

GPTeacher

Software Requirements Specification

소프트웨어공학개론 2 조

김형진, 박혜인, 서민경, 신재욱, 이승민

2024-11-9

CONTENTS

1 Introduction	1
1.1 Purpose	1
1.2 Scope	1
1.3 Definitions, Acronyms, and Abbreviations	1
1.3.1 Definition	1
1.3.2 Acronyms and Abbreviations	1
1.4 References	2
1.5 Overview	2
2 Overall Description	3
2.1 Product Perspective	3
2.1.1 User Interfaces	3
2.1.2. Hardware Interfaces	3
2.1.3. Software Interfaces	3
2.1.4. Communication Interfaces	3
2.1.5 Memory Constraints	4
2.1.6. Operations	4
2.2 Product Functions	4
2.2.1 회원가입	4
2.2.2 로그인	4
2.2.3 단계별 힌트 제공	4
2.2.4 맞춤형 변형 문제 제공	4
2.2.5 맞춤형 관심 분야 문제 제공	4
2.3 User Classes and Characteristics	5
2.3.1 시스템 관리자	5

2.3.2 학습자	5
2.4 Design and Implementation Constraints	5
2.5 Assumptions and Dependencies	5
3 External Interface Requirements	7
3.1 External Interfaces	7
3.1.1. User Interface	7
3.1.2 Hardware Interface	11
3.1.3 Software Interface	11
3.1.4 Communication Interface	12
3.2 Functional Requirement	14
3.2.1. Use Case	14
3.2.2. Use Case Diagram	18
3.2.3. Data Flow Diagram	19
3.3 Performance Requirements	19
3.3.1 Static Numerical Requirement	19
3.3.2 Dynamic Numerical Requirement	19
3.4 Logical Database Requirements	20
3.5 Design constraints	20
3.5.1 Frontend Standard	20
3.5.2 Backend Standard	21
3.5.3 Database Standard	21
3.5.4 GPT Model Integration	22
3.5.5 Security and Privacy Protection	22
3.5.6 Performance	22
3.6 Software System Attributes	22
3.6.1 Reliability	22
3.6.2 Availability	22

3.6.3 Security.....	23
3.6.4 Maintainability.....	23
3.6.5 Portability.....	23
3.7 Software System Attributes	25
3.7.1 Context Model	25
3.7.2 Process Model	25
3.8 System Architecture	26
4 Appendixes	27
4.1 Document History	27

1

Introduction

1.1. Purpose

본 문서는 코딩을 배우고 싶은 모든 비전공자를 대상으로 하는 개인 맞춤형 문제 풀이 서비스에 대한 requirement 를 명세화 하여 design, development, validation, evolution 을 원활히 할 수 있도록 작성되었다.

1.2. Scope

본 프로그램의 이름은 *GPTeacher* 이며, 코딩을 배우고자 하는 학습자의 요청에 한해서 단계별 힌트를 제공하거나, 개인 맞춤형 변형 문제 및 관심 분야 관련 문제를 제공함으로써 학습자에게 개인 선생님과 같은 작업을 수행하게 된다.

1.3. Definitions, Acronyms, and Abbreviations

1.3.1. Definition

LLM: 대규모의 언어 데이터를 학습하여 인간의 언어인 자연어를 이해하고 생성하는 인공지능 언어 모델

프롬프트: 인공지능 모델에 입력하는 텍스트

시스템: 알고리즘 트레이닝 시스템

1.3.2. Acronyms and Abbreviations

LLM: Large Language Model

GPT: Generative Pre-trained Transformer

GHz: Gigahertz

GB: Gigabyte

Gbps: Gigabits per second

Mbps: Megabits per second

CPU: Central Processing Unit

RAM: Random Access Memory

DB: Database

3NF: Third Normal Form

HTTPS: Hypertext Transfer Protocol Secure

API: Application Programming Interface

OS: Operating System

JSON: JavaScript Object Notation

1.4. References

python

PEP 8: <https://peps.python.org/pep-0008/>

PEP 484: <https://peps.python.org/pep-0484/>

PEP 257: <https://peps.python.org/pep-0257/>

1.5. Overview

본 소프트웨어 요구사항 명세서는 네 개의 챕터로 구성되어 있다.

첫번째 챕터에서는 우리 2 조가 만드는 코딩 교육 프로그램 *GPTeacher*가 제공하는 서비스와 그 목적에 대한 소개, 본 명세서에서 사용하는 용어 및 약어를 설명하고 참조한 문서의 목록을 제시한다.

두번째 챕터에서는 제품의 개발 동기/목적/효과를 설명하고 제품이 수행할 주요 기능과 그 관계를 다이어그램 등을 사용해 나타낸다. 개발자의 선택을 제한하는 기타 항목 및 명세서의 요구 사항에 미치는 요소들을 설명한다.

세번째 챕터에서는 입출력 처리 과정과 소프트웨어와 소프트웨어, 소프트웨어와 사람 사이의 상호작용 시 요구되는 정적/동적 요구사항을 설명한다. 데이터베이스에 저장될 정보에 대한 논리적인 요구사항을 정의하며, 제약사항을 설명한다. 제품의 5 가지 특성을 구현하는 요소들에 대해 설명한다.

네번째 챕터는 본 명세서가 참고한 기준과 문서에 대해 설명한다.

2

Overall Description

2.1. product perspective

4 차 산업 시대에 코딩을 배우고자 하는 비전공자들이 늘어나고 있고, 코딩 교육에 대한 수요가 증가하고 있다. 또한 LLM 은 ChatGPT 를 필두로 다양한 분야의 지식을 가지고 여러 프로그램을 해석하거나 제작하는 능력을 갖추었다. 이에 LLM 을 활용해 비전공자도 코딩을 재미있게 배울 수 있는 문제 풀이 서비스를 제공하고자 한다. 사용자는 자신이 풀기 어려운 문제를 LLM 을 통해 단계별로 힌트를 얻어 해결할 수 있다. 또한 사용자는 LLM 이 제작한 자신이 관심있는 분야의 문제를 해결할 기회를 얻어 높은 성취감을 얻을 수 있다. 이 같은 경험을 제공하여 비전공자도 자신이 해결하고자 하는 문제를 코딩을 통해 해결하는 능력을 함양할 수 있다.

2.1.1. User Interfaces

사용자는 자신의 로컬 컴퓨터에 연결된 모니터 상에서 서비스에 대한 인터페이스를 제공받는다. 사용자는 자신의 정보를 시스템에 등록하고, 문제를 풀기 위한 코드를 작성하며, 힌트 제공 버튼 및 변형 문제 생성 버튼을 이용해 해당 기능들을 이용할 수 있다.

2.1.2. Hardware Interfaces

본 서비스는 호스트 서버에서 구동되며, 웹 사이트 기반으로 제공된다. 사용자의 로컬 컴퓨터는 1.2hz 이상의 클럭 속도를 가지는 CPU 를 가져야 한다.

2.1.3. Software Interfaces

사용자의 로컬 컴퓨터는 Window 10 또는 11 이상의 운영체제가 설치되어 있어야 하며, Google Chrome, Safari, Edge, FireFox 를 이용할 수 있어야 한다.

2.1.4. Communication Interfaces

LLM 은 ChatGPT -4o 버전을 이용한다. Host Server 는 문제, 학습 진도, 고객의 관심 분야를 바탕으로 힌트 제공 또는 문제 제작 요청 프롬프트 엔지니어링을 수행한다.

2.1.5 Memory Constraints

사용자의 로컬 컴퓨터는 4GB 이상의 RAM 을 탑재하고 있어야 한다.

2.1.6. Operations

Database Administrator 는 고객의 개인 정보에 접근하여 정보의 무결성을 체크할 수 있어야 하고, data backup 에 대한 책임을 진다.

2.2. Product Function

2.2.1 회원가입

시스템을 사용하기 위해 학습자에 대한 등록을 하기 위해 필요한 기능이다. 자신의 이름, 핸드폰 번호, 이메일 등의 개인정보와 아이디, 비밀번호, 그리고 관심 분야를 등록하고 이러한 정보들은 데이터베이스에 저장되게 된다. 입력한 핸드폰 번호를 통해 본인인증을 진행하게 되어, 다른 사용자와 중복되는 정보를 사용할 수 없게 된다.

2.2.2 로그인

학습자가 자신이 등록한 아이디와 비밀번호를 입력함으로써 식별되고, 서비스를 이용할 수 있게 된다. 백업 시스템을 통해 저장된 자신의 학습 이력과 문제 풀이 이력 등을 불러와, 자신이 학습했던 부분부터 이어서 다시 학습할 수 있다.

2.2.3 단계별 힌트 제공

학습자가 단계별 힌트를 요청할 시, 문제의 키워드나 중요한 문장을 강조해주고, 의미와 개념을 설명해주고, 마지막에 문제 해결에 필요한 자료구조와 알고리즘을 설명해줌으로써 학습자 스스로 자료구조 및 알고리즘을 추론할 수 있게 하는 학습능력을 키우게 한다.

2.2.4 변형 문제 제공

학습자가 문제 풀이를 완료한 후 변형 문제 제공을 요청할 시, 해당 문제의 변형 문제를 제공한다. 이를 통해 학습자는 복습하며 문제 풀이 능력을 향상시킬 수 있다.

2.2.5 맞춤형 관심 분야 문제 제공

학습자는 맞춤형 변형문제 제공 기능을 통해 관심있는 분야에 대한 문제를 제공받을 수 있다. 관심분야는 학습자가 사전에 회원가입 시 혹은 회원가입 이후에 등록하며, 문제에 사용되는 알고리즘 및 기능은 학습자가 자유롭게 선택

가능하고 학습자의 학습이력을 기반으로 프로그램이 추천해 주기도 한다. 정형화된 문제가 아닌 개인 맞춤형 문제를 통해 학습자는 흥미롭게 학습할 수 있게 된다.

2.3 User Characteristics

2.3.1 시스템 관리자

기술적 숙련도: 웹 서버에 대한 지식, 네트워크에 대한 지식, 데이터베이스에 대한 지식, LLM 에 대한 지식

수행 기능: 웹 사이트 관리, LLM 프롬프트 엔지니어링

2.3.2 학습자

기술적 숙련도: 기본적인 프로그래밍 언어에 대한 지식과 사용 경험, 간단한 웹 사이트와 상호작용할 수 있는 능력

수행 기능: 회원가입 및 로그인, 문제 풀이 및 제출, 단계별 힌트 요청, 변형 문제 요청, 맞춤형 문제 요청

2.4 constraints

- 사용자는 문제를 풀고 코드를 작성, 제출하기 위해 모바일 폰이 아닌 PC 나 노트북을 이용하여야 함.
- 사용자는 자바 스크립트가 활성화된 인터넷 브라우저를 사용해야 함.
- 사용자는 해당 서비스를 이용하기 위해 반드시 로그인을 하여야 함.
- 사용자는 코딩의 기본 개념을 익힌 후 문제를 풀 수 있는 능력이 있는 학습자로 상정한다.
- LLM 은 chatgpt-4o 를 사용한다.
- LLM 의 새로운 버전, 새로운 LLM 이 나왔을 때 시스템의 업그레이드를 고려해야 한다.
- 시스템의 개발 비용을 고려해야 한다.
- 시스템의 유지 관리 비용을 고려해야 한다.
- 사용자는 LLM 생성 기능을 하루에 최대 20 번까지 이용할 수 있다.

2.5. Assumptions and Dependencies

- 이 시스템은 웹 브라우저(Chrome, Firefox, Safari, Edge)에서 접근할 수 있으며 구형 브라우저나 잘 알려지지 않은 브라우저에서는 일부 기능이 제대로 작동하지 않을 수 있다.

- 사용자는 시스템의 UI를 통해 원활하게 인터페이스를 사용할 수 있다고 가정한다.
- 본 문서에 언급된 최소 사양을 충족하지 않는 장치에서는 시스템이 원활하게 작동하지 않을 수 있다.

3

Specific Requirements

3.1 External interfaces

3.1.1. User interface

항목 이름	사용자 등록 인터페이스
목적	사용자의 정보를 시스템에 등록한다.
입력 소스/출력 대상	사용자/데이터베이스 서버
범위/정확도/허용 오차	없음
측정 단위	등록 사용자 수
제공 타이밍	회원가입 정보 입력 완료 시
다른 입력/출력과의 관계	로그인 시 입력된 정보와 대조
데이터 형식 및 구성	Query

Table 3.1

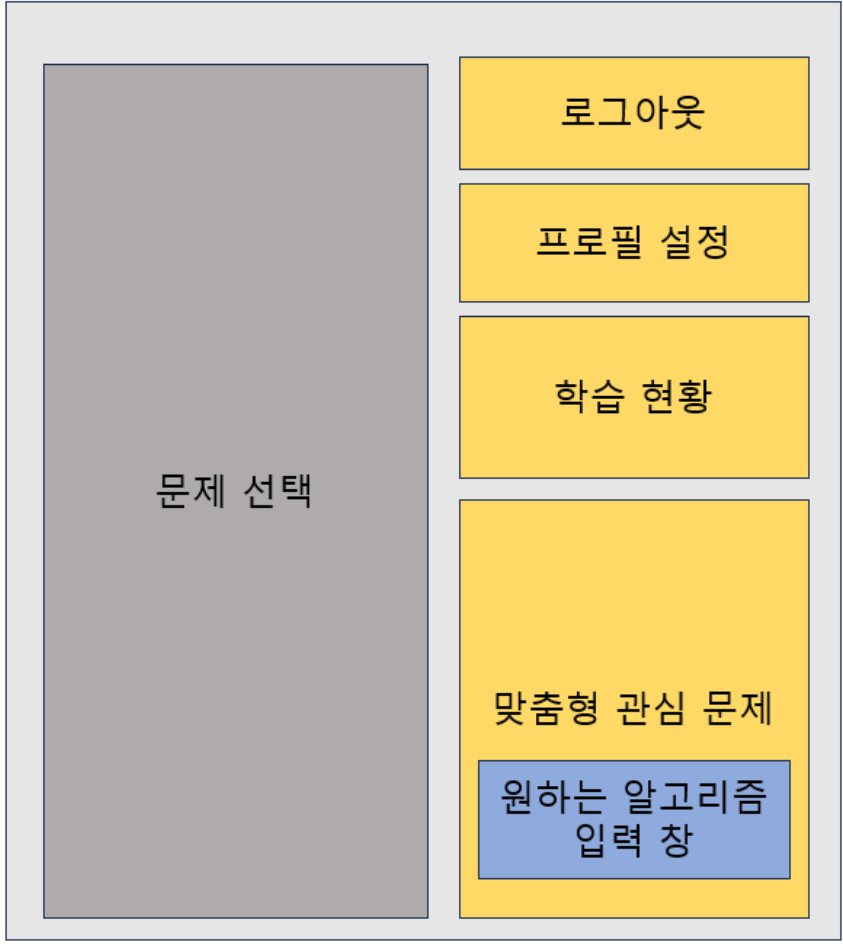
항목 이름	시스템 사용자에게 제공하는 메인 화면 인터페이스
목적	사용자에게 문제 선택, 맞춤형 문제 생성 및 해당 생성을 위한 알고리즘 입력 등의 기능을 제공하는 메인 화면을 제공한다.
입력 소스/출력 대상	호스트 서버/사용자 디스플레이
범위/정확도/허용 오차	없음
측정 단위	사용자 화면
제공 타이밍	메인 화면 진입 시
다른 입력/출력과 의 관계	사용자의 기능 요청에 대응되는 출력
데이터 형식 및 구성	텍스트
화면 형식 및 구성	 <p>The diagram illustrates the main screen layout. On the left is a large gray rectangular area labeled '문제 선택' (Problem Selection). On the right is a vertical column of yellow rectangular buttons. From top to bottom, these buttons are labeled: '로그아웃' (Logout), '프로필 설정' (Profile Settings), '학습 현황' (Learning Status), and '맞춤형 관심 문제' (Customized Interest Problem). Inside the '맞춤형 관심 문제' button, there is a smaller blue rectangular box labeled '원하는 알고리즘 입력 창' (Desired Algorithm Input Window).</p>

Table 3.2


항목 이름	시스템 사용자에게 제공하는 문제 풀이 인터페이스
목적	시스템 사용자에게 문제 풀이 인터페이스를 제공한다.
입력 소스/출력 대상	호스트 서버/사용자 디스플레이
범위/정확도/허용 오차	<ul style="list-style-type: none"> • 범위: 화면 상 코드 작성 부분, 기능 버튼 • 정확도: 클라이언트의 코드 작성의 정확도 및 기능 버튼 클릭에 대한 요청 처리의 정확도 • 허용 오차: 해당 없음
측정 단위	사용자 화면
타이밍	사용자의 작성 또는 요청 직후
다른 입력/출력과의 관계	사용자의 기능 요청에 대응되는 출력
데이터 형식 및 구성	텍스트
화면 형식 및 구성	

Table 3.3


항목 이름	단계별 힌트 제공 인터페이스
목적	문제 풀이 시 주요 부분 강조, 강조 부분에 대한 자세한 설명, 해당 문제에 어떤 알고리즘을 도입하는 것이 왜 필요한 지에 대한 논리적 설명을 제공한다.
입력 소스/출력 대상	힌트 버튼 클릭/단계별 문제 해설
범위/정확도/허용 오차	<ul style="list-style-type: none"> • 범위: 문제에 사용해야 할 알고리즘 및 해당 알고리즘이 필요한 이유를 자연어로 설명한다. • 정확도: 문제와 관련된 핵심 알고리즘 및 핵심 논리 도출 정확도 • 허용 오차: 도출된 핵심 알고리즘 및 주요 논리는 같아야 하고, 문장 내 조사는 다를 수 있다.
측정 단위	힌트 단계의 수
타이밍	힌트 버튼 클릭 시
다른 입력/출력과의 관계	출력된 문제에 강조 표시
데이터 형식 및 구성	텍스트
화면 형식 및 구성	

Table 3.4

3.1.2 Hardware Interface

항목 이름	마우스 및 키보드를 통한 입력 처리
목적	사용자가 마우스 클릭 및 키보드 입력을 통하여 명령을 원활히 실행할 수 있다.
입력 소스/출력 대상	사용자/시스템 기능
범위/정확도/허용 오차	<ul style="list-style-type: none"> • 범위: 특정 버튼 및 텍스트 필드 • 정확도: 실시간 입력 처리 정확도 • 허용 오차: 0.001 초 내 반응
측정 단위	마우스 클릭, 키보드 입력
제공 타이밍	입력 직후
다른 입력/출력과 관계	문제 제출, 문제 해설, 문제 변형 기능 등 시스템 내 입출력 기능들과 연동
데이터 형식 및 구성	해당 없음

Table 3.5

3.1.3 Software Interface

항목 이름	웹 사이트
목적	웹 사이트 상에서 서비스를 제공한다.
입력 소스/출력 대상	없음
범위/정확도/허용 오차	Chrome, Safari, Edge, FireFox
측정 단위	없음
타이밍	상시
다른 입력/출력과 관계	없음
데이터 형식 및 구성	해당 없음

Table 3.6

3.1.4 Communication Interface

항목 이름	호스트 서버 - 클라이언트 간 통신
목적	클라이언트가 호스트 서버에 접속하고, 힌트 및 생성 기능을 요청하며, 호스트 서버는 클라이언트가 제출한 코드 실행 결과를 클라이언트에게 전달할 수 있다.
입력 소스/출력 대상	호스트 서버/클라이언트
범위/정확도/허용 오차	해당 없음
측정 단위	패킷
타이밍	클라이언트 요청 직후
다른 입력/출력과의 관계	해당 없음
데이터 형식 및 구성	JSON

Table 3.7

항목 이름	호스트 서버 - LLM 간 통신
목적	사용자의 LLM 서비스 요청을 처리한다.
입력 소스/출력 대상	사용자가 힌트를 요청한 문제/해당 힌트 사용자가 요청한 변형 문제/해당 문제
범위/정확도/허용 오차	<ul style="list-style-type: none"> • 범위: 요청된 힌트와 변형 문제 • 정확도: 목표에 맞는 힌트 및 문제 제공 • 허용 오차: 최대 5 초 이내 응답 제공
측정 단위	처리 시간(초)
타이밍	사용자 요청 즉시 서비스 서버에서 LLM 서버로 요청
다른 입력/출력과의 관계	사용자의 힌트 버튼 클릭 입력, 변형 문제 요청 버튼 클릭 입력에 대해 반응
데이터 형식 및 구성	JSON

Table 3.8

3.2. Functional Requirements

3.2.1. Use Case

Use case name	회원가입
Actor	등록되지 않은 사용자, 데이터베이스
Description	등록되지 않은 사용자가 학습을 위해 처음 시스템에 들어왔을 때, 시스템의 기능을 사용하기 위해 자신의 개인정보 및 관심 분야를 등록한다.
Precondition	<ul style="list-style-type: none"> • 아직 시스템에 등록되지 않은 사용자이어야 한다. • 형식에 맞는 아이디 및 비밀번호를 설정해야 한다. • 다른 사용자와 중복되는 정보를 사용할 수 없다.
Postcondition	입력된 개인정보 및 관심 분야를 데이터베이스에 저장한다.
Normal flow	<ol style="list-style-type: none"> 1. 사용자가 시스템에 들어오면 로그인 페이지가 나타난다. 2. 아직 시스템에 사용자 등록하지 않은 사용자의 경우, 회원가입 버튼을 누른다. 3. 회원가입 페이지로 들어와 자신의 이름, 핸드폰 번호, 이메일, 아이디, 비밀번호, 그리고 관심 분야를 선택하여 작성한다. 4. 입력한 핸드폰 번호로 본인인증을 한다. 5. 작성을 다 끝낸 후, 완료 버튼을 눌러 회원가입 절차를 마치고 다시 로그인 페이지로 돌아온다.
Exception flow	<ol style="list-style-type: none"> 1. 이미 등록된 사용자의 경우: “이미 등록되어 있는 사용자입니다. 로그인 페이지로 돌아갑니다.”라는 메시지를 사용자에게 표시한 후, 회원가입 페이지가 아닌 로그인 페이지에 머문다. 2. 형식에 맞지 않는 정보를 입력했을 경우: <ol style="list-style-type: none"> 1) “형식에 맞지 않습니다. 다시 입력해주세요.” 라는 메시지를 출력한다. 2) 사용자에게 다시 입력을 받기 위해 형식에 맞지 않는 곳으로 이동한다.

Table 3.9

Use case name	로그인/로그아웃
Actor	등록된 사용자, 데이터베이스
Description	<ul style="list-style-type: none"> 로그인은 시스템에 등록된 사용자가 학습을 위해 시스템에 들어오기 위한 프로세스이다. 로그아웃은 학습을 마친 사용자가 시스템에서 나가려고 하기 위한 프로세스이다.
Precondition	<ul style="list-style-type: none"> 로그인 <ul style="list-style-type: none"> 이미 회원가입을 마치고, 시스템에 등록된 사용자여야 한다. 회원가입시 등록한 아이디 및 비밀번호가 있어야 한다. 로그아웃 <ul style="list-style-type: none"> 로그인 되어 시스템을 사용하고 있던 사용자여야 한다.
Postcondition	사용자가 시스템에 들어오거나 나가게 된다.
Normal flow	<ul style="list-style-type: none"> 로그인: <ol style="list-style-type: none"> 사용자가 시스템에 들어오면 로그인 페이지가 나타난다. 회원가입시 등록한 아이디와 비밀번호를 입력한다. 정확한 정보 입력 시, 사용자는 시스템에 들어와 학습 기능들을 이용할 수 있다. 로그아웃: <ol style="list-style-type: none"> 사용자가 시스템을 종료하기 원하는 경우, 로그아웃 버튼을 누른다. 시스템에서 나가게 된다.
Exception flow	<p>로그인 시, 입력한 정보가 올바르지 않은 경우:</p> <p>"입력한 정보가 올바르지 않습니다. 다시 한번 확인하여 주세요."라는 메시지를 표시한다.</p>

Tabel 3.10

Use case name	단계별 힌트 제공
Actor	등록된 사용자, 데이터베이스, LLM
Description	학습자가 문제 풀이 중 어떻게 접근해야 할지 모를 때, 단계별 힌트를 제공하여 문제 해결에 필요한 자료구조와 알고리즘을 추론할 수 있도록 한다.
Precondition	학습자는 시스템에 로그인 되어있는 상태이고, 문제 풀이 중이어야 한다.
Postcondition	단계별 힌트가 성공적으로 제공되어, 사용자는 문제 해결을 위한 추가 정보를 얻을 수 있다.
Normal flow	<ol style="list-style-type: none"> 1. 학습자가 문제 풀이 중 힌트 요청 버튼을 누른다. 2. 1 단계 힌트로 학습자에게 문제의 주요 키워드 및 문장에 대한 강조 표시를 제공한다. 3. 학습자가 추가 힌트를 요청한다. 4. 2 단계 힌트로 강조된 키워드 및 문장의 의미와 개념을 상세히 설명한다. 5. 학습자가 마지막 힌트를 요청한다. 6. 3 단계 힌트로 문제 해결에 필요한 자료구조와 알고리즘을 제공한다.
Exception flow	<p>3 단계 힌트 제공 이후에도 사용자가 힌트를 요청하는 경우:</p> <ol style="list-style-type: none"> 1. "더 이상 제공될 수 있는 힌트가 없습니다. 관련된 내용을 다시 학습하세요." 라는 메시지를 학습자에게 표시한다. 2. 관련된 자료구조 및 알고리즘에 관련된 강의 링크를 제공한다

Table 3.11

Use case name	변형 문제 제공
Actor	등록된 사용자, 데이터베이스, LLM
Description	학습자가 풀고 있는 문제를 완료 후, 해당 문제에 대한 변형문제 생성을 요청하면 시스템이 변형 문제를 제공한다.
Precondition	학습자는 현재 풀고 있는 문제 풀이를 완료해야 하며, 시스템은 문제를 변형하여 새로운 문제를 생성할 수 있어야 한다.
Postcondition	학습자는 변형 문제를 제공받고, 이를 통해 방금 해결한 문제에서 부족하다고 느끼는 부분을 다시 학습할 수 있다.
Normal flow	<ol style="list-style-type: none"> 1. 학습자가 변형 문제 제공 버튼을 누른다. 2. 시스템이 현재 문제를 기반으로 한 변형문제를 제공한다. 3. 학습자는 생성된 변형 문제를 화면에서 확인하고 풀이를 시작할 수 있다.
Exception flow	<ul style="list-style-type: none"> • 학습자가 문제 풀이를 완료하지 않고 버튼을 누르는 경우: "아직 문제 풀이를 완료하지 않았습니다. 문제 풀이를 완료한 후, 요청하여 주시기 바랍니다."라는 메시지를 학습자에게 표시한다. • 변형 문제를 5 번 제공했음에도 버튼을 누르는 경우: "더 이상 변형할 수 없습니다. 다른 문제를 시도해보세요."라는 메시지를 학습자에게 표시한다.

Table 3.12

Use case name	맞춤형 문제 제공 - 관심 분야 관련 문제
Actor	등록된 사용자, 데이터베이스, LLM
Description	학습자가 학습한 이력을 바탕으로 알고리즘을 추천해주고, 회원가입 시 등록한 관심사를 기반으로 하여 맞춤형 문제를 생성한다.
Precondition	학습자의 관심 분야가 학습자의 개인정보에 등록되어 있어야 한다.
Postcondition	학습자는 자신의 관심 분야를 반영한 맞춤형 변형 문제를 제공받고, 이를 통해 흥미 있고 효과적으로 학습할 수 있다.
Normal flow	<ol style="list-style-type: none"> 1. 학습자가 관심 분야 기반 맞춤형 문제 제공 기능을 요청한다. 2. 학습자가 학습한 이력을 기반으로 알고리즘을 추천한다. 3. 추가하고 싶은 알고리즘이 있으면 직접 입력하여 추가한다. 4. 이 알고리즘과 등록 되어있는 관심 분야를 기반으로 맞춤형 문제를 생성하여 제공한다. 5. 학습자는 생성된 맞춤형 문제를 화면에서 확인하고 풀이를 시작할 수 있다.
Exception flow	<p>학습자의 관심 분야가 등록되어 있지 않은 경우:</p> <ol style="list-style-type: none"> 1. "학습자의 관심 분야가 등록되어 있지 않습니다. 먼저 관심 분야를 선택하여 등록해주시기 바랍니다."라는 메시지를 학습자에게 표시한다. 2. 시스템은 학습자에게 선호하는 주제 및 관심 분야를 입력하거나 선택하도록 요청한다. 3. 학습자가 선택하여 제출한 관심 분야를 학습자의 개인정보에 등록한다.

Table 3.13

3.2.2 Use Case Diagram

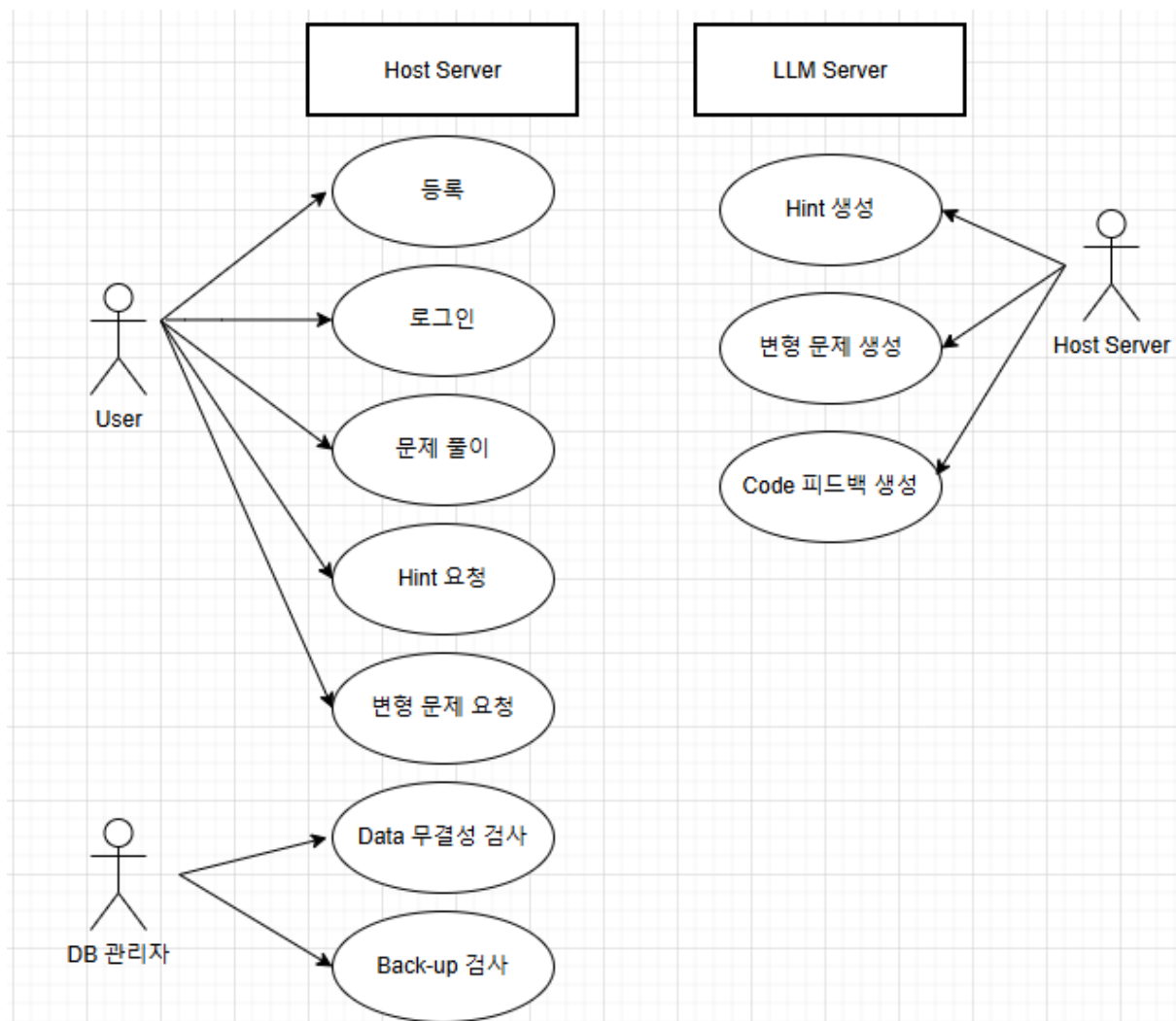


Figure 3.1: Use Case Diagram

3.2.3. Data Flow Diagram

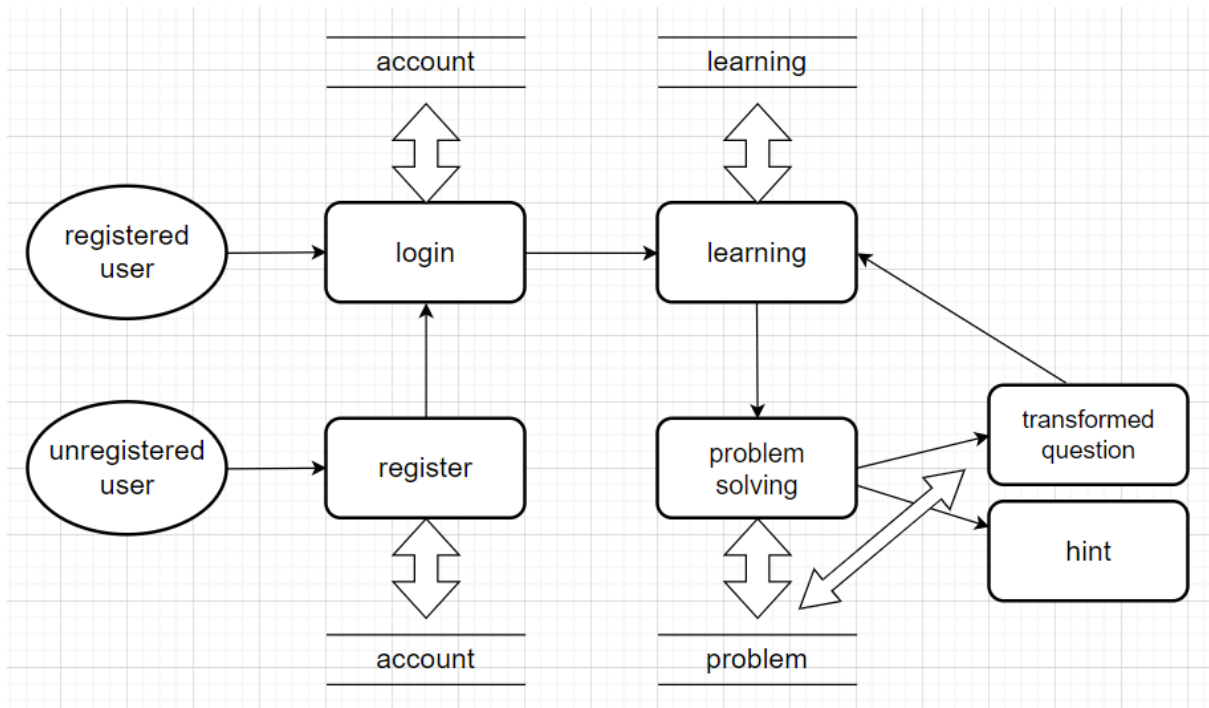


Figure 3.2: Data Flow Diagram

3.3 Performance Requirement

3.3.1 Static Numerical Requirement

시스템은 최대 1 만명의 사용자의 정보를 저장해야 한다.

시스템은 최대 3 만개의 알고리즘 문제를 제공해야 한다.

시스템은 최대 1000 명의 사용자의 동시 접속을 지원하고 요청을 처리해야 한다.

서버는 시스템의 안정적인 운영을 위해서 2.0GHz, 4 코어 이상의 CPU, 32GB 이상의 메모리, 5Gbps 이상의 대역폭, 용량 100GB 이상의 데이터베이스가 필요하다.

사용자는 시스템을 안정적으로 이용하기 위해서 웹 브라우저를 실행할 수 있는 환경, 1.0GHz, 1 코어 이상의 CPU, 4GB 이상의 메모리, 1Mbps 이상의 대역폭이 필요하다.

3.3.2 Dynamic Numerical Requirement

사용자의 회원가입, 로그인, 로그아웃 요청을 1 초 이내로 처리해야 한다.

사용자의 문제 조회 요청을 5 초 이내로 처리해야 한다.

사용자의 문제 풀이 제출 및 기록 저장을 5 초 이내로 처리해야 한다.

사용자가 제출한 소스코드의 채점을 1 분 이내로 처리해야 한다.

사용자의 힌트 제공 요청을 10 초 이내로 처리해야 한다.

사용자의 변형 문제 제공 요청을 20 초 이내로 처리해야 한다.

사용자의 코드 리뷰 및 최적화 요청을 10 초 이내로 처리해야 한다.

시스템은 초당 최대 500 개의 트랜잭션을 처리해야 한다.

시스템은 하루에 24 시간동안 항상 이용 가능해야 한다.

3.4 Logical database requirements

데이터 베이스는 PostgreSQL 을 사용한다. 데이터 베이스에는 각 사용자의 정보, 학습 정보, 문제 정보, 알고리즘 정보가 저장되어야 한다. 사용자 데이터는 사용자 ID, 비밀번호, 개인정보(핸드폰 번호, 이메일), 학습 이력, 관심 분야를 포함한다. 학습 데이터는 사용자 ID, 사용자별 학습 정보, 제출 코드, 제출 결과, 학습 결과, 제공된 피드백을 포함한다. 문제 데이터는 문제 ID, 제목, 문제 설명, 정답, 관련 알고리즘, 키워드를 포함한다. 사용자는 개인 정보를 열람 및 수정이 가능하고, 제공받았던 피드백, 문제풀이 이력을 확인하여 학습에 이용 가능하다.

사용빈도 높음: 문제 데이터, 알고리즘 데이터

사용자가 문제를 풀거나 피드백을 받을 때 마다 사용

사용빈도 낮음: 사용자 개인정보, 학습 결과, 학습 이력, 제공된 피드백

학습이 완료된 후 업데이트되거나 사용자가 원할 때 열람한다.

일반 사용자는 다른 사용자의 사용자 데이터, 학습 데이터를 제외한 모든 정보에 접근 가능하다. 다른 사용자의 학습 데이터는 학습 성취도를 비교하기 위해 일부 제공될 수 있다. 개인정보는 편집 가능하지만 대부분의 데이터는 읽기만 가능하다.

관리자는 모든 데이터에 접근 가능하며 데이터를 직접 편집할 수 있는 권한이 있다.

3.5 Design Constraints

3.5.1 Frontend Standard

언어 및 프레임워크

- React 를 사용하여 javascript 기반으로 UI 를 구축한다.
- Javascript 버전 정보: ECMAScript 2024 (ES2024)
- React 버전 정보: React 18

코딩 규칙

코드 스타일

- 본 시스템의 모든 Javascript 코드는 Airbnb JavaScript Style Guide 를 준수하여 작성한다.
- ESLint 와 Prettier 를 사용해 코드 스타일을 강제하고 자동 포매팅을 수행한다.

3.5.2 Backend Standard

언어 및 프레임워크

- 백엔드 서버는 Python 기반의 Django 프레임워크를 사용한다.
- Python 버전 정보: Python 3.12
- Django 버전 정보: Django 5.0

코딩 규칙

코드 스타일

- 본 시스템의 모든 Python 코드는 PEP 8 스타일 가이드를 준수하여 작성한다.
- PEP8: <https://peps.python.org/pep-0008/>
- 코드 스타일의 일관성 유지를 위해 Black 및 Flake8 도구를 사용하여 포매팅 및 스타일 검사를 자동화한다.

타입 힌트

- 본 시스템의 모든 Python 코드는 PEP 484 를 준수하여 작성한다.
- PEP484: <https://peps.python.org/pep-0484/>
- 타입 오류를 빠르게 찾기 위해 mypy 도구를 사용하여 정적 타입 검사를 수행한다.

Docstring

- 본 시스템의 모든 Python 코드는 PEP 257 을 준수하여 작성한다.
- PEP257: <https://peps.python.org/pep-0257/>

3.5.3 Database Standard

- 백업 및 복구를 제외한 DB 와의 모든 상호작용은 Django 를 통해 이루어진다.
- 백업 및 복구 작업은 DBMS 에 직접 접근해서 수행한다.

데이터베이스 관리 시스템(DBMS)

- 사용 DBMS: PostgreSQL
- DBMS 버전 정보: PostgreSQL 17

3.5.4 GPT Model Integration

- 문제 생성 기능은 GPT API 와 연동된다.
- 응답 속도가 교육 플랫폼의 실시간 요구를 만족할 수 있어야 하며, 응답 시간이 길어질 때 타임아웃 및 오류 처리 로직을 반드시 포함해야 한다.

3.5.5 Security and Privacy Protection

- 모든 데이터 전송은 HTTPS 을 통해 암호화되어야 하며, 사용자 데이터는 데이터베이스에서 암호화된 상태로 저장되어야 한다.
- 사용자가 생성한 문제 데이터는 보안 요건을 준수하여 외부 시스템과 연동될 경우에도 개인 정보가 노출되지 않도록 주의해야 한다.

3.5.6 Performance

- 문제 생성 요청은 평균 2 초 이내에 응답해야 하며, 최대 5 초 이상 지연되지 않도록 시스템을 설계해야 한다.
- GPT API 호출에 따른 응답 지연을 최소화하기 위해 캐싱 및 재사용 가능한 문제 목록을 생성하고, 사용자가 문제 생성 요청 시 캐시된 데이터를 활용하도록 한다.

3.6 Software system attributes

3.6.1 Reliability

데이터베이스에 저장되는 모든 고객의 학습 데이터는 무결성을 유지하여야 하며, 하루 단위로 백업되어야 한다.

모든 서비스가 웹사이트에서 충돌 없이 동작하여야 하며, 오류가 발생하였을 시 자동으로 복구되어야 한다.

오류로부터 복구되는 시간은 오류의 종류에 따라 최대 1~5 분으로 한다.

서비스 배포 시 자동화된 정기적 Back-to-back test 를 수행하여 모든 기능이 정상적으로 동작함을 상시로 확인하여야 한다.

시스템 하드웨어의 사용량을 모니터링하고, 일정 사용량을 넘지 않도록 서비스 사용 부하를 조절하여야 한다.

3.6.2. Availability

사용자가 필요할 때 항상 접근 가능하고 안정적으로 작동함으로써, 본 시스템의 학습자가 언제든지 학습할 수 있고 지속할 수 있도록 시스템은 24 시간 사용 가능해야 하며, 장애가 발생하였을 경우에도 그 복구시간을 5 분 이내로 하여야 한다. 일반적인 유지보수 및 업데이트가 필요한 경우에는, 최대한 시스템을 중단하지 않으면서 이루어질 수 있도록 해야 한다.

3.6.3 Security

모든 데이터의 전송은 HTTPS 를 통해 암호화되어야 한다.

사용자는 회원가입 시 핸드폰 번호 인증, 이메일 인증을 진행해야 한다. 또한 비밀번호 설정 시 최소 길이 및 영문, 숫자, 특수문자 포함 조건을 만족해야 한다.

사용자는 시스템의 서비스 이용을 위해서 로그인을 해야 한다.

사용자는 다른 사용자들의 정보를 열람할 수 없어야 한다.

사용자는 소스코드를 비공개로 설정한 다른 사용자가 제출한 소스코드를 열람할 수 없어야 한다.

사용자는 1 분에 소스코드를 최대 5 회까지만 제출할 수 있어야 한다.

사용자는 시스템 데이터베이스에 직접적으로 접근할 수 없어야 한다.

3.6.4 Maintainability

데이터 table 간 고유키는 한 개만 존재해야 하며, 고유키들은 외래 키로 연결되어 있어야 한다. 중복되거나 상이한 데이터가 존재해서는 안 되며, 데이터는 3NF 을 유지하여 모듈성을 높인다.

문제 풀이, 피드백 제공, 키워드 제공 등의 주요기능은 별도의 소프트웨어로 만들어 관리하고 이들의 기능이 결합되지 않도록 한다.

각 소프트웨어 간의 인터페이스, 데이터 베이스와 소프트웨어 간의 인터페이스가 모두 정의되어 명세화 되어야 한다.

데이터 접근 방식과 소프트웨어에 사용되는 알고리즘은 $n\log(n)$ 미만의 시간 복잡도를 가져야 하고, scalability 를 만족해야 한다. 코드 구조는 객체지향을 통해 일반화되어야 하고, 코드는 subsystem 별로 별도의 파일로 관리되어야 한다.

3.6.5 Portability

1) 멀티플랫폼 지원을 위한 웹 기반 환경 구축

- 시스템은 웹 어플리케이션으로 구축하여, 브라우저만 있으면 사용 가능하게 하여 플랫폼 독립성을 높인다.

2) 모듈화 및 API 기반 설계

- 각 기능들을 모듈화 해 관리하고, 다른 기능들과 API 를 이용해 상호작용할 수 있도록 한다.

3) 플랫폼 독립적인 개발 언어와 프레임워크 사용

- 백엔드와 프론트엔드 개발에서 각각 플랫폼 독립적인 언어와 프레임워크를 사용해 이식성을 높인다.

4) GPT 모델 인터페이스 표준화

- 시스템과 GPT 와의 인터페이스를 추상화하여 이후 모델 업그레이드에도 유연하게 대응할 수 있도록 한다.
- GPT 호출 부분을 인터페이스로 분리하면 다른 언어 모델로 교체하거나 추가 기능을 사용할 때 코드 변경을 최소화할 수 있다.

5) 확장 가능한 데이터베이스 구조

- 문제와 학습 데이터를 저장하는 데이터베이스 구조를 설계할 때 확장성을 고려해 설계한다.
- 문제의 난이도, 알고리즘 등을 메타데이터로 관리하여 다양한 학습 상황에 맞게 문제를 추천할 수 있도록 한다.

3.7 Organizing the specific requirements

이 절에서는 Unified Modeling Language(UML) 표기법을 사용하여 시스템을 모델로서 나타낸다. 시스템 모델은 시스템과 서브 시스템들 간의 관계를 나타낸다.

3.7.1 Context Model

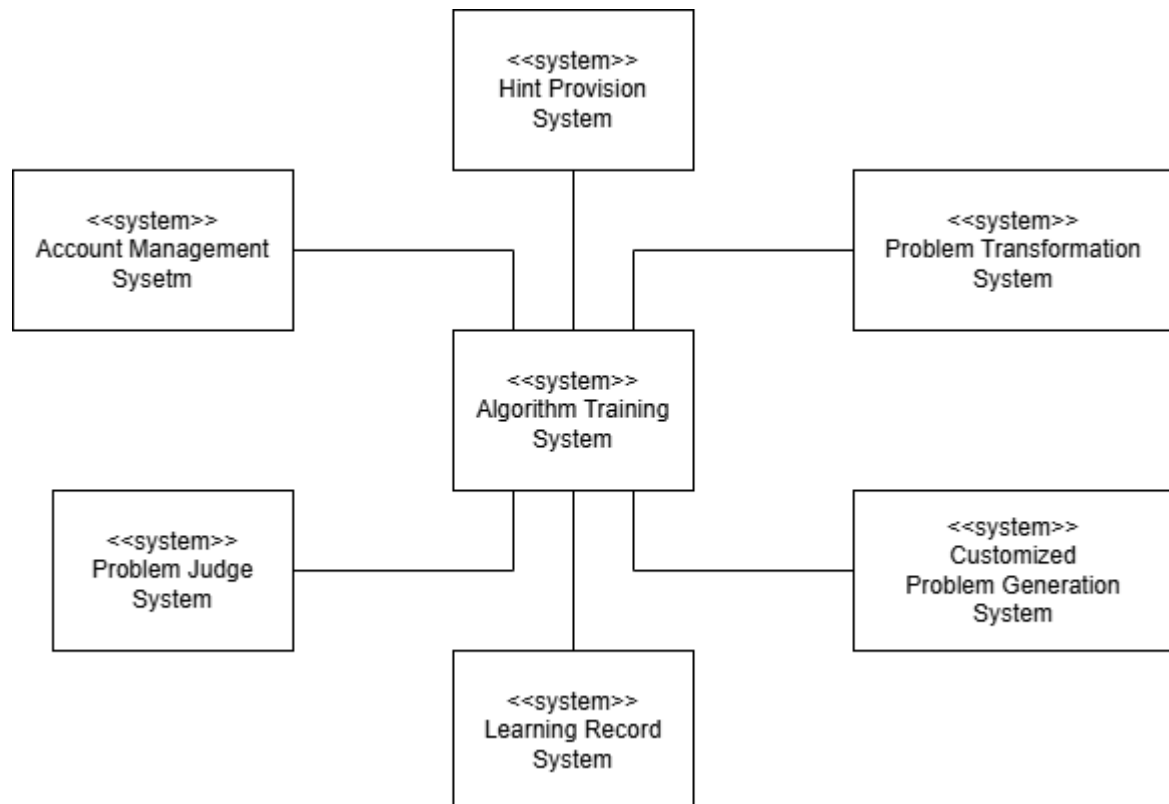


Figure 3.3: Context Model

3.7.2 Process Model

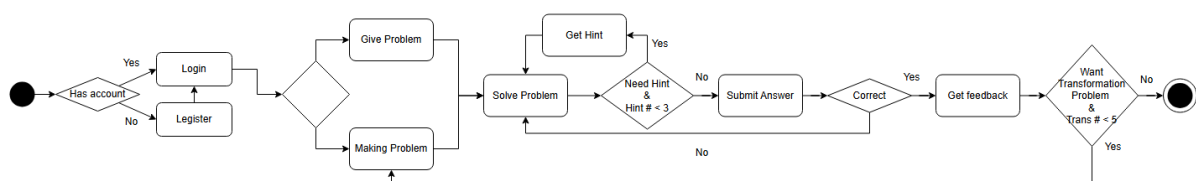


Figure 3.4: Process Model

3.8 System Architecture

이 절에서는 시스템 아키텍처에 대한 high-level 개요를 제시한다. 전체 시스템을 구성하고 있는 구성요소와 그 서브 시스템들을 개략적으로 나타내고 그들 간의 통신을 명시한다.

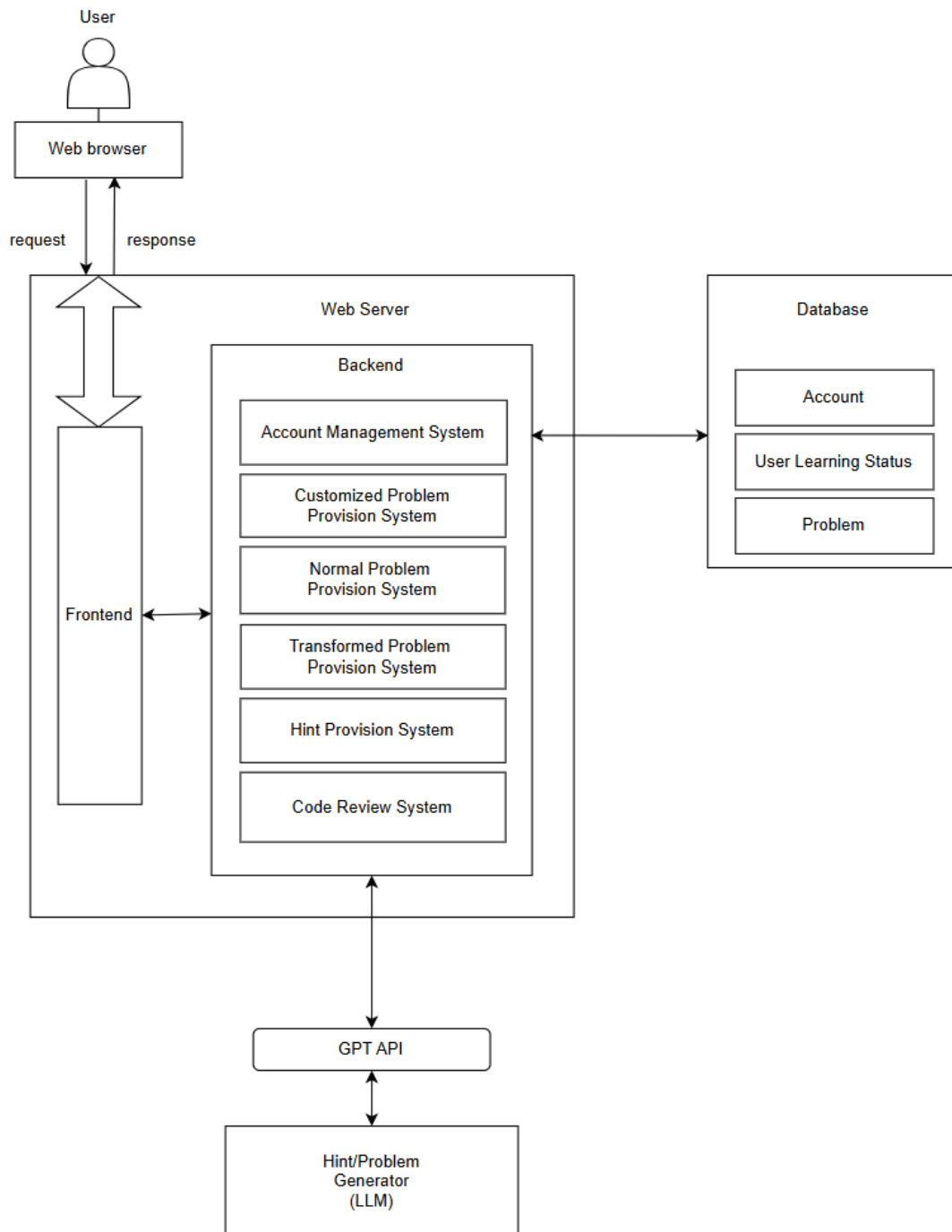


Figure 3.5: System Architecture

4

Appendix

4.1 Document History

Document History			
Date	Edit	Description	Writer
10-25	Add contents	Add Functional Requirements	All
11-1	Add contents	Add Overall description & Introduction	
11-4	Add contents	Add Diagram	
11-9	Organizing	Documentation	