

BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC CẦN THƠ  
TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG



LUẬN VĂN TỐT NGHIỆP  
NGÀNH CÔNG NGHỆ THÔNG TIN

Đề tài

THIẾT KẾ VÀ PHÁT TRIỂN HỆ THỐNG ĐẶT VÉ XEM PHIM TRỰC TUYẾN  
SỬ DỤNG KIẾN TRÚC MICROSERVICES VÀ GIAO DIỆN ĐA NỀN TẢNG

Sinh viên: Phan Thanh Thủ Ý

MSSV: B2014795

Khóa: 46

Cần Thơ, 11/2024

BỘ GIÁO DỤC VÀ ĐÀO TẠO  
TRƯỜNG ĐẠI HỌC CẦN THƠ  
TRƯỜNG CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG  
KHOA TRUYỀN THÔNG ĐA PHƯƠNG TIỆN



LUẬN VĂN TỐT NGHIỆP  
NGÀNH CÔNG NGHỆ THÔNG TIN  
CHUYÊN NGÀNH TIN HỌC ỨNG DỤNG

Đề tài

THIẾT KẾ VÀ PHÁT TRIỂN HỆ THỐNG ĐẶT VÉ XEM PHIM TRỰC TUYẾN  
SỬ DỤNG KIẾN TRÚC MICROSERVICES VÀ GIAO DIỆN ĐA NỀN TẢNG

Cán bộ hướng dẫn

ThS. Vũ Duy Linh

Sinh viên: Phan Thanh Thường

MSSV: B2014795

Khóa: 46

Cần Thơ, 11/2024

## LỜI CẢM ƠN

Báo cáo luận văn với chủ đề “Thiết kế và phát triển hệ thống đặt vé xem phim trực tuyến sử dụng kiến trúc microservices và giao diện đa nền tảng” là kết quả của sự cố gắng không ngừng của tôi và sự giúp đỡ, động viên của thầy cô, bạn bè, đồng nghiệp và những người thân thiết.

Qua trang này, tôi xin gửi lời cảm ơn đến những người đã giúp đỡ tôi trong suốt quá trình học tập và nghiên cứu luận văn. Tôi xin bày tỏ lòng kính trọng và biết ơn sâu sắc nhất tới Thạc sĩ thầy Vũ Duy Linh đã trực tiếp hướng dẫn tận tình và cung cấp những thông tin khoa học cần thiết cho luận văn này.

Xin chân thành cảm ơn các cấp lãnh đạo của Trường Đại học Cần Thơ, Trường Công Nghệ Thông Tin, Khoa truyền thông đa phương tiện và của Bộ môn Tin học ứng dụng đã tạo điều kiện cho phép tôi thực hiện luận văn của mình.

Tôi cũng xin cảm ơn các cá nhân và đơn vị công tác đã giúp đỡ tôi trong quá trình học tập và làm báo cáo luận văn.

Cuối cùng, kính chúc các thầy cô luôn luôn khỏe mạnh và tiếp tục có những đóng góp quý báu cho sự phát triển giáo dục và đất nước.

Trân trọng.

*Cần Thơ, ngày ... tháng ... năm 2024*

**Sinh viên thực hiện**

**Phan Thanh Thường**

## **NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Cần Thơ, ngày .... Tháng .... Năm 2024

**Giảng viên hướng dẫn**

(Ký và ghi rõ họ tên)

Ths. Vũ Duy Linh

## NHẬN XÉT CỦA CÁN BỘ PHẢN BIỆN

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

---

Cần Thơ, ngày .... Tháng .... Năm 2024

**Cán bộ phản biện**

(Ký và ghi rõ họ tên)

## MỤC LỤC

|  |      |
|--|------|
| LỜI CẢM ƠN .....   | i    |
| NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN .....                              | ii   |
| NHẬN XÉT CỦA CÁN BỘ PHẢN BIỆN .....                                  | iii  |
| MỤC LỤC.....   | iv   |
| DANH MỤC HÌNH ẢNH .....  | vii  |
| DANH MỤC BẢNG.....   | ix   |
| DANH MỤC KÝ HIỆU VÀ VIẾT TẮT .....                                   | xii  |
| TÓM TẮT .....  | xii  |
| ABSTRACT.....  | xiii |
| PHẦN GIỚI THIỆU .....  | 1    |
| 1.1. Đặt vấn đề .....  | 1    |
| 1.2. Tóm tắt lịch sử giải quyết vấn đề [13] .....                    | 2    |
| 1.3. Mục tiêu đề tài.....  | 3    |
| 1.3.1. Mục tiêu tổng quát .....                                      | 3    |
| 1.3.2. Mục tiêu cụ thể.....  | 3    |
| 1.4. Đối tượng và phạm vi nghiên cứu.....                            | 4    |
| 1.4.1. Đối tượng nghiên cứu .....                                    | 4    |
| 1.4.2. Phạm vi nghiên cứu.....                                       | 4    |
| 1.5. Nội dung nghiên cứu.....  | 4    |
| 1.6. Những đóng góp của chính của đề tài.....                        | 5    |
| 1.7. Bố cục luận văn.....  | 5    |
| PHẦN NỘI DUNG .....  | 6    |
| Chương 1. MÔ TẢ BÀI TOÁN .....                                       | 6    |
| 1.1. Mô tả chi tiết bài toán .....                                   | 6    |
| 1.1.1. Mục tiêu .....  | 6    |
| 1.1.2. Các chức năng của hệ thống .....                              | 8    |
| 1.2. Phân tích đánh giá các giải pháp có liên quan đến bài toán..... | 8    |
| 1.2.1. Phân tích các giải pháp cho phần back-end của hệ thống.....   | 8    |

|   |    |
|---|----|
| 1.2.2. Phân tích các giải pháp phần giao diện của hệ thống .....  | 11 |
| 1.3. Tiếp cận giải quyết vấn đề, chọn lựa giải pháp.....          | 14 |
| Chương 2. THIẾT KẾ VÀ CÀI ĐẶT GIẢI PHÁP .....                     | 16 |
| 2.1. Thiết kế kiến trúc .....                                     | 16 |
| 2.1.1. Các vấn đề thiết kế kiến trúc Microservices [11][14] ..... | 16 |
| 2.1.2. Giải pháp kiến trúc .....                                  | 16 |
| 2.2. Sơ đồ Use case của hệ thống.....                             | 19 |
| 2.2.1. Sơ đồ Use Case tổng quát của hệ thống.....                 | 20 |
| 2.2.2. Sơ đồ Use Case của khách vãng lai .....                    | 21 |
| 2.2.3. Sơ đồ Use case của khách hàng đã có tài khoản.....         | 21 |
| 2.2.4. Sơ đồ Use Case của ADMIN .....                             | 22 |
| 2.3. Thiết kế cơ sở dữ liệu.....                                  | 23 |
| 2.3.1. Cơ sở dữ liệu của Auth Service.....                        | 24 |
| 2.3.2. Cơ sở dữ liệu của Movie Service .....                      | 28 |
| 2.3.3. Cơ sở dữ liệu của Seat Service .....                       | 35 |
| 2.3.4. Cơ sở dữ liệu của Reservation Service .....                | 38 |
| 2.3.5. Cơ sở dữ liệu của Recommend Service .....                  | 42 |
| 2.3.6. Mô hình CDM toàn hệ thống .....                            | 43 |
| 2.4. Thiết kế giao diện.....                                      | 45 |
| 2.4.1. Mục tiêu .....   | 45 |
| 2.4.2. Cấu trúc các màn hình chính.....                           | 45 |
| 2.4.3. Minh họa .....   | 46 |
| 2.5. Thiết kế các dịch vụ chính .....                             | 51 |
| 2.5.1. Movie Service .....  | 51 |
| 2.5.2. Authenticate Service .....                                 | 61 |
| 2.5.3. Reservation Service.....                                   | 66 |
| 2.5.4. Seat Service.....  | 69 |
| 2.5.5. Recommend Service .....                                    | 71 |
| 2.5.6. Gateway Service .....                                      | 75 |
| 2.5.7. Media Service .....  | 75 |

|   |    |
|---|----|
| 2.6. Giao tiếp giữa các services.....                   | 76 |
| Chương 3. ĐÁNH GIÁ VÀ KIỂM THỦ.....                     | 77 |
| 3.1. Giới thiệu .....                                   | 77 |
| 3.1.1. Mục tiêu kiểm thử .....                          | 77 |
| 3.1.2. Phạm vi kiểm thử .....                           | 77 |
| 3.2. Chi tiết kế hoạch kiểm thử .....                   | 77 |
| 3.2.1. Các chức năng được kiểm thử.....                 | 77 |
| 3.2.2. Cách tiếp cận .....                              | 77 |
| 3.2.3. Các tiêu chí kiểm thử thành công/ thất bại ..... | 78 |
| 3.3. Quản lý kiểm thử.....                              | 78 |
| 3.3.1. Kế hoạch kiểm thử .....                          | 78 |
| 3.3.2. Môi trường .....                                 | 78 |
| 3.3.3. Trách nhiệm và quyền hạn .....                   | 78 |
| 3.4. Kịch bản kiểm thử.....                             | 79 |
| PHẦN KẾT LUẬN .....                                     | 96 |
| 1. Kết quả đạt được .....                               | 96 |
| 2. Hướng phát triển .....                               | 98 |
| TÀI LIỆU THAM KHẢO.....                                 | 99 |

## DANH MỤC HÌNH ẢNH

|   |    |
|---|----|
| Hình 1-1: Kiến trúc Microservices của X(Twitter) .....                    | 7  |
| Hình 1-2: Kiến trúc Monolithic và kiến trúc microservices .....           | 9  |
| Hình 1-3: Kiến trúc microservices .....                                   | 10 |
| Hình 1-4: Next.js và Angular .....  | 13 |
| Hình 2-1: Sơ đồ tổng quan dự án .....                                     | 18 |
| Hình 2-2: Sơ đồ Use Case tổng quát .....                                  | 20 |
| Hình 2-3: Sơ đồ Use case của khách vãng lai .....                         | 21 |
| Hình 2-4: Sơ đồ Use case của khách hàng đã có tài khoản .....             | 22 |
| Hình 2-5: Sơ đồ Use case ADMIN .....                                      | 23 |
| Hình 2-6: Cơ sở dữ liệu của Auth Service .....                            | 24 |
| Hình 2-7: Một document của Auth service dưới dạng đối tượng .....         | 25 |
| Hình 2-8: Cơ sở dữ liệu của Movie Service .....                           | 28 |
| Hình 2-9: Một document của Movie service dưới dạng đối tượng .....        | 29 |
| Hình 2-10: Cơ sở dữ liệu Seat Service .....                               | 35 |
| Hình 2-11: Một document của Seat service dưới dạng đối tượng .....        | 35 |
| Hình 2-12: Cơ sở dữ liệu Reservation Service .....                        | 38 |
| Hình 2-13: Một document của Reservation service dưới dạng đối tượng ..... | 39 |
| Hình 2-14: Cơ sở dữ liệu của Recommend Service .....                      | 42 |
| Hình 2-15: Sơ đồ thực thể quan hệ .....                                   | 44 |
| Hình 2-16: Giao diện trang Home .....                                     | 46 |
| Hình 2-17: Giao diện trang Schedule .....                                 | 46 |
| Hình 2-18: Giao diện trang Tickets .....                                  | 47 |
| Hình 2-19: Giao diện trang Profile .....                                  | 47 |
| Hình 2-20: Giao diện trang Favorite .....                                 | 48 |
| Hình 2-21: Giao diện mobile trang Home .....                              | 48 |
| Hình 2-22: Giao diện mobile trang Schedule .....                          | 49 |
| Hình 2-23: Giao diện mobile trang Tickets .....                           | 49 |
| Hình 2-24: Giao diện mobile trang Profile .....                           | 50 |
| Hình 2-25: Giao diện mobile trang Favorite .....                          | 50 |
| Hình 2-26: Giao diện trang Thêm mới một bộ phim .....                     | 54 |
| Hình 2-27: Giao diện trang Search .....                                   | 55 |
| Hình 2-28: Giao diện Chi tiết phim .....                                  | 58 |
| Hình 2-29: Giao diện Sắp lịch .....                                       | 59 |
| Hình 2-30: Giao diện Bình luận .....                                      | 60 |
| Hình 2-31: Giao diện Đăng ký .....  | 62 |
| Hình 2-32: Giao diện Đăng nhập .....                                      | 63 |
| Hình 2-33: Giao diện Quản lý tài khoản .....                              | 65 |

|   |    |
|---|----|
| Hình 2-34: Giao diện Thông tin vé .....                             | 67 |
| Hình 2-35: Giao diện Chi tiết hóa đơn.....                          | 68 |
| Hình 2-36: Giao diện Chọn ghế .....                                 | 70 |
| Hình 2-37: TF-IDF (Term Frequency-Inverse Document Frequency) ..... | 72 |
| Hình 2-38: Cosine Similarity .....                                  | 73 |
| Hình 2-39: Giao diện Thanh tìm kiếm .....                           | 74 |
| Hình 2-40: Giao tiếp giữa các services .....                        | 76 |

## DANH MỤC BẢNG

|  |    |
|--|----|
| Bảng 2-1: Thực thể User .....  | 25 |
| Bảng 2-2: Thực thể Authentication .....                                    | 25 |
| Bảng 2-3: Bảng kèm ảnh hưởng ràng buộc liên thuộc tính bảng User.....      | 26 |
| Bảng 2-4: Bảng kèm ảnh hưởng ràng buộc liên bộ bảng User.....              | 26 |
| Bảng 2-5: Bảng kèm ảnh hưởng ràng buộc liên thuộc tính bảng Auth .....     | 27 |
| Bảng 2-6: Bảng kèm ảnh hưởng ràng buộc liên bộ bảng Auth.....              | 27 |
| Bảng 2-7: Thực thể Movie .....   | 29 |
| Bảng 2-8: Thực thể Genre.....  | 31 |
| Bảng 2-9: Thực thể Detail Showing Type.....                                | 31 |
| Bảng 2-10: Thực thể Gallery.....   | 31 |
| Bảng 2-11: Thực thể User Favorite Movie .....                              | 32 |
| Bảng 2-12: Thực thể Showing .....  | 32 |
| Bảng 2-13: Thực thể Comment.....   | 32 |
| Bảng 2-14: Bảng kèm ảnh hưởng ràng buộc liên thuộc tính bảng Movie.....    | 33 |
| Bảng 2-15: Bảng kèm ảnh hưởng ràng buộc liên bộ bảng Movie .....           | 34 |
| Bảng 2-16: Bảng kèm ảnh hưởng ràng buộc liên thuộc tính bảng Showing ..    | 34 |
| Bảng 2-17: Bảng kèm ảnh hưởng ràng buộc liên bộ bảng Showing .....         | 34 |
| Bảng 2-18: Thực thể Auditorium .....                                       | 36 |
| Bảng 2-19: Thực thể Seat.....  | 36 |
| Bảng 2-20: Thực thể Seat_Detail .....                                      | 36 |
| Bảng 2-21: Bảng kèm ảnh hưởng ràng buộc liên thuộc tính bảng Seat .....    | 37 |
| Bảng 2-22: Bảng kèm ảnh hưởng ràng buộc liên thuộc bộ bảng Seat.....       | 37 |
| Bảng 2-23: Bảng kèm ảnh hưởng ràng buộc liên thuộc tính bảng SeatDetail    | 38 |
| Bảng 2-24: Bảng kèm ảnh hưởng ràng buộc liên bộ bảng SeatDetail.....       | 38 |
| Bảng 2-25: Thực thể Ticket.....  | 39 |
| Bảng 2-26: Thực thể Order .....  | 40 |
| Bảng 2-27: Thực thể Payment.....   | 40 |
| Bảng 2-28: Bảng kèm ảnh hưởng ràng buộc liên thuộc tính bảng Order .....   | 41 |
| Bảng 2-29: Bảng kèm ảnh hưởng ràng buộc liên bộ bảng Order .....           | 41 |
| Bảng 2-30: Bảng kèm ảnh hưởng ràng buộc liên thuộc tính bảng Ticket .....  | 42 |
| Bảng 2-31: Bảng kèm ảnh hưởng ràng buộc liên bộ bảng Ticket .....          | 42 |
| Bảng 2-32: Thực thể Keyword.....   | 43 |
| Bảng 2-33: Thành phần giao diện chức năng Thêm mới bộ phim.....            | 54 |
| Bảng 2-34: Dữ liệu được sử dụng của chức năng Thêm mới bộ phim.....        | 54 |
| Bảng 2-35: Thành phần giao diện chức năng Truy xuất danh sách phim.....    | 56 |
| Bảng 2-36: Dữ liệu được sử dụng của chức năng Tim kiếm phim.....           | 56 |
| Bảng 2-37: Dữ liệu được sử dụng của chức năng Truy xuất chi tiết phim .... | 58 |

|  |    |
|--|----|
| Bảng 2-38: Thành phần giao diện chức năng Sắp lịch .....                   | 59 |
| Bảng 2-39 : Dữ liệu được sử dụng của chức năng Sắp lịch.....               | 59 |
| Bảng 2-40: Thành phần giao diện chức năng Thêm mới bình luận.....          | 60 |
| Bảng 2-41: Dữ liệu được sử dụng của chức năng Thêm mới bình luận .....     | 61 |
| Bảng 2-42: Thành phần giao diện chức năng Đăng ký.....                     | 62 |
| Bảng 2-43: Dữ liệu được sử dụng của chức năng Đăng ký .....                | 62 |
| Bảng 2-44: Thành phần giao diện chức năng Đăng nhập .....                  | 64 |
| Bảng 2-45: Dữ liệu được sử dụng của chức năng Đăng nhập .....              | 64 |
| Bảng 2-46: Thành phần giao diện chức năng Quản lý tài khoản.....           | 65 |
| Bảng 2-47: Dữ liệu được sử dụng của chức năng Quản lý tài khoản .....      | 65 |
| Bảng 2-48: Thành phần giao diện chức năng Truy xuất thông tin vé.....      | 67 |
| Bảng 2-49: Dữ liệu được sử dụng của chức năng Truy xuất thông tin vé ..... | 67 |
| Bảng 2-50: Thành phần giao diện chức năng Thanh toán lại .....             | 68 |
| Bảng 2-51: Dữ liệu được sử dụng của chức năng Thanh toán lại.....          | 68 |
| Bảng 2-52: Thành phần giao diện chức năng Chọn ghế thời gian thực.....     | 70 |
| Bảng 2-53: Dữ liệu được sử dụng của chức năng Chọn ghế thời gian thực ..   | 71 |
| Bảng 2-54: Thành phần giao diện chức năng Gọi ý từ khóa tìm kiếm.....      | 74 |
| Bảng 2-55: Dữ liệu được sử dụng của chức năng Gọi ý từ khóa tìm kiếm ...   | 75 |
| Bảng 3-1: Trách nhiệm và quyền hạn kiểm thử .....                          | 78 |
| Bảng 3-2: Kịch bản kiểm thử chi tiết chức năng Đăng ký.....                | 80 |
| Bảng 3-3: Kịch bản kiểm thử chi tiết chức năng Đăng nhập.....              | 82 |
| Bảng 3-4: Kịch bản kiểm thử chi tiết chức năng Đăng nhập với Google.....   | 84 |
| Bảng 3-5: Kịch bản kiểm thử chi tiết chức năng Đặt vé.....                 | 85 |
| Bảng 3-6: Kịch bản kiểm thử chi tiết chức năng Update thông tin user.....  | 86 |
| Bảng 3-7: Kịch bản kiểm thử chi tiết chức năng Thanh toán lại .....        | 88 |
| Bảng 3-8: Kịch bản kiểm thử chi tiết chức năng Bình luận.....              | 90 |
| Bảng 3-9: Kịch bản kiểm thử chi tiết chức năng Sắp lịch .....              | 91 |
| Bảng 3-10: Kịch bản kiểm thử chi tiết chức năng Thêm mới bộ phim .....     | 92 |
| Bảng 3-11: Kịch bản kiểm thử chi tiết Giao tiếp giữa các services .....    | 94 |

### DANH MỤC KÝ HIỆU VÀ VIẾT TẮT

| STT | Ký hiệu | Ý nghĩa                                   |
|-----|---------|---|
| 1   | UI      | User interface                            |
| 2   | UX      | User Experience                           |
| 3   | HTML    | HyperText Markup Language                 |
| 4   | CSS     | Cascading Style Sheets                    |
| 5   | JS      | Javascript                                |
| 6   | MVC     | Model-View-Controller                     |
| 7   | SSR     | Server Side Rendering                     |
| 8   | SSG     | Static Site Generation                    |
| 9   | SEO     | Search Engine Optimization                |
| 10  | API     | Application Programming Interface         |
| 11  | JWT     | JSON Web Token                            |
| 12  | MRR     | Mean Reciprocal Rank                      |
| 13  | TF-IDF  | Term Frequency-Inverse Document Frequency |
| 14  | SQL     | Structured Query Language                 |
| 15  | DB      | Database                                  |

## TÓM TẮT

Trong thời đại công nghệ, sự tiện lợi là chìa khóa mở ra cơ hội kinh doanh mới. Ngành giải trí, đặc biệt là lĩnh vực xem phim, đã thay đổi mạnh mẽ nhờ cuộc cách mạng công nghệ, đáp ứng nhu cầu trải nghiệm linh hoạt, dễ dàng của người tiêu dùng.

Đề tài này nhằm xây dựng một website giúp người dùng tìm kiếm, đặt vé xem phim nhanh chóng, không cần chờ đợi. Website cung cấp giao diện trực quan, hiện đại, cùng hệ thống chọn ghế thời gian thực, nâng cao trải nghiệm mua vé.

Hệ thống gồm hai phần chính: giao diện được phát triển bằng Next.js, Angular và các thư viện UI, đảm bảo thân thiện, hiệu suất cao; phía máy chủ sử dụng Spring Boot, kiến trúc Microservices, kết hợp MongoDB, MySQL và các công nghệ như Kafka, Redis, WebSocket.

Dự án tích hợp các tính năng nổi bật như chọn ghế trực tuyến thời gian thực, thanh tìm kiếm gợi ý thông minh và giao diện tối ưu hiệu năng. Kiến trúc Microservices giúp dễ dàng bảo trì, nâng cấp, và mở rộng, nhờ khả năng phân tách logic giữa các dịch vụ.

**Từ khóa:** Đặt vé xem phim, Hệ thống đặt vé trực tuyến, Đặt phim, Moon Movie.

## ABSTRACT

In today's technological era, convenience is the key to unlocking new business opportunities. The entertainment industry, especially the cinema sector, has undergone significant transformation thanks to the technological revolution, meeting consumers' demands for flexible and seamless experiences.

This project aims to develop a website that allows users to easily search for and book movie tickets without waiting. The website offers a modern, user-friendly interface and a real-time seat selection system to enhance the ticket booking experience.

The system consists of two main components: the interface, developed using Next.js, Angular, and UI libraries to ensure a friendly and high-performance user experience; and the backend, built with Spring Boot and a Microservices architecture, integrating MongoDB, MySQL, and technologies like Kafka, Redis, and WebSocket.

The project includes standout features such as real-time online seat selection, an intelligent keyword suggestion search bar, and a performance-optimized modern interface. The Microservices architecture facilitates easy maintenance, upgrades, and scalability by enabling logical separation between services.

**Keywords:** Movie ticket booking, Online ticket booking system, Movie booking system, Moon Movie.

## PHẦN GIỚI THIỆU

### 1.1. Đặt vấn đề

Trong bối cảnh ngành công nghiệp giải trí ngày càng phát triển, việc phân phối và đặt vé xem phim qua các nền tảng trực tuyến đã trở thành một phần thiết yếu. Giả sử rằng các bước hợp đồng với các nhà làm phim đã được ký kết và hoàn tất, hệ thống này sẽ tập trung vào khía cạnh quản lý và đăng tải các bộ phim và phân phối các suất chiếu phim. Điều này cho phép hệ thống bỏ qua các vấn đề liên quan đến bản quyền, coi đây là một yếu tố tiên đề đã được giải quyết từ trước.

Hệ thống này nhằm mục đích giải quyết các hạn chế của các hệ thống cũ. Chẳng hạn, giao diện không thân thiện, trải nghiệm người dùng không được tối ưu, các chức năng thiếu tính linh hoạt và thiếu tích hợp với các công nghệ hiện đại. Những hạn chế này khiến người dùng gặp khó khăn trong việc tìm kiếm phim phù hợp, hoặc là không thể sử dụng hệ thống một cách liền mạch.

Hệ thống bán vé xem phim là một hệ thống cung cấp cho người dùng có thể đặt vé xem ở bất cứ đâu và vào bất cứ thời gian nào mà không cần phải đòi hỏi nhiều thủ tục quá rườm rà, phức tạp và mệt mỏi. Người dùng chỉ cần một thiết bị điện tử, một kết nối mạng ổn định là có thể bắt đầu quá trình đặt vé xem phim. Qua đó có thể tối ưu quy trình đặt vé và nâng cao trải nghiệm của người dùng mỗi khi họ muốn đi xem phim. Bên cạnh đó hệ thống còn cung cấp các chức năng quản lý thông dụng giúp cho quá trình quản lý trở nên hiệu quả, tiết kiệm đáng kể thời gian của các doanh nghiệp.

Với giao diện được thiết kế với ngôn ngữ thiết kế hiện đại thay vì các giao diện cổ điển và thô sơ như các website hiện nay. Không chỉ cung cấp giao diện đẹp mắt mà còn đem lại năng ổn định và trên hết đáp ứng trên mọi thiết bị giúp cho người dùng không cần thay đổi thói quen sử dụng công nghệ của mình.

Dự án “Thiết kế và phát triển hệ thống đặt vé xem phim trực tuyến sử dụng kiến trúc microservices và giao diện đa nền tảng” được trang bị thêm nhiều chức năng nổi bật để nâng cao đáng kể trải nghiệm người dùng như là: gợi ý từ khóa thông minh thông qua mô hình TF-IDF và Cosine Similarity để hỗ trợ người dùng tìm tên phim nhanh chóng và chính xác, đặt vé thời gian thực sử dụng websocket đem lại giao diện tương tác trực tiếp giữa các người dùng.

Hệ thống sử dụng kiến trúc Microservices (Microservices Architecture) như một giải pháp tối ưu cho bài toán xây dựng và phát triển các ứng dụng đa dạng dịch vụ như

hiện nay. Để đem lại khả năng mở rộng độc lập và dễ dàng phát triển hệ thống trong tương lai.

## 1.2. Tóm tắt lịch sử giải quyết vấn đề /13/

### Giai đoạn đầu (Trước năm 2010):

- Xuất hiện các trang web đặt vé xem phim trực tuyến tiên phong như: Lotte Cinema: <https://www.lottecinemavn.com/LCHS/Contents/Movie/Movie-List.aspx>, CGV: <https://www.cgv.vn/>, Galaxy Cinema,...
- Chức năng chính: Cung cấp thông tin phim, rạp chiếu và đặt vé đơn giản.
- Hạn chế: Hệ thống chưa hoàn thiện, ít người sử dụng do thói quen mua vé trực tiếp và e ngại thanh toán trực tuyến.

### Giai đoạn phát triển (2010 - 2020):

- Sự bùng nổ của smartphone và internet thúc đẩy sự phát triển của thương mại điện tử, trong đó có bán vé xem phim trực tuyến.
- Xuất hiện nhiều ứng dụng đặt vé như: MoMo: <https://momo.vn/>, Galaxy Play: <https://galaxyplay.vn/>....
- Nâng cấp tính năng: Thanh toán đa dạng, ưu đãi hấp dẫn, tích điểm đổi quà,... Tuy vậy giao diện vẫn còn khá đơn giản.
- Thống kê:
  - Năm 2017, doanh thu bán vé xem phim trực tuyến tại Việt Nam đạt 110 tỷ đồng, tăng 30% so với năm 2016. (<https://vtcnews.vn/>)
  - Năm 2019, 70% vé xem phim được bán trực tuyến. (<https://tuoitre.vn/>)

### Giai đoạn bùng nổ (2020 - nay):

- Dịch COVID-19 thúc đẩy thói quen mua sắm online, trong đó có vé xem phim.
- Các nền tảng OTT (Over-the-top) như: Netflix: <https://www.netflix.com/>, Disney+: <https://www.disneyplus.com/>, Galaxy Play: <https://galaxyplay.vn/>,... cung cấp dịch vụ xem phim trực tuyến kèm theo gói vé xem phim tại rạp.
- Áp dụng công nghệ tiên tiến: AI, Big Data,... để cá nhân hóa trải nghiệm người dùng, đề xuất phim phù hợp.

- Thông kê:

- o Năm 2021, thị trường bán vé xem phim trực tuyến Việt Nam đạt 300 tỷ đồng, tăng 170% so với năm 2020. (<https://tuoitre.vn/>)
- o Dự kiến đến năm 2025, thị trường này sẽ đạt 1 tỷ USD. (<https://vtcnews.vn/>)

### 1.3. Mục tiêu đề tài

Dự án “Thiết kế và phát triển hệ thống đặt vé xem phim trực tuyến sử dụng kiến trúc microservices và giao diện đa nền tảng” được thiết kế nhằm mục đích cung cấp những trải nghiệm mới mẻ, chuyên nghiệp, thuận lợi và linh hoạt và hoàn toàn là trực tuyến cho khách hàng. Ngoài ra nó còn cung cấp sự dễ dàng và tiện lợi cho việc quản lý các bộ phim và suất chiếu đối với vai trò của những người vận hành. Đồng thời thông qua dự án, bản thân người xây dựng đề tài cũng sẽ tìm hiểu và học hỏi được một số công nghệ mới để có thể xây dựng một hệ thống Microservices như: Spring Boot, API Gateway, Eureka Server, Kafka, ... Từ đó đưa ra các phân tích, đánh giá và học hỏi về kiến trúc Microservices cũng như cách thiết kế và vận hành một hệ thống theo kiến trúc Microservices hiệu quả.

#### 1.3.1. Mục tiêu tổng quát

Xây dựng và phát triển website hỗ trợ đặt vé xem phim: front-end Next.js và Angular, back-end Spring Boot theo kiến trúc Microservices.

#### 1.3.2. Mục tiêu cụ thể

- Tìm hiểu cách thức xây dựng website với các công nghệ Spring Boot, Next.js và Angular.
- Tìm hiểu cách thức thiết kế và xây dựng back-end một website theo kiến trúc Microservices.
- Tìm hiểu cách phát triển các giao diện hiện đại hiệu suất cao với các framework Next.js và Angular.
- Tạo một website nơi mà khách hàng có thể tìm kiếm và đặt vé trực tiếp bằng giao diện hiện đại, trực quan và thân thiện, có thể thanh toán trực tiếp trên website.
- Tích hợp thanh toán trực tuyến ZaloPay và VNPay để tối ưu hóa trải nghiệm mua sắm trực tuyến của khách hàng.
- Tích hợp một vài giải thuật AI để tăng trải nghiệm của người dùng khi sử dụng website.

## 1.4. Đối tượng và phạm vi nghiên cứu

### 1.4.1. Đối tượng nghiên cứu

Đề tài sẽ tập trung vào nghiên cứu các đối tượng sau:

- Nghiên cứu về quy trình, nghiệp vụ xử lý các thao tác đặt vé và nhận vé online qua hệ thống.
- Nghiên cứu về công nghệ lập trình, cơ sở dữ liệu phù hợp để xây dựng nền các tính năng cần thiết cho sản phẩm.
- Nghiên cứu cách tích hợp Zalopay và VNPay vào website để hỗ trợ thanh toán trực tuyến.
- Nghiên cứu các công nghệ Spring Boot, Next.js, Angular, MongoDB, MySql,...
- Nghiên cứu cách triển khai và vận hành một hệ thống Microservices.

### 1.4.2. Phạm vi nghiên cứu

Về mặt lý thuyết:

- Tìm hiểu cách xây dựng một RestfulAPI thông qua công nghệ Spring Boot.
- Tìm hiểu cách sử dụng Next.js, Angular và các thư viện có liên quan để xây dựng các website có giao diện thân thiện, dễ sử dụng và hiệu suất cao.
- Tìm hiểu cách sử dụng và thiết kế cơ sở dữ liệu MongoDB và MySql.
- Tìm hiểu cách thiết kế và triển khai kiến trúc Microservices.
- Tìm hiểu về các hình thức thanh toán online thông qua các ví điện tử Zalopay, VNPay.

Về mặt lập trình:

- Backend: Sử dụng Spring Boot kết hợp với Microservices để thiết kế, xây dựng nền các API phục vụ cho quá trình xử lý tính toán của website.
- Frontend: Sử dụng Next.js kết hợp với thư viện ui Shadcn/ui để xây dựng giao diện người dùng, Angular và Angular Material để xây dựng giao diện quản lý ở phía Admin.
- Database: Sử dụng MongoDB và MySql để lưu dữ liệu.

## 1.5. Nội dung nghiên cứu

- Quy trình đặt vé xem phim trực tuyến thực tế.
- Quy trình xây dựng hệ thống đặt vé xem phim.
- Công nghệ lập trình với Spring Boot và Spring Ecosystem.

- Công nghệ lập trình với Next.js, Angular.
- Cơ sở dữ liệu MongoDB và MySQL.
- Cách sử dụng Zalopay và VNPay APIs.
- Nghiên cứu về cách thức triển khai kiến trúc Microservices và cách thức kết nối hay giao tiếp giữa các Services với nhau.

### 1.6. Những đóng góp của chính của đề tài

Về mặt xã hội: Dự án đặt mục tiêu cung cấp một nền tảng trực tuyến hoàn chỉnh cho việc mua vé xem phim. Bằng cách này, hy vọng người dùng sẽ nâng cao được trải nghiệm, tăng cường hiệu quả vận hành và thúc đẩy sự phát triển của ngành công nghiệp giải trí.

Về mặt giáo dục: Cung cấp tài liệu hữu ích để các khóa sau có thể tham khảo, vận dụng và mở rộng đề tài.

Về mặt kinh tế: quảng bá các sản phẩm giải trí rộng rãi đến mọi người, quảng bá một cách thức hiệu quả trong việc đặt vé xem phim, thúc đẩy ngành công nghiệp giải trí mạnh trên cả nước.

### 1.7. Bộ cục luận văn

Bộ cục của quyển luận văn gồm ba phần: phần giới thiệu, phần nội dung và phần kết luận. Phần giới thiệu sẽ đặt ra vấn đề, lịch sử giải quyết vấn đề cho biết tại sao cần thực hiện đề tài, giới thiệu các nội dung nghiên cứu để thực hiện và nêu những đóng góp chính của đề tài. Tiếp đến là phần nội dung, phần này sẽ mô tả chi tiết bài toán cần thực hiện những gì, sau đó phân tích và đưa ra đánh giá cho từng giải pháp, trình bày các thiết kế về kiến trúc hệ thống và dữ liệu. Mô tả chi tiết, cách xử lý của những chức năng, đồng thời sẽ lên kế hoạch cụ thể, tiến hành thực hiện đánh giá kiểm thử cho hoạt động của từng chức năng. Phần kết luận sẽ tổng kết lại các kết quả đã đạt được của đề tài, cũng như nêu những hạn chế, khó khăn đã gặp trong quá trình thực hiện, đồng thời nêu lên hướng phát triển của đề tài trong tương lai. Mục tài liệu tham khảo sẽ trình bày các nguồn tài liệu đã được tham khảo. Ngoài ra, trước phần giới thiệu sẽ là phần mục lục, danh mục bảng và hình ảnh được trình bày trong ba phần chính.

## PHẦN NỘI DUNG

### Chương 1. MÔ TẢ BÀI TOÁN

#### 1.1. Mô tả chi tiết bài toán

##### 1.1.1. Mục tiêu

Dự án đặt mục tiêu cung cấp một nền tảng trực tuyến hoàn chỉnh cho việc mua vé xem phim, kết nối mọi đối tượng liên quan, từ người dùng đến các rạp chiếu phim. Bằng cách này, hy vọng người dùng sẽ nâng cao được trải nghiệm, tăng cường hiệu quả vận hành và thúc đẩy sự phát triển của ngành công nghiệp giải trí.

Hệ thống có hai nhóm khách hàng chính là khách vãng lai, khách hàng đã có tài khoản. Khách hàng vãng lai sau khi đăng ký tài khoản có thể trở thành một khách hàng và được nhận các ưu đãi của hệ thống cũng như việc lưu lại các thông tin cho các lần mua vé.

Khách vãng lai là người dùng truy cập vào website là những khách hàng chưa có tài khoản trên hệ thống. Các khách hàng này có thể xem chi tiết các bộ phim mà họ muốn đặt, xem các bình luận cũng như việc tìm kiếm các bộ phim mà họ đang có nhu cầu. Để có thể có thêm các quyền hạn, khách vãng lai có thể tiến hành đăng ký tài khoản với website để có thể được lưu lại các thông tin cho các lần mua vé tiếp theo. Việc đăng ký cũng đơn giản và dễ dàng với nhiều tùy chọn giúp tối ưu thời gian đăng ký.

Nhóm khách hàng đã có tài khoản là nhóm khách hàng chính của hệ thống. Khách hàng có thể đăng nhập vào website bằng tên người dùng hoặc là email và mật khẩu đã đăng ký. Khách hàng đã có tài khoản có thể thực hiện toàn bộ chức năng của nhóm khách hàng vãng lai, đồng thời có thêm các chức năng khác như: được lưu thông tin tài khoản, có thể để lại bình luận tại một bộ phim, có thể tiến hành quá trình đặt mua vé, có thể thêm các bộ phim vào danh sách yêu thích, có thể thay đổi các thông tin cá nhân, ngoài ra khi đặt vé khách hàng có thể thanh toán trực tiếp trên hệ thống thông qua các ví điện tử Zalopay hoặc VNPay và nhận thông tin vé trực tiếp về email của mình.

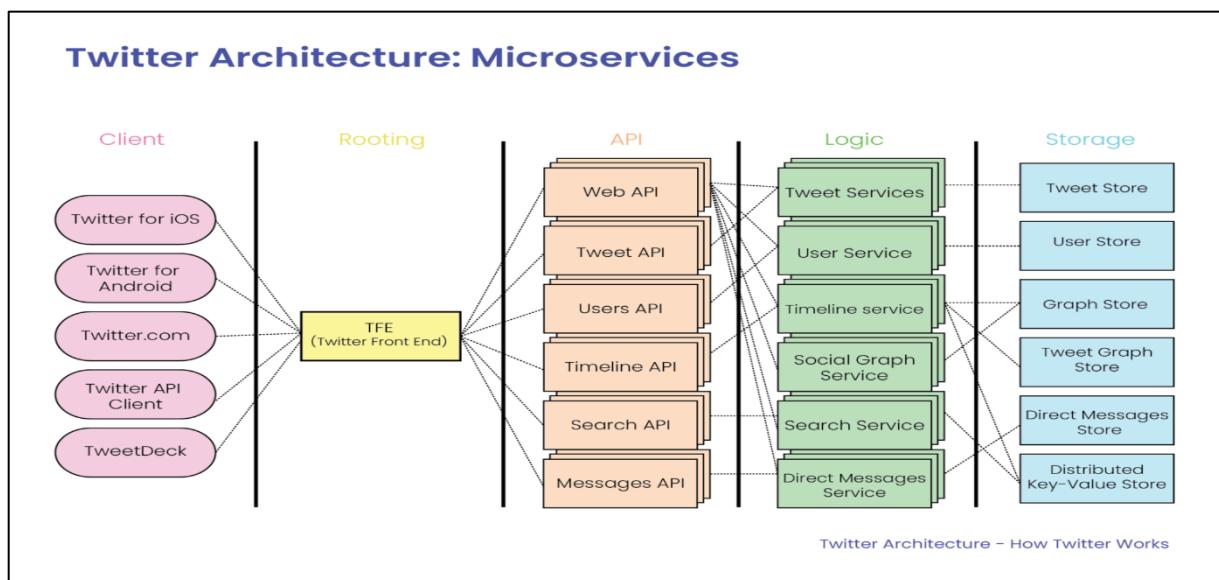
Ngoài hai nhóm khách hàng chính ở trên. Hệ thống hoạt động được quản lý bởi tài khoản Admin giúp cho việc thao tác và vận hành hệ thống dễ dàng hơn. Tài khoản Admin trong hệ thống có các vai trò như: thay đổi trạng thái của một bộ phim, thêm các tài nguyên cần thiết, thêm các rạp chiếu phim, xem trạng thái của các tài khoản, sắp lịch chiếu phim, xem các thống kê theo năm của hệ thống.

Website cũng cung cấp hệ thống thanh toán trực tuyến thông qua các ví điện tử phổ biến tại Việt Nam hiện nay là Zalopay và VNPay, hỗ trợ tối đa cho khách hàng trong việc mua sắm của họ. Cả hai ví điện tử ZaloPay và VNPay là hai trong số những giải pháp thanh toán điện tử phổ biến tại Việt Nam, cung cấp nhiều tiện ích cho người dùng và mở ra cơ hội lớn cho các nhà phát triển ứng dụng và dịch vụ.

Trong thời đại mà các công nghệ mới phát triển liên tục, nhiều công nghệ mới ra đời như Cloud Computing, Containerization (Docker, Kubernetes), DevOps. Cùng với sự phát triển của các công nghệ mới các ngôn ngữ lập trình mới cũng lần lượt ra đời như: Golang, Rust,... mang theo những thế mạnh của các ngôn ngữ lập trình cũ hơn làm thay đổi xu hướng lập trình của các nhà phát triển. Bên cạnh các công nghệ phần mềm thì phần các công ty phát triển phần cứng cũng đã làm rất tốt trong việc làm các linh kiện máy tính ngày càng rẻ, nhanh và mạnh mẽ hơn bao giờ hết. Xu hướng lưu trữ dữ liệu cũng có nhiều sự chuyển biến lớn từ cách tạo cơ sở dữ liệu truyền thống theo đối tượng, giờ đây xu hướng mới hướng tới việc lưu trữ các dữ liệu theo các sự kiện, các cơ sở dữ liệu mới như NoQuery, New Query dần trở thành xu hướng mới.

Vì vậy một kiến trúc microservices đã trở thành một xu hướng phổ biến trong thiết kế và triển khai các ứng dụng phần mềm hiện đại, đặc biệt là trong các môi trường đòi hỏi sự linh hoạt cao và khả năng mở rộng.

Hiện nay nhiều tập đoàn lớn đã sử dụng kiến trúc này để phát triển các sản phẩm của họ. Tiêu biểu có thể kể đến như: X (Twitter), Google, Amazon, Ebay,...



Hình 1-1: Kiến trúc Microservices của X(Twitter)

### **1.1.2. Các chức năng của hệ thống**

Hệ thống đặt vé xem phim trực tuyến sử dụng kiến trúc microservices và giao diện đa nền tảng được xây dựng nhằm mục đích giúp người dùng có thể tương tác với hệ thống một cách nhanh chóng, dễ dàng và thuận tiện. Toàn bộ hệ thống được xây dựng để đáp ứng các nhu cầu của ba nhóm đối tượng chính. Khách vãng lai, khách hàng đã có tài khoản và ADMIN.

#### **1.1.2.1. Chức năng của Khách vãng lai**

- Đăng ký.
- Đăng nhập.
- Tìm kiếm và lọc các bộ phim.
- Xem thông tin các bộ phim.

#### **1.1.2.2. Chức năng của Khách hàng đã có tài khoản**

- Thêm một bộ phim vào danh sách yêu thích.
- Đặt vé xem phim thời gian thực.
- Bình luận một bộ phim
- Quản lý thông tin tài khoản.
- Quản lý các vé xem phim đã đặt.
- Thanh toán.
- Đăng xuất.

#### **1.1.2.3. Chức năng của ADMIN**

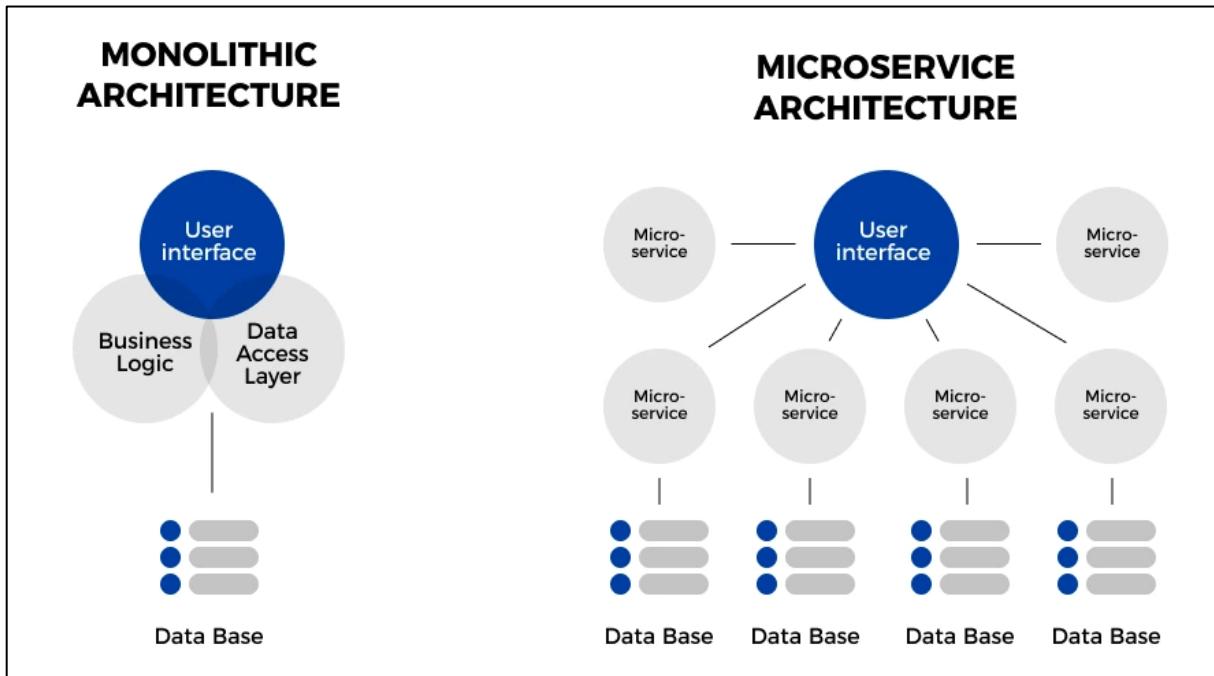
- Thống kê doanh thu, người dùng,...
- Quản lý các bộ phim.
- Quản lý các hóa đơn.
- Quản lý các rạp chiếu.
- Quản lý các thể loại của phim.
- Quản lý lịch chiếu phim.
- Quản lý người dùng.

## **1.2. Phân tích đánh giá các giải pháp có liên quan đến bài toán**

### **1.2.1. Phân tích các giải pháp cho phần back-end của hệ thống**

Khi phát triển một hệ thống có dạng như các hệ thống liên quan đến việc đặt vé trực tuyến, có hai kiến trúc phổ biến có thể được cân nhắc. Kiến trúc nguyên khối (Monolithic Architecture) và Kiến trúc microservices (Microservices Architecture). Tuy

vậy mỗi kiến trúc đều có những ưu điểm và nhược điểm riêng, ảnh hưởng đến hiệu quả hoạt động, bảo trì và khả năng mở rộng của hệ thống. Vì vậy lựa chọn một hệ thống phù hợp cần cân nhắc rất nhiều yếu tố khác nhau.



Hình 1-2: Kiến trúc Monolithic và kiến trúc microservices

#### 1.2.1.1. Kiến trúc nguyên khối (Monolithic Architecture) [11]

Kiến trúc nguyên khối là mô hình truyền thống, trong đó toàn bộ ứng dụng được đóng gói thành một khối duy nhất. Các thành phần như xử lý giao diện người dùng (UI), xử lý nghiệp vụ (business logic), và quản lý cơ sở dữ liệu (database management) đều được tích hợp trong một khối code thống nhất.

##### Ưu điểm:

- **Đơn giản trong phát triển và triển khai:** Với kiến trúc nguyên khối, tất cả các thành phần của hệ thống được quản lý trong một dự án duy nhất. Điều này giúp giảm tối thiểu độ phức tạp trong quá trình phát triển, triển khai và kiểm thử.
- **Hiệu suất cao trong quy mô nhỏ:** Khi quy mô hệ thống còn nhỏ hoặc đối với các dự án nhỏ, kiến trúc nguyên khối tỏ ra độ hiệu quả và hiệu suất cao, hoạt động cực kỳ hiệu quả vì không có nhiều lớp kết nối và trao đổi dữ liệu giữa các dịch vụ.

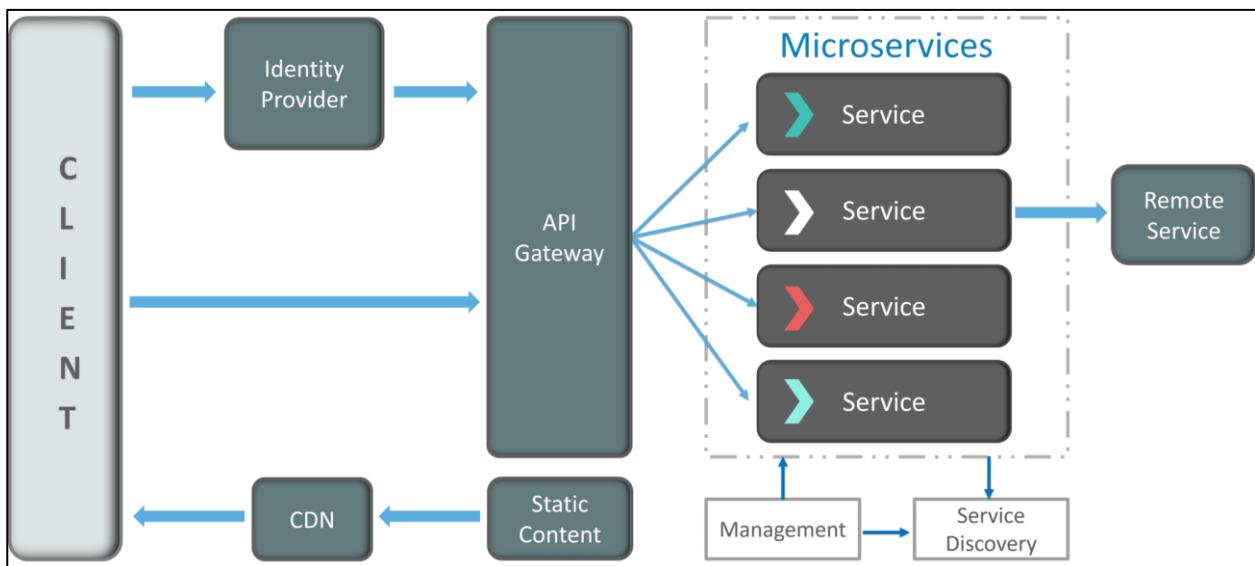
- **Độ nhất quán dữ liệu duy trì:** Dễ dàng quản lý tính nhất quán dữ liệu do toàn bộ logic nằm trong cùng một khái niệm, giúp các tác vụ liên quan đến giao dịch (transactional tasks) trở nên đơn giản hơn.

### Nhược điểm:

- **Khó bảo trì và mở rộng:** Khi hệ thống mở rộng với nhiều chức năng phức tạp, codebase sẽ trở nên lớn, phức tạp và khó bảo trì. Việc thêm chức năng mới có thể ảnh hưởng đến toàn bộ hệ thống.
- **Khả năng chịu tải kém khi quy mô tăng:** Với một lượng người dùng lớn, kiến trúc nguyên khái có thể gặp vấn đề về hiệu suất vì toàn bộ hệ thống phải chịu tải đồng đều mà không thể phân tách theo từ dịch vụ riêng biệt.
- **Độ linh hoạt thấp:** Việc cập nhật, nâng cấp hay thay đổi một phần của ứng dụng đòi hỏi phải triển khai lại toàn bộ hệ thống, làm tăng nguy cơ lỗi, thời gian bảo trì cũng như thời gian gián đoạn của ứng dụng.

#### 1.2.1.2. Kiến trúc microservices (Microservices Architecture) [3]/[11]

Kiến trúc microservices là một kiểu kiến trúc phần mềm phân tán trong đó ứng dụng được phát triển như là một tập hợp các dịch vụ nhỏ, độc lập nhau và có khả năng hoạt động tự lập. Mỗi dịch vụ này thực hiện một chức năng cụ thể của ứng dụng và có thể được phát triển, triển khai và quản lý riêng biệt.



Hình 1-3: Kiến trúc microservices

### Ưu điểm:

- **Khả năng mở rộng linh hoạt:** Với kiến trúc microservices, mỗi dịch vụ có thể được triển khai độc lập và mở rộng riêng biệt. Điều này rất quan trọng trong trường hợp hệ thống cần xử lý lượng người dùng tăng đột biến trong những dịp cao điểm.
- **Dễ dàng bảo trì và nâng cấp:** Các dịch vụ nhỏ hơn, độc lập giúp dễ dàng phát triển, bảo trì, và kiểm thử. Nếu có lỗi ở một dịch vụ nào đó, việc sửa chữa chỉ ảnh hưởng đến dịch vụ đó mà không làm gián đoạn toàn bộ hệ thống.
- **Khả năng tích hợp công nghệ đa dạng:** Mỗi dịch vụ có thể được xây dựng bằng các công nghệ phù hợp nhất cho từng phần việc, như Spring Boot cho xử lý nghiệp vụ và WebSocket cho thông báo thời gian thực.

### Nhược điểm:

- **Độ phức tạp trong triển khai và quản lý:** Kiến trúc microservices yêu cầu nhiều kỹ thuật để quản lý và giám sát các dịch vụ, đòi hỏi có hệ thống quản lý kết nối và bảo mật phức tạp hơn.
- **Chi phí cao về tài nguyên:** Mỗi dịch vụ hoạt động như một ứng dụng độc lập, nên tiêu tốn nhiều tài nguyên hệ thống hơn. Việc theo dõi và đồng bộ dữ liệu cũng trở nên phức tạp, đặc biệt là trong các giao dịch liên dịch vụ (cross-service transactions).
- **Khả năng thất bại dây chuyền:** Một dịch vụ gặp lỗi có thể ảnh hưởng đến các dịch vụ khác, đòi hỏi hệ thống phải có các cơ chế giám sát và khôi phục để giảm thiểu tác động.

#### 1.2.2. Phân tích các giải pháp phần giao diện của hệ thống

Về phần giao diện người dùng việc lựa chọn công nghệ sẽ có ảnh hưởng lớn và trực tiếp đến trải nghiệm người dùng, hiệu suất và khả năng bảo trì của ứng dụng. Thông thường ta vẫn có hai cách tiếp cận chính được xem xét: **phát triển giao diện truyền thống** và **phát triển giao diện với các framework hiện đại** như Next.js và Angular.

##### 1.2.2.1. Phát triển giao diện truyền thống

Phát triển giao diện truyền thống chủ yếu sử dụng HTML, CSS, và JavaScript thuần (JavaScript Vanilla). Trong cách tiếp cận này, phần lớn xử lý giao diện và logic diễn ra trên phía client (trình duyệt) hoặc dựa vào các template từ phía server từ mô hình MVC.

### Ưu điểm:

- **Dễ dàng triển khai và học hỏi:** HTML, CSS, và JavaScript là các công nghệ cơ bản trong phát triển web, dễ tiếp cận và không yêu cầu nhiều kiến thức về các framework.
- **Kiểm soát trực tiếp và chi phí thấp:** Cách tiếp cận này thường không đòi hỏi sử dụng các framework phức tạp nên có thể giảm chi phí phát triển. Nhà phát triển có toàn quyền kiểm soát cấu trúc và tổ chức mã nguồn.
- **Hiệu suất ổn định trên quy mô nhỏ:** Trong các dự án đơn giản và ít phức tạp, giao diện truyền thống có thể hoạt động tốt mà không cần đến sự phức tạp của các framework.

### Nhược điểm:

- **Khả năng tái sử dụng mã nguồn thấp:** JavaScript và HTML truyền thống không có cơ chế module hóa hiệu quả, dẫn đến mã nguồn khó tái sử dụng và dễ bị lặp lại.
- **Thiếu tính năng hỗ trợ hiện đại:** Không có công cụ tích hợp sẵn để quản lý trạng thái, routing, hoặc tối ưu hóa hiệu suất trên trình duyệt, điều này ảnh hưởng đến trải nghiệm người dùng khi giao diện trở nên phức tạp.
- **Bảo trì và mở rộng khó khăn:** Khi hệ thống lớn lên, việc kiểm soát mã nguồn sẽ phức tạp, làm tăng chi phí bảo trì và khả năng phát sinh lỗi.

#### 1.2.2.2. Phát triển giao diện với các framework (Next.js và Angular) [6] [7]

Next.js và Angular là các framework hiện đại, mang lại nhiều tính năng và công cụ hỗ trợ phát triển giao diện người dùng hiệu quả, đồng thời cho phép tối ưu hóa trải nghiệm người dùng trên nhiều thiết bị. Do bản thân là framework nên có nhiều công cụ hữu ích hỗ trợ cho việc phát triển từ đó giúp các nhà phát triển có thể tập trung vào viết các logic mà không cần bận tâm nhiều về việc chuẩn bị.



Hình 1-4: Next.js và Angular

### Ưu điểm:

- **Quản lý trạng thái và routing hiệu quả:** Next.js và Angular cung cấp các công cụ quản lý trạng thái (state management) và routing (định tuyến) giúp tối ưu hóa quá trình di chuyển và xử lý dữ liệu giữa các trang mà không cần phải tải lại toàn bộ.
- **Tối ưu hóa hiệu suất và SEO:** Next.js, với khả năng render phía server (Server-Side Rendering - SSR) và khả năng render trước khi tải (Static Site Generation - SSG), giúp cải thiện thời gian tải trang và tối ưu hóa SEO. Điều này đặc biệt hữu ích cho hệ thống đặt vé xem phim, khi người dùng cần tải nhanh thông tin về lịch chiếu và chi tiết phim.
- **Hỗ trợ phát triển giao diện đa nền tảng:** Angular và Next.js đều hỗ trợ responsive design, cho phép ứng dụng hoạt động mượt mà trên nhiều loại thiết bị, từ desktop đến điện thoại.
- **Dễ bảo trì và mở rộng:** Với cách tổ chức mã nguồn theo component (thành phần), mã nguồn trở nên dễ bảo trì và mở rộng hơn. Các thành phần có thể được tái sử dụng và kiểm thử độc lập.

### Nhược điểm:

- **Đòi hỏi kiến thức chuyên sâu hơn:** Việc sử dụng Next.js hay Angular đòi hỏi kiến thức về framework và cần nhiều thời gian để học hỏi. Các framework này phức tạp hơn phát triển truyền thống, đặc biệt với các tính năng như SSR (render phía server) hay quản lý trạng thái.
- **Chi phí tài nguyên cao hơn:** Các framework này có thể yêu cầu nhiều tài nguyên hệ thống hơn và yêu cầu setup phức tạp hơn cho quá trình triển khai, giám sát và bảo trì.
- **Tính phụ thuộc vào framework:** Khi sử dụng Next.js hoặc Angular, ứng dụng sẽ phụ thuộc vào các cập nhật của framework, có thể cần điều chỉnh lại mã nguồn khi có phiên bản mới.

### 1.3. Tiếp cận giải quyết vấn đề, chọn lựa giải pháp

Với các yêu cầu, mục tiêu được mô tả chi tiết và các phân tích ở trên, có thể thấy rằng **hệ thống đặt vé xem phim trực tuyến** yêu cầu một giải pháp có khả năng mở rộng cao, dễ bảo trì và đảm bảo hiệu suất trong điều kiện lượng người dùng tăng đột biến. Điều này nhằm đáp ứng nhu cầu truy cập và thao tác đa dạng từ phía người dùng (chọn phim, chọn ghế, thanh toán) một cách ổn định và liền mạch. Vì vậy có thể thấy rằng, kiến trúc microservices và các framework xây dựng UI hiện đại như Next.js và Angular là lựa chọn phù hợp để làm cơ sở phát triển hệ thống.

#### 1.3.1. Lựa chọn kiến trúc cho hệ thống back-end

Kiến trúc microservices được lựa chọn cho hệ thống back-end nhờ các đặc điểm phù hợp với yêu cầu của hệ thống đặt vé xem phim trực tuyến.

- **Khả năng phát triển độc lập:** Microservices cho phép chúng ta bóc tách hệ thống lớn thành các thành phần nhỏ hơn để dễ dàng cài đặt, phát triển và triển khai. Điều này giúp hệ thống có thể tăng khả năng chịu tải ở những thành phần có nhu cầu cao hơn bằng việc cung cấp thêm các service có khả năng xử lý tương tự mà không ảnh hưởng đến hiệu suất toàn bộ hệ thống.
- **Độ tin cậy cao và dễ bảo trì:** Vì các dịch vụ được phân tách, nếu có sự cố ở một dịch vụ cụ thể, các dịch vụ khác vẫn có thể hoạt động bình thường. Điều này tăng độ tin cậy và giảm thiểu thời gian gián đoạn. Ngoài ra, khi cần sửa lỗi hoặc nâng cấp, chỉ cần tập trung vào dịch vụ cần thiết mà không làm gián đoạn các dịch vụ khác.
- **Khả năng tích hợp công nghệ đa dạng:** Với kiến trúc microservices, các công nghệ khác nhau có thể được sử dụng cho các phần khác nhau của hệ

thống (ví dụ: Spring Boot cho logic nghiệp vụ, WebSocket cho chọn ghế thời gian thực), tạo nên một hệ thống tối ưu cả về hiệu suất và tính năng.

### 1.3.2. *Lựa chọn giải pháp cho phần giao diện hệ thống*

Giao diện là bộ mặt của một ứng dụng, với các yêu cầu ngày càng khắt khe về mặt giao diện cũng như các trải nghiệm khi sử dụng một trang web đòi hỏi các nhà phát triển cần các giải pháp đáp ứng được độ phức tạp khi thiết kế các giao diện có tính tương tác cao, phải thân thiện và phải đi kèm với một hiệu suất cao và ổn định. Vì vậy các framework hiện đại như Next.js và Angular là các sự lựa chọn phù hợp nhờ khả năng đáp ứng các yếu tố trên:

- **Hiệu suất và tối ưu hóa SEO:** Next.js cung cấp khả năng render phía server (SSR) và render trước (SSG), giúp cải thiện thời gian tải trang và tối ưu hóa SEO. Điều này đặc biệt quan trọng với một hệ thống đặt vé xem phim, nơi người dùng cần truy cập nhanh vào thông tin phim và lịch chiếu.
- **Tính năng quản lý trạng thái và routing hiệu quả:** Angular và Next.js hỗ trợ quản lý trạng thái (state management) và định tuyến (routing), giúp người dùng dễ dàng chuyển đổi giữa các trang mà không cần tải lại toàn bộ, tăng tốc độ và trải nghiệm mượt mà khi thao tác trên hệ thống.
- **Tích hợp đa nền tảng:** Các framework này hỗ trợ giao diện responsive một cách đơn giản nhờ tích hợp dễ dàng các thư viện hỗ trợ như tailwindcss, giúp hệ thống có khả năng hiển thị tốt trên nhiều loại thiết bị, từ máy tính đến điện thoại di động, đảm bảo trải nghiệm người dùng nhất quán trên các nền tảng.

## Chương 2. THIẾT KẾ VÀ CÀI ĐẶT GIẢI PHÁP

### 2.1. Thiết kế kiến trúc

#### 2.1.1. Các vấn đề thiết kế kiến trúc Microservices [11]/[14]

Được tin tưởng và sử dụng bởi nhiều ông lớn trên thị trường công nghệ như: X(Twitter), Amazon hay Ebay. Kiến trúc microservices trở thành một Design Pattern phổ biến trong việc phát triển các ứng dụng theo xu hướng linh hoạt như hiện nay.

Một Service thực hiện một tác vụ riêng biệt và tách biệt với các service còn lại, lấy ví dụ: quản lý các bộ phim, quản lý đặt ghế,... Mỗi Microservice là một ứng dụng nhỏ có đủ cấu trúc của một ứng dụng Spring Boot bao gồm các controllers, các business logic và một cơ sở dữ liệu riêng. Các Service này còn có thêm các cài đặt để có thể giao tiếp với các service khác thông qua API hay hệ thống Message Queue.

Vấn đề tách biệt các service là một công việc quan trọng khi muốn triển khai một hệ thống theo kiến trúc Microservices. Cần cân nhắc về tính độc lập không ảnh hưởng lẫn nhau khi hoạt động, tính đảm bảo một tính năng cần có của một service. Mỗi service phải đảm bảo hoàn thành một tính năng mà nó được cài đặt, chẳng hạn như tính năng xác thực, tính năng thanh toán, v.v... Do đó việc phân tích dự án và phân chia ra các microservice phù hợp là một công việc rất khó khăn.

Do việc các service hoạt động độc lập với nhau nên nó cũng sẽ có các cơ sở dữ liệu riêng biệt. Điều này rất quan trọng, chúng ta cần đảm bảo nếu các service cần giao tiếp với nhau thì có đủ dữ liệu để có thể trao đổi mà không phải lưu trữ quá nhiều dữ liệu dư thừa. Các dữ liệu trên các service phải được đồng bộ với bất cứ thay đổi nào nếu chúng có mối liên hệ với nhau nhưng lại ở các service khác nhau để đảm bảo toàn hệ thống có thể hoạt động một cách chính xác nhất.

#### 2.1.2. Giải pháp kiến trúc

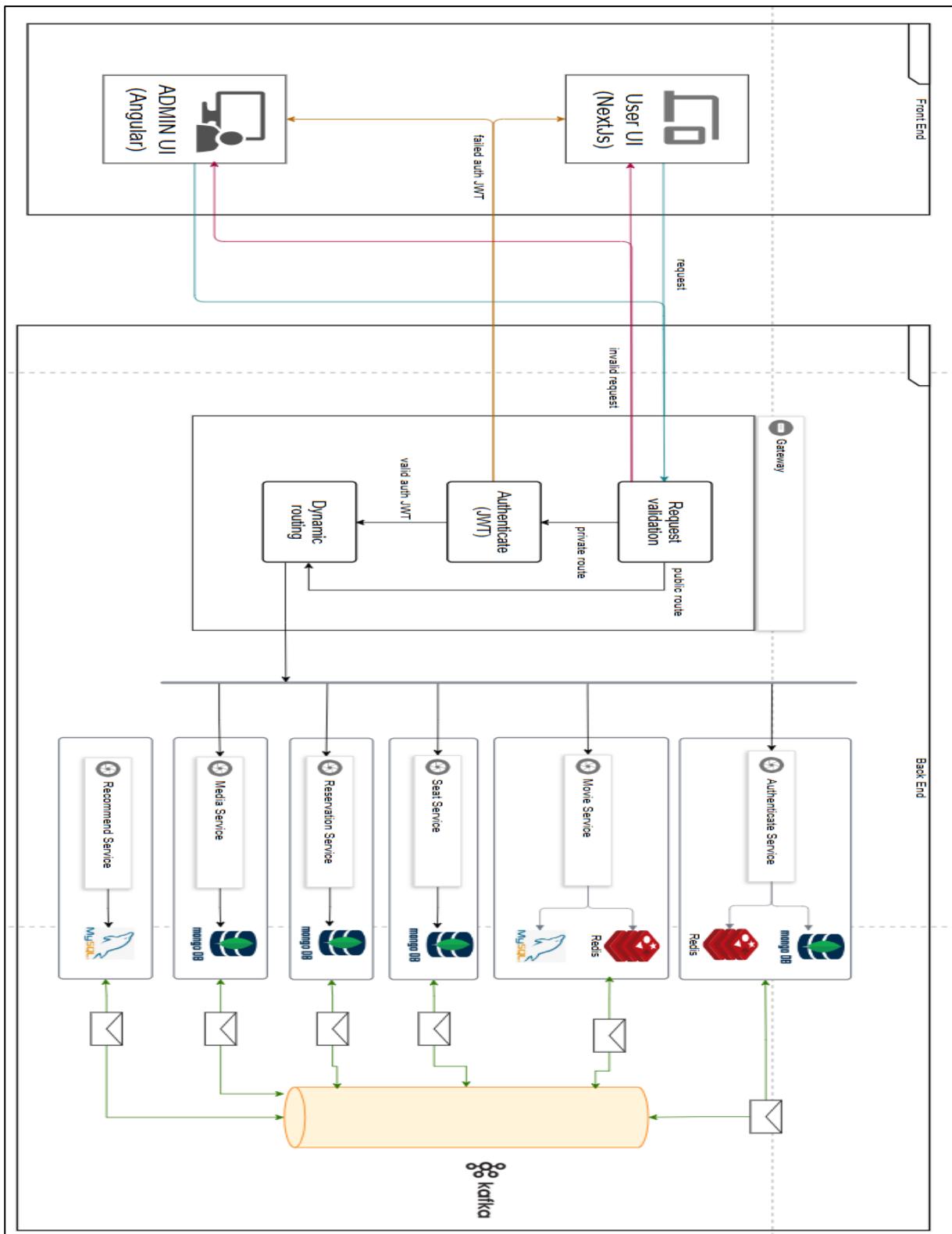
Phát triển hệ thống đặt vé xem phim trực tuyến với kiến trúc Microservices sẽ sử dụng Spring Boot và hệ quản trị cơ sở dữ liệu MongoDB và MySql để lưu trữ và truy xuất dữ liệu.

Các microservices trong hệ thống đã được phân chia để có thể hoạt động tách biệt và độc lập. Đồng thời cơ sở dữ liệu cũng được thiết kế để có thể hoạt động một cách đồng bộ với nhau. Các services trong hệ thống bao gồm:

- Registry service: chịu trách nhiệm quản lý các service khác.

- Gateways-service: nơi xử lý các request đến, xác thực bảo mật, phân bổ các request đến các service phù hợp để xử lý chúng, nhận kết quả từ các service sau đó trả về cho người dùng. Tất cả các request đến đều phải thông qua gateway-service.
- Movie-service: là service chính xử lý các tác vụ liên quan đến việc xử lý đối tượng movie.
- Auth-service: là service xử lý các tác vụ liên quan đến việc xác thực người dùng và điều chỉnh các thông tin của người dùng người dùng.
- Seat-service: là service xử lý các tác vụ liên quan đến việc chọn ghế, quản lý ghế của các rạp chiếu.
- Reservation-service: là service xử lý các công việc liên quan đến việc đặt vé, quản lý vé đã đặt và thanh toán,...
- Media-service: là service chuyên xử lý các công việc liên quan đến các tài nguyên như hình ảnh và video.
- Recommend-service: là service xử lý các công việc liên quan đến gợi ý các từ khóa tìm kiếm và các bộ phim tương tự thông qua các giải thuật phân tích ngôn ngữ.

## Hệ thống bán vé xem phim trực tuyến



Hình 2-1: Sơ đồ tổng quan dự án

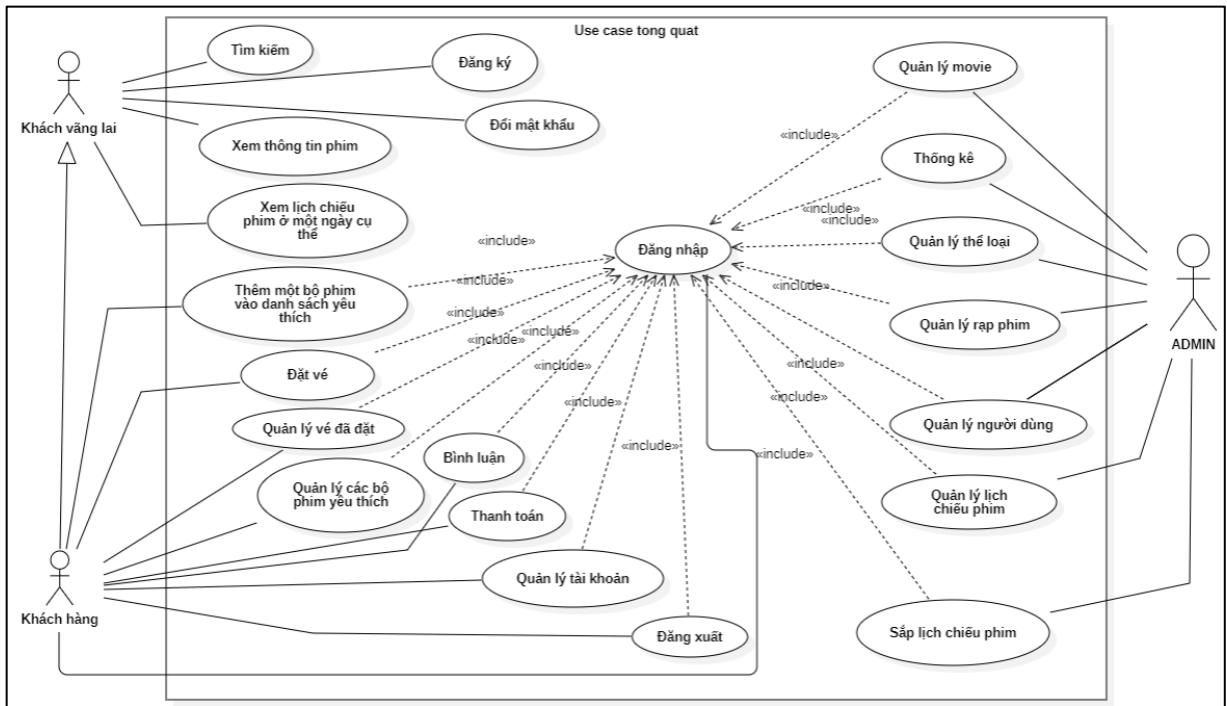
Luồng hoạt động của dự án như sau:

- **Gửi request:** khi người dùng tương tác với website, client sẽ tiến hành gửi các request đến cho server thông qua gateway.
- **Nhận request:** khi một request được gửi từ client đến server, server sẽ nhận định và xác định loại request và endpoint cụ thể mà request đó đích đến. Công việc này sẽ do API Gateway đảm nhận.
- **Routing và xác thực:**
  - o Đầu tiên API gateway sẽ xác định request đến có hợp lệ không, tức là các microservices có thể xử lý request đó không. Trường hợp là một request lạ thì server từ chối xử lý, ngược lại thì tiếp tục bước tiếp theo.
  - o Xác thực request, nếu request thuộc dạng public thì bỏ qua bước này, ngược là các request thuộc private route phải tiến hành xác thực trước khi tới bước tiếp theo. Sử dụng JWT để xác thực thông tin của request. Xác thực bao gồm thời hạn của JWT và khóa của JWT. Nếu request được xác thực không thành công, server từ chối xử lý request, ngược lại tới bước tiếp theo.
  - o Sau khi qua bước xác thực, gateway gửi request đến dịch vụ microservices thích hợp, request có thể chứa các tham số và dữ liệu cần thiết.
- **Xử lý request:** Một microservice sẽ nhận request và thực hiện xử lý tương ứng với chức năng của nó. Điều này có thể bao gồm truy xuất cơ sở dữ liệu, xử lý business logic, tính toán, hoặc giao tiếp với các microservices khác nếu cần.
- **Trả về kết quả:** Sau khi xử lý request, microservice sẽ trả kết quả về lại cho gateway thông qua response. Kết quả này sẽ là dữ liệu yêu cầu hoặc là báo lỗi nếu có.
- **Tổng hợp và response cho client:** Gateway nhận lại kết quả từ dịch vụ microservices và tổng hợp thành response cuối cùng để gửi về client.
- **Cập nhật kết quả:** Client nhận kết quả từ server rồi cập nhật lại giao diện dựa trên kết quả hoặc hiển thị lỗi nếu có.

## 2.2. Sơ đồ Use case của hệ thống

Hệ thống được chia thành ba nhóm người dùng chính: khách vãng lai, khách hàng đã có tài khoản và vai trò ADMIN.

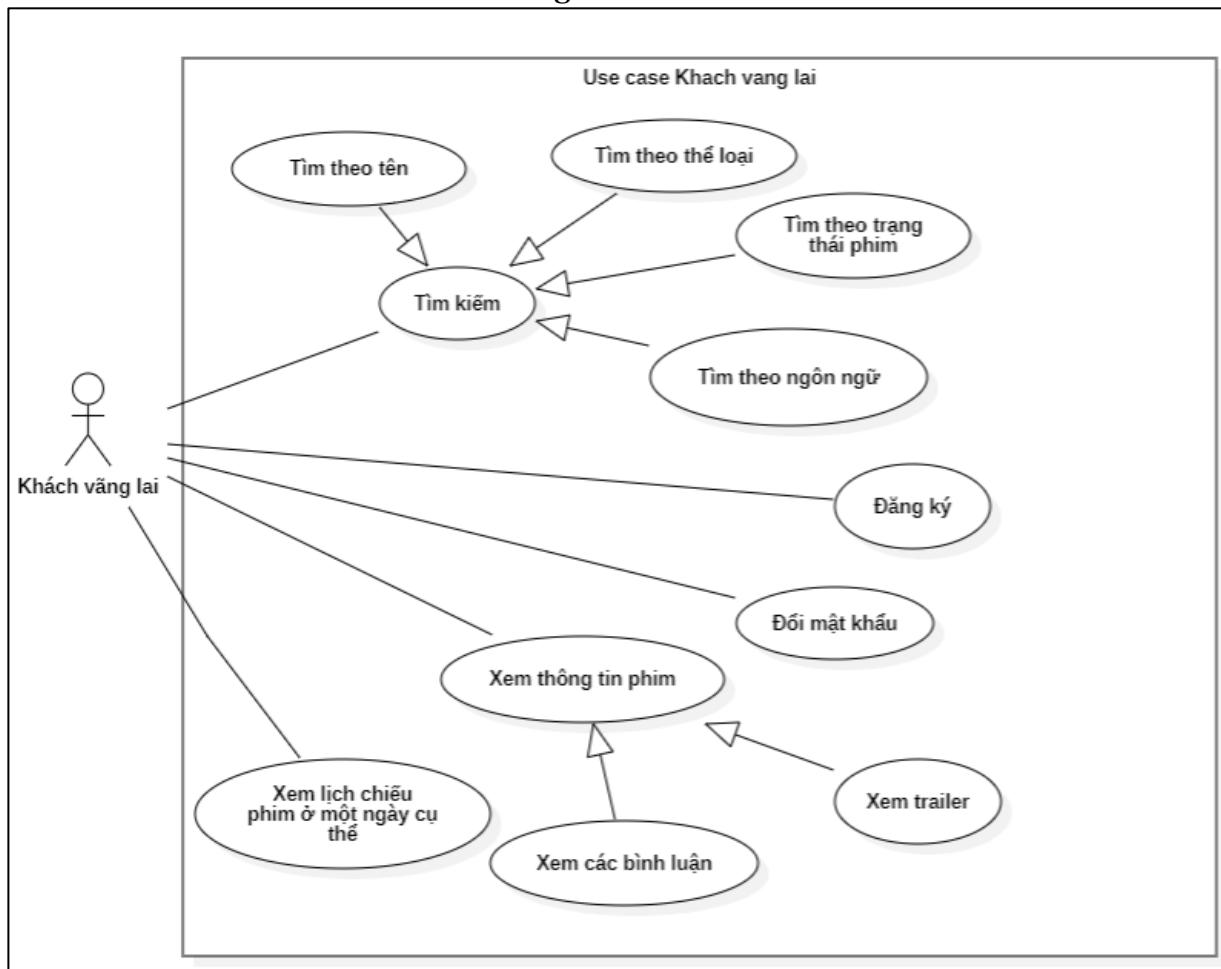
### 2.2.1. Sơ đồ Use Case tổng quát của hệ thống



Hình 2-2: Sơ đồ Use Case tổng quát

Sơ đồ Use case tổng quát ở mức 0 của hệ thống cung cấp một cái nhìn tổng quát về chức năng của từng nhóm người dùng trong hệ thống. Hệ thống cung cấp các chức năng chính như: tìm kiếm phim, đặt vé, quản lý vé đã đặt, xem thông tin chi tiết và quản lý các tài nguyên của hệ thống (Admin). Có ba nhóm người dùng chính bao gồm: Khách vãng lai, Khách hàng và Admin.

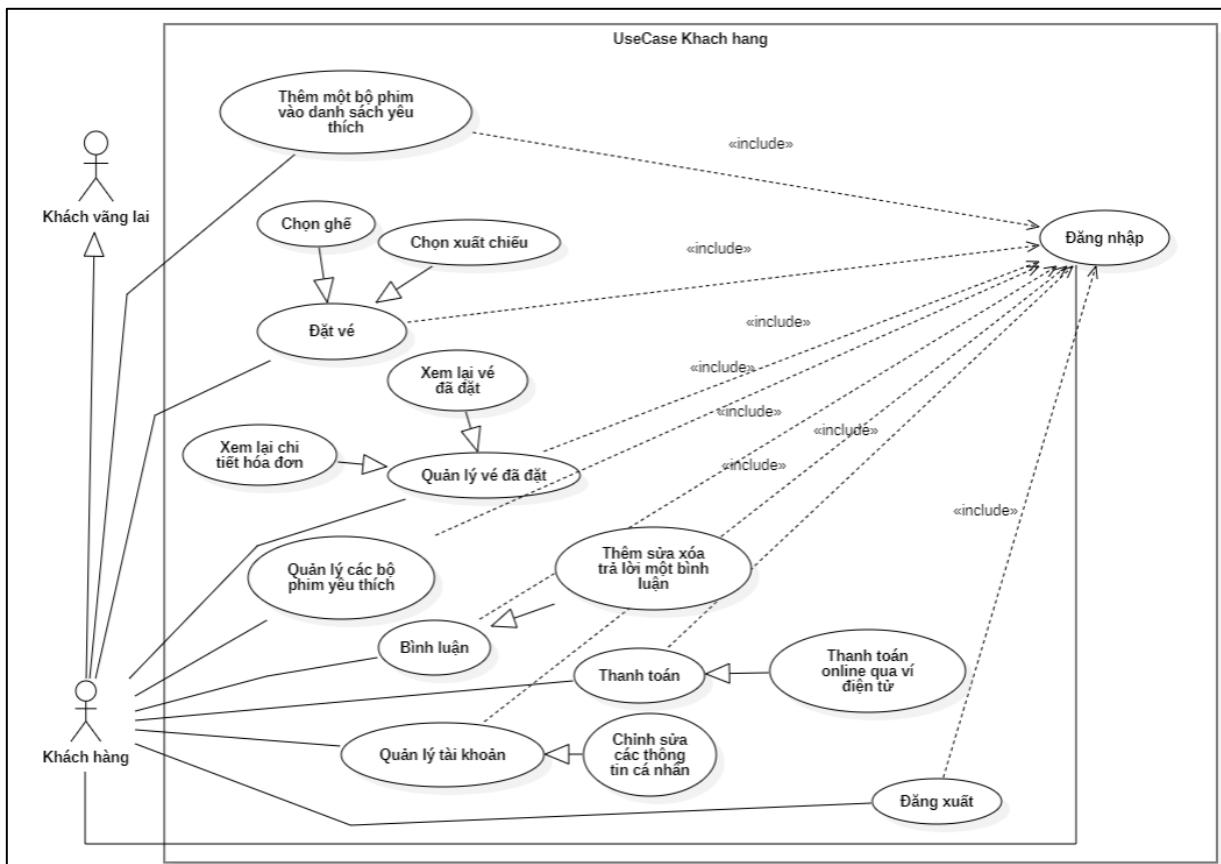
### 2.2.2. Sơ đồ Use Case của khách vãng lai



Hình 2-3: Sơ đồ Use case của khách vãng lai

Sơ đồ cụ thể Use case của Khách vãng lai cung cấp các chức năng chi tiết một người dùng chưa có tài khoản có thể tương tác với hệ thống bao gồm: Tìm kiếm các bộ phim, đăng ký, đăng nhập, xem chi tiết một bộ phim và xem lịch chiếu phim hàng ngày.

### 2.2.3. Sơ đồ Use case của khách hàng đã có tài khoản

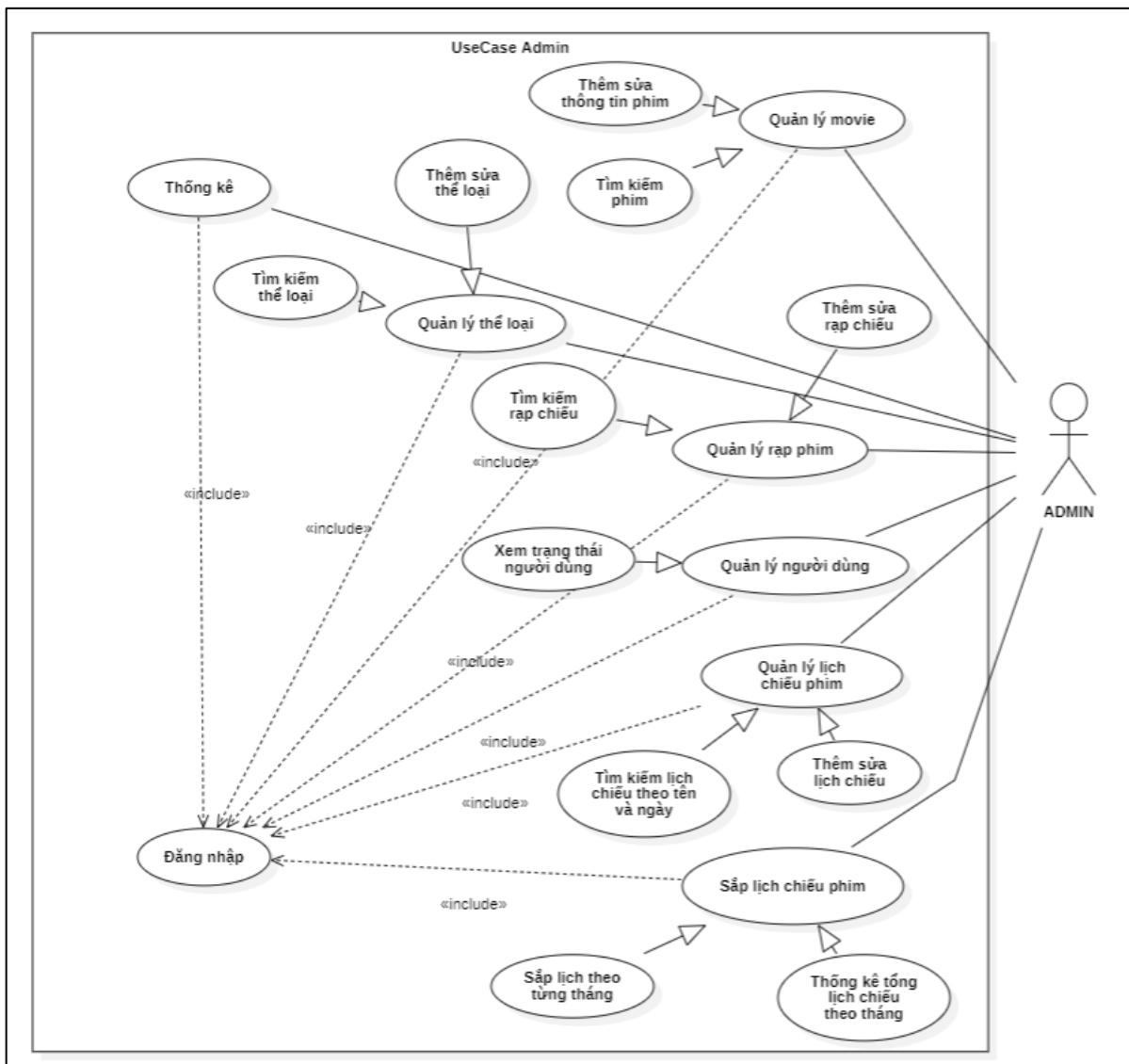


Hình 2-4: Sơ đồ Use case của khách hàng đã có tài khoản

Sơ đồ Use case của Khách hàng cho thấy thêm được nhiều chức năng quan trọng mà người dùng khi đã đăng ký tài khoản với hệ thống có thể sử dụng bao gồm các chức năng như: Đặt vé, chọn ghế thời gian thực, thanh toán online, quản lý tài khoản, thêm mới bình luận,...

#### 2.2.4. Sơ đồ Use Case của ADMIN

Sơ đồ Use case Admin để biểu diễn các chức năng quản lý các tài nguyên quan trọng của hệ thống. Admin là người dùng có quyền để quản lý các tài nguyên như các bộ phim, thể loại phim, các suất chiếu, lịch chiếu,... Đây là vai trò quan trọng vì nó ảnh hưởng rất nhiều đến hai vai trò đã nêu ở trên.

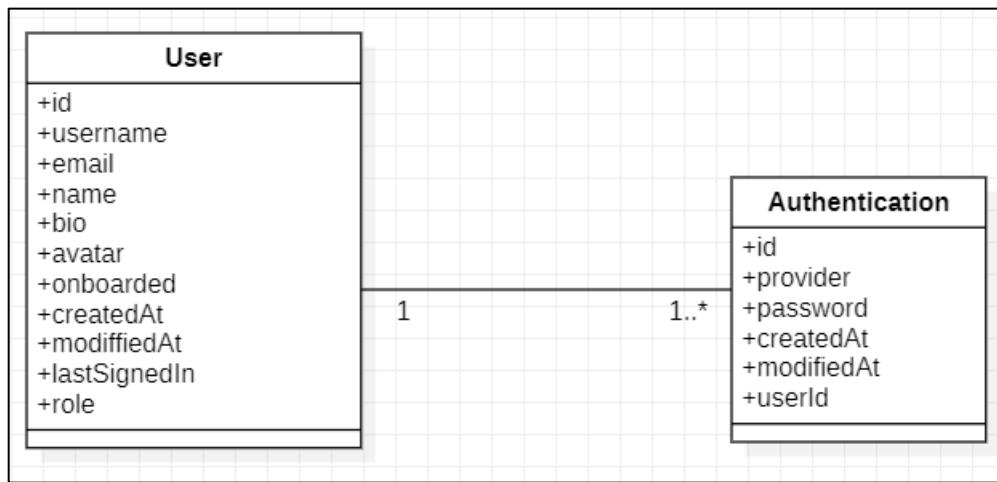


Hình 2-5: Sơ đồ Use case ADMIN

### 2.3. Thiết kế cơ sở dữ liệu

Do hệ thống được chia thành nhiều Service khác nhau nên mỗi Service phải được thiết kế một cơ sở dữ liệu riêng biệt để có thể hoạt động độc lập, hạn chế phụ thuộc vào dữ liệu của Service khác nhất có thể tuy vậy vẫn đảm bảo tính đồng nhất dữ liệu giữa các services. Đảm bảo tính độc lập của kiến trúc Microservices.

### 2.3.1. Cơ sở dữ liệu của Auth Service



Hình 2-6: Cơ sở dữ liệu của Auth Service

Auth Service là service phụ trách các tác vụ liên quan đến người dùng. Các công việc chính của service này là xác thực thông tin của người dùng. Tạo token giúp xác thực cho mỗi request của người dùng. Ngoài ra các tác vụ liên quan đến việc xác thực người dùng,... điều được xử lý tại đây.

```

_id: ObjectId('66e287375ec9df386f0abee7')
username : "pitithuong"
email : "pitithuong@gmail.com"
name : "Thường Phan Thành"
bio : "Let's leave the world behind, this is new bio I just added."
avatar : "https://lh3.googleusercontent.com/a/ACg8ocK08AoizT-5cAfgZRQE-vhrHDbbn... "
onboarded : true
createdAt : 2024-09-12T06:16:23.515+00:00
modifiedAt : 2024-10-25T05:06:07.490+00:00
lastSignedIn : 2024-10-25T09:41:51.803+00:00
authentications : Array (2)
  0: Object
    provider : "local"
    password : "$2a$10$jMEjqttjQnqp2E01Ea9aheiVSwdfessR16orDX9HL2vez4r72X.da"
    createdAt : 2024-09-12T06:16:23.515+00:00
    modifiedAt : 2024-09-12T06:16:23.515+00:00
  1: Object
    provider : "google"
    createdAt : 2024-09-14T05:27:23.302+00:00
    modifiedAt : 2024-09-14T05:27:23.302+00:00
role : "ADMIN"
_class : "com.moonmovie.auth_service.models.User"
  
```

Hình 2-7: Một document của Auth service dưới dạng đối tượng

### 2.3.1.1. Thực thể user

Người dùng (User): Đây là thực thể để lưu lại các thông tin của một người dùng có tài khoản trên hệ thống. Mỗi đơn vị thực thể User lưu trữ một thông tin của một người dùng cụ thể với các trường thông tin như: username, email, name, role,...

Bảng 2-1: Thực thể User

| Tên thuộc tính | Điễn giải                          | Kiểu dữ liệu | Khóa chính | Khóa ngoại | Ràng buộc     |
|----------------|------------------------------------|--------------|------------|------------|---------------|
| <b>id</b>      | Mã người dùng                      | varchar(24)  | X          |            | Not null      |
| username       | Tên đăng nhập                      | varchar(30)  |            |            | Not null      |
| email          | Email người dùng                   | varchar(150) |            |            | Not null      |
| name           | Tên người dùng                     | varchar(60)  |            |            |               |
| bio            | Bio người dùng                     | Text         |            |            | <= 1000 ký tự |
| avatar         | Ảnh đại diện                       | varchar(255) |            |            |               |
| onboarded      | Người dùng đăng nhập lần đầu       | boolean      |            |            |               |
| createdAt      | Ngày tạo tài khoản                 | datetime     |            |            |               |
| modifiedAt     | Ngày chỉnh sửa thông tin tài khoản | datetime     |            |            |               |
| lastSigned     | Lần cuối đăng nhập                 | datetime     |            |            |               |
| role           | Quyền của người dùng               | varchar(10)  |            |            | Not null      |

### 2.3.1.2. Thực thể Authentication

Xác thực (Authentication): Đây là thực thể để lưu lại các thông tin về phương thức đăng nhập của một người dùng trên hệ thống. Mỗi người dùng có thể đăng nhập theo cách sử dụng tài khoản đã đăng ký hoặc đăng nhập với Google và các phương thức khác.

Bảng 2-2: Thực thể Authentication

| Tên thuộc tính | Điễn giải                 | Kiểu dữ liệu | Khóa chính | Khóa ngoại | Ràng buộc |
|----------------|---------------------------|--------------|------------|------------|-----------|
| <b>id</b>      | Id phương thức đăng nhập  | varchar(24)  | X          |            | Not null  |
| <b>userId</b>  | Id người dùng             | varchar(24)  |            | X          | Not null  |
| provider       | Tên phương thức đăng nhập | varchar(6)   |            |            | Not null  |

| Tên thuộc tính | Diễn giải   | Kiểu dữ liệu | Khóa chính | Khóa ngoại | Ràng buộc |
|----------------|---|--------------|------------|------------|-----------|
| password       | Mật khẩu được mã hóa hoặc trống nếu đăng nhập bằng các phương thức khác như google. | varchar(30)  |            |            |           |
| createdAt      | Ngày tạo  | datetime     |            |            |           |
| modifiedAt     | Ngày chỉnh sửa  | datetime     |            |            |           |

### 2.3.1.3. Các ràng buộc toàn vẹn dữ liệu

#### 2.3.1.3.1. Bảng User

##### a. Ràng buộc liên thuộc tính

**Tên ràng buộc:** CHK\_User\_Email\_Format

**Mô tả:** Email của người dùng (email) phải có định dạng hợp lệ.

**Nội dung:**

$$\forall u \in User (m. email \in ValidEmailFormat)$$

**Ý nghĩa:** Đảm bảo chỉ lưu trữ các email hợp lệ.

**Bảng tầm ảnh hưởng:**

Bảng 2-3: Bảng tầm ảnh hưởng ràng buộc liên thuộc tính bảng User

|      | Thêm | Xóa | Sửa      |
|------|------|-----|----------|
| User | +    | -   | +(email) |

##### b. Ràng buộc liên bộ

**Tên ràng buộc:** UNQ\_User\_Email

**Mô tả:** Email phải là duy nhất trong bảng User

**Nội dung:**

$$\forall u1, u2 \in User, u1 \neq u2 \Rightarrow u1. email \neq u2. email$$

**Ý nghĩa:** Đảm bảo không có hai người dùng nào có cùng email.

**Bảng tầm ảnh hưởng:**

Bảng 2-4: Bảng tầm ảnh hưởng ràng buộc liên bộ bảng User

|      | Thêm | Xóa | Sửa      |
|------|------|-----|----------|
| User | +    | -   | +(email) |

#### 2.3.1.3.2. Bảng Authentication

##### a. Ràng buộc liên thuộc tính

**Tên ràng buộc:** CHK\_Auth\_Provider

**Mô tả:** Nguồn gốc đăng nhập (provider) phải có giá trị và google với việc đăng nhập bằng google hoặc là local nếu đăng nhập theo phương thức truyền thống.

**Nội dung:**

$$\forall a \in Authentication \ (a.provider \ IN \ \{ 'google', 'local' \})$$

**Ý nghĩa:** Đảm bảo người dùng chỉ có thể xác thực qua 2 phương thức mà hệ thống cung cấp.

**Bảng tầm ảnh hưởng:**

Bảng 2-5: Bảng tầm ảnh hưởng ràng buộc liên thuộc tính bảng Auth

|                       | Thêm | Xóa | Sửa         |
|-----------------------|------|-----|-------------|
| <b>Authentication</b> | +    | -   | +(provider) |

### b. Ràng buộc liên bộ

**Tên ràng buộc:** UNQ\_Password

**Mô tả:** Nguồn gốc đăng nhập(provider) phải kèm theo giá trị mật khẩu (password) phù hợp. Đối với google thì không cần mật khẩu, ngược lại nếu là local phải có một mật khẩu phù hợp.

**Nội dung:**

$$\forall u \in Authentication,$$

$$u.provider = 'google' \Rightarrow u.password = NULL$$

$$\forall u \in Authentication, u.provider = 'local' \Rightarrow u.password \neq NULL$$

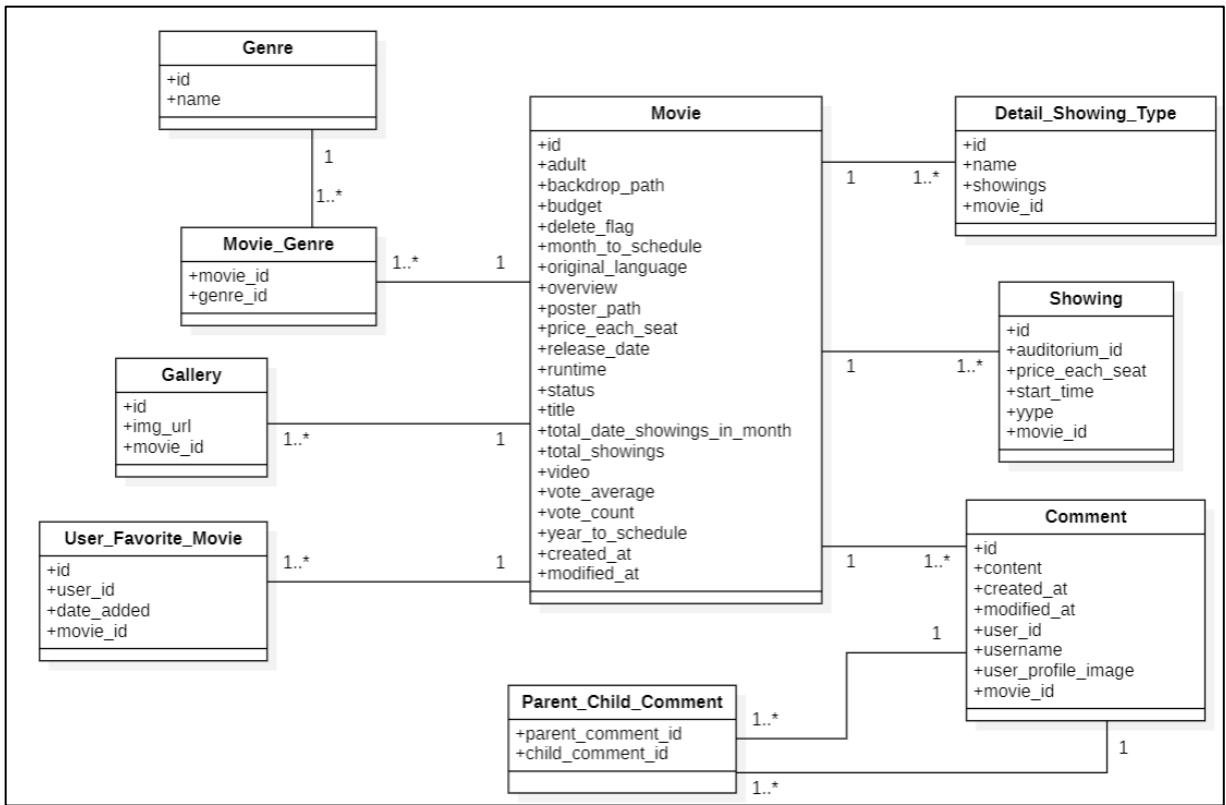
**Ý nghĩa:** Đảm bảo các phương thức đăng nhập khác nhau đều có dữ liệu hợp lệ.

**Bảng tầm ảnh hưởng:**

Bảng 2-6: Bảng tầm ảnh hưởng ràng buộc liên bộ bảng Auth

|                       | Thêm | Xóa | Sửa                   |
|-----------------------|------|-----|-----------------------|
| <b>Authentication</b> | +    | -   | +(provider, password) |

### 2.3.2. Cơ sở dữ liệu của Movie Service



Hình 2-8: Cơ sở dữ liệu của Movie Service

Movie Service dùng để lưu trữ, quản lý các dữ liệu liên quan đến các bộ phim bao gồm các thông tin như: title, video, ... Công việc chính của service này là xử lý các thao tác liên quan đến đối tượng chính của dự án là các bộ phim. Ngoài ra nó còn có thể thực hiện quá trình sắp lịch từ một danh sách các bộ phim.

```
{
  "id": 1,
  "title": "Ultraman: Rising",
  "adult": false,
  "budget": 0,
  "originalLanguage": "en",
  "overview": "A star athlete reluctantly returns home to take over his father's duties as Ultraman, shielding Tokyo from giant monsters as he becomes a legendary hero.",
  "status": "Released",
  "video": "no",
  "posterPath": "https://image.tmdb.org/t/p/w500/i6JHSNF6JdjI7mpEqAcYRhu52qg.jpg",
  "backdropPath": "http://localhost:8272/api/v2/moon-movie/media/images/defaultBackdrop.jpg",
  "voteAverage": 8.392,
  "voteCount": 222,
  "runtime": 117,
  "releaseDate": "2024-06-14",
  "deleteFlag": false,
  "createdAt": "2024-11-07T03:18:05",
  "modifiedAt": "2024-11-07T03:18:05",
  "genres": [],
  "monthToSchedule": 8,
  "yearToSchedule": 2024,
  "totalShowings": 15,
  "totalDateShowingsInMonth": 10,
  "priceEachSeat": 78000,
  "detailShowingTypes": [],
  "galleries": [],
  "userFavoriteMovies": []
}
```

Hình 2-9: Một document của Movie service dưới dạng đối tượng

### 2.3.2.1. Thực thể Movie

Bộ phim (Movie): Đây là thực thể để lưu thông tin của một bộ phim trong hệ thống. Đây có thể coi là một thực thể đóng vai trò quan trọng nhất trong toàn bộ hệ thống này.

Bảng 2-7: Thực thể Movie

| Tên thuộc tính | Diễn giải               | Kiểu dữ liệu | Khóa chính | Khóa ngoại | Ràng buộc |
|----------------|-------------------------|--------------|------------|------------|-----------|
| <b>id</b>      | Mã bộ phim              | int          | X          |            | Not null  |
| adult          | Phim dành cho người lớn | boolean      |            |            | Not null  |

| Tên thuộc tính               | Diễn giải                                 | Kiểu dữ liệu | Khóa chính | Khóa ngoại | Ràng buộc |
|------------------------------|---|--------------|------------|------------|-----------|
| backdrop_path                | Ảnh nền                                   | varchar(255) |            |            | Not null  |
| budget                       | Ngân sách                                 | Int          |            |            | $\geq 0$  |
| delete_flag                  | Đánh dấu bộ phim được xóa                 | Boolean      |            |            | Not null  |
| month_to_schedule            | Tháng để sắp lịch                         | Int          |            |            | [1,12]    |
| original_language            | Ngôn ngữ gốc                              | varchar(64)  |            |            | Not null  |
| overview                     | Tổng quan                                 | text         |            |            | Not null  |
| poster_path                  | Ảnh poster                                | varchar(255) |            |            | Not null  |
| price_each_seat              | Giá khi đặt vé                            | int          |            |            | $\geq 0$  |
| release_date                 | Ngày ra mắt                               | datetime     |            |            | Not null  |
| runtime                      | Tổng thời gian chiếu                      | int          |            |            | $>0$      |
| status                       | Trạng thái                                | varchar(10)  |            |            | Not null  |
| title                        | Tên phim                                  | varchar(150) |            |            | Not null  |
| total_date_showings_in_month | Tổng số ngày cản sắp lịch trong một tháng | Int          |            |            | Not null  |
| total_showings               | Tổng số suất chiếu phim trong một tháng   | Int          |            |            | $\geq 0$  |
| video                        | trailer                                   | varchar(255) |            |            | Not null  |
| vote_average                 | Điểm đánh giá trung bình                  | int          |            |            | $\geq 0$  |
| vote_count                   | Tổng số lượt đánh giá                     | int          |            |            | $\geq 0$  |
| year_to_schedule             | Năm sắp lịch                              | int          |            |            | $\geq 0$  |
| created_at                   | Ngày thêm                                 | datetime     |            |            |           |
| modified_at                  | Ngày chỉnh sửa                            | datetime     |            |            |           |

### 2.3.2.2. Thực thể Genre

Thể loại (Genre): Là thể loại của một bộ phim. Thực thể này mô tả một bộ phim bao gồm những thể loại nào.

Bảng 2-8: Thực thể Genre

| Tên thuộc tính | Diễn giải    | Kiểu dữ liệu | Khóa chính | Khóa ngoại | Ràng buộc |
|----------------|--------------|--------------|------------|------------|-----------|
| <b>id</b>      | Mã thể loại  | Int          | X          |            | Not null  |
| name           | Tên thể loại | varchar(60)  |            |            | Not null  |

### 2.3.2.3. Thực thể Detail Showing Type

Các chi tiết về suất chiếu (Detail Showing Type): Là thực thể thể hiện chi tiết về các suất chiếu của một bộ phim.

Bảng 2-9: Thực thể Detail Showing Type

| Tên thuộc tính  | Diễn giải                   | Kiểu dữ liệu | Khóa chính | Khóa ngoại | Ràng buộc |
|-----------------|-----------------------------|--------------|------------|------------|-----------|
| <b>id</b>       | Mã                          | int          | X          |            | Not null  |
| <b>movie_id</b> | Mã bộ phim                  | int          |            | X          | Not null  |
| name            | Tên loại suất chiếu         | varchar(60)  |            |            | Not null  |
| showings        | Tổng số suất chiếu loại này | int          |            |            | Not null  |

### 2.3.2.4. Thực thể Gallery

Thư viện hình ảnh (Gallery): Là thực thể để lưu trữ thư viện ảnh của một bộ phim bao gồm các ảnh nền, poster và các hình ảnh liên qua đến bộ phim.

Bảng 2-10: Thực thể Gallery

| Tên thuộc tính  | Diễn giải    | Kiểu dữ liệu | Khóa chính | Khóa ngoại | Ràng buộc |
|-----------------|--------------|--------------|------------|------------|-----------|
| <b>id</b>       | Mã hình ảnh  | int          | X          |            | Not null  |
| <b>movie_id</b> | Mã bộ phim   | int          |            | X          | Not null  |
| img_url         | Url hình ảnh | varchar(60)  |            |            | Not null  |

### 2.3.2.5. Thực thể User Favorite Movie

Các movie yêu thích của User (User Favorite Movie): thực thể này lưu lại thông tin danh sách các bộ phim yêu thích của user để họ có thể theo dõi thông tin về bộ phim vào các lần duyệt web tiếp theo.

Bảng 2-11: Thực thể User Favorite Movie

| Tên thuộc tính    | Diễn giải  | Kiểu dữ liệu | Khóa chính | Khóa ngoại | Ràng buộc |
|-------------------|--|--------------|------------|------------|-----------|
| <b>id</b>         | Mã   | int          | X          |            | Not null  |
| <b>movie_id</b>   | Mã bộ phim   | int          |            | X          | Not null  |
| <b>user_id</b>    | Id người dùng  | varchar(24)  |            | X          | Not null  |
| <b>date_added</b> | Ngày người dùng thêm bộ phim vào danh sách các bộ phim yêu thích | datetime     |            |            |           |

### 2.3.2.6. Thực thể Showing

Suất chiếu phim (Showing): là thực thể đại diện cho thông tin của một suất chiếu phim cụ thể. Nó sẽ bao gồm các thông tin như: loại suất chiếu, giá của suất chiếu, thời gian bắt đầu chiếu phim,...

Bảng 2-12: Thực thể Showing

| Tên thuộc tính       | Diễn giải         | Kiểu dữ liệu | Khóa chính | Khóa ngoại | Ràng buộc |
|----------------------|-------------------|--------------|------------|------------|-----------|
| <b>id</b>            | Mã suất chiếu     | int          | X          |            | Not null  |
| <b>movie_id</b>      | Mã bộ phim        | int          |            | X          | Not null  |
| <b>auditorium_id</b> | Id rạp chiếu      | varchar(24)  |            | X          | Not null  |
| price each seat      | Giá               | int          |            |            | >=0       |
| start time           | Thời gian bắt đầu | datetime     |            |            | Not null  |
| type                 | Thể loại          | varchar(11)  |            |            | Not null  |

### 2.3.2.7. Thực thể Comment

Bình luận (Comment): là thực thể đại diện cho thông tin của một đánh giá của các khách hàng sau khi trải nghiệm dịch vụ của website, đánh giá về nội dung bộ phim hoặc bàn luận với các người dùng khác,... Các khách hàng cũng có thể trả lời lại bình luận của các khách hàng khác để lại sự tiện lợi khi chúng ta muốn trò chuyện và trao đổi với nhau.

Bảng 2-13: Thực thể Comment

| Tên thuộc tính | Diễn giải    | Kiểu dữ liệu | Khóa chính | Khóa ngoại | Ràng buộc |
|----------------|--------------|--------------|------------|------------|-----------|
| <b>id</b>      | Mã bình luận | int          | X          |            | Not null  |

| Tên thuộc tính     | Diễn giải                        | Kiểu dữ liệu | Khóa chính | Khóa ngoại | Ràng buộc |
|--------------------|----------------------------------|--------------|------------|------------|-----------|
| movie_id           | Mã bộ phim                       | int          |            | X          | Not null  |
| user_id            | Id người dùng                    | varchar(24)  |            | X          | Not null  |
| content            | Nội dung                         | Int          |            |            | Not null  |
| created_at         | Ngày tạo                         | datetime     |            |            |           |
| modified_at        | Ngày chỉnh sửa                   | datetime     |            |            |           |
| username           | Username người bình luận         | varchar(30)  |            |            | Not null  |
| user_profile_image | Url ảnh đại diện người bình luận | varchar(255) |            |            | Not null  |

### 2.3.2.8. Các ràng buộc toàn vẹn dữ liệu

#### 2.3.2.8.1. Bảng Movie

##### a. Ràng buộc liên thuộc tính

Tên ràng buộc: CHK\_Movie\_ReleaseDate

Mô tả: Giá trị của thuộc tính release\_date (ngày phát hành) trong bảng Movie phải nhỏ hơn hoặc bằng ngày hiện tại.

Nội dung:

$$\forall m \in Movie (m.release_date \leq current_date)$$

Ý nghĩa: Đảm bảo không thể lưu trữ ngày phát hành trong tương lai cho một bộ phim vì không thể tiến hành bán vé cho một bộ phim chưa phát hành.

Bảng tầm ảnh hưởng:

Bảng 2-14: Bảng tầm ảnh hưởng ràng buộc liên thuộc tính bảng Movie

|       | Thêm | Xóa | Sửa             |
|-------|------|-----|-----------------|
| Movie | +    | -   | +(release_date) |

##### b. Ràng buộc liên bộ

Tên ràng buộc: UNQ\_Movie\_Title\_Year

Mô tả: Tên phim (title) và năm phát hành trong (release\_date) phải là duy nhất trong bảng.

Nội dung:

$$\forall m1, m2 \in Movie, m1 \neq m2 \Rightarrow (m1.title$$

$$\neq m2.title \vee m1.release_date.year \neq m2.release_date.year)$$

Ý nghĩa: Đảm bảo không có hai bộ phim trùng tên được phát hành cùng một năm.

Bảng tầm ảnh hưởng:

Bảng 2-15: Bảng tầm ảnh hưởng ràng buộc liên bộ bảng Movie

|       | Thêm | Xóa | Sửa                      |
|-------|------|-----|--------------------------|
| Movie | +    | -   | +(title,<br>realse_date) |

### 2.3.2.8.2. Bảng Showing

#### a. Ràng buộc liên thuộc tính

Tên ràng buộc: CHK\_Showing\_Type

Mô tả: Giá trị của thuộc tính type (loại suất chiếu) phải là 1 trong bốn loại và hệ thống cung cấp bao gồm (2D, 2D subtitles, 3D, 3D subtitles).

Nội dung:

$\forall s \in Showing (s.type \in \{2D, 2D subtitles, 3D, 3D subtitles\})$

Ý nghĩa: Đảm bảo các suất chiếu có đúng các thể loại mà hệ thống cung cấp.

Bảng tầm ảnh hưởng:

Bảng 2-16: Bảng tầm ảnh hưởng ràng buộc liên thuộc tính bảng Showing

|         | Thêm | Xóa | Sửa     |
|---------|------|-----|---------|
| Showing | +    | -   | +(type) |

#### b. Ràng buộc liên bộ.

Tên ràng buộc: UNQ\_Showing\_Start\_Time

Mô tả: Thời gian bắt đầu một suất chiếu (start\_time) ở một rạp chiếu (auditorium\_id) phải là duy nhất trong bảng.

Nội dung:

$\forall s1, s2 \in Showing, s1 \neq s2$

$\Rightarrow (s1.start\_time \neq s2.start\_time \vee s1.auditorium\_id \neq s2.auditorium\_id)$

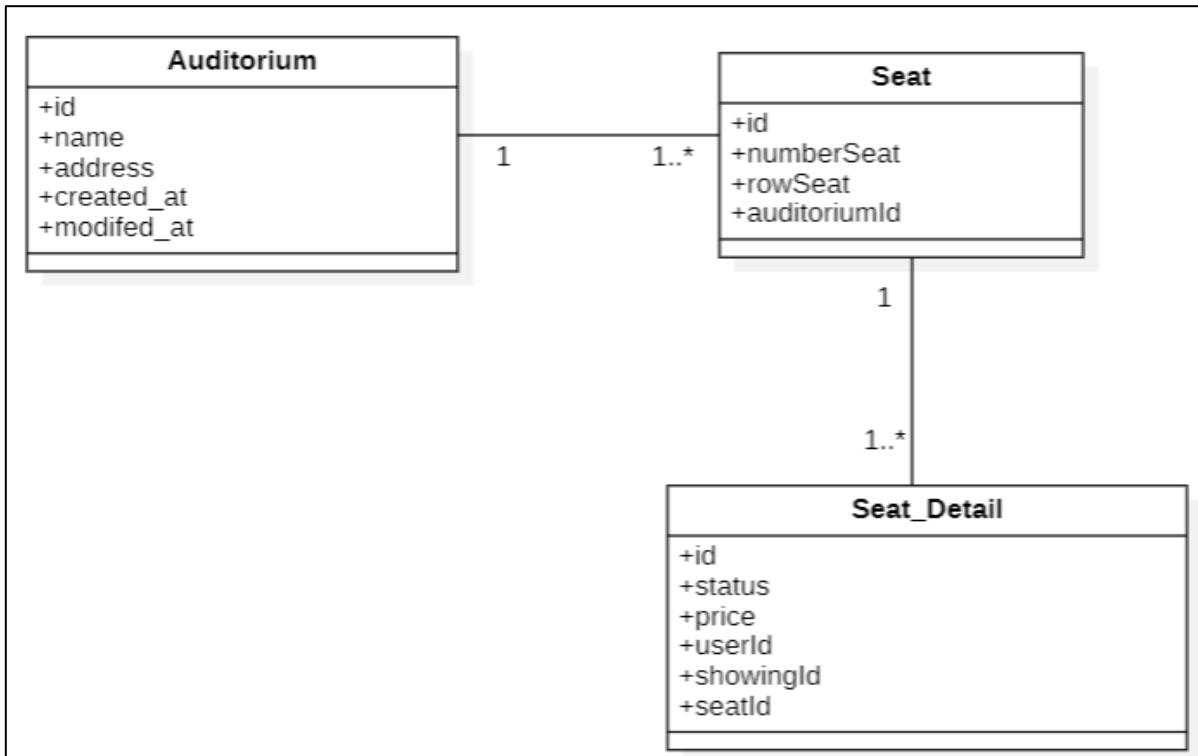
Ý nghĩa: Đảm bảo tại cùng một thời điểm không có một bộ phim được chiếu nhiều lần tại một rạp.

Bảng tầm ảnh hưởng:

Bảng 2-17: Bảng tầm ảnh hưởng ràng buộc liên bộ bảng Showing

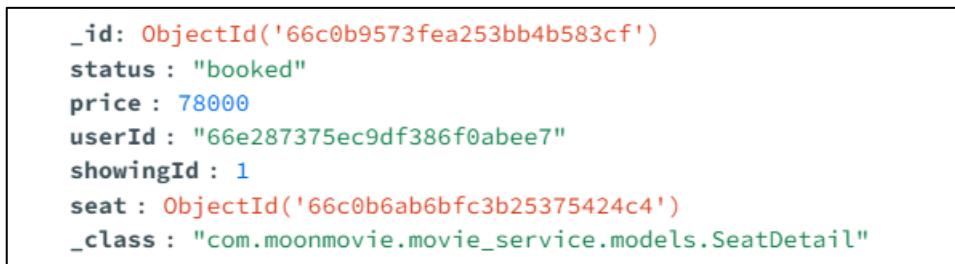
|         | Thêm | Xóa | Sửa                             |
|---------|------|-----|---------------------------------|
| Showing | +    | -   | +(start_time,<br>auditorium_id) |

### 2.3.3. Cơ sở dữ liệu của Seat Service



Hình 2-10: Cơ sở dữ liệu Seat Service

Seat Service là một service xử lý các tác vụ liên quan đến việc đặt và chọn chỗ ngồi từ người dùng. Ngoài ra nó còn cung cấp các tính năng liên quan đến việc quản lý các rạp chiếu phim. Đây là một service phụ trách công việc chọn ghế thời gian thực.



Hình 2-11: Một document của Seat service dưới dạng đối tượng

#### 2.3.3.1. Thực thể Auditorium

Rạp chiếu (Auditorium): là thực thể chứa các thông tin về các rạp chiếu phim.



### 2.3.3.4. Các ràng buộc toàn vẹn dữ liệu

#### 2.3.3.4.1. Bảng Seat

##### a. Ràng buộc liên thuộc tính

**Tên ràng buộc:** CHK\_Seat\_Row

**Mô tả:** Hàng của ghế (rowSeat) phải là các giá trị từ A -> K.

**Nội dung:**

$$\forall s \in \text{Seat} (s.\text{rowSeat} \in \{'A', \dots, 'K'\})$$

**Ý nghĩa:** Đảm bảo có đủ 12 hàng ghế không hơn và thiếu cho từng rạp.

**Bảng tầm ảnh hưởng:**

Bảng 2-21: Bảng tầm ảnh hưởng ràng buộc liên thuộc tính bảng Seat

|      | Thêm | Xóa | Sửa        |
|------|------|-----|------------|
| Seat | +    | -   | +(rowSeat) |

##### b. Ràng buộc liên bộ

**Tên ràng buộc:** UNQ\_Num\_Seat\_Row

**Mô tả:** Mỗi hàng ghế (rowSeat) sẽ có số lượng ghế (numberSeat) nhất định.

**Nội dung:**

$$\begin{aligned} \forall s \in \text{Seat}, s.\text{rowSeat} = r \Rightarrow & s.\text{numberSeat} > 0 \vee s.\text{numberSeat} \\ & < \text{MaxNumOfSeatsPerRow} \end{aligned}$$

**Ý nghĩa:** Đảm bảo các hàng ghế có số lượng ghế đúng chuẩn layout của rạp chiếu.

**Bảng tầm ảnh hưởng:**

Bảng 2-22: Bảng tầm ảnh hưởng ràng buộc liên thuộc bộ bảng Seat

|      | Thêm | Xóa | Sửa                       |
|------|------|-----|---------------------------|
| Seat | +    | -   | +(rowSeat,<br>numberSeat) |

#### 2.3.3.4.2. Bảng Seat Detail

##### a. Ràng buộc liên thuộc tính

**Tên ràng buộc:** CHK\_Seat\_Detail\_Status

**Mô tả:** Trạng thái (status) của chi tiết ghế phải là các giá trị available, choosing, và booked.

**Nội dung:**

$$\forall sd \in \text{Seat_Detail} (sd.\text{status} \in \{'available', 'choosing', 'booked'\})$$

**Ý nghĩa:** Đảm bảo các trạng thái ghế mà hệ thống có thể xử lý được.

### Bảng tầm ảnh hưởng:

Bảng 2-23: Bảng tầm ảnh hưởng ràng buộc liên thuộc tính bảng SeatDetail

|             | Thêm | Xóa | Sửa       |
|-------------|------|-----|-----------|
| Seat_Detail | +    | -   | +(status) |

### b. Ràng buộc liên bộ

Tên ràng buộc: UNQ\_User\_Seat

Mô tả: Các trạng thái của ghế (status) phải đi kèm với cột giá trị người dùng (userId).

### Nội dung:

$\forall sd \in SeatDetail, sd.status = 'available' \Rightarrow sd.userId = NULL$

$\forall sd \in SeatDetail, sd.status = 'choosing' \Rightarrow sd.userId \neq NULL$

$\forall sd \in SeatDetail, sd.status = 'booked' \Rightarrow sd.userId \neq NULL$

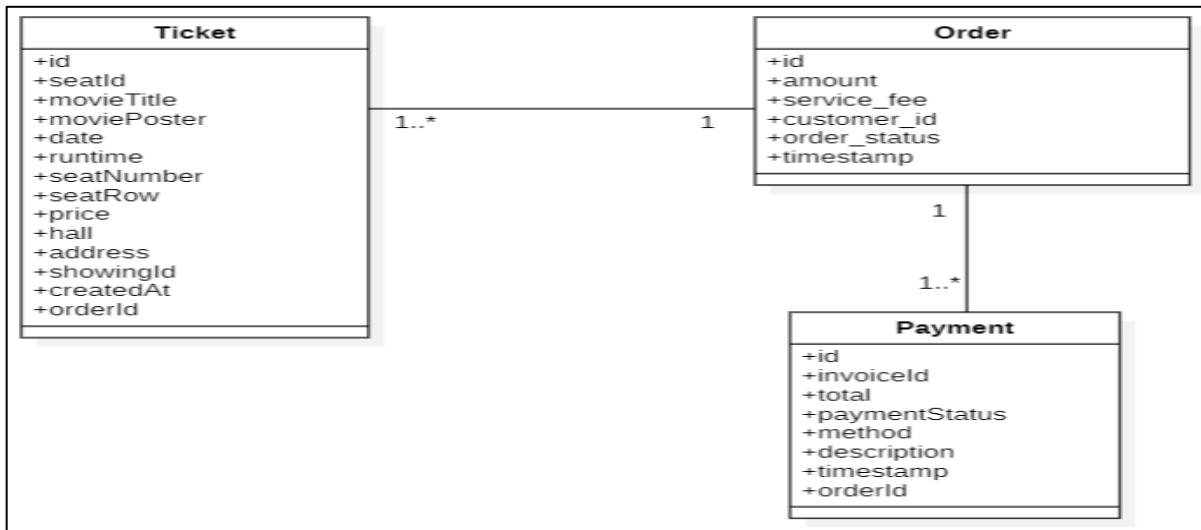
Ý nghĩa: Đảm bảo các trạng thái được chọn và đã đặt có thể truy xuất thông tin ghế khi cần thiết.

### Bảng tầm ảnh hưởng:

Bảng 2-24: Bảng tầm ảnh hưởng ràng buộc liên bộ bảng SeatDetail

|             | Thêm | Xóa | Sửa               |
|-------------|------|-----|-------------------|
| Seat_Detail | +    | -   | +(status, userId) |

### 2.3.4. Cơ sở dữ liệu của Reservation Service



Hình 2-12: Cơ sở dữ liệu Reservation Service

Reservation service là một service phụ trách các tác vụ liên quan đến việc đặt vé, xem lại thông tin vé và thanh toán online.

```

_id: ObjectId('66e564b81395da68ce11b15c')
amount: 234000
serviceFee: 11700
customerId: "66e452d1f5aec52245921e14"
orderStatus: "complete"
timestamp: 2024-09-14T10:26:00.734+00:00
payments: Array (1)
  0: Object
    invoiceId: "240914_954974"
    total: 234000
    paymentStatus: "paid"
    method: "zalopay"
    description: "Pay for Moon Movie tickets"
    timestamp: 2024-09-14T10:26:55.573+00:00
    _class: "com.moonmovie.movie_service.models.Order"
  
```

Hình 2-13: Một document của Reservation service dưới dạng đối tượng

#### 2.3.4.1. Thực thể Ticket

Vé (Ticket): Là một thực thể lưu các thông tin chi tiết về vé vào của một buổi chiếu phim bao gồm các thông tin cơ bản như tên bộ phim, số ghế, hàng ghế, rạp chiếu, thời gian bắt đầu,...

Bảng 2-25: Thực thể Ticket

| Tên thuộc tính   | Điễn giải               | Kiểu dữ liệu | Khóa chính | Khóa ngoại | Ràng buộc |
|------------------|-------------------------|--------------|------------|------------|-----------|
| <b>id</b>        | Mã vé                   | varchar(24)  | X          |            | Not null  |
| <b>seatId</b>    | Mã ghế ngồi             | varchar(24)  |            | X          | Not null  |
| <b>orderId</b>   | Id đơn hàng vé này      | varchar(24)  |            | X          | Not null  |
| <b>showingId</b> | Id suất chiếu           | int          |            | X          | Not null  |
| <b>movieId</b>   | Id bộ phim              | int          |            | X          | Not null  |
| movie_title      | Tên bộ phim             | varchar(150) |            |            | Not null  |
| movie_poster     | Poster bộ phim          | varchar(255) |            |            | Not null  |
| date             | Thời gian chiếu         | datetime     |            |            | Not null  |
| runtime          | Tổng thời lượng bộ phim | int          |            |            | Not null  |
| seatNumber       | Số ghế                  | int          |            |            | >=0       |
| seatRow          | Hàng ghế                | varchar(1)   |            |            | Not null  |
| price            | Giá vé                  | int          |            |            | >=0       |

| Tên thuộc tính | Điễn giải          | Kiểu dữ liệu | Khóa chính | Khóa ngoại | Ràng buộc |
|----------------|--------------------|--------------|------------|------------|-----------|
| hall           | Tên rạp chiếu phim | varchar(30)  |            |            | Not null  |
| address        | Địa chỉ rạp chiếu  | varchar(255) |            |            | Not null  |
| createdAt      | Ngày tạo           | datetime     |            |            |           |

#### 2.3.4.2. Thực thể Order

Đơn hàng (Order): là thực thể lưu các thông tin của đơn hàng sau khi một khách hàng thực hiện quá trình mua vé. Nó sẽ bao gồm các thông tin thanh toán và phương thức đã chọn để thanh toán.

Bảng 2-26: Thực thể Order

| Tên thuộc tính    | Điễn giải            | Kiểu dữ liệu | Khóa chính | Khóa ngoại | Ràng buộc |
|-------------------|----------------------|--------------|------------|------------|-----------|
| <b>id</b>         | Mã đơn hàng          | varchar(24)  | X          |            | Not null  |
| <b>customerId</b> | Mã khách hàng        | varchar(24)  |            | X          | Not null  |
| amount            | Tổng tiền            | int          |            |            | $\geq 0$  |
| serviceFee        | Phí dịch vụ hệ thống | Int          |            |            | $\geq 0$  |
| orderStatus       | Trạng thái           | varchar(7)   |            |            | Not null  |
| timestamp         | Thời gian tạo        | datetime     |            |            |           |

#### 2.3.4.3. Thực thể Payment

Chi tiết thanh toán (Payment): là thực thể lưu các thông tin thanh toán khi khách hành thực hiện thanh toán qua bên thanh toán thứ ba.

Bảng 2-27: Thực thể Payment

| Tên thuộc tính | Điễn giải              | Kiểu dữ liệu | Khóa chính | Khóa ngoại | Ràng buộc |
|----------------|------------------------|--------------|------------|------------|-----------|
| <b>id</b>      | Mã hóa đơn             | varchar(24)  | X          |            | Not null  |
| <b>orderId</b> | Mã đơn hàng            | varchar(24)  |            | X          | Not null  |
| total          | Tổng thanh toán        | int          |            |            | $\geq 0$  |
| paymentStatus  | Trạng thái thanh toán  | varchar(10)  |            |            | Not null  |
| method         | Phương thức thanh toán | varchar(7)   |            |            | Not null  |
| description    | Mô tả                  | text         |            |            |           |
| timestamp      | Thời gian tạo          | datetime     |            |            |           |

#### 2.3.4.4. Các ràng buộc toàn vẹn dữ liệu

##### 2.3.4.4.1. Bảng Order

###### a. Ràng buộc liên thuộc tính

Tên ràng buộc: CHK\_Order\_Status

Mô tả: Trạng thái của đơn hàng (status) phải có một trong các giá trị create, pending, complete, cancel.

Nội dung:

$\forall o \in Order (s.status \in \{ 'create', 'pending', 'complete', 'cancel' \})$

Ý nghĩa: Đảm bảo các trạng thái của order cụ thể cho từng xử lý.

Bảng tầm ảnh hưởng:

Bảng 2-28: Bảng tầm ảnh hưởng ràng buộc liên thuộc tính bảng Order

|       | Thêm | Xóa | Sửa       |
|-------|------|-----|-----------|
| Order | +    | -   | +(status) |

###### b. Ràng buộc liên bộ

Tên ràng buộc: DP\_Order\_Amount\_Fee

Mô tả: Phí dịch vụ (serviceFee) là chi phí để duy trì hệ thống phải luôn bằng 5% tổng giá trị đơn hàng (amount).

Nội dung:

$\forall o \in Order, s.serviceFee = s.amount * 0.05$

Ý nghĩa: Đảm bảo cách tính phí dịch vụ một cách chính xác.

Bảng tầm ảnh hưởng:

Bảng 2-29: Bảng tầm ảnh hưởng ràng buộc liên bộ bảng Order

|       | Thêm | Xóa | Sửa                   |
|-------|------|-----|-----------------------|
| Order | +    | -   | +(amount, serviceFee) |

##### 2.3.4.4.2. Bảng Ticket

###### a. Ràng buộc liên thuộc tính

Tên ràng buộc: CHK\_Ticket\_Seat

Mô tả: Mã số ghế (seatId) phải là một trong các mã số ghế có trong bảng Seat.

Nội dung:

$\forall t \in Ticket, t.seatId \in \{ s.id \mid s \in Seat \}$

Ý nghĩa: Đảm bảo chỗ ngồi cho vé đã được đặt.

Bảng tầm ảnh hưởng:

Bảng 2-30: Bảng tầm ảnh hưởng ràng buộc liên thuộc tính bảng Ticket

|        | Thêm | Xóa | Sửa       |
|--------|------|-----|-----------|
| Ticket | +    | -   | +(seatId) |

### b. Ràng buộc liên bộ

**Tên ràng buộc:** UNQ\_Ticket

**Mô tả:** Thông tin bộ phim trên vé như tên bộ phim (movieTitle) và poster của bộ phim (moviePoster) phải thuộc cùng một bộ phim.

**Nội dung:**

$$\begin{aligned} \forall t \in Ticket, m \in Movie, m.id = t.movieId \Rightarrow t.movieTitle \\ = m.title, t.moviePoster = m.posterPath \end{aligned}$$

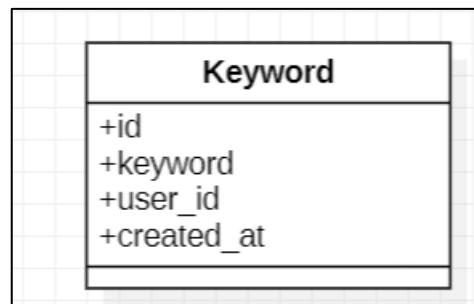
**Ý nghĩa:** Đảm bảo vé có các thông tin đúng.

**Bảng tầm ảnh hưởng:**

Bảng 2-31: Bảng tầm ảnh hưởng ràng buộc liên bộ bảng Ticket

|        | Thêm | Xóa | Sửa                           |
|--------|------|-----|-------------------------------|
| Ticket | +    | -   | +(movieTitle,<br>moviePoster) |

### 2.3.5. Cơ sở dữ liệu của Recommend Service



Hình 2-14: Cơ sở dữ liệu của Recommend Service

Recommend Service là một dịch vụ chuyên xử lý các tác vụ liên quan đến việc nâng cao trải nghiệm người dùng bao gồm việc gợi ý các tên bộ phim thông qua từ khóa tìm kiếm, gợi ý các bộ phim tương tự, lưu lịch sử tìm kiếm của người dùng... Do đây là một service đặc biệt lấy dữ liệu từ cơ sở dữ liệu của Movie Service để nạp các thông tin ban đầu nên cơ sở dữ liệu khá đơn giản chỉ bao gồm một bảng để có thể lưu lại lịch sử các từ khóa tìm kiếm của user.

### 2.3.5.1. Thực thể Keyword

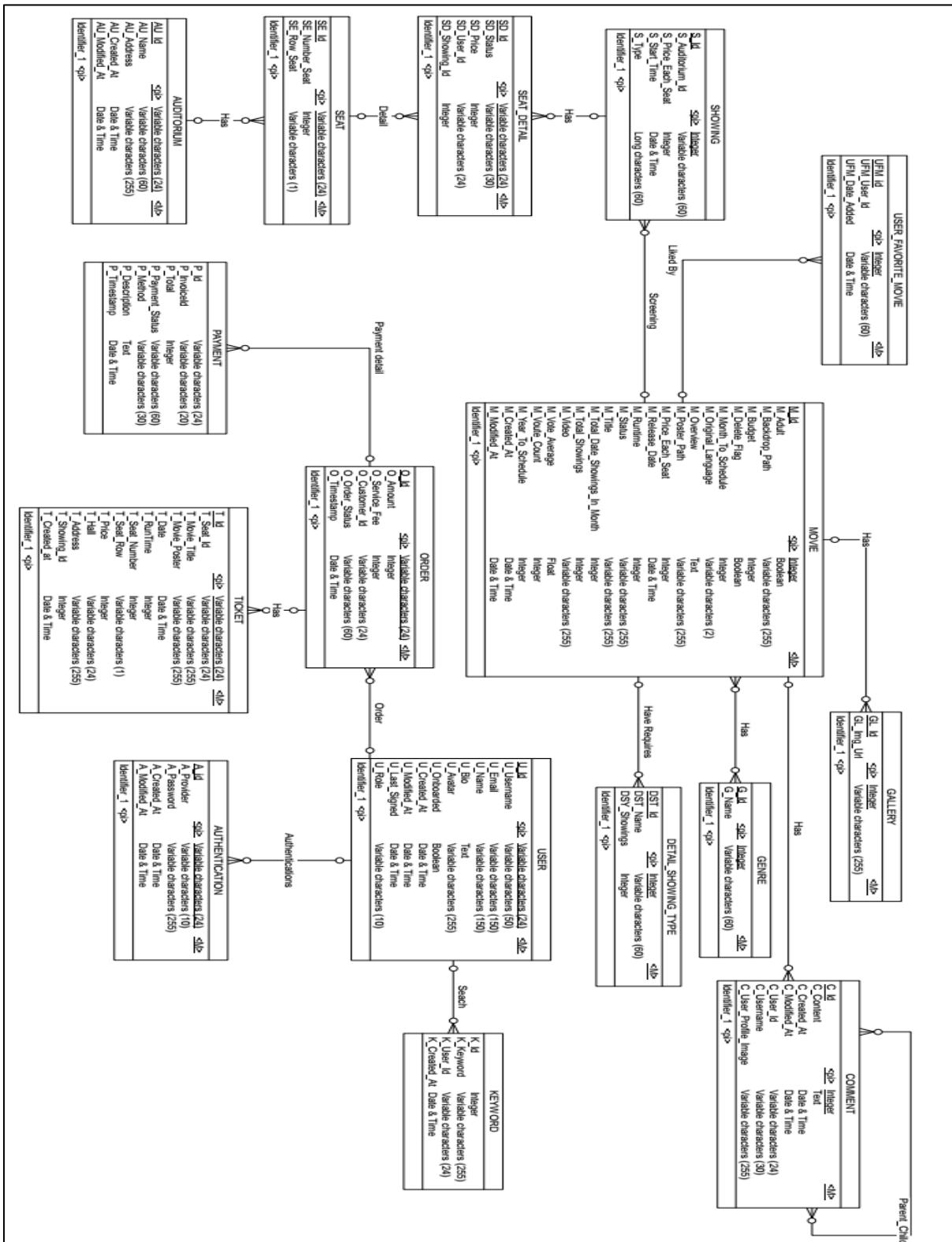
Thực thể Keyword (từ khóa): là một thực thể dùng để lưu thông tin các từ khóa tìm kiếm của user trước đó nhằm nâng cao trải nghiệm người dùng khi họ sử dụng website.

Bảng 2-32: Thực thể Keyword

| Tên thuộc tính | Điễn giải        | Kiểu dữ liệu | Khóa chính | Khóa ngoại | Ràng buộc |
|----------------|------------------|--------------|------------|------------|-----------|
| <b>id</b>      | Mã từ khóa       | varchar(24)  | X          |            | Not null  |
| <b>userId</b>  | Id người dùng    | varchar(24)  |            | X          | Not null  |
| keyword        | Từ khóa tìm kiếm | varchar(255) |            |            |           |
| created_at     | Ngày thêm        | datetime     |            |            |           |

### 2.3.6. Mô hình CDM toàn hệ thống

## Hệ thống bán vé xem phim trực tuyến



Hình 2-15: Sơ đồ thực thể quan hệ

Đây là mô hình thể hiện mối quan hệ của tất cả các thực thể trong toàn bộ dự án. Do dự án được thiết kế theo kiến trúc microservices nên đây như là một sơ đồ quan hệ tổng thể để ta có thể dễ dàng nắm bắt cũng như có cái nhìn tổng quát nhất cách mà các đối tượng có các mối quan hệ với nhau.

## 2.4. Thiết kế giao diện

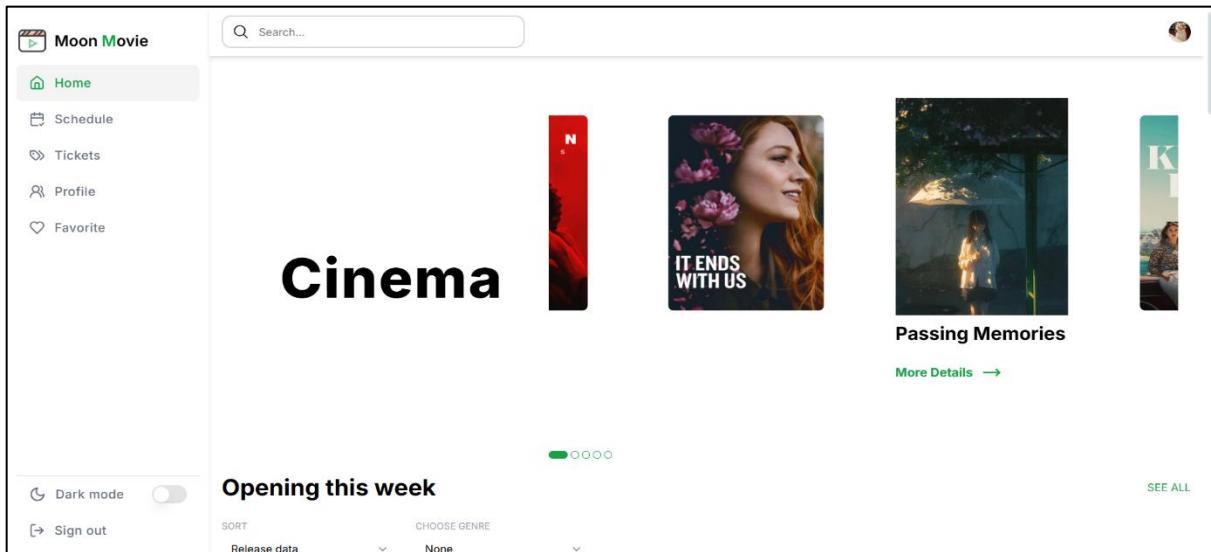
### 2.4.1. Mục tiêu

- Trải nghiệm người dùng (UX): Đảm bảo rằng người dùng có thể dễ dàng sử dụng và thao tác các chức năng trên hệ thống, từ việc xem lịch chiếu, đặt vé đến thanh toán.
- Tính thẩm mỹ (UI): Thiết kế giao diện hấp dẫn và chuyên nghiệp, phù hợp với thương hiệu của hệ thống.
- Tương thích đa nền tảng: Đảm bảo giao diện hiển thị tốt trên cả máy tính và thiết bị di động, phù hợp với thói quen người dùng hiện đại.

### 2.4.2. Cấu trúc các màn hình chính

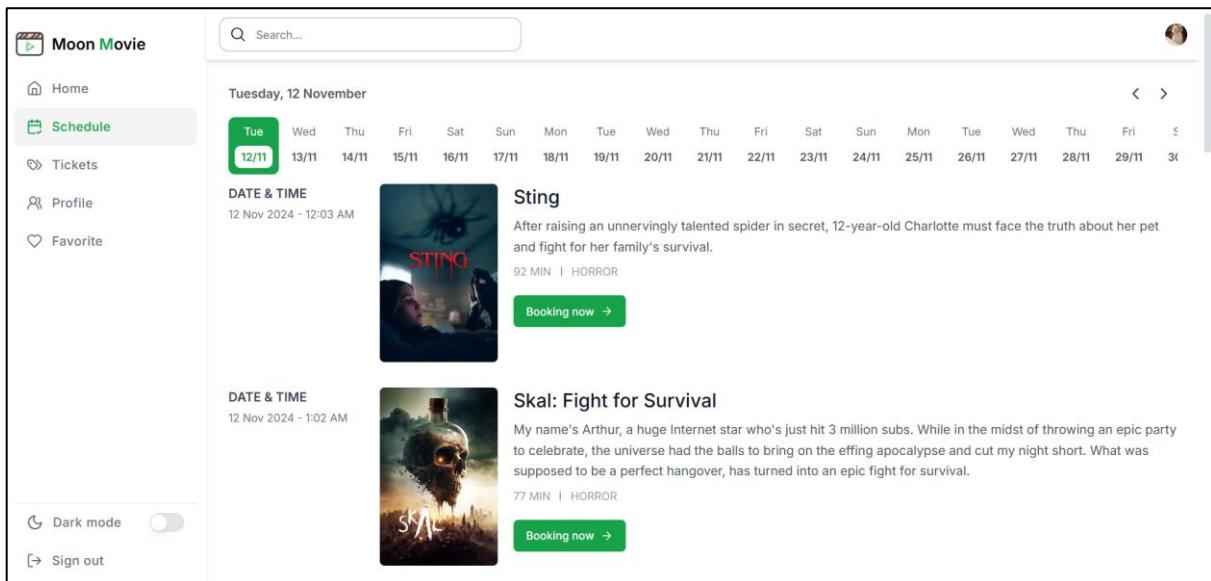
- **Trang chủ:** Hiển thị các phim nổi bật, suất chiếu sắp tới. Cung cấp nút truy cập nhanh đến các chức năng chính như xem lịch chiếu, đăng nhập và đăng ký.
- **Trang lịch chiếu:** Người dùng có thể xem các suất chiếu theo phim và thời gian các suất chiếu cụ thể.
- **Trang các vé đã đặt:** Người dùng có thể xem lại chi tiết các vé đã đặt ở trang này.
- **Trang chọn vé:** Hiển thị giao diện chọn ghế thân thiện, giúp người dùng dễ dàng chọn được chỗ ngồi.
- **Trang tài khoản cá nhân:** Cho phép người dùng xem thông tin cá nhân và quản lý tài khoản một cách thuận tiện.

### 2.4.3. Minh họa



Hình 2-16: Giao diện trang Home

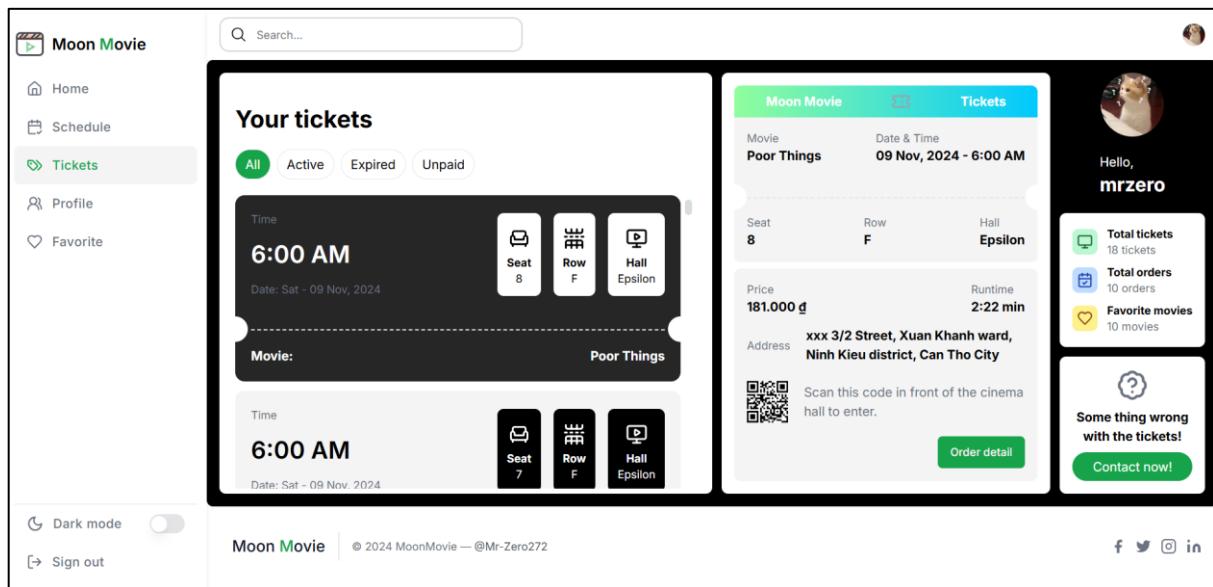
Trang home là trang chủ của hệ thống, ở đây nó sẽ hiển thị các bộ phim nổi bật đang của hệ thống.



Hình 2-17: Giao diện trang Schedule

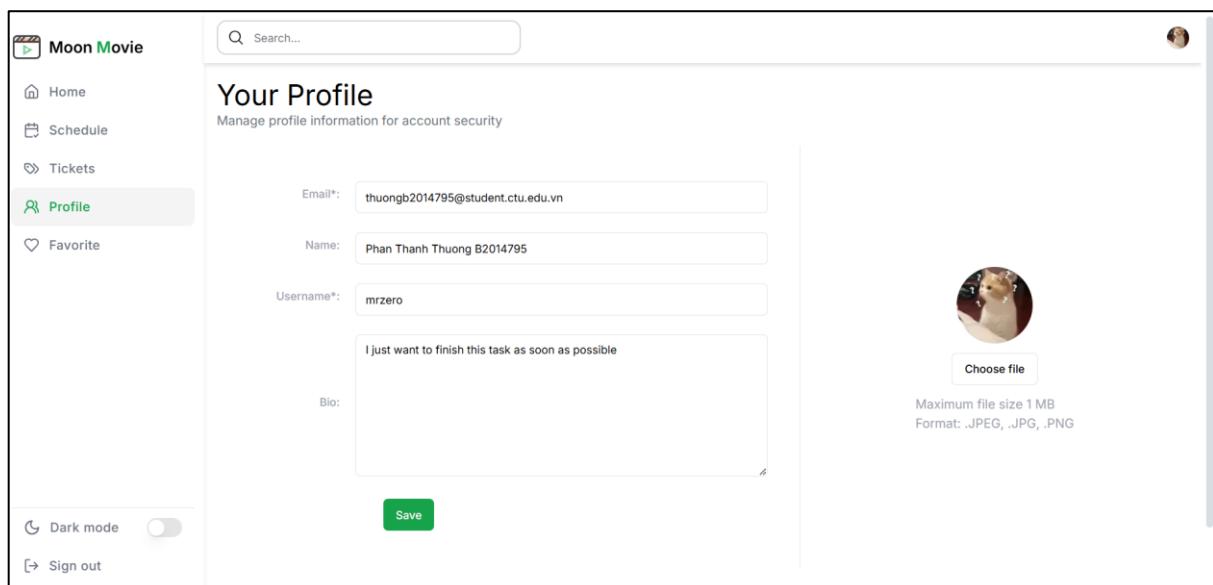
Trang schedule là trang mà người dùng quan tâm nhất khi nó hiển thị các suất chiếu trong ngày và các ngày tiếp theo.

## Hệ thống bán vé xem phim trực tuyến



Hình 2-18: Giao diện trang Tickets

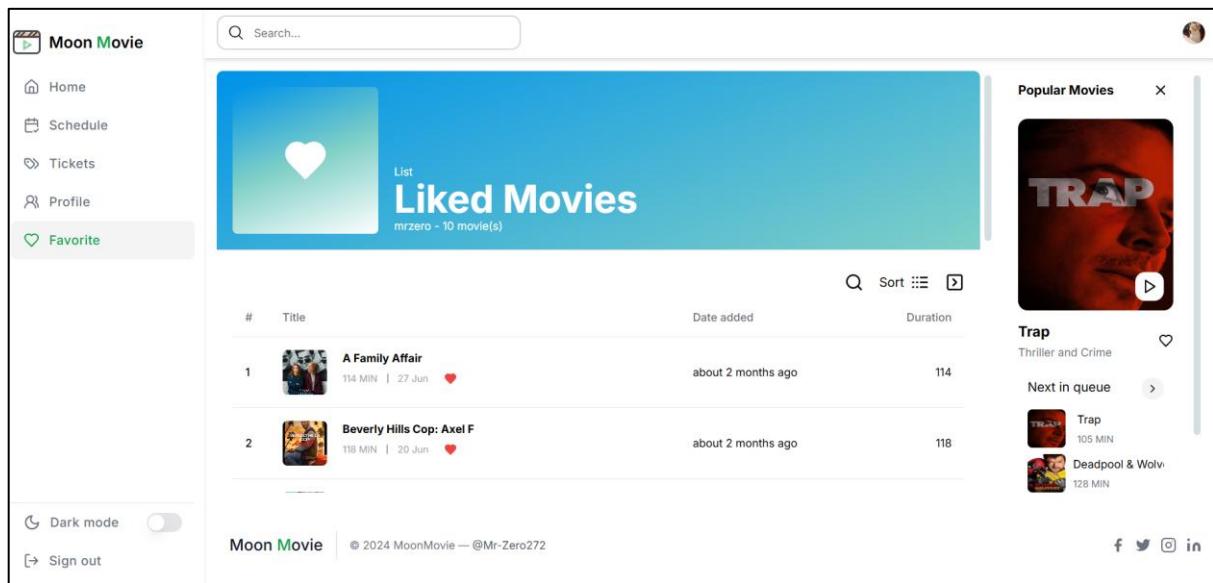
Trang Tickets là nơi mà ta có thể theo dõi lại các vé mà chúng ta đã đặt.



Hình 2-19: Giao diện trang Profile

Trang Profile là trang quản lý thông tin của người dùng, nơi mà người dùng có thể cập nhật lại các thông tin của họ.

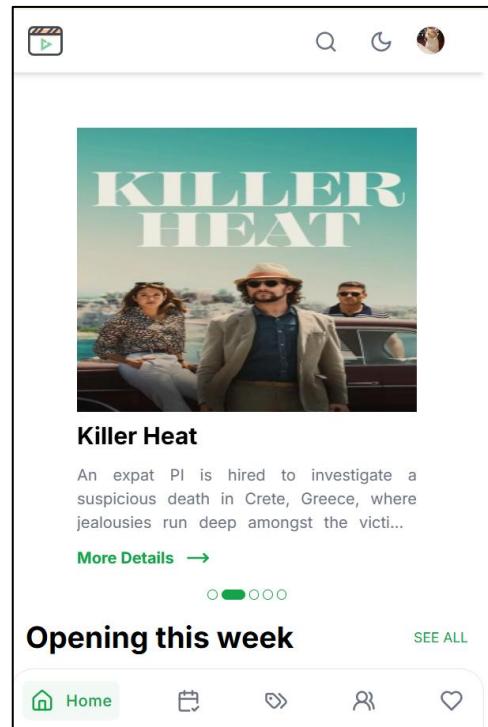
## Hệ thống bán vé xem phim trực tuyến



Hình 2-20: Giao diện trang Favorite

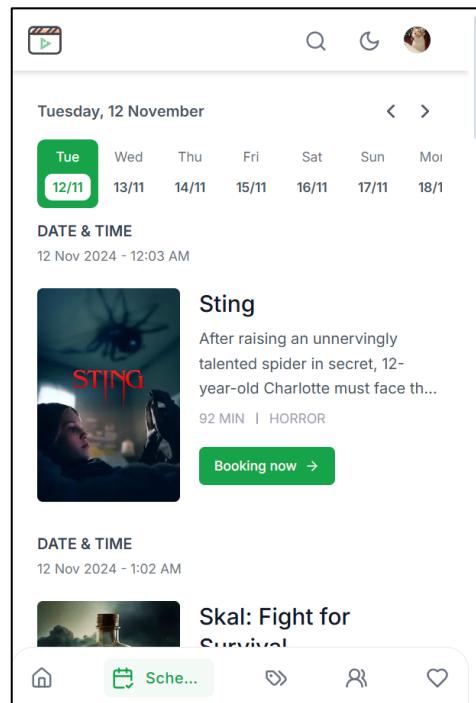
Trang Favorite là nơi lưu lại các bộ phim yêu thích của người dùng cho các lần tiếp theo truy cập website.

Phản tiếp theo là các giao diện tương tự nhưng ở các thiết bị mobile.

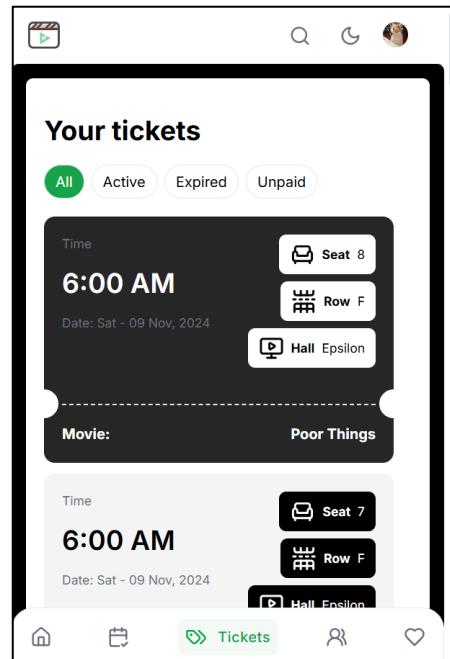


Hình 2-21: Giao diện mobile trang Home

## Hệ thống bán vé xem phim trực tuyến

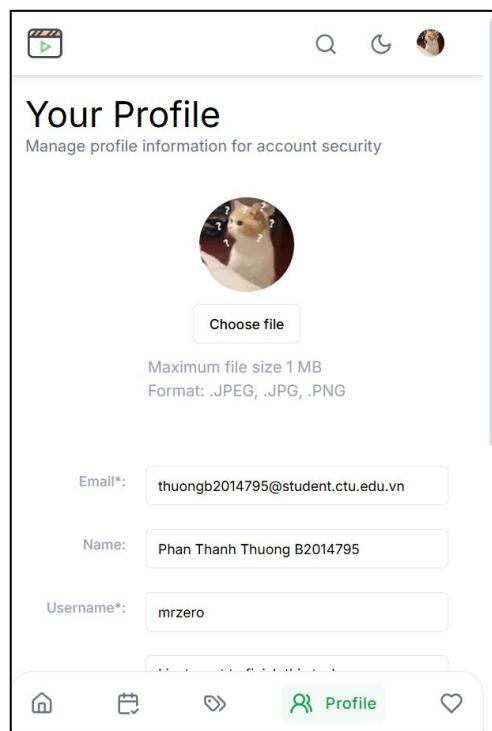


Hình 2-22: Giao diện mobile trang Schedule

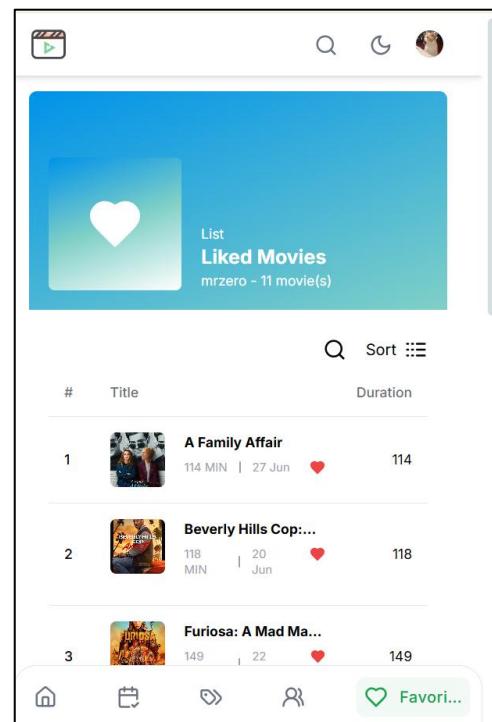


Hình 2-23: Giao diện mobile trang Tickets

## Hệ thống bán vé xem phim trực tuyến



Hình 2-24: Giao diện mobile trang Profile



Hình 2-25: Giao diện mobile trang Favorite

## 2.5. Thiết kế các dịch vụ chính

### 2.5.1. Movie Service

#### 2.5.1.1. Tổng quan

Mục tiêu: Movie Service có nhiệm vụ xử lý các tác vụ liên quan đến thực thể chính của dự án là các bộ phim bao gồm các công việc như lưu trữ, cung cấp các thông tin bộ phim, thêm mới và chỉnh sửa các bộ phim. Ngoài ra nó còn có nhiệm vụ quản lý các thành phần khác có liên quan đến các bộ phim như là các thể loại và các bình luận của bộ phim.

Các chức năng chính:

- Thêm, xóa và cập nhật thông tin phim.
- Try xuất danh sách các bộ phim theo tên, thể loại, ngôn ngữ và trạng thái của phim.
- Cung cấp chi tiết về một bộ phim cụ thể.
- Thông kê số lượng phim được thêm mới mỗi tháng, số lượng bộ phim được sắp lịch mỗi tháng.
- Cung cấp thông tin chi tiết về một suất chiếu cụ thể.
- Cung cấp danh sách các bộ phim tương tự với một bộ phim nào đó.
- Sắp lịch chiếu phim.
- Thêm, xóa và cập nhật thể loại phim.
- Truy xuất danh sách các thể loại theo tên
- Truy xuất danh sách các lịch chiếu phim theo tên phim, rạp chiếu và ngày chiếu.
- Thêm, xóa và cập nhật một lịch chiếu phim
- Thêm một và xóa một bộ phim ở danh sách các bộ phim yêu thích.
- Truy xuất danh sách các bộ phim yêu thích.
- Thêm, xóa và chỉnh sửa một bình luận về một bộ phim.
- Thêm, xóa và chỉnh sửa một trả lời của một bình luận về một bộ phim.
- Lấy danh sách các bình luận của một bộ phim.
- Lấy danh sách các câu hỏi đáp của một bình luận.

#### 2.5.1.2. Thêm mới bộ phim

Mục đích: cho phép người quản lý hệ thống thêm mới một bộ phim vào hệ thống.

Giao diện:

**Movie title**

1 Title

**Genres**

2 Search Mockups, Logos...

**Budget**      **Runtime (min)**

3 0      0 4

**Vote average**      **Vote count**      **Release date**

5 0      0 6      11/30/2024 7

**Original language**

8 English

**Overview**

9 Overview...

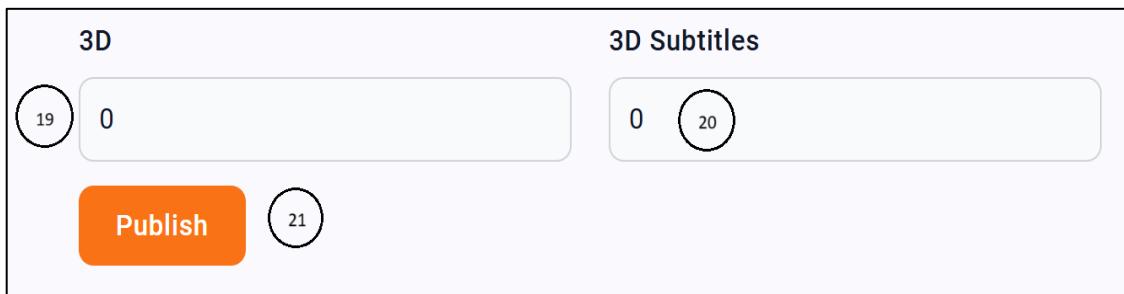
10 Adult movie

**Movie images** 11

You need at least 2 vertical photos and 2 horizontal photos for this movie and need at least 4 photos. You can add more photos but 2 vertical and 2 horizontal photo photos are required.

**Movie trailer** 12

|                             |                            |
|-----------------------------|----------------------------|
| <b>Total date showings</b>  | <b>Price for each seat</b> |
| 13 0                        | 0 14                       |
| <b>Month to schedule</b>    | <b>Year to schedule</b>    |
| 15 01                       | 2024 16                    |
| <b>Detail showing types</b> |                            |
| 2D                          | <b>2D Subtitles</b>        |
| 17 0                        | 0 18                       |



Hình 2-26: Giao diện trang Thêm mới bộ phim

Các thành phần trong giao diện:

Bảng 2-33: Thành phần giao diện chức năng Thêm mới bộ phim

| STT | Loại điều khiển | Mô tả                               |
|-----|-----------------|-------------------------------------|
| 1   | Textbox         | Tên phim                            |
| 2   | Combobox        | Thể loại                            |
| 3   | Textbox         | Ngân sách                           |
| 4   | Textbox         | Thời gian chiếu                     |
| 5   | Textbox         | Điểm vote trung bình                |
| 6   | Textbox         | Tổng số lượng vote                  |
| 7   | Textbox         | Ngày ra mắt                         |
| 8   | Select          | Ngôn ngữ                            |
| 9   | Textarea        | Tổng quan                           |
| 10  | Checkbox        | Có phải là phim dành cho người lớn  |
| 11  | Select file     | Chọn ảnh cho bộ phim                |
| 12  | Select file     | Chọn trailer cho bộ phim            |
| 13  | Textbox         | Tổng số suất chiếu cần sắp lịch     |
| 14  | Textbox         | Giá của ghế ngồi                    |
| 15  | Textbox         | Tháng để sắp lịch                   |
| 16  | Textbox         | Năm để sắp lịch                     |
| 17  | Textbox         | Số suất chiếu cho loại 2D           |
| 18  | Textbox         | Số suất chiếu cho loại 2D Subtitles |
| 19  | Textbox         | Số suất chiếu cho loại 3D           |
| 20  | Textbox         | Số suất chiếu cho loại 3D Subtitles |
| 21  | Button          | Nút lưu                             |

Dữ liệu được sử dụng:

Bảng 2-34: Dữ liệu được sử dụng của chức năng Thêm mới bộ phim

| STT | Tên bảng/ Cấu trúc dữ liệu | Phương thức |     |     |          |
|-----|----------------------------|-------------|-----|-----|----------|
|     |                            | Thêm        | Sửa | Xóa | Truy vấn |
| 1   | Movie                      | x           |     |     | x        |

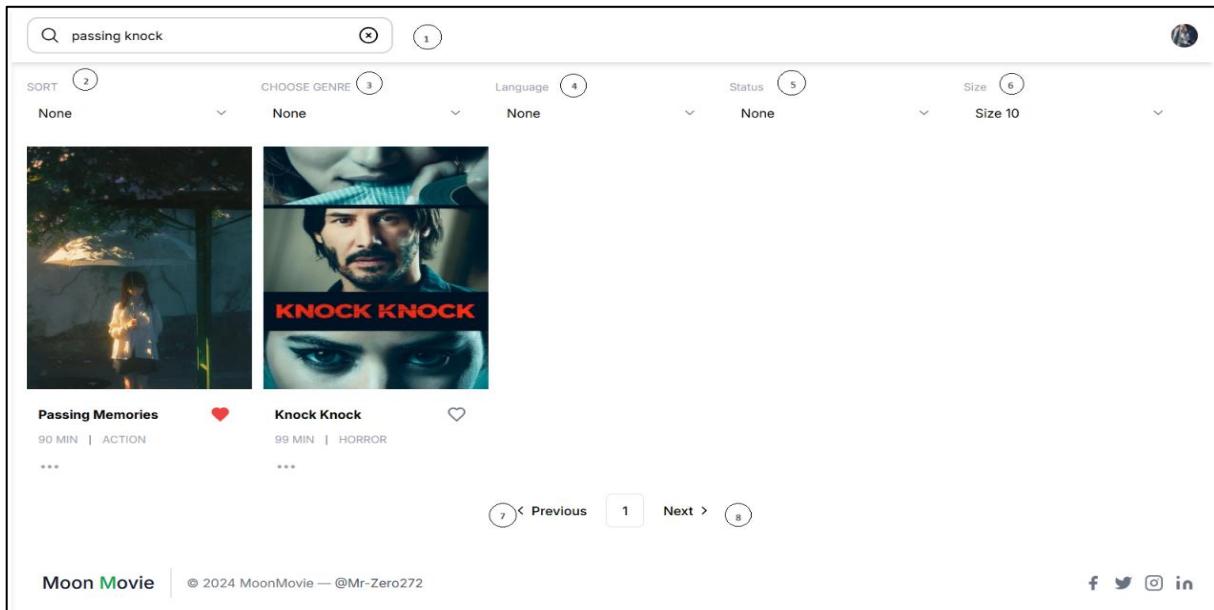
### Luồng xử lý:

- Người dùng tiến hành nhập các thông tin để thêm mới bộ phim. Các trường đều có những điều kiện nhất định, phải đảm bảo thỏa mãn tất cả các điều kiện đó.
- Người dùng ấn nút **Publish**
- Yêu cầu thêm mới bộ phim được gửi về phía server
- Server phản hồi lại cho người dùng:
  - o Nếu có bộ phim cùng tên đã được thêm trước đó, thông báo lỗi.
  - o Nếu tháng sắp lịch đã được sắp lịch rồi thông báo lỗi.
  - o Nếu số suất chiếu còn lại nhỏ hơn số suất chiếu đang muốn thêm thông báo lỗi.
  - o Ngược lại thông báo thêm mới bộ phim thành công. Reset form để có thể tiếp tục thêm bộ phim khác.

#### 2.5.1.3. Try xuất danh sách các bộ phim theo từ khóa

Mục đích: Cho phép khách hàng hoặc người quản lý có thể dễ dàng tìm kiếm các bộ phim mà mình mong muốn. Đặc biệt hệ thống hỗ trợ người dùng tìm kiếm theo từ khóa, tức là không cần nhớ toàn bộ tên bộ phim. Ngoài ra api còn cung cấp thêm các phần lọc để giảm thời gian tìm kiếm cũng như tích hợp phân trang.

Giao diện:



Hình 2-27: Giao diện trang Search

Các thành phần trong giao diện:

Bảng 2-35: Thành phần giao diện chức năng Truy xuất danh sách phim

| STT | Loại điều khiển | Mô tả                     |
|-----|-----------------|---------------------------|
| 1   | Textbox         | Thanh tìm kiếm            |
| 2   | Select          | Sắp xếp                   |
| 3   | Select          | Chọn thể loại             |
| 4   | Select          | Chọn ngôn ngữ             |
| 5   | Select          | Chọn trạng thái           |
| 6   | Select          | Chọn kích thước của trang |
| 7   | Button          | Trang trước               |
| 8   | Button          | Trang sau                 |

Dữ liệu được sử dụng:

Bảng 2-36: Dữ liệu được sử dụng của chức năng Tìm kiếm phim

| STT | Tên bảng/ Cấu trúc dữ liệu | Phương thức |     |     |          |
|-----|----------------------------|-------------|-----|-----|----------|
|     |                            | Thêm        | Sửa | Xóa | Truy vấn |
| 1   | Movie                      |             |     |     | x        |
| 2   | Genre                      |             |     |     | x        |

Luồng xử lý:

- Người dùng nhập các từ khóa vào thanh tìm kiếm. Có thể thêm các tùy chọn lọc để nhanh chóng tìm kiếm được kết quả mong muốn.
- Người dùng ấn **Enter**
- Yêu cầu tìm kiếm các bộ phim được gửi về phía server
- Server phản hồi lại cho người dùng:
  - o Nếu không có kết quả phù hợp hiển thị thông báo.
  - o Ngược lại thì hiển thị các kết quả.
- Sau khi tìm được kết quả người dùng có thể tiếp tục áp dụng bộ lọc để tiếp tục lọc lại kết quả tìm kiếm. Ngoài ra url cũng có thể chia sẻ cho những người dùng khác với kết quả tương tự.

#### 2.5.1.4. Cung cấp chi tiết về một bộ phim

Mục đích: giúp khách hàng có thêm chi tiết về một bộ phim cụ thể họ có thể tìm hiểu tổng quan bộ phim, các thông tin có liên quan, các bình luận, đánh giá của người dùng khác về bộ phim này...

Giao diện:

## Hệ thống bán vé xem phim trực tuyến

Passing Memories

Action Comedy Romance

Average vote 7.8

Watch trailer

Showings

Saturday, 09 November

|       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |       |
|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
| Sat   | Sun   | Mon   | Tue   | Wed   | Thu   | Fri   | Sat   | Sun   | Mon   | Tue   | Wed   | Thu   | Fri   | Sat   | Sun   | Mon   | Tue   |
| 09/11 | 10/11 | 11/11 | 12/11 | 13/11 | 14/11 | 15/11 | 16/11 | 17/11 | 18/11 | 19/11 | 20/11 | 21/11 | 22/11 | 23/11 | 24/11 | 25/11 | 26/11 |

This date does not have any showings. ☹

Booking now →

**Detail**

Budget \$1,240,000.00

Language en

Runtime 90 min

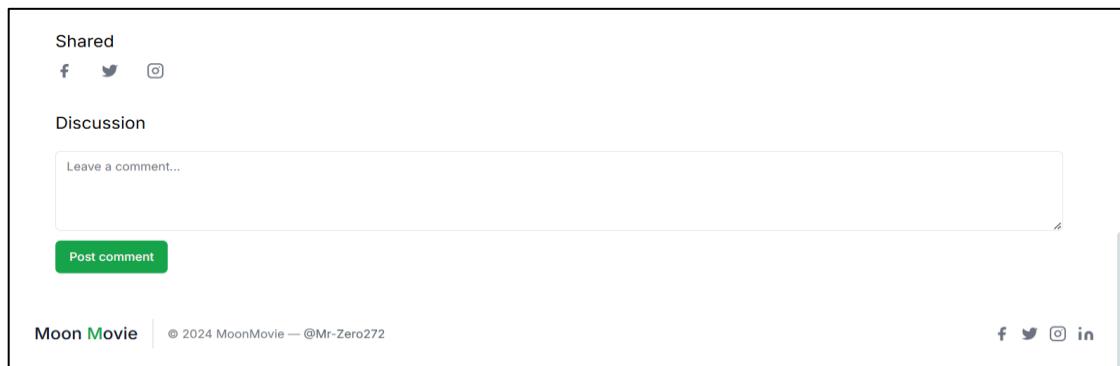
Release 03 Oct 2024

**Overview**

**Gallery**

Recommend movies

|  |   |  |   |  |
|--|---|--|---|--|
|  |   |  |   |  |
| <b>The Family Plan</b><br>118 MIN   ACTION | <b>The Fall Guy</b><br>126 MIN   ACTION | <b>My Spy The Eternal City</b><br>103 MIN   COMEDY | <b>Anyone But You</b><br>103 MIN   ROMANTIC | <b>The Family Plan</b><br>118 MIN   ACTION |



Hình 2-28: Giao diện Chi tiết phim

Dữ liệu được sử dụng:

Bảng 2-37: Dữ liệu được sử dụng của chức năng Truy xuất chi tiết phim

| STT | Tên bảng/ Cấu trúc dữ liệu | Phương thức |     |     |          |
|-----|----------------------------|-------------|-----|-----|----------|
|     |                            | Thêm        | Sửa | Xóa | Truy vấn |
| 1   | Movie                      |             |     |     | x        |
| 2   | Comment                    |             |     |     | x        |
| 3   | Showing                    |             |     |     | x        |

Luồng xử lý:

- Người dùng khi truy xuất các bộ phim có thể ấn vào bộ phim để có thể xem chi tiết một bộ phim.
- Yêu cầu lấy thông tin chi tiết về một bộ phim được gửi về phía server
- Server phản hồi lại cho người dùng:
  - o Hiển thị các thông tin của một bộ phim, kèm theo các suất chiếu theo ngày và các bình luận về bộ phim đó.

#### 2.5.1.5. Sắp lịch

Mục đích: giúp người quản lý nhanh chóng sắp lịch cho các movie trong một tháng nhất định. Ngoài ra admin vẫn có thể thêm các suất chiếu bằng tay, hoặc chỉnh sửa các suất chiếu sau khi sắp lịch.

Giao diện:

Hình 2-29: Giao diện Sắp lịch

Các thành phần trong giao diện:

Bảng 2-38: Thành phần giao diện chức năng Sắp lịch

| STT | Loại điều khiển | Mô tả        |
|-----|-----------------|--------------|
| 1   | Button          | Nút trở về   |
| 2   | Select          | Chọn tháng   |
| 3   | Select          | Chọn năm     |
| 4   | Button          | Nút sắp lịch |

Dữ liệu được sử dụng:

Bảng 2-39 : Dữ liệu được sử dụng của chức năng Sắp lịch

| STT | Tên bảng/ Cấu trúc dữ liệu | Phương thức |     |     |          |
|-----|----------------------------|-------------|-----|-----|----------|
|     |                            | Thêm        | Sửa | Xóa | Truy vấn |
| 1   | Showing                    | x           |     |     |          |
| 2   | SeatDetail                 | x           |     |     |          |

Luồng xử lý:

- Người quản lý chọn tháng và năm muôn sắp lịch. Tuy vậy phải đảm bảo là ta đã chỉnh sửa thông tin cho các bộ phim muôn sắp lịch ở các trường như tháng sắp lịch, năm sắp lịch, số suất chiếu,...
- Người dùng ám **Schedule**
- Yêu cầu sắp lịch được gửi về phía server
- Server phản hồi lại cho người dùng:
  - o Nếu người dùng không phải ADMIN, thông báo lỗi.
  - o Nếu tháng đã chọn đã được sắp lịch, thông báo lỗi.
  - o Nếu Seat Service đang không hoạt động, thông báo lỗi.

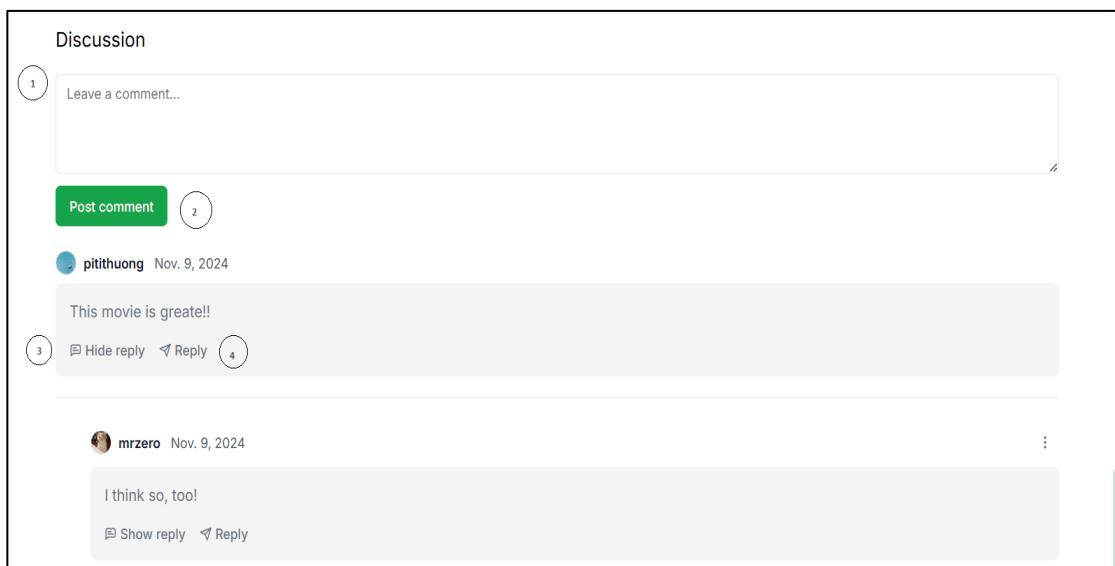
## Hệ thống bán vé xem phim trực tuyến

- Ngược lại sắp lịch các bộ phim theo yêu cầu của từng bộ phim và hiển thị kết quả.
- Sau khi sắp lịch xong người quản lý có thể xem lại thống kê tổng số suất chiếu theo ngày ở tab bên trái.

### 2.5.1.6. Thêm mới bình luận

Mục đích: để người dùng có thể để lại các bình luận sau khi xem xong một bộ phim, hoặc là có thể bàn luận cũng những người dùng khác.

Giao diện:



Hình 2-30: Giao diện Bình luận

Các thành phần trong giao diện:

Bảng 2-40: Thành phần giao diện chức năng Thêm mới bình luận

| STT | Loại điều khiển | Mô tả                   |
|-----|-----------------|-------------------------|
| 1   | Textarea        | Nội dung bình luận      |
| 2   | Button          | Đăng comment            |
| 3   | Button          | Hiển thị các replies    |
| 4   | Button          | Trả lời lại một comment |

Dữ liệu được sử dụng:

Bảng 2-41: Dữ liệu được sử dụng của chức năng Thêm mới bình luận

| STT | Tên bảng/ Cấu trúc dữ liệu | Phương thức |     |     |          |
|-----|----------------------------|-------------|-----|-----|----------|
|     |                            | Thêm        | Sửa | Xóa | Truy vấn |
| 1   | Comment                    | x           | x   | x   | x        |

Luồng xử lý:

- Người dùng để lại nội dung bình luận.
- Người dùng ấn **Post comment**
- Yêu cầu thêm mới comment được gửi về phía server
- Server phản hồi lại cho người dùng:
  - o Hiển thị comment mới.
- Đối với chức năng trả lời một comment có luồng hoạt động tương tự.

### 2.5.2. Authenticate Service

#### 2.5.2.1. Tổng quan

Mục tiêu: Authenticate service là service phục vụ các tác vụ liên quan đến việc xác thực người dùng, đăng ký mới và chỉnh sửa thông tin người dùng. Ngoài ra service này cũng cung cấp các thống kê về các user đã đăng ký mới theo tháng và năm.

Các chức năng chính:

- Xác thực người dùng với username hoặc email và mật khẩu.
- Xác thực người dùng với tài khoản google.
- Đăng ký mới người dùng.
- Chỉnh sửa thông tin người dùng.
- Thống kê các người dùng mới theo tháng và năm.

#### 2.5.2.2. Chức năng đăng ký

Mục đích: cho phép khách hàng đăng ký mới tài khoản để có thể sử dụng được đa dạng các chức năng do hệ thống cung cấp.

Giao diện:

**Sign Up**

Enter your base information to sign up with us

1 [Sign in with Google](#)

or

Username\*  
2 Enter username

Email\*  
3 mail@loopple.com

Password\*  
4 Enter password

5  Keep me logged in

6 **Sign Up**

Already have an account? [Sign in](#) 7

Hình 2-31: Giao diện Đăng ký

Các thành phần trong giao diện:

Bảng 2-42: Thành phần giao diện chức năng Đăng ký

| STT | Loại điều khiển | Mô tả                                    |
|-----|-----------------|--|
| 1   | Button          | Đăng nhập với google                     |
| 2   | Textbox         | Tên người dùng                           |
| 3   | Textbox         | Email                                    |
| 4   | Textbox         | Mật khẩu                                 |
| 5   | Checkbox        | Nhớ tài khoản cho lần truy cập tiếp theo |
| 6   | Button          | Nút đăng ký                              |
| 7   | Link            | Link chuyển sang trang đăng nhập         |

Dữ liệu được sử dụng:

Bảng 2-43: Dữ liệu được sử dụng của chức năng Đăng ký

| STT | Tên bảng/ Cấu trúc dữ liệu | Phương thức |     |     |          |
|-----|----------------------------|-------------|-----|-----|----------|
|     |                            | Thêm        | Sửa | Xóa | Truy vấn |
| 1   | User                       | x           |     |     | x        |

Luồng xử lý:

- Người dùng tiến hành nhập các thông tin cơ bản cho việc đăng ký bao gồm username, email và mật khẩu. Người dùng cũng có thể tùy chọn nhớ tài khoản cho lần đăng nhập tiếp theo hoặc không.
- Người dùng ấn nút **Sign Up**
- Yêu cầu đăng ký được gửi về phía server
- Server phản hồi lại cho người dùng:
  - o Nếu email được đã được sử dụng cho tài khoản google và username chưa có người sử dụng tạo thêm liên kết với tài khoản google đã có.
  - o Nếu email và username chưa có người sử dụng tạo mới tài khoản.
  - o Các trường hợp còn lại thì thông báo đăng ký người dùng mới thất bại và hiển thị chi tiết lỗi lên màn hình

#### 2.5.2.3. Chức năng đăng nhập

Mục đích: cho phép khách hàng đăng nhập với username hoặc email và password mà người dùng đã đăng ký tài khoản với hệ thống. Hệ thống sử dụng jwt lưu trong cookies để duy trì trạng thái đăng nhập của người dùng. Nếu chọn ghi nhớ đăng nhập người dùng sẽ được duy trì tài khoản 7 ngày, ngược lại chỉ duy trì một phiên làm việc.

Giao diện:

The screenshot shows a 'Sign In' page with the following elements and numbered callouts:

- Sign In** (Title)
- Enter your email and password
- Sign in with Google** button (Callout 1)
- or** separator
- Username or Email\*** input field (Callout 2)
- mail@loopple.com** (Value in input field)
- Password\*** input field (Callout 3)
- Enter password** (Placeholder in input field)
- Keep me logged in** checkbox (Callout 4)
- Forget password?** link (Callout 5)
- Sign In** button (Callout 6)
- Not registered yet? Create an Account** link (Callout 7)

Hình 2-32: Giao diện Đăng nhập

Các thành phần trong giao diện:

Bảng 2-44: Thành phần giao diện chức năng Đăng nhập

| STT | Loại điều khiển | Mô tả                                    |
|-----|-----------------|--|
| 1   | Button          | Đăng nhập với google                     |
| 2   | Textbox         | Tên người dùng hoặc email                |
| 3   | Textbox         | Mật khẩu                                 |
| 4   | Checkbox        | Nhớ tài khoản cho lần truy cập tiếp theo |
| 5   | Link            | Link chuyển sang trang quên mật khẩu     |
| 6   | Button          | Nút đăng nhập                            |
| 7   | Link            | Link chuyển sang trang đăng nhập         |

Dữ liệu được sử dụng:

Bảng 2-45: Dữ liệu được sử dụng của chức năng Đăng nhập

| STT | Tên bảng/ Cấu trúc dữ liệu | Phương thức |     |     |          |
|-----|----------------------------|-------------|-----|-----|----------|
|     |                            | Thêm        | Sửa | Xóa | Truy vấn |
| 1   | User                       |             |     |     | x        |

Luồng xử lý:

- Người dùng có thể sử dụng username hoặc là email cùng với mật khẩu để tiến hành đăng nhập vào hệ thống. Người dùng cũng có thể tùy chọn nhớ tài khoản cho lần đăng nhập tiếp theo hoặc không.
- Người dùng ấn nút **Sign In**
- Yêu cầu đăng nhập được gửi về phía server
- Server phản hồi lại cho người dùng:
  - o Nếu với username hoặc email người dùng cung cấp không tồn tại, hiển thị lỗi.
  - o Nếu tên người dùng hoặc email và mật khẩu cung cấp không khớp, hiển thị lỗi.
  - o Ngược lại thông báo đăng nhập người dùng mới thành công sau đó chuyển họ sang trang home.

#### 2.5.2.4. Chức năng quản lý tài khoản

Mục đích: cho phép khách hàng quản lý và chỉnh sửa thông tin tài khoản cá nhân của họ kể cả việc thay đổi ảnh đại diện.

Giao diện:

The screenshot shows a user profile management interface titled "Your Profile". It includes fields for "Email\*", "Name", "Username\*", and "Bio". There is also a section for uploading a profile picture with a maximum file size of 1 MB and supported formats (.JPEG, .JPG, .PNG). A "Save" button is located at the bottom.

|   |             |   |
|---|-------------|---|
| 1 | Email*      | pitithuong@gmail.com                      |
| 2 | Name        | Thường Phan Thanh                         |
| 3 | Username*   | pitithuong                                |
| 4 | Bio:        | Let's leave the world behind. Just do it. |
| 5 | Choose file | (Placeholder for profile picture)         |
| 6 | Save        |   |

Hình 2-33: Giao diện Quản lý tài khoản

Các thành phần trong giao diện:

Bảng 2-46: Thành phần giao diện chức năng Quản lý tài khoản

| STT | Loại điều khiển | Mô tả                     |
|-----|-----------------|---------------------------|
| 1   | Textbox         | Email                     |
| 2   | Textbox         | Tên người dùng            |
| 3   | Textbox         | Username                  |
| 4   | Textarea        | Tiểu sử người dùng        |
| 5   | Button          | Nút chọn ảnh đại diện mới |
| 6   | Button          | Nút lưu chỉnh sửa         |

Dữ liệu được sử dụng:

Bảng 2-47: Dữ liệu được sử dụng của chức năng Quản lý tài khoản

| STT | Tên bảng/ Cấu trúc dữ liệu | Phương thức |     |     |          |
|-----|----------------------------|-------------|-----|-----|----------|
|     |                            | Thêm        | Sửa | Xóa | Truy vấn |
| 1   | User                       |             | x   |     | x        |

Luồng xử lý:

- Người dùng tiến hành chỉnh sửa các thông tin, hệ thống vẫn cho phép người dùng sửa lại tên đăng nhập và email, tuy vậy vẫn phải đảm bảo là chưa có ai sử dụng và hai trường này là bắt buộc.

- Sau khi chỉnh sửa người dùng ấn nút **Save**
- Yêu cầu cập nhật thông tin được gửi về cho server
- Server phản hồi lại cho người dùng:
  - o Nếu các trường yêu cầu bị bỏ trống, hiển thị lỗi.
  - o Nếu username và email mới thay đổi đã có người sử dụng, hiển thị lỗi.
  - o Ngược lại thì thông báo cập nhật thông tin thành công.

### **2.5.3. Reservation Service**

#### **2.5.3.1. Tổng quan**

Mục tiêu: Reservation service là service phục vụ các tác vụ liên quan đến việc quản lý đơn hàng của người dùng, các vé được người dùng mua cũng như các thông tin thanh toán mà người dùng đã sử dụng. Ngoài ra nó còn cho phép người dùng thanh toán lại các đơn hàng đã thất bại trước đó cũng như tích hợp các api phục vụ việc thanh toán online qua các ví điện tử.

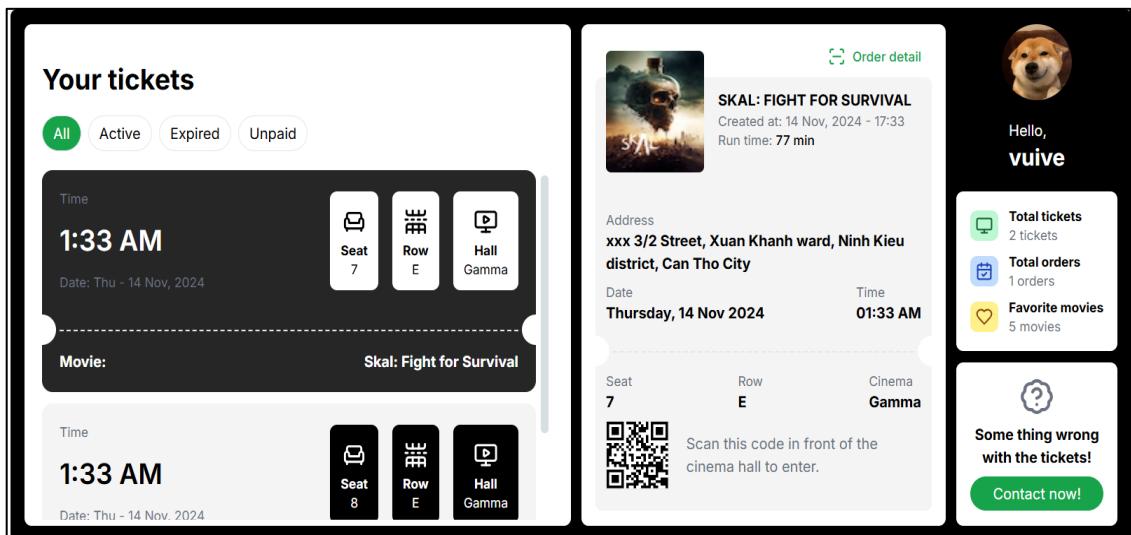
Các chức năng chính:

- Tạo mới đơn hàng.
- Tạo mới thanh toán.
- Thanh toán lại các đơn hàng thất bại.
- Truy xuất thông tin các vé đã được đặt.
- Truy xuất thông tin chi tiết các hóa đơn đã được đặt
- Thanh toán online qua Zalopay và VNPay.
- Thông kê tổng thu theo tháng, tổng đơn hàng.
- Gửi thông tin vé qua email cho khách hàng.

#### **2.5.3.2. Truy xuất thông tin các vé đã đặt**

Mục đích: giúp người dùng có thể truy xuất lại danh sách vé mà mình đã đặt, ngoài ra các vé được sắp xếp theo thời gian mua và cũng kèm một bộ lọc phụ vụ cho việc tìm kiếm hiệu quả hơn.

Giao diện:



Hình 2-34: Giao diện Thông tin vé

Các thành phần trong giao diện:

Bảng 2-48: Thành phần giao diện chức năng Truy xuất thông tin vé

| STT | Loại điều khiển | Mô tả                 |
|-----|-----------------|-----------------------|
| 1   | Select          | Bộ lọc                |
| 2   | Button          | Chi tiết vé           |
| 3   | Button          | Xem chi tiết đơn hàng |

Dữ liệu được sử dụng:

Bảng 2-49: Dữ liệu được sử dụng của chức năng Truy xuất thông tin vé

| STT | Tên bảng/ Cấu trúc dữ liệu | Phương thức |     |     |          |
|-----|----------------------------|-------------|-----|-----|----------|
|     |                            | Thêm        | Sửa | Xóa | Truy vấn |
| 1   | Ticket                     |             |     |     | x        |

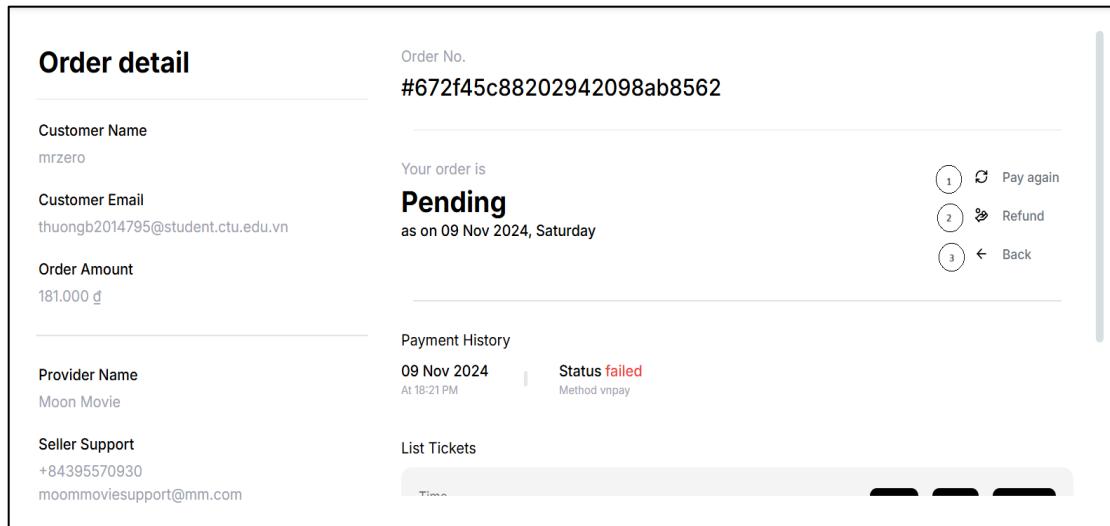
Luồng xử lý:

- Khi người dùng vào trang Tickets
- Hệ thống gửi yêu cầu lấy thông tin vé về cho server
- Server phản hồi lại cho người dùng:
  - o Nếu không có vé nào hiển thị thông báo
  - o Ngược lại hiển thị danh sách vé cho người dùng.

### 2.5.3.3. Thanh toán lại

Mục đích: khi người dùng đang thực hiện một order mà bị gián đoạn thì họ vẫn có thể thanh toán lại để hoàn thành order của mình mà không cần thực hiện lại quá trình đặt vé.

Giao diện:



Hình 2-35: Giao diện Chi tiết hóa đơn

Các thành phần trong giao diện:

Bảng 2-50: Thành phần giao diện chức năng Thanh toán lại

| STT | Loại điều khiển | Mô tả               |
|-----|-----------------|---------------------|
| 1   | Button          | Thanh toán lại      |
| 2   | Button          | Hoàn tiền           |
| 3   | Button          | Trở lại trang trước |

Dữ liệu được sử dụng:

Bảng 2-51: Dữ liệu được sử dụng của chức năng Thanh toán lại

| STT | Tên bảng/ Cấu trúc dữ liệu | Phương thức |     |     |          |
|-----|----------------------------|-------------|-----|-----|----------|
|     |                            | Thêm        | Sửa | Xóa | Truy vấn |
| 1   | Order                      |             |     |     | x        |
| 2   | Payment                    | x           |     |     | x        |
| 3   | Ticket                     |             |     |     | x        |

Luồng xử lý:

- Khi người dùng vào trang **Tickets** có thể truy xuất trực tiếp vào chi tiết đơn hàng đó.
- Hệ thống gửi yêu cầu lấy thông tin về chi tiết đơn hàng về server
- Server phản hồi lại cho người dùng:
  - o Hiển thị chi tiết đơn hàng.
- Người dùng ấn vào nút thanh toán lại.
- Nhập thông tin email và chọn phương thức thanh toán lại.
- Hệ thống gửi yêu cầu thanh toán với phương thức đã chọn.
- Người dùng tiến hành thanh toán lại:
  - o Nếu thanh toán thành công chuyển về trang **Tickets** và gửi thông tin vé về email.
  - o Nếu thanh toán thất bại thì có thể tiếp tục thực hiện lại quá trình trên.

#### **2.5.4. Seat Service**

##### **2.5.4.1. Tổng quan**

Mục tiêu: Seat service là service phục vụ các tác vụ liên quan đến việc quản lý các rạp chiếu phim và những ghế ngồi ở mỗi suất chiếu phim.

Các chức năng chính:

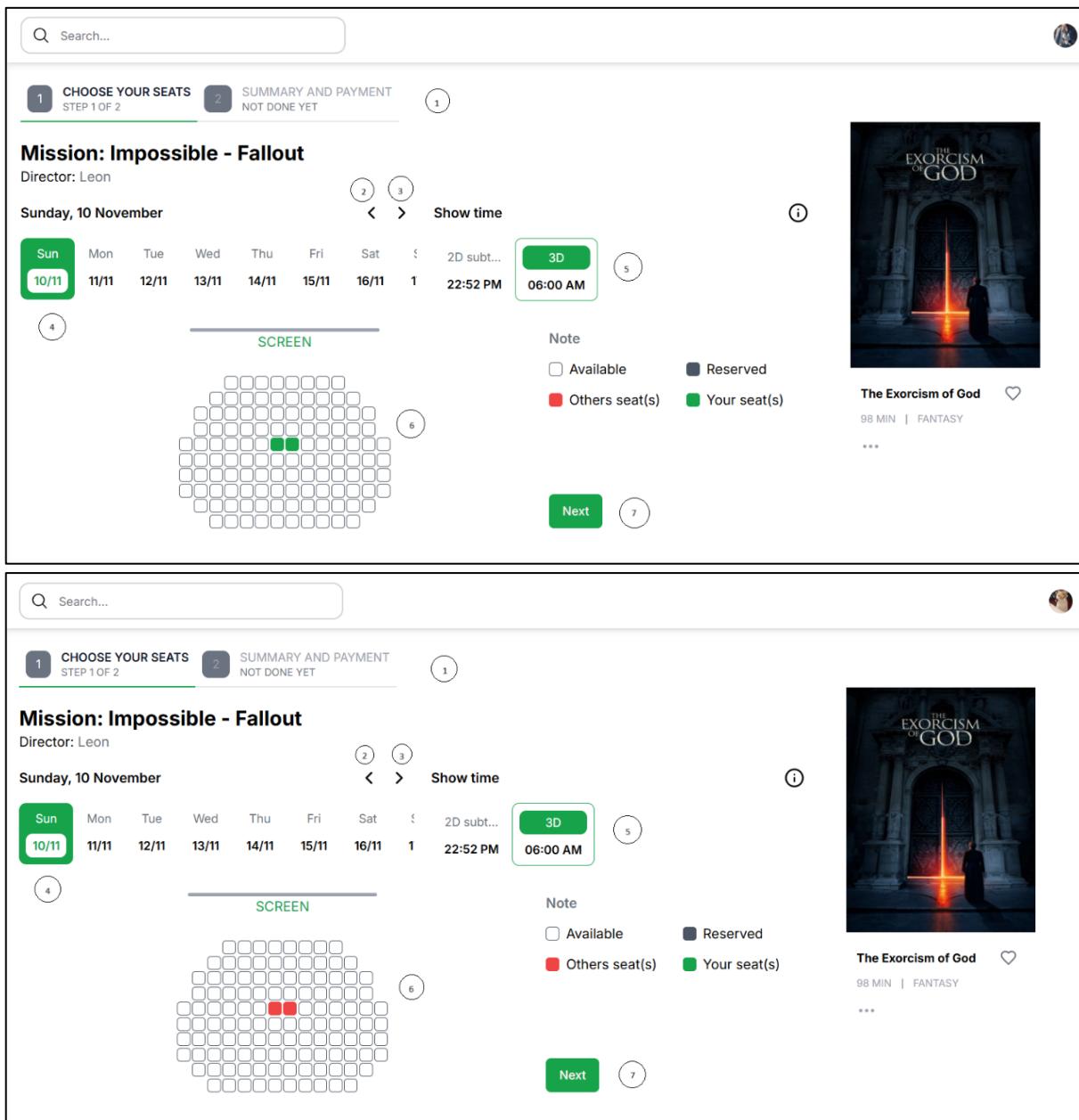
- Truy xuất danh sách ghế cho một suất chiếu ở một rạp phim nhất định.
- Chọn ghế thời gian thực giữa các khách hàng.
- Chính sửa thông tin một rạp chiếu phim.

##### **2.5.4.2. Chọn ghế thời gian thực**

Mục đích: cho phép các người dùng chọn và theo dõi danh sách ghế theo thời gian thực cập nhật liên tục để tránh các trường hợp mua vé thất bại do ghế đã có người chọn mua trong quá trình thực hiện thanh toán của khách hàng.

Giao diện:

## Hệ thống bán vé xem phim trực tuyến



Hình 2-36: Giao diện Chọn ghế

Các thành phần trong giao diện:

Bảng 2-52: Thành phần giao diện chức năng Chọn ghế thời gian thực

| STT | Loại điều khiển | Mô tả           |
|-----|-----------------|-----------------|
| 1   | Button          | Chọn bước       |
| 2   | Button          | Scroll qua trái |
| 3   | Button          | Scroll qua phải |

|   |        |                 |
|---|--------|-----------------|
| 4 | Button | Chọn ngày       |
| 5 | Button | Chọn suất chiếu |
| 6 | Button | Chọn ghế.       |
| 7 | Button | Bước tiếp theo. |

Dữ liệu được sử dụng:

Bảng 2-53: Dữ liệu được sử dụng của chức năng Chọn ghế thời gian thực

| STT | Tên bảng/ Cấu trúc dữ liệu | Phương thức |     |     |          |
|-----|----------------------------|-------------|-----|-----|----------|
|     |                            | Thêm        | Sửa | Xóa | Truy vấn |
| 1   | SeatDetail                 |             | x   |     | x        |
| 2   | Showing                    |             |     |     | x        |
| 3   | Movie                      |             |     |     | x        |

Luồng xử lý:

- Khi người dùng tiến hành đặt vé. Hệ thống sẽ tạo một kết nối trực tiếp đến server thông qua websocket. Kênh websocket này phụ trách trao đổi dữ liệu chọn ghế giữa các người dùng.
- Hệ thống gửi yêu cầu để thấy thông tin của suất chiếu mà người dùng đang chọn.
- Server trả về kết quả:
  - o Giao diện hiển thị danh sách ghế cũng như các thông tin khác của suất chiếu này.
- Khi người dùng tiến hành chọn một ghế. Thay vì dùng http, hệ thống sử dụng websocket để gửi dữ liệu về server.
- Server sau khi xử lý yêu cầu chọn ghế thì phát nó lên kênh socket mà các user khác cùng đăng ký khi họ đang đặt vé cùng một suất chiếu.
- Khi nhận thông tin có ghế đang được chọn giao diện cập nhật lại ngay lập tức, Ngoài ra để tối ưu tốc độ thay vì load lại toàn bộ danh sách ghế. Hệ thống nhận diện ghế có người chọn và chỉ cập nhật tại ghế đó, đảm bảo hiệu suất hoạt động của hệ thống.

### 2.5.5. Recommend Service

#### 2.5.5.1. Tổng quan [11]

Mục tiêu: Recommend service là service phục vụ các tác vụ liên quan đến việc tăng trải nghiệm người dùng. Nhiệm vụ chính của service là phân tích và xử lý tên của các bộ phim để tạo một hệ thống gợi ý các từ khóa dựa trên từ mà người dùng gõ vào

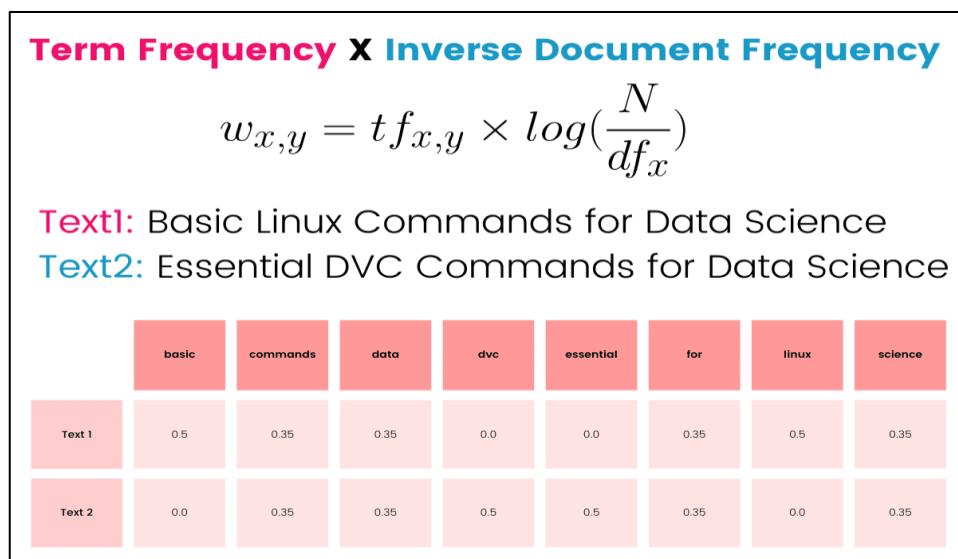
thanh tìm kiếm. Ngoài ra nó cũng dựa vào đó và cung cấp các bộ phim có tính tương tự cho người dùng.

Công nghệ sử dụng: Recommend service sử dụng 2 công nghệ chính là TF-IDF (Term Frequency-Inverse Document Frequency) và Cosine Similarity để tạo các vector và so sánh để đưa ra các gợi ý có điểm gần giống nhau nhất, đảm bảo tính đơn giản nhưng cũng mang lại một phần hiệu quả giúp tăng đáng kể trải nghiệm của người sử dụng.

Bài toán: Khi người dùng gõ các từ khóa vào thanh tìm kiếm, mục tiêu là gợi ý tên các bộ phim có trong cơ sở dữ liệu hệ thống nhưng phải có liên quan đến từ khóa đó của người dùng.

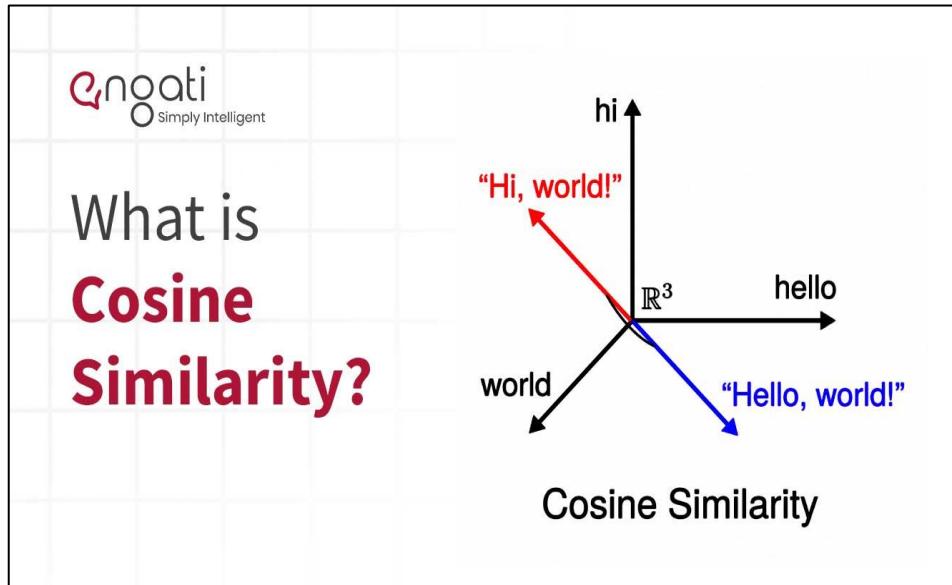
Sử dụng **TF-IDF** để biểu diễn **Phim** và **Từ khóa**: TF-IDF là một cách để thể hiện mỗi tên phim bằng các con số (vector) dựa trên tần suất các từ trong tên phim đó.

- TF (Term Frequency) là tần suất xuất hiện của một từ trong tên phim. Ví dụ, nếu từ "love" xuất hiện nhiều lần trong tên phim, TF của từ "love" sẽ cao hơn trong tên phim đó.
- IDF (Inverse Document Frequency) là giá trị giảm bớt độ quan trọng của các từ phổ biến (như "the", "a" hay "movie") và làm nổi bật các từ ít phổ biến hơn nhưng có tính đặc trưng cao. Ví dụ, từ "love" có thể xuất hiện trong rất nhiều tên phim, nhưng từ "wizard" sẽ ít phổ biến hơn và có tính đặc trưng cao hơn cho các phim về phép thuật.



Hình 2-37: TF-IDF (Term Frequency-Inverse Document Frequency)

Dùng **Cosine Similarity** để **Tìm Độ Tương Đồng**: Sau khi biểu diễn các tên phim và từ khóa bằng vector TF-IDF, ta cần cách so sánh các vector này để xem chúng tương đồng ra sao. Cosine Similarity đơn giản một phép đo độ tương tự về góc giữa hai vector, tương tự như cách so sánh hướng của hai mũi tên. Nếu hướng của hai vector giống nhau (càng gần nhau), thì tên phim càng phù hợp với từ khóa người dùng đã nhập.



Hình 2-38: Cosine Similarity

Các chức năng chính:

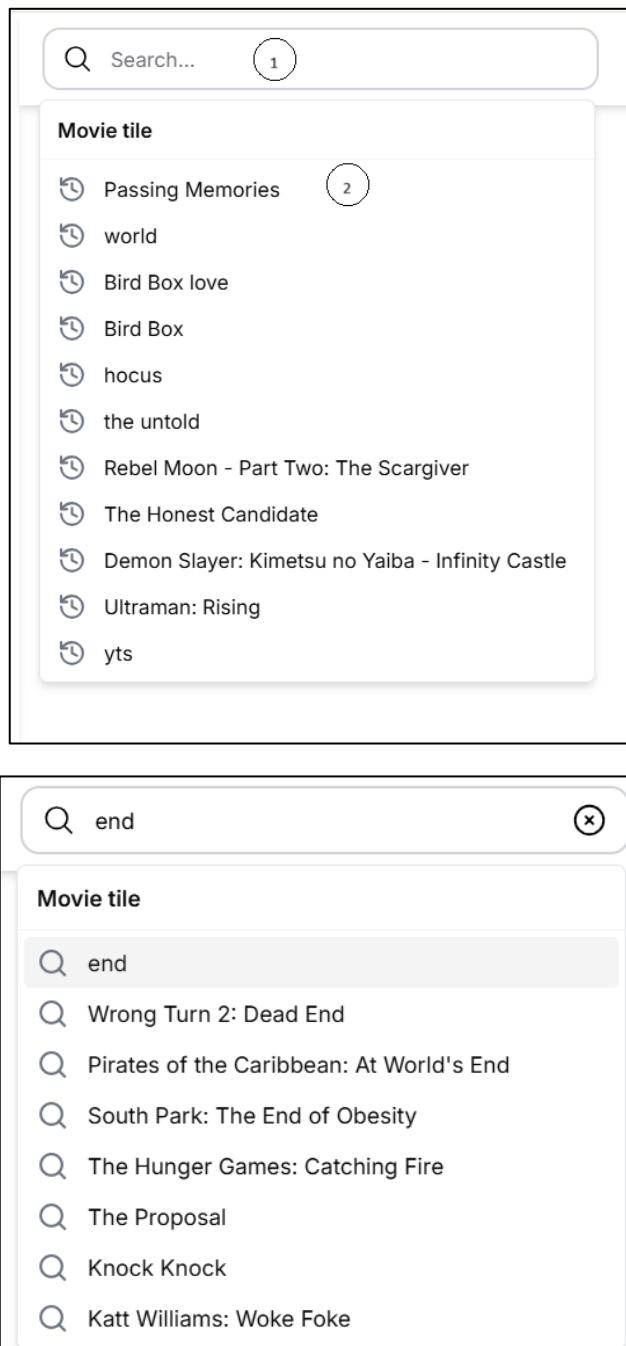
- Gợi ý từ khóa tìm kiếm cho người dùng.
- Gợi ý các bộ phim liên quan cho người dùng.
- Lưu và cập nhật lịch sử các từ khóa tìm kiếm của người dùng.

#### 2.5.5.2. *Gợi ý từ khóa tìm kiếm*

Mục đích: giúp người dùng có thể nhanh chóng tìm kiếm tên bộ phim yêu thích mà không cần nhớ toàn bộ tên, ngoài ra người dùng có thể khám phá thêm các tên khác có liên quan.

Giao diện:

## Hệ thống bán vé xem phim trực tuyến



Hình 2-39: Giao diện Thanh tìm kiếm

Các thành phần trong giao diện:

Bảng 2-54: Thành phần giao diện chức năng Gợi ý từ khóa tìm kiếm

| STT | Loại điều khiển | Mô tả          |
|-----|-----------------|----------------|
| 1   | Textbox         | Thanh tìm kiếm |

|   |        |         |
|---|--------|---------|
| 2 | Button | Kết quả |
|---|--------|---------|

Dữ liệu được sử dụng:

Bảng 2-55: Dữ liệu được sử dụng của chức năng Gợi ý từ khóa tìm kiếm

| STT | Tên bảng/ Cấu trúc dữ liệu | Phương thức |     |     |          |
|-----|----------------------------|-------------|-----|-----|----------|
|     |                            | Thêm        | Sửa | Xóa | Truy vấn |
| 1   | Keyword                    |             |     |     | x        |

Luồng xử lý:

- Khi server khởi động nó sẽ truy cập cơ sở dữ liệu của Movie Service để lấy các thông tin bộ phim liên quan từ đó tạo các vector phụ vụ cho việc gợi ý.
- Khi người dùng gõ từ khóa.
- Hệ thống gửi yêu cầu về cho server.
- Server tiến hành vector hóa từ khóa mà user đang gõ sau đó so sánh và lấy danh sách các **tên phim** có liên quan nhất để gợi ý cho người dùng.
- Đồng thời hệ thống cũng lọc các kết quả tìm kiếm từ lịch sử tìm kiếm của user.
- Server trả về kết quả:
  - o Giao diện cập nhật lại các kết quả mà server trả về.
- Nếu người dùng tiếp tục gõ thêm từ khóa thì thực hiện lại quá trình trên.

### 2.5.6. Gateway Service

Mục tiêu: Gateway service là service phục vụ các tác vụ liên quan đến việc điều hướng các yêu cầu đến hệ thống. Tất cả các yêu cầu sẽ được thông qua Gateway Service, sau đó các yêu cầu sẽ được chuyển đến các service phù hợp để xử lý các tác vụ. Ngoài ra gateway còn phục trách việc xác thực người dùng, ở đây sử dụng JWT. Đối với các yêu cầu và xử lý liên quan đến các tài nguyên cá nhân các yêu cầu cần thông qua xác thực để có thể tiếp tục quá trình. Chi tiết đã được đề cập ở mục [3.1](#).

Các chức năng chính:

- Chuyển hướng các yêu cầu để các service phù hợp để xử lý sau đó nhận và trả kết quả.
- Xác thực các yêu cầu nếu đó không phải là các yêu cầu public.

### 2.5.7. Media Service

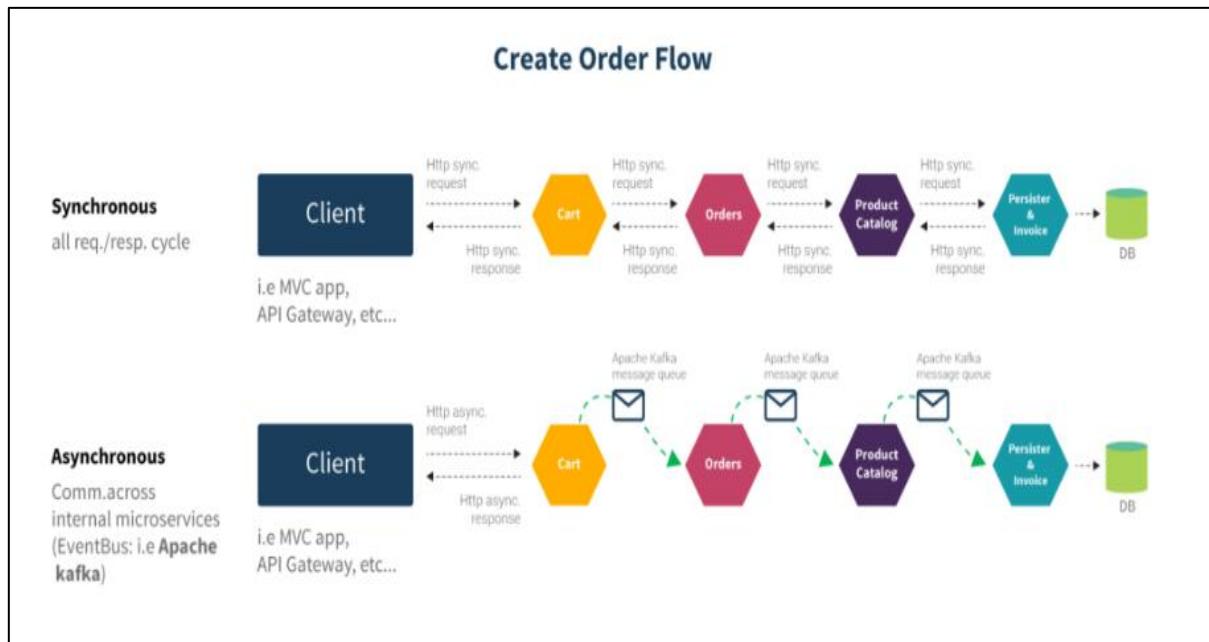
Mục tiêu: Media service là service chuyên xử lý các tác vụ liên quan đến việc lưu trữ và truy xuất các tài nguyên như hình ảnh, video, file...

Các chức năng chính:

- Lưu và xóa các tài nguyên media như hình ảnh và video.
- Truy xuất các tài nguyên media thông qua url

## 2.6. Giao tiếp giữa các services

Trong kiến trúc Microservices, việc giao tiếp giữa các dịch vụ là một phần quan trọng để đảm bảo tính nhất quán và hoạt động hiệu quả của hệ thống. Dưới đây là hai cách thức giao tiếp giữa các dịch vụ trong kiến trúc Microservices sẽ được sử dụng trong hệ thống:



Hình 2-40: Giao tiếp giữa các services

- Giao tiếp đồng bộ: các services tham gia gửi và nhận dữ liệu phải chờ phản hồi từ Service còn lại. Một cách thức điển hình của loại hình giao tiếp này là thông qua HTTP. Khi một yêu cầu gửi đến nó sẽ chờ một service nhận xử lý và trả lại kết quả mới có thể thực hiện tiếp các tác vụ khác.
- Giao tiếp bất đồng bộ: các services tham gia gửi và nhận dữ liệu mà không cần đợi phản hồi từ services còn lại. Giao tiếp bất đồng bộ giữa các services thường được thực hiện với sự giúp đỡ của một hệ thống Message Queue như là một server để lưu trữ và chuyển tiếp các thông điệp giữa các microservices. Các hệ thống thường dùng cho việc này thường là Kafka và RabbitMQ.

### Chương 3. ĐÁNH GIÁ VÀ KIỂM THỬ

Chương này sẽ trình bày quy trình và đánh giá kết quả kiểm thử đối với một số chức năng cơ bản của hệ thống

#### 3.1. Giới thiệu

##### 3.1.1. Mục tiêu kiểm thử

- Nhằm kiểm soát và phát hiện các lỗi tiềm ẩn có trong hệ thống.
- Kiểm tra các tính năng có hoạt động đúng với các yêu cầu đặt ra hay không
- Liệt kê các kết quả có được sau khi kiểm thử.
- Làm tài liệu cho giai đoạn bảo trì

##### 3.1.2. Phạm vi kiểm thử

Quy trình kiểm thử được thực hiện qua các công đoạn:

- Kiểm thử thiết kế: kiểm tra giao diện thiết kế có đúng với đặc tả.
- Kiểm thử chấp nhận: kiểm thử chức năng hệ thống có hoạt động và đáp ứng đúng yêu cầu.
- Kiểm thử chức năng: kiểm thử chức năng có xử lý đúng dữ liệu.
- Kiểm thử cài đặt: tìm và sửa các lỗi xảy ra khi kiểm thử.

#### 3.2. Chi tiết kế hoạch kiểm thử

##### 3.2.1. Các chức năng được kiểm thử

- Xác thực người dùng
- Đặt vé
- Cập nhật thông tin người dùng
- Thanh toán lại
- Bình luận
- Sắp lịch
- Thêm mới bộ phim
- Giao tiếp giữa các services

##### 3.2.2. Cách tiếp cận

Với mỗi tính năng chính hay các nhóm tính năng sẽ được kiểm thử theo thứ tự từ trên xuống dưới và từ trái qua phải để đảm bảo rằng sẽ kiểm thử không bỏ sót chức năng cần kiểm thử.

### 3.2.3. Các tiêu chí kiểm thử thành công/ thất bại

Tiêu chuẩn kiểm thử thành công là kết quả thực hiện chức năng đúng với mong đợi, phù hợp với đặc tả yêu cầu.

Tiêu chuẩn kiểm thử thất bại là kết quả không như mong đợi, xuất hiện lỗi, không phù hợp với các yêu cầu đặc tả.

## 3.3. Quản lý kiểm thử

### 3.3.1. Kế hoạch kiểm thử

- Lập kế hoạch kiểm thử.
- Tạo các test-case.
- Tiến hành kiểm thử.
- Ghi nhận kết quả.

### 3.3.2. Môi trường

Phần cứng:

- Vi xử lý: Intel core i5 10300H.
- Ram: 24GB.
- SSD: 1TB.

Phần mềm:

- Hệ điều hành Win 11 Home.
- Postman.
- Trình duyệt Chrome.
- IntelliJ và Vs code.

### 3.3.3. Trách nhiệm và quyền hạn

Bảng 3-1: Trách nhiệm và quyền hạn kiểm thử

| Họ và tên         | Công việc |          |          |           |            |          |
|-------------------|-----------|----------|----------|-----------|------------|----------|
|                   | Quản lý   | Thiết kế | Chuẩn bị | Thực hiện | Chứng kiến | Kiểm tra |
| Phan Thanh Thường | X         | X        | X        | X         | X          | X        |

### 3.4. Kịch bản kiểm thử

#### 3.4.1. Kiểm thử và đánh giá hiệu suất và độ chính xác của chức năng gợi ý từ khóa [14]

Dữ liệu: kiểm thử dựa trên tập dữ liệu có khoảng 8430 bản ghi.

Nội dung kiểm thử: thử gõ và tìm kiếm các từ khóa trực tiếp trên giao diện của người dùng để kiểm tra độ chính xác của các từ khóa gợi ý, hiệu quả xử lý và trải nghiệm người dùng. Kết quả sẽ đạt được sẽ được đề cập chi tiết ở phần dưới đây.

##### 3.4.1.1. Độ chính xác (Accuracy Metrics):

Được tính toán theo công thức sau:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Với:

- **Precision:** Đánh giá tỷ lệ giữa số kết quả gợi ý đúng (relevant) và tổng số kết quả được gợi ý. Ví dụ trong top 10 kết quả gợi ý, có 7 kết quả có liên quan => Precision = 7/10.
- **Recall:** Đánh giá tỷ lệ giữa số kết quả gợi ý đúng và tổng số kết quả liên quan trong cơ sở dữ liệu. Ví dụ: Cơ sở dữ liệu có 12 kết quả liên quan, nhưng chỉ 7 kết quả được gợi ý đúng => Recall = 7/12.

Kết quả:

- **Precision:** Hệ thống đạt Precision trung bình là **75%**, tức là trong top 10 kết quả gợi ý trả về, trung bình có 7-8 kết quả liên quan trực tiếp đến từ khóa người dùng nhập.
- **Recall:** Với Recall đạt **80%**, hệ thống đã gợi ý đúng 8/10 tên phim liên quan trong tổng số kết quả có thể có trong cơ sở dữ liệu.
- **F1-Score:** Chỉ số F1-Score của hệ thống đạt **77%**, thể hiện sự cân bằng tốt giữa Precision và Recall.

##### 3.4.1.2. Hiệu quả xử lý (Efficiency Metrics):

- **Thời gian phản hồi truy vấn (Query Response Time):** Hệ thống trả về kết quả trung bình trong 0.17 giây trên một dữ liệu thử nghiệm khoảng 8430 tên phim, đảm bảo thời gian gần như ngay lập tức cho người dùng.

- **Tài nguyên sử dụng (Memory Usage):** Bộ nhớ tiêu thụ trung bình là **75MB** trong quá trình tạo vector TF-IDF và tính toán Cosine Similarity cho tập dữ liệu thử nghiệm.

#### **3.4.1.3. Trải nghiệm người dùng (User Experience Metrics):**

- **Mean Reciprocal Rank (MRR):** Giá trị MRR đạt **0.85**, nghĩa là kết quả đúng đầu tiên thường xuyên xuất hiện trong top 1-3 kết quả gợi ý. Tức là người dùng luôn tìm được toàn bộ tên bộ phim họ muốn chỉ với các từ khóa nhập vào.

#### **3.4.1.4. Kết luận tổng quan**

Nhìn chung, phương pháp này đáp ứng tốt cả về tốc độ và độ chính xác với yêu cầu ban đầu đặt ra là nhanh chóng tìm được tên phim mong muốn với các từ khóa tìm kiếm. Trong tương lai có thể tối ưu hóa và tích hợp thêm các hệ thống nâng cao để xử lý ngôn ngữ có ý nghĩa hơn hay là tích hợp thêm các hệ thống học sâu,...

### **3.4.2. Kiểm thử chức năng xác thực người dùng**

#### **3.4.2.1. Chức năng đăng ký**

Mô tả: Kiểm tra quá trình đăng ký người dùng mới.

Kỳ vọng: Người dùng có thể đăng ký mới tài khoản với username và email là duy nhất.

Kịch bản chi tiết:

Bảng 3-2: Kịch bản kiểm thử chi tiết chức năng Đăng ký

| ID       | Mô tả kiểm thử                                 | Dữ liệu đầu vào                    | Bước thực hiện  | Kỳ vọng   | Kết quả kiểm thử  | Đánh giá |
|----------|--|------------------------------------|---|---|---|----------|
| TC_SI_01 | Kiểm tra đăng ký thành công với dữ liệu hợp lệ | Username, email và mật khẩu hợp lệ | 1. Mở trang đăng ký.<br>2. Nhập username, email và mật khẩu hợp lệ.<br>3. Nhấn nút “Đăng ký”. | Người dùng đăng ký thành công, nhận được thông báo và chuyển về trang home. | Người dùng đăng ký thành công, nhận được thông báo và chuyển về trang home. | Đạt      |
| TC_SI_02 | Kiểm tra lỗi khi                               | Username hợp lệ, email đã tồn tại  | 1. Mở trang đăng ký.  | Hiển thị lỗi “Email đã tồn tại”   | Hiển thị lỗi “Email đã tồn tại”   | Đạt      |

| ID       | Mô tả kiểm thử                         | Dữ liệu đầu vào   | Bước thực hiện  | Kỳ vọng  | Kết quả kiểm thử   | Đánh giá |
|----------|--|---|---|--|--|----------|
|          | email đã tồn tại                       | tồn tại và mật khẩu hợp lệ.                             | 2. Nhập username, email đã tồn tại và mật khẩu hợp lệ.<br>3. Nhấn nút “Đăng ký”.                                  |  |  |          |
| TC_SI_03 | Kiểm tra lỗi khi email không hợp lệ    | Username hợp lệ, email không hợp lệ và mật khẩu hợp lệ. | 1. Mở trang đăng ký.<br>2. Nhập username, email không hợp lệ và mật khẩu hợp lệ.<br>3. Nhấn nút “Đăng ký”.        | Hiển thị lỗi “Email không hợp lệ”  | Hiển thị lỗi “Email đã tồn tại”  | Đạt      |
| TC_SI_04 | Kiểm tra lỗi khi username đã tồn tại   | Username đã tồn tại, email hợp lệ và mật khẩu hợp lệ.   | 1. Mở trang đăng ký.<br>2. Nhập username đã tồn tại, email hợp lệ và mật khẩu hợp lệ.<br>3. Nhấn nút “Đăng ký”.   | Hiển thị lỗi “Username đã tồn tại”   | Hiển thị lỗi “Username đã tồn tại”   | Đạt      |
| TC_SI_05 | Kiểm tra lỗi khi username không hợp lệ | Username không hợp lệ, email hợp lệ và mật khẩu hợp lệ. | 1. Mở trang đăng ký.<br>2. Nhập username không hợp lệ, email hợp lệ và mật khẩu hợp lệ.<br>3. Nhấn nút “Đăng ký”. | Hiển thị lỗi “Username phải chứa ít nhất 3 ký tự” hoặc “Username phải chứa tối đa 30 ký tự” tùy trường hợp | Hiển thị lỗi “Username phải chứa ít nhất 3 ký tự” hoặc “Username phải chứa tối đa 30 ký tự” tùy trường hợp | Đạt      |

| ID       | Mô tả kiểm thử                         | Dữ liệu đầu vào   | Bước thực hiện  | Kỳ vọng  | Kết quả kiểm thử   | Đánh giá |
|----------|--|---|---|--|--|----------|
| TC_SI_06 | Kiểm tra lỗi khi mật khẩu không hợp lệ | Username hợp lệ, email hợp lệ và mật khẩu không hợp lệ. | 1. Mở trang đăng ký.<br>2. Nhập username hợp lệ, email hợp lệ và mật khẩu không hợp lệ.<br>3. Nhấn nút “Đăng ký”. | Hiển thị lỗi “Mật khẩu phải chứa ít nhất một chữ cái viết hoa, một chữ cái viết thường và một chữ số.” | Hiển thị lỗi “Mật khẩu phải chứa ít nhất một chữ cái viết hoa, một chữ cái viết thường và một chữ số.” | Đạt      |
| TC_SI_07 | Kiểm tra các trường bị bỏ trống        | Không nhập username, email và mật khẩu.                 | 1. Mở trang đăng ký.<br>2. Không nhập username, email và mật khẩu.<br>3. Nhấn nút “Đăng ký”.                      | Hiển thị thông báo yêu cầu điền đầy đủ các trường bắt buộc   | Hiển thị thông báo yêu cầu điền đầy đủ các trường bắt buộc   | Đạt      |

### 3.4.2.2. *Chức năng đăng nhập*

Mô tả: Kiểm tra quá trình đăng nhập tài khoản của người dùng.

Kỳ vọng: Người dùng có thể đăng nhập với tài khoản đã đăng ký trước đó.

Kịch bản chi tiết:

Bảng 3-3: Kịch bản kiểm thử chi tiết chức năng Đăng nhập

| ID       | Mô tả kiểm thử                                    | Dữ liệu đầu vào                                     | Bước thực hiện  | Kỳ vọng   | Kết quả kiểm thử   | Đánh giá |
|----------|---|---|---|---|--|----------|
| TC_SU_01 | Kiểm tra đăng nhập thành công với dữ liệu hợp lệ. | Username hoặc email chính xác và mật khẩu chính xác | 1. Mở trang đăng nhập.<br>2. Nhập username hoặc email hợp lệ và | Người dùng đăng nhập thành công, nhận được thông báo và chuyển về trang home. | Người dùng đăng nhập thành công, nhận được thông báo và chuyển | Đạt      |

| ID          | Mô tả kiểm thử                                       | Dữ liệu đầu vào   | Bước thực hiện   | Kỳ vọng  | Kết quả kiểm thử   | Đánh giá |
|-------------|--|---|--|--|--|----------|
|             |  |   | mật khẩu hợp lệ.<br>3. Nhấn nút “Đăng nhập”.   |  | về trang home.   |          |
| TC _ SU _02 | Kiểm tra lỗi khi username hoặc email không chính xác | Username hoặc email không chính xác và mật khẩu chính xác | 1. Mở trang đăng nhập.<br>2. Nhập username hoặc email không chính xác và mật khẩu chính xác.<br>3. Nhấn nút “Đăng nhập”. | Hiển thị lỗi “Username hoặc email hoặc là mật khẩu không đúng”.  | Hiển thị lỗi “Username hoặc email hoặc là mật khẩu không đúng”.  | Đạt      |
| TC _ SU _03 | Kiểm tra lỗi khi mật khẩu không chính xác            | Username hoặc email chính xác và mật khẩu không chính xác | 1. Mở trang đăng nhập.<br>2. Nhập username hoặc email chính xác và mật khẩu không chính xác.<br>3. Nhấn nút “Đăng nhập”. | Hiển thị lỗi “Username hoặc email hoặc là mật khẩu không đúng”.  | Hiển thị lỗi “Username hoặc email hoặc là mật khẩu không đúng”.  | Đạt      |
| TC _ SU _04 | Kiểm tra lỗi khi username hoặc email không hợp lệ    | Username hoặc không hợp lệ và mật khẩu hợp lệ.            | 1. Mở trang đăng nhập.<br>2. Nhập username hoặc email không lệ, và mật khẩu hợp lệ.                                      | Hiển thị lỗi “Username phải chứa ít nhất 3 ký tự” hoặc “Username phải chứa tối đa 30 ký tự” tùy trường hợp | Hiển thị lỗi “Username phải chứa ít nhất 3 ký tự” hoặc “Username phải chứa tối đa 30 ký tự” tùy trường hợp | Đạt      |

| ID          | Mô tả kiểm thử                         | Dữ liệu đầu vào   | Bước thực hiện   | Kỳ vọng  | Kết quả kiểm thử   | Đánh giá |
|-------------|--|---|--|--|--|----------|
|             |  |   | 3. Nhấn nút “Đăng nhập”.   |  |  |          |
| TC _ SU _05 | Kiểm tra lỗi khi mật khẩu không hợp lệ | Username hợp lệ hoặc email hợp lệ và mật khẩu không hợp lệ. | 1. Mở trang đăng nhập.<br>2. Nhập username hoặc email hợp lệ và mật khẩu không hợp lệ.<br>3. Nhấn nút “Đăng nhập”. | Hiển thị lỗi “Mật khẩu phải chứa ít nhất một chữ cái viết hoa, một chữ cái viết thường và một chữ số.” | Hiển thị lỗi “Mật khẩu phải chứa ít nhất một chữ cái viết hoa, một chữ cái viết thường và một chữ số.” | Đạt      |
| TC _ SU _06 | Kiểm tra các trường bị bỏ trống        | Không nhập username, email và mật khẩu.                     | 1. Mở trang đăng nhập.<br>2. Không nhập username, email và mật khẩu.<br>3. Nhấn nút “Đăng nhập”.                   | Hiển thị thông báo yêu cầu điền đầy đủ các trường bắt buộc   | Hiển thị thông báo yêu cầu điền đầy đủ các trường bắt buộc   | Đạt      |

### 3.4.2.3. Chức năng đăng nhập với Google

Mô tả: Kiểm tra quá trình đăng nhập của người dùng với tài khoản google.

Kỳ vọng: Người dùng có thể đăng nhập với tài khoản google.

Kịch bản chi tiết:

Bảng 3-4: Kịch bản kiểm thử chi tiết chức năng Đăng nhập với Google

| ID     | Mô tả kiểm thử     | Dữ liệu đầu vào | Bước thực hiện      | Kỳ vọng              | Kết quả kiểm thử     | Đánh giá |
|--------|--------------------|-----------------|---------------------|----------------------|----------------------|----------|
| TC SUG | Kiểm tra đăng nhập |                 | 1. Mở trang đăng ký | Người dùng đăng nhập | Người dùng đăng nhập | Đạt      |

| ID        | Mô tả kiểm thử                                 | Dữ liệu đầu vào | Bước thực hiện   | Kỳ vọng  | Kết quả kiểm thử   | Đánh giá |
|-----------|--|-----------------|--|--|--|----------|
| _01       | với google lần đầu tiên                        |                 | hoặc đăng nhập.<br>2. Nhấn nút “Đăng nhập với google”.                     | thành công và chuyển sang trang điện username.             | thành công và chuyển sang trang điện username.             |          |
| TC_SUG_02 | Kiểm tra đăng nhập với google sau lần đầu tiên |                 | 1. Mở trang đăng ký hoặc đăng nhập.<br>2. Nhấn nút “Đăng nhập với google”. | Người dùng đăng nhập thành công và chuyển sang trang home. | Người dùng đăng nhập thành công và chuyển sang trang home. | Đạt      |

### 3.4.3. Kiểm thử chức năng đặt vé

Mô tả: Kiểm tra quá trình đặt vé của người dùng.

Kỳ vọng: Người dùng có thể đặt vé thành công.

Kịch bản chi tiết:

Bảng 3-5: Kịch bản kiểm thử chi tiết chức năng Đặt vé

| ID       | Mô tả kiểm thử                                  | Dữ liệu đầu vào                                | Bước thực hiện   | Kỳ vọng  | Kết quả kiểm thử   | Đánh giá |
|----------|---|--|--|--|--|----------|
| TC_BT_01 | Kiểm tra đặt vé thành công với thông tin hợp lệ | Suất chiếu có ghế trống, thanh toán thành công | 1. Chọn suất chiếu.<br>2. Chọn ghế trống.<br>3. Tiến hành thanh toán.          | Vé được đặt thành công, chuyển về trang Tickets, gửi vé về email người dùng. | Vé được đặt thành công, chuyển về trang Tickets, gửi vé về email người dùng. | Đạt      |
| TC_BT_02 | Kiểm tra đặt vé khi thanh toán thất bại         | Suất chiếu có ghế trống, thanh toán thất bại   | 1. Chọn suất chiếu.<br>2. Chọn ghế trống.<br>3. Tiến hành thanh toán thất bại. | Người dùng được chuyển sang trang thanh toán thất bại.                       | Người dùng được chuyển sang trang thanh toán thất bại.                       | Đạt      |

| ID       | Mô tả kiểm thử  | Dữ liệu đầu vào               | Bước thực hiện                                | Kỳ vọng  | Kết quả kiểm thử   | Đánh giá |
|----------|---|-------------------------------|---|--|--|----------|
| TC_BT_03 | Kiểm tra đặt vé khi không chọn ghế                    | Suất chiếu có ghế trống       | 1. Chọn suất chiếu.<br>2. Không chọn ghế      | Hiển thị thông báo “Vui lòng chọn ghế trước khi sang bước tiếp theo” | Hiển thị thông báo “Vui lòng chọn ghế trước khi sang bước tiếp theo” | Đạt      |
| TC_BT_04 | Kiểm tra hiển thị tổng giá vé đúng khi chọn nhiều ghế | Suất chiếu có nhiều ghế trống | 1. Chọn suất chiếu.<br>2. Chọn ghế nhiều ghế. | Hiển thị tổng giá vé theo số ghế đã chọn                             | Hiển thị tổng giá vé theo số ghế đã chọn                             | Đạt      |

#### 3.4.4. Kiểm thử chức năng cập nhật thông tin người dùng

Mô tả: Kiểm tra quá trình thay đổi thông tin cá nhân của người dùng.

Kỳ vọng: Người dùng có thể quản lý và thay đổi các thông tin cơ bản của mình.

Kịch bản chi tiết:

Bảng 3-6: Kịch bản kiểm thử chi tiết chức năng Update thông tin user

| ID        | Mô tả kiểm thử   | Dữ liệu đầu vào                              | Bước thực hiện  | Kỳ vọng  | Kết quả kiểm thử   | Đánh giá |
|-----------|--|--|---|--|--|----------|
| TC_MUI_01 | Kiểm tra cập nhật thông tin user khi các username, email, name, avatar và bio hợp lệ | Username, email, name, avatar và bio hợp lệ. | 1. Mở trang profile.<br>2. Điều chỉnh username, email, name, avatar và bio hợp lệ<br>3. Nhấn nút “Lưu”. | Người dùng cập nhật thông tin thành công, nhận được thông báo. | Người dùng cập nhật thông tin thành công, nhận được thông báo. | Đạt      |
| TC_MUI_02 | Kiểm tra lỗi khi   | Email đã tồn tại và các trường               | 1. Mở trang profile.  | Hiển thị lỗi “Email đã tồn tại”                                | Hiển thị lỗi “Email đã tồn tại”                                | Đạt      |

| ID        | Mô tả kiểm thử                           | Dữ liệu đầu vào                                   | Bước thực hiện   | Kỳ vọng  | Kết quả kiểm thử   | Đánh giá |
|-----------|--|---|--|--|--|----------|
|           | email đã tồn tại                         | còn lại hợp lệ                                    | 2. Điều chỉnh email đã tồn tại, và các trường còn lại hợp lệ<br>3. Nhấn nút “Lưu”.                               |  |  |          |
| TC_MUI_03 | Kiểm tra lỗi khi username đã tồn tại     | Username đã tồn tại, và các trường còn lại hợp lệ | 1. Mở trang profile.<br>2. Điều chỉnh username đã tồn tại, và các trường còn lại hợp lệ<br>3. Nhấn nút “Lưu”.    | Hiển thị lỗi “Username đã tồn tại”                         | Hiển thị lỗi “Username đã tồn tại”                         | Đạt      |
| TC_MUI_04 | Kiểm tra lỗi khi các trường không hợp lệ | Một trường không hợp lệ và các trường khác hợp lệ | 1. Mở trang profile.<br>2. Điều chỉnh một trường không hợp lệ và các trường còn lại hợp lệ<br>3. Nhấn nút “Lưu”. | Hiển thị lỗi tương ứng với trường không hợp lệ             | Hiển thị lỗi tương ứng với trường không hợp lệ             | Đạt      |
| TC_MUI_05 | Kiểm tra các trường bị bỏ trống          | Không nhập tất cả các trường                      | 1. Mở trang đăng ký.<br>2. Không nhập tất cả các trường<br>3. Nhấn nút “Đăng ký”.                                | Hiển thị thông báo yêu cầu điền đầy đủ các trường bắt buộc | Hiển thị thông báo yêu cầu điền đầy đủ các trường bắt buộc | Đạt      |

### 3.4.5. Kiểm thử chức năng thanh toán lại

Mô tả: Kiểm tra quá trình thanh toán lại một đơn hàng thất bại của người dùng

Kỳ vọng: Người dùng có thể thành công thanh toán lại đơn hàng thất bại

Bảng 3-7: Kịch bản kiểm thử chi tiết chức năng Thanh toán lại

| ID       | Mô tả kiểm thử                                | Dữ liệu đầu vào                     | Bước thực hiện   | Kỳ vọng  | Kết quả kiểm thử   | Đánh giá |
|----------|---|-------------------------------------|--|--|--|----------|
| TC_RP_01 | Kiểm tra thanh toán lại thành công            | Tiến hành thanh toán lại thành công | 1. Mở trang “Chi tiết hóa đơn”.<br>2. Nhấn nút “Thanh toán lại”.<br>3. Chọn phương thức thanh toán lại và điền email.<br>4. Tiến hành thanh toán lại thành công. | Người dùng thanh toán lại đơn hàng thành công, chuyển về trang Tickets | Người dùng thanh toán lại đơn hàng thành công, chuyển về trang Tickets | Đạt      |
| TC_RP_01 | Kiểm tra lỗi khi thanh toán tiếp tục thất bại | Tiến hành thanh toán lại thất bại   | 1. Mở trang “Chi tiết hóa đơn”.<br>2. Nhấn nút “Thanh toán lại”.<br>3. Chọn phương thức thanh toán lại và điền email.<br>4. Tiến hành thanh toán lại thất bại.   | Hiển thị thông báo thanh toán thất bại.                                | Hiển thị thông báo thanh toán thất bại.                                | Đạt      |
| TC_RP_03 | Kiểm tra lỗi khi các vé đã có                 | Các vé trong đơn hàng đó đã có      | 1. Mở trang “Chi tiết hóa đơn”.  | Hiển thị lỗi “Một vài vé đã được đặt”                                  | Hiển thị lỗi “Một vài vé đã được đặt”                                  | Đạt      |

| ID       | Mô tả kiểm thử   | Dữ liệu đầu vào                               | Bước thực hiện  | Kỳ vọng   | Kết quả kiểm thử                                    | Đánh giá |
|----------|--|---|---|---|---|----------|
|          | người khác đặt   | người khác đặt                                | 2. Nhấn nút “Thanh toán lại”.<br>3. Chọn phương thức thanh toán lại và điền email.  | bởi người khác”                                     | bởi người khác”                                     |          |
| TC_RP_04 | Kiểm tra lỗi khi không chọn phương thức thanh toán lại | Không chọn phương thức thanh toán             | 1. Mở trang “Chi tiết hóa đơn”.<br>2. Nhấn nút “Thanh toán lại”.<br>3. Không chọn phương thức thanh toán lại và điền email. | Hiển thị lỗi “Vui lòng chọn phương thức thanh toán” | Hiển thị lỗi “Vui lòng chọn phương thức thanh toán” | Đạt      |
| TC_RP_05 | Kiểm tra lỗi khi không để email trống                  | Để email trống và chọn phương thức thanh toán | 1. Mở trang “Chi tiết hóa đơn”.<br>2. Nhấn nút “Thanh toán lại”.<br>3. Chọn phương thức thanh toán lại và để email trống.   | Hiển thị lỗi “Vui lòng điền email”                  | Hiển thị lỗi “Vui lòng điền email”                  | Đạt      |

#### 3.4.6. Kiểm thử chức năng bình luận

Mô tả: Kiểm tra quá trình bình luận cho một bộ phim hoặc là trả lời một bình luận của người dùng khác.

Kỳ vọng: Người dùng có thể thành công bình luận một bộ phim và có thể thành công trả lời một bình luận khác.

Bảng 3-8: Kịch bản kiểm thử chi tiết chức năng Bình luận

| ID        | Mô tả kiểm thử                     | Dữ liệu đầu vào      | Bước thực hiện   | Kỳ vọng   | Kết quả kiểm thử  | Đánh giá |
|-----------|------------------------------------|----------------------|--|---|---|----------|
| TC_CMT_01 | Thêm một bình luận thành công      | Bình luận hợp lệ     | 1. Mở trang “Chi tiết phim”.<br>2. Dièn bình luận<br>3. Nhấn nút “Đăng bình luận”                        | Người dùng bình luận thành công và danh sách bình luận được cập nhật lại.         | Người dùng bình luận thành công và danh sách bình luận được cập nhật lại.         | Đạt      |
| TC_CMT_02 | Trả lời một bình luận thành công   | Bình luận hợp lệ     | 1. Mở trang “Chi tiết phim”.<br>2. Nhấn nút “Trả lời”<br>2. Dièn bình luận<br>3. Nhấn nút “Đăng trả lời” | Người dùng trả lời bình luận thành công và danh sách bình luận được cập nhật lại. | Người dùng trả lời bình luận thành công và danh sách bình luận được cập nhật lại. | Đạt      |
| TC_CMT_03 | Kiểm tra lỗi bình luận bị bỏ trống | Không điền bình luận | 1. Mở trang “Chi tiết phim”.<br>2. Bỏ trống bình luận<br>3. Nhấn nút “Đăng bình luận”                    | Không có gì thay đổi  | Không có gì thay đổi  | Đạt      |

#### 3.4.7. Kiểm thử chức năng sắp lịch

Mô tả: Kiểm tra quá trình sắp các lịch chiếu phim từ vai trò của người quản lý ADMIM

Kỳ vọng: ADMIN có thể tiến hành sắp lịch các bộ phim trong một tháng

Bảng 3-9: Kịch bản kiểm thử chi tiết chức năng Sắp lịch

| ID        | Mô tả kiểm thử                     | Dữ liệu đầu vào      | Bước thực hiện   | Kỳ vọng   | Kết quả kiểm thử  | Đánh giá |
|-----------|------------------------------------|----------------------|--|---|---|----------|
| TC_SDL_01 | Thêm một bình luận thành công      | Bình luận hợp lệ     | 1. Mở trang “Chi tiết phim”.<br>2. Điền bình luận<br>3. Nhấn nút “Đăng bình luận”                        | Người dùng bình luận thành công và danh sách bình luận được cập nhật lại.         | Người dùng bình luận thành công và danh sách bình luận được cập nhật lại.         | Đạt      |
| TC_SDL_01 | Trả lời một bình luận thành công   | Bình luận hợp lệ     | 1. Mở trang “Chi tiết phim”.<br>2. Nhấn nút “Trả lời”<br>2. Điền bình luận<br>3. Nhấn nút “Đăng trả lời” | Người dùng trả lời bình luận thành công và danh sách bình luận được cập nhật lại. | Người dùng trả lời bình luận thành công và danh sách bình luận được cập nhật lại. | Đạt      |
| TC_SDL_01 | Kiểm tra lỗi bình luận bị bỏ trống | Không điền bình luận | 1. Mở trang “Chi tiết phim”.<br>2. Bỏ trống bình luận<br>3. Nhấn nút “Đăng bình luận”                    | Không có gì thay đổi  | Không có gì thay đổi  | Đạt      |

#### 3.4.8. Kiểm thử chức năng thêm mới một bộ phim

Mô tả: Kiểm tra quá trình thêm mới một bộ phim từ vai trò của người quản lý ADMIM

Kỳ vọng: ADMIN có thêm thành công thông tin một bộ phim mới.

Bảng 3-10: Kịch bản kiểm thử chi tiết chức năng Thêm mới bộ phim

| ID       | Mô tả kiểm thử  | Dữ liệu đầu vào  | Bước thực hiện  | Kỳ vọng   | Kết quả kiểm thử  | Đánh giá |
|----------|---|--|---|---|---|----------|
| TC_AM_01 | Kiểm tra thêm một bộ phim thành công với tất cả các trường hợp lệ | Tất cả các trường đều hợp lệ                           | 1. Mở trang “Thêm mới bộ phim”.<br>2. Điền các trường với dữ liệu hợp lệ.<br>3. Nhấn nút “Thêm mới”                           | Hiển thị thông báo thêm thành công và reset form về trạng thái ban đầu                  | Hiển thị thông báo thêm thành công và reset form về trạng thái ban đầu                  | Đạt      |
| TC_AM_02 | Kiểm tra lỗi khi trường images không đủ các ảnh yêu cầu           | Các trường đều hợp lệ trừ trường iamges chỉ chọn 1 ảnh | 1. Mở trang “Thêm mới bộ phim”.<br>2. Điền các trường với dữ liệu hợp lệ trừ trường images.<br>3. Nhấn nút “Thêm mới”         | Hiển thị thông báo lỗi “Bạn cần ít nhất 2 poster” hoặc “Bạn cần ít nhất 2 ảnh backdrop” | Hiển thị thông báo lỗi “Bạn cần ít nhất 2 poster” hoặc “Bạn cần ít nhất 2 ảnh backdrop” | Đạt      |
| TC_AM_03 | Kiểm tra lỗi khi trường images bỏ trống                           | Các trường đều hợp lệ trừ trường iamges bỏ trống       | 1. Mở trang “Thêm mới bộ phim”.<br>2. Điền các trường với dữ liệu hợp lệ, không chọn trường images.<br>3. Nhấn nút “Thêm mới” | Hiển thị thông báo lỗi “Trường này là bắt buộc”   | Hiển thị thông báo lỗi “Trường này là bắt buộc”   | Đạt      |
| TC_AM_03 | Kiểm tra lỗi khi trường images bỏ trống                           | Các trường đều hợp lệ trừ trường iamges bỏ trống       | 1. Mở trang “Thêm mới bộ phim”.<br>2. Điền các trường với dữ liệu hợp lệ, không chọn trường images.                           | Hiển thị thông báo lỗi “Trường này là bắt buộc”   | Hiển thị thông báo lỗi “Trường này là bắt buộc”   | Đạt      |

| ID       | Mô tả kiểm thử   | Dữ liệu đầu vào  | Bước thực hiện   | Kỳ vọng   | Kết quả kiểm thử  | Đánh giá |
|----------|--|--|--|---|---|----------|
|          |  |  | 3. Nhấn nút “Thêm mới”   |   |   |          |
| TC_AM_04 | Kiểm tra lỗi khi thêm một bộ phim vào tháng đã sắp lịch rồi                        | Trường tháng sắp lịch là tháng đã sắp lịch rồi, các trường còn lại hợp lệ. | 1. Mở trang “Thêm mới bộ phim”.<br>2. Dièn trường tháng sắp lịch là tháng đã sắp lịch rồi, các trường còn lại hợp lệ.<br>3. Nhấn nút “Thêm mới”                  | Hiển thị thông báo lỗi “Tháng này đã được sắp lịch rồi”               | Hiển thị thông báo lỗi “Tháng này đã được sắp lịch rồi”               | Đạt      |
| TC_AM_05 | Kiểm tra lỗi khi tên bộ phim đã tồn tại  | Tên bộ phim đã tồn tại, các trường khác hợp lệ                             | 1. Mở trang “Thêm mới bộ phim”.<br>2. Dièn trường tên bộ phim đã tồn tại, các trường còn lại hợp lệ.<br>3. Nhấn nút “Thêm mới”                                   | Hiển thị thông báo lỗi “Bộ phim này đã tồn tại rồi”                   | Hiển thị thông báo lỗi “Bộ phim này đã tồn tại rồi”                   | Đạt      |
| TC_AM_06 | Kiểm tra lỗi số suất chiếu trong tháng nhỏ hơn số suất chiếu của bộ phim muốn thêm | Số suất chiếu nhiều hơn số suất chiếu còn lại trong tháng                  | 1. Mở trang “Thêm mới bộ phim”.<br>2. Dièn số suất chiếu trong nhiều hơn số suất chiếu còn lại trong tháng, các trường còn lại hợp lệ.<br>3. Nhấn nút “Thêm mới” | Hiển thị thông báo lỗi “Tổng số suất chiếu tháng này đã đạt giới hạn” | Hiển thị thông báo lỗi “Tổng số suất chiếu tháng này đã đạt giới hạn” | Đạt      |

| ID       | Mô tả kiểm thử                                 | Dữ liệu đầu vào                | Bước thực hiện  | Kỳ vọng   | Kết quả kiểm thử  | Đánh giá |
|----------|--|--------------------------------|---|---|---|----------|
| TC_AM_07 | Kiểm tra lỗi khi điền các trường không hợp lệ. | Điền các trường không hợp lệ   | 1. Mở trang “Thêm mới bộ phim”.<br>2. Điền các trường không hợp lệ.<br>3. Nhấn nút “Thêm mới” | Hiển thị thông báo lỗi tùy vào điều kiện của trường không hợp lệ  | Hiển thị thông báo lỗi tùy vào điều kiện của trường không hợp lệ  | Đạt      |
| TC_AM_08 | Kiểm tra lỗi khi để trống trường bắt buộc      | Không nhập các trường bắt buộc | 1. Mở trang “Thêm mới bộ phim”.<br>2. Bỏ trống các trường bắt buộc.<br>3. Nhấn nút “Thêm mới” | Hiển thị thông báo “Trường này là bắt buộc ở các trường bắt buộc” | Hiển thị thông báo “Trường này là bắt buộc ở các trường bắt buộc” | Đạt      |

### 3.4.9. Kiểm tra giao tiếp giữa các services

Mô tả: Kiểm tra quá trình giao tiếp giữa các service với cả 2 cách thức là giao tiếp đồng bộ và giao tiếp bất đồng bộ.

Kỳ vọng: Các service có thể giao tiếp và hoạt động một cách chính xác.

Bảng 3-11: Kịch bản kiểm thử chi tiết Giao tiếp giữa các services

| ID        | Mô tả kiểm thử                                | Dữ liệu đầu vào                     | Bước thực hiện                            | Kỳ vọng  | Kết quả kiểm thử   | Đánh giá |
|-----------|---|-------------------------------------|---|--|--|----------|
| TC_CBS_01 | Kiểm tra giao tiếp đồng bộ giữa 2 service     |                                     | 1. Mở trang “Chi tiết phim”.              | Hai service là Movie và Recommend hoạt động và hiển thị các bộ phim tương tự | Hai service là Movie và Recommend hoạt động và hiển thị các bộ phim tương tự | Đạt      |
| TC_CBS_02 | Kiểm tra giao tiếp bất đồng bộ giữa 2 service | Suất chiếu có ghé trống, thanh toán | 1. Chọn suất chiếu.<br>2. Chọn ghế trống. | Reservation service gửi thông điệp bất đồng bộ danh sách                     | Reservation service gửi thông điệp bất đồng bộ danh sách                     | Đạt      |

Hệ thống bán vé xem phim trực tuyến

| ID | Mô tả kiểm thử | Dữ liệu đầu vào | Bước thực hiện           | Kỳ vọng  | Kết quả kiểm thử   | Đánh giá |
|----|----------------|-----------------|--------------------------|--|--|----------|
|    |                | thành công      | 3. Tiến hành thanh toán. | ghế đã được đặt cho Seat service. Seat service nhận thông tin và cập nhật thông tin không cần phản hồi lại | ghế đã được đặt cho Seat service. Seat service nhận thông tin và cập nhật thông tin không cần phản hồi lại |          |

## PHẦN KẾT LUẬN

### 1. Kết quả đạt được

Sau một quá trình học hỏi và thực hành đề tài “Thiết kế và phát triển hệ thống đặt vé xem phim trực tuyến sử dụng kiến trúc microservices và giao diện đa nền tảng” tôi đã đạt được những kết quả sau:

| STT | Các vấn đề cần quan tâm đối với hệ thống Microservices | Hệ thống đạt được   |
|-----|--|---|
| 1   | Xác định phạm vi của mỗi Service                       | Hệ thống được chia thành 7 Services với các nhóm chức năng cụ thể   |
| 2   | Phân tán cơ sở dữ liệu                                 | Hệ thống có cơ sở dữ liệu riêng biệt cho từng dịch vụ cụ thể.   |
| 3   | Liên lạc giữa các Service                              | Hệ thống cung cấp hai hình thức giao tiếp cho các Services:<br>- Đồng bộ: sử dụng Eureka Client.<br>- Bất đồng bộ: sử dụng Kafka. |
| 4   | Hệ thống có tính phục hồi                              | Hệ thống có xử lý các request giữa các Services.  |

### Kiến thức đạt được:

- Giao diện người dùng (UI): nắm được các bước trong việc xây dựng UI với framework Next.js và Angular, phân tách các thành phần giao diện, triển khai các sự kiện, cũng như xử lý các business login ở phần front-end. Đem lại giao diện thân thiện, mượt mà, đáp ứng mọi thiết bị và dễ sử dụng đối với người dùng.
- Biết cách tận dụng kỹ thuật SSR (Server Side Rendering) để tăng tốc độ giao diện, tối ưu cho việc SEO và cải thiện đáng kể UX (User experience)
- Học được cách xây dựng, phân tách các lớp cho một api viết bằng Spring boot. Ngoài ra còn hiểu thêm về cách xây dựng và thiết kế cơ sở dữ liệu theo hướng code-first, xử lý dữ liệu với Spring Date Jpa, cách thêm bảo mật cho ứng dụng bằng Spring Security với JWT.
- Biết cách tích hợp giữa hai thành phần Front-end (Next.js và Angular) và Back-end (Spring boot) một cách liền mạch và ổn định, dễ dàng bảo trì.

- Tiếp xúc thực tế với quá trình mua bán vé xem phim trực tuyến, bao gồm việc xử lý các thông tin liên quan đến vé và thanh toán trực tuyến qua ví điện tử.

### Kinh nghiệm thực tiễn:

- Phát huy được tư duy sáng tạo, kỹ năng phân tích, thiết kế và lập trình hệ thống qua việc giải quyết vấn đề và thực tiễn và xây dựng các chức năng.
- Nâng cao kỹ năng quản lý thời gian.
- Hiểu được cơ bản nghiệp vụ của việc bán vé xem phim trực tuyến.
- Nâng cao khả năng chủ động học hỏi các công nghệ mới thông qua việc đọc các tài liệu.

Qua quá trình thực hiện đề tài, phân tích và xây dựng quyển báo cáo. Và thông qua việc nghiên cứu, sử dụng các công cụ, kỹ thuật thì đề tài có được các ưu điểm và nhược điểm như sau:

### Ưu điểm

- Xây dựng hệ thống theo kiến trúc Microservices đem lại tính linh hoạt, khả năng mở rộng, dễ dàng bảo trì, nâng cấp cũng như mở rộng.
- Website được thiết kế với giao diện hiện đại, thân thiện và đẹp mắt với hiệu suất được tối ưu, giúp đem lại trải nghiệm cao cho người sử dụng.
- Cung cấp chức năng chọn ghế thời gian thực tăng đáng kể trải nghiệm cho người sử dụng.
- Các logic ở phía server được xử lý cẩn thận giảm đáng kể khả năng xảy ra lỗi ngoại lệ không mong muốn lúc vận hành và tăng cao hiệu suất ứng dụng.
- Đáp ứng hầu hết các yêu cầu cần có của một website bán vé xem phim.

### Nhược điểm

- Chưa tối ưu việc sử dụng cơ sở dữ liệu MongoDB để xây dựng cơ sở dữ liệu.
- Chưa tối ưu bảo mật nhiều lớp cho ứng dụng và chưa có các chiến thuật chống tấn công từ bên ngoài.
- Không đủ thời gian để tập trung tối ưu các chức năng.

## 2. Hướng phát triển

Mở rộng tính năng: Bổ sung các tính năng như chương trình khuyến mãi, khách hàng thân thiết và các tính năng thông báo thời gian thực.

Cải thiện bảo mật: Triển khai các biện pháp bảo mật nâng cao như xác thực 2 lớp (2FA) hoặc mã hóa dữ liệu mạnh hơn nhằm bảo vệ thông tin khách hàng.

Phân tích dữ liệu và báo cáo: Tích hợp hệ thống phân tích dữ liệu để theo dõi xu hướng người dùng, đưa ra các quyết định kinh doanh chiến lược dựa trên dữ liệu thu thập.

## TÀI LIỆU THAM KHẢO

### Ebook

1. Craig Walls, Spring in Action - Sixth Edition, 2022, Manning.
2. John Larsen, React hooks in action with suspense and concurrent mode, 2021, Manning.
3. Sam Newman, *Building Microservices: Designing Fine-Grained Systems* - Second Edition, 2021, O'Reilly Media.
4. Bonnie Eisenman, *Learning React: Modern Patterns for Developing React Apps* - Second Edition, 2020, O'Reilly Media.
5. Bonnie Eisenman, *Learning React: Modern Patterns for Developing React Apps* - Second Edition, 2020, O'Reilly Media.

### Website

6. Next.js, the React framework for production-grade web applications, <https://nextjs.org/docs>, visited 2024.
7. Angular, the platform for building mobile and desktop web applications, <https://angular.dev/overview>, visited 2024.
8. Kafka, the distributed event streaming platform, <https://kafka.apache.org/>, visited 2024.
9. Spring Framework, the comprehensive programming and configuration model for Java-based enterprise applications, <https://spring.io/projects/spring-framework/>, visited 2024.
10. Baeldung, the resource for Java, Spring, and technical tutorials, <https://www.baeldung.com/>, visited 2024.
11. GeeksforGeeks, the computer science portal offering programming articles, quizzes, and practice problems, <https://www.geeksforgeeks.org/>, visited 2024
12. TanStack, the suite of open-source libraries for enhancing web development with tools for data fetching, routing, table management, and virtualization, <https://tanstack.com/>, visited 2024.
13. OpenAI, ChatGPT, the AI assistant for generating text-based solutions and knowledge sharing, <https://openai.com/>, visited 2024.
14. Scripted Shadows, insights on software development and technology, <https://scriptedshadows.medium.com/>, visited 2024.