Movie Recommendation with Agent AI

Kiến trúc phần mềm

Phiên bản <1.1>

Lịch sử thay đổi

|  |  |  |  |
| --- | --- | --- | --- |
| **Ngày** | **Phiên bản** | **Mô tả thay đổi** | **Tác giả** |
| 11/05/2025 | 1.0 | Viết nội dung đầu tiên | Lê Châu Hữu Thọ |
| 23/05/2025 | 1.1 | Thêm sơ đồ triển khai, cáu trúc thư mục, chỉnh sửa mô hình ML | Lê Châu Hữu Thọ |
|  |  |  |  |
|  |  |  |  |

Mục lục

[1. Giới thiệu 4](#_Toc197893842)

[1.1 Mục đích 4](#_Toc197893843)

[1.2 Phạm vi 4](#_Toc197893844)

[1.3 Định nghĩa, Từ viết tắt 4](#_Toc197893845)

[1.4 Tài liệu Tham khảo 5](#_Toc197893846)

[1.5 Tổng quan về Tài liệu Kiến trúc Phần mềm 5](#_Toc197893847)

[2. Các mục tiêu và ràng buộc kiến trúc 5](#_Toc197893848)

[3. Sơ đồ Use-Case 6](#_Toc197893849)

[4. Góc nhìn luận lý (Logical View) 7](#_Toc197893850)

[4.1 Thành phần: Giao diện người dùng và Quản trị 7](#_Toc197893851)

[4.2 Thành phần: Hệ thống Backend (FastAPI) 7](#_Toc197893852)

[4.2.1 Thành phần con: User Service (Đăng ký/Đăng nhập) 8](#_Toc197893853)

[4.2.2 Thành phần con: Tools 8](#_Toc197893854)

[4.2.3 Thành phần con: Agent Query 8](#_Toc197893855)

[4.2.4 Thành phần con: Recommendation System 8](#_Toc197893856)

[4.3 Thành phần: Cơ sở dữ liệu (MongoDB và MySQL) 9](#_Toc197893857)

[5. Triển khai 9](#_Toc197893858)

[6. Góc nhìn ở mức độ cài đặt (Implementation View) 9](#_Toc197893859)

[7. Mô hình ML 9](#_Toc197893860)

[7.1 Tổng quan về workflow 9](#_Toc197893861)

[7.2 Thu thập và tiền xử lý dữ liệu 9](#_Toc197893862)

[7.3 Xử lý đặc trưng 10](#_Toc197893863)

[7.4 Huấn luyện mô hình 10](#_Toc197893864)

[7.5 Đánh giá mô hình 10](#_Toc197893865)

[7.6 Triển khai mô hình 11](#_Toc197893866)

[7.7 Vận hành, giám sát và bảo trì 11](#_Toc197893867)

Kiến trúc phần mềm

# Giới thiệu

## Mục đích

Tài liệu Kiến trúc Phần mềm (SAD) này cung cấp mô tả toàn diện về kiến trúc của hệ thống **Movie Recommendation with Agent AI (MRAA)**. Mục đích chính của tài liệu này là trình bày cấu trúc, các thành phần chính, mối quan hệ giữa các thành phần đó và các nguyên tắc thiết kế đã được áp dụng. Tài liệu này nhằm mục đích cung cấp một sự hiểu biết chung cho tất cả các bên liên quan, bao gồm các nhà phát triển, kiểm thử viên, quản lý dự án, và các bên liên quan khác, về cách hệ thống được tổ chức và hoạt động ở mức kiến trúc. Nó cũng đóng vai trò là kim chỉ nam cho quá trình thiết kế chi tiết, triển khai và bảo trì hệ thống.

## Phạm vi

Tài liệu này mô tả kiến trúc phần mềm của hệ thống **Movie Recommendation with Agent AI**, bao gồm các khía cạnh chính như:

* Kiến trúc tổng thể và các thành phần cấp cao.
* Kiến trúc giao diện người dùng (User Interface) và giao diện quản trị (Admin Interface).
* Kiến trúc Backend và các thành phần con xử lý logic nghiệp vụ, đề xuất, và tương tác với AI.
* Kiến trúc dữ liệu, bao gồm việc sử dụng các loại cơ sở dữ liệu khác nhau.
* Kiến trúc tích hợp với các dịch vụ bên ngoài (ví dụ: hệ thống rạp chiếu phim, Google Calendar).

Tài liệu này không đi sâu vào các chi tiết như yêu cầu nghiệp vụ chi tiết (được trình bày trong các tài liệu yêu cầu), thiết kế giao diện người dùng cụ thể (UI/UX mockups), chi tiết triển khai cấp độ code cho từng lớp (class) hay hàm (function), hoặc kế hoạch quản lý dự án chi tiết (được trình bày trong Tài liệu Kế hoạch Phát triển Phần mềm - SDP), trừ khi chúng có ảnh hưởng trực tiếp hoặc cần thiết để giải thích kiến trúc.

## Định nghĩa, Từ viết tắt

Bảng dưới đây liệt kê các thuật ngữ, định nghĩa và từ viết tắt được sử dụng trong tài liệu này để đảm bảo tính nhất quán và rõ ràng:

|  |  |
| --- | --- |
| **Thuật ngữ / Từ viết tắt** | **Định nghĩa / Giải thích** |
| **SAD** | Software Architecture Document - Tài liệu Kiến trúc Phần mềm |
| **MRAA** | Movie Recommendation with Agent AI - Tên dự án |
| **AI** | Artificial Intelligence - Trí tuệ nhân tạo |
| **API** | Application Programming Interface - Giao diện lập trình ứng dụng |
| **UI** | User Interface - Giao diện người dùng |
| **UX** | User Experience - Trải nghiệm người dùng |
| **SDP** | Software Development Plan - Kế hoạch Phát triển Phần mềm |
| **RUP** | Rational Unified Process - Quy trình phát triển phần mềm |

## Tài liệu Tham khảo

Các tài liệu sau đây đã được sử dụng làm nguồn thông tin hoặc có liên quan đến nội dung của tài liệu này:

* Group 01 - Use-Case Document, Phiên bản 1.0, Ngày 18/04/2025
* Group 01 - Vision Document, Phiên bản 1.0, Ngày 03/04/2025
* Group 01 - SDP (Kế hoạch phát triển phần mềm), Phiên bản 1.0, Ngày 03/04/2025

## Tổng quan về Tài liệu Kiến trúc Phần mềm

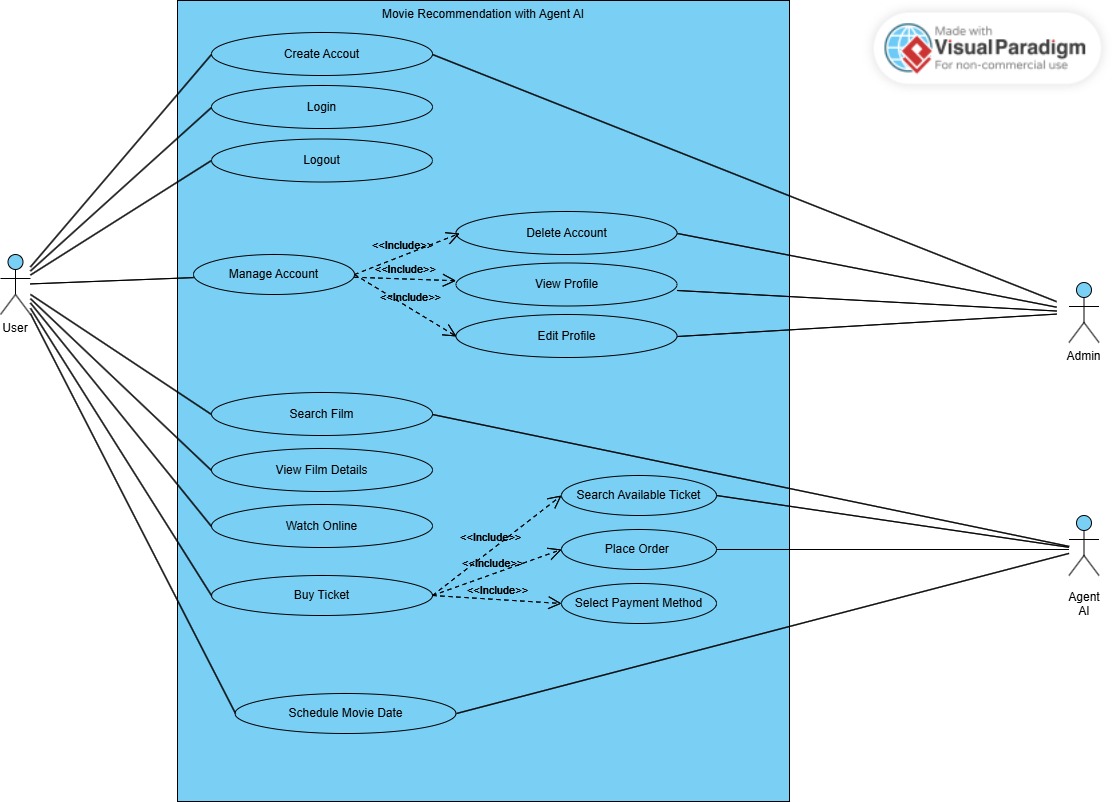
* **Phần 1: Giới thiệu:** Giới thiệu mục đích, phạm vi, định nghĩa, từ viết tắt, tài liệu tham khảo và tổng quan về Tài liệu Kiến trúc Phần mềm.
* **Phần 2: Các mục tiêu và ràng buộc kiến trúc:** Nêu bật các yêu cầu phi chức năng và ràng buộc kỹ thuật có ảnh hưởng đáng kể đến thiết kế kiến trúc.
* **Phần 3: Sơ đồ Use-Case:** Trình bày sơ đồ các trường hợp sử dụng chính của hệ thống để cung cấp bối cảnh nghiệp vụ.
* **Phần 4: Góc nhìn luận lý (Logical View):** Mô tả kiến trúc hệ thống dưới dạng các thành phần chính, các thành phần con và mối quan hệ giữa chúng, tập trung vào logic nghiệp vụ và cấu trúc hệ thống.
* **Phần 5: Triển khai:** Mô tả cách các thành phần luận lý được triển khai trên các môi trường vật lý
* **Phần 6: Góc nhìn ở mức độ cài đặt (Implementation View):** Cung cấp cấu trúc mã nguồn của dự án
* **Phần 7: Mô hình ML (ML Model):** Giới thiệu workflow và quy trình xây dụng mô hình ML

*(Lưu ý: Cấu trúc phần còn lại của tài liệu tuân theo mẫu SAD ban đầu đã cung cấp, có điều chỉnh tên gọi các phần cho phù hợp với nội dung đã điền).*

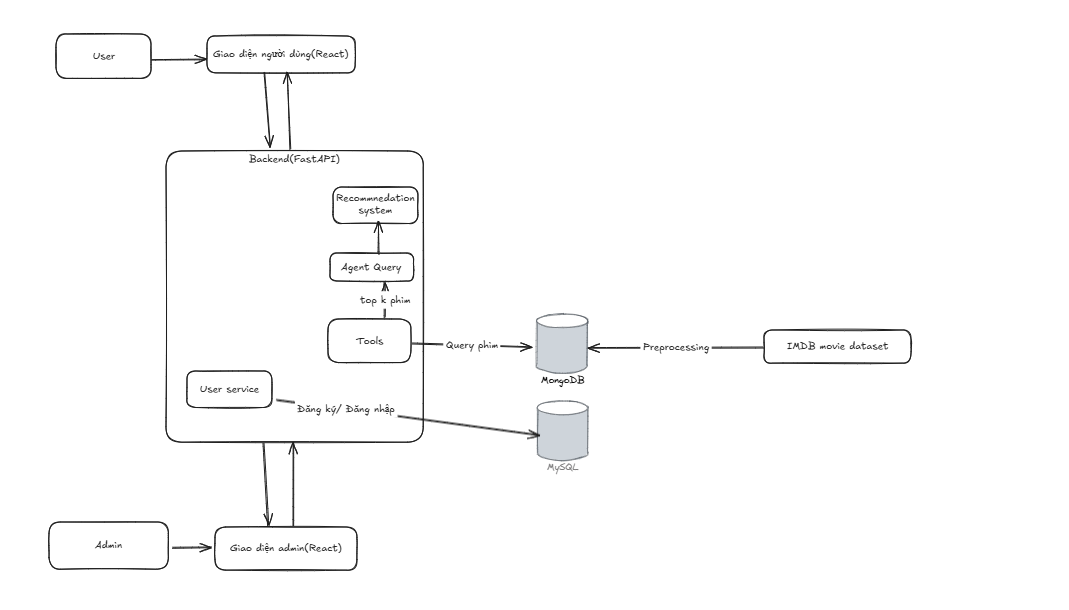
# Các mục tiêu và ràng buộc kiến trúc

* Hệ thống hoạt động trên nền tảng web, có thể phát triển trên các nền tảng khác (nếu có đủ nguồn lực)
* Frontend được phát triển bằng ReactJS, backend sử dụng Python (FastAPI/Flask).
* Giao tiếp giữa frontend và backend qua RestAPI (HTTP/HTTPS).
* Áp dụng kiến trúc microservices để dễ dàng mở rộng theo chiều ngang.
* Hệ thống hỗ trợ tối thiểu 1.000 người dùng đồng thời mà không làm giảm hiệu năng.
* Tốc độ phản hồi:
* Tải trang chủ ≤ 1.5s
* Tìm kiếm phim ≤ 2s với 10.000+ bản ghi
* Chatbot AI phản hồi ≤ 3s với độ chính xác ≥ 85%
* Tuân thủ pháp luật: phải đáp ứng yêu cầu Luật An ninh mạng Việt Nam và GDPR.
* Không lưu trữ thông tin thẻ tín dụng trong hệ thống.
* Cảnh báo độ tuổi phim (13+, 16+, 18+) theo chuẩn phân loại nội dung.
* Người dùng có quyền xóa lịch sử và kiểm soát dữ liệu cá nhân.
* Hỗ trợ đa ngôn ngữ (tiếng Việt và tiếng Anh).
* Hệ thống phải hoạt động trên các trình duyệt phổ biến: Chrome, Firefox, Safari.

# Sơ đồ Use-Case



# Góc nhìn luận lý (Logical View)



## Thành phần: Giao diện người dùng và Quản trị

* **Mô tả:** Bao gồm hai giao diện riêng biệt được xây dựng bằng React: "Giao diện người dùng" cho người dùng cuối và "Giao diện admin" cho quản trị viên.
* **Trách nhiệm/Dịch vụ:** Cung cấp giao diện trực quan cho người dùng và quản trị viên để tương tác với hệ thống. Giao diện người dùng xử lý các tác vụ như tìm kiếm, xem phim, đặt vé, quản lý hồ sơ. Giao diện admin hỗ trợ quản lý hệ thống, người dùng, nội dung, v.v.
* **Phương thức giao tiếp:** Giao tiếp với Hệ thống Backend thông qua các API (dự kiến là HTTP/HTTPS).

## Thành phần: Hệ thống Backend (FastAPI)

* **Mô tả:** Đây là trái tim của hệ thống, được xây dựng bằng framework FastAPI. Backend xử lý các yêu cầu từ giao diện người dùng và quản trị, điều phối các tác vụ, và tương tác với các thành phần khác của hệ thống.
* **Trách nhiệm/Dịch vụ:** Xử lý logic nghiệp vụ chính, quản lý phiên làm việc, điều hướng yêu cầu đến các thành phần con, và giao tiếp với cơ sở dữ liệu và các dịch vụ bên ngoài.
* **Phương thức giao tiếp:** Giao tiếp với Giao diện người dùng/Quản trị (HTTP/HTTPS API), Cơ sở dữ liệu, Agent Query, Recommendation System, và các dịch vụ bên ngoài thông qua Tools.

*Các thành phần con bên trong Hệ thống Backend:*

### Thành phần con: User Service (Đăng ký/Đăng nhập)

* **Mô tả:** Thành phần này chuyên xử lý các nghiệp vụ liên quan đến quản lý người dùng.
* **Trách nhiệm/Dịch vụ:** Đăng ký tài khoản mới, xác thực đăng nhập, quản lý phiên người dùng, và các tác vụ liên quan đến hồ sơ người dùng (chỉnh sửa, xóa - như mô tả trong Use Case).
* **Phương thức giao tiếp:** Hoạt động bên trong Backend, tương tác với các thành phần khác của Backend và Cơ sở dữ liệu (MySQL cho thông tin người dùng).

### Thành phần con: Tools

* **Mô tả:** Thành phần này đóng vai trò là bộ công cụ hoặc adapter, xử lý các tác vụ cụ thể và tích hợp với các nguồn dữ liệu khác. Dựa trên sơ đồ, nó tương tác trực tiếp với cơ sở dữ liệu MongoDB và MySQL. Dựa trên tài liệu Use Case, thành phần này có thể cũng bao gồm logic hoặc tích hợp để tìm kiếm phim, xử lý yêu cầu từ Agent Query, và khả năng tương tác với API của rạp phim cho việc đặt vé.
* **Trách nhiệm/Dịch vụ:** Truy vấn dữ liệu từ cơ sở dữ liệu (MongoDB cho phim, MySQL cho dữ liệu khác), xử lý các yêu cầu cụ thể từ Agent Query và Recommendation System, và có thể bao gồm các adapter để giao tiếp với các dịch vụ bên ngoài (ví dụ: API rạp chiếu phim).
* **Phương thức giao tiếp:** Hoạt động bên trong Backend, giao tiếp với MongoDB, MySQL, Agent Query, và Recommendation System.

### Thành phần con: Agent Query

* **Mô tả:** Thành phần này tiếp nhận và xử lý các truy vấn từ Agent AI, đặc biệt là các truy vấn bằng ngôn ngữ tự nhiên liên quan đến tìm kiếm và đề xuất phim.
* **Trách nhiệm/Dịch vụ:** Phân tích yêu cầu từ Agent AI, chuyển đổi chúng thành các truy vấn mà hệ thống có thể xử lý (thông qua Tools), và trả về kết quả cho Agent AI hoặc Recommendation System.
* **Phương thức giao tiếp:** Hoạt động bên trong Backend, giao tiếp với Tools.

### Thành phần con: Recommendation System

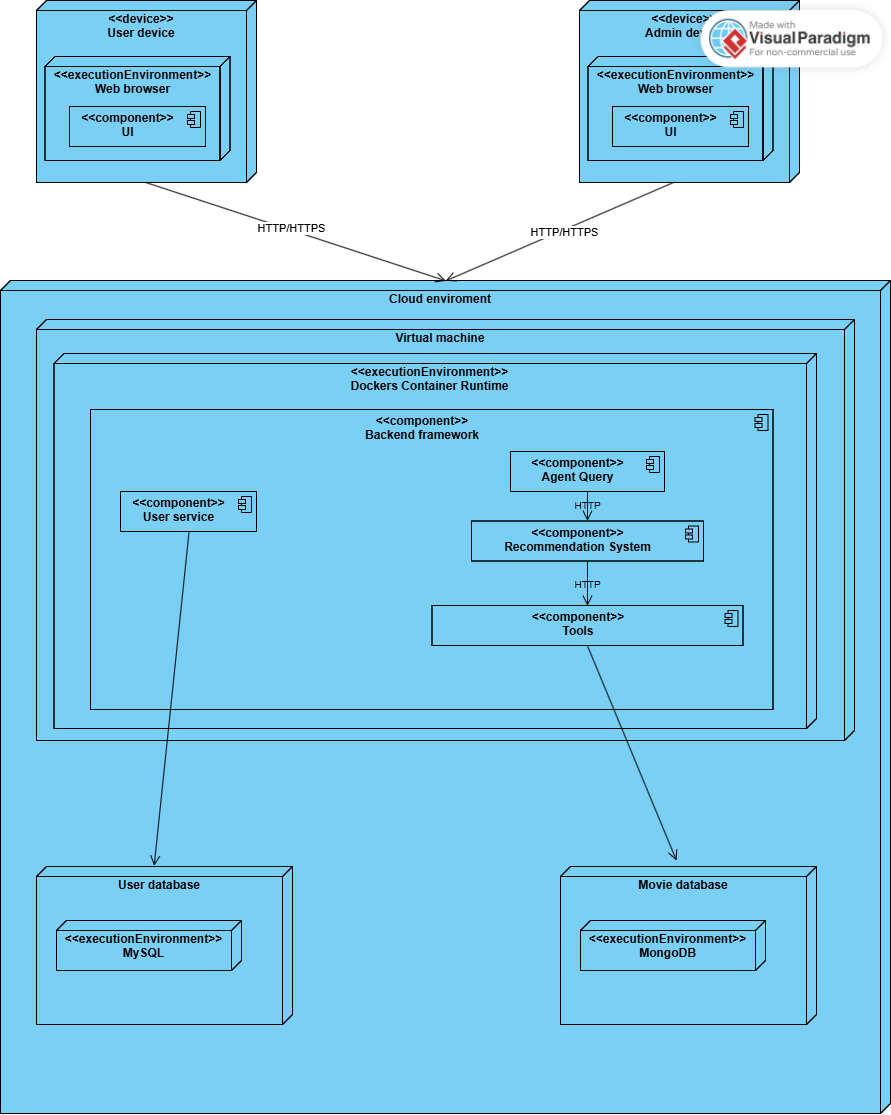
* **Mô tả:** Thành phần này chịu trách nhiệm tạo ra các đề xuất phim cho người dùng.
* **Trách nhiệm/Dịch vụ:** Áp dụng các thuật toán đề xuất (có thể dựa trên lịch sử xem, sở thích, hoặc các yếu tố khác) để gợi ý phim phù hợp cho người dùng. Nó sử dụng dữ liệu từ cơ sở dữ liệu (thông qua Tools) và có thể nhận đầu vào từ Agent Query.
* **Phương thức giao tiếp:** Hoạt động bên trong Backend, giao tiếp với Backend chính và Tools.

## Thành phần: Cơ sở dữ liệu (MongoDB và MySQL)

* **Mô tả:** Hệ thống sử dụng hai loại cơ sở dữ liệu. MongoDB được sử dụng để lưu trữ dữ liệu phim, có thể được tiền xử lý từ các nguồn như IMDB movie dataset. MySQL được sử dụng cho dữ liệu khác, khả năng cao là thông tin người dùng, lịch sử hoạt động, v.v.
* **Trách nhiệm/Dịch vụ:** Lưu trữ bền vững và cung cấp dữ liệu cho Hệ thống Backend và các thành phần con của nó.
* **Phương thức giao tiếp:** Được truy cập bởi thành phần Tools bên trong Backend thông qua các giao thức cơ sở dữ liệu tương ứng.

# Triển khai

## Sơ đồ triển khai



## Mô tả và giải thích

### User device (Thiết bị người dùng)

* **Mô tả:** Đây là thiết bị mà người dùng cuối sử dụng để truy cập hệ thống.
* **Thành phần con (component):**
  + **Web browser (Trình duyệt web):** Môi trường thực thi cho giao diện người dùng.
  + **UI (User Interface - Giao diện người dùng):** Thành phần giao diện người dùng được phát triển bằng ReactJS.
  + **Giao tiếp:** Giao tiếp với môi trường Cloud thông qua HTTP/HTTPS.

### Admin device (Thiết bị quản trị)

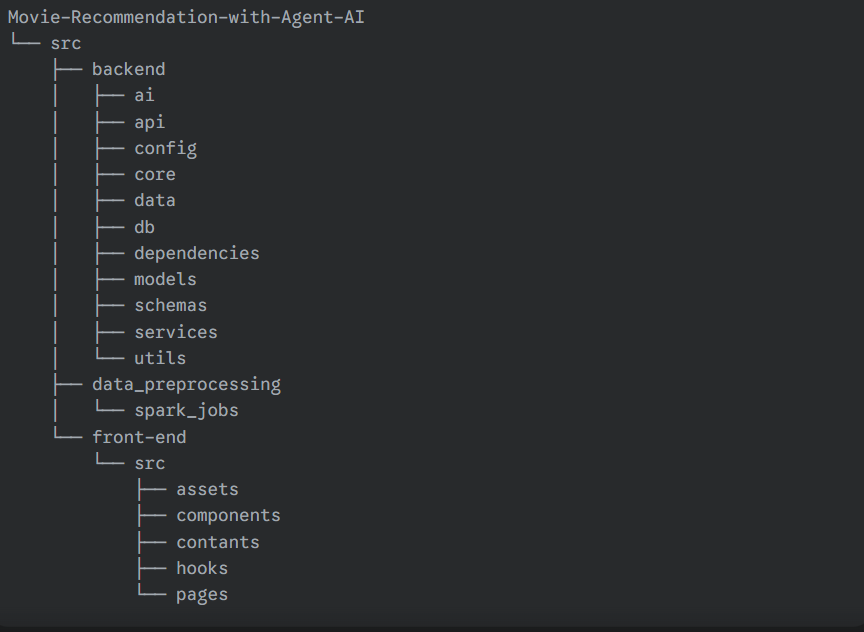
* **Mô tả:** Đây là thiết bị mà quản trị viên sử dụng để truy cập hệ thống.
* **Thành phần con:**
  + **Web browser (Trình duyệt web):** Môi trường thực thi cho giao diện quản trị.
  + **UI (User Interface - Giao diện quản trị):** Thành phần giao diện quản trị được phát triển bằng ReactJS.
  + **Giao tiếp:** Giao tiếp với môi trường Cloud thông qua HTTP/HTTPS.

### Cloud environment (Môi trường đám mây)

* **Mô tả:** Môi trường ảo hóa nơi các thành phần backend và cơ sở dữ liệu được triển khai.
* **Thành phần con:**
  + **Virtual machine (Máy ảo):** Một máy chủ ảo chạy trong môi trường đám mây.
  + **Dockers Container Runtime:** Môi trường thực thi cho các container Docker, cho phép triển khai các dịch vụ một cách đóng gói và linh hoạt.
  + **Backend framework:** Đây là nền tảng backend chính của hệ thống, được xây dựng bằng FastAPI (hoặc Flask). Nó chứa các thành phần logic nghiệp vụ.
  + **User service (Dịch vụ người dùng):** Component chuyên xử lý các nghiệp vụ liên quan đến quản lý người dùng (đăng ký, đăng nhập, quản lý phiên).
  + **Agent Query:** Component tiếp nhận và xử lý các truy vấn từ Agent AI.
  + **Recommendation System (Hệ thống đề xuất):** Component chịu trách nhiệm tạo ra các đề xuất phim cho người dùng.
  + **Tools:** Component đóng vai trò bộ công cụ hoặc adapter, xử lý các tác vụ cụ thể và tích hợp với các nguồn dữ liệu khác.
  + **User database (Cơ sở dữ liệu người dùng):**
  + **MySQL:** Môi trường thực thi cho cơ sở dữ liệu MySQL, lưu trữ thông tin người dùng, lịch sử hoạt động, và sở thích.
  + **Movie database (Cơ sở dữ liệu phim):**
  + **MongoDB:** Môi trường thực thi cho cơ sở dữ liệu MongoDB, lưu trữ dữ liệu phim đã được tiền xử lý.
* **Kết nối giữa các node:**
  + **HTTP/HTTPS:** Các thiết bị người dùng và quản trị giao tiếp với môi trường đám mây sử dụng giao thức HTTP/HTTPS.
  + **Liên kết bên trong Cloud:** Các thành phần bên trong Backend framework (như Agent Query, Recommendation System, Tools) giao tiếp với nhau qua HTTP hoặc các giao thức nội bộ.
  + **Kết nối với cơ sở dữ liệu:** User service kết nối với User database (MySQL), và Tools kết nối với cả User database (MySQL) và Movie database (MongoDB).

# Góc nhìn ở mức độ cài đặt (Implementation View)

## Cấu trúc thư mục hiện tại



## Giải thích cấu trúc thư mục

**src**: Chứa mã nguồn chính của ứng dụng.

* **backend**: Chứa toàn bộ mã nguồn cho phần backend của hệ thống, được phát triển bằng Python (FastAPI/Flask).
  + **ai**: Các module liên quan đến trí tuệ nhân tạo, ví dụ embedder.py cho các tác vụ nhúng (embedding) dữ liệu.
  + **api**: Định nghĩa các endpoint API để giao tiếp với frontend và các dịch vụ khác, bao gồm auth.py cho xác thực và movie.py cho các tác vụ liên quan đến phim.
  + **config**: Chứa các file cấu hình cho ứng dụng, cơ sở dữ liệu (db\_config.py) và các cấu hình liên quan đến nhúng phim (movie\_embedding\_configs.yaml).
  + **core**: Chứa các logic cốt lõi hoặc tiện ích chung của backend, ví dụ auth.py cho các nghiệp vụ xác thực.
  + **data**: Các script liên quan đến việc xử lý và nhập dữ liệu vào cơ sở dữ liệu (ví dụ: import.sh, insert\_elastic\_search\_data.py).
  + **db**: Các module tương tác trực tiếp với các cơ sở dữ liệu khác nhau (MongoDB, MySQL, Redis, Chroma, Elasticsearch), ví dụ mongo\_client.py, crud\_user.py cho các thao tác với người dùng trong MySQL.
  + **dependencies**: Chứa các dependency injection hoặc các phụ thuộc khác của FastAPI.
  + **models**: Định nghĩa các mô hình dữ liệu (ví dụ: user.py cho mô hình người dùng).
  + **schemas**: Định nghĩa các schema dữ liệu (ví dụ: Pydantic schemas) cho API.
  + **services**: Chứa logic nghiệp vụ chính, nơi các yêu cầu được xử lý và điều phối giữa các thành phần khác nhau (ví dụ: auth.py cho logic xác thực, movie.py cho logic liên quan đến phim). Các service này có thể tương ứng với các "thành phần con" như User Service và Recommendation System về mặt logic.
  + **utils**: Các hàm tiện ích chung.
* **data\_preprocessing**: Chứa các script và notebook cho quá trình thu thập và tiền xử lý dữ liệu.
  + **spark\_jobs**: Các script xử lý dữ liệu sử dụng Apache Spark (ví dụ: xử lý dữ liệu IMDB movie dataset).
* **front-end**: Chứa mã nguồn cho giao diện người dùng (UI) và giao diện quản trị (Admin UI), được phát triển bằng ReactJS.
  + **src**: Mã nguồn chính của ứng dụng React.
    - **assets**: Tài nguyên tĩnh như hình ảnh, biểu tượng.
    - **components**: Các thành phần UI có thể tái sử dụng (ví dụ: Header, Footer, Card).
    - **contants**: Các hằng số được sử dụng trong frontend.
    - **hooks**: Các React Hooks tùy chỉnh (ví dụ: useFetch.js cho việc gọi API).
    - **pages**: Các trang chính của ứng dụng (ví dụ: Home, Login, DetailsPage, SearchPage).

# Mô hình ML

## Tổng quan về workflow

Quy trình làm việc tổng thể của hệ thống đề xuất phim dựa trên ML tuân theo một chu trình liên tục nhằm cung cấp các đề xuất chất lượng cao và phù hợp với người dùng. Quy trình này bao gồm các bước chính: thu thập và chuẩn bị dữ liệu từ các nguồn khác nhau, kỹ thuật đặc trưng hóa để biến dữ liệu thô thành định dạng có ý nghĩa cho mô hình, huấn luyện mô hình ML trên tập dữ liệu đã chuẩn bị, đánh giá hiệu suất của mô hình, triển khai mô hình vào môi trường production, và cuối cùng là vận hành, giám sát và bảo trì mô hình để đảm bảo hiệu quả và cập nhật theo thời gian. Mục tiêu cuối cùng là tạo ra các dự đoán (đề xuất phim) được gửi đến người dùng thông qua giao diện trên web.

## Thu thập và tiền xử lý dữ liệu

Bước đầu tiên trong quy trình ML là thu thập dữ liệu. Dữ liệu cho hệ thống đề xuất phim được thu thập từ hai nguồn chính đã xác định:

* **Cơ sở dữ liệu MongoDB:** Chứa dữ liệu về phim, có nguồn gốc từ việc tiền xử lý (preprocessing) các tập dữ liệu phim lớn như IMDB movie dataset. Dữ liệu này bao gồm thông tin chi tiết về phim như tiêu đề, thể loại, diễn viên, đạo diễn, mô tả, v.v.
* **Cơ sở dữ liệu MySQL:** Lưu trữ dữ liệu người dùng và các tương tác của họ với hệ thống và phim. Bao gồm thông tin tài khoản, lịch sử xem, đánh giá (vote sao, bình luận), phim yêu thích, lịch sử đặt vé, và các sở thích đã lưu của người dùng.

Quá trình chuẩn bị dữ liệu bao gồm làm sạch dữ liệu (xử lý các giá trị thiếu, không nhất quán), chuyển đổi định dạng nếu cần, và tích hợp dữ liệu từ hai nguồn này để tạo ra tập dữ liệu huấn luyện đầy đủ.

## Xử lý đặc trưng

Sau khi dữ liệu được thu thập và chuẩn bị, bước tiếp theo là kỹ thuật đặc trưng hóa (Feature Engineering). Mục đích là tạo ra các đặc trưng (features) có ý nghĩa mà mô hình ML có thể sử dụng để học và đưa ra dự đoán. Các loại đặc trưng có thể được tạo bao gồm:

* **Đặc trưng về Phim (Item Features):** Biểu diễn các thuộc tính của phim dưới dạng số hoặc vector (ví dụ: vector one-hot encoding cho thể loại, nhúng (embedding) cho tiêu đề hoặc mô tả, đặc trưng về sự phổ biến).
* **Đặc trưng về Người dùng (User Features):** Biểu diễn sở thích hoặc hành vi của người dùng (ví dụ: vector sở thích dựa trên các thể loại/diễn viên yêu thích, thống kê về số lượng phim đã xem/đánh giá, đặc trưng nhân khẩu học nếu có).
* **Đặc trưng tương tác (Interaction Features):** Biểu diễn mối quan hệ giữa người dùng và phim (ví dụ: ma trận tương tác người dùng-phim ghi lại rating hoặc hành động xem).

Quá trình này rất quan trọng để đảm bảo mô hình có đủ thông tin chất lượng cao để học hiệu quả.

## Huấn luyện mô hình

Bước huấn luyện mô hình là quá trình sử dụng tập dữ liệu đã được đặc trưng hóa để dạy cho thuật toán đề xuất cách đưa ra dự đoán. Mô hình sẽ học các mẫu (patterns) từ dữ liệu, chẳng hạn như mối quan hệ giữa sở thích người dùng và các đặc điểm phim, hoặc sự tương đồng giữa các người dùng hay các phim.

* **Thuật toán:** Ban đầu dữ liệu không có lịch sử của người dùng, vì vậy ý tưởng sẽ là sử dụng content-based để huấn luyện dữ liệu. Người dùng khi tạo tài khoản, sẽ được điều hướng để lựa chọn một số tag (gần tương đương với đặc trưng) nhất định. Mô hình sẽ đề xuất các phim tương đồng nhất với các tag mà người dùng đã chọn. Sau khi ứng dụng được phát hành và dữ liệu người dùng đủ lớn, có thể sẽ chuyển sang sử dụng collaborative filtering để tối ưu hơn trải nghiệm của người dùng.
* **Kiến trúc:** Vì dữ liệu phim người thị trường được thêm mới rất nhanh, nên nhóm lựa chọn mô hình huấn luyện động để phù hợp với thời gian thực, việc thêm mới dữ liệu được thêm vào hằng tuần.
* **Quá trình huấn luyện:** Chia dữ liệu huấn luyện thành các tập train/validation/test. Tập train được đưa vào dữ liệu huấn luyện, điều chỉnh các tham số nội bộ để giảm thiểu lỗi dự đoán và tối ưu hóa khả năng đưa ra đề xuất chính xác.

## Đánh giá mô hình

Đánh giá hiệu suất của Content-based System ở bước đầu (offline) sẽ cần người kiểm thử thủ công để đánh giá (ước tính số tập đánh giá khoảng 100 - 200 bộ). Sau khi ứng dụng phát hành, sẽ đánh giá theo thực tế.

### Các chỉ số đánh giá offline:

* **Precision và Recall (Top-N Recommendation):** Tỷ lệ phim được đề xuất chính xác so với các tag được cung cấp ban đầu.
* **F1-Score:** Kết hợp Precision và Recall.
* **Diversity:** Đánh giá sự đa dạng của các phim được đề xuất trong danh sách Top-N.
* **Novelty:** Đánh giá mức độ "mới lạ" của các đề xuất (liệu có đề xuất những phim mà người dùng ít có khả năng tự tìm thấy không).

### Đánh giá dựa trên thực tế (Online Evaluation):

* **Tỷ lệ nhấp (Click-Through Rate - CTR):** Tỷ lệ người dùng nhấp vào các phim được đề xuất.
* **Thời gian xem/tương tác:** Thời gian người dùng dành cho các phim được đề xuất.
* **Khảo sát người dùng:** Trực tiếp thu thập phản hồi từ người dùng về mức độ hài lòng với các đề xuất.

## Triển khai mô hình

Sau khi mô hình được đánh giá là đạt yêu cầu, nó sẽ được triển khai (deployment) vào môi trường production. Điều này có nghĩa là tích hợp mô hình đã huấn luyện vào thành phần Recommendation System đang chạy trong môi trường thực tế của Backend.

* **Cách triển khai:** Mô hình có thể được triển khai dưới dạng một service độc lập mà Backend gọi đến, hoặc được tích hợp trực tiếp vào code của thành phần Recommendation System.
* **Khả năng mở rộng:** Kiến trúc cần đảm bảo rằng hệ thống đề xuất có thể xử lý lượng lớn yêu cầu dự đoán một cách hiệu quả, phù hợp với yêu cầu về khả năng xử lý đồng thời của hệ thống.

## Vận hành, giám sát và bảo trì

Sau khi triển khai, mô hình ML cần được vận hành, giám sát và bảo trì liên tục:

* **Vận hành:** Mô hình sẵn sàng nhận yêu cầu từ Backend (ví dụ: yêu cầu đề xuất phim cho người dùng khi họ đăng nhập hoặc truy cập trang chủ) và trả về kết quả dự đoán.
* **Giám sát (Monitoring):** Theo dõi hiệu suất của mô hình trong môi trường thực tế. Các chỉ số cần giám sát bao gồm thời gian phản hồi đề xuất, tỷ lệ nhấp (click-through rate) vào các đề xuất, sự thay đổi trong phân phối rating dự đoán, và các lỗi phát sinh. Giám sát giúp phát hiện sớm các vấn đề như suy giảm hiệu suất mô hình (model degradation).
* **Bảo trì (Maintenance):** Định kỳ (hoặc khi cần thiết) thực hiện bảo trì cho mô hình. Điều này có thể bao gồm:
  + **Tái huấn luyện (Retraining):** Huấn luyện lại mô hình với dữ liệu mới (phim mới, tương tác mới của người dùng) để giữ cho các đề xuất luôn tươi mới và chính xác.
  + **Cập nhật thuật toán:** Nghiên cứu và áp dụng các thuật toán đề xuất mới hoặc cải tiến để nâng cao hiệu suất.
  + **Cập nhật đặc trưng:** Điều chỉnh hoặc thêm các kỹ thuật đặc trưng hóa mới.

Quy trình vận hành, giám sát và bảo trì đảm bảo rằng hệ thống đề xuất vẫn hiệu quả theo thời gian và đáp ứng được sự thay đổi trong hành vi người dùng và dữ liệu phim.