

Software Requirement Specification

Eco Master

by

임성훈, 황인성, 양승환, 김민성, 문수현

TEAM 8

Instructor: 이은석

Document Date: 2024/05/05

Faculty: Sungkyunkwan University

Contents

1 Introduction	
1.1 Purpose	4
1.2 Scope	4
1.3 Definitions, Acronyms, and Abbreviation	5
1.4 References	5
1.5 Overview	5
2 Overall Description	
2.1 Product Perspective	
2.1.1 System Interfaces	7
2.1.2 User Interfaces	7
2.1.3 Hardware Interfaces	7
2.1.4 Software Interfaces	7
2.1.5 Memory Constraints	8
2.1.6 Operations	8
2.2 Product Functions	
2.2.1 사용자의 코드 입력	8
2.2.2 데이터 송수신	8
2.2.3 전송된 코드 실행	9
2.2.4 탄소배출량 계산	9
2.2.5 Green Pattern 을 적용할 부분 탐색 및 수정	9
2.2.6 탄소배출량 데이터 분석	9
2.2.7 사용자에게 분석 결과 전송	9
2.2.8 Green Quiz	9
2.3 User Characteristics	
2.3.1 개발자, 학생	10
2.3.2 기업	10
2.4 Constraints	10
2.5 Assumptions and Dependencies	10
3 Specific Requirements	
3.1 External Interface Requirements	11
3.1.1. User Interface	11
3.1.1.1 Input processing via mouse and keyboard	11
3.1.1.2 Input processing via monitor	12
3.1.2. Hardware Interface	16
3.1.3. Software Interface	17
3.2 Functional Requirements	18
3.2.1 Use Case	18
3.2.2 Use Case Diagram	20
3.2.3 Data Dictionary	21
3.2.4 Data Flow Diagram	22
3.3 Performance Requirements	22
3.3.1 Static Numerical Requirement	22
3.3.2 Dynamic Numerical Requirement	22

3.4	Logical Database Requirements	23
3.5	Design Constraints	23
3.5.1	Physical design constraints	23
3.5.2	Standards compliance	23
3.6	Software System Characteristics	23
3.6.1	Product Requirements	23
3.6.2	Organizational Requirements	24
3.6.3	External Requirements	24
3.7	Organizing the specific requirements	25
3.7.1	System mode	25
3.7.2	User class	25
3.7.3	Objects	25
3.7.4	Feature	26
3.7.5	Stimulus	26
3.7.6	Response	26
3.7.7	Functional hierarchy	27
3.7.8	Additional comments	27
3.8	System Evolution	27
3.8.1	Limitation and Assumption	27
3.8.2	Evolution of Hardware and Change of User Requirements	28
4	Supporting Information	
4.1	Software Requirement Specification	29
4.2	Document History	29

Introduction

1.1 Purpose

이 문서는 개발자들을 대상으로 자신이 작성한 코드의 탄소 배출량을 가시화하여 경각심을 가지도록 하고, 이를 개선하는 코드를 제공하여 결과적으로 환경에 부담이 적은 코드를 제공하는 프로그램의 **requirements** 를 정확히 하고, 유지보수하는 과정에서 참고할 수 있도록 제작되었다.

1.2 Scope

EcoMaster 는 사용자의 코드를 입력받아 문제가 되는 부분을 제시하고, 이를 개선하여 친환경적이고 효율적인 코드를 제공하는 프로그램이다. 사용자가 제공한 코드에 적용할 **Green Pattern** 이외에도 퀴즈를 통해 다양한 **Green Pattern** 을 사용자에게 제공하여 효율적인 코드를 작성할 수 있도록 한다.

1.3 Definitions, Acronyms, and Abbreviation

탄소 배출량	코드를 컴파일하고 실행하는 모든 과정에서 컴퓨터가 사용하는 에너지를 생성하기 위해 발생하는 탄소의 양
Green Pattern	같은 결과의 코드지만 탄소 배출량을 적게 만드는 코드의 Pattern
UI	User Interface 의 약자로, 사용자가 사용하는 웹페이지에서 마주하는 디자인적 요소와, 사용자 입력에 의해 발생하는 동작 등을 포함
UX	User Experience 의 약자로, 사용자가 제품이나 서비스에 대해 전체적으로 느끼는 사용자의 경험을 의미함.
프로그래밍 언어	실제로 프로그램을 작성할 때 사용되는 언어로, 기계어와, 어셈블리어에 의해 번역되는 어셈블리 언어, 컴파일러에 의해 번역되는 컴파일러 언어, 인터프리터에 의해 번역되는 베이식 등이 있음.
고급 언어	프로그래밍 언어 중 프로그래머가 특정 형식의 컴퓨터와는 무관하게, 독립적으로 프로그램을 작성할 수 있는 언어로, 기계어보다 인간의 언어에 더 가깝고, 보통 컴파일러나 인터프리터가 해석하여 기계어로 바꾸어 실행하게 됨.
JAVA	자바는 1996 년 1 월에 배포된 객체지향 프로그래밍 (Object Oriented Programming, OOP) 언어로, 운영체제에 독립적으로 실행이 가능하기 때문에 빠른 시간 내에 많은 사용자를 확보하여 현재까지도 전 세계에서 많이 쓰이는 언어 중 하나이며, 다양한 운영체제 환경에서 사용할 수 있는 웹 개발에 적합한 언어로 많이 활용됨..
LLM	Large Language Model 의 약자로, 인간의 텍스트 문자를 이해할 수 있는 인공지능 모델이다.

1.4 References

EE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications, In IEEEExplore Digital Library

<http://ieeexplore.ieee.org/Xplore/guesthome.jsp>

2022 Fall 41 class team1

https://github.com/skkuse/2022fall_41class_team1

1.5. Overview

본 소프트웨어 요구사항 명세서는 네 개의 주요 챕터로 구성된다. 첫 번째 챕터에서는 본 프로젝트의 결과물인 'Ecomaster'의 배경 및 중요성에 대해 설명하며, 해당 문서에서 사용되는 전문 용어 및 약어에 대한 정의를 제공한다. 두 번째 챕터에서는 사용자의 입력을 처리하는 시스템의 전반적인 인터페이스와 기능, 디자인 및 구현 제약사항 등 시스템 전반에 대한 개요를 자연어 형식으로 간략하게 밝힌다. 세 번째 챕터에서는 인터페이스, 기능, 성능 등에 대한

요구사항을 구체적으로 제시한다. 마지막 챕터에서는 본 요구사항 명세서를 작성하기 위해
참고한 기준이 되는 문서를 밝힌다..

2

Overall Description

2.1 Product Perspective

소프트웨어 산업의 발전과 인공지능 모델, 특히 LLM의 발전에 따라 소프트웨어 개발과 정보 탐색이 매우 용이해졌다. 그러나 이에 따라 무분별하게 코드를 생성하고, 비효율적이고 과도한 구조를 가진 코드들이 많아지면서 이를 컴파일하고 실행하는데에 막대한 에너지를 사용하고 있다. Ecomaster는 이러한 문제를 해결하기 위해 사용자에게 같은 기능을 가지지만 더 효율적으로 작동하는 코드를 제공하고, 이를 작성하는데에 필요한 **Green Pattern**에 대한 지식을 제시한다. 이에 따라 더 친환경적이고 효율적인 코드를 작성하도록 장려한다.

2.1.1 System Interfaces

사용자들은 **JAVA**를 사용해 작성한 코드를 프로그램에 입력하고, 프로그램은 제공받은 코드를 서버에 전달하고, 서버는 탄소배출량을 측정하고 개선할 부분을 표시하여 개선한 코드와 함께 프로그램에게 제공한다. 프로그램은 서버로부터 전달받은 코드들과 탄소배출량을 사용자가 볼 수 있도록 웹페이지에 출력한다. 사용자가 제공한 코드에서 개선할 부분이 존재한다면 개선할 부분을 표시하여 사용자에게 제공하고 이상적으로 개선된 코드도 함께 제공한다.

2.1.2 User Interfaces

인터페이스는 웹페이지를 통해 제공한다. 사용자는 Ecomaster라는 웹페이지에 접속하여 좌측 코드입력창에 코드를 입력하여 웹페이지의 이용을 시작할 수 있다. 코드를 입력 후 웹페이지에서 계산된 탄소배출량과 개선할 부분, 개선한 코드를 표시하면 코드를 복사할 수 있고, **Green Quiz**에 응시할 수 있다.

2.1.3 Hardware Interfaces

본 시스템은 웹 어플리케이션으로, 컴퓨터를 통해 사용할 수 있다. 권장되는 사양은 1.0GHz 이상의 싱글 프로세서, 4GB 이상의 RAM을 가진 데스크탑 및 노트북이다.

2.1.4 Software Interfaces

본 시스템은 Window10 및 Window11 환경을 기준으로 개발되었으나 이전 환경에서도 사용이 가능하다. 또한 Chrome 버전 100.0, Microsoft Edge 버전 100.0, Firefox 버전 99 이상의 웹 브라우저를 사용함을 권장한다.

2.1.5 Memory Constraints

본 시스템은 적어도 4GB의 RAM을 가진 컴퓨터에서 접속하는 것을 권장한다.

2.1.6 Operations

System Administrator

탄소배출량 계산: 코드의 탄소배출량을 계산한다. 입력된 Java 코드의 탄소배출량을 사용자가 확인할 수 있도록 한다.

개선 코드 제공: 코드의 비효율적인 부분을 개선한 코드를 사용자에게 제공한다.

2.2 Product Functions

2.2.1 사용자의 코드 입력

사용자는 입력하는 Java 코드의 탄소 발생량과 전력 사용량을 확인하기 위해 웹사이트에 자신의 Java 코드를 입력한다. 사용자가 처음 'EcoMaster'에 접속하면 메인 페이지의 가장 위쪽에는 Ecomaster의 로고가 있고, 그 아래에 Java 코드를 입력할 수 있는 칸이 있다. 이 칸은 가로와 세로로 길게 설정되어 있어 사용자는 자신의 코드를 편하게 한 눈에 볼 수 있다. 사용자는 지정된 칸에 자신의 Java 코드를 입력하고, 컴파일 버튼을 통해 코드의 탄소 발생량, 전력 사용량을 계산하기 위한 다음 단계로 나아간다. 사용자의 Java 코드 입력을 위한 입력란은 하나 이상의 탭으로 구성되어 있다. 이 탭은 코드 입력란 위에 있는 버튼을 사용하여 추가·삭제·전환할 수 있으며, 여러 탭을 생성하여 다중 클래스가 필요한 코드의 경우에도 별도의 추가작업 없이 각 클래스의 코드를 입력할 수 있다.

2.2.2 데이터 송수신

사용자가 입력란에 Java 코드를 입력하면, 웹사이트는 서버에 Java 코드에 대한 정보를 전송한다. 이후 서버는 사용자에게 계산의 결과값을 반환한다. 사용자가 수신하는 정보에는 Java 코드와 코드가 실행되는 서버 정보와 탄소배출량 및 데이터 분석 결과가 포함되어 있고, 코드가 비효율적인 원인이 되는 부분을 표시하고, 이를 개선한 코드가 포함되어 있다. 만약 여러 개의 탭을 이용해 코드를 입력했을 경우, 웹사이트는 각 탭의 코드를 종합하여 한 번에 전송한다.

2.2.3 전송된 코드 실행

서버는 웹사이트로부터 전송받은 코드를 실행한다. 이 때 두 가지 경우를 고려해야 한다. 1) 코드가 오류 없이 실행 가능할 경우 서버는 코드를 실행하여 코드를 실행 과정에서 소요된 런타임을 측정하며 측정된 런타임을 서버의 하드웨어 정보와 결합하여 코드의 탄소배출량을 계산하는 데에 사용한다. 2) 코드에 오류가 존재하여 실행 불가능할 경우 서버는 사용자에게 오류가 존재하여 코드를 실행할 수 없음을 알리는 팝업창을 띄우고, 사용자는 오류를 수정하여 다시 입력란에 코드를 제출해야 한다.

2.2.4 탄소배출량 계산

서버는 웹 사이트로부터 전송받은 Java 코드가 발생시키는 탄소배출량을 계산한다. 탄소배출량의 계산 과정에서는 컴파일 시간 및 서버 정보를 활용한다.

2.2.5 Green Pattern 을 적용할 부분 탐색 및 수정

서버는 Java 코드를 분석하여 보유한 Green Pattern 으로 코드의 탄소배출량을 절감할 수 있는 부분을 탐색하고, 존재한다면 표시해두고 이를 Green Pattern 을 사용해 개선한 코드를 생성한다. 개선한 코드의 탄소배출량도 계산한다.

2.2.6 탄소배출량 데이터 분석

서버는 계산된 탄소배출량을 사용자가 더 직관적으로 받아들일 수 있도록 코드의 탄소배출량이나 전력 사용량을 실생활과 연관지어 분석한다. 예를 들어, 현재 입력된 Java 코드가 발생시키는 탄소배출량과 같은 양의 탄소를 배출하며 가솔린 자동차가 이동할 수 있는 거리로 환산한다. 전력사용량의 경우 전기자동차가 같은 양의 전기를 사용하며 이동할 수 있는 거리로 환산할 수 있다. 같은 방식으로 휴대폰 충전 횟수, 에어컨 가동 시간 등으로도 환산할 수 있다.

2.2.7 사용자에게 분석 결과 전송

서버는 탄소배출량 계산 결과, 데이터 분석 결과와 서버 정보를 다시 사용자에게 전송하고, 사용자는 이를 직접 확인하며 자신의 코드가 환경에 어떤 영향을 미치는지 파악할 수 있다. 웹사이트의 결과창에는 탄소배출량과 하드웨어 정보, 데이터 분석 결과가 수치와 도표 형태로 표현되고, 개선전 코드와 개선된 코드가 나란히 표현된다. 이를 통해 사용자가 자신의 코드와 개선된 코드를 쉽게 비교할 수 있다.

2.2.8 Green Quiz

사용자는 웹페이지가 제공하는 두 코드 중 더 효율적인 코드라고 생각하는 코드를 선택할 수 있다. 이를 통해 자신의 코드에 존재하지 않는 다른 Green Pattern 에 대한 정보를 함께 얻어갈 수 있다.

2.3 User Characteristics

2.3.1 개발자, 학생

현재 많은 에너지가 IT 분야에 사용되고 있으며, 최적화되지 않은 프로그램들과 무분별한 코드의 재생산으로 많은 에너지가 낭비되고 있다. 개발자들이 온라인에 깃허브에 게시하는 비효율적인 코드를 LLM 이 학습하는데 사용하면 결국 비효율적인 코드를 재생산하는데에 기여하게 되는 것이다. 개발자들은 본 웹 사이트를 통해 어떤 형태의 코드가 탄소를 많이 배출하는지, 어떤 형태의 코드가 환경친화적이고 효율적인지 습득할 수 있다. 시각화된 자료들을 통해 친환경적인 코딩의 중요성을 알게 될 것이다.

2.3.2 기업

현재 기업들은 대부분의 경영과정을 소프트웨어로 지탱하고 있다. 대규모의 데이터가 여러번 송수신되는 상황에서 이를 처리하는데에 많은 에너지를 사용하고 있고, 이에 따른 비용도 많이 소비하고 있다. 이런데에 사용하는 비용을 최소화해야 하는 기업 입장에서 결과창에 나타난 탄소배출량, 에너지 사용량을 개선한 코드를 통해 어떤 방식으로 코드를 변경해야 에너지 사용에 따른 비용을 절감할 수 있는지 알게 될 것이다.

2.4 Constraints

이 시스템은 본 문서에 기재된 요구 사항과 제약사항을 바탕으로 개발된다. 본 문서에 명시되어 있지 않은 요구사항과 제약사항은 개발자의 재량에 의해 조정할 수 있다. 아래의 사항들은 프로그램을 구현할 때의 준수 사항이다.

- 사용자가 이해하고 사용하기 쉬운 방향으로 시스템을 설계한다.
- 가능한 공개된 오픈소스를 사용한다.
- 사용자가 충분한 양의 코드를 입력하여 사용할 수 있도록 한다.
- 오류가 발생할 경우 빠르게 수정할 수 있도록 설계한다.
- 사용자가 **Green Pattern**에 대해 직관적이고 확실하게 알 수 있도록 설계한다.
- 사용자가 해당 프로그램 이용을 위해 다른 프로그램을 설치할 필요가 없도록 설계한다.

2.5 Assumptions and Dependencies

본 프로그램은 다음과 같은 환경을 가정하고 개발하므로, 이와 다른 환경에서 이 프로그램을 사용하려면 원하는 결과가 도출되지 않을 가능성이 있다.

- 1.0GHz 이상의 프로세서
- 4GB 이상의 RAM
- Windows, MacOS, Linux 환경
- 최신 버전의 Chrome, Microsoft Edge, Firefox 등 웹 브라우저
- 인터넷 연결

3

Specific Requirements

3.1 . External Interface Requirements

3.1.1. User Interface

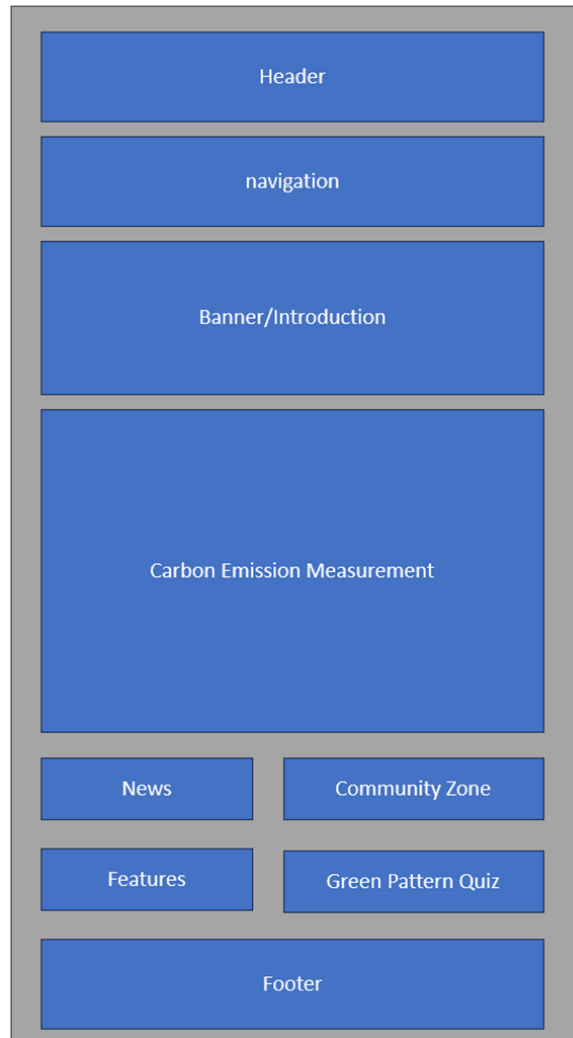
3.1.1.1 Input processing via mouse and keyboard

항목 이름	마우스 및 키보드를 통한 입력 처리
목적	사용자가 소스 코드의 탄소 배출량을 측정하고 결과를 확인할 수 있는 인터페이스 제공
입력 소스 또는 출력 대상	키보드, 마우스
유효 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
측정 단위	키 입력 수, 마우스 이동 거리 등
타이밍	실시간
다른 입력/출력과 관계	다른 입력/출력과 관계 없이 독립적으로 작동해야 함
화면 형식 및 구성	해당 없음
윈도우 형식 및 구성	해당 없음
데이터 형식 및 구성	입력된 데이터는 Java code 또는 숫자 형식으로 표시
명령 형식	사용자의 명령을 인식하고 실행할 수 있어야 함
종료 메시지	해당 없음

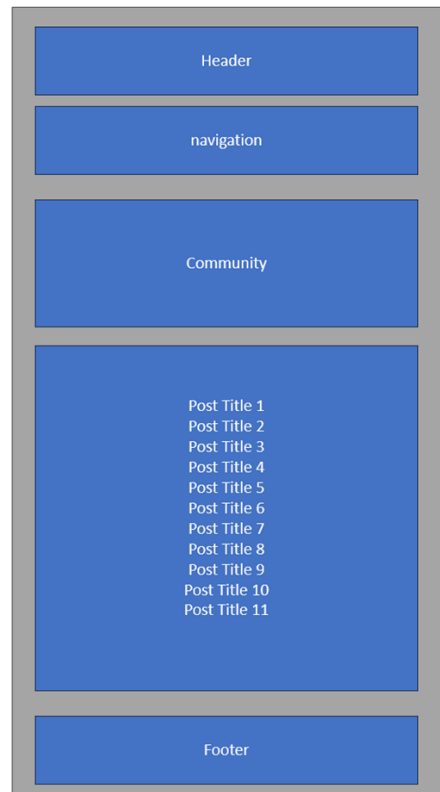
3.1.1.2 Input processing via monitor

항목 이름	모니터를 통한 메인 화면 출력
목적	사용자가 소스 코드의 탄소 배출량을 측정하고 결과를 확인할 수 있는 인터페이스 제공
입력 소스 또는 출력 대상	클라이언트/사용자
유효 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
측정 단위	화면에 표시되는 정보의 크기 및 해상도
타이밍	실시간
다른 입력/출력과의 관계	다른 입력/출력과의 간섭 없이 모니터에 정보를 표시해야 함

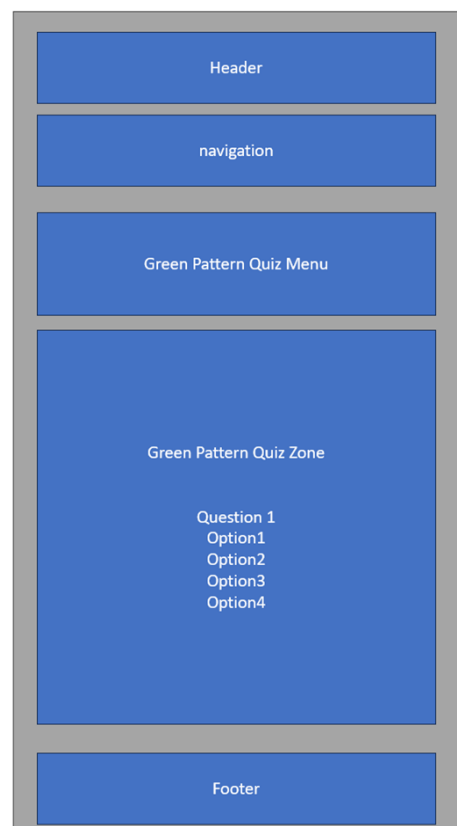
Main website



Community website



Green pattern Quiz Website



	<ul style="list-style-type: none"> • Main website 는 header, navigation, banner/introduction, carbon emission measurement, news, features, footer 로 구성되어 있음 • Main website 하단부분과 navigation 안에는 community zone 과 green pattern quiz 사이트로 가는 링크가 연결되어 있음 • Community zone 에는 header, navigation, community, 게시글, footer 로 구성되어 있음 • Green Pattern Quiz zone 에는 header, navigation, menu, quiz zone, footer 로 구성되어 있음 • 사용자는 주 3 가지(carbon emission measurement, community, quiz) 프로그램을 자유롭게 선택해서 사용할 수 있음 • Introduction 을 통해 사용자에게 웹사이트 사용방법과 위치를 정확하게 알려드림
윈도우 형식 및 구성	<ul style="list-style-type: none"> • Flexbox layout 형식으로 웹사이트 구성 • Flexbox layout 형식으로 community 와 quiz 구성 • Java 코드를 넣는 IDE 창
데이터 형식 및 구성	화면에 표시되는 정보는 텍스트, 그래픽 또는 숫자 형식으로 제공
명령 형식	해당 없음
종료 메시지	해당 없음

3.1.2. Hardware Interface

항목 이름	시스템에서 사용 가능한 디바이스
목적	키보드, 마우스를 사용한 사용자의 입력
입력 소스 또는 출력 대상	사용자/서버
유효 범위/정확도/허용 오차	해당 없음
측정 단위	해당 없음
타이밍	사용자의 입력/문제 풀이에 해당하는 처리
다른 입력/출력과 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

3.1.3. Software Interface

항목 이름	웹 사이트
목적	화면 출력
입력 소스 또는 출력 대상	해당 없음
유효 범위/정확도/허용 오차	Chrome, Edge, Firefox, Safari 와 같은 웹 브라우저에서 사용 가능
측정 단위	해당 없음
타이밍	즉각적인 반응
다른 입력/출력과의 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

3.2. Functional Requirements

3.2.1. Use Case

- 로그인

Actor(s)	모든 사용자
Description	커뮤니티 기능을 사용하기 위해 로그인 하는 과정이다.
Precondition	아이디, 비밀번호의 형식이 올바르게 기입되어야 한다.
Post Conditions	ID, password 를 올바르게 입력하면 커뮤니티 글 수정, 삭제가 가능하다.
Normal Course	<p>커뮤니티 페이지에 글, 댓글 작성 시 로그인이 완료된 상태에서만 접근할 수 있다.</p> <p>사용자는 글 작성 페이지로 이동하고, 형식에 맞는 회원 정보를 입력한다.</p> <ol style="list-style-type: none">1. ID2. Password

- Code Analyze

Actor(s)	모든 사용자
Description	사용자의 코드를 입력하고 탄소배출량을 계산하며, 변경된 코드를 제공하는 과정이다.
Precondition	<p>입력 코드 글자 수 제한</p> <p>입력 코드는 Java 코드로 제한된다.</p>
Post Conditions	코드를 분석한 후 탄소배출량 계산 결과와 변경된 코드를 출력한다.
Normal Course	<p>웹사이트 접속 시 코드를 입력할 수 있는 텍스트 필드가 나타난다.</p> <p>사용자가 코드를 입력하고, analyze 버튼을 누르면 결과를 확인할 수 있다.</p> <ul style="list-style-type: none">- 기존 코드의 탄소 배출량을 수치로 출력한다.- 변경된 코드를 출력하고, copy 버튼으로 코드를 복사할 수 있다.- 코드 변경 후 감소한 탄소 배출량 수치를 출력한다.- 탄소 배출 감소량을 탄소를 배출하는 다른 활동의 몇 배인지 계산해 이 수치를 그림과 함께 출력한다.- 사용자는 결과를 jpg 형식으로 다운로드 받을 수 있다.

- Community

Actor(s)	모든 사용자
Description	웹사이트 운영자와 사용자가 자유롭게 글을 포스트하고, 작성된 글에 댓글을 작성할 수 있다.
Precondition	글 작성 및 수정 시 사용자가 로그인 상태여야 한다.
Post Conditions	사용자가 등록한 글과 댓글을 ID, password 와 데이터베이스에 저장한다.
Normal Course	<p>웹사이트 상단에 커뮤니티 / 퀴즈 탭이 나타난다.</p> <p>사용자는 커뮤니티 버튼을 눌러 커뮤니티 페이지로 이동할 수 있다.</p> <p>커뮤니티 페이지에서는 글 목록과 작성 버튼, 검색 버튼이 나타난다.</p> <p>글 목록은 '사용자 ID / 글 제목 / 등록 날짜' 형식으로 출력된다.</p> <p>사용자는 운영자가 작성한 공지사항, 이벤트, 새로운 green pattern 정보 등의 글을 확인할 수 있다.</p> <p>사용자는 작성 버튼을 누르면 글을 입력할 수 있는 페이지로 이동하며, ID 와 password 를 설정하고, 텍스트 필드에 자유롭게 글을 작성할 수 있고 저장 버튼으로 글을 포스트 할 수 있다.</p>

- Green Quiz

Actor(s)	모든 사용자
Description	Green pattern 에 관한 퀴즈를 통해 사용자가 green code 작성에 필요한 정보를 습득할 수 있다.
Precondition	해당 없음
Post Conditions	퀴즈 결과를 출력한다.

Normal Course

웹사이트 상단에 커뮤니티 / 퀴즈 탭이 나타난다.

로그인 상태의 사용자는 퀴즈 버튼을 눌러 **green quiz** 페이지로 이동할 수 있다.

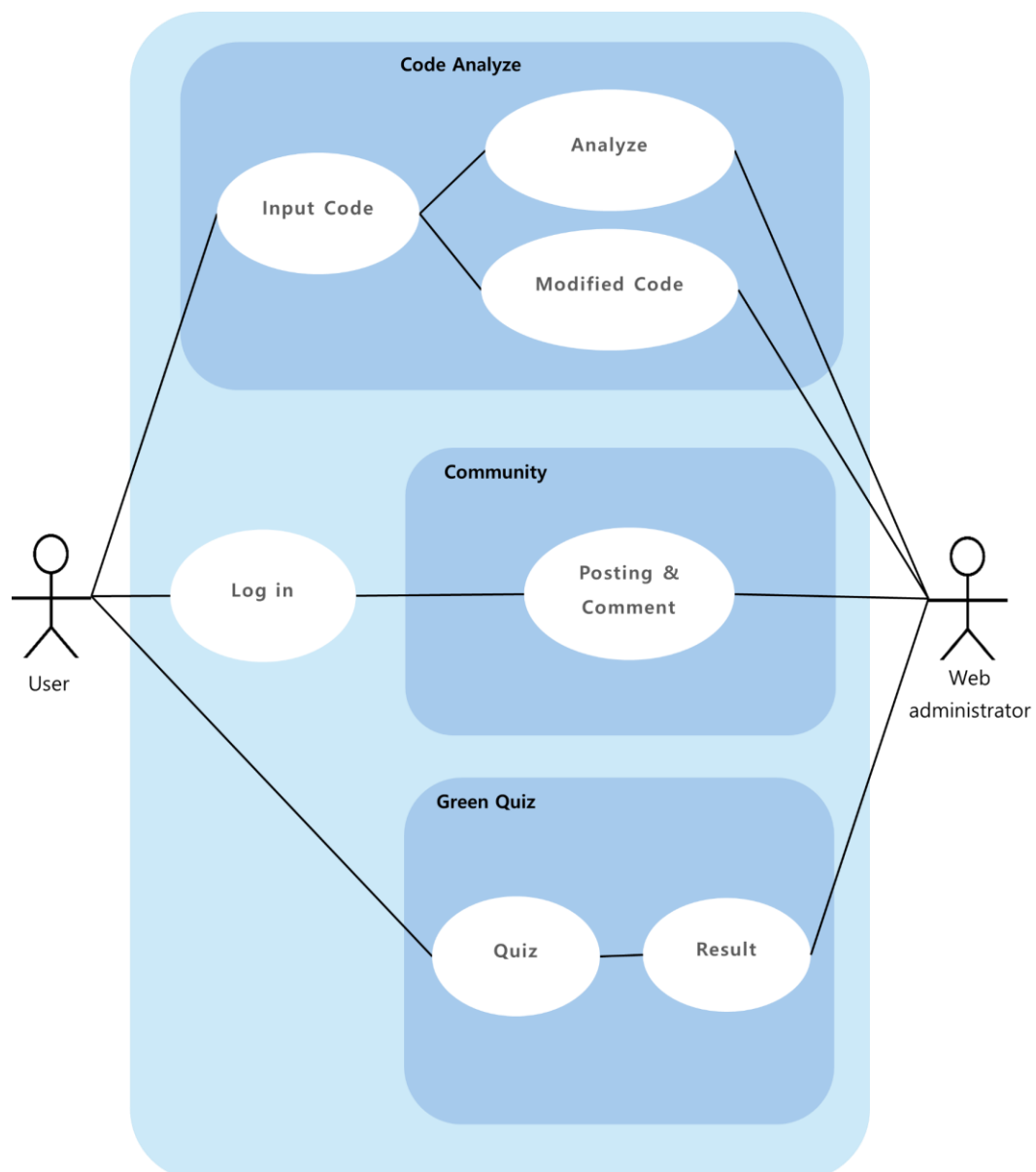
시작 버튼을 눌러 퀴즈를 시작한다.

문제는 객관식으로 버튼으로 선택해 답을 입력할 수 있고, 한 문제를 완료하면 다음 버튼을 눌러 다음 문제로 넘어갈 수 있다.

퀴즈를 마지막까지 완료하면 완료 버튼으로 퀴즈를 끝내고, 결과가 점수 형태로 출력된다.

다른 사용자들의 퀴즈 점수를 순위를 매겨 'ID / 점수' 형태의 목록으로 출력한다.

3.2.2. Use Case Diagram



3.2.3. Data Dictionary

- Code Analyze

Field	Description
usercode	User input code
modified	Modified code
usercodeFootprint	User input code's carbon footprint
modifiedFootprint	Modified code's carbon footprint

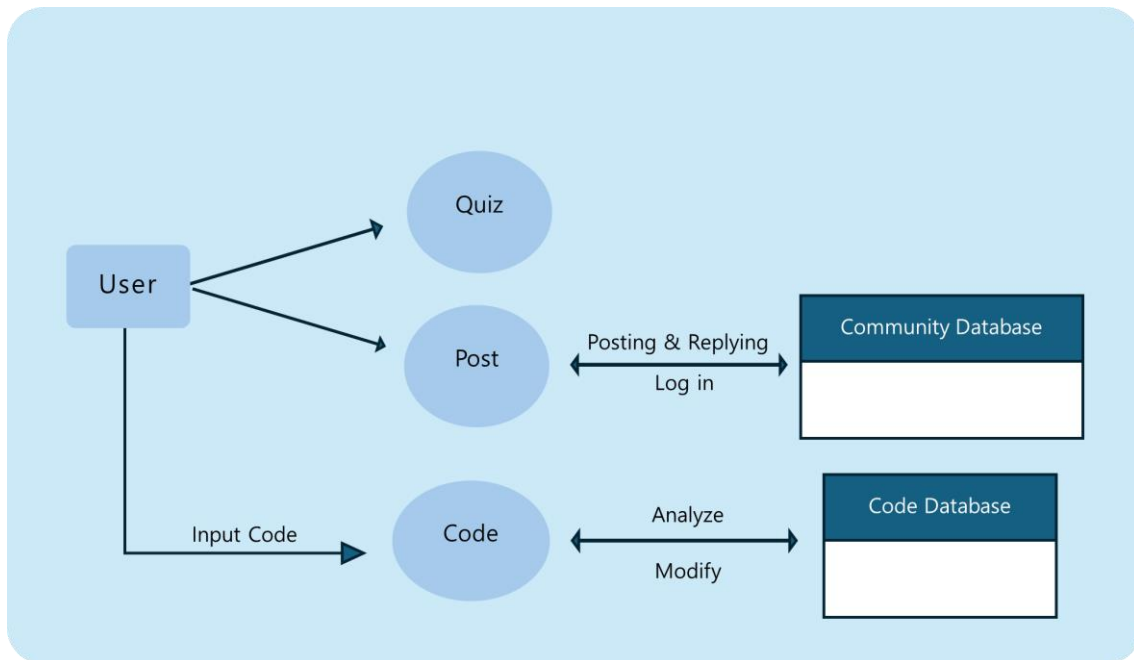
- Community

Field	Description
id	User id
password	User password
posting	User's post
time	User posting timestamp
reply	User posting's reply

- Green Quiz

Field	Description
id	User id
ans1~	User's answer of the green quiz
score	User's result of the green quiz

3.2.4. Data Flow Diagram



3.3. Performance Requirements

시스템이 특정 부하에서 반응 속도와 안정성에 관련된 최소 기준을 정의하는 부분이다. 이를 통해 소프트웨어가 효율적으로 운영될 수 있도록 한다.

3.3.1. Static Numerical Requirement

- 시스템은 여러 사용자가 접속하여 사용하고 있더라도 동시에 자신의 코드에 대한 탄소배출량과 최적화 코드를 제공받을 수 있도록 지원해야 한다. 이때 사용자는 프로그래밍에 대한 지식을 가지고 있어야 한다.
- 시스템은 최소 1.5GHz 이상의 싱글 프로세서, 2GB 이상의 메모리가 보장되는 환경의 PC에서 부드럽게 동작해야 한다.
- 1Mbps 이상의 인터넷 연결 속도가 보장되는 환경이 요구된다. 운영체제는 윈도우 7 이상, 웹 어플리케이션이 지원되어야 한다.

3.3.2. Dynamic Numerical Requirement

- 시스템은 최소 동시 접속자를 100명 이상으로 유지하도록 하여, 100명 이상의 사용자가 동시 접속하여 사용하고 있어도 끊임없이 동작하는 것을 목표로 한다. 또한 최소 10000개 이상의 게시물과 그에 대한 정보(게시물 내용, 비밀번호, 작성 시간), 그리고 1GB 이상의 그린 코드 정보를 저장할 수 있어야 한다.
- 사용자가 입력한 코드의 탄소배출량을 분석하고 최적화를 진행하는데 소요되는 시간이 2분 이내로 완료되어야 한다. 이후 계산된 결과를 출력하는데 10초 이내에 완료해야 한다.

- 사용자가 다른 항목으로 이동할 때 소요되는 시간이 5 초 이내가 되어야 한다. 그리고 사용자가 그린 코드에 대한 정보 제보 및 수정을 요청할 때에도 5 초 이내에 완료되어야 한다.
- 사용자가 퀴즈를 푸는 동안 문제 출력, 선택지 제출 후 검증 등 계산이 소요되는 모든 과정이 5 초 이내에 완료되어야 한다.

3.4. Logical Database Requirements

시스템은 Flask 데이터베이스 서비스를 이용하여 데이터를 관리한다. 해당 데이터베이스를 통하여 시스템 사용자들이 작성한 게시물 정보와 그린 코드 정보를 저장한다. 사용자가 원한다면 자신이 입력했던 코드에 대한 정보를 개인적으로 저장할 수 있도록 한다.

3.5. Design Constraints

3.5.1. Physical design constraints

이번 시스템의 목적은 사용자의 그린 코드 교육 및 사용 유도이다. 따라서 PC(데스크탑, 노트북 등) 사용을 적극 권장하며 그 외의 기기는 사용에 불편함이 있을 수 있습니다. 시스템은 Flask 데이터베이스를 이용해, 필요 데이터를 해당 DB에 저장할 수 있어야 한다.

3.5.2. Standards compliance

시스템은 웹 어플리케이션으로 JAVA와 Pigma를 이용해 개발된다. 변수 이름은 lower 카멜 케이스를 따르며, 함수와 데이터베이스는 파스칼 케이스를 따른다.

3.6. Software System Characteristics

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

3.6.1. Product Requirements

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

Usability Requirements

이 시스템은 비전문가와 전문가 모두를 대상으로 하는 그린 코드 교육 시스템이다. JAVA 언어에 대한 기초적 문법만 알고 있다면 이용 가능합니다. 또한 JAVA 언어로 구성된 코드가 얼마나 그린 코드에 가까운지 그 결과값을 쉽게 이해할 수 있는 UI가 제공되어야 한다.

Performance Requirements

탄소배출량과 최적화 코드에 대한 그린 코드 설명은 비전문가가 보아도 이해할 수 있는 수준이 될 정도로 코드 및 설명의 길이가 짧아야 하며, 탄소배출량 계산과 최적화 코드 출력까지 걸리는 실행 시간이 5 초를 넘기지 않도록 한다.

Security Requirements

사용자들은 시스템을 이용할 때 코드를 입력하는데, 이때 입력한 코드는 탄소배출량을 계산하는 데에만 사용되며 입력된 코드는 유출되어선 안 된다. 시스템 관리자를 제외한 일반 사용자는 시스템 관리자만큼의 권한을 가질 수 없다. 일반 사용자는 자신이 작성한 게시물의 수정 권한만 가질 수 있으며, 시스템 데이터베이스에 직접적으로 접근할 수 없도록 한다.

3.6.2. Organizational Requirements

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

Environmental Requirements

PC(데스크탑, 노트북 등) 환경에서 웹 브라우저 환경으로 접속하는 것을 권장한다.

Operational Requirements

이 시스템의 사용자들은 탄소 배출량 계산을 위해 코드를 입력하거나, 그린 코드에 대한 제안 및 수정을 요청할 수 있다. 또한 시스템 서버가 항상 가동되기에 사용자는 시간에 상관없이 자유롭게 접속하여 이용할 수 있다.

3.6.3. External Requirements

외부적 요구사항 항목에서는 시스템의 외부적 요인으로 인해서 발생하는 다양한 요구사항들에 관해서 설명한다.

Safety / Security Requirements

이 시스템에 저장된 사용자들의 어떤 개인 정보도 외부에 의한 접근을 허용해서는 안 된다. 또한 이 시스템에

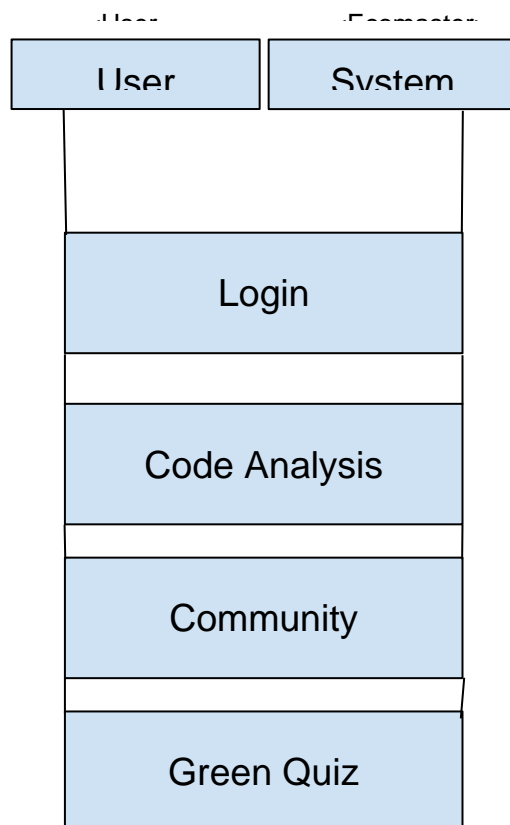
Regulatory Requirements

이 시스템에 저장되고 사용되는 모든 그린 코드 정보들은 탄소배출 계산과 그린 코드 퀴즈 출제를 위해서만 사용됩니다. 또한 그린 코드에 대한 제안 및 수정을 요청할 때 작성자에 대한 개인정보(IP 등)를 수집해서도 안 되며, 이 정보들이 유출되어서도 안 된다. 그 외의 경우 법에 따라 정보들을 보호하며, 시스템은 국제 개인 정보 보호 표준에 따라 개발되어야 한다.

3.7. 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

3.7.1. System mode



3.7.2. User class

- **일반적인 사용자:** 소스코드의 탄소배출량을 측정하고, 탄소배출량을 줄이기 위한 제안을 받기 위해 시스템을 사용하는 사용자들.
- **관리자:** 커뮤니티 존과 그린패턴 퀴즈존의 콘텐츠를 관리하고, 사용자들의 활동을 모니터링하는 역할을 수행하는 사용자들.

3.7.3. Objects

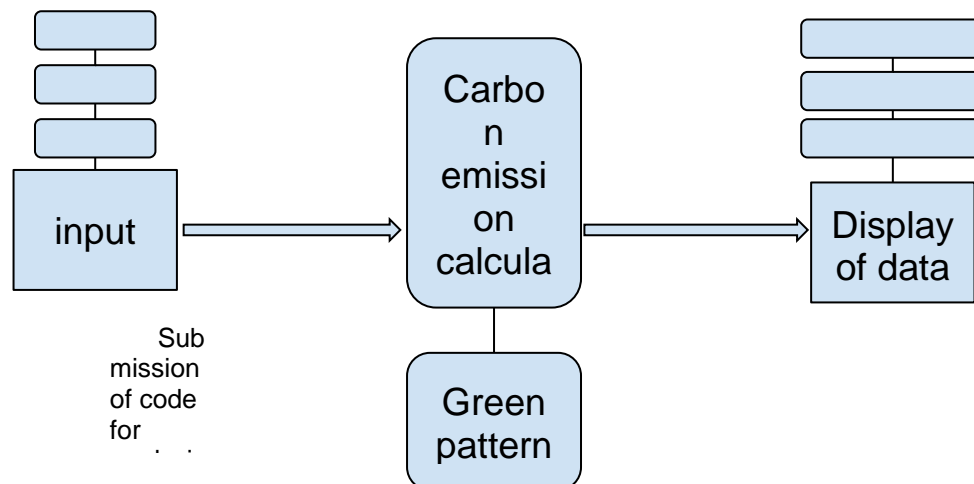
- **User (사용자):** 응용 프로그램과 상호 작용하는 개별 사용자를 나타냅니다. 각 사용자에는 사용자 이름, 비밀번호 및 프로필 정보와 같은 속성
- **Code solutions (코드):** 코드 자체, 메타데이터 및 분석 결과와 같은 속성

- **Community (커뮤니티):** 응용 프로그램의 커뮤니티 섹션에서 사용자가 작성한 게시물을 나타냅니다. 게시물 내용, 작성자 정보 및 타임스탬프와 같은 속성이 포함됩니다.
- **댓글:** 댓글 내용, 작성자 정보 및 타임스탬프와 같은 속성이 포함됩니다.
- **Quiz (퀴즈):** 응용 프로그램의 그린 퀴즈 섹션에 제시된 질문을 나타냅니다. 질문 텍스트, 옵션, 정답 및 관련 메타데이터와 같은 속성이 포함됩니다.
- **Learnings (응답):** 사용자가 퀴즈 질문에 대한 응답을 제공한 것을 나타냅니다. 선택한 옵션, 정확성 및 관련 메타데이터와 같은 속성이 포함됩니다.

3.7.4. Feature

- **소스 코드 분석 (Code Analysis):** 사용자가 제공한 코드 스니펫을 분석하여 탄소 배출량을 평가하고 최적화 영역을 식별하는 기능.
- **Carbon Emission Measurements (탄소 배출량 측정):** 분석된 코드 스니펫과 관련된 탄소 배출량을 측정하는 기능.
- **Code Optimization Suggestions (커뮤니티 피드백):** 탄소 배출량을 줄이기 위한 친환경적인 코드 개선에 대한 제안과 권고 기능.
- **Green Quiz Zone (그린 퀴즈 존):** 친환경 코딩 관행에 초점을 맞춘 퀴즈 섹션으로, 사용자를 educate 하고 환경 인식을 강화합니다.
- **Community Feedback (커뮤니티 피드백 메커니즘):** 사용자 인증 및 개인정보 관리: 안전한 사용자 인증 시스템과 사용자 개인 정보 (이름, 비밀번호 및 관련 정보를 관리할 수 있는 기능)

3.7.5. Stimulus



3.7.6. Response

- 사용자가 소스코드를 입력하면, 시스템은 해당 코드를 분석하여 탄소배출량을 측정하고 즉시 결과를 제공한다. 이때 결과는 탄소배출량의 양과 함께 어떤 부분이 환경에 미치는 영향을 가장 크게 하는지에 대한 정보도 포함한다. 또한, 시스템은 탄소배출량을 줄이기 위한 구체적인 제안을 제공하여 개발자가 코드를 최적화하고 환경 부하를 감소시킬 수 있도록 돕는다.
- 사용자가 커뮤니티 존에 접근하면 사용자들은 게시물을 올릴 권한을 얻으면 자유롭게 자신의 코드나 유용한 정보들을 올릴 수 있다.

- 사용자가 그린패턴 퀴즈존을 이용하면, 시스템은 환경 보호에 관련된 퀴즈를 제공하고 사용자의 지식 습득을 돕는다. 이러한 응답 기능은 사용자가 환경 보호에 대한 인식을 높이고 실제로 환경에 도움이 되는 행동을 취할 수 있도록 지원한다.

3.7.7. Functional hierarchy

- 소스코드 분석 및 탄소배출량 측정
 - 정적 분석
 - 동적 분석
 - 탄소배출량 감소를 위한 제안 제공
- 커뮤니티 존
 - 게시물 작성
 - 댓글 작성
- 그린패턴 퀴즈존
 - 퀴즈 풀기
 - 결과 확인

3.7.8. Additional comments

그린패턴 퀴즈 앱은 환경 인식과 지속 가능한 실천을 촉진하기 위해 노력합니다. 정기적인 업데이트와 커뮤니티 피드백 메커니즘은 앱이 관련성을 유지할 수 있도록 보장하는 데 필수적입니다.

3.8. System Evolution

이 절에서는 시스템의 기초가 되는 기본 가정과 하드웨어 진화, 사용자 요구 변화 등으로 인한 모든 예상할 수 있는 변화를 설명한다. 또한 **System Evolution** 은 시스템 설계자가 시스템의 향후 변경을 제약하는 문제가 생길 수 있는 설계의 결정을 피하는데 도움이 될 수 있으며 유용하게 사용될 것이다.

3.8.1. Limitation and Assumption

시스템 개발 및 운영 중에는 몇 가지 제약이 존재할 것으로 예상된다. 예를 들어, 소프트웨어는 코드의 정적 분석을 기반으로 탄소배출량을 추정하기 때문에, 코드에 대한 많은 알고리즘, 그린패턴의 충분한 정보와 그것을 제대로 분석할 수 있는 프로그램에 따라 실제 배출량과의 차이가 발생할 수 있다. 또한, 사용자는 정확한 결과를 얻기 위해 입력하는 정보의 정확성에 대해 책임을 져야 할 것으로 예상된다.

커뮤니티 존과 퀴즈 존의 시스템 개발 및 운영에는 여러 가지 제한과 가정이 존재할 것으로 예상된다. 예를 들어, 커뮤니티 존에서는 사용자들의 참여와 활발한 토론이 이루어져야 하므로, 적극적인 참여와 적절한 모니터링이 필요할 것으로 예상된다. 또한, 콘텐츠의 질과 다양성을 유지하기 위해서는 지속적인 업데이트와 품질 관리가 필요할 것으로 예상된다.

3.8.2. Evolution of Hardware and Change of User Requirements

시스템의 하드웨어 요구사항은 시간이 지남에 따라 변화할 것으로 예상된다. 초기에는 일반적인 개발자용 컴퓨터나 서버에서 실행될 수 있겠지만, 더 많은 데이터나 복잡한 분석을 위해 보다 강력한 하드웨어가 필요해질 수 있다. 또한, 사용자 요구사항은 기술적인 발전이나 환경 변화에 따라 변화할 것으로 예상되며, 이에 따라 시스템은 유연하게 적응하고 발전해야 할 것으로 예상된다.

그리고 커뮤니티 존과 퀴즈 존은 시간이 지남에 따라 하드웨어 요구사항과 사용자 요구사항이 변화할 것으로 예상된다. 특히, 커뮤니티 존에서는 사용자 수가 증가함에 따라 서버 및 네트워크 인프라의 확장이 필요할 수 있으며, 사용자들의 다양한 요구를 수용하기 위한 기능 추가 및 개선이 이루어질 것으로 예상된다. 또한, 환경 보호 및 탄소 중립에 대한 인식이 높아짐에 따라 사용자들의 요구사항이 변화할 것으로 예상되며, 이에 따라 적절한 콘텐츠 및 기능이 제공되어야 할 것으로 예상된다.

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
2024/05/05	V1.00	Overall description	임성훈
2024/05/05	V1.00	External Interface Requirements, System Evolution	양승환
2024/05/05	V1.00	Functional Requirements	김민성
2024/05/05	V1.00	Logistic Database Requirements, Design Constraints, Software system characteristics	황인성
2024/05/05	V1.00	Introduction, Organizing the specific requirements	문수현