TRƯỜNG ĐẠI HỌC CẦN THƠ

KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

**BỘ MÔN TIN HỌC ỨNG DỤNG**



LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC

**XÂY DỰNG HỆ THỐNG WEBSITE**

**QUẢN LÝ KINH DOANH COFFESHOP VỚI SPRING FRAMEWORK**

*Sinh viên thực hiện*

**Huỳnh Đặng Thành Đạt**

**B1400942**

*Giáo viên hướng dẫn*

**ThS. Huỳnh Phụng Toàn**

**Cần Thơ – 2018**

TRƯỜNG ĐẠI HỌC CẦN THƠ

KHOA CÔNG NGHỆ THÔNG TIN & TRUYỀN THÔNG

**BỘ MÔN TIN HỌC ỨNG DỤNG**



LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC

**XÂY DỰNG HỆ THỐNG WEBSITE**

**QUẢN LÝ KINH DOANH COFFE SHOP VỚI SPRING FRAMEWORK**

*Sinh viên thực hiện*

**Huỳnh Đặng Thành Đạt**

**B1400942**

*Giáo viên hướng dẫn*

**ThS. Huỳnh Phụng Toàn**

Cán bộ phản biện

Luận văn được bảo vệ tại: Hội đồng chấm luận văn tốt nghiệp Bộ môn Tin học

Ứng dụng, Khoa CNTT&TT, Trường Đại học Cần Thơ vào ngày 25 tháng 05 năm

2018

Mã số đề tài:

Có thể tìm hiểu luận văn tại:

-Trung tâm Học liệu, Trường Đại học Cần Thơ

-Website: http://www.lrc.ctu.edu.vn/

**Cần Thơ – 2018**

# LỜI CẢM ƠN

Lời nói đầu tiên em xin cảm ơn các Thầy Cô của Khoa Công Nghệ Thông Tin và Truyền Thông trường Đại Học Cần Thơ đã hướng dẫn và dạy dỗ em trong suốt quá trình học tại trường. Trong suốt quá trình học Thầy Cô đã tạo mọi điều kiện để em có thể học tập và tiếp thu những kiến thức từ cơ bản nhất đến chuyên sâu, tìm hiểu những kiến thức mới nhất và khai thác triệt để.

Đặc biệt, em xin chân thành cảm ơn Thầy Huỳnh Phụng Toàn đã tận tâm hướng dẫn em qua từng buổi học trên lớp, những buổi trò chuyện, thảo luận về đề tài nghiên cứu và có những góp ý, hướng dẫn và tạo mọi điều kiện tốt nhất cũng như thời gian trong quá trình thực hiện Luận Văn. Một lần nữa, em xin chân thành cảm ơn Thầy.

Mặc dù đã cố gắng hoàn thành Luận Văn với tất cả nổ lực của bản thân, sự trao đổi với các bạn bè cùng thực hiện, và đặt biệt là sự hướng dẫn tận tình của Thầy Huỳnh Phụng Toàn, nhưng trong quá trình thực hiện chắc chắn sẽ không tránh khỏi những sai sót, em rất mong quý Thầy Cô và bạn bè đóng góp ý kiến để Luận Văn cũng như kiến thức của em được hoàn thiện hơn.

Trân trọng

Cần Thơ, ngày tháng năm 2018

Huỳnh Đặng Thành Đạt

# MỤC LỤC

[CHƯƠNG 1. LỜI CẢM ƠN 4](#_Toc530298974)

[CHƯƠNG 2. MỤC LỤC 5](#_Toc530298975)

[CHƯƠNG 3. DANH MỤC HÌNH 8](#_Toc530298976)

[CHƯƠNG 4. DANH MỤC BẢNG 11](#_Toc530298977)

[CHƯƠNG 6. TÓM TẮT 12](#_Toc530298978)

[CHƯƠNG 7. ABSTRACTION 13](#_Toc530298979)

[CHƯƠNG 1. TỔNG QUAN 14](#_Toc530298980)

[1.1. ĐẶT VẤN ĐỀ 14](#_Toc530298981)

[1.2. LỊCH SỬ GIẢI QUYẾT VẤN ĐỀ 14](#_Toc530298982)

[1.3. PHẠM VI CỦA ĐỀ TÀI 14](#_Toc530298983)

[1.4. HƯỚNG DẪN GIẢI QUYẾT VẤN ĐỀ 15](#_Toc530298984)

[1.5. PHƯƠNG PHÁP GIẢI QUYẾT 15](#_Toc530298985)

[1.6. MỤC TIÊU CẦN ĐẠT ĐƯỢC 15](#_Toc530298986)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 17](#_Toc530298987)

[1.1. SPRING FRAMEWORK 17](#_Toc530298988)

[2.1.1. Giới thiệu Spring Framework 17](#_Toc530298989)

[2.1.2. Lịch sử phát triển của Spring Framework 17](#_Toc530298990)

[2.1.3. Ưu điểm của Spring Framework 18](#_Toc530298991)

[2.1.4. Spring boot 19](#_Toc530298992)

[2.1.5. Cấu trúc thư mục và tập tin của Spring Framework 21](#_Toc530298993)

[2.1.6. Những tính năng nổi bật của Spring Framework 21](#_Toc530298994)

[1.2. Mô hình MVC 22](#_Toc530298995)

[2.2.1. Giới thiệu mô hình MVC 22](#_Toc530298996)

[2.2.2. Xuất xứ mô hình MVC 22](#_Toc530298997)

[2.2.3. Kiến trúc mô hình MVC 23](#_Toc530298998)

[2.2.4. Đặc điểm mô hình MVC 25](#_Toc530298999)

[2.2.5. Ưu Khuyết điểm của mô hình MVC 26](#_Toc530299000)

[1.3. JAVA FRAMEWORK 26](#_Toc530299001)

[1.3.1. Giới thiệu JAVA Framework 26](#_Toc530299002)

[1.3.2. Tính linh hoạt cơ bản của Java Framework 27](#_Toc530299003)

[1.3.3. Các thành phần cơ bản của một Java Framework 27](#_Toc530299004)

[1.4. jQuery LIBRARY 31](#_Toc530299005)

[1.4.1. Giới thiệu về jQuery 31](#_Toc530299006)

[1.4.2. jQuery làm được những gì? 31](#_Toc530299007)

[1.4.3. Tại sao jQuery làm việc tốt? 32](#_Toc530299008)

[1.5. Bộ thư viện Gradle 33](#_Toc530299009)

[1.5.1. Giới thiệu về Gradle 33](#_Toc530299010)

[1.5.2. Sự khác nhau giữa Gradle và Maven 34](#_Toc530299011)

[1.6. Bootstrap Framework 34](#_Toc530299015)

[1.6.1. Giới thiệu về Bootstrap Framework 34](#_Toc530299016)

[1.6.2. Vì sao nên sử dụng Bootstrap? 34](#_Toc530299017)

[1.7. HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU QUAN HỆ 35](#_Toc530299018)

[1.7.1. Giới thiệu MySQL 35](#_Toc530299019)

[1.7.2. Một số đặc điểm của MySQL 36](#_Toc530299020)

[1.7.3. Mô hình hoạt động của một ứng dụng Web động 36](#_Toc530299021)

[1.8. Mô tả WEBSITE 37](#_Toc530299022)

[1.8.1. Mô tả trang người sử dụng (Users) 37](#_Toc530299023)

[1.8.2. Mô tả trang người quản trị (Administrators) 37](#_Toc530299024)

[CHƯƠNG 3. NỘI DUNG VÀ KẾT QUẢ NGHIÊN CỨU 39](#_Toc530299025)

[3.1. PHÂN TÍCH VÀ THIẾT KẾ CÁC SƠ ĐỒ 39](#_Toc530299026)

[3.1.1. Sơ đồ phân rã chức năng (Bussiness Function Diagram – BFD) 39](#_Toc530299027)

[3.1.2. Sơ đồ luồng dữ liệu (Data Flow Diagram – DFD) 40](#_Toc530299028)

[3.1.3. Sơ đồ trường hợp sử dụng (Use case Diagram) 44](#_Toc530299029)

[3.1.4. Bảng mô tả từng trường hợp sử dụng (Use Case) 48](#_Toc530299030)

[3.1.5. Sơ đồ tuần tự (Sequence Diagram) 51](#_Toc530299031)

[3.2. CƠ SỞ DỮ LIỆU 55](#_Toc530299032)

[3.2.1. Mô hình thực thể mối quan hệ (Entity Relationship) 55](#_Toc530299033)

[3.2.2. Mô hình vật lý (Physical Diagram) 56](#_Toc530299034)

[3.2.3. Mô tả chi tiết các bảng dữ liệu 57](#_Toc530299035)

[3.3. KẾT QUẢ NGHIÊN CỨU 73](#_Toc530299036)

[3.3.1. Cấu trúc của hệ thống Website 73](#_Toc530299037)

[3.3.2. Giao diện của hệ thống Website 75](#_Toc530299038)

[4.1. KẾT LUẬN 78](#_Toc530299039)

[4.2. CÁC CHỨC NĂNG ĐÃ THỰC HIỆN ĐƯỢC 78](#_Toc530299040)

[4.3. HƯỚNG PHÁT TRIỂN 78](#_Toc530299041)

[CHƯƠNG 8. TÀI LIỆU THAM KHẢO 80](#_Toc530299042)

# DANH MỤC HÌNH

[Hình 2.1 Các module của Spring Framework 16](#_Toc530334721)

[Hình 2.2Cấu trúc thư mục và tập tin của Spring Framework 17](#_Toc530334722)

[Hình 2.3 Cấu trúc thư mục và tập tin của Spring Framework 18](#_Toc530334723)

[Hình 2.4 Mô hình MVC (Model – View – Controller) 21](#_Toc530334724)

[Hình 2.5 Sơ đồ tuần tự của mô hình MVC 22](#_Toc530334725)

[Hình 2.6 Thống kê độ phổ biến của các Java Framework năm 2015 23](#_Toc530334726)

[Hình 2.7 Spring MVC 25](#_Toc530334727)

[Hình 2.8 Struts Framework 26](#_Toc530334728)

[Hình 2.9 Hibernate Framework 27](#_Toc530334729)

[Hình 2.10 JQuery Library 28](#_Toc530334730)

[Hình 2.11 Bootstrap Framework 31](#_Toc530334731)

[Hình 2.12 Hệ quản trị cơ sở dữ liệu MySQL 32](#_Toc530334732)

[Hình 2.13 Mô hình hoạt động của một ứng dụng Web động 33](#_Toc530334733)

[Hình 3.1 Sở đồ use case mức 0 41](file:///C:\Program%20Files%20(x86)\sts-4.0.1.RELEASE\workspace\TheCoffeShop\DESIGN\LuanVan.docx#_Toc530334734)

[Hình 3.2 Sơ đồ Use Case mức 1 các chức năng quản lý của Admin 42](#_Toc530334735)

[Hình 3.3 Sơ đồ Use Case mức 1 các chức năng quản lý của nhân viên pha chế 44](#_Toc530334736)

[Hình 3.4 Sơ đồ tuần tự chức năng đăng nhập 48](file:///C:\Program%20Files%20(x86)\sts-4.0.1.RELEASE\workspace\TheCoffeShop\DESIGN\LuanVan.docx#_Toc530334737)

[Hình 3.5 Sơ đồ tuần tự chức năng thêm mới sản phẩm 49](#_Toc530334738)

[Hình 3.6Sơ đồ tuần tự chức năng cập nhật trạng thái hóa đơn 50](file:///C:\Program%20Files%20(x86)\sts-4.0.1.RELEASE\workspace\TheCoffeShop\DESIGN\LuanVan.docx#_Toc530334739)

[Hình 3.7 Giao diện xem sản phẩm dành cho khách hàng 74](file:///C:\Program%20Files%20(x86)\sts-4.0.1.RELEASE\workspace\TheCoffeShop\DESIGN\LuanVan.docx#_Toc530334740)

[Hình 3.8 Giao diện đặt bàn 75](file:///C:\Program%20Files%20(x86)\sts-4.0.1.RELEASE\workspace\TheCoffeShop\DESIGN\LuanVan.docx#_Toc530334741)

# DANH MỤC BẢNG

[Bảng 3.1 Mô tả bảng dữ liệu Unit (Đơn vị) 55](#_Toc530335640)

[Bảng 3.2 Mô tả bảng dữ liệu Material (Danh mục) 56](#_Toc530335641)

[Bảng 3.3 Mô tả bảng dữ liệu ExportBill (xuất hóa đơn) 57](#_Toc530335642)

[Bảng 3.4 Mô tả bảng dữ liệu MaterialDetail (Chi tiết nguyên liệu) 58](#_Toc530335643)

[Bảng 3.5 Mô tả bảng dữ liệu importBill (hoa đơn nhập) 59](#_Toc530335644)

[Bảng 3.6 Mô tả bảng dữ liệu Supplier (Nhà cung cấp) 59](#_Toc530335645)

[Bảng 3.7 Mô tả bảng dữ liệu members (Tài khoản người dùng) 60](#_Toc530335646)

[Bảng 3.8 Mô tả bảng dữ liệu customer (Khách hàng) 61](#_Toc530335647)

[Bảng 3.9 Mô tả bảng dữ liệu voucher (Mã giảm giá) 61](#_Toc530335648)

[Bảng 3.10 Mô tả bảng dữ liệu bill (Hóa đơn) 62](#_Toc530335649)

[Bảng 3.11 Mô tả bảng dữ liệu billDetail (chi tiết hóa đơn) 63](#_Toc530335650)

[Bảng 3.12 Mô tả bảng dữ liệu product (Sản phẩm) 64](#_Toc530335651)

[Bảng 3.13 Mô tả bảng dữ liệu price (Giá) 64](#_Toc530335652)

[Bảng 3.14 Mô tả bảng dữ liệu image (Hình ảnh) 65](#_Toc530335653)

[Bảng 3.15 Mô tả bảng dữ liệu salary (bảng lương) 65](#_Toc530335654)

[Bảng 3.16 Mô tả bảng dữ liệu ImageOfEmployee (hình ảnh của nhân viên) 66](#_Toc530335655)

[Bảng 3.17 Mô tả bảng dữ liệu ImageOfProduct (hình ảnh của sản phẩm) 66](#_Toc530335656)

[Bảng 3.18 Mô tả bảng dữ liệu CategoryProduct (loại sản phẩm) 66](#_Toc530335657)

[Bảng 3.19 Mô tả bảng dữ liệu Employee (nhân viên) 67](#_Toc530335658)

[Bảng 3.20 Mô tả bảng dữ liệu DinnerTable (bàn ăn) 67](#_Toc530335659)

[Bảng 3.21 Mô tả bảng dữ liệu BookTable (Đặt bàn) 68](#_Toc530335660)

[Bảng 3.22 Mô tả bảng dữ liệu TableStatus (Trạng thái bàn) 68](#_Toc530335661)

[Bảng 3.23 Mô tả bảng dữ liệu Position (Chức vụ) 68](#_Toc530335662)

[Bảng 3.24 Mô tả bảng dữ liệu Position (Chức vụ) 69](#_Toc530335663)

[Bảng 3.25 Mô tả bảng dữ liệu Register (Đăng kí ca làm) 69](#_Toc530335664)

[Bảng 3.26 Mô tả bảng dữ liệu Schedule (Ca làm) 69](#_Toc530335665)

[Bảng 3.27 Mô tả bảng dữ liệu Schedule (Ca làm) 70](#_Toc530335666)

[Bảng 3.28 Bảng thuộc tính gắn cờ 70](#_Toc530335667)

# TÓM TẮT

Những năm gần đây, trong bối cảnh đất nước đang trong giai đoạn công nghiệp hóa hiện đại hóa. Hầu hết các ngành nghề, các lĩnh vực đều đang thay đổi các thức hoạt động để bắt kịp xu hướng tiến bộ đổi mới hơn trước đây. Tất cả đã và đang áp dụng các biện pháp kỹ thuật công nghệ cao nhằm nâng cao chất lượng sản phẩm, năng xuất và giảm thiểu chi phí đầu tự. Ngoài những sự hỗ trợ đắt lực của thiết bị máy móc hiện đại, giúp tiết kiệm nhân công, thời gian sản xuất và thu hoạch, bên cạnh đó các ngành nghề cần phải áp dụng các biện pháp công nghệ thông tin vào việc quản lý các quy trình hoạt động. Đặc biệt là các ngành kinh doanh, buôn bán.

Trước đây hầu hết các ngành kinh đoanh đều sử dụng các phương thức thủ công trong việc quản lý nhân công, quảng cáo sản phẩm, cũng như quản lý các giấy tờ trong hoạt động kinh doanh. Điều này làm mất rất nhiều thời gian và nhân lực, kể cả chi phí để các quy trình hoạt động. Do vậy,các ngành nghề kinh doanh cần phải áp dụng các giải pháp công nghệ thông tin vào các hoạt động kinh của mình.

Mục tiêu của đề tài lại nhằm đáp ứng nhu cầu trên. Giúp cho việc quản lý được trơ tru và giải quyết nhanh chóng các vấn đề trong kinh doanh.

Đề tài đã đáp ứng được các yêu cầu đã nêu trên. Tuy nhiên, khi được đưa vào thực tiễn sẽ không tránh khỏi những thiếu sót cũng như phát sinh thêm những yêu cầu về chức năng khác.

# ABSTRACTION

In recent years, in the context of the country in the period of industrialization and modernization. Most of the trades, the fields are all changing the operation to catch the trend of progressive innovation than before. All are technical measures applied high technology to improve product quality, productivity and minimize cost. In addition to the support of expensive modern equipment, help save labor time producing and harvesting, besides the profession need to apply the measures of information technology on the management of the operational processes. Especially the business sector, traders.

Previously most of the industry đoanh are using the manual method of managing employees, advertising products, as well as manage the papers in business activities. This takes a lot of time and Manpower, including costs to the active process. Therefore, the industry needs to apply information technology solutions into their business activities.

The objective of the subject back to meet the above needs. Help for the management to be inert and quick resolution tru issues in business.The subject has met the above requirements. However, when put into practice will be inevitable shortcomings as well as arise other functional requirements.

# TỔNG QUAN

## ĐẶT VẤN ĐỀ

Từ xưa đến nay việc quản lý kinh doanh của các ngành nghề đều dựa theo phương pháp thủ công. Lưu trữ bằng giấy tờ và phải cần rất nhiều nhân lực cho khâu quản lý kinh doanh. Điều này dẫn đến rất nhiều sai sót, sử dụng nhiều nhân lực và cần phải bỏ ra một chi phí tương đối lớn để quản lý kinh doanh, không những thế chi phí cho việc quản cáo dịch vụ của của hàng cũng rất cao. Đó là các hạn chế của cách thức quản lý kinh doanh cũ của các ngành nghề nói chung và ngành kinh doanh ăn uống nói riêng. Chưa kể đến việc khách hàng phải mất nhiều thời gian để đến tận cửa hàng ăn uống, chọn món và phải đợi chế biến.

Ngày nay, với sự phát triển vượt bậc của Công nghệ thông tin, Internet đã trở nên quen thuộc với tất cả mọi người. Cùng với sự tiếp cận nhanh chóng của những thiết bị hiện đại, những ngành nghề cũng đã có thể dễ dàng sử dụng Internet một cách dễ dàng. Chỉ cần có một chiếc Laptop hay một chiếc Smartphone thì mọi người đã có thể truy cập Internet.

Vì vậy, vấn đề được đặt ra là làm sao để quản lý tình hình kinh doanh của cửa hàng Coffe Shop khi chỉ cần có một thiết bị có thể kết nối được Internet. Không những vậy làm sao để khách hàng có thể tiết kiệm thời gian mà vẫn có thể sử dụng dịch vụ của Coffe Shop một cách trọn vẹn.

## LỊCH SỬ GIẢI QUYẾT VẤN ĐỀ

Hiện nay, có rất nhiều cửa hàng kinh doanh dịch vụ ăn uống sử dụng các cách thức cũ kỹ như lưu trữ hóa đơn dưới hình thức giấy tờ, quản lý phát lương nhân viên bằng biên lai- sổ sách, quảng cáo dịch vụ cửa hàng bằng tờ rơi ngoài ra còn có cửa hàng cho phép khách hàng đặt bàn, đặt thức uống bằng hình thức gọi điện và các nhân viên phải dành nhiều thời gian tư vấn cho khách hàng. Các cách thức làm mất nhiều công sức, tiền bạc , không bảo vệ môi trường, phí phạm thời gian của khách hàng.

Nhằm áp dụng công nghệ thông tin vào thực tiễn, giúp các cửa hàng và khách hàng hạn chế việc mất thời gian và tiền bạc, giúp các cửa hàng dễ dàng quản lý tình hình kinh doanh, quản lý nhân sự, giúp khách hàng có thể đặt bàn, gọi món online .

## PHẠM VI CỦA ĐỀ TÀI

Đề tài này là ứng dụng Internet, xây dựng hệ thống Website để đáp ứng nhu cầu giái quyết các vấn đề về quản lý kinh doanh coffe shop và hỗ trợ việc đặt bàn đặt thức uống trực tuyến cho khách hàng..

Đề tài sẽ giúp coffe shop có những hướng giải quyết cụ thể, dựa vào việc quản lý trực tuyến có thể đưa ra các giải pháp khắc phục cấn vấn đề bên trên. Bên cạnh đó Website còn có các danh mục thống kê thu chi, thống kê doanh thu theo tuần, tháng, năm.

Đề tài cho khách hàng đặt câu hỏi trực tiếp trên Webstie và sẽ được nhân viên tư vấn một cách nhanh nhất có thể, cho phép khách hàng bình luận đóng góp ý kiến về các dịch vụ, món ăn của coffe shop. Góp phần giúp coffe shop nâng cao dịch vụ.

## HƯỚNG DẪN GIẢI QUYẾT VẤN ĐỀ

Người quản trị cần tổng hợp các giải đáp của các chuyên gia về nông nghiệp, các cách phòng ngừa, điều trị bệnh hại cây trồng, thông tin về các loại cây trồng theo từng vùng miền, đất đai, thời tiết,… ngoài việc đưa ra các câu trả lời cho người nông dân, cần phải đưa ra lời khuyên, cũng như cách phòng ngừa sau này, giúp người nông dân đạt hiệu quả năng suất cao nhất.

Người nông dân đặt câu hỏi về vấn đề mà mình gặp phải (kèm hình ảnh chụp trực tiếp từ cây bị bệnh). Mỗi câu hỏi cần có những thông tin cần thiết: tên bệnh (nếu biết), thời gian phát hiện gần nhất, màu sắc, độ lớn (diện tích của bệnh trên lá, trên thân cây hoặc trên quả).

## PHƯƠNG PHÁP GIẢI QUYẾT

Xây dựng hệ thống Website sử dụng ngôn ngữ lập trình kịch bản PHP (PHP: Hypertext Preprocessor) dựa trên nền tảng Laravel Framework 5.1, là một framework dùng để xây dựng web application, được thiết kế dựa trên mô hình MVC (Model, Controller, View), toàn bộ source code được đặt trên github. Theo kết qủa khảo sát của các Developer vào tháng 12 năm 2013, thì Laravel Framework đứng top 1 một trong những framework phổ biến nhất, tiếp sau là Phalcon, Symfony2, CodeIgniter và các framework khác. Tháng 8 năm 2014, Laravel Framework được xem như là một dự án PHP phổ biến nhất trên Github.

Tìm kiếm thông tin về bệnh hại cây trồng để xây dựng và thiết kế nội dung Website. Tìm hiểu thực trạng cây trồng từng vùng miền, các thông tin thời sự về nông nghiệp để xây dựng các mục khuyến nông, kiến thức nhà nông.

Sử dụng HTML5, Ajax & JavaScript Component Library và Bootstrap Freamwork để xây dựng giao diện cho các layout và form.

## MỤC TIÊU CẦN ĐẠT ĐƯỢC

Xây dựng một hệ thống Website hỗ trợ coffe shop quản lý tình hình kinh doanh, quản lý nhân sự, quảng bá dịch vụ và quản lý khách hàng. Hỗ trợ khách hàng đặt bàn, xem sản phẩm và đặt hàng sản phẩm trực tuyến. Website có giao diện thân thiện, dễ sử dụng trên nền tảng Spring Framework.

# CƠ SỞ LÝ THUYẾT

## SPRING FRAMEWORK

### Giới thiệu Spring Framework

Có thể nói rằng Spring Framework ra đời là nhờ sự thành công của **EJB**. EJB (Enterprise JavaBeans) là một kiến trúc phát triển dành cho các ứng dụng doanh nghiệp đòi hỏi tính mạnh mẽ và khả năng mở rộng cao. EJB là một thành phần quan trọng của nền tảng Java EE. EJB được tạo ra bởi hai gã khổng lồ công nghệ là IBM và Sun vào những năm cuối thập niên 90. Dưới sự phát triển của hai tập đoàn này, EJB đã trở nên nặng nề hơn. Tiêu chuẩn EJB cũng nhanh chóng được thông qua. Sau tất cả, EJB được ca ngợi là giải pháp tốt nhất để xây dựng các ứng dụng phân tán dành cho doanh nghiệp.

Nhưng rồi những chỉ trích về chi tiết kỹ thuật của EJB đã tăng nhanh còn hơn sự nổi tiếng của nó. EJB là một framework lớn với độ phức tạp cao. Các nhà phát triền khi làm việc với EJB đã nhận ra rằng nó không hoạt động như những gì nó đã đề ra. EJB chỉ tuyệt vời trên bàn giấy, nhưng ra thực tế đó lại là một chuyện khác.

EJB vẫn được tiếp tục phát triển, dễ sử dụng hơn nhiều so với buổi ban đầu. Nhưng những chỉ trích vẫn còn đó, nỗi thất vọng trong cộng đồng Java ngày càng tăng lên, đòi hỏi cần có một giải pháp thay thế. Và đó là lý do Spring ra đời.

Vào năm 2002, Spring Framework phát hành phiên bản đầu tiên bởi Rod Johnson. Việc xây dựng các ứng dụng doanh nghiệp trở nên đơn giản và dễ dàng hơn. Rod Johnson đã cho xuất bản một cuốn sách được phổ biến rộng rãi có nhan đề "J2EE Development without EJB". Điều này đã cho Spring Framework rất nhiều động lực để thay thế EJB.

Theo đà phát triển đó, Spring đã trở thành framework mã nguồn mở phổ biến nhất để xây dựng các ứng dụng doanh nghiệp. Cách tiếp cận thực tế ban đầu của Rod Johnson tiếp tục được phát triển và hướng tới một bộ công cụ hoàn chỉnh dành cho xây dựng các ứng dụng doanh nghiệp. Theo một số nguồn, trên 50% các ứng dụng web Java hiện nay đang sử dụng Spring.

### Lịch sử phát triển của Spring Framework

Phiên bản đầu tiên của Spring được viết bởi Rod Joinson, cùng lúc đó ông cũng cho ra cuốn sách *Expert One-on-One J2EE Design and Development* vào tháng 10 năm 2002.

Ban đầu, bộ khung được phát hành dưới giấy phép Apache 2.0 vào tháng 06 2003.

Cột mốc đầu tiên, bản 1.0, được phát hành tháng 03 năm 2004, và lần lượt các mốc phát hành kế tiếp vào tháng 09 năm 2004 và tháng 03 năm 2005.

Phiên bản Spring 1.2.6 nhận Jolt Productivity award JAX Innovation Award vào năm 2006.

Spring 2.0 được phát hành vào tháng 10 năm 2006.

Spring 2.5 phát hành vào tháng 11 năm2007.

Spring 3.0 phát hành vào tháng 12 năm 2009.

Spring 3.1 phát hành tháng 12 năm 2011.

Spring 3.2.5 phát hàng tháng 10 năm 2013.

Spring Framework 4.0 được phát hành vào tháng 12 năm 2013. Cải tiến đáng chú ý của phiên bản này gồm có hỗ trợ Java SE 8, Groovy 2, một số phần của Java EE7, và Websocket

Spring Framework 4.2.0 được phát hành ngày 31 tháng 07 năm 2015 và ngay sau đó được nâng cấp lên phiên bản 4.2.1, được phát hành ngày 01 tháng 09 cùng năm.[[6]](https://vi.wikipedia.org/wiki/Spring_Framework#cite_note-6) Phiên bản này được mô tả là *"tương thích với Java 6, 7 và 8, chú trọng vào những cải tiến cốt lõi và tính năng web hiện đại"*.[[7]](https://vi.wikipedia.org/wiki/Spring_Framework#cite_note-7)

Spring Framework 4.3 được phát hành ngày 10 tháng 06 năm 2016. Phiên bản 4.3.0.RC1 [[8]](https://vi.wikipedia.org/wiki/Spring_Framework#cite_note-8) có hiệu lực. Nó *"sẽ là thế hệ cuối cùng cho những yêu cầu của Spring 4 (Java 6+, Servlet 2.5+), chuẩn bị cho phiên bản mở rộng 4.3.x với thời gian hỗ trợ đến năm 2019"*.

Spring 5 được thông báo rằng sẽ dựng trên [Reactive Streams](https://en.wikipedia.org/wiki/Reactive_Streams) tương thích với Reactor Core.

### Ưu điểm của Spring Framework

Đơn giản hóa công việc phát triển thông qua việc sử dụng các POJO (Plain Old Java Object).

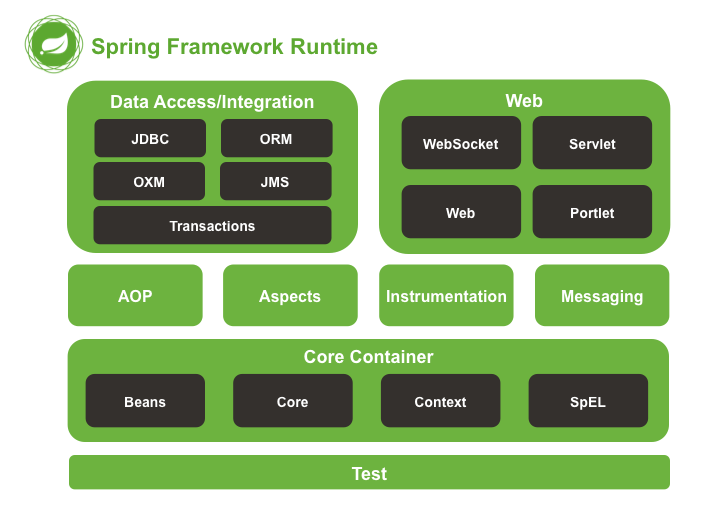
Nới lỏng ràng buộc giữa các thành phần thông qua việc sử dụng Dependency Injection. Và Spring Container giúp các thành phần này gắn kết lại thông qua cấu hình.

Gọn nhẹ trong việc phát triển ứng dụng.

Spring MVC FrameWork được sử dụng để phát triển các ứng dụng web nhanh chóng dễ dàng và hỗ trợ nhiều tính năng như webservice, json.

Hỗ trợ quản lý transaction, JDBC operations, File uploading, Exception Handling,… rất dễ dàng bằng cách cấu hình được rút gọn, thay vào đó là sử dụng annotation hoặc spring bean configuration file.

Spring Framework chi thành nhiều module riêng biệt.Do đó việc sử dụng rất tự do. Bạn chỉ cần sử dụng module này cho ứng dụng của bạn những module khác bạn không cần quan tâm.



Hình 2.1 *Các module của Spring Framework*

Spring framework hỗ trợ hầu hết các tính năng của Java EE, thậm chí còn nhiều hơn nữa.

### Spring boot

Spring Boot là một dự án nổi bật trong hệ sinh thái Spring Framework. Nếu như trước đây, công đoạn khởi tạo một dự án Spring khá vất vả từ việc khai báo các dependency trong file pom.xml cho đến cấu hình bằng XML hoặc annotation phức tạp, thì giờ đây với Spring Boot, chúng ta có thể tạo các ứng dụng Spring một cách nhanh chóng và cấu hình cũng đơn giản hơn.

Các ưu điểm:

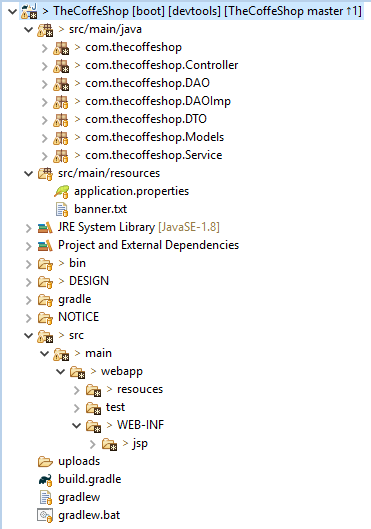
* Đóng gói ứng dụng Spring dưới dạng một file JAR (stand-alone). Chúng ta có thể dễ dàng start ứng dụng
* Tối ưu công đoạn cấu hình cho ứng dụng Spring, không sinh code cấu hình và không yêu cầu phải cấu hình bằng XML.
* Cung cấp một loạt các tính năng phi chức năng phổ biến cho các dự án lớn như nhúng trực tiếp web server như Tomcat, Jetty, ... vào ứng dụng, bảo mật, health check...

Phiên bảng đang sử dụng v2.07

Sự khác nhau giữa Spring Boot với Spring MVC:

* Spring boot nhúng web server vào bản thân ứng dụng thì khi riển khai tất cả những gì ta cần chỉ là một đóng gói của chính bản thân ứng dụng và các cấu hình sẽ không bị tản mác ở nhiều nơi mà chỉ nằm ở chính bản thân của ứng dụng mà thôi. Lúc đó sẽ dễ dàng hơn cho ta trong việc quản lý. Điều này rất có ý nghĩa khi thiết kế các hệ thống và xây dựng úng dụng theo mô hình Microservice.
* Bản thân ứng dụng sẽ có tiến trình (process) của riêng nó .Và ứng dụng được viết bằng Spring Boot từ bên ngoài nhìn người ta sẽ nhìn thấy ứng dụng thay vì một web server. Từ đó dễ dàng hơn trong quá trình giám sát và vận hành hệ thống.

Hình 2.2Cấu trúc thư mục và tập tin của Spring Framework

****

Hình 2.3 *Cấu trúc thư mục và tập tin của Spring Framework*

Thư mục *src/main/java:* Chứa code ứng dụng của bạn. Bao gồm: các controller, các hàm xử lý dữ liệu, model để mapping database.

Thư mục *src/main/webapp:* Chứa một số file của bootstrap framework và các jsp.

Thư mục src/main/resources: chức các đoạn code cấu hình Spring boot và cấu hình mapping Hibernate Framework.

### Những tính năng nổi bật của Spring Framework

*Module:* Spring hỗ trợ nhiều module cho các mục đích khác nhau, không cần quan tâm đến module không sử dụng.

*ORM (object relation mapping):* ánh xạ các đối tượng và quan hệ cơ sở dữ liệu, cung cấp các phương thức nội bộ để thực thi đồng thời cũng bổ sung các tính năng hạn chế về mối quan hệ giữa các đối tượng cơ sở dữ liệu. Eloquent ORM trình bày các bảng trong cơ sở dữ liệu dưới dạng các lớp, cung cấp thêm lựa chọn truy cập cơ sở dữ liệu trực tiếp mới mẻ hơn, chuyên nghiệp hơn.

*Application logic:* Là một phần của phát triển ứng dụng, được sử dụng bởi bộ điều khiển controllers.

*Routes:* Định nghĩa mối quan hệ giữa các đường dẫn (url), các liên kết (link) bằng Annotation việc này rút ngắn được code.

*Restful Controller:* cung cấp các tùy chọn để tách các logic phía sau các request HTTP POST, GET.

*View:* chưa các JSP, hiển thị dữ liệu được chỉ định bởi controller.

*JUnit Testing:* JUnit testting chứa rất nhiều các hệ thống unit testing, giúp phát hiện và ngăn chặn lỗi trong khuôn khổ nhất định. Unit Testing có thể được chạy thông qua tiện ích command-line.

## Mô hình MVC

### Giới thiệu mô hình MVC

MVC là viết tắt của Model – View – Controller. Là một trong những design pattern. Được vận hành để tách mã lệnh thành 3 phần riêng biệt. Ở mỗi phần MVC sẽ có những chức năng đặc thù. Để xử lý các tác vụ mà request gởi tới. MVC làm cho mã lệnh trở nên trong sáng, dễ phát triển và dễ nâng cấp theo thời gian.

Để làm việc tốt đối với MVC, chúng ta cần nắm thật vững kiến thức OOP. Bản chất của các framework khác cũng được hình thành trên lý thuyết MVC. Do vậy nếu chúng ta nắm tốt MVC. Thì ở những framework khác chắc chắn sẽ không cảm thấy khó hiểu.

### Xuất xứ mô hình MVC

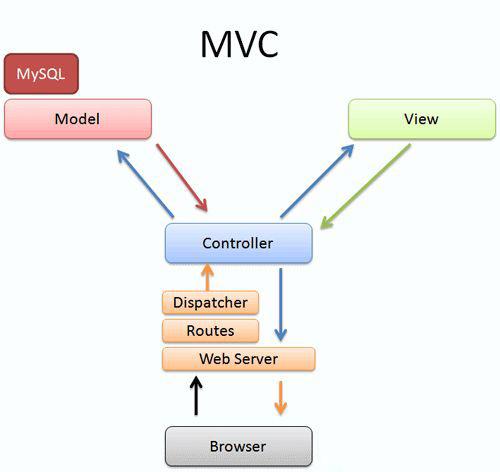
Tất cả bắt đầu vào những năm 70 của thế kỷ 20, tại phòng thí nghiệm Xerox PARC ở Palo Alto. Sự ra đời của giao diện đồ họa (Graphical User Interface) và lập trình hướng đối tượng ( Object Oriented Programming) cho phép lập trình viên làm việc với những thành phần đồ họa như những đối tượng đồ họa có thuộc tính và phương thức riêng của nó. Không dừng lại ở đó, những nhà nghiên cứu ở Xerox PARC còn đi xa hơn khi cho ra đời cái gọi là kiến trúc MVC ( viết tắt của Model – View – Controller).

MVC được phát minh tại Xerox Parc vào những năm 70, bởi TrygveReenskaug. MVC lần đầu tiên xuất hiện công khai là trong Smalltalk-80. Sau đó trong một thời gian dài hầu như không có thông tin nào về MVC, ngay cả trong tài liệu 80 Smalltalk. Các giấy tờ quan trọng đầu tiên đƣợc công bố trên MVC là “A Cookbook for Using the Model-View-Controller User Interface Paradigm in Smalltalk – 80”, bởi Glenn Krasner và Stephen Pope, xuất bản trong tháng 8 / tháng 9 năm 1988.

### Kiến trúc mô hình MVC

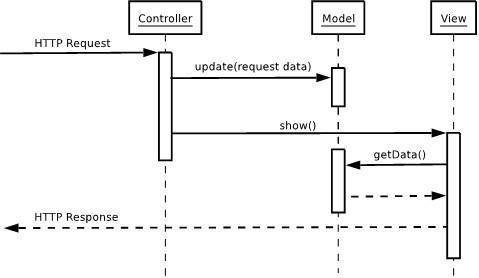
Trong kiến trúc MVC, một đối tượng đồ họa người dùng(GUI Compone nt) bao gồm 3 thành phần cơ bản: Model, View, và Controller. Model có trách nhiệm đối với toàn bộ dữ liệu cũng như trạng thái của đối tượng đồ họa. View chính là thể hiện trực quan của Model, hay nói cách khác chính là giao diện của đối tượng đồ họa. Và Controller điều khiển việc tương tác giữa đối tượng đồ họa với người sử dụng cũng như những đối tượng khác.

* *Model:* Là thành phần chịu trách nhiệm xử lý các thao tác trên database. Và gởi trả kết quả thông qua view.
* *View:* Là phần hiển thị thông tin trên website, sau khi đi qua controller và nhận kết quả từ phía model thì view là bước cuối cùng để chuyển thông tin tới người dùng.
* *Controller:* Là phần điều hướng các request tới những tác vụ tương ứng. Controller là một phần không thể thiếu ở bất cứ framework nào. Vì nó có trách nhiệm gởi và nhận request từ hệ thống tới người sử dụng.



Hình 2.4 *Mô hình MVC (Model – View – Controller)*

Khi người sử dụng hoặc những đối tượng khác cần thay đổi trạng thái của đối tượng đồ họa, nó sẽ tương tác thông qua Controller của đối tượng đồ họa. Controller sẽ thực hiện việc thay đổi trên Model. Khi có bất kỳ sự thay đổi nào xảy ra ở Model, nó sẽ phát thông điệp ( broadcast message) thông báo cho View và Controller biết. Nhận được thông điệp từ Model, View sẽ cập nhật lại thể hiện của mình, đảm bảo rằng nó luôn là thể hiện trực quan chính xác của Model. Còn Controller, khi nhận được thông điệp từ Model, sẽ có những tương tác cần thiết phản hồi lại người sử dụng hoặc các đối tượng khác.



Hình 2.5 *Sơ đồ tuần tự của mô hình MVC*

### Đặc điểm mô hình MVC

Lợi ích quan trọng nhất của mô hình MVC là nó giúp cho ứng dụng dễ bảo trì, module hóa các chức năng, và được xây dựng nhanh chóng. MVC tách các tác vụ của ứng dụng thành các phần riêng lẻ model, view, controller giúp cho việc xây dựng ứng dụng nhẹ nhàng hơn. Dễ dàng thêm các tính năng mới, và các tính năng cũ có thể dễ dàng thay đổi. MVC cho phép các nhà phát triển và các nhà thiết kế có thể làm việc đồng thời với nhau. MVC cho phép thay đổi trong 1 phần của ứng dụng mà không ảnh hưởng đến các phần khác

Sở dĩ như vậy vì kiến trúc MVC đã tách biệt (decoupling) sự phụ thuộc giữa các thành phần trong một đối tượng đồ họa, làm tăng tính linh động (flexibility) và tính tái sử dụng (reusebility) của đối tượng đồ họa đó. Một đối tượng đồ họa bấy giờ có thể dễ dàng thay đổi giao diện bằng cách thay đổi thành phần View của nó trong khi cách thức lưu trữ (Model) cũng như xử lý (Controller) không hề thay đổi. Tương tự, ta có thể thay đổi cách thức lưu trữ (Model) hoặc xử lý (Controller) của đối tượng đồ họa mà những thành phần còn lại vẫn giữ nguyên.

Chính vì vậy mà kiến trúc MVC đã được ứng dụng để xây dựng rất nhiều framework và thư viện đồ họa khác nhau. Tiêu biểu là bộ thư viện đồ họa của ngôn ngữ lập trình hướng đối tượng SmallTalk (cũng do Xerox PARC nghiên cứu và phát triển vào thập niên 70 của thế kỷ 20). Các Swing Components của Java cũng được xây dựng dựa trên kiến trúc MVC. Đặc biệt là nền tảng ASP.NET MVC Frame work, Zend Framework, CodeIgniter Freamwork và Laravel Framework hay Struts Framework,…

### Ưu Khuyết điểm của mô hình MVC

*Ưu điểm:* MVC làm cho ứng dụng trở nên trong sáng, giúp lập trình viên phân tách ứng dụng thành ba lớp một cách rõ ràng. Điều này sẽ rất giúp ích cho việc phát triển những ứng dụng xét về mặt lâu dài cho việc bảo trì và nâng cấp hệ thống. MVC hiện đang là mô hình lập trình tiên tiến bậc nhất hiện nay, điều mà các framework vẫn đang nổ lực để hướng tới sự đơn giản và yếu tố lâu dài cho người sử dụng.

*Khuyết điểm:* Mặc dù, MVC tỏ ra lợi thế hơn nhiều so với cách lập trình thông thường. Nhưng MVC luôn phải nạp, load những thư viện đồ sộ để xử lý dữ liệu. Chính điều này làm cho mô hình trở nên chậm chạp hơn nhiều so với việc code tay thuần túy. MVC đòi hỏi người tiếp cận phải biết qua OOP, có kinh nghiệm tương đối cho việc thiết lập và xây dựng một ứng dụng hoàn chỉnh. Sẽ rất khó khăn nếu OOP của người sử dụng còn yếu. MVC tận dụng mảng là thành phần chính cho việc truy xuất dữ liệu. Nhất là với việc sử dụng active record để viết ứng dụng. Chúng luôn cần người viết phải nắm vứng mô hình mảng đa chiều.

## JAVA FRAMEWORK

****

Hình 2.6 *Thống kê độ phổ biến của các Java Framework năm 2015*

### Giới thiệu JAVA Framework

Với câu khẩu hiệu là, "viết một lần, chạy mọi nơi", cũng không có gì khó hiểu khi Java đã trở thành ngôn ngữ lập trình phổ biến nhất được sử dụng bởi các lập trình viên. Ngoài ra Java còn là một ngôn ngữ mã nguồn mở, với cộng đồng đông đảo. Java được viết theo hướng đối tượng và tính năng độc lập nền tảng có thể thực thực được trên hầu hết hệ điều hành hiện nay.

*Java FrameWork* là “bộ khung”.*Java* Framework là một thư viện các mã lệnh được xây dựng sẵn để chúng ta sử dụng, nó cung cấp cho chúng ta các chức năng cơ bản cần thiết mà người lập trình viên thường xuyên phải sử dụng khi tiến hành phát triển phần mềm, web,… Có thể ví *Java* framework như tập các “Vật liệu” ở từng lĩnh vực cho người lập trình viên, thay vì họ phải mất nhiều thời gian để tự thiết kế trước khi dùng. Do vậy, người lập trình viên chỉ cần tìm hiểu và khai thác các vật liệu này rồi thực hiện ( tức lập trình ) để gắn kết chúng lại với nhau, tạo ra sản phẩm.

### Tính linh hoạt cơ bản của Java Framework

*Cách tổ chức tập tin:* Tất cả các Java Framework đều có một cấu trúc nhất định, điều này giúp các lập trình viên tổ chức cấu trúc của một ứng dụng đượ dễ dàng hơn và tốn ít thời gian hơn.

*Tái sử dụng mã lệnh:* Các lập trình viên sẽ giảm thiểu được số dòng lệnh và thời gian viết lệnh vì hầu hết các thư viện đó đều được hỗ trợ trong Framework.

*Hỗ trợ của cộng đồng:* Tất cả các Java Framework phổ biến hiện nay luôn được sự trợ giúp của cộng đồng. Mọi thắc mắc hay sự cố gặp phải đều nhanh chóng được giúp đỡ và khắc phục.

*Hiệu suất cao:* Các Java Framework hiện nay rất tuyệt vời với khả năng nạp năng động như chỉ nạp các tài nguyên khi cần thiết, các trang khác nhau có thể nạp những phần khác nhau của thư viện.

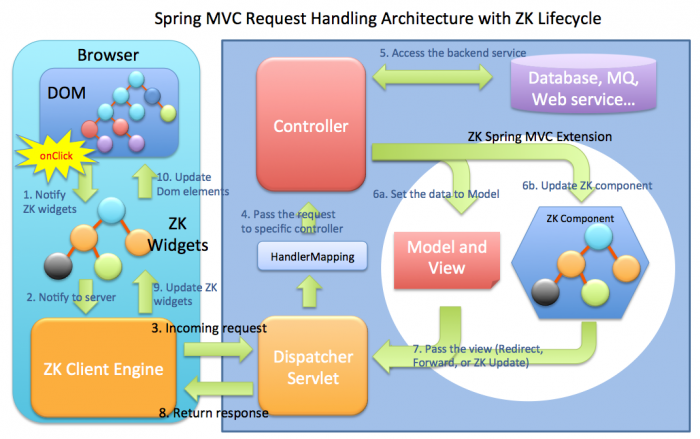
### Các thành phần cơ bản của một Java Framework

*Thư viện mã lệnh (Code Library):* thư viện source code giúp các lập trình viên tra cứu về lập trình. Hỗ trợ rất nhiều ngôn ngữ lập trình: C#, Java, VB, PHP, Javascript…

*Ngôn ngữ kịch bản (Scripting Language):* sử dụng ngôn ngữ kịch bản

*Giao diện lập trình ứng dụng (Application Programming Interface):* sử dụng các lớp, các hàm và các phương thức được xây dựng sẵn

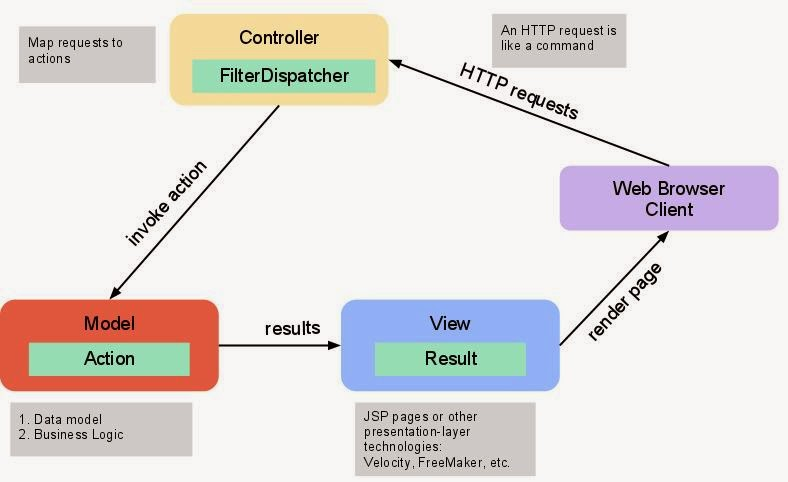
* + 1. **Giới thiệu các Java Framework phổ biến hiện nay**
       1. **Spring Framework**



Hình 2.7 *Spring MVC*

Spring MVC vẫn đi trước hơn một thập kỷ kể từ khi phát hành lần đầu tiên so với các framework khác. Sau khi được phát triển để trở thành MVC hoàn chỉnh, Spring vẫn không ngừng thay đổi để trở nên phổ biến cho các ứng dụng Internet, cung cấp cho các kỹ sư phần mềm một bộ công cụ mạnh mẽ để phát triển ứng dụng web và cấu hình ứng dụng cũng như cho các dự án về bảo mật. Với những ưu điểm đó, Spring MVC đang dần trở nên phổ biến trong cộng đồng lập trình. Nó có một hệ sinh thái được phát triển mạnh mẽ, với nhiều adds-on từ các dịch vụ web REST-APIs, SOAP, bảo mật.

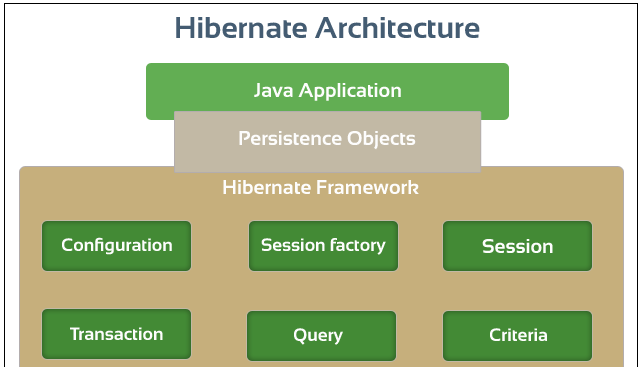
* + - 1. **Struts 2 Framework**

****

Hình 2.8 *Struts Framework*

**Java framework** này là một sự lựa chọn phù hợp cho các kỹ sư làm việc với việc xây dựng các ứng dụng website Java EE hiện đại. Apache Software Foundation đã trang bị cho các developer một công cụ mở rộng để tạo các ứng dụng trực tuyến, sẵn sàng cho doanh nghiệp, tối ưu hóa quá trình phát triển từ đầu đến cuối. Khái niệm này là sự lựa chọn đúng nếu bạn đổ mồ hôi trên các hệ thống có tải trọng cao như cổng phát sóng.

* + - 1. **Hibernate Framework**

**

Hình 2.9 *Hibernate Framework*

Hibernate ra đời năm 2001 bởi nhà sáng lập Gavin King như một sự thay thế cho EJB2 kiểu thực thể bean. Hiện nay, phiên bản mới nhất của Hibernate là 5.2.10.Final. Seri bài hướng dẫn học Hibernate này cung cấp các khái niệm về Hibernate Framework cùng với các ví dụ đơn giản và nâng cao. Seri bài học này rất hữu ích cho người mới bắt đầu và những người đã có kinh nghiệm.

Hibernate được trang bị một bộ tính năng tuyệt vời giúp các nhà phát triển phụ trợ tinh chỉnh lớp truy cập dữ liệu. Đây là một khung ORM tiên tiến cho phép bạn thực hiện hoạt động cơ sở dữ liệu trên các đối tượng Java.

Các lợi ích khi sử dụng Hibernate:

Mã nguồn mở và nhẹ: Hibernate Framework là mã nguồn mở có giấy phép LGPL và nhẹ.

Hiệu suất nhanh: Hiệu suất của Hibernate Framework là nhanh bởi vì bộ nhớ cache được sử dụng trong nội bộ Hibernate Framework. Có hai loại bộ nhớ cache trong Hibernate Framework, gồm bộ nhớ cache cấp một và bộ nhớ cache cấp hai. Bộ nhớ cache cấp một được bật bằng lệnh mặc định.

Truy vấn cơ sở dữ liệu độc lập: HQL (Hibernate Query Language) là phiên bản hướng đối tượng của SQL. Nó tạo ra các truy vấn cơ sở dữ liệu độc lập. Vì vậy, bạn không cần phải viết các truy vấn cơ sở dữ liệu cụ thể. Trước Hibernate, nếu dự án có cơ sở dữ liệu bị thay đổi, chúng ta cần phải thay đổi truy vấn SQL dẫn đến sự cố bảo trì.

Tạo bảng tự động: Hibernate framework cung cấp phương tiện để tạo ra các bảng cơ sở dữ liệu tự động. Vì vậy, không cần phải tạo ra các bảng trong cơ sở dữ liệu bằng tay.

Đơn giản lệnh join phức tạp: Có thể lấy dữ liệu từ nhiều bảng một cách dễ dàng với Hibernate framework.

Cung cấp thống kê truy vấn và trạng thái cơ sở dữ liệu: Hibernate hỗ trợ bộ nhớ cache truy vấn và cung cấp số liệu thống kê về truy vấn và trạng thái cơ sở dữ liệu.

## jQuery LIBRARY

### Giới thiệu về jQuery



Hình 2.10 *JQuery Library*

Với sự phát triển nhanh chống của Internet, người dùng ngày càng quan tâm hơn đến hình thức của một trang web. Trước đây một trang web chỉ cần có banner, nội dung và footer đơn giản là đã được cho là một trang web hoàn chỉnh. Nhưng ngày nay, một trang web cần phải có banner bắt mắt, nội dung hay và còn nhiều hiệu ứng mới có thể thu hút được người đọc.

Chính vì thế những web designer bắt đầu chú ý đến các thư viện JavaScript mở như jQuery để tạo ra các hiệu ứng có thể tương tác trực tiếp với người đọc một cách nhanh chóng và dễ dàng hơn rất nhiều là sử dụng thuần JavaScript.

### jQuery làm được những gì?

*Hướng tới các thành phần trong tài liệu HTML:* Nếu không sử dụng thư viện JavaScript này, chúng ta phải viết rất nhiều dòng code mới có thể đạt được mục tiêu là di chuyển trong cấu trúc cây (hay còn gọi là DOM = Document Object Model) của một tài liệu HTML và chọn ra các thành phần liên quan. Jquery cho phép chúng ta chọn bất cứ thành phần nào của tài liệu để thực hiện một cách dễ dàng như sử dụng CSS.

*Thay đổi giao diện của một trang web:* CSS là công cụ rất mạnh để định dạng một trang web nhưng nó có một nhược điểm là không phải tất cả các trình duyệt đều hiển thị giống nhau. Cho nên jQuery ra đời để lấp chỗ trống này, vì vậy chúng ta có thể sử dụng nó để giúp trang web có thể hiển thị tốt trên hầu hết các trình duyệt. Hơn nữa jQuery cũng có thể thay đổi class hoặc những định dạng CSS đã được áp dụng lên bất cứ thành phần nào của tài liệu HTML ngay cả khi trang web đó đã được trình duyệt load thành công. Thay đổi nội dung của tài liệu. Jquery không phải chỉ có thể thay đổi bề ngoài của trang web, nó cũng có thể thay đổi nội dung của chính tài liệu đó chỉ với vài dòng code. jQuery có thể thêm hoặc bớt nội dung trên trang, hình ảnh có thể được thêm vào hoặc đổi sang hình khác, danh sách có thể được sắp xếp lại hoặc thậm chí cả cấu trúc HTML của một trang web cũng có thể được viết lại và mở rộng. Tất cả những điều này chúng ta hoàn toàn có thể làm được nhờ sự giúp đỡ của API (Application Programming Interface: Giao diện lập trình ứng dụng).

*Tương tác với người dùng:* Cho dù công cụ chúng ta dùng có mạnh mẽ đến mấy, nhưng nếu chúng ta không có quyền quyết định khi nào nó được sử dụng thì công cụ đó cũng xem như không thể sử dụng được. Với thư viện javaScript như jQuery, chúng ta có nhiều cách để tương tác với người dùng ví dụ như khi người dùng nhấp chuột vào đường link thì sẽ có gì xảy ra. Nhưng cái hay của nó là không làm cho code HTML của bạn rối tung lên chính là nhờ các Event Handlers. Hơn nữa Event Handler API sẽ bảo đảm rằng trang web của chúng ta tương thích hầu hết với các trình duyệt.

*Tạo hiệu ứng động cho những thay đổi của tài liệu:* Để tương tác tốt với người dùng, các web designer phải cho người dùng thấy được hiệu ứng gì sẽ xảy ra khi họ làm một tác vụ nào đó. Jquery cho phép bạn sử dụng rất nhiều hiệu ứng động, giúp tạo sự sinh động thêm cho một website.

*Lấy thông tin từ server mà không cần tải lại trang web:* Đây chính là công nghệ ngày càng trở nên phổ biến Asynchronous JavaScript And XML (AJAX), nó giúp người thiết kế web tạo ra những trang web tương tác cực tốt và nhiều tính năng. Thư viện jQuery loại bỏ sự phức tạp của trình duyệt trong quá trình này và cho phép người phát triển web có thể tập trung vào các tính năng đầu cuối. Đơn giản hoá các tác vụ javaScript. Ngoài những tính năng như đã nêu ở trên, jQuery còn cho phép chúng ta viết code javaScript đơn giản hơn nhiều so với cách truyền thống như là các vòng lặp và điều khiển mảng.

### Tại sao jQuery làm việc tốt?

*Tận dụng kiến thức về CSS:* Các jQuery Selector hoạt động giống như CSS Selector với cùng cấu trúc và cú pháp. Chính vì thế thư viện jQuery là cửa ngõ cho các web designer muốn thêm nhiều tính năng hơn nữa cho trang web của mình. Bởi vì điều kiện tiên quyết để trở thành một web designer chuyên nghiệp là khả năng sử dụng CSS thuần thục. Với kiến thức có sẵn về CSS, bạn sẽ có sự khởi đầu thuận lợi với jQuery.

*Hỗ trợ Plugin:* Để tránh bị rơi vào trạng thái quá tải tính năng, jQuery cho phép người dùng tạo và sử dụng Plugin nếu cần. Cách tạo một plugin mới cũng khá đơn giản và được hướng dẫn cụ thể, chính vì thế cộng đồng sử dụng jQuery đã tạo ra một loạt những plugin đầy tính sáng tạo và hữu dụng.

*Xoá nhoà sự khác biệt giữa trình duyệt:* Một thức tế tồn tại là mỗi một hệ thống trình duyệt lại có một kiểu riêng để đọc trang web. Dẫn đến một điều làm đau đầu các web designer là làm thế nào để cho trang web có thể hiển thị tốt trên mọi trình duyệt. Cho nên đôi khi người ta phải làm hẳn một phần code phức tạp để đảm bảo rằng trang web của họ được hiển thị gần như tương đồng ở các trình duyệt phổ biến. Jquery giúp bạn thêm một lớp bảo vệ cho sự khác biệt của trình duyệt và giúp quá trình này diễn ra dễ dàng hơn rất nhiều.

*Luôn làm việc với Set:* Nếu chúng ta yêu cầu jQuery tìm tất cả các thành phần có class là delete và ẩn chúng đi. Chúng ta không cần phải loop qua từng thành phần được trả về. Thay vào đó, những phương pháp như là hide() được thiết kế ra để làm việc với set thay vì từng thành phần đơn lẻ. Kỹ thuật này được gọi là vòng lặp ẩn, điều đó có nghĩa là chúng ta không phải tự viết code để loop nữa mà nó vẫn được thực thi, chính vì thế code của chúng ta sẽ ngắn hơn rất nhiều.

*Cho phép nhiều tác vụ diễn ra trên cùng một dòng:* Để tránh phải sử dụng những biến tạm hoặc các tác vụ lặp tốn thời gian, jQuery cho phép chúng ta sử dụng kiểu lập trình được gọi là Chaining cho hầu hết các method của nó. Điều đó có nghĩa là kết quả của các tác vụ được tiến hành trên một thành phần chính là thành phần đó, nó sẵn sàng cho tác vụ tiếp theo được áp dụng lên nó. Những chiến lược được nêu ở trên giúp kích thước của jQuery rất nhỏ bé chỉ khoảng trên dưới 20Kb dạng nén. Nhưng vẫn đảm bảo cung cấp cho chúng ta những kỹ thuật để giúp code trên trang nhỏ gọn và mạch lạc.

jQuery sở dĩ trở nên phổ biến là do cách sử dụng đơn giản và bên cạnh đó còn có một cộng đồng sử dụng mạnh mẽ vẫn ngày ngày phát triển thêm Plugin và hoàn thiện những tính năng trọng tâm của jQuery. Cho dù thực tế là vậy, nhưng jQuery lại là thư viện javaScript hoàn toàn miễn phí cho mọi người sử dụng. jQuery được bảo vệ bởi luật GNU Public License và MIT License, và có thể sử dụng trong hầu hết các trường hợp kể cả thương mại lẫn cá nhân.

## Bộ thư viện Gradle

### Giới thiệu về Gradle

Trong quá trình phát triển phần mềm, việc tự động hóa quá trình build và quản lý các thư viện rất quan trọng, nhằm tối ưu hóa công đoạn biên dịch, đóng gói; thuận tiện hơn cho việc phát triển phần mềm. Và Gradle (release version 1.0 năm 2012) là 1 trong những công cụ được nhiều người lựa chọn.

Năm 2000, [**Apache Ant**](https://en.wikipedia.org/wiki/Apache_Ant) ra đời như là một trong những tool đóng gói hiện đại và nhanh chóng được sử dụng trong các dự án sử dụng ngôn ngữ Java. Ant cho phép nhà phát triển mô tả meta-data của dự án cũng như các hoạt động trong quá trình build thông qua bộ cú pháp XML. Tuy nhiên những đoạn mã viết theo tiêu chuẩn của Ant quá dài và phức tạp.

Năm 2004, [**Apache Maven**](https://en.wikipedia.org/wiki/Apache_Maven) ra đời và đưa ra các cải thiện so với Ant. Bằng việc đưa ra một cấu trúc dự án tiêu chuẩn, Maven cho phép nhà phát triển sử dụng một trong các plugin của mình thay vì phải viết toàn bộ các dòng code mô tả quá trình đóng gói. Ngoài ra Maven còn cho phép tải về các thư viện phụ thuộc thông qua internet, điều này giúp cho việc chia sẻ cũng như quản lý phiên bản trở nên dễ dàng hơn. Đổi lại, nhà phát triển sẽ mất khá nhiều công sức để tùy chỉnh nếu cấu trúc dự án hiện tại không tuân theo Maven Standard Directory Layout. Và việc vẫn sử dụng XML làm cho file POM (Project Object Model) trở nên dài dòng và phức tạp để nắm bắt logic.

[**Gradle**](https://gradle.org/)kết hợp các ưu điểm của Ant và Maven đồng thời thêm vào các cải thiện mới và sử dụng **Groovy** trong việc giải quyết bài toán đóng gói.

Gradle hỗ trợ build cho **60 ngôn ngữ lập trình** khác nhau bao gồm Java, Scala, Python, C/C++, Android, iOS.

### Sự khác nhau giữa Gradle và Maven

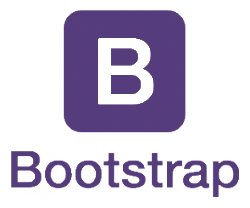
### Build script viết bằng Gradle ngắn gọn và trực quan dễ đọc hơn

### Thời gian build của gradle nhanh hơn

### Gradle tương thích với Ant, Maven.

## Bootstrap Framework

### Giới thiệu về Bootstrap Framework



Hình 2.11 *Bootstrap Framework*

Bootstrap Framework là một thư viện CSS, HTML và JavaScript được phát triển bởi Mark Otto và Jacob Thornton tại Twitter được xuất bản vào năm 2011, framework này cho phép bạn thiết kế giao diện thân thiện với các thiết bị di động hay gọi là responsive design, Bootstrap thiết kế sẵn cho bạn nhiều biểu mẫu cơ bản bằng HTML CSS và JavaScript bạn chỉ việc sử dụng và tùy chỉnh lại theo ý của mình.

### Vì sao nên sử dụng Bootstrap?

*Bootstrap sở dĩ được nhiều người sử dụng* mặc dù mới ra đời trong thời gian vài năm trở lại đây bởi sự ổn định, dễ sử dụng, chạy tốt trên mọi kích cỡ màn hình phát triển mã nguồn mở nên được cộng đồng thế giới đóng góp và hoàn thiện hơn, bạn có thể tạo giao diện web nhanh hơn bình thường rất nhiều.

*Bootstrap cũng như các CSS Framework khác sử dụng đạng grid* để định hình giao diện web nên bạn có thể tùy chỉnh dễ dàng. Grid Bootstrap phân ra 12 cột tương tự như Grid960.

*Bootstrap có cả các bài document chi tiết* rất dễ học, có ví dụ cho từng biểu mẫu nên bạn cứ canh chỉnh theo hướng dẫn và dùng.

*Sử dụng một thư viện JavaScript rất phong phú* gần như hỗ trợ tất cả các biểu mẫu cần thiết cho một website.

Cơ chế có thể tự phát triển theo ý mình dựa trên nền tảng Bootstrap, đây là một trong những điểm mạnh của Bootstrap mà ít có Framework nào hiện tại cạnh tranh được.

## HỆ QUẢN TRỊ CƠ SỞ DỮ LIỆU QUAN HỆ

### Giới thiệu MySQL

**

Hình 2.12 *Hệ quản trị cơ sở dữ liệu MySQL*

*MySQL* là hệ quản trị cơ sở dữ liệu tự do nguồn mở phổ biến nhất thế giới và được các nhà phát triển rất ưa chuộng trong quá trình phát triển ứng dụng. Vì MySQL là cơ sở dữ liệu tốc độ cao, ổn định và dễ sử dụng, có tính khả chuyển, hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh. Với tốc độ và tính bảo mật cao, MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên internet. MySQL miễn phí hoàn toàn cho nên bạn có thể tải về MySQL từ trang chủ. Nó có nhiều phiên bản cho các hệ điều hành khác nhau: phiên bản Win32 cho các hệ điều hành dòng Windows, Linux, Mac OS X, Unix, FreeBSD, NetBSD, Novell NetWare, SGI Irix, Solaris, SunOS,...

*MySQL* là một trong những ví dụ rất cơ bản về Hệ Quản trị Cơ sở dữ liệu quan hệ sử dụng Ngôn ngữ truy vấn có cấu trúc (SQL).

*MySQL* được sử dụng cho việc bổ trợ PHP, Perl, và nhiều ngôn ngữ khác, nó làm nơi lưu trữ những thông tin trên các trang web viết bằng PHP hay Perl,...

### Một số đặc điểm của MySQL

*MySQL* là một phần mềm quản trị CSDL dạng server-based (gần tương đương với SQL Server của Microsoft).

*MySQL* quản lý dữ liệu thông qua các CSDL, mỗi CSDL có thể có nhiều bảng quan hệ chứa dữ liệu.

*MySQL* có cơ chế phân quyền người sử dụng riêng, mỗi người dùng có thể được quản lý một hoặc nhiều CSDL khác nhau, mỗi người dùng có một tên truy cập (user name) và mật khẩu tương ứng để truy xuất đến CSDL.

Khi ta truy vấn tới CSDL MySQL, ta phải cung cấp tên truy cập và mật khẩu của tài khỏan có quyền sử dụng CSDL đó. Nếu không, chúng ta sẽ không làm được gì cả giống như quyền chứng thực người dùng trong SQL Server.

### Mô hình hoạt động của một ứng dụng Web động

Việc kết hợp CSDL với ứng dụng Web, dữ liệu xử lý do PHP sẽ tương tác với CSDL như lấy dữ liệu, làm nội dung trang Web từ trạng thái tĩnh sang trạng thái động. Sự linh hoạt này là cốt lõi của một ứng dụng Web động (dynamic web application).



Hình 2.13 *Mô hình hoạt động của một ứng dụng Web động*

Quá trình bắt đầu khi người dùng truy cập một trang Web thông qua trình duyệt web. Người dùng có thể nhập URL của trang web trên thanh địa chỉ của trình duyệt, nhấp vào liên kết tới một trang web động, nhấn nút submit form chứa dữ liệu mà web động có thể xử lý.

Trong mỗi trường hợp trên, trình duyệt gửi một HTTP request tới Web Server. Khi web server nhận được HTTP request từ trình duyệt, nó dựa vào phần mở rộng của yêu cầu để xác định xem máy chủ hay chương trình sẽ xử lý yêu cầu. Đối với trang web PHP, web server chuyển yêu cầu tới PHP interpreter (trình thông dịch PHP) chạy trên web server. Khi PHP interpreter hoàn thành xử lý trang PHP, nó sinh ra trang HTML và trả về web server. Sau đó web server sẽ gửi trang web tới trình duyệt. Trình duyệt có nhiệm vụ hiển thị mã HTML mà nó nhận được từ web server.

PHP dùng để gửi yêu cầu cho Server và PHP đóng vai trò làm thông dịch và lấy kết quả có được trả lại cho Client ở dạng mã HTM L cho người dùng hiểu và sử dụng. Đây là những gì diễn ra khi Client yêu cầu Server cho phép gửi một yêu cầu lên form.

*Mô hình tương tác giữa client – server: PHP và MySQL*:

* Server sử dụng PHP để thực hiện việc chuyển tải yêu cầu.
* PHP thực hiện việc cần làm, rồi gửi báo cáo đến MySQL để lưu trữ tin nhắn, PHP lưu yêu cầu vào MySQL và sau này lấy lại để sử dụng.
* MySQL đồng ý cho PHP thao tác trên mình.
* PHP trả lời cho Server MySQL đã chấp nhận yêu cầu, lệnh đã được gửi. Server cho thể in nội dung tin nhắn.

## Mô tả WEBSITE

### Mô tả trang người sử dụng (Users)

* + - 1. **Trang chủ**

Đây là giao diện chính của Website, chứa các thông tin và tin tức mới nhất về sản phẩm, giỏ hàng cho khách hàng đặt hàng trực tuyến; có các mục hỗ trợ lọc điều kiện tìm kiếm; trang đặt bàn, trang đặt hàng trực tuyến; trang xem chi tiết sản phẩm, trang xem phản hồi từ khách hàng khác.

* + - 1. **Trang đặt bàn và đặt hàng**

*Chứa các form để khách hàng nhập thông tin cá nhân. Ngoài ra còn có hiển thị các loại bàn, hiển thị các sản phẩm mà khách hàng đã đưa vào giỏ hàng để có thể tiện chỉnh sửa.*

* + - 1. **Trang xem chi tiết sản phẩm**

*Chứa thông tin về sản phẩm đó. Chẳng hạng như mức giá, mức giảm giá, tên sản phẩm, loại sản phẩm, mô tả, các hình ảnh về sản phẩm.*

* + - 1. **Trang bình luận về sản phẩm**

Chứa các lời phản hồi của khách hàng khác về chất lượng sản phẩm, cũng như các lời góp ý để cải thiện chất lượng sản phẩm.

### Mô tả trang người quản trị (Administrators)

* + - 1. **Trang chủ**

Trang chủ chứa các phần truy cập nhanh của Website (Administrators), gồm các phần chính của Website (Administrators): Quản lý bàn, quản lý đơn đặt bàn, dặt hàn trực tuyến, chat trực tuyến với khách hàng.

* + - 1. **Quản lý kho**

Người quản trị Website (Administrators) có thể *thêm sửa xóa các nguyên liệu nhập vào kho, nguyên liệu xuất kho, các hóa đơn nhập và xuất nguyên liệu, các sản phẩm tồn kho,...*

* + - 1. **Quản lý các thông tin kinh doanh**

Người quản trị Website (Administrators) có thể thêm sửa xóa các sản phẩm, bàn ghế, các thông tin khách hàng,...

* + - 1. **Quản lý nhân viên**

Người quản trị Website (Administrators) có thể thêm sửa xóa các nhân viên, xem ca làm của nhân viên, xem lương nhân viên, thống kê lương nhân viên,...

* + - 1. **Thống kê**

Người quản trị Website (Administrators) có thể thống kê tình hình kinh doanh , các sản phẩm bán chạy, thu chi, lãi lỗ theo ngày, tuần, tháng, năm,...

# NỘI DUNG VÀ KẾT QUẢ NGHIÊN CỨU

## PHÂN TÍCH VÀ THIẾT KẾ CÁC SƠ ĐỒ

### Sơ đồ phân rã chức năng (Bussiness Function Diagram – BFD)

Sơ đồ phân rã chức năng (Bussiness Function Diagram – BFD) mô tả chức năng của hệ thống, phân rã có thứ bậc đơn giản các công việc cần thực hiện.

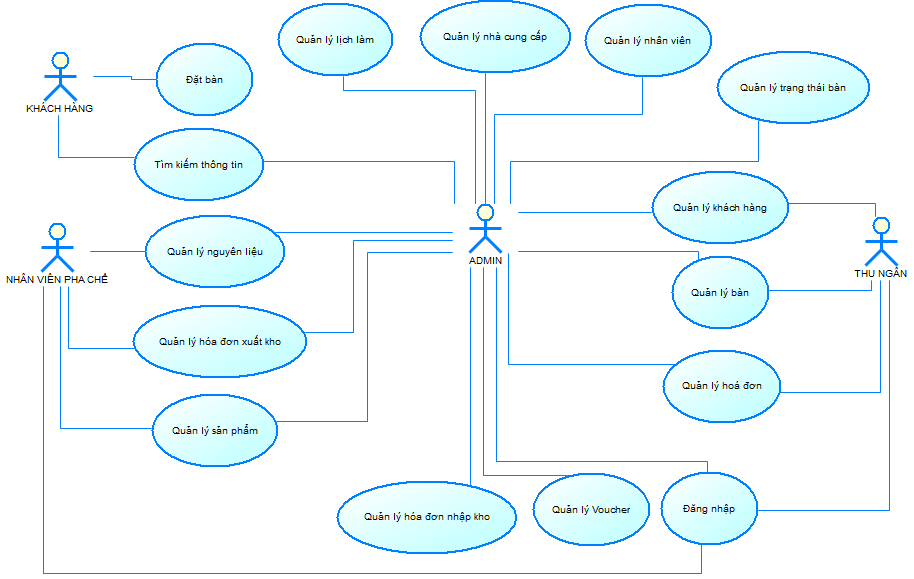
* + - 1. **Sơ đồ BFD vai trò người sử dụng (Users)**
      2. **Sơ đồ BFD vai trò người quản trị Website (Administrators)**
  1. ***Sơ dồ BFD vai trò người quản trị Website (Administrators)***

### Sơ đồ luồng dữ liệu (Data Flow Diagram – DFD)

Sơ đồ luồng dữ liệu (Data Flow Diagram – DFD) là sơ đồ mô tả sự trao đổi, luân chuyển thông tin dữ liệu giữa các đối tượng (Administrators – User).

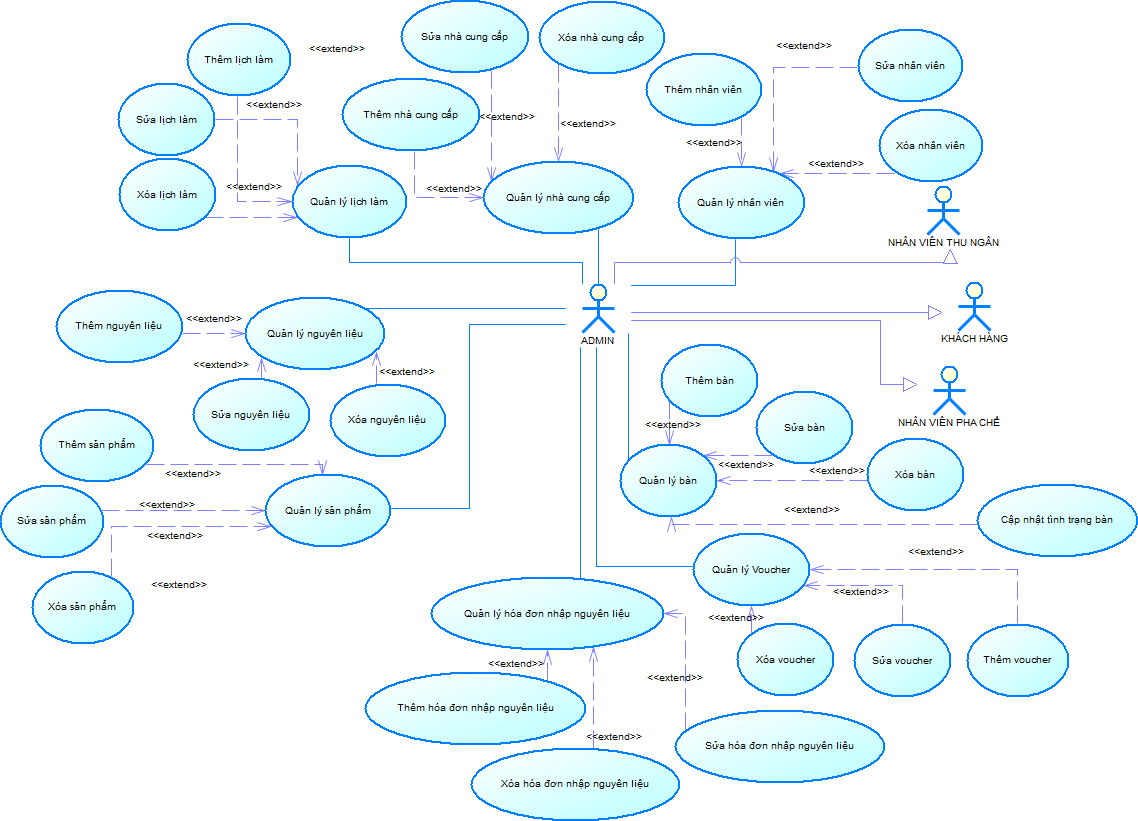
* Cung cấp cái nhìn tổng thể về mối quan hệ giữa các thành phần *Dữ liệu* và *Xử lý.*
* Cơ sở ban đầu cho việc mô hình hóa thành phần *Xử lý.*
  + - 1. **Sơ đồ DFD mức 0**
      2. **Sơ đồ DFD mức 1**
      3. **Sơ đồ DFD mức 2 phân rã chức năng 1.0, 2.0**
      4. **Sơ đồ DFD mức 2 phân rã chức năng 3.0**
      5. **Sơ đồ DFD mức 2 phân rã chức năng 4.0**
      6. **Sơ đồ DFD mức 2 phân rã chức năng 5.0**
      7. **Sơ đồ DFD mức 2 phân rã chức năng 6.0**

### Sơ đồ trường hợp sử dụng (Use case Diagram)

* + - 1. **Sơ đồ Use case mức 0**

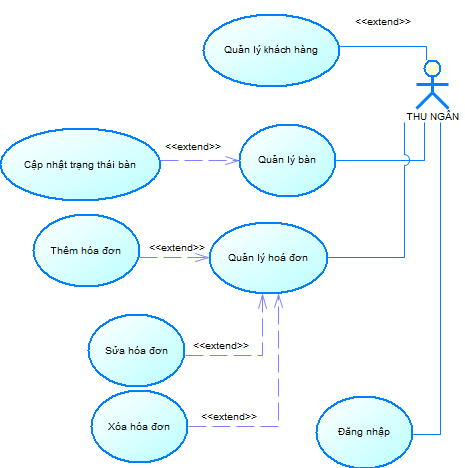
**Hình 3.1** Sở đồ use case mức 0

* + - 1. **Sơ đồ Use case mức 1 của Admin**

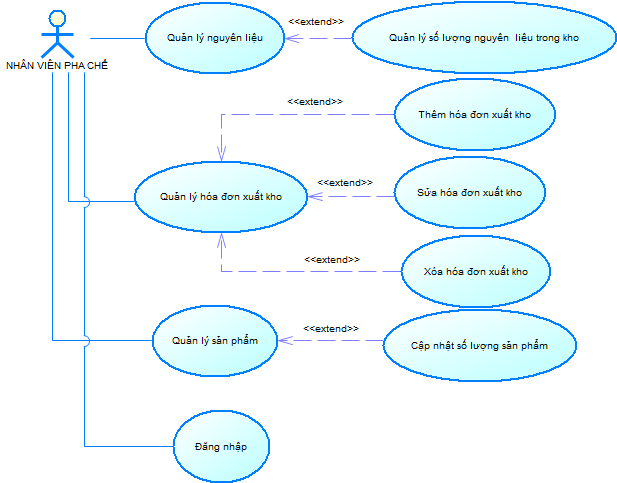


Hình 3.2 *Sơ đồ Use Case mức 1 các chức năng quản lý của Admin*

* + - 1. **Sơ đồ Use case mức 1 các chức năng quản lý của nhân viên thu ngân**

****

* 1. *Sơ đồ Use Case mức 1 các chức năng quản lý của nhân viên thu ngân*
     + 1. **Sơ đồ Use case mức 1 các chức năng quản lý của nhân viên pha chế**

****

Hình 3.3 *Sơ đồ Use Case mức 1 các chức năng quản lý của nhân viên pha chế*

### Bảng mô tả từng trường hợp sử dụng (Use Case)

* + - 1. **Mô tả Use case các chức năng quản lý của admin**

|  |  |
| --- | --- |
| Trường hợp sử dụng | Quản lý tất cả hệ thống |
| Tác nhân | Admin. |
| Tóm tắt | Tất cả các chức năng của website đều được admin sử dụng. |
| Các dòng sự kiện chính | 1. Quản lý thêm sửa xóa nhân viên, thêm sửa xóa ca làm của nhân viên 2. Quản lý Website thêm, sửa hoặc xóa nhà cung cấp. 3. Quản lý Website thêm sửa xóa bàn,. 4. Quản lý Website thêm sửa xóa trạng thái bàn. 5. Quản lý Website thêm sửa xóa khách hàng. 6. Quản lý Website thêm sửa xóa hóa đơn. 7. Quản lý Website thêm sửa xóa Voucher. 8. Quản lý Webstite thêm sửa xóa hóa đơn nhập kho và hóa đơn xuất kho. 9. Quản lý Website thêm sửa xóa sản phẩm |
| Các dòng sự kiện phụ | 3.1. Hệ thống yêu cầu nhập thông tin và chọn hình ảnh hoặc videos cho bài viết, bản tin.  4.1. Hệ thống yêu cầu quản lý website xác nhận có thật sự muốn xóa thông tin đã chọn.  5.1. Thông tin còn thiếu hoặc không hợp lệ (tên bài viết, bản tin quá ngắn hoặc quá dài), hệ thống yêu cầu nhập đầy đủ thông tin và phải hợp lệ. |
| Yêu cầu tiên quyết | Admin quản lý Website phải đăng nhập vào hệ thống. |
| Yêu cầu cơ bản | Giao diện thân thiện với người dùng. |

* 1. ***Mô tả Use case các chức năng quản lý của admin***
     + 1. **Mô tả Use case các chức năng quản lý của nhân viên thu ngân**

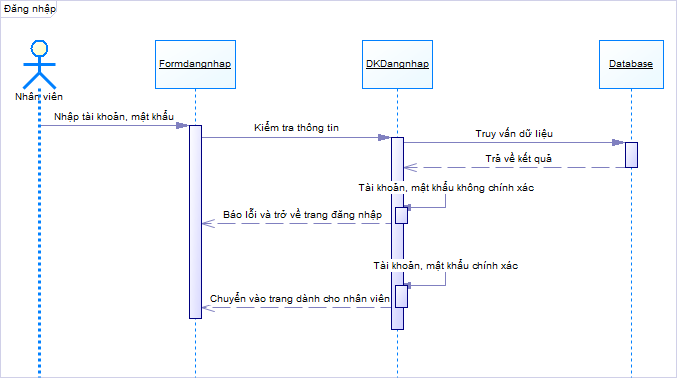
|  |  |
| --- | --- |
| Trường hợp sử dụng | Quản lý khách hàng, cập nhật bàn, thêm sửa xóa hóa đơn mua hàng của khách hàng. |
| Tác nhân | Nhân viên thu ngân. |
| Tóm tắt | Nhân viên thu ngân có nhu cầu xem danh sách, thêm, sửa, xóa khách hàng. Có nhu cầu cập nhật trạng thái bàn. Thêm sửa xóa hóa đơn của khach hàng. |
| Các dòng sự kiện chính | 1. Quản lý thêm sửa xóa thông tin khách hàng. 2. Cập nhật trạng thái bàn. 3. Xem, thêm, sửa, xóa hóa đơn. |
| Các dòng sự kiện phụ | 3.1. Hệ thống yêu cầu nhập thông tin khách hàng.  4.1. Hệ thống yêu cầu quản lý website xác nhận có thật sự muốn xóa bản tin đã chọn.  5.1. Thông tin bản tin còn thiếu hoặc không hợp lệ hệ thống yêu cầu nhập đầy đủ thông tin và phải hợp lệ. |
| Yêu cầu tiên quyết | Nhân viên thu ngân Website phải đăng nhập vào hệ thống. |
| Yêu cầu cơ bản | Giao diện thân thiện với người dùng. |

* 1. ***Mô tả Use case các chức năng quản lý của nhân viên thu ngân***
     + 1. **Mô tả use case các chức năng quản lý của nhân viên pha chế**

|  |  |
| --- | --- |
| Trường hợp sử dụng | Quản lý nguyên liệu, quản lý hóa đơn nhập kho, quản lý hóa đơn xuất kho, quản lý sản phẩm. |
| Tác nhân | Nhân viên quản lý Website. |
| Tóm tắt | Nhân viên pha chế có nhu cầu xem số lượng nguyên liệu, thêm sửa xóa hóa đơn xuất kho, cập nhật số lượng sản phẩm. |
| Các dòng sự kiện chính | 1. Hệ thống hiển thị số lượng nguyên liệu, số lượng nguyên liệu trong kho. 2. Hệ thống cho phép xem, thêm, sửa, xóa hóa đơn xuất kho. 3. Hệ thống cho phép xem và cập nhật số lượng sản phẩm. 4. Hệ thống thực hiện yêu cầu của quản lý Website và thực hiện yêu cầu thành công hay thất bại. |
| Các dòng sự kiện phụ | 3.1. Hệ thống yêu cầu nhập thông tin.  4.1. Thông tin câu trả lời còn thiếu hoặc không hợp lệ (câu trả lời quá ngắn hoặc quá dài), hệ thống yêu cầu nhập đầy đủ thông tin và phải hợp lệ. |
| Yêu cầu tiên quyết | Nhân viên pha chế phải đăng nhập vào hệ thống. |
| Yêu cầu cơ bản | Giao diện thân thiện với người dùng. |

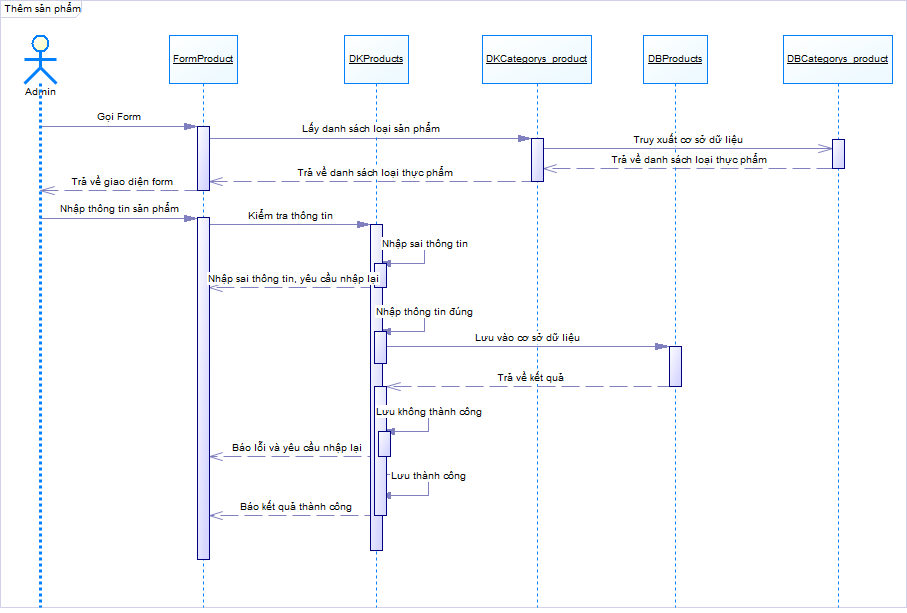
* 1. ***Mô tả use case các chức năng quản lý của nhân viên pha chế***

### Sơ đồ tuần tự (Sequence Diagram)

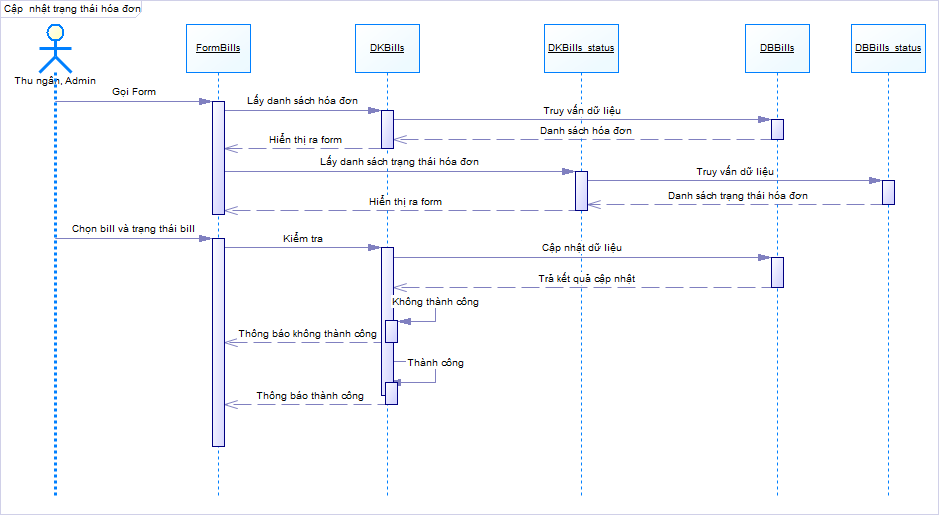
* + - 1. **Sơ đồ tuần tự chức năng đăng nhập**

Hình 3.4 *Sơ đồ tuần tự chức năng đăng nhập*

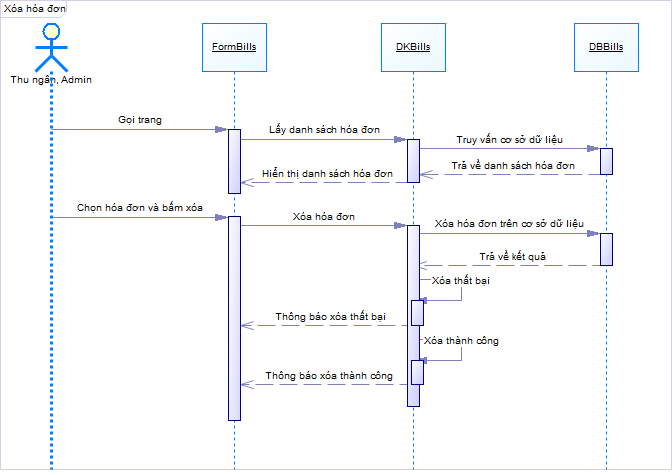
* + - 1. **Sơ đồ tuần tự thêm mới sản phẩm**

****

Hình 3.5 *Sơ đồ tuần tự chức năng thêm mới sản phẩm*

* + - 1. **Sơ đồ tuần tự cập nhật trạng thái hóa đơn**

Hình 3.6*Sơ đồ tuần tự chức năng cập nhật trạng thái hóa đơn*

* + - 1. **Sơ đồ tuần tự xóa hóa đơn**
  1. *Sơ đồ tuần tự chức năng xóa hóa đơn*

## CƠ SỞ DỮ LIỆU

### Mô hình thực thể mối quan hệ (Entity Relationship)

* 1. Mô hình thực thể mối quan hệ (Entity Relationship)

### Mô hình vật lý (Physical Diagram)

* 1. Mô hình vật lý (Physical Diagram)

### Mô tả chi tiết các bảng dữ liệu

* + - 1. **Bảng dữ liệu Unit**

|  |  |  |  |
| --- | --- | --- | --- |
| **Unit** | | | |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Mô tả |
| unitID | text | Khóa chính | ID phân biệt các loại đơn vị với nhau. |
| name | text | Không rỗng | Tên đơn vị. |

Bảng 3.1 *Mô tả bảng dữ liệu Unit (Đơn vị)*

* **Ràng buộc toàn vẹn cho bảng Unit**
* R1: Các loại đơn vị phải có mã số phân biệt với nhau.
* R2: Tên phải là duy nhất trong bảng Unit.
  + - 1. **Bảng dữ liệu Material**

|  |  |  |  |
| --- | --- | --- | --- |
| **Material** | | | |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Mô tả |
| materialID | integer | Khóa chính | ID phân biệt các nguyên liệu với nhau. |
| unitID | Text | Khóa ngoại | Mã số của đơn vị của bảng Unit và nguyên liệu được tính theo đơn vị nào. |
| name | Text | Khóa ngoại | Mã số loại danh mục mà danh mục thuộc về. |

Bảng 3.2 *Mô tả bảng dữ liệu Material (Danh mục)*

* **Ràng buộc toàn vẹn cho bảng *Material***
* R1: Các nguyện liệu phải có mã số phân biệt với nhau.
* R2: Nguyên liệu phải được tính theo một đơn vị nào đó thì đơn vị đó phải có trong danh sách các đơn vị.
  + - 1. **Bảng dữ liệu ExportBill**

|  |  |  |  |
| --- | --- | --- | --- |
| **ExportBill** | | | |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Mô tả |
| exportBillID | integer | Khóa chính | ID phân biệt các hóa đơn xuất với nhau. |
| productID | Text | Khóa ngoại | Hóa đơn xuất để chế biến cho sản phẩm nào. |
| materialDetailID | integer | Khóa ngoại | Mã số của chi tiết hóa đơn xuất. |
| employeeID | text | Khóa ngoại | Mã số nhân viên lập hóa đơn |
| quantity | integer | Lớn hơn 0 | Số lượng nguyên liệu xuất. |

Bảng 3.3 Mô tả bảng dữ liệu ExportBill (xuất hóa đơn)

* **Ràng buộc toàn vẹn cho bảng *ExportBill***
* R1: Các hóa đơn phải có phải có mã số phân biệt với nhau.
* R2: Hóa đơn chế biến cho sản phẩm nào, thì sản phẩm đó phải tồn tại trong bảng Product.
* R3: Mã số của chi tiết phiếu xuất phải tồn tại trong danh sách materialDetail.
* R4: ExportBill do một nhân viên nào đó tạo ra thì nhân viên phải có trong danh sách Employee.
  + - 1. **Bảng dữ liệu MaterialDetail**

|  |  |  |  |
| --- | --- | --- | --- |
| **MaterialDetail** | | | |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Mô tả |
| materialDetailID | integer | Khóa chính | ID phân biệt các chi tiết nguyên liệu với nhau. |
| importBillID | integer | Khóa ngoại | Mã số hóa đơn nhập mà chi tiết nguyên liệu thuộc về. |
| materialID | integer | Khóa ngoại | Mã số nguyên liệu mà chi tiết nguyên liệu thuộc về. |
| dateOfManufacture | date | Không rỗng | Ngày nguyên liệu sản xuất. |
| expirationDate | date | Không rỗng và lớn hơn dateOfManufacture. | Ngày nguyên liệu hết hạn. |
| quantity | integer | Lớn hơn 0 | Số lương của nguyên liệu nhập vào kho. |

Bảng 3.4 Mô tả bảng dữ liệu MaterialDetail (Chi tiết nguyên liệu)

**Ràng buộc toàn vẹn cho bảng MaterialDetail**

* R1: Các chi tiết nguyên liệu phải có mã số phân biệt với nhau.
* R2: Mã số hóa đơn nhập phải tồn tại trong danh sách hóa đơn nhập.
* R3: Mã số nguyên liệu phải tồn tại trong danh sách các nguyên liệu.
* R3: Ngày hết hạn phải lớn hơn ngày sản xuất.
  + - 1. **Bảng dữ liệu ImportBill**

|  |  |  |  |
| --- | --- | --- | --- |
| **ImportBill** | | | |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Mô tả |
| importBillID | integer | Khóa chính | ID phân biệt các hóa đơn nhập khác. |
| supplierID | integer | Khóa ngoại | Mã số nhà cung cấp cung cấp nguyên liệu. |

Bảng 3.5 Mô tả bảng dữ liệu importBill (hoa đơn nhập)

* **Ràng buộc toàn vẹn cho bảng *importBill***
* R1: Các hóa đơn nhập phải có mã số phân biệt với nhau.
* R2: Mã số của nhà cung cấp phải thuộc danh sách các nhà cup cấp trong bảng Supplier .
  + - 1. **Bảng dữ liệu Supplier**

|  |  |  |  |
| --- | --- | --- | --- |
| **Supplier** | | | |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Mô tả |
| supplierID | integer | Khóa chính | ID phân biệt các nhà cung cấp với nhau. |
| name | varchar(255) | Không rỗng | Tên nhà cung cấp. |
| address | varchar(255) | Không rỗng | Email nhà cung cấp. |
| phone | numberic | Không rỗng | Số điện thoại nhà cung cấp. |

Bảng 3.6 Mô tả bảng dữ liệu Supplier (Nhà cung cấp)

* **Ràng buộc toàn vẹn cho bảng Supplier**
* R: Các nhà cung cấp phải có mã số phân biệt với nhau.
  + - 1. **Bảng dữ liệu BillStatus (tình trạng hóa đơn)**

|  |  |  |  |
| --- | --- | --- | --- |
| **BillStatus** | | | |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Mô tả |
| billStatusID | Varchar(7) | Khóa chính | ID phân biệt các BillStatus với nhau. |
| name | varchar | Không rỗng | Tên BillStatus. |

Bảng 3.7 Mô tả bảng dữ liệu members (Tài khoản người dùng)

* **Ràng buộc toàn vẹn cho bảng BillStatus**
  + R1: Các BillStatus phải có mã số phân biệt với nhau.
  + R2: Tên của BillStatus phải là duy nhất.
    - 1. **Bảng dữ liệu Customer (khách hàng)**

|  |  |  |  |
| --- | --- | --- | --- |
| **customer** | | | |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Mô tả |
| customerID | integer | Khóa chính | ID phân biệt các khách hàng với nhau. |
| name | Varchar | Cho phép rỗng | Tên khách hàng. |
| address | varchar | Cho phép rỗng | Địa chỉ khách hàng. |
| phone | number | Cho phép rỗng | Số điện thoại. |

Bảng 3.8 Mô tả bảng dữ liệu customer (Khách hàng)

* **Ràng buộc toàn vẹn cho bảng customer**
* R1: Các customer phải có mã số phân biệt với nhau

**Bảng dữ liệu Voucher**

|  |  |  |  |
| --- | --- | --- | --- |
| **voucher** | | | |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Mô tả |
| id voucherID | integer | Khóa chính | ID phân biệt các voucher. |
| name | varchar(50) | Không rỗng | Tên voucher. |
| startDateTime | Date time | Không rỗng | Ngày áp dụng |
| endDate | Date time | Không rỗng | Ngày kết thúc |
| number | integer | Không rỗng | Số lượng phát hàng |
| count | Integer | Không rỗng | Số lượng |
| saleof | Integer | Không rỗng | Giá trị giảm giá |

Bảng 3.9 Mô tả bảng dữ liệu voucher (Mã giảm giá)

* **Ràng buộc toàn vẹn cho bảng voucher**
* R: Các voucher phải có mã số phân biệt với nhau
  + - 1. **Bảng dữ liệu Bill**

|  |  |  |  |
| --- | --- | --- | --- |
| **Bill** | | | |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Mô tả |
| billID | integer | Khóa chính | ID phân biệt các hóa đơn hỏi đáp với nhau. |
| billStatusID | text | Khóa ngoại | Mã số tình rạng hóa đơn |
| dinnerTableID | Integer | Khóa ngoại | Mã số bàn ăn |
| customerID | Integer | Khóa ngoại | Mã số khách hàng |
| employeeID | Varchar(7) | Khóa ngoại | Mã số nhân viên |
| voucherID | Integer | Khóa ngoại | Mã giảm giá |
| startDateTime | Time | Không rỗng | Ngày giao hàng đối với đặt hàng trực tuyến và là ngày tạo hóa đơn đối với khách hàng đến quán. |
| endDate | Date | Không rỗng | Ngày thanh toán hóa đơn |
| notice | Varchar(255) | Cho phép rỗng | Ghi chú |

Bảng 3.10 Mô tả bảng dữ liệu bill (Hóa đơn)

* **Ràng buộc toàn vẹn cho bảng *bill***
* R1: Hóa đơn phải có mã số phân biệt với nhau.
* R2: Mã tình trạng hóa đơn phải tồn tại trong danh sách BillStatus.
* R3: Mã bàn ăn phải tồn tại trong danh sách DinnerTable.
* R4: Mã khách hàng phải tồn tại trong danh sách Customer.
* R5: Mã nhân viên phải tồn tại trong danh sách Employee.
* R6: Ngày thanh toán không nhỏ hơn ngày giao hàng.
  + - 1. **Bảng BillDetail**

|  |  |  |  |
| --- | --- | --- | --- |
| **BillDetail** | | | |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Mô tả |
| productID | Varchar(7) | Khóa ngoại | Mã sản phẩm. |
| billID | integer | Khóa ngoại | Mã hóa đơn mà billDetail thuộc về. |
| quantity | integer | Phải lớn hơn 0 | Số lượng sản phẩm bán. |

Bảng 3.11 Mô tả bảng dữ liệu billDetail (chi tiết hóa đơn)

* **Ràng buộc toàn vẹn cho bảng billDetail**
* R1: Mã sản phẩm phải tồn tại trong danh sách Product.
* R2: Mã hóa đơn phải tồn tại trong danh sách Bill.
* R3: Số lượng sản phẩm bán phải lớn hơn 0.
  + - 1. **Bảng dữ liệu sản phẩm**

|  |  |  |  |
| --- | --- | --- | --- |
| **loaisanpham** | | | |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Mô tả |
| productID | Varchar(7) | Khóa chính | ID phân biệt các sản phẩm với nhau. |
| categoryProductID | Varchar(7) | Khóa ngoại | Mã loại sản phẩm. |
| name | Varchar(255) | Không rỗng | Tên sản phẩm |
| description | Varchar(255) | Cho phép rỗng | Mô tả |
| numberInventory | integer | Phải >= 0 | Số lượng tồn kho |

Bảng 3.12 Mô tả bảng dữ liệu product (Sản phẩm)

* **Ràng buộc toàn vẹn cho bảng loaisanpham**
* R1: Sản phẩm phải có mã số để phân biệt với nhau.
* R2: Mã số loại sản phẩm phải tồn tại trong danh sách loại sản phẩm.
* R3: Số lượng phải lớn hơn bằng 0.
  + - 1. **Bảng dữ liệu price**

|  |  |  |  |
| --- | --- | --- | --- |
| **Price** | | | |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Mô tả |
| priceID | integer | Khóa chính | ID phân biệt các giá với nhau. |
| productID | Varchar(7) | Khóa ngoại | Mã sản phẩm có giá. |
| startDateTime | Date | Không rỗng | Ngày giá được áp dụng |
| price | Integer | Giá phải lớn hơn 0 | Giá. |

Bảng 3.13 Mô tả bảng dữ liệu price (Giá)

* **Ràng buộc toàn vẹn cho bảng price**
* R1: Các giá trong danh sách giá phải có mã số phân biệt với nhau.
* R2: Mã sản phẩm phải tồn tại trong danh sách Price.
* R3: Giá phải lớn hơn 0.
* **Bảng dữ liệu image**

|  |  |  |  |
| --- | --- | --- | --- |
| **Image** | | | |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Mô tả |
| imageID | integer | Khóa chính | ID phân biệt các hình ảnh với nhau. |
| name | Varchar(255) | Không rỗng | Tên hình ảnh. |

Bảng 3.14 Mô tả bảng dữ liệu image (Hình ảnh)

* **Ràng buộc toàn vẹn cho bảng image**
* R: Các hình ảnh trong danh sách phải có mã số phân biệt với nhau
  + - 1. **Bảng dữ liệu salary**

|  |  |  |  |
| --- | --- | --- | --- |
| **salary** | | | |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Mô tả |
| salaryID | integer | Khóa chính | ID phân biệt các mức lương với nhau. |
| employeeID | Varchar(7) | Khóa ngoại | ID nhân viên có mức lương. |
| salaryOnHour | integer | Không rỗng và lớn hơn 0 | Mức lương trong 1 giờ. |
| startDate | Date | Không rỗng | Ngày áp dụng. |

Bảng 3.15 Mô tả bảng dữ liệu salary (bảng lương)

* **Ràng buộc toàn vẹn cho bảng salary**
* R1**:** Các mức lương trong bảng salary phải được phân biệt bằng mã số.
* R2: Mã số nhân viên trong bảng mức lương thì mã số đó phải có trong danh sách nhân viên.
  + - 1. **Bảng dữ liệu ImageOfEmpoyee**

|  |  |  |  |
| --- | --- | --- | --- |
| **ImageOfEmployee** | | | |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Mô tả |
| imageID | integer | Khóa ngoại | Mã hình ảnh |
| employeeID | Varchar(7) | Khóa ngoại | Mã nhân viên. |

Bảng 3.16 Mô tả bảng dữ liệu ImageOfEmployee (hình ảnh của nhân viên)

* **Ràng buộc toàn vẹn cho bảng binhluan**
* R1: Mã hình phải tồn tại trong danh sách Image.
* R2: Mã nhân viên phải tồn tại trong danh sách Employee.
  + - 1. **Bảng dữ liệu ImageOfProduct**

|  |  |  |  |
| --- | --- | --- | --- |
| **ImageOfEmployee** | | | |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Mô tả |
| imageID | integer | Khóa ngoại | Mã hình ảnh |
| productID | Varchar(7) | Khóa ngoại | Mã sản phẩm. |

Bảng 3.17 Mô tả bảng dữ liệu ImageOfProduct (hình ảnh của sản phẩm)

* **Ràng buộc toàn vẹn cho bảng binhluan**
* R1: Mã hình phải tồn tại trong danh sách Image.
* R2: Mã sản phẩm phải tồn tại trong danh sách Product.
  + - 1. **Bảng dữ liệu CategoryProduct**

|  |  |  |  |
| --- | --- | --- | --- |
| **CategoryProduct** | | | |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Mô tả |
| categoryProductID | Varchar(7) | Khóa chính | Mã loại sản phẩm |
| name | Varchar(255) | Không rỗng | Tên loại sản phẩm. |

Bảng 3.18 Mô tả bảng dữ liệu CategoryProduct (loại sản phẩm)

* **Ràng buộc toàn vẹn cho bảng CategoryProduct**
* R1: Mã loại sản phẩm phải phân biệt với loại sản phẩm khác trong danh sách CategoryProduct.
  + - 1. **Bảng dữ liệu Employee**

|  |  |  |  |
| --- | --- | --- | --- |
| **Employee** | | | |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Mô tả |
| employeeID | Varchar(7) | Khóa chính | Mã nhân viên |
| name | Varchar(255) | Không rỗng | Tên nhân viên. |
| sex | Boolean | Không rỗng | Giới tính |
| phone | Numberic | Không rỗng | Điện thoại |
| address | Varchar(255) | Không rỗng | Địa chỉ |
| usename | Varchar(255) | Không rỗng và duy nhất | Tài khoản |
| password | Varchar(255) | Không rỗng | Mật khẩu |

Bảng 3.19 Mô tả bảng dữ liệu Employee (nhân viên)

* **Ràng buộc toàn vẹn cho bảng Employee**
* R1: Mã nhân viên phải phân biệt với nhân viên khác trong danh sách Employee.
* R2: Tài khoản phải phân việt với các tài khoản khác trong danh sách Tài khoản của Employee e.
  + - 1. **Bảng dữ liệu DinnerTable**

|  |  |  |  |
| --- | --- | --- | --- |
| **DinnerTable** | | | |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Mô tả |
| dinnerTableID | integer | Khóa chính | Mã bàn ăn. |
| name | Varchar(255) | Không rỗng | Tên bàn ăn. |
| countChair | integer | Không rỗng | Số lượng ghế. |

Bảng 3.20 Mô tả bảng dữ liệu DinnerTable (bàn ăn)

* **Ràng buộc toàn vẹn cho bảng DinnerTable**
* R1: Mã bàn ăn phải phân biệt với khác mã khác trong danh sách DinnerTable.
* của Employee e.
  + - 1. **Bảng dữ liệu BookTable**

|  |  |  |  |
| --- | --- | --- | --- |
| **Booktable** | | | |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Mô tả |
| customerID | Integer | Khóa chính | Mã khách hàng |
| dinnerTableID | Integer | Khóa ngọai | Mã bàn ăn |
| employeeID | Varchar(7) | Khóa ngọai | Mã nhân viên |
| startDateTime | Date time | Không nhỏ hơn ngày hiện tại | Ngày đến |
| status | Boolean | Không rỗng | Trạng thái duyệt |
| countPeople | Integer | Không rỗng | Số người |
| notice | Varchar(255) | Có thể rỗng | Ghi chú |

Bảng 3.21 Mô tả bảng dữ liệu BookTable (Đặt bàn)

* **Ràng buộc toàn vẹn cho bảng BookTable**
* R1: Mã số khách hàng trong bảng đặt bàn thì mã số đó phải có trong danh sách khách hàng.
* R2: Mã số bàn ăn trong bảng đặt bàn thì mã số đó phải có trong danh sách bàn ăn.
* R3: Mã số nhân viên trong bảng đặt bàn thì mã số đó phải có trong danh sách nhân viên.
* R4: Mã số nhân viên trong bảng mức lương thì mã số đó phải có trong danh sách nhân viên.
  + - 1. **Bảng dữ liệu TableStatus**

|  |  |  |  |
| --- | --- | --- | --- |
| **TableStatus** | | | |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Mô tả |
| tableStatusID | Integer | Khóa chính | Mã trạng thái bàn |
| name | Integer | Không rỗng | Tên bàn |

Bảng 3.22 Mô tả bảng dữ liệu TableStatus (Trạng thái bàn)

* **Ràng buộc toàn vẹn cho bảng TableStatus**
* R1: Mã số trạng thái bàn phải là duy nhất trong danh sách trạng thái bàn.
* R2: Tên trạng thái bàn phải là duy nhất trong danh sách trạng thái bàn.
  + - 1. **Bảng dữ liệu Position**

|  |  |  |  |
| --- | --- | --- | --- |
| **Position** | | | |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Mô tả |
| positionID | Varchar(255) | Khóa chính | Mã chức vụ |
| name | Varchar(255) | Không rỗng | Tên chức vụ |

Bảng 3.23 Mô tả bảng dữ liệu Position (Chức vụ)

* **Ràng buộc toàn vẹn cho bảng Postion**
* R1: Mã số chức vụ phải là duy nhất trong danh chức vụ.
* R2: Tên trạng chức vụ phải là duy nhất trong danh sách trạng chức vụ.
  + - 1. **Bảng dữ liệu AtPosition**

|  |  |  |  |
| --- | --- | --- | --- |
| **AtPosition** | | | |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Mô tả |
| employeeID | Varchar(7) | Khóa ngoại | Mã nhân viên |
| positionID | Varchar(255) | Khóa ngoại | Mã chức vụ |
| startDate | Date | Không nhỏ hơn ngày hiện tại | Ngày áp dụng |

Bảng 3.24 Mô tả bảng dữ liệu Position (Chức vụ)

* **Ràng buộc toàn vẹn cho bảng Postion**
* R1: Mã số chức vụ phải tồn tại trong danh chức vụ.
* R2: Mã nhân viên phải tồn tại trong danh sách nhân viên.
* R3: Ngày áp dụng phải lớn hơn hoặc bằng ngày hiện tại.
  + - 1. **Bảng dữ liệu Register**

|  |  |  |  |
| --- | --- | --- | --- |
| **Register** | | | |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Mô tả |
| employeeID | Varchar(7) | Khóa ngoại | Mã nhân viên |
| scheduleID | Varchar(255) | Khóa ngoại | Mã ca làm |
| registerID | Date | Không nhỏ hơn ngày hiện tại | Ngày làm |
| result | numeric(3) | Không rỗng | Kết quả xét duyệt |

Bảng 3.25 Mô tả bảng dữ liệu Register (Đăng kí ca làm)

* **Ràng buộc toàn vẹn cho bảng Register**
* R1: Mã số nhân viên phải tồn tại trong danh nhân viên.
* R2: Mã ca làm phải tồn tại trong danh sách ca làm.
* R3: Ngàylàm phải lớn hơn hoặc bằng ngày hiện tại.
  + - 1. **Bảng dữ liệu Schedule**

|  |  |  |  |
| --- | --- | --- | --- |
| **Schedule** | | | |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Mô tả |
| scheduleID | Varchar(7) | Khóa chính | Mã ca làm |
| startTime | Time | Không rỗng | Thời gian bắt đầu |
| endTime | Time | Không rỗng | Thời gian kết thúc |
| payRate | float | Không rỗng và lớn hơn 0 | Kết quả xét duyệt |

Bảng 3.26 Mô tả bảng dữ liệu Schedule (Ca làm)

* **Ràng buộc toàn vẹn cho bảng Schedule**
* R1: Mã số ca làm phải phân biệt với mã số ca làm khác trong danh sách Ca làm.
* R2: Thời gian kết thúc phải lớn hơn thời gian bắt đầu.
  + - 1. **Bảng dữ liệu TableStatusDetail**

|  |  |  |  |
| --- | --- | --- | --- |
| **Schedule** | | | |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Mô tả |
| dinnerTableID | integer | Khóa ngoại | Mã bàn ăn |
| tableStatusID | integer | Khóa ngoại | Mã trạng thái bàn |

Bảng 3.27 Mô tả bảng dữ liệu Schedule (Ca làm)

* **Ràng buộc toàn vẹn cho bảng Schedule**
* R1: Mã bàn ăn làm phải tồn tại trong danh sách Bàn ăn.
* R2: Mã trạng thái bàn làm phải tồn tại trong danh sách Trạng thái bàn.

Ngoài ra còn có các thuộc tính cờ được tạo ra ở mỗi bảng nhằm cung cấp thông tin cho người lập trình như sau:

|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
| Tên cột | Kiểu dữ liệu | Ràng buộc | Mô tả |
| createBy | Varchar(7) |  | Mã nhân viên tạo |
| createAt | Date time |  | Thời gian tạo |
| updateBy | Varchar(7) |  | Mã nhân viên sửa |
| updateAt | Date time |  | Thời gian sửa |
| isDelete | Boolean |  | Trạng thái xóa |
| deleteBy | Varchar(7) |  | Mã nhân viên xóa |
| deleteAt | Date time |  | Thời gian xóa |

Bảng 3.28 Bảng thuộc tính gắn cờ

## KẾT QUẢ NGHIÊN CỨU

### Cấu trúc của hệ thống Website

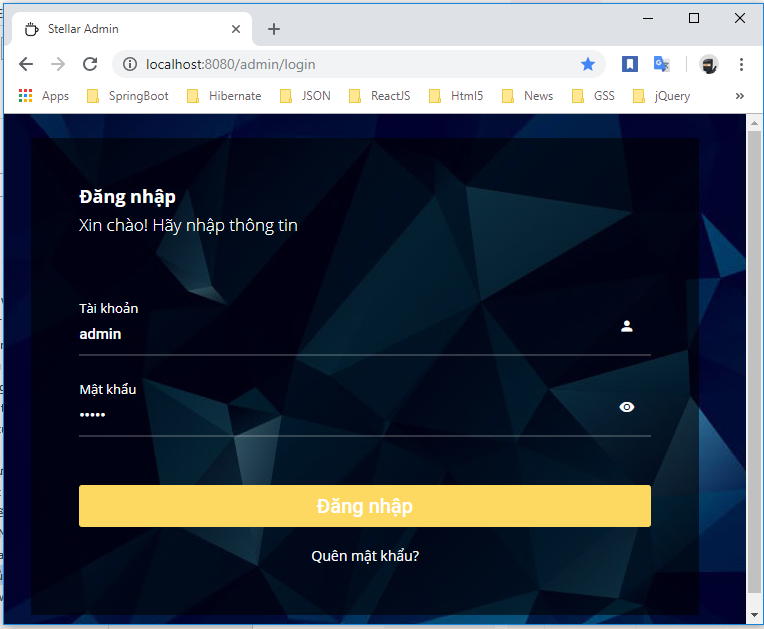
* + - 1. **Model và Controllers**

|  |  |
| --- | --- |
| * 1. ***Cấu trúc của Website (1)*** | * 1. ***Cấu trúc của Website (2)*** |

* + - 1. **View Administrator (Quản trị) và view User (Người dùng)**

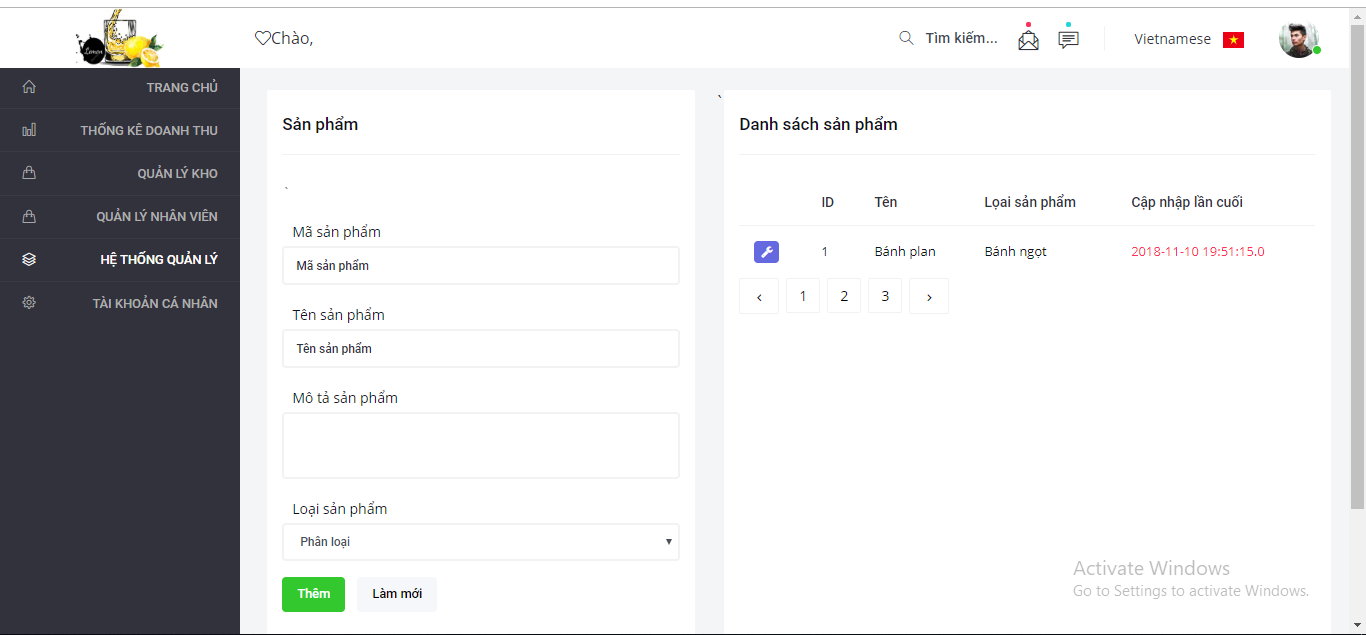
|  |  |
| --- | --- |
| * 1. ***Cấu trúc của Website (3)*** | * 1. ***Cấu trúc của Website (4)*** |

### Giao diện của hệ thống Website

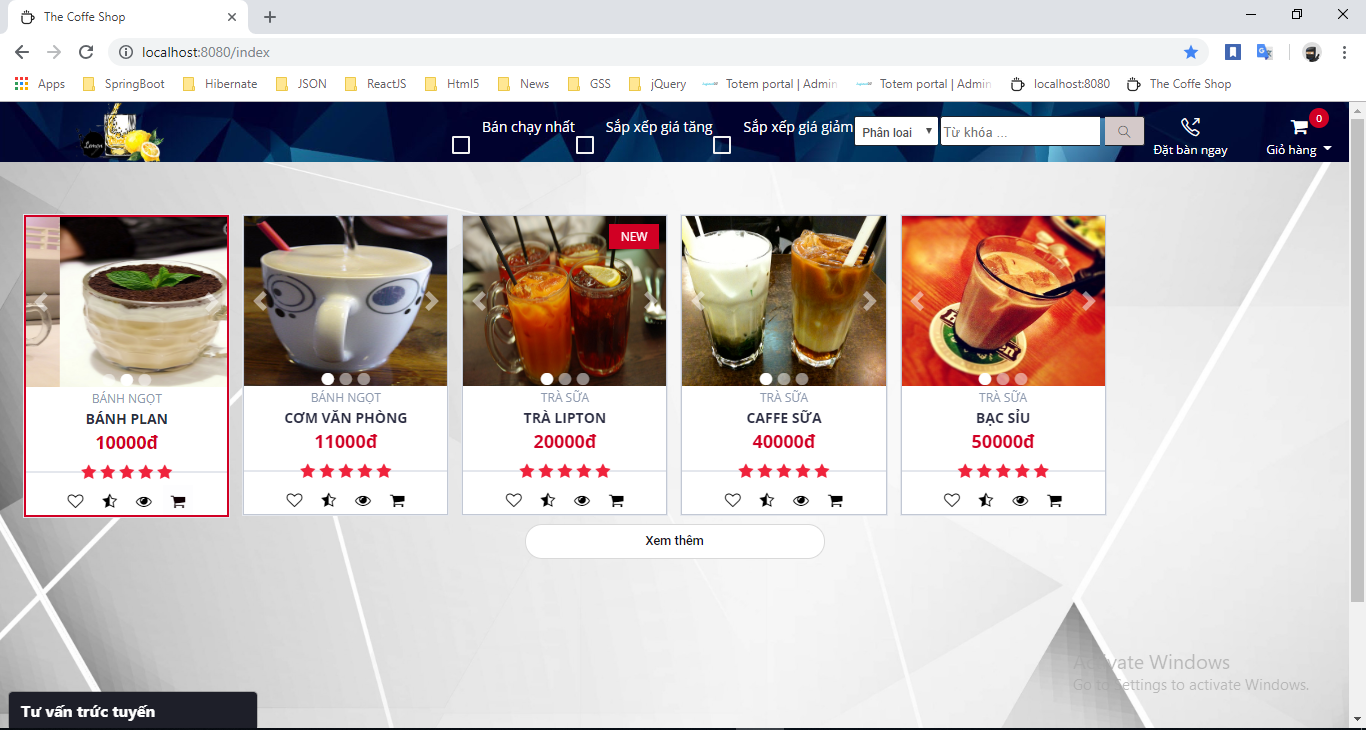
* + - 1. **Giao diện đăng nhập (Quản trị Website)**
  1. *Giao diện đăng nhập (Quản trị Website)*

Quản lý website nhập đầy đủ thông tin đăng nhập gồm email và mật khẩu, sau đó nhấn nút *Đăng nhập* để yêu cầu hệ thống xác thực tài khoản. hệ thống kiểm tra tài khoản trong cơ sở dữ liệu và thông báo kết quả xác thực cho quản lý website.

Trường hợp quản lý website nhập sai email hoặc mật khẩu, hệ thống Website sẽ thông báo đăng nhập không thành công và yêu cầu quản lý website nhập lại thông tin tài khoản.

* + - 1. **Giao diện quản lý sản phẩm**
  1. Giao diện quản lý sản phẩm

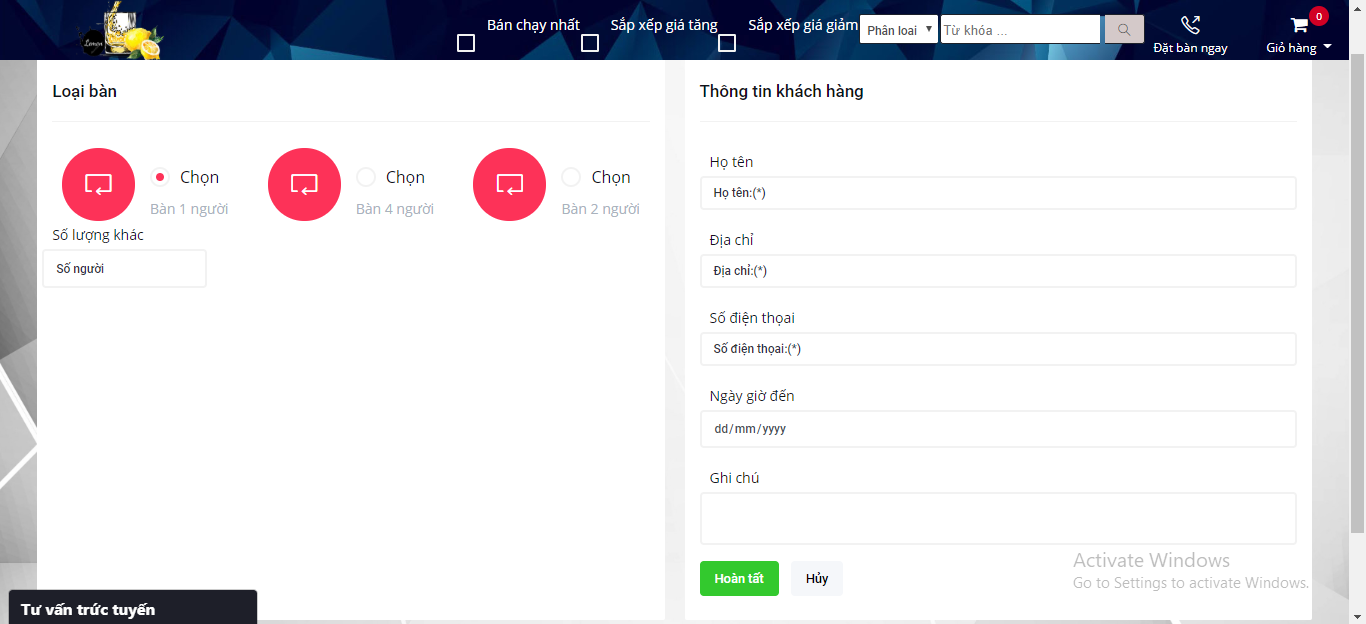
Ở trang này, quản lý website có thể xem danh sách các sản phẩm, thêm sản phẩm mới, sửa sản phẩm, xóa sản phẩm.

* + - 1. **Giao diện dành cho khách hàng**

Hình 3.7 *Giao diện xem sản phẩm dành cho khách hàng*

Ở trang này, khách hàng có thể xem sản phẩm.

* + - 1. **Giao diện dành cho khách hàng đặt bàn**



Hình 3.8 Giao diện đặt bàn

Ở trang này, khách hàng có thể điền thông tin và đặt bàn.

**CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN**

## KẾT LUẬN

Website đã đáp ứng nhu cầu thực tế và hiện thực hóa được nhiều tính năng cấp thiết giúp các coffe shop đơn giản hóa việc quản lý kinh doanh, cũng như quản lý nhân sự, giúp tiết kiệm nhiều loại chi phí. Ngoài ra website còn giúp khách hàng có thể thuận tiện trong việc gọi món, đặt bàn, tiết kiệm nhiều thời gian quý báu.. Trang web có giao diện thân thiện, dễ sử dụng, tính năng tương tác với người dùng cao.

## CÁC CHỨC NĂNG ĐÃ THỰC HIỆN ĐƯỢC

Xây dựng hệ thống Website người quản trị:

* Chức năng quản lý hóa đơn bán hàng, hóa đơn nhập nguyên liệu, hóa đơn xuất nguyên liệu
* Chức năng quản lý bàn ăn.
* Chức năng quản lý nhân viên.
* Chức năng quản lý đặt bàn.
* Chức năng quản lý nhà kho.
* Chức năng thống kê.

Xây dựng hệ thống Website cho khách hàng:

* Cho phép khách hàng xem thông tin sản phẩm
* Khách hàng có thể đặt bàn, đặt món trực tuyến.
* Trợ người dùng tìm kiếm thông tin sản phẩm.

## HƯỚNG PHÁT TRIỂN

Nhằm hỗ trợ tốt hơn cho coffe shop, hệ thống website có thể mở rộng thêm các chức năng cũng và nâng cấp về công nghệ:

* Chat trực tuyến với người dùng.
* Bình luận và xem bình luận về sản phẩm.
* Hỗ trợ thời gian thực.
* Phát triển thêm ứng dụng dành cho khách hàng sử dụng mobile.

Trong thời gian sắp tới hệ thống website sẽ được đưa vào thực tế, tôi sẽ tiếp tục phát triển hệ thống này theo hướng là một website service và một ứng dụng di dộng dành cho khách hàng và ứng dụng di động dành cho nội bộ nhân viên Coffe Shop.

# TÀI LIỆU THAM KHẢO

1. Website: http://spring.io/
2. Webstie: https://spring.io/guides/gs/spring-boot/
3. Website: https://docs.spring.io/spring-boot/docs/current-SNAPSHOT/reference/htmlsingle/
4. Website: <https://w3schools.com/>
5. Webstie: <https://getbootstrap.com/>
6. Website: <https://stackoverflow.com/>
7. Website: <http://docs.jboss.org/hibernate/orm/5.3/userguide/html_single/Hibernate_User_Guide.html#hql-select>
8. Website: <https://github.com/>