**최종 보고서**

**헬스케어 다이어트 도움 앱**

**스마트 PT**

**1조**

**팀장 : 17011473 정지영**

**팀원 : 15010986 신현철**

**17011609 유재현**

**17011596 황연주**

목차

[1. 개요 5](#_Toc57322625)

[1.1 개발 배경 5](#_Toc57322626)

[1.2 개발 목표 및 내용 5](#_Toc57322627)

[**1.2.1 기술개발 최종목표** 5](#_Toc57322628)

[**1.2.2 기술개발 세부내용** 5](#_Toc57322629)

[1. Google Fit와 Mi-Fit 연동 후, Google Fit Api를 이용하여 기록된 운동정보를 받아옴 2. 운동 종류(걷기, 달리기)와 운동시 소모된 시간, 거리를 측정하여 저장 5](#_Toc57322630)

[1.3 차별성 7](#_Toc57322631)

[**1.3.2 기존 시장 조사** 7](#_Toc57322632)

[**1.3.2** **차별성** 9](#_Toc57322633)

[1.4 개발환경 및 체계 10](#_Toc57322634)

[**1.4.1 개발 환경** 10](#_Toc57322635)

[**1.4.2** **시스템 구성도** 10](#_Toc57322636)

[1.5 개발 추진 계획 14](#_Toc57322637)

[1.6 기대효과 14](#_Toc57322638)

[2. 설계 사양서 15](#_Toc57322639)

[2.1설계 사양서 산출물 체계 15](#_Toc57322640)

[**2.1.1 산출물 코드 및 ID 부여 방법** 15](#_Toc57322641)

[**2.1.2 산출물 목록** 15](#_Toc57322642)

[2.2 요구사항 분석서 16](#_Toc57322643)

[**2.2.1 요구사항 목록** 16](#_Toc57322644)

[**2.2.2 요구사항 세부명세** 17](#_Toc57322645)

[2.3 유스케이스 다이어그램 25](#_Toc57322646)

[**2.3.1 유스케이스 다이어그램** 25](#_Toc57322647)

[**2.3.2 세부 유스케이스 다이어그램** 26](#_Toc57322648)

[**2.3.3 유스케이스 기술서** 27](#_Toc57322649)

[2.4 클래스 다이어그램 33](#_Toc57322650)

[**2.4.1 설계 클래스도** 33](#_Toc57322651)

[2.5 시퀀스 다이어그램 34](#_Toc57322652)

[**2.5.1 시퀀스 목록** 34](#_Toc57322653)

[**2.5.2 시퀀스 다이어그램** 35](#_Toc57322654)

[개인정보 입력 35](#_Toc57322655)

[섭취한 음식 등록 36](#_Toc57322656)

[IoT 연동 37](#_Toc57322657)

[IoT 데이터 저장 38](#_Toc57322658)

[통계 정보 확인 39](#_Toc57322659)

[추천 운동 조회 40](#_Toc57322660)

[추천 식단 확인 41](#_Toc57322661)

[2.6 데이터베이스 설계도 42](#_Toc57322662)

[2.6 개발 환경 명세서 45](#_Toc57322663)

[**2.6.1 주 개발 프로그램** 45](#_Toc57322664)

[**2.6.2 보조 개발 프로그램** 45](#_Toc57322665)

[**2.6.3 협업 프로그램** 45](#_Toc57322666)

[3. 개발 추진 계획 45](#_Toc57322667)

[4. 개발 내용 46](#_Toc57322668)

[4.1 주요 기능 구현 46](#_Toc57322669)

[섭취한 음식 입력 47](#_Toc57322670)

[섭취한 음식 칼로리 및 영양소 정보 제공 48](#_Toc57322671)

[활동한 운동제공 49](#_Toc57322672)

[추천식단 제공 49](#_Toc57322673)

[추천 운동 제공 50](#_Toc57322674)

[미핏과 구글핏 연동 51](#_Toc57322675)

[섭취 칼로리 통계 제공 52](#_Toc57322676)

[인바디 통계 제공 53](#_Toc57322677)

[5. 테스트 명세서 53](#_Toc57322678)

[5.1 기능테스트 53](#_Toc57322679)

[5.2 성능테스트 55](#_Toc57322680)

[6. 요구사항 대비 시스템 구현 현황 55](#_Toc57322681)

[6.1 제안서 대비 수정 내용 55](#_Toc57322682)

[6.2 요구사항 구현현황 57](#_Toc57322683)

[7. 개발 과정에서의 문제점 57](#_Toc57322684)

[8. 향후계획 58](#_Toc57322685)

# 개요

## 1.1 개발 배경

|  |  |
| --- | --- |
|  |  |
| <그림1> 다이어트 필요성의 여론조사> | <그림2> 다이어트 방법 여론조사 |

<그림1>과 <그림2>에 따르면, 대다수의 현대인들은 다이어트가 필요하다고 생각하며 운동과 식단조절을 기반으로 한다. 그러나 현대 스마트 헬스케어 시장을 보면 정보를 모두 수기로 입력하는 불편함이 따르며 이것이 곧 앱의 사용감을 떨어트려 다이어트 실패를 야기한다. 따라서 운동기록과 체성분을 하드웨어와 연동하여 수기로 입력하는 불편함 없이 자동으로 기록해 한눈에 볼 수 있게 하고, 카메라로 간편하게 식단을 찍어 필요한 영양소를 알 수 있을 뿐 아니라 식단까지 추천해주어 다이어트 성공을 돕는 어플을 만들게 되었다.

## 1.2 개발 목표 및 내용

### **1.2.1 기술개발 최종목표**

평소 시간이 없는 직장인들, 헬스장 비용과 PT비용이 부담되는 학생들, 코로나 19로 인한 사회적 거리두기를 하고 있는 생활 속, 혼자서 아무런 정보 없이 건강관리하기란 힘든 일이다. 이러한 환경 속, 균형 잡힌 식단과 활동 대사량을 자동으로 측정할 수 있다면 간편하게 건강관리를 할 수 있을 것이다. 건강관리하기 힘든 현대인들에게 음식 사진을 찍고, 스마트 워치를 차고 다니는 행동만으로 해당 정보들을 간편하게 확인할 수 있도록 도움을 주는 앱을 개발하자.

### **1.2.2 기술개발 세부내용**

|  |  |
| --- | --- |
| 미밴드 연동 | 1. Google Fit와 Mi-Fit 연동 후, Google Fit Api를 이용하여 기록된 운동정보를 받아옴  2. 운동 종류(걷기, 달리기)와 운동시 소모된 시간, 거리를 측정하여 저장 |
| 미스케일 체중계 연동 | 1. [Bluetooth를 이용하여](#_6.1_제안서_대비) [[1]](#footnote-2) 기록된 사용자의 인바디/체중 정보를 받아옴 2. 체중, 인바디 정보 측정하여 저장 |
| 식단 추천 | 1. 사용자에게 칼로리를 입력받아, 해당 칼로리와 비슷한 식단을 여러개 추천한다. |
| 데이터 관리 | 1. 체중계 - 회원의 현재 인바디 정보 - 누적된 회원의 인바디 정보 비교 2. 스마트 워치 - Daily 운동, 칼로리 정보 제공  - 추천 운동 표시  3. 칼로리  - 섭취한 음식 칼로리 통계를 일/주/월별로 제공 |
| 사진 분석 | 1. TensorFlow 모델 개발 후 TensorFlow Lite 형식으로 변환하여 앱에서 사용할 수 있도록함 2. 인공지능을 통한 음식 판단 및 칼로리, 영양소 제공  3. 하단의 <표1>과 <표2>를 합친 총 60가지의 음식을 구별 가능. |
| 오늘 섭취한 음식 기록 | 1. 아침, 점심, 저녁별로 섭취한 음식을 기록.  2. 상단의 ‘사진 분석’을 이용하거나, 직접  음식이름을 입력하여 음식을 기록.  3. 먹은 양(0.5인분/1인분/2인분) 기록.  4. 섭취한 음식의 칼로리와 탄수화물, 단백질, 지방 정보 제공. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 밥(여러종류) | 배추 | 깻잎 | 멜론 | 복숭아 | 요구르트 |
| 식빵 | 깍두기 | 토마토(방울토마토) | 바나나 | 귤 | 콩자반 |
| 감자 | 고사리 | 단호박 | 수박 | 딸기 |  |
| 고구마 | 표고버섯 | 피망 | 아보카도 | 키위 |  |
| 도토리묵 | 상추 | 망고 | 사과 | 포도 |  |

<표1> 분류할 음식목록 27종 - 체중관리용 음식

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 라면 | 갈비구이 | 김밥 | 떡볶이 | 새우튀김 | 족발 |
| 쫄면 | 갈비탕 | 김치찌개 | 만두 | 소세지볶음 | 찜닭 |
| 햄버거 | 갈치구이 | 깻잎장아찌 | 물회 | 순대 | 해물찜 |
| 단무지 | 감자채볶음 | 꼬막찜 | 배추김치 | 육회 | 피자 |
| 후라이드치킨 | 계란찜 | 된장찌개 | 삼계탕 | 짜장면 | 수정과 |
| 식혜 | 약과 | 한과 |  |  |  |

<표2> 분류할 음식목록 33종 – 자주 먹을만한 음식

## 1.3 차별성

### **1.3.2 기존 시장 조사**

다이어트 신

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
| 장점 | 칼로리 사전을 제공한다 |
| 다이어트 다이어리 작성 |
| 매일매일 체성분 ( 체지방량 / 골격근량 )을 작성할수 있다 |
| 운동 별 소모되는 칼로리 제공 |
| 만보기 기능을 통한 운동량 기록 |
| 단점 | 음식과 운동, 체성분 기록을 수기로 작성해야 함. |
| 만보기를 사용하지 않는 운동은 자동 입력을 지원하지 않음. |

다이닝 노트

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
| 장점 | 아침, 점심, 저녁 등으로 나누어 식단과 운동내용을 기록가능 |
| 기록한 내용을 통계로 확인이 가능 |
| 인터페이스가 간편 |
| 단점 | 먹은 식단과 운동을 사용자가 수기로 입력해야함. |
| 별도의 칼로리 계산은 제공하지 않음. |

다이어트 AI

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
| 장점 | 카메라 사진 촬영으로 칼로리 계산이 가능 |
| 나이, 성별, 체중등을 설정해주면 목표 칼로리 제공 |
| 프리미엄 서비스 이용 시 식사기록 / 갤러리 기능 제공 |
| 단점 | 음식 사진을 찍으면 영양소에 따라 피드백을 제공하지만 식단 제공까지는 하지 않음. |
| 운동정보는 제공해주지 않음 |
| 프리미엄 서비스를 사용해야만 14일 이상의 식단 기록이 가능. |

### **차별성**

SmartPT는 기존의 앱 서비스와 다르게   
첫째, mi-band와 연동하여 운동한 기록을 가져옴으로써 사용자의 운동량을 별도의 입력없이 간편하게 볼 수 있다.  
둘째, 선택적으로 스마트 체중계와 연동을 통해 사용자가 원할 때마다 체성분을 기록하고 변화를 한 눈에 볼 수 있도록 한다.  
셋째, 식단을 텍스트로 입력하지 않고도 카메라로 식단을 찍으면 칼로리와 탄수화물, 지방, 단백질의 함량을 보여주며 이에 따라 식단을 추천해준다.  
넷째, 갤럭시, 애플워치는 기능이 잘 되어있으나 비싼 것이 단점인데, 이보다 저렴한 mi band 와 샤오미 체중계를 선택함으로서 가격 경쟁력을 갖추고 진입장벽은 낮추면서 앱의 시장성을 증가시킨다.

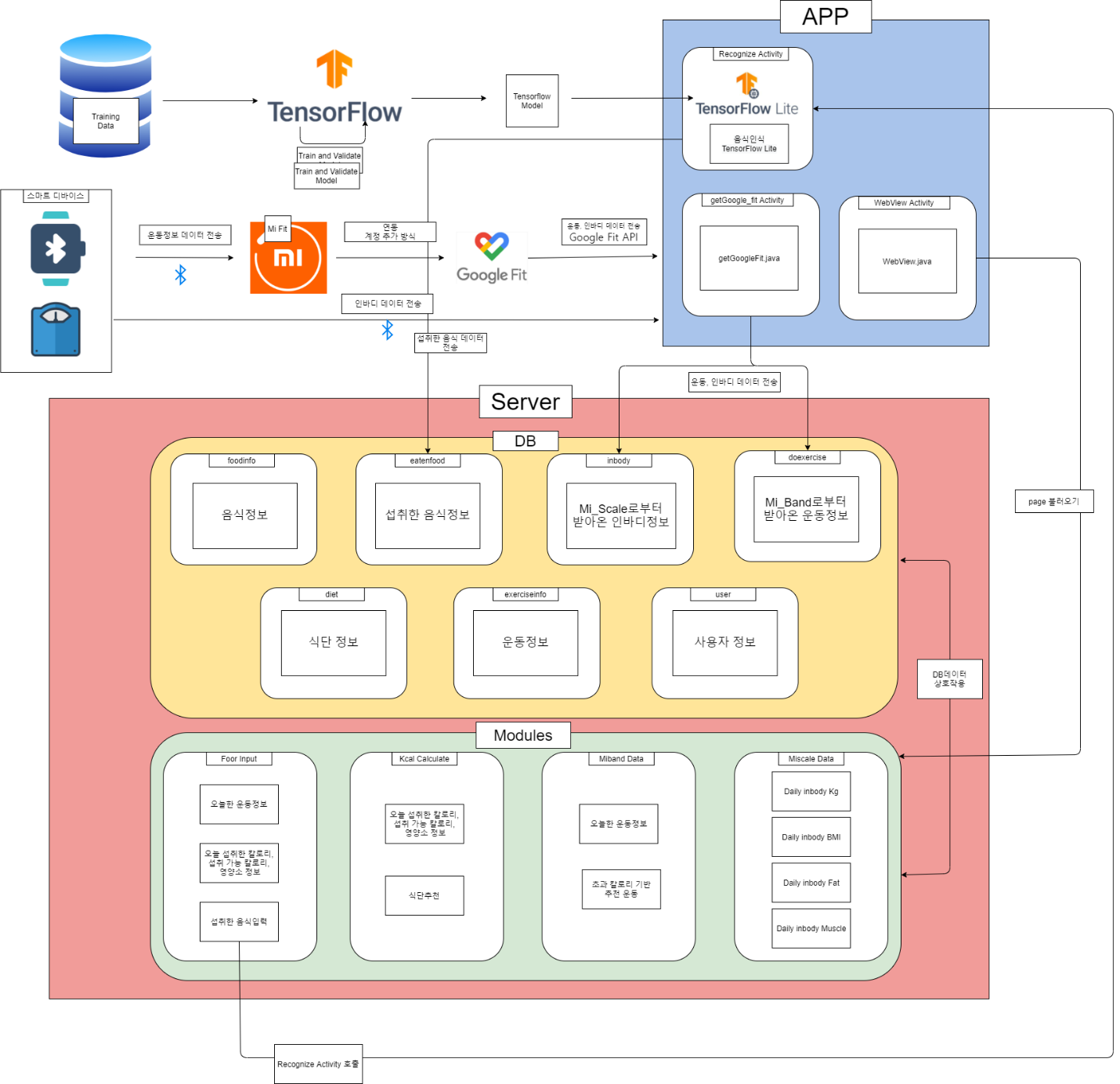
## 1.4 개발환경 및 체계

### **1.4.1 개발 환경**

|  |  |  |
| --- | --- | --- |
| Back\_End | Server Side | Google Cloud Plaform  Ubuntu |
| Camera API | Python |
| Front\_end | Web | Html, PHP, javascript |
| Android | Java |
| Developement Tools | Android Studio, SublimeText3, Eclipse, Xshell6, Vi Editor | |
| Library | Tensorflow, JQuery, Google Fit Api | |

### **시스템 구성도**

**시스템 전체구조**



시스템 구성도는 크게 스마트 기기, APP, Server 로 나뉜다. 스마트 기기로는 운동 정보를 기록하는 미밴드와 인바디 데이터를 기록하는 미스케일이 있다. 스마트 밴드는 미밴드(스마트 밴드)랑 미핏(어플)은 블루투스를 통해 연결되고 이 미핏이랑 구글핏이랑 연동을 시킨 다음, 구글핏 api 를 통해서 데이터를 받아오는 형식으로 구성했다. 스마트 체중계는 미스케일(스마트 체중계)랑 app을 블루투스로 바로 연동하여 데이터를 받아오는 형식으로 구성했다. 인공지능 모델은 텐서플로우에서 모델을 만든 후, 텐서플로우 lite 형식으로 모델을 변환해서 앱에 저장된다.

APP은 크게 음식분석, 스마트 기기 정보 저장, 웹화면을 보여주는 기능으로 나뉜다. 음식분석은 Tensforflow Lite를 통해 음식을 인식하고, DB에 해당 음식 정보를 저장한다. 스마트 기기 정보 저장은 스마트 기기를 통해 전달된 운동, 인바디 정보를 DB에 저장한다.

서버는 크게 DB와 Web page로 나뉜다. DB에는 음식정보, 운동정보, 식단정보, 사용자 정보 등의 앱에게 필요한 정보들이 저장된다. Web page는 앱의 실질적인 화면을 담당한다.

**서브시스템**

|  |  |
| --- | --- |
|  |  |
| <Kcal Calculate>  오늘의 운동정보와 등록한 음식의 칼로리/영양소 정보가 출력되고, 음식을 등록할 수 있는 화면 | <Food input>  몸무게, 섭취한 칼로리, 인바디 정보를 확인할 수 있는 화면 |
|  |  |
| <Miband Data>  오늘 섭취한 칼로리와 앞으로 섭취 가능한 칼로리 정보를 출력하고, 식단을 추천해주는 화면 | <Miscale Data>  미밴드를 통해 기록된 오늘의 운동 정보, 섭취한 칼로리, 섭취 가능한 칼로리정보를 출력하고, 운동을 추천해주는 화면 |
|  |  |
| <스마트 기기 연결>  앱과 스마트 기기를 연결하고 전달된 정보를 DB에 저장 | <DB> |

**역할 분담 내용**

|  |  |  |
| --- | --- | --- |
| 정지영 | 17011473 | 인공지능 모델 개발을 위한 사진 수집  앱과 스마트 밴드, 스마트 체중계 연동 |
| 신현철 | 15010986 | 음식분석 인공지능 모델 개발  카메라 api |
| 유재현 | 17011609 | Gcp내 우분투 서버 구성 및 보수 유지  웹 앱에 사용되는 웹페이지 구현  웹과 앱 연동 |
| 황연주 | 17011596 | 데이터베이스 설계 / 구현 / 관리  웹 앱에 사용되는 웹페이지 구현 |

## 개발 추진 계획



## 1.6 기대효과

삶의 질이 높아지고, 기대 수명이 증가함에 따라 현대인들의 건강에 대한 관심이 높아지고 있다. 이러한 사회에서 개개인의 운동기록과 식습관을 연동시켜 식단과 운동을 추천해주는 ‘스마트 PT앱’의 등장은 많은 사람들의 관심을 끌 수 있을 것으로 보인다.   
 식단관리를 하지 못하는 이유 중 하나는 식단을 짜는 것이 번거롭고 다양성을 확보하기가 힘들다는 것인데, 이 어플을 이용하면 카메라로 찍어서 자신의 식단을 간단히 기록하고, 이를 바탕으로 다양한 식단을 추천하며 통계까지 확인할 수 있으므로 사용자의 건강한 식습관을 이끌어 줄 수 있다.   
 다이어트를 하는 개개인 뿐만 아니라, 헬스장의 회원들이나 기관의 관리를 받는 운동선수들 또한 운동량과 식습관을 중요시 여기기 때문에 기관에서의 이용가능성도 기대해 볼 수 있다. 또한 운동량과 식습관과 같은 단순한 수치뿐만 아니라, 인바디를 통해 현재 몸 상태와 연관 지을 수 있기 때문에 보다 현실적이고 체계적인 관리가 가능하다.   
 단순히 칼로리만 계산해주는 것이 아니라, 하루 권장 영양소 섭취량을 기준으로 식단 추천을 해주기 때문에 병원이나 학교와 같은 공공 기관에서도 사용 가능할 것으로 기대된다.

# 2. 설계 사양서

## 2.1설계 사양서 산출물 체계

### **2.1.1 산출물 코드 및 ID 부여 방법**

산출물 산출물 ID의 기본적인 부여 방법은 다음과 같이 제시한다. 단, 프로젝트에 별도의 산출물 ID 부여 방법이 존재할 경우는 프로젝트에서 정하는 바에 따라 사용할 수 있다.

산출물 코드는 분석 단계에서는 R0, 설계 단계에서는 D0의 형태로 부여한다.

프로젝트 ID: SP

ID 명: SP\_UR\_001, 요구사항 분석서  
ID 명: SP\_UD\_001, 유스케이스 다이어그램  
ID 명: SP\_CD\_001, 클래스 다이어그램  
ID 명: SP\_SD\_001, 시퀀스 다이어그램   
ID 명: SP\_PT\_001, 테스트 명세서

### **2.1.2 산출물 목록**

|  |  |  |
| --- | --- | --- |
| 단계 | 코드 | 산출물 |
| 분석 | UR | 요구사항 분석서 |
| 설계 | UD | 유스케이스 다이어그램 |
|  | CD | 클래스 다이어그램 |
|  | SD | 시퀀스 다이어그램 |
| 테스트 | PT | 테스트 명세서 |

## 2.2 요구사항 분석서

### **2.2.1 요구사항 목록**

|  |  |  |  |
| --- | --- | --- | --- |
| * 회원정보관리 * 회원정보 입력 * 회원 기본칼로리 계산 * 섭취한 음식 입력 | * 초과 칼로리 제시 * 식단추천 * 추천운동제공 * 미핏과 구글핏 연동 | * 운동 데이터 받아오기 * 인바디 데이터 받아오기 * 인바디 통계 제공 | * 섭취한 음식 칼로리   통계 제공   * 섭취한 음식정보 제공 * 운동 통계 제공 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 공통 | | | | |
| 코드 번호 | 분류 | 요구사항명 | 상세내용 | 난이도 |
| SP\_UR\_001 | COMMON | 회원정보관리 | 각 회원마다 고유 ID를 부여하여 회원 별 정보 열람 및 서비스 이용 | 하 |
| SP\_UR\_002 | COMMON | 회원정보입력 | 사용자로부터 사용자의 키, 체중, 나이, 성별, 운동빈도, 다이어트 목표, 체중계 사용여부를 받아온다. | 중 |
| SP\_UR\_003 | COMMON | 회원 목표 칼로리 설정 | 회원이 직접 자신이 원하는 목표칼로리를 입력한다. | 하 |
| SP\_UR\_004 | APP | 섭취한 음식 입력 | 사용자가 카메라로 찍은 음식 사진을 인공지능을 통한 인식 후 칼로리 정보 제공 | 상 |
| SP\_UR\_005 | APP | 초과 칼로리 제시 | 사용자의 섭취 칼로리가 목표 칼로리보다 많은 경우 초과된 칼로리 정보 제시 | 중 |
| SP\_UR\_006 | APP | 식단추천 | 사용자의 현재 일일 섭취상태에 따라 식단추천을 해준다. | 중 |
| SP\_UR\_007 | APP | 추천 운동 제공 | 사용자가 하루동안 섭취한 음식의 칼로리가 목표 칼로리를 넘어선 경우 운동을 추천해준다. | 중 |
| SP\_UR\_008 | APP | 미핏과 구글핏 연동 | Mi-fit application과 Google fitness application을 연동한다 | 중 |
| SP\_UR\_009 | APP | 운동 데이터  받아 오기 | 사용자가 행한 운동 이름, 운동 시간, 그리고 소모한 칼로리를 받아온다. | 상 |
| SP\_UR\_010 | APP | 인 바디 데이터  받아 오기 | 사용자의 인 바디 정보(BMI, 체중, 지방 량, 근육 량)를 받아온다. | 상 |
| SP\_UR\_011 | APP | 인 바디 통계 제공 | 사용자의 인 바디 정보(근육 량, 체지방 량, BMI, 체중)를 동기화를 실행한 날짜 별로 그래프로 시각화 하여 화면을 통해 보여준다. | 중 |
| SP\_UR\_012 | APP | 섭취한 음식 칼로리  통계 제공 | 사용자가 섭취한 음식의 칼로리를 일/주/월 별로 그래프로 시각화 하여 화면을 통해 보여준다. | 중 |
| SP\_UR\_013 | APP | 섭취한 음식 정보 제공 | 하루동안 사용자가 섭취한 음식 정보(칼로리, 탄수화물,  단백질, 지방) 를 아침, 점심, 저녁으로 나누어 텍스트와  그래프 형식으로 보여준다. | 중 |
| SP\_UR\_014 | APP | 운동 통계 제공 | 하루동안 실행한 운동 정보 ( 운동이름, 운동시간, 소모  , 칼로리) 를 화면을 통해 보여준다. | 중 |

### **2.2.2 요구사항 세부명세**

회원정보관리

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | 회원정보관리 | 요구사항번호 | SP\_UR\_001 |
| 요구사항구분 | COMMON | 작성자 | 유재현 |
| 요구사항설명 | 각 회원마다 고유 ID를 부여하여 회원 별 정보 열람 및 서비스 이용 | | |
| 구현 시 고려사항 | DB설계시, 사용자관리에 알맞도록 하여야함 | | |
| 위험요소 | 1. 고유 식별 ID가 rand()함수로 생성되기 때문에 사용자의 고유ID가 겹칠 수 있다. | | |
| 위험요소 해결방안 | 1. 고유 ID를 부여할 때, smartpt [DB] ->user [table] 에 있는 기존 회원들의 고유ID와 비교하여 겹칠 경우, 겹치지 않을 때까지 rand()함수를 이용하여 고유 ID를 생성한 뒤, 부여한다. | | |
| 관련요구사항 | 서버 관리 | | |
| 시나리오 | 1. 각 사용자 어플에 고유 ID를 부여  (첫 접속 시, 고유 ID 부여한다. )  고유 ID 형식 : 9자리의 숫자를 rand()함수를 이용하여 랜덤으로 부여  Ex) user1의 고유ID : 456231235  2. 부여한 고유 ID를 smartpt [DB] ->user [table]에 업데이트 한다.  3. 다음 접속부턴 자신의 고유ID를 통해 식별이 가능함으로 이를 이용하여 자신의 정보 및 서비스를 사용할 수 있도록 한다. | | |

회원정보 입력

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | 회원정보 입력 | 요구사항번호 | SP\_UR\_002 |
| 요구사항구분 | COMMON | 작성자 | 유재현 |
| 요구사항설명 | 사용자는 사용자의 키(INT), 체중(INT), 나이(INT), 성별(INT), 운동빈도(INT), 목표칼로리(INT), 체중계사용여부 (INT) 를 입력한다. | | |
| 구현 시 고려사항 | 각 사용자들마다 개인정보를 관리할 수 있게 한다.  사용자 신상 정보 변경 시 수정할 수 있게 한다. | | |
| 위험요소 | 1. 사용자가 자신의 신상정보를 엉터리로 작성할 수 있다. | | |
| 위험요소 해결방안 | 1-1. 사용자의 입력칸에 limit를 설정하여 엉터리로 작성할 수 없도록 설정.  키 ( 100 ~ 250 까지 입력가능 )  체중 ( 10 ~ 200 까지 입력가능 )  나이 ( 1 ~ 100 까지 입력가능 )  성별 ( radio button으로 입력받음 )  운동빈도 ( radio button으로 입력받음)  목표칼로리 ( 0 ~ 3000 까지 입력가능 )  체중계사용여부 ( radio button으로 입력받음 ) | | |
| 관련요구사항 | 서버 연결 관리 | | |
| 시나리오 | 1. 앱을 통해 사용자가 접속한다.  2. 사용자 정보 칸으로 이동하여 사용자의키(INT), 체중(INT), 나이(INT), 성별(INT), 운동빈도(INT), 목표칼로리(INT), 체중계사용여부 (INT) 를 받아온다.  3. 입력 / 수정된 사용자 정보는 smartpt [DB] ->user [table] 에 자동으로 업데이트 된다. | | |

회원 목표 칼로리 설정

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | 회원 목표 칼로리 설정 | 요구사항번호 | SP\_UR\_003 |
| 요구사항구분 | COMMON | 작성자 | 유재현 |
| 요구사항설명 | 사용자는 직접 자신이 원하는 목표칼로리(INT)를 입력한다. | | |
| 구현 시 고려사항 | 권장칼로리 : 성인여자=354-6.91x연령(세)+PA[9.36x체중(kg)+726x신장(m)]  성인남자=662-9.53x연령(세)+PA[15.91x체중(kg)+539.6x신장(m)]  ※ PA(신체활동계수): 1.0(비활동적), 1.12(저활동적), 1.27(활동적),1.45(매우 활동적) | | |
| 위험요소 | 사용자가 터무니없는 목표칼로리를 설정한다. | | |
| 위험요소 해결방안 | 목표칼로리 설정 시, 임계치(0~3000)를 두어 관리한다. | | |
| 관련요구사항 | 서버 연결관리 | | |
| 시나리오 | 1. 사용자가 원하는 목표칼로리를 입력한다.  2. 입력된 목표칼로리 정보는 smartpt [DB] ->user [table] 에 업데이트 된다. | | |

섭취한 음식 입력

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | 섭취한 음식 입력 | 요구사항번호 | SP\_UR\_004 |
| 요구사항구분 | APP | 작성자 | 신현철 |
| 요구사항설명 | 사용자가 카메라로 찍은 음식 사진을 인공지능을 통해 인식 후 정보 제공 | | |
| 구현 시 고려사항 | 인식 가능한 음식의 범위를 60가지로 지정해 놓는다.  사용자가 카메라로 음식 사진을 찍으면 인공지능에서 인식된 정확도가 높은 세가지 음식을 사용자에게 띄워준 후 사용자가 선택할 수 있도록 한다.  사진 인식을 못한 사용자가 수기 입력화면으로 넘어가 카테고리에서 해당 음식을 선택한다.  선택된 음식의 칼로리, 영양 정보를 smartpt[DB]-> foodinfo[table]에서 가져와 화면에 출력, 일일 섭취 영양정보에 추가 | | |
| 위험요소 | 1. 인식 가능한 음식 60가지를 제외한 음식은 인식이 불가능하다. | | |
| 위험요소 해결방안 | 1-1. 인식 못하는 음식들은 직접 선택해야 함 | | |
| 관련 요구사항 |  | | |
| 시나리오 | 1. 사용자는 앱의 메인 화면에서 음식을 아침/점심/저녁으로 나눠 카메라를 이용해 촬영하거나 수기로 입력한다.  2. 인식된 음식의 칼로리만큼 오늘 하루 섭취한 칼로리의 양이 증가한다. 인식된 음식은 DB에 저장된다.  3. 사용자는 하루동안 먹은 음식 정보를 확인할 수 있다. | | |

칼로리 초과 표시

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | 칼로리 초과 표시 | 요구사항번호 | SP\_UR\_005 |
| 요구사항구분 | APP | 작성자 | 유재현 |
| 요구사항설명 | 사용자의 섭취 칼로리(INT)가 목표 칼로리(INT)보다 많은 경우 초과된 칼로리 정보 제시 | | |
| 구현 시 고려사항 | 사용자의 섭취 칼로리(INT)가 목표 칼로리(INT)보다 높은 경우, 초과 칼로리 정보(INT)를 칼로리 통계 정보를 확인하는 란에서 사용자가 확인할 수 있도록 제시해준다. | | |
| 위험요소 | 사용자가 목표 칼로리(INT)보다 적거나 같게 음식을 섭취할 수 있다. | | |
| 위험요소 해결방안 | 초과 칼로리를 0Kcal로 표시해준다. | | |
| 관련 요구사항 |  | | |
| 시나리오 | 1. 사용자의 섭취량이 목표 칼로리 보다 많다.  2. 초과된 칼로리 정보를 칼로리 통계 정보와 함께 제시해준다. | | |

식단 추천

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | 식단 추천 | 요구사항번호 | SP\_UR\_006 |
| 요구사항구분 | APP | 작성자 | 신현철 |
| 요구사항설명 | 사용자의 현재 일일 섭취상태에 따라 식단추천을 해준다. | | |
| 구현 시 고려사항 | 사용자가 입력한 칼로리 내에서 미리 짜여진 다이어트 식단 중 조건에 맞는 최대 3가지를 추천해준다.  같은 칼로리의 식단이 3개 초과인 경우 영양소를 기준으로 현재섭취비율(현재까지 섭취한 영양소/권장 영양소) (단백질우선 같을경우->탄수화물->지방 순서로 비교) 이 낮은 영양소를 많이 포함한 음식 순으로 추천 | | |
| 위험요소 | 1. 목표 칼로리를 초과하여 섭취했을 시  2. 오늘 섭취한 음식이 없을 때 | | |
| 위험요소 해결방안 | 1-1. 경고 메시지 띄움  1-2. 물 추천 (0Kcal)  2-1. 입력한 섭취할 목표 칼로리 기준으로 추천 | | |
| 관련 요구사항 |  | | |
| 시나리오 | 1. 사용자가 아무것도 입력하지 않은 상태에서는 아침 식단을 추천한다.  2. 아침 식단이 입력되어 있는 경우 점심 식단을 추천한다.  3. 나머지 칼로리로 저녁 식사를 추천한다. | | |

추천 운동 제공

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | 추천 운동 제공 | 요구사항번호 | SP\_UR\_007 |
| 요구사항구분 | APP | 작성자 | 황연주 |
| 요구사항 설명 | 사용자가 하루동안 섭취한 음식의 칼로리(INT)가 목표 칼로리(INT)를 넘어선 경우 운동을 추천해준다. | | |
| 구현 시 고려사항 | 1. 사용자가 하루동안 섭취한 음식의 칼로리가 목표 칼로리를 넘어서지 않은 경우에는 운동 추천을 띄워주지 않는다.  2. smartpt[DB]->exerciseinfo[table]에 4가지 ( 걷기, 수영, 줄넘기, 사이클 ) 운동 정보 ( 운동이름, 1분 당 소모칼로리) 를 저장해 놓는다. | | |
| 위험요소 | 1. 사용자가 4가지 운동 외에 다른 운동을 할 수 있다. | | |
| 위험요소 해결방안 | 1-1. 가장 대중적인 4가지 운동을 선정하였으며, 4가지 이외의 경우는 고려하지 않는다. | | |
| 관련 요구사항 |  | | |
| 시나리오 | 1. 섭취한 칼로리가 목표 칼로리를 넘어선 시점에 mi band 연동 화면에서 추천 운동 4가지를 (섭취한 칼로리 – 목표 칼로리)에 해당하는 칼로리를 소비할 수 있는 시간만큼 추천해준다. | | |

Mi-fit application과 Google fitness application연동

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | Mi-fit application과 Google  fitness application 연동 | 요구사항번호 | SP\_UR\_008 |
| 요구사항구분 | APP | 작성자 | 정지영 |
| 요구사항설명 | Mi-fit application에서 Google fitness application을 등록하여 두 앱을 연동한다. | | |
| 구현 시 고려사항 | 1. Mi-band와 미스케일의 정보는 Mi-fit application에 저장된다.  2. Mi-fit application과 Google fitness application을 연결하여 Mi-fit application의 정보를 Google fitness application에 동기화한다. | | |
| 위험요소 | 인터넷 환경 및 서버 환경 불안정에 따른 연동 오류 | | |
| 위험요소 해결방안 | 인터넷이 가능한 환경인지 확인 | | |
| 관련요구사항 |  | | |
| 시나리오 | 1. 사용자가 Google fitness application을 다운받는다.  2. 사용자가 Mi-fit application을 다운받는다.  3. Mi-fit application의 프로필 화면에서 계정추가 버튼을 누른다.  4. 그중 Google Fit을 추가하여 Mi-fit과 Google fitness가 연결한다.  5. Mi-fit application의 정보가 Google fitness application에 동기화된다. | | |

운동 데이터 받아오기(mi band 연동)

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | Mi-band 연동 | 요구사항번호 | SP\_UR\_009 |
| 요구사항구분 | APP | 작성자 | 정지영 |
| 요구사항설명 | 사용자가 행한 운동 이름, 운동 시간, 그리고 소모한 칼로리를 받아와서 DB에 저장한다. | | |
| 구현 시 고려사항 | 1. 사용자들이 쉽게 접근할 수 있는 무료 어플리케이션에서 사용자가 원할 때 Mi-band를 통해 기록된 사용자의 운동데이터를 받아 DB에 저장한다.  2. 사용자의 운동데이터에는 운동 이름(String 형식), 운동 시간(int 형식), 그리고 소모한 칼로리(float 형식)가 있다.  3. 요구사항명 “Mi-fit application과 Google fitness application 연동” 이 선행되어야 한다.  4. Mi-band와 Mi-fit application을 블루투스로 연결한다.  5. Google fitness application에서 Google Fit API를 통해 Mi-band의 정보를 받아온다.  6. 받아온 정보에서 운동데이터를 추출하여 DB에 저장한다. | | |
| 위험요소 | 1. 인터넷 환경 및 서버 환경 불안정에 따른 데이터 전송 오류.  2. 요구사항명 “Mi-fit application과 Google fitness application 연동”이 선행되지 않음. | | |
| 위험요소 해결방안 | 1. 인터넷 가능한 환경인지 확인.  2. 요구사항명 “Mi-fit application과 Google fitness application 연동”을 선행. | | |
| 관련요구사항 |  | | |
| 시나리오 | 1. 요구사항명 “Mi-fit application과 Google fitness application 연동” 이 선행되어야 하고, 이를 통해 Mi-fit application의 정보가 Google fitness application에 동기화된다.  2. Mi-fit application과 Mi-band를 블루투스로 연결한다.  3. Google fitness API를 통해 Google fitness application에서 사용자가 행한 운동 이름, 운동 시간, 그리고 소모한 칼로리를 받아 smartpt[DB]의 doexercise[table]에 저장한다.  4. 사용자는 자정부터 현재시간까지((예)10/01/00:00~10/01/16:40) Mi-band를 통해 기록된 운동 이름, 운동 시간, 그리고 소모한 칼로리를 어플리케이션에서 수치로 확인할 수 있다. | | |

인바디 데이터 받아오기 (미스케일 연동)

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | Mi-scale 연동 | 요구사항번호 | SP\_UR\_010 |
| 요구사항구분 | APP | 작성자 | 정지영 |
| 요구사항설명 | 사용자의 인바디 정보(BMI, 체중, 지방량, 근육량)를 받아와서 DB에 저장한다. | | |
| 구현 시 고려사항 | 1. 사용자들이 쉽게 접근할 수 있는 무료 어플리케이션에서 사용자가 원할 때 미스케일을 통해 기록된 사용자의 인바디 정보를 받아온다.  2. 사용자의 인바디 정보에는 BMI(float 형식), 체중(float 형식), 지방량(float 형식) 그리고 근육량(float 형식)이 있다.  3. 미스케일과 Application을 블루투스로 연결한 후 미스케일의 기기 정보(기기명, 주소)를 받아와 저장한다.  4. 블루투스를 통해 미스케일에서 측정한 인바디 정보를 받아 smartpt[DB]의 inbody[table]에 저장한다.  5. 인바디 정보를 하루에(00:00~23:59) 여러 번 저장 요청을 할 경우, 가장 최신의 것으로 갱신한다. 즉, 하루에 한개만 저장된다. | | |
| 위험요소 | 1. 인터넷 환경 및 서버 환경 불안정에 따른 데이터 전송 오류.  2. 블루투스 연동오류 | | |
| 위험요소 해결방안 | 1. 인터넷 가능한 환경인지 확인.  2. 블루투스 연결이 가능한 상태인지 확인. | | |
| 관련요구사항 |  | | |
| 시나리오 | 1. 블루투스를 사용 가능한 상태여야 한다..  2. Application과 미스케일을 블루투스로 연결한다.  3. 블루투스를 통해 미스케일에서 사용자의 인바디 정보를 받아 DB에 저장한다.  4. 사용자는 과거부터 현재까지 미스케일을 통해 기록된 인바디 정보들을 어플리케이션에서 그래프로 확인할 수 있다. | | |

인바디 통게 제공

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | 인바디 통계 제공 | 요구사항번호 | SP\_UR\_011 |
| 요구사항구분 | APP | 작성자 | 황연주 |
| 요구사항 설명 | 사용자의 인바디 정보(근육량 (int형), 체지방량 (int형), BMI (int형), 체중 (int형))를 동기화를 실행한 날짜 별로 그래프로 시각화 하여 화면을 통해 보여준다. | | |
| 구현 시 고려사항 | 인바디 데이터는 smartpt[DB] -> inbody[table]로부터 가져온다. | | |
| 위험요소 | 1. 사용자가 미스케일(체중계)을 연동하지 않을 수 있다. | | |
| 해결방안 | 1-1. 인바디 통계정보를 제공하지 않는다. | | |
| 관련 요구사항 |  | | |
| 시나리오 | 1. 사용자가 미스케일이 연동된 경우, 동기화 버튼을 클릭한다.  2. google fit 으로부터 받아온 정보를 동기화를 실행한 날짜 별로 그래프를 통해 제공한다. | | |

섭취한 음식 칼로리 통계 제공

|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | 섭취한 음식 칼로리 통계 제공 | 요구사항번호 | SP\_UR\_012 |
| 요구사항구분 | APP | 작성자 | 황연주 |
| 요구사항 설명 | 사용자가 섭취한 음식의 칼로리를 일/주/월 별로 그래프로 시각화 하여 화면을 통해 보여준다. | | |
| 구현 시 고려사항 | 음식 칼로리 데이터는 DB의 eatenfood 테이블로부터 가져온다. | | |
| 위험요소 |  | | |
| 해결방안 |  | | |
| 관련 요구사항 |  | | |
| 시나리오 | 1. 사용자가 섭취한 음식을 입력한다 ( 카메라 혹은 수기로 )  2. 인식된 음식 정보는 smartpt[DB]->eatenfood[table] 에 저장된다.  3. 음식 칼로리 정보를 일 / 주 / 월 별로 그래프를 통해 제공한다. | | |

섭취한 음식 정보 제공

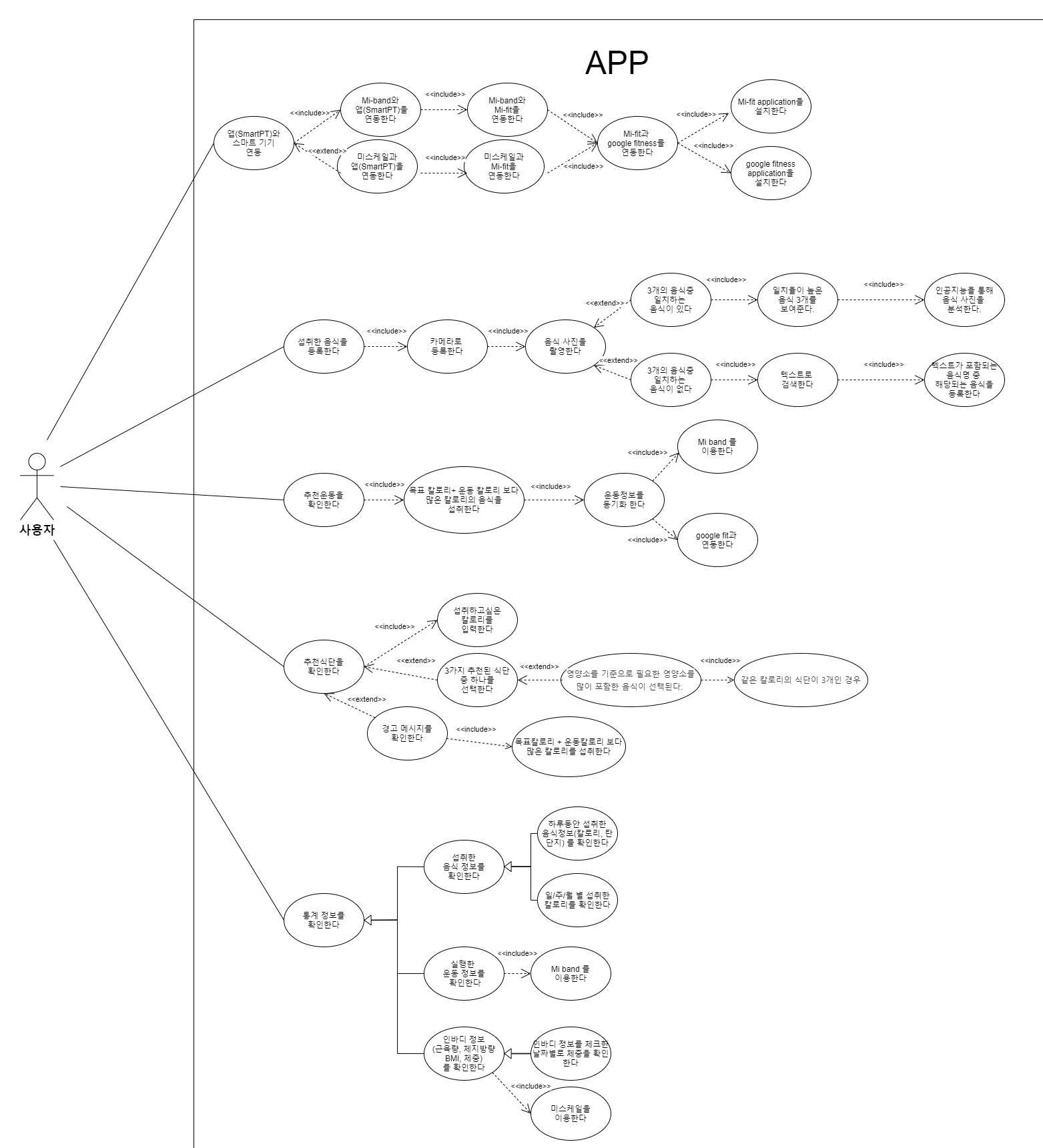
|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | 섭취한 음식 정보 제공 | 요구사항번호 | SP\_UR\_013 |
| 요구사항구분 | APP | 작성자 | 황연주 |
| 요구사항 설명 | 하루동안 사용자가 섭취한 음식 정보( 칼로리, 탄수화물, 단백질, 지방)를 아침, 점심, 저녁으로 나누어 텍스트와 그래프 형식으로 보여준다. | | |
| 구현 시 고려사항 | 1. 음식 정보는 smartpt[DB] -> foodinfo[table]로부터 가져온다.  2. 음식 이름은 텍스트로, 칼로리, 탄수화물, 단백질 ,지방은 그래프로 보여준다.  3. 섭취한 음식정보는 smartpt[DB] ->eatenfood[table]에 저장된다. | | |
| 위험요소 |  | | |
| 해결방안 |  | | |
| 관련 요구사항 |  | | |
| 시나리오 | 1. 사용자는 앱의 메인화면에서 아침, 점심, 저녁 버튼을 클릭하여 먹은 음식을 카메라를 이용해 촬영하거나 수기로 입력한다.  2. 해당되는 시간 ( 아침, 점심, 저녁 ) 에 따라 인식된 음식의 칼로리(int형) , 탄수화물 (int형), 단백질 (int형) ,지방 (int형) 함량을 메인화면에서 보여준다. 인식된 음식은 DB의 eatenfood 테이블에 저장된다.  3. 앱을 껐다 키더라도 하루동안 정보는 유지된다. | | |

운동 통계 제공

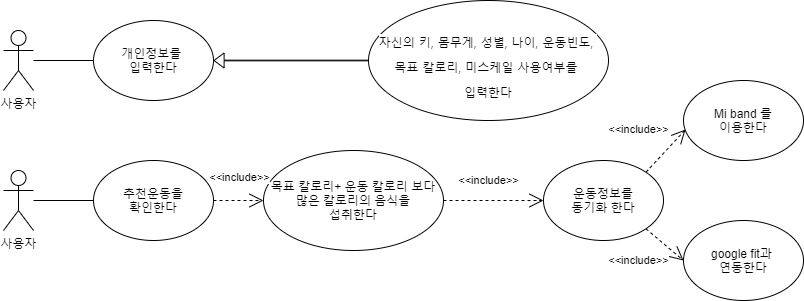
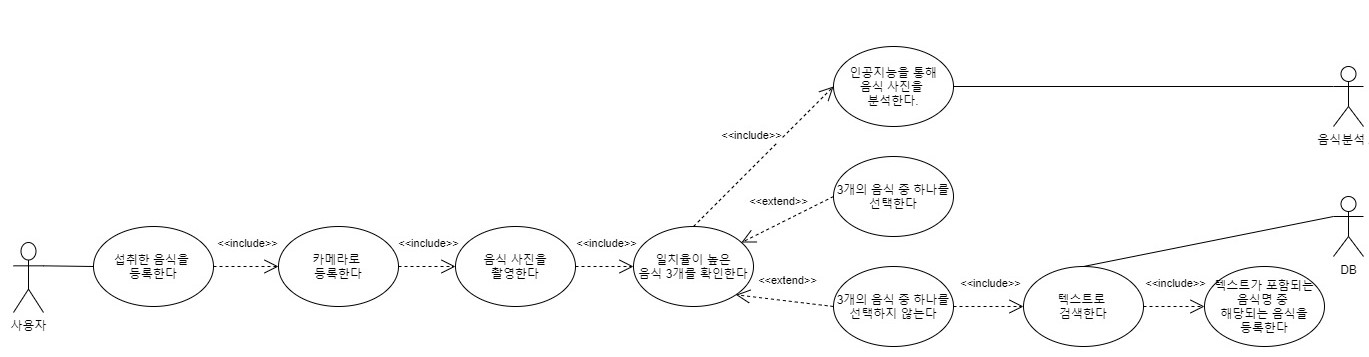
|  |  |  |  |
| --- | --- | --- | --- |
| 요구사항명 | 운동 통계 제공 | 요구사항번호 | SP\_UR\_014 |
| 요구사항구분 | APP | 작성자 | 황연주 |
| 요구사항 설명 | 하루동안 실행한 운동 정보 ( 운동이름, 운동시간, 소모칼로리) 를 화면을 통해 보여준다. | | |
| 구현 시 고려사항 | 1. 운동 정보는 smartpt[DB] -> doexercise[table]로부터 가져온다.  2. 동기화 버튼을 누르면 하루 누적 운동 정보가 갱신된다 ( Update ) | | |
| 위험요소 |  | | |
| 위험요소 해결방안 |  | | |
| 관련 요구사항 |  | | |
| 시나리오 | 1. 사용자가 동기화 버튼을 클릭한다.  2. google fit으로부터 하루동안 실행한 운동 정보를 받아와서 메인화면과 Mi band 연동 화면에 그래프를 통해 보여준다. | | |

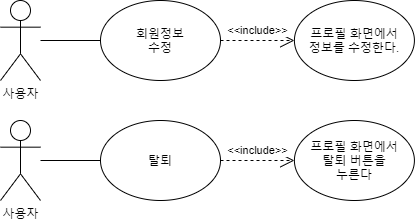
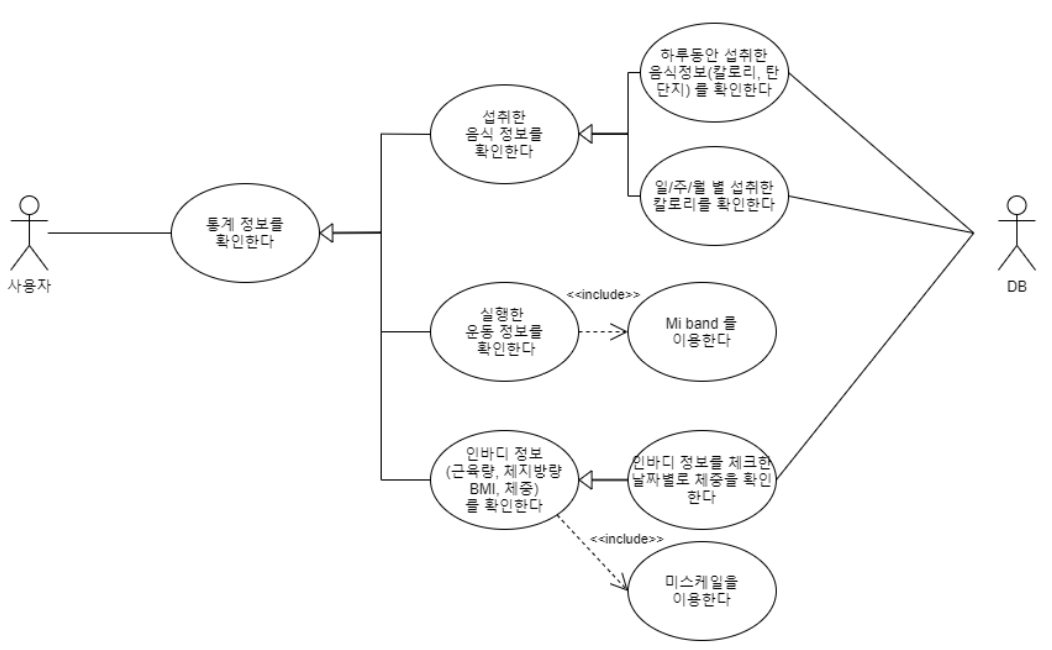
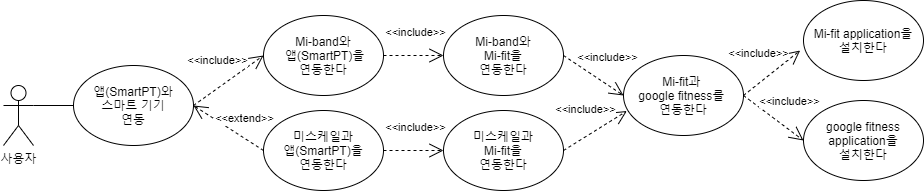
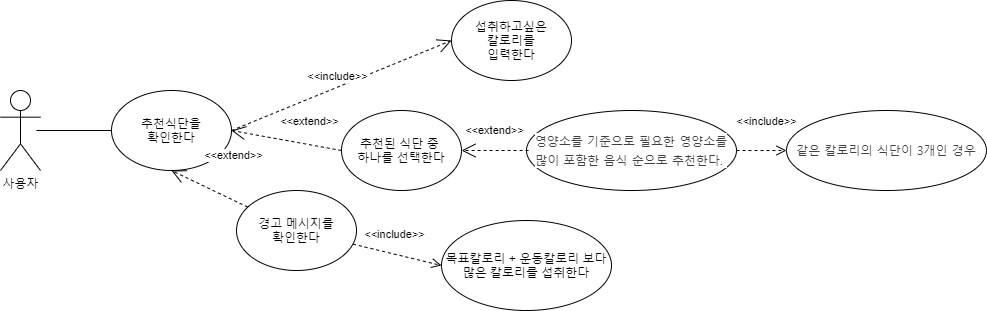
## 2.3 유스케이스 다이어그램

### **2.3.1 유스케이스 다이어그램**



### **2.3.2 세부 유스케이스 다이어그램**





### **2.3.3 유스케이스 기술서**

섭취한 음식 등록

|  |  |
| --- | --- |
| 코드 | SP\_UD\_001 |
| 유스케이스명 | 섭취한 음식 등록 |
| 액터명 | 사용자, 카메라, DB |
| 개요 | 사용자는 하루동안 섭취한 음식을 스마트폰 카메라로 찍거나 검색한다. |
| 사전조건 | 사용자는 음식의 사진을 촬영할 수 있는 상태이거나, 음식명을 알고 있어야 한다. |
| 사후조건 | 앱에 사용자가 섭취한 음식이 등록되었다. |
| 기본흐름 | 1. 사용자는 앱의 메인화면에서 해당 시간(아침, 점심 저녁)을 선택 2. 먹은 음식을 카메라를 이용해 촬영한다. 3. 인식된 음식은 smartpt{DB} -> eatenfood [table]에 저장   3-1. 음식 저장 format  음식id(INT) - userid(INT) - 먹은날(date) -먹은 시기(아침,점심,저녁)(INT) - 먹은 인분(INT) |
| 대체흐름 | 1a. 음식 분석 알고리즘이 제공하는 음식 60 가지 이외의 음식인 경우  1a-1.음식 이름을 DB 정보를 가지고있는 후보음식(1200가지) 에서 선택(클릭)  1a-2. 섭취한 음식정보가 저장되고 유스케이스는 종료된다 |

IoT 기기 연결

|  |  |
| --- | --- |
| 코드 | SP\_UD\_002 |
| 유스케이스명 | IoT 기기 연결 |
| 액터명 | 사용자 |
| 개요 | 사용자는 앱과 IoT기기를 연결한다. |
| 사전조건 | 사용자는 블루투스를 이용할 수 있는 상황이어야 하며 어플(구글 피트니스/미핏)을 다운받을 수 있는 상황이거나, 해당 어플이 다운받아져 있어야한다. |
| 사후조건 | 앱에서 사용자의 IoT에 기록된 정보들(소모칼로리, 운동시간, 인바디정보)을 확인할 수 있다. |
| 기본흐름 | 1. 사용자는 google fitness application을 설치한다 2. 사용자는 Mi-fit application을 설치한다. 3. 사용자는 google fitness application에서 Mi-fit application를 등록한다. 4. 사용자는 Mi-band나 미스케일을 Mi-fit application과 블루투스로 연결한다. |

개인정보 입력

|  |  |
| --- | --- |
| 코드 | SP\_UD\_003 |
| 유스케이스명 | 개인 정보 입력 |
| 액터명 | 사용자 |
| 개요 | 사용자는 앱에 자신의 키(INT), 체중(INT), 나이(INT), 성별(INT), 운동빈도(INT), 목표칼로리(INT), 체중계사용여부 (BLOB) 를 입력해야 한다. |
| 사전조건 | 사용자는 앱에 자신의 개인정보를 입력하지 않았거나 수정을 원한다. |
| 사후조건 | 사용자의 개인정보가 앱에 등록되었다. |
| 기본흐름 | 1. 유스케이스는 사용자가 자신의 개인정보를 입력/수정할 때 시작한다.  2. 사용자는 개인정보란으로 이동한다.  3. 사용자는 자신의키(INT), 체중(INT), 나이(INT), 성별(INT), 운동빈도(INT), 목표칼로리(INT), 체중계사용여부 (BLOB) 를 입력함으로써 유스케이스는 종료된다. |

추천운동 확인

|  |  |
| --- | --- |
| 코드 | SP\_UD\_004 |
| 유스케이스명 | 추천 운동 확인 |
| 액터명 | 사용자 |
| 개요 | 사용자는 앱으로부터 하루 운동량(INT)과 섭취한 칼로리(INT)를 기반으로, 초과 칼로리(INT)에 해당하는 운동을 추천받는다 |
| 전조건 | 섭취한 음식등록 유스케이스가 먼저실행되어있어야한다  사용자가 Mi band 를 사용하며 Mi fit 과 Google fit 이 연동 되어있는 상태이어야 함  사용자가 동기화 버튼을 눌러야 함  목표 칼로리와 운동량을 합친 것보다 하루동안 섭취한 칼로리가 더 많아야 함 |
| 사후조건 | 사용자는 초과한 칼로리를 소모 시킬 수 있는 운동을 알게 되었다. |
| 기본흐름 | 1. 섭취한 칼로리(INT)가 목표 칼로리를 넘어섰는지확인  2. 섭취한 칼로리(INT)가 목표 칼로리(INT)를 넘어선 시점에 mi band 연동 화면에서 추천 운동 4가지를 ‘섭취한 칼로리(INT) – 목표 칼로리(INT)’가 양의 정수인경우 해당하는 칼로리를 소비할 수 있는 시간만큼 추천해준다. |

추천 식단 확인

|  |  |
| --- | --- |
| 코드 | SP\_UD\_005 |
| 유스케이스명 | 추천 식단 확인 |
| 액터명 | 사용자 |
| 개요 | 사용자는 앱으로부터 현재 일일 음식 섭취상태에 따라 식단을 추천 받는다 |
| 사전조건 | 사용자는 목표 칼로리(INT)를 설정해 놔야 함  음식을 섭취한 경우, 음식 정보가 정확히 입력되어 있어야함  사용자는 첫 끼를 자유롭게 먹고, 섭취하고 싶은 칼로리(INT)를 입력함 |
| 사후조건 | 식단 추천이 성공적으로 실행된 경우 시스템은 사용자의 조건에 맞는 최대 3가지 음식을 추천해줌 |
| 기본흐름 | 1. 사용자는 식단 탭을 선택한다.  2. (사용자의 목표 칼로리(INT) + 운동칼로리(INT) – 사용자가 오늘 하루 섭취한 칼로리(INT)) 내에서 사용자는 원하는 칼로리만큼 수기로 입력한다.  3. 사용자는 smartpt[DB]- diet[table]로부터 해당 칼로리와 가장 근접한 식단 3가지를 추천받는다. |
| 대체흐름1 | 2a. 섭취할 수 있는 칼로리를 초과해서 섭취했을 시  2a-1. 경고메시지를 사용자가 섭취칼로리(INT)/목표칼로리(INT)를 확인하는 란에 하단에 띄워준다. |
| 대체흐름2 | 2b. 오늘 섭취한 음식이 없는 경우  2b-1. 입력한 섭취할 목표 칼로리 기준으로 추천한다. |
| 대체흐름3 | 3a. 같은 칼로리의 식단이 3개인 경우  3a-1. 영양소를 기준으로 필요한 영양소를 많이 포함한 음식 순으로 추천한다. |

통계 정보 확인

|  |  |
| --- | --- |
| 코드 | SP\_UD\_006 |
| 유스케이스명 | 통계 정보 확인 |
| 액터명 | 사용자 |
| 개요 | 사용자는 앱으로부터 통계 정보를 확인한다  통계정보는  1) 섭취한 음식정보 ( 칼로리, 탄/단/지 )  2) 실행한 운동정보 ( 운동이름, 운동시간, 소모한 칼로리)  3) 인바디 정보 ( 체중, 근육량, 체지방량 , BMI )  로 구성된다. |
| 사전조건 | 사용자가 Mi band 를 사용하며 Mi fit 과 Google fit 이 연동 되어있는 상태이어야 함  인바디 정보의 경우 사용자는 미스케일을 사용해야 하며, 이는 Google fit 과 연동 되어있는 상태이어야 함  운동정보와 인바디 정보의 경우 사용자가 동기화 버튼을 눌러야 함  섭취한 음식정보의 경우 사용자는 음식 정보를 입력해야 함 |
| 사후조건 | 해당 없음 |
| 기본흐름 | 섭취한 음식 정보 |
| 1. 사용자는 앱의 메인 화면에서 해당 시간(아침, 점심 저녁)의 먹은 음식을 카메라를 이용해 촬영하거나 수기로 입력한다.  2. 시스템은 음식 인식 알고리즘을 이용하여 음식을 분석한다.  3. 분석된 음식과 일치하는 음식의 정보를 smartpt[DB] -> foodinfo[table]로부터 가져온다.  4. 섭취한 음식정보는 smartpt[DB]-> eatenfood[table]에 저장된다.  5. 음식 정보 중 음식이름은 텍스트로, 칼로리와 탄수화물, 단백질 ,지방 함량은 그래프로 화면에 출력된다.  6. 앱을 껐다가 키더라도 하루 동안은 정보가 유지된다. |
| 실행한 운동 정보 |
| 1. 사용자가 동기화 버튼을 클릭한다.  2. 시스템은google fit으로부터 운동정보( 운동이름, 운동시간, 소모 칼로리 )를 받아온다.  3. 운동정보는 smartpt[DB] -> doexercise[tabe]에 저장된다.  4. 운동정보는 화면에 출력된다.  4-1. 운동종류(text) / 운동시간(분) / 소모칼로리(INT) |
| 인바디 정보 |
| 1. 사용자가 미스케일이 연동된 경우, 동기화 버튼을 클릭한다.  2. 시스템은 google fit으로부터 인바디 정보( 체중, 근육량, 체지방량 ,BMI )를 받아온다.  3. 인바디 정보는 smartpt[DB] -> inbody[table]에 업데이트된다.  4. 통계화면에서 동기화를 실행한 날짜 별로 그래프로 화면에 출력된다. |

## 2.4 클래스 다이어그램

### **2.4.1 설계 클래스도**



## 2.5 시퀀스 다이어그램

### **2.5.1 시퀀스 목록**

|  |  |  |
| --- | --- | --- |
| 시퀀스 ID | 시퀀스 명 | 시퀀스 설명 |
| SP\_SD\_ 001 | 개인정보 입력 | 1. 사용자는 앱의 개인정보 입력란에서 개인정보를 입력한다.  2. 해당 개인정보는 DB에 저장된다. |
| SP\_SD\_002 | 섭취한 음식 등록 | 1. 사용자는 음식을 등록하기 위해 음식을 촬영한다.  2. 인공지능에서 해당 사진이 어떤음식인지 구별한다.  3. 구별된 음식이 맞을 경우 DB에 등록되고, 그렇지 않을 경우 음식선택란으로 전환된다.  4. 음식선택란에서 선택된 음식이 DB에 등록된다.  5. DB에 등록된 음식은 메인화면에 칼로리와 탄단지 정보가 출력된다. |
| SP\_SD\_003 | IoT 연동 | 1. 사용자는 미핏과 구글 피트니스 어플을 다운받는다.  2. 미핏 어플에서 구글 피트니스 어플을 등록한다.  3. 미핏 어플과 IoT는 블루투스로 연결한다. |
| SP\_SD\_004 | IoT 데이터 저장 | 1. 사용자가 데이터 동기화 버튼을 누른다.  2. 앱에서 구글 피트니스에 정보를 요청한다.  3. 해당 정보는 DB에 저장된다. |
| SP\_SD \_005 | 통계 정보 확인 | 1. DB에 사용자의 칼로리/몸무게/인바디 정보를 요청한다.  2. 해당 정보를 앱에 그래프 형식으로 출력한다. |
| SP\_SD\_006 | 추천 운동 조회 | 1. 섭취칼로리>(운동칼로리+목표칼로리) 의 경우 초과된 칼로리를 바탕으로 운동을 추천한다.  2. 섭취칼로리>(운동칼로리+목표칼로리) 의 경우 추천운동을 제공하지 않는다. |
| SP\_SD\_007 | 추천 식단 확인 | 1. 목표칼로리<섭취칼로리 인 경우경고메세지를 출력한다.  2. 그렇지 않은 경우 섭취칼로리와 입력칼로리를 비교한다.  3. 섭취칼로리<입력칼로리의 경우 경고메세지를 출력한다.  4. 그렇지 않은 경우 점심용 식단을 3가지 추천한다.  5. 점심 식단을 선택하면 해당 칼로리를 바탕으로 저녁용 식단을 3가지 추천한다. |

### **2.5.2 시퀀스 다이어그램**

#### 개인정보 입력

|  |  |  |  |
| --- | --- | --- | --- |
| 시퀀스 ID | SP\_SD\_001 | 시퀀스 명 | 개인정보 입력 |
|  | | | |

#### 섭취한 음식 등록

|  |  |  |  |
| --- | --- | --- | --- |
| 시퀀스 ID | SP\_SD\_002 | 시퀀스 명 | 섭취한 음식 등록 |
|  | | | |

#### IoT 연동

|  |  |  |  |
| --- | --- | --- | --- |
| 시퀀스 ID | SP\_SD\_003 | 시퀀스 명 | IoT 연동 |
|  | | | |

#### IoT 데이터 저장

|  |  |  |  |
| --- | --- | --- | --- |
| 시퀀스 ID | SP\_SD\_004 | 시퀀스 명 | IoT 데이터 저장 |
|  | | | |

#### 통계 정보 확인

|  |  |  |  |
| --- | --- | --- | --- |
| 시퀀스 ID | SP\_SD\_005 | 시퀀스 명 | 통계 정보 확인 |
|  | | | |
|  | | | |

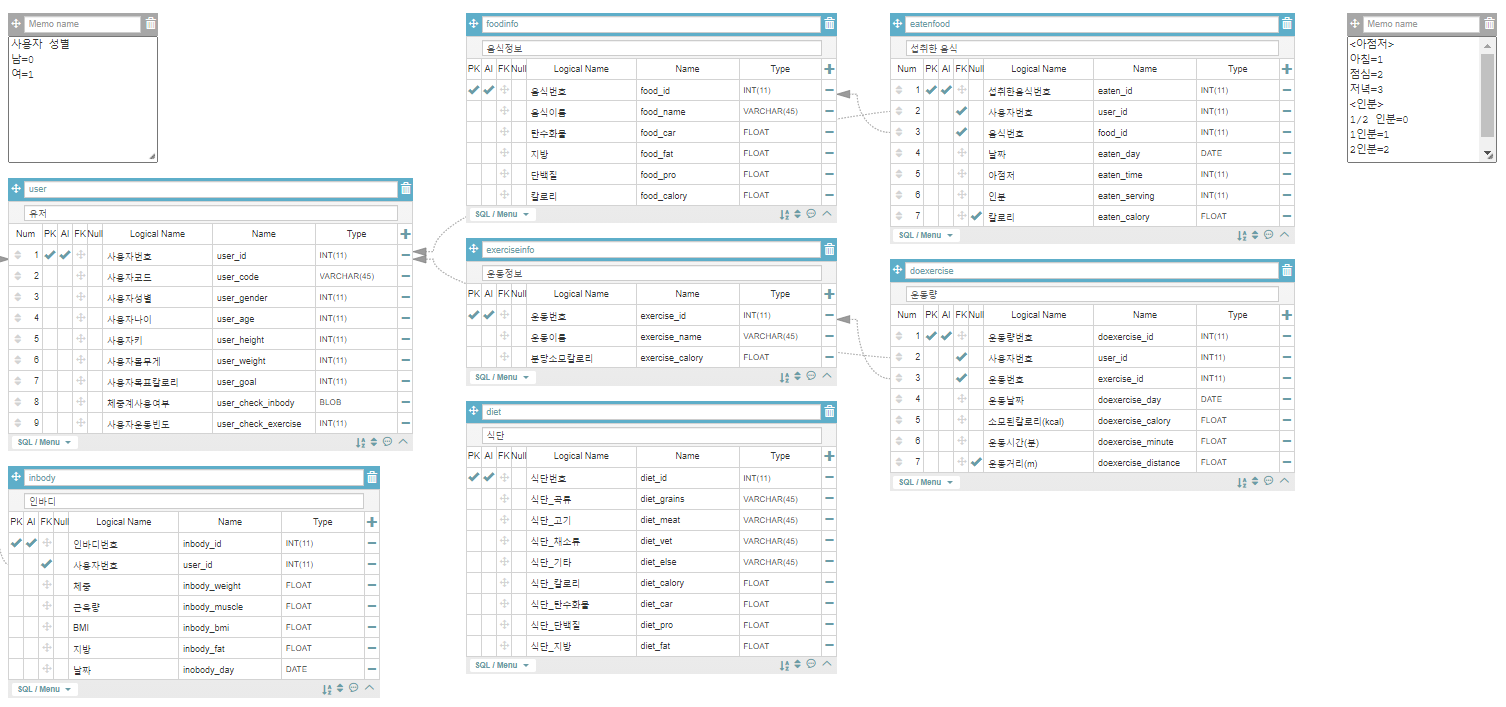
#### 추천 운동 조회

|  |  |  |  |
| --- | --- | --- | --- |
| 시퀀스 ID | SP\_SD\_006 | 시퀀스 명 | 추천 운동 조회 |
|  | | | |

#### 추천 식단 확인

|  |  |  |  |
| --- | --- | --- | --- |
| 시퀀스 ID | SP\_SD\_007 | 시퀀스 명 | 추천 식단 확인 |
|  | | | |

## 2.6 데이터베이스 설계도



테이블 명세

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 데이터베이스명 | smartPT | | 테이블명 | user | | |
| 테이블 설명 | 유저 테이블 | | | | | |
| 초기건수 | | | 보관주기 | | | |
| 0 | | | 영속 | | | |
| 컬럼명 | 컬럼ID | 타입 및 길이 | Not Null | PK | | FK |
| 사용자번호 | user\_id | INT(11) | O | O | |  |
| 사용자코드 | user\_code | VARCHAR(45) | O |  |  | |
| 사용자성별 | user\_gender | INT(11) | O |  |  | |
| 사용자나이 | user\_age | INT(11) | O |  |  | |
| 사용자키 | user\_height | INT(11) | O |  |  | |
| 사용자몸무게 | user\_weight | INT(11) | O |  |  | |
| 사용자목표칼로리 | user\_goal | INT(11) | O |  |  | |
| 체중계사용여부 | user\_check\_inbody | BLOB | O |  |  | |
| 사용자운동빈도 | User\_check\_exercise | INT(11) | O |  |  | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 데이터베이스명 | smartPT | | 테이블명 | foodinfo | | |
| 테이블 설명 | 음식 정보 테이블 | | | | | |
| 초기건수 | | | 보관주기 | | | |
| 5290 | | | 영속 | | | |
| 컬럼명 | 컬럼ID | 타입 및 길이 | Not Null | PK | | FK |
| 음식번호 | food\_id | INT(11) | O | O | |  |
| 음식이름 | food\_name | VARCHAR(45) | O |  |  | |
| 탄수화물 | food\_car | FLOAT | O |  |  | |
| 지방 | food\_fat | FLOAT | O |  |  | |
| 단백질 | food\_pro | FLOAT | O |  |  | |
| 칼로리 | food\_calory | FLOAT) | O |  |  | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 데이터베이스명 | smartPT | | 테이블명 | exerciseinfo | | |
| 테이블 설명 | 운동 정보 테이블 | | | | | |
| 초기건수 | | | 보관주기 | | | |
| 2 | | | 영속 | | | |
| 컬럼명 | 컬럼ID | 타입 및 길이 | Not Null | PK | | FK |
| 운동번호 | exercise\_id | INT(11) | O | O | |  |
| 운동이름 | exercise\_name | VARCHAR(45) | O |  |  | |
| 분당소모칼로리 | exercise\_calory | FLOAT | O |  |  | |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 데이터베이스명 | smartPT | | 테이블명 | diet | | |
| 테이블 설명 | 식단 정보 테이블 | | | | | |
| 초기건수 | | | 보관주기 | | | |
| 60 | | | 영속 | | | |
| 컬럼명 | 컬럼ID | 타입 및 길이 | Not Null | PK | | FK |
| 식단번호 | diet\_id | INT(11) | O | O | |  |
| 식단\_곡류 | diet\_grains | VARCHAR(45) | O |  |  | |
| 식단\_고기 | diet\_meat | VARCHAR(45) | O |  |  | |
| 식단\_채소류 | diet\_vet | VARCHAR(45) | O |  |  | |
| 식단\_기타 | diet\_else | VARCHAR(45) | O |  |  | |
| 식단\_칼로리 | diet\_calory | FLOAT | O |  |  | |
| 식단\_탄수화물 | diet\_car | FLOAT | O |  |  | |
| 식단\_단백질 | diet\_pro | FLOAT | O |  |  | |
| 식단\_지방 | diet\_fat | FLOAT | O |  |  | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 데이터베이스명 | smartPT | | 테이블명 | eatenfood | |
| 테이블 설명 | 섭취한 음식 정보 테이블 | | | | |
| 초기건수 | | | 보관주기 | | |
| 0 | | | 영속 | | |
| 컬럼명 | 컬럼ID | 타입 및 길이 | Not Null | PK | FK |
| 섭취한음식번호 | eaten\_id | INT(11) | O | O |  |
| 사용자번호 | user\_id | INT(11) | O |  | [user]user\_id |
| 음식번호 | food\_id | INT(11) | O |  | [foodinfo]food\_id |
| 날짜 | eaten\_day | DATE | O |  |  |
| 아점저 | eaten\_time | INT(11) | O |  |  |
| 인분 | eaten\_serving | INT(11) | O |  |  |
| 칼로리 | eaten\_calory | FLOAT | O |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 데이터베이스명 | smartPT | | 테이블명 | doexercise | |
| 테이블 설명 | 활동한 운동 정보 테이블 | | | | |
| 초기건수 | | | 보관주기 | | |
| 0 | | | 영속 | | |
| 컬럼명 | 컬럼ID | 타입 및 길이 | Not Null | PK | FK |
| 운동량번호 | doexercise\_id | INT(11) | O | O |  |
| 사용자번호 | user\_id | INT(11) | O |  | [[user]user\_id |
| 운동번호 | exercise\_id | INT(11) | O |  | [exerciseinfo]exercise\_id |
| 운동날짜 | doexercise\_day | DATE | O |  |  |
| 소모된칼로리 | doexercise\_calory | FLOAT | O |  |  |
| 운동시간 | doexercise\_munute | FLOAT | O |  |  |
| 운동거리 | doexercise\_distance | FLOAT | O |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 데이터베이스명 | smartPT | | 테이블명 | inbody | |
| 테이블 설명 | 인바디 정보 테이블 | | | | |
| 초기건수 | | | 보관주기 | | |
| 0 | | | 영속 | | |
| 컬럼명 | 컬럼ID | 타입 및 길이 | Not Null | PK | FK |
| 인바디번호 | inbody\_id | INT(11) | O | O |  |
| 사용자번호 | user\_id | INT(11) | O |  | [user]user\_id |
| 체중 | inbody\_weight | FLOAT | O |  |  |
| 근육량 | inbody\_muscle | FLOAT | O |  |  |
| BMI | inbody\_bmi | FLOAT | O |  |  |
| 지방 | inbody\_fat | FLOAT | O |  |  |
| 날짜 | inbody\_day | DATE | O |  |  |

## 2.6 개발 환경 명세서

### **2.6.1 주 개발 프로그램**

안드로이드 스튜디오, atom, visual studio code

### **2.6.2 보조 개발 프로그램**

sublimetext3, idle3, Xshell

### **2.6.3 협업 프로그램**

Github,Trello

# 3. 개발 추진 계획

|  |  |  |  |
| --- | --- | --- | --- |
| Sub system 명 | Use case 명 | 개발 순서 | |
| 메인 화면 | 섭취 음식 등록 | 11월 2째주까지 개발 완료 할 것 |
| 통계 화면 | 통계 정보 확인 | 11월 2째주까지 개발 완료 할 것 |
| 식단 추천 화면 | 추천 식단 확인 | 11울 1째주까지 개발 완료 할 것 |
| 미밴드 화면 | 추천 운동 확인 | 11월 2째주까지 개발 완료 할 것 |
| 스마트 기기 연결 | IoT 기기 연결 | 11월 2째주까지 개발 완료 할 것 |



# 4. 개발 내용

## 4.1 주요 기능 구현

|  |  |  |
| --- | --- | --- |
| * 섭취한 음식 입력 * 섭취한 음식 칼로리 및 영양소 정보 제공 * 활동한 운동제공 | * 추천식단제공 * 추천운동제공 * 미핏과 구글핏 연동 | * 섭취 칼로리 통계 제공 * 인바디 통계 제공 |

#### 섭취한 음식 입력

|  |  |
| --- | --- |
| Click  Click  Click | 음식을 촬영해 섭취한 음식 기입한다. |
| Click | DB에 저장된 모든 음식을 볼 수 있고 필터링 기능을 이용하여 사용자가 원하는 음식이름을 입력 후, 선택하여 음식을 정보를 기입한다. |

#### 섭취한 음식 칼로리 및 영양소 정보 제공

|  |  |
| --- | --- |
|  | 메인 화면에서 오늘 하루동안 섭취한  칼로리와 영양소 정보를 그래프 형식으로 제공한다 |
|  | 식단 화면에서 오늘 하루동안 섭취한  칼로리와 영양소 정보를 그래프 형식으로 제공하며 , 아침 / 점심 / 저녁별로 해당 시간대에 먹은 음식의 칼로리와 영양소 정보를 텍스트 형식으로 제공한다. |

#### 활동한 운동제공

|  |  |
| --- | --- |
|  | 메인 화면에서 오늘 하루동안 활동한 운동 ( 걷기 / 달리기 ) 정보를 제공한다. |

#### 추천식단 제공

|  |  |
| --- | --- |
|  | 사용자가 원하는 칼로리를 입력하면, 원하는 칼로리보다 낮은 칼로리 중, 가장 근사치의 식단 3가지를 표기하여 주고, 아래로 스크롤 시, 그보다 낮은 식단 정보들을 표기하여 준다. |

#### 추천 운동 제공

|  |  |
| --- | --- |
| Click | 오늘 하루동안 사용자가 섭취한 음식 칼로리가 기존에 등록했던 목표양보다 많으면, 즉 목표보다 많이 섭취했을 경우 메인화면에서 경고 알림이 표시되며 칼로리 부분을 클릭 시 추천 운동을 제공한다. |

#### 미핏과 구글핏 연동

|  |  |
| --- | --- |
| Click  Click  Click | 미핏 앱에서 구글 피트니스 앱을 등록하여 두 앱을 연동한다. |

#### 섭취 칼로리 통계 제공

|  |  |
| --- | --- |
|  | 사용자가 섭취한 음식의 칼로리를 일 / 주 / 월 별로 그래프를 통해 정보를 제공한다. |

#### 인바디 통계 제공

|  |  |
| --- | --- |
|  | 사용자가 미스케일을 연동해서 사용하는 경우 , 인바디 정보(Bmi , 체중, 지방량, 근육량) 를 그래프를 통해 제공한다. |

# 5. 테스트 명세서

## 5.1 기능테스트

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 시험 유형 | | 기능 테스트 | | | |
| 관련 요구사항 ID | | SP\_UR\_003, SP\_UR\_006 | | | |
| 시험시나리오 ID | 시험시나리오 명 | 시험절차 | 만족기준 | 시험결과 | 비고 |
| SP\_PT\_001 | 추천식단 확인 | 원하는 칼로리 입력하고 그 칼로리를 초과하지 않는 식단들이 추천 돼야함 | 초과하지 않는 식단 추천 | 만족 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 시험 유형 | | 기능 테스트 | | | |
| 관련 요구사항 ID | | UR005, UR007 | | | |
| 시험시나리오 ID | 시험시나리오 명 | 시험절차 | 만족기준 | 시험결과 | 비고 |
| SP\_PT\_002 | 추천 운동 확인 | 해당 추천 운동을 통해 초과된 칼로리 모두 소모할 수 있는지 확인 | 추천운동 추천된 시간 만큼 하면 칼로리가 초과하지 않아야함 | 만족 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 시험 유형 | | 기능 테스트 | | | |
| 관련 요구사항 ID | | SP\_UR\_008, SP\_UR\_009, SP\_UR\_010,  SP\_UR\_011, SP\_UR\_012, SP\_UR\_013 | | | |
| 시험시나리오 ID | 시험시나리오 명 | 시험절차 | 만족기준 | 시험결과 | 비고 |
| SP\_PT\_003 | 통계정보확인 | 통계 화면의 데이터가 맞는 값들인지 확인 | 모든 값 일치 | 일치 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 시험 유형 | | 기능 테스트 | | | |
| 관련 요구사항 ID | | SP\_UR\_008, SP\_UR\_009 | | | |
| 시험시나리오 ID | 시험시나리오 명 | 시험절차 | 만족기준 | 시험결과 | 비고 |
| SP\_PT\_004 | 스마트기기 연결 | 연결 후 측정한 데이터값과 가져온 값이 일치하는지 확인 | 모든 값 일치 | 일치 |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 시험 유형 | | 기능 테스트 | | | |
| 관련 요구사항 ID | | SP\_UR\_012 | | | |
| 시험시나리오 ID | 시험시나리오 명 | 시험절차 | 만족기준 | 시험결과 | 비고 |
| SP\_PT\_005 | 섭취한 음식 등록 | 입력한 음식의 이름과 칼로리 정보 일치하는지 확인 | 모든 값 일치 | 일치 |  |

## 5.2 성능테스트

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 시험 유형 | | 성능 테스트 | | | |
| 관련 요구사항 ID | | SP\_UR\_012 | | | |
| 시험시나리오 ID | 시험시나리오 명 | 시험절차 | 만족기준 | 시험결과 | 비고 |
| SP\_PT\_006 | 사진을 통한 음식 인식률 | 음식 사진을 찍고 맞는 음식이 나오는지 확인 | 후보 음식 중 맞는 음식 있어야 함 | 보완필요 | 정확도45% |

# 6. 요구사항 대비 시스템 구현 현황

## 6.1 제안서 대비 수정 내용

1) 스마트 워치(Mi Band)를 통해 저장하는 정보

|  |  |
| --- | --- |
| 기존 내용 | Google Fit와 Mi-Fit 연동 후, Google Fit Api를 이용하여 기록된 운동정보(걷기, 수영, 자전거, 줄넘기의 소모시간과 소모칼로리)를 받아온다 |
| 수정된 내용 | 걷기, 수영, 자전거, 줄넘기의 운동 정보(소모시간, 소모칼로리)를 받아오려 했으나, Mi-fit에서 수영, 자전거, 줄넘기의 운동 정보들을 Google Fit과 공유하지 않기 때문에 걷기, 달리기의 운동정보(소모시간, 소모칼로리)만 가져와 앱과 DB에 저장한다. (블루투스를 통해 미밴드와 연결하는 것은 인증키가 필요하기에 선택하지 않았다.) |

2) smart 체중계(Mi Scale) 연결 방법

|  |  |
| --- | --- |
| 기존 내용 | Google Fit와 Mi-Fit 연동 후, Google Fit Api를 이용하여 기록된 인바디정보를 받아온다 |
| 수정된 내용 | 몸무게, 근육비율, 지방비율, 체수분비율을 받아오려 했으나, Mi-fit에서 몸무게를 제외한 나머지 정보들을 Google Fit과 공유하지 않기 때문에, Google Fit이 아닌 블루투스를 이용하여 미스케일의 정보들을 받아오기로 하였다. 블루투스를 통해 미스케일을 검색하고, 사용자는 검색된 블루투스 기기중 미스케일을 선택한다. 해당 기기와 블루투스를 통해 연결이 된 상태에서 사용자가 미스케일에서 인바디를 측정하면 해당 정보를 앱과 DB에 저장한다. |

3) 회원 목표 칼로리 설정

|  |  |
| --- | --- |
| 기존 내용 | 기초대사량 : 남자 : 66.47 + (13.75 X 체중 ) + (5 X 키) - (6.76 X 나이)  여자 64.41 + (9.56 X 체중 ) + (1.85X 키) - (4.68 X 나이)  활동대사량 : 기초대사량 \* 활동계수 ( 활동계수 : 운동을 전혀하지 않음 : 0.3 / 적당한 운동 : 0.5 /강도 높은 운동:0.7)  권장칼로리 : 기초대사량 + 활동대사량 |
| 수정된 내용 | 권장칼로리 : 성인여자=354-6.91x연령(세)+PA[9.36x체중(kg)+726x신장(m)]  성인남자=662-9.53x연령(세)+PA[15.91x체중(kg)+539.6x신장(m)]  ※ PA(신체활동계수): 1.0(비활동적), 1.12(저활동적), 1.27(활동적),1.45(매우 활동적) |

4) 식단추천

|  |  |
| --- | --- |
| 기존 내용 | 1. 사용자의 목표 칼로리와 권장 영양 섭취 기준을 바탕으로, 사용자가 해당 시점까지 먹은 음식과 비교하여 미리 정해진 식단 리스트 중 가장 적합한 것을 찾아 제공 2. 가장 적합한 것을 찾을 때는 영양소보다 칼로리를 우선순위로 둔다. |
| 수정된 내용 | 사용자가 먹고 싶은 칼로리를 입력하면 입력한 칼로리보다 낮은 칼로리의 식단 3가지를 표기하여 주고, 하단에 그 보다 더 낮은 식단들을 표기하여 준다. |

## 6.2 요구사항 구현현황

|  |  |  |
| --- | --- | --- |
| 요구사항 ID | 요구사항 명 | 구현 결과(%) |
| SP\_UR\_001 | 회원정보 관리 | 100% |
| SP\_UR\_002 | 회원정보 입력 | 100% |
| SP\_UR\_003 | 회원 목표 칼로리 설정 | 100% |
| SP\_UR\_004 | 섭취한 음식 입력 | 90% |
| SP\_UR\_005 | 칼로리 초과 표시 | 100% |
| SP\_UR\_006 | 식단추천 | 90% |
| SP\_UR\_007 | 추천 운동 제공 | 100% |
| SP\_UR\_008 | 운동 데이터 받아오기 | 100% |
| SP\_UR\_009 | 인바디 데이터 받아오기 | 100% |
| SP\_UR\_010 | 인바디 통계 제공 | 100% |
| SP\_UR\_011 | 섭취한 음식 칼로리 통계 제공 | 100% |
| SP\_UR\_012 | 섭취한 음식 정보 제공 | 100% |
| SP\_UR\_013 | 운동 통계 제공 | 100% |

|  |
| --- |
|  |

# 7. 개발 과정에서의 문제점

1. 음식 종류가 많아질수록 인식하는 정확도 감소 -> 후보 최대 3개까지 나타냄 -> 수기 입력기능 추가
2. 블루투스 연결 안 됨

->

미밴드4와 5를 블루투스로 연동하기 위해선 미밴드와 미핏어플간의 인증키가 필요하다. 이 인증키는 확인해본 바로는 휴대폰을 루팅시켜 휴대폰의 DB에 접근하여 인증키를 읽어오는 것인데, 이는 일반적인 사용자가 하기에는 비현실적인 방법이라 생각한다.

->

미밴드에서 운동을 기록하는 것은 미밴드5만의 기능이고, 나머지 버전에서는 걸음수, 칼로리를 기록하는 기능만 있기 때문에 큰 문제가 되지 않을 것이라 생각한다.

# 8. 향후계획

현재는 카메라로 인식할 수 있는 음식이 60가지이지만 음식 가짓수를 늘려 사용자의 편의성을 증진시킬 계획이다.

음식의 다양성, 식단의 다양성 등을 위하여 데이터베이스에 추가적인 정보들을 업데이트할 예정이다.

1. Google Fit Api를 이용하던것에서 Bluetooth이용으로 변경 [↑](#footnote-ref-2)