**TRƯỜNG ĐẠI HỌC BÁCH KHOA**

**KHOA CÔNG NGHỆ THÔNG TIN**

**Logo

Description automatically generated**

**ĐỒ ÁN DỰ ÁN CƠ SỞ LẬP TRÌNH**

**HỆ THỐNG ĐẶT VÉ Ở RẠP PHIM**

Sinh viên thực hiện**:**

**Hoàng Đức An Mã SV: 102220001 NHÓM: 22N15A**

**Ngô Quốc Anh Mã SV: 102220002 NHÓM: 22N15A**

Người hướng dẫn**: PGS. TS. Đặng Hoài Phương**

**Đà Nẵng, 01/2024**

MỤC LỤC

[DANH MỤC HÌNH VẼ ii](#_Toc154313385)

[MỞ ĐẦU i](#_Toc154313388)

[NỘI DUNG 1](#_Toc154313389)

[Chương 1. TỔNG QUAN VỀ ĐỀ TÀI 1](#_Toc154313390)

[1.1. Đặt vấn đề 1](#_Toc154313391)

[1.2. Ý nghĩa của đề tài 1](#_Toc154313392)

[Chương 2. CƠ SỞ LÝ THUYẾT 2](#_Toc154313393)

[2.1. Ngôn ngữ lập trình (Ngôn ngữ C++) 2](#_Toc154313394)

[2.2. Lập trình hướng đối tượng 2](#_Toc154313395)

[2.2.1. Lớp (Class) 2](#_Toc154313396)

[2.2.2. Đối tượng (Object) 3](#_Toc154313397)

[2.2.3. Tính kế thừa (Inheritance) 3](#_Toc154313398)

[2.2.4. Đa hình (Polymorphism) 4](#_Toc154313399)

[2.2.5. Đóng/gói (Encapsulation) 4](#_Toc154313400)

[Chương 3. TỔ CHỨC CHƯƠNG TRÌNH 4](#_Toc154313401)

[3.1. Phát biểu bài toán 4](#_Toc154313402)

[3.1.1. Đầu vào (Input) 4](#_Toc154313403)

[3.1.2. Đầu ra (Output) 5](#_Toc154313404)

[3.2. Các thành phần chính của chương trình 5](#_Toc154313405)

[3.2.1. Admin (Người quản lý hệ thống) 5](#_Toc154313406)

[3.2.2. User (Khách hàng sử dụng) 6](#_Toc154313407)

[3.3. Cấu trúc dữ liệu xây dựng để giải quyết 7](#_Toc154313408)

[3.3.1. Định nghĩa danh sách liên kết đôi 7](#_Toc154313409)

[3.3.2. Cài đặt Danh sách liên kết đôi 8](#_Toc154313410)

[Chương 4. PHÂN TÍCH HƯỚNG ĐỐI TƯỢNG VÀ THỰC HIỆN CHƯƠNG TRÌNH...........................................................................................10](#_Toc154313411)

[4.1. Phân tích hướng đối tượng 10](#_Toc154313412)

[4.1.1. Lớp Person (Class Person.h) 10](#_Toc154313413)

[4.1.2. Lớp Admin (Class Admin.h kế thừa class Person.h) 11](#_Toc154313414)

[4.1.3. Lớp User (class User.h kế thừa từ class Person.h) 12](#_Toc154313415)

[4.1.4. Lớp Movie (class Movie.h) 13](#_Toc154313416)

[4.1.5. Lớp Show (class Show.h) 15](#_Toc154313417)

[4.1.6. Lớp Hall (class Hall.h) 16](#_Toc154313418)

[4.1.7. Lớp HallSeat (class HallSeat.h) 16](#_Toc154313419)

[4.1.8. Lớp Date (class Date.h) 17](#_Toc154313420)

[4.1.9. Lớp Ticket (class Ticket.h) 18](#_Toc154313421)

[4.1.10. Lớp Bill (class Bill.h) 19](#_Toc154313422)

[4.2. Thực hiện chương trình 20](#_Toc154313423)

[4.2.1. Giao diện chính của chương trình 20](#_Toc154313424)

[4.2.2. Nhận xét đánh giá kết quả 26](#_Toc154313426)

[KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 26](#_Toc154313427)

[TÀI LIỆU THAM KHẢO 27](#_Toc154313428)

DANH MỤC HÌNH VẼ

*Hình 1. Class* 3

*Hình 2. Objects* 3

*Hình 3. Inheritance* 3

*Hình 4. Encapsulation* 4

*Hình 5. File MovieDB.txt* 5

*Hình 6. File ShowDB.txt* 5

*Hình 7. Các chức năng chính của Admin* 6

*Hình 8. Các chức năng chính của User* 7

*Hình 9. Danh sách liên kết đôi* 8

*Hình 10. Giao diện đăng nhập* 21

*Hình 11. Menu của khách hàng* 22

*Hình 12. Danh sách phim đang chiếu tại rạp* 22

*Hình 13. Suất chiếu của bộ phim* 23

*Hình 14. Chọn ghế ngồi* 23

*Hình 15. Hóa đơn của khách hàng.* 24

*Hình 16. Lịch sử hóa đơn của khách hàng* 24

*Hình 17. Menu của quản lý* 25

*Hình 18. Doanh thu của rạp* 25

*Hình 19. Thêm một phim mới* 25

*Hình 20. Menu chỉnh sửa thông tin phim* 26

*Hình 21. Thêm một suất chiếu mới* 26

*Hình 22. Xóa một suất chiếu* 26

*Hình 23. Menu chỉnh sủa một suất chiếu* 27

*Hình 24. Danh sách phim* 27

**LỜI CẢM ƠN**

Trong suốt quá trình hoàn thành đồ án, chúng em luôn được sự quan tâm, hướng dẫn và giúp đỡ tận tình của các thầy cô giáo trong khoa Công nghệ thông tin cùng với sự động viên, giúp đỡ của bạn bè trong lớp.

Lời đầu tiên chúng em xin được bày tỏ lòng biết ơn sâu sắc đến Ban giám hiệu Trường Đại học Bách Khoa – Đại học Đà Nẵng đã tận tình giúp đỡ chúng em suốt thời gian học tại trường.

Đặc biệt chúng em xin bày tỏ lòng biết ơn chân thành sâu sắc tới thầy giáo, PGS. TS Đặng Hoài Phương đã trực tiếp giúp đỡ, hướng dẫn chúng em hoàn thành đồ án này.

Chúng em cũng xin bày tỏ lòng biết ơn đến gia đình, người thân, bạn bè đã giúp đỡ, động viên chúng em hoàn thành đồ án.

**Chúng em xin trân trọng cảm ơn!**

MỞ ĐẦU

1. **Mục đích thực hiện đề tài**

Trong thời đại công nghệ thông tin phát triển mạnh mẽ như hiện nay, việc ứng dụng các công nghệ mới vào các lĩnh vực khác nhau của đời sống là một xu hướng không thể thiếu. Ngành công nghiệp điện ảnh là một trong những ngành có nhu cầu cao về việc sử dụng công nghệ thông tin để nâng cao chất lượng dịch vụ, tăng cường trải nghiệm của khách hàng, và tối ưu hóa hiệu suất hoạt động. Một trong những dịch vụ quan trọng của ngành điện ảnh là hệ thống đặt vé xem phim trực tuyến, cho phép người dùng có thể dễ dàng lựa chọn phim, rạp, suất chiếu, ghế ngồi, và thanh toán vé một cách nhanh chóng và tiện lợi.

Hệ thống đặt vé xem phim, hỗ trợ khách hàng đặt vé. Đây là một nhu cầu thực tế của nhiều rạp chiếu phim, nhất là những rạp có lượng khách hàng đông đảo, cần phải xử lý nhanh gọn các yêu cầu đặt vé của khách hàng. Một hệ thống đặt vé xem phim dành cho nhân viên sẽ giúp nhân viên có thể tra cứu thông tin phim, rạp, suất chiếu, ghế ngồi, và thanh toán vé một cách nhanh chóng và chính xác, giảm thiểu thời gian chờ đợi và sai sót của khách hàng, tăng cường sự hài lòng và niềm tin của khách hàng đối với rạp chiếu phim.

Với mục đích đó, nhóm chúng em đã chọn đề tài tạo hệ thống đặt vé xem phim dành cho nhân viên, khách hàng đặt trực tiếp làm đề tài đồ án cơ sở lập trình của mình. Đây là một đề tài có tính ứng dụng cao, yêu cầu nhóm chúng em phải nghiên cứu và ứng dụng công nghệ mới như cơ sở dữ liệu, bảo mật, hướng đối tượng, v.v. vào hệ thống đặt vé xem phim. Qua đó, nhóm chúng em mong muốn đóng góp cho sự phát triển của ngành công nghệ thông tin và ngành điện ảnh trong nước và quốc tế.

1. **Mục tiêu đề tài**

Hệ thống đặt vé xem phim nhằm phục vụ cho khách hàng dễ dàng đặt vé, tiết kiệm thời gian và công sức cho nhân viên và khách hàng khi đặt vé xem phim. Tăng khả năng kiểm soát và quản lý vé bán ra và vé còn lại của nhân viên cũng như sự hài lòng và trải nghiệm của khách hàng khi đi xem phim.

Xuất phát từ những mục đích trên, đồ án này đặt ra yêu cầu cho hệ thống như sau:

Đối với nhân viên:

* Cho phép nhân viên tra cứu thông tin các phim đang chiếu, lịch chiếu, giá vé, số lượng vé đã bán và doanh thu của phim.
* Cho phép nhân viên đặt vé cho khách hàng theo yêu cầu, chọn chỗ ngồi, và thanh toán bằng nhiều hình thức khác nhau.
* Cho phép nhân viên in vé cho khách hàng

Đối với khách hàng:

* Cho phép khách hàng đăng nhập vào hệ thống, quản lý thông tin cá nhân, và xem lịch sử đặt vé.
* Cho phép khách hàng tra cứu thông tin các phim đang chiếu, lịch chiếu.

1. **Đối tượng và phạm vi nghiên cứu**

Đối tượng nghiên cứu: Nghiên cứu các hệ thống đặt vé xem phim tại rạp, nghiên cứu các hệ thống đặt vé xem phim hiện có và phân tích nhu cầu của nhân viên và khách hàng.

Phạm vi nghiên cứu: Nghiên cứu quy trình mua vé xem phim tại các rạp chiếu phim, nghiên cứu và tham khảo các hệ thống đặt vé xem phim hiện có trên thị trường, như: Galaxy Cinema, Starlight, CGV Cinema, Lotte, Rio Cinema,…

1. **Phương pháp nghiên cứu**

* Nghiên cứu về nhu cầu đặt vé xem phim của khách hàng
* Nghiên cứu về hình thức đặt vé xem phim
* Lập trình và kiểm thử các chức năng của hệ thống đặt vé xem phim.
* Triển khai và đánh giá hệ thống đặt vé xem phim trên một số rạp chiếu phim thực tế.

1. **Ý nghĩa khoa học và thực tiễn**

Giúp khách hàng có thể nhanh chóng đặt vé và dễ dàng quản lý, thực hiện các công việc cho nhân viên. Ngoài ra còn giúp các rạp phim dễ dàng đăng những bộ phim rạp mình chiếu và quản lí doanh thu của rạp.

# NỘI DUNG

# TỔNG QUAN VỀ ĐỀ TÀI

## Đặt vấn đề

Vấn đề thực tế mà chúng tôi muốn giải quyết là thời gian chờ đợi và không thoải mái khi mua vé xem phim tại rạp. Việc này thường khiến khách hàng mất nhiều thời gian và không chắc chắn có được suất chiếu mong muốn. Điều này có thể dẫn đến tâm trạng chán nản khi nghĩ đến việc đi xem phim tại rạp.

Để giải quyết tình trạng này, chúng tôi đưa ra giải pháp là một hệ thống đặt vé trực tuyến nhanh chóng và tiện lợi. Điều này mang lại sự thuận tiện và tự do cho người xem, giúp họ chủ động chọn suất chiếu, ghế ngồi, và thậm chí có thể nhận biết các ưu đãi khuyến mãi.

Với giải pháp này, chúng tôi hy vọng không chỉ giúp khách hàng tiết kiệm thời gian mà còn tạo ra trải nghiệm đặt vé xem phim dễ dàng, linh hoạt và thoải mái hơn.

## Ý nghĩa của đề tài

* *Tiện lợi và linh hoạt cho người dùng:*

Hệ thống đặt vé online mang lại sự thuận tiện và linh hoạt cho người xem phim. Người dùng có thể chọn suất chiếu, ghế ngồi mà họ mong muốn từ bất kỳ đâu, mọi lúc chỉ qua một vài thao tác đơn giản trên nền tảng trực tuyến.

* *Tối ưu hóa trải nghiệm người xem:*

Bằng cách loại bỏ quy trình mua vé tại rạp và chờ đợi, hệ thống giúp tối ưu hóa trải nghiệm xem phim. Người xem không chỉ tiết kiệm thời gian mà còn có thể lựa chọn được suất chiếu và ghế ngồi ưa thích, tăng cường sự thoải mái.

* *Phát triển ngành công nghiệp giải trí:*

Đề tài đóng góp vào sự phát triển của ngành công nghiệp giải trí, đặc biệt là trong lĩnh vực điện ảnh. Việc tạo ra một giao diện đặt vé hiện đại và tiện ích giúp thu hút đông đảo khán giả và thúc đẩy doanh số bán vé.

* *Góp phần vào sự phát triển cộng đồng:*

Việc cung cấp một dịch vụ thuận lợi như đặt vé xem phim online góp phần vào sự phát triển và nâng cao chất lượng cuộc sống trong cộng đồng, thúc đẩy hoạt động giải trí và giao lưu văn hóa.

# CƠ SỞ LÝ THUYẾT

## Ngôn ngữ lập trình (Ngôn ngữ C++)

C++ là một ngôn ngữ lập trình hướng đối tượng được mở rộng từ ngôn ngữ C. Trong ngôn ngữ C không có khái niệm lập trình hướng đối tượng, và để có thể lập trình hướng đối tượng thì các nhà phát triển đã thêm chức năng này cùng với nhiều cải tiến khác vào ngôn ngữ C, để tạo ra phiên bản mở rộng của C chính là ngôn ngữ C++.

Có thể hiểu đơn giản C++ chính là ngôn ngữ C với nhiều cải tiến khác nhau, bao gồm cả lập trình hướng đối tượng. Mặc dù được cải tiến từ C, nhưng nó vẫn có khả năng sử dụng các phương thức lập trình thủ tục như ngôn ngữ C truyền thống. Và một điều tuyệt vời nữa là, do C++ và C có tính tương hỗ với nhau, nên chúng ta cũng có thể sử dụng đồng thời cả C++ và C khi viết một chương trình máy tính.Ngôn ngữ C++ là ngôn ngữ lập trình hướng đối tượng do đó có lợi đối với các ứng dụng quản lý. Nhờ vào 4 tính chất chính là kế thừa, đa hình, đóng gói, trừu tượng của ngôn ngữ này ta không chỉ quản lý các dữ liệu tốt hơn mà còn đảmbảo tính bảo mật dữ liệu cao cho chương trình. Bên cạnh đó, ngôn ngữ C++ đã được ra đời từ lâu nên khả năng tương thích, hỗ trợ đa nền tảng của nó tốt hơn so với các ngôn ngữ lập trình khác.

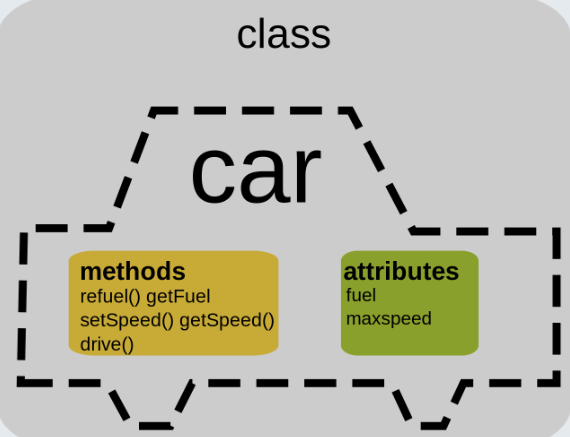
## Lập trình hướng đối tượng

Lập trình hướng đối tượng (OOP - Object-Oriented Programming) là một phương pháp lập trình mà ở đó, mọi thực thể trong chương trình được xem xét như là các đối tượng. Các đối tượng này có thể chứa dữ liệu trong các thuộc tính (attributes) và có thể thực hiện các hành động thông qua các phương thức (methods). OOP tập trung vào việc tổ chức mã nguồn một cách cấu trúc và linh hoạt.

Các khái niệm quan trọng của OOP:

### Lớp (Class)

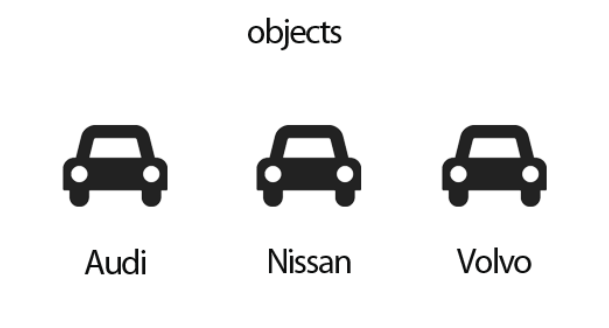
Một lớp là một kiểu dữ liệu bao gồm các thuộc tính và các phương thức được định nghĩa từ trước. Đây là sự trừu tượng hóa của đối tượng. Khác với kiểu dữ liệu thông thường, một lớp là một đơn vị (trừu tượng) bao gồm sự kết hợp giữa các phương thức và các thuộc tính. Hiểu nôm na hơn là các đối tượng có các đặc tính tương tự nhau được gom lại thành một lớp đối tượng.



*Hình 1. Class*

### Đối tượng (Object)

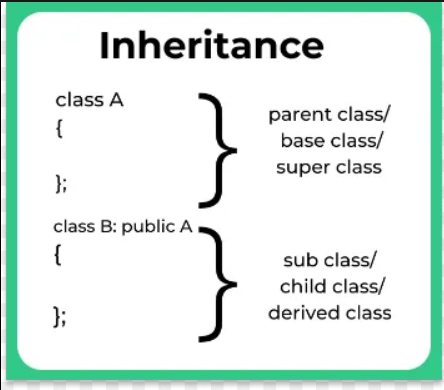
Là một thể hiện cụ thể của một lớp. Đối tượng có thể được tạo ra từ lớp và sẽ chứa các đặc điểm cụ thể (dữ liệu) và có thể thực hiện các hành động (phương thức).



*Hình 2. Objects*

### Tính kế thừa (Inheritance)

Cho phép một lớp sử dụng lại các thuộc tính và phương thức của một lớp khác. Lớp kế thừa được gọi là lớp con, và lớp cung cấp các thuộc tính và phương thức được kế thừa được gọi là lớp cha.



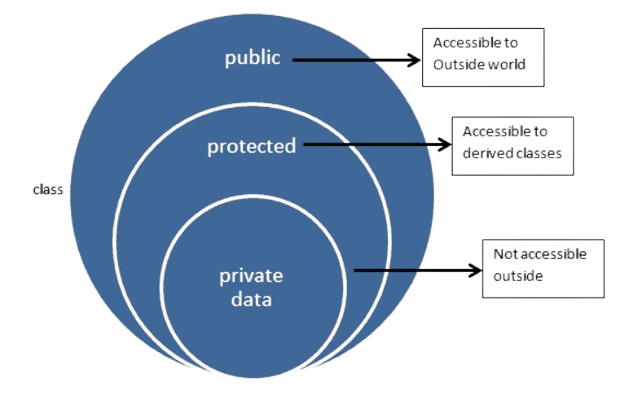
*Hình 3. Inheritance*

### Đa hình (Polymorphism)

Cho phép một đối tượng được xử lý như một đối tượng của lớp cha, mặc dù đối tượng thực sự là của một lớp con. Điều này giúp tăng tính linh hoạt của mã nguồn.

### Đóng/gói (Encapsulation)

Bảo vệ sự ẩn thông tin bằng cách đặt các thuộc tính và phương thức trong một lớp và quyết định xem chúng có thể truy cập từ bên ngoài lớp hay không.



*Hình 4. Encapsulation*

# TỔ CHỨC CHƯƠNG TRÌNH

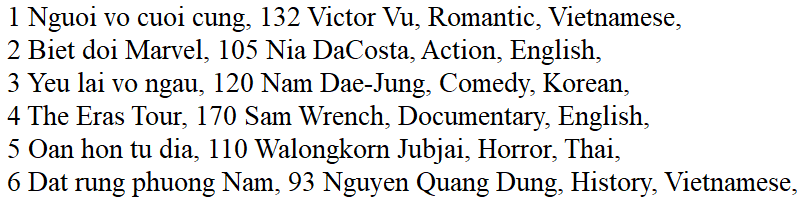
## Phát biểu bài toán

Bài toán cần giải quyết vấn đề đặt vé của khách hàng và các chức năng cơ bản của quản lý rạp chiếu phim.

### Đầu vào (Input)

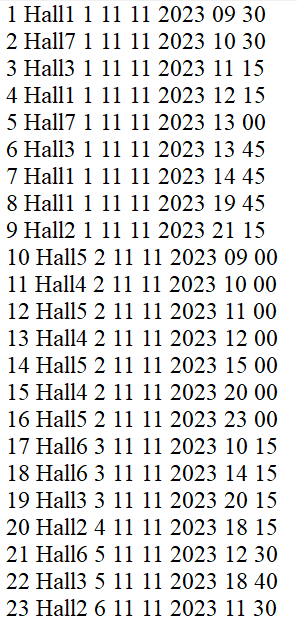
Đầu vào của chương trình gồm hai file dữ liệu:

* File lưu trữ dữ liệu của các bộ phim sẽ công chiếu (Cấu trúc của file: Tên phim, thời lượng, đạo diễn, thể loại và ngôn ngữ)



*Hình 5. File MovieDB.txt*

* File lưu trữ dữ liệu của các xuất chiều của các bộ phim sẽ công chiếu (Cấu trúc của file: ID xuất chiếu, phòng chiếu, ID phim, ngày, tháng, năm, giờ và phút chiều phim)



*Hình 6. File ShowDB.txt*

### Đầu ra (Output)

Đầu ra của chương trình sẽ được lưu lại vào các file input để phục vụ cho những lần thực hiện tiếp theo

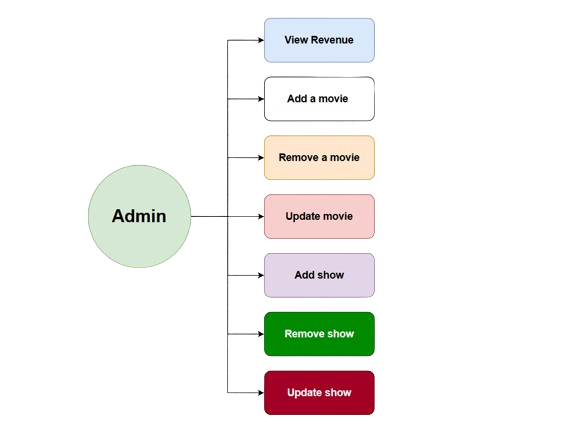
## Các thành phần chính của chương trình

### Admin (Người quản lý hệ thống)

Trong hệ thống đặt vé xem phim này, Admin đóng vai trò quan trọng, đảm nhận nhiều nhiệm vụ quan trọng để duy trì và quản lý hoạt động suôn sẻ, hiệu quả. Admin là người quản lý cao cấp, có quyền truy cập vào các chức năng quan trọng như xem doanh thu, thêm, sửa, và xóa thông tin về phim và lịch chiếu.

Các chức năng chính của Admin:

* Xem doanh thu của rạp
* Thêm một bộ phim/khung giờ để khởi chiếu
* Xóa một bộ phim/khung giờ khỏi rạp
* Cập nhật phim/khung giờ nếu có sai sót



*Hình 7. Các chức năng chính của Admin*

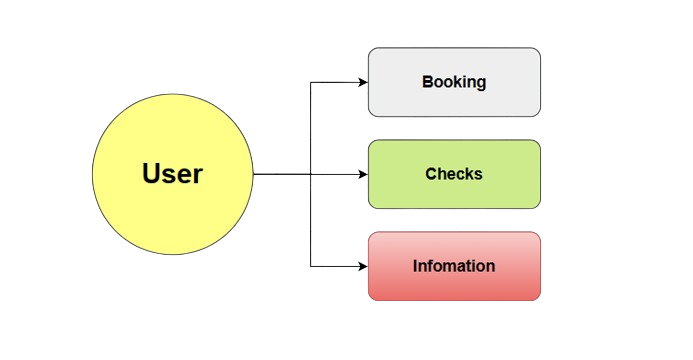
Admin chính là người duy trì sự liên kết giữa doanh nghiệp và người xem, đảm bảo rằng mọi khía cạnh của hệ thống đặt vé xem phim hoạt động một cách suôn sẻ và hiệu quả.

### User (Khách hàng sử dụng)

Người dùng đóng vai trò quan trọng trong hệ thống đặt vé xem phim, là nhóm chủ động tham gia và tận hưởng trải nghiệm giải trí. Họ là những cá nhân đa dạng, từ những người yêu thích điện ảnh đến những người muốn dành những khoảnh khắc giải trí chất lượng cùng gia đình và bạn bè.

Các chức năng chính của User:

* Đặt vé xem phim
* Xem hóa đơn đã đặt
* Xem thông tin khách hàng



*Hình 8. Các chức năng chính của User*

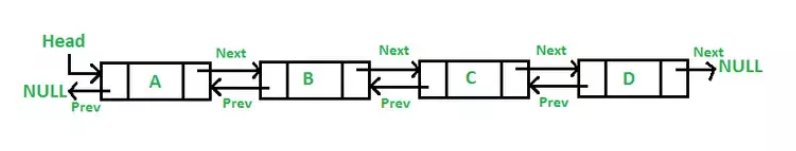
Khách hàng mong đợi có khả năng chia sẻ trải nghiệm, đánh giá phim và nhận phản hồi nhanh chóng từ hệ thống, tạo nên một cộng đồng người yêu điện ảnh tích cực và tương tác. Trong tất cả, khách hàng không chỉ là đối tượng, mà là động lực quan trọng để hệ thống đặt vé xem phim không ngừng phát triển và hoàn thiện.

## Cấu trúc dữ liệu xây dựng để giải quyết

Cấu trúc dữ liệu để lưu trữ và giải quyết bài toán là Danh sách liên kết đôi (Doubly Linked List). Các phim, các xuất chiếu sẽ được luu vào Danh sách liên kết đôi.

### Định nghĩa danh sách liên kết đôi

Danh sách liên kết đôi (Doubly Linked List): Mỗi node trong danh sách liên kết đôi gồm có previous pointer, data và next pointer, previous pointer trỏ tới phần tử đứng trước, next pointer trỏ tới phần tử phía sau. Nếu previous pointer trỏ tới NULL thì có nghĩa node đó là node đứng đầu tiên trong danh sách liên kết đơn. Và tương tự với danh sách liên kết, next pointer trỏ tới NULL thì đó là node cuối trong danh sách liên kết đơn.



*Hình 9. Danh sách liên kết đôi*

Không giống như danh sách liên kết đơn, xóa 1 node trong list cần biết node đứng trước nó. Thì trong danh sách liên kết đôi, có thể xóa 1 node trong list mà không cần dựa vào node đứng trước nó mà chỉ cần dựa vào node đứng sau nó.

Với mỗi node, có thêm một con trỏ previous pointer dẫn tới sẽ cần thêm bộ nhớ.

### Cài đặt Danh sách liên kết đôi

1. *Cài đặt kiểu dữ liệu của Danh sách liên kết*

template<typename T>

struct Node

{

    T data;

    Node\* prev;

    Node\* next;

};

1. *Hàm thêm một node*

        void Insert(T new\_data)

        {

            Node<T>\* new\_node = new Node<T>;

            new\_node->data = new\_data;

            new\_node->prev = nullptr;

            new\_node->next = this->head;

            if (head != nullptr)

            {

                this->head->prev = new\_node;

            }

            this->head = new\_node;

        }

1. *Hàm in các node*

void Display()

        {

            Node<T>\* temp = head;

            while (temp != nullptr && temp->next != nullptr)

            {

                temp = temp->next;

            }

            while (temp != nullptr)

            {

                cout << temp->data;

                temp = temp->prev;

            }

        }

1. *Hàm xóa một node*

void Remove(T value)

        {

            Node<T>\* current = head;

            while (current != nullptr)

            {

                if (current->data == value)

                {

                    if (current == head)

                    {

                        head = current->next;

                        if (head != nullptr)

                        {

                            head->prev = nullptr;

                        }

                    }

                    else

                    {

                        current->prev->next = current->next;

                        if (current->next != nullptr)

                        {

                            current->next->prev = current->prev;

                        }

                    }

                    delete current;

                    return;

                }

                current = current->next;

            }

        }

};

# PHÂN TÍCH HƯỚNG ĐỐI TƯỢNG VÀ THỰC HIỆN CHƯƠNG TRÌNH

## Phân tích hướng đối tượng

Hệ thống chương trình gồm có 10 lớp:

* Lớp Person
* Lớp Admin (Kế thừa lớp Person)
* Lớp User (Kế thừa lớp Person)
* Lớp Movie
* Lớp Show
* Lớp Hall
* Lớp HallSeat
* Lớp Date
* Lớp Ticket
* Lớp Bill

### Lớp Person (Class Person.h)

* *Mô tả:* Dùng để quản lý các thông tin cơ bản của Admin và User gồm ID, tên, ngày sinh, giới tính, số điện thoại.
* *Các thuộc tính của lớp Person:*
* ID (integer): mã số
* Name (string): tên
* Birth (date): ngày tháng năm sinh
* Gender (string): giới tính
* PhoneNumber (string): số điện thoại
* Ngoài ra lớp Person còn có các thuộc tính là danh sách phim, danh sách show và danh sách vé đã đặt
* ListM (DoublyLinkedList<Movie>): danh sách các bộ phim
* ListS (DoublyLinkedList<Show>): danh sách các suất chiếu
* ListT (DoublyLinkedList<Ticket>): danh sách các vé đã đặt
* *Các phương thức của Person gồm có:*

+ Hàm khởi tạo, hàm hủy

+ Hàm ListMovie() để liệt kê các bộ phim

+ Hàm ShowInfo() để xem thông tin cơ bản

+ Các hàm Get() tương ứng từng thuộc tính

class Person

{

    private:

        static int nextid;

        int UID;

        string Account;

        string Password;

        bool Admin;

        string Name;

        Date Birth;

        string Gender;

        string PhoneNumber;

    public:

        DoublyLinkedList<Movie> \*ListM;

        DoublyLinkedList<Show> \*ListS;

        DoublyLinkedList<Ticket> ListT;

    public:

        Person();

        Person(string, Date, string, string);

        ~Person();

        void ListMovie(int);

        void ListShow(int,const Movie&);

        void Booking(int);

        string getName();

        Date getBirth();

        string getAddress();

        string getGender();

        string getPhoneNumber();

        string getEmail();

        void ShowInfo();

};

### Lớp Admin (Class Admin.h kế thừa class Person.h)

* *Mô tả*: Là các chức năng của quản lý rạp phim. Như đã đề cập ở trên, Admin gồm có 7 chức năng chính (cũng là 7 phương thức của Admin.h):

+ Hàm khởi tạo và hàm hủy.

+ Hàm liệt kê Menu() dành cho quản lý.

+ Hàm ListRevenue() dùng để xem doanh thu của rạp.

+ Hàm AddMovie() dùng để thêm một bộ phim vào danh sách để khởi chiếu.

+ Hàm RemoveMovie() dùng để xóa một bộ phim ra khỏi danh sách (sẽ xóa toàn bộ suất chiếu của bộ phim).

+ Hàm EditMovie() dùng để cập nhật, chỉnh sửa những sai sót của bộ phim như tựa phim, thời thượng, đạo diễn, thể loại,...

+ Hàm AddShow() dùng để thêm một suất chiếu cho một bộ phim đã chọn.

+ Hàm RemoveShow() dùng để để xóa một suất chiếu của một bộ phim đã chọn.

+ Hàm EditShow() dùng để cập nhật, chỉnh sửa một suất chiếu của một bộ phim đã chọn như rạp chiếu, ngày chiếu, giờ chiếu.

class Admin : public Person

{

    public:

        Admin();

        ~Admin();

        void ListRevenue();

        void AddMovie();

        void AddShow();

        void RemoveMovie();

        void RemoveShow();

        void EditMovie();

        void EditShow();

        void SaveChange();

        void Menu(int);

        void ShowMenu();

};

### Lớp User (class User.h kế thừa từ class Person.h)

* *Mô tả*: Dành cho khách hàng đơn giản đặt vé xem phim. Khách hàng sẽ chứa thuộc tính là danh sách các hóa đơn đã đặt.

B (DoublyLinkedList<Bill>)

* *Các chức năng (phương thức) chính của khách hàng (User)*

+ Hàm khởi tạo và hàm hủy.

+ Hàm liệt kê Menu() dành cho khách hàng.

+ Hàm Booking(int) dùng cho khách hàng đặt vé xem phim bao gồm: Chọn phim, chọn suất chiếu của phim, tổng vé đặt và chọn ghế.

+ Hàm Bills() xem các vé đã đặt.

+ Hàm ShowInfo() để xem thông tin khách hàng.

class User : public Person

{

    public:

        DoublyLinkedList<Bill> ListB;

    public:

        User();

        ~User();

        void CreateTicket(Ticket&, Show&);

        void Menu(int);

        void ShowMenu();

        void Booking();

        void ListRevenue();

        void AddMovie();

        void AddShow();

};

### Lớp Movie (class Movie.h)

* *Mô tả*: Dùng để quản lý các thông tin cơ bản của bộ phim bao gồm ID phim, tên phim, thời lượng, đạo diễn, thể loại và ngôn ngữ của phim.
* Ngoài ra lớp Movie còn chứa các suất chiếu của phim, tổng các vé đã bán của phim
* ListSM (DoublyLinkedList<Show>): danh sách các suất chiếu của từng phim
* TotalTicket (int): tổng vé đã bán của từng phim.
* *Các thuộc tính của lớp Movie*:
* FilmID (int): ID của phim, để dễ dàng phân biệt các phim khác.
* Tilte (string): tựa đề của bộ phim.
* Duration (int): thời lượng của phim, đơn vị phút.
* Director (string): tên đạo diễn bộ phim.
* Genre (string): thể loại bộ phim.
* Language (string): ngôn ngữ của bộ phim.
* *Các phương thức của lớp Movie bao gồm:*

+ Hàm khởi tạo và hàm hủy.

+ Hàm đa năng toán tử nhập/xuất.

+ Hàm so sánh ID phim.

+ Các hàm Get(), Set() tương ứng từng thuộc tính.

+ Hàm EnterMovie() để nhập các thuộc tính phim từ bàn phím (dành cho quản lý).

+ Hàm Print(ofstream&) để xuất phim ra file dữ liệu.

class Movie

{

    protected:

        int FilmID;

        string Title;

        int Duration;

        string Director;

        string Genre;

        string Language;

    public:

        DoublyLinkedList<Show> ListSM;

        int TotalTicket;

        Movie();

        ~Movie();

        friend istream& operator>>(istream&, Movie&);

        friend ostream& operator<<(ostream&, const Movie&);

        bool operator==(const Movie&);

        int GetFilmID();

        string GetTitle();

        void SetFilmID(int);

        void SetTitle(string);

        void SetDuration(int);

        void SetDirector(string);

        void SetGenre(string);

        void SetLanguage(string);

        void EnterMovie();

        void Print(ofstream&);

};

### Lớp Show (class Show.h)

* *Mô tả*: Dùng để quản lý các suất chiếu của các bộ phim như ID, ngày, tháng năm chiếu, giờ phút chiếu phim, tổng số vé đã đặt của suất đó, màn hình nào sẽ chiếu suất đó.
* *Các thuộc tính của lớp Show:*
* TimeStart (Date): ngày tháng năm, giờ phút bắt đầu suất chiếu.
* Hall (Hall): rạp sẽ chiếu phim
* CreatedTicket (int): vé đã đặt.
* SID (int): ID của suất chiếu.
* MovieID (int): ID phim của suất chiếu.
* *Các phương thức của lớp Show:*

+ Hàm khởi tạo và hàm hủy.

+ Hàm đa năng hóa toán tử nhập/xuất.

+ Hàm Get(), Set() tương ứng từng thuộc tính.

+ Hàm đa năng hóa toán tử so sánh (==).

+ Hàm EnterShow() để nhập các thuộc tính show từ bàn phím (dành cho quản lý).

+ Hàm Print(ofstream&) để xuất show ra file dữ liệu.

class Show

{

    public:

        Date TimeStart;

        Hall h;

        int CreatedTicket;

    private:

        int SID;

        int MovieID;

    public:

        Show();

        ~Show();

        friend istream& operator>>(istream&, Show&);

        friend ostream& operator<<(ostream&, Show&);

        int getMovieID();

        int getSID();

        string getHall();

        int getTotalSeat();

        void setMovieID(int);

        void setSID(const int&);

        bool operator==(const Show&);

        void EnterShow();

        void Print(ofstream&);

};

### Lớp Hall (class Hall.h)

* *Mô tả:* Là lớp để quản lý các đối tượng phòng chiếu phim. Một rạp chiếu phim sẽ có nhiều phòng chiếu phim để chiếu được đa dạng phim cùng một thời điểm.
* *Các thuộc tính của lớp Hall:*
* HallName (string): tên của phòng chiếu phim.
* hs (HallSeat): mỗi một phòng chiếu cố định sẽ có 100 chỗ ngồi. Vì vậy bọn em sử dụng mảng hai chiều gồm 10 hàng và 10 cột để quản lý chỗ ngồi.
* *Các phương thức của lớp Hall bao gồm:*

+ Hàm khởi tạo và hàm hủy.

+ Hàm ShowSeat() để in ra 100 ghế của phòng chiếu theo định dạng hàng và cột để khách hàng có thể hình dung được cấu tạo của phòng chiếu, chọn ghế phù hợp (xa, gần màn hình). Biết được ghế nào đã được đặt, ghế nào còn trống.

+ Hàm đa năng hóa toán tử nhập.

class Hall

{

    public:

        string HallName;

        HallSeat hs[10][10];

    public:

        Hall();

        ~Hall();

        void ShowSeats();

        friend istream& operator>>(istream&, Hall&);

};

### Lớp HallSeat (class HallSeat.h)

* *Mô tả*: Quản lý trạng thái, ví trí của một chỗ ngồi (ghế) của phòng chiếu.
* *Các thuộc tính của lớp HallSeat:*
* SeatStatus (bool): trạng thái của ghế, còn trống hay đã được đặt.
* Hall\_Col (int): cột chứa ghế.
* Hall\_Row (int): hàng chứa ghế
* *Các phương thức của lớp HallSeat:*

+ Hàm khởi tạo và hàm hủy.

+ Hàm đa năng hóa toán tử xuất ghế (<<).

+ Hàm đa năng hóa toán tử gán (=).

+ Hàm ChangeSeatStatus() thay đổi trạng thái của ghế khi có người đặt.

+ Hàm SetCol(int), SetRow(int) để tìm vị trí của ghế.

+ Hàm GetSeatStatus() để xem trạng thái của ghế.

class HallSeat

{

    protected:

        bool SeatStatus;

        int Hall\_Col;

        int Hall\_Row;

    public:

        HallSeat();

        ~HallSeat();

        friend ostream& operator<<(ostream&, const HallSeat&);

        const HallSeat& operator=(const HallSeat&);

        void ChangeSeatStatus();

        void SetCol(int);

        void SetRow(int);

        bool GetStatusSeat();

};

### Lớp Date (class Date.h)

* *Mô tả:* Là một kiểu dữ liệu tự định nghĩa để quản lý thời gian chiếu phim bao gồm: ngày tháng năm (dd/mm/yyyy), giờ phút (hh:mm) chiếu phim.
* *Các thuộc tính của lớp Date*:
* Day (int): ngày
* Month (int): tháng
* Year (int): năm
* Hour (int): giờ
* Minute (int): phút
* *Các phương thức của lớp Date:*

+ Hàm khởi tạo và hàm hủy.

+ Hàm Get...(), Set...() tương ứng từng thuộc tính.

+ Hàm đa năng hóa toán tử nhập >>, xuất << (dưới định dạng dd/mm/yyyy, hh:mm)

class Date

{

    private:

        int Day;

        int Month;

        int Year;

        int Hour;

        int Minute;

    public:

        Date();

        ~Date();

        void setDay();

        void setMonth();

        void setYear();

        void setHour();

        void setMinute();

        int getDay();

        int getMonth();

        int getYear();

        int getHour();

        int getMinute();

        friend istream& operator>>(istream&, Date&);

        friend ostream& operator<<(ostream&, const Date&);

};

### Lớp Ticket (class Ticket.h)

* *Mô tả*: Là lớp quản lý các vé đã đặt của khách hàng. Bao gồm: ID của vé, giá vé, ghế đã đặt, phim, suất chiếu đã đặt,...
* *Các thuộc tính của lớp Ticket:*
* TicketID (int): ID của vé.
* NextID (static int): ID của vé tiếp theo được tăng lên khi được tạo ra, tránh việc trùng ID.
* TicketPrice (int): giá vé.
* MovieName (string): tên phim của vé.
* sh (Show): suất chiếu của vé.
* hs (HallSeat): ghế đã đặt.
* *Các phương thức của lớp Ticket:*

+ Hàm khởi tạo và hàm hủy.

+ Hàm Set, Get tương ứng từng thuộc tính.

+ Hàm đa năng hóa toán tử xuất (<<).

class Ticket

{

    public:

        int TicketID;

        static int NextID;

        const static int Price = 85000;

    private:

        string MovieName;

        Show \*sh;

        HallSeat hs;

    public:

        Ticket();

        ~Ticket();

        void setHallSeatRow(int&);

        void setHallSeatCol(int&);

        void setShow(Show&);

        void setMovieName(string);

        string getMovieName();

        void setHS(HallSeat&);

        friend ostream& operator<<(ostream&, Ticket&);

};

### Lớp Bill (class Bill.h)

* *Mô tả*: Hóa đơn của khách hàng
* *Các thuộc tính của lớp Bill:*
* Length (int): tổng số vé đã đặt.
* TicketList (Ticket): danh sách vé đã đặt.
* HallName (string): rạp chiếu phim số bao nhiêu.
* Date (Date): ngày chiếu phim.
* *Các phương thức của lớp Bill:*

+ Hàm khởi tạo và hàm hủy.

+ Hàm Set, Get tương ứng từng thuộc tính.

+ Hàm đa năng hóa toán tử xuất (<<).

class Bill

{

    private:

        Date DateCreate;

        int length;

        Ticket \*TicketList;

        string HallName;

    public:

        Bill();

        ~Bill();

        void setDate(Date&);

        void setTicketList(Ticket\*);

        void setLength(int);

        void setHallName(string);

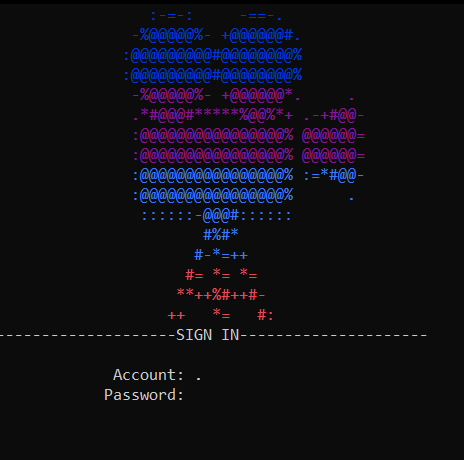
        friend ostream& operator<<(ostream&, Bill&);

};

## Thực hiện chương trình

### Giao diện chính của chương trình

1. *Giao diện đăng nhập của chương trình*



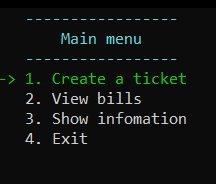
*Hình 10. Giao diện đăng nhập*

### Có thể đăng nhập theo hai tư cách:

* Khách hàng (User)
* Quản lý (Admin)

1. *Với tư cách là khách hàng (User)*

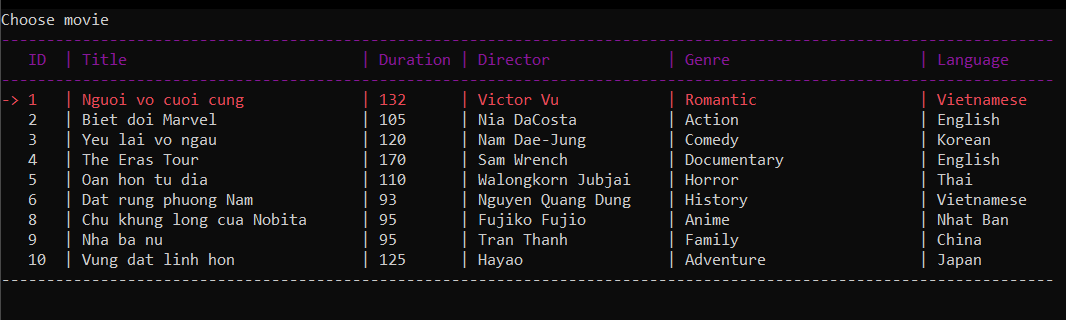
Khách hàng sẽ có 3 chức năng chính: Đặt vé, xem hóa đơn và xem thông tin khách hàng (Dùng phím mũi tên và enter để di chuyển và lựa chọn chức năng).



*Hình 11. Menu của khách hàng*

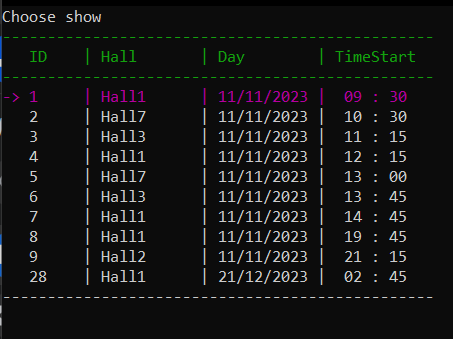
* Chức năng đặt vé xem phim:

Khi chọn chức năng đặt vé xem phim, sẽ liệt kê các phim đang chiếu tại rạp cho khách hàng lựa chọn



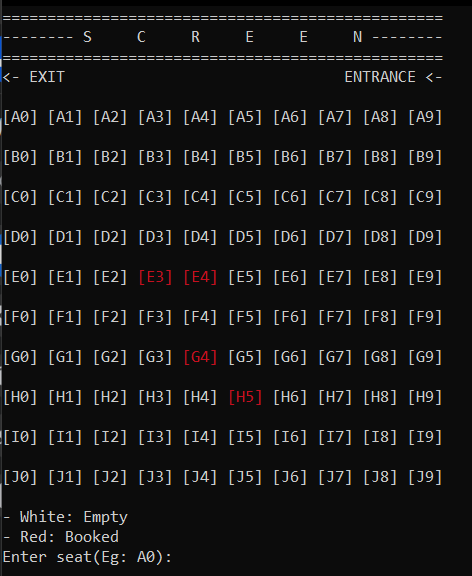
*Hình 12. Danh sách phim đang chiếu tại rạp*

Sau khi đã chọn phim, sẽ liệt kê các suất chiếu hiện có của bộ phim đã chọn để khách hàng lựa chọn khung giờ phù hợp với bản thân



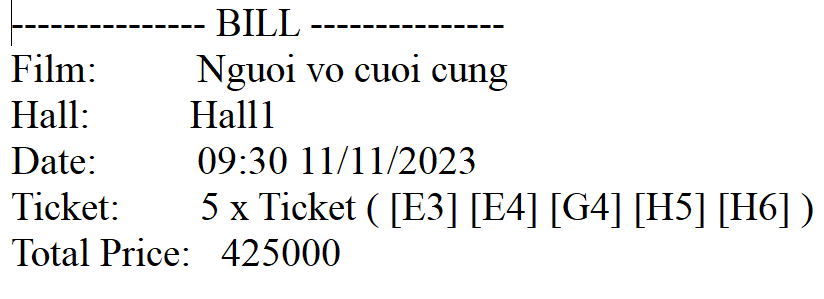
*Hình 13. Suất chiếu của bộ phim*

Khi đã chọn được suất chiếu phù hợp, khách hàng sẽ nhập tổng số ghế muốn mua và chọn ghế.



*Hình 14. Chọn ghế ngồi*

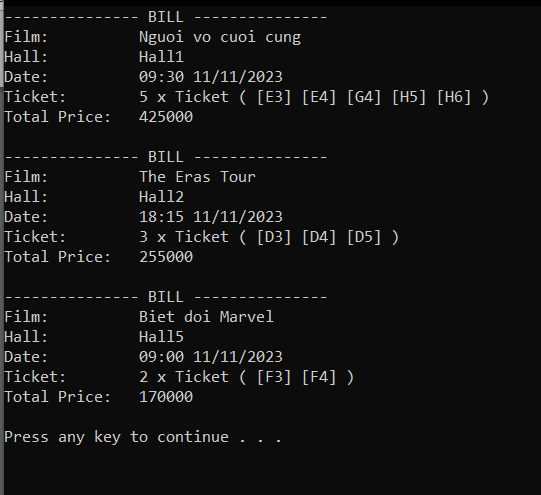
Cuối cùng chương trình sẽ suất hóa đơn cho khách hàng



*Hình 15. Hóa đơn của khách hàng.*

* Chức năng xem lại hóa đơn

Khách hàng sẽ được xem lại tất cả hóa đơn đã đặt phim

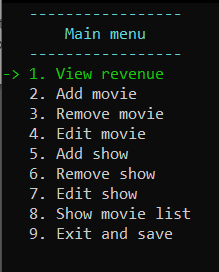


*Hình 16. Lịch sử hóa đơn của khách hàng*

* Chức năng xem thông tin khách hàng như tên, tuổi, giới tính, số điện thoại,...

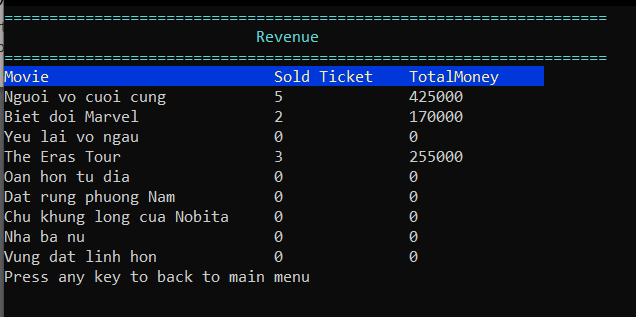
1. *Với tư cách là người quản lý (Admin)*

Quản lý sẽ có 8 chức năng chính, hỗ trợ cho việc đặt vé



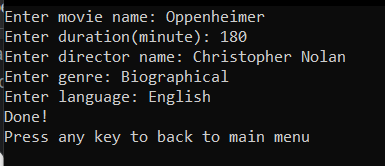
*Hình 17. Menu của quản lý*

* Xem doanh thu của phim



*Hình 18. Doanh thu của rạp*

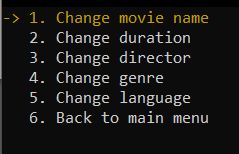
* Thêm một phim mới (nhập từ bàn phím)



*Hình 19. Thêm một phim mới*

* Xóa một bộ phim khỏi danh sách chiếu.
* Thay đổi thông tin của phim

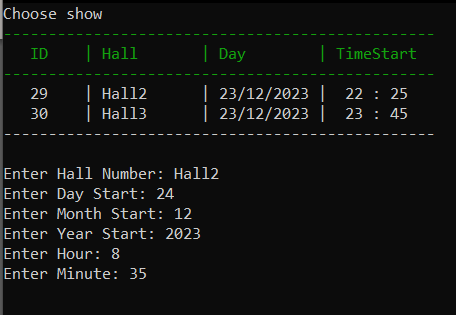
Sau khi chọn phim muốn thay đổi thông tin sẽ có menu chỉnh sửa phim (thông tin mới nhập từ bàn phím)



*Hình 20. Menu chỉnh sửa thông tin phim*

* Thêm suất chiếu cho phim

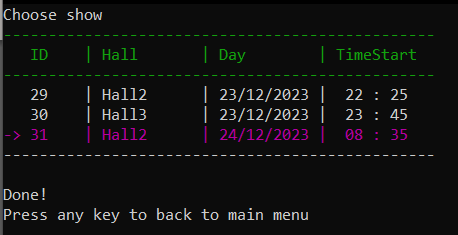
Khi đã chọn phim sẽ thêm suất chiếu cho phim đã chọn nhập từ bàn phím bao gồm rạp, ngày chiếu và khung giờ chiếu phim.



*Hình 21. Thêm một suất chiếu mới*

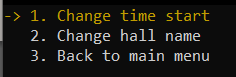
* Xóa một suất chiếu

Xóa một chiếu chiếu của phim đã chọn



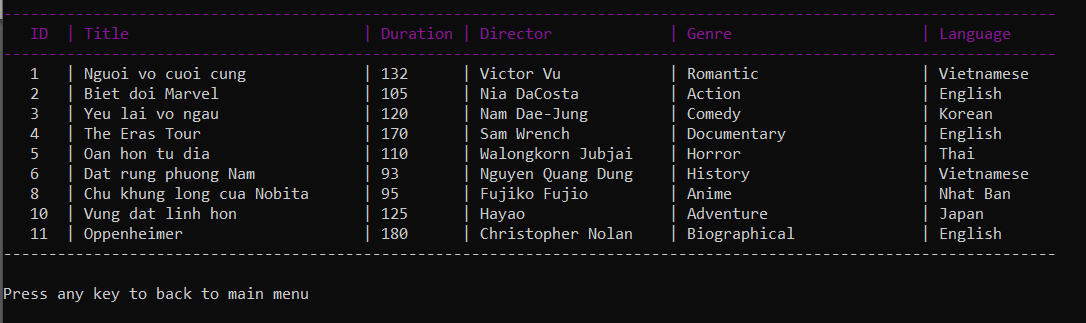
*Hình 22. Xóa một suất chiếu*

* Chỉnh sửa suất chiếu đã chọn của một bộ phim, bao gồm chỉnh sửa các thông tin: Tên rạp, thời gian suất chiếu.



*Hình 23. Menu chỉnh sủa một suất chiếu*

* Xem danh sách phim



*Hình 24. Danh sách phim*

* Exit and save: Sẽ lưu thay đổi toàn bộ những gì quản lý đã thay đổi.

### Nhận xét

Chương trình còn một vài lỗi, khi người dùng sử dụng nhiều sẽ phát sinh nhiều lỗi mà chúng em chưa phát hiện, suy nghĩ đến. Vì là tương tác trên của sổ terminal nên còn nhiều hạn chế và bất tiện cho người sử dụng.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

***1. Kết luận***

Dự án hệ thống đặt vé xem phim và quản lý trên terminal là một dự án trong việc tận dụng công nghệ để cung cấp trải nghiệm đặt vé và quản lý phim thuận tiện và hiệu quả. Bằng cách tích hợp các tính năng như đặt vé, tra cứu phim, và quản lý thông tin, dự án đã mang lại sự thuận tiện cho người dùng và nhân viên quản lý.

Tính năng đặt vé thông qua terminal giúp tiết kiệm thời gian và giảm độ phức tạp của quy trình. Việc triển khai trên terminal giúp đơn giản hóa quá trình sử dụng và tương tác với hệ thống. Nhưng giao diện của chương trình không dễ tiếp cận với người dùng để dễ dàng thao tác mà không gặp khó khăn.

***2. Hướng phát triển***

Đây là một dự án có nhiều tiềm năng và phát triển, phục vụ cho nhu cầu của người sử dụng. Những hướng phát triển này sẽ giúp cải thiện tính năng và sức hút của dự án, tạo ra một hệ thống linh hoạt và thuận tiện cho người dùng.

* Giao diện đồ họa (GUI): Nâng cấp giao diện người dùng bằng cách tích hợp một giao diện đồ họa sẽ tăng trải nghiệm người dùng. Điều này có thể bao gồm hình ảnh, biểu tượng và các yếu tố trực quan khác để làm cho hệ thống trở nên thân thiện hơn và dễ sử dụng hơn.
* Tích hợp thanh toán trực tuyến: Đưa vào tính năng thanh toán trực tuyến giúp người dùng có thể thanh toán vé một cách thuận tiện hơn. Các cổng thanh toán an toàn và phổ biến có thể được tích hợp để cung cấp sự linh hoạt cho người dùng.
* Tích hợp hệ thống đánh giá và phản hồi: Cho phép người dùng đánh giá và viết đánh giá về các bộ phim sẽ tạo ra một cộng đồng trực tuyến. Điều này không chỉ làm phong phú nội dung mà còn giúp người dùng có thêm thông tin khi chọn phim.
* Tối ưu hóa hiệu suất hệ thống: Kiểm tra và tối ưu hóa hiệu suất hệ thống để đảm bảo rằng nó có thể xử lý số lượng lớn người dùng và thông tin về phim mà không gặp vấn đề về hiệu suất.
* Phát triển ứng dụng di động: Tạo ra một ứng dụng di động để mở rộng phạm vi sử dụng đối với người dùng di động.

TÀI LIỆU THAM KHẢO

[Lập trình hướng đối tượng – Wikipedia](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_h%C6%B0%E1%BB%9Bng_%C4%91%E1%BB%91i_t%C6%B0%E1%BB%A3ng)

[C++ MOVIE TICKET BOOKING SYSTEM | C++ Algorithms | cppsecrets.com](https://cppsecrets.com/users/22319897989712197103975756505164103109971051084699111109/C00-MOVIE-TICKET-BOOKING-SYSTEM.php)