

요구 사항 분석서

<딸깍 리뷰>

2025-2

TEAM1

이의빈 ***202114228***

이유환 ***202111343***

최정길 ***202110542***

Table of contents

1.Introduction	03
1.1 Purpose(프로젝트 기획 배경)	03
1.2 Technology Trends&Core Technologies(기술 동향 및 핵심 기술)	03
1.3 Key Points(프로젝트 주요 기능 및 특징)	04
1.4 Team Member	04
1.5 Schedule	04
2. Functional Requirements	05
2.1 Use Case Diagram	05
2.1.1 Actor	05
2.1.2 Actor와 Use Case 간의 관계	05
2.1.1.1 User	05
2.1.1.2 User_Registered	05
2.1.1.3 AI	05
2.2 Use Case Document	06
3. Non-functional Requirements	09
3.1 Usability(사용편리성)	09
3.1.1 사용자의 수준	09
3.1.2 UI/UX	09
3.1.3 사용자에게 제공될 문서	11
3.2 Reliability(신뢰성)	14
3.2.1 Robustness, safety, security	14
3.2.2 문제 발생시 재시동의 용인 수준	14
3.2.3 예외 사항에 대한 조치	14
3.2.4 보안 및 안전 차원의 특기 사항	14
3.3 Performance(성능)	14
3.3.1 반응성 수준	14
3.3.2 지연 수준	14
3.3.3 Time Critical Tasks	15
3.3.4 저장할 데이터의 요구 용량	15
3.4 Supportability, maintainability, portability(이식성)	15
3.4.1 향후 확장성을 위한 고려사항	15
3.4.2 관리자 요구사항	16
3.4.3 향후 포팅을 고려한요구사항	16
3.5 Maintenance(유지관리)	16
3.5.1 시스템 운영 책임 소재	16
3.5.2 유지관리를 위한 조직 요구사항	16
3.6 Implementation(구현상 제약사항)	16
3.6.1 하드웨어 요구사항	16
3.6.2 테스트(Testing)	17
3.7 Interface(인터페이스)	17
3.7.1 타 시스템과의 연동	17
3.7.2 데이터의 호환	17
3.7.3 표준 인터페이스 사용 여부	17
3.8 Legal(법적 제약사항)	17
3.8.1 라이선스	17
3.8.2 라이선스 특이사항	17
3.8.3 시스템 오작동시의 법적 책임소재	18
3.8.4 비용	18

1.Introduction

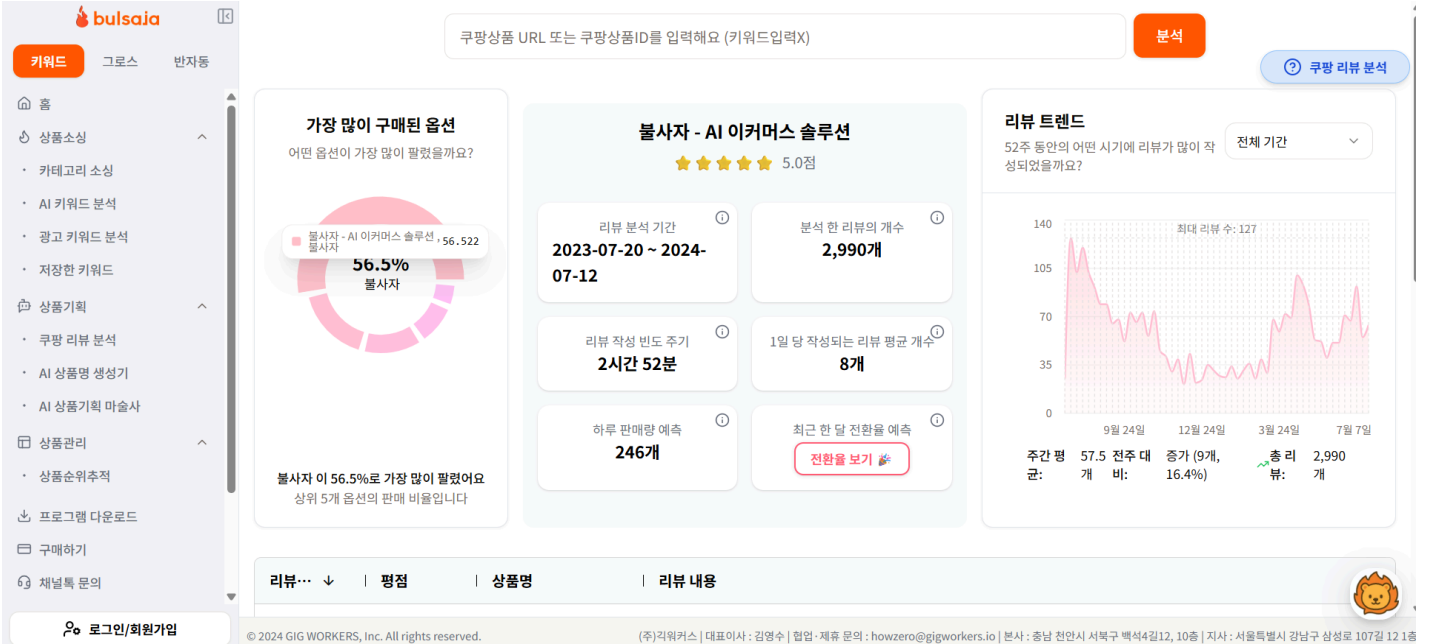
1.1 Purpose(프로젝트 기획 배경)

높은 평점과 추천 리뷰 등 쇼핑물에 노출되는 평가는 주로 긍정적이며, 이를 보고 구매를 하여도 해당 제품이 기대했던 것과 달라서 좋지 않은 구매 경험을 하는 경우가 있다. 또한 많은 수의 리뷰들 중에서 사용자가 중요하게 여기는 정보에 대한 균형있는 접근이 어렵다.

1.2 Technology Trends & Core Technologies(기술 동향 및 핵심 기술)

최근 몇 년동안 온라인 쇼핑이 급격하게 성장함에 따라 소비자의 리뷰 데이터가 방대하게 축적되고 있다. 이에 따라 수동으로 리뷰를 분석하고 판단하는 비효율성을 해소하기 위해 자동화된 분석의 필요와 수요가 증가하고 있고, 사이트 자체의 리뷰 요약 서비스가 등장하고 있다.
현존하는 사이트 자체의 리뷰 요약 서비스는 사이트 자체 기준으로 리뷰를 분석하며, 전체 리뷰 사이즈에 비하여 그 내용이 매우 짧다. 이는 사용자 개인의 주요 관점을 고려하지 않아 필요한 내용이 없을 수 있고, 요약된 내용에 대한 신뢰성이 떨어진다.
이 프로그램은 사용자의 개인의 주요 관점을 고려하고, 실제 리뷰 데이터와 분석 및 요약된 내용에 대한 연관도를 검사하여 높은 신뢰성을 준다.

유사 서비스: *bulsaja*



확장 프로그램 필요

쿠팡 리뷰 분석을 위해 Chrome 확장 프로그램이 필요합니다. 불사자 확장 프로그램을 설치해주세요.

취소설치하러 가기

쿠팡 리뷰를 분석하는 프로그램이지만, 키워드 입력을 허용하지 않는다. 우리 서비스는 *URL*과 *Keyword*를 기반으로 유저가 원하는 리뷰만을 제공하는 서비스를 목표로 한다. 또한 별도의 확장 프로그램을 설치하지도 않는다.

구분	기술명	역할
데이터 수집	웹 스크래핑, 웹 크롤링(<i>Python</i>)	쇼핑몰에서 리뷰 데이터를 자동으로 수집
데이터 처리	데이터 전처리기	리뷰 텍스트 정제 및 분석 전 가공
분석 및 요약	<i>LLM</i> 활용 분석 및 요약	핵심 키워드를 중심으로 리뷰의 경향에 따라 분류하여 분석 및 요약

1.3 Key Points(프로젝트 주요 기능 및 특징)

1. 리뷰 데이터 자동 수집
- 사용자가 입력한 쇼핑몰 링크를 통해 리뷰 데이터를 자동 수집
2. 리뷰 텍스트 전처리
- 수집된 데이터에서 필요한 내용만 추출

- 특수문자나 불필요한 공백 등의 제거를 통해 입력 토큰수 감량
3. 리뷰 분석 및 요약
- LLM을 통해 사용자가 입력한 핵심 키워드를 중심으로 리뷰 분석 및 요약

- 사용자가 요청한 관점을 바탕으로 양면의 정보를 요약 후 제공하여 사용자의 객관적인 판단을 도움

1.4 Team Member

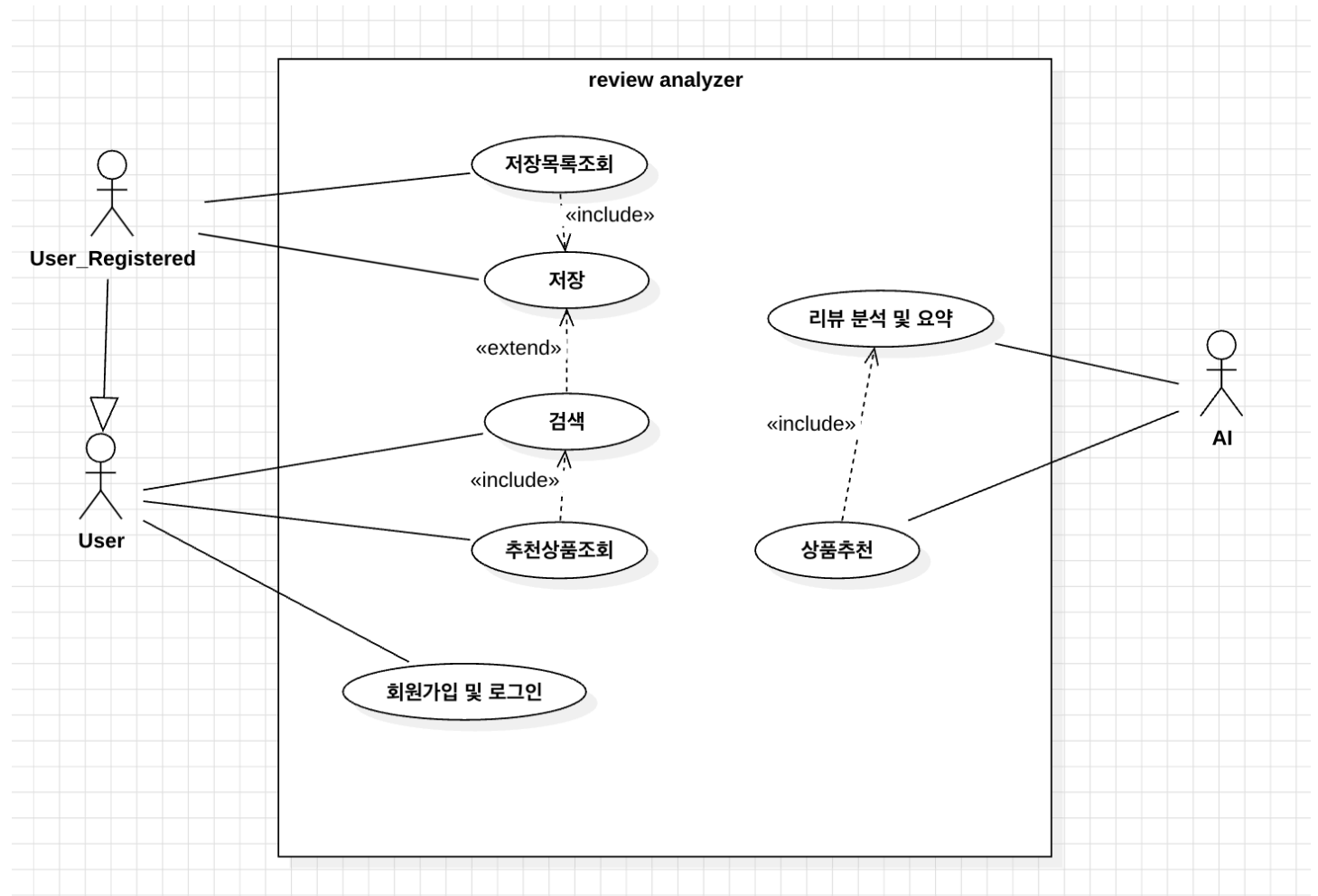
이름	담당 역할
이의빈	데이터 전처리, AI분석 모델
이유환	웹 크롤링, 웹 스크래핑, 입력 처리
최정길	결과 출력/저장, 서버 파트

1.5 Schedule

구분	기간	내용
1단계: 기획 및 구현 가능성 파악	9/14 ~ 9/27	1. 주제 확정 2. 역할 분담 3. 구현 가능성 파악
2단계: 요구 사항 정의 및 기초 구현	9/28 ~ 10/11	1. 요구 사항 분석서 작성 2. 웹 크롤링 및 AI분석 모델, 서버 기초 구현 3. 기초 UI/UX 디자인 및 구현
3단계: 분석 결과 저장 및 출력, 추천 기능	10/12 ~ 10/25	1. 분석 결과 저장 및 조회, 출력 기능 구현 2. 추천 기능 구현
4단계: 1차 통합 및 로그인	10/26 ~ 11/8	1. 회원가입 및 로그인 기능 구현 2. 웹 UI와 구현된 기능 통합
5단계: 최종 통합 및 테스트	11/9 ~ 11/22	1. 전체 기능 통합 2. 서버 테스트 3. UI 완성 4. 전체 프로그램 테스트 및 디버깅
6단계: 최종 검수	11/23 ~ 12/6	1. 전체 프로그램 테스트 및 디버깅

2. Functional Requirements

2.1 Use Case Diagram



2.1.1 Actor

Actor 는 (User, User_Registered, AI) 가 존재합니다. User는 로그인을 하지 않은 사용자를 의미합니다. User_Registered는 로그인을 한 사용자를 의미합니다. AI는 리뷰를 분석하는 AI 모델을 의미합니다.

2.1.2 Actor와 Use Case 간의 관계

2.1.2.1 User

User는 쿠팡 상품 URL을 통해 검색을 할 수 있습니다.

User는 추천 상품 조회를 하기 전에 반드시 검색을 수행해야 합니다. 추천상품조회 -<<include>>->검색.

2.1.2.2 User_Registered

User_Registered는 User의 부분집합입니다.

User_Registered는 User는 사용할 수 없는 저장과 저장목록조회를 사용할 수 있습니다.

저장목록조회를 하기 전에 반드시 저장을 수행해야 합니다. 저장목록조회 -<<include>>->저장.

검색을 수행한 후 선택에 따라 저장을 수행할 수 있습니다. 검색-<<exclude>>->저장.

2.1.2.3 AI

AI는 리뷰 분석 및 요약과 상품추천을 할 수 있습니다.

상품추천을 하기 전에 반드시 리뷰 분석 및 요약을 수행해야 합니다. 상품추천-<<include>>->리뷰분석및요약.

2.2 Use Case Document

Name	검색
Description	사용자가 키워드 중심 분석을 원하는 상품에 대한 검색을 수행
Actors	User
Pre-conditions	사용자가 유효한 상품 URL 입력
Post-conditions	검색 결과를 출력

Main Flow	
Actor Actions	System Responses
1. 상품의 URL 및 키워드 입력	
	1.1. 입력된 URL에 접속 시도 및 접속
	1.2. 리뷰 페이지에서 리뷰 데이터 크롤링 및 스크래핑
	1.3. 스크래핑한 내용을 파일로 저장
2. 분석 결과 확인	

Exception Flow	
Actor Actions	System Responses
1. 잘못된 입력	
	1.1.a. 접속 실패 시 오류 메시지 출력
2. 오류 메시지 확인	

Name	추천상품조회
Description	사용자가 요청 시 검색한 제품과 유사한 제품을 추천
Actors	User
Pre-conditions	1. 사용자의 검색 및 분석 결과 출력 완료 2. 사용자가 검색한 상품과 유사한 상품 탐색 및 분석, 요약 완료
Post-conditions	추천상품 데이터를 개인 라이브러리에 저장 가능

Main Flow	
Actor Actions	System Responses
1. 사용자가 추천상품조회 요청	
	1.1. 유사한 상품의 분석된 결과를 출력
	1.2. 사용자 유형 확인 및 저장 활성화

2. 추천상품조회	
-----------	--

<i>Name</i>	회원가입 및 로그인
<i>Description</i>	검색 결과의 저장을 위한 회원가입과 로그인
<i>Actors</i>	User
<i>Pre-conditions</i>	없음
<i>Post-conditions</i>	회원정보 데이터베이스에 해당 회원 등록

<i>Main Flow</i>	
<i>Actor Actions</i>	<i>System Responses</i>
1. 회원가입 요청	
2. 아이디 입력	
	2.1. 회원정보 데이터베이스에서 해당 회원 정보 조회
	2.2. 회원가입 가능 여부 검증
	2.3. 해당 회원정보를 데이터베이스에 저장
3. 회원가입 성공	

<i>Exception Flow</i>	
<i>Actor Actions</i>	<i>System Responses</i>
	2.2.a. 해당 회원 정보가 데이터베이스에 이미 존재하면 오류 메시지 출력 및 로그인 실패

<i>Main Flow</i>	
<i>Actor Actions</i>	<i>System Responses</i>
1. 로그인 요청	
	1.1. 회원정보 데이터베이스에서 회원 정보 조회
	1.2. 로그인 가능 여부 검증(승인)
2. 로그인 성공	

<i>Exception Flow</i>	
<i>Actor Actions</i>	<i>System Responses</i>
	1.2.a 로그인 가능 여부 검증(거절)

<i>Name</i>	저장
<i>Description</i>	사용자가 검색한 결과를 저장
<i>Actors</i>	User_Registered
<i>Pre-conditions</i>	1. 로그인한 사용자 2. 사용자가 검색한 결과가 존재
<i>Post-conditions</i>	사용자의 개인 라이브러리에 검색한 결과가 저장

<i>Main Flow</i>	
<i>Actor Actions</i>	<i>System Responses</i>
1. 사용자가 검색한 결과 저장 요청	
	1.1. 사용자의 개인 라이브러리에 검색 결과 저장

<i>Name</i>	저장목록조회
<i>Description</i>	사용자가 이전에 저장한 검색 결과를 조회
<i>Actors</i>	User_Registered
<i>Pre-conditions</i>	1. 로그인한 사용자 2. 사용자가 저장한 검색 결과가 존재
<i>Post-conditions</i>	조회 요청한 데이터를 출력

<i>Main Flow</i>	
<i>Actor Actions</i>	<i>System Responses</i>
1. 사용자가 저장 목록 조회 요청	
	1.1. 저장 목록 표시
2. 저장 목록 조회	
3. 저장 목록의 특정 상품 선택	
	3.1. 사용자가 선택한 상품의 저장된 데이터 출력
4. 상품의 분석 결과 데이터 조회	

<i>Name</i>	리뷰 분석 및 요약
<i>Description</i>	크롤링 된 리뷰 데이터를 분석 및 정리하여 요약
<i>Actors</i>	AI
<i>Pre-conditions</i>	1. 크롤링된 데이터 존재 2. 크롤링된 데이터 조회 및 가공
<i>Post-conditions</i>	요약된 데이터를 저장

<i>Main Flow</i>	
<i>Actor Actions</i>	<i>System Responses</i>
	1. 크롤링 및 가공된 데이터 제공
1. 가공된 데이터를 분석하여 요약 및 정리	
	2.1. 요약된 데이터 저장

<i>Name</i>	상품추천
<i>Description</i>	사용자가 검색한 상품과 유사한 상품들 중 사용자의 기준에 적합한 제품을 선별 및 추천
<i>Actors</i>	AI
<i>Pre-conditions</i>	1. 사용자의 상품 검색 2. 사용자가 검색한 상품과 유사한 제품의 크롤링된 리뷰 데이터 존재
<i>Post-conditions</i>	사용자가 저장하지 않은 상품의 데이터 삭제

<i>Main Flow</i>	
<i>Actor Actions</i>	<i>System Responses</i>
1. 유사 상품들의 리뷰 데이터 분석 및 요약	
2. 사용자 키워드 중심의 상품 선별	
	2.1. 선별된 상품들의 분석 내용 임시 저장

3. Non-functional Requirements

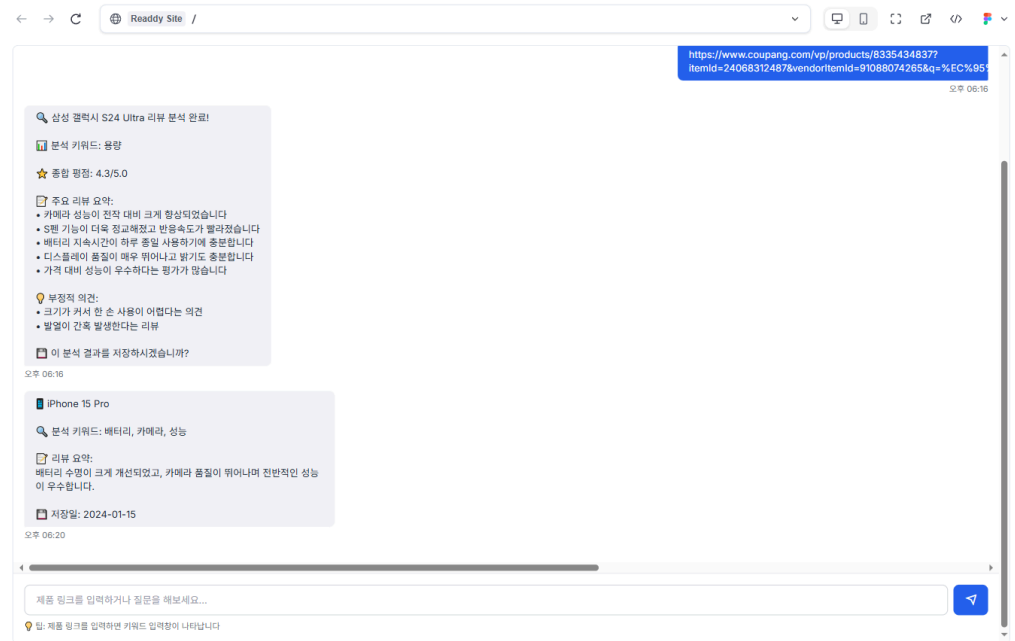
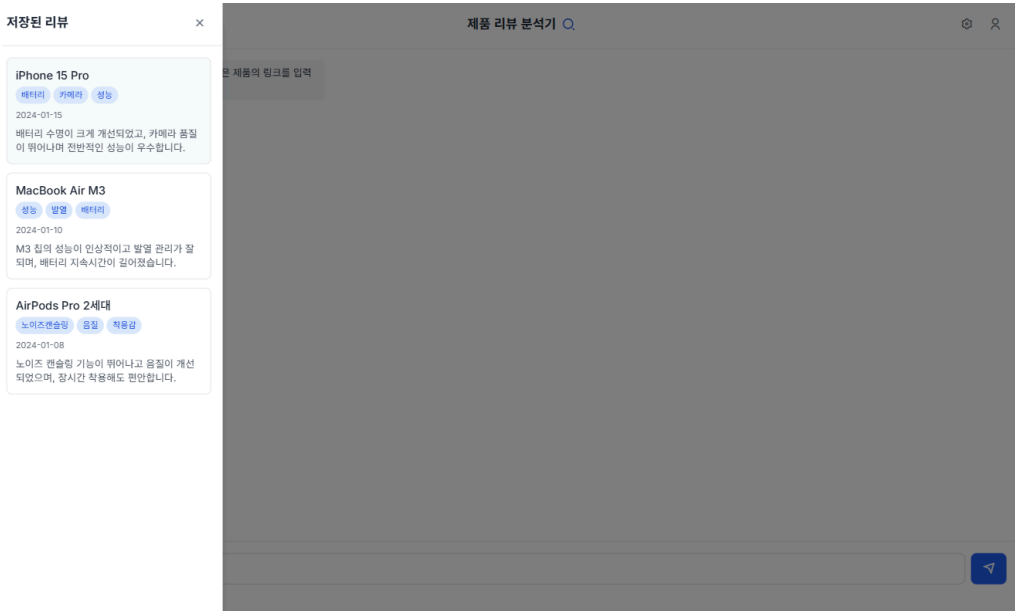
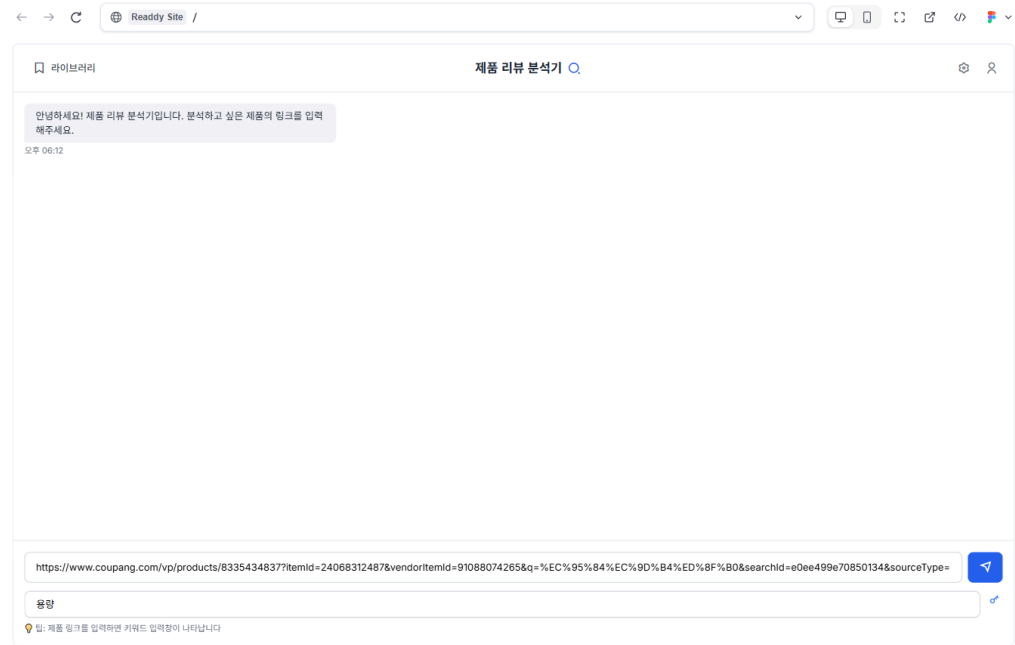
3.1 Usability(사용편리성)

3.1.1 사용자의 수준

쿠광에서 상품 리뷰를 확인하는 일반 소비자를 대상으로 합니다. 따라서 IT 기술에 대한 특별한 지식이 없어도 직관적으로 서비스를 이해하고 사용할 수 있어야 합니다.

3.1.2 UI/UX

최소한의 입력(제품 URL, 키워드)으로 원하는 결과를 얻을 수 있는 단순한 흐름을 제공합니다. 분석 대기 시간에는 로딩 상태를 명확히 표시하여 사용자가 시스템의 현재 상태를 인지할 수 있도록 합니다. 분석 결과는 키워드별로 명확하게 구분하고, 긍/부정 감성 분포와 같은 정보를 시각적으로 표현하여 가독성을 높입니다.



제품 **URL**을 입력하는 최소한의 행동으로 리뷰 분석을 시작할 수 있습니다. 이후 대화형 인터페이스를 통해 원하는 키워드를 전달하면, 해당 키워드를 중심으로 정리된 긍정 및 부정 요약과 종합 평점을 한눈에 제공하여 직관적인 정보 확인이 가능합니다. 또한, 분석했던 제품의 결과는 '라이브러리'에 저장하여 필요할 때마다 손쉽게 다시 찾아볼 수 있는 편의성을 갖추었습니다.

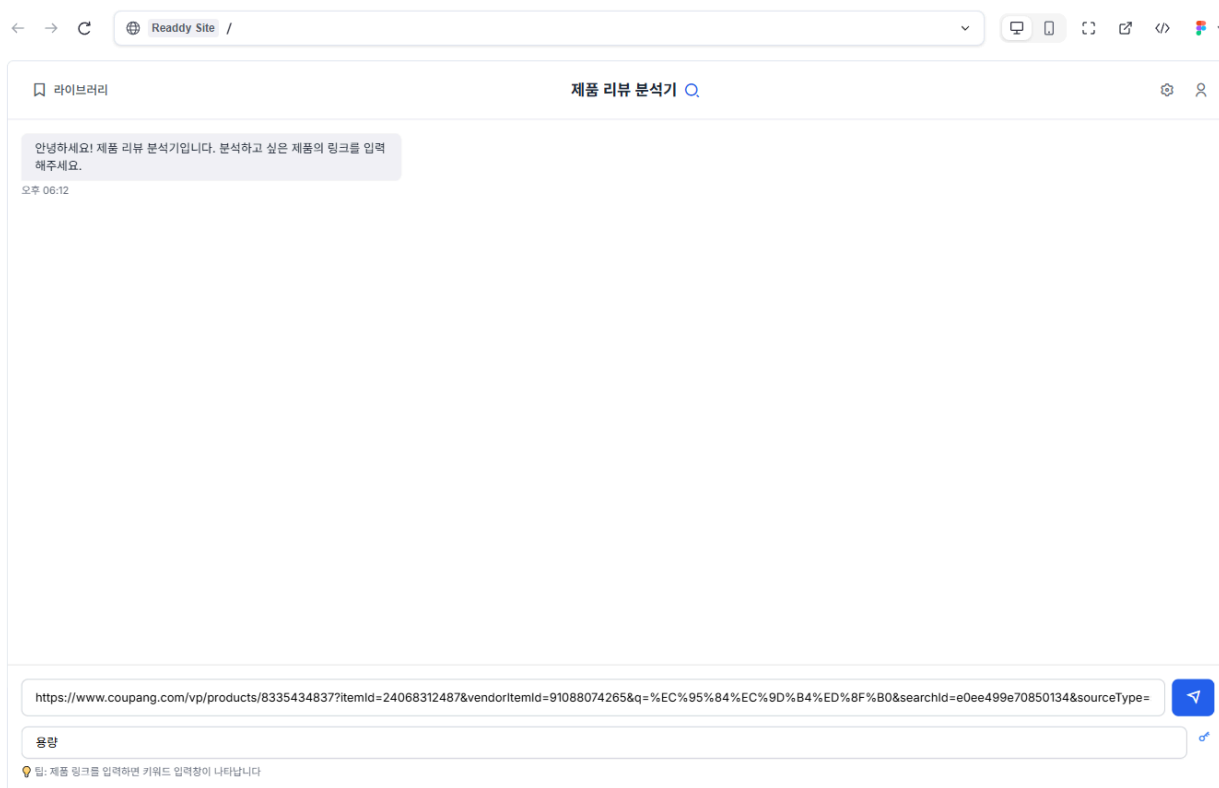
3.1.3 사용자에게 제공될 문서

제품 리뷰 분석 서비스 간단 사용 설명서

수많은 리뷰를 직접 읽지 않고, 원하는 정보만 쏙쏙 골라 확인하세요! 저희 리뷰 분석 서비스가 여러분의 현명한 쇼핑을 도와드립니다.

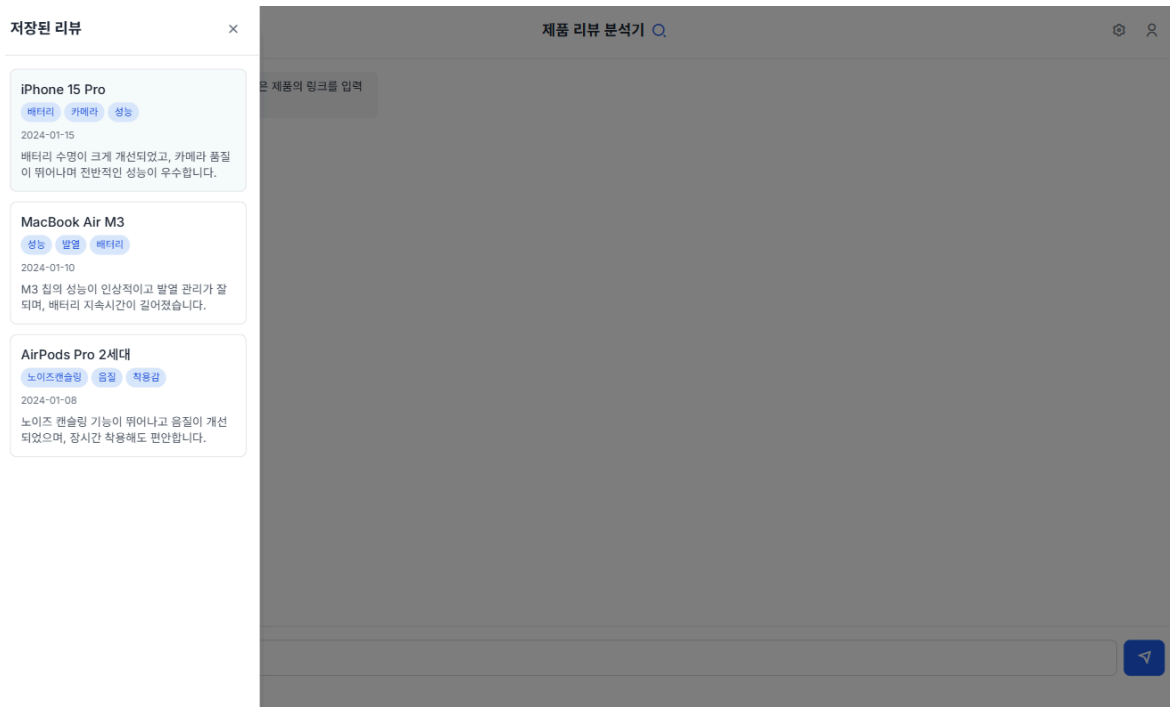
1단계: 제품 링크 입력하기

- 방법: 분석하고 싶은 제품의 웹사이트 주소(**URL**)를 복사한 뒤, 메인 화면의 입력창에 붙여넣고, 키워드를 입력하여 전송 버튼을 눌러주세요



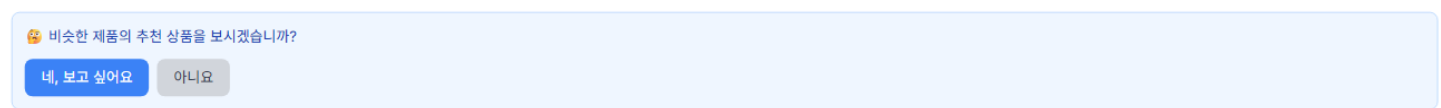
2단계: 분석 결과 한눈에 확인하기

- 방법: 입력하신 키워드를 중심으로 긍정적인 점과 부정적인 점을 요약해 드립니다. 종합 평점도 함께 제공되어 제품의 장단점을 쉽고 빠르게 파악할 수 있습니다.

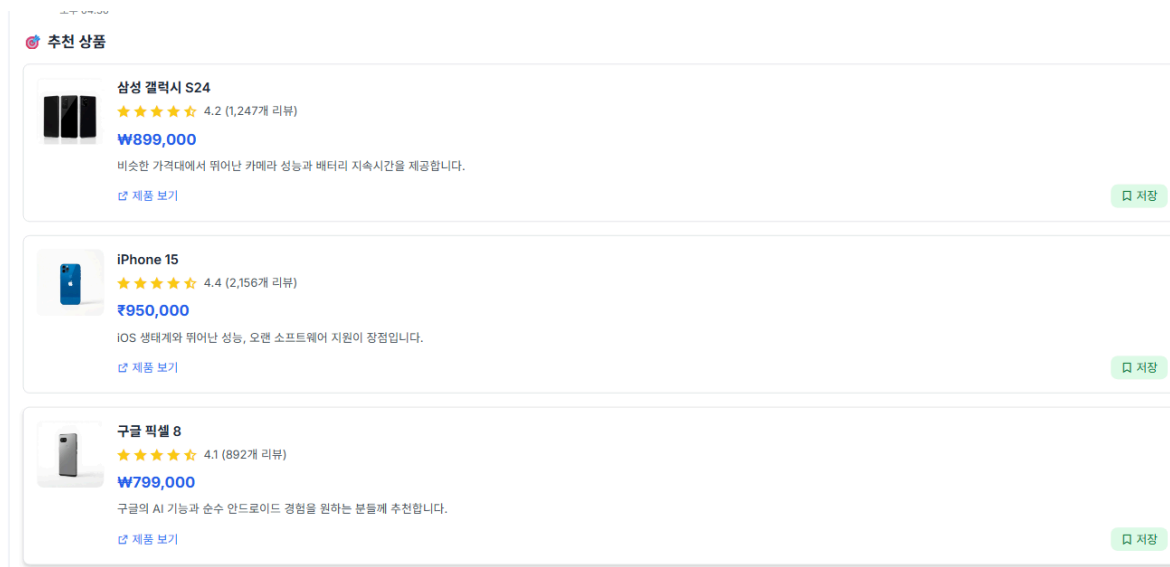


추가 기능: 유사 상품 추천 받기

- 설명: 하나의 제품 분석이 완료된 후, 사용자에게 유사한 제품을 추천받을지 선택할 수 있는 옵션을 제공합니다.



사용자가 '예'를 선택하면, 현재 분석한 상품과 가격대나 핵심 기능이 비슷한 대체 상품 목록을 보여줍니다.



추천된 각 상품은 간단한 요약 정보와 함께 제공되며, 사용자는 이 상품들 또한 '라이브러리'에 저장하여 나중에 다시 확인하거나 기존 상품과 비교해볼 수 있습니다.

3.2 Reliability(신뢰성)

3.2.1 Robustness, safety, security

사용자가 유효하지 않은 URL을 입력하거나 지원하지 않는 쇼핑몰 링크를 입력할 경우, 시스템이 중단되지 않고 적절한 오류 메시지를 사용자에게 안내해야 합니다. 웹 크롤링 시 특정 사이트로부터 차단(block)되지 않도록 안정적인 크롤링 정책을 수립해야 합니다.

3.2.2 문제 발생시 재시동의 용인 수준

개별 사용자의 요청(크롤링, AI 분석) 처리 중 오류가 발생하더라도 전체 서비스가 중단되어서는 안 됩니다. 해당 오류는 사용자에게 즉시 알려져야 하며, 다른 사용자는 정상적으로 서비스를 이용할 수 있어야 합니다. 서버 전체의 재시작은 서버 관리자의 판단하에 수동으로 진행하는 것을 원칙으로 합니다.

3.2.3 예외 사항에 대한 조치

크롤링 실패: 대상 웹사이트의 구조가 변경되었거나, 사이트가 일시적으로 불안정할 경우 "리뷰를 가져올 수 없습니다."와 같은 메시지를 출력합니다.

AI 분석 실패: AI 모델의 응답이 없거나 분석에 부적합한 결과가 반환될 경우, "리뷰 요약에 실패했습니다. 다시 시도해 주세요."와 같은 메시지를 안내합니다.

DB 조회 실패: 데이터베이스에 저장된 결과 조회 시 문제가 발생하면, 신규 분석을 시도하거나 시스템 점검 메시지를 출력합니다.

3.2.4 보안 및 안전 차원의 특기 사항

사용자의 입력값(URL, 키워드)에 대한 기본적인 입력값 검증(Input Validation)을 수행하여 SQL Injection과 같은 웹 공격을 방지한다.. 개인 라이브러리 기능 구현 시, 사용자 정보가 외부로 노출되지 않도록 서버 보안을 유지합니다.

3.3 Performance(성능)

본 시스템은 사용자에게 쾌적한 경험을 제공하기 위해 아래의 성능 기준을 준수해야 합니다.

3.3.1 응답 시간 (Response Time)

시스템의 각 기능은 명시된 시간 내에 처리를 완료하고 사용자에게 결과를 제시하는 것을 원칙으로 합니다.

- 리뷰 스크래핑
 - Delay 시간: 0~10 초 사이의 random한 지연 시간을 부여합니다.
 - 리뷰 선별: 본문이 없는 리뷰는 수집하지 않아야 합니다. 따라서 리뷰파일(reviews.xlsx)에 존재해서는 안 됩니다.
- 신규 리뷰 분석 요청
 - 정상 처리 기준: 일반적인 네트워크 환경에서 '분석'을 요청한 시점부터 요약 결과를 화면에 표시하기까지의 총 처리 시간은 90초를 초과해서는 안 됩니다.
 - 성공률 목표: 전체 신규 분석 요청 중 95% 이상이 위 기준 시간 내에 성공적으로 완료되어야 합니다.
 - 단계별 요구 시간:
 - 1단계 (데이터 수집 및 분류): 최대 60초
 - 2단계 (AI 요약 및 결과 제공): 최대 30초
 - 진행 상태 알림: 분석 시작 시, 사용자에게 로딩 애니메이션 등 시각적 피드백을 제공하여 시스템이 정상 동작 중임을 명확히 인지시켜야 합니다.
- 라이브러리 조회 요청
 - 사용자가 저장된 분석 결과를 조회할 경우, 요청 시점으로부터 3초 이내에 화면에 결과를 표시해야 합니다.

3.3.2 처리 실패 및 지연 정책 (Timeout & Delay Policy)

- 시간 초과 (Timeout) 처리: 신규 리뷰 분석 요청의 전체 처리 시간이 90초를 초과할 경우, 시스템은 해당 작업을 실패로 간주하고 즉시 중단해야 한다.
- 사용자 알림: 시간 초과로 작업이 실패하면, 사용자에게 "분석 시간이 초과되었습니다. 잠시 후 다시 시도해 주세요."와 같은 명확한 안내 메시지를 표시해야 한다.

3.3.3 데이터 저장 요구 용량 (Data Storage Requirement)

- 저장 데이터 항목: 각 분석 결과에 대해 아래 항목을 데이터베이스(MySQL)에 저장한다.
 - 사용자 식별자, 제품 URL, 요청 키워드, AI 요약 결과 원본, 긍/부정 분류 결과, 종합 평점, 생성 일시 등
- 초기 용량 할당: 서비스 초기 운영을 위해 최소 10GB의 저장 공간을 확보해야 한다.
- 확장성 요구사항: 데이터베이스의 전체 용량 사용률이 80%에 도달하면 시스템 관리자에게 자동으로 경고(Alert)를 보내야 한다. 이는 향후 서비스 확장 및 마이그레이션을 위한 사전 조치이다.

3.3.4 키워드-리뷰 요약 연관성 (Keyword-Summary Relevance)

AI가 생성한 요약 결과는 사용자가 입력한 키워드와 명확한 연관성을 가져야 한다. 연관성 판별 기준은 아래 표를 따르며, 시스템은 전체 요약 문장 중 '높음' 또는 '중간' 등급의 비율이 85% 이상이 되도록 보장해야 한다.

[키워드 연관성 판별 기준표]

등급	기준	예시 (키워드: "배터리")
높음 (High)	키워드가 요약 문장의 핵심 주제이며, 키워드에 대한 구체적인 장점 또는 단점을 직접적으로 설명함.	"배터리 지속 시간이 매우 길어 하루 종일 사용해도 충분하다는 의견이 많습니다."
중간 (Medium)	키워드가 문장에 포함되어 있으며, 관련된 부가 정보나 상황을 함께 언급함.	"영상을 오래 볼 경우 배터리가 빠르게 소모된다는 일부 의견이 있었습니다."
낮음 (Low)	키워드가 단순히 언급만 되거나, 직접적인 맥락 없이 다른 내용에 섞여 있음.	"이 제품은 뛰어난 카메라와 오래가는 배터리를 탑재했습니다."
무관 (Irrelevant)	키워드가 포함되어 있지 않거나, 전혀 다른 주제의 내용을 요약함.	"디스플레이의 선명도와 디자인에 대한 만족도가 높습니다."

3.4 Supportability, Maintainability, Portability (지원성, 유지보수성, 이식성)

3.4.1 향후 확장성을 위한 고려사항

현재 시스템은 서버 주도형 데이터 수집 방식으로, 안정성 검증을 위해 단일 사용자 환경을 기준으로 설계되었습니다. 향후 다수의 동시 접속자를 안정적으로 지원하기 위해, 데이터 수집 방식을 클라이언트 사이드(Client-Side) 모델로 전환하는 확장을 고려해야 합니다.

- 클라이언트 모듈 도입: 사용자가 자신의 PC에서 직접 리뷰 데이터를 스크래핑할 수 있도록 브라우저 확장 프로그램 또는 경량 데스크톱 애플리케이션 형태의 클라이언트 모듈을 개발해야 합니다. 이를 통해 데이터 수집 부하를 각 사용자에게 분산시켜, 중앙 서버는 데이터 처리 및 AI 분석 요청에만 집중할 수 있도록 구성합니다.
- 서버 부하 및 IP 차단 문제 해결: 이 구조는 스크래핑으로 인한 서버의 직접적인 부하를 제거하며, 요청이 각 사용자의 IP로 분산되므로 특정 쇼핑몰의 IP 차단 위험을 근본적으로 해결합니다. 이를 통해 저비용으로 안정적인 서비스 확장을 달성합니다.
- 데이터 전송 API 설계: 클라이언트가 수집한 데이터를 중앙 서버로 안전하고 효율적으로 전송할 수 있는 전용 API를 설계해야 합니다. 서버는 이 API를 통해 데이터를 수신하여 AI 분석, 결과 제공, 라이브러리 저장의 역할에 집중함으로써 시스템의 역할을 명확히 하고 유지보수성을 향상시킵니다.

3.4.2 관리자 요구사항

안정적인 시스템 운영 및 신속한 장애 대응을 위해 아래와 같은 관리자 기능을 제공해야 합니다.

- 시스템 모니터링 대시보드: 서버 관리자는 웹 기반의 대시보드를 통해 주요 시스템 자원(CPU, 메모리, 디스크 사용량, 네트워크 트래픽)의 상태를 실시간으로 확인할 수 있어야 합니다. 자원 사용량이 사전에 설정된 임계치(예: 80%)를 초과할 경우, 관리자에게 이메일 또는 슬랙(Slack)으로 경고 알림을 발송해야 합니다.
- 상세 로그 기록 및 조회: 시스템의 모든 주요 동작과 오류는 로그로 기록되어야 합니다. 특히 아래 상황에서는 원인 파악이 용이하도록 상세 정보를 포함한 로그를 생성해야 합니다.
 - 스크래핑/크롤링 실패: 대상 URL, 실패 시점, HTTP 상태 코드, 오류 메시지 등
 - AI 분석 API 연동 오류: 요청 데이터, API 응답 코드, 오류 메시지 등
 - 데이터베이스 오류: 실행 실패 쿼리, 오류 코드 등

3.4.3 이식성 요구사항

본 시스템은 특정 인프라 환경에 종속되지 않고, 다양한 서버 환경으로 쉽게 이전(Migration)할 수 있도록 개발되어야 합니다.

- 표준 기술 스택 사용: 백엔드는 Python, 프론트엔드는 표준 웹 기술(HTML/CSS/JS)을 사용하여 특정 플랫폼에 대한 종속성을 최소화합니다.
- 컨테이너화(Containerization) 도입: 도커(Docker)를 사용하여 애플리케이션을 컨테이너화하는 것을 원칙으로 합니다. 이를 통해 개발, 테스트, 운영 환경의 일관성을 보장합니다.

3.5 Maintenance (유지관리)

3.5.1 시스템 운영 및 버전 관리

- 버전 관리: 모든 소스 코드는 Git을 통해 버전을 관리하며, 중앙 원격 저장소로 GitHub를 사용해야 합니다.
- 브랜치 관리: main 브랜치의 소스 코드는 항상 안정적으로 실행 가능한 상태(Stable)로 유지되어야 합니다. 기능 개발 및 수정은 별도의 브랜치에서 진행한 후, 충분한 테스트를 거쳐 main 브랜치에 병합(Merge)하는 것을 원칙으로 합니다.

3.5.2 유지관리를 위한 조직 요구사항

- 정기 회의: 프로젝트의 원활한 진행과 소통을 위해 매주 1회 정기 회의를 진행하여 진행 상황, 이슈, 변경사항을 공유해야 합니다.
- 문서 관리: 모든 공식 산출물(요구사항 분석서, 설계서, 회의록 등)은 Google Workspace를 활용하여 작성하며, 최종 문서는 GitHub 저장소 내 docs 디렉터리에 기록하고 관리해야 합니다.

3.6 Implementation (구현상 제약사항)

3.6.1 하드웨어 요구사항

- 개발 환경: 모든 개발자는 Docker Desktop의 최소 시스템 요구사항을 충족하는 개인 PC 또는 Laptop을 갖추어야 합니다.

- 배포 환경: 서비스의 배포 및 운영은 **NHN Cloud**를 사용하며, 초기에는 가상 서버(**VM**) 1대를 기준으로 시스템을 구축합니다.

3.6.2 테스트(*Testing*) 요구사항

- 단위 테스트 (**Unit Test**): 개발자는 함수, 클래스 등 코드의 가장 작은 단위가 의도대로 동작하는지 검증하는 단위 테스트 코드를 작성하고 실행해야 합니다.
- 통합 테스트 (**Integration Test**): 각 모듈을 결합하는 과정에서 발생하는 인터페이스 및 데이터 연동 문제를 검증하는 통합 테스트를 수행해야 합니다. 이는 *main* 브랜치로 코드를 병합하기 전에 필수적으로 수행되어야 합니다.
- 시스템 테스트 (**System Test**): 최종 사용자의 관점에서 시스템이 요구사항 명세서의 모든 기능을 만족시키는지 종합적으로 테스트해야 합니다.

3.7 Interface (인터페이스)

3.7.1 외부 시스템 연동 의존성

본 시스템은 안정적인 기능 제공을 위해 다음과 같은 외부 시스템에 의존하며, 해당 시스템의 정책 변경 시 유지보수가 필요합니다.

- 웹 스크래핑 대상 시스템: 쿠팡(**Coupang**) 웹사이트의 **HTML** 구조 및 로봇 배제 표준(**robots.txt**) 정책에 직접적으로 의존합니다. 쿠팡 웹사이트의 구조가 변경될 경우, 스크래핑 모듈의 즉각적인 수정이 요구됩니다.
- AI 분석 시스템: **Google AI Studio**에서 제공하는 **API**를 사용합니다. 해당 **API**의 명세, 사용량 제한, 인증 방식 등 정책 변경에 영향을 받으므로 이를 지속적으로 준수해야 합니다.

3.7.2 데이터 호환성

- 클라이언트(웹 브라우저)와 서버(**Flask Application**) 간의 모든 데이터 통신은 표준 **JSON(JavaScript Object Notation)** 형식을 사용해야 합니다.

3.7.3 표준 인터페이스 사용

- 프론트엔드: 웹 표준인 **HTML5**, **CSS3**, **JavaScript**를 사용하여 개발합니다.
- 백엔드: 외부 통신을 위한 인터페이스는 **RESTful API** 설계 원칙을 준수하여 구현해야 합니다.

3.8 Legal (법적 제약사항)

3.8.1 라이선스

- 오픈소스 정책: 개발에 사용되는 주요 프레임워크 및 라이브러리(**Flask**, **PyMySQL** 등)는 상업적 활용이 가능한 오픈소스 라이선스를 따르는 것을 원칙으로 합니다.
- 라이선스 명시: 사용하는 모든 오픈소스 라이브러리의 종류와 해당 라이선스 정보는 **docs/licenses.md** 파일에 명시하고 지속적으로 관리해야 합니다.

3.8.2 서비스 운영 정책

- 웹 스크래핑: 대상 웹사이트(쿠팡)의 **robots.txt** 정책을 반드시 준수해야 하며, 과도한 트래픽을 유발하지 않도록 요청 사이에 지연(**Delay**) 시간을 설정해야 합니다. 수집된 리뷰 데이터는 본 프로젝트 기간 동안 학술 및 교육 목적으로만 사용하며, 어떠한 상업적 용도로도 활용하지 않습니다.
- **AI API: Google AI Studio**의 서비스 이용 약관과 사용자 정책을 모두 준수해야 합니다.

3.8.3 법적 책임 소재

- 본 프로젝트는 상업적 이익을 목적으로 하지 않는 학술 및 교육용 프로젝트입니다. 따라서 시스템이 제공하는 정보의 정확성이나 신뢰성을 보증하지 않으며, 시스템 오작동으로 인해 발생하는 문제에 대한 법적 책임은 프로젝트 팀에 없습니다.

3.8.4 비용

- 개발 비용: 모든 개발은 오픈소스 소프트웨어를 기반으로 진행하여 별도의 라이선스 구매 비용을 발생시키지 않습니다.
- 운영 비용: 서버 운영은 **NHN Cloud**에서 프로젝트 기간 동안 제공하는 무료 크레딧 또는 프리 티어(**Free Tier**) 범위 내에서만 이루어지는 것을 원칙으로 합니다.