

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM**  
**TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**  
**KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KỲ**  
**CHUYÊN ĐỀ CÔNG NGHỆ PHẦN MỀM**

# **Website Quản Lý Quán Karaoke Bằng C#**

*Người hướng dẫn:* **VŨ ĐÌNH HỒNG**

*Người thực hiện:* **PHẠM VŨ QUỐC CƯỜNG – 517H0040**

**TRẦN PHẠM ANH TUẤN–517H0093**

**PHẠM CÔNG–517H0039**

**TÔ QUỐC THẮNG–517H0082**

Lớp : 17050210-17050211

Khoá : 21

THÀNH PHỐ HỒ CHÍ MINH, NĂM 2020

**TỔNG LIÊN ĐOÀN LAO ĐỘNG VIỆT NAM  
TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG  
KHOA CÔNG NGHỆ THÔNG TIN**



**ĐỒ ÁN CUỐI KỲ  
CHUYÊN ĐỀ CÔNG NGHỆ PHẦN MỀM**

# **Website Quản Lý Quán Karaoke Bằng C#**

*Người hướng dẫn:* **VŨ ĐÌNH HỒNG**

*Người thực hiện:* **PHẠM VŨ QUỐC CƯỜNG – 517H0040**

**TRẦN PHẠM ANH TUẤN–517H0093**

**PHẠM CÔNG–517H0039**

**TÔ QUỐC THẮNG–517H0082**

Lớp : 17050210-17050211

Khoá : 21

**THÀNH PHỐ HỒ CHÍ MINH, NĂM 2020**

## LỜI CẢM ƠN

Em xin gửi lời cảm ơn chân thành nhất đến quý thầy cô TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG, những người đã dìu dắt em tận tình, đã truyền đạt cho em những kiến thức và bài học quý báu trong suốt thời gian em theo học tại trường.

Em xin trân trọng gửi lời cảm ơn đến tất cả các thầy cô trong khoa Công Nghệ Thông Tin, đặc biệt là thầy Vũ Đình Hồng đã tận tình hướng dẫn và hỗ trợ chúng em trong suốt quá trình hoàn thành đề tài này. Với sự chỉ bảo tận tình của thầy, em đã có những định hướng tốt trong việc triển khai và thực hiện các yêu cầu của đề tài. Tuy nhiên, với vốn kiến thức còn hạn chế nên sai sót là điều không thể tránh khỏi trong suốt quá trình làm bài. Chúng em rất mong nhận được những ý kiến đóng góp, sửa chữa, phê bình từ phía thầy cô. Để có thể cải thiện thêm kiến thức của mình và biến chúng thành hành trang vô cùng đáng quý của chúng em trong tương lai sau này.

Ngoài ra, em cũng xin gửi lời cảm ơn tới tất cả bạn bè, đặc biệt các bạn trong lớp luôn gắn bó, cùng học tập và đưa ra quan điểm góp phần xây dựng đề tài này.

Em xin chân thành cảm ơn!

## **ĐỒ ÁN ĐƯỢC HOÀN THÀNH TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG**

Chúng tôi xin cam đoan đây là sản phẩm đồ án của riêng chúng tôi và được sự hướng dẫn của thầy Vũ Đình Hồng. Các nội dung nghiên cứu, kết quả trong đề tài này là trung thực và chưa công bố dưới bất kỳ hình thức nào trước đây. Những số liệu trong các bảng biểu phục vụ cho việc phân tích, nhận xét, đánh giá được chính tác giả thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo.

Ngoài ra, trong đồ án còn sử dụng một số nhận xét, đánh giá cũng như số liệu của các tác giả khác, cơ quan tổ chức khác đều có trích dẫn và chú thích nguồn gốc.

**Nếu phát hiện có bất kỳ sự gian lận nào tôi xin hoàn toàn chịu trách nhiệm về nội dung đồ án của mình.** Trường đại học Tôn Đức Thắng không liên quan đến những vi phạm tác quyền, bản quyền do tôi gây ra trong quá trình thực hiện (nếu có).

*TP. Hồ Chí Minh, ngày 30 tháng 11 năm 2020*

*Tác giả*

*(ký tên và ghi rõ họ tên)*

*Phạm Vũ Quốc Cường*

*Phạm Công*

*Tô Quốc Thắng*

*Trần Phạm Anh Tuấn*

## **PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN**

### **Phần xác nhận của GV hướng dẫn**

---

---

---

---

---

---

---

Tp. Hồ Chí Minh, ngày    tháng    năm 2020

(kí và ghi họ tên)

### **Phần đánh giá của GV chấm bài**

---

---

---

---

---

---

---

Tp. Hồ Chí Minh, ngày    tháng    năm 2020

(kí và ghi họ tên)

## MỤC LỤC

<b>LỜI CẢM ƠN .....</b>	<b>II</b>
<b>ĐỒ ÁN ĐƯỢC HOÀN THÀNH .....</b>	<b>III</b>
<b>TẠI TRƯỜNG ĐẠI HỌC TÔN ĐỨC THẮNG .....</b>	<b>III</b>
<b>PHẦN XÁC NHẬN VÀ ĐÁNH GIÁ CỦA GIẢNG VIÊN .....</b>	<b>IV</b>
<b>MỤC LỤC .....</b>	<b>1</b>
<b>DANH MỤC CÁC HÌNH ẢNH VÀ BẢNG BIỂU .....</b>	<b>4</b>
<b>I. GIỚI THIỆU ĐỀ TÀI .....</b>	<b>5</b>
I.1. LỜI NÓI ĐẦU VỀ ĐỀ TÀI: .....	5
I.2. PHÂN TÍCH PHẦN MỀM: .....	5
<i>I.2.1 Khảo sát nghiệp vụ: .....</i>	<i>5</i>
<i>I.2.2 Mô tả hệ thống: .....</i>	<i>6</i>
I.3. XÁC ĐỊNH YÊU CẦU: .....	8
<i>I.3.1 Yêu cầu chức năng nghiệp vụ: .....</i>	<i>8</i>
<i>I.3.2 Yêu cầu hệ thống: .....</i>	<i>8</i>
<i>I.3.3 Yêu cầu phi chức năng: .....</i>	<i>9</i>
<b>II. CƠ SỞ LÝ THUYẾT .....</b>	<b>10</b>
II.1. HỖ TRỢ: .....	10
II.2. ASP.NET MVC: .....	11
III.3 ĐỊNH TUYẾN URL (ROUTING) .....	15
<b>III. THIẾT KẾ CÁC DIAGRAM .....</b>	<b>16</b>
III.1. USE CASE DIAGRAM: .....	16
<i>III.1.1.Actor .....</i>	<i>17</i>

<i>III.1.2. Use case</i> .....	17
III.2. ENTITY RELATIONSHIP DIAGRAM: .....	21
III.3. CLASS DIAGRAM: .....	23
III.4. SEQUENCE DIAGRAM: .....	25
<i>III.4.1 Admin:</i> .....	25
<i>III.4.2 Staff:</i> .....	26
<i>III.4.2 Accountant:</i> .....	27
<b>V. XÂY DỰNG HỆ THỐNG VÀ GIẢI THÍCH CODE</b> .....	<b>28</b>
V.1. XÂY DỰNG CHỨC NĂNG PHÂN QUYỀN CÁC TÀI KHOẢN .....	28
<i>V.1.1 Quá trình xây dựng:</i> .....	28
<i>V.1.2 Code demo minh chứng:</i> .....	29
V.2. XÂY DỰNG CHỨC NĂNG ĐẶT PHÒNG (BOOKING) .....	31
<i>V.2.1 Quá trình xây dựng:</i> .....	31
<i>V.2.2 Code demo minh chứng:</i> .....	32
V.3. XÂY DỰNG CHỨC NĂNG QUẢN LÝ DOANH THU (REVENUE) .....	37
<i>V.3.1 Quá trình xây dựng:</i> .....	37
<i>V.3.2 Code demo minh chứng:</i> .....	38
<b>VI. KẾT QUẢ ĐẠT ĐƯỢC</b> .....	<b>43</b>
VI.1. HOME PAGE: .....	43
VI.2. HÓA ĐƠN: .....	45
VI.3. BIỂU ĐỒ: .....	49
<b>VII. KẾT LUẬN</b> .....	<b>50</b>
VII.1. ƯU, KHUYẾT ĐIỂM CỦA ĐỀ TÀI .....	50
VII.2. NHỮNG GÌ ĐÃ LÀM ĐƯỢC: .....	50

VII.3. NHỮNG GÌ CHƯA LÀM ĐƯỢC:.....	51
4. HƯỚNG PHÁT TRIỂN CHO ĐỀ TÀI:.....	51
<b>VIII. TÀI LIỆU THAM KHẢO.....</b>	<b>52</b>



## DANH MỤC CÁC HÌNH ẢNH VÀ BẢNG BIỂU

### DANH MỤC HÌNH

<b>Hình 1 1:</b> Mô hình tổ chức quản lý hoạt động của doanh nghiệp .....	6
<b>Hình 3 1:</b> Karaoke Use case Diagram .....	16
<b>Hình 3 2:</b> Karaoke ER Diagram .....	22
<b>Hình 3 3:</b> Karaoke Class Diagram.....	24
<b>Hình 3 4:</b> Admin Karaoke Sequence Diagram.....	25
<b>Hình 3 5:</b> Staff Karaoke Sequence Diagram .....	26
<b>Hình 3 6:</b> Accountant Karaoke Sequence Diagram .....	27
<b>Hình 6 1:</b> Home Page .....	43
<b>Hình 6 2:</b> Hóa đơn .....	<b>Error! Bookmark not defined.</b>
<b>Hình 6 3:</b> Biểu đồ doanh thu .....	<b>Error! Bookmark not defined.</b>

## **I. GIỚI THIỆU ĐỀ TÀI**

### **I.1. Lời nói đầu về đề tài:**

Ngày nay công nghệ thông tin vô cùng phát triển thì mọi người đều sử dụng máy vi tính để làm việc. Công nghệ thông tin cũng được áp dụng rất nhiều vào các lĩnh vực mà điển hình là lĩnh vực quản lý kinh doanh.

Trong một xã hội hiện đại, năng động ngày nay, giải trí đóng vai trò rất quan trọng. Nó giúp con người giải tỏa những căng thẳng trong công việc, học tập... Karaoke là một trong những loại hình giải trí được đông đảo người dân lựa chọn. Như chúng ta đã biết tại các quán karaoke, việc quản lý kinh doanh thì rất quan trọng nhưng lại còn thủ công, chỉ xây dựng và lưu lại trên sổ sách gây mất thời gian, khó khăn trong việc thống nhất, theo dõi và quản lý cho người dùng. Từ thực tế đó, việc xây dựng được phần mềm quản lý kinh doanh cho quán karaoke là rất cần thiết.

Vì vậy em đã “Xây dựng Website quản lý quán karaoke” với mục đích nghiên cứu, tìm hiểu về và xây dựng website để có thể đáp ứng được nhu cầu quản lý sổ sách cho người dùng, giúp cho mọi người có thể tiết kiệm công sức, thời gian và quản lý dễ dàng, tiện lợi hơn.

### **I.2. Phân tích phần mềm:**

#### **I.2.1 Khảo sát nghiệp vụ:**

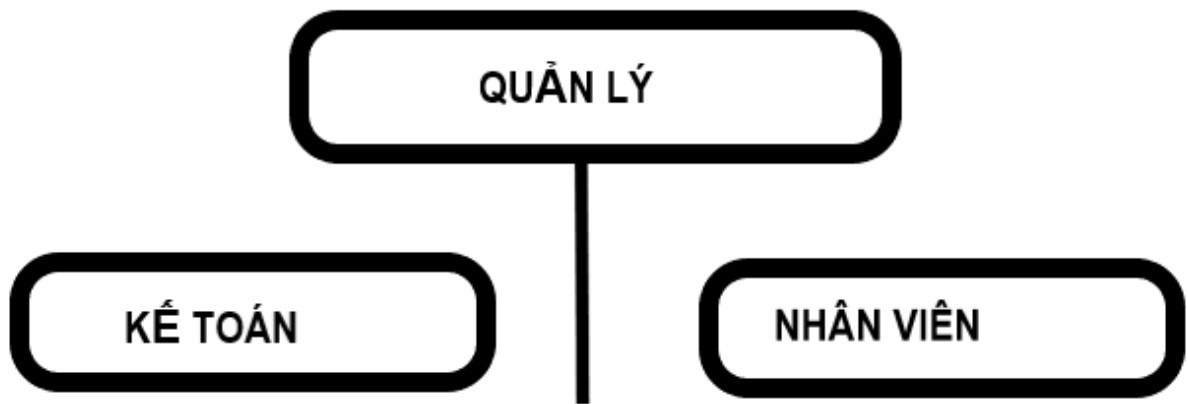
Qua quá trình khảo sát hiện tại đã thu thập được những thông tin nghiệp vụ như sau:

Doanh nghiệp kinh doanh Karaoke có nhiều phòng, mỗi phòng có những thông tin lưu trữ như mã phòng, loại phòng, giá phòng và trạng thái của phòng. Đồng thời sẽ có một trang menu quản lý những thức ăn, nước uống để có thể phục vụ cho khách hàng. Mỗi món sẽ chứa các thông tin như là mã thức ăn, tên thức ăn, giá cả và số lượng hiện đang có trong kho hàng. Mỗi khách hàng sau khi sử dụng xong dịch vụ sẽ được

tính tiền và nhận hóa đơn từ nhân viên, thông tin hóa đơn bao gồm: tên nhân viên xử lý hóa đơn, mã hóa đơn, thông tin khách hàng, thời gian sử dụng phòng, thông tin phòng, tiền giờ, tiền dịch vụ, thuế thu nhập VAT.

### I.2.2 Mô tả hệ thống:

Doanh nghiệp Karaoke có người quản lý tất cả các hoạt động của doanh nghiệp. Mô hình tổ chức như sau:



**Hình 1 1:** Mô hình tổ chức quản lý hoạt động của doanh nghiệp

Nghịệp vụ của từng nhân viên:

- Kế toán (Accountant): tham mưu quản lý các hạng mục doanh nghiệp để tra cứu, thanh toán hóa đơn cho khách hàng. Tổng kết doanh thu, đơn hàng và báo cáo tình hình của doanh nghiệp.
- Nhân viên (Staff): Tham mưu quản lý các phòng, dịch vụ của doanh nghiệp. Tiếp nhận thanh toán, xử lý các yêu cầu dịch vụ của khách hàng.

- Quản lý (Admin): Quản lý tất cả các hoạt động của doanh nghiệp, bộ phận thu ngân, nhân viên. Thu nhận tổng kết doanh thu, chi phí của doanh nghiệp. Đề ra kế hoạch kinh doanh, các hoạt động của doanh nghiệp.

Các hoạt động nghiệp vụ hệ thống:

- Quản lý thuê phòng:
- Chi tiết sử dụng dịch vụ.
- Nhu cầu đổi phòng vì những sự cố bất thường.
- Thanh toán.
- In hóa đơn.

Quản lý các doanh mục:

Doanh mục phòng:

- Thêm phòng mới.
- Xóa phòng hiện tại.

Doanh mục dịch vụ:

- Thêm dịch vụ mới.
- Xóa dịch vụ hiện tại.
- Sửa thông tin dịch vụ. Các thông tin như giá tiền và số lượng đang có.

Doanh mục tài khoản: (Do quản lý thực hiện)

- Thêm tài khoản mới.
- Xóa tài khoản hiện tại.
- Sửa thông tin tài khoản. Các thông tin như tên, số điện thoại, phân quyền tài khoản, mật khẩu tài khoản.

Thống kê doanh thu: (Do bộ phận kế toán quản lý)

- Thống kê theo doanh thu theo tháng.
- Thống kê số lượng đơn được đặt theo tháng.
- Thống kê theo các loại phòng đã được đặt.
- Báo cáo doanh thu theo năm.

Tra cứu thông tin, lọc danh sách thuê phòng. (Theo ngày và tên khách hàng).

### **I.3. Xác định yêu cầu:**

#### **I.3.1 Yêu cầu chức năng nghiệp vụ:**

Hỗ trợ các chức năng quản lý tổng quát của phần mềm: quản lý thuê phòng, quản lý các doanh mục, tài khoản, thống kê, tra cứu. Ứng dụng có giao diện thân thiện dễ sử dụng, có khả năng nâng cấp và bổ sung sửa chữa sau này.

#### **I.3.2 Yêu cầu hệ thống:**

##### **a. Phân quyền sử dụng:**

- ✓ Người quản trị quản lý (admin): được phép sử dụng tất cả các chức năng công cụ của toàn hệ thống.
- ✓ Bộ phận thu ngân và nhân viên: chỉ được phép sử dụng các chức năng công cụ phù thuộc vào chức vụ của mình.

##### **b. Môi trường:**

- ✓ Máy cần cài đặt mạng.
- ✓ Máy in trắng đen hoặc cao hơn và khổ giấy cần dùng là A4 theo yêu cầu của file pdf.

### **I.3.3 Yêu cầu phi chức năng:**

- Hình thức tra cứu tiện dụng, dễ sử dụng cho cả những người không chuyên về tin học. Hỗ trợ tra cứu nâng cao, có hình ảnh minh họa dễ hiểu, dễ sử dụng, thiết kế đơn giản, sắp xếp hợp lí, dễ dàng tìm thấy các công cụ chức năng khi cần.
- Các màn hình có sự nhất quán chung: Các trang đều được hiển thị theo một form chính.
- Tốc độ tra cứu và sử dụng các tính năng từ trung bình đến nhanh.

## II. CƠ SỞ LÝ THUYẾT

### II.1. Hỗ trợ:

#### MSSQL:

- SQL Server được phát triển bởi Microsoft.
- Là một hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System hay được viết tắt là RDBMS) sử dụng câu lệnh SQL (**Transact-SQL**)  
SQL Server là một máy chủ cơ sở dữ liệu, là sản phẩm phần mềm có chức năng chính là lưu trữ và truy xuất dữ liệu theo yêu cầu của các ứng dụng phần mềm khác.
- Ngoài ra SQL Server được tối ưu để có thể chạy trên môi trường cơ sở dữ liệu rất lớn (Very Large Database Environment) lên đến Tera-Byte và có thể phục vụ cùng lúc cho nhiều user. SQL Server cũng có thể kết hợp với các server khác như Microsoft Internet Information Server (IIS), E-Commerce Server, Proxy Server... hay các nền tảng như ASP.NET, C# xây dựng Winform cũng có thể chạy hoàn hảo với SQL Server.
- Các hàng và các cột của các bảng dữ liệu chính là đối tượng của SQL Server. Hàng là bản ghi của bảng, cột được gọi là trường dữ liệu. Cột dữ liệu và kiểu dữ liệu xác định tạo nên cấu trúc của bảng. Khi bảng được tổ chức thành một hệ thống cho một mục đích sử dụng cụ thể vào công việc nào đó sẽ trở thành một cơ sở dữ liệu.

**HTML (Hypertext Markup Language):** Giúp người dùng tạo và cấu trúc các thành phần trong trang web hoặc ứng dụng, phân chia các đoạn văn, heading, links, Blockquote element ...

**JavaScript:** Là ngôn ngữ lập trình phổ biến được sử dụng để tạo ra những trang web động. Nó được tích hợp và nhúng trong HTML. JavaScript cho phép kiểm soát các hành vi của trang web tốt hơn so với khi chỉ có một mình HTML. JavaScript là ngôn ngữ lập trình được hỗ trợ hầu như trên tất cả các trình duyệt như Firefox, Chrome... thậm chí các trình duyệt trên thiết bị di động.

**Entity Framework:** Là một framework ánh xạ quan hệ đối tượng (ORM) dành cho ADO.NET, là 1 phần của .NET Framework. Entity Framework cho phép các nhà phát triển web tương tác với dữ liệu quan hệ theo phương pháp hướng đối tượng đặc trưng. Giúp lập trình viên giảm thiểu việc lập trình mã nguồn cần thiết để truy cập và tương tác với cơ sở dữ liệu. Entity Framework là công cụ làm việc với cơ sở dữ liệu được Microsoft khuyến nghị.

## II.2. ASP.NET MVC:

**ASP.NET:** Là một nền tảng dành cho phát triển web, được Microsoft phát hành và cung cấp đầu tiên vào năm 2002. Nền tảng được sử dụng và tạo ra các ứng dụng web-based. Thành phần của ASP.NET và đặc điểm chính:

- **Ngôn ngữ:** Có rất nhiều ngôn ngữ lập trình khác nhau tồn tại trong .NET framework. Các ngôn ngữ này là VB.net và C#, có thể được sử dụng để phát triển các ứng dụng web.
- **Thư viện:** .NET Framework gồm có một bộ các lớp library chuẩn. Library phổ biến nhất được sử dụng cho các ứng dụng web trong .NET là Web library. Web library bao gồm tất cả các thành phần cần thiết để sử dụng trong phát triển các ứng dụng web-based.
- **Thời gian chạy ngôn ngữ chung (CLR):** Các chương trình .net đều được chạy trên nền tảng này. CLR được sử dụng để thực hiện các hoạt động chính.



Các hoạt động bao gồm xử lý ngoại lệ (các lỗi xảy ra trong quá trình thực hiện ứng dụng) và thu gom rác (loại bỏ các tài nguyên không mong muốn khi chúng không còn cần thiết).

- **Code Behind Mode/ trạng thái code rời:** Khái niệm về tách rời thiết kế và mã code giúp việc duy trì ứng dụng ASP.NET dễ dàng hơn. Loại file thông dụng ASP.NET là ASPX.
- **State Management / Quản lí trạng thái:** ASP.NET có cơ sở để quản lý trạng thái. HTTP được biết đến như một giao thức phi trạng thái.
- **Caching / bộ nhớ đệm:** ASP.NET cũng có thể thực hiện chức năng của caching, qua đó cải thiện hiệu suất làm việc cho ứng dụng. Với việc lưu bộ nhớ đệm, các trang thường xuyên được người dùng yêu cầu có thể lưu trữ ở một vị trí tạm thời. Các trang này có thể truy xuất nhanh hơn và người dùng có thể nhận được phản hồi tốt hơn.

**MVC:.** Viết tắt của Model – View – Controller, là mô hình phân bố source code thành 3 phần, mỗi thành phần có một nhiệm vụ riêng biệt và độc lập với các thành phần khác.

- **Model:** Là bộ phận có chức năng lưu trữ toàn bộ dữ liệu của ứng dụng. Bộ phận này là cầu nối giữa 2 thành phần View và Controller. Model thể hiện dưới hình thức là một cơ sở dữ liệu hoặc có khi chỉ đơn giản là một file XML bình thường. Model thể hiện rõ các thao tác với cơ sở dữ liệu như cho phép xem, truy xuất, xử lý dữ liệu...
- **View:** Là phần giao diện dành cho người sử dụng. Nơi mà người dùng có thể lấy được thông tin dữ liệu của MVC thông qua các thao tác truy vấn như tìm kiếm hoặc sử dụng thông qua các website. Trong MVC View là nơi tạo ra các HTML. Bên cạnh đó, View cũng có chức năng ghi nhận hoạt động của người dùng để tương tác với Controller. Tuy nhiên, View không có mối quan hệ trực

tiếp nào với Controller, cũng không lấy được dữ liệu từ Controller mà chỉ hiện thị yêu cầu chuyển cho Controller.

- **Controller:** Là nơi tiếp nhận những yêu cầu xử lý được gửi từ người dùng, nó sẽ gồm những class/function xử lý nghiệp vụ logic giúp lấy đúng dữ liệu thông tin cần thiết nhờ các nghiệp vụ lớp Model cung cấp và hiện thị dữ liệu đó ra cho người dùng nhờ lớp View.

### Sử dụng chú thích (Data Annotation)

Chú thích dữ liệu (Data Annotation) trong Entity Framework giúp bạn định nghĩa dữ liệu thuộc tính trực tiếp trong mã nguồn được gieo từ các bảng cơ sở dữ liệu và giúp bạn kiểm soát được dữ liệu nhập vào từ người dùng tốt nhất cũng như cách định nghĩa dữ liệu hiển thị trên giao diện.

Các thuộc tính bao gồm:

**Validation Attributes (thuộc tính xác thực):** Dùng để thêm các tập luật xác nhận cho dữ liệu

- **Timestamp:** Chỉ áp dụng cho kiểu dữ liệu thuộc mảng byte, dùng để tạo ra cột dữ liệu thời gian trong database SQL.
- **ConcurrencyCheck:** Áp dụng cho một hay nhiều thuộc tính và các cột tương ứng trong database sẽ được kiểm tra đồng bộ qua mệnh đề Where().
- **Require:** Chỉ định thuộc tính phải có dữ liệu nhập vào trước khi submit về server.
- **MinLength:** Chiều dài tối thiểu của thuộc tính.
- **MaxLength:** Chiều dài tối đa của thuộc tính.
- **StringLength:** Định nghĩa chiều dài của thuộc tính, cho phép đặt cả chiều dài tối đa lẫn tối thiểu.
- **Range:** Định nghĩa giá trị tối thiểu và tối đa của một thuộc tính.

**Display Attributes:** Để định nghĩa thuộc tính hiển thị, đơn giản bạn chỉ cần thêm dòng [DisplayName(“ tên hiển thị ”)] vào trước các thuộc tính cần hiển thị.

**Modelling Attributes:** Nếu tạo dự án theo kiểu Code First thì bạn bắt buộc phải dùng các chú thích thuộc tính để định nghĩa rõ thuộc tính nào là khóa chính, khóa ngoại... cũng như các mối quan hệ giữa các lớp bảng trong database sẽ được đặt từ Code First.

- **Key:** Định nghĩa thuộc tính nào là khóa chính.
- **Table:** Bảng.
- **Column:** Cột.
- **ForeignKey:** Khóa ngoại.

### Xác nhận dữ liệu (Model Validation)

Kiểm tra input của User có hợp lệ hay không. Giúp kiểm tra và hiển thị lỗi nếu cần. Việc xác nhận dữ liệu có thể diễn ra ở 2 nơi:

- **Client-side validation:** Dữ liệu được kiểm tra ngay ở trên trình duyệt mà không cần phải gửi về server giúp tiết kiệm được băng thông và tăng sự trải nghiệm thay vì phải đợi request đến server và trả về.
- **Server-side validation:** Dữ liệu được kiểm tra ở server trong trường hợp các javascript trên trình duyệt bị disable hoặc là không thể chạy. Ngoài ra còn giúp những dữ liệu mà ở client-side không thể xác nhận được thì sẽ được server xác nhận lại.

Ngoài ra ta còn có **ModelState** là nơi lưu chi tiết các thuộc tính của các giá trị được cập nhật lên model và các thông tin lỗi trong quá trình validation xuất hiện trong mỗi thuộc tính.

Những validation attributes thường dùng: Required, MinLength, MaxLength, Range, Datatype, RegularExpression, Url, Compare.

### III.3 Định tuyến URL (Routing)

ASP.NET MVC Framework có một hệ thống định tuyến URL (URL Routing System) linh hoạt cho phép xác định các quy tắc ánh xạ địa chỉ URL bên trong ứng dụng. Mỗi hệ thống định tuyến có 2 tác dụng:

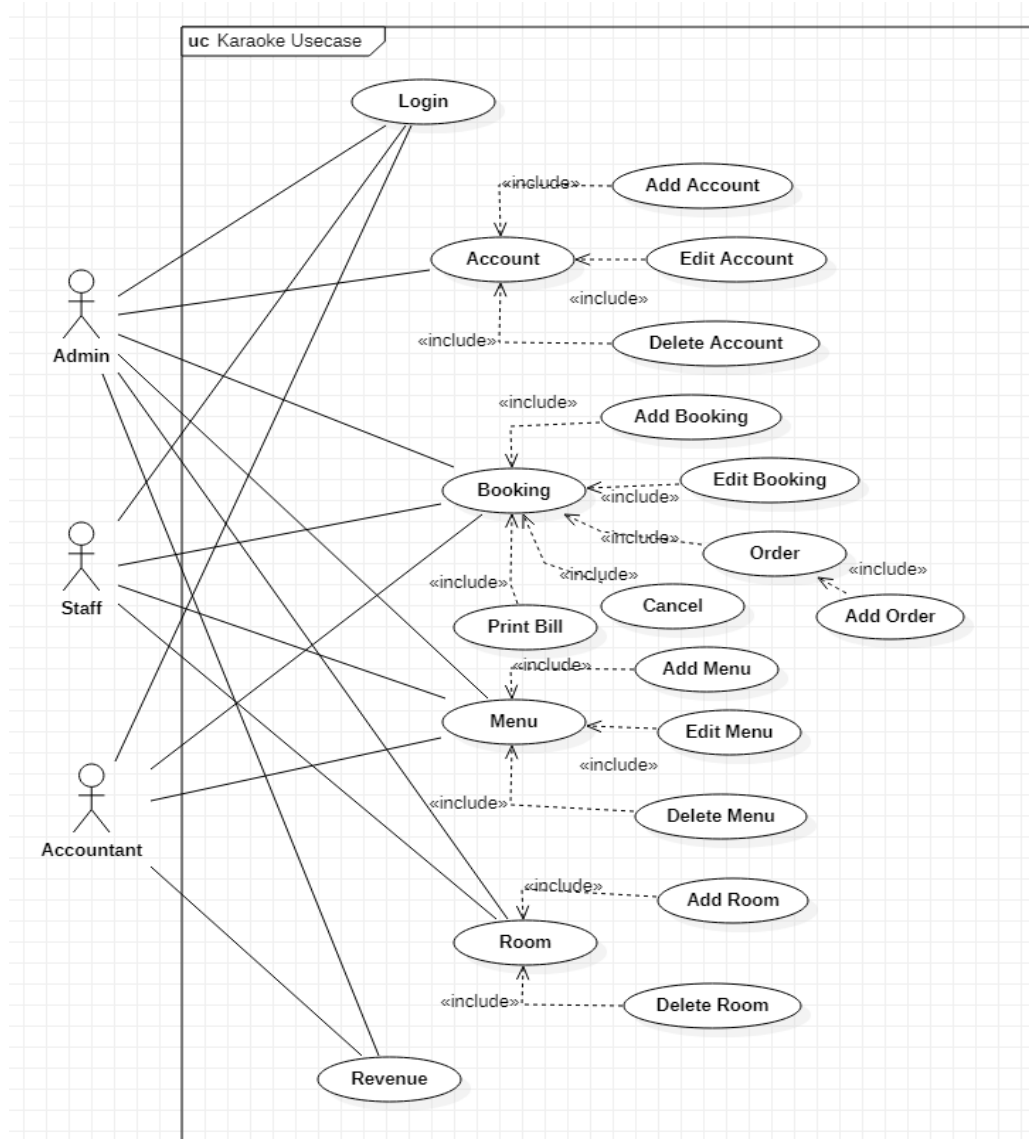
- Xây dựng một tập hợp cái URL đi vào ứng dụng và định tuyến chúng tới các Controller và thực thi các phương thức Action để xử lý.
- Xây dựng các URL gửi đi mà có thể gọi ngược trở lại Controllers Action.

Sử dụng các quy tắc ánh xạ URL để điều khiển URL đi vào và đi ra để tăng tính mềm dẻo cho việc lập trình ứng dụng, nghĩa là nếu muốn thay đổi cấu trúc URL (ví dụ Catalog thành Products) có thể thay đổi một tập hợp quy tắc ánh xạ mức ứng dụng mà không cần phải viết lại mã lập trình bên trong Controllers và Views.

### III. THIẾT KẾ CÁC DIAGRAM

#### III.1. Use case Diagram:

Hệ thống sẽ được phân quyền cho 3 đối tượng chính bao gồm admin, staff và accountant. Các đối tượng sẽ có vai trò trên hệ thống theo use case biểu diễn như sau:



**Hình 3 1:** Karaoke Use case Diagram

### **III.1.1.Actor**

**Admin:** Là người quản lý trực tiếp của quán, mỗi quản lý sẽ được cấp một tài khoản admin với mã khác nhau, quản lý có nhiệm vụ giám sát và theo dõi quá trình hoạt động của quán karaoke. Admin sau khi đăng nhập vào hệ thống có thể truy cập và thao tác trên tất cả các trang. Chỉ admin mới có quyền tạo thêm tài khoản vào hệ thống.

**Staff:** Là nhân viên lễ tân và nhân viên phục vụ, các nhân viên được giao nhiệm vụ này sẽ được cấp một tài khoản staff với mã khác nhau. Các staff sẽ có quyền truy cập và thao tác trên các trang bao gồm: Booking, Menu, Room. Nhân viên lễ tân sẽ thực hiện đặt phòng hát karaoke theo yêu cầu của khách. Nhân viên phục vụ sẽ nhận order từ khách khi đang hát karaoke và truy cập để thêm món vào mã đặt phòng của khách. Khi kết thúc, nhân viên lễ tân sẽ thực hiện thanh toán (check out) và in hoá đơn cho khách, nhân viên phục vụ sẽ cầm hoá đơn để đưa và thu tiền cho khách trong phòng karaoke. Nếu phòng có sự cố không sử dụng được, nhân viên có thể truy cập vào trang quản lý phòng để thay đổi trạng thái phòng.

**Accountant:** Là nhân viên kế toán, sẽ có vai trò quản lý doanh thu theo ngày/tháng/năm hoặc theo quý để có các kế hoạch trả lương, thưởng, quảng cáo dịch vụ, quản lý chất lượng. Nhân viên kế toán có quyền truy cập vào trang booking và quản lý doanh thu.

### **III.1.2.Use case**

#### ***III.1.2.1 Login:***

Nhân viên và quản lý phải đăng nhập vào hệ thống bằng tài khoản đã được cung cấp mới có thể thao tác trên hệ thống.

#### ***III.1.2.2 Account:***

Chỉ quản lý mới có thể sử dụng use case này. Quản lý có thể xem danh sách account hiện tại đã có trên hệ thống.

#### ***III.1.2.3 Add Account:***

Quản lý có thể thêm account vào hệ thống. Các account không thể trùng mã với nhau.

#### ***III.1.2.4 Edit Account***

Quản lý có thể chỉnh sửa thông tin của account trên hệ thống, ví dụ như thay đổi mật khẩu, tên, số điện thoại...

#### ***III.1.2.5 Delete Account***

Quản lý có thể xoá account hiện có trên hệ thống.

#### ***III.1.2.6 Booking***

Quản lý, nhân viên và kế toán khi truy cập vào trang này có thể xem danh sách đặt phòng hiện tại. có thể thực hiện tìm kiếm thông tin đặt phòng trong danh sách theo tên người đặt, mã đặt phòng, mã phòng, số điện thoại, thời gian đặt. Ngoài ra có thể xem lịch sử đặt phòng/ thanh toán bằng cách sử dụng filter ngày. Có thể sort danh sách theo chữ cái, số hoặc ngày theo tăng dần hoặc giảm dần. Có thể thay đổi hiển thị số lượng đặt phòng tối đa (5, 10,15,...). Nhân viên lễ tân sẽ thao tác check in (nhận phòng) cho khách khi tới giờ đặt phòng.

#### ***III.1.2.7 Add Booking***

Nhân viên lễ tân có thể thực hiện đặt phòng, các thông tin đặt phòng bao gồm tên khách hàng, số điện thoại liên hệ, số lượng khách, thời gian bắt đầu, thời gian kết thúc...

#### ***III.1.2.8 Edit Booking***

Nhân viên lễ tân có thể chỉnh sửa mọi thông tin đặt phòng ngoại trừ mã đặt phòng.

#### ***III.1.2.9 Cancel***

Nhân viên lễ tân có thể huỷ lệnh đặt phòng khi trạng thái phòng đang là booking trên danh sách hiển thị. Người dùng không thể huỷ lệnh đặt khi trạng thái đang là using (đang sử dụng) hoặc đã payment success (thanh toán thành công).

### ***III.1.2.10 Order***

Nhân viên và quản lý có thể xem danh sách món ăn và nước uống đã được đặt của khách trong quá trình hát karaoke. Các thông tin được hiển thị trong danh sách gồm tên, số lượng, tổng đơn.

### ***III.1.2.11 Add Order***

Nhân viên phục vụ sẽ là nhân tố chính sử dụng chức năng này. Khi được khách gọi đặt món, nhân viên phục vụ sẽ truy cập vào hệ thống để thêm các món khách yêu cầu.

### ***III.1.2.12 Print Bill***

Nhân viên lễ tân sẽ thực hiện chức năng này để thực hiện thanh toán sau khi khách check out (kết thúc sử dụng). Dữ liệu sẽ được lưu vào lịch sử đặt phòng và trạng thái sẽ được chuyển thành payment success (thanh toán thành công).

### ***III.1.2.13 Menu***

Nhân viên và quản lý có thể truy cập để xem danh sách đồ uống và thức ăn hiện tại đang có trong menu.

### ***III.1.2.14 Add Menu***

Nhân viên và quản lý có thể thêm món ăn hoặc đồ uống và menu. Thông tin món ăn bao gồm tên, đơn giá, số lượng đang có trong kho.

### ***III.1.2.15 Edit Menu***

Nhân viên và quản lý có thể chỉnh sửa thông tin món ăn trong menu như tên, đơn giá.

### ***III.1.2.16 Delete Menu***

Nhân viên và quản lý có thể xóa các món trong menu.

### ***III.1.2.17 Room***

Nhân viên và quản lý có thể xem danh sách phòng hiện có. Thông tin các phòng sẽ bao gồm tên phòng, loại phòng, giá phòng/giờ, trạng thái.



***III.1.2.18 Add Room***

Nhân viên và quản lý có thể thêm phòng vào danh sách phòng.

***III.1.2.19 Delete Room***

Nhân viên và quản lý có thể xoá thông tin phòng trong danh sách.

***III.1.2.20 Revenue***

Quản lý và kế toán có thể truy cập để xem thống kê số liệu và doanh thu hàng tháng. Các thông tin sẽ bao gồm thống kê doanh thu theo tháng được thể hiện bằng biểu đồ đường (line chart), tỉ lệ các loại phòng được book (Standard, Vip, Luxury) được thể hiện bằng biểu đồ tròn (pie chart), số lượng sử dụng phòng mỗi tháng được thể hiện bằng biểu đồ cột (bar chart), tổng doanh thu cả năm.

### III.2. Entity Relationship Diagram:

Để triển khai hệ thống từ các Use case, nhóm đã thảo luận và thiết kế các mô hình thực thể gồm các thực thể như sau:

**Booking:** Thực thể này chứa thông tin đặt phòng. Mã Booking ID sẽ là thuộc tính khoá, các thuộc tính khác bao gồm Start time, End Time, Duration, Total Payment, Activity Status, Amount Of customer, Customer phone.

**Room:** Thực thể này chứa thông tin phòng hát. Mã Room ID là khoá chính, các thuộc tính còn lại bao gồm Room Type, Price, Room status.

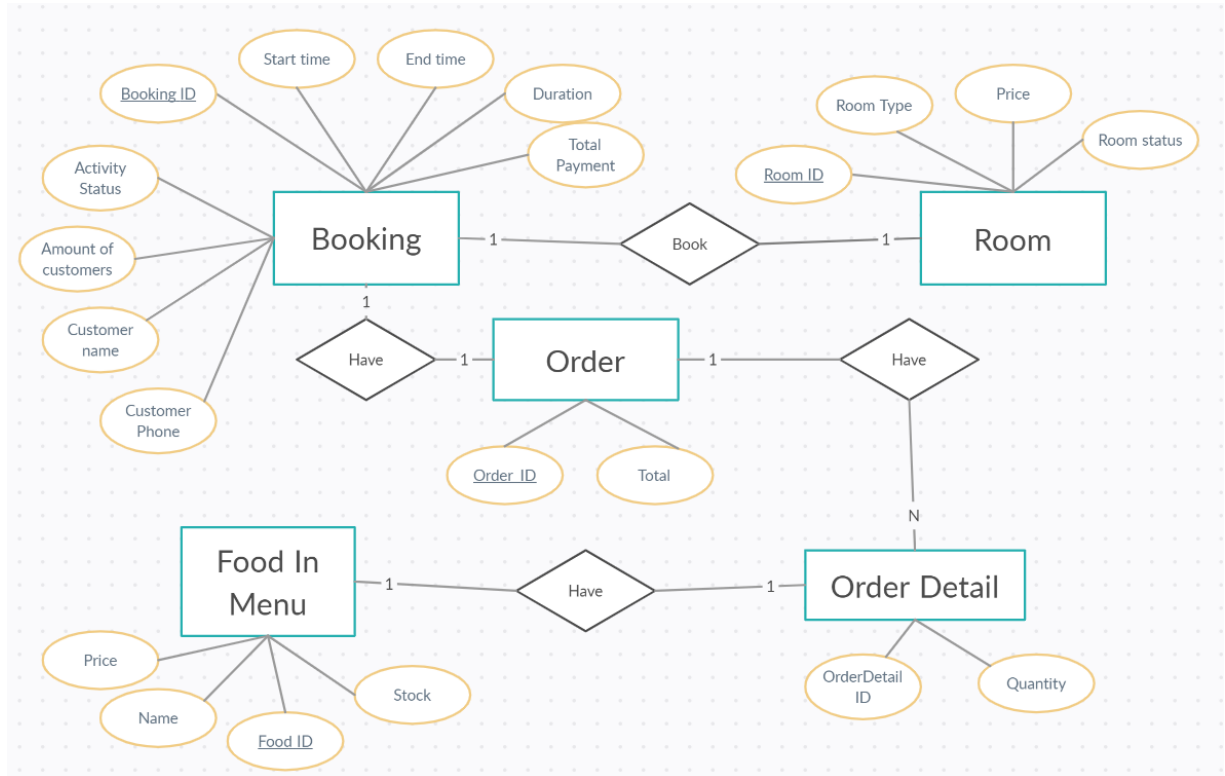
**Order:** Thực thể này sẽ là quan hệ Order giữa Booking với Order Detail. Chứa mã Order ID là khoá chính và thuộc tính Total.

**Order Detail:** Thực thể này chứa thông tin đặt món, chứa mã Order Detail ID là khoá chính và số lượng món Quantity.

**Food In Menu:** Thực thể này chứa thông tin món đang có trong menu. Mã Food ID là khoá chính, các thực thể khác là Price, Name, Stock.

Quan hệ ràng buộc giữa các thực thể được thiết kế như sau:

- Một Booking chỉ Book được một Room.
- Một Room chỉ có thể được book trong một thời điểm.
- Một Booking có thể có một Order.
- Một Order có thể có nhiều Order Detail.
- Một Order Detail chỉ chứa một Food trong Menu.



**Hình 3 2:** Karaoke ER Diagram

### III.3. Class Diagram:

Để thực hiện như các thiết kế trên nhóm đã thảo luận và đưa ra bản thiết kế class diagram để tối ưu yêu cầu nhất. Hệ thống sẽ có 5 bảng bao gồm:

**Booking:** Bảng này sẽ chứa thông tin đặt phòng. Khoá chính sẽ PayID, RoomID là khoá phụ của bảng Room, DateTime chứa thông tin bắt đầu, EndTime chứa thông tin kết thúc, Order\_ID là khoá phụ của bảng Order, Duration chứa thông tin thời gian sử dụng từ khi bắt đầu đến lúc kết thúc. Name\_Cus chứa thông tin tên khách hàng, Phone\_Cus chứa thông tin số điện thoại của khách, Amount\_Cus chứa thông tin số lượng khách hàng sử dụng phòng, P\_Status chứa thông tin trạng thái phòng (3 trạng thái bao gồm Booking, Using và Payment Success) của đơn đặt, Total chứa tổng số tiền khách phải trả.

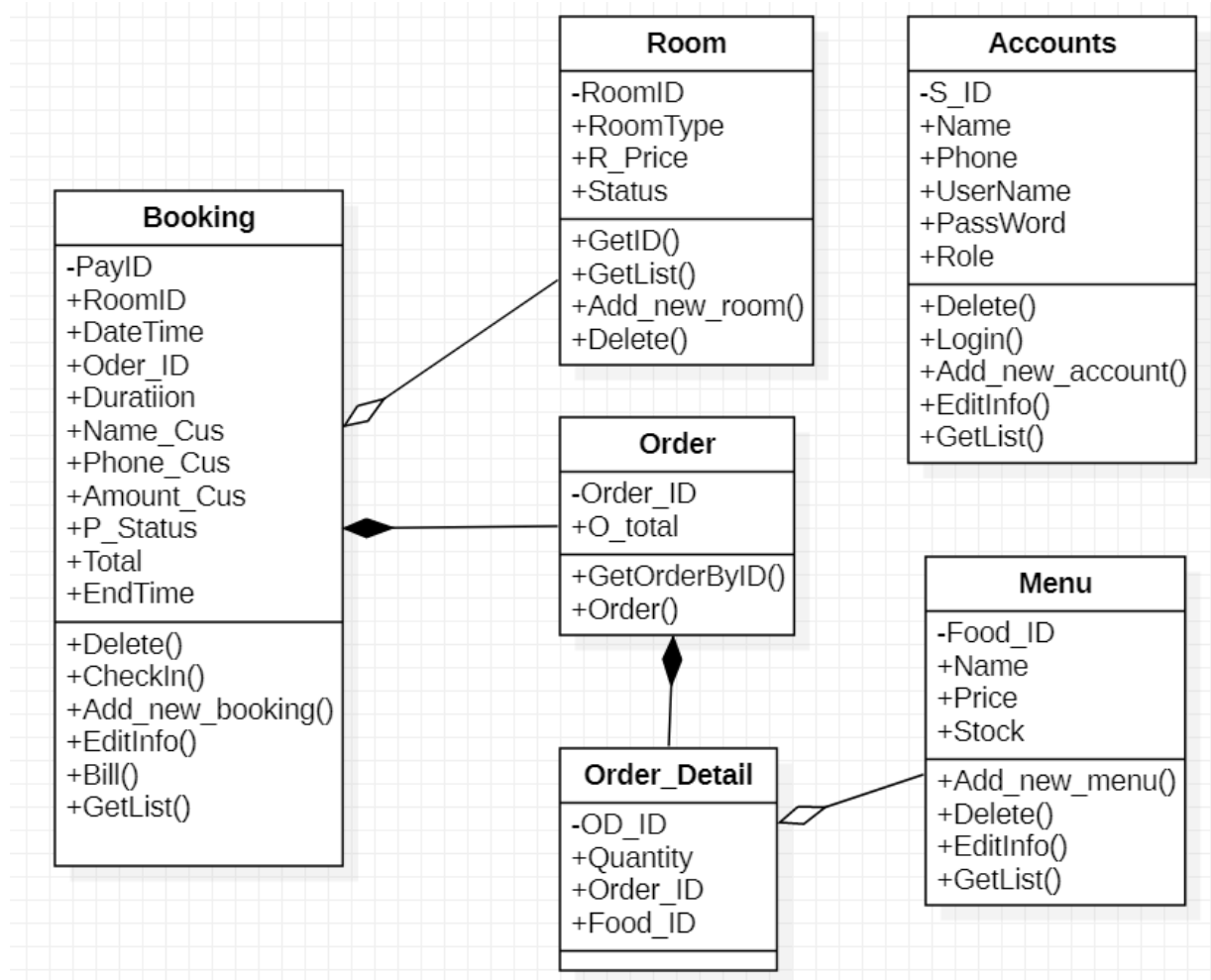
**Room:** Bảng này sẽ chứa thông tin của phòng. Khoá chính là RoomID, RoomType chứa thông tin loại phòng của phòng đó (quán phục vụ 3 loại phòng bao gồm Standard, Vip, Luxury). R\_Price chứa thông tin giá phòng (mỗi loại phòng sẽ có một mức giá khác nhau). Status chứa thông tin trạng thái phòng (Available hoặc Not Available).

**Order:** Là bảng quan hệ giữa bảng Booking và bảng Oder Detail chỉ được tạo ra và gắn liền với một dòng trong bảng booking. Mã Order\_ID là khoá chính, O\_total chứa thông tin tổng giá tiền các món ăn và đồ uống của khách đã gọi trong bảng Oder\_detail.

**Order\_detail:** Chứa thông tin chi tiết các món khách đã gọi trong quá trình sử dụng dịch vụ. mã OD\_ID là khoá chính, Order\_ID là khoá phụ liên kết tới bảng quan hệ Order, Food\_ID chứa mã món là khoá phụ của bảng Menu, Quantity chứa thông tin số lượng của món đã gọi.

**Menu:** Là bảng chứa danh sách món để phục vụ quản lý các món quán đang cung cấp. Mã Food\_ID là khoá chính, Name chứa tên của món, Price chứa giá của món, Stock chứa thông tin số lượng của món đó đang có trong kho.

**Account:** Là bảng chứa danh sách account đang có trên hệ thống. Mã S\_ID là khoá chính, Name chứa tên người dùng, Phone chứa thông tin số điện thoại người dùng, Username chứa thông tin tên đăng nhập, Password chứa thông tin mật khẩu, Role chứa thông tin vai trò của tài khoản trong hệ thống (có 3 vai trò được phân quyền hệ thống khác nhau bao gồm Admin, Staff, Accountant).

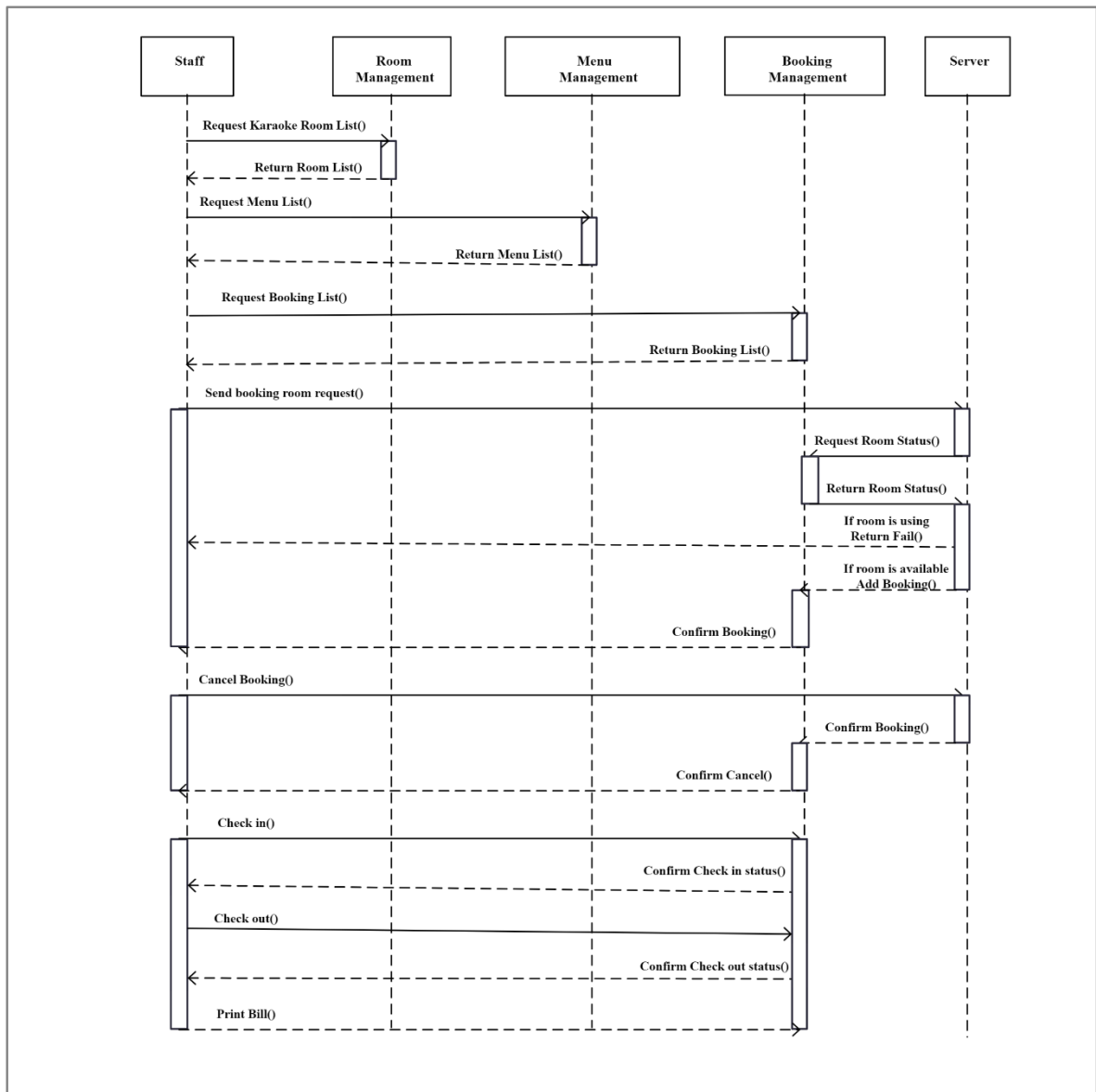


**Hình 3 3:** Karaoke Class Diagram

### III.4. Sequence Diagram:

#### III.4.1 Admin:

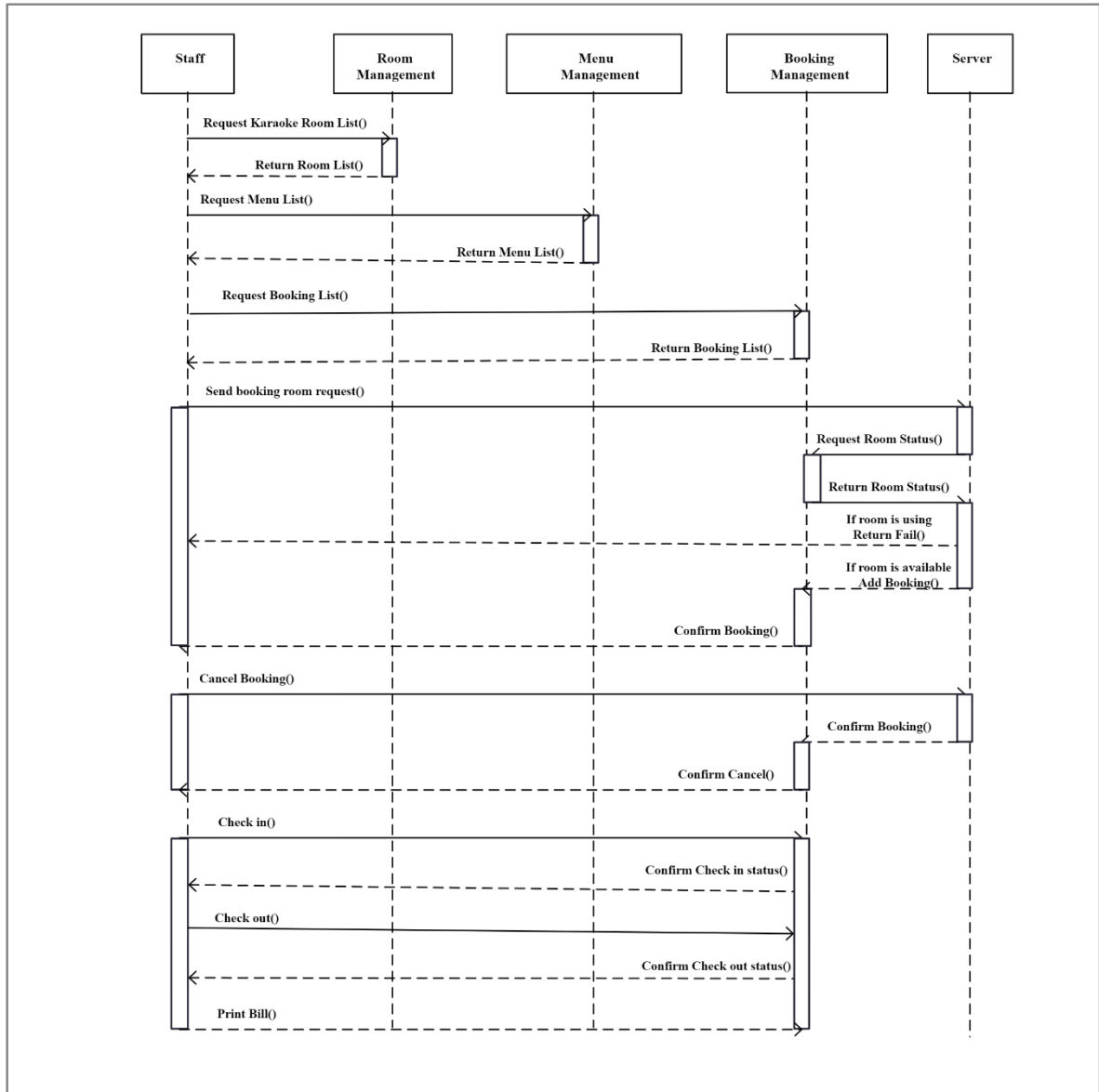
Các thao tác trên hệ thống của admin được miêu tả qua diagram sau:



**Hình 3 4:** Admin Karaoke Sequence Diagram

### III.4.2 Staff:

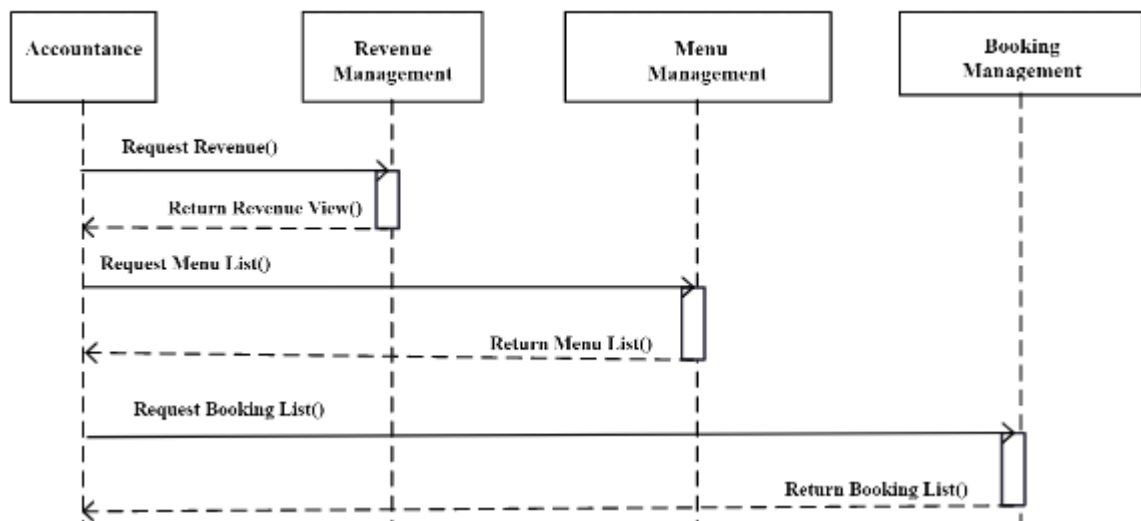
Các thao tác trên hệ thống của staff được miêu tả qua diagram sau:



**Hình 3 5:** Staff Karaoke Sequence Diagram

### III.4.2 Accountant:

Các thao tác trên hệ thống của accountant được miêu tả qua diagram sau:



**Hình 3 6:** Accountant Karaoke Sequence Diagram



## V. XÂY DỰNG HỆ THỐNG VÀ GIẢI THÍCH CODE

### V.1. Xây dựng chức năng phân quyền các tài khoản

#### V.1.1 Quá trình xây dựng:

Sử dụng phương pháp “Session” để lưu tên user cũng như role của user đó. Sau đó sử dụng các vòng if else để xét quyền truy cập cho từng role của hệ thống nếu trường hợp role không có quyền truy cập vào trang được cấp quyền thì sẽ hiển thị thông báo cảnh cáo. Các Role trong hệ thống:

- Admin (Quản lý hệ thống): Có quyền truy cập và chỉnh sửa ở tất cả các trang.
- Accountant (Kế toán): Có quyền truy cập vào trang Booking phòng để in hóa đơn phòng cho khách hàng, trang Menu để xem số lượng hàng hóa và trang Revenue để xem thống kê doanh thu theo tháng và số lượng đơn đặt của từng tháng cũng như tổng doanh thu cả năm.
- Staff (Nhân viên): Có quyền truy cập vào trang “Room management” để dễ dàng biết được phòng có đang được sử dụng hay không tiện lợi cho việc đặt phòng trực tiếp nếu khách chưa đặt phòng từ trước đó, quyền truy cập vào trang menu để xem sản phẩm đó còn hàng hay không để tiện cho việc đặt món ăn của khách hàng, quyền truy cập vào trang Booking phòng để đặt phòng-đặt giờ-đặt món ăn cho khách hàng và cũng như sửa đổi các thông tin đặt phòng nếu cần thiết.

### V.1.2 Code demo minh chứng:

```

        if(ModelState.IsValid)
        {
            var f_password = GetMD5(Password);

            var data = db.Accounts.FirstOrDefault(s => s.UserName == UserName &&
s.Password == Password);
            if (data != null)
            {
                //add Session
                Session["Name"] = data.Name;
                Session["UserName"] = data.UserName;
                Session["S_ID"] = data.S_ID;
                Session["Role"] = data.Role;
                return RedirectToAction("Admin_index");
            }
            else
            {
                ViewBag.exc = 1;
                return View();
            }
        }
        ViewBag.exc = 1;
        return View();
    
```

Sau khi kiểm tra tài khoản và mật khẩu hợp lệ thì tên, vai trò, cũng như username của nhân viên được lưu trữ vào Session.

```

<center>
    @if (Session["Role"].ToString() == "Admin")
    {
        
    }
    else if (Session["Role"].ToString() == "Staff")
    {
        
    }
    else if (Session["Role"].ToString() == "Accountant")
    {
        
    }
    else
    {
        
    }

    @if (Session["S_ID"] != null)
    {
        <h4>@Session["Name"].ToString()</h4>
    }

```

```

        <h5>@Session["Role"].ToString()</h5>
    }
    else
    {
        <h4>
            You are not logged in
        </h4>
    }
}

</center>

```

Phân code trên để hiển thị tên và quyền của nhân viên tại sidebar của trang quản lý. Mỗi role đều có một ảnh đại diện khác nhau để dễ dàng phân biệt.

```

        if (Session["S_ID"] != null && Session["Role"].ToString() != "Accountant")
        ...
        else if (Session["S_ID"] != null)
        {
            TempData["ErrorMessage"] = "You are not authorized to access this
page";
            return RedirectToAction("Admin_index", "Home");
        }
        ...

        @if (TempData["ErrorMessage"] != null)
        {
            <script type="text/javascript">
                window.onload = function () {
                    alert("@TempData["ErrorMessage"]");
                };
            </script>
        }
    }

```

Mỗi trang đều được phân quyền bằng Session, nếu Role của nhân viên hợp lệ thì được phép truy cập vào trang đó. Nếu Role của nhân viên không hợp lệ thì hệ thống sẽ trả nhân viên về trang chủ cũng như pop up lên thông báo cảnh cáo quyền truy cập cho nhân viên thấy “You are not authorized to access this page”.

## **V.2. Xây dựng chức năng đặt phòng (booking)**

### **V.2.1 Quá trình xây dựng:**

Sử dụng các chức năng truy xuất dữ liệu cơ bản như `get()` và `post()` để điền thông tin khách hàng, chọn phòng, chọn lịch đặt phòng phù hợp không bị trùng thời gian với các đơn đặt khác, đặt món ăn cũng như in hóa đơn với nguyên tắc là:

- Start Time và End Time của đơn mới đặt không được trùng hoặc nằm trong khoản thời gian của đơn đã đặt sẵn.
- Nếu tất cả các thông tin đều đầy đủ và phù hợp thì đơn đặt phòng được tạo thành công và đồng thời một giỏ hàng các món ăn được tạo cho đơn đó sau đó khách hàng có thể gọi món ăn theo menu của quán nếu món ăn đó hết hàng (Out Of Stock) thì sẽ không hiển thị để người dùng đặt hoặc nếu số lượng trong kho không đủ với số lượng mà người dùng đặt thì cũng sẽ có thông báo về đặt quá số lượng hàng trong kho.
- Phòng chỉ có thể được sử dụng (Check In) nếu thời gian đặt của phòng và thời gian thực tế trùng nhau, không được Check In sớm hơn để tránh trường hợp trùng lịch với các lịch đặt trước đó.
- Khi thanh toán phòng Server sẽ tính toán hết tất cả các chi phí sử dụng của khách hàng để in thành hóa đơn cho khách hàng.

## V.2.2 Code demo minh chứng:

### V.2.2.1 Quá trình đặt phòng:

```

try
{
    if (ModelState.IsValid)
    {
        model.Total = 0;
        model.P_Status = 1;
        model.Duration = 0;
        //Dung` de khoi tao lai Roomlist neu gap loi

        //var room = db.Rooms.Where(r => r.RoomID ==
model.RoomID).FirstOrDefault();
        //room.Status = 2;

        var booking = db.Bookings
            .Where(b => b.RoomID == model.RoomID && b.P_Status != 3 &&
                ((b.DateTime < model.DateTime && model.DateTime < b.EndTime) ||
                (b.DateTime < model.EndTime && model.EndTime < b.EndTime)))
            .FirstOrDefault();
        if (booking != null)
        {
            TempData["ErrorMessage"] = "This time has been booked. Please
choose another time";
            return View(model);
        }
        //Khoi tao gio hang` moi
        Order order = new Order();
        order.O_total = 0;
        var entity = db.Orders.Add(order);
        model.Order_ID = entity.Order_ID;

        db.Bookings.Add(model);
        db.SaveChanges();
        return RedirectToAction("Admin_index");
    }
    return View(model);
}
catch
{
    return View();
}

```

Phần code trên thể hiện nếu tất cả thông tin về khách hàng điều hợp lệ thì Server sẽ thực hiện quá trình kiểm tra thời gian của đơn đặt xem có trùng với đơn đặt nào trước đây

hay không. Nếu có thì sẽ thông báo cho nhân viên “This time has been booked. Please choose another time”.

Nếu tất cả đều phù hợp thì Server sẽ tạo một giỏ hàng trống cho đơn hàng đó cũng như set các status mặc định cho đơn hàng lúc khởi tạo, sau đó lưu đơn hàng vào hệ thống và trả về trang Booking cho nhân viên thấy kết quả.

### V.2.2.2 Quá trình đặt đồ ăn:

```

var Order = db.Orders.Where(o => o.Order_ID ==
model.Order_Detail.Order_ID).FirstOrDefault();
    var Food = db.Order_Details
        .Where(f => f.Order_ID == model.Order_Detail.Order_ID &&
f.Food_ID == model.Order_Detail.Food_ID)
        .FirstOrDefault();
    var Stock = db.Menus.Where(s => s.Food_ID ==
model.Order_Detail.Food_ID).FirstOrDefault();
    if (Stock.Stock < model.Order_Detail.Quantity)
    {
        //ViewBag.err = "The amount of food you ordered is out of stock";
        TempData["Message"] = "The amount of food you ordered is out of stock";
        return RedirectToAction("Order");
    }

    Stock.Stock -= model.Order_Detail.Quantity;
    // nếu đã có món ăn này trong giỏ hàng rồi
    if (Food != null)
    {
        // update lại tổng tiền giỏ hàng
        Order.O_total = model.Order.O_total + model.Order_Detail.Quantity *
Stock.Price;

        // update số lượng của món đó
        Food.Quantity += model.Order_Detail.Quantity;
        db.SaveChanges();
        return RedirectToAction("Order");
    }
    // nếu chưa có món ăn này khởi tạo liên kết và số lượng món đó
    Order_Detail od = new Order_Detail();
    od.Order_ID = model.Order_Detail.Order_ID;
    od.Food_ID = model.Order_Detail.Food_ID;
    od.Quantity = model.Order_Detail.Quantity;

    // update lại tổng tiền giỏ hàng
    Order.O_total = model.Order.O_total + model.Order_Detail.Quantity *
Stock.Price;

    //save món mới
    db.Order_Details.Add(od);
    db.SaveChanges();

    return RedirectToAction("Order");

```

Phần code trên mô tả các quá trình đặt món ăn:

Sau khi chọn món ăn, số lượng và gửi lên hệ thống, hệ thống sẽ kiểm tra số lượng hàng trong kho có đủ để đáp ứng nhu cầu của khách hàng hay không, nếu không đủ sẽ thông báo về cho nhân viên, nếu đủ thì sẽ trừ đi số lượng trong kho.

Sau đây tiếp tục kiểm tra xem món ăn đó đã có trong giỏ hàng hay chưa, nếu có rồi thì sẽ tăng số lượng, nếu không thì sẽ thêm vào giỏ hàng, chỉnh sửa tổng giá tiền của giỏ hàng.

Sau khi tất cả hoàn tất sẽ lưu vào hệ thống.

### V.2.2.2 Quá trình in hóa đơn:

```
public class Bill
{
    public IEnumerable<Order_Detail> Order_Details { get; set; }
    public Booking Booking { get; set; }
    public string AccountName { get; set; }
}

...

[HttpGet]
public ActionResult Bill(int id, string staff)
{
    var booking = db.Bookings.Where(b => b.PayID == id).FirstOrDefault();
    if (booking == null)
    {
        return RedirectToAction("Admin_index");
    }

    Bill bill = new Bill();
    bill.Booking = booking;
    bill.Order_Details = new Order_DetailDAO().GetListOrder(booking.Order_ID);
    bill.AccountName = staff == null ? Session["Name"].ToString() : staff;

    return View("Bill", bill);
}

[HttpPost]
public ActionResult Bill(Bill model)
{
    var booking = db.Bookings.Where(b => b.PayID ==
model.Booking.PayID).FirstOrDefault();
    if(booking == null)
    {
        return RedirectToAction("Admin_index");
    }
    booking.Duration = model.Booking.Duration;
    booking.Total = model.Booking.Total;
    booking.EndTime = model.Booking.EndTime;
    booking.P_Status = 3;
```



```

        var room = db.Rooms.Where(r => r.RoomID ==
model.Booking.Room.RoomID).FirstOrDefault();
        room.Status = 1;

        db.SaveChanges();
        return new ActionAsPdf("Bill", new { id = model.Booking.PayID, staff =
Session["Name"].ToString() });
        //return new ViewAsPdf(model);

        //{
        //    FileName = Server.MapPath("~/Content/PDF.pdf")
        //};
    }

```

Sử dụng multiple model để lấy được tất cả thông tin về đơn hàng, các món ăn trong giỏ, tên nhân viên xác nhận thanh toán in ra hóa đơn.

Dùng thư viện “Rotativa” để hỗ trợ covert hóa đơn thành pdf.

Sau khi nhân viên xác nhận hóa đơn thì tất cả thông tin sẽ được cập nhật lên hệ thống, đồng thời trạng thái của đơn hàng sẽ được chuyển thành đã thanh toán thành công.

### **V.3. Xây dựng chức năng quản lý doanh thu (Revenue)**

#### **V.3.1 Quá trình xây dựng:**

Sử dụng các biểu đồ cột, đường, tròn để thống kê doanh thu, số lượng đơn đặt, tỉ lệ các phòng được đặt theo từng tháng, và tổng doanh thu trong năm.

Xây dựng model Revenue để tổng hợp dữ liệu của quán, đưa các dữ liệu vào các list và chuyển hóa thành dạng JSON để đưa vào các template biểu đồ.

### V.3.2 Code demo minh chứng:

#### V.3.2.1 Xây dựng model Revenue:

```
public class RevenueModel
{
    public BarChart DoanhThuChart { get; set; }

    public LineChart TotalDT { get; set; }

    public PieChart TypeRoom { get; set; }
}

public class LineChart
{
    public List<string> Labels1 { get; set; } = new List<string>();
    public List<int> Data1 { get; set; } = new List<int>();
}

public class BarChart
{
    public List<string> Labels { get; set; } = new List<string>();
    public List<int> Data { get; set; } = new List<int>();

    public int max { get; set; } = new int();
}

public class PieChart
{
    public List<string> LabelsPie { get; set; } = new List<string>();
    public List<int> DataPie { get; set; } = new List<int>();
}
```

Phần code trên thể hiện model Revenue với 3 dữ liệu của 3 loại bảng: cột, đường, tròn.

Mỗi loại bảng đều có List dữ liệu:

- Labels: Dữ liệu về tiêu đề của tháng.
- Data: Dữ liệu thống kê của tháng đó về doanh thu, số lượng đơn đặt, số lượng của từng loại phòng.

### V.3.2.1 Quá trình xây dựng lớp DAO và lấy dữ liệu:

```

public class RevenueDao
{
    public QuanLyContext db = new QuanLyContext();

    public int GetTotal(int month)
    {
        return db.Bookings.OrderBy(d => d.DateTime).Where(d => d.DateTime.Month ==
month).ToList().Sum(item => item.Total);
    }

    public List<int> GetTRoom()
    {
        // chỉnh điều kiện để lọc RoomID
        var std = db.Bookings.OrderBy(i => i.RoomID).Where(r => r.Room.RoomType ==
"Standard").ToList().Count;
        var vip = db.Bookings.OrderBy(i => i.RoomID).Where(r => r.Room.RoomType ==
"Vip").ToList().Count;
        var lux = db.Bookings.OrderBy(i => i.RoomID).Where(r => r.Room.RoomType ==
"Luxury").ToList().Count;
        List<int> list = new List<int>() { std, vip, lux };
        return list;
    }

    public int GetNumGuest(int month)
    {
        return db.Bookings.OrderBy(d => d.DateTime).Where(d => d.DateTime.Month ==
month).ToList().Count;
    }

    public List<int> GetNumBook()
    {
        var list = new List<int>();
        for (int i = 1; i <= 12; i++)
        {
            list.Add(db.Bookings.OrderBy(d => d.DateTime).Where(d =>
d.DateTime.Month == i).ToList().Count);
        }
        return list;
    }

    public int GetMax()
    {
        var list = new List<int>();
        for (int i = 1; i <= 12; i++)
        {
            list.Add(db.Bookings.OrderBy(d => d.DateTime).Where(d =>
d.DateTime.Month == i).ToList().Count);
        }
        return list.Max();
    }

    public int GetTotalYear()

```

```

    {
        return db.Bookings.OrderBy(d => d.DateTime).Where(d => d.DateTime.Year ==
2020).ToList().Sum(item => item.Total);
    }
}

```

Phần code trên là phần xây dựng lớp DAO để truy cập vào database lấy dữ liệu cho các biểu đồ. Gồm các phương thức truy xuất cơ bản như OrderBy(), Where(), ToList().

```

var revenueDao = new RevenueDao();

//var labels = new List<string>() { "Jan", "Feb", "Mar", "Apr", "May",
"Jun", "Jul", "Sep", "Oct", "Nov", "Dec" };
var labels = new List<string>() { "Jan", "Feb", "Mar", "Apr", "May",
"Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec" };
var labels1 = new List<string>() { "Jan", "Feb", "Mar", "Apr", "May",
"Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec" };

var labelsPie = new List<string>() { "Standard", "VIP", "Luxury" };
//test numguest

//var data = new List<int>() { 500, 320, 132, 346, 567, 621, 712, 811,
818, 902, 985 };

var data = new List<int>();
var data1 = new List<int>();

var dataPie = new List<int>();

for (int x = 0; x < 12; x++)
{
    data1.Add(revenueDao.GetTotal(x + 1));
}

var doanhthu = new BarChart()
{
    Labels = labels,

    Data = revenueDao.GetNumBook(),
    max = revenueDao.GetMax(),

};

var totaldt = new LineChart()
{
    Labels1 = labels1,
    Data1 = data1
};

```

```

        var typeRoom = new PieChart()
        {
            LabelsPie = labelsPie,
            DataPie = revenueDao.GetTRoom(),

        };

        ViewBag.TotalYear = revenueDao.GetTotalYear();
        var model = new RevenueModel()
        {
            DoanhThuChart = doanhthu,
            TotalDT = totaldt,
            TypeRoom = typeRoom
        };

        return View(model);
    }

```

Phân code trên thể hiện quá trình đổ dữ liệu vào các tiêu đề của các bảng như tên các tháng, list các loại phòng.

Đồng thời gọi các phương thức của hàm DAO để lấy dữ liệu của từng tháng, từng loại phòng cho các loại biểu đồ.

### V.3.2.2 Quá trình xây dựng lớp DAO và lấy dữ liệu:

```

<script type="text/javascript">
    var dataJson = @Html.Raw(Newtonsoft.Json.JsonConvert.SerializeObject(Model));
    console.log(dataJson); //mô console len de xem du lieu sau khi convert
    $(document).ready(function () {
        console.log(dataJson.DoanhThuChart.Data);
        //truyen du lieu vao` function(trc tien phai tao. 1 funciton va` them du
        lieu dau` vao`)
        setDataForBarChart(dataJson.DoanhThuChart.Labels,
        dataJson.DoanhThuChart.Data, dataJson.DoanhThuChart.max);
    });

    $(document).ready(function () {
        console.log(dataJson.TotalDT.Data);
        setDataForLineChart(dataJson.TotalDT.Labels1, dataJson.TotalDT.Data1);
    });

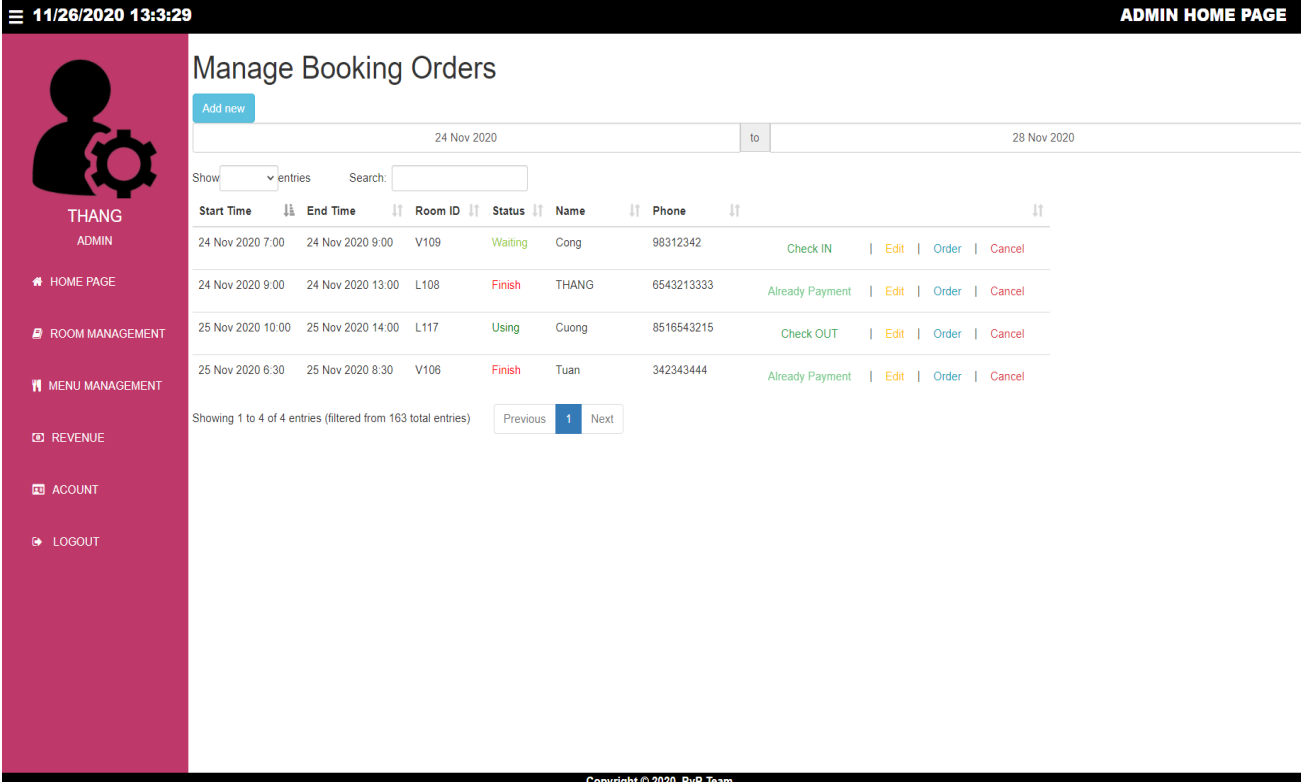
    $(document).ready(function () {
        console.log(dataJson.TypeRoom.Data);
        setDataForPieChart(dataJson.TypeRoom.LabelsPie, dataJson.TypeRoom.DataPie);
    });
</script>

```

Quá trình convert dữ liệu của model thành dạng JSON sau đó thêm vào các template biểu đồ.

## VI. KẾT QUẢ ĐẠT ĐƯỢC

### VI.1. Home Page:



The screenshot shows the Admin Home Page with the following components:

- Header:** Date and time: 11/26/2020 13:3:29. Page title: ADMIN HOME PAGE.
- Sidebar:**
  - THANG ADMIN (with user icon)
  - HOME PAGE
  - ROOM MANAGEMENT
  - MENU MANAGEMENT
  - REVENUE
  - ACCOUNT
  - LOGOUT
- Main Content Area:**
  - Manage Booking Orders**
    - [Add new](#)
    - Date range: 24 Nov 2020 to 28 Nov 2020
    - Show  entries. Search:
  - Table of Booking Orders:**

Start Time	End Time	Room ID	Status	Name	Phone	
24 Nov 2020 7:00	24 Nov 2020 9:00	V109	Waiting	Cong	98312342	Check IN   Edit   Order   Cancel
24 Nov 2020 9:00	24 Nov 2020 13:00	L108	Finish	THANG	6543213333	Already Payment   Edit   Order   Cancel
25 Nov 2020 10:00	25 Nov 2020 14:00	L117	Using	Cuong	8516543215	Check OUT   Edit   Order   Cancel
25 Nov 2020 6:30	25 Nov 2020 8:30	V106	Finish	Tuan	342343444	Already Payment   Edit   Order   Cancel
  - Showing 1 to 4 of 4 entries (filtered from 163 total entries) | Previous | 1 | Next

Hình 6 1: Home Page

Hiển thị danh sách của các phòng hiện tại đã được đặt bởi khách hàng. Bao gồm:

**Start time & End time:** Thời gian mà khách hàng dự muốn bắt đầu và dự kiến kết thúc việc sử dụng phòng.

**Room ID:** ID của phòng đang được đặt hoặc đang được sử dụng bởi khách hàng.

**Status:** tình trạng hiện tại của phòng (có 3 loại trạng thái phòng được cài đặt là:

- **Waiting** – Chưa được khách hàng sử dụng
- **Using** – Đang được khách hàng sử dụng
- **Finish** – Đã được khách hàng kết thúc thanh toán).



**Name:** Tên khách hàng đặt phòng.

**Phone:** Số điện thoại của khách hàng đặt phòng.

**Các nút chức năng:**

**Đặt trạng thái phòng:** Có 3 trạng thái tương đương với 3 trạng thái của **Status**. Khi **Status** phòng đang là **Waiting** thì nút sẽ trong trạng thái **Check-in** và khi được bấm vào thì nút sẽ chuyển sang trạng thái **Check-out** và **Status** của phòng cũng sẽ được thay đổi thành **Using**. Cuối cùng khi kết thúc việc sử dụng phòng thì nút **Check-out** sẽ được bấm và sẽ đưa nhân viên đến một page hóa đơn để có thể in ra hóa đơn, đồng thời **Status** của phòng sẽ chuyển sang trạng thái **Finish**.

- **Edit:** Chỉnh sửa tất cả thông tin của phòng.
- **Order:** Ghi lại thông tin các món ăn, nước uống mà khách hàng đã đặt trong thời gian sử dụng phòng đồng thời cũng sẽ được thêm vào trong hóa đơn khi thanh toán.
- **Cancel:** Hủy phòng trong ra khỏi danh sách nếu khách hàng không muốn sử dụng phòng.

**Thanh chọn thời gian:** Để hiển thị danh sách tất cả các phòng trong khoảng thời gian được chọn tránh trường hợp hiển thị những phòng không cần thiết.

**Thanh tìm kiếm:** Để tìm kiếm thông tin cụ thể của 1 phòng bao gồm **Status**, **Room ID**, **Name**, **Phone** trong khoảng thời gian hiển thị của thanh chọn thời gian.


**Drop list:** Để giới hạn số lượng phòng hiển thị trong khoảng thời gian hiển thị của thanh chọn thời gian

**Nút bấm ADD NEW:** Nơi thêm mới đơn đặt.

## VI.2. Order:

11/29/2020 19:46:13

ORDER



THANG

ADMIN

[HOME PAGE](#)
[ROOM MANAGEMENT](#)
[MENU MANAGEMENT](#)
[REVENUE](#)
[ACCOUNT](#)
[LOGOUT](#)

Food Name	Quantity	Price	Total
Heineken	24	40000	960000 VND
Khô gà	2	80000	160000 VND
Khô bò	2	80000	160000 VND
Yaourt trái cây	5	80000	400000 VND
Combo trái cây 3	4	380000	1520000 VND
Chivas 18	2	2500000	5000000 VND
Hạt dẻ	2	50000	100000 VND
Nước suối	10	20000	200000 VND
-	-	Total All	8500000 VND

Add to order list

-----Select Food-----

0

Add

Copyright © 2020, PvP Team

**Hình 6 2:** Trang Order món ăn

Hiện thị danh sách các món ăn mà khách hàng đã đặt trong quá trình sử dụng phòng bao gồm:

**Food Name:** Tên món ăn, đồ uống mà khách hàng đã đặt.

**Quantity:** Số lượng của món ăn, đồ uống mà khách hàng đã đặt.

**Price:** Giá tiền của món ăn, đồ uống dựa trên đơn vị là 1.

**Total:** Tổng giá của món ăn đó khi được tính cùng với số lượng đã được đặt bởi khách hàng (**Quantity x Price**).


**Total All:** Tổng giá tiền của tất cả các món ăn, đồ uống mà khách hàng đã đặt trong khi sử dụng phòng.

**Drop list:** Hiển thị tất cả các món ăn có trong menu của quán (**Menu Management**).

**Input box:** Nhập số lượng món ăn mà khách hàng đã đặt – số nhập vừa dựa trên **Stock** còn lại trong **Menu Management** nếu vượt quá stock còn lại sẽ được thông báo lỗi.

**Button ADD:** Thêm món ăn khách hàng đã đặt vào trong hóa đơn.

### VI.3. Hóa đơn:



## BILLS PAY

**Staff:** Thang

**Customer:** Tuan      **Phone:** 342343444

**Time in:** 06:30 AM      **Time out:** 13:01 PM

**Duration:** 6.52

**Room:** V106 - **Vip:** 150,000 đ

**Date:** Wednesday, November 25, 2020

**Your order:**

Tôm kim sa	60,000 x 4
Heineken	40,000 x 24
Nước suối	20,000 x 5
Combo trái cây 3	380,000 x 1
Cơm chiên dương châu	85,000 x 2

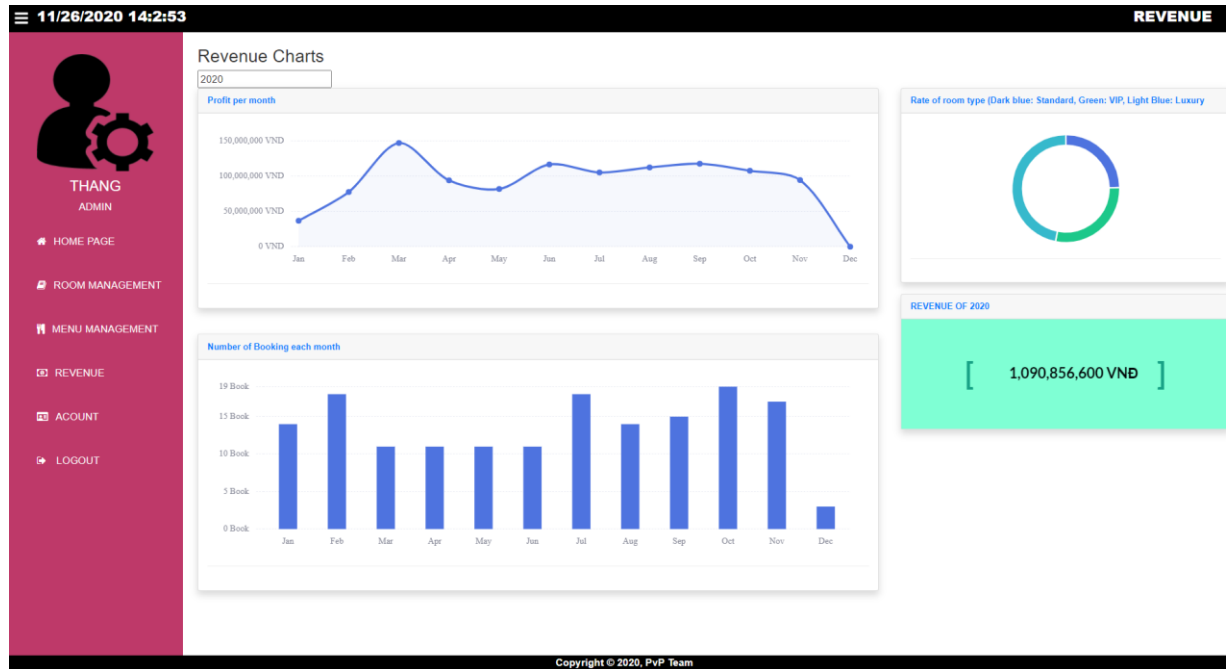
<b>Total Services</b>	<b>1,850,000 đ</b>
<b>Total Hourly</b>	<b>978,000 đ</b>
<b>Tax 10%</b>	<b>282,800 đ</b>
<b>Total</b>	<b>3,110,800 đ</b>

**Hình 6 3:** Hóa đơn

Hiện thị thông tin phòng, khách hàng và danh sách các món mà khách hàng đã sử dụng trong quá trình sử dụng phòng cùng với tổng giá tiền của từng loại và tổng chi phí khách hàng phải trả.

- **Staff:** Tên nhân viên thanh toán được lấy từ thông tin tài khoản đã đăng nhập từ lúc nút bấm **Check-out** được bấm.
- **Customer:** Tên của khách hàng đã đặt phòng được lấy từ thông tin phòng.
- **Phone:** Số điện thoại của khách hàng đã đặt phòng được lấy từ thông tin phòng.
- **Time in:** Thời gian mà phòng bắt đầu được sử dụng được lấy theo thông tin phòng.
- **Time out:** Thời gian mà phòng được kết thúc sử dụng và được lấy theo thời gian hiện thị của máy tính.
- **Duration:** Thời gian mà phòng được sử dụng trong bao lâu.
- **Room:** Mã của phòng được sử dụng, loại phòng đồng thời là giá tiền của loại phòng đó trong một giờ sử dụng.
- **Date:** Ngày, tháng, năm của hóa đơn được thanh toán.
- **Your order:** Danh sách các món ăn, nước uống và số lượng đã được đặt bởi khách hàng trong quá trình sử dụng phòng.
- **Total Services:** Tổng giá tiền của các món ăn dựa trên các món ăn, nước uống đã được đặt ở **Your order**.
- **Total Hourly:** Giá tiền phòng được quy đổi từ **Duration** (giá tiền 1h nhân với thời gian sử dụng).
- **Tax (10%):** Khoản phí phụ thu VAT là giá trị 10% của tổng tiền **Total Services** và **Total Hourly**.
- **Total:** Tổng giá tiền của **Total Services**, **Total Hourly** và **Tax**.

## VI.4. Biểu đồ:



**Hình 6 4: Biểu đồ doanh thu**

- **Biểu đồ đường:** Thể hiện doanh thu hàng tháng của quán trải dài từ tháng 1 cho đến tháng 12 trong một năm dựa trên **Total** của hóa đơn mà khi khách hàng đã hoàn thành thanh toán.
- **Biểu đồ cột:** Thể hiện số lượng khách hàng đã đặt phòng bắt đầu từ tháng 1 cho đến tháng 12 trong năm dựa trên số lượng phòng đã được đặt hàng tháng.
- **Biểu đồ tròn:** Thể hiện tỉ lệ các loại phòng được đặt bởi khách hàng dựa trên loại phòng mà khách hàng đã đặt.
- Hiện thị tổng doanh thu của quán trong 1 năm.

## VII. KẾT LUẬN

### VII.1. Ưu, khuyết điểm của đề tài

#### Ưu điểm:

- Có cái nhìn rõ hơn về cách vận hành của một trang web quản lí và cụ thể là quản lí karaoke.
- Biết được bố cục của một trang web như thế nào.
- Thành thạo hơn trong quản lí và làm việc với cơ sở dữ liệu.
- Hiểu rõ hơn về cách hoạt động của mô hình MVC.

#### Khuyết điểm:

- Khó khăn trong việc tìm kiếm tài liệu tham khảo, cụ thể là trang quản lí đã được đưa lên sử dụng.
- Có nhiều kiến thức mới nên sẽ bị mất khoảng thời gian đầu để làm quen.

### VII.2. Những gì đã làm được:

- Lọc được loại account khi đăng nhập vào trong web.
- Hiện thị được các tính năng liên quan đến thời gian.
- Hiện thị được trạng thái và thông tin của từng phòng.
- Tương tác qua lại giữa các table trong cơ sở dữ liệu.
- Thống kê doanh thu theo tháng qua biểu đồ.
- Tính toán theo thời gian thực.
- Nhiều lựa chọn trong quá trình đặt phòng.
- Thanh toán và order món ăn.
- In được hóa đơn thành file pdf.
- Có thanh filter thời gian để dễ dàng quản lý.

**VII.3. Những gì chưa làm được:**

- Thống kê doanh thu theo năm qua biểu đồ.
- Giao diện còn đơn giản.
- Chưa có nhiều tính năng ngoài những tính năng tối thiểu.
- Không có nhiều hình ảnh.

**4. Hướng phát triển cho đề tài:**

- Mở rộng hệ thống cho người dùng ngoài những nhân viên của quán.
- Có tính năng đặt online để khách hàng có thể tự chủ động đặt phòng (cần phải đăng nhập).
- Hệ thống order chủ động dành cho khách hàng (chỉ áp dụng cho loại tài khoản là khách hàng)
- Tính năng khuyến mãi.



## VIII. TÀI LIỆU THAM KHẢO

1. <https://www.chartjs.org/docs/latest/>
2. <https://www.howkteam.vn/learn/lap-trinh/lap-trinh-c-net-7-5>
3. <https://www.w3schools.com/>
4. <https://weblog.west-wind.com/posts/2007/Sep/24/ASPNET-and-Styles-CSS-Embedding>
5. <https://stackoverflow.com/>
6. <https://www.codemag.com/>
7. <https://dotnet.microsoft.com/apps/aspnet/mvc>
8. <https://viblo.asia/newest>
9. <https://xuanthulab.net/asp-net-core-mvc-tao-ung-dung-mvc-dau-tien-c-csharp.html>
10. <https://www.c-sharpcorner.com/article/asp-net-mvc5-rotativa-easy-way-to-create-pdf-and-image-files/>