**소프트웨어 요구사항 명세서**

**(Software Requirement Specification)**

탄소 배출량 측정 웹 서비스 개발

(Development of a web service for measuring carbon emissions)

2023년 10월 29일

성균관대학교

소프트웨어공학

Team. 9

2018312081 박태현

2018310169 김한얼

2021313747 이진혁

2019315441 이한준

2018312277 한상안

2020313088 한채윤

**목 차**

**1. 서론 (Introduction)**

1.1 목적 (Purpose)

1.2 범위 (Scope)

1.3 용어 및 약어 정리 (Definitions, Acronyms, and Abbreviations)

1.3.1 정의 (Definitions)

1.3.2 약어 (Abbreviations)

1.4 참고 문헌 (References)

1.5 개 요 (Overview)

**2. 종합 기술 (Overall Description)**

2.1 프로젝트 결과물 전망 (Product Perspective)

2.1.1 시스템 인터페이스 (System Interfaces)

2.1.2 사용자 인터페이스 (User Interface)

2.1.3 하드웨어 인터페이스 (Hard Interface)

2.1.4 소프트웨어 인터페이스 (Software Interfaces)

2.2 결과물의 기능 (Product Functions)

2.2.1 코드 입력 및 계산

2.2.2 히스토리 확인 및 비교

2.2.3 Sequence Diagram

2.3 사용자 유형 및 특성 (User Classes and Characteristics)

2.3.1 시스템 관리자 (System Administrator)

2.3.2 사용자 (User)

2.4 제약사항 (Design and Implement Constraints)

2.5 가정 및 의존성 (Assumptions and Dependencies)

**3. 상세 요구사항 (Specific Requirements)**

3.1 외부 요구사항 (External Interfaces)

3.1.1 UI 요구사항 (User Interface)

3.1.1.1 메인 페이지

3.1.1.2 히스토리 페이지

3.1.1.3 결과 비교 페이지

3.1.1.4 실험 조회 페이지

3.1.1.5 그린화 패턴 게시판 페이지

3.1.1.6 그린화 패턴 개별 게시물 페이지

3.2 기능 요구사항 (Functional Requirements)

3.2.1 Use Case

3.2.2 Use Case Diagram

3.2.3 Data Flow Diagram

3.3 성능 요구 사항 (Performance Requirements)

3.3.1 정적 예시 (Static Requirement)

3.3.2 동적 예시 (Dynamic Requirement)

3.4 논리적 데이터베이스 요구 사항 (Logical Database Requirements)

3.5 설계 제약 조건 (Design Constraints)

3.5.1 물리적 설계 제약 (Physical Design Constraints)

3.5.2 표준 준수 (Standards Compliance)

3.6 소프트웨어 시스템 속성 (Software System Attributes)

3.6.1 신뢰성 (Reliability)

3.6.2 가용성 (Availability)

3.6.3 보안성 (Security)

3.6.4 유지가능성 (Maintainability)

3.6.5 이식성 (Portability)

**4. 부록 (Appendixes)**

4.1 소프트웨어 요구사항 명세서 (Software Requirements Specification)

**1. 서론 (Introduction)**

**1.1 목적 (Purpose)**

이 문서는 코드의 탄소 배출을 계산하는 웹 서비스를 구현하는 과정의 requirements를 명확하게 하고, 추후 유지보수 과정에서 참고할 수 있도록 제작되었다.

**1.2 범위 (Scope)**

이 웹사이트는 입력 받는 Java 코드를 분석해 탄소 배출량을 계산하는 웹사이트이다.

사용자가 입력한 코드들을 저장하여 히스토리 페이지로 만들고, 비교 분석을 할 수 있게 한다.

**1.3 용어 및 약어 정리 (Definitions, Acronyms, and Abbreviations)**

**1.3.1 정의 (Definitions)**

- 실험

사용자가 자바코드를 제출 시 생성되는 런타임 및 탄소배출량에 대한 결과 리포트이다. ‘실험’의 내용으로는 제목, 탄소배출량, 런타임, 자바코드 그리고 각종 탄소배출량 변환 수치가 있다.

**1.3.2 약어 (Abbreviations)**

**1.4 참고 문헌 (References)**

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

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

**1.5 개 요 (Overview)**

본 소프트웨어 요구사항 명세서는 3개의 Section으로 구성된다. 첫 번째 Section은 본 문서의 개요로, 산출물의 범위, 용어 및 약어 정리, 참고 문헌으로 구성된다. 두 번째 Section은 종합적 기술로서 소프트웨어의 전체적인 조망, 기능, 사용자의 특성, 제약 사항, 가정 및 의존성, 요구사항 분배에 대해 서술한다. 세 번째 Section은 소프트웨어의 상세 요구사항으로, 외부 인터페이스 요구사항, 시스템 특징, 성능 요구사항, 설계 제약 조건으로 구성된다.

**2. 종합 기술 (Overall Description)**

**2.1 프로젝트 결과물 전망 (Product Perspective)**

이 과제의 배경은 개발자들의 탄소 배출 현황이 높아지고 있다는 데에서 시작한다. 과도한 탄소 배출은 지구 온난화의 원인이 되기 때문에, 개발자들은 탄소가 덜 배출되는 코드를 최대한 이용하는 것이 바람직하다. 따라서 이 과제의 목적은 사용자가 입력한 코드의 탄소 배출량을 계산하는 웹 서비스를 개발하는 것이다.

**2.1.1 시스템 인터페이스 (System Interfaces)**

사용자는 탄소 배출량을 계산하고자 하는 Java 코드를 입력해 본인이 짠 코드의 탄소 배출량을 확인할 수 있다. 또한 사용자는 히스토리 기능을 이용해 시스템 접속 후 사용자가 작성했던 코드들의 탄소 배출량을 확인할 수 있고, 비교 분석까지 진행할 수 있다.

**2.1.2 사용자 인터페이스(User Interfaces)**

인터페이스는 컴퓨터 화면을 통해 제공한다. 사용자는 입력한 코드의 탄소 배출량을 바로 확인할 수 있고, 히스토리 탭으로 이동해 기록 또한 확인할 수 있다.

**2.1.3 하드웨어 인터페이스(Hard Interfaces)**

본 시스템은 웹 어플리케이션으로 컴퓨터를 통해 사용할 수 있다. 요구되는 사양은 1.0GHz 이상의 싱글 프로세서 데스크탑 및 노트북이다.

**2.1.4 소프트웨어 인터페이스 (Software Interfaces)**

본 시스템은 적어도 Windows 7 이상의 OS에서 동작할 수 있으며 Windows10 및 Windows11의 환경을 타겟으로 하고 있다. 또한 Chrome 버전100.0, Microsoft Edge 버전 100.0, Firefox 버전 99이상의 웹 브라우저의 사용을 권장한다. 그리고 학습을 정상적으로 진행하기 위해 학생 및 교수의 웹 브라우저에서 JavaScript 실행을 허용해야 한다.

**2.2 결과물의 기능 (Product Functions)**

**2.2.1 코드 입력 및 계산**

사용자의 시스템 환경, 즉 코어 개수, 코어 타입, 코어 모델 및 사용가능한 메모리의 크기를 먼저 확인한다. 그 후 사용자는 코드 입력창에 Java 코드를 입력하고, 해당 Java 코드의 컴파일 에러 혹은 런타임 에러를 확인하여 예외처리를 진행한다. 정상적으로 실행이 되었다면 해당 코드의 탄소배출량 결과를 확인할 수 있다.

**2.2.2 히스토리 확인 및 비교**

사용자가 접속 후 입력했던 Java 코드의 기록을 히스토리 페이지에서 확인할 수 있다. 사용자는 각각의 히스토리에 설명을 추가하거나, 이름을 변경할 수 있고, 각 히스토리는 생성된 시간이 표시된다. 각 히스토리를 클릭하여 입력했던 코드와 탄소 배출량을 확인할 수 있고, 여러 히스토리를 이용해 탄소배출량 비교를 진행할 수 있다.

**2.2.3 Sequence Diagram**

**텍스트, 스크린샷, 도표이(가) 표시된 사진

자동 생성된 설명**

**2.3 사용자 유형 및 특성 (User Classes and Characteristics)**

**2.3.1 시스템 관리자 (System Administrator)**

웹 서비스에 충분한 지식을 가진 사람이 System administrator 역할을 할 수 있다. 컴퓨터 공학을 전공하였거나, 그에 준하는 웹 개발 관련 지식을 가진 사람이 추후 발생하는 문제를 다룰 수 있다.

**2.3.2 사용자 (User)**

실질적으로 코드를 작업하는 사람이 User 역할을 할 수 있다. Java 언어를 코드로 작성할 수 있는 사람은 코드 오류로 인한 문제를 발생시키지 않고 탄소 배출량 측정 서비스를 원활하게 사용할 수 있다.

**2.4 제약사항 (Design and Implement Constraints)**

탄소 배출량 측정 웹 서비스의 제약 사항은 다음과 같다.

* 입력되는 코드는 Java 언어로 제한됨.
* 개발은 Windows 10 환경을 대상으로 하였음.
* 입력되는 코드의 에러가 없어야 정상적인 측정이 가능함.
* 시스템 개발 및 유지 관리 비용을 고려해야 함.
* 시스템의 미래 확장성 및 가용성을 고려해야 함.

**2.5 가정 및 의존성 (Assumptions and Dependencies)**

웹 서비스의 목표는 탄소 배출량의 측정이기 때문에, 사용자는 Java 코드를 충분히 이해하고 잘 작성할 수 있다고 가정합니다. 또한 웹 기반의 서비스이기 때문에 앞서 언급된 최소 사양을 충족하지 않는 경우, 시스템이 원활하게 작동하지 않을 수 있다.

**3. 상세 요구사항 (Specific Requirements)**

**3.1 외부 요구사항 (External Interfaces)**

**3.1.1 UI 요구사항 (User Interface)**

**3.1.1.1 메인 페이지**

|  |  |
| --- | --- |
| 이름 | 메인 페이지 |
| 목적/내용 | 메인 페이지에 접속하였을 때 처음 보이는 페이지이다.  유저가 자바코드를 입력하고 탄소배출량 결과를 확인할 수 있다.  상단의 메뉴바를 통해 메인 페이지, 히스토리 페이지, 그린화 패턴 게시판 페이지로 넘어갈 수 있다. |
| 입력 주체/출력 목적지 | 클라이언트 / 서버  서버 / 클라이언트 |
| 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 | 사용자 입력에 따른 서버간 통신 시간, 사용자 입력에 따른 화면 전환 |
| 타 입출력과의 관계 | 없음. |
| 화면 형식 및 구성 | 1. Navigation bar    1. Logo – Navigation bar 좌측에 로고를 표시한다.    2. Nav Links – 로고 우측에 Home, History, Tips 페이지로 이동할 수 있는 링크 버튼이 표시된다. 해당 버튼을 클릭할 시, 각각 해당하는 페이지로 이동한다.    3. Tip Banner – Nav Links 우측에 그린화 패턴의 예시 중 랜덤으로 하나를 보여준다. 2. Body    1. Code section       1. Experiment name: code section 상단에 위치한다. 사용자가 시험결과를 저장할 때 저장할 이름을 입력한다.       2. Run button: code section 우측 상단에 위치한다. Code editor에 적힌 내용과 experiment name을 서버로 전송한다.       3. Code editor: 사용자가 코드를 입력하고 편집할 수 있는 영역이다.    2. Result section:       1. System specifications: result section 왼쪽에 위치한다. 서버의 시스템 사양을 표시한다.       2. Result: result section 오른쪽에 위치한다. 실험 결과를 탄소배출량의 수치뿐만 아니라 일상 생활의 탄소 배출량과 비교하여 보여준다. |
| 데이터 형식 및 구성 | 이미지, 텍스트 |
| 명령 형식 | 버튼 클릭 |
| 종료 메시지 | 없음. |

**3.1.1.2 히스토리 페이지**

|  |  |
| --- | --- |
| 이름 | 히스토리 페이지 |
| 목적/내용 | Navigation bar의 history link를 클릭했을 시 보여주는 페이지이다.  사용자가 지금까지 한 실험 결과를 확인할 수 있다.  실험 이름을 클릭하면 실험 조회 페이지로 이동할 수 있다.  실험 중 2개 이상을 선택하여 비교 버튼을 누르면 결과 비교 페이지로 넘어갈 수 있다.  상단의 메뉴바를 통해 메인 페이지, 히스토리 페이지, 그린화 패턴 게시판 페이지로 넘어갈 수 있다. |
| 입력 주체/출력 목적지 | 클라이언트 / 클라이언트 |
| 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 | 사용자 입력에 따른 화면 전환 |
| 타 입출력과의 관계 | 없음. |
| 화면 형식 및 구성 | 1. Navigation bar    1. Logo – Navigation bar 좌측에 로고를 표시한다.    2. Nav Links – 로고 우측에 Home, History, Tips 페이지로 이동할 수 있는 링크 버튼이 표시된다. 해당 버튼을 클릭할 시, 각각 해당하는 페이지로 이동한다.    3. Tip Banner – Nav Links 우측에 그린화 패턴의 예시 중 랜덤으로 하나를 보여준다. 2. Body    1. Result table       1. Record: experiment name / 탄소 배출량 / 런타임 을 표시한다. Experiment name을 클릭하면 실험 결과 조회 페이지로 이동한다.       2. Check box: record 왼쪽에 체크박스를 만들어 특정 record를 선택할 수 있게 한다.       3. Compare Button: record를 2개 이상 선택하면 클릭할 수 있고 클릭하면 결과 비교 페이지로 이동한다. |
| 데이터 형식 및 구성 | 이미지, 텍스트 |
| 명령 형식 | 버튼 클릭 |
| 종료 메시지 | 없음. |

**3.1.1.3 결과 비교 페이지**

|  |  |
| --- | --- |
| 이름 | 결과 비교 페이지 |
| 목적/내용 | History page에서 결과 비교 버튼을 클릭했을 시 보여주는 페이지이다.  사용자가 history에서 선택한 실험들을 비교하여 보여준다.  상단의 메뉴바를 통해 메인 페이지, 히스토리 페이지, 그린화 패턴 게시판 페이지로 넘어갈 수 있다. |
| 입력 주체/출력 목적지 | 클라이언트 / 클라이언트 |
| 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 | 사용자 입력에 따른 화면 전환 |
| 타 입출력과의 관계 | 없음. |
| 화면 형식 및 구성 | 1. Navigation bar    1. Logo – Navigation bar 좌측에 로고를 표시한다.    2. Nav Links – 로고 우측에 Home, History, Tips 페이지로 이동할 수 있는 링크 버튼이 표시된다. 해당 버튼을 클릭할 시, 각각 해당하는 페이지로 이동한다.    3. Tip Banner – Nav Links 우측에 그린화 패턴의 예시 중 랜덤으로 하나를 보여준다. 2. Body    1. Result section:       1. Result: 탄소 배출량, 런타임을 비교하여 보여준다. 일상 생활의 탄소 배출량으로 변환된 값도 비교하여 보여준다. |
| 데이터 형식 및 구성 | 이미지, 텍스트 |
| 명령 형식 | 버튼 클릭 |
| 종료 메시지 | 없음. |

**3.1.1.4 실험 조회 페이지**

|  |  |
| --- | --- |
| 이름 | 실험 조회 페이지 |
| 목적/내용 | History page에서 실험 이름을 클릭했을 시 보여주는 페이지이다.  사용자가 history에서 선택한 실험의 결과를 메인 페이지와 동일한 형태로 보여준다.  상단의 메뉴바를 통해 메인 페이지, 히스토리 페이지, 그린화 패턴 게시판 페이지로 넘어갈 수 있다. |
| 입력 주체/출력 목적지 | 클라이언트 / 클라이언트 |
| 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 | 사용자 입력에 따른 화면 전환 |
| 타 입출력과의 관계 | 없음. |
| 화면 형식 및 구성 | 1. Navigation bar    1. Logo – Navigation bar 좌측에 로고를 표시한다.    2. Nav Links – 로고 우측에 Home, History, Tips 페이지로 이동할 수 있는 링크 버튼이 표시된다. 해당 버튼을 클릭할 시, 각각 해당하는 페이지로 이동한다.    3. Tip Banner – Nav Links 우측에 그린화 패턴의 예시 중 랜덤으로 하나를 보여준다. 2. Body    1. Code section       1. Code: 사용자가 입력했던 코드를 보여준다.    2. Result section:       1. System specifications: result section 왼쪽에 위치한다. 서버의 시스템 사양을 표시한다.       2. Result: result section 오른쪽에 위치한다. 실험 결과를 탄소배출량의 수치뿐만 아니라 일상 생활의 탄소 배출량과 비교하여 보여준다. |
| 데이터 형식 및 구성 | 이미지, 텍스트 |
| 명령 형식 | 버튼 클릭 |
| 종료 메시지 | 없음. |

**3.1.1.5 그린화 패턴 게시판 페이지**

|  |  |
| --- | --- |
| 이름 | 그린화 패턴 게시판 페이지 |
| 목적/내용 | Navigation bar의 Tips를 클릭했을 시 보여주는 페이지이다.  그린화 패턴 예시를 적어 두어 사용자가 참고할 수 있도록 만든 페이지이다.  게시글을 클릭하면 상세한 설명을 확인할 수 있다.  상단의 메뉴바를 통해 메인 페이지, 히스토리 페이지, 그린화 패턴 게시판 페이지로 넘어갈 수 있다. |
| 입력 주체/출력 목적지 | 클라이언트 / 클라이언트 |
| 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 | 사용자 입력에 따른 화면 전환 |
| 타 입출력과의 관계 | 없음. |
| 화면 형식 및 구성 | 1. Navigation bar    1. Logo – Navigation bar 좌측에 로고를 표시한다.    2. Nav Links – 로고 우측에 Home, History, Tips 페이지로 이동할 수 있는 링크 버튼이 표시된다. 해당 버튼을 클릭할 시, 각각 해당하는 페이지로 이동한다.    3. Tip Banner – Nav Links 우측에 그린화 패턴의 예시 중 랜덤으로 하나를 보여준다. 2. Body    1. Board section:       1. Title: 그린화 패턴에 대한 제목을 보여준다. 클릭 시 해당 게시물의 페이지로 넘어간다. |
| 데이터 형식 및 구성 | 이미지, 텍스트 |
| 명령 형식 | 버튼 클릭 |
| 종료 메시지 | 없음. |

**3.1.1.6 그린화 패턴 개별 게시물 페이지**

|  |  |
| --- | --- |
| 이름 | 그린화 패턴 개별 게시물 페이지 |
| 목적/내용 | 그린화 패턴 게시판 페이지에서 게시물을 클릭했을 시 보여주는 페이지이다.  그린화 패턴 예시를 적어 두어 사용자가 참고할 수 있도록 만든 페이지이다. 자세한 설명이 포함되어 있다.  상단의 메뉴바를 통해 메인 페이지, 히스토리 페이지, 그린화 패턴 게시판 페이지로 넘어갈 수 있다. |
| 입력 주체/출력 목적지 | 클라이언트 / 클라이언트 |
| 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 | 사용자 입력에 따른 화면 전환 |
| 타 입출력과의 관계 | 없음. |
| 화면 형식 및 구성 | 1. Navigation bar    1. Logo – Navigation bar 좌측에 로고를 표시한다.    2. Nav Links – 로고 우측에 Home, History, Tips 페이지로 이동할 수 있는 링크 버튼이 표시된다. 해당 버튼을 클릭할 시, 각각 해당하는 페이지로 이동한다.    3. Tip Banner – Nav Links 우측에 그린화 패턴의 예시 중 랜덤으로 하나를 보여준다. 2. Body    1. Board section:       1. Title: 그린화 패턴에 대한 제목을 보여준다.       2. Contents: 그린화 패턴에 대한 상세한 설명을 표시한다. |
| 데이터 형식 및 구성 | 이미지, 텍스트 |
| 명령 형식 | 버튼 클릭 |
| 종료 메시지 | 없음. |

**3.2 기능 요구사항 (Functional Requirements)**

**3.2.1 Use case**

|  |  |
| --- | --- |
| Actor | 세션키가 없는 사용자(첫 방문 사용자) |
| Description | 세션키가 발급 받지 않은 사용자가 세션키를 쿠키에 저장하는 과정이다. |
| Normal Course | 1. 모든 사용자는 쿠키 권한에 대해 동의하는지 확인하는 모달이 나타난다. 2. 쿠키 권한 관련 내용을 확인하고 동의 버튼을 누른다. 이때 웹사이트에서는 서버로 세션키 발급 요청을 한다. 3. 서버에서는 세션키를 발급하고 웹사이트에서는 해당 세션키를 쿠키로 저장한다. |
| Precondition | * 사용자는 사이트에 접속할 때 세션키를 쿠키로 저장하고 있지 않아야 한다. * 세션키는 랜덤으로 생성되며 다른 사용자와 중복되지 않도록 하여야 한다. |
| Post Condition | 발급된 세션키는 브라우저 쿠키에 저장이 되고, 실험을 생성할 때 마다 현재 세션키가 해당 실험의 owner가 된다. |
| Assumptions | 해당 사항 없음 |

|  |  |
| --- | --- |
| Actor | 세션키를 가지고 있는 사용자 |
| Description | 사용자가 코드 에디터에 직접 코드를 작성하거나, 파일을 선택하면 에디터에 해당 코드가 입력이 된다. 그리고 자바 코드를 제출하면 코드에 대한 탄소배출량 및 런타임 결과를 담고 있는 ‘실험’이 생성된다. |
| Normal Course | 1. 사용자가 코드 에디터에 코드를 입력한다.    1. copy & paste 또는 직접 입력 입력한다.    2. 파일 선택을 하여 파일 내용을 코드 에디터로 불러온다. 2. 제출 버튼을 눌러 서버로 코드 내용을 전송한다. 3. 해당 코드에 대한 실험(결과)가 생성된 것을 확인한다. |
| Precondition | * 사용자는 세션키를 가지고 있어야 한다. * 입력한 자바 코드는 실행가능한 코드이다. |
| Post Condition | 생성된 실험은 데이터베이스에 업데이트 된다. |
| Assumptions | 해당 사항 없음 |

|  |  |
| --- | --- |
| Actor | 세션키를 가지고 있는 사용자 |
| Description | 사용자가 이전에 생성한 단일 실험에 대한 내용을 확인하는 과정이다. |
| Normal Course | 1. 웹사이트의 탭에서 history 탭을 선택하여 접속한다. 2. 실험 리스트(본인이 이전에 생성한 실험) 중에서 조회하고자 하는 실험 하나를 클릭한다. 3. 실험 리스트 밑에 해당 실험에 대한 결과를 확인한다. |
| Precondition | * 사용자는 세션키를 가지고 있어야 한다. * 사용자는 현재 세션키로 이전에 생성한 실험이 한 개 이상이다. |
| Post Condition | 단일 실험에 대한 탄소배출량 및 런타임 결과가 리스트 하단에 표출된다. |
| Assumptions | 해당 사항 없음 |

|  |  |
| --- | --- |
| Actor | 세션키를 가지고 있는 사용자 |
| Description | 사용자가 이전에 생성한 다중 실험에 대한 내용을 비교하는 과정이다. |
| Normal Course | 1. 웹사이트의 탭에서 history 탭을 선택하여 접속한다. 2. 실험 리스트(본인이 이전에 생성한 실험) 중에서 조회하고자 하는 실험 두 개 이상 선택한다. 3. 다중 실험에 대해 런타임, 탄소배출량, 코드를 비교하는 페이지로 이동하여 확인한다. |
| Precondition | * 사용자는 세션키를 가지고 있어야 한다. * 사용자는 이전에 생성한 실험이 두 개 이상이다. |
| Post Condition | 다중 실험에 대해 비교하는 페이지(또는 모달)이 표출된다. |
| Assumptions | 해당 사항 없음 |

**3.2.2 Use case Diagram**

**텍스트, 도표, 라인, 스크린샷이(가) 표시된 사진

자동 생성된 설명**

**3.2.3 Data Dictionary**

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment** | | | |
| **Field** | **Key** | **Constraint** | **Description** |
| id | PK | Not null | Experiment ID |
| title |  | Default: str(idx) | Experiment title |
| footprint |  | Not null | Carbon footprint result |
| run\_time |  | Not null | Running time of java code |
| code |  | Not null | Content of java code |
| car\_index |  | Not null | Conversion result of automobile carbon emissions |
| plane\_index |  | Not null | Conversion result of airplane carbon emissions |
| tree\_index |  | Not null | Conversion results of tree carbon purification |
| Create\_time |  | Not null | Created time |

**3.2.3 Data Flow Diagram**

**텍스트, 도표, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명**

**3.3 성능 요구 사항 (Performance Requirements)**

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

**3.3.1 정적 예시 (Static Requirement)**

* 시스템은 여러 명의 사용자가 동시에 탄소배출량을 확인할 수 있도록 지원한다.
* 1Mbps 이상의 인터넷 연결 속도 환경을 요구한다. 운영체제는 윈도우 7 이상이며, 웹 어플리케이션을 지원하는 환경에서 정상적으로 동작한다.

**3.3.2 동적 예시 (Dynamic Requirement)**

* 시스템은 최소한 동시에 100명의 사용자 접속을 유지할 수 있어야 한다. 각 웹페이지는 동시에 최소한 100명의 사용자가 접속하여도 부드럽게 동작하여야 한다.
* 사용자가 코드를 입력하고 탄소 배출량을 확인하는데 걸리는 시간은 3분 이내로 완료되어야 한다. 계산된 결과는 5초 이내 DB에 저장되고, 이를 나타낼 수 있어야 한다.
* 사용자가 다른 페이지로 이동할 시 해당 페이지가 5초 이내로 표시되어야 한다.
* 사용자의 히스토리 비교 분석 요청을 5초 이내로 완료해야 한다.

**3.4 논리적 데이터베이스 요구 사항 (Logical Database Requirements)**

시스템에는 사용자가 생성한 실험들이 저장되고 조회된다. 사용자는 현재 자신의 세션키로 생성된 히스토리에 대해서만 접근할 수 있다. 사용자의 접속이 끊어지면 해당 데이터는 전부 초기화되며, 히스토리 페이지의 기록 또한 사라져야 한다. 해당 히스토리는 코드 내용, 입력된 시간, 탄소 배출량이 포함되어 있어야 한다.

**3.5 설계 제약 조건 (Design Constraints)**

**3.5.1 물리적 설계 제약 (Physical Design Constraints)**

시스템의 목적은 사용자의 Java 코드의 탄소 배출량 측정이므로, 데스크탑 또는 노트북 기기를 이용할 것을 권장하며, 모바일 기기 환경은 권장하지 않는다.

**3.5.2 표준 준수 (Standards Compliance)**

시스템은 웹 어플리케이션으로, JavaScript React와 Python Django를 이용해 개발되며, Frontend application과 Backend application은 RESTful API로 통신한다.

**3.6 소프트웨어 시스템 속성 (Software System Attributes)**

**3.6.1 신뢰성 (Reliability)**

Java 코드의 탄소배출량을 계산할 때, 코드를 실행하는 시스템 정보를 명확하게 나타내어 탄소배출량 계산의 정확도 요구를 충족시킨다. 또한, 탄소배출량 계산식과 해당하는 변수의 값을 User에게 제공하여 User가 탄소배출량 결과를 신뢰할 수 있게 한다.

만약 Java 코드에서 컴파일 에러 혹은 런타임 에러 등이 발생하여 탄소배출량 결과를 얻을 수 없는 경우에는 해당 에러 결과를 User에게 명시하여서 에러를 수정하고 다시 코드를 입력할 수 있도록 한다.

**3.6.2 가용성 (Availability)**

웹서비스에 접속하여 이용할 때, 세션이 유지되는 동안 User가 입력한 코드와 이에 대한 분석 결과는 User가 History 창을 통해 자유롭게 접근하여 탄소배출량 결과 비교를 가능하게 한다. History 창 저장된 분석 결과는 User가 입력한 Code에 대한 간단한 설명을 추가할 수 있게 하여 체크포인트를 세션이 유지되는 내에서 설정할 수 있게 한다.

**3.6.3 보안성 (Security)**

User가 입력한 Java 코드를 다른 User가 접근하지 못하게 관리한다. 또한, Data dictionary 내 Code를 저장하는 상황에서, Code의 내용이 아닌 File\_path와 같은 파일 경로를 저장하도록 한다. 추가적으로, Session Key를 시간과 UUID를 활용해 중복이 발생하지 않도록 한다. 또한 다중 IP 접속을 방지하여 User의 히스토리를 다른 User가 확인할 수 없도록 한다.

**3.6.4 유지가능성 (Maintainability)**

탄소배출량 측정 웹서비스의 웹페이지 종류는 크게, Main, History, Result 페이지로 나눌 수 있다. 각 웹페이지는 별도의 모듈로 구성되며, 이러한 모듈은 개별적으로 관리된다. 따라서, 웹서비스에 문제가 생겼을 때, 해당 부분을 효율적으로 수정할 수 있다. 특히 Main에서는 코드 입력 칸, 배너, 시스템 정보 등 다양한 정보를 나타내는 부분이 있는데 이를 개별적으로 관리한다. 그리고 웹서비스의 코드, 아키텍쳐 등은 문서화를 유지하여 서비스를 개선시키는데 도움이 되도록 한다.

**3.6.5 이식성 (Portability)**

해당 웹서비스는 Windows 운영 체제 외에도 Firefox, Chrome 등의 웹 브라우저를 통해 다양한 플랫폼에서 접속 가능하게 한다. 또한, 데스크톱 컴퓨터와 같은 전자통신 장치, 혹은 휴대전화 및 태블릿과 같은 모바일 장치에서도 웹 브라우저를 통해 접속 가능하게 하여 호환성을 갖춘다.

웹서비스를 원활하게 이용하기 위해서 요구되는 브라우저 버전은 Chrome 98, Firefox 98이며 지속적인 브라우저 버전 업데이트를 권장한다.

**4. 부록 (Appendixes)**

**4.1 소프트웨어 요구사항 명세서 (Software Requirements Specification)**

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