INTERNATIONAL STANDARD

ISO/IEC/IEEE 29148 : 2017

< 지구온난화에 따라 변화하는 제철 농산물의 수확 시기 및   
도매상 API정보를 통한 농산물의 가격 예측 서비스 >

**Requirements Analysis** **Document**

**Version 0.2**

Team CP/CF

**2024.09.20  
양 호 준**

**Table of Contents**

Table of Contents

**List of Table and Figures**

**Revision History**

[**1.** **서론** 4](#_Toc173832065)

[1.1 **개발 목적** 4](#_Toc173832066)

[**1.2** **문서 규칙** 4](#_Toc173832067)

[**1.3** **용어정의 및 약어** 4](#_Toc173832068)

[**1.4** **제품 범위** 5](#_Toc173832069)

[**1.5** **참고문헌 및 자료** 5](#_Toc173832070)

[**2.** **전반적인 기술** 6](#_Toc173832071)

[**2.1** **제품 관점** 6](#_Toc173832072)

[**2.2** **제품 기능** 6](#_Toc173832073)

[**2.3** **사용자 계층과 특성** 8](#_Toc173832074)

[**2.4** **운영 환경** 9](#_Toc173832075)

[**2.5** **설계 및 제약 사항** 11](#_Toc173832076)

[**2.6** **사용자 문서** 12](#_Toc173832084)

[**2.7** **가정 및 종속성** 13](#_Toc173832085)

[**3.** **외부 인터페이스 요구** ※ **임의로 작성. 백엔드 개발 시 업데이트 예정** 13](#_Toc173832086)

[**3.1** **사용자 인터페이스(User Interfaces)** 13](#_Toc173832087)

[**3.2** **하드웨어 인터페이스(HardwareInterfaces)** 13](#_Toc173832088)

[**3.3** **소프트웨어 인터페이스(Software Interfaces)** 13](#_Toc173832089)

[**3.4** **통신 인터페이스(Communications Interfaces)** 14](#_Toc173832090)

[**4.** **시스템 특징 (Use Cases)** 15](#_Toc173832091)

[**4.1** **Use Case < 은퇴자의 저소득 고령층 진입 시기 예측 >** 15](#_Toc173832092)

[공통사항 16](#_Toc173832093)

[**5.** **기타 비기능적 요구사항** 18](#_Toc173832094)

[**5.1** **성능요건** 18](#_Toc173832095)

[**5.2** **안전 요구 사항** 18](#_Toc173832096)

[**5.3** **보안 요구 사항***.* 18](#_Toc173832097)

[**5.4** **Software Quality Attributes** 18](#_Toc173832098)

[**6.** **Other Requirements** 19](#_Toc173832099)

**List of Table and Figures**

[표1. 용어정의 및 약어](#_Toc381699876) 4

[표2. 기능적 요구사항1 7](#_Toc381699877)

[표3. 기능적 요구사항2 7](#_Toc381699878)

[표4.사용자 그룹 정의 및 요구사항1 8](#_Toc381699877)

[표5.사용자 그룹 정의 및 요구사항2 9](#_Toc381699877)

[그림1. 운영환경 10](#_Toc381699878)

[그림2. User flow diagram1 15](#_Toc381699878)

[그림2. User flow diagram2 17](#_Toc381699878)

**Revision History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** | **Date of Approval** |
| 양 호 준 | 2024.08.28. | 초안 작성 | Ver0.1 | -- |
| 양 호 준 | 2024.09.20. | 다이어그램 추가 | Ver0.2 |  |

# **서론**

#데이터 작성 완료 후 예측값 간략설명

## **개발배경과 목적**

1.1.1 개발배경

지구 온난화가 진행됨에 따라, 기후변화로 인한 농작물의 재배적지 변경 및 생산량 감소가 발생하였고, 이에 따라 수입물가와 국내물가가 증가하는 악영향이 발생하고 있다. 유엔 산하 정부 간 기후변화협의체(IPCC)의 기후변화 4차 평가보고서에 따르면, 2100년까지 평균 1.1∼6.4℃ 상승할 것으로 예상하고 있다. 또한, 폭염, 집중호우 등 이상기상 현상의 빈도나, 태풍의 세기가 강해질 것이라 예측하고 있다. 국내의 국립기상연구소(2011,2012)의 기후변화 시나리오에 따르면 (2070~2099)의 한반도 평균 기온은 6.0℃ 상승, 강수량은 20.4% 증가가 예상하고 있다.   
  
 이러한 기후변화는 농산물 재배 생산량과 농산물 가격에 큰 영향을 미친다. 국내 인플레이션에 미치는 영향(국민 은행)'에 따르면 대부분의 선행연구(Parker, 2018; Mukherjee and Ouattara, 2021 등)는 기후변화가 부정적 공급충격을 통해 물가를 높이는 요인으로 작용하며, 국내의 연구(농촌진흥천, 2022) 또한, 과실의 생산량 감소의 원인을 온난화로 지목하고 있다. 이와 같이 국내, 외 적으로 기후 농산물의 연관성에 대한 연구에 주목하고있다. 몇 가지 대표 사례를 살펴보면 '기후변화가 농업부문에 미치는 경제적 영향 분석 (김창길 외 1, 2009)'에 따르면 온 상승으로 인해 작물의 생육기간이 변화하고 있으며, 이는 농산물의 생산시기와 품질에 직접적인 영향을 미친다고 보고 있으며, '기후변화가 국내 인플레이션에 미치는 영향 (조병수 외1, 한국은행,2024)'에 따르면 간접경로(글로벌 기후변화 ->국제 식량가격 변화-> 수입물가 변화 -> 국내물가 변화)와 직접경로(국내 기후변화 ->국내 농산물가격 변화)를 통해 영향을 끼치며, 기온이 1℃ 상승하는 경우 농산물가격 상승률은 0.4 ~ 0.5%p, 전체 소비자물가지수 상승률은 0.07%p 높아지는 것으로 분석하였다

1.1.2 개발목적

' 농산물 가격 예측 모형 개발 연구 (김관수 외, 2017 )'는 정확한 농산물 가격 예측이 농가 소득 안정화와 소비자 물가 안정에 중요한 역할을 한다고 밝히고 있다. 이에 따라 본 프로젝트의 목적은 **온난화**에 따른 **기후 변화**로 인해 변화하는 **제철 농산물의 수확 시기와 도소매 가격을** 예측하여 알려줌으로써 농부와 소비자에게 유용한 정보를 제공해줌으로써 농산물 재배와 소비자 물가를 안정시킴에 있다.

## **문서 규칙**

본 문서는 ISO/IEC/IEEE 29148:2017표준의 틀을 따르며, 세부 항목에 대한 기술은 TTAS.KO-11.0022, 소프트웨어 요구명세서 표준을 참고하여 작성하였고, 카테고리 구분 및 요구사항은 IEEE 829-1998표준에 따라 다음과 같이 구분한다.

1.2.1 작성 표준 예 : REQ-FUNC-001

- FUNC: 기능적 요구사항

- PERF: 성능 요구사항

- INTF: 인터페이스 요구사항

- SEC: 보안 요구사항

- DATA: 데이터 요구사항

1.2.2 기술적 용어나 약어는 처음 사용될 때 정의되며, 필요한 경우 용어집에 포함됨.

* + 1. 표와 그림은 '표 X' 또는 '그림 X' 형식으로 번호가 매겨지며, 여기서 X는 순차적인 번호

## **용어정의 및 약어**

|  |  |  |
| --- | --- | --- |
| **ID** | **Stakeholder** | **Description** |
| S-1 | 시계열 분석 | 시계열(시간의 흐름에 따라 기록된 것) 자료(data)를 분석하고 여러 변수들간의 인과관계를 분석하는 방법론 |
| S-2 | LSTM 모델 | RNN의 한 종류로, RNN의 장기 의존성 문제(long-term dependencies)를 해결하기 위해서 나온 모델 |
| S-3 | UNFCCC | 유엔기후변화협약 |
| S-4 | STL | 시계열 분해“Seasonal and Trend decomposition using Loess(Loess를 사용한 계절성과 추세 분해)”의 약자로 여기에서 Loess는 비선형 관계를 추정하기 위한 기법 |
| S-5 | 시계열 | 일정 시간 간격으로 배치된 데이터들의 수열 |

<표1. 용어정의 및 약어>

## **제품 범위**

본 프로젝트는 농산물 가격 예측 및 제철 농산물 추천 서비스로, 기후변화에 따른 농산물 생산 시기 변화와 가격 변동에 대응하기 위해 개발함.

본 서비스는 농업 생산자, 유통업자, 소비자, 그리고 농업 정책을 수립하는 정부 관계자들에게 농산물의 수확 시기 변화와 가격 예측 결과를 제공하여 적절한 대응 전략을 수립할 수 있도록 지원하는 기능을 가짐.  
  
 1차적으로 'LSTM모델과 시계열 분해 기법을 활용한 농산물 가격예측'을 구현 후 보완하여, 특정 시기의 농산물 가격을 예측하고, 2차적으로 지구 온난화에 따른 계절의 변화를 알아내어 제철 농산물의 수확시기를 찾아낸다. 최종적으로 두 가지를 합쳐서 지구온난화에 따라 변화하는 수확시기와 이에 따른 농산물 가격 변화의 타당성을 확인 후, 이후의 예측 결과를 소비자와 농부에게 제공한다.

이 서비스는 팬데믹으로 인한 급격한 인구증감 및 경제변동을 고려하지 않는다

**1.4.1 주요 기능**

1. 기상 데이터 분석을 통한 미래 온도 예측
2. 온도 변화에 따른 제철 농산물 수확 시기 예측(1년)
3. 농산물의 도소매 가격 예측 (1주, 2주, 4주 후 )
4. 실시간 기온 데이터를 활용한 농산물 추천 서비스
5. 예측 및 분석 결과의 시각화 및 리포트 생성

**1.4.2 시스템 범위**

1. 기상 데이터 및 도소매 가격 데이터 수집 및 분석: 다양한 데이터 소스에서 기상 데이터와 도소매 가격 데이터를 수집하고, 이를 분석하여 예측에 활용합니다.
2. 수확 시기 예측 모델 개발: 농산물의 최적 수확 시기를 예측하는 LSTM 기반 모델을 개발합니다.
3. 가격 변동 예측 알고리즘 구현: 농산물의 도소매 가격 변동을 예측하는 알고리즘을 구현합니다.
4. 데이터 시각화 및 웹 기반 정보 제공 시스템 구축: 예측 데이터를 시각화하고 웹 기반 정보 제공 시스템을 구축하여 사용자에게 제공됩니다.

본 프로젝트의 전체 생명주기 동안 지속적인 요구사항 관리와 추가적인 데이터 분석을 통해 데이터의 정밀도를 높이는 것을 목표로 한다.

## **참고문헌 및 자료**

[1] [제2024-18호] 기후변화가 국내 인플레이션에 미치는 영향. 김창길 외 1, 2009

[2] 기후변화가 국내 인플레이션에 미치는 영향, 조병수 외1, 2024

[3] 기후변화가 식량공급에 미치는 영향분석과 대응방안, 김창길 외 4, 2012

[4] 이상기후가 농업부분에 미치는 경제적 영향 분석, 성재훈 외 2, 2019

[5] 기후변화가 세계농업에 미치는 영향과 대응, 이변우, 2012

[6] 농산물 가격 예측 모형 개발 연구, 김광수 외, 2017

[7] 팀가온 LSTM모델과 시계열 분해 기법을 활용한 농산물 가격 예측

[8] 기후변화가 세계농업에 미치는 영향과 대응, 2021 농산물 가격예측 AI 경진대회

# **전반적인 기술**

## **제품 관점**

본 프로젝트는 기후변화에 따른 농산물 생산 및 가격 변동에 대응하기 위한 종합적인 예측 및 추천 서비스로, 1차적으로 LSTM 모델과 시계열 분해 기법을 활용하여 높은 정확도의 농산물 가격 예측을 제공, 2차적으로 기상 온도를 분석하여 계절 변화를 예측하고, 1차의 생산시기를 더하여 제철 음식의 수확시기를 예측하고자 한다..

1. 시스템 인터페이스: 시스템 인터페이스
   1. 시스템 인터페이스 열거
      1. 농넷(Nongnet) API 인터페이스: 일별 농산물 거래 데이터 수집
      2. 기상청 API 인터페이스: 기상 데이터 수집
      3. 웹 애플리케이션 서버 인터페이스: Spring Boot 기반
      4. 데이터베이스 서버 인터페이스: MySQL 8.0
      5. 분석 서버 인터페이스: Python 3.8, TensorFlow 2.4
   2. 요구사항 달성을 위한 기능성 파악 및 연결 페이스
      1. 농넷 API와의 연동을 통한 실시간 농산물 거래 데이터 수집 (일 1회)
      2. 기상청 API를 통한 기상 데이터 수집 (시간별)
      3. 웹 서버와 분석 서버 간 예측 결과 전송 인터페이스 (RESTful API)
      4. 데이터베이스와 분석 서버 간 대용량 데이터 처리 인터페이스 (Batch Processing)
2. 사용자 인터페이스
   1. 소프트웨어 제품과 사용자간의인터페이스들의 논리적 특성
      1. React 기반의 반응형 웹 디자인
      2. 대시보드 형태의 메인 화면 (농산물 가격 추이 그래프 중심)
      3. 드롭다운 메뉴를 통한 2개 농산물 품목 선택 기능
      4. 날짜 선택기를 통한 1주, 2주, 4주 예측 기간 설정 기능
   2. Ex)화면 형식, 윈도우 배치, 보고서나 메뉴 내용, 프로그래밍 가능한 기능(화면설계서 같은 내용)
      1. 상단 메뉴바: 홈, 농산물 선택, 예측 결과, 추천 농산물, 설정
      2. 중앙 대시보드: 선택된 농산물의 가격 추이 그래프, 예측 가격 표시
      3. 우측 사이드바: 실시간 추천 농산물 목록 (현재 기온 기준)
      4. 하단: 상세 데이터 테이블 및 CSV 다운로드 옵션
3. 하드웨어 인터페이스
   1. 소프트웨어 제품과 사용자간의인터페이스들의 논리적 특성
      1. React 기반의 반응형 웹 디자인
      2. 대시보드 형태의 메인 화면 (농산물 가격 추이 그래프 중심)
      3. 드롭다운 메뉴를 통한 2개 농산물 품목 선택 기능
      4. 날짜 선택기를 통한 1주, 2주, 4주 예측 기간 설정 기능
   2. Ex)화면 형식, 윈도우 배치, 보고서나 메뉴 내용, 프로그래밍 가능한 기능(화면설계서 같은 내용)
      1. 상단 메뉴바: 홈, 농산물 선택, 예측 결과, 추천 농산물, 설정
      2. 중앙 대시보드: 선택된 농산물의 가격 추이 그래프, 예측 가격 표시
      3. 우측 사이드바: 실시간 추천 농산물 목록 (현재 기온 기준)
      4. 하단: 상세 데이터 테이블 및 CSV 다운로드 옵션
4. 소프트웨어 인터페이스
   1. 필요한 소프트웨어 제품
      1. 운영체제: Ubuntu Server 20.04 LTS
      2. 웹 서버: Nginx 1.18.0
      3. 애플리케이션 서버: Apache Tomcat 9
      4. 데이터베이스: MySQL 8.0
      5. 백엔드 프레임워크: Spring Boot 2.5.0
      6. 프론트엔드 프레임워크: React 17.0.2
      7. 데이터 분석 도구: Python 3.8, pandas 1.2.4, NumPy 1.20.2
      8. 머신러닝 프레임워크: TensorFlow 2.4, Scikit-learn 0.24.2
   2. 소프트웨어 제품 상세 정보
      1. 이름: TensorFlow
      2. 버전: 2.4
      3. 출처: tensorflow.org
      4. 용도: LSTM 모델 구현 및 학습
5. 통신 인터페이스
   1. HTTP/HTTPS 프로토콜 사용
   2. RESTful API 구현 (Spring Boot)
   3. WebSocket 프로토콜 (실시간 데이터 업데이트용)
6. 기억 장치
   1. 특이사항 없음으로 인한 생략
7. 운영
   1. 사용자 조직에서의 운영모드: 웹 기반 대시보드를 통한 운영
   2. 자료 처리 자료 처리 지원 기능:

1. 시간 데이터 처리 (농넷 API를 통한 일별 데이터 수집)

2. 일일 배치 처리 (23:00에 다음날 예측 모델 실행)

3. 백업과 복구 기능: 일일 증분 백업 (01:00) 주간 전체 백업 (일요일 03:00)

1. 사이트 적용 요구사항
   1. 특이사항 없음으로 인한 생략

**‘2차 베이비부머 세대의 은퇴 후 저소득 구간 진입 시기 및 인구 증감 예측 서비스’**는 STC(팀 명)의 국가 복지 정보 시스템을 보완하기 위한 서비스 중 하나로 해당 서비스는 경제/복지 정책 입안자들이 더 정확한 예측 데이터를 바탕으로 정책을 수립할 수 있도록 지원하고자 고안되었다.

주요 특징

* 독립성: 본 시스템은 독립적으로 운영되지만, 필요시 기존 국가 복지 정보 시스템과 연동될 수 있다.
* 사용 데이터 : KOSIS, 한국노동연구원 등 다양한 외부 데이터 소스로부터 정보를 수집하고 통합하여 분석한다.
* 사용자 중심 설계: 경제/복지 정책 입안자들이 쉽게 이해하고 활용할 수 있는 인터페이스를 제공합니다.
* 확장성: 향후 추가적인 데이터 소스(지역별 노인 인구 데이터)를 통해 지역별 저소득 고령층 인구 예측 가능

시스템 구성

* 이 시스템은 웹 애플리케이션 서버, 데이터베이스 서버, 그리고 빅데이터 분석 서버 기능을 하나의 서버에서 처리하며 사용자는 웹 브라우저를 통해 시스템에 접근한다.

## **제품 기능**

1. 데이터 수집 및 전처리
   * REQ-FUNC-001: 농넷 API를 통한 일별 농산물 거래 데이터 자동 수집
   * REQ-FUNC-002: 기상청 API를 통한 시간별 기상 데이터 자동 수집
   * REQ-FUNC-003: 수집된 데이터의 자동 정제 및 표준화 처리

세부 요구사항:

* + 21개 농산물 품목에 대한 일별 거래량 및 가격 데이터 수집
  + 결측치 처리: 거래가 없는 날의 데이터는 전후 날짜의 평균값으로 대체
  + 이상치 탐지 및 처리: Z-score 방법을 사용하여 이상치 식별 후 처리

1. 기상 데이터 분석 및 온도 예측
   * REQ-FUNC-004: 수집된 기상 데이터를 활용한 단기 기온 예측 (1주, 2주, 4주)
   * REQ-FUNC-005: 장기 기온 추세 분석 및 예측 ( 1년 )

세부 요구사항:

* + ARIMA 모델을 사용한 단기 기온 예측
  + Prophet 모델을 사용한 장기 기온 추세 분석

1. 농산물 가격 예측
   * REQ-FUNC-006: LSTM 모델을 활용한 농산물 도매 가격 예측 (1주, 2주, 4주 후)
   * REQ-FUNC-007: 시계열 분해 기법(STL)을 활용한 농산물 가격의 계절성, 추세, 잔차 분석
   * REQ-FUNC-008: 예측 가격의 신뢰구간 계산 및 제공

세부 요구사항:

* + 21개 농산물 품목 각각에 대한 개별 LSTM 모델 구축
  + 입력 데이터: 과거 21일간의 가격, 거래량, 기상 데이터, 요일 정보
  + 출력 데이터: 1주, 2주, 4주 후의 예측 가격
  + 95% 신뢰구간 제공

1. 실시간 농산물 추천
   * REQ-FUNC-009: 현재 기온 데이터를 바탕으로 제철 농산물 추천
   * REQ-FUNC-010: 사용자 위치 기반 지역 특산물 추천

세부 요구사항:

* + 기온별 최적 농산물 매핑 테이블 구축
  + 사용자 IP 기반 위치 추정 및 해당 지역 특산물 정보 제공

1. 데이터 시각화 및 리포트 생성
   * REQ-FUNC-011: 예측 결과를 인터랙티브 그래프 및 차트로 시각화
   * REQ-FUNC-012: 맞춤형 PDF 리포트 생성 및 다운로드 기능

세부 요구사항:

* + D3.js를 활용한 인터랙티브 그래프 구현
  + 사용자 선택에 따른 맞춤형 PDF 리포트 자동 생성 (PyPDF2 라이브러리 사용)

1. 사용자 관리 및 알림 서비스
   * REQ-FUNC-013: 사용자 계정 관리 (등록, 수정, 삭제)
   * REQ-FUNC-014: 관심 농산물에 대한 가격 변동 알림 서비스

세부 요구사항:

* + Spring Security를 활용한 사용자 인증 및 권한 관리
  + WebSocket을 활용한 실시간 가격 변동 알림 구현

1. 데이터 관리 및 업데이트
   * REQ-FUNC-015: 일일 데이터 자동 업데이트 및 백업
   * REQ-FUNC-016: 데이터 이상치 탐지 및 처리 기능

세부 요구사항:

* + Cron job을 활용한 일일 자동 데이터 업데이트 (매일 00:05에 실행)
  + 증분 백업 (매일 01:00), 전체 백업 (매주 일요일 03:00)
  + Isolation Forest 알고리즘을 사용한 이상치 탐지

1. API 제공
   * REQ-FUNC-017: 외부 시스템 연동을 위한 RESTful API 제공

세부 요구사항:

* + JWT를 활용한 API 인증
  + Swagger를 활용한 API 문서 자동 생성

1. 성능 모니터링 및 개선
   * REQ-FUNC-018: 예측 모델의 성능 모니터링 및 자동 재학습

세부 요구사항:

* + MAE(Mean Absolute Error)를 기준으로 모델 성능 평가
  + 성능 저하 시 자동 알림 및 재학습 트리거

<표2. 요구사항 명세서1>

**※ *세부정보 < 4. 시스템 특징 > 참고.***

**※*비기능적 요구사항 < 5. 기타 비기능적 요구사항 > 참고.***

## **사용자 계층과 특성**

**2.3.1 사용자 그룹 정의 및 요구사항**

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **사용자 계층** | **특성** | 주요 업무 및 관심사 |
| U-1 | 농업 생산자 | - 농산물 재배 및 생산 담당 - 농작물 선택에 있어 계절을 주요여김  - 시장 가격 동향에 민감 | - 농산물 가격 예측  - 재배 계획 수립 |
| U-2 | 소비자 | - 농산물 구매자  - 가격 변동에 민감 | - 농산물 가격 동향 파악  - 제철 농산물 정보 |
| U-3 | 연구자 | - 농업 경제 연구 담당 | - 농산물 가격 결정 요인 분석 |

<표4.사용자 그룹 정의 및 요구사항1>

|  |  |  |
| --- | --- | --- |
| 입력정보 | 활동 | 기대산출물 |
| - 재배 품목  - 알고 싶은 기간 | 품목 별 수확 시기 예측  품목별 가격 예측 | 최적의 재배 시기 추천  적당한 가격대와 제철 농산물 추천 |

<표5.사용자 그룹 정의 및 요구사항2>

**2.3.2 사용자 우선 순위**

1. 최우선 순위:
   * 농업 생산자
   * 소비자
2. 차순위:
   * 연구자

## **운영 환경**





**2.4.1 백엔드**

* Java 17: 주요 애플리케이션 로직 개발
* Spring Boot: 웹 애플리케이션 프레임워크
* Spring Framework: 종합적인 Java 애플리케이션 개발 플랫폼
* Python: 데이터 분석 및 머신러닝 모델 개발
* MySQL: 주 데이터베이스 관리 시스템
* JWT (JSON Web Token): 사용자 인증 및 권한 관리
* Gradle: 프로젝트 빌드 및 의존성 관리 도구

**2.4.2 프론트엔드**

* React.js: 사용자 인터페이스 구축을 위한 JavaScript 라이브러리

**2.4.3 개발 도구**

* IntelliJ IDEA: Java 및 Spring 개발을 위한 통합 개발 환경
* Visual Studio Code: 프론트엔드 및 Python 개발을 위한 코드 에디터
* GitHub: 버전 관리 및 협업 플랫폼

**2.4.4 DevOps**

* Docker: 애플리케이션 컨테이너화 플랫폼

**2.4.5 API**

* DACON API: 외부 데이터 소스 연동 (농산물 가격 데이터 등)

**2.4.6 추가 구성 요소**

* 하드웨어:
  + 서버: 고성능 CPU (예: Intel Xeon), 128GB RAM 이상, SSD 스토리지
  + GPU: NVIDIA Tesla V100 또는 동급 (딥러닝 모델 학습용)
* 네트워크:
  + 고속 인터넷 연결 (최소 1Gbps)
  + 보안 방화벽
* 모니터링 및 로깅:
  + ELK Stack (Elasticsearch, Logstash, Kibana)
  + Prometheus + Grafana
* 추가 데이터베이스:
  + Redis: 캐싱 및 세션 관리
  + MongoDB: 비정형 데이터 저장 (필요시)
* CI/CD:
  + Jenkins: 지속적 통합 및 배포
* 클라우드 플랫폼 (선택적):
  + AWS, Azure, 또는 Google Cloud Platform
* 보안:
  + SSL/TLS 암호화
  + OAuth 2.0 (추가 인증 옵션)

## **설계 및 제약 사항**

### **2.5.1 기술 스택 제약**

1. 백엔드:
   * 언어: Java 11 이상
   * 프레임워크: Spring Boot 2.5.x
   * 빌드 도구: Maven 3.6.x
2. 프론트엔드:
   * 프레임워크: React 17.x
   * 상태 관리: Redux
   * 패키지 관리자: npm
3. 데이터베이스:
   * RDBMS: PostgreSQL 13.x
   * ORM: Hibernate 5.x
4. 머신러닝
   * Python

### **2.5.2 외부 시스템 연동**

1. KOSIS (국가통계포털) Open API 연동
   * 연동 목적: 통계 데이터 수집
   * 주요 수집 데이터: 연령 계층별 추계인구, 소득 분위, 고령자 관련 통계 등
   * 데이터 갱신 주기: 관공서 데이터 업데이트 주기에 맞춤
2. 고용노동부 Open API 연동
   * 연동 목적: 은퇴 전후의 소득 관련 데이터 수집
   * 주요 수집 데이터: 연령별 고용률, 임금 통계, 은퇴 관련 통계 등
   * 데이터 갱신 주기: 관공서 데이터 업데이트 주기에 맞춤

### **2.5.3 보안 및 규제 준수**

1. 개인정보보호법 준수
   * 주민등록번호 등 민감 정보 암호화 저장
   * 개인정보 수집 최소화 및 동의 절차 구현
2. 공공데이터의 제공 및 이용 활성화에 관한 법률 준수
   * 데이터 출처 명시
   * 2차 가공 데이터 제공 시 원본 데이터 참조 기능 구현

### **2.5.4 성능 요구사항**

1. 응답 시간:
   * 웹 페이지 로딩 3초 이내
   * 진입시기 예측 30초 이내
   * 전체 인구수 예측 1분 이내
2. 동시 접속자:
   * 최소 100명 이상 처리 가능

### **2.5.5 배포 및 운영 제약**

1. 컨테이너화: Docker 사용 필수
2. 오케스트레이션: Kubernetes 사용
3. CI/CD: Jenkins 파이프라인 구축

### **2.5.6 코드 품질 및 문서화**

1. 코드 컨벤션: Google Java Style Guide 준수
2. 문서화: Javadoc 사용 필수
3. API 문서: Swagger UI 통합

### **2.5.7 접근성 및 호환성**

1. 웹 표준: HTML5, CSS3 준수
2. 브라우저 호환성: 최신 2개 버전의 주요 브라우저 지원 (Chrome, Firefox, Safari, Edge)
3. 반응형 웹: 모바일 기기 지원 필수

## **사용자 문서**

2.6.1 **사용자 매뉴얼**:

 웹사이트 내 Help 섹션에 포함

 주요 내용:

* 시스템 개요 및 주요 기능 소개
* 농산물 가격 예측 조회 방법
* LSTM 모델과 시계열 분해 기법에 대한 간단한 설명
* 대시보드 사용법

2.6.2 **API 문서**:

 기술 설명 WebPag에 포함

 주요 내용:

* RESTful API 엔드포인트, 요청/응답 형식, 인증 방법 등 상세 설명
* 각 API의 사용법, 요청 및 응답 형식, 예제 코드, 인증 방법(JWT 활용) 등을 포함합니다.

2.6.3 **기술 문서**:

* **시스템 아키텍처**
  + 시스템 내부 구조 및, 분석 원리에 대한 기술적인 정보 제공
    - 시스템 아키텍처
    - 데이터베이스 스키마
    - LSTM 모델 및 시계열 분해(STL) 기법의 구현 방법
    - 운영 환경 설정 방법 등
* **데이터 해석 가이드**
  + 웹 기반 대시보드 및 분석 데이터 해석 방법 제공
  + 주요 내용:
    - 농산물 가격 추세 그래프 읽는 법
    - 예측 신뢰구간 해석 방법
    - 이상치 및 특이값 식별 방법

## **가정 및 종속성**

실제로 사용됨에 있어 PC 혹은 스마트폰 등 웹에 접속 가능한 모든 기계에서 사용되며 사용자가 원하는 데이터를

2.7.1 예측 분석에 대한 가정 및 종속성

* 1. 데이터 신뢰성
     + 시스템에 사용하는 data(도소매 , 농업 기상 data)등의 외부 API는 신뢰 가능하며, 정기적으로 업데이트된다.
     + 예측에 사용된 LSTM 모델& 시계열 분해 기법은 가격 예측에 효과적이다.
     + 전쟁과 같은 극단적 상황으로 인한 농작물 재배 감소가 없을 것
     + 예측 모델의 정확도가 70% 이상 유지될 것
  2. 가정에 따른 종속성:
     + 농넷 API: 일별 농산물 거래 데이터 수집
     + 기상청 API: 기상 데이터 수집.
     1. 프로그램 종속성:

1. 외부 API 종속성:
   * 농넷 API: 일별 농산물 거래 데이터 수집
   * 기상청 API: 기상 데이터 수집
   * DACON API: 추가 농산물 가격 데이터 연동
2. 소프트웨어 종속성:
   * Java 17 및 Spring Boot 환경
   * Python 3.8 이상 (데이터 분석 및 머신러닝용)
   * MySQL 8.0 데이터베이스 시스템
   * React.js 17.x (프론트엔드)
   * Docker 20.x (컨테이너화)
3. 하드웨어 종속성:
   * NVIDIA GPU (CUDA 지원): LSTM 모델 학습 가속화를 위해 필요
4. 라이브러리 종속성:
   * TensorFlow 2.x: LSTM 모델 구현
   * Pandas, NumPy: 데이터 처리
   * Statsmodels: 시계열 분해(STL) 구현
5. 네트워크 종속성:
   * 안정적인 인터넷 연결 (최소 10Mbps)
   * 보안 방화벽 및 로드 밸런서
6. 보안 종속성:
   * SSL/TLS 인증서: HTTPS 통신 보안
   * JWT 라이브러리: 사용자 인증
7. 클라우드 서비스 종속성 (선택적):
   * AWS, Azure, 또는 Google Cloud Platform 서비스
8. 오픈소스 라이브러리 종속성:
   * 지속적인 유지보수 및 보안 업데이트 필요

# **외부 인터페이스 요구사항** ※ **임의로 작성. 백엔드 개발 시 업데이트 예정**

**※** 기능적 요구사항

|  |  |  |  |
| --- | --- | --- | --- |
| **요구사항 고유번호** | RQ-ID-040-010 | | |
| **요구사항 명칭** | RQ-ID-040-010 | | |
| **요구사항 분류** | 기능 요구사항 | 응락수준 | 필수 |
| **요구**  **사항**  **상세**  **설명** |  | | |

<표2. 데이터 요구사항1>

## **사용자 인터페이스(User Interfaces)**

3.1.1 웹 기반 대시보드

* 기술: React.js를 이용한 SPA(Single Page Application)
* 주요 기능: a. 농산물 가격 예측 결과 시각화 (그래프 및 차트) b. 품목별, 기간별 데이터 필터링 c. 실시간 데이터 업데이트
* 반응형 디자인: 데스크톱, 태블릿, 모바일 지원
* 접근성: WCAG 2.1 지침 준수

3.1.2 관리자 콘솔

* Spring Boot 기반의 웹 애플리케이션
* 주요 기능: a. 사용자 관리 b. 시스템 로그 모니터링 c. 모델 성능 지표 확인 및 재학습 트리거

3.1.3 모바일 앱 (향후 개발 예정)

* React Native를 이용한 크로스 플랫폼 앱
* 주요 능: a. 푸시 알림을 통한 가격 변동 알림 b. 간편한 농산물 가격 조회

## **하드웨어 인터페이스(HardwareInterfaces)**

요구사항 없음

## **소프트웨어 인터페이스(Software Interfaces)**

3.3.1 운영체제

- 서버: Ubuntu Server 20.04 LTS

- 클라이언트: 크로스 플랫폼 (Windows, macOS, Linux)

3.3.2 데이터베이스

- MySQL 8.0

- 인터페이스: JDBC, JPA

3.3.3 외부 API

­1. 농넷 API

- 목적: 일별 농산물 거래 데이터 수집

- 프로토콜: RESTful API

- 데이터 형식: JSON

2. 기상청 API

- 목적: 기상 데이터 수집

- 프로토콜: RESTful API

- 데이터 형식: XML, JSON

3. DACON API

- 목적: 추가 농산물 가격 데이터 연동

- 프로토콜: RESTful API

- 데이터 형식: JSON

4. 머신러닝 프레임워크

- TensorFlow 2.x

- 인터페이스: Python API

5. 웹 서버

- Nginx 1.18.0

- Apache Tomcat 9.0.45

## **통신 인터페이스(Communications Interfaces)**

3.4.1 네트워크 프로토콜

* HTTP/HTTPS: 웹 서비스 및 API 통신
* WebSocket: 실시간 데이터 업데이트
* TCP/IP: 기본 네트워크 통신

3.4.2 데이터 교환 형식

* JSON: API 응답 및 요청
* Protocol Buffers: 대용량 데이터 직렬화 (선택적)

3.4.3. 보안 프로토콜

* 요구사항 없음

# **시스템 특징 (Use Cases)**

## **기상 데이터 분석 및 추수 시기 예측**

|  |  |
| --- | --- |
| **Brief Description:** | 기상청 API를 통해 수집한 데이터를 분석하고, 미래 온도를 예측하여 농산물 생산 및 추수 시기를 예측함. |
| **Business Trigger:** | 사용자(농업 생산자 혹은 제철 음식 정보를 원하는 소비자)가 시스템에 접근 |
| **Preconditions:** | - 기상청 API 데이터에 신뢰성이 필요.  - 최소 2년 이상의 data가 필요  - LSTM 모델이 학습되어 있음 |

|  |  |  |  |
| --- | --- | --- | --- |
| **Basic Flow:**  <그림2 Use Case Diagram>  1. 사용자가 가격 예측을 원하는 농산물과 기간을 선택.  2. 시스템은 선택된 농산물의 과거 데이터를 조회  3. STL 방법을 사용하여 시계열 데이터를 계절성, 추세, 가격 성분으로 분해  4. LSTM 모델은 분해된 데이터를 입력받아 미래 가격을 예측.  5. 예측 결과와 신뢰구간이 사용자에게 그래프로 표시  시나리오 재작성 필요.   1. **시작 (Start)**    * 사용자는 웹 홈페이지에 접속합니다. 2. **저소득층 진입 예상 서비스 페이지로 이동**    * 사용자는 저소득층 진입 예상 서비스 페이지로 이동합니다.    * 사용자는 예측 서비스를 선택합니다. 3. **서비스 선택**    * 사용자는 '은퇴 후 저소득 고령층 진입 시기 예측' 페이지나 '저소득 고령층 인구 증가 비율 예측' 페이지 중 하나를 선택합니다. 4. **세부 설정 선택**    * 선택된 페이지에서 사용자에게 세부 설정 옵션이 제공됩니다.    * 사용자는 설정을 선택하거나 선택하지 않을 수 있습니다.      + 선택 시: 체크박스 체크 단계로 이동.      + 선택하지 않을 시: 예측 결과와 데이터를 즉시 제공. 5. **체크박스 체크**    * 사용자가 선택한 세부 설정을 확인합니다.    * 확인된 경우, 예측 결과와 데이터 제공. 6. **결과 제공**    * 예측 결과 데이터가 사용자에게 제공됩니다.    * 서비스 종료 (End).  공통사항  * **Input Information**   + 나이   + 은퇴시기   + 은퇴 전후 소득 | | | |
| **Assumptions:** XXX*<the assumptions that are made for the basic flow, e.g. all items are for collection.>* | | | |
| **Line** | **System Actor Action** | | **System Response** |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | | 시스템은 예측 결과를 사용자에게 표시한다. |
| **Post Condition:** | | 예측 결과가 데이터베이스에 저장되고, 사용자에게 표시됨 | |

|  |
| --- |
| **Business Rules:** |
|  |

|  |
| --- |
| **Non Functional Requirements:** |
|  |

|  |
| --- |
| **Data Requirements:** |
|  |

|  |
| --- |
| **Activity Diagram:** |
| <그림3 User flow diagram> |

|  |
| --- |
| **Prototype Screen:** |
| *추후에 이미지 추가* |

|  |  |  |
| --- | --- | --- |
| **Screen Entry Exception Table:** *추후에 이미지 추가* | | |
| **Field** | **Constraint** | **Response** |
| 1. e.g. Customer Surname | 30 chars max | Message: “Exceeds 30 chars max – please re-enter” |
|  |  |  |

## **제철 농산물의 가격 예측**

|  |  |
| --- | --- |
| **Brief Description:** | 농산물의 가격 변화를 예측 하여 사용자에게 제공하는 시스템 |
| **Business Trigger:** | 사용자(판매를 원하는 생산자와 농산물 가격 정보를 원하는 소비자)가 시스템에 접근 |
| **Preconditions:** | - 현재 기상 데이터가 시스템에 입력되어 있음  - 최신 농산물 가격과 예측 값이 5%이내여야함. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Basic Flow:**  <그림2 Use Case Diagram>  1. 사용자가 추천 서비스에 접속  2. 시스템은 현재 기상 조건, 계절, 농산물 가격 예측 결과를 분석함  3 .분석 결과를 바탕으로 현재 조건에 가장 적합한 제철 농산물 리스트를 생성  4. 추천 리스트는 가격, 품질, 수급 상황 등을 고려하여 우선순위 정렬.  5. 사용자에게 추천 농산물 리스트가 표시되며, 각 항목에 대한 상세 정보를 제공 | | | |
| **Assumptions:** XXX*<the assumptions that are made for the basic flow, e.g. all items are for collection.>* | | | |
| **Line** | **System Actor Action** | | **System Response** |
|  |  | |  |
|  |  | |  |
|  |  | |  |
|  |  | | 시스템은 예측 결과를 사용자에게 표시한다. |
| **Post Condition:** | | 예측 결과가 데이터베이스에 저장되고, 사용자에게 표시됨 | |

|  |
| --- |
| **Business Rules:** |
|  |

|  |
| --- |
| **Non Functional Requirements:** |
|  |

|  |
| --- |
| **Data Requirements:** |
|  |

|  |
| --- |
| **Activity Diagram:** |
| <그림3 User flow diagram> |

|  |
| --- |
| **Prototype Screen:** |
| *추후에 이미지 추가* |

|  |  |  |
| --- | --- | --- |
| **Screen Entry Exception Table:** *추후에 이미지 추가* | | |
| **Field** | **Constraint** | **Response** |
| 1. e.g. Customer Surname | 30 chars max | Message: “Exceeds 30 chars max – please re-enter” |
|  |  |  |

# **기타 비기능적 요구사항**

## **성능요건**

* 백엔드 개발시 변경 사항 수정 및 추가 예정
* 웹 페이지 로딩 시간은 3초 이내여야 한다.
* 데이터 분석은 30초 이내여야 한다.
* 데이터베이스 쿼리 응답 시간은 1초 이내여야 한다.
* 시스템은 피크 시간대에 초당 100개의 트랜잭션을 처리할 수 있어야 한다.

## **안전 요구 사항**

* 시스템은 사용자의 개인정보를 안전하게 보호해야 한다.
* 예측 결과는 오용될 가능성에 대한 경고와 함께 제공되어야 한다.

※ 예측결과가 틀릴 가능성에 대한 경고 포함.

## **보안 요구 사항***.*

* 백엔드 개발 시 삭제 또는 수정 예정
* 모든 데이터 전송은 HTTPS를 통해 암호화되어야 한다.
* 사용자 인증은 JWT(JSON Web Token)를 사용해야 한다.
* 비밀번호는 bcrypt 등의 안전한 해시 알고리즘을 사용하여 저장해야 한다.

## **Software Quality Attributes**

* 백엔드 개발 시 삭제 또는 수정 예정
* 가용성: 시스템은 연간 99% 이상의 가동 시간을 유지해야 한다.
* 확장성: 다른 프로젝트에 본 프로젝트의 데이터가 사용되더라도 성능 저하 없이 작동해야 한다.
* 유지보수성: 코드는 명확하게 문서화되어야 하며, 모듈화된 구조를 가져야 한다.
* 사용성: 초보 사용자도 5분 이내에 주요 기능을 이해하고 사용할 수 있어야 한다.