BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**



**BÁO CÁO ĐỒ ÁN PHÁT TRIỂN ỨNG DỤNG WEB**

**ĐỀ TÀI: XÂY DỰNG WEBSITE QUẢN LÝ QUÁN KARAOKE**

|  |  |
| --- | --- |
| **Giảng viên:** | **Bùi Chí Thành** |
| **Nhóm thực hiện:** | **Nguyễn Dương Anh Tuấn – 62133366**  **Hồ Hoàng Kha – 62130808**  **Nguyễn Đăng Khoa – 62130863** |

Khánh Hòa, tháng 6 năm 2023

BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC NHA TRANG**

**KHOA CÔNG NGHỆ THÔNG TIN**

****

**BÁO CÁO ĐỒ ÁN PHÁT TRIỂN ỨNG DỤNG WEB**

**ĐỀ TÀI: XÂY DỰNG WEBSITE QUẢN LÝ QUÁN KARAOKE**

|  |  |
| --- | --- |
| **Giảng viên:** | **Bùi Chí Thành** |
| **Nhóm thực hiện:** | **Nguyễn Dương Anh Tuấn – 62133366**  **Hồ Hoàng Kha – 62130808**  **Nguyễn Đăng Khoa – 62130863** |

Khánh Hòa, tháng 6 năm 2023

MỤC LỤC

[MỤC LỤC 3](#_Toc136645985)

[CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI 4](#_Toc136645986)

[1.1 Lý do chỌn đỀ tài: 4](#_Toc136645987)

[A. Mục tiêu của đề tài: 4](#_Toc136645988)

[1.2 GiỚi hẠn và phẠm vi cỦa đỀ tài: 4](#_Toc136645989)

[1.3 Tính khẢ thi: 4](#_Toc136645990)

[1.4 KHẢO SÁT HIỆN TRẠNG: 4](#_Toc136645991)

[A. Mô tả địa điểm khảo sát 4](#_Toc136645992)

[B. Hiện trạng tổ chức 5](#_Toc136645993)

[C. Hiện trạng nghiệp vụ: 5](#_Toc136645994)

[D. Hiện trạng tin học: 7](#_Toc136645995)

[CHƯƠNG 2. CỞ SỞ LÝ THUYẾT 8](#_Toc136645996)

[2.1 TỔng quan vỀ ngôn ngỮ Asp.NET MVC 8](#_Toc136645997)

[A. Giới thiệu Asp.NET MVC framework 8](#_Toc136645998)

[B. Lý do chọn Asp.NET MVC 12](#_Toc136645999)

[2.2 TỔNG QUAN vỀ SQL SERVER 13](#_Toc136646000)

[A. Khái niệm hệ quản trị CSDL SQL Server 13](#_Toc136646001)

[B. Đặc điểm của SQL Server và đối tượng làm việc 13](#_Toc136646002)

[CHƯƠNG 3: NỘI DUNG THỰC HIỆN 14](#_Toc136646003)

[3.1 DANH SÁCH CHỨC NĂNG YÊU CẦU CHỨC NĂNG NGHIỆP VỤ 14](#_Toc136646004)

[A. Nhà quản lý 14](#_Toc136646005)

[B. Bộ phận kho 15](#_Toc136646006)

[C. Bộ phận thu ngân: 15](#_Toc136646007)

[3.2 Danh sách yêu CẦU chức năng HỆ THỐNG 16](#_Toc136646008)

[A. Danh sách yêu cầu phi chức năng 17](#_Toc136646009)

[B. Yêu cầu đặc tả phần mềm 17](#_Toc136646010)

[ThiẾt kẾ hỆ thỐng 22](#_Toc136646011)

[A. Thiết kế dữ liệu 22](#_Toc136646012)

CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI

* 1. Lý do chỌn đỀ tài:

Cùng với sự phát triển của cuộc cách mạng công nghiệp 4.0, công cuộc xây dựng và đổi mới đất nước theo hướng công nghiệp hóa – hiện đại hóa,cùng với đó là sự ra đời của các trang thương mại điện tử phục vụ cho nhu cầu mua sắm online, giao dịch trực tuyến. Tận dụng lợi thế của các trang thương mại điện tử trong việc quảng bá sản phẩm, thương hiệu cung ứng đầy đủ, toàn diện và chi tiết các loại sản phẩm và về đến khách hàng, nên chúng em chọn đề tài “quản lí Karaoke Khánh Dương ”.

1. Mục tiêu của đề tài:

* Trau dồi kiến thức cho môn học, có kinh nghiệm phân tích, thiết kế các bài toán thực tế.
* Tăng kỹ năng làm việc nhóm, cách nhìn tổng quan về vấn đề.
* Tin học hóa quy trình quản lý tại phòng hát.
* Ứng dụng xây dựng website quản lý phòng hát Karaoke với các tính năng sau: giao diện thân thiện, dễ sử dụng, tìm kiếm, xem thông tin sản phẩm, thống kê, tương tác dễ dàng, nhanh chóng,...
  1. GiỚi hẠn và phẠm vi cỦa đỀ tài:

Tương đối nhỏ, trước mắt sẽ triển khai tại phòng hát Karaoke Khánh Dương ở thành phố Nha Trang.

Đối tượng sử dụng: Phần mềm được sử dụng bởi quản lý, nhân viên bình thường.

* 1. Tính khẢ thi:

Xây dựng được website quản lý phòng hát Karaoke theo yêu cầu của đề tài.

Xây dựng được hệ thống quản lý phòng hát Karaoke có đủ các chức năng cơ bản.

* 1. KHẢO SÁT HIỆN TRẠNG:

1. Mô tả địa điểm khảo sát

Phòng hát Karaoke Khánh Dương bắt đầu kinh doanh dịch vụ từ tháng 11 năm 2022, chuyên cung cấp các sản phẩm và dịch vụ liên quan đến công nghiệp karaoke. Cửa hàng cung cấp cho khách hàng một bộ sưu tập bài hát đa dạng và phong phú, từ các bản hit mới nhất cho đến những bài hát cổ điển được yêu thích nhất. Không chỉ có những bài hát đa dạng, phòng hát Karaoke Khánh Dương còn được trang bị các thiết bị âm thanh và ánh sáng chuyên nghiệp để mang đến cho khách hàng một trải nghiệm âm nhạc tuyệt vời nhất. Bên cạnh đó, đội ngũ nhân viên cũng rất nhiệt tình và chuyên nghiệp, sẵn sàng hỗ trợ khách hàng trong suốt quá trình hát.

1. Hiện trạng tổ chức

|  |  |
| --- | --- |
| **Tên bộ phận** | **Chức năng và nhiệm vụ** |
| Nhà quản lý | Quản lý toàn bộ hoạt động của nhà hát Karaoke. Điều hành mọi hoạt động và quản lý nhân sự trong nhà hát |
| Bộ phận kho | Quản lý nguồn cung của nhà hát Khánh Dương. Đảm bảo cập nhật nhanh và đầy đủ nhất các mặt hàng mới. Tiếp nhận và quản lý mặt hàng nhập vào và bán ra, quản lý bảo quản mặt hàng trong kho, lập hóa đơn và báo cáo xuất nhập mặt hàng, báo cáo mặt hàng tồn kho |
| Bộ phận thu ngân | Quản lý chi phí hoạt động kinh doanh của nhà hát. Lập hóa đơn khách hàng, thanh toán hóa đơn, báo cáo thống kê hóa đơn nhập xuất. |
| Bộ phận kinh doanh | Quản lý hoạt động kinh doanh của nhà sách. Có nhiệm vụ giúp người dùng có thể tiếp cận với nhà sách một cách tốt nhất. Tiếp nhận vận chuyển sách với số lượng lớn cho khách hàng, giới thiệu tư vấn cho khách hàng, tìm hiểu thị trường và nhu cầu, báo cáo doanh thu. |

1. Hiện trạng nghiệp vụ:
   1. Các nghiệp vụ hiện tại:

* Nhà quản lý:
* Quản lý thông tin nhân viên
* Quản lý thông tin mặt hàng trong kho
* Quản lý thông tin khách hàng
* Tra cứu thông tin nhân viên
* Tính lương nhân viên
* Quản lý tài khoản hệ thống
* Quản lý thu chi
* Bộ phận kho:
* Quản lý thông tin mặt hàng trong kho
* Thêm mặt hàng vào kho
* Tra cứu mặt hàng
* Nhập thêm hàng vào kho
* Thống kê mặt hàng tồn kho
* Bộ phận thu ngân
* Tra cứu thông tin khách hàng
* Lập hóa đơn bán hàng
* Quản lý thông tin khách hàng
* Tính toán doanh thu
* Báo cáo doanh thu bán hàng
* Quản lý đơn đặt phòng của khách hàng
  1. Đánh giá nghiệp vụ
* Hiện nay cửa hàng karaoke Khánh Dương đang vận hành tương đối ổn định. Tuy nhiên, cửa hàng vẫn đang đối mặt với nhiều khó khăn thách thức:
* Đối đầu với cạnh tranh: có rất nhiều cửa hàng Karaoke khác đang hoạt động trong cùng khu vực hoạt động của cửa hàng. Do đó, cửa hàng phải đối mặt với sự cạnh tranh khốc liệt trong việc thu hút khách hàng.
* Quản lý tài chính: kinh doanh cửa hàng Karaoke đòi hỏi một khoản đầu tư ban đầu lớn để mua sắm trang thiết đồng thời cũng phải quản lý chi phí, thu nhập, đặc biệt là khi kinh doanh trong thời gian khó khăn như hiện nay.
* Quản lý nhân viên: cần phải tuyển dụng, huấn luyện và quản lý đội ngũ nhân viên của mình. Điều này đòi hỏi phải có khả năng lãnh đạo và quản lý để đảm bảo nhân viên luôn làm việc hiệu quả.
* Giải quyết các vấn đề kỹ thuật: trong khi hoạt động, cửa hàng Karaoke có thể gặp phải các sự cố kỹ thuật như máy tính bị hỏng, hệ thống âm thanh không hoạt động. Phải có kiến thức về kỹ thuật để giải quyết các vấn đề đó hoặc tìm kiếm giải pháp từ các chuyên gia.
* Đáp ứng nhu cầu của khách hàng: khách hàng có nhu cầu đa dạng về bài hát, thiết bị và dịch vụ, cần phải tìm cách đáp ứng nhu cầu đó để giữ chân khách hàng và thu hút khách hàng mới.

1. Hiện trạng tin học:
   1. Các nghiệp vụ hiện tại:

* Thiết bị hiện tại: máy tính để bàn.
* Số lượng: 3.
* Thông số kỹ thuật mỗi máy: Vi xử lý Intel Core i7-6700 3.40 GHz, 8GB RAM, Bộ nhớ ngoài 1TB SSD, màn hình 15” HD+, card đồ họa tích hợp GTX 1050.
* Vị trí: Một máy đặt ở quầy thu ngân dành cho bộ phận thu ngân, một máy đặt ở kho lưu trữ mặt hàng dành cho bộ phận kho, một máy ở phòng quản lý dành cho nhà quản lý và bộ phận kinh doanh.
  1. Về phần mềm
* Hệ điều hành: Windows 10 Home phiên bản 22H2.
* Hệ quản trị cơ sở dữ liệu: SQL Server 2017.
* Các phần mềm khác như Google Chrome, MS Word, MS Excel,…
* Phần mềm quản lý: hiện tại nhà hát chưa sử dụng phần mềm quản lý nào
  1. Về nhân sự
* Nhân sự: Một quản lý, hai nhân viên thu ngân, một nhân viên kho, một nhân viên kinh doanh.
* Trình độ: Quản lý có trình độ tin học nâng cao, các nhân viên còn lại có trình độ cơ bản.

CHƯƠNG 2. CỞ SỞ LÝ THUYẾT

* 1. TỔng quan vỀ ngôn ngỮ Asp.NET MVC

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 Forms. 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.

1. Giới thiệu Asp.NET MVC framework

A diagram of a model

Description automatically generated with medium confidenceASP.NET là một nền tảng ứng dụng web (web application framework) được phát triể và cung cấp bởi Microsoft, cho phép những người lập trình tạo ra những trang web động, những ứng dụng web và những dịch vụ web. Dựa trên ASP.NET, ASP.NET MVC cho phép các nhà phát triển phần mềm xây dựng một ứng dụng web dựa trên mẫu thiết kế MVC. MVC là một mẫu thiết kế (design pattern) chuẩn, được sử dụng nhằm chia ứng dụng thành ba thành phần chính: model, view và controller. Phiên bản ASP.NET MVC: ASP.NET MVC 5 (10/2013), ASP.NET MVC 5.2.7 (11/2017). Nền tảng MVC bao gồm các thành phần dưới đây:

Hình 1: Mẫu Model - Controller - View

**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 liên quan. Trong trường hợp này, dữ liệu được lấy như là một đối tượng model (hơn là tầng model)

**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.

Lấy ví dụ một GUI Component (thành phần đồ họa người dùng ) đơn giản là checkbox. Checkbox có thành phần Model để quản lý trạng thái của nó là check hay uncheck, thành phần View để thể hiện nó với trạng thái tương ưng lên màn hình, và thành phần Controller để xử lý nhưng sự kiện khi có sự tương tác của người sử dụng hoặc các đối tượng khác lên checkbox. Khi ngươi sử dụng nhần chuột vào Check box , thành phần Controller của Checkbox sẽ xử lý sự kiện này, yêu cầu thành phần Model thay đổi dữ liệu trạng thái. Sau khi thay đổi trạng thái, thành phần Model phát thông điệp đến thành phần View và Controller. Thành phần View của Checkbox nhận được thông điệp sẽ cập nhật lại thể hiện của Checkbox, phản ánh chính sác trạng thái Checkbox do Model lưu giữ. Thành phần Controller nhận được thông điệp do Model gởi tới sẽ có nhưng tương tác phản hòi với người sử dụng nếu cần thiết.

**Tiềm hiểu thêm về Controler**

Controller có trách nhiệm chính là điều hướng các yêu cầu của người sử dụng. Như vậy trên toàn ứng dụng của ta, tất cả các request đều sẽ phải đi tới controller. Và tại đây, ứng với các tham số người sử dụng truyền mà ta đưa họ đến một tác vụ nào đó trên ứng dụng. Tại các tác vụ này, chúng sẽ thông qua lớp model để làm việc và trả kết quả trở về controller. Cuối cùng controller sẽ đẩy dữ liệu thao tác tới view. View là thành phần cuối cùng mà người sử dụng nhận được khi họ giở request tới ứng dụng. Có thể hiểu controller, giống với kỹ thuật đa cấp với các tác vụ chạy ứng dụng phân cấp theo từng nhánh riêng biệt như: Module, action,… Qua hình vẽ này, ta hiểu rằng. Để có thể thao tác với các action (hành động) ta cần phải đi qua file index.php. Lúc này file index đóng vai trò như một controller được dùng để điều hướng các request. Ứng với các request thì nó sẽ trả về một controller khác để xử lý tác vụ một cách cụ thể. Tại controller con, nó sẽ gọi các action riêng biệt. Ở đó, thông qua các action mà nó gọi tới các file xử lý giản đơn.

**Tìm hiểu Model**

Model là thành phần chủ yếu được sử dụng để thao tác xử lý dữ liệu. Trong các framework, Model vẫn thường sử dụng theo phương thức Active Record. Một trong những design pattern. Chúng có tác dụng rút ngắn thời gian viết câu truy vấn cho người sử dụng. Biến những câu truy vấn phức tạp trở nên gần gũi và thân thiện với người sử dụng thông qua các thư viện được định nghĩa sẵn.

Model thường sẽ là các phương thức có trách nhiệm xử lý các tác vụ như: select, insert, update, delete các record trong database. Ứng với các lấy dữ liệu, model thường sử dụng mảng để gởi trả kết quả về. Vì mảng có thể cho phép model lưu trữ nhiều thông tin hơn, nên thường các record khi bóc tách chúng sẽ mang các dữ liệu của database một cách chi tiết. Khi sử dụng models, ta cũng cần tuân theo nguyên tắc chính của chúng là không xuất giá trị trực tiếp trong model. Mà tất cả những dữ liệu ấy, phải đưa vào mảng và trả về theo phương thức. Và tiếp tục ở view ta sẽ sử dụng nó để lấy dữ liệu ra.

**Tìm hiều về View**

View là phần hiển thị thông tin tương phản khi gởi và nhận request. Trước đây, khi người lập trình chưa nghĩ tới view. Họ thường thao tác xử lý dữ liệu ngay trực tiếp trên ứng dụng và đổ cả dữ liệu ngay trên file PHP đó. Điều này làm cho ứng dụng trở nên cồng kềnh, và đặc biệt rất khó cho việc bảo trì nâng cấp sau này. Nhất là đối với designer, việc thay đổi giao diện của một website luôn làm cho họ cảm thấy đau đầu vì phải vọc thẳng vào core. Trước đây, để giải quyết tình huống này. Người ta thường sử dụng template để phân tách website thành 2 mảng riêng biệt. Một là giao diện và một là core. Việc chỉnh sửa giao diện trở nên đơn giản hơn đối với họ so với cách viết thập cẩm kia. Tuy nhiên, các thư viện này thực chất sẽ làm cho ứng dụng của chúng ta trở nên chậm chạp hơn bao giờ hết. Bởi chúng phải phiên dịch nhiều lần các kịch bản. Chẳng hạn: Để dễ thao tác, smarty sẽ dịch ngược các yêu cầu của bạn sang ngôn ngữ của nó. Sau đó chúng sẽ chuyển ngôn ngữ đó sang PHP và thao tác xử lý trên nó. Việc này sẽ làm ứng dụng chậm chạp, do cứ phải dịch qua, dịch lại một kịch bản. Trong khi, với sự kết hợp của PHP thuần, ứng dụng của bạn sẽ nhanh và ổn định hơn nhiều. Và view cũng là một phần trong việc nâng cấp những hạn chế ấy. Chúng giúp giảm thiểu tối đa quá trình biên dịch nhiều lần. Và làm cho ứng dụng trở nên mạnh mẽ và chuyên nghiệp hơn nhiều so với cách lập trình thuần.

1. Lý do chọn Asp.NET MVC
   1. Các tính năng của 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 (.mater) Bạn có thể dụng các tình năng có sẵn của ASP.NET như dùng 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 MVC5 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.
  1. TỔNG QUAN vỀ SQL SERVER

1. Khái niệm hệ quản trị CSDL SQL Server

SQL Server là một hệ thống quản lý cơ sở dữ liệu RDBMS sử dụng Transact-SQL để trao đổi dữ liệu giữa Client computer và SQL Server computer. Một RDBMS bao gồm Databases, Database engine và các ứng dụng dùng để quản lý dữ liệu và các bộ phận khác nhau trong RDBMS. 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 hàng ngàn User. SQL Server 2000 có thể kết hợp "ăn ý" với các Server khác như Microsoft Internet Information Server (IIS), ECommerce Server, Proxy Server...

1. Đặc điểm của SQL Server và đối tượng làm việc
   1. Đặc điểm

* SQL là ngôn ngữ tựa tiếng Anh.
* SQL là ngôn ngữ phi thủ tục, nó không yêu cầu ta cách thức truy nhập CSDL như thế nào. Tất cả các thông báo của SQL đều rất dễ sử dụng và ít khả năng mắc lỗi .
* SQL cung cấp tập lệnh phong phú cho các công việc hỏi đáp dữ liệu
* Chèn, cập nhật, xoá các hàng trong một quan hệ.
* Tạo, sửa đổi, thêm và xoá các đối tượng trong của CSDL để đảm bảo tính bảo mật của cơ sở dữ liệu.
* Đảm bảo tính nhất quán và sự ràng buộc của CSDL.
* Yêu cầu duy nhất để sử dụng cho các hỏi đáp là phải nắm vững được các cấu trúc CSDL của mình.

CHƯƠNG 3: NỘI DUNG THỰC HIỆN

3.1 DANH SÁCH CHỨC NĂNG YÊU CẦU CHỨC NĂNG NGHIỆP VỤ

1. Nhà quản lý

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên yêu cầu chức năng** | **Loại yêu cầu** | **Quy định/Biểu mẫu liên quan** | **Mô tả** |
| 1 | Quản lý thông tin nhân viên | Lưu trữ |  | Thêm, sửa, xóa thông tin của nhân viên trong nhà hát. |
| 2 | Tra cứu thông tin nhân viên | Tra cứu |  | Tìm kiếm thông tin nhân viên trong nhà hát theo mã, tên,... |
| 3 | Tính lương nhân viên | Tính toán |  | Tính toán lương của nhân viên theo ca và số ngày làm việc. |
| 4 | Quản lý thông tin dịch vụ trong kho | Lưu trữ |  | Xem, sửa, xóa thông tin của các dịch vụ trong nhà hát. |
| 5 | Quản lý thông tin khách hàng | Lưu trữ |  | Xem, sửa, xóa thông tin của khách hàng. |
| 6 | Quản lý tài khoản hệ thống | Lưu trữ |  | Thêm, sửa, xóa thông tin tài khoản đăng nhập vào hệ thống. |
| 7 | Quản lý chi phí | Lưu trữ |  | Xem, quản lý các nguồn chi phí của nhà hát. |

1. Bộ phận kho

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên yêu cầu chức năng** | **Loại yêu cầu** | **Quy định/Biểu mẫu liên quan** | **Mô tả** |
| 1 | Quản lý thông tin dịch vụ trong kho | Lưu trữ |  | Xem, sửa, xóa thông tin của các dịch vụ trong nhà hát. |
| 2 | Thêm dịch vụ | Lưu trữ | BM1, QĐ1 | Thêm dịch vụ mới chưa có vào kho. |
| 3 | Nhập thêm dịch vụ vào kho | Lưu trữ |  | Lập phiếu nhập các dịch vụ đã có trong kho từ nhà cung cấp. |
| 4 | Thống kê dịch vụ tồn kho | Kết xuất | BM4 | Thống kê số lượng nhập, xuất, tồn các dịch vụ trong kho theo loại dịch vụ. |
| 5 | Tra cứu dịch vụ | Tra cứu | BM2 | Tra cứu thông tin dịch vụ trong kho theo mã, tên, loại, nhà cung cấp, giá, số lượng tồn. |

1. Bộ phận thu ngân:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên yêu cầu chức năng** | **Loại yêu cầu** | **Quy định/Biểu mẫu liên quan** | **Mô tả** |
| 1 | Quản lý thông tin khách hàng | Lưu trữ |  | Xem, sửa, xóa thông tin của khách hàng. |
| 2 | Lập hóa đơn bán hàng | Tính toán | BM3, QĐ2 | Tính toán tiền và xuất hóa đơn dịch vụ. |
| 3 | Tính toán doanh thu bán hàng | Tính toán |  | Tính toán doanh thu hàng tháng của nhà hát. |
| 4 | Tra cứu khách hàng | Tra cứu |  | Tra cứu thông tin khách hàng theo mã, tên, số điện thoại. |
| 5 | Báo cáo doanh thu bán hàng | Kết xuất |  | Báo cáo doanh thu hàng tháng của nhà hát. |
| 6 | Quản lý đơn đặt phòng của khách hàng | Lưu trữ |  | Thêm, xóa, sửa đơn đặt phòng của khách hàng. |

3.2 Danh sách yêu CẦU chức năng HỆ THỐNG

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên yêu cầu chức năng** | **Mô tả chi tiết** | **Ghi chú** |
| 1 | Phân quyền sử dụng | * Nhà quản lí: tất cả các chức năng. * Bộ phận kho: tất cả các chức năng ngoại trừ sao lưu, phục hồi và cấu hình hệ thống. * Bộ phận thu ngân: tất cả các chức năng ngoại trừ sao lưu, phục hồi và cấu hình hệ thống. * Bộ phận kinh doanh: tất cả các chức năng ngoại trừ sao lưu, phục hồi và cấu hình hệ thống. |  |
| 2 | Sao lưu hệ thống | * Nhà quản lý: toàn quyền sao lưu hệ thống * Bộ phận kho: không có quyền * Bộ phận thu ngân: không có quyền * Bộ phận kinh doanh: không có quyền |  |
| 3 | Phục hồi hệ thống | * Nhà quản lý: toàn quyền phục hồi hệ thống * Bộ phận kho: không có quyền * Bộ phận thu ngân: không có quyền * Bộ phận kinh doanh: không có quyền |  |
| 4 | Cấu hình hệ thống | * Nhà quản lý: toàn quyền cấu hình hệ thống * Bộ phận kho: không có quyền * Bộ phận thu ngân: không có quyền * Bộ phận kinh doanh: không có quyền |  |
| 5 | Tự động thông báo | * Nhà quản lý: tự động thông báo khi có nhân viên mới được thêm, xóa, sửa. * Bộ phận kho: tự động thông báo khi có thông tin nhập hàng từ nhà cung cấp. * Bộ phận thu ngân: tự động thông báo khi có đơn đặt hàng từ khách hàng. * Bộ phận kinh doanh: tự động thông báo khi có yêu cầu chăm sóc khách hàng. |  |

1. Danh sách yêu cầu phi chức năng

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên yêu cầu** | **Loại yêu cầu** | **Mô tả** |
| 1 | Cho phép thay đổi số lượng nhập và đơn giá nhập tối thiểu, tối đa | Tiến hóa | Người dùng có thể điều chỉnh số lượng nhập và đơn giá nhập tối thiểu, tối đa khi thêm mới dịch vụ. |
| 2 | Tốc độ thực hiện tính toán và lập hóa đơn bán hàng nhanh | Hiệu quả | Tối đa 30 giây khi thực hiện tính toán tổng tiền trong hóa đơn. |
| 3 | Hình thức biểu diễn danh sách và tra cứu trực quan, dễ sử dụng | Tiện dụng | Hỗ trợ tra cứu gần đúng, tra cứu theo nội dung và cho phép nhập lại mà không chuyển trang. |
| 4 | Đảm bảo bảo mật dữ liệu | Bảo mật | Đảm bảo tránh việc truy cập trái phép vào dữ liệu, sử dụng các kỹ thuật mã hóa dữ liệu trong quá trình truyền và lưu trữ. |
| 5 | Tương thích với hệ điều hành và các phần mềm tiện ích | Tương thích | Không bị xung đột với hệ điều hành và có thể dễ dàng kết nối với các phần mềm Word, Excel,… |

1. Yêu cầu đặc tả phần mềm
   1. Chức năng thêm dịch vụ (Lưu Trữ)

|  |  |
| --- | --- |
| A diagram of a flowchart  Description automatically generated with low confidence | * D1: Thông tin về dịch vụ cần nhập bao gồm: tên dịch vụ, số lượng nhập, đơn giá nhập, loại dịch vụ, đơn vị tính, nhà cung cấp. * D2: Danh mục loại dịch vụ, đơn vị tính, nhà cung cấp; kết quả thông báo thành công/thất bại. * D3: Danh mục loại dịch vụ, đơn vị tính, nhà cung cấp; quy định về thông tin dịch vụ nhập (số lượng nhập tối thiểu, số lượng nhập tối đa; đơn giá nhập tối thiểu, đơn giá nhập tối đa) * D4: D1 + Mã dịch vụ * D5: D4. |
| Các bước xử lý:   * B1: Kết nối cơ sở dữ liệu. * B2: Đọc D3. * B3: Hiển thị D2 (danh mục loại dịch vụ, đơn vị tính, nhà cung cấp). * B4: Nhận D1 từ người dùng. * B5: Kiểm tra D1 (đơn giá nhập, số lượng nhập) có thỏa các quy định về thông tin dịch vụ nhập hay không (số lượng nhập tối thiểu, tối đa; đơn giá nhập tối thiểu, tối đa), nếu không chuyển xuống B8. * B6: Lưu D4. * B7: Xuất D5. * B8: Hiển thị D2 (kết quả thành công/thất bại). * B9: Ngắt kết nối cơ sở dữ liệu và kết thúc. | |

* 1. Chức năng tra cứu dịch vụ (Tra cứu)

|  |  |
| --- | --- |
| A diagram of a flowchart  Description automatically generated with low confidence | * D1: Điều kiện tra cứu gồm: Mã dịch vụ, tên dịch vụ, giá tiền tối thiểu, giá tiền tối đa. * D2: Danh mục loại dịch vụ, nhà cung cấp; kết quả tra cứu bao gồm: mã dịch vụ, tên dịch vụ, loại dịch vụ, đơn vị tính, nhà cung cấp, đơn giá, số lượng tồn. * D3: Danh mục loại dịch vụ, nhà cung cấp; thông tin dịch vụ được tìm thấy. * D4: Không có. * D5: D2 (Thông tin dịch vụ được tìm thấy). |
| Các bước xử lý:   * B1: Kết nối cơ sở dữ liệu. * B2: Đọc D3. * B3: Hiển thị D2 (danh mục loại dịch vụ, nhà cung cấp). * B4: Nhận D1. * B5: Tìm kiếm theo D3. * B6: Hiển thị D2 (kết quả tra cứu). * B7: Xuất D5. * B8: Đóng kết nối cơ sở dữ liệu. * B9: Kết thúc. | |

* 1. Chức năng lập hóa đơn bán hàng(Tính toán)

|  |  |
| --- | --- |
| A diagram of a flowchart  Description automatically generated with low confidence | * D1: Thông tin hóa đơn bao gồm: ngày lập, giờ vào, giờ ra; thông tin các dịch vụ bao gồm: mã dịch vụ, số lượng. * D2: Danh mục khách hàng, phòng bàn; thông tin các dịch vụ bao gồm: tên dịch vụ, đơn vị, đơn giá; thành tiền, tổng tiền; kết quả thành công/thất bại. * D3: Danh mục khách hàng, phòng bàn; thông tin các dịch vụ bao gồm: tên dịch vụ, đơn vị, đơn giá; quy định về số lượng dịch vụ (số lượng tối thiểu, tối đa) * D4: D1 + Mã hóa đơn. * D5: D4. |
| Các bước xử lý:   * B1: Kết nối cơ sở dữ liệu. * B2: Đọc D3. * B3: Hiển thị D2 (danh mục khách hàng, phòng bàn). * B4: Nhận D1. * B5: Hiển thị D2 (thông tin các dịch vụ) * B6: Tính toán thành tiền và tổng tiền qua D1 và D3 * B7: Hiển thị D2 (thành tiền, tổng tiền; kết quả thành công/thất bại) * B8: Đóng kết nối cơ sở dữ liệu. * B9: Kết thúc. | |

* 1. Chức năng Thống kê

|  |  |
| --- | --- |
| A diagram of a flowchart  Description automatically generated with low confidence | * D1: Thông tin về thống kê, bao gồm bắt đầu thống kê , kết thúc thống kê, loại dịch vụ * D2: Thông tin các dịch vụ thống kê được bao gồm: Mã dịch vụ, tên dịch vụ, tồn đầu kì, số lượng nhập, số lượng bán, tồn cuối kì. * D3: Danh mục loại dịch vụ. * D4: không có * D5: D2. |
| Các bước xử lý:   * B1: Kết nối cơ sở dữ liệu. * B2: Đọc D3 danh mục loại dịch vụ. * B3: Nhận D1. * B4: Thống kê các dịch vụ trong khoảng thời gian đã chọn * B5: Hiển thị D2. * B6: Xuất D5 (nếu cần). * B7: Ngắt kết nối cơ sở dữ liệu. * B8: Kết thúc. | |

3.3

3.2 ThiẾt kẾ hỆ thỐng

1. A picture containing text, diagram, screenshot, font

   Description automatically generatedA screenshot of a computer

   Description automatically generated with low confidenceThiết kế dữ liệu

A picture containing text, screenshot, diagram, font

Description automatically generated

Bảng KhachHang

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Miền giá trị | Ý nghĩa |
| 1 | MaKhachHang | Chuỗi | Tối đa 9 kí tự | Mã khách hàng |
| 2 | TenKhachHang | Chuỗi | Tối đa 50 kí tự | Tên khách hàng |
| 3 | SoDienThoai | Số nguyên | Tối đa 20 số | Số điện thoại khách hàng |
| 4 | DiaChi | Chuỗi | Tối đa 100 kí tự | Địa chỉ khách hàng |
| 5 | MaTaiKhoan | Chuỗi | Tối đa 9 | Mã Tài khoản khách hàng |

Bảng TaiKhoan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Miền giá trị | Ý nghĩa |
| 1 | MaTaiKhoan | Chuỗi | Tối đa 9 kí tự | Mã Tài khoản |
| 2 | TenNguoiDung | Chuỗi | Tối đa 50 kí tự | Tên Người dùng |
| 3 | TenDangNhap | Số nguyên | Tối đa 50 số | Tên đăng nhập |
| 4 | MatKhau | Chuỗi | Tối đa 50 kí tự | Mật khẩu người dùng |
| 5 | PhanQuyen | Chuỗi | Tối đa 9 kí tự | Phân quyền người dùng |

Bảng NhanVien

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Miền giá trị | Ý nghĩa |
| 1 | MaNhanVien | Chuỗi | Tối đa 7 kí tự | Mã nhân viên |
| 2 | TenNhanVien | Chuỗi | Tối đa 100 kí tự | Tên nhân viên |
| 3 | GioiTinh | Bool | 2 giá trị 0 và 1 | Giới tính nhân viên |
| 3 | NgaySinh | Ngày | < Ngày hiện tại | Ngày sinh nhân viên |
| 4 | Luong | Int |  | Lương của nhân viên |
| 5 | MaTaiKhoan | Chuỗi | Tối đa 9 kí tự | Tài khoản của nhân viên |

Bảng PhanQuyen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Miền giá trị | Ý nghĩa |
| 1 | MaPhanQuyen | Chuỗi | Tối đa 9 kí tự | Mã nhân viên |
| 2 | TenPhanQuyen | Chuỗi | Tối đa 50 kí tự | Tên nhân viên |

Bảng DatPhong

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Miền giá trị | Ý nghĩa |
| 1 | MaDatPhong | Chuỗi | Tối đa 9 kí tự | Mã nhân viên |
| 2 | GioVao | Time | >= Giờ hiện tại | Giờ khách hàng vào |
| 3 | GioRa | Time | >= GioVao | Giờ khách hàng ra |
| 4 | NgayDat | Ngày | Ngày hiện tại | Ngày sinh nhân viên |
| 5 | MaKhachHang | Chuỗi | Tối đa 9 kí tự | Lương của nhân viên |
| 6 | MaPhongHat | Chuỗi | Tối đa 9 kí tự | Tài khoản của nhân viên |
| 7 | TrangThai | Chuỗi | Tối đa 9 kí tự | Trang thái của phòng |

Bảng TrangThai

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Miền giá trị | Ý nghĩa |
| 1 | MaTrangThai | Chuỗi | Tối đa 9 kí tự | Mã nhân viên |
| 2 | TenTrangThai | Chuỗi | Tối đa 50 kí tự | Trạng thái phòng |

Bảng HoaDon

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Miền giá trị | Ý nghĩa |
| 1 | MatHoaDon | Chuỗi | Tối đa 9 kí tự | Mã hóa đơn |
| 2 | NgayLap | Ngày | Ngày hiện tại | Ngày lập hóa đơn |
| 3 | MaDatPhong | Chuỗi | Tối đa 9 kí tự | Mã phòng bàn |
| 4 | MaNhanVien | Chuỗi | Tối đa 9 kí tự | Mã nhân viên lập |
| 5 | TongTien | Int |  | Tổng tiền thanh toán |

Bảng PhieuNhap

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Miền giá trị | Ý nghĩa |
| 1 | MaPhieuNhap | Chuỗi | Tối đa 9 kí tự | Mã hóa đơn |
| 2 | NgayLapPN | Ngày | Ngày hiện tại | Ngày lập phiếu nhập |
| 3 | MaNhanVien | Chuỗi | Tối đa 9 kí tự | Mã nhân viên lập |
| 4 | TongTien | Int |  | Tổng tiền nhập |

Bảng ChiTietHoaDon

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Miền giá trị | Ý nghĩa |
| 1 | MaHoaDon | Chuỗi | Tối đa 9 kí tự | Mã hóa đơn |
| 2 | MaDichVu | Chuỗi | Ngày hiện tại | Mã của dịch vụ |
| 3 | DonGiaBan | Int | Tối đa 9 kí tự | Đơn giá dịch vụ trong hóa đơn |
| 4 | SoLuongBan | Int |  | Số lượng bán |

Bảng ChiTietPhieuNhap

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Miền giá trị | Ý nghĩa |
| 1 | MaPhieuNhap | Chuỗi | Tối đa 9 kí tự | Mã Phiếu Nhập |
| 2 | MaDichVu | Chuỗi | Ngày hiện tại | Mã của dịch vụ |
| 3 | DonGiaMua | Int | Tối đa 9 kí tự | Đơn giá nhập dịch vụ trong phiếu nhập |
| 4 | SoLuongMua | Int |  | Số lượng nhập dịch vụ trong phiếu nhập |

Bảng PhongHat

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Miền giá trị | Ý nghĩa |
| 1 | MaPhongHat | Chuỗi | Tối đa 9 kí tự | Mã phòng hát |
| 2 | TenPhongHat | Chuỗi | Tối đa 50 kí tự | Tên phòng hát |
| 3 | LoaiPhong | Chuỗi | Tối đa 9 kí tự | Loại phòng hát |
| 4 | TrangThai | Chuỗi | Tối đa 9 kí tự | Trạng thái của phòng hát |

Bảng LoaiPhong

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Miền giá trị | Ý nghĩa |
| 1 | MaLoaiPhong | Chuỗi | Tối đa 9 kí tự | Mã phòng hát |
| 2 | TenLoaiPhong | Chuỗi | Tối đa 50 kí tự | Tên phòng hát |
| 3 | GiaPhong | Int |  | Giá của phòng hát |

Bảng DichVu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Miền giá trị | Ý nghĩa |
| 1 | MaDichVu | Chuỗi | Tối đa 9 kí tự | Mã dịch vụ |
| 2 | TenDichVu | Chuỗi | Tối đa 50 kí tự | Tên dịch vụ |
| 3 | LoaiDichVu | Chuỗi | Tối đa 9 kí tự | Loại của dịch vụ |
| 4 | DonViTinh | Chuỗi | Tối đa 9 kí tự | Đơn vị tính dịch vụ |
| 5 | DonGia | Int |  | Tổng tiền thanh toán |
| 6 | SoLuongTon | Int |  | Số lượng tồn |
| 7 | NhaCungCap | Chuỗi | Tối đa 9 kí tự | Nhà cung cấp của dịch vụ |

Bảng NhaCungCap

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Miền giá trị | Ý nghĩa |
| 1 | MaNhaCungCap | Chuỗi | Tối đa 9 kí tự | Mã của nhà cung cấp |
| 2 | TenNhaCungCap | Chuỗi | Tối đa 50 kí tự | Tên của nhà cung cấp |
| 3 | SoDienThoai | Chuỗi | Tối đa 20 kí tự | Số điện thoại của nhà cung cấp |
| 4 | DiaChi | Chuỗi | Tối đa 100 kí tự | Địa chỉ nhà cung cấp |

Bảng DonViTinh

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Miền giá trị | Ý nghĩa |
| 1 | MaDonVi | Chuỗi | Tối đa 9 kí tự | Mã đơn vị |
| 2 | TenDonVi | Chuỗi | Tối đa 50 kí tự | Tên đơn vị |

Bảng LoaiDichVu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Thuộc tính | Kiểu | Miền giá trị | Ý nghĩa |
| 1 | MaLoaiDV | Chuỗi | Tối đa 9 kí tự | Mã loại dịch vụ |
| 2 | TenLoaiDV | Chuỗi | Tối đa 50 kí tự | Tên loại đơn vị |