

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN TỐT NGHIỆP

ĐỀ TÀI

XÂY DỰNG WEB QUẢN LÝ RẠP CHIẾU PHIM

Giảng viên hướng dẫn : Ths. Vũ Huấn
Sinh viên thực hiện : Vũ Công Đoàn
Mã sinh viên : 191212658
Lớp : CNTT 3
Khóa : K60

Hà Nội – 2023

TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI
KHOA CÔNG NGHỆ THÔNG TIN

ĐỒ ÁN TỐT NGHIỆP

ĐỀ TÀI

XÂY DỰNG WEB QUẢN LÝ RẠP CHIẾU PHIM

Giảng viên hướng dẫn : Ths. Vũ Huân

Sinh viên thực hiện : Vũ Công Đoàn

Mã sinh viên : 191212658

Lớp : CNTT 3

Khóa : K60

Hà Nội – 2023

LỜI CẢM ƠN

Trong thời gian nghiên cứu và hoàn thành đồ án tốt nghiệp, em đã nhận được nhiều sự giúp đỡ, đóng góp ý kiến và chỉ bảo nhiệt tình của thầy cô, gia đình và bạn bè.

Lời đầu tiên em xin gửi lời cảm ơn chân thành nhất đến ThS. Vũ Huân, người đã trực tiếp hướng dẫn, chỉ bảo cũng như luôn quan tâm, động viên trong suốt quá trình làm đồ án để em có thể hoàn thành đồ án tốt nghiệp của mình một cách tốt nhất.

Em cũng xin chân thành cảm ơn các thầy cô giáo trường Đại học Giao Thông Vận Tải đã luôn quan tâm, tạo điều kiện để chúng em được trang bị những kiến thức nền tảng tốt nhất trong suốt quá trình học tập tại trường, là cơ sở lý thuyết vững vàng để từ đó em có thể nghiên cứu và hoàn thành đồ án tốt nghiệp của mình.

Cuối cùng, em xin chân thành cảm ơn gia đình và bạn bè, đã luôn tạo điều kiện, quan tâm, giúp đỡ, động viên em trong suốt quá trình học tập và hoàn thành đồ án tốt nghiệp.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế của một sinh viên, đồ án tốt nghiệp này không thể tránh được những thiếu sót, em rất mong nhận được sự góp ý quý báu của tất cả các thầy cô của em được hoàn thiện hơn.

Em xin chân thành cảm ơn!

MỤC LỤC

LỜI CẢM ƠN.....	4
CHƯƠNG 1: GIỚI THIỆU CHUNG.....	13
1.1 Giới thiệu đề tài.....	13
1.1.1 Tên đề tài.....	13
1.1.2 Mục tiêu đề tài.....	13
1.1.3 Mục tiêu nghiên cứu.....	13
1.2 Cơ sở lý thuyết.....	13
1.2.1 Framework Vue.js.....	13
1.2.2 Vite và Vuex.....	16
1.2.3 ASP.NET WEB API và Dapper.....	17
1.2.3 Cơ sở dữ liệu MySQL.....	19
1.2.5 Cơ sở dữ liệu Firebase.....	20
CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG.....	22
2.1 Phá biểu bài toán.....	22
2.2 Phân tích hệ thống.....	22
2.2.1 Mục đích và phạm vi.....	22
2.2.2 Ai được dùng hệ thống? Làm gì?.....	22
2.2.3 Các đối tượng quản lý.....	23
2.2.4 Quan hệ giữa các đối tượng.....	24
2.3 Biểu đồ use case case và kịch bản chuẩn.....	24
2.3.1 Biểu đồ use case tổng quan.....	24
2.3.2 Use case đăng nhập/đăng ký.....	26
2.3.3 Use case chức năng nhân viên.....	27
2.3.4 Use case Quản lý phim.....	28
2.3.5 Use case Quản lý vé phim.....	29
2.3.6 Use case Quản lý suất chiếu.....	30
2.3.7 Use case Quản lý lịch sử và doanh thu.....	31
2.3.8 Use case Quản lý tài khoản.....	32
2.3.9 Use case Quản lý phòng chiếu.....	34
2.3.10 Use case Quản lý thông tin cá nhân.....	36
2.3.11 Use case Tra cứu lịch chiếu đặt vé.....	37
2.4 Biểu đồ trạng thái.....	38
2.4.1 Đăng nhập/đăng ký.....	38

2.4.2 Quản lý phim.....	39
2.4.3 Quản lý suất chiếu.....	39
2.4.4 Quản lý vé.....	40
2.4.5 Quản lý tài khoản.....	40
2.4.6 Quản lý phòng chiếu.....	41
2.5 Biều đồ giao tiếp.....	41
2.5.1 Đăng nhập/đăng ký.....	41
2.5.2 Quản lý phim.....	43
2.5.3 Quản lý suất chiếu.....	44
2.5.4 Quản lý vé.....	45
2.5.5 Quản lý tài khoản.....	47
2.5.6 Quản lý phòng chiếu.....	48
2.6 Biều đồ tuần tự.....	49
2.6.1 Đăng nhập/ đăng ký.....	49
2.6.2 Quản lý phim.....	51
2.6.3 Quản lý suất chiếu.....	52
2.6.4 Quản lý vé.....	53
2.6.5 Quản lý tài khoản.....	54
2.6.6 Quản lý phòng chiếu.....	55
2.7 Biều đồ lớp.....	56
2.7.1 Mô tả hệ thống.....	56
2.7.2 Trích các danh từ.....	56
2.7.3 Đánh giá và lựa chọn các danh từ làm lớp thực thể hoặc thuộc tính.....	56
2.7.4 Xác định quan hệ số lượng giữa các thực thể.....	56
2.7.5 Vẽ biều đồ lớp.....	57
2.8 Thiết kế biều đồ lớp thực thể.....	57
2.9 Thiết kế CSDL.....	58
2.10 Kết luận chương 2.....	64
CHƯƠNG 3: CÀI ĐẶT HỆ THỐNG.....	65
3.1 Yêu cầu hệ thống.....	65
3.2 Các công cụ lập trình, thư viện hỗ trợ.....	65
3.3 Cài đặt.....	65
3.4 Giao diện hệ thống.....	66
KẾT LUẬN VÀ KIẾN NGHỊ.....	74
Đánh giá kết quả đạt được.....	74
Phương hướng phát triển.....	74

TÀI LIỆU THAM KHẢO.....	75
Tài liệu:	75
Website:	75

DANH MỤC CÁC TỪ VIẾT TẮT

STT	Từ viết tắt	Diễn giải
1	CSDL	Cơ sở dữ liệu
2	ASP	Active Server Pages
3	API	Application Programming Interface
4	DOM	Document Object Model
5	SASS	Syntactically Awesome Style Sheets
6	IDE	Integrated Development Environment
7	REST	Representational State Transfer of Oriented Gradients
8	HTTP	Hyper Text Transfer Protocol
9	UC	Use case

DANH MỤC BẢNG BIỂU

Bảng 3.1. Diễn giải bảng account	60
Bảng 3.2. Diễn giải bảng accounttemp	61
Bảng 3.3. Diễn giải bảng cinematenant.....	61
Bảng 3.4. Diễn giải bảng categorymovie.....	61
Bảng 3.5. Diễn giải bảng dboptions.....	61
Bảng 3.6. Diễn giải bảng history	61
Bảng 3.7. Diễn giải bảng listcinema	62
Bảng 3.8. Diễn giải bảng movie	62
Bảng 3.9. Diễn giải bảng categorymoviedetail.....	62
Bảng 3.10. Diễn giải bảng room	62
Bảng 3.11. Diễn giải bảng roomcinema	63
Bảng 3.12. Diễn giải bảng seat	63
Bảng 3.13. Diễn giải bảng statusroomcinema	63
Bảng 3.14. Diễn giải bảng templateticketmovie.....	63
Bảng 3.15. Diễn giải bảng templatetimemovieon.....	63
Bảng 3.16. Diễn giải bảng ticket.....	64
Bảng 3.17. Diễn giải bảng typemovie.....	64

DANH MỤC HÌNH ẢNH

Hình 1.1. Mô hình MVVM.....	14
Hình 1.2. Vòng đời của một Vue Instance.....	15
Hình 1.3. Sơ đồ làm việc của Vuex	17
Hình 1.4. RESTful API.....	18
Hình 1.5. Cấu trúc của Dapper	19
Hình 2.1. Biểu đồ use case tổng quan.....	25
Hình 2.2. Biểu đồ use case module đăng nhập/đăng ký	26
Hình 2.3. Biểu đồ use case module chức năng nhân viên	27
Hình 2.4. Biểu đồ use case module quản lý phim	28
Hình 2.5. Biểu đồ use case module quản lý vé phim	29
Hình 2.6. Biểu đồ use case module quản lý suất chiếu	30
Hình 2.7. Biểu đồ use case module lịch sử và doanh thu	31
Hình 2.8. Biểu đồ use case module quản lý tài khoản.....	32
Hình 2.9. Biểu đồ use case module quản lý phòng chiếu.....	34
Hình 2.10. Biểu đồ use case module quản lý thông tin cá nhân.....	36
Hình 2.11. Biểu đồ use case module lịch chiếu và đặt vé	37
Hình 2.12. Biểu đồ trạng thái module đăng nhập	38
Hình 2.13. Biểu đồ trạng thái module đăng ký.....	38
Hình 2.14. Biểu đồ trạng thái module quản lý phim	39
Hình 2.15. Biểu đồ trạng thái module quản lý suất chiếu	39
Hình 2.16. Biểu đồ trạng thái module quản lý vé	40
Hình 2.17. Biểu đồ trạng thái module quản lý tài khoản.....	40
Hình 2.18. Biểu đồ trạng thái module quản lý phòng chiếu.....	41
Hình 2.19. Biểu đồ giao tiếp module đăng nhập.....	42
Hình 2.20. Biểu đồ giao tiếp module đăng ký	43
Hình 2.21. Biểu đồ giao tiếp module quản lý phim.....	44
Hình 2.22. Biểu đồ giao tiếp module quản lý suất chiếu.....	45
Hình 2.23. Biểu đồ giao tiếp module quản lý vé	46
Hình 2.24. Biểu đồ giao tiếp module quản lý tài khoản.....	47
Hình 2.25. Biểu đồ giao tiếp module quản lý phòng.....	48
Hình 2.26. Biểu đồ tuần tự module đăng nhập	49
Hình 2.27. Biểu đồ tuần tự module đăng ký	50
Hình 2.28. Biểu đồ tuần tự module quản lý phim	51
Hình 2.29. Biểu đồ tuần tự module quản lý suất chiếu.....	52
Hình 2.30. Biểu đồ tuần tự module quản lý vé.....	53
Hình 2.31. Biểu đồ tuần tự module tài khoản.....	54
Hình 2.32. Biểu đồ tuần tự module quản lý phòng chiếu.....	55
Hình 2.33. Biểu đồ lớp.....	57
Hình 2.34. Biểu đồ lớp thực thể.....	58
Hình 2.35. Biểu đồ CSDL Tài khoản	59

Hình 2.36. Biểu đồ RM CSDL Tài khoản	60
Hình 3.1. Cài đặt NodeJS	65
Hình 3.2. Hình giao diện đăng nhập.....	65
Hình 3.3. Hình giao diện đăng ký	65
Hình 3.4. Hình giao diện quản lý phim	65
Hình 3.5. Hình giao diện thêm mới/sửa phim	65
Hình 3.6. Hình giao diện danh sách suất chiếu	65
Hình 3.7. Hình giao diện chi tiết suất chiếu	65
Hình 3.8. Hình giao diện quản lý vé	65
Hình 3.9. Hình giao diện lịch sử	65
Hình 3.10. Hình giao diện quản lý phòng chiếu	65
Hình 3.11. Hình giao diện chi tiết phòng chiếu	65
Hình 3.12. Hình giao diện danh sách tài khoản	65
Hình 3.13. Hình giao diện chi tiết tài khoản	65
Hình 3.14. Hình giao diện phê duyệt tài khoản.....	65
Hình 3.15. Hình giao diện danh mục	65
Hình 3.16. Hình giao diện tùy chọn.....	65

MỞ ĐẦU

Với sự phát triển vượt bậc của ngành công nghiệp giải trí, rạp chiếu phim trở thành một địa điểm hấp dẫn thu hút rất nhiều khách hàng. Ngày nay, việc quản lý một rạp chiếu phim đòi hỏi sự chuyên nghiệp và hiệu quả trong việc tổ chức các suất chiếu, quản lý vé và tương tác với khách hàng.

Việc xây dựng một hệ thống quản lý rạp chiếu phim trực tuyến sẽ đáp ứng được những yêu cầu trên. Hệ thống này sẽ cung cấp một giao diện trực quan và thuận tiện cho quản lý rạp chiếu phim, từ việc quản lý thông tin về phim, suất chiếu, vé cho đến quản lý khách hàng và nhân viên.

Mục tiêu của dự án này là nghiên cứu và triển khai các công nghệ xây dựng hệ thống quản lý rạp chiếu phim hiện đại. Chúng ta sẽ tìm hiểu và áp dụng những phương pháp phát triển phần mềm tiên tiến để xây dựng một giao diện người dùng thân thiện, cung cấp các chức năng quản lý mạnh mẽ và đáng tin cậy.

Trong đồ án em sẽ tập trung trình bày một số nội dung chính như sau:

- **Chương 1: Giới thiệu chung**
Nội dung chương 1 sẽ giới thiệu khái quát về đề tài, giới thiệu cơ sở lý thuyết để có thể xây dựng một website, giới thiệu các công nghệ được áp dụng để xây dựng website.
- **Chương 2: Phân tích và thiết kế hệ thống**
Nội dung chương 2 là phân tích và làm rõ các yêu cầu về mặt chức năng, thiết kế cơ sở dữ liệu cho hệ thống.
- **Chương 3: Cài đặt hệ thống**
Nội dung chương 3 mô tả các bước cài hệ thống và mô tả giao diện hệ thống sau khi đã xây dựng hoàn chỉnh
- **Phần kết luận:**
Tổng kết lại những kết quả đã đạt được và đưa ra phương hướng phát triển trong tương lai

CHƯƠNG 1: GIỚI THIỆU CHUNG

Chương 1 của đề án gồm 2 nội dung chính:

- Giới thiệu chung về đề tài được chọn
- Trình bày về cơ sở lý thuyết và các công nghệ được sử dụng để xây dựng sản phẩm

1.1 Giới thiệu đề tài

1.1.1 Tên đề tài

Đề tài em lựa chọn cho đề án tốt nghiệp là “Xây dựng website Quản lý Rạp chiếu phim”.

Đề tài bao gồm việc tìm hiểu và nghiên cứu về cơ sở lý thuyết của các công nghệ được áp dụng như các ngôn ngữ lập trình, framework, cơ sở dữ liệu... từ đó xây dựng nên một trang web hoàn chỉnh, bao gồm đầy đủ các tính năng cần thiết của một trang web mua bán linh kiện máy tính.

1.1.2 Mục tiêu đề tài

- Tìm hiểu, nghiên cứu về framework Vue.js và các thư viện hỗ trợ để xây dựng một ứng dụng Vue.js như Vite và Vuex.
- Tìm hiểu về xây dựng web API sử dụng ASP.NET Core và Dapper.
- Áp dụng các công nghệ để xây dựng một website hoàn chỉnh với đầy đủ các tính năng cần thiết.

1.1.3 Mục tiêu nghiên cứu

- Tìm hiểu và nghiên cứu về các công nghệ được sử dụng để tạo ra một website hoàn chỉnh.
- Xây dựng được một website quản lý rạp chiếu phim nhằm giúp các rạp chiếu phim có thể dễ dàng quản lý rạp chiếu phim, lịch chiếu, đặt vé cũng như doanh thu một cách nhanh chóng, tiện lợi.

1.2 Cơ sở lý thuyết

1.2.1 Framework Vue.js

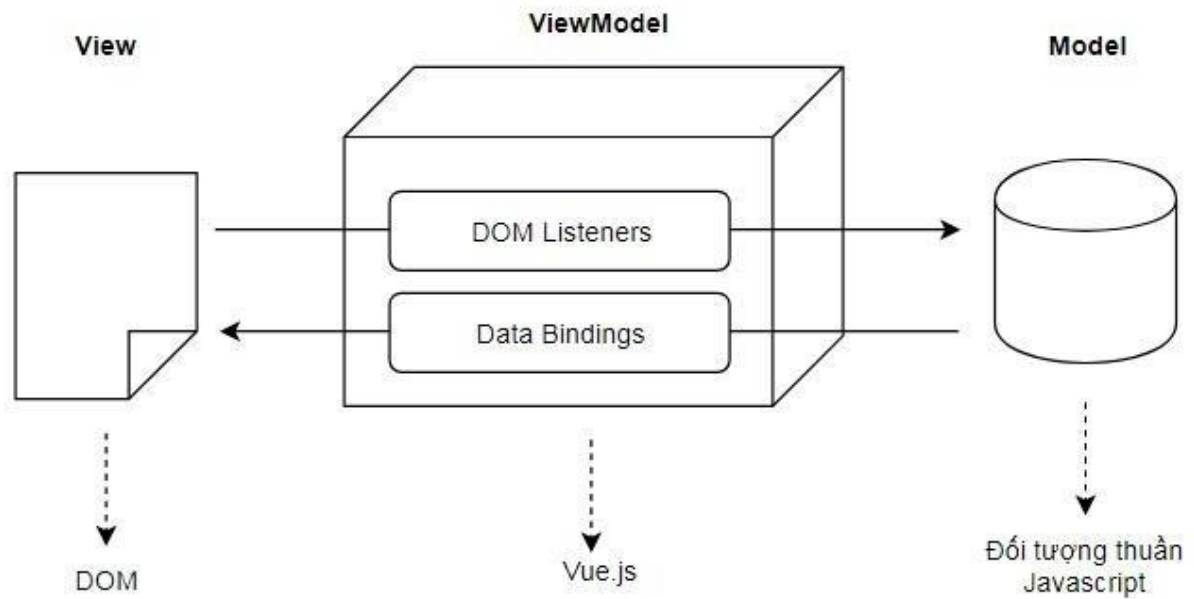
- Giới thiệu chung

Vue là một framework javascript tiên tiến trong xây dựng giao diện người dùng. Không giống như các framework khác, Vue được xây dựng từ những đoạn mã cơ bản nhất nhằm tối ưu tốc độ. Thư viện của Vue chỉ tập trung vào lớp hiển thị, rất đơn giản để tiếp cận và dễ dàng tích hợp vào các hệ thống khác. Vue cũng có khả năng cung cấp các ứng dụng web đơn trang Single Page Application (toàn bộ website chỉ là một trang) cho phép kết hợp với nhiều các công cụ hiện đại, như Laravel chẳng hạn. Vue.js được sử dụng để xây dựng giao diện người dùng giống như React, Angular, Ember... Tuy nhiên, Vue.js có tốc độ tạo trang (render) rất nhanh và chiếm khá ít bộ nhớ.

- Mô hình MVVM

MVVM là viết tắt của Model-View-ViewModel là một mô hình được áp dụng

trong Vue.js.

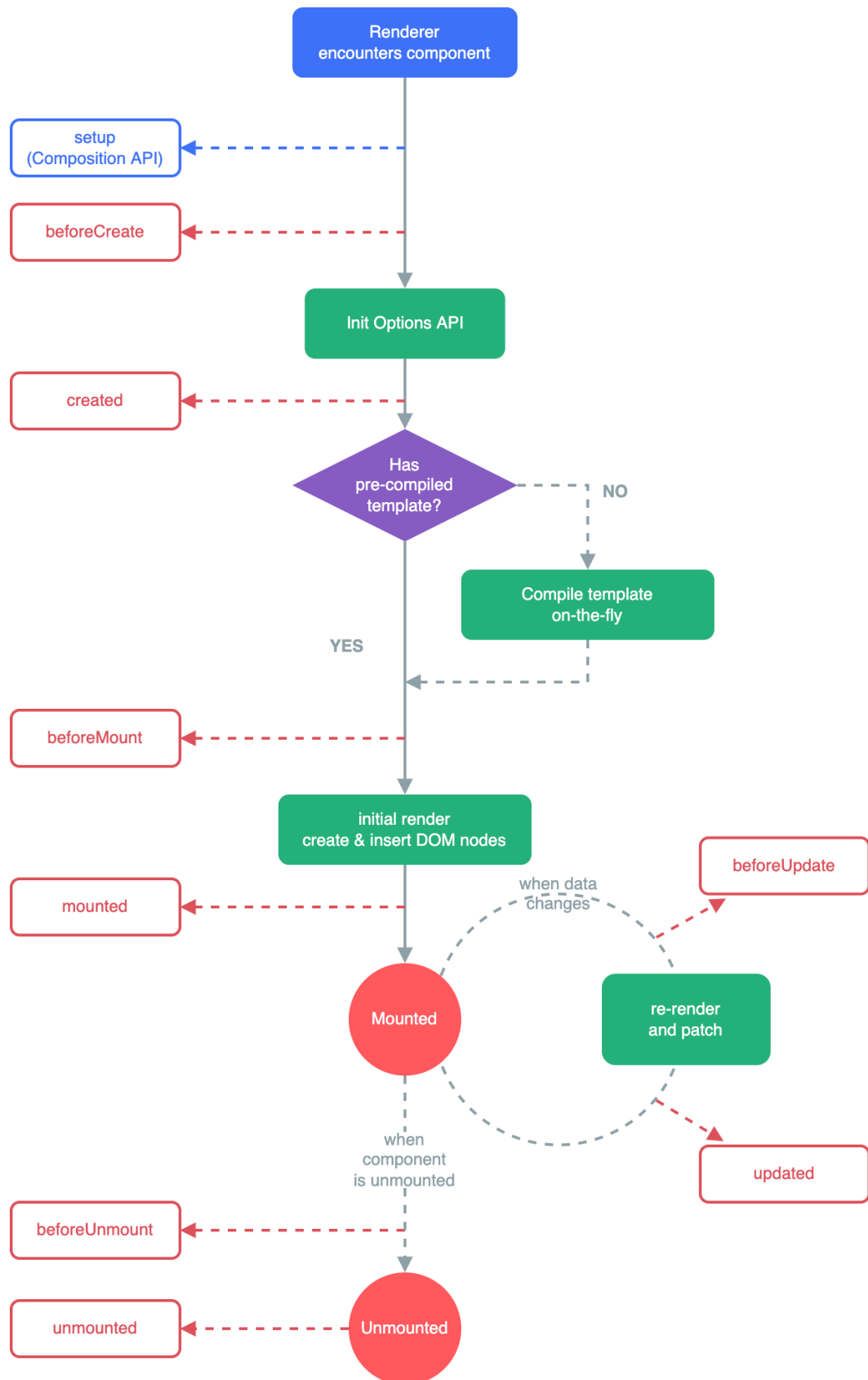


Hình 1.1 Mô hình MVVM

Trong mô hình này, dữ liệu mỗi khi được gán vào View hoặc Model sẽ đều được Vue.js tự động gán cho phần còn lại. Tức là khi dữ liệu thay đổi ở Model nó sẽ tự động được cập nhật sang View. Ngược lại, khi người dùng thay đổi dữ liệu trên View (ví dụ nhập các ký tự vào một ô nhập liệu của thẻ `<input>`) thì dữ liệu cũng được tự động cập nhật sang Model. Mô hình này còn được gọi với một thuật ngữ khác là gán dữ liệu hai chiều (two-way data binding).

- **Vòng đời của một Vue Instance**

Để tiếp cận và sử dụng Vue.js, lập trình viên cần nắm được khái niệm về vòng đời của một instance (hay một đối tượng Vue) từ lúc khởi tạo ứng dụng, trong lúc chạy ứng dụng và tới lúc kết thúc ứng dụng.



Hình 1.2 Vòng đời của một Vue Instance

Sơ đồ trên mô tả vòng đời của một đối tượng, bao gồm các giai đoạn: Khởi tạo đối tượng, Gắn kết DOM, Cập nhật DOM, Hủy bỏ đối tượng.

Ở từng giai đoạn sẽ có các phương thức được thực thi trong mỗi giai đoạn vòng đời,

trong Vue.js gọi là các lifecycle hooks. Tương ứng với các giai đoạn kể trên sẽ có các phương thức:

- `beforeCreate()`: chạy khi khởi tạo đối tượng, tại thời điểm này dữ liệu và các sự kiện chưa được thiết lập.
- `created()`: chạy khi dữ liệu và các sự kiện đã được thiết lập thành công, tại đây đã có thể truy cập tới dữ liệu và các sự kiện.
- `beforeMount()`: chạy ngay sau `created()` và trước khi gắn kết vào DOM. Tại thời điểm này chưa thể truy cập tới các phần tử trong DOM.
- `mounted()`: DOM đã được gắn kết, các phần tử trong DOM đã có thể được truy cập tại thời điểm này.
- `beforeUpdate()`: chạy sau khi có sự thay đổi về mặt dữ liệu và trước khi cập nhật lại giao diện và hiển thị ra cho người dùng.
- `updated()`: chạy ngay sau `beforeUpdate()`, sử dụng khi cần truy cập DOM sau khi thay đổi dữ liệu.
- `beforeDestroy()`: Là giai đoạn trước khi đối tượng bị hủy. Đây là nơi để quản lý tài nguyên, xóa tài nguyên, dọn dẹp các thành phần còn lại.
- `destroyed()`: mọi thành phần đã được hủy bỏ hết.

1.2.2 Vite và Vuex

- Vite

Vite là một công cụ phát triển phần mềm hiệu suất cao và dễ sử dụng, được xây dựng dựa trên JavaScript. Với mục tiêu tăng cường quá trình phát triển ứng dụng web, Vite mang lại sự tối ưu hóa vượt trội và trải nghiệm phát triển đáng kinh ngạc.

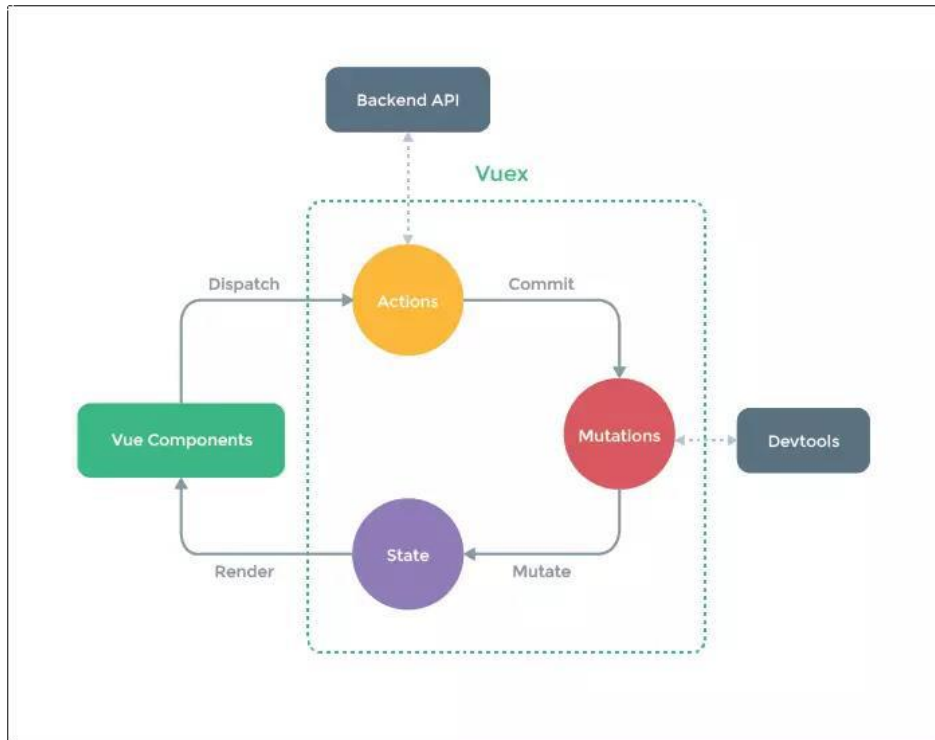
Các tính năng của Vite:

- Quá trình xây dựng ứng dụng trở nên nhanh chóng và hiệu quả hơn
- Xử lý các phụ thuộc và tải các tài nguyên. Thay vì tạo một gói hoàn chỉnh trước khi chạy ứng dụng, Vite sẽ tải các module và tài nguyên cần thiết trực tiếp từ mã nguồn trong quá trình phát triển. Điều này giúp giảm thời gian khởi động và cải thiện trải nghiệm phát triển tức thì
- Hỗ trợ nhiều ngôn ngữ và framework như Vue.js, React, TypeScript và JavaScript thuần
- Cung cấp các công cụ mạnh mẽ như hot module replacement (HMR), giúp thay đổi ngay lập tức mà không cần làm mới trang
- Tạo ra các bundle nhỏ gọn và tối ưu hóa quy trình tải trang

- Vuex

Trong mô hình MVVM có 3 đối tượng là View, Model và ViewModel, cùng với sự tương tác qua lại giữa Model và View khi có sự thay đổi về mặt dữ liệu. Tuy nhiên, khi hệ thống lớn dần lên, các tác động qua lại trở nên phức tạp và đôi khi chỉ cần một thay đổi trên View dẫn đến hàng trăm nghìn các tác động ngược lại Model và

từ đó lại tác động ngược lại View làm hệ thống trở nên khó kiểm soát.



Hình 1.3 Sơ đồ làm việc của Vuex

Vuex được xây dựng dựa trên giải pháp luồng dữ liệu 1 chiều để giải quyết vấn đề này. Luồng dữ liệu sẽ chỉ theo một chiều (one way data flow), khi có một dữ liệu mới, luồng này sẽ bắt đầu lại từ đầu.

1.2.3 ASP.NET WEB API và Dapper

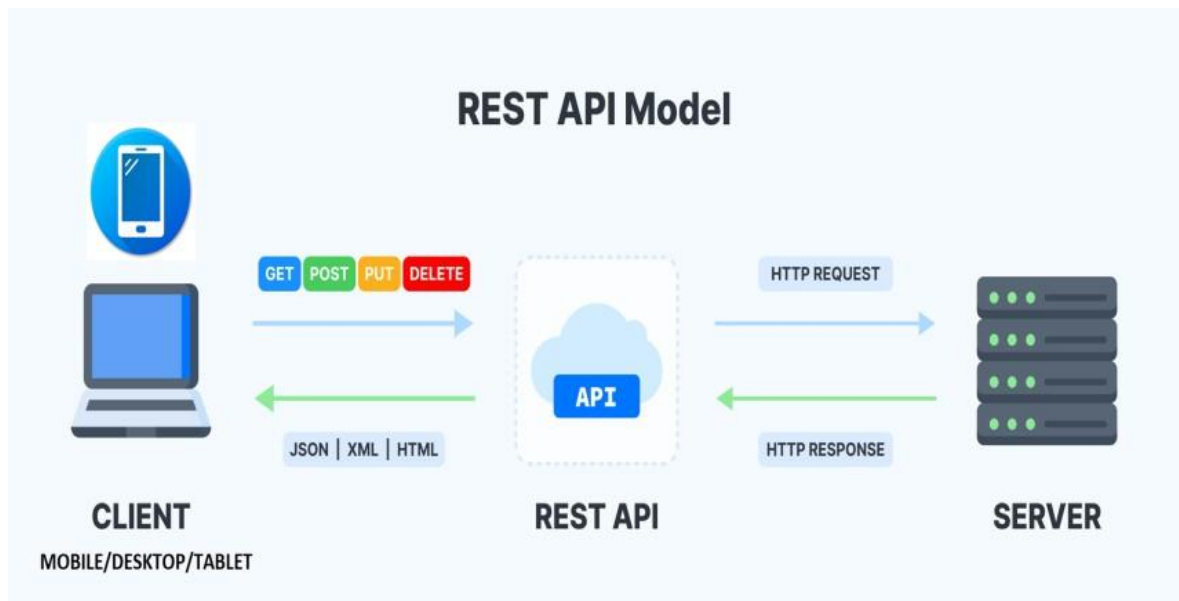
- ASP.NET Web API

Thuật ngữ API là viết tắt của Application Programming Interface - Giao diện lập trình ứng dụng. ASP.NET Web API là một framework được cung cấp bởi Microsoft, giúp dễ dàng xây dựng API Web, tức là các dịch vụ dựa trên giao thức HTTP. Framework này thường được sử dụng để tạo ra các RESTful API.

RESTful API là các API được thiết kế theo cấu trúc REST. REST (REpresentational State Transfer) là một dạng chuyển đổi cấu trúc dữ liệu, một kiểu kiến trúc để viết API. Nó sử dụng phương thức HTTP đơn giản để tạo cho giao tiếp giữa các máy. Vì vậy, thay vì sử dụng một URL cho việc xử lý một số thông tin người dùng, REST gửi một yêu cầu HTTP như GET, POST, DELETE... đến một URL để xử lý dữ liệu.

REST hoạt động chủ yếu dựa vào giao thức HTTP. Các hoạt động cơ bản nêu trên sẽ sử dụng những phương thức HTTP riêng:

- GET (SELECT): Trả về tài nguyên hoặc một danh sách tài nguyên.
- POST (CREATE): Tạo mới một tài nguyên.
- PUT (UPDATE): Cập nhật thông tin cho tài nguyên.
- DELETE (DELETE): Xóa một tài nguyên.



Hình 1.4 RESTful API

Những phương thức hay hoạt động này thường được gọi là CRUD tương ứng với Create, Read, Update, Delete – Tạo, Đọc, Sửa, Xóa.

- **Dapper**

Dapper là một micro ORM (Object-Relational Mapping) mã nguồn mở cho .NET, được thiết kế nhằm cung cấp một cách đơn giản và hiệu quả để truy cập và tương tác với cơ sở dữ liệu.

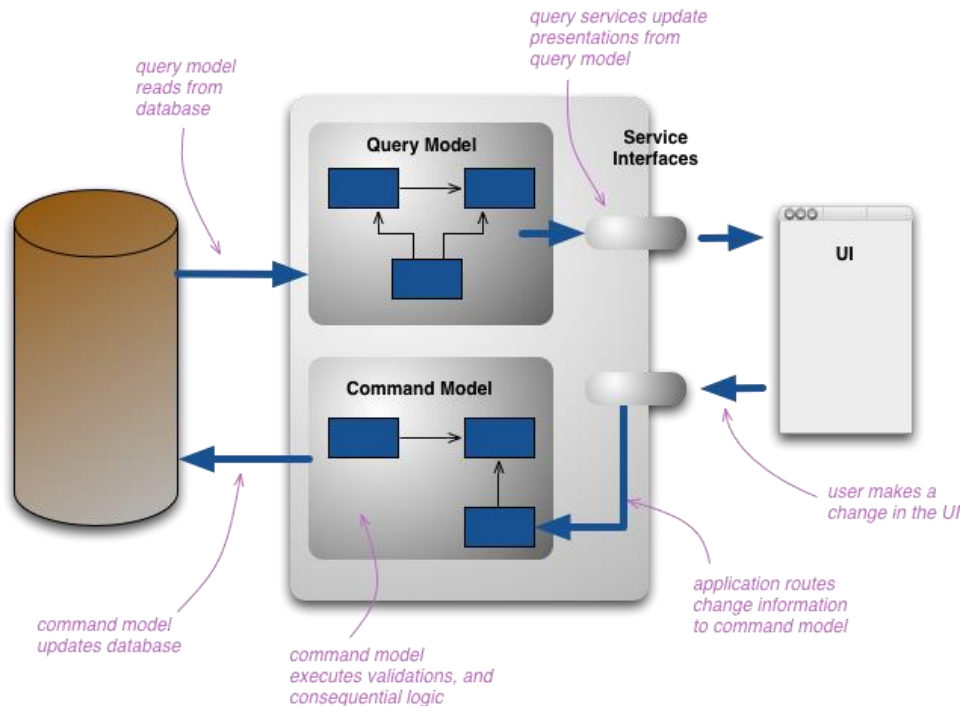
Dapper tập trung vào hiệu năng và tối ưu hóa truy vấn dữ liệu, cho phép bạn thực hiện các truy vấn SQL trực tiếp và ánh xạ kết quả truy vấn thành các đối tượng C#. Với Dapper, bạn có thể viết các truy vấn SQL một cách tự nhiên và linh hoạt, mà không cần phải tạo các lớp đối tượng phức tạp.

Các tính năng của Dapper:

- **Hiệu suất cao:** Dapper được thiết kế để có hiệu suất tối đa. Nó sử dụng cơ chế truy vấn dữ liệu raw và sử dụng ADO.NET để tương tác với cơ sở dữ liệu một cách nhanh chóng và hiệu quả.
- **Dễ sử dụng:** Dapper là một ORM rất đơn giản và dễ sử dụng. Nó không đòi hỏi nhiều cấu hình phức tạp và cho phép bạn viết truy vấn SQL trực tiếp mà không cần phải tạo các tầng trung gian.
- **Tương thích đa nền tảng:** Dapper hoạt động trên nhiều nền tảng, bao gồm SQL Server, MySQL, PostgreSQL, Oracle và SQLite. Điều này cho phép bạn sử dụng Dapper với các hệ quản trị cơ sở dữ liệu khác nhau mà không cần thay đổi mã nguồn.
- **Hỗ trợ LINQ:** Mặc dù Dapper không cung cấp một tầng LINQ to Entities như Entity Framework, nhưng nó hỗ trợ việc sử dụng LINQ để xây dựng các truy vấn SQL. Điều này giúp mã truy vấn dễ đọc và linh hoạt hơn.
- **Mã nguồn mở:** Dapper là một dự án mã nguồn mở, điều này cho phép cộng

đồng đóng góp vào phát triển và nâng cao tính năng của nó.

Cấu trúc Dapper được mô tả đơn giản qua hình sau:



Hình 1.5. Cấu trúc của Dapper

Tóm lại, Dapper là một micro ORM nhẹ nhàng và hiệu quả, giúp đơn giản hóa việc truy cập cơ sở dữ liệu và ánh xạ dữ liệu vào các đối tượng C#. Với tính năng tốc độ và dễ sử dụng, Dapper là một lựa chọn phổ biến cho việc xử lý.

1.2.3 Cơ sở dữ liệu MySQL

MySQL là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là hệ quản trị cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet.

Ưu điểm của MySQL:

- **Dễ sử dụng:** MySQL là cơ sở dữ liệu tốc độ cao, ổn định, dễ sử dụng và hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh.
- **Độ bảo mật cao:** MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên Internet khi sở hữu nhiều nhiều tính năng bảo mật thậm chí là ở cấp cao.
- **Đa tính năng:** MySQL hỗ trợ rất nhiều chức năng SQL được mong chờ từ một hệ quản trị cơ sở dữ liệu quan hệ cả trực tiếp lẫn gián tiếp.
- **Khả năng mở rộng và mạnh mẽ:** MySQL có thể xử lý rất nhiều dữ liệu và hơn thế nữa nó có thể được mở rộng nếu cần thiết.
- **Nhanh chóng:** Việc đưa ra một số tiêu chuẩn cho phép MySQL để làm việc rất hiệu quả và tiết kiệm chi phí, do đó nó làm tăng tốc độ thực thi.

Nhược điểm của MySQL:

- **Giới hạn:** Theo thiết kế, MySQL không có ý định làm tất cả và nó đi kèm với các hạn chế về chức năng mà một vào ứng dụng có thể cần.
- **Độ tin cậy:** Cách các chức năng cụ thể được xử lý với MySQL (ví dụ tài liệu tham khảo, các giao dịch, kiểm toán...) làm cho nó kém tin cậy hơn so với một số hệ quản trị cơ sở dữ liệu quan hệ khác.
- **Dung lượng hạn chế:** Nếu số bản ghi lớn dần lên thì việc truy xuất dữ liệu là khá khó khăn, khi đó cần phải áp dụng nhiều biện pháp để tăng tốc độ truy xuất dữ liệu như là chia tải CSDL ra nhiều máy chủ, hoặc tạo cache MySQL.

1.2.5 Cơ sở dữ liệu Firebase

Firebase Real-time Database là một giải pháp mạnh mẽ để xây dựng các ứng dụng đa người dùng thời gian thực. Với cấu trúc dữ liệu JSON và tính năng đồng bộ tức thì, nó cho phép bạn tạo ứng dụng có khả năng phản hồi nhanh, phát triển dễ dàng và đáng tin cậy.

Ưu điểm của FireBase:

- **Đồng bộ thời gian thực:** Firebase Real-time Database cho phép đồng bộ dữ liệu tức thì giữa các client và máy chủ. Khi dữ liệu thay đổi, nó sẽ tự động được cập nhật trên tất cả các client kết nối, tạo ra trải nghiệm ứng dụng mượt mà và đồng nhất. Selenium Remote Control (RC).
- **Dễ sử dụng và triển khai:** Firebase Real-time Database cung cấp các API đơn giản và dễ sử dụng cho việc truy xuất và cập nhật dữ liệu. Nó cũng cung cấp các thư viện SDK cho nhiều ngôn ngữ lập trình và nền tảng, giúp việc triển khai dễ dàng trên nhiều loại ứng dụng.
- **Offline Persistence:** Firebase hỗ trợ tính năng lưu trữ dữ liệu cục bộ, cho phép ứng dụng hoạt động ngay cả khi không có kết nối Internet. Dữ liệu được lưu trữ trên thiết bị và được đồng bộ lại khi có kết nối mạng
- **Tích hợp dịch vụ Firebase khác:** Firebase Real-time Database có thể dễ dàng tích hợp với các dịch vụ khác của Firebase như xác thực người dùng, lưu trữ đám mây và thông báo đẩy, tạo ra một hệ thống ứng dụng đa chức năng và linh hoạt.

Nhược điểm của Firebase:

- **Hạn chế về truy vấn:** Firebase Real-time Database có hạn chế về khả năng truy vấn và lọc dữ liệu. Nó không hỗ trợ các truy vấn phức tạp và chỉ hỗ trợ các truy vấn đơn giản dựa trên đường dẫn.
- **Khả năng mở rộng có hạn:** Firebase Real-time Database được thiết kế để xử lý các ứng dụng có quy mô nhỏ đến trung bình. Khi ứng dụng của bạn phát triển và có lượng dữ liệu lớn, hiệu suất có thể bị ảnh hưởng và gặp hạn chế về khả năng mở rộng. Trong những trường hợp này, bạn có thể cần xem xét sử dụng các giải pháp cơ sở dữ liệu khác như Firestore của Firebase hoặc các cơ sở dữ liệu truyền thống
- **Không hỗ trợ ACID transactions:** Firebase Real-time Database không hỗ trợ

các giao dịch ACID (Atomicity, Consistency, Isolation, Durability). Điều này có nghĩa là nếu bạn cần thực hiện các thay đổi đồng thời và đảm bảo tính nhất quán của dữ liệu, bạn sẽ cần xử lý các vấn đề về xung đột và đồng bộ hóa dữ liệu một cách tự động

- **Giới hạn về quyền truy cập dữ liệu:** Firebase Real-time Database có hệ thống quyền truy cập dựa trên quy tắc (Security Rules), nhưng nó có thể hạn chế về khả năng xác định các quyền truy cập phức tạp. Điều này có thể làm cho việc thiết lập và quản lý quyền truy cập dữ liệu phức tạp hơn đối với các ứng dụng phức tạp.
- **Tùy chọn lưu trữ dữ liệu có giới hạn:** Firebase Real-time Database cung cấp lưu trữ dữ liệu có giới hạn. Với phiên bản miễn phí của Firebase, bạn có một số giới hạn về dung lượng lưu trữ và băng thông, và phải nâng cấp lên các gói trả phí để mở rộng khả năng lưu trữ.

CHƯƠNG 2: PHÂN TÍCH THIẾT KẾ HỆ THỐNG

2.1 Phá biểu bài toán

Thiết kế một trang web quản lý rạp chiếu phim, cung cấp các chức năng phục vụ cho hai đối tượng chính là admin và người sử dụng (khách hàng). Với admin, hệ thống cần cung cấp các chức năng quản lý lịch chiếu, quản lý vé và quản lý thông tin phim và rạp. Hệ thống cũng nên hỗ trợ cho admin xem thống kê để giúp họ dễ dàng quản lý và vận hành hệ thống. Với người sử dụng, hệ thống cần cung cấp các chức năng cơ bản như tra cứu lịch chiếu, đặt vé, xem thông tin phim và rạp chiếu.

Ngoài ra, hệ thống cần có giao diện đẹp, thân thiện và thuận tiện cho người sử dụng, nhằm tạo trải nghiệm tốt cho họ. Hệ thống cũng cần đảm bảo tốc độ xử lý nhanh chóng và chính xác, đồng thời tương thích với các trình duyệt web khác nhau.

2.2 Phân tích hệ thống

2.2.1 Mục đích và phạm vi

- Mục đích

Mục đích chính của hệ thống web quản lý rạp chiếu phim là cung cấp một nền tảng trực tuyến để quản lý và vận hành hoạt động của rạp chiếu phim. Hệ thống giúp tạo ra một trải nghiệm thuận tiện cho người sử dụng và admin, giúp họ dễ dàng tra cứu thông tin phim, lịch chiếu, đặt vé và quản lý thông tin liên quan đến phim và rạp.

- Phạm vi hệ thống

- Bao gồm:
 - Quản lý lịch chiếu
 - Quản lý vé
 - Quản lý thông tin phim và rạp
 - Quản lý phòng chiếu phim
 - Quản lý tài khoản
 - Khách hàng đặt hàng
 - Giao diện người dùng cung cấp chức năng tra cứu lịch chiếu và đặt vé xem thông tin phim, rạp
- Không bao gồm:
 - Quản lý nhân viên: thông tin của nhân viên, tiền lương, lịch làm việc, chấm công,...
 - Quản lý các thông tin khác về rạp: cơ sở vật chất, diện tích, phương tiện,....

2.2.2 Ai được dùng hệ thống? Làm gì?

- Khách hàng

- Đăng ký tài khoản

- Đăng nhập
 - Tra cứu lịch chiếu: Người sử dụng có thể xem thông tin lịch chiếu của các bộ phim tại rạp chiếu phim
 - Đặt vé: Người sử dụng có thể chọn suất chiếu và đặt vé trực tuyến cho bộ phim mong muốn
 - Xem thông tin phim: Người sử dụng có thể xem thông tin chi tiết về các bộ phim, bao gồm nội dung, diễn viên, đạo diễn và đánh giá
 - Thanh toán
- Quản trị viên
- Quản lý lịch chiếu: Admin có quyền tạo, chỉnh sửa và xóa lịch chiếu các bộ phim, bao gồm thời gian chiếu, phòng chiếu và rạp chiếu phim
 - Quản lý vé: Admin có thể quản lý thông tin vé, bao gồm giá vé, số lượng vé còn lại và các ưu đãi khác.
 - Quản lý thông tin phim và rạp: Admin có quyền cập nhật thông tin về phim, bao gồm nội dung, diễn viên và đạo diễn. Admin cũng có thể quản lý thông tin về rạp chiếu phim, bao gồm địa chỉ, số điện thoại và các thông tin khác
 - Xem thống kê: Admin có thể xem các báo cáo thống kê về doanh thu, số lượng vé bán được và các dữ liệu quan trọng khác để hỗ trợ quản lý và ra quyết định.
 - Xem lịch sử: Admin có thể xem lịch sử đặt vé của rạp chiếu phim, bao gồm thời gian đặt vé, số lượng, giá vé, phim
 - Quản lý tài khoản: Admin có thể xem danh sách tài khoản quản trị, xóa tài khoản quản trị, sửa một số thông tin tài khoản (không bao gồm mật khẩu và tên tài khoản, không thể thay đổi thông tin khách hàng). Thêm tài khoản quản trị mới, phê duyệt tài khoản đăng ký tài khoản nhân viên.
 - Quản lý phòng chiếu: Admin có thể xem thông tin phòng chiếu của rạp, thay đổi tình trạng ghế trong rạp. Thêm phòng chiếu mới, xóa phòng chiếu.
 - Quản lý các tùy chọn chung, tài khoản đang được đăng nhập hiện tại (thay đổi mật khẩu)
 - Phân quyền: Tài khoản quản trị có 3 phân quyền (admin, nhân viên, quản lý). Với mức quyền cao nhất là admin và thấp nhất là nhân viên. Tài khoản nhân viên chỉ có quyền xem thông tin của phim, phòng chiếu, và chức năng đặt vé

2.2.3 Các đối tượng quản lý

- Nhóm thông tin liên quan đến con người
 - User: Tên đăng nhập, email, mật khẩu, họ tên, số điện thoại
 - Khách hàng: Khách hàng sau khi đăng ký tài khoản sẽ trở thành một user của hệ thống.
 - Admin: Là một user có các quyền thao tác đặc biệt nhằm mục đích quản lý hệ thống.
- Nhóm thông tin liên quan đến rạp chiếu

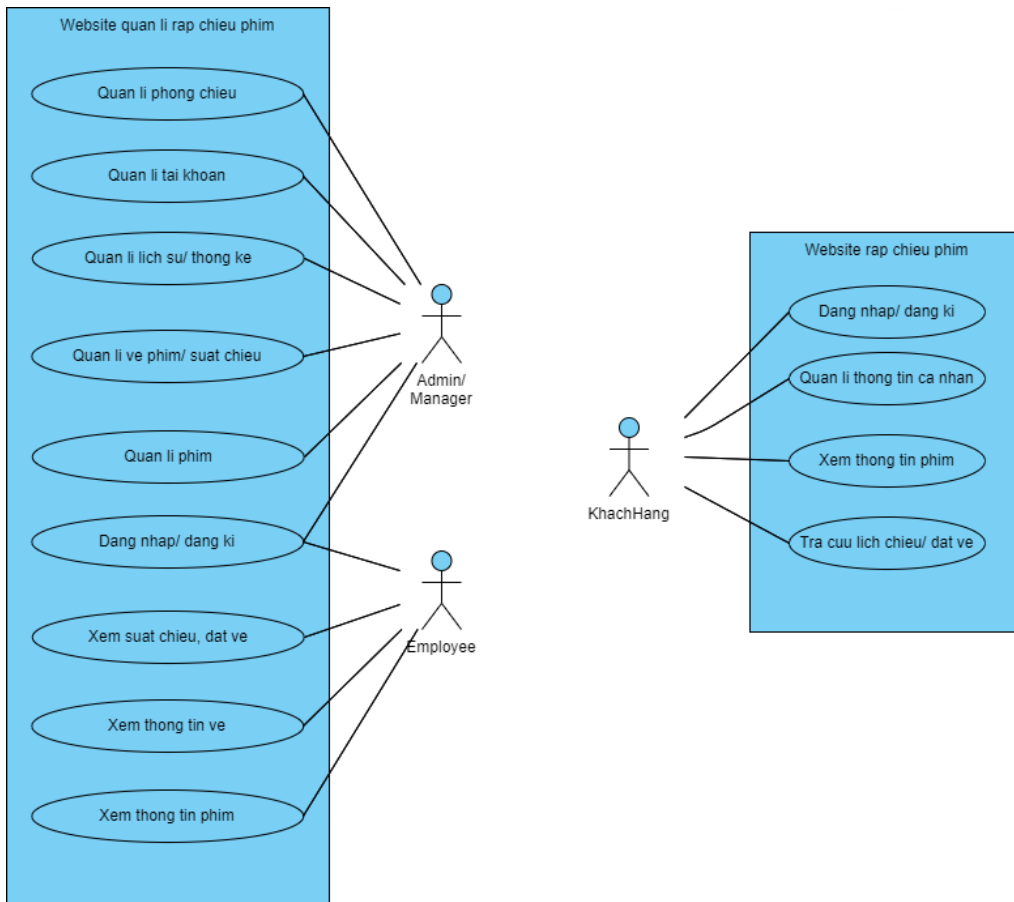
- Phòng chiếu: Tên phòng, số ghế, loại ghế, tình trạng ghế
- Phim: Thông tin về phim, gồm tên phim, thể loại, diễn viên, đạo diễn, thời lượng, mô tả, poster
- Thương hiệu: Tên thương hiệu, logo
- Lịch chiếu: Thông tin phim, thời gian chiếu, giá vé theo ghế
- Rạp chiếu: Thông tin tên rạp, địa chỉ rạp, số điện thoại, giới thiệu.

2.2.4 Quan hệ giữa các đối tượng

- Quan hệ giữa Rạp chiếu và Phòng chiếu:
 - Mỗi Rạp chiếu có thể có nhiều Phòng chiếu.
 - Mỗi Phòng chiếu thuộc về một Rạp chiếu cụ thể.
- Quan hệ giữa Phim và Lịch chiếu:
 - Mỗi Phim có thể có nhiều Lịch chiếu.
 - Mỗi Lịch chiếu liên kết với một Phim cụ thể.
- Quan hệ giữa Lịch chiếu và Phòng chiếu:
 - Mỗi Lịch chiếu sẽ diễn ra trong một Phòng chiếu cụ thể.
 - Mỗi Phòng chiếu có thể có nhiều Lịch chiếu trong ngày khác nhau.
- Quan hệ giữa Lịch chiếu và Vé:
 - Mỗi Lịch chiếu sẽ có một số lượng vé được phân bổ.
 - Mỗi Vé sẽ liên kết với một Lịch chiếu cụ thể.
- Quan hệ giữa Người sử dụng và Đặt vé:
 - Mỗi Người sử dụng có thể Đặt vé cho nhiều Lịch chiếu.
 - Mỗi Lịch chiếu có thể có nhiều Đặt vé từ các Người sử dụng khác nhau.

2.3 Biểu đồ use case case và kịch bản chuẩn

2.3.1 Biểu đồ use case tổng quan



Hình 2.1 Biểu đồ use case tổng quan

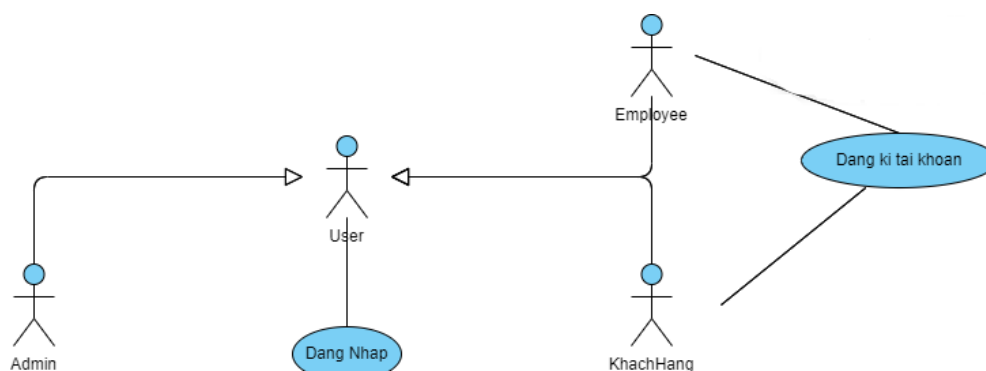
Mô tả use case:

- UC Đăng nhập/đăng ký: Cho phép quản trị viên, nhân viên đăng nhập, đăng kí và trang quản trị. Cho phép người dung đăng nhập đăng kí và trang rạp phim.
- UC Xem thông tin phim, vé, suất chiếu: Cho phép nhân viên xem các thông tin phim, vé, suất chiếu, hiện tại trên website.
- UC Đặt vé: Cho phép nhân viên đặt vé tại suất chiếu.
- UC Quản lý phim: Cho phép quản trị viên thay đổi, thêm mới, xóa thông tin phim.
- UC Quản lý vé phim và suất chiếu: Cho phép quản trị viên thêm suất chiếu, xóa suất chiếu, thay đổi thông tin vé (giá vé) tại suất chiếu.
- UC Lịch sử và thống kê: Cho phép quản trị viên xem báo cáo thống kê doanh thu, lịch sử đặt vé.
- UC Quản lý tài khoản: Cho phép quản trị viên thay đổi một số thông tin tài khoản, xem danh sách tài khoản quản trị (không bao gồm thông tin khách hàng), phê duyệt tài khoản.
- UC Quản lý phòng chiếu: Cho phép quản trị viên thêm, sửa tình trạng, xóa phòng chiếu.
- UC Quản lý thông tin cá nhân: Cho phép khách hàng xem thông tin cá nhân, thay đổi một số thông tin.
- UC Xem thông tin phim: Cho phép khách hàng theo dõi thông tin phim đang được

chiều

- UC Tra cứu lịch chiếu/ đặt vé: Cho phép khách hàng tra cứu lịch chiếu của phim và đặt vé

2.3.2 Use case đăng nhập/đăng ký

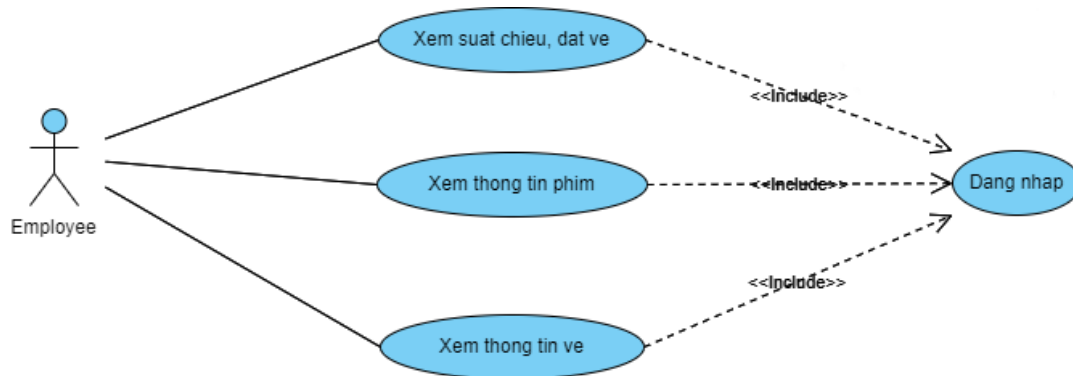


Hình 2.2 Biểu đồ use case module đăng nhập/ đăng ký

- Kịch bản chuẩn

- Đăng nhập:
 1. Người dùng truy cập vào hệ thống quản lí/ trang rạp phim.
 2. Giao diện đăng nhập hiện ra gồm: ô nhập email, tên đăng nhập, ô nhập mật khẩu, nút đăng nhập, nút đăng ký tài khoản.
 3. Người dùng nhập:
 - Email/ Tên đăng nhập: user112@gmail.com
 - Mật khẩu: 123@Abc
 4. Người dùng click đăng nhập
 5. Đăng nhập thành công, giao diện trang quản trị/ trang rạp phim hiện ra.
- Đăng ký:
 1. Nhân viên hoặc người dùng truy cập vào hệ thống.
 2. Giao diện đăng nhập hiện ra gồm: ô nhập email, ô nhập mật khẩu, nút đăng nhập, nút đăng ký tài khoản.
 3. Người dùng click đăng ký tài khoản.
 4. Giao diện đăng nhập hiện ra gồm: ô nhập họ tên, ô nhập email, ô nhập mật khẩu, ô nhập lại mật khẩu, nút đăng ký.
 5. Người dùng nhập:
 - Họ tên: Nguyễn Tuấn
 - Email/ Tên đăng nhập: user112@gmail.com
 - Mật khẩu: 123@Abc
 - Nhập lại mật khẩu: 123@Abc
 6. Người dùng click đăng ký
 7. Thông báo thành công (với nhân viên – thông báo cần được quản trị phê duyệt), giao diện đăng nhập hiện ra để người dùng đăng nhập vào hệ thống.

2.3.3 Use case chức năng nhân viên

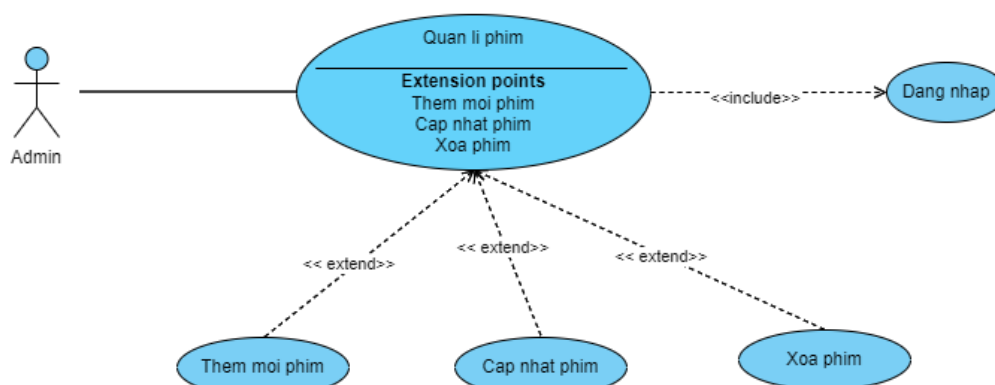


Hình 2.3 Biểu đồ use case module chức năng nhân viên

- Kịch bản chuẩn

1. Nhân viên đăng nhập vào hệ thống
2. Giao diện trang chủ quản trị hiện ra bao gồm danh sách thông tin phim đang chiếu
3. Nhân viên chọn vào quản lý suất chiếu
4. Giao diện quản lý suất chiếu hiện ra bao gồm danh sách phim đang chiếu
5. Nhân viên chọn một phim
6. Giao diện danh sách suất chiếu của phim hiện ra
7. Nhân viên chọn một suất chiếu
8. Giao diện chọn ghế (vé) của suất chiếu hiện ra
9. Nhân viên chọn khách hàng mua vé tại ô danh sách khách hàng
10. Nhân viên chọn ghế, click vào ô đặt vé
11. Thông báo đặt vé thành công hiện ra
12. Nhân viên chọn quản lý vé
13. Danh sách thông tin vé hiện ra

2.3.4 Use case Quản lý phim



Hình 2.4 Biểu đồ use case module quản lý phim

- Thêm mới phim:

1. Admin đăng nhập vào hệ thống.
2. Giao diện trang chủ của admin hiện ra, gồm các lựa chọn: quản lý phim, quản lý suất chiếu v.v
3. Admin chọn Quản lý phim.
4. Giao diện danh sách phim hiện ra
5. Admin click thêm mới.
6. Giao diện nhập liệu thông tin phim hiện ra gồm các thông tin: mã phim, tên phim, đạo diễn, diễn viên, ngày chiếu, ngày kết thúc, ngày xuất bản, thời lượng, nội dung, mô tả, poster, ngôn ngữ
7. Admin nhập đầy đủ các thông tin và nhấn Lưu
8. Thông báo lưu thành công, trở về giao diện danh sách phim.

- Cập nhật phim:

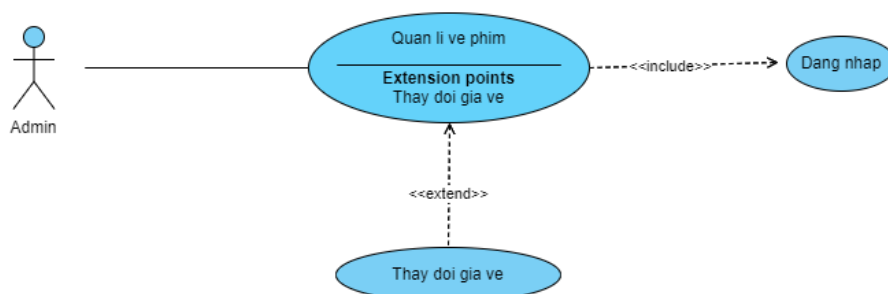
1. Admin đăng nhập vào hệ thống.
2. Giao diện trang chủ của admin hiện ra, gồm các lựa chọn: quản lý phim, quản lý suất chiếu v.v
3. Admin chọn Quản lý phim.
4. Giao diện danh sách phim hiện ra
5. Admin click vào mũi tên hướng xuống ở mỗi ô thông tin phim.
6. Các nút sửa, xóa hiện ra
7. Admin click vào nút sửa hình bút chì
8. Giao diện thông tin phim hiện ra gồm các thông tin: : mã phim, tên phim, đạo diễn, diễn viên, ngày chiếu, ngày kết thúc, ngày xuất bản, thời lượng, nội dung, mô tả, poster, ngôn ngữ

9. Admin sửa một số thông tin và nhấn Lưu
10. Thông báo lưu thành công, trở về giao diện danh sách phim.

- **Xoá phim:**

1. Admin đăng nhập vào hệ thống.
2. Giao diện trang chủ của admin hiện ra, gồm các lựa chọn: quản lý phim, quản lý suất chiếu v.v.
3. Admin chọn Quản lý phim.
4. Giao diện danh sách phim hiện ra
5. Admin chọn mũi tên hướng xuống ở một ô thông tin phim
6. Nút sửa và xóa phim hiện ra
7. Admin click vào nút xóa hình thùng rác
8. Hiện thị thông báo “Bạn có chắc muốn xóa phim này không?”
9. Admin click Đồng ý.
10. Thông báo xóa thành công, trở về giao diện danh sách phim.

2.3.5 Use case Quản lý vé phim

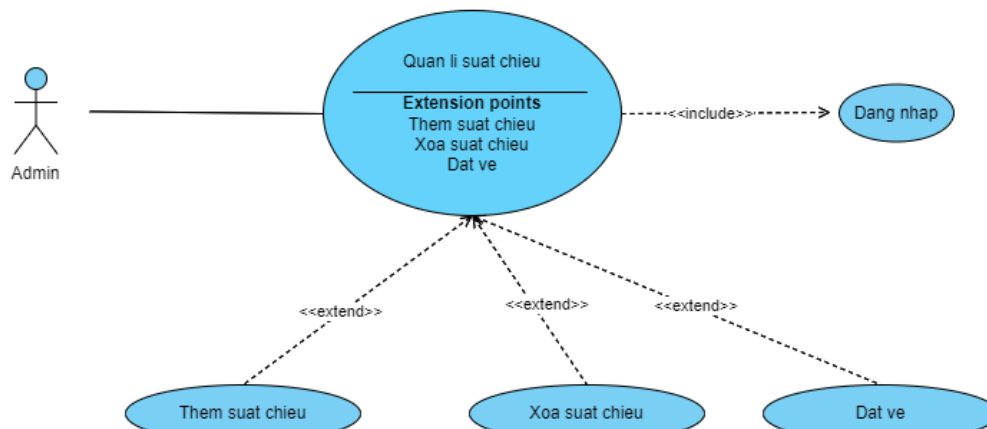


Hình 2.5 Biểu đồ use case module quản lý vé phim

- **Kịch bản chuẩn**

1. Admin đăng nhập vào hệ thống.
2. Giao diện trang chủ của admin hiện ra, gồm các lựa chọn: quản lý phim, quản lý vé v.v
3. Admin chọn Quản lý vé.
4. Giao diện danh sách vé hiện ra.
5. Admin chọn vé muốn thay đổi giá
6. Admin click và nút đổi giá
7. Ô nhập giá hiện ra
8. Admin thay đổi giá và ô nhập giá
9. Admin ấn lưu
10. Thông báo thay đổi giá vé thành công, trở về giao diện quản lý vé

2.3.6 Use case Quản lý suất chiếu



Hình 2.6 Biểu đồ use case module quản lý suất chiếu

- Kịch bản chuẩn

- **Thêm suất chiếu**

1. Admin đăng nhập vào hệ thống.
2. Giao diện trang chủ hiện ra.
3. Admin chọn quản lý suất chiếu.
4. Giao diện quản lý suất chiếu hiện ra danh sách phim đang chiếu.
5. Admin click vào một phim.
6. Nút Thêm suất chiếu hiện ra.
7. Admin click và nút Thêm suất chiếu
8. Giao diện thêm suất chiếu hiện ra bao gồm: Chọn ngày chiếu, chọn phòng chiếu
9. Admin chọn ngày chiếu và phòng chiếu
10. Ô chọn khung giờ chiếu hiện ra
11. Admin chọn khung giờ chiếu
12. Admin ấn Lưu
13. Thông báo Thêm khung giờ chiếu thành công và trở lại trang quản lý suất chiếu

- **Xóa suất chiếu**

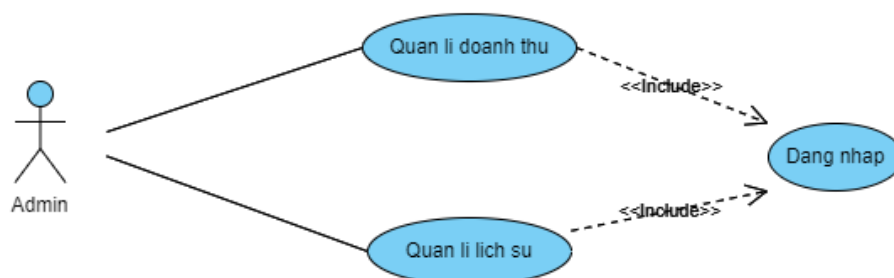
1. Admin đăng nhập vào hệ thống
2. Giao diện trang chủ hiện ra
3. Admin chọn quản lý suất chiếu
4. Giao diện quản lý suất chiếu hiện ra danh sách phim đang chiếu.
5. Admin click vào một phim
6. Nút Xem suất chiếu hiện ra
7. Admin click vào nút xem suất chiếu

8. Giao diện danh sách suất chiếu hiện ra
9. Admin click chọn suất chiếu
10. Giao diện chọn ghế của suất chiếu hiện ra
11. Admin click vào nút xóa phòng
12. Cảnh báo “Bạn có muốn xóa phòng chiếu” hiện ra
13. Admin click vào nút đồng ý
14. Thông báo xóa thành công hiện ra và trở về trang quản lí suất chiếu

- **Đặt vé**

1. Admin đăng nhập vào hệ thống
2. Giao diện trang chủ hệ thống hiện ra
3. Admin chọn quản lí suất chiếu
4. Giao diện quản lí suất chiếu hiện ra với danh sách phim
5. Admin click chọn một phim
6. Nút xem suất chiếu hiện ra
7. Admin chọn xem suất chiếu
8. Giao diện danh sách suất chiếu hiện ra
9. Admin chọn một suất chiếu
10. Giao diện chọn ghế suất chiếu hiện ra
11. Admin chọn ghế cho khách hàng
12. Admin chọn khách hàng mua vé tại danh sách khách hàng
13. Admin ấn đồng ý
14. Thông báo đặt vé thành công
15. Giao diện trở lại màn hình chọn ghế

2.3.7 Use case Quản lý lịch sử và doanh thu



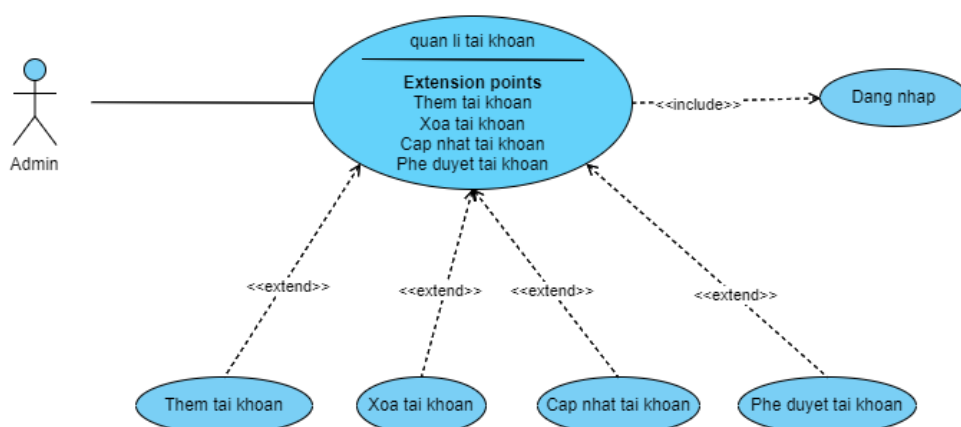
Hình 2.7 Biểu đồ use case module lịch sử và doanh thu

- **Kịch bản chuẩn**

1. Admin đăng nhập vào hệ thống

2. Giao diện trang chủ hiện ra.
3. Admin chọn quản lí doanh thu
4. Giao diện quản lí doanh thu hiện ra
5. Giao diện các biểu đồ doanh thu theo ngày, tháng, năm hiện ra
6. Admin chọn quản lí lịch sử
7. Giao diện quản lí lịch sử hiện ra.
8. Giao diện danh sách các dòng lịch sử đặt vé hiện ra

2.3.8 Use case Quản lý tài khoản



Hình 2.8 Biểu đồ use case module quản lý tài khoản

- Kịch bản chuẩn

• Thêm tài khoản

1. Admin đăng nhập vào hệ thống
2. Giao diện trang chủ quản trị hiện ra
3. Admin click vào quản lý tài khoản
4. Giao diện quản lý tài khoản hiện ra bao gồm danh sách tài khoản quản trị với tài khoản khách hàng
5. Click vào nút thêm mới
6. Giao diện thêm mới tài khoản hiện ra bao gồm: tài khoản, mật khẩu, tên, email, ngày sinh, giới tính, điện thoại, quyền, rap phim, ảnh đại diện
7. Điền đầy đủ các thông tin
8. Nhấn lưu
9. Thông báo thêm mới tài khoản thành công và trở về trang quản lý tài khoản

• Xóa tài khoản

1. Admin đăng nhập vào hệ thống
2. Giao diện trang chủ hiện ra
3. Admin chọn quản lý tài khoản
4. Giao diện quản lý tài khoản hiện ra bao gồm danh sách tài khoản quản trị với tài

khoản khách hàng

5. Admin click vào nút xóa trên ô tài khoản quản trị muốn xóa
6. Thông báo bạn có chắc chắn muốn xóa tài khoản hiện ra
7. Admin click đồng ý
8. Thông báo xóa thành công hiện ra và trở về trang quản lí tài khoản

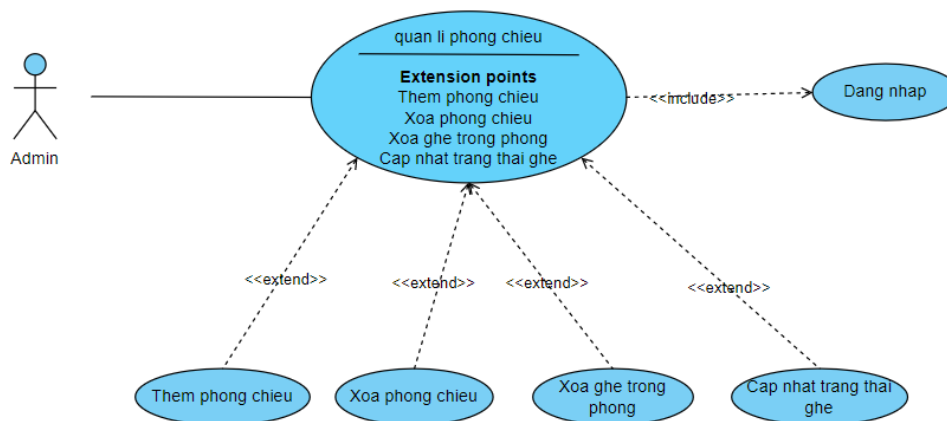
- **Cập nhật tài khoản**

1. Admin đăng nhập vào hệ thống
2. Giao diện trang chủ hiện ra
3. Admin chọn quản lí tài khoản
4. Giao diện quản lí tài khoản hiện ra bao gồm danh sách tài khoản quản trị với tài khoản khách hàng
5. Admin click vào nút chi tiết trên ô tài khoản muốn cập nhật
6. Giao diện thay đổi tài khoản hiện ra bao gồm: tài khoản, mật khẩu, email, ngày sinh, giới tính, điện thoại, quyền, rập phim hiện ra
7. Admin điền thay đổi thông tin được phép thay đổi (trừ tên đăng nhập, mật khẩu)
8. Ấn lưu sau khi điền đủ thông tin
9. Thông báo cập nhật thành công hiện ra và trở về trang quản lí tài khoản

- **Phê duyệt tài khoản**

1. Admin đăng nhập vào hệ thống
2. Giao diện trang chủ hiện ra
3. Admin chọn quản lí tài khoản
4. Giao diện tài khoản hiển ra bao gồm danh sách tài khoản quản trị và tài khoản khách hàng
5. Admin click vào ô phê duyệt tài khoản
6. Danh sách tài khoản yêu cầu phê duyệt hiện ra
7. Admin chọn tài khoản cần phê duyệt nhấn vào nút phê duyệt
8. Thông báo phê duyệt thành công hiện ra và trở về trang quản lí tài khoản

2.3.9 Use case Quản lý phòng chiếu



Hình 2.9 Biểu đồ use case module phòng chiếu

- Kịch bản chuẩn

- **Thêm tài khoản**

1. Admin đăng nhập vào hệ thống
2. Giao diện trang chủ hệ thống hiện ra
3. Admin chọn quản lý phòng chiếu
4. Giao diện quản lý phòng chiếu hiện ra bao gồm danh sách các phòng chiếu
5. Admin chọn Nút thêm mới
6. Giao diện thêm mới phòng chiếu hiện ra bao gồm các thông tin: Tên phòng chiếu, số ghế hàng ngang và dọc
7. Admin điền đầy đủ thông tin
8. Admin click vào nút lưu
9. Thông báo thêm phòng chiếu thành công hiện ra và trở về trang quản lý phòng chiếu

- **Xóa phòng chiếu**

1. Admin đăng nhập vào hệ thống
2. Giao diện trang chủ hệ thống hiện ra
3. Admin chọn quản lý phòng chiếu
4. Giao diện quản lý phòng chiếu hiện ra bao gồm danh sách các phòng chiếu
5. Admin click vào phòng chiếu
6. Giao diện khung phòng chiếu hiện ra bao gồm danh sách vị trí, tình trạng, loại ghế hiện ra
7. Admin click chọn xóa phòng
8. Thông báo bạn có muốn xóa phòng hiện ra

9. Admin click đồng ý
10. Thông báo xóa thành công và trở lại trang quản lí phòng chiếu

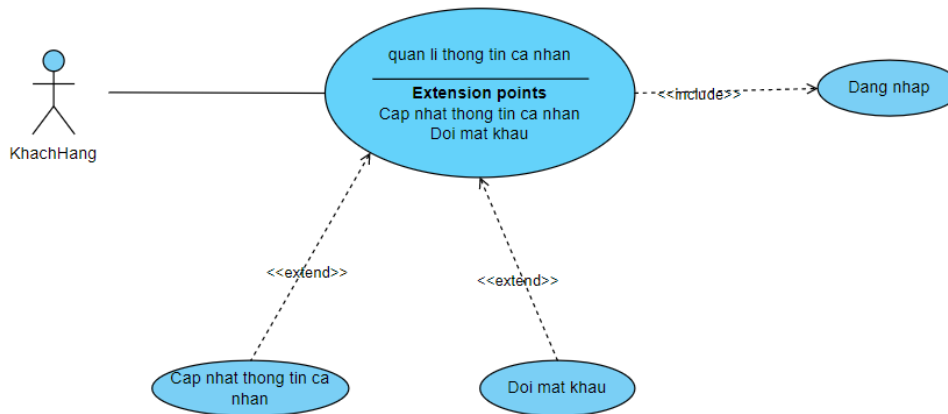
- **Xóa ghế trong phòng chiếu**

1. Admin đăng nhập vào hệ thống
2. Giao diện trang chủ hệ thống hiện ra
3. Admin chọn quản lí phòng chiếu
4. Giao diện quản lí phòng chiếu hiện ra bao gồm danh sách phòng chiếu
5. Admin click chọn vào phòng chiếu
6. Giao diện khung phòng chiếu hiện ra bao gồm danh sách, vị trí, tình trạng, loại ghế hiện ra
7. Admin click chọn những ghế muốn xóa ra khỏi khung phòng chiếu
8. Thông báo bạn có muốn xóa hiện ra
9. Admin click và đồng ý
10. Thông báo xóa ghế thành công và trở lại giao diện khung phòng chiếu đang được chọn

- **Cập nhật ghế trong phòng chiếu**

1. Admin đăng nhập vào hệ thống
2. Giao diện trang chủ hệ thống hiện ra
3. Admin chọn quản lí phòng chiếu
4. Giao diện quản lí phòng chiếu hiện ra
5. Admin chọn phòng chiếu cụ thể
6. Giao diện khung phòng chiếu hiện ra bao gồm danh sách ghế trong phòng chiếu được sắp xếp theo vị trí, loại ghế, tình trạng.
7. Admin chọn ghế muốn cập nhật
8. Admin click vào nút Cập nhật thường, Cập nhật VIP để cập nhật loại ghế, click vào nút bảo trì để cập nhật tình trạng ghế
9. Thông báo cập nhật thành công và trở về giao diện khung phòng chiếu đang được chọn

2.3.10 Use case Quản lý thông tin cá nhân



Hình 2.10 Biểu đồ use case module quản lý thông tin cá nhân

- Kịch bản chuẩn

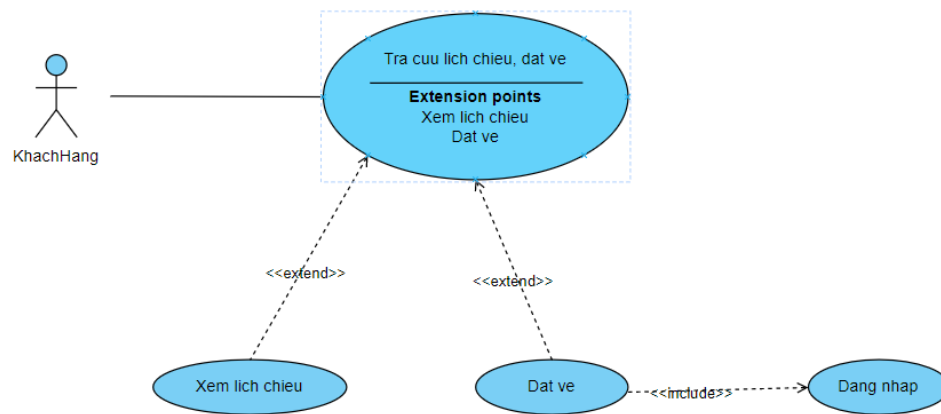
- **Cập nhật thông tin cá nhân**

1. Khách hàng đăng nhập vào trang rạp phim
2. Trang chủ rạp phim hiện ra
3. Khách hàng nhấn vào nút thông tin tài khoản
4. Thông tin tài khoản hiện ra bao gồm: Họ tên, số điện thoại, email, ngày sinh, giới tính, địa chỉ
5. Khách hàng cập nhật các trường thông tin (trừ email)
6. Sau khi điền đủ thông tin khách hàng click vào lưu
7. Thông báo lưu thành công

- **Đổi mật khẩu**

1. Khách hàng đăng nhập vào rạp phim
2. Trang chủ rạp phim hiện ra
3. Khách hàng nhấn vào nút thông tin tài khoản
4. Thông tin tài khoản hiện ra
5. Khách hàng nhấn vào nút đổi mật khẩu
6. Khách hàng nhập mật khẩu cũ và nhập mật khẩu mới và nhập lại mật khẩu mới
7. Sau khi điền đủ thông tin
8. Khách hàng nhấn đồng ý
9. Thông báo đổi mật khẩu thành công

2.3.11 Use case Tra cứu lịch chiếu đặt vé



Hình 2.11 Biểu đồ use case module lịch chiếu, đặt vé

- Kịch bản chuẩn

- **Xem lịch chiếu**

1. Khách hàng vào trang rạp phim
2. Giao diện trang chủ hiện ra.
3. Khách hàng chọn phim đang chiếu.
4. Danh sách ngày chiếu hiện ra.
5. Khách hàng chọn ngày chiếu
6. Danh sách khung giờ chiếu hiện ra

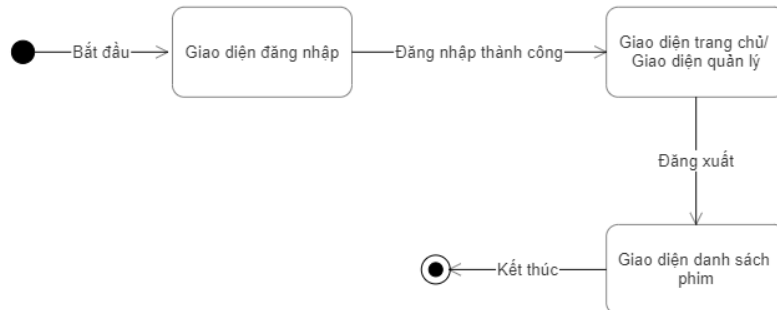
- **Đặt vé**

1. Khách hàng đăng nhập vào trang rạp phim
2. Trang chủ rạp phim hiện ra
3. Khách hàng chọn phim đang chiếu
4. Danh sách ngày chiếu hiện ra
5. Khách hàng chọn ngày chiếu
6. Danh sách khung giờ chiếu hiện ra
7. Khách hàng chọn khung giờ chiếu
8. Danh sách khung phòng chiếu hiện ra bao gồm: Vị trí ghế, tình trạng ghế (đã mua, chưa mua)
9. Khách hàng chọn ghế chưa mua
10. Khách hàng nhấn vào đặt mua
11. Khách hàng thanh toán
12. Thông báo đặt vé thành công

2.4 Biểu đồ trạng thái

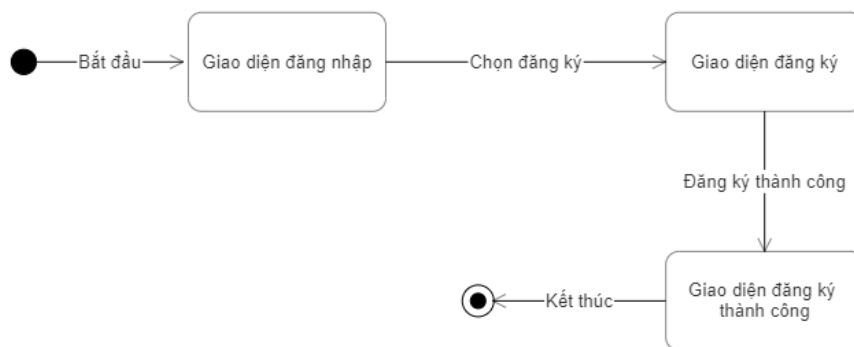
2.4.1 Đăng nhập/đăng ký

- Đăng nhập:



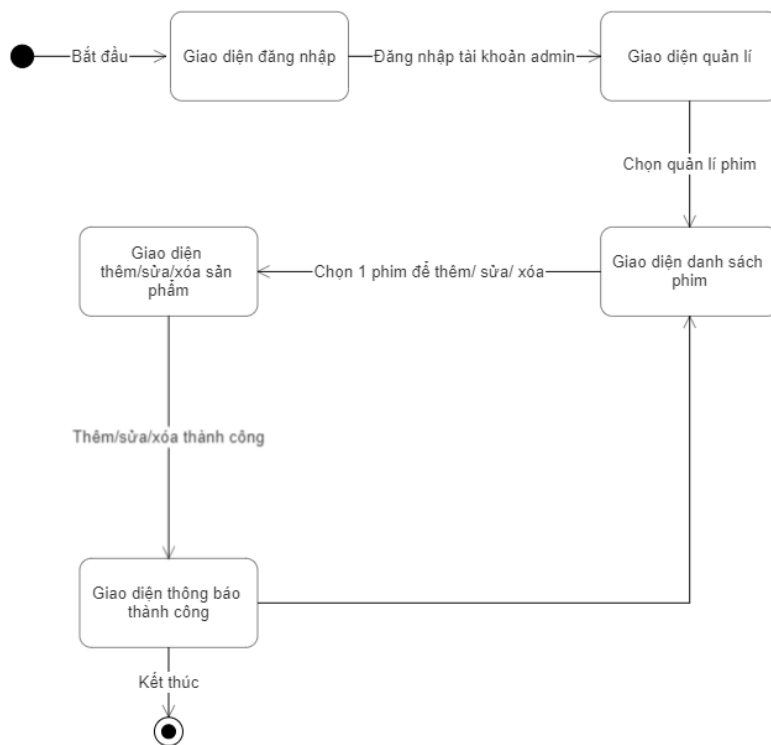
Hình 2.12 Biểu đồ trạng thái module đăng nhập

- Đăng ký:



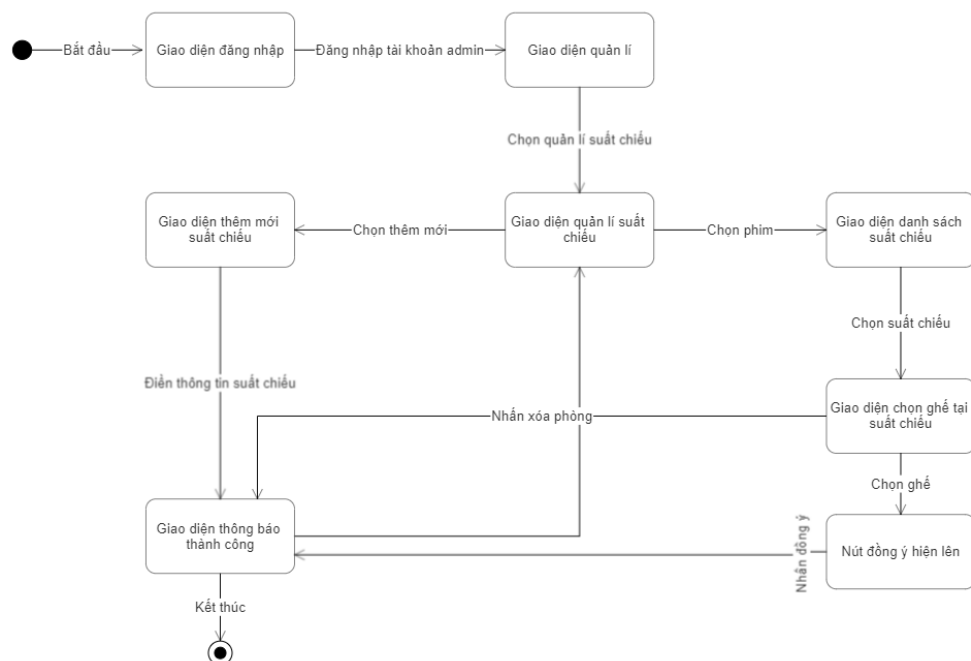
Hình 2.13 Biểu đồ trạng thái module đăng ký

2.4.2 Quản lý phim



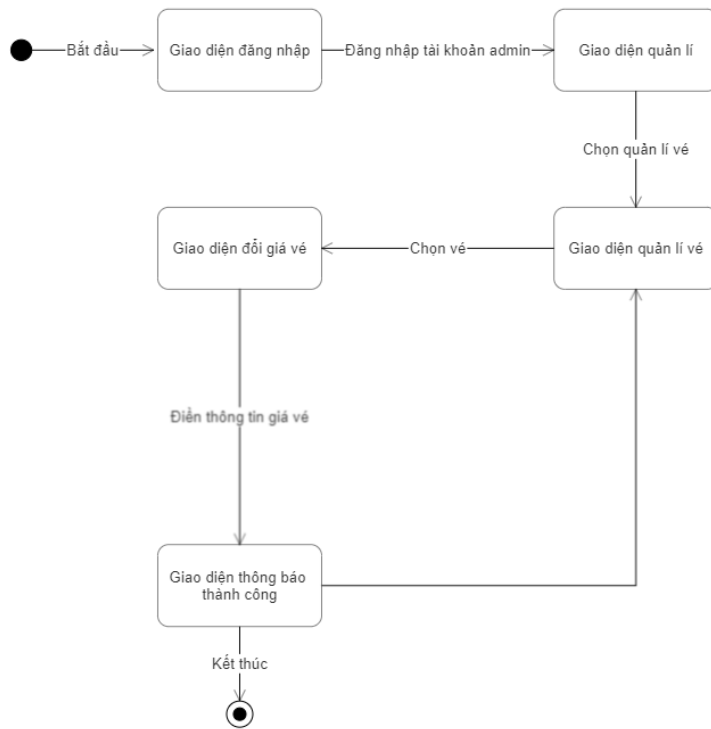
Hình 2.14 Biểu đồ trạng thái module quản lý phim

2.4.3 Quản lý suất chiếu



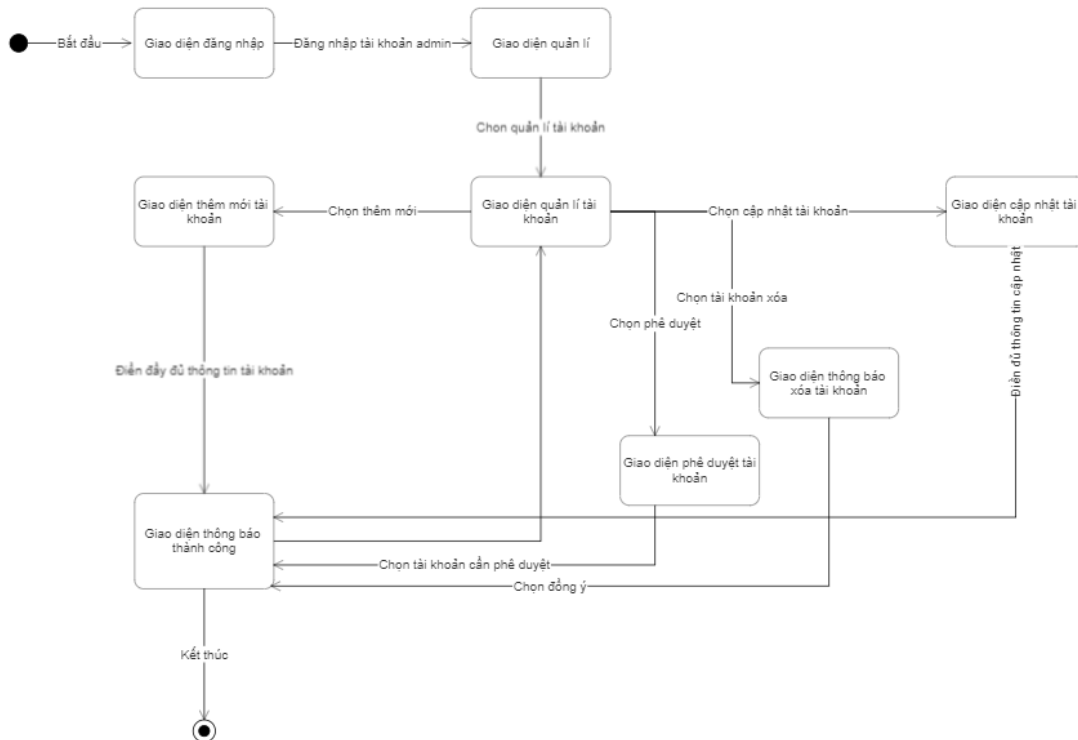
Hình 2.15 Biểu đồ trạng thái module quản lý suất chiếu

2.4.4 Quản lý vé



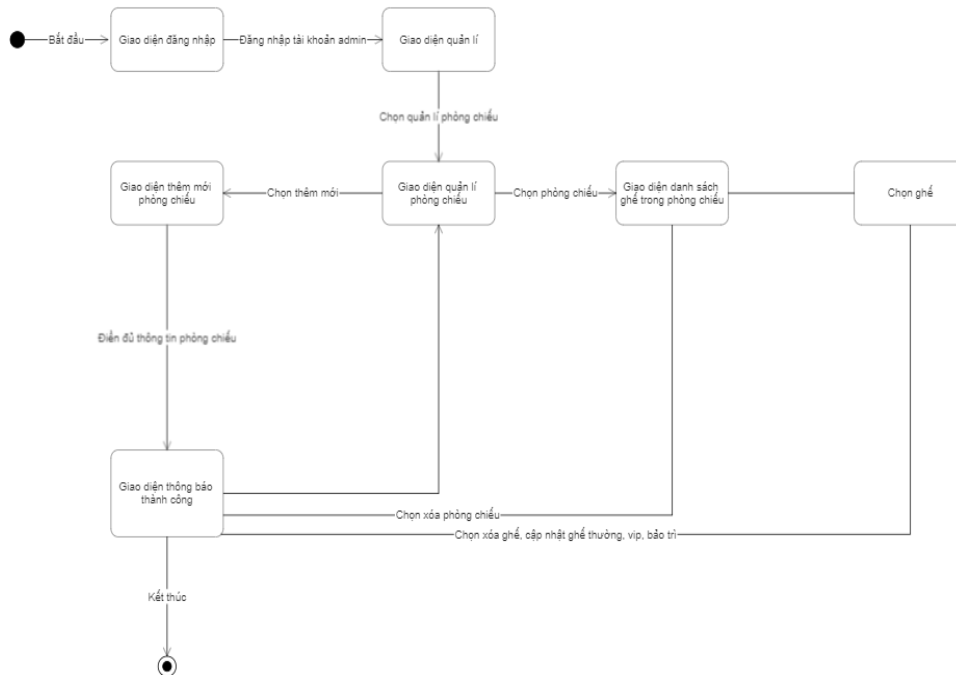
Hình 2.16 Biểu đồ trạng thái module quản lý vé

2.4.5 Quản lý tài khoản



Hình 2.17 Biểu đồ trạng thái module quản lý tài khoản

2.4.6 Quản lý phòng chiếu



Hình 2.18 Biểu đồ trạng thái module quản lý phòng chiếu

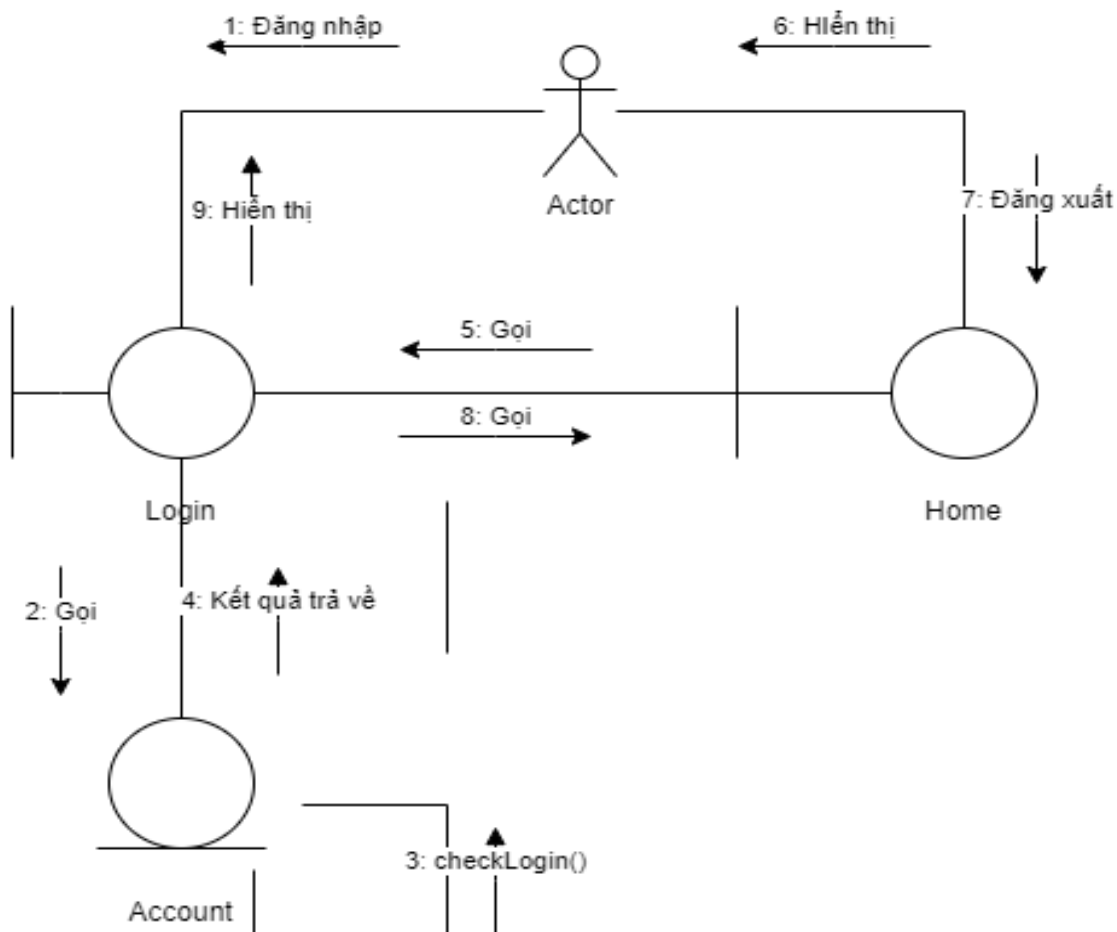
2.5 Biểu đồ giao tiếp

2.5.1 Đăng nhập/đăng ký

Đăng nhập/Đăng xuất:

Kịch bản V2:

1. Người dùng nhập thông tin đăng nhập và click Đăng nhập.
2. Lớp Login gọi lớp Account.
3. Lớp Account kiểm tra thông tin đăng nhập.
4. Lớp Account trả kết quả cho lớp Login
5. Lớp Login gọi lớp Home.
6. Lớp Home hiển thị cho người dùng.
7. Người dùng đăng xuất.
8. Lớp Home gọi lớp Login.
9. Lớp Login hiển thị cho người dùng.

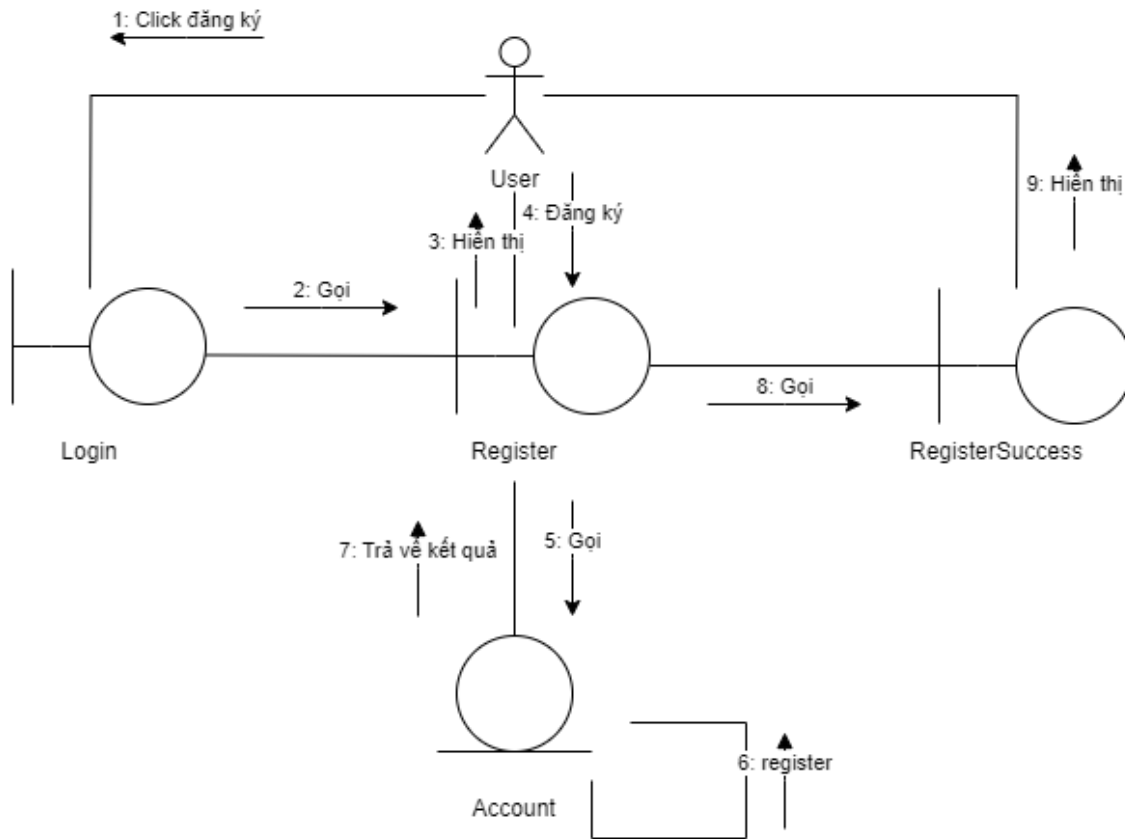


Hình 2.19 Biểu đồ giao tiếp đăng nhập

Đăng ký:

Kịch bản V2:

1. Người dùng click đăng ký từ trang đăng nhập.
2. Lớp Login gọi lớp Register.
3. Lớp Register hiển thị cho người dùng.
4. Người dùng nhập các thông tin và click đăng ký.
5. Lớp Register gọi lớp Account.
6. Lớp Account thực hiện đăng ký tài khoản.
7. Lớp Account trả kết quả cho lớp Register.
8. Lớp Register gọi lớp RegisterSuccess.
9. Lớp RegisterSuccess hiển thị cho người dùng.



Hình 2.20 Biểu đồ giao tiếp module đăng ký

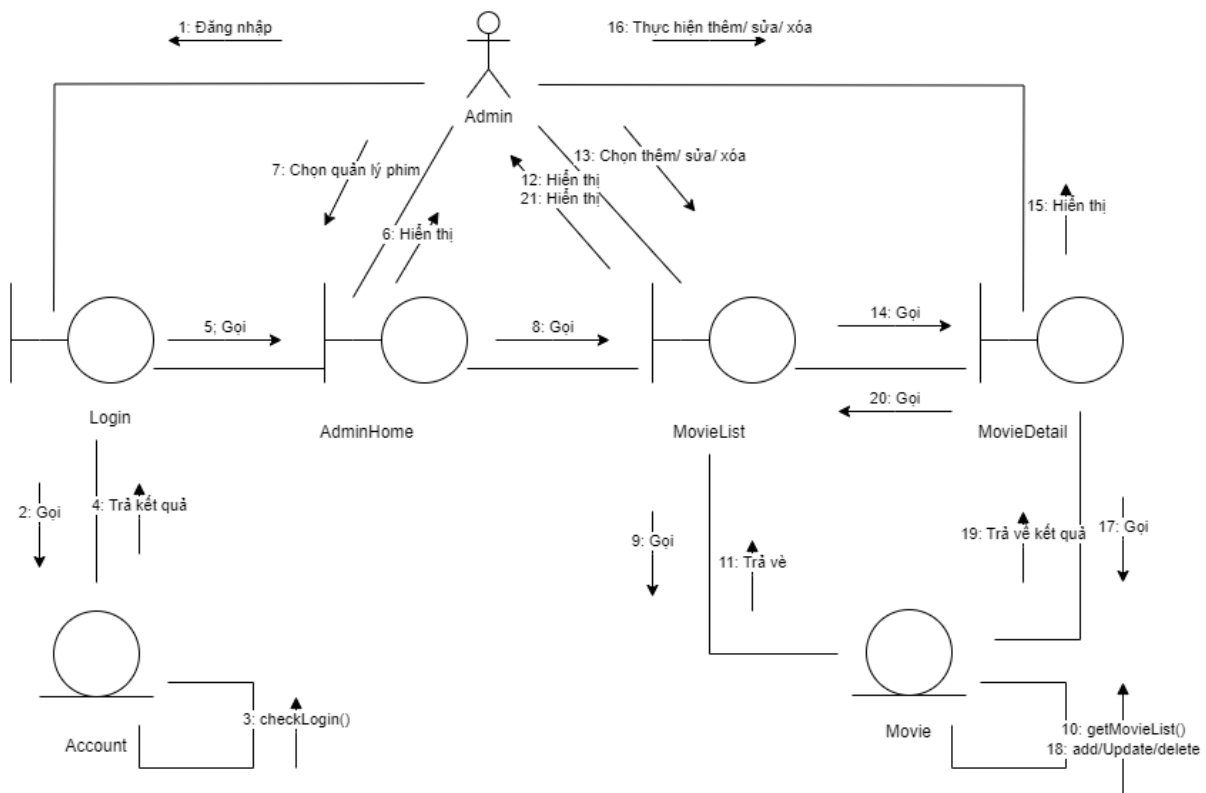
2.5.2 Quản lý phim

Kịch bản V2:

1. Admin nhập thông tin đăng nhập (tài khoản admin) và click Đăng nhập.
2. Lớp Login gọi lớp Account.
3. Lớp Account kiểm tra thông tin đăng nhập.
4. Lớp Account trả kết quả cho lớp Login
5. Lớp Login gọi lớp AdminHome.
6. Lớp AdminHome hiển thị cho admin.
7. Admin chọn Quản lý phim.
8. Lớp AdminHome gọi lớp MovieList.
9. Lớp MovieList gọi lớp Movie.
10. Lớp Movie gọi hàm getMovieList() lấy danh sách phim.
11. Lớp Movie trả kết quả cho lớp MovieList.
12. Lớp MovieList hiển thị cho admin.
13. Admin chọn thêm/sửa/xoá phim.
14. Lớp MovieList gọi lớp MovieDetail.
15. Lớp MovieDetail hiển thị cho admin.
16. Admin thực hiện thêm/sửa/xoá.
17. Lớp MovieDetail gọi lớp Movie.
18. Lớp Movie gọi hàm add/update/delete.
19. Lớp Movie trả kết quả cho lớp MovieDetail.

20. Lớp MovieDetail gọi lớp MovieList.

21. Lớp MovieList hiển thị cho admin.



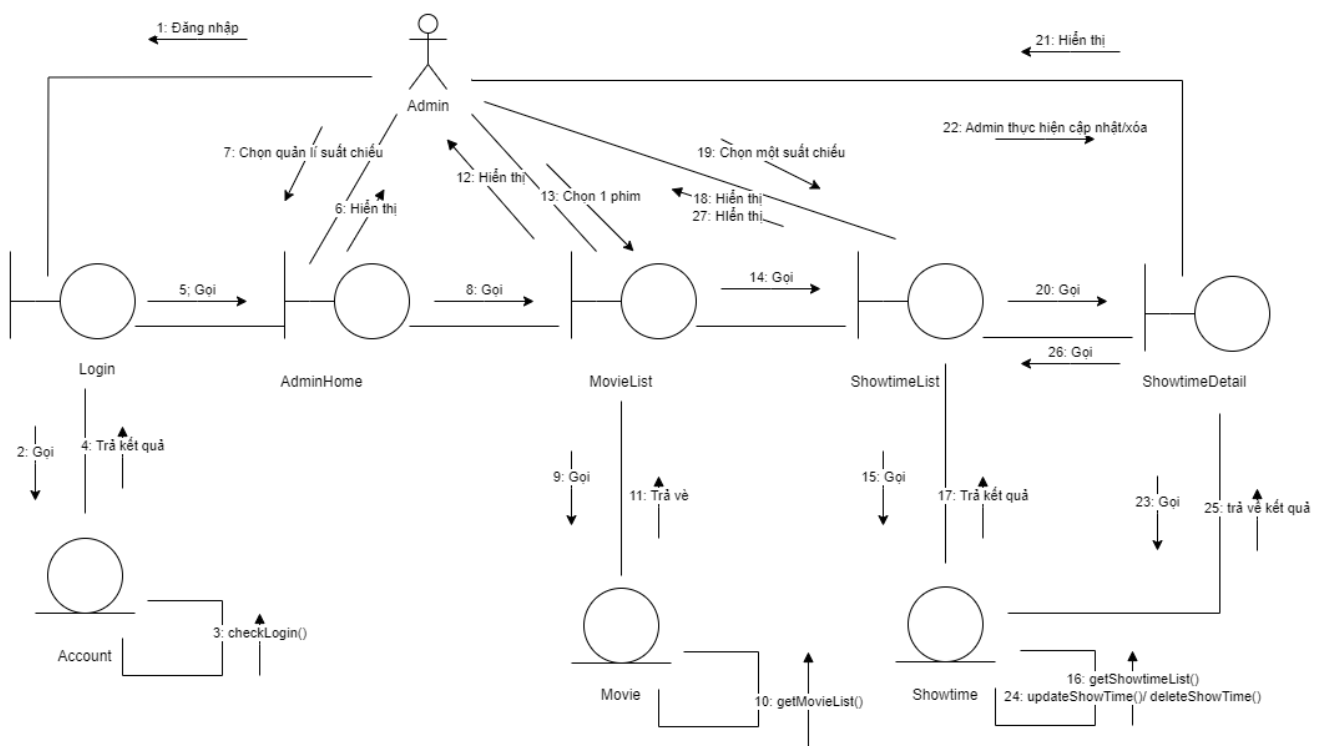
Hình 2.21 Biểu đồ giao tiếp module quản lý phim

2.5.3 Quản lý suất chiếu

Kịch bản V2:

1. Admin nhập thông tin đăng nhập (tài khoản admin) và click Đăng nhập.
2. Lớp Login gọi lớp Account.
3. Lớp Account kiểm tra thông tin đăng nhập.
4. Lớp Account trả kết quả cho lớp Login
5. Lớp Login gọi lớp AdminHome.
6. Lớp AdminHome hiển thị cho admin.
7. Admin chọn Quản lý suất chiếu.
8. Lớp AdminHome gọi lớp MovieList.
9. Lớp MovieList gọi lớp Movie.
10. Lớp Movie gọi hàm getMovieList() lấy danh sách phim.
11. Lớp Movie trả kết quả cho lớp MovieList.
12. Lớp MovieList hiển thị cho admin.
13. Admin chọn một Phim.
14. Lớp MovieList gọi lớp ShowtimeList.
15. Lớp ShowtimeList gọi lớp Showtime

16. Lớp Showtime gọi hàm getShowtimeList() lấy danh sách suất chiếu
17. Lớp Showtime trả ra kết quả cho lớp ShowtimeList
18. Lớp ShowtimeList hiển thị cho admin.
19. Admin chọn một suất chiếu
20. Lớp ShowtimeList gọi lớp ShowtimeDetail
21. Lớp ShowtimeDetail hiển thị cho admin
22. Admin thực hiện cập nhật/ xóa suất chiếu
23. Lớp ShowtimeDetail gọi lớp Showtime.
24. Lớp Showtime gọi hàm updateShowtime/ deleteShowtime.
25. Lớp Showtime trả kết quả cho lớp ShowtimeDetail.
26. Lớp ShowtimeDetail gọi lớp ShowtimeList.
27. Lớp ShowtimeList hiển thị cho admin.



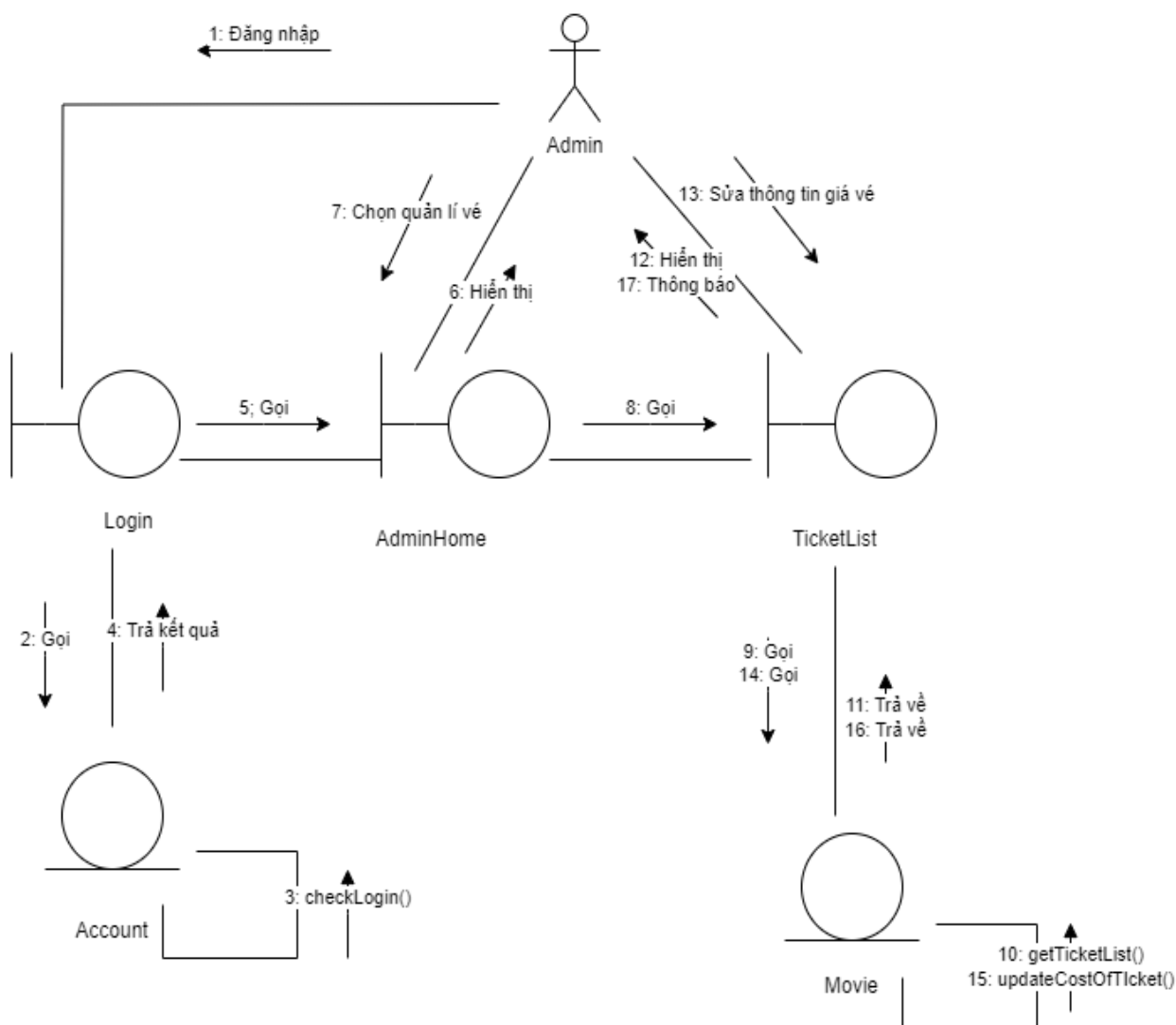
Hình 2.22 Biểu đồ giao tiếp module quản lý suất chiếu

2.5.4 Quản lý vé

Kịch bản V2:

1. Admin nhập thông tin đăng nhập (tài khoản admin) và click Đăng nhập.
2. Lớp Login gọi lớp Account.
3. Lớp Account kiểm tra thông tin đăng nhập.
4. Lớp Account trả kết quả cho lớp Login
5. Lớp Login gọi lớp AdminHome.
6. Lớp AdminHome hiển thị cho Admin.
7. Admin chọn Quản lý vé

8. Lớp AdminHome gọi lớp TicketList.
9. Lớp TicketList gọi lớp Ticket.
10. Lớp Ticket gọi hàm getTicketList() lấy thông tin vé.
11. Lớp Ticket trả kết quả cho lớp TicketList.
12. Lớp TicketList hiển thị cho người dùng.
13. Người dùng sửa thông tin giá vé và nhấn Lưu.
14. Lớp TicketList gọi lớp Ticket.
15. Lớp Ticket gọi hàm updateCostOfTicket() cập nhật thông tin.
16. Lớp Ticket trả kết quả cho lớp TicketList.
17. Lớp TicketList thông báo update thành công cho người dùng

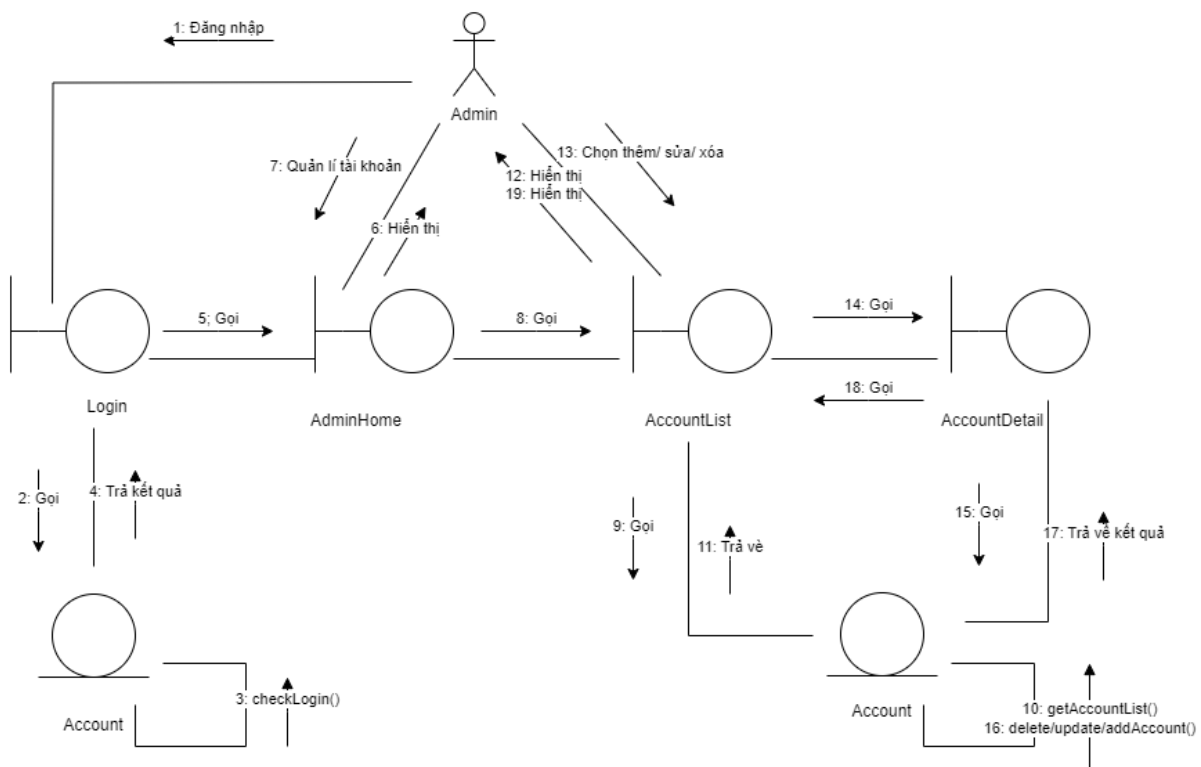


Hình 2.23 Biểu đồ giao tiếp module quản lý vé

2.5.5 Quản lý tài khoản

Kịch bản V2:

1. Admin nhập thông tin đăng nhập (tài khoản admin) và click Đăng nhập.
2. Lớp Login gọi lớp Account.
3. Lớp Account kiểm tra thông tin đăng nhập.
4. Lớp Account trả kết quả cho lớp Login
5. Lớp Login gọi lớp HomeAdmin.
6. Lớp HomeAdmin hiển thị cho admin.
7. Admin chọn Quản lý tài khoản.
8. Lớp HomeAdmin gọi lớp AccountList.
9. Lớp AccountList gọi lớp Account.
10. Lớp Account gọi hàm getAccountList() lấy tài khoản.
11. Lớp Account trả kết quả cho lớp AccountList.
12. Lớp AccountList hiển thị cho admin.
13. Admin chọn thêm/sửa/xóa tài khoản.
14. Lớp AccountList gọi lớp AccountDetail.
15. Lớp AccountDetail gọi lớp Account.
16. Lớp AccountDetail gọi hàm delete/update/addAccount() thực hiện thêm sửa xóa tài khoản
17. Lớp Account trả kết quả cho lớp AccountDetail.
18. Lớp AccountDetail gọi lớp AccountList.
19. Lớp AccountList thông báo thêm sửa xóa thành công cho admin.

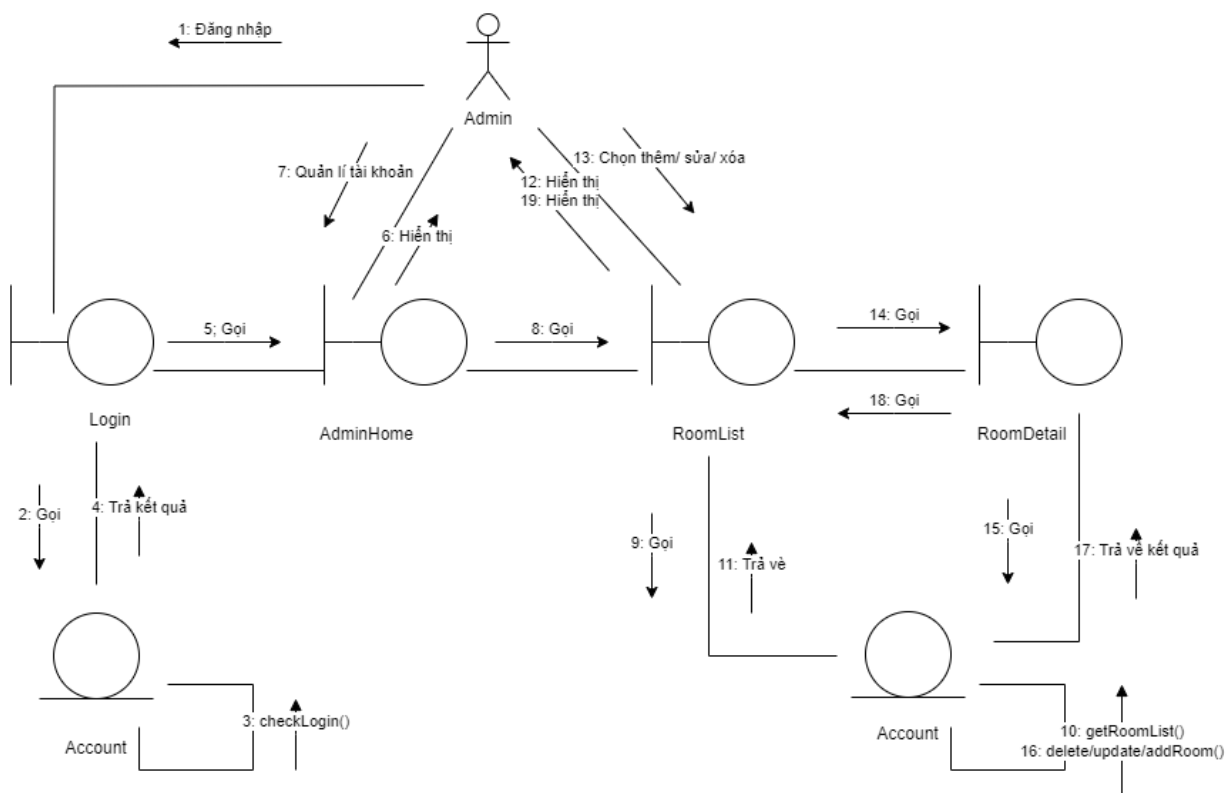


Hình 2.24 Biểu đồ giao tiếp module quản lý tài khoản

2.5.6 Quản lý phòng chiếu

Kịch bản V2:

1. Admin nhập thông tin đăng nhập (tài khoản admin) và click Đăng nhập.
2. Lớp Login gọi lớp Account.
3. Lớp Account kiểm tra thông tin đăng nhập.
4. Lớp Account trả kết quả cho lớp Login
5. Lớp Login gọi lớp HomeAdmin.
6. Lớp HomeAdmin hiển thị cho admin.
7. Admin chọn Quản lí tài khoản.
8. Lớp HomeAdmin gọi lớp RoomList.
9. Lớp RoomList gọi lớp Room.
10. Lớp Room gọi hàm getRoomList () lấy tài khoản.
11. Lớp Room trả kết quả cho lớp Room List.
12. Lớp RoomList hiển thị cho admin.
13. Admin chọn thêm/sửa/xóa tài khoản.
14. Lớp RoomList gọi lớp RoomDetail
15. Lớp RoomDetail gọi lớp Room
16. Lớp RoomDetail gọi hàm delete/update/addRoom() thực hiện thêm sửa xóa tài khoản
17. Lớp Room trả kết quả cho lớp RoomDetail.
18. Lớp RoomDetail gọi lớp RoomList.
19. Lớp RoomList thông báo thêm sửa xóa thành công cho admin

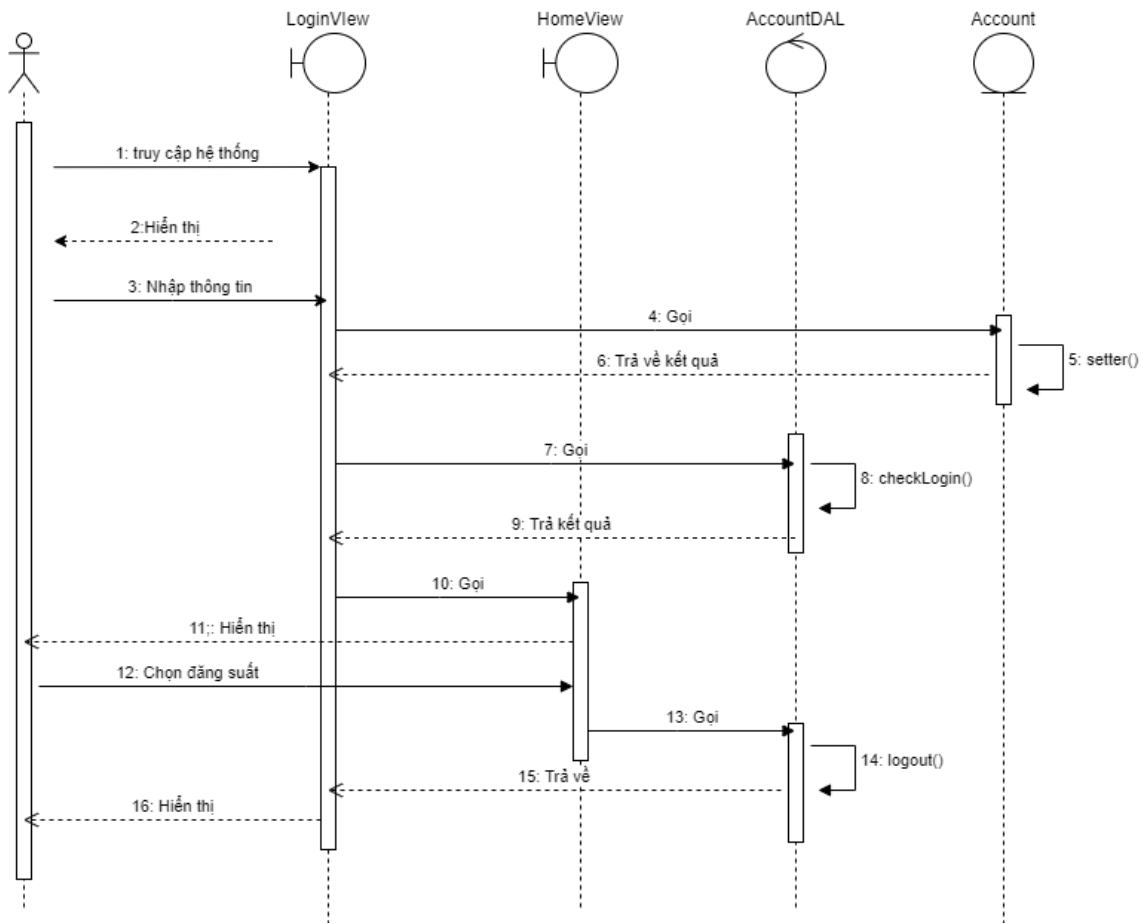


Hình 2.25 Biểu đồ giao tiếp module quản lý phòng chiếu

2.6 Biểu đồ tuần tự

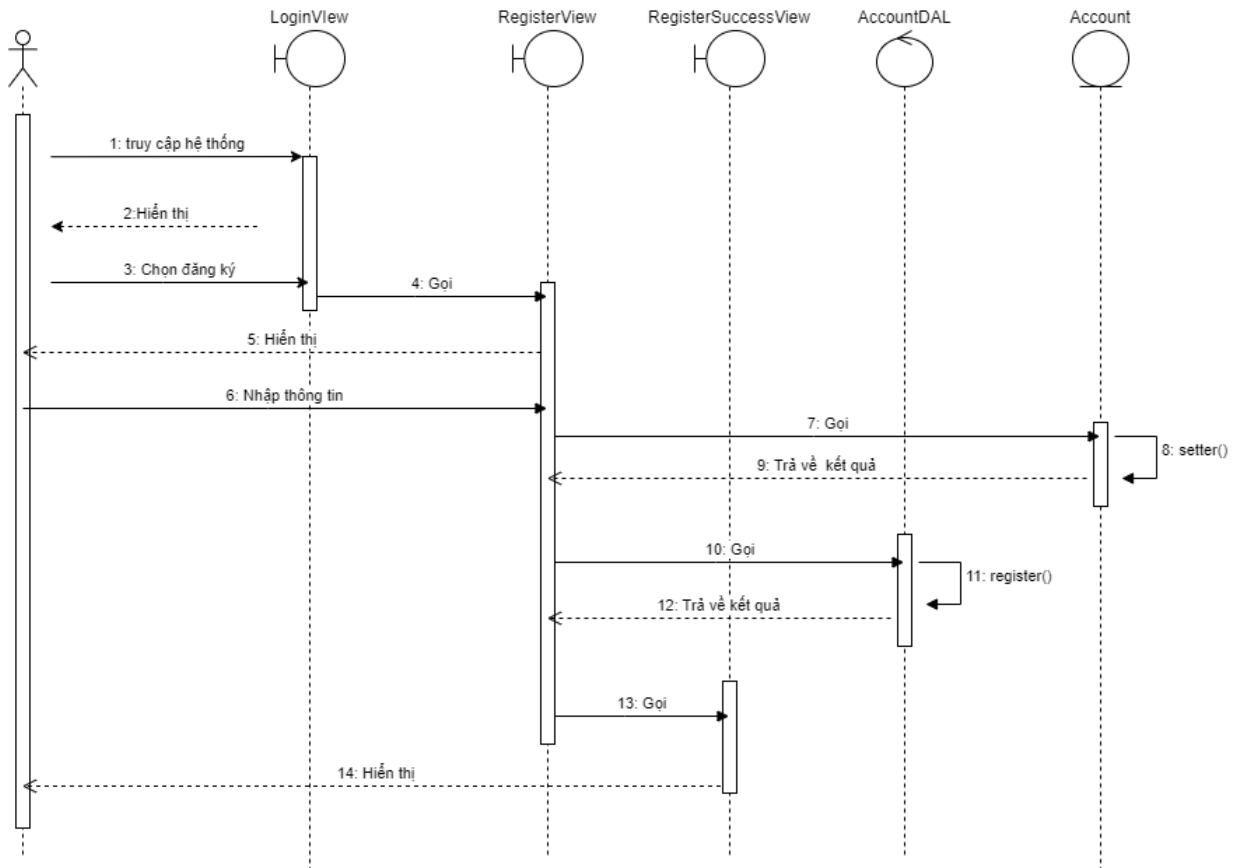
2.6.1 Đăng nhập/ đăng ký

Đăng nhập/ đăng ký:



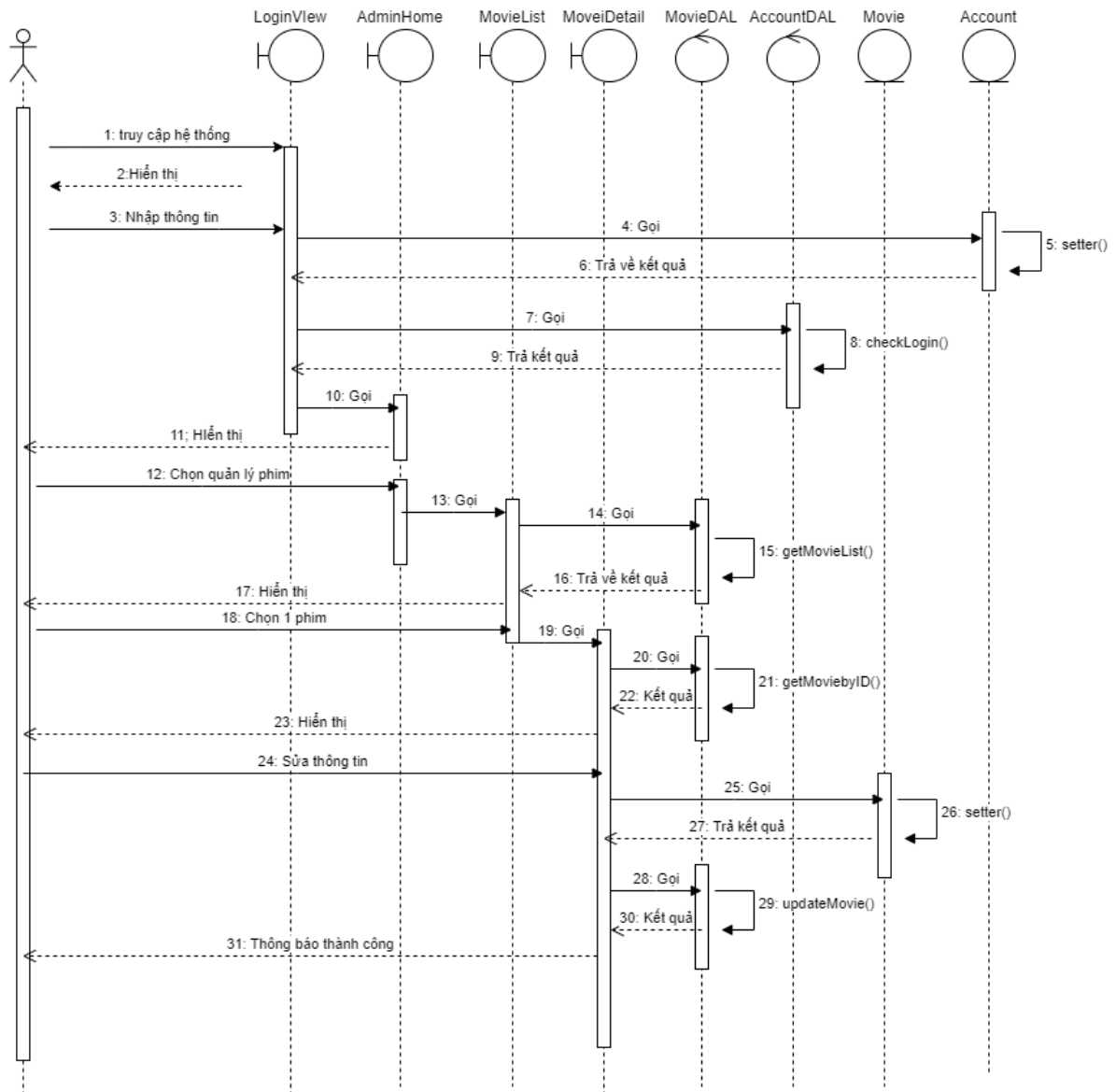
Hình 2.26 Biểu đồ tuần tự module đăng nhập

Đăng ký:



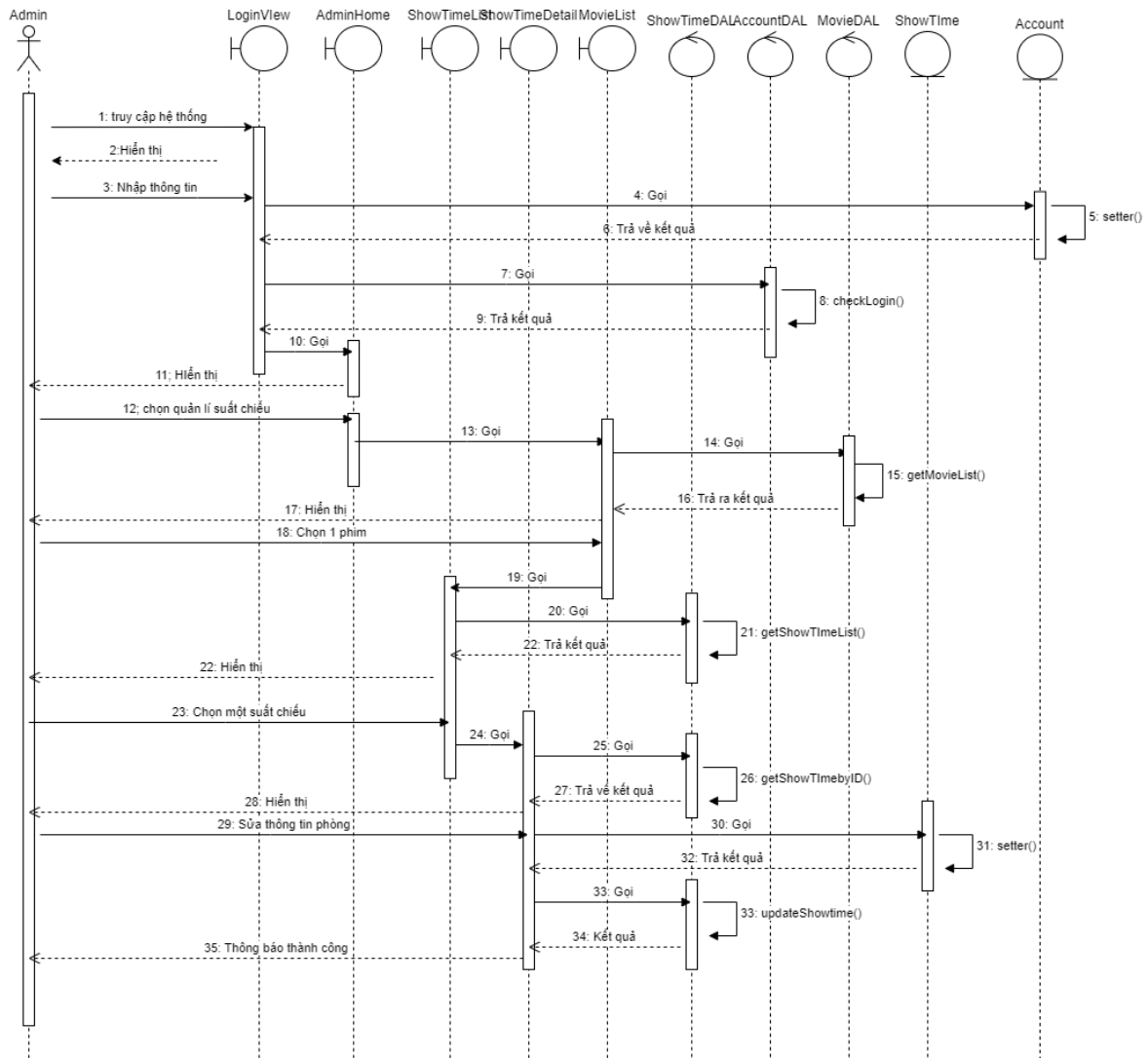
Hình 2.27 Biểu đồ tuần tự module đăng ký

2.6.2 Quản lý phim



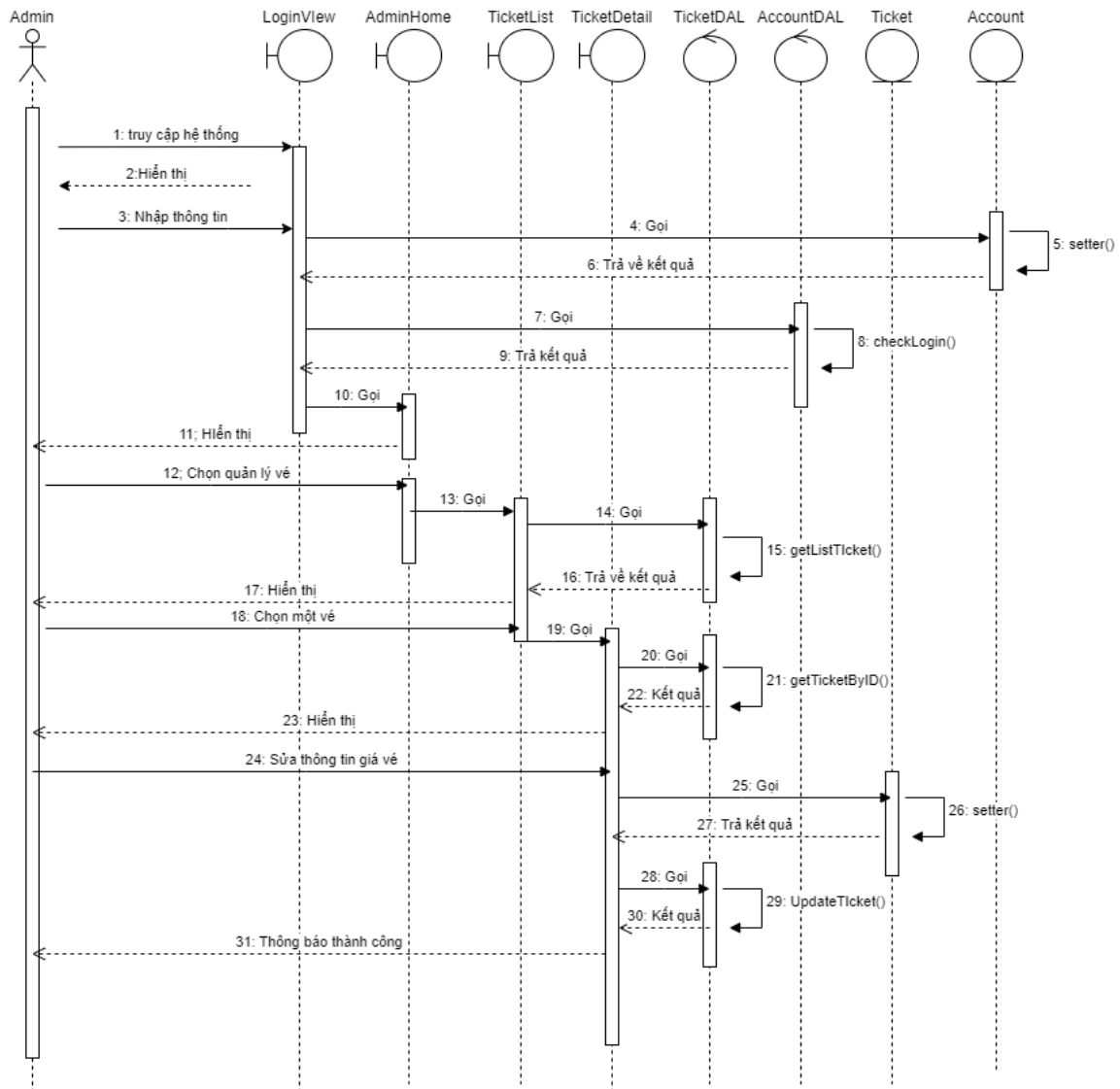
Hình 2.28 Biểu đồ tuần tự module quản lý phim

2.6.3 Quản lý suất chiếu



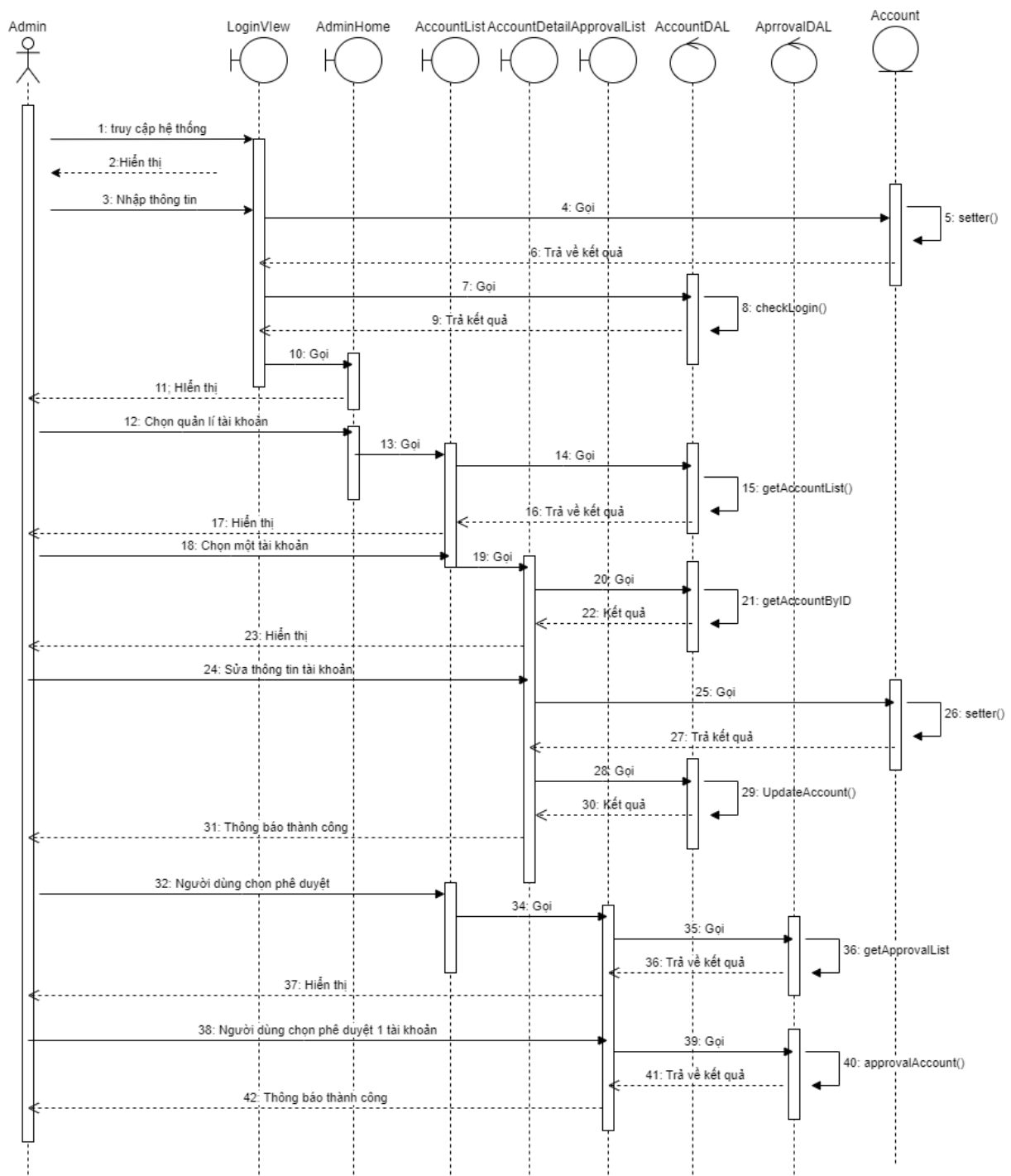
Hình 2.29 Biểu đồ tuần tự module quản lý suất chiếu

2.6.4 Quản lý vé



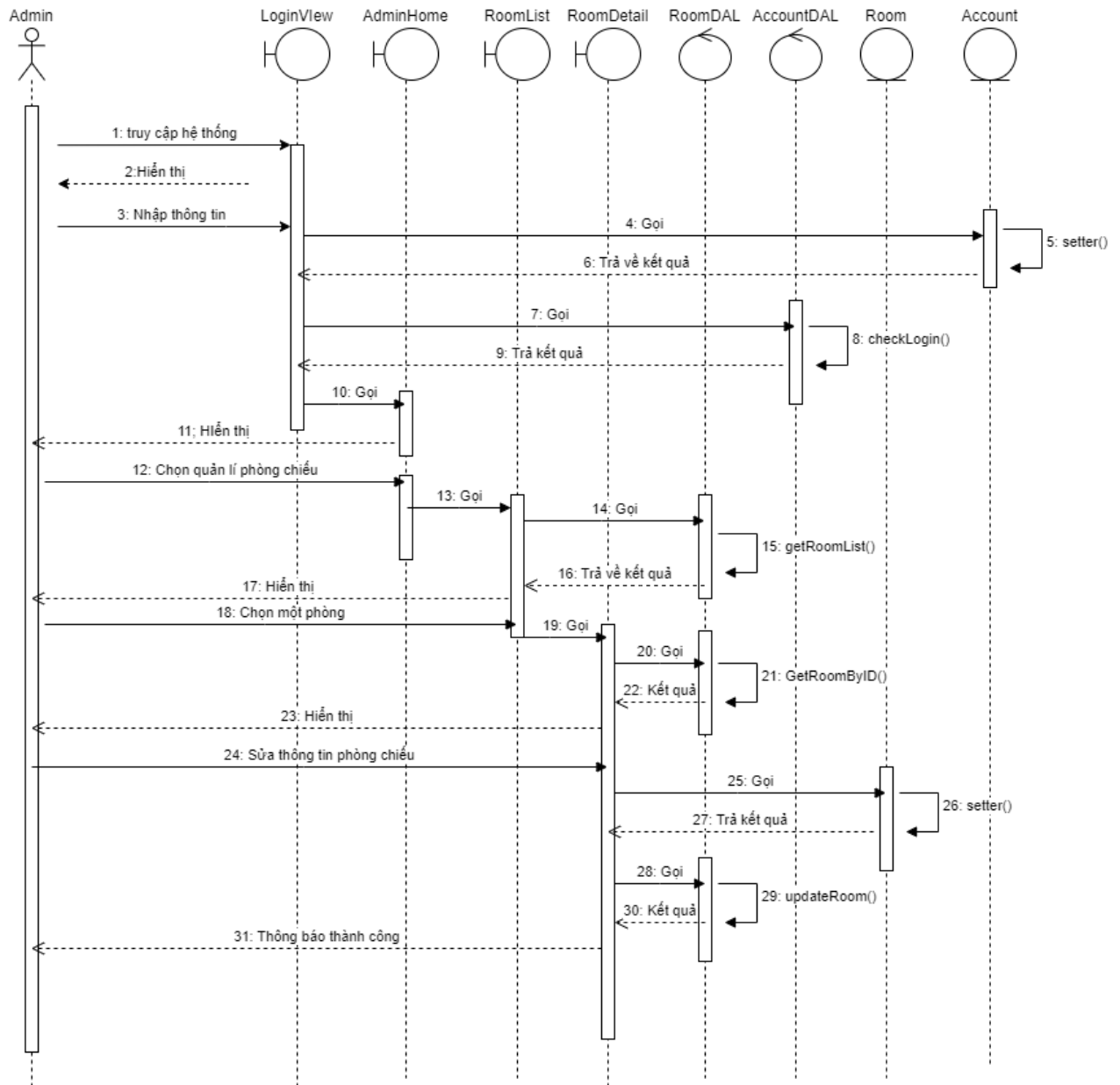
Hình 2.30 Biểu đồ tuần tự module quản lý vé

2.6.5 Quản lý tài khoản



Hình 2.31 Biểu đồ tuần tự module quản lý tài khoản

2.6.6 Quản lý phòng chiếu



Hình 2.32 Biểu đồ tuần tự module quản lý phòng chiếu

2.7 Biểu đồ lớp

2.7.1 Mô tả hệ thống

Hệ thống cho phép khách hàng có thể đặt vé trực tuyến. Cụ thể, khi truy cập vào website này, khách hàng có thể tạo tài khoản và đăng nhập, xem thông tin phim, tìm kiếm phim và suất chiếu theo nhu cầu, đặt vé nhanh chóng tiện lợi. Admin sử dụng hệ thống để quản lý phim, vé và các suất chiếu, tài khoản, phòng chiếu có trên hệ thống.

2.7.2 Trích các danh từ

- Các danh từ liên quan đến người: admin, khách hàng.
- Các danh từ liên quan đến vật: phim, vé, phòng chiếu.
- Các danh từ liên quan đến thông tin: tài khoản, suất chiếu, lịch sử.

2.7.3 Đánh giá và lựa chọn các danh từ làm lớp thực thể hoặc thuộc tính

- Tài khoản -> Lớp User: họ tên, email, mật khẩu, số điện thoại.
- Admin -> Lớp Admin, kế thừa lớp User.
- Khách hàng -> Lớp Client, kế thừa lớp User.
- Phim -> Lớp Movie: Mã phim, tên, đạo diễn, diễn viên, loại phim, dạng phim, ngày bắt đầu chiếu, ngày kết thúc, ngày xuất bản, mô tả, thời lượng.
- Loại phim -> Lớp MovieCategory: mã loại, tên loại.
- Dạng phim -> Lớp MovieType: mã dạng, tên dạng.
- Vé -> Lớp Ticket: mã vé, giá tiền, loại, mã phim.
- Phòng chiếu -> Lớp Room: Mã phòng, số ghế ngang, số ghế dọc.
- Suất chiếu -> Lớp ShowTime: Mã suất chiếu, mã phòng, mã khung giờ chiếu, số ghế dọc, số ghế ngang
- Khung giờ -> Lớp TemplateTime: Mã khung giờ chiếu, giờ chiếu
- Lịch sử -> Lớp History: Mã phim, mã vé, mã khung giờ chiếu, mã phòng, email, ngày tạo, số tiền thanh toán

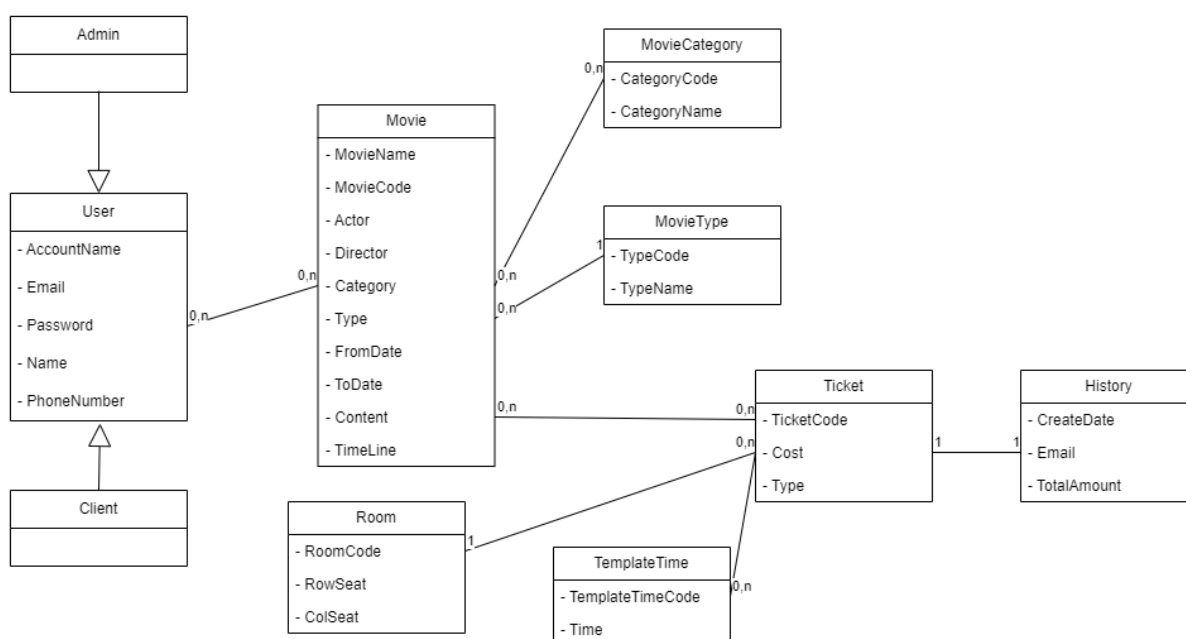
2.7.4 Xác định quan hệ số lượng giữa các thực thể

- Một "User" có thể là "Admin" hoặc "Client". Quan hệ này là một quan hệ một một (one-to-one). Mỗi người dùng chỉ có thể là admin hoặc khách hàng.
- Mỗi "Movie" có thể thuộc nhiều "MovieCategory" (loại phim) và mỗi "MovieCategory" có thể có nhiều "Movie". Quan hệ này là một quan hệ nhiều nhiều (many-to-many).
- Mỗi "Movie" có thể thuộc một "MovieType" (dạng phim) và mỗi "MovieType" có thể có nhiều "Movie". Quan hệ này là một quan hệ một nhiều (one-to-many).
- Mỗi "Movie" có thể có nhiều "Ticket" (vé) và mỗi "Ticket" chỉ thuộc về một "Movie". Quan hệ này là một quan hệ một nhiều (one-to-many).
- Mỗi "Room" (phòng chiếu) có thể có nhiều "ShowTime" (suất chiếu) và mỗi "ShowTime" chỉ thuộc về một "Room". Quan hệ này là một quan hệ một nhiều (one-to-many).

- Mỗi "ShowTime" có thể thuộc một "TemplateTime" (khung giờ) và mỗi "TemplateTime" có thể có nhiều "ShowTime". Quan hệ này là một quan hệ một nhiều (one-to-many).
- Mỗi "Ticket" (vé) thuộc về một "Movie" và mỗi "Movie" có thể có nhiều "Ticket". Quan hệ này là một quan hệ một nhiều (one-to-many).
- Mỗi "History" (lịch sử) ghi nhận thông tin về một "Movie", một "Ticket", một "TemplateTime", một "Room" và một "User". Mỗi "Movie", "Ticket", "TemplateTime", "Room" và "User" có thể có nhiều "History". Quan hệ này là một quan hệ một nhiều (one-to-many).

2.7.5 Vẽ biểu đồ lớp

Từ các bước phân tích, thu được biểu đồ lớp như hình bên dưới:

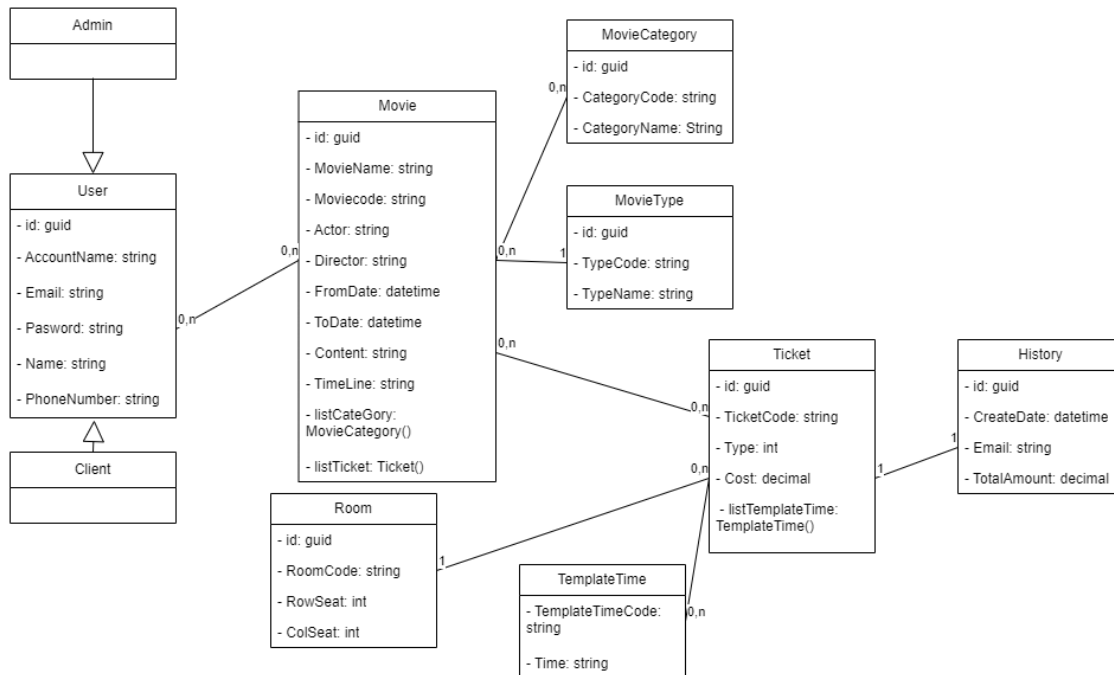


Hình 2.33 Biểu đồ lớp

2.8 Thiết kế biểu đồ lớp thực thể

Từ biểu đồ lớp đã phân tích ở trên, thiết kế biểu đồ lớp thực thể theo các bước:

- Thêm id cho các thực thể không kế thừa thực thể khác.
- Gán kiểu dữ liệu cho các thuộc tính.



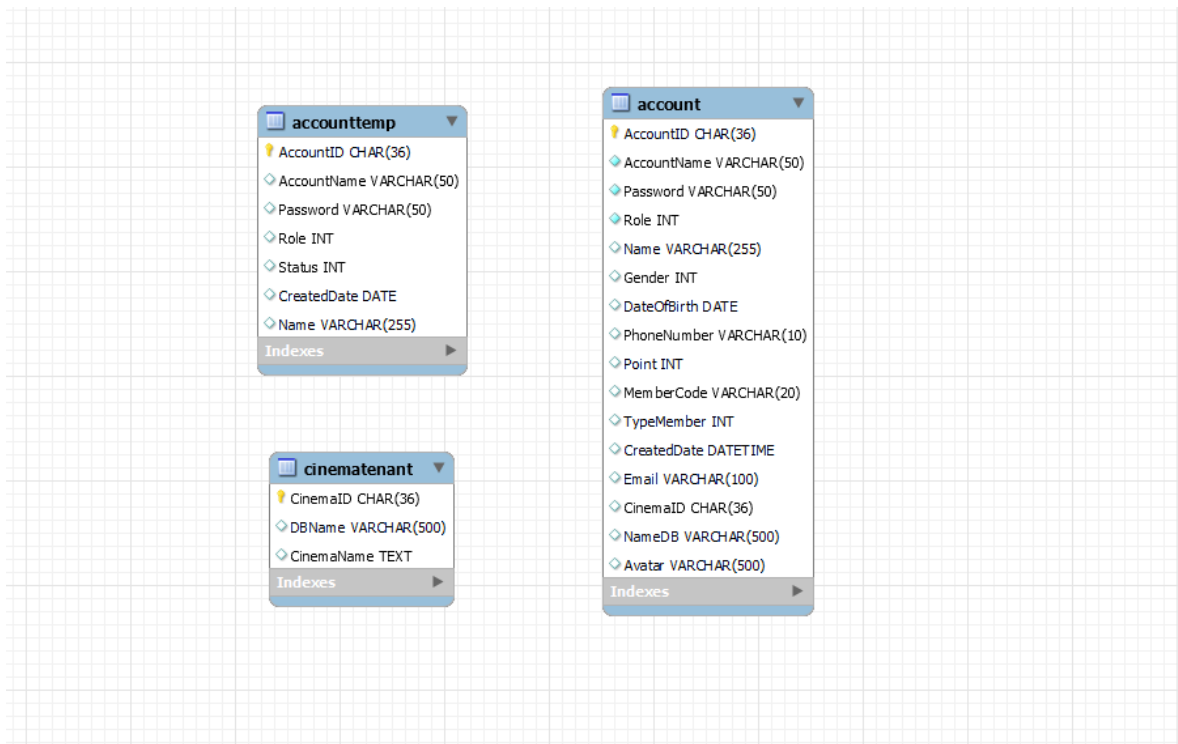
Hình 2.34 Biểu đồ thực thể

2.9 Thiết kế CSDL

- CSDL gồm nhiều CSDL: Một lưu danh sách tài khoản, còn lại là các CSDL lưu thông tin của hệ thống tương ứng với từng rạp phim.
- Trên CSDL lưu tài khoản bao gồm các bảng: account, accounttemp, cinematenant
- Trên CSDL lưu thông tin hệ thống bao gồm các bảng: categorymovie, dboptions, history, listcinema, movie, moviecategorydetail, room, roomcinema, seat, statusroomcinema, templateticketmovie, templatetimemovieon, ticket, typemovie.
- Với mỗi lớp, các thuộc tính có kiểu dữ liệu không phải đối tượng sẽ là các thuộc tính của bảng.
- Chuyển quan hệ số lượng giữa các thực thể thành quan hệ số lượng giữa các bảng:
 - movie và typemovie: 1-n.
 - movie và moviecategorydetail: 1-n.
 - categorymovie và moviecategorydetail: 1-n.
 - movie và roomcinema: 1-n.
 - roomcinema và ticket: 1-n.
 - templatetimemovieon và roomcinema: 1-n.
 - listcinema và room: 1-n.
 - room và seat: 1-n.
 - seat và ticket: 1-n.
- Thêm khoá ngoại:
 - Bảng movie có khoá ngoại TypeID tham chiếu đến khoá chính TypeID của bảng typemovie.
 - Bảng roomcinema có khoá ngoại MovieID tham chiếu đến khoá chính MovieID của bảng movie.

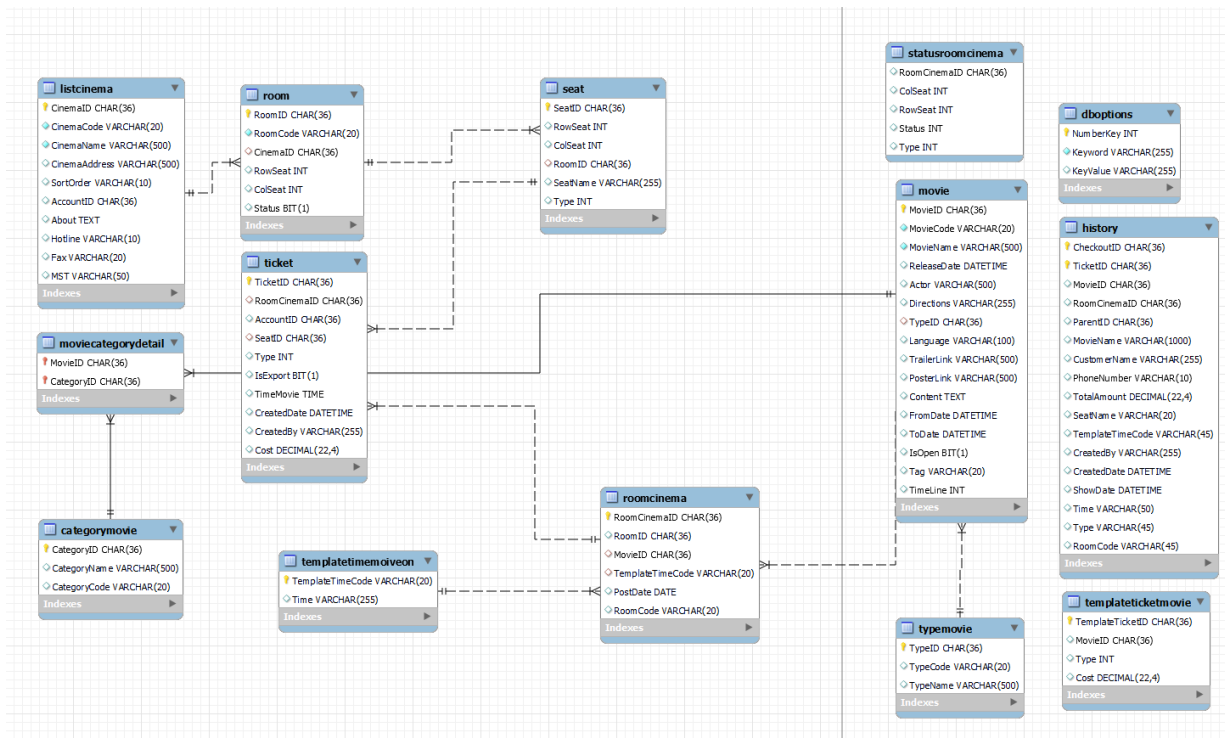
- Bảng roomcinema có khoá ngoại TemplateTimeCode tham chiếu đến khoá chính TemplateTimeCode của bảng templatetimemovieon.
 - Bảng ticket có khoá ngoại SeatID tham chiếu đến khoá chính SeatID của bảng seat.
 - Bảng ticket có khoá ngoại RoomCinemaID tham chiếu đến khoá chính id của bảng roomcinema.
 - Bảng moviecategorydetail có khoá ngoại MovieID tham chiếu đến khoá chính id của bảng movie.
 - Bảng moviecategorydetail có khoá ngoại CategoryID tham chiếu đến khoá chính CategoryID của bảng categorymovie.
 - Bảng room có khoá ngoại CinemaID tham chiếu đến khoá chính id của bảng listcinema.
 - Bảng seat có khoá ngoại RoomID tham chiếu đến khoá chính id của bảng room.
- Kết quả thu được biểu đồ thực thể liên kết mô tả CSDL của hệ thống như sau:

- CSDL Tài khoản



Hình 2.35 Biểu đồ RM CSDL Tài khoản

- CSDL Dữ liệu



Hình 2.36 Biểu đồ RM CSDL Dữ liệu

- Diễn giải các bảng
 - CSDL Tài khoản

Bảng 3.1 Bảng account

Tên trường	Kiểu dữ liệu	Ràng buộc	Comment
AccountID	char(36)	Primary	ID khóa chính
AccountName	varchar(50)	Not null	Tên tài khoản
Password	varchar(50)	Not null	Mật khẩu
Role	int	Not null	Quyền
Name	varchar(255)	Not null	Tên
Gender	int	Null	Giới tính
DateOfBirth	date	Null	Ngày sinh
PhoneNumber	varchar(10)	Not null	Điện thoại
Point	int	Null	Điểm
MemberCode	varchar(20)	Null	Mã thành viên
TypeMember	int	Null	Loại thành viên
CreatedDate	datetime	Null	Ngày tạo
Email	varchar(100)	Not null	Email
CinemaID	char(36)	Null	ID rạp phim
NameDB	varchar(500)	Null	Tên CSDL
Avatar	varchar(500)	Null	Avatar link

Bảng 3.2 Bảng accounttemp

Tên trường	Kiểu dữ liệu	Ràng buộc	Comment
AccountID	char(36) PK	Not null	ID khóa chính
AccountName	varchar(50)	Not null	Tên tài khoản
Password	varchar(50)	Not null	Mật khẩu
Role	int	Not null	Quyền
Status	int	Not null	Tên
CreatedDate	date	Null	Ngày tạo
Name	varchar(255)	Null	Ngày sinh

Bảng 3.3 Bảng cinematenant

Tên trường	Kiểu dữ liệu	Ràng buộc	Comment
CinemaID	char(36) PK	Not null	ID khóa chính
DBName	varchar(500)	Not null	Tên CSDL
CinemaName	text	Not null	Tên Rạp

- CSDL Rạp phim

Bảng 3.4 Bảng categorymovie

Tên trường	Kiểu dữ liệu	Ràng buộc	Comment
CategoryID	char(36) PK	Not null	ID khóa chính
CategoryName	varchar(500)	Not null	Tên loại
CategoryCode	varchar(20)	Not null	Mã Loại

Bảng 3.5 Bảng dboptions

Tên trường	Kiểu dữ liệu	Ràng buộc	Comment
NumberKey	int AI PK	Not null	ID khóa chính
Keyword	varchar(255)	Not null	Từ khóa
KeyValue	varchar(255)	Not null	Giá trị

Bảng 3.6 Bảng history

Tên trường	Kiểu dữ liệu	Ràng buộc	Comment
CheckoutID	char(36) PK	Not null	ID khóa chính
TicketID	char(36) PK	Not null	ID khóa chính
MovieID	char(36)	Not null	ID phim
RoomCinemaID	char(36)	Not null	ID phòng chiếu
ParentID	char(36)	Null	ID khóa chính cha
MovieName	varchar(1000)	Null	Tên phim
CustomerName	varchar(255)	Null	Tên khách hàng
PhoneNumber	varchar(10)	Not null	Điện thoại
TotalAmount	decimal(22,4)	Null	Tổng tiền
SeatName	varchar(20)	Null	Ghế
TemplateTimeCode	varchar(45)	Null	Khung giờ
CreatedBy	varchar(255)	Null	Người tạo
CreatedDate	datetime	Not null	Ngày tạo
ShowDate	datetime	Null	Ngày xem
Time	varchar(50)	Null	Thời lượng
Type	varchar(45)	Null	Loại vé
RoomCode	varchar(45)	Null	Mã phòng

Bảng 3.7 Bảng listcinema

Tên trường	Kiểu dữ liệu	Ràng buộc	Comment
CinemaID	char(36) PK	Not null	ID khóa chính
CinemaCode	varchar(20)	Not null	Mã rạp
CinemaName	varchar(500)	Not null	Tên rạp
CinemaAddress	varchar(500)	null	Địa chỉ
SortOrder	varchar(10)	Null	Thứ tự
AccountID	char(36)	Null	ID tài khoản
About	text	Null	Mô tả
Hotline	varchar(10)	null	Hotline
Fax	varchar(20)	Null	Fax
MST	varchar(50)	Null	Mã số thuế

Bảng 3.8 Bảng movie

Tên trường	Kiểu dữ liệu	Ràng buộc	Comment
MovieID	char(36) PK	Not null	ID khóa chính
MovieCode	varchar(20)	Not null	Mã phim
MovieName	varchar(500)	Not null	Tên phim
ReleaseDate	datetime	null	Ngày công chiếu
Actor	varchar(500)	Null	Diễn viên
Directions	varchar(255)	Null	Đạo diễn
TypeID	char(36)	Null	ID dạng phim
Language	varchar(100)	null	Ngôn ngữ
TrailerLink	varchar(500)	Null	Link trailer
PosterLink	varchar(500)	Null	Link poster
Content	text	Null	Nội dung
FromDate	datetime	Null	Từ ngày
ToDate	datetime	Null	Đến ngày
IsOpen	bit(1)	Null	Đang chiếu
Tag	varchar(20)	Null	Tag
TimeLine	int	Null	Thời lượng

Bảng 3.9 Bảng categorymoviedetail

Tên trường	Kiểu dữ liệu	Ràng buộc	Comment
MovieID	char(36) PK	Not null	ID khóa chính ID phim
CategoryID	char(36) PK	Not null	ID khóa chính ID loại

Bảng 3.10 Bảng room

Tên trường	Kiểu dữ liệu	Ràng buộc	Comment
RoomID	char(36) PK	Not null	ID khóa chính
RoomCode	varchar(20)	Not null	Mã phòng
CinemaID	char(36)	null	ID rạp
RowSeat	int	null	Số hàng
ColSeat	int	Null	Số cột
Status	bit(1)	Null	Tình trạng

Bảng 3.11 Bảng roomcinema

Tên trường	Kiểu dữ liệu	Ràng buộc	Comment
RoomCinemaID	char(36) PK	Not null	ID khóa chính
RoomID	char(36)	Not null	ID phòng
MovieID	char(36)	null	ID phim
TemplateTimeCode	varchar(20)	null	Mã khung giờ
PostDate	date	Null	Ngày chiếu
RoomCode	varchar(20)	Null	Mã phòng

Bảng 3.12 Bảng seat

Tên trường	Kiểu dữ liệu	Ràng buộc	Comment
SeatID	char(36) PK	Not null	ID khóa chính
RowSeat	int	Not null	Hàng
ColSeat	int	Not null	Cột
RoomID	char(36)	not null	ID phòng chiếu
SeatName	varchar(255)	Null	Tên ghế
Type	int	Null	Loại ghế

Bảng 3.13 Bảng statusroomcinema

Tên trường	Kiểu dữ liệu	Ràng buộc	Comment
RoomCinemaID	char(36)	Not null	ID phòng chiếu
ColSeat	int	Not null	Hàng
RowSeat	int	Not null	Cột
Status	int	not null	Tình trạng
Type	int	Null	Loại ghế

Bảng 3.14 Bảng templateticketmovie

Tên trường	Kiểu dữ liệu	Ràng buộc	Comment
TemplateTicketID	char(36) PK	Not null	ID khóa chính
MovieID	char(36)	Not null	ID phim
Type	int	Not null	Loại vé
Cost	decimal(22,4)	not null	Giá tiền
Type	int	Null	Loại ghế

Bảng 3.15 Bảng templatetimemovieon

Tên trường	Kiểu dữ liệu	Ràng buộc	Comment
TemplateTimeCode	varchar(20) PK	Not null	Mã khóa chính
Time	varchar(255)	Not null	Giờ

Bảng 3.16 Bảng ticket

Tên trường	Kiểu dữ liệu	Ràng buộc	Comment
TicketID	char(36) PK	Not null	ID khóa chính
RoomCinemaID	char(36)	Not null	ID phòng chiếu
AccountID	char(36)	Not null	ID tài khoản
SeatID	char(36)	not null	ID ghế
Type	int	Null	Loại ghế
IsExport	bit(1)	Null	Đã xuất vé
TimeMovie	time	Null	Giờ chiếu
CreatedDate	datetime	null	Tạo ngày
CreatedBy	varchar(255)	Null	Người tạo
Cost	decimal(22,4)	Null	Giá tiền

Bảng 3.17 Bảng typemovie

Tên trường	Kiểu dữ liệu	Ràng buộc	Comment
TypeID	char(36) PK	Not null	ID khóa chính
TypeCode	varchar(20)	Not null	Mã dạng
TypeName	varchar(500)	null	Tên dạng phim

2.10 Kết luận chương 2

Chương 2 đã trình bày phân phân tích và thiết kế hệ thống, nhờ đó các thiết kế về chức năng và CSDL của website được làm rõ hơn. Trên cơ sở các phân tích thiết kế ở chương 2, chương 3 sẽ trình bày về phân cài đặt và kết quả triển khai hệ thống.

CHƯƠNG 3: CÀI ĐẶT HỆ THỐNG

3.1 Yêu cầu hệ thống

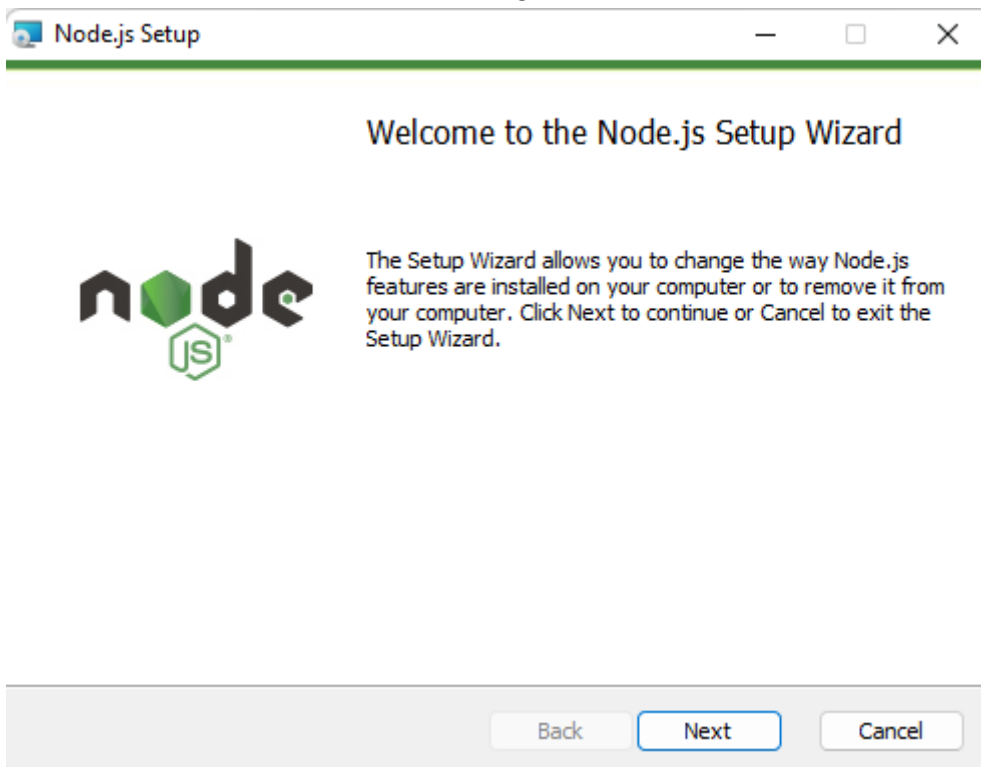
- Hệ điều hành: Windows, Linux, macOS.
- Cài đặt NodeJS.
- Cài đặt VueJS.
- Cài đặt MySQL.

3.2 Các công cụ lập trình, thư viện hỗ trợ

- Công cụ lập trình: Visual Studio Code, Visual Studio 2022, Vue devtool.
- Công cụ thiết kế: Visual Paradigm.
- Thư viện: Vite, Vuex, Bootstrap.
- Các ngôn ngữ và công cụ khác:
 - SCSS: tiền xử lý CSS.
 - Ngôn ngữ TypeScript.
 - I18n: Thay đổi ngôn ngữ
 - Firebase Storage: Lưu ảnh

3.3 Cài đặt

- Cài đặt NodeJS
 - Bước 1: Tải bộ cài NodeJS phiên bản LTS
 - Bước 2: Sau khi tải thành công, tiến hành cài đặt NodeJS



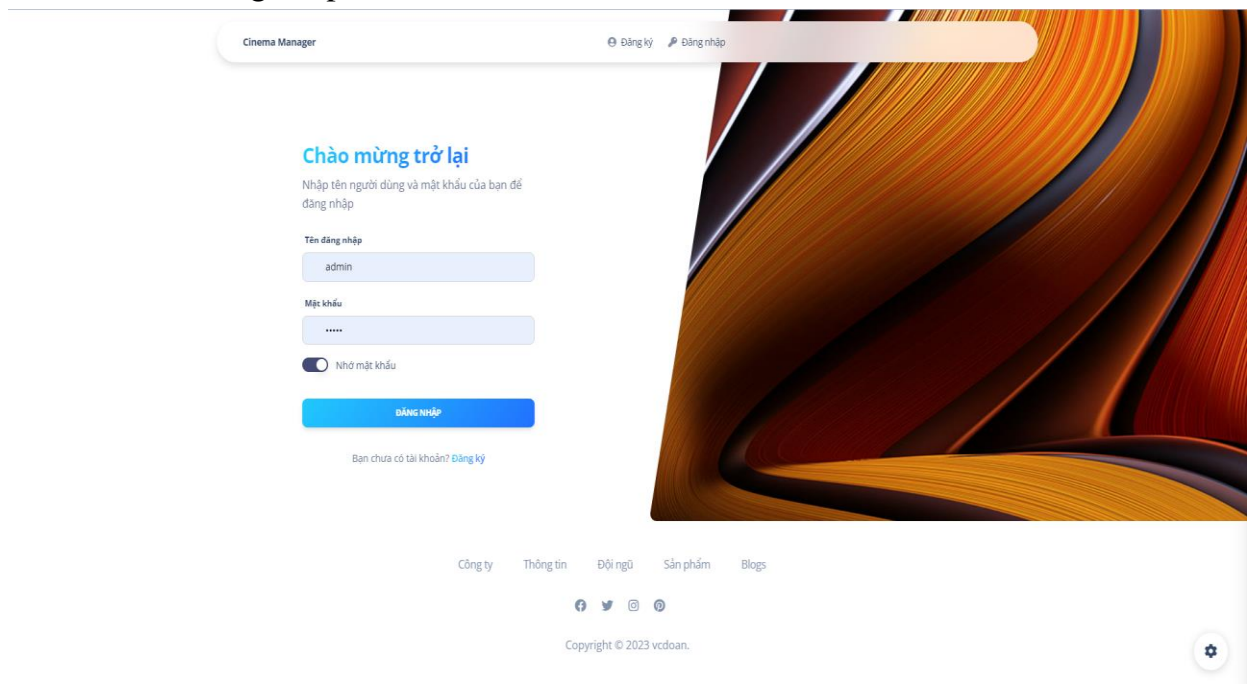
Hình 3.1 Hình giao diện cài đặt NodeJS

- Cài đặt VueJS:

- Tạo một thư mục mới chứa project.
- Chạy các lệnh sau ở terminal để cài đặt Vue CLI và khởi tạo project:
 - `npm install --global @vue/cli`
 - `vue init webpack my-project`
 - `cd my-project`
 - `npm run dev`

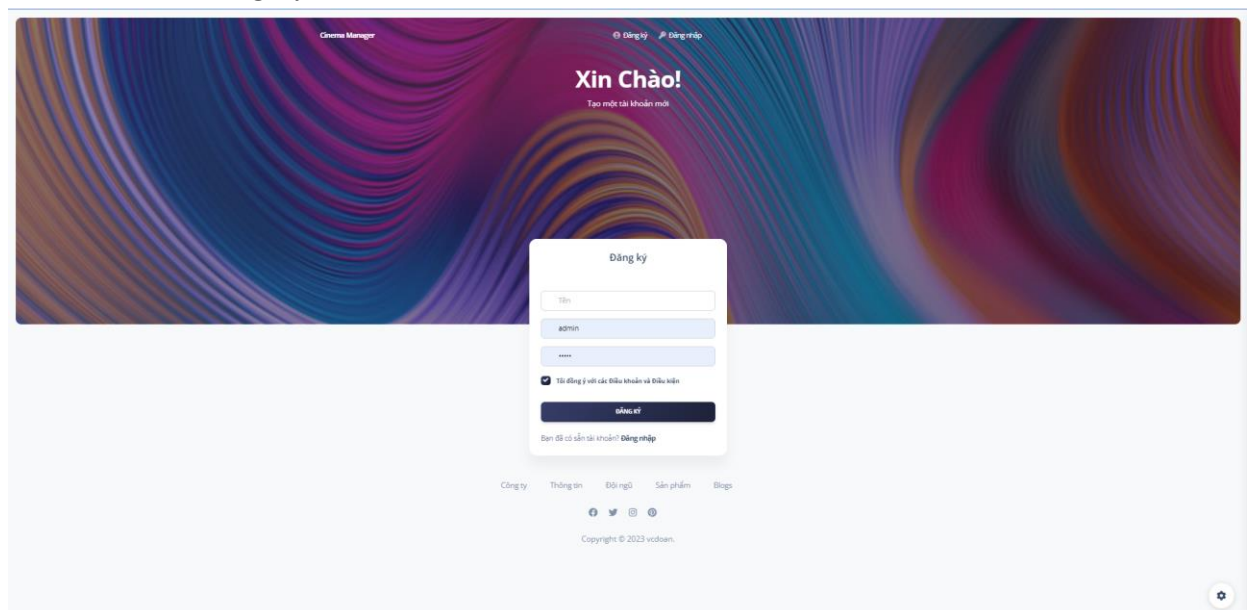
3.4 Giao diện hệ thống

- Giao diện đăng nhập:



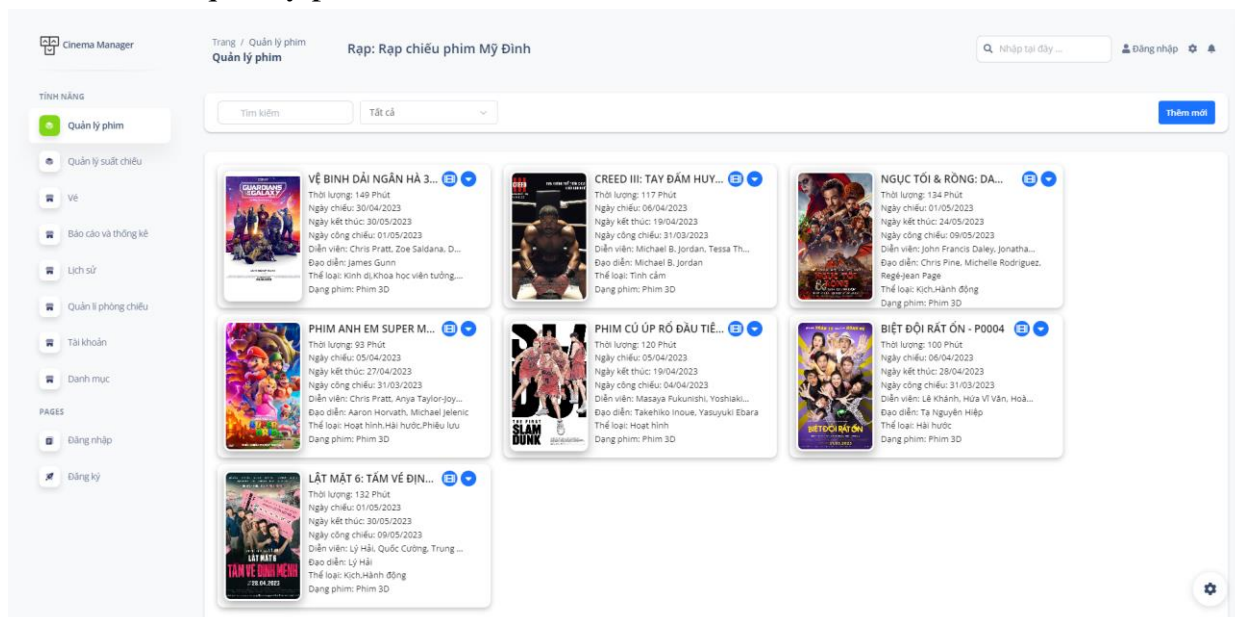
Hình 3.2 Hình giao diện đăng nhập

- Giao diện đăng ký

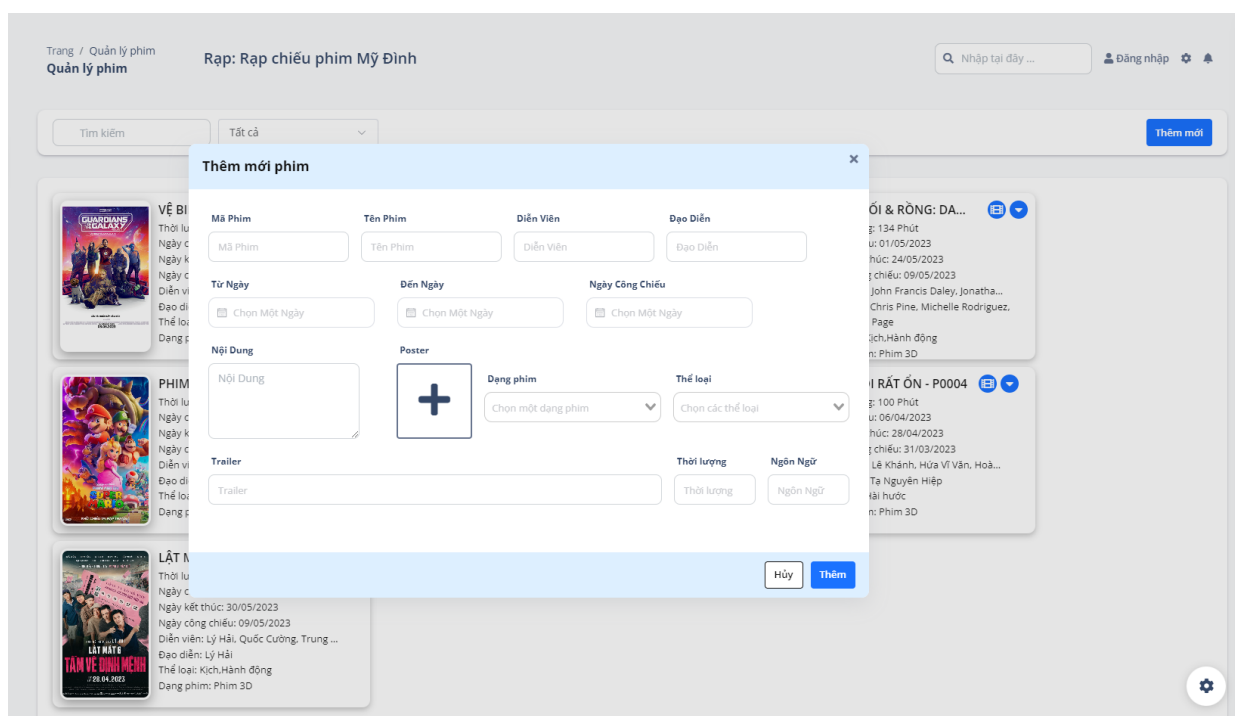


Hình 3.3 Hình giao diện đăng ký

- Giao diện quản lý phim:



Hình 3.4 Hình giao diện quản lý phim



Hình 3.5 Hình giao diện thêm mới/ sửa phim

- Giao diện quản lý suất chiếu:

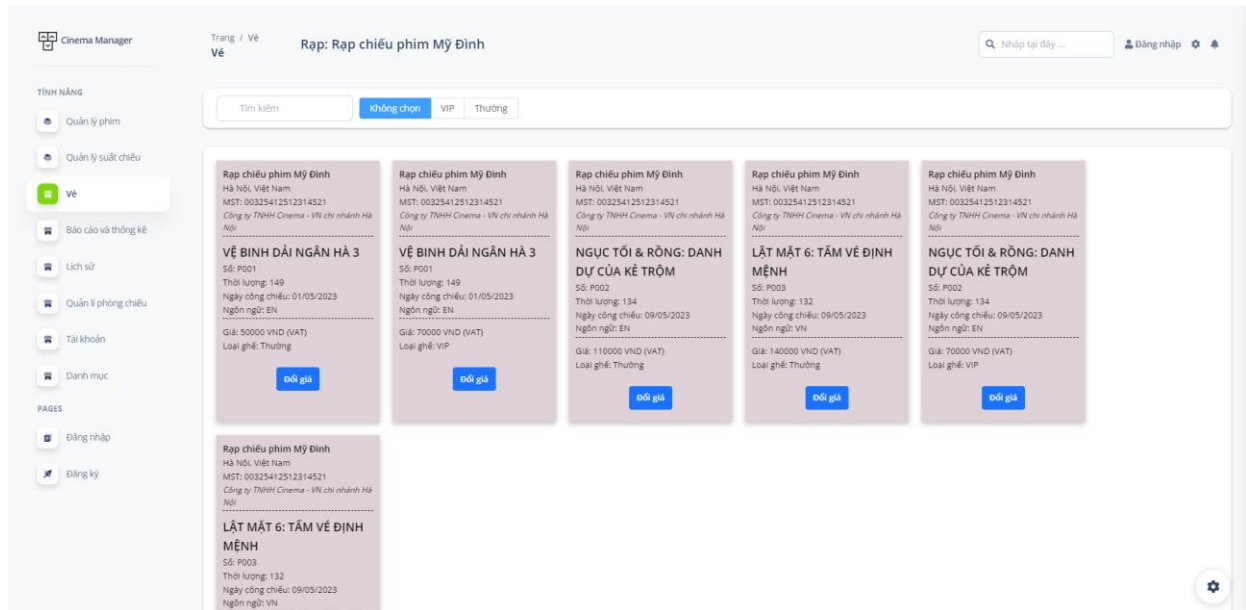


Hình 3.6 Hình danh sách suất chiếu



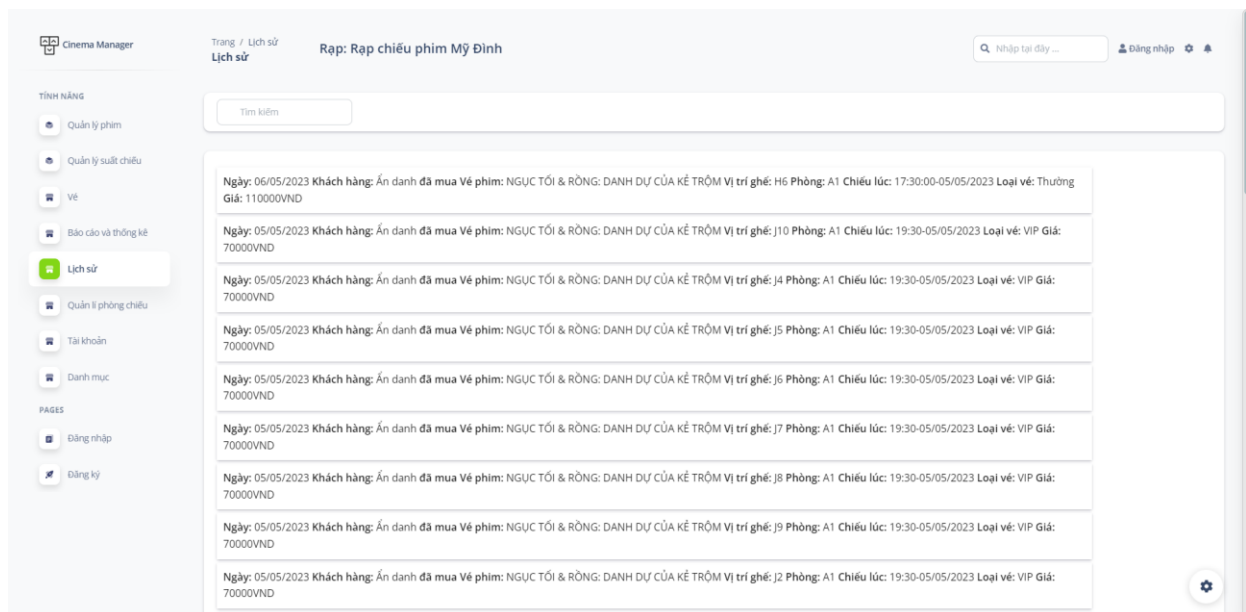
Hình 3.7 Hình giao diện chi tiết suất chiếu

- Giao diện quản lý vé:



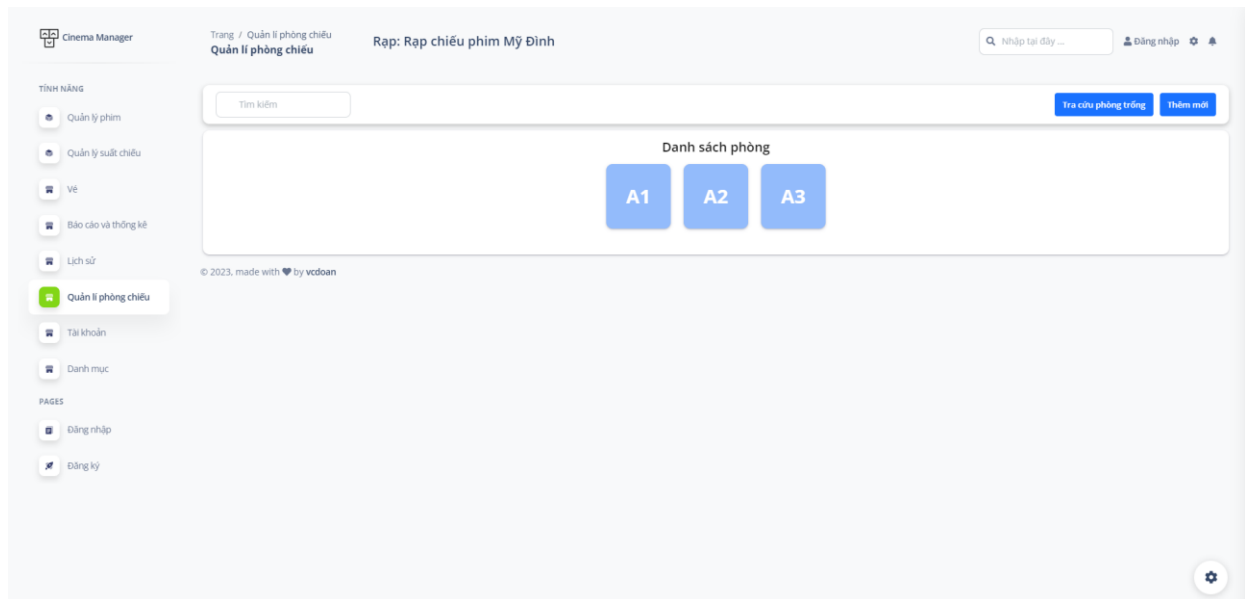
Hình 3.8 Hình giao diện quản lý vé

- Giao diện lịch sử:

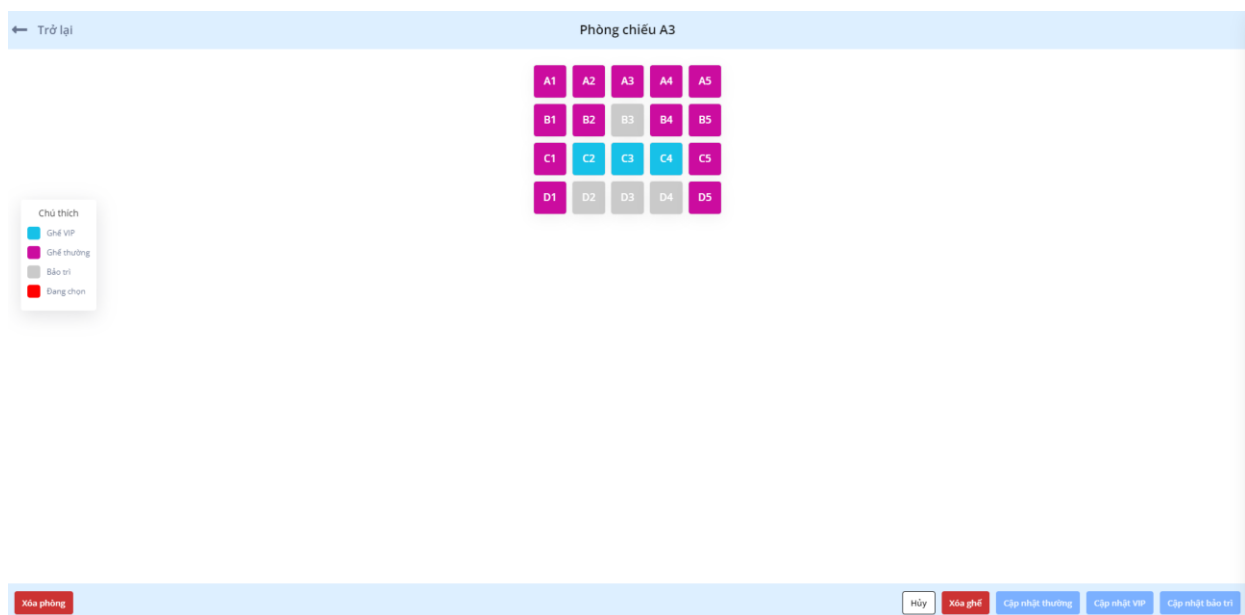


Hình 3.9 Hình giao diện lịch sử

- Giao diện quản lý phòng chiếu:

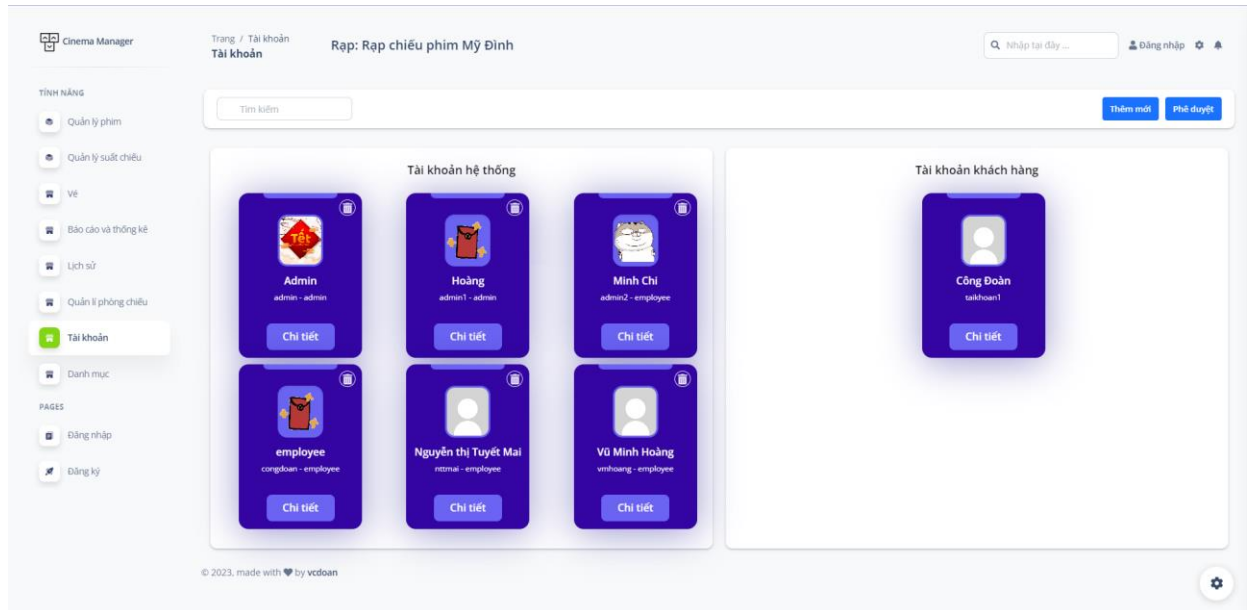


Hình 3.10 Hình giao diện quản lý phòng chiếu

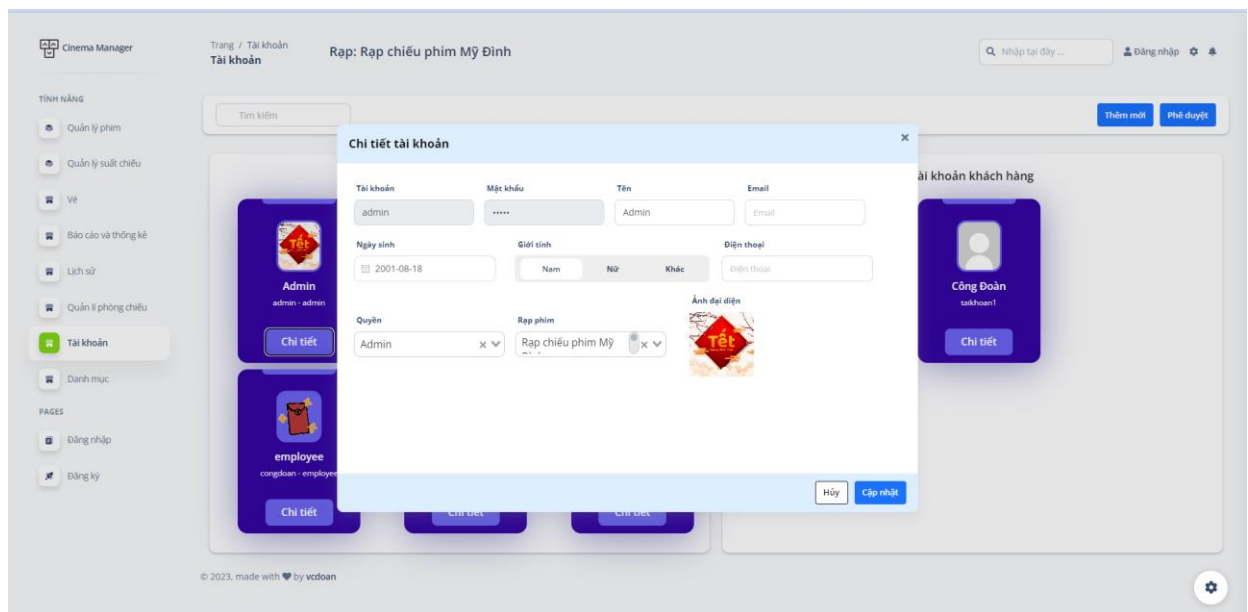


Hình 3.11 Hình giao diện chi tiết phòng chiếu

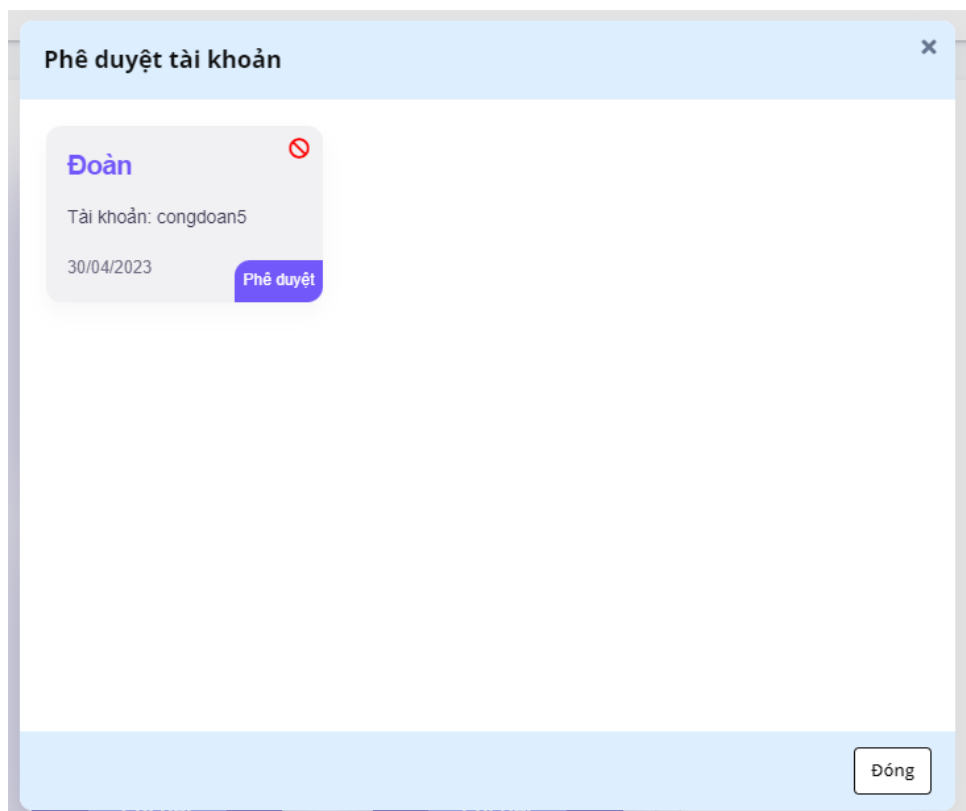
- Giao diện quản lý tài khoản:



Hình 3.12 Hình giao diện danh sách tài khoản

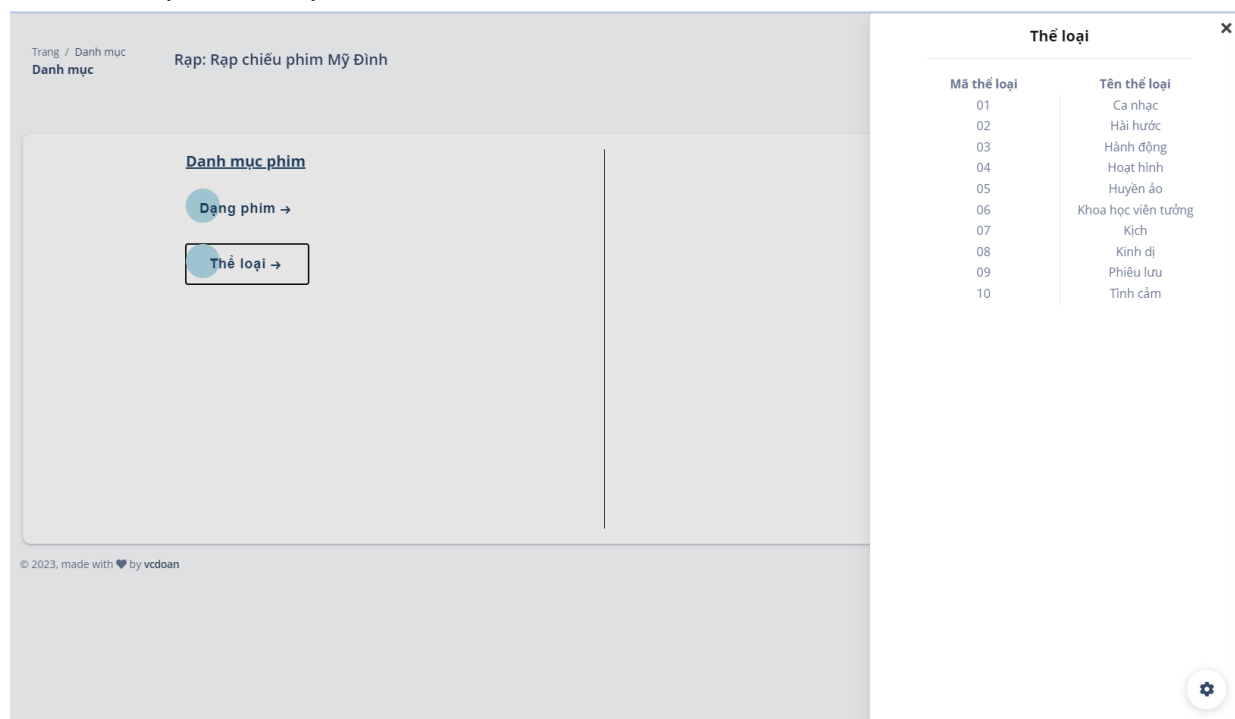


Hình 3.13 Giao diện chi tiết tài khoản



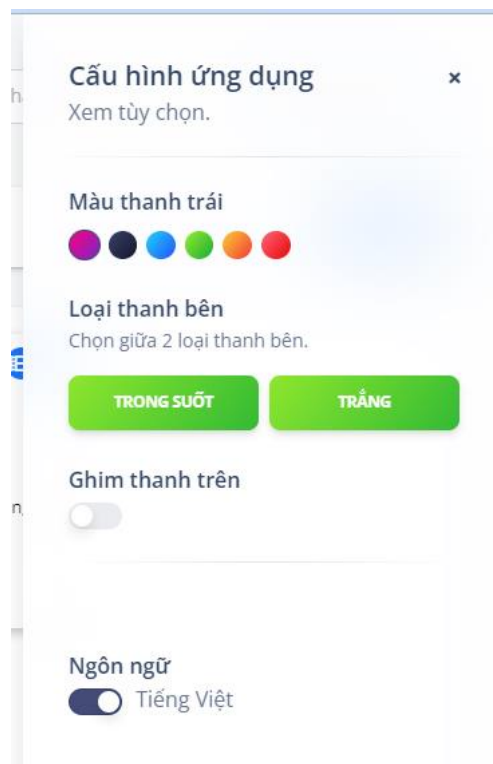
Hình 3.14 Hình giao diện phê duyệt tài khoản

- Giao diện danh mục:



Hình 3.15 Hình giao diện danh mục

- Giao diện tùy chọn



Hình 3.16 Hình giao diện tùy chọn

KẾT LUẬN VÀ KIẾN NGHỊ

Đánh giá kết quả đạt được

- Kết quả đạt được:
 - Xây dựng hoàn chỉnh website với các chức năng cơ bản theo yêu cầu ban đầu đã đặt ra, website thực hiện các chức năng thực hiện đúng nghiệp vụ.
 - Giao diện website khá đẹp mắt, thân thiện, dễ sử dụng.
 - Nắm rõ cách sử dụng các công cụ như Visual Studio, Visual Studio Code, Vue Devtool, Chrome Devtool để có thể đạt được hiệu quả cao nhất trong việc xây dựng website.
 - Nắm được quy trình xây dựng một trang web e-commerce.
 - Hiểu về framework Vue.js, các thư viện Vuex và cách sử dụng chúng để xây dựng website và ứng dụng vào trong sản phẩm của đồ án.
 - Hiểu về ASP.NET Web API, Dapper và sử dụng để xây dựng các API cho trang web và ứng dụng vào trong đồ án.
 - Bên cạnh những kết quả đạt được đã nêu trên, việc nghiên cứu và thực hiện đồ án tốt nghiệp này cũng là cơ hội để em trau dồi và hoàn thiện thêm kỹ năng lập trình cho bản thân.
- Những điểm cần cải tiến:
 - Tính năng tiện ích của chương trình chưa được tối ưu.
 - Chưa đảm bảo việc tối ưu hoá về mặt hiệu năng, bộ nhớ và tốc độ xử lý của chương trình.

Phương hướng phát triển

- Phát triển thêm các chức năng khác để website trở nên tiện ích và đa dụng hơn như: xây dựng chức năng thông tin nhân viên, kiểm soát sự cố, chat online với nhân viên kỹ thuật,...
- Tối ưu hiệu năng, tốc độ xử lý để trang web có thể hoạt động tốt trên môi trường thực tế với lượng dữ liệu lớn.

TÀI LIỆU THAM KHẢO

Tài liệu:

- [1] Kouraklis, J., 2016. MVVM as Design Pattern. In MVVM in Delphi (pp. 1-12). Apress, Berkeley, CA.
- [2] Vue.js 3 By Example: Blueprints to learn Vue web development, full-stack development, and cross-platform development quickly.
- [3] Halliday, P., 2018. Vue.js 2 Design Patterns and Best Practices: Build enterprise ready, modular Vue.js applications with Vuex and Nuxt. Packt Publishing Ltd.
- [4] Miller, M.A., Schwartz, T., Pickett, B.E., He, S., Klem, E.B., Scheuermann, R.H., Passarotti, M., Kaufman, S. and O'Leary, M.A., 2015. A RESTful API for access to phylogenetic tools via the CIPRES science gateway. Evolutionary
- [5] Object-Oriented and Classical Software Engineering, Stephen R. Schach, Eighth Edition, Mc Graw Hill, 2010.

Website:

- [1] <https://vuejs.org/guide>
- [2] <https://learn.microsoft.com/en-us/aspnet/core>
- [3] <https://learn.microsoft.com/en-us/ef/>
- [4] <https://vitejs.dev/guide/>
- [5] <https://vuexfire.vuejs.org/guide/getting-started.html>
- [6] <https://vuex.vuejs.org/guide/>