**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



**BÁO CÁO ĐỒ ÁN**

**LẬP TRÌNH TRÊN WINDOWS**

**QUẢN LÝ RẠP PHIM**

**SVTH: Võ Minh Hiếu**

**MSSV: 17110136**

**SVTH: Lê Minh Tiến**

**MSSV: 17110236**

**SVTH: Vũ Hoàng Hiệp**

**MSSV: 17110138**

**GVHD: Lê Thanh Vinh**

**TP. Hồ Chí Minh, tháng 5 năm 2018**

**LỜI NÓI ĐẦU**

Lời nói đầu tiên, nhóm xin cảm ơn quý thầy cô đã đọc bài báo cáo và chấm điểm đồ án cho nhóm. Nói sơ lược về nội dung báo cáo, bài báo cáo là tổng hợp những nội dung quan trọng mà nhóm sử dụng để làm đồ án. Nội dung được dựa trên những gì đã học của môn “Lập trình trên Windows”. Như tiêu đề ở trang bìa, đề tài nghiên cứu của nhóm là “Quản lý rạp phim”. Đây là lần đầu nhóm thực hiện một đồ án như vậy, nên không thể tránh khỏi những sai sót. Kính mong quý thầy cô và bạn đọc thông cảm bỏ qua. Chân thành cảm ơn.

MỤC LỤC

[I. Lý do chọn đề tài 5](#_Toc8929945)

[II. Giới thiệu phần mềm 5](#_Toc8929946)

[III. Nội dung 5](#_Toc8929947)

[2. Software architecture 5](#_Toc8929948)

[a) DBMain 5](#_Toc8929949)

[b) BSLayer 8](#_Toc8929950)

[c) renderSeat 13](#_Toc8929951)

[3. Giao diện người dùng 16](#_Toc8929952)

[a) Sử dụng User Controls. 16](#_Toc8929953)

[b) QuanLyNhanVien.cs 17](#_Toc8929954)

[c) QuanLyKhachHang.cs 20](#_Toc8929955)

[d) QuanLyPhim.cs 22](#_Toc8929956)

[e) PhimUC.cs 24](#_Toc8929957)

[4. Database Diagram 24](#_Toc8929958)

[f) Bảng NhanVien 25](#_Toc8929959)

[g) Bảng DatVe 25](#_Toc8929960)

[h) Bảng KhachHang 26](#_Toc8929961)

[i) Bảng Rap 27](#_Toc8929962)

[j) Bảng Phim 27](#_Toc8929963)

[IV. Kết luận 28](#_Toc8929964)

[1. Ưu điểm: 28](#_Toc8929965)

[2. Nhược điểm: 28](#_Toc8929966)

DANH MỤC CÁC HÌNH

[Figure 1. Database Diagram 6](#_Toc8754134)

[Figure 2. Design NhanVien 6](#_Toc8754135)

[Figure 3. Data NhanVien 6](#_Toc8754136)

[Figure 4. Design DatVe 7](#_Toc8754137)

[Figure 5. Data DatVe 7](#_Toc8754138)

[Figure 6. Design KhachHang 7](#_Toc8754139)

[Figure 7. Data KhachHang 8](#_Toc8754140)

[Figure 8. Design Rap 8](#_Toc8754141)

[Figure 9. Data Rap 8](#_Toc8754142)

[Figure 10. Design của Phim 8](#_Toc8754143)

[Figure 11. Code BLDatVe 9](#_Toc8754144)

[Figure 12. Code BLKhachHang 10](#_Toc8754145)

[Figure 13. Code BLNhienVien 11](#_Toc8754146)

[Figure 14. Code BLPhim 12](#_Toc8754147)

[Figure 15. Code BLRap 13](#_Toc8754148)

[Figure 16. Code BDMain 14](#_Toc8754149)

## Lý do chọn đề tài

Là sinh viên, chúng ta có khá nhiều thời gian để giải trí sau những giờ học tập căng thẳng. Có rất nhiều các để giải trí, nhưng trong số vô vàng lựa chọn, sinh viên của chúng ta thường lựa chọn đó là đi xem phim ở các rạp phim. Lý do có vẻ đơn giản, nhưng đó chính là nguồn động lực để thúc đẩy nhóm chọn đề tài liên quan đến lĩnh vực này – “Phần mềm quản lý rạp phim”

## Giới thiệu phần mềm

Phần mềm là 1 chuỗi các hệ thống quản lý bao gồm: Quản lý nhân viên, quản lý các khách hàng, quản lý lịch phát các phim trong rạp. Các tính năng quản lý cao cho người dùng. Các tính năng này sẽ được nói rõ hơn ở phần nội dung.

Phần mềm dùng ADO.NET để kết nối với Database và thao tác trên nó.

## Nội dung

1. ADO.NET là gì?

ADO.NET là một bộ các thư viện hướng đối tượng (OOP) cho phép bạn tương tác với dữ liệu nguồn. Thông thường thì dữ liệu nguồn là một cơ sở dữ liệu (database), nhưng nó cũng có thể là file text, exel hoặc XML. Theo những mục tiêu của hướng dẫn này, chúng ta sẽ chỉ xem xét tới cách ADO.NET làm việc với database.

Như bạn có thể biết rằng, có rất nhiều loại database hiện nay như Microsoft SQL Server, Microsoft Access, Oracle, Borland Interbase, và IBM DB2,… Để làm rõ hơn phạm vi của loạt bài này, tất cả ví dụ sẽ sử dụng SQL Server.

### Software architecture

Code chính của chương trình được chia làm 3 phần với BSLayer (Các query tương tác với database), DBLayer (Thiết lập các lênh truy vấn), renderSeat (tạo các ghế của rạp).

Đa số code được thực hiện qua DataSet. Một DataSet tương tự như một tập tin database vật lý hoàn chỉnh nhưng được lưu trong bộ nhớ. DataSet bao gồm các DataTable, DataTable bao gồm các DataColumn, DataRow, các constraint.

#### DBMain

Là tầng kết nối với database. Là trung gian để mở các kết nối dẫn các câu truy vấn từ client tới database. Được thực hiện qua DataSet. Với ExecuteNonQuery: kết quả trả về là số dòng bị ảnh hưởng.

Khởi tạo các đối tượng ban đầu để kết nối với database:

* SqlConnection

Để tương tác với database, bạn phải có một kết nối tới nó. Kết nối giúp xác định database server, database name, user name, password, và các tham số cần thiết để kết nối tới database. Một đối tượng connection được dùng bởi đối tượng command vì thế chúng sẽ biết database nào để thực thi lệnh.

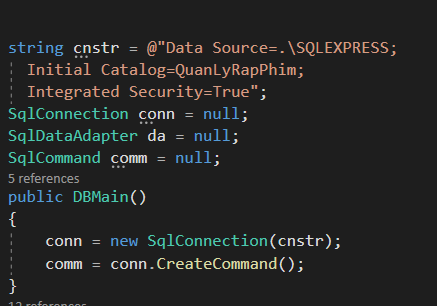
* SqlCommand

Quá trình tương tác với database cần phải biết hành động nào bạn muốn xảy ra. Điều này được thực hiện bởi đối tượng command. Bạn dùng đối tượng command để gửi một câu lệnh SQL tới database. Một đối tượng command dùng một đối tượng connection để xác định database nào sẽ được truy xuất. Bạn có thể dùng một đối tượng command riêng lẻ để thực thi lệnh trực tiếp, hoặc để gắn một tham chiếu của đối tượng command cho một SqlDataAdapter – đối tượng giữ các command sẽ làm việc trên một nhóm dữ liệu như sẽ đề cập tới trong phần dưới.

* SqlDataAdapter

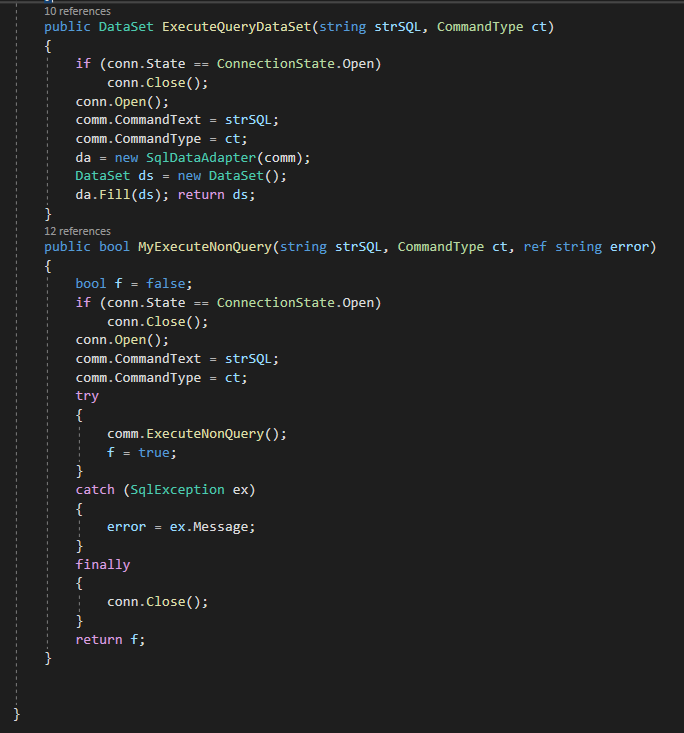
Đôi lúc dữ liệu mà bạn làm việc là read-only và bạn ít khi cần thay đổi dữ liệu nguồn. Vài trường hợp cần lưu trữ tạm dữ liệu trong bộ nhớ để hạn chế truy xuất đến database. Data adapter làm điều này dễ dàng bằng cách giúp bạn quản lý dữ liệu trong chế độ ngắt kết nối. Data adapter sẽ đổ vào DataSet khi đọc dữ liệu và thực hiện thay đổi dữ liệu một lượt vào database.

Data adapter chứa một tham chiếu đến đối tượng connection và mở/đóng kết nối tự động khi đọc và ghi dữ liệu vào database. Hơn nữa, data adapter chứa đối tượng command cho những thao tác SELECT, INSERT, UPDATE và DELETE trên dữ liệu. Bạn sẽ có một data adapter được định nghĩa cho mỗi table trong một DataSet và nó sẽ quản lý các giao tiếp với database cho bạn. Tất cả những gì bạn cần làm là chỉ cho data adapter khi nào nạp hoặc ghi vào database.[1]



* DataSet

Đối tượng DataSet là một thể hiện của dữ liệu trong bộ nhớ. Chúng chứa nhiều đối tượng DataTable, bên trong DataTable lại có nhiều column và row, giống như các database table thông thường. Bạn thậm chí có thể định nghĩa dữ liệu giữa các table để tạo các quan hệ parent-child. DataSet được thiết kế đặc biệt để giúp quản lý dữ liệu trong bộ nhớ và để hỗ trợ các thao tác không cần kết nối (disconnected) trên dữ liệu. DataSet là một đối tượng được dùng bởi tất cả Data Provider, đó là lý do tại sao nó không có một Data Provider prefix trong tên gọi.



##### Figure . Code BDMain

* ExecuteQueryDataSet: Đổ dữ liệu từ database thông qua câu truy vấn strSQL ( câu truy vấn SELECT Object)
* MyExcuteNonQuery: Thực thi thao tác là thay đổi database (INSERT, UPDATE và DELETE của Object). Nếu thông tin chính xác và đầy đủ thì trả về true và tiến hành thay đổi database, nếu thông tin không chính xác thì trả về false và hiện thông báo lỗi. Sau khi thực hiện hết các bước trên ta sẽ đóng database lại.

#### BSLayer

Ở tầng BSLeyer sẽ là các query truy vấn can thiệp vào thêm, lưu sửa xoá các dữ liệu có trong database. Mỗi một bảng của database sẽ là 1 class trong BSLayer: BLDatVe.cs, BLKhachHang.cs, BLNhanVien.cs, BLPhim.cs, BLRap.cs

Đa số các code của mỗi class đều giống nhau. Nên chúng ta sẽ giải thích chung cho tất cả các Class. Ta gọi Obj thay cho các đối tượng cụ thể của các Class

+ DataSet LayThongTinObj(): Dùng câu truy vấn “Select \* from Obj” và ExecuteQueryDataSet đã được định nghĩa ở DBMain để lấy tất cả thông tin của Obj trong database.

+ DataSet findObj(): Dùng câu truy vấn “$"select column from DatVe where Col= +

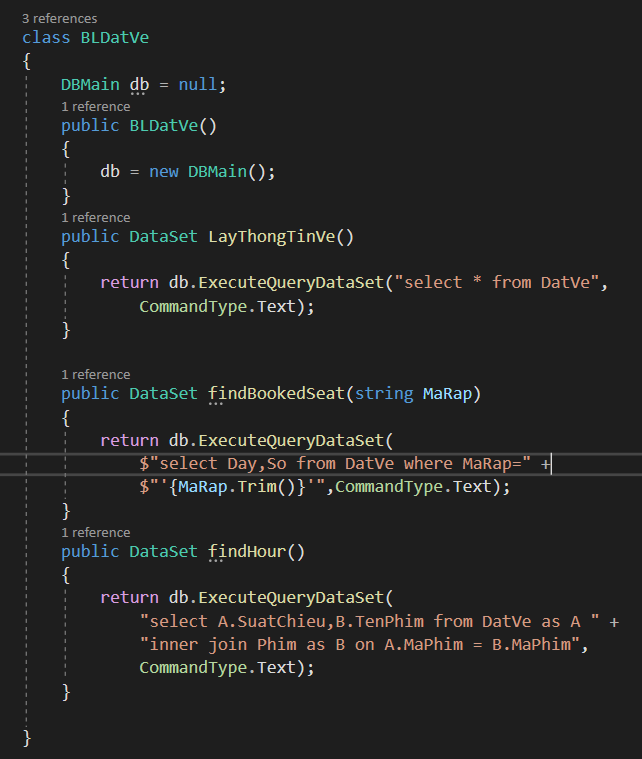
$"'{Row’s value}'"” và ExecuteQueryDataSet đã được định nghĩa ở DBMain để lấy thông tin cụ thể cần tìm của Obj trong database. Với “column” là tên các cột có trong database, “Col” là các hàng ứng với Row’s value, “Row’s value” giá trị của hàng đó ứng với Col.

+ Bool ThemObj(): Dùng câu lệnh “ Insert Into Obj Values + ‘infor’ “ và MyExcuteNonQuery đã được định nghĩa ở DBMain để thêm thông tin vào Database. Với infor là tất cả các thông tin mới ứng với dòng cột cần thêm vào database.

+ Bool XoaObj(): Dùng câu lệnh " Delete From Obj Where ‘Col = Row’s Value’ “. Với “Col” là các hàng ứng với Row’s value, “Row’s value” giá trị của hàng đó ứng với Col. Để xoá các thông tin cụ thể mà chúng ta cần xoá trong database.

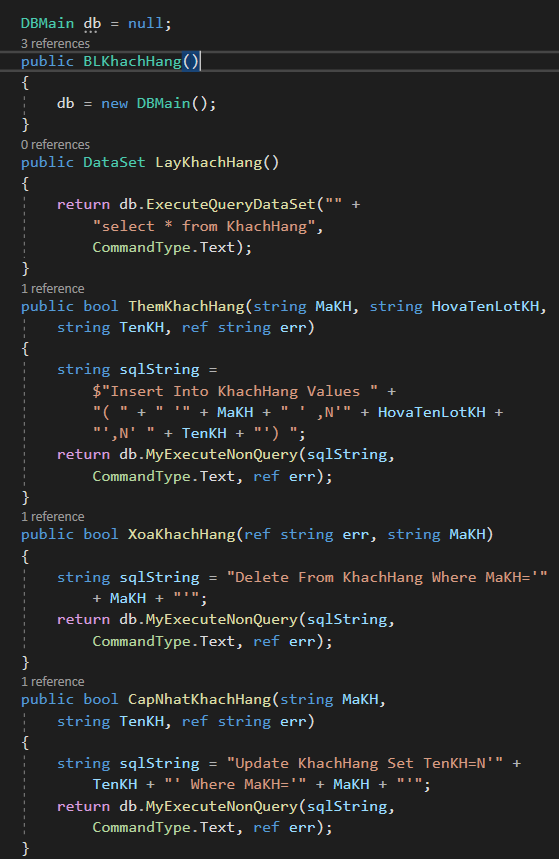
+ Bool CapNhatObj: Dùng câu lệnh “ Update Obj Set [Thông tin cần thay đổi] where ‘Col = Row’s Value’ “. “. Với “Col” là các hàng ứng với Row’s value, “Row’s value” giá trị của hàng đó ứng với Col. Để cập nhật lại các thông tin cụ thể mà chúng ta muốn thay đổi.

* Lưu ý: các cậu truy vấn trên là mã giả để người xem dể hiểu. Code thật sẽ được trích dẫn theo từng Class.
* Class BLDatVe: các câu truy vấn bao gồm LayThongTinVe(); findBookSeat()\_Tìm ghế book; findHour()\_tìm suất chiếu;



##### Figure . Code BLDatVe

* Class BLKhachHang: chứa các câu truy vấn bao giồm LayKhachHang()\_Lấy thông tin khách hàng; ThemKhachHang()\_Thêm thông tin khách hàng mới vào database; XoaKhachHang()\_Xoá khách hàng đã có trong database; CapNhatKhachHang()\_Thay đổi thông tin đã có trong database.



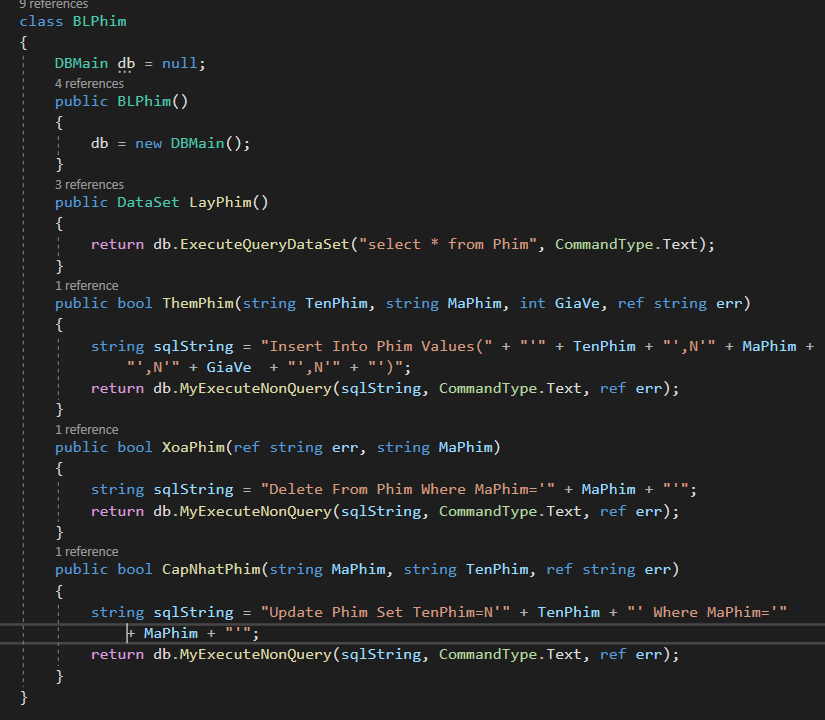
##### Figure . Code BLKhachHang

* Class BLNhanVien: chứa các câu truy vấn bao giồm LayNhanVien()\_Lấy thông tin nhân viên; ThemNhanVien()\_Thêm thông tin nhân viên mới vào database; XoaNhanVien()\_Xoá nhân viên đã có trong database; CapNhatNhanVien()\_Thay đổi thông tin đã có trong database. TimKiemNhanVien()\_Tìm kiếm nhân viên trong database bằng tên của nhân viên đó.



##### Figure . Code BLNhienVien

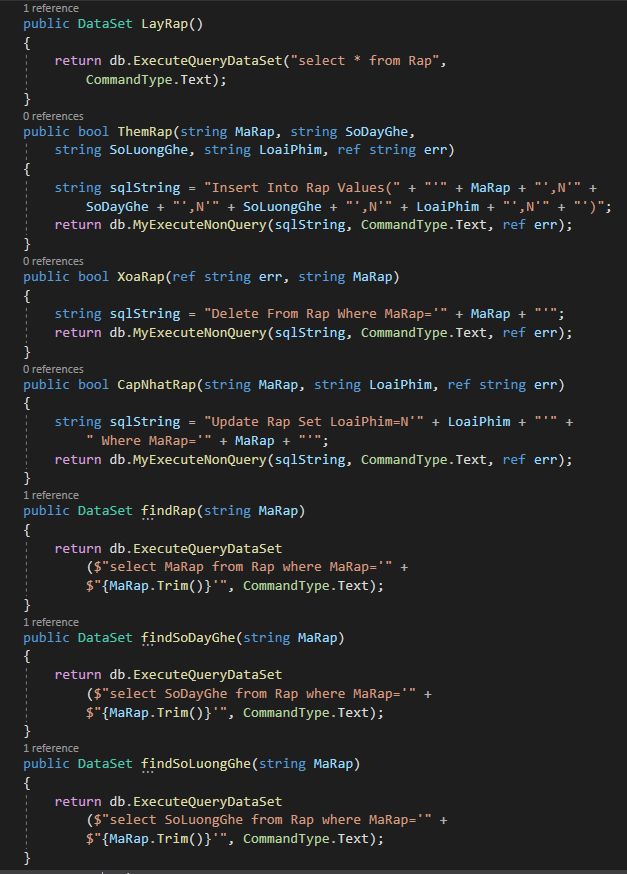
* Class BLPhim: chứa các câu truy vấn bao giồm LayPhim()\_Lấy thông tin phim; ThemPhim()\_Thêm thông tin phim mới vào database; XoaPhim()\_Xoá phim đã có trong database; CapNhatPhim()\_Thay đổi thông tin đã có trong database.



##### Figure . Code BLPhim

* Class BLRap: chứa các câu truy vấn bao giồm LayRap()\_Lấy thông tin Rap; ThemRap()\_Thêm thông tin Rap mới vào database; XoaRap()\_Xoá Rap đã có trong database; CapNhatRap()\_Thay đổi thông tin đã có trong database; findRap()\_tìm rạp trong database; findSoDayGhe()\_tìm số dãy ghế trong database; findSoLuongGhe()\_

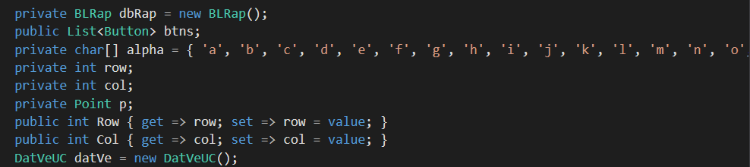
tìm số dãy ghế trong database.



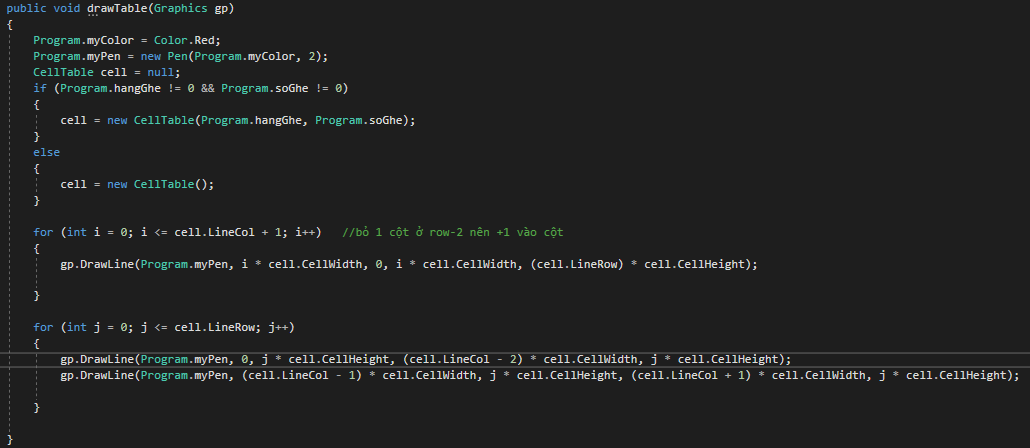
##### Figure . Code BLRap

#### renderSeat

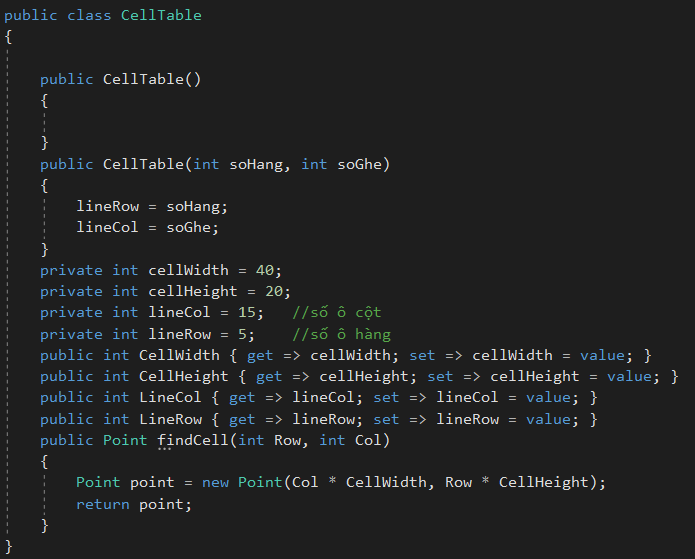
* + - * Render.cs



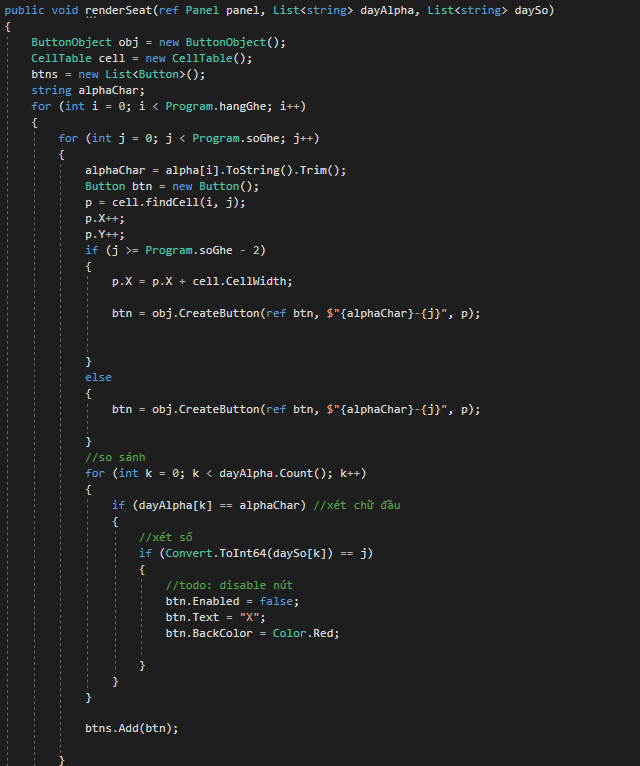
* Các biến cơ bản:
* Một list các button để chứa button cho chỗ ngồi.
* Alpha là dãy kí tự từ a đến z
* P là biến vị trí để đặt chỗ cho button.
* Hàm drawTable



* Hàm này dùng để vẽ các đường màu đỏ để định hình khung cho rạp.
* Dòng for đầu là vẽ cột. dòng for sau vẽ hàng(vẽ hàng sẽ phức tạp hơn tí).
* Ngoài ra có 2 biến static là hangGhe và soGhe: hàng ghế là số hàng ghế còn số ghế là số ghế trên 1 hàng ghế. Dựa 2 biến này để vẽ ra khung.
* Class CellTable

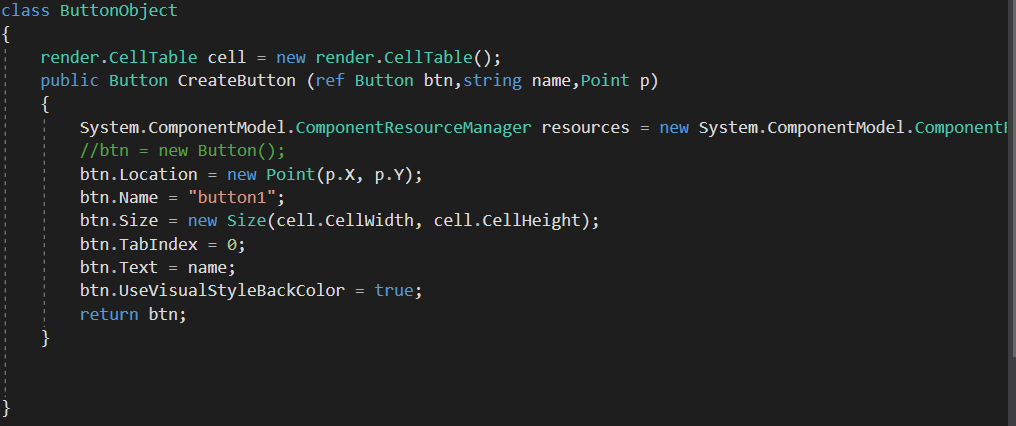


* Là class định hình cho chiều dài chiều rộng của nút và cũng như là ô của khung. Nếu không có số hàng và số ghế thì mặc định hàng sẽ là 5 và ghế sẽ là 15.
* findCell là hàm trả về vị trí của 1 ô trên bảng.
* hàm renderSeat





* Hàm này là hàm quan trọng nhất. hàm này cần truyền vào panel cần vẽ, dãy alpha là 1 list chữ lấy từ database. Và daySo cũng là số ghế đã được đặt trong database. Dùng 2 biến này để đánh dấu những ghế đã có người đặt
* Dùng 2 vòng lặp. dòng lặp đầu để render hàng và vòng lặp sau render cột tương ứng với ghế.. trong khi render dòng thì gắn chữ cái cho ghế. Vừa render vừa add vào list<button>
* Vòng lặp cuối cùng là dùng để đánh dấu nút nào vừa viết trùng với nút trên database để đánh dấu là đã có người đặt.
* Class buttonObject (createObject.cs)



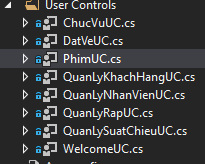
Là class dùng để tạo nút bằng code thay vì kéo thả. Làm vậy sẽ tiện để rtender hơn thay vì phải kéo thả. Chỉ cần truyền nút, tên và vị trí

### Giao diện người dùng

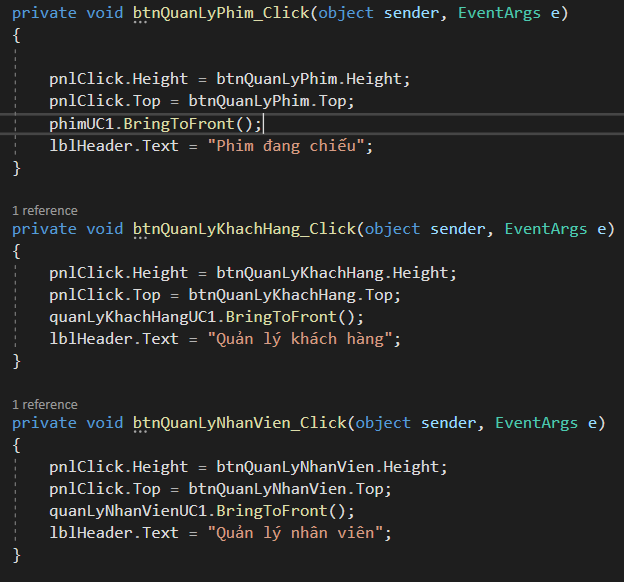
#### Sử dụng User Controls.

Thông thường khi chúng ta muốn mở một form mới, ta sẽ tạo 1 form mới xuất hiện cùng lúc với form cũ. Việc làm như vậy sẽ dẫn tới việc tiêu tốn tài nguyên của máy, logic của phần mềm cũng giảm đi. Vì vậy, để tất cả mọi thứ đều đổ vào 1 form ta sẽ sử dụng User Controls.

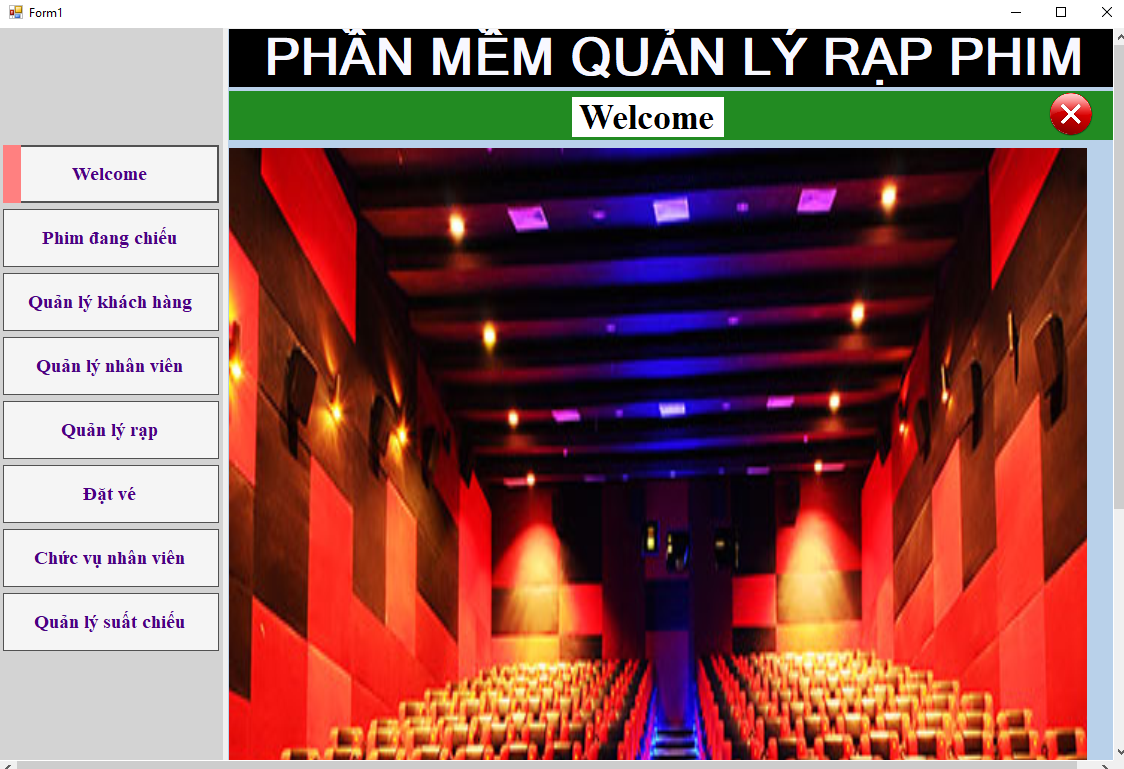
Tất các các User Controls sẽ được đặc vào thư mục User Controls.



Ở From chính (form1.cs) ta sẽ dùng lệnh BringToFront() để đẩy từng User Controls và Form chính.



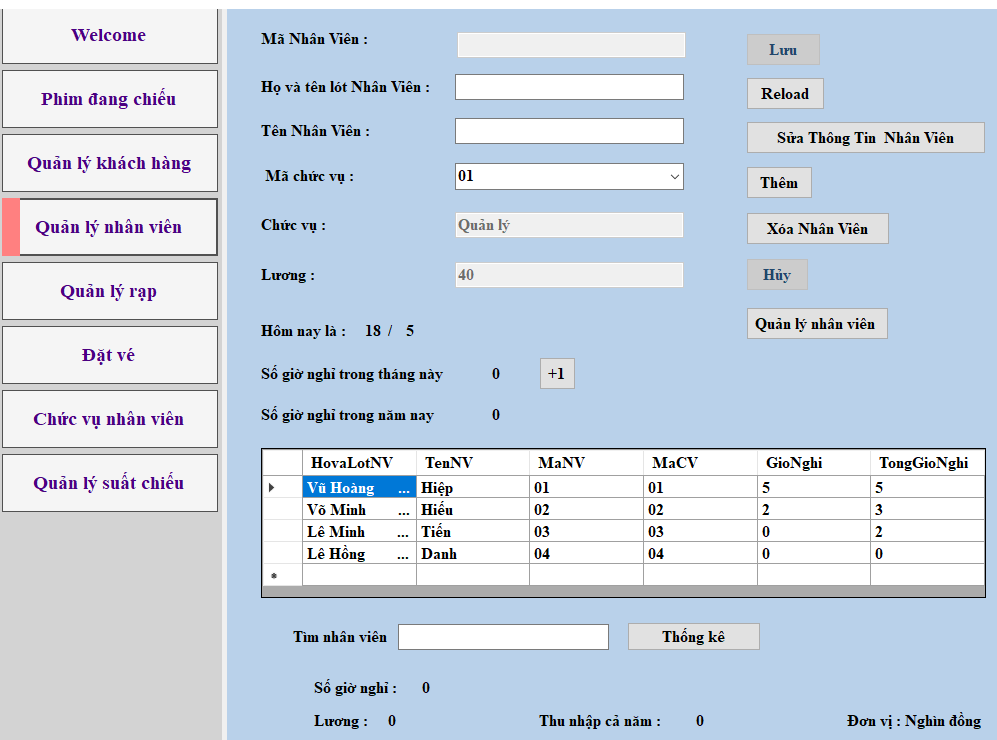
Giao diện form chính.

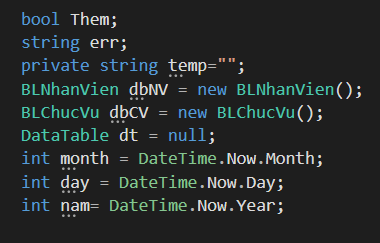


Mỗi User Controls sẽ là đại diện quản lý 1 table trong Database. Nó sẽ bao gồm các nút Thêm, Sửa, Xoá, Huỷ, Lưu, các textbox tương ứng với từng thông số trong table đó và một DataGridView chứa các thông tin của Table trong Database. Ví dụ như user controls quản lý nhân viên.

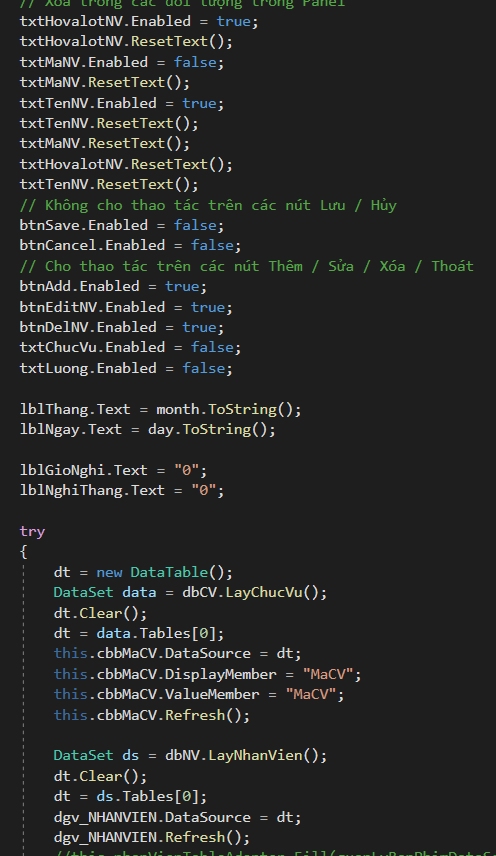
Đa số các chứa năng được đề cập ở trên giữa các User Controls là hoàn toàn giống nhau, chỉ khác mỗi giao diện. Vì vậy ta sẽ giải thích 1 User Control “quản lý nhân viên”. Source code mỗi control khác và những tính năng đặc thù của mỗi User Controls sẽ được đề cập bên dưới.

#### QuanLyNhanVien.cs

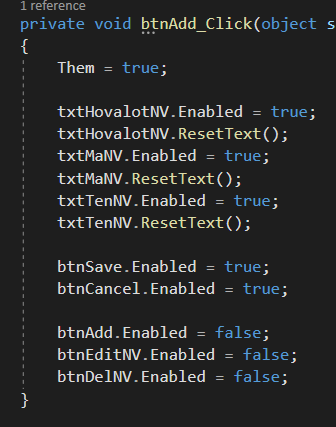




* Đầu tiên khai báo đối tượng chính của User Control (NhanVien)

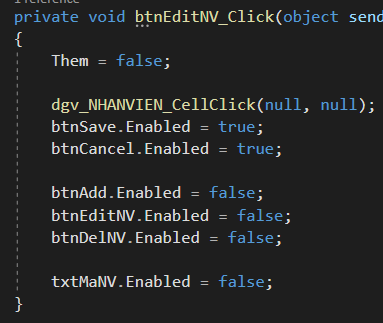


* Định nghĩa hàm LoadData():
  + - * Hàm LoadData là hàm tiền khởi đầu cho mỗi user controls. Mỗi khi sự kiên Form\_Load được bật lên. Thì hàm này sẽ được thực hiện. Trong LoadData, ta sẽ bật/khoá các chứng năng của các nút cần/không cần thiết. Ta sẽ không dùng các nút Lưu/Huỷ khi loadform. Nên ta sẽ tắt nó bằng button.Enable = false. Các nút Thêm, Sửa, Xoá, Thoát, các textbox chứa các giá trị thuộc tính của database sẽ được phép thao tác (Enable = True).
      * Ta tạo 1 DataTable để chứa thông tin từ bảng chức vụ. Sao đó lấy thông tin từ bảng chức vụ cho vào combobox.
      * Tạo mới 1 DataSet để và dùng hàm LayNhanVien để lấy các thông tin của bảng Nhân viên. Sao đó cho dữ liệu vào datagridview.
* Sự kiện click của button add: Khi click vào nút add thì sự kiện click sẽ được bật lên. Đầu tiên là bật cờ them ở chế độ true. Sau đó, ta sẽ mở khoá tất các textbox ( textbox.Enable = true) đồng thời ResetText các textbox. Tiếp đến là mở khoá chức năng Lưu và Huỷ (button.Enable = true), tắt các chức năng Thêm, Sửa, Xoá ( button.Enable = false).

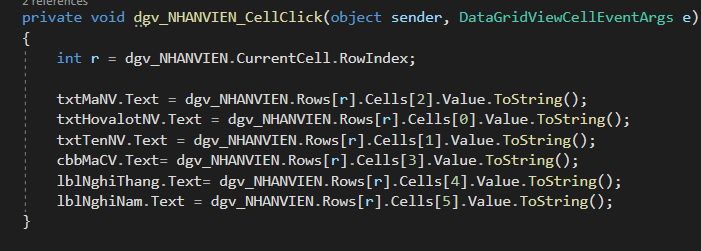


* Sự kiện click của button Edit: Khi click vào nút Sửa thì sự kiện click sẽ được bật lên. Đầu tiên là bật cờ them ở chế độ false. Sau đó, ta sẽ mở khoá các textbox liên quan (textbox.Enable = true), trừ textbox của khoá chính\_khoá chính chỉ được phép thêm, không được phép sửa ( key’s textbox = false), Tiếp đến là mở khoá chức năng Lưu và Huỷ (button.Enable = true), tắt các chức năng Thêm, Sửa, Xoá ( button.Enable = false).

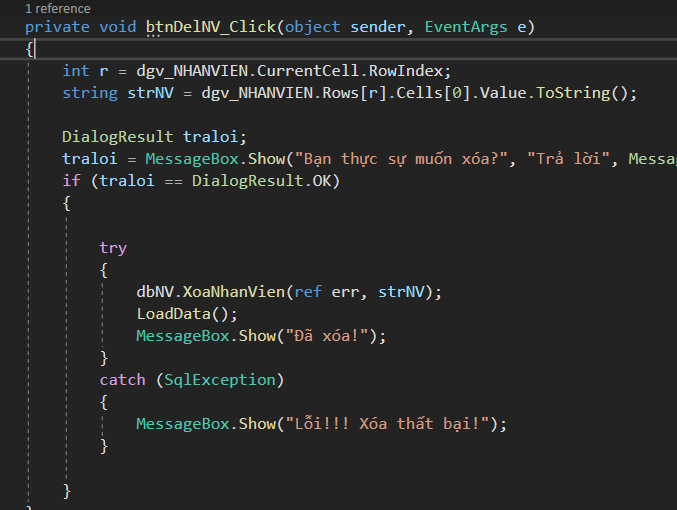
Ngoài ra ta còn có sự kiện CellClick của DataGridView. Sự kiện này giúp ta lấy thông số từ datagridview ra ngoài textbox.



* Sự kiện CellClick của DataGridView: Là sự kiện mà chúng ta sẽ lấy thông của 1 dòng khi chúng ta click chuột vô dòng đó. Và thông tin sẽ được lưu vào trong textbox phía trên. Vd: MaMV(key) sẽ được lấy từ cột 2, HovaLot sẽ được lấy từ cột 0, Ten sẽ được lấy từ cột 1, v.v.



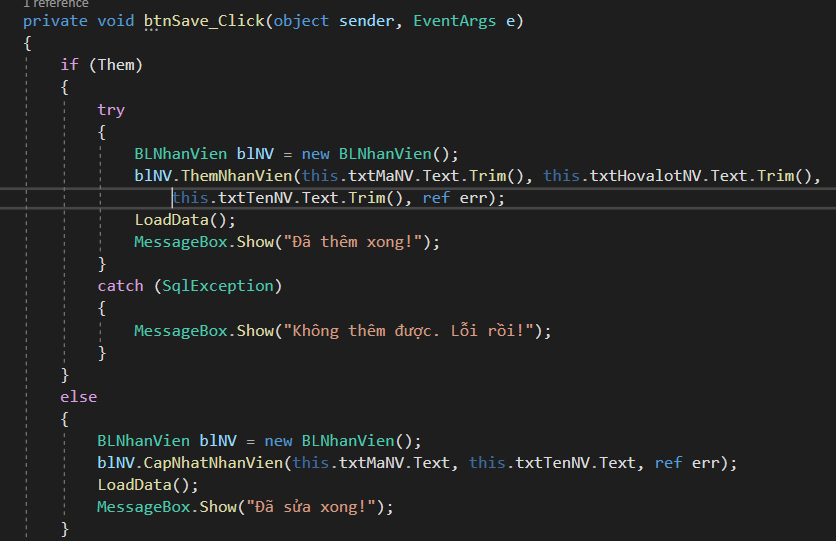
* Sự kiện click của button Xoá: ta sẽ thấy thông tin cần xoá bằng cách lấy khoá chính từ datagridview qua dòng strNV. Sau đó dùng hàm XoaNhanVien đã được định nghĩa ở tầng BSLayer để tiến hành xoá nó khỏi database. Để chương trình không bị lỗi khi đang chạy ta cùng try catch và SqlExeption để bắt lỗi.



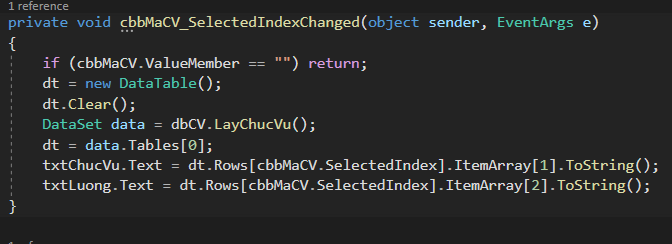
* Sự kiện click của button Lưu: Lúc này, khi lúc vào nút Lưu. Thì nó sẽ kiểm tra xem cờ them (cho phép thêm) đã được bật hay chưa?

+ Nếu đã bật: ta sẽ khởi tạo mới một BLNhanVien và dùng hàm ThemNhanVien đã định nghĩa ở tầng BSLayer để tiến hành lưu. Khi lưu xong thì sẽ xuất hiện 1 thông báo đã thêm xong, còn nếu ko lưu được SqlExeption bắt lỗi và thông báo lỗi cho người dùng.

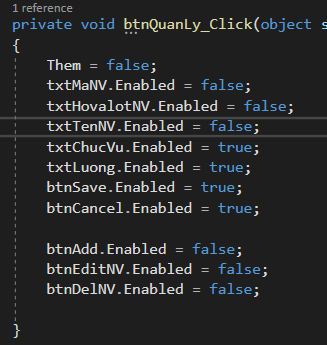
+ Nếu cờ them không bật: ta sẽ hiểu đó là cập nhật database. Thay vì dùng hàm ThemNhanVien ta sẽ dùng hàm CapNhatNhanVien. Các thao tác logic còn lại giống như khi bật cờ them.



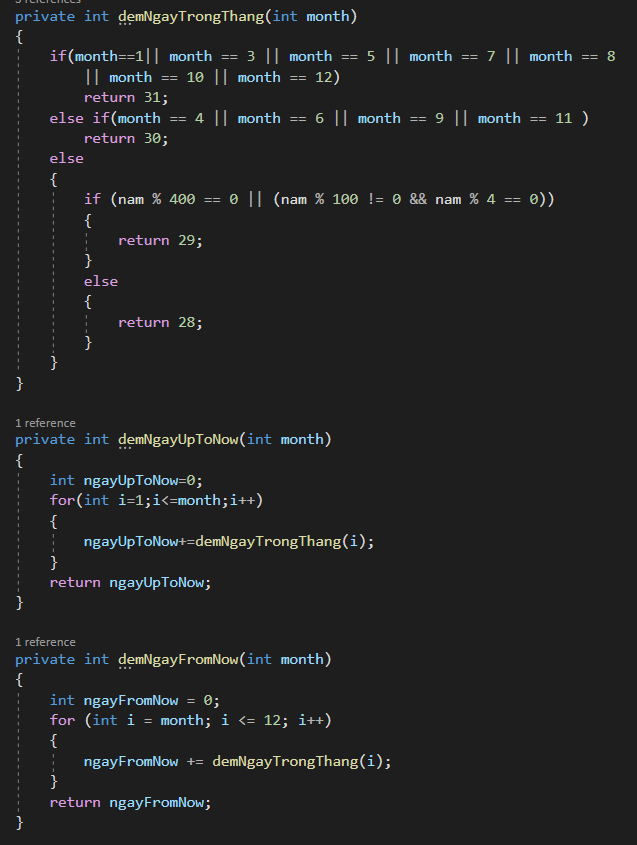
* Sự kiện SelectecdIndexChanged của combobox: dữ liệu được lấy ở bảng Chức Vụ, sao đó được đổ vào combobox và txtxchucvu, txtluong.



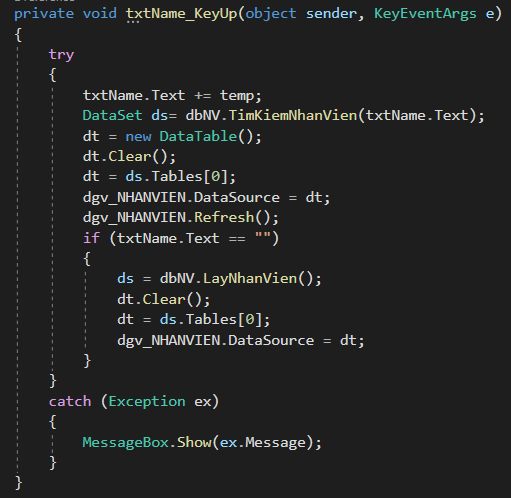
* Sự kiện click của button Quản Lý Nhân Viên: buton này cho phép thay đổi các chức vụ mà mức lương của nhân viên. Ta sẽ Enable = false tất cả, chỉ để lại combobox chức vụ và mã chức vụ mức lương.



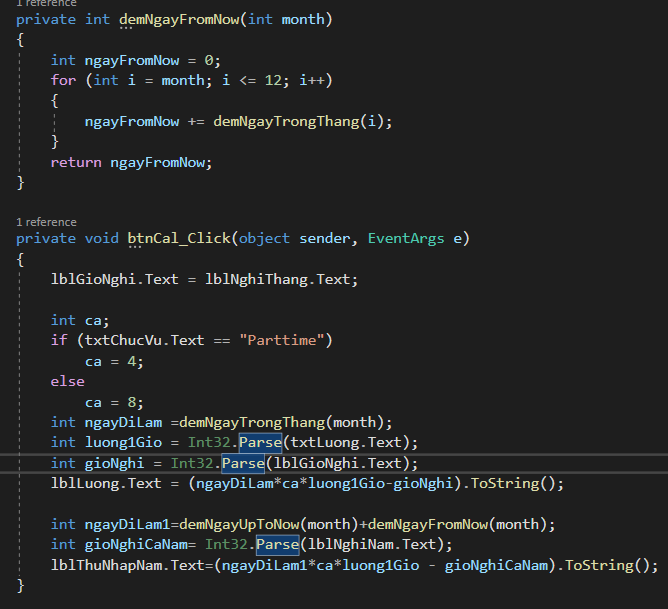
* Các hàm đếm ngày: các hàm này phụ trợ cho phần thống kê nhân viên bên dưới:



* Sự kiện KeyUp của textbox tìm kiếm(txtName): Hàm sẽ tìm kiếm tên của nhân viên là hiện lên DataGridView.



* Sự kiện click của Button thống kê: Sao khi tìm được nhân viên trong database bằng sự kiện KeyUp của textbox. Ta sẽ phân thời gian làm của nhân viên đó. Nếu parttime thì set 4 tiếng/ngày, còn lại là 8 tiếng/ ngày. Sau đó xét số giờ nghĩ, số ngày đi làm để tiến hành tính tiền lương cho nhân viên.



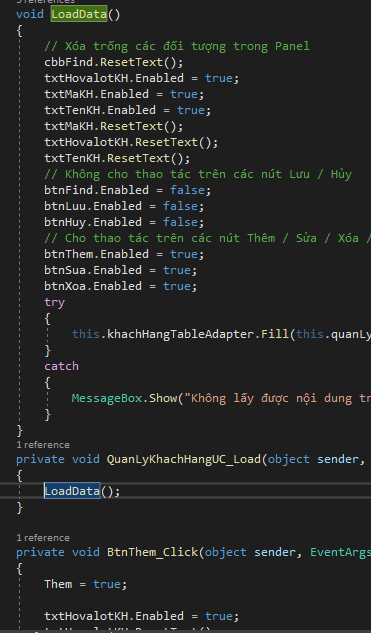
#### QuanLyKhachHang.cs

Đây là giao diện của tầng quản lý khách hàng. Ở tầng quản lý khách hàng. Ta chỉ có các nút cơ bản như Thêm, Sửa, Huỷ, Xoá và Lưu. Kết cấu thì hệt như QuanLyNhanVien

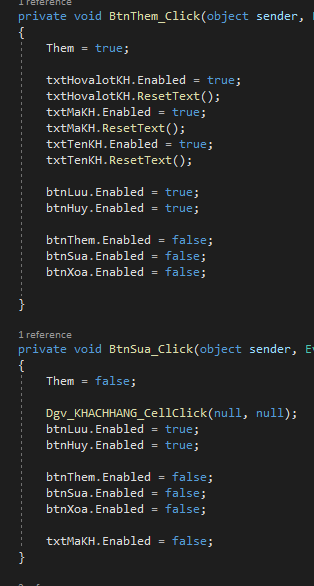


Đây là Source code: Code của Class này bao gồm hàm LoadData(), các sự kiện click của Thêm, Xoá, Sửa, Huỷ, Lưu, sự kiện CellClick của DataGridView.

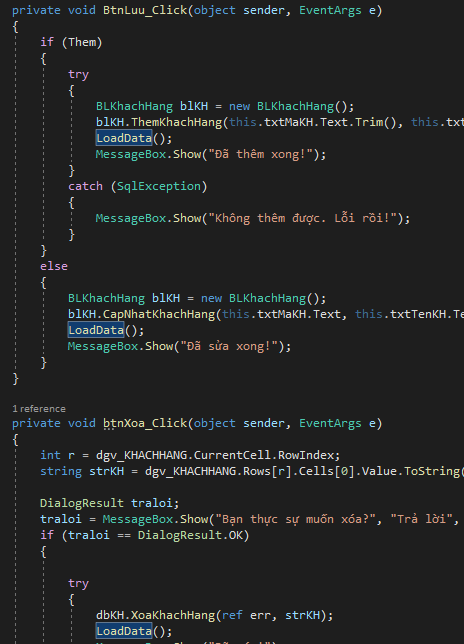
* Hàm LoadDatat();



* Sự kiện click của Button Click và Buttom Sửa.

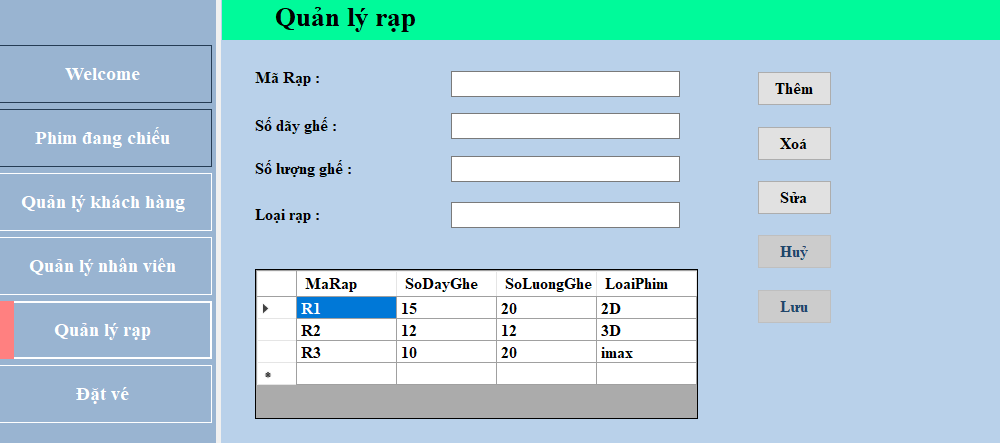


* Sự kiện click của Button Lưu và button Xoá.

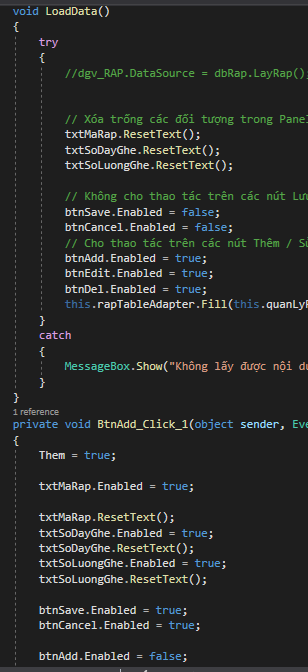


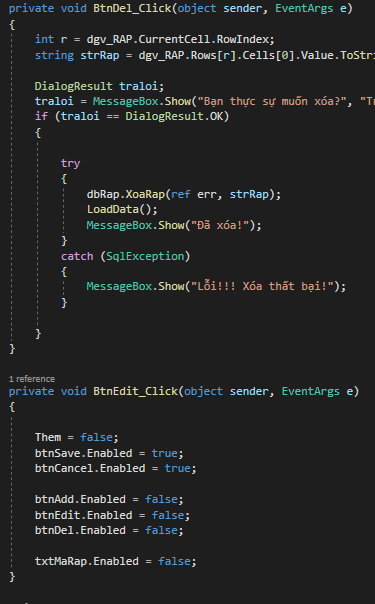
#### QuanLyRap.cs

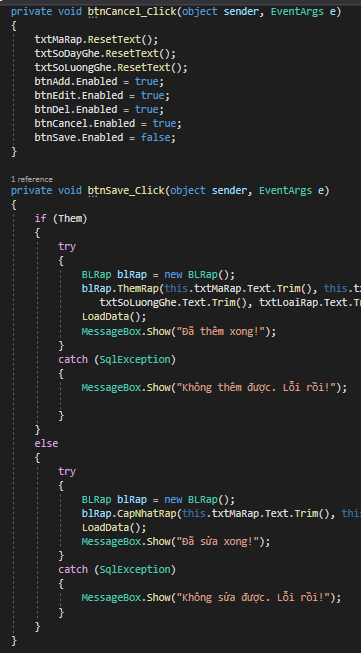
Giao diện của tầng QuanLyRap



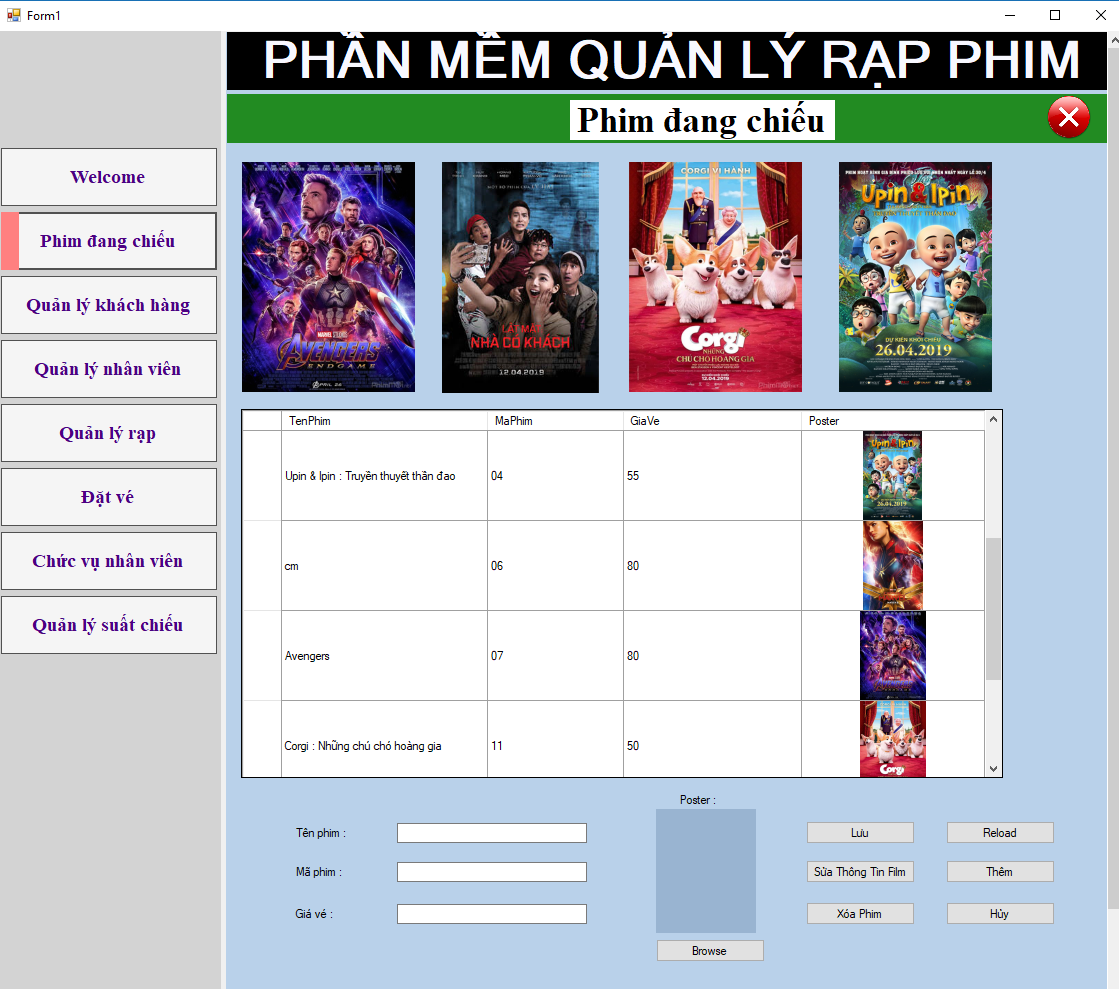
Code của Class này cũng bao gồm hàm LoadData(), các sự kiện click của Thêm, Xoá, Sửa, Huỷ, Lưu, sự kiện CellClick của DataGridView.



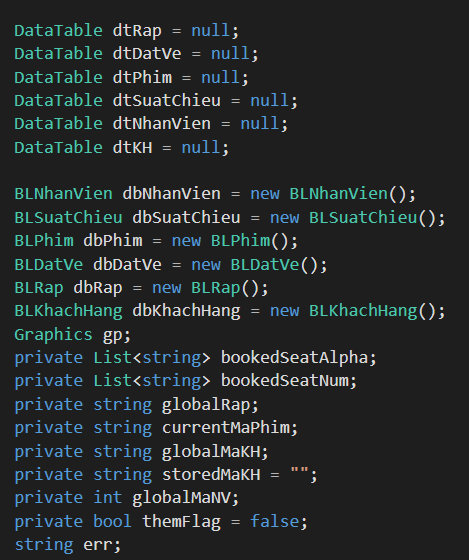




#### PhimUC.cs



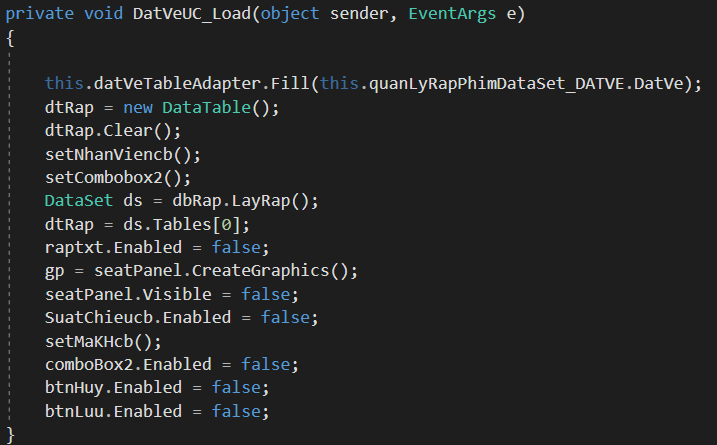
#### DatVe.cs

* Biến chung

Tạo các datatable liên quan đến đặt vé và các dataset.

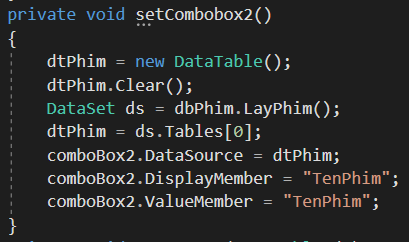
Ngoài ra tạo 2 list ghế và số ghế, biến cục rạp. mã phim, mã khách hàng (xóa cái stored makh đi). Cờ thêm, biến error

* Hàm load UC



Tạo rạp và khởi tạo combobo các combobox

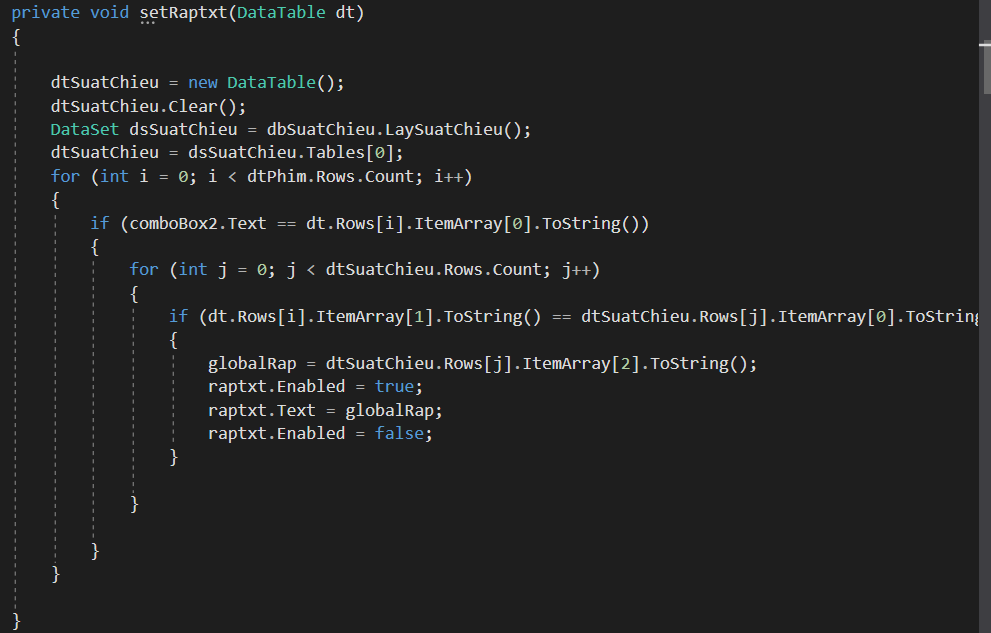
* Combobox2





Đổ dữ liệu tên phim vào combobox.

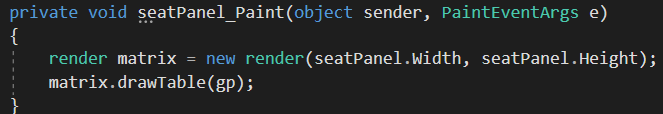
* setRaptxt



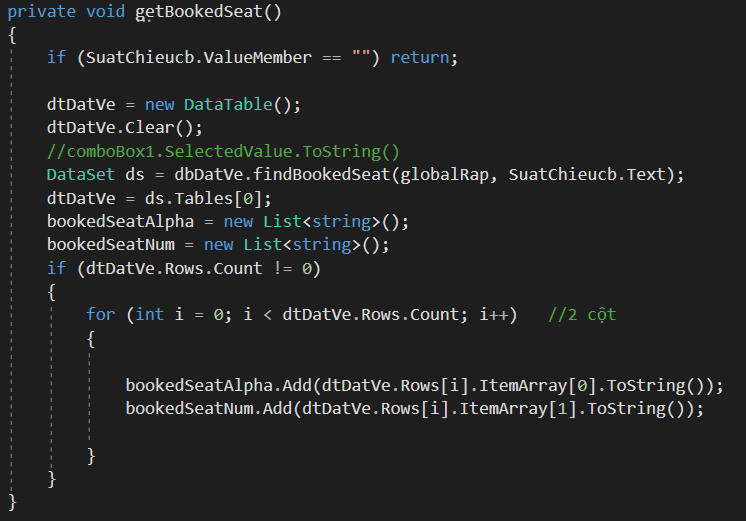


Đổ dữ liệu khi đã chọn tên phim và chọn suất chiếu.

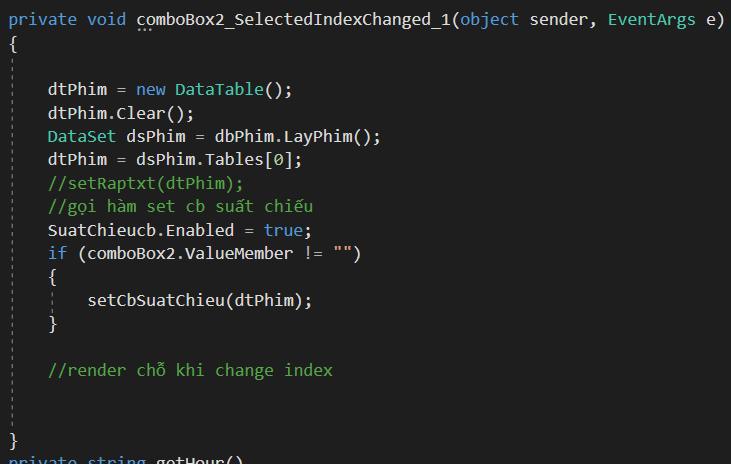
* seatpanelPaint



Hàm vẽ trên panel

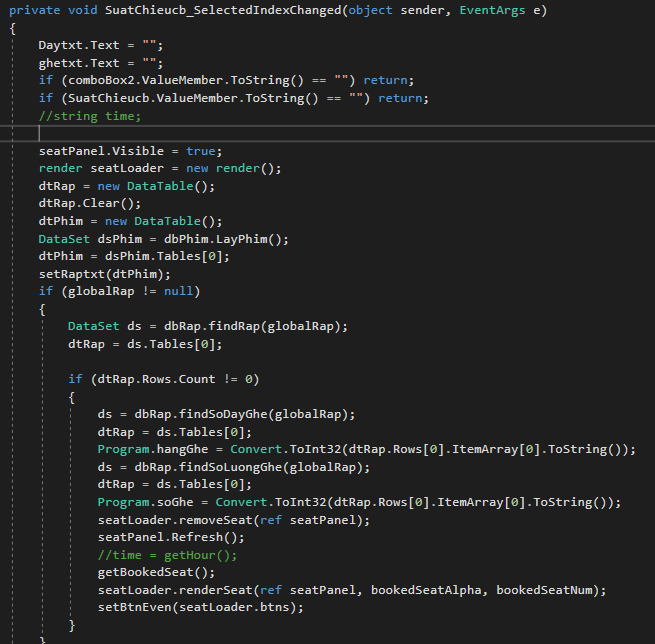


Hàm lấy dữ liệu từ đặt vé để render lên bảng ghế.

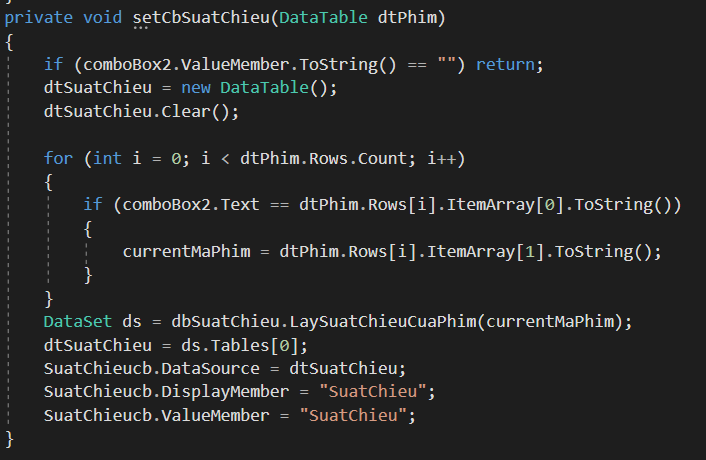


Set suất chiếu. bỏ cái dòng xanh thứ 3 nha

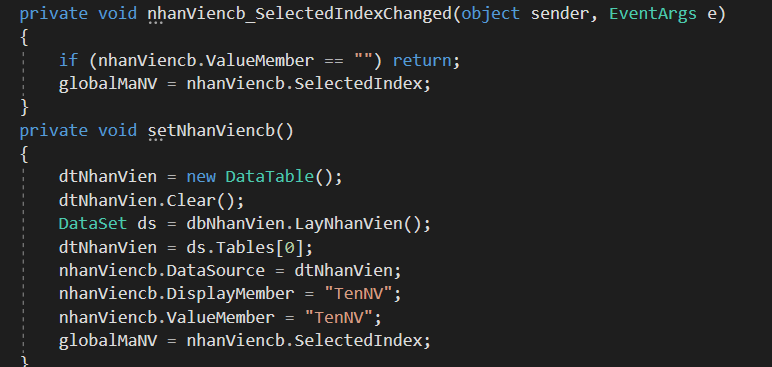
* Event suất chiếu



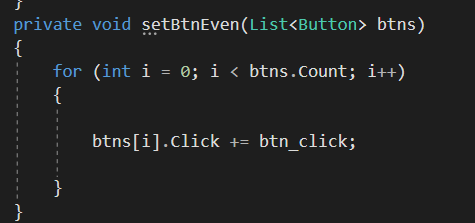
* Set combobox suất chiếu:



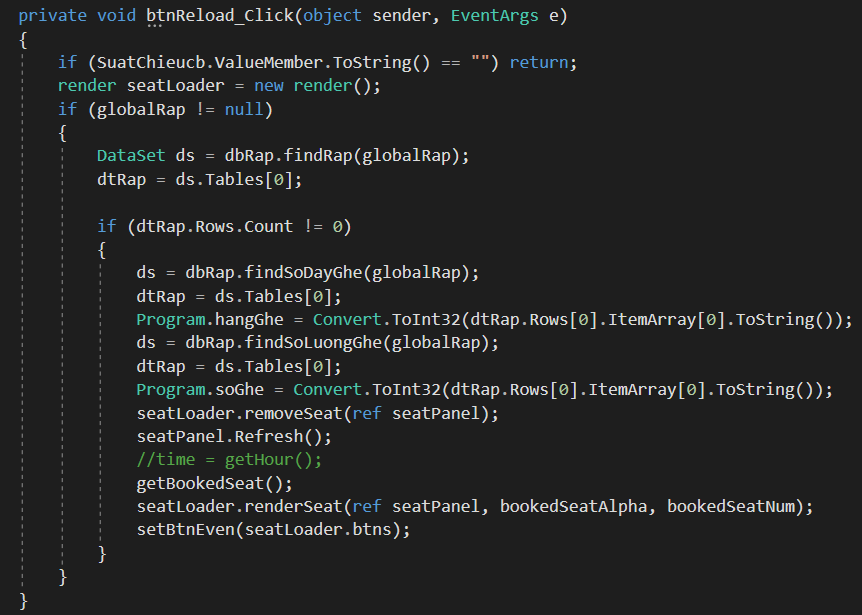
* Set combobox nhân viên:



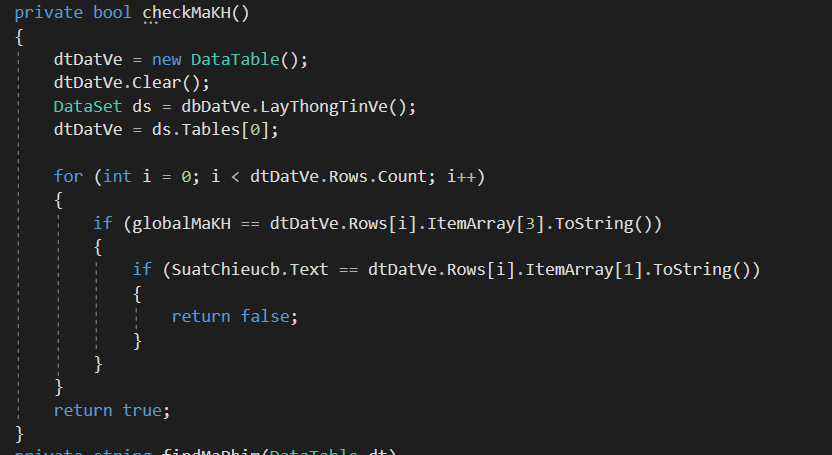
* Event click button



* Set nút reload

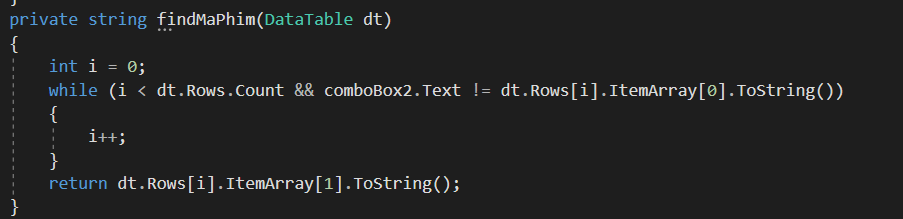


* Hàm checkMaKH



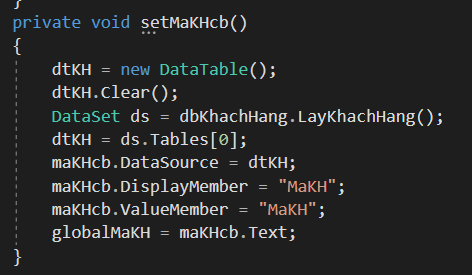
Kiểm tra xem khách hàng đã đăng kí chưa.

* Tìm mã phim



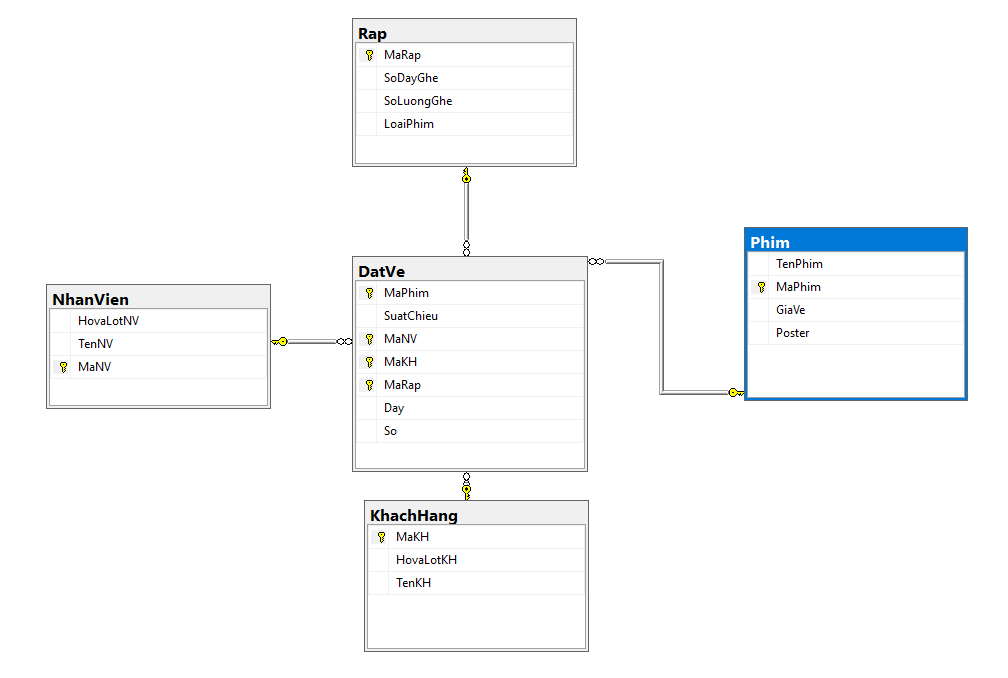
Hàm tìm mã phim

* Hàm set combobox mã kahch1 hàng



### Database Diagram

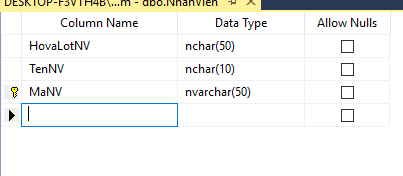
Đây là database diagram của đề tài. Bao gồm 5 bảng: Rap, NhanVien, DatVe, Phim, KhachHang.



##### Figure . Database Diagram

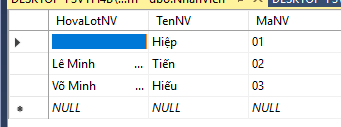
#### Bảng NhanVien

Phần design kiểu dữ liệu của bảng NhanVien, được thiết lập với khoá chính là MaNV.



##### Figure . Design NhanVien

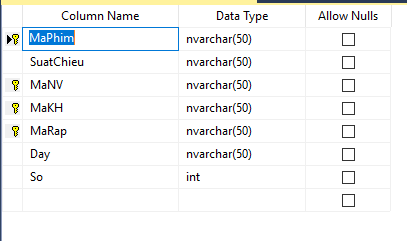
Phần data của NhanVien.



##### Figure . Data NhanVien

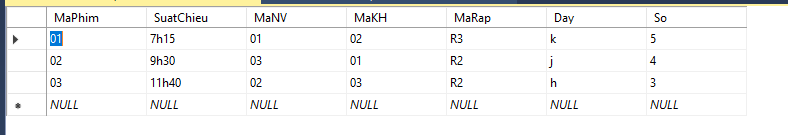
#### Bảng DatVe

Phần design kiểu dữ liệu của bảng DatVe với các khoá chính gồm có: MaPhim, MaNV, MaKH, MaRap.



##### Figure . Design DatVe

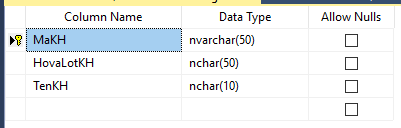
Phần data của DatVe



##### Figure . Data DatVe

#### Bảng KhachHang

Phần design kiểu dữ liệu của bảng KhachHang với khoá chính là MaKH.



##### Figure . Design KhachHang

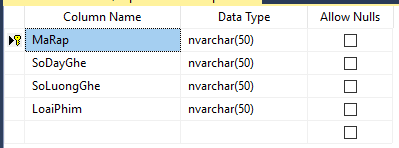
Phần data của KhachHang.



##### Figure . Data KhachHang

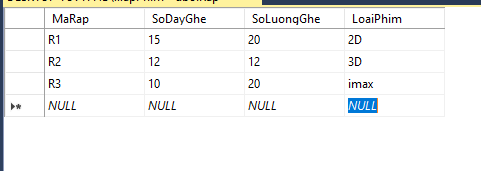
#### Bảng Rap

Phần design kiểu dữ liệu của bảng Rap với khoá chính là MaRap.



##### Figure . Design Rap

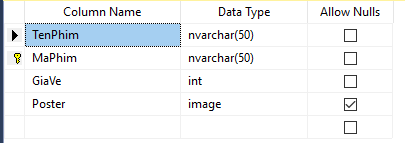
Phần data của Rap



##### Figure . Data Rap

#### Bảng Phim

Phần design kiểu dữ liệu của bảng Phim với khoá chính là MaPhim.



##### Figure . Design của Phim

Phần data của Phim

//Sẽ up load sau khi thêm được hình.

## Kết luận

### Ưu điểm:

### Nhược điểm:

* Hàm renderSeat:
* Khi load khá chậm nếu như rạp quá lớn.
* Mất 1 khoảng thời gian để render khiến người dùng phải chờ đợi.
* Vì sử dụng list<button> để lưu lại nút nên chậm hơn cách kéo thả