**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỎ ĐỊA CHẤT**

**KHOA: CÔNG NGHỆ THÔNG TIN**



**BÀI TẬP LỚN**

**HỌC PHẦN: ĐỒ ÁN 1**

**ĐỀ : XÂY DỰNG PHẦN MỀM QUẢN LÝ QUẦY BÁN VÉ XEM PHIM**

|  |  |  |
| --- | --- | --- |
| **Sinh viên thực hiện** | **Mã sinh viên** | **Lớp** |
| **Nguyễn Minh Quân** | **2221050125** | **DCCTKH67A** |
| **Phạm Cường Quốc** | **2221050002** | **DCCTKH67A** |
| **Bùi Phương Duy** | **2221050398** | **DCCTKH67A** |

**Giảng viên hướng dẫn:**   
**TS.Nguyễn Thị Mai Dung**

**HÀ NỘI, năm 2024**

**Bắc Ninh, năm 2024**

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC MỎ ĐỊA CHẤT**

**KHOA: CÔNG NGHỆ THÔNG TIN**

**BÀI TẬP LỚN**

**HỌC PHẦN: ĐỒ ÁN 1**

**ĐỀ TÀI: XÂY DỰNG PHẦN MỀM QUẢN LÝ QUẦY BÁN VÉ XEM PHIM**



|  |  |
| --- | --- |
|  |  |

**HÀ NỘI, năm 2024**

**Bắc Ninh, năm 2024**

MỤC LỤC

[MỤC LỤC 3](#_Toc170351724)

[MỤC LỤC HÌNH ẢNH 5](#_Toc170351725)

[MỞ ĐẦU 7](#_Toc170351726)

[I. Giới thiệu tổng quan đề tài 7](#_Toc170351727)

[II. Phân tích thiết kế hệ thống 7](#_Toc170351728)

[1. Xác định yêu cầu của đề tài 7](#_Toc170351729)

[a) Đối với Admin 7](#_Toc170351730)

[b) Đối với nhân viên 7](#_Toc170351731)

[2. Sơ đồ hoạt động của hệ thống 8](#_Toc170351732)

[3. Sơ đồ phân rã chức năng của hệ thống 8](#_Toc170351733)

[4. Biểu đồ Usecase của hệ thống 9](#_Toc170351734)

[5. Biểu đồ tuần tự bán vé của hệ thống 12](#_Toc170351735)

[III. Phân công công việc 13](#_Toc170351736)

[CHƯƠNG I: CƠ SỞ LÝ THUYẾT 14](#_Toc170351737)

[I. Công cụ xây dựng đề tài 14](#_Toc170351738)

[1. Visual Studio Code 14](#_Toc170351739)

[2. MySQL Workbench 8.0 CE 14](#_Toc170351740)

[3. GitHub 14](#_Toc170351741)

[4. Qt designer 15](#_Toc170351742)

[II. Các module/thư viện Python được sử dụng 15](#_Toc170351743)

[CHƯƠNG II: XÂY DỰNG CHƯƠNG TRÌNH 17](#_Toc170351744)

[1. Cấu trúc dữ liệu của chương trình 17](#_Toc170351745)

[a) Database 17](#_Toc170351746)

[b) Cấu trúc file dữ liệu 20](#_Toc170351747)

[2. Cấu trúc mã nguồn 21](#_Toc170351748)

[3. Các giao diện 23](#_Toc170351749)

[a) Giao diện đăng nhập đối với Admin 23](#_Toc170351750)

[b) Giao diện quản lý nhân viên đối với Admin 24](#_Toc170351751)

[c) Giao diện quản lý phim đối với Admin 25](#_Toc170351752)

[d) Giao diện báo cáo thống kê đối với Admin 26](#_Toc170351753)

[e) Giao diện quản lý lịch chiếu đối với Admin 27](#_Toc170351754)

[f) Giao diện đăng nhập đối với nhân viên 28](#_Toc170351755)

[g) Giao diện bán vé đối với nhân viên 29](#_Toc170351756)

[h) Giao diện đối với nhân viên soát vé 30](#_Toc170351757)

[CHƯƠNG III: KẾT QUẢ THỰC NGHIỆM CHƯƠNG TRÌNH 32](#_Toc170351758)

[I. Demo chương trình 32](#_Toc170351759)

[1. Đăng nhập với vai trò Admin 32](#_Toc170351760)

[2. Trang quản lý nhân viên 32](#_Toc170351761)

[3. Trang quản lý phim 35](#_Toc170351762)

[4. Trang quản lý lịch chiếu 38](#_Toc170351763)

[5. Báo cáo thống kê 42](#_Toc170351764)

[6. Trang bán vé 43](#_Toc170351765)

[II. Dữ liệu để test 44](#_Toc170351766)

[1. Đối với bán vé 44](#_Toc170351767)

[2. Đối với thêm phim 45](#_Toc170351768)

3. Biểu đồ trực quan hóa………………………………………………………45

[KẾT LUẬN 46](#_Toc170351769)

[Kết quả đạt được 46](#_Toc170351770)

[a) Đối với Admin 46](#_Toc170351771)

[b) Đối với nhân viên 46](#_Toc170351772)

[c) Khác 46](#_Toc170351773)

[Định hướng và phát triển 46](#_Toc170351774)

[TÀI LIỆU THAM KHẢO 47](#_Toc170351775)

MỤC LỤC HÌNH ẢNH

[Ảnh 1. Sơ đồ hoạt động của quản lý quầy bán vé 8](#_Toc170351783)

[Ảnh 2. Sơ đồ phân rã chức năng quản lý quầy bán vé 8](#_Toc170351784)

[Ảnh 3. Usecase quản lý nhân viên 9](#_Toc170351785)

[Ảnh 4. Usecase quản lý phim 10](#_Toc170351786)

[Ảnh 5. Usecase quản lý lịch chiếu 10](#_Toc170351787)

[Ảnh 6. Usecase báo cáo thống kê 11](#_Toc170351788)

[Ảnh 7. Usecase quản lý bán vé 11](#_Toc170351789)

[Ảnh 8. Biểu đồ tuần tự bán vé 1 12](#_Toc170351790)

[Ảnh 9. Biểu đồ tuần tự bán vé 2 12](#_Toc170351791)

[Ảnh 10. Biểu đồ tuần tự bán vé 3 12](#_Toc170351792)

[Ảnh 11. Mô tả cấu trúc data 17](#_Toc170351793)

[Ảnh 12. Cây thư mục 21](#_Toc170351794)

[Ảnh 13. Cấu trúc mã nguồn 23](#_Toc170351795)

[Ảnh 14. Đăng nhập 24](#_Toc170351796)

[Ảnh 15. Thông tin nhân viên 24](#_Toc170351797)

[Ảnh 16. Thêm nhân viên 25](#_Toc170351798)

[Ảnh 17. Thông tin phim 25](#_Toc170351799)

[Ảnh 18. Thêm phim 26](#_Toc170351800)

[Ảnh 19. Thống kê doanh thu 26](#_Toc170351801)

[Ảnh 20. Thống kê phim 27](#_Toc170351802)

[Ảnh 21. Thống kê vé bán 27](#_Toc170351803)

[Ảnh 22. Thông tin lịch chiếu 28](#_Toc170351804)

[Ảnh 23. Thêm lịch chiếu 28](#_Toc170351805)

[Ảnh 24. Đăng nhập đối với nhân viên 29](#_Toc170351806)

[Ảnh 25. Đặt vé 29](#_Toc170351807)

[Ảnh 26. Hoá đơn thanh toán 30](#_Toc170351808)

[Ảnh 27. Đăng nhập với nhân viên soát vé 30](#_Toc170351809)

[Ảnh 28. Checkin vé 31](#_Toc170351810)

[Ảnh 29. Đăng nhập với vai trò admin 32](#_Toc170351811)

[Ảnh 30. Giao diện tất cả nhân viên 32](#_Toc170351812)

[Ảnh 31. Thêm nhân viên 34](#_Toc170351813)

[Ảnh 32. Giao diện tất cả phim 35](#_Toc170351814)

[Ảnh 33. Thêm phim 36](#_Toc170351815)

[Ảnh 34. Thêm lịch chiếu 38](#_Toc170351816)

[Ảnh 35. Sửa lịch chiếu 40](#_Toc170351817)

[Ảnh 36. Thống kê doanh thu 42](#_Toc170351818)

[**Ảnh 37. Trực quan hóa………………………………………………………………45**](#_Toc170351818)

MỞ ĐẦU

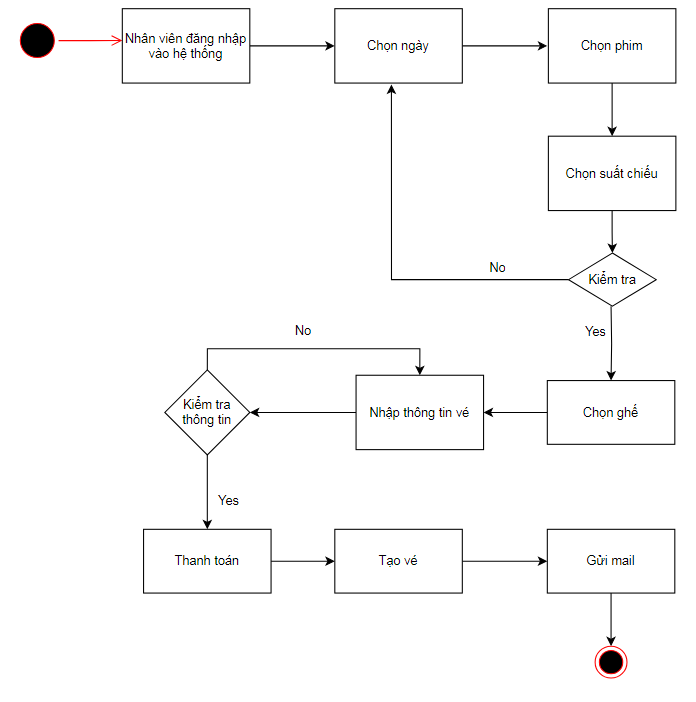
* 1. Giới thiệu tổng quan đề tài

Ngày nay, với sự phát triển không ngừng của công nghệ nhất là ngành công nghệ thông tin Công nghệ thông tin được ứng dụng trong rất nhiều các lĩnh vực khác nhau. Với khả năng ứng dụng rộng rãi đó cùng với những chính sách phát triển ngành công nghệ của nhà nước, cùng với việc không ngừng nâng cao chất lượng dịch vụ cũng như chương trình quản lý các ngành dịch vụ.

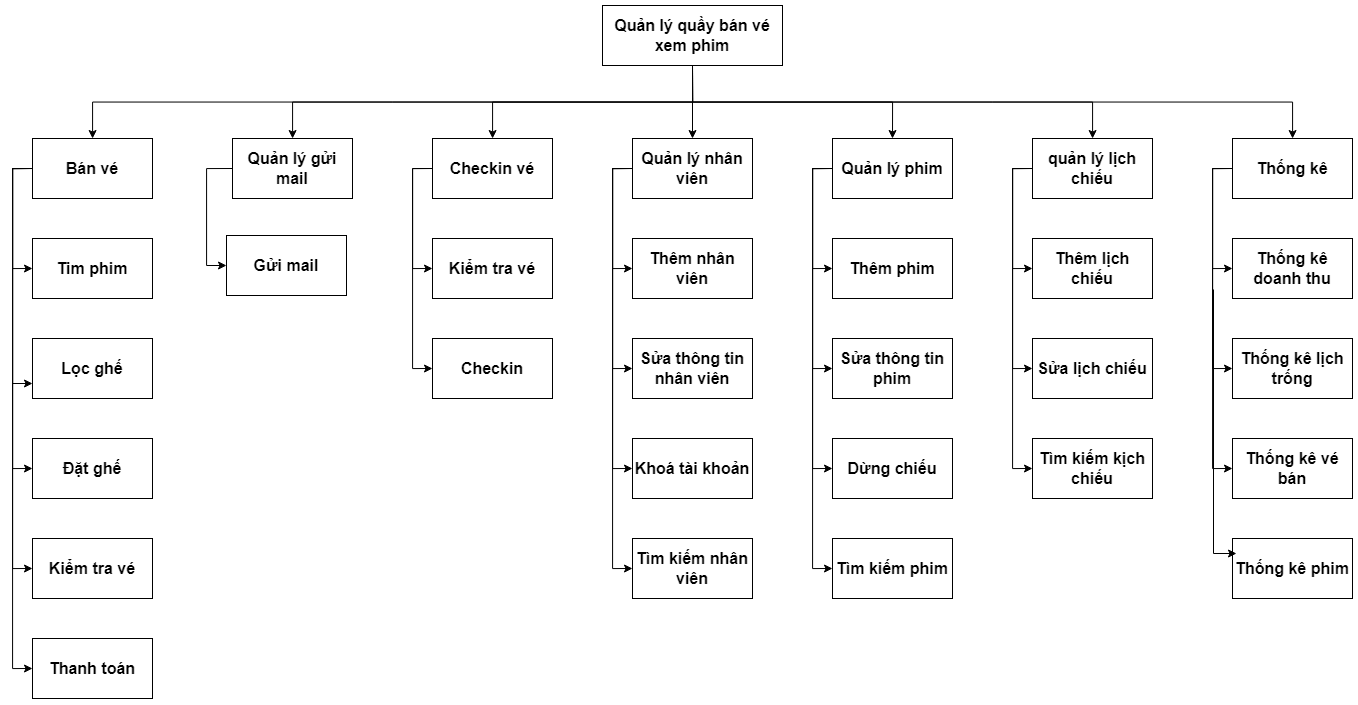
Với khả năng ứng dụng rộng rãi của ngành công nghệ thông tin, việc đưa tin học vào hỗ trợ các công việc quản lý bán vé xem phim cho ngành dịch vụ ứng dụng này nhằm giảm tải được công việc của những người bán về và người quản lý thông tin phim, lịch chiếu, chỗ ngồi, giá vé, tình trạng vé, báo cáo doanh thu. Đây là những ứng dụng rất cần thiết đối với các rạp chiếu phim nhất là các rạp chiếu phim lớn

Xuất phát từ nhu cầu đó, nhóm chúng em thực hiện đề tài “**Xây dựng phần mềm quản lý quầy bán vé xem phim”.** Hệ thống này nhằm nâng cao chất lượng quản lý hệ thống bán vé tại rạp, tránh các sai sót vẫn mặc phải khi còn làm các phương pháp có điện. Trọng tâm của đề tài là phân tích, xây dựng phần mềm quản lý quầy bán vé xem phim giúp người sử dụng dễ dàng làm việc, người xem dễ hiểu.

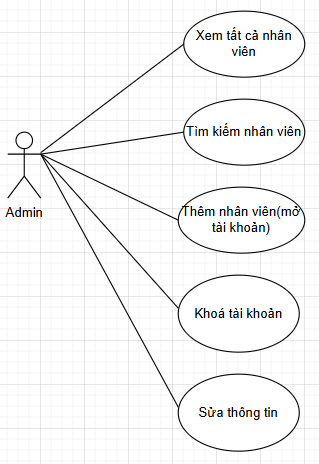
* 1. Phân tích thiết kế hệ thống
     1. Xác định yêu cầu của đề tài
        + 1. Đối với Admin
* Quản lý nhân viên: thêm nhân viên, sửa thông tin nhân viên, khoá tài khoản, tìm kiếm nhân viên.
* Quản lý lịch chiếu: thêm lịch chiếu, sửa lịch chiếu, tìm kiếm lịch chiếu.
* Quản lý phim: thêm phim, sửa thông tin phim, dừng chiếu phim, tìm kiếm phim.
* Thống kê: thống kê doanh thu, thống kê lịch trống, thống kê vé bán, thống kê phim.
  + - * 1. Đối với nhân viên
* Bán vé: tìm phim, lọc ghế, đặt ghế, kiểm tra, thanh toán
* Checkin: kiểm tra vé, checkin
  + 1. Sơ đồ hoạt động của hệ thống



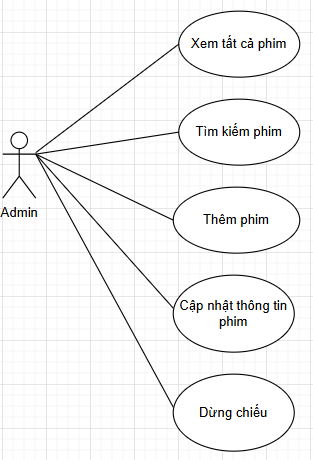
1. Sơ đồ hoạt động của quản lý quầy bán vé
   * 1. Sơ đồ phân rã chức năng của hệ thống



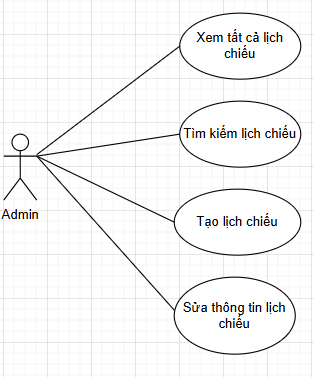
1. Sơ đồ phân rã chức năng quản lý quầy bán vé
   * 1. Biểu đồ Usecase của hệ thống



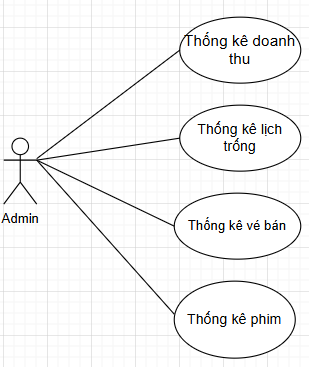
1. Usecase quản lý nhân viên



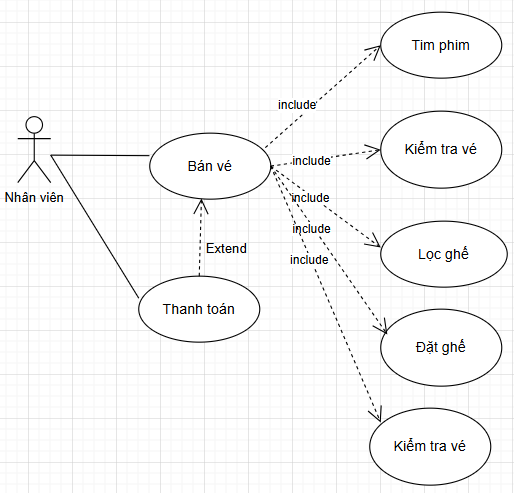
1. Usecase quản lý phim



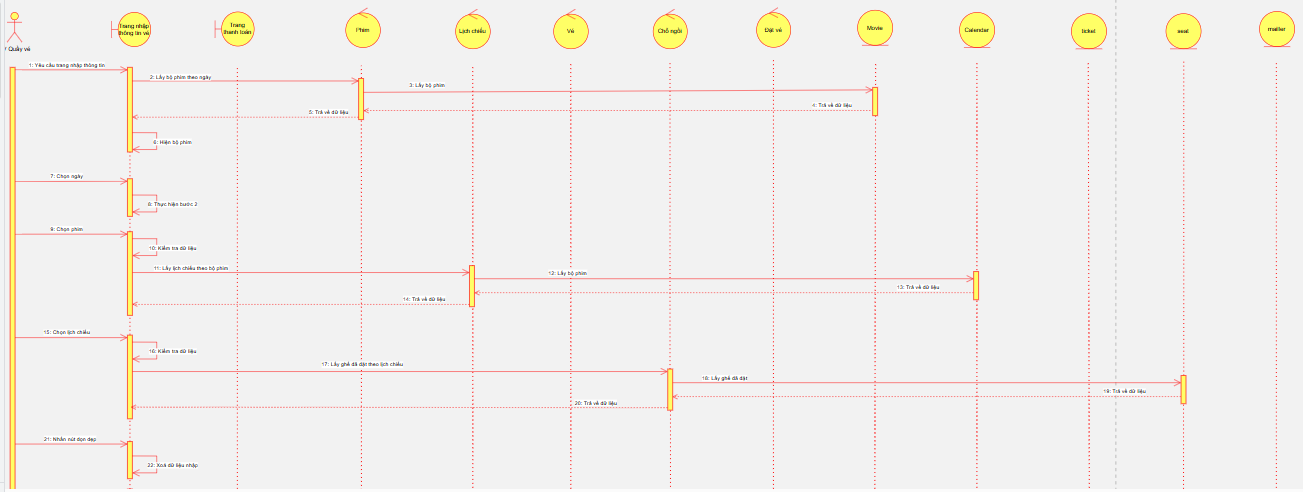
1. Usecase quản lý lịch chiếu



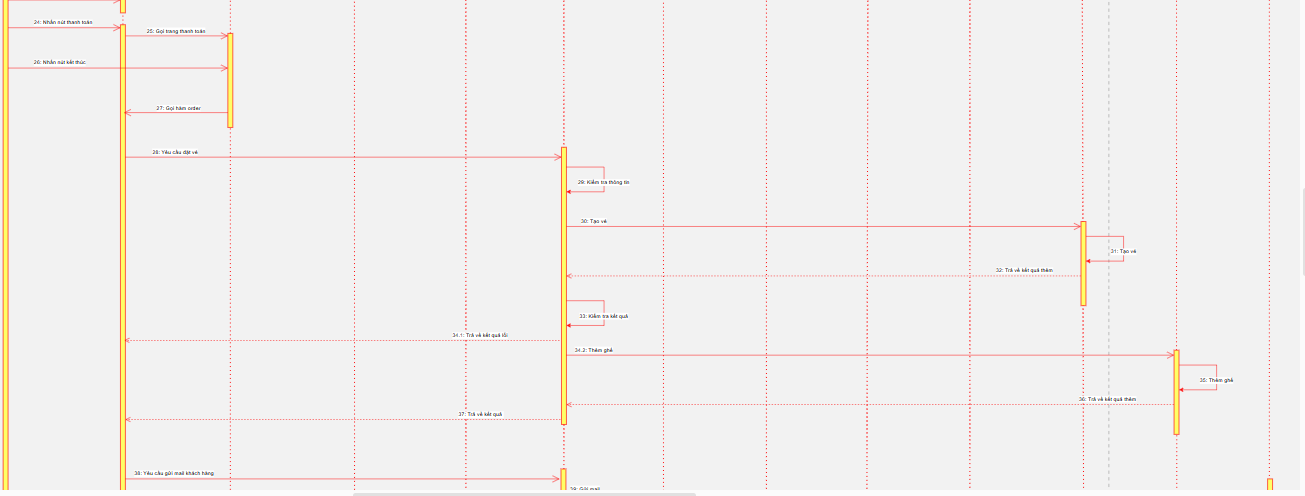
1. Usecase báo cáo thống kê



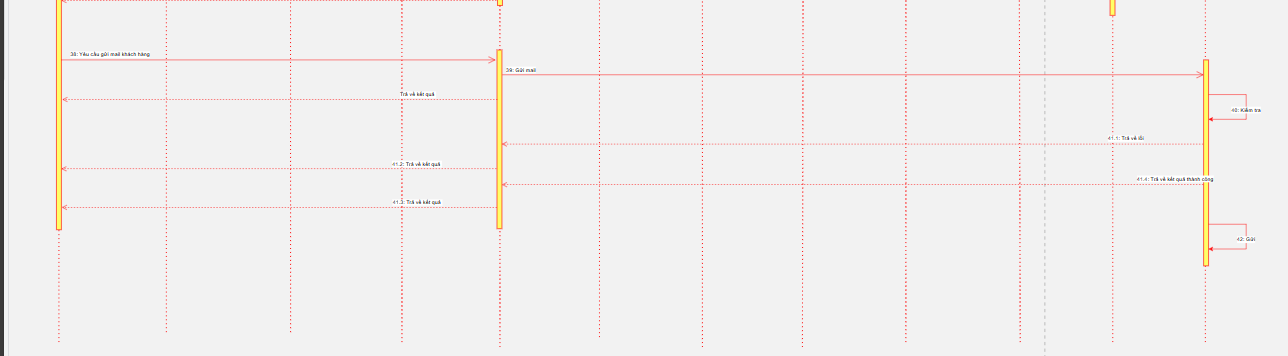
1. Usecase quản lý bán vé
   * 1. Biểu đồ tuần tự bán vé của hệ thống



1. Biểu đồ tuần tự bán vé 1



1. Biểu đồ tuần tự bán vé 2



1. Biểu đồ tuần tự bán vé 3
   1. Phân công công việc

|  |  |
| --- | --- |
| Họ tên thành viên | Công việc |
| Nguyễn Minh Quân | +) Quản lý nhân viên  +) Quản lý phim  +) Quản lý bán vé  +) Làm báo cáo  +) Thuyết trình  +) Tạo dữ liệu |
| Phạm Cường Quốc | +) Quản lý lịch chiếu  +) Quản lý bán vé  +) Thống kê  +) Làm báo cáo  +) Làm slide |
| Bùi Phương Duy | +) Quản lý gửi email  +) Check In vé |

1. CƠ SỞ LÝ THUYẾT
   1. Công cụ xây dựng đề tài
      1. Visual Studio Code

Visual Studio Code (VS Code) là một trình soạn thảo mã nguồn mở, miễn phí và đa nền tảng, được phát triển bởi Microsoft. Nó được thiết kế để hỗ trợ nhiều ngôn ngữ lập trình và các công cụ phát triển khác nhau, làm cho việc viết, gỡ lỗi và duy trì mã trở nên dễ dàng hơn. Các công dụng chính của VS Code:

* Phát triển và chỉnh sửa mã nguồn
* Quản lý dự án và tệp
* Gỡ lỗi (Debugging)
* Tích hợp Git và Quản lý phiên bản
* Tiện ích mở rộng (Extensions)
* Tích hợp Terminal
* Tính năng cộng tác
* Trực quan hóa dữ liệu
* Tùy chỉnh và cá nhân hóa
* Hỗ trợ phát triển web và ứng dụng
  + 1. MySQL Workbench 8.0 CE

MySQL Workbench 8.0 CE (Community Edition) là một công cụ quản lý cơ sở dữ liệu miễn phí và mã nguồn mở, được phát triển bởi Oracle. Đây là một ứng dụng đồ họa mạnh mẽ dành cho quản trị cơ sở dữ liệu MySQL, cung cấp nhiều tính năng để thiết kế, phát triển và quản lý cơ sở dữ liệu. Công dụng của MySQL Workbench 8.0 CE:

* Thiết kế cơ sở dữ liệu (Database Design)
* Phát triển SQL (SQL Development)
* Quản trị cơ sở dữ liệu (Database Administration)
* Tối ưu hóa và kiểm tra hiệu năng (Performance Tuning and Monitoring)
* Di chuyển cơ sở dữ liệu (Database Migration)
* Tích hợp và tùy chỉnh (Integration and Customization)
  + 1. GitHub

GitHub là một nền tảng lưu trữ mã nguồn và cộng tác phát triển phần mềm, sử dụng hệ thống quản lý phiên bản Git. Nó được thành lập vào năm 2008 và hiện nay thuộc sở hữu của Microsoft. GitHub cung cấp một loạt các công cụ và dịch vụ giúp các lập trình viên và nhóm phát triển phần mềm làm việc hiệu quả hơn, từ quản lý mã nguồn đến tự động hóa quy trình triển khai. Công dụng chính của GitHub

* Lưu trữ mã nguồn (Code Hosting)
* Quản lý phiên bản (Version Control)
* Cộng tác và làm việc nhóm (Collaboration)
* Quản lý dự án (Project Management)
* Tích hợp và triển khai liên tục (CI/CD)
* Mạng xã hội cho lập trình viên
* Bảo mật
  + 1. Qt designer

Qt Designer là một công cụ thiết kế giao diện đồ họa (GUI) trực quan, cung cấp giao diện kéo-thả để tạo các cửa sổ và widget (thành phần giao diện) của ứng dụng. Các thiết kế được tạo ra bằng Qt Designer được lưu trữ dưới dạng các tệp XML với phần mở rộng.ui. Công dụng của Qt Designer:

* Thiết kế giao diện người dùng trực quan
* Tạo và quản lý các tệp “.ui”
* Tích hợp với các công cụ Qt khác
  1. Các module/thư viện Python được sử dụng

|  |  |
| --- | --- |
| **Tên thư viện** | **Chức năng** |
| dotenv | Sử dụng với công dụng đẩy dữ liệu từ file .env vào môi trường hệ thống |
| smtplib | Sử dụng với chức năng gửi gmail |
| ssl | Sử dụng để tạo kết nối an toàn khi giao tiếp với máy chủ smtp multiprocessing |
| jinja2 | Sử dụng để đẩy dữ liệu vào file html |
| calendar |  |
| mysql | Sử dụng với mục đích kết nối với cơ sở dữ liệu mysql và truy vấn dữ liệu |
| locale | Sử dụng với việc chuyển số thành dạng tiền tệ |
| hashlib | Mã hoá dữ liệu |
| hmac | Sử dụng để so sánh 2 bytes |
| base64 | Sử dụng trong việc chuyển bytes thành chuỗi base64 |
| random | Sử dụng để lấy ra ngẫu nhiên 1 số |
| PyQt6 | Sử dụng để tạo ra giao diện trực quan |
| pyzbar | Sử dụng để giải mã ảnh và tìm ra mã qr bên trong |
| opencv-python | Sử dụng để lấy hình ảnh và xử lý ảnh từ camera |
| re | Sử dụng với mục đích kiểm tra chuỗi bằng đoạn mã mô tả dữ liệu |

1. XÂY DỰNG CHƯƠNG TRÌNH
   * 1. Cấu trúc dữ liệu của chương trình
        + 1. Database

A screenshot of a computer program

Description automatically generated

1. Mô tả cấu trúc data

|  |  |
| --- | --- |
| Đối tượng: Staff (nhân viên) | Kiểu dữ liệu |
| **idnv** | varchar(25) |
| name | varchar(40) |
| sdt | varchar(10) |
| email | varchar(50) |
| sex | int(11) |
| rank | enum |
| blockAt | bigint(20) |
| password | varchar(255) |

|  |  |
| --- | --- |
| Đối tượng: movie (phim) | Kiểu dữ liệu |
| **id** | varchar(25) |
| name | varchar(255) |
| age | int(11) |
| minPrice | int(11) |
| hideAt | bigint(20) |
| createAt | bigint(20) |
| time | int(11) |
| image | varchar(100) |

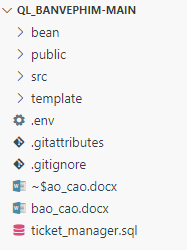
|  |  |
| --- | --- |
| Đối tượng: calendar(lịch chiếu) | Kiểu dữ liệu |
| **id** | varchar(25) |
| time | double |
| *idMovie* | varchar(25) |
| cancleAt | bigint(20) |
| room | int(11) |

|  |  |
| --- | --- |
| Đối tượng: ticket(vé) | Kiểu dữ liệu |
| **id** | varchar(25) |
| *idCalendar* | varchar(25) |
| name | varchar(40) |
| numPerson | int(11) |
| numPopcorn | int(11) |
| numWater | int(11) |
| priceWater | int(11) |
| priceTicket | int(11) |
| pricePopcorn | int(11) |
| email | varchar(50) |
| createAt | bigint(20) |
| *createBy* | varchar(25) |
| checkinAt | bigint(20) |
| *checkinBy* | varchar(25) |

|  |  |
| --- | --- |
| Đối tượng: seat(Chỗ ngồi) | Kiểu dữ liệu |
| **id** | bigint(20) |
| *idTicket* | varchar(25) |
| location | varchar(3) |
| *idCalendar* | varchar(25) |

Chú thích: những dòng in đậm, gạch chân ở dưới là khóa chính của bảng; những dòng in nghiêng, gạch chân ở dưới là khóa ngoại của bảng.

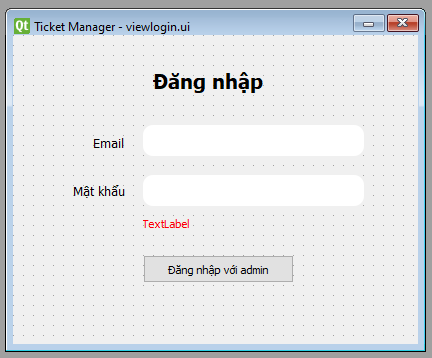
* + - * 1. Cấu trúc file dữ liệu



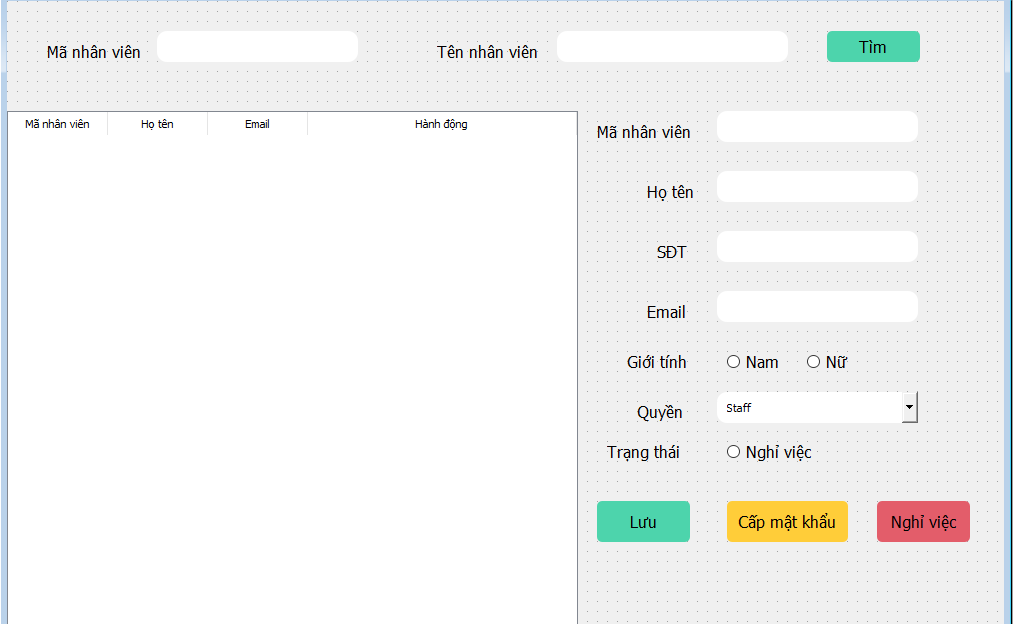
1. Cây thư mục
   * 1. Cấu trúc mã nguồn

|  |  |
| --- | --- |
|  |  |
|  |  |

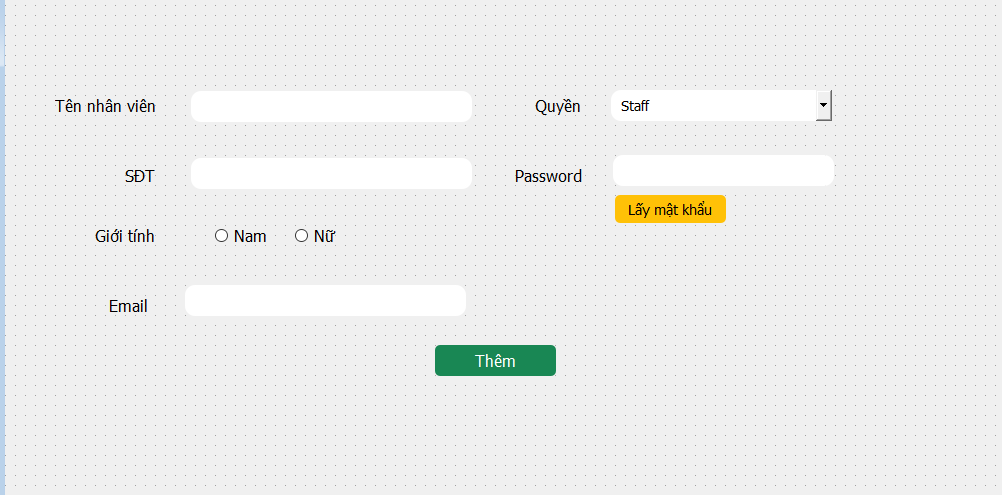
1. Cấu trúc mã nguồn
   * 1. Các giao diện
        + 1. Giao diện đăng nhập đối với Admin



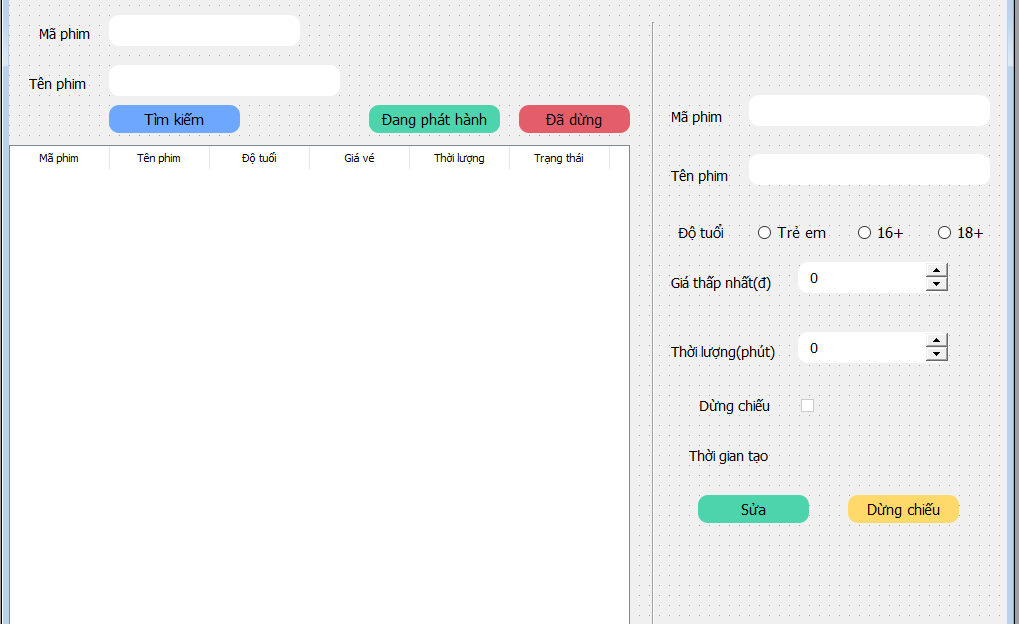
1. Đăng nhập
   * + - 1. Giao diện quản lý nhân viên đối với Admin



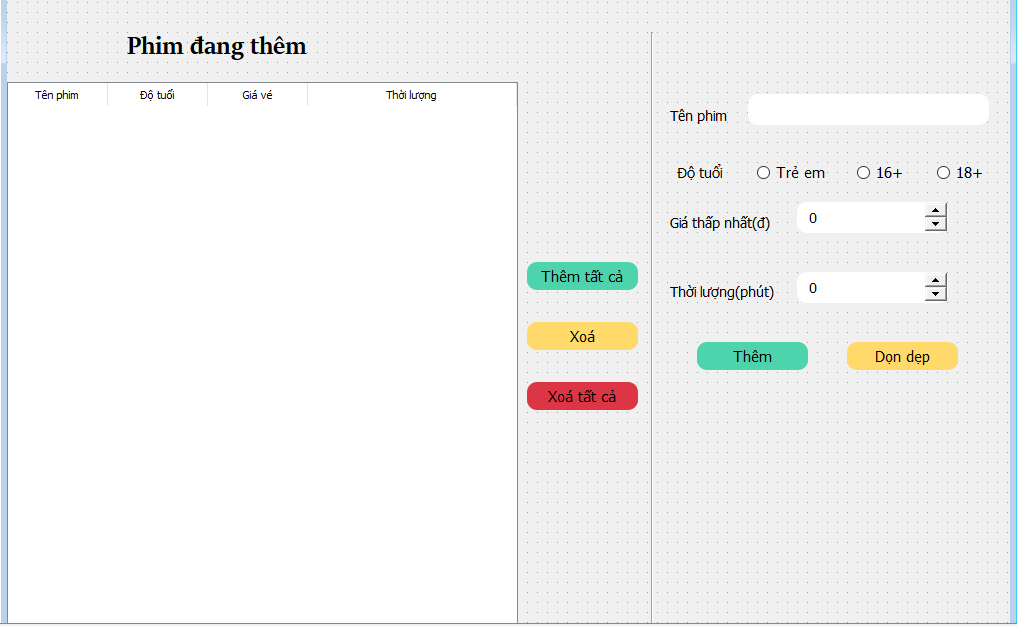
1. Thông tin nhân viên



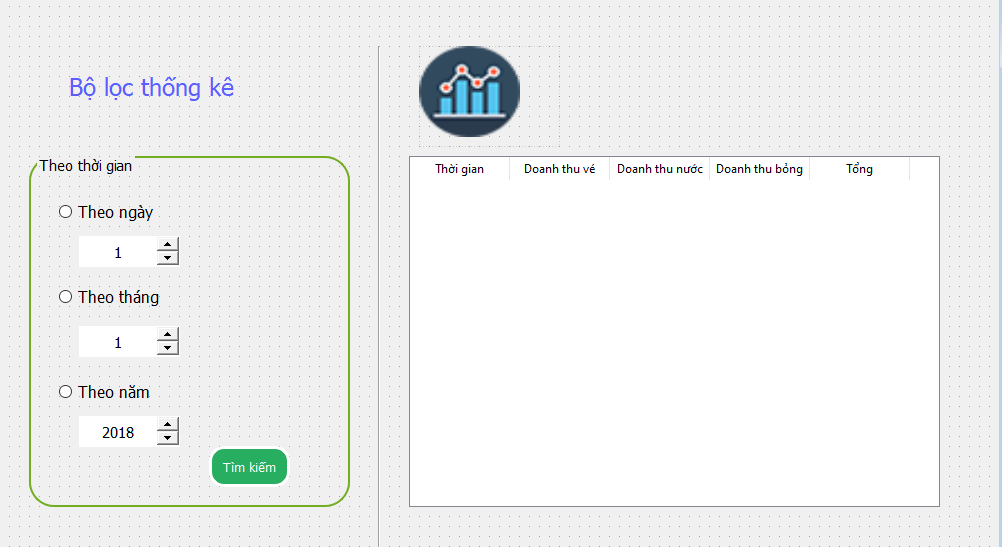
1. Thêm nhân viên
   * + - 1. Giao diện quản lý phim đối với Admin



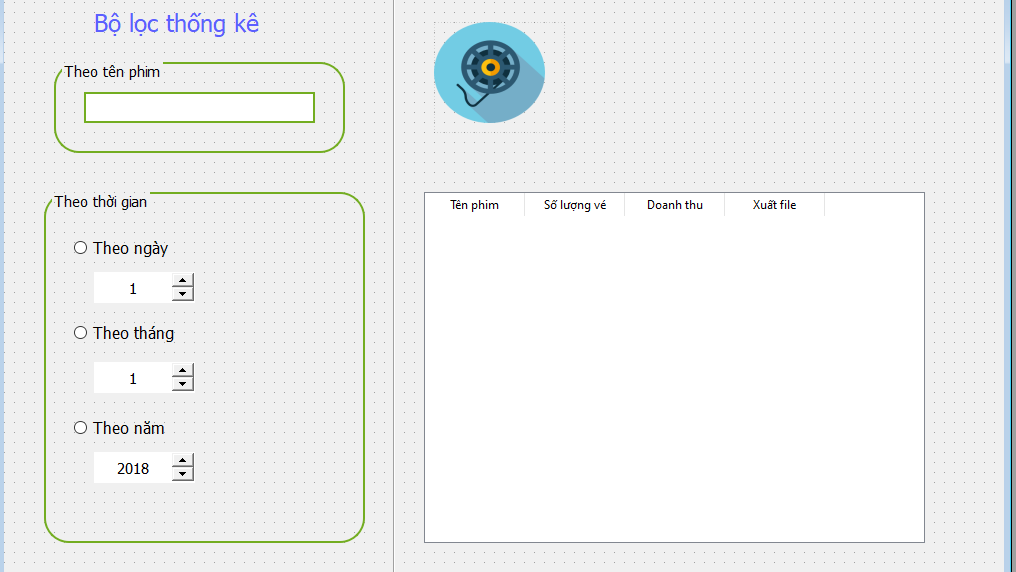
1. Thông tin phim



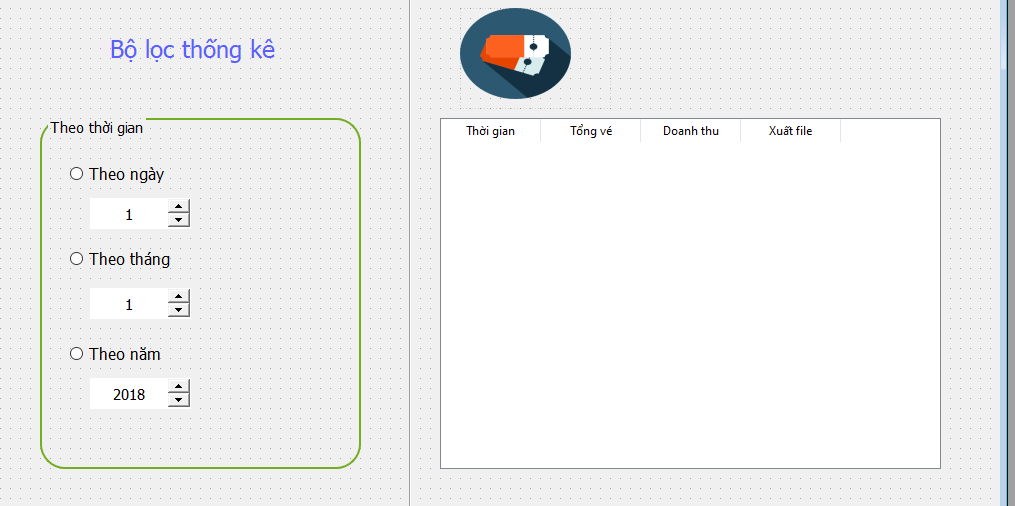
1. Thêm phim
   * + - 1. Giao diện báo cáo thống kê đối với Admin



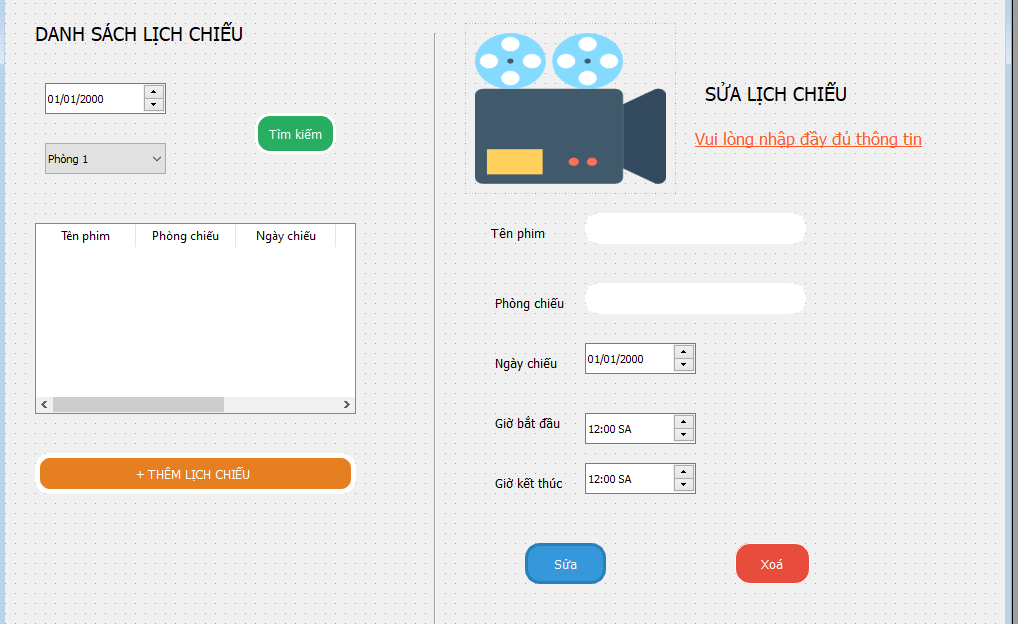
1. Thống kê doanh thu



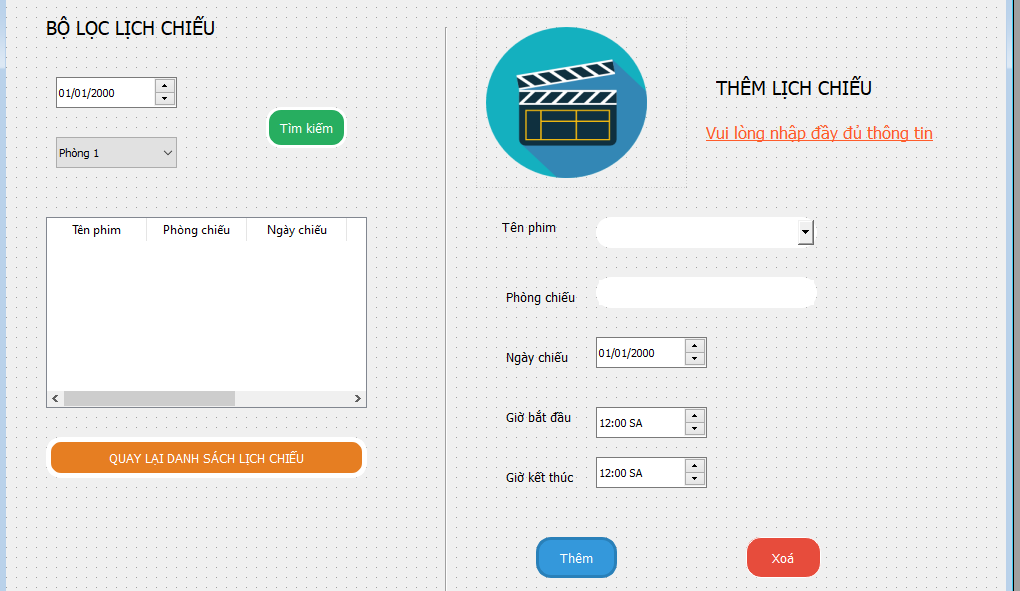
1. Thống kê phim



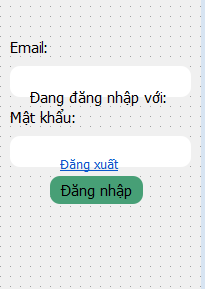
1. Thống kê vé bán
   * + - 1. Giao diện quản lý lịch chiếu đối với Admin



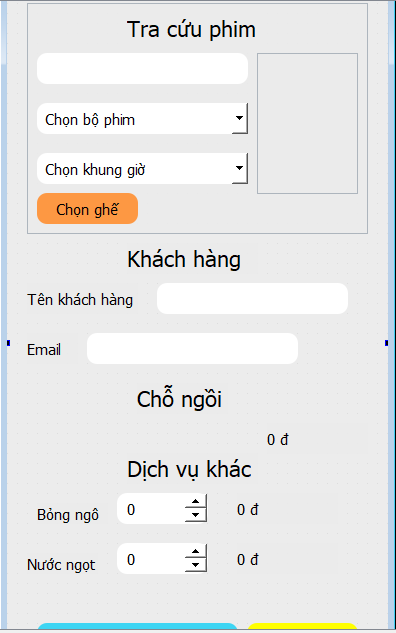
1. Thông tin lịch chiếu



1. Thêm lịch chiếu
   * + - 1. Giao diện đăng nhập đối với nhân viên



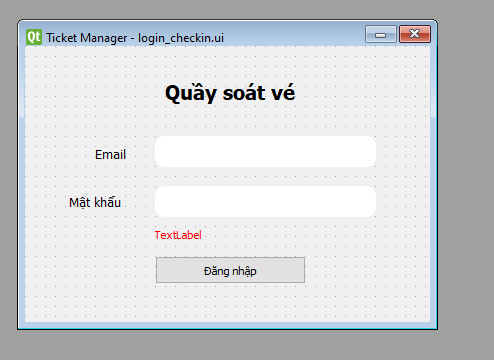
1. Đăng nhập đối với nhân viên
   * + - 1. Giao diện bán vé đối với nhân viên



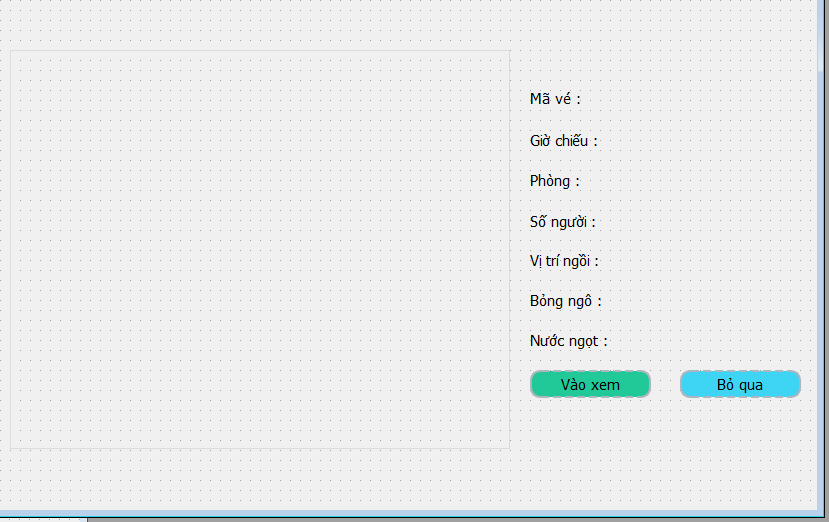
1. Đặt vé



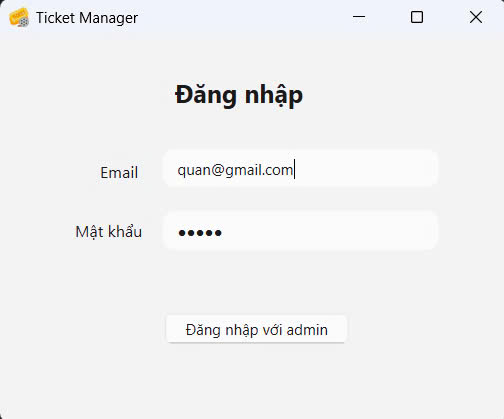
1. Hoá đơn thanh toán
   * + - 1. Giao diện đối với nhân viên soát vé



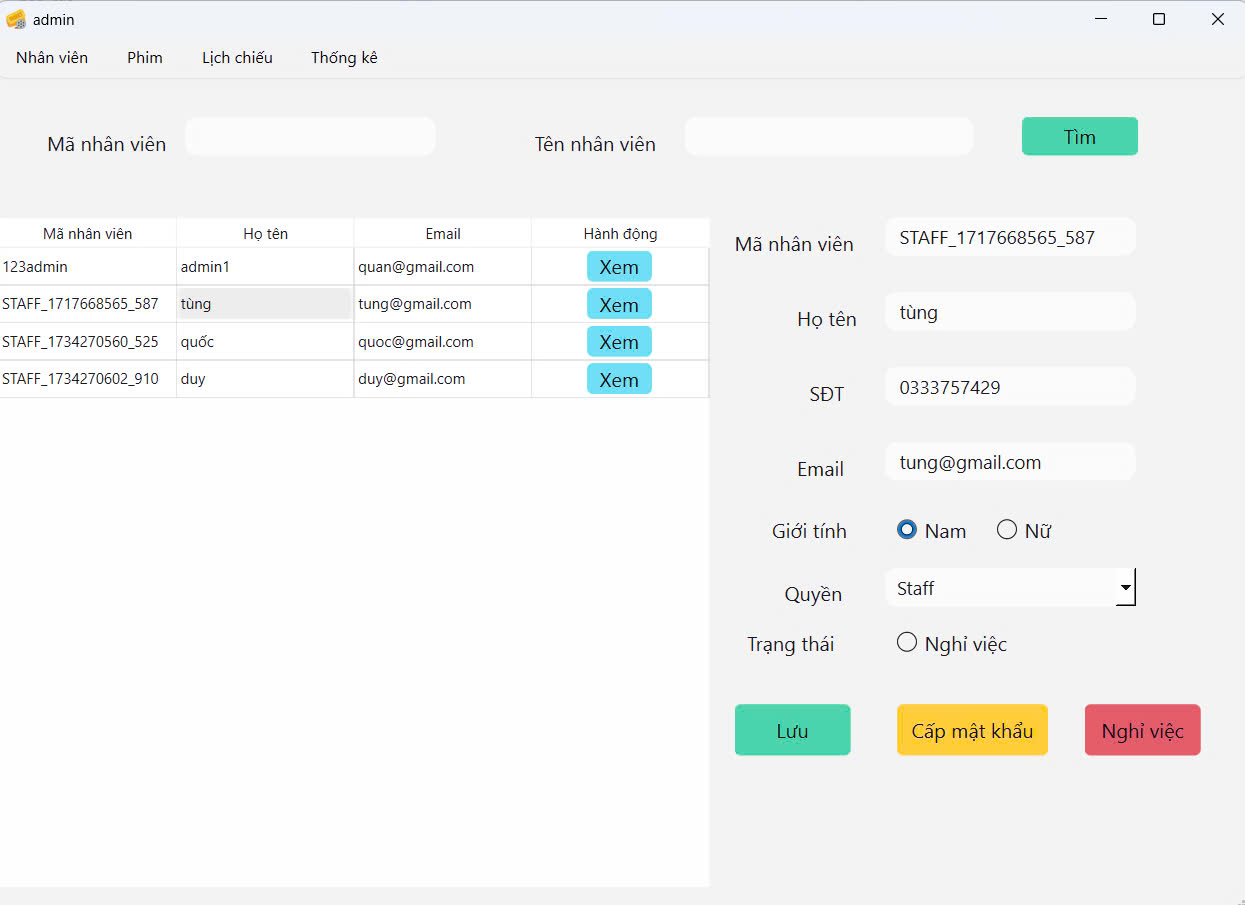
1. Đăng nhập với nhân viên soát vé



1. Checkin vé
2. KẾT QUẢ THỰC NGHIỆM CHƯƠNG TRÌNH
   1. Demo chương trình
      1. Đăng nhập với vai trò Admin



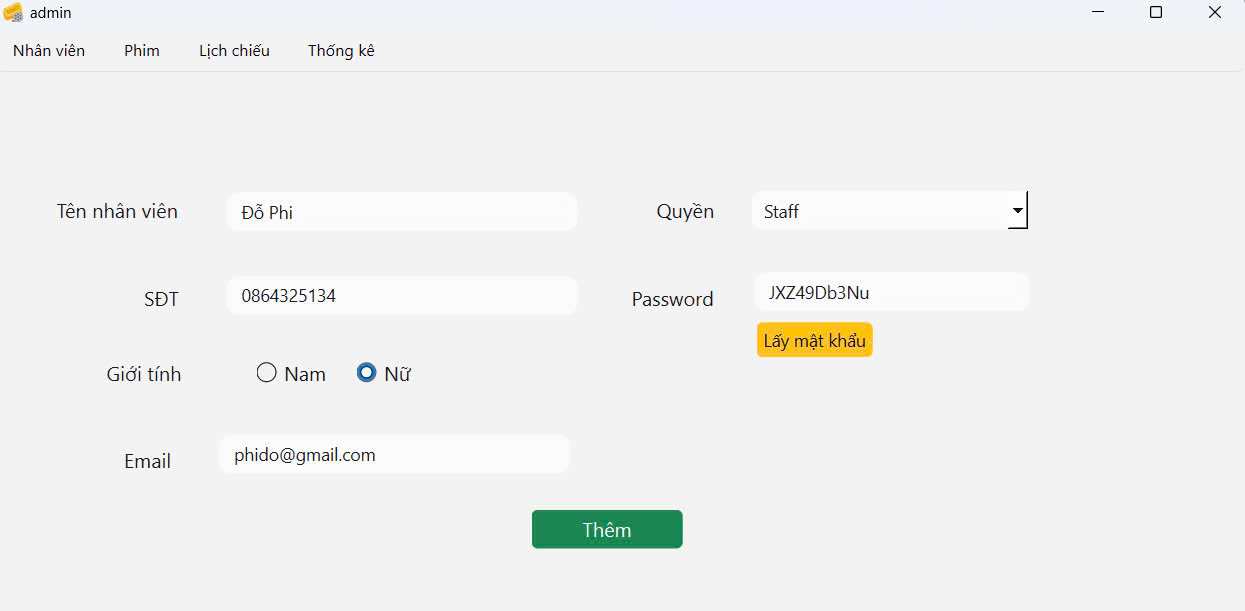
1. Đăng nhập với vai trò admin
   * 1. Trang quản lý nhân viên



1. Giao diện tất cả nhân viên

Mã nguồn

|  |
| --- |
| def findStaff(self, id, name):  if id == "" and name == "":  return Res(False, "Dữ liệu không hợp lệ")  if id != "" and name != "":  result = self.staffRepository.getStaffByIdOrName(id, name)  if len(result) == 0:  return Res(False, "Không tìm thấy thông tin")  return Res(True, data=self.staffUtil.getInfoTableFromArray(result))  if id != "":  result = self.staffRepository.getStaffById(id)  if result is None:  return Res(False, "Không tìm thấy thông tin")  return Res(True, data=[self.staffUtil.getInfoTable(result)])  if name != "":  result = self.staffRepository.getStaffByName(name)  if result is None:  return Res(False, "Không tìm thấy thông tin")  return Res(True, data=self.staffUtil.getInfoTableFromArray(result)) |

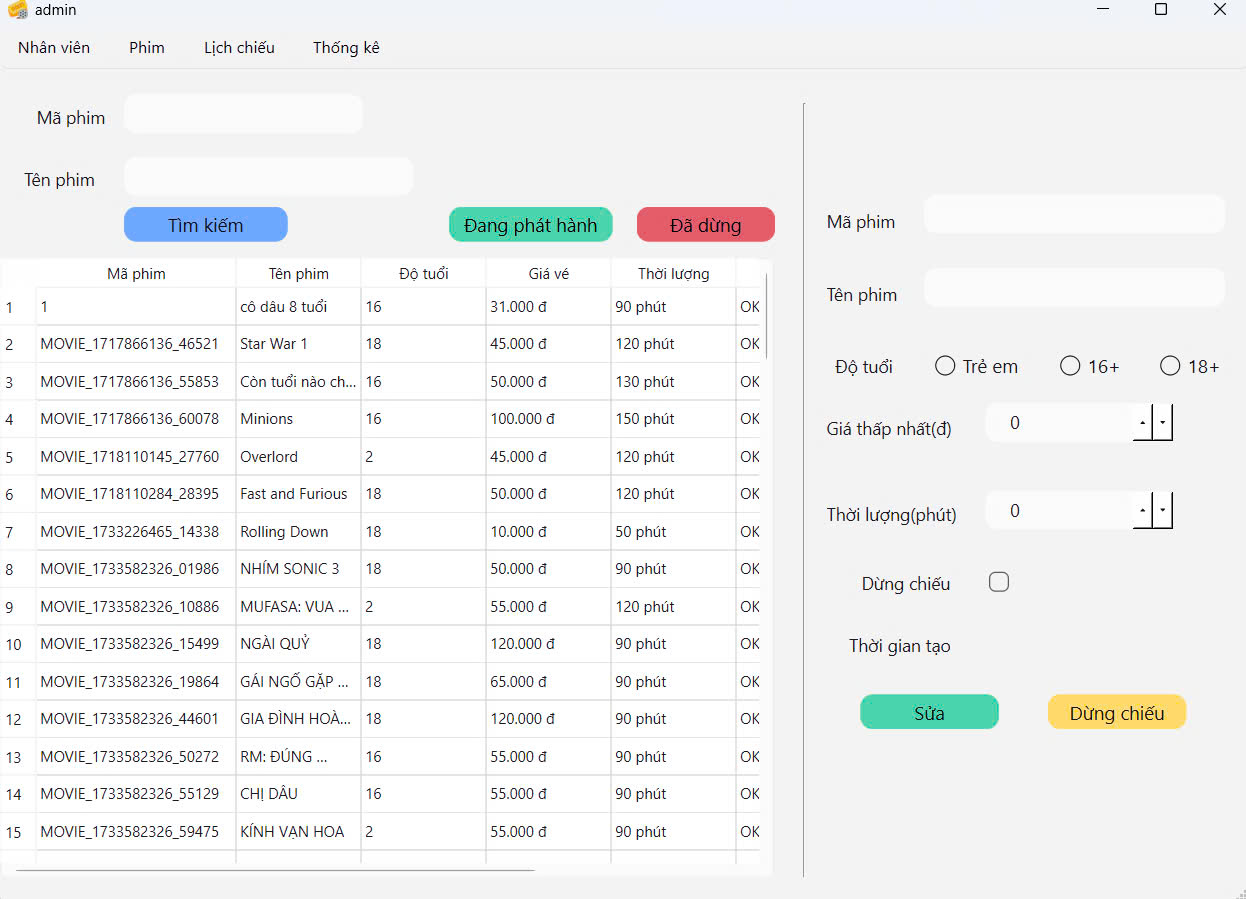


1. Thêm nhân viên

Mã nguồn

|  |
| --- |
| def addStaff(self, staff: Staff, passwordStr):  if staff.name.strip() == "" or staff.name is None:  return Res(False, "Thiếu trường tên")  if staff.sdt.strip() == "" or staff.sdt is None:  return Res(False, "Thiếu trường sđt")  insert = self.staffRepository.addStaff(staff)  if insert == 0:  return Res(False, "Thêm thất bại")  content = {  "name": staff.name,  "email": staff.email,  "password": passwordStr,  }  body = self.render.generate\_html("mail\_create\_staff.html", content)  if (  self.mail.SendMail("Cung cấp tài khoản làm việc", body, staff.email)  is False  ):  print("Lỗi gửi mail")  return Res(True) |

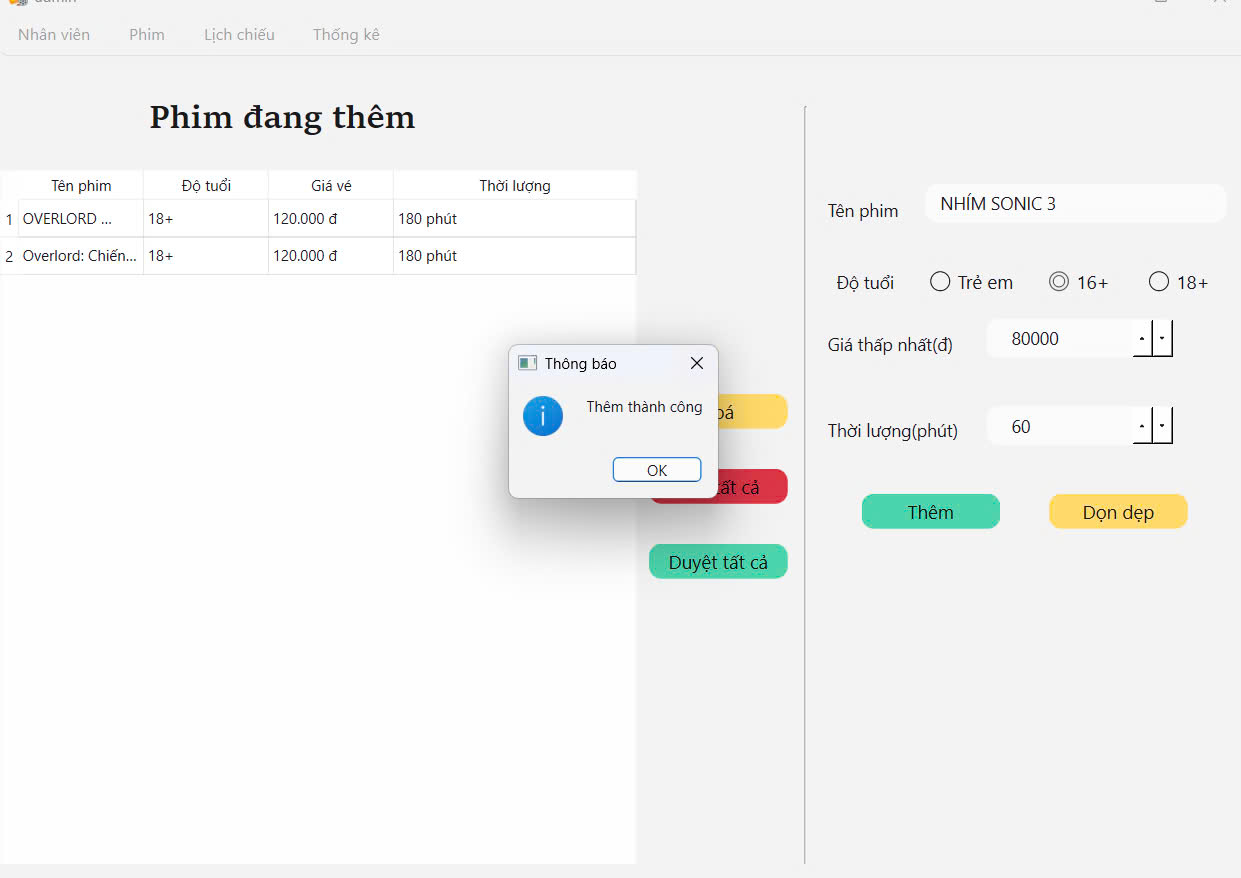
* + 1. Trang quản lý phim



1. Giao diện tất cả phim

Mã nguồn

|  |
| --- |
| def findMovie(self, id, name) -> Res:  if id.strip() == "" and name.strip() == "":  return Res(False, "Nhập thông tin phim")  data = None  if id.strip() != "" and name.strip() != "":  data = self.movieRepository.findMovieByIdAndName(id, name)  elif id.strip() != "":  data = self.movieRepository.findMovieById(id)  else:  data = self.movieRepository.findMovieByName(name)  if data is None:  return Res(False, "Lỗi truy vấn dữ liệu")  return Res(True, data=data) |

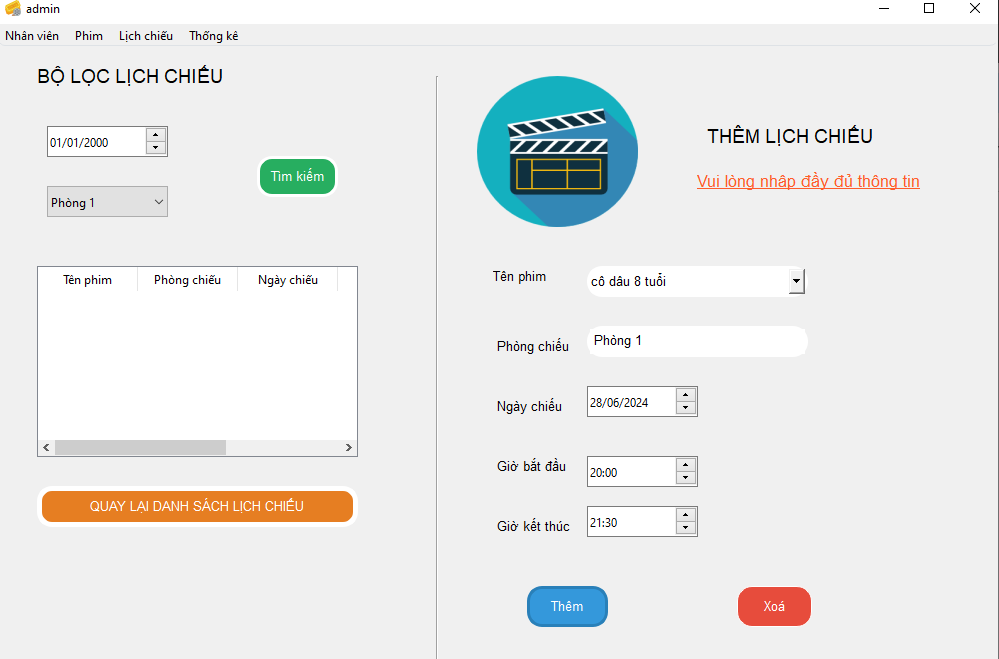


1. Thêm phim

Mã nguồn

|  |
| --- |
| def addListMovie(self, list: list[Movie]) -> bool:  try:  if len(list) == 0:  return Res(False, "Không có dữ liệu")  now = gen\_time.getNowTimestamp()  for movie in list:  id = f"MOVIE\_{int(now)}\_{gen\_number.genarateNumber(5)}"  movie.setId(id)  movie.setCreateAt(now)  movie.age = (  2  if movie.age == 1  else 16 if movie.age == 2 else 18 if movie.age == 3 else 0  )  dataInsert = [  (  movie.id,  movie.name,  movie.age,  movie.minPrice,  movie.createAt,  movie.time,  )  for movie in list  ]  rowAffect = self.movieRepository.insertListMovie(dataInsert)  if rowAffect == 0:  return Res(False, "Thêm thất bại")  return Res(True, data=rowAffect)  except (Exception, ValueError) as e:  print(e)  return Res(False, "Có lỗi xảy ra") |

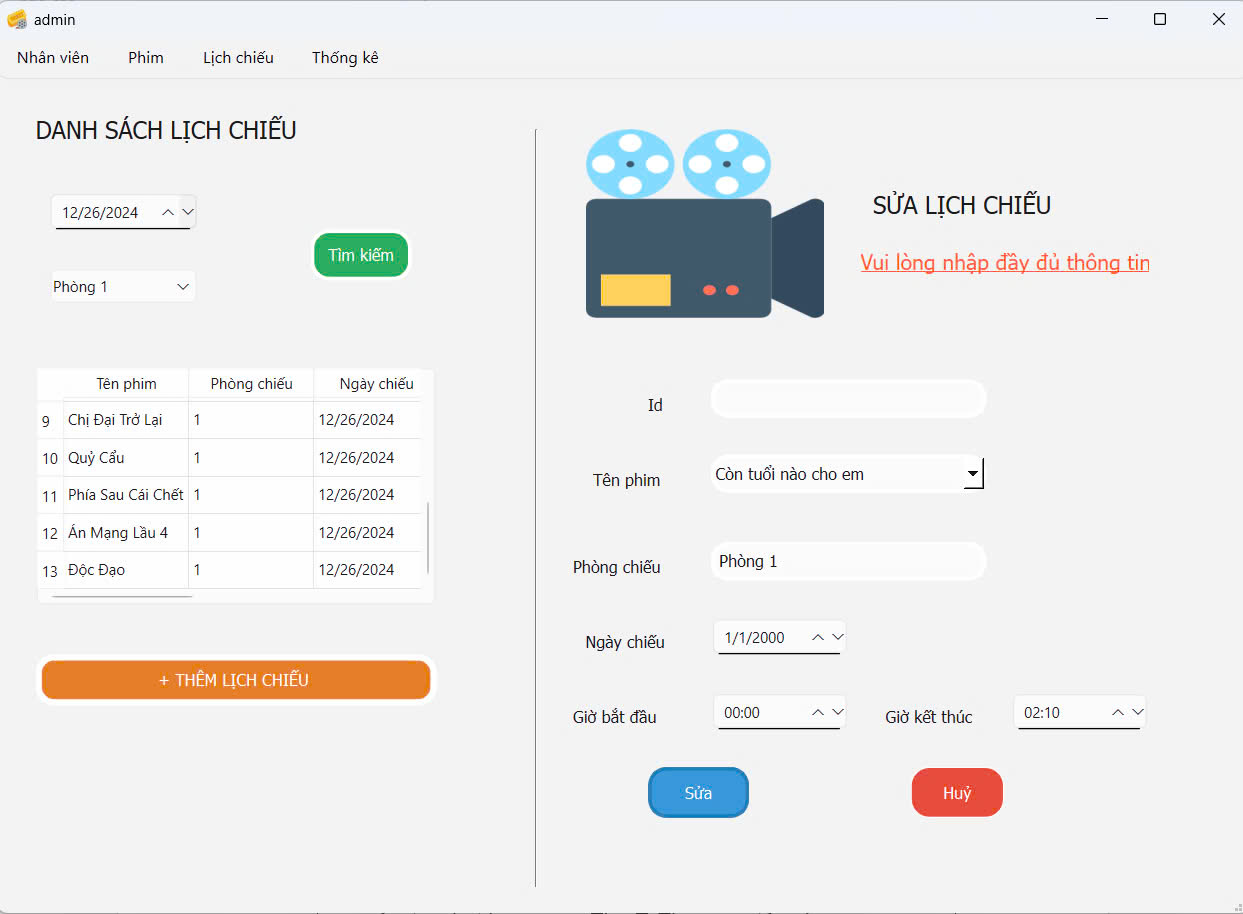
* + 1. Trang quản lý lịch chiếu



1. thêm lịch chiếu

mã nguồn

|  |
| --- |
| def add\_lichchieu(self, movie, room: str, datetime):  playAt = int(time.convertTimeToTimestamp(datetime, "%H:%M %d/%m/%Y"))  if playAt < gen\_time.getNowTimestamp():  return Res(False, "Thời gian chiếu không hợp lệ")  movies = self.movieRepository.findMovieById(movie)  if len(movies) == 0:  return Res(False, "Phim không đúng")  infoMovie = movies[0]  timeMovie = infoMovie[4]  playAt = playAt - 15 \* 60  endPlay = playAt + timeMovie \* 60  roomId = room.split()[1]  calendars = self.lichChieuRepository.get\_all\_calendar(playAt, endPlay, roomId)  if calendars is None:  return Res(False, "Lỗi truy vấn")  if len(calendars) > 0:  return Res(False, "Trùng lịch chiếu")  id = (  f"CALENDAR\_{int(gen\_time.getNowTimestamp())}\_{gen\_number.genarateNumber(3)}"  )  inserted = self.lichChieuRepository.add\_calendar(id, movie, playAt, roomId)  if inserted == 0:  return Res(False, "Thêm thất bại")  return Res(True) |

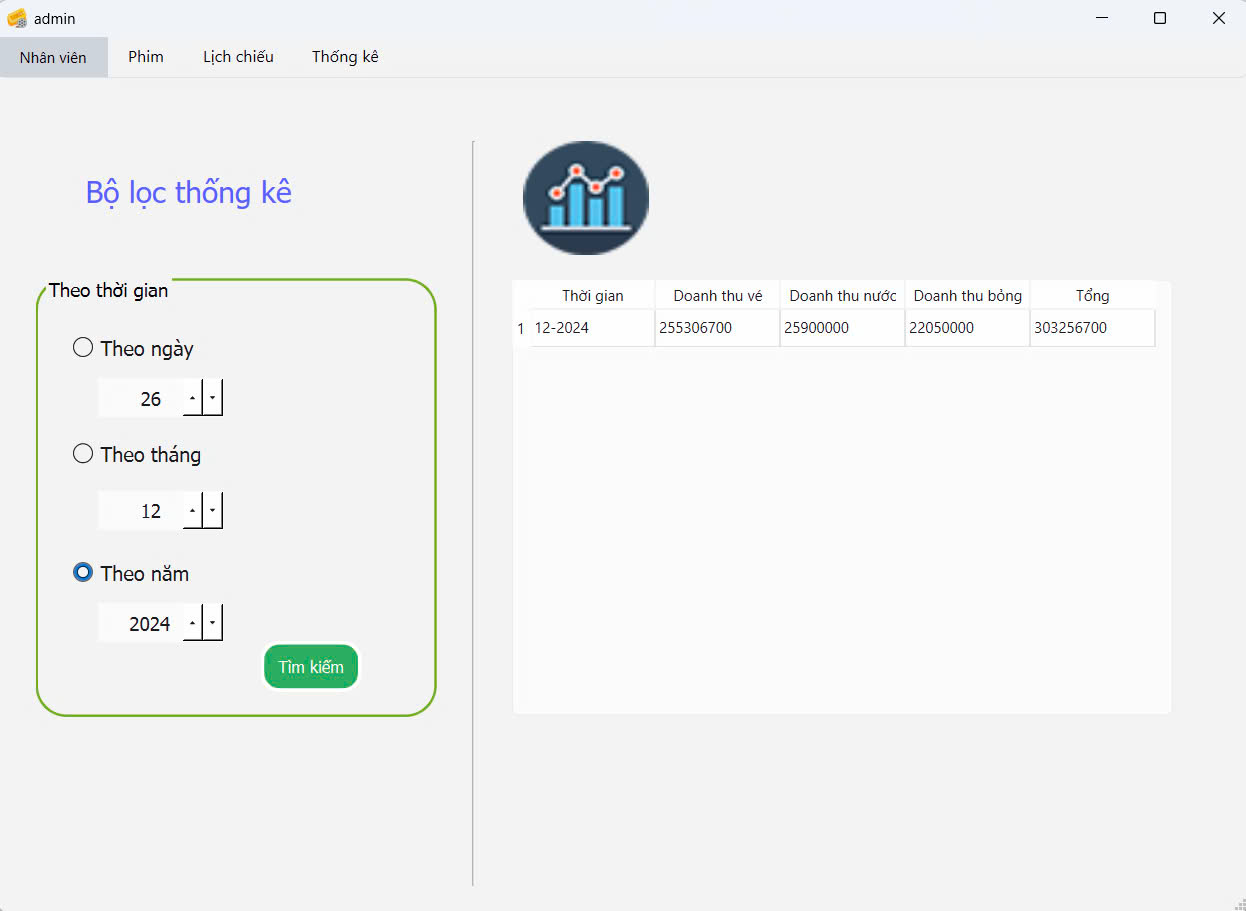


1. sửa lịch chiếu

Mã nguồn

|  |
| --- |
| def edit(self, id, movie, room, datetime):  if id.strip() == "":  return Res(False, "Id không hợp lệ")  playAt = int(time.convertTimeToTimestamp(datetime, "%H:%M %d/%m/%Y"))  if playAt < gen\_time.getNowTimestamp():  return Res(False, "Thời gian chiếu không hợp lệ")  movies = self.movieRepository.findMovieById(movie)  if len(movies) == 0:  return Res(False, "Phim không đúng")  infoMovie = movies[0]  timeMovie = infoMovie[4]  playAt = playAt - 15 \* 60  endPlay = playAt + timeMovie \* 60  roomId = room.split()[1]  calendars = self.lichChieuRepository.get\_all\_calendar(playAt, endPlay, roomId)  if calendars is None:  return Res(False, "Lỗi truy vấn")  if len(calendars) > 1:  return Res(False, "Trùng lịch chiếu")  if len(calendars) == 1:  calendar = calendars[0]  if calendar[0] != id:  return Res(False, "Trùng lịch chiếu")  updated = self.lichChieuRepository.updateCalendar(id, movie, playAt, roomId)  if updated == 0:  return Res(False, "Thay đổi thất bại")  return Res(True) |

* + 1. Báo cáo thống kê



1. Thống kê doanh thu

Mã nguồn

|  |
| --- |
| def fillter(self, style, day, month, year):  start\_timestamp = 0  end\_timestamp = 0  if style == 1:  start\_timestamp = time.convertTimeToTimestamp(  f"{day}-{month}-{year}", "%d-%m-%Y"  )  end\_timestamp = start\_timestamp + 86400  elif style == 2:  start\_timestamp = time.convertTimeToTimestamp(f"{month}-{year}", "%m-%Y")  days = calendar.monthrange(year, month)[1]  end\_timestamp = start\_timestamp + 86400 \* days  elif style == 3:  start\_timestamp = time.convertTimeToTimestamp(f"{year}", "%Y")  days = 366 if calendar.isleap(year) else 365  end\_timestamp = start\_timestamp + 86400 \* days  else:  return Res(False, "Không hỗ trợ, vui lòng nhập lại")  result = self.Repository.getInfo(start\_timestamp, end\_timestamp)  if result is None:  return Res(False, "Lỗi truy vấn dữ liệu")  if len(result) == 0:  return Res(False, "Không có dữ liệu")  ketQuaThongKe = None  if style == 1:  ketQuaThongKe = self.thongKeNgay(result)  elif style == 2:  ketQuaThongKe = self.thongKeThang(result)  elif style == 3:  ketQuaThongKe = self.thongKeNam(result)  return Res(True, data=ketQuaThongKe) |

* + 1. Trang bán vé

Mã nguồn

|  |
| --- |
| def order(self, ticket: Ticket, chairs: list, idStaff: str) -> Res:  if ticket is None:  return Res(False, "Không có thông tin")  if ticket.authen != "OK":  return Res(False, "Lỗi xác thực")  if str(idStaff).strip() == "":  return Res(False, "Nhân viên không tồn tại")  ticket.setCreateAt(gen\_time.getNowTimestamp())  insertTicket = self.ticketRepository.orderTicket(ticket, idStaff)  if insertTicket == None or insertTicket == 0:  return Res(False, "Lỗi đặt vé")  dataSeat = [(ticket.id, ticket.calendar, c) for c in chairs]  insertSeat = self.seatRepository.orderSeat(dataSeat)  if insertSeat == None or insertSeat == 0:  return Res(False, "Lỗi đặt ghế")  return Res(True) |

* 1. Dữ liệu để test
     1. Đối với bán vé

|  |  |
| --- | --- |
| **Tên trường** | **Dữ liệu đầu vào** |
| Chọn phim | Overlord 5 |
| Chọn lịch chiếu | 21h ngày 25/12/2024 |
| Chọn ghế | A1, H10,B1,A8 |
| Tên khách hàng | Thanh |
| Email | phuongthanh@gmai.com |
| Bỏng ngô | 1 |
| Nước ngọt | 2 |

* + 1. Đối với thêm phim

|  |  |
| --- | --- |
| **Tên trường** | **Dữ liệu đầu vào** |
| Tên phim | Overlord 5 |
| Độ tuổi | 18+ |
| Giá thấp nhất | 120000 |
| Thời lượng phim | 180 |

* + 1. Biểu đồ trực quan hóa

A screenshot of a computer

Description automatically generated

1. Trực quan hóa

KẾT LUẬN

Kết quả đạt được

Qua quá trình tìm hiểu và triển khai và phát triển phần mềm “Quản lý quầy bán vé xem phim”, nhóm chúng tôi đã hoàn thành các chức năng và giao diện cần thiết của một phần mềm góp phần nâng cao hiệu quả hoạt động và trải nghiệm dành cho người dùng. Dưới đây là kết quả đã đạt được:

* + - * 1. Đối với Admin
* Quản lý nhân viên: thêm nhân viên, sửa thông tin nhân viên, khoá tài khoản, tìm kiếm nhân viên.
* Quản lý lịch chiếu: thêm lịch chiếu, sửa lịch chiếu, tìm kiếm lịch chiếu.
* Quản lý phim: thêm phim, sửa thông tin phim, dừng chiếu phim, tìm kiếm phim.
* Thống kê: thống kê doanh thu, thống kê vé bán, thống kê phim.
  + - * 1. Đối với nhân viên
* Bán vé: tìm phim, lọc ghế, đặt ghế, kiểm tra, thanh toán
* Checkin: kiểm tra vé, checkin
  + - * 1. Khác
* Mã hoá mật khẩu tài khoản
* Gửi mail sau khi tạo nhân viên và sau khi đặt vé thành công

Bên cạnh những kết quả đạt được, phần mềm vẫn còn thiếu sót và chưa hoàn thiện như:

* Thống kê lịch trống

Định hướng và phát triển

Nắm bắt được tầm quan trọng của phần mềm “Quản lý quầy bán vé xem phim”, trong tương lai nhóm chúng tôi sẽ tìm hiểu và mở rộng ra thêm nhiều tính năng hơn như: thống kê sẽ phát triển thêm xuất file vào để người quản lý cần thì sẽ xuất file ra, tích hợp tính năng thanh toán bằng QR, chuyển khoản Giao diện gần gũi với người dùng hơn và thông tin được bảo mật hơn, chức năng người dùng có thể lấy lại thông tin vé của mình, mở rộng thêm chiều chức năng thống kê khác cho admin

TÀI LIỆU THAM KHẢO

[1].<https://www.studocu.com/vn/document/truong-dai-hoc-thuong-mai/hanh-vi-khach-hang/nhom-15-bai-tap-lon-python/72214414>( truy cập vào ngày 24/10/2024 lúc 21h)

[2]. <https://www.w3schools.com/python/> (truy cập vào ngày 3/11/2024 lúc 16h)