<Food Supply Inventory Management and Recipe Recommendation System>

Software Requirements Specification

For <식재료 재고관리와 재고를 사용하는 레시피 제공>

Version <1.7.4>

2025.06.05

[Note 식재료 재고 관리 및 재고를 사용하는 레시피 추천 시스템을 구축하는 요구사항 정의서 입니다.]

[식재료를 구매 후 식용품을 직접 시스템에 입력을 하거나 영수증을 촬영하여 이미지 센서를 통해 등록을 한다. 냉장고에 있는 식재료를 사용하는 레시피나 1~2개의 주재료를 구매하면 가능한 레시피를 추천해준다. 사용자들이 자신만의 레시피를 커뮤니티를 이용하여 공유할 수 있게 하여 사용자들의 레시피 제공을 촉진 시킨다]

작성자: 팀장 최민준

팀원 김영규, 성하윤, 한규태

계정 이력

|  |  |  |  |
| --- | --- | --- | --- |
| **날짜** | **버전** | **설명** | **작성자** |
| **<2025/03/27>초안** | **<1.0>** | **<식용품 재고관리 및 레시피 제공>** | **<김영규>** |
| **<2025/04/08> 수정** | **<1.0.1>** | **목차 및 내용 정리** | **<최민준>** |
| **<2025/04/10> 수정** | **<1.0.2>** | **목차 및 내용 정리** | **<한규태>** |
| **<2025/04/28> 수정** | **<1.1>** | **기능적, 비기능적 요구사항 업데이트** | **<성하윤>** |
| **<2025/05/01> 수정** | **<1.2>** | **유스케이스 조사 업데이트** | **<김영규>** |
| **<2025/05/08> 수정** | **<1.2.1>** | **목차 및 내용 정리** | **<한규태>** |
| **<2025/05/10> 수정** | **<1.3>** | **간트차트 업데이트 및 내용 수정** | **<최민준>** |
| **<2025/05/17> 수정** | **<1.4>** | **클래스 다이어그램 업데이트** | **<한규태>** |
| **<2025/05/18> 수정** | **<1.5>** | **시퀀스 다이어그램 업데이트** | **<김영규>** |
| **<2025/05/22> 수정** | **<1.6>** | **UI 설계 업데이트** | **<성하윤>** |
| **<2025/05/29> 수정** | **<1.7>** | **시스템 구성도 업데이트** | **<최민준>** |
| **<2025/05/29> 수정** | **<1.7.1>** | **목차 및 내용 정리** | **<최민준>** |
| **<2025/05/30> 수정** | **<1.7.2>** | **오타 수정** | **<최민준>** |
| **<2025/06/03> 수정** | **<1.7.3>** | **최종 이미지 정리** | **<한규태>** |
| **<2025/06/05> 수정** | **<1.7.4>** | **최종 내용 정리** | **<최민준>** |

목차

1. 개요 5

1.1 목적 5

1.2 범위 5

1.3 정의 및 약어 5

1.4 참고 자료 5

1.5 문서 개요 5

2. 시스템 설명 7

2.1 유스케이스 모델 조사 7

2.2 가정 및 의존성 7

2.3 시스템 구성도 8

2.4 개발 일정(간트차트) 9

3. 상세 요구 10

3.1 기능적 요구사항 10

3.2 비기능적 요구사항 11

3.3 기타 요구사항 / 보충 요구사항 11

4. 유스케이스 다이어그램 12

4.1 전체 유스케이스 12

4.2 식재료 관리 유스케이스 13

4.3 레시피 추천 유스케이스 20

4.4 장바구니 유스케이스 26

4.5 시스템 설정 유스케이스 32

4.6 커뮤니티 유스케이스 39

5. 클래스 다이어그램 51

5.1 전체 클래스 51

5.2 클래스 설명 51

6. 시퀀스 다이어그램 52

6.1 레시피 추천 기능 52

6.2 영수증 입력 기능 53

6.3 커뮤니티 기능 54

6.4 장바구니 기능 55

6.5 회원가입/로그인 기능 56

7. UI 설계 57

7.1 회원가입/로그인 화면 57

7.2 메인화면 59

7.3 식재료 화면 61

7.4 영수증 화면 63

7.5 장바구니 화면 65

7.6 레시피 화면 67

7.7 커뮤니티 화면 69

8. 지원 정보 71

9. 참고 문헌 72

요구 명세서

# 개요

“식용품 재고관리와 재고를 사용하는 제공”앱의 개발 목적과 적용 범위, 주요 용어의 정의, 관련 참고자료를 포함한 전반적인 개요를 다룬다. 이 앱은 사용자가 소지한 식재료를 등록하고, 유통기한을 관리하며, 해당 재고를 바탕으로 레시피를 추천해줌으로써 식자재 낭비를 줄이고 효율적인 요리 생활을 지원하는 것을 목표로 한다.

## 목적

이 문서는 “나만의 냉장고” 앱 개발을 위한 시스템 요구사항 명세서로서, 사용자의 식재료 관리와 이를 기반으로 한 레시피 추천 기능을 통해 식재료 낭비를 줄이고 요리 생활의 편의성을 향상시키는 것을 목적으로 한다.

## 범위

본 시스템은 사용자에게 다음과 같은 기능을 제공한다:

• 식재료 입력, 수정, 삭제 및 유통기한 알림

• 영수증 이미지 기반 자동 식재료 등록

• 재고 기반 및 추가 재료 기반 레시피 추천

• 커뮤니티 기능 (레시피 공유, 댓글, 신고 등)

• 관리자 기능 (신고 처리, 사용자 관리 등)

## 정의 및 약어

• 식재료: 사용자가 보유한 식품 혹은 재료

• 유통기한: 재료의 소비 가능 기간

• 레시피: 입력된 재료를 기반으로 제공되는 요리 정보

• 알림: 유통기한이 임박했을 때 사용자에게 알려주는 기능

• AI: 인공지능 시스템, 레시피 추천 및 OCR 인식에 사용

• OCR: Optical Character Recognition, 영수증 문자 인식 기술

• API: Application Programming Interface, 응용 프로그램 인터페이스

• DB: Database, 데이터베이스

• FCM: Firebase Cloud Messaging, 파이어베이스 제공 교차 플랫폼 푸시 알림 서비스

## 참고 자료

• 공공데이터 포털 – 식품 영양정보 및 유통기한 정보 API

• Spoonacular / Edamam 레시피 API

• Kakao Delveloper 문서 (로그인 연동)

• Firebase FCM 공식 문서

## 문서 개요

본 문서는 총 9장으로 구성되며, 시스템 설명부터 UI 설계까지 개발 과정에서 필요한 모든 기술 사양과 흐름을 포함한다. 특히 유스케이스 및 시퀀스 다이어그램, 클래스 구조, UI 흐름 등을 포함해 개발자와 이해관계자 간의 의사소통을 원활하게 한다.

• 1장은 앱의 개발 목적, 범위, 정의된 용어, 참고 자료 등 전체 개요를 다룬다.

• 2장은 사용자가 어떻게 사용하는지를 나타내는 유스케이스 흐름과 시스템의 의존성 및 전제조건을 설명하고 이제 따른 시스템 구성도와 개발 일정을 설명한다.

• 3장은 시스템의 기능적/비기능적 요구사항을 기술하고, 추가적인 보완 사항까지 포함하여 구체화한다.

• 4장은 사용자와 시스템 간의 주요 상호작용을 유스케이스 형태로 도식화하고, 각 기능별 유스케이스 흐름을 상세히 기술한다.

• 5장은 시스템의 객체 및 클래스 구조를 도식화하고, 각 클래스에 대한 설명을 제공한다.

• 6장은 기능 실행 흐름을 시간 순서대로 시각화하여 시스템 동작 절차를 이해하기 쉽게 표현한다.

• 7장은 사용자 인터페이스의 화면 흐름을 중심으로 각 기능별 주요 UI 구조를 설명한다.

• 8장은 시스템의 개발 환경, 사용된 언어, 프레임워크 및 도구 등의 정보를 명시한다.

• 9장은 문서 작성 및 시스템 구현에 참고한 기술 자료, 논문, 온라인 문서 등을 수록한다.

# 시스템 설명

본 시스템은 사용자의 스마트폰 앱(React Native)과 서버(Node.js) 간의 RESTful API 통신을 기반으로 하며, AI API 및 OCR API 등 외부 인공지능 기능을 서버에서 호출하여 레시피 추천 및 영수증 식재료 분석을 제공한다. 또한 Firebase FCM을 통해 유통기한 알림 기능을 제공하고 사용자는 커뮤니티 공간을 통해 다른 사용자와 레시피를 공유할 수 있다. 이러한 기능의 구성도와 개발 일정, 시스템이 정상적으로 작동하기 위한 전제 조건과 외부 의존성을 기술한다.

## 유스케이스 모델 조사

1. **사용자 로그인**

• 사용자는 카카오 로그인으로 앱에 접속할 수 있다.

1. **식재료 등록 및 관리**

• 사용자는 식재료를 직접 입력하거나, 영수증 이미지를 통해 자동 등록할 수 있다.

• 앱은 입력된 식재료 목록을 저장하고 표시하며, 수량 및 정보를 수정할 수 있다.

1. **유통기한 알림**

• 앱은 등록된 식재료의 유통기한을 추적하고, 임박 시 Firebase 푸시 알림을 발송한다.

1. **장바구니 관리 및 공유**

• 사용자는 장바구니에 필요한 재료를 추가할 수 있으며, 해당 목록을 확인하거나 공유할 수 있다.

1. **레시피 추천**

• 앱은 현재 보유 중인 식재료를 기반으로 외부 AI API를 호출해 요리를 추천한다.

1. **커뮤니티 기능**

• 사용자는 커뮤니티에서 레시피 게시글을 열람하고, 공감하거나 댓글을 달고, 부적절한 글을 신고할 수 있다.

1. **관리자 기능**

• 관리자는 신고된 게시글 및 댓글을 확인하고, 삭제 또는 차단 등의 조치를 취할 수 있다.

## 가정 및 의존성

• 사용자가 스마트폰을 통해 앱에 접근

• 인터넷 연결이 되어 있어야 외부 레시피 API 접근 가능

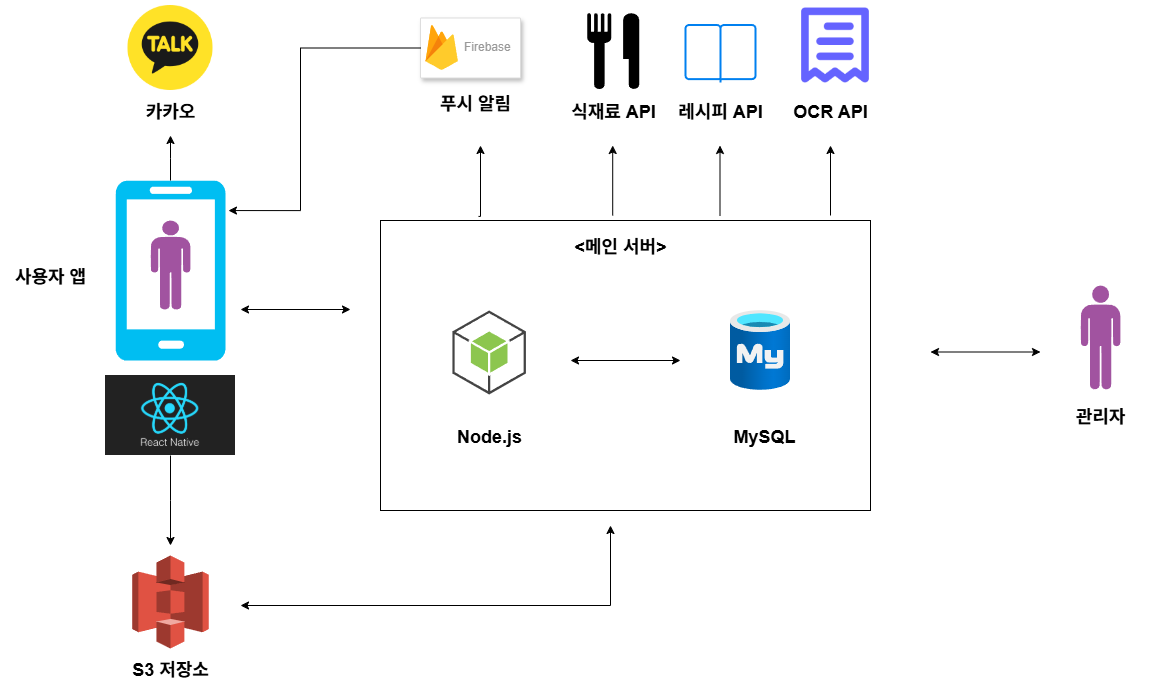
• AI 기반의 레시피 추천 및 OCR 기능은 외부 API를 통해 제공되며, 해당 API가 동작하지 않을 경우 기능 제한이 발생할 수 있다.

• AWS S3를 통해 이미지 저장이 이루어지며, 사용자의 기기 또는 네트워크 환경에 따라 업로드 성능에 차이가 있을 수 있다.

• 관리자 웹 페이지는 로컬 서버 환경에서 백엔드와 동일한 API를 공유한다.

## 시스템 구성도

시스템은 프론트엔드 앱(React Native), 백엔드 서버(Node.js), 관계형 DB(MySQL), 외부 API, FCM, S3 이미지 저장소 등으로 구성된다. 구성도는 이러한 모듈들의 상호작용과 데이터 흐름을 시각적으로 설명한다.



## 개발 일정 (간트 차트)

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **대분류** | **작업 내용** | **2025. 03** | | | | **2025. 04** | | | | **2025. 05** | | | | **2025. 06** | | | |
| 1주 | 2주 | 3주 | 4주 | 1주 | 2주 | 3주 | 4주 | 1주 | 2주 | 3주 | 4주 | 1주 | 2주 | 3주 | 4주 |
| **계획** | **팀 구성 및 역할 분담** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **주제선정** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **목표설정** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **분석** | **문제분석** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **요구사항 수집, 파악** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **유스케이스 분석** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **설계** | **요구사항 명세** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **유스케이스 설계** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **다이어그램 설계** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **UI 설계** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# 상세 요구사항

앱의 기능적 요구사항 비기능적 요구사항을 설명한다. 유스케이스를 기반으로 각 기능의 흐름을 명확히 정의하고 성능이나 보안과 같은 요구사항도 포함한다.

## 기능적 요구사항(Functional Requirements)

1)**회원 관리 기능**• 사용자는 카카오 로그인을 통해 시스템에 접근할 수 있어야 한다.

•사용자는 닉네임 및 알림 설정을 포함한 프로필 정보를 수정할 수 있어야 한다.  
  
• 사용자는 회원 탈퇴를 요청할 수 있어야 한다.

2)**식재료 관리 기능**• 사용자는 식재료를 이름, 수량, 유통기한, 카테고리 정보를 포함하여 직접 입력할 수 있어야 한다.  
  
• 사용자는 식재료 목록에서 특정 항목을 수정 또는 삭제할 수 있어야 한다.  
  
• 시스템은 유통기한 임박 시 해당 식재료에 대해 푸시 알림을 제공해야 한다.  
  
• 사용자는 OCR 영수증 이미지를 통해 자동으로 식재료를 등록할 수 있어야 한다.

3) **레시피 기능**• 사용자는 보유 식재료 기반으로 AI 추천 레시피 목록을 받아볼 수 있어야 한다.  
  
• 사용자는 원하는 레시피를 저장하거나 즐겨찾기에 등록할 수 있어야 한다.  
  
• 레시피는 제목, 내용, 필요한 재료 목록, 작성자로 구성되어야 한다.

4) **장바구니 기능**• 사용자는 식재료를 장바구니에 추가하거나 삭제할 수 있어야 한다.

• 장바구니에 추가된 재료는 구매 여부 상태로 구분되어야 한다.  
  
• 레시피에서 필요한 재료가 부족할 때 부족한 재료를 장바구니에 넣을 수 있어야 한다.  
  
5) **커뮤니티 기능**• 사용자는 레시피를 커뮤니티 게시글로 작성할 수 있어야 한다.  
  
• 사용자는 커뮤니티 게시글에 댓글을 달 수 있어야 한다.  
  
• 사용자는 부적절한 게시글이나 댓글을 신고할 수 있어야 한다.

6) **관리자 기능**• 관리자는 신고된 게시글 및 댓글을 확인할 수 있어야 한다.  
  
• 관리자는 신고 처리 여부를 변경하고, 필요 시 해당 게시글을 삭제할 수 있어야 한다.

## 비기능적 요구사항(Non-Functional Requirements)

• 응답 시간  
레시피 추천 결과는 3초 이내에 사용자에게 표시되어야 한다.  
  
• 플랫폼 호환성  
시스템은 iOS와 Android 양쪽에서 동작해야 한다.  
  
•보안  
사용자 정보(아이디, 비밀번호)는 암호화되어 저장되어야 한다.  
  
• 데이터 정합성  
유통기한, 수량 등 입력된 재료 데이터는 유효성 검사를 거쳐야 한다.  
  
•시스템 확장성  
추가 기능(API, 이미지 인식 등)이 추가되더라도 구조적으로 연동 가능해야 한다.  
  
• 접근성  
시각장애인을 위한 색상 대비, 음성 안내 등의 기능을 고려해야 한다.  
  
• 유지보수성  
설계 문서와 코드에 주석과 명세를 포함하여 쉽게 유지 보수할 수 있어야 한다.

• 서버 처리량  
시스템에서 동시 접속 사용자가 100명 이상이더라도 처리가 가능해야 한다.

• 데이터 암호화  
JWT 토큰 기반 인증 및 SSL 암호화 적용을 하여 암호화 시켜야 한다..

## 기타 요구사항 / 보충 요구사항

• 성능: 레시피 추천은 3초 이내에 응답 가능해야 한다.

• 보안: 사용자 데이터는 로컬 또는 암호화 시켜야 한다.

• 호환성: Android/iOS 모두 지원한다.

• 확장 가능성: 향후 사용자별 냉장고 다중 관리 기능, 재료 자동 분류 기능 등 추가 기능을 모듈 형태로 통합할 수 있어야 한다.

• 반응성 및 UX: 사용자의 행동(터치, 스와이프)에 대해 0.2초 이내의 반응성을 보장한다.

• 이식성: 안드로이드, ios, 태블릿 환경에서도 UI 및 기능에 큰 차이 없이 사용할 수 있어야 한다.

• 장애 대응: 외부 API 장애 발생 시 예외 처리를 통해 사용자에게 실패 메시지를 제공하고 기본 기능(수동 등록)은 유지되도록 한다.

# 유스케이스 다이어그램

## 전체 유스케이스



## 식재료 관리 유스케이스



## 4.2.1 재료 등록

|  |  |  |
| --- | --- | --- |
| **Use Case #001** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 사용자가 직접 보유 식재료를 입력하여 등록한다. (검색하면 식재료api 연결) |
| 작성자 | | 한규태 |
| 액터 | | 사용자, 식재료 API |
| 사전 조건 | | 사용자는 애플리케이션에 로그인된 상태여야 한다. |
| 사후 조건 | | 재료가 냉장고 목록에 성공적으로 등록된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 재료 등록 버튼을 클릭한다. | |
| 2 | 시스템이 재료 정보 입력 폼을 표시 한다. | |
| 3 | 사용자가 재료 명, 수량, 유통기한을 입력한다. | |
| 4 | 시스템이 입력 값의 유효성을 검증한다. | |
| 5 | 시스템이 재료 정보를 데이터베이스에 저장한다. | |
| 6 | 사용자에게 ‘등록 완료’ 메시지를 출력한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 3-a | 재료 명을 입력하지 않은 경우 -> 시스템은 ‘재료 명을 입력하세요’ 라는 메시지를 띄우고 다시 입력을 요청한다. | |
| 3-b | 유통기한 날짜가 지난 경우 -> 시스템은 ‘유통기한이 유효하지 않습니다’라는 메시지를 출력한다 | |
| 4-a | 입력 값 중 수량이 음수인 경우 -> 시스템은 ‘수량은 0 이상의 숫자여야 합니다’라고 경고한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 5-e | 데이터베이스 오류 발생 시 -> 시스템은 ‘등록에 실패했습니다’라는 메시지를 출력한다. | |
| 6-e | 네트워크 오류로 등록 결과를 수신하지 못한 경우 -> 시스템은 ‘서버와의 연결이 원활하지 않습니다’ 라고 출력한다. | |

## 4.2.2 재료 삭제

|  |  |  |
| --- | --- | --- |
| **Use Case #002** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 더 이상 필요하지 않은 식재료를 삭제한다. |
| 작성자 | | 한규태 |
| 액터 | | 사용자 |
| 사전 조건 | | 삭제할 재료가 등록되어 있어야 한다. |
| 사후 조건 | | 재료가 냉장고 목록에서 제거된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 삭제할 재료를 선택한다. | |
| 2 | 시스템이 삭제 확인 메시지를 표시한다. | |
| 3 | 사용자가 삭제를 확인한다. | |
| 4 | 시스템이 해당 자료를 데이터베이스에서 삭제한다. | |
| 5 | 사용자에게 ‘삭제 완료’ 메시지를 출력한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 3-a | 사용자가 삭제를 취소한 경우 -> 시스템은 재료 목록 화면으로 복귀한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 4-e | 삭제 중 데이터베이스 오류 발생 시 -> 시스템은 ‘삭제 실패’ 메시지를 출력한다. | |

## 4.2.3 보유 재료 검색

|  |  |  |
| --- | --- | --- |
| **Use Case #003** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 등록한 재료 내에서 재료를 검색한다. |
| 작성자 | | 한규태 |
| 액터 | | 사용자 |
| 사전 조건 | | 최소 1개 이상의 재료가 등록되어 있어야 한다. |
| 사후 조건 | | 검색 결과에 따라 재료 목록이 출력된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 검색 창에 재료명을 입력한다. | |
| 2 | 시스템이 입력된 키워드로 재료 목록을 필터링한다. | |
| 3 | 검색 결과를 사용자에게 표시한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 1-a | 입력한 재료 명이 존재하지 않을 경우 -> 시스템은 ‘검색 결과가 없습니다’라고 안내한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 2-e | 시스템 오유로 검색이 실패한 경우 -> 시스템은 ‘검색을 수행할 수 없습니다’라고 메시지를 출력한다. | |

## 4.2.4 수량 설정

|  |  |  |
| --- | --- | --- |
| **Use Case #004** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 식재료의 보유 수량을 설정하거나 수정한다. |
| 작성자 | | 한규태 |
| 액터 | | 사용자 |
| 사전 조건 | | 등록된 재료가 존재해야 한다. |
| 사후 조건 | | 입력한 수량으로 저장된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 수량을 입력하거나 수정한다. | |
| 2 | 시스템이 수량이 0 이상인지 확인한다. | |
| 3 | 유효한 경우 수량을 저장한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 2-a | 입력한 수량이 음수인 경우 -> 시스템은 ‘수량은 0 이상이어야 합니다’라고 안내한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 3-e | 데이터베이스 저장 오류 발생 시 -> 시스템은 ‘수량 저장 실패’ 메시지를 출력한다. | |

## 4.2.5 영수증 입력

|  |  |  |
| --- | --- | --- |
| **Use Case #005** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 영수증 이미지를 업로드 하면 AI가 식재료 목록을 추출해 자동 등록한다. |
| 작성자 | | 한규태 |
| 액터 | | 사용자, OCR API |
| 사전 조건 | | 사용자가 영수증 이미지를 촬영하거나 업로드 할 숭 있어야 한다. |
| 사후 조건 | | 영수증 이미지로부터 인식된 재료 명과 수량이 입력 폼에 자동 반영되고, 사용자가 이를 확인할 수 있다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 영수증을 촬영 하거나 업로드한다. | |
| 2 | 시스템(AI)이 영수증 이미지를 분석한다. | |
| 3 | 인식된 재료 명과 수량을 입력 폼에 자동으로 채운다. | |
| 4 | 시스템이 댓글을 데이터베이스에 저장한다. | |
| 5 | 사용자에게 '댓글이 등록되었습니다' 메시지를 출력한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 2-a | 이미지 품질이 낮아 인식이 불가능한 경우 -> 시스템은 ‘이미지를 인식할 수 없습니다’라고 안내한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 2-e | AI 분석 중 오류 발생 시 -> 시스템은 ‘재료 정보를 불러오지 못했습니다’라는 메시지를 출력한다. | |

## 4.2.6 유통기한 설정

|  |  |  |
| --- | --- | --- |
| **Use Case #006** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 식재료별 유통기한을 설정해 관리한다. |
| 작성자 | | 한규태 |
| 액터 | | 사용자 |
| 사전 조건 | | 사용자는 재료 등록 과정 중이어야 한다. |
| 사후 조건 | | 유통기한이 설정되어 저장된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 유통기한을 입력한다. | |
| 2 | 시스템이 날짜 형식 및 유효성(미래 날짜)을 검사한다. | |
| 3 | 검사 통과 시 유통기한 정보를 저장한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 2-a | 입력한 날짜가 과거일 경우 -> 시스템은 ‘유효하지 않은 날짜입니다’라고 안내한다. | |
| 2-b | 형식이 YYYY-MM-DD 가 아닐 경우 -> 시스템은 ‘날짜 형식을 확인해주세요’라고 안내한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 3-e | 유통기한 저장 중 DB 오류 발생 시 -> 시스템은 ‘유통기한 저장 실패’라는 메시지를 출력한다. | |

## 레시피 추천 유스케이스



## 4.3.1 레시피 추천 받기

|  |  |  |
| --- | --- | --- |
| **Use Case #001** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 보유한 재료를 기반으로 AI가 추천한 레시피를 받는다. |
| 작성자 | | 한규태 |
| 액터 | | 사용자, 레시피 API |
| 사전 조건 | | 사용자가 보유 재료 정보를 등록해둬야 한다. |
| 사후 조건 | | AI가 추천한 레시피 목록이 사용자에게 제공된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 ‘레시피 추천’ 기능을 선택한다. | |
| 2 | 시스템이 보유 재료 정보를 AI에게 전달한다. | |
| 3 | AI가 분석 후 추천 레시피를 시스템에 전달한다. | |
| 4 | 시스템이 추천 레시피 목록을 사용자에게 표시한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 2-a | 추천 가능한 레시피가 없는 경우 -> 시스템은 ‘추천할 레시피가 없습니다’라는 메시지를 출력한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 2-e | AI 서버 응답 지연/실패 -> 시스템은 ‘AI 추천을 불러오는데 실패했습니다’라는 메시지를 출력한다. | |

## 4.3.2 사용 가능한 재료 필터링

|  |  |  |
| --- | --- | --- |
| **Use Case #002** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 추천 전에 사용할 수 있는 재료만 골라 필터링한다. |
| 작성자 | | 한규태 |
| 액터 | | 사용자 |
| 사전 조건 | | 사용자 보유 재료 중 일부가 사용 가능 상태로 설정되어야 한다. |
| 사후 조건 | | 사용 가능한 재료만 포함된 필터 조건이 적용된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 필터링 조건으로 ‘사용 가능한 재료만 보기’를 선택한다.. | |
| 2 | 시스템이 유효 재료만 추려내 필터링 한다. | |
| 3 | 필터링 된 재료 목록에 따라 추천 레시피가 갱신된다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 2-a | 모든 재료가 사용 불가능한 상태일 경우 -> 시스템은 ‘사용 가능한 재료가 없습니다’라는 메시지를 출력한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 2-e | 필터 처리 중 오류 발생 ->시스템은 ‘재료 필터링에 실패했습니다’라는 메시지를 출력한다. | |

## 4.3.3 재료 일부로만 레시피 탐색

|  |  |  |
| --- | --- | --- |
| **Use Case #003** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 선택된 일부 재료만으로 만들 수 있는 레시피를 조회한다. |
| 작성자 | | 한규태 |
| 액터 | | 사용자, 레시피API |
| 사전 조건 | | 사용자 보유 재료 중 일부를 선택할 수 있어야 한다. |
| 사후 조건 | | 선택된 재료만으로 만들 수 있는 레시피가 표시된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 사용하고자 하는 재료를 선택한다. | |
| 2 | 시스템이 선택된 재료를 기준으로 가능한 레시피를 탐색한다. | |
| 3 | 해당 조건을 만족하는 레시피 목록을 사용자에게 제공한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 2-a | 선택된 재료로 만들 수 있는 레시피가 없는 경우 -> 시스템은 ‘조건을 만족하는 레시피가 없습니다’라고 출력한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 2-e | 레시피 검색 실패 -> 시스템은 ‘레시피를 탐색하는 데 실패했습니다’라는 메시지를 출력한다. | |

## 4.3.4 추천 레시피 저장

|  |  |  |
| --- | --- | --- |
| **Use Case #004** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 추천받은 레시피를 저장해 나중에 볼 수 있다. |
| 작성자 | | 한규태 |
| 액터 | | 사용자 |
| 사전 조건 | | 추천 레시피가 존재해야 한다. |
| 사후 조건 | | 선택한 레시피가 사용자 저장소에 저장된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 저장하고 싶은 레시피를 선택한다. | |
| 2 | 시스템이 해당 레시피를 사용자 계정에 저장한다. | |
| 3 | 시스템이 ‘저장 완료’라는 메시지를 출력한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 1-a | 사용자가 이미 저장한 레시피를 다시 저장하려는 경우 -> 시스템은 ‘이미 저장된 레시피입니다.’라는 메시지를 출력한다. | |
| 1-b | 사용자가 로그인되지 않은 상태에서 레시피를 저장하려는 경우 -> 시스템은 ‘로그인이 필요합니다’라는 메시지를 출력하고, 로그인 화면으로 이동한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 2-e | 저장 공간 부족 또는 권한 오류 발생 시 -> 시스템은 ‘레시피 저장에 실패했습니다’라는 메시지를 출력한다. | |
| 2-e | 네트워크 오류 등으로 저장 요청이 서버에 도달하지 못한 경우 -> 시스템은 ‘서버 연결이 원활하지 않습니다. 다시 시도해주세요.’라는 메시지를 출력한다. | |

## 4.3.5 추천 레시피 공유

|  |  |  |
| --- | --- | --- |
| **Use Case #005** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 추천된 레시피를 커뮤니티 또는 외부로 공유한다. |
| 작성자 | | 한규태 |
| 액터 | | 사용자 |
| 사전 조건 | | 공유하려는 레시피가 추천 결과로 존재해야 한다. |
| 사후 조건 | | 레시피가 커뮤니티 또는 외부 플랫폼에 공유된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 공유 기능을 선택한다. | |
| 2 | 시스템이 공유 옵션(링크 복사, 외부 앱 열기 등)을 사용자에게 제공한다. | |
| 3 | 사용자가 원하는 공유 방식으로 공유를 진행한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 1-a | 사용자가 공유를 취소한 경우 -> 시스템은 공유 프로세스를 중단하고 이전 화면으로 복귀한다. | |
| 2-a | 사용자가 선택한 공유 플랫폼이 현재 지원되지 않는 경우 -> 시스템은 ‘해당 플랫폼은 현재 지원되지 않습니다’라는 메시지를 출력하고, 다른 옵션을 안내한다. | |
| 2-b | 사용자가 공유 옵션 선택 후 아무 작업도 하지 않고 앱을 닫은 경우 -> 시스템은 공유가 완료되지 않았음을 감지하고 상태를 유지한 채 이전 화면으로 돌아간다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 3-e | 네트워크 오류로 인해 공유가 실패한 경우 -> 시스템은 ‘레시피 공유에 실패했습니다. 인터넷 연결을 확인해주세요.’라는 메시지를 출력한다. | |
| 3-e | 외부 앱 연동 실패 또는 시스템 권한 거부 발생 시 -> 시스템은 ‘외부 앱 실행에 실패했습니다. 권한을 확인해주세요.’라는 메시지를 출력한다. | |

## 장바구니 유스케이스



## 4.4.1 장바구니 재료 추가

|  |  |  |
| --- | --- | --- |
| **Use Case #001** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 장바구니에 필요한 재료를 추가한다 |
| 작성자 | | 김영규 |
| 액터 | | 사용자, 식재료 API |
| 사전 조건 | | 사용자가 장바구니 기능에 접근할 수 있어야 한다. |
| 사후 조건 | | 장바구니에 사용자가 선택한 재료가 추가된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 로그인을 하고 장바구니 관리 아이콘을 클릭한다 | |
| 2 | 추가할 재료를 입력한 다음 ‘+’ 아이콘을 클릭하면 장바구니 리스트에 저장이 된다 | |
| 3 | 재료를 장바구니에 추가하고, 결과를 사용자에게 알린다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 2-a | 추가 가능한 재료가 없는 경우 → 시스템은 ‘추가할 수 있는 재료가 없습니다’라는 메시지를 출력한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 3-e | 시스템 오류 또는 네트워크 장애 → 시스템은 ‘재료 추가에 실패했습니다’라는 메시지를 출력한다. | |

## 4.4.2 장바구니 재료 검색

|  |  |  |
| --- | --- | --- |
| **Use Case #002** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 장바구니에 필요한 재료를 검색한다 |
| 작성자 | | 김영규 |
| 액터 | | 사용자 |
| 사전 조건 | | 사용자가 장바구니에 재료를 추가해두어야 한다. |
| 사후 조건 | | 검색 결과가 사용자에게 제공된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 ‘장바구니 재료 검색’ 기능을 선택한다. | |
| 2 | 시스템이 검색창을 제공한다. | |
| 3 | 사용자가 검색어를 입력한다. | |
| 4 | 시스템이 해당 재료를 검색하여 결과를 표시한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 4-a | 검색 결과가 없을 경우 → 시스템은 ‘검색 결과가 없습니다’라는 메시지를 출력한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 4-e | 시스템 오류 발생 → 시스템은 ‘검색에 실패했습니다’라는 메시지를 출력한다. | |

## 4.4.3 장바구니 재료 삭제

|  |  |  |
| --- | --- | --- |
| **Use Case #003** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 장바구니에 필요한 재료를 삭제한다 |
| 작성자 | | 김영규 |
| 액터 | | 사용자 |
| 사전 조건 | | 장바구니에 삭제할 재료가 있어야 한다. |
| 사후 조건 | | 재료가 장바구니에서 삭제된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 ‘장바구니 재료 삭제’ 기능을 선택한다. | |
| 2 | 시스템이 장바구니 목록을 보여준다. | |
| 3 | 사용자가 삭제할 재료를 선택한다. | |
| 4 | 시스템이 해당 재료를 장바구니에서 삭제하고 결과를 표시한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 3-a | 삭제할 재료가 없는 경우 → 시스템은 ‘삭제할 재료가 없습니다’라는 메시지를 출력한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 4-e | 시스템 오류 발생 → 시스템은 ‘재료 삭제에 실패했습니다’라는 메시지를 출력한다. | |

## 4.4.4 장바구니 수량 설정

|  |  |  |
| --- | --- | --- |
| **Use Case #004** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 장바구니의 재료 수량을 설정한다. |
| 작성자 | | 김영규 |
| 액터 | | 사용자 |
| 사전 조건 | | 장바구니에 수량을 설정할 재료가 있어야 한다. |
| 사후 조건 | | 재료의 수량이 변경된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 ‘장바구니 수량 설정’ 기능을 선택한다. | |
| 2 | 시스템이 장바구니 목록을 보여준다. | |
| 3 | 사용자가 수량을 변경할 재료를 선택한다. | |
| 4 | 사용자가 새로운 수량을 입력한다. | |
| 5 | 시스템이 수량을 변경하고 결과를 표시한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 3-a | 수량을 변경할 재료가 없는 경우 → 시스템은 ‘수량을 변경할 재료가 없습니다’라는 메시지를 출력한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 5-e | 시스템 오류 발생 → 시스템은 ‘수량 변경에 실패했습니다’라는 메시지를 출력한다. | |

## 4.4.5 구매 완료 처리

|  |  |  |
| --- | --- | --- |
| **Use Case #005** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 장바구니의 재료 구매를 완료 처리한다. |
| 작성자 | | 김영규 |
| 액터 | | 사용자 |
| 사전 조건 | | 장바구니에 구매할 재료가 있어야 한다. |
| 사후 조건 | | 재료 구매가 완료되고, 장바구니에서 해당 재료가 제거된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 ‘구매 완료 처리’ 기능을 선택한다. | |
| 2 | 시스템이 장바구니 목록을 보여준다. | |
| 3 | 사용자가 구매할 재료를 선택한다. | |
| 4 | 시스템이 구매를 처리하고 결과를 표시한다. | |
| 5 | 구매된 재료를 장바구니에서 제거한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 3-a | 구매할 재료가 없는 경우 → 시스템은 ‘구매할 재료가 없습니다’라는 메시지를 출력한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 4-e | 시스템 오류 발생 → 시스템은 ‘구매 완료 처리에 실패했습니다’라는 메시지를 출력한다. | |

## 시스템 설정 유스케이스



## 4.5.1 회원가입

|  |  |  |
| --- | --- | --- |
| **Use Case #001** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 사용자가 회원가입을 통해 새로운 계정을 생성한다. |
| 작성자 | | 김영규 |
| 액터 | | 사용자 |
| 사전 조건 | | 회원가입을 위한 정보(이메일, 비밀번호 등)를 준비해야 한다. |
| 사후 조건 | | 계정이 생성되고, 사용자는 서비스 이용이 가능하다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 ‘회원가입’ 기능을 선택한다. | |
| 2 | 시스템이 회원가입 양식을 제공한다. | |
| 3 | 사용자가 정보를 입력하고 제출한다. | |
| 4 | 시스템이 계정을 생성하고 가입 완료 메시지를 표시한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 3-a | 필수 입력 정보가 누락된 경우 → 시스템은 ‘필수 정보를 입력하세요’라는 메시지를 출력한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 4-e | 시스템 오류 발생 → 시스템은 ‘회원가입에 실패했습니다’라는 메시지를 출력한다. | |

## 4.5.2 로그인

|  |  |  |
| --- | --- | --- |
| **Use Case #002** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 사용자가 계정으로 로그인한다. |
| 작성자 | | 김영규 |
| 액터 | | 사용자 |
| 사전 조건 | | 회원가입이 완료된 계정 정보가 있어야 한다. |
| 사후 조건 | | 사용자가 서비스에 로그인 된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 ‘로그인’ 기능을 선택한다. | |
| 2 | 시스템이 로그인 양식을 제공한다. | |
| 3 | 사용자가 계정 정보를 입력하고 제출한다. | |
| 4 | 시스템이 인증 후 로그인 처리 및 환영 메시지를 표시한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 3-a | 입력 정보가 일치하지 않는 경우 → 시스템은 ‘로그인 정보가 올바르지 않습니다’라는 메시지를 출력한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 4-e | 시스템 오류 발생 → 시스템은 ‘로그인에 실패했습니다’라는 메시지를 출력한다. | |

## 4.5.3 로그아웃

|  |  |  |
| --- | --- | --- |
| **Use Case #003** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 사용자가 서비스에서 로그아웃한다. |
| 작성자 | | 김영규 |
| 액터 | | 사용자 |
| 사전 조건 | | 사용자가 로그인 상태여야 한다. |
| 사후 조건 | | 사용자가 로그아웃되고, 서비스 이용이 제한된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 ‘로그아웃’ 기능을 선택한다. | |
| 2 | 시스템이 로그아웃 처리를 수행한다. | |
| 3 | 시스템이 로그아웃 완료 메시지를 표시한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 1-a | 이미 로그아웃된 상태에서 시도 시 → 시스템은 ‘이미 로그아웃된 상태입니다’라는 메시지를 출력한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 2-e | 시스템 오류 발생 → 시스템은 ‘로그아웃에 실패했습니다’라는 메시지를 출력한다. | |

## 4.5.4 회원탈퇴

|  |  |  |
| --- | --- | --- |
| **Use Case #005** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 사용자가 닉네임을 설정하거나 변경한다. |
| 작성자 | | 김영규 |
| 액터 | | 사용자 |
| 사전 조건 | | 사용자가 로그인 상태여야 한다. |
| 사후 조건 | | 닉네임이 설정 또는 변경된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 ‘닉네임 설정’ 기능을 선택한다. | |
| 2 | 시스템이 닉네임 입력란을 제공한다. | |
| 3 | 사용자가 닉네임을 입력하고 저장한다. | |
| 4 | 시스템이 닉네임을 저장하고 완료 메시지를 표시한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 3-a | 사용자가 탈퇴를 취소하는 경우 → 시스템은 ‘회원탈퇴가 취소되었습니다’라는 메시지를 출력한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 4-e | 시스템 오류 발생 → 시스템은 ‘회원탈퇴에 실패했습니다’라는 메시지를 출력한다. | |

## 4.5.5 닉네임 설정

|  |  |  |
| --- | --- | --- |
| **Use Case #005** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 사용자가 닉네임을 설정하거나 변경한다. |
| 작성자 | | 김영규 |
| 액터 | | 사용자 |
| 사전 조건 | | 사용자가 로그인 상태여야 한다. |
| 사후 조건 | | 닉네임이 설정 또는 변경된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 ‘닉네임 설정’ 기능을 선택한다. | |
| 2 | 시스템이 닉네임 입력란을 제공한다. | |
| 3 | 사용자가 닉네임을 입력하고 저장한다. | |
| 4 | 시스템이 닉네임을 저장하고 완료 메시지를 표시한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 3-a | 사용자가 탈퇴를 취소하는 경우 → 시스템은 ‘회원탈퇴가 취소되었습니다’라는 메시지를 출력한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 4-e | 시스템 오류 발생 → 시스템은 ‘회원탈퇴에 실패했습니다’라는 메시지를 출력한다. | |

## 4.5.6 알림 설정

|  |  |  |
| --- | --- | --- |
| **Use Case #006** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 사용자가 알림 수신 여부를 설정한다. |
| 작성자 | | 김영규 |
| 액터 | | 사용자 |
| 사전 조건 | | 사용자가 로그인 상태여야 한다. |
| 사후 조건 | | 알림 설정이 저장된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 ‘알림 설정’ 기능을 선택한다. | |
| 2 | 시스템이 알림 설정 옵션을 제공한다. | |
| 3 | 사용자가 알림 수신 여부를 선택한다. | |
| 4 | 시스템이 설정을 저장하고 완료 메시지를 표시한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 3-a | 설정할 옵션이 없는 경우 → 시스템은 ‘설정 가능한 알림이 없습니다’라는 메시지를 출력한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 4-e | 시스템 오류 발생 → 시스템은 ‘알림 설정에 실패했습니다’라는 메시지를 출력한다. | |

## 커뮤니티 유스케이스



## 4.6.1 레시피 게시글 작성

|  |  |  |
| --- | --- | --- |
| **Use Case #001** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 사용자가 요리법을 입력하여 레시피 게시글을 작성한다. |
| 작성자 | | 성하윤 |
| 액터 | | 사용자 |
| 사전 조건 | | 사용자는 애플리케이션에 로그인된 상태여야 한다. |
| 사후 조건 | | 작성한 게시글이 게시판에 성공적으로 등록된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 '레시피 작성' 버튼을 클릭한다. | |
| 2 | 시스템이 레시피 작성 폼을 표시한다. | |
| 3 | 사용자가 제목, 재료, 조리 방법 등의 내용을 입력한다. | |
| 4 | 시스템이 입력값의 유효성을 검사한다. | |
| 5 | 시스템이 게시글을 데이터베이스에 저장한다. | |
| 6 | 사용자에게 '작성 완료' 메시지를 출력한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 3 | 제목 또는 본문을 입력하지 않은 경우 | |
| 3-a | 시스템은 '제목과 본문을 입력해주세요' 메시지를 출력하고 다시 입력을 요청한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 2 | 시스템 장애로 작성 폼이 로드되지 않는 경우 | |
| 2-e | 시스템은 '작성 화면을 불러오는 데 실패했습니다' 메시지를 출력하고 홈 화면으로 이동한다. | |

## 4.6.2 레시피 게시글 수정

|  |  |  |
| --- | --- | --- |
| **Use Case #002** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 사용자가 작성한 레시피 게시글을 수정한다. |
| 작성자 | | 성하윤 |
| 액터 | | 사용자 |
| 사전 조건 | | 사용자는 애플리케이션에 로그인된 상태여야 한다. |
| 사후 조건 | | 수정된 게시글이 저장되고 갱신된 내용이 표시된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 자신의 게시글에서 '수정' 버튼을 클릭한다. | |
| 2 | 시스템이 기존 내용이 포함된 수정 폼을 표시한다. | |
| 3 | 사용자가 내용을 수정한다. | |
| 4 | 시스템이 수정된 입력값을 검사한다. | |
| 5 | 시스템이 게시글을 업데이트하여 저장한다. | |
| 6 | 사용자에게 '수정 완료' 메시지를 출력한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 3 | 제목 또는 본문을 비워둔 경우 | |
| 3-a | 시스템은 '제목과 본문은 필수 입력 항목입니다' 메시지를 출력하고 입력을 요청한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 2 | 사용자가 작성자가 아닌 경우 | |
| 2-e | 시스템은 '수정 권한이 없습니다' 메시지를 출력하고 접근을 차단한다. | |

## 4.6.3 레시피 게시글 삭제

|  |  |  |
| --- | --- | --- |
| **Use Case #003** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 사용자가 작성한 레시피 게시글을 삭제한다. |
| 작성자 | | 성하윤 |
| 액터 | | 사용자 |
| 사전 조건 | | 사용자는 애플리케이션에 로그인된 상태여야 한다. |
| 사후 조건 | | 게시글이 삭제되고 목록에서 사라진다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 게시글의 '삭제' 버튼을 클릭한다. | |
| 2 | 시스템이 삭제 여부를 묻는 팝업을 표시한다. | |
| 3 | 사용자가 삭제를 확인한다. | |
| 4 | 시스템이 게시글을 데이터베이스에서 제거한다. | |
| 5 | 사용자에게 '삭제 완료' 메시지를 출력한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 3 | 사용자가 삭제를 취소한 경우 | |
| 3-a | 시스템은 게시글을 유지하고 이전 화면으로 돌아간다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 1 | 사용자가 작성자가 아닌 경우 | |
| 1-e | 시스템은 '삭제 권한이 없습니다' 메시지를 출력하고 삭제를 차단한다. | |

## 4.6.4 공감 표시

|  |  |  |
| --- | --- | --- |
| **Use Case #004** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 사용자가 다른 사용자의 게시글에 공감을 표시한다. |
| 작성자 | | 성하윤 |
| 액터 | | 사용자 |
| 사전 조건 | | 사용자는 애플리케이션에 로그인된 상태여야 한다. |
| 사후 조건 | | 공감 수가 증가하고 해당 정보가 저장된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 게시글의 '공감' 버튼을 클릭한다. | |
| 2 | 시스템이 해당 사용자의 공감 여부를 확인한다. | |
| 3 | 시스템이 공감 수를 1 증가시킨다. | |
| 4 | 사용자에게 '공감 완료' 메시지를 출력한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 2 | 사용자가 이미 공감을 표시한 경우 | |
| 2-a | 시스템은 '이미 공감하셨습니다' 메시지를 출력하고 공감을 중복 저장하지 않는다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 1 | 게시글이 존재하지 않는 경우 | |
| 1-e | 시스템은 '해당 게시글을 찾을 수 없습니다' 메시지를 출력한다. | |

## 4.6.5 댓글 작성

|  |  |  |
| --- | --- | --- |
| **Use Case #005** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 사용자가 게시글에 댓글을 작성한다. |
| 작성자 | | 성하윤 |
| 액터 | | 사용자 |
| 사전 조건 | | 사용자는 애플리케이션에 로그인된 상태여야 한다. |
| 사후 조건 | | 댓글이 게시글 하단에 표시된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 게시글 하단의 댓글 입력창에 내용을 작성한다. | |
| 2 | 사용자가 '댓글 등록' 버튼을 클릭한다. | |
| 3 | 시스템이 입력값을 검사한다. | |
| 4 | 시스템이 댓글을 데이터베이스에 저장한다. | |
| 5 | 사용자에게 '댓글이 등록되었습니다' 메시지를 출력한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 3 | 댓글 내용이 비어 있는 경우 | |
| 3-a | 시스템은 '댓글을 입력해주세요' 메시지를 출력하고 다시 입력을 요청한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 1 | 게시글이 삭제되었거나 존재하지 않는 경우 | |
| 1-e | 시스템은 '댓글을 작성할 수 없습니다. 게시글이 존재하지 않습니다' 메시지를 출력한다. | |

## 4.6.6 댓글 수정

|  |  |  |
| --- | --- | --- |
| **Use Case #006** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 사용자가 작성한 댓글을 수정한다. |
| 작성자 | | 성하윤 |
| 액터 | | 사용자 |
| 사전 조건 | | 사용자는 애플리케이션에 로그인된 상태여야 한다. |
| 사후 조건 | | 댓글이 수정되어 반영된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 댓글 옆의 '수정' 버튼을 클릭한다. | |
| 2 | 시스템이 기존 댓글이 포함된 수정 입력창을 표시한다. | |
| 3 | 사용자가 내용을 수정하고 저장 버튼을 클릭한다. | |
| 4 | 시스템이 입력값을 검사한다. | |
| 5 | 시스템이 수정된 댓글을 저장한다. | |
| 6 | 사용자에게 '댓글이 수정되었습니다' 메시지를 출력한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 3 | 댓글을 모두 삭제한 경우 | |
| 3-a | 시스템은 '댓글 내용을 입력해주세요' 메시지를 출력하고 다시 입력을 요청한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 1 | 사용자가 댓글 작성자가 아닌 경우 | |
| 1-e | 시스템은 '수정 권한이 없습니다' 메시지를 출력하고 수정 화면을 차단한다. | |

## 4.6.7 댓글 삭제

|  |  |  |
| --- | --- | --- |
| **Use Case #007** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 사용자가 작성한 댓글을 삭제한다. |
| 작성자 | | 성하윤 |
| 액터 | | 사용자 |
| 사전 조건 | | 사용자는 애플리케이션에 로그인된 상태여야 한다. |
| 사후 조건 | | 댓글이 삭제되어 더 이상 표시되지 않는다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 삭제할 댓글의 '삭제' 버튼을 클릭한다. | |
| 2 | 시스템이 삭제 확인 팝업을 표시한다. | |
| 3 | 사용자가 삭제를 확인한다. | |
| 4 | 시스템이 댓글을 데이터베이스에서 제거한다. | |
| 5 | 사용자에게 '댓글이 삭제되었습니다' 메시지를 출력한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 3 | 사용자가 삭제를 취소하는 경우 | |
| 3-a | 시스템은 아무 작업도 하지 않고 이전 화면으로 복귀한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 1 | 사용자가 댓글 작성자가 아닌 경우 | |
| 1-e | 시스템은 '삭제 권한이 없습니다' 메시지를 출력하고 삭제를 차단한다. | |

## 4.6.8 게시글/댓글 신고

|  |  |  |
| --- | --- | --- |
| **Use Case #008** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 사용자가 부적절한 게시글이나 댓글을 신고한다. |
| 작성자 | | 성하윤 |
| 액터 | | 사용자 |
| 사전 조건 | | 사용자는 애플리케이션에 로그인된 상태여야 한다. |
| 사후 조건 | | 신고 내용이 관리 시스템에 저장된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 게시글 또는 댓글에서 '신고' 버튼을 클릭한다. | |
| 2 | 시스템이 신고 사유를 입력받는 팝업을 표시한다. | |
| 3 | 사용자가 신고 사유를 입력하고 '제출' 버튼을 클릭한다. | |
| 4 | 시스템이 신고 내용을 저장하고 관리자에게 알림을 보낸다. | |
| 5 | 사용자에게 '신고가 접수되었습니다' 메시지를 출력한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 3 | 신고 사유가 입력되지 않은 경우 | |
| 3-a | 시스템은 '신고 사유를 입력해주세요' 메시지를 출력하고 다시 입력을 요청한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 2 | 신고하려는 게시글/댓글이 존재하지 않는 경우 | |
| 2-e | 시스템은 '신고 대상이 존재하지 않습니다' 메시지를 출력한다. | |

## 4.6.9 신고 확인 및 조치

|  |  |  |
| --- | --- | --- |
| **Use Case #009** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 관리자가 신고된 내용을 확인하고 조치를 취한다. |
| 작성자 | | 성하윤 |
| 액터 | | 관리자 |
| 사전 조건 | | 신고된 게시글이나 댓글이 존재해야 한다. |
| 사후 조건 | | 신고에 대한 조치가 완료된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 관리자가 신고 관리 페이지에 접속한다. | |
| 2 | 시스템이 신고 내역을 리스트로 제공한다. | |
| 3 | 관리자가 신고 상세 내용을 확인한다. | |
| 4 | 관리자가 조치 유형(삭제, 경고 등)을 선택한다. | |
| 5 | 시스템이 해당 조치를 적용한다. | |
| 6 | 시스템이 조치 결과를 신고자와 피신고자에게 알린다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 4 | 관리자가 조치 내용을 미리보기로 확인하고 변경하는 경우 | |
| 4-a | 시스템은 미리보기 결과를 반영하여 다시 조치 유형 선택 화면으로 돌아간다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 4 | 조치 유형이 선택되지 않은 경우 | |
| 4-e | 시스템은 '조치 유형을 선택해주세요' 메시지를 출력하고 선택을 요청한다. | |

## 4.6.10 커뮤니티 정책 관리

|  |  |  |
| --- | --- | --- |
| **Use Case #010** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 관리자가 커뮤니티 정책을 설정하고 관리한다. |
| 작성자 | | 성하윤 |
| 액터 | | 관리자 |
| 사전 조건 | | 관리자는 관리자 권한으로 로그인되어 있어야 한다. |
| 사후 조건 | | 설정한 정책이 시스템에 반영된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 관리자가 정책 설정 페이지에 접속한다. | |
| 2 | 시스템이 현재 운영 중인 정책을 표시한다. | |
| 3 | 관리자가 정책 항목을 수정하거나 새 항목을 추가한다. | |
| 4 | 시스템이 정책 내용을 저장한다. | |
| 5 | 사용자에게 적용된 정책이 반영된다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 3 | 관리자가 기존 정책을 복사하여 새 정책을 생성하는 경우 | |
| 3-a | 시스템은 기존 정책을 기반으로 새 정책 초안을 생성하고 수정할 수 있도록 표시한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 3 | 조건에 맞는 레시피가 없을 경우 | |
| 3-e | 시스템은 '추천 가능한 레시피가 없습니다'라는 메시지를 출력한다. | |
| 4 | 저장 중 시스템 오류가 발생한 경우 | |
| 4-e | 시스템은 '정책 저장 중 오류가 발생했습니다' 메시지를 출력하고 이전 상태로 복원한다. | |

## 4.6.11 인기 게시글 작성

|  |  |  |
| --- | --- | --- |
| **Use Case #011** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 시스템이 좋아요 수 등을 기준으로 인기 게시글을 선정한다. |
| 작성자 | | 성하윤 |
| 액터 | | 관리자 |
| 사전 조건 | | 게시판에 등록된 게시글이 있어야 한다. |
| 사후 조건 | | 인기 게시글이 별도 영역에 노출된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 시스템이 주기적으로 게시글의 공감 수, 조회 수 등을 수집한다. | |
| 2 | 인기 점수를 계산하여 상위 게시글을 선정한다. | |
| 3 | 선정된 게시글을 '인기 게시글' 영역에 노출한다. | |
| 4 | 시스템이 입력값의 유효성을 검사한다. | |
| 5 | 시스템이 게시글을 데이터베이스에 저장한다. | |
| 6 | 사용자에게 '작성 완료' 메시지를 출력한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 2 | 공감 수가 유사한 게시글이 많은 경우 | |
| 2-a | 시스템은 조회 수와 댓글 수를 추가 고려하여 최종 순위를 결정한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 1 | 게시글 수가 부족한 경우 | |
| 1-e | 시스템은 '인기 게시글을 선정할 수 없습니다' 메시지를 출력한다. | |

# 클래스 다이어그램

## 전체 클래스



## 클래스 설명

|  |  |
| --- | --- |
| 클래스 | 설명 |
| 사용자(User) | 사용자 설정 관련 처리 |
| 식재료(Ingredient) | 식재료 이름, 수량, 유통기한 작업 처리 |
| 레시피(Recipe) | 레시피 추천 작업 처리 |
| 장바구니(Cart) | 추가 식재료 작업 처리 |
| 커뮤니티(Community) | 커뮤니티 공간 작업 처리 |
| 영수증(Receipt) | 영수증 식재료 AI 등록 처리 |
| AI시스템 | 분석, 식재료 공공 API처리 |

# 시퀀스 다이어그램

## 레시피 추천 기능

사용자가 추천 레시피 버튼을 클릭하면, UI가 레시피 API에 추천 요청을 보냅니다.  
레시피 API는 레시피 DB에서 다양한 레시피를 조회합니다.  
보유한 재료로 만들 수 있는 레시피를 선별해 사용자에게 추천 결과를 제공합니다.



## 영수증 입력 기능

사용자가 영수증 이미지를 업로드하면, UI가 OCR 엔진에 이미지를 전달합니다.  
OCR 엔진이 영수증에서 재료 정보를 추출해 재료 DB에 등록합니다.  
등록된 재료 정보가 사용자에게 안내됩니다.



## 커뮤니티 기능

사용자가 레시피 게시글이나 댓글을 작성하면, UI가 작성 내용을 커뮤니티DB에 저장 요청합니다.  
커뮤니티DB에 저장이 완료되면, 작성 완료 메시지가 사용자에게 전달됩니다.



## 장바구니 기능

사용자가 레시피에 필요한 재료를 장바구니에 추가하거나 구매를 요청하면, UI가 장바구니 DB와 결제 시스템에 요청을 보냅니다. 재료 추가 및 결제 처리가 완료되면, 처리 결과가 사용자에게 안내됩니다. 

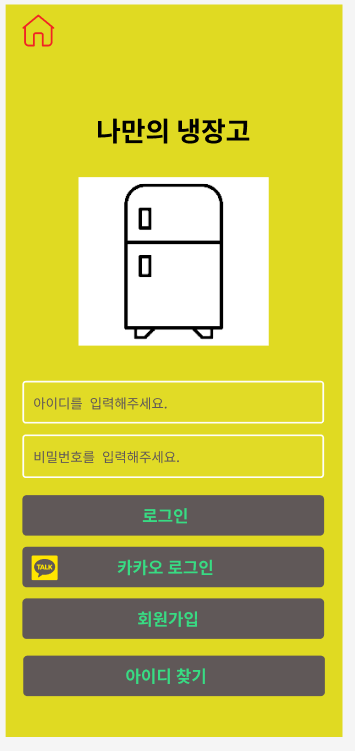
## 회원가입/로그인 기능

사용자가 회원가입이나 로그인을 시도하면, UI가 회원 정보를 회원 DB에 확인 또는 암호화하여 저장 요청합니다. 처리 결과에 따라 회원가입 또는 로그인 성공 여부가 사용자에게 안내됩니다.



# UI 설계

## 회원가입/로그인 화면



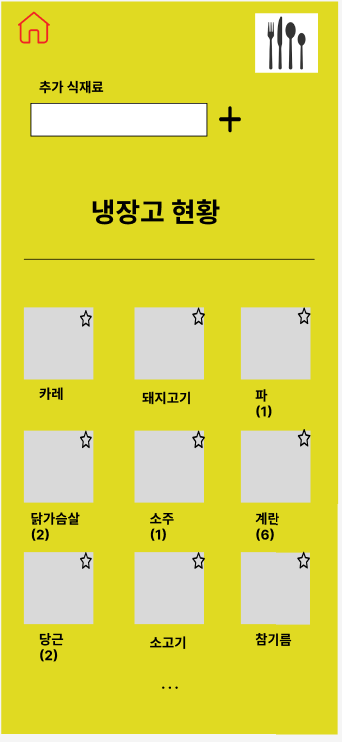
|  |  |  |
| --- | --- | --- |
| **UI #001** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 나만의 냉장고 로그인 기능 |
| 작성자 | | 성하윤 |
| 액터 | | 사용자 |
| 사전 조건 | | 사용자는 앱을 설치한 상태여야 한다. |
| 사후 조건 | | 로그인 성공 시 사용자는 홈 화면에 진입해 앱의 주요 기능을 사용할 수 있다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 어플을 실행한다. | |
| 2 | 아이디와 비밀번호를 입력한다. | |
| 2-a | 카카오톡을 로그인을 한다. | |
| 2-b | 회원가입 버튼을 누르고 회원가입을 한다. | |
| 3 | 회원가입 버튼을 누른다. | |
| 4 | 로그인 버튼을 누른다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 2 | 사용자가 아이디 또는 비밀번호를 잘못 입력했을 경우 | |
| 2-a | "아이디 또는 비밀번호가 잘못되었습니다"라는 오류 메시지를 표시한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 2 | 로그인 또는 회원가입 중 네트워크 연결이 끊긴 경우 | |
| 2-e | "네트워크 오류가 발생했습니다. 다시 시도해주세요." 메시지를 표시한다. | |

## 메인화면



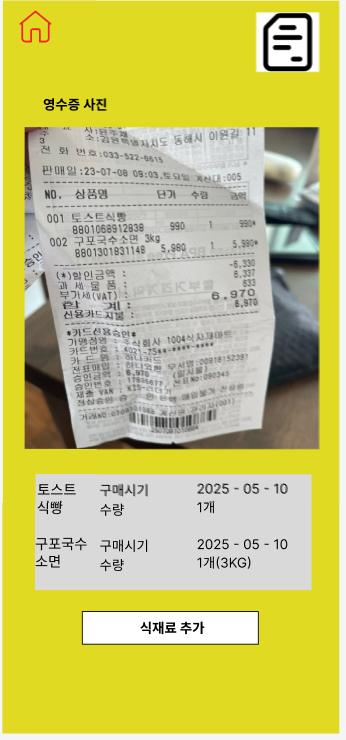
|  |  |  |
| --- | --- | --- |
| **UI #002** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 메인 화면 메뉴 |
| 작성자 | | 성하윤 |
| 액터 | | 사용자 |
| 사전 조건 | | 앱은 정상적으로 실행되어 있어야 한다. |
| 사후 조건 | | 사용자는 선택한 기능 화면으로 정상적으로 이동하여 해당 기능을 사용할 수 있다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 메인화면이 나온다 | |
| 1-a | 식재료 등록 버튼을 클릭할 경우, 식재료 등록 화면으로 넘어간다. | |
| 1-b | 영수증 등록 버튼을 클릭할 경우, 영수증 등록 화면으로 넘어간다. | |
| 1-c | 장바구니 관리 버튼을 클릭할 경우, 장바구니 관리로 넘어간다. | |
| 1-d | 레시피 추천을 클릭할 경우, 레시피 추천 화면으로 넘어간다. | |
| 1-e | 커뮤니티 게시판 버튼을 클릭할 경우, 커뮤니티 게시판 화면으로 넘어간다. | |
| 2 | 환경설정 버튼 클릭할 경우 | |
| 2-a | 앱을 로그아웃 시킨다. | |
| 2-b | 회원 탈퇴를 한다. | |
| 2-c | 시스템 알림 설정을 한다 | |
| 2-d | 닉네임 설정을 한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 1 | 메인화면이 안 나오는 경우 | |
| 1-a | 새로고침을 한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 1 | 특정 기능 화면으로 이동 중 예외가 발생한 경우 | |
| 1-e | "화면을 불러오는 데 실패했습니다. 다시 시도 해 주세요." 메시지를 표시하고, 메인화면에 머무른다. | |

## 식재료 화면



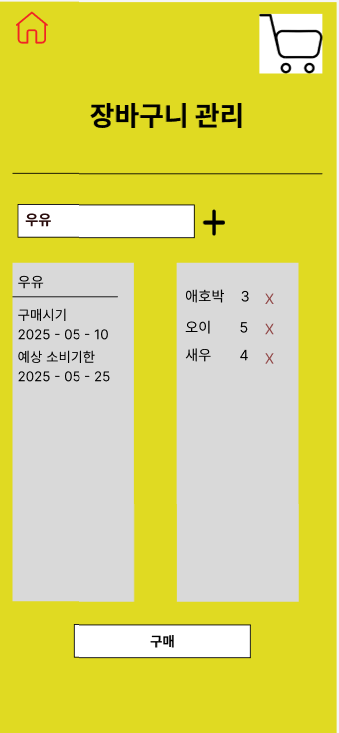
|  |  |  |
| --- | --- | --- |
| **UI #003** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 식재료 등록 |
| 작성자 | | 성하윤 |
| 액터 | | 사용자 |
| 사전 조건 | | 앱이 정상 실행되고 있어야 한다. |
| 사후 조건 | | 사용자가 등록한 식재료 정보가 앱에 저장된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 사용자가 보유 식재료를 입력하여 등록한다. | |
| 2 | 등록된 식재료 정보를 입력한다. | |
| 3 | 이미 등록된 재료들을 확인한다. | |
| 3-a | 등록된 재료를 삭제한다. | |
| 3-b | 등록된 재료를 수정한다, | |
| 4 | 메인화면 메뉴로 이동한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 1 | 등록 중 저장 오류가 발생한 경우 | |
| 1-a | 등록에 실패했습니다. 다시 시도해주세요" 메시지를 표시한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 3-a,b | 삭제/수정 중 앱이 강제 종료되거나 오류가 발생할 경우 | |
| 3-a,b-e | "작업 중 문제가 발생했습니다" 메시지를 표시하고 목록을 새로 고친다. | |

## 영수증 화면



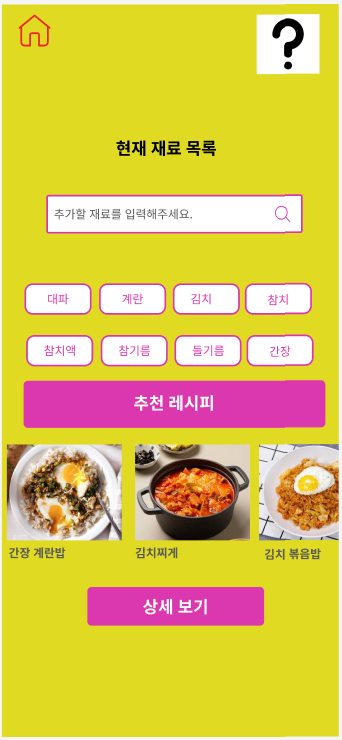
|  |  |  |
| --- | --- | --- |
| **UI #004** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 영수증 등록 |
| 작성자 | | 성하윤 |
| 액터 | | 사용자 |
| 사전 조건 | | 카메라 사용 권한이 허용되어 있어야 한다. |
| 사후 조건 | | 인식된 식재료가 사용자의 냉장고에 추가된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 카메라로 영수증을 촬영한다. | |
| 2 | 영수증 확인 후 식재료를 확인한다. | |
| 3 | 식재료 수량을 확인한다. | |
| 4 | 식재료 유통기한을 확인한다. | |
| 5 | 확인된 식재료를 냉장고에 추가한다. | |
| 6 | 메인화면 메뉴로 이동한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 1 | 영수증 글자가 흐리거나 인식이 어려운 경우 | |
| 1-a | "영수증을 정확히 인식하지 못했습니다. 다시 촬영해주세요." 메시지를 표시하고 재촬영 유도. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 1 | 사용자가 카메라 권한을 거부했을 경우 | |
| 1-e | "카메라 권한이 필요합니다"라는 메시지를 띄우고 권한 설정 화면으로 유도한다. | |

## 장바구니 화면



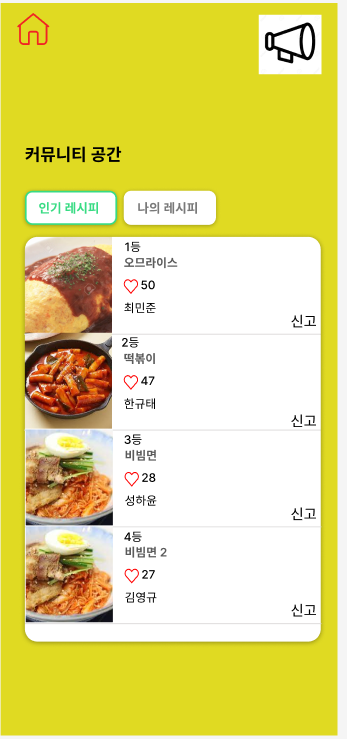
|  |  |  |
| --- | --- | --- |
| **UI #005** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 장바구니 관리 |
| 작성자 | | 성하윤 |
| 액터 | | 사용자 |
| 사전 조건 | | 앱은 정상적으로 실행되고 있어야 한다. |
| 사후 조건 | | 장바구니 내 재료가 최신 상태로 관리된다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 필요하거나 부족한 식재료를 장바구니에 추가한다. | |
| 2 | 장바구니에서 재료를 검색한다. | |
| 3 | 장바구니에 담긴 재료 정보를 수정한다. | |
| 4 | 장바구니에서 특정 재료를 제거한다. | |
| 5 | 장바구니 내 재료의 수량을 설정 또는 변경한다. | |
| 6 | 장바구니에 있는 재료를 구입한다. | |
| 7 | 메인화면 메뉴로 이동한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 5 | 수량을 0 이하로 입력하거나 잘못된 형식을 입력한 경우 | |
| 5-a | "올바른 수량을 입력해주세요." 메시지를 표시하고 수정 요청. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 2 | 사용자가 검색한 재료가 장바구니 내에 없을 경우 | |
| 2-e | “일치하는 재료가 없습니다.” 메시지 표시. | |

## 레시피 화면



|  |  |  |
| --- | --- | --- |
| **UI #006** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 레시피 추천 |
| 작성자 | | 성하윤 |
| 액터 | | 사용자 |
| 사전 조건 | | AI 추천 기능이 활성화되어 있어야 한다. |
| 사후 조건 | | 사용자는 자신에게 맞는 레시피를 찾고 실행할 수 있다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 부족한 식재료를 추가한다. | |
| 2 | 보유한 재료를 기반으로 AI가 추천한 레시피를 받는다. | |
| 3 | 재료 일부만 선택하여 만들 수 있는 레시피를 조회한다. | |
| 4 | 상세보기 버튼을을 누른다. | |
| 4-a | 추천받은 레시피를 저장해 나중에 볼 수있다. | |
| 4-b | 추천된 레시피를 커뮤니티 또는 외부로 공유한다 | |
| 5 | 메인화면 메뉴로 이동한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 2 | 보유 식재료가 부족할 경우 | |
| 2-a | 부족한 재료를 자동 제안하거나 장바구니로 유도한다. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 2 | 서버 오류 또는 네트워크 문제로 레시피 추천이 실패할 경우 | |
| 2-e | “레시피 추천에 실패했습니다. 잠시 후 다시 시도 해 주세요.” 메시지 표시. | |

## 커뮤니티 화면



|  |  |  |
| --- | --- | --- |
| **UI #007** | | |
| **GENERAL CHARACTERISTICS** | | |
| 요약 | | 커뮤니티 게시판 |
| 작성자 | | 성하윤 |
| 액터 | | 사용자 |
| 사전 조건 | | 커뮤니티 기능이 활성화되어 있어야 한다 |
| 사후 조건 | | 사용자가 작성한 레시피는 커뮤니티 내에서 게시되고, 본인이 자유롭게 관리할 수 있다. |
| **기본 흐름** | | |
| 단계 | 행위 | |
| 1 | 인기 레시피 클릭 | |
| 1-a | 시스템이 좋아요 수 등을 기준으로 인기 게시글을 선정한다. | |
| 1-b | 부적절한 게시물을 신고한다. | |
| 2 | 나의 레시피 클릭한다. | |
| 3 | 사용자가 직접 요리법을 입력하여 레시피 게시글을 작성한다. | |
| 4 | 작성한 레시피를 작성, 편집 한다. | |
| **대안 흐름** | | |
| 단계 | 행동 | |
| 3 | 다른 사용자의 게시글을 수정 시도하거나, 삭제 권한이 없는 경우 | |
| 3-a | “해당 게시글을 수정/삭제할 수 없습니다.” 메시지 표시. | |
| **예외 흐름** | | |
| 단계 | 행동 | |
| 1 | 네트워크 오류나 서버 오류로 게시글 로딩에 실패할 경우 | |
| 1-e | “게시글을 불러오지 못했습니다. 다시 시도 해 주세요.” 메시지 표시. | |

# 지원 정보

• 프론트엔드: React Native (iOS/Android 크로스 플랫폼)

• 백엔드: Node.js (Express.js 기반 REST API)

• 데이터베이스: MySQL (AWS RDS 또는 로컬)

• 클라우드 서비스

* AWS S3 (이미지 저장)
* Firebase Cloud Messaging (푸시 알림)
* Kakao Login API (사용자 인증)

• AI 연동: 외부 레시피 추천 API, OCR 분석 API

• 배포 환경: 로컬 서버 또는 Firebase Hosting 테스트 환경

# 참고 문헌

• 전공책 (sw 융합 시스템 분석 설계, 최용식, 2022)

• 구글

• GPT-4, OpenAI (참고용 사용)

• 공공데이터포털 – 식품 영양 정보 및 유통기한 API

• Kakao Developers 문서 – <https://developers.kakao.com>

• Firebase Cloud Messaging (FCM) 공식 문서 – <https://firebase.google.com/docs/cloud-messaging>

• Edamam 레시피 API – <https://developer.edamam.com>

• Spoonacular 레시피API - <https://spoonacular.com/food-api/docs>

• AWS S3 공식 문서 - <https://docs.aws.amazon.com/s3/>

• OCR 깃허브 - <https://github.com/tesseract-ocr/tesseract>