

Basic Open-Source Soft Ware, Section 2

# Team Project AI Health Care 보고서

유준혁 32212808

*Department of Mobile Systems Engineering*

김화완 32217234

*Department of Mobile Systems Engineering*

박경빈 32221620

*Department of Computer Engineering*

정승일 32224152

*Department of Statistics and Data Science*

GitHub: <https://github.com/Open-Source-T/Open-Source>

2025-12-01

## Table of Contents

1. 서론
2. 소프트웨어 요구사항 기술 (Requirements)
3. 소프트웨어 설계 (Design)
4. 소프트웨어 구현 (Implementation)
5. 시험 & 디버깅 (Verification)
6. 소프트웨어 유지보수 (Maintenance)
7. 결론
8. 참고 문헌 & 용어 설명

# 1. 서론

## A. 개발 목적

본 프로젝트는 인공지능 기반 사용자 맞춤 건강 관리 서비스를 개발하는 프로젝트다. 사용자가 음식 사진을 업로드하면, 인공지능이 이를 분석하여 칼로리와 영양성분 정보를 제공하고, 사용자 맞춤 건강 코칭을 제공한다. 개발 목적에 대한 더 자세한 사항은 제안서를 참고하기를 바란다.

# 2. 소프트웨어 요구사항 기술 (Requirements)

## A. 사용자 시나리오

1	회원 가입을 하고 로그인한다.
2	음식 사진을 업로드한다.
3	음식의 칼로리와 영양성분 정보, 맞춤 건강 코칭을 확인, 저장, 조회한다.

## B. 기능적 요구사항

회원 관리
회원 가입, 로그인, 로그아웃 기능을 제공해야 한다.
사용자 정보 저장 기능을 제공해야 한다.
음식 사진 분석
사용자가 사진을 업로드하면 칼로리와 영양성분 정보를 시각화 하여 제공해야 한다.
건강 코칭
사용자 정보, 칼로리와 영양성분 정보 기반 건강 코칭을 제공해야 한다.
식단 기록
분석 결과와 사용자 맞춤 건강 코칭을 저장하고 조회하는 기능을 제공해야 한다.

## C. 비기능적 요구사항

성능
서버의 응답 속도와 확장성을 보장해야 한다.
보안
데이터 보호와 접근 제어를 보장해야 한다.
신뢰성
인공지능 모델의 정확성과 서버의 안정성을 보장해야 한다.
확장성
UI/UX 와 분석 결과의 시각화가 정상적으로 동작해야 한다.

소프트웨어 요구사항에 대한 더 자세한 사항은 제안서를 참고하기를 바란다.

### 3. 소프트웨어 설계 (Design)

#### A. 아키텍처 설계

본 프로젝트는 Django 를 사용하기 때문에 MVT (Model View Template) 구조로 모듈화하였고 아키텍처 설계를 따로 하여 변경하지 않는다.

#### B. UI (User Interface) & UE (User Experience) 설계

디자인 컨셉
글래스모피즘
카드와 상단 바에 반투명한 배경을 적용하여 깊이감 있는 연출을 적용한다.
색상 팔레트
신뢰감을 주는 파랑과 남색 계열을 버튼 색깔로 사용하고, 눈이 편안한 회색 계열을 배경과 텍스트에 사용함으로써 정보의 가독성을 향상한다. 에러는 붉은색, 성공은 초록색, 영양소는 각기 다른 고유색으로 지정하여 구분이 용이하도록 한다.
레이아웃
반응형 웹
데스크탑 뿐만이 아니라 모바일 환경에서도 입력 폼과 결과 화면이 최적화되어 보이도록 설계한다.
카드형 UI
모든 주요 콘텐츠를 둑근 모서리 카드 안에 배치하여 시선이 자연스럽게 중앙으로 집중될 수 있도록 유도한다.
데이터 시각화
영양 성분 Bar
단순히 숫자만 보여주는 것이 아닌, 애니메이션이 포함된 가로 막대 그래프를 통하여 칼로리와 영양소 비율을 직관적이고 가독성 높게 파악할 수 있도록 설계한다.

### C. 데이터베이스 설계

UserInfo			
		PK	Primary
string	username		사용자 아이디
string	password		비밀번호
string	email		이메일 주
string	first_name		이
string	last_name		성
int	age		나이
string	sex		성별 (M/F,
int	height		키 (cm,
int	weight		몸무게 (kg, Nullable)
text	description		추가 설명
datetime	date_joined		가입
boolean	is_active		계정 활성 상
boolean	is_staff		관리자 페이지 접근 권한
boolean	is_superuser		최고 관리자 권한

데이터베이스는 위와 같이 설계하였다.

## 4. 소프트웨어 구현 (Implementation)

### A. 개발 일정

단계	기간	주요 작업
인공지능 모델 개발	6~8 주차	데이터셋 수집, 모델 학습 코드 작성, 모델 학습, 모델 평가
프론트엔드 개발	8~10 주차	React 기반 프론트엔드 시험
백엔드 개발	10~12 주차	서버 코드 작성, 서버 시험
시험 및 디버깅 및 기타	12~14 주차	요구사항 검증, 운영, 보고서 작성, 발표

개발 일정은 위와 같다. 개발 일정에 대한 더 자세한 사항은 제안서를 참고하기를 바란다.

### B. 역할 분담

구분	항목	유준혁	김화완	박경빈	정승일
인공지능 모델 개발	데이터셋 수집	참여	참여	참여	
	모델 학습 코드 작성	참여	참여		
	모델 학습	참여	참여		
	모델 평가	참여	참여	참여	
프론트엔드 개발	HTML 작성	참여		참여	참여
	CSS 작성	참여		참여	참여
	React TypeScript 작성		참여		
	프론트엔드 시험	참여	참여		
백엔드 개발	서버 코드 작성	참여			
	서버 시험	참여	참여		
시험 및 디버깅 및 기타	요구사항 검증	참여			
	운영	참여			
	보고서 작성	참여	참여	참여	참여
	발표				

본 프로젝트는 단순히 개발에만 초점을 두는 것이 아니라, 오픈 소스 개발 문화와 전체 개발 파이프라인과 학습에 중점을 두었다. 이를 통해 모든 팀원의 협업 능력을 강화하고, 폭넓은 역량을 습득하는 것을 목표로 하였다. 따라서 모든 팀원이 전 과정을 함께 수행하는 방식으로 진행하였으며 구체적인 참여도는 위와 같다.

## C. 개발 환경 & 도구

Operating System	
Amazon Linux 2023	
Kernel	
Linux 6.1.156	
IDE (Integrated Development Environment) & Editor	
Vim 9.1	
Programming language	
Python 3.9.24, HTML, CSS, TypeScript(React)	
Framework & Library	
accelerate	PyTorch 또는 Transformer 기반 AI 모델을 여러 개의 GPU 와 파라미터 설정을 통해 쉽게 학습할 수 있게 하는 유ти 라이브러리
django	백 엔드의 웹 서버 구현을 위한 파이썬 프레임워크
pillow	이미지 처리를 위한 파이썬 라이브러리
React	사용자 친화 UI & UX 구성을 위한 프론트 엔드 프레임워크
torch	딥러닝 모델 생성 & 학습 프레임워크
transformers	Hugging Face 에서 Transformer 기반 모델을 불러오고 사용하기 위한 라이브러리
ultralytics	YOLO 모델 전용 학습 & 추론 프레임워크
Build tool	
pip 21.3.1, npm	
Data Base	
SQLite3	

## D. 모듈별 기능 설명

인공지능 관련
YOLO(You Only Look Once, 객체 탐지와 분류를 동시에 실행하는 모델) best.pt: yolov8n.pt 사전학습 모델을 AI HUB 의 공공 음식 데이터셋(30GB)을 사용하여 가중치를 설정하고 파인튜닝하여 음식 객체 탐지에 최적화한 모델
Gemini 2.5 Flash-lite gemini_client.py: 백 엔드에서 Google AI Studio API 를 통해 Google Gemini API 와 통신하여 사용자 맞춤형 영양 코칭 정보를 생성하는 스크립트
프론트 엔드 관련
React App.tsx: 로그인 여부에 따라 접근 권한을 제어하며 페이지 간의 이동을 관리하기 위한 파일 api.ts: API 를 통해 사용자의 입력에 대한 서버의 응답을 해석하고 화면에 반영하기 위한 파일
백 엔드 관련
Python Django manage.py: Django 프로젝트의 진입점으로 마이그레이션과 서버 실행 등의 관리 명령을 실행하기 위한 스크립트 models.py: UserInfo(AbstractUser) 모델을 통해 나이, 성별, 신장, 체중, 코칭 기록을 저장하는 DB 구조를 정의하는 파일 views.py: 회원 가입, 로그인, 로그아웃, 이미지 업로드, 기록 조회를 처리하고, YOLO 모델의 가중치와 Gemini API 를 연결하여 음식 사진을 분석하고 사용자 맞춤 코칭을 생성하고 이를 반환하는 핵심 로직 urls.py: 클라이언트의 API 요청 경로(URL)을 views.py 의 적절한 함수와 연결해주는 라우팅 설정 파일 settings.py: 프로젝트와 앱의 연결, 데이터베이스 연결, 커스텀 유저 모델 지정, 보안 키 관리 등 프로젝트의 모든 환경 설정을 총괄하는 스크립트

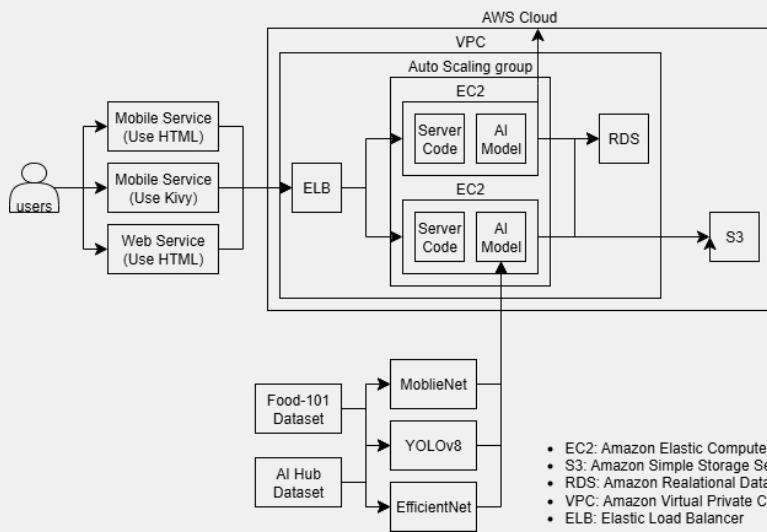
## 5. 시험 & 디버깅 (Verification)

시험 목록	입력	출력	상태
서버의 응답 속도	각 GET, POST Request	서버의 Response	Pass
서버의 확장성	여러 사용자(4 명)의 접속	정상적 HTML 제공	Pass
데이터 보호, 접근 제어	없음	없음	Pass
인공지능 모델의 정확성	테스트 데이터셋	모델 분석 결과 그래프	Pass
서비스의 안정성	여러 번 서비스 이용	서비스 정상 동작	Pass
UI/UX 정상 동작 여부	브라우저 기반 테스트	정상적인 UI 동작	Pass
시각화 정상 동작 여부	브라우저 기반 테스트	정상적인 시각 효과	Pass

서버의 응답 속도
본 프로젝트의 비기능적 요구사항에서 서버의 응답 속도는 5 초 이내를 목표로 하였다. 서버의 응답 속도는 Python 의 time 라이브러리를 이용하여 측정하였다. 각 GET, POST Request 에 대한 서버의 Response 는 대부분 5 초 이내 이루어지는 데 성공하였지만, 음식 업로드 후, 분석 기록을 얻기 까지에는 1 분 이상의 시간이 걸렸다. 이는 음식을 분석하는 YOLO 모델과 음식의 칼로리와 영양 성분 정보를 분석하고 사용자 맞춤 건강 코칭을 제공하는 LLM 모델의 실행 시간이 길기 때문이다. 이는 AWS 의 VM 인 EC2 의 유형을 T2.micro 에서 m5.large 로 변경함으로써 해결하였다.
서버의 확장성
본 프로젝트의 비기능적 요구사항에서 다수의 사용자가 동시에 접속하더라도 서버는 안정적으로 동작해야 함을 요구한다. AWS 는 Auto Scaling 서비스를 제공한다. Auto Scaling 은 사용자의 수가 늘 경우 AWS 의 VM 인 EC2 를 증가시키는 방식으로 확장성을 보장한다. 또한 AWS 는 ELB (Elastic Load Balancing)라는 로드 밸런서를 제공한다. ELB 는 네트워크 트래픽을 분산시키는 방식으로 확장성을 보장한다. ELB 와 EC2 쉽게 적용 가능한 클라우드 컴퓨팅 서비스이기 때문에 언제든지 적용 가능하다. 본 프로젝트의 사용자 맞춤 건강 코칭 제공 서비스는 사용자가 극히 적기 때문에 ELB 와 Auto Scaling 을 적용하지 않았지만, 사용자가 늘 경우 언제든지 적용 가능하다.

## 데이터 보호, 접근 제어

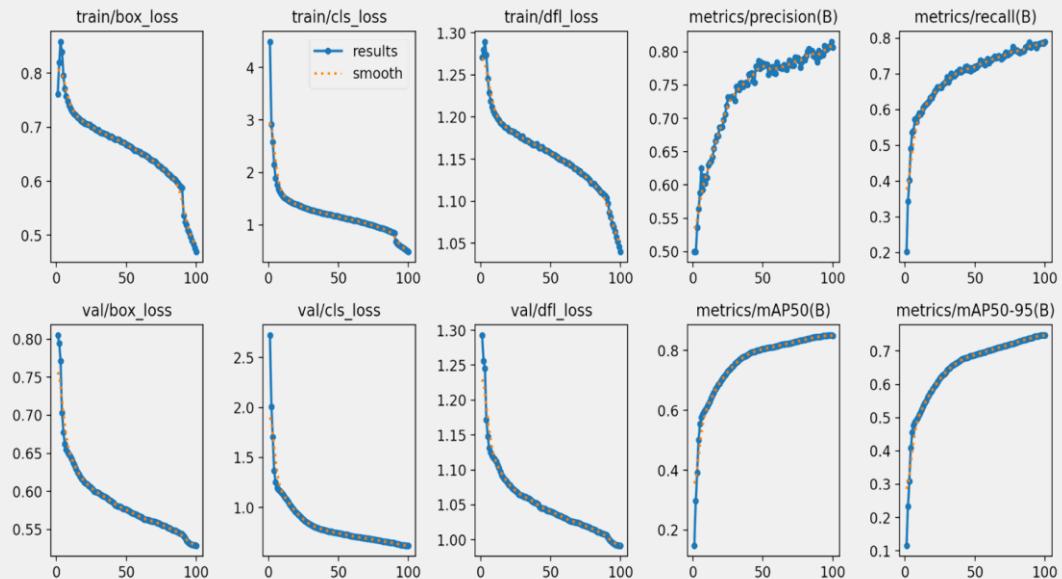
### AI Health Care UML



본 프로젝트의 비기능적 요구사항에서 정보의 안전한 저장을 요구한다. 처음에는 이를 위해서 위와 같은 구조의 서버를 구축하고자 하였다. AWS Cloud의 VPC(Virtual Private Cloud)는 여러 개의 Subnet을 통해 가상 네트워크 환경을 제공한다. 그리고 VPC 외부 서비스이고 IAM(Identity and Access Management)을 통해 안전한 접근이 가능한 S3(Simple Storage Service)를 활용하여 보안성을 높이고자 하였다. 그러나 학생용 AWS 계정의 자원 제한으로 인해 일부 IAM 및 RDS 설정에 제약이 있었으며, 이에 따라 Django의 Abstract User 라이브러리를 사용하는 방식으로 보안성을 강화하였다.

## 인공지능 모델의 정확성

본 프로젝트의 비기능적 요구사항에서 모델은 80% 이상의 정확도를 달성하는 것을 목표로 하였다. 이를 위해서 epochs, batch 등 여러 가지 옵션을 변경해가며 모델을 평가하였다. 학습 데이터셋은 공공 데이터셋 허브(AI HUB)에서 30GB 분량의 음식 사진 데이터셋을 사용하였다. 결측치 혹은 라벨데이터가 일치하지 않는 파일을 전처리하고, 학습용과 검증용 데이터셋으로 8:2 비율로 분리한 뒤, Geforce RTX 4080 GPU를 사용하여 로컬 환경에서 총 27시간 동안 학습을 진행하였다. 사전학습 모델은 yolov8n.pt를 사용하였다.



위 그래프에서 box\_loss, cls\_loss, dfl\_loss는 손실값 변화, mAP@0.5, mAP@0.5:0.95는 모델의 정확도, Precision, Recall은 모델의 정밀도와 재현율의 변화를 나타낸다. 손실값이 줄고, 정확도가 80% 이상임을 알 수 있다.

## 서비스의 안정성

본 프로젝트의 비기능적 요구사항의 서비스 안정성이란 서비스가 제기능을 하지 못하고, 중단되거나, 실패하거나, 오작동하지는 않는 가에 대한 평가이다. 본 프로젝트는 여러 번의 시험 결과 서비스가 중단되거나, 실패하거나, 오작동하는 경우는 없었다.

## UI & UX 정상 동작 여부

코드를 검토하고 브라우저 기반 테스트 결과, 사용자가 가입, 로그인, 사진 업로드, 결과 확인으로 이어지는 시나리오를 정상적으로 수행하는데 있어 논리적인 결함은 발견되지 않았고, UI와 UX가 의도한 대로 매끄럽게 동작하는 것을 확인하였다.

## 시각화 정상 동작 여부

본 프로젝트의 비기능적 요구사항의 시각화 정상 동작 여부란 음식의 칼로리와 영양 성분 정보가 시각 자료로 제공되는 가에 대한 평가이다. 브라우저 기반 테스트 결과, React로 구현한 여러 시각화 동작은 모두 정상적으로 동작하였으며, 단순한 텍스트 나열이 아닌 애니메이션이 포함된 직관적인 그래프 형태로 구현이 되어있어 사용자 경험을 향상시켰다.

## 6. 소프트웨어 유지보수 (Maintenance)

모듈화 설계 & 확장성
기능을 각각의 모듈로 분리하였다. 단일 책임 원칙 (SRP, Single Responsibility Principle)을 준수하여 코드 수정 시 다른 모듈에 대한 영향을 최소화했다.
에러 처리
API 호출 및 데이터 처리 시 발생할 수 있는 예외 상황을 try-except 구문으로 정리하였다. 서버는 적절한 HTTP 코드와 명확한 에러 메시지를 반환하고 클라이언트는 이를 사용자에게 UI로 피드백 하는 것을 구현하여 시스템의 안정성을 확보했다.
코드 가독성
모든 모듈과 함수에 대한 주석을 작성하여 코드 이해도를 높였다.

## 7. 결론

현대 사회에서 건강 관리를 위한 균형 잡힌 식단이 점점 더 중요해지고 있다. 특히 고혈압, 당뇨, 비만 등 식습관과 관련된 질환이 증가한다는 점에서 식단 관리의 필요성은 매우 크다고 할 수 있다. 본 서비스는 기존 서비스의 단점인 사용자가 음식의 이름, 칼로리, 영양성분 정보 등을 일일이 입력해야 하는 점과, 기록만 하며 사용자 맞춤 코칭을 제공하지 못하는 점을 개선하였다. 이를 통해 사용자의 건강 증진에 기여하고, 궁극적으로는 사용자의 자기 주도적 식단 관리를 유도하는 효과가 있을 것이다. 한편 교육적 관점에서 모든 팀원들이 실무와 매우 유사한 학습 환경을 경험함으로써 개발자로 성장하는데 필요한 핵심 역량을 다질 수 있을 것으로 기대되며, 오픈 소스 관점에서는 프로젝트의 개발과 운영 과정을 문서와 함께 공개함으로써 향후 기능이 고도화되고 서비스가 확장되고 다양한 플랫폼을 지원하는 프로젝트로 발전할 수 있을 것으로 기대된다.

## 8. 참고 문헌 & 용어 설명

### A. 참고 문헌

인공지능 모델 개발 관련
Food-101 데이터셋 사용 방법 ( <a href="https://bulky.kr/1n4eMhf">https://bulky.kr/1n4eMhf</a> )
ML5 ( <a href="https://ml5js.org">https://ml5js.org</a> )
TensorFlow.js ( <a href="https://bulky.kr/BpFqH4P">https://bulky.kr/BpFqH4P</a> )
TensorFlow 사용 방법 ( <a href="https://bulky.kr/BpFqH8q">https://bulky.kr/BpFqH8q</a> )
YOLOv8 메인 문서 ( <a href="https://docs.ultralytics.com">https://docs.ultralytics.com</a> )
한국 AI HUB 공공 데이터셋 사용법 ( <a href="https://www.aihub.or.kr/">https://www.aihub.or.kr/</a> )
프론트 엔드 개발 관련
React 공식 문서 ( <a href="https://ko.react.dev">https://ko.react.dev</a> )
React 튜토리얼 ( <a href="#">React Tutorial</a> )
백 엔드 개발 관련
Gemini API 공식 문서 ( <a href="#">Gemini API 문서 살펴보기</a> )
구글 Open Food Facts ( <a href="https://bulky.kr/7x7AhHp">https://bulky.kr/7x7AhHp</a> )
식품 의약품 안전처의 식품 영양성분 데이터베이스 ( <a href="https://bulky.kr/15PcRSP">https://bulky.kr/15PcRSP</a> )

### B. 용어 설명

Decoupling
시스템 구성 요소(프론트/백)를 독립시켜 유지보수 및 확장을 쉽게 만드는 설계 방식
Glass Morphism
투명한 유리 배경과 흐림 처리를 이용해 세련된 깊이감을 주는 UI 디자인 기법
RESTful API
React 와 서버가 데이터를 주고받기 위한 통신 규칙
YOLO
You Only Look One 의 약자로, 실시간 객체 탐지 모델이며, 사용자가 업로드한 사진에서 음식 클래스를 빠르게 분류시키는데 사용