# Functional Design Document (FDD)

**PROJECT:** FULGO OMS

**Module:** Admin site

**Document Version:** 1.0

**Date:** 2025-09-14

**Prepared by:** Hansol (Thor) Choi

## I. Use of Document

이 문서는 FULGO OMS(이하 “프로젝트”)의 기능 설계 문서입니다. 해당 문서의 모든 내용은 SotaTek의 소유이며, 사전 허가 없이 복제, 전송, 사용될 수 없습니다. 문서는 프로젝트 진행에 따라 변경될 수 있으며, 변경 사항은 Section II의 문서 이력에 기록되어야 합니다.

## II. Document Control

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 2025-09-14 | Hansol (Thor) Choi | Initial Version |
| - | - | - | - |

## III. Table of Contents

* I. Use of Document
  1. Document Control
  2. Table of Contents
  3. Introduction
* V. User Site (Use Cases)
  1. Non-functional Requirements
  2. API Contracts
  3. Data Models (summary)
  4. Acceptance Criteria
* X. Appendix (Wireframes / Notes)

## IV. Introduction

### 4.1 Document Purpose

이 FDD는 PROJECT\_NAME의 화면, 기능, 입력/출력, 비즈니스 규칙 및 수용 기준을 명확히 정의하여 개발 및 QA에 필요한 구현 가이드를 제공합니다.

### 4.2 Project Abstract

PROJECT\_NAME은 건강 상담을 위한 AI 기반 서비스입니다. 본 문서는 Admin(관리자) 및 유저 사이트의 핵심 화면과 흐름을 기술합니다.

### 4.3 Intended Audience

* Project Managers
* Business Analysts
* Development Leads
* QA Engineers

### 4.4 Terms and Abbreviations

* BR: Business Rule
* UC: Use Case
* API: Application Programming Interface

### 4.5 Scope of Services

서비스 범위에 포함되는 항목: 프로젝트 관리, UI/UX 디자인, 기능 설계 문서(FDD), 품질 보증, 교육, 고객 지원(필요 시).

### 4.6 Scope of Work

본 문서는 다음 화면/기능의 상세 정의를 포함합니다: 회원 인증(가입/로그인/로그아웃), 온보딩, 상품/분류(예시) 관리, 주문 관리, 몰 연동 설정 등.

## V. User Site — Use Cases (요약)

### UC1: Sign up (회원가입)

* ID: UC1
* Use case name: Sign up
* Author: Ha.do
* Description: 사용자가 계정을 생성하는 흐름입니다.

#### Trigger

사용자가 ‘시작하기/Start’ 버튼을 클릭합니다.

#### Pre-conditions

N/A

#### Post-conditions

사용자가 정상적으로 가입되어 앱에 접근할 수 있습니다.

#### Business Rules

* BR-01: 인증코드는 전송 후 5분(300초) 후 만료됩니다.

#### Flow / Details

1. Welcome → 약관 동의 → 본인인증(휴대폰) → 개인정보 입력(이름, 생년월일, 성별) → 가입 완료

#### 화면별 요소 / Validation (요약)

* 약관 동의: 전체동의 체크박스가 개별 체크박스와 연동됩니다.
* 생년월일: YYYY.MM.DD 포맷, 연령 제한(1930 ~ 현재연도-14)
* 휴대폰: 숫자만 입력, 11자리
* 인증번호: 숫자 6자리, 타이머 3:00

### UC2: Log in (로그인)

* ID: UC2
* Use case name: Log in
* Author: Ha.do
* Description: 기존 계정으로 앱에 로그인합니다.

#### Trigger

Welcome 화면에서 ‘로그인하기’ 버튼 클릭

#### Pre-conditions

사용자가 이미 회원 가입을 완료한 상태

#### Post-conditions

사용자가 로그인되어 홈 화면으로 이동합니다.

#### Business Rules

* BR-01: 인증코드는 전송 후 5분 내 사용해야 합니다.

#### Flow / Details

* 전화번호 입력 → 인증 요청 → 인증번호 입력/확인 → 로그인

### UC3: Log out (로그아웃)

* ID: UC3
* Use case name: Log out
* Description: 사용자가 세션을 종료합니다.

#### Trigger

설정 메뉴에서 ‘로그아웃’ 버튼 클릭

#### Post-conditions

사용자가 로그아웃되고 Welcome 화면으로 리다이렉트됩니다.

## VI. Non-functional Requirements (요약)

* 반응성: 모바일/데스크톱에서 응답성 보장
* 보안: 인증 토큰은 안전하게 저장(예: HttpOnly cookie 또는 secure storage)
* 접근성: 기본적인 키보드 내비게이션 및 대체 텍스트 제공

## VII. API Contracts (요약)

* POST /api/auth/send-code { phone } → 200 { success }
* POST /api/auth/verify-code { phone, code } → 200 { token }
* GET /api/products?query=&category=&brand=&page=&sort= → 200 { products[], total }

## VIII. Data Models (summary)

* Product (id, code, name, brandId, price, variants[], classificationPath[])
* User (id, phone, name, dob, gender)

## IX. Acceptance Criteria

* 필수 항목: 인증 코드 타이머 동작(3:00), 필수 필드 검증, 전화번호/인증 흐름 정상 동작

## X. Appendix

* Wireframes / Figma links:

### Notes

원문에 포함된 상세한 화면 단위 항목(각 버튼, 라벨, 동작 규칙)은 개별 스크린 FDD(예: signup.md, login.md)로 분리하여 보관하는 것을 권장합니다. 전체 변환 및 페이퍼 문서 업로드를 원하시면 알려주세요.

# FDD: 상품 목록 (Products List)

## 1. 기본 정보

* 화면 이름 (한국어): 상품 목록
* 화면 경로(Route): /products
* 메뉴 ID: products-list
* 담당자: TBD
* 우선순위: P0

## 2. 목표 및 설명

* 목적: 상품을 검색, 필터링, 정렬하고 다중 선택하여 일괄 작업(엑스포트, 일괄 상태 변경 등)을 수행할 수 있도록 합니다.

## 3. 주요 사용자 플로우

* 플로우 1: 검색어 입력 → 필터 선택(분류/브랜드) → 결과 확인 → 항목 선택 → 엑스포트
* 플로우 2: 개별 상품 클릭 → 상세 페이지로 이동

## 4. UI 구성 요소

* 상단: 검색 입력, 분류 필터(HierarchicalSelect), 브랜드/상태 드롭다운, 엑스포트 버튼
* 주요: 상품 테이블(이미지, 이름, 코드, 분류, 브랜드, 가격, 재고, 액션)
* 모달: 대량 작업 확인 모달(삭제/비활성화 등)

## 5. 입력값 / 출력값 (Contract)

* 입력값: GET /api/products?query=&category=&brand=&page=&sort= → 반환: products[], total
* 출력값: 엑스포트 파일(다운로드), 선택된 상품의 일괄 상태 변경 결과

## 6. 데이터 모델 / 스키마(요약)

* Product: id, code, name, brandId, price, variants[{id, sku, stock}], classificationPath[]

## 7. API 및 백엔드 요구사항

* Read: 페이징/필터/정렬 지원하는 GET /api/products
* Write: bulk update endpoint 권장 POST /api/products/bulk 또는 PATCH /api/products

## 8. 수용 기준 (Acceptance Criteria)

* 검색어/필터 조합으로 결과가 정확히 필터링되어야 한다.
* HierarchicalSelect가 모달 내부에서도 정상 동작해야 한다.

## 9. 엣지 케이스 및 오류 처리

* 이미지 로드 실패 시 대체 이미지 표시
* 서버 에러 시 사용자에게 에러 메시지와 재시도 버튼 제공

## 10. 테스트 시나리오(간단)

* 검색 결과 필터링 테스트
* 엑스포트 파일 내용 검증

## 11. 노트 및 구현 힌트

* HierarchicalSelect 사용: 포탈 렌더링, 컬럼 너비 조정

# FDD: 분류 편집 (Product Classifications)

## 1. 기본 정보

* 화면 이름 (한국어): 분류 편집
* 화면 경로(Route): /settings/classifications
* 메뉴 ID: product-classifications
* 담당자: TBD
* 우선순위: P0

## 2. 목표 및 설명

* 목적: 내부 상품 분류를 계층형 트리로 관리하여 상품 카테고리 검색/분류에 사용합니다.

## 3. 주요 사용자 플로우

* 플로우 1: 루트 분류 추가 → 하위 분류 추가 → 편집 → 삭제(삭제 미리보기 확인)

## 4. UI 구성 요소

* 트리 뷰: 각 노드에 액션(추가/편집/삭제)
* 모달: 추가/편집 모달, 삭제 미리보기 모달

## 5. 입력값 / 출력값 (Contract)

* 입력값: 로컬 productClassificationsTree 또는 GET /api/classifications
* 출력값: 트리 변경사항(POST/PUT/DELETE 권장)

## 6. 데이터 모델 / 스키마(요약)

* Classification: id, name, parentId, children[], path[]

## 7. API 및 백엔드 요구사항

* Read: GET /api/classifications
* Write: POST/PUT/DELETE /api/classifications

## 8. 수용 기준 (Acceptance Criteria)

* 삭제 시 삭제 미리보기에 영향을 받는 모든 하위 노드가 정확히 표시되어야 한다.
* AnimatedCollapse의 접기/펼치기가 자연스럽게 동작해야 한다.

## 9. 엣지 케이스 및 오류 처리

* 트리의 깊이에 따른 퍼포먼스 이슈 및 무한 루프 방지

## 10. 테스트 시나리오(간단)

* 루트/하위 추가 및 로컬 저장 확인

## 11. 노트 및 구현 힌트

* 로컬 스토리지와 서버 동기화 정책 필요(충돌 해결 방식 정의)

# FDD: 카테고리 매핑 (Category Mapping)

## 1. 기본 정보

* 화면 이름 (한국어): 카테고리 매핑
* 화면 경로(Route): /malls/category-mapping
* 메뉴 ID: category-mapping
* 담당자: TBD
* 우선순위: P0

## 2. 목표 및 설명

* 목적: 내부 분류와 쇼핑몰 카테고리 간의 대응 관계를 생성/관리하여 상품 등록/수출 시 카테고리 자동 할당에 사용합니다.

## 3. 주요 사용자 플로우

* 플로우 1: 쇼핑몰 선택 → 내부 카테고리 선택 → 쇼핑몰 카테고리 선택 → 매핑 생성 → 엑스포트

## 4. UI 구성 요소

* 쇼핑몰 타일 그리드, 내부 카테고리 리스트(HierarchicalSelect 또는 리스트), 쇼핑몰 카테고리 패널, 매핑 테이블

## 5. 입력값 / 출력값 (Contract)

* 입력값: GET /api/malls, GET /api/classifications, GET /api/mall-categories?mallId=
* 출력값: POST /api/mappings 생성, GET /api/mappings 조회

## 6. 데이터 모델 / 스키마(요약)

* Mapping: id, mallId, internalCategoryId, mallCategoryId, active, createdAt

## 7. API 및 백엔드 요구사항

* Read: GET /api/mappings?mallId=
* Write: POST /api/mappings, DELETE /api/mappings/:id, PATCH /api/mappings/:id(active)

## 8. 수용 기준 (Acceptance Criteria)

* 동일한 internalCategoryId + mallCategoryId + mallId 조합이 중복 생성되지 않아야 한다.

## 9. 엣지 케이스 및 오류 처리

* 쇼핑몰 카테고리 구조가 매우 깊을 때의 렌더링 퍼포먼스

## 10. 테스트 시나리오(간단)

* 매핑 생성 → 목록에 반영 → 중복 매핑 차단 확인

## 11. 노트 및 구현 힌트

* 엑스포트는 CSV/XLSX로 지원, 큰 파일은 스트리밍 방식 고려

# FDD: 상품 상세 (Product Detail)

## 1. 기본 정보

* 화면 이름 (한국어): 상품 상세
* 화면 경로(Route): /products/:id
* 메뉴 ID: product-detail
* 담당자: TBD
* 우선순위: P0

## 2. 목표 및 설명

* 목적: 개별 상품의 모든 메타데이터, 옵션, 재고, 가격, 설명을 확인하고 일부 설정(설명 편집, 설정 모달)을 조작할 수 있도록 합니다.

## 3. 주요 사용자 플로우

* 플로우 1: 목록에서 상품 클릭 → 상세 조회 → 설명 편집 모달 열기 → 저장(시뮬레이션)

## 4. UI 구성 요소

* 상단: 목록 복귀, 상품 설정 버튼, 설명 편집 버튼
* 주요: 이미지 영역, 핵심 정보(가격/원가/마진), 옵션 테이블, 상세 설명(프리뷰)
* 모달: 설명 편집 모달, 설정 모달

## 5. 입력값 / 출력값 (Contract)

* 입력값: GET /api/products/:id → 반환: product
* 출력값: 변경된 설명(저장 시 API POST/PUT 권장)

## 6. 데이터 모델 / 스키마(요약)

* Product: id, code, name, description(html), images[], variants[], price, cost

## 7. API 및 백엔드 요구사항

* Read: GET /api/products/:id
* Write: PUT /api/products/:id (설명/설정 변경)

## 8. 수용 기준 (Acceptance Criteria)

* 정상 상품 ID 접근 시 모든 필수 필드를 렌더.
* 존재하지 않는 상품 ID는 친절한 에러 메시지를 표시.

## 9. 엣지 케이스 및 오류 처리

* 이미지 로드 실패/빈 variants 배열 처리

## 10. 테스트 시나리오(간단)

* 설명 편집 모달 열기/닫기 및 입력/저장 시나리오 검증

## 11. 노트 및 구현 힌트

* 마진 계산은 프론트엔드에서 표기용으로 계산하며, 백엔드와 동일 로직 사용 권장

# FDD: 주문 목록 (Orders List)

## 1. 기본 정보

* 화면 이름 (한국어): 주문 목록
* 화면 경로(Route): /orders
* 메뉴 ID: orders-list
* 담당자: TBD
* 우선순위: P0

## 2. 목표 및 설명

* 목적: 주문을 검색/필터/상태 변경하고, 주문 상세로 이동하여 처리를 진행할 수 있도록 합니다.

## 3. 주요 사용자 플로우

* 플로우 1: 주문 검색 → 상태 필터 적용 → 주문 상세로 이동 → 상태 변경

## 4. UI 구성 요소

* 검색/필터 바, 주문 테이블, 상세 패널

## 5. 입력값 / 출력값 (Contract)

* 입력값: GET /api/orders?query=&status=&from=&to=
* 출력값: PATCH /api/orders/:id/status 등

## 6. 데이터 모델 / 스키마(요약)

* Order: id, orderNumber, date, customer{name,phone}, items[], total, status

## 7. API 및 백엔드 요구사항

* Read: 페이징/필터 지원 GET /api/orders
* Write: 상태 변경 PATCH /api/orders/:id(status)

## 8. 수용 기준 (Acceptance Criteria)

* 검색/필터 적용 결과가 정확히 반영되어야 한다.

## 9. 엣지 케이스 및 오류 처리

* 동시성으로 인한 상태 충돌: 서버에서의 최종 상태 확인 필요

## 10. 테스트 시나리오(간단)

* 주문 상태 변경 후 UI 및 이력 표시 확인

## 11. 노트 및 구현 힌트

* 대량 데이터에 대비한 서버사이드 페이징 권장

# FDD: 주문 상세 (Order Detail)

## 1. 기본 정보

* 화면 이름 (한국어): 주문 상세
* 화면 경로(Route): /orders/:id
* 메뉴 ID: order-detail
* 담당자: TBD
* 우선순위: P0

## 2. 목표 및 설명

* 목적: 주문의 품목, 결제/배송/고객 정보를 확인하고 발송/환불 등의 내부 처리를 수행하도록 지원합니다.

## 3. 주요 사용자 플로우

* 플로우 1: 주문 목록에서 상세 진입 → 발송 처리(송장 입력) → 상태 변경

## 4. UI 구성 요소

* 주문 요약, 품목 리스트, 고객/배송 정보, 내부 메모, 상태 변경 드롭다운

## 5. 입력값 / 출력값 (Contract)

* 입력값: GET /api/orders/:id
* 출력값: PATCH /api/orders/:id(status, shipping info)

## 6. 데이터 모델 / 스키마(요약)

* Order: id, orderNumber, items[{productId,variantId,qty,price}], shipping{carrier,trackingNumber,status}

## 7. API 및 백엔드 요구사항

* Read: GET /api/orders/:id
* Write: PATCH /api/orders/:id(status, shipping)

## 8. 수용 기준 (Acceptance Criteria)

* 발송 처리 시 송장번호 입력 후 상태가 발송 완료로 변경되어야 한다.

## 9. 엣지 케이스 및 오류 처리

* 잘못된 송장번호 포맷이나 배송사 미지원 케이스 처리

## 10. 테스트 시나리오(간단)

* 발송 처리 시 상태 변경 및 UI 반영 확인

## 11. 노트 및 구현 힌트

* 외부 배송사 API 연동 시 비동기 처리 및 실패 리트라이 정책 고려

# FDD: 통합(Integrations)

## 1. 기본 정보

* 화면 이름 (한국어): 통합 관리
* 화면 경로(Route): /integrations
* 메뉴 ID: integrations-list
* 담당자: TBD
* 우선순위: P1

## 2. 목표 및 설명

* 목적: 외부 서비스 연동(결제, 배송, 마켓)을 등록/관리하고 테스트 이벤트를 전송해 연동을 검증합니다.

## 3. 주요 사용자 플로우

* 플로우 1: 연동 추가 → 인증 정보 입력 → 테스트 이벤트 전송 → 결과 확인

## 4. UI 구성 요소

* 연동 리스트, 연동 상세(자격증명 입력 폼), webhook 테스트 버튼

## 5. 입력값 / 출력값 (Contract)

* 입력값: GET /api/integrations
* 출력값: POST /api/integrations

## 6. 데이터 모델 / 스키마(요약)

* Integration: id, type, name, credentials, lastSync

## 7. API 및 백엔드 요구사항

* Read/Write: GET/POST/PUT /api/integrations

## 8. 수용 기준 (Acceptance Criteria)

* 테스트 이벤트 전송 후 결과가 UI에 표시되어야 한다.

## 9. 엣지 케이스 및 오류 처리

* 인증 실패 또는 webhook 실패 시 상세 에러 메시지 제공

## 10. 테스트 시나리오(간단)

* 연동 추가 후 테스트 이벤트 전송 및 수신 확인

## 11. 노트 및 구현 힌트

* 민감한 자격증명은 서버에 안전히 저장되어야 함

# FDD: 설정 (Settings)

## 1. 기본 정보

* 화면 이름 (한국어): 설정
* 화면 경로(Route): /settings
* 메뉴 ID: settings
* 담당자: TBD
* 우선순위: P2

## 2. 목표 및 설명

* 목적: 앱 전반의 기본 설정(통화, 세금, 알림 등) 및 사용자 권한 관리를 제공합니다.

## 3. 주요 사용자 플로우

* 플로우 1: 기본 통화/세금 설정 변경 → 저장

## 4. UI 구성 요소

* 일반 설정 폼, 알림 토글, 사용자/권한 관리 섹션

## 5. 입력값 / 출력값 (Contract)

* 입력값: GET /api/settings
* 출력값: PUT /api/settings

## 6. 데이터 모델 / 스키마(요약)

* Settings: defaultCurrency, taxRate, notificationChannels[]

## 7. API 및 백엔드 요구사항

* Read/Write: GET/PUT /api/settings

## 8. 수용 기준 (Acceptance Criteria)

* 설정 변경 저장 시 시스템 전체에 반영되는지 확인(시뮬레이션)

## 9. 엣지 케이스 및 오류 처리

* 잘못된 포맷의 입력값(예: 통화 코드)에 대한 유효성 검사

## 10. 테스트 시나리오(간단)

* 설정 변경 후 저장 및 백엔드 동기화 확인

## 11. 노트 및 구현 힌트

* 사용자/권한 관리는 향후 ACL 연동 필요

# FDD: 매장 (Malls)

## 1. 기본 정보

* 화면 이름 (한국어): 매장 목록 및 설정
* 화면 경로(Route): /malls
* 메뉴 ID: malls-list
* 담당자: TBD
* 우선순위: P1

## 2. 목표 및 설명

* 목적: 연결된 쇼핑몰(판매 채널)을 등록/관리하고 동기화 상태 및 로그를 모니터링합니다.

## 3. 주요 사용자 플로우

* 플로우 1: 매장 목록 조회 → 신규 매장 추가(자격증명 입력) → 동기화 상태 확인

## 4. UI 구성 요소

* 매장 타일/리스트, 매장 상세(설정 폼), 동기화 로그 뷰

## 5. 입력값 / 출력값 (Contract)

* 입력값: GET /api/malls
* 출력값: POST /api/malls (연결 정보 저장)

## 6. 데이터 모델 / 스키마(요약)

* Mall: id, name, platform, credentials{clientId,secret}, lastSync,status

## 7. API 및 백엔드 요구사항

* Read/Write: GET/POST/PUT /api/malls

## 8. 수용 기준 (Acceptance Criteria)

* 매장 추가 폼 제출 시 리스트에 반영되어야 함.

## 9. 엣지 케이스 및 오류 처리

* 잘못된 자격증명 입력 시 친절한 에러 메시지 제공

## 10. 테스트 시나리오(간단)

* 매장 추가 및 동기화 로그 열람 테스트

## 11. 노트 및 구현 힌트

* 실제 연동은 백엔드 서비스와의 토큰 교환 로직 필요

# FDD: 공유 컴포넌트 및 유틸리티 (Shared)

## 1. 기본 정보

* 항목: HierarchicalSelect, AnimatedCollapse, Modal sizing helpers
* 위치: src/components/common/
* 담당자: TBD

## 2. 목표 및 설명

* 목적: 여러 화면에서 재사용 가능한 UI 컴포넌트의 동작 정의, props 계약, 접근성 요구사항을 명확히 합니다.

## 3. 개별 컴포넌트 사양

* HierarchicalSelect
  + Props: items, selectedId, onSelect, colWidth?, maxDepth?
  + 동작: 포탈로 렌더, 컬럼 기반 뎁스 표시, 키보드 접근성 지원
* AnimatedCollapse
  + Props: isOpen, children, duration?
  + 동작: scrollHeight 기반 max-height 트랜지션
* Modal sizing helpers
  + 목적: 다양한 컨텍스트에서 일관된 모달 크기 제공 (max-w-3xl, max-w-2xl, max-w-xl)

## 4. 수용 기준

* 컴포넌트의 기본 props 계약이 문서와 일치하고, 주요 화면에서 재사용 가능해야 합니다.

## 5. 테스트 시나리오

* 포탈화된 드롭다운의 표시/숨김, 키보드 네비게이션 테스트

# FDD 템플릿 (Feature Design Document)

사용 방법: 각 화면별로 이 템플릿을 복사하여 필요한 값을 채우고, 구현 관련 추가 메모(API, 데이터 모델, 에지 케이스)를 작성하세요.

## 1. 기본 정보

* 화면 이름 (한국어):
* 화면 경로(Route):
* 메뉴 ID:
* 담당자:
* 우선순위: (P0/P1/P2)

## 2. 목표 및 설명

* 목적: 이 화면이 사용자에게 제공하는 핵심 가치와 성공 기준.

## 3. 주요 사용자 플로우

* 플로우 1: (예: 리스트에서 항목 선택 → 상세 모달 열기 → 수정 → 저장)
* 플로우 2:

## 4. UI 구성 요소

* 상단(헤더)/검색바/필터:
* 주요 테이블/카드/폼:
* 모달/팝오버:
* 액션 버튼(저장/삭제/엑스포트 등):

## 5. 입력값 / 출력값 (Contract)

* 입력값(Props/API):
  + (예) GET /api/products?filter=... → 반환: product[]
* 출력값(사용자 동작의 결과):

## 6. 데이터 모델 / 스키마(요약)

* 필요한 엔티티 및 핵심 필드:
  + Product: id, code, name, price, variants[]
  + Classification: id, name, parentId, path[]

## 7. API 및 백엔드 요구사항

* 읽기(Read):
* 생성/수정/삭제(Write):
* 권한/인증 요구사항:

## 8. 수용 기준 (Acceptance Criteria)

* 필수 동작(기능적):
* 비기능적 요구(성능, 접근성):

## 9. 엣지 케이스 및 오류 처리

* 네트워크 실패:
* 권한 오류:
* 데이터 없음(빈 상태):

## 10. 테스트 시나리오(간단)

* 유닛 테스트:
* 통합/엔드투엔드 테스트:

## 11. 노트 및 구현 힌트

* UI/스타일 가이드 링크:
* 재사용 컴포넌트: