**BỘ THÔNG TIN VÀ TRUYỀN THÔNG**

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**------------------------------**

****

**BÁO CÁO**

**THỰC TẬP TỐT NGHIỆP**

*Đề tài:*

**XÂY DỰNG WEBSITE QUẢN LÝ KHÁCH SẠN**

Người hướng dẫn : ThS. NGUYỄN NGỌC DUY

Sinh viên thực hiện : TRẦN THỊ LOAN

Mã số sinh viên : N13DCCN105

Lớp : D13CQIS01\_N

Khoá : 2013 -2018

Hệ : ĐẠI HỌC CHÍNH QUY

TP.HCM, tháng 08/2018

**LỜI CẢM ƠN**

Được sự phân công của khoa Công Nghệ Thông Tin II Trường Học Viện Công Nghệ Bưu Chính Viễn Thông và sự đồng ý của Thầy giáo hướng dẫn **ThS. NGUYỄN NGỌC DUY** em đã thực hiện đề tài “**Xây dựng Website Quản lý Khách Sạn**”

Để hoàn thành đề tài Thực tập tốt nghiệp này. Em xin cảm ơn các thầy cô giáo đã tận tình hướng dẫn, giảng dạy trong suốt quá trình học tập, nghiên cứu và rèn luyện tại Trường Học Viện Công Nghệ Bưu Chính Viễn Thông cơ sở Thành Phố Hồ Chí Minh.

Xin chân thành cảm ơn Thầy giáo hướng dẫn **ThS. NGUYỄN NGỌC DUY,** người đã tận tình, chu đáo giúp em thực hiện và hoàn thành đề tài tốt nghiệp này.

Mặc dù có nhiều cố gắng để thực hiện đề tài Thực tập một cách hoàn chỉnh nhất. Song do còn thiếu nhiều kinh nghiệm và hạn chế nhất định về kiến thức nên không tránh khỏi những thiếu sót mà bản thân chưa thấy được. Em rất mong nhận được các ý kiến đóng góp của quý Thầy, Cô giáo để đề tài được hoàn chỉnh hơn.

Em xin chân thành cảm ơn!

TP. Hồ Chí Minh, ngày 05 tháng 08 năm 2018

**Sinh viên thực hiện**

TRẦN THỊ LOAN

MỤC LỤC

[CHƯƠNG 1: GIỚI THIỆU VỀ ĐỀ TÀI ………………………………………1](#_Toc501444019)

[1.1 Mục đích của đề tài……………………………………………………..3](#_Toc501444023)

1.2 Khảo sát…………………………..................

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT, CÔNG NGHỆ VÀ PHẦN MỀM DÙNG ĐỂ PHÁT TRIỂN HỆ THỐNG](#_Toc501444026) ………………………………………………

2.1 Tìm hiểu về .Net framework và Ngôn ngữ lập trình C#

2.1.1 Tìm hiểu về .Net framework

2.1.2 Ngôn ngữ lập trình C#

[2.2 Tổng quan về mô hình MVC 6](#_Toc501444027)

2.2.1 Nền tảng của mô hình MVC

[2.2.2 Lợi ích của mô hình MVC 6](#_Toc501444028)

[2.2.3 Các tính năng của mô hình MVC 8](#_Toc501444029)

[2.2.4 Quan Hệ Giữa Các Đối Tượng Trong HiberAME=nate 9](#_Toc501444030)

[2.3 TỔNG QUAN VỀ ENTITY FRAMEWORK 11](#_Toc501444031)

[2.3.1 Định nghĩa về Entity Framework………………………………](#_Toc501444032)

**[2.3.2 Vị trí của](#_Toc501444033)**[[Entity Framework](#_Toc501444033)](https://www.dammio.com/glossary/entity-framework)**[trong mô hình lập trình và phát triển Web](#_Toc501444033)** [2.3.3 **Mô hình kiến trúc**[Entity Framework](http://www.dammio.com/glossary/entity-framework)……………………………. 11](#_Toc501444034)

[2.4 TÌM HIỂU NGÔN NGỮ SQL 15](#_Toc501444043)

[2.4.1 SQL là ngôn ngữ của cơ sở dữ liệu. 15](#_Toc501444044)

[2.4.2 Vai trò của SQL 15](#_Toc501444045)

[2.4.3 Các bảng cơ sở dữ liệu 16](#_Toc501444046)

[2.4.4 Các truy vấn SQL 17](#_Toc501444047)

[2.4.5 Thao tác dữ liệu SQL 17](#_Toc501444048)

[2.4.6 Định nghĩa dữ liệu SQL 17](#_Toc501444049)

[2.4.7 SQL và ASP 17](#_Toc501444050)

[CHƯƠNG 3: XÂY DỰNG CHƯƠNG TRÌNH. 18](#_Toc501444051)

3.1 PHÂN TÍCH NGHIỆP VỤ QUẢN LÝ KHÁCH SẠN

[3.1.1 Hoạt động, cơ cấu tổ chức, chức năng, nhiệm vụ của từng bộ phận : 18](#_Toc501444052)

[3.1.2 Nghiên cứu, đánh giá hệ thống thông tin quản lý của khách sạn. 18](#_Toc501444053)

3.1.3 [Phân tích hệ thống thông tin quản lý của khách sạn. 18](#_Toc501444054)

[3.2 : PHÂN TÍCH THIẾT KẾ HỆ THỐNG QUẢN LÝ KHÁCH SẠN 23](#_Toc501444055)

[3.2.1 Biểu đồ ngữ cảnh của hệ thống 23](#_Toc501444056)

[3.2.2 Lập sơ đồ chức năng kinh doanh – BFD 24](#_Toc501444057)

[3.2.3 Biểu đồ luồng dữ liệu của hệ thống khách sạn : 25](#_Toc501444058)

3.2.4 [DFD mức 0 25](#_Toc501444059)

[3.2.5 Biểu đồ luồng dữ liệu chức năng quản lý đặt phòng. 27](#_Toc501444060)

[3.2.6 Biểu đồ luồng dữ liệu mức 1 chức năng quản lý khách vào (Check in) 27](#_Toc501444061)

3.2.7 [Biểu đồ luồng dữ liệu mức 1 chức năng khách sử dụng dịch vụ 28](#_Toc501444062)

[3.2.8 Biều đồ luồng dữ liệu mức 1 chức năng quản lý khách trả phòng check out: 29](#_Toc501444063)

[3.2.9 Biểu đồ luồng dữ liệu phân rã chức năng mức 1 chức năng quản lý tài nguyên : 29](#_Toc501444064)

[3.2.10 Mô hình thực thể kết hợp ERD 32](#_Toc501444066)

[3.2.11 Mô hình Diagram 33](#_Toc501444067)

3.3 : THIẾT KẾ GIAO DIỆN

[CHƯƠNG 4: KẾT QUẢ THỰC HIỆN 81](#_Toc501444134)

[4.1 Quy trình chạy chương trình web Quản lý Khách sạn 81](#_Toc501444135)

CHƯƠNG 5 : KẾT LUẬN [81](#_Toc501444136)

[5.1 Kết quả đạt được: 81](#_Toc501444137)

[5.2 Hướng phát triển : 82](#_Toc501444138)

[TÀI LIỆU THAM KHẢO 72](#_Toc501444139)

**DANH MỤC HÌNH**

**MỞ ĐẦU**

Ngành du lịch là một ngành kinh doanh có từ rất lâu đời trên thế giới. Tại các nước phát triển, ngành du lịch phát triển hết sức mạnh mẽ và có lợi nhuận cao với sự hỗ trợ đắc lực của công nghệ thông tin trong tổ chức và hoạt động. Tại nước ta, những năm gần đây, ngành du lịch mới thực sự phát triển về quy mô cũng như ảnh hưởng của nó trong các lĩnh vực kinh tế xã hội ngày càng được mở rộng. Việt Nam là một nước có rất nhiều các danh lam thắng cảnh, các di tích lịch sử, và 1 trong 7 kỳ quan của thế giới (Vịnh Hạ Long), vì vậy hằng năm lượng khách du lịch tới Việt Nam rất đông, cả khách trong nước và ngoài nước, điều này rất thuận lợi cho các dịch vụ nhà ở, khách sạn phát triển. Theo đó, nó đòi hỏi các nhà quản lý khách sạn phải xây dựng cho hệ thống khách sạn của mình hệ thống thông tin phù hợp với tình hình thực tế của khách sạn để đảm bảo cho nhu cầu ngày càng cao của khách hàng về sự nhanh chóng, thuận tiện trong những lần đặt phòng, thuê phòng tại khách sạn. Không những thế hệ thống thông tin quản lý khách sạn sẽ giúp cho doanh nghiệp quản lý được tốt hơn về mọi mặt như về : quản lý phòng, quản lý hiện trạng phòng, tiện nghi có ở trong phòng. Các dich vụ ăn uống, giặt là của khách sạn.

Vì những nhu cầu về phát triển du lịch trên, việc tạo nên 1 trang web để Quản lý Khách sạn, vừa đáp ứng nhu cầu khách hàng, vừa giúp Khách sạn có thể thuận tiện trong việc kinh doanh dịch vụ nghỉ ngơi này. Và em hi vọng, với việc hoàn thiện Đề tài này, nếu có thể ứng dụng vào thực tiễn và hoạt động mạnh mẽ trên các website tìm kiếm của những khách hàng có nhu cầu. Qúa trình làm đề tài của em còn nhiều thiếu sót và chưa hoàn thiện. Em kính mong quý Thầy cô xem xét, đóng góp ý kiến để em được hoàn thiện tốt hơn. Em xin chân thành cảm ơn !

**CHƯƠNG 1: GIỚI THIỆU VỀ ĐỀ TÀI**

* 1. **Mục đích của đề tài Quản lý Khách sạn**

Xây dựng các modul quản lý như:

* Quản lý nhân viên
* Ghi nhận thông tin khách hàng thuê phòng
* Ghi nhận đăng ký đặt phòng
* Ghi nhận thuê phòng (check in)
* Tra cứu phòng qua sơ đồ phòng
* Ghi nhận việc sử dụng dịch vụ (minibar, giặt ủi, điện thoại, thức ăn, các dịch vụ khác)
* Tính tiền khi trả phòng – check out ( lập hóa đơn )
* Cập nhật giá phòng, giá dịch vụ
  1. **Khảo sát**
* Nhân viên: mỗi nhân viên thuộc một bộ phận riêng (lễ tân, buồng, nhà hàng) phụ trách công việc nhất định. Một bộ phận có thể có nhiều nhân viên.
* Quy định đăng ký đặt phòng: khách sạn có nhiều loại phòng (Standard, Super, VIP…), kiểu phòng (1 giường đôi, 2 giường chiếc, 1 giường đôi và 1 giường chiếc…) ; Kết hợp loại và kiểu phòng ta có hạng phòng ; mỗi phòng sẽ thuộc 1 hạng phòng. Mỗi hạng phòng sẽ có giá nhất định từ 1 thời điểm xác định ; Chỉ nhận giữ chỗ nếu vào thời gian đó phòng còn trống và chưa được đặt chỗ trước. Phiếu đặt phòng ghi nhận thông tin khách đặt, khách thuê, số lượng phòng thuộc 1 hạng phòng được đặt, ngày bắt đầu thuê, số ngày thuê dự kiến ...
* Một phòng có thể có các trạng thái sau : sẵn sàng, có khách, dơ, đặt trước, bảo trì.
* Quy định thuê phòng: Có 2 loại phiếu thuê: phiếu thuê trên 1 phiếu đặt trước, hoặc phiếu thuê cho khách vãng lai (không đặt trước) ; Một phiếu thuê phòng quy định ngày đến, giờ đến, số phòng và thông tin các khách ở trong phòng đó. Lưu ý 1 phòng trong phiếu thuê có thể check in trễ hoặc check out sớm. Ta có thể giảm giá phòng cho nguyên đoàn hoặc 1 số phòng trong đoàn.
* Quy định sử dụng dịch vụ: khách sạn có phục vụ thêm 1 số lọai dịch vụ (ăn uống, giặt là, điện thoại, gửi fax, ... ) với đơn giá riêng cho từng loại. Riêng giặt ủi có 3 giá : giá giặt, giá ủi, giá hấp.
* Quy định tính tiền thuê phòng: mỗi hạng phòng có đơn giá riêng, giá phòng không có giá giờ, không phân biệt ngày lễ ngày thường. Giá phòng có thể giảm do khuyến mãi trong 1 khoảng thời gian.
* Thông tin khách hàng : ghi nhận thông tin liên lạc của khách hàng khi lần đầu tiên khách hàng đặt phòng hoặc thuê phòng. Nếu những thông tin của khách hàng có thay đổi, sẽ cập nhật thông tin khi khách thuê phòng.
* Người sử dụng có thể thay đổi các quy định sau: thay đổi số lượng khách tối đa, đơn giá các loại phòng, tiện nghi của phòng, danh mục dịch vụ, đơn giá các loại dịch vụ.
* Người quản trị hệ thống có thể thực hiện thao tác thêm, xóa, phân quyền theo từng bộ phận cho nhân viên khi đăng nhập vào hệ thống.

**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT, CÔNG NGHỆ VÀ PHẦN MỀM DÙNG PHÁT TRIỂN HỆ THỐNG**

**2.1 TÌM HIỂU VỀ .NET FRAMEWORK VÀ NGÔN NGỮ**

**LẬP TRÌNH C#**

**2.1.1 Tìm hiểu về .Net framework**

* **Nền tảng của .Net framework**

|  |
| --- |
| **.NET Framework** là cơ sở hạ tầng cho việc xây dựng, triển khai và chạy các ứng dụng và dịch vụ Web. Nó cung cấp một môi trường đa ngôn ngữ, dựa trên nền các chuẩn với hiệu nǎng cao, cho phép tích hợp những đầu tư ban đầu với các ứng dụng và dịch vụ thế hệ kế tiếp và giải quyết những thách thức của việc triển khai và vận hành các ứng dụng trên quy mô Internet                                                                                                                        Thực tế **.NET Framework** là kết quả của hai dự án lớn của Microsoft:   * Dự án đầu tiên nhằm mục đích cải tiến công việc phát triển các ứng dụng trên nền Windows, hãy nhìn cụ thể vào công việc cải tiến COM (Component Object Model - Mô hình Đối tượng Thành phần) của Microsoft. * Dự án thứ hai tập trung tạo ra nền tảng cho việc chuyển giao phần mềm như một dịch vụ. Sản phẩm này đã cải thiện đáng kể nǎng suất cho người phát triển, đơn giản hóa công việc triển khai và thực thi ứng dụng một cách đáng tin cậy. * **Mục đích thiết kế .NET Framework** |
| [https://sites.google.com/site/niitdotnet2011/_/rsrc/1468736994367/-net-framework/nhung-diem-dac-trung-cua-net-framework/DotNetFramework.JPG](https://sites.google.com/site/niitdotnet2011/-net-framework/nhung-diem-dac-trung-cua-net-framework/DotNetFramework.JPG?attredirects=0)  Hình 1: sơ đồ thiết kế của mô hình .Net framework  **.NET Framework** là thành quả tối ưu của sự kết hợp công sức và trí tuệ của Microsoft, nhằm tạo ra một nền tảng cho việc xây dựng và triển khai nhanh chóng các dịch vụ và ứng dụng Web XML. Tầm nhìn của nền tảng **.NET Framework** kết hợp một mô hình lập trình đơn giản, dễ sử dụng với các giao thức mở và biến đổi được của Internet. Để đạt được tầm nhìn này, việc thiết kế **.NET Framework** nhằm một số mục đích:   * **Sự hợp nhất thông qua các chuẩn Internet công cộng:** Để giao tiếp với những đối tác kinh doanh, những khách hàng phụ thuộc vào các khu vực theo vị trí địa lý, thậm trí cả những ứng dụng cho tương lai, những giải pháp phát triển cần được đề nghị hỗ trợ cho các chuẩn Internet mở và tích hợp chặt chẽ với các giao thức mà không bắt buộc người phát triển phải thông hiểu cơ sở hạ tầng bên dưới nó. * **Khả nǎng biến đổi được thông qua một kiến trúc "ghép nối lỏng":** Đa số các hệ thống lớn, biến đổi được trên thế giới được xây dựng trên những kiến trúc không đồng bộ dựa trên nền thông điệp (message-based). Nhưng công việc xây dựng các ứng dụng trên một kiến trúc như vậy thường phức tạp và có ít các công cụ hơn so với những môi trường phát triển ứng dụng N lớp (N-tier) "ghép nối chặt". **.NET Framework** được xây dựng để đem lại những lợi thế về nǎng suất của kiến trúc "ghép nối chặt" với khả nǎng biến đổi được và vận hành với nhau của kiến trúc "ghép nối lỏng". * **Hỗ trợ nhiều ngôn ngữ:**Các nhà phát triển sử dụng những ngôn ngữ khác nhau do mỗi ngôn ngữ riêng có những ưu thế đặc thù : một số ngôn ngữ đặc biệt thích hợp với thao tác toán học; một số khác lại đa dạng ở các hàm tính toán tài chính v.v. **.NET Framework** cho phép các ứng dụng được viết trong nhiều ngôn ngữ lập trình khác nhau và chúng có khả nǎng tích hợp với nhau một cách chặt chẽ. Ngoài ra, với **.NET Framework**, các công ty còn có thể tận dụng những lợi thế của kỹ nǎng phát triển sẵn có mà không cần phải đào tạo lại và cho phép những người phát triển sử dụng ngôn ngữ mà họ ưa thích. * **Nâng cao nǎng suất cho các nhà phát triển:**Với số lượng các nhà phát triển ứng dụng không nhiều nên mỗi giờ làm việc họ phải cho ra kết quả công việc cụ thể. Các nhóm phát triển với **.NET Framework** có thể loại bỏ những công việc lập trình không cần thiết và tập trung vào viết các lôgic doanh nghiệp. Chẳng hạn như .NET Framework có ưu điểm tiết kiệm thời gian như thực hiện các giao dịch tự động và dễ sử dụng, quản lý bộ nhớ một cách tự động và có chứa một tập các đối tượng điều khiển đa dạng bao hàm nhiều tác vụ phát triển chung. * **Bảo vệ những sự đầu tư thông qua việc bảo mật đã được cải tiến:** Một trong những vấn đề liên quan lớn nhất đến Internet hiện nay là bảo mật. Kiến trúc bảo mật của **.NET Framework** được thiết kế từ dưới lên để đảm bảo các ứng dụng và dữ liệu được bảo vệ thông qua một mô hình bảo mật dựa-trên-bằng-chứng (evidence-based) và tinh vi. * **Tận dụng những dịch vụ của hệ điều hành:**Windows cung cấp một số lượng đa dạng các dịch vụ có sẵn với bất kỳ nền tảng nào; như truy cập dữ liệu một cách toàn diện, bảo mật tích hợp, các giao diện người dùng tương tác, mô hình đối tượng thành phần đáng tin cậy và các giám sát quá trình giao dịch. **.NET Framework** đã tận dụng lợi thế đa dạng và phong phú này để đưa ra cho mọi người theo cách dễ sử dụng nhất. * **Những đặc tính của .NET Framework** * **Hỗ trợ các chuẩn dịch vụ Web XML**   Sự sử dụng ngôn ngữ đánh dấu mở rộng XML ở khắp mọi nơi: XML là một định dạng dữ liệu cho việc trao đổi tài liệu lẫn nhau có cấu trúc trên Web. **.NET Framework** sử dụng XML ở khắp mọi nơi từ việc mô tả các đối tượng cho đến bảo mật các tập tin cấu hình.   * **Khả nǎng vận hành với nhau qua việc hỗ trợ SOAP: .NET Framework** cho phép những người phát triển đưa ra và sử dụng các dịch vụ Web XML một cách trong suốt (transparently) thông qua SOAP (Simple Object Access Protocol), một vǎn phạm XML chuẩn tạo khả nǎng thao tác giữa các phần trong ứng dụng. * **Mô tả dễ dàng các dịch vụ Web XML với WSDL:** **.NET Framework** tạo ra những mô tả WSDL (Web Services Description Language - Ngôn ngữ Mô tả các Dịch vụ Web) cho các dịch vụ Web XML. * **Đưa ra những dịch vụ Web với SOAP Discovery (viết tắt Disco):**SOAP Discovery cung cấp cơ chế bởi những dịch vụ Web XML nào được tìm thấy trên một máy phục vụ Web (Web server) ; **.NET Framework** cung cấp một cách thức để công bố các dịch vụ Web XML thông qua SOAP Discovery. * **Hỗ trợ UDDI: .NET Framework** sử dụng UDDI (Universal Description, Discovery and Integration) cho phép những dịch vụ Web XML cung cấp cơ chế bởi những dịch vụ Web XML nào được tìm thấy trên Internet. * **Hiệu suất cho người phát triển** * **Tích hợp nhiều ngôn ngữ lập trình**:  **.NET Framework** cung cấp khả nǎng tích hợp ngôn ngữ lập-trình-chéo (cross-programming language), chặt chẽ, thúc đẩy nǎng suất do sự cho phép người phát triển mở rộng những thành phần một ngôn ngữ lập trình bên trong một ngôn ngữ khác theo cách kế thừa ngôn-ngữ-chéo (cross-language), gỡ lỗi và xử lý lỗi. * "Versioning" tự động là phần của bản chất tự mô tả của mỗi thành phần và ứng dụng **.NET Framework** ở chỗ mỗi thành phần đó có một tên (name) duy nhất được lưu giữ bên trong. Do các ứng dụng kết các tên duy nhất của các thành phần lại với nhau, bộ thực hiện (runtime) của **.NET Framework** có khả nǎng điều khiển ?versioning? của những thành phần dùng chung một cách thông minh và làm mất đi cái gọi là "DLL hell". * **Triển khai theo kiểu "No-touch" (không can thiệp):** **.NET Framework** bao hàm những đặc tính được cải tiến cho việc triển khai các ứng dụng. Việc cài đặt một ứng dụng vào trong hệ thống cũng đơn giản như việc sao chép nó vào trong một thư mục đích và cho nó chạy mà không cần đǎng ký như Windows Registry. * **Quản lý bộ nhớ tự động:   .NET Framework** là một môi trường "gom rác" (garbage-ollected). Công việc "gom rác" giải phóng các ứng dụng sử dụng các đối tượng  **.NET Framework** từ sự cần thiết loại bỏ dứt khoát các đối tượng này và làm giảm đi những lỗi lập trình chung không đáng có. * **Các thành phần tự-mô-tả:**Siêu dữ liệu (metadata) mà mỗi đối tượng trong **.NET Framework** chứa đựng cho phép bộ thực hiện (runtime) "chất vấn" các đối tượng về những kiểu dữ liệu, chức nǎng hoạt động, v.v. để xác định các đối tượng đã được gọi đúng cách chưa hơn là để cho lời gọi xảy ra và nhận lấy thất bại. Đặc tính này được gọi là "reflection". * Mô hình điều khiển giao diện người dùng ASP.NET và các đối tượng điều khiển được cải thiện làm tǎng thêm nǎng suất và hiệu quả do việc "đóng gói" những tương tác phức tạp trong các thành phần (component) trên máy phục vụ. * Sự tách biệt giữa mã và nội dung cho phép người phát triển và người tạo ra nội dung làm việc song song với nhau do việc lưu giữ nội dụng trong một file tách biệt khỏi các mã của ứng dụng. * **Tích hợp nền tảng chặt chẽ:**   **.NET Framework** cho phép những nhà phát triển sử dụng tất cả các ứng dụng Windows và các dịch vụ hiện hữu. Với ưu thế đó, người phát triển có thể sử dụng mã đang tồn tại trong khi tận dụng những lợi điểm, thế mạnh trong **.NET Framework**.   * **Viết ít mã hơn:**do **.NET Framework** sử dụng thiết kế "thành phần hóa" cao, những nhà phát triển có thể tập trung vào việc viết lôgic doanh nghiệp hơn là những công việc như quản lý bộ nhớ, quản lý trạng thái hay xác định khả nǎng của một trình duỵệt client? * **Sự nhạy cảm khi giải quyết các thách thức đối với doanh nghiệp thời nay** * **Hỗ trợ những chuẩn Internet công cộng:**   Những chuẩn được xem là phần lõi để chuyển giao phần mềm như là một dịch vụ. Microsoft trình bày những đặc tả cho ngôn ngữ lập trình C# và cơ sở hạ tầng ngôn ngữ chung CLI (Common Language Infrastructure) và chuyển tới ECMA cho việc chuẩn hóa.   * **Hỗ trợ dị bộ (không đồng bộ):**   **.NET Framework** tích hợp hai công nghệ truyền thông dị bộ cho khả nǎng biến đổi được và tính đáng tin cậy: **SOAP** và **MSMQ** (Microsoft Message Queuing Services).   * **Hỗ trợ giao dịch:**   Những nhà phát triển ứng dụng có thể thực hiện cả những công việc thao tác **.NET Framework** bên trong các giao dịch có chứa những hoạt động khác như cập nhật CSDL chẳng hạn. **.NET Framework** hỗ trợ các giao dịch thông qua **MTS** (Microsoft Transaction Services - các Dịch vụ Giao dịch của Microsoft) và **COM+[16]** và cung cấp một giao diện XA tương hợp các chuẩn.   * **Biên dịch ASP.NET** làm tǎng khả nǎng thực hiện bởi việc biên dịch các trang thay vì phải thông dịch chúng. * **Giám sát phiên ASP.NET**tự động và khởi động lại làm tǎng thêm độ tin cậy qua việc giám sát chạy các ứng dụng ASP.NET và thậm chí dừng và bắt đầu lại quá trình chạy nếu thấy cần thiết. * **Truy nhập dữ liệu toàn bộ với ADO.NET:**   **.NET Framework** có chứa ADO.NET, một giao diện hiệu nǎng truy nhập tới bất kỳ một CSDL nào được thiết kế riêng cho kiểu "ghép nối lỏng". ADO.NET cung cấp các dịch vụ truy cập dữ liệu cho các ứng dụng trên nền Web biến đổi được và các dịch vụ Web XML, bao gồm cả sự hỗ trợ mô hình dữ liệu đang kết nối (connected) cũng như ngừng kết nối (disconnected).  **Cải thiện các thao tác**   * **An ninh bảo mật dựa trên nền tảng bằng chứng (evidence-based):**   Hệ thống bảo mật truy cập mã của **.NET Framework** cho phép các nhà phát triển định ra những "giấy phép" được yêu cầu rằng mã của họ cần để hoàn thành sản phẩm. Bộ thực thi ngôn ngữ chung (common language runtime) chịu trách nhiệm bảo đảm rằng những "giấy phép" này là phù hợp hay bị loại bỏ, phụ thuộc vào chứng cớ bao gồm nhận diện người dùng, nhận diện mã, mã nào thực sự đang được sử dụng, mã được bắt nguồn từ nơi nào?   * **Windows Authentication tích hợp:**   **.NET Framework** cũng tích hợp với Windows Authentication (Chứng thực Windows). Windows Authentication tích hợp trước đây được biết như chứng thực NT LAN Manager và Windows NT Challenge/Response. Trong Windows Authentication tích hợp, trình duyệt cố gắng sử dụng những (giấy) ủy nhiệm của người dùng hiện hành từ một đǎng nhập tên miền. Nếu những ủy nhiệm này không được chấp nhận thi Windows Authentication tích hợp sẽ nhắc người dùng cho một tên và mật khẩu người dùng thông qua một hộp thoại. Khi Windows Authentication tích hợp đang được sử dụng thì mật khẩu người dùng không qua được từ máy khách đến máy phục vụ. nếu một người dùng đã đǎng nhập như là một người sử dụng tên miền trên một máy tính cục bộ, người dùng sẽ không bị chứng thực lần nữa khi truy nhập một máy tính mạng trong miền đó.  **Chứng thực Internet:**  Người dùng Internet thường cần những cơ chế chứng thực khác nhau. Các ứng dụng dùng **.NET Framework** có thể nắm lấy lợi thế và được cấu hình cho chứng thực sử dụng một sự kết hợp của máy phục vụ Web (Web server) và các nhà cung cấp chứng thực .**NET Framework.**   * **Hiệu suất được cải thiện:**   **.NET Framework** cải thiện hiệu suất các ứng dụng Web đặc thù. ASP.NET sử dụng khả nǎng tối ưu hoá cho những bộ xử lý nào đó và với việc cất giữ một cách linh động ASP.NET có thể cải thiện hiệu suất hơn hai đến ba lần so với các ứng dụng ASP hiện hành. |

**2.1.2 TÌM HIỂU NGÔN NGỮ LẬP TRÌNH C#**

* ĐỊNH NGHĨA VỀ C#

C# (hay C sharp) là một ngôn ngữ lập trình đơn giản, được phát triển bởi đội ngũ kỹ sư của Microsoft vào năm 2000, trong đó người dẫn đầu là Anders Hejlsberg và Scott Wiltamuth.

C# là ngôn ngữ lập trình hiện đại, hướng đối tượng và nó được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java.

C# được thiết kế cho Common Language Infrastructure (CLI), mà gồm Executable Code và Runtime Environment, cho phép chúng ta sử dụng các ngôn ngữ high-level đa dạng trên các nền tảng và cấu trúc máy tính khác nhau.

C# với sự hỗ trợ mạnh mẽ của .NET Framework giúp cho việc tạo một ứng dụng Windows Forms hay WPF (Windows Presentation Foundation), . . . trở nên rất dễ dàng.

* ĐẶC ĐIỂM CỦA NGÔN NGỮ LẬP TRÌNH C#

Các đặc điểm để làm cho C# là ngôn ngữ lập trình chuyên nghiệp được sử dụng rộng rãi:

### C# là ngôn ngữ đơn giản

Như ta đã biết thì ngôn ngữ C# dựng trên nền tảng C++ và Java nên ngôn ngữ C# khá đơn giản. Nếu chúng ta thân thiện với C và C++ hoậc thậm chí là Java, chúng ta sẽ thấy C# khá giống về diện mạo, cú pháp, biểu thức, toán tử và những chức năng khác được lấy trực tiếp từ ngôn ngữ C và C++, nhưng nó đã được cải tiến để làm cho ngôn ngữ đơn giản hơn. Một vài trong các sự cải tiến là loại bỏ các dư thừa, hay là thêm vào những cú pháp thay đổi.

### C# là ngôn ngữ hiện đại

Một vài khái niệm khá mới mẻ khá mơ hồ với các bạn vừa mới học lập trình, như xử lý ngoại lệ, những kiểu dữ liệu mở rộng, bảo mật mã nguồn..v..v... Đây là những đặc tính được cho là của một ngôn ngữ hiện đại cần có. Và C# chứa tất cả các đặt tính ta vừa nêu trên. Các bạn sẽ dần tìm hiểu được các đặt tính trên qua các bài học trong series này.

### C# là một ngôn ngữ lập trình thuần hướng đối tượng

Lập trình hướng đối tượng(tiếng Anh: Object-oriented programming, viết tắt: OOP) là một phương pháp lập trình có 4 tính chất. Đó là tính trừu tượng (*abstraction*), tính đóng gói (encapsulation), tính đa hình (polymorphism) và tính kế thừa (inheritance). C# hỗ trợ cho chúng ta tất cả những đặc tính trên. Và để hiểu rõ hơn thì chúng ta sẽ có một chương trình bày về phần này.

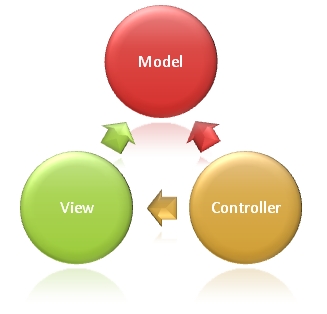
### C# là một ngôn ngữ ít từ khóa

C được sử dụng để mô tả thôn# là ngôn ngữ sử dụng giới hạn những từ khóa (gồm khoảng 80 từ khóa và mười mấy kiểu dữ liệu xây dựng sẵn). Nếu bạn nghĩ rằng ngôn ngữ có càng nhiều từ khóa thì sẽ càng mạnh mẽ hơn. Điều này không phải sự thật, lấy ví dụ ngôn ngữ C# làm điển hình nhé. Nếu bạn học sâu về C# bạn sẽ thấy rằng ngôn ngữ này có thể được sử dụng để làm bất cứ nhiệm vụ nào.

Ngoài những đặc điểm trên thì còn một số ưu điểm nổi bật của C#:

* C# có cấu trúc khá gần gũi với các ngôn ngữ lập trình truyền thống, nên cũng khá dể dàng tiếp cận và học nhanh với C#.
* C# có thể biên dịch trên nhiều nền tảng máy tính khác nhau.
* C# được xây dựng trên nền tảng của C++ và Java nên nó được thừa hưởng những ưu điểm của ngôn ngữ đó.
* C# là một phần của .NET Framework nên được sự chống lưng khá lớn đến từ bộ phận này.
* C# có IDE Visual Studio cùng nhiều plug-in vô cùng mạnh mẽ.

2.2  **TỔNG QUAN VỀ MÔ HÌNH MVC ASP.NET**

Mẫu kiến trúc Model – View – Controller được sử dụng nhằm chi ứng dụng thành ba thành phần chính: model, view và controller. Nền tảng ASP.NET MVC giúp cho chúng ta có thể tạo được các ứng dụng web áp dụng mô hình MVC thay vì tạo ứng dụng theo mẫu ASP.NET Web Forsm. Nền tảng ASP.NET MVC có đặc điểm nổi bật là nhẹ (lighweigt), dễ kiểm thử phần giao diện (so với ứng dụng Web Forms), tích hợp các tính năng có sẵn của ASP.NET. Nền tảng ASP.NET MVC được định nghĩa trong namespace System.Web.Mvc và là một phần của name space System.Web.  
 MVC là một mẫu thiết kế (design pattern) chuẩn mà nhiều lập trình viên đã quen thuộc. Một số loại ứng dụng web sẽ thích hợp với kiến trúc MVC. Một số khác vẫn thích hợp với ASP.NET Web Forms và cơ chế postbacks. Đôi khi có những ứng dụng kết hợp cả hai kiến trúc trên.  
  
**2.2.1 Nền tảng MVC bao gồm các thành phần dưới đây:**  
  
  
  
*Hình 02: Mẫu Model – View – Controller*  
*Models*: Các đối tượng Models là một phần của ứng dụng, các đối tượng này thiết lập logic của phần dữ liệu của ứng dụng. Thông thường, các đối tượng model lấy và lưu trạng thái của model trong CSDL. Ví dụ như, một đối tượng Product (sản phẩm) sẽ lấy dữ liệu từ CSDL, thao tác trên dữ liệu và sẽ cập nhật dữ liệu trở lại vào bảng Products ở SQL Server.  
  
Trong các ứng dụng nhỏ, model thường là chỉ là một khái niệm nhằm phân biệt hơn là được cài đặt thực thụ, ví dụ, nếu ứng dụng chỉ đọc dữ liệu từ CSDL và gởi chúng đến view, ứng dụng khong cần phải có tầng model và các lớp lien quan. Trong trường hợp này, dữ liệu được lấy như là một đối tượng model (hơn là tầng model).  
  
*Views*: Views là các thành phần dùng để hiển thị giao diện người dùng (UI). Thông thường, view được tạo dựa vào thông tin dữ liệu model. Ví dụ như, view dùng để cập nhật bảng Products sẽ hiển thị các hộp văn bản, drop-down list, và các check box dựa trên trạng thái hiện tại của một đối tượng Product.  
  
*Controllers*: Controller là các thành phần dùng để quản lý tương tác người dùng, làm việc với model và chọn view để hiển thị giao diện người dùng. Trong một ứng dụng MVC, view chỉ được dùng để hiển thị thông tin, controller chịu trách nhiệm quản lý và đáp trả nội dung người dùng nhập và tương tác với người dùng. Ví dụ, controller sẽ quản lý các dữ liệu người dùng gởi lên (query-string values) và gởi các giá trị đó đến model, model sẽ lấy dữ liệu từ CSDL nhờ vào các giá trị này.   
  
Mẫu MVC giúp bạn tạo được các ứng dụng mà chúng phân tách rạch ròi các khía cạnh của ứng dụng (logic về nhập liệu, logic xử lý tác vụ và logic về giao diện). Mẫu MVC chỉ ra mỗi loại logic kể trên nên được thiếp lập ở đâu trên ứng dụng. Logic giao diện (UI logic) thuộc về views. Logic nhập liệu (input logic) thuộc về controller. Và logic tác vụ (Business logic – là logic xử lý thông tin, mục đích chính của ứng dụng) thuộc về model. Sự phân chia này giúp bạn giảm bớt được sự phức tạp của ứng dụng và chỉ tập trung vào mỗi khía cạnh cần được cài đặt ở mỗi thời điểm. Ví dụ như bạn chỉ cần tập trung vào giao diện (views) mà không phải quan tâm đến logic xử lý thông tin của ứng dụng.  
  
Để quản lý sự phức tạp của ứng dụng, mẫu MVC giúp cho chúng ta có thể kiểm thử ứng dụng dễ dàng hơn hẳn so với khi áp dụng mẫu Web Forms. Ví dụ, trong một ứng dụng ASP.NET Web Forms, một lớp thường được sử dụng để hiển thị thông tin xuất ra cho người dùng và đồng thời xử lý thông tin người dùng nhập. Việc xây dựng các bộ test tự động cho ứng dụng Web Forms là rất phức tạp, bởi để kiểm thử mỗi trang web, bạn phải khởi tạo đối tượng trang, khởi tạo tất cả các control được sử dụng trong trang và các lớp phụ thuộc trong ứng dụng. Và bởi vì có quá nhiều lớp cần được khởi tạo để chạy được trang, thật khó để có thể viết các test chỉ tập trung vào một khía cạnh nào đó của ứng dụng. Và vì thế, kiểm thử đối với các ứng dụng dứa trên nền tảng Web Forms sẽ khó khăn hơn nhiều so với khi áp dụng trên ứng dụng MVC. Hơn thế nữa, việc kiểm thử trên nền tảng Web Forms yêu cầu phải sử dụng đến web server. Nền tảng MVC phân tách các thành phần và sử dụng các interface (khái niệm giao diện trong lập trình hướng đối tượng), và nhờ đó có thể kiểm thử các thành phần riêng biệt trong tình trạng phân lập với các yếu tố còn lại của ứng dụng.  
  
Sự phân tách rạch ròi ba thành phần của ứng dụng MVC còn giúp cho việc lập trình diễn ra song song. Ví dụ như một lập trình viên làm việc với view, lập trình viên thứ hai lo cài đặt logic của controller và lập trình viên thứ ba có thể tập trung vào logic tác vụ của model tại cùng một thời điểm.

Lựa chọn áp dụng MVC trong xây dựng ứng dụng  
Bạn cần phải xem xét kỹ càng việc áp dụng mô hình ASP.NET MVC hay mô hình ASP.NET Web Forms khi xây dựng một ứng dụng. Mô hình MVC không phải là mô hình thay thế cho Web Forms, bạn có thể dùng một trong hai mô hình.  
Trước khi quyết định sử dụng MVC hay Web Forms cho một web site cụ thể, bạn cần phải phân tích lợi ích khi chọn một trong hai hướng.

**2.2.2 Lợi ích của ứng dụng web dựa trên mô hình MVC**  
Nền tảng ASP.NET MVC mang lại những lợi ích sau:

* Dễ dàng quản lý sự phức tạp của ứng dụng bằng cách chia ứng dụng thành ba thành phần model, view, controller
* Nó không sử dụng view state hoặc server-based form. Điều này tốt cho những lập trình viên muốn quản lý hết các khía cạnh của một ứng dụng.
* Nó sử dụng mẫu Front Controller, mẫu này giúp quản lý các requests (yêu cầu) chỉ thông qua một Controller. Nhờ đó bạn có thể thiết kế một hạ tầng quản lý định tuyến.
* Hỗ trợ tốt hơn cho mô hình phát triển ứng dụng hướng kiểm thử (TDD)
* Nó hỗ trợ tốt cho các ứng dụng được xây dựng bởi những đội có nhiều lập trình viên và thiết kế mà vẫn quản lý được tính năng của ứng dụng.

Lợi ích của ứng dụng được xây dựng trên nền tảng Web Forms

* Nó hỗ trợ cách lập trình hướng sự kiện, quản lý trạng thái trên giao thức HTTP, tiện dụng cho việc phát triển các ứng dụng Web phục vụ kinh doanh. Các ứng dụng trên nền tảng Web Forms cung cấp hàng tá các sự kiện được hỗ trợ bởi hàng trăm các server controls.
* Sử dụng mẫu Page Controller.
* Mô hình này sử dụng view state hoặc server-based form, nhờ đó sẽ giúp cho việc quản lý trạng thái các trang web dễ dàng.
* Nó rất phù hợp với các nhóm lập trình viên quy mô nhỏ và các thiết kế, những người muốn tận dụng các thành phần giúp xây dựng ứng dụng một cách nhanh chóng.
* Áp dụng Web Forms giúp giảm bớt sự phức tạp trong xây dựng ứng dụng, bởi vì các thành phần (lớp Page, controls,…) được tích hợp chặc chẽ và thường thì giúp bạn viết ít code hơn là áp dụng theo mô hình MVC.

**2.2.3 Các tính năng của nền tảng ASP.NET MVC**

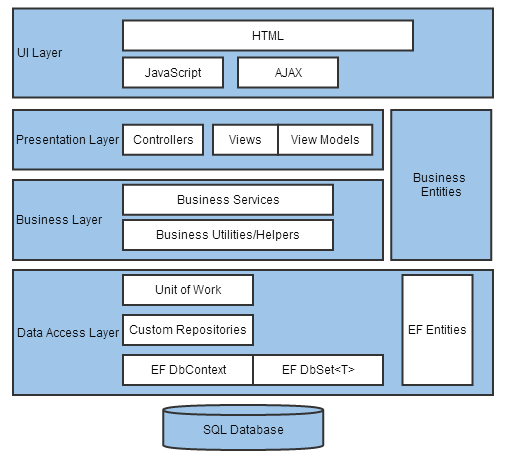
* Tách bạch các tác vụ của ứng dụng (logic nhập liệu, business logic, và logic giao diện), dễ dàng kiểm thử và mặc định áp dụng hướng phát triển TDD. Tất cả các tính năng chính của mô hình MVC được cài đặt dựa trên interface và được kiểm thử bằng cách sử dụng các đối tượng mocks, mock object là các đối tượng mô phỏng các tính năng của những đối tượng thực sự trong ứng dụng. Bạn có thể kiểm thử unit-test cho ứng dụng mà không cần chạy controller trong tiến trình ASP.NET, và điều đó giúp unit test được áp dụng nhanh chóng và tiện dụng. Bạn có thể sử dụng bất kỳ nền tảng unit-testing nào tương thích với nền tảng .NET.
* MVC là một nền tảng khả mở rộng (extensible) & khả nhúng (pluggable). Các thành phần của ASP.NET MVC được thiết kể để chúng có thể được thay thế một cách dễ dàng hoặc dễ dàng tùy chỉnh. Bạn có thể nhúng thêm view engine, cơ chế định tuyến cho URL, cách kết xuất tham số của action-method và các thành phần khác. ASP.NET MVC cũng hỗ trợ việc sử dụng Dependency Injection (DI) và Inversion of Control (IoC). DI cho phép bạn gắn các đối tượng vào một lớp cho lớp đó sử dụng thay vì buộc lớp đó phải tự mình khởi tạo các đối tượng. IoC quy định rằng, nếu một đối tượng yêu cầu một đối tượng khác, đối tượng đầu sẽ lấy đối tượng thứ hai từ một nguồn bên ngoài, ví dụ như từ tập tin cấu hình. Và nhờ vậy, việc sử dụng DI và IoC sẽ giúp kiểm thử dễ dàng hơn.
* ASP.NET MVC có thành phần ánh xạ URL mạnh mẽ cho phép bạn xây dựng những ứng dụng có các địa chỉ URL xúc tích và dễ tìm kiếm. Các địa chỉ URL không cần phải có phần mở rộng của tên tập tin và được thiết kế để hỗ trợ các mẫu định dạng tên phù hợp với việc tối ưu hóa tìm kiếm (URL) và phù hợp với lập địa chỉ theo kiểu REST.
* Hỗ trợ sử dụng đặc tả (các thẻ) của các trang ASP.NET(.aspx), điều khiển người dùng (.ascx) và trang master page (.marter). Bạn có thể sử dụng các tính năng có sẵn của ASP.NET như là sử dụng lồng các trang master page, sử dụng in-line expression (<%= %>), sử dụng server controls, mẫu, data-binding, địa phương hóa (localization) và hơn thế nữa.
* Hỗ trợ các tính năng có sẵn của ASP.NET như cơ chế xác thực người dùng, quản lý thành viên, quyền, output caching và data caching, seession và profile, quản lý tình trạng ứng dụng, hệ thống cấu hình…
* ASP.NET MVC 3 còn bổ sung một view engine mới là Razor View Engine cho phép thiết lập các view nhanh chóng, dễ dàng và tốn ít công sức hơn so với việc sử dụng Web Forms view engine.

**2.3 TÌM HIỂU VỀ ENTITY FRAMEWORK**

**2.3.1 Định nghĩa về Entity Framework**

- [Entity Framework](https://www.dammio.com/glossary/entity-framework)**(**[**EF**](https://www.dammio.com/glossary/ef)**)** là một framework ánh xạ quan hệ đối tượng (ORM) dành cho ADO.NET, là 1 phần của .NET Framework. [Entity Framework](https://www.dammio.com/glossary/entity-framework) cho phép các nhà phát triển Web tương tác với dữ liệu quan hệ theo phương pháp hướng đối tượng đặc trưng. Lợi ích lớn nhất của  [**Entity Framework**](https://www.dammio.com/glossary/entity-framework)là 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**](https://www.dammio.com/glossary/entity-framework)được Microsoft hỗ trợ phát triển lâu dài và bền vững, vì vậy [**Entity Framework**](https://www.dammio.com/glossary/entity-framework)là 1 framework mạnh nhất hiện nay để phát triển ứng dụng Web với sự hỗ trợ đông đảo của các nhà phát triển Web.  
  
**2.3.2 Vị trí của**[**Entity Framework**](https://www.dammio.com/glossary/entity-framework)**trong mô hình lập trình và phát triển Web**

Chắc hẳn, bạn đã biết về mô hình Web 3 tầng (n tầng) khi đọc sách, giáo trình, bài giảng về nội dung lập trình Web. Tuy nhiên, vấn đề bạn cần phải hiểu rõ là: việc định rõ số lượng các tầng và các mối liên hệ giữa các tầng trong mô hình phát triển Web cũng rất đa dạng, tùy theo cách hiểu các lập trình viên và dự án Web. Do đó, việc hiểu [**Entity Framework**](https://www.dammio.com/glossary/entity-framework) nằm ở đâu trong mô hình Web 3 tầng cũng không hoàn toàn dễ dàng.

Đầu tiên, chúng ta có vị trí của [**Entity Framework**](https://www.dammio.com/glossary/entity-framework) trong mô hình dự án Web ASP.NET [MVC](https://www.dammio.com/glossary/mvc).  
[](https://dammio.files.wordpress.com/2016/11/ef_n_tier1.png)

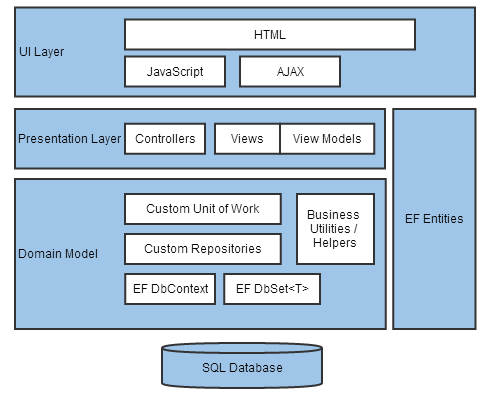
Hình 03: vị trí của [**Entity Framework**](https://www.dammio.com/glossary/entity-framework) trong mô hình dự án

Web ASP.NET [MVC](https://www.dammio.com/glossary/mvc).

- Trong mô hình trên, chúng ta thấy vị trí của [**Entity Framework**](https://www.dammio.com/glossary/entity-framework) nằm trọn trong tầng Data Access Layer, tức là [**Entity Framework**](https://www.dammio.com/glossary/entity-framework)đóng vai trò là nơi trung gian để tương tác dữ liệu từ database với các tầng cao hơn, chẳng hạn như Business Layer. Hay nói cách khác, [**Entity Framework**](https://www.dammio.com/glossary/entity-framework) đóng vai trò như là phiên bản mới của ADO.NET. Tuy nhiên vị trí của [**Entity Framework**](https://www.dammio.com/glossary/entity-framework) trong mô hình này chỉ có trên lý thuyết.

Thật sự, khi bắt tay xây dựng nhiều dự án Web, vị trí thường thấy [**Entity Framework**](https://www.dammio.com/glossary/entity-framework) ở các dự án website trên thực tế như mô hình sau.  
[](https://dammio.files.wordpress.com/2016/11/ef_n_tier_alternate1.png)

Hình 04: mô hình Entity framework trên dự án website

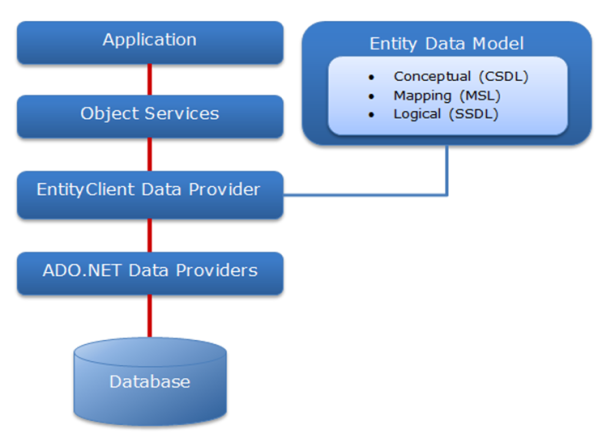
Trong hình trên, [**Entity Framework**](https://www.dammio.com/glossary/entity-framework) dường như có vị trí rất tự do, xuất hiện ở cả 3 tầng Data Access Layer, Business Layer, và cả Presentation Layer. Các bạn có thể hình dung [**Entity Framework**](https://www.dammio.com/glossary/entity-framework) như là 1 thư viện, hễ tầng nào cần thì chỉ cần gọi đến. Mô hình này xuất phát có thể là sự cẩu thả của lập trình viên khi không thích tuân theo tiêu chuẩn quan hệ giữa các tầng hoặc cách giải thích khác là sự lỏng lẻo giúp cho việc lập trình thuận tiện hơn. Trong vài mô hình người ta gom tầng Data Access Layer và Business Layer chỉ làm tầng Data Model, và vị trí [**Entity Framework**](https://www.dammio.com/glossary/entity-framework) cũng bao hàm tầng Data Model và Presentation Layer như hình.  
[](https://dammio.files.wordpress.com/2016/11/ef_domain_model_abstracted1.png)

**2.3.3 .Mô hình kiến trúc**[**Entity Framework**](http://www.dammio.com/glossary/entity-framework)

**Application**  
Application (ứng dụng) là tầng chứa giao diện trang Web (HTML, [CSS](http://www.dammio.com/glossary/css), Javascript, hình ảnh, …) và các đoạn mã nguồn (C#, VB) để tương tác dữ liệu với các tầng khác trong mô hình thông qua Object Services.

**Object Services**  
Object Services (tạm dịch là **các dịch vụ đối tượng**) là tầng chứa quá trình tương tác giữa ứng dụng và database, hay nói cách khác nó là nơi chủ yếu để truy cập dữ liệu từ database và trả ngược kết quả về giao diện. Object Services cung cấp các tiện ích để truy vết các thay đổi và quản lý nhận dạng, đồng thời là các quan hệ và thay đổi ở database.

**EntityClient Data Provider**  
Đây là tầng cung cấp các kết nối, diễn dịch các truy vấn thực thể thành truy vấn nguồn dữ liệu (chuyển L2E – [LINQ](http://www.dammio.com/glossary/linq) to Entity hay các truy vấn thực thể SQL thành truy vấn SQL), trả về data reader để [EF](http://www.dammio.com/glossary/ef) dùng chuyển dữ liệu thực thể thành các đối tượng. Phần này kết nối ADO.NET Data Providers để gửi hoặc lấy dữ liệu từ database. Tầng này hoàn toàn khác với EDM (Entity Data Model) khi thực thi các truy vấn tương tự như cách thực hiện ở ADO.NET Provider.

[](https://dammio.files.wordpress.com/2016/11/entity_framework_architecture.png)

Hình 05: **Mô hình kiến trúc**[Entity Framework](http://www.dammio.com/glossary/entity-framework)

EntityClient Data Provider sử dụng ESQL (Entity SQL), một ngôn ngữ truy vấn độc lập dạng văn bản, tương tự như SQL.

**ADO.NET Data Providers**  
Đây là tầng thấp nhấp để dịch các truy vấn L2E ([**LINQ**](http://www.dammio.com/glossary/linq) to Entity) thông qua cây lệnh thành các câu lệnh SQL và thực thi các câu lệnh trong hệ thống DBMS (database management system – hệ quản lý dữ liệu) nào đó. Tầng này kết với database sử dụng ADO.NET.

**EDM (Entity Data Model)**  
EDM (tạm dịch là **mô hình dữ liệu thực thể**) chứa 3 phần chính: mô hình khái niệm (CSDL – Conceptual schema definition language), mô hình ánh xạ (MSL – mapping specification language) và mô hình lưu trữ (SSDL – store schema definition language). EDM khác với EntityClient Data Provider ở chỗ EDM sử dụng [**LINQ**](http://www.dammio.com/glossary/linq) là ngôn ngữ truy vấn tương tác với database.

**Mô hình khái niệm (CSDL – Conceptual schema definition language)**  
Mô hình khái niệm chứa các lớp mô hình và mối quan hệ giữa các lớp này. Điều này để độc lập với mô hình quan hệ các bảng trong database.

**Mô hình lưu trữ (SSDL – store schema definition language)**  
Mô hình lưu trữ là 1 mô hình thiết kế database bao gồm các bảng, view, stored procedure (thủ tục), và mối quan hệ giữa chúng và các khóa. Mô hình này thể hiện gần giống mô hình quan hệ các bảng trong database.

**Mô hình ánh xạ (MSL – mapping specification language)**  
Mô hình ánh xạ gồm thông tin về cách mô hình khái niệm được ánh xạ đến mô hình lưu trữ.

**L2E (**[**LINQ**](http://www.dammio.com/glossary/linq) **to Entities)**: là 1 ngôn ngữ truy vấn được dùng để viết các truy vấn trái với mô hình đối tượng. L2E trả về các thực thể, được định nghĩa bởi mô hình khái niệm. Chúng ta có thể dùng [**LINQ**](http://www.dammio.com/glossary/linq)trong ngôn ngữ này.

**2.4 TÌM HIỂU NGÔN NGỮ SQL**

## 2.4.1 SQL là ngôn ngữ của cơ sở dữ liệu.

SQL,  viết  tắt  của  Structured  Query  Language  (ngôn  ngữ  hỏi  có  cấu  trúc),  là công cụ sử dụng để tổ chức, quản lý và truy xuất dữ liệu đuợc lưu trữ trong các cơ sở dữ liệu. SQL là một hệ thống ngôn ngữ bao gồm tập các câu lệnh sử dụng để tương tác với cơ sở dữ liệu quan hệ.

SQL là một chuẩn ANSI (American National Standards Institute - Viện Tiêu chuẩn Quốc gia Mỹ) cho các hệ thống truy xuất cơ sở dữ liệu. Các phát biểu SQL dùng để truy tìm và cập nhật  dữ liệu trong một cơ sở dữ liệu.

SQL làm việc với các trình quản lý cơ sở dữ liệu như Access, DB2, Informix, Microsoft SQL Server, Oracle, Sybase, và nhiều trình khác (đáng tiếc là đa số trong chúng có các phần mở rộng ngôn ngữ SQL riêng).

Khả năng của SQL vượt xa so với một công cụ truy xuất dữ liệu, mặc d ù đây là mục đích ban đầu khi SQL được xây dựng nên và truy xuất dữ liệu vẫn còn là một trong những chức năng quan trọng của nó. SQL được sử dụng để điều khiển tất cả các chức năng mà một hệ quản trị cơ sở dữ liệu cung cấp cho người dùng bao gồm:

Định nghĩa dữ liệu: SQL cung cấp khả năng định nghĩa các cơ sở dữ liệu, các cấu trúc lưu trữ và tổ chức dữ liệu cũng như mối quan hệ giữa các thành phần dữ liệu.

Truy xuất và thao tác dữ liệu: Với SQL, người dùng có thể dễ dàng thực hiện các thao tác truy xuất, bổ sung, cập nhật và loại bỏ dữ liệu trong các cơ sở dữ liệu

Điều khiển truy cập: SQL có thể được sử dụng để cấp phát và kiểm soát các thao tác của người sử dụng trên dữ liệu, đảm bảo sự an toàn cho cơ sở dữ liệu

 Đảm bảo toàn vẹn dữ liệu: SQL định nghĩa các ràng buộc toàn vẹn trong cơ sở dữ liệu nhờ đó đảm bảo tính hợp lệ và chính xác của dữ liệu trước các thao tác cập nhật cũng như các lỗi của hệ thống.

Như vậy,  có thể nói rằng SQL là một ngôn ngữ hoàn thiện được sử dụng trong các hệ thống cơ sở dữ liệu và là một thành phần không thể thiếu trong các hệ quản trị c ơ sở dữ liệu. Mặc dù SQL không phải là một ngôn ngữ lập trình như C, C++, Java,... song các câu lệnh mà SQL cung cấp có thể được nhúng vào trong các ngôn ngữ lập trình nhằm xây dựng các ứng dụng tương tác với cơ sở dữ liệu.

Khác với các ngôn ngữ lập trình quen thuộc như C, C++, Java,... SQL là ngôn ngữ có tính khai báo. Với SQL, người dùng chỉ cần mô tả các yêu cầu cần phải thực hiện trên cơ sở dữ liệu mà không cần phải chỉ ra cách thức thực hiện các yêu cầu như thế nào. Chính vì vậy, SQL là ngôn ngữ dễ tiếp cận và dễ sử dụng.

## 2.4.2 Vai trò của SQL

Bản thân SQL không phải là một hệ quản trị cơ sở dữ liệu, nó không thể tồn tại độc lập. SQL thực sự là một phần của hệ quản trị cơ sở dữ liệu, nó xuất hiện trong các hệ quản trị c ơ sở dữ liệu với vai trò ngôn ngữ và là công cụ giao tiếp giữa người sử dụng và hệ quản trị cơ sở dữ liệu.

Trong hầu hết các hệ quản trị cơ sở dữ liệu quan hệ, SQL có những vai trò như sau:

SQL  là  ngôn  ngữ hỏi có  tính  tương  tác: Người  sử  dụng  có  thể  dễ dàng thông qua các trình tiện ích để gởi các yêu cầu dưới dạng các câu lệnh SQL đến cơ sở dữ liệu và nhận kết quả trả về từ cơ sở dữ liệu

SQL là ngôn ngữ lập trình cơ sở dữ liệu: Các lập trình viên có thể nhúng các  câu  lệnh SQL  vào  trong  các  ngôn  ngữ   lập  trình để  xây  dựng  nên  các chương trình ứng dụng giao tiếp với cơ sở dữ liệu

SQL là ngôn ngữ quản trị cơ sở dữ liệu: Thông qua SQL, người quản trị cơ sở dữ liệu có thể quản lý được cơ sở dữ liệu, định nghĩa các cấu trúc lưu trữ dữ liệu, điều khiển truy cập cơ sở dữ liệu,...

SQL là ngôn ngữ cho các hệ thống khách/chủ (client/server) : Trong các hệ thống cơ sở dữ liệu khách/chủ, SQL được sử dụng như là công cụ để giao tiếp giữa các trình ứng dụng phía máy khách với máy chủ cơ sở dữ liệu.

SQL là ngôn ngữ truy cập dữ liệu trên Internet: Cho đến nay, hầu hết các máy chủ Web cũng như các máy chủ trên Internet sử dụng SQL với vai trò là ngôn ngữ để tương tác với dữ liệu trong các cơ sở dữ liệu.

SQL là ngôn ngữ cơ sở dữ liệu phân tán: Đối với các hệ quản trị cơ sở dữ liệu phân tán, mỗi một hệ thống sử dụng SQL để giao tiếp với các hệ thống khác tr ên mạng, gởi và nhận các yêu cầu truy xuất dữ liệu với nhau.

SQL là ngôn ngữ sử dụng cho các cổng giao tiếp cơ sở dữ liệu: Trong một hệ thống mạng máy tính với nhiều hệ quản trị cơ sở dữ liệu khác nhau, SQL thường được sử dụng như là một chuẩn ngôn ngữ để giao tiếp giữa các hệ quản trị cơ sở dữ liệu.

## 2.4.3 Các bảng cơ sở dữ liệu

Cơ sở dữ liệu chứa các đối tượng gọi là các Bảng (Tables).

Các Mẩu tin (Records) lưu trong các bảng này. Các bảng được gọi theo tên bảng (như "Persons", "Orders", "Suppliers"). Các bảng chứa các Cột (Columns) và các Dòng (Rows) dữ liệu. Dòng chứa các mẩu tin (như mẩu tin về một người). Cột chứa dữ liệu (như First Name, Last Name, Address, và City). Một ví dụ là bảng "Persons" sau:

|  |  |  |  |
| --- | --- | --- | --- |
| LastName | FirstName | Address | City |
| Hansen | Ola | Timoteivn 10 | Sandnes |
| Svendson | Tove | Borgvn 23 | Sandnes |
| Pettersen | Kari | Storgt 20 | Stavanger |

LastName, FirstName, Address, và City là các cột của bảng. Các dòng chứa ba mẩu tin của 3 người.

## 2.4.4 Các truy vấn SQL

Với SQL, chúng ta có thể truy vấn một cơ sở dữ liệu và nhận được một kết quả trả về với dạng bảng.  Một truy vấn giống như sau:

SELECT LastName FROM Persons

Sẽ trả về một kết quả giống như sau:

|  |
| --- |
| LastName |
| Hansen |
| Svendson |
| Pettersen |

Chú ý: Vài hệ cơ sở dữ liệu cần một dấu “;” ở cuối phát biểu SQL. Chúng ta không dùng dấu “;” trong bài viết này.

## 2.4.5 Thao tác dữ liệu SQL

SQL là một cú pháp để thực hiện các truy vấn. Nhưng ngôn ngữ SQL cũng chứa các cú pháp cập nhật các mẩu tin (record), chèn các mẩu tin mới và xóa các mẩu tin đang tồn tại.

Các lệnh truy vấn và cập nhật này thuộc dạng Ngôn ngữ Thao tác Dữ liệu (Data Manipulation Language - DML) một phần của SQL:

SELECT – trích dữ liệu từ một cơ sở dữ liệu

UPDATE – cập nhật dữ liệu trong một cơ sở dữ liệu

DELETE – xóa dữ liệu từ một cơ sở dữ liệu

INSERT – chèn dữ liệu mới vào trong một cơ sở dữ liệu

## 2.4.6 Định nghĩa dữ liệu SQL

Ngôn ngữ Định nghĩa Dữ liệu (Data Definition Language - DDL) một phần của SQL, cho phép tạo hay xóa các bảng cơ sở dữ liệu. Chúng ta cũng có thể định nghĩa các chỉ mục (các khóa - key), chỉ định liên kết giữa các bảng, và ràng buột giữa các bảng cơ sở dữ liệu.

Các phát biểu DDL quan trọng nhất trong SQL là::

CREATE TABLE – tạo một bảng cơ sở dữ liệu mới

ALTER TABLE – thay đổi (alters) một bảng cơ sở dữ liệu

DROP TABLE – xóa một bảng cơ sở dữ liệu

CREATE INDEX – tạo một chỉ mục (khóa tìm kiếm)

DROP INDEX – xoá một chỉ mục

## SQL và ASP

SQL là một phần quan trọng của ASP (Active Server Pages), vì ADO (Active Data Object) được dùng trong ASP để truy xuất cơ  sở dữ liệu, ADO dựa trên SQL để truy xuất dữ liệu.

**CHƯƠNG 3: XÂY DỰNG CHƯƠNG TRÌNH**

**3.1 PHÂN TÍCH NGHIỆP VỤ QUẢN LÝ KHÁCH SẠN**

Hình 06: Cơ cấu tổ chức của Khách Sạn

## 

## 3.1.1 Hoạt động, cơ cấu tổ chức, chức năng, nhiệm vụ của từng bộ phận :

• Giám đốc khách sạn: Có nhiệm vụ quản lý trực tiếp khách sạn, mọi vấn đề của khách sạn đều phải thông qua giám đốc và giám đốc có quyền quyết định tất cả mọi thông tin xử lý trong khách sạn.

• Phòng lễ tân: Có nhiệm vụ tiếp nhận khách thuê phòng , trả lời những yêu cầu thắc mắc cho khách hàng, quản lý thông tin về phòng và khác hàng. Lập phiếu đăng ký khi khách thuê phòng, nhận yêu cầu khi khách hàng muốn sử dụng dịch vụ của khách sạn. Lập phiếu thanh toán khi khách hàng trả phòng và chuyển sang phòng tài chính để lập hóa đơn thanh toán.

• Phòng bảo vệ: Bảo vệ có nhiệm vụ giữ xe, khuân vác hành lý và giữ gìn an ninh trật tự trong khách sạn. Theo dõi và chịu trách nhiệm về các thiết bị của khách sạn.

• Phòng phục vụ: Các nhân viên trong phòng phục vụ có nhiệm vụ dọn dẹp vệ sinh phòng, phục vụ nhu cầu ăn uống, giặt là khi khách yêu cầu, kiểm tra hiện trạng cơ sở vật chất trong phòng khi khách trả phòng.

• Phòng tài chính: Quản lý thông tin tài chính thu chi trong khách sạn, có nhiệm vụ lập hóa đơn thanh toán cho khách hàng, lập các báo cáo việc kinh doanh của khách sạn. Hàng tháng phòng tài chính phải tiến hành lập các báo cáo kiểm kê tài chính thu chi sau đó gửi lên cho giám đốc.

## 3.2.2 Nghiên cứu, đánh giá hệ thống thông tin quản lý của khách sạn.

### 5.2.1 Phân tích hệ thống thông tin quản lý của khách sạn.

Hệ thống quản lý khách sạn phải làm những hoạt động sau:

+ Quản lý khách hàng:

Mỗi khách hàng khi đăng ký thuê phòng phải cung cấp các thông tin cá nhân để khách sạn dễ dàng quản lý. Những thông tin này sẽ được quản lý khi khách còn ở trong khách sạn. Sau khi khách trả phòng thông tin sẽ được lưu lại nhằm tạo nên nguồn khách hàng thân thiết cho nhưng lần sau khách quay lại

+ Quản lý việc đặt phòng và trả phòng :

Các nguồn khách đặt phòng

- Nguồn khách đặt phòng trực tiếp: Khách trực tiếp đến khách sạn; khách gọi điệnthoại; gửi thư tín, Fax, thư điện tử (email); qua internet  
- Nguồn khách đặt phòng qua các đại lý trung gian: đại lý du lịch, hãng lữ hành, hãng hàng không, văn phòng du lịch địa phương

- Nguồn khách qua hệ thống đặt phòng trung tâm  
Hệ thống đặt phòng trung tâm thường được thiết lập giữa các khách sạn trong cùng một tập đoàn hoặc các tập đoàn khách sạn khác nhau nhằm mục đích tạo thuận lợi cho khách hàng trong việc đặt phòng khách sạn.

Các hình thức đặt phòng  
- Đặt phòng có đảm bảo (Guaranteed Reservation)  
Đặt phòng có đảm bảo là thoả thuận giữa khách sạn và khách, theo đó khách sạn phải đảm bảo giữ phòng cho khách tới thời điểm check - out của ngày hôm sau tính theo ngày khách định đến. Nếu khách không sử dụng phòng và không báo huỷ, theo quy định của khách sạn thì khách phải đền bù tiền cho khách sạn.  
Các phương thức đảm bảo: thanh toán trước tiền phòng (Prepayment); tiền đặt cọc (Deposit); thẻ tín dụng (Credit card); các đại lý du lịch, công ty.  
- Đặt phòng không được đảm bảo (Non - guaranteed reservation)  
Đặt phòng không được đảm bảo là việc đăng ký giữ chỗ trước mà khách sạn chỉ giữ phòng cho khách tới một thời điểm nhất định tuỳ theo quy định của từng khách sạn (thường là 18.00pm – cancellation hour) của ngày khách định đến.

Quy trình nhận đặt phòng khách sạn :  
Bước 1: Nhận yêu cầu đặt phòng và xác định khả năng đáp ứng của khách sạn .

Hệ thống đặt phòng phải nắm được đầy đủ những thông tin sau:  
- Tên khách, tên đoàn khách, tên người đăng ký.  
- Địa chỉ, số ĐT, số fax (kể cả mã vùng), email của khách.  
- Số khách cùng đi trong đoàn.  
- Ngày giờ đến và số đêm lưu trú.  
- Số lượng phòng và loại phòng.  
- Giá phòng và hình thức thanh toán  
- Loại đặt phòng (đảm bảo hay không đảm bảo)  
- Các yêu cầu đặc biệt của khách.

Lưu ý :Trong khi tiếp nhận yêu cầu đặt phòng của khách từ đó xác nhận với khách hàng, hệ thống phải kiểm tra xem số lượng phòng và loại phòng còn trống (available room) của khách sạn có đáp ứng yêu cầu của khách hay không? (Kiểm tra trên hệ thống phần mềm)  
Chấp nhận Overbooking với một tỷ lệ phần trăm nhất định nhằm tối đa công suất sử dụng phòng.  
- Số phòng thực trống được tính theo công thức:  
Tổng số phòng của khách sạn:  
Trừ đi   
- Số phòng không thể đón khách (OOO)  
- Số phòng đang có khách ở (Occupied)  
- Số phòng đã được đặt trước (Resevation)  
Cộng thêm   
+ Số phòng đặt trước không chắc chắn (tính theo xác suất Noshow)  
+ Số phòng mới huỷ bỏ (cancellation)  
+ Số phòng khách trả sớm hơn dự định (Understay)  
Bằng Số phòng thực trống (Room available)

Bước 2 : Khách sạn khẳng định lại việc đặt phòng

Thông thường, các khách sạn yêu cầu khách khẳng định lại việc đặt phòng trước ngày khách đến tuỳ theo quy định của từng khách sạn tại các thời điểm khác nhau (thường là 15 ngày cho khách đoàn và 3 ngày cho khách lẻ)  
Nếu khách có sửa đổi hoặc huỷ bỏ thì nhân viên phải thực hiện các bước sau:  
- Xác định lại phiếu đặt phòng trước đó  
- Nhận thông tin sửa đổi hoặc huỷ bỏ đặt phòng của khách   
Sửa đổi:  
Kiểm tra xem khách sạn có khả năng đáp ứng được yêu cầu mới của khách hay không?   
Khẳng định lại với khách về sự thay đổi của khách.  
Huỷ bỏ:  
Nếu khách muốn huỷ bỏ đặt phòng thì nhân viên vẫn phải vui vẻ, tế nhị với khách, tìm hiểu lý do khách muốn huỷ và thuyết phục khách thêm một lần nữa.  
Cảm ơn khách đã thông báo cho khách sạn và hẹn dịp khác khách sạn được đón khách.  
Trong trường hợp cần thiết phải thông báo cho khách về số tiền khách phải trả cho việc huỷ bỏ đặt phòng.  
Chú ý: Nhân viên cần ghi lại đầy đủ và chính xác các thông tin sau:  
- Thay đổi hoặc huỷ bỏ đặt phòng được yêu cầu.  
- Tên người yêu cầu thay đổi hoặc huỷ bỏ đặt phòng  
- Ngày, tháng, năm thực hiện.  
- Tên nhân viên ghi nhận thông tin.  
- Sửa biểu mẫu, sơ đồ đặt phòng của khách sạn, cập nhật tình trạng đặt phòng.  
- Liên hệ với khách đặt phòng trong danh sách khách chờ nếu khách sạn có thể đáp ứng được yêu cầu của khách.

\* Một số thông tin cần khẳng định trong việc xác nhận đặt phòng của khách:  
- Tên khách hàng, địa chỉ, SĐT  
- Loại phòng, giá phòng  
- Số lượng phòng và số lượng khách.  
- Ngày giờ đến và số đêm lưu trú.  
- Các yêu cầu đặc biệt của khách.  
- Đặt phòng có đảm bảo hay không  
- Phương thức thanh toán  
- Số đặt phòng (số của phiếu đặt phòng)

Bước 3 : Cập nhật lại thông tin đặt phòng  
Mọi thông tin về đặt phòng của khách phải được cập nhật và lưu giữ đầy đủ theo thứ tự ngày đến và danh sách khách được xếp theo vần Alphabet.  
Bước 4: Tổng hợp tình hình đặt phòng   
Hằng ngày, nhân viên đặt phòng phải chuẩn bị danh sách khách dự định sẽ đến ở để chuẩn bị phòng và đón tiếp dựa vào phiếu đăng ký trước đó và số lượng khách đi, phòng sẽ trả trong ngày để nhân viên thông báo và chuẩn bị thủ tục hóa đơn trả phòng cho khác, đồng thời cũng thống kê số lượng khách và phòng mà khách đang sử dụng trong ngày hôm đó, nhằm mục đích giúp cho khách sạn nắm được số lượng khách và nhân viên Lễ tân chuẩn bị hồ sơ cho khách 1 cách tốt nhất.  
Lưu ý : Khi nhận đặt phòng của khách đoàn, hệ thống cần xử lý được các công việc :  
- Xác định rõ số phòng và loại phòng được đặt.  
- Ghi rõ các dịch vụ khách sạn cung cấp cho khách theo thoả thuận.  
- Những yêu cầu chi tiết về ăn uống (nếu có)  
- Thời gian đặt cọc, số tiền đặt cọc, phương thức thanh toán.  
- Hạn cuối cùng khẳng định lại số lượng phòng và danh sách khách trong đoàn cho khách sạn.  
- Ghi lại tên, điạ chỉ, SĐT, fax, email của người đặt.  
- Để giữ được nguồn khách lâu dài nên chú ý đến chính sách khuyến mại đặc biệt đối với khách đi theo đoàn.  
- Kiểm tra thông tin trước đây của đoàn khách này đã từng ở khách sạn gồm: số phòng báo huỷ, số phòng đặt nhưng không tới và những thay đổi.

+ Quản lý hiện trạng từng phòng :

Việc quản lý hiện trạng phòng được nhân viên phục vụ kiểm tra trong quá trình dọn phòng.

+ Quản lý các dịch vụ mà khách sạn cung cấp:

Các khách hàng muốn sử dụng dịch vụ nào đó sẽ được nhân viên lễ tân ghi vào phần dịch vụ sử dụng riêng và những dịch vụ này khi thanh toán sẽ được tính vào tiền sử dụng dịch vụ.

+ Quản lý phòng, nhân viên:

Các phòng được phân loại và quản lý theo khu vực, theo tầng và các nhân viên khách sạn cũng được quản lý theo khu vực làm việc.

• Quy trình hoạt động của hệ thống:

- Khi có khách hàng đăng ký việc thuê phòng trên website của khách sạn thì bộ phận tiếp tân sẽ nhận sau đó gọi điện thoại xác nhận với khách hàng, kiểm tra những phòng trống phù hợp với yêu cầu của khách hoặc thay đổi phòng hay dịch vụ nếu như khách hàng yêu cầu thay đổi dựa trên khả năng phục vụ của khách sạn, từ đó cập nhật lại phiếu đăng ký, cập nhật thông tin về khách hàng vào hệ thống quản lý.

- Nhân viên bảo vệ sẽ tiếp nhận giữ xe, khuân hành lý và dẫn khách lên nhận phòng.

- Khi khách có nhu cầu sử dụng thêm dịch vụ, nhân viên lễ tân sẽ ghi vào phần dịch vụ sử dụng riêng và những dịch vụ này khi thanh toán sẽ được tính vào tiền sử dụng dịch vụ khi khách hang thanh toán trả phòng.

- Nhân viên phục vụ sẽ hàng ngày lên dọn dẹp phòng hoặc phục vụ nhu cầu ăn uống, giặt là khi khách yêu cầu.

**3.2 PHÂN TÍCH THIẾT KẾ HỆ THỐNG QUẢN LÝ KHÁCH SẠN**

## 3.2.1 Biểu đồ ngữ cảnh của hệ thống

Cho ta cái nhìn tổng quan về hệ thống khách sạn nó gồm 3 phần: hệ thống quản lý khách sạn, 2 nhân tố khách và ban quản lý, các luồng dữ liệu đi từ tác nhân tới hệ thống và ngược lại .



## 3.2.2 Lập sơ đồ chức năng kinh doanh – BFD



• Quản lý tài nguyên

Khách sạn có các tài nguyên cần quản lý : phòng , loại phòng, tiện nghi trong loại phòng, các tiện nghi của khách sạn, quản lý giá theo gói dịch vụ phòng, quản lý các dịch vụ mà khách sạn cung cấp như: dịch vụ giặt là, dịch vụ ăn uống.

• Quản lý phòng đặt

Cho phép quản lý các thông tin về các cuộc đặt phòng của khách như tên khách, giới tính, địa chỉ liên lạc, thời gian đến và đi, số tiền đặt cọc, số phòng, loại phòng, các dịch vụ yêu cầu …

• Quản lý khách vào

Nhân viên lễ tân cung cấp các thông tin về phòng trống trong khách sạn cho khách thông qua việc truy vấn máy tính vào kho dữ liệu danh sách phòng, khi khách xác định thuê phòng hệ thống sẽ lưu các thông tin vào cơ sở dữ liệu. Đối với những khách đã đặt trước thì chuẩn bị thủ tục đón khách, chuẩn bị phòng để đón khách và cập nhật thông tin cho phiếu đặt phòng.

• Quản lý khách ra

Dựa vào phiếu đặt để quản lý những khách đến thời hạn trả phòng từ đó lập hóa đơn thanh toán cho khách hàng, nếu trường hợp khách muốn ở tiếp thì cập nhật lại phiếu thuê phòng.

• Quản lý sử dụng dịch vụ

Khi khách trả phòng nhân viên phục vụ phòng sẽ vào dọn dẹp phòng và kiểm tra các dịch vụ trong phòng mà khách đã sử dụng từ đó tính vào chi phí sử dụng dịch vụ của khách hàng, trong quá trình ở nếu khách có yêu cầu sử dụng dịch vụ thì nhân viên sẽ them thông tin về dịch vụ mà khách sử dụng để khi thanh toán trả phòng sẽ tính vào chi phí sử dụng dịch vụ của khách.

Đặt phòng qua mạng

- Sau khi truy vấn trên mạng khách hàng có thể lựa chọn việc đặt phòng tại khách sạn thông qua mạng. Để có thể đặt phòng khách hàng sẽ được cung cấp một tài khoản với email của khách hàng đã có password

- Việc đặt phòng qua mạng được tiến hành như sau: khách sẽ đưa ra các thông tin liên quan đến cuộc đặt phòng: số lượng phòng đặt, số khách tới checkin, số lượng phòng đơn, phòng đôi, phòng tập thể, ngày checkin, ngày checkout, ngày đặt phòng phải trước ngày trả phòng.

## 

## 3.2.3 Biểu đồ luồng dữ liệu của hệ thống khách sạn :

### 3.2.4 DFD mức 0



### 3.2.5 Biểu đồ luồng dữ liệu chức năng quản lý đặt phòng.



Mô tả chức năng : Chức năng quản lý đặt phòng cho phép quản lý các thông tin về các cuộc đặt phòng của khách. Thông tin về các cuộc đặt bao gồm : thông tin về khách như: tên khách giới tính, địa chỉ liên lạc, số điện thoại, số fax …thông tin đặt phòng bao gồm : số phòng, ngày đến, giờ đến, ngày đi … Khi khách có yêu cầu đặt phòng thì hệ thống sẽ tra cứu thông tin về phòng trống sau khi kiểm tra được phòng trống thì hệ thống sẽ bắt đầu lưu thông tin phòng được đặt, thông tin về gói phòng khách chọn, thông tin cá nhân của khách hàng để tiện cho tra cứu và quản lý thông tin sau đó sẽ thông báo với khách là khách đặt phòng thành công hay thất bại do hết phòng trống.

### 3.2.6 Biểu đồ luồng dữ liệu mức 1 chức năng quản lý khách vào (Check in)

Mô tả chức năng : Khách hàng đã đặt trước khi đến nhận phòng thì khách phải đưa thông tin liên quan đến cuộc đặt phòng. Nhân viên lễ tân sẽ tra cứu thông tin về cuộc đặt phòng đó trên hệ thống và làm các thủ tục cập nhật về thông tin khách ở hướng dẫn khách hàng đến phòng ở. Thong tin check in bao gồm thông tin về khách và thông tin khách thuê.



### 3.2.7 Biểu đồ luồng dữ liệu mức 1 chức năng khách sử dụng dịch vụ



Mô tả chức năng : Khi khách hàng có nhu cầu sử dụng thêm các dịch vụ ngoài dịch vụ trọn gói theo gói dịch vụ mà khách hàng đã chọn thì nhân viên sẽ trả lời yêu cầu khách hàng. Các thông tin về việc sử dụng dịch vụ ngoài gói dịch vụ của khách sẽ được lưu lại vào cơ sở dữ liệu và được tính toán tự động vào số tền khách hàng phải trả khi lưu trú tại khách sạn.

### 

### 3.2.8 Biều đồ luồng dữ liệu mức 1 chức năng quản lý khách trả phòng check out:



Mô tả chức năng : Khi khách có yêu cầu trả phòng bộ phận tra cứu sẽ có trách nhiệm kiểm tra thông tin thông qua bộ phận quản lý sử dụng dịch vụ để lấy thông tin về số tiền khách sử dụng dịch vụ ngoài gói dịch vụ kèm theo phòng. Thông tin này sẽ được kết hợp với số tiền thuê phòng của khách để đưa qua bộ phận thanh toán để thanh toán với khách hàng. Sau khi thanh toán bộ phận check out sẽ cập nhật tình trạng phiếu đặt phòng vừa trả.

### 3.2.9 Biểu đồ luồng dữ liệu phân rã chức năng mức 1 chức năng quản lý tài nguyên :

Mô tả chức năng :

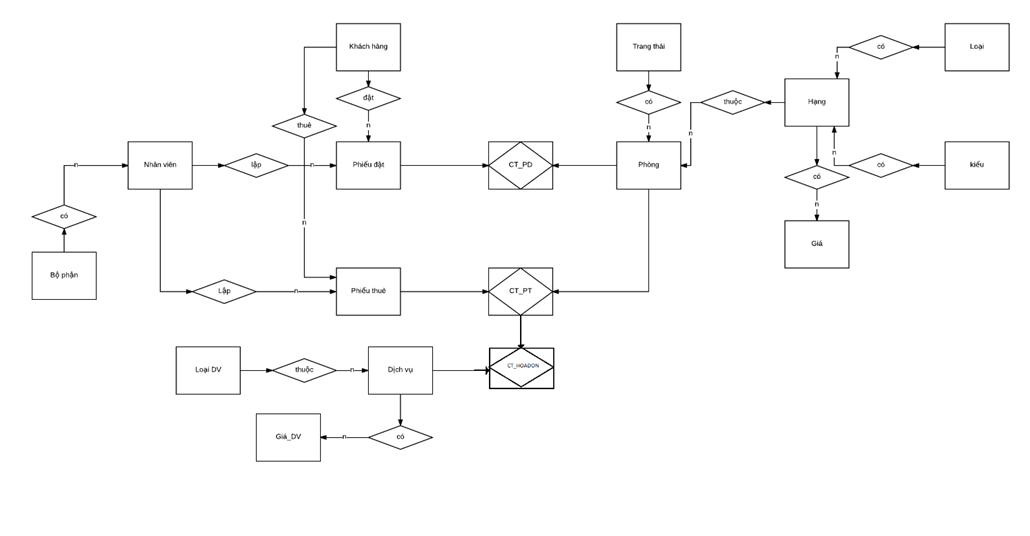
Quản lý Phòng : với yêu cầu thường xuyên cập nhật thông tin Phòng.

Quản lý Loại Phòng : với yêu cầu thường xuyên cập nhật thông tin Loại Phòng.

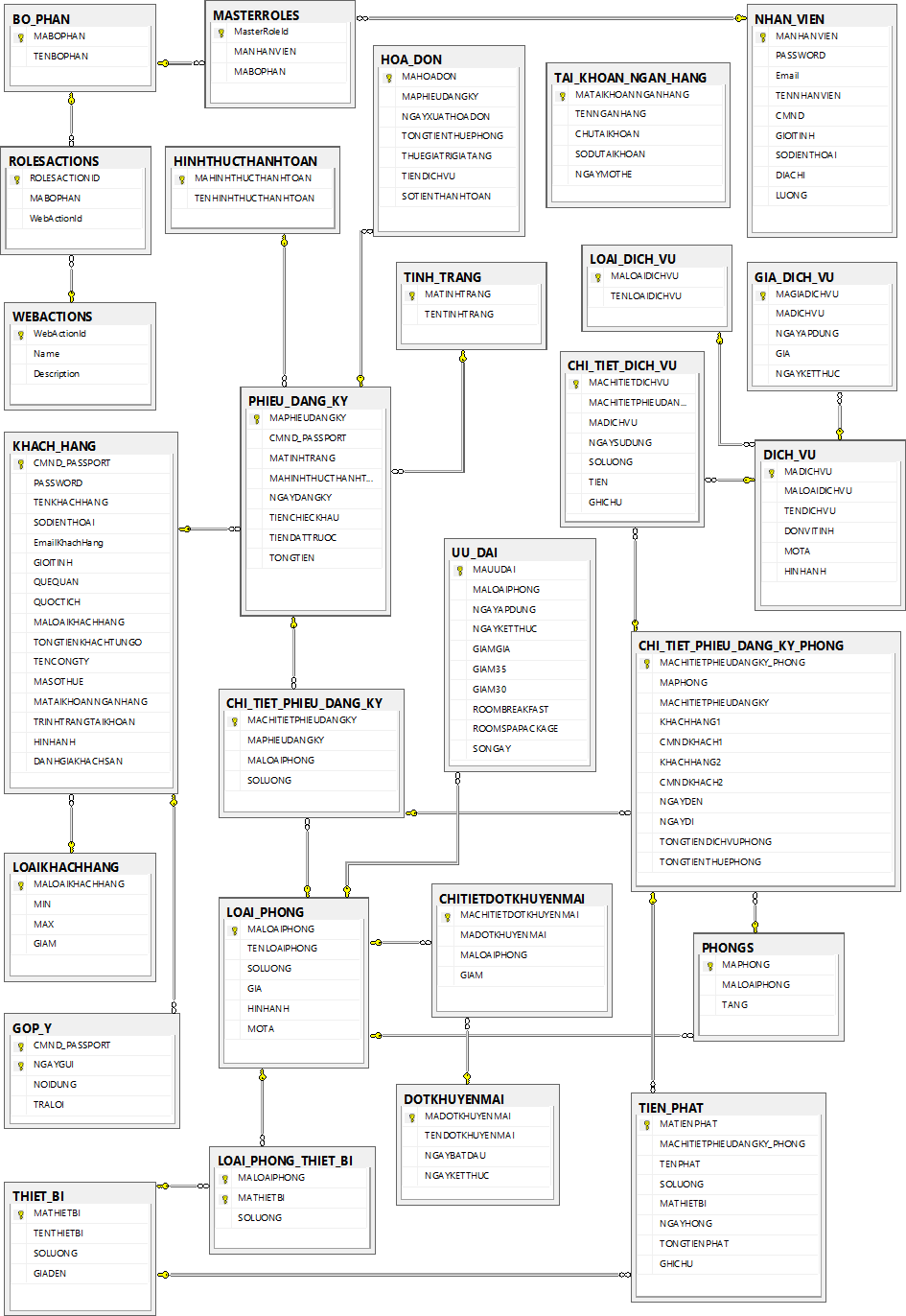
Quản lý Dịch vụ : với yêu cầu thường xuyên cập nhật thông tin Dịch vụ.



## 3.2.10 Mô hình thực thể kết hợp ERD

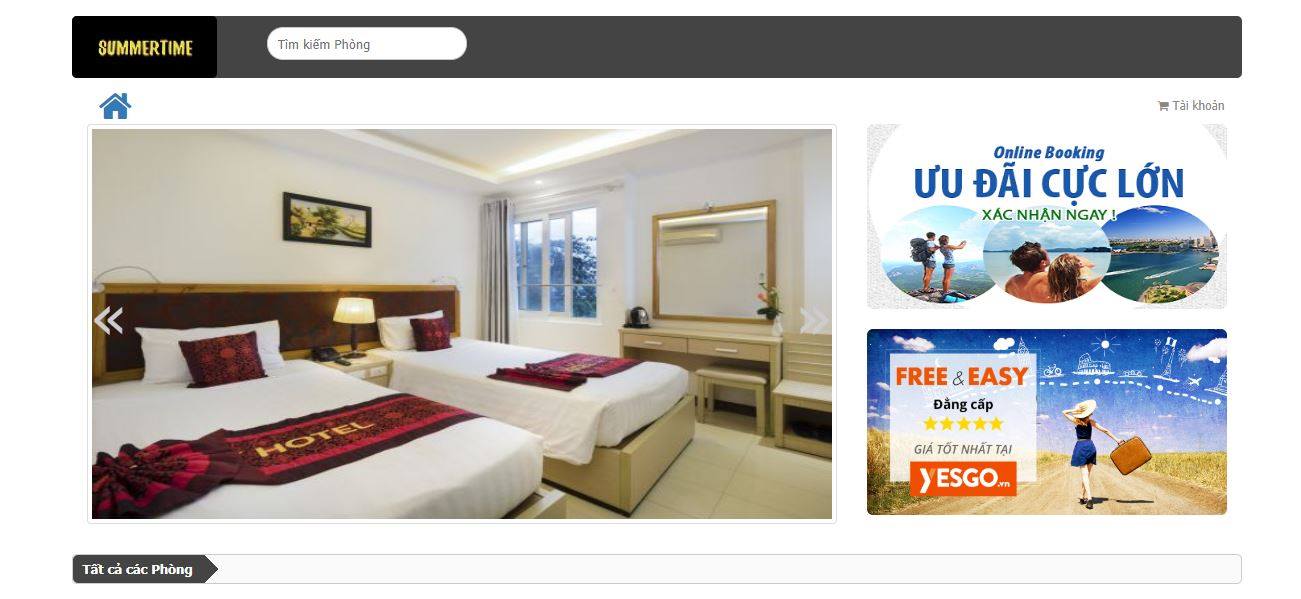


## 3.2.11 Mô hình Diagram

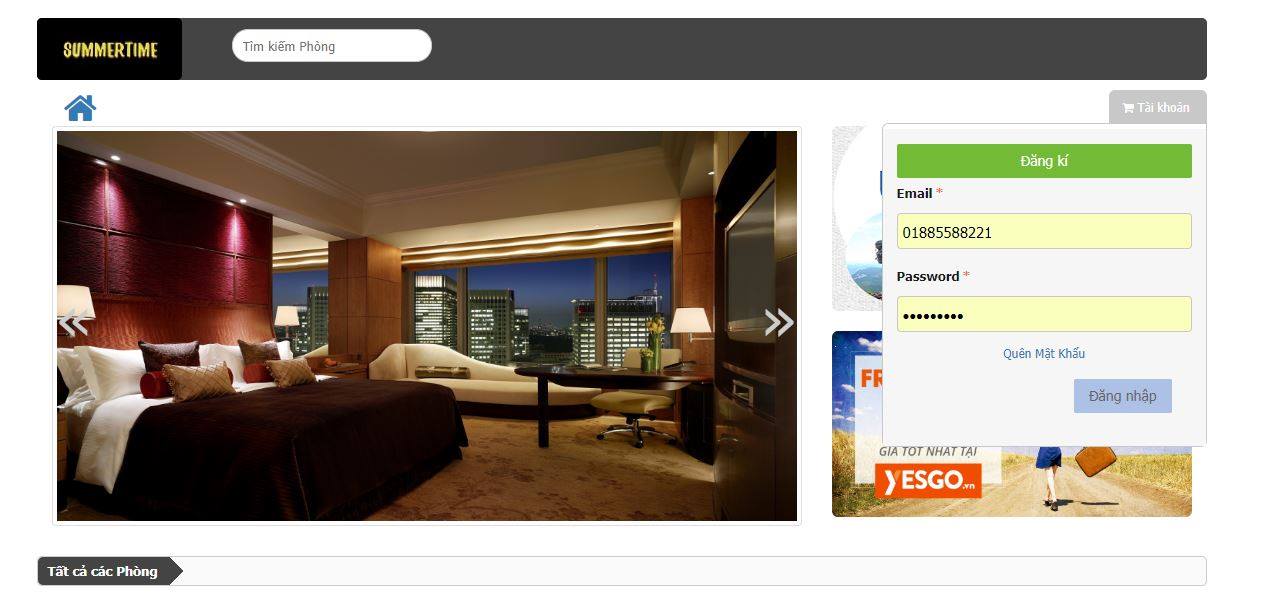


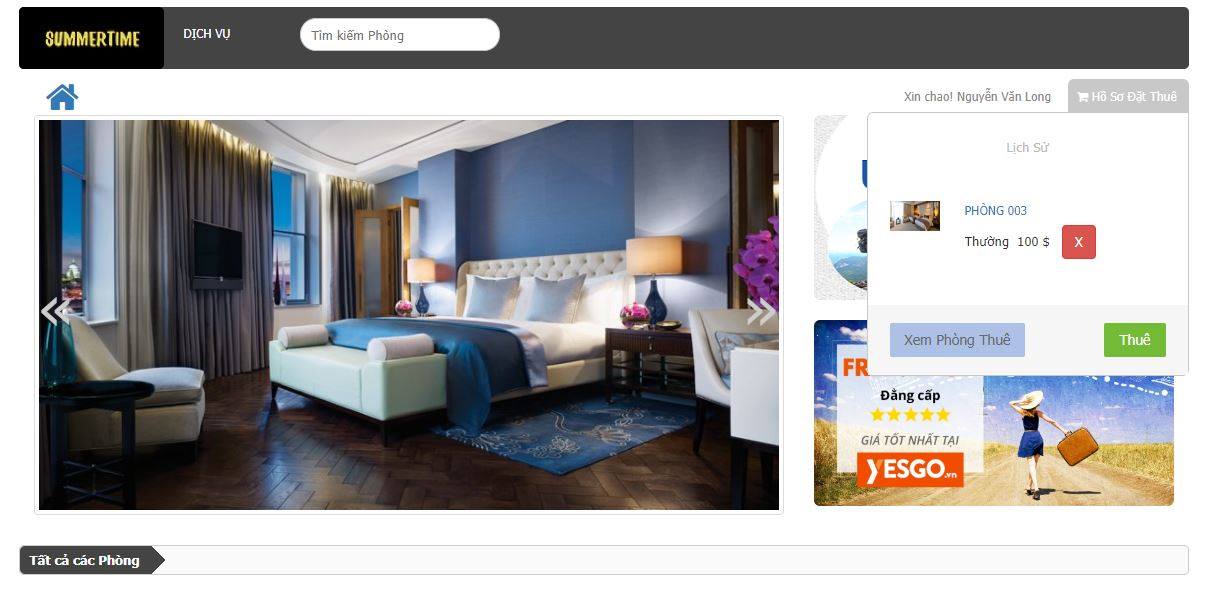
**3.3 : THIẾT KẾ GIAO DIỆN**

### 3.3.1 Trang chủ Khách Hàng



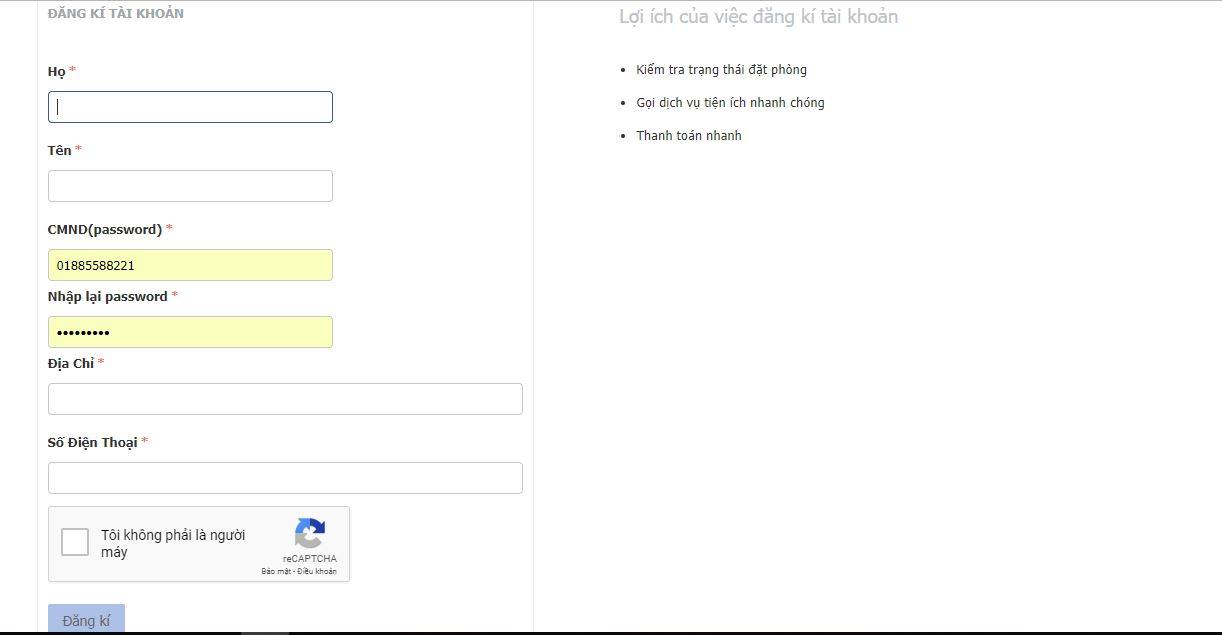
Hình 07: Trang chủ

Hình 08: Phòng Vip



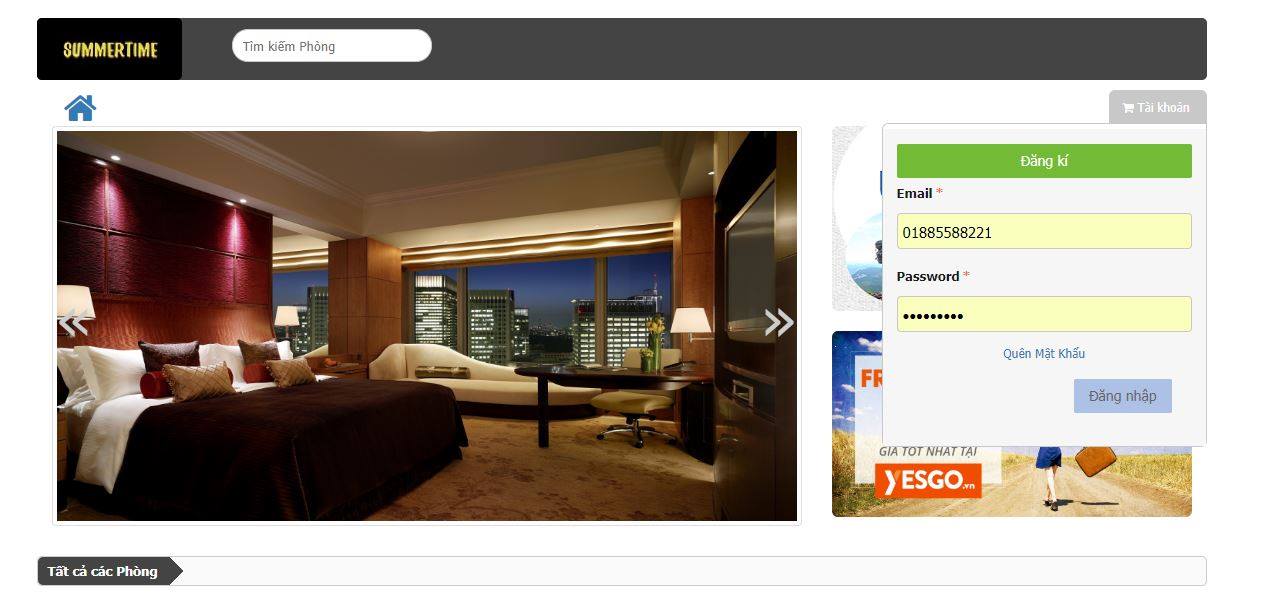
Hình 09 : Phòng hạng sang

* **Đăng ký tài khoản khách hàng**



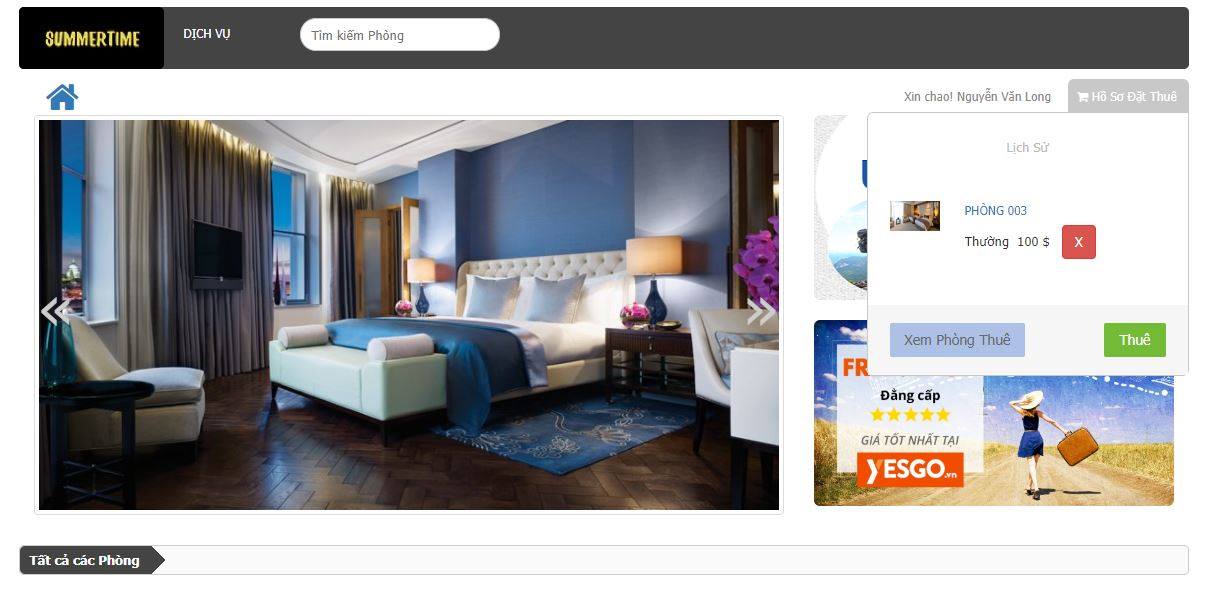
Hình 10 : Đăng ký tài khoản cho khách hàng

* **Đăng nhập tài khoản khách hàng**



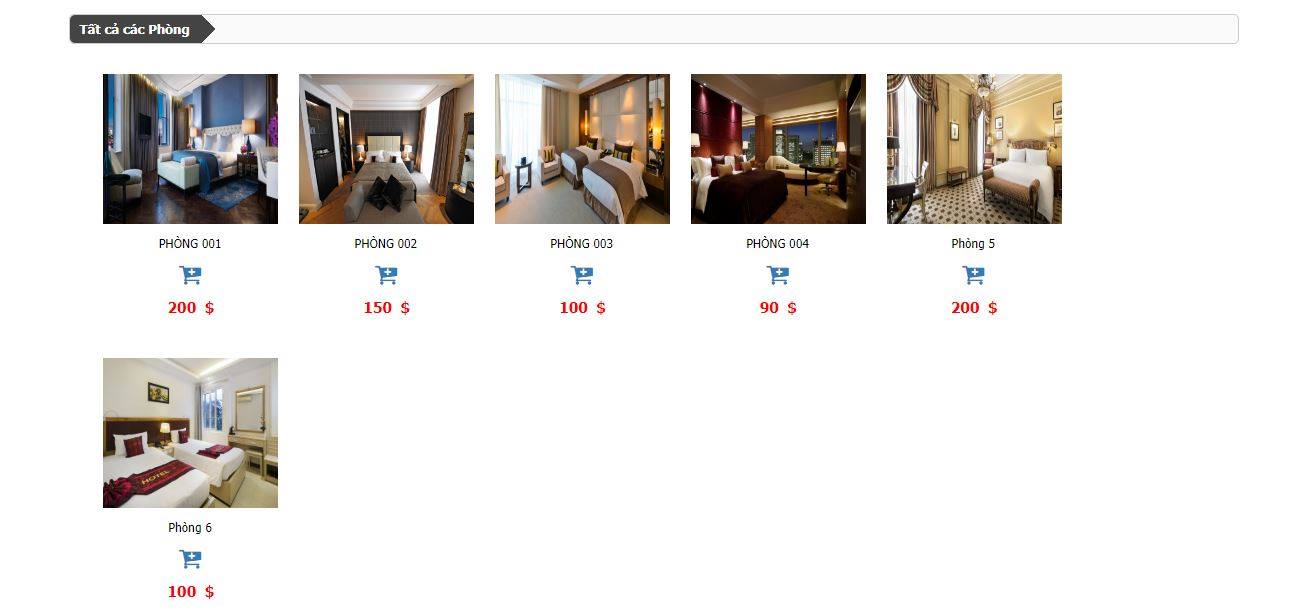
Hình 11 : Đăng nhập tài khoản khách hàng

* **Đặt phòng**



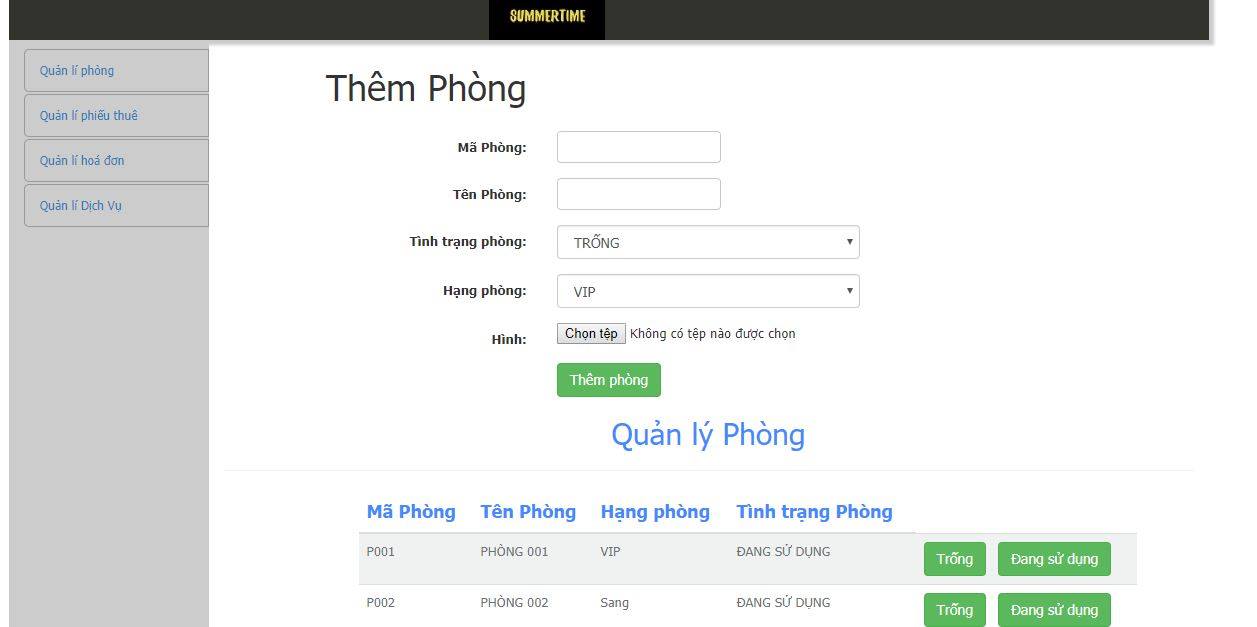
Hình 12: Khách hàng chọn phòng theo ý muốn.

* **Danh sách Phòng**



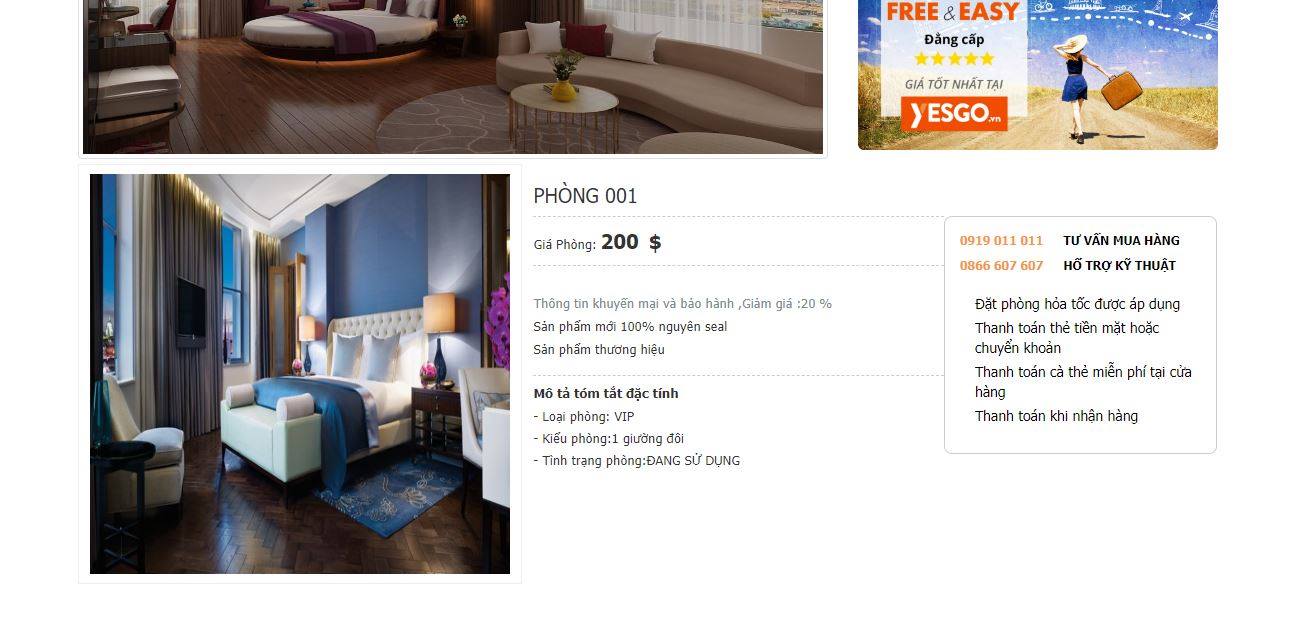
Hình 13: Danh sách phòng

* + **Quản lý phòng**



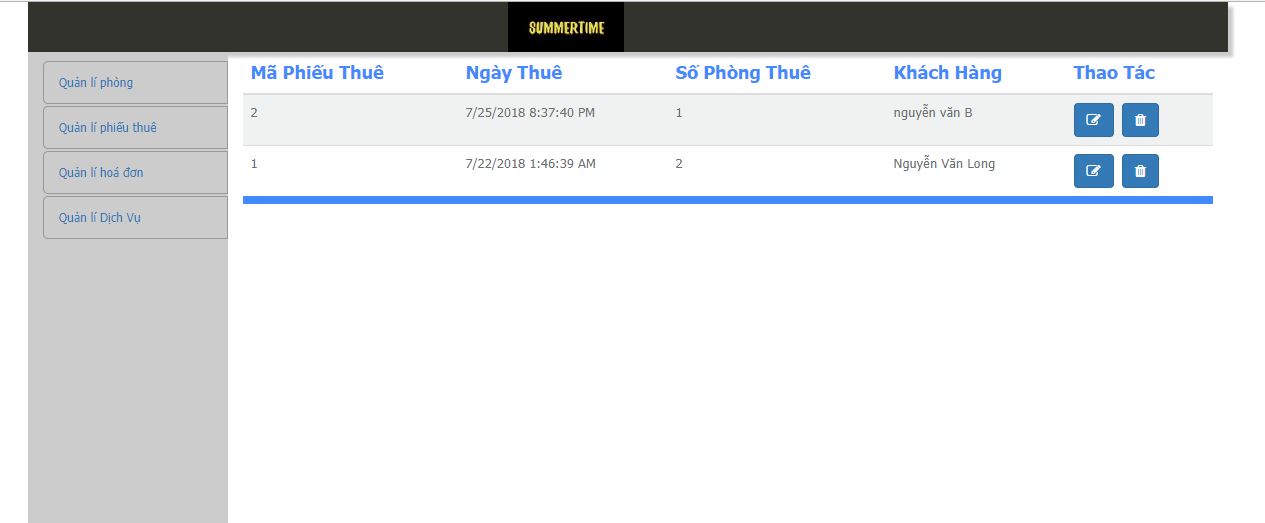
Hình 14: quản lý phòng

* **Chi tiết phòng**



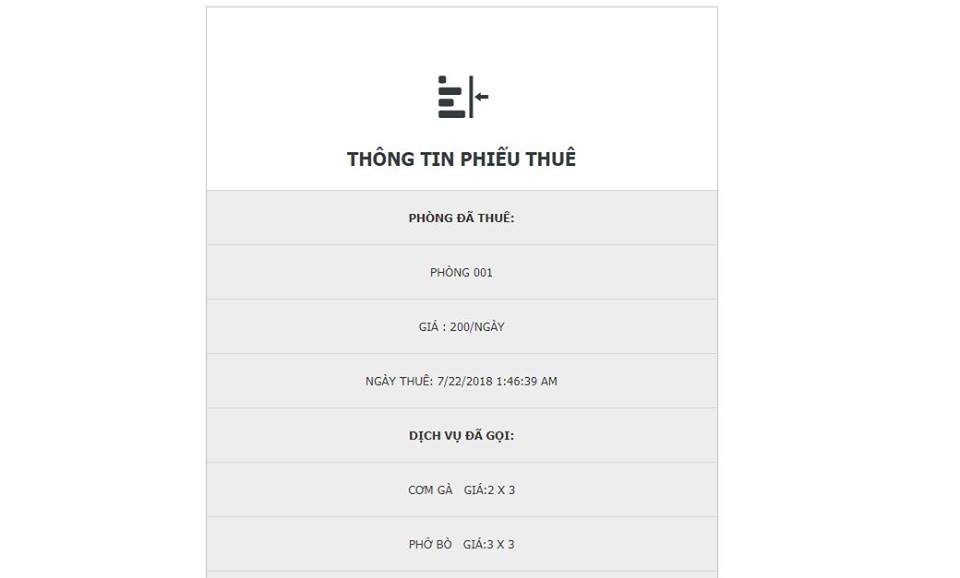
Hình 15: Chi tiết phòng .

* + **Phiếu Thuê**



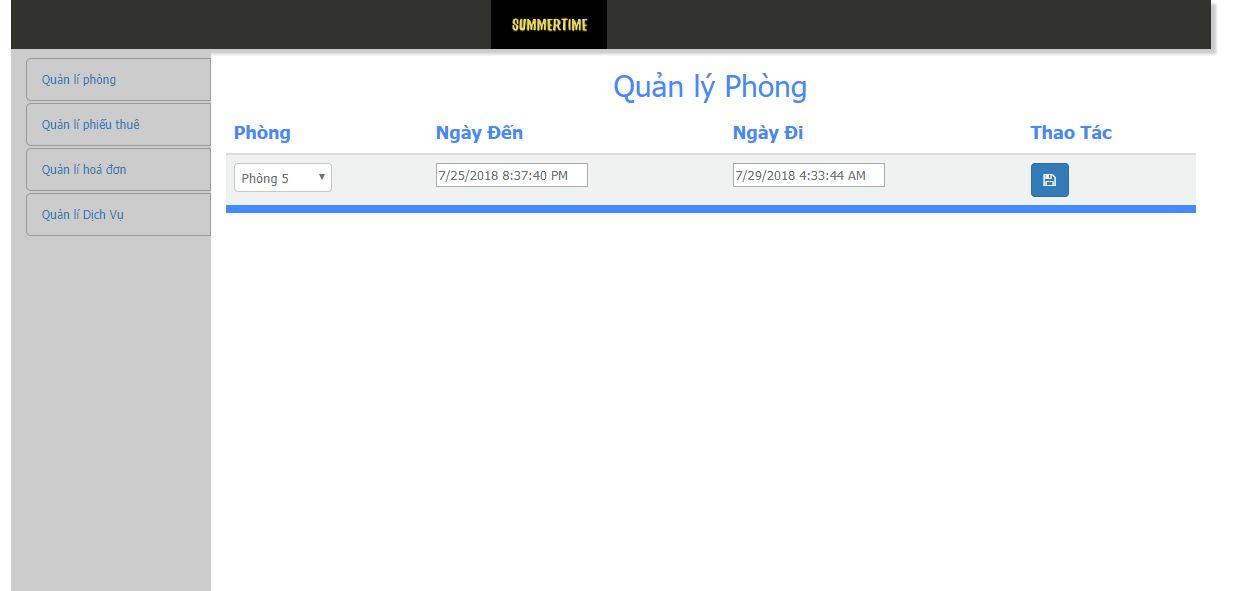
Hình 16 : phiếu thuê

* + **Chi tiết Phiếu thuê**



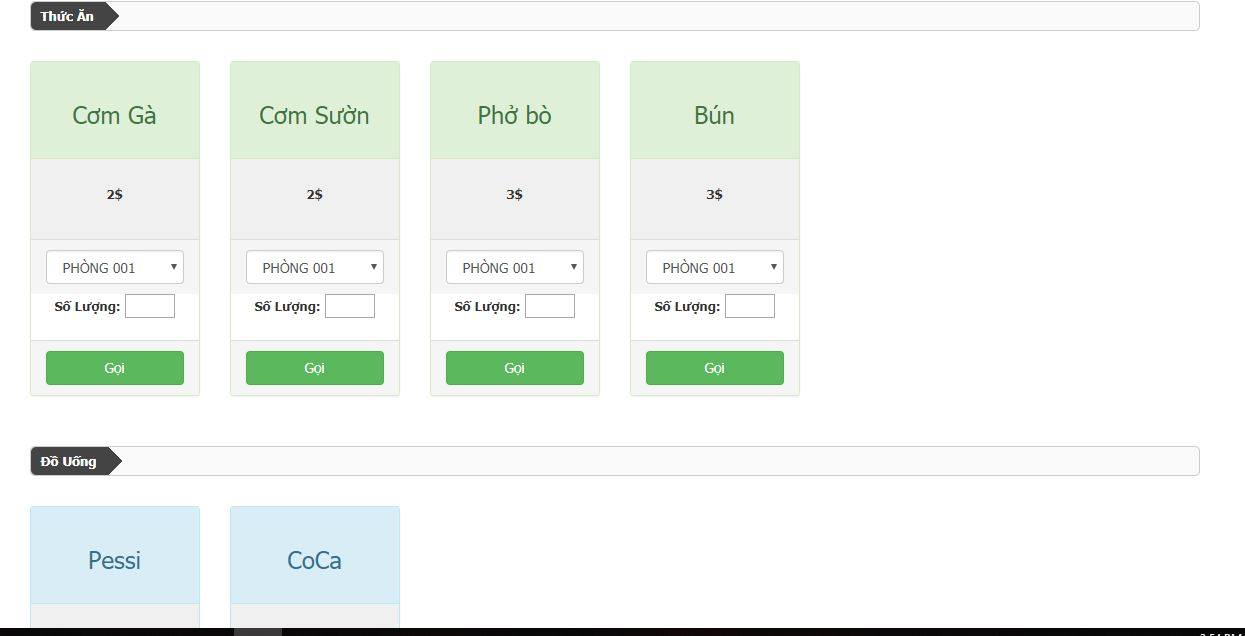
Hình 17: Chi tiết phiếu thuê

* + **Quản lý Chi tiết Phiếu thuê**



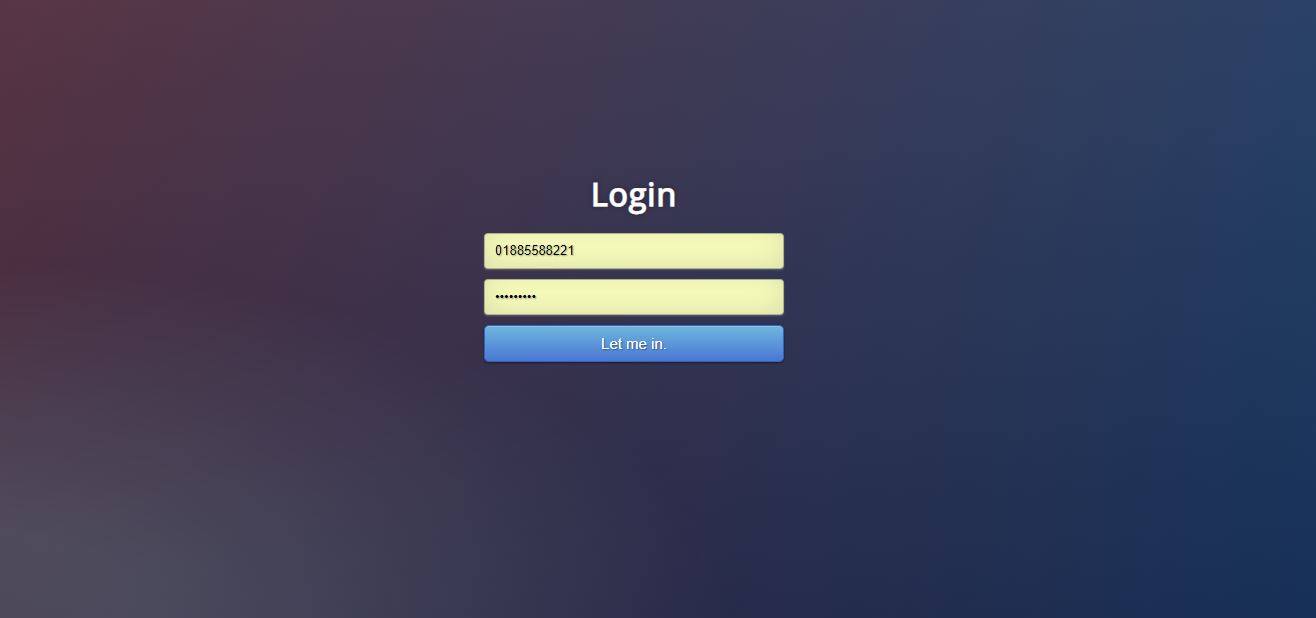
Hình 18 : quản lý chi tiết phiếu thuê

* + **Dịch vụ**



Hình 19: các dịch vụ

## 3.3.2 trang admin.



Hình 20 : Trang chủ Admin.

**CHƯƠNG 4 : KẾT QUẢ THỰC HIỆN**

**4.1 Quy trình chạy chương trình web Quản lý Khách sạn**

**4.1.1 Quy trình xử lý:**

Lập phiếu thuê phòng :

*Khi khách đến khách sạn:*

*+ Đối với khách đã đặt phòng từ trước: khi yêu cầu nhận phòng phải đưa thông tin liên quan tới cuộc đặt phòng của mình. Nhân viên lễ tân sẽ tìm kiếm thông tin đặt phòng trên máy tính, xác nhận lại các thông tin và lưu vào cơ sở dữ liệu. Thông tin bao gồm thông tin về khách và thông tin phòng khách đặt.*

*+ Còn với khách chưa đặt phòng trước: có yêu cầu thuê phòng, nhân viên lễ tân sẽ cung cấp thông tin về các phòng trống hiện có cho khách thông qua việc truy vấn vào kho dữ liệu danh sách phòng. Khách xác nhận thuê phòng, nhân viên lễ tân sẽ lưu thông tin vào cơ sở dữ liệu.*

**ĐÓN TIẾP**

**XÁC NHẬN VIỆC**

**ĐẶT PHÒNG**

**LÀM THỦ TỤC**

**ĐĂNG KÝ THUÊ PHÒNG**

**Đã đặt chỗ trước**

**TRA CỨU TÌNH TRẠNG PHÒNG**

**Chưa đặt chỗ trước**

**LẬP PHIẾU THUÊ PHÒNG**

**Có phòng trống**

***Quy trình phục vụ khách hàng trong thời gian khách lưu trú tại khách sạn:***

Trong thời gian khách lưu trú tại khách sạn, khi khách có yêu cầu sử dụng dịch vụ, khách sạn sẽ cung cấp các dịch vụ phục vụ nhu cầu của khách. Bộ phận lễ tân thu thập các phiếu sử dụng của khách hàng để tiến hành lập và tổng hợp lại các khoản chi phí của khách.

Sơ đồ quy trình phục vụ khách

**NHẬN YÊU CẦU SỬ DỤNG DỊCH VỤ**

**CUNG CẤP DỊCH VỤ**

**LƯU THÔNG TIN SỬ DỤNG DỊCH VỤ**

***Quy trình làm thủ tục và thanh toán cho khách:***

Khi khách có yêu cầu trả phòng, bộ phận lễ tân có nhiệm vụ thông báo cho các bộ phận khác trong khách sạn về việc trả phòng của khách nhằm xác định các khoản chưa thanh toán. Khi khách trả phòng, nhân viên lập hoá đơn tổng hợp và tiến hành thanh toán với khách.

Cuối mỗi ngày bộ phận lễ tân sẽ viết báo cáo tổng hợp về tình hình khách đến, khách đi, thực trạng phòng trống, phòng bận cũng như thu chi của ngày cũng như ý kiến phản hồi của khách… cho ban giám đốc khách sạn.

Sơ đồ quy trình làm thủ tục và thanh toán cho khách:

**NHẬN YÊU CẦU**

**TRẢ PHÒNG**

**XÁC NHẬN THÔNG TIN**

**TRẢ PHÒNG**

**LƯU THÔNG TIN TRẢ PHÒNG**

**TÍNH TIỀN PHÒNG VÀ TIỀN DỊCH VỤ**

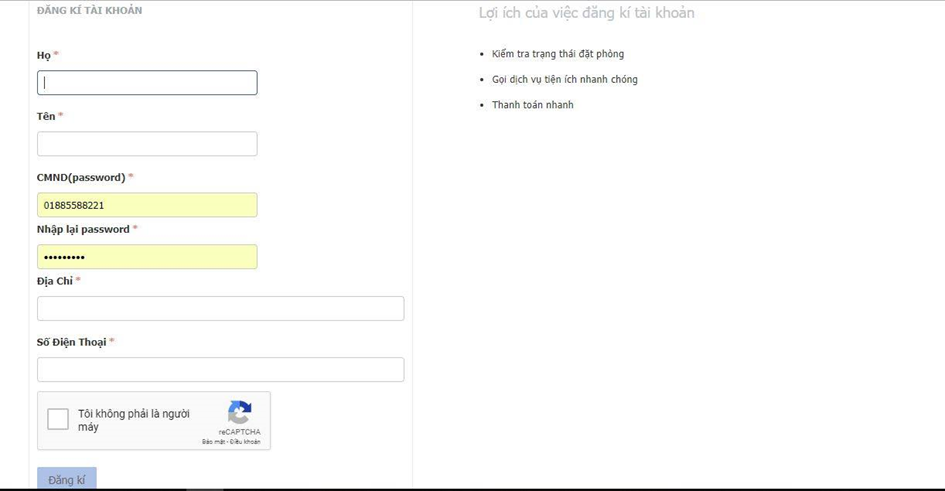
**4.1.2 Chương trình Quản lý Khách sạn**

*Chương trình dành cho khách hàng*

* Trang chủ

Form Trang chủ bao gồm các chức năng đăng ký, đăng nhập, đổi mật khẩu, kiểm tra các đơn hàng đã đặt, các menu lệnh, chế độ đa ngôn ngữ bao gồm tiếng anh và tiếng việt để đáp ứng nhu cầu sử dụng của khách hàng trong và ngoài nước ta chỉ cần nhấp vào chế độ ta muốn hệ thống sẽ tự động cập nhật thay đổi ngôn ngữ theo ý muốn của khách hàng, ngoài ra nó còn giới thiệu sơ lược về các loại phòng, tiện nghi, các chính sách ưu đãi khi khách hàng đến với khách sạn. Ngoài ra nó còn cho phép khách hàng chọn thời gian đến và đi từ đó để hệ thống tìm kiếm các phòng phù hợp cho giai đoạn khách ở. Khi click vào các loại phòng hệ thống sẽ chuyển trang cho khách hàng đến trang chi tiết loại phòng đó

* Đăng ký tài khoản:

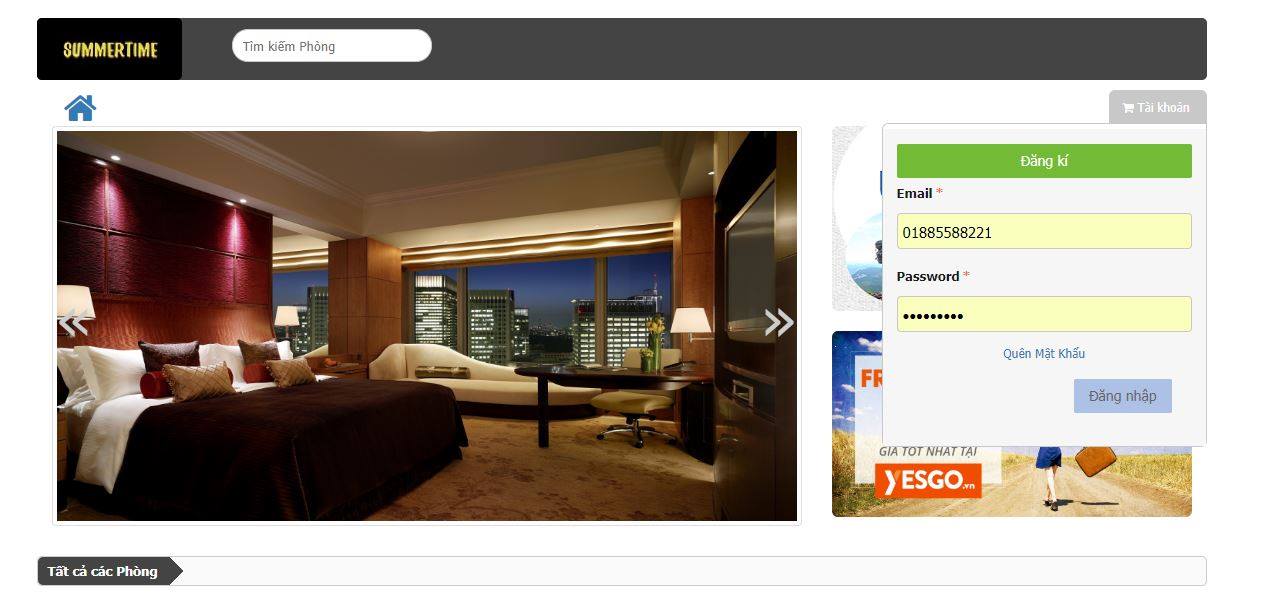


Khách hàng lần đầu đặt phòng sẽ được yêu cầu đăng ký 1 tài khoản cá nhân.

Tài khoản này sẽ giúp khách hàng dễ dàng cập nhật mọi thông tin về khách sạn, các dịch vụ cũng như những thông tin để đặt phòng, sử dụng dịch vụ và thanh toán.

Ở trang này, khách hàng sẽ phải cung cấp một số thông tin cá nhân : họ, tên, số chứng minh nhân dân ( mặc định số chứng minh nhân dân phải có 9 số) ,số điện thoại (mặc định 10-11 số) và địa chỉ khách hàng ( tỉnh/thành phố mà khách hàng sinh sống hoặc làm việc )

* Đăng nhập tài khoản



Sau khi đăng ký tài khoản thành công, khách hàng sẽ sử dụng tài khoản đó để đăng nhập những lần tiếp theo, và sau này. Việc này sẽ tạo thông tin khách hàng thân thiết, sẽ nhận được nhiều ưu đãi khuyến mại của Khách san.

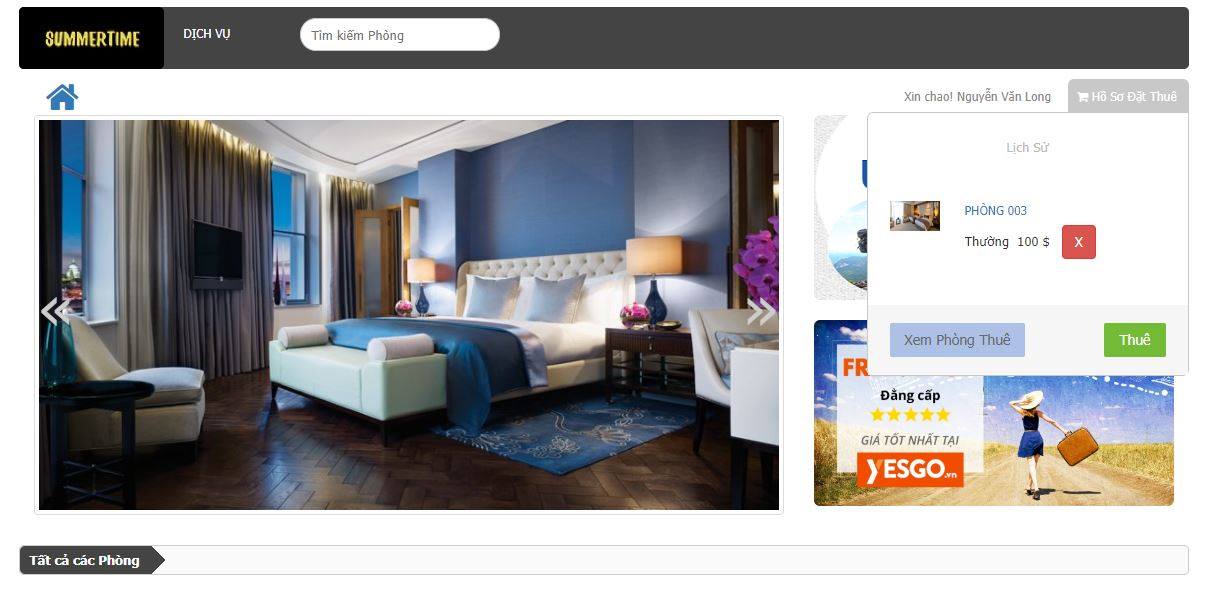
Thông tin đăng nhập gồm:

Tài khoản: là số điện thoại khi khách hàng đăng ký trước đó

Mật khẩu sử dụng là số chứng minh nhân dân của khách hàng

* **Đặt phòng**

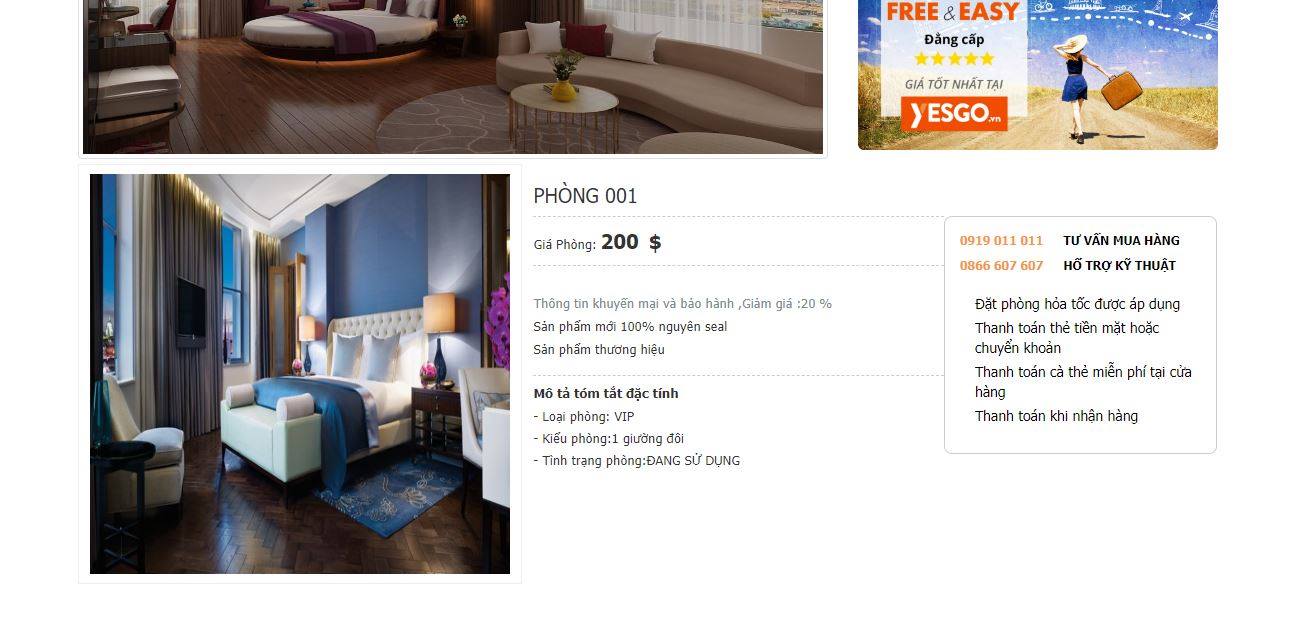
Sau khi đăng nhập thành công, hệ thống sẽ hiển thị Họ tên của khách hàng, khi đó khách hàng có thể xem thông tin chi tiết về các phòng nghỉ, các dịch vụ cũng như các tiện nghi, giá cả… Khách hàng kích chọn vào các hình ảnh sẽ hiển thị mọi thông tin mà khách hàng đang tìm kiếm: phòng đơn, phòng đôi, phòng tập thể, tiện nghi gồm những gì ( điều hòa, ti vi, tủ lạnh, máy nóng lạnh,…), hạng phòng (vip,sang, trung bình ), giá cả,….



Hình 7.6 Khách hàng chọn phòng theo ý muốn.

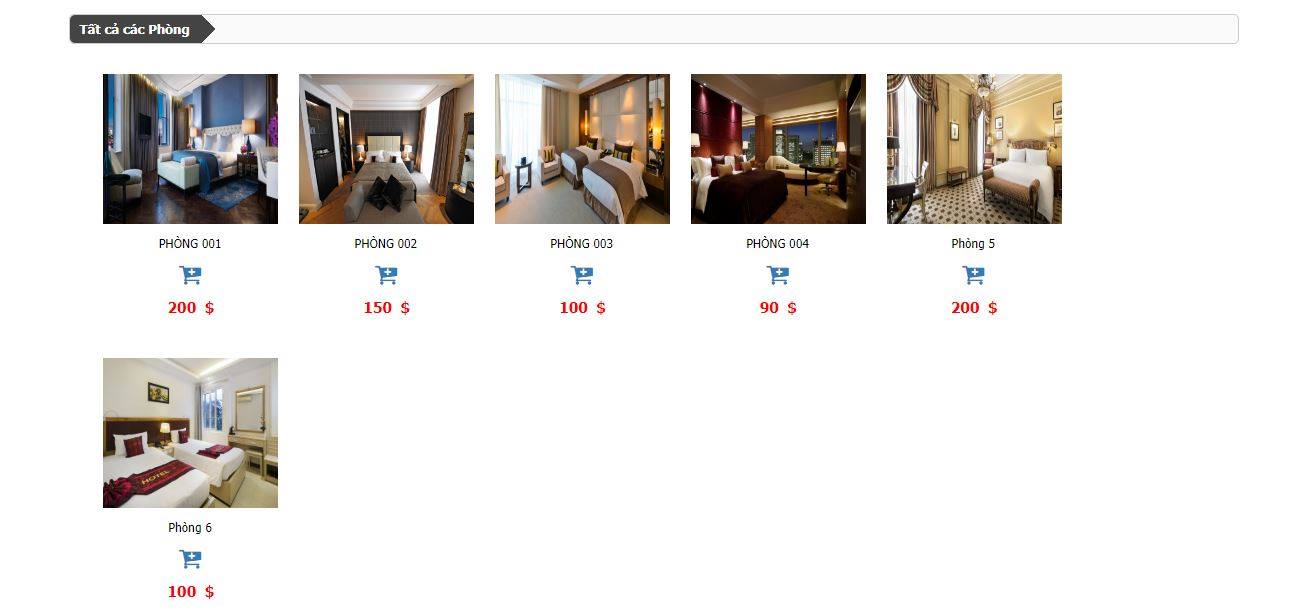
Khi khách hàng đã chọn ngày đến và ngày đi xong và nhấp vào nút đặt ngay thì hệ thống sẽ chuyển trang cho khách hàng đến trang này để khách hàng chọn các phòng theo ý muốn của khách hàng.

* **Chi tiết phòng**



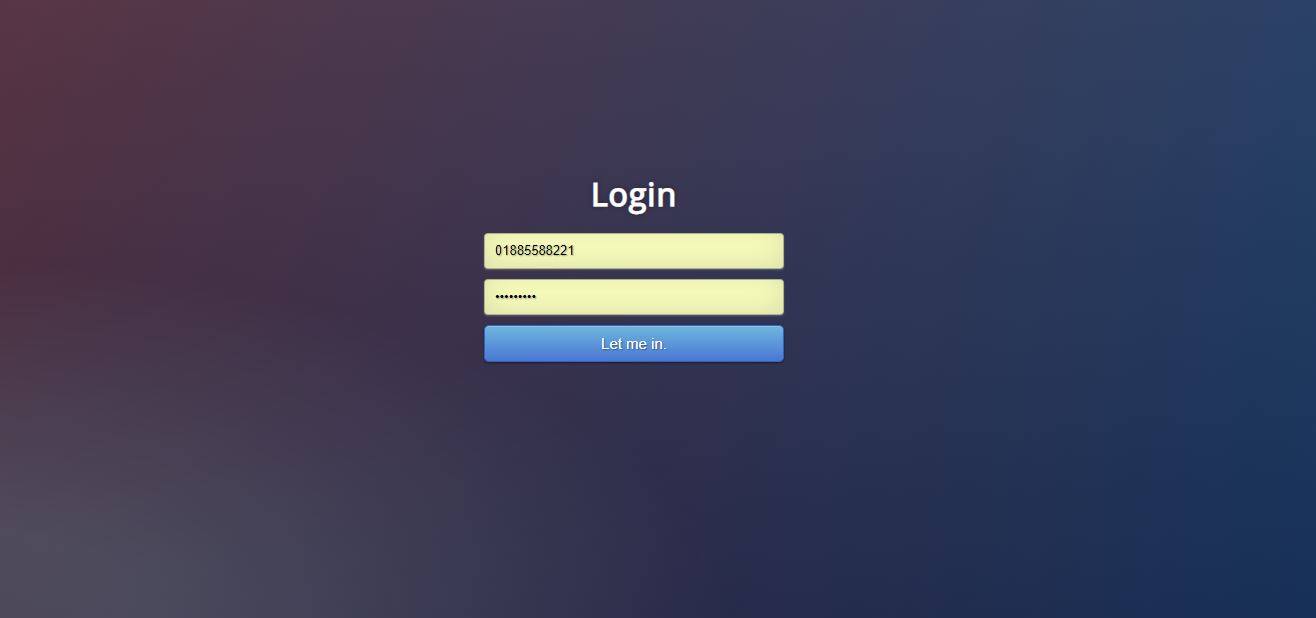
Hình 7.8 Chi tiết phòng

* **Danh sách Phòng**

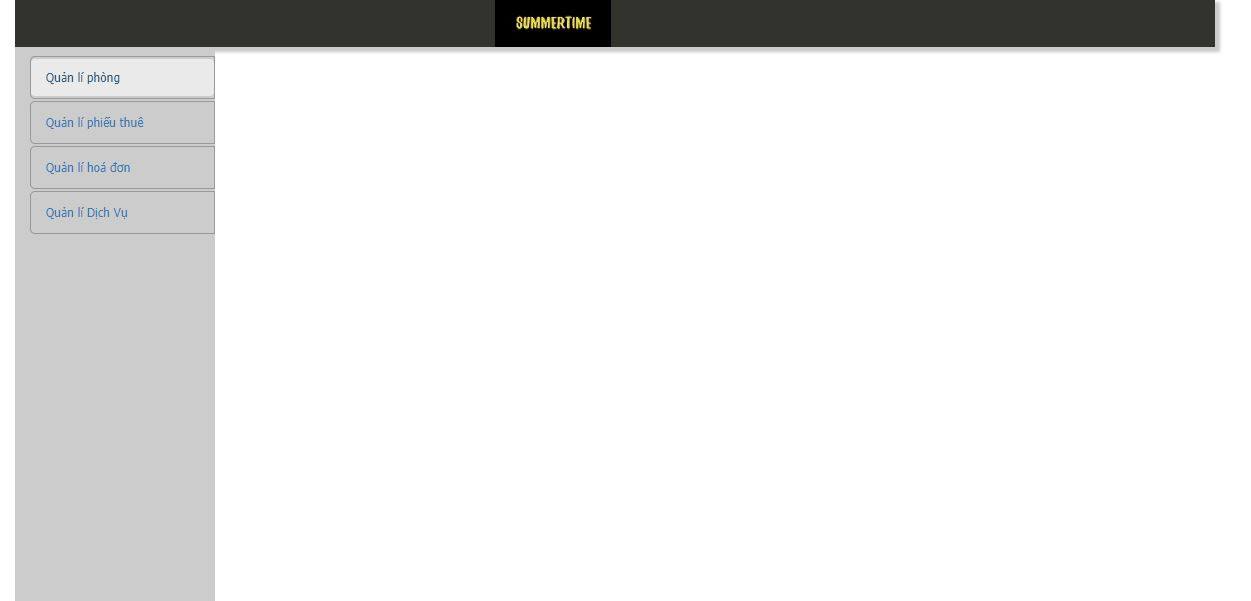


Trong quá trình kiểm tra khách hàng có thể loại bỏ từng phòng mà khách hàng không muốn đặt nữa qua lệnh Remore của phòng đó hoặc Clear để xóa toàn bộ các phòng trong giỏ hàng để trả về cho khách hàng giỏ phòng trống hoặc Update để cập nhật giỏ hàng. Nếu khách hàng đã đồng ý với các phòng mà mình đã chọn thì có thể nhấn vào nút Checkout hệ thống sẽ tự động kiểm tra khách hàng đã đăng nhập hay chưa nếu chưa đăng nhập thì hệ thống sẽ yêu cầu khách hàng đăng nhập, nếu khách hàng chưa có tài khoản hệ thống sẽ hỗ trợ khách hàng đăng ký tài khoản thông qua đường dẫn đăng ký nằm ở đầu trang web, nếu khách hàng đã đăng nhập hệ thống sẽ lấy thông tin khách hàng ra thông tin lại lần nữa các thông tin về phiếu đặt phòng cũng như thông tin về các phòng mà khách hàng đã chọn đặt.

***7.2 Hướng dẫn sử dụng Website bên Admin.***



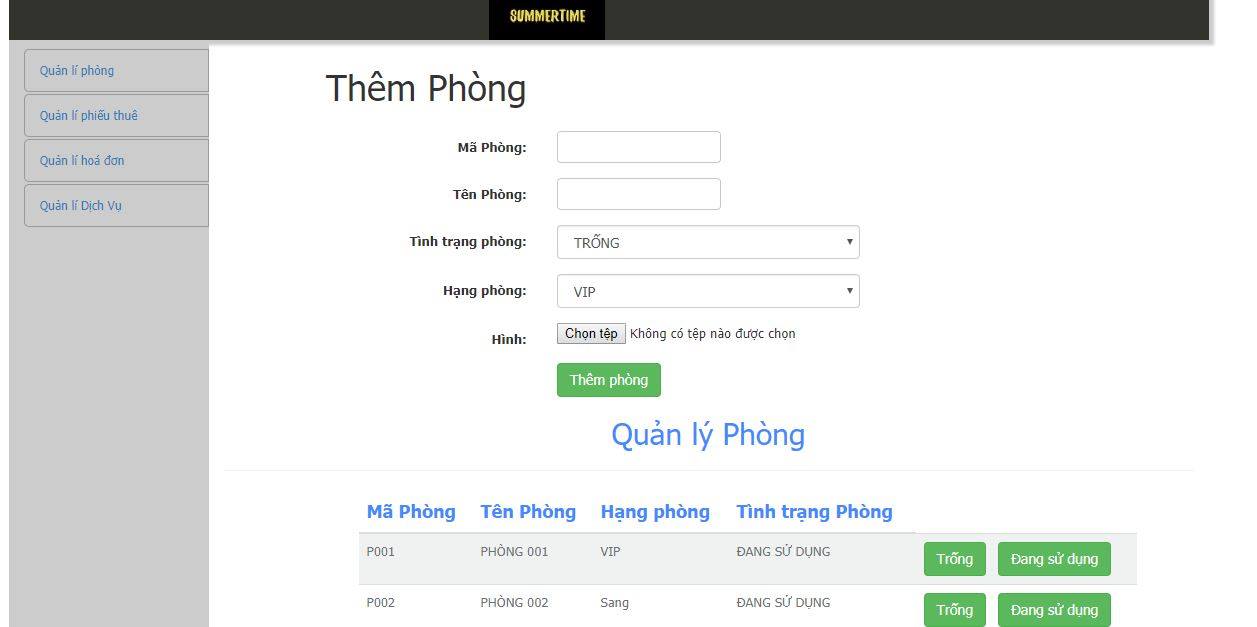
Hình: Trang chủ Admin.



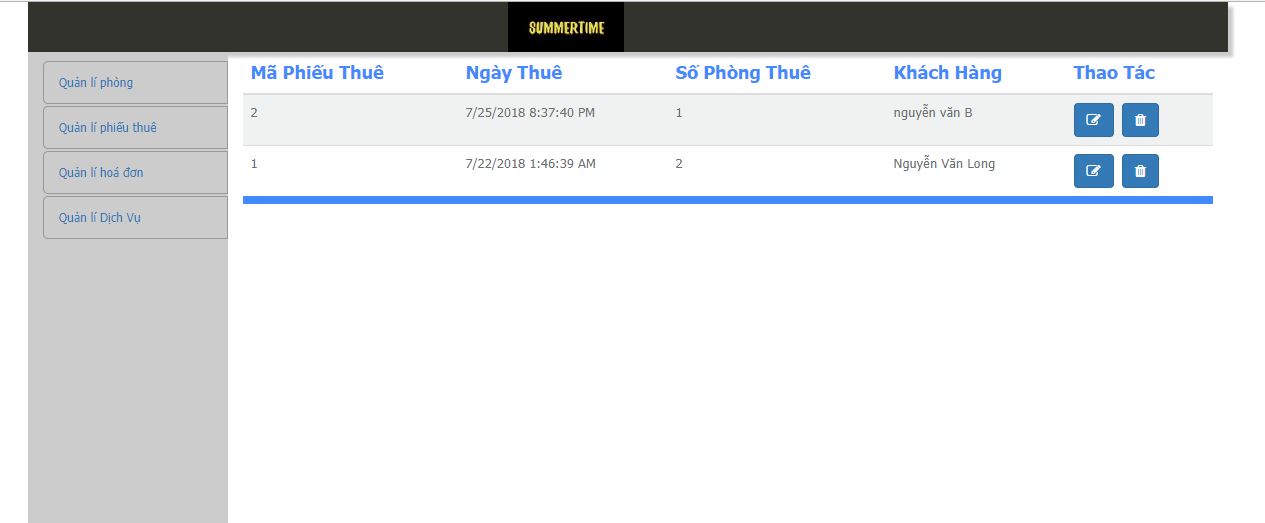
Hình 7.10 Trang chủ Admin

From này giúp các nhân viên đăng nhập vào hệ thống Admin để vào thực hiện các tác vụ, lưu ý rằng chỉ các tài khoản của nhân viên được phân đúng quyền thì nhân viên đó mới có thể vào và thực hiện chức năng đó. Việc phân quyền người dùng được thực hiện trên bộ phận, các nhân viên trong cũng 1 bộ phận có thể có các quyền như nhau.

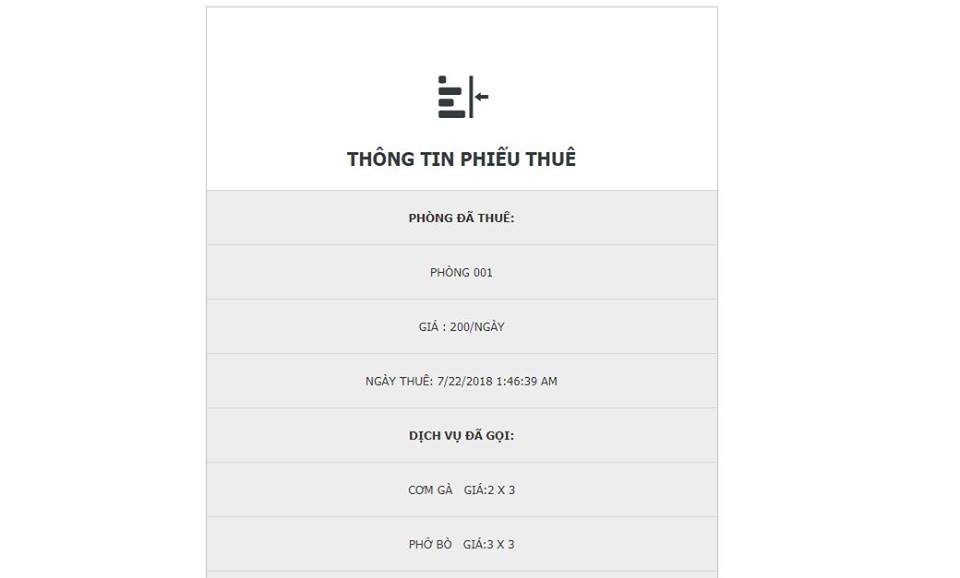
* **Quản lý phòng**



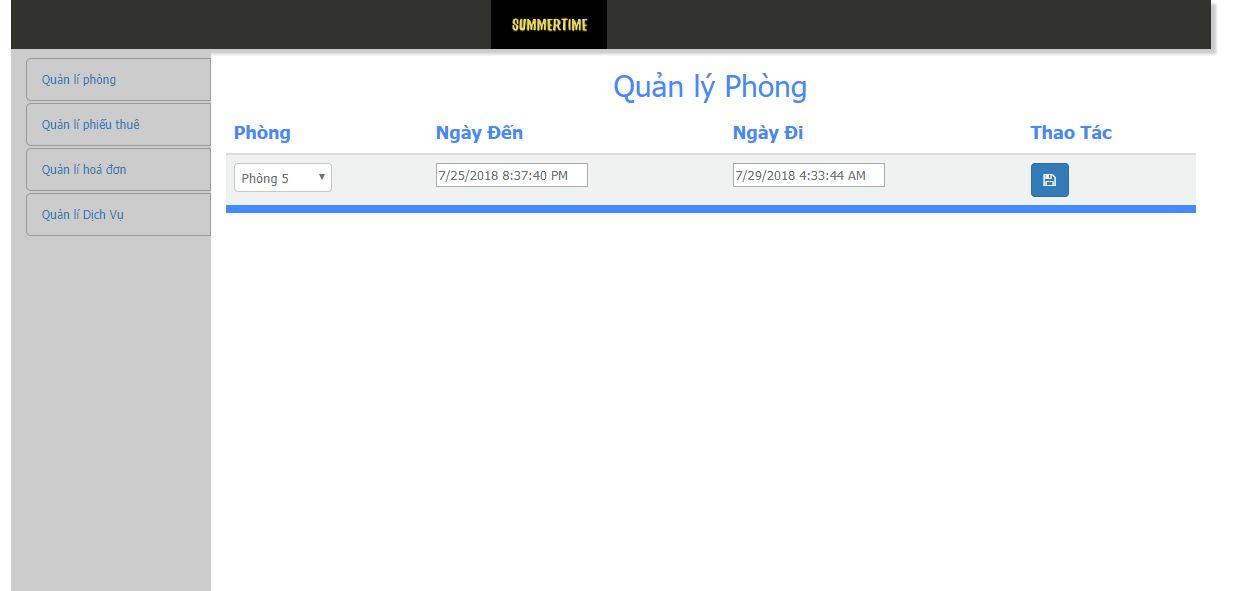
* **Phiếu Thuê**

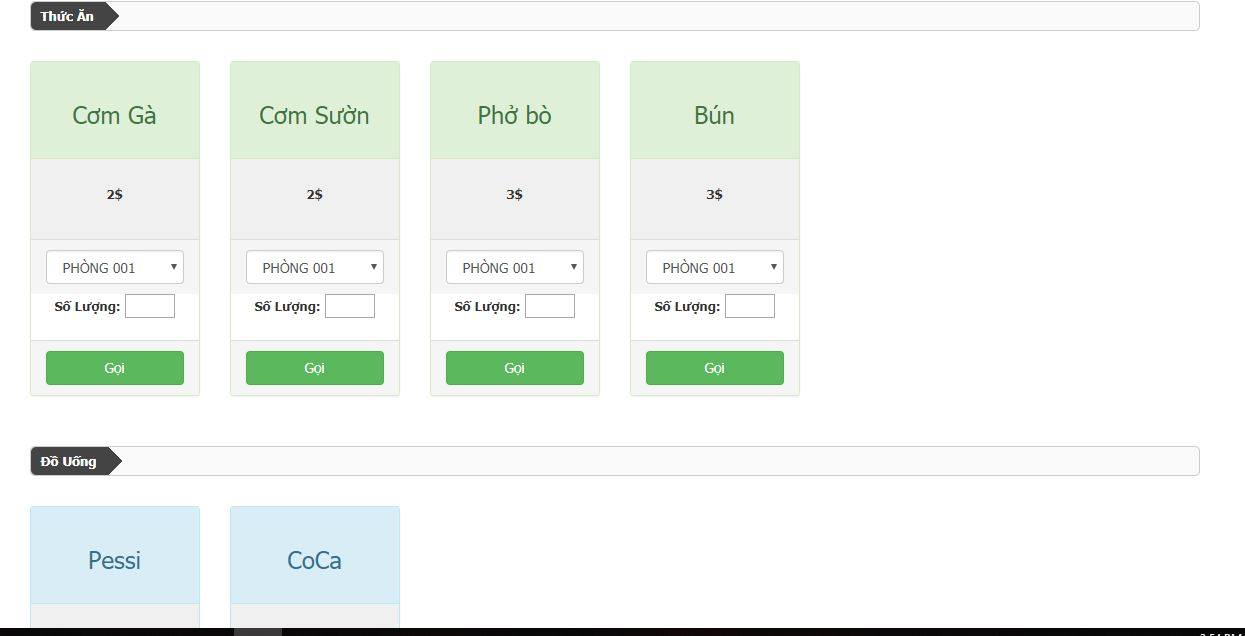


* **Chi tiết Phiếu thuê**



* **Quản lý Chi tiết Phiếu thuê**



* + **Dịch vụ**

## 

## Chương 5 : kết luận

## 5.1 Kết quả đạt được:

+ Xây dựng thành công Website phía Khách hàng và phía Admin

+ Cho phép khách đặt phòng qua mạng tự động tìm kiếm phòng trống gán vào phiếu đặt của khách.

+ Cho phép khách hàng xóa phòng đã đặt khi chưa đến ở, cũng như thay đôi thời gian lưu trú và phòng ở tại khách sạn.

+ Quản lý Phòng.

+ Quản lý nhân viên

+ Ghi nhận thông tin khách hàng thuê phòng

+ Ghi nhận đăng ký đặt phòng

+ Ghi nhận thuê phòng (check in)

+ Tra cứu phòng qua sơ đồ phòng

+ Ghi nhận việc sử dụng dịch vụ (minibar, giặt ủi, điện thoại, thức ăn, các dịch vụ khác)

+ Tính tiền khi trả phòng – check out ( lập hóa đơn )

+ Cập nhật giá phòng, giá dịch vụ

* *Ưu điểm*

Tránh được những điều sau:

• Mất nhiều thời giờ

• Nhân viên phải ghi chép sổ sách rất nhiều, và thỉnh thoảng lại bị nhầm lẫn

• Sự chậm trễ trong quá trình liên lạc giữa các bộ phận nghiệp vụ trong khách sạn, cùng với việc tra cứu thông tin chậm trễ hoặc không chính xác có thể làm lỡ các cơ hội cho thuê phòng. Bất kỳ sai sót nào dù nhỏ đều có thể khiến uy tín khách sạn bị giảm sút, một điều không bao giờ bạn mong đợi…

• Ngoài ra giúp các nhà quản lý cập nhật thông tin, công việc dễ dàng tạo thuận lợi cho vệc quản lý

* *Nhược điểm*

• Cũng như phần lớn các doanh nghiệp trong nước vẫn giữ hình thức tổ chức và quản lý theo hướng chức năng (khai thác hiệu quả của từng bộ phận riêng rẽ) chỉ có một số doanh nghiệp chuyển đổi theo hình thức quản lý liên kết các bộ phận chức năng( khai thác hiệu quả do liên kết giữa các bộ phận chức năng)

• Chi phí cho việc ứng dụng thường lớn và việc thay, ứng dụng công nghệ phần lớn phụ thuộc vào tư duy, phương pháp lãnh đạo, áp lực cạnh tranh nên không đồng bộ cho toàn hệ thống

• Hệ thống còn nhiều thiếu sót do quá trình làm đề tài gấp rút, hiểu biết về nghiệp vụ Quản lý Khách sạn còn hạn chế, chưa thực tế, e rất mong nhận được sự thông cảm từ quý Thầy Cô.

**5.2 Hướng phát triển :**

Cho phép khách hàng đặt phòng theo các thời gian khác nhau trong cùng 1 phiếu đăng ký.

Cải tiến tối ưu hóa câu lệnh để hệ thống chạy mượt hơn và nhanh hơn.

Hỗ trợ thanh toán trực tuyến.

Mở các tour du lịch cho các địa điểm thăm quan nổi tiếng gần khách sạn.

Mở rộng xây dựng thêm phần nhà hàng, các gói dịch vụ Spa vào khách sạn …

# TÀI LIỆU THAM KHẢO

<https://spring.io/>

<http://stackoverflow.com>

<https://www.w3schools.com/>

<https://www.tutorialspoint.com>

<https://www.hoteljob.vn/tin-tuc/quy-trinh-lam-viec-cua-nhan-vien-dat-phong-khach-san>

<http://khanhspring.com/tong-quan-ve-hibernate/>

<http://congdongjava.com/forum/threads/gi%E1%BB%9Bi-thi%E1%BB%87u-webservice.509/>