##### Project Document

Test Plan / Test Cases Design Document

|  |  |
| --- | --- |
| Project Name | 사용자 이미지 기반 AI 챗봇 |

9조

201902770 허정우

202002470 김보배

지도교수: 임성수 교수님 (서명)

Table of Contents

[1. Introduction 3](#_Toc197954646)

[1.1. Objective 3](#_Toc197954647)

[2. Test Plan 4](#_Toc197954648)

[3. Test Cases 8](#_Toc197954649)

[4. AI 도구 활용 정보 9](#_Toc197954650)

# Introduction

## Objective

이 문서는 사용자 이미지 기반 AI 챗봇 서비스의 테스트 계획, 테스트 케이스 명세를 포함하고 있다.

테스트 계획 문서는 구현된 시스템을 대상으로 어떤 요소를 테스트할 것인지, 또한 어떤 방법으로 테스트를 수행할 것인지 등 테스트 활동에 대한 전반적인 계획을 포함하고 있다.

테스트 케이스 명세는 대상 시스템에서 테스트될 요소를 테스트하기 위해 필요한 테스트 케이스를 구체적으로 기술한다.

# Test Plan

|  |
| --- |
| **1. 배경과 목적** |
| 1.1 배경 |
| 이 문서는 사용자 이미지 기반 AI 챗봇에 대한 테스트 계획의 개요를 설명한 것이다. 중요한 핵심 기능들은 다음과 같다.   * 사용자의 이미지 기반 질의 처리 * 새로운 이미지 업로드 * 이미지 삭제 |
| 1.2 테스트 목적 |
| 테스트를 통해 프로젝트의 핵심 기능과 처리가 만족할 만한 수준으로 신뢰성 있게 개발되었는지 확인하는 것이다. |
| **2. 테스트 상세** |
| 2.1 테스트 항목 |
| 사용자 이미지 기반 AI 챗봇 |
| 2.2 테스트될 요소(features) |
| * 사용자의 질의에 대해 적절한 응답 생성 * 사용자가 새로운 이미지 업로드 * 사용자가 기존 이미지 삭제 |
| 2.3 테스트되지 않을 요소 |
| * 로그인, 회원가입 기능에 기반한 사용자 인증 * 시스템 응답 속도 및 처리 속도 * LLM injection등의 보안 요소 * 과도한 트래픽 대처 같은 서버 관련 요소 |
| 2.4 접근 방법 |
| * 단위 테스트(**Mock 데이터 이용**) * 시스템 테스트(기능 테스트) |
| 2.5 테스트 항목의 pass/fail 기준 |
| 단위 테스트   * 각 함수의 입출력 값이 예상과 일치해야 함 * 예외 상황에 대해 적절한 오류 처리 로직이 동작해야 함   시스템 테스트   * 시스템 요구사항 명세서에 명시된 모든 기능이 정상적으로 수행되어야 하며, 각 입력 케이스에 대해 적절한 출력이 나와야한다. |
| 2.6 테스트 산출물(deliverables) |
| * 테스트 계획 * 테스트 케이스 명세 * 테스트 결과 보고서 |
| **3. 테스트 관리** |
| 3.1 작업 |
| 1. 시스템 요구사항 기반으로 테스트 계획 작성  2. 테스트 수행  3. 결함 확인  4. 결함 수정  5. 재 테스트 진행  6. 결과 보고 작성 |
| 3.2 기술 자원 |
| * 테스트에 필요한 사진 데이터 * 그래프 DB * 테스트 서버, 테스트 클라이언트 * LLM * 정보 추출에 사용될 모델 |
| 3.3 책임과 권한 (인력 자원) |
| * 사진 데이터를 제공할 수 있는 사용자 * 그래프 DB를 구축한 IT 인력 * 기능별 파이프라인을 구축한 백엔드 개발자 * UI를 제공하는 프론트엔드 개발자 |
| 3.4 훈련 |
| * 사진 데이터를 제공한 사용자에게 시스템의 동작 방법과 사용 방법을 설명함 |
| 3.5 일정 |
| * 미리 사진 데이터가 준비되어 있다면 바로 테스트를 수행 가능함 |
| 3.6 위험 요소와 비상 대처 상황 |
| 없다. |

# Test Cases

|  |
| --- |
| **1. 서론** |
| 1.1 테스트 범위 |
| 본 테스트는 챗봇 서비스의 각 기능 별 파이프라인를 모듈로 나눠 단위 테스트를 진행한다. 각 파이프라인은 다음과 같고 아래 세부 모듈로 구성되어있다.   * 사용자가 업로드한 이미지로부터 정보를 추출하여 그래프에 저장하는 파이프라인   + 이미지에서 정보 추출   + 정보를 그래프 데이터베이스 쿼리로 변환   + 그래프 데이터베이스에 저장 * 질의를 가지고 그래프를 검색하여 RAG방식으로 응답을 생성하는 파이프라인   + 자연어 질의를 그래프 데이터베이스 쿼리로 변환   + 그래프 데이터베이스 검색으로 정보 추출   + 정보로 LLM이 응답 생성 * 사용자가 이미지를 삭제하면 그래프에서 정보도 삭제하는 파이프라인   + 그래프 데이터베이스 쿼리 생성   + 그래프 데이터베이스에서 정보 삭제 |
| 1.2 테스트 상황 |
| 테스트는 개발 도중 각 모듈이 완성될 때 개발자에 의해 진행되며 개발 환경에서 진행된다. |
| 1.3 문서 표기법 |
| 각 기능을 세부 모듈 단위로 나누어 테스트 진행하므로, Id에 (코드번호)-순서 로 표기하여 기능을 표시함   * 사용자가 업로드한 이미지로부터 정보를 추출하여 그래프에 저장하는 기능(U1, U2, U3) * 질의를 가지고 그래프를 검색하여 RAG방식으로 응답을 생성하는 기능(M1, M2, M3) * 사용자가 이미지를 삭제하면 그래프에서 정보도 삭제하는 기능(D1, D2) |
| **2. 테스트 케이스** |
| 2.1 테스트 케이스 명세 |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | Id | 테스트 대상 | 테스트 조건 | 테스트 데이터 | 예상 결과 | | U1 | 이미지에서 정보 추출 | 이미지는 수신했다고 가정 | 테스트용 이미지 | 메타데이터, 객체, 객체 사이 관계 등의 정보가 추출됨 | | U2 | 정보를 데이터베이스 쿼리로 변환 | 없음 | Json 형식의 정보 데이터 | 바로 실행가능한 쿼리를 출력 | | U3 | 그래프 데이터베이스에 저장 | 그래프 데이터베이스가 존재함 | CREATE 쿼리 | 그래프 데이터 베이스에 노드와 엣지가 추가됨 | | M1 | 자연어 질의를 그래프 데이터베이스 쿼리로 변환 | 입력을 클라이언트로부터 전송받았다고 가정함 | 테스트용 자연어 질의 | 바로 실행가능한 쿼리를 출력 | | M2 | 그래프 데이터베이스 검색으로 정보 추출 | 그래프 데이터베이스에 데이터가 존재함 | MATCH 쿼리 | 검색 결과들이 나온다 | | M3 | 결과를 바탕으로 자연어 응답 생성 | LLM 모델이 있음 | 검색된 결과 | 자연어 응답 출력 | | D1 | 데이터베이스 쿼리 생성 | 삭제하려는 사진의 ID가 주어짐 | 사진의 고유 ID | 바로 실행 가능한 Delete 쿼리로 출력 | | D2 | 그래프 데이터베이스에서 정보 삭제 | 삭제하려고 하는 사진의 ID가 DB에 존재함. | Delete 쿼리 | 데이터베이스 상에서 사진과 관련된 노드와 엣지가 삭제됨. | |
| 2.2 테스트 환경 |
| * 개발 서버(python) * 그래프 DB(neo4j) * LLM(GPT API) |
| 2.3 테스트 절차 요구사항 |
| * 그래프 데이터베이스가 미리 구축되어 있어야 함. * LLM이 있어야함 * 테스트용 이미지가 존재해야함 |

|  |
| --- |
| **1. 서론** |
| 1.1 테스트 범위 |
| 본 테스트는 챗봇 서비스의 기능 테스트를 서술한다. 테스트를 진행할 기능은 다음과 같다.   * 사용자가 이미지를 업로드하면 그래프에 저장하는 기능 (Upload) * 질의를 입력받고 응답을 생성하는 기능 (Match) * 사용자가 이미지를 삭제하면 그래프에서 정보도 삭제하는 기능(Delete) |
| 1.2 테스트 상황 |
| 테스트는 개발 완료 후 테스트 사용자에 의해 진행된다. |
| 1.3 문서 표기법 |
| 각 기능을 테스트 데이터 별로 나누어 예상 결과를 서술한다. 각 id는 Upload, Match, Delete에 해당하는 기능이다. |
| **2. 테스트 케이스** |
| 2.1 테스트 케이스 명세 |
| |  |  |  |  |  | | --- | --- | --- | --- | --- | | id | 테스트 대상 | 테스트 조건 | 테스트 데이터 | 예상 결과 | | Upload1 | 사용자가 이미지를 업로드하면 그래프에 저장하는 기능 | 그래프 데이터베이스 존재, 이미지 파일은 전송받음 | 정상적인 이미지 파일 | 그래프에 노드와 엣지 추가, 성공 메시지 | | Upload2 | 사용자가 이미지를 업로드하면 그래프에 저장하는 기능 | 이미지 파일은 전송받음 | 이미지 파일은 맞는데 아무런 정보가 없거나 심하게 흔들림 | 오류로 처리 | | Upload3 | 사용자가 이미지를 업로드하면 그래프에 저장하는 기능 | 이미지 파일은 전송받음 | 이미지 파일이 아님 | 오류로 처리 | | Match1 | 질의를 입력받고 응답을 생성하는 기능 | 그래프 데이터베이스에 미리 정보가 입력됨 | 데이터 베이스에 있는 데이터에 대한 질의 입력 | 정상적인 응답 생성 | | Match2 | 질의를 입력받고 응답을 생성하는 기능 | 그래프 데이터베이스가 존재함 | 데이터베이스에 없는 데이터에 대한 질의(날짜 정보가 없거나, 장소 정보가 없거나) | 정보를 찾을 수 업다는 응답 생성 | | Match3 | 질의를 입력받고 응답을 생성하는 기능 | 그래프 데이터베이스가 존재함 | 질의가 이상한 경우 | 질의를 다시 입력하라는 응답 생성 | | Delete1 | 사용자가 이미지를 삭제하면 그래프에서 정보도 삭제하는 기능 | 그래프 데이터베이스가 존재함 | 데이터베이스에 존재하는 이미지 ID | 그래프에서 관련 노드와 엣지 모두 삭제 | | Delete2 | 사용자가 이미지를 삭제하면 그래프에서 정보도 삭제하는 기능 | 그래프 데이터베이스가 존재함 | 데이터베이스에 존재하지 않는 이미지 ID | 오류로 처리 | |
| 2.2 테스트 환경 |
| * 서버(python) * 그래프 데이터베이스(neo4j) * LLM(GPT API) |
| 2.3 테스트 절차 요구사항 |
| * 데이터 베이스 미리 구축, 접근 가능 * 테스트용 이미지 데이터들 존재 * LLM 접근 가능 |

# AI 도구 활용 정보

|  |  |
| --- | --- |
| *사용 도구* |  |
| *사용 목적* |  |
| *프롬프트* |  |
| *반영 위치* |  |
| *수작업*  *수정* |  |