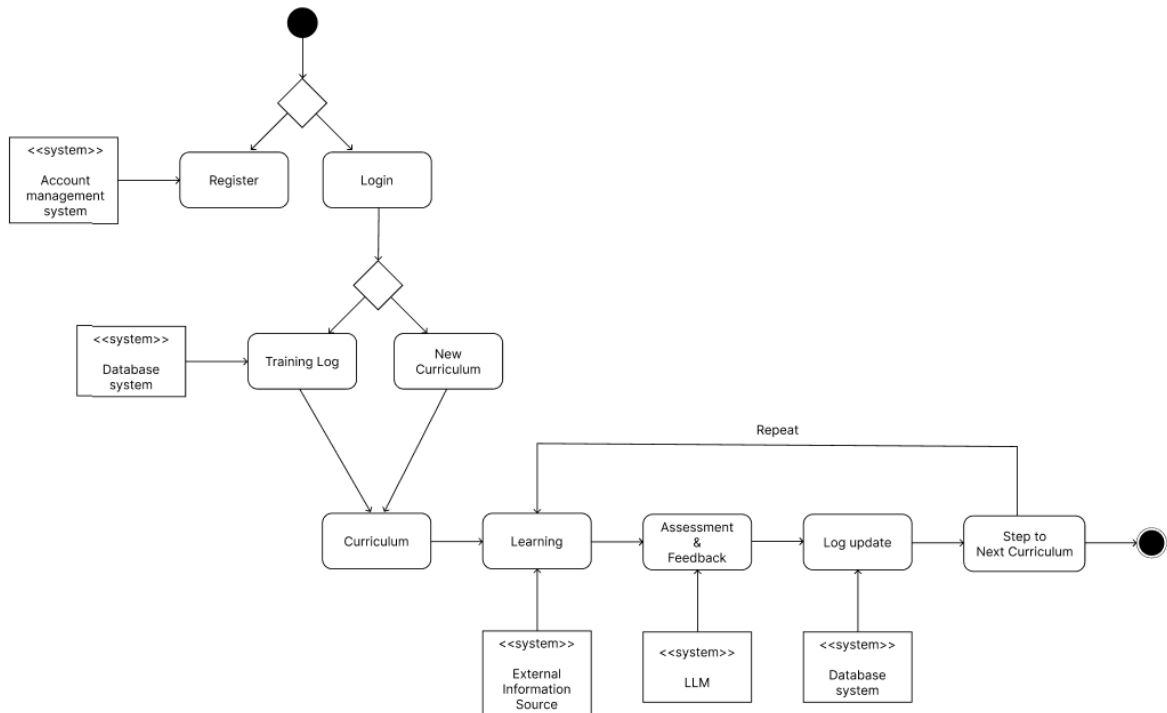


Figure 4: Process Model

Curriculum, Learning, Step to Next Curriculum의 어느 때든 사용자는 Log out을 통해 학습을 종료할 수 있다. Log out하는 경우 현재까지의 Training Log를 데이터베이스에 업데이트하고 저장한다.

3.8 System Architecture

본 챕터에서는 프로젝트의 system architecture에 대한 high-level 개요를 제시한다. System component에 대한 정보를 제시하고 각각의 관계 및 통신의 관계를 명시한다.

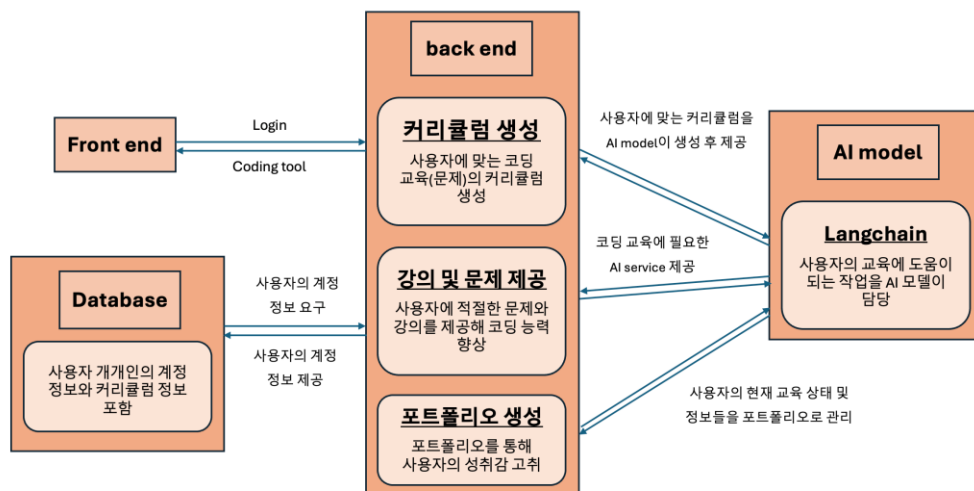


Figure 5: System Architecture

3.9 System Evolution

본 시스템은 사용자의 개개인에 맞는 코딩 교육 커리큘럼을 제공하고 상당한 수준의 LLM 서비스를 제공해 코딩 교육의 퀄리티를 높인다. 하지만 LLM은 Langchain으로 서비스 되기 때문에 Langchain이 담당하는 data가 주기적으로 시대에 맞게 최신화 되지 않으면 코딩 교육 서비스의 질이 떨어질 수 있다. 때문에 트렌드와 시대에 맞는 코딩 교육 서비스를 제공하기 위해 핵심 기능인 LLM의 Langchain의 웹 data를 현재의 트렌드에 맞게 읽어서 시대에 뒤떨어지지 않는 코딩 교육 서비스를 제공할 것이다.

또한 교육 서비스 중간중간에 사용자의 피드백을 받아 교육의 방향을 조정하는 기능까지 추가할 것이다. 사용자마다 개개인의 기호와 스타일이 다르기 때문에 사용자의 피드백을 바탕으로 교육 서비스의 방향을 조정하면 더 나은 코딩 교육의 질을 기대할 수 있다.