# **[2023-2 Software Engineering]**

소프트웨어 개발 계획서(SDP)

**AjouCodeXpert**  
아주대학교 전공생을 위한   
프로그래밍 과제 채점 서비스

제출일: 2023년 10월 27일

팀 명: CodeMates

팀 원: 공명규, 박종현, 정은재, 최동훈, 허성철

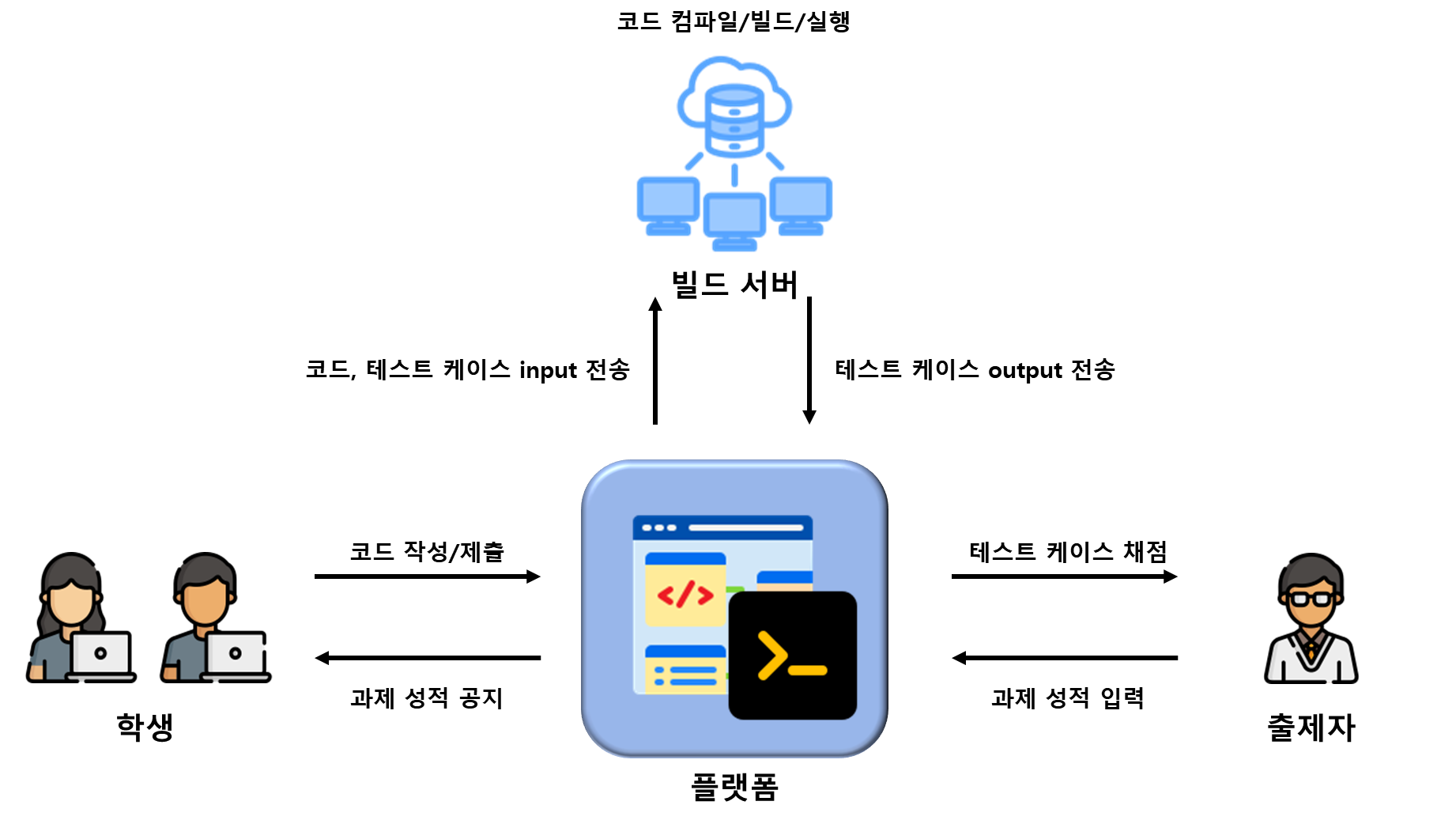
1. 개요

1.1. 프로젝트 개요

본 프로젝트는 아주대 소프트웨어학과를 전공하는 신입생과 복수전공⋅부전공하는 재학생 및 편입생들(이하 “학생”)을 대상으로 실시하는 기초 프로그래밍 강의에서 활용 가능한 코드 채점 서비스(이하 “본 서비스”) 구축을 목표로 한다.

본 서비스를 통해 학생들은 소속된 반에 가입하여, 해당 강좌에서 출제되는 문제를 확인하고 답안 코드를 작성하여 제출할 수 있다. 해당 강좌의 교수나 조교(이하 “출제자”)는 자신의 반에 소속된 학생들이 제출한 코드를 확인할 수 있다. 또한 서비스가 제공하는 실행 결과 확인 및 자동 채점 기능을 통해 과제 점수를 쉽게 부여하고, 부여된 점수 및 코드에 대한 피드백을 작성할 수 있다. 또한 ChatGPT API를 통해 학생들의 코드에서 문제점을 분석하고, 학생들이 코드 개선 사항에 대한 설명을 제공받을 수 있다.

본 서비스는 개발 환경을 웹으로 제공하여 기초 프로그래밍이나 알고리즘을 학습하는 학생들이 OS 환경이나 프로그래밍 언어 버전에 따라 마주할 수 있는 제약 사항을 줄여주며, 출제자의 경우 자동 채점 기능을 활용하여 더 편리하고 정확하게 과제 점수를 부여할 수 있도록 도와준다.

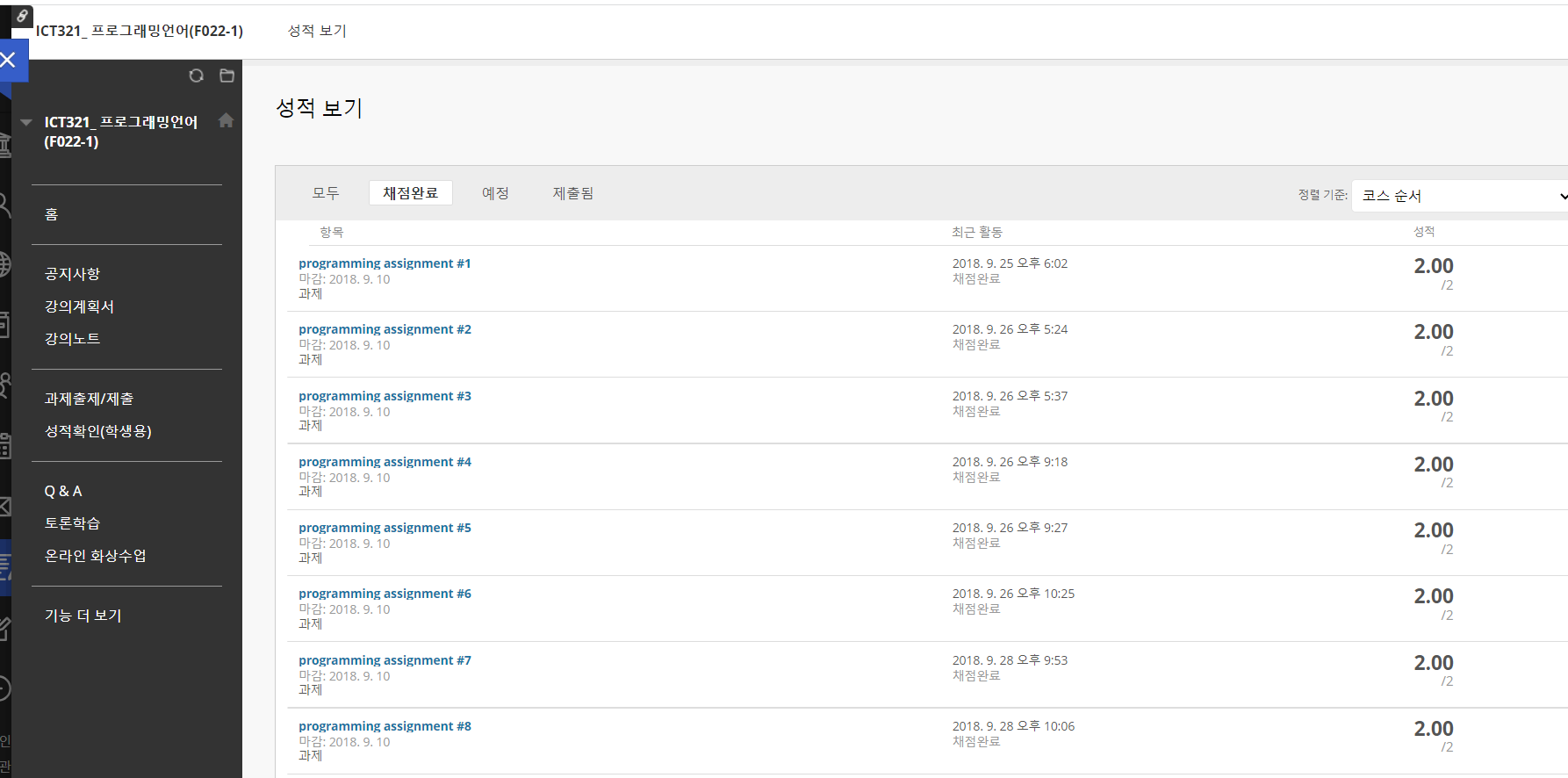
[그림 1] 시스템 기본 개념도

1.2. 주제선정 배경 및 차별성

1.2.1. 개발 배경

아주대학교 소프트웨어학과에서는 기초 프로그래밍 과목으로 ‘컴퓨터 프로그래밍 및 실습’ 강좌를 실시하고 있으며, 매 학기마다 5~7개의 강의가 개설되고 있다. 해당 과목의 수강생은 소프트웨어학과 및 인공지능학과 신입생, 전과 및 복수⋅부전공 학생, 외국인 학생 등 대부분 프로그래밍을 처음 배우는 학생들로 구성된다.

과목의 특성 상 매주 실습을 통한 과제가 출제되고 있으며, 다른 과목군과 마찬가지로 아주대에서 채택한 온라인 학습관리 시스템인 Blackboard Learn을 통해 학생들의 과제 수행과 교수자의 성적 부여가 이루어진다. 그러나 프로그래밍을 처음 배우는 학생들 중, 미숙한 개발환경 프로그램 사용과 랩탑에 따라 각기 다른 개발환경 설정 방식으로 인해 수업을 제대로 따라가기 어려워 하는 경우를 다소 확인할 수 있었다. 과제를 출제하고 채점하는 교수자의 입장에서도 학생들의 코드를 모두 실행하거나 확인 하는 작업으로 인해 많은 시간이 소요되거나 오채점을 범하는 상황이 발생하기도 한다. 따라서 이를 해결하기 위해 본 서비스를 기획하게 되었다.



[그림 2] 기존 Blackboard Learn을 통한 채점 시스템

또한 학생들의 과제에 대한 피드백을 제공하기 위해 ChatGPT를 활용하여 프로그래밍 문제에서의 에러 코드 보완 등의 기능을 제공하여 초보 개발자의 문제 해결 능력 향상에 도움을 제공하고자 한다.

1.2.2. 기존 시스템의 문제점 및 개선사항

본 프로젝트에서는 기초 프로그래밍 과목에 한정하여, 아주대학교에서 기존에 사용해오던 학습관리 시스템인 Blackboard Learn의 한계와 이를 보완할 새로운 시스템을 구축하고자 한다. 해당 시스템을 통해 이루어지는 프로그래밍 과목군의 과제 및 실습 과정에서 확인할 수 있었던 문제점들은 다음과 같다.

1. **과제 제출**

기존 시스템의 경우 아주BB를 통해 자신이 작성한 코드나 실행결과를 캡쳐한 후 업로드하는 방식으로 과제를 제출한다. 또한 Blackboard Learn에서는 텍스트 및 이미지를 포함한 일부 파일의 미리보기는 지원하지만, 프로그래밍 언어로 구성된 파일(\*.c, \*.py, \*.java 등)의 경우 자신이 제출한 코드를 즉각 확인하기 어렵다는 단점이 있다. 때문에 학생들이 과제 파일을 잘못 업로드하거나 제출한 소스파일에서 오타 등의 문제를 해결하지 못해 감점을 받는 경우가 발생할 수 있다.

1. **과제 채점**

2023년 상반기와 현재 컴퓨터프로그래밍및실습 과목에서 조교로 활동한 경험이 있거나 활동중인 인원을 대상으로 인터뷰를 실시한 결과, 대부분의 실습에서 채점의 편의를 위해 코드가 작성된 텍스트 파일과 코드 실행 결과 캡처본을 대조하여 채점을 진행하는 것을 알 수 있었다. 이를 통해 학생들의 과제를 채점하고 평가하는 과정에서 현행 시스템이 갖는 문제점을 확인할 수 있었다.

1. **실습**

위와 마찬가지로 인터뷰를 통해 컴퓨터 프로그래밍및실습 과목의 실습 진행 방식에 대한 조사를 진행했다. 조사를 통해 모든 과목에서 Visual Studio라는 IDE를 위주로 실습이 진행되는 것을 확인할 수 있었다. 또한 Windows가 아닌 다른 OS(Mac 등)을 사용하는 학생들의 경우 자체적으로 다른 IDE를 찾아 사용법을 익혀야 했다. 때문에 Visual Studio를 사용하지 못하는 환경에서 학생들은 실습을 따라가기 어려워하는 문제점을 발견할 수 있었다.

1. **과제 피드백**

일반적으로 피드백 기능은 교수자가 학생이 제출한 과제의 점수에 대한 설명이나 감점 요인을 학생들에게 고지하는 용도로 사용되고 있다. 때문에 피드백이 대부분 점수에 대한 내용으로만 간략하게 구성된 경우가 많고, 이러한 피드백은 실질적으로 학생들의 코딩 실력을 향상시키기에는 부족함이 크다.

본 프로젝트에서 구현하고자하는 시스템은 다음과 같은 기능 및 서비스를 제공하여, 위와 같은 문제점들을 해결하고자 한다.

1. **웹 기반 개발 환경 제공**

학생들은 웹 브라우저를 통해 손쉽게 접근할 수 있는 개발 환경을 제공받으며, 실행한 코드를 즉시 제출할 수 있도록 도움을 받는다. 이로 인해 운영 체제나 개발 환경에 구애없이 코드 작성 및 제출이 가능하며, 이를 통해 학습자들은 더욱 편리하게 과제를 수행할 수 있다.

1. **테스트 케이스 기반 코드 채점**

과제 채점을 자동화하고 테스트 케이스를 기반으로 코드를 평가한다. 이를 통해 교수자들은 번거로운 수작업 채점을 피하고, 보다 정확하고 일관된 평가를 학생들에게 제공할 수 있다.

1. **LLM 기반 피드백 제공**

ChatGPT를 통해 학습자들에게 개선 사항과 함께 예시 답안을 제공하는 자동 피드백 시스템을 도입한다. 이를 통해 교수자의 피드백이 간략할 경우에도 학습자들은 올바른 방향으로 자신의 작업을 개선할 수 있다.

[문제점]

* 과제 제출 (개발환경 프로그램과 코드 제출 플랫폼이 다름으로 인해 잘 못 제출하는 경우)
* 과제 채점 (일일이 실행 + 확인, 오채점)
* 실습 (개발환경 프로그램 미숙지로 인해 따라가지 못함)
* 과제 피드백 (교수자의 피드백이 간략할 경우 제대로된 피드백 제공 어려움)

[개선 사항]

* 웹 기반 개발 환경 제공(실행한 코드를 즉시 제출 + OS와 무관하게 코드 작성 가능)
* 테스트 케이스 기반 코드 채점(번거로운 채점 X)
* LLM 기반 피드백 제공 (학생에게 개선 사항 및 예시 답안 제공)

1.3. 핵심 주요기능

1.3.1. 사용자 등록 및 관리

시스템을 이용할 사용자를 등록하고 계정을 발급한다. 발급된 사용자 계정은 ‘학생’과 ‘TA(조교)’ 로 구분되며, 역할에 따라 서비스를 이용할 수 있는 권한을 다르게 부여한다.

1.3.2 반 개설 및 관리

‘TA’ 사용자는 과목 명과 과목 코드를 입력하여 반을 개설할 수 있다. ‘TA’는 여러 개의 반을 개설할 수 있으며, 하나의 반에 여러 ‘TA’ 계정이 등록되어 해당 반을 관리할 수 있다. 반을 생성하거나 반에 등록된 ‘TA’는 자신이 속한 반에 ‘학생’들을 초대할 수 있다. 이후 ‘TA’는 자신이 관리하는 반에 소속된 ‘학생’들을 대상으로 과제를 생성하거나 제출된 과제에 대한 점수를 부여하는 등의 작업을 수행할 수 있다.

1.3.3 과제 등록 및 관리

**과제** : ‘TA’는 자신이 관리하는 반에 과제를 생성하고 해당 과제 내에 문제(이하 ‘문항’)들을 등록할 수 있다. 과제 생성시 ‘과제 명’과 ‘마감 기한’, 각 문항 별 ‘배점’ 정보들을 포함해야 하며 생성된 과제는 해당 반에 소속된 ‘학생’들에게 노출된다.

**문항** : 각 문항은 Input과 Output의 순서쌍으로 구성된 테스트케이스를 등록해야 하며 이를 기준으로 채점이 진행된다. 조교가 허용하는 프로그래밍 언어를 기준으로 학생들에게 문항이 출제되며 과제 출제시 ‘TA’는 문항별로 제한 및 요구 사항들을 추가로 등록할 수 있다.

1.3.4 과제물 시도 및 제출

**과제 확인** : 학생 자신이 속한 반에서 생성된 과제를 확인할 수 있으며 과제의 마감 기한, 각 과제에 속한 문항에 대한 내용, 허용 언어, 제약 사항을 확인할 수 있다.

**과제물 시도 및 제출** : 학생은 문항에 대한 답안을 작성하여 이를 제출할 수 있다. 제출 전에 학생은 자신의 코드가 정상적으로 동작하는지 실행해볼 수 있으며 ‘TA’에 의해 등록된 테스트 케이스와 비교해볼 수 있다. 만약 빌드 및 컴파일 에러 등으로 실행이 실패했을 경우에는 에러 메시지를 받아볼 수 있어야 한다. 학생은 여러 번 과제를 시도 및 제출할 수 있으며, 이전 제출물 중에서 과제 평가를 위한 최종 제출물을 선택할 수 있다. 또한 만일 특정 문항에 대해 이전 제출 이력이 있다면 다음 시도시 가장 최근에 제출한 코드를 불러올 수 있다.

1.3.5 과제의 채점

시스템은 과제 마감 기한이 지나면 학생들의 제출물을 일괄적으로 채점하여 점수를 산정한다. 시스템은 각각 정확도, 코드 유사도에 대한 검사를 진행하며 학생이 제출한 코드에 대한 전반적인 리뷰를 제공해주고, 조교는 이를 기반으로 과제 피드백을 작성하거나 최종 과제 점수를 산정할 수 있다.

**정확도 검사** : 시스템은 각 문항에 등록된 테스트케이스를 기반으로 정확도 검사를 실시한다. 각 문항들은 학생들이 제출한 시점에서 채점이 되며, 과제가 마감되는 순간 해당 과제에 포함된 문항들의 점수를 합산하여 정확도 점수를 계산한다.

**유사도 검사** : 시스템은 과제 마감 시점 이후 다른 제출물들과 비교하여 코드 유사도를 비교하고 이를 조교가 확인할 수 있도록 표시한다.

**코드 리뷰** : 시스템은 과제 마감 시점 이후 학생의 제출물에 대한 리뷰를 작성한다. 만약 학생의 코드에 에러가 발생했거나 테스트케이스에서 감점이 발생했다면 생성형 AI를 통해 코드의 문제점을 분석하고 개선 사항과 예시 답안을 작성해준다.

1.3.6 성적 및 리뷰 확인

TA는 각 학생의 제출물에 대한 시스템 채점 결과와 코드 리뷰를 토대로 해당 과제의 최종 점수를 산정하고 부여할 수 있다. 또한 학생의 코드와 자동 생성된 코드 리뷰를 토대로 제출물에 대한 코멘트를 추가할 수 있다. 성적 및 코멘트 등록을 마치면 학생에게 알림이 가고 학생은 과제 점수 및 각 문항별 코드 리뷰, 코멘트와 부여된 점수를 확인할 수 있다.

2. 관련 기술 현황

코딩 플랫폼, Online Judge 서비스들의 특징을 비교한다. 우리가 제작하려는 서비스는 기존에 존재하는 웹 서비스를 바탕으로 발전시켜 제작하는 것이기에 이미 시장에 출시되어 있는 관련 서비스의 특징에 대해 소개할 것이다.

아주대 자체의 Online Judge 서비스를 통해 학습자에게 추가적인 피드백을 자체적으로 제공하여 학습에 대한 능률을 끌어올릴 수 있다. 사용자가 제출한 코드는 OpenAI API를 사용하여 평가될 것이며 사용자에게 적절한 피드백을 제공할 수 있을 것이다.

문제 풀이 페이지에서 editor를 통한 코드 작성이 가능하다. 실행 버튼을 통해 작성 코드의 실행 결과를 얻을 수 있으며 채점을 통해 등록한 테스트 케이스에 대한 정답을 확인할 수 있다. 마지막으로 제출을 통해 최종적인 정답을 확인할 수 있으며 이 때 다양한 metric에 근거한 Feedback 서비스가 제공된다.

2.1. 백준 Online Judge

****

[그림 3] 백준 OJ

백준 Online Judge는 국내 최대 규모 알고리즘 사이트이다. 특징으로는 표준 입출력 기반 풀이를 작성하며 50가지 프로그래밍 언어를 지원한다. 단계별, 카테고리 별 분류된 문제를 제공하고 있으며 문제 풀이 성공과 실패 외에도 소요 시간과 메모리 사용량 등 다양한 평가 결과를 제공한다. 국내 알고리즘 대회 플랫폼으로도 많이 쓰이고 있다.

2.2. Programmers

****

[그림 4] Programmers

네이버, 카카오 등 여러 국내 기업의 코딩테스트 플랫폼으로 사용되고 있다. 입출력 기반의 백준과는 다르게 솔루션 함수 작성 기반 문제 풀이다. 12가지 프로그래밍 언어를 제공하고 있으며 테스트 케이스 채점 결과를 사용자에게 알려주어 가시성을 높이기도 하였다.

3. 개발 목표 및 내용

3.1. 시스템 구성 및 개발 범위

3.1.1. 사용자 시나리오

본 서비스의 주요 기능과 기능 간의 흐름을 3.1.1의 사용자 시나리오 구성을 기준으로 아래와 같이 정리하였다.

* 사용자 계정 관리

사용자(학생 또는 TA)은 서비스에 자신의 학적사항을 입력해서 회원가입을 할 수 있다. 학생은 서비스를 통해 과제를 수행하게 되며, TA는 반과 과제를 등록하고 학생의 제출물을 채점한다. 사용자들은 개인 페이지에서 필요한 알림을 수신할 수 있다.  
 사용자 계정의 권한을 통제하는 관리자 계정이 존재한다. 관리자는 사용자가 TA로 가입했을 경우 이를 승인하며, TA의 반 생성을 승인하며, 모든 사용자 정보를 확인 가능하다.

* TA가 효율적으로 채점할 수 있는 시스템

TA는 담당하는 반을 운영하기 위해 서비스에서 반을 개설하게 된다. 반 개설 이후, TA는 반 페이지에서 학생들을 반에 등록시키는 데 사용할 ‘초대 코드’를 확인할 수 있고, 이를 이용해 등록 요청을 보낸 학생들의 요청을 수락한다.  
 TA는 개설된 반에서 주차별로 과제를 등록할 수 있다. 과제는 학생이 수행해야 할 프로그래밍 문제들을 포함한다. TA는 과제 상세 페이지에서 과제에 대한 설명을 적어주고 마감일을 명시한 후 문제를 등록한다. 각 문제에 대해 TA는 학생들이 작성해야 할 코드에 대한 설명과, 학생의 코드가 input을 받았을 시 출력해야 할 output를 나타내는 테스트케이스를 제시해야 하며, 추가적으로 시스템에서 제출된 코드를 검토하는 데 사용할 지시사항과 키워드를 제시한다. 또한 문제마다 배점을 설정한다.  
 TA가 일일이 제출된 코드를 직접 빌드할 필요 없이, 시스템에서 학생들의 코드들을 직접 실행하여 통과한 테스트케이스만큼의 점수를 산정한다. 그리고 과제 마감일 이후 표절 방지를 위한 유사도 검사와 제시된 지시사항과 키워드를 참고하여 코드 리뷰를 생성하게 된다.

* 학생이 서비스 상에서 코드 작성, 실행 및 제출이 가능한 플랫폼

학생은 메인 페이지에서 초대 코드를 제출하여 소속된 반에 등록 요청을 하게 된다. 등록이 되었으면 해당 반 페이지에 접속할 수 있으며, 과제 페이지에서 출제된 과제들을 확인한다. 과제를 클릭하여 과제 상세 페이지에 접속하여 과제에 대한 설명을 확인한 후, 문제를 클릭하여 풀어야 할 프로그래밍 문제를 확인한다. 문제 상세 페이지 상에서 설명을 읽고, 별도의 IDE 프로그램을 사용하지 않고도 페이지 상의 에디터에 직접 코드를 입력, 실행시켜서 결과를 확인할 수 있다. 코드 작성을 완료했을 시 코드를 제출한다.

* TA의 성적 입력, 학생의 성적 확인

과제 마감일 이후, TA는 성적 페이지에서 해당 주차 과제에 대한 모든 학생의 제출 코드와 시스템에서 생성한 리포트(각 문제 점수, 유사도 검사도, 코드 리뷰)를 확인하여 최종 과제 점수를 부여하고, 첨언이 필요하다고 생각하는 제출물에는 TA 리뷰를 등록한다.  
 TA가 과제 점수를 등록한 이후, 학생은 성적 페이지에서 자신의 최종 과제 점수와 리포트를 확인 가능하다.

3.1.2. 주요 기능 구성

본 서비스의 주요 기능과 기능 간의 흐름을 3.1.1의 사용자 시나리오 구성을 기준으로 아래와 같이 정리하였다.

* 사용자 계정 관리
  + 회원 가입  
     사용자는 회원가입 페이지에서 이름, 소속 학과, 학번 정보를 입력하여 서비스에 가입할 수 있다.
  + 개인 페이지  
     회원가입 시 입력했던 정보를 확인할 수 있는 페이지이며, 별도의 알림이 나타난다.  
     학생의 경우, 반에 성공적으로 등록되었을 시, 새로운 과제가 등록되었거나 기존 과제가 수정되었을 시, 그리고 채점이 완료되어 과제 점수가 등록되었을 시 알림을 받는다.
  + 관리자 계정  
     TA 계정과 TA의 반 개설을 승인, 그리고 모든 반과 학생의 정보를 조회 가능한 계정이다.
* TA가 효율적으로 채점할 수 있는 시스템
  + 반 개설 및 학생 등록

각 반은 TA가 제시하는 과목명과 과목코드 정보를 기반으로 생성된다. 해당 반 페이지는 각 반마다 할당되는 고유한 초대 코드 값을 표시한다. 이 초대 코드를 통해 반 등록 요청을 한 학생 또는 TA의 요청을 구성원 관리 페이지에서 확인할 수 있도록 한다. 또한 구성원 관리 페이지에서는 반의 구성원들을 확인 가능하다.

* + 과제 등록 및 관리

주차별 과제를 등록하기 위해, 과제에 대한 설명과 마감일을 입력받는다. 과제 상세 페이지에서 주차별로 등록된 과제의 정보를 확인할 수 있으며, 과제에 문제를 등록할 수 있다. 문제 등록 페이지에서 문제 설명, 사용할 언어, 테스트케이스와 시스템의 코드 리뷰를 위한 지시사항을 입력받는다.

* + 과제물 채점 시스템

학생이 제출한 코드가 성공적으로 컴파일되었을 경우, 코드를 실행하면서 테스트케이스의 input에 대한 output을 대조한다. 통과한 테스트케이스의 비율에 따라 문제에 대한 점수를 산정한다.  
 과제 마감일 이후, 시스템은 학생들의 모든 제출물에 대해 유사도 검사를 수행하고 코드 리뷰를 생성해야 한다. 과제의 각 문제마다 시스템은 아래와 같이 작업을 수행하게 된다. 이 작업은 마감기한을 넘긴 과제물에 대해서도 동일하게 수행된다.

* + - 다른 학생이 같은 문제에 대해 제출한 모든 코드와 유사도를 계산한다.
    - 학생이 작성한 코드에 대해 ChatGPT 기반 코드 리뷰를 생성한다. 코드가 문제의 의도에 잘 작성되었는지와 조교의 지시 사항을 잘 반영하였는지, 그리고 그 외 추가적인 개선 사안을 제안할 수 있다.
  + 즉 시스템은 학생의 과제 제출물에 대해 각 문제별 점수, 유사도, 코드 리뷰를 생성한다. 이렇게 산출된 결과(이하 리포트)는 차후 TA와 학생의 열람이 가능하다.
* 학생이 서비스 상에서 코드 작성, 실행 및 제출이 가능한 플랫폼
  + 반 등록하기

메인 페이지에서 초대 코드를 입력받을 수 있으며, 사용자가 초대 코드를 입력했을 시 해당 반의 TA에게 요청이 가며, 승인되었을 시 반에 성공적으로 등록된다.

* + 과제 수행을 위한 페이지

학생은 과제 페이지에서 주차별 과제들을, 과제 상세 페이지에서 각 과제에 대한 설명과 문제들을 확인할 수 있다. 문제를 클릭 시 문제 설명, 테스트케이스, 그리고 코드 작성이 가능한 에디터, 결과 창이 포함된 문제 페이지가 나타난다. 문제 페이지에서 학생은 작성한 코드의 임시 저장이 가능하며, 임시 저장된 코드는 다시 해당 페이지를 접속했을 때 불러올 수 있다.  
 에디터에서 작성한 코드를 실행 또는 제출하였을 시, 시스템은 코드를 컴파일하고 실행한다. 컴파일이나 런타임 도중 실패했을 시 실패 원인이 결과 창에 나타난다. 실행까지 성공적이었을 경우, 모든 테스트케이스에 대해 성공적으로 값을 출력하였는지 혹은 일부 오답이 있었는지 여부를 결과 창에서 보여준다.  
 학생의 제출 기한을 넘긴 제출물 및 중복 제출을 허용한다. 시스템은 가장 최근에 제출한 제출물을 학생의 최종 제출물로 간주한다.

* TA의 성적 입력, 학생의 성적 확인
  + TA의 최종 과제 점수와 리뷰 입력

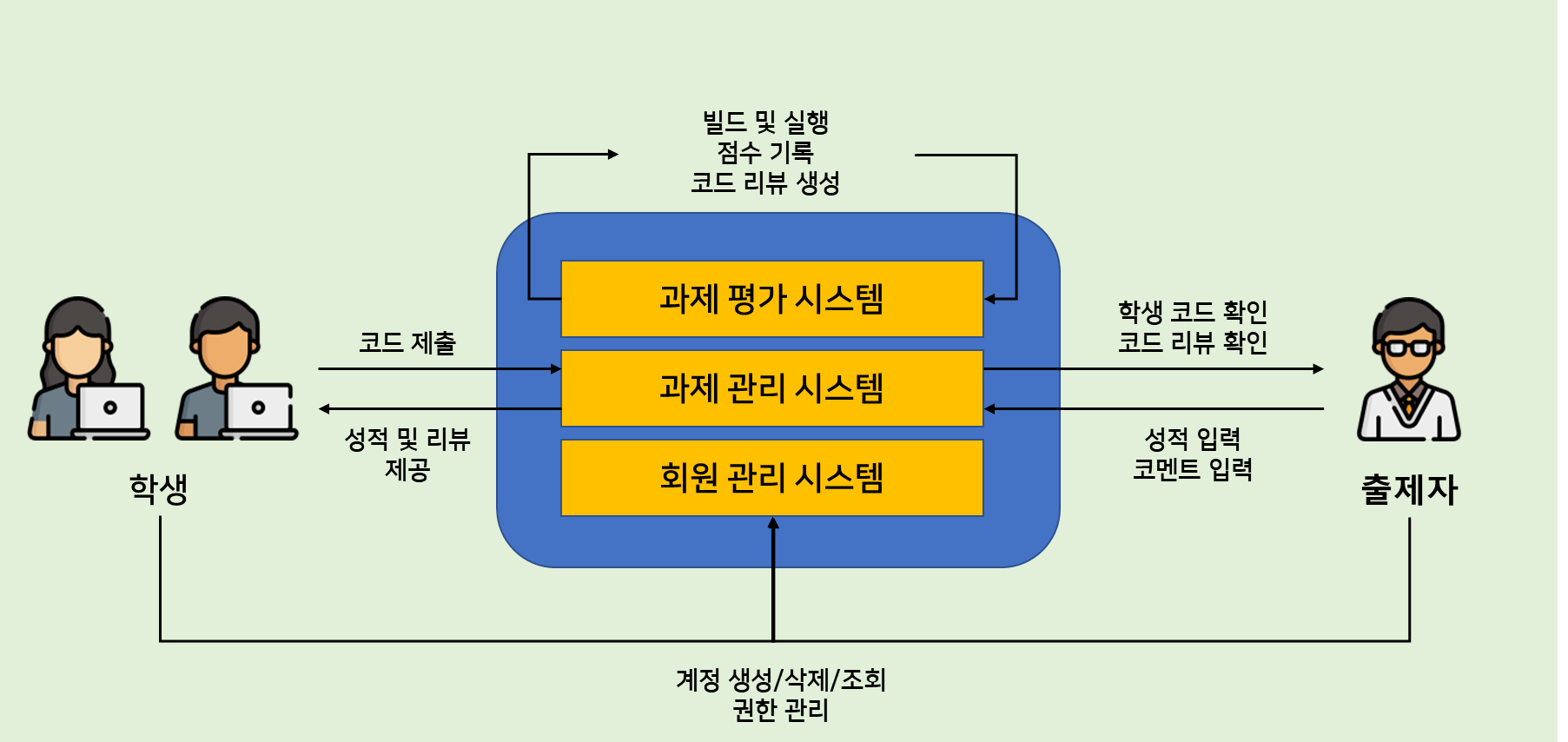
과제 마감 이후, TA는 주차별 과제에 대한 제출 확인 페이지에서 각 학생별로 학생이 작성한 코드들과 시스템이 생성한 리포트를 확인할 수 있다. TA로 하여금 리포트를 참고하여 각 학생의 과제 점수를 기입 및 수정, 별도의 TA 리뷰를 추가할 수 있도록 한다.

* + 학생의 성적 확인

TA의 과제 점수 등록 이후 학생은 TA가 제출한 과제 점수와 TA 리뷰, 그리고 리포트를 최종적으로 확인할 수 있게 된다.

3.1.3. 서비스 아키텍처

* 시스템 아키텍처



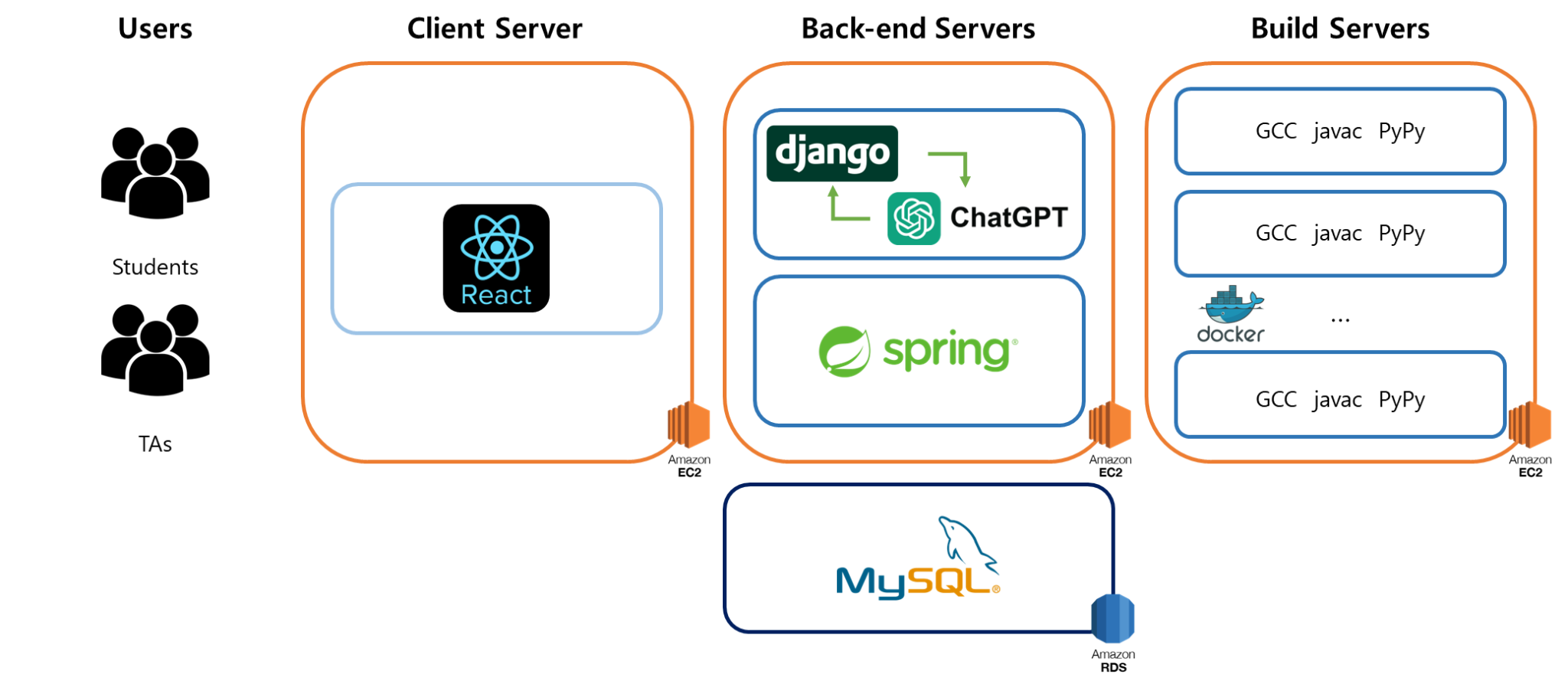
[그림 5] 시스템 아키텍처

본 시스템은 크게 3가지 하위 시스템으로 구성된다. 회원 관리 시스템의 경우 학생 및 TA 계정을 생성하고 계정 정보를 조회하거나 회원 탈퇴시 계정 정보를 삭제하는 기능을 수행한다. 또한 계정 타입에 따라 TA, 학생으로 권한을 구분하고 각 권한에 맞는 서비스를 제공한다.

과제 관리 시스템의 경우 TA가 과제 및 문제를 생성할 수 있게 도와주고 학생의 답안을 입력받는 기능을 제공한다. 학생으로부터 코드를 입력받고 과제 평가 시스템과 통신하여 실행 결과를 응답받아 학생에게 제공한다. 또한 과제에 대한 총점을 TA로부터 입력받아 학생에게 알림을 통해 성적을 보여준다.

과제 평가 시스템의 경우 과제 관리 시스템으로부터 전달받은 코드를 실행시키고 그 결과를 반환한다. 또한 생성형 AI인 ChatGPT에게 코드 리뷰를 요청하고 받은 결과를 DB에 저정할 수 있도록 반환한다. 각 시스템은 독립적으로 동작할 수 있어야 하며 이를 통해 특정 시스템의 변경이 다른 시스템에 주는 영향을 최소화할 수 있어야 한다.

* SW 아키텍처



[그림 6] SW 아키텍처

[그림 6]은 전반적인 SW 구조도를 나타낸 것이다. AWS의 EC2 인스턴스를 통해 서버 인스턴스를 배포하고, 각 인스턴스는 클라이언트, 백엔드, 빌드 서버로 사용된다.

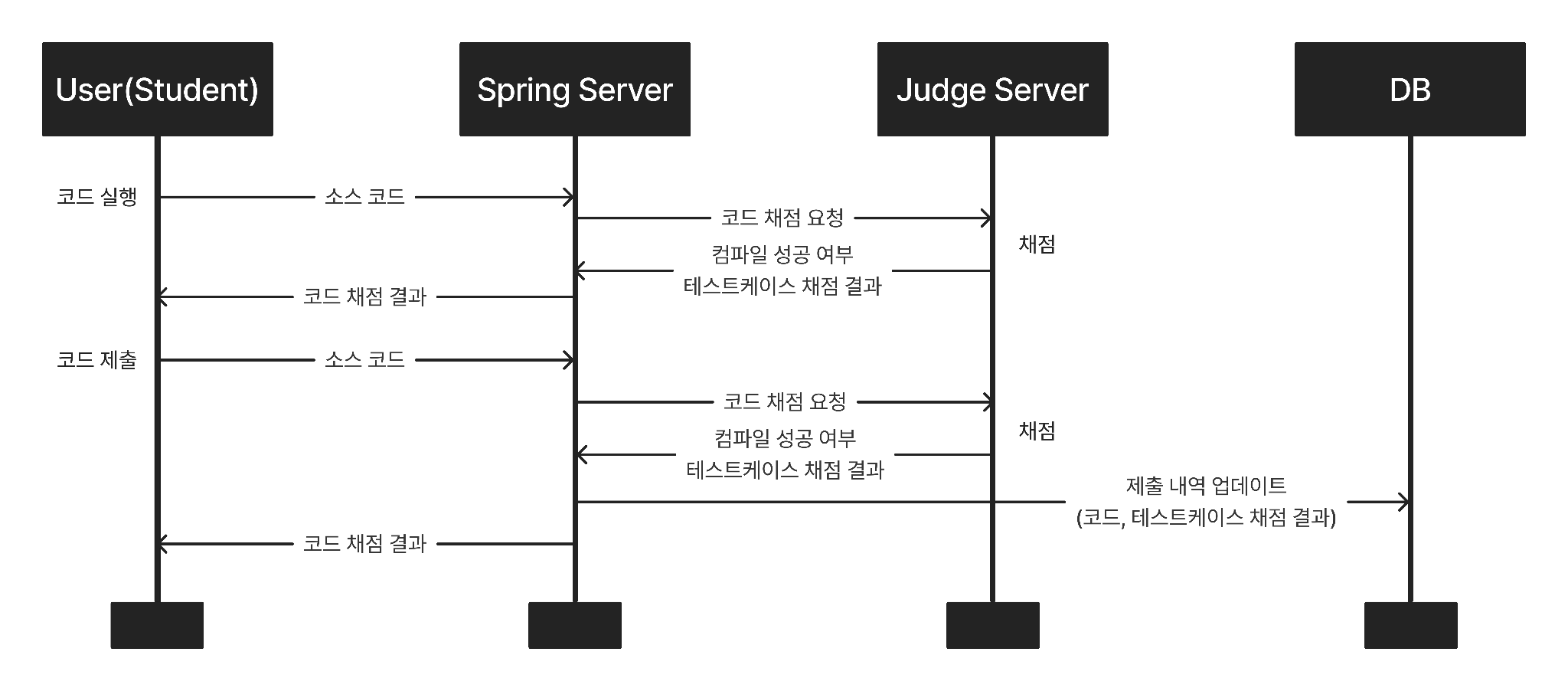
클라이언트 서버의 경우 React로 구현된 웹 어플리케이션을 동작시키고 사용자로부터 회원가입 및 로그인, 코드 작성, 성적 입력 및 확인 등 사용자가 서비스를 이용할 수 있도록 인터페이스를 제공한다.

백엔드 서버의 경우 도커를 통해 각각 Django와 Spring 서버를 위한 가상 컨테이너를 운용한다. Spring 서버는 회원 및 과제 관리를 위한 전반적인 비즈니스 로직을 담당하고, 코드 유사도 검사 및 ChatGPT API 호출이 필요할 경우 Django 서버로 요청을 날려 응답을 받는다. 데이터 삽입/삭제/조회 등의 비즈니스 로직이 실행될 때엔 RDS 서버에서 제공하는 MySQL을 통해 DB 기능을 활용한다.

마지막으로 빌드 서버의 경우 컴파일러 환경이 구축된 도커 컨테이너를 운용하여 백엔드 서버로부터 넘어온 빌드 요청을 수행하고 실행 결과 및 에러 메시지를 반환한다. 사용자의 여러 요청에도 수행할 수 있도록 가용성을 높이기 위해 각 언어에 해당하는 컴파일러가 설치된 컨테이너를 복제하여 운용하는 구조로 설계할 예정이다.

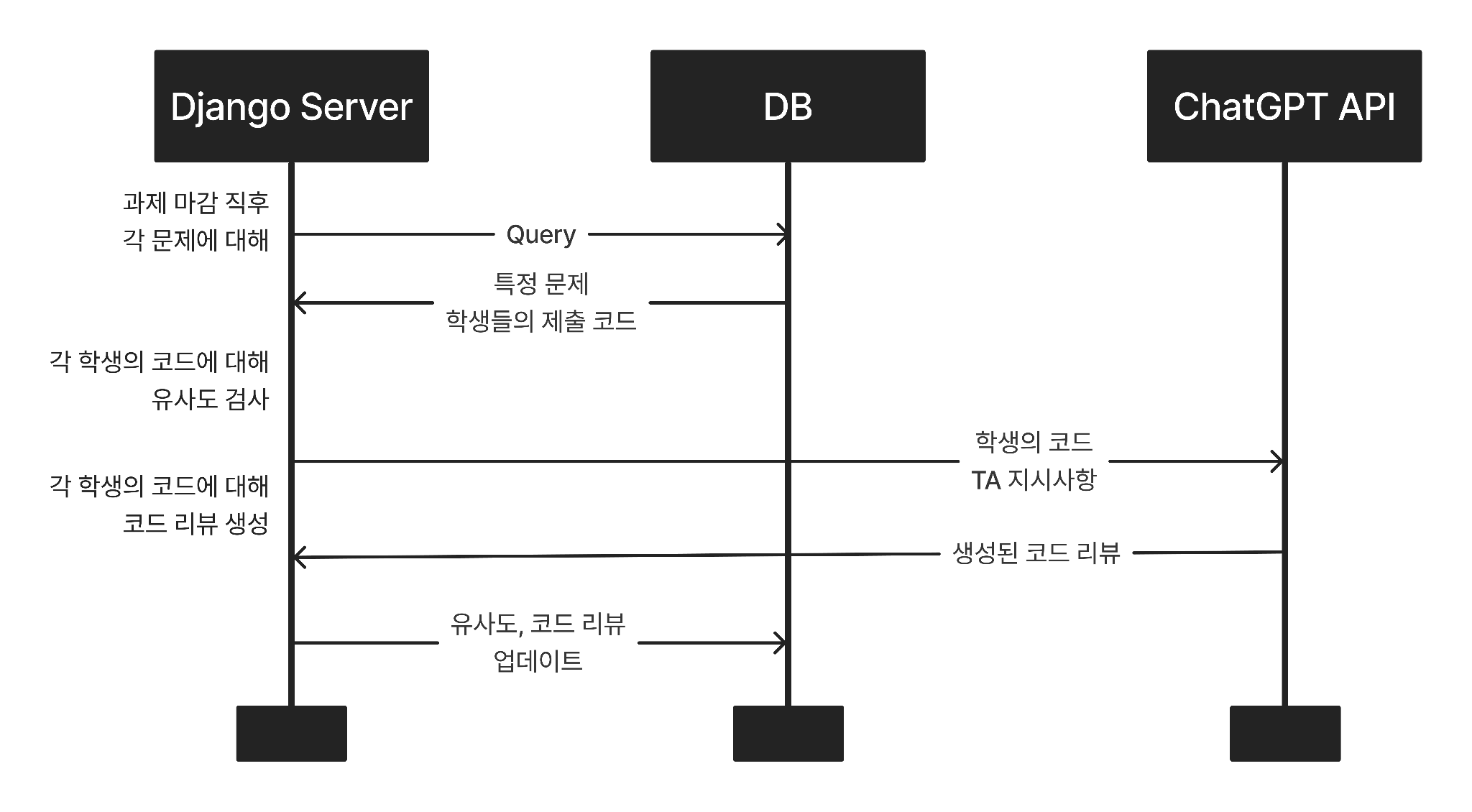
3.1.4. 프론트엔드-백엔드 데이터 흐름

백엔드 사용자의 활동에 따라 프론트엔드와 백엔드에서 주고받는 데이터들의 개략적인 흐름은 아래와 같다.



[그림 7] 학생의 코드 실행 및 제출 과정

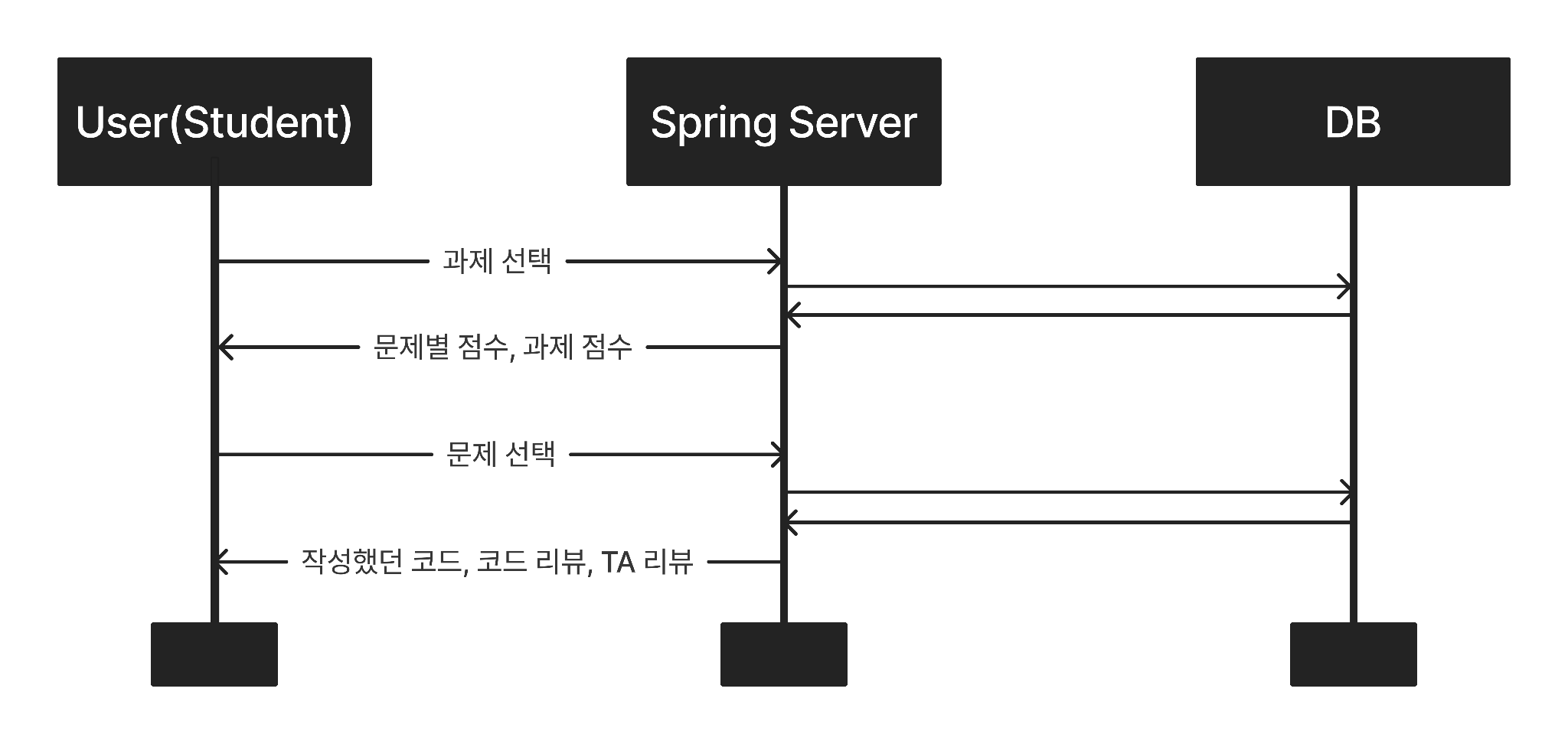
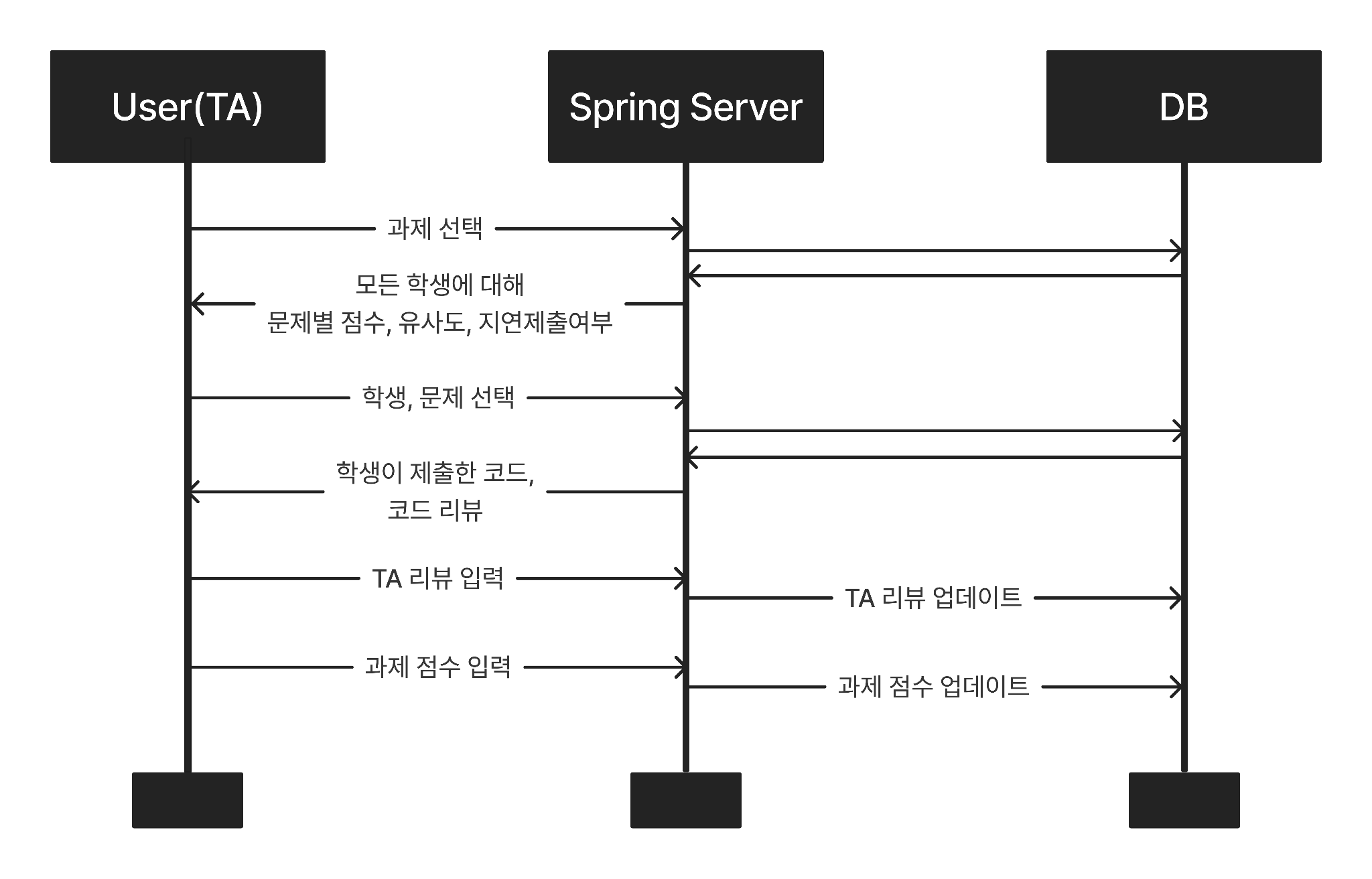
학생이 작성한 소스 코드는 실행 및 제출 시 Spring 서버에 전달되고, Spring 서버에서는 빌드 서버에 소스 코드와 테스트케이스를 전달해 채점을 요청한다. 빌드 서버가 컴파일과 실행 과정을 통해 채점을 마치면 결과를 Spring 서버에 보내게 되며, Spring 서버는 결과를 토대로 사용자에게 출력할 메시지를 전달하게 된다. 코드 제출의 경우 제출한 코드와 채점 결과가 DB에 업데이트된다.



[그림 8] 시스템의 리포트 생성 과정

과제 마감 이후, Django 서버에서 코드 유사도 검사와 코드 리뷰 생성을 담당한다. 과제의 각 문제에 대해 학생들의 제출 코드들을 가져와 각 코드에 대해 유사도를 검사하고 ChatGPT API 호출을 통해 코드 리뷰를 생성한다. 생성된 리포트는 DB에 업데이트된다.

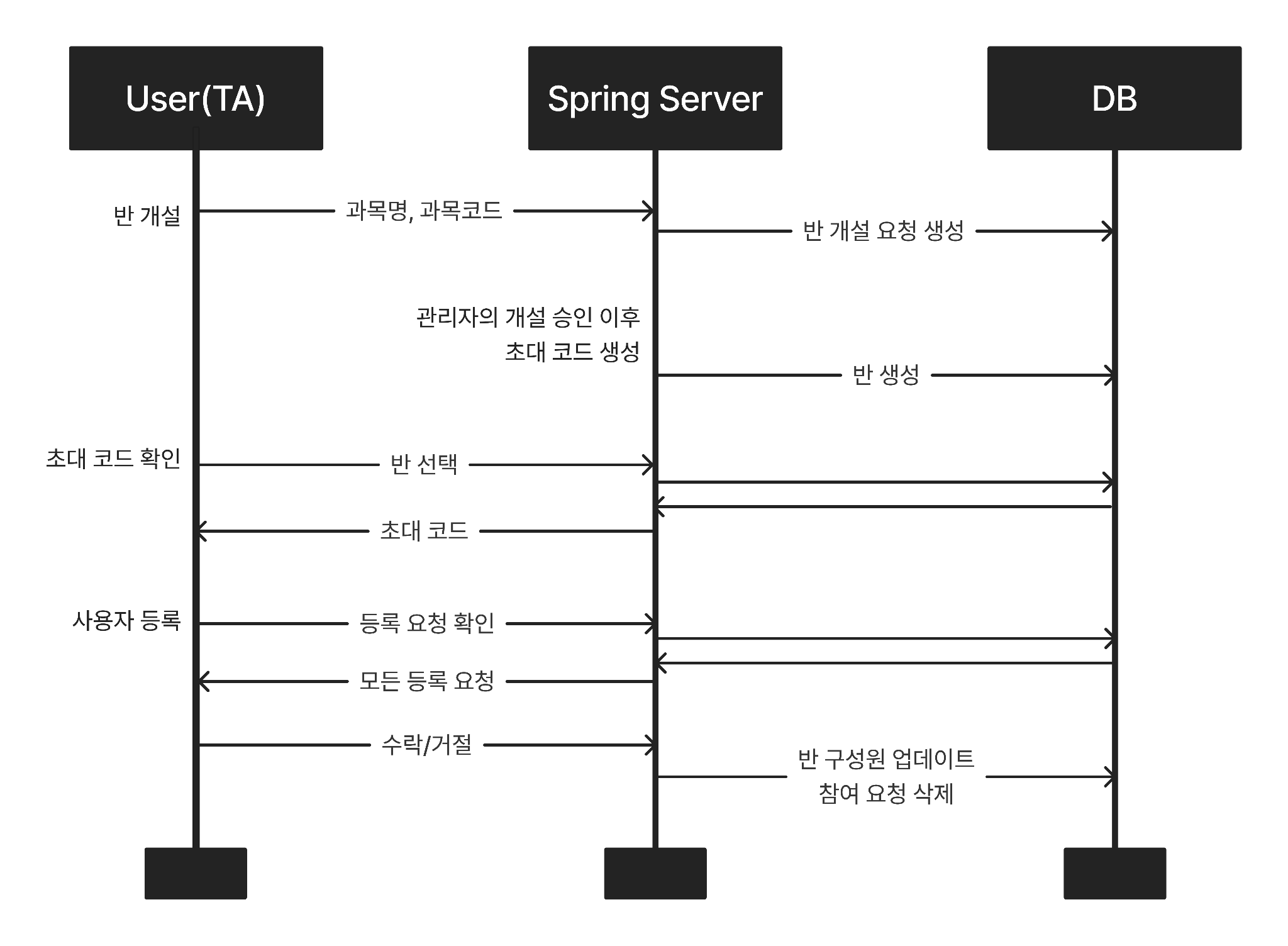
지연 제출의 경우, 리포트 생성을 위해 Spring 서버에서 Django 서버에 요청을 보낼 수 있다.



[그림 9] (좌)TA의 제출물 확인 및 TA 리뷰, 과제 점수 입력 과정

[그림 10] (우)학생의 성적 확인 과정

TA가 입력하는 TA 리뷰와 과제 점수가 DB에 업데이트 된 후, 학생은 그 결과를 확인 가능하다.



[그림 11] TA의 반 개설, 구성원 등록 과정

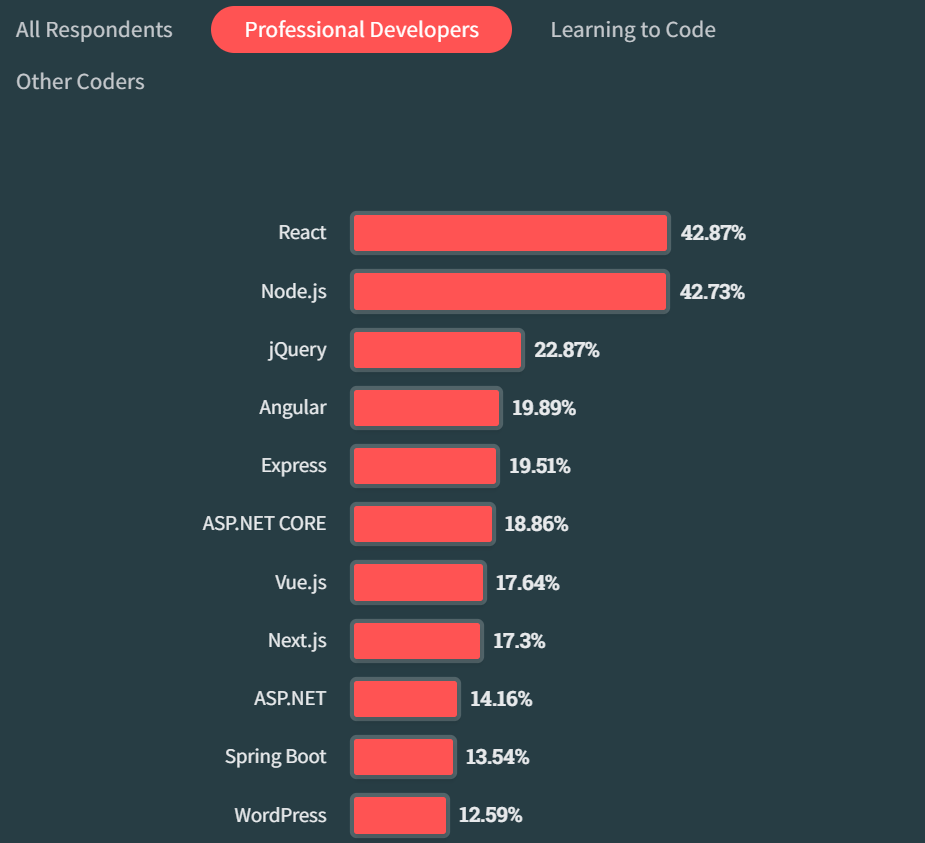
반 개설이 관리자에 의해 승인된 후, Spring 서버는 초대 코드를 생성하고 반 정보를 DB에 생성한다. TA는 반에 등록하고자 하는 구성원이 초대 코드를 입력해서 생성된 참여 요청을 확인하여 수락 및 거부할 수 있고, 수락한 경우 구성원 정보가 업데이트되며 DB에서 참여 요청은 삭제된다.

3.2. 개발환경 및 기술스택

3.2.1. Front-end

* React.js

React는 웹 애플리케이션을 만들기 위한 JavaScript 라이브러리로, 컴포넌트 기반으로 UI를 구성하며, 가상 DOM을 사용하여 성능을 최적화한다. 데이터는 단방향으로 흐르며 JSX 문법을 통해 UI를 선언할 수 있다. React는 큰 개발자 커뮤니티와 다양한 확장 기능을 제공하며, 웹 및 모바일 애플리케이션을 개발할 때 유용하게 사용된다. 밑과 같이 해당 라이브러리의 웹 개발 점유율이 최상위권을 차지하고 있음을 확인할 수 있다.



[그림 12] Which web frameworks and web technologies have you done extensive development work in over the past year (stackoverflow)

3.2.2. Back-end

* Spring Boot

웹 어플리케이션 및 API 서버 설계에 사용되는 대표적인 서버 프레임워크다. 내장 서버를 갖추고 있어 별도의 웹서버 구축 과정이 필요없으며 Spring에서 제공하는 기능들을 간편하게 이용 가능하도록 제공하고 있어 빠르고 안정적인 구현이 가능하다. 해당 프레임워크는 계정 생성, 과제 및 문제 출제, 점수 등록 등 전반적인 서비스 로직을 담당하는 서버 구현에 활용할 예정이다.

* Django

보안이 우수하고 유지보수가 편리한 웹사이트를 신속하게 개발하는 하도록 도움을 주는 파이썬 웹 프레임워크이다. ChatGPT API 사용 및 코드 유사도 검사 등 자연어 처리 기능을 구현하기 위한 서버를 설계할 때 본 프레임워크를 사용하기로 하였다.

* MySQL

데이터베이스 구축의 경우 보안 측면의 취약성과 직면하기 때문에 레퍼런스와 관련 자료의 유무가 더욱 중요하게 고려되었다. 데이터베이스의 활용빈도를 중점으로 조사했고, 결과적으로 세계에서 가장 널리 사용되며, 전문 개발자들이 가장 많이 사용하는 MySQL을 채택하게 되었다.

3.2.3. AI Model

* ChatGPT는 Pre-trained Transformer(GPT)와 Chat의 합성어로, OpenAI에서 개발한 프로토타입 대화형 인공지능 챗봇이다. 지도학습과 강화학습을 모두 사용해 파인 튜닝되었다.
* 학생이 과제관리시스템을 통해 제출한 코드를 ChatGPT API를 이용해 분석하여 사용자에게 제시하고 기존 코드와 비교할 수 있다.

3.2.4. 서버 및 배포

* Amazon Web Service(AWS)

AWS는 배포 및 스토리지 서버 구축, RDS 서버 제공, 네트워크 보안 및 VPC 구성 기능을 제공하는 대표적인 클라우드 서비스 플랫폼이다. 서버 구축의 편의성과 팀원들의 사용 경험이 가장 많은 서비스라는 측면에서 해당 플랫폼을 채택하였고 본 프로젝트에서 구현한 서비스를 AWS 플랫폼에서 제공하는 서버 인스턴스를 통해 배포할 예정이다.

* Docker

가상 환경을 통해 개발, 배포, 실행을 효율적으로 관리해주는 플랫폼이다. 컨테이너 기반 가상화를 지원하여 어플리케이션 실행에 필요한 환경 및 종속성들을 포함하여 배포할 수 있게 해준다. Docker를 활용해 여러 개의 빌드 서버용 컨테이너를 구성하여 서버 인스턴스의 가용성을 높일 수 있다.

* GitHub Actions

GitHub에서 제공하는 플랫폼이자 빌드 및 테스트, 배포 프로세스의 자동화와 지속 배포 기능을 제공하는 CI/CD 툴이다. GitHub Actions를 이용해 개발 생산성을 향상시키고 민감 파일들을 암호화하여 사용할 수 있기 때문에 해당 플랫폼을 사용하기로 하였다.

3.2.5. 코드 형상관리

* GitHub (<https://github.com/dhcryan/2023_Software_Engineering>)



[그림 13] git

Git은 컴퓨터 파일의 변경사항을 추적하고 여러 명의 사용자들 간에 해당 파일들의 작업을 조율하기 위한 스냅샷 스트림 기반의 분산 버전 관리 시스템이다. 또는 이러한 명령어를 가리킨다. 소프트웨어 개발에서 소스 코드 관리에 주로 사용되지만 어떠한 파일 집합의 변경 사항을 지속적으로 추적하기 위해 사용될 수 있다. 기하학적 불변 이론을 바탕으로 설계됐고, 분산 버전 관리 시스템으로서 빠른 수행 속도에 중점을 두고 있는 것이 특징이며 데이터 무결성, 분산, 비선형 워크플로를 지원한다.

3.2.6. UI/UX 디자인 관리



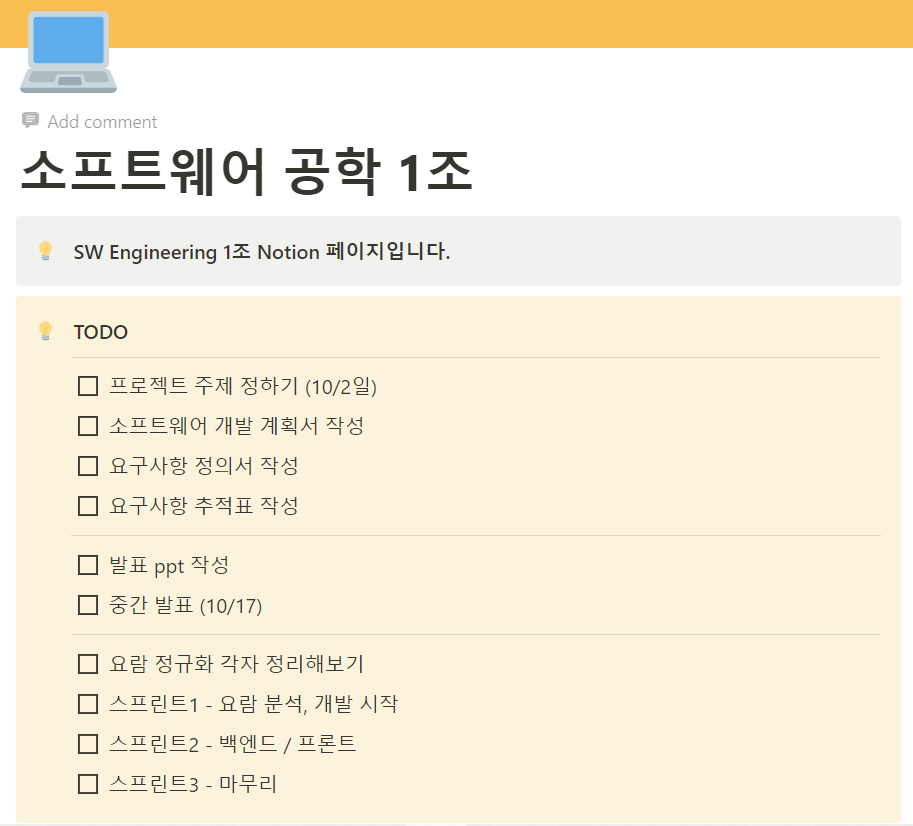
[그림 14] Figma

* Figma

Figma는 인터페이스 설계를 위한 협업 웹 애플리케이션으로, 맥OS와 윈도우용 데스크톱 애플리케이션에 의해 추가적인 오프라인 작업이 가능하다. Figma의 기능들은 사용자 인터페이스와 사용자 경험 설계에 중점을 두고 실시간 협업을 강조하며, 다양한 벡터 그래픽 편집기 및 프로토타이핑 도구를 활용한다. 안드로이드 및 iOS용 피그마 모바일 앱은 모바일 및 태블릿 기기에서 실시간으로 Figma 프로토타입을 볼 수 있다.

3.2.7. 문서관리 및 협업 관리

* 문서 관리 : Google Drive 공유문서함 및 Google Docs
* 소통 : Google meeting, Slack
* 프로젝트 및 일정 관리 : Gantt Chart, Burndown Chart
* 기록 및 협업 관리 : Notion



[그림 15] Notion을 통한 회의 기록

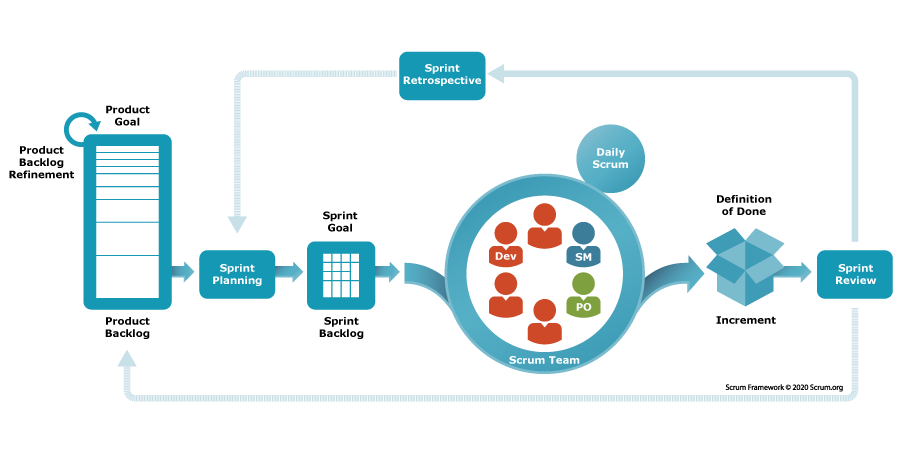
3.2.8. 개발 환경 (IDE)

* Visual Studio Code
* Vim

4. 세부 추진 계획

4.1. 개발방법론 선정 및 WBS정의

4.1.1. 애자일 프로세스 및 스크럼 계획

애자일 원칙과 스크럼 방법론을 준수하면서, 각 스프린트마다 명확한 목표와 스토리 포인트를 설정하여 프로젝트를 진행한다. 스크럼 마스터는 팀의 진행을 지속적으로 모니터링하고 장애물을 제거한다. 매주 스프린트 회고를 통해 개선점을 도출하고 프로젝트를 지속적으로 발전시킨다. 전체 프로젝트 기간을 다섯 단계로 쪼개, 각 스프린트를 2주일 단위로 설정한다. 개발 단계에서 스프린트 및 데일리 스크럼을 바탕으로 프로젝트를 진행하고, 스프린트 종료 시 그에 따른 스프린트 리뷰를 진행한다.

[그림 16] Scrum process

이에 따른 효과는 다음과 같다.

첫째, 짧은 스크럼 주기는 빠른 개발과 빠른 고객 피드백을 가능하게 한다. 매주 진행 상황을 검토하고 개선할 수 있으므로, 제품을 빠르게 개발하고 필요한 조치를 신속하게 취할 수 있다.

둘째, 스크럼은 투명성을 높인다. 주기적인 미팅을 통해 팀원들은 작업 상황을 공유하고 문제점을 공개적으로 토론하므로, 빠른 조치를 취할 수 있다. 팀은 스스로 작업을 계획하고 문제를 해결하며, 주기적인 회고를 통해 성과를 개선할 수 있다.

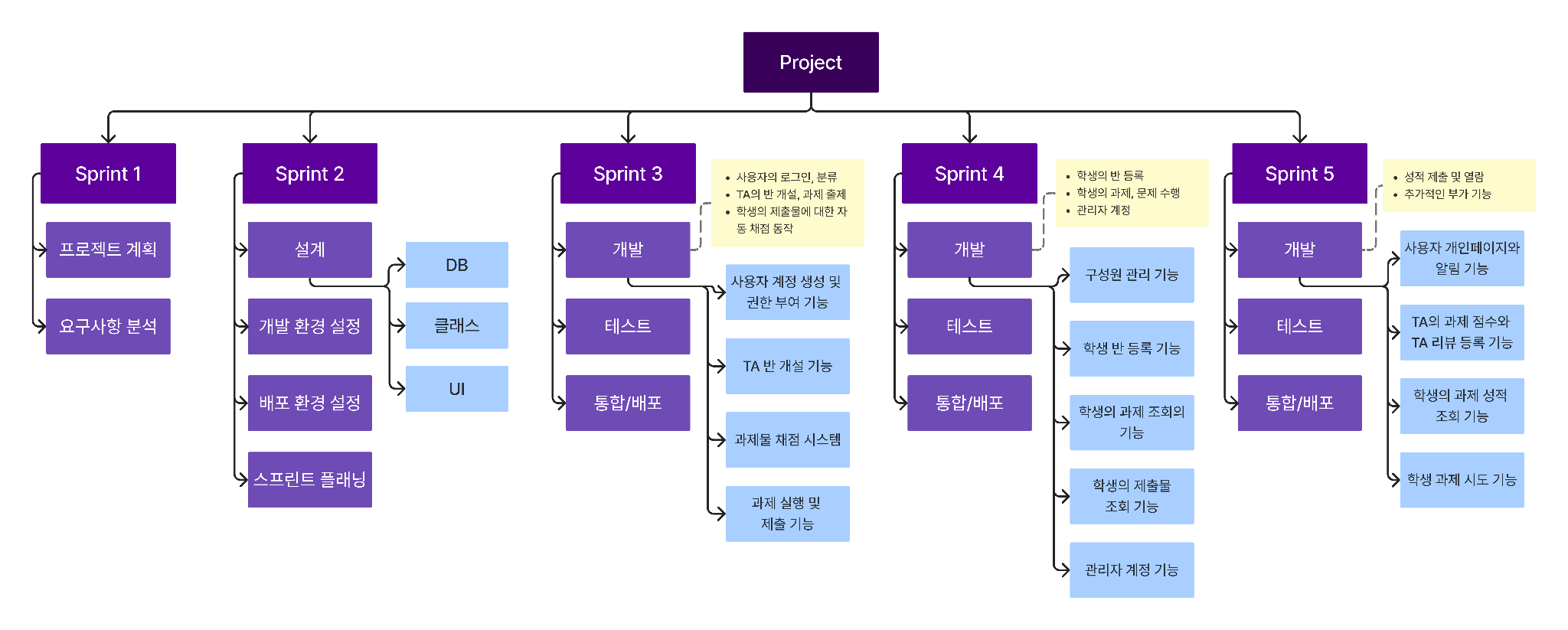
이러한 이유로 스크럼 방법론은 프로젝트를 효과적으로 관리하고 고객의 요구를 빠르게 충족시키며, 제품 품질을 향상시키는 데 도움이 될 것이다.

4.1.2 WBS

WBS는 프로젝트에서 수행해야 할 모든 작업들의 분해 구조이며, 일종의 가계도처럼 생각될 수 있다. 작성된 WBS를 토대로 프로젝트의 범위를 파악하고 업무 일정을 수립함으로써 프로젝트 진행 단계에서 업무들을 효율적으로 관리할 수 있다.

본 프로젝트에서는 요구사항 분석 단계를 토대로 설계, 기능 구현, 테스트, 배포 등의 절차에 수반되는 작업들을 243개의 최소 단위로 세분화하였다. 세분화된 단위들은 애자일 프로세스와 스크럼 방법론을 고려, 전체적으로는 스프린트 단위로 분류되었으며, 각 스프린트에서 취해야 할 액티비티 기반으로 구분되어진다.

아래의 WBS 차트는 전체 프로젝트에서의 스프린트와 액티비티 구조, 그리고 일부 태스크를 보여주며, 개발 액티비티에서의 목표를 기술한다.



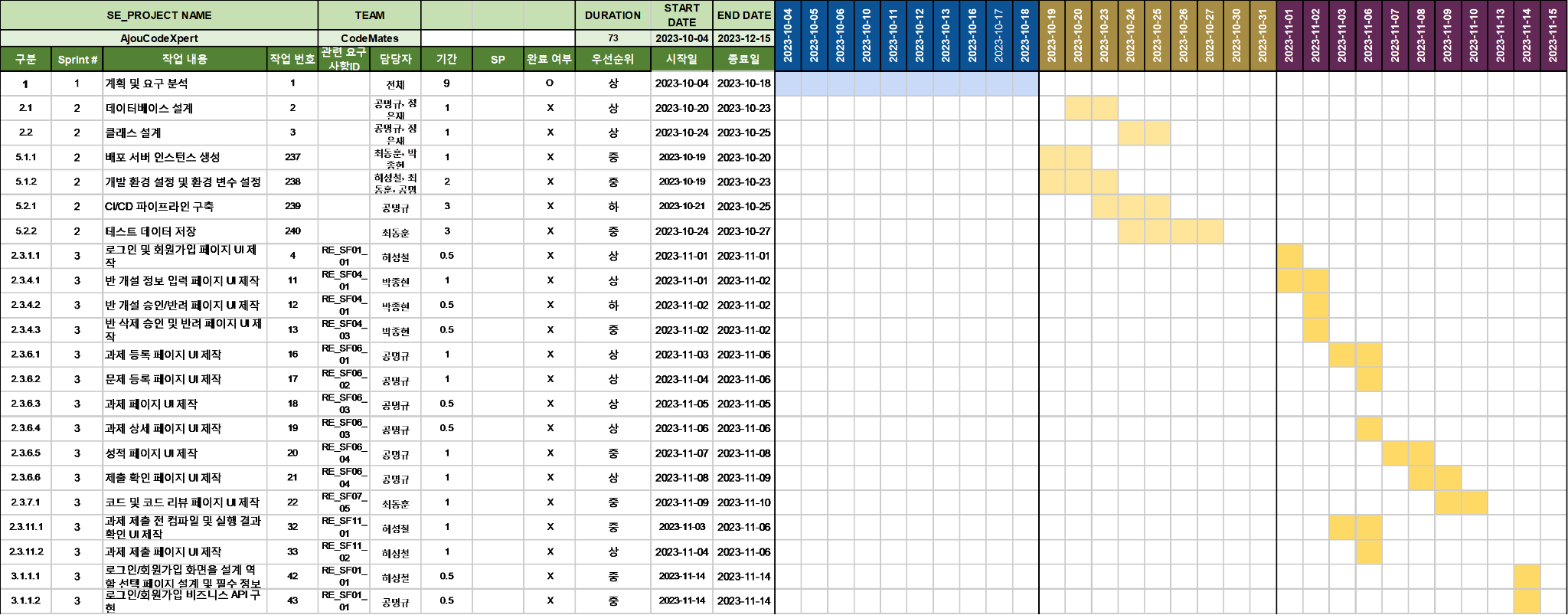
[그림 17] WBS 차트

4.2. 역할분담

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **이름** | 공명규 | 박종현 | 정은재 | 최동훈 | 허성철 |
| **개발 영역** | Spring 서버 구현  백엔드 로직 설계 | 페이지 제작  Wireframe 기획 | DB 설계  아키텍처 설계 | Django 서버 구현  챗GPT 연동 구현 | 웹 App 설계 |
| **비개발 영역** | 기획 및 총괄 | 일정 관리 및 진행 사항 관리 | 테스트 설계 및 문서화 | 문서 관리  자료 조사 | UI 기획  이슈 관리 |

[표 1] 역할 분담 표

4.3. 개발일정



1. 참고자료

* What is Blackboard Learn?  
  <https://help.blackboard.com/ko-kr/Learn/Instructor/Ultra/Getting_Started/What_Is_Blackboard_Learn>
* IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements
* Specifications, In IEEEXplore Digital Library
* Django Software Foundation, “Django Project”. https://www.djangoproject.com/