

Báo cáo: Web bán vé xem phim

1. Các chức năng và Use Case

Hệ thống **Web bán vé xem phim** triển khai đa dạng tính năng cho quản trị viên, nhân viên và khách hàng:

- **Quản lý người dùng:** Cho phép đăng ký tài khoản, đăng nhập/đăng xuất, quản lý thông tin cá nhân (thông tin, điểm tích lũy) ¹ ². Người dùng được phân thành hai loại tài khoản: *Employee* (nhân viên/quản trị) và *Customer* (khách hàng) ³. Chức năng hỗ trợ “ghi nhớ đăng nhập” qua cookie để tự động đăng nhập lại.
- **Quản lý phim và thể loại phim:** Các chức năng thêm/sửa/xóa phim và danh mục phim (thể loại). Phim bao gồm các thông tin: tên, ngày phát hành, hạn chế tuổi, hình ảnh, thời lượng, ngôn ngữ, mô tả, đạo diễn, trạng thái chiếu, trailer... ⁴ ⁵. Mỗi phim liên kết với nhiều thể loại (TheLoai). Hệ thống có danh sách phim đang chiếu và sắp chiếu (home page hiển thị phim đang chiếu, sắp chiếu) ⁶. Tính năng tìm kiếm phim theo từ khóa (tên, đạo diễn, ngôn ngữ...) và lọc theo thể loại cũng được hỗ trợ ⁷.
- **Quản lý rạp và phòng chiếu:** Cho phép thêm/sửa/xóa thông tin **rạp chiếu** (RapChieu) và **phòng chiếu** (PhongChieu). Rạp có tên, địa chỉ, ảnh, mô tả, trạng thái hiển thị. Phòng chiếu thuộc một rạp, có màn hình, kích thước, loại phòng. Tính năng thay đổi trạng thái (ẩn/hiện) hoặc xóa nếu không có phòng con liên kết ⁸.
- **Quản lý ghế và loại ghế:** Trong mỗi phòng chiếu có hệ thống ghế (Ghe) với các loại ghế (Vip, thường, ...). Tính năng thêm/sửa/xóa ghế và loại ghế (theo hàng/ngang, trạng thái) tương ứng (có thể tham chiếu qua *Permission*).
- **Quản lý suất chiếu (Lịch chiếu):** Thêm/sửa/xóa suất chiếu cho từng phim tại các phòng chiếu, gồm ngày/giờ bắt đầu kết thúc, số ghế trống, giá vé cơ bản. Các suất chiếu này liên kết đến phim, phòng chiếu. Khách xem có thể chọn ngày giờ thích hợp để đặt vé.
- **Quản lý loại vé:** Hệ thống có loại vé (Ve) khác nhau (người lớn, trẻ em, người cao tuổi...), mỗi loại có giá riêng. Cho phép CRUD loại vé ⁹.
- **Quản lý sản phẩm và combo (thức ăn/đồ uống):** Thêm/sửa/xóa món ăn (Combo/ThucPham) bán kèm (nước ngọt, bỏng ngô, combo gia đình...). Khách có thể chọn mua kèm vé.
- **Quản lý khuyến mãi (Discount/Promotion):** Tạo mã khuyến mãi, thiết lập điều kiện (giảm giá theo %, mã giảm giá). Khi khách đặt vé, hệ thống cho nhập mã khuyến mãi và tự động áp dụng nếu hợp lệ. Mã khuyến mãi sau khi sử dụng sẽ được đánh dấu đã dùng.
- **Quản lý tài khoản và nhóm quyền:** Hệ thống có bảng **NhomQuyen** (Role) và **Quyen** (Permission) để phân quyền. Mỗi nhóm quyền được gán một tập các quyền (ví dụ `create:phim, read:phim, update:phim, delete:phim, create:rapchieu,`

`thongke`, `configwebsite`...) ¹⁰ ¹¹. Hàm `needAnyPermissionOrDie(...)` trong controller kiểm tra quyền truy cập tương ứng ¹² ¹³.

- **Cấu hình website:** Admin có thể cấu hình nội dung hiển thị trang chủ, ví dụ cập nhật banner, thông tin liên hệ, logo... qua trang **Cài đặt website** (phải có quyền `CONFIG_WEBSITE`) ¹².
- **Thống kê và báo cáo:** Chức năng thống kê doanh thu theo tag phim, theo rạp, so sánh doanh thu năm nay vs năm ngoái... Giao diện **Tổng quan** (Dashboard) hiển thị biểu đồ, số liệu về doanh thu vé, đồ ăn, tổng thu nhập ¹⁴. Yêu cầu quyền `THONG_KE` để truy cập báo cáo này.
- **Đặt vé và quy trình thanh toán:** Khách hàng chọn phim, suất chiếu, sau đó chọn ghế và các combo đi kèm. Khi thanh toán, hệ thống sẽ lưu tạm dữ liệu đặt vé (trong `$_SESSION['bookingData']`), sau đó gọi dịch vụ thanh toán qua ví điện tử. Hệ thống hỗ trợ thanh toán qua **Momo** và **ZaloPay**: sau khi khách thanh toán thành công, callback tương ứng (`/pay/callback/momo` hoặc `/pay/callback/zalopay`) được xử lý để lưu đơn hàng (đặt vé) vào cơ sở dữ liệu ¹⁵ ¹⁶ ¹⁷. Quy trình thanh toán gồm: gọi API ví, nhận kết quả callback, lưu thông tin đơn vé (gọi `OrderService::saveOrder(...)` và sử dụng mã khuyến mãi nếu có) ¹⁵, rồi chuyển hướng đến trang “thanh toán thành công” hiển thị mã đơn hàng, mã vạch... Hoặc nếu thất bại, hiển thị “thanh toán thất bại” và hủy bỏ dữ liệu đặt vé.
- **Tìm kiếm và xem lịch chiếu:** Có trang tìm kiếm cho phép khách tìm phim theo từ khóa, lọc theo rạp/chuyên mục. Trang “Lịch chiếu” cho phép xem lịch chiếu của tất cả phim theo ngày tháng.

Sơ đồ Use Case (ví dụ): Các **tác nhân** chính gồm *Quản trị viên*, *Nhân viên* và *Khách hàng*. Quản trị viên/Nhân viên thực hiện các use case như quản lý người dùng, phim, rạp, lịch chiếu, khuyến mãi, xem thống kê, cấu hình website...; Khách hàng thực hiện use case tìm kiếm phim, đặt vé, thanh toán, xem lịch chiếu.

2. Kiến trúc hệ thống

Hệ thống được xây dựng theo mô hình **MVC**:

- **Model/Service:** Tầng này bao gồm các lớp xử lý tương tác với cơ sở dữ liệu (MySQL). Ví dụ, `PhimService` xử lý truy vấn và thao tác bảng `Phim` (xem tạo mới phim, lấy danh sách phim đang chiếu/sắp chiếu) ⁴ ⁶; `CinemaService` xử lý bảng `RapChieu`, `PhongChieu` ⁸; `UserService`, `OrderService`, `PromotionService`... xử lý các logic nghiệp vụ tương ứng. Lớp `QueryBuilder` và `Database` trong `App\Core` hỗ trợ xây dựng và thực thi câu SQL.
- **Controller:** Mỗi route URL được gắn với một phương thức controller thông qua **Route attribute**. Ví dụ, `HomeController::index()` xử lý trang chủ (route `"/`) ¹⁸; `AdminController::setting()` xử lý trang cấu hình website ¹²; `PayController::momoCallback()` xử lý callback Momo ¹⁹. Hàm `Router::dispatch()` trong `index.php` sẽ gọi tương ứng controller phù hợp dựa trên URL.
- **View:** Kết quả trả về thường là các template PHP trong thư mục `app/views`, được gọi qua hàm `view("ten_template", $data)`. Ví dụ trang `home.view.php`, `admin/tong-quan.view.php`, `chi-tiet.view.php`... Các view này sử dụng HTML, TailwindCSS và có thể nhúng Alpine.js/jQuery để tương tác phía client.

- **Cơ sở dữ liệu:** Sử dụng MySQL, các bảng chính gồm `NguoiDung`, `TaiKhoan`, `Phim`, `RapChieu`, `PhongChieu`, `SuatChieu`, `Ghe`, `LoaiVe`, `KhuyenMai`, `HoaDon`, `ThucPham` (combo), `NhomQuyền`, `Quyền`, vv. Mỗi service tương ứng xử lý bảng riêng.

Theo thiết kế, khi khởi động (`core/Preload.php`), hệ thống **kết nối cơ sở dữ liệu** (`Database::init_db()`) và đăng ký tất cả controllers qua `Router::load_from_class(ClassName)` ²⁰. Khi có request, `Router::dispatch()` sẽ chọn controller phù hợp gọi đến action tương ứng.

3. Công nghệ – Thư viện – Framework sử dụng

- **Backend:** PHP (với Composer và PSR autoloader). Mã nguồn sử dụng PHP 8+ (có `enum`, attributes), và Core classes tự viết cho Router, Database, QueryBuilder.
- **Cơ sở dữ liệu:** MySQL.
- **Frontend:** Các giao diện sử dụng **TailwindCSS** (xem `tailwind.config.js` để cấu hình theme, plugin daisyUI) ²¹ ²². Phía client còn dùng **jQuery** và **Alpine.js** (giúp xử lý tương tác DOM/form động) để tạo các hiệu ứng/và xử lý AJAX cơ bản.
- **Tích hợp thanh toán:** Hỗ trợ thanh toán bằng ví điện tử **Momo** và **ZaloPay**. Các route callback `/pay/callback/momo` và `/pay/callback/zalopay` lần lượt nhận dữ liệu trả về từ cổng thanh toán và ghi nhận kết quả ¹⁹ ²³. Trong `PayController`, hệ thống chọn chiến lược thanh toán (`getPaymentStrategy`) tương ứng và xử lý lưu đơn hàng.
- **Thư viện/khung khác:** Sử dụng một số thư viện hỗ trợ như `Carbon` (xử lý ngày/tháng), `dompdf` (tạo file vé in), plugin `daisyUI` cho Tailwind ²².

4. Tác nhân trong hệ thống

- **Quản trị viên (Admin):** Tài khoản loại *Employee* có thể truy cập dashboard admin (route `/admin`), xem thống kê, cấu hình website, xem nhật ký hệ thống... Có quyền cao nhất, có thể tạo/sửa/xóa hầu hết các dữ liệu (phim, rạp, suất chiếu, giá vé, combo, khuyến mãi, người dùng, nhóm quyền...) tùy theo quyền được gán. Ví dụ phân quyền `CONFIG_WEBSITE` để truy cập trang cài đặt ¹², `THONG_KE` để xem báo cáo ¹⁴. Một số tài khoản nhân viên khác (cũng loại *Employee*) có thể có các nhóm quyền hạn chế hơn (chỉ quản lý vé/bán vé, hoặc chỉ quản lý combo, v.v).
- **Nhân viên bán vé (Employee):** Tài khoản cũng thuộc *Employee*, có thể là nhân viên phòng vé hoặc quản lý rạp. Thường có quyền truy cập các chức năng quản lý nghiệp vụ (thêm vé, kiểm soát đặt vé, in vé, v.v) nhưng có thể không có toàn quyền như admin (phụ thuộc nhóm quyền được thiết lập).
- **Khách hàng (Customer):** Tài khoản loại *Customer* chỉ có thể truy cập giao diện người dùng: xem phim, xem lịch chiếu, tìm kiếm, đặt vé, thanh toán. Họ không được truy cập các trang quản trị nội bộ. Khi đăng nhập thành công, nếu là Customer sẽ được chuyển hướng về trang chủ bình thường ²⁴.

Mỗi nhóm quyền (`NhomQuyền`) sẽ có một danh sách các quyền (`Quyền`) cụ thể như: `read:phim`, `update:phim`, `create:rapchieu`, `delete:nhomquyen`, `thongke`, `configwebsite`... ¹⁰ ¹¹. Trong quá trình login, hệ thống sẽ gán quyền vào session dựa trên nhóm quyền của tài khoản ¹ ¹. Khi truy cập các trang yêu cầu, controller dùng hàm `needPermission()` hoặc `needAnyPermissionOrDie()` để kiểm tra quyền.

5. Các điểm nổi bật khác

- **Cấu hình hệ thống:** Các thông số như kết nối DB, API keys (Momo/ZaloPay), cài đặt auth (secret key, thời gian cookie) được lưu trong file `config.php`. Khi khởi động, `index.php` nạp file config và gọi `Database::init_db()` khởi tạo kết nối. ²⁰
- **Bảo mật:** Mật khẩu người dùng được mã hóa trong database (xem `AccountService` ở `TaiKhoan`). Hệ thống khóa tài khoản nếu nhập sai mật khẩu quá nhiều lần (trạng thái khóa `TrangThai=2`). Cơ chế *remember me* sử dụng cookie HMAC để tránh cookie cũ còn hiệu lực sau khi đổi mật khẩu ²⁵ ²⁶. Các route admin đều kiểm tra session user đã đăng nhập và quyền tương ứng trước khi thực hiện.
- **Quy trình thanh toán và đặt vé:** Quá trình gồm các bước:
 - **Chọn ghế:** Tại trang chi tiết phim, chọn suất chiếu. Hệ thống lấy danh sách ghế trống của phòng chiếu đó để hiển thị.
 - **Chọn vé & combo:** Người dùng đánh dấu ghế muốn mua, chọn số lượng combo/phần ăn kèm, nhập mã khuyến mãi nếu có.
 - **Tạo đơn và thanh toán:** Khi xác nhận, hệ thống lưu thông tin vào session (`$_SESSION['bookingData']`) rồi chuyển hướng đến cổng thanh toán (ví Momo/ZaloPay) qua API.
 - **Callback thanh toán:** Ví dụ, sau khi Momo thanh toán thành công, Momo gọi lại `GET /pay/callback/momo` với thông tin kết quả. Ở đây, `PayController::momoCallback()` sẽ kiểm tra kết quả (thành công hay thất bại), nếu thành công thì gọi hàm nội bộ `createOrderModel()` để lưu đơn vào database (gọi `OrderService::saveOrder(...)`) ¹⁵ và chuyển đến trang “Thanh toán thành công” hiển thị mã vé, thanh toán loại nào, mã vạch... Nếu thất bại sẽ hiện thông báo lỗi và bỏ đơn.
 - **In vé/EMA:** Hệ thống sử dụng thư viện `dompdf` để tạo file vé PDF có mã vạch tương ứng với mã đơn, người dùng có thể tải về hoặc nhận qua email.
 - **Sơ đồ:** Tóm lại, ứng dụng gồm ba tầng chính Model (cơ sở dữ liệu + dịch vụ), Controller (xử lý request, gọi service và view), View (template HTML/CSS/JS). Mỗi quan hệ là Controller điều phối luồng giữa view và model: ví dụ `HomeController` lấy dữ liệu phim từ `PhimService` rồi trả về view `home` ¹⁸, `AdminController` lấy thống kê từ `StatisticService` rồi trả về view `tong-quan` ¹⁴.

Nguồn tham khảo: Tất cả thông tin trên dựa vào phân tích trực tiếp mã nguồn trong repository `Mirai3103/movie-ticket` (các controller và service liên quan) ¹⁸ ¹² ¹⁵ ²¹ ¹⁰.

¹ ² ³ ²⁵ ²⁶ `UserService.php`

<https://github.com/Mirai3103/movie-ticket/blob/541afdde2cd3119105fa57012236b541c23549d7/app/services/UserService.php>

⁴ ⁵ ⁶ `PhimService.php`

<https://github.com/Mirai3103/movie-ticket/blob/541afdde2cd3119105fa57012236b541c23549d7/app/services/PhimService.php>

7 18 24 **HomeController.php**

<https://github.com/Mirai3103/movie-ticket/blob/541afdde2cd3119105fa57012236b541c23549d7/app/controllers/HomeController.php>

8 **CinemaService.php**

<https://github.com/Mirai3103/movie-ticket/blob/541afdde2cd3119105fa57012236b541c23549d7/app/services/CinemaService.php>

9 10 11 **Permission.php**

<https://github.com/Mirai3103/movie-ticket/blob/541afdde2cd3119105fa57012236b541c23549d7/app/dtos/Permission.php>

12 13 14 **AdminController.php**

<https://github.com/Mirai3103/movie-ticket/blob/541afdde2cd3119105fa57012236b541c23549d7/app/controllers/admin/AdminController.php>

15 16 17 19 23 **PayController.php**

<https://github.com/Mirai3103/movie-ticket/blob/541afdde2cd3119105fa57012236b541c23549d7/app/controllers/PayController.php>

20 **Preload.php**

<https://github.com/Mirai3103/movie-ticket/blob/541afdde2cd3119105fa57012236b541c23549d7/core/Preload.php>

21 22 **tailwind.config.js**

<https://github.com/Mirai3103/movie-ticket/blob/541afdde2cd3119105fa57012236b541c23549d7/tailwind.config.js>