Requirement Specification

Green Code

소프트웨어 공학개론 10 조

2020312627 권서영

2019312104 김동현

2019310333 남윤성

2018313451 송재현

2018314520 전윤혁

2019314159 조영길

Contents

1. Introduction

1.1 Purpose

해당 문서는 프로그래밍 코드의 탄소 배출량을 계산하는 프로그램의 요구사항을 명시하고 있다. 해당 문서의 대상 독자는 프로젝트 개발 팀과 제품 사용자로, 프로그램의 요구사항을 명확히 하여 제품의 개발과 유지보수를 지원한다.

1.2 Scope

Green Code 는 입력된 프로그래밍 코드의 탄소 배출량을 계산하는 웹사이트이다. 기본적으로 사용자의 Java 코드를 입력으로 받아 해당 코드의 탄소 배출량을 계산하고, Java 외 다른 코드의 탄소배출량 계산은 지원하지 않는다. 사용자는 실행 환경을 임의로 설정할 수 없지만, 코드를 실행한 환경이 결과에 표시된다.

1.3 Definitions, acronyms, and abbreviations

Table 1. Acronyms & Abbreviations

PUE	Power Usage Effectiveness
PSF	Pragmatic Scaling Factor

Table 2. Definitions

Carbon footprint	제품 및 서비스의 전 과정에서 발생하는 탄소(온실가스)가 기후변화에 미치는 영향을 계량적으로 나타낸 지표
그린화 패턴	입력되는 코드의 탄소 배출량이 감소되는 패턴

1.4 References

- IEEE Recommended Practice for Software Requirements Specifications (IEEE Std 830-1998)

https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=720574

- Green Algorithms website
  https://www.green-algorithms.org/

- Green Algorithms github
  https://github.com/GreenAlgorithms/green-algorithms-tool

- 2023 Spring 41 class team 1
  https://github.com/skkuse/2023spring_41class_team1

- 2023 Spring 41 class team 10
  https://github.com/skkuse/2023spring_41class_team10

1.5 Overview

본 요구사항 명세서는 크게 네 파트로 구성되어 있다.

Introduction 파트에서는 Green Code 웹사이트 서비스의 목적과 주요 기능, 문서 문서 전반에서 사용되는 약어 및 용어에 대한 설명, 문서에 관련된 참고 자료가 설명되어 있다.

Overall Description 파트에서는 product perspective 에 대한 전반적인 설명을 제공하고, 시스템 인터페이스와 제품의 주요 기능을 설명하며, 고객 유형과 외부 요소 등 제품과 해당 요구 사항에 영향을 미치는 일반적인 요소들을 설명한다.

External Interface requirements 파트에서는 소프트웨어 시스템이 외부 입력 및 출력과 상호 작용하는 방법에 대한 세부 정보를 명시하는데, 이는 이름, 목적, 소스, 대상, 정확도, 측정 단위, 타이밍 및 다른 입력 및 출력과의 관계를 포함한다.

Appendix 파트에서는 본 요구사항 명세서 작성 시 참조한 외부 자료를 명시하고, 명세서의 생성 과정을 간략하게 보여준다.

2. Overall Description

2.1 Product perspective

소프트웨어 개발과 운영 과정에서 발생하는 탄소 배출은 점차 심각한 환경 부하로 작용하고 있으며, 이러한 환경 부하는 개발, 유지보수, 그리고 반복적 사용으로 누적된다. 현재, 탄소 배출로 인한 기후 및 환경 변화는 우리 사회에서 더욱 심각한 문제로 인식되고 있다. 이러한 문제를 해결하고 환경 부하를 감소시키기 위해, 개발자는 자신이 작성한 코드로 인한 탄소배출을 측정하고 개선하는 노력을 기울일 필요가 있다.

이에 따라, 이 제품은 주로 소프트웨어 개발 및 운영 과정에서 발생하는 탄소배출을 측정하는 목적으로 개발되었다. 소프트웨어의 개발, 유지보수, 및 반복적 사용으로 인한 탄소배출량을 측정하고 이를 개발자 및 환경 관리자에게 제공함으로써, 기후와 환경 변화로부터의 경감을 목표로 하고 있다.

이 제품은 사용자로부터 제공된 JAVA 코드를 실행하고 실행된 환경에서 소비된 전력, 탄소배출량 등을 계산한다. 더불어, 사용자가 입력한 추가적인 parameter, 예를 들어 국가, PUE, PSF 등과 같은 정보도 함께 고려하고 이렇게 수집된 정보는 탄소배출량 및 에너지 소비량을 계산하는 공식에 적용된다.

또한, 이 제품은 사용자에게 시각적으로 정보를 제공하여, 코드 실행으로 인한 환경 영향을 이해하기 쉽게 한다. 계산된 탄소배출량과 에너지 소비량에 대한 정보를 이용해 동등한 탄소배출량의 예시를 제공함으로써, 사용자는 자신의 코드가 얼마나 탄소를 방출했는지를 시각적으로 파악할 수 있다.

이 제품은 환경 보호와 탄소배출 감소를 위한 노력의 일환으로, 소프트웨어 개발과 운영에서 발생하는 탄소배출을 측정하고 개선하기 위한 도구로 개발되었다.

사용자는 자체 작성한 코드의 환경 영향을 이해하고, 더 친환경적인 소프트웨어 개발을 실현하는데 이 제품을 활용할 수 있다. 이는 환경과 기후 문제에 대한 인식을 높이게 해줄 것이다.

2.2 Product functions

2.2.1 JAVA 코드 및 parameter 입력

사용자로부터 코드를 입력 받으면, JAVA 를 컴파일하고 실행할 수 있는 Back-end 서버로 코드를 전달한다. 코드를 전달할 때, country, PUE, PSF 와 같은 parameter 들 역시 같이 전달한다.

2.2.2 실행 환경 정보 수집

Front-end 서버로부터 전달받은 코드와 parameter 들을 사용하여 Back-end 서버에서 처리한다. 코드의 경우 JAVA 를 실행할 수 있는 환경에서 컴파일 하여 runtime, cores, usage, memory 와 같은 값 정보를 수집한다.

2.2.3 탄소배출량 계산

Front-end 서버로부터 전달받은 parameter 와 Back-end 환경에서 컴파일 하여 얻은 실행 환경 정보들을 활용하여 탄소배출량과 에너지 소비량을 구하는 공식에 값을 대입하여 각각의 값을 구한다.

2.2.4 시각적 표현

공식으로 얻은 결과 탄소 배출량에 대한 구체적인 비유(자동차 탄소 배출량 등)를 제공하며, 사용자가 얼마나 탄소를 방출했는지 시각적으로 쉽게 이해할 수 있도록 한다.

2.3 User classes and characteristics

2.3.1 일반적인 Java 개발자

이 제품을 사용하려는 사용자는 JAVA 코드를 사용하는 개발자이다. 코드 작성, 디버깅 및 프로그램 실행에 JAVA 를 사용하는 개발자들은 자체 작성한 코드에 대한 탄소배출량을 확인하며 환경에 대한 문제를 인식하여 환경 친화적인 개발을 지향할 수 있다.

2.3.2 환경이 열악한 지역의 JAVA 개발자

이 제품은 환경 보호 및 탄소배출 감소에 관심이 있는 환경이 열악한 지역의 개발자에게 유용하다. 환경이 열악한 개발자들은 주변 환경에 대한 민감성을 가지고 있기 때문에 환경 파괴를 최소화하려는 노력이 필요하다. 따라서 환경 친화적인 개발을 추구하며 코드를 작성하고 그에 따른 탄소배출량 수치를 확인할 수 있다.

2.4 Design and implementation constraints

Green Code 과 관련된 디자인은 현 요구사항 명세서에 작성된 대로 설계될 것이다. 다음은 개발 및 제품 구현을 제한하는 제약사항이다.

- 하드웨어 제한: GPU 를 사용하지 않고 실행 가능한 코드여야 한다.
- 사용자가 실행가능한 JAVA 코드를 입력하여야 한다.
- 코딩을 위한 화면 크기의 제한으로 인해 휴대폰이 아닌 컴퓨터를 사용해야 한다.

- 자바스크립트가 활성화된 인터넷 브라우저를 사용해야 한다.

- 탄소 배출량에 대한 지식이 없는 사람들도 직관적으로 이해할 수 있게 동등한 탄소 배출량의 예시를 제공해야 한다.

2.5 Assumptions and dependencies

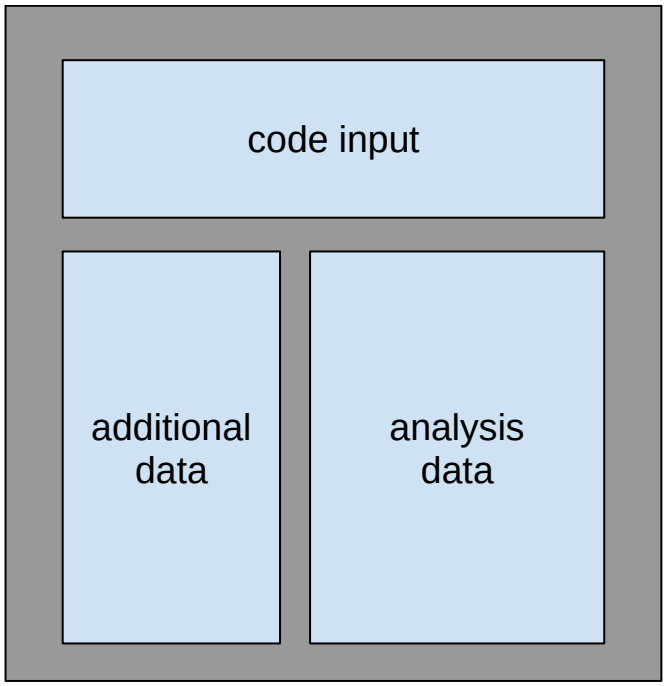
이 제품은 우선 JAVA 를 실행할 수 있는 환경이 요구된다. 코드 input 을 JAVA 로 받기 때문에 JAVA 언어를 실행시킬 수 없는 환경이라면 해당 제품을 사용할 수 없다. 또한 이 제품의 작동은 정상적으로 작동하는 코드를 받는다는 전제로 이루어진다. 이 제품의 목적이 실행된 코드의 탄소 배출량 측정이기 때문에, runtime 을 계산하기 위해 코드가 정상적으로 컴파일 되고 실행되어야 한다.

3. Specific Requirements

3.1. Design and implementation constraints

3.1.1. User Interface

이름	마우스 및 키보드를 통한 입력
목적/내용	키보드, 마우스를 사용한 사용자의 java code 입력 및 환경 설정
입력 주체/출력 목적지	사용자/웹페이지
범위/정확도/허용 오차	해당 없음
단위	텍스트 입력/버튼 클릭
시간/속도	해당 없음
타 입출력과의 관계	입력 code 를 서버로 보내 필요 정보 연산
화면 형식 및 구성	해당 없음
윈도우 형식 및 구성	해당 없음
데이터 형식 및 구성	텍스트
명령 형식	해당 없음
종료 메시지	해당 없음

이름	모니터를 통한 메인 화면 출력
목적/내용	사용자가 입력한 java code 의 탄소 배출량을 확인
입력 주체/출력 목적지	사용자/웹페이지
범위/정확도/허용 오차	해당 없음
단위	화면
시간/속도	해당 없음
타 입출력과의 관계	Java code 와 설정한 환경에 알맞은 정보 표기 후 사용자의 새로운 입력 대기
화면 형식 및 구성	 - Code input 란에 실행할 java 코드를 입력 넣음 - Additional data 란에 실행환경에 관한 정보를 선택/입력 넣음 - Analysis data 란에 해당 환경에서 실행한 java 코드가 배출한 탄소량을 다양한 시각화 정보로 표시하여 보여줌
윈도우 형식 및 구성	각 정보 혹은 입력란을 분리하여 표시
데이터 형식 및 구성	이미지, 텍스트
명령 형식	텍스트로 code 를 입력하며, 버튼을 클릭하여 탄소배출량을 확인함
종료 메시지	해당 없음

3.1.2. Hardware Interface

이름	시스템에서 사용 가능한 디바이스
목적/내용	키보드, 마우스를 사용한 사용자의 입력
입력 주체/출력 목적지	사용자/웹페이지
범위/정확도/허용 오차	해당 없음
단위	해당 없음
시간/속도	해당 없음
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	해당 없음
종료 메시지	해당 없음

3.1.3. Software Interface

이름	웹사이트
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	해당 없음
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

3.1.4. Communication Interface

이름	호스트 서버 - 클라이언트
목적/내용	입력받은 java code 와 환경 설정을 통해 각 탄소 배출량을 연산하고 다시 웹페이지로 전달
입력 주체/출력 목적지	클라이언트와 호스트서버
범위/정확도/허용 오차	해당 없음
단위	패킷
시간/속도	
타 입출력과의 관계	해당 없음
화면 형식 및 구성	해당 없음
윈도우 형식 및 구성	해당 없음
데이터 형식 및 구성	
명령 형식	send()콜에 의한 통신
종료 메시지	close()콜에 의한 소켓 종료

3.2. Functional Requirements

입력을 받아들이고, 출력을 처리하고, 생성하는 데 있어서 소프트웨어에서 해야 하는 기본 작업들을 설명한다.

3.2.1. Objective

이 장에서는 시스템의 구조, 속성 및 기능들을 설명하고 Green Algorithm Website 에서 각 구성 요소에 기대하는 책임을 설명한다.

3.2.2. Use Case

Table 01 Functional Requirement Abstraction

No.	Function Name	Function Description
F1	source_code_input	User(사용자)가 Green Algorithm 웹사이트에 자신의 소스코드를 입력할 수 있어야 한다.
F2	read_data	배출 탄소량 계산을 위해 필요한 변수에 해당하는 값을 csv 파일로부터 읽어 후에 계산식을 준비해야 한다. Carbon intensity (countries), PUE (Power Usage Effectiveness), PSF (Pragmatic Scaling Factor).
F3	hardware_spec	Green Algorithm 서버가 소스코드를 실행하는 하드웨어 서버 사양을 출력할 수 있어야 한다.
F4	countries_run	User 가 소스코드를 실행하는 나라를 선택해 입력할 수 있어야 한다.
F5	calculate_runtime	User 가 입력한 소스코드의 실행 시간 (runtime)을 계산할 수 있어야 한다.
F6	power_draw	Power draw for cores 와 power draw for memory 를 계산할 수 있어야 한다. Power draw for cores 는 csv 파일에서 값을 읽어오며, 사용자의 소스코드가

		cores 를 얼마나 사용하는지 (usage)를 계산할 수 있어야 한다.
F7	energy_needed	입력된 값들과 계수들을 이용해 얼마만큼의 에너지가 사용되었는지 계산할 수 있어야 한다. 계산식은 다음과 같다: Energy_needed = runtime * power_draw_for_cores * usage + power_draw_for_memory * PUF * PSF 그리고 계산한 필요한 에너지 값을 웹사이트에 보일 수 있어야 한다.
F8	carbon_footprint	소스코드의 탄소배출량을 계산할 수 있어야 한다. 계산식은 다음과 같다: Carbon_footprint = energy_needed * carbon_intensity 그리고 탄소배출량 값을 웹사이트에 보일 수 있어야 한다.
F9	reference	소스코드의 탄소배출량을 비교할 수 있는 참고값(reference)를 csv 파일에서 읽어올 수 있어야 한다.
F10	calc_tree	계산된 탄소 배출량이 얼마만큼의 나무의 훼손을 의미하는지 계산할 수 있어야 한다. 또한, 계산한 동량의 나무 훼손 값을 웹사이트에 보일 수 있어야 한다.
F11	calc_car	계산된 탄소 배출량이 자동차를 얼마만큼의 거리를 운영했을 때 배출되는지 값을 계산할 수 있어야 한다. 또한, 계산한 동량의 자동차 운행 거리를 웹사이트에 보일 수 있어야 한다.
F12	calc_plane	계산된 탄소 배출량이 비행기를 어마만큼의 거리를 운영했을 때 배출되는지 값을 계산할 수 있어야 한다. 또한, 계산한 동량의 자동차 운행 거리를 웹사이트에 보일 수 있어야 한다.

F13	calc_netflix	계산된 탄소 배출량이 넷플릭스를 한시간 시청했을 때 소모되는 전기량과 배출되는 탄소량과 어떻게 비례하는지 웹사이트에 보일 수 있어야 한다.
F14	calc_google	계산된 탄소 배출량이 구글에서 검색을 했을 때 소모되는 전기량과 배출되는 탄소량과 어떻게 비례하는지 웹사이트에 보일 수 있어야 한다.

Table 02 Use Case of "source_code_input"

Name	source_code_input
Actors	User, Operator
Description	User(사용자)가 Green Algorithm 웹사이트에 자신의 소스코드를 입력할 수 있어야 한다.
Data	Source code
Stimulus	User 가 소스코드를 입력 창에 입력하고 '확인' 버튼을 누른다.
Response	그린 알고리즘이 사용자가 입력한 소스코드를 실행하여 소스코드의 탄소 배출량과 그 배출량에 따른 예시 자료를 웹사이트에 보인다. 만약 소스코드가 Java 가 아니라면 오류창을 띄운다.

Table 03 Use Case of "read_data"

Name	read_data
Actors	Operator
Description	operator 가 데이터 파일에서 필요한 계수 값들을 끌어와 다른 함수들의 필요한 값에 접근할 수 있도록 준비한다.
Data	Carbon Intensity: datav2.1/CI_aggregated.csv

	PUE (Power Usage Effectiveness): data/v2.1/default_PUE.csv PSF (Pragmatic Scaling Factor): blob/master/app.py#L399
Stimulus	사용자가 소스코드를 입력하고 확인창을 누르면 바로 실행해, 탄소배출량 계산을 준비한다.
Response	사용자에게 특정되는 확인메세지는 없다. Constructor 역할으로, operator 는 특정 변수값들이 initialize 된다. 따로 확인 메세지는 없다.

Table 04 Use Case of "hardware_spec"

Name	hardware_spec
Actors	Operator
Description	Green Algorithm 서버가 소스코드를 실행하는 하드웨어 서버 사양을 출력할 수 있어야 한다. 코어의 갯수, 모델을 뜻한다.
Data	CPU: datav2.1/TDP_cpu.csv
Stimulus	사용자가 소스코드를 입력하고 확인 버튼을 누르면 함수가 실행된다.
Response	그린 알고리즘 웹사이트에 소스코드가 실행되는 하드웨어 사양을 보인다.

Table 05 Use Case of "countries_run"

Name	countries_run
Actors	User, Operator
Description	User 가 소스코드를 실행하는 나라를 선택해 입력할 수 있어야 한다.
Data	Carbon Intensity: dataV2.1/CI_aggregated.csv
Stimulus	사용자가 나라를 입력하고 확인 버튼을 누르면 함수가 실행된다.
Response	사용자에게 보이는 확인 메세지는 없다. Operation 안에서 carbon

	intensity 특정 변수에 값이 할당된다.

Table 06 Use Case of "calculate_runtime"

Name	calculate_runtime
Actors	Operator
Description	User 가 입력한 소스코드의 실행 시간 (runtime)을 계산할 수 있어야 한다.
Data	User input = source code
Stimulus	사용자가 소스코드를 입력하고 확인 버튼을 누르면 calculate_runtime 함수가 바로 실행된다.
Response	그린 알고리즘 웹사이트에 실행시간 값을 보인다. 또한 operation 내부에서는 실행시간을 의미하는 변수에 계산된 값을 할당한다. H / M / S : Hours / Minutes / Seconds 형식으로 표현한다.

Table 07 Use Case of "power_draw"

Name	power_draw
Actors	Operator
Description	Power draw for cores 와 power draw for memory 를 계산할 수 있어야 한다. Power draw for cores 는 csv 파일에서 값을 읽어오며, 사용자의 소스코드가 cores 를 얼마나 사용하는지 (usage)를 계산할 수 있어야 한다.
Data	W/GB 와 Watt metrics 를 사용한다. Memory power: data/V2.1/referenceValues.csv Power draw for cores: data/V2.1/TDP_cpu.csv Usage = Number of cores: data/V2.1/TDP_cpu.csv
Stimulus	사용자가 탄소 배출량 계산하고 싶은 소스코드를 입력창에 입력하고 확인

	버튼을 누르면 함수가 실행된다.
Response	사용자에게 바로 보여지는 확인 메세지는 없다. Operation 내부에서 power draw for cores, power draw for memory, usage 를 뜻하는 변수들에게 값을 부여한다.

Table 08 Use Case of "energy_needed"

Name	energy_needed
Actors	Operator
Description	입력된 값들과 계수들을 이용해 얼마만큼의 에너지가 사용되었는지 계산할 수 있어야 한다. 계산식은 다음과 같다: Energy_needed = runtime * power_draw_for_cores * usage + power_draw_for_memory * PUF * PSF 그리고 계산한 필요한 에너지 값을 웹사이트에 보일 수 있어야 한다.
Data	Runtime = calculate_runtime 함수 Power draw for cores = power_draw 함수 Usage = power_draw 함수 Power draw for memory = power_draw 함수 PUF = read_data 함수 PSF = read_data 함수
Stimulus	사용자가 소스코드를 입력창에 입력하고 확인 버튼을 누른다. 그 다음 Data 에 명시된 함수들의 차례가 다 지나고 실행된다.
Response	그린 알고리즘 웹사이트에 계산된 값을 보인다.

Table 09 Use Case of "carbon_footprint"

Name	carbon_footprint
Actors	Operator
Description	소스코드의 탄소배출량을 계산할 수 있어야 한다. 계산식은 다음과 같다: Carbon_footprint = energy_needed * carbon_intensity 그리고 탄소배출량 값을 웹사이트에 보일 수 있어야 한다.
Data	Energy needed = energy_needed 함수 Carbon_intensity = countries_run 함수
Stimulus	사용자가 입력창에 소스코드를 입력하고 확인 버튼을 누른다. 그 후 순차적으로 함수들이 실행되고 마지막으로 energy_needed 함수가 실행된 후 실행된다.
Response	그린 알고리즘 웹사이트에 탄소배출량을 보인다.

Table 10 Use Case of "reference"

Name	reference
Actors	Operator
Description	소스코드의 탄소배출량을 비교할 수 있는 참고값(reference)를 csv 파일에서 읽어올 수 있어야 한다.
Data	data/V2.1/referenceValues.csv
Stimulus	사용자가 탄소배출량을 계산하고픈 소스코드를 입력하고 확인 버튼을 누른다. 그 다음 순차적으로 함수들이 실행된다. 마지막으로 carbon_footprint 함수가 실행되고 그 다음 실행된다.
Response	사용자에게 즉각적으로 보이는 확인 메세지는 없다. 그러나 reference 값들을 대변하는 변수들에게 값을 부여한다.

Table 11 Use Case of "calc_tree"

Name	calc_tree
Actors	Operator
Description	계산된 탄소 배출량이 얼마만큼의 나무의 훼손을 의미하는지 계산할 수 있어야 한다. 또한, 계산한 동량의 나무 훼손 값을 웹사이트에 보일 수 있어야 한다.
Data	data/V2.1/referenceValues.csv
Stimulus	사용자가 소스코드를 입력창에 입력하고 확인 버튼을 누른다. 그 다음 순차적으로 함수가 실행된다. 마지막으로 reference 함수가 실행되고 그 다음 실행된다.
Response	그린 알고리즘 웹사이트에 1 년 단위로 자란 나무들의 몇 개 정도가 훼손되었는지 보이는 출력창을 띄운다.

Table 12 Use Case of "calc_car"

Name	calc_car
Actors	Operator
Description	계산된 탄소 배출량이 자동차를 얼마만큼의 거리를 운영했을 때 배출되는지 값을 계산할 수 있어야 한다. 또한, 계산한 동량의 자동차 운행 거리를 웹사이트에 보일 수 있어야 한다.
Data	data/V2.1/referenceValues.csv
Stimulus	사용자가 소스코드를 입력창에 입력하고 확인 버튼을 누른다. 그 다음 순차적으로 함수가 실행된다. 마지막으로 reference 함수가 실행되고 그 다음 실행된다.
Response	그린 알고리즘 웹사이트에 계산된 탄소 배출량과 동일한 탄소 배출량이 몇 KM 를 달린 자동차와 비슷하다는 결과를 보이는 출력창을 띄운다.

Table 13 Use Case of “calc_plane”

Name	calc_plane
Actors	Operator
Description	계산된 탄소 배출량이 비행기를 어마만큼의 거리를 운영했을 때 배출되는지 값을 계산할 수 있어야 한다. 또한, 계산한 동량의 자동차 운행 거리를 웹사이트에 보일 수 있어야 한다.
Data	data/V2.1/referenceValues.csv
Stimulus	사용자가 소스코드를 입력창에 입력하고 확인 버튼을 누른다. 그 다음 순차적으로 함수가 실행된다. 마지막으로 reference 함수가 실행되고 그 다음 실행된다.
Response	그린 알고리즘 웹사이트에 비행기를 몇 시간 운행하면 입력된 소스코드가 만드는 탄소 배출량을 배출하는지 비교하는 결과를 보이는 출력창을 띄운다.

Table 14 Use Case of “calc_netflix”

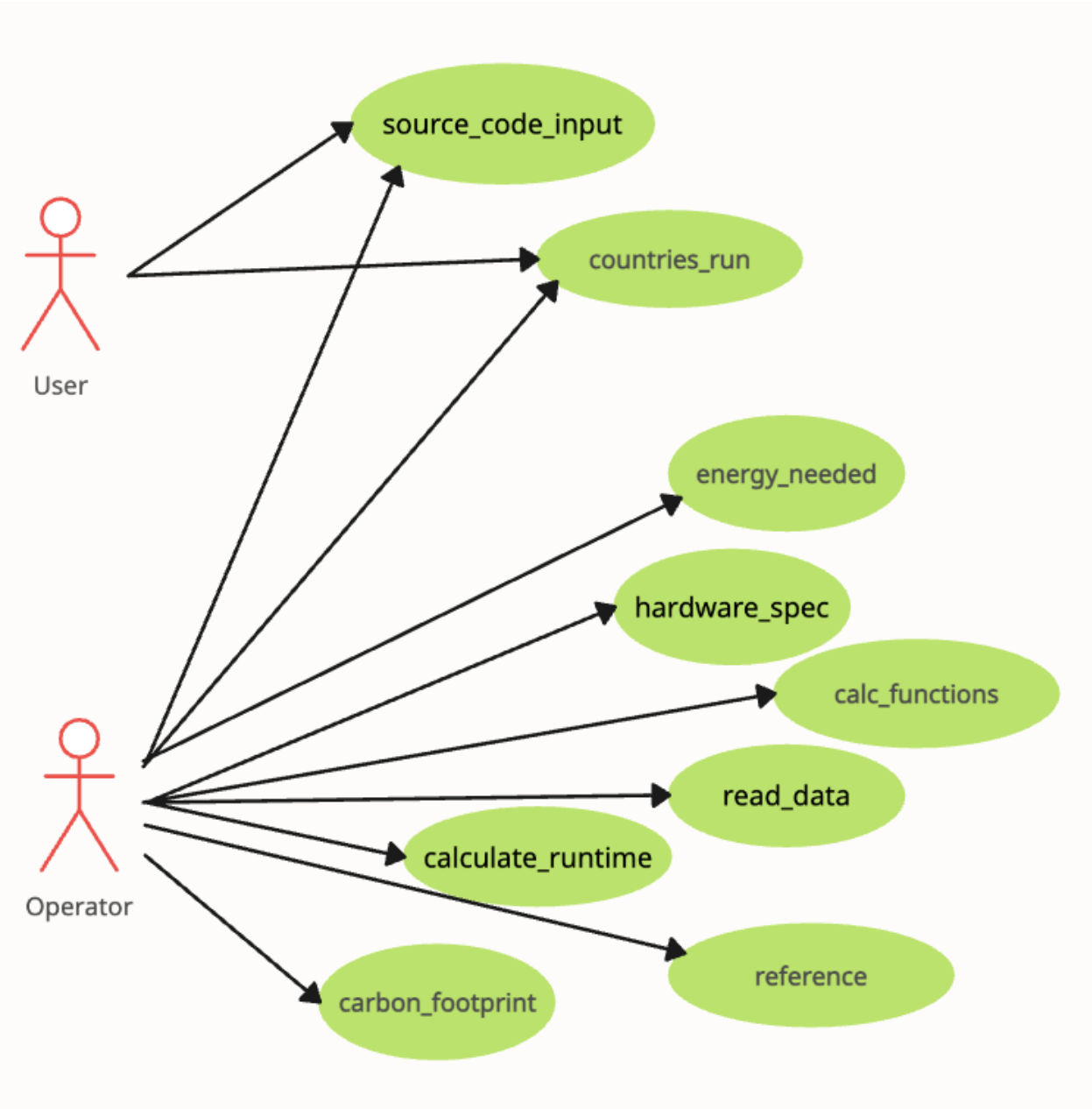
Name	calc_netflix
Actors	Operator
Description	계산된 탄소 배출량이 넷플릭스를 한시간 시청했을 때 소모되는 전기량과 배출되는 탄소량과 어떻게 비례하는지 웹사이트에 보일 수 있어야 한다.
Data	data/V2.1/referenceValues.csv
Stimulus	사용자가 소스코드를 입력창에 입력하고 확인 버튼을 누른다. 그 다음 순차적으로 함수가 실행된다. 마지막으로 reference 함수가 실행되고 그 다음 실행된다.
Response	그린 알고리즘 웹사이트에 넷플릭스를 한시간 단위로 얼마나 보여야 입력된 소스코드가 배출하는 탄소 배출량을 배출하는지 결과값을 비교해

	보이는 출력창을 띄운다.

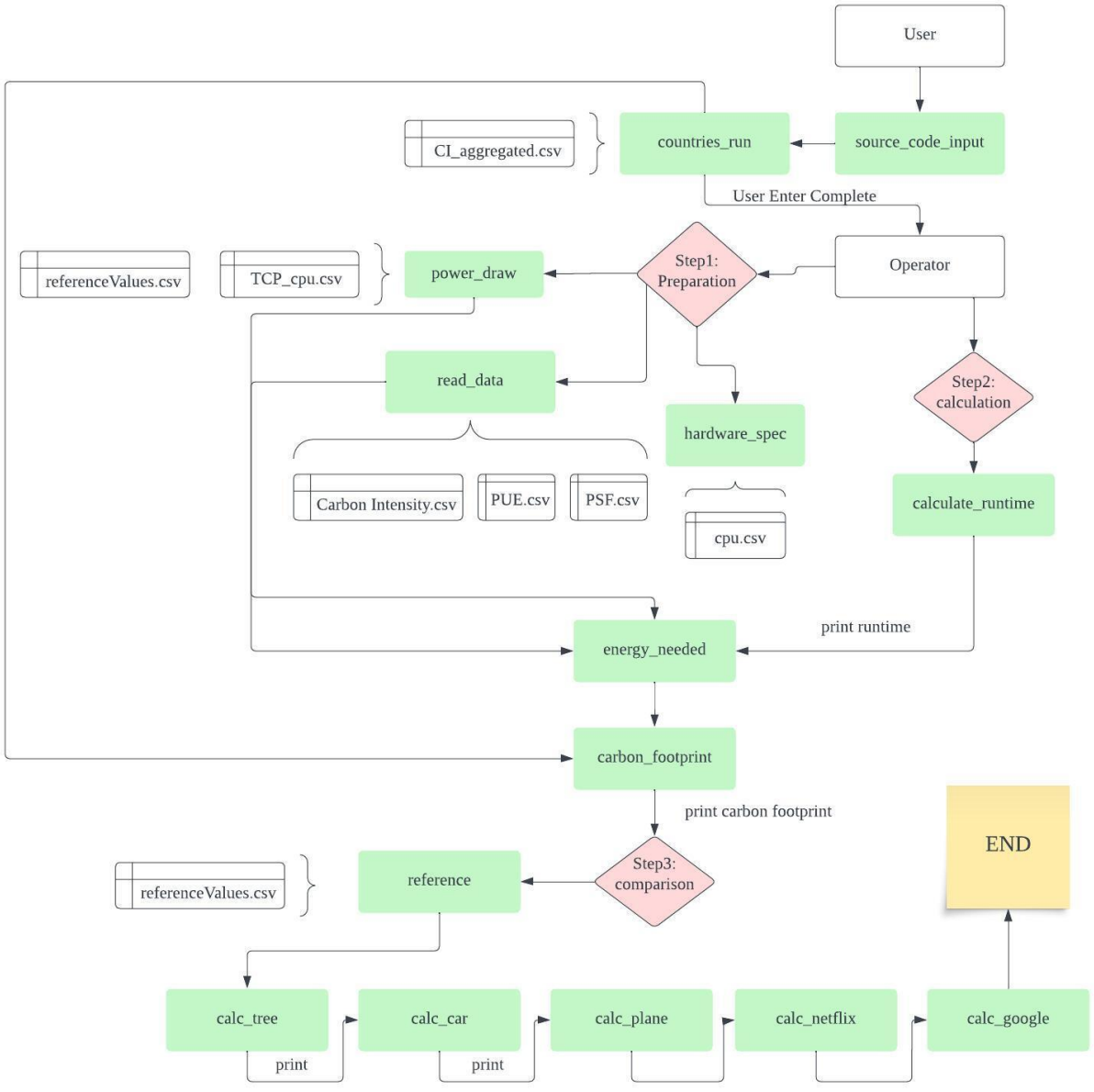
Table 15 Use Case of “calc_google”

Name	calc_google
Actors	Operator
Description	계산된 탄소 배출량이 구글에서 검색을 했을 때 소모되는 전기량과 배출되는 탄소량과 어떻게 비례하는지 웹사이트에 보일 수 있어야 한다.
Data	data/V2.1/referenceValues.csv
Stimulus	사용자가 소스코드를 입력창에 입력하고 확인 버튼을 누른다. 그 다음 순차적으로 함수가 실행된다. 마지막으로 reference 함수가 실행되고 그 다음 실행된다.
Response	그린 알고리즘 웹사이트에 구글 서버에서 검색을 몇 개를 해야 입력된 소스코드가 만드는 탄소 배출량에 똑같은 결과를 보이는지 비교하는 값을 출력창에 띄운다.

3.2.3 Use Case Diagram



3.2.4 Data Flow Diagram



3.3 Performance Requirements

아래는 본 시스템의 성능 요구사항에 관한 것이다. 예측에 기반한 내용이며 실제 구현시에 달라질 수 있다.

3.3.1 Static Numerical Requirement

- 서버 응답 시간은 사용자의 요청을 받고 처리하는 데 최대 1 초 이내로 제한된다.
- 최소한 동시에 500 명의 사용자 요청을 처리할 수 있어야 한다.
- 웹 사이트는 365 일 24 시간 가동되며 연간 가용성은 최소 99.9%를 유지해야 한다.
- 최소 100%의 확장성을 가질 수 있어야 하며, 이 과정에서 성능 저하는 10% 미만으로 유지되어야 한다.
- 모든 사용자 데이터는 256 비트 암호화를 통해 보호되며, https 프로토콜을 사용하여 모든 데이터 전송이 암호화된다.

3.3.2 Dynamic Numerical Requirement

- 서버 응답 시간은 피크 시간 동안에도 1 초를 넘지 않아야 한다. 여기서 피크 시간은 하루 중 사용자가 가장 많이 접속하는 시간대를 의미한다.
- 사용자 수가 증가함에 따라 적절하게 처리 용량을 조정할 수 있어야 한다.
- 웹사이트의 연간 가용성은 최소 99.9%를 유지해야 하지만, 피크 시간 동안에는 100% 가용성을 유지해야 한다.
- 사용자 수가 증가함에 따라 시스템의 확장성 또한 증가해야 한다
- 보안 위협이 증가하는 경우, 그에 따라 보안 수준을 높여야 한다.

3.4 Logical Database Requirements

데이터 베이스의 사용이 불필요하여 사용되지 않는다.

3.5 Design Constraints

3.5.1 Physical design constraints

모바일 기기에서의 사용을 고려하지 않으며, 최적화되지 않았다. 또한 Chrome, Firefox, Safari, Edge 이외의 브라우저와의 호환성을 보장하지 않는다.

3.5.2 Standards compliance

OWASP(Open Web Application Security Project)와 같은 보안 표준을 준수하고, GDPR(General Data Protection Regulation) 및 기타 관련 규정에 따라 개인 정보 보호를 보장한다. W3C 와 같은 웹 표준 및 웹 접근성 규칙을 준수하여 사용자가 웹 사이트를 쉽게 이해하고 이용할 수 있도록 한다.

3.6 Software system attributes

3.6.1 Reliability

- 사용자의 잘못된 입력이나 예상치 못한 상황에 대처할 수 있어야 한다.
- 계산이 정확하고 일관적이어야 한다.

3.6.2 Availability

- 중단을 대비하여 특정 시점에 현재 상태를 저장하고, 이를 복구에 사용한다.

- 예기치 않은 오류나 중단시에 이전의 안정적인 상태로 복원한다.

3.6.3 Security

- 데이터 검증 및 처리를 통해 악의적인 코드를 통한 공격을 방어할 수 있다.

3.6.4 Maintainability

- 코드를 독립적인 모듈 또는 구성 요소로 분리하여 유지 보수에 드는 비용을 줄인다.
- 코드의 복잡성을 최소화하고 가독성을 높이기 위해 적절한 주석 및 가이드라인을 사용한다.

3.6.5 Portability

- 여러 웹 브라우저에서 일관된 방식으로 작동할 수 있도록 코드를 작성해야 한다.
- 데스크톱, 노트북의 화면 크기에 대응하도록 반응형 디자인을 구현한다.
- 웹 표준을 준수하여 웹 사이트를 구축해야 한다. 이는 웹 접근성 및 다양한 플랫폼에서의 호환성을 보장한다.
- 다양한 환경에서의 테스트와 검증을 통해 웹 사이트의 이식성을 확인하고 문제를 해결해야 한다.

3.7 Overall Requirements

아래는 본 시스템의 몇 가지 비기능적 요구사항에 관련된 내용이다. 제품 요구사항, 조직 요구사항, 외부 요구사항으로 나누어 기술된다.

3.7.1 Product Requirements

제품 요구사항은 시스템 실행 중 작동 방식에 대해 설명한다.

- 빠른 실행과 응답 시간을 제공해야 한다. 코드 입력부터 탄소 배출량 산정 및 결과 출력까지의 시간을 최소화해야 한다.
- 다양한 양과 복잡도의 코드를 처리할 수 있어야 한다.

3.7.2 Organizational Requirements

조직 요구사항은 조직 내부의 프로세스, 팀 협업, 관리 방법 등에 대해 설명한다.

- Git 을 통한 버전 관리와 팀원 간의 협업으로 효과적인 개발 프로세스를 유지하는 데 집중해야 한다.
- 테스트 프로세스를 도입하여 지속적으로 시스템 품질을 관리해야 한다.

3.7.3 External Requirements

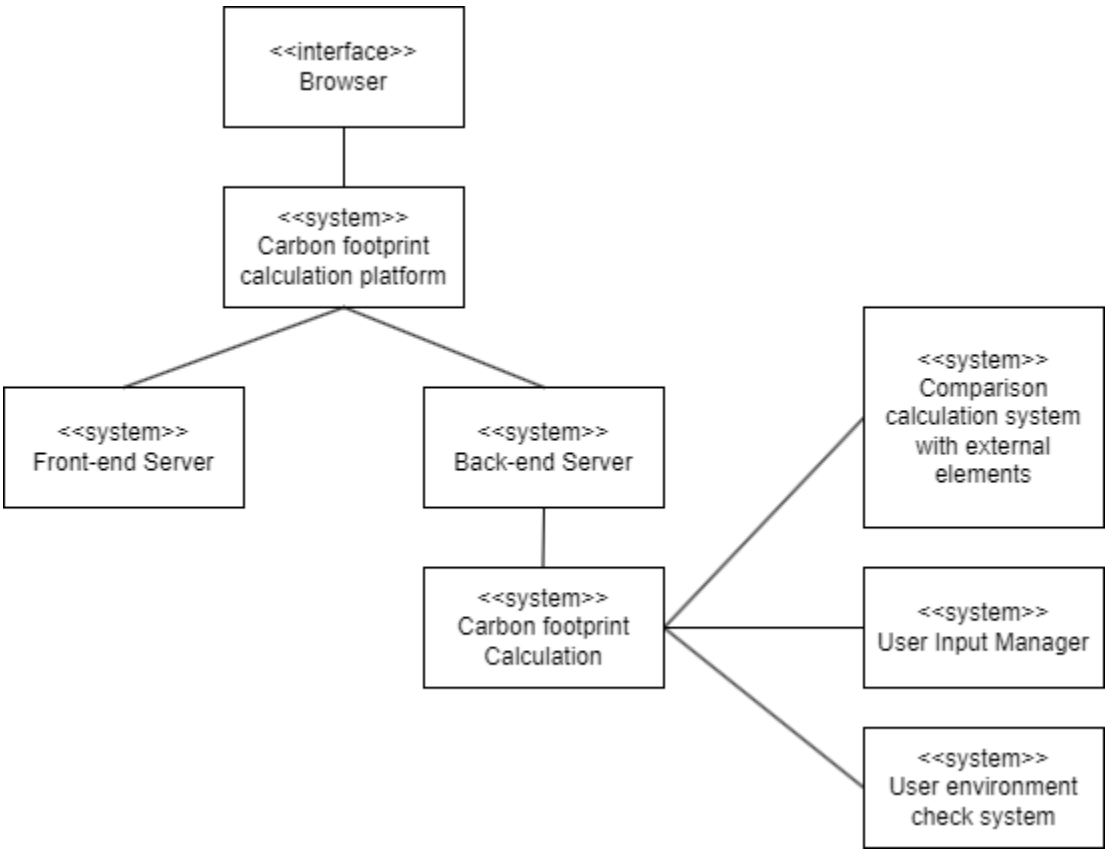
외부 요구사항은 외부 환경과 상호작용하기 위해 준수해야 하는 외부적인 규정, 규제에 대해 설명한다.

- OWASP 의 보안 가이드라인을 준수하고, 또한 GDPR 에 따라 사용자의 개인정보 보호를 보장한다
- Chrome, Firefox, Safari, Edge 브라우저와의 호환성을 보장하여 사용자가 쉽게 시스템에 접근할 수 있도록 한다.

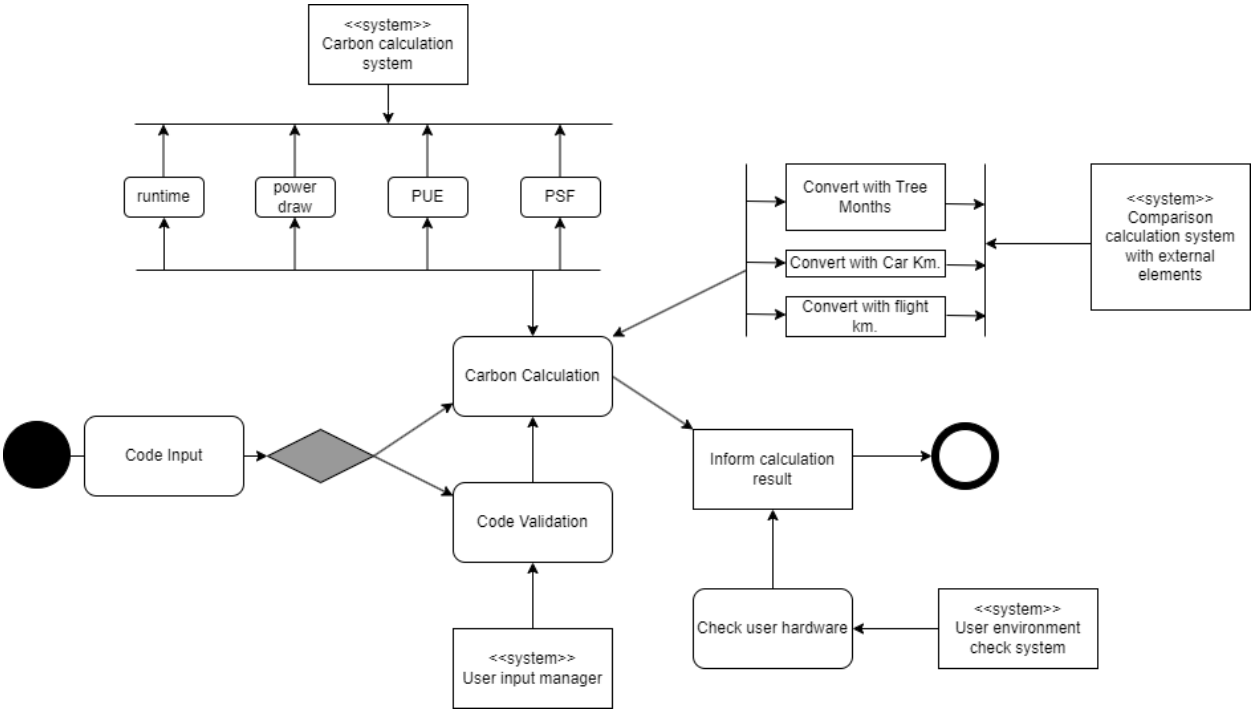
3.8. Organizing the Specific Requirements

이 구간에서는 Unified Modeling Language(UML) 및 표 형식 기반의 그래픽 표기법을 사용하여 시스템 모델을 설명한다. 시스템 모델은 시스템, 서브 시스템 간의 관계를 설명한다.

3.8.1. Context Model



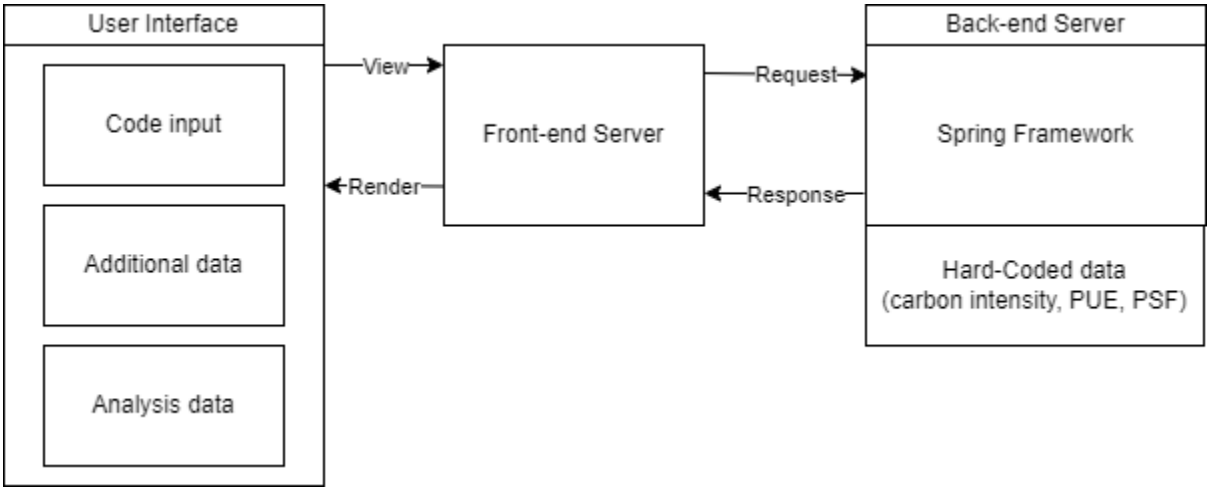
3.8.2. Process Model



3.8.3. Interaction Model

Use Case Diagram 참조

3.8.4 System Architecture



3.9 Evolution of Hardware and Change of User Requirements

Green Code 의 장점은 사용자 로그인과 같은 부수적인 작업 없이 간단하게 코드의 탄소 배출량을 측정할 수 있다는 점이다. 하지만 특정 하드웨어 환경에서의 코드 탄소 배출량을 측정하거나, Java 이외의 코드에 대한 탄소 배출량을 측정하려는 사용자 요구가 발생할 수 있다. Green Code 는 이렇게 서비스의 범용성 면에 초점을 맞춰 발전해 갈 것이다.

4. Appendixes

4.1. Software Requirements Specification

본 요구사항 명세서는 IEEE Recommend Practice for Software Requirements Specifications, IEEE-Std-830)의 형식에 따라 작성되었다.

4.2 Document History

Document History			
Date	Version	Description	Writer
10/29	V1.00	Introduction Context Model Process Model System Architecture Appendixes	전윤혁
10/29	V1.00	Product perspective Product functions User classes and characteristics Design and implementation constraints Assumptions and dependencies	남윤성
10/29	V1.00	Design and implementation constraints User Interface Hardware Interface Software Interface Communication Interface	김동현
10/29	V1.00	Functional Requirements	권서영

		Objective Use Case Use Case Diagram Data Flow Diagram	
10/29	V1.00	Static Numerical Requirement Dynamic Numerical Requirement Physical design constraints Standards compliance Software system attributes Overall Requirements	송재현