

# 프로젝트 설계서

식단관리를 위한 AI를 이용한 성분 분석 어플리케이션

---

## 4조

팀장 | 홍성호

팀원 | 이동제, 윤현서, 최현빈, 홍창희

# 목차

table of contents

- 1 서론
- 2 사용자요구
- 3 아키텍처설계
- 4 상세설계

# 1

# 서론

---

# 식단관리를 위한 AI를 이용한 성분 분석 어플리케이션

- 섭취 음식의 칼로리와 영양소 정보를 효율적으로 기록
- 건강 목표에 맞춘 식단 관리에 도움



# 시스템의 목적

사용자가 섭취한 음식을 손쉽게 기록하고 관리하도록 지원  
음식 사진 분석하여 칼로리와 영양소 정보를 제공하고 기록의 정확도를 높임  
데이터를 바탕으로 건강 목표에 맞는 맞춤형 식단 계획 제안  
데이터베이스 기반 기록 관리를 통해 사용자에게 통계 및 인사이트 제공

# 2

## 사용자 요구

---

# 주체별 요구 사항

## 사용자

- 음식사진업로드 자동으로칼로리와영양성분정보확인
- 하루식단기록 및 영양섭취통계확인
- 개인건강목표 설정및맞춤형식단
- 식단관련제품 검색및구매

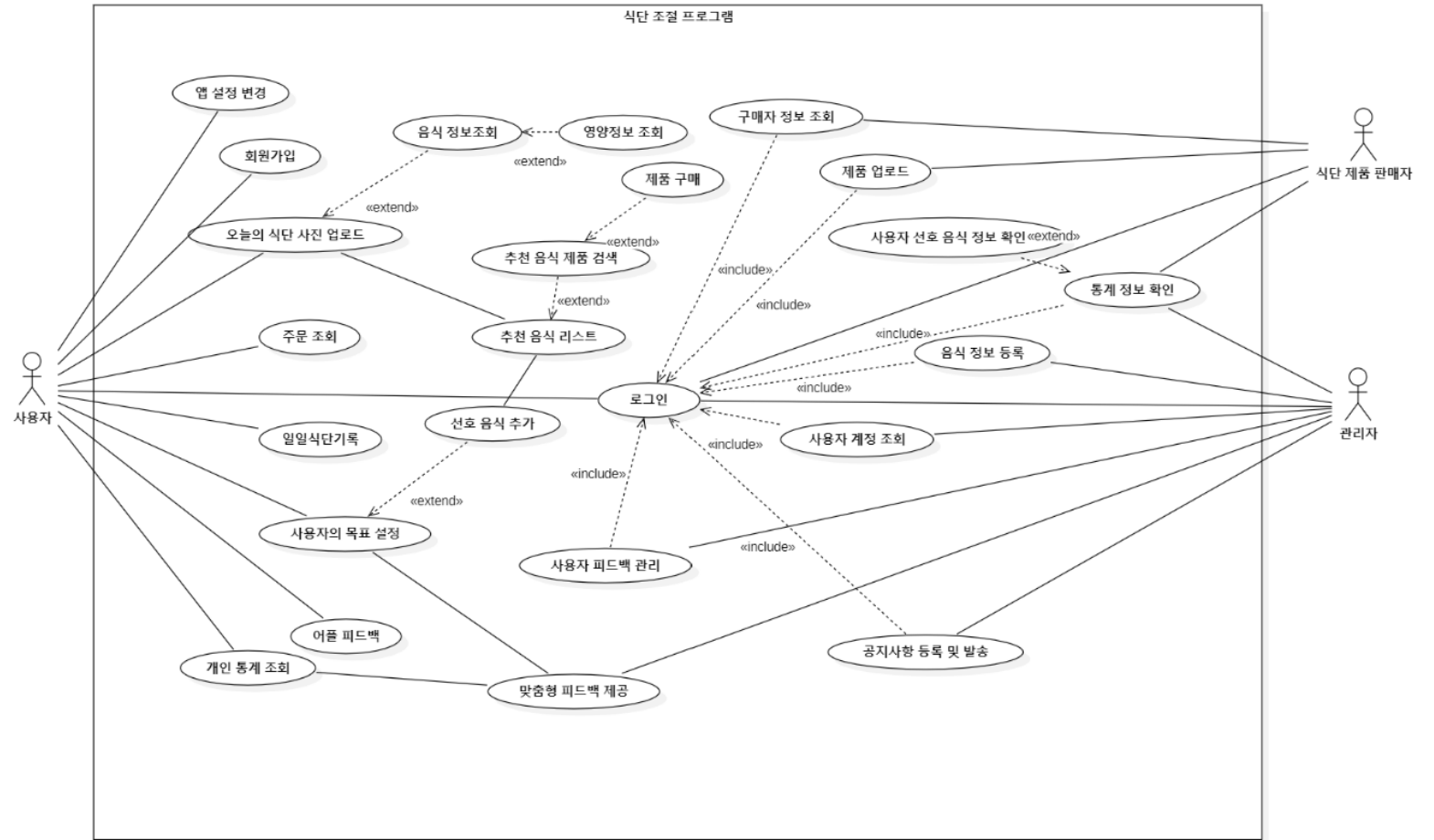
## 관리자

- 어플리케이션내음식정보등록및수정 가능
- 사용자피드백검토후시스템개선가능
- 사용자및판매자의개정 정보확인, 필요에따라계정활성여부 결정
- 사용자선호데이터분석후개선된추천서비스제공

## 판매자

- 어플리케이션에신규제품등록가능
- 제품구매자조회, 배송정보확인및관리가능
- 판매통계확인, 이를통해판매전략계획가능
- 사용자요구기반할인쿠폰또는프로모션제공가능

# 사용사례 다이아그램





# 사용사례 테이블

모든 사용사례테이블은 설계서원본에 첨부

2.3.18 기능 #18 (음식 정보 등록)

구분	내용
요구사항명	음식 정보 등록
액터명	관리자
개요 및 설명	관리자가 새로운 음식 데이터를 추가하여 사용자들이 다양한 음식의 영양 정보를 확인할 수 있도록 합니다.
사전 조건	관리자는 로그인 상태여야 하며, 음식 정보 등록 권한이 있어야 합니다.
기본 흐름	1. 관리자가 '음식 정보 등록' 메뉴를 선택합니다. 2. 시스템은 음식명, 칼로리, 영양 성분 등의 정보를 입력할 수 있는 폼을 제공합니다. 3. 관리자가 정보를 입력하고 저장 버튼을 누릅니다. 4. 시스템은 입력된 음식을 데이터베이스에 저장하고, 사용자가 조회할 수 있도록 합니다.
예외 흐름	예외 상황 1: 필수 정보가 누락된 경우 > 시스템은 '필수 정보를 모두 입력해주세요'라는 메시지를 표시하고, 입력을 유도합니다.
대안 흐름	대안 흐름 1: 정보를 임시 저장하여 나중에 편집하고 싶은 경우 > 시스템은 임시 저장 기능을 제공하여 관리자가 이후에 내용을 완성할 수 있도록 합니다.
사후 조건	음식 정보가 데이터베이스에 저장되어 사용자들이 검색할 수 있게 됩니다.

2.3.9 기능 #9 (영양 정보 조회)

구분	내용
요구사항명	영양 정보 조회
액터명	사용자
개요 및 설명	사용자가 업로드한 음식 사진을 바탕으로 시가 분석한 음식의 영양 성분 정보를 조회하는 과정입니다.
사전 조건	음식 사진 업로드 및 AI 분석이 완료되어야 합니다.
기본 흐름	1. 시스템이 AI 분석 결과를 바탕으로 음식의 칼로리와 영양 정보를 표시합니다. 2. 사용자는 화면에서 해당 음식의 칼로리, 탄수화물, 단백질, 지방 등의 성분을 확인합니다.
예외 흐름	예외 상황 1: 분석 결과가 불확실한 경우 > 시스템은 '분석 결과가 명확하지 않습니다. 사진을 다시 업로드해주세요'라는 메시지를 표시합니다.
대안 흐름	대안 흐름 1: 사용자가 영양 성분을 직접 입력하고 싶은 경우 > 시스템은 사용자가 직접 입력할 수 있도록 텍스트 입력 기능을 제공합니다.
사후 조건	사용자는 음식의 영양 성분 정보를 조회하여 식단에 반영할 수 있습니다.

2.3.26 기능 #26 (제품 업로드)

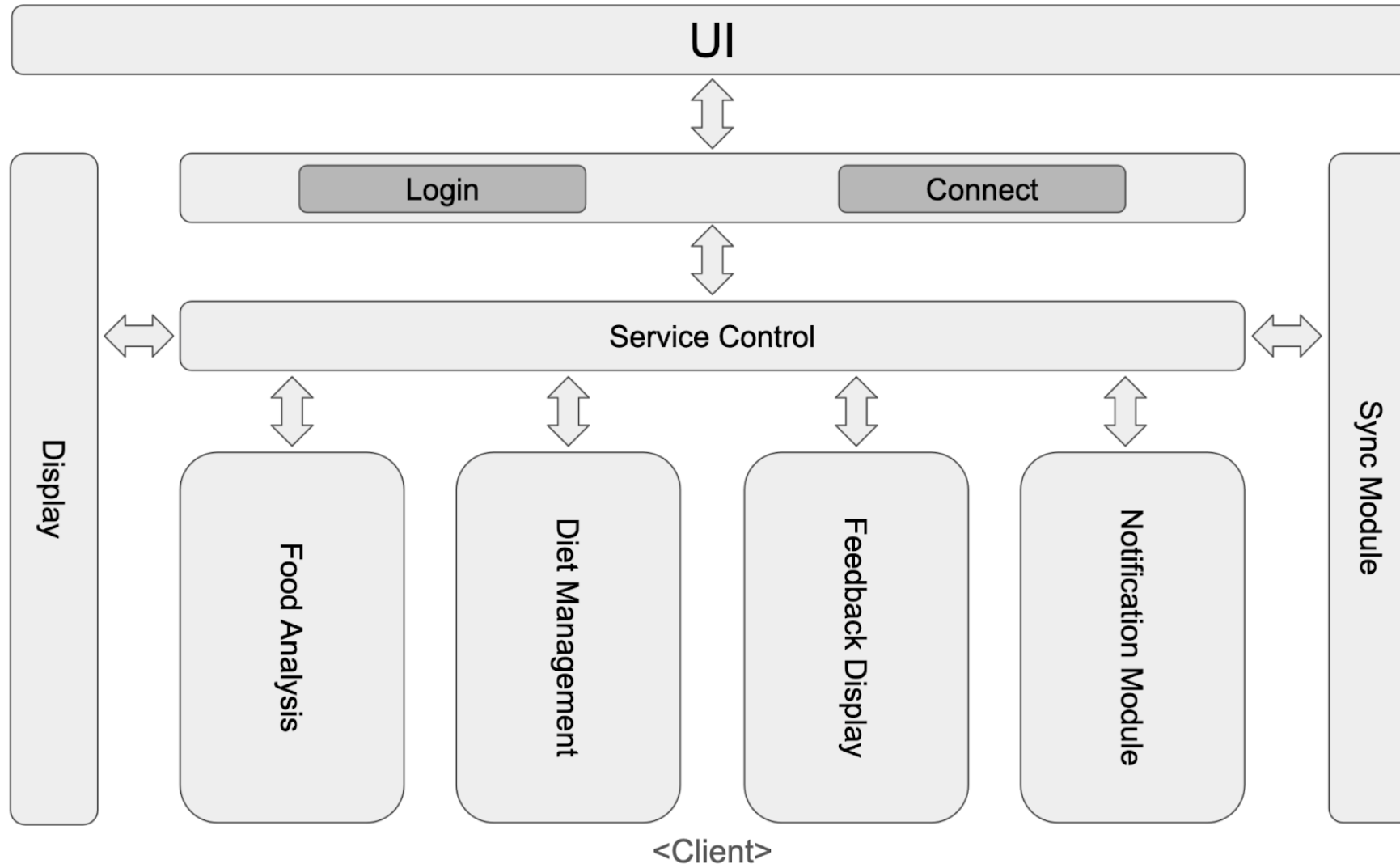
구분	내용
요구사항명	제품 업로드
액터명	식단 제품 판매자
개요 및 설명	식단 제품 판매자가 새로운 제품을 어플리케이션에 업로드하여 판매할 수 있도록 합니다.
사전 조건	판매자는 관리자 계정을 통해 로그인되어 있어야 하며, 제품 등록 권한을 보유해야 합니다.
기본 흐름	1. 판매자가 '제품 업로드' 메뉴를 선택합니다. 2. 시스템은 제품명, 설명, 가격, 이미지 등을 입력할 수 있는 양식을 제공합니다. 3. 판매자가 필요한 정보를 입력하고 '저장' 버튼을 클릭합니다. 4. 시스템은 입력된 정보를 데이터베이스에 저장하고, 사용자들에게 제품을 표시합니다.
예외 흐름	예외 상황 1: 필수 입력 항목이 누락된 경우 > 시스템은 '필수 정보를 모두 입력해주세요'라는 메시지를 표시하고, 누락된 항목을 강조하여 입력을 유도합니다. 예외 상황 2: 제품 이미지가 지원하지 않는 형식인 경우 > 시스템은 '지원되지 않는 이미지 형식입니다'라는 메시지를 표시하고, 적합한 형식을 업로드하도록 안내합니다.
대안 흐름	대안 흐름 1: 판매자가 제품 정보를 임시 저장하고 나중에 추가하거나 수정하고 싶은 경우 > 시스템은 '임시 저장' 기능을 제공하여 판매자가 나중에 내용을 수정 및 완성할 수 있도록 지원합니다.
사후 조건	제품 정보가 어플리케이션에 표시되며, 사용자들이 제품을 확인하고 구매할 수 있습니다.

# 3

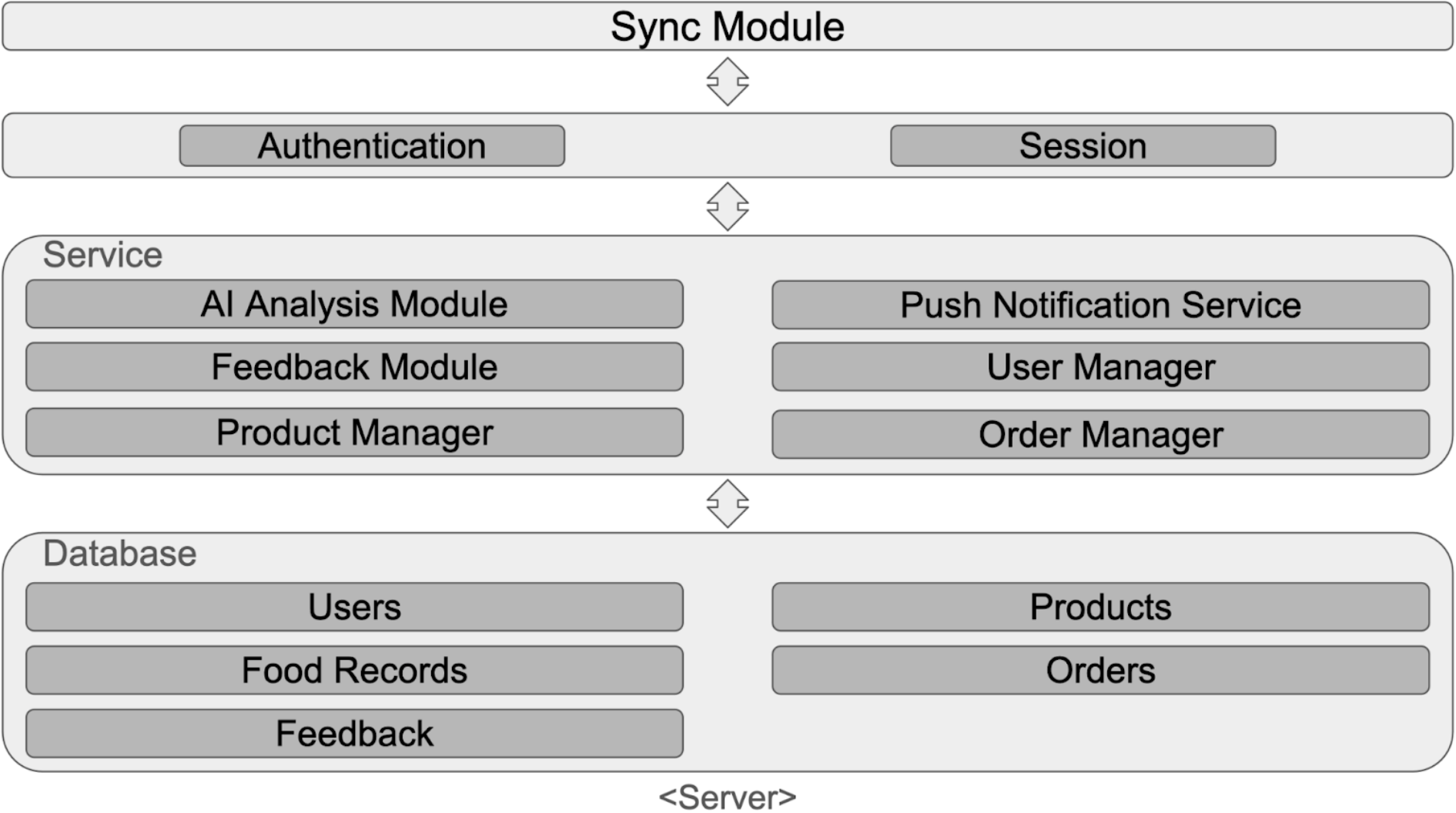
## 아키텍처 설계

---

## 시스템아키텍처



# 시스템아키텍처



1

Authentication Subsystem (인증 서브시스템)

역할	사용자의 로그인, 로그아웃, 인증 및 권한 관리를 처리
관련 클래스	<ul style="list-style-type: none"><li>• User: 사용자 정보를 저장하고 인증 정보를 처리</li><li>• Admin: 관리자의 인증과 관련된 작업을 수행</li><li>• Authentication Module: 사용자 ID와 비밀번호를 검증하고 세션을 생성</li></ul>
주요 동작	<ul style="list-style-type: none"><li>• 로그인 및 로그아웃 처리</li><li>• 인증 토큰 생성 및 세션 관리</li><li>• 권한 검증 (사용자, 판매자, 관리자)</li></ul>

2

User Management Subsystem (사용자관리서브시스템)

역할	사용자 계정 및 활동 데이터를 관리
관련 클래스	<ul style="list-style-type: none"><li>User: 사용자 정보를 관리 (ID, 비밀번호, 나이, 연락처 등)</li><li>UserGoal: 사용자가 설정한 목표 데이터를 관리</li><li>Feedback: 사용자에게 제공되는 피드백 정보를 저장하고 관리</li></ul>
주요 동작	<ul style="list-style-type: none"><li>사용자 계정 생성, 수정, 삭제</li><li>사용자 목표 설정</li><li>사용자별 피드백 제공</li></ul>

3

Product Management Subsystem (제품관리서브시스템)

역할	제품 정보 및 판매 관련 작업을 관리
관련 클래스	<ul style="list-style-type: none"><li>Product: 제품 ID, 이름, 공급자 등 제품 관련 데이터를 관리</li><li>Seller: 판매자의 제품 목록과 관련 데이터를 관리</li><li>Order: 사용자가 구매한 주문 데이터를 저장</li><li>Payment: 결제 정보를 관리</li></ul>
주요 동작	<ul style="list-style-type: none"><li>제품 정보 등록 및 수정</li><li>주문 생성 및 조회</li><li>결제 상태 업데이트</li></ul>

4

Food Analysis Subsystem (음식 분석 서브시스템)

역할	업로드 된 음식 이미지를 분석하고 영양 정보를 제공
관련 클래스	<ul style="list-style-type: none"><li>FoodPicture: 사용자가 업로드한 음식 이미지를 저장합니다.</li><li>AI Service: AI 모델을 통해 음식 이미지를 분석합니다.</li><li>Food: 분석된 음식 정보(칼로리, 영양소 등)를 저장합니다.</li><li>FoodRecommend: 사용자 맞춤형 음식 추천 데이터를 제공</li></ul>
주요 동작	<ul style="list-style-type: none"><li>음식 사진 분석</li><li>음식 데이터 조회 및 요약 정보 생성</li><li>사용자 맞춤형 음식 추천</li></ul>



5

Feedback Subsystem (피드백 서브시스템)

역할	사용자의 데이터와 목표를 기반으로 맞춤형 피드백을 제공
관련 클래스	<ul style="list-style-type: none"><li>Feedback: 피드백 데이터를 저장</li><li>AdminFeedback: 관리자가 생성한 피드백 정보를 저장</li><li>User: 피드백을 요청하고, 받은 피드백 데이터를 처리</li></ul>
주요 동작	<ul style="list-style-type: none"><li>사용자 활동 및 목표 분석</li><li>개인화된 피드백 생성</li><li>관리자 피드백 작성 및 발송</li></ul>

6

Database Subsystem (데이터베이스 서브시스템)

역할	모든 데이터를 효율적으로 저장하고 관리
관련 클래스	<ul style="list-style-type: none"><li>Database: 시스템의 중심 데이터베이스로 사용자, 음식, 피드백 등의 데이터를 저장</li><li>FoodRecords: 사용자의 식단 기록 데이터를 저장</li><li>Users: 사용자 정보를 저장</li><li>Feedback: 피드백 데이터를 저장</li></ul>
주요 동작	<ul style="list-style-type: none"><li>데이터 저장 및 검색</li><li>데이터 무결성 및 동기화 유지</li></ul>

## Subsystem 간 상호작용

Authentication Subsystem ↔ User Management Subsystem

로그인 후 사용자 정보 조회 및 권한 확인

User Management Subsystem ↔ Feedback Subsystem

사용자 활동 데이터를 기반으로 피드백 생성

Product Management Subsystem ↔ Database Subsystem

주문, 제품 데이터를 데이터베이스에 저장 및 검색

Food Analysis Subsystem ↔ Database Subsystem

음식 분석 결과를 데이터베이스에 저장 및 검색

1

## Authentication Subsystem (인증 서브시스템)

### 모듈

- Login Module: 사용자 ID와 비밀번호를 입력 받아 데이터베이스와 비교해 인증을 수행
- Session Module: 인증된 사용자를 위한 세션을 생성하고 관리
- Authorization Module: 사용자의 권한(일반 사용자, 관리자, 판매자)확인 후 적절한 액세스 제공

### 주요 역할

- 사용자 로그인 및 로그아웃 처리
- 세션 타임아웃 및 갱신
- 사용 권한에 따라 다른 화면 및 기능 제공

2

## User Management Subsystem (사용자관리서브시스템)

### 모듈

- User Profile Module: 사용자의 개인 정보를 저장 및 수정
- Goal Management Module: 사용자가 설정한 목표(예: 다이어트, 체중 증가 등) 저장 및 업데이트
- Feedback Storage Module: 사용자가 받은 피드백 데이터를 저장하고 불러옴

### 주요 역할

- 사용자 계정 생성, 삭제, 정보 수정
- 목표 설정 및 수정
- 사용자별 맞춤 피드백 기록 관리

3

## Product Management Subsystem (제품관리서브시스템)

### 모듈

- Product Catalog Module: 모든 제품의 목록을 관리하고 사용자에게 표시
- Order Management Module: 사용자가 생성한 주문 데이터를 처리하고 상태 업데이트
- Payment Module: 사용자의 결제를 처리하고 결제 상태를 저장
- Seller Module: 판매자의 제품 등록 및 관리 기능을 처리

### 주요 역할

- 제품 정보 등록, 조회, 수정
- 주문 생성 및 상태 관리
- 결제 처리 및 기록

4

## Food Analysis Subsystem (음식 분석 서브시스템)

### 모듈

- Image Upload Module: 사용자가 업로드한 이미지를 수집하고 AI 분석 모듈로 전달
- AI Food Analysis Module: 업로드 된 이미지를 분석하여 음식의 칼로리와 영양 정보를 추출
- Food Information Module: 분석된 음식 데이터를 저장하고 조회
- Recommendation Module: 사용자의 기록과 목표를 기반으로 맞춤형 음식을 추천

### 주요 역할

- 음식 이미지 분석
- 음식 정보 저장 및 조회
- 사용자 맞춤형 음식 추천

5

## Feedback Subsystem (피드백 서브시스템)

### 모듈

- Feedback Generation Module: 사용자 데이터를 분석하여 개인화된 피드백을 생성
- Admin Feedback Module: 관리자가 작성한 피드백 데이터를 저장하고 사용자에게 전달
- Feedback Display Module: 사용자가 받은 피드백을 조회하고 UI에 표시

### 주요 역할

- 개인화된 피드백 생성 및 제공
- 관리자 피드백 작성 및 저장
- 사용자 피드백 데이터 조회



6

## Database Subsystem (데이터베이스 서브시스템)

### 모듈

- User Database Module: 사용자 관련 데이터를 저장하고 제공 (예: ID, 이름, 나이 등)
- Food Database Module: 음식 및 영양 정보 데이터를 저장하고 제공
- Order Database Module: 주문 정보를 저장하고 관리
- Feedback Database Module: 피드백 데이터를 저장하고 제공

### 주요 역할

- 데이터 연속성 보장
- 빠른 검색 및 저장
- 데이터 무결성 유지

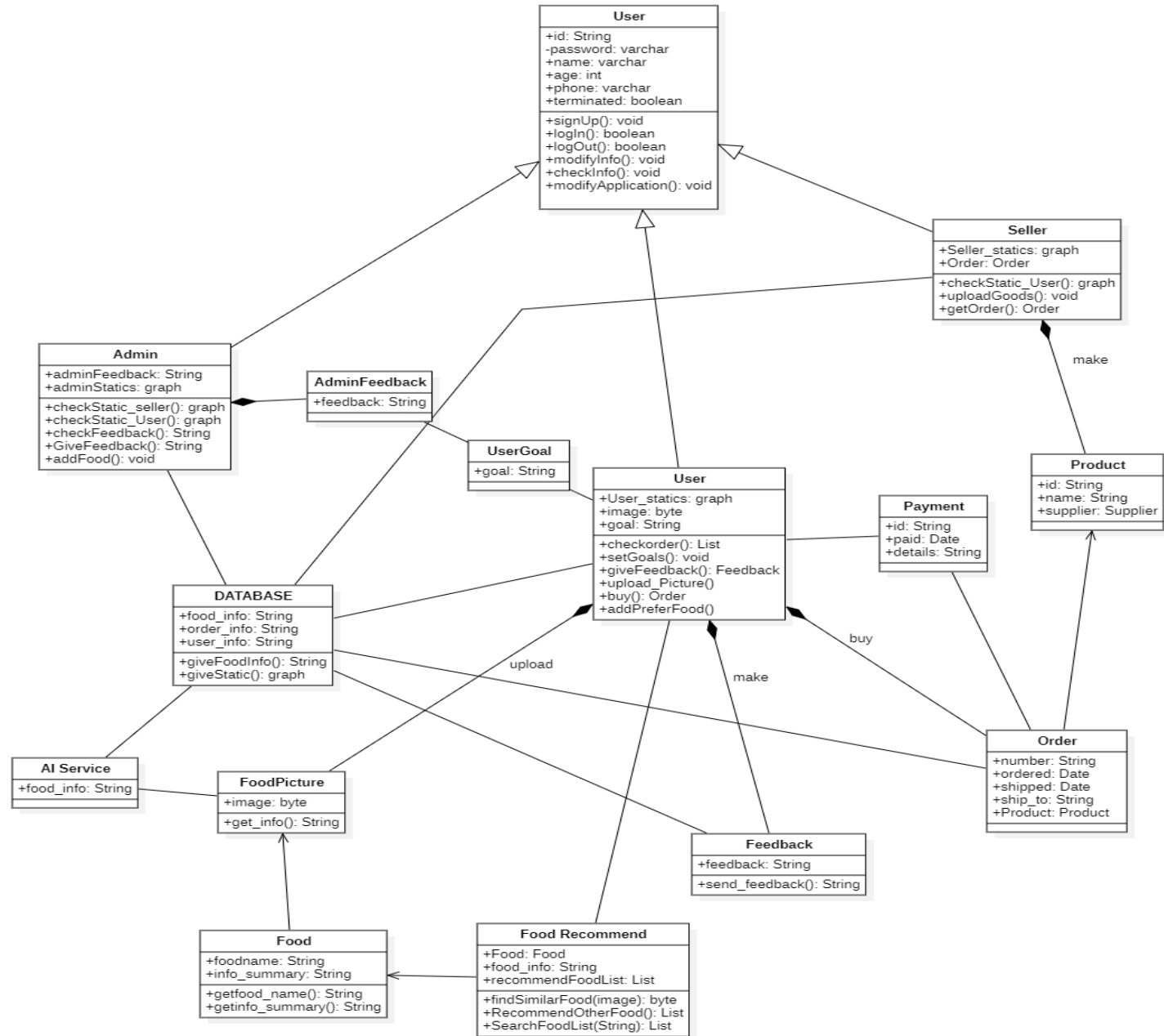
# 4

## 상세설계

---

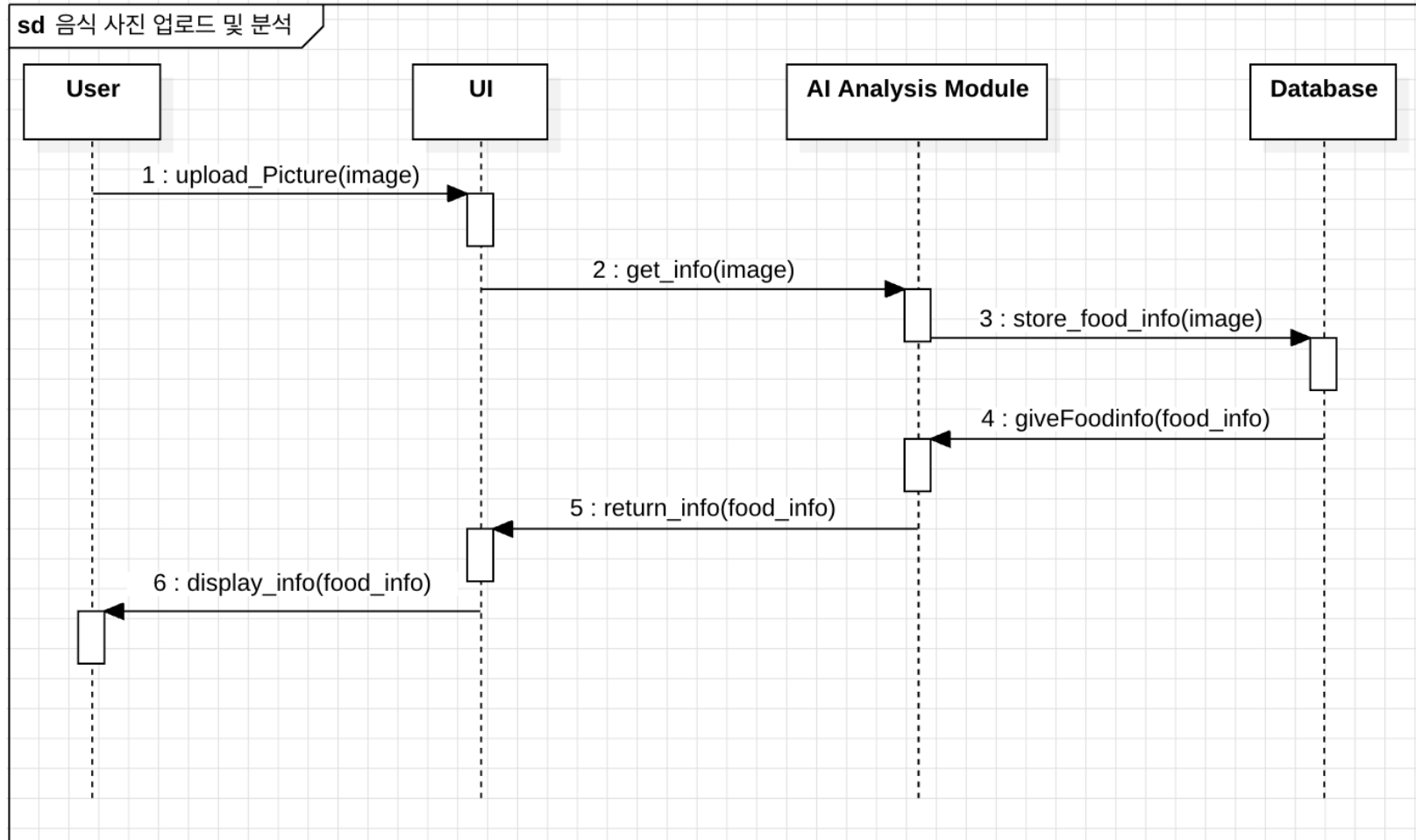
## Part 4.1. 클래스 다이어그램

# 클래스 다이어그램



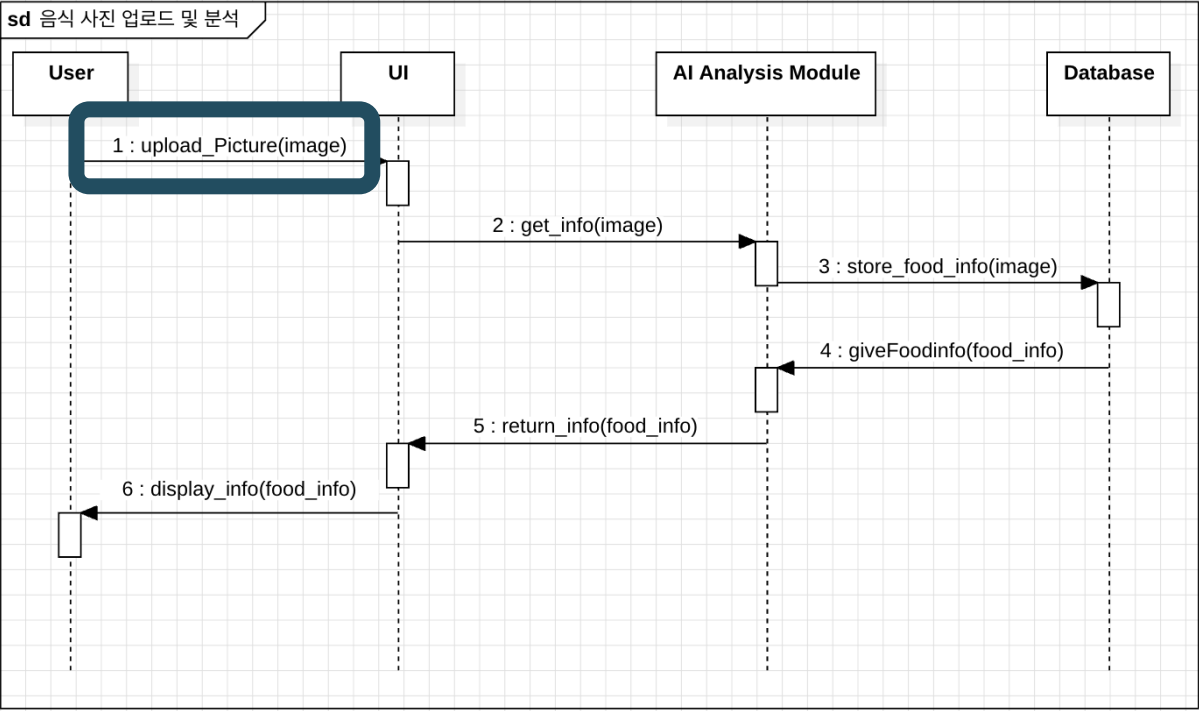
1

## 음식 사진 업로드 및 분석(사용자)



1

# 음식사진업로드및분석(사용자)

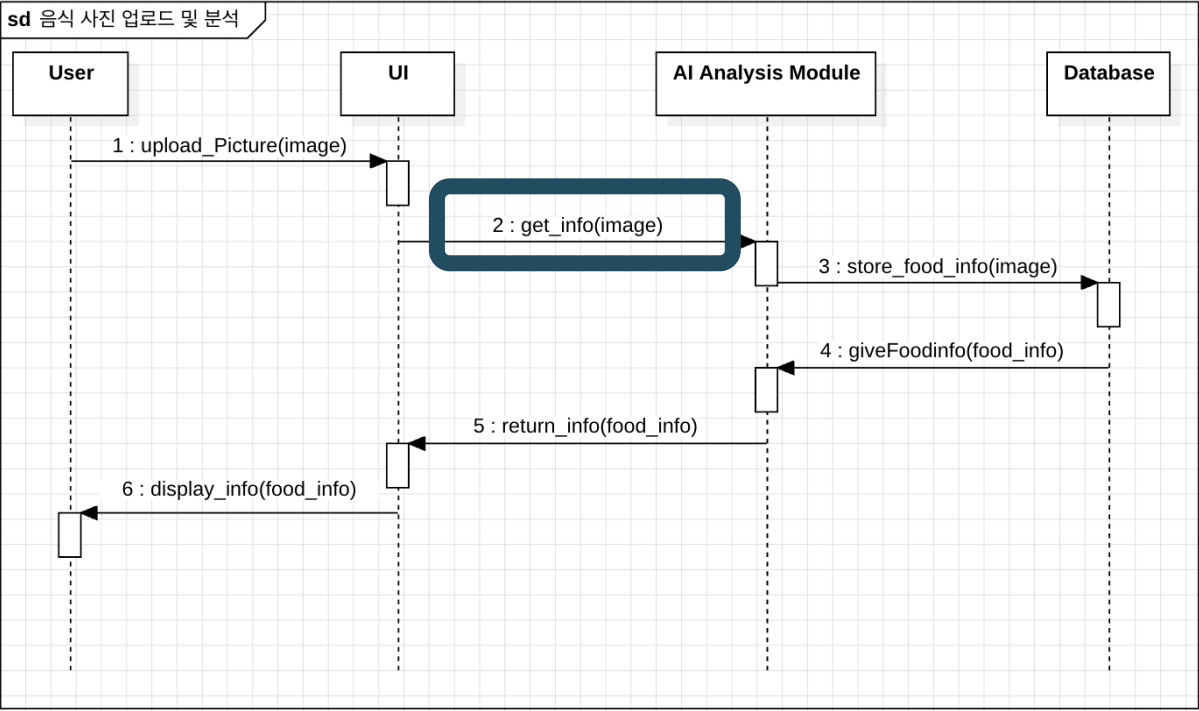


## upload\_Picture(image)

설명	사용자가 UI에 음식 사진을 업로드하는 메서드
역할	업로드한 이미지를 시스템에 전달하여 분석 요청
참여객체	User → UI
파라미터	image: 사용자가 업로드한 음식 사진 데이터

1

음식사진업로드및분석(사용자)

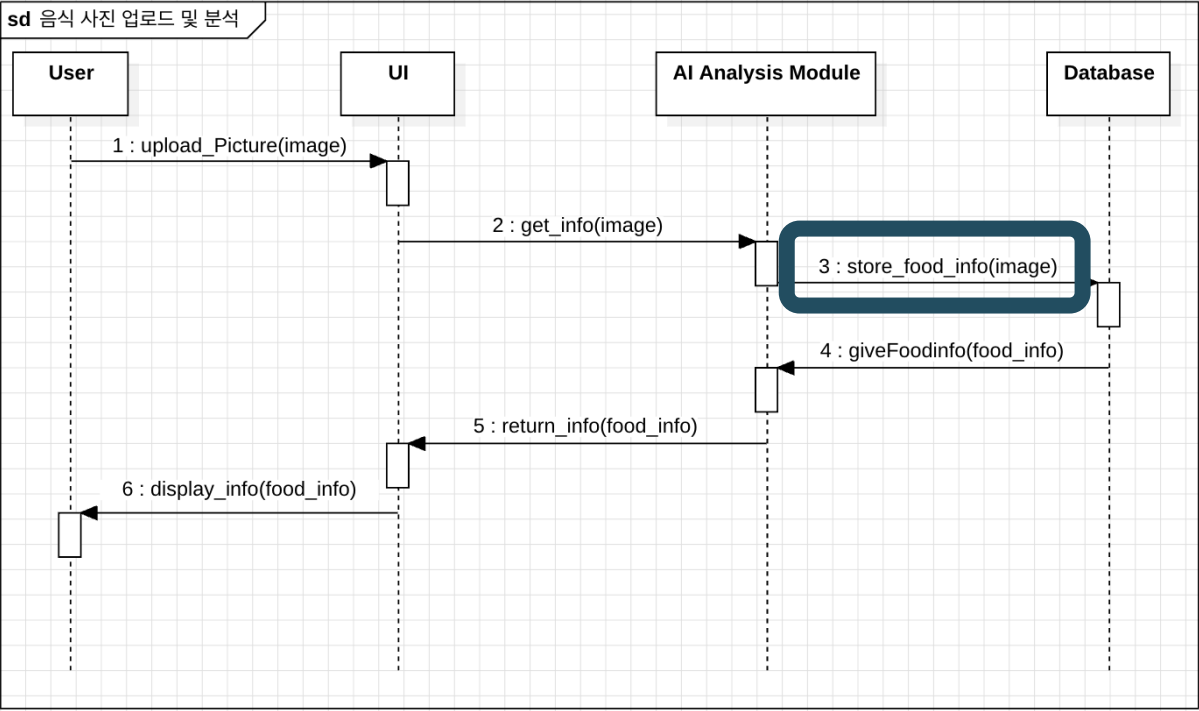


get\_info(image)

설명	UI가 AI 분석 모듈에 이미지 분석 요청을 보내는 메서드
역할	AI 분석 모듈로 전달하여 칼로리 및 영양 정보를 분석
참여객체	UI → AI Analysis Module
파라미터	image: 분석 대상 음식 사진 데이터

1

음식사진업로드및분석(사용자)

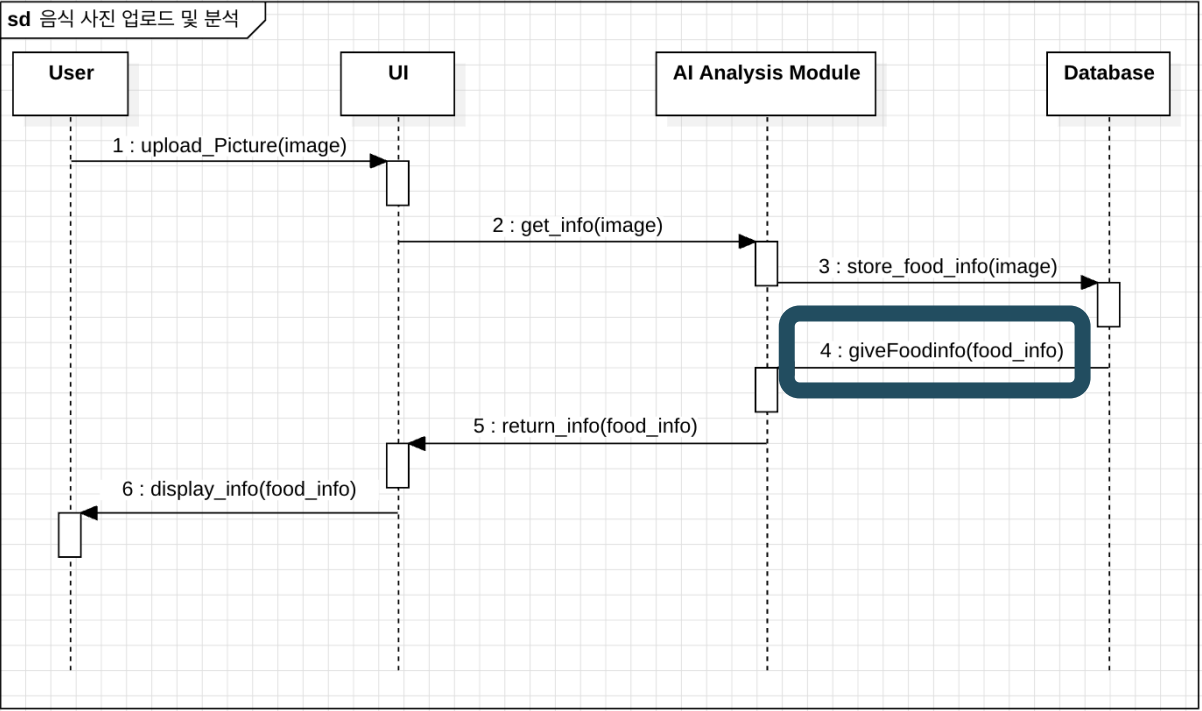


store\_food\_info(image)

설명	AI 분석 모듈이 데이터베이스에 이미지 데이터 저장
역할	분석 중간 데이터를 데이터베이스에 저장하여 기록을 남기고, 이후 참고 가능
참여객체	AI Analysis Module → Database
파라미터	image: 음식 사진 데이터

1

# 음식사진업로드및분석(사용자)



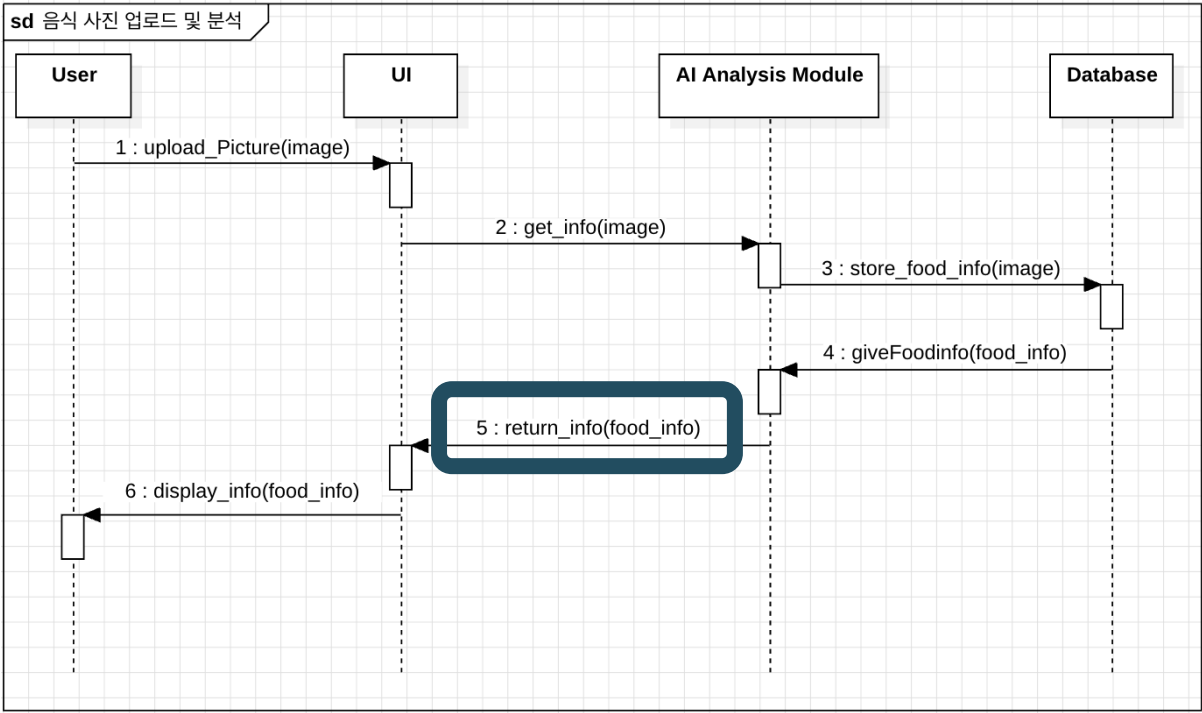
## giveFoodinfo(food\_info)

설명	데이터베이스가 분석된 음식 정보를 AI 분석 모듈에 반환하는 메서드
역할	AI 분석 모듈이 요청한 음식 정보를 데이터베이스에서 검색하여 반환
참여객체	Database → AI Analysis Module
파라미터	food_info: 음식의 칼로리 및 영양 성분 정보



1

음식사진업로드및분석(사용자)

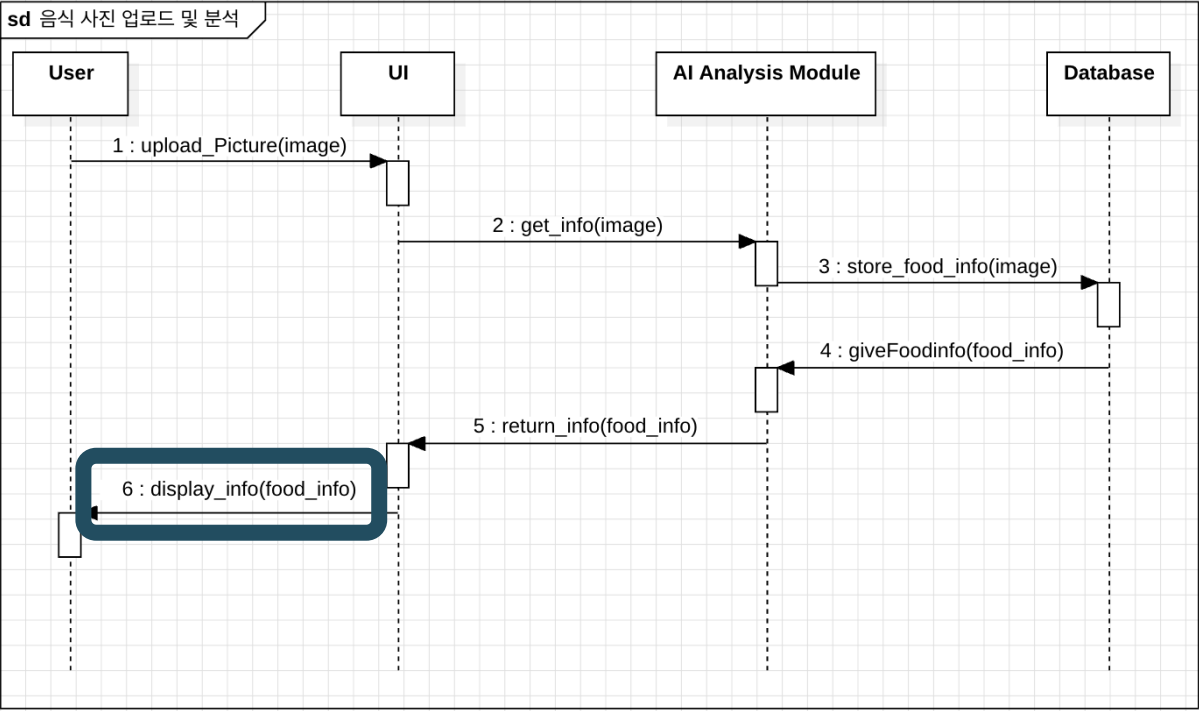


return\_info(food\_info)

설명	AI 분석 모듈이 분석 결과를 UI에 반환하는 메서드
역할	분석된 음식 정보를 사용자 인터페이스(UI)로 전달하여 사용자에게 결과를 준비
참여객체	AI Analysis Module → UI
파라미터	food_info: 음식의 칼로리 및 영양 성분 정보

1

음식사진업로드및분석(사용자)

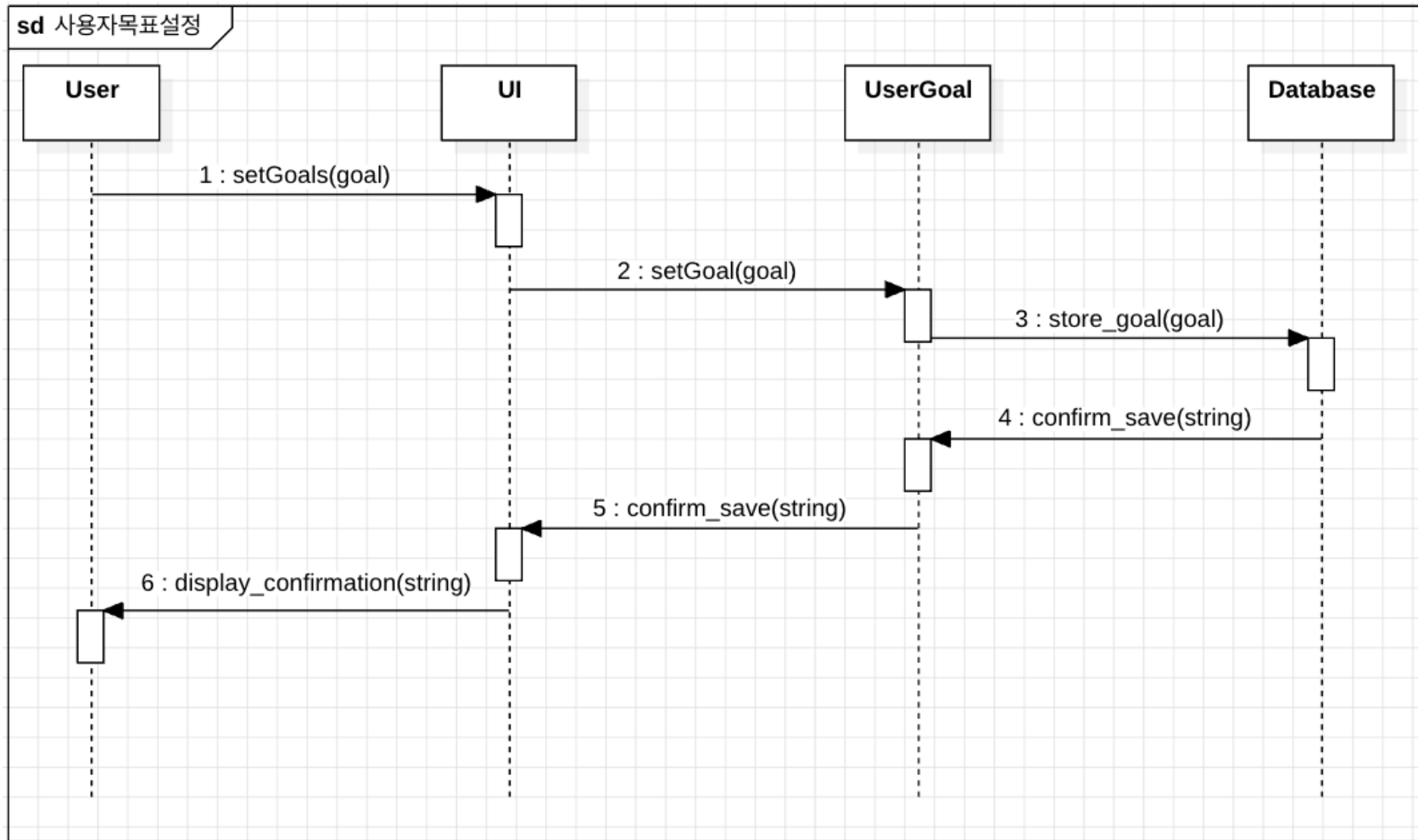


display\_info(food\_info)

설명	UI가 사용자에게 분석된 음식 정보를 출력하는 메서드
역할	최종 분석 결과를 사용자가 이해할 수 있는 형태로 화면에 표시
참여객체	UI → User
파라미터	food_info: 음식의 칼로리 및 영양 성분 정보

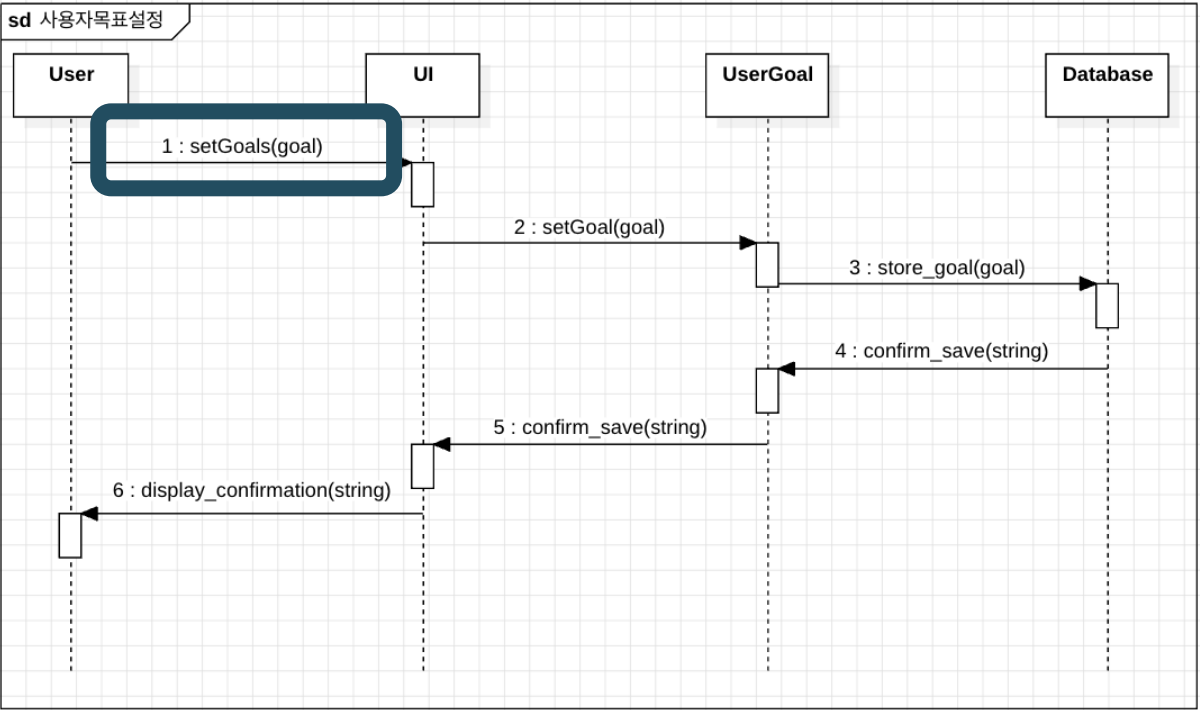
2

## 사용자 목표 설정(사용자)



2

# 사용자목표 설정(사용자)

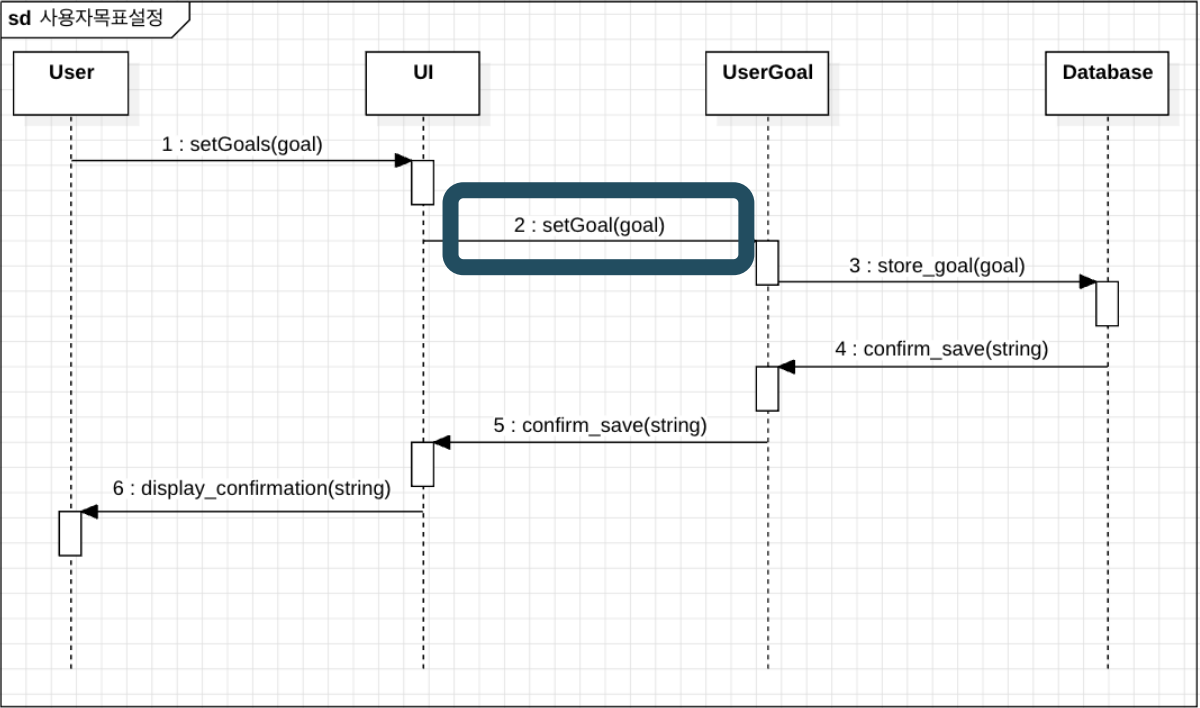


## setGoals(goal)

설명	사용자가 목표를 설정하는 요청을 UI에 전달하는 메서드
역할	사용자가 다이어트, 근육 증가, 건강 관리 등의 목표를 UI로 전송
참여객체	User → UI
파라미터	goal: 사용자가 설정한 목표 (예: “체중 감량”, “칼로리 제한”)

2

# 사용자목표 설정(사용자)

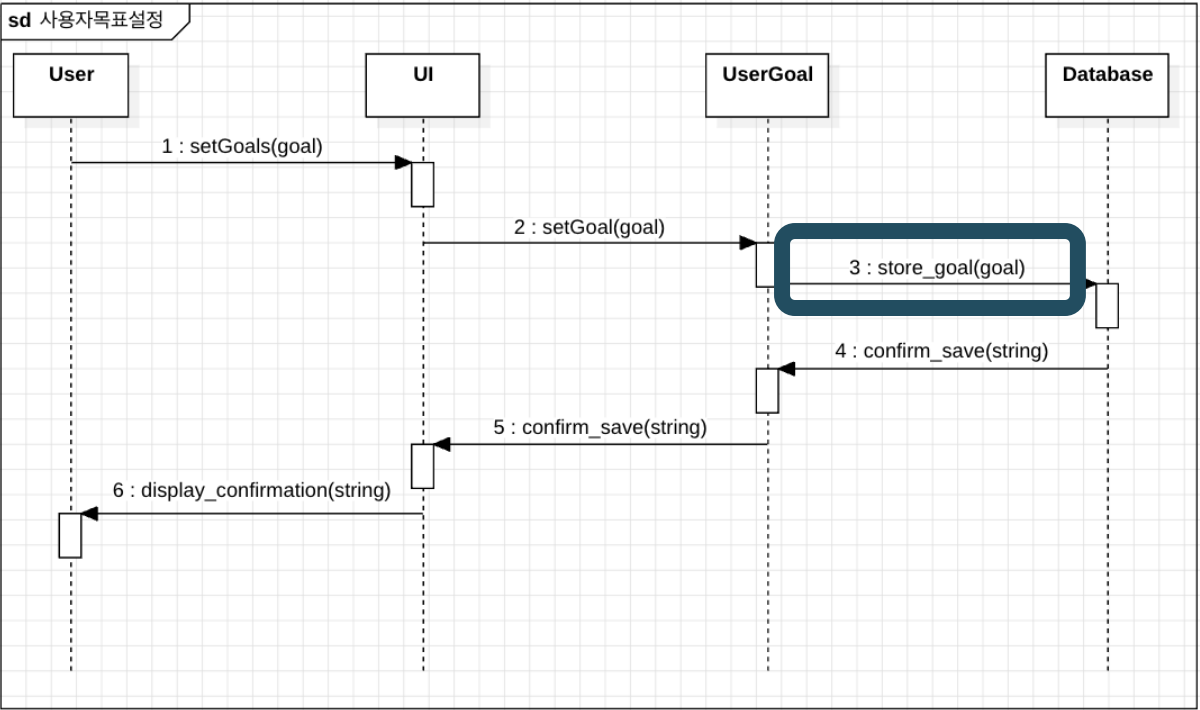


## setGoal(goal)

설명	UI가 UserGoal 클래스에 사용자가 설정한 목표를 전달하는 메서드
역할	사용자가 선택한 목표를 UserGoal 클래스로 전달하여 저장 요청을 시작
참여객체	UI → UserGoal
파라미터	goal: 사용자 목표 데이터

2

# 사용자목표 설정(사용자)

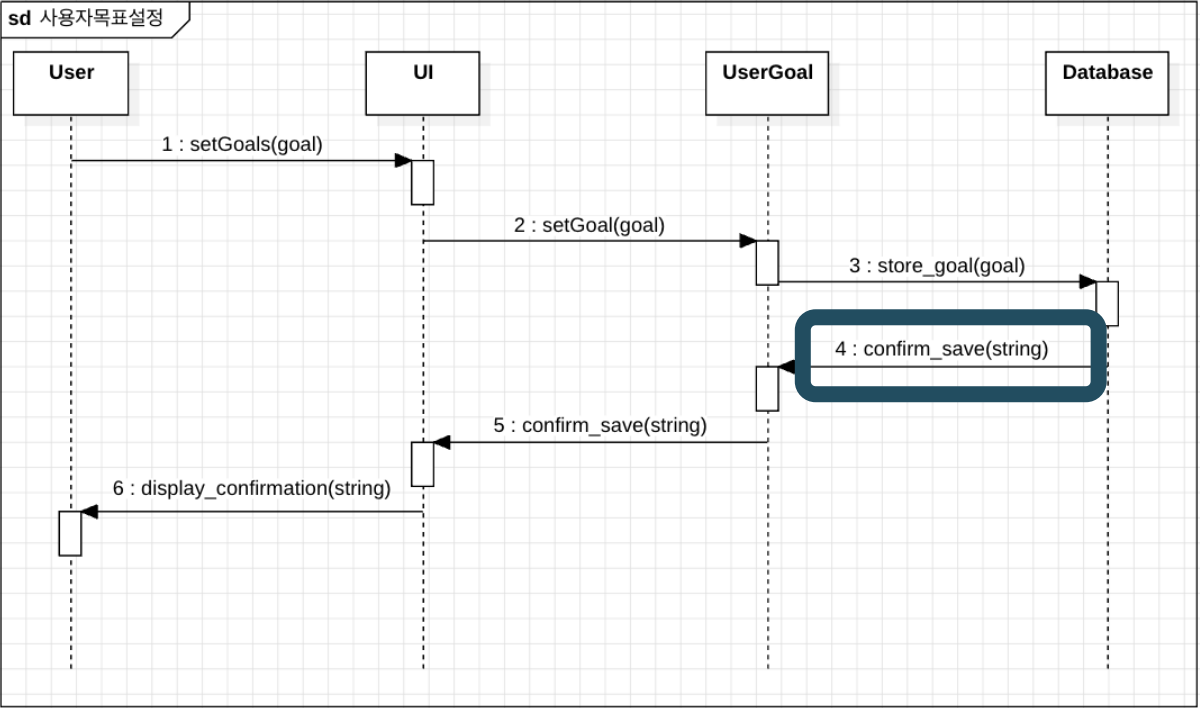


## store\_goal(goal)

설명	UserGoal 클래스가 데이터베이스에 목표를 저장하는 메서드
역할	사용자의 목표를 영구적으로 저장하기 위해 데이터베이스로 전송
참여객체	UserGoal → Database
파라미터	goal: 저장될 사용자 목표 데이터

2

# 사용자목표 설정(사용자)

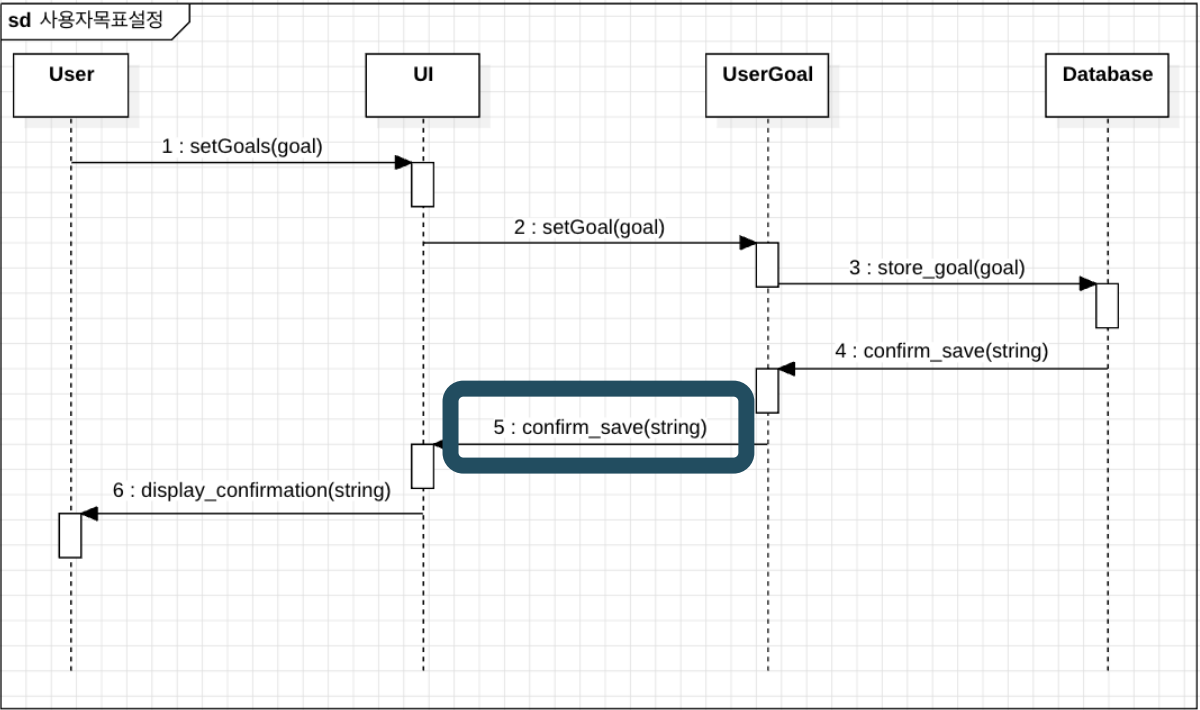


## confirm\_save(string)

설명	데이터베이스가 목표 저장 작업이 완료되었음을 UserGoal에 알리는 메서드
역할	목표 저장 성공 여부를 응답
참여객체	Database → UserGoal
파라미터	string: 저장 완료 메시지 (예: "목표가 저장되었습니다")

2

# 사용자목표 설정(사용자)



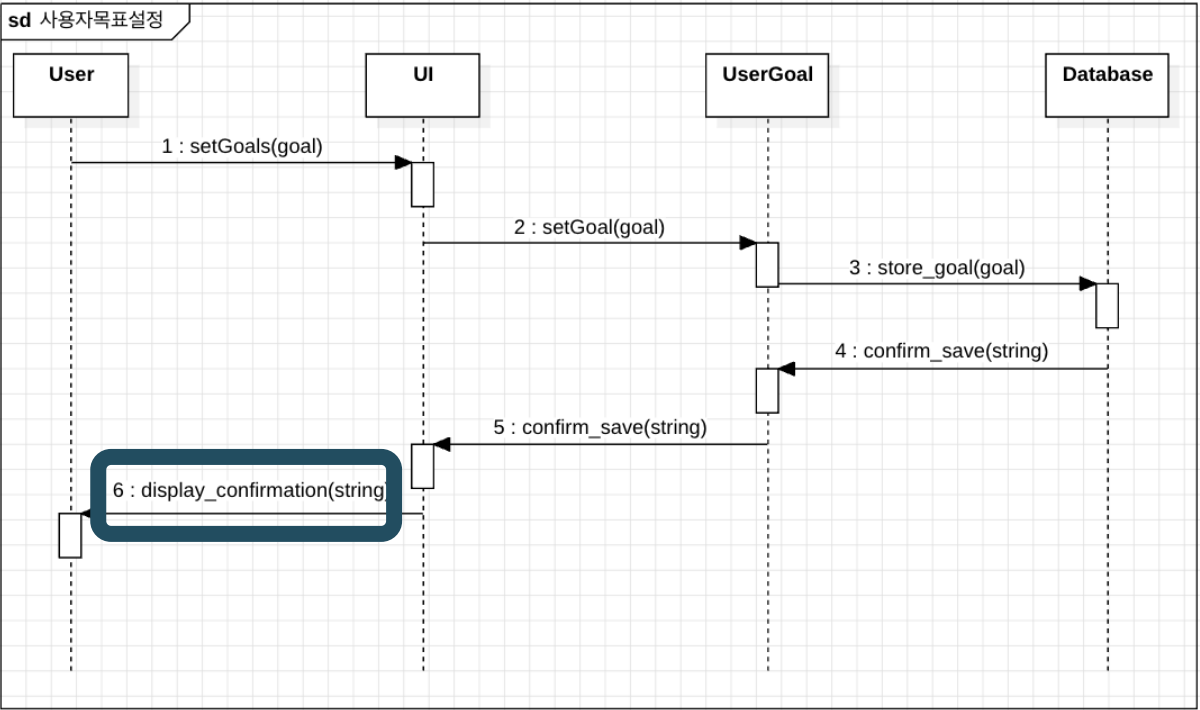
## confirm\_save(string)

설명	UserGoal이 저장 성공 여부를 UI에 알리는 메서드
역할	목표 저장 작업이 성공했음을 사용자에게 전달
참여객체	UserGoal → UI
파라미터	string: 사용자에게 전달될 확인 메시지



2

# 사용자목표 설정(사용자)

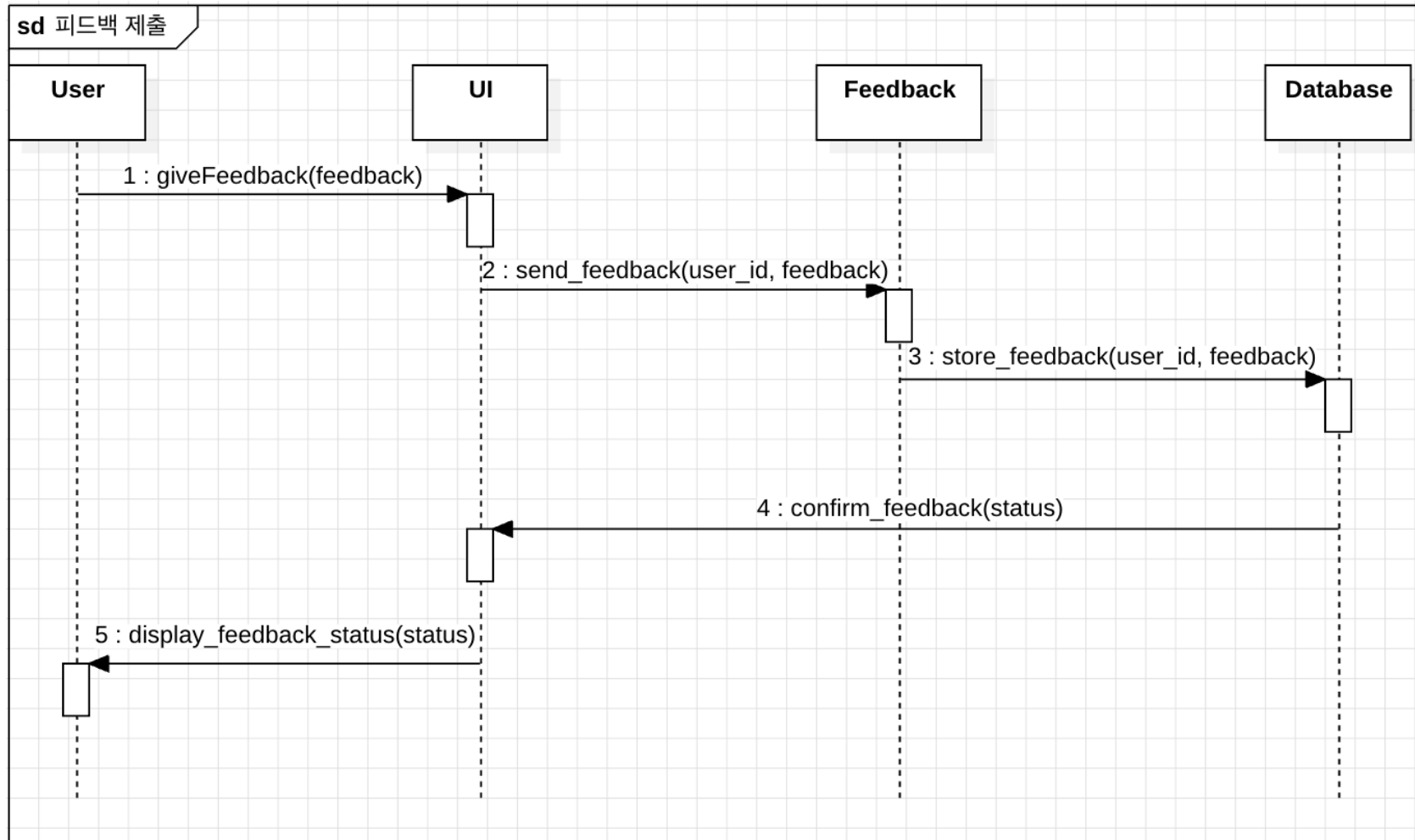


## display\_confirmation(string)

설명	UI가 사용자에게 목표 저장 완료 메시지를 화면에 표시하는 메서드
역할	저장이 완료되었음을 사용자가 확인
참여객체	UI → User
파라미터	string: 사용자에게 표시될 확인 메시지 (예: "목표 설정이 완료되었습니다")

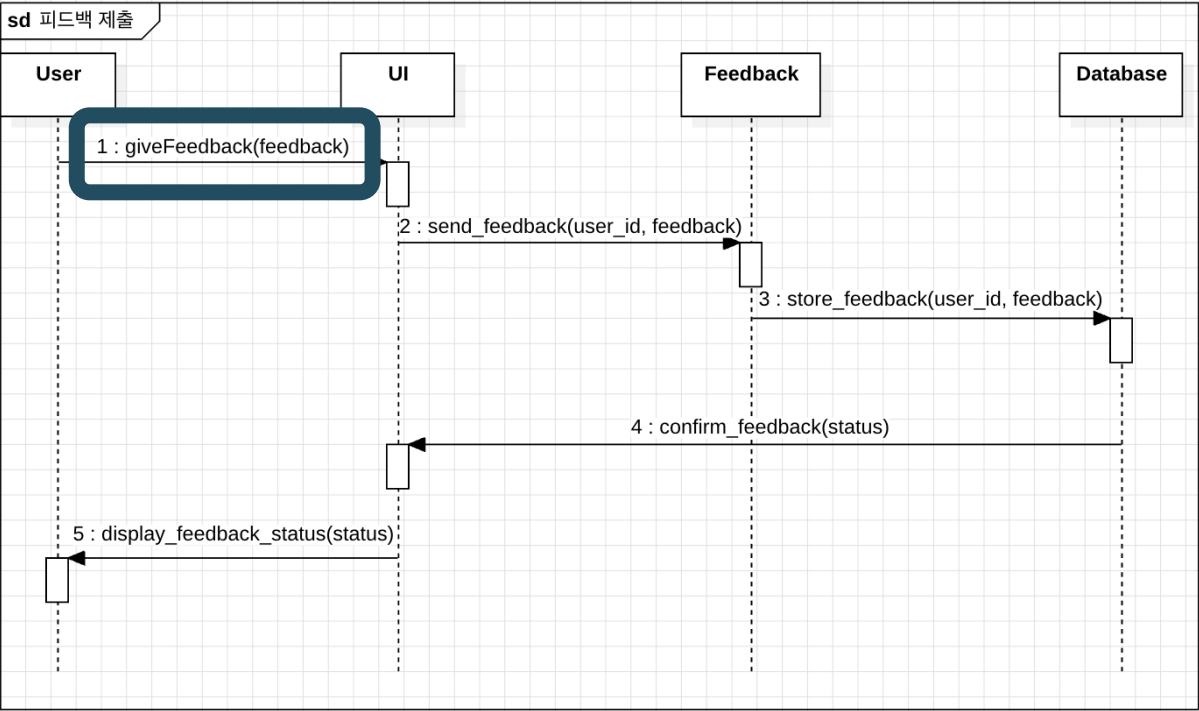
3

## 피드백 제출(사용자)



3

# 피드백제출(사용자)

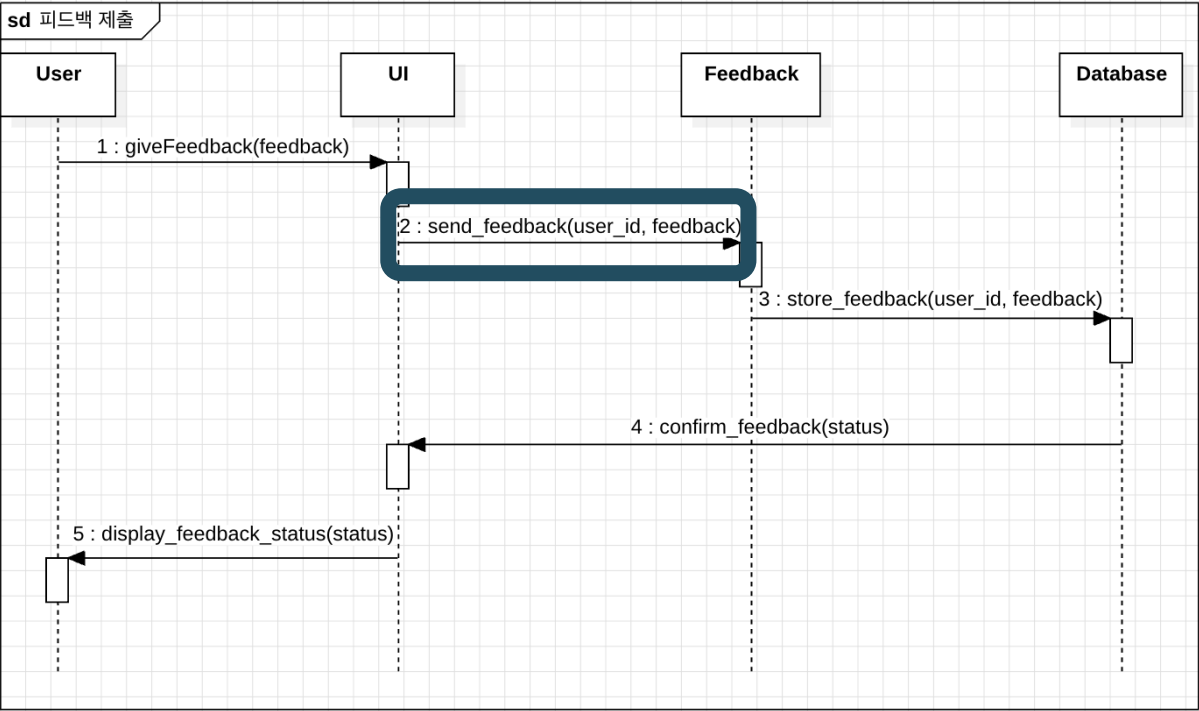


## giveFeedback(feedback)

설명	사용자가 작성한 피드백 데이터를 UI로 전달하는 메서드
역할	사용자가 피드백을 제출하기 위해 필요한 초기 요청
참여객체	User → UI
파라미터	feedback: 사용자가 입력한 피드백 내용

3

# 피드백제출(사용자)

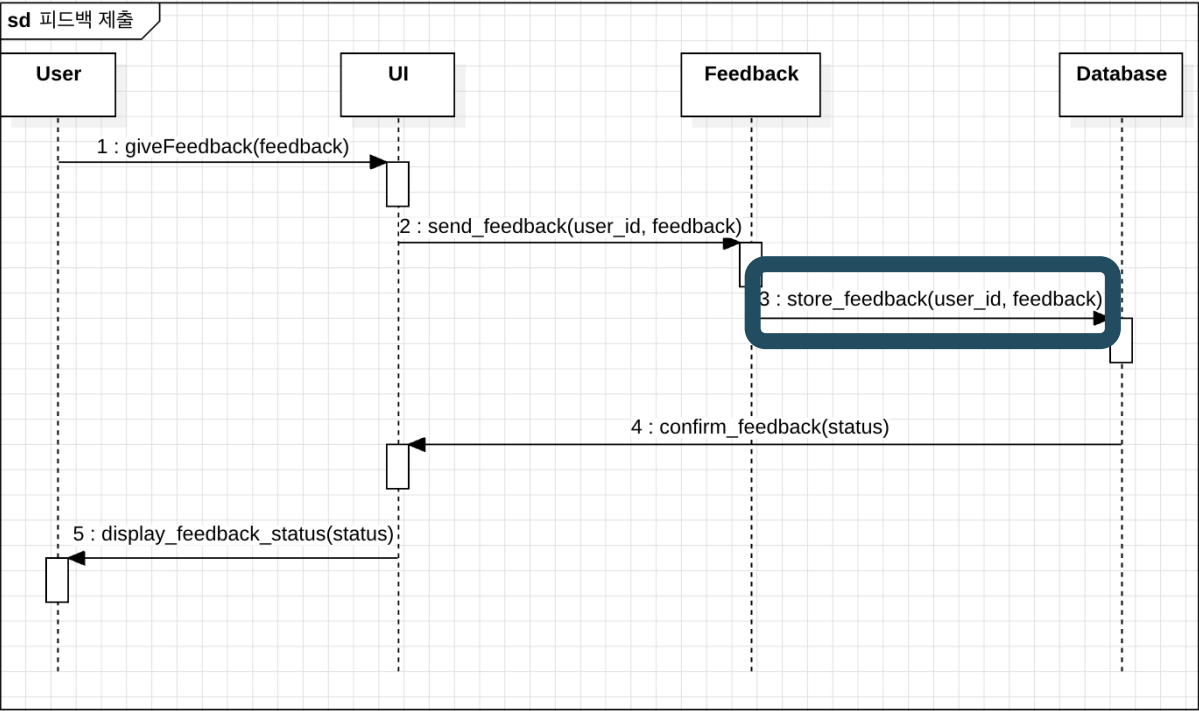


## send\_feedback(user\_id, feedback)

설명	UI가 피드백 데이터와 사용자 ID를 Feedback 클래스로 전달하는 메서드
역할	피드백 데이터를 Feedback 클래스에 전달하여 저장 요청
참여객체	UI → Feedback
파라미터	user_id: 피드백을 보낸 사용자의 고유 ID feedback: 사용자가 입력한 피드백 내용

3

# 피드백제출(사용자)

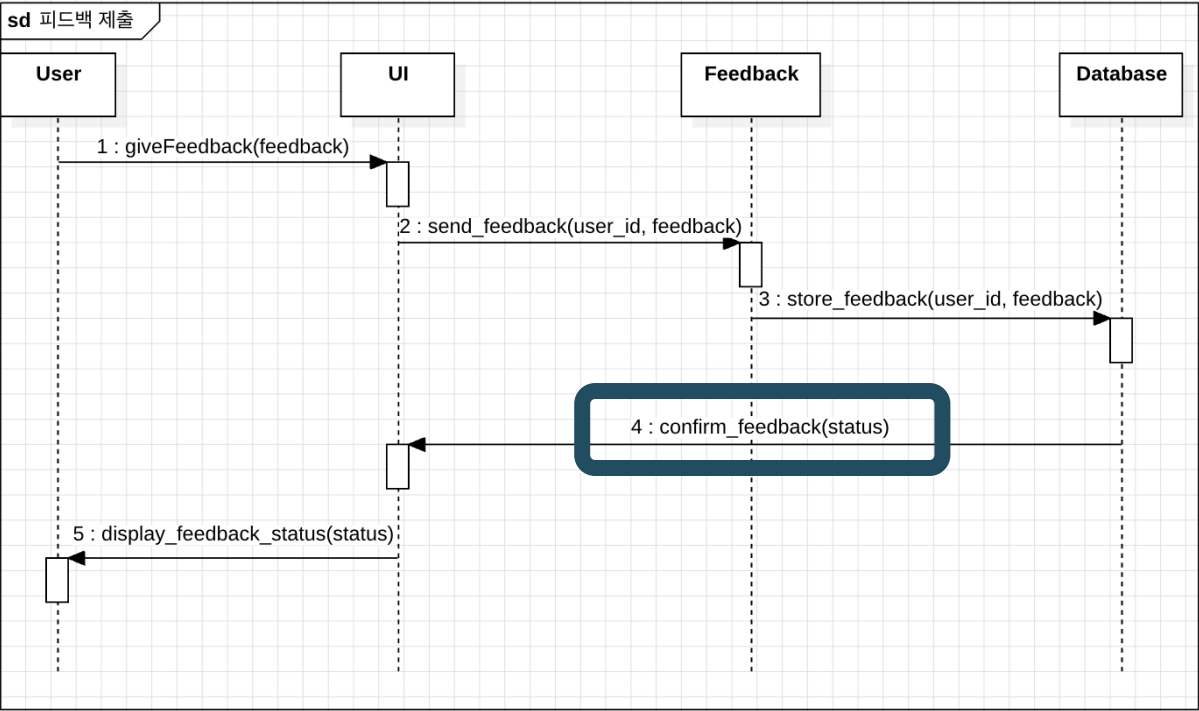


## store\_feedback(user\_id, feedback)

설명	Feedback 클래스가 데이터베이스에 피드백 데이터를 저장하는 메서드
역할	사용자 피드백과 관련 정보를 영구적으로 저장
참여객체	Feedback → Database
파라미터	user_id: 피드백을 보낸 사용자의 고유 ID feedback: 저장될 피드백 내용

3

# 피드백제출(사용자)

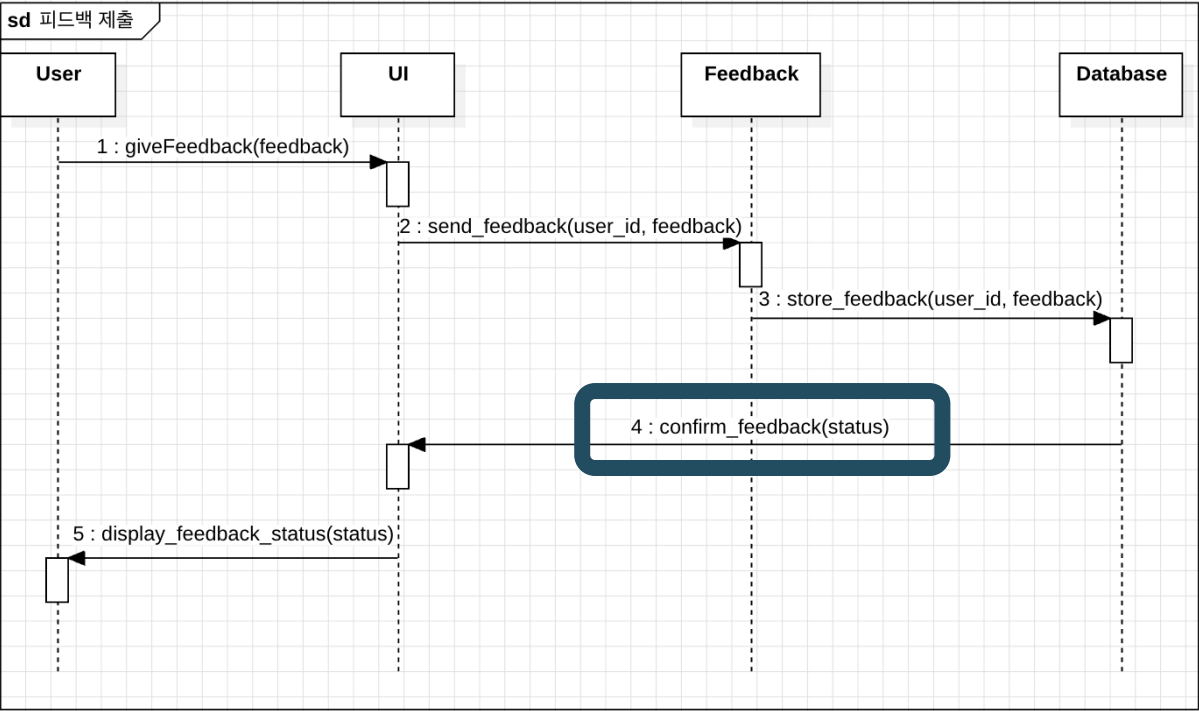


## confirm\_feedback(status)

설명	데이터베이스가 피드백 저장 성공 여부를 Feedback 클래스에 알리는 메서드
역할	저장 성공 메시지 전달
참여객체	Database → Feedback
파라미터	status: 저장 성공 여부를 나타내는 상태 (예: "저장 완료")

3

# 피드백제출(사용자)

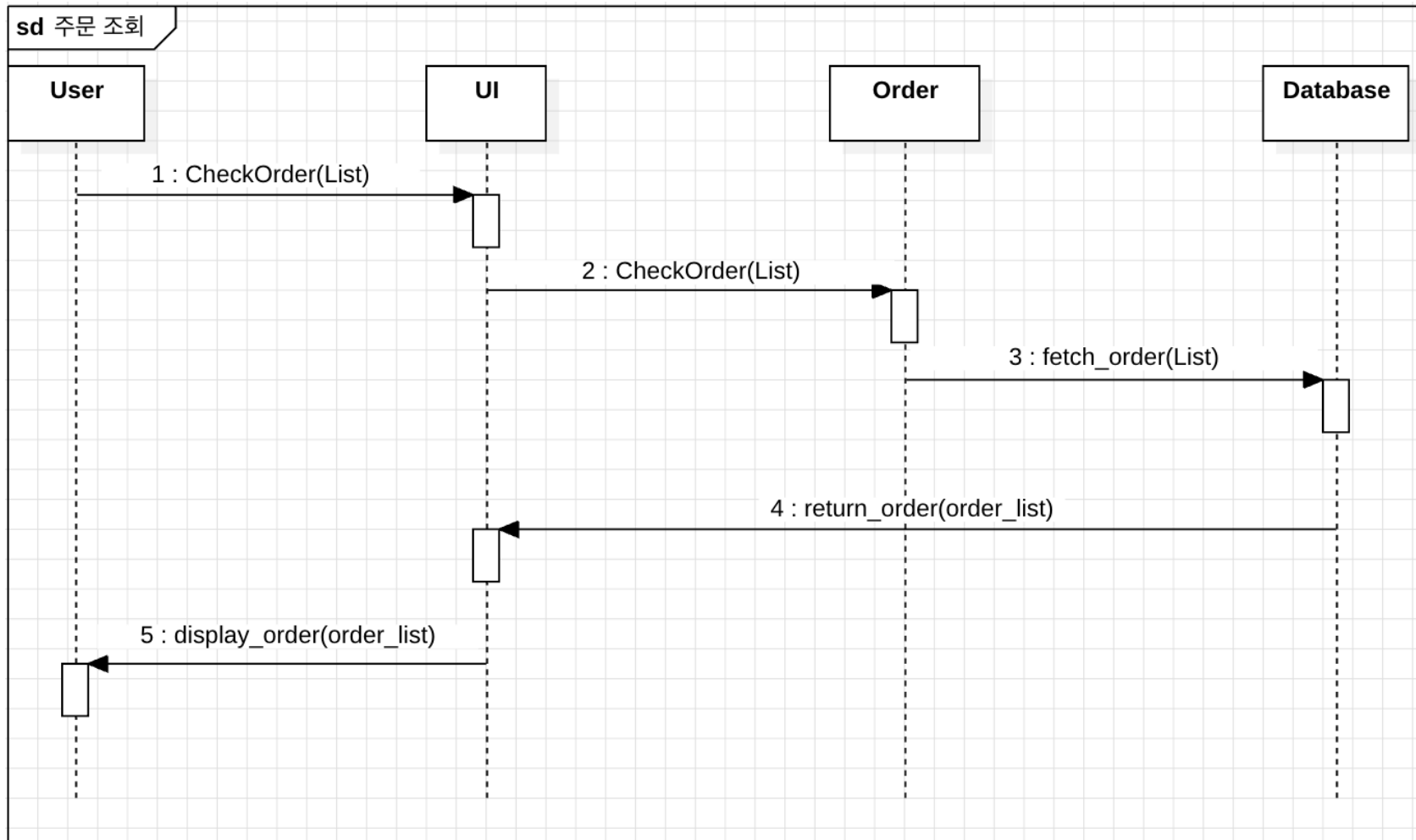


## display\_feedback\_status(status)

설명	UI가 피드백 저장 완료 상태를 사용자에게 보여주는 메서드
역할	피드백 저장 성공 여부를 사용자에게 시각적으로 알림
참여객체	Feedback → UI → User
파라미터	status: 저장 상태 메시지 (예: “피드백이 성공적으로 저장되었습니다”)

4

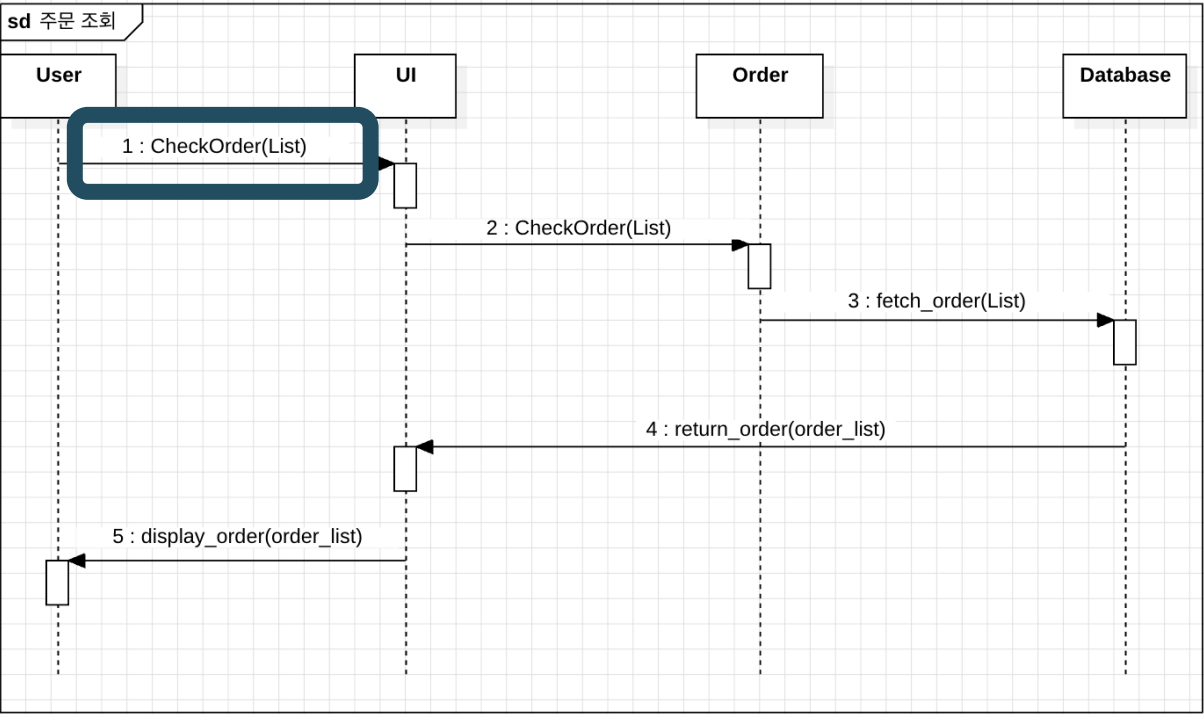
## 주문 조회(사용자)





4

# 주문 조회(사용자)

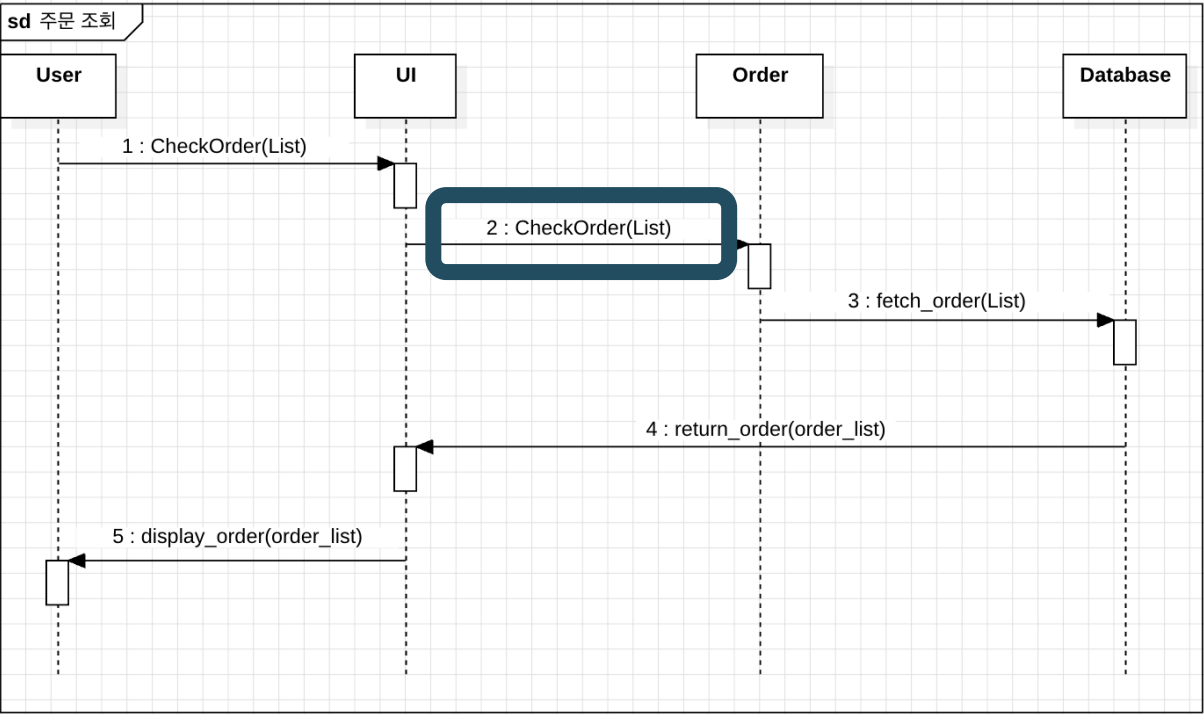


## CheckOrder(List)

설명	사용자가 주문 내역을 확인하기 위해 UI에 요청
역할	사용자가 자신의 주문 내역을 확인하기 위한 초기 요청 단계
참여객체	User → UI
파라미터	List: 확인하고자 하는 주문의 리스트

4

# 주문 조회(사용자)

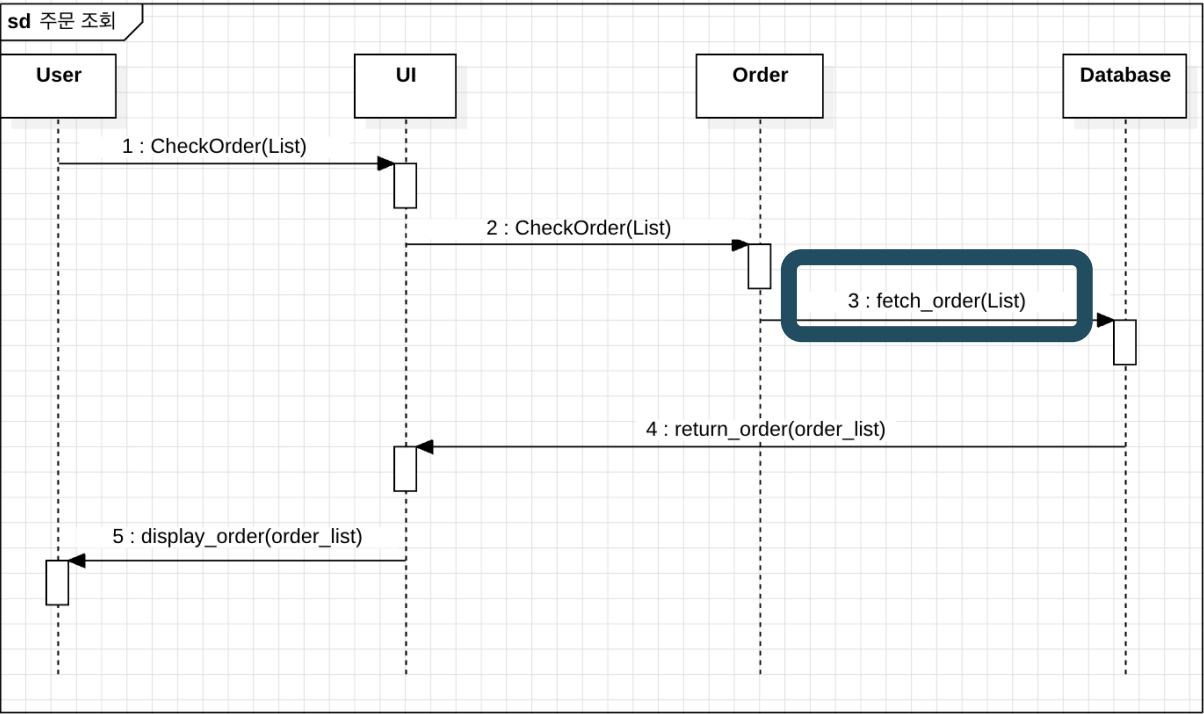


## CheckOrder(List)

설명	UI가 주문 조회 요청을 Order 클래스에 전달
역할	사용자가 요청한 주문 리스트 정보를 Order 클래스로 넘김
참여객체	UI → Order
파라미터	List: 사용자가 조회하려는 주문 리스트

4

# 주문 조회(사용자)

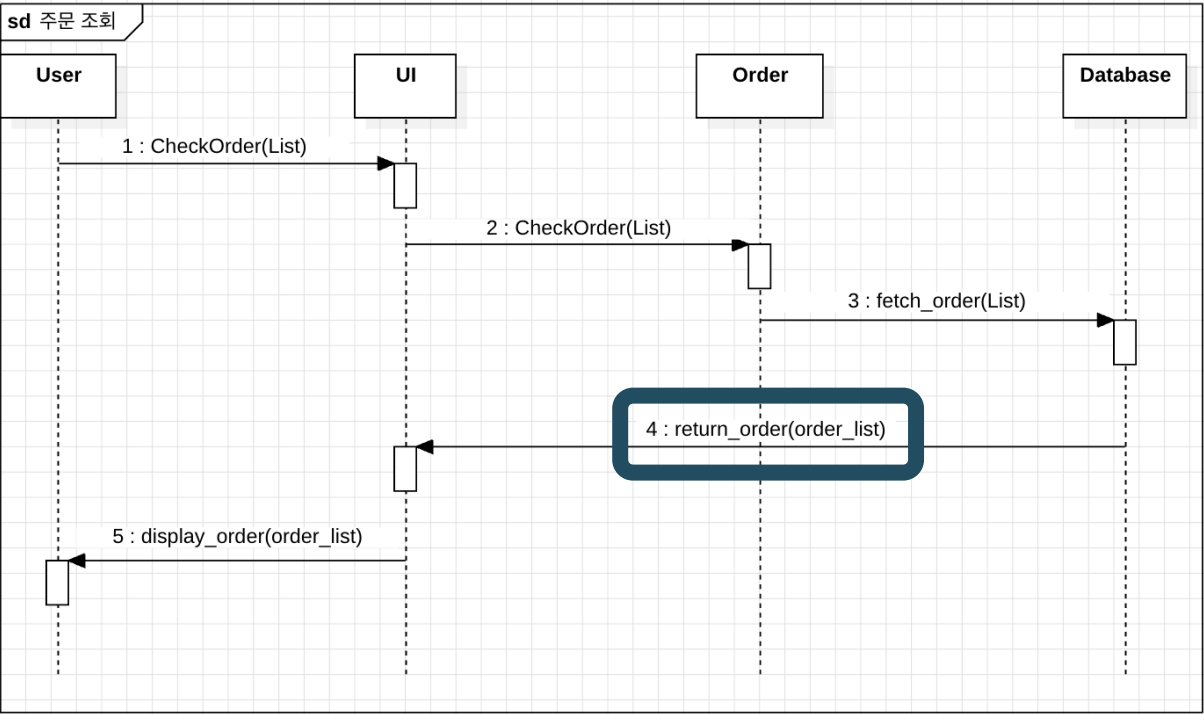


## fetch\_order(List)

설명	Order 클래스가 데이터베이스에서 주문 데이터를 가져옴
역할	주문 내역 데이터를 데이터베이스에서 검색 및 조회
참여객체	Order → Database
파라미터	List: 데이터베이스에서 가져올 주문 리스트

4

# 주문 조회(사용자)

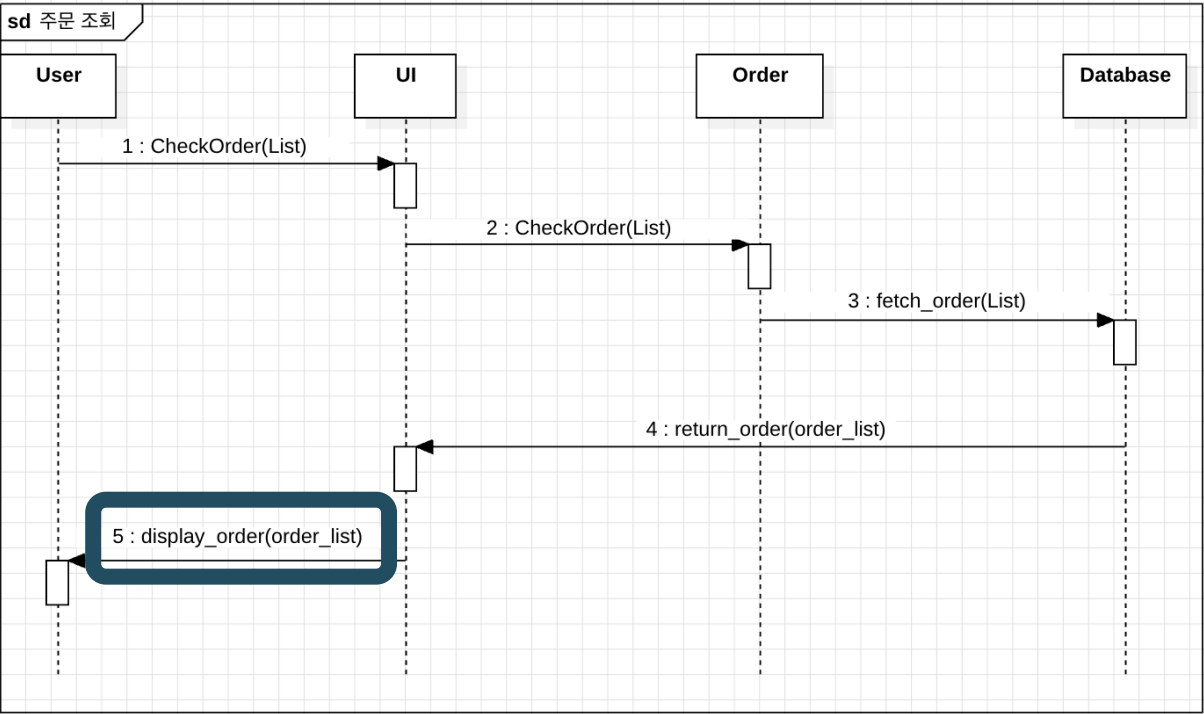


## return\_order(order\_list)

설명	데이터베이스에서 조회된 주문 내역 데이터를 Order 클래스에 반환
역할	데이터베이스에서 조회된 주문 데이터를 반환
참여객체	Database → Order
파라미터	order_list: 조회된 주문 내역 리스트

4

# 주문 조회(사용자)

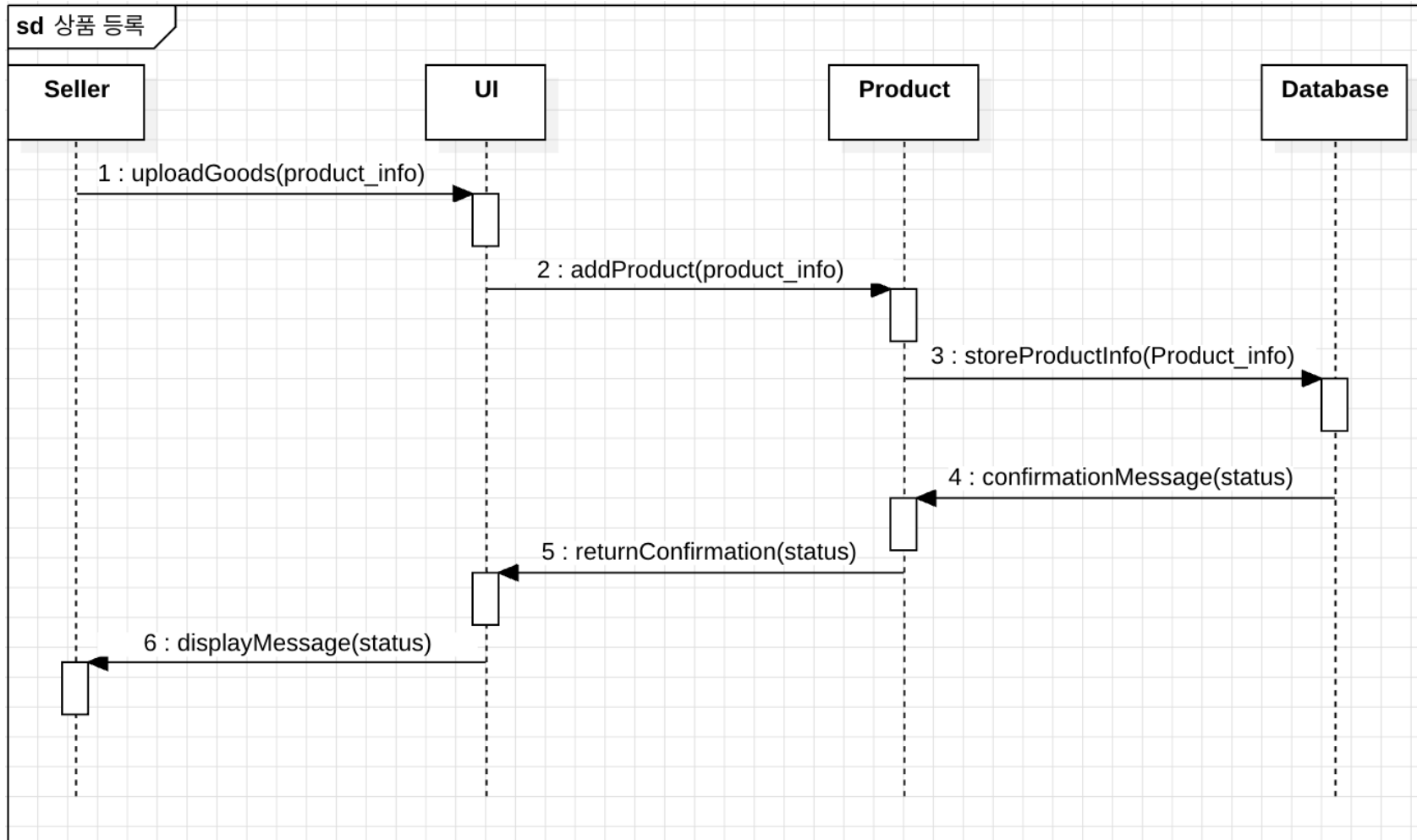


## display\_order(order\_list)

설명	UI가 사용자가 요청한 주문 내역을 화면에 표시
역할	조회된 주문 내역을 사용자에게 표시
참여객체	Order → UI → User
파라미터	order_list: 사용자에게 표시할 주문 내역 리스트

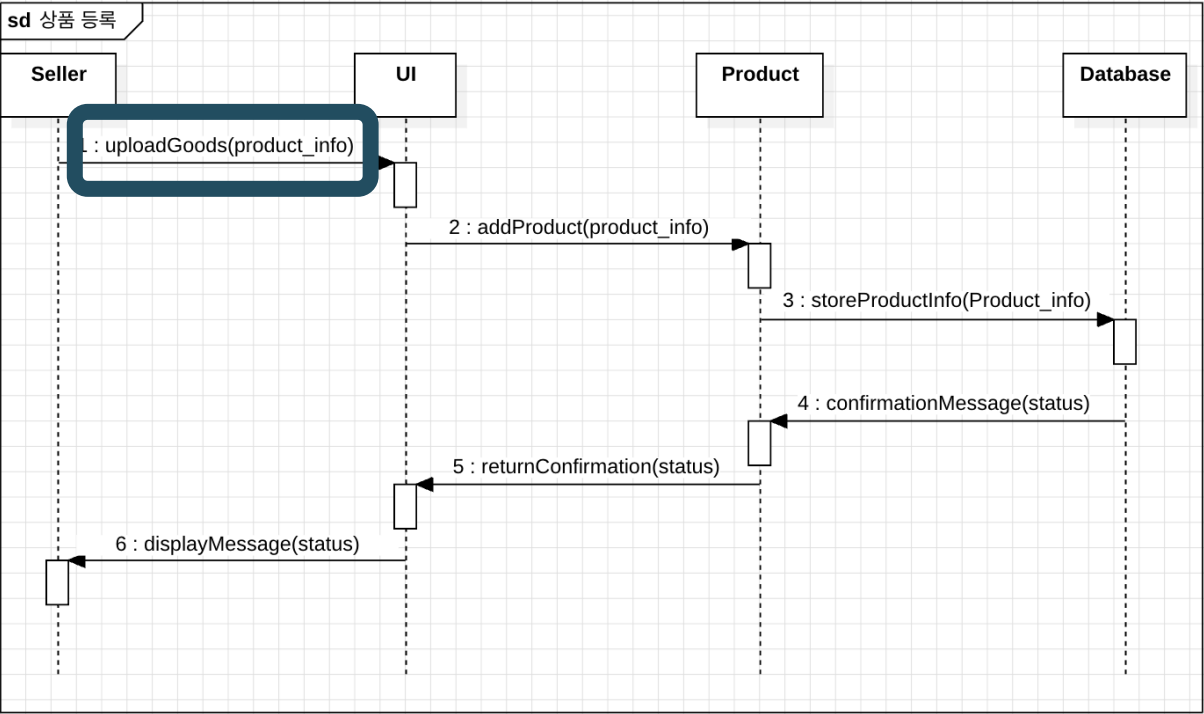
5

## 상품 등록(판매자)



5

상품등록(판매자)

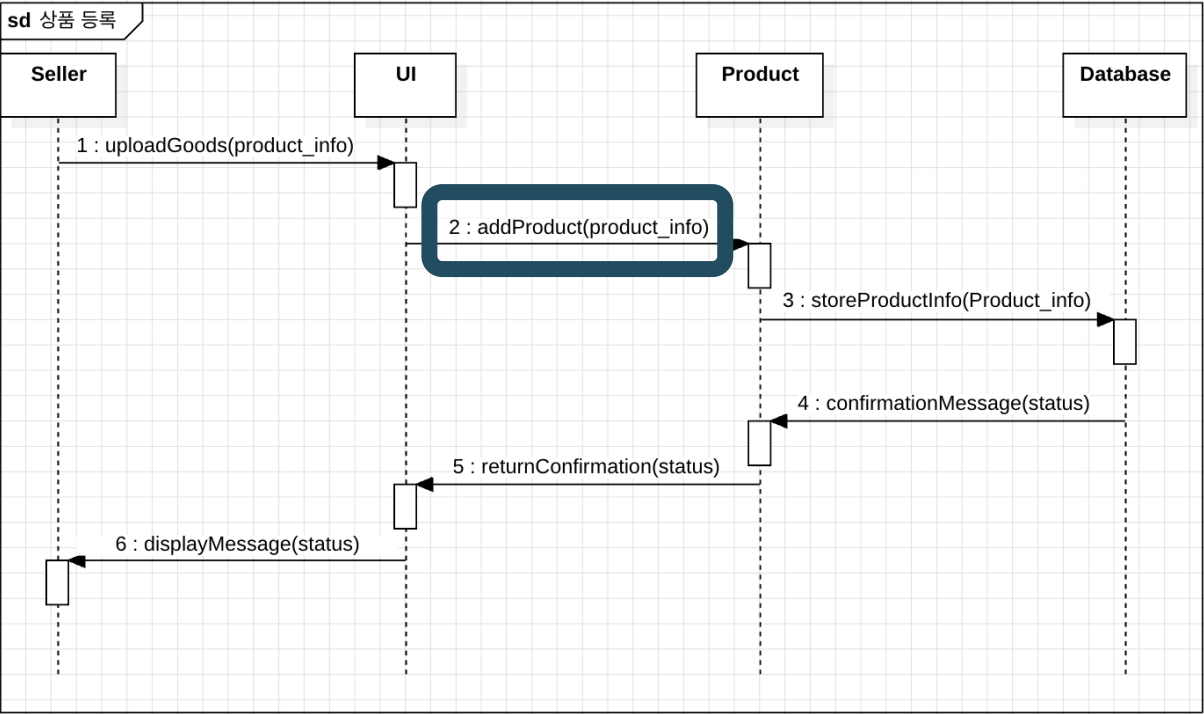


uploadGoods(product\_info)

설명	판매자가 상품 정보를 입력하여 UI에 상품 등록 요청
역할	상품 등록 프로세스의 시작 단계
참여객체	Seller → UI
파라미터	product_info: 등록하고자 하는 상품에 대한 정보

5

상품등록(판매자)



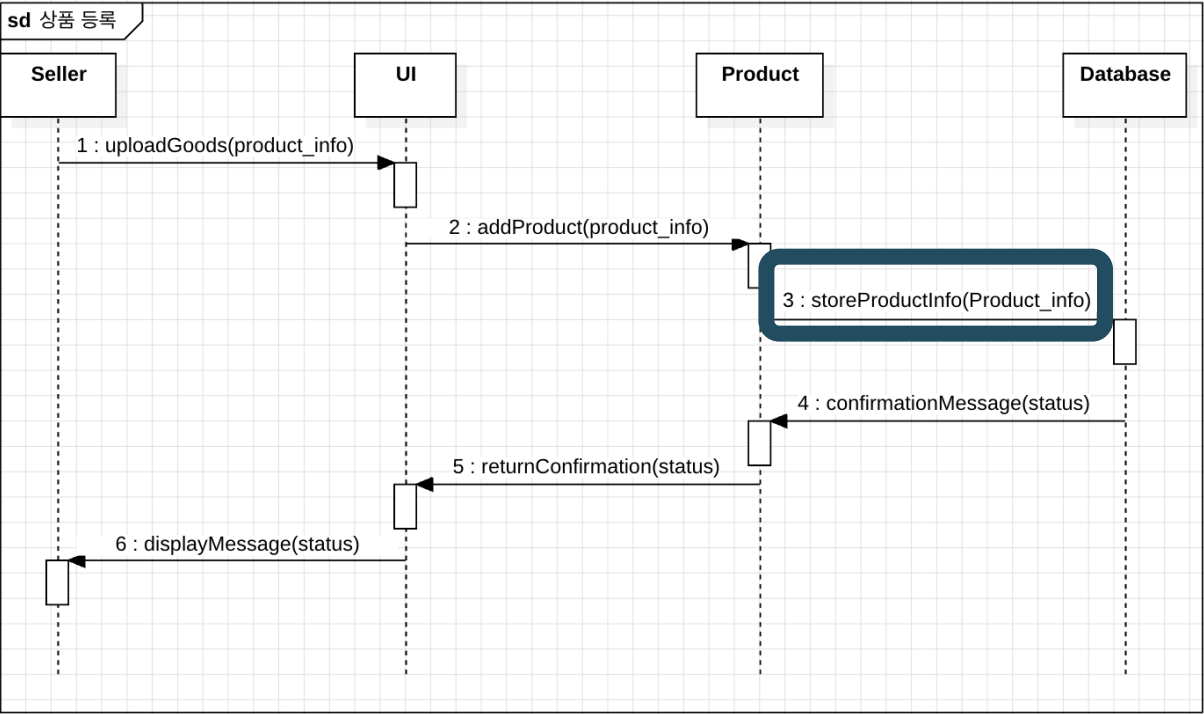
addProduct(product\_info)

설명	UI가 입력된 상품 정보를 Product 클래스에 전달
역할	판매자의 입력값을 시스템 내부로 전달
참여객체	UI → Product
파라미터	product_info: 상품의 이름, 가격, 설명 등 상품 관련 정보



5

상품등록(판매자)

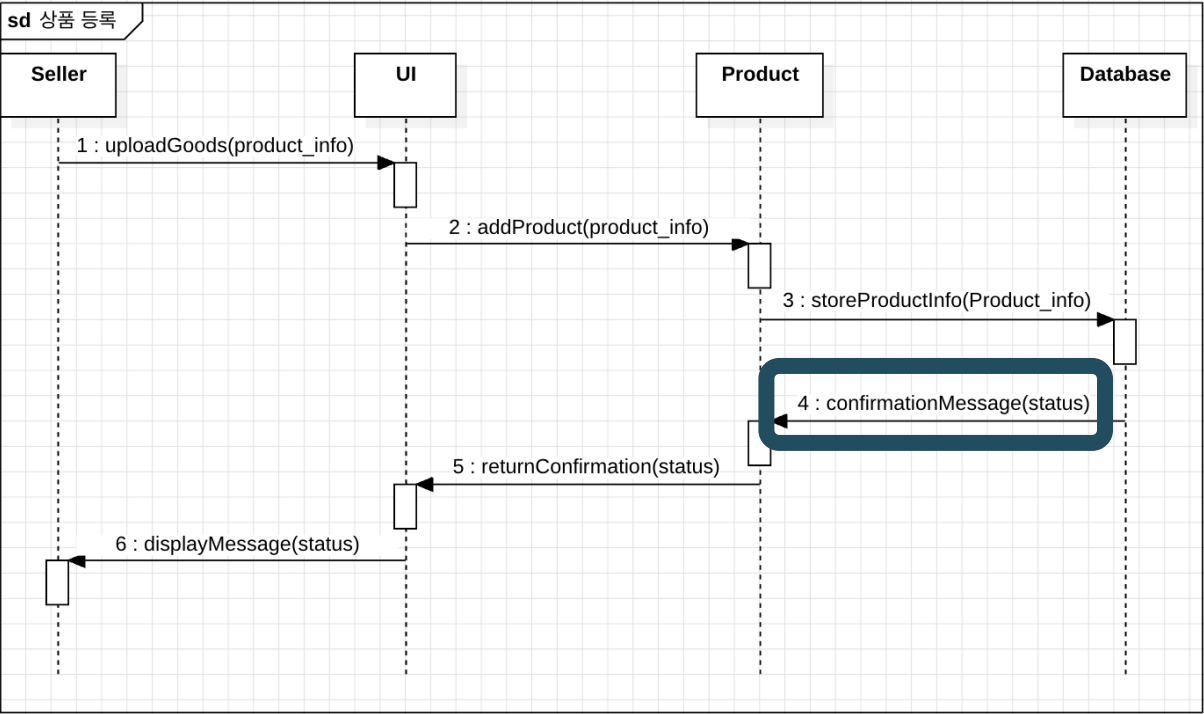


storeProductInfo(product\_info)

설명	Product 클래스가 상품 정보를 데이터베이스에 저장
역할	상품 정보 영구 저장 처리
참여객체	Product → Database
파라미터	product_info: 데이터베이스에 저장될 상품 정보

5

상품등록(판매자)

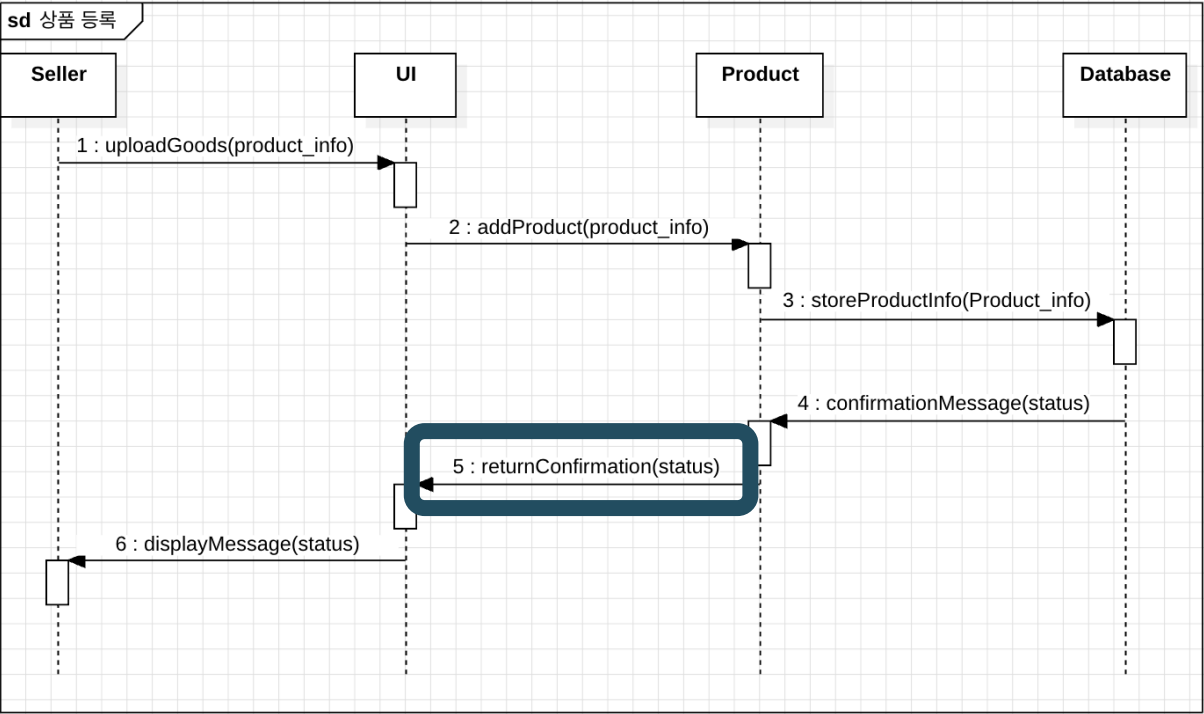


confirmationMessage(status)

설명	데이터베이스가 상품 정보 저장 상태를 Product 클래스에 전달
역할	상품 정보가 성공적으로 저장되었음을 알림
참여객체	Database → Product
파라미터	status: 저장 성공 여부를 나타내는 상태 메시지

5

상품등록(판매자)

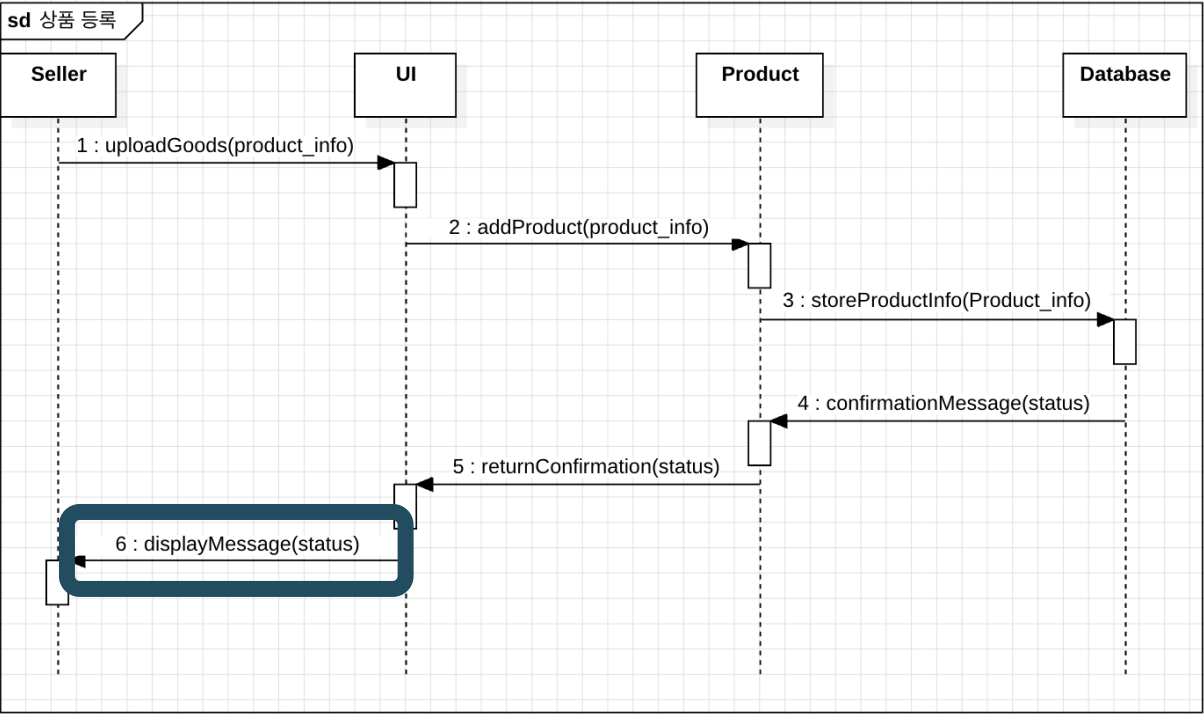


returnConfirmation(status)

설명	Product 클래스가 상품 저장 결과를 UI에 반환
역할	UI로 상태 정보를 전달하여 사용자에게 표시
참여객체	Product → UI
파라미터	status: 저장 성공 여부를 나타내는 상태 메시지

5

# 상품등록(판매자)

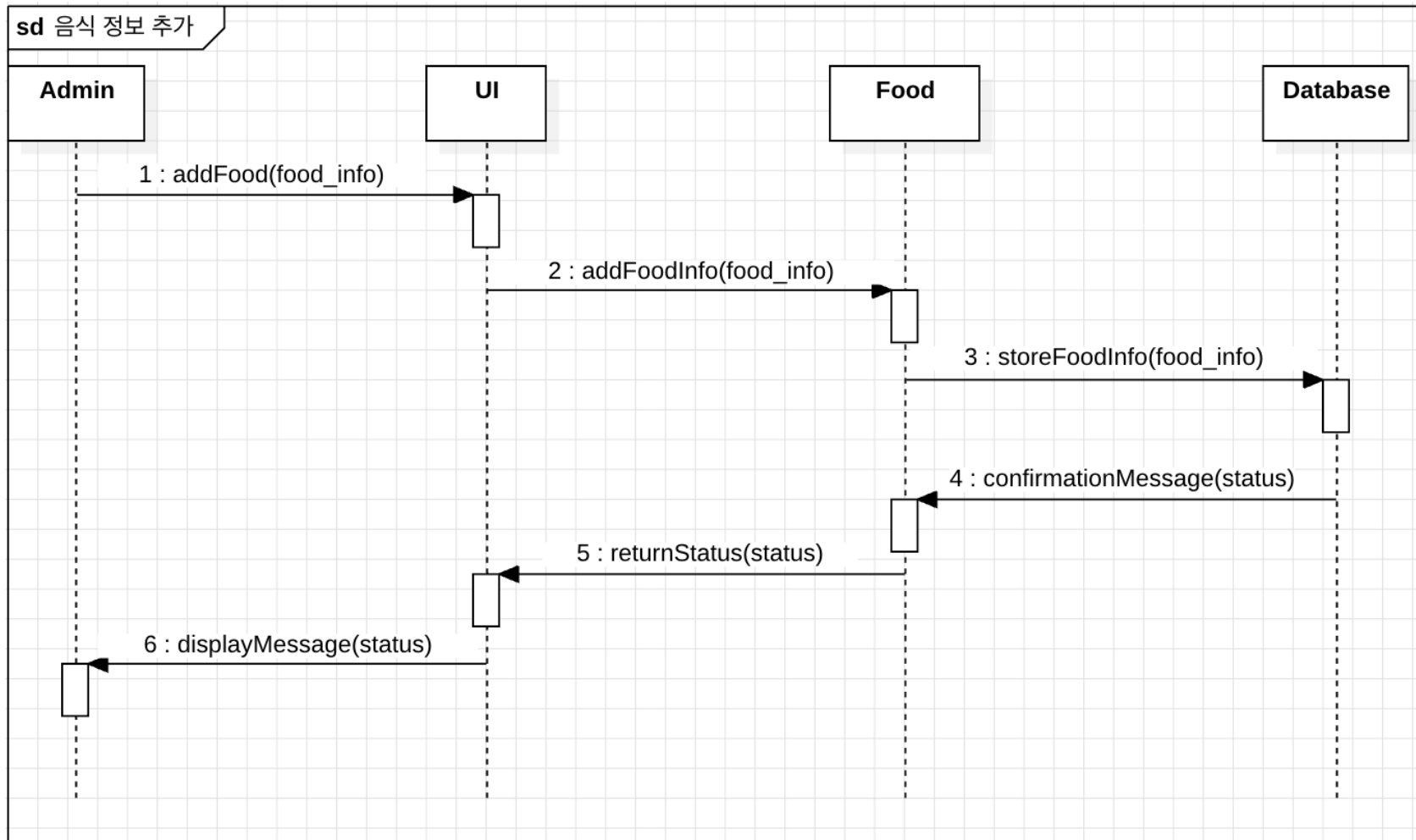


## displayMessage(status)

설명	UI가 저장 결과 메시지를 판매자에게 화면에 표시
역할	상품 등록 성공 여부를 판매자에게 전달
참여객체	UI → Seller
파라미터	status: 상품 등록 성공 여부 메시지

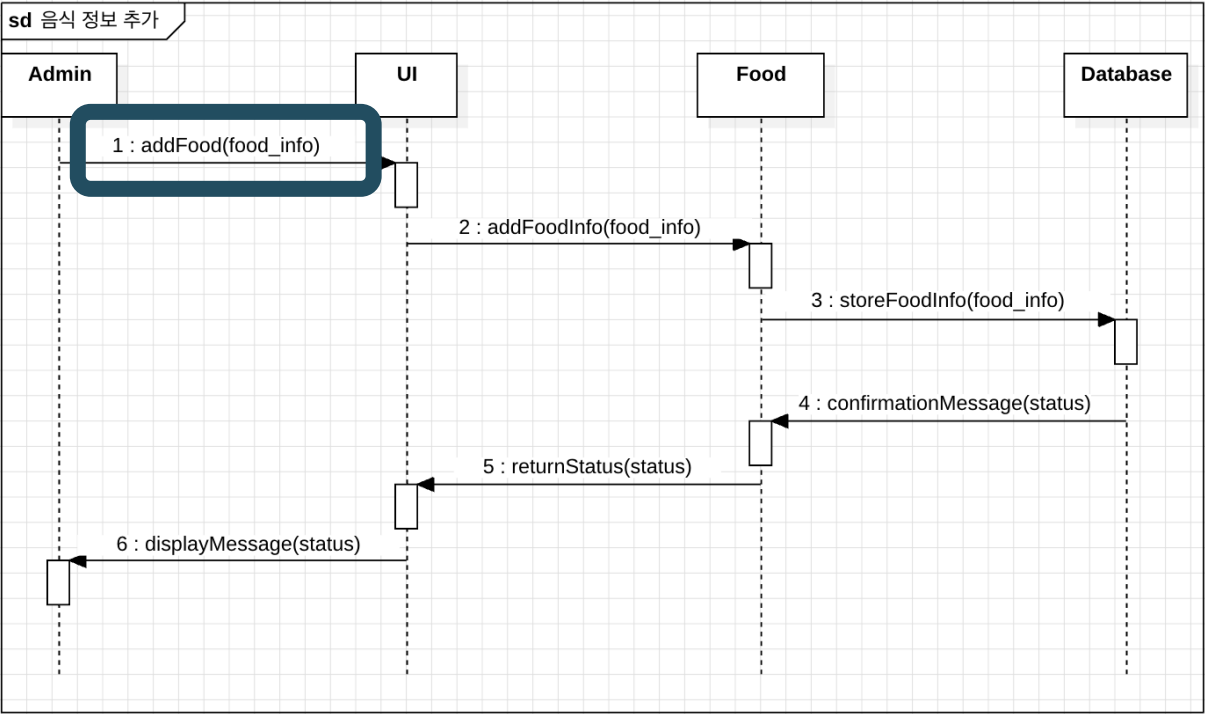
6

## 음식 정보 추가(관리자)



6

# 음식 정보 추가(관리자)

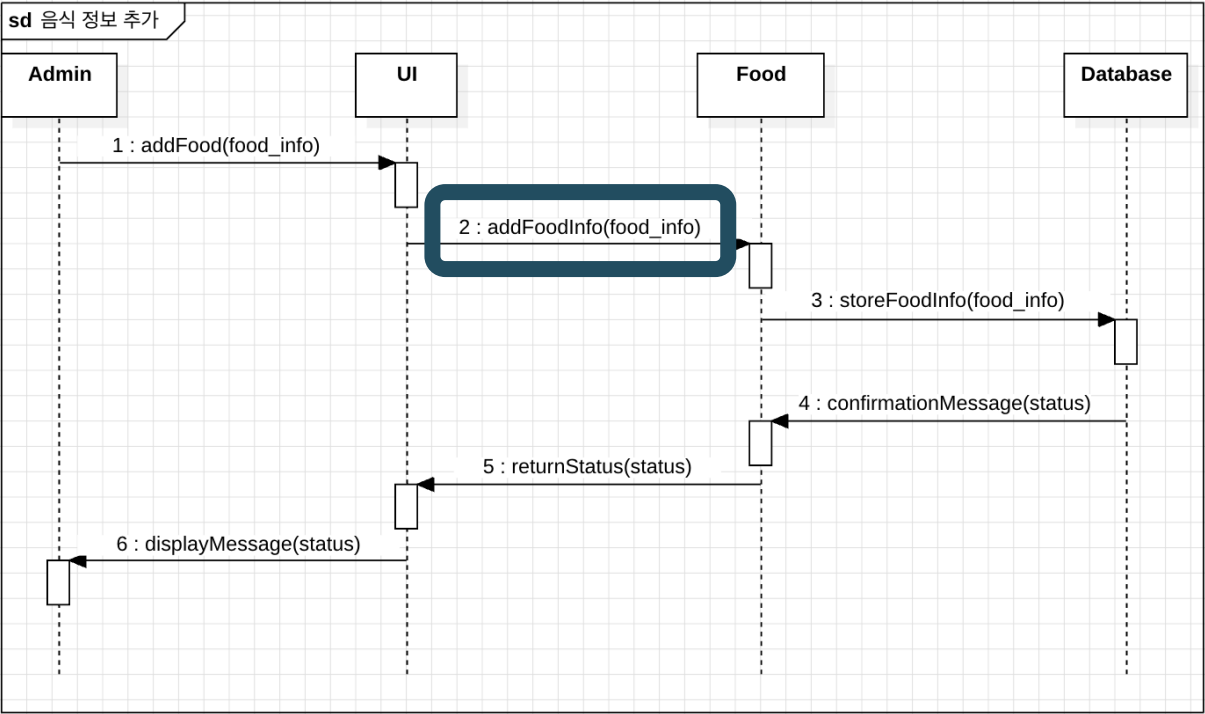


## addFood(food\_info)

설명	관리자가 UI를 통해 새로운 음식 정보를 추가 요청
역할	음식 추가 프로세스의 시작 단계
참여객체	Admin → UI
파라미터	food_info: 추가하려는 음식의 정보 (예: 이름, 칼로리, 영양소 등)

6

음식 정보 추가(관리자)

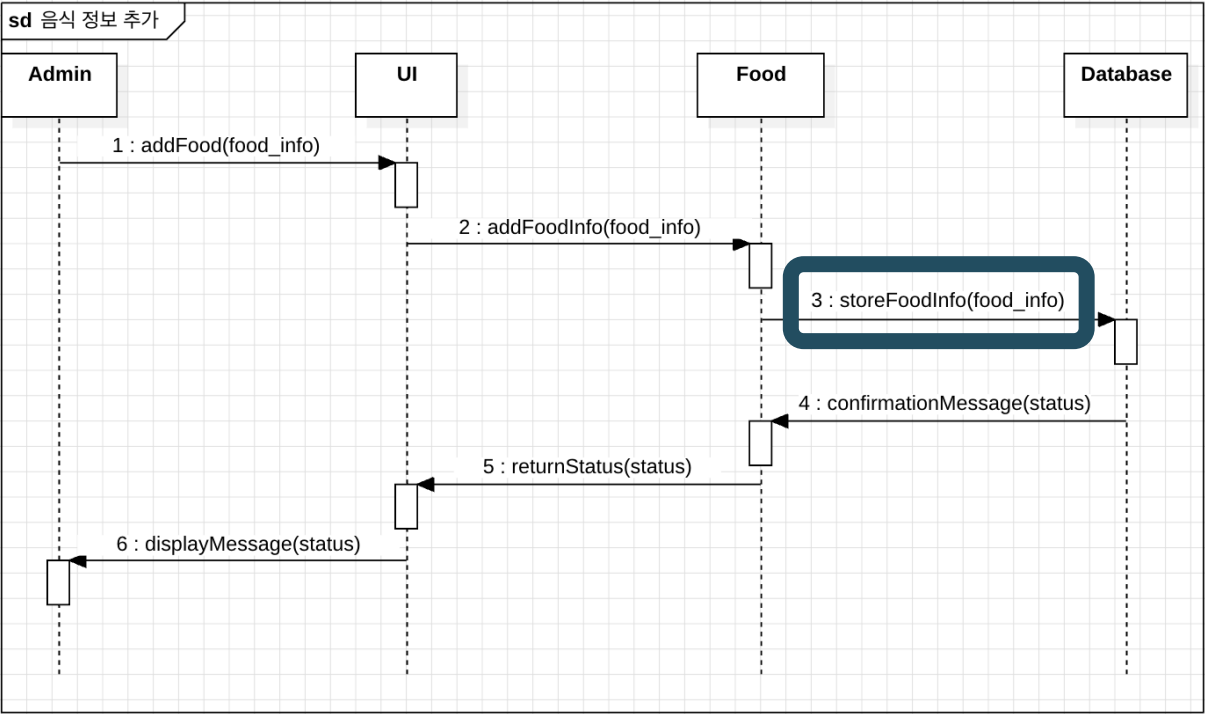


addFood(food\_info)

설명	UI가 입력된 음식 정보를 Food 클래스에 전달
역할	관리자의 입력값을 시스템 내부로 전달
참여객체	UI → Food
파라미터	food_info: 음식 이름, 영양 성분 등 음식에 대한 상세 정보

6

# 음식 정보 추가(관리자)



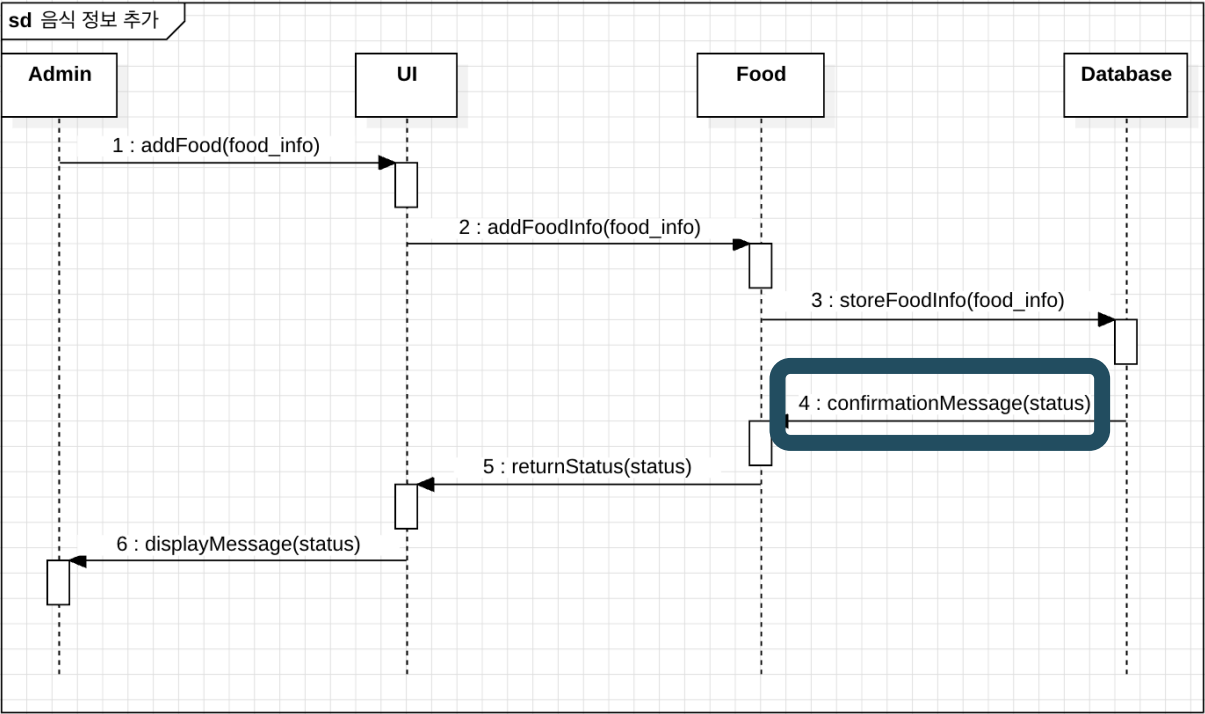
## storeFoodInfo(food\_info)

설명	Food 클래스가 전달받은 음식 정보를 데이터베이스에 저장
역할	음식 정보를 영구 저장
참여객체	Food → Database
파라미터	food_info: 데이터베이스에 저장될 음식 정보



6

# 음식 정보 추가(관리자)

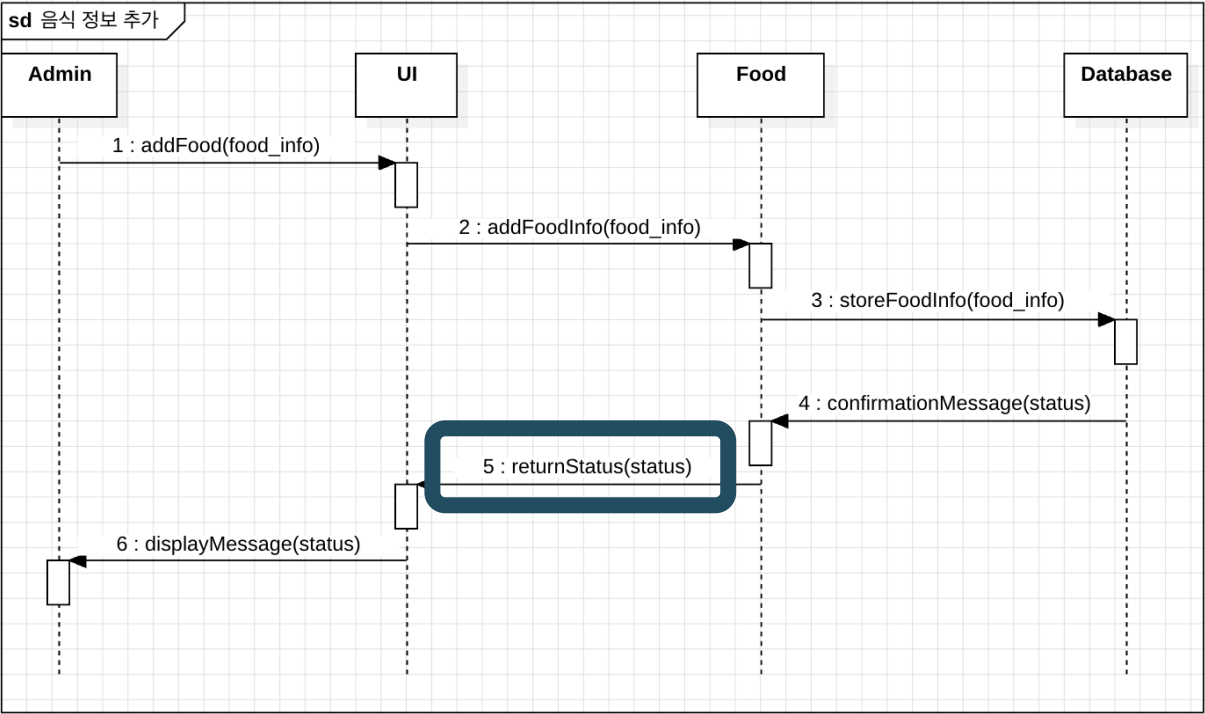


## confirmationMessage(status)

설명	데이터베이스가 저장 상태를 Food 클래스에 전달
역할	음식 정보가 성공적으로 저장되었음을 알림
참여객체	Database → Food
파라미터	status: 저장 성공 여부를 나타내는 상태 메시지

6

# 음식 정보 추가(관리자)

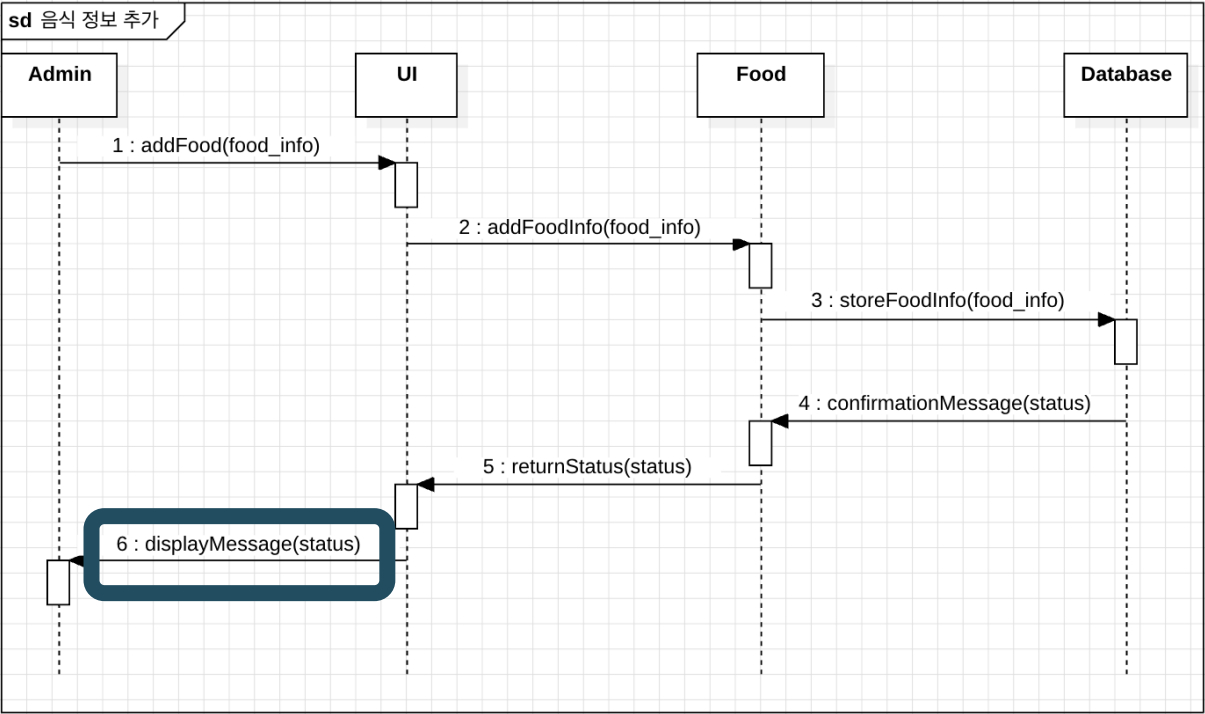


## returnStatus(status)

설명	Food 클래스가 저장 결과를 UI에 반환
역할	UI로 상태 정보를 전달하여 관리자에게 알림
참여객체	Food → UI
파라미터	status: 음식 정보 저장 성공 여부 메시지

6

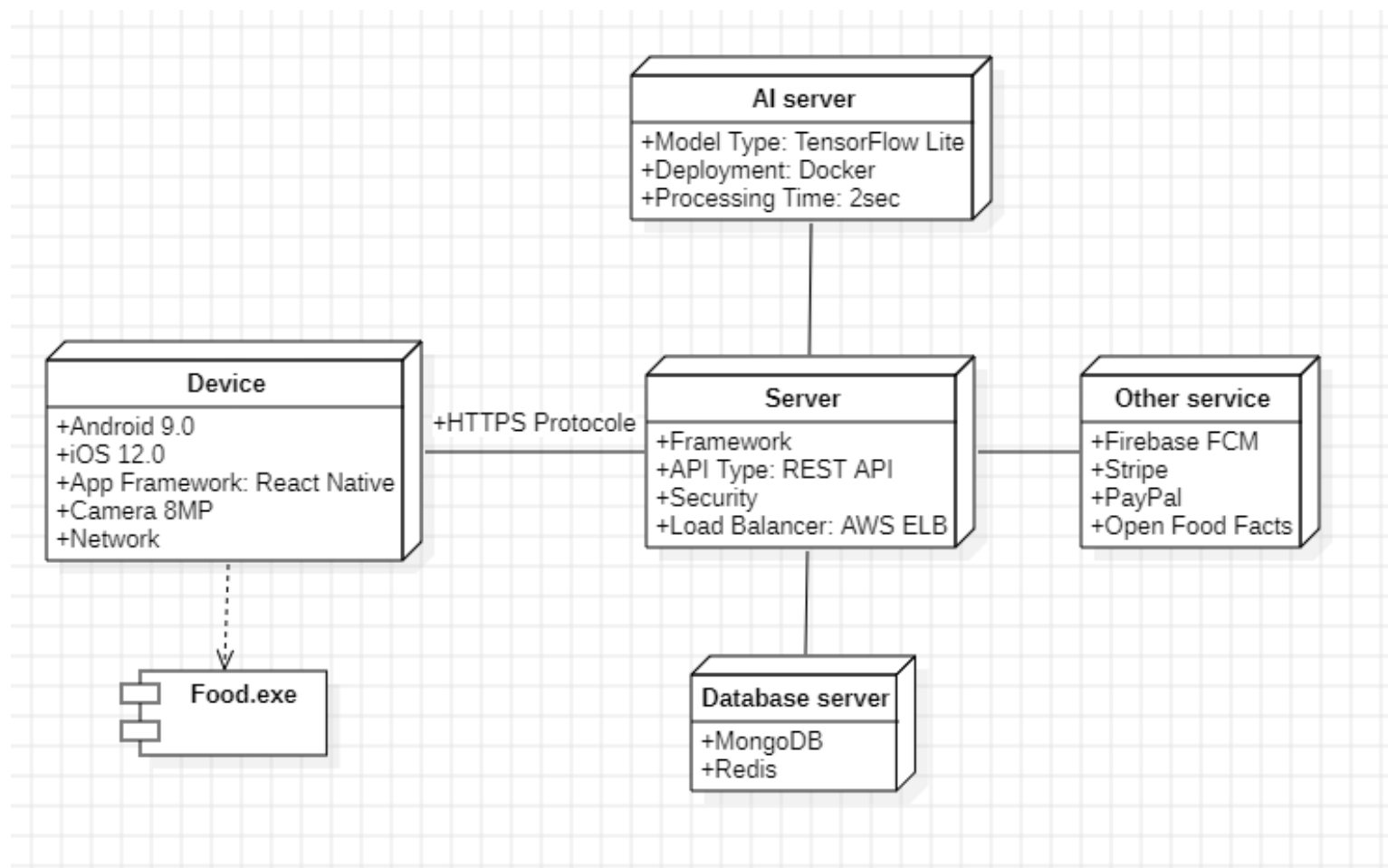
# 음식 정보 추가(관리자)



## displayMessage(status)

설명	UI가 저장 결과 메시지를 관리자에게 화면에 표시
역할	음식 정보 추가 성공 여부를 관리자에게 전달
참여객체	UI → Admin
파라미터	status: 음식 정보 추가 성공 여부 메시지

# 배치 다이어그램



# 1 초기 설정 화면



앱 시작 화면



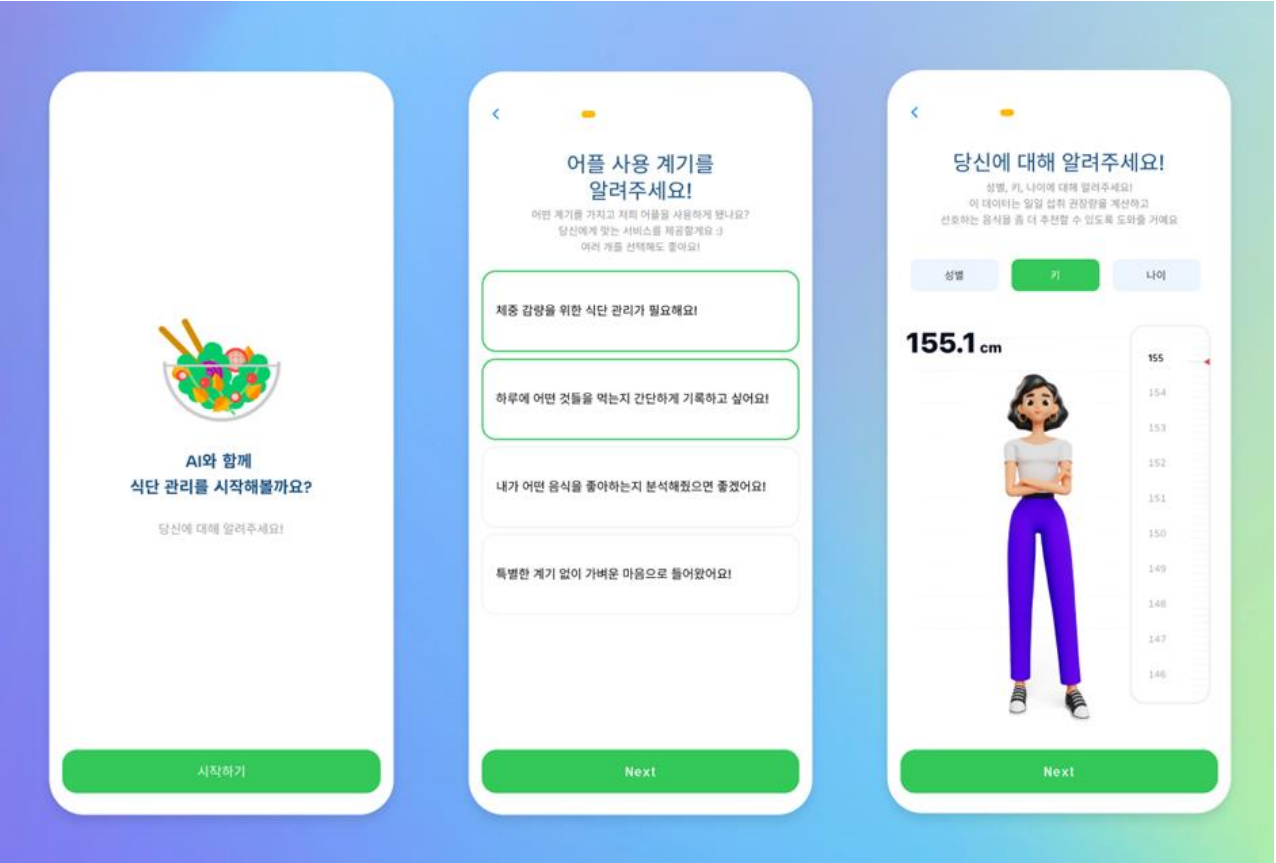
앱 사용 목적 설정 화면



기본 사용자정보입력

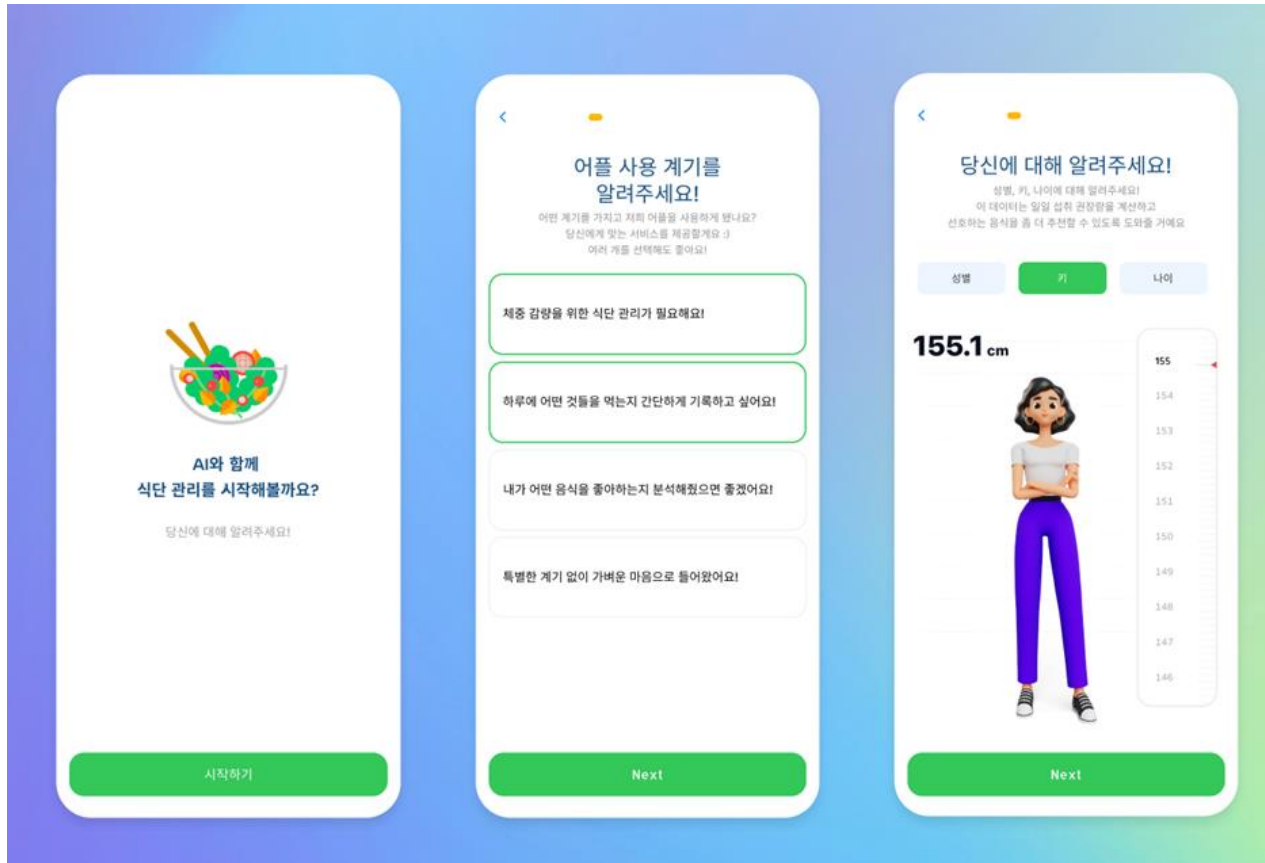
1

# 초기 설정 화면



설명	앱의 초기 화면으로, 사용자가 앱을 처음 사용하는 단계에서 정보를 입력하고 개인화된 서비스를 설정하는 과정
구성 기능	<div>1. 앱 소개 및 시작 버튼 제공<ul style="list-style-type: none"><li>“AI와 함께 식단 관리를 시작해볼까요?”라는 메시지와 함께 앱의 목적을 소개.</li><li>하단의 “시작하기” 버튼 클릭 시 다음 화면 이동</li></ul></div> <div>2. 사용 계기 선택<ul style="list-style-type: none"><li>사용자가 앱을 사용하는 이유를 선택하도록 유도</li><li>체중 감량, 식단 기록, 음식 분석 등 옵션 제공, 선택 결과는 앱의 초기 맞춤 서비스에 반영</li></ul></div> <div>3. 기본 정보 입력<ul style="list-style-type: none"><li>사용자의 성별, 키, 나이를 입력 받는 화면</li><li>입력 값에 따라 아바타(캐릭터)가 실시간으로 변화하여 직관적 피드백 제공</li><li>데이터는 사용자 맞춤 피드백과 통계 생성 활용</li></ul></div>

# 1 초기 설정 화면



## 주요 기능

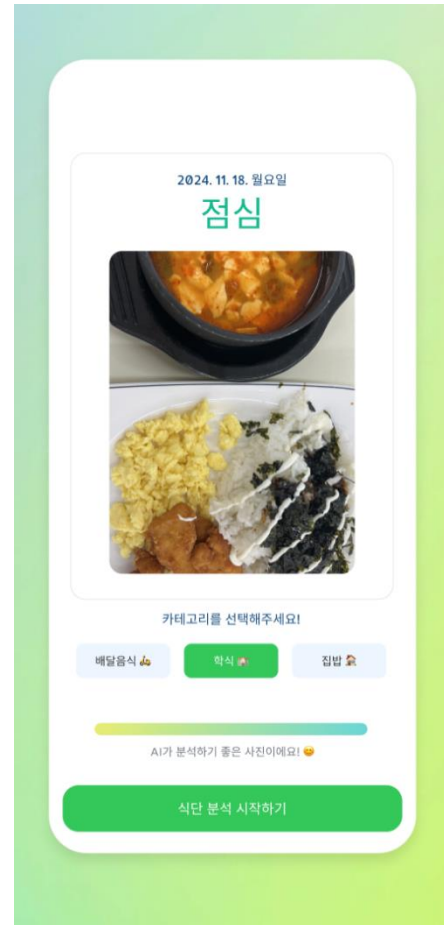
- 앱 소개와 사용 시작  
앱의 목적과 기능을 간략히 소개하고 사용자가 본격적으로 설정을 시작할 수 있도록 유도
- 개인화 설정  
사용자의 사용 목적과 기본 정보를 통해 개인화된 데이터 설정 및 서비스를 시작.

## 흐름

1. 사용자: "시작하기" 버튼 클릭 → 다음 화면 이동
2. 사용자: "어플 사용 계기를 알려주세요!" 선택 항목 클릭 → 선택 결과는 서버에 저장
3. 사용자: 성별, 키, 나이 입력 → 다음 화면으로 이동

2

## 사용자 정보 입력 및 데이터 수집 화면



식단 업로드 화면

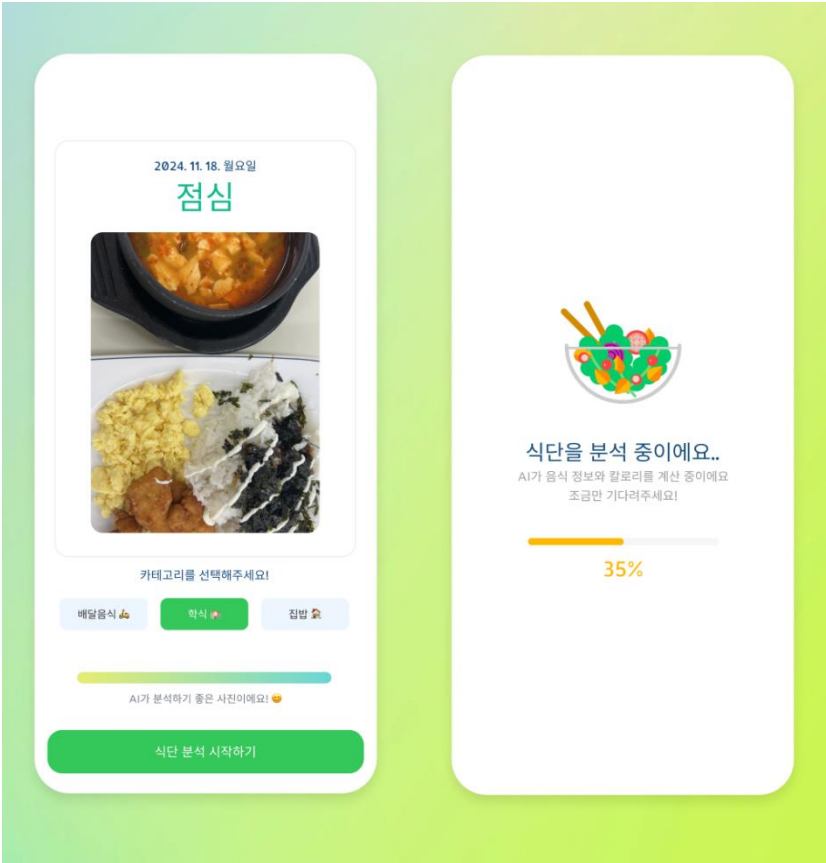


식단 분석 진행 중 화면



2

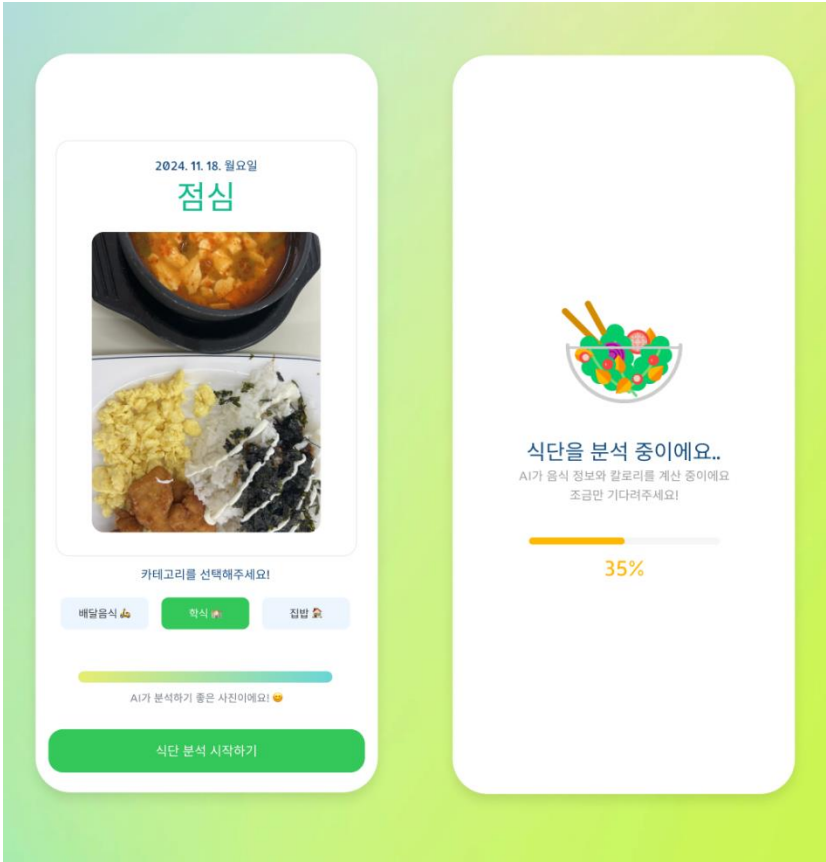
사용자 정보 입력 및 데이터 수집 화면



설명	사용자가 직접 음식 사진을 업로드하고 AI가 이를 분석하여 칼로리와 영양 정보를 추출하는 과정
구성 기능	<div>1. 음식 사진 업로드<ul style="list-style-type: none"><li>• 사용자가 특정 시간(점심, 간식 등)을 선택하고 음식 사진을 업로드</li><li>• 카테고리(배달 음식, 한식, 집밥 등)를 지정해 더 세부적인 데이터 기록 가능</li></ul></div> <div>2. AI 분석 시작<ul style="list-style-type: none"><li>• “식단 분석 시작하기” 버튼 클릭 시, 업로드된 사진이 서버로 전송되어 분석 시작</li><li>• AI는 음식의 칼로리와 주요 영양 성분 추출</li></ul></div> <div>3. 진행 상황 시각화<ul style="list-style-type: none"><li>• 진행 바를 통해 분석 진행 상황을 표시, 사용자 기대감을 유지.</li><li>• 진행 완료 시 결과 화면으로 이동</li></ul></div>

2

# 사용자 정보 입력 및 데이터 수집 화면



## 주요 기능

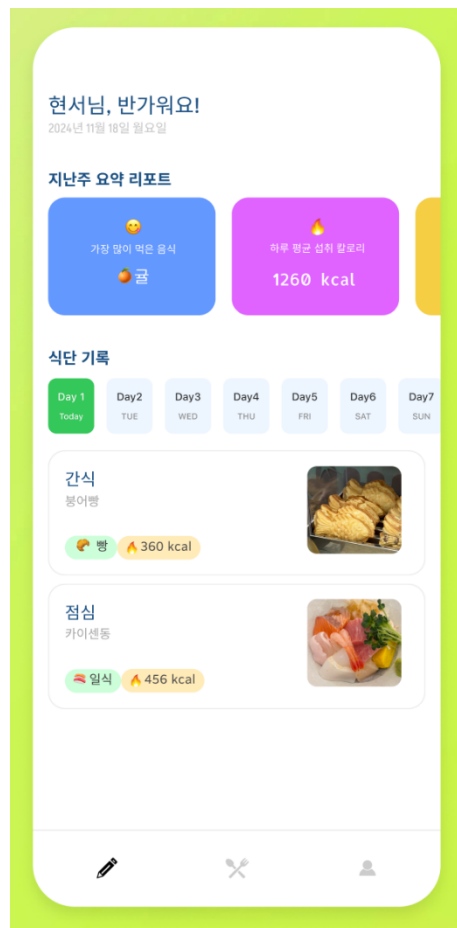
- 음식 사진 업로드  
사용자가 직접 사진을 업로드하여 기록을 추가
- 카테고리 지정  
선택한 시간과 카테고리 기반으로 데이터를 구체적으로 저장
- AI 분석  
업로드 된 데이터를 바탕으로 AI가 자동 분석을 진행.

## 흐름

1. 사용자: 점심 시간 선택 → 음식 사진 업로드
2. 사용자: 카테고리 지정 → “식단 분석 시작하기” 버튼
3. 시스템: AI 분석 진행(사진 → 영양 데이터 추출)
4. 사용자: 완료된 결과 화면으로 이동

3

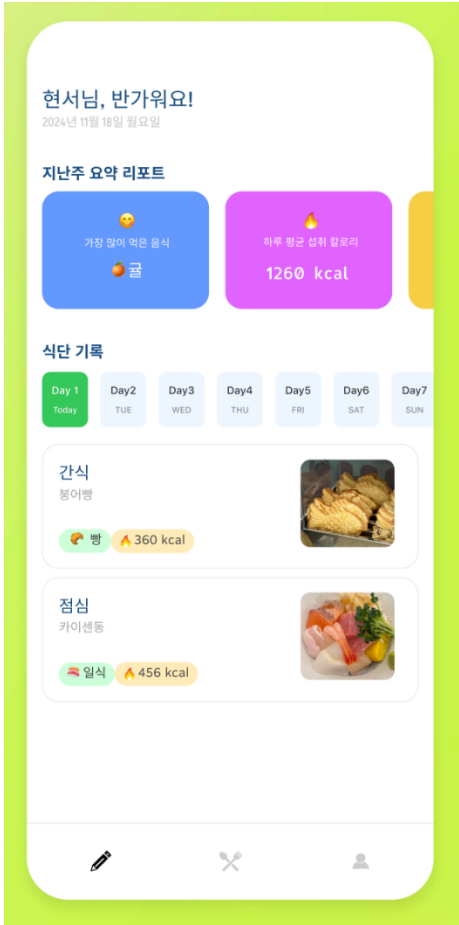
## 사용자 통계 및 기록 관리 화면



식단 기록 및 통계 화면

3

사용자통계및기록관리화면



설명

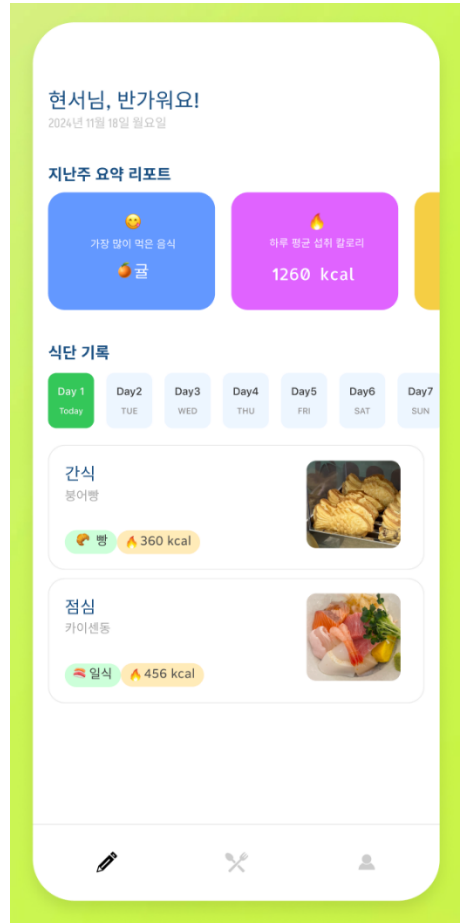
사용자가 하루 동안 기록한 식단 데이터를 확인하고, 요약 통계를 통해 식단을 효율적 관리할 수 있는 메인 화면

구성 기능

1. 요약 리포트 제공
  - 사용자의 주간 리포트를 상단에 시각화 해 제공 (가장 많이 섭취한 음식, 하루 평균 칼로리 등).
  - 데이터는 서버에서 받아와 분석 결과를 표시
2. 식단 기록 확인
  - 간식, 점심 등 사용자가 기록한 식단 데이터를 시간대별로 표시
  - 각 음식별 칼로리 정보와 세부 내용을 확인 가능
3. 날짜별 기록 관리
  - 날짜별로 데이터를 추가, 수정, 삭제할 수 있는 기능 제공
  - 사용자 스크롤이나 선택으로 주간 데이터를 탐색 가능

3

## 사용자통계및기록관리화면



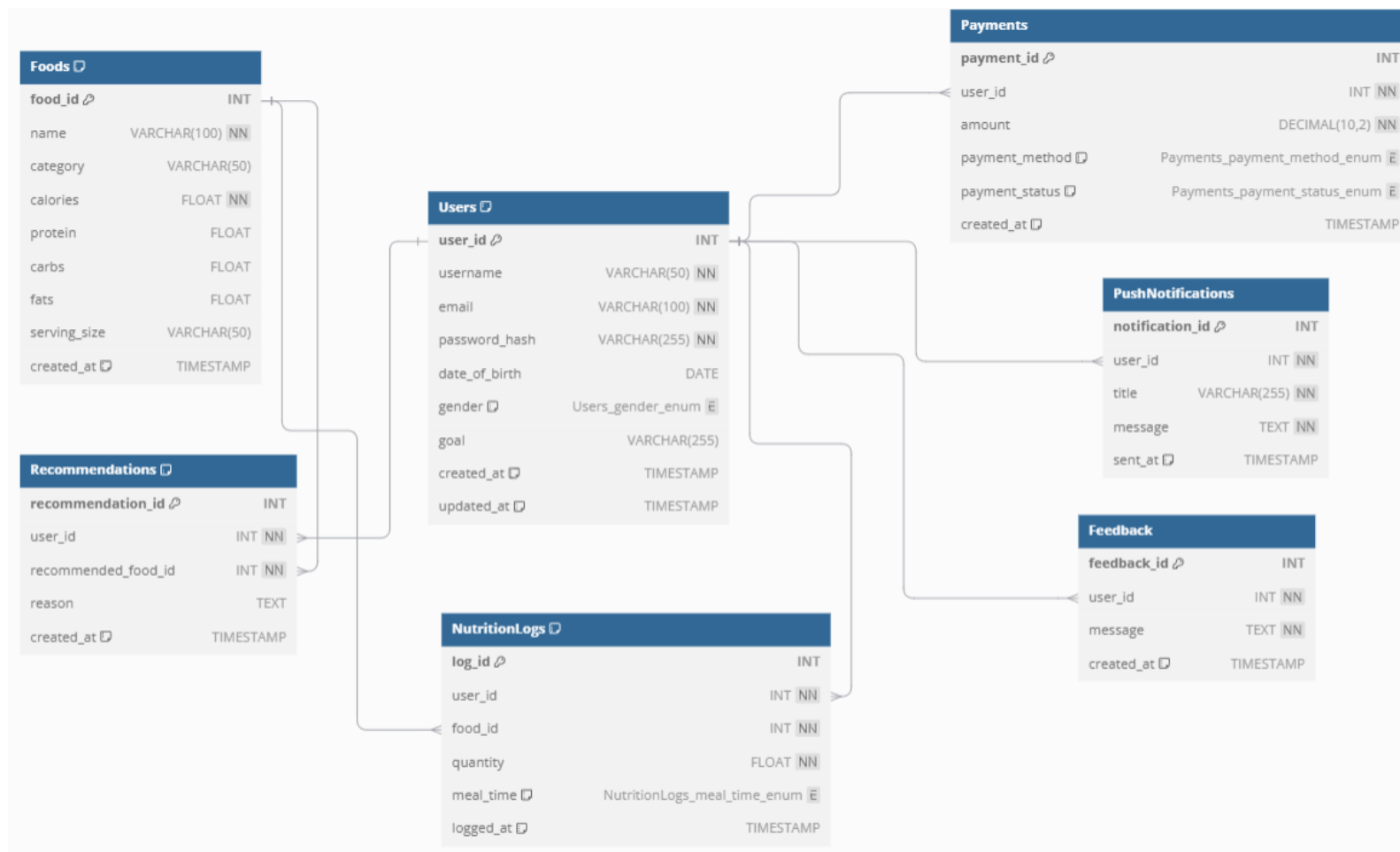
### 주요 기능

- 통계 데이터 제공  
지난주 섭취 데이터 및 칼로리 통계를 요약하여 표시
- 식단 기록 관리  
사용자가 기록한 데이터를 시간대별로 정리하고, 수정

### 흐름

1. 사용자: 화면 상단에서 요약 통계 확인
2. 사용자: 날짜 선택 → 기록된 식단 상세 확인
3. 사용자: 데이터를 추가/수정/삭제

# DB 스키마



# 감사합니다

## 4조

팀장 | 홍성호

팀원 | 이동제, 윤현서, 최현빈, 홍창희