

**BỘ GIÁO DỤC VÀ ĐÀO TẠO**  
**TRƯỜNG ĐẠI HỌC NÔNG LÂM TP HCM**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**Môn: Thiết kế hướng đối tượng (Nhóm 10)**

**Đề:** Hệ thống quản lý vé xem phim

**Giảng Viên Hướng Dẫn:**

ThS. Trần Thị Thanh Nga

**Sinh Viên Thực Hiện:**

Nguyễn Khải Nam – 21130449 – Nhóm trưởng

Cao Thành Nam – 21130448

Nguyễn Ngọc Minh Quốc – 21130501

Đặng Thanh Long – 21130432 – *Đã chuyển trường*

Lư Trân Bảo Ngọc – 21130454

TP. HCM, tháng 6 năm 2022

## MỤC LỤC

<b>Lời mở đầu .....</b>	<b>3</b>
<b>Lời cảm ơn .....</b>	<b>4</b>
<b>Nhận xét của giảng viên .....</b>	<b>5</b>
<b>Phần I: Mở đầu .....</b>	<b>6</b>
1.    Lý do chọn đề tài.....	6
2.    Khách thể và đối tượng nghiên cứu.....	6
3.    Giới hạn phạm vi nghiên cứu.....	6
4.    Mục tiêu nghiên cứu .....	6
5.    Nhiệm vụ nghiên cứu .....	7
6.    Phương pháp nghiên cứu.....	7
7.    Ý nghĩa của bài báo cáo .....	7
<b>Phần II: Tóm tắt.....</b>	<b>8</b>
<b>Phần III: Nội dung.....</b>	<b>13</b>
1.    Phần diagram .....	13
2.    Chia chương trình .....	13
3.    Phần đăng nhập.....	14
4.    Phần giao diện chung.....	16
5.    Phần thông tin nhân viên .....	19
6.    Phần bán vé.....	20
7.    Phần thông tin phim .....	22
8.    Phần giờ chiếu phim .....	25

---

9.	Phần lịch sử ghi nhận .....	26
10.	Phần đăng xuất.....	27
11.	Phần menu .....	27
12.	Thanh tìm kiếm.....	28
13.	Hộp thông báo .....	29
14.	Các giao diện khác.....	30
<b>Phần IV: Tổng kết .....</b>		<b>31</b>
<b>Phần V: Bảng phân công.....</b>		<b>33</b>
<b>Phần VI: Tài liệu tham khảo .....</b>		<b>36</b>

## Lời mở đầu

*Ngành công nghệ thông tin là lĩnh vực liên quan đến việc thu nhập, xử lý, lưu trữ và truyền thông bằng cách sử dụng các công nghệ và công cụ máy tính. Nó bao gồm các khía cạnh rộng rãi từ phần cứng máy tính, phần mềm, mạng máy tính, quản lý dữ liệu và ứng dụng vào nhiều lĩnh vực khác nhau.*

*Phân tích và thiết kế hướng đối tượng là một kỹ thuật tiếp cận phổ biến dùng để phân tích, thiết kế một ứng dụng, hệ thống. Nó dựa trên bộ các nguyên tắc chung, đó là một tập các hướng dẫn để giúp chúng ta tránh khỏi một thiết kế xấu. Việc ứng dụng các nguyên tắc này là rất quan trọng trong các đồ án dù lớn hay nhỏ, nó giúp cho chúng ta tiết kiệm thời gian cho sau này khi mở rộng, nâng cấp, ...*

*Để có thể ứng dụng tốt các mẫu được học chúng em – nhóm 10 đã thống nhất với nhau thực hiện đồ án “**Hệ thống quản lý vé xem phim**”. Đồ án được viết bằng chương trình java swing với mục tiêu là giúp rạp chiếu phim có được một ứng dụng giúp giảm tải được công việc của những người bán vé và người quản lý, tiết kiệm thời gian trong việc đặt vé xem phim, xem các thông tin về phim, ... Trong báo cáo đồ án **word** này chúng em xin trình bày nhiều về phân thiết kế và ít đề cập đến phần code.*

## **Lời cảm ơn**

*Chúng em xin trân thành cảm ơn cô Trần Thị Thanh Nga đã tận tình giảng dạy, chỉ bảo, trang bị cho chúng em những kiến thức cần thiết để thực hiện đề tài. Trong quá trình học tập và thực hiện đề tài chúng em đã học hỏi được rất nhiều kiến thức bổ ích và kinh nghiệm quý báu làm nền tảng cho quá trình học tập, làm việc và nghiên cứu sau này. Chúng em đã nỗ lực và quyết tâm cao để thực hiện đề tài này, song cũng không tránh khỏi những thiếu sót, mong cô sẽ tận tình chỉ bảo thêm cho chúng em.*

## Nhận xét của giảng viên

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Ngày tháng năm 2023

(Ký tên)

## Phần I: Mở đầu

### 1. Lý do chọn đề tài

*Hệ thống quản lý vé xem phim yêu cầu cần tổ chức một cách hợp lý để chương trình có thể mở rộng và với mục tiêu ứng dụng các kiến thức đã học một cách tốt nhất, chúng em đã thống nhất lựa chọn đề tài này.*

### 2. Khách thể và đối tượng nghiên cứu

*Khách thể nghiên cứu: Hoạt động đặt vé, xem lịch sử ghi nhận, xem thông tin về phim, quản lý nhân viên, ...*

*Đối tượng nghiên cứu:*

- + *Thiết kế giao diện trên java swing*
- + *Áp dụng các mẫu được học*
- + *Vẽ diagram*
- + *Thiết kế giao diện trên figma*
- + *Các chức năng của hệ thống: chọn vé, chọn ghế, xem thông tin phim, ...*

### 3. Giới hạn phạm vi nghiên cứu

*Về phần giao diện: Chúng em chỉ nghiên cứu trên giao diện java swing.*

*Về phần chức năng: Chúng em làm về các chức năng của phía nhân viên.*

*Về phần kiến thức: Chúng em đã cố gắng áp dụng các mẫu phổ biến và được giới thiệu trong quá trình học tập song cũng trong tránh khỏi các sai lầm có thể mắc trong quá trình làm.*

### 4. Mục tiêu nghiên cứu

*Các mục tiêu trong quá trình nghiên cứu đồ án:*

- + *Giao diện hệ thống phía nhân viên và người quản lý*

- + *Áp dụng các mẫu một cách hợp lý*
- + *Chương trình đáp ứng các chức năng cơ bản: đặt vé, xem thông tin phim,...*
- + *Diagram trực quan*
- + *Đáp ứng mở nếu cần mở rộng*

## **5. Nhiệm vụ nghiên cứu**

*Mỗi người trong nhóm đều được phân công một công việc và được kiểm soát bởi nhóm trưởng. Mỗi cá nhân có trách nhiệm hoàn thành và tự tìm hiểu các vấn đề liên quan để hoàn thành chương trình một cách tốt nhất.*

## **6. Phương pháp nghiên cứu**

*Quá trình làm việc:*

- + *Cùng nhau thống nhất và bàn về các công việc cơ bản*
- + *Vẽ ra giao diện cơ bản bằng figma*
- + *Vẽ diagram cơ bản*
- + *Tiến hành chuyển đổi giao diện từ figma sang java swing*
- + *Chia phần giao diện về chức năng của nhân viên cho mỗi cá nhân thực hiện*
- + *Kiểm tra điều chỉnh về phần code*
- + *Hoàn chỉnh diagram*
- + *Kiểm tra và chỉnh sửa thêm về hệ thống*
- + *Viết báo cáo*

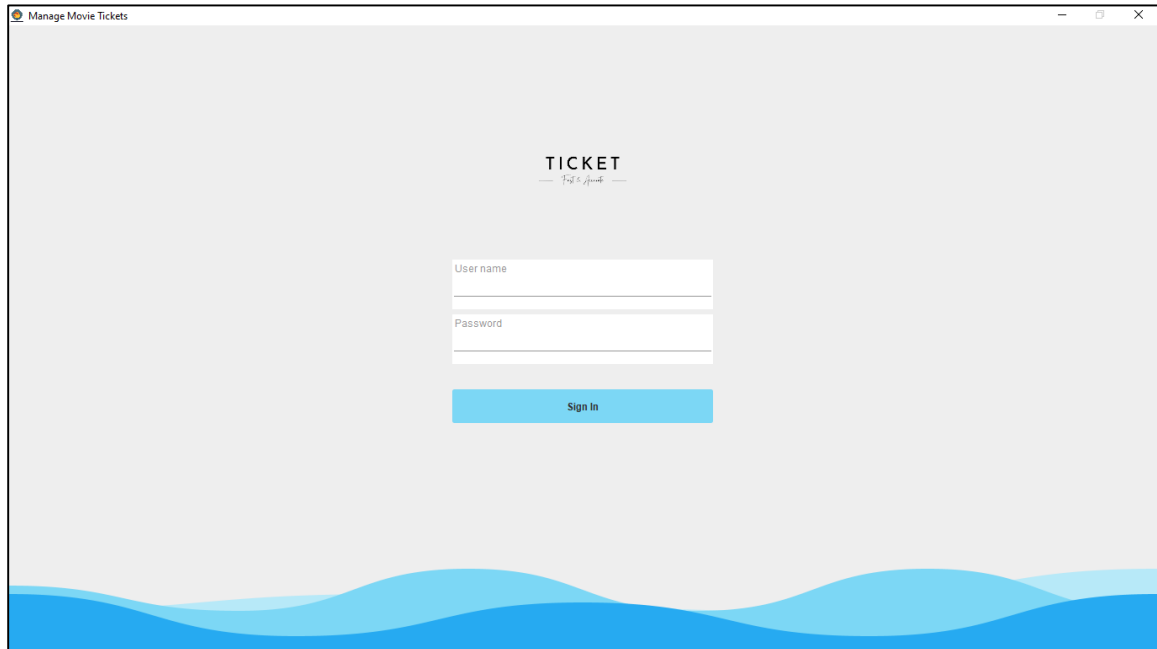
## **7. Ý nghĩa của bài báo cáo**

*Bài báo cáo cung cấp một nền tảng để tìm hiểu về các công nghệ, phương pháp và quy trình cần thiết để xây dựng một hệ thống hiệu quả và linh hoạt. Ngoài ra thực hiện đồ án giúp nâng cao kỹ năng làm việc nhóm, kỹ năng tự nghiên cứu.*



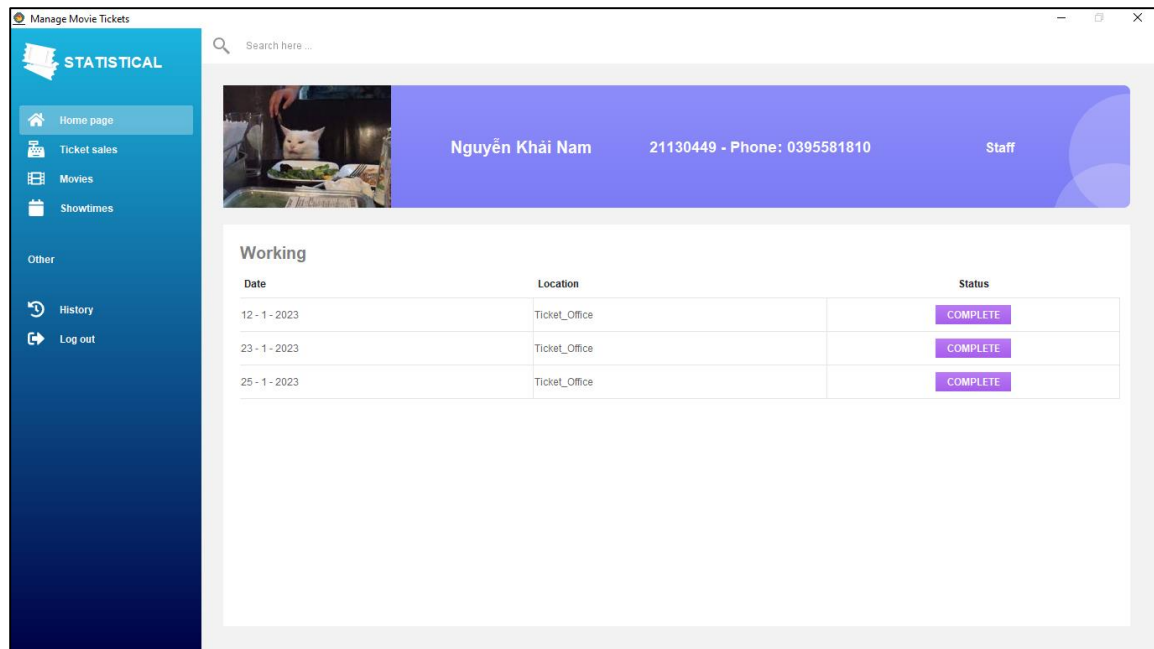
## Phần II: Tóm tắt

Đầu tiên khi khởi động hệ thống người dùng sẽ được đưa vào giao diện đăng nhập. Trong hệ thống sẽ tồn tại hai loại tài khoản: **admin** và **staff**. Do đó tùy nhu cầu mà người dùng đăng nhập tài khoản thích hợp.



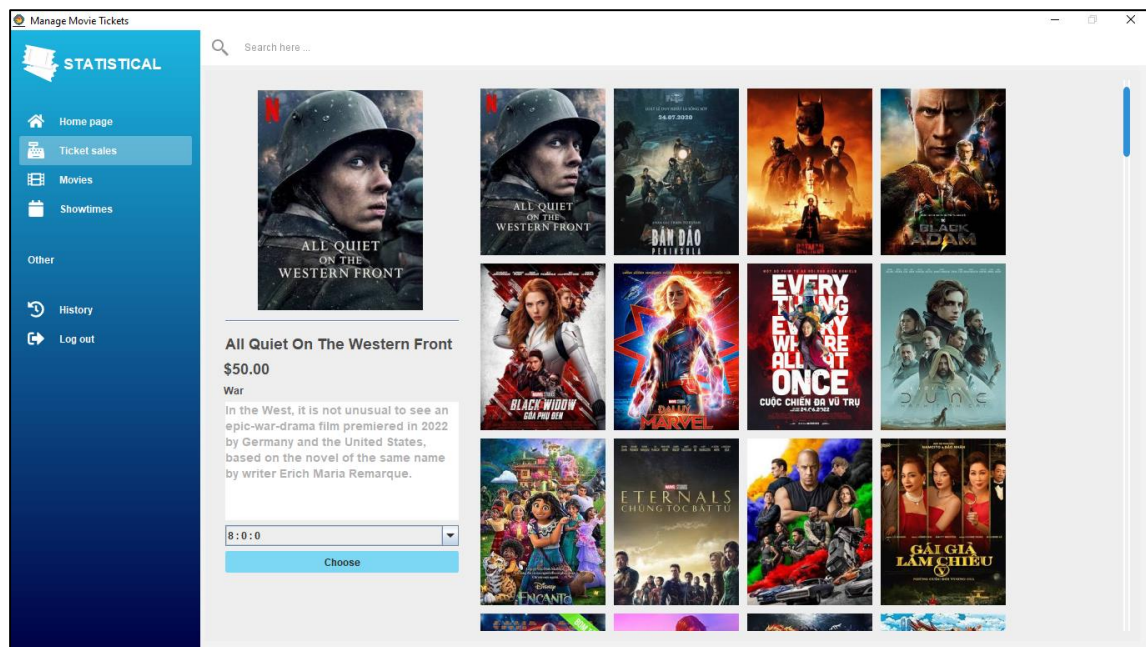
Giao diện bắt đầu

Vì chương trình chỉ hoàn tất mọi giao diện về phần **staff** nên chúng em sẽ chỉ xoay quanh phần **staff**. Sau khi đăng nhập vào bằng tài khoản được cấp cho **staff** người dùng sẽ nhìn thấy các thông tin của mình để đảm bảo đúng tài khoản. Ngoài ra thông tin ban đầu còn cho nhân viên biết mình đã được đánh dấu đã đi làm vào thời gian nào và trạng thái hoàn thành của công việc.



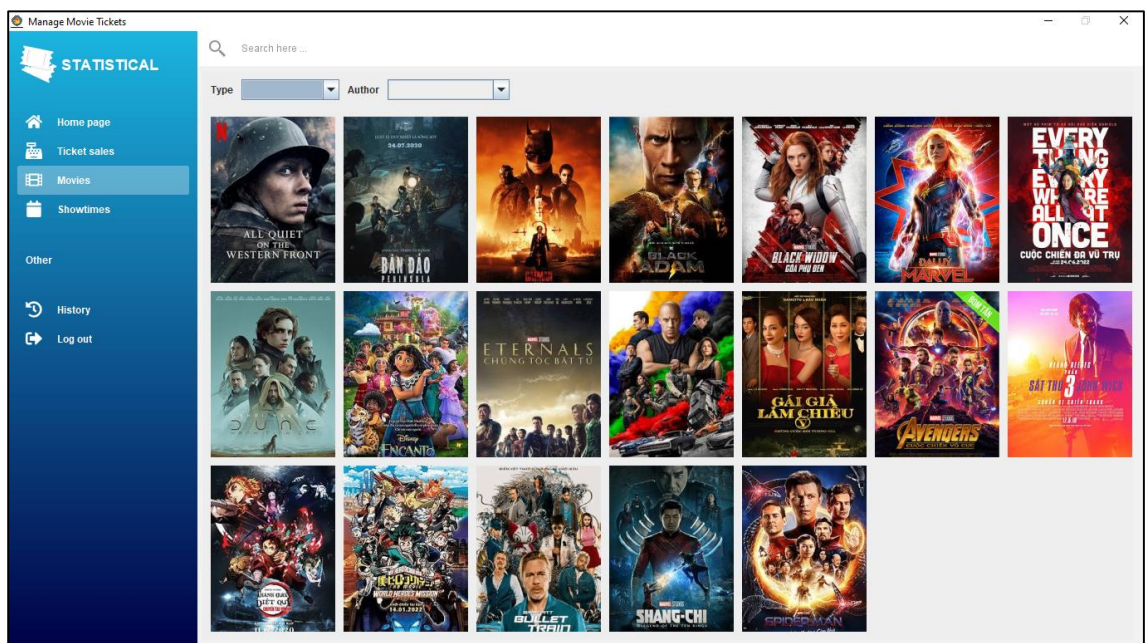
Giao diện nhân viên

*Khi có khách đến và đặt vé thì nhân viên có hai cách là tìm kiếm trực tiếp tên phim hoặc lựa chọn vào phần “**Ticket sales**” trên phần menu để truy cập vào việc đặt vé xem phim. Việc chọn phim sẽ yêu cầu 2 thông tin đó là phim chọn và giờ chiếu, sau khi lựa chọn xong thì người dùng sẽ được đưa tới phần lựa chọn ghế và cuối cùng là phần hóa đơn.*



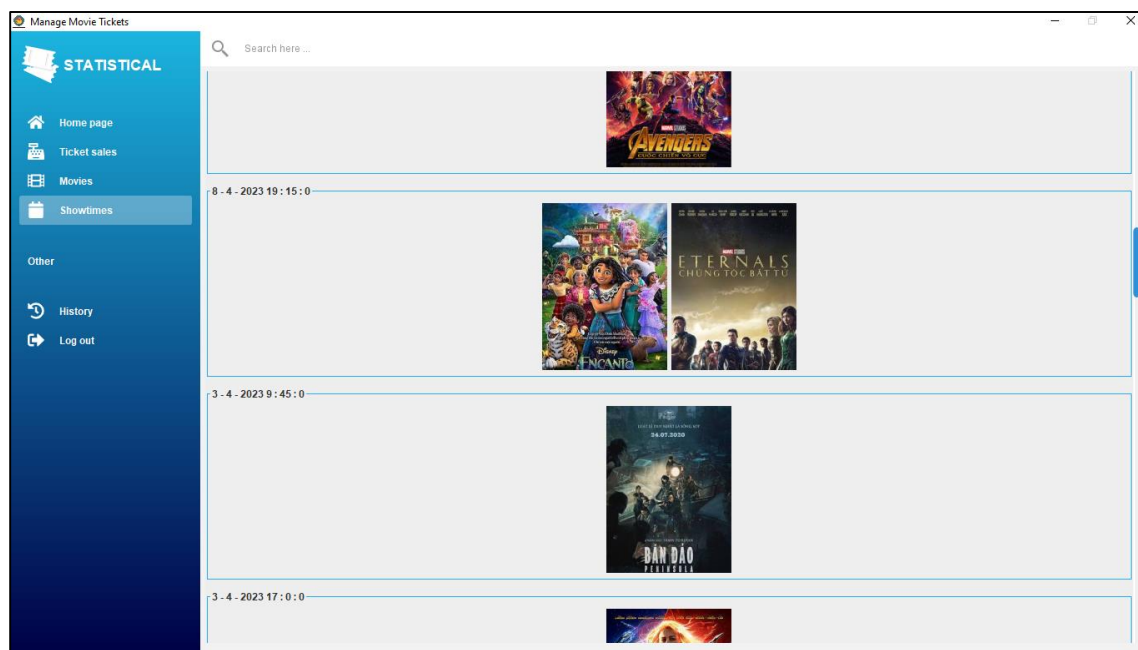
Giao diện lựa chọn phim

Khi cần thêm các thông tin về bộ phim nào đó thì nhân viên tiến hành lựa chọn phần “**Movies**” trên menu. Ngoài ra có thể lọc để tìm nhanh.



Giao diện thông tin phim

Tiếp đó nếu cần thông tin giờ chiếu thì nhân viên có thể chọn **“Showtimes”**. Các bộ phim trùng tên sẽ được phân biệt bằng thời gian.



Giao diện giờ chiếu

**“History”** ghi nhận lại các vé đã đặt trong quá trình làm việc của nhân viên này. Các tên phim trùng được phân biệt bằng giờ lựa chọn.

Date	Movie	Price
2023-06-11 11:11:18	All Quiet On The Western Front	150.0
2023-06-11 11:11:24	All Quiet On The Western Front	150.0
2023-06-11 11:11:31	The Batman (2022)	150.0
2023-06-11 11:11:37	Everything Everywhere All at Once	100.0

Lịch sử đặt vé

*“Log out” đưa người dùng về phần giao diện đăng nhập.*

## Phần III: Nội dung

### 1. Phần diagram

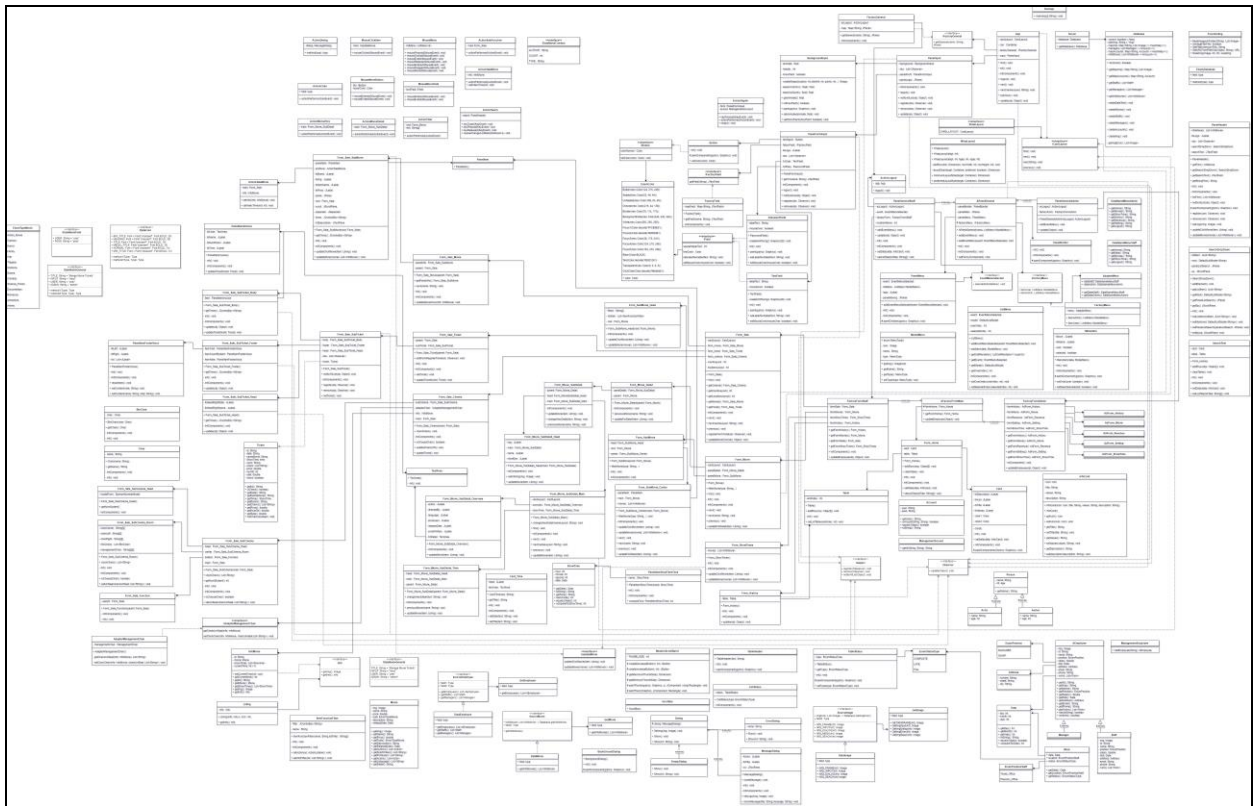
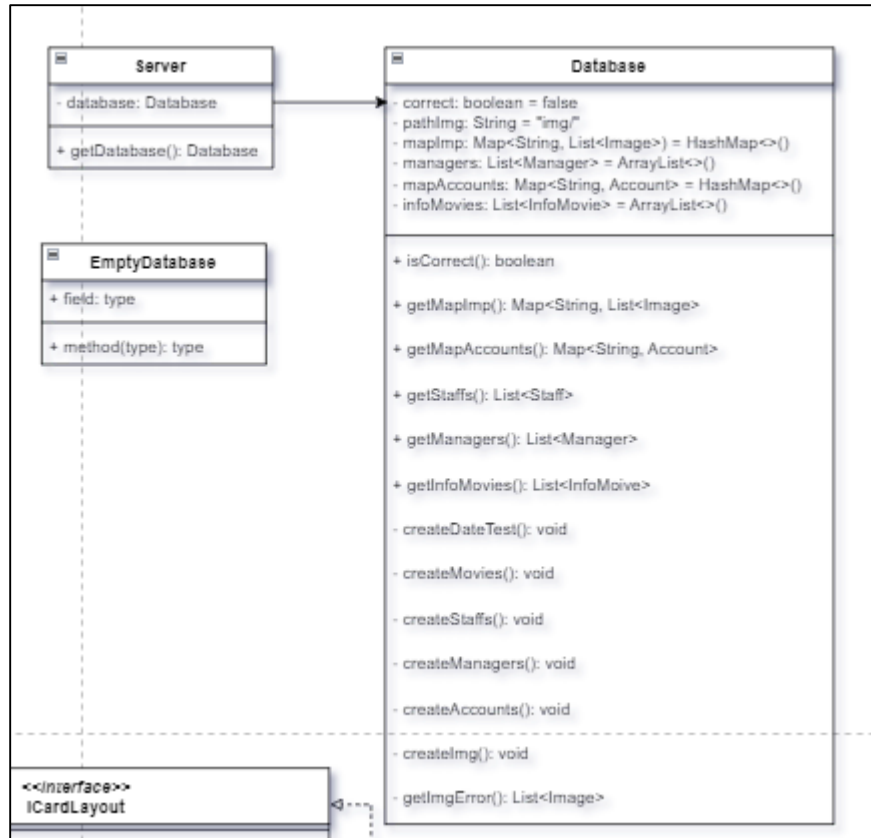


Diagram toàn chương trình

Diagram trên là diagram của toàn bộ chương trình “**hệ thống quản lý vé xem phim**” và hệ thống của chúng em xây dựng trên mô hình **MVC**.

### 2. Chia chương trình

Trong hệ thống quản lý này chúng em chia chương trình thành 2 thành phần cơ bản mà chúng em tham khảo được từ các hệ thống khác đó là phần **giao diện** và **database**, tuy nhiên ở phần **database** vì chúng em chưa đủ kiến thức để xử lý và tương tác với các cơ sở dữ liệu thực tế nên chúng em giả định một class sẽ đứng ra làm vai trò **database**. Trong thực tế **database** sẽ được tương tác thông qua một phần nào đó trong chương trình và ở đây chúng em sử dụng một lớp có tên là **server**.



Giả lập tương tác database

Để giống với thực tế về việc database có thể không có dữ liệu thì chúng em tạo ra class có tên là **EmptyDatabase**.

Về phần giao diện chương trình chính sẽ có tên là **App**, các hoạt động từ input tới chương trình chính bên trong phải thông qua chương trình chính **App** này.

### 3. Phần đăng nhập

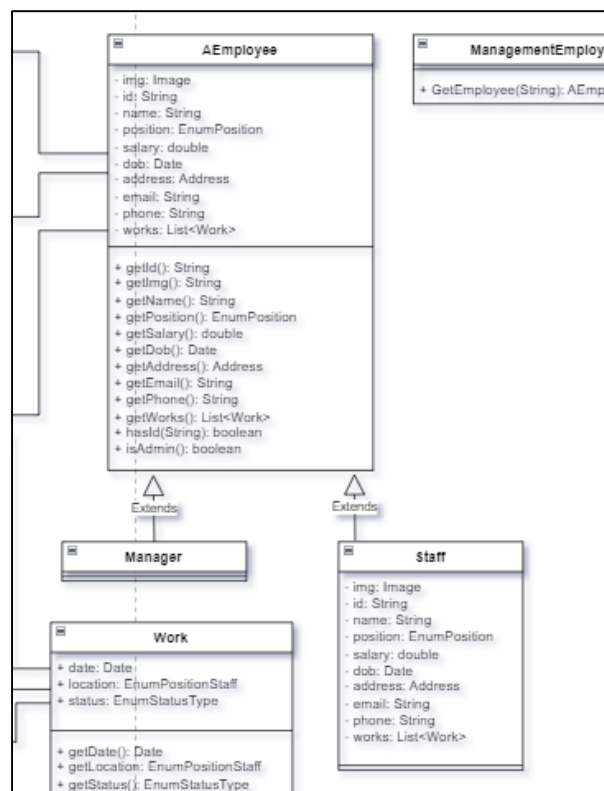
Mỗi nhân viên sẽ được ứng với một tài khoản khác nhau và các tài khoản này được phân biệt bằng mã của nhân viên đó. Do đó khi người dùng đăng nhập vào hệ thống sẽ kiểm tra các trường thông tin và nếu như thỏa mãn thì input sẽ gửi thông tin nhân viên đó tới cho App.



```
// Lấy ra nhân viên theo id được truyền  
public static AEmployee getEmployee(String id) {  
    // Thực hiện tìm nhân viên  
    for (AEmployee aEmployee : employees)  
        if (aEmployee.hasId(id))  
            return aEmployee;  
  
    return null;  
}
```

Trả về tài khoản nhân viên tương ứng

Trong đó **AEmployee** định dạng cho cả nhân viên là **admin** và **staff**. Và việc trả về **AEmployee** là đảm bảo quy tắc về tính trừu tượng trong thiết kế hệ thống, nó giúp ta không phải mất nhiều thời gian kiểm tra xem tài khoản đó thuộc về loại nào mà chính class nào cần thông tin sẽ tự kiểm tra điều đó.



Áp dụng tính đa hình

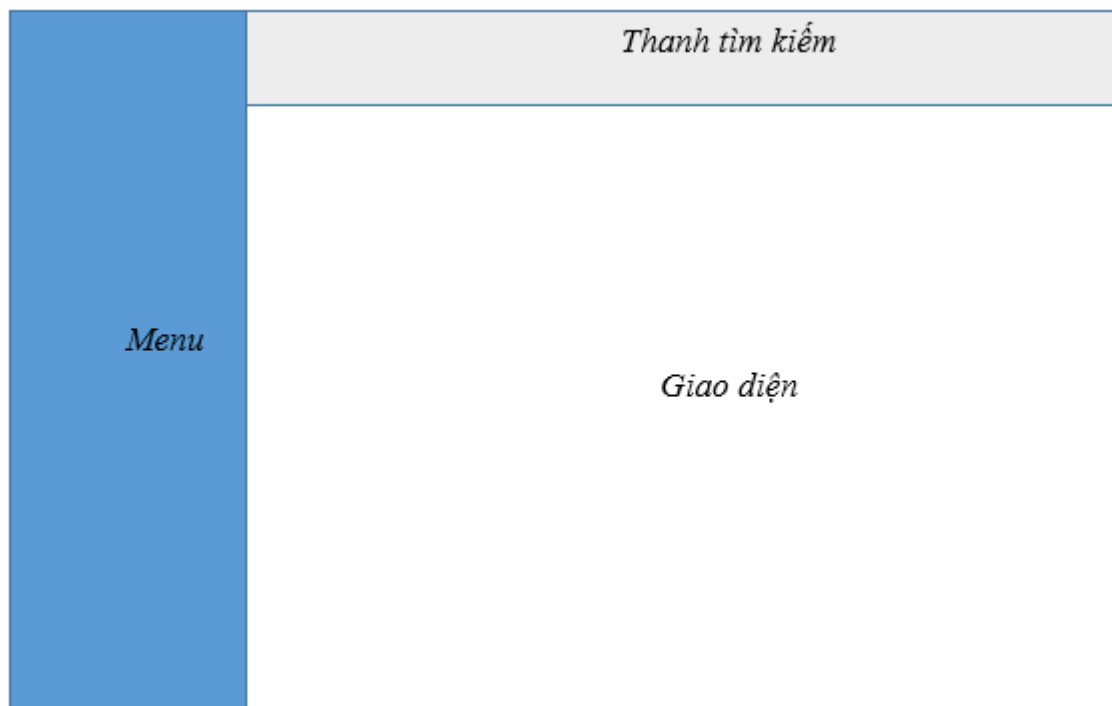


Và để thực hiện được việc nhận thông tin đăng nhập từ input thì ta sẽ sử dụng mẫu **Observer**, trong đó **PanelInput** là **Subject** và **App** sẽ là **Observer**.

Ngoài ra phần đăng nhập còn một chức năng là khởi động lại phông đăng nhập khi người dùng đăng xuất. Cũng giống như việc đăng nhập để nhận được thông tin đăng xuất thì **PanelInput** lúc này sẽ là **Observer** nên khi đó **PanelInput** sẽ nhận được thông tin đăng xuất từ phía chương trình chính.

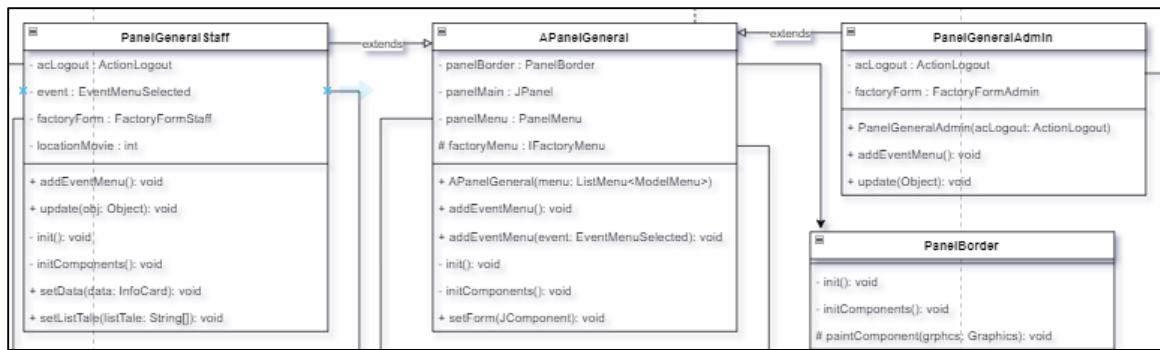
#### 4. Phần giao diện chung

Để đồng nhất giao diện ở cả hai trạng thái chương trình **admin** và **staff** thì cả hai giao diện này đều sử dụng chung một bố cục chương trình



Đồng nhất giao diện

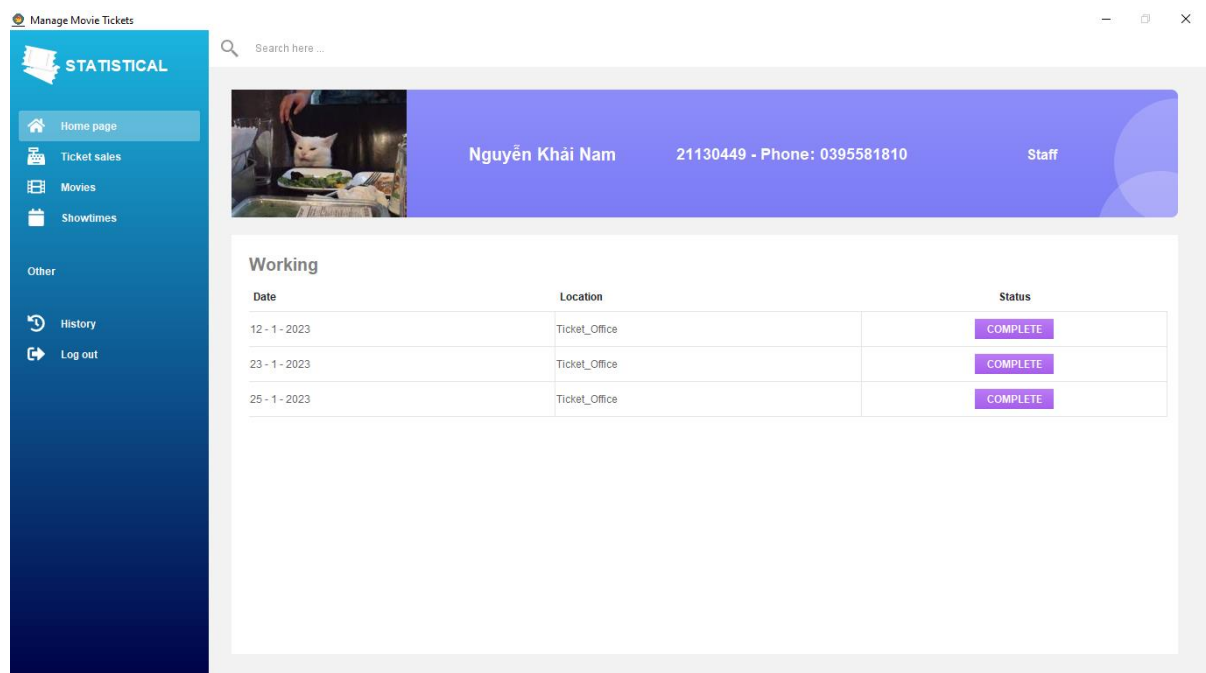
Và đây là thiết kế ở phía diagram cũng như cách tổ chức trong chương trình:



Cách tổ chức cho 2 loại tài khoản

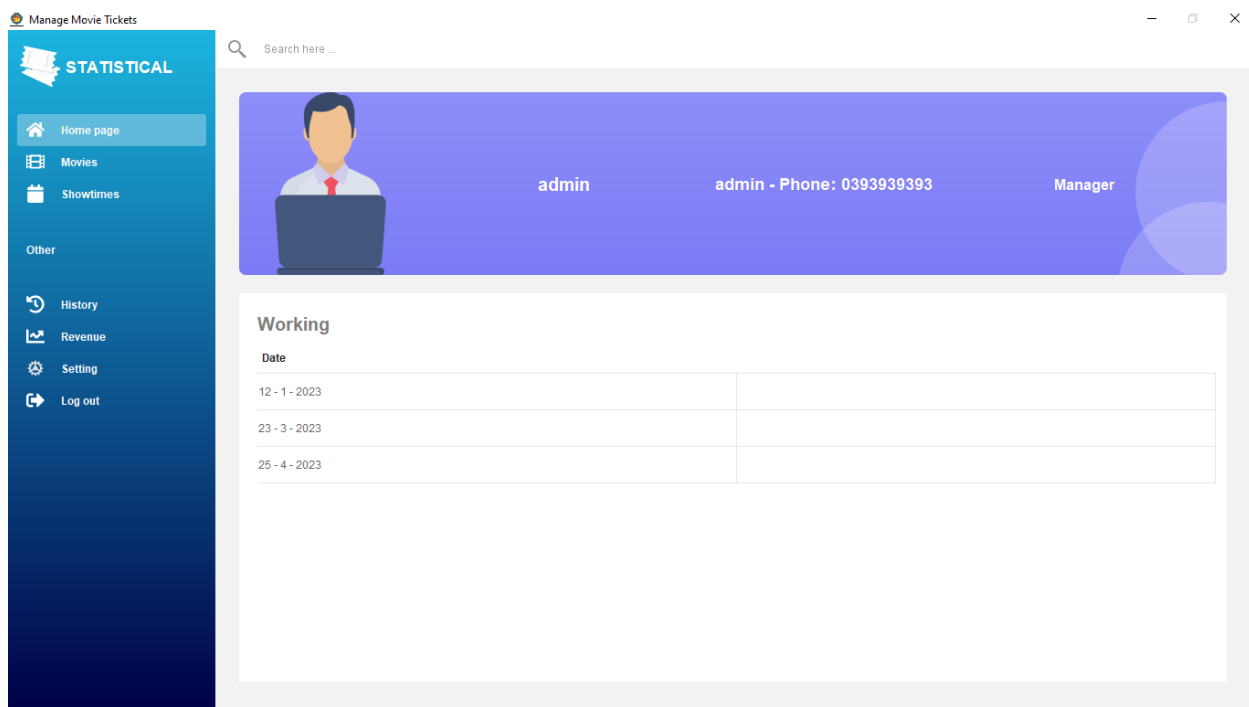
Các giao diện sẽ tùy theo nhu cầu cần thiết từ phía rạp chiếu phim mà sẽ được tùy chỉnh thêm bớt các phần tử để phù hợp. Ở đây chúng em đã tổng hợp các tính năng phổ biến nhất ở trong một hệ thống quản lý vé xem phim.

Đây là giao diện ở phía nhân viên, nó sẽ bao gồm các chức năng có thể có: Phần thông tin nhân viên; Phần bán vé; Thông tin phim; Thông tin giờ chiếu; Lịch sử ghi nhận vé đã in; Đăng xuất



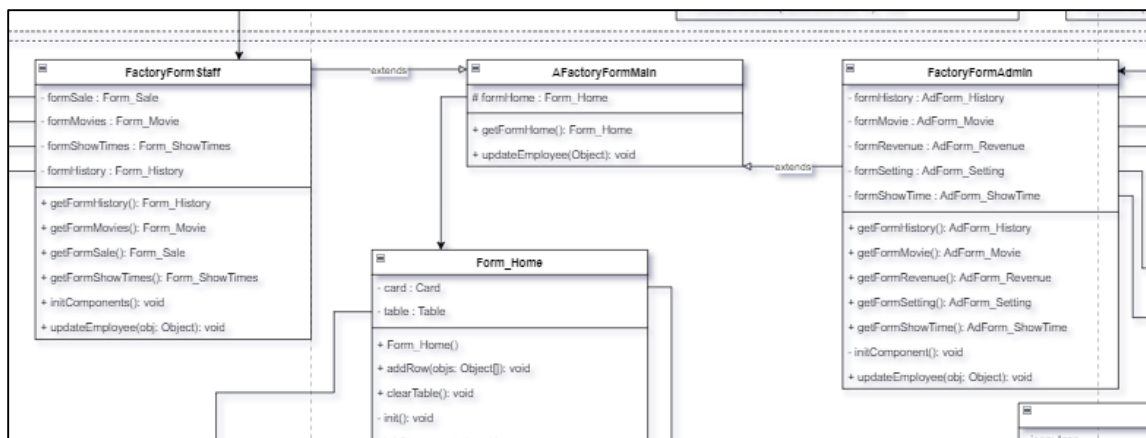
Tổng quan về chức năng của nhân viên

Ở phía admin cũng có một số chỗ giống như: Phần thông tin, do đó chúng em đã sử dụng chung phần này cho cả hai phía. Một số chức năng sẽ có ở phần admin (mở rộng): Phần thông tin nhân viên; Phần thông tin phim – khác với nhân viên admin có quyền chỉnh sửa thông tin; Phần thời gian chiếu – chỉ admin mới có quyền chỉnh thời gian chiếu; Lịch sử ghi nhận vé đã in - admin có thể thấy mọi lịch sử bán; Phần thống kê; Cài đặt; Đăng xuất.



Tổng quan về phía admin

Dựa vào việc có sự khởi tạo giống nhau chúng em đã áp dụng mẫu **Factory Abstract** và mẫu **Template** vào chương trình. Điều này giúp cho những chương trình tiết kiệm được một phần bộ nhớ khởi tạo và nó cũng minh bạch hơn trong việc khởi tạo các giao diện.



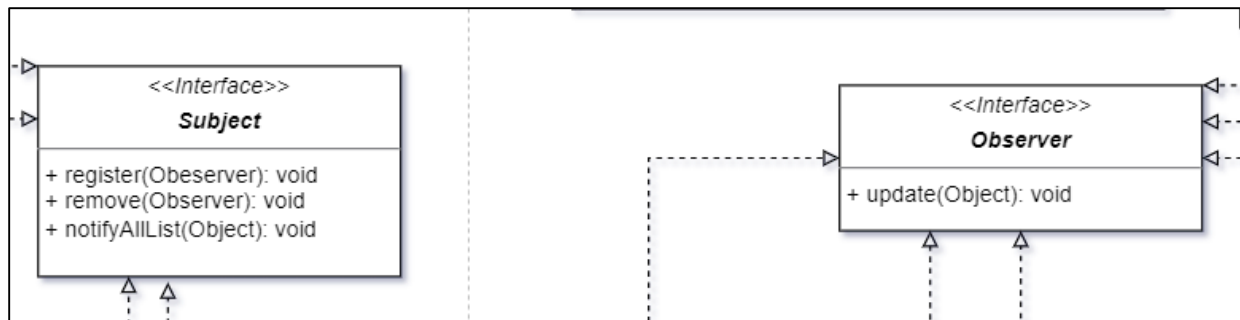
Ứng dụng Factory

Ngoài ra thanh tìm kiếm và nơi chứa giao diện khi lựa chọn trên menu cũng được sử dụng chung. Và kể từ phần này trở đi chúng em sẽ chỉ nói về giao diện ở phía **staff**.

## 5. Phần thông tin nhân viên

Dựa vào mẫu **Observer** ở phần input, **Form\_Home** – giao diện thông tin nhân viên, sẽ nhận được thông tin rằng nhân viên nào đã đăng nhập vào hệ thống từ đó hiển thị các thông tin cơ bản như tên, mã, số điện thoại, chức vụ. Ngoài ra mỗi nhân viên khi đi làm còn được ghi nhận lại ngày đã đi làm, làm tại vị trí nào và trạng thái hoàn thành công việc ra sao.

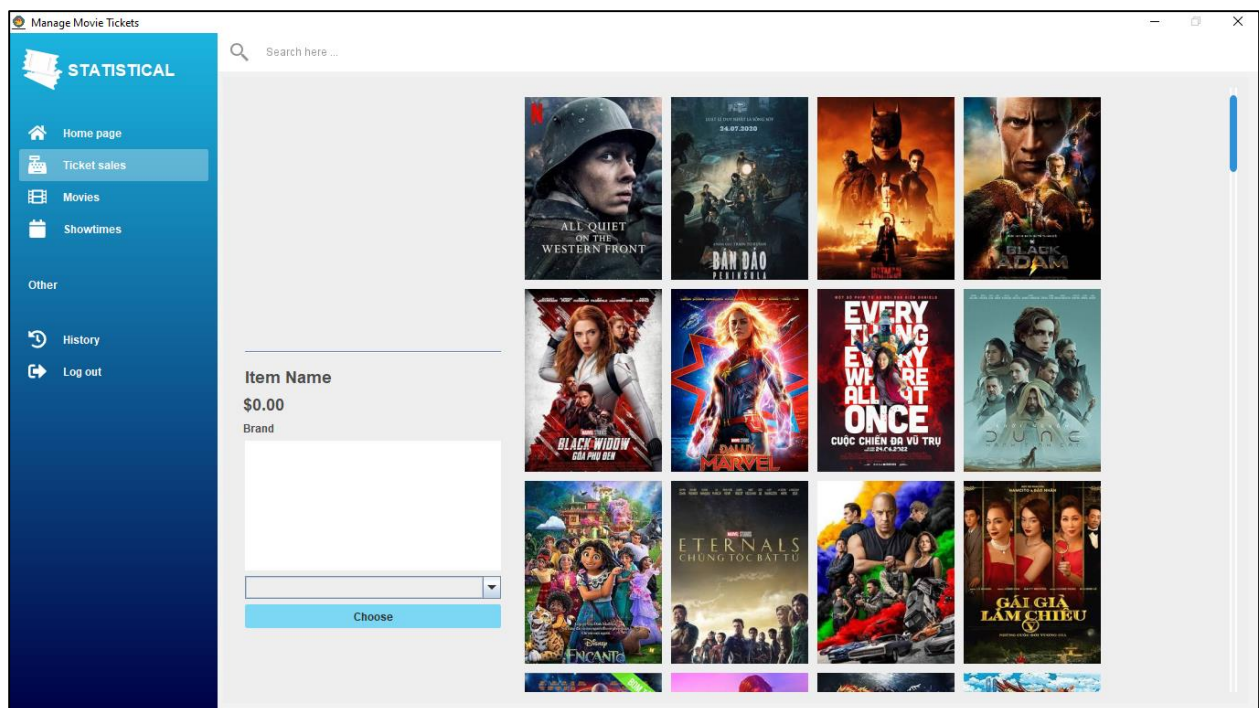
Để biết được thông tin ngày làm việc input làm việc với lớp quản lý thời gian làm việc của mỗi cá nhân và khi thông báo tới các phần tử đăng ký nhận thông tin từ nó (cụ thể ở đây là **Form\_Home**) nó sẽ truyền một “khối” bao gồm thông tin cơ bản và thông tin làm việc của nhân viên. Lý do làm được như vậy là trong mô hình **Observer** chúng em sử dụng là kiểu **Object**, điều này giúp cho việc mở rộng và tái sử dụng dễ dàng hơn.



Mẫu Observer

## 6. Phần bán vé

*Đây là giao diện ban đầu của phần bán vé*



Giao diện lựa chọn phim

*Trước tiên người dùng phải lựa chọn bộ phim nếu không hệ thống sẽ báo lỗi và yêu cầu lựa chọn bộ phim mới có thể di chuyển tiếp, bằng cách áp dụng mô hình **Observer** và mô hình **Adapter** giao diện được chuyển đổi về dạng thông tin và được kiểm tra xem bộ phim nào đã được lựa chọn (các bộ phim có mã id riêng).*

Khi lựa chọn và chuyển bộ phim và thời gian chiếu ta sẽ được tiếp đến giao diện chọn ghế.

Student:

SCREEN

A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14
B1	B2	B3	B4	B5	B6	B7	B8	B9	B10	B11	B12	B13	B14
C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	C11	C12	C13	C14
D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14
E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14
F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14
G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14
H1	H2	H3	H4	H5	H6	H7	H8	H9	H10	H11	H12	H13	H14
I1	I2	I3	I4	I5	I6	I7	I8	I9	I10	I11	I12	I13	I14
J1	J2	J3	J4	J5	J6	J7	J8	J9	J10	J11	J12	J13	J14

Seated

Un Seated

Choose


Previous

Next

Lựa chọn ghế ngồi cho bộ phim

Gần giống với ý tưởng về tài khoản của mỗi nhân viên, mỗi bộ phim và từng khung giờ của nó có một bảng ghế khác nhau và để đọc được số ghế không thể ngồi được của bộ phim vào khung giờ đó thì chúng em sử dụng mẫu **Observer** kết hợp với việc sử dụng mẫu **Adapter** chuyển đổi dữ liệu từ giao diện về phần tử và chuyển nó tới cho phần chọn ghế. Và cũng giống như bộ phim nếu chúng ta không lựa chọn bất kỳ ghế nào mà nhấn **next** thì hệ thống sẽ báo lỗi.

Đi tiếp sẽ là phần giao diện vé, đây là phần giao diện nhân viên sẽ thấy trước khi xuất vé in cho khách hàng. Và như ở trên chúng em vẫn sử dụng **Observer** và **Adapter** để cung cấp thông tin.

 <b>Object-oriented design Group 10</b>	
Sologan is slogan	
#1-273719426	
<b>Invoice</b>	
Date: 2023-06-12 17:28:41	
Peninsula	Room
8 : 30 : 0	A
Chair: G7, G8	
Student:	0
Total:	100.0
*Note: NO GIVE BACK!!!	

Previous

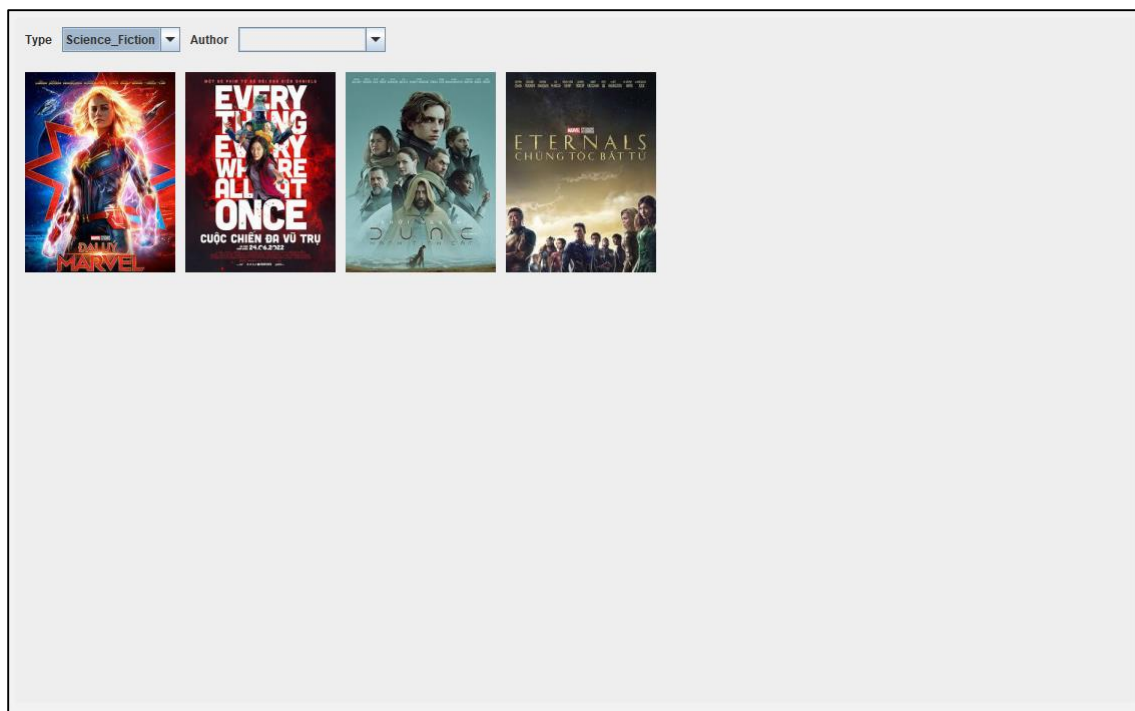
Next

Hóa đơn

Ngoài ra trong phần giao diện chúng em cũng áp dụng thêm mẫu **Factory** và mẫu **Singleton** cho các giao diện.

## 7. Phần thông tin phim

Đây là nơi tham khảo giúp cho nhân viên nếu cần thêm thông tin gì về bộ phim để tư vấn cho khách hàng. Ngoài ra chúng em cũng cung cấp cơ chế lọc để tìm nhanh bộ phim ở phần này.

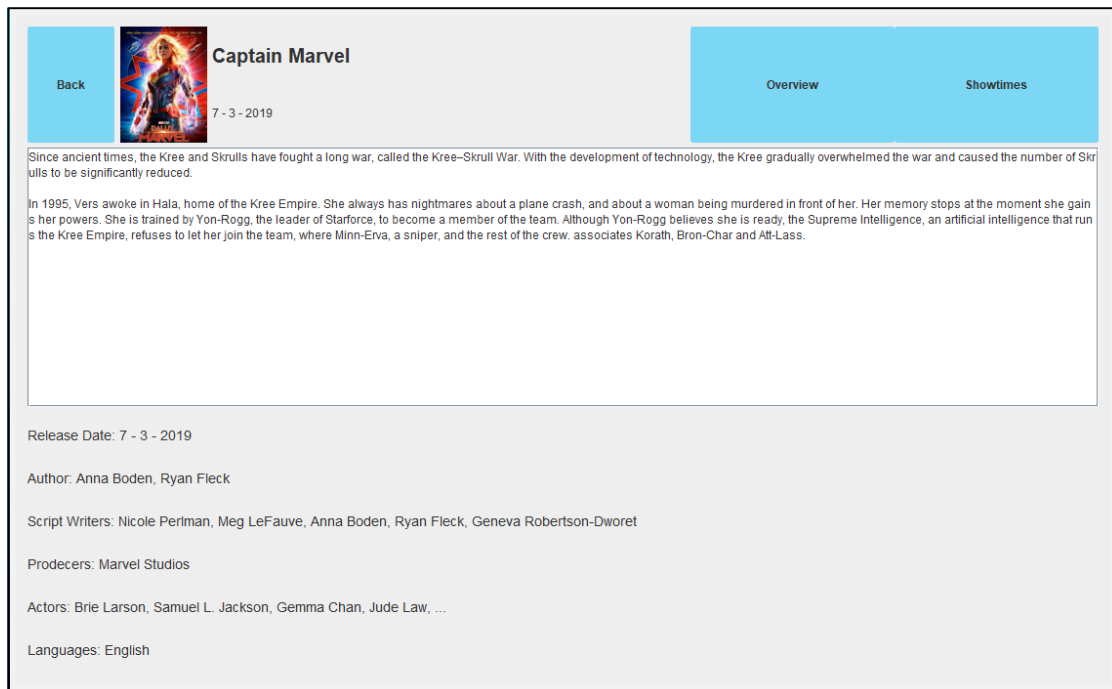


Lọc phim theo thông tin

Trong phần giao diện này chúng em phân tích sẽ có hai khu vực là phần **head** – nơi chứa bộ lọc; phần **center** – nơi chứa bộ phim khi đã lọc. Hai phần này có mối quan hệ với nhau thông qua mô hình **Observer** bằng cách **center** sẽ nhận thông tin mà người dùng đã lựa chọn lọc từ phần **head**.

Khi lựa chọn vào một bộ phim nhân viên sẽ thấy các thông tin cần thiết.

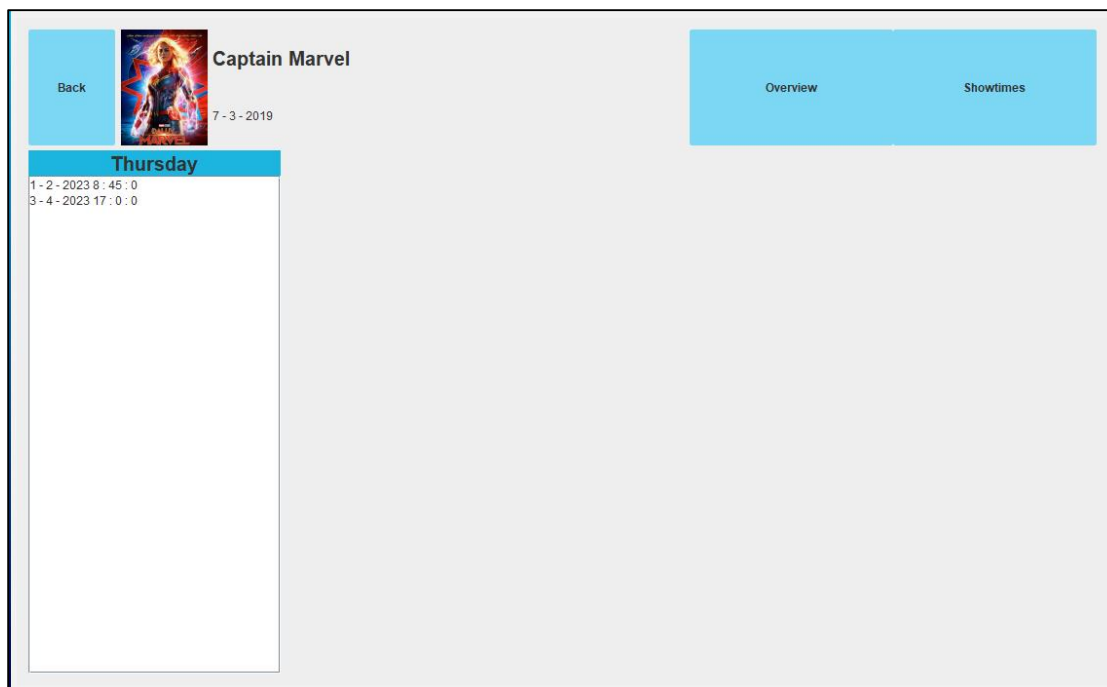




#### Thông tin về bộ phim

Và thông tin này bao gồm 2 phần cơ bản: thông tin chi tiết, thông tin giờ chiếu. Và cũng gần giống với phân loại phim chúng em phân tích giao diện ra làm hai phần **head** và **center** sau đó sử dụng mẫu **Observer** để thông báo từ head tới center rằng giao diện nào cần được hiển thị.

Giao diện khi lựa chọn **Showtimes**

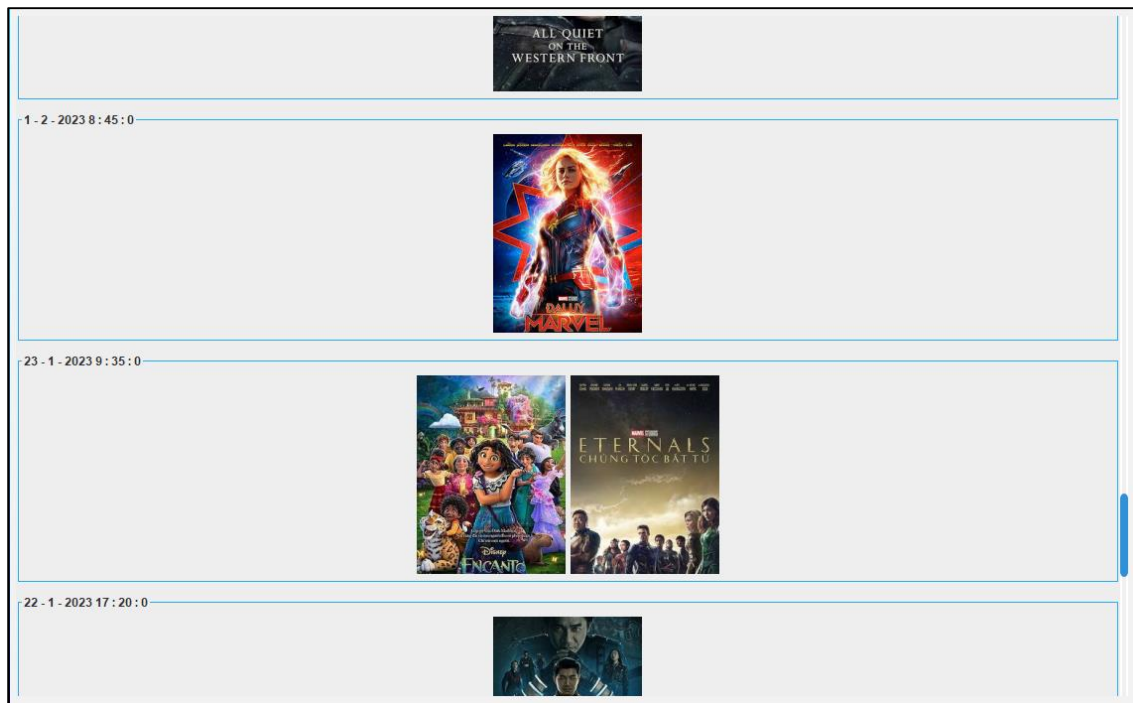


Thông tin giờ chiếu

Ngoài ra chúng em cũng áp dụng mẫu **Factory Simple** trong việc khởi tạo các phân tử trên giao diện và mẫu **Adapter** trong việc chuyển đổi từ giao diện về thông tin.

## 8. Phần giờ chiếu phim

Phần này thường phục vụ cho việc nắm bắt thông tin về các bộ phim sẽ sắp chiếu. Các bộ phim này có thể trùng giờ trùng ngày chiếu tuy nhiên các rạp chiếu sẽ khác nhau. Ngoài ra phần này cũng có thể xem trong phần giao diện chi tiết bộ phim.



Khung giờ chiếu của hệ thống

## 9. Phần lịch sử ghi nhận

Việc hiển thị các thông tin vé chỉ được diễn ra khi nhân viên nhấn in vé ở lúc cuối trong phần bán vé. Và cũng như hầu hết việc nhận thông tin từ giao diện khác trong chương trình chúng em đã sử dụng mẫu **Observer** kết hợp với mẫu **Adapter**. Trong thông tin chỉ hiển thị cơ bản tuy nhiên trong quá trình nhận dữ liệu các thông tin vẫn đầy đủ vì **Observer** được sử dụng tham số truyền vào **Object**, do nhu cầu hiển thị ít nên chúng em chỉ sử dụng vài thông tin cơ bản.

History		
Date	Movie	Price
2023-06-12 17:28:41	Peninsula	100.0

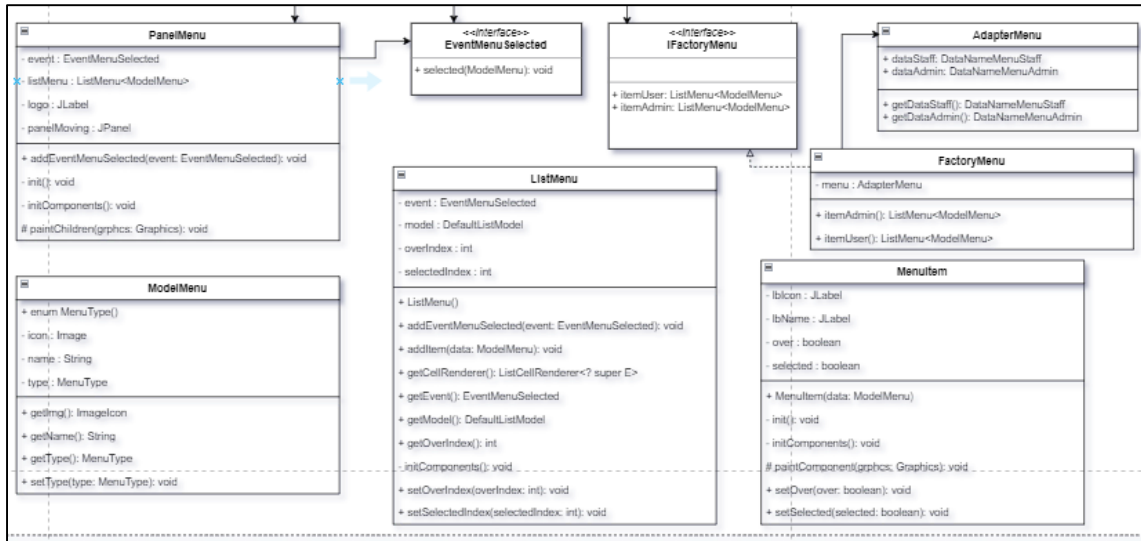
Lịch sử vé đã xuất

## 10. Phần đăng xuất

Khi nhấn vào “**Log out**” dựa vào mẫu **Observer** chương trình chính sẽ thông báo cho **App** và từ đó thông báo tới **input**. App sẽ chuyển giao diện về phần đăng nhập, và input sẽ làm sạch dữ liệu cho phòng đăng nhập.

## 11. Phần menu

Dựa vào mẫu **Composite** chúng em đã thiết kế menu có thể mở rộng với các bậc. Menu sẽ được chia làm hai dạng, phần tử cuối cùng và phần tử có thể mở rộng, trong đó phần tử cuối cùng sẽ chỉ gồm tên, còn phần tử có thể mở rộng sẽ có khả năng chứa phần tử cuối cùng hoặc phần tử có thể mở rộng khác.

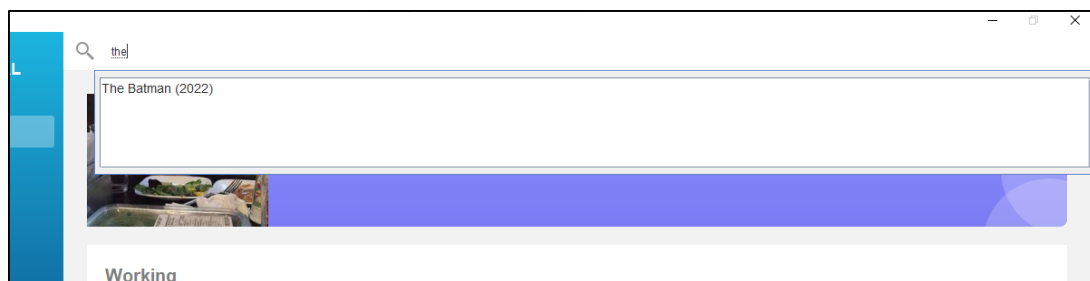


Thiết kế của menu

Ngoài ra chúng em cũng kết hợp trong đó mẫu **Adapter** – tương tác lấy thông tin từ **model** đưa cho **view** trong mô hình **MVC** và sử dụng **Factory** – trong việc khởi tạo.

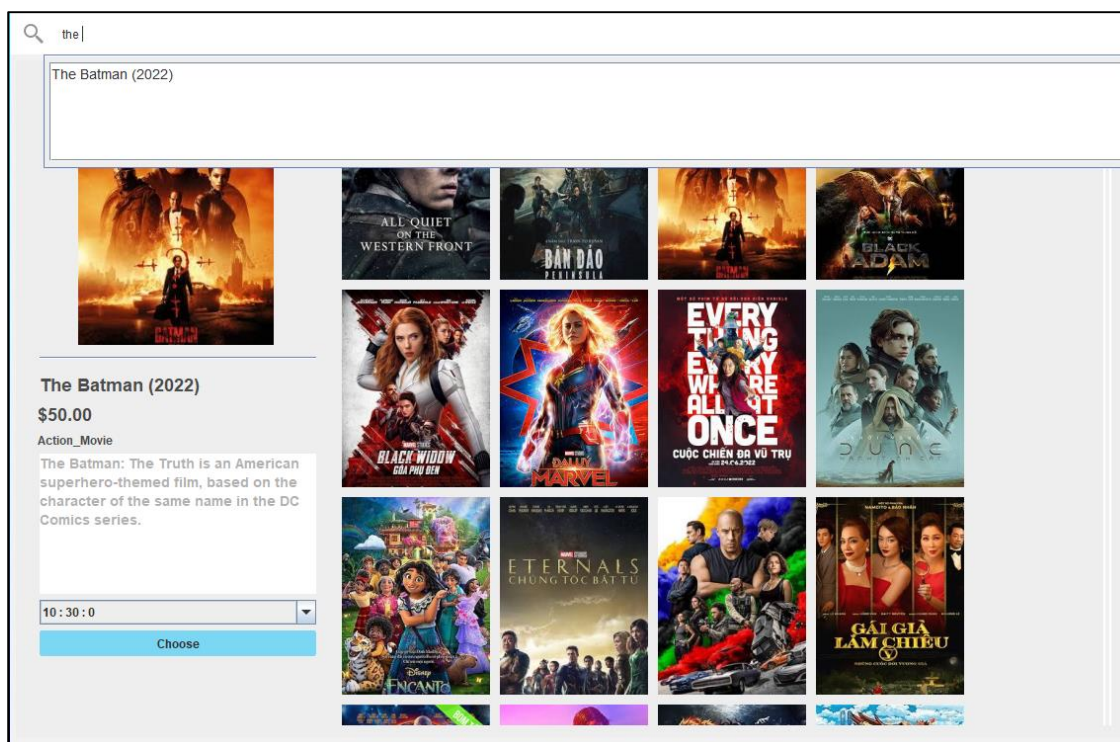
## 12. Thanh tìm kiếm

Người dùng có thể tìm kiếm trực tiếp bộ phim cần đặt vé trên thanh tìm kiếm.



Thanh tìm kiếm

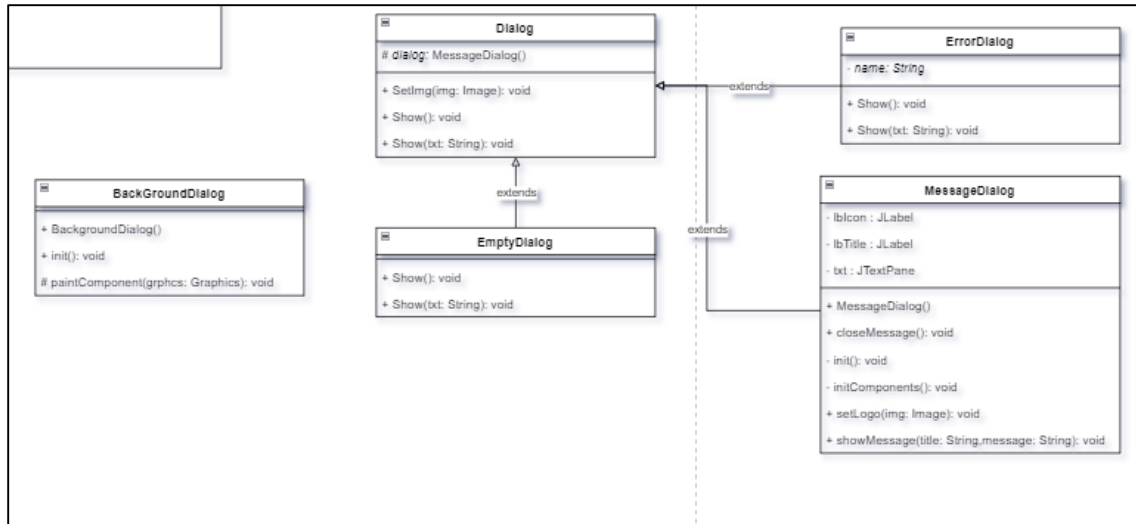
Nếu bộ phim tìm kiếm có trong hệ thống thì hệ thống sẽ đưa ra lựa chọn và khi chọn vào thì hệ thống sẽ đưa ta đến với nơi bán vé, từ đó giúp tiết kiệm được rất nhiều thời gian trong việc chọn lựa.



Tìm kiếm giúp chuyển đến phần bán vé

### 13.Hộp thông báo

Bất kỳ tác vụ nào không hợp lệ hay có điều gì đó xảy ra thì thông báo sẽ được tung ra để nhân viên kiểm tra. Vì có một số việc thực hiện giống nhau trên các hộp thông báo khác nhau nên chúng em đã sử dụng mẫu **Template** trong trường hợp này.



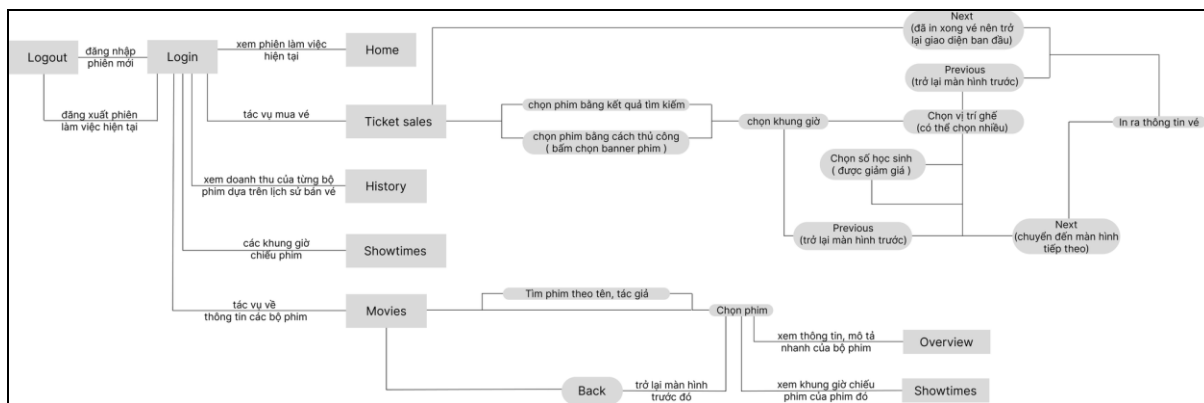
Thông báo trong chương trình

## 14. Các giao diện khác

Ngoài ra chúng em cũng đã tùy chỉnh lại giao diện của một số thành phần trên hệ thống **java swing** và kết hợp các mẫu để phù hợp với chương trình như: thanh trượt, thanh nhập dữ liệu, thông báo, ....

## Phần IV: Tổng kết

Ở phần này chúng em sẽ tổng kết lại các hoạt động của chương trình.



Hoạt động của chương trình

Đầu tiên khi nhân viên truy cập vào hệ thống họ sẽ được yêu cầu phải đăng nhập vào tài khoản của mình. Sau đó họ sẽ được đưa tới giao diện đầu tiên là giao diện thông tin, ở giao diện này người dùng tiến hành kiểm tra thông tin xem đã chính xác chưa. Sau đó tùy vào công việc thì người dùng tiến hành lựa chọn trên thanh menu tương ứng.

Khi chọn vào phần đặt vé xem phim người dùng sẽ phải lựa chọn bộ phim và khung giờ chiếu, ngoài ra người dùng có thể truy cập vào việc đặt vé ở bất kỳ đâu bằng cách tìm tên phim bằng thanh công cụ tìm kiếm. Sau khi đã lựa chọn bộ phim thì người dùng tiếp tục lựa chọn chỗ ngồi, mỗi màu sẽ qui định cho loại ghế khác nhau, nếu nhấn quay lại chương trình sẽ hủy đi việc chọn ghế vừa rồi và đưa người dùng trở lại giao diện chọn phim, nếu nhấn đi tiếp hệ thống sẽ đưa người dùng tới giao diện vé, ở đây người dùng tiến hành kiểm tra các thông tin trên vé và xuất vé cho khách hàng. Tuy nhiên nếu ở giao diện chọn vé người dùng nhấn lui về thì hệ thống sẽ đưa về giao diện chọn ghế và hủy đi thông tin vừa nhận từ lúc trước.



Lựa chọn “**Movies**” trên menu sẽ cho phép người dùng xem các thông tin chi tiết về một bộ phim, ở giao diện này người dùng còn có thể xem những khung giờ sẽ chiếu của phim đó. Ngoài ra cũng có phần lọc để tìm nhanh các bộ phim.

Phần “**Showtimes**” sẽ cho xem các khung giờ chiếu phim của rạp.

“**History**” ghi lại lịch sử vé của người dùng đã xuất.

Và cuối cùng “**Log out**” sẽ đưa người dùng trở về giao diện đăng nhập ban đầu.

### Phần V: Bảng phân công

	Nguyễn Khải Nam (21130449)	Cao Thành Nam (21130448)	Nguyễn Ngọc Minh Quốc (21130501)	Lư Trần Bảo Ngọc (21130454)	Đặng Thanh Long (21130432) – Đã chuyển trường
Tổng quan chương trình			✓	✓	
Tìm kiếm thông tin tham khảo cho chương trình	✓				
Thiết kế giao diện bằng Figma			✓	✓	✓
Giao diện đăng nhập và chức năng	✓	✓			
Giao diện thông tin nhân viên	✓				
Giao diện đặt vé và chức năng	✓	✓			

<i>Giao diện chọn ghế và chức năng</i>	✓	✓			
<i>Giao diện hóa đơn</i>	✓	✓			
<i>Giao diện chi tiết phim và chức năng</i>	✓		✓		
<i>Giao diện giờ chiếu và chức năng</i>	✓				
<i>Giao diện lịch sử</i>	✓			✓	
<i>Thay đổi giao diện thành phần (thanh trượt, bảng,...)</i>	✓				
<i>Màu sắc</i>	✓				
<i>Ảnh phim</i>				✓	
<i>Nội dung phim</i>	✓				
<i>Dữ liệu để kiểm tra</i>	✓				

Thanh tìm kiếm	✓	✓			
Các chức năng khác (đăng nhập, chuyển giao điện, ...)	✓				
Vẽ diagram	✓		✓	✓	
Viết word	✓				
Áp dụng mẫu	✓				
Kiểm tra và chỉnh sửa	✓				

## Phần VI: Tài liệu tham khảo

Giao diện Figma: <https://www.figma.com/file/p0uOVFTWaaKe0Zgbx2QSkb/TKHDT?node-id=0%3A1&t=zi5Xl9ewfpkvnKCy-0>

Đồ án tham khảo: <https://khotrithucso.com/doc/p/xay-dung-chuong-trinh-quan-ly-ban-ve-xem-phim-253839>

Hình ảnh, icon: <https://www.iconarchive.com/tag/java-swing>

Youtuber về java swing: <https://youtube.com/@LaingRaven>