# LỜI CẢM ƠN

# LỜI MỞ ĐẦU

Ngày nay, với sự phát triển mạnh mẽ của CNTT và những lợi ích từ việc ứng dụng nó đi vào đời sống, máy tính và Internet đã không còn là những khái niệm xa lạ đối với mọi người, mà nó dần trở thành công cụ hữu ích hỗ trợ con người học tập, giải trí và làm việc.

Trong nền kinh tế hiện nay, với xu thế toàn cầu hóa nền kinh tế thế giới, mọi mặt của đời sống ngày càng được nâng cao, đặc biệt nhu cầu mua sắm các loại mặt hàng của con người ngày càng tăng. Môi trường Internet tự do, rộng lớn là mãnh đất màu mỡ cho các doanh nghiệp có thể quảng bá sản phẩm của mình đến người tiêu dùng một cách nhanh chóng, ít tốn kém, giúp mở rộng hoạt động kinh doanh, tăng trưởng doanh thu.

Năm 2021, khi con người vẫn còn đang đối mặt với đại dịch Covid thì việc quảng bá các mặt hàng online đã được đẩy mạnh để đáp ứng nhu cầu mọi người, sự tiện lợi, an toàn trong mùa dịch mà không cần phải đến nơi đông người để mua sắm đã góp phần thay đổi đời sống một cách hiện đại hóa.

Với mục tiêu tìm hiểu cấu trúc, cách thức hoạt động và khái quát hóa về thương mại điện tử trên mạng Internet ở Việt Nam, em đã xây dựng và đưa vào thử nghiệm thực tế một website bán thức ăn nhanh. Đây sẽ là bước đi đầu tiên của em trước khi tiến cận với công việc quản lý website của một cửa hàng thực sự.

**MỤC LỤC**

Contents

[LỜI CẢM ƠN 6](#_Toc83840098)

[LỜI MỞ ĐẦU 7](#_Toc83840099)

[CHƯƠNG I: CƠ SỞ LÝ THUYẾT 11](#_Toc83840100)

[**1.** **Giới thiệu** 11](#_Toc83840101)

[**2.** **Tìm hiểu lý thuyết** 11](#_Toc83840102)

[CHƯƠNG II: PHÂN TÍCH THIẾT KẾ HỆ THỐNG THÔNG TIN WEBSITE BÁN THỨC ĂN NHANH 21](#_Toc83840103)

[**1.** **Mô tả hệ thống và chức năng** 21](#_Toc83840104)

[**2.** **Sơ đồ Usecase** 22](#_Toc83840105)

[**3.** **Sơ đồ Class** 25](#_Toc83840106)

[**4.** **Cơ sở dữ liệu** 26](#_Toc83840107)

[CHƯƠNG III: THIẾT KẾ VÀ XÂY DỰNG WEBSITE BÁN THỨC ĂN NHANH 29](#_Toc83840108)

[**1.** **Chi tiết các chức năng và phương thức** 29](#_Toc83840109)

[**2.** **Chi tiết giao diện các chức năng** 39](#_Toc83840110)

[CHƯƠNG IV: THỰC NGHIỆM VÀ TRIỂN KHAI 50](#_Toc83840111)

[**1.** **Demo chức năng chính** 50](#_Toc83840112)

[CHƯƠNG V: KẾT LUẬN 55](#_Toc83840113)

[**1.** **Kết luận** 55](#_Toc83840114)

[**2.** **Hướng phát triển** 55](#_Toc83840115)

**DANH MỤC CÁC HÌNH**

[Hình 1 3 thành phần cơ bản để thiết kế front-end 13](#_Toc83840045)

[Hình 2 Quá trình chuyển đổi MSIL code thành native code (C#) 14](#_Toc83840046)

[Hình 3 Microsoft .NET Framework 15](#_Toc83840047)

[Hình 4 Microsoft SQL Server dùng để lưu trữ dữ liệu 17](#_Toc83840048)

[Hình 5 Mô hình MVC trong ASP.NET 20](#_Toc83840049)

[Hình 6 Usecase Tổng quát 22](#_Toc83840050)

[Hình 7 Usecase Truy cập website 22](#_Toc83840051)

[Hình 8 Usecase Đặt món 23](#_Toc83840052)

[Hình 9 Usecase Quản lý đơn hàng 23](#_Toc83840053)

[Hình 10 Usecase Quản lý món ăn 24](#_Toc83840054)

[Hình 11 Usecase Quản lý thông tin khách hàng 24](#_Toc83840055)

[Hình 12 Sơ đồ Class website bán thức ăn nhanh 25](#_Toc83840056)

[Hình 13 Lược đồ CSDL quan hệ của website bán thức ăn nhanh 26](#_Toc83840057)

[Hình 14 dbKiwiFastfood.dbml 29](#_Toc83840058)

[Hình 15 Template mẫu sử dụng cho website Kiwi’s fast food 30](#_Toc83840059)

[Hình 16 Giao diện trang chủ (1) 39](#_Toc83840060)

[Hình 17 Giao diện trang chủ (2) 40](#_Toc83840061)

[Hình 18 Giao diện trang chủ (3) 40](#_Toc83840062)

[Hình 19 Giao diện trang thực đơn (1) 41](#_Toc83840063)

[Hình 20 Giao diện trang thực đơn (2) 41](#_Toc83840064)

[Hình 21 Giao diện trang liên hệ (1) 42](#_Toc83840065)

[Hình 22 Giao diện trang liên hệ (2) 42](#_Toc83840066)

[Hình 23 Giao diện thông tin chi tiết món ăn 43](#_Toc83840067)

[Hình 24 Giao diện trang giỏ hàng (1) 43](#_Toc83840068)

[Hình 25 Giao diện trang giỏ hàng (2) 44](#_Toc83840069)

[Hình 26 Giao diện trang đăng ký tài khoản 44](#_Toc83840070)

[Hình 27 Giao diện trang đăng nhập 45](#_Toc83840071)

[Hình 28 Giao diện trang đặt hàng (1) 45](#_Toc83840072)

[Hình 29 Giao diện trang đặt hàng (2) 46](#_Toc83840073)

[Hình 30 Giao diện trang đăng nhập hệ thống quản trị 46](#_Toc83840074)

[Hình 31 Giao diện trang quản lý đơn hàng 47](#_Toc83840075)

[Hình 32 Giao diện trang quản lý loại món 47](#_Toc83840076)

[Hình 33 Giao diện quản lý khách hàng 48](#_Toc83840077)

[Hình 34 Quản lý thông tin món ăn 48](#_Toc83840078)

[Hình 35 Giao diện trang thêm món ăn 49](#_Toc83840079)

[Hình 36 Giao diện trang điều chỉnh thông tin món ăn 49](#_Toc83840080)

[Hình 37 Demo chức năng người dùng (1) 50](#_Toc83840081)

[Hình 38 Demo chức năng người dùng (2) 50](#_Toc83840082)

[Hình 39 Demo chức năng người dùng (3) 51](#_Toc83840083)

[Hình 40 Demo chức năng người dùng (4) 51](#_Toc83840084)

[Hình 41 Demo chức năng người dùng (5) 52](#_Toc83840085)

[Hình 42 Demo chức năng người dùng (6) 52](#_Toc83840086)

[Hình 43 Demo chức năng người dùng (7) 53](#_Toc83840087)

[Hình 44 Demo chức năng người dùng (8) 53](#_Toc83840088)

[Hình 45 Demo chức năng người quản trị (1) 54](#_Toc83840089)

[Hình 46 Demo chức năng người quản trị (2) 54](#_Toc83840090)

[Hình 47 Demo chức năng người quản trị (3) 54](#_Toc83840091)

CHƯƠNG I: CƠ SỞ LÝ THUYẾT

1. **Giới thiệu**
   1. **Lý do chọn đề tài**

Vào những năm gần đây, việc ứng dụng CNTT về quản lý web ngày càng phát triển mạnh mẽ, giúp cho các cửa hàng kiếm được lợi nhuận nhiều hơn nhờ vào các trang website bán hàng trực tuyến. Hiểu được yêu cầu đó, em quyết định chọn đề tài “Xây dựng website bán thức ăn nhanh” làm mục tiêu chính cho đồ án lần này, với mong muốn kinh doanh các món ăn nhanh như gà rán, pizza, xúc xích xông khói, mì Ý và nước ngọt v.v... Ngoài ra, với giao diện website thân thiện cùng với tông màu đơn giản sẽ tạo thiện cảm cho khách hàng, giúp khách hàng đặt món tại nhà mà không cần tốn nhiều thời gian ra cửa hàng. Đó là sự tiện lợi mà mọi khách hàng nào cũng muốn!

* 1. **Mục tiêu nghiên cứu**
* Xây dựng được một trang web bán hàng hoàn chỉnh.
* Hiểu được cấu trúc full-stack, sự tương tác giữa front-end và back-end.
* Nghiên cứu các công nghệ phát triển Website.
* Giải quyết tối ưu hóa quá trình QL bán hàng và bán hàng.
  1. **Phương pháp nghiên cứu**
* Tìm hiểu, đọc thêm tài liệu cụ thể về front-end và back-end.
* Trang bị kiến thức cơ bản về HTML/CSS/JS và ngôn ngữ lập trình C#.
* Phân tích, hiểu rõ quá trình tương tác giữa người dùng và hệ thống website.
* Nghiên cứu về giao diện tiện lợi, dễ sử dụng, thu hút khách hàng.

1. **Tìm hiểu lý thuyết**
   1. **Tìm hiểu về Full-stack**

Full-stack là sự tổng hợp từ kiến thức, sự hiểu biết trực quan và sâu sắc về cả front-end và back-end, cũng như nắm vững các khái niệm. Một Full-stack Web Developer là người có thể làm việc trên cả front-end và back-end của một ứng dụng. Front-end nói chung là phần mà người dùng có thể thấy được và tương tác được, còn back-end là phần ứng dụng xử lý logic, tương tác cơ sở dữ liệu, chứng thực người dùng, cấu hình máy chủ, v.v…

Để có thể lập trình full-stack thì cần có các kỹ năng như sau:

* + 1. Html/Css/JavaScript

- HTML hay còn được gọi là HyperText Markup Language – Ngôn ngữ đánh dấu siêu văn bản, là ngôn ngữ được sử dụng cho các tài liệu web. HTML không phải là một ngôn ngữ lập trình, HTML giống như một ngôn ngữ xác định đâu là ý nghĩa, mục đích và cấu trúc của một tài liệu. Cùng với CSS và JavaScript, HTML tạo ra bộ ba nền tảng kỹ thuật cho các website. Chức năng chính của HTML là xây dựng cấu trúc siêu văn bản trên một website, hoặc khai báo các tập tin kỹ thuật số (media) như hình ảnh, video, nhạc. HTML thường được dùng để phân chia các đoạn văn, heading, links, blockquotes,… Với HTML, chúng ta có thể:

* Thêm tiêu đề, định dạng đoạn văn, ngắt dòng điều khiển.
* Tạo danh sách, nhấn mạnh văn bản, tạo ký tự đặc biệt, chèn hình ảnh, tạo liên kết.
* Xây dựng bảng, điều khiển một số kiểu mẫu

- CSS là viết tắt của cụm từ “Cascading Style Sheets”, tạm dịch: ngôn ngữ tạo phong cách cho trang web. Ngôn ngữ lập trình này quy định cách các thành phần HTML của trang web thực sự sẽ xuất hiện trên front-end như thế nào. Hiểu đơn giản, CSS sẽ giúp webmaster xác định styles và định nghĩa nhiều loại nội dung của website. CSS được tạo ra để kết hợp với ngôn ngữ markup HTML để tạo phong cách cho trang web.

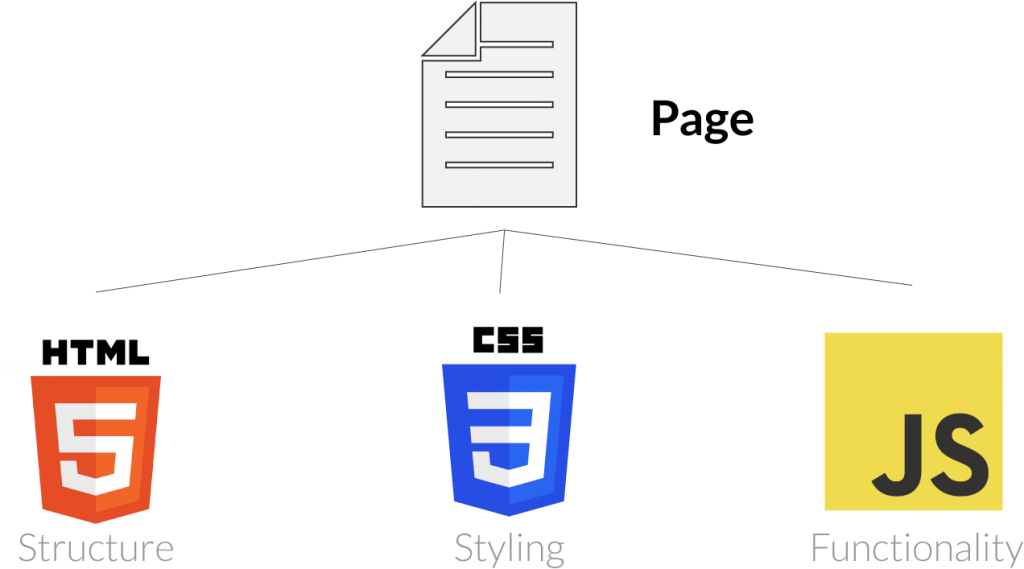
Có 3 loại style CSS chính:

* Style CSS Internal: là style được tải lên mỗi khi trang web được refresh.
* Style CSS Inline: có thể chỉnh sửa một yếu tố nào đó mà không cần truy cập trực tiếp vào file CSS.
* External style: có thể tạo phong cách ở file khác áp dụng CSS vào trang tùy thích. External style sẽ cải thiện thời gian tải trang rất nhiều.

- JavaScript hay còn gọi là “JS” là một ngôn ngữ lập trình được Brendan Eich (đồng sáng lập dự án Mozilla, quỹ Mozilla và tập đoàn Mozilla) cho ra mắt vào năm 1995 với tên LiveScript. JS có tác dụng giúp chuyển website từ trạng thái tĩnh sang động, tạo tương tác để cải thiện hiệu suất máy chủ và nâng cao trải nghiệm người dùng. Hiểu đơn giản, JavaScript là ngôn ngữ được sử dụng rộng rãi khi kết hợp với HTML/CSS để thiết kế web động. Sử dụng JavaScript, chúng ta sẽ:

* Dễ dàng bắt đầu với các bước nhỏ, với thư viện ảnh, bố cục có tính thay đổi … nhờ sự linh hoạt của JavaScript.
* Có thể tạo ra các trò chơi, hoạt họa 2D hoặc 3D, ứng dụng cơ sở dữ liệu toàn diện,…
* Tăng cường các hành vi và kiểm soát mặc định của trình duyệt.

- Một số front-end framework nổi tiếng: React.js, Vue.js, Angular.js.



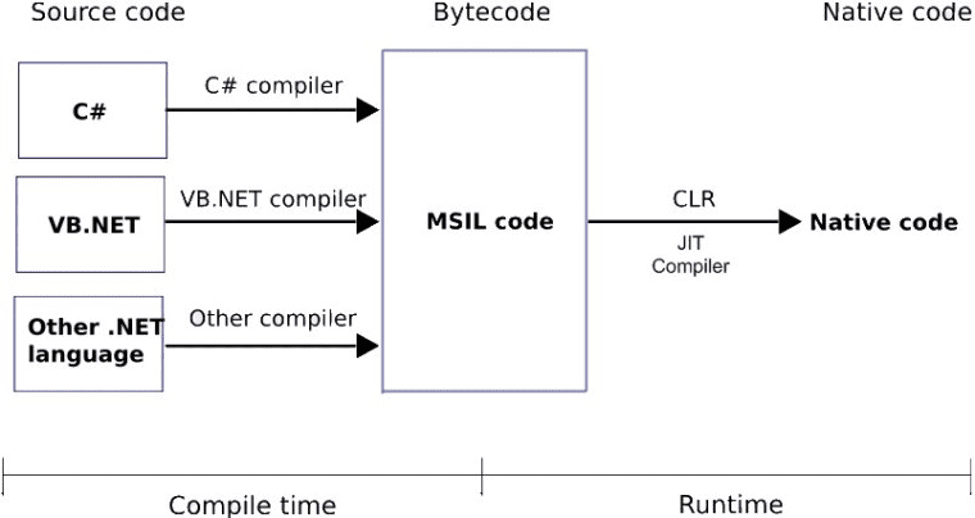
Hình 3 thành phần cơ bản để thiết kế front-end

* + 1. Back-end Language (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. C# là ngôn ngữ lập trình hiện đại, hướng đối tượng và được xây dựng trên nền tảng của hai ngôn ngữ mạnh nhất là C++ và Java.

- Trong các ứng dụng Windows truyền thống, mã nguồn chương trình được biên dịch trực tiếp thành mã thực thi của hệ điều hành.

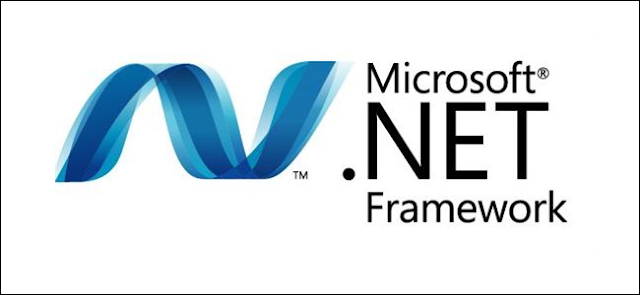
- Trong các ứng dụng sử dụng .NET Framework, mã nguồn chương trình (C#, VB.NET) được biên dịch thành mã ngôn ngữ trung gian MSIL (Microsoft intermediate language). Sau đó mã này được biên dịch bởi Common Language Runtime (CLR) để trở thành mã thực thi của hệ điều hành. Hình bên dưới thể hiện quá trình chuyển đổi MSIL code thành native code.



Hình 2 Quá trình chuyển đổi MSIL code thành native code (C#)

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

* Đặc trưng của ngôn ngữ C#:
* C# là ngôn ngữ đơn giản:
* C# loại bỏ một vài sự phức tạp và rối rắm của những ngôn ngữ như Java và c++, bao gồm việc loại bỏ những macro, những template, đa kế thừa, và lớp cơ sở ảo (virtual base class).
* Ngôn ngữ C# đơn giản vì nó dựa trên nền tảng C và C++. Nếu chúng ta thân thiện với C và C++ hoặc 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.
* C# là ngôn ngữ hiện đại:
* C# có thể xử lý ngoại lệ, thu gom bộ nhớ tự động, những kiểu dữ liệu mở rộng và bảo mật mã nguồn.
* C# là một ngôn ngữ lập trình thuần hướng đối tượng:
* Lập trình hướng đối tượng (OOP: Object-oriented programming) 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.
* C# là một ngôn ngữ ít từ khóa:
* C# là ngôn ngữ sử dụng giới hạn những từ khóa. Phần lớn các từ khóa được sử dụng để mô tả thông tin. Chúng ta có thể nghĩ rằng một ngôn ngữ có nhiều từ khóa thì sẽ mạnh hơn. Điều này không phải sự thật, ít nhất là trong trường hợp ngôn ngữ C#, chúng ta có thể tìm thấy rằng ngôn ngữ này có thể được sử dụng để làm bất cứ nhiệm vụ nào.
* Nền tảng .NET:
* .NET Framework được Microsoft đưa ra chính thức từ năm 2002. .NET Framework chỉ hoạt động trên Windows. Những nền tảng ứng dụng như WPF, Winforms, ASP.NET(1-4) hoạt động dựa trên .NET Framework.
* Mono là phiên bản cộng đồng nhằm mang .NET đến những nền tảng ngoài Windows. Mono được phát triển chủ yếu nhằm xây dựng những ứng dụng với giao diện người dùng và được sử dụng rất rộng rãi: Unity Game, Xamarin…
* Cho đến năm 2013, Microsoft định hướng đi đa nền tảng và phát triển .NET core. .NET core hiện được sử dụng trong các ứng dụng Universal Windows platform và ASP.NET Core. Từ đây, C# có thể được sử dụng để phát triển các loại ứng dụng đa nền tảng trên các hệ điều hành khác nhau (Windows, Linux, MacOS,…)



Hình Microsoft .NET Framework

* + 1. Database (Microsoft SQL Server)
* SQL:
* SQL là ngôn ngữ phi thủ tục, không yêu cầu cách thức truy cập cơ sở dữ liệu như thế nào. Tất cả các thông báo của SQL rất dễ dàng sử dụng và ít mắc lỗi.
* SQL cung cấp các tập lệnh phong phú cho các công việc hỏi đáp dữ liệu như:
* Chèn, xóa và cập nhật các hàng trong 1 quan hệ
* Tạp, thêm, xóa và sửa đổi các đối tượng trong của cơ sở dữ liệu.
* Điều khiển việc truy cấp tới cơ sở dữ liệu và các đối tượng của cơ sở dữ liệu để đảm bảo tính bảo mật, tính nhất quán và sự ràng buộc của cơ sở dữ liệu.
* Đối tượng của SQL Server là các bảng dữ liệu với các cột và các hàng. Cột được gọi là trường dữ liệu và hàng là bản ghi của bảng. Cột dữ liệu và kiểu dữ liệu xác định tạo nên cấu trúc của bảng. Khi bảng được tổ chức thành một hệ thống cho một mục đích sử dụng cụ thể vào công việc nào đó sẽ trở thành một cơ sở dữ liệu.
* Microsoft SQL Server:

- SQL server hay còn được gọi là Microsoft SQL Server, nó từ viết tắt của MS SQL Server. Đây chính là một loại phần mềm đã được phát triển bởi Microsoft và nó được sử dụng để có thể dễ dàng lưu trữ cho những dữ liệu dựa theo tiêu chuẩn RDBMS. Từ đó, người ta sẽ lưu trữ dữ liệu dựa vào tiêu chuẩn RDBMS và nó cũng là một  trong những hệ quản trị cơ sở dữ liệu dạng quan hệ đối tượng.

- SQL Server có khả năng cung cấp đầy đủ các công cụ cho việc quản  lý từ giao diện GUI đến sử dụng ngôn ngữ cho việc truy vấn SQL. Điểm mạnh của SQL điểm mạnh của nó là có nhiều nền tảng được kết hợp cùng như: ASP.NET, C# để xây dựng Winform cũng chính nó có khả năng hoạt động độc lập.

* Mục đích khi sử dụng SQL Server:

SQL Server thông thường được sử dụng cho mục đích lưu trữ dữ liệu. Ngoài ra, nó còn mang lại những tính năng làm việc giúp người dùng làm việc hiệu quả hơn như sau:

* Giúp người sử dụng có thể duy trì việc lưu trữ bền vững.
* Cho phép bạn tạo ra nhiều cơ sở dữ liệu hơn.
* Có khả năng phân tích dữ liệu bằng SSAS.
* Nó có khả năng bảo mật cao.
* Việc tạo ra được các báo cáo bằng SSRS — SQL Server Reporting Services sẽ được dễ dàng hơn.
* Các quá trình sẽ được thực hiện bằng SSIS — SQL Server Integration Services.
* Một số khái niệm cơ bản về các xử lý:
* Thông thường, các xử lý bên trong một ứng dụng thường có thể sẽ được chia thành 2 loại xử lý, bao gồm: xử lý trên máy chủ và xử lý trên trạm.
* Nó có khả năng đọc và cập nhật được các dữ liệu.
* Quá trình tính toán, hiển thị dữ liệu ở trên màn hình giao diện.
* Bạn có thể sử dụng được nhiều loại ngôn ngữ lập trình khác nhau như là: C#, Java,…
* Xử lý các tiêu chuẩn trên máy chủ Database Server
* Xử lý những yêu cầu liên quan đến ghi/đọc dữ liệu.
* Thực hiện quản lý đồng bộ mọi dữ liệu giữa những yêu cầu đọc ghi từ các máy trạm gửi tới.
* Thực hiện các dịch vụ quản trị dữ liệu tự động dựa theo định kỳ như backup/restore dữ liệu.



Hình Microsoft SQL Server dùng để lưu trữ dữ liệu

* 1. **Full-stack với ASP.NET MVC:**
     1. ASP.NET

- ASP.Net là một nền tảng phát triển web được cung cấp bởi Microsoft và được sử dụng để tạo các ứng dụng dựa trên web. ASP.Net được phát hành lần đầu tiên vào năm 2002. Các ứng dụng ASP.Net cũng có thể được viết bằng nhiều ngôn ngữ .Net. Chúng bao gồm C#, VB.Net và J#.

* Đặc điểm chính của khung ASP.NET:
* Chế độ mã phía sau – Đây là khái niệm phân tách thiết kế và mã. Bằng cách thực hiện việc phân tách này, việc duy trì ứng dụng ASP.Net trở nên dễ dàng hơn. Loại tệp chung của tệp ASP.Net là aspx. Giả sử chúng ta có một trang web có tên MyPage.aspx. Sẽ có một tệp khác gọi là MyPage.aspx.cs sẽ biểu thị phần mã của trang. Vì vậy, Visual Studio tạo các tệp riêng biệt cho mỗi trang web, một cho phần thiết kế và phần còn lại cho mã.
* Quản lý trạng thái – ASP.Net có cơ sở để kiểm soát quản lý trạng thái. HTTP được biết đến như một giao thức phi trạng thái. Hãy lấy một ví dụ về một ứng dụng giỏ hàng. Bây giờ, khi người dùng quyết định những gì muốn mua từ trang web, anh ta sẽ nhấn nút xác nhận. Ứng dụng cần ghi nhớ các mục mà người dùng chọn mua. Điều này được gọi là ghi nhớ trạng thái của một ứng dụng tại thời điểm hiện tại. HTTP là một giao thức phi trạng thái. Khi người dùng truy cập trang mua hàng, HTTP sẽ không lưu trữ thông tin trên các mục giỏ hàng. Cần mã hóa bổ sung để đảm bảo rằng các mặt hàng trong giỏ hàng có thể được chuyển đến trang mua hàng. Đôi khi việc thực hiện có thể trở nên phức tạp. Nhưng ASP.Net có thể thay mặt bạn quản lý trạng thái. Vì vậy, ASP.Net có thể nhớ các mục giỏ hàng và chuyển nó đến trang mua hàng.
* Bộ nhớ đệm – ASP.Net có thể thực hiện khái niệm về bộ đệm. Điều này cải thiện hiệu suất của ứng dụng. Bằng cách lưu trữ những trang mà người dùng thường yêu cầu có thể được lưu trữ ở một vị trí tạm thời. Các trang này có thể được truy xuất nhanh hơn, và phản hồi tốt hơn có thể được gửi đến người dùng. Vì vậy, bộ nhớ đệm có thể cải thiện đáng kể hiệu suất của một ứng dụng.

- ASP.Net là ngôn ngữ phát triển được sử dụng để xây dựng các ứng dụng dựa trên web. ASP.Net được thiết kế để hoạt động với giao thức HTTP tiêu chuẩn.

* + 1. ASP.NET mở rộng .NET

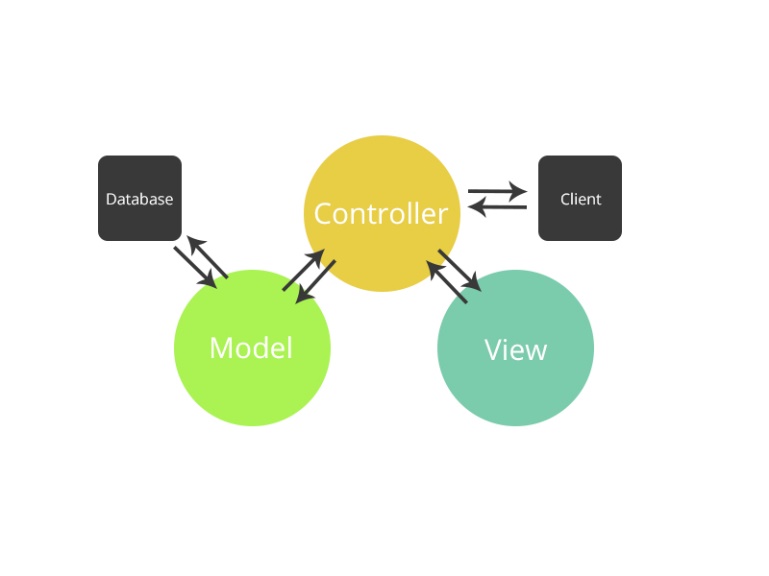
- ASP.NET mở rộng nền tảng .NET với các công cụ và thư viện dành riêng cho việc xây dựng các ứng dụng web.

* Đây là một số điều mà ASP.NET thêm vào nền tảng .NET:
* Khung cơ sở để xử lý các yêu cầu web.
* Cú pháp tạo khuôn mẫu trang web, được gọi là Razor, để xây dựng các trang web động bằng C #.
* Các thư viện cho các mẫu web phổ biến, chẳng hạn như [Model View Controller](https://vi.wikipedia.org/wiki/MVC) (MVC)
* Hệ thống xác thực bao gồm thư viện, cơ sở dữ liệu và trang mẫu để xử lý thông tin đăng nhập, bao gồm xác thực đa yếu tố và xác thực bên ngoài với [Google](https://www.google.com.vn/?hl=vi), [Twitter](https://twitter.com/?lang=vi), v.v.
* Phần mở rộng trình chỉnh sửa để cung cấp làm nổi bật cú pháp, hoàn thành mã và các chức năng khác dành riêng cho việc phát triển trang web.
  + 1. Ưu và nhược khi sử dụng mô hình ASP. NET MVC để phát triển Web
* Ưu điểm:
* Dễ dàng bảo trì mã, dễ dàng mở rộng và phát triển.
* Các thành phần của mô hình này có thể được kiểm tra hoàn toàn độc lập với người dùng.
* Dễ dàng hỗ trợ cho các khách hàng mới.
* Có thể thực hiện song song việc phát triển các thành phần khác nhau.
* Đơn giản hóa bằng cách chia web thành ba phần: Model, View, Controller.
* Chỉ sử dụng mẫu Front Controller để xử lý các yêu cầu của web thông qua một bộ điều khiển duy nhất.
* Hỗ trợ tốt nhất cho việc phát triển web theo hướng thử nghiệm.
* MVC hoạt động tốt với các ứng dụng được hỗ trợ bởi giới lập trình web.
* Thân thiện với công cụ tìm kiếm.
* Tất cả các đối tượng độc lập với nhau, do vậy bạn có thể kiểm tra một cách hoàn toàn riêng biệt.

- Không những vậy, MVC tương đối nhẹ và tiết kiệm diện tích băng thông bởi nó không cần sử dụng Viewstate. Điều đó giúp website hoạt động tốt và ổn định khi người dùng thực hiện quá nhiều thao tác tương tác như gửi hay nhận dữ liệu liên tục.

* Nhược điểm
* Khó đọc, thay đổi, kiểm tra đơn vị hoặc sử dụng lại mô hình này.
* Đôi lúc điều hướng khung có thể phức tạp vì nó giới thiệu các lớp trừu tượng mới, đòi hỏi người dùng phải thích ứng với các tiêu chí phân tách của MVC.
* Không hỗ trợ việc xác thực chính thức.
* Vừa làm tăng sự phức tạp vừa làm giảm hiệu quả của dữ liệu.
* Gây khó khăn khi sử dụng với giao diện người dùng hiện đại.
* Bắt buộc phải có nhiều lập trình viên để tiến hành lập trình song song.
* Cần có kiến ​​thức tổng hợp về công nghệ.
* Bảo trì nhiều mã trong Bộ điều khiển.

- MVC được sử dụng trong các dự án lớn. Với các dự án nhỏ, việc áp dụng mô hình MVC không được thích hợp bởi nó khá cồng kềnh. Nó cũng tiêu tốn nhiều thời gian để phát triển cũng như trung chuyển dữ liệu. Tuy nhiên, so với những mô hình khác, MVC vẫn là sự lựa chọn hàng đầu cho [lập trình ứng dụng](https://teky.edu.vn/blog/lap-trinh-ung-dung-di-dong/) nói chung và cả lập trình web nói riêng.



Hình Mô hình MVC trong ASP.NET

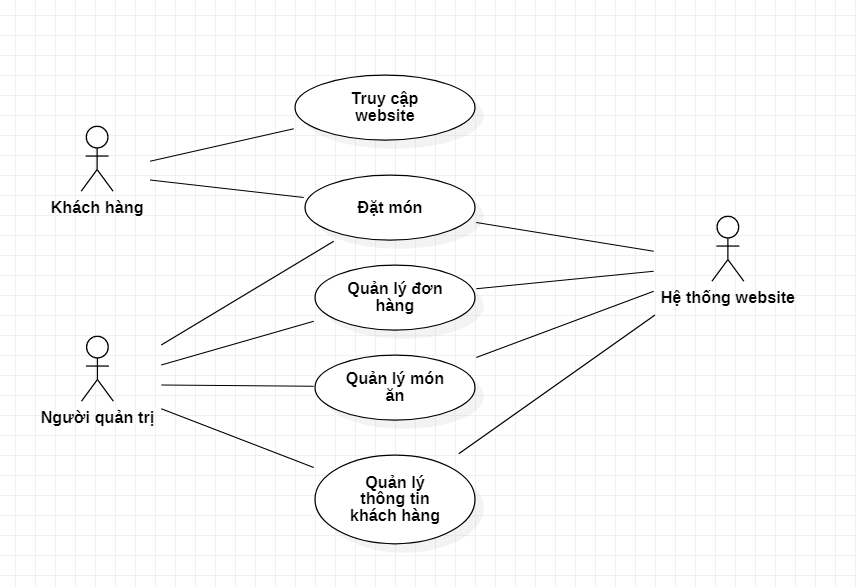
CHƯƠNG II: PHÂN TÍCH THIẾT KẾ HỆ THỐNG THÔNG TIN WEBSITE BÁN THỨC ĂN NHANH

1. **Mô tả hệ thống và chức năng**
   1. **Mô tả hệ thống**

* Kiwi’s fast food là một hệ thống cửa hàng bán thức ăn nhanh mới đưa vào hoạt động cuối năm 2021, đến với website Kiwi’s fast food khách hàng có thể dễ dàng đặt món tại nhà, tìm thấy nhiều loại pizza ngon, được chế biến công phu từ đôi bàn tay của các đầu bếp chuyên nghiệp. Ngoài Pizza ra cửa hàng còn có các loại thức ăn nhanh khác như: gà rán, nước ngọt, xúc xích xông khói, mì Ý, v.v… Cùng tiêu chí không ngừng hoàn thiện, nâng cao chất lượng món ăn, dịch vụ, Kiwi’s fast food cam kết sẽ là lựa chọn tốt cho quý khách hàng yêu thích thức ăn nhanh!
  1. **Chức năng**
* Chức năng chính của ứng dụng đối với người quản trị:
* Đăng nhập vào hệ thống quản trị.
* Quản lý loại, món ăn.
* Quản lý thông tin, danh sách khách hàng.
* Quản lý đơn đặt hàng.
* Chức năng chính của ứng dụng đối với người dùng:
* Tham quan website.
* Xem danh sách món ăn có trong thực đơn.
* Xem chi tiết món ăn.
* Lọc món ăn theo thực đơn.
* Thêm vào giỏ hàng.
* Quản lý giỏ hàng.
* Đăng ký tài khoản thành viên.
* Đăng nhập.
* Hiển thị hóa đơn, tổng tiền.
* Đặt hàng trực tuyến.

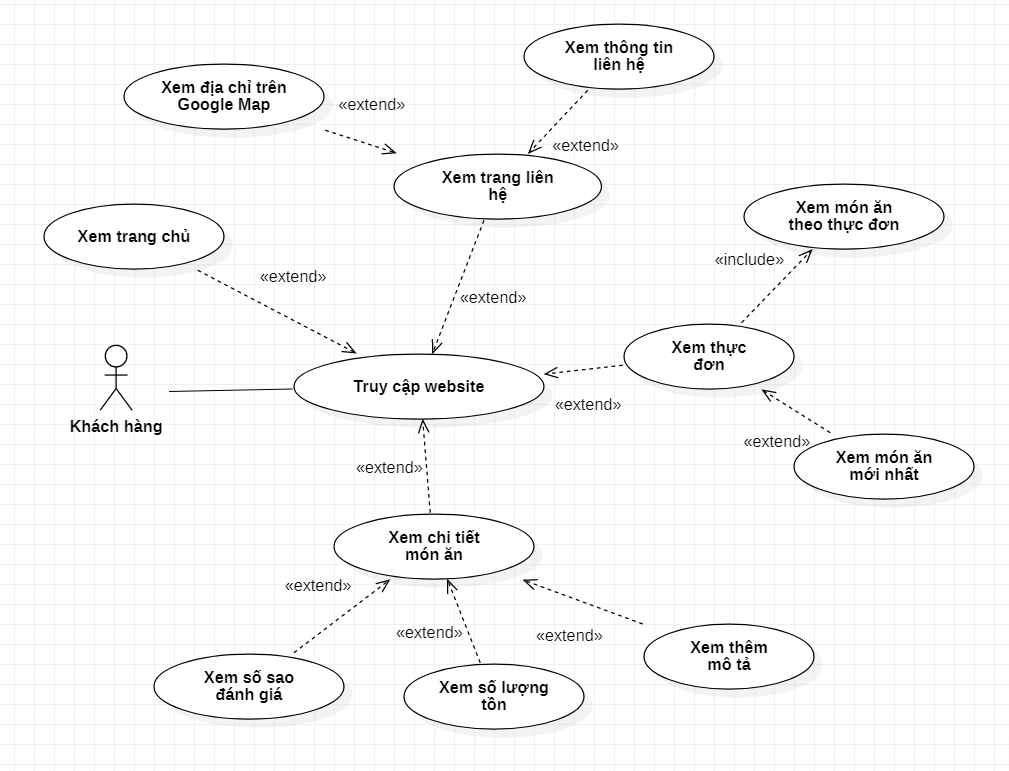
1. **Sơ đồ Usecase**

* Usecase Tổng quát



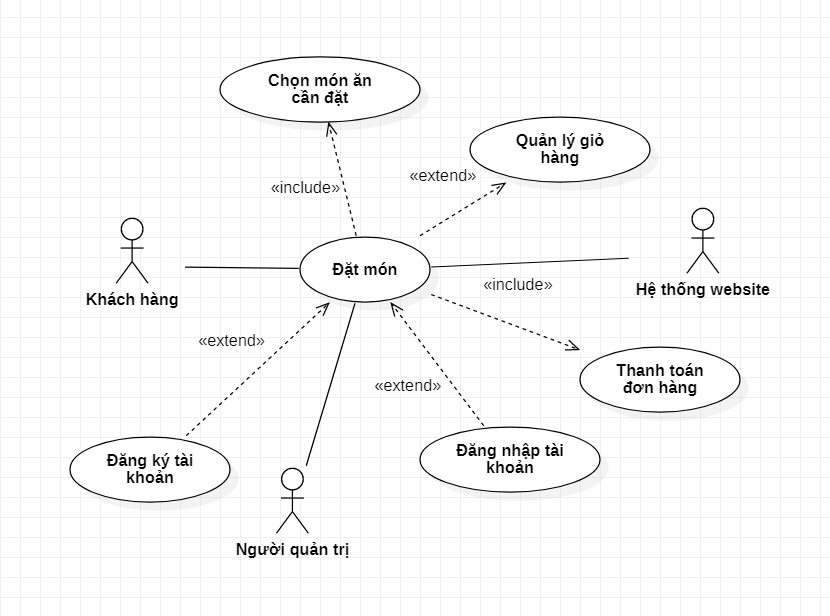
Hình Usecase Tổng quát

* Usecase Truy cập website



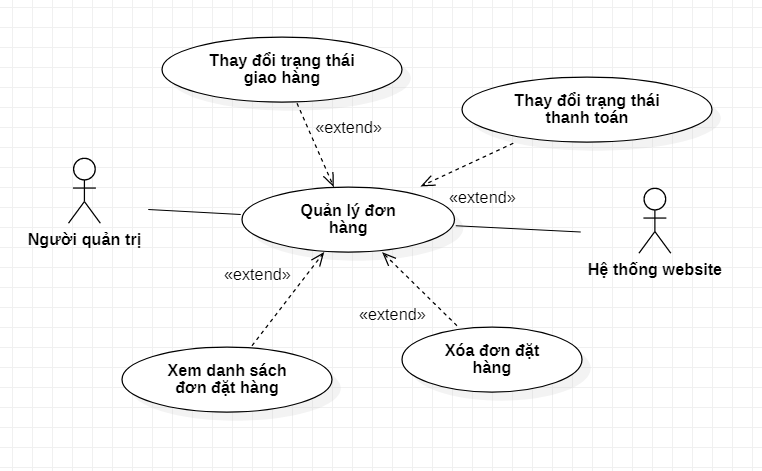
Hình Usecase Truy cập website

* Usecase Đặt món



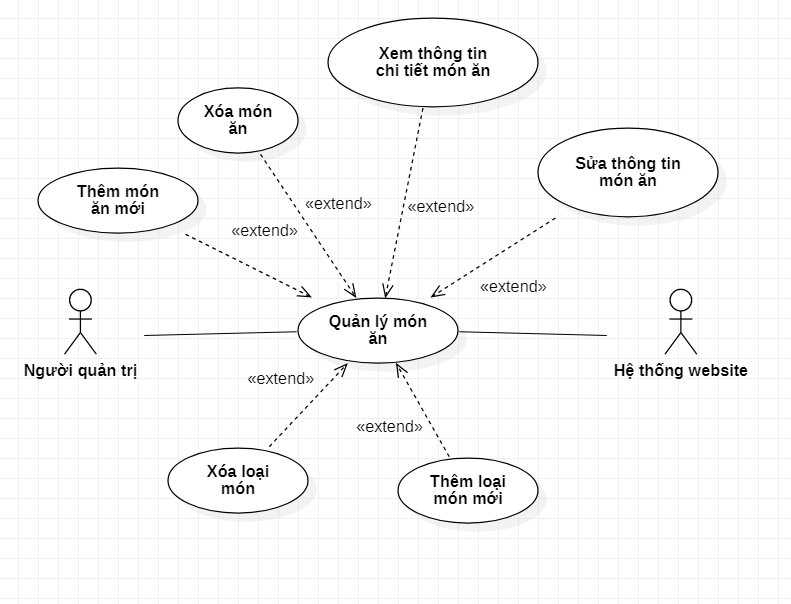
Hình Usecase Đặt món

* Usecase Quản lý đơn hàng



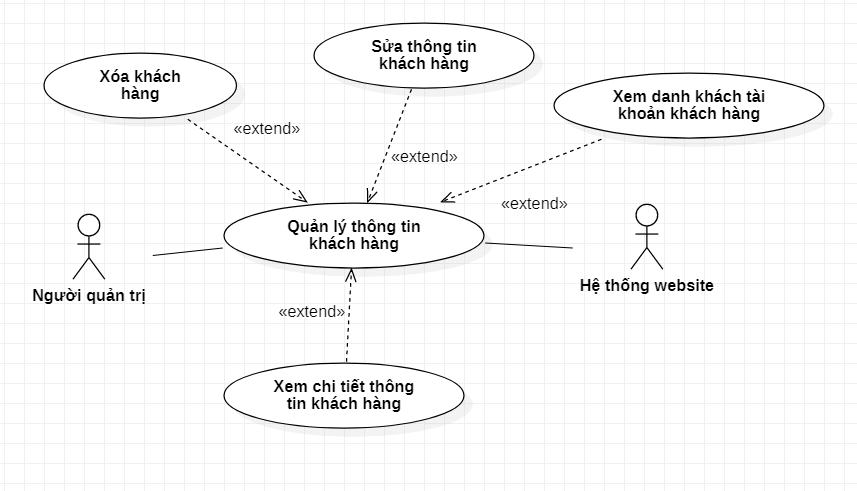
Hình Usecase Quản lý đơn hàng

* Usecase Quản lý món ăn



Hình Usecase Quản lý món ăn

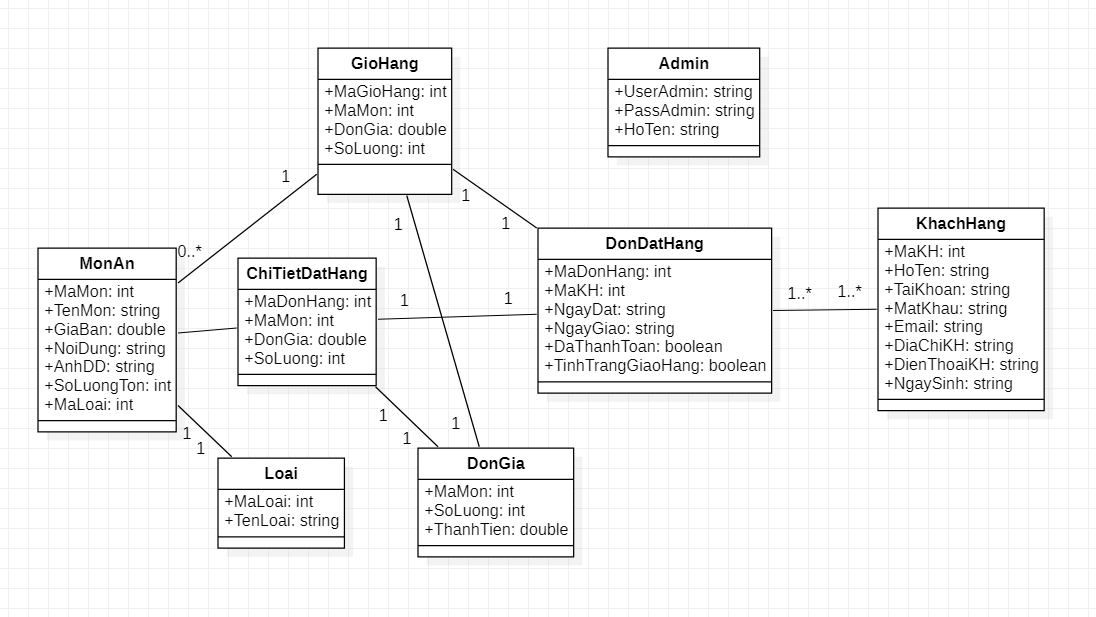
* Usecase Quản lý thông tin khách hàng



Hình Usecase Quản lý thông tin khách hàng

1. **Sơ đồ Class**

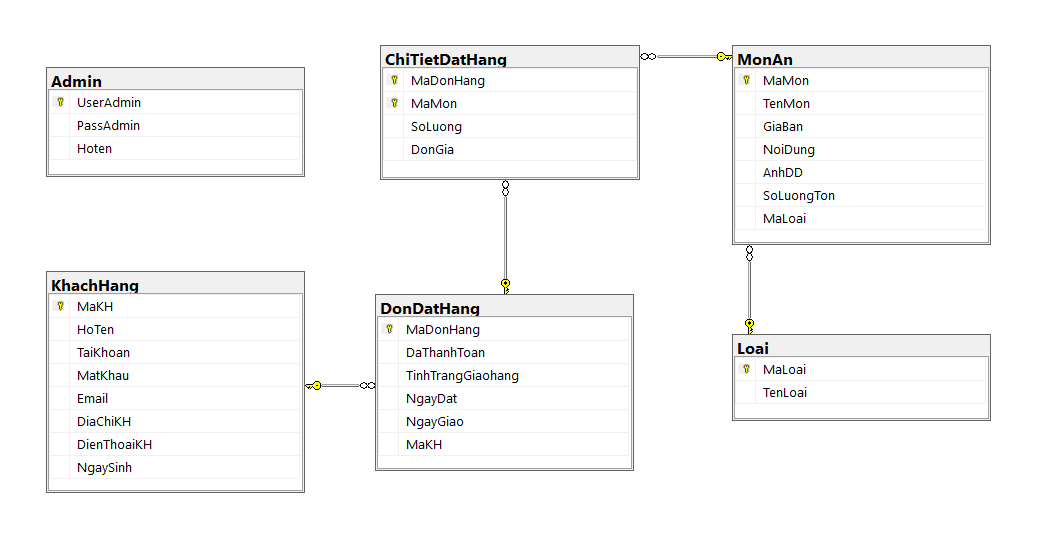
* Dựa vào các Usecase ở trên, ta xác định được các lớp: KhachHang, Loai, MonAn, DonGia, GioHang, DonDatHang, ChiTietDatHang, Admin.
* Xác định thuộc tính và phương thức các lớp
* KhachHang: MaKH int, HoTen string, TaiKhoan string, MatKhau string, Email string, DiaChiKH string, DienThoaiKH string, NgaySinh string.
* MonAn: MaMon int, TenMon string, GiaBan double, NoiDung string, AnhDD string, SoLuongTon int, MaLoai int.
* Loai: MaLoai int, TenLoai string.
* DonGia: MaMon int, SoLuong int, ThanhTien double.
* GioHang: MaGioHang int, MaMon int, DonGia double, SoLuong int.
* DonDatHang: MaDonHang int, MaKH int, NgayDat string, NgayGiao string, DaThanhToan boolean, TinhTrangGiaoHang boolean.
* ChiTietDatHang: MaDonHang int, MaMon int, DonGia double, SoLuong int.
* Admin: UserAdmin string, PassAdmin string, HoTen string.



Hình Sơ đồ Class website bán thức ăn nhanh

1. **Cơ sở dữ liệu**
   1. **Lược đồ CSDL quan hệ**

* KhachHang (**MaKH**, HoTen, TaiKhoan, MatKhau, Email, DiaChiKH, DienThoaiKH, NgaySinh)
* MonAn (**MaMon**, TenMon, GiaBan, NoiDung, AnhDD, SoLuongTon, MaLoai)
* Loai (**MaLoai**, TenLoai)
* DonDatHang (**MaDonHang**, DaThanhToan, TinhTrangGiaohang, NgayDat, NgayGiao, MaKH)
* ChiTietDatHang (**MaDonHang**, **MaMon**, SoLuong, DonGia)
* Admin (**UserAdmin**, PassAdmin, Hoten)



Hình Lược đồ CSDL quan hệ của website bán thức ăn nhanh

* 1. **Câu lệnh truy vấn SQL**

﻿