

# Software Requirements Specification bitCO2e

by 김영민, 박경린, 성주용, 송태현, 이찬



Instructor: 이은석  
Document Date: 05 May, 2024  
Faculty: SungKyunKwan University

## Document History

| Date           | Version | Description                                       | Writer        |
|----------------|---------|---|---------------|
| 29 April, 2024 | 1.1     | Introduction, Overall description 작성              | 송태현, 이찬       |
| 02 May, 2024   | 1.2     | External Interface Requirements 작성                | 김영민, 박경린, 성주용 |
| 04 May, 2024   | 1.3     | Introduction, Overall description 수정              | 김영민, 박경린      |
| 05 May, 2024   | 1.4     | External Interface Requirements 수정, Appendices 작성 | 성주용, 송태현, 이찬  |

# Contents

|   |           |
|---|-----------|
| <b>1. Introduction</b>                      | <b>7</b>  |
| 1.1. Purpose                                | 7         |
| 1.2. Scope                                  | 7         |
| 1.3. Constraints                            | 7         |
| 1.4. Definition, acronyms and abbreviations | 7         |
| 1.5. References                             | 8         |
| 1.6. Overview                               | 8         |
| <b>2. Overall Description</b>               | <b>8</b>  |
| 2.1. Product Perspective                    | 8         |
| 2.1.1. System Interfaces                    | 8         |
| 2.1.2. User Interfaces                      | 8         |
| 2.1.3. Hardware Interfaces                  | 9         |
| 2.1.4. Software Interfaces                  | 9         |
| 2.1.5. Memory Constraints                   | 9         |
| 2.1.6. Operations                           | 9         |
| 2.2. Product Functions                      | 9         |
| 2.2.1. 계정관리                                 | 9         |
| 2.2.2. Code refactoring                     | 9         |
| 2.2.3. Bit 획득                               | 10        |
| 2.2.4. 분석결과 확인                              | 10        |
| 2.2.5. 회원 나무 등급 확인                          | 10        |
| 2.2.6. 광고권 구매                               | 10        |
| 2.2.7. 광고 승인                                | 10        |
| 2.3. User Classes and Characteristics       | 10        |
| 2.3.1. 비회원                                  | 10        |
| 2.3.2. 회원                                   | 10        |
| 2.3.3. 관리자                                  | 11        |
| 2.4. Constraints                            | 11        |
| 2.5. Assumption and Dependencies            | 11        |
| <b>3. External Interface Requirements</b>   | <b>12</b> |
| 3.1. External Interfaces                    | 12        |
| 3.1.1. User Interface                       | 12        |
| 3.1.2. Hardware Interface                   | 13        |
| 3.1.3. Software Interface                   | 14        |
| 3.1.4. Communication Interface              | 15        |
| 3.2. Function requirement                   | 16        |
| 3.2.1. Use Case                             | 16        |
| 3.2.2. Use Case Diagram                     | 20        |
| 3.2.3. Data Dictionary                      | 21        |
| 3.2.4. Data Flow Diagram                    | 25        |

|        |                                      |           |
|--------|--------------------------------------|-----------|
| 3.3.   | Performance requirements             | 25        |
| 3.3.1. | Static Requirement                   | 25        |
| 3.3.2. | Dynamic Requirement                  | 26        |
| 3.4.   | Logical database requirements        | 26        |
| 3.5.   | Design constraints                   | 27        |
| 3.6.   | Software system attributes           | 27        |
| 3.6.1. | Reliability                          | 27        |
| 3.6.2. | Availability                         | 27        |
| 3.6.3. | Security                             | 27        |
| 3.6.4. | Maintainability                      | 27        |
| 3.6.5. | Portability                          | 27        |
| 3.7.   | Organizing the specific requirements | 27        |
| 3.7.1. | Process diagram                      | 27        |
| 4.     | <b>Appendices</b>                    | <b>28</b> |
| 4.1.   | Green Algorithms 탄소배출량 공식            | 28        |

## List of Figures

|                                      |    |
|--------------------------------------|----|
| [Figure 1] Use case diagram          | 20 |
| [Figure 2] Table of User             | 21 |
| [Figure 3] Table of Code Refactoring | 21 |
| [Figure 4] Table of UserBit          | 22 |
| [Figure 5] Table of UserTree         | 23 |
| [Figure 6] Table of Advertisement    | 23 |
| [Figure 7] ERD                       | 24 |
| [Figure 8] Data flow diagram         | 25 |
| [Figure 9] Process diagram           | 28 |

## List of Tables

|  |    |
|--|----|
| [Table 1] Table of definition, acronyms and abbreviations        | 8  |
| [Table 2] User interface of input processing                     | 12 |
| [Table 3] User interface of main page                            | 13 |
| [Table 4] Hardware interface of applicable device for the system | 14 |
| [Table 5] Software Interface - Website                           | 14 |
| [Table 6] Software Interface - Firebase Realtime Database        | 15 |
| [Table 7] Software Interface - Firebase Firestore                | 15 |
| [Table 8] Communication Interface                                | 16 |
| [Table 9] Use case of Register                                   | 17 |
| [Table 10] Use case of Registering User's code                   | 17 |
| [Table 11] Use case of Result statistics                         | 18 |
| [Table 12] Use case of System Information                        | 18 |
| [Table 13] Use case of Growing tree                              | 19 |
| [Table 14] Use case of Gaining bit                               | 19 |
| [Table 15] Use case of Buying advertisement                      | 19 |
| [Table 16] Use case of Approving advertisement                   | 20 |
| [Table 17] Description of table User                             | 21 |
| [Table 18] Description of table CodeRefactoring                  | 22 |
| [Table 19] Description of table UserBit                          | 22 |
| [Table 20] Description of table UserTree                         | 23 |
| [Table 21] Description of table Advertisement                    | 24 |
| [Table 22] Signs of the Green Algorithm's formula                | 29 |

## 1. Introduction

### 1.1. Purpose

이 프로젝트의 목표는 프로그램의 탄소배출량 감축을 목적으로, 개발자가 작성한 자바코드를 효율적이고 친환경적인 코드로 수정하는 것이다. 이 문서는 프로젝트의 requirements를 명확히 하여 서비스 개발을 돕고 추후 변경사항 발생 시 유지보수를 수월하게 하기 위해 작성되었다.

### 1.2. Scope

이 명세에서 개발할 서비스의 명칭은 'bitCO2e'이다. bitCO2e는 '적은'이라는 의미의 영단어 'bit'과 '이산화탄소환산량'을 의미하는 'CO2e'의 합성어로, '미래를 위한 작은 발걸음이 의미 있는 탄소배출량 차이를 만들어 낸다'는 의미를 갖는다.

bitCO2e에서는 사용자가 입력한 자바코드를 리팩토링하며 이를 통해 절감한 탄소배출량을 분석하여 사용자에게 제공한다. 또한 로그인한 회원에 한하여 포인트(bit)를 제공하며 회원은 bit를 소모하여 자신의 비영리적 광고를 웹사이트에 게시할 수 있다.

자바 이외의 프로그래밍 언어는 이 시스템의 범위에 포함되지 않는다.

### 1.3. Constraints

이 프로젝트는 2024년 6월 14일까지 완료되어야 한다.

### 1.4. Definition, acronyms and abbreviations

| Terms           | Definitions   |
|-----------------|---|
| refactoring     | refactoring은 소프트웨어 코드의 구조를 재조정하여 가독성, 유지보수성, 확장성을 향상시키는 과정을 의미한다.<br>bitCO2e에서는 탄소배출량을 감소시키는 방향으로 코드를 수정하는 것을 포함한다. |
| 탄소 배출량          | 특정 코드를 실행함으로 인해 배출되는 탄소의 양이다.   |
| code formatting | 코드의 형식을 일관되게 맞추는 과정으로서, 들여쓰기, 줄바꿈, 공백, 괄호, 주석 정리 등이 해당된다.   |
| 비영리성 광고         | 이익을 추구하는 기업이나 상품을 판매하는 광고와는 구분되는, 개인의 깃허브 계정과 같은 개인적인 영역에서 홍보할 수 있는 광고를 말한다.  |
| bit             | 광고권 구매 및 가상의 나무 키우기 등 시스템 내에서 회원을 대상으로 제공되는 서비스들을 이용하기 위한 일종의 포인트를 의미한다.  |
| 해싱              | 이 명세에서 사용되는 해싱은 각 사용자의 일반 텍스트 암호를   |

|  |   |
|--|---|
|  | 단방향 수학 함수를 통해 암호화하는 것을 의미한다. 해싱은 회원의 비밀번호를 데이터베이스에 안전하게 저장하기 위해 사용된다. |
|--|---|

[Table 1] Table of definition, acronyms and abbreviations

## 1.5. References

Green Algorithms, <https://www.green-algorithms.org/>

## 1.6. Overview

본 명세서는 크게 overall description과 requirements로 구성된다. overall description에서는 bitCO2e 서비스의 전반적인 기능과 일반적 특성, 사용자 유형, 가정과 제약사항을 기술한다. requirements는 external interfaces, function requirement, performance requirements, logical database requirements, design constraints, software system attributes로 구성된다. external interfaces에서는 bitCO2e 시스템의 입출력 방식에 대해 설명한다. function requirement는 bitCO2e 시스템이 어떤 기능을 수행하는지, input을 어떻게 처리하여 어떤 결과를 도출하는지 설명한다. performance requirements는 소프트웨어와 사용자간 정적 요구사항과 동적 요구사항을 기술한다. logical database requirements는 데이터베이스에 저장될 정보와 논리적 요구사항을 나타내며 design constraints는 시스템의 제약사항을, software system attributes는 bitCO2e 시스템의 일반적 특성을 제시한다.

## 2. Overall Description

### 2.1. Product Perspective

탄소중립적 코드 개선 플랫폼인 bitCO2e 시스템은 사용자의 Java 소스코드를 입력으로 받고 이를 탄소 중립적이고 환경 친화적인 방식으로 refactor하고 그 결과물을 분석 및 시각화 하여 출력하도록 설계되었다. 사용자는 시스템을 이용하여 자신의 코드를 refactor하는 과정을 통해 bit를 획득하고 이를 통해 가상의 나무를 키워 다른 사용자와 선의의 경쟁을 할 수 있고 또한 웹 사이트의 광고권을 획득해 자신의 홍보물을 게시하는 등 여러가지 이익을 취할 수 있다.

#### 2.1.1. System Interfaces

사용자는 refactoring 하고자 하는 Java 소스 코드를 시스템에 업로드 한다. 시스템은 그린화 패턴을 통해 입력된 소스 코드를 refactoring 하고 이를 분석한다. 사용자는 시스템으로부터 refactoring 된 코드와 절감된 탄소 배출량, bit 획득량 등 각종 분석 데이터들을 제공받는다.

#### 2.1.2. User Interfaces



인터페이스는 사용자 PC 화면을 통해 제공된다. 사용자는 로그인 및 회원가입을 진행할 수 있고 로그인 여부에 무관하게 시스템의 핵심 기능인 소스 코드 refactoring 및 결과 분석 기능을 이용할 수 있다. bit 획득 및 사용 등 일부 기능은 로그인 사용자를 대상으로만 제공된다.

#### 2.1.3. Hardware Interfaces

시스템은 웹 기반이므로 인터넷 연결이 가능한 컴퓨터를 통해 사용할 수 있다. 코드 입력 및 수정, 그에 대한 분석 및 refactoring 결과 출력, 기타 시스템 및 사용자 정보 등의 출력 등을 위해 필요한 최소한의 화면 크기 제약으로 인해 모바일 기기가 아닌 PC 환경만을 대상으로 한다.

#### 2.1.4. Software Interfaces

시스템은 Windows 10 이상의 OS를 대상으로 한다. 또한 Chrome 웹 브라우저를 통해 시스템을 이용할 것을 권장한다. 시스템의 정상적인 작동을 위해 사용자의 웹 브라우저에서 JavaScript 실행이 허용되어야 한다.

#### 2.1.5. Memory Constraints

32비트 시스템을 이용하는 경우 최소 1GB, 64비트 시스템을 이용하는 경우 최소 2GB의 RAM 용량이 필요하다.

#### 2.1.6. Operations

시스템 관리자로 로그인한 경우 사용자 광고 승인, 사용자 차단 등의 기능을 통해 시스템을 관리할 수 있다.

### 2.2. Product Functions

#### 2.2.1. 계정관리

계정 관리는 회원가입, 로그인 기능을 포함한다. 회원가입은 등록되지 않은 사용자가 bitCO2e 서비스에 자신의 계정 정보를 등록하는 기능이다.

로그인 기능은 코드 리팩토링 기능을 사용하는 사용자에게 bit를 부여하고 bitCO2e 웹사이트 내 광고를 게시할 수 있도록 사용자 식별 기능을 제공한다.

#### 2.2.2. Code refactoring

Code refactoring은 사용자가 작성한 자바 코드를 입력으로 받아 해당 코드를 탄소배출량을 감소시키는 최적화된 자바 코드로 변환해주는 기능이다. 프로그램 실행 시간과 메모리 사용량 등의 cost를 증가시키는 코드를 식별하여 수정한다. 또한 부가적으로 code formatting 등의 기능도 수행한다.

#### 2.2.3. Bit 획득

로그인한 사용자가 code refactoring 기능을 사용할 경우 bit를 획득할 수 있다. 획득한 누적 bit는 사용자의 탄소배출량 절감 랭킹으로 나타나며, bit를 사용해 웹사이트에 광고를 게시할 수 있다.

#### 2.2.4. 분석결과 확인

사용자가 code refactoring을 한 이후 탄소배출량과 관련된 분석 결과를 나타내는 기능이다. 사용자가 refactoring을 통해 감축한 탄소배출량을 여러 시각적 데이터로 표현한다.

#### 2.2.5. 회원 나무 등급 확인

사용자는 누적 bit를 통해 서비스 내 자신의 랭킹과 나무 등급을 확인할 수 있다. 누적 bit가 쌓일 수록 나무가 자라는 기능이다.

#### 2.2.6. 광고권 구매

사용자는 자신이 보유한 bit를 가지고 웹서비스에 비영리적 광고를 게재할 수 있다. 광고권은 일 단위로 구매할 수 있으며, 사용자는 규격에 맞는 광고 이미지를 업로드하고 관리자의 승인을 받아 웹사이트에 광고를 게재한다.

#### 2.2.7. 광고 승인

관리자는 사용자가 업로드한 광고 이미지를 manual하게 검수하고 광고 게재를 승인한다. 승인된 이미지는 다음날 00시부터 웹사이트 광고 배너에 게재된다.

### 2.3. User Classes and Characteristics

#### 2.3.1. 비회원

비회원은 bitCO2e 서비스에 회원가입하지 않고 서비스를 이용하는 사용자를 의미한다. bitCO2e 서비스는 회원 등록 절차 없이 refactoring, 분석결과 확인 등의 기능을 수행한다. 비회원은 code refactoring을 통한 bit를 획득할 수 없으며 광고 게시, 랭킹 및 나무 등급 확인 등의 회원 기능을 사용할 수 없다.

#### 2.3.2. 회원

회원은 bitCO2e 서비스에 회원가입한 등록된 사용자를 의미한다. 회원은 code refactoring을 통해 자신이 입력한 자바코드를 최적화하고 분석 결과를 확인할 수 있다. 등록된 회원은 이 과정에서 bit를 획득할 수 있다. 획득한 bit는 회원별 등급과 랭킹으로 표현된다. 또한 bit를 소모하여 자신의 비영리성 광고를 bitCO2e 웹사이트 광고배너에 게재할 수 있다.

### 2.3.3. 관리자

관리자는 bitCO2e 서비스를 제공하는 주체이다. 관리자는 bitCO2e 웹사이트 광고 배너에 표시될 광고를 승인하는 역할을 수행한다.

## 2.4. Constraints

시스템은 본 문서의 요구사항을 바탕으로 개발된다. 본 문서에 명시되지 않은 세부 사항들은 아래의 제약사항을 준수하는 범위 내에서 개발자의 능력과 역량, 재량에 따라 구현될 수 있다.

- Refactoring 대상 언어는 Java로 한다.
- 사용자 정보 보호를 위해 로그인 정보 등 개인 정보는 서버에서 암호화하여 저장하고 다룬다.
- 사용자 편의적이고 직관적인 UI를 지향한다.
- 사용자 수 증가, 지원 언어 증가, 기능 증가 등 추후 시스템의 확장 가능성을 염두에 둔다.
- 시스템 유지보수에 용이하도록 일관성과 가독성을 유지한다.
- 악의적 사용자로부터의 비정상적인 시스템 이용을 고려한다.

## 2.5. Assumption and Dependencies

시스템은 웹 기반으로 제공되는 서비스로, 사용자가 적절한 입출력 장치와 통신기능을 가진 기기를 통해 서비스를 이용할 것으로 가정한다. 따라서, 시스템이 제공하는 서비스 외 네트워크 연결 및 기타 주변 장치의 문제에 대해서는 고려하지 않는다. 기준이 되는 Windows 10 환경과 Chrome 브라우저의 최소 시스템 요구사항은 다음과 같다.

- (1) 1GHz 이상의 프로세서 (Core i3 또는 그 이상)
- (2) 1GB(32비트) 또는 2GB(64비트) 이상의 RAM
- (3) 16GB(32비트) 또는 32GB(64비트) 이상의 하드 드라이브 공간
- (4) DirectX 9 이상(WDDM 1.0 드라이버 포함) 호환 그래픽 카드
- (5) 800x600 이상의 디스플레이

시스템은 상기한 요구사항들을 만족하고 인터넷에 연결되어 있으며 마우스, 키보드 등의 적절한 HID를 갖춘 사용자의 접근을 가정하고, 그러한 경우에 올바른 작동을 보장한다.

### 3. Specific Requirements

#### 3.1. External Interface Requirements

##### 3.1.1. User Interface

| 이름           | 마우스 및 키보드를 통한 입력 처리   |
|--------------|---|
| 목적/내용        | 시스템 사용자가 키보드 및 마우스의 입력을 통해 시스템에 명령 전달                                       |
| 입력 주체/출력 목적지 | 사용자/Windows 기반의 컴퓨터 기기  |
| 범위/정확도/허용 오차 | 범위: 화면에서의 버튼의 개수에 따른 입력 범위<br>정확도: 유저의 마우스 및 키보드 입력에 따른 정확도<br>허용 오차: 해당 없음 |
| 단위           | 버튼 클릭/키보드 입력  |
| 시간/속도        | 비정기적인 사용자의 입력/즉각적인 사용자 명령 수행  |
| 타 입출력과의 관계   | 입력 내용에 따라 클라이언트에서 처리 또는 서버로 전달하여 처리   |
| 화면 형식 및 구성   | 해당 없음   |
| 윈도우 형식 및 구성  | 해당 없음   |
| 데이터 형식 및 구성  | Java 코드를 text로 입력   |
| 명령 형식        | Java 코드 입력, Refactoring 버튼 클릭   |
| 종료 메시지       | 해당 없음   |

[Table 2] User interface of input processing

| 이름           | 모니터를 통한 메인 화면 출력  |
|--------------|---|
| 목적/내용        | 사용자에게 제공하는 인터페이스  |
| 입력 주체/출력 목적지 | 클라이언트 / 사용자   |
| 범위/정확도/허용 오차 | 범위: 화면에서의 버튼의 개수에 따른 입력 범위<br>정확도: 유저의 마우스 및 키보드 입력에 따른 정확도<br>허용 오차: 해당 없음 |
| 단위           | 모니터 화면  |
| 시간/속도        | 사용자의 입력에 따른 화면 전환   |
| 타 입출력과의 관계   | 사용자의 입력을 위한 인터페이스로서 출력 후 사용자의 입력  |

|             |   |
|-------------|---|
|             | 대기  |
| 화면 형식 및 구성  | <ul style="list-style-type: none"> <li>- 개발할 웹사이트는 상단 nav에 서비스명과 로그인 정보 표기</li> <li>- (수정할 코드), (수정된 코드) 부분에서 사용자 코드를 입력받아 수정된 코드를 표시함</li> <li>- 코드가 실행된 시스템 정보와 분석 내용이 웹페이지 하단에 표시됨</li> <li>- 우측에는 광고 배너 위치</li> </ul> |
| 윈도우 형식 및 구성 | <ul style="list-style-type: none"> <li>- 3 Column Layout 형식으로 페이지를 구성</li> <li>- Java code를 입력하는 IDE창</li> <li>- 개선된 Java code를 출력하는 창</li> <li>- 코드가 동작된 시스템 정보와 분석결과를 표시하는 창</li> <li>- 광고 배너가 위치한 창</li> </ul>         |
| 데이터 형식 및 구성 | 이미지, 텍스트  |
| 명령 형식       | 해당 없음   |
| 종료 메시지      | 해당 없음   |

[Table 3] User interface of main page

### 3.1.2. Hardware Interface

| 이름           | 시스템에서 사용 가능한 디바이스     |
|--------------|-----------------------|
| 목적/내용        | 키보드, 마우스를 사용한 사용자의 입력 |
| 입력 주체/출력 목적지 | 사용자/서버                |
| 범위/정확도/허용 오차 | 해당 없음                 |
| 단위           | 해당 없음                 |
| 시간/속도        | 사용자의 입력/리팩토링에 해당하는 처리 |
| 타 입출력과의 관계   | 해당 없음                 |

|             |       |
|-------------|-------|
| 화면 형식 및 구성  | 해당 없음 |
| 윈도우 형식 및 구성 | 해당 없음 |
| 데이터 형식 및 구성 | 해당 없음 |
| 명령 형식       | 코드    |
| 종료 메시지      | 해당 없음 |

[Table 4] Hardware interface of applicable device for the system

### 3.1.3. Software Interface

| 이름           | Website            |
|--------------|--------------------|
| 목적/내용        | 화면 출력              |
| 입력 주체/출력 목적지 | 해당 없음              |
| 범위/정확도/허용 오차 | 웹 브라우저에서 사용 가능     |
| 단위           | 해당 없음              |
| 시간/속도        | 새로고침에 따른 즉각적인 처리   |
| 타 입출력과의 관계   | 해당 없음              |
| 화면 형식 및 구성   | 웹 브라우저를 통한 웹사이트 출력 |
| 윈도우 형식 및 구성  | 해당 없음              |
| 데이터 형식 및 구성  | 해당 없음              |
| 명령 형식        | 해당 없음              |
| 종료 메시지       | 해당 없음              |

[Table 5] Software Interface - Website

| 이름           | Firebase Realtime Database         |
|--------------|------------------------------------|
| 목적/내용        | 어플리케이션에서 생성되는 데이터를 저장하기 위한 데이터 저장소 |
| 입력 주체/출력 목적지 | 클라이언트                              |
| 범위/정확도/허용 오차 | 허용 오차 : 없음                         |

|             |                    |
|-------------|--------------------|
| 단위          | Row                |
| 타이밍         | 서버 입력에 따른 데이터 저장   |
| 타 입출력과 관계   | 사용자의 입력한 정보에 대한 서버 |
| 화면 형식 및 구성  | 해당 없음              |
| 윈도우 형식 및 구성 | 해당 없음              |
| 데이터 형식 및 구성 | Row                |
| 명령 형식       | 쿼리                 |
| 종료 메시지      | 해당 없음              |

[Table 6] Software Interface - Firebase Realtime Database

| 이름           | Firebase Firestore                     |
|--------------|--|
| 목적/내용        | 어플리케이션에서 생성되는 이미지 데이터를 저장하기 위한 데이터 저장소 |
| 입력 주체/출력 목적지 | 클라이언트                                  |
| 범위/정확도/허용 오차 | 허용 오차 : 없음                             |
| 단위           | jpg, png 등                             |
| 타이밍          | 서버 입력에 따른 데이터 저장                       |
| 타 입출력과 관계    | 사용자의 입력한 정보에 대한 서버                     |
| 화면 형식 및 구성   | 해당 없음                                  |
| 윈도우 형식 및 구성  | 해당 없음                                  |
| 데이터 형식 및 구성  | 이미지                                    |
| 명령 형식        | 쿼리                                     |
| 종료 메시지       | 해당 없음                                  |

[Table 7] Software Interface - Firebase Firestore

#### 3.1.4. Communication Interface

|    |            |
|----|------------|
| 이름 | 서버 - 클라이언트 |
|----|------------|

|              |  |
|--------------|--|
| 목적/내용        | 클라이언트에서 페이지를 불러오고, 사용자가 입력한 Java 코드를 서버로 전송<br>서버에서는 입력받은 코드를 리팩토링하여 개선된 코드, 분석결과를 클라이언트에 제공 |
| 입력 주체/출력 목적지 | 클라이언트 / 백엔드 서버<br>백엔드 서버 / 클라이언트   |
| 범위/정확도/허용 오차 | 허용 오차 : 없음   |
| 단위           | HTTP request/ HTTP response  |
| 시간/속도        | 클라이언트/서버와의 HTTP 통신 시간  |
| 타 입출력과의 관계   | 해당 없음  |
| 화면 형식 및 구성   | 해당 없음  |
| 윈도우 형식 및 구성  | 해당 없음  |
| 데이터 형식 및 구성  | JSON 구조로 데이터를 전송   |
| 명령 형식        | HTTP   |
| 종료 메시지       | 해당 없음  |

[Table 8] Communication Interface

### 3.2. Function requirement

#### 3.2.1. Use Case

| 이름            | Register  |
|---------------|---|
| Actor         | 등록되지 않은 사용자   |
| Description   | 등록되지 않은 사용자가 광고권 구매, 누적 bit 관리 등을 위해 회원으로 등록하는 과정이다.  |
| Normal Course | <ol style="list-style-type: none"> <li>1. 사용자는 웹사이트 접속 후 우측 상단에 위치한 로그인 버튼을 클릭한다.</li> <li>2. 사용자는 로그인 페이지로 이동한다.</li> <li>3. 하단에 있는 Register 버튼을 클릭하면 등록 페이지로 이동된다.</li> <li>4. 사용자는 등록 양식에 맞추어 정보를 입력한다. 필요한 정보는 다음과 같다 <ol style="list-style-type: none"> <li>a. ID</li> <li>b. pw</li> <li>c. pw확인</li> </ol> </li> <li>5. 양식을 작성한 후 사용자가 등록 버튼을 클릭하면 메인</li> </ol> |



|                |   |
|----------------|---|
|                | 페이지로 이동한다.  |
| Precondition   | <ol style="list-style-type: none"> <li>1. 사용자가 시스템에 등록되지 않은 상태이어야 한다.</li> <li>2. 이미 등록된 사용자와 동일한 계정명을 사용해서는 안된다.</li> <li>3. pw, pw확인 string이 동일해야하며 사용자가 입력한 pw가 pw 조건에 부합하는지 확인해야 한다.</li> </ol> |
| Post Condition | pw를 해싱하여 데이터베이스에 저장한다   |

[Table 9] Use case of Register

| 이름             | Refactoring user's code  |
|----------------|--|
| Actor          | 사용자  |
| Description    | 사용자가 사용자의 java 코드를 탄소 배출량을 계산하고 절감하는 방식과 코드 포매팅등의 방식으로 리팩토링하기 위해 시스템을 이용한다.  |
| Normal Course  | <ol style="list-style-type: none"> <li>1. 사용자는 자신의 코드가 실행되는 환경을 볼 수 있다.</li> <li>2. 사용자가 자신의 java 코드를 입력하고, 버튼을 클릭한다.</li> <li>3. 사용자는 코드가 리팩토링될 때까지 일정 시간 기다린다.</li> <li>4. 사용자는 인터페이스를 통해 리팩토링된 코드를 볼 수 있다.</li> </ol> |
| Precondition   | <ol style="list-style-type: none"> <li>1. 사용자는 Java 코드 외에 다른 코드를 입력해서는 안된다.</li> <li>2. 사용자는 리팩토링이 되는 동안 버튼을 다시 클릭할 수 없다.</li> <li>3. 사용자는 결과값을 수정할 수 없다.</li> </ol>   |
| Post Condition | 해당 Java 코드를 탄소배출량이 감소되고 코드가 포매팅된 형태로 리팩토링하여 사용자에게 출력한다.  |

[Table 10] Use case of Registering User's code

| 이름          | Result statistics   |
|-------------|---|
| Actor       | 사용자   |
| Description | 사용자가 자신이 적은 코드가 리팩토링이 된 후 탄소 배출량이 어느정도 절감 되었는지를 수치와 도표로 확인할 수 있다. |

|                |  |
|----------------|--|
| Normal Course  | <ol style="list-style-type: none"> <li>1. 사용자가 입력한 코드가 리팩토링된다.</li> <li>2. 탄소 배출 절감 결과가 수치와 도표로 나타난다.</li> </ol> |
| Precondition   | 사용자는 결과가 나오기 전, 리팩토링 버튼을 다시 누를 수 없다.   |
| Post Condition | 사용자의 탄소 배출 절감 결과가 수치와 도표로 나타난다.  |

[Table 11] Use case of Result statistics

| 이름             | System Information   |
|----------------|--|
| Actor          | 사용자  |
| Description    | 사용자의 자바코드가 실행된 서버의 시스템 정보를 제공한다.   |
| Normal Course  | <ol style="list-style-type: none"> <li>1. 사용자가 자신의 자바코드를 입력하고 리팩토링 버튼을 클릭한다.</li> <li>2. 사용자가 입력한 코드가 서버에 전달되어 실행시간, 메모리 사용량 등의 탄소배출 요소를 측정한다.</li> <li>3. 해당 서버의 CPU, 메모리 등의 시스템 정보가 사용자 화면에 표시된다.</li> </ol> |
| Precondition   | <ol style="list-style-type: none"> <li>1. 리팩토링 버튼을 누르기 전에는 시스템정보를 표기하는 창이 표시되지 않는다.</li> </ol>   |
| Post Condition | <ol style="list-style-type: none"> <li>1. 서버의 시스템 정보가 리팩토링된 코드 좌측 하단에 표기된다.</li> </ol>   |

[Table 12] Use case of System Information

| 이름            | Growing tree   |
|---------------|--|
| Actor         | 사용자  |
| Description   | 누적 bit를 사용하여 나무를 키운다.  |
| Normal Course | <ol style="list-style-type: none"> <li>1. 사용자가 리팩토링을 할 때마다 bit를 1씩 얻는다.</li> <li>2. 사용자가 획득한 bit가 누적 bit에 쌓인다.</li> <li>3. 누적 bit가 30을 넘을 때마다 나무가 다음 단계로 성장한다.</li> <li>4. 사용자는 인터페이스를 통해 자신의 나무 성장 현황을 확인할 수 있다.</li> </ol> |

|                |                              |
|----------------|------------------------------|
| Precondition   | 하루 최대 얻을 수 있는 bit는 15bit이다.  |
| Post Condition | 리팩토링을 많이 할수록 나무 성장 단계가 증가한다. |

[Table 13] Use case of Growing tree

| 이름             | Gaining bit   |
|----------------|---|
| Actor          | 사용자   |
| Description    | 리팩토링을 할 때마다 bit가 1씩 쌓인다.  |
| Normal Course  | <ol style="list-style-type: none"> <li>1. 사용자가 입력한 코드에 대해서 리팩토링이 이루어진다.</li> <li>2. 사용자의 bit가 1씩 증가한다.</li> </ol> |
| Precondition   | <p>사용자가 로그인한 상태이어야 한다.<br/> 로그인한 사용자가 자신의 코드로 리팩토링을 수행한다.<br/> 하루 최대 획득 가능한 bit 수(15 bit)를 초과하지 않은 상태이어야 한다.</p>  |
| Post Condition | 사용자는 자신이 얻은 bit를 이용해 나무를 키우고 광고를 구매할 수 있게 된다.   |

[Table 14] Use case of Gaining bit

| 이름             | Buying advertisement   |
|----------------|--|
| Actor          | 사용자  |
| Description    | 사용자가 자신의 bit를 사용하여 웹사이트에 자신의 광고를 게시할 수 있는 광고권을 구매한다.   |
| Normal Course  | <ol style="list-style-type: none"> <li>1. 사용자가 마이페이지에 접속한다.</li> <li>2. 구매할 광고권 기간을 선택한다.(1일 30bit 단위)</li> <li>3. 정해진 규격에 맞는 이미지를 업로드한다.</li> </ol> |
| Precondition   | 1. 사용자는 자신이 선택한 광고권의 bit 소모량 이상의 bit를 보유하고 있어야 한다.   |
| Post Condition | 1. 사용자가 업로드한 광고 이미지가 DB에 저장된다.   |

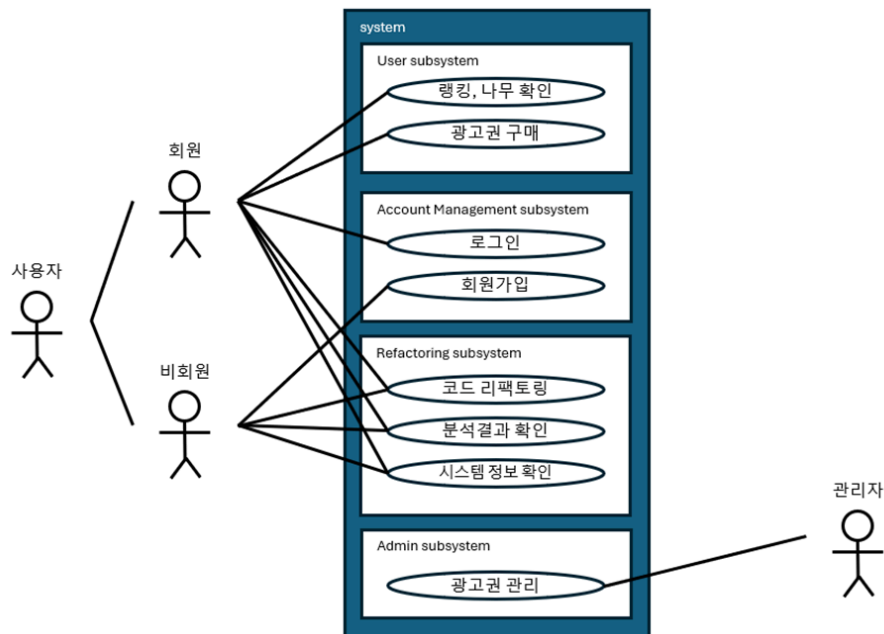
[Table 15] Use case of Buying advertisement

| 이름 | Approving advertisement |
|----|-------------------------|
|----|-------------------------|

|                |   |
|----------------|---|
| Actor          | 매니저   |
| Description    | 광고권을 구매한 사용자가 업로드한 광고 이미지를 매니저가 승인하여 웹사이트 게재를 승인한다.   |
| Normal Course  | <ol style="list-style-type: none"> <li>1. 매니저는 웹사이트에 admin 계정으로 로그인한다.</li> <li>2. 매니저는 광고권을 구매한 회원들이 업로드한 광고 리스트를 확인한다.</li> <li>3. 광고 리스트 중 승인할 광고를 선택하여 웹사이트 게재를 승인한다.</li> </ol>  |
| Precondition   | 1. 광고권을 구매한 회원들이 업로드한 광고 이미지가 DB에 저장되어 있어야 한다.  |
| Post Condition | <ol style="list-style-type: none"> <li>1. 매니저가 승인한 광고에 한하여 다음날 00시부터 웹페이지에 광고가 게시된다.</li> <li>2. 한 날짜에 여러 광고가 게시되는 경우 광고는 슬라이드 형태로 게시된다.</li> <li>3. 매니저가 승인한 광고에 대해 회원의 bit가 차감된다.</li> <li>4. 매니저가 광고를 게재 거절한 경우 사용자의 bit가 차감되지 않으며 사용자에게 게재가 거절되었다는 문구가 표시된다.</li> </ol> |

[Table 16] Use case of Approving advertisement

### 3.2.2. Use Case Diagram



[Figure 1] Use case diagram

### 3.2.3. Data Dictionary

| User     |      |    |
|----------|------|----|
| id       | int  | PK |
| username | text |    |
| email    | text |    |
| password | text |    |

[Figure 2] Table of User

| Field    | Description           |
|----------|-----------------------|
| id       | User id               |
| username | User unique name      |
| email    | User email            |
| password | User encoded password |

[Table 17] Description of table User

| CodeRefactoring  |      |    |
|------------------|------|----|
| id               | int  | PK |
| user_id          | int  | FK |
| pattern_num      | enum |    |
| decreased_carbon | int  |    |
| payed_bit        | int  |    |
| today_bit_count  | int  |    |

[Figure 3] Table of CodeRefactoring

| Field            | Description   |
|------------------|---|
| id               | CodeRefactoring id  |
| user_id          | User id (Foreign key)                                     |
| pattern_number   | Pattern number of reducing carbon amount (one of [1,2,3]) |
| decreased_carbon | Amount of decreased carbon amount by refactoring          |
| payed_bit        | Payed bit point to user                                   |
| today_bit_count  | Count of paid bit point of user due to daily quota        |

[Table 18] Description of table CodeRefactoring

| UserBit     |     |    |
|-------------|-----|----|
| user_id     | int | PK |
| total_bit   | int |    |
| current_bit | int |    |

[Figure 4] Table of UserBit

| Field       | Description                        |
|-------------|------------------------------------|
| user_id     | User id (Foreign key)              |
| total_bit   | User's total bit (tree)            |
| current_bit | User's current bit (advertisement) |

[Table 19] Description of table UserBit

| UserTree  |      |    |
|-----------|------|----|
| user_id   | int  | PK |
| total_bit | int  |    |
| tree_tier | enum |    |

[Figure 5] Table of UserTree

| Field     | Description  |
|-----------|--|
| user_id   | User id (Foreign key)  |
| total_bit | User's total bit   |
| tree_tier | Tier of tree (one of [seed, sprout, sapling, young tree, mature tree]) |

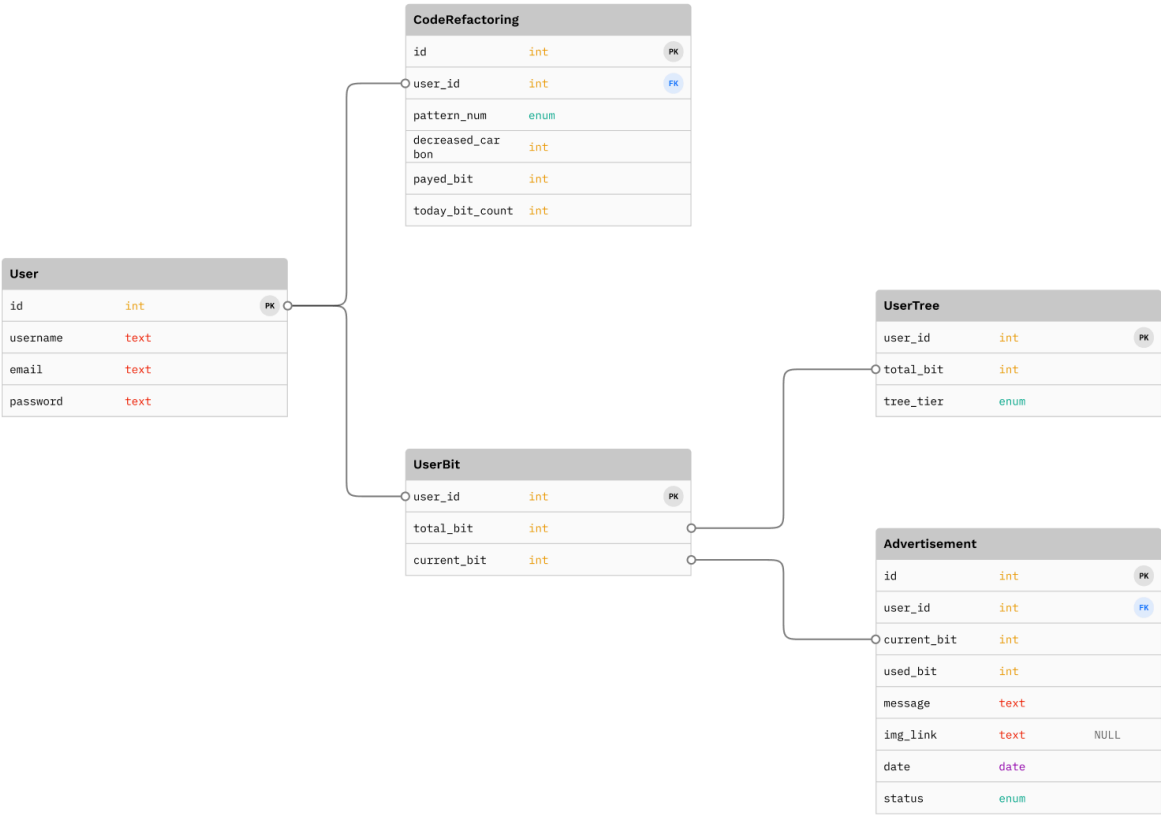
[Table 20] Description of table UserTree

| Advertisement |      |      |
|---------------|------|------|
| id            | int  | PK   |
| user_id       | int  | FK   |
| current_bit   | int  |      |
| used_bit      | int  |      |
| message       | text |      |
| img_link      | text | NULL |
| date          | date |      |
| status        | enum |      |

[Figure 6] Table of Advertisement

| Field       | Description   |
|-------------|---|
| id          | Advertisement id  |
| user_id     | User id (Foreign key)   |
| current_bit | User's current bit  |
| used_bit    | Used bit for advertisement apply                                  |
| message     | Advertisement message   |
| img_link    | Advertisement image link of Firestore                             |
| date        | Advertisement created date  |
| status      | Approval status by manager (one of [applied, approved, rejected]) |

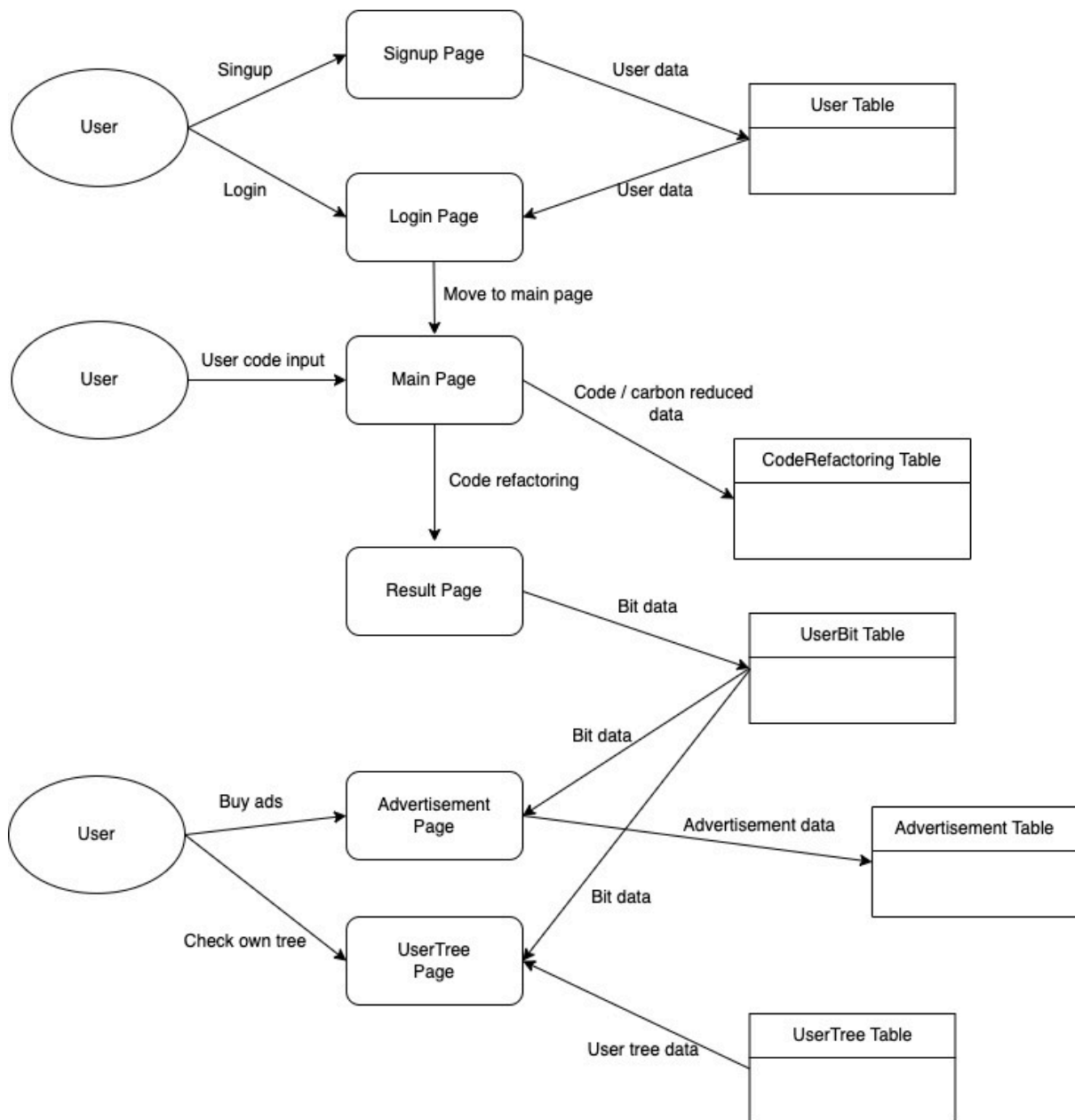
[Table 21] Description of table Advertisement



[Figure 7] ERD



### 3.2.4. Data Flow Diagram



[Figure 8] Data flow diagram

### 3.3. Performance requirements

#### 3.3.1. Static Requirement

- 시스템은 여러 명의 사용자가 동시에 코드를 입력하고 리팩토링된 코드를 받아볼 수 있도록 해야 한다.
- 시스템은 2GHz 이상의 싱글 프로세서, 2GB 이상의 메모리 환경의 데스크탑 윈도우에서 부드럽게 동작해야 한다.

- 최소 대역폭 5Mbps의 업로드/ 다운로드 대역폭을 가진 네트워크 환경에서 원활하게 작동한다.
- 운영체제는 윈도우 10 이상이며, 웹 어플리케이션을 지원하는 환경에서 정상적으로 동작한다.

### 3.3.2. Dynamic Requirement

- 서비스는 최대 100명의 사용자가 동시에 접속해도 부드럽게 동작해야 하며, 최소 10000건 이상의 리팩토링 사용내역을 데이터베이스에 저장할 수 있어야 한다.
- 웹 브라우저가 메인 페이지를 5초 이내에 로드할 수 있어야 한다.
- 사용자의 신규 등록 요청을 5초 이내로 완료하고 데이터 베이스에 저장되어야 한다.
- 사용자의 로그인/로그아웃 요청을 5초 이내로 완료해야 한다.
- 사용자가 Java code 입력 후 빌드 및 실행시간을 제외하고 5초 이내에 결과 페이지를 로드할 수 있어야 한다.
  - 단, 1000줄 이하의 code에 대해서는 10초 이내에 빌드, 실행하여 결과페이지를 로드할 수 있어야 한다.

## 3.4. Logical database requirements

일반 데이터베이스로는 Firebase Realtime database, 이미지 저장 데이터베이스로는 Firebase Firestore를 사용하여 데이터 관리를 한다.

사용자를 관리 하기 위해 사용자 테이블을 두어 사용자의 정보를 저장한다.

사용자의 누적 bit량에 따라 Tree 테이블에 사용자의 나무 성장이 반영된다.

사용자는 현재 보유한 bit를 사용하여 광고를 구매할 수 있고, 해당 정보는 광고 테이블에 기록된다.

사용자는 우선 자신의 코드 리팩토링 결과를 저장하기 위해 코드 테이블을 활용할 수 있다. 이 테이블은 id 로 구별되어 해당 코드의 제출 날짜, 사용자가 입력한 코드, 리팩토링 된 코드, 그리고 탄소 배출량의 감소량 등의 정보를 포함한다.

나무 테이블을 두어 사용자는 현재 자신의 나무 성장 현황을 저장할 수 있다.

또한, 광고 테이블을 두어 자신이 신청한 광고의 배너 이미지, 신청 날짜, 개시 날짜 등을 저장할 수 있다.

### 3.5. Design constraints

본 시스템은 web application 으로 web browser가 동작할 수 있는 환경에서만 실행이 가능하다. 사용자가 입력하는 코드는 Java 언어만 허용하며, 그 외에는 error 처리가 필요하다. 사용자가 입력한 코드에 대해서는 단 하나의 refactoring 코드만 제공한다. 사용자가 bit로 광고를 구매할 시에는 사업적인 목적이 아닌 개인적인 광고만 허용하며, 이는 매니저의 판단 아래에서 승인된 광고만 게시 가능하다.

### 3.6. Software system attributes

#### 3.6.1. Reliability

시스템 배포시, 믿을 수 있는 인증서 기관에서 SSL/TLS 인증서를 발급받아서 배포하고, https로만 통신할 수 있도록 한다.

#### 3.6.2. Availability

사용자는 자신이 입력한 코드의 리팩토링 과정을 실시간으로 확인할 수 있어야 한다. 대기가 길어진다면, 얼마나 더 기다려야 하는지 알 수 있도록 해야한다. 또한, 사용자는 자신의 코드 리팩토링에 대한 탄소배출량 절감 결과를 즉각 확인할 수 있어야 한다.

#### 3.6.3. Security

하루 리팩토링 횟수 제한을 두어 어떤 사용자가 웹 서비스를 다른 악의적인 목적으로 쓰는 경우를 방지할 수 있다. 또한, 광고 구매시 매니저가 판단하여 악의적인 광고 개시를 방지할 수 있다.

#### 3.6.4. Maintainability

유닛 테스트, 통합 테스트 등을 자동화하여 코드 변경이 시스템 전체에 미치는 영향을 빠르게 파악하고 예기치 않은 오류를 방지할 수 있다.

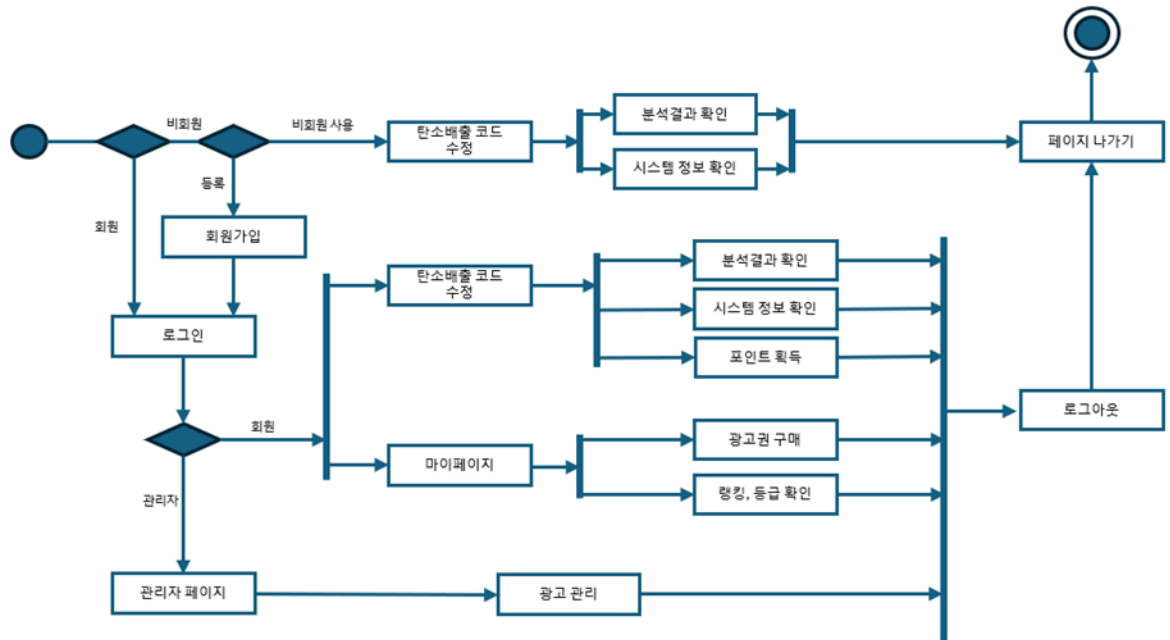
팀 전체의 코딩 컨벤션을 확립하여, 앞으로의 유지보수성을 향상시킬 수 있다.

#### 3.6.5. Portability

이 시스템은 웹 어플리케이션이 실행될 수 있는 환경에서는 어느 곳에서든지 이식이 가능하다.

### 3.7. Organizing the specific requirements

#### 3.7.1. Process diagram



[Figure 9] Process diagram

#### 4. Appendices

##### 4.1. Green Algorithms 탄소배출량 공식

$$PUE = \frac{P_{total}}{P_{compute}}$$

$$E = t \times (n_c \times P_c \times u_c + n_m + P_m) \times PUE \times 0.001$$

$$C = E \times CI$$

$$C = t \times (n_c \times P_c \times u_c + n_m + P_m) \times PUE \times CI \times 0.001$$

| 기호  | 설명      | 단위       | 비고 |
|-----|---------|----------|----|
| $C$ | 탄소 배출량  | $gCO_2e$ |    |
| $E$ | 에너지 소비량 | $kWh$    |    |

|       |   |              |   |
|-------|---|--------------|---|
| $PUE$ | Power Usage Effectiveness,<br>데이터 센터의 에너지 효율성 | -            | $PUE = \frac{P_{total}}{P_{compute}},$<br>1.00이 이상적인 값,<br>데이터센터들의 평균적인<br>측정 값은 1.67 |
| $t$   | Runtime                                       | $h$          | 단위에 유의  |
| $n_c$ | 코어 수  | -            |   |
| $P_c$ | 코어 연산에 따른 전력 소비량                              | $W$          | TDP, CPU 별 제조사에서<br>제공하는 값이 있음  |
| $u_c$ | CPU 사용 비율                                     | -            |   |
| $n_m$ | 사용가능한 메모리 사이즈                                 | $GB$         |   |
| $P_m$ | 메모리 사용량에 따른 전력 소비<br>비율                       | $W/GB$       |   |
| $CI$  | Carbon Intensity                              | $gCO_2e/kWh$ |   |

[Table 22] Signs of the Green Algorithm's formula