

**Diet In Body Application**

**과목명 소프트웨어공학**

**담당교수님 김상철교수님**

**201904458 이준용**

**202001574 박찬근**

**201901940 송경민**

목차

1. 개요

1.1. 프로젝트 개요

1.2. 프로젝트 산출물

1.3. 정의, 약어

2. 자원 및 일정 예측

2.1. 자원

2.1.1. 인력

2.1.2. 비용

2.2. 일정

3. 조직 구성 및 인력 배치

3.1. 조직 구성/담당 업무

3.2. 직무기술

4. WBS

5. 기술 관리 방법

6. 표준 및 개발 절차

6.1. 개발 방법론

7. 검토회의

7.1. 검토 회의 일정

7.2. 검토 회의 진행 방법

8. 개발환경

9. 성능 시험 방법

10. 문서화

11. 유지보수

12. 설치, 인수

13. 참고 문헌 및 부록

**1. 개요**

**1.1. 프로젝트 개요**

올바른 식생활은 우리의 건강한 삶을 위해 필수적인 요소입니다. 올바른 식생활은 우리의 신체적, 정신적 건강을 유지하는 데 도움을 주며, 영양 정보들은 우리가 올바른 식단을 선택하고 필요한 영양소를 충분히 섭취할 수 있도록 도와줍니다. 우리는 식단 다이어리 작성, 영양 정보 라벨 확인 등의 실천 방법을 통해 건강한 식생활을 실천할 수 있지만 여러 현실적인 이유로 혼자 지속하기 힘든 경우가 많습니다. 그래서 이를 좀 더 쉽게 할 수 있도록 도움을 드릴 수 있는 어플을 만들고자 저희는 “Diet In Body Application”을 계획하게 되었습니다. 우리가 삶에서 쉽게 볼 수 있는 많은 음식의 영양 정보들이 데이터화 되어 있어 각 음식의 영양정보와 하루 섭취한 칼로리 및 영양소를 매일매일 쉽게 체크할 수 있고, 맞춤 식단을 추천해주는 기능도 있으며 체중 추적 기능을 통해 체중변화를 확인해 볼 수 있습니다. 또한, 달마다 제공하는 Body 리포트를 받아보며 그에 따라 목표 설정할 수 있도록 하는 기능도 포함시킬 계획입니다.

**1.2. 프로젝트 산출물**

- 식단 다이어리: 매일 먹은 음식의 칼로리와 영양정보를 기록하고 관리합니다.

- 다이어트 달력: 하루 칼로리 섭취량과 소모량을 확인할 수 있습니다.

- Body 리포트: 지난 달에 섭취한 음식들의 영양소를 분석해서 적절한 비율로 먹었는지 체크하고 체중, 체지방 등 정한 목표의 달성률이 표시되며 이에 대한 피드백을 제공합니다.

**1.3. 정의, 약어**

- IF (Intermittent Fasting): 간헐적 단식

- OMAD (One Meal a Day): 1일 1식

- ADF (Alternate day fasting): 하루 걸러 하루만 식사하는 방식

**2. 자원 및 일정 예측**

**2.1. 자원**

**2.1.1. 인력**

팀 명 : 15조

팀 구성원 3명

팀장 : 이준용

팀원: 이준용, 박찬근, 송경민

|  |  |
| --- | --- |
| 팀장 | 이준용 |
| 개요, 프로젝트 산출물 | 박찬근 |
| 일정, WBS(앱/웹 구현) | 이준용 |
| 지원팀 | 송경민 |

2.1.2. 비용

|  |  |
| --- | --- |
| 물품명 | 비용 |
| 크롤링 서버 구축(MYSQL, AWS EC2) |  |
| 대용량 데이터를 위한 클라우드 저장소 | 0USD(경기도에서 IT창업 무료 클라우드 받기) |

**2.2. 일정**



텍스트, 스크린샷, 번호, 라인이(가) 표시된 사진

자동 생성된 설명

스크린샷, 텍스트, 라인, 그래프이(가) 표시된 사진

자동 생성된 설명

**3. 조직 구성 및 인력 배치**

**3.1. 조직 구성/담당 업무**

① 조직 구성

**팀 장**

**이 준 용**

**개발사**

**박 찬 근**

**지 원**

**송 경 민**

② 담당 업무

**<팀장>**

- 프로젝트 총괄, 관리 및 감독

- 주요사안의 최종 의사결정

- 개발 지원 및 산출물 검토

**<개발사, 지원팀>**

- 프로젝트 관리, 추진 및 계획

- 설계 및 구축

- 앱/웹 설계, 구현

**3.2. 직무기술**

①모델 학습: 회원의 칼로리를 계산하고 다이어트 또는 벌크업의 목표를 설정함으로써 회원이 어떤 상태인지 학습을 시켜줍니다.

②App/Web 구현: 학습된 모델을 바탕으로 통계자료와 그래프를 보여줌과 동시에 식단을 추천해주는 것으로 구현됩니다.

③ DataBase : 사용자의 다이어트 또는 벌크업을 위한 균형잡힌 식단을 저장하여 관리하는 DB를 구현합니다.

**4. WBS**

텍스트, 스크린샷, 폰트, 도표이(가) 표시된 사진

자동 생성된 설명

Figma로 WBS 작성 -> [웹으로 자세히 보기](https://www.figma.com/file/wJG4Jg4BN4kaELD5lFpUtu/%EC%86%8C%ED%94%84%ED%8A%B8%EC%9B%A8%EC%96%B4%EA%B3%B5%ED%95%99-%ED%8C%80%ED%94%84%EB%A1%9C%EC%A0%9D%ED%8A%B8-WBS?type=whiteboard&node-id=0%3A1&t=zMn1I40HkBFMbwcu-1)

**5. 기술 관리 방법**

**계속적인 요구 변경:** 사용자나 클라이언트의 요구사항이 계속해서 변경되는 경우를 말합니다. 이런 상황을 관리하기 위해선 최대 변경 상한선을 정하거나, 정보 은닉, 점증적 개발 등의 방법을 사용할 수 있습니다.

**외부 모양의 빈약:** 사용자 인터페이스(UI) 등 외부적인 측면에서 부족함이 있는 경우를 의미합니다. 벤치마킹, 검사, 대조 확인, 성숙도 분석 등으로 이를 보완할 수 있습니다.

**외부 기능의 빈약:** 제공하는 기능 자체가 부족하거나 불충분한 경우입니다. 이럴 때는 사전 검증을 통해 필요성을 확인하고, 설계 경연이나 팀 작업으로 보완하는 방법이 있습니다.

실시간 성능의 빈약: 시스템의 성능(응답 시간 등)이 충분하지 않은 경우입니다. 시뮬레이션, 벤치마킹, 모델링과 같은 방식으로 성능을 개선하거나 프로토타입을 만들어서 테스트하고 튜닝하는 방법 등이 있습니다.

**기술적 취약:** 사용하는 기술 자체에 문제가 있는 경우입니다. 예를 들어 오래된 기술일수록 지원 중단될 확률이 높아지므로 새로운 기술로 전환해야 할 필요가 생길 수 있습니다. 그러므로 현재 사용 중인 기술에 대해 분석하고 비용-수익 분석도 진행하여 적절한 결정을 내릴 수 있게 합니다.

|  |  |
| --- | --- |
| 위험요인 | 위험관리기법 |
| 잘못된 UI 구현 | 시나리오 작성, 태스크 분석, 사용자 분류(기능, 스타일, 업무), 프로토타이핑 |
| 잘못되 기능 구현 | 사용자 지침서의 조기 작성, 조직분석, 직능 분석, 사용자 회람, 프로토타이핑 |
| 비현실적 일정계획 및 예산 | 원가 분석, 보다 세부적인 비용, 일정 예측 |
| 성능 부족 | 모델링, 벤치마킹, 성능 튜닝. 시뮬레이션, 프로토타이핑 |
| 하드웨어 부하 과중 | 비용-수익 분석, 기술 분석, 프로토타이핑 |
| 지속적인 요구사항 변경 | 최대 변경 상한선, 다음 release까지 변경 연기, 점증적 개발 |
| 인력부족 | 유능한 인력모집 또는 사전 확보, 팀 구성, 교육 수행 |

**6. 표준 및 개발 절차**

**6.1. 개발 방법론**

텍스트, 폰트, 스크린샷, 로고이(가) 표시된 사진

자동 생성된 설명

"주문 ➪ 디자인 ➪ 기능구현 ➪ 테스팅 ➪ 배포" x ∞

짧고 반복적인 과정으로 빠르고 유연하며 수정도 쉽다는 장점이 있다.

과거와는 비교조차 하기 어려울 정도로 시장과 고객 등 변화가 빨라지고 한치 앞을 예상하기 어려운 환경 때문에, 최근에는 거의 모든 IT업계에서는 애자일 방식을 선호하고 있고, 우리 팀도 위와 같은 내용과 마찬가지로 Agile(에자일)방식을 선택했다.

|  |  |
| --- | --- |
|  | Agile |
| 장점 | 소규모 팀이 여러 과제를 각각 할당 받아 처리 가능 |
| 짧고 반복적인 스프린트로 구성 -> 빠르게 결함 식별 및 수정가능 |
| 개발과정이 빠르고 유연 |
| 개발 과정 중에 신속하게 제품 변경 가능(짧은 반복과정 거치기 때문) |
| 단점 | 수많은 변경사항이 있을 수 있으므로 번거로움 발생 가능 |
| 빠른 반복 작업에 익숙한 숙련된 사람이 필요 |
| 적합한 조직 | 크고 복잡한 회사들이 프로세스를 간소화함으로써 변화에 신속대응 하고자 할 때  (ex, AT&T, IBM, 마이크로 소프트) |
| 고품질의 결과물과 지속적 개선에 초점을 맞춘 조직 |
| 결과물에 대해 빠른 피드백이 필요한 팀 |

**7. 검토회의**

**7.1. 검토 회의 일정**

① 정기회의 : 매주 화요일 18시 ZOOM 화상회의

② 긴급회의 : 프로젝트 수행 중 의견 또는 문제 발생시 , 그룹채팅을 통해 수시로 회의 진행

**7.2. 검토 회의 진행 방법**

① 팀장이 현재 진행 상황과 일정을 브리핑

② 한 주간 각자 수행한 업무상태를 점검하고 확인

③ 수정이 필요하거나 부족한 부분은 팀원들의 힘과 의견을 모아 업무 수행

④ 새로운 업무 분담 후 회의 종료

**8. 개발환경**

① PyCharm-selenium: 웹 크롤링을 위해 파이썬에서 제공하는 웹스크래핑 프로그램인 selenium을 활용하여 대량의 데이터를 수집할 예정.

② MYSQL: 효율적인 데이터 수집을 자동으로 할 수 있도록 MYSQL DB에 저장

③ Visual Studio Code(VSCODE): Responsive Web을 개발하기 위해 React-Redux가 잘 활용할 수 있는 개발환경

④ Postman, Filezilla, Xshell7: 서버에서 Web으로 데이터 출력이 잘 되는지 확인과 함께 EC2서버에 front코드와 크롤링 코드 전송하기 위한 여러 툴

**9. 성능 시험 방법**

① 각 팀원들의 업무수행 상태를 점검하고 부족한 부분은 검토회의 중 팀원들의 의견을 모아 함께 업무를 수행한다.

**10. 문서화**

① 검토 회의 시 지원팀은 회의내용을 요약하여 작성

② 검토 회의 중 팀원 개인간의 업무수행 상태를 함께 점검 후 문제점, 부족한 부분을 기록하여 서로 공유

**11. 유지보수**

① 크롤링 서버: 자동으로 웹 크롤링시 서버다운 될 수 있으므로 수시로 서버 작동되고 있는지 확인 필요

② 데이터 출력: WEB에 크롤링 데이터 미출력 버그 발생시 fix

**12. 설치**

App store에서 Diet In Body App(식단 인바디 앱) 검색 및 다운로드

Google play store에서 Diet In Body App(식단 인바디 앱) 검색 및 다운로드

Galaxy Store에서 Diet In Body App(식단 인바디 앱) 검색 및 다운로드

**13. 참고 문헌 및 부록**

[**https://ko.redux.js.org/introduction/getting-started/**](https://ko.redux.js.org/introduction/getting-started/)

[**https://yamoo9.github.io/axios/guide/**](https://yamoo9.github.io/axios/guide/)

[**https://formik.org/docs/api/formik**](https://formik.org/docs/api/formik)

[**https://tanstack.com/**](https://tanstack.com/)