

PODCAST: ADMIN PANEL CINEBOOK

PHẦN 2: DATABASE VÀ THUẬT TOÁN - "Bí mật đằng sau màn hình"

Thời lượng ước tính: 30-35 phút

Microphone icon: CHÀO MỪNG TRỞ LẠI

Xin chào các bạn đã quay trở lại với phần 2 của series! Ở phần trước, chúng ta đã hiểu được cấu trúc tổng quan của Admin Panel. Hôm nay, mình sẽ đưa các bạn vào hậu trường - nơi những thuật toán thú vị đang hoạt động âm thầm.

Chúng ta sẽ khám phá ba chủ đề chính:

- Làm sao tạo ra sơ đồ ghế ngồi trong rạp?
- Công thức tính giá vé phức tạp như thế nào?
- Hệ thống QR check-in hoạt động ra sao?

Sẵn sàng chưa? Bắt đầu thôi!

Microphone icon: PHẦN 1: QUẢN LÝ PHÒNG CHIẾU - "Kiến trúc sư của rạp phim"

Hãy tưởng tượng bạn là kiến trúc sư, được thuê để thiết kế một phòng chiếu phim. Bạn cần trả lời những câu hỏi:

- Phòng này có bao nhiêu hàng ghế?
- Mỗi hàng có bao nhiêu ghế?
- Ghế nào là ghế thường, ghế nào là VIP, ghế nào là ghế đôi cho cặp đôi?
- Màn hình loại gì - 2D thường, 3D, hay IMAX?

Trong hệ thống của chúng ta, tất cả những thông tin này được lưu trữ và quản lý một cách có hệ thống.

Ba loại ghế ngồi:

Đầu tiên là ghế Standard - ghế tiêu chuẩn. Đây là loại ghế phổ biến nhất, giá rẻ nhất, thường ở phía trước và hai bên. Màu xanh lá cây.

Thứ hai là ghế VIP - ghế cao cấp. Vị trí đẹp nhất, thường ở giữa phòng, nơi nhìn và âm thanh tốt nhất. Giá cao hơn. Màu vàng gold.

Thứ ba là ghế Couple - ghế đôi dành cho cặp đôi. Thực chất là hai ghế liền nhau được nối thành một. Thường ở hàng cuối, riêng tư hơn. Giá bằng khoảng một ghế VIP nhân đôi. Màu hồng.

Thuật toán tạo sơ đồ ghế:

Bây giờ, khi admin tạo phòng mới, họ chỉ cần nhập:

- Tên phòng: "Phòng 1" chẳng hạn
- Số hàng: ví dụ 8 hàng

- Số ghế mỗi hàng: ví dụ 10 ghế
- Loại màn hình: 2D, 3D, hay IMAX

Hệ thống sẽ tự động tạo ra 80 ghế - 8 nhân 10 - và đánh số chúng.

Cách đánh số rất thông minh. Hàng đầu tiên là hàng A, tiếp theo là B, rồi C, cứ thế cho đến Z. Mỗi ghế trong hàng được đánh số từ 1 đến 10. Vậy ghế đầu tiên của hàng đầu là A1, ghế cuối của hàng cuối là H10.

Mặc định, tất cả ghế đều là ghế Standard. Nhưng admin có thể tùy chỉnh - click vào ghế nào đó, đổi thành VIP hoặc Couple.

Tính năng hay: Template sẵn

Để tiết kiệm thời gian, hệ thống có các template sẵn:

Template "VIP Center" - VIP ở giữa: Tự động đặt ghế VIP ở khu vực trung tâm phòng, nơi góc nhìn đẹp nhất. Ghế Standard ở hai bên và phía trước.

Template "Couple Back" - Ghế đôi hàng cuối: Tự động biến hàng cuối cùng thành toàn ghế đôi. Ghế một và hai thành một cặp, ghế ba và bốn thành một cặp, cứ thế.

Template "Cinema Style" - Phong cách rạp thật: Kết hợp cả hai. Hàng trước là Standard, giữa phòng là VIP, hàng cuối là Couple. Giống như các rạp CGV hay Lotte thật.

Admin click một nút, cả sơ đồ tự động thay đổi. Không cần chọn từng ghế một.

PHẦN 2: GIAO DỊCH DATABASE - "Không ai bị bỏ lại phía sau"

Đây là khái niệm cực kỳ quan trọng trong lập trình, mà rất ít người giải thích rõ ràng. Mình sẽ cố gắng làm cho nó dễ hiểu nhất.

Hãy tưởng tượng bạn đang chuyển tiền qua ngân hàng. Bạn chuyển một triệu đồng từ tài khoản A sang tài khoản B. Quá trình này gồm hai bước:

1. Trừ một triệu từ tài khoản A
2. Cộng một triệu vào tài khoản B

Bây giờ, điều gì xảy ra nếu hệ thống lỗi sau bước 1, trước bước 2? Tiền của bạn đã bị trừ, nhưng người nhận chưa được cộng. Tiền biến mất!

Đây là lý do chúng ta cần Transaction - giao dịch. Transaction đảm bảo rằng: hoặc TẤT CẢ các bước đều thành công, hoặc KHÔNG có bước nào xảy ra. Không có chuyện "nửa vời".

Ví dụ thực tế trong dự án:

Khi tạo một suất chiếu mới, hệ thống phải làm những việc sau:

1. Tạo bản ghi suất chiếu - phim gì, phòng nào, ngày nào, giờ nào
2. Tạo bảng giá cho suất đó - mỗi loại ghế giá bao nhiêu
3. Tạo trạng thái cho 80 ghế - tất cả đều "còn trống"

Nếu bước 1 và 2 thành công, nhưng bước 3 lỗi ở ghế thứ 50? Chúng ta có suất chiếu, có giá, nhưng chỉ có 50 ghế. 30 ghế còn lại không tồn tại trong hệ thống. Khách hàng không thể đặt những ghế đó. Hỗn loạn!

Với Transaction, quy trình như sau:

- Bắt đầu giao dịch - "Này database, từ giờ tất cả thay đổi đều là tạm thời nhé"
- Thực hiện bước 1, 2, 3
- Nếu tất cả OK: Xác nhận - "OK, lưu hết đi"
- Nếu có lỗi: Hoàn tác - "Quên hết đi, coi như chưa làm gì"

Kết quả là database luôn trong trạng thái nhất quán. Không có "dữ liệu rác".

⌚ PHẦN 3: CÔNG THỨC TÍNH GIÁ VÉ - "Bài toán tiền bạc"

Đây là phần khách hàng quan tâm nhất, và cũng là phần thú vị nhất về mặt logic.

Giá vé không cố định, mà phụ thuộc vào nhiều yếu tố. Hãy nghĩ về việc mua vé máy bay - giá phụ thuộc vào hạng ghế, thời điểm, hãng bay, đúng không? Vé xem phim cũng vậy.

Công thức cộng dồn:

Giá vé cuối cùng bằng giá cơ bản, cộng phụ phí màn hình, cộng phụ phí giờ cao điểm.

Hãy phân tích từng phần:

Phần 1: Giá cơ bản theo loại ghế

Mỗi loại ghế có một mức giá "nền":

- Ghế Standard: 80 nghìn đồng
- Ghế VIP: 120 nghìn đồng
- Ghế Couple: 180 nghìn đồng - rẻ hơn mua hai ghế VIP riêng lẻ

Đây là giá "tối thiểu" - chưa tính gì cả.

Phần 2: Phụ phí loại màn hình

Xem phim 3D hay IMAX cần công nghệ đắt tiền hơn, nên có phụ phí:

- Màn hình 2D: không phụ phí
- Màn hình 3D: cộng thêm 30 nghìn
- Màn hình IMAX: cộng thêm 50 nghìn

Phụ phí này đã được gắn với phòng chiếu. Khi tạo phòng, admin chọn loại màn hình. Khi suất chiếu được xếp vào phòng đó, phụ phí tự động được áp dụng.

Phần 3: Phụ phí giờ cao điểm

Đây là phần linh hoạt nhất. Mỗi suất chiếu, admin có thể đặt một mức phụ thu riêng.

Ví dụ:

- Suất 9 giờ sáng ngày thường: phụ thu 0 - ít người xem

- Suất 7 giờ tối cuối tuần: phụ thu 30 nghìn - nhiều người xem

Admin toàn quyền quyết định. Họ có thể chạy khuyến mãi bằng cách đặt phụ thu âm - giảm giá.

Ví dụ tính cụ thể:

Khách hàng muốn mua 2 vé VIP, xem phim IMAX, suất 9 giờ tối thứ Bảy.

Giá một vé:

- Giá cơ bản VIP: 120.000
- Phụ phí IMAX: 50.000
- Phụ phí giờ cao điểm: 30.000
- Tổng một vé: 200.000 đồng

Hai vé: 400.000 đồng.

Hệ thống tính toán tự động, hiển thị rõ ràng cho khách hàng trước khi thanh toán.

■ PHẦN 4: QUẢN LÝ SUẤT CHIẾU - "Người sắp xếp lịch"

Suất chiếu là "linh hồn" của rạp phim. Nó kết nối phim với phòng, với thời gian, với giá cả.

Thông tin của một suất chiếu:

- Chiếu phim gì
- Ở phòng nào
- Ngày nào
- Giờ nào
- Giá mỗi loại ghế bao nhiêu

Quy tắc quan trọng: Không được trùng lịch

Một phòng không thể chiếu hai phim cùng lúc. Vì vậy, khi tạo suất chiếu mới, hệ thống phải kiểm tra: "Phòng này, ngày này, giờ này đã có suất nào chưa?"

Nếu đã có, từ chối và báo lỗi: "Suất chiếu này đã bị chiếm."

Nếu chưa có, cho phép tạo.

Tạo trạng thái ghế cho suất chiếu:

Khi suất chiếu được tạo, hệ thống tự động tạo "bản sao trạng thái" của tất cả ghế trong phòng đó.

Tại sao cần làm vậy?

Vì ghế A1 ở suất 7 giờ tối có thể đã có người đặt, nhưng ghế A1 ở suất 9 giờ tối vẫn còn trống. Mỗi suất chiếu cần theo dõi trạng thái ghế riêng.

Ban đầu, tất cả ghế đều "available" - còn trống. Khi có người đặt, ghế chuyển sang "reserved" - đang giữ. Khi thanh toán xong, ghế chuyển sang "booked" - đã đặt.

➡ PHẦN 5: VÒNG ĐỜI CỦA MỘT ĐƠN ĐẶT VÉ - "Từ lúc sinh ra đến khi hoàn thành"

Mỗi đơn đặt vé đều có một "cuộc đời" với nhiều giai đoạn. Hiểu được điều này rất quan trọng.

Giai đoạn 1: Pending - Chờ xử lý

Khách hàng chọn ghế, bấm "Đặt vé". Đơn được tạo với trạng thái "pending" - đang chờ thanh toán. Ghế được "giữ" tạm thời trong 15 phút.

Tại sao 15 phút? Để khách có thời gian nhập thông tin thanh toán, nhưng không giữ ghế quá lâu khiến người khác không đặt được.

Giai đoạn 2: Confirmed - Đã xác nhận

Khách thanh toán thành công. Đơn chuyển sang "confirmed". Ghế chính thức thuộc về khách. Hệ thống gửi mã QR để check-in.

Giai đoạn 3: Checked - Đã check-in

Khách đến rạp, nhân viên quét mã QR. Đơn chuyển sang "checked". Khách có thể vào phòng chiếu.

Giai đoạn 4: Completed - Hoàn thành

Sau khi suất chiếu kết thúc, đơn tự động chuyển sang "completed". Đây là trạng thái cuối cùng.

Nhưng không phải đơn nào cũng đi hết con đường...

Có hai "ngõ cụt":

Expired - Hết hạn: Nếu sau 15 phút khách chưa thanh toán, đơn tự động "expired". Ghế được trả lại, người khác có thể đặt.

Cancelled - Bị hủy: Admin có thể hủy đơn vì lý do nào đó - khách yêu cầu hoàn tiền, có lỗi kỹ thuật... Ghế cũng được trả lại.

Thuật toán hủy đơn:

Khi hủy đơn, hệ thống không chỉ đổi trạng thái đơn, mà còn phải "giải phóng" ghế. Quy trình:

1. Kiểm tra đơn có thể hủy không - đơn đã hủy hoặc đã hết hạn thì không hủy lại được
2. Đổi trạng thái đơn thành "cancelled"
3. Với mỗi ghế trong đơn, tìm đến suất chiếu tương ứng và đổi trạng thái ghế về "available"
4. Tất cả được bọc trong Transaction - nếu lỗi thì hoàn tác hết

Đây là ví dụ điển hình của việc dùng Transaction. Không thể để xảy ra tình huống đơn bị hủy nhưng ghế vẫn "booked" - khách khác không đặt được.

➡ PHẦN 6: HỆ THỐNG QR CHECK-IN - "Vé điện tử thông minh"

Đây là tính năng hiện đại mà mình rất thích. Thay vì in vé giấy, khách hàng nhận một mã QR trên điện thoại. Khi đến rạp, chỉ cần quét mã là xong.

Cách tạo mã QR:

Khi khách thanh toán thành công, hệ thống tạo một mã QR unique - duy nhất, không trùng với bất kỳ mã nào khác.

Mã này được tạo bằng thuật toán SHA-256 - một thuật toán mã hóa nổi tiếng. Input là sự kết hợp của:

- ID của đơn đặt vé
- Thông tin ghế
- Thời gian tạo chính xác đến micro giây

Output là một chuỗi 64 ký tự, ví dụ: "a1b2c3d4e5f6..." Chuỗi này gần như không thể đoán được hay giả mạo.

Quy trình check-in:

1. Nhân viên rạp vào trang QR Check-in trên máy tính
2. Khách đưa điện thoại có mã QR
3. Nhân viên nhập mã hoặc quét bằng camera
4. Hệ thống kiểm tra: Mã này có tồn tại không? Đã dùng chưa?
5. Nếu hợp lệ: Hiển thị thông tin - tên khách, phim gì, ghế nào, giờ nào
6. Nhân viên xác nhận check-in
7. Mã được đánh dấu "đã sử dụng" - không thể dùng lại

An toàn và chống gian lận:

Tại sao khó gian lận?

Thứ nhất, mã QR được tạo bằng hash - không thể "biết" ra một mã hợp lệ.

Thứ hai, mã chỉ dùng được một lần. Nếu ai đó chụp lại mã của bạn, họ cũng không thể vào vì bạn đã check-in trước.

Thứ ba, hệ thống ghi lại thời gian check-in. Nếu có tranh chấp, có bằng chứng rõ ràng.

Tính năng Preview:

Trước khi check-in, nhân viên có thể "xem trước" thông tin mà không cần đánh dấu là đã sử dụng. Điều này hữu ích khi cần kiểm tra mà chưa muốn check-in - ví dụ khách hỏi "tôi ngồi ghế nào nhỉ?"

❸ PHẦN 7: LỌC VÀ TÌM KIẾM - "Tìm kim trong đống rơm"

Khi có hàng nghìn đơn đặt vé, làm sao admin tìm được đơn cần tìm?

Tìm kiếm theo từ khóa:

Admin gõ email hoặc tên khách hàng. Hệ thống tìm tất cả đơn có thông tin khớp.

Ví dụ: Gõ "nguyen", hệ thống tìm tất cả đơn của khách có tên chứa "nguyen" - Nguyễn Văn A, Lê Thị Nguyên, v.v.

Lọc theo tiêu chí:

Admin có thể lọc:

- Theo trạng thái: chỉ xem đơn "confirmed", hoặc chỉ "pending"

- Theo trạng thái thanh toán: đã trả tiền hay chưa
- Theo ngày: đơn đặt ngày hôm nay, hoặc trong tuần này

Kết hợp lọc và tìm kiếm:

Đây là phần hay. Admin có thể vừa tìm "nguyen" vừa lọc "chỉ đơn đã hủy". Kết quả là tất cả đơn bị hủy của các khách có tên chứa "nguyen".

Kỹ thuật xây dựng truy vấn linh hoạt:

Thay vì viết nhiều truy vấn cố định, hệ thống xây dựng truy vấn "từng bước". Bắt đầu từ truy vấn cơ bản, rồi thêm điều kiện nếu admin chọn lọc.

Nếu admin chọn lọc theo trạng thái? Thêm điều kiện trạng thái. Nếu admin nhập từ khóa tìm kiếm? Thêm điều kiện tìm kiếm. Không chọn gì? Trả về tất cả.

Cách này linh hoạt và dễ mở rộng. Muốn thêm bộ lọc mới? Chỉ cần thêm một vài dòng.

✍ PHẦN 8: PHÂN TRANG - "Chia nhỏ để dễ nuốt"

Tưởng tượng bạn có 10.000 bộ phim trong hệ thống. Nếu hiển thị tất cả trên một trang, hai vấn đề xảy ra:

Một là chậm. Tải 10.000 phim cần thời gian và bộ nhớ.

Hai là khó đọc. Ai muốn cuộn qua 10.000 dòng?

Giải pháp là phân trang - chia thành nhiều trang nhỏ, mỗi trang 20 phim chẳng hạn.

Cách hoạt động:

Khi admin vào trang danh sách phim, hệ thống chỉ lấy 20 phim đầu tiên từ database. Nhanh và nhẹ.

Bên dưới danh sách có các số trang: 1, 2, 3... 500. Admin click trang 3, hệ thống lấy 20 phim tiếp theo - từ phim thứ 41 đến 60.

Giữ lại bộ lọc khi chuyển trang:

Đây là chi tiết nhỏ nhưng quan trọng. Giả sử admin đang lọc "chỉ phim đang chiếu", trang 1 có 20 phim đang chiếu. Họ click trang 2. Hệ thống phải nhớ bộ lọc và tiếp tục lọc ở trang 2.

Nếu không nhớ, click trang 2 sẽ hiện tất cả phim, không chỉ phim đang chiếu. Rất khó chịu!

Hệ thống giải quyết bằng cách "gắn" bộ lọc vào địa chỉ trang. Khi admin lọc, địa chỉ thành "/admin/movies?status=now_showing". Khi click trang 2, địa chỉ thành "/admin/movies?status=now_showing&page=2". Bộ lọc được giữ nguyên.

👉 PHẦN 9: TỐI ƯU HIỆU NĂNG - "Nhanh hơn, mạnh hơn"

Đây là chủ đề chuyên sâu, nhưng mình sẽ giải thích đơn giản.

Vấn đề N+1 Query:

Đây là lỗi hiệu năng phổ biến nhất trong lập trình web. Hãy tưởng tượng:

Bạn muốn hiển thị 100 đơn đặt vé, mỗi đơn cần hiện tên khách hàng.

Cách tệ: Lấy 100 đơn bằng 1 truy vấn. Rồi với MỖI đơn, thực hiện 1 truy vấn nữa để lấy thông tin khách. Tổng cộng: $1 + 100 = 101$ truy vấn!

Cách tốt: Lấy 100 đơn bằng 1 truy vấn. Rồi lấy tất cả khách hàng liên quan bằng 1 truy vấn nữa. Tổng cộng: 2 truy vấn.

Sự khác biệt rất lớn. 101 truy vấn mất vài giây. 2 truy vấn mất vài mili giây.

Giải pháp: Eager Loading

Hệ thống được thiết kế để tự động tải trước các thông tin liên quan. Khi lấy đơn đặt vé, đồng thời lấy luôn thông tin khách hàng, thông tin phim, thông tin ghế. Tất cả trong một vài truy vấn.

Admin không cần quan tâm điều này - nó xảy ra tự động đằng sau. Nhưng kết quả là trang tải nhanh hơn nhiều.

TÓM TẮT PHẦN 2

Chúng ta đã khám phá những bí mật kỹ thuật thú vị:

1. **Quản lý phòng chiếu** với ba loại ghế và các template sẵn có
2. **Transaction** đảm bảo dữ liệu luôn nhất quán
3. **Công thức giá vé** linh hoạt với nhiều yếu tố
4. **Vòng đời đơn đặt vé** từ pending đến completed
5. **QR Check-in** an toàn với mã SHA-256
6. **Lọc và tìm kiếm** linh hoạt
7. **Phân trang** giúp tải nhanh
8. **Eager Loading** tối ưu hiệu năng

Trong phần tiếp theo, chúng ta sẽ nói về thiết kế giao diện - tại sao màu sắc được chọn như vậy, tại sao nút bấm đặt ở vị trí đó, và làm sao để admin làm việc hiệu quả nhất.

Hẹn gặp lại ở phần 3!

Tiếp theo: Phần 3 - Thiết Kế UX/UI