**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN**

**KHOA KHOA HỌC MÁY TÍNH**

****

**BÁO CÁO**

**ĐỒ ÁN CƠ SỞ 1**

***Đề tài:* QUẢN LÝ BÁN VÉ XEM PHIM**

**Sinh viên thực hiện :** **Nguyễn Văn Hiếu**

**Mã sinh viên :** **24IT067**

**GVHD** **: Th.S Đặng Thị Kim Ngân**

***Đà Nẵng, tháng 5 năm 2025***

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG VIỆT - HÀN**

**KHOA KHOA HỌC MÁY TÍNH**

****

**BÁO CÁO**

**ĐỒ ÁN CƠ SỞ 1**

***Đề tài:* QUẢN LÝ BÁN VÉ XEM PHIM**

**Sinh viên thực hiện :** **Nguyễn Văn Hiếu**

**Mã sinh viên :** **24IT067**

**GVHD** **: Th.S Đặng Thị Kim Ngân**

***Đà Nẵng, tháng 5 năm 2025***

LỜI CẢM ƠN

Xin chân thành cảm ơn các thầy cô khoa Khoa học máy tính đã hết lòng hỗ trợ và hướng dẫn em trong quá trình phát triển ứng dụng đồ án cơ sở 1.

Đặc biệt, em xin gửi đến giảng viên hướng dẫn cô Đặng Thị Kim Ngân, người đã tận tình hướng dẫn, giúp đỡ em trong suốt quá trình hoàn thành đồ án một lời cảm ơn sâu sắc nhất.

Trong suốt quá trình thực hiện đồ án này, em luôn nhận được sự quan tâm và chỉ dẫn tận tình từ các thầy cô. Các thầy cô không chỉ truyền đạt những kiến thức chuyên môn sâu rộng, mà còn tận tâm hướng dẫn em cách tiếp cận và giải quyết các vấn đề một cách khoa học và hiệu quả.

Khi em gặp phải những khó khăn trong việc thiết kế, xây dựng và triển khai ứng dụng, các thầy cô đã dành rất nhiều thời gian và tâm huyết để chỉ bảo, đưa ra những lời khuyên quý báu giúp em vượt qua những thách thức đó. Các thầy cô không chỉ hướng dẫn em về mặt kỹ thuật, mà còn chia sẻ những kinh nghiệm quý giá về quy trình phát triển phần mềm, giúp em nắm vững hơn những kiến thức và kỹ năng cần thiết cho công việc tương lai.

Mặc dù đã rất cố gắng hoàn thiện đồ án với tất cả sự nỗ lực, tuy nhiên đồ án “Xây dựng ứng dụng quản lí bán vé xem phim” chắc chắn sẽ không tránh khỏi những thiếu sót.  Em rất mong nhận được sự quan tâm, thông cảm và những lời góp ý của quý thầy cô và các bạn để đồ án có thể hoàn thiện tốt nhất.

Sự hỗ trợ tận tình và tâm huyết của các thầy cô thật sự là nguồn động lực vô giá giúp em hoàn thành tốt đồ án cơ sở 1. Em xin bày tỏ lòng biết ơn sâu sắc nhất đến các thầy cô.

Em xin chân thành cảm ơn!

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

MỤC LỤC

[LỜI CẢM ƠN i](#_Toc5123)

[NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN ii](#_Toc19195)

[MỤC LỤC iii](#_Toc5791)

[DANH MỤC HÌNH ẢNH v](#_Toc22013)

[DANH MỤC BẢNG BIỂU vi](#_Toc5427)

[DANH MỤC CÁC TỪ VIẾT TẮT vii](#_Toc29462)

[MỞ ĐẦU 1](#_Toc3210)

[CHƯƠNG 1: CƠ SỞ LÝ THUYẾT 4](#_Toc21955)

[1.1 Java 4](#_Toc27790)

[1.1.1 Khái niệm 4](#_Toc30519)

[1.1.2 Lịch sử phát triển 4](#_Toc14027)

[1.1.3 Vai trò 4](#_Toc13143)

[1.1.4 Đặc điểm 4](#_Toc15887)

[1.2 SQL Server 4](#_Toc17987)

[1.2.1 Khái niệm 4](#_Toc30782)

[1.2.2 Lịch sử phát triển 5](#_Toc30580)

[1.2.3 Vai trò 5](#_Toc7056)

[1.2.4 Đặc điểm 5](#_Toc26229)

[1.3 Intelliji IDEA 5](#_Toc19143)

[1.3.1 Khái niệm 6](#_Toc20163)

[1.3.2 Lịch sử phát triển 6](#_Toc28712)

[1.3.3 Vai trò 6](#_Toc861)

[1.3.4 Đặc điểm 6](#_Toc9537)

[CHƯƠNG 2: PHÂN TÍCH VÀ THIẾT KẾ HẾ THỐNG 7](#_Toc10239)

[2.1 Yêu cầu người dùng 7](#_Toc13751)

[2.2 Phân Tích Nghiệp Vụ Quản Lý 8](#_Toc7464)

[2.2.1 Đối với ADMIN 8](#_Toc32183)

[2.2.2 Đối với Người Dùng 9](#_Toc32087)

[2.3 Mô tả và thiết kế hệ thống 10](#_Toc10431)

[2.3.1 Kiến trúc tổng thể 10](#_Toc15649)

[2.3.2 Mô tả và Thiết kế cơ sở dữ liệu 12](#_Toc15319)

[2.3.3 Lưu đồ thuật toán/hoạt động cho bài toán quản lí bán vé xem phim 16](#_Toc19731)

[2.3.3.1 Đăng nhập vào hệ thống 16](#_Toc28278)

[2.3.3.2 Hệ thống đối với khách hàng 17](#_Toc21368)

[2.3.3.3 Hệ thống đối với Admin 18](#_Toc12195)

[2.3.3.4 Lưu ý 19](#_Toc1403)

[2.4 Thư viện, API của Java được sử dụng 19](#_Toc24857)

[CHƯƠNG 3: TRIỂN KHAI XÂY DỰNG GIAO DIỆN VÀ CHỨC NĂNG 21](#_Toc8770)

[3.1 Giao diện đăng nhập và chức năng 21](#_Toc24682)

[3.1.1 Đăng nhập 21](#_Toc26612)

[3.1.2 Đăng Ký (Người dùng) 24](#_Toc30338)

[3.2 Giao diện quản lý Admin 25](#_Toc22978)

[3.2.1 Trang chủ 25](#_Toc18423)

[3.2.2 Thông tin phim 26](#_Toc10093)

[3.2.3 Phòng chiếu 28](#_Toc29949)

[3.2.4 Suất chiếu 28](#_Toc18919)

[3.2.5 Nhân viên 32](#_Toc19976)

[3.2.6 Khách hàng 33](#_Toc17913)

[3.2.7 Thống kê 34](#_Toc27224)

[3.3 Giao diện Người dùng 35](#_Toc19733)

[3.3.1 Danh sách phim 35](#_Toc6146)

[3.3.2 Lịch sử đặt vé 38](#_Toc30141)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 39](#_Toc6457)

[TÀI LIỆU THAM KHẢO 42](#_Toc20521)

DANH MỤC HÌNH ẢNH

[Hình 3.1 Giao diện đăng nhập 21](#_Toc199347248)

[Hình 3.2 Đăng nhập thành công 23](#_Toc199347249)

[Hình 3.3 Giao diện Đăng Ký 24](#_Toc199347250)

[Hình 3.4 Tài khoản được mã hóa lưu vào CSDL 25](#_Toc199347251)

[Hình 3.5 Trang chủ 25](#_Toc199347252)

[Hình 3.6 Thông tin phim 26](#_Toc199347253)

[Hình 3.7 Chức năng Cập nhật 27](#_Toc199347254)

[Hình 3.8 Chức năng Thêm Phim 28](#_Toc199347255)

[Hình 3.9 Suất chiếu 29](#_Toc199347256)

[Hình 3.10 Khởi tạo suất chiếu 30](#_Toc199347257)

[Hình 3.11 Cập nhật trạng thái tự động của Suất chiếu 31](#_Toc199347258)

[Hình 3.12 Nhân viên 33](#_Toc199347259)

[Hình 3.13 Thông tin khách hàng A 33](#_Toc199347260)

[Hình 3.14 Thống kê 34](#_Toc199347261)

[Hình 3.15 Danh sách phim 35](#_Toc199347262)

[Hình 3.16 Xem tất cả suất chiếu của một phim 36](#_Toc199347263)

[Hình 3.17 Đặt vé 36](#_Toc199347264)

[Hình 3.18 Xác nhận thanh toán 37](#_Toc199347265)

[Hình 3.19 Hóa đơn 37](#_Toc199347266)

[Hình 3.20 Lịch sử đặt vé 38](#_Toc199347267)

DANH MỤC BẢNG BIỂU

[Bảng 2.1 [dbo].[Admin] 13](#_Toc199347268)

[Bảng 2.2 [dbo].[Customer] 13](#_Toc199347269)

[Bảng 2.3 [dbo].[BookingHistory] 13](#_Toc199347270)

[Bảng 2.4 [dbo].[Movie] 14](#_Toc199347271)

[Bảng 2.5 [dbo].[Genre] 14](#_Toc199347272)

[Bảng 2.6 [dbo].[Country] 14](#_Toc199347273)

[Bảng 2.7 [dbo].[Room] 15](#_Toc199347274)

[Bảng 2.8 [dbo].[Seat] 15](#_Toc199347275)

[Bảng 2.9 [dbo].[Showtime] 15](#_Toc199347276)

[Bảng 2.10 [dbo].[Ticket] 16](#_Toc199347277)

[Bảng 2.11 [dbo].[Staff] 16](#_Toc199347278)

DANH MỤC CÁC TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| **STT** | **Viết tắt** | **Cụm từ** |
| 1 | SQL | Structured Query Language |
| 2 | GUI | Graphical User Interface |
| 3 | IDE | Integrated Development Environment |
| 4 | DBMS | Database Management System |
| 5 | XML | eXtensible Markup Language |
| 6 | AWT | Abstract Window Toolkit |
| 7 | BUS | Business Logic Layer |
| 8 | DAO | Data Access Object |
| 9 | UTIL | Utilities / Helper Classes |
| 10 | CRUD | Create, Read, Update, Delete |

MỞ ĐẦU

##### Giới thiệu đề tài (Lý do chọn đề tài)

Trong bối cảnh công nghệ và mạng Internet phát triển mạnh mẽ, nhu cầu đổi mới các phương thức quản lý truyền thống ngày càng trở nên cấp thiết. Việc xây dựng một ứng dụng quản lý bán vé xem phim không chỉ là một giải pháp ngắn hạn nhằm giải quyết các vấn đề hiện tại mà còn hướng đến tạo ra một nền tảng bền vững, kết nối hàng triệu khán giả yêu thích điện ảnh với các rạp chiếu phim trên toàn quốc. Ứng dụng này cho phép người dùng dễ dàng tạo tài khoản cá nhân, theo dõi lịch sử xem phim, và tìm kiếm các suất chiếu phù hợp với thời gian rảnh rỗi hoặc sở thích chỉ với vài thao tác đơn giản trên thiết bị di động hoặc máy tính. Đồng thời, các rạp chiếu phim cũng được hưởng lợi khi có thể đăng tải thông tin lịch chiếu một cách nhanh chóng và chính xác, quản lý số lượng vé hiệu quả, theo dõi tình trạng ghế trống theo thời gian thực, đồng thời dễ tiếp cận các khách hàng tiềm năng. Xuất phát từ những thực tế trên, cùng với xu hướng số hóa ngày càng phổ biến trong ngành giải trí, em đã quyết định chọn đề tài "Thiết kế và xây dựng ứng dụng quản lý bán vé xem phim" như một dự án mang tính thực tiễn cao, góp phần nâng cao trải nghiệm của cả khách hàng và nhân viên rạp.

##### Mục tiêu đề tài

* **Phát triển ứng dụng quản lý bán vé xem phim**: Tạo ra một ứng dụng có giao diện thân thiện, dễ sử dụng, tích hợp đầy đủ các chức năng cần thiết như đặt vé, quản lý suất chiếu, và thống kê doanh thu, phục vụ hiệu quả cho cả quản lý rạp chiếu phim và khách hàng trong việc tối ưu hóa quy trình quản lý vé cũng như nâng cao trải nghiệm giải trí.
* **Cung cấp công cụ quản lý thông minh**: Xây dựng một hệ thống thông minh giúp quản lý rạp dễ dàng theo dõi thông tin chi tiết về các bộ phim đang chiếu, lịch chiếu theo ngày hoặc tuần, và doanh thu theo thời gian thực một cách khoa học và chính xác. Đồng thời, ứng dụng hỗ trợ khách hàng thực hiện các thao tác đặt vé nhanh chóng, chính xác, với khả năng chọn ghế theo ý muốn và thanh toán trực tuyến an toàn.
* **Ứng dụng kiến thức thực hành**: Sử dụng các kiến thức đã học về lập trình Java để xử lý logic, thiết kế cơ sở dữ liệu SQL Server để lưu trữ và quản lý thông tin, cùng với kỹ thuật phát triển giao diện người dùng (GUI) bằng Swing/AWT để tạo ra một ứng dụng hiện đại, tối ưu hóa trải nghiệm người dùng, đồng thời thể hiện khả năng áp dụng lý thuyết vào thực tiễn.

##### Phương pháp nghiên cứu

* **Phương pháp thu thập dữ liệu**: Thực hiện khảo sát thực tế tại một số rạp chiếu phim tiêu biểu ở Đà Nẵng, nơi có mật độ khán giả cao và nhu cầu sử dụng dịch vụ điện ảnh lớn. Quá trình này bao gồm phỏng vấn trực tiếp với nhân viên rạp để tìm hiểu các khó khăn trong quản lý vé, lịch chiếu, và doanh thu, cũng như thu thập ý kiến từ khách hàng về mong muốn khi sử dụng ứng dụng (ví dụ: tính năng chọn ghế, thanh toán nhanh, thông báo suất chiếu). Ngoài ra, còn tham khảo các tài liệu chuyên ngành về lập trình Java, quản lý cơ sở dữ liệu, và thiết kế giao diện để xây dựng nền tảng lý thuyết vững chắc, đảm bảo ứng dụng được phát triển trên cơ sở khoa học.
* **Phương pháp phân tích và xử lý dữ liệu**: Sau khi thu thập dữ liệu, em tiến hành phân tích yêu cầu một cách chi tiết dựa trên kết quả khảo sát để xác định các chức năng quan trọng và luồng hoạt động của hệ thống. Dữ liệu được xử lý để đánh giá mức độ phù hợp với nhu cầu thực tế, sau đó thực hiện kiểm thử ứng dụng với các dữ liệu mẫu (ví dụ: lịch chiếu giả lập, thông tin khách hàng) để đánh giá hiệu quả, phát hiện lỗi, và cải thiện trước khi triển khai chính thức.

##### Phạm vi nghiên cứu

* **Về nội dung**: Đề tài tập trung phát triển các tính năng chính của ứng dụng như quản lý thông tin phim (tên phim, thể loại, thời lượng), đặt vé (chọn ghế, thanh toán), quản lý suất chiếu (lịch trình, trạng thái ghế), và thống kê doanh thu (theo ngày, tuần, tháng), phục vụ một rạp chiếu phim quy mô vừa tại Đà Nẵng với khoảng 5-10 phòng chiếu.
* **Về công nghệ**: Sử dụng ngôn ngữ lập trình Java để xây dựng logic ứng dụng, SQL Server để thiết kế và quản lý cơ sở dữ liệu, cùng với Swing và AWT để phát triển giao diện người dùng. Môi trường phát triển được lựa chọn là IntelliJ, hỗ trợ tích hợp các công cụ lập trình và kiểm thử hiệu quả.
* **Về địa lý**: Ứng dụng được thiết kế ban đầu cho một rạp chiếu phim tại Đà Nẵng, nơi có điều kiện thuận lợi để thử nghiệm và thu thập phản hồi. Tuy nhiên, hệ thống có tiềm năng mở rộng sang các rạp khác trong tương lai khi bổ sung các tính năng như tích hợp đa rạp hoặc hỗ trợ thanh toán quốc tế.
* **Về thời gian**: Dự án được thực hiện từ ngày 15/03/2025 đến ngày 27/05/2025, với các mốc thời gian cụ thể như khảo sát, thiết kế, lập trình, và kiểm thử, nhằm đảm bảo hoàn thành đúng hạn và đáp ứng đầy đủ các yêu cầu đã đề ra.

##### Nội dung và kế hoạch thực hiện

* **Khảo sát, thu thập thông tin**: Tiến hành khảo sát tại các rạp chiếu phim ở Đà Nẵng, phỏng vấn nhân viên về quy trình bán vé, quản lý lịch chiếu, và các khó khăn gặp phải (như thất lạc vé, nhập liệu chậm). Đồng thời, thu thập ý kiến khách hàng về trải nghiệm đặt vé hiện tại và mong muốn cải tiến, ví dụ như giao diện trực quan hơn hoặc thông báo suất chiếu qua ứng dụng.
* **Phân tích, đánh giá các tính năng cần thiết**: Dựa trên dữ liệu khảo sát, em xác định các chức năng quan trọng như đăng nhập tài khoản, đặt vé với lựa chọn ghế, quản lý suất chiếu, và thống kê doanh thu. Sau đó, lập kế hoạch thiết kế luồng hoạt động chi tiết để đảm bảo hệ thống vận hành trơn tru.
* **Lựa chọn công nghệ và công cụ**: Quyết định sử dụng Java để lập trình logic ứng dụng, SQL Server để xây dựng cơ sở dữ liệu lưu trữ thông tin khách hàng và lịch chiếu, Swing để thiết kế giao diện người dùng với các nút bấm và bảng biểu, và IntelliJ làm môi trường phát triển tích hợp để hỗ trợ lập trình và gỡ lỗi.
* **Thiết kế giao diện người dùng và tính năng**: Thiết kế các màn hình chính như giao diện đăng nhập (tên người dùng, mật khẩu), đặt vé (bản đồ ghế ngồi, thông tin suất chiếu), và thống kê doanh thu (biểu đồ doanh thu), đảm bảo giao diện trực quan, dễ sử dụng, và phù hợp với cả nhân viên quản lý và khách hàng.
* **Xây dựng và phát triển ứng dụng**: Lập trình các chức năng chính như đăng nhập (xác thực tài khoản), đặt vé (lưu thông tin ghế, thanh toán giả lập), và thống kê doanh thu (tổng hợp dữ liệu từ cơ sở dữ liệu), đồng thời tích hợp với SQL Server để lưu trữ và truy xuất dữ liệu.
* **Kiểm thử, đánh giá và hoàn thiện**: Thử nghiệm ứng dụng với dữ liệu mẫu như lịch chiếu 10 phim trong một tuần, kiểm tra các tình huống lỗi (ví dụ: ghế đã bán, lỗi thanh toán), sửa lỗi, cải thiện hiệu suất, và hoàn thiện báo cáo, hoàn thành vào ngày 27/05/2025.

##### Bố cục báo cáo

* **Chương 1. Cơ sở lý thuyết**: Trình bày các khái niệm cơ bản về ngôn ngữ lập trình Java, cách thiết kế và quản lý cơ sở dữ liệu bằng SQL Server, cũng như vai trò của IntellJi trong phát triển ứng dụng.
* **Chương 2. Phân tích và thiết kế hệ thống**: Bao gồm phân tích yêu cầu của thiết kế giao diện, phân tích nhu cầu của quản trị viên và khách hàng, cùng với phân tích chi tiết về hệ thống quản lý bán vé. Mô tả kiến trúc tổng thể, cấu trúc cơ sở dữ liệu, lưu đồ thuật toán/hoạt động quản lý bán vé xem phim
* **Chương 3. Triển khai xây dựng giao diện và các chức năng**: Chi tiết quá trình lập trình, tích hợp các thành phần chức năng và triển khai ứng dụng.
* **Cuối cùng là Kết luận và hướng phát triển** (đề xuất mở rộng tính năng)
* **Tài liệu tham khảo** (danh sách tài liệu và nguồn tham khảo)

# CHƯƠNG 1: CƠ SỞ LÝ THUYẾT

## Java

### **Khái niệm**

* Java là một ngôn ngữ lập trình hướng đối tượng, phổ biến và được sử dụng rộng rãi trên nhiều nền tảng.
* Java được phát triển bởi Sun Microsystems (nay là một phần của Oracle) vào năm 1995.
  + 1. Lịch sử phát triển
* Java được thiết kế để có thể chạy trên bất kỳ thiết bị nào (write once, run anywhere) thông qua Java Virtual Machine (JVM).
* Các phiên bản Java được cập nhật thường xuyên, cải thiện tính năng, hiệu suất và bảo mật.
* Java trở thành một trong những ngôn ngữ lập trình phổ biến nhất trên thế giới, được sử dụng rộng rãi trong các ứng dụng web, mobile, game, Big Data và nhiều lĩnh vực khác.
  + 1. Vai trò
* Java được sử dụng rộng rãi trong phát triển ứng dụng web, mobile, desktop, game, Big Data và các hệ thống phân tán.
* Java cung cấp một nền tảng lập trình mạnh mẽ, an toàn và có tính di động cao.
* Java được ứng dụng trong các ngành công nghiệp như tài chính, y tế, giáo dục, viễn thông và nhiều lĩnh vực khác.
  + 1. Đặc điểm
* Java là ngôn ngữ hướng đối tượng, có cú pháp đơn giản và dễ đọc.
* Java có thư viện phong phú, hỗ trợ đa nền tảng và an toàn bảo mật.
* Java sử dụng Java Virtual Machine (JVM) để thực thi mã, giúp ứng dụng chạy độc lập với nền tảng phần cứng.
* Java có cơ chế quản lý bộ nhớ tự động (Garbage Collection) và hỗ trợ đa luồng.

## SQL Server

* + 1. Khái niệm
* SQL Server là hệ quản trị cơ sở dữ liệu (DBMS) do Microsoft phát triển.
* SQL Server sử dụng ngôn ngữ truy vấn cơ sở dữ liệu SQL (Structured Query Language) để tương tác với dữ liệu
  + 1. Lịch sử phát triển
* SQL Server được sử dụng rộng rãi trong các ứng dụng doanh nghiệp, quản lý cơ sở dữ liệu lớn, phân tích dữ liệu và báo cáo.
* SQL Server cung cấp các tính năng nâng cao như phân tích dữ liệu thời gian thực, Machine Learning, BI và an ninh dữ liệu.
* SQL Server được tích hợp chặt chẽ với các sản phẩm khác của Microsoft như Windows Server, Visual Studio và Office.
  + 1. Vai trò
* SQL Server được sử dụng rộng rãi trong các ứng dụng doanh nghiệp, quản lý cơ sở dữ liệu lớn, phân tích dữ liệu và báo cáo.
* SQL Server cung cấp các tính năng nâng cao như phân tích dữ liệu thời gian thực, Machine Learning, BI và an ninh dữ liệu.
* SQL Server được tích hợp chặt chẽ với các sản phẩm khác của Microsoft như Windows Server, Visual Studio và Office.
  + 1. Đặc điểm
* SQL Server sử dụng ngôn ngữ SQL để thao tác với dữ liệu, như tạo, truy vấn, cập nhật và xóa dữ liệu.
* SQL Server hỗ trợ nhiều kiểu dữ liệu như số, văn bản, ngày tháng, thời gian, nhị phân, v.v.
* SQL Server cung cấp các tính năng nâng cao như bảo mật, tính sẵn sàng cao, khả năng mở rộng và tích hợp với các công nghệ khác.
* SQL Server có thể chạy trên các nền tảng như Windows, Linux và Docker.

## IntelliJ IDEA

* + 1. Khái niệm
* IntelliJ IDEA là một môi trường phát triển tích hợp (IDE) mạnh mẽ, được sử dụng rộng rãi để phát triển phần mềm, đặc biệt là các ứng dụng Java.
* Được phát triển bởi JetBrains, IntelliJ IDEA ra mắt lần đầu vào năm 2001 và nhanh chóng trở thành một trong những IDE hàng đầu cho các lập trình viên.
  + 1. Lịch sử phát triển
* IntelliJ IDEA ban đầu được thiết kế tập trung cho lập trình Java, nhưng hiện đã hỗ trợ nhiều ngôn ngữ khác như Kotlin, Python, JavaScript, TypeScript, Go, v.v.
* JetBrains liên tục cập nhật IntelliJ IDEA với các phiên bản mới, bổ sung tính năng, cải thiện hiệu suất và sửa lỗi. Phiên bản mới nhất là IntelliJ IDEA 2024.2, được phát hành vào tháng 8 năm 2024.
* IntelliJ IDEA có hai phiên bản chính: Community (miễn phí, mã nguồn mở) và Ultimate (trả phí, cung cấp các tính năng nâng cao).
  + 1. Vai trò
* IntelliJ IDEA là một IDE hàng đầu, đặc biệt phổ biến trong phát triển ứng dụng Java, Android và các ứng dụng doanh nghiệp.
* Cung cấp một nền tảng mạnh mẽ, linh hoạt để xây dựng, triển khai và quản lý các dự án phần mềm với hiệu suất cao.
* Tích hợp sâu với các công cụ và framework như Maven, Gradle, Git, Docker, Kubernetes, Spring, và nhiều công cụ khác, giúp tăng năng suất lập trình.
  + 1. Đặc điểm
* IntelliJ IDEA sở hữu giao diện người dùng trực quan, hiện đại, với khả năng tùy chỉnh cao để phù hợp với nhu cầu của lập trình viên.
* Cung cấp các tính năng nổi bật như trình soạn thảo mã thông minh (code completion), trình gỡ lỗi mạnh mẽ, công cụ refactoring, phân tích mã theo thời gian thực, và tích hợp với hệ thống kiểm soát phiên bản.
* Hỗ trợ đa nền tảng, chạy mượt mà trên Windows, macOS và Linux.
* Được tối ưu hóa để tăng tốc độ phát triển với các tính năng như gợi ý mã, phát hiện lỗi nhanh và tích hợp AI (thông qua plugin hoặc tính năng AI Assistant trong phiên bản Ultimate).

# PHÂN TÍCH VÀ THIẾT KẾ HẾ THỐNG

## Yêu cầu người dùng

***Đơn giản và dễ sử dụng :***

* Giao diện người dùng cần được thiết kế đơn giản và dễ sử dụng để đảm bảo người dùng có thể dễ dàng tìm hiểu và tương tác với phần mềm. Điều này có thể đạt được thông qua việc sử dụng một cấu trúc giao diện rõ ràng, kèm theo biểu đồ và hình ảnh minh họa nhằm hỗ trợ người dùng hiểu rõ hơn về cách sử dụng phần mềm.
* Người dùng nên có thể dễ dàng tìm hiểu và thực hiện các tùy chọn trong phần mềm mà không gặp bất kỳ khó khăn nào trong quá trình sử dụng.
* Giao diện cũng cần đảm bảo sự rõ ràng trong việc hiển thị thông tin cũng như trong các thao tác tương tác, giúp người dùng thao tác nhanh chóng và chính xác.

***Thiết kế giao diện hấp dẫn:***

* Giao diện người dùng cần có thiết kế hấp dẫn để thu hút sự chú ý của người dùng và tạo cảm giác hứng thú khi sử dụng phần mềm.
* Việc sử dụng màu sắc phù hợp, hình ảnh chất lượng cao và chỉnh sửa hình ảnh một cách hợp lý là những yếu tố quan trọng giúp xây dựng nên một giao diện trực quan và bắt mắt.
* Ngoài ra, cần chú trọng đến cách bố trí và trình bày các phần tử trong giao diện. Việc sắp xếp các thành phần như nút bấm, hình ảnh và văn bản cần được thực hiện một cách hài hòa, cân đối và có thẩm mỹ để tạo nên trải nghiệm người dùng thân thiện và chuyên nghiệp.

***Tốc độ tải các chức năng nhanh:***

* Tốc độ chuyển qua lại giữa các chức năng là một yếu tố quan trọng trong trải nghiệm người dùng. Giao diện người dùng cần được tối ưu hóa để đảm bảo rằng các chức năng hoạt động nhanh chóng và không gây chồng chéo các giao diện lên nhau trong quá trình sử dụng. Điều này có thể đạt được thông qua việc:
* Tối ưu hóa hình ảnh để giảm thời gian tải,
* Tổ chức bố cục hợp lý để tránh dư thừa tài nguyên hệ thống,
* Hạn chế các hiệu ứng không cần thiết gây ảnh hưởng đến hiệu suất phần mềm.
* Việc đảm bảo tốc độ phản hồi nhanh sẽ giúp phần mềm vận hành mượt mà, tăng mức độ hài lòng và hiệu quả làm việc cho người dùng.

***Tính nhất quán và thống nhất:***

* Giao diện người dùng cần có tính nhất quán và thống nhất trong cách hiển thị các phần tử và các thành phần giao diện khác nhau trong phần mềm. Các yếu tố thiết kế như màu sắc, phông chữ, nút và biểu tượng nên được sử dụng một cách nhất quán để tạo ra một trải nghiệm đồng nhất trên toàn bộ ứng dụng. Điều này giúp người dùng dễ dàng nhận diện và sử dụng các thành phần giao diện một cách nhất quán.

***Tính năng tương tác:***

* Giao diện người dùng cần cung cấp các tính năng tương tác để tạo sự thú vị và hấp dẫn cho người dùng trong quá trình sử dụng phần mềm.
* Các phần tử như: Nút bấm, thanh trượt, vùng cuộn, tương tác bằng bàn phím,... có thể được tích hợp để cho phép người dùng dễ dàng thao tác và tương tác với các tính năng của phần mềm. Việc tăng cường tính tương tác sẽ giúp cải thiện trải nghiệm người dùng, tăng hiệu quả sử dụng phần mềm và tạo cảm giác thân thiện.

***Sử dụng hình ảnh và đồ họa chất lượng cao:***

* Hình ảnh và đồ họa chất lượng cao có thể tạo nên một giao diện hấp dẫn và thu hút sự chú ý của người dùng. Việc sử dụng hình ảnh đẹp, có ý nghĩa và liên quan đến nội dung sẽ giúp truyền đạt thông tin một cách sinh động, đồng thời tạo ra một trải nghiệm thú vị cho người dùng.
* Tuy nhiên, điều quan trọng là phải sử dụng những hình ảnh:
* Có bản quyền rõ ràng,
* Hoặc được cung cấp theo các giấy phép hợp lệ như Creative Commons, giấy phép thương mại,…
* Tuyệt đối không sử dụng hình ảnh vi phạm bản quyền, vì điều này có thể dẫn đến các vấn đề pháp lý nghiêm trọng, bao gồm khiếu nại, gỡ bỏ sản phẩm, hoặc bị xử lý theo quy định pháp luật.

## Phân Tích Nghiệp Vụ Quản Lý

Trước khi xây dựng chương trình quản lý bán vé xem phim, việc phân tích nghiệp vụ là quan trọng để xác định rõ các chức năng cần thiết. Dưới đây là kết quả phân tích nghiệp vụ:

* + 1. Đối với ADMIN

Lớp AdminFrame là giao diện đồ họa chính dành cho quản trị viên (admin) trong hệ thống quản lý bán vé xem phim, được xây dựng bằng Java Swing. Lớp này cung cấp một môi trường trực quan, thân thiện và hiệu quả, cho phép admin quản lý toàn diện các hoạt động liên quan đến phim, phòng chiếu, suất chiếu, nhân viên, khách hàng, và thống kê doanh thu. AdminFrame đóng vai trò như một trung tâm điều khiển, tích hợp các chức năng nhập liệu, hiển thị dữ liệu, cập nhật thời gian thực, và cung cấp quyền truy cập đến các lớp xử lý logic nghiệp vụ. Lớp này đảm bảo admin có thể giám sát, vận hành, và tối ưu hóa hoạt động của rạp chiếu phim một cách dễ dàng, chính xác và an toàn.

**Chức năng chính**:

* **Hiển thị giao diện tổng quan**: Cung cấp một giao diện chính với menu điều hướng (sidebar) và khu vực nội dung động, cho phép admin truy cập nhanh các chức năng như quản lý phim, phòng chiếu, suất chiếu, nhân viên, khách hàng, và thống kê doanh thu.
* **Quản lý thông tin phim**: Cho phép admin thêm, sửa, xóa, và xem chi tiết thông tin phim, bao gồm tiêu đề, mô tả, thời lượng, đạo diễn, thể loại, poster, ngày bắt đầu, ngày kết thúc, năm sản xuất, quốc gia, và giới hạn tuổi.
* **Quản lý phòng chiếu**: Hỗ trợ thêm, sửa, xóa thông tin phòng chiếu, với các thuộc tính như tên phòng, sức chứa, và giá vé.
* **Quản lý suất chiếu**: Cho phép tạo suất chiếu mới, gán phim và nhân viên vào suất chiếu, cập nhật trạng thái suất chiếu (Sắp công chiếu, Đang chiếu, Đã chiếu xong), và ẩn suất chiếu khỏi giao diện người dùng.
* **Quản lý nhân viên**: Cung cấp chức năng thêm, sửa, xóa thông tin nhân viên, bao gồm tên và email, để quản lý đội ngũ nhân sự phụ trách các suất chiếu.
* **Quản lý khách hàng**: Hiển thị danh sách khách hàng với thông tin như tên, email, và số vé đã đặt, giúp theo dõi hoạt động đặt vé.
* **Thống kê doanh thu**: Hiển thị biểu đồ doanh thu theo ngày sử dụng JFreeChart, tự động cập nhật định kỳ, với tùy chọn tải lại dữ liệu thủ công để admin nắm bắt tình hình tài chính.
* **Cập nhật thời gian thực**: Tích hợp đồng hồ thời gian thực hiển thị ngày giờ hiện tại và cơ chế tự động cập nhật trạng thái suất chiếu mỗi 30-60 giây.
* **Tìm kiếm và lọc dữ liệu**: Hỗ trợ tìm kiếm phim hoặc suất chiếu theo tiêu chí như tên phim, ngày chiếu, hoặc phòng chiếu, giúp admin truy xuất thông tin nhanh chóng.
* **Quản lý tài khoản admin**: Cho phép admin đăng xuất an toàn, quay về màn hình đăng nhập (LoginFrame), và đảm bảo dừng các tác vụ nền để bảo vệ hệ thống.
* **Xử lý lỗi và xác nhận hành động**: Hiển thị thông báo lỗi qua JOptionPane khi xảy ra ngoại lệ (ví dụ: lỗi cơ sở dữ liệu) và yêu cầu xác nhận trước khi xóa dữ liệu để tránh thao tác nhầm.
  + 1. Đối với Người Dùng

Sau khi đăng nhập vào hệ thống, người dùng sẽ truy cập vào lớp:

* Lớp Giao diện Đăng ký Khách hàng (RegisterFrame)

Mục đích: Lớp RegisterFrame là giao diện đồ họa cho phép người dùng mới đăng ký tài khoản khách hàng trong hệ thống quản lý bán vé xem phim, được xây dựng bằng Java Swing. Lớp này cung cấp môi trường thân thiện để nhập thông tin cá nhân (họ tên, mật khẩu, email) và tạo tài khoản, giúp người dùng trở thành khách hàng chính thức.

* Lớp Giao diện Người dùng (UserFrame)

Mục đích: Lớp UserFrame là giao diện đồ họa chính dành cho khách hàng đã đăng nhập vào hệ thống quản lý bán vé xem phim, được xây dựng bằng Java Swing. Lớp này cung cấp môi trường trực quan để khách hàng xem danh sách phim, kiểm tra lịch sử đặt vé, và đặt vé cho các suất chiếu mong muốn, đóng vai trò trung tâm điều khiển cho khách hàng.

* Lớp Giao diện Đặt vé (BookingFrame)

Mục đích: Lớp BookingFrame là giao diện đồ họa cho phép khách hàng đặt vé xem phim trong hệ thống quản lý bán vé xem phim, được xây dựng bằng Java Swing. Lớp này cung cấp môi trường tương tác để chọn ghế ngồi, xem thông tin suất chiếu, tính toán chi phí, và chuyển sang thanh toán.

* Lớp Giao diện Thanh toán (PaymentFrame)

Mục đích: Lớp PaymentFrame là giao diện đồ họa cho phép khách hàng hoàn tất thanh toán vé xem phim trong hệ thống quản lý bán vé xem phim, được xây dựng bằng Java Swing. Lớp này cung cấp môi trường an toàn để xác nhận thông tin đặt vé, thực hiện thanh toán, và xuất hóa đơn dưới dạng XML

## Mô tả và thiết kế hệ thống

* + 1. Kiến trúc tổng thể

Hệ thống được thiết kế theo mô hình 3 lớp (Three-layer architecture) để đảm bảo tính module hóa, dễ bảo trì và mở rộng. Cấu trúc bao gồm 3 tầng chính: Presentation Layer (GUI), Business Logic Layer (BUS), và Data Access Layer (DAO), cùng với các thành phần bổ trợ từ các thư mục khác. Dưới đây là mô tả chi tiết:

Tầng giao diện người dùng chịu trách nhiệm hiển thị và tương tác trực tiếp với người dùng. Các lớp giao diện chính bao gồm:

* AdminFrame: Giao diện quản trị viên, hỗ trợ quản lý phim, lịch chiếu, phòng, thống kê doanh thu,...
* UserFrame: Giao diện khách hàng, cung cấp chức năng xem phim, đặt vé, lịch sử đặt vé,...
* BookingFrame: Giao diện đặt vé, hiển thị suất chiếu, ghế ngồi và xử lý chọn ghế.
* PaymentFrame: Giao diện thanh toán, hỗ trợ xác nhận và hoàn tất giao dịch.
* LoginFrame: Giao diện đăng nhập cho admin và khách hàng.
* RegisterFrame: Giao diện đăng ký tài khoản khách hàng.

Tầng này tập trung vào trải nghiệm người dùng (UI/UX), đảm bảo giao diện trực quan và

Tầng logic nghiệp vụ xử lý các quy tắc và quy trình, làm cầu nối giữa giao diện và cơ sở dữ liệu. Các lớp BUS bao gồm:

* AdminBUS: Quản lý quyền hạn và chức năng của admin.
* BookingHistoryBUS: Xử lý lịch sử đặt vé.
* CustomerBUS: Quản lý thông tin khách hàng (cập nhật, xác thực,...).
* MovieBUS: Xử lý nghiệp vụ liên quan đến phim (thêm, sửa, xóa,...).
* RevenueBUS: Tính toán và thống kê doanh thu.
* RoomBUS: Quản lý phòng chiếu (thêm phòng, kiểm tra sức chứa,...).
* SeatBUS: Xử lý trạng thái ghế (đặt, kiểm tra,...).
* ShowtimeBUS: Quản lý lịch chiếu (tạo, kiểm tra trạng thái,...).
* StaffBUS: Quản lý thông tin nhân viên.
* TicketBUS: Quản lý vé (tạo, hủy,...).

Tầng này đảm bảo các logic nghiệp vụ được thực thi chính xác, ví dụ: kiểm tra ghế trống trước khi đặt vé hoặc tính toán doanh thu theo thời gian thực.

Tầng truy cập dữ liệu thực hiện các thao tác đọc/ghi (CRUD) với cơ sở dữ liệu. Các lớp DAO bao gồm:

* BookingHistoryDAO: Quản lý dữ liệu lịch sử đặt vé.
* CustomerDAO: Quản lý dữ liệu khách hàng.
* MovieDAO: Truy vấn và cập nhật dữ liệu phim.
* RevenueDAO: Truy vấn dữ liệu để thống kê doanh thu.
* RoomDAO: Quản lý dữ liệu phòng chiếu.
* SeatDAO: Quản lý dữ liệu ghế ngồi.
* ShowtimeDAO: Quản lý dữ liệu suất chiếu.
* StaffDAO: Quản lý dữ liệu nhân viên.
* TicketDAO: Quản lý dữ liệu vé.

Tầng này tách biệt logic nghiệp vụ và truy cập dữ liệu, giúp dễ dàng thay đổi cơ sở dữ liệu khi cần.

Tầng mô hình (Model) định nghĩa các thực thể (Entities) đại diện cho dữ liệu trong hệ thống, bao gồm:

* BookingHistory: Lưu trữ thông tin lịch sử đặt vé.
* Customer: Lưu trữ thông tin khách hàng.
* Movie: Lưu trữ thông tin phim.
* Revenue: Lưu trữ dữ liệu doanh thu.
* Room: Lưu trữ thông tin phòng chiếu.
* Seat: Lưu trữ thông tin ghế ngồi.
* Showtime: Lưu trữ thông tin suất chiếu.
* Staff: Lưu trữ thông tin nhân viên.
* Ticket: Lưu trữ thông tin vé.

Tầng này cung cấp cấu trúc dữ liệu cho các lớp DAO và BUS.

Tầng mạng hỗ trợ kết nối và xử lý thời gian thực, bao gồm:

* DataUpdater: Cập nhật dữ liệu theo thời gian thực.
* SocketClient: Xử lý kết nối client với server.
* SocketServer: Quản lý server socket để hỗ trợ giao tiếp đa luồng.
* ThreadManager: Quản lý luồng để đảm bảo hiệu suất xử lý đồng thời.

Tầng này hữu ích cho các tính năng như thông báo suất chiếu hoặc cập nhật trạng thái ghế theo thời gian thực.

Tầng tiện ích cung cấp các công cụ hỗ trợ hệ thống, bao gồm:

* DBConnection: Quản lý kết nối với cơ sở dữ liệu.
* GenerateAdminPassword: Tạo mật khẩu ngẫu nhiên cho admin.
* PasswordEncrypter: Mã hóa mật khẩu để bảo mật.
* Main: Điểm khởi chạy chương trình, khởi tạo toàn bộ hệ thống.

Tầng này hỗ trợ các chức năng phụ trợ như kết nối CSDL, bảo mật, và khởi động ứng dụng.

* Người dùng tương tác qua tầng GUI (ví dụ: đặt vé qua BookingFrame).
* Tầng GUI gọi tầng BUS (ví dụ: TicketBUS để xử lý logic đặt vé).
* Tầng BUS sử dụng tầng DAO (ví dụ: TicketDAO) để lưu dữ liệu vào CSDL.
* Dữ liệu từ Model được sử dụng để định nghĩa các thực thể.
* Tầng Network hỗ trợ cập nhật thời gian thực (nếu cần).
* Tầng UTIL đảm bảo kết nối CSDL và bảo mật.
* Kết quả được trả ngược lại từ DAO → BUS → GUI để hiển thị.
  + 1. Mô tả và Thiết kế cơ sở dữ liệu

Dựa trên phân tích nghiệp vụ của hệ thống quản lý bán vé xem phim đã được thiết kế cơ sở dữ liệu (CSDL) để lưu trữ thông tin liên quan. Dưới đây là mô tả chi tiết về cấu trúc của cơ sở dữ liệu, bao gồm các bảng,các thuộc tính và các trường trong bảng cũng như mối quan hệ giữa chúng và mục đích sử dụng.

**Tất cả các bảng của Database Movie Ticket Booking trong SQL Server:**

Bảng tài khoản:

Bảng . [dbo].[Admin]

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type |  |
| AdminID(FK) | Int | not null |
| Username | Varchar(50) | not null |
| Password | Varchar(255) | not null |
| FullName | Varchar(100) |  |
| Email | Varchar(100) |  |

Bảng này dùng để lưu trữ và quản lý thông tin tài khoản của các quản trị viên (Admin) trong hệ thống. Nó bao gồm các thông tin như AdminID (định danh duy nhất), Username, Password, FullName và Email. Tác dụng chính là hỗ trợ đăng nhập, xác thực và phân quyền cho Admin, giúp họ thực hiện các nhiệm vụ quản lý như kiểm soát hệ thống, quản lý phim, lịch chiếu, hoặc xử lý các vấn đề phát sinh. Bảng này đảm bảo rằng chỉ những người có quyền mới có thể truy cập vào các chức năng nhạy cảm.

Bảng . [dbo].[Customer]

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type |  |
| CustomerID(FK) | Int | not null |
| Username | Varchar(50) | not null |
| Password | Varchar(100) | not null |
| FullName | Varchar(100) |  |
| Email | Varchar(100) |  |

Bảng này quản lý thông tin tài khoản của khách hàng, bao gồm CustomerID (định danh duy nhất), Username, Password, FullName và Email. Tác dụng chính là hỗ trợ khách hàng đăng nhập, lưu trữ thông tin cá nhân để cá nhân hóa trải nghiệm, và quản lý các giao dịch như đặt vé hoặc thanh toán.

Bảng quản lí lịch sử đặt vé:

Bảng . [dbo].[BookingHistory]

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type |  |
| HistoryID(FK) | Int | Not null |
| CustomerID | Int | Not null |
| TicketID | Int | Not null |
| BookingDate | Datetime | Not null |
| MovieTitle | Nvarchar(100) | Not null |
| RoomName | Nvarchar(100) | Not null |
| SeatNumber | Nvarchar(100) | Not null |
| Price | Decimal(10, 2) | Not null |

Bảng này lưu trữ lịch sử các giao dịch đặt vé của khách hàng, bao gồm thông tin chi tiết như ID giao dịch, thông tin khách hàng, vé đã đặt, thời gian đặt, và trạng thái giao dịch. Tác dụng của nó là giúp theo dõi toàn bộ lịch sử đặt vé, hỗ trợ việc tra cứu, phân tích hành vi khách hàng, và cung cấp dữ liệu để cải thiện dịch vụ. Ngoài ra, bảng này còn hữu ích trong việc xử lý tranh chấp hoặc hoàn tiền nếu có vấn đề xảy ra.

Bảng quản lý nội dung và danh mục:

Bảng . [dbo].[Movie]

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type |  |
| MovieID(FK) | Int | Not null |
| Title | Nvarchar(100) | Not null |
| Description | Nvarchar(MAX) |  |
| Poster | Nvarchar(200) |  |
| Duration | Int |  |
| Director | Nvarchar(100) |  |
| GenreID | Int |  |
| StarDate | Date |  |
| EndDate | Date |  |
| ProductionYear | Int |  |
| CountryID | Int |  |
| AgeRestriction | Int |  |

Bảng này chứa thông tin chi tiết về các bộ phim, như mã phim, tên phim, đạo diễn, năm sản xuất, thời lượng, thể loại, mô tả,... . Tác dụng chính là cung cấp dữ liệu để hiển thị danh sách phim trên giao diện khách hàng, hỗ trợ tìm kiếm và lọc phim, đồng thời liên kết với các bảng khác như Showtime để quản lý lịch chiếu. Bảng này cũng giúp quản trị viên dễ dàng thêm, sửa, xóa thông tin phim trong hệ thống.

Bảng . [dbo].[Genre]

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type |  |
| GenreID(FK) | Int | Not null |
| GenreName | Nvarchar(50) | Not null |

Bảng này lưu trữ thông tin về các thể loại phim (ví dụ: hành động, hài, kinh dị), bao gồm mã thể loại và tên thể loại. Tác dụng của nó là phân loại phim theo thể loại, giúp khách hàng dễ dàng tìm kiếm hoặc lọc phim theo sở thích. Ngoài ra, bảng này hỗ trợ hệ thống gợi ý phim phù hợp với khách hàng, phân tích xu hướng yêu thích của người dùng, và hỗ trợ quản trị viên trong việc quản lý danh mục phim.

Bảng . [dbo].[Country]

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type |  |
| CountryID(FK) | Int | Not null |
| CountryName | Nvarchar(50) | Not null |

Bảng này lưu trữ thông tin về các quốc gia, có thể bao gồm mã quốc gia, tên quốc gia. Giúp thêm thông tin của phim một cách nhanh chóng

Bảng quản lý cơ sở vật chất:

Bảng . [dbo].[Room]

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type |  |
| RoomID(FK) | Int |  |
| RoomName | Nvarchar |  |
| Capacity | Int |  |
| Price | Decimal (10, 2) | Not null |

Bảng này quản lý thông tin về các phòng chiếu trong rạp, bao gồm mã phòng, tên phòng, sức chứa, hoặc trạng thái phòng. Tác dụng của nó là hỗ trợ hệ thống sắp xếp lịch chiếu phim, quản lý không gian chiếu, và đảm bảo rằng các phòng được sử dụng hiệu quả. Bảng này cũng liên kết với bảng Seat để quản lý ghế ngồi và bảng Showtime để kiểm tra tình trạng phòng theo thời gian.

Bảng . [dbo].[Seat]

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type |  |
| SeatID(FK) | Int |  |
| RoomID | Int |  |
| SeatNumber | Nvarchar(10) |  |
| Status | Nvarchar(20) | Not null |

Bảng này lưu trữ thông tin về ghế ngồi trong các phòng chiếu, bao gồm mã ghế, số ghế, trạng thái (đã đặt hay còn trống), và liên kết với phòng chiếu. Tác dụng chính là hỗ trợ khách hàng chọn ghế khi đặt vé, giúp hệ thống quản lý tình trạng ghế theo từng suất chiếu, và đảm bảo không có trường hợp trùng lặp khi đặt ghế. Bảng này cũng hỗ trợ hiển thị sơ đồ ghế trên giao diện đặt vé.

Bảng quản lý lịch chiếu và vé:

Bảng . [dbo].[Showtime]

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type |  |
| ShowtimeID(FK) | Int | Not null |
| MovieID | Int |  |
| RoomID | Int | Not null |
| ShowDate | Date | Not null |
| StaffID | Int |  |
| Status | Nvarchar(50) |  |
| IsVisible | Bit |  |

Bảng này lưu trữ thông tin về lịch chiếu phim, bao gồm mã suất chiếu, mã phim, mã phòng, thời gian bắt đầu, thời gian kết thúc, và trạng thái suất chiếu. Tác dụng của nó là cung cấp lịch trình chiếu phim cho khách hàng, hỗ trợ đặt vé theo thời gian, và giúp quản trị viên quản lý lịch chiếu một cách hiệu quả. Bảng này cũng liên kết với các bảng như Movie và Room để đảm bảo thông tin nhất quán.

Bảng . [dbo].[Ticket]

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type |  |
| TicketID(FK) | Int | Not null |
| CustomerID | Int | Not null |
| ShowtimeID | Int | Not null |
| Price | Decimal (10, 2) | Not null |
| SeatID | Int | Not null |
| SeatNumber | Nvarchar(50) | Not null |

Bảng này lưu trữ thông tin về vé đã bán, bao gồm mã vé, mã suất chiếu, mã ghế, mã khách hàng, giá vé, và trạng thái thanh toán. Tác dụng chính là quản lý các giao dịch bán vé, cung cấp thông tin để in vé hoặc gửi vé điện tử cho khách hàng, và hỗ trợ theo dõi doanh thu.

Bảng quản lý nhân sự:

Bảng . [dbo].[Staff]

|  |  |  |
| --- | --- | --- |
| Column Name | Data Type |  |
| StaffID | Int | Not null |
| StaffName | Nvarchar(100) |  |
| Email | Nvarchar(100) |  |

Bảng này quản lý thông tin về nhân viên trong rạp chiếu hoặc hệ thống, bao gồm mã nhân viên, tên, thông tin liên lạc, và có thể là tài khoản đăng nhập sau này phát triển chức năng cho nhân viên. Tác dụng của nó là hỗ trợ quản lý nhân sự, phân công công việc, và theo dõi hiệu suất làm việc.

**Mối quan hệ giữa các bảng:**

* Mối quan hệ giữa các bảng được xác định thông qua các khóa chính khóa ngoại, đảm bảo tính nhất quán và an toàn dữ liệu.

**Mục Đích Sử Dụng CSDL:**

* Cơ sở dữ liệu được thiết kế để lưu trữ thông tin chi tiết về phim, khách hàng, suất chiếu mà khách hàng đã đặt vé,số tiền mà khách hàng cần trả,...
* Giúp quản lý và theo dõi vấn đề.
* Cung cấp khả năng truy vấn dữ liệu dễ dàng để tạo báo cáo và thống kê.
* Đảm bảo tính nhất quán và hiệu suất của hệ thống thông qua mối quan hệ giữa các bảng.
  + 1. Lưu đồ thuật toán/hoạt động cho bài toán quản lí bán vé xem phim
       1. Đăng nhập vào hệ thống
* Hệ thống khởi động với giao diện đăng nhập được thiết kế thông qua lớp Login, cung cấp trải nghiệm người dùng trực quan và bảo mật. Người dùng được phân loại thành hai vai trò chính: Admin và Khách hàng, với các yêu cầu đăng nhập riêng biệt để đảm bảo quyền truy cập phù hợp.
* Đối với Admin:  
  Hệ thống yêu cầu nhập chính xác tên đăng nhập và mật khẩu đã được cấp trước đó trong cơ sở dữ liệu. Admin không được phép tạo tài khoản mới nhằm duy trì tính bảo mật và kiểm soát chặt chẽ. Nếu thông tin đăng nhập chính xác, Admin được chuyển đến giao diện quản lý hệ thống, nơi họ có thể thực hiện các tác vụ quản lý doanh thu và chi phí. Nếu thông tin sai, hệ thống hiển thị thông báo lỗi, yêu cầu nhập lại thông tin đăng nhập.
* Đối với Khách hàng:  
  Khách hàng có thể đăng nhập bằng tài khoản đã tạo hoặc chọn đăng ký tài khoản mới nếu chưa có. Trong trường hợp đăng ký, khách hàng cần cung cấp thông tin cá nhân bao gồm họ tên, địa chỉ email và mật khẩu. Hệ thống kiểm tra tính hợp lệ của thông tin (ví dụ: định dạng email, độ dài mật khẩu) trước khi tạo tài khoản và lưu vào cơ sở dữ liệu. Nếu đăng nhập thất bại, hệ thống hiển thị thông báo lỗi và cung cấp tùy chọn nhập lại thông tin hoặc chuyển sang giao diện đăng ký. Sau khi đăng nhập thành công, khách hàng được chuyển đến trang quản lý chính để thực hiện các tác vụ liên quan đến mua vé.
  + - 1. Hệ thống đối với khách hàng
* Trang quản lý chính của khách hàng  
  Khi khách hàng đăng nhập, hệ thống kiểm tra trạng thái tài khoản để xác định đây là lần đăng nhập đầu tiên hay đã có thông tin trước đó.  
  Nếu là lần đầu đăng nhập, tài khoản sẽ không có thông tin cá nhân hoặc mã khách hàng (ID). Hệ thống tự động chuyển khách hàng đến giao diện đăng ký thông tin cá nhân, nơi họ cần nhập đầy đủ các thông tin bắt buộc như họ tên, số điện thoại, địa chỉ email, và các thông tin tùy chọn như ngày sinh hoặc địa chỉ. Sau khi khách hàng nhấn nút đăng ký, hệ thống xác nhận tính hợp lệ của dữ liệu (ví dụ: không để trống các trường bắt buộc, kiểm tra định dạng email), sau đó tạo mã khách hàng duy nhất và lưu thông tin vào cơ sở dữ liệu. Mã khách hàng này sẽ được sử dụng để liên kết với các giao dịch mua vé sau này. Khách hàng sau đó được chuyển đến trang quản lý chính.  
  Nếu khách hàng đã đăng nhập trước đó, hệ thống hiển thị thông tin cá nhân đã lưu (họ tên, mã khách hàng, email) và cung cấp các tùy chọn chính: đăng ký vé xem phim hoặc theo dõi vé đã đăng ký. Giao diện này được thiết kế để dễ sử dụng, với các nút điều hướng rõ ràng và thông tin được trình bày trực quan, giúp khách hàng nhanh chóng truy cập các chức năng cần thiết.
* Trang đăng ký vé xem phim  
  Khách hàng truy cập giao diện đăng ký vé, nơi hệ thống hiển thị danh sách các phim đang chiếu, bao gồm thông tin chi tiết như tên phim, thể loại, thời lượng, và các suất chiếu khả dụng. Mỗi suất chiếu đi kèm với thông tin về giờ chiếu, phòng chiếu, và số lượng ghế còn trống.  
  Khách hàng thực hiện các bước sau để đăng ký vé:  
  Chọn phim từ danh sách, sau đó chọn suất chiếu phù hợp với lịch trình của mình.  
  Hệ thống hiển thị sơ đồ ghế ngồi của phòng chiếu đã chọn, trong đó các ghế được đánh dấu rõ ràng: ghế trống, ghế đã đặt, và ghế đang được giữ tạm thời. Khách hàng chọn số lượng ghế và vị trí ghế mong muốn.  
  Hệ thống kiểm tra trạng thái ghế:  
  Nếu ghế còn trống, hệ thống tạm khóa ghế để tránh xung đột đặt chỗ từ người dùng khác và chuyển sang bước xác nhận.  
  Nếu ghế đã được đặt, hệ thống thông báo và yêu cầu khách hàng chọn lại ghế khác.  
  Sau khi chọn ghế thành công, khách hàng xem lại thông tin vé (phim, suất chiếu, ghế, tổng giá vé) và cam kết thanh toán bằng cách nhấn nút xác nhận. Hệ thống lưu thông tin vé vào cơ sở dữ liệu, liên kết với mã khách hàng, và chuyển sang bước thanh toán hoặc theo dõi vé.
* Trang theo dõi vé  
  Giao diện theo dõi vé cho phép khách hàng xem danh sách tất cả các vé đã đăng ký, bao gồm thông tin chi tiết như tên phim, suất chiếu, số ghế, và giá vé. Giao diện được thiết kế trực quan, với các bộ lọc để khách hàng dễ dàng tìm kiếm vé theo ngày hoặc trạng thái.  
  Nếu thanh toán thành công, hệ thống cập nhật trạng thái vé, gửi vé điện tử qua email hoặc SMS (bao gồm mã QR hoặc mã vé để kiểm tra tại rạp), và cho phép in vé tại quầy nếu cần(sẽ phát triển sau này, hiện tại chỉ xuất file xml vào trong thư mục của dự án).  
  Nếu sau 3 lần thử thanh toán online mà vẫn chưa thanh toán hết, hệ thống thông báo yêu cầu khách hàng đến quầy gặp Admin để thanh toán trực tiếp. Thông tin nợ được ghi nhận và liên kết với mã khách hàng để Admin xử lý.  
  Khách hàng cũng có thể hủy vé (nếu trong thời gian cho phép) hoặc yêu cầu hỗ trợ từ Admin thông qua giao diện này.
  + - 1. Hệ thống đối với Admin
* Quản lý doanh thu bán vé  
  Admin truy cập giao diện quản lý sau khi đăng nhập thành công. Giao diện này cung cấp các công cụ để ghi nhận và theo dõi thông tin bán vé, bao gồm:  
  Thông tin khách hàng (tên, mã khách hàng, email).  
  Chi tiết vé (mã vé, phim, suất chiếu, ghế, số tiền, trạng thái thanh toán: đã thanh toán.  
  Admin có các quyền quản lý sau:  
  Thêm, sửa, hoặc xóa thông tin khách hàng và vé trong cơ sở dữ liệu, với các biện pháp xác nhận để tránh thao tác sai sót.  
  Hệ thống tự động tính toán tổng doanh thu từ bán vé dựa trên các giao dịch đã hoàn tất, cung cấp báo cáo chi tiết theo ngày. Báo cáo này bao gồm số lượng vé bán ra, tổng số tiền giúp Admin dễ dàng theo dõi tình hình tài chính của rạp.
* Quản lý chi phí vận hành  
  Doanh thu từ bán vé được sử dụng để chi trả các chi phí vận hành rạp phim, bao gồm:  
  + Chi phí bản quyền phim và vận hành thiết bị chiếu (máy chiếu, hệ thống âm thanh).  
  + Lương nhân viên, bao gồm quản lý, nhân viên bán vé, nhân viên kỹ thuật, và nhân viên vệ sinh.  
  + Chi phí bảo trì cơ sở vật chất (sửa chữa ghế, hệ thống điều hòa, ánh sáng).  
  + Các chi phí khác như điện, nước, internet, và vật liệu vệ sinh.  
  Admin ghi nhận chi tiết từng khoản chi, bao gồm số tiền, ngày chi, loại chi phí, và mục đích sử dụng.   
  Hệ thống tự động tính toán tổng chi phí vận hành theo từng loại chi phí và cung cấp báo cáo tổng hợp. Báo cáo này hỗ trợ Admin đánh giá hiệu quả tài chính, xác định các khoản chi lớn, và lập kế hoạch ngân sách cho các hoạt động tương lai.
  + - 1. Lưu ý
* Hệ thống được thiết kế với các biện pháp bảo mật mạnh mẽ, sử dụng mã hóa dữ liệu để bảo vệ thông tin cá nhân của khách hàng và giao dịch thanh toán.
* Trạng thái ghế được cập nhật theo thời gian thực thông qua cơ chế đồng bộ hóa cơ sở dữ liệu, đảm bảo không xảy ra xung đột khi nhiều khách hàng đặt vé cùng lúc.
* Các báo cáo doanh thu và chi phí được lưu trữ lâu dài, hỗ trợ phân tích xu hướng và lập kế hoạch kinh doanh.
* Hệ thống cung cấp giao diện thân thiện với người dùng, với các thông báo lỗi rõ ràng và hướng dẫn chi tiết để hỗ trợ cả khách hàng và Admin trong quá trình sử dụng.

## Thư viện, API của Java được sử dụng

* pom.xml:

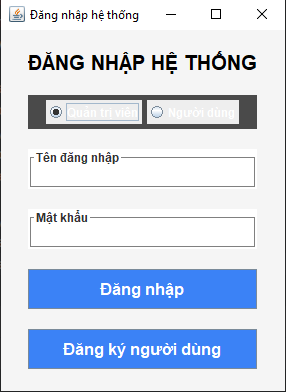
<project xmlns="http://maven.apache.org/POM/4.0.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">  
 <modelVersion>4.0.0</modelVersion>  
 <groupId>com.movie</groupId>  
 <artifactId>MovieTicketBooking</artifactId>  
 <version>1.0-SNAPSHOT</version>  
 <properties>  
 <maven.compiler.source>17</maven.compiler.source>  
 <maven.compiler.target>17</maven.compiler.target>  
 </properties>  
 <dependencies>  
 <!-- SQL Server JDBC Driver -->  
 <dependency>  
 <groupId>com.microsoft.sqlserver</groupId>  
 <artifactId>mssql-jdbc</artifactId>  
 <version>12.8.1.jre11</version>  
 </dependency>  
 <dependency>  
 <groupId>org.jfree</groupId>  
 <artifactId>jfreechart</artifactId>  
 <version>1.5.3</version>  
 </dependency>  
 <!-- For Swing -->  
 <dependency>  
 <groupId>org.openjfx</groupId>  
 <artifactId>javafx-controls</artifactId>  
 <version>17</version>  
 </dependency>  
 <!-- For encryption (BCrypt) -->  
 <dependency>  
 <groupId>org.mindrot</groupId>  
 <artifactId>jbcrypt</artifactId>  
 <version>0.4</version>  
 </dependency>  
 </dependencies>  
 <build>  
 <plugins>  
 <!-- Compiler Plugin -->  
 <plugin>  
 <groupId>org.apache.maven.plugins</groupId>  
 <artifactId>maven-compiler-plugin</artifactId>  
 <version>3.11.0</version>  
 <configuration>  
 <source>17</source>  
 <target>17</target>  
 </configuration>  
 </plugin>  
 <!-- Exec Plugin -->  
 <plugin>  
 <groupId>org.codehaus.mojo</groupId>  
 <artifactId>exec-maven-plugin</artifactId>  
 <version>3.1.0</version>  
 <configuration>  
 <mainClass>com.movie.MainClass</mainClass>  
 </configuration>  
 </plugin>  
 </plugins>  
 </build>  
</project>

# TRIỂN KHAI XÂY DỰNG GIAO DIỆN VÀ CHỨC NĂNG

## Giao diện đăng nhập và chức năng

* + 1. Đăng nhập

Khi chạy chương trình ta sẽ có giao diện đăng nhập như sau:



Hình . Giao diện đăng nhập

Đầu tiên là đăng nhập vào hệ thống qua giao diện login lớp Login.

Giao diện phân quyền giữa Quản trị viên và Người dùng(nút Đăng ký người dùng)

Đối với người dùng là Admin thì bắt buộc phải đúng yêu cầu về mật khẩu và tên đăng nhập, không được tạo tài khoản mới và phải thông qua class GenerateAdminPassword:

public class GenerateAdminPassword {  
 public static void main(String[] args) {  
 String password = "123456789";  
 String hashedPassword = PasswordEncrypter.*hashPassword*(password);   
 System.*out*.println("Mật khẩu đã mã hóa cho admin: " + hashedPassword);  
 }  
}

Từ đây ta sẽ có được mật khẩu đã mã hóa và Insert các thông tin vào bảng Admin trong Database

Quản trị viên (Admin) và Người dùng (Customer). Sử dụng Java Swing để xây dựng giao diện đăng nhập và triển khai cơ chế phân quyền dựa trên lựa chọn của người dùng (Admin hoặc Customer) và xác thực thông tin đăng nhập.

**Mục đích**: Hệ thống phân quyền để đảm bảo rằng:

* Quản trị viên (Admin) có quyền truy cập vào giao diện quản lý (AdminFrame).
* Người dùng (Customer) có quyền truy cập vào giao diện người dùng (UserFrame).

**Cách triển khai**:

* Giao diện đăng nhập cho phép người dùng chọn vai trò (Admin hoặc Customer) thông qua JradioButton:
* JRadioButton adminRadio = new JRadioButton("Quản trị viên");  
  JRadioButton userRadio = new JRadioButton("Người dùng");  
  adminRadio.setForeground(Color.*WHITE*); // Chữ trắng  
  userRadio.setForeground(Color.*WHITE*); // Chữ trắng  
  ButtonGroup group = new ButtonGroup();  
  group.add(adminRadio);  
  group.add(userRadio);  
  adminRadio.setSelected(true); // Mặc định chọn Quản trị viên
* Dựa trên vai trò được chọn, hệ thống sử dụng các lớp xử lý nghiệp vụ khác nhau (AdminBUS hoặc CustomerBUS) để xác thực thông tin đăng nhập.
* Sau khi xác thực thành công, hệ thống chuyển hướng đến giao diện tương ứng với vai trò.

Hàm authenticate:

private boolean authenticate(String username, String password) {  
 if (isAdminLogin) {  
 return adminBUS.validateAdmin(username, password);  
 } else {  
 return customerBUS.validateUserPlain(username, password);  
 }  
}

* Hàm này kiểm tra thông tin đăng nhập dựa trên vai trò được chọn (isAdminLogin).
* Nếu là Admin (isAdminLogin = true):

Gọi adminBUS.validateAdmin(username, password) để xác thực:

public boolean validateAdmin(String username, String password) {  
 String query = "SELECT Password FROM Admin WHERE Username = ?";  
 try (Connection conn = DBConnection.*getConnection*();  
 PreparedStatement stmt = conn.prepareStatement(query)) {  
 stmt.setString(1, username);  
 ResultSet rs = stmt.executeQuery();  
 if (rs.next()) {  
 String hashedPassword = rs.getString("Password");  
 return PasswordEncrypter.*checkPassword*(password, hashedPassword);  
 }  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 return false;  
}

* Nếu là Customer (isAdminLogin = false):

Gọi customerBUS.validateUserPlain(username, password) để xác thực:

public boolean validateUser(String username, String password) {  
 try {  
 // Lấy thông tin khách hàng từ cơ sở dữ liệu  
 Customer customer = customerDAO.getCustomerByUsername(username);  
  
 // Kiểm tra khách hàng có tồn tại không  
 if (customer != null) {  
 // So sánh mật khẩu người dùng nhập với mật khẩu đã mã hóa trong DB  
 return PasswordEncrypter.*checkPassword*(password, customer.getPassword());  
 }  
 } catch (SQLException e) {  
 e.printStackTrace();  
 }  
 return false;  
}

**Xử lý nút "Đăng nhập"**:

* Khi nhấn nút "Đăng nhập", mã lấy thông tin từ usernameField và passwordField, kiểm tra vai trò qua adminRadio.isSelected() để gán giá trị cho isAdminLogin.
* Nếu xác thực thành công (authenticate trả về true):
* Kết nối đến server thông qua SocketClient (địa chỉ localhost:5000).
* Gửi thông điệp đăng nhập ("Login: " + username).



Hình . Đăng nhập thành công

* Chuyển hướng giao diện:

Nếu là Admin: Mở AdminFrame.

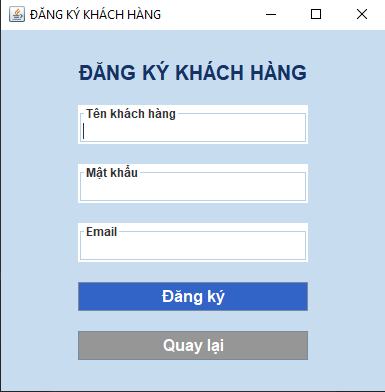
Nếu là Customer: Lấy customerId từ customerBUS.getCustomerIdByUsername và mở UserFrame với customerId.

* Đóng khung đăng nhập (dispose()).
* Nếu xác thực thất bại hoặc không kết nối được server, hiển thị thông báo lỗi bằng JOptionPane.

loginButton.addActionListener(e -> {  
 String username = usernameField.getText().trim();  
 String password = new String(passwordField.getPassword()).trim();  
 isAdminLogin = adminRadio.isSelected();  
 if (authenticate(username, password)) {  
 SocketClient client = new SocketClient("localhost", 5000);  
 client.start();  
 int attempts = 20;  
 while (attempts > 0 && !client.isConnected()) {  
 try {  
 Thread.*sleep*(100);  
 } catch (InterruptedException ie) {  
 ie.printStackTrace();  
 }  
 attempts--;  
 }  
 if (client.isConnected()) {  
 client.sendMessage("Login: " + username);  
 } else {  
 JOptionPane.*showMessageDialog*(null, "Không thể kết nối đến server!", "Lỗi", JOptionPane.*ERROR\_MESSAGE*);  
 return;  
 }

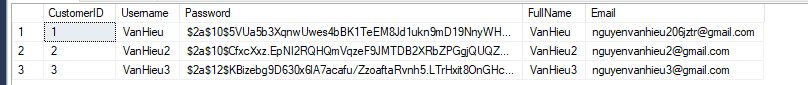
### Đăng Ký (Người dùng)

Khi ấn Đăng ký trong giao diện Đăng nhập cửa sổ sẽ hiện ra:



Hình . Giao diện Đăng Ký

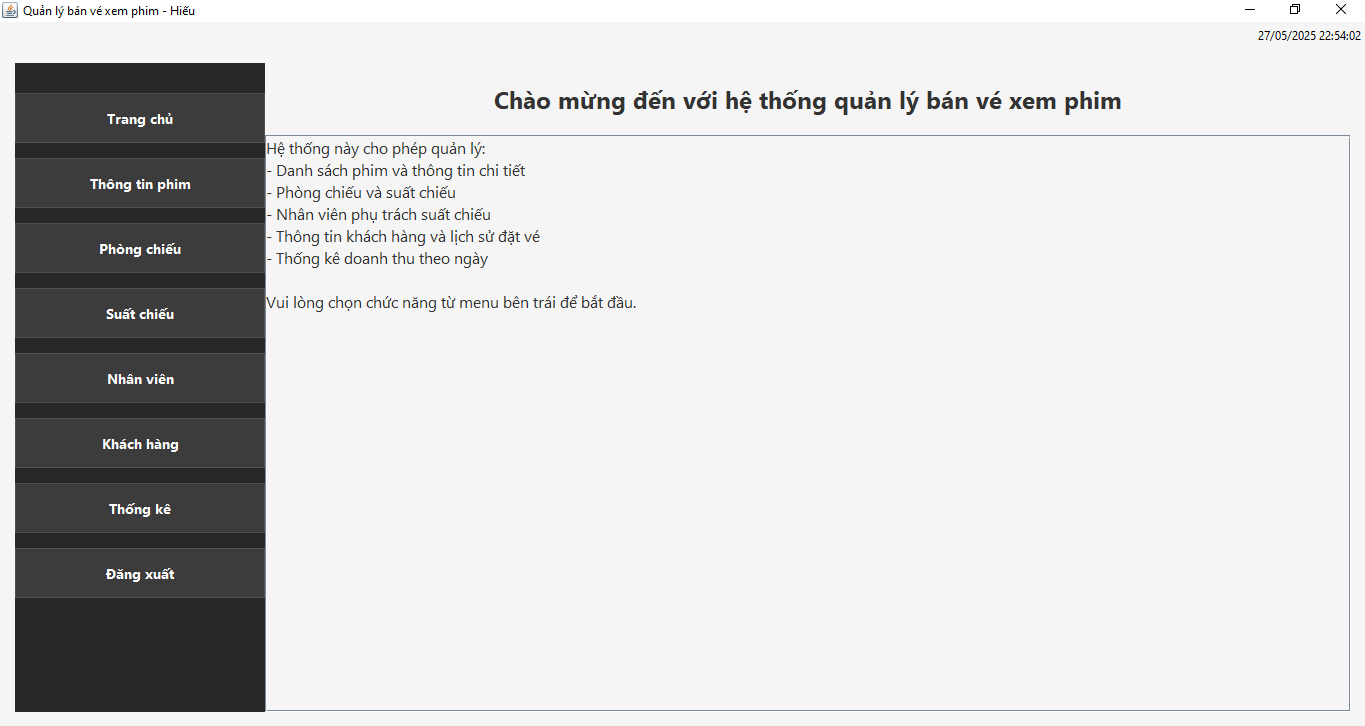
Mỗi lúc khách đăng ký thì sẽ lưu thông tin khách hàng vào CSDL với mật khẩu thì sử dụng thư viện JBCrypt để mã hóa một chiều đảm bảo tính bảo mật và an toàn cho khách hàng



Hình . Tài khoản được mã hóa lưu vào CSDL

## Giao diện quản lý Admin

* + 1. Trang chủ



Hình . Trang chủ

* Ở trang chủ, hướng dẫn sử dụng và giới thiệu về các chức năng của giao diện Admin.
* **Thanh bên (Sidebar)**:

Được tạo trong phương thức createMainView():

* Các nút như "Trang chủ", "Thông tin phim", v.v., được tạo và gán ActionListener để chuyển đổi panel thông qua showPanel(String panelName).
* Mỗi nút được tùy chỉnh giao diện trong styleButton(JButton button), bao gồm màu nền, viền, hiệu ứng hover.
* Ở góc trên bên phải của giao diện sẽ hiển thị logo của quán cùng với ngày, tháng, năm và giờ hiện tại. Việc này giúp người dùng dễ dàng theo dõi thời gian và quản lý công việc hiệu quả hơn, đồng thời hỗ trợ trong việc kiểm tra, đối chiếu thông tin theo thời gian thực một cách thuận tiện.

Lập trình để xuất hiện thời gian:

private void startClock() {  
 timeLabel = new JLabel();  
 timeLabel.setFont(new Font("Segoe UI", Font.*PLAIN*, 12));  
 timeLabel.setForeground(Color.*BLACK*);  
 JPanel timePanel = new JPanel(new FlowLayout(FlowLayout.*RIGHT*));  
 timePanel.add(timeLabel);  
 add(timePanel, BorderLayout.*NORTH*);  
  
 new Thread(() -> {  
 SimpleDateFormat sdf = new SimpleDateFormat("dd/MM/yyyy HH:mm:ss");  
 while (running) {  
 timeLabel.setText(sdf.format(new java.util.Date()));  
 try {  
 Thread.*sleep*(1000);  
 } catch (InterruptedException e) {  
 System.*err*.println("Clock thread interrupted: " + e.getMessage());  
 }  
 }  
 }).start();  
}

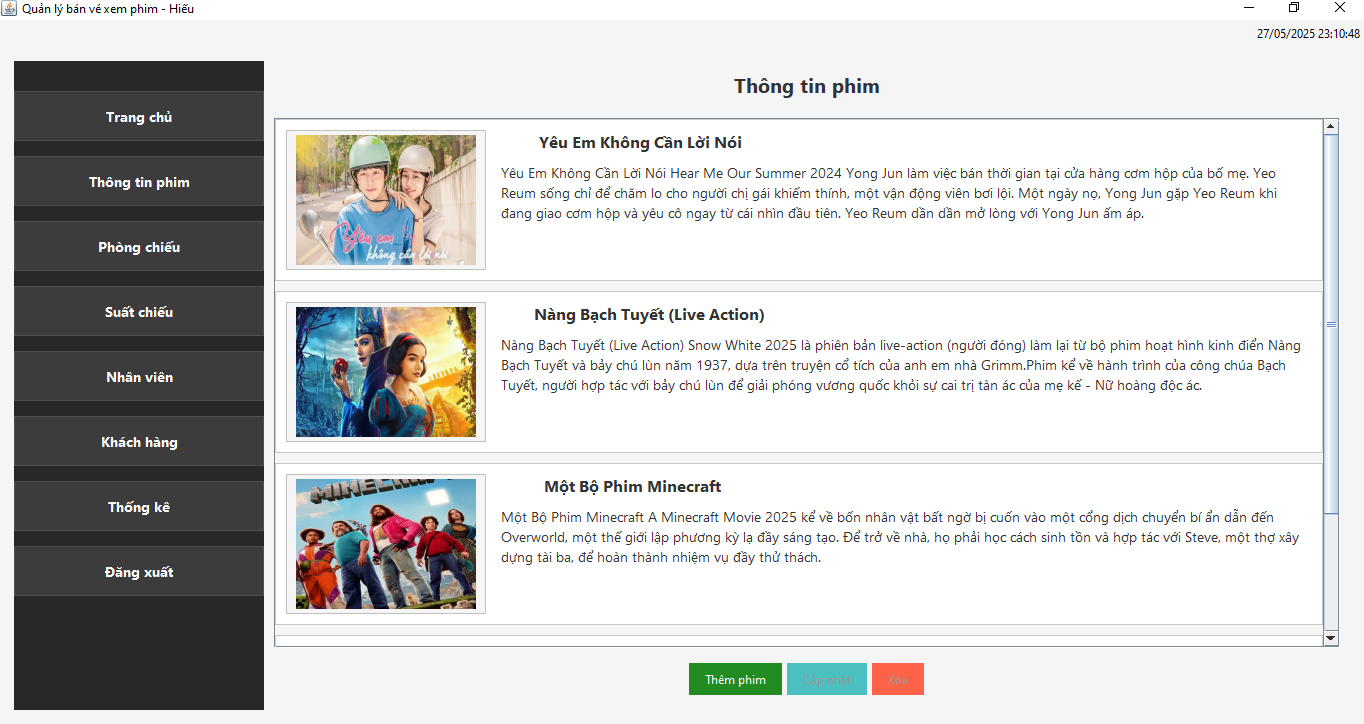
Tạo một luồng riêng(Thread) cập nhật liên tục để xử lý việc cập nhật thời gian liên tục.

Trong khi biến running là true, sẽ liên tục:

* Lấy thời gian hiện tại (new java.util.Date()).
* Format thời gian rồi gán vào timeLabel.

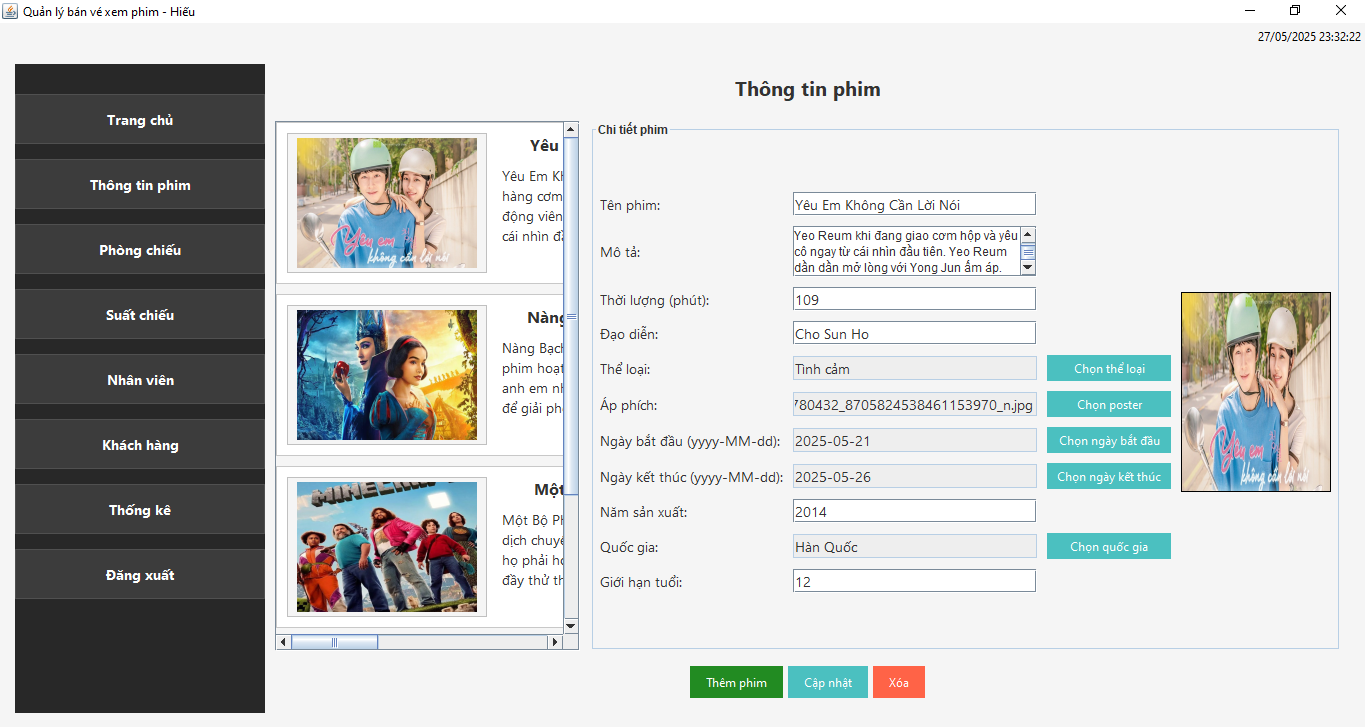
Sau mỗi lần cập nhật, thread ngủ 1 giây (Thread.sleep(1000)) rồi tiếp tục cập nhật.

* + 1. Thông tin phim



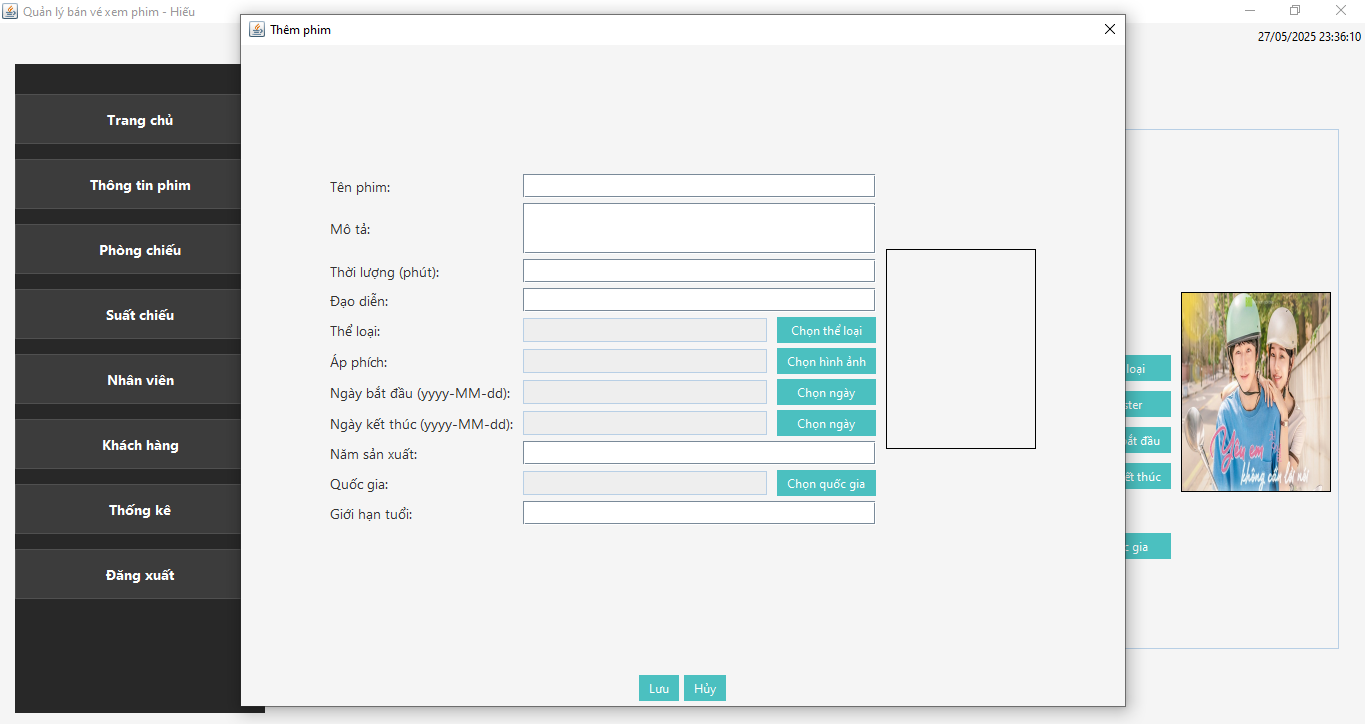
Hình . Thông tin phim

* Hiển thị danh sách phim:
* Danh sách phim được tải từ cơ sở dữ liệu và hiển thị dưới dạng các panel, mỗi panel đại diện cho một bộ phim.
* Mỗi panel hiển thị hình poster (nếu có) và mô tả phim.
* **Quản lý phim**:
* **Thêm phim**: Nút "Thêm phim" mở một hộp thoại để nhập thông tin phim mới.
* **Cập nhật phim**: Nút "Cập nhật" (hiện đang tắt) cho phép chỉnh sửa thông tin phim đã chọn.
* **Xóa phim**: Nút "Xóa" (hiện đang tắt) cho phép xóa phim đã chọn.



Hình . Chức năng Cập nhật

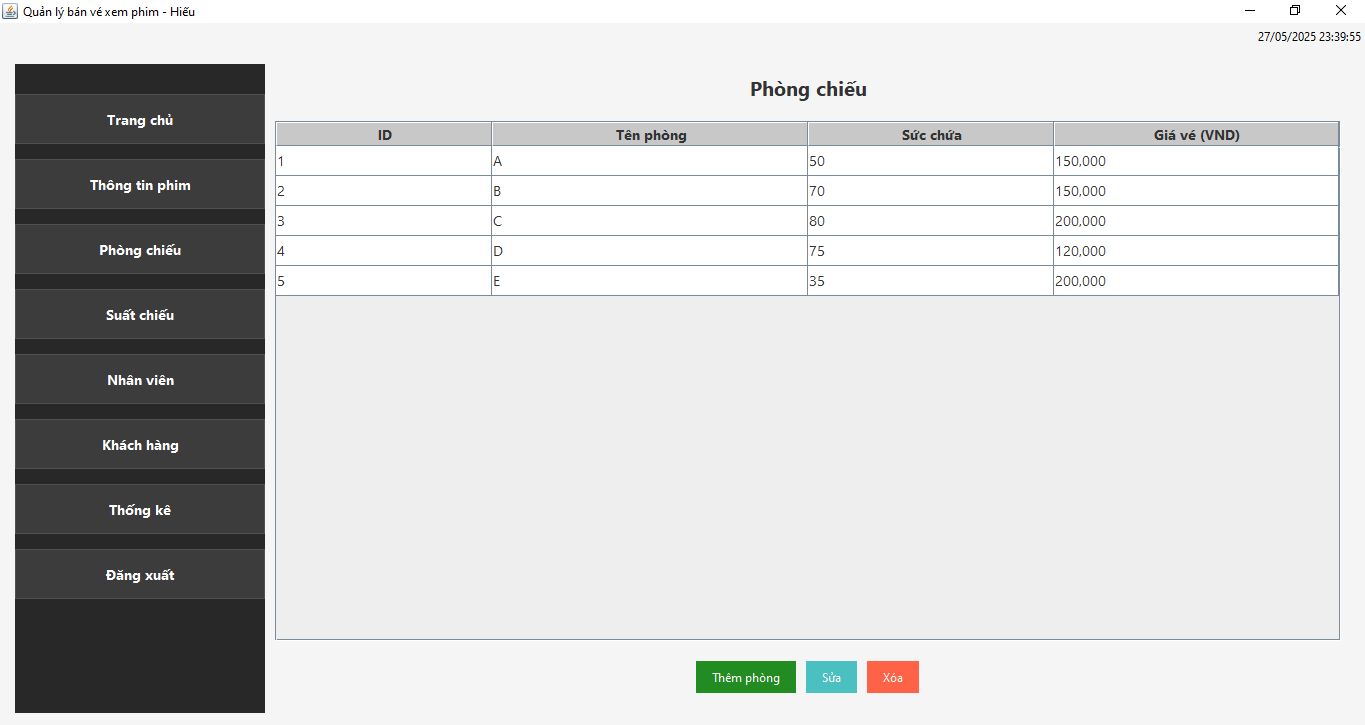
Nút "Cập nhật" và "Xóa" ban đầu bị tắt (setEnabled(false)), chỉ kích hoạt khi người dùng chọn một phim (xử lý trong MouseListener của createMoviePanel).



Hình . Chức năng Thêm Phim

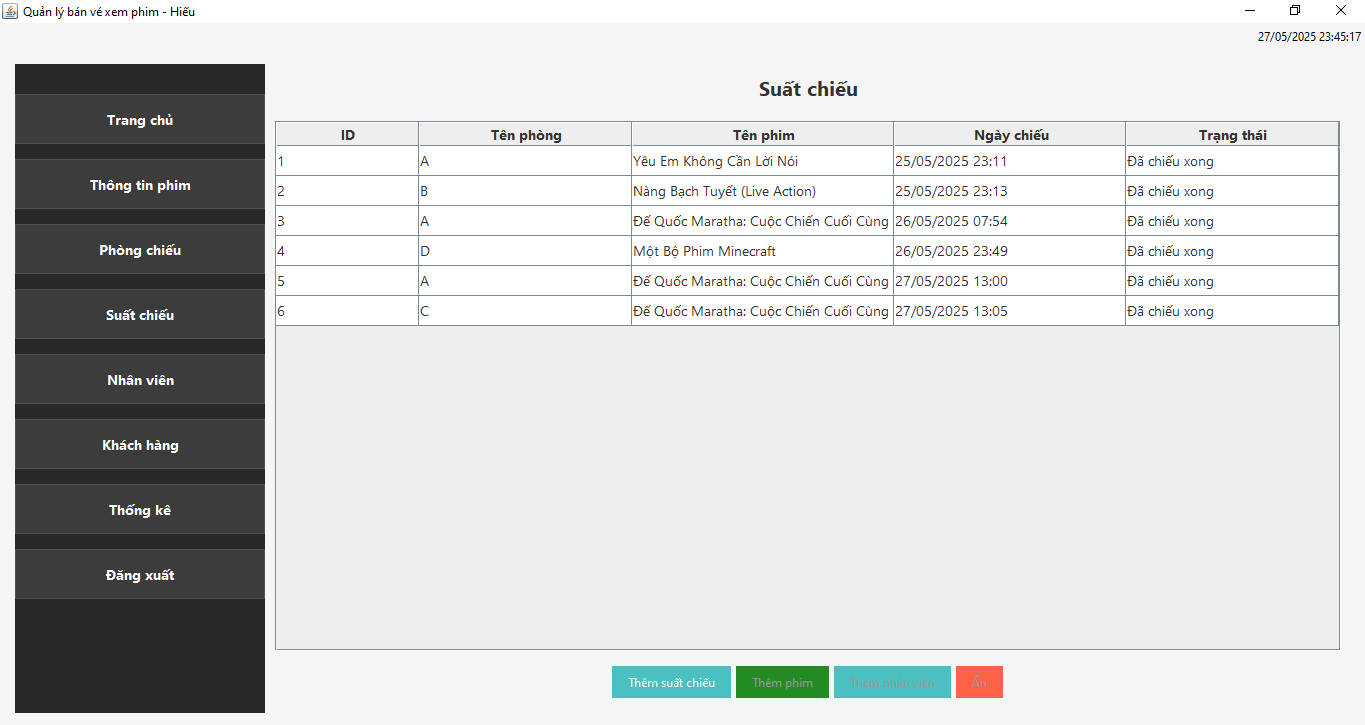
Dùng để thêm phim, các thông tin của Phim sẽ đi vào CSDL sau đó được load ra panel Thông tin phim

* + 1. Phòng chiếu

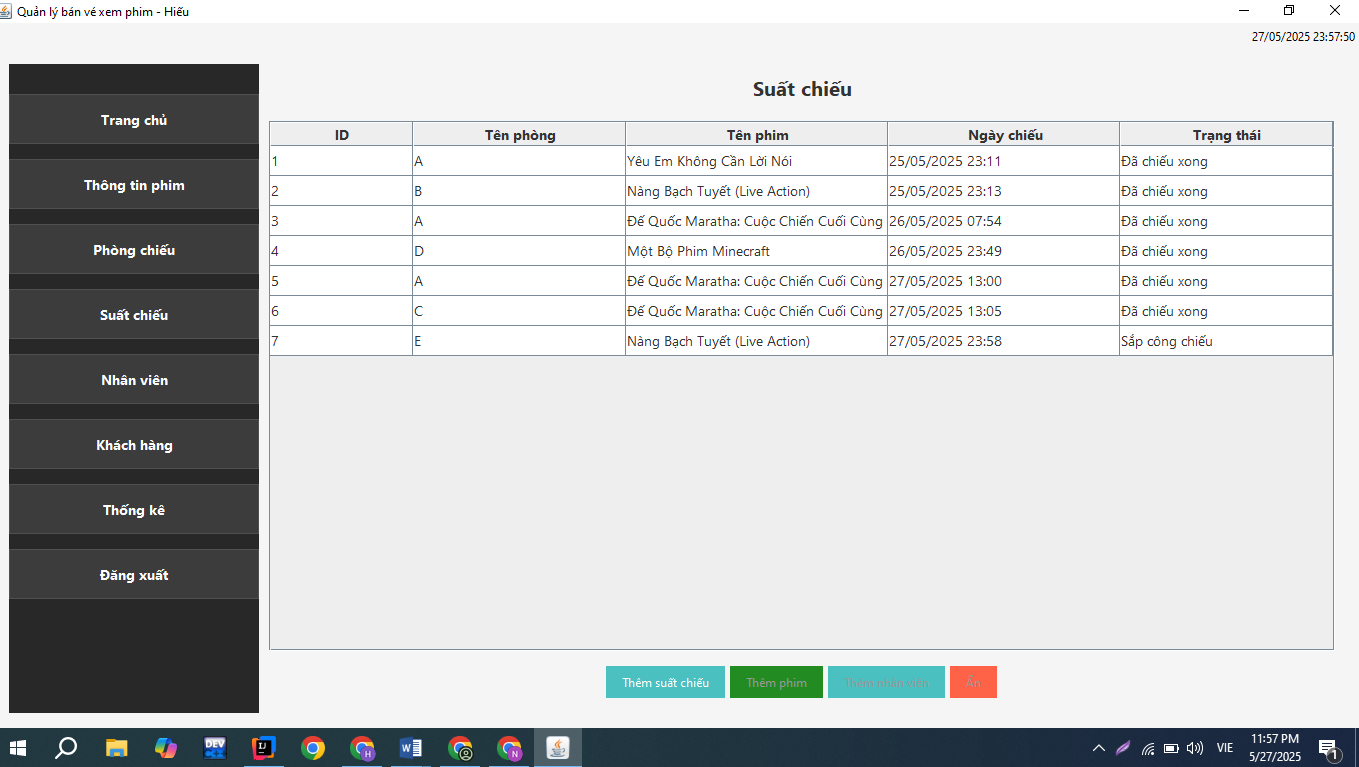


Phần này Quản lý có thể dễ dàng thêm xóa sửa phòng, đặt giá mặc định cho các phòng để có thể thêm các phim hot “hay” không “hot” vào những phòng đó

* + 1. Suất chiếu

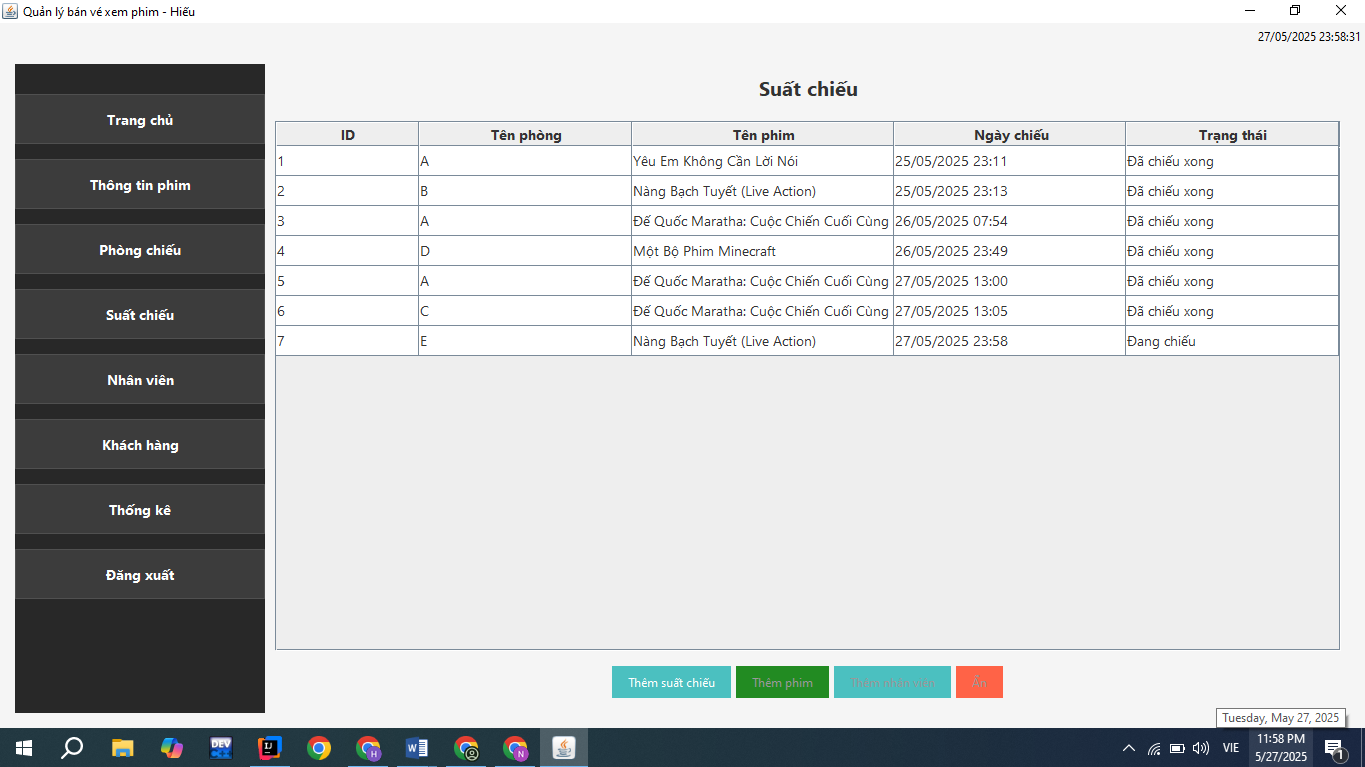


Hình . Suất chiếu

* **Quản lý suất chiếu**:
* **Thêm suất chiếu**: Tạo mới một suất chiếu với thông tin phòng chiếu và thời gian chiếu.
* **Thêm phim vào suất chiếu**: Gán một bộ phim cho suất chiếu đã chọn.
* **Thêm nhân viên vào suất chiếu**: Gán nhân viên phụ trách cho suất chiếu.
* **Ẩn suất chiếu**: Ẩn suất chiếu khỏi giao diện người dùng bằng cách thay đổi trường isVisible từ 0 (hiển thị) thành 1 (ẩn) trong cơ sở dữ liệu.
* **Hiển thị danh sách suất chiếu**: Hiển thị danh sách suất chiếu với thông tin như ID, tên phòng, tên phim, ngày giờ chiếu, và trạng thái.
* **Cập nhật trạng thái suất chiếu**: Tự động cập nhật trạng thái của suất chiếu dựa trên thời gian hiện tại và thời lượng phim, bao gồm các trạng thái: "Không chiếu", "Sắp công chiếu", "Đang chiếu", và "Đã chiếu xong".
* 

Hình 3.10 Khởi tạo suất chiếu

* **Hiển thị danh sách suất chiếu**:
* Trong phương thức createShowtimePanel, một bảng (JTable) được tạo để hiển thị danh sách suất chiếu với các cột: ID, Tên phòng, Tên phim, Ngày chiếu, và Trạng thái.
* Dữ liệu được tải thông qua phương thức loadShowtimes, sử dụng SwingWorker để truy xuất danh sách suất chiếu từ showtimeBUS.getAllShowtimes() và cập nhật bảng giao diện.
* Bảng được làm mới mỗi 30 giây thông qua Timer để đảm bảo thông tin trạng thái luôn cập nhật
* **Nút chức năng**:
* **Thêm suất chiếu**: Mở dialog (showAddShowtimeDialog) để chọn phòng và nhập thời gian chiếu. Suất chiếu mới được tạo với trạng thái mặc định là "Không chiếu" và isVisible = 0.
* **Thêm phim**: Mở dialog (showAddMovieToShowtimeDialog) để gán phim cho suất chiếu đã chọn, cập nhật movieID và trạng thái thành "Sắp công chiếu" nếu hợp lệ.
* **Thêm nhân viên**: Mở dialog (showAddStaffToShowtimeDialog) để gán nhân viên phụ trách, cập nhật staffID trong suất chiếu.
* **Ẩn suất chiếu**: Nút "Ẩn" chỉ được kích hoạt khi suất chiếu có trạng thái "Đang chiếu" hoặc "Đã chiếu xong". Khi nhấn, phương thức showtimeBUS.updateShowtimeVisibility được gọi để đặt isVisible = 1, sau đó làm mới danh sách suất chiếu



Hình . Cập nhật trạng thái tự động của Suất chiếu

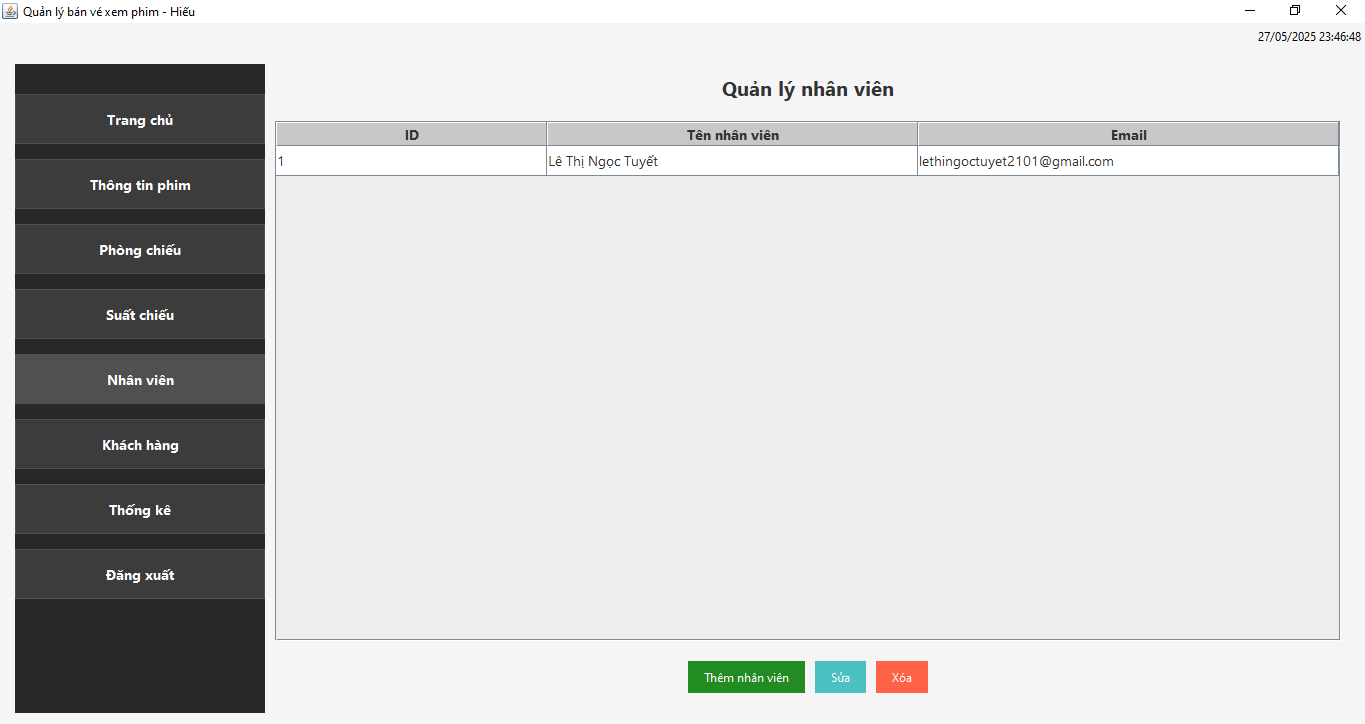
* Phương thức updateShowtimeStatuses trong ShowtimeBUS được gọi mỗi khi getAllShowtimes thực thi, đảm bảo trạng thái suất chiếu được cập nhật trước khi trả về dữ liệu

private void updateShowtimeStatuses(List<Showtime> showtimes) throws SQLException {  
 long currentTime = System.*currentTimeMillis*();   
 for (Showtime showtime : showtimes) {  
 String currentStatus = showtime.getStatus();  
  
 if (currentStatus != null && !currentStatus.equals("Ẩn")) {  
 if (showtime.getMovieID() == 0 || showtime.getShowDate() == null) {  
 if (!currentStatus.equals("Không chiếu")) {  
 showtimeDAO.updateShowtimeStatus(showtime.getShowtimeID(), "Không chiếu");  
 showtime.setStatus("Không chiếu");  
 }  
 continue;  
 }  
  
 long showTime = showtime.getShowDate().getTime();  
 Movie movie = movieBUS.getMovieById(showtime.getMovieID());  
 if (movie == null || movie.getDuration() <= 0) {  
 if (!currentStatus.equals("Không chiếu")) {  
 showtimeDAO.updateShowtimeStatus(showtime.getShowtimeID(), "Không chiếu");  
 showtime.setStatus("Không chiếu");  
 }  
 continue;  
 }  
  
 long duration = movie.getDuration() \* 60 \* 1000L;   
 String newStatus;  
 if (currentTime < showTime) {  
 newStatus = "Sắp công chiếu";  
 } else if (currentTime >= showTime && currentTime < showTime + duration) {  
 newStatus = "Đang chiếu";  
 } else {  
 newStatus = "Đã chiếu xong";  
 }  
 if (!newStatus.equals(currentStatus)) {  
 showtimeDAO.updateShowtimeStatus(showtime.getShowtimeID(), newStatus);  
 showtime.setStatus(newStatus);  
 }  
 }  
 }  
}

* Khi người quản trị nhấn nút "Ẩn" trong giao diện, phương thức showtimeBUS.updateShowtimeVisibility(showtimeId, 1) được gọi, cập nhật trường isVisible trong cơ sở dữ liệu từ 0 (hiển thị) thành 1 (ẩn).
* Trong ShowtimeBUS, phương thức updateShowtimeVisibility gọi showtimeDAO.updateShowtimeVisibility để thực hiện cập nhật này:

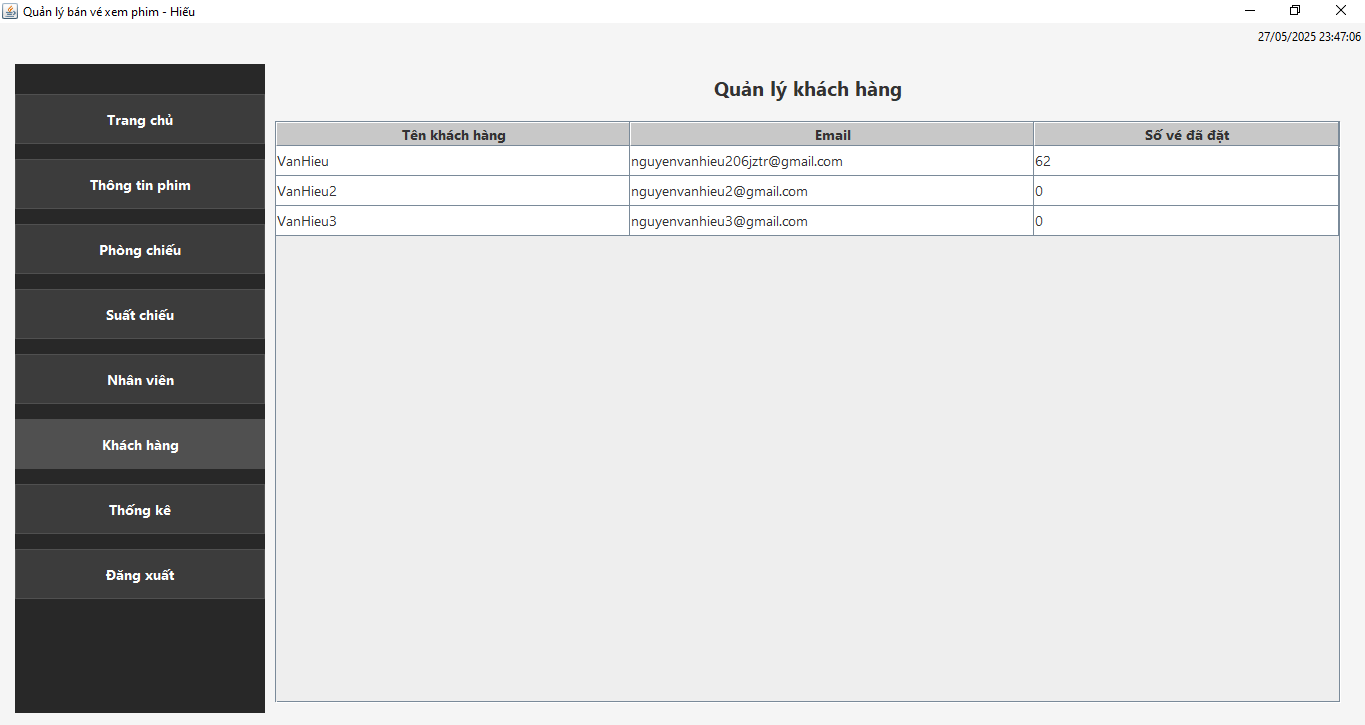
public void updateShowtimeVisibility(int showtimeID, int isVisible) throws SQLException {  
 showtimeDAO.updateShowtimeVisibility(showtimeID, isVisible);  
}

* Sau khi cập nhật, loadShowtimes được gọi để làm mới bảng, và các suất chiếu với isVisible = 1 sẽ không được hiển thị trên giao diện người dùng (logic chọn lọc trong ShowtimeDAO.getAllShowtimes).
* **Ảnh hưởng lên giao diện người dùng**:
* Giao diện người dùng sẽ chỉ hiển thị các suất chiếu có isVisible = 0. Điều này đảm bảo các suất chiếu đã ẩn (bị quản trị viên ẩn thủ công) không xuất hiện trong danh sách công khai.
* Trong AdminFrame, danh sách suất chiếu vẫn hiển thị tất cả suất chiếu (bao gồm cả các suất chiếu ẩn) để quản trị viên có thể xem và quản lý toàn bộ dữ liệu.
  + 1. Nhân viên



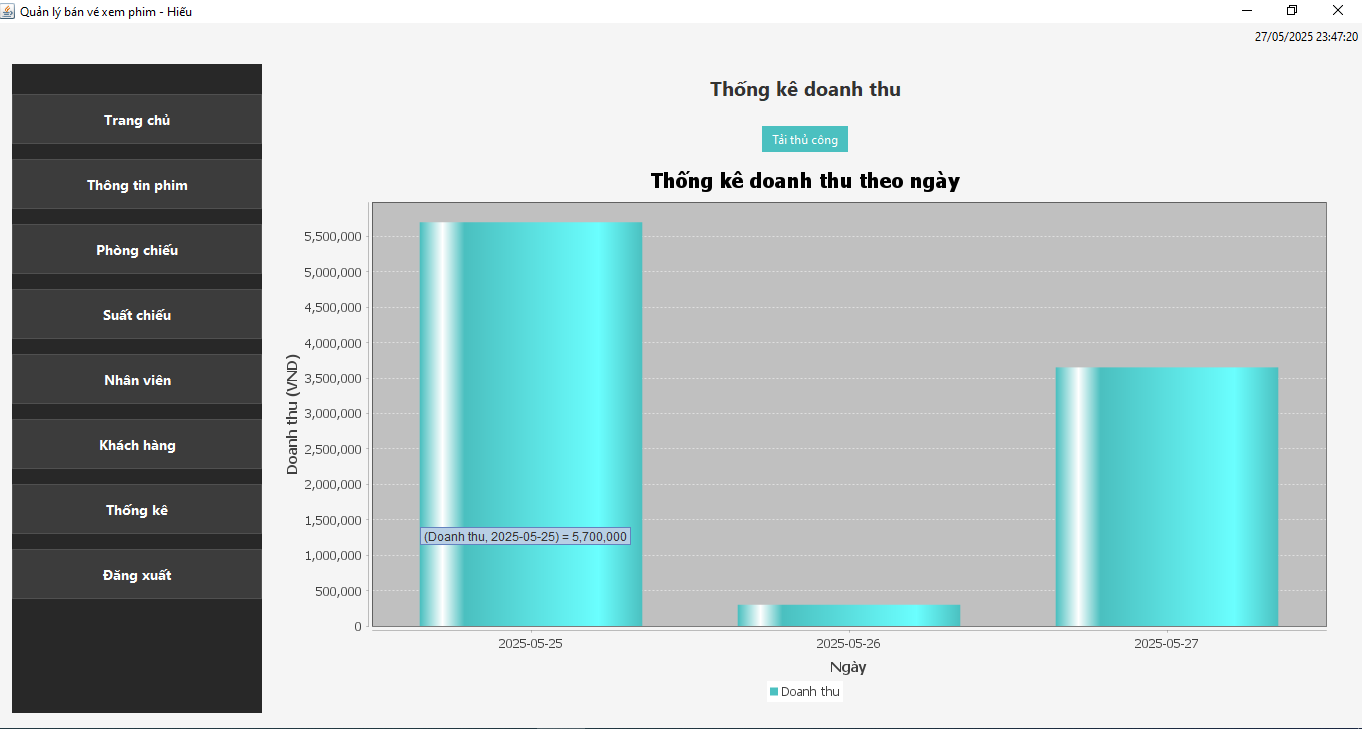
Hình . Nhân viên

* Phần này giúp Quản lý có thể dễ dàng quản lý được thông tin nhân viên và Thêm Sửa Xóa nhân viên
* Sử dụng CSDL là chính và tương tác với nó
  + 1. Khách hàng



Hình . Thông tin khách hàng A

* Phần này giúp Quản lý dễ dàng thông tin khách hàng, số vé khách hàng đã đặt. Từ đó tìm ra khách hàng tiềm năng hoặc khách hàng thân thiết, dễ dàng phân loại đối tượng để áp dụng các ưu đãi tùy từng đối tượng.
  + 1. Thống kê



Hình 3.14 Thống kê

* Hiển thị doanh thu theo từng ngày, được truy xuất từ cơ sở dữ liệu thông qua RevenueDAO.getDailyRevenue:

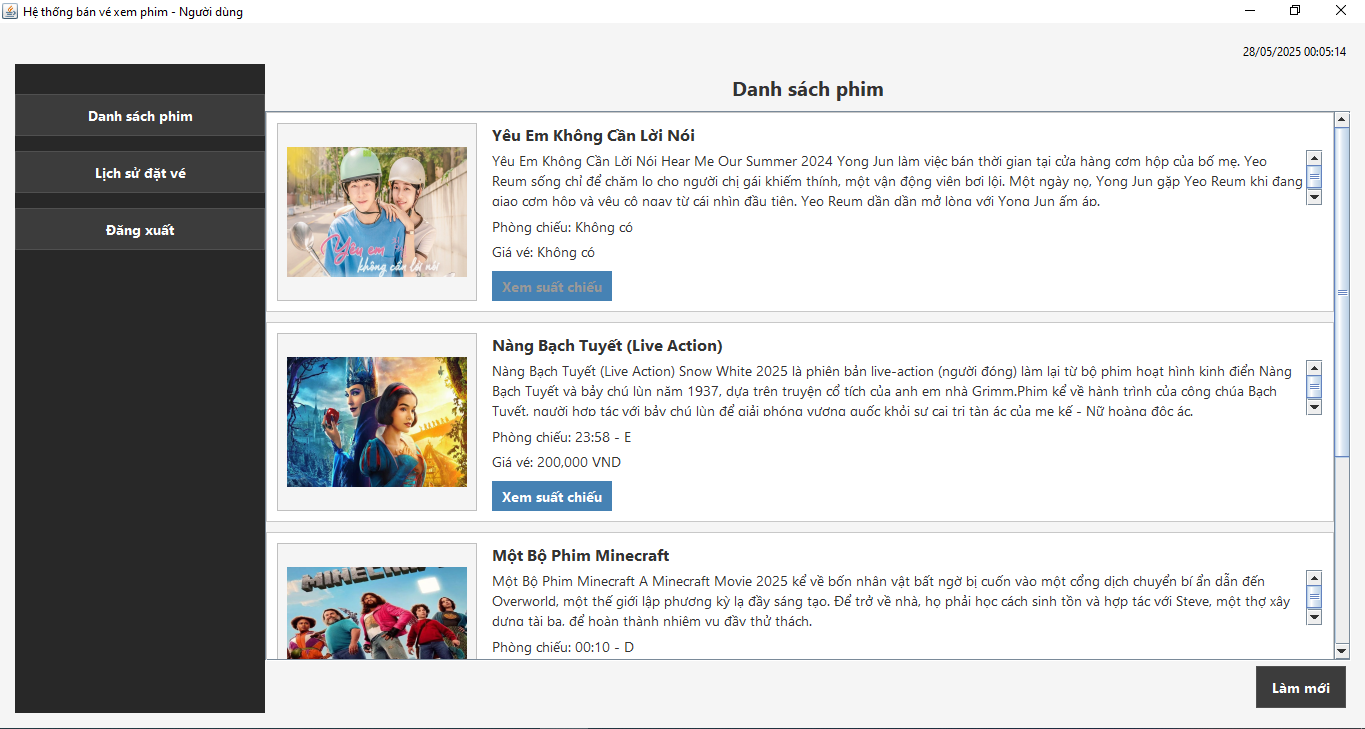
public Map<String, Double> getDailyRevenue() throws SQLException {  
 Map<String, Double> dailyRevenue = new TreeMap<>(); // Sử dụng TreeMap để tự động sắp xếp theo ngày  
 try (Connection conn = DBConnection.*getConnection*();  
 PreparedStatement stmt = conn.prepareStatement(*GET\_DAILY\_REVENUE*);  
 ResultSet rs = stmt.executeQuery()) {  
 while (rs.next()) {  
 String bookingDay = rs.getString("BookingDay");  
 double revenue = rs.getDouble("TotalRevenue");  
 dailyRevenue.put(bookingDay, revenue);  
 }  
 } catch (SQLException e) {  
 System.*err*.println("Error retrieving daily revenue: " + e.getMessage());  
 throw e;  
 }  
 return dailyRevenue;  
}

* **Giao diện**: Sử dụng biểu đồ cột (bar chart) được tạo bởi thư viện JFreeChart, hiển thị trên một ChartPanel:
* **Cơ chế cập nhật**: Tự động làm mới dữ liệu mỗi 30 giây và hỗ trợ tải thủ công thông qua nút "Tải thủ công":

// Lên lịch chạy task mỗi 30 giây  
timer.scheduleAtFixedRate(task, 0, 30000);

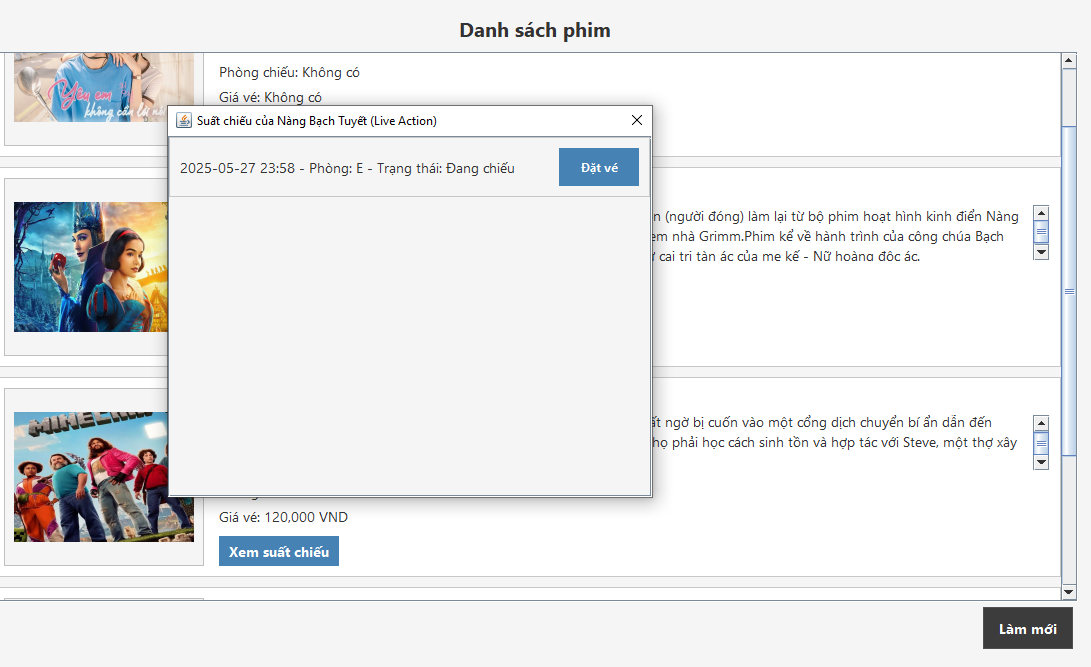
## Giao diện Người dùng

* + 1. Danh sách phim



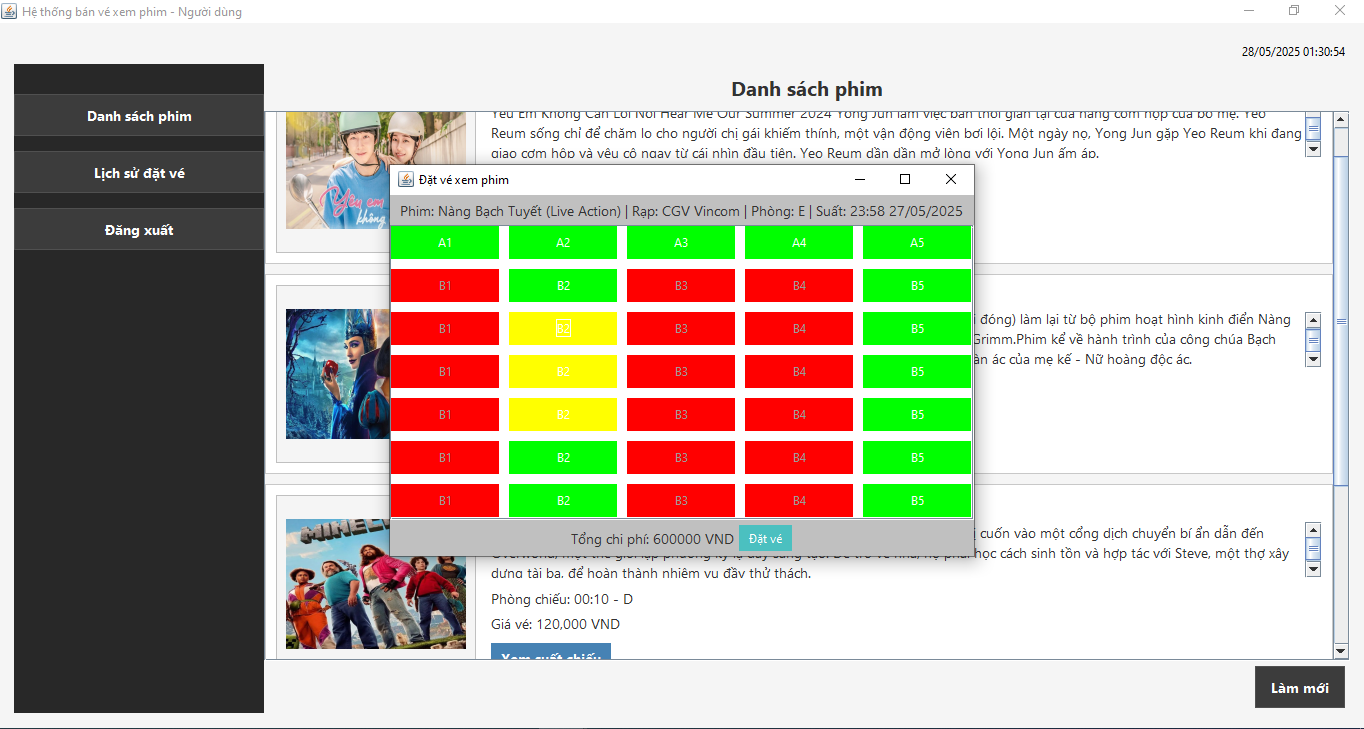
Hình . Danh sách phim

* Phần danh sách phim trong giao diện (phương thức createMoviesPanel trong UserFrame.java) cho phép người dùng xem thông tin các bộ phim hiện có, bao gồm tiêu đề, mô tả, hình ảnh, và các nút tương tác (như "Xem suất chiếu"). Dữ liệu phim được truy xuất từ movieBUS.getAllMovies()
* Đây là giao diện chính để người dùng chọn phim và tiến hành đặt vé.
* **Mục tiêu chính**: Hiển thị danh sách phim từ cơ sở dữ liệu, hỗ trợ người dùng chọn phim để xem chi tiết suất chiếu.
* **Giao diện**: Danh sách các panel chứa thông tin phim, được hiển thị trong một vùng cuộn (scrollable panel).
* **Cơ chế cập nhật**: Dữ liệu được tải khi người dùng truy cập panel hoặc nhấn nút "Làm mới".
* Khi đặt vé, người dùng sẽ lựa chọn phim muốn xem rồi xem tất cả suất chiếu:



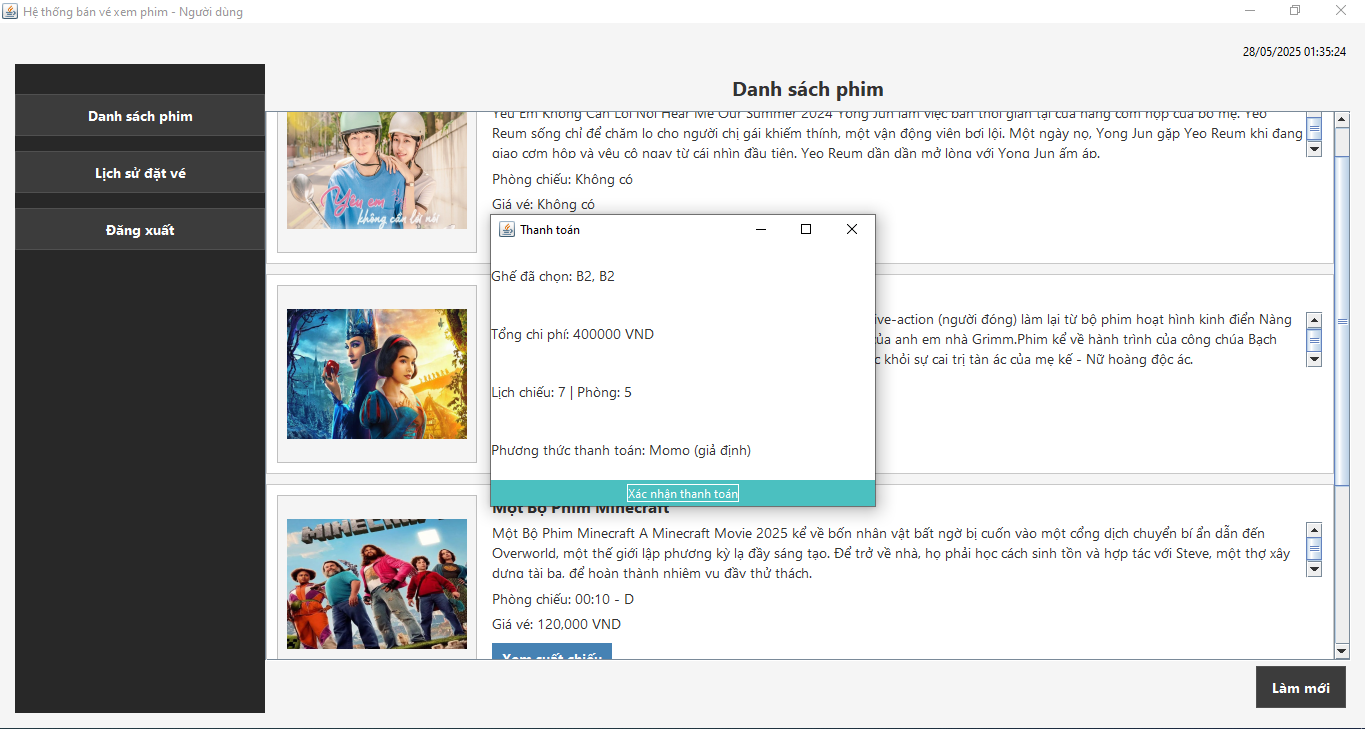
Hình . Xem tất cả suất chiếu của một phim

* Sau đó, người dùng sẽ chọn suất chiếu phù hợp cho mình rồi tiến hành đặt vé và chọn số lượng vị trí ghế



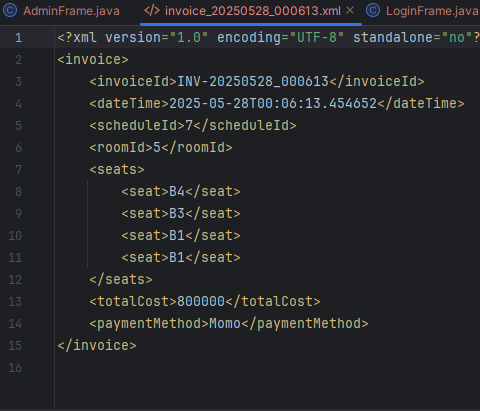
Hình . Đặt vé

* Người dùng đặt vé sẽ chọn số chỗ ngồi và ghế, hệ thống sẽ nhanh chóng tự động tính toán tổng bill cho bạn
* Trạng thái ghế trong suất chiếu;
* Ghế xanh: Còn trống
* Ghế đỏ: Đã có người đặt
* Ghế vàng: Ghế trống đang được bạn chọn



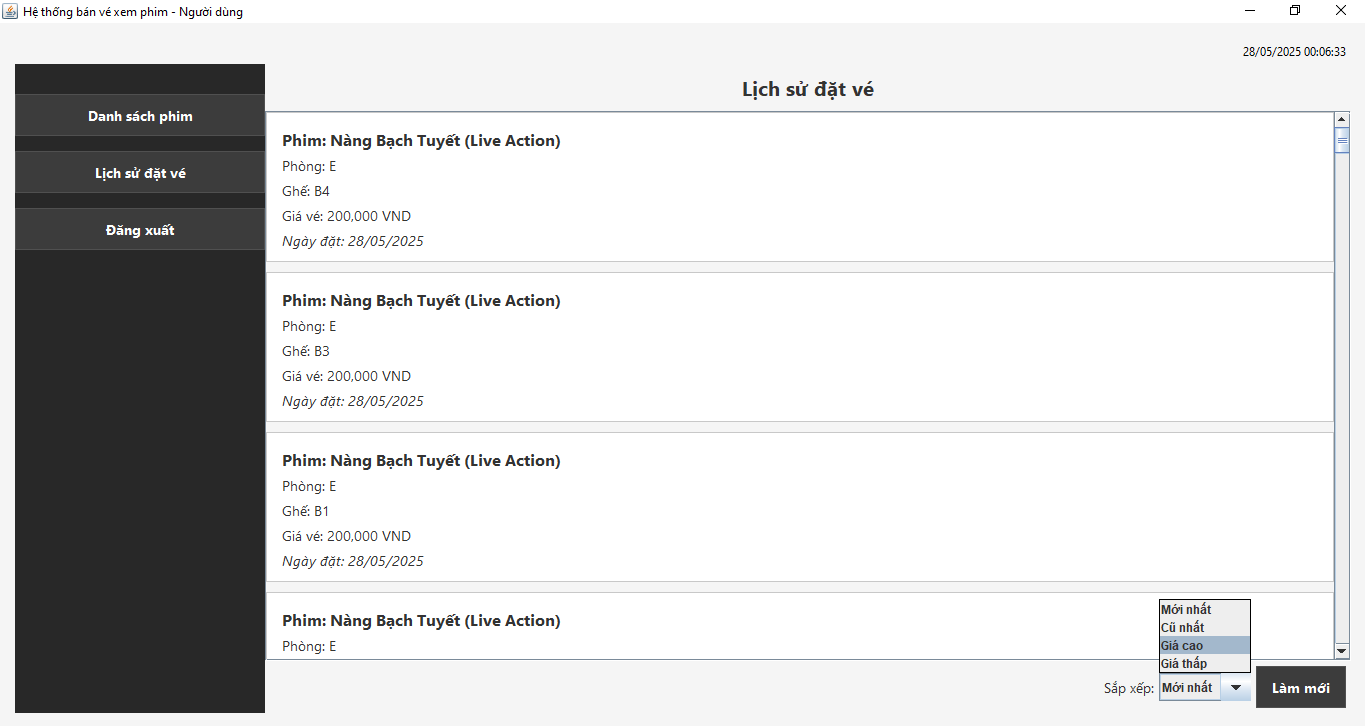
Hình . Xác nhận thanh toán

* Sau khi ấn Đặt vé, Cửa sổ Xác nhận thanh toán sẽ hiện ra với các thông tin về chi phí thanh toán, ghế, phương thức thanh toán, lịch chiếu.
* Khi đã thanh toán hệ thống sẽ tự động xuất hóa đơn dưới dạng file xml được lưu trong đường dẫn invoices trong dự án:



Hình . Hóa đơn

* + 1. Lịch sử đặt vé
* Khi đặt vé thành công, hệ thống sẽ tự động lưu vào CSDL lịch sử đặt vé theo từng người dùng:



Hình . Lịch sử đặt vé

* Người dùng dễ dàng truy xuất lịch sử đặt vé theo 4 kiểu sắp xếp: Mới nhất, Cũ nhất, Giá cao, Giá thấp.
* Lịch sử đặt vé với thiết kế dễ nhìn, đầy đủ thông tin như phòng, ghế, phim, giá và ngày đặt giúp người dùng dễ dàng kiểm soát, nắm bắt thông tin

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Kết luận: Trong thời gian làm đồ án:

##### Kết quả đạt được

* Về mặt lý thuyết:
* Nghiên cứu kỹ lưỡng các khái niệm liên quan đến hệ thống quản lý bán vé xem phim, bao gồm cách tổ chức dữ liệu như danh sách phim, suất chiếu, vé, và doanh thu trong cơ sở dữ liệu.
* Đã tìm hiểu các công nghệ cần thiết để phát triển hệ thống, chẳng hạn như sử dụng ngôn ngữ lập trình Java Swing để xây dựng giao diện người dùng, kết nối cơ sở dữ liệu để truy vấn thông tin, và hiển thị dữ liệu dưới dạng biểu đồ trực quan.
* Đã xây dựng tài liệu phân tích yêu cầu, xác định các chức năng chính của hệ thống:
* Đối với người dùng: Xem danh sách phim với thông tin như tiêu đề, mô tả, hình ảnh, và khả năng đặt vé.
* Đối với quản trị viên: Quản lý thông tin phim, suất chiếu, và theo dõi doanh thu.

Ví dụ: Hệ thống cho phép người dùng xem danh sách phim với hình ảnh minh họa, mô tả ngắn gọn, và thông tin về phòng chiếu, giá vé, giúp họ dễ dàng chọn phim phù hợp.

* Về thực nghiệm:
* **Chức năng chính đã triển khai**:
* Hệ thống đã xây dựng thành công giao diện cho người dùng, hiển thị danh sách phim với các thông tin cơ bản như tiêu đề, mô tả, và hình ảnh minh họa.
* Người dùng có thể xem lịch sử đặt vé của mình, với khả năng sắp xếp theo các tiêu chí như thời gian (mới nhất, cũ nhất) hoặc giá vé (cao đến thấp, thấp đến cao).
* Đối với quản trị viên, hệ thống cung cấp khả năng theo dõi doanh thu theo từng ngày, hiển thị dưới dạng biểu đồ cột, với tính năng tự động làm mới dữ liệu định kỳ (ví dụ: mỗi 30 giây).
* **Ổn định và hiệu suất**:
* Hệ thống đã được thử nghiệm với các tình huống cơ bản như người dùng đặt vé, xem lịch sử, và quản trị viên theo dõi doanh thu. Tất cả các chức năng đều hoạt động ổn định, không xảy ra lỗi nghiêm trọng.
* Dữ liệu được tải bất đồng bộ để đảm bảo giao diện không bị treo, ví dụ khi hiển thị danh sách phim hoặc lịch sử đặt vé, ngay cả khi có hàng trăm bản ghi.

Ví dụ: Khi quản trị viên muốn xem doanh thu, hệ thống tự động cập nhật biểu đồ mà không cần làm mới thủ công, giúp theo dõi dữ liệu gần thời gian thực.

* **Thành tựu nổi bật**:
* Hệ thống đảm bảo tính trực quan: Người dùng dễ dàng nắm bắt thông tin phim qua giao diện thân thiện, với hình ảnh và mô tả rõ ràng.
* Tính năng sắp xếp lịch sử đặt vé giúp người dùng nhanh chóng tìm kiếm thông tin theo nhu cầu, như xem các vé đặt gần đây nhất hoặc các vé có giá thấp nhất.
* Biểu đồ doanh thu cung cấp cái nhìn tổng quan về hiệu suất tài chính, hỗ trợ quản trị viên đưa ra quyết định kinh doanh.

##### Hạn chế và hướng phát triển

**Hạn chế:**

* Danh sách phim chưa có bộ lọc nâng cao, chẳng hạn như lọc theo thể loại (hành động, hài, kinh dị), ngày chiếu, hoặc độ tuổi phù hợp.
* Thông tin doanh thu chỉ hiển thị theo ngày, chưa hỗ trợ phân tích chi tiết theo phim, phòng chiếu, hoặc khoảng thời gian tùy chỉnh (ví dụ: theo tuần, tháng).
* Giao diện người dùng còn đơn giản, chưa hỗ trợ các tính năng tương tác như tìm kiếm phim theo từ khóa, gợi ý phim dựa trên lịch sử, hoặc hiển thị thông báo khi có suất chiếu mới.
* Hệ thống chưa tối ưu về hiệu suất khi xử lý dữ liệu lớn.

Hướng phát triển:

**Mục tiêu**:

* Đề xuất các giải pháp khắc phục hạn chế hiện tại và mở rộng hệ thống để đáp ứng nhu cầu người dùng trong tương lai.

**Nội dung chi tiết**:

* **Hạn chế hiện tại**:
* **Hiệu suất**: Việc tải toàn bộ dữ liệu mỗi lần làm mới gây chậm trễ khi số lượng dữ liệu lớn, ví dụ khi có hàng nghìn phim hoặc hàng chục nghìn giao dịch đặt vé.
* **Tính năng**:
  + Người dùng không thể lọc danh sách phim theo các tiêu chí như thể loại, ngày chiếu, hoặc độ tuổi, khiến việc tìm kiếm phim phù hợp mất nhiều thời gian.
  + Chưa có tính năng tìm kiếm phim theo từ khóa, ví dụ nhập tên phim hoặc đạo diễn để tìm nhanh.
  + Hệ thống chưa gợi ý phim dựa trên lịch sử đặt vé, ví dụ đề xuất các phim cùng thể loại với phim mà người dùng đã xem.
  + Thông tin doanh thu chỉ hiển thị theo ngày, chưa hỗ trợ phân tích chi tiết theo phim, phòng chiếu, hoặc khoảng thời gian tùy chỉnh.
* **Khả năng mở rộng**: Hệ thống hiện chỉ chạy trên máy tính để bàn, chưa hỗ trợ truy cập qua web hoặc ứng dụng di động.
* **Tính tương tác**: Giao diện chưa hỗ trợ các tính năng như thông báo về các suất chiếu mới.
* **Hướng phát triển chi tiết**:

**Hoàn thiện hệ thống** **bằng cách:**

* **Tối ưu hiệu suất**:

Áp dụng cơ chế lưu trữ tạm thời (cache) để giảm số lần truy vấn cơ sở dữ liệu, ví dụ chỉ tải dữ liệu mới hoặc dữ liệu thay đổi.

Thêm tính năng phân trang, mỗi lần chỉ hiển thị 20 phim hoặc 20 giao dịch đặt vé, thay vì tải toàn bộ.

* **Thêm tính năng nâng cao**:

Tích hợp bộ lọc cho danh sách phim, cho phép người dùng chọn thể loại (hành động, hài, kinh dị), ngày chiếu, hoặc độ tuổi phù hợp (trẻ em, người lớn).

Thêm ô tìm kiếm để người dùng nhập từ khóa (tên phim, đạo diễn, diễn viên) và hệ thống tự động lọc kết quả theo thời gian thực.

Cung cấp gợi ý phim dựa trên lịch sử đặt vé, ví dụ nếu người dùng thường xem phim hành động, hệ thống sẽ ưu tiên hiển thị các phim hành động mới.

Cải tiến tính năng theo dõi doanh thu để hỗ trợ phân tích chi tiết, ví dụ hiển thị doanh thu theo từng phim, từng phòng chiếu, hoặc theo khoảng thời gian tùy chỉnh (tuần, tháng, năm).

* **Cải thiện giao diện**:

Thêm hiệu ứng động khi người dùng di chuột qua thông tin phim, ví dụ làm nổi bật khung phim hoặc hiển thị thêm thông tin chi tiết.

Thêm thông báo đẩy (notification) để thông báo về các suất chiếu mới hoặc khuyến mãi.

* **Mở rộng phát triển**:
* **Triển khai đa nền tảng**:

Phát triển phiên bản web để người dùng có thể truy cập từ trình duyệt, giúp tăng khả năng tiếp cận.

Xây dựng ứng dụng di động cho iOS và Android, cho phép người dùng đặt vé mọi lúc mọi nơi.

* **Tích hợp dịch vụ mới**:

Thêm tính năng thanh toán trực tuyến qua ví điện tử hoặc thẻ ngân hàng, giúp người dùng thanh toán nhanh chóng mà không cần đến quầy.

Tích hợp hệ thống điểm thưởng cho khách hàng thân thiết, ví dụ cứ 10 vé được giảm giá 10% cho lần đặt tiếp theo.

* **Hỗ trợ đa ngôn ngữ**:

Thêm tùy chọn ngôn ngữ tiếng Anh và tiếng Việt để hỗ trợ người dùng quốc tế hoặc khách du lịch.

TÀI LIỆU THAM KHẢO