

# HappyMeal 설계 문서

## 서비스 상세 요구사항 정의서

### 1. 사용자 관리 (User Management) & 인증 (Authentication)

#### 1.1 비로그인 사용자

##### 1.1.1. 로그인 및 세션

- **구글 OAuth 로그인:**

- 사용자는 구글 계정을 통해 서비스에 로그인할 수 있다. (백엔드 Spring Security OAuth2 Client 및 JWT 발급 로직 통해 처리)
- 구글 로그인 실패 또는 사용자 취소 시, 프론트엔드는 다시 로그인 페이지로 리디렉션한다.
- 최초 구글 로그인 시, 구글에서 제공하는 사용자 정보(구글 ID, 이메일, 닉네임, 프로필 사진 URL)를 기반으로 서비스에 자동 회원가입 처리된다.
- 자동 회원가입 시, 초기 닉네임은 사용자의 구글 프로필 닉네임으로 설정된다.
- 서비스 DB 에는 사용자의 구글 고유 ID, 이메일, (초기)닉네임, 프로필 사진 URL 이 저장된다.

- **로그인 상태 유지:**

- Refresh Token 을 사용하여 로그인 상태를 유지한다.
- 사용자가 별도로 "로그인 상태 유지"를 선택하는 옵션은 없으며, 기본적으로 로그인 상태가 유지된다.
- Access Token 유효 기간: 5 시간 (18,000,000ms).
- Refresh Token 유효 기간: 7 일 (604,800,000ms).
- (추후 정의 필요: Access Token 만료 시 Refresh Token 을 사용한 자동 갱신 로직 - 예: `POST /api/auth/refresh`)

- (추후 정의 필요: 로그아웃 시 Refresh Token 처리 방안 - 현재 클라이언트 측 삭제. 서버 측 무효화 고려)
- (추후 정의 필요: Refresh Token 저장 방식 및 탈취 대응책)

### 1.1.2. 조회 기능

- **공개 게시글 목록 조회 (필터링 및 정렬 포함):**

- **API 엔드포인트:** GET  
/api/boards?categoryId={categoryId}&sortBy={sortBy}&page={page}&size={size}
- 기본 정렬 순서: 최신순 (sortBy="latest").
- 요청 파라미터를 통해 인기순(sortBy="popular") 정렬을 선택할 수 있다.
- 페이징: 한 페이지에 기본 10 개의 게시글을 반환한다 (page 기본값 0, size 기본값 10).
- 목록에 표시되는 정보: 게시글 ID, 제목, 작성자 닉네임, 작성일, 조회수, 좋아요 수, 댓글 수, 내용 일부 요약, 썸네일 이미지.

- **게시글 상세 내용 조회:**

- **API 엔드포인트:** GET /api/boards/{boardId}
- 조회수 증가: 게시글 상세 보기 API 요청 시 서버에서 해당 게시글의 조회수를 1 증가시킨다. 동일 사용자의 단기간 내 반복 조회 시에도 계속 증가시킨다.
- 존재하지 않거나 삭제된 boardId 로 API 직접 접근 시, EntityNotFoundException (또는 이에 상응하는 커스텀 예외)을 발생시키고, "게시글을 찾을 수 없습니다. ID: [boardId]" 메시지와 함께 HTTP 404 Not Found 응답을 반환한다. (게시글 삭제 시 목록에서는 즉시 제거됨)

- **음식 검색 및 영양정보 조회:**

- **API 엔드포인트:** GET  
/api/foods/search?name={name}&sortBy={sortBy}&page={page}&size={size}
- 검색 방식: 음식명에 대한 부분 일치 검색을 지원한다.
- 데이터 출처: 공공기관 제공 음식 데이터베이스를 활용한다.
- 제공 영양 정보 항목: 칼로리(kcal), 탄수화물(g), 단백질(g), 지방(g), 당류(g).
- 영양 정보 기준 단위: 100g 당 정보를 기준으로 표시한다.

- **전체 음식 정보 조회 (페이징 및 정렬):**
  - **API 엔드포인트:** GET /api/foods?sortBy={sortBy}&page={page}&size={size}
- **특정 음식 정보 조회 (ID 기반):**
  - **API 엔드포인트:** GET /api/foods/{foodId}
- **카테고리별 음식 추천 조회:**
  - **API 엔드포인트:** GET /api/foods/recommendations?category={category}
- **게시글 필터링 조회:** (위의 GET /api/boards 엔드포인트가 categoryId 파라미터를 통해 이 기능 수행)
  - 카테고리별로 게시글을 필터링하여 조회할 수 있다.
  - 특정 카테고리를 선택한 후, 해당 카테고리 내의 게시글들을 최신순 또는 인기순으로 정렬하여 조회할 수 있다.
  - 전체 카테고리(categoryId 미지정 시)에 대해서도 최신순 또는 인기순으로 정렬하여 조회할 수 있다.
- **특정 게시글의 댓글/대댓글(1 단계) 목록 조회:**
  - **API 엔드포인트:** GET /api/boards/{boardId}/comments
- **특정 게시글의 좋아요 상태 및 총 좋아요 수 조회:**
  - **API 엔드포인트:** GET /api/boards/{boardId}/like
- **특정 게시글의 총 좋아요 수 조회:**
  - **API 엔드포인트:** GET /api/boards/{boardId}/likes/count

## 1.2 로그인 사용자

### 1.2.1. 개인정보 관리

- **마이페이지 사용자 정보 조회:**
  - **API 엔드포인트:** GET /api/mypages/profile
- **프로필 정보 수정 (닉네임, 프로필 이미지):**
  - **API 엔드포인트:** PUT /api/mypages/profile
  - **닉네임 변경:**
    - 사용자는 자신의 닉네임을 변경할 수 있다.
    - 닉네임 유효성 검사: 중복 검사, 길이 제한 (DTO 정의에 따름), 문자 제한 (DTO 정의에 따름).
    - 변경 성공 시: 프론트엔드에서 별도의 성공 메시지 없이 수정 모드가 종료되고 변경된 닉네임이 화면에 즉시 반영된다.

(백엔드는 HTTP 200 OK 와 함께 업데이트된 사용자 정보(UserDto) 반환)

- 변경 실패 시: 프론트엔드에서 "닉네임 수정에 실패했습니다." 또는 구체적인 실패 사유를 Alert 메시지로 표시한다.  
(백엔드는 GlobalExceptionHandler 를 통해 적절한 HTTP 상태 코드와 오류 메시지 반환)
- 닉네임 변경 횟수 제한 또는 쿨다운 기간: 없음.

○ **프로필 이미지 업로드/변경:**

- 사용자는 자신의 프로필 이미지를 업로드하거나 변경할 수 있다.
- 허용 파일 형식: 프론트엔드에서 accept="image/\*" 속성을 사용하여 이미지 파일만 선택하도록 유도한다. 백엔드에서는 실제 파일 타입 및 용량 제한을 검증해야 한다. (구체적인 형식 및 크기 제한 추후 정의)
- 기본 프로필 이미지: 제공된다. 사용자가 이미지를 삭제하여 기본 이미지로 돌아가는 기능은 현재 구현되어 있지 않다.
- 이미지 저장 방식: 서버의 로컬 파일 시스템에 저장한다. (저장 경로 및 파일명 정책 정의 필요)
- 업로드 실패 시: 백엔드에서 오류 발생 시 해당 오류를 로그로 기록하고, 프론트엔드는 상위 컴포넌트에서 에러를 처리하여 사용자에게 알림을 표시한다.

### 1.2.2. 식단 기록 관리

• **일별 식단 기록:**

○ **식단 기록 추가:**

- **API 엔드포인트:** POST /api/meallogs
- 요청 바디: MealLogDto (실제로는 MealLogCreateRequestDto 사용 권장. foodId, quantity, mealDate, mealType, imgUrl 포함)
- 사용자는 각 식사를 아침/점심/저녁/간식 유형으로 구분하여 기록할 수 있다.

○ **음식 검색을 통한 식단 추가:** (GET /api/foods/search 활용 후, 선택된 foodId 와 사용자 입력 섭취량 등으로 POST /api/meallogs 호출)

- 섭취량 입력: 사용자가 직접 수량(예: 0.5, 1, 1.5 등)을 입력하면, 해당 음식의 기본 제공 기준량과 사용자 입력 수량을 곱하여 최종 섭취량 및 영양 정보를 계산하여 기록한다.
  - 검색된 음식이 없을 경우: 현재는 직접 추가 기록 불가. "1.2.5 음식 추가 요청" 기능을 통해 음식 등록을 요청할 수 있다.
- **식단 수정/삭제:**
  - **식단 기록 수정:**
    - **API 엔드포인트:** PUT /api/meallogs/{logId}
    - 요청 바디: MealLogRequestDto (foodId, quantity, mealType, imgUrl, mealDate 는 수정 불가)
    - 본인 식단 기록만 수정 가능 (백엔드에서 작성자 일치 여부 검증)
  - **식단 기록 삭제:**
    - **API 엔드포인트:** DELETE /api/meallogs/{logId}
    - 본인 식단 기록만 삭제 가능 (백엔드에서 작성자 일치 여부 검증)
  - 삭제 시 확인 절차: 없음.
- **식단 기록 조회:**
  - **사용자의 전체 식단 기록 조회:**
    - **API 엔드포인트:** GET /api/meallogs (본인 기록만, 페이징 및 정렬 추후 정의 필요)
  - **특정 날짜 식단 기록 조회:**
    - **API 엔드포인트:** GET /api/meallogs?date={YYYY-MM-DD} (본인 기록만)
  - **식단 기록 상세 조회:**
    - **API 엔드포인트:** GET /api/meallogs/{logId} (본인 기록만)
- **식단 캘린더 (날짜별 조회):** (프론트엔드 기능, GET /api/meallogs?date={YYYY-MM-DD} API 활용)
  - 사용자는 캘린더 UI 를 통해 특정 날짜를 선택할 수 있다.
  - 특정 날짜 선택 시, 해당 날짜에 기록된 모든 식단 기록을 상세히 확인할 수 있다.

- 캘린더 각 날짜 셀에는 별도의 요약 정보가 표시되지 않고, 날짜 선택 기능만 제공한다.
- 월별 식단 기록 조회: (식단 캘린더의 날짜 선택 기능으로 대체)
- 영양소 통계:
  - 일별 통계:
    - **API 엔드포인트:** GET /api/meallogs/stats?date={YYYY-MM-DD}
    - 사용자가 선택한 특정 날짜의 총 섭취 칼로리 및 탄수화물, 단백질, 지방 각각의 합계를 제공한다.
  - 주별 통계 (일별 데이터 리스트):
    - **API 엔드포인트:** GET /api/meallogs/stats/weekly?date={YYYY-MM-DD}
    - 사용자가 선택한 날짜를 마지막 날(end day)로 하여, 이전 6 일을 포함한 총 7 일간의 **각 일별** 영양소 통계(탄수화물, 단백질, 지방 각각의 합계 및 총 칼로리) 리스트를 제공한다.

### 1.2.3. 게시판 기능

- 게시글 관리:
  - 게시글 작성:
    - **API 엔드포인트:** POST /api/boards
    - 요청 바디: BoardCreateRequestDto (제목(최대 255 자), 내용(단순 텍스트), 이미지 여러 개 첨부 가능 - 용량 제한 없음)
  - 게시글 수정:
    - **API 엔드포인트:** PUT /api/boards/{boardId}
    - 본인이 작성한 게시글만 수정 가능. 요청 바디: BoardUpdateRequestDto.
  - 게시글 삭제:
    - **API 엔드포인트:** DELETE /api/boards/{boardId}
    - 본인이 작성한 게시글만 삭제 가능.
  - 내가 작성한 게시글 목록 조회:
    - **API 엔드포인트:** GET /api/mypages/posts?sortBy={sortBy}&page={page}&size={size}
    - 페이징 처리되며, 기본 정렬 순서는 최신순이다.
- 댓글 기능:
  - 댓글/대댓글(1 단계) 작성:

- **API 엔드포인트:** POST /api/boards/{boardId}/comments
- 요청 바디: CommentRequestDto (content, parentCommentId - 대댓글 작성 시 부모 댓글 ID)
- 댓글 수정:
  - (API 엔드포인트 정의 필요: 예: PUT /api/comments/{commentId} - CommentController 에 구현)
  - 본인이 작성한 댓글만 수정 가능.
- 댓글 삭제:
  - **API 엔드포인트:** DELETE /api/comments/{boardId}/{commentId} (현재 CommentController 구현. /api/comments/{commentId}로 단순화 고려)
  - 본인이 작성한 댓글만 삭제 가능.
- 대댓글(답글) 기능 상세:
  - 최상위 댓글에 대한 대댓글 작성을 지원한다. (1 단계 깊이)
  - 대댓글에 대한 대댓글(2 단계 이상 깊이)은 지원하지 않는다.
  - UI: 최상위 댓글 아래에 해당 대댓글들이 표시된다.  
대댓글에는 추가적인 '댓글 작성하기' 버튼이 표시되지 않는다.
- 댓글/대댓글 작성 시 알림 기능: 없음.
- 내가 작성한 댓글 목록 조회:
  - **API 엔드포인트:** GET  
/api/mypages/comments?sortBy={sortBy}&page={page}&size={size}
  - 표시 정보: 댓글 내용, 작성 일자, 댓글이 달린 원본 게시물 정보(예: 게시물 제목 또는 링크).
  - 페이징 처리되며, 기본 정렬 순서는 최신순이다.
- 좋아요 기능:
  - 게시물 좋아요/취소 (토글):
    - **API 엔드포인트:** POST /api/boards/{boardId}/like
    - 사용자당 게시물별 1 회만 가능 (토글 방식).
  - 댓글 좋아요: (현재 명세에 없음)
  - 좋아요한 게시물 목록 조회:
    - **API 엔드포인트:** GET  
/api/mypages/likes?sortBy={sortBy}&page={page}&size={size}

- 페이징 처리되며, 기본 정렬 순서는 최신순이다.

- **검색 및 필터링 (게시글 목록):**

- (제목 검색: GET /api/boards/search/title, 작성자 검색: GET /api/boards/search/author API 활용)
- 게시글 목록에서 제목 또는 내용에 특정 키워드가 포함된 게시글을 검색할 수 있다. (내용 검색은 별도 API 또는 기존 검색 API 확장 필요)
- 게시글 목록에서 특정 작성자(닉네임)가 작성한 게시글을 검색할 수 있다.
- **정렬 기능 (인기순):** '인기순' 정렬 시, 1 차: 좋아요 수(내림차순), 2 차: 조회 수(내림차순), 3 차: 댓글 수(내림차순), 4 차: 최신 작성일(내림차순) 순으로 우선순위를 적용한다. (적용 대상: GET /api/boards 등 목록 조회 API)

#### 1.2.5. 음식 추가 요청

- **사용자 음식 등록 요청 생성:**

- **API 엔드포인트:** POST /api/food-requests
- 요청 바디: FoodRequestDto.Create (음식명, 1 회 제공량(값과 단위 명시 필요), 칼로리, 탄/단/지, 음식 이미지(선택 사항))

- **처리 방식:** 요청된 음식은 관리자의 승인 절차를 거친 후, 승인되면 전체 음식 DB 에 추가되어 모든 사용자가 검색 및 이용할 수 있게 된다.

- **사용자 알림:** 음식 추가 요청의 승인/거절 상태 변경 시 사용자에게 별도의 알림은 현재 제공되지 않는다.

- **요청 상태 확인:**

- **API 엔드포인트 (본인 요청 목록):** GET /api/food-requests/my
- **API 엔드포인트 (특정 요청 상세):** GET /api/food-requests/{foodRequestId} (본인 또는 관리자만)
- 사용자는 '마이페이지 > 음식 등록 요청' 메뉴에서 자신이 요청한 음식들의 처리 상태(대기중/승인/거절)를 확인할 수 있다. 승인 대기 중이거나 거절된 음식은 식단 기록에 사용할 수 없다.



## 1.3 관리자

- **계정 생성 및 권한 부여:** 데이터베이스에서 관리자가 수동으로 특정 사용자에게 `ROLE_ADMIN` 역할을 부여한다.
- **1.3.1. 사용자 관리:** (현재 API 미구현, AdminController 확장 필요)
  - 관리자는 전체 사용자 목록을 조회할 수 있다. (페이징 처리, 검색/필터링 기준 추후 정의)
  - 현재 관리자는 사용자 정보 수정 기능은 가지고 있지 않다. (조회만 가능)
- **1.3.2. 게시판 관리:** (현재 API 미구현, AdminController 확장 필요)
  - 관리자는 전체 게시글 및 댓글 목록을 조회할 수 있다. (페이징 처리, 검색/필터링 기준 추후 정의)
  - 부적절한 게시글/댓글 판단 기준 및 삭제 기능: 현재 정의되어 있지 않다.
  - 콘텐츠 삭제 시 작성자 알림/패널티 정책: 현재 없음.
  - 관리자 활동에 대한 로그(audit log) 기록: 현재 없음.
- **1.3.3. 음식 데이터베이스 관리:**
  - 관리자 직접 음식 정보 추가:
    - **API 엔드포인트:** `POST /api/admin/foods`
  - 관리자 직접 음식 정보 수정:
    - **API 엔드포인트:** `PUT /api/admin/foods/{foodId}`
  - 관리자 직접 음식 정보 삭제:
    - **API 엔드포인트:** `DELETE /api/admin/foods/{foodId}`
  - 사용자 음식 추가 요청 목록 조회 (관리자용):
    - **API 엔드포인트:** `GET /api/food-requests` (관리자 전용)
  - 사용자 음식 추가 요청 상태 변경 (승인/거절 - 관리자용):
    - **API 엔드포인트:** `PATCH /api/food-requests/{foodRequestId}/status`
    - '등록(승인)' 처리 시, 해당 음식 정보가 시스템의 메인 음식 DB에 저장된다.

## 2. 시스템 요구사항

### 2.1. 보안

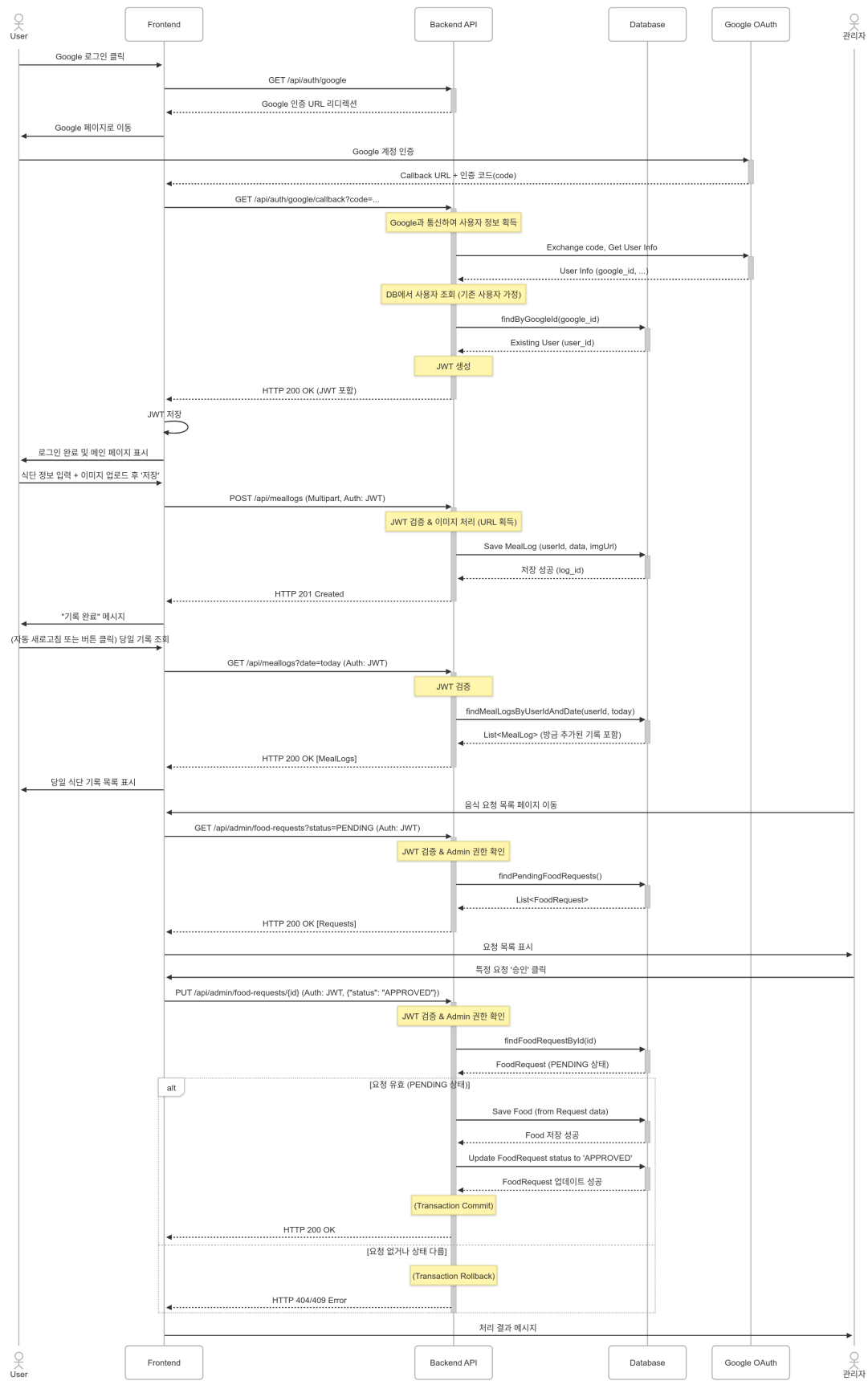
- **사용자 인증:**
  - JWT(JSON Web Token)를 기반으로 인증 및 인가 처리를 수행한다.
  - JWT 포함 정보(Claims): userId, role.
  - **토큰 갱신:** (API 정의 필요: 예 POST /api/auth/refresh) Access Token 만료 시 Refresh Token 을 사용하여 새로운 Access Token 을 발급받는다.
  - **세션 관리:** JWT 를 사용하므로 기본적으로 Stateless 방식이나, Refresh Token 관리를 위한 별도 저장소 및 정책이 필요하다.
- **데이터 보안:**
  - **비밀번호 암호화:** 현재 구글 OAuth 만 사용하므로 서비스 자체 비밀번호는 없으나, 향후 자체 회원가입 기능 추가 시 강력한 단방향 해시 함수(예: bcrypt)를 사용해야 한다.
  - **개인정보 암호화:** (암호화 대상 및 방식 추후 구체화) 전송 계층 암호화(HTTPS)는 기본 적용한다.
- **API 보안:**
  - 입력값 유효성 검증 (DTO 레벨 @Valid, 서비스 레벨).
  - SQL Injection 방지 (MyBatis 사용 시 동적 쿼리 부분 주의, 일반적으로는 PreparedStatement 자동 적용).
  - XSS 방지 (프론트엔드 출력 시 데이터 이스케이핑, 백엔드에서도 필요시 처리).
  - (추후 정의 필요: API 요청 속도 제한, CSRF 방지책)

### 2.2. 성능

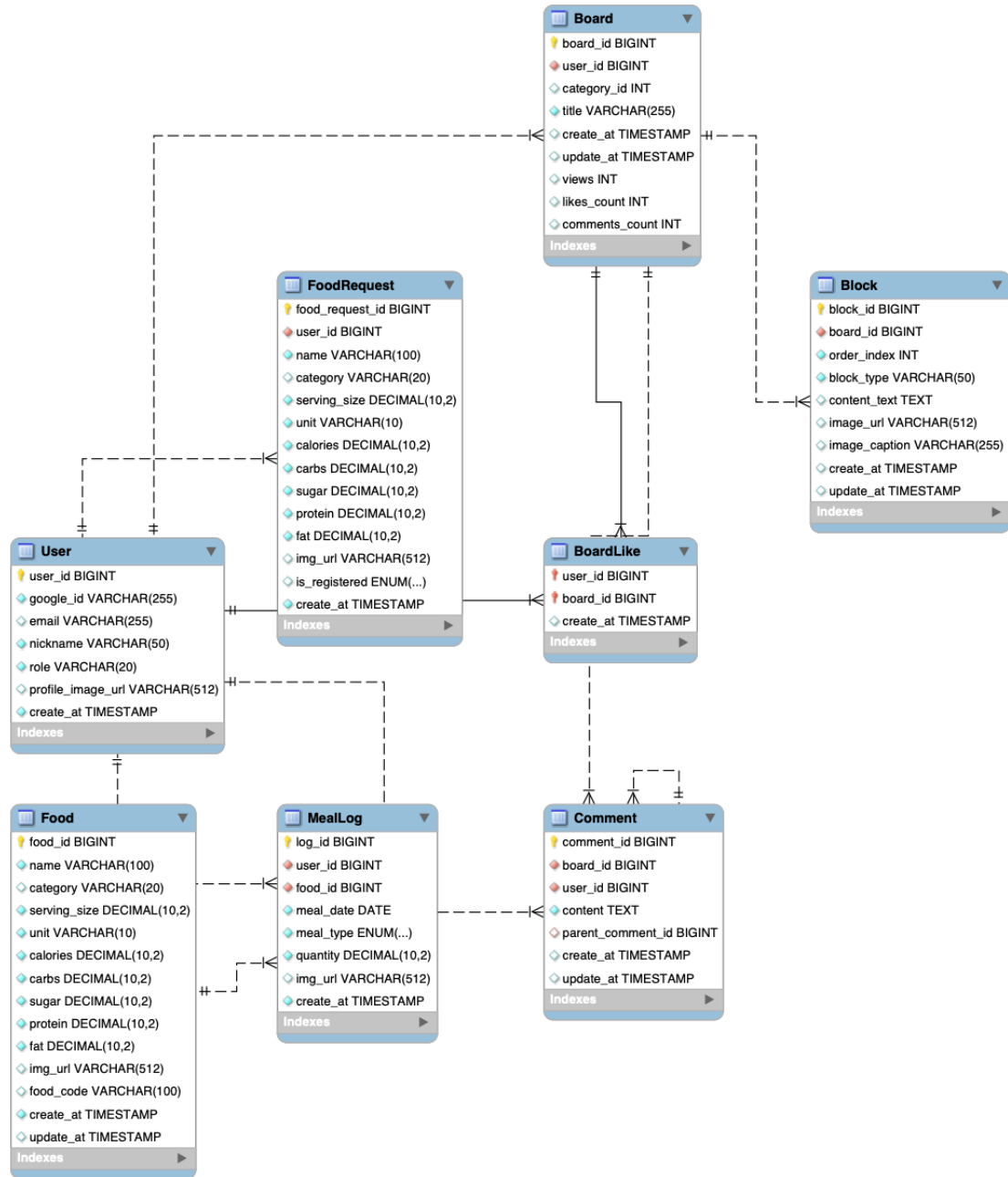
- **Redis 서버 활용 개성 (추천 음식):**
  - **"추천 음식" 기준:** 음식 카테고리(건강식, 벌크업식, 다이어트식, 치팅식)별로 정의된 목표 열량에 가장 부합하는 음식들을 선정하여 제공한다. (목표 열량 계산 로직 및 개인화 여부 추후 정의)
  - **캐시 업데이트 전략:** 매일 새벽 2 시에 스케줄러를 통해 캐시를 업데이트하며, 캐시 데이터는 25 시간 후 만료(무효화)된다.

- **추가 캐싱 대상:** 현재 없음.
- **API 응답 속도 목표:** (추후 정의 필요)
- **동시 사용자 처리:** 캐싱된 추천 음식 조회 요청에 대해서는 동시성 제어 로직이 일부 구현되어 있다.

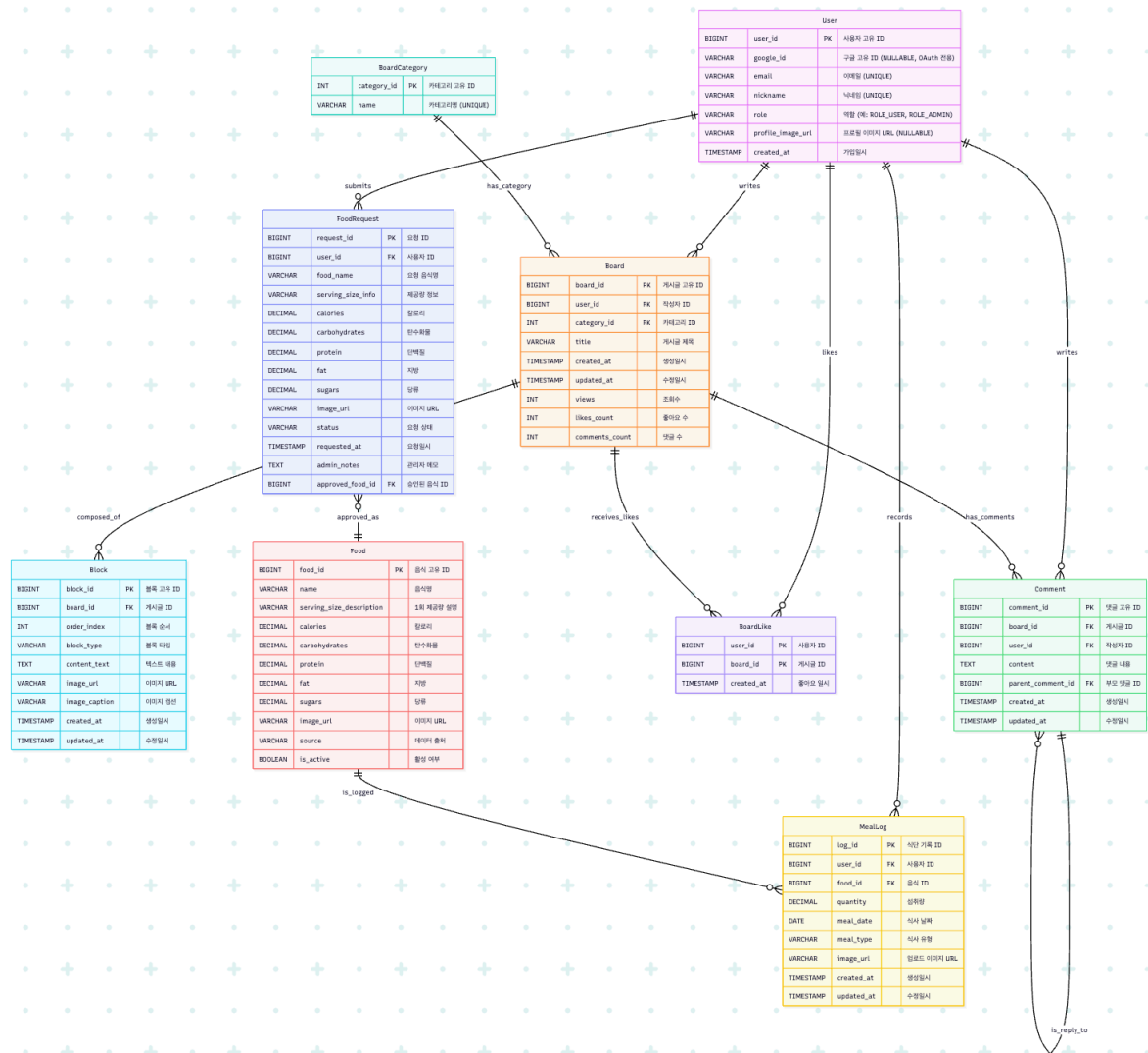
# USECASE 다이어그램



# 클래스 다이어그램



ER 다이어그램



간트차트

작업 번호	대분류	소분류 (기능)	기능	시작일 (2025)	종료일 (2025)	기간 (일)	비고 (의존성, 우선순위)
Phase 1: 기존 BE 기능의 FE 구현 및 신규 핵심 BE (5/12 ~ 5/14, 3일)							
1	Food	음식 검색, 검색/상세 조회 FE	FE	5/12	5/13	2	최우선 FE 작업
2	MealLog	식단 기록 추가 폼, 기록/통계 조회 FE	FE	5/12	5/14	3	최우선 FE 작업, Food 검색과 연동 고려
3	Board	게시글 조회/상세/카테고리별 BE	BE	5/12	5/13	2	신규 핵심 기능 BE 선행
4	Board	게시글 작성 BE (이미지 포함)	BE	5/12	5/13	2	신규 핵심 기능 BE 선행
5	Food	사용자 음식 등록 요청 폼, 목록 조회 FE	FE	5/13	5/14	2	Food 관련 FE 마무리
6	MealLog	식단 상세 조회 BE	BE	5/14	5/14	1	신규 기능
7	Food	음식 카테고리별 추천 BE	BE	5/14	5/14	1	메인페이지용
Phase 2: 신규 핵심 FE 및 추가 기능 BE/FE (5/15 ~ 5/19, 3일 - 5/16(금), 5/19(월))							
8	Board	게시글 조회/상세/카테고리별 FE	FE	5/15	5/16	2	BE 완료 후
9	Board	게시글 작성 FE (이미지 포함)	FE	5/15	5/16	2	BE 완료 후
10	MealLog	식단 상세 조회 FE	FE	5/15	5/15	1	BE 완료 후
11	Food	음식 카테고리별 추천 FE (웹인)	FE	5/16	5/16	1	BE 완료 후
12	Admin	관리자용 음식 등록 요청 목록/상세 변경 FE	FE	5/16	5/16	1	기존 BE의 FE 마무리
13	User	개인정보 조회, 본인 작성 글 조회 BE	BE	5/15	5/16	2	미디어페이지 핵심
14	Login	구글 OAuth 연동 FE (필요시)	FE	5/19	5/19	1	
15	User	개인정보 조회, 본인 작성 글 조회 FE	FE	5/19	5/19	1	BE 완료 후
Phase 3: 추가 기능 및 관리자 기능, 버그 (5/20 ~ 5/21, 2일)							
16	User	개인정보 수정, 본인 글 수정/삭제 BE (선택적)	BE	5/20	5/20	1	시간 부족 시 삭제만 또는 다음으로
17	Admin	사용자 목록/삭제 BE, 음식 CRUD BE (선택적)	BE	5/20	5/21	2	우선순위 가장 낮음
18	User	개인정보 수정, 본인 글 수정/삭제 FE (선택적)	FE	5/21	5/21	1	BE 완료 후, 시간 부족 시 범위 축소
19	Admin	사용자 목록/삭제 FE, 음식 CRUD FE (선택적)	FE	5/21	5/21	1	BE 완료 후, 시간 부족 시 범위 축소
상시/미지각							
20	공통	통합 테스트, 버그 수정, 최종 점검	모두	(상시)	5/21 (1-2)		매일 작업의 시간 할애, 마지막 날은 집중 (실질적으로 부족할 수 있음)

# 화면 설계서

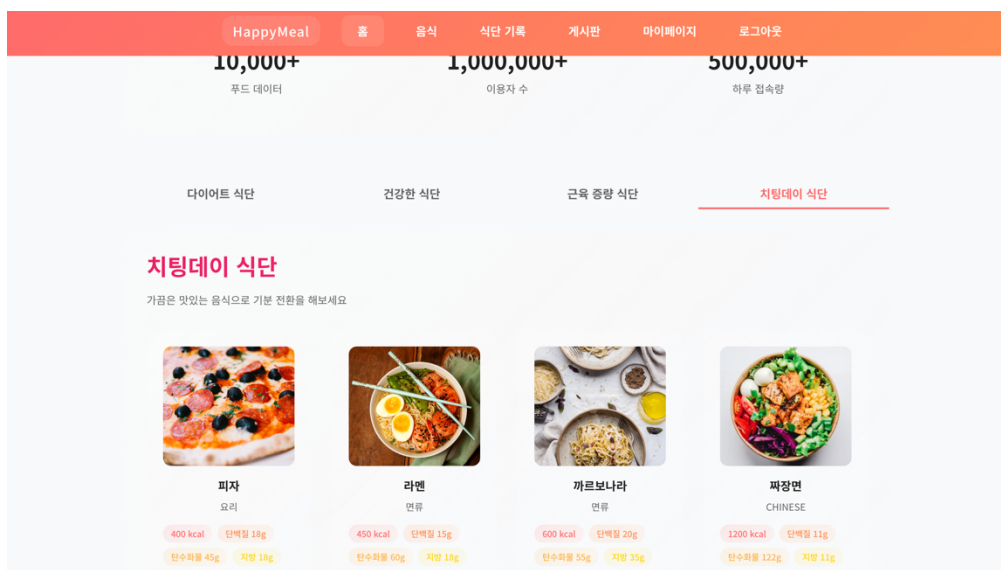
## 1. 전체 네비게이션 구조

- 홈
- 음식
- 식단 기록
- 게시판
- 마이페이지
- 로그인/로그아웃

## 2. 주요 화면별 설계

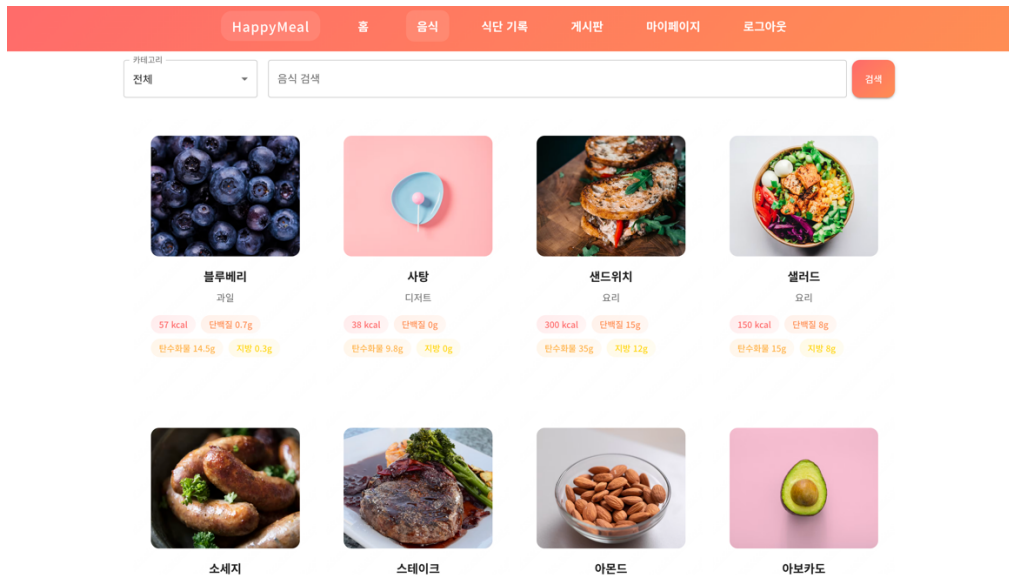
### 2.1 홈 (메인) 화면

- **설명:** 서비스 소개, 주요 통계, 카테고리별 추천 식단, 커뮤니티 미리보기 등으로 구성
- **주요 UI:**
  - 상단 배너(서비스 슬로건)
  - 통계(푸드 데이터, 이용자 수, 접속량)
  - 카테고리별 식단 탭 및 카드
  - 커뮤니티(Health Challenge) 미리보기
- **연동 API:**
  - GET /api/foods/recommendations?category={카테고리}: 추천 식단 조회
- **화면 이미지**



## 2.2 음식 목록 화면

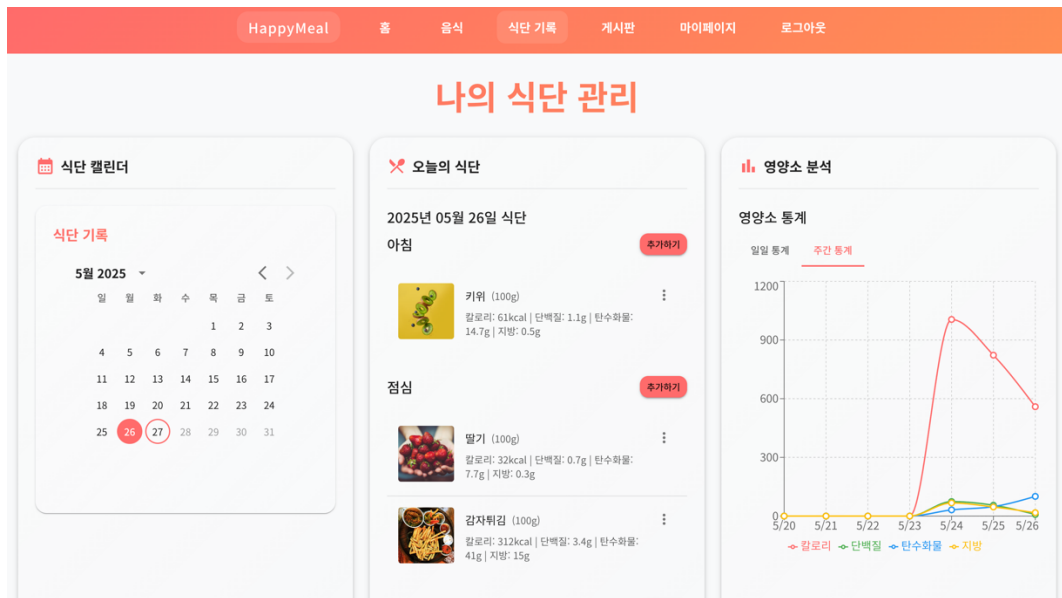
- **설명:** 음식 데이터 전체/카테고리별/검색 결과를 카드형으로 제공
- **주요 UI:**
  - 카테고리 드롭다운
  - 검색창
  - 음식 카드 리스트(영양 정보 포함)
  - 페이지네이션
- **연동 API:**
  - GET /api/foods: 전체 목록
  - GET /api/foods/search?name=검색어: 이름 검색
  - GET /api/foods/{id}: 상세
- **화면 이미지**





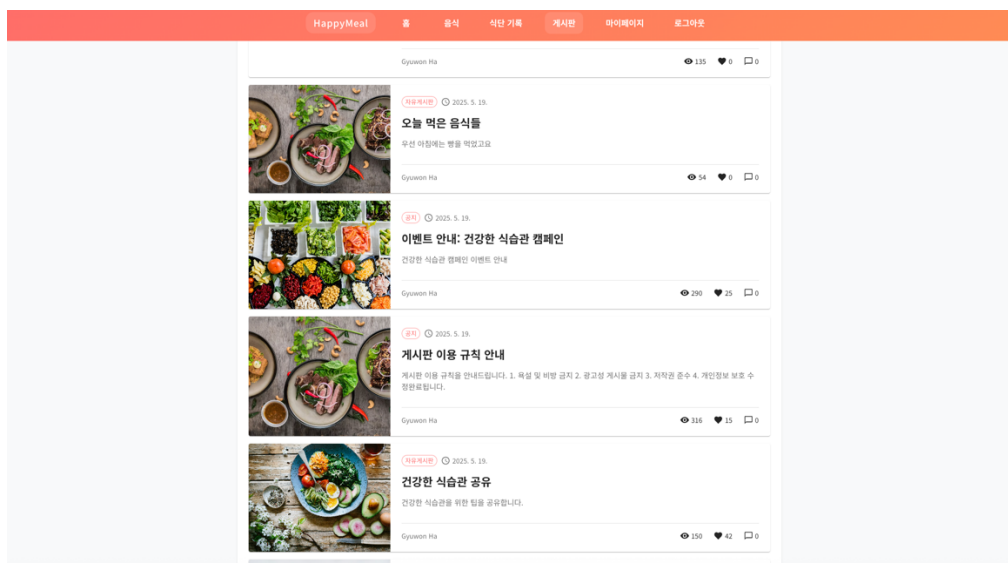
## 2.3 식단 기록 화면

- **설명:** 캘린더 기반 식단 기록, 오늘의 식단, 영양소 분석 제공
- **주요 UI:**
  - 식단 캘린더
  - 오늘의 식단(아침/점심/저녁/간식)
  - 영양소 통계(일일/주간)
- **연동 API:**
  - GET /api/meallogs?date=YYYY-MM-DD: 일별 식단 기록
  - POST /api/meallogs: 식단 추가
  - DELETE /api/meallogs/{id}: 식단 삭제
- **화면 이미지**



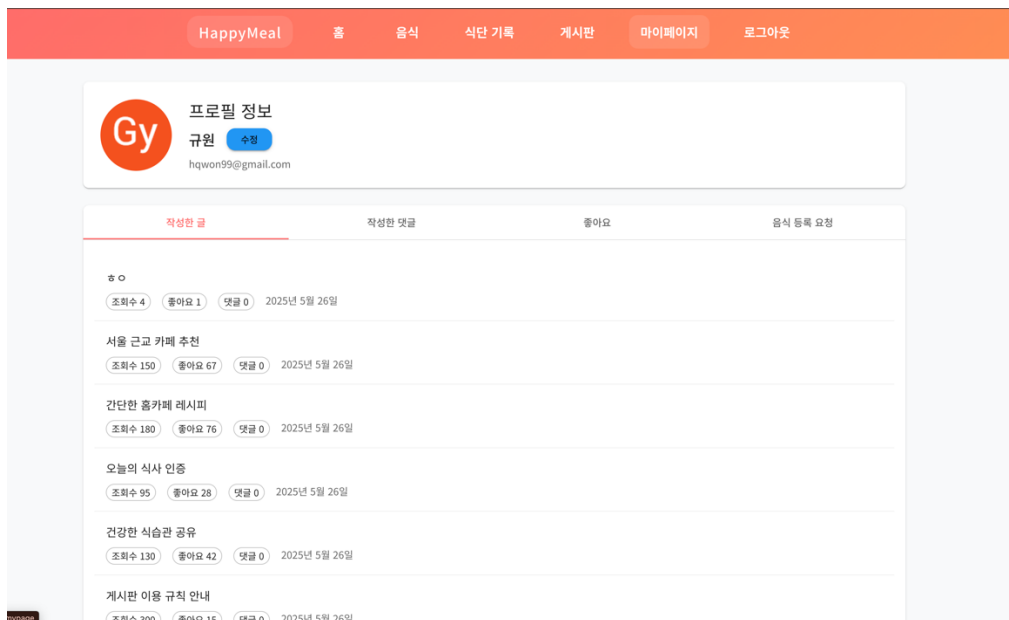
## 2.4 게시판 시스템

- **설명:** 커뮤니티형 게시판, 카테고리별/검색/좋아요/댓글 등 지원
- **주요 UI:**
  - 검색(제목/작성자)
  - 카테고리 탭
  - 게시글 카드(썸네일, 태그, 제목, 날짜, 조회/좋아요/댓글 수)
  - 글쓰기 버튼
  - 페이지네이션
- **연동 API:**
  - GET /api/boards: 게시글 목록
  - GET /api/boards/search/title?title=검색어: 제목 검색
  - GET /api/boards/search/author?nickname=닉네임: 작성자 검색
  - GET /api/boards/{id}: 상세
  - POST /api/boards: 작성
  - PUT /api/boards/{id}: 수정
  - DELETE /api/boards/{id}: 삭제
  - POST /api/boards/{id}/like: 좋아요 토글
  - GET /api/boards/{id}/comments: 댓글 목록
  - POST /api/boards/{id}/comments: 댓글 작성
  - DELETE /api/comments/{boardId}/{commentId}: 댓글 삭제
- **화면 이미지**



## 2.5 마이페이지/프로필

- **설명:** 내 정보, 내가 쓴 글/댓글/좋아요/음식 등록 요청 등 탭별 제공
- **주요 UI:**
  - 프로필 정보(이름, 이메일, 등급)
  - 탭(작성한 글, 댓글, 좋아요, 음식 등록 요청)
  - 각 탭별 리스트
- **연동 API:**
  - GET /api/users/me: 내 정보
  - GET /api/users/me/boards: 내가 쓴 글
  - GET /api/users/me/comments: 내가 쓴 댓글
  - GET /api/users/me/likes: 내가 좋아요한 글
- **화면 이미지**



## 참고 문서

- [Redis 공식 문서](https://redis.io/docs/)
- [OAuth 2.0 공식 문서](https://oauth.net/2/)
- [React 공식 문서](https://react.dev/)
- [Spring Boot 공식 문서](https://docs.spring.io/spring-boot/docs/current/reference/html/)
- [JWT 공식 문서](https://jwt.io/introduction)
- [Java 공식 문서](https://docs.oracle.com/en/java/)