**TRƯỜNG ĐẠI HỌC THĂNG LONG**

**KHOA TOÁN TIN**



**BÁO CÁO BÀI TẬP LỚN**

**XÂY DỰNG ỨNG DỤNG HỖ TRỢ CÁC CÔNG VIỆC KINH DOANH CỦA RẠP CHIẾU PHIM**

**Nhóm thực hiện: A26167 – Nguyễn Đại Dương**

**A24664 – Nguyễn Thị Minh Loan**

**A25763 – Nguyễn Thị Trang Thư**

**A24764 – Lâm Khánh Linh**

**A24483 – Đỗ Thị Ánh**

**HÀ NỘI – 2017**

**MỤC LỤC**

[PHẦN 1. Tổng quan Hệ thống 1](#_Toc496616218)

[1.1. Giới thiệu bài toán 1](#_Toc496616219)

[1.1.1. Những khó khăn đối với Người quản lý cả Hệ thống rạp và Nhân viên phụ trách tại một chi nhánh rạp 1](#_Toc496616220)

[1.1.2. Khó khăn đối với Khách hàng – Người có nhu cầu xem phim tại rạp 2](#_Toc496616221)

[1.1.3. Hệ thống Cinema Booking 2](#_Toc496616222)

[1.2. Mô tả Hệ thống 2](#_Toc496616223)

[1.2.1. Phạm vi Hệ thống 2](#_Toc496616224)

[1.2.2. Các chức năng chính có trong Hệ thống 2](#_Toc496616225)

[1.2.3. Các đối tượng liên quan đến quá trình phát triển và sử dụng Hệ thống 3](#_Toc496616226)

[1.2.4. Usecase tổng quan của Hệ thống 3](#_Toc496616227)

[1.2.5. Quy trình nghiệp vụ cơ bản của Hệ thống 7](#_Toc496616228)

[1.3. Các loại mô hình sử dụng trong Hệ thống 8](#_Toc496616229)

[1.3.1. Mô hình Hoạt động 8](#_Toc496616230)

[1.3.2. Mô hình Usecase 8](#_Toc496616231)

[PHẦN 2. Kiến trúc tổng thể của hệ thống 10](#_Toc496616232)

[2.1. Sơ đồ kiến trúc tổng thể 10](#_Toc496616233)

[2.2. Mô hình lập trình 10](#_Toc496616234)

[2.2.1. Commons 11](#_Toc496616235)

[2.2.2. Endpoint 11](#_Toc496616236)

[2.2.3. DAO 11](#_Toc496616237)

[2.2.4. SERVICE 11](#_Toc496616238)

[2.3. Công nghệ sử dụng 11](#_Toc496616239)

[2.3.1. Java 11](#_Toc496616240)

[2.3.2. Spring framework 11](#_Toc496616241)

[2.3.3. Spring Boot 12](#_Toc496616242)

[2.3.4. Spring Rest 12](#_Toc496616243)

[2.3.5. Spring Security 14](#_Toc496616244)

[2.3.6. Hibernate ORM 14](#_Toc496616245)

[2.3.7. Spring LDAP 16](#_Toc496616246)

[2.3.8. Swagger 2.0 19](#_Toc496616247)

[2.3.9. Angular 4 20](#_Toc496616248)

[PHẦN 3. Thiết kế api trong hệ thống 22](#_Toc496616249)

[3.1. Các tài nguyên trong hệ thống 22](#_Toc496616250)

[3.2. Danh sách các API 23](#_Toc496616251)

[PHẦN 4. Thiết kế cơ sở dữ liệu 62](#_Toc496616252)

[PHẦN 5. Đặc Tả Các chức năng trong Hệ thống 63](#_Toc496616253)

[5.1. UC#001: Đăng nhập 63](#_Toc496616254)

[5.2. UC#002: Quản lý thành phố 70](#_Toc496616255)

[5.3. UC#003: Quản lý quận huyện 79](#_Toc496616256)

[5.4. UC#004: Quản lý chi nhánh rạp chiếu phim 88](#_Toc496616257)

[5.5. UC#005: Quản lý tài khoản 95](#_Toc496616258)

[5.6. UC#006: Quản lý thể loại phim 102](#_Toc496616259)

[5.7. UC#007: Quản lý quốc gia sản xuất phim 110](#_Toc496616260)

[5.8. UC#008: Quản lý diễn viên 118](#_Toc496616261)

[5.9. UC#009: Quản lý phim 125](#_Toc496616262)

[5.10. UC#010: Quản lý loại phòng 135](#_Toc496616263)

[5.11. UC#011: Quản lý phòng 138](#_Toc496616264)

[5.12. UC#012: Quản lý ghế 146](#_Toc496616265)

[5.13. UC#013: Quản lý suất chiếu 149](#_Toc496616266)

[5.14. UC#014: Quản lý giá vé xem phim 156](#_Toc496616267)

[5.15. UC#015: Tìm kiếm 165](#_Toc496616268)

[5.16. UC#016: Đăng ký tài khoản KH 168](#_Toc496616269)

[5.17. UC#017: Đặt vé 170](#_Toc496616270)

[5.18. UC#018: Quản lý thanh toán 177](#_Toc496616271)

[5.19. UC#019: Đăng xuất 183](#_Toc496616272)

**DANH MỤC ĐỊNH NGHĨA, THUẬT NGỮ, VIẾT TẮT**

|  |  |
| --- | --- |
| **Từ viết tắt** | **Nội dung** |
| DTO | Data Transfer Object |
| REST | Reppresentational State Transfer |
| ORM | Object-Relational Mapping |
| RDBMS | Database Management System |
| API | Application Program Interface |
| DB | Database |
| JDBC | Java Database Connectiviy |
| AD | Active Directory |
| OU | Organizational Unit |
| LDAP | Lightweight Directory Access Protocol |

**DANH MỤC HÌNH ẢNH**

[Hình 1.1 Sơ đồ Usecase Tổng quan cho Hệ thống 6](#_Toc496616273)

[Hình 1.2 Mô hình hóa quy trình đặt vé – thanh toán và thao tác xử lý 7](#_Toc496616274)

[Hình 2.1. Sơ đồ kiến trúc tổng thể 10](#_Toc496616275)

[Hình 2.2. Spring Modules 12](#_Toc496616276)

[Hình 2.3. Ví dụ publish Spring REST 14](#_Toc496616277)

[Hình 2.4. Phân vùng miền Active Directory 17](#_Toc496616278)

[Hình 2.5. Sự tương tác giữa client và server Directory 17](#_Toc496616279)

[Hình 2.6. LDAP mô hình dữ liệu một đơn vị (OU) 18](#_Toc496616280)

[Hình 2.7. LDAP mô hình xác định tên 18](#_Toc496616281)

[Hình 2.8. LDAP mô hình an ninh, đăng nhập với tài khoản được cấp 19](#_Toc496616282)

[Hình 2.9. Swagger UI 20](#_Toc496616283)

[Hình 2.10. Tổng quan kiến trúc AngularJS 1 21](#_Toc496616284)

[Hình 4.1. Thiết kế cơ sở dữ liệu 62](#_Toc496616285)

[Hình 5.1. Giao diện trang đăng nhập 66](#_Toc496616286)

[Hình 5.2. Đăng nhập thành công vào hệ thống 66](#_Toc496616287)

[Hình 5.3. Thông báo nhập nhiếu thông tin 67](#_Toc496616288)

[Hình 5.4. Thông báo nhập không đúng tên miền 67](#_Toc496616289)

[Hình 5.5. Thông báo lỗi đăng nhập 68](#_Toc496616290)

[Hình 5.6. Giao diện quản lý thành phố 74](#_Toc496616291)

[Hình 5.7. Giao diện xem chi tiết một thành phố 75](#_Toc496616292)

[Hình 5.8. Giao diện thêm mới một thành phố 75](#_Toc496616293)

[Hình 5.9. Giao diện chỉnh sửa thông tin thành phố 76](#_Toc496616294)

[Hình 5.10. Giao diện xóa 1 thành phố 76](#_Toc496616295)

[Hình 5.11. Giao diện xóa nhiều thành phố 77](#_Toc496616296)

[Hình 5.12. Giao diện tìm kiếm thành phố 77](#_Toc496616297)

[Hình 5.13. Giao diện quản lý quận huyện 83](#_Toc496616298)

[Hình 5.14. Giao diện xem chi tiết một quận huyện 83](#_Toc496616299)

[Hình 5.15. Giao diện thêm mới quận huyện 84](#_Toc496616300)

[Hình 5.16. Giao diện chỉnh sửa quận huyện 84](#_Toc496616301)

[Hình 5.17. Giao diện xóa một quận huyện 85](#_Toc496616302)

[Hình 5.18. Giao diện xóa nhiều quận huyện 85](#_Toc496616303)

[Hình 5.19. Giao diện tìm kiếm quận huyện 86](#_Toc496616304)

[Hình 5.20. Giao diện quản lý chi nhánh rạp chiếu phim 90](#_Toc496616305)

[Hình 5.21. Giao diện xem chi tiết một chi nhánh rạp chiếu phim 90](#_Toc496616306)

[Hình 5.22. Giao diện thêm mới chi nhánh rạp chiếu phim 91](#_Toc496616307)

[Hình 5.23. Giao diện chỉnh sửa chi nhánh rạp chiếu phim 91](#_Toc496616308)

[Hình 5.24. Giao diện xóa một chi nhánh rạp chiếu phim 92](#_Toc496616309)

[Hình 5.25. Giao diện xóa nhiều chi nhánh rạp chiếu phim 92](#_Toc496616310)

[Hình 5.26. Giao diện tìm kiếm chi nhánh rạp chiếu phim 93](#_Toc496616311)

[Hình 5.27. Giao diện quản lý thông tin cá nhận của user đăng nhập (1) 99](#_Toc496616312)

[Hình 5.28. Tab thay đổi ảnh đại diện (2) 99](#_Toc496616313)

[Hình 5.29. Tab thay đổi mật khẩu đăng nhập 100](#_Toc496616314)

[Hình 5.30. Giao diện quản lý loại phim 105](#_Toc496616315)

[Hình 5.31. Giao diện xem chi tiết một loại phim 105](#_Toc496616316)

[Hình 5.32. Giao diện thêm mới loại phim 106](#_Toc496616317)

[Hình 5.33. Giao diện chỉnh sửa loại phim 106](#_Toc496616318)

[Hình 5.34. Giao diện xóa một loại phim 107](#_Toc496616319)

[Hình 5.35. Giao diện xóa nhiều loại phim 107](#_Toc496616320)

[Hình 5.36. Giao diện tìm kiếm loại phim 108](#_Toc496616321)

[Hình 5.37. Giao diện quản lý quốc gia 113](#_Toc496616322)

[Hình 5.38. Giao diện xem chi tiết một quốc gia 113](#_Toc496616323)

[Hình 5.39. Giao diện thêm mới quốc gia 114](#_Toc496616324)

[Hình 5.40. Giao diện chỉnh sửa quốc gia 114](#_Toc496616325)

[Hình 5.41. Giao diện xóa một quốc gia 115](#_Toc496616326)

[Hình 5.42. Giao diện xóa nhiều quốc gia 115](#_Toc496616327)

[Hình 5.43. Giao diện tìm kiếm quốc gia 116](#_Toc496616328)

[Hình 5.44. Giao diện quản lý diễn viên 120](#_Toc496616329)

[Hình 5.45. Giao diện xem chi tiết một diễn viên 121](#_Toc496616330)

[Hình 5.46. Giao diện thêm mới diễn viên 121](#_Toc496616331)

[Hình 5.47. Giao diện chỉnh sửa diễn viên 122](#_Toc496616332)

[Hình 5.48. Giao diện xóa một diễn viên 122](#_Toc496616333)

[Hình 5.49. Giao diện xóa nhiều diễn viên 123](#_Toc496616334)

[Hình 5.50. Giao diện tìm kiếm diễn viên 123](#_Toc496616335)

[Hình 5.51. Giao diện quản lý phim 129](#_Toc496616336)

[Hình 5.52. Giao diện xem chi tiết một phim 130](#_Toc496616337)

[Hình 5.53. Giao diện thêm mới phim 130](#_Toc496616338)

[Hình 5.54. Giao diện chỉnh sửa phim 131](#_Toc496616339)

[Hình 5.55. Giao diện xóa một phim 131](#_Toc496616340)

[Hình 5.56. Giao diện xóa nhiều phim 132](#_Toc496616341)

[Hình 5.57. Giao diện tìm kiếm phim 132](#_Toc496616342)

[Hình 5.58. Giao diện quản lý phòng 140](#_Toc496616343)

[Hình 5.59. Giao diện xem chi tiết một phòng 141](#_Toc496616344)

[Hình 5.60. Giao diện thêm mới phòng 142](#_Toc496616345)

[Hình 5.61. Giao diện chỉnh sửa phòng 143](#_Toc496616346)

[Hình 5.62. Giao diện xóa một phòng 143](#_Toc496616347)

[Hình 5.63. Giao diện xóa nhiều phòng 144](#_Toc496616348)

[Hình 5.64. Giao diện tìm kiếm phòng 144](#_Toc496616349)

[Hình 5.65. Giao diện quản lý ghế 147](#_Toc496616350)

[Hình 5.66. Giao diện quản lý suất chiếu 151](#_Toc496616351)

[Hình 5.67. Giao diện xem chi tiết một suất chiếu 151](#_Toc496616352)

[Hình 5.68. Giao diện thêm mới suất chiếu 152](#_Toc496616353)

[Hình 5.69. Giao diện chỉnh sửa suất chiếu 152](#_Toc496616354)

[Hình 5.70. Giao diện xóa một suất chiếu 153](#_Toc496616355)

[Hình 5.71. Giao diện xóa nhiều suất chiếu 153](#_Toc496616356)

[Hình 5.72. Giao diện tìm kiếm suất chiếu 154](#_Toc496616357)

[Hình 5.73. Giao diện quản lý giá vé 160](#_Toc496616358)

[Hình 5.74. Giao diện xem chi tiết một giá vé 160](#_Toc496616359)

[Hình 5.75. Giao diện thêm mới giá vé 161](#_Toc496616360)

[Hình 5.76. Giao diện chỉnh sửa giá vé 161](#_Toc496616361)

[Hình 5.77. Giao diện xóa một giá vé 162](#_Toc496616362)

[Hình 5.78. Giao diện xóa nhiều giá vé 162](#_Toc496616363)

[Hình 5.79. Giao diện tìm kiếm giá vé 163](#_Toc496616364)

[Hình 5.80. Giao diện đăng xuất 184](#_Toc496616365)

# Tổng quan Hệ thống

## Giới thiệu bài toán

### Những khó khăn đối với Người quản lý cả Hệ thống rạp và Nhân viên phụ trách tại một chi nhánh rạp

Hiện nay, tại Việt nam có rất nhiều Hệ thống các cụm rạp chiếu phim xuất hiện để phục vụ nhu cầu xem phim màn ảnh rộng của đông đảo khán giả. Cùng với sự phát triển đó, khi số lượng chi nhánh rạp của Một hệ thống ngày càng tăng lên thì cũng đồng nghĩa với việc mức độ phức tạp trong quy trình quản lý cũng ngày càng khó hơn.

Mỗi chi nhánh rạp đặt tại một địa điểm khác nhau lại phải áp dụng một mô hình tổ chức rạp khác nhau sao cho phù hợp với nhu cầu của khán giá điạ phương. Chính vì vậy, đối với người quản lý rạp nói chung, điều này sẽ rất khó để tổ chức hoạt động, cũng như đánh giá năng suất kinh doanh. Ví dụ như: Các chi nhánh rạp thuộc các thành phố lớn thường có nhiều khung giờ chiếu hơn, thể loại phim phong phú đa dạng hơn đồng thời mức giá cũng cao hơn so với các cụm chi nhánh rạp thuộc vùng nông thôn.

Thông thường, các hoạt động quản lý và tổ chức rạp chiếu phim thường diễn ra như sau:

* Người quản lý tiến hành định hình cấu trúc rạp bao gồm: phân chia các phòng, số lượng ghế mỗi phòng, chất lượng dịch vụ cho mỗi phòng.
* Người quản lý tiến hành nhập và lưu trữ thông tin về các bộ phim sẽ công chiếu (sau phim đã được mua bản quyền từ nhà cung cấp)
* Người quản lý tiến hành sắp xếp các xuất chiếu theo từng bộ phim khác khau.
* Người quản lý tiến hành định giá vé phụ thuộc theo suất chiếu cũng như chất lượng phòng, chất lượng ghế.
* Người quản lý thông báo thông tin về phim, suất chiếu, giá vé cho nhân viên trong rạp để hỗ trợ họ trong việc phục vụ khách hàng
* Người quản lý đồng thời cũng phải tạo các chiến dịch quảng các để cung cấp các thông tin trên tới khách hàng nhằm hỗ trợ họ trong việc lựa chọn phim
* Trong thời gian phim chiếu, người quản lý có thể phải thay đổi vài thông tin quan trọng. Ví dụ: Thay đổi suất chiếu, thay giá vé dựa theo tỉ suất người xem phim. Những thay đổi đó phải được thông báo cho khách hàng, cũng như nhân viên trong rạp

Như vậy, để quản lý một chi nhánh rạp chiếu phim, Người quản lý sẽ cần lưu trữ tất cả các thông tin cơ bản về: phòng, ghế, phim, suất chiếu, ... Đồng thời cũng phải triển khai thông tin đó đến nhân viên và khách hàng một cách nhanh nhất, đặc biệt là những trường hợp thay đổi. Nếu các thao tác này chỉ được thực hiện thủ công thì sẽ không chỉ gây khó khăn cho việc quản lý mà còn ảnh hưởng tới doanh thu của rạp chiếu phim.

### Khó khăn đối với Khách hàng – Người có nhu cầu xem phim tại rạp

Đối với Khách hàng, trở ngại lớn nhất chính là các bước rườm ra trong thủ tục đặt phim:

* Khách hàng phải chủ động liên lạc thông qua việc gọi điện hỏi về phim đang chiếu, suất chiếu, cũng như giá vé.
* Khách hàng phải đặt vé và thanh toán trực tiếp tại quầy bán vé (trong trường hợp lượng người xem đông thì có thể phải chờ đợi và xếp hàng)
* Khách hàng khó nắm bắt những thay đổi về giá vé, suất chiếu, số lượng ghế còn lại trong rạp.

### Hệ thống Cinema Booking

Hệ thống quản lý rạp chiếu phim Cinema Booking được xây dựng dựa trên những nhu cầu thực tế của Nhà quản lý cũng như Khách hàng nhằm giải quyết khó khăn đang gặp phải, giảm thiểu và hạn chế các rủi ro trong quá trình hoạt động của rạp.

Mục đích của hệ thống Cinema Booking bao gồm

* Đối với Nhà quản lý: Hỗ trợ tối ưu nhằm tiết kiệm chi phí trong công tác quản lý, dịch vụ quảng cáo nhằm đưa thông tin phim tới khách hàng. Là phương thức góp phần thu hút nhiều khách hàng tới rạp từ đó tăng doanh thu cho rạp nhờ vào việc nâng cao chất lượng dịch vụ đặt vé và thanh toán
* Đối với Khách hàng: Thuận tiên hơn trong quá trình đặt phim và thanh toán.

Hệ thống sẽ được triển khai như một ứng dụng Web và một ứng dụng di động với giao diện thân thiện, gần gũi, dễ sử dụng. Trước mắt đội phát triển sẽ đặt mục tiêu hoàn thành Hệ thống trên Website.

## Mô tả Hệ thống

### Phạm vi Hệ thống

Hệ thống sẽ được triển khai cho một Công ty bao gồm nhiều chi nhánh khách nhau đặt tại những địa điểm khách nhau.

### Các chức năng chính có trong Hệ thống

Xây dựng một phần mềm quản lý rạp chiếu phim, bao gồm các chức năng chính được phân chia thành 2 loại theo 2 nhóm đối tượng cơ bản sử dụng dịch vụ như sau:

* Khách hàng: Xem và tìm kiếm các thông tin liên quan đến phim (nội dung phim, suất chiếu, ghế trống, giá vé); Đặt vé
* Nhân viên quản lý: Quản lý các thông tin liên quan đến phòng chiếu (ghế ngồi), suất chiếu, phim, giá vé; quản lý thông tin khách hàng; quản lý thông tin nhân viên; quản lý thanh toán.
* Ngoài ra, để Hệ thống có thể triển khai cho nhiều chi nhánh rạp khách nhau thì Hệ thống cần có một vài chức năng dành cho Người quản trị Hệ thống như: Quản lý chi nhánh rạp; quản lý quận huyện; quản lý tỉnh thành

### Các đối tượng liên quan đến quá trình phát triển và sử dụng Hệ thống

|  |  |  |
| --- | --- | --- |
| **Tên đối tượng liên quan** | **Đại diện** | **Trách nhiệm** |
| 1. Người sử dụng | Người quản trị Hệ thống | Quản lý chung các chi nhánh rạp của một công ty |
| Nhân viên quản lý rạp | Quản lý cho một chi nhánh rạp |
| Người dùng | Đặt vé, Thanh toán. |
| 2. Đội triển khai phần mềm | Nhóm phát triển phần mềm và nhân viên dùng thử của Hệ thống Cinema Booking | Chịu trách nhiệm thiết kế, xây dựng, bảo trì hệ thống |
| 3. Trưởng nhóm dự án | Trưởng nhóm dự án xây dựng phần mềm và Giám đốc công ty cung cấp dịch vụ rạp chiếu phim | Theo dõi tiến độ dự án |
| 4. Nhà tài trợ | Công ty cung cấp dịch vụ rạp chiếu phim | Hỗ trợ chi phí dự án |

### Usecase tổng quan của Hệ thống

*Mô tả các tác nhân trong Usecase:*

* Tác nhân Người quản trị Hệ thống: Phụ trách quản lý hoạt động và thông tin liên quan đến toàn bộ các chi nhánh rạp.
* Tác nhân Nhân viên Quản lý rạp: Phụ trách quản lý các các hoạt động và thông tin liên quan của duy nhất một chi nhánh rạp
* Tác nhân Khách hàng: Xem thông tin liên quan tới phim, suất chiếu từ đó quyết định Đặt vé và thanh toán trực tiếp trên Hệ thống

*Mô tả các Usecase trong hệ thống:*

***UC#001: Đăng nhập***

* Usecase này cho phép cả 3 tác nhân đăng nhập vào Hệ thống để sử dụng các chức năng khách trong đó:

2 tác nhân Người quản trị hệ thống và Nhân viên quản lý sẽ đăng nhập vào trang quản trị:

* Tác nhân Người quản trị Hệ thống sẽ tự tạo tài khoản cho mình.
* Tác nhân Nhân viên quản lý sẽ được Người quản trị Hệ thống tạo và cung cấp tài khoản. (Quy trình đăng kí tài khoản diễn ra bên ngoài luồng xử lý của Hệ thống)
* Tác nhân Khách hàng sẽ tự đăng kí tài khoản sử dụng với Hệ thống và đăng nhập vào trang Fronend tương tác với Khách Hàng.

***UC#002: Quản lý thành phố***

* Chức năng này cho phép người dùng thêm, xem chi tiết, sửa, xóa, tìm kiếm thành phố.

***UC#003: Quản lý quận huyện***

* Chức năng này cho phép người dùng thêm, xem chi tiết, sửa, xóa, tìm kiếm quận huyện.

***UC#004: Quản lý chi nhánh rạp chiếu phim***

* Use case này cho phép người quản trị quản lý các chi nhánh rạp chiếu phim trong hệ thống. Bao gồm việc thêm thay đổi và tìm kiếm thông tin chi nhánh trong hệ thống.

***UC#005: Quản lý tài khoản***

* Chức năng này cho phép Nhân viên quản lý thực hiện một số thao tác như: thêm mới tài khoản, cập nhật thông tin tài khoản, xem chi tiết tài khoản, tìm kiếm tài khoản, xóa tài khoản. Khách hàng cũng có thể thực hiện đổi mật khẩu, và cập nhật thông tin tài khoản của mình.

***UC#006: Quản lý thể loại phim***

* Chức năng này cho phép Nhân viên quản lý thực hiện một số chức năng: thêm mới thể loại phim, cập nhật thông tin thể loại phim, xem chi tiết thể loại phim, xóa thể loại phim, tìm kiếm thể loại phim.

***UC#007: Quản lý quốc gia***

* Chức năng này cho phép Quản trị hệ thống thực hiện một số chức năng: thêm, sửa, xóa, tìm kiếm các quốc gia sản xuất phim.

***UC#008: Quản lý diễn viên***

* Use case này cho phép người quản trị quản lý các diễn viên trong hệ thống. Bao gồm việc thêm thay đổi và tìm kiếm thông tin diễn viên trong hệ thống.

***UC#009: Quản lý phim***

* Usecase này cho phép người dùng quản lý thông tin liên quan đến các Phim được lưu trữ trong Hệ thống, bao gồm:

***UC#010: Quản lý loại phòng***

* Use case này cho phép người quản trị quản lý các loại phòng trong hệ thống. Bao gồm việc thêm, thay đổi và tìm kiếm thông tin loại phòng trong hệ thống.

***UC#0011: Quản lý phòng***

* Use case này cho phép người quản trị quản lý các phòng trong hệ thống. Bao gồm việc thêm, thay đổi và tìm kiếm thông tin phòng trong hệ thống.

***UC#012: Quản lý ghế***

* Use case này cho phép người quản trị quản lý các ghế trong hệ thống. Bao gồm việc thêm, thay đổi và tìm kiếm thông tin ghế trong hệ thống.

***UC#013: Quản lý suất chiếu***

* Use case này cho phép người quản trị quản lý các suất chiếu trong hệ thống. Bao gồm việc thêm, thay đổi và tìm kiếm thông tin suất chiếu trong hệ thống.

***UC#014: Quản lý giá vé xem phim***

* Usecase này cho phép người dùng quản lý thông tin liên quan đến các Phim được lưu trữ trong Hệ thống, bao gồm:

Thêm mới thông tin Giá Vé Xem Phim vào Hệ thống

Xóa bỏ thông tin Giá Vé Xem Phim khỏi Hệ thống

Sửa đổi (cập nhập) thông tin Giá Vé Xem Phim đã có trong Hệ thống

***UC#015: Tìm kiếm***

* Chức năng này cho phép Khách hàng có thể tìm kiếm phim, hệ thống rạp mà mình muốn xem.

***UC#016: Đăng ký tài khoản KH***

* Chức năng này cho phép khách hàng đăng ký một tài khoản để đăng nhập và sử dụng các dịch vụ của hệ thống.

***UC#017: Đặt vé***

* Chức năng này cho phép người dùng đặt vé xem phim trên hệ thống

***UC#0018: Quản lý thanh toán***

* Chức năng này cho phép người dùng thực hiện xem chi tiết hóa đơn, thanh toán, xóa hóa đơn phim

***UC#019:*** ***Đăng xuất***

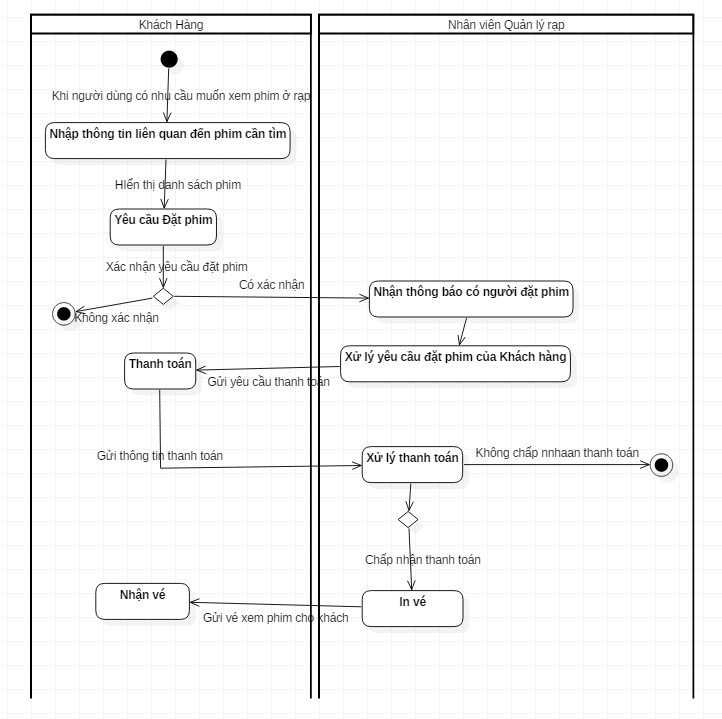
* Chức năng này cho phép người dùng có tài khoản đăng xuất khỏi hệ thống.



Hình 1.1 Sơ đồ Usecase Tổng quan cho Hệ thống

### Quy trình nghiệp vụ cơ bản của Hệ thống

* Quy trình nghiệp vụ đặt vé, thanh toán giữa Khách Hàng và Nhân viên quản lý



Hình 1.2 Mô hình hóa quy trình đặt vé – thanh toán và thao tác xử lý

## Các loại mô hình sử dụng trong Hệ thống

### Mô hình Hoạt động

* Các ký hiệu trong sơ đồ

|  |  |
| --- | --- |
| **Ký hiệu** | **Giải thích** |
|  | Nút Start thể hiện điểm bắt đầu quy trình |
|  | Activity mô tả một hoạt động trong hệ thống. Các hoạt động này do các đối tượng thực hiện. |
|  | Branch thể hiện rẽ nhánh trong mệnh đề điều kiện. |
|  | Dòng điều khiển |
|  | End thể hiện điểm kết thúc qui trình. |
|  | Fork thể hiện cho trường hợp thực hiện xong một hoạt động rồi sẽ rẽ nhánh thực hiện nhiều hoạt động tiếp theo |

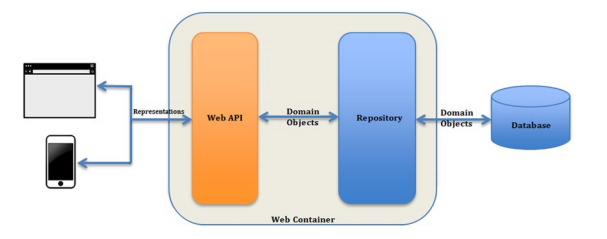
### Mô hình Usecase

* Các kí hiệu trong sơ đồ

|  |  |
| --- | --- |
| **Ký hiệu** | **Giải thích** |
|  | Actor được dùng để chỉ người sử dụng hoặc một đối tượng nào đó bên ngoài tương tác với hệ thống chúng ta đang xem xét. |
|  | Use Case là chức năng mà các Actor sẽ sử dụng. Nó được ký hiệu như sau |
|  | Association thường được dùng để mô tả mối quan hệ giữa Actor và Use Case và giữa các Use Case với nhau |
|  | Include là quan hệ giữa các Use Case với nhau, nó mô tả việc một Use Case lớn được chia ra thành các Use Case nhỏ để dễ cài đặt (module hóa) . |
|  | Generalization được sử dụng để thể hiện quan hệ thừa kế giữa các Actor hoặc giữa các Use Case với nhau |

# Kiến trúc tổng thể của hệ thống

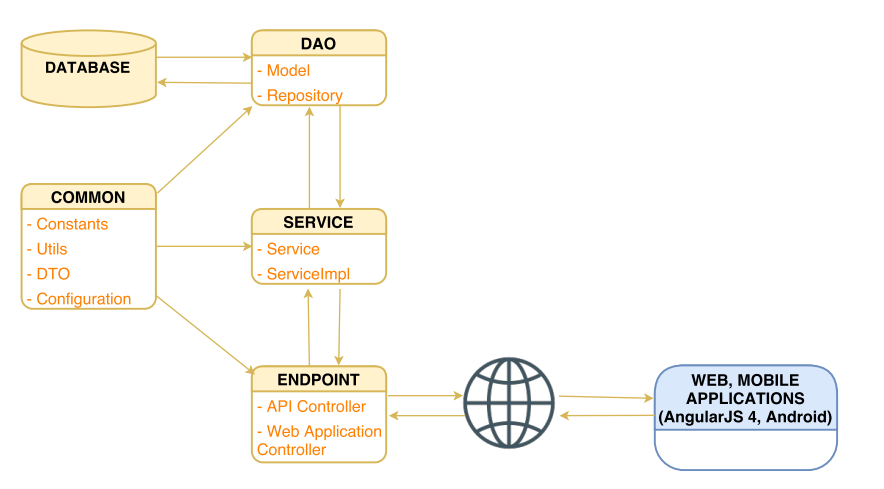
## Sơ đồ kiến trúc tổng thể



Hình 2.1. Sơ đồ kiến trúc tổng thể

Hệ thống được thiết kế theo mô hình client-server. Mô hình client-server là một kiến trúc phân phối ứng dụng, phân chia nhiệm vụ/công việc giữa nhà cung cấp của một tài nguyên/dịch vụ (server) và người yêu cầu dịch vụ (client).

## Mô hình lập trình



Hình 2 Mô hình lập trình

### Commons

Common chứa *Constants*, *Utils* (các method hay dùng cho nhiều mục đích)

và DTO (Data Transfer Object)

### Endpoint

* Chứa các file HTML, CSS, JS, Template…
* Các hàm Controller, các API
* Giao tiếp với người dùng cuối để thu thập dữ liệu và hiển thị kết quả/dữ liệu thông qua các thành phần trong giao diện người sử dụng hoặc thông qua API
* Thao tác với tầng service để trao đổi dữ liệu.

### DAO

Thực hiện các nghiệp vụ liên quan đến lưu trữ và truy xuất dữ liệu của ứng dụng như đọc, lưu, cập nhật cơ sở dữ liệu.

### SERVICE

* Xử lý chính các dữ liệu trước khi được đưa lên hiển thị trên màn hình hoặc xử lý các dữ liệu trước khi chuyển xuống DAO để lưu dữ liệu xuống cơ sở dữ liệu.
* Kiểm tra ràng buộc, các yêu cầu nghiệp vụ, tính toán, xử lý các yêu cầu và lựa chọn kết quả trả về cho EndPoint.
* Mapping DTO và Entity.

## Công nghệ sử dụng

### Java

Java là ngôn ngữ lập trình cấp cao ban đầu được phát triển bởi Sun Microsystems. Java chạy trên đa nền tảng. Trong hệ thống này, java dùng để xây dựng:

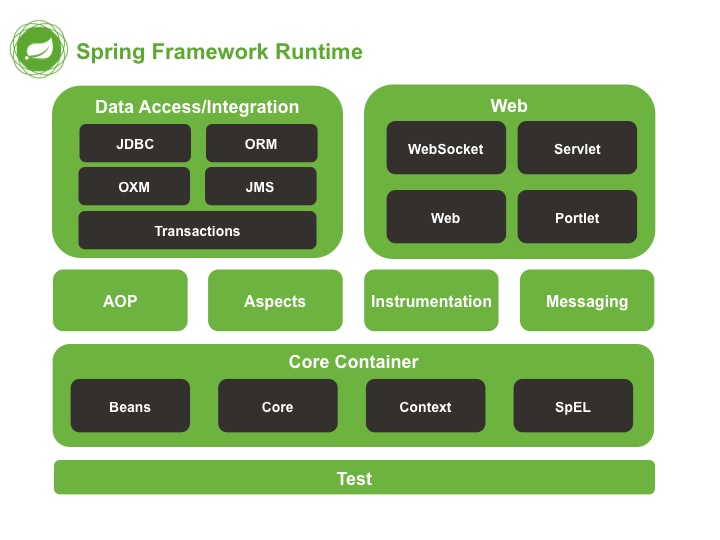
* Phần Backend Services.
* Android Mobile App.

### Spring framework

Spring được mô tả là một lightweight framework để xây dựng các ứng dụng Java, nó có 2 điểm đáng quan tâm ở đây:

* Ta có thể dùng Spring để xây dựng bất kỳ ứng dụng nào trong Java (Ví dụ: stand-alone, web hay các ứng dụng Java Enterprise Edition (JEE))
* Lightweight framework có ý nghĩa là những thay đổi đến mã ứng dụng để đạt được những ích mà Spring mang lại.

Spring Framework bao gồm các tính năng được tổ chức thành khoảng 20 mô-đun.



Hình 2.2. Spring Modules

### Spring Boot

Là một cách mới để tạo các ứng dụng Spring nhanh chóng. Với Spring lập trình viên thường phải cấu hình các phụ thuộc (dependency, jars) thủ công thông qua các file XML hay chú giải annotations thì với Spring Boot mọi thứ được tự động hóa. Một vài tính năng nổi bật như:

* Tạo ứng dụng Spring sẵn sàng để phát hành;
* Tạo ứng dụng standalone dùng server nhúng sẵn;
* Một trong những tính năng là công nghệ “opinioned” giúp ta làm theo hướng làm tốt nhất cho việc tạo ứng dụng có khả năng mở rộng và mạnh mẽ.

### Spring Rest

Là phần Endpoint trong mô hình lập trình. Đây là một mô đun trong hệ sinh thái Spring Framework dùng để xây dựng RESTful API.

#### Tìm hiểu về RestFul

Bản thân REST không phải là một loại công nghệ, mà là phương thức tạo API với nguyên lý tổ chức nhất định. Những nguyên lý này nhằm hướng dẫn lập trình viên tạo môi trường xử lý API request được toàn diện hơn.

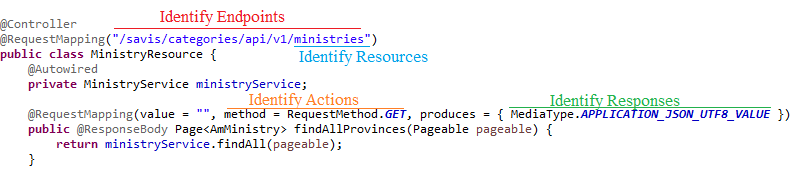
REST là viết tắt của Reppresentational State Transfer và là một kiểu kiến trúc để thiết kế các ứng dụng mạng phân tán. *Roy Fielding* đặt ra thuật ngữ REST trong luận văn tiến sĩ của mình và đề xuất sáu trở ngại hoặc nguyên tắc dưới đây làm căn cứ   
của nó:

* **Client-Server**: Những thứ liên quan nên được tách biệt giữa các máy khách và máy chủ. Điều này cho phép các thành phần máy khách và máy chủ phát triển độc lập và lần lượt cho phép hệ thống có thể mở rộng quy mô.
* **Stateless**: Sự giao tiếp giữa máy khách và máy chủ nên là vô trạng thái. Máy chủ không cần phải nhớ trạng thái của máy khách. Thay vào đó, khách hàng phải bao gồm tất cả các thông tin cần thiết trong yêu cầu để máy chủ có thể hiểu và xử lý nó.
* **Layered System**: Nhiều lớp phân cấp như cổng (gateways), tường lửa (firewalls) và proxies có thể tồn tại giữa máy khách và máy chủ. Các lớp có thể được thêm, sửa đổi, sắp xếp lại, hoặc xoá một cách minh bạch để cải thiện khả năng mở rộng.
* **Cache**: Các phản hồi (responses) từ máy chủ phải được khai báo là có thể lưu trữ được hoặc không thể lưu trữ. Điều này sẽ cho phép các client hoặc các thành phần trung gian của nó để lưu trữ phản hồi (responses) và dùng lại chúng cho các yêu cầu (requests) sau này. Điều này làm giảm tải (load) trên máy chủ và giúp cải thiện hiệu suất.
* **Uniform Interface**: Tất cả các tương tác giữa client, máy chủ, và các thành phần trung gian được dựa trên sự thống nhất của các giao diện của chúng. Điều này đơn giản hóa kiến trúc tổng thể vì các thành phần có thể phát triển độc lập miễn là chúng thực hiện theo quy tắc đặt ra.
* **Code on demand**: Client có thể mở rộng chức năng của chúng bằng cách tải xuống (download) và thực thi (execute) mã theo yêu cầu. Ví dụ bao gồm các tập lệnh JavaScript, các Applet Java, Silverlight, v.v. Đây là một ràng buộc tùy chọn.

#### Xây dựng RESTful API

Thiết kế và triển khai một API RESTful tốt không chỉ là một nghệ thuật. Phải mất thời gian, nỗ lực, và một số lặp đi lặp lại. Một API RESTful được thiết kế tốt cho phép người dùng cuối của bạn sử dụng API một cách dễ dàng và làm cho việc áp dụng nó trở nên dễ dàng hơn. Ở mức độ cao, dưới đây là các bước liên quan đến việc xây dựng API RESTful:

* Identify Resources: Trung tâm đến REST là các tài nguyên. Ta bắt đầu tạo mẫu các tài nguyên khác nhau mà đến người sử dụng quan tâm. Thông thường, các tài nguyên này có thể là miền hoặc các thực thể của ứng dụng.
* Identify Endpoints: Bước tiếp theo là thiết kế các URI ánh xạ các tài nguyên tới các endpoints.
* Identify Actions: Xác định các phương thức HTTP có thể được sử dụng để thực hiện các hoạt động trên các tài nguyên.
* Identify Responses: Xác định các thể hiện (reprentation) tài nguyên được hỗ trợ cho yêu cầu và đáp ứng cùng với mã trạng thái phù hợp để được trả lại.



Hình 2.3. Ví dụ publish Spring REST

### Spring Security

Spring Security cung cấp các dịch vụ bảo mật toàn diện cho các ứng dụng phần mềm doanh nghiệp dựa trên Java EE. Có một sự nhấn mạnh đặc biệt về hỗ trợ các dự án được xây dựng bằng Spring Framework, đây là giải pháp Java EE hàng đầu cho phát triển phần mềm doanh nghiệp.

Hệ thống Cinema Booking System sẽ được bảo mật với:

* Tích hợp công nghệ SSL (Secure Socket Layer) để thiết lập liên kết được mã hóa giao tiếp giữa client và server.
* Xác thực người dùng với Spring Security Outhen2 cho phép các ứng giới hạn các truy cập đến người dùng dịch vụ (HTTP service).

### Hibernate ORM

ORM (Object-Relational Mapping), một kỹ thuật lập trình để chuyển đổi dữ liệu giữa DB và các ngôn ngữ lập trình hướng đối tượng như Java, C#, …

Hibernate là một giải pháp ORM cho Java và được biết đến với tên gọi Open Source Persistent Framework. Hibernate ánh xạ các lớp Java đến các bảng trong DB và từ các kiểu dữ liệu Java đến các kiểu dữ liệu SQL.

#### Tại sao nên dùng Hibernate

Hibernate cung cấp nhiều lợi ích nhà phát triển, một trong số đó là:

* **Relational Persistence for JAVA**: Làm việc với cả phần mềm hướng đối tượng và cơ sở dữ liệu quan hệ là một công việc phức tạp với JDBC[[1]](#footnote-1) vì có sự không phù hợp giữa cách dữ liệu được thể hiện trong các đối tượng so với cơ sở dữ liệu quan hệ. Vì vậy, với JDBC, nhà phát triển phải viết mã để ánh xạ một mô hình dữ liệu mô hình đối tượng sang mô hình dữ liệu quan hệ và lược đồ cơ sở dữ liệu tương ứng của nó. Hibernate là giải pháp ORM linh hoạt và mạnh mẽ để ánh xạ các lớp Java tới các bảng cơ sở dữ liệu. Hibernate tự quản lý việc ánh xạ này bằng cách sử dụng các tệp XML vì vậy nhà phát triển không cần phải viết mã cho điều này.
* **Transparent Persistence**: Việc ánh xạ tự động các đối tượng Java với các bảng cơ sở dữ liệu và ngược lại được gọi là Sự trong suốt trong suốt (Transparent Persistence). Hibernate cung cấp sự bền bỉ trong suốt và nhà phát triển không cần phải viết mã một cách rõ ràng để lập bản đồ các bảng cơ sở dữ liệu cho các đối tượng ứng dụng trong tương tác với RDBMS. Với JDBC chuyển đổi này sẽ được các nhà phát triển làm thủ công với nhiều dòng mã.
* **Support for Query Language**: JDBC chỉ hỗ trợ ngôn ngữ truy vấn có cấu trúc gốc (SQL). Nhà phát triển phải tìm ra cách hiệu quả để truy cập cơ sở dữ liệu, tức là chọn truy vấn hiệu quả từ một số truy vấn để thực hiện cùng một tác vụ. Hibernate cung cấp một ngôn ngữ truy vấn mạnh mẽ Hibernate Query Language (độc lập với loại cơ sở dữ liệu) được thể hiện trong cú pháp quen thuộc giống như SQL và bao gồm hỗ trợ đầy đủ các truy vấn đa hình. Hibernate cũng hỗ trợ các câu lệnh SQL thuần. Nó cũng chọn một cách hiệu quả để thực hiện một thao tác cơ sở dữ liệu cho một ứng dụng.
* **Maintenance Cost**: Với JDBC, trách nhiệm của nhà phát triển là phải xử lý kết quả của JDBC và chuyển nó sang đối tượng Java thông qua mã để sử dụng dữ liệu liên tục trong ứng dụng. Vì vậy, với JDBC, ánh xạ giữa các đối tượng Java và các bảng cơ sở dữ liệu được thực hiện thủ công. Hibernate làm giảm các dòng mã bằng cách duy trì ánh xạ bảng đối tượng và trả lại kết quả cho ứng dụng dưới dạng các đối tượng Java. Nó giải phóng lập trình từ xử lý thủ công dữ liệu liên tục, do đó giảm thời gian phát triển và chi phí bảo trì.

### Spring LDAP

#### Active Directory là gì

Nếu không có Active Directory, các máy tính trong văn phòng của ta được thiết lập trong các nhóm làm việc: các thiết bị cá nhân đang chạy trên cùng một mạng LAN (Local Area Network). Điều này có nghĩa là mỗi máy tính là một máy độc lập, chúng không giao tiếp với nhau dễ dàng và yêu cầu người dùng phải trải qua quá trình xác thực mỗi lần bạn muốn trao đổi thông tin từ người khác. Không có tài khoản toàn cục liên kết tất cả các máy với nhau.

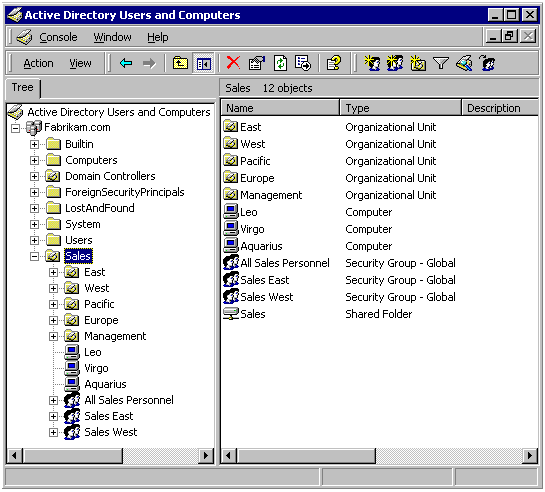
Giả sử bạn có một văn phòng với năm nhân viên, và bạn nghĩ rằng tích hợp Active Directory là không cần thiết, bởi vì hoạt động của bạn là quá nhỏ. Trong trường hợp đó, bạn sẽ có năm nhóm làm việc độc lập được thiết lập-một cho mỗi máy. Nhân viên của bạn, *Steve, Phil, Sue, Linda* và *Ed* đều có tên người dùng và mật khẩu riêng biệt được lưu trữ trong máy của họ, những tài khoản này sử dụng để đăng nhập vào máy tính   
của họ.

Mỗi máy tính không biết được sự tồn tại của máy khác khi liên lạc qua mạng, nghĩa là mỗi người dùng muốn truy cập vào thông tin từ một máy khác, một máy cần phải liên lạc với người kia và chứng thực sự trao đổi. Để Steve kết nối với máy của Sue, anh ta cần phải biết địa chỉ IP của cô, và sau đó cần phải nhập tên người dùng và mật khẩu đã được thiết lập thông qua máy của cô ấy.

#### Tại sao các tổ chức cần Active Directory

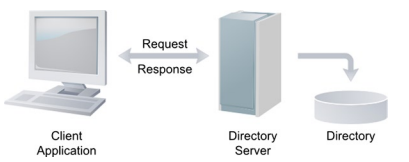
Nếu văn phòng của bạn sử dụng Active Directory, tất cả các máy sẽ được kết nối trên một miền, có nghĩa là tất cả các thông tin được lưu trữ ở một vị trí trung tâm chứ không phải cục bộ trên các ổ cứng máy tính cá nhân. Miền được kiểm soát bởi một danh mục toàn cục, theo dõi tất cả các thiết bị được đăng ký vào mạng. Cửa hàng toàn cầu lưu trữ các địa chỉ IP, tên máy tính và người dùng để quản trị viên toàn cục có thể giám sát mọi thứ xảy ra trên miền. Để truy cập máy tính của người khác, người dùng chỉ cần tên máy tính đó, bởi vì mọi thứ đã được liên kết ở cuối (backend).

Khi bạn đang sử dụng Active Directory, mọi thứ đã được phép từ bộ điều khiển tên miền. Điều này có nghĩa là quản trị viên mạng đã yêu cầu bộ điều khiển tên miền những gì cho phép gán cho mỗi người dùng. Điều này làm cho tất cả các truyền thông kỹ thuật số hiệu quả hơn, bởi vì mọi thứ đều dễ dàng tiếp cận, và thông tin có sẵn.



Hình 2.4. Phân vùng miền Active Directory

#### LDAP là gì



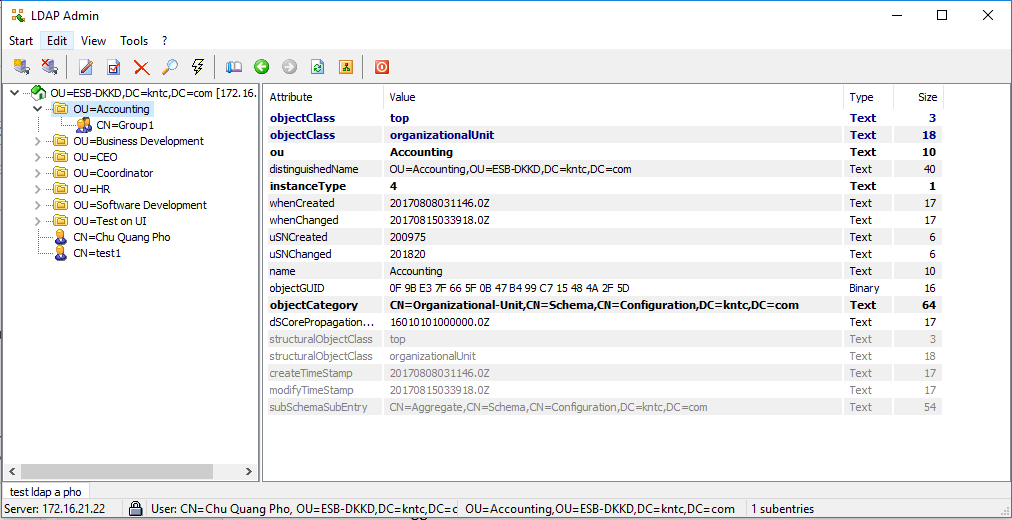
Hình 2.5. Sự tương tác giữa client và server Directory

Lightweight Directory Access Protocol (LDAP) cung cấp mô hình giao thức chuẩn cho giao tiếp với directory.

LDAP định nghĩa một giao thức thông điệp được sử dụng bởi các trình khách thư mục và các máy chủ thư mục. LDAP có thể được hiểu rõ hơn bằng cách xem xét bốn mô hình sau đây dựa trên đó:

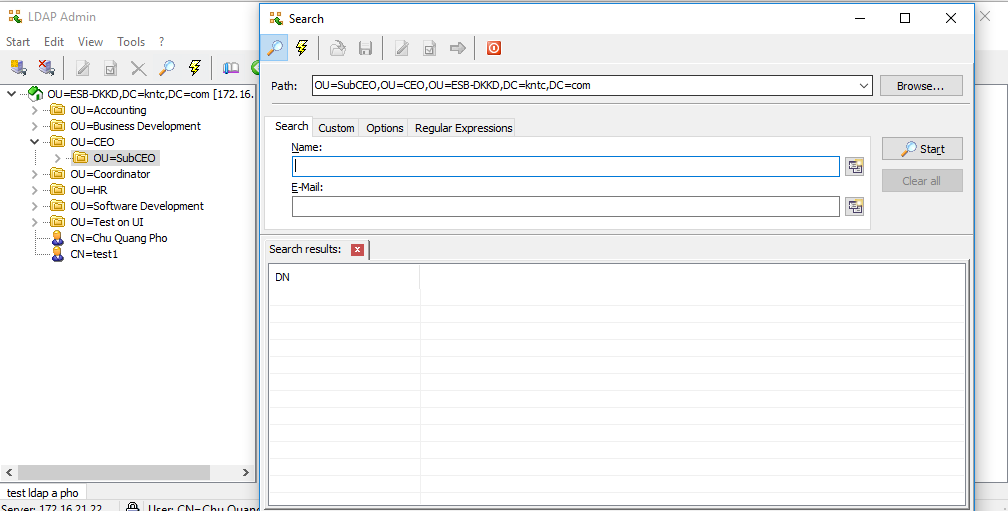
* Mô hình thông tin xác định cấu trúc của thông tin được lưu trữ trong thư mục.

Ví dụ: Mô hình này tương tự các lớp (class) trong các ngôn ngữ lập trình, ta xác định tên lớp, các thuộc tính.



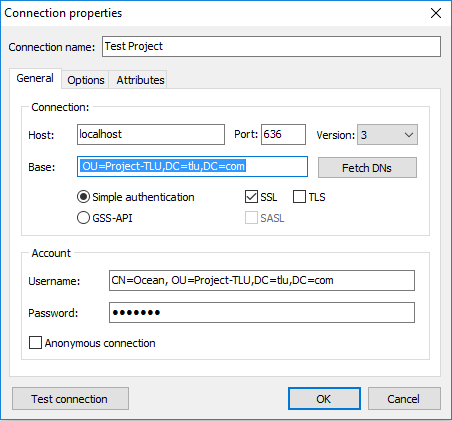
Hình 2.6. LDAP mô hình dữ liệu một đơn vị (OU[[2]](#footnote-2))

* Mô hình đặt tên xác định cách thông tin được tổ chức và xác định trong thư mục. Ví dụ dưới đây OU tên là ‘SubCEO’ thuộc OU là ‘CEO’ xác định bằng đường dẫn duy nhất như sau:



Hình 2.7. LDAP mô hình xác định tên

* Mô hình chức năng xác định các hoạt động có thể được thực hiện trên thư mục. Chẳng ta có thể thực hiện sửa dữ liệu một đơn vị hay tìm kiếm users.
* Mô hình an ninh xác định làm thế nào để bảo vệ thông tin khỏi những truy cập trái phép.



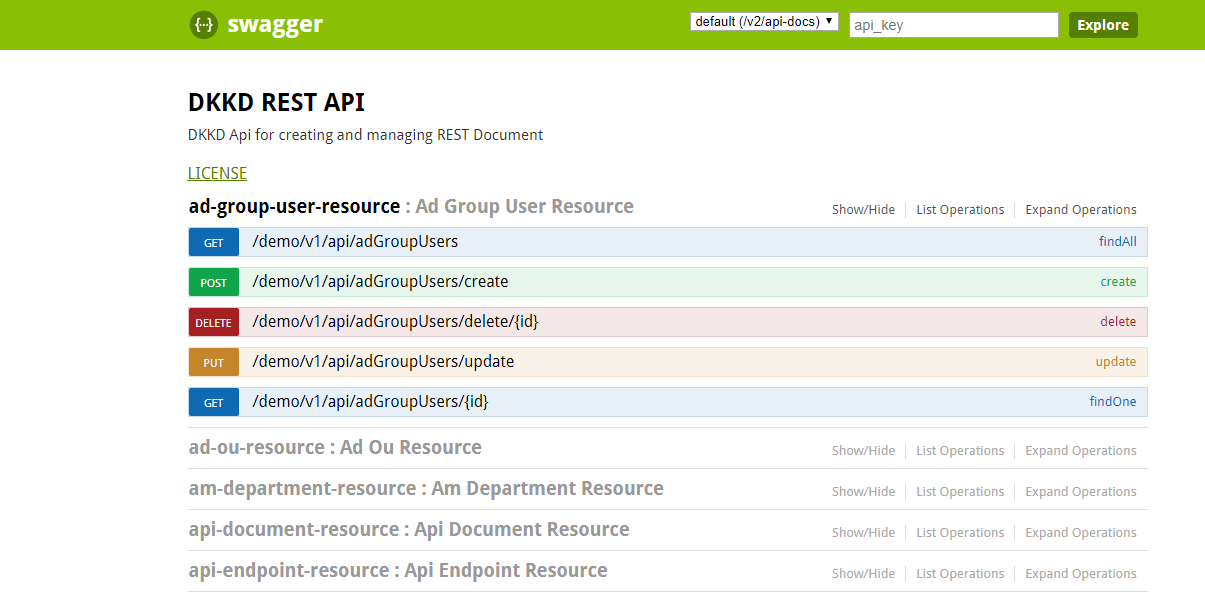
Hình 2.8. LDAP mô hình an ninh, đăng nhập với tài khoản được cấp

**Spring LDAP** là một thư viện cho việc lập trình LDAP đơn giản hơn trong Java.

### Swagger 2.0

Tài liệu là một khía cạnh quan trọng của bất kỳ dự án nào. Điều này đặc biệt đúng đối với doanh nghiệp và các dự án nguồn mở, trong đó nhiều người cùng hợp tác để xây dựng dự án. Swagger một công cụ đơn giản hoá tài liệu REST API.

Swagger (http://swagger.io) là một đặc điểm kỹ thuật và một framework để tạo tài liệu REST API tương tác. Nó cho phép tài liệu được đồng bộ với bất kỳ thay đổi được thực hiện cho các dịch vụ REST.



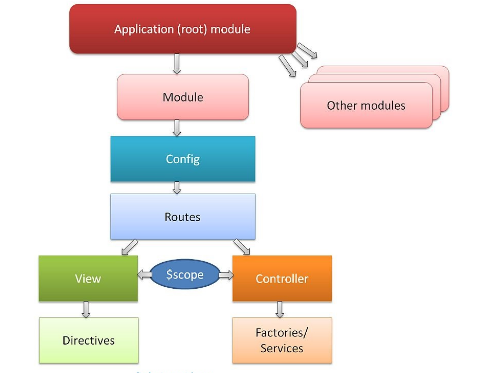
Hình 2.9. Swagger UI

### Angular 4

Phát triển các ứng dụng trong JavaScript luôn là một thách thức. Do tính chất dẻo dai và thiếu kiểm tra kiểu, việc xây dựng một ứng dụng kích thước phong phú trong JavaScript rất khó. Hơn nữa, chúng ta sử dụng JavaScript cho tất cả các loại quy trình như thao tác User Interface (UI), tương tác client server, thẩm định hay phát triển nghiệp vụ. Kết quả là, chúng ta kết thúc với mã rối, khó khăn để duy trì và thử nghiệm.

Các thư viện như jQuery làm một công việc tuyệt vời để chăm sóc các trình duyệt khác nhau và cung cấp các cấu trúc có thể dẫn đến giảm tổng thể các dòng mã. Tuy nhiên, các thư viện này thiếu bất kỳ hướng dẫn kết cấu nào có thể giúp chúng ta khi phát triển mã nguồn.

Do đó các mô hình kiến trúc như Model View Controller (MVC) và các frameworkds như AngularJS xuất hiện. AngularJS là **Javascript framework**. Nó có thể được nhúng nào HTML với thẻ **<script>.** AngularJS mở rộng các thuộc tính HTML với **Directives**, và liên kết (binds) dữ liệu ra HTML với Expressions .



Hình 2.10. Tổng quan kiến trúc AngularJS 1

Trong hệ thống Cinema Booking System này, phần client được xây dựng với phiên bản AngularJS 4 mới nhất (được xuất bản vaof 05/2017).

# Thiết kế api trong hệ thống

## Các tài nguyên trong hệ thống

Hệ thống quản lý rạp chiếu phim bao gồm các tài nguyên:

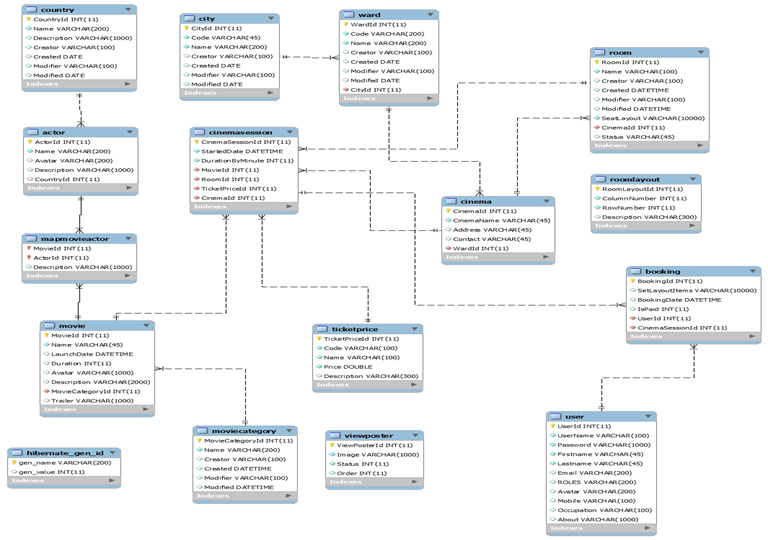
* Quản lý quốc gia sản xuất phim
* Quản lý thành phố
* Quản lý quận huyện
* Quản lý chi nhánh rạp
* Quản lý diễn viên
* Quản lý tài khoản người dùng
* Quản lý loại phim
* Quản lý phim
* Quản lý loại phòng
* Quản lý diện tích phòng
* Quản lý phòng
* Quản lý ghế trong phòng
* Quản lý giá vé
* Quản lý suất chiếu phim
* Quản lý đặt vé
* Quản lý thanh toán
* Quản lý bảng quảng cáo

## Danh sách các API

**URL Request:** http://localhost:8080/cinema-booking-system-api-0.0.1-SNAPSHOT/+ URI

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| STT | Tài nguyên | Tên API | Phương Thức | URI | Tham số | Ghi chú |
|  | **QUỐC GIA** | Lấy danh sách quốc gia (tất cả) | GET | /contries/list |  |  |
|  | Lấy danh sách quốc gia (phân trang) | GET | /contries | - page: number  - size: number  … | - page: số thứ tự của trang muốn lấy  - size: số bản ghi muốn lấy trên 1 trang. Nếu không truyền size, mặc định size = 20 |
|  | Lấy chi tiết 1 quốc gia bằng ID | GET | /contries/{contryId} | contryId: number |  |
|  | Lấy chi tiết 1 quốc gia bằng mã quốc gia | GET | /contries/has-code/{contryCode} | code: string |  |
|  | Tìm kiếm quốc gia | PUT | /contries/advance-filter | - code: string  - name: string | Truyền dưới dạng JSON:  {  "code": "",  "name": ""  } |
|  | Thêm mới quốc gia | POST | /contries/create | - code: string  - name: string | Truyền dưới dạng JSON:  {  "code": "",  "name": ""  } |
|  | Cập nhật quốc gia | PUT | /contries/update | - contryId: number  - code: string  - name: string | Truyền dưới dạng JSON:  {  "contryId": ,  "code": "",  "name": ""  } |
|  | Xóa 1 quốc gia | DELETE | /contries/delete | contryId: number |  |
|  | Xóa nhiều quốc gia | DELETE | /contries/delete-in-batch/{contryIds} | contryIds: number[] |  |
|  |  |  |  |  |  |  |
|  | **THÀNH PHỐ** | Lấy danh sách thành phố (tất cả) | GET | /cities/list |  |  |
|  | Lấy danh sách thành phố (phân trang) | GET | /cities | - page: number  - size: number  … | - page: số thứ tự của trang muốn lấy  - size: số bản ghi muốn lấy trên 1 trang. Nếu không truyền size, mặc định size = 20 |
|  | Lấy chi tiết 1 thành phố bằng ID | GET | /cities/{cityId} | cityId: number |  |
|  | Lấy chi tiết 1 thành phố bằng mã thành phố | GET | /cities/has-code/{cityCode} | code: string |  |
|  | Tìm kiếm thành phố | PUT | /cities/advance-filter | - code: string  - name: string | Truyền dưới dạng JSON:  {  "code": "",  "name": ""  } |
|  | Thêm mới thành phố | POST | /cities/create | - code: string  - name: string | Truyền dưới dạng JSON:  {  "code": "",  "name": ""  } |
|  | Cập nhật thành phố | PUT | /cities/update | - cityId: number  - code: string  - name: string | Truyền dưới dạng JSON:  {  "cityId": ,  "code": "",  "name": ""  } |
|  | Xóa 1 thành phố | DELETE | /cities/delete | cityId: number |  |
|  | Xóa nhiều thành phố | DELETE | /cities/delete-in-batch/{cityIds} | cityIds: number[] |  |
|  |  |  |  |  |  |  |
|  | **QUẬN HUYÊN** | Lấy danh sách quận huyện (tất cả) | GET | /ministries/list |  |  |
|  | Lấy danh sách quận huyện (phân trang) | GET | /ministries | - page: number  - size: number  … | - page: số thứ tự của trang muốn lấy  - size: số bản ghi muốn lấy trên 1 trang. Nếu không truyền size, mặc định size = 20 |
|  | Lấy chi tiết 1 quận huyện bằng ID | GET | /ministries/{ministryId} | ministryId: number |  |
|  | Lấy chi tiết 1 quận huyện bằng mã quận huyện | GET | /ministries/has-code/{ministryCode} | code: string |  |
|  | Tìm kiếm quận huyện | PUT | /ministries/advance-filter | - cityId: number  - code: string  - name: string | Truyền dưới dạng JSON:  {  "city":{  "cityId":  },  "code": "",  "name": ""  } |
|  | Thêm mới quận huyện | POST | /ministries/create | - code: string  - name: string | Truyền dưới dạng JSON:  {  "city":{  "cityId":  },  "code": "",  "name": ""  } |
|  | Cập nhật quận huyện | PUT | /ministries/update | - ministryId: number  - code: string  - name: string | Truyền dưới dạng JSON:  {  "city":{  "cityId":  },  "ministryId": ,  "code": "",  "name": ""  } |
|  | Xóa 1 quận huyện | DELETE | /ministries/delete | ministryId: number |  |
|  | Xóa nhiều quận huyện | DELETE | /ministries/delete-in-batch/{entityIds} | ministryIds: number[] |  |
|  |  |  |  |  |  |  |
|  | **CHÍNH NHÁNH RẠP** | Lấy danh sách chi nhánh rạp (tất cả) | GET | /cinemas/list |  |  |
|  | Lấy danh sách chi nhánh rạp (phân trang) | GET | /cinemas | - page: number  - size: number  … | - page: số thứ tự của trang muốn lấy  - size: số bản ghi muốn lấy trên 1 trang. Nếu không truyền size, mặc định size = 20 |
|  | Lấy chi tiết 1 chi nhánh rạp bằng ID | GET | /cinemas/{cinemaId} | cinemaId: number |  |
|  | Lấy chi tiết 1 chi nhánh rạp bằng tên chi nhánh rạp | GET | /cinemas/has-name/{cinemaName} | cinemaName: string |  |
|  | Tìm kiếm chi nhánh rạp | PUT | /cinemas/advance-filter | - wardId: number  - cinemaName: string  - address: string | Truyền dưới dạng JSON:  {  "ward":{  "wardId":  },  "cinemaName": "",  "address": ""  } |
|  | Thêm mới chi nhánh rạp | POST | /cinemas/create | - cinemaName: string  - address: string | Truyền dưới dạng JSON:  {  "ward":{  "wardId":  },  "cinemaName": "",  "address": ""  } |
|  | Cập nhật chi nhánh rạp | PUT | /cinemas/update | - cinemaId: number  - cinemaName: string  - address: string | Truyền dưới dạng JSON:  {  "ward":{  "wardId":  },  "cinemaId": ,  "cinemaName": "",  "address": ""  } |
|  | Xóa 1 chi nhánh rạp | DELETE | /cinemas/delete | cinemaId: number |  |
|  | Xóa nhiều chi nhánh rạp | DELETE | /cinemas/delete-in-batch/{entityIds} | cinemaIds: number[] |  |
|  |  |  |  |  |  |  |
|  | **DIỄN VIÊN** | Lấy danh sách diễn viên (tất cả) | GET | /actors/list |  |  |
|  | Lấy danh sách diễn viên (phân trang) | GET | /actors | - page: number  - size: number  … | - page: số thứ tự của trang muốn lấy  - size: số bản ghi muốn lấy trên 1 trang. Nếu không truyền size, mặc định size = 20 |
|  | Lấy chi tiết 1 diễn viên bằng ID | GET | /actors/{actorId} | actorId: number |  |
|  | Lấy chi tiết 1 diễn viên bằng tên diễn viên | GET | /actors/has-name/{name} | name: string |  |
|  | Tìm kiếm diễn viên | PUT | /actors/advance-filter | - contryId: number  - name: string  - profile: string | Truyền dưới dạng JSON:  {  "contry":{  "contryId":  },  "name": "",  "profile": ""  } |
|  | Thêm mới diễn viên | POST | /actors/create | - name: string  - profile: string | Truyền dưới dạng JSON:  {  "contry":{  "contryId":  },  "name": "",  "profile": ""  } |
|  | Cập nhật diễn viên | PUT | /actors/update | - actorId: number  - name: string  - profile: string | Truyền dưới dạng JSON:  {  "contry":{  "contryId":  },  "actorId": ,  "name": "",  "profile": ""  } |
|  | Xóa 1 diễn viên | DELETE | /actors/delete | actorId: number |  |
|  | Xóa nhiều diễn viên | DELETE | /actors/delete-in-batch/{actorIds} | actorIds: number[] |  |
|  |  |  |  |  |  |  |
|  | **NGƯỜI DÙNG** | Lấy danh sách người dùng (tất cả) | GET | /users/list |  |  |
|  | Lấy danh sách người dùng (phân trang) | GET | /users | - page: number  - size: number  … | - page: số thứ tự của trang muốn lấy  - size: số bản ghi muốn lấy trên 1 trang. Nếu không truyền size, mặc định size = 20 |
|  | Lấy chi tiết 1 người dùng bằng ID | GET | /users/{userId} | userId: number |  |
|  | Lấy chi tiết 1 người dùng bằng tên người dùng | GET | /users/has-userName/{userName} | userName: string |  |
|  | Tìm kiếm người dùng | PUT | /users/advance-filter | - userName: string  - password: string  - firstname: string  - lastname: string  - email: string  - roles: string  - avatar: string | Truyền dưới dạng JSON:  {  "userName": "",  "password": "",  "firstname": "",  "lastname": "",  "email": "",  "roles": "",  "avatar": "",  } |
|  | Thêm mới người dùng | POST | /users/create | - userName: string  - password: string  - firstname: string  - lastname: string  - email: string  - roles: string  - avatar: string | Truyền dưới dạng JSON:  {  "userName": "",  "password": "",  "firstname": "",  "lastname": "",  "email": "",  "roles": ""  } |
|  | Đăng tải ảnh đại diện người dùng | POST | /users/upload-avatar | - avatar: file |  |
|  | Cập nhật người dùng | PUT | /users/update | - userId: number  - userName: string  - password: string  - firstname: string  - lastname: string  - email: string  - roles: string  - avatar: string | Truyền dưới dạng JSON:  {  "userId": ,  "userName": "",  "password": "",  "firstname": "",  "lastname": "",  "email": "",  "roles": "",  "avatar": ""  } |
|  | Xóa 1 người dùng | DELETE | /users/delete | userId: number |  |
|  | Xóa nhiều người dùng | DELETE | /users/delete-in-batch/{userIds} | userIds: number[] |  |
|  |  |  |  |  |  |  |
|  | **LOẠI PHIM** | Lấy danh sách loại phim (tất cả) | GET | /movie-categories/list |  |  |
|  | Lấy danh sách loại phim (phân trang) | GET | /movie-categories | - page: number  - size: number  … | - page: số thứ tự của trang muốn lấy  - size: số bản ghi muốn lấy trên 1 trang. Nếu không truyền size, mặc định size = 20 |
|  | Lấy chi tiết 1 loại phim bằng ID | GET | /movie-categories/{movieCategoryId} | movieCategoryId: number |  |
|  | Lấy chi tiết 1 loại phim bằng tên loại phim | GET | /movie-categories/has-name/{ movieCategoryName} | name: string |  |
|  | Tìm kiếm loại phim | PUT | /movie-categories/advance-filter | - name: string | Truyền dưới dạng JSON:  {  "name":"" } |
|  | Thêm mới loại phim | POST | /movie-categories/create | - name: string  - description: string | Truyền dưới dạng JSON:  {  "name": "",  "description": "",  } |
|  | Cập nhật loại phim | PUT | /movie-categories/update | - movieCategoryId: number  - name: string  - description: string | Truyền dưới dạng JSON:  {  "movieCategoryId": ,  "name": "",  "status": ,  "description": "",  } |
|  | Xóa 1 loại phim | DELETE | /movie-categories/delete | movieCategoryId: number |  |
|  | Xóa nhiều loại phim | DELETE | /movie-categories/delete-in-batch/{movieCategoryIds} | movieCategoryIds: number[] |  |
|  |  |  |  |  |  |  |
|  | **PHIM** | Lấy danh sách phim (tất cả) | GET | /movies/list |  |  |
|  | Lấy danh sách phim sắp chiếu (dựa trên ‘launchDate’) | GET | /movies/upcoming |  |  |
|  | Lấy danh sách phim (phân trang) | GET | /movies | - page: number  - size: number  … | - page: số thứ tự của trang muốn lấy  - size: số bản ghi muốn lấy trên 1 trang. Nếu không truyền size, mặc định size = 20 |
|  | Lấy chi tiết 1 phim bằng ID | GET | /movies/{movieId} | movieId: number |  |
|  | Tìm kiếm phim | PUT | /movies/advance-filter | - name: string  - duration: number  - launchDate: date | Truyền dưới dạng JSON:  {  "name":"",  "duration": ,  "launchDate":"" } |
|  | Thêm mới phim | POST | /movies/create | - name: string  - duration: number  - launchDate: date  - avatar: string  - trailer: string  - description: string | Truyền dưới dạng JSON:  {  "name":"",  "duration": ,  "launchDate":"",  "avatar":"",  "trailer":"",  "description":"", } |
|  | Đăng tải ảnh đại diện phim | POST | /movies/upload-avatar | -avatar: file |  |
|  | Tải ảnh đại diện phim | GET | /movies/load-avatar/{fullFileName} | fullFileName: string |  |
|  | Cập nhật phim | PUT | /movies/update | - movieId: number  - name: string  - duration: number  - launchDate: date  - avatar: string  - trailer: string  - description: string | Truyền dưới dạng JSON:  {  "movieId":"",  "name":"",  "duration": ,  "launchDate":"",  "avatar":"",  "trailer":"",  "description":"", } |
|  | Xóa 1 phim | DELETE | /movies/delete | movieId: number |  |
|  | Xóa nhiều phim | DELETE | /movies/delete-in-batch/{movieIds} | movieIds: number[] |  |
|  |  |  |  |  |  |  |
|  | **LOẠI PHÒNG** | Lấy danh sách loại phòng (tất cả) | GET | /room-types/list |  |  |
|  | Lấy danh sách loại phòng (phân trang) | GET | /room-types | - page: number  - size: number  … | - page: số thứ tự của trang muốn lấy  - size: số bản ghi muốn lấy trên 1 trang. Nếu không truyền size, mặc định size = 20 |
|  | Lấy chi tiết 1 loại phòng bằng ID | GET | /room-types/{roomTypeId} | roomTypeId: number |  |
|  | Lấy chi tiết 1 loại phòng bằng tên loại phòng | GET | /room-types/has-name/{roomTypeName} | name: string |  |
|  | Tìm kiếm loại phòng | PUT | /room-types/advance-filter | - name: string  - status: number | Truyền dưới dạng JSON:  {  "name": "",  "status": ,  "description": "",  } |
|  | Thêm mới loại phòng | POST | /room-types/create | - name: string  - status: number  - description: string | Truyền dưới dạng JSON:  {  "name": "",  "status": ,  "description": "",  } |
|  | Cập nhật loại phòng | PUT | /room-types/update | - roomTypeId: number  - name: string  - status: number  - description: string | Truyền dưới dạng JSON:  {  " roomTypeId": ,  "name": "",  "status": ,  "description": "",  } |
|  | Xóa 1 loại phòng | DELETE | /room-types/delete | roomTypeId: number |  |
|  | Xóa nhiều loại phòng | DELETE | /room-types/delete-in-batch/{roomTypeIds} | roomTypeIds: number[] |  |
|  |  |  |  |  |  |  |
|  | **DIỆN TÍCH PHÒNG** | Lấy danh sách diện tích phòng (tất cả) | GET | /room-layouts/list |  |  |
|  | Lấy danh sách diện tích phòng (phân trang) | GET | /room-layouts | - page: number  - size: number  … | - page: số thứ tự của trang muốn lấy  - size: số bản ghi muốn lấy trên 1 trang. Nếu không truyền size, mặc định size = 20 |
|  | Lấy chi tiết 1 diện tích phòng bằng ID | GET | /room-layouts/{roomLayoutId} | roomLayoutId: number |  |
|  | Tìm kiếm diện tích phòng | PUT | /room-layouts/advance-filter | - name: string  - columnNumber: number  - rowNumber: number | Truyền dưới dạng JSON:  {  "name":"",  "columnNumber": ,  "rowNumber": ,  } |
|  | Thêm mới diện tích phòng | POST | /room-layouts/create | - name: string  - columnNumber: number  - rowNumber: number  - description: string | Truyền dưới dạng JSON:  {  "name":"",  "columnNumber": ,  "rowNumber": ,  "description": ""  } |
|  | Cập nhật diện tích phòng | PUT | /room-layouts/update | - roomLayoudId: number  - name: string  - columnNumber: number  - rowNumber: number  - description: string | Truyền dưới dạng JSON:  {  "roomLayoutId": ,  "name":"",  "columnNumber": ,  "rowNumber": ,  "description": ""  } |
|  | Xóa 1 diện tích phòng | DELETE | /room-layouts/delete | roomLayoutId: number |  |
|  | Xóa nhiều diện tích phòng | DELETE | /room-layouts/delete-in-batch/{roomLayoutIds} | roomLayoutIds: number[] |  |
|  |  |  |  |  |  |  |
|  | **PHÒNG** | Lấy danh sách phòng (tất cả) | GET | /rooms/list |  |  |
|  | Lấy danh sách phòng đang rỗi (từ thời điểm gọi) | GET | /rooms/no-calendar |  |  |
|  | Lấy danh sách phòng (phân trang) | GET | /rooms | - page: number  - size: number  … | - page: số thứ tự của trang muốn lấy  - size: số bản ghi muốn lấy trên 1 trang. Nếu không truyền size, mặc định size = 20 |
|  | Lấy chi tiết 1 phòng bằng ID | GET | /rooms/{roomId} | roomId: number |  |
|  | Tìm kiếm phòng | PUT | /rooms/advance-filter | - name: string | Truyền dưới dạng JSON:  {  "name":""  } |
|  | Thêm mới phòng | POST | /rooms/create | - cinemaId: number  - name: string  - seatLayout (Bản đồ chỗ ngồi trong phòng): JSON | Truyền dưới dạng JSON:  {  "cinema": {  "cinemaId": ,  }  "name":"",  "seatLayout": ""  } |
|  | Cập nhật phòng | PUT | /rooms/update | - roomId: number  - cinemaId: number  - name: string  - seatLayout (Bản đồ chỗ ngồi trong phòng): JSON  - status: number | Truyền dưới dạng JSON:  {  "roomId":"",  "cinema": {  "cinemaId": ,  }  "name":"",  "seatLayout": "",  "statis": ,  } |
|  | Xóa 1 phòng | DELETE | /rooms/delete | roomId: number |  |
|  | Xóa nhiều phòng | DELETE | /rooms/delete-in-batch/{roomIds} | roomIds: number[] |  |
|  |  |  |  |  |  |  |
|  | **GIÁ VÉ** | Lấy danh sách giá vé (tất cả) | GET | /ticket-prices/list |  |  |
|  | Lấy danh sách giá vé (phân trang) | GET | /ticket-prices | - page: number  - size: number  … | - page: số thứ tự của trang muốn lấy  - size: số bản ghi muốn lấy trên 1 trang. Nếu không truyền size, mặc định size = 20 |
|  | Lấy chi tiết 1 giá vé bằng ID | GET | /ticket-prices/{ticketPriceId} | ticketPriceId: number |  |
|  | Tìm kiếm giá vé | PUT | /ticket-prices/advance-filter | - code: string  - name: string  - price: number | Truyền dưới dạng JSON:  {  "code": "",  "name": "",  "price":  } |
|  | Thêm mới giá vé | POST | /ticket-prices/create | - code: string  - name: string  - price: number  - description: string | Truyền dưới dạng JSON:  {  "code": "",  "name": "",  "price": ,  "description": "",  } |
|  | Cập nhật giá vé | PUT | /ticket-prices/update | - ticketPriceId: number  - code: string  - name: string  - price: number  - description: string | Truyền dưới dạng JSON:  {  "code": "",  "name": "",  "price": ,  "description": "",  } |
|  | Xóa 1 giá vé | DELETE | /ticket-prices/delete | ticketPriceId: number |  |
|  | Xóa nhiều giá vé | DELETE | /ticket-prices/delete-in-batch/{ticketPriceIds} | ticketPriceIds: number[] |  |
|  |  |  |  |  |  |  |
|  | **SUẤT CHIẾU PHIM** | Lấy danh sách suất chiếu phim (tất cả) | GET | /cinema-sessions/list |  |  |
|  | Lấy danh sách suất chiếu phim đang chiếu | GET | /cinema-sessions/is-showing |  |  |
|  | Lấy danh sách suất chiếu phim sắp chiếu | GET | / cinema-sessions/upcoming |  |  |
|  | Lấy danh sách suất chiếu phim (phân trang) | GET | /cinema-sessions | - page: number  - size: number  … | - page: số thứ tự của trang muốn lấy  - size: số bản ghi muốn lấy trên 1 trang. Nếu không truyền size, mặc định size = 20 |
|  | Lấy chi tiết 1 suất chiếu phim bằng ID | GET | /cinema-sessions/{cinemaSessionId} | cinemaSessionId: number |  |
|  | Tìm kiếm suất chiếu phim | PUT | /cinema-sessions/advance-filter | - movieId: number  - cinemaId: number  - roomId: number  - startedDate: datetime | Truyền dưới dạng JSON:  {  "movie": {  "movieId": ,  },  "cinema": {  "cinemaId": ,  },  "room": {  "roomId": ,  },  "startedDate":  } |
|  | Thêm mới suất chiếu phim | POST | /cinema-sessions/create | - movieId: number  - cinemaId: number  - roomId: number  - ticketPriceId:  - startedDate: datetime  - durationByMinute: number | Truyền dưới dạng JSON:  {  "movie": {  "movieId": ,  },  "cinema": {  "cinemaId": ,  },  "room": {  "roomId": ,  },  "ticketprice": {  "ticketPriceId": ,  },  "startedDate": ,  "durationByMinute":  } |
|  | Cập nhật suất chiếu phim | PUT | /cinema-sessions/update | - cinemaSessionId: number  - movieId: number  - cinemaId: number  - roomId: number  - ticketPriceId:  - startedDate: datetime  - durationByMinute: number | Truyền dưới dạng JSON:  {  "cinemaSessionId": ,  "movie": {  "movieId": ,  },  "cinema": {  "cinemaId": ,  },  "room": {  "roomId": ,  },  "ticketprice": {  "ticketPriceId": ,  },  "startedDate": ,  "durationByMinute":  } |
|  | Xóa 1 suất chiếu phim | DELETE | /cinema-sessions/delete | cinemaSessionId: number |  |
|  | Xóa nhiều suất chiếu phim | DELETE | /cinema-sessions/delete-in-batch/{cinemaSessionIds} | cinemaSessionIds: number[] |  |
|  |  |  |  |  |  |  |
|  | **ĐẶT VÉ** | Lấy danh sách đặt vé (tất cả) | GET | /bookings/list |  |  |
|  | Lấy danh sách đặt vé (phân trang) | GET | /bookings | - page: number  - size: number  … | - page: số thứ tự của trang muốn lấy  - size: số bản ghi muốn lấy trên 1 trang. Nếu không truyền size, mặc định size = 20 |
|  | Lấy chi tiết 1 đặt vé bằng ID | GET | /bookings/{bookingId} | bookingId: number |  |
|  | Tìm kiếm đặt vé | PUT | /bookings/advance-filter | - userId: number  - cinemaSessionId: number  - cinemaId: number  - movieId: number  - bookingDate: datetime | Truyền dưới dạng JSON:  {  "user": {  "userId": ,  },  "cinemaSession": {  "cinemaSessionId": ,  "cinema": {  "cinemaId": ,  }  "movie": {  "movieId": ,  }  },  "bookingDate":  } |
|  | Thêm mới đặt vé | POST | /bookings/create | - userId: number  - cinemaSessionId: number  - bookingDate: datetime  - seatLayoutItems: seat[] | Truyền dưới dạng JSON:  {  "user": {  "userId": ,  },  "cinemaSession": {  "cinemaSessionId": ,  },  "bookingDate": ,  "seatLayoutItems": [  {  "roomId": ,  "seatId": ,  }  …  ]  } |
|  | Cập nhật đặt vé | PUT | /bookings/{bookingId}/update | - Tham số đường dẫn (Path Param): bookingId: number - Tham số phần thân (body):  + seatLayoutItems: string | Truyền dưới dạng JSON phần body:  "seatLayoutItems": [  {  "roomId": ,  "seatId": ,  }  …  ] |
|  | Xóa 1 đặt vé | DELETE | /bookings/delete | bookingId: number |  |
|  | Xóa nhiều đặt vé | DELETE | /bookings/delete-in-batch/{bookingIds} | bookingIds: number[] |  |
|  |  |  |  |  |  |  |
|  | **THANH TOÁN** | Lấy danh sách thanh toán (tất cả) | GET | /payments/list |  |  |
|  | Lấy danh sách thanh toán (phân trang) | GET | /payments | - page: number  - size: number  … | - page: số thứ tự của trang muốn lấy  - size: số bản ghi muốn lấy trên 1 trang. Nếu không truyền size, mặc định size = 20 |
|  | Lấy chi tiết 1 thanh toán bằng ID | GET | /payments/{paymentId} | paymentId: number |  |
|  | In vé đã thanh toán | GET | /payments/{paymentId}/tickets | paymentId: number |  |
|  | Tìm kiếm thanh toán | PUT | /payments/advance-filter | - isPaid: number | Truyền dưới dạng JSON:  {  "isPaid":  } |
|  | Thêm mới thanh toán | POST | /payments/create | - cinemaSessionId: number  - seatId: number  - isPaid: number | Truyền dưới dạng JSON:  {  "cinemaSession": {  "cinemaSessionId": ,  },  "seatId": ,  "isPaid": ,  } |
|  | Cập nhật thanh toán | PUT | /payments/{paymentId}/update/{isPaid} | - paymentId: number  - isPaid: number |  |
|  | Xóa 1 thanh toán | DELETE | /payments/delete | paymentId: number |  |
|  | Xóa nhiều thanh toán | DELETE | /payments/delete-in-batch/{paymentIds} | paymentIds: number[] |  |
|  |  |  |  |  |  |  |
|  | **BẢNG QUẢNG CÁO** | Lấy danh sách bảng quảng cáo (tất cả) | GET | /posters/list |  |  |
|  | Lấy danh sách bảng quảng cáo (phân trang) | GET | /posters | - page: number  - size: number  … | - page: số thứ tự của trang muốn lấy  - size: số bản ghi muốn lấy trên 1 trang. Nếu không truyền size, mặc định size = 20 |
|  | Lấy chi tiết 1 bảng quảng cáo bằng ID | GET | /posters/{posterId} | posterId: number |  |
|  | Đăng tải ảnh đại diện bảng quảng cáo | POST | /posters/upload-avatar | - avatar: file |  |
|  | Tải ảnh đại diện bảng quảng cáo | GET | /posters/load-avatar/{fullFileName} | fullFileName: string |  |
|  | Thêm mới bảng quảng cáo | POST | /posters/create | - title: string  - avatar: string  - order: number | Truyền dưới dạng JSON:  {  "title": "",  "avatar": "",  "order":  } |
|  | Cập nhật bảng quảng cáo | PUT | /posters/update | - posterId: number  - title: string  - avatar: string  - order: number | Truyền dưới dạng JSON:  {  "posterId": ,  "title": "",  "avatar": "",  "order":  } |
|  | Xóa 1 bảng quảng cáo | DELETE | /posters/delete | posterId: number |  |
|  | Xóa nhiều bảng quảng cáo | DELETE | /posters/delete-in-batch/{posterIds} | posterIds: number[] |  |

# Thiết kế cơ sở dữ liệu



Hình 4.1. Thiết kế cơ sở dữ liệu

# Đặc Tả Các chức năng trong Hệ thống

## UC#001: Đăng nhập

|  |  |  |  |
| --- | --- | --- | --- |
| **UC #001** | | **ĐĂNG NHẬP** | **Độ phức tạp : Medium** |
| **Mô tả** | | Usecase này cho phép người dùng đăng nhập thông qua tên đăng nhập và mật khẩu để sử dụng các chức năng của hệ thống  Trong Usecase này sẽ mô tả hai phương thức đăng nhập như sau:   * Đăng nhập đối với tài khoản thuộc trang quản trị: * Mật khẩu và Tên đăng nhập của Người quản trị Hệ thống được tạo mặc định * Mật khẩu và Tên đăng nhập của Nhân viên quản lý rạp do Người quản trị Hệ thống cung cấp. * Đăng nhập đối với tài khoản Khách hàng: * Mật khẩu và Tên đăng nhập của Khách hàng do khách hàng tự đăng kí | |
| **Tác nhân chính** | | Người quản trị Hệ thống, Nhân viên quản lý rạp, Khách hàng | |
| **Tiền điều kiện** | | Người dùng đã có tài khoản trong Hệ thống | |
| **Hậu điều kiện** | **Thành công** | Khi đăng nhập thành công, người dùng có thể sử dụng các chức năng trong Hệ thống | |
| **Lỗi** | Người dùng phải đăng nhập lại hoặc hủy thao tác | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Usecase này bắt đầu khi Khách Hàng muốn sử dụng các chức năng của Hệ thống.  *Luồng chính đối với phương thức đăng nhập của Khách Hàng*   * Người dùng truy cập vào ứng dụng, và lựa chọn Đăng nhập. * Hệ thống hiển thị ***Form Đăng nhập*** và yêu cầu người dùng nhập đầy đủ thông tin bao gồm: * Tên tài khoản (ví dụ: linh123) * Mật khẩu   (2 thông tin trên do người dùng đã đăng kí với Hệ thống)   * Người dùng nhập thông tin theo Form và nhấn Đăng nhập để gửi thông tin lên Hệ thống * Hệ thống kiểm tra tính chính xác cho thông tin mà người dùng vừa nhập, sau đó xử lý để trả về kết quả Đăng nhập cho người dùng * Nếu thông tin Tên đăng nhập và Mật khẩu khớp với dữ liệu đã lưu trữ trên Hệ thống, thì Hệ thống thông báo cho người dùng đã đăng nhập thành công, đồng thời trở về Màn hình Tìm kiếm phim cho người dùng. * Luồng sự kiện kết thúc, Người dùng đăng nhập thành công và sử dụng các chức năng trong Hệ thống   *Luồng chính đối với phương thức Đăng nhập trang quản trị.*   * Người dùng truy cập vào ứng dụng, và lựa chọn Đăng nhập. * Hệ thống hiển thị ***Form Đăng nhập*** và yêu cầu người dùng nhập đầy đủ thông tin bao gồm: * Tên tài khoản (dưới dạng tên tài khoản Gmail theo tên miền công ty ví dụ: [admin@cinemaBooking.com](mailto:admin@cinemaBooking.com), tên tài khoản do công ty cung cấp) * Mật khẩu (mật khẩu mặc định được cung cấp là: 123456, người dùng có thể trực tiếp thay đổi mật khẩu trong chức năng Quản lý tài khoản) * Người dùng nhập thông tin và nhấn Đăng nhập để gửi thông tin lên Hệ thống * Hệ thống kiểm tra tính chính xác cho thông tin mà người dùng vừa nhập, sau đó xử lý để trả về kết quả Đăng nhập cho người dùng * Nếu thông tin Tên đăng nhập và Mật khẩu khớp với dữ liệu đã lưu trữ trên Hệ thống, thì Hệ thống thông báo cho người dùng đã đăng nhập thành công, đồng thời chuyển sang Màn hình giao diện chính của trang quản lý. * Luồng sự kiện kết thúc, Người dùng đăng nhập thành công và sử dụng các chức năng trong Hệ thống | | | |
| **Luồng thay thế của sự kiện** | | | |
| *Luồng thay thế đối với phương thức đăng nhập của Khách hàng*  Tại bước 3 của luồng chính, Người dùng nhập thông tin Tên tài khoản là Tên tài tài khoản Gmail đã đăng kí với Hệ thống (ví dụ:lamlinh1707@gmail.com).  Các bước tiếp theo của luồng thay thế tương tự như luồng chính  *Luồng phụ (của luồng thay thế):*  (tương tự như 4 luồng phụ của luồng chính) | | | |
| **Luồng phụ của sự kiện/Kịch bản phát sinh** | | | |
| *Luồng phụ đối với cả 2 phương thức Đăng nhập của khách hàng và Đăng nhập trang quản trị*  *Luồng phụ 1: Người dùng nhập thiếu thông tin trong Form đăng nhập*   * Tại bước 3 của luồng chính, Người dùng nhập thiếu thông tin trong Form đăng nhập * Hệ thống thông báo “Người dùng không được bỏ trống” tại ô thông tin mà người dùng nhập thiếu * Người dùng nhập lại đầy đủ thông tin trong Form đăng nhập và nhấn Đăng nhập để gửi lên Hệ thống * Các bước tiếp theo tương tự như luồng chính   *Luồng phụ 2: Người dùng nhập sai thông tin Tên tài khoản hoặc Mật khẩu*   * Tại bước 3 của luồng chính, Người dùng nhập không chính xác Tên tài khoản đăng nhập * Hệ thống thông báo “Tài khoản chưa tồn lại hoặc không chính xác” và yêu cầu Người dùng nhập lại * Người dùng nhập lại chính xác Tên đăng nhập và Mật khẩu, sau đó nhấn Đăng nhập để gửi lên Hệ thống * Các bước tiếp theo tương tự như luồng chính   *Luồng phụ 3: Người dùng nhập sai hoặc thiếu định dạng miền của Tên tài khoản chỉ xảy ra với phương thức đăng nhập trang quản trị)*   * Tại bước 3 của luồng chính, Người dùng nhập sai hoặc thiếu định dạng tên miền của Tên tài khoản   (ví dụ [admin@cinemaBooking.com](mailto:admin@cinemaBooking.com) thì Người dùng chỉ nhập Tên tài khoản là admin hoặc nhập sai định dang miền @cinnemaBooking.com)   * Hệ thống thônng báo “Tên tài khoản không hợp lệ” và yêu cầu Người dùng nhập lại * Người dùng nhập lại chính xác Tên tài khoản và nhấn Đăng nhập để gửi lên Hệ thống * Các bước tiếp theo tương tự như luồng chính   *Luồng phụ 4: Người dùng nhập chính xác thông tin trong Form đăng nhập, nhưng không thể tiếp tục Đăng nhập*   * Tại bước 4 của luồng chính, Hệ thống không xử lý được dữ liệu Đăng nhập cho người dùng, Hệ thống thông báo cho người dùng “Lỗi Hệ thống, không thể đăng nhập”. * Hệ thống gợi ý người dùng thử Đăng nhập lại. * Người dùng lựa chọn thực hiện lại thao tác Đăng nhập hoặc thoát khỏi Hệ thống. Luồng sự kiên kết thúc | | | |
| **Giao diện minh họa** | | | |
| Hình 5.1. Giao diện trang đăng nhập    Hình 5.2. Đăng nhập thành công vào hệ thống    Hình 5.3. Thông báo nhập nhiếu thông tin    Hình 5.4. Thông báo nhập không đúng tên miền    Hình 5.5. Thông báo lỗi đăng nhập | | | |

|  |
| --- |
| **THIẾT KẾ UML** |
| **Sơ đồ Lớp phân tích (Analysis class diagram)** |
|  |
| **Sơ đồ Trình tự (Sequence diagram)** |
|  |

## UC#002: Quản lý thành phố

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#002** | | **QUẢN LÝ THÀNH PHỐ** | **Độ phức tạp:** |
| **Mô tả** | | Chức năng này cho phép người dùng thêm, xem chi tiết, sửa, xóa, tìm kiếm thành phố. | |
| **Tác nhân** | **Chính** | Quản trị hệ thống | |
| **Phụ** | Không | |
| **Tiền điều kiện** | | Sau khi người dùng đăng nhập và chọn chức năng Quản lý thành phố. | |
| **Hậu điều kiện** | **Thành công** | Cập nhật thành công cho quản lý thành phố và lưu dữ liệu vào hệ thống. | |
| **Lỗi** | Trạng thái hệ thống sẽ không thay đổi. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Chức năng này bắt đầu khi người dùng muốn thêm, xem chi tiết, sửa, xóa, tìm kiếm thành phố.  Sau khi người dùng chọn “Quản lý thành phố”, hệ thống hiển thị danh sách thành phố với các thông tin:   * Tên thành phố * Mã thành phố * Người tạo * Ngày tạo * Hành động (Xem chi tiết, Sửa, Xóa) * Checkbox   Người dùng có thể sắp xếp các trường thông tin Tên thành phố, Mã thành phố, Người tạo, Ngày tạo:   * Tên thành phố, Người tạo được sắp xếp theo bảng chữ cái alphabet * Mã thành phố được sắp xếp theo mức độ ưu tiên: số, ‘-’, alphabet * Ngày tạo được sắp xếp theo ngày tạo gần nhất, xa nhất.   Bên trong chức năng này có các tác vụ: thêm, xem chi tiết, sửa, xóa, tìm kiếm   * Khi người dùng chọn tác vụ thêm thì **luồng 1 - Thêm thành phố** được thực hiện * Khi người dùng chọn tác vụ xem chi tiết thì **luồng 2 – Xem chi tiết thành phố** được thực hiện. * Khi người dùng chọn tác vụ sửa thì **luồng 3 - Sửa thành phố** được thực hiện * Khi người dùng chọn tác vụ xóa thì **luồng 4 - Xóa thành phố** được thực hiện * Khi người dùng chọn tác vụ tìm kiếm thì **luồng 5 - Tìm kiếm thành phố** được thực hiện   **Luồng 1 - Thêm thành phố**  Khi người dùng chọn tác vụ Thêm thành phố, hệ thống hiển thị form thêm yêu cầu người dùng nhập/chọn các trường thông tin:   * Mã thành phố\* * Tên thành phố\* * Button Lưu, Quay lại   Những trường thông tin có dấu \* là bắt buộc  Khi người dùng nhập/chọn đầy đủ các thông tin một cách hợp lệ thì chọn Lưu để hệ thống cập nhật.  Khi người dùng không muốn tiếp tục việc thêm thành phố thì chọn Quay lại để quay lại danh sách thành phố.  Hệ thống hiển thị lại thông tin về thành phố đó ở danh sách thành phố.  **Luồng 2 - Xem chi tiết thành phố**  Người dùng chọn 1 dòng dữ liệu thành phố bất kỳ trong danh sách thành phố,  Có 2 cách chọn:   * Chọn vào button Xem chi tiết ở dòng dữ liệu thành phố muốn xem * Chọn vào tên thành phố muốn xem   Hệ thống sẽ hiển thị thông tin của thành phố đó: mã thành phố, tên thành phố và button Quay lại để quay lại danh sách thành phố.  **Luồng 3 - Sửa thành phố**  Khi người dùng muốn sửa thành phố: Chọn vào button Sửa của dòng dữ liệu thành phố muốn sửa  Hệ thống sẽ hiển thị form sửa với các thông tin như luồng 1 – Thêm thành phố  Sau khi sửa xong, người dùng chọn “cập nhật” để hệ thống cập nhật lại thông tin.  Nếu người dùng không muốn tiếp tục thực hiện thao tác sửa thì chọn button Quay lại để quay lại danh sách thành phố.  Hệ thống hiển thị thông tin thành phố đã được cập nhật ở danh sách thành phố.  **Luồng 4 - Xóa thành phố**  Có 2 cách thức xóa:   * Cách 1: Xóa 1 thành phố   Chọn button xóa ở 1 thành phố bất kỳ trong danh sách thành phố  Hoặc chọn check box ở 1 thành phố muốn xóa, hệ thống sẽ hiểu thị button Xóa, chọn button Xóa   * Cách 2: Xóa nhiều thành phố:   Chọn check box các thành phố muốn xóa hoặc chọn check box đầu tiên để xóa hết các thành phố  Hệ thống sẽ hiển thị button Xóa, chọn button Xóa để xóa các thành phố đã chọn  Hệ thống hiển thị popup yêu cầu người dùng xác nhận: chọn “Có” để xóa, còn “Không” là không xóa.   * Nếu người dùng chọn “Có” thì hệ thống sẽ cập nhật lại thông tin, hiển thị thông báo “Xóa thành công” và hiển thị danh sách thành phố. * Nếu người dùng chọn “Không” thì hệ thống quay lại màn hình hiển thị danh sách thành phố.   Nếu thành phố đã có các quận huyện kèm theo thì không thể xóa. Hệ thống sẽ báo lỗi “Không thể xóa”.  Nếu thành phố chưa có quận huyện nào thì có thể xóa  **Luồng 5 - Tìm kiếm thành phố**  Khi người dùng muốn tìm kiếm thành phố, người dùng cần nhập dữ liệu tìm kiếm tại tác vụ tìm kiếm.  Thông tin tìm kiếm theo: Mã thành phố, tên thành phố.  Người dùng có thể nhập các ký tự trong tên thành phố, mã thành phố (chữ hoa, chữ thường, dấu cách đều được chấp nhận)  Sau khi người dùng chọn Tìm kiếm (hoặc nhấn Enter), hê thống sẽ tìm kiếm và hiển thị danh sách thành phố tìm được. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| **Nhập thiếu thông tin**  Khi người dùng không nhập các trường thông tin bắt buộc (\*) thì hệ thống sẽ báo lỗi “Thông tin bắt buộc” ở các trường đó  **Nhập trùng thông tin**  Khi người dùng nhập thông tin trường Mã thành phố, Tên thành phố trùng với thông tin đã có trên hệ thống thì hệ thống sẽ báo lỗi  **Nhập dấu cách**  Khi người dùng nhập ký tự space ở đầu, lớn hơn 2 space ở giữa, cuối, hoặc ký tự tab ở các trường thông tin Mã thành phố, Tên thành phố thì hệ thống sẽ báo lỗi  **Nhập ký tự không phù hợp**  Trường thông tin Mã thành phố chỉ chấp nhận ký tự chữ (thường, hoa), ký tự số, dấu ‘-’, các ký tự khác sẽ không được chấp nhận. Khi người dùng nhập các ký tự không phù hợp thì hệ thống sẽ báo lỗi.  Trường thông tin Tên thành phố chỉ chấp nhận ký tự chữ (hoa, thường), ký tự số, các ký tự khách sẽ không được chấp nhận. Khi người dùng nhập các ký tự không phù hợp thì hệ thống sẽ báo lỗi. | | | |
| **Giao diện minh họa** | | | |
| Hình 5.6. Giao diện quản lý thành phố    Hình 5.7. Giao diện xem chi tiết một thành phố    Hình 5.8. Giao diện thêm mới một thành phố    Hình 5.9. Giao diện chỉnh sửa thông tin thành phố    Hình 5.10. Giao diện xóa 1 thành phố    Hình 5.11. Giao diện xóa nhiều thành phố    Hình 5.12. Giao diện tìm kiếm thành phố | | | |
| **THIẾT KẾ UML** | | | |
| **Sơ đồ Lớp phân tích (Analysis class diagram)** | | | |
|  | | | |
| **Sơ đồ Trình tự (Sequence diagram)** | | | |
|  | | | |

## UC#003: Quản lý quận huyện

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#003** | | **QUẢN LÝ QUẬN HUYỆN** | **Độ phức tạp:** |
| **Mô tả** | | Chức năng này cho phép người dùng thêm, xem chi tiết, sửa, xóa, tìm kiếm quận huyện. | |
| **Tác nhân** | **Chính** | Quản trị hệ thống | |
| **Phụ** | Không | |
| **Tiền điều kiện** | | Sau khi người dùng đăng nhập và chọn chức năng Quản lý quận huyện. | |
| **Hậu điều kiện** | **Thành công** | Cập nhật thành công cho quản lý quận huyện và lưu dữ liệu vào hệ thống. | |
| **Lỗi** | Trạng thái hệ thống sẽ không thay đổi. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Chức năng này bắt đầu khi người dùng muốn thêm, xem chi tiết, sửa, xóa, tìm kiếm quận huyện.  Sau khi người dùng chọn “Quả lý quận huyện”, hệ thống hiển thị danh sách quận huyện với các thông tin:   * Tên quận huyện * Mã quận huyện * Tên thành phố * Người tạo * Ngày tạo * Hành động (Xem chi tiết, Sửa, Xóa)   Bên trong chức năng này có các tác vụ: thêm, xem chi tiết, sửa, xóa, tìm kiếm   * Khi người dùng chọn tác vụ thêm thì **luồng 1 - Thêm quận huyện** được thực hiện * Khi người dùng chọn tác vụ xem chi tiết thì **luồng 2 – Xem chi tiết quận huyện** được thực hiện * Khi người dùng chọn tác vụ sửa thì **luồng 3 - Sửa quận huyện** được thực hiện * Khi người dùng chọn tác vụ xóa thì **luồng 4 - Xóa quận huyện** được thực hiện * Khi người dùng chọn tác vụ tìm kiếm thì **luồng 5 - Tìm kiếm quận huyện** được thực hiện   **Luồng 1 - Thêm quận huyện**  Khi người dùng chọn tác vụ Thêm quận huyện, hệ thống hiển thị form thêm yêu cầu người dùng nhập/chọn các trường thông tin:   * Tên quận huyện\* * Mã quận huyện \* * Tên thành phố \* * Button Lưu, Quay lại   Những trường thông tin có dấu \* là bắt buộc  Khi người dùng nhập/chọn đầy đủ các thông tin một cách hợp lệ thì chọn Lưu để hệ thống cập nhật. Hệ thống hiể thị thông báo “Thêm mới thành công”  Khi người dùng không muốn tiếp túc việc thêm quận huyện thì chọn Quay lại để quay lại danh sách quận huyện.  Hệ thống hiển thị lại thông tin về quận huyện đó.  **Luồng 2 - Xem chi tiết quận huyện**  Người dùng chọn 1 dòng dữ liệu quận huyện bất kỳ trong danh sách quận huyện  Có 2 cách chọn:   * Chọn vào button Xem chi tiết ở dòng dữ liệu quận huyện muốn xem * Chọn vào tên quận huyện muốn xem   Hệ thống sẽ hiển thị thông tin của quận huyện đó: mã quận huyện, tên quận huyện, tên thành phố và button Quay lại để quay lại danh sách quận huyện.  **Luồng 3 - Sửa quận huyện**  Khi người dùng muốn sửa quận huyện: Chọn vào button Sửa của dòng dữ liệu quận huyện muốn sửa  Hệ thống sẽ hiển thị form sửa với các thông tin như luồng 1 – Thêm quận huyện  Sau khi sửa xong, người dùng chọn “cập nhật” để hệ thống cập nhật lại thông tin. Hệ thống hiển thị thông báo “Sửa thành công”. Nếu người dùng không muốn tiếp tục thực hiện thao tác sửa thì chọn button Quay lại để quay lại danh sách quận huyện.  Hệ thống hiển thị thông tin quận huyện đã được cập nhật ở danh sách thành phố.  **Luồng 4 - Xóa quận huyện**  Có 2 cách thức xóa:   * Cách 1: Xóa 1 quận huyện   Chọn button xóa ở 1 quận huyện bất kỳ trong danh sách quận huyện  Hoặc chọn check box ở 1 quận huyện muốn xóa, hệ thống sẽ hiểu thị button Xóa, chọn button Xóa   * Cách 2: Xóa nhiều quận huyện:   Chọn check box các quận huyện muốn xóa hoặc chọn check box đầu tiên để xóa hết các quận huyện  Hệ thống sẽ hiển thị button Xóa, chọn button Xóa để xóa các quận huyện đã chọn  Hệ thống hiển thị popup yêu cầu người dùng xác nhận: chọn “Có” để xóa, còn “Không” là không xóa.   * Nếu người dùng chọn “Có” thì hệ thống sẽ cập nhật lại thông tin, hiển thị thông báo “Xóa thành công” và hiển thị danh sách quận huyện. * Nếu người dùng chọn “Không” thì hệ thống quay lại màn hình hiển thị danh sách quận huyện.   Nếu quận huyện đã có các thành phố kèm theo thì không thể xóa.  Nếu quận huyện chưa có quận huyện nào thì có thể xóa  **Luồng 4 - Tìm kiếm quận huyện**  Khi người dùng muốn tìm kiếm quận huyện, người dùng cần nhập dữ liệu tìm kiếm tại tác vụ tìm kiếm.  Thông tin tìm kiếm theo: Tên quận huyện, mã quận huyện, tên thành phố  Người dùng có thể nhập các ký tự trong tên quận huyện, mã quận huyện, tên thành phố (chữ hoa, chữ thường đều được chấp nhận)  Người dùng chọn Tìm kiếm.  Hệ thống sẽ tìm kiếm và hiển thị danh sách quận huyện tìm được. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| **Nhập thiếu thông tin**  Khi người dùng không nhập các trường thông tin bắt buộc (\*) thì hệ thống sẽ báo lỗi “Thông tin bắt buộc” ở các trường đó  **Nhập trùng thông tin**  Khi người dùng nhập thông tin trường Mã quận huyện, Tên quận huyện trùng với thông tin đã có trên hệ thống thì hệ thống sẽ báo lỗi  **Nhập dấu cách**  Khi người dùng nhập ký tự space ở đầu, lớn hơn 2 space ở giữa, cuối, hoặc ký tự tab ở các trường thông tin Mã quận huyện, Tên quận huyện thì hệ thống sẽ báo lỗi  **Nhập ký tự không phù hợp**  Trường thông tin Mã quận huyện chỉ chấp nhận ký tự chữ (thường, hoa), ký tự số, dấu ‘-‘, các ký tự còn lại sẽ không được chấp nhận. Khi người dùng nhập các ký tự không phù hợp thì hệ thống sẽ báo lỗi.  Trường thông tin Tên thành phố, tên quận huyện chỉ chấp nhận ký tự chữ (hoa, thường), ký tự số, các ký tự khách sẽ không được chấp nhận. Khi người dùng nhập các ký tự không phù hợp thì hệ thống sẽ báo lỗi. | | | |
| **Giao diện minh họa** | | | |
| Hình 5.13. Giao diện quản lý quận huyện    Hình 5.14. Giao diện xem chi tiết một quận huyện    Hình 5.15. Giao diện thêm mới quận huyện    Hình 5.16. Giao diện chỉnh sửa quận huyện    Hình 5.17. Giao diện xóa một quận huyện    Hình 5.18. Giao diện xóa nhiều quận huyện    Hình 5.19. Giao diện tìm kiếm quận huyện | | | |
| **THIẾT KẾ UML** | | | |
| **Sơ đồ Lớp phân tích (Analysis class diagram)** | | | |
|  | | | |

|  |
| --- |
| **Sơ đồ Trình tự (Sequence diagram)** |
|  |

## UC#004: Quản lý chi nhánh rạp chiếu phim

|  |  |  |  |
| --- | --- | --- | --- |
| **UC #004** | | **QUẢN LÝ CHI NHÁNH RẠP CHIẾU PHIM** | **Độ phức tạp: High** |
| **Mô tả** | | Use case này cho phép người quản trị quản lý các chi nhánh rạp chiếu phim trong hệ thống. Bao gồm việc thêm, thay đổi, xoá và tìm kiếm thông tin chi nhánh trong hệ thống. | |
| **Tác nhân** | **Chính** | Người quản trị hệ thống | |
| **Phụ** |  | |
| **Tiền điều kiện** | | Người quản trị phải đăng nhập vào hệ thống trước khi use case bắt đầu | |
| **Hậu điều kiện** | **Thành công** | Nếu use case thành công, thông tin về chi nhánh được thêm, cập nhật, xoá khỏi hệ thống. | |
| **Lỗi** | Trạng thái của hệ thống không thay đổi. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Use case này bắt đầu khi người quản trị muốn thêm, sửa và/hoặc xoá thông tin chi nhánh trong hệ thống.   * Hệ thống hiển thị danh sách các chi nhánh có trong cơ sở dữ liệu. * Người quản trị xác định chức năng mà anh/cô ấy muốn sử dụng (thêm hoặc cập nhật thông tin một chi nhánh) * Ngay khi người quản trị hệ thống xác định xong chức năng muốn thực hiện, một trong số các luồng phụ sẽ được thực thi. Nếu người quản trị chọn “thêm chi nhánh”, luồng phụ *Thêm chi nhánh* sẽ được thực thi. Nếu người quản trị chọn “cập nhật chi nhánh”, luồng phụ *Cập nhật thông tin chi nhánh* sẽ được thực thi. Nếu người quản trị hệ thống chọn “Tìm kiếm”, luồng phụ *Tìm kiếm chi nhánh* sẽ được thực thi, luồng phụ *Xoá chi nhánh* sẽ được thực thi.  1. ***Thêm chi nhánh:***  * Hệ thống yêu cầu người quản trị nhập thông tin chi nhánh cần thêm, bao gồm: * Tên chi nhánh (bắt buộc) * Địa chỉ * Số điện thoại * Mô tả * Trạng thái: kích hoạt hoặc không * Ngay khi người quản trị hệ thống cung cấp các thông tin được yêu cầu, hệ thống tạo ra và gán cho chi nhánh mới một ID duy nhất và tạo phương thức kích hoạt mặc định “active” (nếu người dùng không chọn trạng thái). Chi nhánh mới sẽ được thêm vào hệ thống. * Hệ thống cung cấp cho người quản trị ID mới của chi nhánh.  1. ***Cập nhật chi nhánh:***  * Hệ thống thực hiện truy vấn trong cơ sở dữ liệu và hiển thị thông tin chi tiết chi nhánh vừa chọn. * Người quản trị thực hiện thay đổi thông tin chi nhánh. Bao gồm các thông tin đã được ghi rõ trong luồng phụ *Thêm chi nhánh.* * Ngay khi người quản trị cập nhật thông tin cần thiết, hệ thống lưu vào bản ghi các thông tin được cập nhật.  1. ***Tìm kiếm chi nhánh:***  * Hệ thống hiển thị và yêu cầu người dùng nhập vào thông tin muốn tìm kiếm theo các trường:   + Tên chi nhánh   + Số điện thoại   + Địa chỉ * Nhân viên quản lý nhập vào thông tin và chọn “Search”. * Hệ thống kiểm tra và hiển thị dữ liệu tương ứng.  1. ***Xoá chi nhánh:***  * Hệ thống đưa ra cho người quản trị yêu cầu xác nhận xóa. * Người quản trị xác nhận yêu cầu xóa. * Hệ thống xác nhận yêu cầu xóa của người quản trị và lập tức loại bỏ chi nhánh được chọn khỏi hệ thống. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| **<<Luồng phát sinh 1>> Thông tin thêm mới hoặc cập nhật không hợp lệ** | | | |
| Nếu trong luồng phụ *Cập nhật chi nhánh* hoặc luồng phụ *Thêm chi nhánh*, người quản trị nhập thiếu hoặc nhập vào dữ liệu không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi. Sau đó người quản trị có thể nhập lại hoặc huỷ thao tác. | | | |
| **<<Luồng phát sinh 2>> Huỷ bỏ lệnh xoá** | | | |
| Nếu trong luồng phụ *Xoá chi nhánh*, người quản trị hệ thống quyết định không xoá, lệnh xoá sẽ bị huỷ và luồng chính được khởi động lại trạng thái ban đầu. | | | |
| **Giao diện minh họa** | | | |
| Hình 5.20. Giao diện quản lý chi nhánh rạp chiếu phim    Hình 5.21. Giao diện xem chi tiết một chi nhánh rạp chiếu phim    Hình 5.22. Giao diện thêm mới chi nhánh rạp chiếu phim    Hình 5.23. Giao diện chỉnh sửa chi nhánh rạp chiếu phim    Hình 5.24. Giao diện xóa một chi nhánh rạp chiếu phim    Hình 5.25. Giao diện xóa nhiều chi nhánh rạp chiếu phim    Hình 5.26. Giao diện tìm kiếm chi nhánh rạp chiếu phim | | | |
| **THIẾT KẾ UML** | | | |
| **Sơ đồ Lớp phân tích (Analysis class diagram)** | | | |
| Site | | | |

|  |
| --- |
| **Sơ đồ Trình tự (Sequence diagram)** |
|  |

## UC#005: Quản lý tài khoản

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#005** | | **Quản lý tài khoản** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Chức năng này cho phép Nhân viên quản lý thực hiện một số thao tác như: thêm mới tài khoản, cập nhật thông tin tài khoản, xem chi tiết tài khoản, tìm kiếm tài khoản, xóa tài khoản. Khách hàng cũng có thể thực hiện đổi mật khẩu, và cập nhật thông tin tài khoản của mình. | |
| **Tác nhân** | **Chính** | Nhân viên quản lý | |
| **Phụ** | Khách hàng | |
| **Tiền điều kiện** | | Đăng nhập thành công vào hệ thống. | |
| **Hậu điều kiện** | **Thành công** | Dữ liệu được cập nhật vào hệ thống (nếu có) | |
| **Lỗi** | Trạng thái hệ thống không thay đổi. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Chức năng này bắt đầu khi Nhân viên quản lý chọn chức năng *“Quản lý tài khoản”*.  Với mỗi tác nhân, hệ thống sẽ hiển thị giao diện tương ứng với loại tài khoản mà tác nhân đăng nhập.  ***Đối với Nhân viên quản lý:***  Hệ thống hiển thị danh sách thông tin tài khoản có trong hệ thống bao gồm:   * Tên tài khoản * Địa chỉ mail * Số điện thoại * Loại tài khoản   *(Với mỗi kết quả hiển thị, có các chức năng tùy chọn cho Nhân viên quản lý: xem thông tin chi tiết tài khoản, cập nhật thông tin tài khoản).*  Nếu Nhân viên quản lý chọn chức năng nào thì chức năng đó sẽ được thực hiện: thêm mới tài khoản, cập nhật tài khoản, xem chi tiết tài khoản, tìm kiếm tài khoản, xóa tài khoản.   1. Thêm mới tài khoản  * Hệ thống hiển thị trang thêm tài khoản và yêu cầu nhập đầy đủ các thông tin:   Họ tên  Giới tính  Địa chỉ mail  Tên đăng nhập  Mật khẩu  Nhập lại mật khẩu  Địa chỉ  Số điện thoại  Loại tài khoản (quản lý, khách hàng)   * Nhân viên quản lý nhập vào thông tin và chọn nút “Thêm mới”. * Hệ thống kiểm tra và cập nhật dữ liệu vào DB.  1. Cập nhật tài khoản  * Hệ thống hiển thị và yêu cầu Nhân viên quản lý vào thông tin để cập nhật bao gồm:   Họ tên  Giới tính  Địa chỉ mail  Tên đăng nhập  Mật khẩu  Nhập lại mật khẩu  Địa chỉ  Số điện thoại  Loại tài khoản (quản lý, khách hàng)   * Nhân viên quản lý nhập vào thông tin và chọn nút “Cập nhật”. * Hệ thống kiểm tra và cập nhật dữ liệu vào DB.  1. Xem chi tiết tài khoản  * Hệ thống hiển thị thông tin chi tiết của tài khoản mà Nhân viên quản lý chọn bao gồm:   Họ tên  Giới tính  Địa chỉ mail  Tên đăng nhập  Mật khẩu  Nhập lại mật khẩu  Địa chỉ  Số điện thoại  Loại tài khoản (quản lý, khách hàng)   * Nhân viên quản lý có thể chọn nút “Cập nhật tài khoản” để cập nhật thông tin tài khoản nếu muốn. * Nếu Nhân viên quản lý chọn nút “Cập nhật tài khoản” thì hệ thống sẽ hiển thị chức năng cập nhật tài khoản như trên.  1. Tìm kiếm tài khoản  * Hệ thống hiển thị và yêu cầu Nhân viên quản lý nhập vào thông tin muốn tìm kiếm theo các trường:   Tên tài khoản  Email  Số điện thoại  Loại tài khoản   * Nhân viên quản lý nhập vào thông tin và chọn nút “Tìm kiếm”. * Hệ thống kiểm tra và hiển thị dữ liệu tương ứng.  1. Xóa tài khoản  * Hệ thống hiển thị danh sách tài khoản. * Nhân viên quản lý chọn muốn xóa. * Hệ thống sẽ hiển thị xác nhận xóa. * Nhân viên quản lý xác nhận xóa. * Hệ thống xóa tài khoản đã chọn và cập nhật lại danh sách.  1. Cập nhật thông tin khách hàng   Khách hàng tạo tài khoản trên hệ thống có thể vào thay đổi và cập nhật thông tin và mật khẩu tài khoản của mình.   * 1. Nếu Khách hàng chọn “Cập nhật thông tin”. * Hệ thống sẽ hiển thị trang thông tin cập nhật bao gồm:   Họ tên  Giới tính  Địa chỉ mail  Địa chỉ  Số điện thoại   * Khách hàng điền các thông tin muốn cập nhật và chọn nút “Cập nhật”. * Hệ thống kiểm tra lại thông tin và cập nhật dữ liệu vào DB.   1. . Nếu Khách hàng chọn nút “Đổi mật khẩu” * Hệ thống hiển thị trang đổi mật khẩu với các thông tin:   Mật khẩu hiên tại  Mật khẩu mới  Nhập lại mật khẩu mới   * Khách hàng nhập đầy đủ thông tin và chọn “Đổi mật khẩu”. * Hệ thống kiểm tra lại thông tin và cập nhật lại DB. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| **Thêm mới tài khoản/Cập nhật tài khoản – Tên đăng nhập đã tồn tại**  Nếu tên đăng nhập Nhân viên quản lý nhập vào đã tồn tại trong DB thì hệ thống báo lỗi. Nhân viên quản lý có thể nhập lại hoặc hủy hành động này.  **Thêm mới tài khoản – Nhâp thiếu thông tin**  Nếu Nhân viên quản lý nhập thiếu thông tin hệ thống yêu cầu thì hệ thống sẽ báo lỗi. Nhân viên quản lý có thể bổ sung thông tin hoặc hủy hành động này.  **Tìm kiếm tài khoản – Không tìm thấy dữ liệu**  Nếu hệ thống không tìm thấy dữ liệu Nhân viên quản lý vào thì hệ thống sẽ báo lỗi. Nhân viên quản lý có thể nhập lại hoặc hủy hành động này.  **Xóa tài khoản – Tồn tại ràng buộc giữa các bảng dữ liệu**  Nếu tài khoản Nhân viên quản lý muốn xóa có tồn tại ràng buộc giữa các bảng dữ liệu thì việc xóa thất bại. | | | |
| **Giao diện minh họa** | | | |
| Hình 5.27. Giao diện quản lý thông tin cá nhận của user đăng nhập (1)    Hình 5.28. Tab thay đổi ảnh đại diện (2)    Hình 5.29. Tab thay đổi mật khẩu đăng nhập | | | |
| **THIẾT KẾ UML** | | | |
| **Sơ đồ Lớp phân tích (Analysis class diagram)** | | | |
|  | | | |
| **Sơ đồ Trình tự (Sequence diagram)** | | | |
|  | | | |

## UC#006: Quản lý thể loại phim

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#006** | | **Quản lý thể loại phim** | **Độ phức tạp: trung bình** |
| **Mô tả** | | Chức năng này cho phép Nhân viên quản lý thực hiện một số chức năng: thêm mới thể loại phim, cập nhật thông tin thể loại phim, xem chi tiết thể loại phim, xóa thể loại phim, tìm kiếm thể loại phim. | |
| **Tác nhân** | **Chính** | Nhân viên quản lý | |
| **Phụ** | Không | |
| **Tiền điều kiện** | | Đăng nhập thành công vào hệ thống. | |
| **Hậu điều kiện** | **Thành công** | Dữ liệu trong hệ thống được cập nhật (nếu có). | |
| **Lỗi** | Dữ liệu trong hệ thống không thay đổi. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Chức năng này bắt đầu khi Nhân viên quản lý muốn quản lý các thể loại phim. Nhân viên quản lý chọn chức năng *“Quản lý thể loại phim”*, hệ thống sẽ hiển thị danh sách thể loại trong hệ thống bao gồm:   * Tên thể loại phim * Người tạo * Ngày tạo   Nếu Nhân viên quản lý chọn một trong các chức năng: thêm mới thể loại phim, cập nhật thể loại phim, xem chi tiết thể loại phim, xóa thể loại phim, tìm kiếm thể loại phim. Chức năng đó sẽ được thực hiện   1. Thêm mới thể loại phim  * hệ thống hiển thị và yêu cầu Nhân viên quản lý nhập vào các thông tin bao gồm:   Tên thể loại phim (bắt buộc)   * Nhân viên quản lý nhập vào thông tin và chọn nút “Thêm mới”. * Hệ thống kiểm tra và cập nhật dữ liệu vào DB.  1. Cập nhật thể loại phim  * Hệ thống hiển thị thông tin chi tiết của thể loại phim vừa chọn và yêu cầu Nhân viên quản lý nhập vào các thông tin muốn cập nhật:   Tên thể loại phim   * Nhân viên quản lý nhập thông tin và chọn nút “Cập nhật”. * Hệ thống kiểm tra và cập nhật dữ liệu vào DB.  1. Xem chi tiết thể loại phim  * Hệ thống hiển thị thông tin chi tiết dự án mà Nhân viên quản lý chọn:   Tên thể loại phim   1. Xóa thể loại phim  * Nhân viên quản lý chọn thể loại phim muốn xóa và chọn nút “xóa”. * Hiển thị thông báo xác nhận xóa. * Nhân viên quản lý bấm xác nhận. * Hệ thống xóa thể loại phim đã chọn và cập nhật lại DB.  1. Tìm kiếm thể loại phim  * Hệ thống yêu cầu Nhân viên quản lý nhập vào thông tin cần tìm kiếm. * Nhân viên quản lý nhập vào thông tin và chọn nút “Tìm kiếm”. * (Nhân viên quản lý nhập thông tin tìm kiếm Tên thể loại phim) * Hệ thống kiểm tra và hiển thị dữ liệu tương ứng. * (Nhân viên quản lý có thể chọn chức năng “Cập nhật thể loại phim” hoặc “Xem chi tiết thể loại phim” với mỗi kết quả tìm kiếm.) * Nếu Nhân viên quản lý chọn chức năng “Cập nhật thể loại phim” hệ thống hiển thị chức năng cập nhật thể loại phim như mô tả mục 2 ở trên. * Nếu Nhân viên quản lý chọn chức năng “Xem chi tiết thể loại” hệ thống hiển thị chức năng xem chi tiết thể loại phim như mô tả mục 3 ở trên. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| **Thêm mới thể loại phim/Cập nhật thể loại phim – Nhập thiếu thông tin**  Nếu Nhân viên quản lý nhập thiếu thông tin Tên thể loại phim thì hệ thống sẽ báo lỗi. Nhân viên quản lý có thể nhập lại hoặc hủy hành động này.  **Thêm mới thể loại phim/Cập nhật thể loại phim – Tên thể loại phim đã tồn tại**  Nếu Tên thể loại phim nhập vào đã tồn tại thì hệ thống sẽ báo lỗi. Nhân viên quản lý có thể nhập lại hoặc hủy hành động này.  **Tìm kiếm thể loại phim – Không tìm thấy dữ liệu**  Nếu hệ thống không tìm thấy thông tin mà Nhân viên quản lý nhập vào thì hệ thống sẽ hiển thị thông báo “Không tìm thấy dữ liệu”. Nhân viên quản lý có thể nhập để tìm kiếm lại hoặc hủy hành động này.  **Xóa thể loại phim - Tồn tại ràng buộc giữa các bảng dữ liệu**  Nếu thể loại phim Nhân viên quản lý muốn xóa tồn tại ràng buộc giữa các bảng dữ liệu khác thì việc xóa thất bại. | | | |
| **Giao diện minh họa** | | | |
| Hình 5.30. Giao diện quản lý loại phim    Hình 5.31. Giao diện xem chi tiết một loại phim    Hình 5.32. Giao diện thêm mới loại phim    Hình 5.33. Giao diện chỉnh sửa loại phim    Hình 5.34. Giao diện xóa một loại phim    Hình 5.35. Giao diện xóa nhiều loại phim    Hình 5.36. Giao diện tìm kiếm loại phim | | | |
| **THIẾT KẾ UML** | | | |
| **Sơ đồ Lớp phân tích (Analysis class diagram)** | | | |
|  | | | |
| **Sơ đồ Trình tự (Sequence diagram)** | | | |
|  | | | |

## UC#007: Quản lý quốc gia sản xuất phim

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#007** | | **Quản lý quốc gia sản xuất phim** | **Độ phức tạp: Trung bình** |
| **Mô tả** | | Chức năng này cho phép Quản trị hệ thống thực hiện một số chức năng: thêm, sửa, xóa, tìm kiếm các quốc gia sản xuất phim. | |
| **Tác nhân** | **Chính** | Quản trị hệ thống. | |
| **Phụ** | Không | |
| **Tiền điều kiện** | | Đăng nhập thành công vào hệ thống. | |
| **Hậu điều kiện** | **Thành công** | Dữ liệu trong hệ thống được cập nhật (nếu có). | |
| **Lỗi** | Dữ liệu trong hệ thống không thay đổi. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Chức năng này bắt đầu khi Quản trị hệ thống muốn quản lý các quốc gia sản xuất phim. Quản trị hệ thống chọn chức năng *“Quản lý quốc gia sản xuất phim”*, hệ thống sẽ hiển thị danh sách thể loại trong hệ thống bao gồm:   * Tên quốc gia sản xuất phim * Người tạo quốc gia * Ngày tạo * Mô tả   Nếu Quản trị hệ thống chọn một trong các chức năng: thêm mới quốc gia sản xuất phim, cập nhật quốc gia sản xuất phim, xóa quốc gia sản xuất phim, tìm kiếm quốc gia sản xuất phim. Chức năng đó sẽ được thực hiện   1. Thêm mới quốc gia sản xuất phim  * Quản trị viên nhấn vào nút “Thêm mới”. * Hệ thống hiển thị và yêu cầu Quản trị hệ thống nhập vào các thông tin bao gồm:   Tên quốc gia sản xuất phim  Mô tả   * Quản trị hệ thống nhập vào thông tin và chọn nút “Lưu lại”. * Hệ thống kiểm tra và cập nhật dữ liệu vào DB.  1. Cập nhật quốc gia sản xuất phim  * Người dùng nhấn vào nút có biểu tượng “Cập nhật”. * Hệ thống hiển thị thông tin chi tiết của dự án vừa chọn và yêu cầu Quản trị hệ thống nhập vào các thông tin muốn cập nhật:   Tên quốc gia sản xuất phim  Mô tả   * Quản trị hệ thống nhập thông tin và chọn nút “Lưu lại”. * Hệ thống kiểm tra và cập nhật dữ liệu vào DB.  1. Xóa quốc gia sản xuất phim  * Quản trị hệ thống chọn quốc gia sản xuất phim muốn xóa và chọn nút có biểu tượng “Xóa”. * Hiển thị thông báo xác nhận xóa. * Quản trị hệ thống bấm xác nhận. * Hệ thống xóa quốc gia sản xuất phim đã chọn và cập nhật lại DB.  1. Tìm kiếm quốc gia sản xuất phim  * Hệ thống yêu cầu Quản trị hệ thống nhập vào thông tin cần tìm kiếm. * Quản trị hệ thống nhập vào thông tin và chọn nút “Tìm kiếm”.   Quản trị hệ thống có thể nhập thông tin tìm kiếm theo các trường:  Tên quốc gia sản xuất phim   * Hệ thống kiểm tra và hiển thị dữ liệu tương ứng. * (Quản trị hệ thống có thể chọn chức năng “Cập nhật quốc gia sản xuất phim”, “xóa” các quốc gia sản xuất phim với mỗi kết quả tìm kiếm) * Nếu Quản trị hệ thống chọn chức năng “Cập nhật quốc gia sản xuất phim” hệ thống hiển thị chức năng cập nhật quốc gia sản xuất phim như mô tả mục 2  ở trên. * Nếu Quản trị hệ thống chọn chức năng “xóa” để xóa quốc gia sản xuất phim hệ thống hiển thị chức năng xóa quốc gia sản xuất phim như mô tả mục 3  ở trên. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| **Thêm mới quốc gia sản xuất phim/Cập nhật quốc gia sản xuất phim – Nhập thiếu thông tin**  Nếu Quản trị hệ thống nhập thiếu thông tin tên quốc gia (yêu cầu bắt buộc và có dấu \* đỏ) thì hệ thống sẽ báo lỗi. Quản trị hệ thống có thể nhập lại hoặc hủy hành động này.  **Thêm mới quốc gia sản xuất phim/Cập nhật quốc gia sản xuất phim – Tên quốc gia đã tồn tại**  Nếu Tên quốc gia sản xuất phim hoặc Mã số quốc gia sản xuất phim nhập vào đã tồn tại thì hệ thống sẽ báo lỗi. Quản trị hệ thống có thể nhập lại hoặc hủy hành động này.  **Tìm kiếm quốc gia sản xuất phim – Không tìm thấy dữ liệu**  Nếu hệ thống không tìm thấy thông tin mà Quản trị hệ thống nhập vào thì hệ thống sẽ hiển thị thông báo “Không tìm thấy dữ liệu”.  **Xóa quốc gia sản xuất phim - Tồn tại ràng buộc giữa các bảng dữ liệu**  Nếu quốc gia sản xuất phim Quản trị hệ thống muốn xóa tồn tại ràng buộc giữa các bảng dữ liệu khác thì việc xóa thất bại. | | | |
| **Giao diện minh họa** | | | |
| Hình 5.37. Giao diện quản lý quốc gia    Hình 5.38. Giao diện xem chi tiết một quốc gia    Hình 5.39. Giao diện thêm mới quốc gia    Hình 5.40. Giao diện chỉnh sửa quốc gia    Hình 5.41. Giao diện xóa một quốc gia    Hình 5.42. Giao diện xóa nhiều quốc gia    Hình 5.43. Giao diện tìm kiếm quốc gia | | | |
| **Sơ đồ Lớp phân tích (Analysis class diagram)** | | | |
|  | | | |
| **Sơ đồ Trình tự (Sequence diagram)** | | | |
|  | | | |

## UC#008: Quản lý diễn viên

|  |  |  |  |
| --- | --- | --- | --- |
| **UC #008** | | **QUẢN LÝ DIỄN VIÊN** | **Độ phức tạp:  Trung bình** |
| **Mô tả** | | Use case này cho phép người quản trị quản lý các diễn viên trong hệ thống. Bao gồm việc thêm thay đổi tìm kiếm và xoá thông tin diễn viên trong hệ thống. | |
| **Tác nhân** | **Chính** | Người quản trị hệ thống | |
| **Phụ** | Không | |
| **Tiền điều kiện** | | Người quản trị phải đăng nhập vào hệ thống trước khi use case bắt đầu | |
| **Hậu điều kiện** | **Thành công** | Nếu use case thành công, thông tin về diễn viên được thêm, cập nhật, xoá khỏi hệ thống. | |
| **Lỗi** | Trạng thái của hệ thống không thay đổi. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Use case này bắt đầu khi người quản trị muốn thêm và/hoặc sửa thông tin diễn viên trong hệ thống.   * Hệ thống hiển thị danh sách các diễn viên có trong cơ sở dữ liệu. * Người quản trị xác định chức năng mà anh/cô ấy muốn sử dụng (thêm hoặc cập nhật thông tin một diễn viên) * Ngay khi người quản trị hệ thống xác định xong chức năng muốn thực hiện, một trong số các luồng phụ sẽ được thực thi. Nếu người quản trị chọn “thêm diễn viên”, luồng phụ *Thêm diễn viên* sẽ được thực thi. Nếu người quản trị chọn “cập nhật diễn viên”, luồng phụ *Cập nhật thông tin diễn viên* sẽ được thực thi. Nếu người quản trị hệ thống chọn “Tìm kiếm”, luồng phụ *Tìm kiếm diễn viên* sẽ được thực thi. Nếu người quản trị hệ thống chọn “Xoá”, luồng phụ *Xoá diễn viên* sẽ được thực thi.  1. ***Thêm diễn viên:***  * Hệ thống yêu cầu người quản trị nhập thông tin diễn viên cần thêm, bao gồm: * Tên diễn viên (bắt buộc) * Quốc gia (chọn từ hệ thống) * Tiểu sử * Ngay khi người quản trị hệ thống cung cấp các thông tin được yêu cầu, hệ thống tạo ra và gán cho diễn viên mới một ID duy nhất. Diễn viên mới sẽ được thêm vào hệ thống. * Hệ thống cung cấp cho người quản trị ID mới của diễn viên.  1. ***Cập nhật diễn viên:***  * Hệ thống thực hiện truy vấn trong cơ sở dữ liệu và hiển thị thông tin chi tiết diễn viên vừa chọn. * Người quản trị thực hiện thay đổi thông tin diễn viên. Bao gồm các thông tin đã được ghi rõ trong luồng phụ *Thêm diễn viên.* * Ngay khi người quản trị cập nhật thông tin cần thiết, hệ thống lưu vào bản ghi các thông tin được cập nhật.  1. ***Tìm kiếm diễn viên***  * Hệ thống hiển thị và yêu cầu người dùng nhập vào thông tin muốn tìm kiếm theo các trường:   + Tên diễn viên   + Quốc gia * Nhân viên quản lý nhập vào thông tin và chọn “Search”. * Hệ thống kiểm tra và hiển thị dữ liệu tương ứng.  1. ***Xoá diễn viên:***  * Hệ thống thực hiện truy vấn trong cơ sở dữ liệu và hiển thị thông tin chi tiết diễn viên vừa chọn. * Hệ thống đưa ra cho người quản trị yêu cầu xác nhận xóa. * Người quản trị xác nhận yêu cầu xóa. * Hệ thống xác nhận yêu cầu xóa của người quản trị và lập tức loại bỏ diễn viên được chọn khỏi hệ thống. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| **<<Luồng phát sinh 1>> Thông tin thêm mới hoặc cập nhật không hợp lệ** | | | |
| Nếu trong luồng phụ *Cập nhật diễn viên* hoặc luồng phụ *Thêm diễn viên*, người quản trị nhập thiếu hoặc nhập vào dữ liệu không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi. Sau đó người quản trị có thể nhập lại hoặc huỷ thao tác. | | | |
| **<<Luồng phát sinh 2>> Huỷ bỏ lệnh xoá** | | | |
| Nếu trong luồng phụ *Xoá diễn viên*, người quản trị hệ thống quyết định không xoá, lệnh xoá sẽ bị huỷ và luồng chính được khởi động lại trạng thái ban đầu. | | | |
| **Giao diện minh họa** | | | |
| Hình 5.44. Giao diện quản lý diễn viên    Hình 5.45. Giao diện xem chi tiết một diễn viên    Hình 5.46. Giao diện thêm mới diễn viên    Hình 5.47. Giao diện chỉnh sửa diễn viên    Hình 5.48. Giao diện xóa một diễn viên    Hình 5.49. Giao diện xóa nhiều diễn viên    Hình 5.50. Giao diện tìm kiếm diễn viên | | | |

|  |
| --- |
| **THIẾT KẾ UML** |
| **Sơ đồ Lớp phân tích (Analysis class diagram)** |
|  |
| **Sơ đồ Trình tự (Sequence diagram)** |
|  |

## UC#009: Quản lý phim

|  |  |  |  |
| --- | --- | --- | --- |
| **UC #009** | | **QUẢN LÝ PHIM** | **Độ phức tạp : Medium** |
| **Mô tả** | | Usecase này cho phép người dùng quản lý thông tin liên quan đến các Phim được lưu trữ trong Hệ thống, bao gồm:   * Xem chi tiết thông tin phim * Thêm mới thông tin phim vào Hệ thống * Xóa bỏ thông tin phim khỏi Hệ thống * Sửa đổi (cập nhập) thông tin phim đã có trong Hệ thống * Tìm kiếm thông tin phim đã có trong Hệ thống | |
| **Tác nhân chính** | | Nhân viên quản lý | |
| **Tác nhân phụ** | | Người quản trị Hệ thống | |
| **Tiền điều kiện** | | Người dùng phải đăng nhập vào Hệ thống trước khi Usecase bắt đầu | |
| **Hậu điều kiện** | **Thành công** | Nêu Usecase thành công, thông tin liên quan đến Giá Vé Xem Phim sẽ được liệt kê, thêm mới, cập nhập hoặc xóa bỏ trong Hệ thống | |
| **Lỗi** | Trạng thái của Hệ thống không thay đổi | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Usecase này bắt đầu khi Người dùng muốn thực hiện các thao tác thêm, sửa, xóa hoặc xem các thông tin liên quan đến Phim trong Hệ thống   * Hệ thống hiển thị danh sách Phim đã được lưu trữ trong Hệ thống.   Mỗi phim trong danh sách bao gồm các thông tin: Tên phim, Thể loại, Poster.   * Người dùng lựa chọn thao tác muốn thực hiện (bao gồm thêm, sửa, xóa thông tin liên quan đến Phim trong Hệ thống). Luồng con sẽ được thực thi dựa theo yêu cầu cầu của Người dùng * Trường hợp Người dùng chọn trực tiếp vào Tên phim, luồng con xem chi tiết thông tin phim sẽ được thực thi. * Trường hợp Người dùng chọn “Thêm phim”, luồng con thêm mới thông tin cho phim sẽ được thực thi * Trường hợp Người dùng chọn “Cập nhập phim”, luồng con cập nhật thông tin cho phim sẽ được thực thi * Trường hợp Người dùng chọn “Xóa phim”, luồng con thêm mới thông tin cho phim sẽ được thực thi * Trường hợp Người dùng có nhu cầu tìm kiếm thông tin liên quan đến phim, luồng con tìm kiếm phim sẽ được thực thi.   *Luồng con 1: Xem chi tiết thông tin phim*   * Hệ thống hiển thị đầy đủ và chi tiết các thông tin liên quan phim bất kì mà Người dùng đã lựa chọn   *Luồng con 2: Thêm phim*   * Hệ thống yêu cầu Người dùng nhập đầy đủ các thông tin liên quan đến Phim cần thêm bao gồm * Tên phim\* * Poster\* * Thời lượng chiếu\* * Diễn viên\*(bắt buộc lựa chọn từ danh sách thể loại mà Người dùng đã nhập trong chức năng Quản lý Diễn viên) * Quốc gia\*(bắt buộc lựa chọn từ danh sách thể loại mà Người dùng đã nhập trong chức năng Quản lý Quốc gia) * Thể loại \*(bắt buộc lựa chọn từ danh sách thể loại mà Người dùng đã nhập trong chức năng Quản lý thể loại phim) * Trailer phim.   (các thông tin dánh \*: Yêu cầu người dùng bắt buộc phải nhập)   * Người dùng cung cấp đầy đủ thông tin và gửi yêu cầu thêm Phim cho Hệ thống * Hệ thống tự sinh ra và gán cho Phim đó một Mã phim duy nhất. * Hệ thống cung cấp cho Người dùng Mã phim của phim mới và yêu cầu người dùng xác nhận việc thêm phim vào Hệ thống * Hệ thống lưu thông tin của phim mới   *Luồng con 3: Cập nhật phim.*   * Hệ thống hiển thị chi tiết thông tin của phim mà Người dùng đã chọn sửa. * Người dùng cập nhập trực tiếp thông tin cần thay đổi, và gửi yêu sửa cho Hệ thống. * Hệ thống yêu cầu Người dùng xác nhận việc thay đổi thông tin Phim * Hệ thống lưu thông tin vừa cập nhập   *Luồng con 4: Xóa thông tin phim*   * Hệ thống yêu cầu Người dùng xác nhận xóa Phim đã lựa chọn * Người dùng xác nhận việc xóa * Hệ thống xóa bỏ tất cả các thông tin liên quan đến Phim đó.   *Luồng con 5: Tìm kiếm phim*   * Hệ thống yêu cầu Người dùng nhập từ khóa liên quan đến nôi dung cần tìm kiếm (giới hạn nội dung từ khóa có thể nhập bao gồm: mã phim, tên phim, thể loại, diễn viên, quốc gia, thời lượng chiếu) * Người dùng nhập từ khóa liên quan đến phim cần tìm kiếm và gửi yêu cầu tìm kiếm * Hệ thống tiến hành truy vấn dữ liệu để tìm thông tin khớp tới từ khóa mà Người dùng cung cấp * Luồng sự kiện kết thúc khi Hệ thống hiện lên danh sách kết quả các phim liên quan đến Từ khóa tìm kiếm | | | |
| **Luồng phụ của sự kiện/Kịch bản phát sinh** | | | |
| *Luồng phụ của Luồng con 2:*  *Luồng phụ 2.1:* Người dùng không nhập đầy đủ các thông tin bắt buộc mà Hệ thống yêu cầu   * Tại bước 2 của luồng con Người dùng không nhập đầy đủ bắt buộc liên quan đến phim cần thêm mới mà Hệ thống yêu cầu * Hệ thống thông báo cho Người dùng “Nhập đầy đủ các thông tin bắt buộc” * Người dùng Nhập lại đầy đủ thông tin và gửi yêu cầu thêm mới phim * Các bước sau của luồng phụ tương tự như luồng con   *Luồng phụ 2.2*: Hệ thống chưa lưu thông tin phim mới vào Hệ thống   * Thông tin của phim mới chưa được thêm vào hệ thống. Hệ thống thông báo “Quá trình thêm mới không thành công” (có thể là do lỗi hệ thống) * Nếu người dùng vẫn có nhu cầu thêm phim thì thực hiện lại các thao tác từ bước 1 của luồng con, ngườc lại luồng sự kiện kết thúc.   *Luồng phụ của Luồng con 3:*  *Luồng phụ 3.2*: Hệ thống chưa lưu thông tin phim vừa mới được cập nhập vào Hệ thống   * Thông tin của phim do mà Người dùng mới cập nhập chưa được lưu vào hệ thống. Hệ thống thông báo “Quá trình cập nhập không thành công” (có thể là do lỗi hệ thống) * Nếu người dùng vẫn có nhu cầu sửa thông tin phim thì thực hiện lại các thao tác từ bước 1 của luồng con, ngườc lại luồng sự kiện kết thúc.   *Luồng phụ của Luồng con 4:*  Luồng phụ 4.1: Người dùng không xác nhận yêu cầu xóa phim khỏi Hệ thống   * Tại bước 1 của luồng con, Người dùng không xác nhận yêu cầu xóa phim * Hệ thống hủy bỏ yêu cầu xóa phim, luồng sự kiện kết thúc   *Luồng phụ 4.2:* Hệ thống chưa xóa bỏ thông tin người dùng yêu cầu   * Thông tin của phim đó chưa được xóa bỏ khỏi vào hệ thống. Hệ thống thông báo “Quá trình xóa không thành công” (có thể là do lỗi hệ thống) * Nếu người dùng vẫn có nhu cầu xóa phim thì thực hiện lại các thao tác từ bước 1 của luồng con, ngườc lại luồng sự kiện kết thúc.   *Luồng phụ 5 của Luồng con 5:*  *Luồng phụ 5.1*: Hệ thống không tìm kiếm thấy thông tin phim liên quan đến từ khóa cần tìm kiếm   * Tại bước 3 của Luồng con, sau khi Người dùng cung cấp từ khóa cho Hệ thống, Hệ thống không thể tìm thấy thông tin phim liên quan đến từ khóa tìm kiếm. (có thể do từ khóa Người dùng cung cấp chưa tồn tại hoặc không nằm trong giới hạn từ khóa, hoặc do lỗi Hệ thống). * Hệ thống thông báo cho Người dùng “Không thấy kết quả cần tìm kiếm. Thông tin bạn nhập có thể chưa tồn tại trong Hệ thống”. * Nếu người dùng vẫn có nhu cầu tìm kiếm thì thực hiện lại các thao tác từ bước 1 của luồng con, ngược lại luồng sự kiện kết thúc | | | |
| **Giao diện minh họa** | | | |
| Hình 5.51. Giao diện quản lý phim    Hình 5.52. Giao diện xem chi tiết một phim    Hình 5.53. Giao diện thêm mới phim    Hình 5.54. Giao diện chỉnh sửa phim    Hình 5.55. Giao diện xóa một phim    Hình 5.56. Giao diện xóa nhiều phim    Hình 5.57. Giao diện tìm kiếm phim | | | |

|  |
| --- |
| **Sơ đồ Lớp phân tích (Analysis class diagram)** |
|  |

|  |
| --- |
| **Sơ đồ Trình tự (Sequence diagram)** |
| SequenceDiagram1 |

## UC#010: Quản lý loại phòng

|  |  |  |  |
| --- | --- | --- | --- |
| **UC #010** | | **QUẢN LÝ LOẠI PHÒNG** | **Độ phức tạp: High** |
| **Mô tả** | | Use case này cho phép người quản trị quản lý các loại phòng trong hệ thống. Bao gồm việc thêm, thay đổi và tìm kiếm thông tin loại phòng trong hệ thống. | |
| **Tác nhân** | **Chính** | Người quản trị hệ thống, Nhân viên quản lý | |
| **Phụ** |  | |
| **Tiền điều kiện** | | Người quản trị phải đăng nhập vào hệ thống trước khi use case bắt đầu | |
| **Hậu điều kiện** | **Thành công** | Nếu use case thành công, thông tin về loại phòng được thêm, cập nhật trong hệ thống. | |
| **Lỗi** | Trạng thái của hệ thống không thay đổi. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Use case này bắt đầu khi người quản trị muốn thêm, sửa, và/hoặc tìm kiếm thông tin loại phòng trong hệ thống.   * Hệ thống hiển thị danh sách các loại phòng có trong cơ sở dữ liệu. * Người quản trị xác định chức năng mà anh/cô ấy muốn sử dụng (thêm hoặc cập nhật hoặc tìm kiếm thông tin một loại phòng) * Ngay khi người quản trị hệ thống xác định xong chức năng muốn thực hiện, một trong số các luồng phụ sẽ được thực thi. Nếu người quản trị chọn “thêm loại phòng”, luồng phụ *Thêm loại phòng* sẽ được thực thi. Nếu người quản trị chọn “cập nhật loại phòng”, luồng phụ *Cập nhật thông tin loại phòng* sẽ được thực thi. Nếu người quản trị hệ thống chọn “Tìm kiếm”, luồng phụ *Tìm kiếm loại phòng* sẽ được thực thi.   ***1. Thêm loại phòng:***   * Hệ thống yêu cầu người quản trị nhập thông tin loại phòng cần thêm, bao gồm: * Tên loại phòng (bắt buộc) * Mô tả * Trạng thái: kích hoạt hoặc không * Ngay khi người quản trị hệ thống cung cấp các thông tin được yêu cầu, hệ thống tạo ra và gán cho loại phòng mới một ID duy nhất và tạo phương thức kích hoạt mặc định “active” (nếu người dùng không chọn trạng thái). Loại phòng mới sẽ được thêm vào hệ thống. * Hệ thống cung cấp cho người quản trị ID mới của loại phòng.   ***2. Cập nhật loại phòng:***   * Hệ thống thực hiện truy vấn trong cơ sở dữ liệu và hiển thị thông tin chi tiết loại phòng vừa chọn. * Người quản trị thực hiện thay đổi thông tin loại phòng. Bao gồm các thông tin đã được ghi rõ trong luồng phụ *Thêm loại phòng.* * Ngay khi người quản trị cập nhật thông tin cần thiết, hệ thống lưu vào bản ghi các thông tin được cập nhật.   ***3. Tìm kiếm loại phòng:***   * Hệ thống hiển thị và yêu cầu người dùng nhập vào thông tin muốn tìm kiếm theo các trường:   + Tên loại phòng * Nhân viên quản lý nhập vào thông tin và chọn “Search”. * Hệ thống kiểm tra và hiển thị dữ liệu tương ứng. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| **<<Luồng phát sinh >> Thông tin thêm mới hoặc cập nhật không hợp lệ** | | | |
| Nếu trong luồng phụ *Cập nhật loại phòng* hoặc luồng phụ *Thêm loại phòng*, người quản trị nhập thiếu hoặc nhập vào dữ liệu không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi. Sau đó người quản trị có thể nhập lại hoặc huỷ thao tác. | | | |
| **Giao diện minh họa** | | | |
|  | | | |
| **THIẾT KẾ UML** | | | |
| **Sơ đồ Lớp phân tích (Analysis class diagram)** | | | |
| RoomType | | | |

|  |
| --- |
| **Sơ đồ Trình tự (Sequence diagram)** |
| **RoomType** |

## UC#011: Quản lý phòng

|  |  |  |  |
| --- | --- | --- | --- |
| **UC #011** | | **QUẢN LÝ PHÒNG** | **Độ phức tạp: High** |
| **Mô tả** | | Use case này cho phép người quản trị quản lý các phòng trong hệ thống. Bao gồm việc thêm, thay đổi, xoá và tìm kiếm thông tin phòng trong hệ thống. | |
| **Tác nhân** | **Chính** | Người quản trị hệ thống, Nhân viên quản lý | |
| **Phụ** |  | |
| **Tiền điều kiện** | | Người quản trị phải đăng nhập vào hệ thống trước khi use case bắt đầu | |
| **Hậu điều kiện** | **Thành công** | Nếu use case thành công, thông tin về phòng được thêm, cập nhật, xoá khỏi hệ thống. | |
| **Lỗi** | Trạng thái của hệ thống không thay đổi. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Use case này bắt đầu khi người quản trị muốn thêm, sửa, và/hoặc tìm kiếm thông tin phòng trong hệ thống.   * Hệ thống hiển thị danh sách các phòng có trong cơ sở dữ liệu. * Người quản trị xác định chức năng mà anh/cô ấy muốn sử dụng (thêm hoặc cập nhật hoặc tìm kiếm thông tin một phòng) * Ngay khi người quản trị hệ thống xác định xong chức năng muốn thực hiện, một trong số các luồng phụ sẽ được thực thi. Nếu người quản trị chọn “thêm phòng”, luồng phụ *Thêm phòng* sẽ được thực thi. Nếu người quản trị chọn “cập nhật phòng”, luồng phụ *Cập nhật thông tin phòng* sẽ được thực thi. Nếu người quản trị hệ thống chọn “tìm kiếm phòng”, luồng phụ *Tìm kiếm phòng* sẽ được thực thi, luồng phụ *Xoá phòng* sẽ được thực thi.  1. ***Thêm phòng:***  * Hệ thống yêu cầu người quản trị nhập thông tin phòng cần thêm, bao gồm: * Tên phòng (bắt buộc) * Loại phòng (chọn từ hệ thống) * Trạng thái: kích hoạt hoặc không * Ngay khi người quản trị hệ thống cung cấp các thông tin được yêu cầu, hệ thống tạo ra và gán cho phòng mới một ID duy nhất và tạo phương thức kích hoạt mặc định “active” (nếu người dùng không chọn trạng thái). Phòng mới sẽ được thêm vào hệ thống. * Hệ thống cung cấp cho người quản trị ID mới của phòng.  1. ***Cập nhật phòng:***  * Hệ thống thực hiện truy vấn trong cơ sở dữ liệu và hiển thị thông tin chi tiết phòng vừa chọn. * Người quản trị thực hiện thay đổi thông tin phòng. Bao gồm các thông tin đã được ghi rõ trong luồng phụ *Thêm phòng.* * Ngay khi người quản trị cập nhật thông tin cần thiết, hệ thống lưu vào bản ghi các thông tin được cập nhật.  1. ***Tìm kiếm phòng:***  * Hệ thống hiển thị và yêu cầu người dùng nhập vào thông tin muốn tìm kiếm theo các trường:   + Tên phòng   + Tên loại phòng * Nhân viên quản lý nhập vào thông tin và chọn “Search”. * Hệ thống kiểm tra và hiển thị dữ liệu tương ứng.  1. ***Xoá phòng:***  * Hệ thống đưa ra cho người quản trị yêu cầu xác nhận xóa. * Người quản trị xác nhận yêu cầu xóa. * Hệ thống xác nhận yêu cầu xóa của người quản trị và lập tức loại bỏ phòng được chọn khỏi hệ thống. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| **<<Luồng phát sinh 1>> Thông tin thêm mới hoặc cập nhật không hợp lệ** | | | |
| Nếu trong luồng phụ *Cập nhật phòng* hoặc luồng phụ *Thêm phòng*, người quản trị nhập thiếu hoặc nhập vào dữ liệu không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi. Sau đó người quản trị có thể nhập lại hoặc huỷ thao tác. | | | |
| **<<Luồng phát sinh 2>> Huỷ bỏ lệnh xoá** | | | |
| Nếu trong luồng phụ *Xoá phòng*, người quản trị hệ thống quyết định không xoá, lệnh xoá sẽ bị huỷ và luồng chính được khởi động lại trạng thái ban đầu. | | | |
| **Giao diện minh họa** | | | |
| Hình 5.58. Giao diện quản lý phòng    Hình 5.59. Giao diện xem chi tiết một phòng    Hình 5.60. Giao diện thêm mới phòng    Hình 5.61. Giao diện chỉnh sửa phòng    Hình 5.62. Giao diện xóa một phòng    Hình 5.63. Giao diện xóa nhiều phòng    Hình 5.64. Giao diện tìm kiếm phòng | | | |
| **Sơ đồ Lớp phân tích (Analysis class diagram)** | | | |
| Room | | | |

|  |
| --- |
| **Sơ đồ Trình tự (Sequence diagram)** |
| C:\Users\MyPC\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Room.jpg |

## UC#012: Quản lý ghế

|  |  |  |  |
| --- | --- | --- | --- |
| **UC #012** | | **QUẢN LÝ GHẾ** | **Độ phức tạp: High** |
| **Mô tả** | | Use case này cho phép người quản trị quản lý các ghế trong hệ thống. Bao gồm việc thay đổi thông tin các ghế. | |
| **Tác nhân** | **Chính** | Người quản trị hệ thống, Nhân viên quản lý | |
| **Phụ** |  | |
| **Tiền điều kiện** | | Người quản trị phải đăng nhập vào hệ thống trước khi use case bắt đầu | |
| **Hậu điều kiện** | **Thành công** | Nếu use case thành công, thông tin về ghế được cập nhật trong hệ thống. | |
| **Lỗi** | Trạng thái của hệ thống không thay đổi. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Use case này bắt đầu khi người quản trị hoặc nhân viên quản lý muốn sửa thông tin ghế trong hệ thống.   * Hệ thống hiển thị danh sách các ghế có trong cơ sở dữ liệu. Danh sách ghế mặc định là ma trận 10x10 gồm 100 ghế đánh số từ A1, A2… * Người quản trị xác định chức năng mà anh/cô ấy muốn sử dụng. * Ngay khi người quản trị hệ thống xác định xong chức năng muốn thực hiện, luồng phụ sẽ được thực thi.   ***Cập nhật ghế:***   * Hệ thống thực hiện truy vấn trong cơ sở dữ liệu và hiển thị thông tin chi tiết ghế vừa chọn. * Người quản trị thực hiện thay đổi thông tin ghế. Bao gồm các thông tin: * Tên ghế * Trạng thái: kích hoạt hoặc không * Ngay khi người quản trị cập nhật thông tin cần thiết, hệ thống lưu vào bản ghi các thông tin được cập nhật. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| **<<Luồng phát sinh>> Thông tin thêm mới hoặc cập nhật không hợp lệ** | | | |
| Nếu trong luồng phụ *Cập nhật ghế,* người quản trị nhập thiếu hoặc nhập vào dữ liệu không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi. Sau đó người quản trị có thể nhập lại hoặc huỷ thao tác. | | | |
| **Giao diện minh họa** | | | |
| Hình 5.65. Giao diện quản lý ghế | | | |
| **THIẾT KẾ UML** | | | |
| **Sơ đồ Lớp phân tích (Analysis class diagram)** | | | |
| seat | | | |

|  |
| --- |
| **Sơ đồ Trình tự (Sequence diagram)** |
|  |

## UC#013: Quản lý suất chiếu

|  |  |  |  |
| --- | --- | --- | --- |
| **UC #013** | | **QUẢN LÝ SUẤT CHIẾU** | **Độ phức tạp: High** |
| **Mô tả** | | Use case này cho phép người quản trị quản lý các suất chiếu trong hệ thống. Bao gồm việc thêm, thay đổi, xoá và tìm kiếm thông tin suất chiếu trong hệ thống. | |
| **Tác nhân** | **Chính** | Người quản trị hệ thống, Nhân viên quản lý | |
| **Phụ** |  | |
| **Tiền điều kiện** | | Người quản trị phải đăng nhập vào hệ thống trước khi use case bắt đầu | |
| **Hậu điều kiện** | **Thành công** | Nếu use case thành công, thông tin về suất chiếu được thêm, cập nhật, xoá trong hệ thống. | |
| **Lỗi** | Trạng thái của hệ thống không thay đổi. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Use case này bắt đầu khi người quản trị muốn thêm, sửa, và/hoặc tìm kiếm thông tin suất chiếu trong hệ thống.   * Hệ thống hiển thị danh sách các suất chiếu có trong cơ sở dữ liệu. * Người quản trị xác định chức năng mà anh/cô ấy muốn sử dụng (thêm hoặc cập nhật hoặc tìm kiếm thông tin một suất chiếu) * Ngay khi người quản trị hệ thống xác định xong chức năng muốn thực hiện, một trong số các luồng phụ sẽ được thực thi. Nếu người quản trị chọn “thêm suất chiếu”, luồng phụ *Thêm suất chiếu* sẽ được thực thi. Nếu người quản trị chọn “cập nhật suất chiếu”, luồng phụ *Cập nhật thông tin suất chiếu* sẽ được thực thi. Nếu người quản trị hệ thống chọn “tìm kiếm suất chiếu”, luồng phụ *Tìm kiếm suất chiếu* sẽ được thực thi, luồng phụ *Xoá suất chiếu* sẽ được thực thi.  1. ***Thêm suất chiếu:***  * Hệ thống yêu cầu người quản trị nhập thông tin suất chiếu cần thêm, bao gồm: * Ngày giờ chiếu * Thời lượng chiếu * Tên phim (chọn từ hệ thống) * Phòng (chọn từ hệ thống) * Trạng thái: kích hoạt hoặc không * Ngay khi người quản trị hệ thống cung cấp các thông tin được yêu cầu, hệ thống tạo ra và gán cho suất chiếu mới một ID duy nhất và tạo phương thức kích hoạt mặc định “active” (nếu người dùng không chọn trạng thái). Suất chiếu mới sẽ được thêm vào hệ thống. * Hệ thống cung cấp cho người quản trị ID mới của suất chiếu.  1. ***Cập nhật suất chiếu:***  * Hệ thống thực hiện truy vấn trong cơ sở dữ liệu và hiển thị thông tin chi tiết suất chiếu vừa chọn. * Người quản trị thực hiện thay đổi thông tin suất chiếu. Bao gồm các thông tin đã được ghi rõ trong luồng phụ *Thêm suất chiếu.* * Ngay khi người quản trị cập nhật thông tin cần thiết, hệ thống lưu vào bản ghi các thông tin được cập nhật.  1. ***Tìm kiếm suất chiếu:***  * Hệ thống hiển thị và yêu cầu người dùng nhập vào thông tin muốn tìm kiếm theo các trường:   + Tên phim   + Ngày giờ chiếu   + Phòng * Nhân viên quản lý nhập vào thông tin và chọn “Search”. * Hệ thống kiểm tra và hiển thị dữ liệu tương ứng.  1. ***Xoá suất chiếu:***  * Hệ thống đưa ra cho người quản trị yêu cầu xác nhận xóa. * Người quản trị xác nhận yêu cầu xóa. * Hệ thống xác nhận yêu cầu xóa của người quản trị và lập tức loại bỏ suất chiếu được chọn khỏi hệ thống. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| **<<Luồng phát sinh 1>> Thông tin thêm mới hoặc cập nhật không hợp lệ** | | | |
| Nếu trong luồng phụ *Cập nhật suất chiếu* hoặc luồng phụ *Thêm suất chiếu*, người quản trị nhập thiếu hoặc nhập vào dữ liệu không hợp lệ, hệ thống sẽ hiển thị thông báo lỗi. Sau đó người quản trị có thể nhập lại hoặc huỷ thao tác. | | | |
| **<<Luồng phát sinh 2>> Huỷ bỏ lệnh xoá** | | | |
| Nếu trong luồng phụ *Xoá suất chiếu*, người quản trị hệ thống quyết định không xoá, lệnh xoá sẽ bị huỷ và luồng chính được khởi động lại trạng thái ban đầu. | | | |
| **Giao diện minh họa** | | | |
| Hình 5.66. Giao diện quản lý suất chiếu    Hình 5.67. Giao diện xem chi tiết một suất chiếu    Hình 5.68. Giao diện thêm mới suất chiếu    Hình 5.69. Giao diện chỉnh sửa suất chiếu    Hình 5.70. Giao diện xóa một suất chiếu    Hình 5.71. Giao diện xóa nhiều suất chiếu    Hình 5.72. Giao diện tìm kiếm suất chiếu | | | |
| **THIẾT KẾ UML** | | | |
| **Sơ đồ Lớp phân tích (Analysis class diagram)** | | | |
| ShowTimes | | | |

|  |
| --- |
| **Sơ đồ Trình tự (Sequence diagram)** |
|  |

## UC#014: Quản lý giá vé xem phim

|  |  |  |  |
| --- | --- | --- | --- |
| **UC #014** | | **QUẢN LÝ GIÁ VÉ XEM PHIM** | **Độ phức tạp: Medium** |
| **Mô tả** | | Usecase này cho phép người dùng quản lý thông tin liên quan đến các Phim được lưu trữ trong Hệ thống, bao gồm:   * Thêm mới thông tin Giá Vé Xem Phim vào Hệ thống * Xóa bỏ thông tin Giá Vé Xem Phim khỏi Hệ thống * Sửa đổi (cập nhập) thông tin Giá Vé Xem Phim đã có trong Hệ thống | |
| **Tác nhân chính** | | Nhân viên quản lý | |
| **Tác nhân phụ** | | Người quản trị Hệ thống | |
| **Tiền điều kiện** | | Người dùng phải đăng nhập vào Hệ thống trước khi Usecase bắt đầu | |
| **Hậu điều kiện** | **Thành công** | Nêu Usecase thành công, thông tin liên quan đến Giá Vé Xem Phim sẽ được liệt kê, thêm mới, cập nhập hoặc xóa bỏ trong Hệ thống | |
| **Lỗi** | Trạng thái của Hệ thống không thay đổi | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Usecase này bắt đầu khi Người dùng muốn thực hiện các thao tác thêm, sửa, xóa hoặc xem các thông tin liên quan đến Giá Vé Xem Phim trong Hệ thống   * Hệ thống hiển thị danh sách Giá Vé Xem Phim và đầy đủ các thông tin liên quan đã được lưu trữ trong Hệ thống.   Hệ thống cho phép Người dùng lựa chọn phương thức liệt kê danh sách Giá Vé Xem Phim theo các cách như sau:   * Liệt kê danh sách Giá Vé Xem Phim theo thứ tự thêm thông tin (thông tin nào thêm sau cùng sê được liệt kê trước tiên) * Liệt kê theo danh sách Giá Vé Xem Phim (từ cao đến thấp) * Người dùng lựa chọn thao tác muốn thực hiện (bao gồm thêm, sửa, xóa thông tin liên quan đến Giá Vé Xem Phim trong Hệ thống). Luồng con sẽ được thực thi dựa theo yêu cầu cầu của Người dùng * Trường hợp Người dùng chọn “Thêm Giá Vé Xem Phim”, luồng con thêm mới thông tin cho phim sẽ được thực thi * Trường hợp Người dùng chọn “Cập nhập Giá Vé Xem Phim”, luồng con cập thông tin cho phim sẽ được thực thi * Trường hợp Người dùng chọn “Xóa Giá Vé Xem Phim”, luồng con thêm mới thông tin cho phim sẽ được thực thi * Trường hợp Người dùng có nhu cầu tìm kiếm thông tin liên quan đến phim, luồng con tìm kiếm phim sẽ được thực thi.   *Luồng con 1: Thêm phim*   * Hệ thống yêu cầu Người dùng nhập đầy đủ các thông tin liên quan đến Phim cần thêm bao gồm: * Suất chiếu phim chiếu phim (bắt buộc lựa chọn từ danh sách suất chiếu Người dùng đãnhập trong chức năng Quản lý Suất chiếu phim) * Loại ghế (bắt buộc lựa chọn từ danh sách suất chiếu Người dùng đã nhập trong chức năng Quản lý Suất chiếu phim) * Loại phòng (bắt buộc lựa chọn từ danh sách suất chiếu Người dùng đãnhập trong chức năng Quản lý Suất chiếu phim) * Giá Vé Xem Phim * Thời gian áp dụng giá   (các thông tin người dùng bắt buộc phải nhập)   * Người dùng cung cấp đầy đủ thông tin và gửi yêu cầu Thêm phim cho Hệ thống * Hệ thống tự sinh ra và gán cho Giá Vé Xem Phim đó một Mã Giá Vé Xem Phim duy nhất. * Hệ thống cung cấp cho Người dùng Mã Giá Vé Xem Phim và yêu cầu người dùng xác nhận việc thêm Giá Vé Xem Phim vào Hệ thống * Hệ thống lưu thông tin của Giá Vé Xem Phim mới thêm * Luồng sự kiện kết thúc khi Người dùng nhận được thông báo “Thêm mới thành công”.   *Luồng con 2: Cập nhật Giá Vé Xem Phim.*   * Hệ thống hiển thị chi tiết thông tin của Giá Vé Xem Phim mà Người dùng đã chọn sửa. * Người dùng cập nhập trực tiếp thông tin cần thay đổi, và gửi yêu sửa cho Hệ thống. * Hệ thống yêu cầu Người dùng xác nhận việc thay đổi thông tin Giá Vé Xem Phim * Hệ thống lưu thông tin vừa cập nhập * Luồng sự kiến kết thúc khi Người dùng nhận được thông báo “Cập nhập thành công”   *Luồng con 3: Xóa thông tin Giá Vé Xem Phim*   * Hệ thống yêu cầu Người dùng xác nhận xóa Giá Vé Xem Phim đã lựa chọn * Người dùng xác nhận việc xóa * Hệ thống xóa bỏ tất cả các thông tin liên quan đến Giá Vé Xem Phim đó. * Luồng sự kiện kết thúc khi Người dùng nhận được thông báo “Xóa thành công”   *Luồng con 4: Tìm kiếm thông tin liên quan đến Giá Vé Xem Phim*   * Hệ thống yêu cầu Người dùng nhập từ khóa liên quan đến nôi dung cần tìm kiếm (giới hạn nội dung từ khóa có thể nhập bao gồm: mã giá vẽ xem phim, Loại phòng, Suất chiếu, Loại ghế) * Người dùng nhập từ khóa liên quan đến thông tin Giá vé xem phim cần tìm kiếm và gửi yêu cầu tìm kiếm * Hệ thống tiến hành truy vấn dữ liệu để tìm thông tin khớp tới từ khóa mà Người dùng cung cấp * Luồng sự kiện kết thúc khi Hệ thống hiện lên danh sách kết quả các thông tin Giá Vé Xem Phim có liên quan đến Từ khóa tìm kiếm | | | |
| **Luồng phụ của sự kiện/Kịch bản phát sinh** | | | |
| *Luồng phụ của Luồng con 1:*  *Luồng phụ 1.1:* Người dùng không nhập đầy đủ các thông tin bắt buộc mà Hệ thống Yêu cầu   * Tại bước 2 của luồng con Người dùng không nhập đầy đủ bắt buộc liên quan đến Giá vé xem phim cần thêm mới mà Hệ thống yêu cầu * Hệ thống thông báo cho Người dùng “Nhập đầy đủ các thông tin bắt buộc” * Người dùng Nhập lại đầy đủ thông tin và gửi yêu cầu thêm mới phim * Các bước sau của luồng phụ tương tự như luồng con   *Luồng phụ 1.2:* Hệ thống chưa lưu thông tin giá vé mới vào Hệ thống   * Thông tin về Giá vé xem phim mới chưa được thêm vào hệ thống. Hệ thống thông báo “Quá trình thêm mới không thành công” (có thể là do lỗi hệ thống) * Nếu người dùng vẫn có nhu cầu thêm Giá vé xem phim mới thì thực hiện lại các thao tác từ bước 1 của luồng con, ngườc lại luồng sự kiện kết thúc.   *Luồng phụ của Luồng con 2:*  *Luồng phụ 2.1*: Hệ thống chưa cập nhập thông tin giá vé người dùng vừa sửa   * Thông tin của Giá vé xem phim mà Người dùng mới cập nhập chưa được lưu vào hệ thống. Hệ thống thông báo “Quá trình cập nhập không thành công” (có thể là do lỗi hệ thống) * Nếu người dùng vẫn có nhu cầu sửa thông tin về Giá vé xem phim thì thực hiện lại các thao tác từ bước 1 của luồng con, ngườc lại luồng sự kiện kết thúc.   *Luồng phụ của Luồng con 3:*  *Luồng phụ 3.1:* Người dùng không xác nhận yêu cầu xóa Giá vé xem phim khỏi Hệ thống   * Tại bước 1 của luồng con, Người dùng không xác nhận yêu cầu xóa Giá vé xem phim * Hệ thống hủy bỏ yêu cầu xóa Giá vé xem phim, luồng sự kiện kết thúc   *Luồng phụ 3.2:* Người dùng không nhận được thông báo “Xóa thành công”   * Thông tin về Giá vé xem phim đó chưa được xóa bỏ khỏi vào hệ thống. Hệ thống thông báo “Quá trình xóa không thành công” (có thể là do lỗi hệ thống) * Nếu người dùng vẫn có nhu cầu xóa thông tin Giá vé xem phim thì thực hiện lại các thao tác từ bước 1 của luồng con, ngườc lại luồng sự kiện kết thúc.   *Luồng phụ 4 của Luồng con 4:* Hệ thống không tìm kiếm thấy thông tin về Giá vé xem phim liên quan đến từ khóa cần tìm Kiếm   * Tại bước 3 của Luồng con, sau khi Người dùng cung cấp từ khóa cho Hệ thống, Hệ thống không thể tìm thấy thông tin về Giá vé xem phim liên quan đến từ khóa tìm kiếm. (có thể do từ khóa Người dùng cung cấp chưa tồn tại hoặc không nằm trong giới hạn từ khóa, hoặc do lỗi Hệ thống). * Hệ thống thông báo cho Người dùng “Không thấy kết quả cần tìm kiếm. Thông tin bạn nhập có thể chưa tồn tại trong Hệ thống”. * Nếu người dùng vẫn có nhu cầu tìm kiếm thì thực hiện lại các thao tác từ bước 1 của luồng con, ngược lại luồng sự kiện kết thúc | | | |
| **Giao diện minh họa** | | | |
| Hình 5.73. Giao diện quản lý giá vé    Hình 5.74. Giao diện xem chi tiết một giá vé    Hình 5.75. Giao diện thêm mới giá vé    Hình 5.76. Giao diện chỉnh sửa giá vé    Hình 5.77. Giao diện xóa một giá vé    Hình 5.78. Giao diện xóa nhiều giá vé    Hình 5.79. Giao diện tìm kiếm giá vé | | | |
| **Sơ đồ Lớp phân tích (Analysis class diagram)** | | | |
|  | | | |

|  |
| --- |
| **Sơ đồ Trình tự (Sequence diagram)** |
| C:\Users\CUNCON\AppData\Local\Microsoft\Windows\INetCache\Content.Word\SequenceDiagram1.jpg |

## UC#015: Tìm kiếm

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#015** | | **Tìm kiếm phim** | **Độ phức tạp: Cao** |
| **Mô tả** | | Chức năng này cho phép Khách hàng có thể tìm kiếm phim, hệ thống rạp mà mình muốn xem, ngoài ra hệ thống còn cung cấp chức năng tìm kiếm nâng cao theo quốc gia sản xuất phim. | |
| **Tác nhân** | **Chính** | Khách hàng | |
| **Phụ** | Không | |
| **Tiền điều kiện** | | Không | |
| **Hậu điều kiện** | **Thành công** | Hiển thị dữ liệu được tìm kiếm. | |
| **Lỗi** | Trạng thái hệ thống không thay đổi. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Chức năng này bắt đầu khi khách hàng muốn tìm kiếm thông tin mong muốn.  1. Tìm kiếm thông thường  Khách hàng chọn chức năng “Tìm kiếm”. Khách hàng có thể nhập các thông tin tìm kiếm vào “ô tìm kiếm” theo các trường:   * Tên rạp * Tên thể loại phim * Tên phim   Hệ thống kiểm tra và hiển thị dữ liệu tương tứng.  Khách hàng có thể chọn vào dòng dữ liệu để xem chi tiết về dòng dữ liệu đó.  2. Tìm kiếm theo bộ lọc  Khách hàng chọn năng tìm kiếm theo bộ lọc với các trường.   * Thể loại * Quốc gia sản xuất * Chi nhánh rạp * Phim đang chiếu * Phim sắp chiếu * Phim đang hot   Nếu khách hàng chọn trường dữ liệu nào thì hệ thống sẽ trả về dữ liệu tương ứng là danh sách các bộ phim. Khách hàng có thể chọn bộ phim mà mình muốn để xem chi tiết. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| **Tìm kiếm phim – Không tìm thấy dữ liệu**  Nếu hệ thống không tìm thấy thông tin mà Khách hàng nhập vào thì hệ thống sẽ hiển thị thông báo “Không tìm thấy dữ liệu”. Khách hàng có thể nhập để tìm kiếm lại hoặc hủy hành động này. | | | |
| **Giao diện minh họa** | | | |
|  | | | |
| **THIẾT KẾ UML** | | | |
| **Sơ đồ Lớp phân tích (Analysis class diagram)** | | | |
|  | | | |
| **Sơ đồ Trình tự (Sequence diagram)** | | | |
|  | | | |

## UC#016: Đăng ký tài khoản KH

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#16** | | **Đăng ký tài khoản KH** | **Độ phức tạp:**  **Thấp** |
| **Mô tả** | | Chức năng này cho phép khách hàng đăng ký một tài khoản để đăng nhập và sử dụng các dịch vụ của hệ thống. | |
| **Tác nhân** | **Chính** | Khách hàng | |
| **Phụ** | Không | |
| **Tiền điều kiện** | | Khách hàng chọn đăng ký tài khoản trên hệ thống. | |
| **Hậu điều kiện** | **Thành công** | Khách hàng đăng ký tài khoản thàng công. | |
| **Lỗi** | Đăng ký tài khoản không thành công và phải đăng ký lại hoặc hủy thao tác. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Chức năng này bắt đầu khi Khách hàng muốn đăng ký một tài khoản trên hệ thống.  Khách hàng chọn nút “Đăng ký”.  Hệ thống hiển thị trang “Đăng ký” yêu cầu Khách hàng nhập đầy đủ thông tin bao gồm:   * Họ tên * Giới tính * Địa chỉ mail * Tên đăng nhập * Mật khẩu * Nhập lại mật khẩu * Địa chỉ * Số điện thoại * Loại tài khoản: đối với khách hàng thì loại tài khoản mặc định là “khách hàng”   Sau khi Khách nhập đầy đủ thông tin và chọn nút *“Đăng ký”*. Hệ thống sẽ kiểm tra thông tin mà Khách hàng vừa nhập, nếu thông tin hợp lệ thì hệ thống sẽ lưu thông tin tài khoản khách hàng vào DB và thông báo kết quả.   * Nếu khách hàng cung cấp thông tin thiếu/sai thông tin thì hệ thống hiển thị thông báo khách hàng nhập thiếu/sai thông tin. * Nếu tên tài khoản đã được tạo từ trước đó hệ thống sẽ hiển thị thông báo tài khoản đã tồn tại và quay lại trang đăng ký tài khoản. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
|  | | | |
| **Giao diện minh họa** | | | |
|  | | | |
| **THIẾT KẾ UML** | | | |
| **Sơ đồ Lớp phân tích (Analysis class diagram)** | | | |
|  | | | |
| **Sơ đồ Trình tự (Sequence diagram)** | | | |
|  | | | |

## UC#017: Đặt vé

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#017** | | **ĐẶT VÉ** | **Độ phức tạp:** |
| **Mô tả** | | Chức năng này cho phép người dùng đặt vé xem phim trên hệ thống | |
| **Tác nhân** | **Chính** | Khách hàng, Nhân viên quản lý | |
| **Phụ** | Quản trị hệ thống | |
| **Tiền điều kiện** | | Đối với khách hàng: Sau khi khách hàng đăng nhập trang dành cho khách hàng, tìm kiếm phim muốn xem và chọn chức năng Đặt vé.  Đối với Nhân viên quản lý/ Quản trị hệ thống: Sau khi đăng nhập vào trang quản trị, chọn suất chiếu và chọn chức năng Đặt vé | |
| **Hậu điều kiện** | **Thành công** | Đặt vé thành công | |
| **Lỗi** | Trạng thái hệ thống sẽ không thay đổi. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Chức năng này bắt đầu khi người dùng muốn đặt vé xem phim  *Luồng chính dành cho Nhân viên quản lý/ Quản trị hệ thống ở trang quản trị*  Muốn đặt vé xem phim trước tiên Nhân viên quản lý cần chọn suất chiếu muốn đặt vé:  Chọn Quản lý chọn chức năng Quản lý rạp -> Quản lý suất chiếu  Hệ thống hiển thị danh sách suất chiếu hiện có trên hệ thống, chọn suất chiếu muốn đặt bằng 2 cách:   * Cách 1: Chọn tên phim trong danh sách * Cách 2: Tìm kiếm suất chiếu bằng chức năng tìm kiếm suất chiếu – **UC014: Quản lý suất chiếu**   Nhân viên quản lý chon chức năng Đặt vé cần thực hiện tuần tự theo các bước trong form sau:  Bước 1: Xác nhận tài khoản   * Nhân viên quản lý cần phải xác nhận tài khoản trước khi đặt vé cho khách hàng bằng cách nhập tên tài khoản (bắt buộc) và mật khẩu (bắt buộc) của nhân viên đó. * Sau khi xác nhận thành công chọn Tiếp tục để chuyển sang bước tiếp theo.   Bước 2: Đặt ghế   * Chọn tên tài khoản khách hàng (bắt buộc) đặt vé gồm các tài khoản: nhân viên quản lý, quản trị hệ thống, khách hàng đã có trên hệ thống, khách hàng không có tài khoản * Chọn ghế có 2 cách chọn: chọn từng ghế hoặc chọn tất cả các ghế còn trống * Các loại ghế: ghế còn trống (màu xám), ghế đã được thành toán (màu đỏ), ghế đang chọn (màu xanh), ghế đã được chọn nhưng chưa thanh toán (màu vàng). * Nhân viên kích chọn những ghế còn trống sẽ hiển thị thành ghế đang chọn. Chọn tất cả các ghế còn trống bằng cách kích vào checkbox chọn tất cả. * Sau khi chọn ghế thành công người dùng chọn Tiếp tục để chuyển sang bước tiếp theo hoặc chọn Quay lại để quay lại bước trước đó.   Bước 3: Xác nhận   * Hệ thống sẽ hiển thị thông tin khách hàng đặt vé và vị trí chỗ ngồi * Thông tin khách hàng: tên, tên đầy đủ, email * Vị trí ghế ngồi: mã ghế, số hàng, số cột, trạng thái * Sau khi kiểm tra lại thông tin, nhân viên chọn Chấp nhận để hoàn thành việc đặt vé, hệ thống sẽ hiển thị thông báo “Đặt vé thành công” hoặc chọn Quay lại để quay lại bước trước đó.   *Luồng chính dành cho Khách hàng ở trang khách hàng*  Hệ thống sẽ hiển thị danh sách phim, mỗi phim gồm các trường thông tin sau:  Poster quảng cáo phim  Tiêu đề phim  Mô tả: thể loại, thời lượng  Người dùng có thể chọn “Đặt vé” theo 2 cách:  Cách 1: Người dùng chọn nút “Đặt vé” ngay dưới thông tin phim  Cách 2: Người dùng chọn phim bằng cách click vào poster quảng cáo hoặc tiêu đề phim, hệ thống sẽ hiển thị thông tin chi tiết phim. Nhấn nút “Đặt vé” ở dưới poster quảng cáo phim  Khi người dùng chọn nút “Đặt vé”, hệ thống yêu cầu người dùng thực hiện các thao tác với form đặt vé:  Bước 1: Xác thực tài khoản   * Khách hàng cần phải xác nhận tài khoản trước khi đặt vé bằng cách nhập tên tài khoản (bắt buộc) và mật khẩu (bắt buộc). * Sau khi xác nhận thành công chọn Tiếp tục để chuyển sang bước tiếp theo.   Bước 2: Chọn suất chiếu   * Khách hàng chọn suất chiếu muốn xem với các trường thông tin: Chọn rạp (thành phố, quận huyện), chọn phòng   Bước 3: Chọn vị trí ghế   * Các loại ghế: ghế còn trống (màu xám), ghế đã được thành toán (màu đỏ), ghế đang chọn (màu xanh), ghế đã được chọn nhưng chưa thanh toán (màu vàng). * Người dùng chỉ được phép chọn những ghế còn trống * Người dùng chỉ được phép chọn tối đa 8 ghế * Sau khi chọn ghế thành công người dùng chọn Tiếp tục để chuyển sang bước tiếp theo hoặc chọn Quay lại để quay lại bước trước đó.   Bước 4: Xác thực thông tin vé   * Hệ thống hiển thị thông tin vé: Tên rạp, tên phim, số lượng ghế, giá, ngày đặt * Hiển thị thông tin người đặt: tên tài khoản * Hiển thị vị trí ghế * Sau khi người dùng kiểm tra thông tin xong chọn Chấp nhân để hoàn tất việc đặt vé. * Hệ thống sẽ lưu thông tin vào CSDL và hiển thị thông báo “Đặt vé thành công. Mời quý khách đến trước giờ xem phim 30 phút để nhận vé. Cám ơn quý khách đã sử dụng dịch vụ của chúng tôi.” | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| *Đối với Luồng dành cho Nhân viên quản lý/ Quản trị hệ thống ở trang quản trị và Khách hàng ở trang khách hàng*  **Người dùng xác thực thông tin tài khoản không thành công ở bước 1**  Khi người dùng không nhập thông tin xác thực, hệ thống sẽ báo lỗi “Thông tin bắt buộc”  Khi người dùng nhập tài khoản không tồn tại trên hệ thống thì hệ thống sẽ báo lỗi “Tài khoản không tồn tại”  Khi người dùng nhập ít hơn 5 ký tự thì hệ thống sẽ thông báo lỗi “Cần nhập ít nhất 5 ký tự”  **Người dùng không chọn ghế và tên tài khoản của khách đặt ở bước 2**  Khi người dùng không chọn ghế thì khi chọn Tiếp tục hệ thống sẽ thông báo lỗi “Mời bạn chọn ghế”  Khi người dùng không chọn tên tài khoản của khách đặt thì khi chọn Tiếp tục hệ thống sẽ thông báo lỗi “Mời bạn chọn tài khoản của khách hàng” | | | |
| **Giao diện minh họa** | | | |
| Hình 4.80. Giao diện thêm mới đặt vé (bước 1)    Hình 4.81. Giao diện thêm mới đặt vé (bước 2)    Hình 4.82. Giao diện thêm mới đặt vé (bước 3) | | | |
| **Sơ đồ Lớp phân tích (Analysis class diagram)** | | | |
|  | | | |

|  |
| --- |
| **Sơ đồ Trình tự (Sequence diagram)** |
|  |

## UC#018: Quản lý thanh toán

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#018** | | **QUẢN LÝ THANH TOÁN** | **Độ phức tạp:** |
| **Mô tả** | | Chức năng này cho phép người dùng thực hiện xem chi tiết hóa đơn, thanh toán, xóa hóa đơn phim | |
| **Tác nhân** | **Chính** | Nhân viên quản lý | |
| **Phụ** | Quản trị hệ thống | |
| **Tiền điều kiện** | | Sau khi người dùng đăng nhập | |
| **Hậu điều kiện** | **Thành công** | Cập nhật thành công cho việc thanh toán và lưu dữ liệu vào hệ thống. | |
| **Lỗi** | Trạng thái hệ thống sẽ không thay đổi. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Luồng này bắt đầu khi người dùng muốn xem, thanh toán, xóa hóa đơn vé xem phim  Khi người dùng muốn xem chi tiết vé thì **luồng 1 – Xem chi tiết hóa đơn** được thực hiện  Khi người dùng muốn thanh toán vé thì **luồng 2 – Thanh toán** được thực hiện  Khi người dùng muốn xóa hóa đơn **thì luồng 3 – Xóa** **hóa đơn** được thực hiện  **Luồng 1 – Xem chi tiết hóa đơn**  Người dùng chọn button Xem chi tiết của hóa đơn (vé) ở danh sách vé mà người dùng muốn xem  Các thông tin của hóa đơn:  Thông tin khách hàng: Tên, tên đầy đủ, email  Vị trí ghế: mã ghế, số hàng, số cột, trạng thái (Đã thanh toán, Chưa thanh toán), link Xóa  **Luồng 2 – Thanh toán**  Người dùng thực hiện **luồng 1 - Xem chi tiết hóa đơn**  Ở trạng thái người dùng chọn Đã thanh toán và chọn button Lưu để hoàn tất việc thanh toán. Nếu người dùng không muốn thanh toán thì chọn button Quay lại để quay lại trang trước đó.  **Luồng 3 – Xóa hóa đơn**  Có 2 cách xóa hóa đơn:  Cách 1: Chọn button Xóa của hóa đơn ở danh sách hóa đơn  Cách 2: Chọn link Xóa ở **luồng 1 -Xem chi tiết**  Khi người dùng chọn Xóa, hệ thống hiển thị popup yêu cầu xác nhận. Nếu người dùng chọn Có thì hệ thống sẽ xóa và hiển thị thông báo. Nếu người dùng chọn Không thì hệ thống đóng popup.  Đối với những hóa đơn đã thanh toán thì không được phép xóa | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
|  | | | |
| **Giao diện minh họa** | | | |
| Hình 4.83. Giao diện thanh toán    Hình 4.84. Giao diện quản lý đặt vé (hóa đơn)    Hình 4.85. Giao diện xem chi tiết/chỉnh sửa một hóa đơn (vé)    Hình 4.86. Giao diện xóa một vé    Hình 4.87. Giao diện xóa nhiều vé | | | |
| **THIẾT KẾ UML** | | | |
| **Sơ đồ Lớp phân tích (Analysis class diagram)** | | | |
|  | | | |
| **Sơ đồ Trình tự (Sequence diagram)** | | | |
|  | | | |

## UC#019: Đăng xuất

|  |  |  |  |
| --- | --- | --- | --- |
| **UC#019** | | **Đăng xuất** | **Độ phức tạp:**  **Thấp** |
| **Mô tả** | | Chức năng này cho phép người dùng có tài khoản đăng xuất khỏi hệ thống. | |
| **Tác nhân** | **Chính** | Quản trị hệ thống, Nhân viên quản lý, Khách hàng. | |
| **Phụ** | Không | |
| **Tiền điều kiện** | | Đăng nhập thành công vào HT. | |
| **Hậu điều kiện** | **Thành công** | Người dùng đăng xuất khỏi hệ thống. | |
| **Lỗi** | Thực hiện lại hành động hoặc hủy bỏ hành động này. | |
| **ĐẶC TẢ CHỨC NĂNG** | | | |
| **Luồng sự kiện chính/Kịch bản chính** | | | |
| Chức năng này bắt đầu khi Nhân viên quản lý hoặc Khách hàng muốn đăng xuất khỏi hệ thống.  Người dùng chọn “Đăng xuất” ở tài khoản.  Hệ thống sẽ hiển thị thông báo xác nhận đăng xuất.  Người dùng xác nhận để đăng xuất khỏi hệ thống. | | | |
| **Luồng sự kiện phát sinh/Kịch bản phát sinh** | | | |
| **<<Luồng phát sinh>> Tự động đăng xuất theo timeout** | | | |
| Nếu người dùng không thao tác với hệ thống quá ngưỡng thời gian được thiết lập sẵn thì hệ thống sẽ tự động đăng xuất người dùng và giải phóng tài nguyên liên quan đến người đó. Luồng này nhằm tăng thêm tính bảo mật của hệ thống. | | | |
| **Giao diện minh họa** | | | |
| Hình 5.80. Giao diện đăng xuất | | | |
| **THIẾT KẾ UML** | | | |
| **Sơ đồ Lớp phân tích (Analysis class diagram)** | | | |
|  | | | |
| **Sơ đồ Trình tự (Sequence diagram)** | | | |
|  | | | |

1. JDBC (Java Database Connectivity) cho phép các nhà phát triển kết nối, truy vấn và cập nhật cơ sở dữ liệu bằng cách sử dụng Structured Query Language. Chuẩn JDBC API cung cấp cho các nhà phát triển Java tương tác với RDBMS khác nhau và truy cập dữ liệu bảng thông qua ứng dụng Java mà không cần biết các chi tiết của RDBMS và sử dụng Database Specific JDBC Drivers. [↑](#footnote-ref-1)
2. OU (Organizational Unit), mô hình dữ liệu các phòng ban, đơn vị trong Active Directory. [↑](#footnote-ref-2)