**요구사항 명세서**

버전: 0.0.1

작성일: 2024.04.01

작성자: 이재윤

목 차

[1. 개요 5](#_Toc163363779)

[2. 기능적 요구사항 6](#_Toc163363780)

[2.1. 시스템 기능 구조 6](#_Toc163363781)

[2.1.1. 유스케이스 패키지 구조도 6](#_Toc163363782)

[2.1.2. 유스케이스 패키지 개요 6](#_Toc163363783)

[2.2. 유스케이스 패키지 명세: 푸드 코스트 시스템 7](#_Toc163363784)

[2.2.1. 유스케이스 다이어그램 7](#_Toc163363785)

[2.2.2. 액터 개요 7](#_Toc163363786)

[2.2.3. 유스케이스 개요 8](#_Toc163363787)

[2.2.4. 유스케이스 명세: 식재료 관리 8](#_Toc163363788)

[2.2.4.1. 개요 8](#_Toc163363789)

[2.2.4.2. 관련 액터 8](#_Toc163363790)

[2.2.4.3. 우선순위 9](#_Toc163363791)

[2.2.4.4. 선행 조건 9](#_Toc163363792)

[2.2.4.5. 후행 조건 9](#_Toc163363793)

[2.2.4.6. 시나리오 9](#_Toc163363794)

[2.2.4.7. 비기능적 요구사항 10](#_Toc163363795)

[2.2.5. 유스케이스 명세: 요리 레시피 관리 10](#_Toc163363796)

[2.2.5.1. 개요 10](#_Toc163363797)

[2.2.5.2. 관련 액터 11](#_Toc163363798)

[2.2.5.3. 우선순위 11](#_Toc163363799)

[2.2.5.4. 선행 조건 11](#_Toc163363800)

[2.2.5.5. 후행 조건 11](#_Toc163363801)

[2.2.5.6. 시나리오 11](#_Toc163363802)

[2.2.5.7. 비기능적 요구사항 12](#_Toc163363803)

[2.2.6. 유스케이스 명세: 원가 계산 관리 13](#_Toc163363804)

[2.2.6.1. 개요 13](#_Toc163363805)

[2.2.6.2. 관련 액터 13](#_Toc163363806)

[2.2.6.5. 후행 조건 14](#_Toc163363807)

[2.2.6.6. 시나리오 14](#_Toc163363808)

[2.2.6.7. 비기능적 요구사항 14](#_Toc163363809)

[2.2.7. 유스케이스 명세: 정산 관리 14](#_Toc163363810)

[2.2.7.1. 개요 14](#_Toc163363811)

[2.2.7.2. 관련 액터 15](#_Toc163363812)

[2.2.7.3. 우선순위 15](#_Toc163363813)

[2.2.7.4. 선행 조건 15](#_Toc163363814)

[2.2.7.5. 후행 조건 15](#_Toc163363815)

[2.2.7.6. 시나리오 15](#_Toc163363816)

[2.2.7.7. 비기능적 요구사항 16](#_Toc163363817)

[3. 시스템 품질 요구사항 17](#_Toc163363818)

[3.1. 성능 17](#_Toc163363819)

[3.2. 신뢰도 17](#_Toc163363820)

[3.3. 확장성 17](#_Toc163363821)

[3.4 보안성 18](#_Toc163363822)

[4. 개발 제약 사항 19](#_Toc163363823)

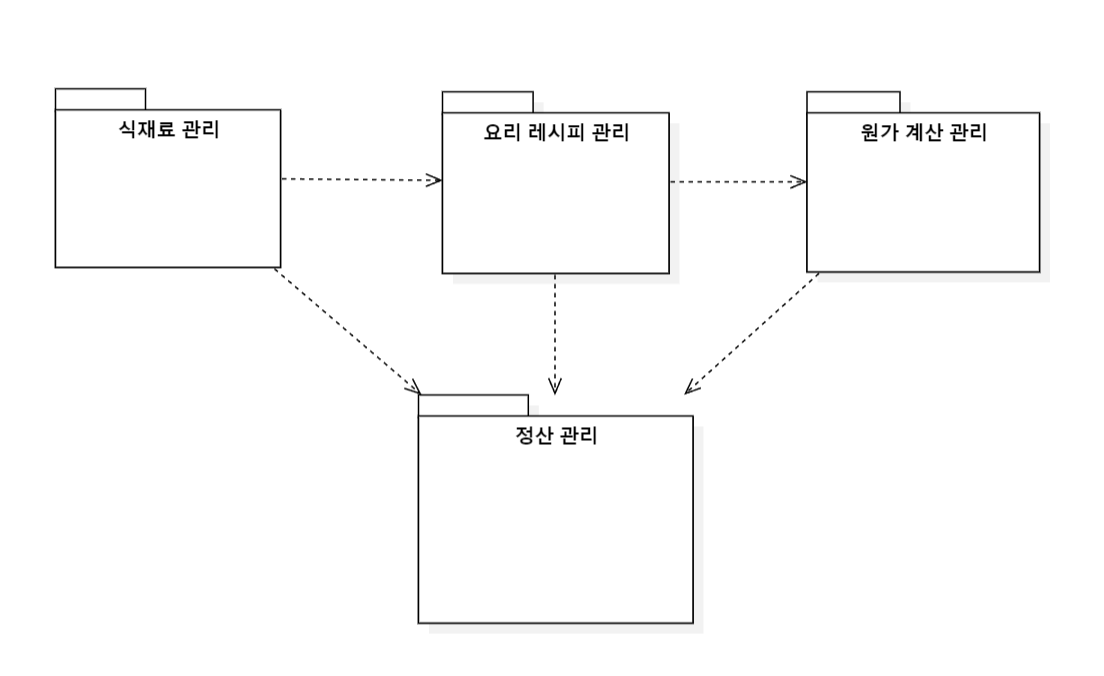
# 개요

|  |
| --- |
| 문서의 목적은 푸드코스트 프로그램의 요구사항을 명확하게 정의하고 문서화하여 프로젝트의 이해를 돕고, 개발 과정을 지침하며, 최종 제품의 품질을 보장하는 데 있습니다.  이 요구사항 명세서는 프로그램의 기능, 사용자 요구사항, 시스템 제약조건 등을 기술합니다. 문서는 개발 프로세스의 시작점으로 사용되며, 개발팀과 이해관계자들 간의 의사소통 도구로 작용합니다. 명세서에는 사용자의 요구사항, 시스템의 기능 및 비기능적 요구사항, 시스템의 제약조건, 데이터 모델, 인터페이스, 테스트 요구사항 등이 포함됩니다. 이 문서는 프로젝트의 전반적인 이해와 개발의 방향성을 제공하기 위해 사용될 것입니다. |

# 기능적 요구사항

## 시스템 기능 구조

### 유스케이스 패키지 구조도



### 유스케이스 패키지 개요

|  |  |
| --- | --- |
| 패키지 명 | 설명 |
| 식재료 관리 패키지 | 보유한 식재료의 수량, 보관위치, 유통기한 관리 |
| 요리 레시피 관리 패키지 | 요리를 할 때 들어가는 식재료의 양, 시간 관리 |
| 원가 계산 패키지 | 위 두 패키지를 이용하여, 레시피에 들어가는 식재료 값 계산 |
| 정산 패키지 | 재고의 변화를 통해 어떠한 식재료가 많이 사용되는지 정리 |

## 유스케이스 패키지 명세: 푸드 코스트 시스템

### 유스케이스 다이어그램

*텍스트, 도표, 라인, 그림이(가) 표시된 사진

자동 생성된 설명*

### 액터 개요

|  |  |  |
| --- | --- | --- |
| 액터 명 | 유형 | 설명 |
| 시스템 사용자 | 사용자 | 프로그램을 사용할 소상공인(사장) |
| 현장 관리자 | 사용자 | 가게에서 일하는 관리자 |
| 유통기한 알림 시스템 | 시스템 | 재고의 유통기한을 알려주는 시스템 |
| 원가 계산 시스템 | 시스템 | 레시피의 원가에 대해 계산해주는 시스템 |
| 정산 관리 시스템 | 시스템 | 가게의 전반적인 매출을 관리해주는 시스템 |

### 유스케이스 개요

|  |  |
| --- | --- |
| 유스케이스 명 | 설명 |
| 식재료 관리 | 식재료의 수량, 보관위치, 유통기한 관리 |
| 요리 레시피 관리 | 요리 별 식재료 사용량 관리 |
| 원가 계산 관리 | 요리 별 레시피를 사용한 가격 관리 |
| 정산 관리 | 남은 재고의 변화 및 매출 관리 |

### 유스케이스 명세: 식재료 관리

#### 개요

* 식재료 관리는 사용자가 보유한 식재료의 수량, 보관위치, 유통기한을 관리하는 기능을 가진다.
* 사용자와 현장관리자가 추가, 수정, 삭제할 수 있다.
* 식재료의 재고의 변화를 파악한다.
* 식재료의 유통기한이 다가올 때 알림을 제공하는 기능을 가진다.

#### 관련 액터

주액터: 사용자, 현장관리자

보조액터: 유통기한 알림 시스템

#### 우선순위

중요도 상, 난이도 중

#### 선행 조건

* 사용자가 시스템에 로그인한 상태여야 한다.
* 유통기한에 관한 알림을 받고자 하는 항목에 대하여 그 식재료의 유통기한에 관한 데이터가 입력되어 있어야 한다.

#### 후행 조건

* 사용자가 입력한 식재료의 데이터는 저장되어야 한다.
* 유통기한이 지난 품목에 표시를 한다.
* 유통기한이 지난 품목의 재고를 재고 없음 상태로 변경시킨다.

#### 시나리오

**데이터 저장 시나리오:**

1. 사용자는 식재료관리 화면에서 추가 버튼을 눌러 사용하는 식재료를 추가한다.
2. 시스템은 추가할 식재료에 입력할 때 필요한 가격, 유통기한, 이름, 수량 등을 요구한다.
3. 입력을 완료하고 “저장” 버튼을 선택한다.
4. 저장된 식재료는 데이터프레임의 형태로 저장된다.
5. 시스템은 저장된 데이터를 보여준다.

**데이터 수정 시나리오:**

1. 사용자는 식재료 관리 화면에서 수정 버튼을 클릭한다.
2. 수정할 데이터를 선택하여 수정하고 “저장” 버튼을 선택한다.
3. 데이터 프레임이 수정된 데이터로 저장된다.

**데이터 삭제 시나리오:**

1. 사용자는 식재료 관리 화면에서 삭제 버튼을 클릭한다.
2. 삭제할 식재료 종류들을 선택하고 “삭제” 버튼을 선택한다.
3. 시스템은 “선택한 데이터를 삭제하겠습니까?” 라는 창을 통해 삭제 여부를 물어본다.
4. “예” 를 클릭할 경우, 데이터를 삭제하고 저장하고, “아니오” 를 클릭할 경우 2. 로 돌아간다.

**유통기한 알림 시나리오: 유통기한을 입력한 데이터에 한해서**

1. 유통기한이 지나게 되면 그 품목에 표시를 해준다.
2. 시스템은 그 품목의 재고를 0으로 바꾼다.

#### 비기능적 요구사항

해당 없음

### 유스케이스 명세: 요리 레시피 관리

#### 개요

* 요리 레시피 관리는 사용자가 가게에서 사용하는 레시피를 입력하는 기능을 가진다.
* 사용자와 현장관리자가 추가, 수정, 삭제할 수 있다.
* 한 음식을 만들 때 필요한 식재료들의 양을 파악한다.
* 파악한 양을 통해 원가의 계산에 사용된다.
* 또한, 재고의 변화와의 비교를 통해 인기메뉴 선정에 사용된다.

#### 관련 액터

주액터 : 사용자, 현장관리자

보조액터 : 원가 계산 관리 시스템

#### 우선순위

중요도 상, 난이도 중

#### 선행 조건

* 사용자가 시스템에 로그인한 상태여야 한다.

#### 후행 조건

* 사용자가 입력한 레시피의 데이터는 저장되어야 한다.

#### 시나리오

**데이터 저장 시나리오:**

1. 사용자는 레시피 관리 화면에서 추가 버튼을 눌러 요리들의 레시피를 입력한다.
2. 시스템은 추가할 레시피의 필요한 식재료의 종류와 양을 요구한다.
3. 입력을 완료하고 “저장” 버튼을 선택한다.
4. 입력이 완료된 레시피는 딕셔너리의 형태로 저장한다.
5. 시스템은 저장된 레시피의 이름만을 보여준다.
6. 레시피의 이름을 클릭할 경우, 시스템은 레시피의 상세정보를 보여준다.

**데이터 수정 시나리오:**

1. 사용자는 레시피 관리 화면에서 수정할 레시피를 클릭한다.
2. 레시피의 상세정보페이지에서 “수정” 버튼을 선택한다.
3. 수정할 데이터를 선택하여 수정하고 “저장” 버튼을 선택한다.
4. 딕셔너리의 데이터를 수정된 데이터로 저장한다.

**데이터 삭제 시나리오:**

1. 사용자는 요리 레시피 관리 화면에서 삭제 버튼을 클릭한다.
2. 삭제할 레시피 종류들을 선택하고 “삭제” 버튼을 선택한다.
3. 시스템은 “선택한 데이터를 삭제하겠습니까?” 라는 창을 통해 삭제 여부를 물어본다.
4. 예” 를 클릭할 경우, 데이터를 삭제하고 저장하고, “아니오” 를 클릭할 경우 2. 로 돌아간다.

#### 비기능적 요구사항

해당 없음

### 2.2.6. 유스케이스 명세: 원가 계산 관리

#### 2.2.6.1. 개요

* 사용자가 입력한 식재료 관리 데이터와, 요리 레시피 관리 데이터를 바탕으로 레시피의 원가를 계산해주는 기능을 가진다.
* 사용자가 요리 레시피를 입력할 경우 자동으로 생성된다.
* 사용자가 식재료 관리나 요리 레시피의 데이터를 수정할 경우 자동으로 데이터가 수정된다.
* 사용자가 판매금액을 설정하였을 경우, 원가율 또한 계산하여 저장한다.

#### 2.2.6.2. 관련 액터

주액터 : 원가 계산관리 시스템

보조액터 : 사용자

2.2.6.3 우선순위

중요도 중, 난이도 상

2.2.6.4 선행 조건

* 사용자는 시스템에 로그인한 상태여야 한다.
* 식재료 관리에서의 항목과, 요리 레시피 항목이 데이터로 저장되어 있어야 한다.
* 원가율에 대한 정보가 필요할 경우 요리 레시피 관리의 항목으로 판매금액을 입력해야 한다.

#### 2.2.6.5. 후행 조건

* 사용자가 입력한 각 레시피의 항목으로 저장되어야 한다.

#### 2.2.6.6. 시나리오

**데이터 생성 시나리오:**

1. 사용자가 식재료 관리에서 데이터를 입력하고, 요리 레시피 관리에서 레시피를 입력한다.
2. 입력된 데이터를 바탕으로 각 레시피의 푸드코스트를 계산하여 레시피의 항목으로 자동 저장한다.
3. 요리 레시피 관리의 항목에서 판매금액이 입력되어 있을 경우 원가율을 계산하여 레시피의 항목으로 저장한다.

#### 2.2.6.7. 비기능적 요구사항

### 2.2.7. 유스케이스 명세: 정산 관리

#### 2.2.7.1. 개요

* 발생한 매출과 지출을 추적하고 관리하는 기능을 제공한다.
* 사용자는 지출내역 및 매출 내역을 작성한다.
* 프로그램은 저장되어 있는 데이터를 기반으로 이익을 계산하여 저장한다.

#### 2.2.7.2. 관련 액터

* 주 액터 : .정산관리시스템
* 보조 액터 : 사용자

#### 2.2.7.3. 우선순위

우선순위 하, 난이도 상

#### 2.2.7.4. 선행 조건

* 사용자는 시스템에 로그인한 상태여야 한다.
* 식재료 관리에서의 항목과, 요리 레시피 항목이 데이터로 저장되어 있어야 한다.
* 매출 및 지출에 대한 데이터가 시스템에 입력되어 있어야 한다.

#### 2.2.7.5. 후행 조건

* 사용자가 입력한 매출 및 지출 데이터는 시스템에 저장되어야 한다.
* 수익과 손익 계산이 완료되어야 한다.
* 재무 상태가 갱신되고 저장되어야 한다.

#### 2.2.7.6. 시나리오

**매출 기록 시나리오:**

* 사용자는 정산 관리 화면에서 매출을 입력한다.
* 매출에 대한 세부 정보(날짜, 품목, 수량, 가격 등)를 입력한다.
* "저장" 버튼을 클릭하여 데이터를 저장한다.
* 시스템은 입력된 매출 데이터를 확인하고 저장한다.

**지출 추적 시나리오:**

* 사용자는 정산 관리 화면에서 지출을 추적한다.
* 지출에 대한 정보(날짜, 품목, 비용, 비고 등)를 입력한다.
* "저장" 버튼을 클릭하여 데이터를 저장한다.
* 시스템은 입력된 지출 데이터를 확인하고 저장한다.

**수익 및 손익 계산 시나리오:**

* 시스템은 저장된 매출 및 지출 데이터를 기반으로 수익과 손익을 계산한다.
* 수익과 손익 계산 결과가 사용자에게 표시된다.

#### 2.2.7.7. 비기능적 요구사항

해당 없음

# 시스템 품질 요구사항

## 3.1. 성능

- 식재료 관리 기능은 재고 업데이트, 유통기한 관리, 식재료 주문 등의 작업을 1초 내외로 처리할 수 있어야 합니다.

- 요리 레시피 관리 기능은 다양한 레시피와 그에 따른 재료 리스트를 빠르게 업데이트 및 검색 할 수 있어야 합니다.

- 원가 계산 관리 기능은 레시피 비용 산출, 재료 가격 등을 정확하게 계산하여 처리할 수 있어야합니다.

## 3.2. 신뢰도

- 시스템이 지속적으로 안정적으로 운영되어야 하며 연간 가동률이 99% 이상 연중무휴 가동을 목표를 하여야합니다.

- 장애 발생 시 자동으로 문제를 감지하고 빠르게 복구할 수 있는 프로세스를 갖추어야 합니다.

- 데이터 백업 및 복구 할 수 있는 매커니즘을 통해 데이터 손실없이 복구 할 수 있어야합니다.

- 장애 발생 및 복구 기록은 투명하게 관리되어야하며, 시스템 안정성을 지속적으로 개선 할 수 있어야 합니다.

## 3.3. 확장성

- 시스템 아키텍처는 서비스 확장을 고려하여 설계되어야합니다.

- 성능 저하 없이 데이터 양이 50% 이상 증가해도 처리 능력을 유지할 수 있는 물리적 인프라 설계가 필요합니다.

- 데이터의 증가에 따라 서비스 병목 현상 없이 안정적으로 운영될 수 있도록 설계되어야 합니다.

- 확장성 검증하기 위해 성능과 용량을 주기적으로 테스트를 해야합니다.

## 3.4 보안성

- 사용자 인증 및 권한 관리는 엄격한 접근 제어 정책을 통해 이루어져야 하며, 모든 사용자 활동을 추적할 수 있어야 합니다.

- 민감한 정보에 대한 접근은 추가적인 보안 검증 절차를 거쳐야 합니다.

- 보안 검증 절차(로그인 기능)에 로그인 5회 이상 실패시 비밀번호를 초기화 할 수 있도록 침입 시도가 발생 시 사전에 차단할 수 있어야 합니다.

# 개발 제약 사항

1. GUI를 사용하여 개발한 경험이 없다.

2. 에자일 기법으로 소프트웨어를 개발하므로 요구사항이 계속해서 변할 수 있음