

식품 섭취 일지 분석 프로그램

1 차 기획서

전공기초프로젝트 2 화목반 1 조

방 준 식 (201711396)

원 성 혁 (201711403)

정 준 원 (201711425)

한 범 석 (201711434)

Contents

1. 개요	5
2. 용어	5
3. 기본 사항	6
3.1. 작동 환경	6
3.2. 프로그램 구성 및 설치	7
3.3. 프로그램 실행	7
4. 프로그램 사용 흐름도	8
5. 데이터 요소	9
5.1. 식품명	9
5.2. 제품명	10
5.3. 바코드 번호	12
5.4. 사용자 정보	12
5.4.1. ID	12
5.4.2. PW	13
5.4.3. 성별	13
5.4.4. 나이	13
5.4.5. 키	14
5.4.6. 몸무게	14
5.5. 연(년도)	14
5.6. 월	14
5.7. 일	14

6. 데이터 파일	15
6.1. 식품영양정보 파일	15
6.2. 유해식품정보 파일	16
6.3. 사용자 정보 파일	18
6.4. 사용자 PW 파일	18
6.5. 식품 섭취 일지 파일	18
6.6. 섭취 제품명 일지 파일	19
7. 시작 메뉴 프롬프트	19
7.1. 회원가입 프롬프트	20
7.1.1. ID입력 프롬프트	20
7.1.2. PW입력 프롬프트	22
7.1.3. 회원정보 입력 프롬프트.....	24
7.1.3.1. 성별 입력 프롬프트	24
7.1.3.2. 나이 입력 프롬프트	24
7.1.3.3. 키 입력 프롬프트	25
7.1.3.4. 몸무게 입력 프롬프트	26
7.2. 로그인 프롬프트	27
7.2.1. ID입력 프롬프트	27
7.2.2. PW입력 프롬프트	27
8. 주 프롬프트: 메인 메뉴	28
8.1. 부 프롬프트1: 일지 작성 명령	29
8.1.1. 부 프롬프트1.1: 식품명 입력	29

8.1.1.1. 부 프롬프트1.1.1: 유사 식품명 선택	30
8.1.2. 부 프롬프트1.2: 제품명 입력	32
8.2. 부 프롬프트2: 일지 불러오기 명령	32
8.2.1. 부 프롬프트2.1: 연도(Year)와 월(Month) 입력	32
8.2.2. 부 프롬프트2.2: 일(Day) 입력 및 일지 보여주기	34
8.2.3. 부 프롬프트2.3: 수행할 동작 입력	36
8.2.3.1. 부 프롬프트2.3.1: 일지 수정하기	37
8.2.3.2. 부 프롬프트2.3.2: 일지 삭제하기	38
8.3. 부 프롬프트3: 일지 분석 명령	38
8.3.1. 부 프롬프트3.1: 영양정보 분석	39
8.3.1.1. 부 프롬프트3.1.1: 날짜 지정해서 분석	41
8.3.2. 부 프롬프트3.2: 유해식품 검사	43
8.4. 부 프롬프트4: 유해식품 검색 명령	44
8.4.1. 부 프롬프트4.1: 제품명으로 검색	45
8.4.2. 부 프롬프트4.2: 바코드 번호로 검색	47
8.5 종료 명령: "0"번 메뉴	48

1. 개요

본 프로그램은 공공데이터를 활용한 식품 섭취 일지 분석 프로그램이다.

Python으로 작성된 CLI 프로그램으로서, 사용자는 자신이 섭취한 식품을 1일 단위로 기록(일지 생성, 불러오기, 수정, 삭제)할 수 있으며, 사용자 이름으로 만들어진 폴더 안에 텍스트 파일로 저장된다. 저장된 일지는 식품의약품안전처 공공데이터(API)에서 받아온 정보를 저장한 텍스트 파일을 활용하여, 1일 단위로 섭취한 영양소를 계산해주고 권장 섭취량과 차이를 출력하며, 유해 식품을 검색할 수 있다.

2. 용어

사용자 폴더: 프로그램 실행 파일이 존재하는 폴더의 User 라는 이름의 폴더 안에 있는 사용자 ID 를 이름으로 갖는 폴더, 예를 들어 D:\wcse 라는 폴더에 프로그램 실행 파일이 있는 상태로 실행하고 computer1 이라는 ID 로 로그인 했을 경우, 사용자 폴더의 경로는 D:\wcse\Users\computer1 이 된다.

업데이트: 프로그램 동작에 필요한 데이터를 API 서버에 있는 최신 데이터로 갱신하는 것

공백: 표준 키보드의 스페이스 바로 입력할 수 있는 문자

숫자: 표준 키보드로 직접 입력할 수 있는 10 종류의 인도-아라비아 숫자들

완전한 한글: 자음과 모음의 조합으로 이루어진 한글 문자 ('ㄱ', 'ㄴ', 'ㄷ', 'ㄹ', 'ㅁ' 등 자음 또는 모음만 있는 문자는 완전한 한글이 아니고, '가', '각' 등 자음과 모음의 조합으로 이루어진 한글 문자를 본 문서에서 완전한 한글이라고 한다.)

개행(혹은 개행 문자): CR 문자, LF 문자, CR/LF 조합을 총칭

행 (줄): 텍스트 파일의 맨 처음이나 어떤 개행 직후부터, 이후 등장하는 첫번째 개행이나 파일의 맨 끝 (EOF) 직전까지. 이 정의에 따르면:

- 하나의 행 속에는 개행이나 EOF가 포함되지 않음
- 곧바로 인접한 두 행들은 한 개의 개행으로 서로 구분됨

오늘: 프로그램을 실행하고 있는 컴퓨터 시스템의 현재 시스템 날짜

메뉴: 프로그램 실행 중 고를 수 있도록 제시되는 명령이나 데이터, 혹은 하위 메뉴(재귀적 정의)들의 목록

식품영양정보 파일: 식품의약품안전처 데이터 활용 서비스'에서 제공하는 '식품영양성분 DB' 공공데이터를 API를 이용해 JSON 형태로 받아와 저장한 프로그램 실행 파일이 존재하는 폴더와 같은 폴더 안에 존재하는 식품영양정보.txt 파일

유해식품정보 파일: '식품의약품안전처 데이터 활용 서비스'에서 제공하는 '회수, 판매중지 정보' 공공데이터를 API를 이용해 JSON 형태로 받아와 저장한 프로그램 실행 파일이 존재하는 폴더와 같은 폴더 안에 존재하는 유해식품정보.txt 파일

식품영양정보 데이터: 본 프로그램에서 식품영양정보 파일을 읽어온 뒤, 본 프로그램 작동에 알맞은 형태의 자료구조로 변환되어 메모리에 존재하는 정보

유해식품정보 데이터: 본 프로그램에서 유해식품정보 파일을 읽어온 뒤, 본 프로그램 작동에 알맞은 형태의 자료구조로 변환되어 메모리에 존재하는 정보

식품 섭취 일지: 사용자가 오늘 섭취한 식품명과 제품명을 기록한 일지, 파일로 저장됨

식품명: 사용자가 식품 섭취 일지에 기록하려는 목적으로 입력한 식품의 이름, 본 프로그램에서 사용될 수 있는 식품명은 식품영양정보 데이터에 있는 식품 이름이어야 한다.

제품명: 사용자가 유해식품 검사를 위한 목적으로 식품 섭취 일지의 제품명 항목에 기록하거나, 직접 유해식품 검색을 할 경우 검색어로 사용되는 제품의 고유한 이름.

3. 기본 사항

3.1. 작동 환경

표준 Python3 인터프리터가 설치되어 있는 MS Windows 10 상의 IDLE 창에서 작동한다.

- Python 3.8이 설치된 환경에서 작동한다.
- 다른 버전의 Python 3.x 버전에서도 작동될 수 있지만, 보장되지 않는다.
- Python 2 버전에서는 작동하지 않는다.
- Python 3.8이 설치되고 환경 변수가 정상적으로 등록된 MS Windows 10의 CMD

창, PowerShell창 및 Python3의 IDLE창에서 CLI 기반으로 작동한다.

- Python 3.8이 설치된 다른 OS에서도 작동될 수 있지만, 보장되지 않는다.

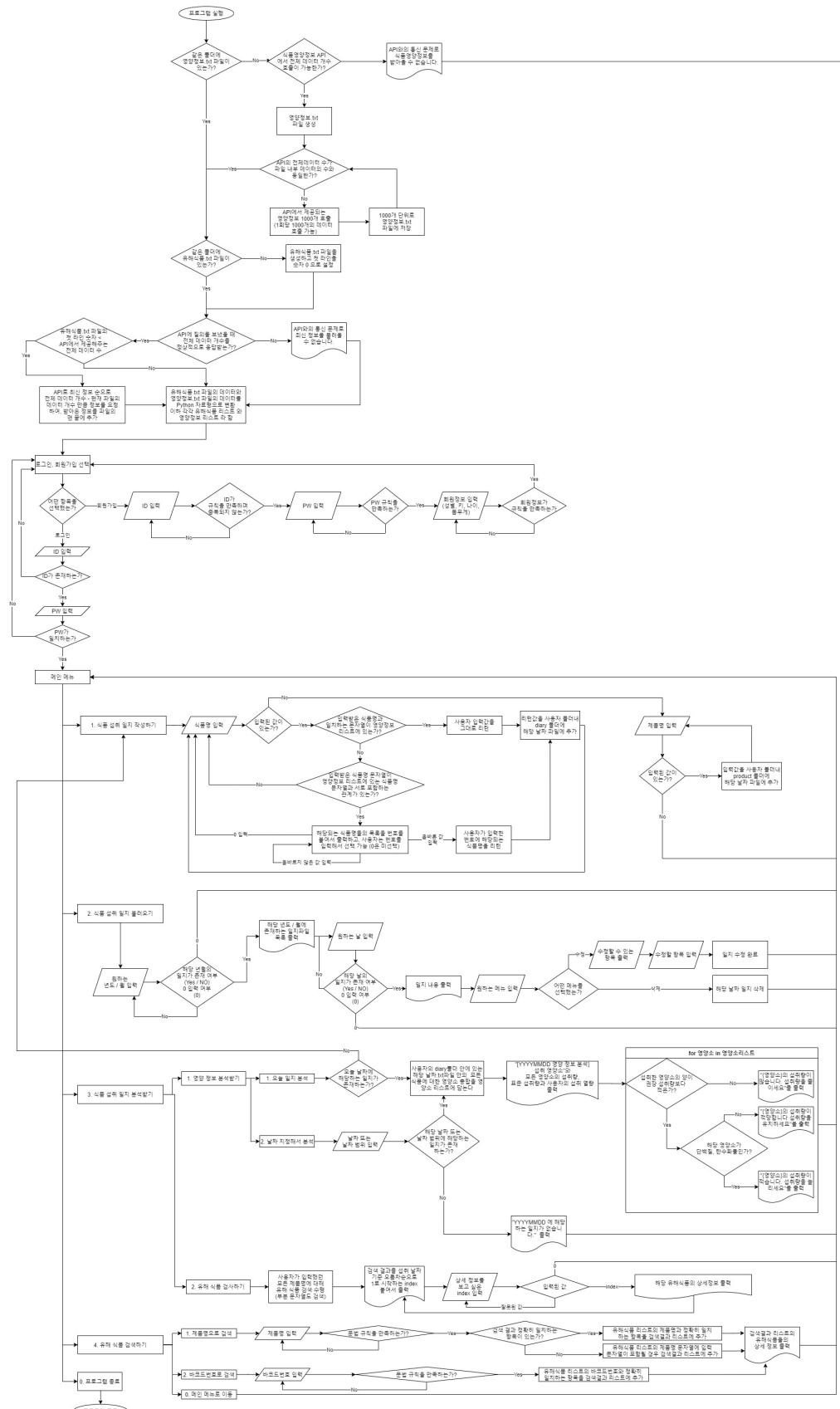
3.2. 프로그램 구성 및 설치

- 프로그램이 배포될 때에는, 주 실행 파일인 main.py와 프로그램 동작에 필요한 python 파일인 badfood_info.py, check.py, diary.py, food_info.py, input.py, loadDiary.py, login.py, menu.py, nutrient_analysis.py, search.py와 프로그램 동작에 필요한 데이터를 담고 있는 파일인 식품영양정보.txt와 유해식품정보.txt파일이 배포된다.
- 본인에게 '실행'과 '쓰기'권한이 있는 경로에 main.py, 영양정보.txt, 유해식품.txt 파일을 복사하면 설치된다.
- 명령행 인자가 없이 프로그램이 실행된다.
- 만약 main.py와 같은 폴더에 식품영양정보.txt 파일 또는 유해식품정보.txt 파일중에서 1개라도 없다면, 인터넷이 연결되어 있고 식품의약품안전처 데이터 활용 서비스에서 제공하는 API와 통신이 가능한 상태이어야 한다.
- 만약 유해식품 정보를 업데이트 하고 싶다면, 인터넷이 연결되어 있고 식품의약품안전처 데이터 활용 서비스에서 제공하는 API와 통신이 가능한 상태이어야 한다.

3.3 프로그램 실행

- Python 3.8이 설치되고 환경 변수가 정상적으로 등록된 MS Windows 10의 CMD 창 또는 PowerShell창에서 프로그램 파일이 위치한 폴더로 이동한 뒤, python main.py 를 입력해서 실행한다.
- IDLE로 실행할 경우, IDLE로 주 실행 파일을 열고 F5키를 눌러 실행시킬 수 있다.

4. 프로그램 사용 흐름도



5. 데이터 요소

5.1. 식품명

문법 형식: 문법적으로 올바른 식품명은 다음 항목들을 모두 만족하는 문자열이다.

- 다음의 항목들 이외에 다른 형태의 문자열을 포함할 수 없다.
 - 자음과 모음이 단독으로 존재하지 않고 조합이 완료된 완전한 한글 문자
 - 와일드카드 문자로서 "?" 기호
 - 와일드카드 문자로서 "*" 기호
- "*" 기호는 2개 이상이 연속으로 올 수 없다.
- 완전한 한글 문자가 1글자 이상 있어야 한다. (와일드카드 문자로만 이루어진 문자열은 문법 형식을 만족하지 못한다.)
- 글자수는 20글자 미만이어야 한다.

검색: 사용자가 입력한 식품명을 검색어로 하여, 식품영양정보 데이터의 식품명들 중에서 검색을 수행하는 경우는 완벽히 일치하는 항목을 찾는 경우와 유사한 항목을 찾는 경우로 나뉜다.

완벽히 일치하는 항목이 있는지 찾는 경우에는, 사용자가 입력한 식품명 문자열과 식품영양정보 데이터의 식품명 문자열이 완벽히 일치하면 매치된 것으로 간주한다. 완벽히 일치하는 문자열이 없을 경우, 아래와 같이 유사한 항목을 찾게 된다.

유사한 항목이 있는지 찾는 경우에는, 와일드카드를 이용한 검색과 부분 문자열을 이용한 검색이 있다.

와일드카드 문자가 1개 이상 포함되어 있을 경우에는

- 와일드카드를 이용한 검색을 수행하고, 그 결과가 없을 경우에는 검색 결과가 없는 것으로 검색이 종료된다. 즉 부분 문자열을 이용한 검색은 수행되지 않는다.
- 와일드카드 문자의 표준 해석 방법에 맞게 해석하여 검색을 수행한다.
- 와일드카드 문자의 표준 해석 방법에 맞게 해석했을 때 식품영양정보 데이터의 식품명들 중에서 매치되는 항목을 찾았더라도 검색을 종료하는 것이 아니라, 식품영양정보 데이터의 모든 식품명과 와일드카드를 이용한 검색을 수행하며, 매치

되는 항목을 리스트에 추가한다.

와일드카드 문자가 없는 경우에는

- 부분 문자열을 이용한 검색을 수행한다.
- 사용자가 입력한 식품명이 식품영양정보 데이터의 식품명의 부분 문자열이 되는 경우와 그 반대의 경우 모두 매치된 것으로 간주한다.
- 매치되는 항목을 찾았더라도 검색을 종료하는 것이 아니라, 식품영양정보 데이터의 모든 식품명과 부분 문자열을 이용한 검색을 수행하며, 매치되는 항목을 리스트에 추가한다.

5.2. 제품명

문법 형식: 문법적으로 올바른 제품명은 다음 항목들을 모두 만족하는 문자열이다.

- 다음의 항목들 이외에 다른 형태의 문자열을 포함할 수 없다.
 - 자음과 모음이 단독으로 존재하지 않고 조합이 완료된 완전한 한글 문자
 - 영문자 (대문자와 소문자 모두 가능)
 - 공백 문자
 - 와일드카드 문자로서 "?" 기호
 - 와일드카드 문자로서 "*"기호
- "*"기호는 2개 이상이 연속으로 올 수 없다.
- 완전한 한글 문자 또는 영문자가 1글자 이상 있어야 한다. (와일드카드 문자와 공백으로만 이루어진 문자열은 문법 형식을 만족하지 못한다.)
- 글자수는 20글자 이하여야 한다.

검색: 사용자가 입력한 제품명을 검색어로 하여, 유해식품 데이터의 제품명들 중에서 일치하는 항목이 있는지 찾을 때에는

- 검색어인 사용자가 입력한 제품명 내부의 공백들을 전부 없앤 상태로(검색 대상인 유해식품 데이터의 제품명은 이미 모두 내부의 공백들을 없앤 상태로 저장되어 있다.)

- 영문자 대/소문자를 구분하지 않으면서
- 검색어인 사용자가 입력한 제품명 문자열과 검색 대상인 유해식품 데이터의 제품명 문자열이 완벽히 일치하면 매치된 것으로 간주하고 검색이 종료된다.
만약 매치된 항목이 없으면 아래 부연한 와일드카드 또는 부분 문자열을 이용한 검색을 수행한다.
즉 매치된 항목이 있으면 부분 문자열을 이용한 검색을 수행하지 않는다.

완벽히 일치하는 항목이 없고, 와일드카드 문자가 1개 이상 포함되어 있을 경우에는

- 와일드카드를 이용한 검색을 수행하고, 그 결과가 없을 경우에는 검색 결과가 없는 것으로 검색이 종료된다. 즉 부분 문자열을 이용한 검색은 수행되지 않는다.
- 검색어인 사용자가 입력한 제품명 내부의 공백들을 전부 없앤 상태로
- 영문자 대/소문자를 구분하지 않으면서
- 와일드카드 문자의 표준 해석 방법에 맞게 해석하여 검색을 수행한다.
- 와일드카드 문자의 표준 해석 방법에 맞게 해석했을 때 유해식품 데이터의 제품명들 중에서 매치되는 항목을 찾았더라도 검색을 종료하는 것이 아니라, 유해식품 데이터의 모든 제품명과 와일드카드를 이용한 검색을 수행하며, 매치되는 항목을 리스트에 추가한다.

완벽히 일치하는 항목이 없고, 와일드카드 문자가 없는 경우에는

- 부분 문자열을 이용한 검색을 수행한다.
- 검색어인 사용자가 입력한 제품명 내부의 공백들을 전부 없앤 상태로
- 영문자 대/소문자를 구분하지 않으면서
- 검색어인 사용자가 입력한 제품명이 검색 대상인 유해식품 데이터의 제품명의 부분 문자열이면 매치된 것으로 간주한다. 즉 반대의 경우로 유해식품 데이터의 제품명이 사용자가 입력한 제품명의 부분 문자열이면 매치되지 않은 것이다.
- 매치되는 항목을 찾았더라도 검색을 종료하는 것이 아니라, 식품영양정보 데이터의 모든 식품명과 부분 문자열을 이용한 검색을 수행하며, 매치되는 항목을 리스트에 추가한다.

5.3. 바코드 번호

문법 형식: 문법적으로 올바른 바코드 번호는 다음 항목들을 모두 만족하는 문자열이다.

- 다음의 항목들 이외에 다른 형태의 문자열을 포함할 수 없다.
 - 숫자
 - 공백 문자
 - "-"기호
- 숫자가 5개 이상 30개 이하이다.

검색: 사용자가 입력한 바코드 번호를 검색어로 하여, 유해식품 데이터의 바코드 번호들 중에서 일치하는 항목이 있는지 찾을 때에는

- 검색어인 사용자가 입력한 바코드 번호에서 숫자를 제외한 문자(공백 문자와 "-"기호)들을 전부 없앤 상태로(검색 대상인 유해식품 데이터의 바코드 번호는 이미 숫자만으로 이루어진 상태이다.)
- 검색어인 사용자가 입력한 바코드 번호 문자열과 검색 대상인 유해식품 데이터의 바코드 번호 문자열이 완벽히 일치하면 매치된 것으로 간주한다. (부분 문자열인 경우는 매치되지 않은 것이다.)

5.4. 사용자 정보

5.4.1. ID

문법 형식: 문법 형식에 맞는 ID는 다음 항목들을 모두 만족하는 문자열이다.

- 다음의 항목들 이외에 다른 형태의 문자를 포함할 수 없다.
 - 영문자 (소문자)
 - 숫자
- 영문자와 숫자는 각각 1개 이상을 포함하고 있어야 한다.
- 4자이상 12자 이하로 설정해야 한다.

동치 비교: 두 ID 간의 동치 비교를 할 때에는, 두 문자열 전체가 서로 완전히 일치해야만 같은 ID로 간주한다.

5.4.2. PW

문법 형식: 문법 형식에 맞는 PW는 다음 항목들을 모두 만족하는 문자열이다.

- 다음의 항목들 이외에 다른 형태의 문자를 포함할 수 없다.
 - 영문자 (대문자, 소문자)
 - 숫자
- 영문자와 숫자는 각각 1개 이상을 포함하고 있어야 한다.
- 4자이상 12자 이하로 설정해야 한다.

동치 비교: 두 PW 간의 동치 비교를 할 때에는, 두 문자열 전체가(영문자 대소문자까지 포함해서) 서로 완전히 일치해야만 같은 PW로 간주한다.

5.4.3 성별

문법 형식: 문법 형식에 맞는 성별은 다음 항목들을 모두 만족하는 문자열이다.

- 다음의 항목들 이외에 다른 형태의 문자열은 오류로 처리한다.
 - 남자
 - 남
 - 여
 - 녀
 - 여자

의미 규칙: 남, 남자는 남자로 여, 녀, 여자는 여자로 해석한다.

5.4.4 나이

문법 형식: 문법적으로 올바른 나이는 숫자만으로 이루어진 문자열이다.

의미 규칙: 1 이상 200 이하의 정수이다.

5.4.5 키

문법 형식: 문법적으로 올바른 키는 숫자만으로 이루어진 문자열이다.

의미 규칙: 1 이상 250 미만의 정수이다.

5.4.6 몸무게

문법 형식: 문법적으로 올바른 키는 숫자만으로 이루어진 문자열이다.

의미 규칙: 1 이상 200 미만의 정수이다.

5.5. 연(년도)

본 문서에서 YYYY 형식이라 한 내용은 본 항목에 관한 것이다.

문법 형식: 문법적으로 올바른 연(년도)은 4 자리의 숫자로 이루어진 문자열이다.

의미 규칙: 1970 이상 2037 이하의 정수이다.

5.6. 월

본 문서에서 MM 형식이라 한 내용은 본 항목에 관한 것이다.

문법 형식: 문법적으로 올바른 월은 2 자리의 숫자로 이루어진 문자열이다. 자릿수를 맞추기 위해 1 이상 9 이하의 숫자는 앞에 0 을 붙여서 01 ~ 09 로 입력해야 한다.

의미 규칙: 1 이상 12 이하의 정수이다.

5.7. 일

본 문서에서 DD 형식이라 한 내용은 본 항목에 관한 것이다.

문법 형식: 문법적으로 올바른 일은 2 자리의 숫자만으로 이루어진 문자열이다. 자릿수를 맞추기 위해 1 이상 9 이하의 숫자는 앞에 0 을 붙여서 01 ~ 09 로 입력해야 한다.

의미 규칙: 앞서 입력된 연(년도)과 월에 따라 그레고리력을 만족하는 날짜가 되어야 한다.

6. 데이터 파일

본 프로그램에서는 6종류의 데이터 파일을 사용하며, 각 파일 종류 별로 상세한 내용은 다음 항목에서 설명한다. 모든 데이터 파일은 txt확장자를 가지는 텍스트 파일이며, 인코딩 형식은 UTF-8 이다.

본 프로그램에서 사용되는 모든 데이터 파일은 프로그램 배포 시 제공 또는 프로그램 실행 중 생성된다. 본 프로그램의 특성 상, 사용자가 입력한 값을 데이터 파일에 저장하기 전 다른 데이터 파일의 내용과 비교해서 올바른 값인지 확인하는 절차가 있는 요소들이 많기 때문에, 사용자가 외부 편집기를 이용해서 데이터 파일을 생성/수정하는 것은 허용되지 않는다. 삭제의 경우 프로그램이 실행 중이지 않은 상태로 다음의 경우만 허용된다.

- 삭제할 사용자 ID를 갖는 사용자 폴더를 통째로 삭제. 즉, 삭제할 사용자 ID를 갖는 사용자 폴더 안에 있는 파일 또는 폴더들만 지우는 것은 허용되지 않고, 프로그램 실행 파일이 있는 폴더의 User 폴더 안에 위치한 사용자 ID를 이름으로 하는 사용자 폴더를 통째로 삭제하는 경우만 허용
- 삭제하고 싶은 날짜의 식품 섭취 일지 파일 또는 섭취 제품명 일지 파일을 삭제
- API서버와의 통신이 원활한 상태일 경우에 식품영양정보 파일 또는 유해식품정보 파일의 삭제, 단 이 경우 정상 작동을 보장하지는 않는다. (API서버에서 정보를 받아오는 중간에 서버측 문제 발생 가능)

6.1. 식품영양정보 파일

프로그램 배포 시 함께 배포되는 파일이며, 식품영양정보 데이터가 저장되어 있는 텍스트 파일이다. 프로그램 주 실행 파일과 같은 폴더에 '식품영양정보.txt'라는 이름으로 있다.

파일에 저장되어 있는 데이터의 형식은 다음과 같다.

- 첫 줄은 식품영양정보 파일 안의 전체 데이터 개수가 저장된다.
- 하나의 식품 데이터의 세부 정보는 ","(콤마)를 사용해 구분한다.
- 각각의 식품 데이터 간의 구분은 줄 바꿈을 사용한다.
- 식품 데이터의 세부 정보 데이터가 존재하지 않을 때 아무것도 없는 상태로 저장된다.
- 가장 처음에는 식품 인덱스 번호 두번째는 식품명, 그 뒤부터는 1회제공 기준 열량 또는 영양분함량이 미리 약속된 순서에 따라 제시된다.
- 열량 또는 영양분함량의 약속된 순서는 왼쪽부터 열량(kcal), 탄수화물(g), 단백질(g), 지방(g), 당류(g), 나트륨(mg), 콜레스테롤(mg), 포화지방산(g)이다.

저장되어 있는 파일의 예시는 다음과 같다.

```
29922
1,펭불고기,500,368.8,39.7,33.5,8.5,16.9,1264.31,106.18,1.9
2,닭갈비,400,595.61,44.9,45.9,25.8,21.2,1535.83,193.4,6,0.2
```

프로그램을 시작할 때 항상 이 파일이 있는지 확인한다. 만약 없다면 '식품의약품안전처 데이터 활용 서비스'에서 제공하는 '식품영양성분db' 공공데이터를 API를 이용해 JSON 형태로 받아온 뒤 위에서 서술한 형태에 맞추어 파일로 저장한다. 만약 이 과정이 실패했다면, 오류의 내용을 출력하고 프로그램은 종료된다.

6.2. 유해식품정보 파일

프로그램 배포 시 함께 배포되는 파일이며, 유해식품정보 데이터가 저장되어 있는 txt파일이다. 프로그램 주 실행 파일과 같은 폴더에 '유해식품정보.txt'라는 이름으로 있다.

파일에 저장되어 있는 데이터의 형식은 다음과 같다.

- 첫 줄은 유해식품정보 파일 안의 전체 데이터 개수가 저장된다.
- 하나의 식품 데이터의 세부 정보는 ","(콤마)를 사용해 구분한다.
- 각각의 식품 데이터의 구분은 줄 바꿈을 사용한다.
- 식품 데이터의 세부 정보 데이터가 존재하지 않을 때 아무것도 없는 상태로 저

장된다.

- 가장 처음에는 식품명, 그 뒤부터는 회수 사유와 업체정보 등 미리 약속된 순서에 따라 제시된다.
- 회수 사유와 업체정보 등 미리 약속된 순서는 왼쪽부터 회수사유, 제조업체명, 업체주소, 전화번호, 바코드번호이다.

저장되어 있는 파일의 예시는 다음과 같다.

519

후레쉬캔디초코맛,리스테리아 모노사이토제네스 검출,(주)주한산업,세종특별자치시 전동면 노장공단길 25-15, 044-863-0887,

짜궁 바닐라맛,자가품질검사결과 리스테리아모노사이토제네스 부적합,(주)주한사업,세종특별자치시 노장공단길 25-15,0448630887,8809089691039

프로그램을 시작할 때 항상 이 파일이 있는지 확인한다. 만약 없다면 '유해식품정보.txt'라는 이름으로 파일을 생성하고, 전체 데이터 개수를 의미하는 첫 줄을 0으로 하여 저장해서 아래 과정에서 업데이트가 이뤄질 수 있도록 한다.

본 파일은 프로그램이 실행될 때 업데이트 과정을 거친다. 이 과정에서 발생할 수 있는 3가지의 상황은 다음과 같다.

- API 호출 중 error발생 시
 - "API 호출error 발생으로 유해식품정보 업데이트를 실패하였습니다." 라는 문장과 함께 기존의 유해식품정보 파일을 사용한다.
 - 다음 줄에는 error 원인을 알려주는 문장을 출력한다. 예시는 다음과 같다.

0	requests.exceptions.ConnectionError	통신 오류입니다. 네트워크를 확인해주세요.
1	INFO-000	정상 처리되었습니다.
2	ERROR-300	필수 값이 누락되어 있습니다. 요청인자를 참고 하십시오.
3	INFO-100	인증키가 유효하지 않습니다. 인증키가 없는 경우, 홈페이지에서 인증키를 신청하십시오.
4	ERROR-301	파일타입 값이 누락 혹은 유효하지 않습니다. 요청인자 중 TYPE을 확인하십시오.
5	ERROR-310	해당하는 서비스를 찾을 수 없습니다. 요청인자 중 SERVICE를 확인하십시오.
6	ERROR-331	요청시작위치 값을 확인하십시오. 요청인자 중 START_INDEX를 확인하십시오.
7	ERROR-332	요청종료위치 값을 확인하십시오. 요청인자 중 END_INDEX값을 확인하십시오.
8	ERROR-334	종료위치보다 시작위치가 더 큼니다. 요청시작조회건수는 정수를 입력하세요.
9	ERROR-336	데이터요청은 한번에 최대 1000건을 넘을 수 없습니다.

10	ERROR-500	서버오류입니다.
11	ERROR-601	SQL 문장 오류입니다.
12	INFO-200	해당하는 데이터가 없습니다.
13	INFO-300	유효 호출건수를 이미 초과하셨습니다.
14	INFO-400	권한이 없습니다. 관리자에게 문의하십시오.

- API 호출로 가져온 데이터의 개수와 기존 유해식품정보 파일의 데이터 개수가 일치 시
 - “변경된 정보가 없기 때문에 업데이트를 하지 않겠습니다.” 라는 문장과 함께 기존의 유해식품정보 파일을 사용한다.
- API 호출로 가져온 데이터의 개수와 기존 유해식품정보 파일의 데이터 개수가 일치하지 않을 시
 - “업데이트를 완료했습니다,” 라는 문장과 함께 기존의 유해식품정보 파일을 API를 호출해 json 형태로 가져온 데이터로 덮어쓰기 한다.

6.3. 사용자 정보 파일

사용자 정보 파일은 사용자 폴더에 위치한 info.txt 라는 이름을 가진 파일로 회원가입 시 사용자가 입력한 정보를 한 줄에 하나씩 성별, 나이, 키, 몸무게 순서대로 저장되어 있는 파일이다.

해당 파일은 사용자가 임의로 생성, 삭제, 수정이 불가능 하다.

6.4. 사용자 PW 파일

사용자 PW 파일은 사용자 폴더에 위치한 password.txt 라는 이름을 가진 파일로 사용자가 회원가입 시 입력한 PW 가 저장되어 있는 파일이다.

해당 파일은 사용자가 임의로 생성, 삭제, 수정이 불가능 하다.

6.5. 식품 섭취 일지 파일

식품 섭취 일지 파일은 사용자 폴더 내에 diary 폴더에 위치한 YYYYMMDD.txt 형식의 이름을 가진 파일로 한 줄에 음식 이름이 하나씩 저장되어 있다.

해당 파일은 사용자가 임의로 생성, 수정이 불가능 하지만, 삭제는 가능하다.

6.6. 섭취 제품명 일지 파일

섭취 제품명 일지 파일은 사용자 폴더 내에 product 폴더에 위치한 YYYYMMDD.txt 형식의 이름을 가진 파일로 한 줄에 섭취 제품명이 하나씩 저장되어 있다.

해당 파일은 사용자가 임의로 생성, 수정이 불가능 하지만, 삭제는 가능하다.

7. 시작 메뉴 프롬프트

식품영양정보와 유해식품정보 데이터 파일의 무결성 검사를 (경고는 있었더라도 오류는 없이) 통과하면, 화면에 다음과 같이 '0. 프로그램 종료', '1. 회원가입', '2. 로그인' 중 하나를 선택할 수 있는 시작 메뉴 프롬프트가 출력된 상태이다.

[시작 메뉴]

0. 프로그램 종료

1. 회원가입

2. 로그인

입력 >

이 프로그램의 사용 흐름 상, 이 프롬프트는 회원가입과 로그인 중 하나를 선택하는 "시작 메뉴"에 해당한다.

문법 형식: 시작 메뉴 프롬프트에서 메뉴를 선택하기 위한 키 입력은 메뉴에 해당하는 번호의 정수만 가능하다.

비정상 결과: 사용자가 입력한 문자가 0이상 2이하의 정수가 아닌 경우, "존재하지 않는 메뉴를 선택했습니다. 다시 입력해주세요." 출력한 후 시작 메뉴 프롬프트를 다시 보여준다.

정상 결과: 사용자가 입력한 문자가 0이상 2이하의 정수인 경우 해당 정수에 해당되는 명령을 수행한다. 0인 경우 프로그램을 종료하고, 1또는 2인 경우에 관해서 각 명령은 아

래 각 소절에서 자세히 부연한다.

7.1 회원가입 프롬프트

ID, PW, 사용자 정보(키, 몸무게, 성별, 나이)를 입력 받는다. 이 과정은 다음 소절에서 자세히 부연한다.

7.1.1. ID 입력 프롬프트

아래와 같이 "ID > "를 출력하여 ID 를 입력 받을 수 있는 ID 입력 프롬프트가 실행된 상태이다.

ID >

문법 형식: ID 의 문법 형식은 5.4.1 절에서 정의한 ID 의 문법 형식을 따른다.

의미 규칙: 중복되는 ID 가 존재하지 않아야 한다. 만약 중복되는 ID 가 있다면, 안내 문구를 출력하고 ID 를 다시 입력 받는다.

비정상 결과: 사용자의 입력이 문법 형식 중 어떤 항목이라도 만족하지 못하는 경우 해당 내용을 사용자에게 알리고 다시 입력 받는다.

- 4자 이하인 경우

- "아이디가 너무 짧습니다. 최소 4자 이상을 입력해주세요." 출력 후 다시 입력 받는다.

아이디가 너무 짧습니다. 최소 4자 이상을 입력해주세요.

사용하실 아이디를 입력해주세요.

입력 >

- 12자 이상인 경우

- "아이디가 너무 길니다. 4자 이상 12자 이하의 아이디를 입력해주세요" 출력 후 다시 입력 받는다.

아이디가 너무 길니다. 4자 이상 12자 이하의 아이디를 입력해주세요.

사용하실 아이디를 입력해주세요.

[입력 >](#)

- 숫자가 없는 경우

- “아이디에 최소 1 개의 숫자가 포함되어야 합니다. 다시 입력해주세요” 출력 후 다시 입력 받는다.

아이디에 최소 1개의 숫자가 포함되어야 합니다. 다시 입력해주세요.

사용하실 아이디를 입력해주세요.

[입력 >](#)

- 소문자가 없는 경우

- “아이디에 최소 1 개의 영문자(소문자)가 포함되어야 합니다. 다시 입력해주세요.” 출력 후 다시 입력 받는다.

아이디에 최소 1개의 영문자(소문자)가 포함되어야 합니다. 다시 입력해주세요

사용하실 아이디를 입력해주세요.

[입력 >](#)

- 대문자가 있는 경우

- “아이디에 대문자는 들어갈 수 없습니다. 소문자만 입력해주세요.” 출력 후 다시 입력 받는다.

아이디에 대문자는 들어갈 수 없습니다. 소문자만 입력해주세요.

사용하실 아이디를 입력해주세요.

[입력 >](#)

- 소문자, 숫자 이외의 문자가 온 경우

- “아이디에 영문자(소문자)와 숫자 외에 다른 문자가 올 수 없습니다. 다시 입력해주세요” 출력 후 다시 입력 받는다.

아이디에 영문자(소문자)와 숫자 외에 다른 문자가 올 수 없습니다. 다시 입력해주세요.

사용하실 아이디를 입력해주세요.

입력 >

정상 결과: 프로그램 실행파일이 존재하는 위치에 있는 "Users" 폴더에 사용자 ID 를 이름으로 하는 폴더를 생성하고, 그 폴더 안에 사용자 폴더에 일지를 저장하는 diary 폴더, 섭취한 제품을 저장하는 product 폴더를 생성한다. 이 후, PW 입력 프롬프트로 이동한다.

7.1.2. PW 입력 프롬프트

아래와 같이 "PW > "를 출력하며 PW 를 입력 받을 수 있는 PW 입력 프롬프트가 실행된 상태이다.

PW >

문법 형식 : PW 의 문법 형식은 5.4.2 절에서 정의한 PW 의 문법 형식을 따른다.

비정상 결과: 사용자의 입력이 문법 형식 중 어떤 항목이라도 만족하지 못하는 경우 해당 내용을 사용자에게 알리고 다시 입력 받는다.

- 4자 이하인 경우
 - "비밀번호가 너무 짧습니다. 최소 4자 이상을 입력해주세요." 출력 후 다시 입력 받는다.

비밀번호가 너무 짧습니다. 최소 4자 이상을 입력해주세요.

사용하실 비밀번호를 입력해주세요

입력 >

- 12자 이상인 경우
 - "비밀번호가 너무 길니다. 4자 이상 12자 이하의 비밀번호를 입력해주세요." 출력 후 다시 입력 받는다.

비밀번호가 너무 겁니다. 4자 이상 12자 이하의 비밀번호를 입력해주세요.

사용하실 비밀번호를 입력해주세요

입력 >

- 영문자가 없는 경우

- “비밀번호에 최소 1개의 영문자가 포함되어야 합니다. 다시 입력해주세요” 출력 후 다시 입력 받는다.

비밀번호에 최소 1개의 영문자가 포함되어야 합니다. 다시 입력해주세요.

사용하실 비밀번호를 입력해주세요

입력 >

- 숫자가 없는 경우

- “비밀번호 보안이 취약합니다. 최소 1개의 숫자를 혼합해서 사용해야 합니다.” 출력 후 다시 입력 받는다.

비밀번호에 최소 1개의 숫자가 포함되어야 합니다. 다시 입력해주세요.

사용하실 비밀번호를 입력해주세요

입력 >

- 영문자, 숫자 이외에 문자가 온 경우

- “비밀번호에 영문자와 숫자 외에 다른 문자가 올 수 없습니다. 다시 입력해주세요” 출력 후 다시 입력 받는다.

비밀번호에 영문자와 숫자 외에 다른 문자가 올 수 없습니다. 다시 입력해주세요.

사용하실 비밀번호를 입력해주세요

입력 >

정상 결과: 사용자의 입력이 모든 문법 형식을 만족하는 경우, 사용자 폴더에 password.txt 라는 이름의 파일을 생성 후 사용자의 입력을 저장한다.

이후, 회원정보 입력 프롬프트로 이동한다.

7.1.3. 회원정보 입력 프롬프트

회원정보 입력 프롬프트에서는 성별, 나이, 키, 몸무게 순서대로 입력 받는다. 이 과정은 다음 소절에서 자세히 부연한다.

7.1.3.1. 성별 입력 프롬프트

아래와 같이 "성별 > "을 출력하며 성별을 입력 받을 수 있는 성별 입력 프롬프트가 실행된 상태이다.

성별 >

문법 형식 : 성별의 문법 형식은 5.4.3 절에서 정의한 성별의 문법 형식을 따른다.

비정상 결과: 사용자의 입력이 문법 형식 중 어떤 항목이라도 만족하지 못하는 경우 "잘못된 입력입니다. 다시 입력해주세요" 출력 후 다시 입력 받는다.

잘못된 입력입니다. 다시 입력해주세요.

성별 >

정상 결과: 사용자의 입력이 모든 문법 형식을 만족하는 경우, 사용자 폴더에 info.txt 라는 이름의 파일을 생성 후 사용자의 입력을 저장한다.

이후, 나이 입력 프롬프트로 이동한다.

7.1.3.2. 나이 입력 프롬프트

아래와 같이 "나이 > "을 출력하며 성별을 입력 받을 수 있는 나이 입력 프롬프트가 실행된 상태이다.

나이 >

문법 형식 : 나이의 문법 형식은 5.4.4 절에서 정의한 나이의 문법 형식을 따른다.

비정상 결과:

- 숫자 이외에 문자를 입력한 경우

- “나이는 숫자만 입력 가능합니다. 다시 입력해주세요” 출력 후 다시 입력 받는다.

나이는 숫자만 입력 가능합니다. 다시 입력해주세요.

나이 >

- 0 이상 200 이하의 숫자를 입력하지 않은 경우

- “나이는 0 이상 200 이하의 정수만 입력 가능합니다. 다시 입력해주세요” 출력 후 다시 입력 받는다.

나이는 0이상 200이하의 정수만 입력 가능합니다. 다시 입력해주세요

나이 >

정상 결과: 사용자의 입력이 모든 문법 형식을 만족하는 경우, 사용자 폴더에 존재하는 info.txt 에 사용자의 입력을 저장한다. 이후, 키 입력 프롬프트로 이동한다.

7.1.3.3 키 입력 프롬프트

아래와 같이 “키 > ”을 출력하며 키를 입력 받을 수 있는 키 입력 프롬프트가 실행된 상태이다.

키 >

문법 형식 : 키의 문법 형식은 5.4.5 절에서 정의한 키의 문법 형식을 따른다.

비정상 결과

- 숫자 이외에 문자를 입력한 경우

- “키는 숫자만 입력 가능합니다. 다시 입력해주세요” 출력 후 다시 입력 받는다.

키는 숫자만 입력 가능합니다. 다시 입력해주세요.

키 >

- 1 이상 250 미만의 숫자를 입력하지 않은 경우

- “키는 1 이상 250 미만의 정수만 입력 가능합니다. 다시 입력해주세요” 출력 후 다시 입력 받는다.

키는 1이상 250미만의 정수만 입력 가능합니다. 다시 입력해주세요.

키 >

정상 결과: 사용자의 입력이 모든 문법 형식을 만족하는 경우, 사용자 폴더에 존재하는 info.txt 에 사용자의 입력을 저장한다.

이후, 몸무게 입력 프롬프트로 이동한다.

7.1.3.4 몸무게 입력 프롬프트

아래와 같이 "몸무게 > "을 출력하며 몸무게를 입력 받을 수 있는 몸무게 입력 프롬프트가 실행된 상태이다.

몸무게 >

문법 형식 : 몸무게의 문법 형식은 5.4.6 절에서 정의한 몸무게의 문법 형식을 따른다.

비정상 결과:

- 숫자 이외에 문자를 입력한 경우
 - "몸무게는 숫자만 입력 가능합니다. 다시 입력해주세요" 출력 후 다시 입력 받는다.

몸무게는 숫자만 입력 가능합니다. 다시 입력해주세요.

몸무게 >

- 1 이상 200 미만의 숫자를 입력하지 않은 경우
 - "몸무게는 1 이상 200 미만의 정수만 입력 가능합니다. 다시 입력해주세요" 출력 후 다시 입력 받는다.

몸무게는 1이상 200미만의 정수만 입력 가능합니다. 다시 입력해주세요.

몸무게 >

정상 결과: 사용자의 입력이 모든 문법 형식을 만족하는 경우, 사용자 폴더에 존재하는 info.txt 에 사용자의 입력을 저장한다. 이후 시작 메뉴 프롬프트로 이동한다.

7.2 로그인 프롬프트

ID, PW 를 사용자로부터 입력 받는다. 이 과정은 다음 소절에서 자세히 부연한다.

7.2.1. ID 입력 프롬프트

아래와 같이 "ID > "를 출력하며 ID 를 입력 받을 수 있는 ID 입력 프롬프트가 실행된 상태이다.

ID >

비정상 결과: 해당 프롬프트에서는 존재하지 않는 ID 가 존재할 수 있어도 비정상 결과는 없다.

정상 결과:

- 입력 받은 사용자의 값과 일치하는 사용자 폴더 이름이 존재할 경우
 - 입력 받은 사용자 값과 일치하는 이름의 사용자 폴더가 존재할 경우 PW 입력 프롬프트로 넘어간다.
- 존재하지 않는 ID 일 경우
 - "존재하지 않는 아이디입니다. 시작 메뉴로 돌아갑니다" 출력 후 다시 시작 메뉴로 돌아간다.

존재하지 않는 아이디입니다. 시작 메뉴로 돌아갑니다.

[시작 메뉴]

0. 프로그램 종료

1. 회원가입

2. 로그인

입력 >

7.2.2. PW 입력 프롬프트

아래와 같이 "PW >"를 출력하며 PW 를 입력 받을 수 있는 PW 입력 프롬프트가 실행된 상태이다.

PW >

비정상 결과: 해당 프롬프트에서는 PW 가 다를 수 있어도 비정상 결과는 없다.

정상 결과:

- PW 일치: 사용자 폴더에 존재하는 password.txt 파일의 내용과 사용자의 입력이 같은 경우 메인 메뉴인 주 프롬프트로 이동한다.
- PW 불일치: 사용자 폴더에 존재하는 password.txt 파일의 내용과 사용자의 입력이 다를 경우 "비밀번호가 다릅니다. 다시 시도해주세요." 출력 후 시작 메뉴 프롬프트로 이동한다.

비밀번호가 다릅니다. 다시 시도해주세요.

[시작 메뉴]

0. 프로그램 종료

1. 회원가입

2. 로그인

입력 >

8. 주 프롬프트: 메인 메뉴

로그인이 성공하면, 화면에 다음과 같이 메뉴를 선택할 수 있는 주 프롬프트가 출력된 상태이다.

[메인 메뉴]

0. 프로그램 종료

1. 일지 작성

2. 일지 불러오기

3. 일지 분석

4. 유해식품 검색

입력 >

이 프로그램의 사용 흐름 상, 이 프롬프트는 메인 메뉴를 번호로 선택하는 “메인 메뉴”에 해당한다.

문법 형식: 주 프롬프트에서 메뉴를 선택하기 위한 키 입력은 메뉴에 해당하는 번호의 정수만 가능하다.

비정상 결과: 사용자가 입력한 문자가 0이상 4이하의 정수가 아닌 경우, 이에 상응하는 오류 문구로 “잘못 입력하셨습니다.”를 출력한 후 주 프롬프트를 다시 보여준다.

정상 결과: 사용자가 입력한 문자가 0이상 4이하의 정수인 경우 해당 숫자에 해당되는 명령을 수행한다. 이에 관해서 각 명령은 아래 각 소절에서 자세히 부연한다.

8.1. 부 프롬프트1: 일지 작성 명령

메인 메뉴인 주 프롬프트에서 “1”를 입력해서 일지 작성으로 들어오면, 화면에 “오늘 섭취한 식품들을 하나씩 Enter키로 구분하여 입력해 주세요. 입력을 완료하셨으면 빈 문자열인 상태로 한번 더 Enter키를 누르세요.” 라는 안내와 함께 오늘 섭취한 식품명을 입력할 수 있는 부 프롬프트1.1로 이동한다.

8.1.1. 부 프롬프트1.1: 식품명 입력

사용자가 식품명을 입력할 수 있는 상태의 프롬프트이다.

문법 형식: 문법적으로 올바른 키 입력 문자열은 5.1 절의 식품명 문법 형식에 맞는 문자열 또는 빈(길이가 0인) 문자열이다.

의미 규칙: 문법적으로만 맞으면, 추가로 준수해야 할 의미 규칙은 없다. 식품명은 제품명과 다르게, 동일한 날짜에 중복 입력을 허용하는데, 이는 같은 식품을 여러 번 또는 여러 인분을 섭취했을 경우가 있을 수 있기 때문이다. 즉, 만약 하루에 어떤 식품을 1인분씩 2번 먹었거나 또는 2인분을 한번에 먹었다면 해당 식품명을 2번 입력하면 된다.

비정상 결과: 사용자가 입력한 값이 문법 형식에 위배되면, 이에 상응하는 오류 문구로 “입력하신 값이 문법 형식에 맞지 않습니다.”를 출력한 후 다시 식품명을 입력 받는다.

정상 결과: 사용자가 입력한 문자열이 문법 형식에 맞으면, 빈 문자열인지 여부에 따라 동작이 달라진다.

- 빈 문자열인 경우: 식품명 입력이 끝난 것으로 간주하여, 화면에 “식품명 입력이 완료되었습니다. 오늘 섭취한 제품명들을 하나씩 Enter키로 구분하여 입력해 주세요. 입력을 완료하셨으면 빈 문자열인 상태로 한번 더 Enter키를 누르세요.”라는 안내와 함께 다음 단계인 제품명 입력을 위한 부 프롬프트1.2로 이동한다.
- 빈 문자열이 아닌 경우: 사용자가 입력한 문자열을 식품영양정보 데이터의 식품명 항목에서 검색했을 때 정확히 일치하는 문자열이 있으면 별도의 출력 없이 해당 식품명을 식품 섭취 일지 파일에 저장한 뒤 다시 식품명을 입력 받고, 없으면 유사한 식품명을 가진 항목들을 검색해서 그 결과에 대한 리스트를 보여주고 하나를 선택하는 부 프롬프트1.1.1로 이동한다. (와일드카드 문자가 포함된 경우도 유사한 식품명을 가진 항목들을 검색하는 과정에서 처리된다. 그 이유는 식품영양정보 데이터의 식품명들 중에는 와일드카드 문자가 포함된 식품명이 없어서, 와일드카드 문자가 포함된 문자열을 그대로 검색했을 때 정확히 일치하는 문자열이 존재할 수 없기 때문이다.)

8.1.1.1. 부 프롬프트1.1.1: 유사 식품명 선택

식품명을 입력 받는 부 프롬프트1.1에서 사용자가 입력한 문자열을 식품영양정보 데이터의 식품명 항목에서 검색했을 때 정확히 일치하는 문자열이 없을 때 5.1 절의 검색 규칙의 유사한 항목을 찾을 때의 방법을 이용해서 검색된 결과로 만든 유사 식품명 목록에서 사용자가 원하는 식품명 1개를 선택하기 위한 프롬프트이다.

문법 형식: 주 프롬프트에서 메뉴를 선택하기 위한 키 입력은 유사 식품명 목록에 해당하는 번호의 정수만 가능하다.

의미 규칙: 문법적으로만 맞으면, 추가로 준수해야 할 의미 규칙은 없다.

비정상 결과: 사용자가 입력한 문자가 0이상 유사 식품명 개수 이하의 정수가 아닌 경우, 이에 상응하는 오류 문구로 "잘못 입력하셨습니다."를 출력한 후 다시 입력 받는다.

정상 결과: 사용자가 입력한 문자가 0이상 유사 식품명 개수 이하의 정수인 경우 0인지 여부에 따라 동작이 달라진다.

- 0인 경우: 사용자가 유사 식품명 목록에서 아무 것도 선택 안 한 것으로 간주하고, "선택 안함"을 출력한 뒤 다시 식품명을 입력할 수 있도록 부 프롬프트1.1로 이동한다.
- 0이 아닌 경우: 사용자가 입력한 숫자에 해당하는 식품명 문자열을 식품 섭취 일지 파일에 저장한 뒤 화면에 해당 식품명을 출력하고, 다시 식품명을 입력할 수 있도록 부 프롬프트1.1로 이동한다.

아래는 식품명 입력의 다양한 상황들을 보여주는 Mockup이다.

오늘 섭취한 식품들을 하나씩 Enter키로 구분하여 입력해 주세요. 입력을 완료하셨으면 빈 문자열인 상태로 한번 더 Enter키를 누르세요.

> 김치

다음 항목 중에서 원하는 항목을 선택하세요.

0. 선택 안함

1. 배추김치

2. 총각김치

3. 열무김치

> 1

배추김치

> 고등어구이

> ??구이

다음 항목 중에서 원하는 항목을 선택하세요.

0. 선택 안함

1. 갈치구이

2. 더덕구이

> 2

더덕구이

>

식품명 입력이 완료되었습니다.

오늘 섭취한 제품명들을 하나씩 Enter키로 구분하여 입력해 주세요. 입력을 완료하셨으면 빈 문자열인 상태로 한번 더 Enter키를 누르세요.

8.1.2. 부 프롬프트1.2: 제품명 입력

문법 형식: 문법적으로 올바른 키 입력 문자열은 5.2 절의 제품명 문법 형식에 맞는 문자열 또는 빈(길이가 0인) 문자열이다.

의미 규칙: 같은 날에 중복되는 제품명 입력을 허용하지 않는다. 이 규칙은 위배되더라도 프로그램이 멈추거나 진행 경로가 아예 달라지는 "오류"가 아니라, 위배될 경우 간단히 문구 출력과 함께 자동적으로 중복을 제거하고 넘어가는 "경고"이다. 즉, 같은 날에 같은 (정확히 일치하는 문자열의) 제품명이 중복(2번 이상)하여 입력될 경우, 두 번째부터 입력되는 중복된 제품명에 대해서, "제품명은 동일 일에 중복으로 입력할 수 없습니다"라는 경고 문구를 출력하고, 입력된 제품명은 파일에 저장하지 않은 채로 다시 입력 받는다.

비정상 결과: 사용자가 입력한 값이 문법 형식에 위배되면, 이에 상응하는 오류 문구로 "입력하신 값이 문법 형식에 맞지 않습니다."를 출력한 후 다시 제품명을 입력 받는다.

정상 결과: 사용자가 입력한 문자열이 문법 형식에 맞으면, 빈 문자열인지 여부에 따라 동작이 달라진다.

- 빈 문자열인 경우: 제품명 입력이 끝난 것으로 간주하여, 메인 메뉴인 주 프롬프트로 이동한다.
- 빈 문자열이 아닌 경우: 사용자가 입력한 문자열을 제품명 일지 파일에 저장한다.

8.2. 부 프롬프트2: 일지 불러오기 명령

메인 메뉴인 주 프롬프트에서 "2"를 입력해서 "식품 섭취 일지 불러오기"로 들어오면, 화면에 "불러오실 일지의 연도와 월을 입력해주세요." 출력 후 연도와 월을 입력 받을 수 있는 부 프롬프트 2.1 로 이동한다.

8.2.1 부 프롬프트 2.1: 연도(Year)와 월(Month) 입력

사용자가 연도와 월을 입력할 수 있는 상태의 프롬프트이다.

불러오실 일지의 년도와 월을 입력해주세요.

입력 >

문법 형식: 문법적으로 올바른 입력 문자열은 5.5 절, 5.6 절의 연도/월 문법 형식에 맞는 문자열 또는 빈(길이가 0 인) 문자열이다.

의미 규칙: 의미적으로 올바른 입력 문자열은 5.5 절, 5.6 절 절의 연도/월 의미 규칙에 맞는 문자열이다.

비정상 결과: 사용자가 입력한 값이 문법 형식 또는 의미 규칙에 위배되면, 아래와 같이 이에 상응하는 오류 문구를 출력한다.

- 입력 형식의 인자보다 적게 입력했을 경우
 - “입력된 인자의 수가 적습니다 다시 입력해 주세요” 출력 후 다시 입력 받는다.

입력된 인자의 수가 적습니다. 다시 입력해 주세요.

불러오실 일지의 년도와 월을 입력해주세요.

입력 >

- 입력 형식의 인자보다 많이 입력했을 경우
 - “입력된 인자의 수가 많습니다 다시 입력해 주세요” 출력 후 다시 입력 받는다.

입력된 인자의 수가 많습니다. 다시 입력해 주세요.

불러오실 일지의 년도와 월을 입력해주세요.

입력 >

- 숫자를 제외한 다른 문자들을 입력했을 경우
 - “숫자 이외의 문자는 입력할 수 없습니다. 다시 입력해주세요.” 출력 후 다시 입력 받는다.

숫자 이외의 문자는 입력할 수 없습니다. 다시 입력해주세요.

불러오실 일지의 년도와 월을 입력해주세요.

입력 >

- 년도가 1970이상 2037이하가 아닌 경우
 - “해당 연도는 지원하지 않습니다. 다시 입력해주세요” 출력 후 다시 입력 받는다.

해당 연도는 지원하지 않습니다. 다시 입력해주세요.

불러오실 일지의 년도와 월을 입력해주세요.

입력 >

- 월을 1이상 12이하의 숫자가 아닌 수를 입력했을 경우
 - “해당 월은 지원하지 않습니다. 다시 입력해주세요” 출력 후 다시 입력 받는다.

해당 월은 지원하지 않습니다. 다시 입력해주세요.

불러오실 일지의 년도와 월을 입력해주세요.

입력 >

- 연/월 이 올해 연/월보다 빠른 경우
 - “현재 년/월보다 빠른 일지는 존재하지 않습니다. 다시 입력해주세요.” 출력 후 다시 입력 받는다.

현재 년/월보다 빠른 일지는 존재하지 않습니다. 다시 입력해주세요.

불러오실 일지의 년도와 월을 입력해주세요.

입력 >

정상 결과:

- 사용자가 입력한 문자열이 문법 형식 및 의미 규칙에 위배되지 않으며 0 이 아니라면 부 프롬프트 2.2 인 날짜 입력 프롬프트로 이동한다.
- 사용자가 입력한 문자열이 0 인 경우 메인 메뉴 프롬프트로 돌아간다.

8.2.2. 부 프롬프트 2.2: 일(Day) 입력 및 일지 보여주기

사용자가 일(Day)을 입력할 수 있는 상태의 프롬프트이다.

해당 월에 작성한 일지 리스트를 보여주며 화면에 “불러오실 일지의 날짜를 입력해주세요” 출력 후 날짜 입력을 받는다.

20201006

20201007

불러오실 일지의 날짜를 입력해주세요.

입력 >

문법 형식: 문법적으로 올바른 입력 문자열은 5.7 절의 날짜 문법 형식에 맞는 문자열 또는 빈(길이가 0 인) 문자열이다.

의미 규칙: 의미적으로 올바른 입력 문자열은 5.7 절의 날짜 의미 규칙에 맞는 문자열이다.

비정상 결과: 사용자가 입력한 값이 문법 형식 또는 의미 규칙에 위배되면, 아래와 같이 이에 상응하는 오류 문구를 출력한다.

- 숫자 이외의 문자를 입력한 경우
 - "숫자 이외의 문자를 입력할 수 없습니다. 다시 입력해주세요." 출력 후 다시 입력 받는다.

숫자 이외의 문자를 입력할 수 없습니다. 다시 입력해주세요.

불러오실 일지의 날짜를 입력해주세요.

입력 >

- 유효하지 않는 날짜를 입력한 경우
 - "해당 날짜는 유효하지 않는 날짜 입니다. 확인 후 다시 입력해주세요" 출력 후 다시 입력 받는다.

해당 날짜는 유효하지 않는 날짜 입니다. 확인 후 다시 입력해주세요.

20201006

20201007

불러오실 일지의 날짜를 입력해주세요.

입력 >

정상 결과:

- 해당 연/월/일에 해당되는 일지가 존재하지 않는 경우

- “해당 날짜에 일지를 적지 않으셨습니다.”후 날짜를 다시 입력 받는다.

해당 날짜에 일지를 적지 않으셨습니다.

20201006

20201007

불러오실 일지의 날짜를 입력해주세요.

입력 >

- 해당 연/월/일에 해당되는 일지가 존재하는 경우
 - 해당 연도/월/날짜 에 해당하는 일지를 불러와서 화면에 출력해준다.

김치

김치찌개용 참치

- 이후, 부 프롬프트 2.3으로 이동한다.
- 0을 입력한 경우
 - 사용자가 0을 입력한 경우 메인 메뉴 프롬프트로 돌아간다.

8.2.3. 부 프롬프트 2.3: 수행할 동작 입력

사용자가 불러온 일지에 대해 수행할 수 있는 동작에 대한 메뉴를 선택할 수 있는 상태이다. 일지 불러오기 메뉴를 보여준 후 어떤 메뉴를 선택할 지 입력을 받는다.

[일지 불러오기 메뉴]

0. 메인 메뉴로 돌아가기

1. 일지 수정하기

2. 일지 삭제하기

입력 >

문법 형식: 일지 불러오기 메뉴 프롬프트에서 입력 받을 수 있는 키 입력은 메뉴에 해당하는 번호의 정수만 가능하다.

비정상 결과: 사용자가 입력한 문자가 0 이상 2 이하의 정수가 아닌 경우, "0 이상 2 이하의 숫자로 입력해 주세요" 출력한 후 다시 입력 받는다.

0이상 2이하의 숫자로 입력해 주세요.

입력 >

정상 결과: 사용자가 입력한 문자가 0 이상 2 이하의 정수인 경우 해당되는 명령을 수행한다. 각각의 명령은 아래 각 소절에서 자세히 설명한다.

8.2.3.1. 부 프롬프트 2.3.1: 일지 수정하기

일지 불러오기 메뉴에서 "1"을 입력해서 "일지 수정하기"로 들어오면, 화면에 "수정할 음식을 선택해주세요" 출력 후 일지 내의 음식들을 리스트로 출력한다.

수정할 음식을 선택해주세요.

0. 김치

1. 김치찌개용 참치

입력 >

문법 형식: 일지 수정하기에서 입력 받을 수 있는 키 입력은 리스트에 해당되는 번호의 정수만 가능하다.

비정상 결과:

- 입력한 문자가 0 이상 리스트의 개수 미만의 정수가 아닌 경우
 - "존재하지 않는 인덱스를 선택했습니다. 다시 입력해주세요." 출력 후 다시 입력 받는다.

존재하지 않는 인덱스를 선택했습니다. 다시 입력해주세요.

입력 >

- 입력한 문자가 숫자 외에 문자일 경우
 - "숫자 외에 문자를 입력할 수 없습니다. 다시 입력해주세요" 출력 후 다시 입력 받는다.

숫자 외에 문자를 입력할 수 없습니다. 다시 입력해주세요.

입력 >

정상 결과: 사용자가 입력한 문자가 0 이상 리스트의 개수 미만의 정수인 경우 정상 결과로서, 식품명을 입력하는 부 프롬프트 1.1 로 이동한다. 단, 다음 항목에 대해서는 부 프롬프트 1.1 에서와 다르게 동작한다.

- 빈 문자열(길이가 0)을 입력한 경우, 원래의 부 프롬프트 1.1 에서와 달리 부 프롬프트 2.3.1 에서 선택한 수정할 식품명을 삭제한 뒤 ~로 이동한다.
- 정상적으로 식품명 입력이 완료되면, 원래의 부 프롬프트 1.1 에서와 달리 부 프롬프트 2.3.1 에서 선택한 수정할 식품명을 해당 식품명으로 바꾼 뒤 ~로 이동한다. 즉, 원래의 부 프롬프트 1.1 에서처럼 식품명 여러 개를 계속해서 입력 받는 것이 아니라 정상적인 식품명을 1 개만 입력 받고 끝낸다.

8.2.3.3. 부 프롬프트 2.3.2: 일지 삭제하기

일지 삭제하기를 선택한 경우 해당 사용자 폴더의 하위 폴더에 위치한 해당 날짜의 일지 파일을 삭제한 이후 메인 메뉴인 주 프롬프트로 이동한다.

8.3. 부 프롬프트3: 일지 분석 명령

메인 메뉴인 주 프롬프트에서 "3"을 입력해서 일지 영양 분석에 들어오면, 화면에 다음과 같이 일지 분석 메뉴를 선택할 수 있는 프롬프트가 출력된 상태이다.

[일지 분석 메뉴]

0. 메인 메뉴로 이동

1. 영양 정보 분석

2. 유해 식품 검사

입력>

이 프로그램의 사용 흐름 상 이 프롬프트는 식품 섭취 일지의 분석과 관련된 메뉴를 번호로 선택하는 "일지 분석 메뉴"에 해당한다.

문법 형식: 메뉴를 선택하기 위한 키 입력은 메뉴에 해당하는 번호의 숫자만 가능하다.

비정상 결과:

사용자의 입력 값이 0미만 또는 2초과의 숫자인 경우 다음과 같은 문구를 출력한 뒤, 다시 사용자의 입력을 받는다.

0이상 2이하의 숫자로 입력해주세요.

입력>

사용자의 입력 값이 숫자가 아닌 경우 다음과 같은 문구를 출력한 뒤, 다시 사용자의 입력을 받는다.

숫자만 입력 가능합니다.

입력>

정상 결과:

사용자가 입력한 문자가 0이상 2이하의 숫자인 경우 해당 정수에 해당되는 명령을 수행한다. '0'인 경우 상위 메뉴인 메인 메뉴인 주 프롬프트로 이동하고, '1'인 경우 부 프롬프트 3.1로 이동하고, '2'인 경우 부 프롬프트 3.2로 이동한다.

8.3.1. 부 프롬프트3.1: 영양정보 분석

부 프롬프트3에서 "1"을 입력해서 영양 정보 분석 메뉴에 들어오면, 화면에 다음과 같이 메뉴를 선택할 수 있는 프롬프트가 출력된 상태이다.

[영양 정보 분석 메뉴]

0.일지 분석으로 이동

1.오늘 일지 분석

2.날짜 지정해서 분석

입력>

이 프로그램의 사용 흐름 상 이 프롬프트는 영양 정보 분석과 관련된 메뉴를 번호로 선택하는 "영양 정보 분석 메뉴"에 해당한다.

문법 형식: 메뉴를 선택하기 위한 키 입력은 메뉴에 해당하는 번호의 숫자만 가능하다.

의미 규칙: 문법 형식에 맞으면, 추가로 꼭 준수해야 할 의미 규칙은 없다.

비정상 결과:

사용자의 입력 값이 0미만 또는 2초과의 숫자인 경우 다음과 같은 문구를 출력한 뒤, 다시 사용자의 입력을 받는다.

0이상 2이하의 숫자로 입력해주세요.

입력>

사용자의 입력 값이 숫자가 아닌 경우 다음과 같은 문구를 출력한 뒤, 다시 사용자의 입력을 받는다.

숫자만 입력 가능합니다.

입력>

정상 결과:

사용자가 입력한 문자가 0이상 2이하의 정수인 경우 해당 숫자에 해당되는 명령을 수행한다.

- 숫자'0'인 경우 상위 메뉴인 부 프롬프트 3으로 이동한다.
- 숫자'1'인 경우 오늘 날짜에 해당하는 식품 섭취 일지가 있을 경우 영양 정보 분석 결과를 출력하고 메인 메뉴인 주 프롬프트로 이동한다. 만약 오늘 날짜에 해당하는 식품 섭취 일지가 없을 경우 일지를 작성할 수 있는 부 프롬프트 1로 이동한다.
- 출력하는 영양 정보 분석 결과는 다음과 같다.
"[YYYYMMDD 영양 정보 분석]"(YYYYMMDD는 해당 날짜)를 출력한 뒤, 그 다음 줄부터 섭취 영양소를 출력한다. 해당 날짜에 입력된 식품명을 식품영양정보 데이터베이스를 활용하여, 해당 일에 섭취한 열량, 탄수화물, 단백질, 지방, 당류, 나트륨, 콜레스테롤, 포화지방산, 트랜스지방의 양을 출력한 뒤, 사용자가 회원가입 시 입력한 정보를 바탕으로 계산된 표준 섭취 열량과 해당 일에 사용자가 섭취한 열량을 출력한다. 그 다음 줄부터 1줄에 1개의 영양소씩 권장 섭취량과 비교했을 때 해당 일의 섭취량이 어떠한지 출력한다.

[YYYYMMDD 영양 정보 분석]

섭취 영양소

열량: 00kcal 탄수화물: 00g 단백질: 00g 지방: 00g 당류: 00g 나트륨:00mg 콜레스테롤: 00mg 포화지방산: 00g

표준 섭취 열량: 00kcal 사용자가 섭취한 열량: 00kcal

열량 섭취량이 많습니다.

탄수화물의 섭취량이 많습니다.

단백질의 섭취량이 적습니다.

지방의 섭취량이 많습니다.

콜레스테롤의 섭취량이 많습니다.
포화지방산의 섭취량이 많습니다.

- 숫자'2'인 경우 부 프롬프트 3.1.1로 이동한다.

8.3.1.1. 부 프롬프트3.1.1: 날짜 지정해서 분석

부 프롬프트 3.1에서 '2'를 입력할 경우, 사용자가 영양 정보 분석을 받고 싶은 일지의 날짜 범위를 입력 받는 프롬프트가 다음과 같이 출력된 상태이다.

[영양 정보 분석 날짜 입력]
-날짜의 기본 형식은 YYYYMMDD인 8자리 정수로 한다.
-날짜 범위로 입력은 "A~B"로 한다(A와 B는 YYYYMMDD형식의 8자리 정수, A<=B).
-날짜를 여러 개 입력 받을 때는 ','로 구분한다.
날짜를 세 개 입력 받는 경우의 예는 "A,B,C"로 한다.(A, B와 C는 YYYYMMDD형식의 8자리 정수)
입력>

문법 형식: 문법적으로 올바른 입력은 다음 항목을 모두 만족해야 한다.

- 각 '날짜'는 5.5절의 연(년도), 5.6절의 월, 5.7절의 일이 순서대로 나열되어 있는 YYYYMMDD형식의 문자열이다.
- 두 개의 '날짜' 사이에 '~'기호를 넣을 수 있고 이렇게 만들어진 A~B(A, B는 '날짜')를 '날짜 범위'라 한다.
- 반드시 1개 이상의 '날짜' 또는 '날짜 범위'를 입력해야 한다.
- 2개 이상의 '날짜' 또는 '날짜 범위'를 입력할 경우 ','기호로 구분한다.

의미 규칙: 날짜 범위를 의미하는 기호인 '~'를 포함한 날짜입력(예: A~B)이 들어올 경우 A가 B보다 더 이전의 날짜이거나 A와 B가 같은 날짜일 경우만 가능하다.

해석:

'A~B'(A와 B는 YYYYMMDD형식의 8자리 정수)가 입력으로 들어올 경우 '~'양옆의 A에서 부터 B까지의 A와 B를 포함한 날짜들을 의미한다.

'A,B' (A, B는 날짜이거나 '~'기호를 포함한 날짜 범위)가 입력으로 들어올 경우 ','로 구분되는 A, B에 해당하는 날짜 또는 날짜 범위를 의미한다.

중복되는 날짜가 입력된 경우 중복을 제거하여 해당 날짜는 1번만 입력된 것으로 간주한다.

비정상 결과:

문법 형식 또는 의미 규칙에 위배될 경우 다음과 같은 문구를 출력한 뒤, 다시 사용자의 입력을 받는다.

올바른 입력 형식이 아닙니다.
입력>

정상 결과:

입력 받은 날짜 혹은 날짜 범위에 해당하는 영양 성분 분석 결과를 1일 단위로 날짜의 오름차순으로 출력한다. 해당 날짜의 섭취 일지가 있을 경우 "[YYYYMMDD 영양 정보 분석]"(YYYYMMDD는 해당 날짜)를 출력한 뒤, 그 다음 줄부터 섭취 영양소를 출력한다. 해당 날짜에 입력된 식품명을 식품영양정보 데이터를 활용하여, 해당 일에 섭취한 열량, 탄수화물, 단백질, 지방, 당류, 나트륨, 콜레스테롤, 포화지방산, 트랜스지방의 양을 출력한 뒤, 사용자가 회원가입 시 입력한 정보를 바탕으로 계산된 표준 섭취 열량과 해당 일에 사용자가 섭취한 열량을 출력한다. 그 다음 줄부터 1줄에 1개의 영양소씩 권장 섭취량과 비교했을 때 해당 일의 섭취량이 어떠한지 출력한다. 마지막으로 빈 줄을 출력하여 다음 날짜와 구분할 수 있도록 한다.

해당 날짜의 섭취 일지가 없을 경우 "YYYYMMDD에 해당하는 일지가 없습니다." (해당 날짜를 YYYYMMDD형태로 출력)를 출력한다. 그 다음 빈 줄을 출력하여 다음 날짜와 구분할 수 있도록 한다.

아래는 날짜 범위로 20201007~20201010을 입력하였을 때의 예이다.

20201007, 20201010은 해당 식품 섭취 일지가 있을 경우로 섭취 영양소에 대한 정보를 출력해 준다. 20201008, 20201009는 해당 식품 섭취 일지가 없을 경우로 "(해당 날짜)YYYYMMDD에 해당하는 일지가 없습니다."를 출력한다. 각 날짜에 해당하는 출력이 끝난 뒤에는 빈 줄을 넣어 날짜 간의 출력에 구분이 가도록 한다.

[20201007 영양 정보 분석]

섭취 영양소

열량: 1415.0kcal 탄수화물: 2336.0g 단백질: 152.5g 지방: 88.0g 당류: 63.5g

나트륨: 38.9mg 콜레스테롤: 5521.9mg 포화지방산: 78.8mg 트랜스지방: 23.3

표준 섭취 열량: 2205.0kcal 사용자가 섭취한 열량: 1415kcal

열량 섭취량이 적습니다. 섭취량을 늘리세요.
탄수화물의 섭취량이 많습니다. 섭취량을 줄이세요.
단백질의 섭취량이 적습니다. 섭취량을 늘리세요.
지방의 섭취량이 적습니다. 섭취량을 늘리세요.
당류의 섭취량이 많습니다. 섭취량을 줄이세요.
나트륨의 섭취량이 적당합니다. 이 섭취량을 유지하세요.
콜레스테롤의 섭취량이 많습니다. 섭취량을 줄이세요.
포화지방산의 섭취량이 많습니다. 섭취량을 줄이세요.
트랜스지방의 섭취량이 많습니다. 섭취량을 줄이세요.

20201008 에 해당하는 일지가 없습니다.

20201009 에 해당하는 일지가 없습니다.

[20201010 영양 정보 분석]

섭취 영양소

열량: 2060.0kcal 탄수화물: 2947.6g 단백질: 296.7g 지방: 107.3g 당류: 82.9g
나트륨: 34.7mg 콜레스테롤: 7653.6mg 포화지방산: 212.9mg 트랜스지방: 18.4
표준 섭취 열량: 2205.0kcal 사용자가 섭취한 열량: 2060kcal

열량 섭취량이 적습니다. 섭취량을 늘리세요.
탄수화물의 섭취량이 많습니다. 섭취량을 줄이세요.
단백질의 섭취량이 적습니다. 섭취량을 늘리세요.
지방의 섭취량이 적습니다. 섭취량을 늘리세요.
당류의 섭취량이 많습니다. 섭취량을 줄이세요.
나트륨의 섭취량이 적당합니다. 이 섭취량을 유지하세요.
콜레스테롤의 섭취량이 많습니다. 섭취량을 줄이세요.
포화지방산의 섭취량이 많습니다. 섭취량을 줄이세요.
트랜스지방의 섭취량이 많습니다. 섭취량을 줄이세요.

모든 날짜에 대한 출력이 완료되었으면, 상위 메뉴인 부 프롬프트 3.1로 이동한다.

8.3.2. 부 프롬프트3.2: 유해식품 검사

식품 섭취 일지 분석 인 부 프롬프트3에서 "2"를 입력해서 유해 식품 검사를 선택하면, 화면에 다음과 같이 사용자가 입력한 모든 제품명에 대해 유해식품 검사를 수행한 결과가 출력되고, 상세 정보를 확인하고 싶은 항목을 선택할 수 있는 프롬프트가 출력된 상

태이다. 유해식품 검색은 5.2절의 검색 규칙에 따라 수행되며, 그 결과는 "0. 상위 메뉴로 이동"을 출력한 이후, 사용자가 작성한 식품명 중에 유해식품이 있을 경우 유해식품을 섭취한 날짜를 오름차순으로 정렬한 뒤 '1'로 시작해서 '1'씩 증가하는 인덱스를 붙여서 나열한다.

0. 상위 메뉴로 이동

1. 20201007 콜라겐

2. 20200923 뽕띠 다쥐르

3. 20200925 리얼 피스타치오

(숫자 '0'을 입력할 경우 메인 메뉴의 3. 식품 섭취 일지 분석하기로 이동)

자세히 알고 싶은 유해식품 index 입력

입력 >

문법 형식: 숫자 0 또는 검색된 유해식품 앞에 붙은 index에 해당하는 숫자만 입력 가능하다.

비정상 결과: 문법 형식에 위배될 경우 "잘못 입력하셨습니다."를 출력한 후 다시 입력 받는다.

정상 결과:

문법 형식에 맞게 입력되면 0인지 여부에 따라 아래 동작이 수행된다.

- '0'인 경우 상위 메뉴인 부 프롬프트 3으로 이동한다.
- '0'이 아닌 경우 아래와 같이 제품명과 함께 제품 정보의 각 항목들(섭취 날짜, 회수 정보, 제조 업체명, 전화번호)를 항목의 이름과 함께 출력한다. 한 항목당 한 줄에 출력한 뒤, 부 프롬프트 3.2를 다시 보여준다.

제품명: 콜라겐

섭취날짜: 20201007

회수 정보: 아조루빈 검출 원료 사용

제조 업체 명: (주)웰팜

전화번호 : 0438813312

8.4. 부 프롬프트4: 유해식품 검색 명령

메인 메뉴인 주 프롬프트에서 "4"를 입력해서 유해식품 검색 메뉴로 들어오면, 화면에

다음과 같이 메뉴를 선택할 수 있는 프롬프트가 출력된 상태이다.

[유해식품 검색 메뉴]
0. 메인 메뉴로 이동
1. 제품명으로 검색
2. 바코드 번호로 검색

이 프로그램의 사용 흐름 상, 이 프롬프트는 유해식품 검색과 관련된 메뉴를 번호로 선택하는 “유해식품 검색 명령 메뉴”에 해당한다.

문법 형식: 메뉴를 선택하기 위한 키 입력은 메뉴에 해당하는 번호의 정수만 입력 가능하다.

비정상 결과: 사용자가 입력한 문자가 0이상 2이하의 정수가 아닌 경우, 이에 상응하는 오류 문구로 “잘못 입력하셨습니다.”를 출력한 후 다시 입력 받는다.

정상 결과: 사용자가 입력한 문자가 0이상 2이하의 정수인 경우 해당 정수에 해당되는 명령을 수행한다. 0인 경우 상위 메뉴인 메인 메뉴로 이동하고, 1인 경우 부 프롬프트 4.1로 이동하고, 2인 경우 부 프롬프트 4.2로 이동한다.

8.4.1. 부 프롬프트4.1: 제품명으로 검색

유해식품 검색 명령 메뉴인 부 프롬프트4에서 “1”를 입력해서 제품명으로 검색 메뉴로 들어오면, 화면에 다음과 같이 검색하고 싶은 제품명을 입력할 수 있는 프롬프트가 출력된 상태이다.

제품명을 입력해주세요.
입력 >

문법 형식: 문법적으로 올바른 키 입력 문자열은 5.2 절의 제품명 문법 형식에 맞는 문자열이다. 즉, 완전한 한글 문자가 최소 1개 이상 들어있는 문자열이다.

의미 규칙: 문법적으로만 맞으면, 추가로 준수해야 할 의미 규칙은 없다.

비정상 결과: 사용자가 입력한 값이 문법 형식에 위배되면, 이에 상응하는 오류 문구로 “입력하신 값이 문법 형식에 맞지 않습니다.”를 출력한 후 상위 메뉴인 “유해식품 검색

명령 메뉴"인 부 프롬프트4로 이동한다.

정상 결과: 사용자가 입력한 값이 문법 형식에 맞으면, 5.2절의 검색 규칙에 따라 사용자가 입력한 제품명을 유해식품정보 데이터의 제품명들을 대상으로 검색을 수행한 결과의 리스트를 출력한다. 이때 제품명과 함께 제품 정보의 각 항목들(사유, 업체명, 전화번호, 바코드 번호)을 항목의 이름과 함께 출력한다. 한 항목당 한 줄에 출력하고 한 제품의 정보를 모두 출력한 이후에는 빈 줄을 1줄 출력해서 다음 제품과 구분할 수 있도록 한다. 어떠한 항목에 대한 정보가 없는 경우 해당 항목은 출력하지 않는다. (항목 이름만 출력한 뒤 빈 칸으로 두는 것이 아니라 해당 항목 자체를 출력하지 않는다. 단, 의미상 해당 항목이 없는 것과 마찬가지로 느껴질 수 있는 것이라도 유해식품정보 데이터에 어떠한 내용이 있다면 그 내용을 그대로 출력한다.)

만약 검색 결과를 저장한 리스트가 비어 있어서 사용자에게 출력할 유해식품정보가 없다면, "입력하신 제품명에 대한 검색 결과가 없습니다."를 출력한다.

검색 결과가 있을 때와 없을 때 모두 사용자에게 문구를 출력한 이후, 메인 메뉴인 주 프롬프트로 이동한다.

아래는 "오징어"를 검색했을 경우의 예시로, 제품 정보 출력 이후 메인 메뉴로 이동하여 출력된 내용은 생략되었다. 유해식품정보 데이터에 "구운 조미오징어"의 제조업체 전화번호는 없으며, 바코드 번호는 "미표시제품"이라고 되어 있다. 따라서 위의 정상 결과에서 서술한 바와 같이 전화번호는 출력하지 않으며, 바코드 번호는 "미표시제품"이라고 출력한다.

제품명을 입력해주세요.

입력 > 오징어

제품명 : 백진미오징어채

사유 : 대장균 기준 초과

업체명 : (주)현대푸드

전화번호 : 0317911785

바코드 번호 : 8804989200800

제품명 : 구운 조미오징어

사유 : 유통기한 초과 표시

업체명 : 동해안 황태채

바코드 번호 : 미표시제품

8.4.2. 부 프롬프트4.2: 바코드 번호로 검색

유해식품 검색 명령 메뉴인 부 프롬프트4에서 "2"를 입력해서 바코드 번호로 검색 메뉴로 들어오면, 화면에 다음과 같이 검색하고 싶은 바코드 번호를 입력할 수 있는 프롬프트가 출력된 상태이다.

바코드 번호를 입력하세요.

입력 >

문법 형식: 문법적으로 올바른 키 입력 문자열은 5.3 절의 바코드 번호 문법 형식에 맞는 문자열이다. 즉, 숫자가 최소 5개 이상 들어있는 문자열이다.

의미 규칙: 문법적으로만 맞으면, 추가로 준수해야 할 의미 규칙은 없다.

비정상 결과: 사용자가 입력한 값이 문법 형식에 위배되면, 이에 상응하는 오류 문구로 "입력하신 값이 문법 형식에 맞지 않습니다."를 출력한 후 상위 메뉴인 "유해식품 검색 명령 메뉴"인 부 프롬프트4를 보여준다.

정상 결과: 사용자가 입력한 값이 문법 형식에 맞으면, 5.3절의 검색 규칙에 따라 사용자가 입력한 바코드 번호를 유해식품정보 데이터의 바코드 번호들을 대상으로 검색을 수행한다.

만약 검색된 결과가 없으면, "입력하신 바코드 번호는 유해식품정보에 등록되어 있지 않은 바코드 번호입니다." 를 출력한다.

만약 검색된 결과가 있으면, 해당 바코드 번호를 가진 제품의 제품명과 제품 정보를 출력한다. 이때 제품명과 함께 제품 정보의 각 항목들(사유, 업체명, 전화번호, 바코드 번호)을 항목의 이름과 함께 출력한다. 한 항목당 한 줄에 출력하고 한 제품의 정보를 모두 출력한 이후에는 빈 줄을 1줄 출력해서 다음 제품과 구분할 수 있도록 한다. 어떠한 항목에 대한 정보가 없는 경우 해당 항목은 출력하지 않는다. (항목 이름만 출력한 뒤 빈 칸으로 두는 것이 아니라 해당 항목 자체를 출력하지 않는다. 단, 의미상 해당 항목이 없는 것과 마찬가지로 느껴질 수 있는 것이라도 유해식품정보 데이터에 어떠한 내용이 있다면 그 내용을 그대로 출력한다.)

검색 결과가 있을 때와 없을 때 모두 사용자에게 문구를 출력한 이후, 메인 메뉴인 주 프롬프트로 이동한다.

아래는 바코드 번호를 입력하여 검색 결과가 있을 때의 예시이다. 제품 정보 출력 이후 메인 메뉴로 이동하여 출력된 내용은 생략되었다.

바코드 번호를 입력하세요.

입력 > 8801956000012

제품명 : 명작 발효홍삼

사유 : 자가품질검사 부적합(세균수 초과)

업체명 : (주)케이지앤에프

전화번호 : 0553291919

바코드 번호 : 8801956000012

8.5 종료 명령: "0"번 메뉴

주 프롬프트에서 선택한 메뉴에 따라 실행되는 명령이고, 프로그램을 종료하는 것 외에 다른 동작을 수행하지 않으므로 이 단계에서는 별도의 문법 형식이나 의미 규칙이 없다.

정상 결과: "프로그램을 종료합니다."를 출력하고, 즉시 프로그램을 종료한다.