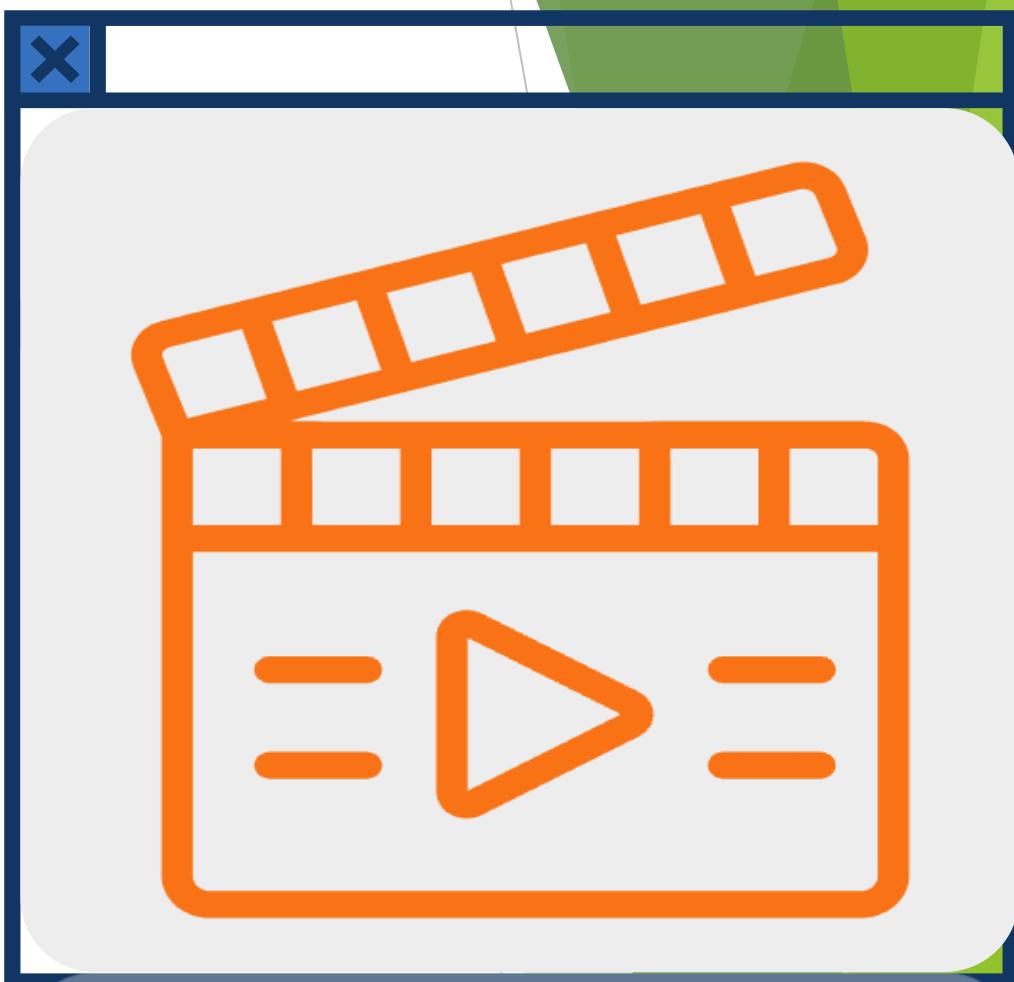


BÁO CÁO KẾT THÚC HỌC PHẦN CÔNG NGHỆ PHẦN MỀM

GVHD: Ts. Nguyễn Bảo Ân

NHÓM 6

XÂY DỰNG HỆ THỐNG ĐẶT VÉ XEM PHIM TRỰC TUYẾN



2 1. TỔNG QUAN

1.1 Lý do chọn đề tài

- ❖ Nhu cầu xem phim tại rạp tăng cao
- ❖ Mua vé truyền thống gây bất tiện:
 - Xếp hàng lâu
 - Khó chọn ghế
 - Dễ quá tải tại quầy vé
- ❖ Người dùng ưa chuộng trải nghiệm số
- Cần giải pháp đặt vé nhanh, tiện, hiện đại

3 1. TỔNG QUAN

1.2 Tên đề tài & Chủ đề

❖ **Tên đề tài:**

 *Movie Ticket Booking System*

❖ **Chủ đề:**

- Thiết kế website đặt vé xem phim
- Tính năng chính:
 -  Xem danh sách phim & lịch chiếu
 -  Đặt ghế, thanh toán giả lập
 -  Quản lý, báo cáo thống kê

1. TỔNG QUAN

1.3 Mục tiêu ứng dụng

- ❖ **Người dùng:** Đăng nhập, đăng ký, cập nhật thông tin, đặt vé nhanh, chọn ghế trực quan.
- ❖ **Rạp chiếu phim:** Quản lý người dùng, rạp, phim, lịch chiếu, tình trạng ghế, vé theo thời gian thực.
- ❖ **Nhà quản lý:** Thống kê doanh thu, dễ mở rộng quy mô hệ thống.

1. TỔNG QUAN

1.4 Tổng quan công nghệ phần mềm

- ❖ Quy trình SDLC: **Phân tích** → **Thiết kế** → **Lập trình** → **Kiểm thử** → **Triển khai**
 - Kiến trúc Client–Server
 - Frontend: React
 - Backend: Node.js + Express
 - Kiến trúc backend: module hóa → dễ bảo trì & mở rộng.
 - Cơ sở dữ liệu: MySQL
- ❖ Triển khai: Docker Compose giúp đóng gói & triển khai đồng nhất môi trường.

⁶ 1. TỔNG QUAN

1.5 Tính cần thiết của phần mềm phân tán/cloud

- ❖ Hệ thống phải phục vụ nhiều người dùng đồng thời
- ❖ Cloud đảm bảo khả năng mở rộng và hiệu năng cao
- ❖ Triển khai dễ dàng, tiết kiệm chi phí

2. PHÂN TÍCH YÊU CẦU HỆ THỐNG

2.1 Chức năng chính (Functional Requirements)

Người dùng (khách hàng)	Quản trị viên (admin)
<ul style="list-style-type: none">➤ Đăng ký, đăng nhập, cập nhật thông tin cá nhân.➤ Xem danh sách phim đang chiếu và sắp chiếu.➤ Xem thông tin chi tiết phim➤ Tìm kiếm, lọc phim theo thể loại/thời gian.➤ Đặt vé: chọn ghế từ sơ đồ, thanh toán, xem lại vé đã mua.	<ul style="list-style-type: none">➤ Đăng nhập vào hệ thống quản trị.➤ Quản lý phim, rạp, cụm rạp, phòng chiếu, suất chiếu.➤ Quản lý người dùng.➤ Xem báo cáo thống kê doanh thu, vé bán, tình trạng phim.

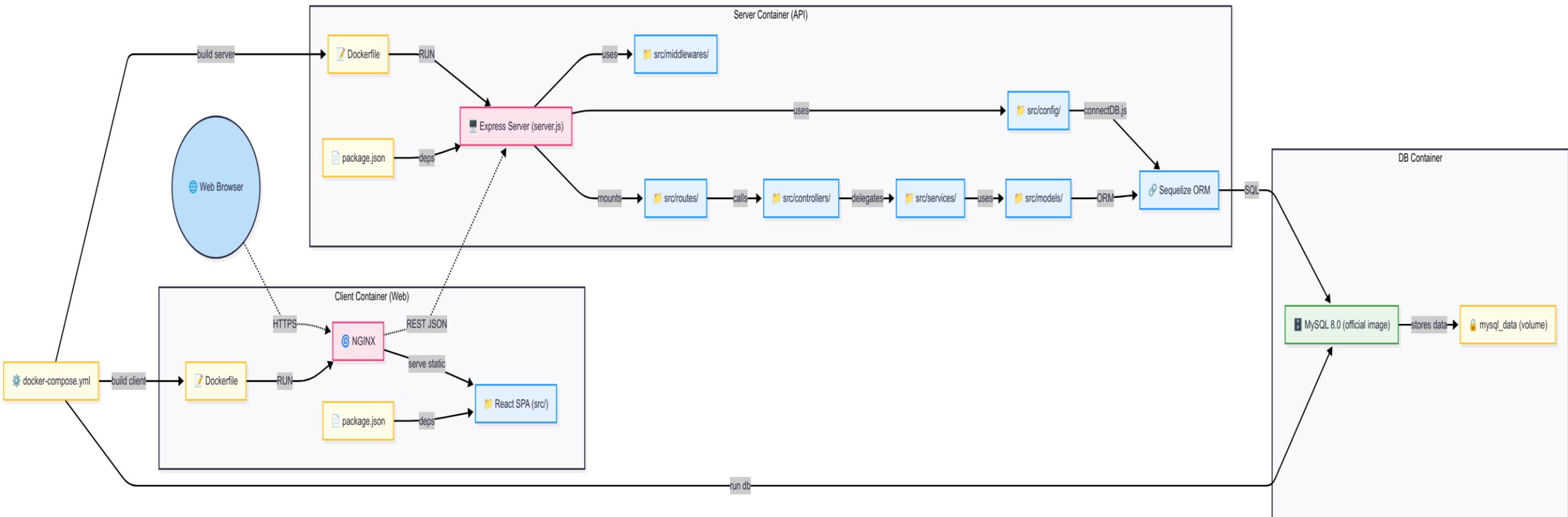
8 2. PHÂN TÍCH YÊU CẦU HỆ THỐNG

2.2 Yêu cầu phi chức năng (Non-functional Requirements)

- ❖ Hiệu suất
- ❖ Quyền riêng tư
- ❖ Tính sẵn sàng
- ❖ Khả năng mở rộng
- ❖ Tính dễ sử dụng
- ❖ Tính duy trì & phát triển
- ❖ Tính bảo mật & an toàn

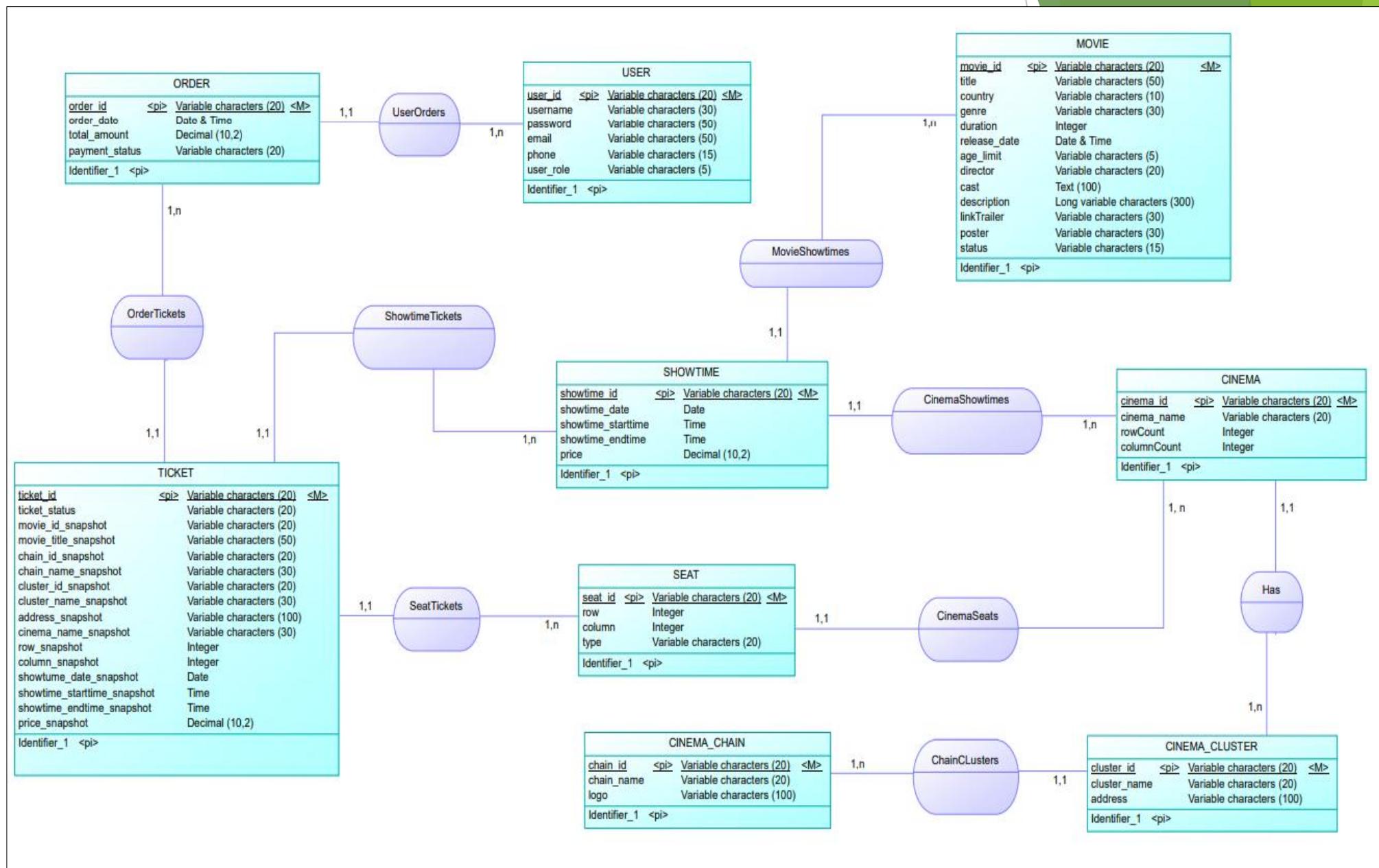
9 3. THIẾT KẾ HỆ THỐNG

3.1 Kiến trúc tổng thể



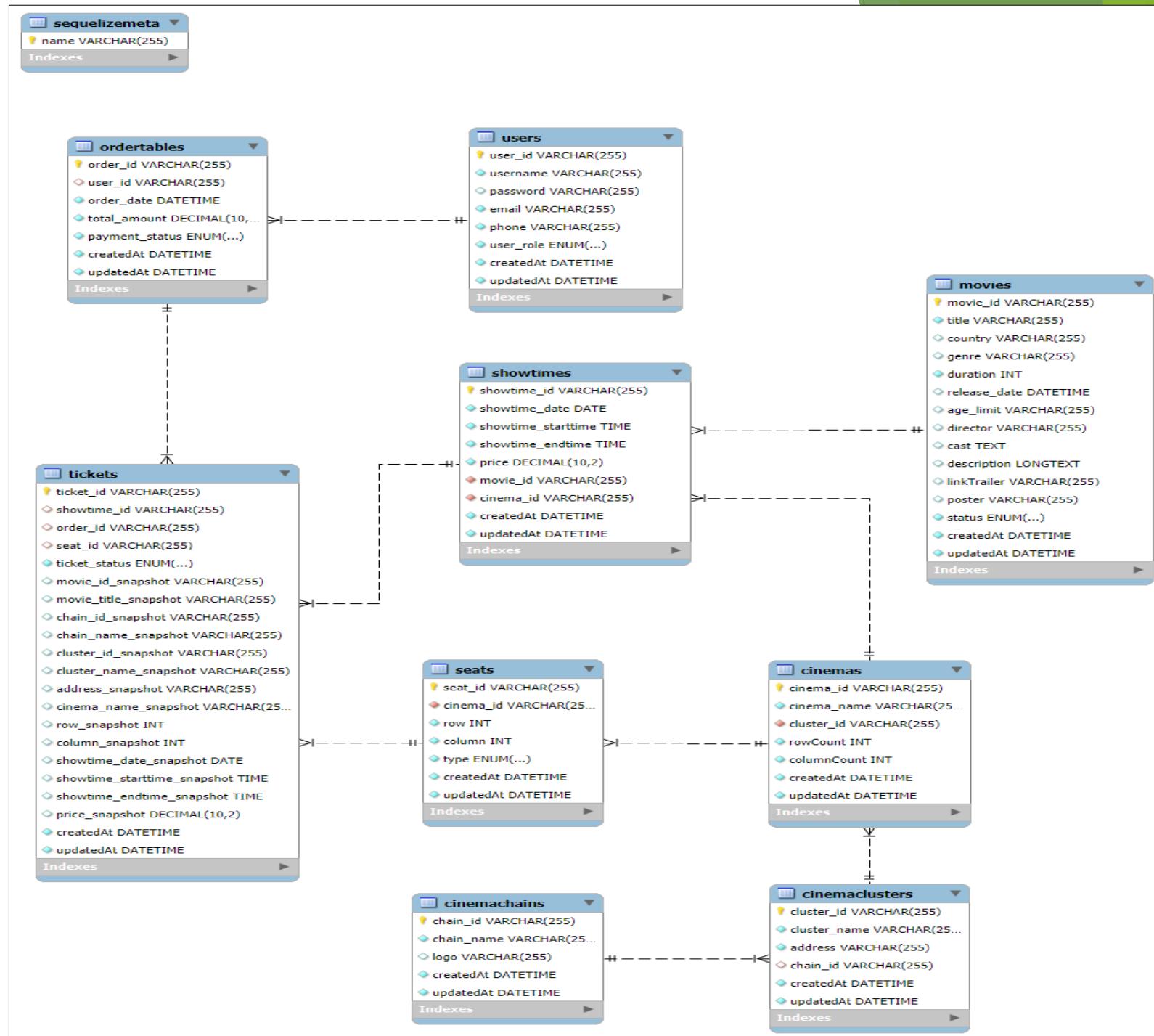
3. THIẾT KẾ HỆ THỐNG

3.2 Mô hình thực thể - quan hệ (ERD)



3. THIẾT KẾ HỆ THỐNG

3.3 Sơ đồ CSDL MySQL



3. THIẾT KẾ HỆ THỐNG

3.4 Thiết kế API

- ❖ Chuẩn: RESTful API theo OpenAPI 3.0
- ❖ Tài liệu hóa: Swagger (/api-docs)
- ❖ Base URL: <http://localhost:5000/api/v1>
- ❖ Xác thực: Bearer Token (JWT) – Header Authorization
- ❖ Định dạng: application/json, riêng upload: multipart/form-data

3. THIẾT KẾ HỆ THÔNG

3.4 Thiết kế API

Nhóm chức năng	Đường dẫn	Thao tác
Authentication	/auth	POST /login, POST /register
Người dùng	/users	GET /, GET /me, PUT /me/profile, PUT /me/password, POST /, PUT /:user_id, DELETE /:user_id, GET /roles
Phim	/movies	GET /, GET /statuses, GET /:id, POST /, PUT /:id, DELETE /:id
Chỗ ngồi	/seats	POST /:cinema_id, GET / (layout)
Chuỗi rạp / cụm rạp / rạp	/cinemachains, /cinemaclusters, /cinemas	GET /, POST /, GET /:id, PUT /:id, DELETE /:id
Suất chiếu	/showtimes	GET /, POST /:movie_id, GET /:id, PUT /:id, DELETE /:id
Vé đã đặt	/users/me/orders	POST tạo đơn, GET lịch sử
Thống kê	/statistics	GET /movies, GET /clusters
Chỉ số tăng giá vé	/configs	GET / (tăng giá ghế VIP)

3. THIẾT KẾ HỆ THỐNG

3.4 Thiết kế API

ion for Cinema Booking System using Node.js, Sequelize, Docker and Swagger

Available authorizations

Value: `DA1WfKMPoWX5ly1D8eBf-Y`

Request URL
`http://localhost:5000/api/v1/users`

Response headers

Code **Details**

200

Response body

```
{
  "err": 0,
  "msg": "OK",
  "response": [
    {
      "count": 7,
      "rows": [
        {
          "user_id": "1814ff4b-2710-4d30-b1ce-34a6a7d77ecf",
          "username": "Nguyễn Hoàng Tuấn",
          "email": "tuanlatoi@gmail.com",
          "phone": "0987013039",
          "user_role": "user",
          "createdAt": "2025-07-17T04:14:05.000Z",
          "updatedAt": "2025-07-17T04:19:03.000Z"
        },
        {
          "user_id": "6de5e761-782c-4e3c-9008-2195c431685c",
          "username": "admin",
          "email": "admin@gmail.com",
          "phone": "038938038",
          "user_role": "admin",
          "createdAt": "2025-07-17T00:52:45.000Z",
          "updatedAt": "2025-07-17T00:52:45.000Z"
        },
        {
          "user_id": "a6910b24-e5fc-44f4-b7e8-45be7bb8aa60",
          "username": "Hoàng Thiên Đức",
          "email": "deptrainhatxom@gmail.com",
          "phone": "0871478995",
          "user_role": "user"
        }
      ]
    }
  ]
}
```

Try it out

Download

3. THIẾT KẾ HỆ THỐNG

Trang chủ

LOGO Trang chủ Lịch chiếu Phim... TÌM KIẾM

Đăng nhập Đăng ký

Trang thêm suất chiếu

Trang thêm cụm rạp

Form Tra

Thống kê

Quản lý khách hàng

Quản lý phim

Quản lý suất chiếu

Quản lý hệ thống rạp

Quản lý cụm rạp

Quản lý rạp

Đăng xuất

Trang chọn ghế

Trang thêm rạp

Thêm cụm rạp

Tên cụm rạp

Địa chỉ

Hệ thống rạp

SUBMIT FORM

Thêm rạp

Tên rạp

Cụm rạp

Số ghế

SUBMIT FORM

Thống kê

Quản lý khách hàng

Quản lý phim

Quản lý suất chiếu

Quản lý hệ thống rạp

Quản lý cụm rạp

Quản lý rạp

Đăng xuất

Chú ý sử dụng thang

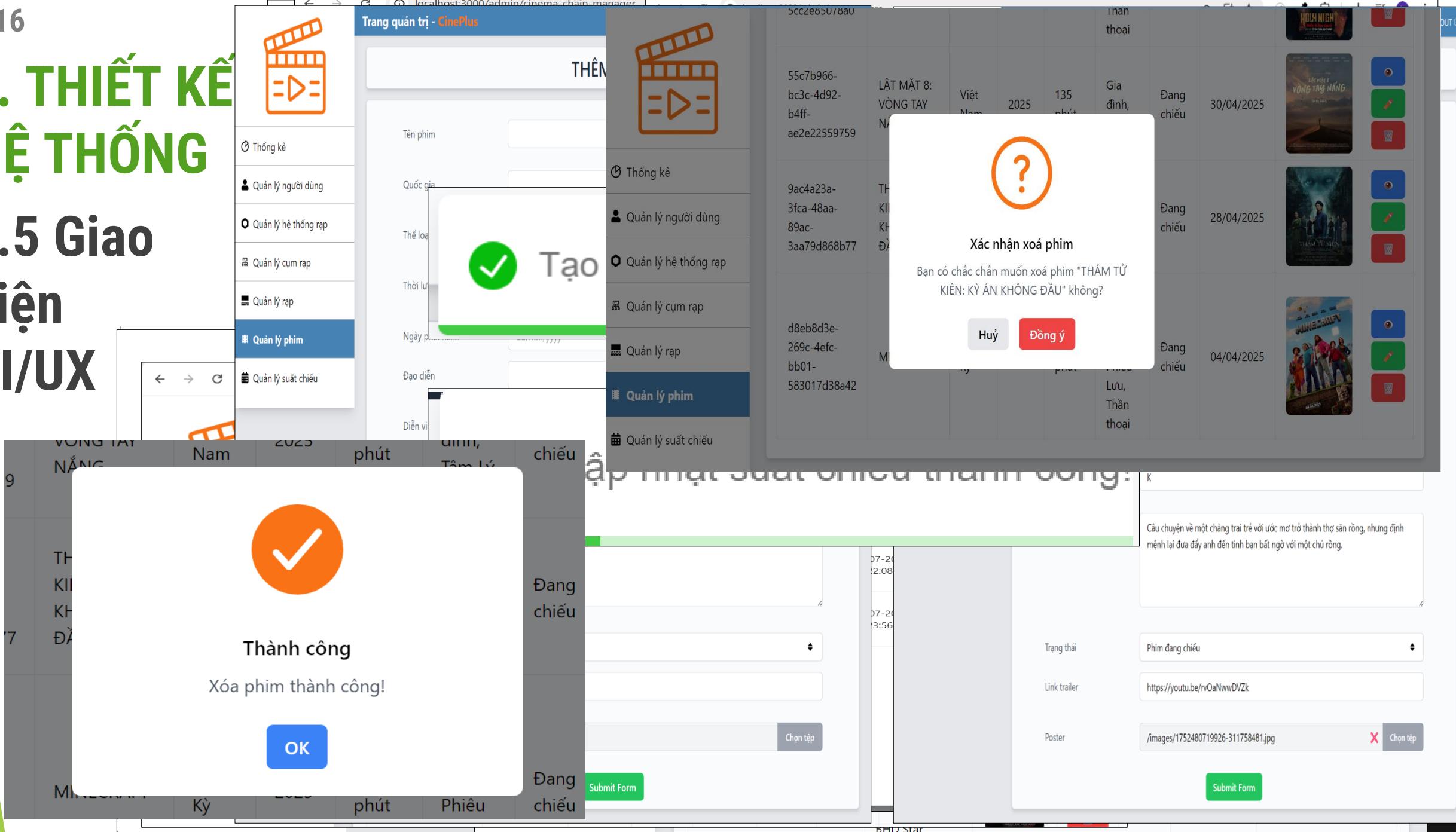
Thường Vip Đã đặt Đang chọn

Thanh toán

© 2025 Cine+, All rights reserved.

3. THIẾT KẾ HỆ THỐNG

3.5 Giao diện UI/UX



4. TRIỂN KHAI & CÔNG NGHỆ SỬ DỤNG

4.1. Công nghệ chính được sử dụng:

- Frontend: ReactJS
- Backend: Node.js, ExpressJS với RESTful API (có kèm Swagger)
- Cơ sở dữ liệu: MySQL + Sequelize
- Kiểm thử và quản lí dự án: Postman, Jira
- Triển khai : Docker
- Quản lý mã nguồn & CI/CD: Git, GitHub, GitHub Actions

5. Quản lý dự án

5.1 Mô hình phát triển phần mềm: Agile

- ❖ Agile/Scrum: linh hoạt, phù hợp nhóm nhỏ
- ❖ Chia sprint theo giai đoạn phát triển

5. Quản lý dự án

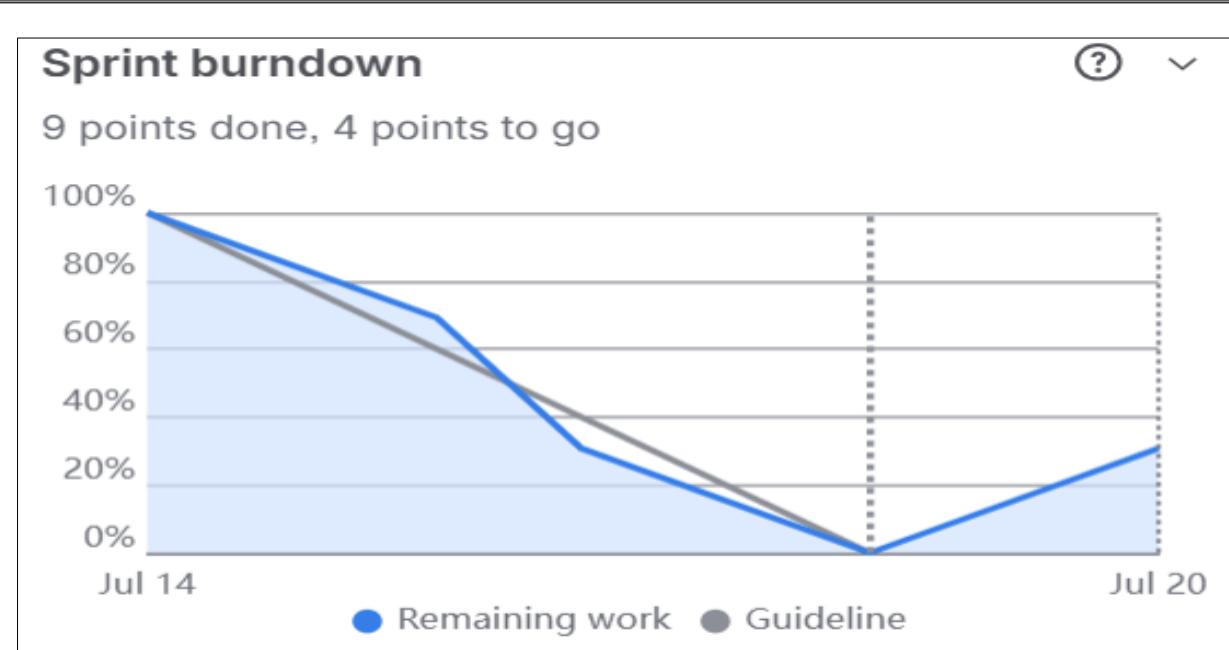
5.2 Các Sprint và Sprint Backlog

MTB Sprint 6 14 Jul – 19 Jul (3 work items)

0 4 9 Complete sprint ...

Chuẩn hóa quá trình phát triển và triển khai bằng cách viết tài liệu API với Swagger, đóng gói hệ thống bằng Docker và tích hợp quy trình CI/CD để tăng tính tự động hóa và độ tin cậy.

MTB-16 Viết tài liệu API với Swagger	MOVIE TICKET BOOK...	DONE	4	NA
MTB-33 Dockerize ứng dụng	MOVIE TICKET BOOK...	DONE	5	L
MTB-34 Tích hợp CI/CD	MOVIE TICKET BOOK...	IN PROGRESS	4	PL



5. Quản lý dự án

5.3 Phân công & kế hoạch: Sprint 1, 2 và 3

Sprint	User Story	Người thực hiện	Point	Sprint Goal
1	Thiết kế trang chủ	Lâm Thanh Đỉnh	3	Hoàn thiện giao diện cơ bản (trang chủ, đăng nhập/đăng ký) và xác thực tài khoản.
	Thiết kế đăng nhập	Nguyễn Hữu Anh	3	
	Phát triển đăng nhập	Phạm Hữu Luân	5	
2	Đổi/đặt lại mật khẩu	Phạm Hữu Luân	5	Hoàn thiện quản lý tài khoản (đổi mật khẩu, chỉnh sửa thông tin cá nhân).
	Thiết kế thông tin	Lâm Thanh Đỉnh	3	
	Cập nhật thông tin	Nguyễn Hữu Anh	3	
3	Thiết kế danh sách	Lâm Thanh Đỉnh	4	Hoàn thiện hiển thị danh sách và chi tiết phim, API quản lý phim.
	Viết API phim	Phạm Hữu Luân	5	
	Hiển thị phim	Lâm Thanh Đỉnh	3	
	Thiết kế chi tiết	Nguyễn Hữu Anh	4	
	Hiển thị chi tiết	Nguyễn Hữu Anh	3	

5. Quản lý dự án

5.3 Phân công & kế hoạch: Sprint 4, 5 và 6

Sprint	User Story	Người thực hiện	Point	Sprint Goal
4	Vết API lịch sử vé	Lâm Thanh Đỉnh	4	Xây dựng đặt vé, xem lịch chiếu, lịch sử vé với giao diện trực quan.
	Xây dựng trang lịch sử vé	Lâm Thanh Đỉnh	4	
	Xây dựng trang lịch chiếu	Nguyễn Hữu Anh	5	
	Viết API lịch chiếu	Phạm Hữu Luân	5	
	Xây dựng trang chọn ghế	Nguyễn Hữu Anh	5	
	Phát triển chức năng chọn ghế để đặt vé	Phạm Hữu Luân	4	
5	Phát triển chức năng tìm kiếm phim	Lâm Thanh Đỉnh	3	Xây dựng đặt vé, lịch chiếu, lịch sử vé với giao diện trực quan.
	Phát triển các chức năng người quản trị	Nguyễn Hữu Anh	5	
	Xây dựng trang quản trị	Phạm Hữu Luân	4	
6	Viết tài liệu API với Swagger	Nguyễn Hữu Anh	4	Chuẩn hóa phát triển với Swagger, Docker, CI/CD để tăng tự động hóa.
	Dockerize ứng dụng	Lâm Thanh Đỉnh	5	
	Tích hợp CI/CD	Phạm Hữu Luân	4	

6.

KẾT QUẢ KIỂM THỬ

HTTP datvephim / user / login

Save Share

PUT DELETE http://localhost:5000/api/v1/cinemachains/_CIAEAdHwBQewgqm6yWkH Send

Params Authorization ● Headers (7) Body Scripts Settings Cookies

Auth Type
Bearer Token

The authorization header will be automatically generated when you send the request. Learn more about [Bearer Token](#) authorization.

Token

Body Cookies Headers (9) Test Results 200 OK 239 ms 351 B Save Response

{ } JSON ▾ Preview ⚡ Visualize

```
1 {  
2   "err": 0,  
3   "msg": "Xóa chuỗi rạp thành công!"  
4 }
```

7. Đánh giá và kết luận

❖ Kết quả đạt được :

- Website đặt vé: đăng nhập, đặt vé, quản lý, thống kê.
- Cron job xóa suất quá hạn.
- UI/UX Figma, React/Tailwind, API RESTful (Postman, Swagger).
- Agile/Scrum, Jira, Git/GitHub.

❖ Khó khăn gặp phải:

- CI/CD chưa xong, thiếu deploy AWS.
- Ít kinh nghiệm full-stack.
- Agile/Jira chưa tối ưu

7. Đánh giá và kết luận

❖ Bài học kinh nghiệm rút ra:

- Nâng cao teamwork, dùng Github, Jira, Postman.
- Thành thạo hơn về Agile, full-stack, API RESTful.
- Hiểu quy trình phát triển.

❖ Đề xuất cải thiện:

- Thêm thanh toán (VNPay, Momo), bình luận, app di động (React Native).
- Áp dụng CI/CD (Docker, GitHub Actions), API quản lý, cải tiến DB (tách enum, tĩnh thành, NoSQL).
- Tối ưu Scrum: chia task nhỏ, dùng Jira/burndown chart theo dõi sát tiến độ.

CẢM ƠN QUÝ THẦY CÔ
ĐÃ LẮNG NGHE!

