

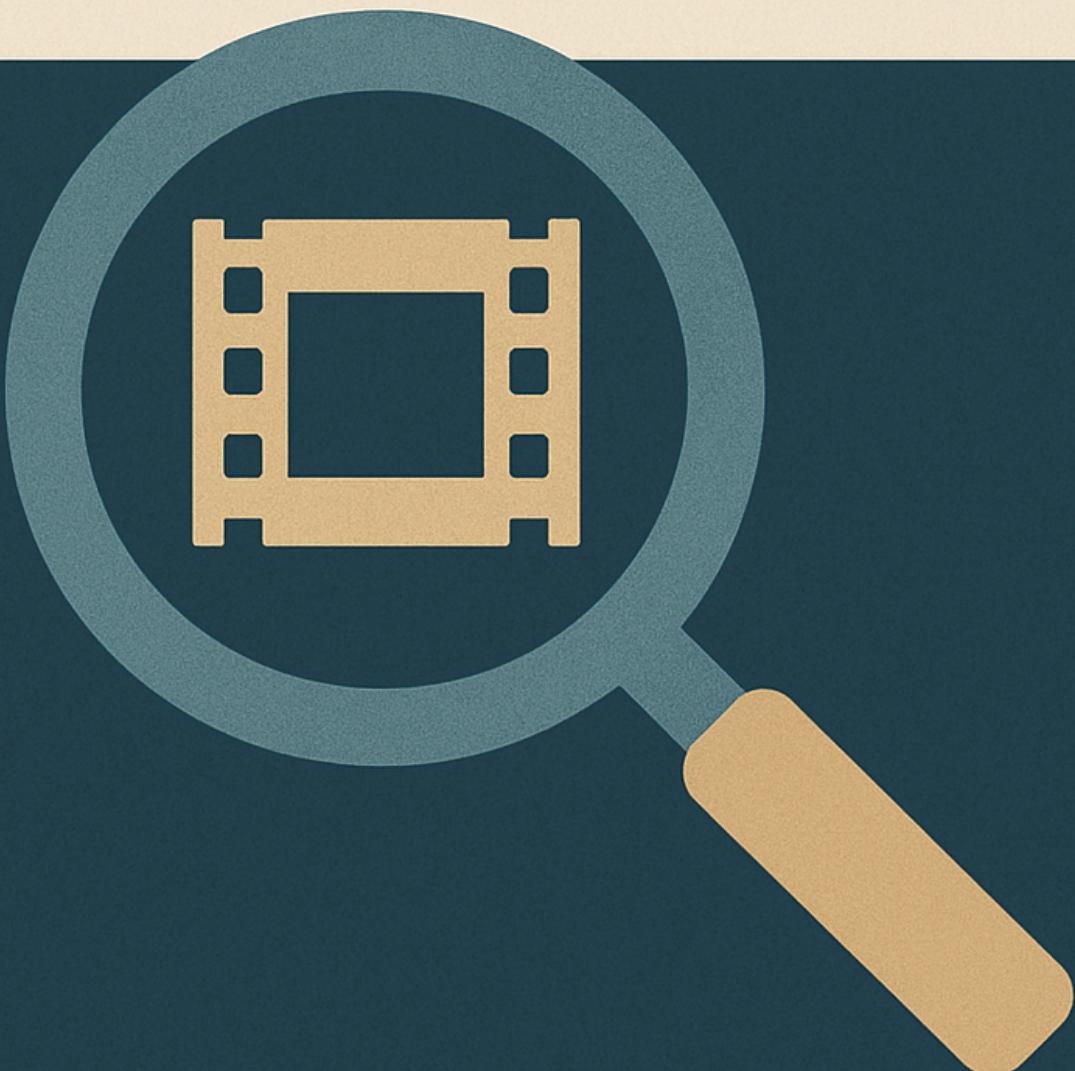
# Requirements Specification

MOVIE CHAT

강병희, 박진우, 박찬원, 이상훈, 이해민

소프트웨어공학개론(SWE3002-41)

성균관대학교 교수 이은석



# Software Requirements Specification

– MovieChat –

by

강병희, 박진우, 박찬원, 이상훈, 이해민

Team 6

Instructor	이은석
Teaching Assistant	김진영, 최동욱, 허진석, 김영경
Document Date	2025. 5. 10.
Faculty	SungKyunKwan University

# Contents

1. Introduction.....	1
1.1 Purpose .....	1
1.2 Scope .....	1
1.3 Definitions, acronyms, and abbreviations.....	1
1.4 References.....	1
1.5 Overview .....	2
2. Overall Description .....	3
2.1 Product Perspective.....	3
2.1.1 System Interfaces .....	3
2.1.2 User Interfaces .....	3
2.1.3 Hardware Interfaces.....	3
2.1.4 Software Interfaces .....	4
2.1.5 Memory Constraints .....	4
2.1.6 Operations.....	4
2.2 Product Functions .....	4
2.2.1 영화 추천 및 대화형 토론.....	4
2.2.2 리뷰 요약 및 영화 정보 시작화 .....	4
2.2.3 북마크 및 아카이빙 .....	4
2.2.4 영화 인물 기반 물입형 대화.....	4
2.3 User Classes and Characteristics .....	5
2.3.1 System Administrator .....	5
2.3.2 General Users.....	5
2.4 Design and Implementation Constraints.....	5
2.5 Assumptions and Dependencies.....	5
3. External Interface Requirements .....	6
3.1 External Interfaces Requirements.....	6
3.1.1 User Interface .....	6
3.1.2 Hardware Interface .....	7
3.1.3 Software Interface.....	7
3.1.4 Communication Interface.....	7
3.2 Functional Requirement .....	8
3.2.1 Use Cases .....	8
3.2.2 Use Case Diagram.....	11
3.2.3 Data Dictionary .....	12

3.2.4 Data Flow Diagram.....	16
3.3 Performance Requirements.....	16
3.3.1 Static Numerical Requirements.....	16
3.3.2 Dynamic Numerical Requirements.....	17
3.4 Logical Database Requirements.....	19
3.4.1 데이터베이스 개요 .....	17
3.4.2 저장되는 주요 데이터 .....	18
3.4.3 데이터베이스 구조 .....	18
3.4.4 데이터베이스 제약 사항, 무결성 및 보안 .....	19
3.5 Design Constraints.....	19
3.5.1 Physical Design Constraints .....	19
3.5.2 Standard Compliance .....	19
3.6 Software System Characteristics.....	20
3.6.1 Product Requirements .....	20
3.6.2 Organizational Requirements .....	21
3.6.3 External Requirements .....	21
3.7 Organizing the Specific Requirements .....	22
3.7.1 Context Model .....	22
3.7.2 Process Model .....	23
3.7.3 Interaction Model .....	23
3.8 System Architecture .....	24
3.9 System Evolution .....	24
3.9.1 Limitation and Assumption .....	25
3.9.2 Evolution of Hardware and Change of User Requirements .....	25
4. Supporting Information .....	26
4.1 Software Requirements Specification .....	26
4.2 Document History .....	26

# 1. Introduction

## 1.1 Purpose

이 문서는 LLM 기반 영화 검색 및 추천 서비스를 제공하는 플랫폼 "MovieChat"의 요구사항을 명확히 정의하고, 향후 디자인 명세서 작성 및 개발, 유지보수 과정에서 참조하기 위해 작성되었다. 또한, 이 문서는 소프트웨어 개발자, 시스템 아키텍트, 프로젝트 관리자, QA 엔지니어, 서비스 기획자 및 클라이언트 등 MovieChat 플랫폼의 개발과 운영에 관련된 관계자를 대상으로 한다.

## 1.2 Scope

MovieChat은 LLM 기반의 지능형 영화 검색 및 추천 플랫폼으로, 기존의 키워드 중심 검색 시스템과 차별화된다. 본 소프트웨어는 사용자의 자연어 입력을 이해하여, 사용자 상태와 맥락에 적합한 영화를 추천하고, 해당 영화에 대한 해석 및 심화 토론 내용을 제공하는 기능을 수행한다.

또한, 사용자는 관심 있는 영화를 북마크하거나, 시청 이력을 기반으로 개인화된 추천을 받을 수 있으며, 영화 등장인물과의 몰입형 대화를 통해 더욱 풍부한 사용자 경험을 제공받을 수 있다. 이러한 대화는 시각적 정보와 연계되어 이해를 돋운다.

본 제품은 영화 검색 및 추천, 대화형 해석 제공, 사용자 맞춤형 인터랙션을 포함하며, 일반적인 영화 스트리밍 기능(영상 재생 등)은 제공하지 않는다.

이 플랫폼은 사용자에게 더 직관적이고 몰입감 있는 영화 탐색 경험을 제공함으로써, 영화 감상 및 이해의 깊이를 확장하는 것을 목표로 한다.

## 1.3 Definitions, Acronyms, and Abbreviations

Terms	Definitions
감성 큐레이션	사용자의 감정에 맞춰 콘텐츠를 추천하는 기능
몰입형 대화	가상 인물 또는 영화 등장인물과 상호작용하는 대화 방식
북마크	관심 있는 영화를 저장해두는 기능
아카이빙	시청한 영화 기록을 저장 및 관리하는 기능
TMDB	영화 및 TV 프로그램 정보 제공 API 서비스

Table 1.1: table of terms and definition

## 1.4 References

IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications

TMDB API 공식 문서 <https://developer.themoviedb.org/docs>

Wikipedia: 영화 및 등장인물 정보 활용

## 1.5 Overview

본 요구사항 명세서는 MovieChat 서비스의 개발을 위해 필요한 요구사항을 정의한다.

1장에서는 MovieChat 프로젝트의 목적과 범위, 주요 용어를 설명하며,

2장에서는 시스템 구성 요소와 제품의 전반적인 특성을 소개한다.

3장에서는 구체적인 외부 인터페이스, 기능, 성능 요건을 상세히 기술하고,

4장에서는 문서 작성 이력을 포함하여 추가 정보를 제공한다.

# 2. Overall Description

## 2.1 Product Perspective

MovieChat은 LLM(Large Language Model) 기술을 활용해 사용자와 자연스러운 대화를 주고받으며 영화를 검색하고 추천하는 대화형 영화 검색 플랫폼이다. 사용자는 단순한 키워드가 아닌 자신의 감정, 상황, 취향 등을 자연어로 표현할 수 있으며, 시스템은 이를 이해하고 적절한 영화를 찾아 제공한다. 이를 통해 기준의 정직하고 제한적인 검색 방식보다 더 편리하고 자연스러운 영화 탐색 경험을 제공한다.

추가적으로, 영화 속 등장인물과의 대화 기능을 통해, 다양한 관점에서 영화를 이야기하고 토론할 수 있다.

MovieChat은 웹 브라우저를 통해 바로 이용 가능하며, 외부의 방대한 영화 정보 데이터와 연동하여 풍부하고 신뢰할 수 있는 콘텐츠를 제공한다.

### 2.1.1 System Interfaces

---

사용자는 웹 기반 인터페이스를 통해 자연어로 영화 추천 및 대화를 요청할 수 있다.

시스템은 입력된 사용자 메시지를 LLM에 전달하고, LLM이 적절한 추천 및 대화 응답을 생성한다.

영화 메타데이터(포스터, 줄거리, 출연진 등)는 TMDB API 또는 위키피디아 크롤링을 통해 수집된다.

추천 및 대화 결과는 서버를 통해 클라이언트에 전달된다.

### 2.1.2 User Interfaces

---

사용자는 자연어 입력창을 통해 자유롭게 메시지를 입력한다.

북마크 및 아카이빙 영화 목록, 대화창(챗봇 인터페이스), 영화 상세 정보 페이지가 제공된다.

사용자는 북마크 및 아카이브 버튼을 통해 영화를 아카이빙하고 ‘좋아요’ 및 ‘싫어요’ 버튼으로 자신의 취향을 입력할 수 있다. 또한, 영화 추천 시 별도의 영화 상세 정보 창이 생성되게 된다. 사용자는 해당 창을 통해 영화에 대한 정보를 수집할 수 있으며, 영화를 선택 시 더 자세한 정보를 열람할 수 있는 상세 정보 창이 생성된다.

### 2.1.3 Hardware Interfaces

---

사용자는 PC를 통해 웹 브라우저로 MovieChat에 접속한다.

웹 서비스이기 때문에, 인터넷에 연결된 환경이어야 하고, 웹 브라우저를 실행할 수 있는 하드웨어이어야 한다.

## 2.1.4 Software Interfaces

---

최소 사양: Windows 10 이상, MacOS 10.14 이상, Android 10 이상, iOS 14 이상

권장 브라우저: Chrome 100 이상, Safari 14 이상, Edge 100 이상, Firefox 100 이상

서버는 FastAPI 기반으로 동작하며, 데이터베이스는 MySQL을 사용한다.

영화 데이터는 TMDB API를 통해 실시간 조회한다.

대화형 응답은 OpenAI API 또는 자체 Fine-tuned LLM 서버를 통해 생성된다.

## 2.1.5 Memory Constraints

---

클라이언트 단에서는 특별한 메모리 제약이 없다.

서버에서는 LLM 인퍼런스와 다수 사용자 요청을 처리하기 위해 최소 16GB RAM 이상의 서버를 요구한다.

## 2.1.6 Operations

---

사용자는 로그인 없이 체험할 수 있으나, 북마크 및 개인 아카이빙 기능을 사용하려면 계정 생성 및 로그인이 필요하다. 서버는 사용자별 대화 히스토리, 북마크 목록, 아카이빙 기록을 관리한다. 관리자 모드는 사용자 로그 분석, API 상태 모니터링, 추천 로직 튜닝 기능을 포함한다.

# 2.2 Product Functions

## 2.2.1 영화 추천 및 대화형 토론

---

사용자가 자연어로 입력한 질의에 따라 관련 영화를 추천하거나, 특정 영화에 대해 챗봇과 자유롭게 토론할 수 있는 기능이다. 영화의 주제, 분위기, 장르 등에 기반해 적절한 영화를 추천하며, 추천된 영화에 대한 리뷰 요약, 해석, 논의 등을 지원한다.

## 2.2.2 리뷰 요약 및 영화 정보 시작화

---

왓챠피디아 등 외부 전문 리뷰 사이트에서 수집한 텍스트 기반 리뷰를 요약해 제공한다. 영화의 포스터, 줄거리, 예고편, 배우 및 감독 정보, 흥행 지표 등을 정리하여 사용자에게 제공한다.

## 2.2.3 북마크 및 아카이빙

---

사용자가 관심 있는 영화를 북마크하거나, 시청한 영화를 자동으로 아카이빙할 수 있도록 한다. 북마크된 영화는 추천 시스템에서 반영되며, 대화 중 언급된 영화도 자동으로 북마크에 추가될 수 있다.

## 2.2.4 영화 인물 기반 몰입형 대화

---

영화 속 등장인물을 바탕으로 한 캐릭터 챗봇과의 몰입형 대화 기능을 제공한다. 위키피디아 등의 외부 자료를 기반으로 인물의 배경과 성격을 재현하며, 대화형 인터페이스로 사용자와 상호작용한다.

## 2.3 User classes and characteristics

### 2.3.1 System Administrator

---

MovieChat의 운영 및 유지보수를 담당하는 역할이다. Python, FastAPI, MySQL, 벡터 DB 및 프론트엔드 프레임워크에 대한 이해가 필요하며, LLM 기반 시스템의 디버깅 및 설정 조정 업무를 수행할 수 있어야 한다.

### 2.3.2 General Users

---

서비스의 주요 대상은 영화에 관심 있는 일반 사용자이다. 자연어로 영화에 대한 질문을 하고, 추천을 받으며, 캐릭터와의 대화를 통해 몰입형 경험을 즐긴다. 특별한 기술 지식 없이도 사용할 수 있도록 직관적인 UI를 제공해야 한다.

## 2.4 Design and implementation constraints

- 사용자 경험을 위한 브라우저 호환성 확보 필요 (최신 Chrome, Edge, Firefox, Safari 기준).
- 실시간 API 호출(TMDB, OpenAI 등) 시 요청 지연(latency) 최소화 필요.
- 리뷰 및 영화 정보 크롤링 시 저작권 및 데이터 사용 정책 준수

## 2.5 Assumptions and dependencies

- 사용자는 자연어로 영화 관련 질의를 입력할 수 있다고 가정한다.
- 영화 데이터는 TMDB API 및 위키피디아 등의 외부 소스에 의존하며, 해당 API의 정상적인 응답을 가정한다.
- 추천/요약/대화 기능은 LLM 및 자체 임베딩 기반 검색 시스템의 정상 작동을 전제로 한다.
- LLM 응답 품질은 모델 종류(OpenAI API, Fine-tuned LLM 등)에 따라 달라질 수 있다.
- 서비스는 기본적으로 인터넷 연결 환경에서의 사용을 전제로 하며, 오프라인 환경은 고려하지 않는다.

# 3. Specific Requirements

## 3.1 External Interface Requirements

### 3.1.1 User Interface (사용자 인터페이스)

---

#### 자연어 입력 기반 채팅 인터페이스

목적/내용	사용자가 자유롭게 감정, 상황, 영화에 대해 입력하고 답변을 받을 수 있도록 한다.
입력주체/출력목적지	사용자 ↔ 서버 (LLM)
범위/정확도/허용오차	문맥 기반 자연어 처리, 오탈자 자동 수정 허용
단위	문장 단위 입력
시간/속도	입력 즉시 서버로 전송, 5초 이내 응답 목표
타입 출력과의 관계	입력에 따른 영화 추천, 대화, 북마크 기능 활성화
화면 형식 및 구성	채팅창 + 추천 영화 카드 뷰
데이터 형식 및 구성	Text(자연어 입력), JSON(서버 응답)
명령 형식	자유 형식 자연어 입력
종료 메시지	필요시 "대화가 종료되었습니다." 안내

**Table 3.1:** User interface of chatting

#### 영화 정보 카드 UI

목적/내용	추천된 영화의 상세 정보를 카드 형태로 보여준다.
입력주체/출력목적지	서버 → 사용자
범위/정확도/허용오차	영화 데이터 기반 시각화
단위	영화 1편당 1 카드
시간/속도	요청 후 5초 이내 로딩
타입 출력과의 관계	카드 클릭 시 상세페이지 이동 또는 북마크 추가
화면 형식 및 구성	포스터 + 제목 + 한줄 설명 + 버튼(자세히 보기/북마크)
데이터 형식 및 구성	이미지(JPEG), 텍스트
명령 형식	버튼 클릭 기반 이벤트 처리
종료 메시지	해당 없음

**Table 3.2:** User interface of movie information card

## 등장인물 몰입형 대화 인터페이스

목적/내용	사용자가 영화 등장인물과 대화하는 경험을 제공한다.
입력주체/출력목적지	사용자 ↔ 서버 (LLM)
범위/정확도/허용오차	인물 말투 반영, 문맥 자연스러움 유지
단위	문장 단위 입력
시간/속도	입력 즉시 서버로 전송, 5초 이내 응답 목표
타입 출력과의 관계	등장인물 설정에 따른 대화 스타일 변화
화면 형식 및 구성	캐릭터 아바타 + 대화창
데이터 형식 및 구성	Text(자연어 입력), JSON(서버 응답)
명령 형식	자연어 대화
종료 메시지	"캐릭터 대화를 종료하시겠습니까?" 안내 후 종료

**Table 3.3:** Interactive dialogue User interface with movie characters

### 3.1.2 Hardware Interface (하드웨어 인터페이스)

이름	시스템 이용 디바이스
목적/내용	웹 브라우저를 통한 MovieChat 접속 및 이용
입력주체/출력목적지	사용자 ↔ 클라이언트(브라우저)
범위/정확도/허용오차	Windows, MacOS, Android, iOS
단위	디바이스 단위
시간/속도	브라우저 로딩 후 바로 사용 가능
타입 출력과의 관계	해당 없음
화면 형식 및 구성	웹 기반 레이아웃
데이터 형식 및 구성	HTML, CSS, JS
명령 형식	HTTP 요청 기반
종료 메시지	해당 없음 한다

**Table 3.4:** Hardware Interface

### 3.1.3 Software Interface (소프트웨어 인터페이스)

이름	서버 API
목적/내용	사용자 요청 처리 및 데이터 송수신
입력주체/출력목적지	클라이언트 → 서버
범위/정확도/허용오차	REST API 표준 준수

단위	API 호출 단위
시간/속도	요청당 3초 이내 응답 목표
타입 출력과의 관계	JSON 형식 데이터 송수신
화면 형식 및 구성	해당 없음
데이터 형식 및 구성	JSON
명령 형식	HTTP POST/GET
종료 메시지	상태 코드로 반환 (200, 400, 500 등)

**Table 3.5:** Software Interface

### 3.1.4 Communication Interface (통신 인터페이스)

이름	서버 ↔ 클라이언트 통신
목적/내용	사용자 입력 전송, 추천 결과 및 대화 응답 수신
입력주체/출력목적지	클라이언트 ↔ 서버
범위/정확도/허용오차	HTTPS 프로토콜, TLS 1.2 이상
단위	요청/응답 패킷
시간/속도	요청 후 5초 이내 응답
타입 출력과의 관계	API 호출 결과 표시
데이터 형식 및 구성	JSON
명령 형식	REST API 방식
종료 메시지	HTTP 연결 종료

**Table 3.6:** Communication Interface

## 3.2 Functional Use case

### 3.2.1 Use Cases (기능별 유스케이스)

<i>Use case name</i>	<i>로그인 및 사용자 등록</i>
Actor	등록되지 않았거나 로그아웃 상태의 등록된 사용자
Description	서비스 이용을 위해 사용자가 계정을 생성하거나 기존 계정으로 로그인하는 과정을 의미한다. 등록 후에는 개인화된 영화 추천, 북마크, 아카이빙 등의 기능을 사용할 수 있다.
Normal Course	<ol style="list-style-type: none"> <li>1. 사용자가 메인 화면에서 '회원가입' 또는 '로그인' 버튼을 클릭한다.</li> <li>2. 회원가입의 경우, 사용자는 이메일, 비밀번호 등의 정보를 입력하여 계정을 생성한다.</li> </ol>

---

3. 로그인 시 사용자는 등록한 이메일과 비밀번호를 입력하여 인증을 진행한다.
4. 시스템은 인증 정보를 확인한 후, 사용자 계정을 활성화하고 홈페이지으로 이동한다.

---

Precondition	<ul style="list-style-type: none"> <li>• 사용자가 시스템에 등록되지 않았거나 로그아웃 상태여야 한다.</li> </ul>
Post Condition	<ul style="list-style-type: none"> <li>• 로그인 성공 시, 사용자 세션이 생성되며 개인화 기능이 활성화된다.</li> <li>• 회원가입 성공 시, 신규 사용자 계정이 데이터베이스에 저장된다.</li> </ul>
Assumptions	<ul style="list-style-type: none"> <li>• 인증 서버가 정상적으로 동작 중이어야 하며, 이메일 유효성 검증이 가능해야 한다.</li> </ul>

---

**Table 3.7:** Use case of login and register

---

<i>Use case name</i>	<b>영화 추천 요청</b>
Actor	등록된 사용자
Description	자연어로 영화 추천을 요청하고, 시스템으로부터 개인 맞춤형 추천 영화를 받는 과정이다.
Normal Course	<ol style="list-style-type: none"> <li>5. 사용자가 메인 화면 또는 챗봇 입력창에 자연어로 영화 추천을 요청한다.</li> <li>6. 시스템은 사용자 입력을 분석하고 추천 알고리즘을 기반으로 적절한 영화를 선택한다.</li> <li>7. 추천된 영화의 제목, 장르, 포스터, 줄거리, 평점 등 핵심 정보를 사용자에게 제공한다.</li> <li>8. 사용자는 추천 영화 중 하나를 선택해 상세 정보를 확인할 수 있다.</li> </ol>
Precondition	<ul style="list-style-type: none"> <li>• 사용자 입력은 분석 가능한 자연어 문장이어야 한다.</li> </ul>
Post Condition	<ul style="list-style-type: none"> <li>• 추천 기록은 사용자 히스토리에 저장된다.</li> </ul>
Assumptions	<ul style="list-style-type: none"> <li>• TMDB API 와 추천 알고리즘이 정상 작동 중이어야 한다.</li> </ul>

---

**Table 3.8:** Use case of movie recommendation

<i>Use case name</i>	<b>영화 관련 리뷰 확인 및 토론</b>
Actor	등록된 사용자
Description	영화에 대한 다양한 리뷰를 확인하고, 해당 내용에 대해 시스템과 자연어로 해석·토론하는 기능이다.
Normal Course	<ol style="list-style-type: none"> <li>1. 사용자가 특정 영화의 리뷰 요약이나 해석을 요청한다.</li> <li>2. 시스템은 크롤링된 리뷰, 줄거리, 시놉시스를 기반으로 요약된 정보를 제공한다.</li> <li>3. 사용자는 추가적으로 인물 관계, 결말 해석, 상징성 등 깊이 있는 질문을 던질 수 있다.</li> <li>4. 시스템은 해당 영화 데이터와 벡터 검색 기반 유사 내용을 바탕으로 답변을 생성한다.</li> </ol>
Precondition	<ul style="list-style-type: none"> <li>• 영화 리뷰 및 줄거리 데이터가 TMDB나 Wikipidea에 정리되어 있어야 한다.</li> </ul>
Post Condition	<ul style="list-style-type: none"> <li>• 사용자 입력은 추천 시스템의 개인화 정보로 활용될 수 있다.</li> </ul>
Assumptions	<ul style="list-style-type: none"> <li>• 리뷰 데이터는 왓챠피디아 등 신뢰 가능한 출처로부터 크롤링된다</li> </ul>

**Table 3.9:** Use case of exploring and discussing movie review

<i>Use case name</i>	<b>북마크 및 아카이빙</b>
Actor	등록된 사용자
Description	관심 있는 영화는 북마크하고, 시청 완료한 영화는 아카이빙할 수 있다. 해당 정보는 이후 추천에 활용된다.
Normal Course	<ol style="list-style-type: none"> <li>1. 사용자가 영화 추천 결과나 검색 화면에서 북마크 아이콘을 클릭한다.</li> <li>2. 시스템은 해당 영화를 사용자의 ‘보고 싶은 영화’ 목록에 추가한다.</li> <li>3. 사용자가 영화를 시청 후 아카이빙 버튼을 누르면, 해당 영화는 ‘시청 완료’ 리스트로 이동된다.</li> <li>4. 북마크/아카이빙 정보는 추후 개인화된 추천에 반영된다.</li> </ol>
Precondition	<ul style="list-style-type: none"> <li>• 사용자의 브라우저는 쿠키 또는 로컬스토리지를 지원해야 한다.</li> </ul>
Post Condition	<ul style="list-style-type: none"> <li>• 북마크/아카이빙 데이터는 브라우저에 임시 저장된다.</li> <li>• 로그인 시 서버 계정으로 이전 가능하다.</li> </ul>

Assumptions	<ul style="list-style-type: none"> <li>추천 시스템은 이 데이터를 기반으로 다음 추천 시 반영하도록 설계되어 있다.</li> </ul>
-------------	--

**Table 3.10:** Use case of bookmark and archiving

<i>Use case name</i>	<b>영화 인물과의 몰입형 대화</b>
Actor	등록된 사용자
Description	사용자가 영화 등장인물과 몰입형 대화를 통해 영화 속 세계관을 경험할 수 있다.
Normal Course	<ol style="list-style-type: none"> <li>사용자가 영화 상세 페이지에서 등장인물을 선택한다.</li> <li>시스템은 해당 인물의 캐릭터 데이터를 기반으로 새로운 대화 세션을 생성한다.</li> <li>사용자는 해당 인물과 자연어로 대화를 진행할 수 있다.</li> <li>시스템은 위키피디아 및 영화 시나리오 정보 기반으로 인물의 말투, 성격, 배경을 반영한 응답을 제공한다.</li> </ol>
Precondition	<ul style="list-style-type: none"> <li>선택된 등장인물에 대한 정보가 사용자에 의해 제공되거나 wikipedia에서 접근 가능해야 한다.</li> </ul>
Post Condition	<ul style="list-style-type: none"> <li>대화 기록이 저장되며, 이후 추천/분석에 활용 가능하다.</li> </ul>
Assumptions	<ul style="list-style-type: none"> <li>인물 관련 데이터는 Wikipedia 등 외부 출처에서 수집된다.</li> </ul>

**Table 3.11:** Use case of immersive conversation with movie character

### 3.2.2 Use Case Diagram (유스케이스 다이어그램)

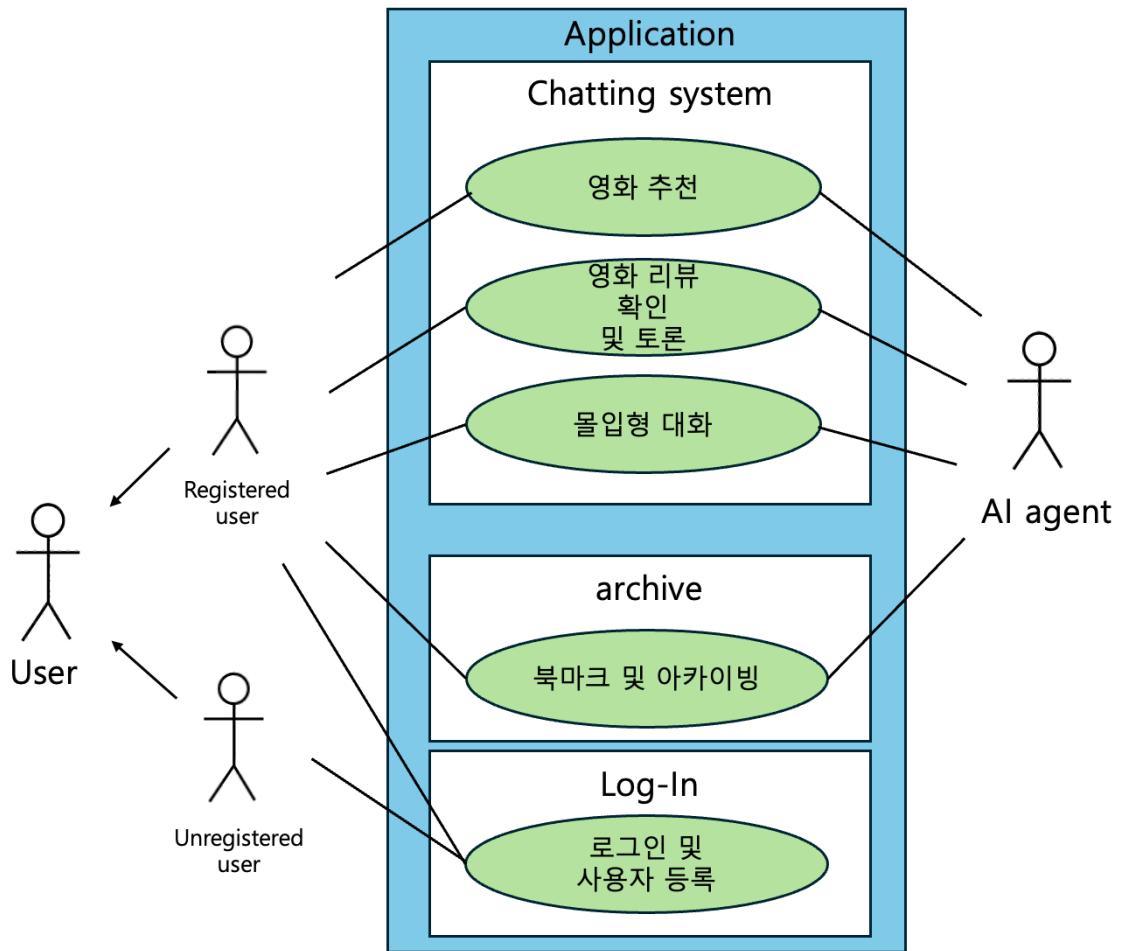


Figure 3.1: Use case diagram

### 3.2.3 Data Dictionary

#### 1. User Table

Field	Key	Constraint	Description
id	PK		사용자 고유 식별자 (UUID)
email		Not Null, Unique	사용자 이메일 주소
password		Not Null	암호화된 비밀번호
nickname		Not Null	사용자 닉네임
created_at		Not Null	계정 생성 일시

Table 3.12: User table

## 2. Movie Table

Field	Key	Constraint	Description
movie_id	PK		영화 고유 ID (TMDB 기준)
title		Not Null	영화 제목
overview			영화 줄거리 요약
release_date			개봉일
poster_url			포스터 이미지 URL
trailer_url			예고편 영상 URL
director_id			감독 ID

**Table 3.13:** Movie table

## 3. Bookmarked Movie Table (inherits Movie Table)

Field	Key	Constraint	Description
movie_id	FK	Not Null	영화 고유 ID (TMDB 기준)
user_id	FK	Not Null	사용자 고유 식별자 (UUID)
(movie_id, user_id)	PK		
created_at		Not Null	Bookmark 생성 일시

**Table 3.14:** Bookmarked Movie table

## 4. Archived Movie Table (inherits Movie Table)

Field	Key	Constraint	Description
movie_id	FK	Not Null	영화 고유 ID (TMDB 기준)
user_id	FK	Not Null	사용자 고유 식별자 (UUID)
(movie_id, user_id)	PK		
rating		Not Null, 0 ~ 5	사용자 별점 (default=0)
created_at		Not Null	Archive 생성 일시

**Table 3.15:** Archived Movie table

## 5. Chat Room Table

Field	Key	Constraint	Description
id	PK		채팅방 고유 ID
user_id	FK	Not Null	대화를 진행한 사용자 ID

created_at		Not Null	채팅방을 생성한 시간
------------	--	----------	-------------

**Table 3.16:** Chat Room table

## 6. Chat History Table

Field	Key	Constraint	Description
chat_id	PK		단일 대화 고유 ID
chat_room_id	FK	Not Null	이 대화가 진행된 채팅방
timestamp		Not Null	이 대화가 발화된 시각
message		Not Null	채팅 메시지

**Table 3.17:** Chat History table

## 7. Chat Response Table

Field	Key	Constraint	Description
ai_chat_id	FK	Not Null	ai agent가 작성한 채팅 ID
user_chat_id	FK	Not Null	user가 작성한 채팅 ID
(ai_chat_id, user_chat_id)	PK		

**Table 3.18:** Chat Response table

## 8. Character Profile Table (for Movie Table)

Field	Key	Constraint	Description
character_id	PK		등장인물 고유 ID
movie_id	FK	Not Null	등장인물이 속한 영화 ID
name		Not Null	등장인물 이름
description		Not Null	등장인물 성격, 특징 요약
tone		Not Null	말투, 어조 등 대화 스타일

**Table 3.19:** Character Profile table

## 9. Movie Genre Table (for Movie Table)

Field	Key	Constraint	Description
genre_id	FK	Not Null	장르 ID
movie_id	FK	Not Null	영화 고유 ID (TMDB 기준)
(genre_id, movie_id)	PK		

**Table 3.20:** Movie Genre table

#### 10. Movie Actor Table (for Movie Table)

Field	Key	Constraint	Description
actor_id	FK	Not Null	연기자 ID
movie_id	FK	Not Null	영화 고유 ID (TMDB 기준)
(actor_id, movie_id)	PK		

**Table 3.21:** Movie Actor table

#### 11. Movie Platform Table (for Movie Table)

Field	Key	Constraint	Description
platform_id	FK	Not Null	스트리밍 지원 플랫폼 ID
movie_id	FK	Not Null	영화 고유 ID (TMDB 기준)
(platform_id, movie_id)	PK		

**Table 3.22:** Movie Platform table

#### 12. Genre Table

Field	Key	Constraint	Description
genre_id	PK		장르 ID
genre_name		Not Null, Unique	장르명

**Table 3.23:** Genre table

#### 13. Actor Table

Field	Key	Constraint	Description
actor_id	PK		연기자 ID
actor_name		Not Null	연기자명

**Table 3.24:** Actor table

#### 14. Platform Table

Field	Key	Constraint	Description
platform_id	PK		스트리밍 플랫폼 ID
platform_name		Not Null, Unique	스트리밍 플랫폼명

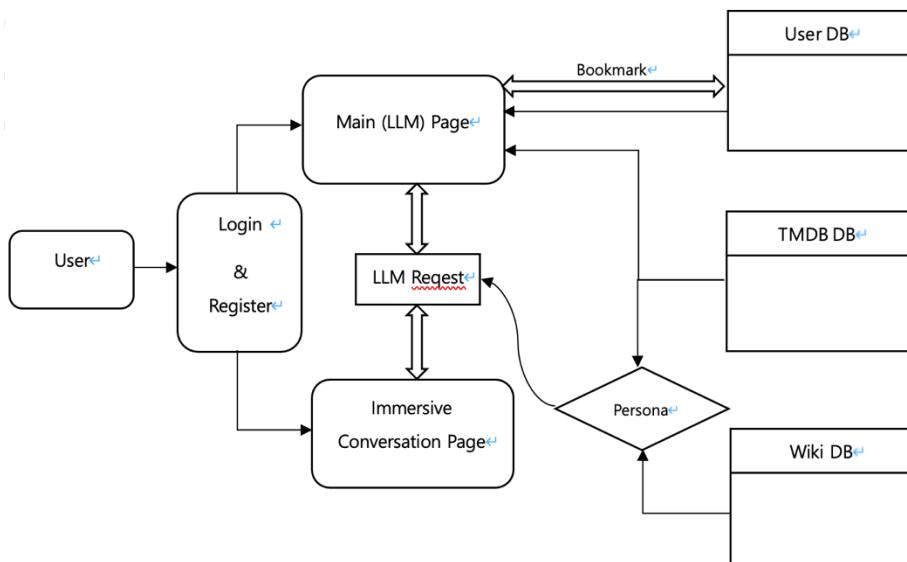
**Table 3.25:** Platform table

## 15. Director Table

Field	Key	Constraint	Description
director_id	PK		감독 ID
director_name		Not Null	감독명

**Table 3.26:** Director table

### 3.2.4 Data Flow Diagram



**Table 3.2:** Data Flow Diagram

## 3.3 Performance Requirements

아래는 본 시스템이 갖추어야 할 성능을 다룬 것이다. 예측에 기반한 내용이며 실제 구현시 제한 사항에 따라 달라질 수 있다.

### 3.3.1 Static Numerical Requirements (정적 수치 요구사항)

- MovieChat 시스템은 기능 테스트를 위해 최소 50명 이상의 동시 접속을 지원할 수 있어야 한다.  
(장기적으로는 200명 이상 동시 접속도 고려한다)

- Windows, macOS, Linux에서 사용하는 최신 브라우저에서 접속이 되어야 한다.
- 클라이언트 환경의 최소한의 사양은 다음과 같다.
  - 운영체제 : 윈도우7 이상, macOS 10.13 이상
  - 인터넷 속도 : 최소 1Mbps
  - 화면 해상도 : 1280×720 이상을 권장
  - 브라우저 : Google Chrome 90 이상, Microsoft Edge 90 이상. 쿠키 및 자바스크립트 활성화 필요.

### 3.3.2 Dynamic Numerical Requirements (동적 수치 요구사항)

---

- 사용자 경험을 충족하기 위해 시스템에서 갖추어야 하는 요청 응답 시간은 다음과 같다. 구글 연구진의 「User Preference and Search Engine Latency」에 따르면 검색엔진은 4초 이내에 결과를 제공해야 한다. 사용자의 직접적인 상호작용과 무관한 기능(예: 로그 기록, 동기화 등)은 이보다 다소 지연된 응답을 허용할 수 있다.
  - 사용자의 프롬프트 입력 후 4초 이내에 결과가 완전히 출력되어야 한다.
  - 다음 프롬프트를 위해 LLM에서 전송된 내용이 데이터베이스에 6초 이내에 반영되어야 한다.
  - 사용자의 요청에 따라 몰입형 대화를 위한 등장인물은 10초 이내에 생성되어야 한다.
  - 주기적인 데이터베이스 업데이트를 위해 영화 데이터를 조회하는 경우, TMDB API 요청은 20초 이내에 완료되어야 한다. 오류가 발생해 실패할 경우 알림과 함께 재시도한다.
- 그 외에 응답 사항과 관련해 일반적으로 요구되는 사항은 다음과 같다.
  - 회원가입 요청 후 5초 이내에 완료 응답이 전달되어야 한다.
  - 로그인 요청 후 3초 이내에 MovieChat 메인 화면 진입해야 한다.
  - 북마크와 시청 기록 저장은 6초 이내에 데이터베이스에 반영되어야 한다.
- 요청 처리율
  - 서버는 반드시 초당 평균 10개 이상의 LLM 요청을 처리해야 하며, 이를 위한 LLM 모델을 선정해야 한다.
  - 사용자 한 명당 하루 평균 100건의 대화를 저장하고, 월 최대 1GB까지 안정적인 데이터 처리가 가능해야 한다.

## 3.4 Logical Database Requirements

### 3.4.1 데이터베이스 개요

---

MovieChat 서비스는 사용자 계정, 영화 정보, 사용자의 북마크 기록, 사용자의 아카이브 기록, 대화 히스토리, 영화 등장인물 프로필 데이터를 효율적으로 저장 및 관리하기 위해 관계형 데이터베이스를 사용한다.

초기 개발 단계에서는 PostgreSQL을 기본 데이터베이스로 사용한다.

### 3.4.2. 저장되는 주요 데이터

---

#### 사용자(User)

- 이메일, 비밀번호, 닉네임
- 북마크한 영화 ID 목록(Bookmarked Movies)
- 아카이빙한 영화 ID 목록(Archived Movies)
- 계정 생성일

#### 영화(Movie)

- TMDB 영화 ID
- 제목, 줄거리, 장르, 개봉일, 감독, 주요 배우
- 포스터 이미지 URL, 예고편 URL
- 스트리밍 서비스 정보들

#### 대화 히스토리(Chat History)

- 대화 내용
- 발화 시작

#### 영화 등장인물 프로필(Character Profile)

- 등장인물 이름
  - 소속 영화
  - 인물 설명(성격, 말투 등)
- 

### 3.4.3 데이터베이스 구조

---

User Tab 다음은 데이터베이스 구조를 ER Diagram(Crow's Foot Diagram)으로 나타낸 것이다.

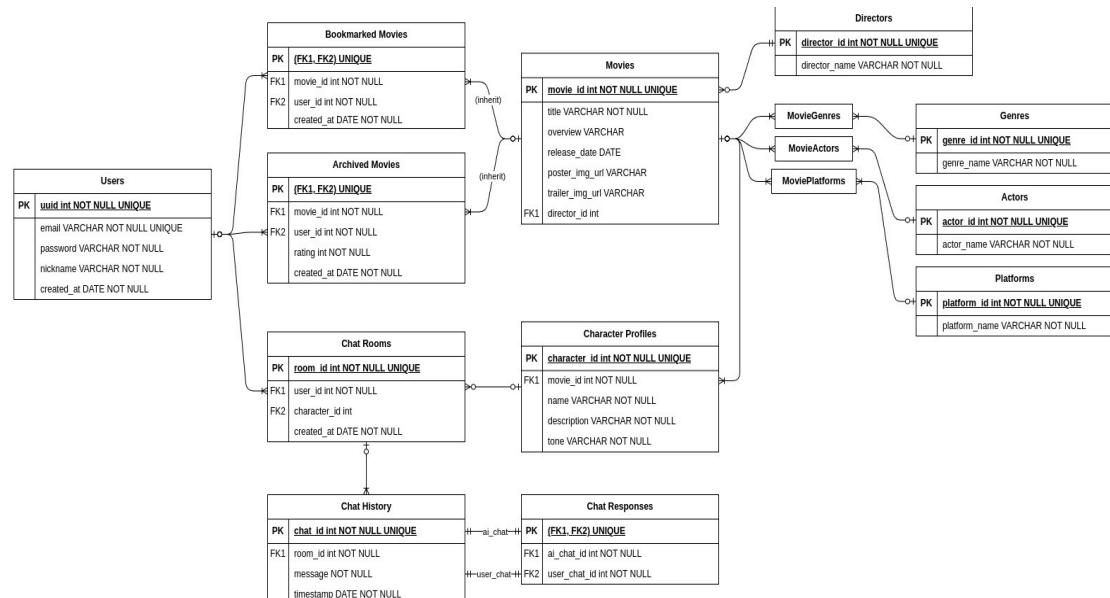


Figure 3.3: Database Architecture

### 3.4.4 데이터베이스 제약 사항, 무결성 및 보안

---

#### 데이터 제약 사항

다른 계정 정보에 동일한 이메일이 존재하면 안 된다.

유저의 rating은 NULL일 수 없고, 0 ~ 5 사이의 정수값이어야 한다. (0, 5 포함)

Chat history의 timestamp는 NULL일 수 없다. (시간 순으로 정렬해서 보여주어야 하기 때문)

#### 데이터 무결성

모든 PK는 고유성을 보장해야 한다. (NOT NULL, UNIQUE)

모든 FK는 참조 무결성을 유지해야 한다. (존재하지 않는 항목을 참조하고 있으면 안 된다)

FK에 NOT NULL 제약이 있다면, 기본적으로 ON DELETE CASCADE 정책을 따른다.

#### 데이터 보안 및 접근 권한

비밀번호는 평문으로 저장하지 않는다. 대신 해시값(Hash)을 저장한다.

유저의 개인 정보는 다른 일반 유저가 조회할 수 없다. (예: chat history, bookmarked movies)

유저는 chat history들을 생성할 수 있고, 자신이 생성한 chat history를 삭제할 수 있다.

유저의 개인 정보는 다른 일반 유저가 변경 및 삭제할 수 없다.

## 3.5 Design Constraints

### 3.5.1 Physical Design Constraints

---

- **플랫폼**
  - MovieChat 은 웹 기반 플랫폼이다. 초기 버전에서는 PC 브라우저를 지원하고, 추후에 모바일 브라우저 확장을 고려한다. 초기 버전에서는 별도의 네이티브 앱(Android/iOS)은 지원하지 않는다.
- **서버/인프라**
  - 백엔드는 **FastAPI** 프레임워크를 기반으로 한다.
  - 데이터베이스는 **MySQL** 을 사용한다.
  - 서버는 **Ubuntu 20.04 이상 Linux 환경** 을에서 운영된다.
- **외부 연동**
  - 영화 정보 제공은 **TMDB API** 를 기반으로 한다.
  - 영화 리뷰, 등장인물 프로필 등 추가 정보는 **Wikipedia API** 및 크롤링을 통해 확보한다.
  - LM 응답은 **OpenRouter** 를 통해 제공된다.(여러 LLM 모델마다 상이한 API 형식을 통일해, 다양한 모델 전환을 용이하게 해 유지보수성을 높이는 걸 목적으로 한다.)

### 3.5.2 Standards Compliance

---

- **코딩 스타일**
  - 변수명은 `lowerCamelCase`, 함수명과 데이터베이스 객체는 `PascalCase`를 사용한다.
  - 백엔드는 `PEP8 (Python Style Guide)` 을 준수하여 작성한다.
  - 프론트엔드는 `HTML5, CSS3, JavaScript (ES6+)` 표준을 따른다.
- **데이터 포맷**
  - 서버-클라이언트 간 통신은 `JSON 포맷`을 기본으로 한다.
  - 모든 API 응답은 오류 발생 시 클라이언트에게 노출되지 않는 표준화된 에러 코드를 반환한다.

## 3.6 Software System Characteristics

### 3.6.1 Product Requirements

---

- **Usability Requirements**
  - MovieChat 은 기존 웹 기반 생성형 인공지능 서비스(ChatGPT, Gemini, Claude 등)를 사용해본 사용자라면 별도의 학습 없이도 즉시 사용할 수 있도록 직관적인 UI/UX를 제공해야 한다.
  - 몰입형 대화 기능은 등장인물 선택만으로 자동 프롬프트가 생성되어야 하며, 사용자는 추가 작업 없이 자연스럽게 대화를 이어갈 수 있어야 한다.
- **Performance Requirements**
  - 사용자가 영화 정보를 요청할 경우, 해당 영화의 포스터, 대표 이미지 및 관련 OTT 링크는 출처를 명시하여 함께 제공되어야 한다.
  - 사용자의 프롬프트 입력 후 결과는 4 초 이내에 화면에 출력되어야 하며, LLM 응답 지연 시간은 5 초를 초과해서는 안 된다. 외부 요인(LLM API)으로 초과 시, 사용자에게 적절한 안내 메시지를 제공해야 한다.
- **Security Requirements**
  - 사용자의 개인정보는 「개인정보 보호법」 등 국내 관련 법률을 준수하여 수집·보관·처리되어야 하며, 서비스 제공 외 목적의 사용은 제한되어야 한다.
  - 모든 개인정보는 암호화하여 저장하고, 사용자는 본인의 정보를 확인하고 관리할 수 있어야 한다.
  - 모든 클라이언트-서버 통신은 HTTPS 를 통해 암호화되어야 하며, 민감한 데이터는 노출되지 않도록 해야 한다.
  - 사용자는 데이터베이스에 직접 접근할 수 없어야 하며, 외부 API(Key 포함)는 서버 측에서 안전하게 관리되어야 하고 클라이언트에 노출되어서는 안 된다.
- **Maintainability Requirements**
  - 모든 서브 시스템은 기능별로 모듈화되어야 하며, 독립적으로 수정 및 배포가 가능해야 한다.
  - 정기적인 코드 리뷰와 문서화를 통해 유지보수성을 확보해야 하며, 프론트엔드와 백엔드는 명확히 분리되어야 한다.
  - 모든 소스 코드는 Git 기반의 버전 관리 시스템을 통해 추적되고 관리되어야 한다.
- **Portability Requirements**

- MovieChat 은 웹 기반 애플리케이션으로, 추후 지원하게 되는 모바일 브라우저 및 태블릿 환경에서도 주요 기능이 동일하게 제공되어야 하며, 각 기기에 적합한 UI/UX 가 구현되어야 한다.

### 3.6.2 Organizational Requirements (조직상의 요구사항)

- **개발 프로세스**
  - MovieChat 은 상세한 요구사항 명세서 작성과 설계 단계를 거쳐 체계적으로 개발되어야 한다.
  - 단순히 개발 과정에서 그치지 않고 개발 완료 후에는 사용자 환경을 고려하여 테스트 및 품질 확인 과정을 거쳐 배포되어야 한다.
- **Operational Requirements**
  - 시스템은 연중무휴 24 시간 서비스를 제공해야 하며, 서비스 안정성을 위해 하루 1 회 이상 자동 백업이 수행되어야 한다.
  - 운영 중 발생하는 장애는 자동 감지되어 관리자에게 즉시 알림이 전달되어야 하며, 복구를 위한 절차는 사전에 문서화되어 있어야 한다.
  - 또한 사용자 피드백은 정기적으로 수집 및 분석되며, 기능 개선의 우선순위 결정과 업데이트 계획 수립에 반영되어야 한다.

### 3.6.3 External Requirements

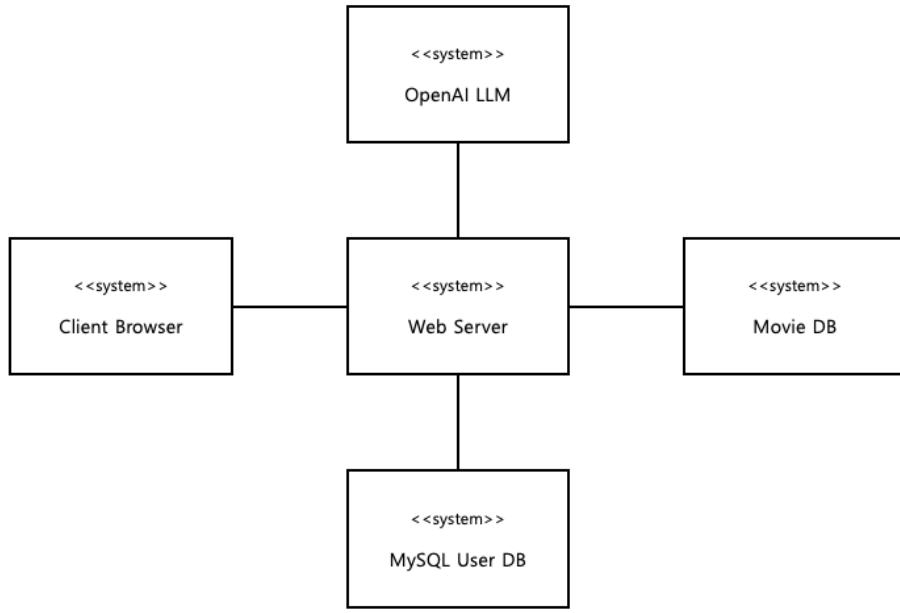
- **Regulatory Requirements**
  - 영화 관련 데이터(줄거리, 포스터, 배우 정보 등)는 TMDB API 의 라이선스 정책을 준수하여 사용되어야 하며, 상업적 이용이 포함될 경우 API 제공자의 사용 범위를 사전에 확인해야 한다.
  - 몰입형 등장인물 기능에 사용되는 정보(이름, 외형, 말투, 설정 등)는 제 3 자의 지적재산권(IP)을 침해할 가능성이 있으므로, 실존 IP(예: 마블, 디즈니 캐릭터 등)를 활용한 생성은 상업적 목적 여부와 관계없이 제약을 받아야 한다.
  - 따라서 운영자는 사용자 생성 캐릭터의 이용 목적을 검토하고, 상업적 또는 악의적 이용이 감지될 경우 해당 기능을 제한하거나 차단하는 제재 시스템을 도입해야 한다.
  - 또한 이용자에게는 관련 법률(저작권법 등)에 기반한 주의사항을 사전에 고지하고, 이용약관 및 서비스 정책에 이를 명시해야 한다.

## 3.7 Organizing the Specific Requirements

이 구간에서는 UML 및 표 형식 기반의 그래픽 표기법을 사용하여 시스템 모델을 설명한다. 시스템 모델은 시스템과 서브 시스템 간의 관계를 설명한다.

### 3.7.1 Context Model (컨텍스트 모델)

MovieChat은 사용자, 서버, 외부 API(특히 TMDB, OpenAI API 등) 간 상호작용을 중심으로 구성된다. 컨텍스트 모델은 시스템이 외부 환경과 어떤 방식으로 상호작용하는지 보여준다.

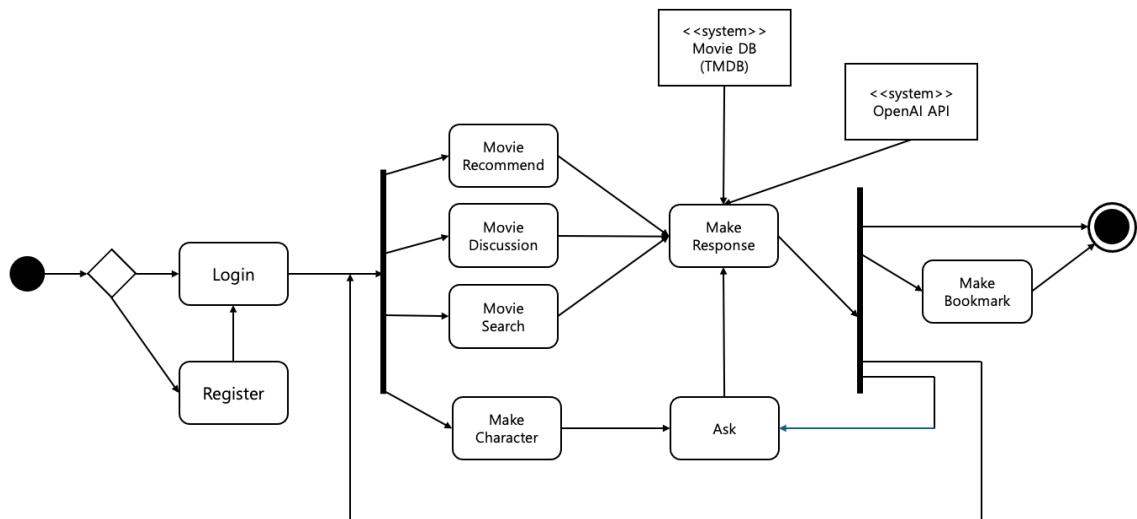


**Figure 3.4:** Context model

### 주요 흐름

- 사용자는 웹 클라이언트를 통해 자연어로 입력
- 클라이언트는 서버로 요청 전송
- 서버는 LLM과 외부 영화 데이터 API를 호출
- 서버는 결과를 클라이언트에 반환
- 사용자는 추천, 대화 결과를 확인하거나 북마크/아카이빙을 수행

### 3.7.2 Process Model (프로세스 모델)



**Figure 3.5:** Process model

MovieChat 시스템에서 주요 프로세스 흐름은 다음과 같다:

1. 사용자 입력 처리
  - 감정, 상황, 영화 관련 질의를 입력
  - 서버가 입력을 파싱하고 의도(Intent) 분석
2. 추천 및 대화 응답 생성
  - 추천: 감정 기반 큐레이션을 위한 영화 리스트 생성
  - 대화: 특정 영화나 등장인물에 대한 자연스러운 대화 생성
3. 영화 정보 연동
  - TMDB API를 통해 영화 상세 정보(포스터, 줄거리 등) 가져오기
  - 등장인물 정보가 필요할 경우 Wikipedia 또는 별도 크롤링 데이터 조회
4. 결과 출력 및 상호작용
  - 추천 결과를 카드 형태로 제공
  - 북마크, 아카이빙 기능 활성화
  - 대화 히스토리 저장
5. 데이터 관리 및 업데이트
  - 사용자 프로필(북마크, 시청 이력) 업데이트
  - 대화 로그 저장

### 3.7.3 Interaction Model (인터랙션 모델)

---

사용자 ↔ MovieChat 시스템 간 상호작용은 크게 다음 3가지로 나뉜다:

- 영화 추천 흐름
  1. 사용자가 감정/상황 입력
  2. 서버가 관련 영화 추천
  3. 추천 영화 리스트 반환 및 북마크/시청 기능 제공
- 영화 토론/해석 대화 흐름
  1. 사용자가 특정 영화에 대해 질문
  2. 서버가 영화 내용을 기반으로 대화 생성
  3. LLM 응답을 사용자에게 제공
- 등장인물 몰입형 대화 흐름
  1. 사용자가 등장인물 선택
  2. 서버가 등장인물 설정 반영 후 대화 생성
  3. 몰입형 자연어 대화 제공

자세한 내용은 3.2.1, 3.2.2 의 use case 참고

## 3.8 System Architecture

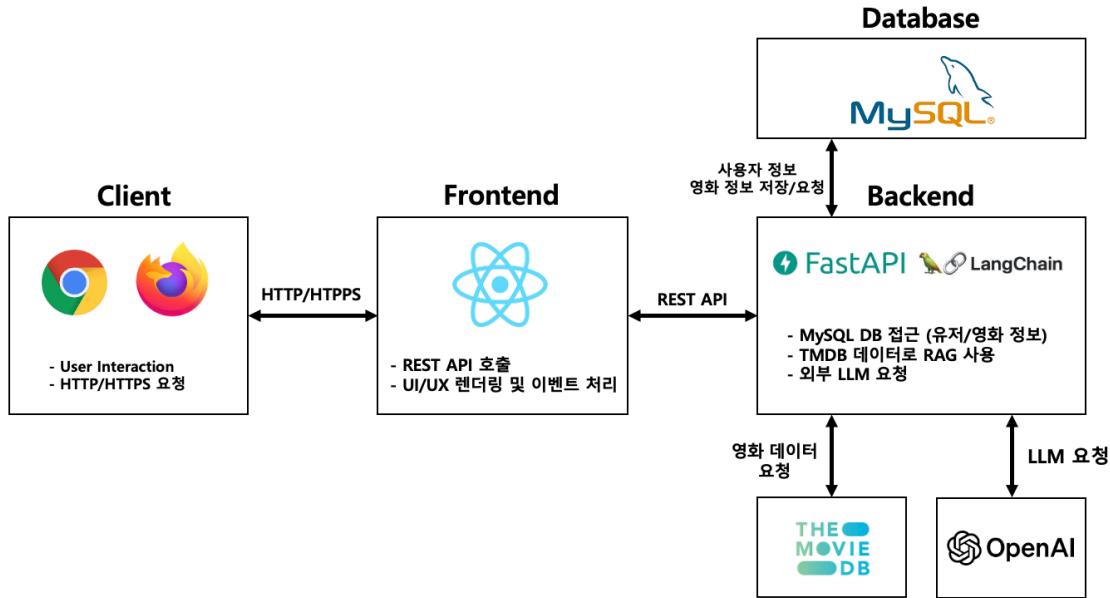


Figure 3.6: System Architecture

## 3.9 System Evolution

### 3.9.1 Limitation and Assumption (현재 한계와 가정)

- 현재 시스템 한계**
  - 초기 버전에서는 영화 추천 및 토큰 기능 중심으로 동작하며, 등장인물 몰입형 대화는 일부 영화에 한정된다.
  - 감정 분석 및 상황 이해 수준은 LLM 성능에 따라 일정 부분 제한이 있을 수 있다.
  - 영화 메타데이터는 주로 TMDB API를 사용하기 때문에, TMDB에 등록되지 않은 영화 정보는 부실할 수 있다.
  - 서버 자원이 제한적이기 때문에 초기에는 대규모 동시 접속 처리에 한계가 있을 수 있다.
  - 등장인물 말투, 성격 표현은 간단한 프롬프트 엔지니어링 기반으로 이뤄지며, 자연스러운 상호작용을 제공하기 위해서는 추가 튜닝이 필요할 수 있다.
- 가정 사항**
  - 사용자는 자연어로 자유롭게 질문할 수 있으며, 기본적인 영화 정보(줄거리, 등장인물)를 이해하고 있다는 것을 가정한다.
  - TMDB API 및 외부 LLM API(OpenAI 등)가 지속적으로 안정적으로 제공된다는 가정 하에 설계한다.
  - 초기 사용자는 MovieChat의 감성 기반 추천, 영화 토큰 기능에 만족하고 추가 기능에 대한 수요가 점진적으로 늘어날 것으로 예상한다.

### 3.9.2 Evolution of Hardware and Change of User Requirements

- 하드웨어/인프라 발전**

- 클라우드 인프라(AWS, GCP) 비용이 지속적으로 낮아지고, 고성능 GPU 서버 임대 비용도 감소하고 있다.
  - 향후 사용자 수 증가 및 등장인물 몰입형 대화 기능 고도화를 위해 자체 LLM 모델 호스팅(파인튜닝 모델 배포)을 고려할 수 있다.
  - 실시간 스트리밍 기능(예: 영화 예고편 실시간 재생, 대화형 추천 스트리밍 등)도 인프라 구성을 통해 지원 가능해질 것이다.
- **사용자 요구 변화**
    - 영화 추천의 단순성 → 감정 기반 개인화 추천 + 토론 중심 경험 강화로 변할 가능성이 높다.
    - 초기에는 "추천"이 중심이지만, 점차 시스템에 익숙해지며 "영화 해석에 대한 심도있는 대화", 등장인물과의 "몰입형 대화", "아카이빙 및 회고" 기능에 대한 수요가 증가할 것으로 예상된다.
    - 더 나아가서는 드라마, 애니메이션, 다큐멘터리 등 "**영화 외 장르**" 확장 요청도 발생할 수 있다.
    - 다국어 지원(특히 영어/일본어/중국어) 수요가 증가할 수 있으므로, 시스템은 다국어 확장성을 염두에 두고 개발해야 한다.

# 4. Supporting Information

## 4.1. Software Requirements Specification

소프트웨어 요구사항 명세서 IEEE 권장사항 (IEEE Recommend Practice for Software Requirements Specifications, IEEE-Std-830)을 참고해 작성되었다

## 4.2. Document History

Document History			
Date	Version	Description	Writer
2025-05-08	V1.00	Scope, Definition, Function Requirement	이해민
2025-05-08	V1.00	Database Requirement, Data Dictionary	박찬원
2025-05-08	V1.00	Performance Requirement, Design Constraints, Software System Attributes	이상훈
2025-05-08	V1.00	Data Flow Diagram	박진우
2025-05-08	V1.00	Organizing the Specific Requirements	강병희
2025-05-09	V1.00	Latex Formatting	강병희