TRƯỜNG ĐẠI HỌC NHA TRANG

**KHOA CÔNG NGHỆ THÔNG TIN**



# Môn học: Phát triển ứng dụng web

**ĐỀ TÀI**

**QUẢN LÝ ĐẶT ĐỒ ĂN ONLINE**

GVBM: Bùi Chí Thành

Thành viên nhóm:

|  |  |  |
| --- | --- | --- |
| 1. | Nguyễn Trọng Tín | 62132235 |
| 2. | Đoàn Duy Thiện | 62132003 |
| 3. | Nguyễn Tiến Dũng | 62130312 |
| 4. | Từ Công Thắng | 62131885 |

***MỤC LỤC***

[DANH SÁCH BẢNG BIỂU 3](#_Toc74414416)

[DANH SÁCH HÌNH VẼ 4](#_Toc74414417)

[DANH SÁCH TỪ VIẾT TẮT 5](#_Toc74414418)

[CHƯƠNG 1. TỔNG QUAN VỀ ĐỀ TÀI 6](#_Toc74414419)

[1.1. Lý do chọn đề tài: 6](#_Toc74414420)

[1.2. Mục tiêu của đề tài 6](#_Toc74414421)

[1.3. Giới hạn và phạm vi của đề tài 6](#_Toc74414422)

[1.4. Tính khả thi 6](#_Toc74414423)

[CHƯƠNG 2: CƠ SỞ LÝ THUYẾT 7](#_Toc74414424)

[2.1 Tổng quan về ngôn ngữ Asp.NET MVC 7](#_Toc74414425)

[2.1.1 Giới thiệu Asp.NET MVC Framework 7](#_Toc74414426)

[2.1.2 Lý do chọn Asp.NET MVC 12](#_Toc74414427)

[2.2 Tổng quan về SQL SERVER 13](#_Toc74414428)

[2.2.1 Hệ quản trị CSDL SQL Server 13](#_Toc74414429)

[CHƯƠNG 3: NỘI DUNG THỰC HIỆN 15](#_Toc74414430)

[3.1 Đối tượng sử dụng website: 15](#_Toc74414431)

[3.2. Các chức năng chính của website: 15](#_Toc74414432)

[3.2.1 Chức năng đối với khách hàng 15](#_Toc74414433)

[3.2.2 Chức năng đối với Admin 16](#_Toc74414434)

[3.3 Phân tích thiết kế hệ thống 17](#_Toc74414435)

[3.3.1 Các yêu cầu chức năng chính của hệ thống 17](#_Toc74414436)

[3.3.2 Các biểu đồ Use Case 19](#_Toc74414437)

[3.4 Thiết kế cơ sở dữ liệu 19](#_Toc74414438)

[3.4.1 Mô hình cơ sở dữ liệu 19](#_Toc74414439)

[3.6.2 Danh sách các lớp đối tượng 20](#_Toc74414440)

[3.5 Thiết kế và đặc tả giao diện 23](#_Toc74414441)

[3.5.1 Giao diện trang chủ 23](#_Toc74414442)

[3.5.2 Giao diện trang đăng nhập 24](#_Toc74414443)

[3.5.3 Giao diện trang đăng ký 25](#_Toc74414444)

[3.5.4 Giao diện trang danh mục sản phẩm 26](#_Toc74414445)

[3.5.5 Giao diện trang chi tiết sản phẩm 26](#_Toc74414446)

[3.6.6 Giao diện trang giỏ hàng 28](#_Toc74414447)

[3.5.7 Giao diện trang đặt hàng và thanh toán 29](#_Toc74414448)

[3.5.8 Giao diện trang quản trị 30](#_Toc74414449)

[3.5.9 Giao diện trang quản lý sản phẩm 31](#_Toc74414450)

[3.5.10 Giao diện trang thêm, sửa, xóa sản phẩm 32](#_Toc74414451)

[3.5.11 Giao diện trang quản lý nhà sản xuất 33](#_Toc74414452)

[3.7.12 Giao diện trang quản lý loại sản phẩm 35](#_Toc74414453)

[CHƯƠNG 4: KẾT LUẬN 36](#_Toc74414454)

[4.1 Kết quả đạt được của đề tài 36](#_Toc74414455)

[4.2 Hạn chế của đề tài 36](#_Toc74414456)

[4.3 Kết luận 36](#_Toc74414457)

DANH SÁCH BẢNG BIỂU

[Bảng 3-1: Danh sách các thuộc tính lớp Nhà sản xuất 20](#_Toc452307838)

[Bảng 3-2: Danh sách các thuộc tính lớp Loại Sản phẩm 20](#_Toc452307839)

[Bảng 3-3: Danh sách các thuộc tính lớp Sản phẩm 21](#_Toc452307839)

[Bảng 3-4: Danh sách các thuộc tính lớp Giỏ Hàng 22](#_Toc452307839)

[Bảng 3-5: Danh sách các thuộc tính lớp Hóa Đơn 22](#_Toc452307839)

[Bảng 3-6: Danh sách các thuộc tính lớp Chi Tiết Hóa Đơn 23](#_Toc452307839)

DANH SÁCH HÌNH VẼ

[Hình 1: Mẫu Model - View - Controller 8](#_Toc452307809)

[Hình 2: Mô hình tuần tự của MVC 10](#_Toc452307809)

[Hình 3: Biểu đồ Use Case tổng quát 19](#_Toc452307809)

[Hình 4: Biểu đồ dữ liệu quan liệu 19](#_Toc452307809)

[Hình 5: Giao diện trang chủ 23](#_Toc452307809)

[Hình 6: Giao diện trang đăng nhập 24](#_Toc452307809)

[Hình 7: Giao diện trang đăng ký 25](#_Toc452307829)

[Hình 8: Giao diện trang danh mục sản phẩm 26](#_Toc452307830)

[Hình 9: Giao diện trang chi tiết sản phẩm 27](#_Toc452307831)

[Hình 10: Giao diện trang giỏ hàng 28](#_Toc452307832)

[Hình 11: Giao diện trang đặt hàng 29](#_Toc452307832)

[Hình 12: Giao diện trang thanh toán 29](#_Toc452307832)

[Hình 13: Giao diện trang quản trị 30](#_Toc452307832)

[Hình 14: Giao diện trang quản lý sản phẩm 31](#_Toc452307832)

[Hình 15: Giao diện trang thêm sản phẩm 32](#_Toc452307832)

[Hình 16: Giao diện trang sửa sản phẩm 32](#_Toc452307832)

[Hình 17: Giao diện trang xóa lý sản phẩm 33](#_Toc452307832)

[Hình 18: Giao diện trang quản lý nhà sản xuất 34](#_Toc452307832)

[Hình 19: Giao diện trang quản lý loại sản phẩm 35](#_Toc452307832)

DANH SÁCH TỪ VIẾT TẮT

|  |  |  |
| --- | --- | --- |
| Từ viết tắt | Từ đầy đủ | Giải thích |
| CLR | Common Language Runtime |  |
| HTML | Hyper Text Markup Language |  |
| HTTP | [Hyper Text Transfer Protocol](http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) |  |
| OOP | Object Oriented Programming |  |
| MSIL | Microsoft Intermediate Language |  |
| IL | Intermediate Language |  |
| JIT | Just-In-Time |  |
| RDMS | Relational Database Management System |  |
| SQL | Structured Query Language | ngôn ngữ truy vấn mang tính cấu trúc |

# TỔNG QUAN VỀ ĐỀ TÀI

## 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 “***“Xây dựng website quản lý cửa hàng bán điện thoại MobiFone”.***

## 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 cửa hàng.
* Ứng dụng xây dựng website quản lý cửa hàng điện thoai với các tính năng sau: giao diện thân thiện, dễ sử dụng, tìm kiếm , xem thông tin sản phẩm dễ dàng, nhanh chóng.

## Giới hạn và phạm vi của đề tài

* Nhỏ, trước mắt sẽ triển khai tại các cửa hàng nhỏ ở thành phố Nha Trang.
* Đối tượng sử dụng: Phần mềm được sử dụng bởi nhân viên bán hàng, nhân viên nhập hàng, người quản lý.

## Tính khả thi

* Xây dựng được website bán điện thoại theo yêu cầu của đề tài.
* Xây dựng được hệ thống quản lý cửa hàng điện thoại có đủ các chức năng cơ bản.

# CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

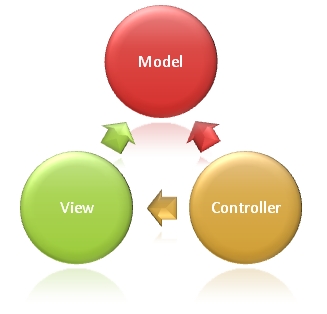
2.1 Tổng quan về ngôn ngữ Asp.NET MVC

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 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.

2.1.1 Giới thiệu Asp.NET MVC Framework

ASP.NET là một nền tảng ứng dụng web (web application framework) được phát triển 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 - 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 liên quan. Trong trường hợp này, dữ liệu được lấy như là một đối tượng model (hơn là tầng model)

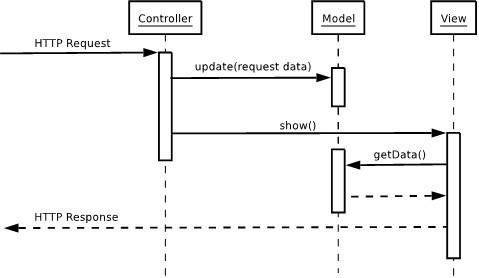
**Views:**  là các thành phần dùng để hiển thị giao diện người dùng (UI). Thông thường, view được tạo dựa vào thông tin dữ liệu model. Ví dụ như, view dùng để cập nhật bảng Products sẽ hiển thị các hộp văn bản, drop-down list, và các check box dựa trên trạng thái hiện tại của một đối tượng Product.

**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.

****

Hình 2 :Mô hình tuần tự của MVC

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.

2.1.2 Lý do chọn Asp.NET MVC

* 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.

2.2 Tổng quan về SQL SERVER

2.2.1 Hệ quản trị CSDL SQL Server

a ) Khái niệm hệ quản trị cơ sở dữ liệu 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...

***b) Đặc điểm của SQL Server và đối tượng làm việc***

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.
* Điều khiển việc truy nhập tới cơ sở dữ liệu và các đối tượng 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

## . Đối tượng sử dụng website:

* Đối với khách hàng: Khách hàng có thể truy cập vào website để xem, tra cứu và cập nhật những sản phẩm mới của website. Ngoài ra, khách hàng còn có thể mua đặt hàng online, đóng góp ý kiến về sản phẩm cho website, có thế đăng ký làm thành viên (nếu muốn).
* Đối với Admin: Admin có quyền quản lý (thêm/sửa/xóa) các loại sản phẩm, sản phẩm, hóa đơn.

## 3.2. Các chức năng chính của website:

3.2.1 Chức năng đối với khách hàng

1. ***Xem sản phẩm***

Khách hàng lựa chọn nước uống muốn xem bằng click vào hình ảnh đồ uống đã được hiển thị lên trang web sau đó sẽ được hệ thống xử lý và tìm kiếm ở cơ sở dữ liệu, sau khi xử lý xong yêu cầu thì khách hàng có thể xem chi tiết các thông tin như tên, giá cả, các nghiên liệu của đồ uống.

1. ***Tìm kiếm sản phẩm***

Khách hàng có thể tìm kiếm sản phẩm theo các tính năng (tên, loại, giá) vào ô tìm kiếm được xây trên website. Sau đó hệ thống sẽ ghi nhận và thực hiện tìm kiếm trong cơ sở dữ liệu tất cả sản phẩm mà tên có chứa từ khóa hay gần đúng với từ khóa, sau khi xử lý xong yêu cầu các thông tin sản phẩm được hiển thị.

1. ***Lựa chọn sản phẩm vào giỏ hàng.***

Khách hàng có thể lựa chọn bất kì một loại sản phẩm (mã đồ uống) nào mình cảm thấy thích để thêm vào giỏ hàng. Hệ thống xử lý dữ liệu, nếu sản phẩm cần thêm đã có trong giỏ hàng thì khi đó nó sẽ tự động tăng thêm một đơn vị, ngược lại chưa có thì tự động thêm mới. Sau khi lưu trữ xong, thông tin sản phẩm sẽ được hiển thị trong giỏ hàng.

1. ***Mua hàng***

Sau khi chọn sản phẩm vào giỏ hàng, khách hàng có thể mua hàng và cung cấp thông tin họ tên, địa chỉ, email, điện thoại, để đặt hàng. Hệ thống sẽ xử lý thông tin sản phẩm, khách hàng để ghi nhận hóa đơn vào cơ sở dữ liệu. Hiển thị thông tin về sản phẩm mà khách hàng đã đặt hàng.

1. ***Đăng ký thành viên***

Khách vãng lai có thể đăng ký làm thành viên thông qua một form được xây dựng sẵn. Khi đó, khách hàng sẽ cung cấp các thông tin như địa chỉ email, password, confirm password. Nếu khách hàng nhập sai email hoặc password, confirm password không đúng thì sẽ báo lỗi và yêu cầu nhập lại. Và sau đó, thông báo kết quả.

1. ***Đăng nhập website***

Khách hàng sau khi có tài khoản thì sẽ đăng nhập vào website bằng cách gõ địa chỉ email, password đã đăng ký. Sau đó hệ thống sẽ kiểm tra cơ sở dữ liệu để xác nhận Hiển thị thông báo đăng nhập thành công hoặc báo lỗi và yêu cầu đăng nhập lại.

3.2.2 Chức năng đối với Admin

1. ***Quản lý tài khoản***

Quản trị viên đăng nhập vào hệ thống bằng cách cung cấp địa chỉ email, password vào form đăng nhập được xây dựng sẵn. Kiểm tra thông tin vào có hợp lệ hay không. Nếu có chuyển đến trang quản trị ngược lại thông báo lỗi và yêu cầu nhập lại.

1. ***Quản lý sản phẩm***

Quản trị viên có quyền thêm, sửa hoặc xóa sản phẩm:

* Thêm:
* Kiểm tra tính hợp lệ của thông tin sản phẩm mới
* Kiểm tra có tồn tại hay chưa?
* Nếu chưa thêm mới sản phẩm vào CSDL.
* Sửa:
* Kiểm tra tính hợp lệ của thông tin sản phẩm mới
* Kiểm tra có tồn tại hay chưa?
* Nếu chưa cập nhật thông tin sản phẩm vào CSDL.
* Xóa:
* Nếu sản phẩm đã thuộc một giỏ hàng hoặc đơn hàng thì không được phép xóa.
* Ngược lại cho phép xóa sản phẩm.

1. ***Quản lý nhà sản xuất, loại sản phẩm***

Quản trị viên có quyền thêm, sửa thông qua việc cập nhật tình trạng (0 - hiển thị, 1- không hiển thị) hoặc xóa các loại sản phẩm, nhà sản xuất.

1. ***Quản lý hóa đơn***

Quản trị viên có thể xem, xóa và cập nhật trạng thái.

* Xem:
* Hiển thị tất cả các đơn hàng, phân theo 3 loại: chờ xác nhận, đang giao hàng, đã giao hàng.
* Xóa:
* Chỉ được phép xóa các đơn hàng đã giao hàng.
* Các đơn hàng chờ xác nhận, đang giao hàng không xóa.
* Cập nhật trạng thái
* Cập nhật trạng thái từ thông báo nhân viên.

3.3 Phân tích thiết kế hệ thống

3.3.1 Các yêu cầu chức năng chính của hệ thống

***a)Yêu cầu lưu trữ***

* Hệ thống cần lưu trữ các thông tin liên quan đến sản phẩm, loại sản phẩm, nhà sản xuất, thông tin đăng nhập, các đơn đặt hàng.
* Thông tin đăng nhập
  + Phần quản trị của website cần đảm bảo tính bảo mật do có thể truy cập tới toàn bộ thông tin về sản phẩm và hoạt động kinh doanh.
* Thôngtin SanPham (MaSanPham, MaLoaiSanPham,MaNhaSanXuat, TenSanPham, CauHinh, MoreImages, Gia, SoLuong, LuotView, TinhTrang, GhiChu).
* Thông tin LoaiSanPham (MaLoaiSanPham, TenLoaiSanPham, TinhTrang).
* Thông tin NhaSanXuat (MaNhaSanXuat, Ten NhaSanXuat, TinhTrang).
* Thông tin HoaDon (ID, NgayTao, NgayDat, NguoiNhan, SDT, DiaChi, TongTien, TrangThai).
* Thông tin ChiTietHoaDon (OrderID, MaSanPham, TenSanPham, SoLuong, Gia, TongTien).

***b) Yêu cầu tính toán***

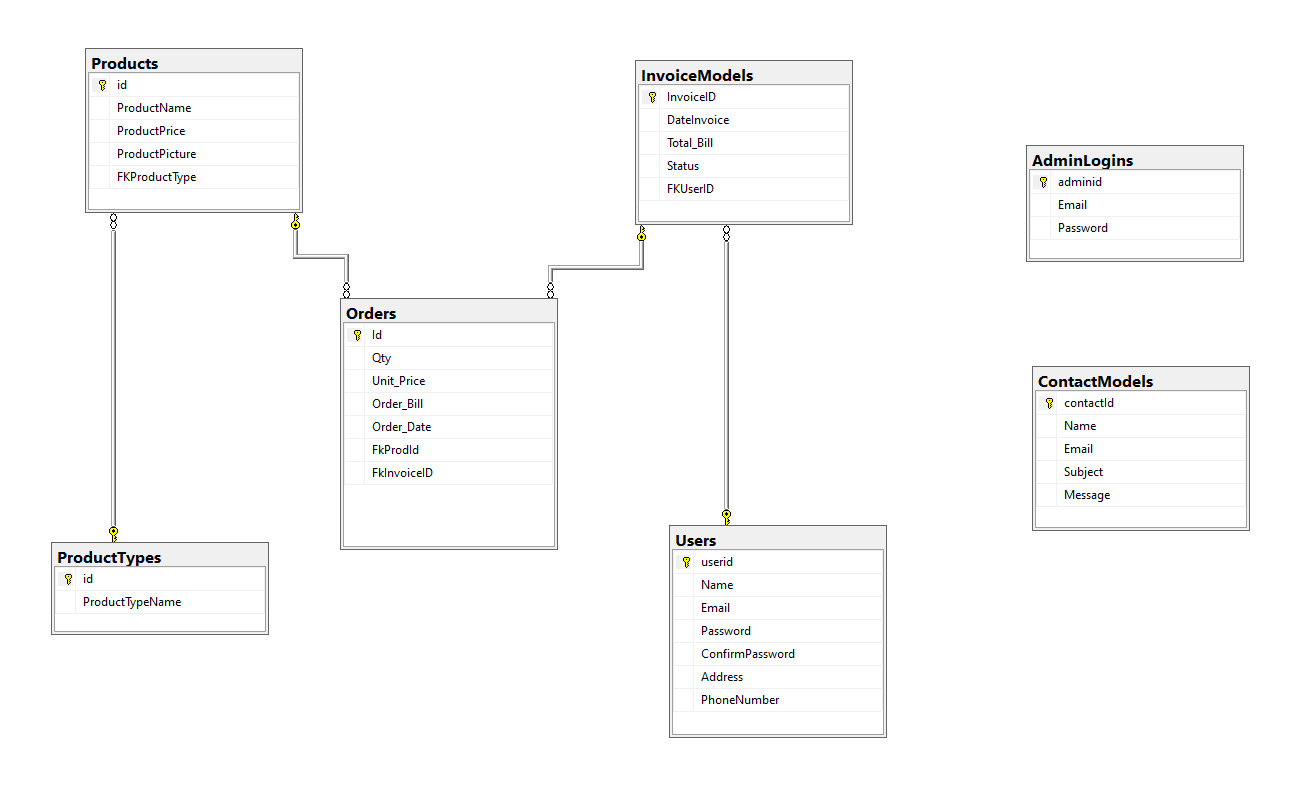
* Tính tiền tự động cho các sản phẩm mà khách hàng mua. Mỗi khi khách hàng mua một sản phẩm, sản phẩm này sẽ được lưu vào giỏ hàng để khách hàng dễ quản lý. Hệ thống tự động tính tổng giá trị giỏ hàng dựa trên việc lấy giá bán của từng sản phẩm lưu trong cơ sở dữ liệu và số lượng mỗi sản phẩm mà khách hàng mua, từ đó tính tổng tiền của tất cả các mặt hàng được mua để khách hàng theo dõi. Việc tính toán phải được thực hiện hoàn toàn tự động. Nếu khách hàng tiến hành đặt hàng, các kết quả tính toán này phải được lưu trong đơn đặt hàng.
* Kết quả tính toán: Tổng số tiền khách hàng phải trả cho mỗi lần mua.

***c)Yêu cầu tra cứu***

* Chọn thông tin tra cứu về sản phẩm.
* Kết quả xuất ra sau khi tra cứu: Danh sách các mặt hàng (nếu có) theo tiêu chuẩn đã được khách hàng đưa ra.

3.4 Thiết kế cơ sở dữ liệu

3.4.1 Mô hình cơ sở dữ liệu



###### ***Hình 4: Biểu đồ dữ liệu quan hệ***

3.4.2 Danh sách các lớp đối tượng

1. ***Lớp nhà sản xuất***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| **1** | MaNhaSanXuat | Nvarchar(50) | Khóa chính | Mã nhà sản xuất |
| **2** | TenNhaSanXuat | Nvarchar(50) |  | Tên nhà sản xuất |
| **3** | TinhTrang | Nvarchar(50) |  | Tình trạng |

*Bảng 3-1: Danh sách các thuộc tính lớp Nhà sản xuất*

1. ***Lớp Loại sản phẩm***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| **1** | MaLoaiSanPham | Nvarchar(50) | Khóa chính | Mã loại sản phẩm |
| **2** | TenLoaiSanPham | Nvarchar(50) |  | Tên loại sản phẩm |
| **3** | TinhTrang | Nvarchar(50) |  | Tình trạng |

*Bảng 3-2: Danh sách các thuộc tính lớp Loại sản phẩm*

1. ***Lớp sản phẩm***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| **1** | MaSanPham | Nvarchar(50) | Khóa chính | Mã sản phẩm |
| **2** | MaLoaiSanPham | Nvarchar(50) |  | Mã loại sản phẩm |
| **3** | MaNhaSanXuat | Nvarchar(50) |  | Mã nhà sản xuất |
| **4** | TenSanPham | Nvarchar(MAX) |  | Tên sản phẩm |
| **5** | CauHinh | Nvarchar(MAX) |  | Cấu Hình |
| **6** | MoreImages | Xml |  | Nhiều hình ảnh |
| **7** | Gia | Int |  | Giá |
| **8** | SoLuongBan | Int |  | Số lượng bán |
| **9** | LuotView | Int |  | Lượt xem |
| **10** | TinhTrang | Nchar(10) |  | Tình trạng |
| **11** | GhiChu | Nvarchar(50) |  | Ghi Chú |

*Bảng 3-3: Danh sách các thuộc tính lớp Sản phẩm*

1. ***Lớp Giỏ hàng***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| **1** | IdGH | Int | Khóa chính | Mã giỏ hàng |
| **2** | MaTaiKhoan | Nchar(128) |  | Mã tài khoản |
| **3** | MaSanPham | Nvarchar(50) |  | Mã sản phẩm |
| **4** | TenSanPham | Nvarchar(MAX) |  | Tên sản phẩm |
| **5** | SoLuong | Int |  | Số lượng |
| **6** | Gia | Int |  | Giá |
| **7** | TongTien | Int |  | Tổng tiền |

*Bảng 3-4: Danh sách các thuộc tính lớp Giỏ hàng*

1. ***Lớp Hóa Đơn***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| **1** | ID | Bigint | Khóa chính | Mã hóa đơn |
| **2** | NgayTao | Datetime |  | Ngày tạo |
| **3** | NguoiDat | Nvarchar(50) |  | Người đặt |
| **4** | NguoiNhan | Nvarchar(50) |  | Người nhận |
| **5** | SDT | varchar(50) |  | Số điện thoại |
| **6** | DiaChi | Nvarchar(50) |  | Địa chỉ |
| **7** | TongTien | Bigint |  | Tổng tiền |
| **8** | TrangThai | Int |  | Trạng thái |

*Bảng 3-5: Danh sách các thuộc tính lớp Hóa đơn*

1. ***Lớp Chi Tiết Hóa Đơn***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Tên trường | Kiểu dữ liệu | Ràng buộc | Mô tả |
| **1** | MaSanPham | Nvarchar(50) | Khóa chính | Mã sản phẩm |
| **2** | ID | Bigint | Khóa chính | Mã hóa đơn |
| **3** | TenSanPham | Nvarchar(MAX) |  | Tên sản phẩm |
| **4** | SoLuong | Int |  | Số lượng |
| **5** | Gia | Int |  | Giá |
| **6** | TongTien | Int |  | Tổng tiền |

*Bảng 3-6: Danh sách các thuộc tính lớp Chi tiết hóa đơn*

3.5 Thiết kế và đặc tả giao diện

3.5.1 Giao diện trang chủ

A picture containing text, screenshot, monitor

Description automatically generated

***Hình 5: Giao diện trang chủ***

* Giao diện trang chủ gồm những thông tin hiện thị cho khách hàng xem, tìm kiếm phục vụ nhu cầu của khách hàng khi đến với website.
* Hiện thị một số danh mục sản phẩm và loại sản phẩm, nhà sản xuất.

3.5.2 Giao diện trang đăng nhập

Graphical user interface, text

Description automatically generated

###### ***Hình 6: Giao diện trang đăng nhập***

* Mục đích: để thực hiện việc đặt hàng (đối với khách hàng) hoặc quản lý sản phẩm (đối với quản trị viên).
* Khách hàng muốn mua hàng thì phải đăng nhập vào hệ thống bằng địa chỉ email và password đã đăng ký.
* Khi đăng nhập thành công:
* Đối với khách hàng: Tên khách hàng sẽ được hiện thị lên trên tiêu đề của trang ngược lại nếu đăng nhập sai thì hệ thống sẽ yêu cầu nhập lại.
* Đối với quản trị: Tên Admin sẽ được hiện thị lên trên tiêu đề của trang ngược lại nếu đăng nhập sai thì hệ thống sẽ yêu cầu nhập lại.
* Khi hoàn tất quá trình đăng nhập:
* Đối với khách hàng: Thì hệ thống sẽ trở lại trang chủ để cho bạn tìm kiếm sản phẩm muốn mua.
* Đối với quản trị: Thì hệ thống sẽ truy cập trực tiếp vào trang quản trị.

3.5.3 Giao diện trang đăng ký

A screenshot of a computer

Description automatically generated

###### ***Hình 7: Giao diện trang đăng ký***

* Mục đích: đăng ký thông tin để thực hiện việc mua sản phẩm tại website.
* Ràng buộc: thông tin phải chính xác.
* Tại trang đăng ký thông tin khách hàng sẽ đăng ký những thông tin mà hệ thống yêu cầu. Sau khi đăng ký xong thông tin sẽ được hệ thống xử lý khách hàng quay về trang đăng nhập để đăng nhập thực hiện mua hàng.
* Thông tin đăng ký phải đăng ký đầy đủ thông tin nếu không điền đủ thông tin thì hệ thống sẽ báo bạn phải điền đủ thông tin. Đối với email thì hệ thống có kiểm tra bạn nhập email có hợp lệ không nếu không hợp lệ yêu cầu bạn nhập lại. Khi nhập mật khẩu bạn phải nhập mật khẩu đúng nếu không trùng khớp nhau hệ thống sẽ phản hồi lại mật khẩu không hợp lệ.

3.5.4 Giao diện trang danh mục sản phẩm

Graphical user interface, application

Description automatically generated

###### ***Hình 8: Giao diện trang danh mục sản phẩm***

* Mục đích: hiện thị danh sách sản phẩm của một dòng sản phẩm nào đó.
* Để thuận lợi cho việc lựa chọn sản phẩm, hệ thống có rất nhiều cách tìm kiếm sản phẩm nhanh đáp ứng nhu cầu khách hàng.
* Khách hàng muốn mua một sản phẩm theo một tên loại sản phẩm, tên nhà sản xuất, giá nào đó thì khách hàng lựa chọn sản phẩm mà mình muốn mua tại danh mục sản phẩm. Tại đây có rất nhiều loại điện thoai khác nhau đáp ứng được nhu cầu khách hàng như: Iphone, Vivo, Xiaomi, Samsung, Oppo.
* Khi khách hàng lựa chọn một loại sản phẩm nào đó sẽ có rất nhiều sản phẩm.

3.5.5 Giao diện trang chi tiết sản phẩm

Graphical user interface, application

Description automatically generated

###### ***Hình9: Giao diện trang chi tiết sản phẩm***

* Mục đích: hiện thị thông tin chi tiết của một sản phẩm như: Bộ nhớ trong, Pin, RAM, SIM, hệ điều hành,...
* Khi muốn xem chi tiết thông tin sản phẩm bạn chọn chi tiết sẽ có các thông tin liên quan đến sản phẩm.

3.5.6 Giao diện trang giỏ hàng

Graphical user interface, application

Description automatically generated

###### ***Hình 10: Giao diện trang giỏ hàng***

* Mục đích: lưu thông tin, hiện thị thông tin sản phẩm đã chọn mua.
* Khi chọn thêm sản phẩm hệ thống sẽ chuyển đến trang giỏ hàng thông tin của sản phẩm mà khách hàng mua sẽ được hiện trong giỏ hàng.
* Các sản phẩm đã có trong giỏ hàng khách hàng có thể thay đổi số lượng của từng sản phẩm, nếu muốn loại bỏ một sản phẩm nào đó trong giỏ hàng bạn chọn “xóa” .
* Sau khi đã mua xong các sản phẩm, bạn chọn thanh toán hệ thống sẽ chuyển đến trang đặt hàng để thực hiện tiếp quá trình mua hàng.

3.5.7 Giao diện trang đặt hàng và thanh toán

Graphical user interface

Description automatically generated with low confidence

###### ***Hình 11: Giao diện trang đặt hàng***

Graphical user interface, text, application

Description automatically generated

###### ***Hình 12: Giao diện trang thanh toán***

* Mục đích: hoàn tất một số thông tin khi xác nhận mua hàng.
* Khi đăng nhập vào hệ thống tên khách hàng được hiện trên thanh tiêu đề của trang.Tại trang giỏ hàng khách hàng chọn nút thanh toán hệ thống sẽ chuyển đến trang đặt hàng để khách hàng thực hiện những thông tin cuối cùng để hoàn tất quá trình mua hàng.
* Khách hàng điền đầy đủ thông tin vào các ô số điện thoại, địa chỉ và chọn ngày giao hàng. Ngoài ra còn có yêu cầu nào khác thì khách hàng ghi nội dung vào ô yêu cầu.
* Thông tin cần kiểm tra là tên khách hàng có đúng như lúc đầu khách hàng đăng ký không, số lượng và tổng số tiền cần thanh toán có khớp với giá hiện hành không.

3.5.8 Giao diện trang quản trị

A picture containing text, monitor, screenshot, screen

Description automatically generated

***Hình 13: Giao diện trang quản trị***

* Mục đích: hiện thị các chức năng của hệ thống như : quản lý sản phẩm, quản lý hóa đơn.

3.5.9 Giao diện trang quản lý sản phẩm

A screenshot of a computer

Description automatically generated

***Hình 14: Giao diện trang quản lý sản phẩm***

* Mục đích: thực hiện nhập, sửa, xóa thông tin sản phẩm.
* Ràng buộc: đăng nhập dưới quyền quản trị.
* Mô tả:
* Nhập đầy đủ thông tin và nhấn nút “Thêm” để lưu thông tin sản phẩm đồng thời chọn sản phẩm khuyến mại.
* Sửa, xóa: chọn sản phẩm rồi nhấn nút “Sửa” hoặc “Xóa” để sửa hoặc xóa thông tin của sản phẩm.

3.5.10 Giao diện trang thêm, sửa, xóa sản phẩm

A picture containing text, monitor, screenshot, screen

Description automatically generated  
***Hình 15: Giao diện trang thêm sản phẩm***

A screenshot of a computer

Description automatically generated

***Hình 16: Giao diện trang sửa sản phẩm***

A picture containing text, monitor, screenshot, screen

Description automatically generated

***Hình 17: Giao diện trang xóa sản phẩm***

* Mục đích: thực hiện việc thêm, sửa, xóa sản phẩm.
* Ràng buộc: đăng nhập dưới quyền của người quản trị.
* Mô tả:
* Thêm: Thực hiện nhập sản phẩm bạn nhập mã sản phẩm, mã loại sản phẩm, tên sản phẩm, giá, nhà sản xuất, ảnh (MoreImages), ghi chú sau đó nhấn nút “thêm” hệ thống sẽ cập nhật thông tin mà bạn vừa nhập vào hệ thống.
* Sửa: Sửa thông tin sản phẩm.
* Xóa: Xóa sản phẩm.

3.5.11 Giao diện trang quản lý nhà sản xuất

A picture containing text, monitor, screenshot, screen

Description automatically generated

***Hình 18: Giao diện trang quản lý nhà sản xuất***

* Mục đích: thực hiện nhập, sửa, xóa thông tin nhà cung cấp.
* Ràng buộc: đăng nhập dưới quyền là người quản trị.
* Mô tả:
* Thực hiện nhập nhà cung cấp bạn nhập mã nhà sản xuất, tên nhà sản xuất, tình trạng sau đó nhấn nút “Thêm” hệ thống sẽ cập nhật thông tin mà bạn vừa nhập vào hệ thống.
* Thực hiện sửa, xóa nhà sản xuất bạn chọn thông tin muốn sửa, xóa sau đó thông tin sẽ hiện lên cho bạn thực hiện. Khi hoàn tất nhấn nút “Xác nhận chỉnh sửa”, “Xác nhận xóa”.
* Khi lựa chọn “ Hiện thị danh sách nhà sản xuất ” các thông tin liên quan đến nhà nhà sản xuất sẽ được hiện thị đầy đủ thành một bảng cho bạn xem và kiểm tra.

3.5.12 Giao diện trang quản lý loại sản phẩm

A picture containing text, screenshot, monitor, screen

Description automatically generated

***Hình 19: Giao diện trang quản lý loại sản phẩm***

* Mục đích: thực hiện nhập, sửa, xóa thông tin loại sản phẩm.
* Ràng buộc: đăng nhập dưới quyền là người quản trị.
* Mô tả:
* Thực hiện nhập nhà cung cấp bạn nhập mã loại sản phẩm, tên loại sản phẩm, tình trạng sau đó nhấn nút “Thêm” hệ thống sẽ cập nhật thông tin mà bạn vừa nhập vào hệ thống.
* Thực hiện sửa, xóa nhà sản xuất bạn chọn thông tin muốn sửa, xóa sau đó thông tin sẽ hiện lên cho bạn thực hiện. Khi hoàn tất nhấn nút “Xác nhận chỉnh sửa”, “Xác nhận xóa”.
* Khi lựa chọn “ Hiện thị danh loại sản phẩm” các thông tin liên quan đến nhà nhà sản xuất sẽ được hiện thị đầy đủ thành một bảng cho bạn xem và kiểm tra.

# CHƯƠNG 4: KẾT LUẬN

4.1 Kết quả đạt được của đề tài

* Xây dựng website Bán điện thoại.
* Giao diện đẹp, màu sắc hài hòa, dễ nhìn, dễ sử dụng.
* Xây dựng được các chức năng cần thiết cho hệ thống: nhập, sửa, xóa thông tin.

4.2 Hạn chế của đề tài

* Một số chức năng còn chưa tối ưu, thực hiện được.
* Chưa có diễn đàn trao đổi cho khách hàng.
* Chưa áp dụng được các biện pháp bảo mật cho website.
* Chưa thực hiện thanh toán online.

4.3 Kết luận

Trong quá trình nguyên cứu và vận dụng các kiến thức đã được học. Chúng em đã hoàn thành phần mềm “Quản lý cửa hàng bán điện thoại” nhằm mục đích hoạt động kinh doanh và ôn lại những kiến thức đã học. Chúng em hy vọng sau hoạt động nhóm này là tiền đề để chúng em phát triển mạnh hơn về kiến thức và chất lượng phần mềm.

Trong quá trình hoàn thành phần mềm với kinh nghiệm thực tế chưa nhiều nên báo cáo không tránh khỏi những sai sót, rất mong sự góp ý của thầy.

Chúng em xin chân thành cảm ơn.

**TÀI LIỆU THAM KHẢO**

* Phạm Hữu Khang, Lập trình ASP.Net MVC. Hà Nội, Việt Nam: Nhà xuất bản Lao Động Xã Hội, 2005.
* Bùi Chí Thành , *Giáo trình Phát triển ứng dụng web.*
* Đoàn Thiện Ngân, Phạm Hữu Khang, Lập trình ASP.NET MVC , Nhà xuất bản lao động xã hội, 2004.
* Phạm Hữu Khanh, [Microsoft SQL Server 2008 - Quản Trị Cơ Sở Dữ Liệu](https://www.vinabook.com/microsoft-sql-server-2008-quan-tri-co-so-du-lieu-tap-1-p37708.html), NXB Lao động - Xã hội,2004.