

Software Requirements Specification

소스코드 탄소 배출량 측정 시스템

SWE3002-41: Prof. 이은석

Software Requirements Specification

소스코드 탄소 배출량 측정 시스템

by

차승일 김민수 임소현 정단호 이시혁 장영우 유규환

TEAM 7

Instructor:	이은석
Teaching Assistant:	김소현, 김영경, 김진영, 최동욱, 허진석
Document Date:	29 October, 2023
Faculty:	SungKyunKwan University

Contents

1	Introduction	1
1.1	Purpose.....	1
1.2	Scope.....	1
1.3	Definitions, Acronyms, and Abbreviation.....	2
1.4	References.....	2
1.5	Overview.....	3
2	Overall Description	4
2.1	Product Perspective.....	4
2.1.1	System Interfaces.....	4
2.1.2	User Interfaces.....	4
2.1.3	Software Constraints.....	4
2.1.4	Hardware Constraints.....	5
2.2	Product Functions.....	6
2.2.1	코드검사.....	6
2.2.2	코드의 탄소 배출량 계산.....	6
2.2.3	수치자료, 이미지 자료를 통해 결과 전달.....	6
2.3	User Characteristics.....	7
2.3.1	Java Developer 및 system manager.....	7
2.3.2	Java 학습자.....	7
2.3.3	Java open source 사용 예정자.....	7
2.4	Constraints.....	7
2.5	Assumptions and Dependencies.....	7
3	Specific Requirements	8
3.1	External Interface Requirements.....	8
3.1.1	User Interface.....	8
3.1.2	Hardware Interface.....	10
3.1.3	Software Interface.....	11
3.1.4	Communication Interface.....	11
3.2	Functional Requirements.....	12
3.2.1	Use Case.....	12
3.2.2	Use Case Diagram.....	13
3.3	Performance Requirements.....	14
3.3.1	Static Requirement.....	14
3.3.2	Dynamic Requirement.....	14
3.4	Design Constraints.....	15
3.4.1	Physical design constraints.....	15
3.4.2	Standards compliance.....	15
3.5	Software System Characteristics.....	15
3.5.1	Product Requirements.....	15
3.5.2	Organizational Requirements.....	15

3.6	Organizing the Specific Requirements.....	16
3.6.1	Context Model.....	16
3.6.2	Process Model.....	16
3.6.3	Interaction Model.....	17
3.7	System Architecture.....	17
3.8	System Evolution.....	17
3.8.1	Limitation and Assumption.....	17
3.8.2	Evolution of Hardware and Change of User Requirements.....	18
4	Supporting Information	19
4.1	Software Requirements Specification.....	19
4.2	Document History.....	19

List of Figures

3.1	Use Case Diagram	13
3.2	Context Model	16
3.3	Overall Process Model	16
3.4	System Architecture	17

List of Tables

1.1	Table of acronyms and abbreviations	1
1.2	Table of terms and definitions	1
3.1	User interface of input processing	6
3.2	User interface of main page	7
3.3	User interface of register	8
3.4	Hardware interface of applicable device for the system	8
3.5	Software interface of applicable device for the system	9
3.6	Communication interface of applicable device for the system	9
3.7	Use Case of Calculate	10
3.8	Use Case of Statistics	11
3.9	Problem	14

1

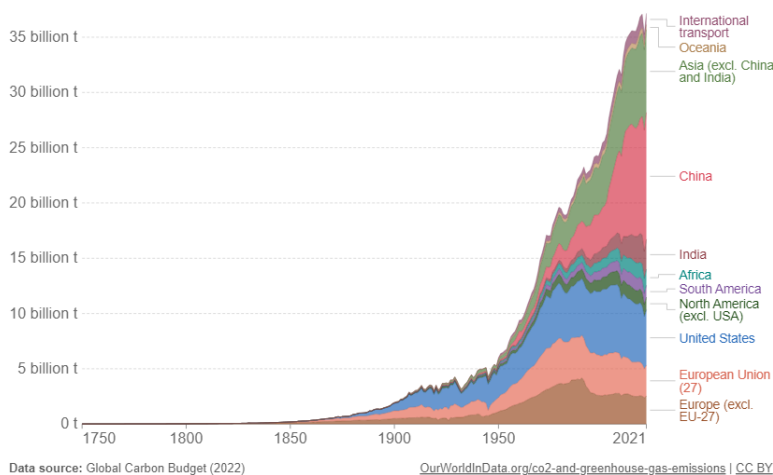
Introduction

1.1. Purpose

탄소 배출량이 증가함에 따라 이산화탄소 농도가 증가하여 온실 효과가 급증하고 있다.

Annual CO₂ emissions by world region

This measures fossil fuel and industry emissions. Land use change is not included.



따라서 본 시스템은 탄소 배출량을 측정하여 사용자에게 UI를 통해 제공해 줌으로 인해 사용자 코드의 탄소 배출량을 직관적으로 확인할 수 있도록 하는 것을 목표로 한다. Frontend로 코드를 입력받고 탄소배출량의 수치를 제공하고, 이를 교통수단에 비유하여 표현하고자 한다.

1.2. Scope

Including :

- 1) Details about your algorithm 부분에 사용자 PC의 스펙을 입력한다.
- 2) 본인의 Java code를 입력한다.
- 3) 출력된 탄소 배출량을 확인한다.

기술적 제약 사항 :

- 1) 본 시스템은 front-end와 back-end로 나누어 제작된다. Front-end로는 React를 주축으로 하고 html, css를 보조적으로 사용하고자 하며, back-end로는 Spring을 사용한다.
- 2) 본 시스템의 제약 사항은 Chrome 10.0.648.127 이상 등이 있으며 자세한 사항은 2.1의 software interfaces, hardware interfaces에 명시하였다.

1.3. Definitions, Acronyms, and Abbreviation

Acronyms & Abbreviations	Explanation
탄소배출량	특정 활동, 프로세스, 장치, 조직 또는 국가에서 발생한 온실가스의 양을 나타내는 측정치

Table 1.1: Table of acronyms and abbreviations

Terms	Definitions
PC 사양 정보	CPU, GPU, Core 개수 등 사용자 PC의 사양에 대한 구체적 수치
IDE 창	Integrated Development Environment의 약자로 소프트웨어 개발을 위한 하나의 통합된 환경을 제공하는 SW application을 의미
Usecase diagram	시스템이 제공하고 있는 기능 및 그와 관련된 외부요소를 사용자의 관점에서 표현하는 다이어그램
context model	컨텍스트 데이터가 구조화되고 유지되는 방식
Process model	프로세스 작업 단계 및 순서, 각 단계 작업 수행의 제약사항이나 조건을 모아 놓은 것

Table 1.2: Table of terms and definitions

1.4. References

Overall	IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications, In IEEEExplore Digital Library
github 주소	https://github.com/skkuse/2023fall_41class_team7
In 1.1. purpose	https://www.researchgate.net/figure/Annual-total-CO2-emissions-by-world-region-source-World-Bank_fig3_357548495
In 2.1.3 Software constraints	https://learn.microsoft.com/ko-kr/deployedge/microsoft-edge-supported-operating-systems
In 2.1. 4 Hardware constraints	https://support.microsoft.com/ko-kr/windows/windows-7-%EC%8B%9C%EC%8A%A4%ED%85%9C-%EC%9A%94%EA%B5%AC-%EC%82%AC%ED%95%AD-df0900f2-3513-a851-13e7-0d50bc24e15f
In 2.1.3 Software constraints	https://support.google.com/chrome/a/answer/7100626?hl=ko

1.5. Overview

본 소프트웨어 요구사항 명세서는 네 챕터로 구성되어 있다. 첫번째 챕터에서는 Team7이 제작하고자 하는 탄소배출량 측정 프로그램의 **purpose, Scope, Definition & Acronyms & Abbreviation, References**를 제시한다.. 두번째 챕터에서는 시스템 인터페이스 및 기능, 다른 시스템과의 상호작용을 포함한 **product perspective**에 대한 전반적인 설명 부분이다. 세번째 챕터에서는 외부 인터페이스, 기능 등을 포함한 자세한 요구사항 명세화를 진행한다. 네번째 챕터는 본 명세서가 어떤 기준이나 문서를 참고하였는지 알려준다. 팀의 모든 멤버는 각자 담당한 부분에 대해 역할을 수행하였고, 공정한 분배가 이루어졌다.

2

Overall Description

2.1. Product Perspective

탄소 배출 증가, 이산화탄소 농도 증가에 따른 온실 효과 증가와 전 세계적인 이상 기후 현상에 따라 코드를 작성하는 모든 이에게 본인이 작성한 코드의 탄소 배출량을 시각적, 직관적으로 인지시켜 에너지 효율적, 자원 최적화 코드 구현을 촉진하기 위해 설계되었다. 회원가입이나, 인증 절차 없이 코드를 작성 하는 누구나 사이트에 접속하여 본인이 작성한 **Java**코드의 탄소 배출량을 확인 할 수 있고, 환경과 관련된 긍정적 **UI/UX**를 통해 사용자들로 하여금 환경에 대해 관심과 경각심을 가지게 할 수 있다.

2.1.1. System Interfaces

본인 **Java**코드의 탄소배출량을 확인하고자 하는 **User**는 사이트에 접속하여 입력 칸에 본인의 코드를 입력하고 계산 버튼을 누른다. 코드를 입력 받은 시스템은 **run time error**이 없는 실행가능한 **Java**코드임을 확인하고 코드의 탄소 배출량을 계산 및 결과를 수치적 자료와 이미지 자료를 통해 **User**에게 전달한다.

2.1.2. User Interfaces

User가 사용하는 전자기기의 디스플레이를 통해 제공한다. 탄소 배출량 계산 사이트에서 코드 입력 칸을 통해 **User**는 코드를 입력하고 계산 버튼을 통해 결과를 수치와 이미지를 통해 디스플레이 상에 보여준다.

2.1.3. Software Constraint

본 시스템은 웹 어플리케이션으로 **Edge** 또는 **Chrome** 브라우저를 통해 네트워크에 접속이 가능한 모든 전자기기를 통해 이용이 가능하며 브라우저 접속을 위한 운영체제 사양은 아래와 같다.

Windows 7 이상

Mac Os 10.15 이상

Linux 64bit Ubuntu 18.04 이상, Debian 10 이상, Fedora Linux 32이상

Windows의 경우 IOS 14이상, ARM 기반 Android Kitkat 4.4이상 실행 디바이스

Chrome의 경우 Android 7.0 Nougat

2.1.4. Hardware Constraint

명시된 software Interface를 만족하기 위해 권장되는 하드웨어 사양은 다음과 같다.

CPU : Intel Pentium4 1GHz 이상

메모리(RAM) : 1GB(32bit), 2GB(64bit)이상

저장장치(HDD) : 20GB 이상

2.2. Product Functions

2.2.1. 코드 검사

User가 입력한 코드가 Run time error가 없는 실행 가능한 자바 코드인지 확인을 한다. 만약, 적절한 자바코드가 아니라면 User에게 에러 메시지를 전달하고 코드 재입력 할 것을 명시해준다.

2.2.2. 코드의 탄소 배출량 계산

코드 검사를 통과한 User의 Java코드가 배출하는 탄소 양을 계산 알고리즘에 따라 계산한다. 또한 코드 탄소배출량과 관련된 3가지 요인에 대해 각각 몇 퍼센트씩 차지하는지 연산하여 제공한다.

2.2.3. 수치자료, 이미지 자료를 통해 결과 전달

계산 결과를 수치적으로 표시하고 수치를 그래프로 표현하여 가시성을 높인다. 탄소 배출량과 비슷 정도의 탄소 배출량을 가지는 교통수단을 이미지로 제시하여 User로 하여금 직관적으로 배출량을 느낄 수 있게 해준다.

2.3. User Characteristics

2.3.1. Java Developer 및 system manager

친환경적이어야 하는 시스템을 담당한 개발자 및 system manager.

만약 해당 시스템이 사용 하는 programming language 중 하나로 Java를 사용한다면, code implementation 및 testing, 유지 보수 단계에서 서비스의 사용이 예상된다.

주 목적은 새로이 작성하는 코드의 검토와 기존 코드를 변경할 때 변경될 코드와의 비교, testing 단계에서의 constraint 충족 여부 검토 등으로 예상된다.

2.3.2. Java 학습자

Java developer를 목표로 새로이 Java를 학습하고자 하는 junior developer.

그 중에서도 특히 보다 나은 code 스타일을 고민하며 그것을 만족할 때, 해당 code가 친환경적이 된다는 것을 인지하고 있는 학습자.

주 목적은 Java 학습 과정 중 특정 문법이나 코드 스타일이 시스템의 부하를 어느정도로 일으키는지, 실행 시간은 어떤 것이 빠른지 등을 개략적으로 파악하는 것으로 예상된다.

2.3.3. Java open source 사용 예정자

system의 조달 단계에서 open source를 사용하고자 결정했고, requirement에 hw의 부하 등 친환경적인 constraint가 존재하는 system의 manager.

요구사항에 충족하는 open source를 채택후, 해당 open source가 Java로 작성되었다면, 채택 전 점검 단계에서 서비스로의 접근이 예상된다.

주 목적은 open source가 기술된 constraint를 충족하는지 검토하는 것으로 예상되며, 이 경우 많은 양의 코드가 입력될 것으로도 예상된다.

2.4. constraint

- 모든 실행 가능한 Java 코드를 처리할 수 있어야함
- 외부 코드를 분석하는 서비스이므로, 보안에 보다 신경쓸것
- 서비스는 24시간 이용 가능해야 함
- 사용자가 너무 오랜 시간 기다리지 않게끔 해야함
- 시스템의 변경 가능 지점을 예측하여 실제 수정 요청이 발생했을 경우 비용이 적고 신속하게 이루어져야함
- 서버 운영 비용이 합리적이어야함
- 사용자가 시스템의 기능을 별다른 설명서 없이도 능숙하게 사용 가능해야 함
- 사용자가 결과창으로 하여금 본인 code의 green화 정도를 파악할 수 있어야 함
- 시스템 자체도 친환경적이어야 함

2.5. Assumptions and Dependencies

- 입력될 Java code는 컴파일 에러 및 런타임 에러가 발생하지 않을 것이다.
- user는 컴퓨터 로우 레벨에 대한 기초적인 수준의 이해가 있을 것이다.

3

Specific Requirements

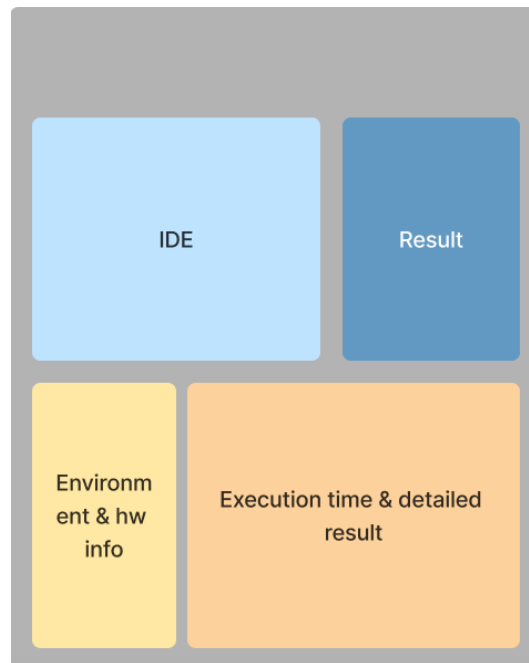
3.1. External Interface Requirements

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. 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 입출력과 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, Int형의 코드 값
명령 형식	각 코드 값에 따른 명령 매핑
종료 메시지	해당 없음

Table 3.1: User interface of input processing

목적/내용	사용자에게 제공하는 인터페이스
입력 주체/출력 목적지	클라이언트/사용자
범위/정확도/허용 오차	<ul style="list-style-type: none"> • 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 따른 화면 전환
타 입출력과 관계	사용자의 입력을 위한 인터페이스로서 출력 후 사용자의 입력 대기
화면 형식 및 구성	



- 개발할 웹사이트는 메인에 **Java code** 입력 IDE 창이 위치, 하단에 탄소 배출 그래프가 배치됨
- 웹사이트의 주 기능인 소스 코드의 탄소 배출량을 사용자는 결과값 출력으로 확인할 수 있음

윈도우 형식 및 구성	1. JAVA 코드 입력 가능한 IDE 창	2. 탄소 배출량 수치를 나타내는 그래프 통계
데이터 형식 및 구성	이미지, 텍스트	
명령 형식	해당 없음	
종료 메시지	해당 없음	

Table 3.2: 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	해당없음
단위	해당없음
시간/속도	코드의 탄소 배출량을 계산하는 데에 해당하는 처리
타 입출력과의 관계	해당 없음
화면 형식 및 구성	해당 없음
윈도우 형식 및 구성	해당없음
데이터 형식 및 구성	해당 없음
명령 형식	코드
종료 메시지	해당 없음

Table 3.4: Hardware interface of applicable device for the syst

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, Safari와 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 웹사이트출력
윈도우 형식 및 구성	해당 없음
데이터 형식 및 구성	해당 없음
명령 형식	해당 없음
종료 메시지	해당 없음

Table 3.5: Software interface of applicable device for the system

3.1.4 Communication Interface

이름	호스트 서버 – 클라이언트
목적/내용	각 클라이언트에서 호스트 서버에 접속을 요청하고, 사용자가 IDE 창에 입력한 코드를 호스트 서버에서 각 클라이언트에게 전달받고 코드의 탄소 배출량을 계산한 결과를 제공
입력 주체/출력 목적지	클라이언트와 호스트서버
범위/정확도/허용 오차	해당 없음
단위	패킷
시간/속도	최소 10Mbps 이상
타 입출력과 관계	해당 없음
데이터 형식	-Struct를 이용한 명령 코드 (탄소 배출량 계산 등) -그래프 형성을 위한 탄소 배출량 관련 수치 데이터
명령 형식	Send() 콜에 의한 통신
종료 메시지	Close()콜에 의한 소켓 종료

Table 3.6: Communication interface of applicable device for the system

3.2 Functional Requirements

3.2.1 Use Case

Use case name	Calculate
Actor	시스템에 접속하는 모든 사용자
Description	사용자가 사용자의 java 코드의 탄소 배출량을 계산하기 위해 시스템을 이용하는 과정이다.
Normal Course	<ol style="list-style-type: none"> 1. 모든 사용자는 웹사이트 접속 후 메인 화면에서 코드를 입력할 수 있는 IDE 창이 나타난다. 2. 회원가입 여부와 상관없이 자유롭게 IDE 창에 코드를 입력 후, 계산 버튼을 클릭한다. 3. 사용자는 코드 계산이 완료될 때까지 일정 시간 기다린다. 4. 사용자는 결과값을 나타내는 인터페이스를 통해, 코드의 탄소배출량 및 코드 탄소 배출의 요인을 확인할 수 있다. 5. 사용자는 계산이 완료된 코드에 추가적인 수정을 통해 코드의 탄소 배출량 및 배출의 요인을 다시 확인할 수 있다. 6. 혹은 리셋 버튼을 클릭하면 빈 IDE 창이 다시 나타나 탄소배출량을 계산할 수 있다.
Precondition	<ol style="list-style-type: none"> 1. 사용자는 JAVA 코드 외에 다른 코드를 입력해서는 안된다. 2. 사용자는 계산 버튼을 클릭 후, 계산 결과를 기다리는 동안 버튼을 다시 클릭할 수 없다. 3. 사용자는 계산에 따른 결과값을 수정하거나 지울 수 없다.
Post Condition	해당 JAVA 코드를 통해 탄소배출량을 계산한 결과값을 사용자에게 출력한다.
Assumptions	해당 없음

Table 3.7: Use Case of Calculate

Use case name	Statistics
Actor	시스템에 접속하는 모든 사용자
Description	statistics 기능은 사용자가 전반적인 탄소배출량 관련 통계를 시각적인 그래프를 이용하여 확인할 수 있는 프로세스이다
Normal Course	<ol style="list-style-type: none"> 1. 모든 사용자는 웹사이트의 하단에서 여러 통계 그래프를 확인한다. 2. 탄소배출량, 배출 요인 계산 후, 계산하는 데 사용된 서버 정보가 하단의 좌측에 위치한 그래프를 통해 표시된다. 3. 계산하는데 소요된 시간 및 사용된 전력과 hw 별 소모된 탄소배출량 세부 정보를 우측 그래프를 통해 표시된다.
Precondition	<ol style="list-style-type: none"> 1. 사용자는 한 번 이상의 탄소 배출량, 배출 요인 계산 기능을 이용해야만 위 그래프를 확인할 수 있다. 2. 각 그래프는 실시간으로 반영되는 정보가 아닌 한 번의 탄소배출량, 배출 요인 계산으로 생성된 데이터를 시각적으로 나타낸다.
Post Condition	다음 탄소배출량, 배출 요인 계산을 통해 데이터가 변경되기 전까지 상태를 유지한다.
Assumptions	해당 없음

Table 3.8: Use Case of Statistics

3.2.2. Use Case Diagram

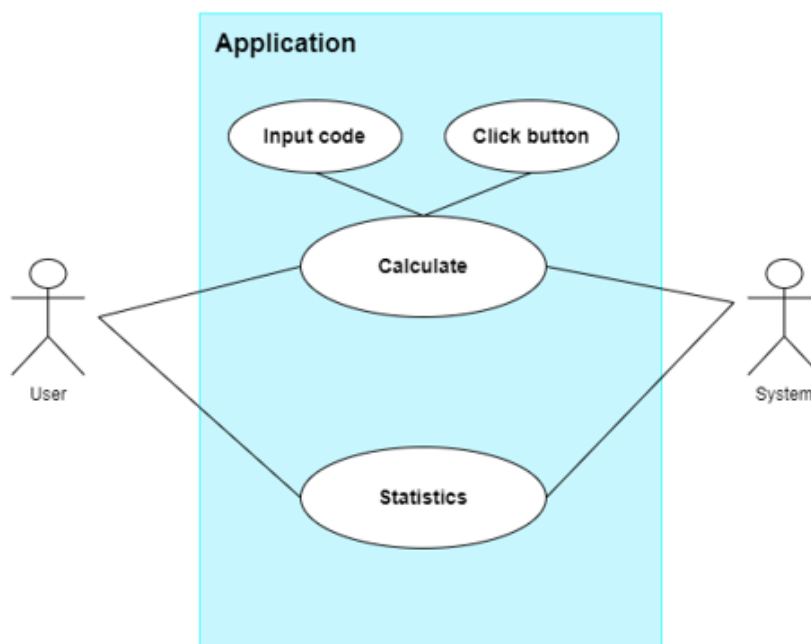


Figure 3.1: Use Case Diagram

Problem			
Field	Key	Constraint	Description
number	PK	Not Null	problem number
context		Not Null	problem context
test case path		Not Null	Test case to validate code (File path)
progress		Not Null	Answer expected (File path)

Table 3.9: Problem

3.3 Performance Requirements

성능 요구사항에 대해 설명한다. 정적 요구사항과 동적 요구사항으로 나누어 설명한다.

3.3.1 Static Requirement

- 시스템은 여러 명의 사용자가 동시에 코드를 입력할 수 있도록 하여야 하며, 주 사용자는 작성한 코드에 대한 탄소 발생량을 측정하려는 개발자이다.
- 탄소 발생량 측정을 위해 입력하는 코드는 **java**로 제한한다.
- GPU를 사용하지 않고 실행 가능한 코드를 입력해야 한다.
- 코드를 실행한 후 실행 환경을 표기해주어야 한다.

3.3.2. Dynamic Requirement

- 본 웹페이지에 최소 100명의 사용자에게 대한 접속을 원활하게 유지할 수 있어야 한다.
- **java**코드 입력시 코드에 대한 탄소 배출량을 구체적인 수치로 명시하며, 계산된 탄소 배출량은 10초 이내로 화면에 표시되어야 한다.
- 사용자는 24시간 코드 입력을 통한 탄소배출량 화면을 확인할 수 있어야한다.
- 입력한 코드의 탄소배출량을 교통수단의 탄소배출량으로 치환하여 시각적으로 교통수단 이미지와 함께 화면에 표시할 수 있어야 하며, 10초 이내로 화면에 표시되어야 한다.
- 또한 코드의 탄소배출에 영향을 미치는 요소들 중 각각 몇퍼센트씩 차지하는지 계산하여 시각화 되어야 한다.

3.4. Design Constraints

3.4.1. Physical design constraints

시스템의 목적은 이용자의 자바 코드를 통해 탄소 배출량을 계산하여 구체적인 수치로 명시를 하는 것이므로 자바 외 다른 언어로 입력을 불가하다. 시스템은 미리 정의된 그린화 패턴들을 입력 코드와 비교하여 계산할 수 있어야 한다.

3.4.2. Standards compliance

시스템은 웹 어플리케이션으로 Java Spring을 이용해 개발되며, Java Spring 코딩 표준(Java style) 및 Jakarta EE 표준을 따른다. 변수 이름은 낙타 표기법(Camel Case)를 따르며, 함수는 파스칼 케이스를 따른다.

3.5. Software System Characteristics

아래는 본 시스템의 몇 가지 비기능적 요구사항에 관련된 내용이다. 비기능적 요구사항은 제품 요구사항, 조직상의 요구사항, 외부적인 요구사항으로 나누어 기술하였다.

3.5.1. Product Requirements

제품 요구사항에서는 본 시스템의 실행 시간 중에 시스템이 어떻게 작동해야 하는지에 대해 설명한다. 본 시스템은 아래의 요구사항을 만족해야 한다.

Usability Requirements

본 시스템은 비전문가, 즉 개발자나 환경 전문가가 아닌 사람들을 대상으로 하는 환경 친화적 시스템으로 탄소 배출량 수치에 대한 이해도가 없이도 본인의 코드가 환경 파괴에 얼마나 기여를 했는지 인지할 수 있어야 한다. 탄소 배출량, 배출 요인 수치의 경우 직관적으로 느끼기엔 한계가 있으므로 해당 수치와 적절한 예시와 함께 제공할 수 있어야 한다.

Performance Requirements

본 시스템은 동시 다발적인 사용자 접속을 지원하며, 이로 인한 메모리 또는 하드웨어 리소스 사용량이 가변적 일 수 있다. 이러한 환경에서 해결하기 위해 각 유저별로 컨테이너를 제공하여 공정한 메모리 사용과 런타임을 계산할 수 있어야 한다. 또한 사용자 경험을 향상시키기 위해 입력부터 결과 표시까지 실행 시간이 5초 내로 유지되어야 한다.

Security Requirements

시스템 관리자를 제외한 일반 사용자는 시스템 관리자만큼의 권한을 가질 수 없고, 사용자는 자신과 관련된 제한된 데이터만 접근이 가능하며, 시스템 데이터베이스에 직접적으로 접근할 수 없어야 한다.

3.5.2. Organizational Requirements

조직상의 요구사항은 사용자 및 개발자가 속한 조직의 정책과 절차에 의해 발생하는 요구사항들에 관한 설명이다.

Environmental Requirements

데스크탑 또는 노트북 웹 브라우저 환경에서 접속을 권장한다.

Operational Requirements

본 시스템의 사용자들은 웹 주소를 통해서 접근이 가능하고 시스템은 사용자가 원하는 때 언제라도 자유롭게 접속하여 본인 코드의 탄소 배출량을 확인할 수 있다.

External Requirements

외부적인 요구사항에서는 시스템의 외부적 요인으로 인해서 발생하는 다양한 요구사항들에 관해서 설명한다.

Safety / Security Requirements

시스템은 사용자들의 그 어떤 개인 정보도 외부의 시스템이 접근할 수 없도록 해야 한다.

Regulatory Requirements

본 시스템은 오로지 사용자 코드의 탄소 배출량 결과 표시를 목적으로 사용자 정보를 필요로 함을 밝히고 이에 필요한 최소한의 정보만을 얻을 수 있도록 한다. 법에 따라 사용자의 개인 정보를 침해해서는 안 되고, 시스템은 국제 개인 정보 보호 표준에 따라 개발되어야 한다.

3.6. Organizing the Specific Requirements

이 부분에서는 Unified Modeling Language(UML)과 표 기반 그래픽 표기로 시스템 모델을 다룬다. 이 모델은 전체 시스템과 서브 시스템 간의 관계를 보여준다.

3.6.1. Context Model

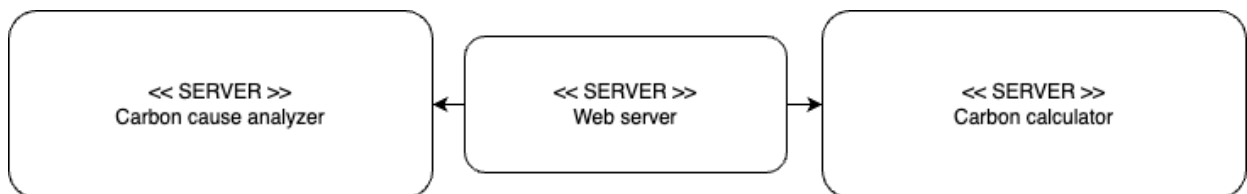


Figure 3.2: Context Model

3.6.2. Process Model

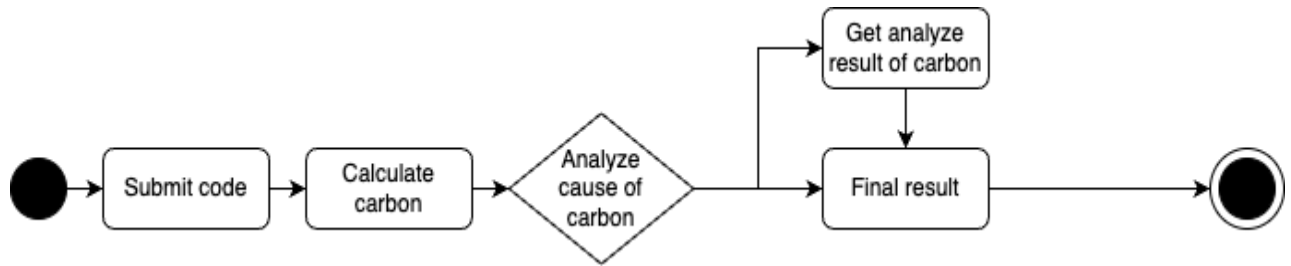


Figure 3.3: Overall Process Model

3.6.3. Interaction Model

Use Case Diagram 참조

3.7. System Architecture

이 부분에서는 개발될 시스템 아키텍처의 전반적인 개요를 소개한다. 각 서브 시스템과 그 구성요소를 파악하며, 서브 시스템 간 어떠한 방식으로 통신하는지도 살펴본다.

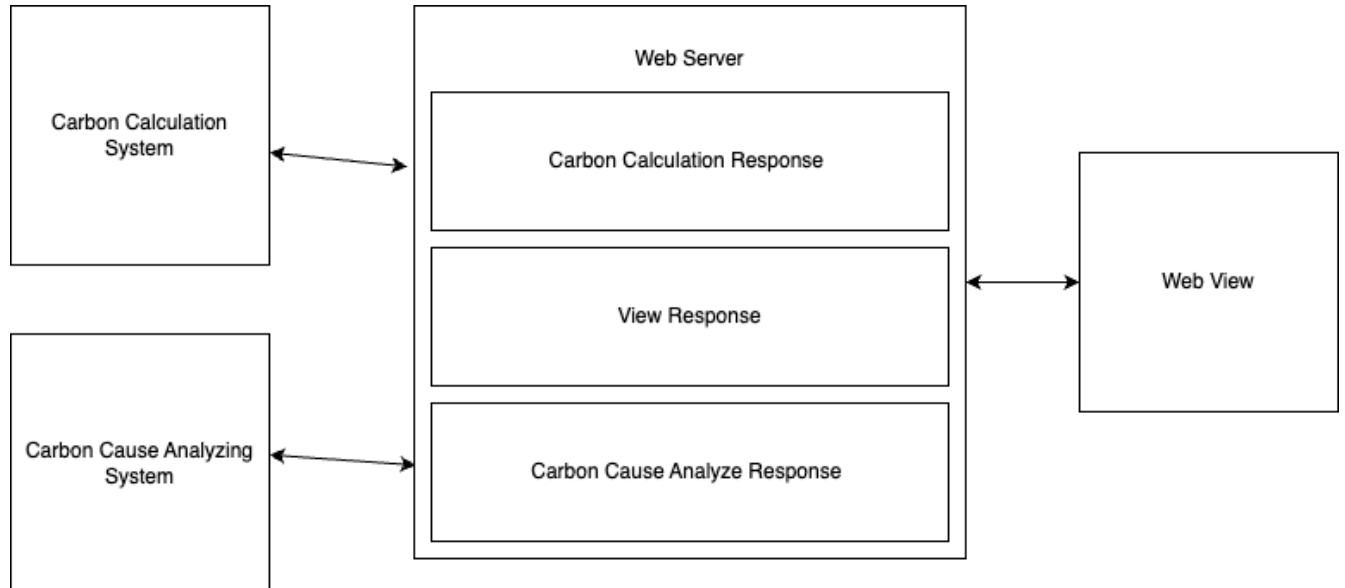


Figure 3.4: System Architecture

3.8. System Evolution

이 부분에서는 시스템을 구축하는 데 필요한 기본적인 가정, 하드웨어의 변화, 그리고 사용자의 요구 사항 변화 등 예상 가능한 모든 변동 사항을 다룬다. **System Evolution**은 시스템의 미래 변화에 대비해 문제가 될 수 있는 설계 결정을 회피하는 데 도움을 줄 것이다. 이는 매우 중요하며, 효과적으로 활용될 것이다.

3.8.1. Limitation and Assumption

코드의 복잡성에 따라 정확한 탄소 배출량을 예측하는 것은 어려울 수 있다. 모든 자바 코드는 동일한 환경에서 실행되지 않으므로, 다양한 환경에서의 탄소 배출량을 측정하는 것은 제한적일 수 있다. 현재 사용되는 프로그래밍 패턴, 라이브러리, 또는 최적화 방식에 따라 탄소 배출량이 달라질 수 있다. 그러므로 해당 시스템의 분석 결과는 참고용으로만 사용해야 한다. 사용자가 제공하는 코드는 문법적으로 올바르다고 가정한다. 또한 코드의 실행 환경에 대한 기본적인 정보나 세팅이 제공될 것이라고 가정한다. 또한 시스템은 주로 서버나 클라우드 기반 환경에서의 탄소 배출을 중점적으로 분석한다고 가정한다.

3.8.2. Evolution of Hardware and Change of User Requirements

Evolution of Hardware:

본 프로젝트는 자바 코드의 탄소 배출량과 원인을 분석하는 서비스를 중심으로 한다. 그러나 외부 요인에 의해 SW 프레임워크나 HW가 변화할 가능성은 현저하게 작다. 본 서비스의 초점은 자바 코드의 탄소 배출량 분석이지만, 사용자들의 요구에 따라 다른 프로그래밍 언어나 환경에 대한 분석 기능이 요청될 수 있다. 사용자들은 점점 더 상세한 탄소 배출 분석을 원할 수 있다. 이에 따라 함수별, 라이브러리별, 모듈별 분석과 같은 더 구체적인 기능에 대한 요구가 생길 것이다. 사용성과 학습효율에 관한 피드백이 주로 받아들여질 것으로 예상되며, 이를 통해 인터페이스의 개선이나 튜토리얼 제공 등의 방향으로 서비스가 발전해 갈 것이다.

4

Supporting Information

4.1. Software Requirements Specification

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

4.2. Document History

Document History			
Date	Version	Description	Writer
2023-10-27	V1.00	1, 4	김민수
2023-10-27	V1.00	2.1 - 2.2	유규환
2023-10-27	V1.00	2.3 - 2.5	이시혁
2023-10-27	V1.00	3.1 - 3.2	임소현
2023-10-27	V1.00	3.3 - 3.4	장영우
2023-10-27	V1.00	3.5 - 3.6	정단호
2023-10-27	V1.00	3.7 - 3.9	차승일