**BÁO CÁO PROJECT**

# Hệ Thống Quản lí Nước

|  |  |
| --- | --- |
| **Học kỳ:** | **Học Kỳ 1** |
| **Lớp:** | **PF02** |
| **Nhóm:** | **Group 3/hệ thống quản lí nước** |
| **Giảng viên hướng dẫn:** | **Đào Văn Đức** |
| **Các thành viên:** | **NDE18014 – Đỗ Văn Hoàng**  **NDE18019 – Vũ Công Tuân** |
|  |  |

# Mục Lục

Hệ thống quản lí nước 1

Mục lục 2

I. Giới thiệu dự án 3

II. Phân tích yêu cầu hệ thống 3

III. Thiết kế chi tiết 9

IV. Phân công công việc cho từng thành viên trong nhóm 14

V. Hướng dẫn cài đặt 16

Phụ lục 17

Định dạng tài liệu 18

# Giới thiệu dự án

1. Hệ thống dự định sẽ làm,

Quản lí thu tiền nước trực tuyến.

1. Mục đích của dự án

Hệ thống quản lí thu tiền nước trực tuyến cơ bản để thực hiện cung cấp cho người quản lí có thể quản lí mọi lúc mọi nơi để thu thập thông tin về khách hàng đang sử dụng dịch vụ cung cấp nước ở bất kì đâu bất kì lúc nào.

1. Phạm vi dự án được ứng dụng

Sử dụng cho quản lí thu tiền nước.

1. Tên hệ thống

Hệ thống quản lí nước.

1. Môi trường triển khai (phần cứng, phần mềm)

Phần cứng: laptop và pc.

Phần mền:chạy trên nền tảng dotnet.core

6. Công cụ sử dụng để phát triển

Ngôn ngữ lập trình : C#

Hệ quản trị cơ sở dữ liệu : MySQL version 8.0.15

Công cụ tạo báo cáo :Microsoft Word

Công cụ sử dụng để phát triển: Visual Studio Code, Violet UML,Mysql Workbench.

7. Xác định yêu cầu khách hàng

Khách hàng muốn tạo hóa đơn

Khách hàng muốn đăng kí hợp đồng dịch vụ cung cấp nước.

Khách hàng muốn xem thông tin cá nhân.

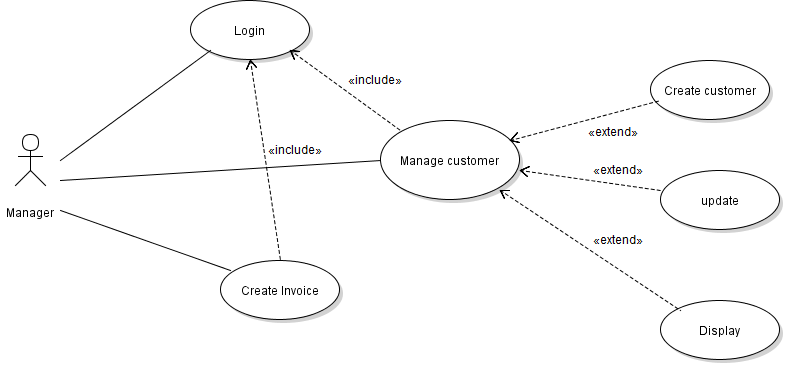
# Phân tích yêu cầu hệ thống

Hệ thống quản lí thu tiền nước được xây dựng dựa trên những nhu cầu thực tế của khách hàng và quản lí nhằm giải quyết những khó khăn gặp phải, giảm thiểu rủi do trong quá trình quản lí thu tiền nước. Hệ thống hướng đến đối tượng là người quản lí. Hệ thống có các chức năng đăng nhập,quản lí khách hàng,tạo hóa đơn. Các chức năng giúp người quản lí dễ dàng xem thông tin khách hàng ,tạo hóa đơn hoặc thay đổi thông tin cá nhân của khách hàng một cách dễ dàng.

Các thực thể chính bao gồm: Thực thể khách hàng, thực thể người quản lí,thực thể hóa đơn.

### Use Case

#### Bản vẽ Use Case (Use Case Diagram)



#### *Mô tả Use Case*

##### Mô tả chức năng đăng nhập

|  |  |
| --- | --- |
| **Tiêu đề** | **Nội dung** |
| **Tên chức năng** | **Đăng nhập** |
| **ID chức năng** | **UC01** |
| **Mô tả** | Chức năng này diễn tả một người dùng đăng nhập vào hệ thống. |
| **Actor** | Quản lí |
| **Sự kiện kích hoạt** | Quản lí muốn đăng nhập vào hệ thống. |
| **Organizational**  **Benefits** | Người quản lí đăng nhập vào hệ thống |
| **Triggers** | Người quản lí:chọn đăng nhập   1. Nhập email và password 2. Chọn đăng nhập vào hệ thống |
| **Tiền điều kiện** | Quản lí phải có tài khoản để đăng nhập để vào hệ thống. |
| **Hậu điều kiện** | Thành công : Người quản lí đăng nhập được vào hệ thống.  Lỗi : Người quản lí không đăng nhập được vào hệ thống. |
| **Luồng sự kiện chính.** | Chức năng này bắt đầu khi người quản lí muốn đăng nhập vào hệ thống.   1. Hệ thống yêu cầu người quản lí nhập email và mật khẩu của mình. 2. Người quản lí nhập email và mật khẩu. 3. Hệ thống xác thực email và mật khẩu và cho phép người quản lí vào hệ thống. 4. Kết thúc chức năng. |
| **Luồng sự kiện phụ.** | Người quản lí nhập Email hoặc mật khẩu không hợp lệ.   1. Hệ thống hiển thị màn hình đăng nhập kèm theo thông báo đăng nhập sai. 2. Người quản lí có thể quay lại bước 2 ở luồng sự kiện chính hoặc chọn huỷ đăng nhập. |
|  |  |

|  |  |
| --- | --- |
| **Ngoại lệ** | Ngoại lệ 1: kết nối lỗi với CSDL   1. Hiển thị:”kết nối lỗi,bạn muốn tải lại chức năng?”   ->chọn yes:tải lại chức năng.  ->Chọn no:quay lại chức năng chính. |

##### ô tả chức năng tạo khách hàng.

|  |  |
| --- | --- |
| **Tiêu đề** | **Nội dung** |
| **Tên chức năng** | **Tạo khách hàng.** |
| **ID chức năng** | **UC02** |
| **Mô tả** | Chức năng này mô tả người quản lí sau đã đăng nhập vào hệ thống cho phép người quản lí thêm khách hàng hoặc xem chi tiết về khách hàng. |
| **Actor** | Khách hàng. |
| **Organizational**  **Benefits** | Người quản lý tạo mới khách hàng. |
| **Triggers** | Người quản lí:chọn manage customer ->chọn create customer->nhập thông tin khách hàng->lưu. |
| **Tiền điều kiện** | Người quản lí đã đăng nhập. |
| **Luồng sự kiện chính.** | Chức năng này bắt đầu khi người quản lí muốn tạo khách hàng.   1. Khi người quản lí muốn tạo mới khách hàng hệ thống sẽ hiển thị mẫu hợp đồng để tạo khách hàng. 2. Người quản lí nhập thông tin của khách hàng và đồng ý tạo. 3. Người quản lí chọn một trong 2 chức năng:   -xem thông tin chi tiết của khách hàng.  -Quay lùi lại menu chính.   1. Hệ thống đưa ra chi tiết thông tin về khách hàng. 2. Người quản lí có thể lùi về menu chính để chọn chức năng khác. 3. Kết thúc chức năng. |
| **Luồng sự kiện phát sinh.** | Luồng 1:thông không hợp lệ  (id chứa khoảng cách,kí tự hoặc trống)     1. Hiển thị kiểm tra lại thông tin đã nhập 2. Nhập lại thông tin khách hàng.   Luồng 2:khách hàng đã tồn tại  1. Hiển thị “khách hàng đã tồn tại”.  2. Nhập lại thông tin khách hàng. |
| **Ngoại lệ** | Ngoại lệ 1: kết nối lỗi   1. Hiển thị:”kết nối lỗi,bạn muốn tải lại chức năng?”   ->chọn yes:tải lại chức năng.  ->Chọn no:quay lại chức năng chính. |

##### Mô tả chức năng cập nhật

|  |  |
| --- | --- |
| **Tiêu đề** | **Nội dung** |
| **Tên chức năng** | **Cập nhật** |
| **ID chức năng** | **UC03** |
| **Mô tả** | Chức năng này mô tả người quản lí có thể cập nhật thông tin khách hàng nếu cần thiết. |
| **Actor** | Người dùng. |
| **Organizational**  **Benefits** | Người quản lí cập nhật thông tin khách hàng. |
| **Triggers** | Người quản lí:chọn manage customer ->chọn Update->nhập mã khách hàng->chỉnh sửa thông tin->lưu. |
| **Tiền điều kiện** | Người quản lí đã đăng nhập ,khách hàng đã được tạo, và muốn sửa thông tin khách hàng. |
| **Hậu điều kiện** | Thành công : Thông tin của khách hàng đã được cập nhật.  Lỗi : Thông tin của khách hàng chưa đc cập nhật. |
| **Luồng sự kiện chính.** | Chức năng này bắt đầu khi người quản lí muốn cập nhật khách hàng.   1. Khi người quản lí muốn cập nhật khách hàng, hệ thống sẽ hiển thị danh sách khách hàng. 2. Người quản lí chọn mã khách hàng. 3. Hệ thống đưa thông tin của khách hàng cần sửa. 4. Người quản lí chỉnh sửa thông tin khách hàng.   Lưu lại thông tin đã thay đổi. |
| **Luồng sự kiện phát sinh** | Luồng 1 : Nếu khách hàng chưa có trong hệ thống.   1. Hiển thị :ID không tồn tại 2. Nhập lại ID khách hàng.   Luồng 2: thông không hợp lệ   1. Hiển thị kiểm tra lại thông tin đã nhập 2. Nhập lại thông tin cần thay đổi.Tiếp tục thực hiện. |
|  |  |
| **Ngoại lệ** | Ngọại lệ 1:kết nối lỗi   1. Hiển thị:”kết nối lỗi,bạn muốn tải lại chức năng?”   ->chọn yes:tải lại chức năng.  ->Chọn no:quay lại chức năng chính. |

##### Mô tả chức năng hiển thị

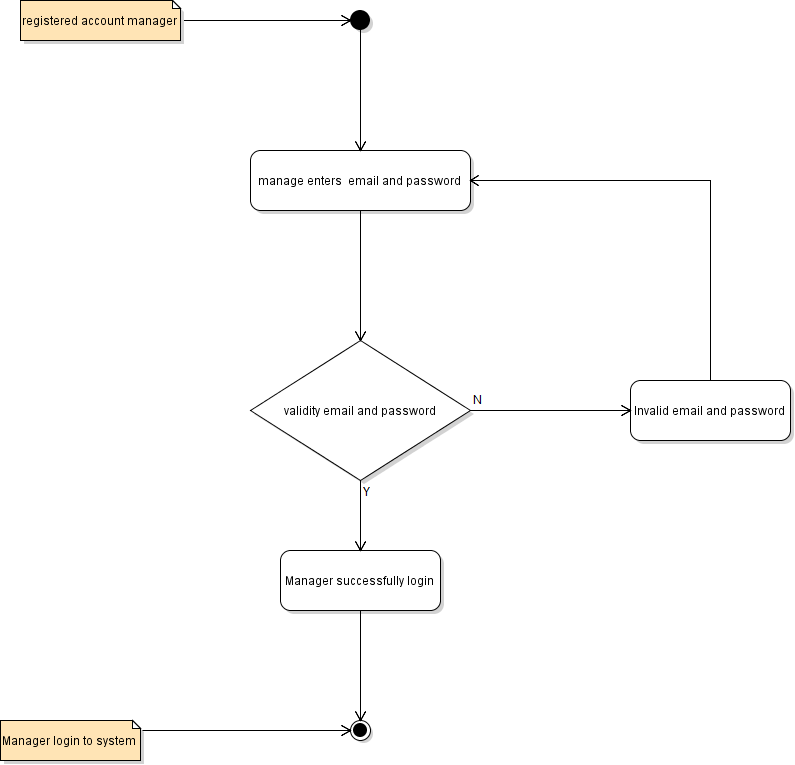
|  |  |
| --- | --- |
| **Tiêu đề** | **Nội dung** |
| **Tên chức năng** | **Hiển thị** |
| **ID chức năng** | **UC04** |
| **Mô tả** | Chức năng này mô tả người quản lí sau đã đăng nhập vào hệ thống cho phép người quản lí xem chi tiết về khách hàng. |
| **Actor** | Quản lí |
| **Organizational**  **Benefits** | Người quản lý muốn xem chi tiết về khách hàng. |
| **Triggers** | Người quản lí:chọn manage customer ->chọn display |
| **Tiền điều kiện** | Người quản lí đã đăng nhập,khách hàng đã được tạo , quản lí muốn xem thông tin về khách hàng. |
| **Hậu điều kiện** | Thành công: Hiển thị danh sách khách hàng chi tiết. |
| **Luồng sự kiện chính.** | Chức năng này bắt đầu khi người quản lí muốn xem chi tiết khách hàng.  1. Khi người quản lí muốn xem chi tiết khách hàng hệ thống sẽ hiển danh sách khách hàng trong hệ thống.   1. Người quản lí chọn xem khách hàng chi tiết 2. Hệ thống hiển thị thông tin chi tiết về khách hàng. 3. Người quản lí có thể lùi về menu chính để chọn chức năng khác. 4. Kết thúc chức năng. |
| **Luồng sự kiện phát sinh.** | Không có |
| **Ngoại lệ** | Ngọại lệ 1:kết nối lỗi   1. Hiển thị:”kết nối lỗi,bạn muốn tải lại chức năng?”   ->chọn yes:tải lại chức năng.  ->Chọn no:quay lại chức năng chính. |

##### Mô tả chức năng tạo hóa đơn

|  |  |
| --- | --- |
| **Tiêu đề** | **Nội dung** |
| **Tên chức năng** | **Tạo hóa đơn** |
| **ID chức năng** | **UC05** |
| **Mô tả** | Chức năng này mô tả người quản lí sau đã đăng nhập vào hệ thống cho phép người quản lí tạo hóa đơn. |
| **Actor** | Quản lí |
| **Organizational**  **Benefits** | Người quản lý tạo hóa đơn. |
| **Triggers** | Người quản lí:chọn tạo hóa đơn. |
| **Tiền điều kiện** | Người quản lí đã đăng nhập ,muốn tạo hóa đơn. |
| **Hậu điều kiện** | Thành công: Hóa đơn đã được tạo trong hệ thống. |
| **Luồng sự kiện chính.** | Chức năng này bắt đầu khi người quản lí muốn tạo hóa đơn.   1. Khi người quản lí muốn tạo hóa đơn hệ thống sẽ hiển thị mẫu hóa đơn. 2. Quản lí nhập thông tin của khách hàng. 3. Người quản lí chọn một trong 2 chức năng:   -xem thông tin chi tiết của hóa đơn.  -Quay lùi lại menu chính.   1. Hệ thống đưa ra chi tiết thông tin về hóa đơn. 2. Người quản lí có thể lùi về menu chính để chọn chức năng khác. 3. Kết thúc chức năng. |
| **Luồng sự kiện phát sinh.** | Luồng 1:thông không hợp lệ  (id chứa khoảng cách,kí tự hoặc trống)     1. Hiển thị kiểm tra lại thông tin đã nhập 2. Nhập lại thông tin khách hàng.   Luồng 2:Hóa đơn đã tồn tại   1. Hiển thị “hóa đơn đã tồn tại”.   Nhập lại thông tin hóa đơn. |
| **Ngoại lệ** | Ngọại lệ 1:kết nối lỗi   1. Hiển thị:”kết nối lỗi,bạn muốn tải lại chức năng?”   ->chọn yes:tải lại chức năng.  ->Chọn no:quay lại chức năng chính. |

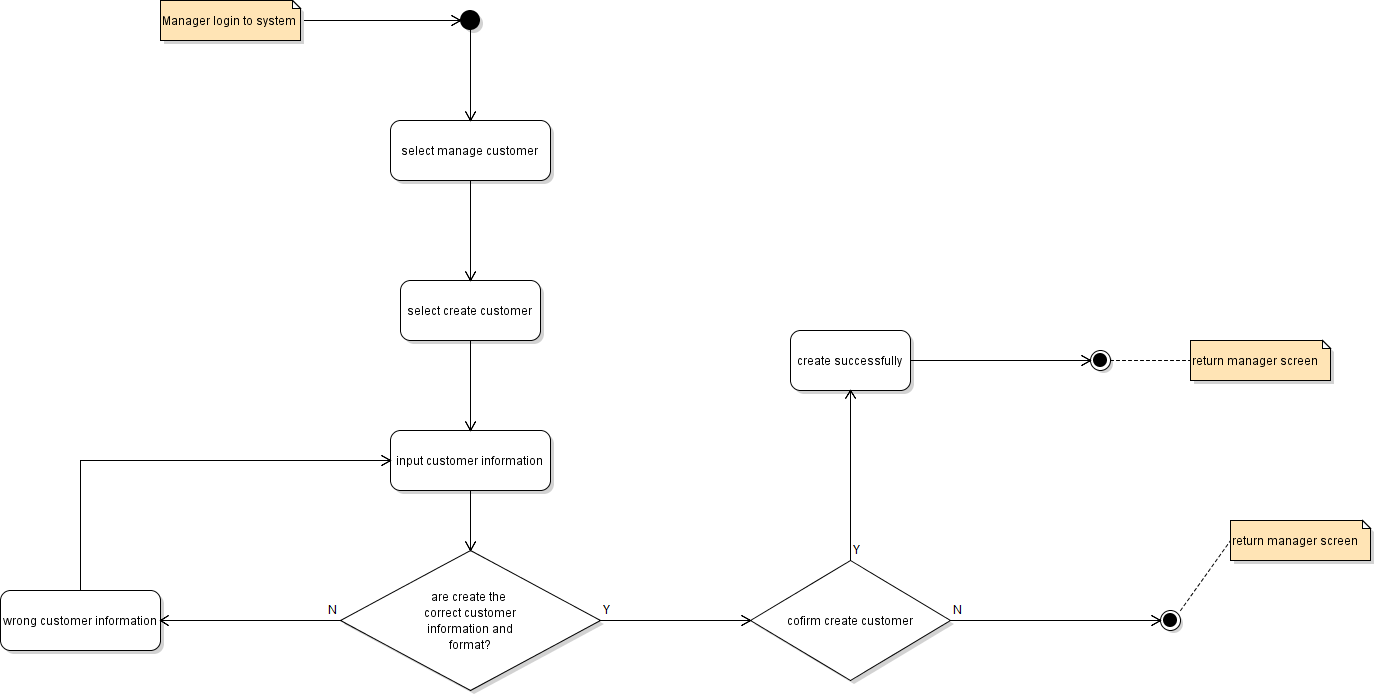
### Sơ đồ hoạt động (Activity Diagram)

#### Đăng nhập (Login)

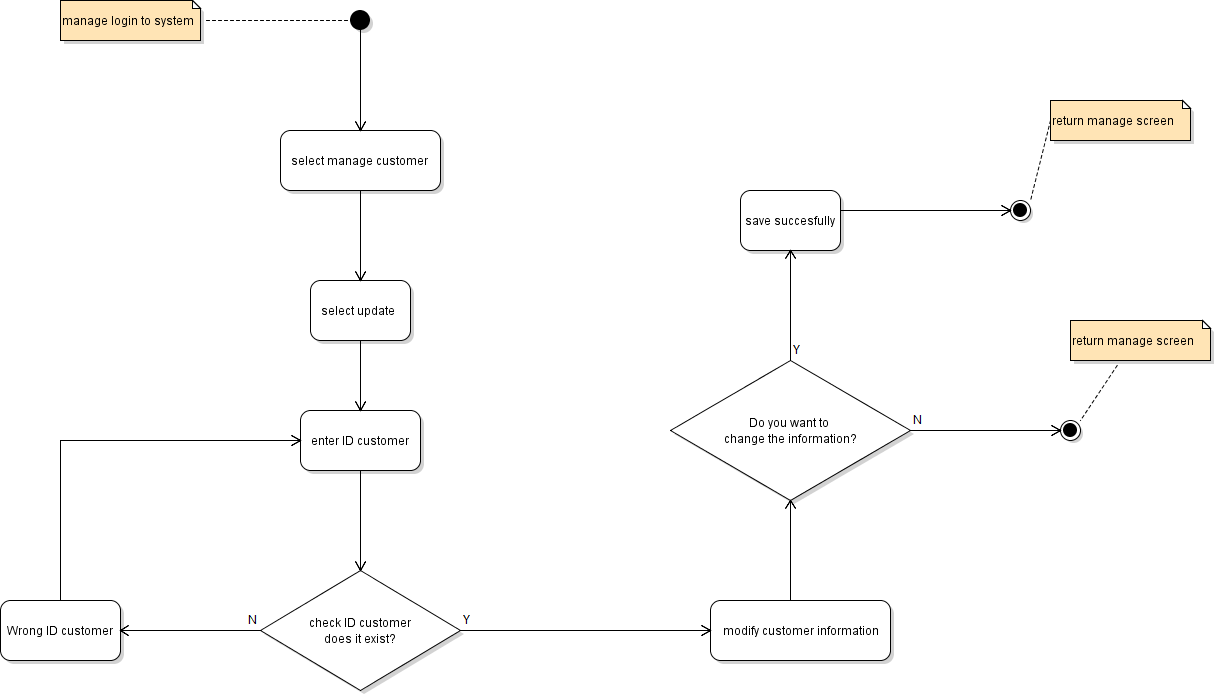


#### 

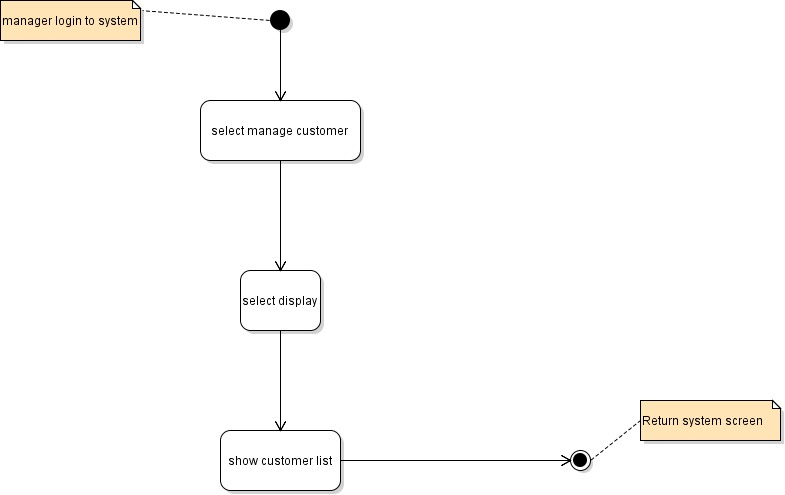
#### Tạo khách hàng (create customer)

****

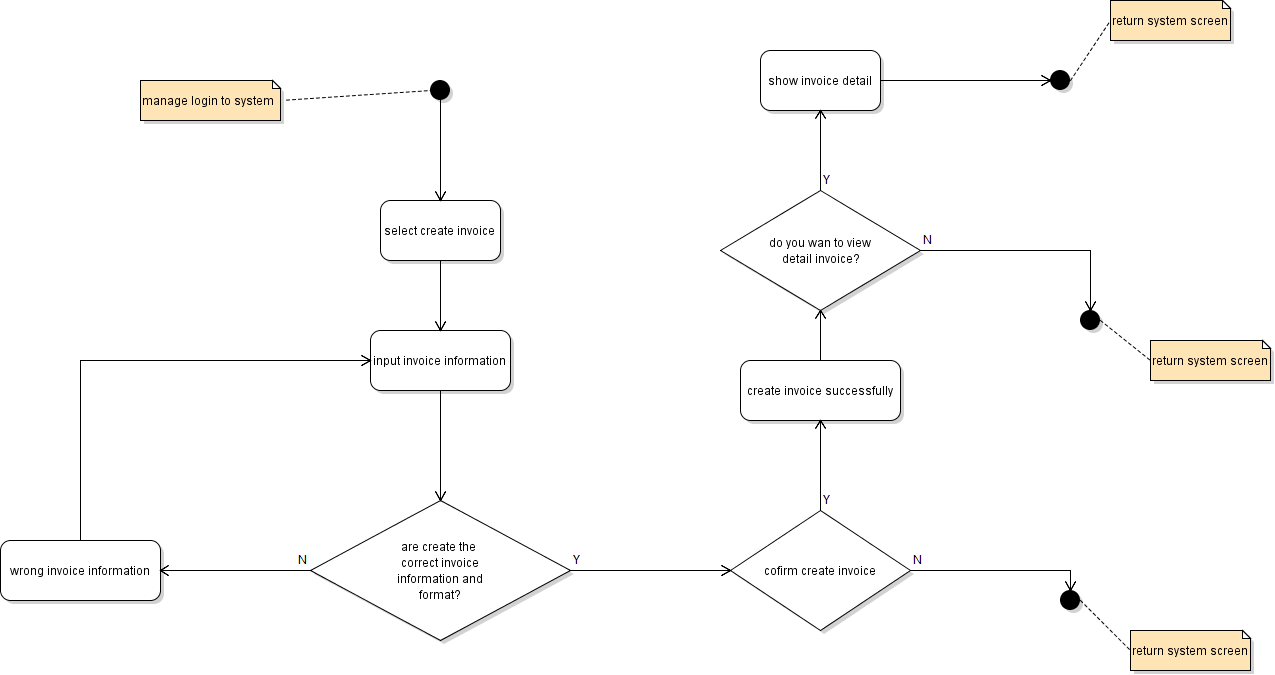
#### Cập nhật khách hàng(update customer)



#### Hiển thị khách hàng(Display)

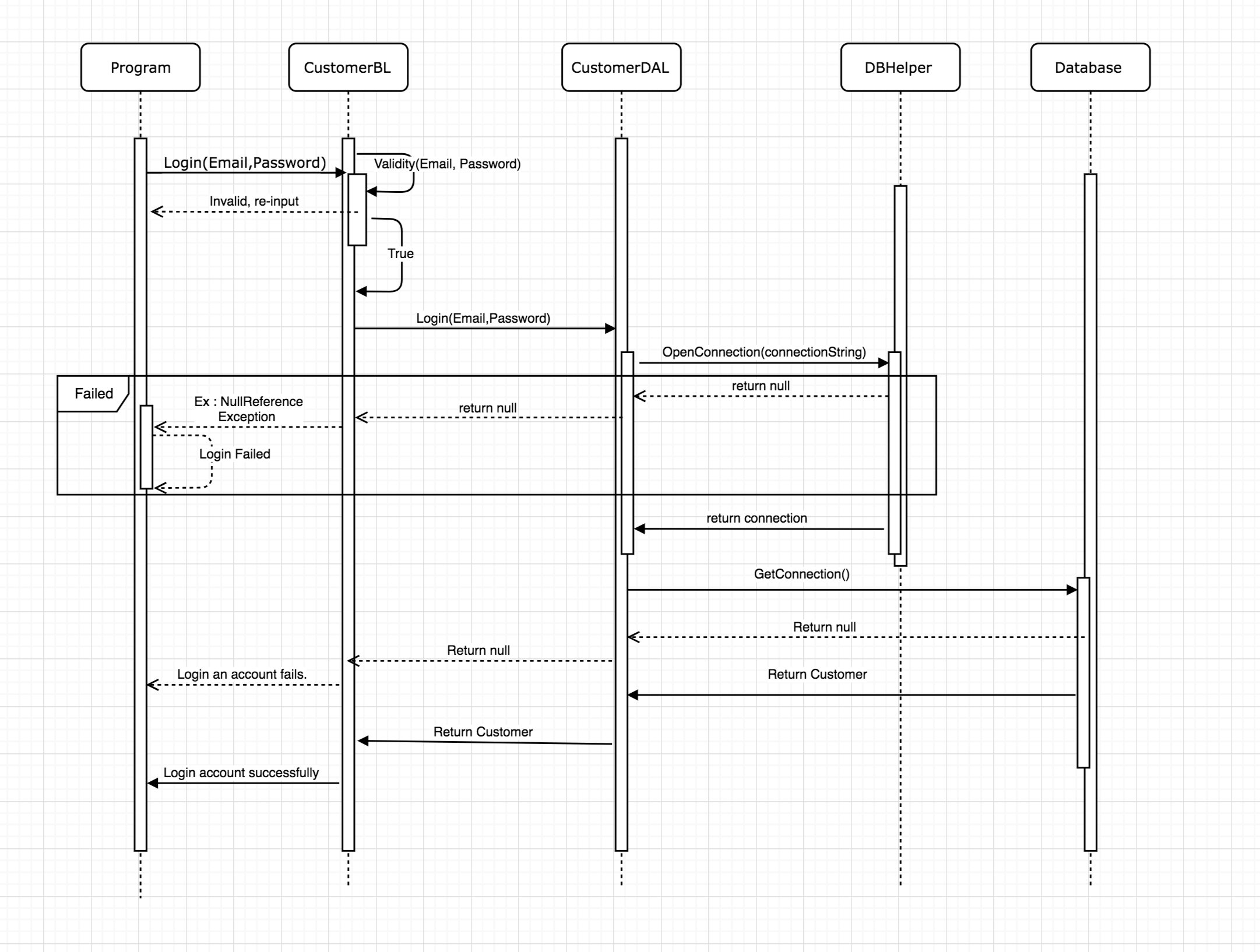


#### Tạo hóa đơn (Create invoice)

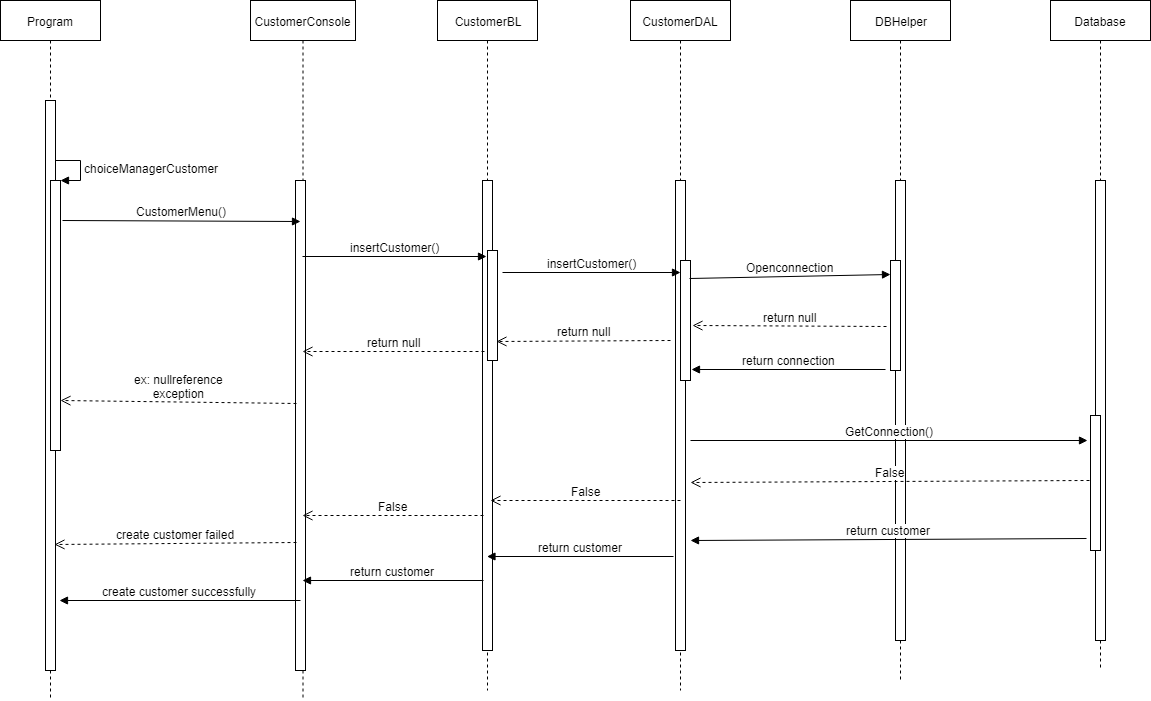


### Sequence Diagram

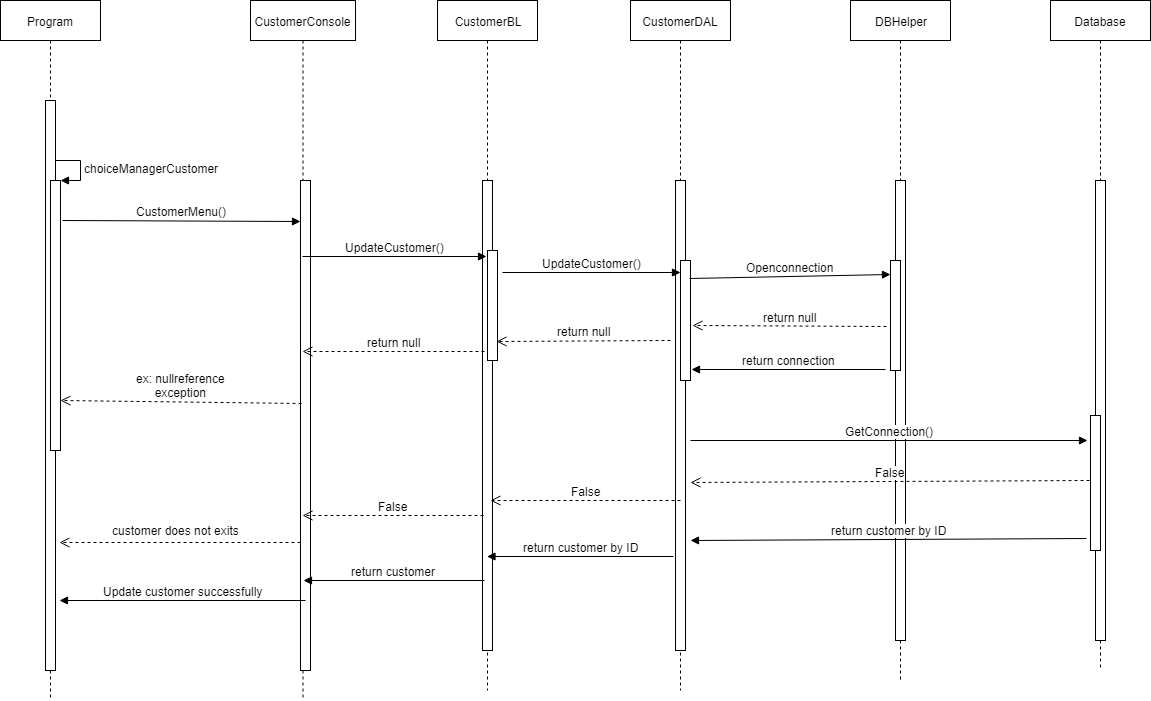
#### Đăng nhập (Login)



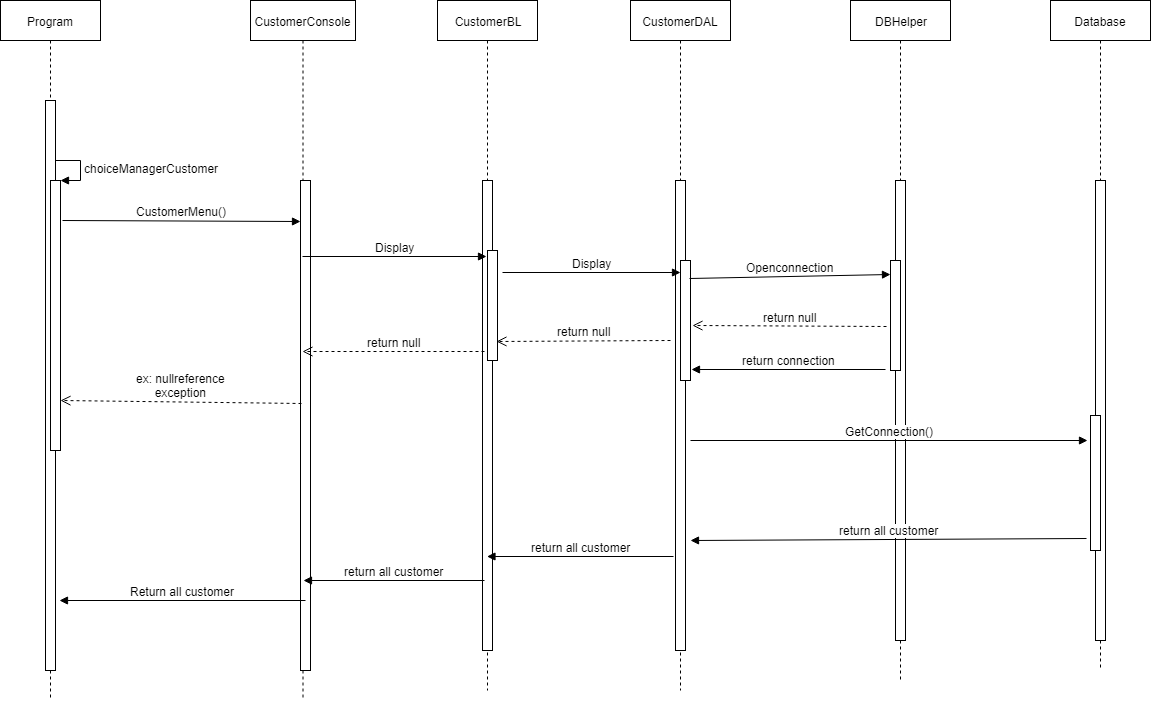
#### Tạo khách hàng



#### Cập nhật khách hàng



#### Hiển thị Khách hàng



#### Tạo hóa đơn

# Thiết kế chi tiết

### Thiết kế giao diện (GUI)

#### Mô tả chức năng giao diện

Chương trình gồm có các chức năng sau :

• **Đăng nhập.**

• **Quản lí khách hàng.**

• **Tạo hóa đơn.**

Giao diện và chi tiết chức năng.

Chương trình có menu chính như bên dưới và khi chọn các lựa chọn thì chương trình sẽ hoạt động như sau.

<https://codeshare.io/2Kv8MP> Link thiết kế giao diện.

#### Thiết kế giao diện

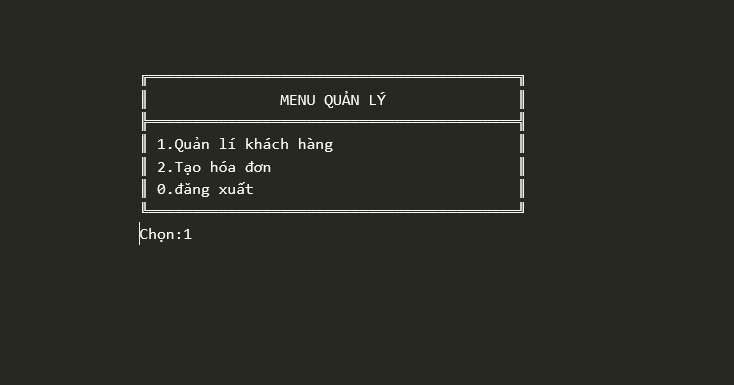
##### Giao diện đăng nhập

Chọn 1, chức năng đăng nhập sẽ như sau

****

##### Giao diện hệ thống quản lí nước.

**Đăng nhập** vào hệ thống sẽ hiển thị menu như sau :

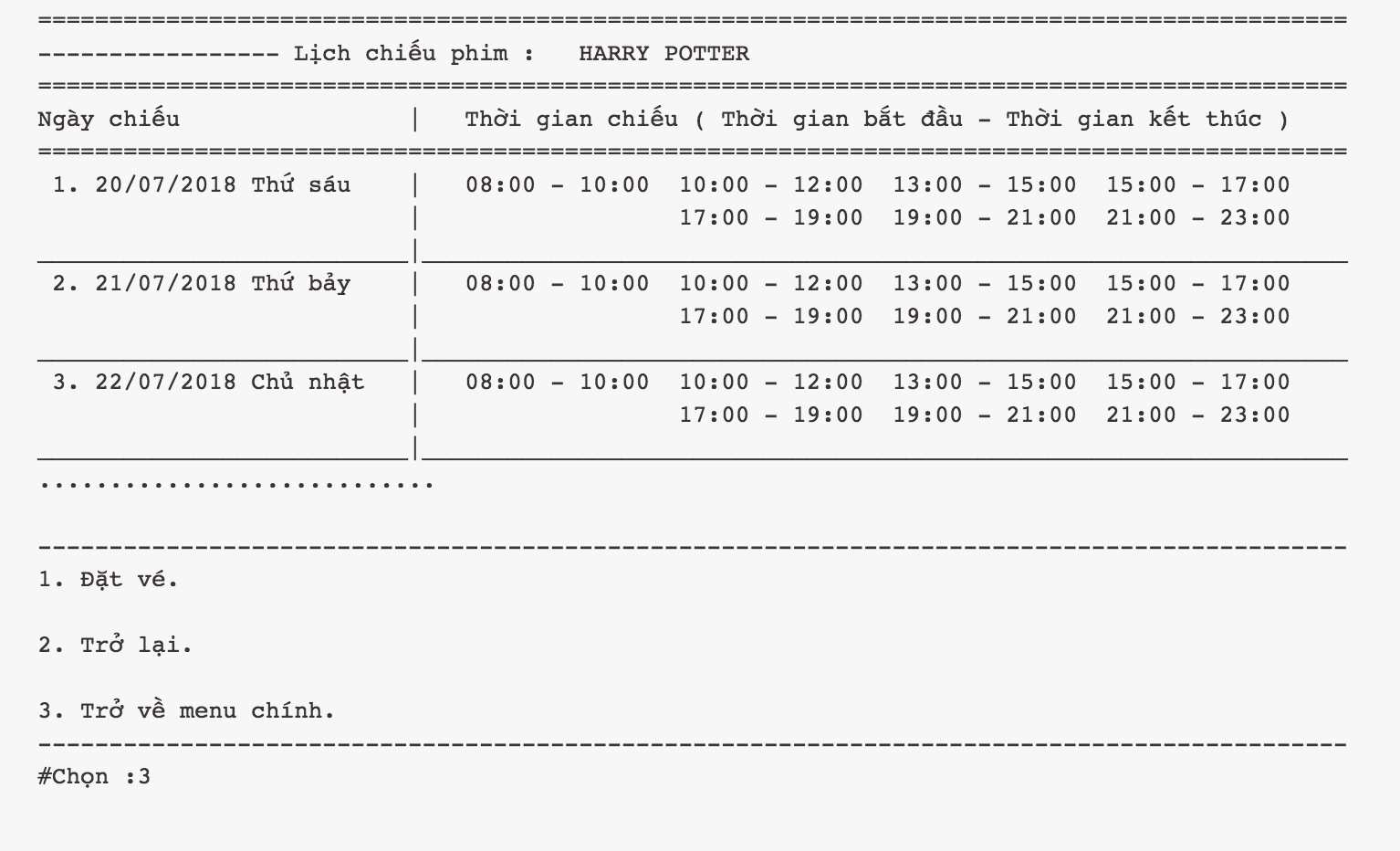


##### Giao diện quản lí khách hàng

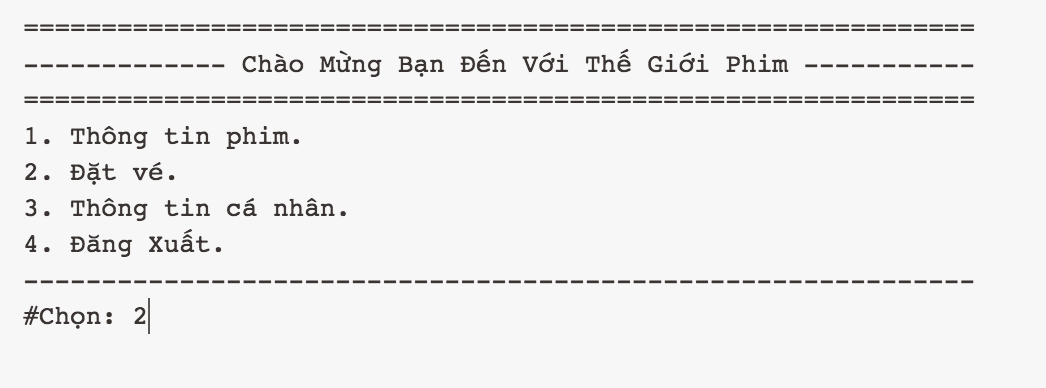
Tiếp tục chọn chức năng 1, sẽ hiển thị menu sau:



Tiếp tục chọn 1 **Tạo mới khách hàng** sẽ hiển thị như sau.

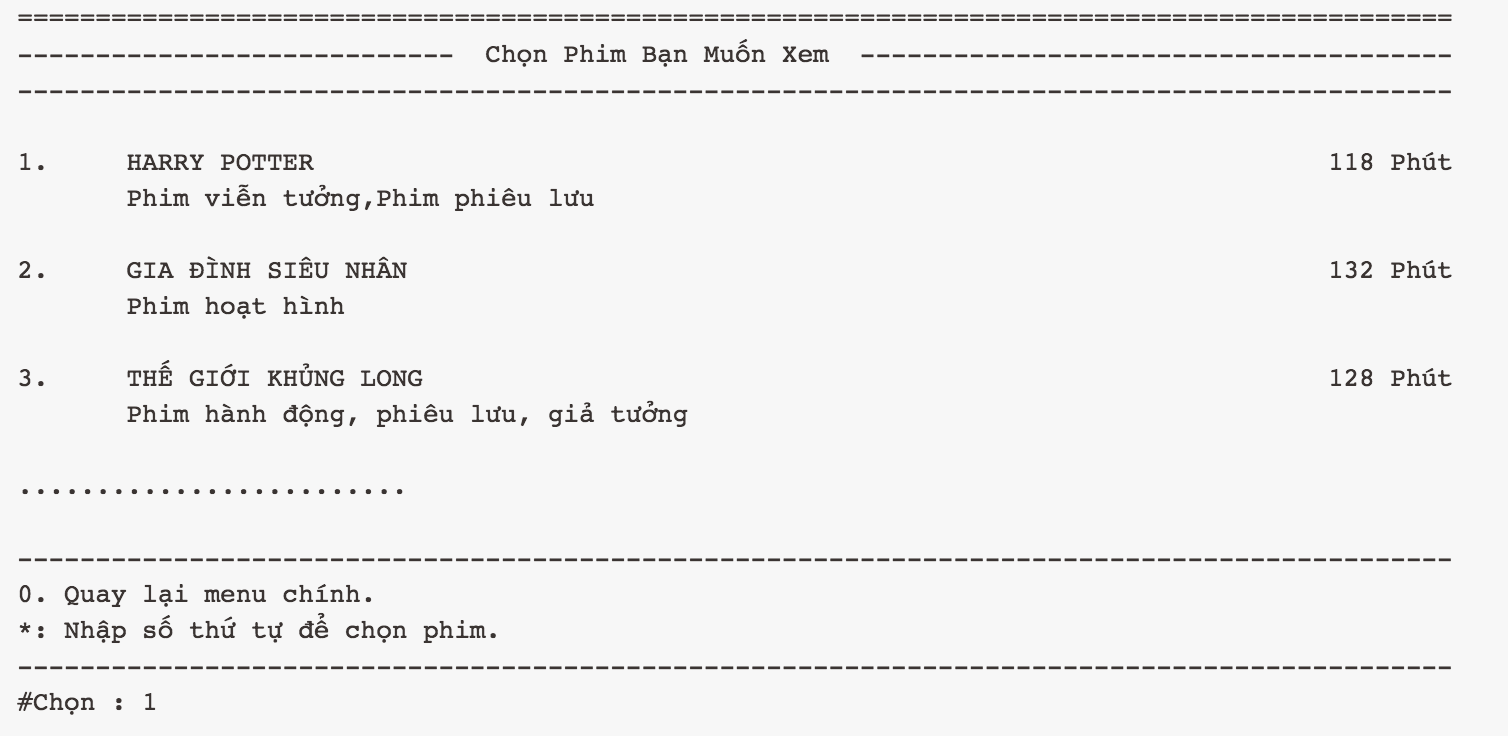


Trở lại với Menu chính khi vừa **Đăng Nhập** xong :

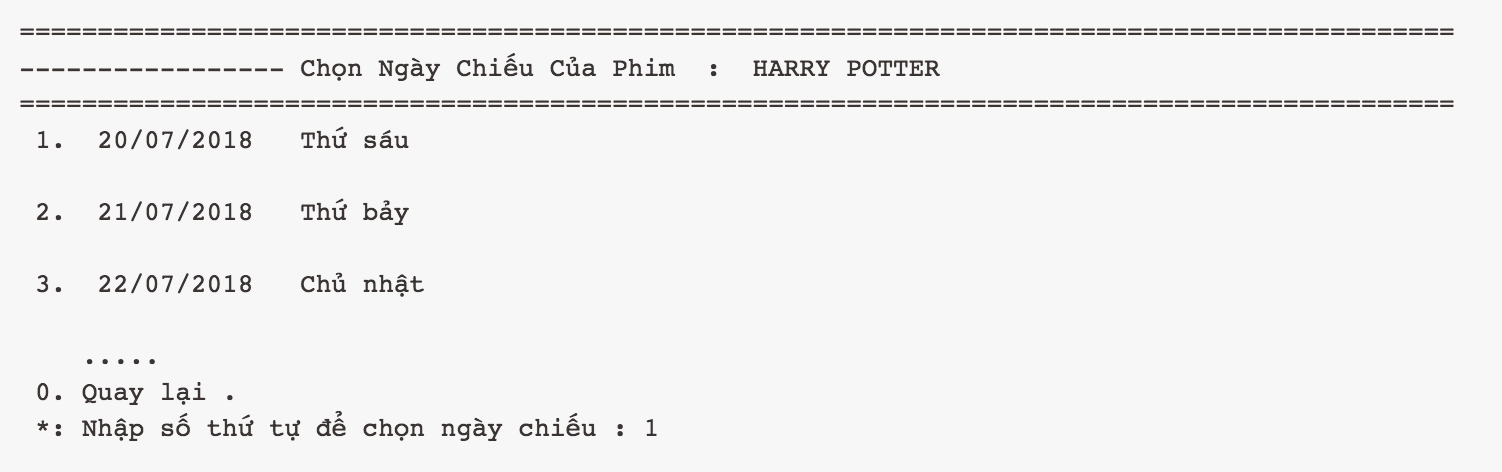
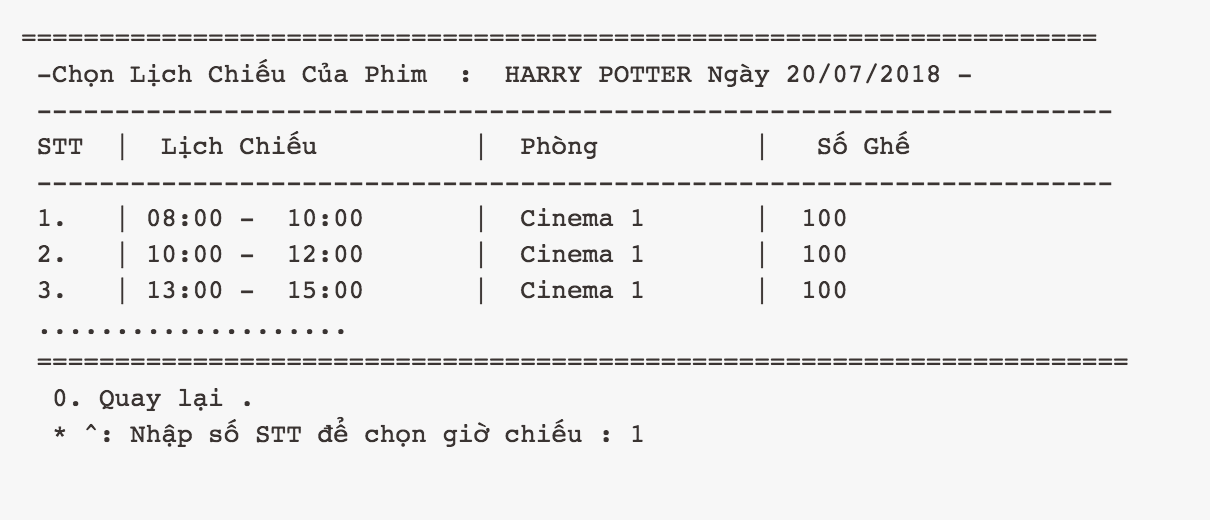


##### Giao diện Đặt vé

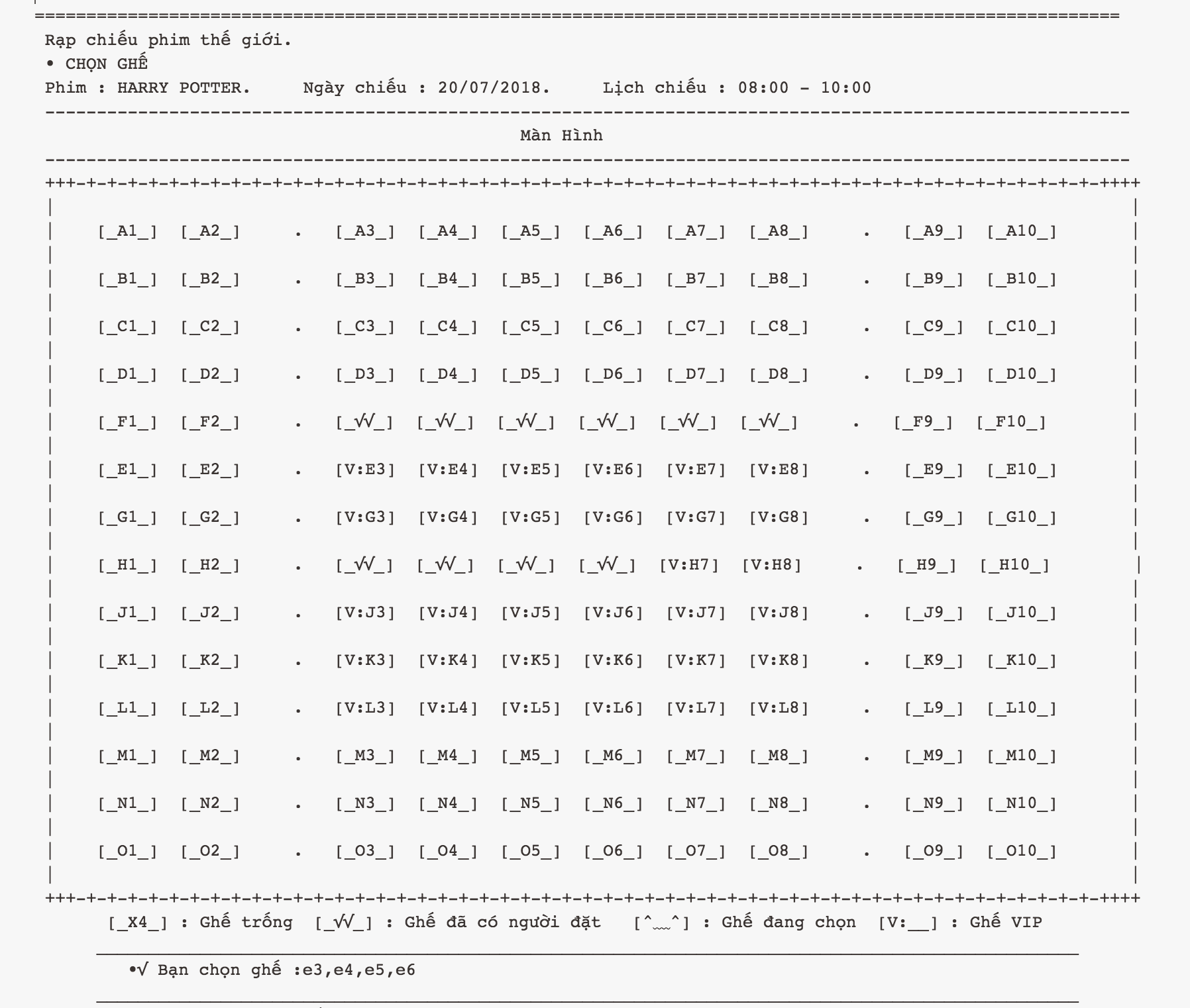
Chọn 2, **Đặt Vé** giao diện sẽ hiển thị như sau.

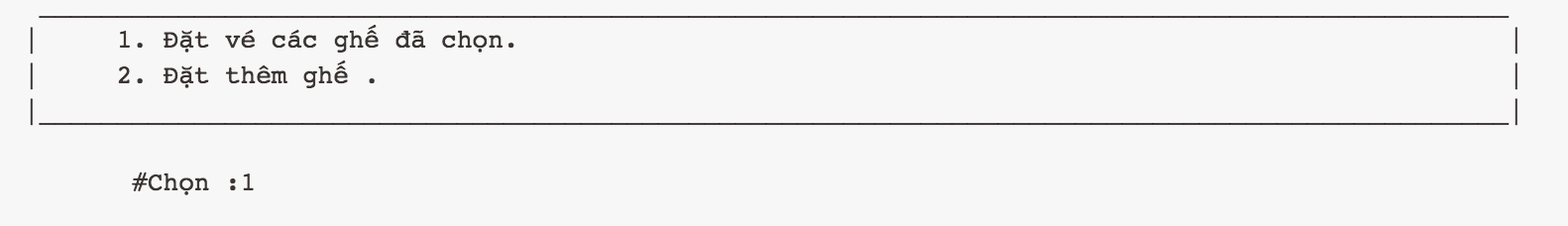


* Giao diện hiển thị lịch chiếu phim

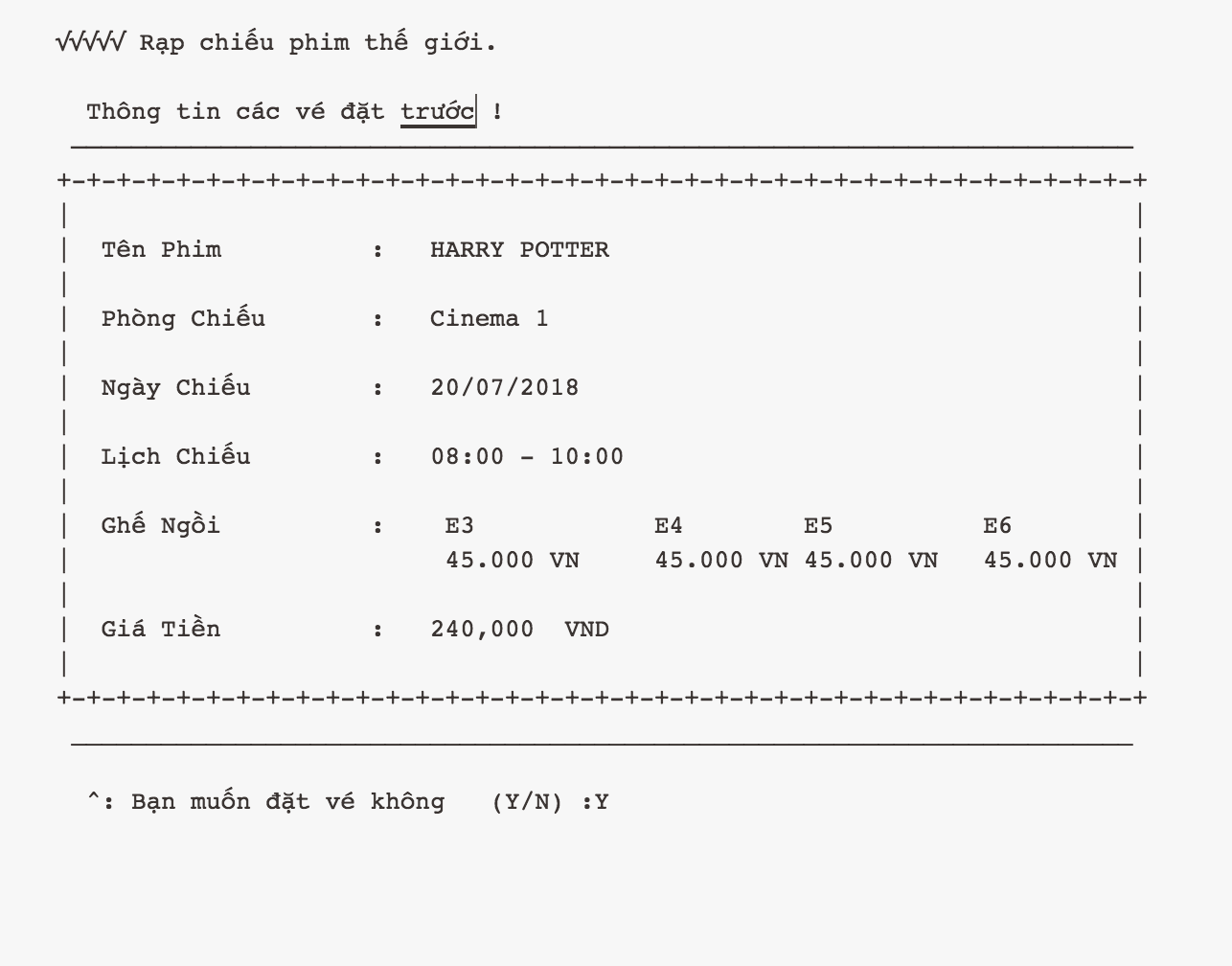


##### Giao diện đặt ghế





Chọn 1: **Đặt vé các ghế đã chọn** !

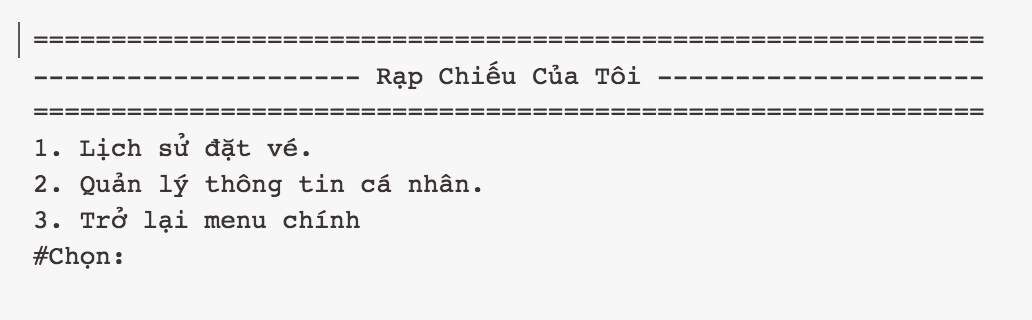


##### Thông tin vé đặt trước

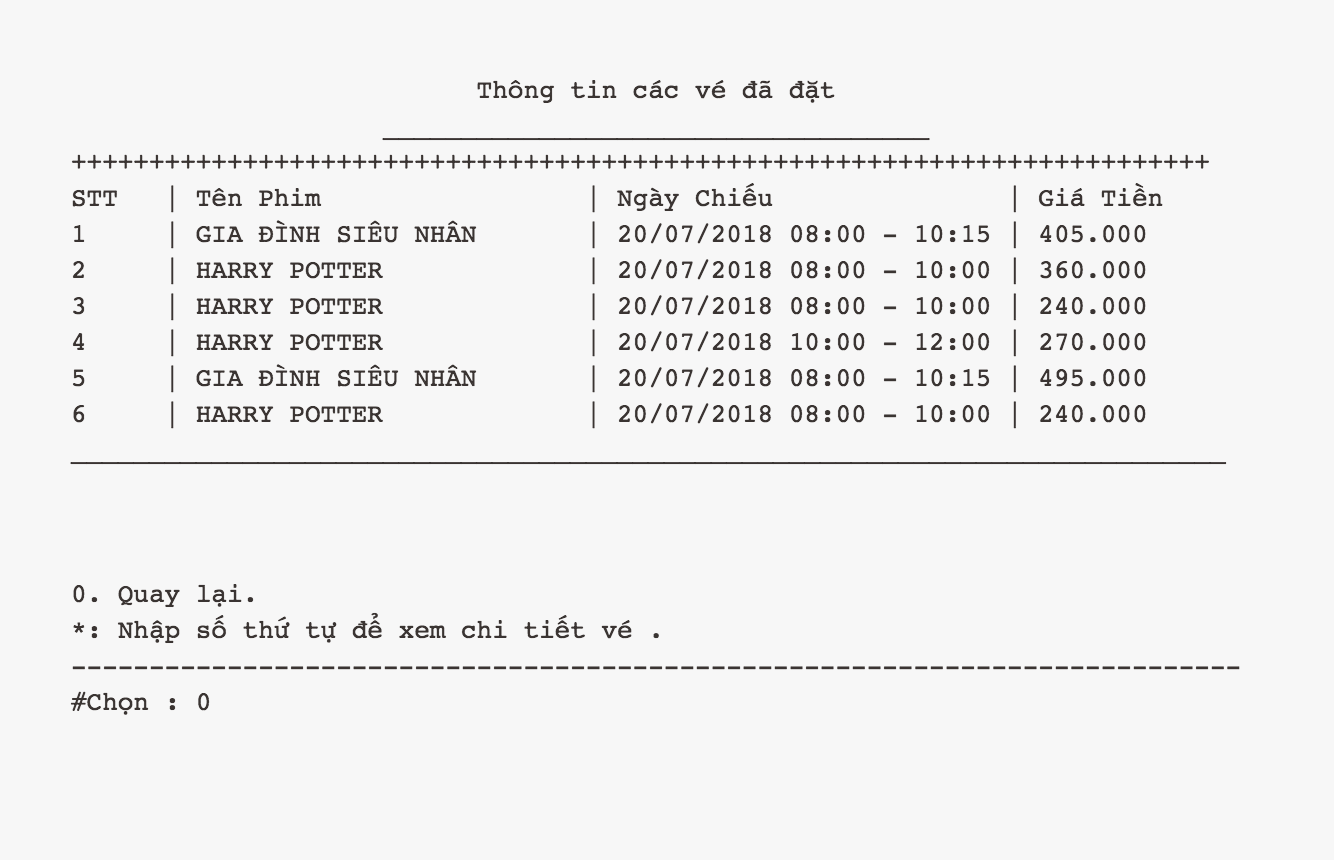


Chọn 2 quay lại menu chính.

##### Giao diện quản lý thông tin cá nhân



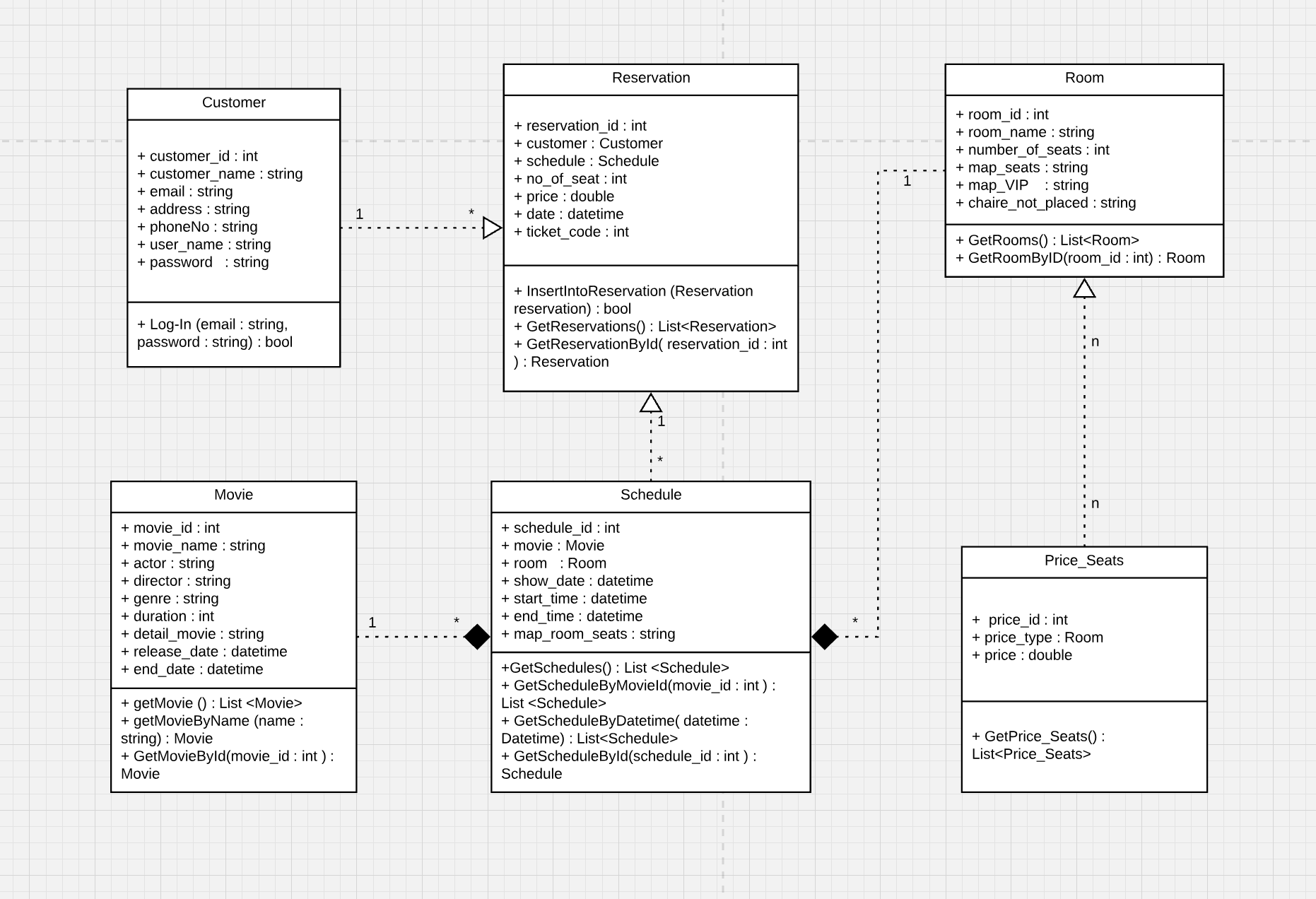
Chọn 1:



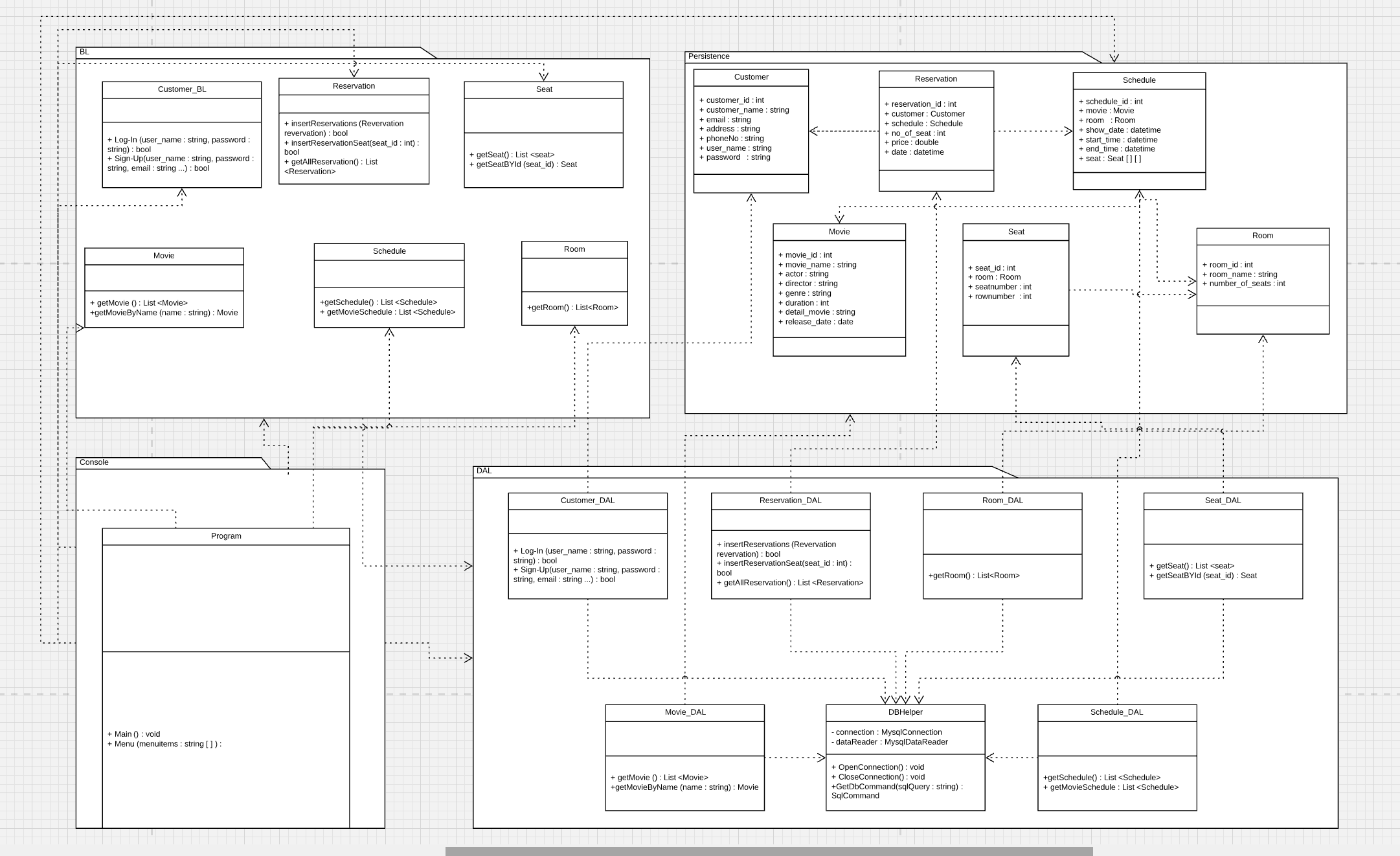


### Thiết kế mã (Class Diagram)

#### Class Diagram

****

#### Mô hình 3 tầng

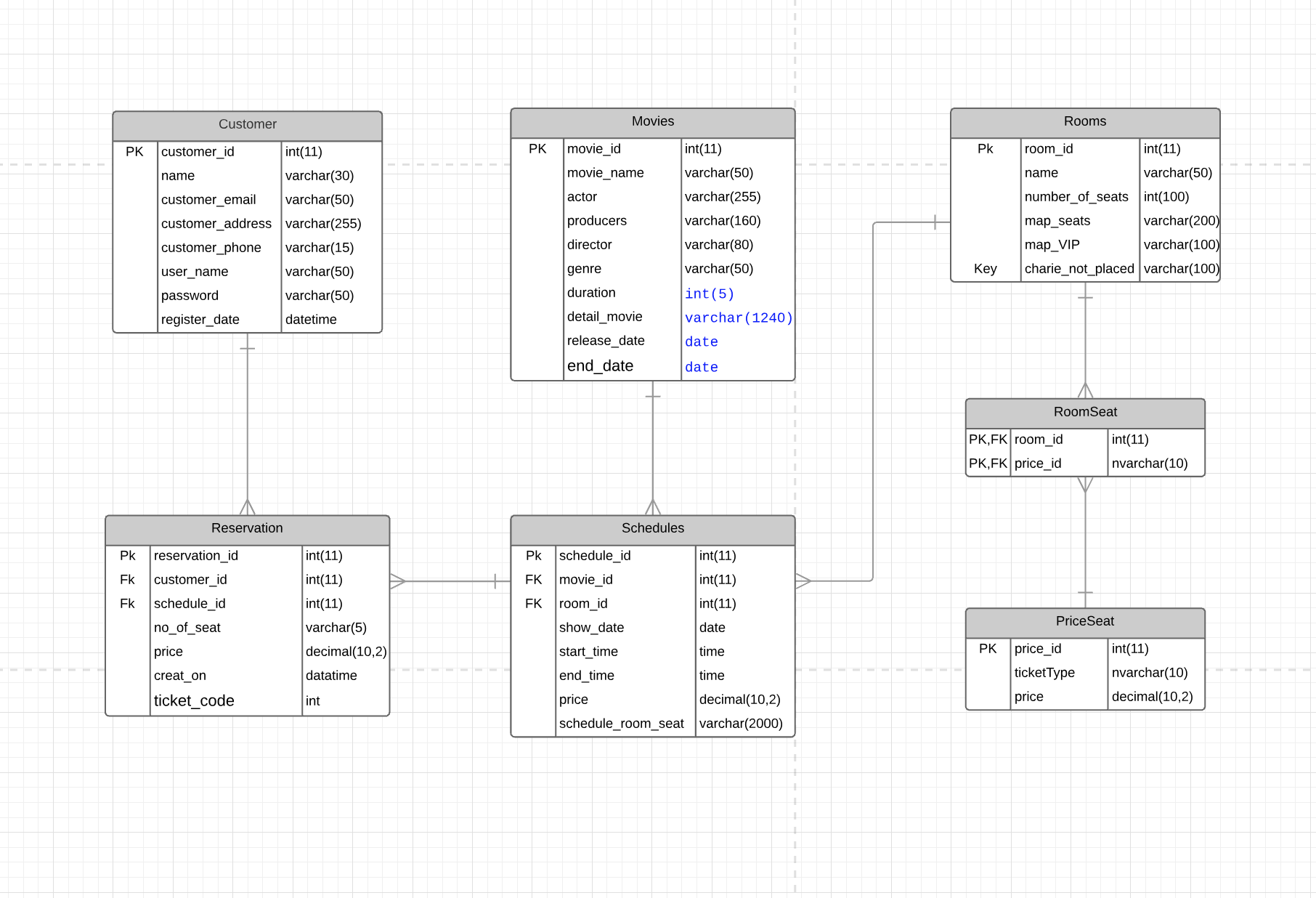


### Thiết kế CSDL (Database)

#### Entity Relationship Diagram

#### 

#### Thiết kế CSDL chi tiết



|  |  |  |  |
| --- | --- | --- | --- |
| **Customer** | | | |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| customer\_id | int(11) | Primary Key, Autoincrement  UNIQUE |  |
| name | varchar(30) | NOT NULL |  |
| customer\_email | varchar(50) | NOT NULL |  |
| customer\_address | varchar(255) | NOT NULL | Địa chỉ. |
| customer\_phone | varchar(15) |  | User Email |
| user\_name | varchar(50) | NOT NULL | Tên hiển thị |
| password | varchar(50) | NOT NULL | Mật khẩu đăng nhập. |
| register\_date | datetime | NOT NULL current\_timestamp |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **Reservation** | | | |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| reservation\_id | int(11) | Primary Key, Autoincrement  UNIQUE |  |
| customer\_Id | int(11) | NOT NULL | Là khoá ngoại tham chiếu đến **id** của **Customer.** |
| schedule\_id | int(11) | NOT NULL | Là khoá ngoại tham chiếu đến **id** của **Schedule** |
| no\_of\_seat | varchar(5) | NOT NULL | Số ghế ngồi. |
| price | decima(10,2) | Not NULL | Thành tiền. |
| create\_on | datetime | current\_timestamp | Thời gian tạo. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Movies** | | | |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| movie\_id | int(11) | Primary Key, Autoincrement  UNIQUE |  |
| movie\_name | varchar(50) | NOT NULL, UNIQUE | Tên của phim. |
| actor | varchar(255) | NOT NULL | Diễn viên trong phim. |
| producers | varchar(160) | NOT NULL | Nhà sản xuất. |
| director | varchar(80) | NOT NULL | Đạo diễn. |
| genre | varchar(50) | NOT NULL | Thể loại. |
| detail\_movie | varchar(1240) | NOT NULL | Nội dung của phim. |
| duration | int(5) | NOT NULL | Thời gian của phim. |
| release\_date | date | NOT NULL | Thời gian khởi chiếu. |
| end\_date | date | NOT NULL | Thời gian kết thúc. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Schedules** | | | |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| schedule\_id | int(11) | Primary Key, Autoincrement  UNIQUE |  |
| room\_id | int(11) | NOT NULL | Là khoá ngoại tham chiếu đến **id** của bảng **Rooms** |
| movie\_id | int(11) | NOT NULL | Là khoá ngoại tham chiếu đến **id** của bảng **Movie** |
| show\_date | date | NOT NULL | Ngày chiếu của phim. |
| start\_time | time | NOT NULL | Là khoá ngoại tham chiếu tới id của bảng **Cinemas**. |
| end\_time | time | NOT NULL | Là khoá ngoại tham chiếu tới id của **Movie** |
| price | decimal(10,2) | NOT NULL | Giá. |
| schedule\_room\_seat | varchar(2000) | NOT NULL | Bản đồ ghế ngồi. |

|  |  |  |  |
| --- | --- | --- | --- |
| **Rooms** | | | |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| **room\_id** | int(11) | Primary Key, Autoincrement  UNIQUE |  |
| room\_name | varchar(50) | NOT NULL, UNIQUE | Tên phòng chiếu. |
| number\_of\_seats | int(100) | NOT NULL | Sức chứa của phòng. |
| map\_seats | varchar(200) | NOT NULL | Bản đồ ghế của phòng. |
| map\_VIP | varchar(200) |  | Ghế VIP của phòng |
| chaire\_not\_placed | varchar(200) |  | Ghế lỗi không thể đặt |

|  |  |  |  |
| --- | --- | --- | --- |
| **Price\_Seats** | | | |
| **Column Name** | **Data Type** | **Constraints** | **Description** |
| **price\_id** | int(11) | Primary Key, Autoincrement  UNIQUE |  |
| price\_Type | varchar(10) | NOT NULL, UNIQUE | Kiểu giá. |
| price | double(10,2) | NOT NULL | Giá ghế. |

# Kiểm thử

### 4.1. Kiểm thử tầng DAL

#### Class CustomerDAL

|  |  |
| --- | --- |
| **ID** | Test\_Case 1 |
| **Tên kiểm thử** | TestLogin |
| **Mô tả kiểm thử** | Kiểm thử phương thức Login() trong Class CustomerDAL |
| **Tiền điều kiện** | Có dữ liệu trong cơ sở dữ liệu , đã khai báo phương thức login() trong Class CustomerDAL tầng DAL. |
| **Đầu vào kiểm thử** | 1. login(“valentinolivgr@gmail.com", "123456”) 2. login("abc d e f'", "123") 3. login(",,'1`12", "as''dfg") |
| **Đầu ra kiểm thử** | 1. Trả về NOTNULL vào giao diện người dùng. 2. Trả về NULL (Email, Password đã nhập không đúng). 3. Trả về NULL(Email, Password không được nhập ký tự đặc biệt). |
| **Các bước kiểm thử** | 1. Nhập Email, Password 2. Email, Password là tham số truyền vào phương thức Login() trong Class CustometDAL 3. Trả về NULL nếu trong cơ sở dữ liệu không có Email và Password như đã nhập.   Trả về một Customer ( NOT NULL ) nếu tìm thấy trong cơ sở dữ liệu có Email và Password giống như đã nhập. |

#### Kiểm thử Class MovieDAL

|  |  |
| --- | --- |
| **ID** | Test\_Case 2.1 |
| **Tên kiểm thử** | Test GetMovieDAL |
| **Mô tả kiểm thử** | Kiểm thử phương thức GetMovies trong tầng DAL |
| **Tiền điều kiện** | Có dữ liệu trong cơ sở dữ liệu, đã viết hàm GetMovies trong tầng DAL. |
| **Đầu vào kiểu thử** | 1. Lấy ra một List<Movie> trong cơ sở dữ liệu. 2. Lấy ra một Movie ngẫu nhiên trong cơ sở dữ liệu. 3. Lấy ra vị trí Top của Movie trong cơ sở dữ liệu. 4. Lấy ra Movie Bottom trong cơ sở dữ liệu |
| **Đầu ra kiểm thử** | 1. Trả về một List<Movie> (NotNull). 2. Trả về một Movie ngẫu nhiên ( Not Null ).Kiểm tra Movide Rand trong cơ sở dữ liệu có trong List<Movie> hay không ? (Contains). 3. Trả về Movie Top (NotNull). So sánh Movie Top trong cơ sở dữ liệu có giống Movie đầu tiên trong List<Movie> vừa lấy ?(True). 4. Trả về Movie Bottom (Not Null ). So sánh Movie Bottom trong cơ sở dữ liệu có giống Movie cuối cùng trong List<Movie> ? (True). |
| **Các bước kiểm thử** | 1. Gọi hàm GetMovies trong tầng DAL lấy ra 1 List<Movie> 2. Viết một query để lấy một Movie ngẫu nhiên trong cơ sở dữ liệu. 3. Viết một query để lấy Movies Top trong cơ sở dữ liệu. 4. Viết một query để lấy Movies Bottom trong cơ sở dữ liệu |

# 

# 

|  |  |
| --- | --- |
| **ID** | Test\_Case 2.2 |
| **Tên kiểm thử** | Test Get Movie By Name |
| **Mô tả kiểm thử** | Kiểm thử phương thức GetMovieByName trong Class Movie\_DAL |
| **Tiền điều kiện** | Có dữ liệu trong cơ sở dữ liệu , đã khai báo phương GetMovieByName() trong DAL. |
| **Đầu vào kiểm thử** | 1. GetMovieByName(“HARRY POTTER”); 2. GetMovieByName(“”); 3. GetMovieByName(“Khong co ten phim”); |
| **Đầu ra kiểm thử** | 1. Trả về Movie có tên Harry Potter ( NOT NULL ). 2. Trả về NULL , (Nhập tên phim không được để trống) 3. Trả về NULL , ( Không tìm thấy tên phim trong DATABASE) |
| **Các bước kiểm thử** | 1. Nhập tên phim. 2. Tìm kiếm tên Phim đấy có trong cơ sở dữ liệu không ? Bằng phương thức GetMovieByName(name). 3. - Nếu Name = “” hoặc Name = NULL : Trả về NULL , tham số đầu vào không được rỗng.   - Nếu tìm kiếm trong cơ sở dữ liệu không thấy tên phim như vừa nhập thì trả về NULL.  - Nếu tìm thấy trong cơ sở dữ liệu có tên giống với tên vừa nhập thì trả về một Movie (NOT NULL) |

# 

|  |  |
| --- | --- |
| **ID** | Test\_Case 2.3 |
| **Tên kiểm thử** | Test Get Movie By ID |
| **Mô tả kiểm thử** | Kiểm thử phương thức GetMovieById() trong DAL |
| **Tiền điều kiện** | Có dữ liệu trong cơ sở dữ liệu , đã khai báo phương thức GetMovieById() trong DAL. |
| **Đầu vào kiểm thử** | 1. GetMovieById(1); 2. GetMovieById(0) |
| **Đầu ra kiểm thử** | 1. Trả về Movie có ID = 1 ( NOT NULL ) 2. Trả về NULL (Không tìm thấy movie\_id = 0 trong Database) |
| **Các bước kiểm thử** | 1. Nhập ID của phim 2. Tìm kiếm ID của phim có trong cơ sở dữ liệu không ? Bằng phương thức GetMovieById(movie\_id); 3. - Nếu movie\_id bằng 0 hoặc movie\_id bằng null. Trả về Null tham số đầu vào không được bằng 0 hoặc null.   - Nếu tìm kiếm trong cơ sở dữ liệu không có movie\_id vừa nhập thì trả về Null.  - Nếu tìm kiếm trong cơ sở dữ liệu có MovieID vừa nhập thì trả về một Movie (NOT NULL) |

#### Kiểm thử Class Room

|  |  |
| --- | --- |
| **ID** | Test\_Case 3.1 |
| **Tên kiểm thử** | TestGetRoomDAL |
| **Mô tả kiểm thử** | Kiểm thử phương thức GetRooms trong tầng DAL |
| **Tiền điều kiện** | Có dữ liệu trong cơ sở dữ liệu, đã viết phương thức Get Room trong tầng DAL. |
| **Đầu vào kiểu thử** | 1. Lấy ra một List<Room> trong Database. 2. Lấy ra một Room ngẫu nhiên trong Database 3. Lấy ra vị trí Top của Room trong Database 4. Lấy ra Room Bottom trong Database |
| **Đầu ra kiểm thử** | 1. Trả về một List<Room> (NotNull). 2. Trả về một Movie Rand (NotNull ). Kiểm tra Room Rand trong Database có trong List<Room> hay không ? (Contains). 3. Trả về Room Top ( NOT NULL ). So sánh Room Top trong Database có giống Room đầu tiên trong List<Room> vừa lấy ?(True). 4. Trả về Room Bottom (NOT NULL ). So sánh Room Bottom trong Database có giống Room cuối cùng trong List<Movie> ? (True). |
| **Các bước kiểm thử** | 1. Gọi phương thức GetRooms trong tầng DAL lấy ra 1 List<Room>. 2. Viết một query để lấy một Room ngẫu nhiên trong Database. 3. Viết một query để lấy Room Top trong Database. 4. Viết một query để lấy Room Bottom trong Database |

|  |  |
| --- | --- |
| **ID** | Test\_Case 3.2 |
| **Tên kiểm thử** | TestGetRoomByID |
| **Mô tả kiểm thử** | Kiểm thử phương thức GetRoomById() trong DAL |
| **Tiền điều kiện** | Có dữ liệu trong cơ sở dữ liệu , đã khai báo phương thức GetRoomById() trong DAL. |
| **Đầu vào kiểm thử** | 1. GetRoomById(1); 2. GetRoomById(0) |
| **Đầu ra kiểm thử** | 1. Trả về Room có ID = 1 ( NOT NULL ) 2. Trả về NULL (Không tìm thấy room\_id = 0 trong Database) |
| **Các bước kiểm thử** | 1. Nhập ID phòng chiếu. 2. Tìm kiếm Room\_ID đấy có trong cơ sở dữ liệu hay không ? bằng phương thức GetRoomByID(Id) ; |

#### Kiểm thử Class Schedule

|  |  |
| --- | --- |
| **ID** | Test\_Case 4.1 |
| **Tên kiểm thử** | TestGetSchedules |
| **Mô tả kiểm thử** | Kiểm thử phương thức GetSchedules trong tầng DAL |
| **Tiền điều kiện** | Có dữ liệu trong cơ sở dữ liệu, đã viết phương thức GetSchedules trong tầng DAL. |
| **Đầu vào kiểu thử** | 1. Lấy ra một List<Schedule> trong Database. 2. Lấy ra một Schedule ngẫu nhiên trong Database. 3. Lấy ra vị trí Top của Schedule trong Database. 4. Lấy ra Schedule Bottom trong Database |
| **Đầu ra kiểm thử** | 1. Trả về một List<Schedule> (NotNull). 2. Trả về một Schedule Rand (NotNull).Kiểm tra Schedule Rand trong Database có trong List<Schedule> hay không ? (Contains). 3. Trả về Schedule Top (NotNull). So sánh Schedule Top trong Database có giống Schedule đầu tiên trong List<Schedule> vừa lấy ?(True). 4. Trả về Schedule Bottom (NotNull ). So sánh Schedule Bottom trong Database có giống Schedule cuối cùng trong List<Schedule> ? (True). |
| **Các bước kiểm thử** | 1. Gọi hàm GetSchedules trong tầng DAL lấy ra 1 List<Schedule>. 2. Viết một query để lấy một Schedule ngẫu nhiên trong Database. 3. Viết một query để lấy Schedule Top trong Database. 4. Viết một query để lấy Schedule Bottom trong Database |

# 

|  |  |
| --- | --- |
| **ID** | Test\_Case 4.2 |
| **Tên kiểm thử** | TestGetScheduleByIDSchedule |
| **Mô tả kiểm thử** | Kiểm tra phương thức GetScheduleByIDSchedule trong Class ScheduleDAL. |
| **Tiền điều kiện** | Có dữ liệu trong cơ sở dữ liệu , đã khai báo phương thức GetScheduleByIDSchedule () trong DAL. |
| **Đầu vào kiểu thử** | 1. GetScheduleByIdSchedule (1 ); 2. GetScheduleByIdSchedule (0); |
| **Đầu ra kiểm thử** | 1. Trả về Schedule có ID bằng 1 ( Not Null ) 2. Trả về Null, không tìm được ID bằng 0 trong Database |
| **Các bước kiểm thử** | 1. Nhập ID của lịch chiếu |

# 

|  |  |
| --- | --- |
| **ID** | Test\_Case 4.3 |
| **Tên kiểm thử** | TestGetScheduleByMovieID |
| **Mô tả kiểm thử** | Kiểm tra phương thức GetScheduleByMovieID trong Class ScheduleDAL. |
| **Tiền điều kiện** | Có dữ liệu trong cơ sở dữ liệu , đã khai báo phương thức GetScheduleByMovieID () trong DAL. |
| **Đầu vào kiểu thử** | 1. GetScheduleByMovieID (1 ); 2. GetScheduleByMovieID (0); |
| **Đầu ra kiểm thử** | 1. Trả về List<Schedule> có Movie ID bằng 1 ( Not Null ) 2. Trả về Null, không tìm được Movie ID bằng 0 trong Database |
| **Các bước kiểm thử** | 1. Nhập ID của Movie ID |

# 

|  |  |
| --- | --- |
| **ID** | Test\_Case 4.4 |
| **Tên kiểm thử** | Test Get Time By Movie ID and Date |
| **Mô tả kiểm thử** | Kiểm thử phương thức SelectTime() trong Class Shedule\_DAL |
| **Tiền điều kiện** | Có dữ liệu trong cơ sở dữ liệu , đã khai báo phương thức SelectTime() trong Class Schedule\_DAL. |
| **Đầu vào kiểm thử** | 1. DateTime timeStart = new DateTime(2018,7,20,0,0,0);   SelectTime(1,timeStart.ToString($“{timeStart:yyyy-MM-dd}”));  2. DateTime timeStart = new DateTime(2018,7,20,0,0,0);  SelectTime(1,timeStart.ToString($“{timeStart:dd-yyyy-dd}”));  3. DateTime timeStart = new DateTime(2018,7,20,0,0,0);  SelectTime(0,timeStart.ToString($“{timeStart:dd-yyyy-dd}”));  4. DateTime timeStart = new DateTime(2017,7,20,0,0,0);  SelectTime(1,timeStart.ToString($“{timeStart:yyyy-MM-dd}”));  5. SelectTime(1,””); |
| **Đầu ra kiểm thử** | 1. Trả về một List<Schedule> với Movie\_ID = 1 và DateTime là 2018-7-20 2. Trả về NULL (Nhập sai định dạng DateTime) 3. Trả về NULL (Không có Movie\_id = 0 trong cơ sở dữ liệu) 4. Trả về NULL (Không tìm thấy trong cơ sở dữ liệu bảng Schedule có movie\_ id = 1 và Datetime là 2017-7-20) 5. Trả về NULL không được để Datetime = null. |
| **Các bước kiểm thử** | 1. Nhập Movie\_id 2. Nhập Datetime |

|  |  |
| --- | --- |
| **ID** | Test\_Case 4.5 |
| **Tên kiểm thử** | TestSelectScheduleByMovieIDAndDateTime |
| **Mô tả kiểm thử** | Kiểm tra phương thức SelectScheduleByDateTimeAndMovieID trong Class ScheduleDAL. |
| **Tiền điều kiện** | Có dữ liệu trong cơ sở dữ liệu , đã khai báo phương thức SelectScheduleByDateTimeAndMovieID () trong DAL. |
| **Đầu vào kiểu thử** | 1. DateTime ShowDate = new DateTime(2018,7,20,0,0,0);   TimeSpan timeSpan = new TimeSpan(8,0,0);  string Date = ShowDate.ToString($“{ShowDate:yyyy-MM-dd};  string Time = string.Format ($"{timeSpan.Hours:D2}:{timeSpan.Minutes:D2}:  {timeSpan.Seconds:D2}");  • SelectScheduleByDateTimeAndMovieID(1,Date,Time);  2. DateTime ShowDate = new ShowDate(2017,7,20,0,0,0);  TimeSpan StartTime = new TimeSpan(1,0,0);  string Date = ShowDate.ToString($“{ShowDate:yyyy-MM-dd};  string Time = string.Format ($"{StartTime.Hours:D2}:{StartTime.Minutes:D2}:  {timeSpan.Seconds:D2}");  • SelectScheduleByDateTimeAndMovieID(1,Date,Time);  3. DateTime ShowDate = new DateTime(2018,7,20,0,0,0);  TimeSpan StartTime = new TimeSpan(8,0,0);  string Date = ShowDate.ToString($“{ShowDate:yyyy-MM-dd};  string Time = string.Format ($"{StartTime.Hours:D2}:{StartTime.Minutes:D2}:  {StartTime.Seconds:D2}");  • SelectScheduleByDateTimeAndMovieID(0,Date,Time);  4. SelectScheduleByDateTimeAndMovieID(1,””,””); |
| **Đầu ra kiểm thử** | 1. Trả về một Schedule có MovieID = 1 , ShowDate = ‘2018-07-20’, start\_time = ‘08:00:00’. (NOT NULL) 2. Trả về NULL không tìm thấy Schedule. 3. Trả về NULL , MovieID không được bằng 0 4. Trả về NULL , ShowDate và Start\_Time không được để NULL. |
| **Các bước kiểm thử** | 1. Nhập MovideID 2. Nhập ShowDate 3. Nhập StartTime |

# 

|  |  |
| --- | --- |
| **ID** | Test\_Case 4.6 |
| **Tên kiểm thử** | TestAddMapSeatsForSchedule |
| **Mô tả kiểm thử** | Kiểm thử phương thức AddMapSeats () trong Class Schedule DAL |
| **Tiền điều kiện** | Có dữ liệu trong cơ sở dữ liệu , đã khai báo phương thức AddMapSeats () trong Class Schedule DAL |
| **Đầu vào kiểm thử** | 1. Schedules schedule = new Schedules (1, 1, 1, new DateTime (2018, 7, 20), new TimeSpan (21, 0, 0), new TimeSpan (21, 0, 0), "MapSeat", 45000);   string mapSeats = "A B C D F E G H J K L M;10;"  • AddMapSeats(schedule,mapSeats);  2. • AddMapSeats(NULL,NULL);  3. • AddMapSeats(NULL,” ”);  4. Schedules schedule = new Schedules (0, 0, 0, new DateTime  (2018, 7, 20, 0, 0, 0), new TimeSpan (21, 0, 0), new TimeSpan  (21, 0, 0), "", 45000);  • AddMapSeats(schedule,””); |
| **Đầu ra kiểm thử** | 1. Trả về True , Insert Into thành công ghế vào cơ sở dữ liệu. 2. Trả về False, Insert Into không thành công do dữ liệu nhập vào không được NULL. 3. Trả về False, Insert Into không thành công do dữ liệu nhập vào không được NULL và MapSeats phải != “”. 4. Trả về False, Insert Into không thành công do dữ liệu Schdule.Schedule\_ID phải != 0, Schdule.Movie\_id phải != 0, Schedule.Schedule\_room\_seat phải != “”. |
| **Các bước kiểm thử** | 1. Nhập dữ liệu Schedule 2. Nhập một MapSeats. |

#### Kiểm thử Class Reservation

|  |  |
| --- | --- |
| **ID** | Test\_Case 5.1 |
| **Tên kiểm thử** | Test GetReservationByCustomerID |
| **Mô tả kiểm thử** | Kiểm thử phương thức GetReservationByCustomerID() trong Class ReservationDAL |
| **Tiền điều kiện** | Có dữ liệu trong cơ sở dữ liệu , đã khai báo phương thức GetReservationByCustomerID() trong Class ReservationDAL. |
| **Đầu vào kiểm thử** | 1. GetReservationByCustomerID(NULL). 2. GetReservationByCustomerID(0). |
| **Đầu ra kiểm thử** | 1. Kết quả trả về NULL. 2. Kết quả trả về NULL. |
| **Các bước kiểm thử** | 1. Nhập CustomerID 2. CustomerID là tham số truyền vào GetReservationByCustomerID(). 3. - Trả về NULL nếu CustomerID = NULL hoặc CustomerID = 0 hoặc CustomerID không có trong cơ sở dữ liệu.   - Trả về NOT NULL nếu tìm thầy CustomerID trong bảng Reservation. |

|  |  |
| --- | --- |
| **ID** | Test\_Case 5.2 |
| **Tên kiểm thử** | Test TestInsertIntoReservation() |
| **Mô tả kiểm thử** | Kiểm thử phương thức TestInsertIntoReservation() trong Class ReservationDAL |
| **Tiền điều kiện** | Có dữ liệu trong cơ sở dữ liệu , đã khai báo phương thức TestInsertIntoReservation() trong Class ReservationDAL. |
| **Đầu vào kiểm thử** | 1. Reservation reservation = new Reservation (1, 1, 1, "A1 A2 A3 A4 A5", 1234567,225000.0, DateTime.Now);   • TestInsertIntoReservation(reservation);  2. Reservation reservation = new Reservation (0, 0, 0,  "A1 A2 A3 A4 A5", 1234567,225000.0, DateTime.Now);  • TestInsertIntoReservation(reservation);  3. Reservation reservation = new Reservation (1, 1, 1,  NULL, 1234567,225000.0, DateTime.Now);  • TestInsertIntoReservation(reservation);  4. Reservation reservation = new Reservation (1, 1, 1,  "", 1234567,225000.0, DateTime.Now);  • TestInsertIntoReservation(reservation); |
| **Đầu ra kiểm thử** | 1. Trả về True , Khách hàng đặt vé thành công. 2. Trả về False, Đặt vé không thành công do không có CustomerID và ScheduleID. 3. Trả về False, Đặt vé không thành công do ghế = NULL. 4. Trả về False, Đặt vé không thành công do ghế = “”. |
| **Các bước kiểm thử** | 1. Nhập đầy đủ dữ liệu Reservation |

### 4.2 Kiểm thử tầng BL

|  |  |
| --- | --- |
| **ID** | Test\_CaseBL 1 |
| **Tên kiểm thử** | TestLoginCinema |
| **Mô tả kiểm thử** | Kiểm thử phương thức Login() trong Class LoginCinema (Email, Password) |
| **Tiền điều kiện** | Đã khai báo phương thức LoginCinema() trong tầng DAL. |
| **Đầu vào kiểm thử** | 1. LoginCinema(“”"", "”) 2. LoginCinema(“‘’’’@#$%^”,”...121233”); 3. LoginCinema(“Manh Dat”,”123456”); |
| **Đầu ra kiểm thử** | 1. Input đầu vào không hợp lệ, không được rỗng. 2. Input đầu vào không hợp lệ, không được chứa ký tự đặc biệt. 3. Input đầu vào hợp lệ, cho vào tầng DAL. |
| **Các bước kiểm thử** | 1. Nhập Email, Password 2. Kiểm tra định dang Matches  * Email : @"^[^<>()[\]\\,;:'\%#^\s@\$&!@]+@((\[[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\.[0-9]{1,3}\])|(([a-zA-Z0-9]+\.)+[a-zA-Z]{2,}))$" * Password : @"^[-.@\_a-zA-Z0-9 ]+$"   Không được chứa ký tự đặc biệt.  3.   * Trả về Matches nếu đúng định dạng * Trả về DoesNotMatch nếu sai định dạng |

|  |  |
| --- | --- |
| **Tên kiểm thử** | TestScheduleTimeByDateAndMovieID |
| **Mô tả kiểm thử** | Kiểm thử phương thức SelectTimeBy |
| **Tiền điều kiện** | Đã khai báo phương thức SelectTimeBy() trong tầng BL. |
| **Đầu vào kiểm thử** | 1. SelectTimeBy(1,’121212’,’1111’); 2. SelectTimeBy(1,”2018-07-26”,”08:00:00”) 3. SelectTimeBy(1,”26-7-2018”,”8:0:0”) |
| **Đầu ra kiểm thử** | 1. Đầu vào không hợp lệ, không đúng định dạng ngày 2. Đầu vào hợp lệ, chuyển sang tầng DAL 3. Đầu vào không hợp lệ, không đúng định dạng giờ. |
| **Các bước kiểm thử** | 1. Nhập Movie ID, Nhập ngày, Nhập giờ 2. Kiểm tra định dạng   Date = @"(?<year>\d{2,4})-(?<month>\d{1,2})-(?<day>\d{1,2})";  Time = @"^(\d{1,2}|\d\.\d{2}):([0-5]\d):([0-5]\d)(\.\d+)?$"  3.  Trả về Matches nếu đúng định dạng, Cho sang tầng DAL  Trả về DoesNotMatch nếu sai định dạng. |

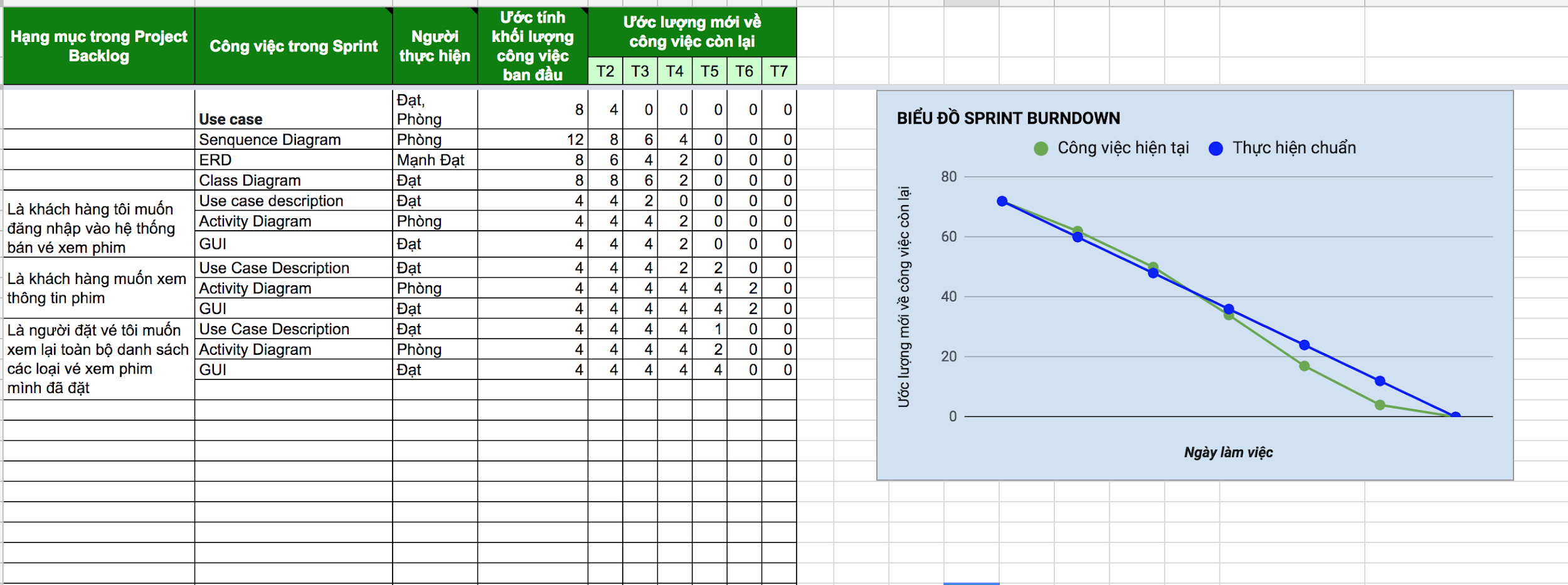
# 

# Phân công công việc cho từng thành viên trong nhóm

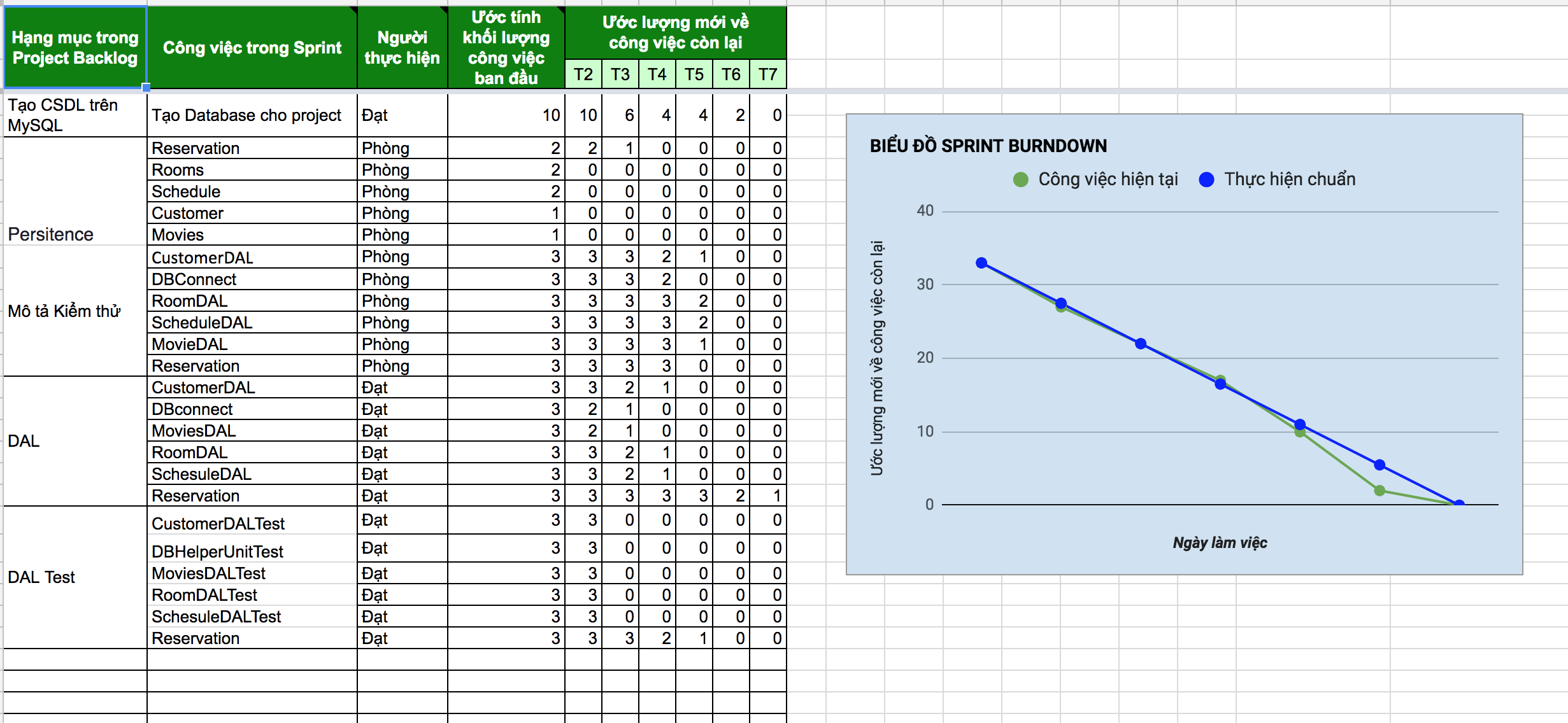
Product Backlog

# 

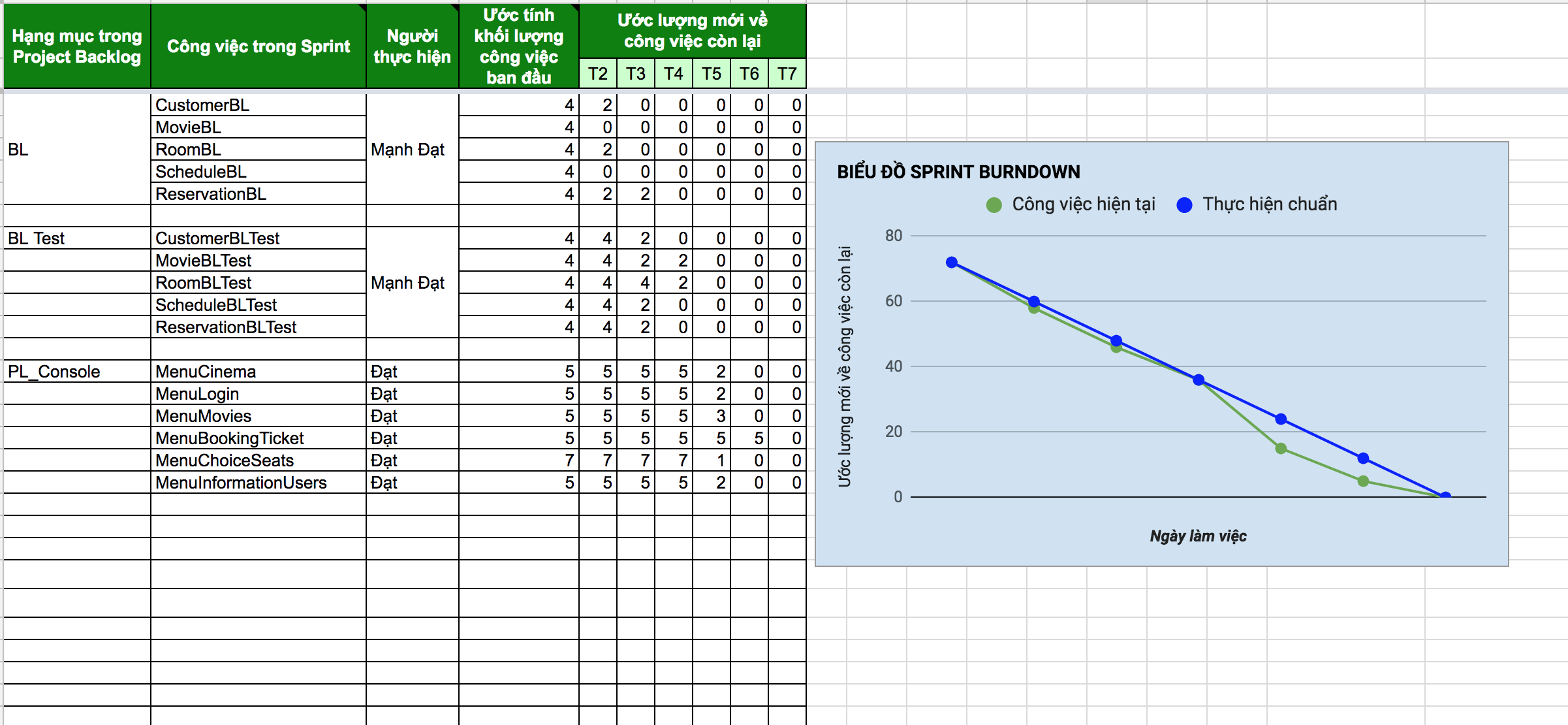
Sprint 1:



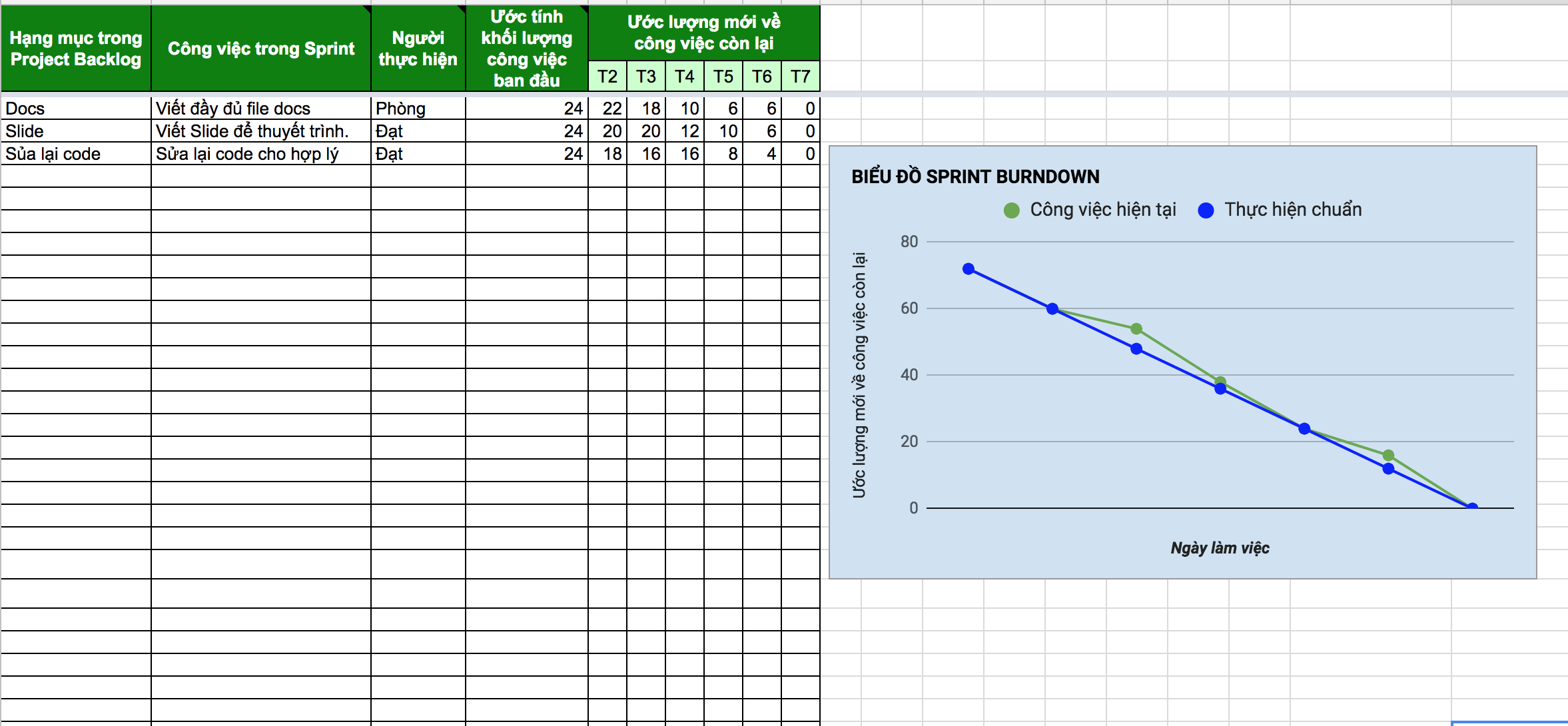
Sprint 2:

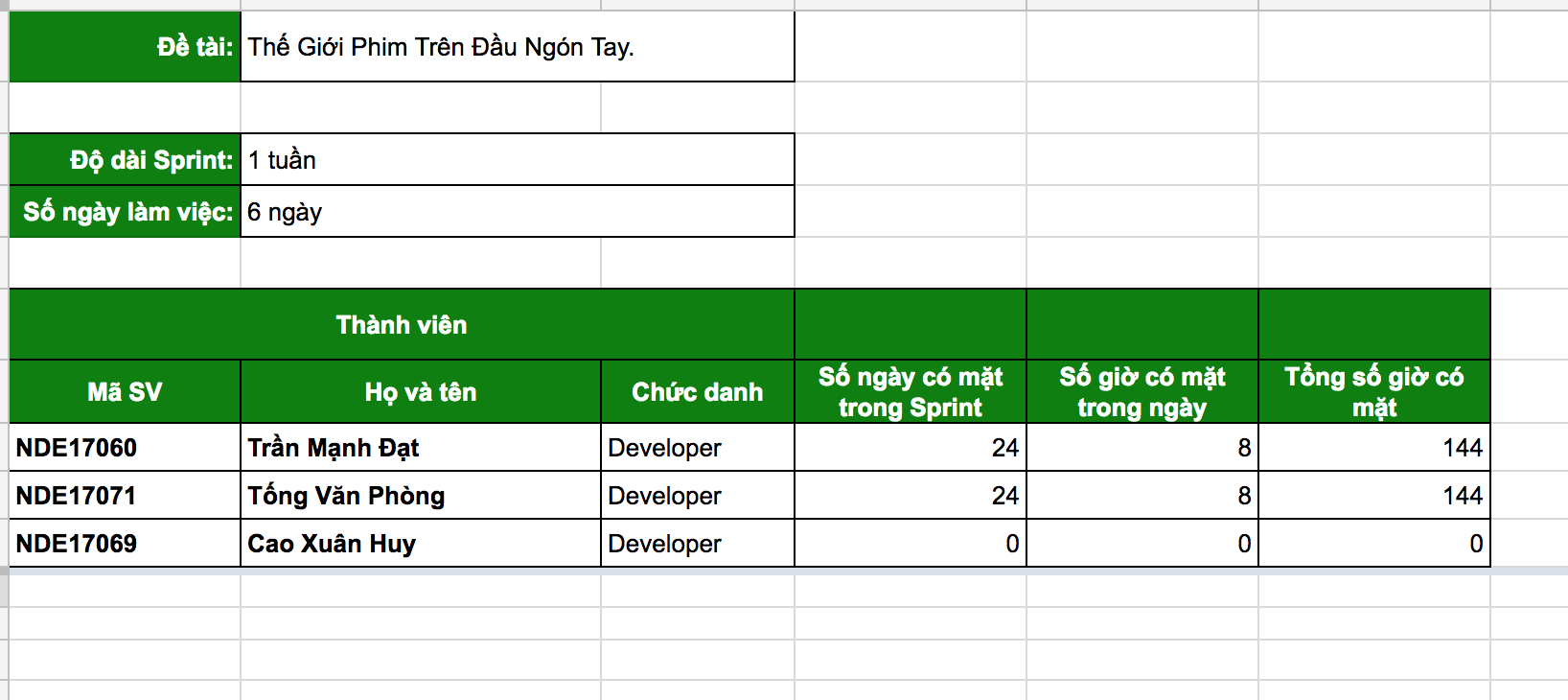


Sprint 3:



Sprint 4:

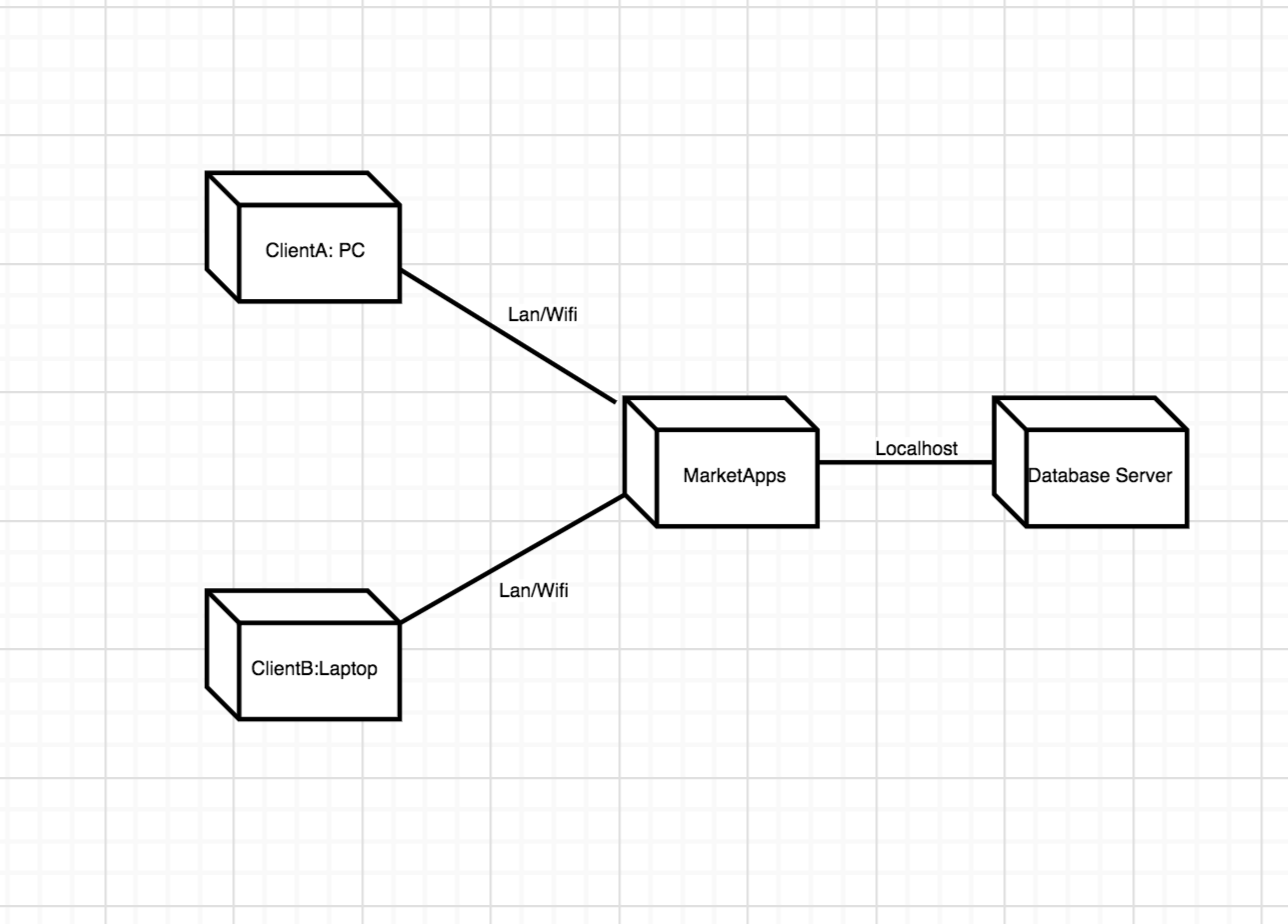




|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nhóm dự án:**  **01** | **Tên dự án: Dự án mua vé xem phim online** | | | | | |
| **STT** | **Tên công việc** | **Mô tả** | **Ngày bắt đầu** | **Ngày kết thúc** | **Thành viên liên quan** | **Tự đánh giá kết quả** |
| 1 | Sprint 1 | 1.Use case diagram  2. Activity diagram  3.Sequence diagram  4.Thiết kế database | 7/6 | 14/6 | Mạnh Đạt , Phòng | Chưa hoàn thành hết được yêu cầu đề ra từ đầu. |
| 2 | Sprint 2 | 1. Hoàn thành những công việc (chưa xong) trong Sprint 1  2. Chuyển hết code lên GitHub và đưa thông tin vào sheet 'Project Info'  3. Tạo CSDL trên MySQL Workbenck  4. Hoàn thành Code của tầng Persistence và DAL | 3/7 | 10/7 | Mạnh Đạt , Phòng | Hoàn thành tốt yêu cầu được đưa ra từ đầu |
| 3 | Sprint 3 | 1.Hoàn thành code của tầng BL 2.Hoàn thành code của tầng BL\_Test 3.Hoàn thành code của tầng PL Console | 11/7 | 18/7 | Đạt | Hoàn thành tốt yêu cầu |
| 4 | Sprint 4 | 1.Viết đầy đủ files Docs 2.Viết siles để thuyết trình 3.Sửa lại code, tối ưu code cho hợp lý | 7/7 | 23/7 | Đạt Phòng | Hoàn thành các yêu cầu. |
| 5 |  |  |  |  |  |  |

# Hướng dẫn cài đặt

#### Biểu đồ triển khai (Deployment Diagram)



#### Các bước cài đặt

* Bước 1 :

Truy cập đường link để vào folder chứa dự án. <https://github.com/ManhDatVTC/Booking-Movie-Tickets-Online>

Clone hoặc download dự án về máy.

* Bước 2 : Cài đặt Mysql Server

Bạn cần cài đặt Mysql Server để chạy dự án.

* Với Windows

Truy cập đường dẫn : <https://www.mysql.com/downloads/> -> Community

-> Mysql Community Server

-> Chọn phiên bản Windows của bạn để download (Windows (x86, 32-bit), MSI Installer)

-> Chọn File và download ứng dụng về máy

-> Tiến hành cài đặt

-> Thực hiện theo các hướng dẫn trên màn hình để cài đặt MySQL.

Đến khi hệ thống hiển thị Accounts and roles bạn nhập Password, Cần phải nhớ password để sử dụng trong toàn bộ quá trình chạy dự án.

-> Bạn đã cài đặt xong MySQL cho Windows của bạn.

-> Mở MySQL Workbench.

* Hướng dẫn cài đặt MySQL trên hệ điều hành macOs

Tải MySQL “DMG Archive” tại đường dẫn <https://dev.mysql.com/downloads/mysql/>

-> Chọn tab "**General Available (GA) Releases** "

-> Chọn hệ điều hành macOS

-> Chọn "**DMG Archive**" thích hợp cho phiên bản Mac OS cụ thể của bạn, nếu có nhiều phiên bản. Nếu không, bạn không còn lựa chọn nào khác!

- Để kiểm tra phiên bản hệ điều hành của bạn: Nhấp vào biểu tượng '**Apple**' -> **"About this Mac**".

- Để kiểm tra xem hệ điều hành Mac của bạn có phải là 32-bit hay 64-bit: Đọc tại <https://support.apple.com/en-us/HT201948> . Trừ khi bạn có một máy Mac đã quá cũ, nếu không, nên sử dụng phiên bản 64-bit.

-> Chọn Download. Tại trang tải xuống, để không cần đăng nhập, bạn chọn No thanks, just start my download

-> Bạn Download xong

-> Trong mục **Downloads**, nhấn đúp vào file ".dmg" đã tải xuống.

-> Nhấn đúp vào file "**mysql-5.7.{xx}-osx10.x-xxx.pkg**"

- Thực hiện theo các hướng dẫn trên màn hình để cài đặt MySQL. Trong quá trình cài đặt, superuser root được tạo ra với mật khẩu ngẫu nhiên tạm thời. HÃY LƯU MẬT KHẨU NÀY LẠI!!!. Đối với MySQL mới nhất, mật khẩu cũng được gửi đến các thông báo.

- MySQL sẽ được càu vào "**/usr/local/mysql**". Hãy nhớ đường dẫn cài đặt này!!

- Eject ".dmg" file.

-> Bạn đã cài đặt xong Mysql cho macOS của bạn.

-> Mở MySQL Workbench..

- Bước 3 : Chạy ứng dụng

-> Vào đường dẫn chứa folder bạn vừa tải về ở bước 1

-> Chọn MovieEngagement.sql và chạy toàn bộ lệnh để lưu cơ sở dữ liệu vào máy tính của bạn trên MySQL Workbench.

->

Truy cập đường dẫn

PL\_Console -> Bin -> DebugDebug -> netcoreapp2.0 -> osx.10.12-x64 -> PL\_Console Exec

chạy PL\_Console.exe và sử dụng hệ thống rạp chiếu phim thế giới.

# Phụ lục

***Các thuật ngữ và từ viết tắt***

*- Không*

***Tài liệu tham khảo :***

* [***https://msdn.microsoft.com/en-us***](https://msdn.microsoft.com/en-us)
* [***https://learn.vtc.edu.vn/***](https://learn.vtc.edu.vn/)
* [***https://www.w3schools.com/sql/***](https://www.w3schools.com/sql/)
* [***https://stackoverflow.com/***](https://stackoverflow.com/)
* [***https://lottecinemavn.com/en-us/default.aspx***](https://lottecinemavn.com/en-us/default.aspx)

### Tổng Kết

#### Kết quả thu được.

* **Học được trách nhiệm khi làm việc nhóm**
* **Học được quy trình làm việc của dự án thật.**
* **Hiểu kĩ và sâu hơn về C# (C Sharp) .**
* **Thiết kế cơ sở dữ liệu chính xác, phù hợp , tối ưu , tiết kiệm dữ liệu.**
* **Vẽ sơ đồ Activity Diagram, Sequence Diagram , Class Diagram**
* **Viết Test Case theo quy trình tăng hiệu suất công việc, giảm thiểu lỗi.**
* **Kỹ năng viết báo cáo, làm Slide.**
* **Tăng khả năng tự học của bản thân.**

#### Hạn chế .

* ***Lần đầu làm việc nhóm ở dự án lớn lên các thành viên trong nhóm chưa có liên kết nhiều với nhau.***
* ***Chưa giải quyết được toàn bộ yêu cầu lúc đầu đặt ra.***
* ***Chưa hiểu hết về nghiệp vụ yêu cầu bài toán, không lường trước được những vấn đề phát sinh khi viết code và phân tích cơ sở dữ liệu .***

# 23 Định dạng tài liệu

***Bìa báo cáo:***

*Bìa báo cáo được in màu xanh.*

*Với định dạng như trang đầu tiên của tài liệu này.*

***Khổ giấy***

*Báo cáo được trình bày trên khổ giấy khổ A4 (210 mm x 297 mm)*

***Lề trên***

*Bên trái: Logo của VTC Academy*

*Bên phải: Tên Project*

*Font chữ: Time New Roman*

*Font size: 13pt*

***Lề dưới***

*Bên trái: Tên lớp – Tên nhóm*

*Bên phải: Số trang*

*Font chữ: Time New Roman*

*Font size: 13pt*

***Nội dung tài liệu:***

*Báo cáo được trình bày theo chiều dài của trang giấy khổ A4 (định hướng bản in theo chiều dài)*

*Font chữ: Time New Roman*

*Font size: 13pt*

*Tối thiểu 20 trang*

***Định lề trang văn bản (đối với khổ giấy A4)***

*Lề trên: cách mép trên từ 20 - 25 mm;*

*Lề dưới: cách mép dưới từ 20 - 25 mm;*

*Lề trái: cách mép trái từ 30 - 35 mm;*

*Lề phải: cách mép phải từ 15 - 20 mm*