

# Software Requirements Specification

for

## Web Coding Test



2017312425 고유안  
2018314384 배재현  
2015314136 조규상  
2018312790 최민석  
2017311028 최주은  
2018310619 황선진

소프트웨어공학개론 6 조

2022. 10. 30

## 내용

<b>1. INTRODUCTION.....</b>	<b>6</b>
<b>1.1. PURPOSE.....</b>	<b>6</b>
<b>1.2. SCOPE.....</b>	<b>6</b>
<b>1.3. DEFINITION, ACRONYMS, AND ABBREVIATION.....</b>	<b>6</b>
<b>1.4. REFERENCES.....</b>	<b>8</b>
<b>1.5. OVERVIEW.....</b>	<b>8</b>
<b>2. OVERALL DESCRIPTION.....</b>	<b>8</b>
<b>2.1. PRODUCT PERSPECTIVE.....</b>	<b>8</b>
<b>2.2. PRODUCT FUNCTIONS.....</b>	<b>9</b>
2.2.1. MAIN PAGE.....	10
2.2.2. RUN RESULT PAGE.....	10
2.2.3. GRADING RESULT PAGE.....	11
2.2.4. SUBMIT & RESULT ANALYSIS PAGE.....	11
<b>2.3. USER CLASSES AND CHARACTERISTICS.....</b>	<b>11</b>
2.3.1. SYSTEM ADMINISTRATOR.....	11
2.3.2. STUDENT.....	12
2.3.3. LECTURER & TA.....	12
<b>2.4. OPERATING ENVIRONMENT.....</b>	<b>12</b>
<b>2.5. DESIGN AND IMPLEMENTATION CONSTRAINTS.....</b>	<b>12</b>
<b>2.6. ASSUMPTIONS AND DEPENDENCIES.....</b>	<b>13</b>
<b>3. EXTERNAL INTERFACE REQUIREMENTS.....</b>	<b>13</b>
<b>3.1. USER INTERFACE.....</b>	<b>13</b>
<b>3.2. HARDWARE INTERFACE.....</b>	<b>20</b>
<b>3.3. SOFTWARE INTERFACES.....</b>	<b>20</b>
<b>3.4. COMMUNICATION INTERFACES.....</b>	<b>21</b>
<b>4. SYSTEM REQUIREMENTS.....</b>	<b>23</b>
<b>4.1. FUNCTIONAL REQUIREMENTS.....</b>	<b>23</b>
4.1.1. USE CASE.....	23

4.1.2. USE CASE DIAGRAM.....	30
4.1.3. DATA FLOW DIAGRAM.....	31
<b>4.2. NON FUNCTIONAL REQUIREMENTS.....</b>	<b>31</b>
4.2.1. PRODUCT REQUIREMENT.....	31
4.2.2. ORGANIZATIONAL REQUIREMENT.....	32
4.2.3. EXTERNAL REQUIREMENT.....	32
<b>4.3. ORGANIZING SYSTEM FLOW.....</b>	<b>33</b>
4.3.1. CONTEXT MODEL.....	33
4.3.2. PROCESS MODEL.....	34
4.3.3. INTERACTION MODEL.....	34
4.3.4. BEHAVIOR MODEL.....	34
<b>4.4. SYSTEM ARCHITECTURE.....</b>	<b>35</b>

## 그림 목차

Figure 1. Design of Main Page.....	15
Figure 2. Design of Main Page – Result of Grading.....	17
Figure 3. Design of Main Page – Default page for Result of Submit.....	19
Figure 4. Design of Main Page – Result of Submit, Extended Code Editor.....	19
Figure 5. Design of Main Page – Result of Submit, Code Description and Related Materials .....	20
Figure 6. Use Case Diagram.....	30
Figure 7. Data Flow Diagram.....	31
Figure 8. Context Model.....	33
Figure 9. Process Model.....	34
Figure 10. Sequence Diagram.....	35
Figure 11. System Architecture.....	36

## 표 목차

Table 1. Acronyms/Abbreviation.....	7
Table 2. Terms and definitions.....	7
Table 3. Table of Product Functions.....	9
Table 4. User Interface of Main Page.....	13
Table 5. User Interface of Main Page – Result for Run.....	15
Table 6. User Interface of Main Page – Result of Grading.....	16
Table 7. User Interface of Main Page – Result of Submit.....	17
Table 8. Hardware Interface for User.....	20
Table 9. software interface of db.....	20
Table 10. Communication Interface of Web browser.....	21
Table 11. Functional requirements abstraction.....	23
Table 12. Use Case of Question Information.....	24

Table 13. Use Case of Code Edit and Run.....	24
Table 14. Use Case of Test Case Check.....	25
Table 15. Use Case of Code Save and Load.....	25
Table 16. Use Case of Code Reset.....	26
Table 17. Use Case of Code Copy.....	26
Table 18. Use Case of Code Download.....	27
Table 19. Use Case of Code Submit.....	27
Table 20. Use Case of Code Grading.....	28
Table 21. Use Case of Code Analysis.....	28
Table 22. Use Case of Reference Material.....	29

# 1. INTRODUCTION

## 1.1 . PURPOSE

본 문서는 “Web Coding Test” 서비스를 위한 소프트웨어 요구 명세서이다. “Web Coding Test” 서비스는 2022 학년 2 학기 성균관대학교 소프트웨어공학개론 수업의 수강생으로 이루어진 Team6 가 설계 및 구현하는 것으로 본 문서에서 해당 서비스를 위한 요구 사항을 정리하고 분석한다. Team6 는 본 문서에 기재된 내용에 기반하여 본 시스템을 개발한다.

본 문서의 주요 독자는 Team6 이다. 부가적으로 앞서 언급한 소프트웨어공학개론 강의의 교수, 조교, 수강자 일체를 본 문서의 독자로 상정한다. 이외에도 교육적, 비상업적 목적으로 본 문서가 열람되거나 사용되는 것을 허용한다.

본 문서의 목적은 “Web Coding Test” 시스템을 개발하고 제공함에 있어서 요구사항을 정리하고 명시하는 데에 있다. “Web Coding Test” 시스템은 사용자에게 양질의 코딩 학습 경험을 제공하기 위해 고안된 코딩 테스트 웹 어플리케이션이다. 해당 시스템의 개요와 그를 구성하는 요소인 UI/UX, 서버, 데이터베이스 사용에 대해 본 문서에서 요구사항을 정리함으로써, 향후 프로젝트 진행에 있어서 그 기반과 근거가 되도록 한다.

## 1.2. SCOPE

“Web Coding Test”는 다양한 난이도의 코딩 문제를 제공함으로써 전공자는 물론 비전공자의 소프트웨어 컴퓨팅 역량을 키울 수 있도록 하는 서비스이다. 따라서 해당 분야에 익숙하지 않은 사람도 5-10 분 이내의 사이트 탐색 과정 있다면 직관적으로 사용 가능한 UI/UX 를 가지고 있어야 한다. 그에 더해 기본적으로 사용자가 시스템 전반을 사용함에 있어서 그를 방해하거나 사용자 본인이 불편을 느낄 수 있는 기능적 오류가 있어서는 안 된다. 사용자가 결과를 확인하고 그것을 통해 본인의 실력을 향상시킬 수 있는 방식으로 정확한 채점 결과와 연관성 높은 관련 자료를 제공해야 한다. 위 사항을 포함한 front-end 와 back-end 에 있어서의 전체 기능에 대한 상세는 이하 문서에서 설명하도록 한다.

## 1.3. DEFINITION, ACRONYMS, AND ABBREVIATION

아래의 표는 본 문서에서 사용된 약어와 그 설명을 나타낸다.

Table 1. Acronyms/Abbreviation

Acronyms/Abbreviation	Description
UI/ UX	User Interface/ User Experiance
DB	Data Base
OS	Operating System
GUI	Graphic User Interface
URL	User Resource Locator

HTTP	HyperText Transfer Protocol
API	Application Programing Interface

아래의 표는 본 문서에서 사용한 용어와 그 정의를 나타낸다.

**Table 2. Terms and definitions**

Term	Definition
Front-end	All software or hardware that is part of a user interface.
Back-end	Parts of a computer application or a program's code that allow it to operate and that cannot be accessed by a user.
Figma	A collaborative web application for interface design, with additional offline features enabled by desktop applications for mac OS and Windows.
Django	A free and open-source, Python-based web framework that follows the model-template-views (MTV) architectural pattern.
SQLite	SQLite is a database engine written in the C programming language.
React	A free and open-source front-end JavaScript library for building user interfaces based on UI components.
Clipboard	A clipboard is a temporary storage area for data that the user wants to copy from one place to another.
Skeleton	Skeleton code is used to assist programmers to develop their code with the fewest errors during the time of compilation.
Actor	Individuals or systems that will interact with the application.
Stimulus	An event or input to a system.
Response	A response to stimulus.

## 1.4. REFERENCES

- "IEEE Guide for Developing System Requirements Specifications," in IEEE Std 1233-1996 , vol., no., pp.1-30, 22 Dec. 1996, doi: 10.1109/IEEESTD.1996.81000.
- Team 2. "Online Coding Class about Web Crawler". SKKU, Last Modified: April. 09, 2022 [https://github.com/skkuse/2022spring\\_41class\\_team2](https://github.com/skkuse/2022spring_41class_team2)
- Team 3. "전교생 Python 코딩 교육 시스템". SKKU, Last Modified: April. 09, 2022 [https://github.com/skkuse/2022spring\\_41class\\_team2](https://github.com/skkuse/2022spring_41class_team2)
- "UML Diagram Types Guide", Creately, Last Modified: September. 28, 2022 <https://creately.com/blog/diagrams/uml-diagram-types-examples/>

## 1.5. OVERVIEW

본 문서의 이하 내용 구성은 다음과 같다. 두 번째 장에서는 제품 관점, 제품 기능, 운영환경, 제약사항 등 시스템에 대한 전체적인 설명을 제공한다. 세 번째 장과 네 번째 장에서는 제품의 요구사항이 기술되는데 세 번째 장은 시스템의 여러가지 인터페이스 관련 사항을 기술한다. 네 번째 장은 시스템의 기능적, 비기능적 요구사항과 시스템 플로우의 모델, 시스템 구조에 관련된 요구사항을 포함하고 있다. 마지막으로 다섯 번째 장에서는 본 문서의 이력 등 추가적 정보를 기술한다.

## 2. OVERALL DESCRIPTION

### 2.1. PRODUCT PERSPECTIVE

성균관대학교 소프트웨어학과 전공 과목 중 자료구조개론, 알고리즘개론, 문제해결 등의 과목에서 출제되는 많은 과제가 주어진 입력에 따라 올바른 결과를 출력하는 코드를 작성하는 ‘코딩 테스트’형 과제에 해당한다. 또한, 성균관대학교에서는 2016년부터 전교생이 수강해야 하는 필수 교양 과목으로 컴퓨팅사고와 SW 코딩, 문제해결과알고리즘 등을 지정하여 소프트웨어학과 또는 이공계열 전공생이 아니더라도 기본적인 소프트웨어 코딩 역량을 갖추도록 하고 있는데, 이러한 과목들도 코딩 테스트형 과제를 주로 출제하고 있다.

하지만 코딩 테스트형 과제를 수행하기 전에는 학생 개인 PC에 C, Python 등의 코드를 작성하기 위한 환경 설정이 필수적이다. 또한, 이후 과제를 수행하는 과정에서는 학생이 작성한 풀이 코드가 오픈 테스트케이스뿐만 아니라 히든 테스트케이스도 통과하는 올바른 코드인지 확인하기 힘들고, 시간복잡도 또는 공간복잡도와 같은 효율과 라인 수, 변수명과 같은 가독성이 높은 코드를 작성했는지 어렵기 때문에 과제를 올바른 방향으로 진행하고 있는지 확인하기 힘들다.

이 시스템에서 제공하는 서비스를 통해 학생들은 불필요한 환경 설정 과정 없이 웹 사이트에 접속하여 편리하게 과제를 수행할 수 있고, 테스트케이스 제공, 상세한 채점 결과와 코드 설명 및 관련 자료 등을 제공 받아 과제를 수행하고 소프트웨어 코딩 역량을 강화하는 데에 도움을 받을 수 있을 것이다. 또한, 교강사 및 조교 역시 학생들이 제출한 풀이 코드를 수합하고 채점하는 과정을 자동화하고, 테스트케이스 통과 여부뿐만 아니라 기능, 효율, 가독성으로 나누어 채점을 진행하여 학생들의 성취도 평가 및 수업 진행에 도움을 받을 수 있을 것이다.

### 2.2. PRODUCT FUNCTIONS

Table 3. Table of Product Functions

Function Requirement No.	Name of Function	Function Description
F1	문제 및 참조/제약사항 설명	문제와 참조/제약사항에 대한 설명을 제공한다.
F2	테스트케이스 설명	테스트케이스에 대한 설명을 제공한다.
F3	테스트케이스 검증/실행	테스트케이스를 검증하고 실행한 결과를 출력한다.



F4	코드 에디터 구현	코드 에디터를 제공한다.
F5	실시간 코드 저장	브라우저를 재실행해도 이전에 작성한 코드를 불러온다.
F6	코드 복사	코드 에디터에 작성한 코드를 복사할 수 있다.
F7	파일 불러오기	파일을 불러와 코드 에디터에 띄울 수 있다.
F8	코드 되돌리기	초기(skeleton code)의 코드로 되돌릴 수 있다.
F9	코드 중간 저장	지정한 횟수만큼 코드를 저장할 수 있다.
F10	테스트케이스 채점	히든 테스트케이스에 대한 내용 및 채점 결과를 제공한다.
F11	표절 검사	작성한 코드의 표절 검사 기능을 제공한다.
F12	기능 채점	코드의 기능에 대한 채점 기능을 제공한다.
F13	효율 채점	코드의 효율에 대한 채점 기능을 제공한다.
F14	가독성 채점	코드의 가독성에 대한 채점 기능을 제공한다.
F15	코드 Diff	정답 코드와 학생 코드 사이의 Diff 기능을 제공한다.
F16	코드 설명	정답 코드에 대한 설명을 자동으로 제공한다.
F17	관련 자료 추천	학생이 해결한 문제와 관련된 자료를 자동으로 제공한다.

### 2.2.1 MAIN PAGE

학생이 처음 시스템에 접속하면 ‘메인 페이지’에 도착한다.

헤드 영역의 과목명, 과제명을 선택하여 사용자가 풀고자 하는 과제를 선택하고 마감까지 남은 시간을 확인할 수 있다. 홈 버튼을 눌러 메인 페이지로 이동하거나 설정 버튼을 눌러 배경색 변경, 코드 에디터 변경 등 각종 설정을 진행할 수 있다.

메인 영역 중 좌측에는 사용자가 선택한 과제의 문제와 참조/제약사항, 테스트케이스를 출력한다. 이때 테스트케이스는 오픈, 히든으로 구분되고, 학생에게는 오픈 테스트케이스만 제공된다. 검증 버튼을 클릭하면 학생이 작성한 풀이 코드의 출력과 테스트케이스의 출력의 일치 여부에 따라 Pass/Fail 여부가 제공되고, Fail 인 경우 학생이 작성한 풀이 코드의 출력이 추가로 제공된다. 중앙에는 문제의 풀이 코드를 적을 수 있는 코드

에디터가 제공되고, 작성하고 있는 코드를 최대 세 번 저장할 수 있다. 하단 톨바의 버튼들을 클릭하여 파일 불러오기, 스켈레톤 코드로 코드 초기화, 작성 중인 코드 복사, 코드 다운로드, 코드 실행, 코드 채점, 코드 제출 등의 기능을 실행할 수 있고, 제출은 최대 세 번 할 수 있다. 우측에는 코드를 실행하거나 채점한 결과를 출력한다.

### 2.2.2. RUN RESULT PAGE

학생이 풀이 코드를 실행하면 ‘실행 결과’ 페이지에 도착한다.

메인 영역 중 우측에 풀이 코드를 실행한 결과를 출력한다. 실행 중 에러가 발생하지 않은 경우 출력값을 제공하고, 에러가 발생한 경우 에러가 발생한 위치를 코드 에디터에 하이라이팅하고 에러 메시지를 제공한다.

### 2.2.3. GRADING RESULT PAGE

학생이 풀이 코드를 채점하면 ‘채점 결과’ 페이지에 도착한다.

메인 영역 중 우측에 풀이 코드를 채점한 결과를 출력한다. 채점 결과는 총점과 각 테스트케이스에 대한 실행 결과로 구성되어 있다. 각 테스트케이스에 대한 실행 결과는 Pass/Fail 여부로 제공되고, 이때 테스트케이스는 오픈, 히든 테스트케이스를 모두 사용한다. 오픈 테스트케이스가 Fail 한 경우 테스트케이스 정보를 제공하고, 히든 테스트케이스가 Fail 한 경우 테스트케이스 정보를 제공하지 않는다.

### 2.2.4. SUBMIT & RESULT ANALYSIS PAGE

학생이 풀이 코드를 성공적으로 제출하면 ‘제출 및 결과 분석’ 페이지에 도착한다.

메인 영역 중 중앙에 코드 에디터 대신 사용자가 지금까지 제출한 세 개의 풀이 코드를 정답 코드와 비교한 Diff 결과를 출력한다. 이때, Diff 결과는 학생의 조작에 따라 영역의 너비를 넓혀 좌우로 표기할 수 있다. 우측에는 풀이 코드를 제출한 후 채점 결과, 코드 설명, 관련 자료를 제공한다. 채점 결과는 기능 점수, 효율 점수, 가독성 점수와 이를 종합한 총점으로 나누어 제공되고, 관련 자료는 문제와 관련된 추천 영상 및 학습 자료를 제공한다.

채점 및 결과 분석을 위해 다양한 공개 라이브러리를 활용할 수 있다.

## 2.3. USER CLASSES AND CHARACTERISTICS

### 2.3.1. SYSTEM ADMINISTRATOR

‘시스템 관리자’는 이 시스템을 운영 및 관리하고 학생, 교강사 및 TA 등의 사용자로부터 추가되는 요구사항을 처리하는 사람을 의미한다. 시스템 관리자는 시스템을 원활하게 운영 및 관리하기 위해 Django 기반 백엔드, React 기반 프론트엔드 웹 서비스를 개발할 수 있고, 백엔드와 프론트엔드 사이의 통신에 사용되는 HTTP 프로토콜 기반의 REST API, 데이터베이스에 사용되는 SQLite에 대한 이해 및 활용 역량을

가지고 있어야 한다. 뿐만 아니라 이 시스템이 제공하는 프로그래밍 언어인 Python에 대한 기본적인 지식을 가지고 있어야 한다.

### 2.3.2. STUDENT

‘학생’은 수강하는 과목의 과제를 이 시스템을 통해 진행하는 사람을 의미한다. 학생은 Chrome과 같은 모던 웹 브라우저를 통해 이 시스템에 접속하고, 기본적인 GUI 및 코드 에디터를 활용할 수 있어야 한다. 또한, 원활한 과제 수행을 위해 Python에 대한 기본적인 지식을 가지고 있어야 한다.

### 2.3.3. LECTURER & TA

‘교강사 또는 TA’는 과목의 과제를 이 시스템을 통해 출제하는 사람을 의미한다. 교강사 또는 TA 역시 Chrome과 같은 모던 웹 브라우저를 통해 이 시스템에 접속하고, 기본적인 GUI를 활용할 수 있어야 한다.

## 2.4. OPERATING ENVIRONMENT

- Linux
- Mac OS
- Windows 10
- Windows 11

## 2.5. DESIGN AND IMPLEMENTATION CONSTRAINTS

본 아래의 제약 사항들을 준수하여 개발, 운영, 사용되어야 한다.

- 이미 널리 성능이 입증된 기술을 사용한다.
- 가능한 공개된 오픈소스를 사용한다.
- 시스템의 성능을 저하시키지 않는 방향으로 설계하고 구현한다.
- UI 설계 시 사용자의 입장에서 받아들이기 쉬운 방향으로 결정한다.
- 사용자에게 악영향을 끼칠 가능성이 있는 코드를 포함하지 않는다.
- 사용자가 본 시스템 외의 다른 제품을 설치할 필요 없이 시스템을 이용할 수 있도록 설계하고 구현한다.
- 시스템이 사용된 이후의 유지보수를 고려하여 설계하고 구현한다.

## 2.6. ASSUMPTIONS AND DEPENDENCIES

본 문서의 시스템은 웹 기반으로 설계 및 구현되므로 사용자가 웹에 접속할 수 있는 환경 하에서 서비스를 사용할 것이라 가정한다. 따라서 웹의 불안정성으로 인해 야기되는 문제에 대해 고려하지 않는다. 위의 2.4. Operating Environment 항목에서 기술한 조건을 만족하는 환경에서의 사용자 접근을 가정하고 그에 한해서 정상적 작동을 보장한다.

### 3. EXTERNAL INTERFACE REQUIREMENTS

#### 3.1. USER INTERFACE

Table 4. User Interface of Main Page

이름	메인 페이지
목적/설명	<p>사용자가 시스템에 접근하면 볼 수 있는 초기 상태의 메인 페이지이다.</p> <p>메인 페이지 헤드 섹션의 과목명, 과제명을 선택하여 사용자가 필요한 과제를 확인하고 코드 입력 창에 코드를 작성한다. 화면의 각 버튼을 누름으로써 작성한 코드를 저장, 채점, 제출할 수 있다.</p>
입력 소스/ 출력 대상	<p>사용자/시스템 서버</p> <p>시스템 서버/사용자</p>
범위/ 정확도/ 오차범위	<p>한 과제에 대한 페이지</p> <p>Html, 데이터 처리 정확도</p>
단위	웹 페이지
시간/속도	사용자와 서버간 통신 시간
기타 입출력관계	N/A
화면 형식과 구성	<p>1. 헤드 섹션 왼쪽의 홈 버튼을 클릭함으로써 메인 페이지로 초기화된다.</p> <p>헤드 섹션에 과목명과 과제명이 표시되며, 사용자가 알맞은 과목, 과제를 선택하면 그에 맞는 문제 내용이 표시된다.</p> <p>헤드 섹션 오른쪽에는 과제 기한과 설정 버튼이 있어 설정 버튼을 클릭하면 메인 페이지의 배경 색상을 변경하거나 코드 에디터를 변경할 수 있다.</p> <p>2. 메인 페이지 왼쪽에는 사용자가 선택한 과제에 맞는 문제 내용과 참조, 제약사항이 표시된다.</p> <p>3. 문제&amp;참조/제약사항 아래에 해당 문제의 공개 테스트케이스가 표시된다. 각 테스트케이스의 검증 버튼을 누르면 사용자가 작성한 코드에 대해 해당</p>

	<p>테스트케이스의 input 을 입력으로 하는 출력을 표시한다.</p> <p>4. 메인 페이지 가운데에는 사용자가 코드를 입력할 수 있는 코드 입력창이 표시된다.</p> <p>코드 입력창에 있는 저장 버튼을 눌러 사용자가 작성중인 코드를 임시 저장할 수 있다.</p> <p>코드 입력창의 아래에 각 기능 버튼이 있어 이 버튼을 누르면 파일 불러오기, 코드 초기화, 코드 복사, 코드 다운로드 기능이 실행된다.</p> <p>5. 그 옆의 실행 버튼을 누르면 사용자가 작성한 코드를 실행하고 코드 입력창의 오른쪽에 그 실행 결과를 표시한다.</p> <p>채점 버튼을 누르면 사용자가 작성한 코드에 공개된 테스트케이스와 비공개 테스트케이스로 해당 코드를 채점한다. 그 채점 결과를 코드 입력창의 오른쪽에 표시한다.</p> <p>제출 버튼을 누르면 사용자가 작성한 코드를 제출한다. 이 기능은 세 번까지 가능하며 코드 제출 결과를 보여주는 페이지로 변경된다.</p>
데이터 유형	텍스트
명령 유형	버튼 클릭에 의한 코드 실행
종료 메시지	N/A

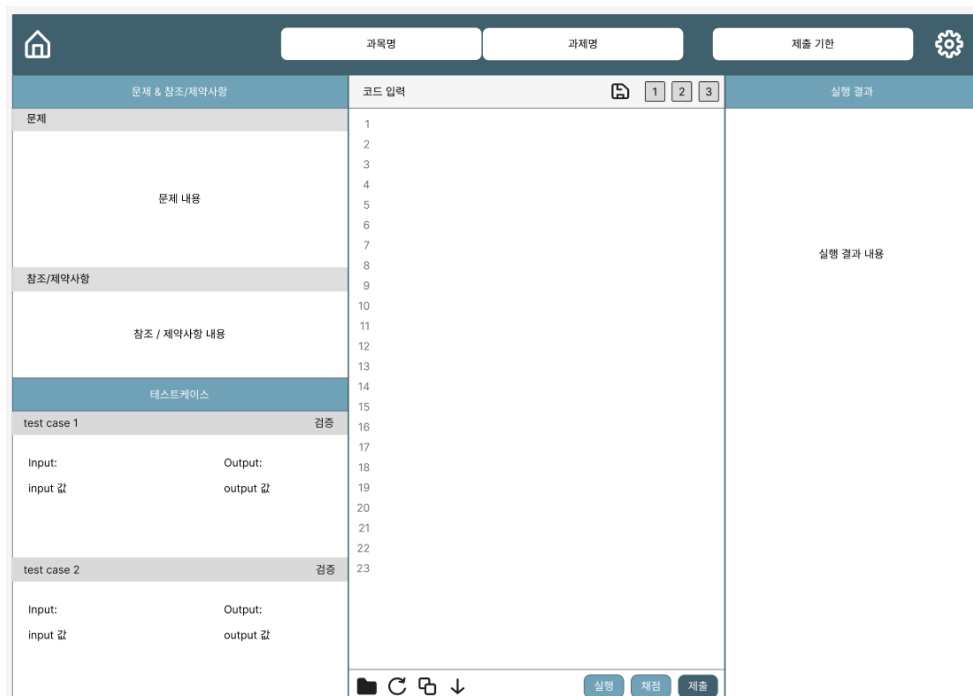


Figure 1. Design of Main Page

Table 5. User Interface of Main Page – Result for Run

이름	메인 페이지-실행 결과
목적/설명	사용자가 메인 페이지의 코드 입력창에 코드를 작성하고 코드 입력창의 아래에 있는 실행 버튼을 누르면, 해당 코드를 실행하여 그 결과를 메인 페이지 오른쪽에 표시한다.
입력 소스/ 출력 대상	사용자/시스템 서버 시스템 서버/사용자
범위/ 정확도/ 오차범위	한 과제에 대한 페이지  Html, 데이터 처리 정확도
단위	웹 페이지
시간/속도	사용자와 서버간 통신 시간
기타 입출력관계	N/A
화면 형식과 구성	<ol style="list-style-type: none"> <li>1. 코드 실행 결과 표시되는 화면은 메인 페이지와 동일하다.</li> <li>2. 실행 결과창에 사용자가 작성한 코드의 실행 결과를 표시한다. 실행 결과 에러가 아닌 경우 출력 값을 제공한다.</li> </ol> <p>실행 결과 에러인 경우 실행 결과창에 에러 메시지를 제공하고 코드 입력창에 해당 에러가 생긴 라인을 하이라이트해 보여준다.</p>
데이터 유형	텍스트
명령 유형	버튼 클릭에 의한 코드 실행
종료 메시지	N/A

Table 6. User Interface of Main Page – Result of Grading

이름	메인 페이지-채점 결과
목적/설명	사용자가 메인 페이지의 코드 입력창에 코드를 작성하고 코드 입력창의 아래에 있는 채점 버튼을 누르면, 해당 코드를 실행하여 문제에 속한 테스트케이스를 입력으로 하는 출력 값을 확인함으로써 코드를 채점한 결과를 실행 결과창에 표시한다.
입력 소스/ 출력 대상	사용자/시스템 서버 시스템 서버/사용자
범위/	한 과제에 대한 페이지

정확도/ 오차범위	Html, 데이터 처리 정확도
단위	웹 페이지
시간/속도	사용자와 서버간 통신 시간
기타 입출력관계	N/A
화면 형식과 구성	<p>1. 코드 채점 결과 표시되는 화면은 메인 페이지와 동일하다.</p> <p>2. 과제에 해당하는 공개 테스트케이스와 비공개 테스트케이스의 input 을 입력으로 했을 때 출력 값을 테스트케이스의 output 값과 비교하여 동일한 개수로 점수를 제공한다.</p> <p>각 테스트케이스에 통과했는지 여부를 표시한다.</p> <p>채점 결과 실패일 경우, 공개 테스트케이스는 그 정보를 표시하고 비공개 테스트케이스에 대해서는 표시하지 않는다.</p>
데이터 유형	텍스트
명령 유형	버튼 클릭에 의한 코드 실행
종료 메시지	N/A

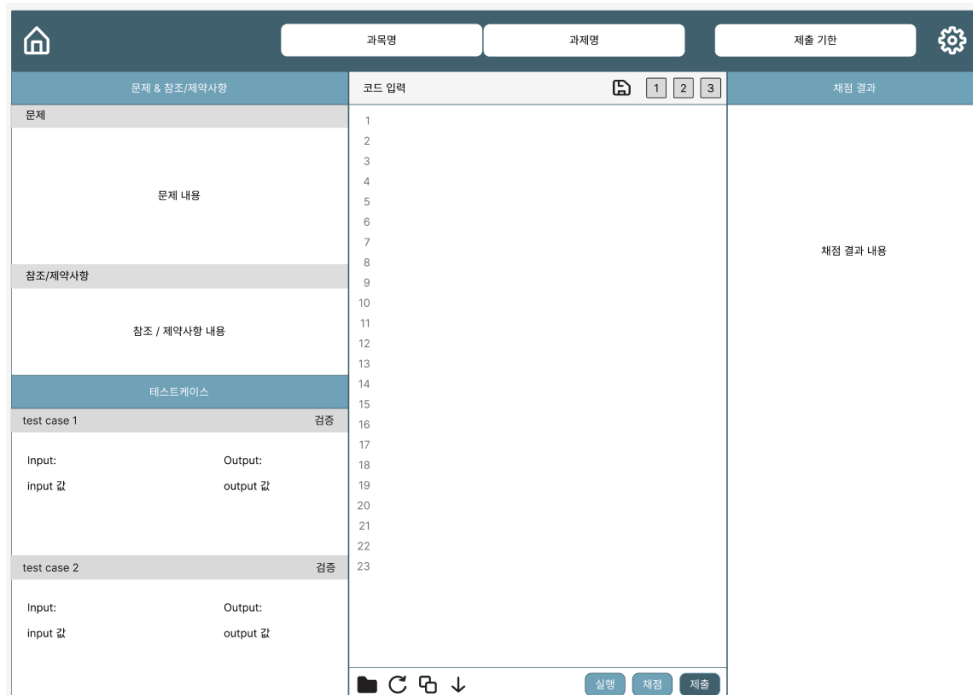


Figure 2. Design of Main Page – Result of Grading

Table 7. User Interface of Main Page – Result of Submit

이름	메인 페이지-제출 결과
목적/설명	사용자가 메인 페이지의 코드 입력창에 코드를 작성하고 코드 입력창의 아래에 있는 제출 버튼을 누르면, 코드에 대한 채점 결과와 코드 설명, 관련 자료, 정답 코드와의 비교 정보를 표시한다.
입력 소스/ 출력 대상	사용자/시스템 서버 시스템 서버/사용자
범위/ 정확도/ 오차범위	한 과제에 대한 페이지  Html, 데이터 처리 정확도
단위	웹 페이지
시간/속도	사용자와 서버간 통신 시간
기타  입출력관계	N/A
화면 형식과 구성	<p>1. 코드 제출 결과 표시되는 화면은 메인 페이지와 동일하다.</p> <p>2. 코드 입력창의 오른쪽에 제출 결과, 코드 설명, 관련 자료 창이 표시된다. 제출 결과 화면이 기본적으로 제공되며, 아래의 코드 설명 탭을 클릭하면 코드 설명과 관련 자료가 표시된다.</p> <p>제출 결과창에는 코드에 대한 기능 점수, 효율 점수, 가독성 점수와 이를 합한 총점을 표시한다. 각 점수의 세부 사항을 클릭하여 그 내용을 확인할 수 있으며 표절을 또한 표시된다.</p> <p>코드 설명 탭을 클릭하면 해당 과제에 대한 코드 설명과 관련 자료 내용이 표시된다.</p> <p>3. 과제에 등록된 정답 코드와 사용자가 제출한 코드를 비교하는 code diff를 수정 방안으로 변경된 코드 입력창에 표시한다. 특정 버튼을 누르면 code diff를 표시하는 수정 방안 창을 확대하고 다시 축소할 수 있다.</p> <p>사용자가 작성한 코드와 정답 코드를 비교하여 다른 부분에 하이라이트를 표시한다.</p>
데이터 유형	텍스트, 그래프
명령 유형	버튼 클릭에 의한 코드 실행
종료 메시지	N/A



문제 & 참조/제약사항		수정 방안	제출 기한
문제	문제 내용	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23	제출 결과
참조/제약사항	참조 / 제약사항 내용		제출 결과 내용
테스트케이스	test case 1 검증		
	Input: input 값      Output: output 값		
	test case 2 검증		
	Input: input 값      Output: output 값		
			코드 설명
			관련 자료

Figure 3. Design of Main Page – Default page for Result of Submit

수정 방안		제출 기한
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23	제출 결과
		제출 결과 내용
		코드 설명
		관련 자료

Figure 4. Design of Main Page – Result of Submit, Extended Code Editor

Figure 5. Design of Main Page – Result of Submit, Code Description and Related Materials

### 3.2. HARDWARE INTERFACE

Table 8. Hardware Interface for User

이름	사용자가 시스템을 실행하는 기기
목적/설명	사용자가 시스템의 웹페이지에 접근하고 해당 시스템을 사용할 수 있도록 한다.

### 3.3. SOFTWARE INTERFACES

Table 9. software interface of db

이름	SQLite
목적/설명	사용자 및 문제 정보를 입출력하기 위한 쿼리 데이터
입력 소스/ 출력 대상	클라이언트/호스트서버 호스트서버/클라이언트
범위/ 정확도	SQLite 의 정확도
구성 단위	쿼리
시간/속도	즉시 반응

기타 입출력과 관계	서버의 모든 입출력 데이터와 연관
화면 형식과 구성	N/A
데이터 유형	SQLite 쿼리
명령 유형	SQLite 명령어
종료 메시지	N/A

### 3.4. COMMUNICATION INTERFACES

Table 10. Communication Interface of Web browser

이름	웹 브라우저
목적/설명	시스템 사용자는 웹 브라우저를 통해 본 서버와 통신하며 사이트에서 제공하는 기능을 사용할 수 있다.
입력 소스/ 출력 대상	클라이언트/호스트서버 호스트서버/클라이언트
범위/ 정확도	해당 프로토콜의 정확도
구성 단위	패킷
시간/속도	네트워크와 해당 프로토콜 알고리즘의 속도
기타 입출력과 관계	사용자와 서버 간 모든 데이터 입출력
화면 형식과 구성	N/A
데이터 유형	웹 브라우저가 사용하는 프로토콜 데이터
명령 유형	해당 프로토콜의 명령
종료 메시지	N/A

## 4. SYSTEM REQUIREMENTS

### 4.1. FUNCTIONAL REQUIREMENTS

#### 4.1.1. USE CASE

Table 11. Functional requirements abstraction

Functional Requirement No.	Name of Function	Function Description
F1	문제 정보 불러오기	사용자가 선택한 문제의 정보를 불러올 수 있어야 한다.
F2	코드 작성 및 실행	사용자가 코드를 작성 및 실행할 수 있어야 한다.
F3	테스트 케이스 검증	사용자가 작성한 코드를 hidden 이 아닌 테스트 케이스로 검증할 수 있어야 한다.
F4	코드 저장 및 불러오기	사용자가 코드를 저장하고 불러올 수 있어야 한다.
F5	코드 초기화	사용자가 코드를 skeleton 으로 초기화할 수 있어야 한다.
F6	코드 복사	사용자가 작성중인 코드를 클립보드에 복사할 수 있어야 한다.
F7	코드 다운로드	사용자가 작성한 코드를 파일로 다운로드 할 수 있어야 한다.
F8	코드 제출	사용자가 작성한 코드를 최대 세 번 까지 제출할 수 있어야 한다.
F9	코드 채점 결과 확인	사용자가 작성한 코드를 채점하고 결과를 볼 수 있어야 한다.
F10	코드 분석 결과 확인	사용자가 제출한 코드를 분석한 결과를 볼 수 있어야 한다.
F11	관련 자료로 이동	사용자가 관련 자료에 있는 외부 링크로 이동할 수 있어야 한다.

Table 12. Use Case of Question Information

Name	Question Information
Actors	학생
Description	사용자가 선택한 문제 정보를 불러온다.
Data	Question
Stimulus	사용자가 헤드 섹션에서 화살표 버튼을 누른다.
Response	문제 본문과 참조, 제약사항, 테스트 케이스에 대한 정보가 바뀐 문제에 맞춰 브라우저에 표시된다.
Comments	

Table 13. Use Case of Code Edit and Run

Name	Code Edit and Run
Actors	학생
Description	사용자가 코드를 작성하고 실행한다.
Data	Code_Saved
Stimulus	사용자가 코드 에디터 섹션에서 코드를 작성한다. 사용자가 각종 기능 버튼 섹션에서 실행 버튼을 누른다.
Response	코드를 작성하면 실시간으로 브라우저에 저장된다. 코드를 실행한 결과의 출력값이 실행 결과 창에 보여진다. 실행 결과가 에러일 경우 에러 메시지에 따른 위치를 확인할 수 있으며 에러 라인 하이라이트 및 에러 메시지가 제공된다.
Comments	

**Table 14. Use Case of Test Case Check**

Name	Test Case Check
Actors	학생
Description	사용자가 작성한 코드를 hidden 이 아닌 테스트 케이스로 검증한다.
Data	Testcase
Stimulus	사용자가 테스트 케이스 섹션에서 검증 버튼을 누른다.
Response	통과시에 pass 가 출력된다. 실패시에 fail 과 학생의 잘못된 output 이 출력된다.
Comments	

**Table 15. Use Case of Code Save and Load**

Name	Code Save and Load
Actors	학생
Description	사용자가 코드를 저장하고 불러온다.
Data	Code_Saved
Stimulus	사용자가 코드 에디터 섹션에서 저장 버튼을 누른다. 사용자가 코드 에디터 섹션에서 번호 버튼을 누른다.
Response	사용자의 코드가 서버에 저장된다. 서버에 저장된 코드가 불러오기된다.
Comments	

**Table 16. Use Case of Code Reset**

Name	Code Reset
------	------------

Actors	학생
Description	사용자가 코드에디터를 skeleton 으로 초기화 한다.
Data	Question
Stimulus	사용자가 각종 기능 버튼 섹션에서 초기화 버튼을 누른다.
Response	코드 에디터가 skeleton 코드로 초기화된다.
Comments	

**Table 17. Use Case of Code Copy**

Name	Code Copy
Actors	학생
Description	사용자가 작성한 코드를 clipboard 에 복사한다.
Data	Code_Saved
Stimulus	사용자가 각종 기능 버튼 섹션에서 복사 버튼을 누른다.
Response	작성한 코드가 clipboard 로 복사된다.
Comments	

**Table 18. Use Case of Code Download**

Name	Code Download
Actors	학생
Description	사용자가 작성한 코드를 파일로 다운로드한다.
Data	Code_Saved

Stimulus	사용자가 각종 기능 버튼 섹션에서 다운로드 버튼을 누른다.
Response	작성한 코드가 다운로드 폴더에 파일로 다운로드된다.
Comments	

**Table 19. Use Case of Code Submit**

Name	Code Submit
Actors	학생
Description	사용자가 작성한 코드를 최대 세 번까지 제출한다.
Data	Code_Submitted
Stimulus	사용자가 각종 기능 버튼 섹션에서 제출 버튼을 누른다.
Response	제출 횟수가 남았다면 사용자가 작성한 코드가 제출되어 서버에 저장되고 제출 횟수를 하나 줄인다.  제출 횟수가 전부 사용되었다면 제출 횟수 초과 메시지를 출력한다.
Comments	

**Table 20. Use Case of Code Grading**

Name	Code Grading
Actors	학생
Description	사용자가 작성한 코드를 채점한 결과를 확인한다.
Data	Testcase, Code_Saved
Stimulus	사용자가 각종 기능 버튼 섹션에서 채점 버튼을 누른다.
Response	작성한 코드가 채점되고, 채점한 결과가 출력된다.



Comments	

**Table 21. Use Case of Code Analysis**

Name	Code Analysis
Actors	학생
Description	사용자가 제출한 코드를 분석한 결과를 확인한다.
Data	Code_Submitted
Stimulus	사용자가 각종 기능 버튼 섹션에서 제출 버튼을 누른다. 사용자가 제출했던 문제를 다시 불러온다.
Response	제출 결과 창에 코드의 분석 결과가 표시된다.
Comments	

**Table 22. Use Case of Reference Material**

Name	Reference Material
Actors	학생
Description	사용자가 관련 자료를 보고 외부 링크로 이동한다.
Data	Question
Stimulus	사용자가 관련 자료 섹션에서 외부 링크를 누른다.
Response	외부 링크로 이동한다.
Comments	외부 링크가 정상 작동해야 한다.

## 4.1.2. USE CASE DIAGRAM

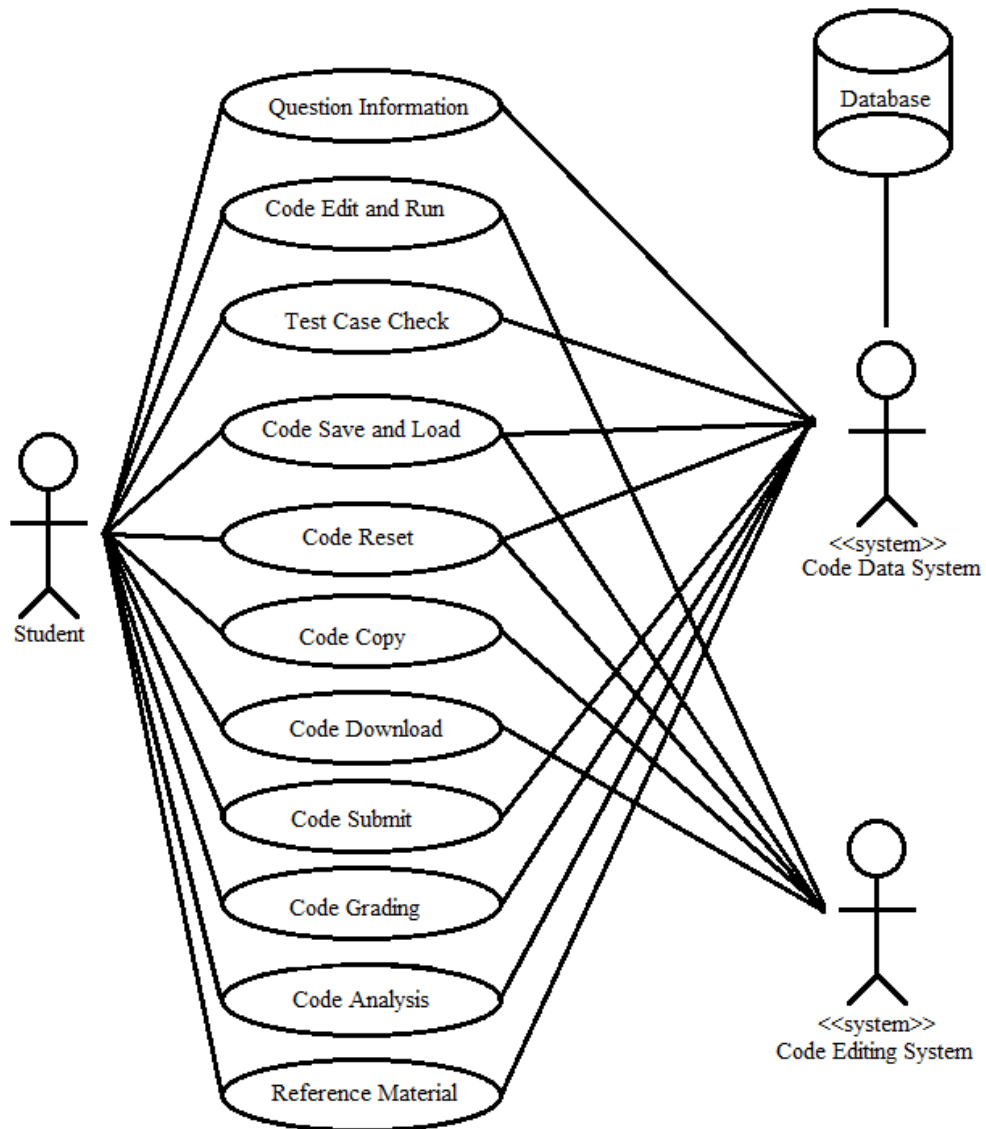


Figure 6. Use Case Diagram

### 4.1.3. DATA FLOW DIAGRAM

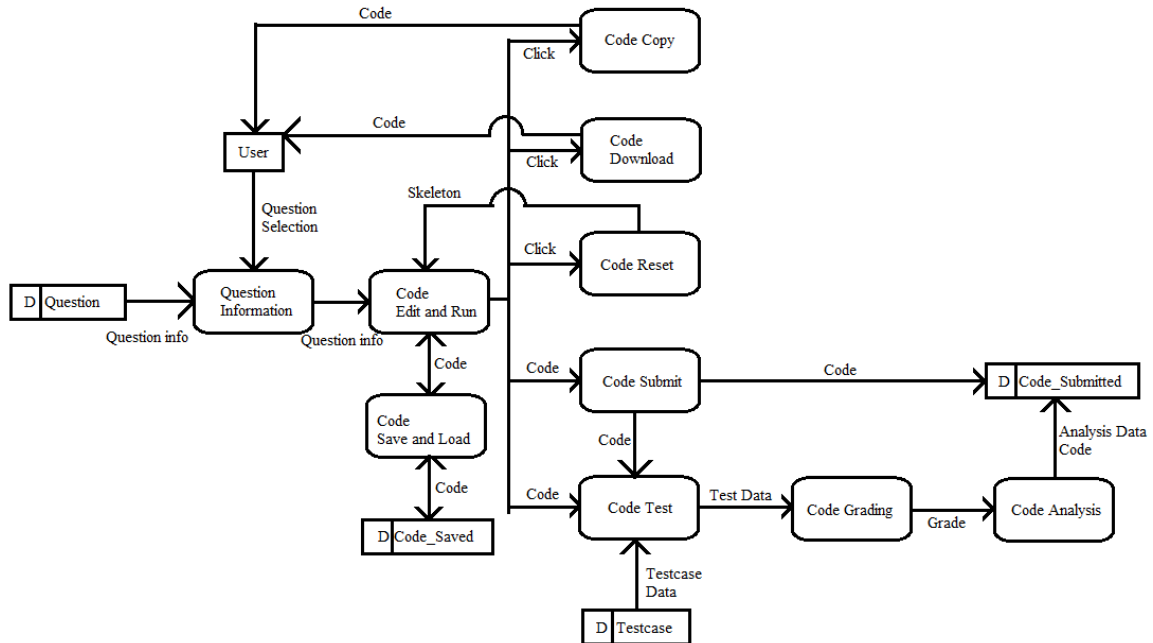


Figure 7. Data Flow Diagram

## 4.2. NON FUNCTIONAL REQUIREMENTS

시스템이 제공하는 서비스나 시스템 속성 및 제약 조건을 정의한다.

### 4.2.1. PRODUCT REQUIREMENT

특정한 방식으로 작동해야 하는 것을 명시하는 요구사항이다.

#### A. USABILITY REQUIREMENT

시스템은 코딩을 공부하는 학생 수준에서 따로 훈련하지 않거나 간단한 한 페이지의 메뉴얼을 읽고 모든 기능을 정상적으로 사용할 수 있어야 한다. 이후 사용에서의 오류율은 5% 미만이어야 한다.

#### B. PERFORMANCE REQUIREMENT

- 사용자와 DB 간 코드만 주고받는 간단한 상호작용은 1 초 이내로 되어야 한다.
- 테스트케이스를 검증하는 과정은 3 초 이내로 완료되어야 한다.
- 코드를 채점하고 결과를 보여주는 과정은 5 초 이내로 완료되어야 한다.

- 코드의 분석 결과를 보여주는 과정은 제출 시 10 초 이내, 이전 결과를 불러올 시 3 초 이내로 완료되어야 한다.

### C. DEPENDABILITY REQUIREMENT

사용자가 생성하는 모든 데이터는 오류 없이 그대로 DB에 저장되어야 한다. 제출한 코드에 대한 정확한 채점 및 분석이 이루어져야 한다.

## 4.2.2. ORGANIZATIONAL REQUIREMENT

사용하는 프로세스 표준, 구현 요구사항 등과 같이 조직의 정책 및 절차의 결과로 발생하는 요구사항이다.

### A. ENVIRONMENTAL REQUIREMENT

웹페이지 디자인을 고려하면 모바일 기기보다 PC를 이용하여 접속할 것이 권장된다.

### B. DEVELOPMENT REQUIREMENT

Front End의 경우 React로 제작하며 Figma로 디자인을 관리한다. Back End의 경우 Django로 제작하며 SQLite로 DB를 관리한다. 또한 정해진 코드 컨벤션을 따라 코드를 작성하는 것이 권장된다. PC 환경의 사용자를 우선시하여 개발을 진행한다.

## 4.2.3. EXTERNAL REQUIREMENT

시스템 및 개발 프로세스의 외부 요인에서 파생된 모든 요구사항을 다룬다.

### A. ETHICAL REQUIREMENT

문제를 외부에서 가져와서 쓰게 된다면 저작권에 위배되지 않게 써야 할 것이다. 문제와 관련된 자료를 추천할 때 출처를 명시하고 저작권을 확인해야 한다.

### B. SAFETY/SECURITY REQUIREMENT

사용자가 작성한 코드도 하나의 저작물이므로 외부로 유출되지 않게 보호해야 한다.

### 4.3. ORGANIZING SYSTEM FLOW

#### 4.3.1. CONTEXT MODEL

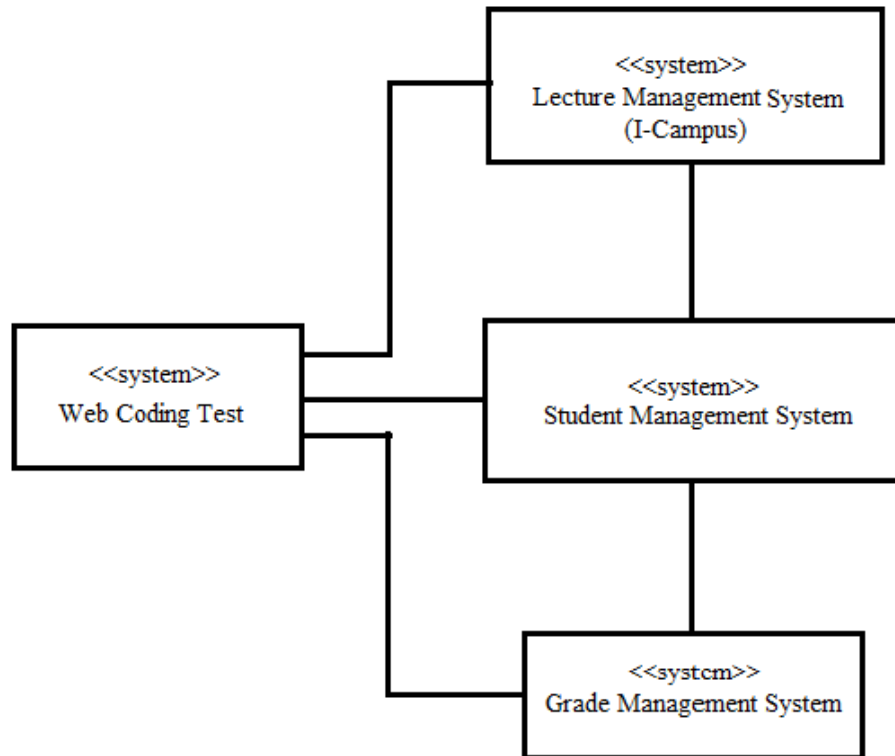


Figure 8. Context Model

### 4.3.2. PROCESS MODEL

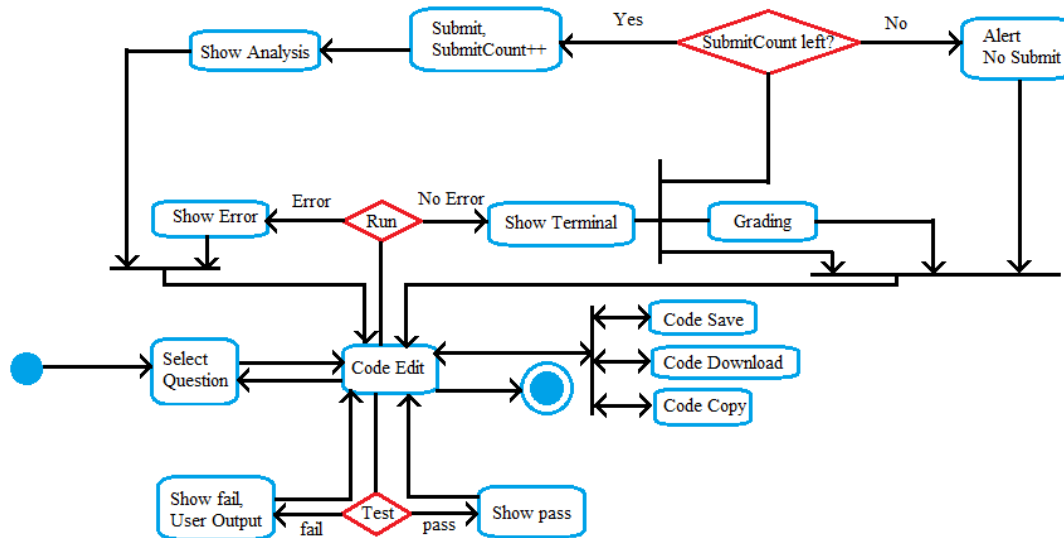


Figure 9. Process Model

### 4.3.3. INTERACTION MODEL

4.1.2. Use Case Diagram 참고

### 4.3.4. BEHAVIOR MODEL

#### A. DATA FLOW DIAGRAM

4.1.3. Data Flow Diagram 참고

#### B. SEQUENCE DIAGRAM

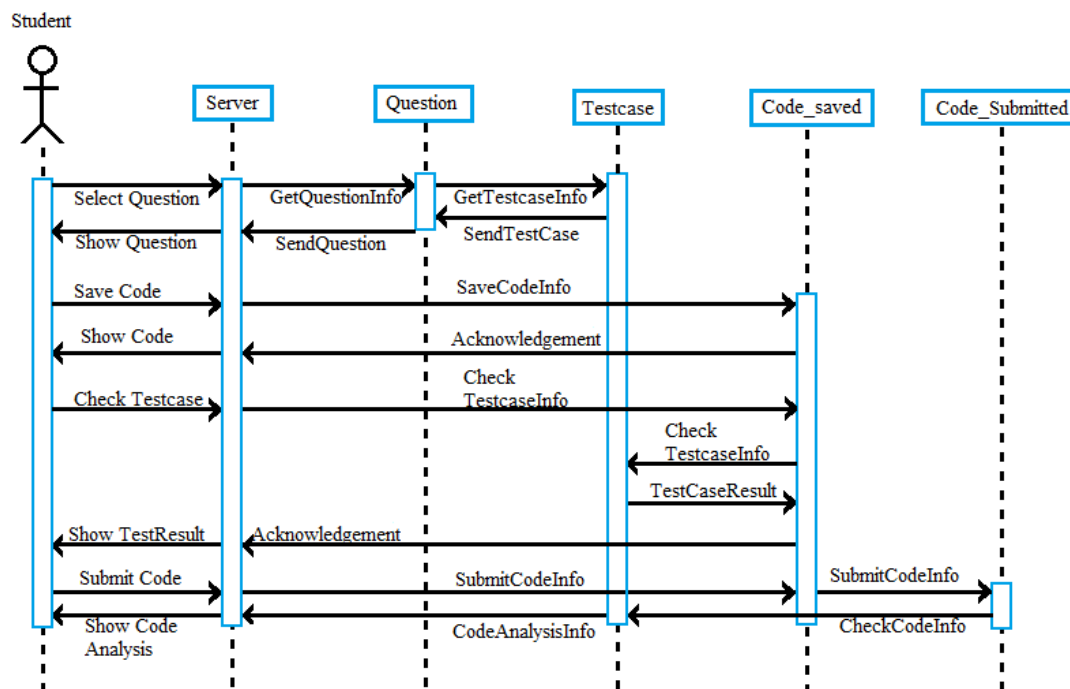


Figure 10. Sequence Diagram

#### 4.4. SYSTEM ARCHITECTURE

이 섹션에서는 예상되는 시스템 아키텍처에 대한 높은 수준의 개요를 제시하고 시스템 모듈 전반의 기능 분포를 보여준다.

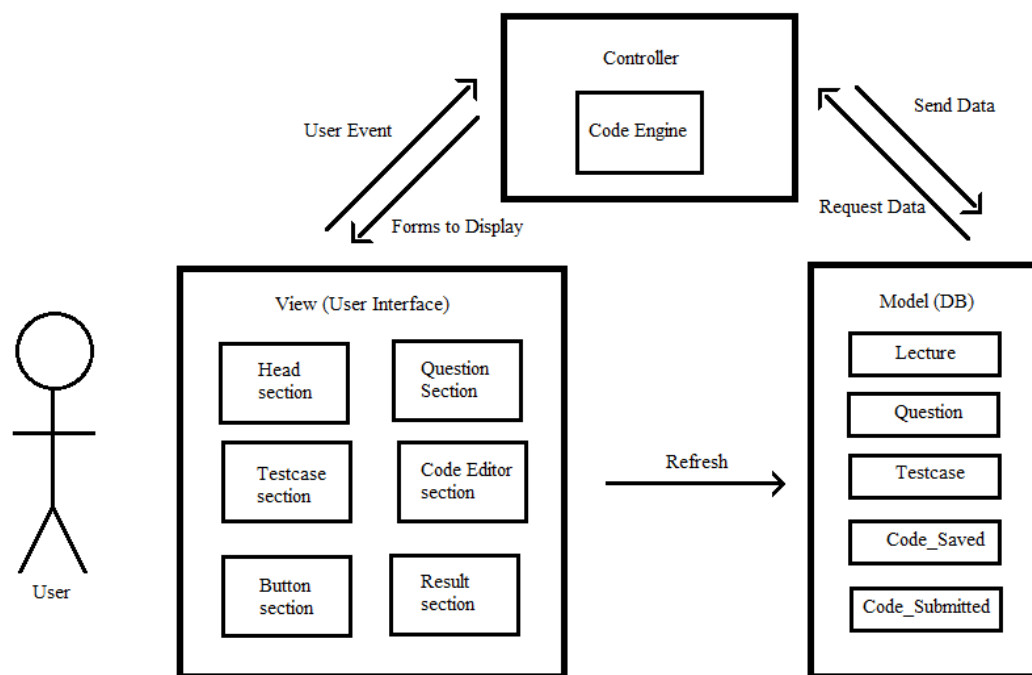


Figure 11. System Architecture