<AI\_GAME\_MANAGEMENT>

Version <1.0>

2025.05.24

[Note 이 문서는 사용자의 게임 성향, 게임결과, 만족도를 기반으로 AI(chatGPT)를 활용해 게임 추천 및 공략법 제공, 게임 이력 저장 및 분석 기능을 구현하기 위한 요구사항 정의서]

[추가 부연 설명으로 본 시스템은 웹페이지를 기반으로 구축되며 추후 하이브리드 앱 형태로 모바일 서비스까지 확장할 예정입니다 사용자는 PC에서 웹 브라우저를 통해, 추후 앱 형태로 확장되면 모바일에서는 전용 앱을 통해 시스템에 접근할 수 있습니다 또한 회원 가입 및 로그인 기능을 통해 사용자 개별 데이터를 관리하며 사용자의 플레이 기록과 만족도 정보를 바탕으로 더욱 정교한 게임 추천을 수행합니다]

작성자: 팀장: 민태균  
  
  
 팀원: 이준수, 최현수

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| **<2025/04/10> 초안** | **<1.0>** | **<초안 작성>** | **<이준수>** |
| **<2025/04/11> 수정** | **<1.0.1>** | **<목차 작성 및 수정>** | **<이준수>** |
| **<2025/04/16> 수정** | **<1.0.2>** | **<목차 작성 및 수정>** | **<최현수>** |
| **<2025/05/03> 수정** | **<1.0.3>** | **<유스케이스 다이어그램 작성>** | **<이준수>** |
| **<2025/05/03> 수정** | **<1.0.4>** | **<목차 수정 및 내용 작성>** | **<최현수>** |
| **<2025/05/04> 수정** | **<1.0.5>** | **<** **2.2 Assumptions and Dependencies, 3.1 AI 작동 방식 내용 작성>** | **<최현수>** |
| **<2025/05/08> 수정** | **<1.0.6>** | **<프로젝트 간트차트 작성>** | **<민태균>** |
| **<2025/05/14> 수정** | **<1.0.7>** | **<기능적 요구사항 비기능적 요구사항 작성>** | **<이준수>** |
| **<2025/05/16> 수정** | **<1.0.8>** | **<시퀀스 다이어그램 작성>** | **<이준수>** |
| **<2025/05/18> 수정** | **<1.0.9>** | **<데이터 흐름도 작성>** | **<최현수>** |
| **<2025/05/22> 수정** | **<1.1.0>** | **<오타 수정, 내용 보완 (간트차트 길이 조정 및 데이터 흐름도 재작성 포함)>** | **<최현수>** |
| **<2025/05/24> 수정** | **<1.1.1>** | **<클래스 다이어그램 작성>** | **<이준수>** |
| **<2025/05/26> 수정** | **<1.1.2>** | **<시스템 구성도 작성 목차 수정>** | **<이준수>** |
| **<2025/05/27> 수정** | **<1.1.3>** | **<흐름 재작성>** | **<최현수>** |

목차

**1. Introduction (서론)**

1.1 Purpose (목적)   
 1.2 Document Scope (문서 범위)

1.3 Definitions, Acronyms and Abbreviations (정의 및 약어)

1.4 References (참고자료)   
 1.5 Overview (개요)

**2. 전체 시스템 설명 (Overall Description)**

2.1 UsecaseDiagram 상세화   
 2.1.1 회원가입  
 2.1.2 로그인  
 2.1.3 게임 추천 받기  
  2.1.4 공략법 요청하기  
  2.1.5 게임 결과 등록  
  2.1.6 만족도 평가  
  2.1.7 마이페이지 조회  
  2.1.8 관리자 - 사용자 정보 조회  
  2.1.9 관리자 - 게임 정보 등록/수정  
  2.1.10 관리자 - 시스템 로그 확인

2.2 Assumptions and Dependencies (가정 및 의존사항)

**3. 시스템 기술 요구사항 (Specific Requirements)**

3.1 AI 작동 방식 (AI Processing Logic) // 입력 데이터 → 벡터화 → 추천 알고리즘 → 출력의 흐름, Prompt 구성 방식, 모델 API 활용 등

3.2 기능적 요구사항(Functional Requirements)

3.3 비기능적 요구사항(Non-Functional Requirements)

**4. 설계 관련 정보 (Design Information)**

4.1 시스템 아키텍처 (Architecture Diagram)

4.1.1 언어 및 프레임워크

4.1.2 운영체제

4.1.3 통신

4.2 시스템 구성도

4.3 데이터 흐름도 (Data Flow Diagram - DFD)  
4.4 SequenceDiagram 상세화

4.5 클래스 다이어그램 (Class Diagram)

4.5.1 클래스 다이어그램 세분화  
4.5.2 플레이 이력

4.5.3 게임 정보

4.5.4 사용자 프로필

4.5.5 공략

4.5.6 만족도평가

4.5.7 추천엔진

4.5.8 CharGPT API

4.5.9 벡터처리기

4.5.10 시스템로그

4.5.11 백엔드

4.5.12 데이터베이스

4.5.13 알림

4.6 UI

4.7 간트차트

**5. 지원 정보 (Supporting Information)**

**6. 참고 문헌 (References)**

# 서론 (Introduction)

본 시스템은 사용자의 게임 장르 선호도, 플레이 이력, 만족도 평가 등을 기반으로 최적의 게임을 추천하고, 인공지능을 활용한 공략 정보를 제공하는 스마트 웹 애플리케이션이다.

## 1.1 목적 (Purpose)

이 문서는 AI 기반 게임 추천 및 공략 정보 제공 시스템(AI Game Management System)의 소프트웨어 요구사항을 정의한다. 이 시스템은 사용자의 게임 플레이 이력, 선호도, 만족도 평가 등을 분석하여 개인화된 게임 추천과 공략 정보를 제공하는 것을 목표로 한다. 본 문서는 개발자, 기획자, 디자이너, 테스트 담당자 등 모든 이해관계자에게 일관된 기준을 제공하며, 시스템 구현 전반에 걸쳐 참조 문서로 사용된다.

## 문서범위 (Document Scope)

본 문서에서 정의하는 시스템은 다음과 같은 기능을 제공한다:

* 사용자 등록 및 인증
* 사용자 취향 기반 게임 추천
* 선택된 게임에 대한 AI 기반 공략 정보 제공
* 게임 플레이 결과 및 만족도 등록
* 사용자 만족도 기반 분석 및 반영
* 관리자 전용 기능: 데이터 관리, 사용자 정보 열람 등

시스템은 웹 기반으로 구현되며, 추후 모바일 및 기타 플랫폼 확장을 고려한다. 외부 API로는 OpenAI API, Steam API 등을 사용하며, 서비스 제공을 위한 백엔드 서버 및 데이터베이스 구축이 전제된다.

## 정의 및 약어 (Definitions, Acronyms and Abbreviations)

|  |  |
| --- | --- |
| AI | Artificial Intelligence. 본 문서에서는 OpenAI의 ChatGPT API를 의미함. |
| 공략 | 게임 플레이를 효율적으로 진행하기 위한 전략 또는 팁. AI에 의해 자동 생성됨. |
| API | Application Programming Interface. 외부 기능을 활용하기 위한 인터페이스. |
| Steam | Valve사에서 운영하는 게임 유통 플랫폼. |

## 참고자료 (References)

## OpenAI API 공식 문서: <https://platform.openai.com/docs>

## Steamworks API 문서: <https://partner.steamgames.com/doc>

## 개요 (Overview)

이 문서는 다음과 같은 구조로 구성되어 있다:

* 2장에서는 시스템의 전반적인 설명과 유스케이스, 흐름 등을 다룬다.
* 3장에서는 기능적 및 비기능적 요구사항을 기술한다.
* 4장에서는 시스템 설계 관련 참고 자료를 포함한다.
* 5장은 구현에 필요한 보조 정보 및 참고 문헌으로 구성된다.

# 전체 시스템 설명 (Overall Description)

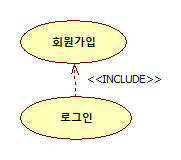
## UsecaseDiagram 상세화

## 회원가입

|  |  |
| --- | --- |
| **Use Case # 001** | |
| **GENERAL CHARACTERISTICS** | |
| Use Case ID | UC001 |
| 유스케이스 명 | 회원가입 |
| 작성자 | 최현수 |
| 액터 | 사용자 |
| 사전 조건 | 사용자가 로그아웃 상태이며, 등록되지 않은 이메일을 보유하고 있어야 한다. |
| 사후 조건 | 계정이 생성되어 로그인할 수 있으며, 초기 사용자 데이터가 저장된다. |
| 트리거 | 사용자가 '회원가입' 버튼을 클릭한다. |

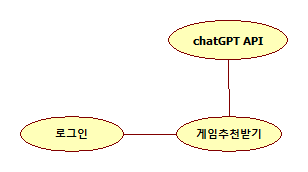
|  |  |  |
| --- | --- | --- |
| 기본 흐름 | | |
| 1 | 사용자 | 사용자가 웹페이지 또는 앱에서 '회원가입' 버튼을 클릭한다. |
| 2 | 시스템 | 시스템은 회원가입 양식(아이디, 이메일, 비밀번호, 선호 게임 장르 등)을 렌더링한다. |
| 3 | 사용자 | 사용자가 양식을 작성하고 '등록' 버튼을 클릭한다. |
| 4 | 시스템(프론트엔드) | 프론트엔드는 입력값 유효성 검사(형식, 공백 등)를 수행하고, 이상 없을 시 백엔드로 요청을 전송한다. |
| 5 | 시스템(백엔드) | 백엔드는 이메일 중복 여부 확인 및 비밀번호 해시 처리를 수행한다. |
| 6 | 시스템(DB) | DB는 사용자 계정 정보를 저장한다. |
| 7 | 시스템(프론트엔드) | 프론트엔드는 '회원가입이 완료되었습니다' 메시지를 표시한다. |
| 8 | 시스템 | 시스템은 초기 로그인 페이지 또는 메인 페이지로 리다이렉션한다. |
| 대안 흐름 | | |
| 3A | 필수 입력값 누락 | "필수 항목을 입력해주세요" 메시지를 출력하고 누락된 입력란을 강조한다. |
| 4A. | 이메일 형식 오류 | "유효한 이메일 주소를 입력해주세요" 메시지를 출력한다. |
| 5A | 이메일 중복 | "이미 등록된 이메일입니다. 다른 이메일을 입력해주세요" 메시지를 출력한다. |
| 6A | DB 저장 실패 | 오류 메시지를 출력하고 재시도 버튼을 제공하며 로그를 기록한다. |

## 로그인



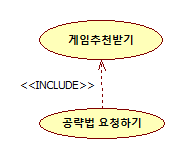
|  |  |
| --- | --- |
| **Use Case # 002** | |
| **GENERAL CHARACTERISTICS** | |
| Use Case ID | UC002 |
| 유스케이스 명 | 로그인 |
| 작성자 | 최현수 |
| 액터 | 사용자 |
| 사전 조건 | 사용자가 이미 회원가입을 완료한 상태이며, 올바른 아이디 또는 이메일과 비밀번호를 알고 있어야 한다. |
| 사후 조건 | 사용자가 인증에 성공하여 로그인되고, 사용자 세션이 생성된다. |
| 트리거 | 사용자가 '로그인' 버튼을 클릭한다. |

|  |  |  |  |
| --- | --- | --- | --- |
| 기본 흐름 | | | |
| 1 | 사용자 | 사용자가 웹페이지 또는 앱에서 '로그인' 버튼을 클릭한다. | |
| 2 | 시스템(프론트엔드) | 시스템은 로그인 양식(아이디, 비밀번호 입력)을 렌더링한다. | |
| 3 | 사용자 | 사용자가 정보를 입력하고 '로그인' 버튼을 클릭한다. | |
| 4 | 시스템(프론트엔드) | 프론트엔드는 입력값 유효성 검사를 수행한 후 백엔드로 요청을 전송한다. | |
| 5 | 시스템(백엔드) | 백엔드는 DB에서 사용자 정보를 조회한다. | |
| 6 | 시스템(DB) | 입력된 비밀번호를 해시화하여 저장된 값과 비교해 인증을 수행한다. | |
| 7 | 시스템(백엔드) | 인증 성공 시, 시스템은 세션을 생성하고 사용자 정보를 저장한다. | |
| 8 | 시스템(프론트엔드) | 프론트엔드는 메인 페이지로 리다이렉션하며 '로그인 성공' 메시지를 출력한다. | |
| 대안 흐름 | | | |
| 3A | 필수 입력값 누락 | | "아이디와 비밀번호를 모두 입력해주세요" 메시지를 출력한다. |
| 4A | 아이디 또는 비밀번호 형식 오류 | | "올바른 형식으로 입력해주세요" 메시지를 출력한다. |
| 5A | 아이디 존재하지 않음 | | "존재하지 않는 계정입니다" 메시지를 출력한다. |
| 6A | 비밀번호 불일치 | | "비밀번호가 일치하지 않습니다" 메시지를 출력한다. |

**2.1.3 게임추천 받기**

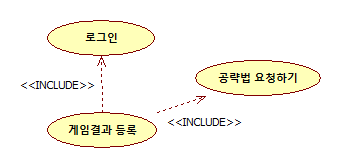
|  |  |  |
| --- | --- | --- |
| 기본 흐름 | | |
| 1 | 사용자 | 사용자가 '게임 추천 받기' 버튼을 클릭한다. |
| 2 | 시스템(프론트엔드) | 프론트엔드는 사용자 ID 기반으로 백엔드에 추천 요청을 전송한다. |
| 3 | 시스템(백엔드) | 백엔드는 사용자 선호도, 게임 이력, 만족도 평가 데이터를 수집한다. |
| 4 | 시스템(백엔드) | 수집된 데이터를 벡터화하고 추천 알고리즘에 입력한다. |
| 5 | 시스템(백엔드) | 백엔드는 ChatGPT API에 프롬프트를 구성하여 추천을 요청한다. |
| 6 | 시스템(프론트엔드) | 프론트엔드는 추천 게임 목록(게임명, 장르, 설명 등)을 사용자에게 표시한다. |
| 대안 흐름 | | |
| 3A | 사용자 이력 부족 | "추천을 위해 더 많은 이력이 필요합니다" 메시지를 출력한다. |
| 5A | API 응답 지연 | 로딩 상태를 유지하고 재시도 버튼을 제공한다. |
| 6A | 렌더링 실패 | "목록을 불러올 수 없습니다" 메시지를 출력한다. |

|  |  |
| --- | --- |
| **Use Case # 003** | |
| **GENERAL CHARACTERISTICS** | |
| Use Case ID | UC003 |
| 유스케이스 명 | 게임추천 받기 |
| 작성자 | 최현수 |
| 액터 | 사용자 |
| 사전 조건 | 사용자가 로그인 상태이며, 시스템에 선호 장르, 게임 플레이 이력, 만족도 데이터가 저장되어 있어야 한다. |
| 사후 조건 | 추천된 게임 목록이 사용자에게 제공되며, 사용자는 이를 선택하거나 정보를 확인할 수 있다. |
| 트리거 | 추천된 게임 목록이 사용자에게 제공되며, 사용자는 이를 선택하거나 정보를 확인할 수 있다. |

**2.1.4 공략법 요청하기**

|  |  |  |  |
| --- | --- | --- | --- |
| 기본 흐름 | | | |
| 1 | 사용자 | | 사용자가 '공략법 요청하기' 버튼을 클릭한다. |
| 2 | 시스템(프론트엔드) | | 프론트엔드는 선택된 게임 ID를 백엔드로 전송한다. |
| 3 | 시스템(백엔드) | | 백엔드는 게임 정보를 확인한 후 ChatGPT API 요청을 준비한다. |
| 4 | 시스템(백엔드) | | 공략 목적에 맞게 프롬프트를 구성하고 API 요청을 전송한다. |
| 5 | 시스템(백엔드) | | 백엔드는 응답을 해석하고 공략 정보를 형식화한다. |
| 6 | 시스템(프론트엔드) | | 프론트엔드는 공략 정보를 사용자에게 시각적으로 표시한다. |
| 대안 흐름 | | | |
| 2A | 게임 미선택 | "공략할 게임을 선택해주세요" 메시지를 출력한다. | |
| 4A | API 요청 실패 | "공략 생성 실패" 메시지와 재시도 버튼을 제공한다. | |
| 6A | 공략 렌더링 오류 | 공략 정보를 텍스트 형태로 표시한다. | |

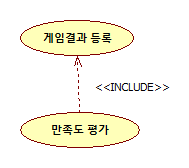
|  |  |
| --- | --- |
| **Use Case # 004** | |
| **GENERAL CHARACTERISTICS** | |
| Use Case ID | UC004 |
| 유스케이스 명 | 공략법 요청하기 |
| 작성자 | 최현수 |
| 액터 | 사용자 |
| 사전 조건 | 사용자가 로그인 상태이며, 공략법 요청 대상 게임이 존재해야 한다. |
| 사후 조건 | 선택한 게임의 공략법이 생성되어 사용자에게 제공된다. |
| 트리거 | 사용자가 '공략법 요청하기' 버튼을 클릭한다. |

**2.1.5 게임결과 등록**

|  |  |
| --- | --- |
| **Use Case # 005** | |
| **GENERAL CHARACTERISTICS** | |
| Use Case ID | UC005 |
| 유스케이스 명 | 게임결과 등록 |
| 작성자 | 최현수 |
| 액터 | 사용자 |
| 사전 조건 | 사용자가 로그인 상태이며, 결과를 등록할 게임이 존재해야 한다. |
| 사후 조건 | 게임 결과가 저장되고 사용자의 플레이 이력이 갱신된다. |
| 트리거 | 사용자가 '게임 결과 등록' 버튼을 클릭한다. |

|  |  |  |  |
| --- | --- | --- | --- |
| 기본 흐름 | | | |
| 1 | 사용자 | | 사용자가 '게임 결과 등록' 버튼을 클릭한다. |
| 2 | 시스템(프론트엔드) | | 시스템은 게임명, 플레이 시간 등의 입력 양식을 제공한다. |
| 3 | 사용자 | | 사용자가 정보를 입력하고 '저장'을 클릭한다. |
| 4 | 시스템(프론트엔드) | | 프론트엔드는 유효성 검사를 수행하고 백엔드에 전송한다. |
| 5 | 시스템(백엔드) | | 백엔드는 DB에 저장 요청을 보낸다. |
| 6 | 시스템(DB) | | DB는 성공 응답을 반환한다. |
| 대안 흐름 | | | |
| 2A | 양식 로딩 실패 | 재시도 버튼을 제공한다. | |
| 3A | 필수 항목 누락 | 누락된 항목을 강조하고 오류 메시지를 표시한다. | |
| 5A | 저장 실패 | 오류 메시지를 표시하고 재시도를 유도한다. | |

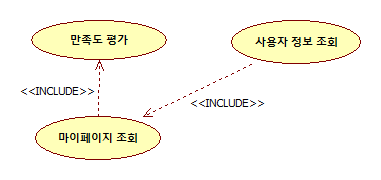
**2.1.6 만족도 평가**



|  |  |  |  |
| --- | --- | --- | --- |
| 기본 흐름 | | | |
| 1 | 사용자 | | 사용자가 마이페이지 또는 평가 요청 화면에 접근한다. |
| 2 | 시스템(프론트엔드) | | 시스템은 최근 게임 결과를 기반으로 평가 대상 목록을 표시한다. |
| 3 | 사용자 | | 사용자가 점수(별점 또는 0~10점) 및 선택적 코멘트를 입력한다. |
| 4 | 사용자 | | 사용자가 '제출' 버튼을 클릭한다. |
| 5 | 시스템(프론트엔드) | | 프론트엔드는 유효성 검사를 수행한 후 백엔드로 전송한다. |
| 6 | 시스템(백엔드) | | 백엔드는 평가 데이터를 저장한다. |
| 대안 흐름 | | | |
| 2A | 평가 대상 없음 | "평가 가능한 게임이 없습니다" 메시지를 출력한다. | |
| 3A | 평가 미입력 | "만족도를 선택해주세요" 메시지를 출력한다. | |
| 6A | 저장 실패 | 오류 메시지를 표시하고 재시도를 유도한다. | |

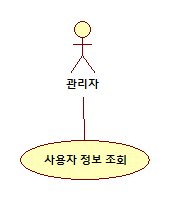
|  |  |
| --- | --- |
| **Use Case # 006** | |
| **GENERAL CHARACTERISTICS** | |
| Use Case ID | UC006 |
| 유스케이스 명 | 만족도 평가 |
| 작성자 | 최현수 |
| 액터 | 사용자 |
| 사전 조건 | 사용자가 로그인 상태이며, 최소 1회 이상 게임 결과를 등록한 이력이 있어야 한다. |
| 사후 조건 | 평가 데이터가 저장되어 시스템 분석에 활용된다. |
| 트리거 | 사용자가 평가 화면에 접근하여 '제출' 버튼을 클릭한다. |

**2.1.7 마이페이지 조회**



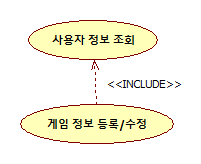
|  |  |  |  |
| --- | --- | --- | --- |
| 기본 흐름 | | | |
| 1 | 사용자 | | 사용자가 마이페이지 메뉴를 선택한다. |
| 2 | 시스템(프론트엔드) | | 프론트엔드는 사용자 ID를 기반으로 백엔드에 조회 요청을 보낸다. |
| 3 | 시스템(백엔드) | | 백엔드는 사용자 데이터를 조회 및 분석한다. |
| 4 | 시스템(DB) | | DB는 사용자 관련 데이터를 반환한다. |
| 5 | 시스템(프론트엔드) | | 프론트엔드는 게임 결과, 만족도, 추천 목록을 표시한다. |
| 6 | 사용자 | | 사용자는 정보를 확인하고 피드백을 제공할 수 있다. |
| 대안 흐름 | | | |
| 3A | DB 연결 실패 | 로딩 실패 메시지 및 새로고침 안내를 출력한다. | |
| 4A | 사용자 정보 없음 | 공란 표시 또는 안내 메시지를 표시한다. | |
| 5A | UI 렌더링 오류 | 최소한의 정보를 텍스트 형태로 제공한다. | |

|  |  |
| --- | --- |
| **Use Case # 007** | |
| **GENERAL CHARACTERISTICS** | |
| Use Case ID | UC007 |
| 유스케이스 명 | 마이페이지 조회 |
| 작성자 | 최현수 |
| 액터 | 사용자 |
| 사전 조건 | 사용자가 로그인 상태여야 한다. |
| 사후 조건 | 사용자 정보, 게임 결과, 추천 목록 등이 사용자에게 제공된다. |
| 트리거 | 사용자가 마이페이지 메뉴를 선택한다. |

**2.1.8 관리자 – 사용자 정보 조회**

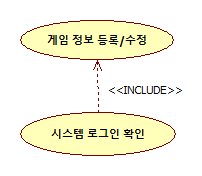
|  |  |  |  |
| --- | --- | --- | --- |
| 기본 흐름 | | | |
| 1 | 관리자 | | 관리자가 사용자 조회 메뉴를 클릭한다. |
| 2 | 시스템(프론트엔드) | | 프론트엔드는 사용자 목록 요청을 전송한다. |
| 3 | 시스템(백엔드) | | 백엔드는 사용자 목록을 조회하고 응답을 구성한다. |
| 4 | 시스템(DB) | | DB는 사용자 정보를 반환한다. |
| 5 | 관리자 | | 관리자가 특정 사용자를 선택한다. |
| 6 | 시스템(프론트엔드) | | 프론트엔드는 상세 정보를 표시한다. |
| 대안 흐름 | | | |
| 3A | DB 연결 실패 | 로딩 실패 메시지를 출력한다. | |
| 4A | 사용자 정보 없음 | 안내 메시지를 출력한다. | |
| 5A | UI 렌더링 오류 | 텍스트 기반 최소 정보를 제공한다. | |

|  |  |
| --- | --- |
| **Use Case # 008** | |
| **GENERAL CHARACTERISTICS** | |
| Use Case ID | UC008 |
| 유스케이스 명 | 관리자 – 사용자 정보 조회 |
| 작성자 | 최현수 |
| 액터 | 관리자 |
| 사전 조건 | 사용자가 관리자 권한을 가진 계정으로 로그인한 상태여야 한다. |
| 사후 조건 | 관리자에게 사용자 목록 또는 특정 사용자의 상세 정보가 제공된다. |
| 트리거 | 관리자가 사용자 정보 메뉴를 클릭한다. |

**2.1.9 관리자 - 게임 정보 등록/수정**

|  |  |  |  |
| --- | --- | --- | --- |
| 기본 흐름 | | | |
| 1 | 관리자 | | 관리자가 게임 정보 관리 메뉴에 진입한다. |
| 2 | 관리자 | | 관리자가 신규 등록 또는 수정할 게임을 선택한다. |
| 3 | 관리자 | | 게임명, 장르, 설명 등을 입력하거나 수정한다. |
| 4 | 관리자 | | '저장' 버튼을 클릭한다. |
| 5 | 시스템(백엔드) | | 백엔드는 입력값 유효성 검사 후 DB에 저장 요청을 보낸다. |
| 6 | 시스템(DB) | | DB는 게임 정보를 저장한다. |
| 대안 흐름 | | | |
| 2A | 수정 대상 없음 | "등록된 게임이 없습니다" 메시지를 출력한다. | |
| 4A | 필수 항목 누락 | 오류 메시지를 출력하고 해당 필드를 강조한다. | |
| 5A | 저장 실패 | 재시도 메시지를 출력한다. | |

|  |  |
| --- | --- |
| **Use Case # 009** | |
| **GENERAL CHARACTERISTICS** | |
| Use Case ID | UC009 |
| 유스케이스 명 | 관리자 – 게임 정보 등록/수정 |
| 작성자 | 최현수 |
| 액터 | 관리자 |
| 사전 조건 | 사용자가 관리자 권한을 가진 계정으로 로그인한 상태여야 한다. |
| 사후 조건 | 게임 정보가 등록되거나 기존 정보가 수정되어 저장된다. |
| 트리거 | 관리자가 게임 정보 관리 메뉴에 진입한다. |

**2.1.10 관리자 - 시스템 로그 확인**

|  |  |  |  |
| --- | --- | --- | --- |
| 기본 흐름 | | | |
| 1 | 관리자 | | 관리자가 시스템 로그 확인 메뉴를 클릭한다. |
| 2 | 시스템(프론트엔드) | | 프론트엔드는 로그 요청을 전송한다. |
| 3 | 시스템(백엔드) | | 백엔드는 로그 데이터를 조회한다. |
| 4 | 시스템(DB) | | DB는 로그 정보를 반환한다. |
| 5 | 시스템(프론트엔드) | | 프론트엔드는 로그 리스트와 필터 옵션을 제공한다. |
| 6 | 관리자 | | 관리자가 로그를 확인하고 조건 검색을 수행한다. |
| 대안 흐름 | | | |
| 3A | 로그 없음 | "로그 데이터가 없습니다" 메시지를 출력한다. | |
| 4A | DB 연결 실패 | 재시도 버튼을 제공한다. | |
| 5A | UI 오류 | 텍스트 형태로 로그를 출력한다. | |

|  |  |
| --- | --- |
| **Use Case # 010** | |
| **GENERAL CHARACTERISTICS** | |
| Use Case ID | UC010 |
| 유스케이스 명 | 관리자 - 시스템 로그 확인 |
| 작성자 | 최현수 |
| 액터 | 관리자 |
| 사전 조건 | 사용자가 관리자 권한을 가진 계정으로 로그인한 상태여야 한다. |
| 사후 조건 | 로그 데이터가 조회되어 관리자에게 제공된다. |
| 트리거 | 관리자가 시스템 로그 확인 메뉴를 클릭한다. |

## 가정 및 의존사항 (Assumptions and Dependencies)

본 시스템이 정상적으로 작동하기 위해서는 다음과 같은 기술적 전제 조건 및 외부 의존 사항이 충족되어야 한다:

* **안정적인 네트워크 환경**  
   AI 모델은 외부 API(OpenAI, Hugging Face 등)와의 통신을 통해 작동하므로, 실시간 연결이 가능한 안정적인 인터넷 환경이 필수적이다.
* **소프트웨어 및 API 의존성**  
   시스템은 Python 기반의 실행 환경과 자연어 처리 라이브러리(spacy, transformers 등), 웹 프레임워크(fastapi 등)에 의존한다. 또한 외부 AI 모델은 클라우드 기반 API를 통해 호출되며, 해당 API 키 발급 및 사용량 정책에 따라 제약을 받을 수 있다.
* **입력 데이터의 정형화 필요성**  
   AI가 상황을 정확히 해석하기 위해, 사용자 입력은 사전에 정의된 형식 또는 구조(예: 명령문, 선택지, 질의응답 등)를 따르는 것이 바람직하다. 이를 위해 입력 전처리 모듈이 함께 동작한다.
* **게임 규칙(룰셋)의 명확성**  
   AI가 일관된 판단을 내리기 위해서는 게임 규칙, 조건, 예외 사항이 명확히 정의되어 있어야 하며, 불명확하거나 누락된 규칙은 시스템의 오작동 또는 비논리적 판단을 초래할 수 있다.
* **보안 및 데이터 보호 고려**  
   사용자 입력 및 처리 결과가 외부 AI API로 전송되는 구조이므로, HTTPS 기반의 암호화 통신 및 민감 정보 최소화를 포함한 기본적인 보안 정책이 적용되어야 한다.

# 시스템 기술 요구사항 (Specific Requirements)

본 장에서는 인공지능 기반 게임 추천 및 공략 제공 시스템이 갖추어야 할 주요 기술 요소와 동작 방식을 정의한다. 특히, 사용자 입력이 처리되어 AI 기반 판단 및 게임 관련 정보 제공으로 이어지는 전체 흐름을 중심으로 서술한다.

## AI 작동 방식 (AI Processing Logic)

AI가 사용자 입력에 대해 의미 있는 판단을 내리고 적절한 출력을 생성하기까지의 처리 흐름은 다음과 같다:

## 입력 데이터 수집 및 파싱 (Input Parsing)

사용자의 텍스트 입력을 수집하며, 게임 맥락(예: 직전 이벤트, 현재 상태, 플레이어 정보 등)을 함께 고려하여 AI 모델이 이해할 수 있는 구조로 전처리한다. 구체적으로:

* 사용자 검색어 및 필터 조건 파싱
* 사용자 프로필 데이터 로드 (선호 장르, 플레이 시간대, 난이도 선호도 등)
* 게임 플레이 이력 데이터 수집 (최근 플레이한 게임, 완료 여부, 플레이 시간 등)
* 만족도 평가 데이터 수집 (별점, 후기 등)

## 의도 분석 및 벡터화 (Preprocessing & Vectorization)

전처리된 입력은 자연어 처리 기법(NLP)을 활용해 구조화된 정보(예: 명령 유형, 대상 객체, 감정 상태 등)로 변환된다. 필요 시 텍스트 임베딩을 통해 벡터화하여 AI 모델의 입력 형식에 맞게 조정한다. 사용자 데이터 벡터화는 다음과 같이 이루어진다:

* 1. 게임 선호도 벡터화:.   
     • 장르별 플레이 시간을 로그 스케일로 정규화 (예: [액션: 0.7, RPG: 0.9, ...])

• 최근성 가중치 적용 (최근 플레이 게임에 더 높은 가중치 부여)  
• 시각적 스타일 선호도를 범주형 벡터로 변환 (예: [픽셀: 0.2, 3D: 0.7, 사실적: 0.4, ...])

* 1. 게임 플레이 이력 벡터화:  
     • 장르별 플레이 시간을 로그 스케일로 정규화 (예: [액션: 0.7, RPG: 0.9, ...])  
     • 최근성 가중치 적용 (최근 플레이 게임에 더 높은 가중치 부여)  
     • 완료율 정보 수치화 (완료: 1.0, 중간 포기: 플레이 진행률)
  2. 만족도 평가 벡터화:  
     • 별점을 0-1 사이 값으로 정규화  
     • 텍스트 리뷰를 감성 분석하여 -1에서 1 사이의 감성 점수 추출  
     • 플레이 시간 대비 만족도 비율 계산
  3. 통합 벡터 구성:  
     • 개별 벡터들을 가중치 합산하여 통합 사용자 프로필 벡터 생성  
     • 차원 축소 기법(PCA 등)을 통해 최적화된 표현 생성  
     • 콜드 스타트 문제 해결을 위한 기본 프로필 벡터와의 보간
* **프롬프트 구성 및 모델 호출 (Prompt Construction & API Call)**

시스템은 사전 정의된 프롬프트 형식에 따라 시스템 메시지(역할 및 기준 설명)와 사용자 입력을 조합해 외부 AI API(OpenAI GPT 등)에 요청을 보낸다. 호출은 RESTful API 방식으로 수행되며, 오류 발생 시 재시도 로직을 포함한다.  
  
프롬프트는 다음과 같은 구조를 가진다:  
  
시스템: 당신은 개인화된 게임 추천 및 공략 정보를 제공하는 AI 시스템입니다. 사용자의 게임 선호도와 플레이 이력을 기반으로 최적의 추천을 제공하세요.

사용자 프로필: {벡터화된 사용자 프로필 정보}

게임 이력: {벡터화된 게임 플레이 이력}

만족도 이력: {벡터화된 만족도 평가}

요청: {사용자 입력 또는 요청 유형}

* **응답 수신 및 해석 (Output Interpretation)**

모델로부터 수신한 응답은 JSON 또는 텍스트 형식으로 전달되며, 내부 파서가 이를 해석하여 의미적 요소(예: 위반 여부 판단, 조치 유형 등)를 추출한다. 응답은 다음과 같은 구조로 정형화된다:  
{

"recommendations": [

{

"game\_id": "game\_123",

"title": "게임 제목",

"genre": "장르",

"confidence": 0.92,

"reason": "사용자가 선호하는 RPG 요소와 최근 플레이한 전략 게임의 특성을 결합한 게임입니다."

},

...

],

"strategy": {

"type": "초보자 가이드",

"content": "게임 공략 내용...",

"difficulty": "중급"

}

}

* **행동 실행 및 로그 저장 (Action Execution & Logging)**

판단 결과에 따라 필요한 게임 이벤트(예: 추천 게임 표시, 공략 정보 제공, 사용자 프로필 업데이트 등)를 수행하며, 모든 결과는 판단 근거와 함께 로그로 저장된다. 이 로그는 추후 분석, 디버깅, 사용자 대응, 그리고 시스템 개선을 위한 피드백 루프에 활용된다.

로그 데이터는 다음과 같은 정보를 포함한다:

• 요청 시간 및 유형

• 입력 벡터의 주요 특성

• 모델 응답 및 신뢰도 점수

• 사용자 반응 및 피드백

• 실행 시간 및 리소스 사용량.

이렇게 수집된 로그는 시스템의 정확도와 효율성을 지속적으로 개선하는 데 활용된다.

## 기능적 요구사항 (Functional Requirements)

* 1. 추천 정제 기능  
     사용자의 피드백을 바탕으로 AI 추천을 개선함
  2. 알림 시스템

새로운 추천, 업데이트 등을 사용자에게 알림

* 1. 다국어 지원   
     영어, 한국어, 일본어 인터페이스 및 콘텐츠 제공
  2. 데이터 내보내기/ 가져오기  
     CSV 내보내기 및 Stream 데이터 가져오기 지원
  3. 사용자 설정 맞춤화   
     추천 알고리즘 및 UI 테마 등 사용자 설정 기능 제공
  4. 관리자 분석 대시보드   
     사용자 통계 및 추천 정확도 등을 시각화하여 관리자에게 제공

## 비기능적 요구사항 (Non-Functional Requirements)

1. 성능   
   빠른 API응답 및 페이지 로드 보장
2. 보안   
   데이터 암호화, 로그인 보호, 개인정보 법규 준수
3. 확장성   
   동시 사용자 및 API 호출 증가에도 안정적인 성능 유지
4. UI/UX   
   반응형 디자인 및 일관된 사용자 인터페이스 제공
5. 접근성  
   장애 사용자 지원(WCAG2.1 AA 기준 준수)
6. 시스템 안정성  
   99.9% 가동률 및 오류 복구 시스템 구현

# 설계 관련 정보 (Design Information)

## 4.1 시스템 아키텍처 (Architecture Diagram) 4.1.1 언어 및 프레임워크

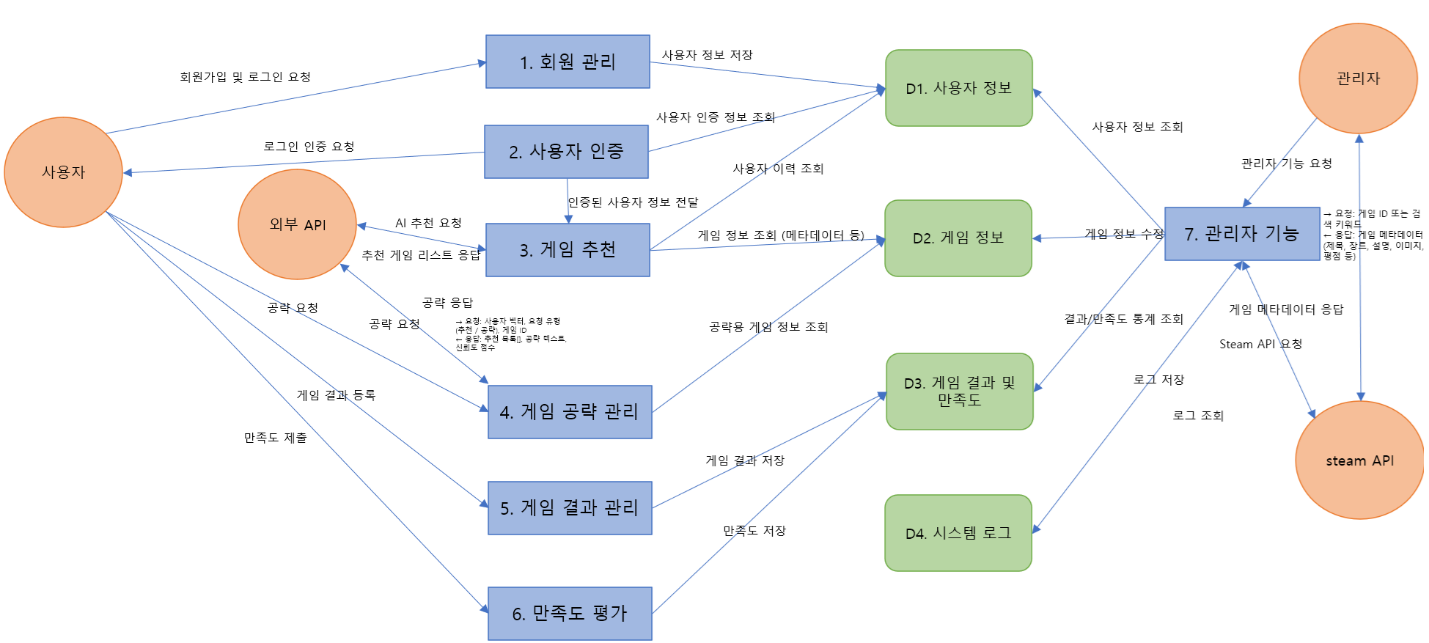
* **프론트엔드:**
* **프레임워크: React/Vue 기반 웹 UI**
* **언어: JavaScript**
* **백엔드:**
* **프레임워크: Node.js + Express 또는 Spring 기반 API 서버**
* **언어: JavaScript(Node.js) or Java(Spring)**
* **AI 처리: Python**
* **데이터베이스:**
* **DB: MySQL8.0 사용**
* **ORM:Sequelize(Node.js) or JPA(Spring) 활용**
* **외부 API:**
* **OpenAI ChatGPT API: Python 기반 호출 모듈 사용**
* **Steam API: RESTful API 호출  
  4.1.2 운영체제**
* **서버:**
* **백엔드 서버: Linux(Ubuntu 22.04 LTS)에서 실행**
* **클라이언트:**
* **웹 클라이언트: Windows 10.11, macOS 12+, Linux에서 실행 가능한 주요 브라우저 지원  
  4.1.3 통신**
* 프로토콜:
* 프론트엔드 ↔ 백엔드: HTTPS 기반 RESTful API 사용
* HTTPS: 데이터 암호화 및 보안 통신 보장(SSL/TLS 적용)
* 백엔드 ↔ 외부 API: RESTful API호출(OpenAI,Steam API)
* 알림 시스템: WebSocket 프로토콜 사용하여 실시간 알림 전송
* 보안:
* 모든 통신은 HTTPS를 통해 암호화
* API 요청시 API키 인증 및 토큰 기반 인증(JWT) 사용
* 민감 데이터(사용자 정보 등)는 최소화하여 전송

## 4.2 시스템 구성도

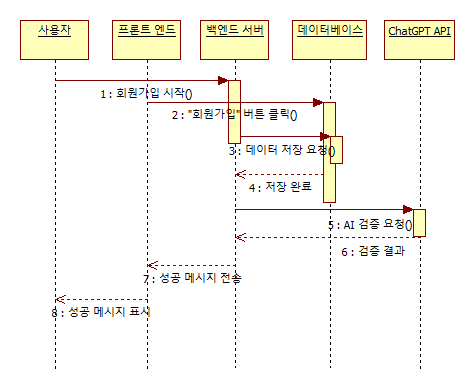
## 4.3 데이터 흐름도 (Data Flow Diagram - DFD)

아래 데이터 흐름도는 본 시스템에서 사용자의 입력부터 AI를 통한 게임 추천 및 공략 제공, 결과 저장, 관리자 기능까지의 주요 흐름을 표현한다.

외부 API(OpenAI 등)와의 연동을 포함하여, 각 기능 모듈의 데이터 이동 경로 및 주요 저장소 간의 상호작용을 시각적으로 나타낸다.



## 4.4 SequenceDiagram 상세화

4.4.1 회원가입   
  
  
 **1: 회원가입 시작**: 사용자가 회원가입 프로세스를 시작하며 시스템에 접근.

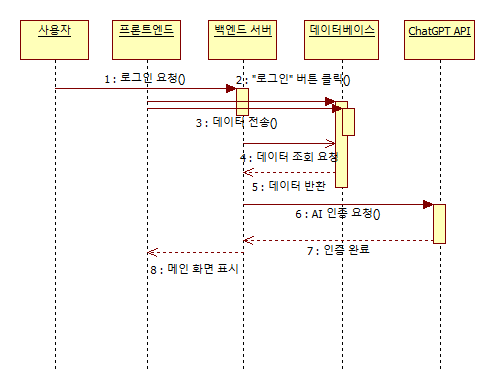
 **2: "회원가입" 버튼 클릭**: 사용자가 데이터를 입력하고 버튼을 클릭해 프론트엔드로 요청.

 **3: 데이터 저장 요청**: 프론트엔드가 데이터를 백엔드로 보내고, 백엔드가 데이터베이스에 저장 요청.

 **4: 저장 완료**: 데이터베이스가 데이터를 저장하고 백엔드에 완료 알림.

 **5: AI 검증 요청**: 백엔드가 ChatGPT API에 데이터 검증 요청.

 **6: 검증 결과**: ChatGPT API가 검증 결과를 백엔드로 반환.

 **7: 성공 메시지**: 백엔드가 프론트엔드에 성공 메시지 전달.   
 **8: 메시지 표시**: 프론트엔드가 사용자에게 성공 메시지 표시.  
  
  
  
  
  
  
  
  
  
4.4.2 로그인  
  
  
  
 **1: 로그인 요청**: 사용자가 로그인 프로세스를 시작하며 시스템에 접근.

 **2: "로그인" 버튼 클릭**: 사용자가 아이디와 비밀번호를 입력하고 버튼을 클릭해 프론트엔드로 요청.

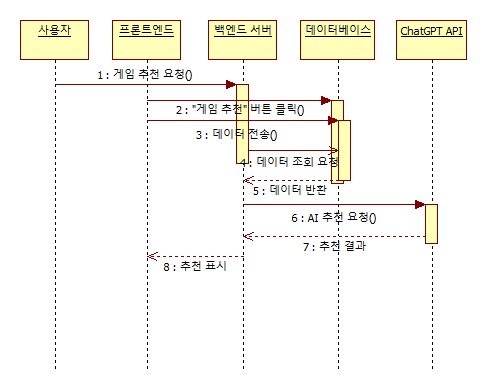
 **3: 데이터 전송 (아이디, 비밀번호)**: 프론트엔드가 입력 데이터를 백엔드 서버로 전송.

 **4: 데이터 조회 요청**: 백엔드 서버가 데이터베이스에 사용자 데이터를 조회 요청.

 **5: 데이터 반환**: 데이터베이스가 사용자 데이터를 백엔드 서버로 반환.

 **6: AI 인증 요청**: 백엔드 서버가 ChatGPT API에 인증 요청.

 **7: 인증 완료**: ChatGPT API가 인증 결과를 백엔드 서버로 반환.

 **8: 메인 화면 표시**: 백엔드 서버가 프론트엔드에 성공 메시지를 보내 메인 화면 표시.  
  
  
  
  
  
  
  
4.4.3 게임 추천  
  
  
  
 **1: 게임 추천 요청**: 사용자가 게임 추천 기능을 시작하며 시스템에 접근.

 **2: "게임 추천" 버튼 클릭**: 사용자가 버튼을 클릭해 프론트엔드로 요청.

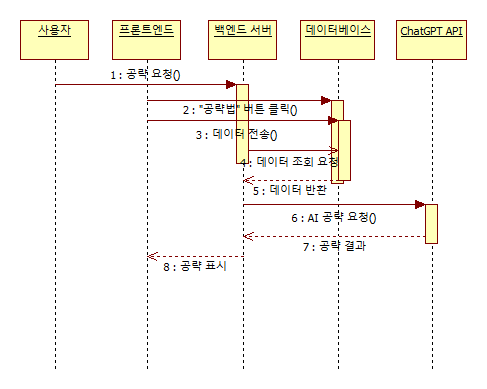
 **3: 데이터 전송 (사용자 ID)**: 프론트엔드가 사용자 ID를 백엔드 서버로 전송.

 **4: 데이터 조회 요청**: 백엔드 서버가 데이터베이스에 사용자 데이터를 조회 요청.

 **5: 데이터 반환**: 데이터베이스가 사용자 데이터를 백엔드 서버로 반환.

 **6: AI 추천 요청**: 백엔드 서버가 ChatGPT API에 개인화된 추천 요청.

 **7: 추천 결과**: ChatGPT API가 추천 결과를 백엔드 서버로 반환.

 **8: 추천 표시**: 백엔드 서버가 프론트엔드에 결과를 보내 사용자에게 표시.  
  
  
  
  
  
  
  
  
  
4.4.4 공략법 요청  
  
  
  
 **1: 공략 요청**: 사용자가 공략법 요청을 시작하며 시스템에 접근.

 **2: "공략법" 버튼 클릭**: 사용자가 버튼을 클릭해 프론트엔드로 요청.

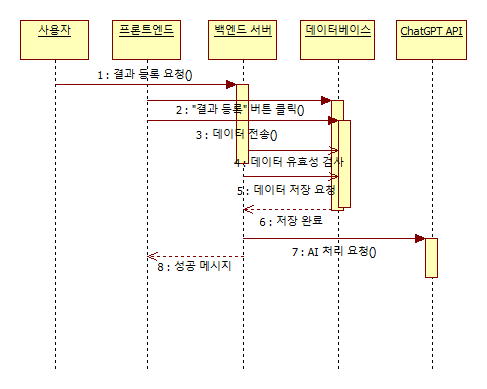
 **3: 데이터 전송 (게임 ID)**: 프론트엔드가 게임 ID를 백엔드 서버로 전송.

 **4: 데이터 조회 요청**: 백엔드 서버가 데이터베이스에 게임 데이터를 조회 요청.

 **5: 데이터 반환**: 데이터베이스가 게임 데이터를 백엔드 서버로 반환.

 **6: AI 공략 요청**: 백엔드 서버가 ChatGPT API에 공략 생성 요청.

 **7: 공략 결과**: ChatGPT API가 공략 결과를 백엔드 서버로 반환.

 **8: 공략 표시**: 백엔드 서버가 프론트엔드에 결과를 보내 사용자에게 표시.  
  
  
  
  
  
  
  
  
4.4.5 게임 결과 등록  
  
  
  
 **1: 결과 등록 요청**: 사용자가 결과 등록 기능을 시작하며 시스템에 접근.

 **2: "결과 등록" 버튼 클릭**: 사용자가 버튼을 클릭해 프론트엔드로 요청.

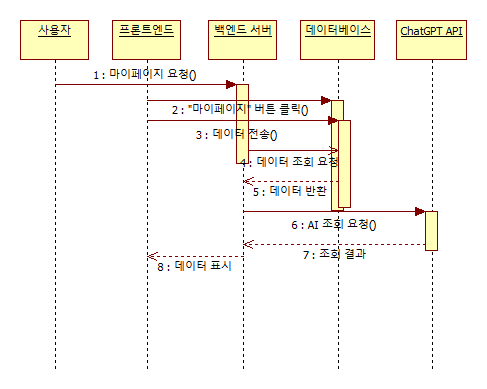
 **3: 데이터 전송 (게임명, 플레이 시간)**: 프론트엔드가 데이터를 백엔드 서버로 전송.

 **4: 데이터 유효성 검사**: 백엔드 서버가 데이터의 유효성을 검사.

 **5: 데이터 저장 요청**: 백엔드 서버가 데이터베이스에 데이터를 저장 요청.

 **6: 저장 완료**: 데이터베이스가 저장을 완료하고 백엔드에 알림.

 **7: AI 처리 요청**: 백엔드 서버가 ChatGPT API에 추가 처리 요청.

 **8: 성공 메시지**: 백엔드 서버가 프론트엔드에 성공 메시지 전달.  
  
  
  
  
  
  
  
  
4.4.6 마이페이지 조회  
  
  
  
 **1: 마이페이지 요청**: 사용자가 마이페이지 조회를 시작하며 시스템에 접근.

 **2: "마이페이지" 버튼 클릭**: 사용자가 버튼을 클릭해 프론트엔드로 요청.

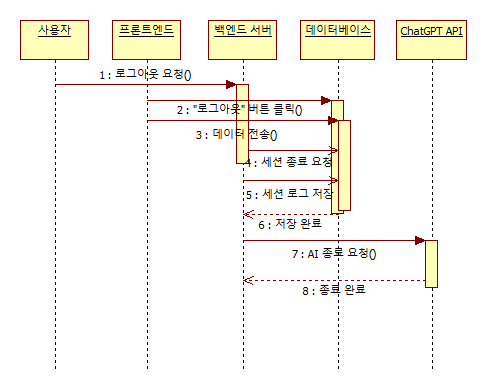
 **3: 데이터 전송 (사용자 ID)**: 프론트엔드가 사용자 ID를 백엔드 서버로 전송.

 **4: 데이터 조회 요청**: 백엔드 서버가 데이터베이스에 사용자 데이터를 조회 요청.

 **5: 데이터 반환**: 데이터베이스가 사용자 데이터를 백엔드 서버로 반환.

 **6: AI 조회 요청**: 백엔드 서버가 ChatGPT API에 개인화된 조회 요청.

 **7: 조회 결과**: ChatGPT API가 조회 결과를 백엔드 서버로 반환.

 **8: 데이터 표시**: 백엔드 서버가 프론트엔드에 결과를 보내 사용자에게 표시.  
  
  
  
  
  
  
  
4.4.7 로그아웃 요청  
  
  
  
 **1: 로그아웃 요청**: 사용자가 로그아웃을 시작하며 시스템에 접근.

 **2: "로그아웃" 버튼 클릭**: 사용자가 버튼을 클릭해 프론트엔드로 요청.

 **3: 데이터 전송 (세션 ID)**: 프론트엔드가 세션 ID를 백엔드 서버로 전송.

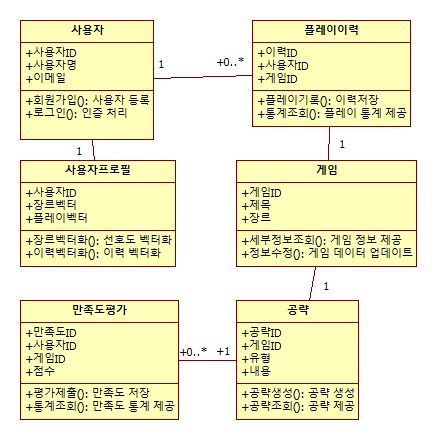
 **4: 세션 종료 요청**: 백엔드 서버가 세션을 종료 요청.

 **5: 세션 로그 저장**: 백엔드 서버가 데이터베이스에 세션 로그를 저장 요청.

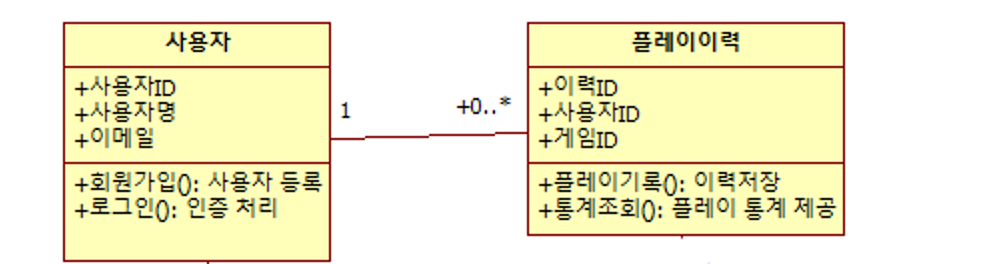
 **6: 저장 완료**: 데이터베이스가 로그 저장을 완료하고 백엔드에 알림.

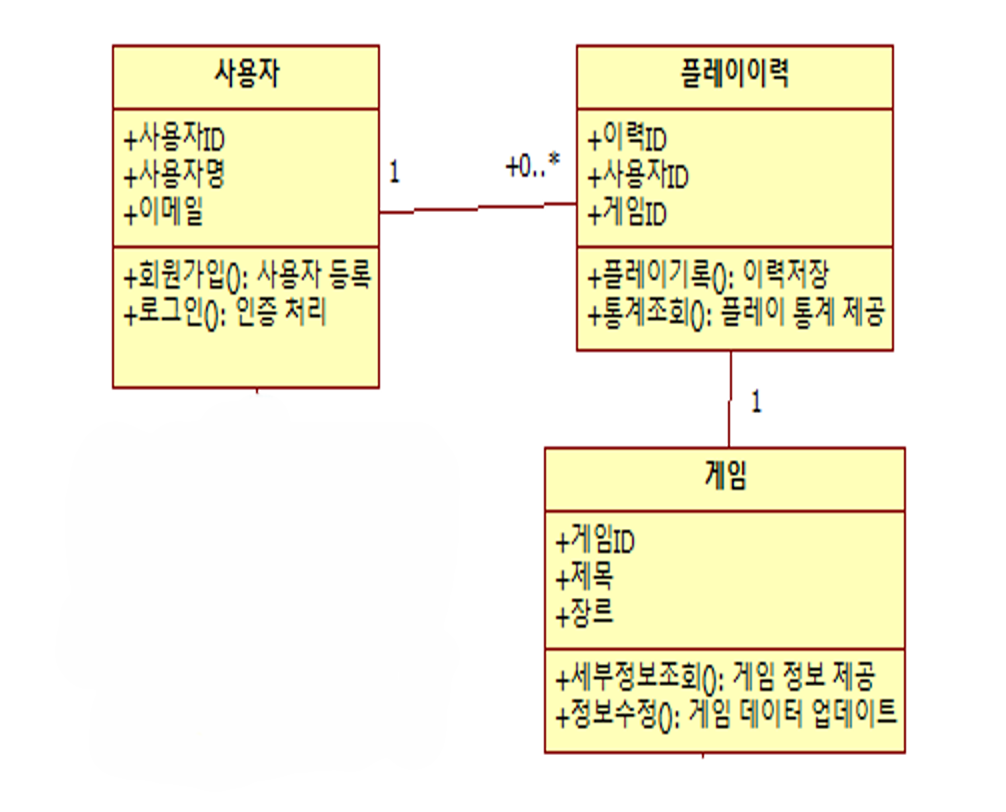
 **7: AI 종료 요청**: 백엔드 서버가 ChatGPT API에 종료 요청.   
 **8: 종료 완료**: ChatGPT API가 종료 결과를 백엔드 서버로 반환.

## 4.5 클래스 다이어그램 (Class Diagram)

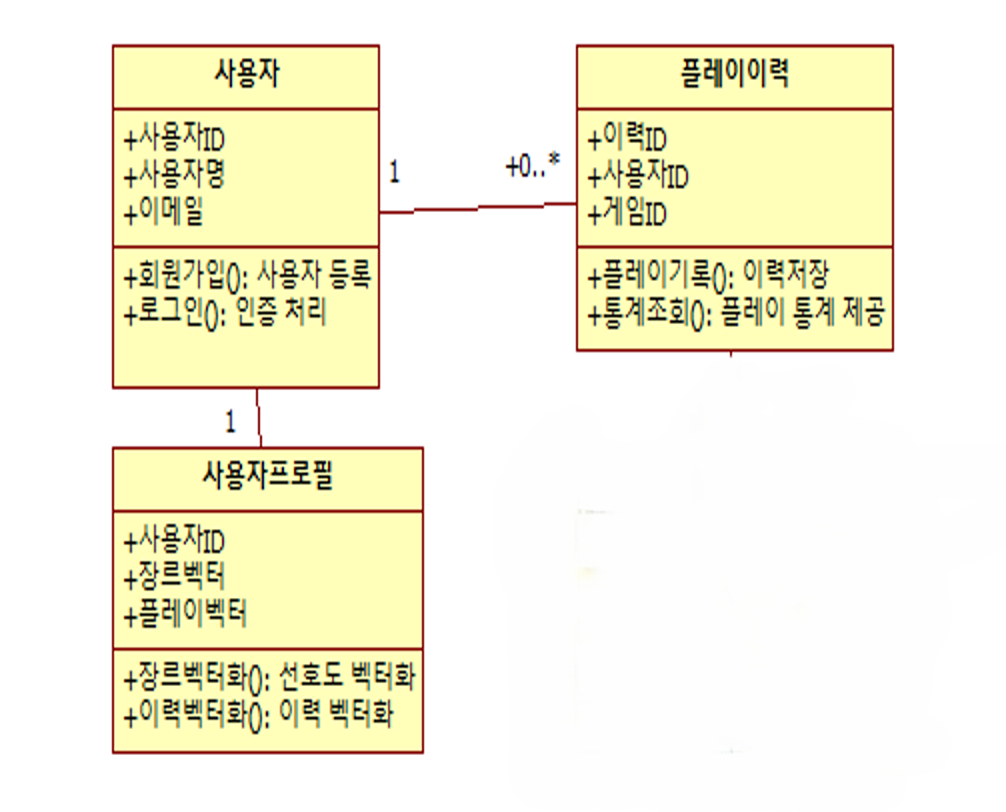


4.5.1클래스 다이어그램 세분화

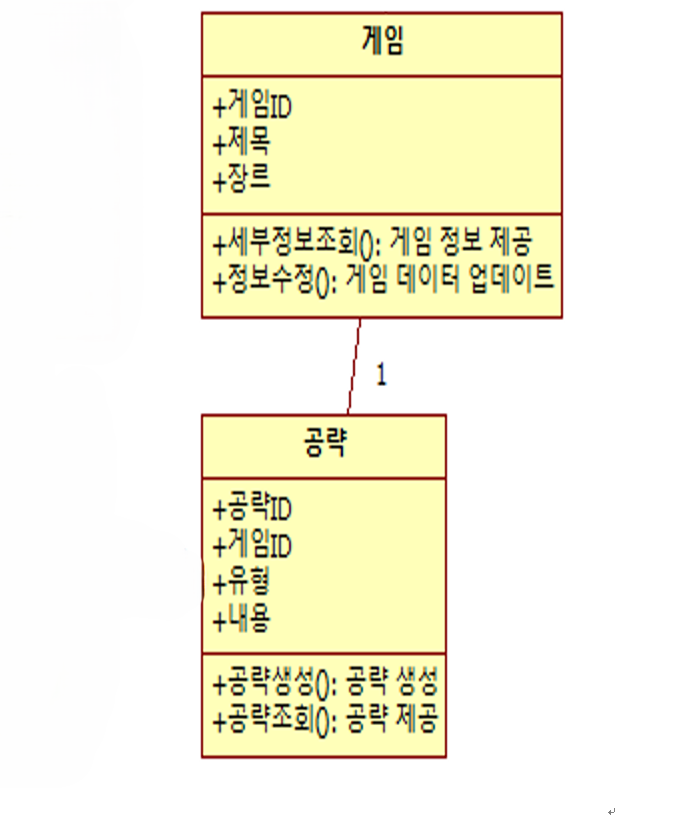
4.5.2 플레이 이력  


4.5.3 게임정보  


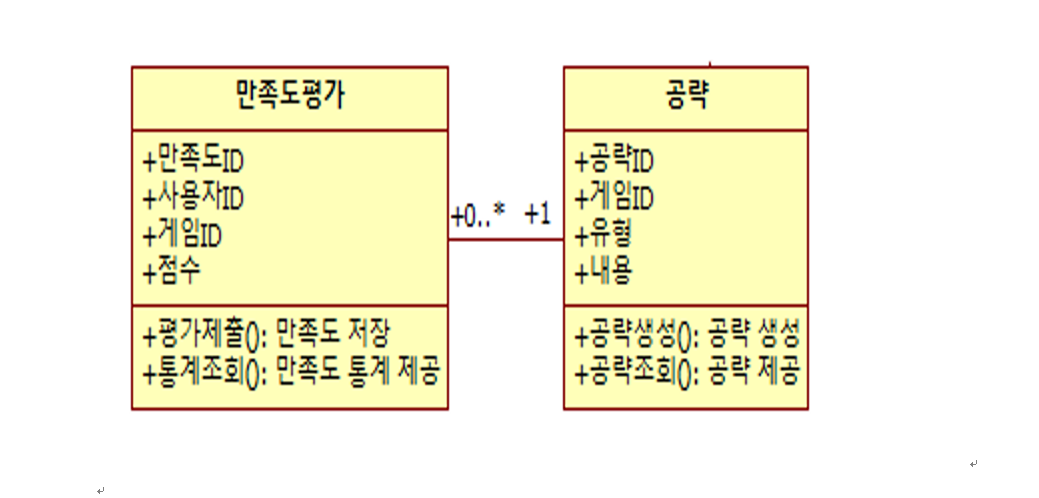
4.5.4 사용자프로필



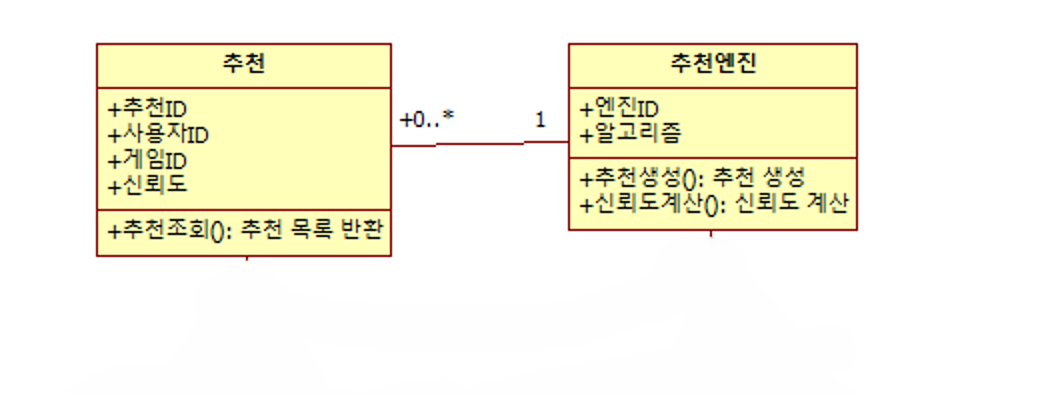
4.5.5 공략



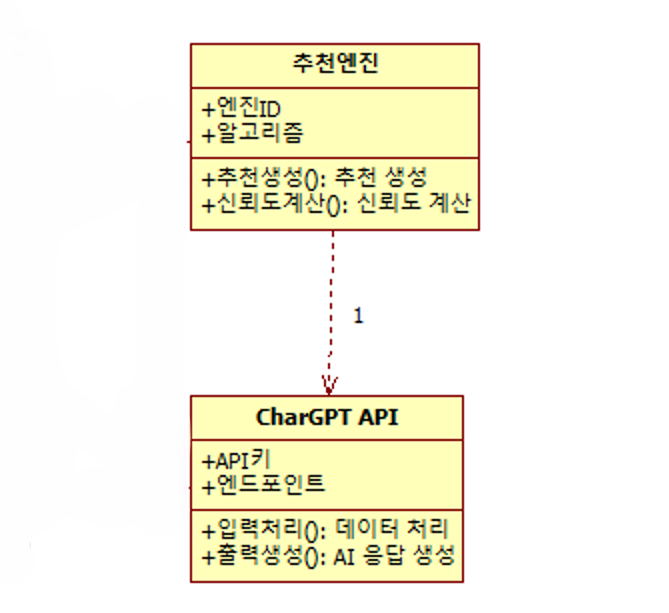
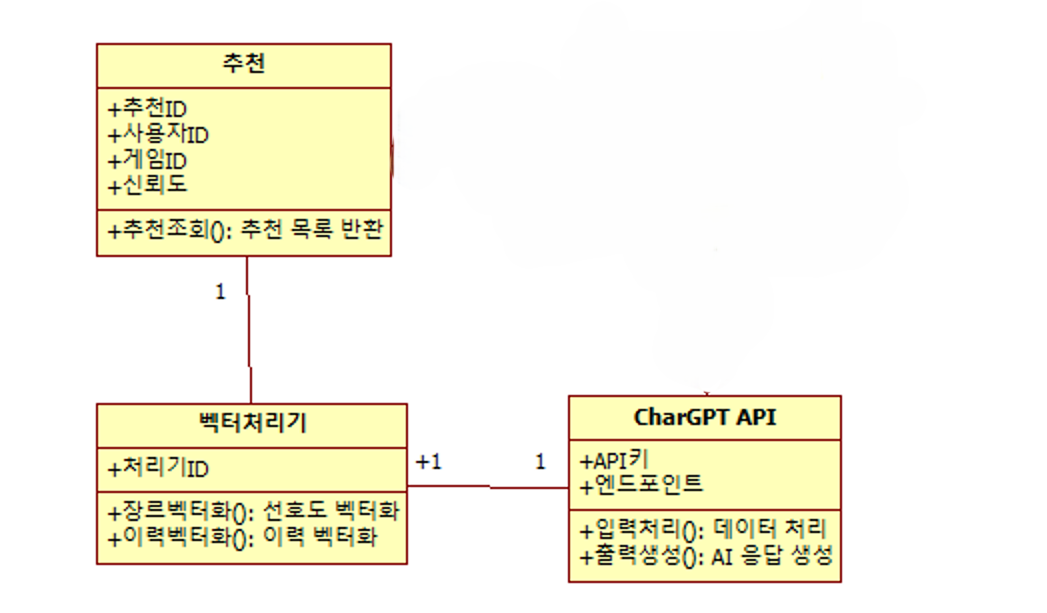
4.5.6 만족도평가

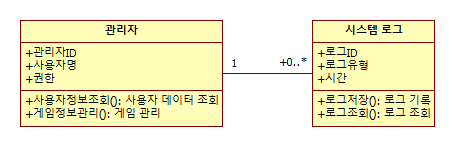


4.5.7 추천엔진

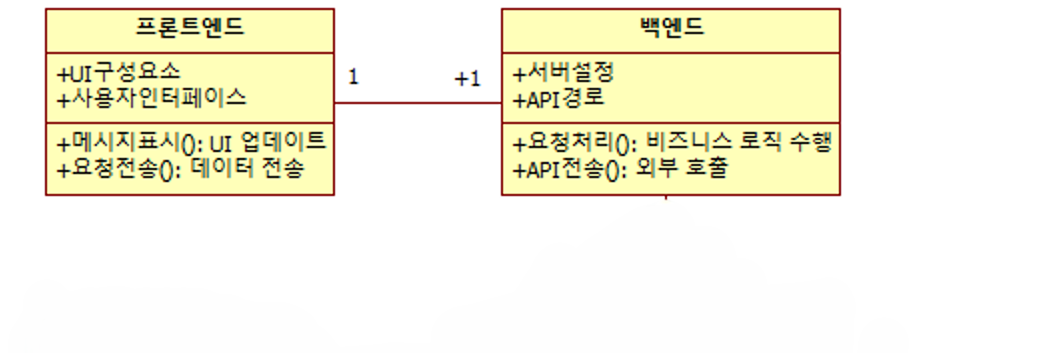


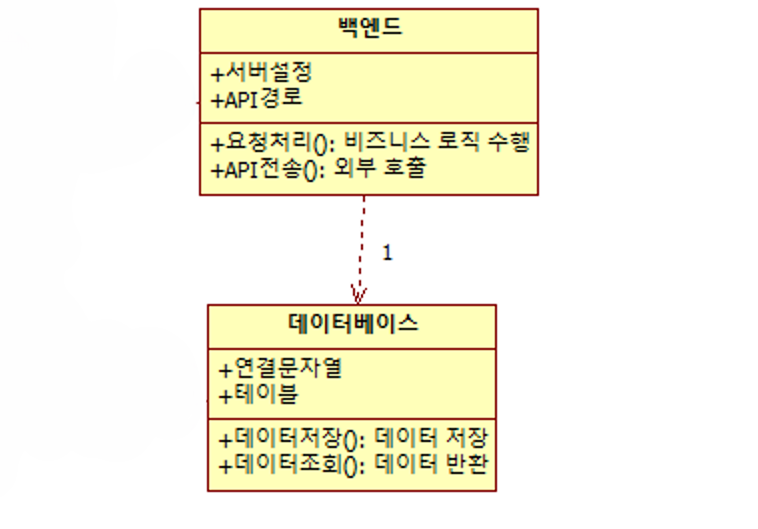
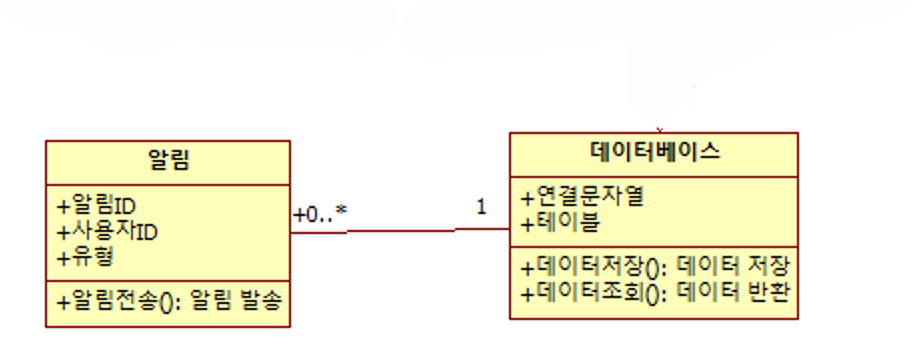
4.5.8 CharGPT API

  
4.5.9 벡터처리기  
  
4.5.10 시스템 로그



4.5.11 백엔드



4.5.12 데이터베이스  
  
4.5.13 알림  


## 4.6 UI

## 4.7 간트차트

# 지원 정보 (Supporting Information)

* **UI 시안 및 설계**
  + 로그인/회원가입 화면
  + 게임 추천 화면
  + 공략 정보 제공 화면
  + 마이페이지
  + 관리자 대시보드
* **개발 환경 및 도구**
  + IDE: Visual Studio Code, IntelliJ
  + 프론트엔드: React.js
  + 백엔드: Node.js (Express) / Spring Boot (선택)
  + DBMS: MySQL
  + API 연동: OpenAI API, Steam API
* **테스트 환경**
  + 로컬 환경: Windows 11 / macOS Ventura
  + 클라우드 서버: AWS EC2 (Ubuntu 22.04)
* **설계 자료**
  + 유스케이스 다이어그램
  + 시퀀스 다이어그램
  + 클래스 다이어그램
  + 데이터 흐름도
* **기타**
  + 데이터베이스 테이블 정의서
  + API 명세서 (Swagger 문서 등)

# 참고 문헌

1 클래스 다이어그램 <https://gmlwjd9405.github.io/2018/07/04/class-diagram.html>

2 시퀀스 다이어그램 https://steady-hello.tistory.com/132

3 유스케이스 다이어그램 https://devjaewoo.tistory.com/15