# 요구사항 정의서

## 1. 프로젝트 개요

프로젝트 이름: 핏퀘스트(FitQuest)  
목적: 운동/식단 기록 관리 및 운동하는 사람들의 커뮤니티 활성화  
주요 목표:  
- 사용자들의 운동 및 식단 기록을 편리하게 관리할 수 있는 기능 제공  
- 사용자가 목표를 설정하고, 성취도를 확인하며, 동기부여를 받을 수 있는 시스템 구현  
- 운동 커뮤니티 활성화를 통해 서로 정보를 공유하고 격려할 수 있는 플랫폼 제공

## 2. 사용자 플로우

1. 메인 페이지  
2. 로그인 여부 확인  
 - 로그인을 하지 않은 상태:  
 - 회원 가입 여부 확인  
 - 회원 가입을 하지 않은 상태 => 회원 가입 페이지로 안내  
 - 회원 가입이 되어 있는 상태 => 로그인 페이지로 안내 => 이후 절차는 아래와 동일  
 - 로그인을 한 상태:  
 - 커뮤니티:  
 - 본인의 게시글 등록  
 - 타인의 게시글 열람  
 - 본인의 게시글 수정  
 - 본인의 게시글 삭제  
 - 게시글 정렬(등록순, 최신순, 추천순, 조회순)  
 - 게시글 검색(제목, 내용, 제목+내용, 카테고리, 닉네임)  
 - '질문' 카테고리의 등록된지 24시간이 지나고 댓글이 1개 이하로 달린 게시글에 챗봇 댓글 등록(테스트 중)  
 - 기록:  
 - 운동과 식단 중 선택  
 - 운동 기록:  
 - 당일 운동 기록 등록  
 - 당일 운동 기록 조회  
 - 식단 기록:  
 - 당일 식단 기록 등록  
 - 당일 식단 기록 조회  
 - 마이페이지:  
 - 회원 정보 열람  
 - 회원 정보 수정  
 - 본인 작성 글 조회  
 - 본인 작성 댓글 조회  
 - 본인 전체 운동 기록 조회  
 - 본인 전체 식단 기록 조회  
 - 회원 탈퇴  
 - 로그아웃

## 3. 백엔드 엔드포인트 목록

### 3.1 AdminController

- /admin/users: 전체 사용자 조회  
- /admin/users/{userId}/posts: 특정 사용자의 게시글 조회  
- /admin/users/{nickname}/comments: 특정 사용자의 댓글 조회  
- /admin/users/{userId}/exercises: 특정 사용자의 운동 기록 조회  
- /admin/users/{userId}/diets: 특정 사용자의 식단 기록 조회

### 3.2 AuthController

- /auth/login: 로그인  
- /auth/logout: 로그아웃

### 3.3 CommentBotController

- /comment-bot/process: 챗봇 댓글 처리 트리거  
- /comment-bot/{postId}/add-comment: 특정 게시글에 챗봇 댓글 추가

### 3.4 CommentController

- /comments: 댓글 등록  
- /comments/post/{postId}: 특정 게시글의 모든 댓글 조회  
- /comments/{commentId}: 댓글 수정  
- /comments/{commentId}: 댓글 삭제  
- /comments/post/{postId}/best: 게시글의 베스트 댓글 조회  
- /comments/{commentId}/recommend: 댓글 추천  
- /comments/user/{userId}: 사용자가 작성한 댓글 조회

### 3.5 DietController

- /diets: 당일 식단 기록 등록  
- /diets/today/{userId}: 당일 식단 기록 조회  
- /diets/{dietId}: 식단 기록 수정  
- /diets/{dietId}: 식단 기록 삭제  
- /diets/{userId}/distribution: 당일 식단 유형 비중 계산

### 3.6 ExerciseController

- /exercises: 운동 기록 등록  
- /exercises/today/{userId}: 당일 운동 기록 조회  
- /exercises/{exerciseId}: 운동 기록 수정  
- /exercises/{exerciseId}: 운동 기록 삭제  
- /exercises/{userId}/month: 당월 운동 날짜 조회  
- /exercises/{userId}/distribution: 당일 운동 유형 비중 계산

### 3.7 PostController

- /posts: 게시글 등록  
- /posts: 전체 게시글 조회  
- /posts/search: 조건 검색 게시글 조회  
- /posts/{postId}: 특정 게시글 조회 및 조회 수 증가  
- /posts/{postId}: 게시글 수정  
- /posts/{postId}: 게시글 삭제  
- /posts/{postId}/recommend: 게시글 추천  
- /posts/user/{userId}: 사용자가 작성한 게시글 조회

### 3.8 UserController

- /users/signup: 회원가입  
- /users/{id}: 사용자 조회 (ID 기준)  
- /users/email/{email}: 이메일 기반 사용자 조회  
- /users/nickname/{nickname}: 사용자 조회 (닉네임 기준)  
- /users/{id}: 사용자 정보 수정  
- /users/{id}/exercises: 사용자 운동 기록 조회  
- /users/{id}/diets: 사용자 식단 기록 조회  
- /users/{id}: 회원 탈퇴

## 4. 기술 스택

백엔드: Spring Boot, Gradle, MyBatis  
프론트엔드: Vue.js Composition API (Pinia, Axios)  
데이터베이스: MySQL  
코드 관리: GitHub  
IDE: IntelliJ IDEA Ultimate, VSCode

- 3계층 구조:  
 - 프론트엔드: Vue.js를 기반으로 사용자가 시스템과 상호작용  
 - 애플리케이션 레이어: Spring Boot에서 비즈니스 로직과 데이터 처리 담당  
 - 데이터베이스 레이어: MySQL에서 데이터를 영구적으로 저장 및 관리

## 6. 데이터 구조

### 6.1 사용자(User)

- 테이블: User  
- 주요 필드:  
 - user\_id: 사용자 ID (Primary Key)  
 - email: 사용자 이메일 (Unique, Not Null)  
 - password: 비밀번호 (암호화된 형태로 저장)  
 - nickname: 닉네임 (Unique, Not Null)  
 - height, weight, age: 사용자 신체 정보  
 - is\_admin: 관리자 여부 (Boolean)

### 6.2 운동 기록(Exercise)

- 테이블: Exercise  
- 주요 필드:  
 - exercise\_id: 운동 기록 ID (Primary Key)  
 - user\_id: 사용자 ID (Foreign Key -> User)  
 - exercise\_name: 운동 이름  
 - exercise\_type: 운동 유형 (근력, 유산소, 코어)  
 - exercise\_set, repetitions, weight: 근력 운동 관련 정보  
 - distance, time, speed: 유산소 운동 관련 정보  
 - indoor\_outdoor: 실내/실외 여부  
 - created\_at: 운동 기록 등록 날짜

### 6.3 식단 기록(Diet)

- 테이블: Diet  
- 주요 필드:  
 - diet\_id: 식단 기록 ID (Primary Key)  
 - user\_id: 사용자 ID (Foreign Key -> User)  
 - food\_name: 음식 이름  
 - calories: 칼로리  
 - meal\_type: 식사 유형 (아침, 점심, 저녁)  
 - created\_at: 식단 기록 등록 날짜

### 6.4 게시글(Post)

- 테이블: Post  
- 주요 필드:  
 - post\_id: 게시글 ID (Primary Key)  
 - user\_id: 사용자 ID (Foreign Key -> User)  
 - category: 게시글 유형 (자유, 질문, 모임)  
 - title, content: 제목과 내용  
 - views, likes: 조회 수 및 추천 수  
 - attachment\_path: 첨부 파일 경로  
 - created\_at: 게시글 작성 날짜

### 6.5 댓글(Comment)

- 테이블: Comment  
- 주요 필드:  
 - comment\_id: 댓글 ID (Primary Key)  
 - post\_id: 게시글 ID (Foreign Key -> Post)  
 - user\_id: 사용자 ID (Foreign Key -> User)  
 - content: 댓글 내용  
 - likes: 추천 수  
 - created\_at: 댓글 작성 날짜

## 7. 비기능 요구사항

1. 보안:  
- 비밀번호는 암호화하여 저장 (e.g., bcrypt 또는 Argon2 사용)  
- JWT(Json Web Token) 기반 인증 및 권한 관리  
- HTTPS를 통해 데이터 전송 암호화  
  
2. 성능:  
- 대량의 게시글 및 댓글 조회 시 페이징 처리 지원  
- Redis와 같은 캐싱 도구 사용 고려 (인기 게시글 등)  
  
3. 확장성:  
- 운동 유형과 식사 기록의 확장 가능성을 염두에 둔 ENUM 설계  
- 서비스가 확장되더라도 데이터베이스와 API가 높은 확장성을 유지  
  
4. 사용성:  
- 모바일 최적화된 UI 제공  
- 직관적인 데이터 입력 및 기록 관리 기능