Software Requirements Specification

BINARY TREE

by

구성현, 권혁준, 김성은, 나인호, 오성현, 이승환

TEAM 4

Instructor:	이은석
Teaching Assistant:	김진영, 최동욱, 허진석, 김영경
Document Date:	04 May, 2024
Faculty:	SungKyunKwan University

Contents

List of Figures

List of Tables

1 Introduction

1.1. Purpose

이 문서는 소스코드 탄소 배출량 측정 도구 개발 프로젝트의 목적을 명시하고, 프로젝트 참여자들에게 명확한 지침을 제공하기 위해 작성되었다. 이는 프로젝트 팀원, 프로젝트 관리자, 개발자 및 최종 사용자를 대상으로 한다.

1.2. Scope

이 요구사항 명세서는 웹 기반 "소스코드 탄소배출량 측정 도구" 개발을 목표로 한다. 이 도구는 개발 과정에서 발생하는 탄소 배출량을 측정하고, 효율적이고 친환경적인 코드 작성을 위한 개선 사항을 제안함으로써 사용자의 코딩 방식을 개선하는 데 도움을 준다. 본 도구는 사용자가 직접 코드를 입력하고 결과를 즉시 확인할 수 있도록 설계되었다.

1.3. Definitions, Acronyms, and Abbreviation

탄소 배출량: 소프트웨어 실행으로 인해 발생하는 전력 소비에 따른 탄소 배출량
UI (User Interface): 사용자 인터페이스.
SRS (Software Requirements Specification): 소프트웨어 요구사항 명세서
API (Application Programming Interface): 응용 프로그래밍 인터페이스

1.4. Reference

Green Algorithms: Measuring Sustainability in AI, Medium.
Green Algorithms GitHub Repository, GitHub.

1.5. Overview

본 요구사항 명세서는 도입부, 시스템 목적, 범위, 정의 및 약어, 참고 문헌 및 개요를 포함하여 구성되어 있다. 후속 섹션에서는 시스템 기능, 사용자 요구사항, 시스템 성능 요구사항, 인터페이스 요구사항, 운영 환경, 설계 제약사항, 품질 특성 및 기타 요구사항에 대해 자세히 다룰 예정이다.

2 Overall Description

2.1. Product Perspective

"소스코드 탄소배출량 측정 도구"는 웹 기반 플랫폼으로서, 소프트웨어 개발 과정에서 발생하는 탄소 배출량을 계산하고, 친환경적인 코드 작성을 위한 개선 제안을 제공한다. 이 시스템은 독립적으로 운영되며, 사용자가 직접 코드를 입력하고 탄소 배출량을 측정할 수 있는 기능을 갖추고 있다. 본 시스템은 기존의 개발 환경과 연동되어 사용자의 개발 과정을 개선하는 데 도움을 줄 수 있도록 설계되었다.

2.1.1 System Interfaces

본 시스템은 사용자가 웹 인터페이스를 통해 코드를 입력하고, 서버는 입력된 코드의 탄소 배출량을 계산하여 결과를 사용자에게 반환한다. 시스템은 기존 개발 환경과의 통합을 위해 RESTful API 를 제공하여, 외부 시스템과의 연동이 가능하다.

2.1.2 User Interfaces

사용자 인터페이스는 웹 페이지 형태로 제공되며, 사용자는 웹 브라우저를 통해 접근할 수 있다. 인터페이스에는 코드 입력 필드, 실행 버튼, 결과 표시 창 등이 포함되어 있어 사용자가 직관적으로 작업을 수행할 수 있다. 또한, 탄소 절감 효과를 시각화 하는 그래픽 요소도 포함된다.

2.1.3 Hardware Interfaces

시스템은 표준 웹 서버 하드웨어에서 운영되며, 사용자는 개인 컴퓨터나 모바일 장치를 통해 접근할 수 있다. 서버 측에서는 계산을 위한 충분한 처리 능력과 메모리가 필요하다.

2.1.4 Software Interfaces

시스템은 HTML, CSS, JavaScript 를 기반으로 하는 웹 프론트엔드와, Python 과 Flask 를 사용한 백엔드로 구성된다. 데이터베이스 관리는 MySQL 을 사용하며, REST API 를 통해 프론트엔드와 백엔드 간의 통신이 이루어진다.

2.1.5 Memory Constraints

웹 서버는 최소 8GB 의 RAM 을 필요로 하며, 충분한 스토리지 공간을 확보해야 한다. 클라이언트 측에서는 특별한 메모리 제약은 없으나, 웹 페이지 로딩과 스크립트 실행을 위해 최신 브라우저의 사용을 권장한다.

2.1.6 Operations

시스템은 코드 입력, 탄소 배출량 계산, 결과 반환의 주요 작업을 수행한다. 사용자는 웹 인터페이스를 통해 코드를 제출하고, 시스템은 제출된 코드의 탄소 배출량을 계산한 후, 그 결과를 사용자에게 시각적으로 표시한다.

2.1.7 Hardware Interfaces (Additional)

서버 하드웨어는 높은 처리 능력과 데이터 처리를 위한 높은 네트워크 대역폭을 갖추어야 한다. 사용자의 입력 처리와 데이터의 저장 및 검색을 신속하게 처리할 수 있도록 구성된다.

2.2. Product Function

본 시스템은 소프트웨어 개발 과정에서 발생하는 탄소 배출을 측정하고, 사용자가 더 효율적이고 친환경적인 코드를 작성할 수 있도록 지원하는 웹 기반 플랫폼을 제공한다.

2.2.1. Account Service

사용자가 손쉽게 계정을 생성하고 관리할 수 있는 기능을 제공한다. 사용자는 회원가입을 통해 아이디와 비밀번호를 생성하고, 시스템은 사용자의 코드와 관련된 활동을 추적하고 분석할 수 있는 개인 대시보드를 제공한다. 해당 대시보드는 2.2.4 Tree Management 기능과 관련이 있다. 또한 시스템은 사용자들 간의 참여를 유도하기 위해 랭킹 시스템을 제공하는데, 이는 2.2.5. Ranking System 에서 자세히 설명한다.

2.2.2. Carbon Emission Measurement

사용자의 코드가 환경에 미치는 영향을 정량화하고 분석할 수 있는 도구를 제공한다. Green Algorithms 기반으로 사용자가 제출한 코드의 탄소 배출량을 보다 정확히 측정할 수 있는 알고리즘과 도구를 개발한다. 사용자가 코드를 제출하면, 시스템은 코드에 대한 탄소 배출량을 측정하여 사용자에게 제공하며 이를 통해 사용자들은 자신의 코드 작성 습관이나 프로세스를 개선할 수 있는 방법을 파악할 수 있다.

2.2.3. Code Improvement

사용자의 코드를 분석하여 개선할 수 있는 방법을 추천하는 기능을 제공한다. 시스템은 사용자가 제출한 코드를 검토하고, 중복되거나 비효율적인 부분을 찾아내어 개선 제안을 제공한다.

2.2.4. Tree Management

사용자가 환경 보호에 기여한 정도를 시각적으로 확인할 수 있는 기능을 제공한다. 사용자가 친환경적인 코드를 작성하고 탄소 배출량을 줄이는 데 기여할 때마다 가상의 나무를 심는다. 사용자는 개인 대시보드에서 나무의 상태를 확인할 수 있다.

2.2.5. Ranking System

사용자들 간의 2.2.4. Tree Management 나무 심기 현황에 따른 순위를 제공하는 랭킹 시스템을 제공한다. 랭킹 시스템을 통해 사용자들은 탄소 절감에 대한 자신의 노력을 비교하고 다른 사용자들과 경쟁을 즐길 수 있다.

2.3. User Characteristics

2.3.1. Code Developers

코드 개발자는 코드 개발에 전문적인 지식과 기술을 가지고 있다. 컴퓨터 공학을 전공하였거나, 코딩 관련 지식을 가진 사람이 이에 해당한다. 코드를 스스로 작성하여 제출하고, 시스템이 제공하는 개선된 코드를 이해하고 활용할 수 있다.

2.3.2. Environmentalists

환경운동가들은 환경 보호에 대한 강력한 열정과 관심을 가진다. 이들은 탄소 절감 활동에 참여하고 다른 사용자들을 동기부여하여 지속 가능한 행동을 촉진하는 역할을 한다. 탄소 절감에 관련된 정보를 제공하고 사용자들이 탄소 배출을 줄이는데 도움이 되는 리소스와 도구를 제공한다.

2.3.3. Data Analysts

데이터 분석가들은 데이터를 분석하고 시각화 하여 환경적 영향을 이해하고 개선할 수 있는 정보를 제공한다. 이들은 탄소 배출량을 분석하여 각종 그래프로 시각화 하여 보여준다.

2.3.4. System Administrators

시스템 관리자는 웹 서비스와 서버를 관리하여 사용자들이 원활하게 플랫폼을 이용할 수 있도록 한다. 이들은 기술적인 문제를 해결하고 사용자 경험을 개선하기 위해 노력한다. 웹 서비스에 충분한 지식을 가진 사람이 이 역할을 수행할 수 있다.

2.4. Constraints

2.4.1. Platform Environment Constraints

본 시스템은 웹 기반 플랫폼으로 개발되어 데스크톱 브라우저를 주로 지원하며, 모바일 환경을 고려하지 않는다. 사용자는 자바스크립트가 활성화된 인터넷 브라우저를 사용한다.

2.4.2. Hardware Constraints

서버 측에서는 충분한 처리 능력과 메모리가 필요하며, 클라이언트 측에서는 최신 브라우저의 사용을 권장한다. 웹 서버는 최소 8GB 의 RAM 을 권장하고, 충분한 스토리지 공간을 확보해야 한다.

2.4.3. Control Performance

서버 측에서는 동시 요청의 수나 처리 속도에 제한이 있을 수 있다. 서버의 성능을 고려하여 요청을 효율적으로 처리해야 한다.

2.4.4. Reliability

시스템은 안정적으로 운영되어야 한다. 장애가 발생할 경우 적절한 조치가 필요하다.

2.5. Assumptions and Dependencies

사용자가 제출한 코드의 탄소 배출량을 정확히 측정할 수 있는 알고리즘은 이미 존재한다고 가정한다. 이는 개발팀이 이러한 알고리즘을 찾거나 개발할 필요가 없음을 의미한다. 웹사이트의 궁극적인 목표는 개발자가 친환경적인 코드를 작성할 수 있도록 도와주는 것이기 때문에 코딩하기 쉬운 환경에서 웹 사이트를 사용한다고 가정한다. 이러한 가정으로, 위에서 언급된 최소 사양을 충족하지 않는 장치에서는 시스템이 원활하게 작동되지 않을 수 있다.

3 External Interface Requirements

3.1. External Interfaces

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 - 정확도: 유저의 마우스 및 키보드 입력에 따른 정확도 - 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 명령 요청
화면 형식 및 구성	해당 없음
윈도우 형식 및 구성	해당 없음
데이터 형식 및 구성	JAVA 코드, Text, Int 형의 코드 값
명령 형식	각 코드 값에 따른 명령 매핑
종료 메시지	해당 없음

Table 1: User interface of input processing

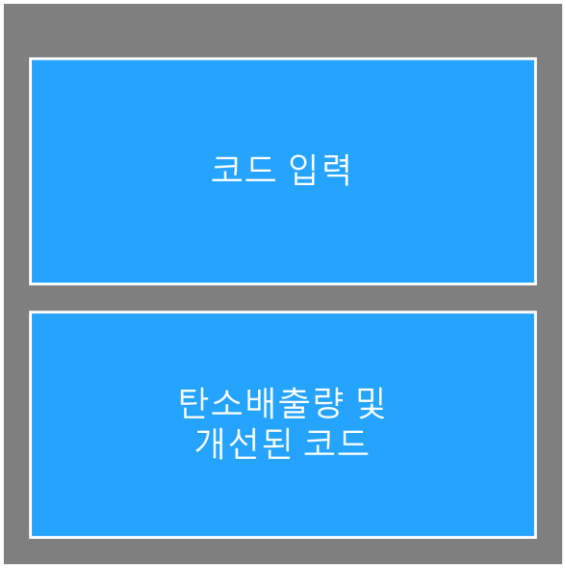
이름	모니터를 통한 메인 화면 출력
목적/내용	시스템 사용자에게 제공하는 인터페이스
입력 주체/출력 목적지	클라이언트/사용자
범위/정확도/허용 오차	- 범위: 화면에서의 버튼의 개수에 따른 입력 범위 - 정확도: 유저의 마우스 및 키보드 입력에 따른 정확도 - 허용 오차: 해당 없음
단위	화면
시간/속도	사용자의 입력에 따른화면 전환
타 입출력과의 관계	사용자의 입력을 위한 인터페이스로서 출력 후 사용자의 입력 대기
화면 형식 및 구성	 - 개발할 웹사이트는 위쪽의 코드를 입력하는 공간과 아래쪽의 결과와 개선된 코드가 나오는 형태로 구성 - 코드를 입력하고 탄소배출량과 개선된 코드를 확인할 수 있음 - 현재 코드의 탄소배출량과 개선된 코드의 탄소배출량을 비교하는 형태로 진행
윈도우 형식 및 구성	- JAVA code 를 작성하는 창 - Linear Layout 형식으로 결과 출력 - 개선된 JAVA code 출력 창
데이터 형식 및 구성	이미지 텍스트
명령 형식	해당 없음
종료 메시지	해당 없음

Table 2: User interface of main page

이름	사용자 등록 인터페이스
목적/내용	시스템 사용자의 정보를 등록
입력 주체/출력 목적지	사용자/서버
범위/정확도/허용 오차	해당 없음
단위	화면
시간/속도	해당 없음
타 입출력과의 관계	해당 없음
화면 형식 및 구성	- 새로운 계정에 대한 정보 입력을 위한 슬롯 - 계정 정보 등록을 위한 버튼
윈도우 형식 및 구성	해당 없음
데이터 형식 및 구성	Query
명령 형식	등록 버튼에 따른 명령 매핑
종료 메시지	“You're registered.”

Table 3: User interface of register

3.1.2. Hardware Interface

이름	사용자에서 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/탄소배출량 계산, 코드 개선에 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	웹 사이트
목적/내용	화면 출력
입력 주체/출력 목적지	해당 없음
범위/정확도/허용 오차	Chrome, Edge, Firefox, Safari 와 같은 웹 브라우저에서 사용 가능
단위	해당 없음
시간/속도	새로 고침에 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 of application device for the system

3.1.4. Communication Interface

이름	호스트 서버-클라이언트
목적/내용	각 클라이언트에서 호스트 서버에 접속을 요청하고, 사용자가 입력한 코드를 호스트 서버에서 각 클라이언트에게 전달하고 이에 해당하는 결과를 제공
입력 주체/출력 목적지	클라이언트와 호스트 서버
범위/정확도/허용 오차	해당 없음
단위	패킷
시간/속도	최소 10Mbps 이상
타 입출력과의 관계	해당 없음
데이터 형식	- Struct 를 이용한 명령 코드 - Struct 를 이용한 계정 로그인 데이터
명령 형식	Send() 콜에 의한 통신
종료 메시지	Close() 콜에 의한 소켓 종료

Table 6: Communication interface of applicable device for the system

3.2. Function requirements

3.2.1. Use Case

Use case name	***Register***
Actor	미등록 사용자
Description	미등록 사용자가 시스템을 이용하기 위해 회원으로 가입을 기도하는 과정이다.
Normal Course	1. 모든 사용자는 웹사이트 접속 후 “회원가입” 버튼을 클릭한다. 2. 사용자가 회원가입 할 수 있도록 회원가입 양식을 제공한다. 3. 사용자는 양식에 따라 정보를 입력해야 한다. 필요한 정보는 다음을 포함한다. (a) ID(이메일 주소) (b) Password (c) 학번 4. 시스템은 이메일 주소가 올바른지 확인하고 암호를 찾는 상황을 준비하기 위해 지정된 메일주소로 확인 코드를 보낸다. 5. 양식에 맞게 작성한 후, “완료" 버튼을 누르면 시스템은 계정을 생성하고 데이터베이스에 저장한다. 6. 저장이 성공되면 시스템은 사용자에게 가입 성공 메시지를 표시하고 로그인 페이지로 돌아간다.
Precondition	사용자가 아직 시스템에 등록되지 않아야 한다. 사용자가 올바른 정보를 입력해야 한다. 이메일 주소를 다른 사용자의 이메일 주소와 중복해서는 안 된다. 잘못된 입력이 있는 경우, 시스템은 이메일 주소 및 비밀번호의 형식을 확인해야 한다.
Postcondition	사용자 계정이 데이터베이스에 저장된다.
Assumptions	해당 없음

Table 7: Use case of register

Use case name	***Log-in/out***
Actor	등록된 사용자
Description	로그인은 시스템에 등록된 사용자가 서비스를 사용하기 위해 시스템에 들어가려고 하는 프로세스이다. 로그아웃은 로그인한 사용자가 시스템에서 나가려고 할 때의 프로세스이다.
Normal Course	로그인 - 시스템에 이미 회원으로 등록된 사용자가 시스템에서 서비스를 사용하려고 한다. - 로그인을 위해 회원 가입에 사용한 ID 와 PW 을 입력한다. - 데이터베이스에 등록된 정보와 일치하면, 시스템은 사용자가 시스템에 접속할 수 있도록 하며, 시스템이 제공하는 서비스를 이용할 수 있다. 로그아웃 - 시스템을 종료하려면 '로그아웃' 버튼을 클릭한다. 사용자가 로그아웃하지 않고 응용 프로그램을 닫은 경우 시스템이 해당 사용자에 대한 세션을 임의로 닫는다.
Precondition	로그인: 사용자가 시스템이 이미 등록되어 있어야 한다. 로그아웃: 사용자가 로그인 상태여야 한다.
Postcondition	사용자가 온라인 상태가 된다.
Assumptions	해당 없음

Table 8: Use case of log-in/out

Use case name	***Code Analysis & Carbon Emission Measurement***
Actor	등록된 사용자
Description	사용자는 코드를 시스템에 제출하여 탄소 배출량을 측정하고, 코드 최적화를 위한 제안을 받는다.
Normal Course	1. 사용자는 사용한 언어, 등을 선택한 후, 코드를 시스템에 제출한다. 2. 시스템은 코드를 분석하고 탄소 배출량을 측정한다. 3. 시스템은 측정 결과와 함께 개선 제안을 사용자에게 표시한다.
Precondition	사용자가 로그인 상태여야 한다.
Postcondition	사용자의 코드의 탄소 배출량은 데이터베이스에 업데이트 된다. 코드의 탄소 배출량이 측정되고 사용자에게 제공된다.
Assumptions	해당 없음

Table 9: Use case of code analysis & carbon Emission Measurement

Use case name	*Improvement Suggestion & Result Display*
Actor	등록된 사용자
Description	시스템은 사용자에게 개선된 코드를 제공하고, 개선된 코드를 적용할 경우의 이점을 시각적으로 보여준다.
Normal Course	1. 사용자는 탄소 배출량 측정 결과 페이지에서 “개선 제안 보이기"버튼을 클릭한다. 2. 시스템은 사용자의 원본 코드에 대한 개선 사항을 제안하고, 개선된 코드를 보여준다. 3. 사용자는 원본 코드와 개선된 코드를 비교하여 볼 수 있다. 4. 시스템은 개선된 코드의 탄소 배출량과 기존 코드 대비 개선된 비율을 함께 제공한다. 5. 사용자는 제안된 개선 사항을 자신의 코드에 적용하여 탄소 배출량을 줄일 수 있다. 6. 사용자가 개선 사항을 코드에 적용하면, 시스템은 개선된 내용을 데이터베이스에 업데이트한다.
Precondition	사용자가 로그인 상태여야 한다. 사용자는 코드를 제출한 상태여야 한다.
Postcondition	사용자가 개선 제안을 볼 수 있으며, 개선된 결과를 기존 결과와 비교할 수 있다.
Assumptions	해당 없음

Table 10: Use case of improvement suggestion & result display

Use case name	***Tree Management***
Actor	등록된 사용자
Description	사용자가 코드를 개선하여 제출한 후 탄소 배출량 절감 수치를 점수화하여 데이터베이스에 추가한다.
Normal Course	1. 사용자는 프로그램 코드를 제출한다. 2. 시스템은 코드를 분석하고 탄소 배출량을 계산하며, 탄소 배출량을 줄일 수 있도록 개선된 코드를 제안한다. 3. 사용자는 제안사항을 참고하여 코드를 수정하고 제출한다. 4. 시스템은 수정된 코드의 탄소 배출량을 계산하고, 탄소 배출량이 더 증가한 경우 재수정을 요구한다. 배출량이 더 작거나 같을 경우 그 차를 계산하여 탄소 배출량 절감 수치를 구하고 데이터베이스를 업데이트한다. 5. 시스템은 기존의 총 탄소 배출량 절감 수치와 함께 대응되는 크기의 나무 그림을 표시한다. 6. 시스템은 코드 개선 후의 총 탄소 배출량 절감 수치와 함께 대응되는 크기의 나무 그림을 표시한다.
Precondition	사용자는 로그인 상태여야 한다.
Postcondition	사용자는 코드 개선을 통한 총 탄소 배출량 감소 효과를 시각적으로 확인할 수 있다.
Assumptions	해당 없음

Table 11: Use case of tree management

Use case name	***Ranking System***
Actor	등록된 사용자
Description	시스템은 사용자에게 현재까지의 탄소 배출량 절감 수치를 점수화하여 랭킹으로 보여준다.
Normal Course	1. 사용자는 웹사이트에 접속한다. 2. 시스템은 웹사이트의 코드 입력란 우측에 1~10 위까지의 탄소 배출량 절감 랭킹과 함께 현재 사용자 랭킹을 표시한다. 3. 사용자는 랭킹 영역을 클릭한다. 4. 세부 랭킹 페이지로 이동하며, 전체 사용자의 탄소 배출량 랭킹과 함께 현재 사용자 랭킹을 표시한다.
Precondition	시스템은 주기적으로 각 사용자의 탄소 절감량을 비교하여 순위를 업데이트하여야 한다.
Postcondition	사용자가 유저 랭킹을 확인할 수 있으며, 자신의 현재 순위를 확인할 수 있다.
Assumptions	해당 없음

Table 12: Use case of ranking system

3.2.2. Use Case Diagram

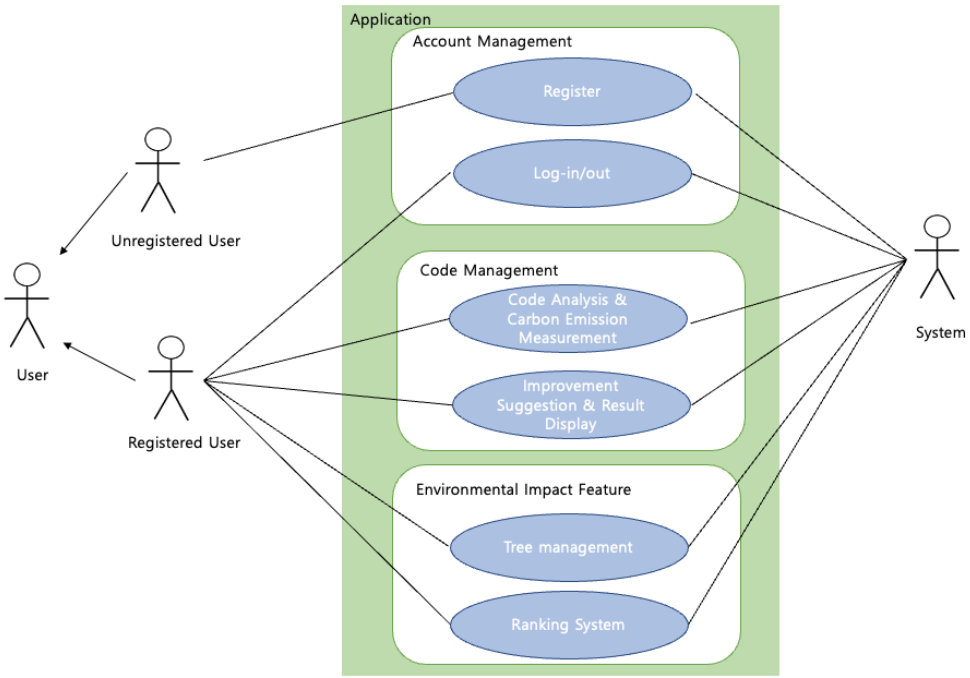


Figure 1: Use case diagram

3.2.3. Data Dictionary

Account			
Field	**key**	**Constraint**	**Description**
id	PK	Not NULL	User ID
password		Not NULL	User PW
name		Not NULL	User Name
total_carbon_emission		Default 0	Total carbon emissions from submitted code
rank			User's rank based on total_carbon_emission

Table 13: Data dictionary of account

Code			
Field	**key**	**Constraint**	**Description**
code_id	PK	Not NULL	code id
user_id	FK	Not NULL	user who submitted the code
original_code		Not NULL	original code submitted by user
improved_code			optimized code by system
original_carbon_emission		Not NULL	calculated carbon emission for original code
improved_carbon_emission			calculated carbon emission for improved code
timestamp		Not NULL	Date and time when code was submitted

Table 14: Data dictionay of code

3.2.4. Data Flow Diagram

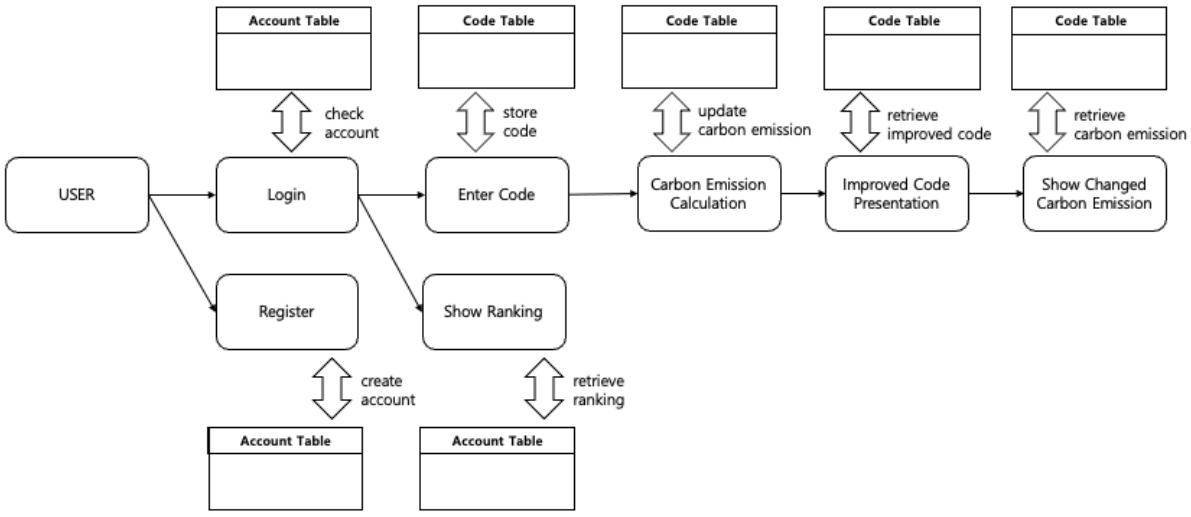


Figure 2: Data flow diagram

user - id(ID) / pw / 총 탄소 배출량 / 랭킹
code - id(code) / id(user) / 원본 코드 / 개선된 코드 / 원본 배출량 / 개선된 배출량 / 코드 제출 시간

3.3. Performance Requirements

아래는 본 시스템의 성능 요구사항에 관한 내용이다. 예측에 기반한 내용이며 실제 구현시 달라질 수 있다.

3.3.1. Static Numerical Requirement

시스템은 여러 명의 사용자가 동시에 소스코드를 입력하고, 탄소 배출량을 측정할 수 있도록 지원한다. 이 사용자들은 일반 사용자, 관리자로 분류된다.

- **서버:** 서버는 여러 사용자의 요청을 동시에 처리할 수 있는 충분한 처리 능력을 가져야 한다. 이를 위해 시스템은 최소 1.5GHz 의 멀티코어 프로세서를 탑재해야 하며, 8GB 이상의 RAM 을 갖추고 있어야 한다.
- **클라이언트:** 사용자의 디바이스는 웹 애플리케이션을 원활하게 구동할 수 있는 최소 사양을 만족해야 한다. 이는 최신 웹 브라우저를 구동할 수 있는 수준의 프로세서 및 메모리가 필요함을 의미한다.
- **네트워크:** 시스템은 최소 1Mbps 이상의 인터넷 연결 속도에서 안정적으로 작동할 수 있어야 한다.

3.3.2. Dynamic Numerical Requirement

- **동시 접속:** 시스템은 최소 200 명의 사용자가 동시에 코드를 입력하고 탄소 배출량을 측정할 수 있도록 지원해야 하며, 시스템은 관리자 및 일반 사용자의 동시 요청을 원활하게 처리할 수 있어야 한다. 그리고 1,000 명까지 사용자 계정 정보를 관리할 수 있어야 한다.
- **응답 시간:** 사용자가 코드 분석을 요청했을 때, 제출한 코드의 탄소 배출량을 측정하고 결과를 5 초 이내에 데이터베이스에 저장하고 사용자에게 알려주어야 한다.
- **처리량:** 시스템은 하루에 최대 10,000 개의 코드 분석 요청을 처리할 수 있어야 한다.
- **회원가입:** 사용자의 회원가입 요청을 5 초 이내로 완료해야 한다. 새로운 회원의 정보는 5 초 이내로 데이터베이스에 저장되어야 한다.
- **로그인/로그아웃:** 사용자의 로그인/로그아웃 요청을 5 초 이내로 완료해야 한다.

3.4. Logical Database Requirements

시스템은 Django 에 기본적으로 설치되어 있는 SQLite 데이터베이스 서비스를 이용하여 데이터를 관리한다. 해당 데이터베이스를 통하여 시스템 사용자들의 계정과 사용자 순위 정보를 저장한다. 각 사용자들의 계정에는 개인정보, 과거에 작성한 알고리즘과 개선된 알고리즘, 줄인 탄소량 등을 저장한다. 사용자가 원한다면 자신의 개인정보, 과거에 작성한 알고리즘과 개선된 알고리즘, 줄인 탄소량 등을 확인할 수 있어야 한다.

3.5. Design Constraints

3.5.1. Physical Design Constraints

시스템의 목적은 JAVA 코드의 탄소배출량을 측정 및 개선이므로 데스크탑 또는 노트북 기기를 이용할 것을 권장하며, 모바일 기기 환경은 권장하지 않는다. 시스템은 _ 데이터베이스를 이용하여 필요한 데이터를 해당 데이터베이스에 저장할 수 있어야 한다.

3.5.2. Standards Compliance

시스템은 웹 어플리케이션으로 Python Django 을 이용해 개발되며, Django 코딩 표준 (Python style) 및 HTML 표준을 따른다. 변수 이름은 lower 카멜 케이스를 따르며, 함수와 데이터베이스는 파스칼 케이스를 따른다.

3.6. Software System Attributes

3.6.1. Reliability

품질 관리 및 테스트를 위해서 테스트케이스를 통해서 진행한다. 에러가 발생할 시 최근의 체크포인트로 이동한다.

3.6.2. Availability

사용자들이 시스템을 이용할 때마다 체크포인트가 기록되며, 시스템이 중단되었을 경우 저장된 체크포인트를 사용하여 시스템을 안정된 상태로 복구한다.

3.6.3. Security

사용자들은 시스템을 이용하기 전에 올바른 절차를 통해 인증을 받고, 시스템에 접속하기 위한 계정을 생성해야 한다. 시스템 관리자를 제외한 일반 사용자는 시스템 관리자만큼의 권한을 가질 수 없고, 사용자가 자신과 관련된 데이터와 다른 사용자의 프로필 정보와 순위 시스템만 볼 수 있으며, 시스템 데이터베이스에 직접적으로 접근할 수 없어야 한다.

3.6.4. Maintainability

시스템은 크게 사용자 인터페이스와 웹 서버로 구분되며 웹 서버는 계정 관리 시스템, 코드 개선 시스템으로 구분되어 각 시스템을 따로 관리할 수 있도록 개발된다. 코드의 가독성을 위해서 문서화 작업을 진행하고 각 함수에 대한 설명을 작성한다.

3.6.5. Portability

시스템은 Python Code 를 실행할 수 있어야 되며 Django 에 기본적으로 설치되어 있는 SQLite 데이터베이스를 사용할 수 있는 환경이어야 한다.

3.7. Organizing the Specific Requirements

3.7.1. Context Model

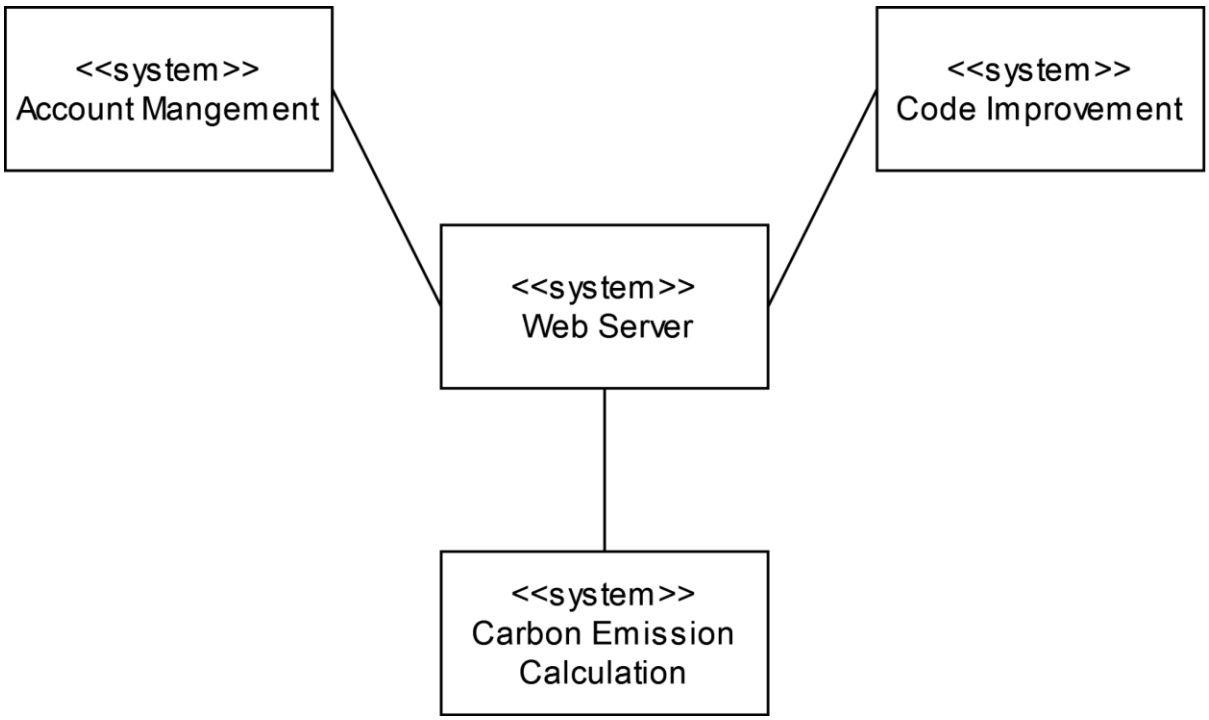


Figure 3: Context model

3.7.2. Process Model

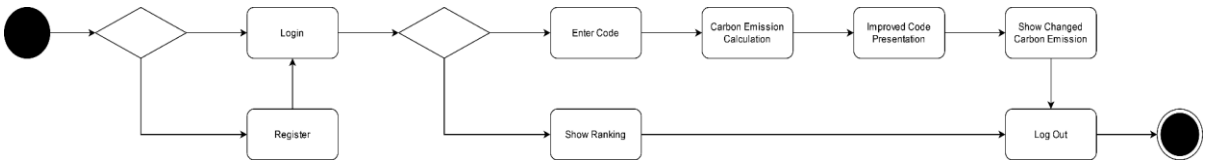


Figure 4: Process model

3.7.3. Interaction Model

Use Case Diagram 참조

3.8. System Architecture

이 절에서는 개발된 시스템 아키텍처에 대한 high-level 개요를 제시한다. 이는 각각의 서브 시스템과 그 구성요소를 밝히고 서브 시스템 간의 통신이 어떻게 이루어지는 지 명시한다.

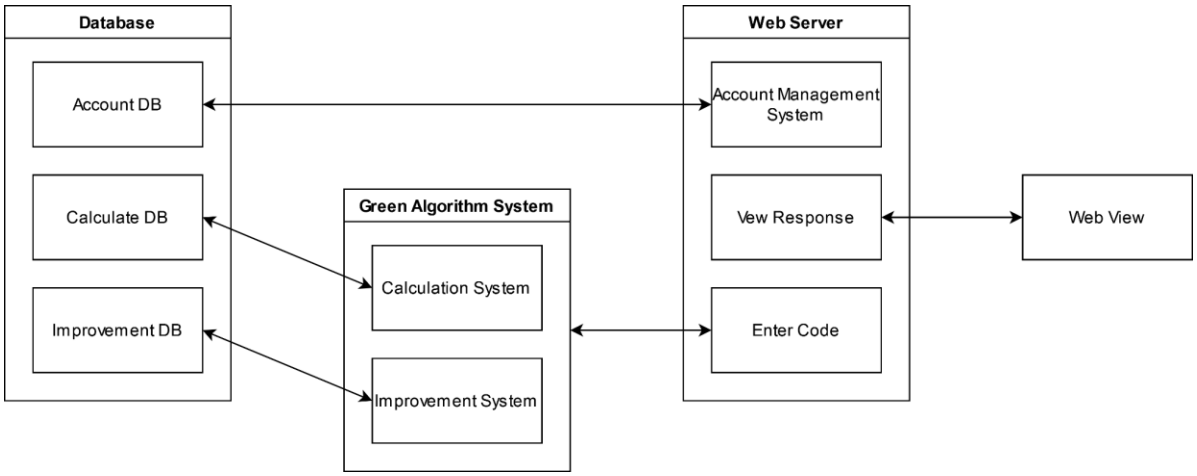


Figure 5: System architecture

4 Appendix

4.1. Software Requirements Specification

소프트웨어 요구사항 명세서 IEEE 권장사항 (IEEE Recommend Practice for Software Requirements Specifications, IEEE-Std-830)에 따라 작성되었다.