

Hà Nội 12/2017

Sinh viên thực hiện: **Nguyễn Thanh Tùng**

Mã sinh viên: 4003858

Lớp: 58PM1

Giảng viên hướng dẫn: **ThS.Đào Thị Ngọc Hân**

**XÂY DỰNG WEBSITE QUẢN LÝ**

**BÁN ĐỒ ĂN NHANH TRÊN**

**NGÔN NGỮ JAVA**

ĐỒ ÁN

**TỐT NGHIỆP ĐẠI HỌC**

NGÀNH CÔNG NGHỆ PHẦN MỀM

TRƯỜNG ĐẠI HỌC XÂY DỰNG

KHOA CÔNG NGHỆ THÔNG TIN

TÓM TẮT

NỘI DUNG ĐỒ ÁN TỐT NGHIỆP

1) Tên đề tài:

Xây dựng Website quản lý bán đồ ăn nhannh dựa trên ngôn ngữ Java.

2) Nội dung thuyết minh:

Chương I: Tổng quan đề tài.

Chương II: Các công nghệ áp dụng.

Chương III: Tìm hiểu công nghệ.

Chương IV: Phân tích thiết kế đề tài.

Chương V: Xây dựng chức năng.

Chương VI: Kết luận.

3) Cán bộ hướng dẫn chính:

THS. ĐÀO THỊ NGỌC HÂN.

4) Ngày giao nhiệm vụ thiết kế: 11/09/2017

5) Ngày hoàn thành nhiệm vụ thiết kế: 20/12/2017

6) Keyword: fastfood mangement, Java, spring MVC,hibernate

PHIẾU GIAO NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

1. Thông tin về sinh viên:

Họ và tên sinh viên: Nguyễn Thanh Tùng

Điện thoại liên lạc : 01233378716 Email: nguyenthanhtung31995@gmail.com

Lớp: 58pm1 Hệ đào tạo: Chính quy Năm tốt nghiệp: 2018

Đồ án tốt nghiệp được thực hiện tại: Hà Nội

Thời gian làm ĐATN: Từ ngày 11/ 09/ 2017 đến 23/ 12/ 2017

2. Mục đích nội dung của ĐATN:

Xây dựng Website quản lý bán đồ ăn nhanh trên ngôn ngữ java.

3. Các nhiệm vụ cụ thể của ĐATN

Xây dựng Website quản lý bán đồ ăn nhanh trên ngôn ngữ java.

- Framework: Spring MVC, hibernate,bootstrap.

- Các chức năng chính: quản lý nhân viên, thực đơn, hóa đơn và báo cáo thống kê.

4. Lời cam đoan của sinh viên:

Tôi – *Nguyễn Thanh Tùng*- cam kết ĐATN là công trình nghiên cứu của bản thân tôi dưới sự hướng dẫn của *Ths.Đào Thị Ngọc Hân* .

Các kết quả nêu trong ĐATN là trung thực, không phải là sao chép toàn văn của bất kỳ công trình nào khác.

|  |  |
| --- | --- |
|  | *Hà Nội, ngày 22 tháng 12 năm 2017*  Tác giả ĐATN  *Nguyễn Thanh Tùng* |

5. Xác nhận của giảng viên hướng dẫn về mức độ hoàn thành ĐATN và cho phép bảo vệ:

|  |  |
| --- | --- |
|  | *Hà Nội, ngày 22 tháng 12 năm2017*  Giảng viên hướng dẫn  *Ths. Đào Thị Ngọc Hân* |

Lời cảm ơn

Em xin chân thành cảm ơn các thầy cô trong trường Đại học Xây Dựng nói chung, các thầy cô trong khoa Công nghệ thông tin, bộ môn Công nghệ phần mềm nói riêng vì đã tận tình giảng dạy, dẫn dắt và truyền đạt những kinh nghiệm, kiến thức quý báu, bổ ích trong xuốt thời gian học tập tại trường.

Em xin được gửi lời cảm ơn đến cô giáo Đào Thị Ngọc Hân giảng viên bộ môn Công nghệ phần mềm, khoa Công nghệ thông tin, trường Đại học Xây dựng vì đã hết lòng giúp đỡ, hướng dẫn và chỉ bảo trong quá trình em làm đồ án tốt nghiệp.

Em cũng xin cảm ơn đến các anh chị hướng dẫn của Fresher Academy FPT SOFTWARE và các bạn đồng nghiệp của lớp Fresher Java HN 07 vì đã giúp đỡ rất nhiều trong quá trình học tập, tìm hiểu framework tại công ty.

*Em xin chân thành cảm ơn!*

Mục lục

[CHƯƠNG I: TỔNG QUAN ĐỀ TÀI. 10](#_Toc502229626)

[1) Giới thiệu đề tài: 10](#_Toc502229627)

[2) Đối tượng và phạm vi nghiên cứu: 10](#_Toc502229628)

[3) Mục đích đề tài: 10](#_Toc502229629)

[Chương II: CÁC CÔNG NGHỆ ÁP DỤNG 11](#_Toc502229630)

[1) Ngôn ngữ lập trình Java 11](#_Toc502229631)

[1.1)Lịch sử phát triển 11](#_Toc502229632)

[1.2) Vài nét về ngôn ngữ Java: 11](#_Toc502229633)

[1.3) Các đặc trưng của Java 11](#_Toc502229634)

[1.4)Các phiên bản Java đã phát hành: 11](#_Toc502229635)

[2)Framework. 12](#_Toc502229636)

[2.1) Khái niệm chung. 12](#_Toc502229637)

[2.2) Spring 12](#_Toc502229638)

[2.3) IOC(Inversion Of Control) 14](#_Toc502229639)

[2.4)DI(Dependecy Ịnection). 14](#_Toc502229640)

[2.5) Spring MVC. 15](#_Toc502229641)

[2.6) Hibernate. 16](#_Toc502229642)

[2.7) ORM framework. 17](#_Toc502229643)

[2.8) Bootstrap. 18](#_Toc502229644)

[3) Các công nghệ khác. 18](#_Toc502229645)

[3.1)JPA. 19](#_Toc502229646)

[3.2) Maven. 19](#_Toc502229647)

[3.3) Tomcat. 20](#_Toc502229648)

[3.4) Eclipse. 20](#_Toc502229649)

[3.5) STS (Spring Tool Suite). 21](#_Toc502229650)

[3.6) JavaScript 21](#_Toc502229651)

[3.7)jQuery. 22](#_Toc502229652)

[CHƯƠNG III: TÌM HIỂU CÔNG NGHỆ 24](#_Toc502229653)

[1) ORM 24](#_Toc502229654)

[2) Hibernate. 24](#_Toc502229655)

[2.1)SessionFactor 24](#_Toc502229656)

[2.2)Session 24](#_Toc502229657)

[2.3)Transaction 25](#_Toc502229658)

[2.4)ConnectionProvider 25](#_Toc502229659)

[2.5)TransactionFactory 25](#_Toc502229660)

[3)Config Spring MVC và JPA. 25](#_Toc502229661)

[1.1)Tạo project. 26](#_Toc502229662)

[1.2)Cấu hình file pom.xml. 27](#_Toc502229663)

[1.3) Cấu hình file web.xml. 28](#_Toc502229664)

[1.4) Tạo package chuẩn mvc. 29](#_Toc502229665)

[1.5) Cấu hình file mvc-config.xml 30](#_Toc502229666)

[1.6) Cấu hình file beans.xml. 32](#_Toc502229667)

[1.7) Viết controller. 32](#_Toc502229668)

[1.8) Chạy ứng dụng trên server Tomcat. 33](#_Toc502229669)

[Chương IV: PHÂN TÍCH - THIẾT KẾ ĐỀ TÀI 36](#_Toc502229670)

[1) Xác định phạm vi, kế hoạch thực hiện. 36](#_Toc502229671)

[2) Xác định yêu cầu người dùng. 36](#_Toc502229672)

[3) Phân tích hệ thống. 36](#_Toc502229673)

[3.1) Đối tượng sử dụng hệ thống: 36](#_Toc502229674)

[3.3) Chức năng của người quản lý: 37](#_Toc502229675)

[4) Sơ đồ hệ thống. 37](#_Toc502229676)

[4.1) Sơ đồ usecase tổng quát. 38](#_Toc502229677)

[4.2)Đăng nhập. 38](#_Toc502229678)

[4.3) Cập nhật hóa đơn 40](#_Toc502229679)

[5) Thiết kế cơ sở dữ liệu. 41](#_Toc502229680)

[5.1) Bảng dữ liệu. 41](#_Toc502229681)

[5.2) Sơ đồ cơ sở dữ liệu quan hệ. 45](#_Toc502229682)

[CHƯƠNG V: XÂY DỰNG CHƯƠNG TRÌNH. 46](#_Toc502229683)

[1) Giao diện người dùng. 46](#_Toc502229684)

[1.1) Màn hình đăng nhập. 46](#_Toc502229685)

[1.2)Màn hình danh sách hóa đơn. 46](#_Toc502229686)

[1.3) Màn hình thêm hóa đơn mới 47](#_Toc502229687)

[1.4) Màn hình chi tiết hóa đơn. 47](#_Toc502229688)

[1.5) Màn hình danh sách nhân viên. 48](#_Toc502229689)

[1.6) Màn hình thêm nhân viên mới. 48](#_Toc502229690)

[1.7) Màn hình thông tin nhân viên. 49](#_Toc502229691)

[CHƯƠNG VI: KẾT LUẬN VÀ HƯỚNG PHÁT TRIÊN 50](#_Toc502229692)

[1) Kết quả đạt được. 50](#_Toc502229693)

[2) Các khó khăn trong quá trình thực hiện đồ án. 50](#_Toc502229694)

[3) Hạn chế: 50](#_Toc502229695)

[4) Định hướng phát triển. 51](#_Toc502229696)

[5) Công cụ mã nguồn. 51](#_Toc502229697)

[4.1) Các công cụ được sử dụng: 51](#_Toc502229698)

[4.2) Mã nguồn, thư viện: 51](#_Toc502229699)

[6) Tài liệu tham khảo. 51](#_Toc502229700)

Danh mục các hình

[Hình 1: Nguyên lý hoạt động của ORM 24](#_Toc502229701)

[Hình 2: Sơ đồ usecase mức tổng quát. 38](#_Toc502229702)

[Hình 3: Activity đăng nhập 39](#_Toc502229703)

[Hình 4: Activity đăng xuất 39](#_Toc502229704)

[Hình 5 Activity thêm hóa đơn. 40](#_Toc502229705)

[Hình 6: Activity sửa hóa đơn. 41](#_Toc502229706)

[Hình 7: Bảng nhân viên. 42](#_Toc502229707)

[Hình 8: Bảng thực đơn. 42](#_Toc502229708)

[Hình 9: Bảng phân quyền. 43](#_Toc502229709)

[Hình 10: Bảng hóa đơn hàng ngày. 43](#_Toc502229710)

[Hình 11: Bảng chi tiết hóa đơn hàng ngày. 44](#_Toc502229711)

[Hình 12: Bảng hóa đơn đã lưu 44](#_Toc502229712)

[Hình 13: Bảng chi tiết hóa đơn đã lưu. 45](#_Toc502229713)

[Hình 14: Sơ đồ cơ sở dữ liệu quan hệ. 45](#_Toc502229714)

[Hình 15:Màn hình đăng nhập. 46](#_Toc502229715)

[Hình 16:Màn hình danh sách hóa đơn. 46](#_Toc502229716)

[Hình 17 Màn hình thêm hóa đơn mới. 47](#_Toc502229717)

[Hình 18: Màn hình chi tiết hóa đơn. 48](#_Toc502229718)

[Hình 19: Màn hình danh sách nhân viên. 48](#_Toc502229719)

[Hình 20: Màn hình thêm nhân viên mới. 49](#_Toc502229720)

[Hình 21: Màn hình sửa nhân viên. 49](#_Toc502229721)

# **CHƯƠNG I: TỔNG QUAN ĐỀ TÀI.**

## 1) Giới thiệu đề tài:

Cùng với sự phát triển vượt bậc của công nghệ hiện nay, các phần mềm xây dựng dựa trên công nghệ web trở nên phổ biến trên toàn thế giới. Các ngôn ngữ lập trình web cũng ngày càng phong phú và đa dạng: java, php, .net,python… Các framework cho từng ngôn ngữ cũng không ngừng phát triển, cung cấp nhiều tiện ích và hỗ trợ cho các lập trình viên.

Sau khi cân nhắc về các lợi ích của nhiều ngôn ngữ lập trình, em đã quyết định áp dụng Java để làm đề tài tốt nghiệp.

Trang web “quản lý bán đồ ăn nhanh” cung cấp cho người dùng các tính năng hữu ích cho việc quản lý cửa hàng đồ ăn nhanh như: quản lý thực đơn, nhân viên, hóa đơn, báo cáo thống kê…

## **2) Đối tượng và phạm vi nghiên cứu:**

+ Đối tượng:

- Java WEB.

- SQL SERVER

- Framework: bootstrap, hibernate, spring MVC, jsp…

+ Phạm vi:

- Nghiên cứu, nắm vững, áp dụng các kiến thức cơ bản của Java.

- Kết hợp các framework để thiết kế, xây dựng trang web.

## **3) Mục đích đề tài:**

Xây dựng website quản lý bán đồ ăn nhanh, áp dụng Javaweb và các framework.

# **Chương II: CÁC CÔNG NGHỆ ÁP DỤNG**

## 1) Ngôn ngữ lập trình Java

### 1.1)Lịch sử phát triển

Năm 1990, Sun MicroSystems thực hiện dự án Green nhằm phát triển phần mềm trong các thiết bị dân dụng. James Gosling, chuyên gia lập trình đã tạo ra một ngôn ngữ lập trình mới có tên là Oak. Ngôn ngữ này có cú pháp gần giống như C++ nhưng bỏ qua các tính năng nguy hiểm của C++ như truy cập trực tiếp tài nguyên hệ thống, con trỏ, định nghĩa chồng các tác tử…

Khi ngôn ngữ Oak trưởng thành, WWW cũng đang vào thời kỳ phát triển mạnh mẽ, Sun cho rằng đây là một ngôn ngữ thích hợp cho Internet. Năm 1995, Oak đổi tên thành Java và sau đó đến 1996 Java đã được xem như một chuẩn công nghiệp cho Internet.

### 1.2) Vài nét về ngôn ngữ Java:

Java là ngôn ngữ lập trình hướng đối tượng do Sun Microsystem phát triển.

Java được viết dựa trên ngôn ngữ C, với dự định thay thế C++.

Java chạy trên môi trường máy ảo Java Visual Machine (JVM) , vì vậy nó có thể chạy được trên bất kì nền tảng nào có cài đặt JVM.

Phương châm của Java là “Viết một lần, chạy mọi nơi” (write once, run anywhere).

Java có lượng thành viên đông đảo, sẵn sàng hỗ trợ cho người sử dụng.

### 1.3) Các đặc trưng của Java

+ Hướng đối tượng: Java là ngôn ngữ hướng đối tượng hoàn toàn, hướng đối tượng trong Java không có tính đa kế thừa, mà sử dụng interface để hỗ trợ đa kế thừa.

+ Nền tảng độc lập: chương trình Java có thể chạy được trên bất kì máy tính nào có cài đặt JVM, không quan tâm đến hệ điều hành của máy.

+ Đa nhiệm – đa luồng (mutiltasking-mutilthreading): Java cho phép nhiều luồng chạy song song tại cùng thời điểm và tương tác với nhau.

### 1.4)Các phiên bản Java đã phát hành:

JDK 1.0 (23 tháng 01, 1996).

JDK 1.1 (19 tháng 2 năm 1997).

JDK 1.1.5 (Pumpkin) 03 tháng 12 năm 1997.

JDK 1.1.6 (Abigail) 24 tháng 4 năm 1998.

JDK 1.1.7 (Brutus) 28 tháng 9 năm 1998.

JDK 1.1.8 (Chelsea) 08 tháng 4 năm 1999.

J2SE 1.2 (Playground) 08 tháng 12 năm 1998.

J2SE 1.2.1 (không có) 30 tháng 3 năm 1999.

J2SE 1.2.2 (Cricket) 08 tháng 7 năm 1999.

J2SE 1.3 (Kestrel) 08 tháng 5 năm 2000.

J2SE 1.3.1 (Ladybird) 17 tháng 5 năm 2001.

J2SE 1.4.0 (Merlin) 06 tháng 02, 2002.

J2SE 1.4.1 (Hopper) 16 tháng 9 năm 2002.

J2SE 1.4.2 (Mantis) 26 tháng 6 năm 2003.

J2SE 5 (1.5.0) (Tiger) 30 tháng 9 năm 2004.

Java SE 6 (Mustang), 11 tháng 12 năm 2006

Java SE 7 (Dolphin), 28 tháng 7 .năm 2011.

JDK 8, 18 tháng 3 năm 2014.

JDK 9, 21 tháng 9 năm 2017, là phiên bản mới nhất tại thời điểm hiện tại.

## 2)Framework.

### 2.1) Khái niệm chung.

Framework là các “khung”, bao gồm các thư viện và các cấu trúc được thiết kế sẵn để hỗ trợ cho việc lập trình, giúp quản lý và lập trình trở nên dễ dàng hơn.

### 2.2) Spring



Spring là 1 framework mã nguồn mở cho Java, tích hợp nhiều công nghệ như logging, ORM framework, JDK timer…

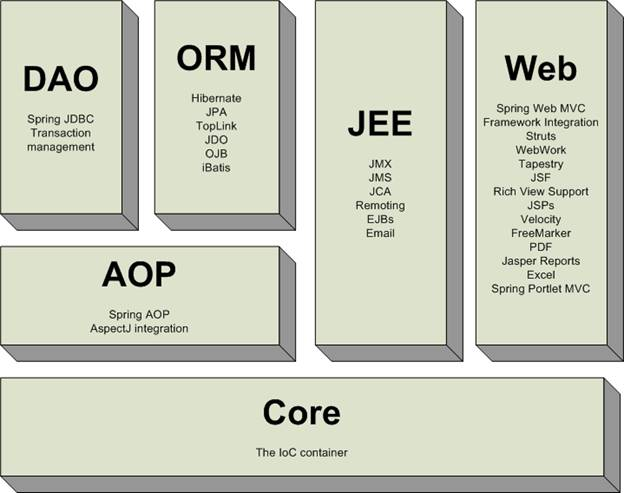
Tiêu chí của Spring là gọn nhẹ, dễ kiểm thử, module hóa và tái sử dụng.

Spring có thể dùng để phát triển bất kì ứng dụng Java nào, từ app, mobile, đến web, và hỗ trợ web rất tốt.

Spring sử dụng cơ chế IOC và DI để cải thiện hiệu năng và việc quản lý tài nguyên.

Phần đầu tiên của Spring Framework ban đầu nguyên được Rod Johnson viết vào năm 2000.

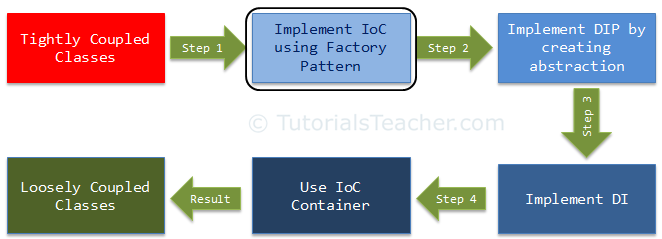
Cấu trúc Spring framework:



**- IoC Container**: Đây là phần quan trọng nhất và cũng là phần cơ bản, nền tảng của Spring. Nó giữ vai trò về cấu hình và quản lý vòng đời (Lifecycle) của các đối tượng java.

- **DAO, ORM, AOP, WEB:** Các module này là tool hoặc là framework có sẵn được tích hợp vào Spring.

### 2.3) IOC(Inversion Of Control)



Thông thường, người lập trình sẽ là người sinh ra các Object (Student s = new Student()), và tự quản lý chúng.

IOC, đảo ngược điều khiển, là nguyên lý đi ngược lại với việc đó. Việc tạo và quản lý các Object sẽ được container hoặc framework quản lý. Framework sẽ quản lý các luồng chạy của chương trình, gọi đến các hàm của người dùng định nghĩa.

Áp dụng IOC mang lại những công dụng như:

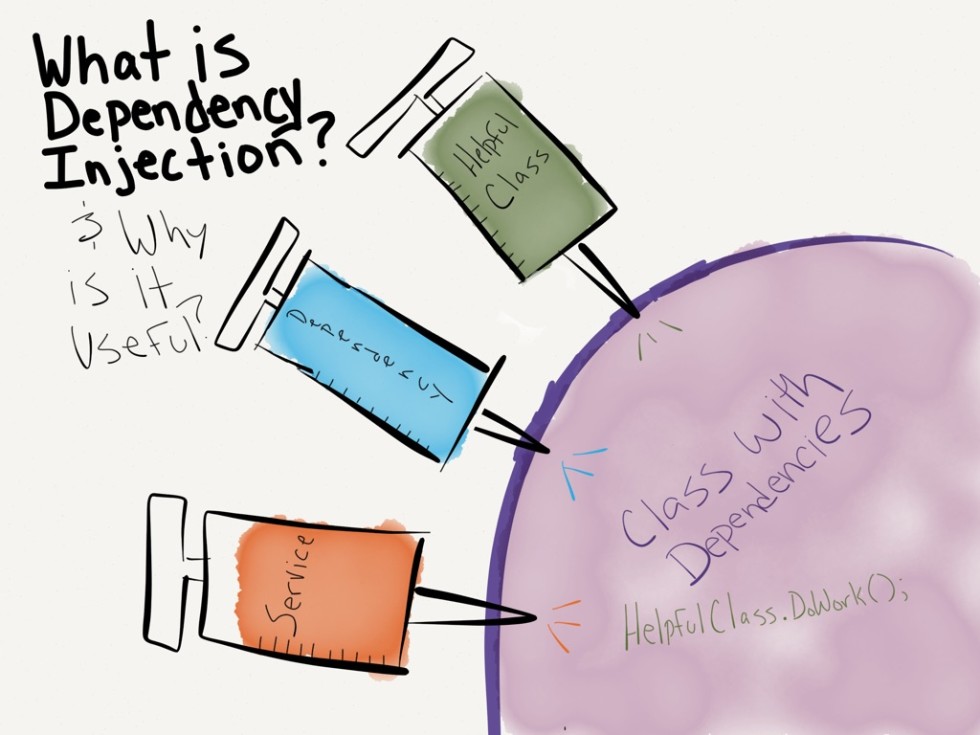
- Giảm sự phụ thuộc giữa phần cài đặt và thực thi.

- Giúp chuyển đổi giữa các cài đặt khác nhau trở nên dễ dàng hơn.

- Dễ dàng chia nhỏ chương trình.

- Dễ dàng kiểm tra từng thành phần.

### 2.4)DI(Dependecy Ịnection).



DI là 1 mẫu thiết kế để thể hiện cho IOC, dảo ngược điều khiển qua việc thiết lập các đối tượng phụ thuộc.

Thường khi các Object có quan hệ sở hữu (has-a), ta sẽ phải khởi tạo các thành phần bên trong, rồi mới đưa vào các Object.

Với DI, người dùng sẽ chỉ khai báo các đối tượng bên trong mà không khởi tạo chúng.IOC container sẽ đảm nhiệm việc khởi tạo và quản lý. DI sẽ thực hiện viêc truyền đối tượng đó vào qua hàm khởi tạo hoặc qua setter.

### 2.5) Spring MVC.



Spring MVC là 1 thành phần của Spring, thiết kế cho lập trình web với cấu trúc MVC: Model, View và Controller.

Model sẽ chứa các class entity.

View sẽ chứa các html, jsp.

Controller sẽ chứa các hàm xử lý request.

Ngoài ra, còn có Controller Handler, xử lý việc mapping đường dẫn.

### 2.6) Hibernate.



Hibernate là 1 ORM framework. Hibernate sẽ tiếp quản việc thao tác tầng persistence. Lập trình viên sẽ không cần bận tâm vào tầng persistence nữa, mà tập trung vào các tầng trên. Hibernate giúp lưu trữ, truy vấn các dữ liệu quan hệ nhanh chóng và mạnh mẽ.

Hibernate hỗ trợ các quan hệ một-một, một-nhiều, nhiều-một, và nhiều-nhiều.

Hibernate còn hỗ trợ việc tự sinh database dựa trên các class entities đã khai báo, nhưng cần cẩn thận bởi việc kiểm soát nó không hề dễ dàng cho những người không có kiến thức chuyên sâu.

Việc config Hibernate để chạy một cách hoàn hảo cũng không hề dễ dàng, cần có kinh nghiệm và nắm chắc kiến thức về nó.

Các lợi ích của hibernate:

- Mã nguồn mở và nhẹ.

- Hiệu suất nhanh.

- Truy vấn cơ sở dữ liệu độc lập.

- Tạo bảng tự động.

- Đơn giản hóa lệnh join phức tạp.

- Cung cấp thống kê truy vấn và trạng thái cơ sở dữ liệu

### 2.7) ORM framework.



Khi chúng ta làm việc với các hệ thống hướng đối tượng, sẽ gặp phải vấn đề không tương thích giữa kiểu đối tượng và cơ sở dữ liệu quan hệ. Các hệ quản trị cơ sở dữ liệu quan hệ (RDBMSs) biểu diễn dữ liệu trong định dạng bảng, trong khi đó các ngôn ngữ lập trình hướng đối tượng như Java, C#, C++ biểu diễn dữ liệu dưới dạng đồ thị liên kết các đối tượng. ORM là giải pháp để giải quyết vấn đề không tương thích gây cản trở trên.

ORM(Object Relational Mapping) sẽ thao tác với các bảng dữ liệu quan hệ trong database thông qua các đối tượng. Lập trình viên chỉ cần sử dụng các đối tượng, không cần quan tâm đến database mình đang sử dụng là gì.

Ưu điểm:

-Tính độc lập: Có thể sử dụng với nhiều RDBMS khác nhau. Khi thay đổi hệ quản trị cơ sở dữ liệu chỉ cần thay đổi driver tương tác mà không cần phải thay đổi bất kỳ một dòng code nào.

- Tính đơn giản, dễ sử dụng: So với cách thông thường phải sử dụng SQL để truy vấn và xử lý dữ liệu trực tiếp với csdl, ORM cung cấp các API đơn giản và rất dễ sử dụng.

- Tiết kiệm thời gian lập trình, source code ngắn gọn, dễ hiểu, dễ bảo trì

Nhược điểm:

- Do ORM tự động gen ra các câu lệnh SQL nên sẽ khó có thể tác động vào để tối ưu câu lệnh.

- Việc quản lý session khá phức tạp.

### 2.8) Bootstrap.



Bootstrap là framework mã nguồn mở dùng để thiết kế giao diện frontend cho các trang web. Nó bao gồm các thư viện css và javascript được viết sẵn để làm cho trang web được đẹp hơn và thống nhất về thiết kế. Người dùng chỉ cần thêm thư viện vào trang của mình và sử dụng các class do bootstrap cung cấp.

## 3) Các công nghệ khác.

### 3.1)JPA.



Jpa(Java Persistence API) cung cấp các API hỗ trợ người lập trình tạo ra cơ sở dữ liệu từ các đối tượng hoặc ngược lại 1 cách dễ dàng.

Jpa cung cấp sẵn các phương thức CRUD (create, read, update, delete), giúp người sử dụng hạn chế việc viết sql.

Ngoài ra Jpa còn hỗ trợ truy vấn vào cơ sở dữ liệu quan hệ 1 cách hiệu quả.

Vì Jpa chỉ là API (interface), nên nó cần đi cùng với các framework ORM cài đặt các API đó.

Hibernate là ORM framework được áp dụng trong đồ án này.

Các khái niệm trong JPA:

- Entity: là đối tượng tương ứng với 1 bảng trong cơ sở dữ liệu, là class khai báo đối tượng trong Java.

- EntityManager: interface, cung cấp các API cho phép tương tác với cơ sở dữ liệu.

- EntityManagerFactory: tạo ra và cung cấp EntityManager.

### 3.2) Maven.



Các thư viện trong Java sẽ được đóng gói lại dưới dạng các file .jar. Khi muốn sử dụng thư viện nào, lập trình viên thường sẽ phải tải từng thư viện mình cần và thêm vào project của mình.

Maven là công cụ quản lý và thiết lập tự động các thư viện Java trong 1 project. Maven phục vụ mục đích tương tự như Apache Ant, nhưng nó dựa trên khái niệm khác và cách hoạt động khác. Maven được quản lý bởi Apache Software Foundation, nó là một phần của dự án Jakarta Project.

Maven dùng khái niệm Project Object Model (POM) để mô tả việc build project, các thành phần phụ thuộc và các module. Nó định nghĩa trước các target cho việc khai báo task, trình biên dịch, đóng gói và thứ tự hoạt động để mọi việc diến ra tốt nhất.

Maven tải các thư viện, plug-in từ 1 hoặc nhiều repositories. Maven cung cấp việc hỗ trợ tải file từ repository trung tâm của Maven và các repositories khác. Cũng có thể upload các sản phẩm lên repository sau khi đã build xong.

Maven được xây dựng dựa trên kiến trúc plugin - cho phép sử dụng bất kỳ ứng dụng điều khiển nào thông qua đầu vào tiêu chuẩn. Về mặt lý thuyết, điều này cho phép bất kỳ ai cũng có thể viết các plugin để giao tiếp với các build tool (trình biên dịch, unit test tool, etc) cho bất kỳ ngôn ngữ nào khác. Nhưng trong thực tế, hỗ trợ và sử dụng cho các ngôn ngữ khác đã bị thu hẹp. Hiện tại thì plugin cho.Net đã có và vẫn được duy trì, và một plugin cho C/C++ cũng được duy trì cho Maven.

Các ứng dụng Web và Spring rất nên dùng Maven, bởi nó đảm bảo việc đồng bộ các thư viện, tránh thiếu sót, sai phiên bản.

Maven cũng giúp cho việc trong tập thể trở nên dễ dàng hơn. Các thành viên trong nhóm chỉ cần tải project về máy, import vào và chạy Maven để tự tải thư viện.

Nhược điểm của Maven là cần có mạng mỗi khi bạn thay đổi phiên bản của các thư viện di kèm, và tốc độ tải kém hơn bình thường, vì Maven vừa tải về vừa kiểm tra. Khi Maven đang update, nếu ta dừng việc update lại, sẽ rất dễ sinh ra lỗi, buộc ta phải xóa các thư viện bị tải lỗi bằng tay trước khi có thẻ chạy lại Maven.

### 3.3) Tomcat.



Apache Tomcat là một Java Servlet được phát triển bởi Apache Software Foundation (ASF). Tomcat thi hành các ứng dụng Java Servlet và JavaServer Pages (JSP) từ Sun Microsystems, và cung cấp một máy chủ HTTP cho ngôn ngữ Java thuần túy để thực thi các chương trình lệnh viết bằng ngôn ngữ Java.

Tomcat không nên hiểu nhầm với các máy chủ HTTP Apache - dùng để thực thi các câu lệnh viết bằng ngôn ngữ C trên máy chủ HTTP; có 2 máy chủ web được kết nối với nhau. Apache Tomcat cung cấp các công cụ cho việc cấu hình và quản lý, nhưng cũng có thể được cấu hình bởi việc soạn thảo các file cấu hình viết bằng XML.

### 3.4) Eclipse.



Eclipse là phần mềm miễn phí, được các nhà phát triển sử dụng để xây dựng những ứng dụng J2EE,sử dụng Eclipse nhà phát triển có thể tích hợp với nhiều công cụ hỗ trợ khác để có được một bộ côngcụ hòan chỉnh mà không cần dùng đến phần mềm riêng nào khác.Eclipse SDK bao gồm 3 phần chính:Platform, Java Development Toolkit (JDT), Plug-in Development Environment (PDE). Với JDT,Eclipse được xem như là một môi trường hỗ trợ phát triển Java mạnh mẽ. PDE hỗ trợ việc mở rộngEclipse, tích hợp các Plug-in vào Eclipse Platform. Eclipse Platform là nền tảng của toàn bộ phần mềmEclipse, mục đích của nó là cung cấp những dịch vụ cần thiết cho việc tích hợp những bộ công cụ pháttriển phần mếm khách dưới dạng Plug-in, bản thân JDT cũng có thể được coi như là một Plug-in làmcho Eclipse như là một Java IDE (Integrated Development Enviroment).

### 3.5) STS (Spring Tool Suite).



Spring Tool Suite (STS) là một IDE mở rộng cho Eclipse, giúp hỗ trợ cho việc lập trình Spring, có thể cài đặt riêng biệt hoặc tích hợp vào Eclipse như 1 plugin.

### 3.6) JavaScript

JavaScript là một ngôn ngữ lập trình đa nền tảng (cross-platform), ngôn ngữ lập trình kịch bản, hướng đối tượng. JavaScript là một ngôn ngữ nhỏ và nhẹ (small and lightweight). Khi nằm bên trong một môi trường (host environment), JavaScript có thể kết nối tới các object của môi trường đó và cung cấp các cách quản lý chúng (object).

JavaScript chứa các thư viện tiêu chuẩn cho các object, ví dụ như: Array, Date, và Math, và các yếu tố cốt lõi của ngôn ngữ lập trình như: toán tử (operators), cấu trúc điều khiển (control structures), và câu lệnh. JavaScript có thể được mở rộng cho nhiều mục đích bằng việc bổ sung thêm các object; ví dụ:

- Client-side JavaScript - JavaScript phía máy khách, JavaScript được mở rộng bằng cách cung cấp các object để quản lý trình duyệt và Document Object Model (DOM) của nó. Ví dụ, phần mở rộng phía máy khách cho phép một ứng dụng tác động tới các yếu tố trên một trang HTML và phản hồi giống các tác động của người dùng như click chuột, nhập form, và chuyển trang.

- Server-side JavaScript - JavaScript phía Server, JavaScript được mở rộng bằng cách cung cấp thêm các đối tượng cần thiết để để chạy JavaScript trên máy chủ. Ví dụ, phần mở rộng phía server này cho phép ứng dụng kết nối với cơ sở dữ liệu (database), cung cấp thông tin một cách liên tục từ một yêu cầu tới phần khác của ứng dụng, hoặc thực hiện thao tác với các tập tin trên máy chủ.

So sánh javascript và java

|  |  |
| --- | --- |
| **Java** | **Java** |
| Hướng đối tượng (Object-oriented). Không phân biệt giữa kiểu (type) của các đối tượng (object). Tính kế thừa thông qua cơ chế nguyên mẫu (prototype), và các thuộc tính (property) cũng như phương thức có thể thêm vào bất cứ đối tượng nào một cách năng động. | Class-based (nền tảng lớp.).Đối tượng được thành các lớp với tất cả kế thừa thông qua hệ thống phân cấp lớp. Các lớp không thể thêm vào các thuộc tính và phương thức mới một cách năng động. |
| Không khai báo kiểu dữ liệu cho biến (dynamic typing). | Phải khai báo kiểu dữ liệu cho biến (static typing). |
| Không thể tự động ghi vào ổ đĩa cứng. | Có thể tự động ghi dữ liệu vào đĩa cứng. |

### 3.7)jQuery.



jQuery là một thư viện được tạo ra từ JavaScript, Ajax hay nói cách khác đó là một bước phát triển mới của JavaScript. Jquery làm cho chúng ta dễ dàng thao tác hơn và đỡ mất thời gian hơn đối với khi thao tác trên Ajax cũng như JavaScript.

jQuery phần mềm mã nguồn mở, kép cấp phép bởi MIT Giấy phép GNU General Public License, phiên bản 2.

jQuery hỗ trợ di chuyển document, chọn ra các element của DOM, tạo ra hoạt hình, xử lý Sự kiện, và phát triển ứng dụng Ajax. jQuery cũng cung cấp khả năng cho các nhà phát triển để tạo ra plug-in trên đầu trang của thư viện JavaScript.

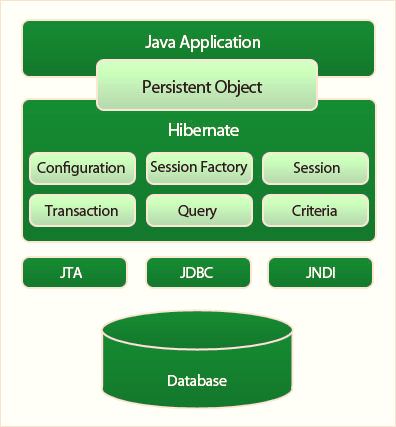
# CHƯƠNG III: TÌM HIỂU CÔNG NGHỆ

## 1) ORM



Hình : Nguyên lý hoạt động của ORM

## 2) Hibernate.



Hình 2: Các thành phần của Hibernate.

### 2.1)SessionFactor

SessionFactory : Sinh ra và cung cấp session khi người dùng yêu cầu.

### 2.2)Session

Session là kết nối vật lý đến database.

Session cung cấp các API để liên kết giữa ứng dụng và database, chỉ tồn tại trong thời gian ngắn và sử dụng JDBC.

Session sinh ra Transaction, Query and Criteria , cung cấp API hỗ trợ CRUD.

### 2.3)Transaction

Transaction là thành phần trực tiếp làm việc với database.

### 2.4)ConnectionProvider

ConnectionProvider là nhà máy tạo ra và cung cấp các kết nối JDBC.

### 2.5)TransactionFactory

TransactionFactory là nhà máy tạo ra và cung cấp các Transaction.

## 3)Config Spring MVC và JPA.

Môi trường và ứng dụng yêu cầu:

- JDK 7.0.

- STS 3.7 (hoặc eclipse có cài STS).

- Tomcat 7.

Để có thể viết được 1 chương trình JavaWeb sử dụng Spring, ta cần phải trải qua các bước sau.

(1)Tạo project.

(2) Cấu hình file pom.xml.

(3) Cấu hình file web.xml.

(4) Tạo package chuẩn mvc.

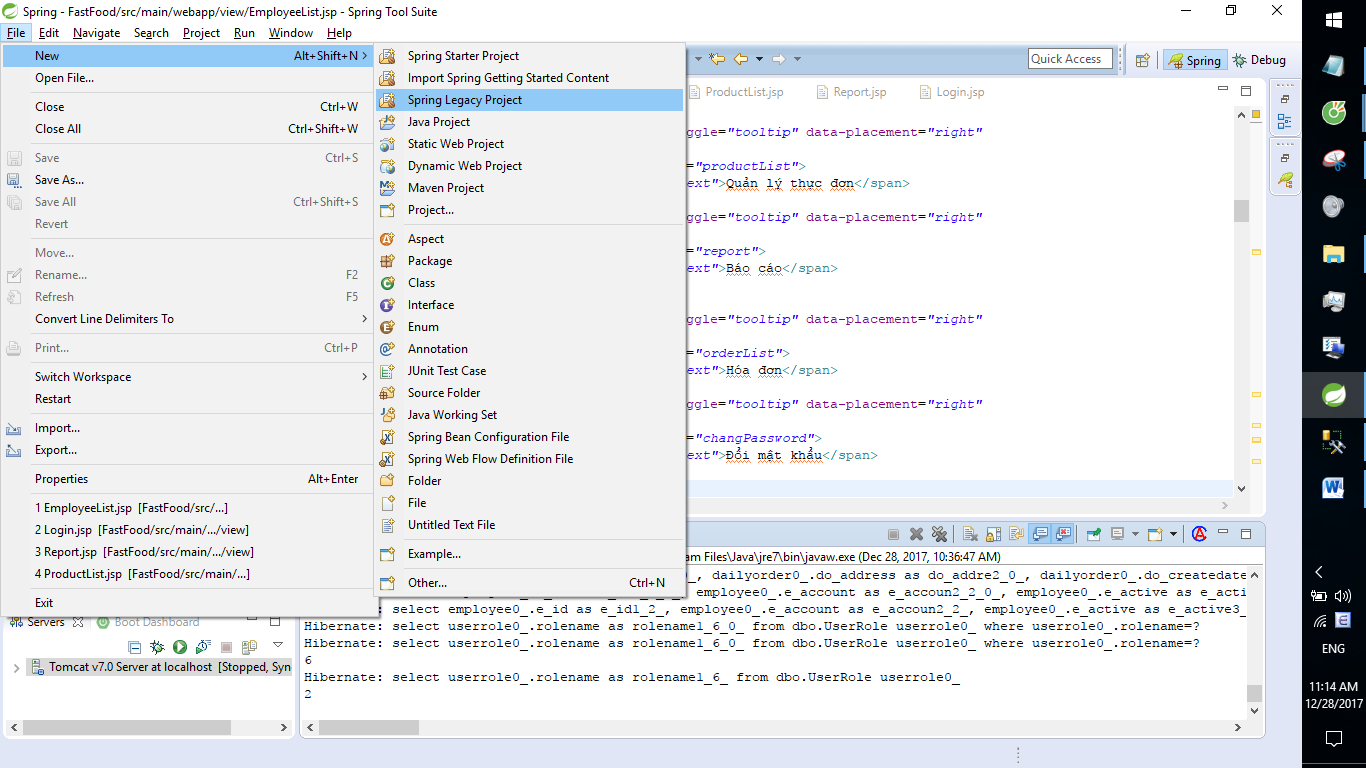
(5) Cấu hình file mvc-config (hoặc spring-mvc.xml nếu như dùng eclipse).

(6) Cấu hình file beans.xml.

(7) Viết controller.

(8) Chạy ứng dụng trên server Tomcat.

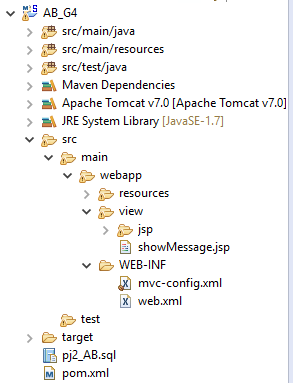
### 1.1)Tạo project.



Sau khi khởi động STS, ta chọn workspace (nơi lưu trữ các thông tin về các project, không nhất thiết phải lưu project chung với workspace),tiến hành tạo project mới:

- Chọn File => new Spring Legacy project => Simple Spring Web Maven và đặt tên cho project => finish.

- Đợi chương trình build project.



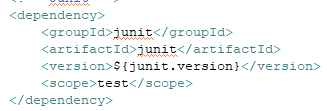
### 1.2)Cấu hình file pom.xml.

Sau khi chương trình đã build xong project, ta tiến hành cấu hình file pom.xml. Đây là bước quan trọng nhất, quyết định xem project có chạy được hay không.

- Trong Package Explorer, chọn đến project vừa tạo, cuối project sẽ là file pom.xml.

- Mở file pom.xml và chọn sang tag pom.xml.

- Trong thẻ dependencies, ta sẽ viết các thư viện cần thiết vào để có thể chạy được project.



- groupId: Tên package khi lưu trong project, thường để mặc định để dễ tìm kiếm, kiểm tra.

- artifactId: Tên thư viện cần tải về.

- vesion: phiên bản của thư viện.

Để tiện, ta sẽ viết theo định dạng sau: (groupId)(artifactId)(version)

- Các thư viện cần thiết:

+ (junit)(junit)(4.12): thư viện dùng để viết các test case, là các chương trình dùng để test chức năng của từng hàm.

+ (javax.servlet)( javax.servlet-api)( 4.3.10.RELEASE): Chứa các API để lập trình web.

+ ( org.springframework)( spring-core)( 4.3.10.RELEASE). Để chạy Spring.

+ (org.springframework)( spring-beans)( 4.3.10.RELEASE). Để chạy Spring.

+ (org.springframework)( spring-context)( 4.3.10.RELEASE). Để chạy Spring.

+ (org.springframework)( spring-jdbc)( 4.3.10.RELEASE). Để chạy Spring.

+ (org.springframework)( spring-test)( 4.3.10.RELEASE). Để chạy Spring.

+ (org.springframework)( spring-webmvc)( 4.3.10.RELEASE). Để chạy Spring.

+ (org.springframework)( spring-orm)( 4.3.10.RELEASE). Để chạy Spring.

+ (org.springframework.data)( spring-data-jpa)( 1.10.2.RELEASE). Để chạy jpa.

+ (org.hibernate)( hibernate-core)( 4.3.10.Final). Thư viện của hibernate.

+ (org.hibernate)( hibernate-entitymanager)( 4.3.10.Final). Thư viện của hibernate.

+ (org.hibernate.javax.persistence)( hibernate-jpa-2.1-api)( 1.0.0.Final). Thư viện cung cấp cài đặt Jpa của hibernate.

+ (com.microsoft.sqlserver)( mssql-jdbc)( 6.1.0.jre7). Dùng để kết nối đến SQL SERVER

+ (javax.servlet)(jstl)(1.2). Thư viện jstl dung để code trang jsp.

- Sau các Dependency là Build.

+ finalName: tên trang web khi chạy trên server.

+ complier plugin: thông báo cho IDE biết phiên bản Java dùng để biên dịch

+ war plugin: war là file sản phẩm. Sau khi trang web được lập trình xong, ta sẽ dùng file war để chạy.



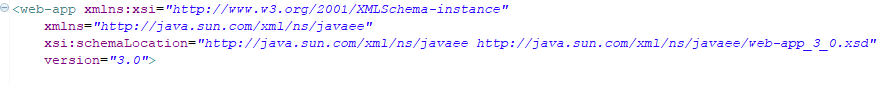
- Sau khi cấu hình xong, ta click chuột phải vào project, chọn maven update và chờ maven tải thư viện về.

### 1.3) Cấu hình file web.xml.

- Sau khi maven update xong, ta tiến hành cài đặt file web.xml.

- file web.xml nằm trong đường dẫn: src => main => webapp => WEB-INF => web.xml.

- Lên mạng tìm với keyword:”web 3.0 header” và copy header về, thay thế header hiện tại của web.xml.



- Sau đó ta lại update Maven 1 lần nữa.



- Trong file web.xml, ta sẽ cấu hình các thông số sau:

+ welcome-file-list: Danh sách các trang index.

+ Context param: đường dẫn đến các file config khác, dùng để đọc các biến khai báo sẵn, mặc định lưu trong application-config,xml.

+ listener: Dùng để gọi các file context. xml trong context param mỗi khi chương trình được khởi động.

+ servlet: initparam: đọc file config cho server.

+ servlet-mapping: liệt kê các định dạng url nào sẽ được servlet xử lý.

### 1.4) Tạo package chuẩn mvc.

- Đến đường dẫn src/main/java click phải chọn new package.

- Ta tạo 1 package gốc để chứa các package xử lý (M và C).

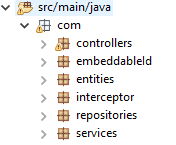
- Lần lượt tạo các package:

+ controller: Chứa các class controller, thực hiện xử lý các request và trả lại response.

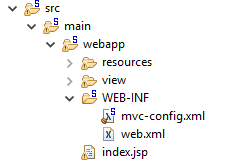
+ entities: chứa các class entity, là các object được định nghĩa tương đương với bảng cơ sở dữ liệu quan hệ.

+ repositories: các interface extends từ JpaRepository, dùng để khai báo các hàm xử lý với tầng database.

+ services : tầng trung gian giữa controller và repository, dùng để xử lý nghiệp vụ.



- Đến đường dẫn src => main => webapp.



- webapp sẽ là root của trang web.

- Đây là nơi lưu các file “view”, tức là các trang jsp, html, và các resourse của các trang đó như css, js,image…

- Tạo package view để chứa các file web,

- Tạo package resourses để chứa các file còn lại.

### 1.5) Cấu hình file mvc-config.xml

mvc-config.xml là file dùng để cấu hình các thông số để xử lý, ví dụ như: jpa, đường dẫn đến thư mục control…

mvc-config.xml có đường dẫn được viết trong file web.xml, khi server chạy, Spring sẽ load file web.xml, đọc file application-config.xml và file mvc-config.xml.

Spring sử dụng cơ chế DI để đưa các object vào sử dụng, các object được gọi là các bean, tức là các hạt đậu, sống hết chu kì rồi lại chết.

mở mvc-config.xml, chọn tab Namespace, tích chọn jpa, repository, tx., quay lại source.

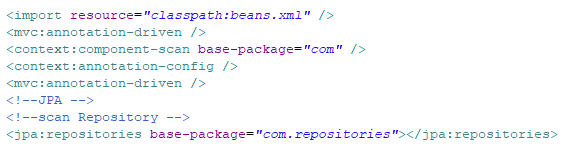
Để chạy được Jpa, ta cần phải config 2 thẻ sau:

- transactionManager.

- entityManagerFactory.



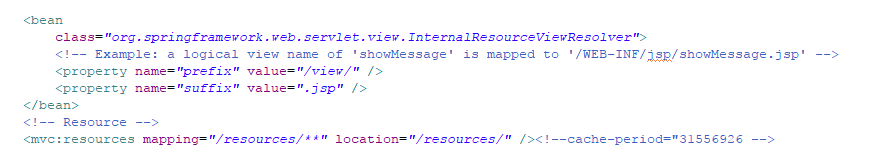
entityManagerFactory sẽ thực hiện kết nối với cơ sở dữ liệu, với dataSource là config vào database, jpaProperties là các thông số cài đặt hibernate.



file beans.xml là resource được lưu tại đường dẫn src/main/resources, chứa config đến database.

component-scan: đường dẫn đến package chứa các control, được tạo ở bước 1.4.

jpa:repositories: đường dẫn đến package repositories tạo trong bước 1.4

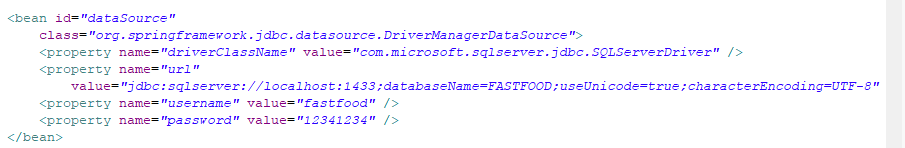


internalResourseViewResolver: xử lý các request, nếu request trùng với mapping trong controller thì sẽ gọi controller, nếu không sẽ gẳn prefig và suffix.

vd: request gửi 1 xâu “addStudent’, resourceResolver sẽ kiểm tra xem tất cả controller có controller nào xử ls không, nếu không có sẽ gọi đến trang /view/addUser.jsp.

mvc:resourse: chỉ ra đường đẫn đến file resource như js,image. Các yêu cầu về resource sẽ không đi qua resolver, mà đi thẳng đến thư mục chứa resource.

### 1.6) Cấu hình file beans.xml.

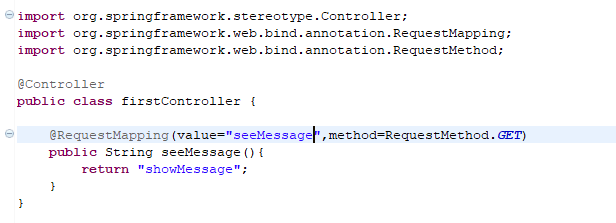


file bean.xml là file chứa các bean, tức là các tham số để khởi tạo đối tượng.

### 1.7) Viết controller.

Sau khi đã config xong, ta bắt đầu viết controller đầu tiên

- Vào package controllers tạo ở bước 1.4, tạo 1 class controller, ví dụ: firstController.java



- @Controller là annotation, dùng để khái báo rằng đây là 1 class controller.

- requestmapping:

+ value: đường dẫn để vào controller này.

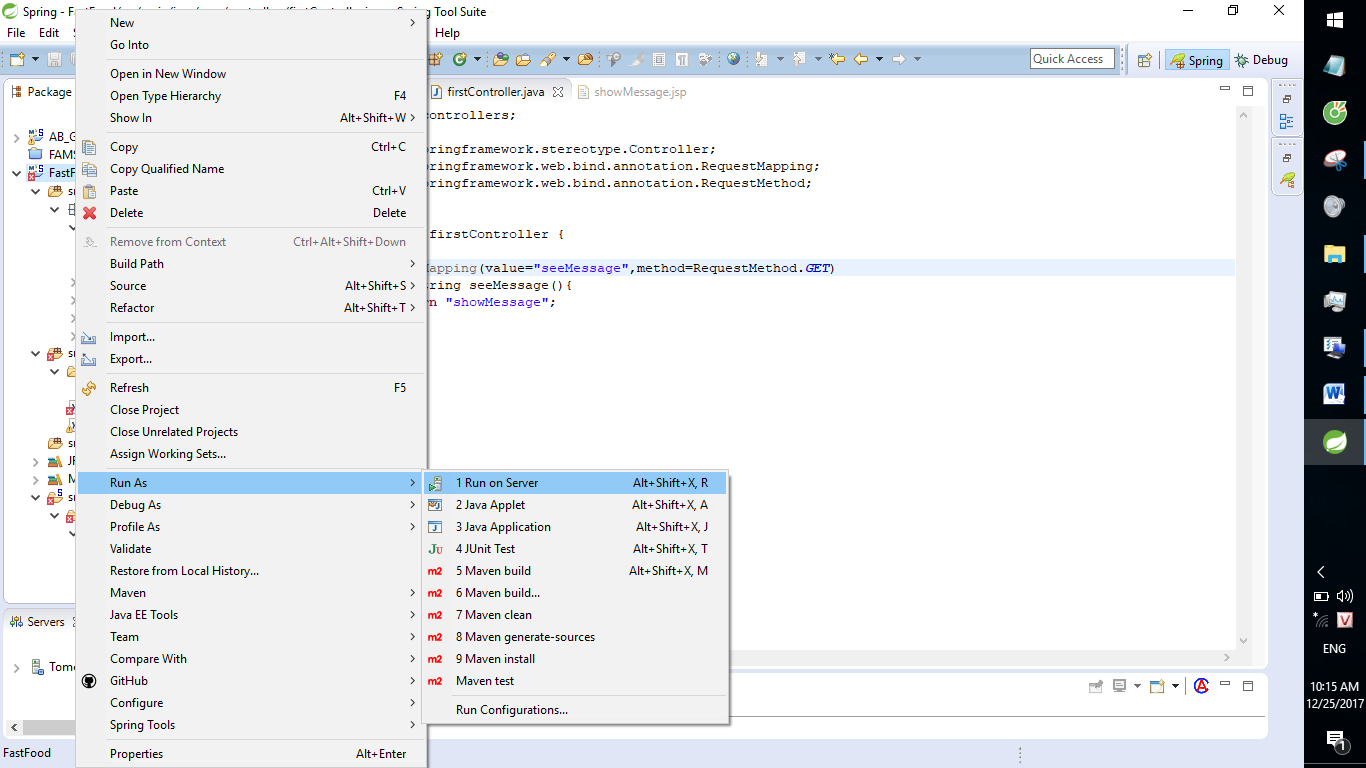
+ method: method của request.

+ seeMessage: tên method, dùng để phân biệt giữa các method trong 1 class.

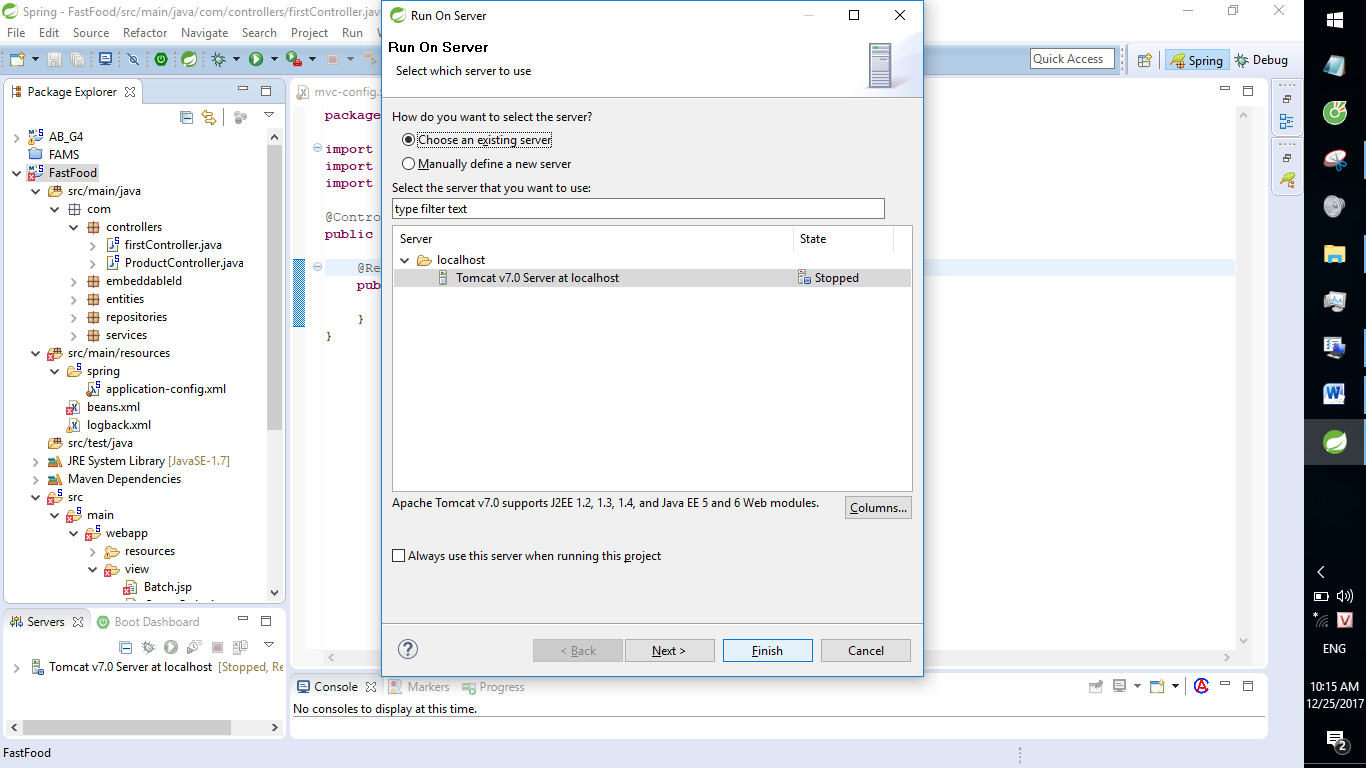
+ return “showMessage”: trả lại xâu showMessage. Resolver sẽ bắt xâu này và trả về trang showMessage.jsp.

### 1.8) Chạy ứng dụng trên server Tomcat.

- Click chuột phải vào project, chọn run on server.

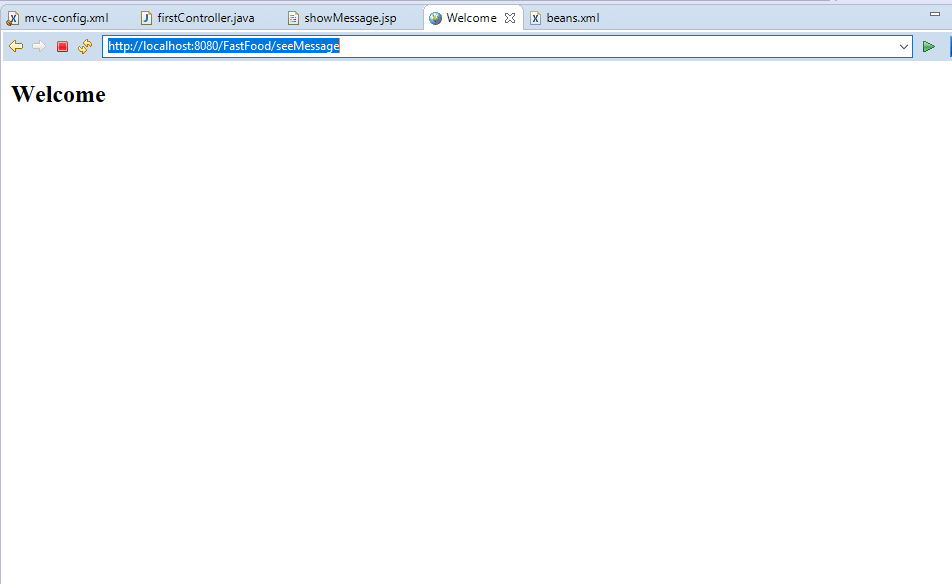


- Chọn server Tomcat 7 và finish.



- Sau khi Tomcat 7 khởi động xong, trình duyệt sẽ mở về trang index mặc định khai báo trong file web.xml.

- Sửa đường dẫn đến controller vừa viết ở trên.

- Trình duyệt sẽ đưa đến trang showMessage.jsp.

# Chương IV: PHÂN TÍCH - THIẾT KẾ ĐỀ TÀI

## 1) Xác định phạm vi, kế hoạch thực hiện.

Đề tài lựa chọn: Xây dựng website “Quản lý bán đồ ăn nhanh” trên ngôn ngữ Java. Có sử dụng các Framework Hibernate, Spring MVC, Jsp, Bootstrap.

+ Dùng SQL SERVER để lưu trữ data.

+ Tìm hiểu ngôn ngữ Java và các Framework.

+ Kết hợp các Framework vào thiết kế lập trình trang web.

Với yêu cầu trên, em đặt ra kế hoạch thực hiện đề tài như sau:

|  |  |  |
| --- | --- | --- |
| STT | Tên công việc | Thời gian thực hiện |
| 1 | Tìm hiểu nghiệp vụ, xác định phạm vi bài toán | 11/09/2017-22/09/2017 |
| 2 | Thiểt kế chức năng, giao diện, cơ sở dữ liệu | 23/09/2017-13/10/2017 |
| 3 | Tìm hiểu công nghệ | 14/10/2017-27/10/2017 |
| 4 | Lập trình các chức năng | 28/10/2017-15/12/2017 |
| 5 | Viết báo cáo, kiểm tra chức năng, hoàn thiện chương trình | 16/12/2017-22/12/2017 |

Bảng : Kế hoạch công việc

## 2) Xác định yêu cầu người dùng.

Quản lý gồm rất nhiều công việc như thực đơn, nhân viên, báo cáo, thống kê doanh thu,… Người quản lý sẽ phải kiêm rất nhiều công việc dẫn đến kinh doanh không hiệu quả.

Website quản lý bán đồ ăn nhanh sẽ hỗ trợ việc quản lý được hiệu quả hơn.

+ Khách hàng sẽ đến quầy để gọi món ăn.

+ Nhân viên sẽ thêm các món ăn vào hóa đơn theo yêu cầu của khách hàng.

+ Sau khi chọn xong hóa đơn sẽ được tính toán và in ra.

+ Khách hàng sẽ thanh toán luôn tại quầy trước khi nhận phần ăn.

+ Danh sách món ăn khách hàng đã chọn sẽ được đưa xuống bếp để thực hiện.

+ Phần ăn sẽ được chuyển từ bếp lên quầy khi xong.

+ Nhân viên có thể hủy hóa đơn khi khách hàng có yêu cầu đổi, trả.

+ Mỗi ngày sẽ có thống kê, báo cáo chi tiết về doanh thu.

## 3) Phân tích hệ thống.

### 3.1) Đối tượng sử dụng hệ thống:

Có 2 đối tượng sử dụng hệ thống: quản lý và trực quầy.

3.2) Chức năng của nhân viên trực quầy:

+ Đăng nhập, đăng xuất.

+ Chỉnh sửa 1 số thông tin cá nhân, mật khẩu.

+ Quản lý hóa đơn,

### 3.3) Chức năng của người quản lý:

+ Đăng nhập quản trị, đăng xuất.

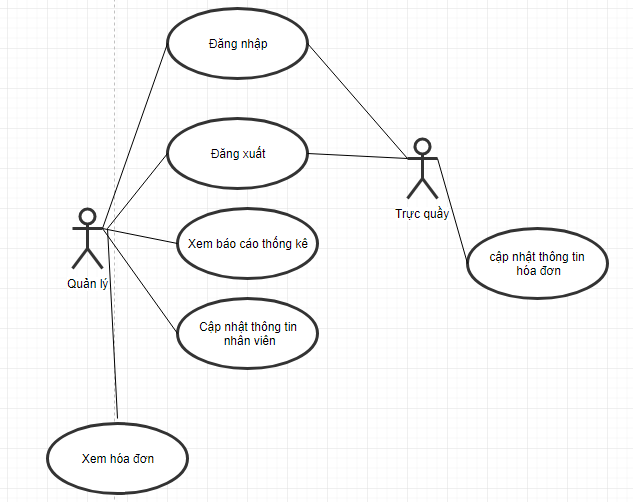
+ Quản lý hóa đơn (Xem, in, xuất file csv…).

+ Quản lý nhân viên.

+ Báo cáo thống kê.

## 4) Sơ đồ hệ thống.

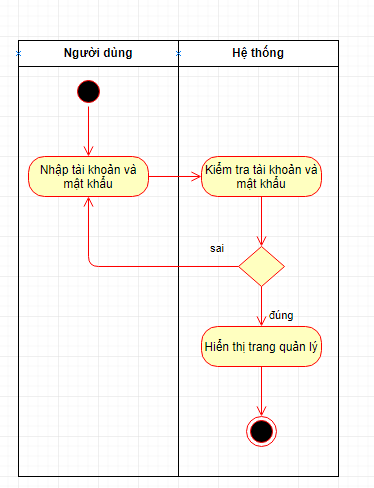
### 4.1) Sơ đồ usecase tổng quát.



Hình : Sơ đồ usecase mức tổng quát.

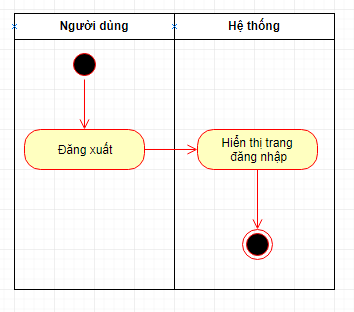
### 4.2)Đăng nhập.

4.2.1)Activity đăng nhập



Hình : Activity đăng nhập

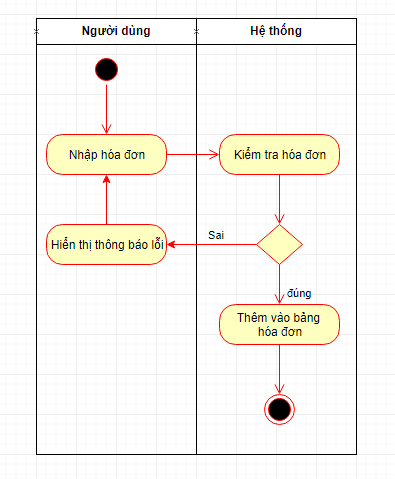
4.2.2)Activity đăng xuất.



Hình : Activity đăng xuất

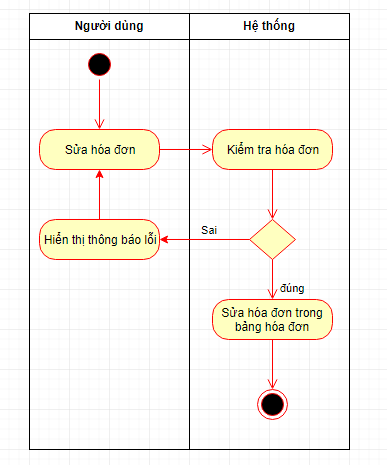
### 4.3) Cập nhật hóa đơn

4.3.1) Activity thêm hóa đơn.



Hình Activity thêm hóa đơn.

4.3.2) Activity sửa hóa đơn.



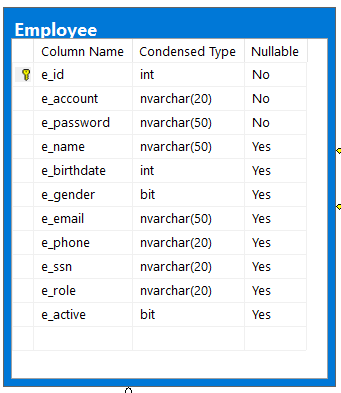
Hình : Activity sửa hóa đơn.

## 5) Thiết kế cơ sở dữ liệu.

### 5.1) Bảng dữ liệu.

5.1.1) Bảng nhân viên.

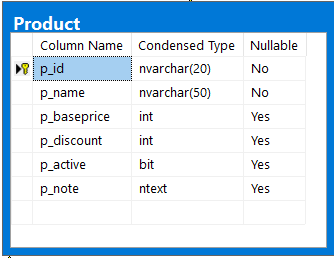
Lưu các thông tin của nhân viên.



Hình : Bảng nhân viên.

5.1.2) Bảng thực đơn.

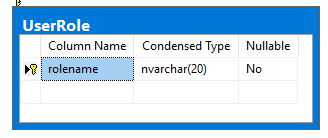
Lưu trữ các thực đơn.



Hình : Bảng thực đơn.

5.1.3) Bảng phân quyền.

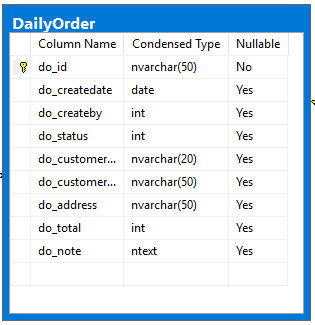
Lưu trữ các quyền truy cập.



Hình : Bảng phân quyền.

5.1.4) Bảng hóa đơn hàng ngày.

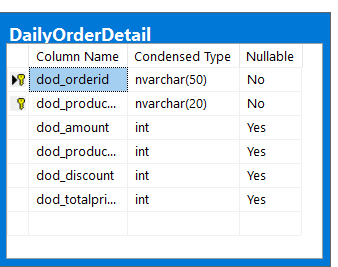
Lưu trữ các hóa đơn được tạo tao trong ngày.



Hình : Bảng hóa đơn hàng ngày.

5.1.5) Bảng chi tiết hóa đơn hàng ngày.

Lưu trữ chi tiết các hóa đơn trong ngày.



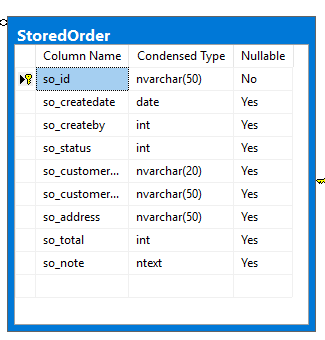
Hình : Bảng chi tiết hóa đơn hàng ngày.

5.1.6) Bảng hóa đơn đã lưu.

Để tăng hiệu năng khi tạo hóa đơn, em tạo thêm bảng hóa đơn đã lưu.

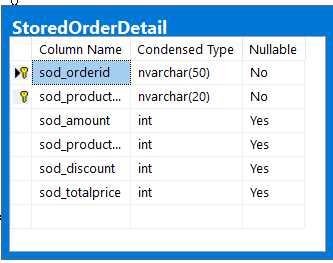
Trực quầy sẽ thêm hóa đơn vào bảng hóa đơn hàng ngày.

Quản lý sẽ nhấn nút “lưu hóa đơn hàng ngày” để chuyển tất cả hóa đơn hàng ngày sang bảng hóa đơn đã lưu( vào cuối ngày sau khi đã làm xong báo cáo).



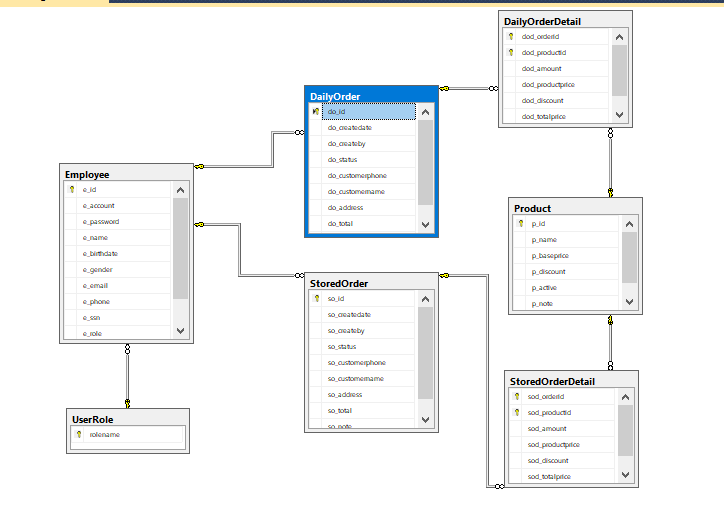
Hình : Bảng hóa đơn đã lưu

5.1.7) Bảng chi tiết hóa đơn đã lưu.



Hình : Bảng chi tiết hóa đơn đã lưu.

### 5.2) Sơ đồ cơ sở dữ liệu quan hệ.

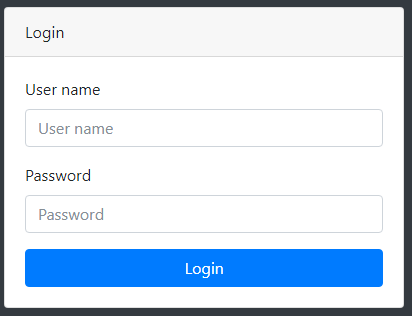


Hình : Sơ đồ cơ sở dữ liệu quan hệ.

# CHƯƠNG V: XÂY DỰNG CHƯƠNG TRÌNH.

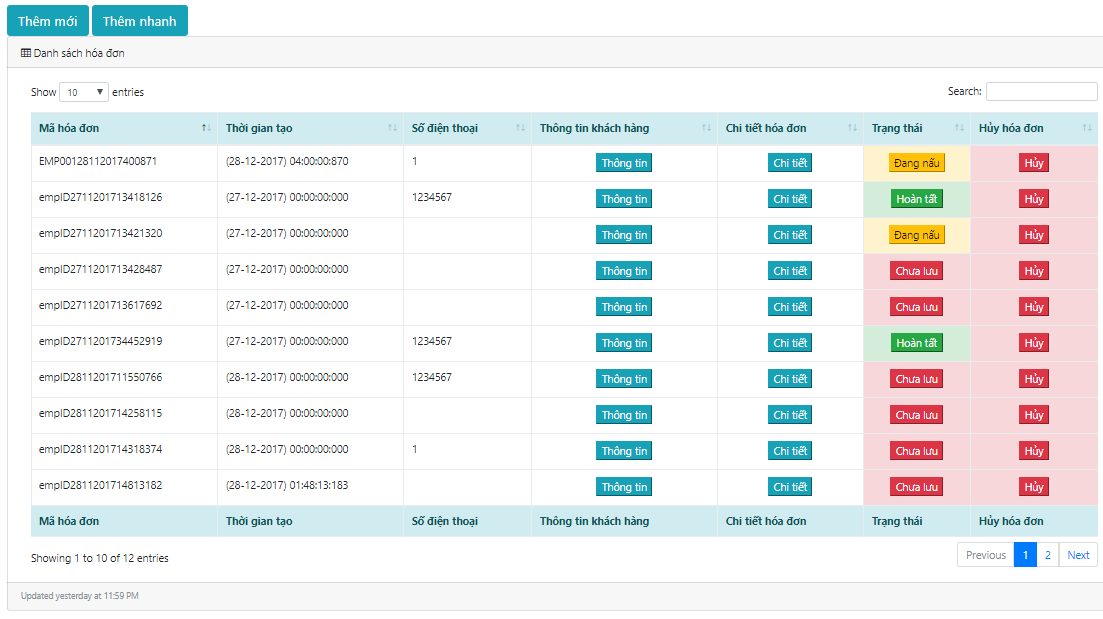
## 1) Giao diện người dùng.

### 1.1) Màn hình đăng nhập.



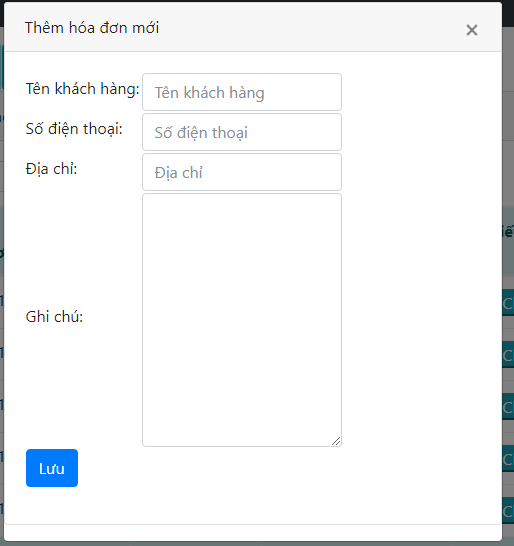
Hình :Màn hình đăng nhập.

### 1.2)Màn hình danh sách hóa đơn.



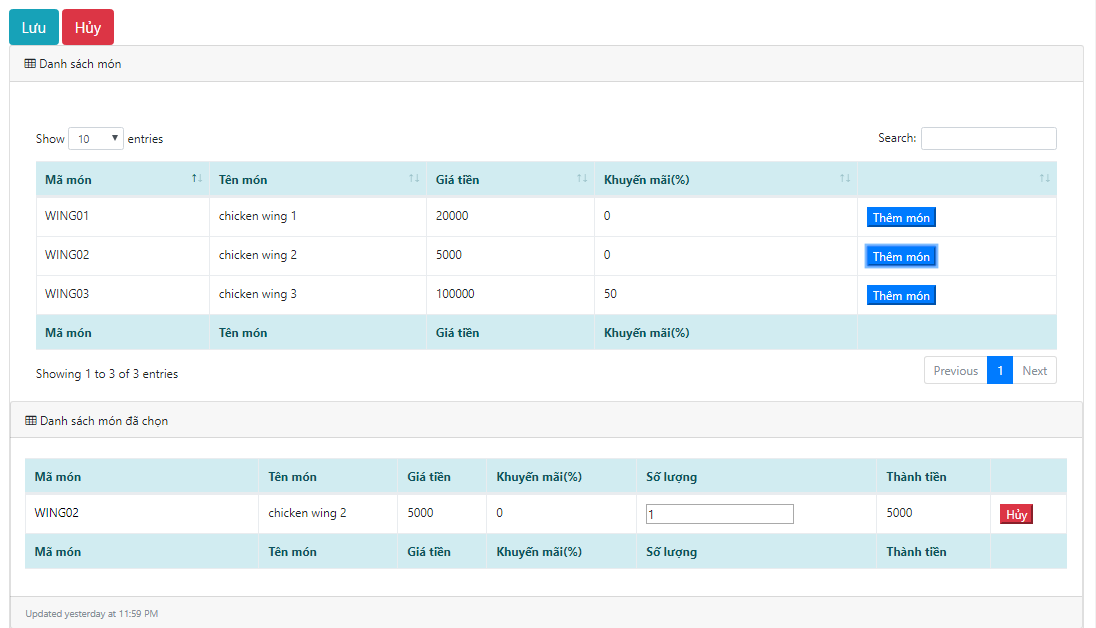
Hình :Màn hình danh sách hóa đơn.

### 1.3) Màn hình thêm hóa đơn mới



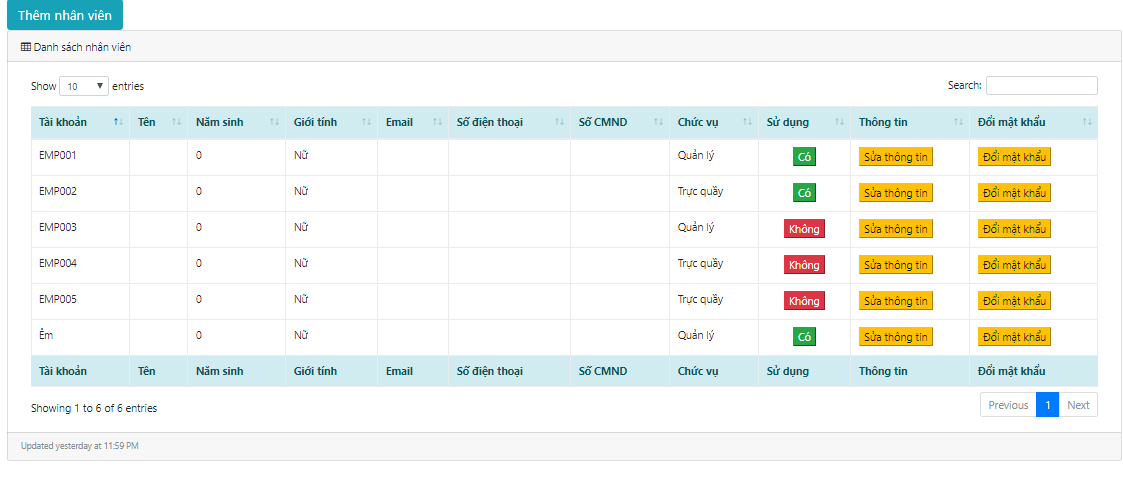
Hình Màn hình thêm hóa đơn mới.

### 1.4) Màn hình chi tiết hóa đơn.



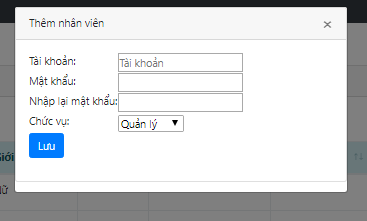
Hình : Màn hình chi tiết hóa đơn.

### 1.5) Màn hình danh sách nhân viên.



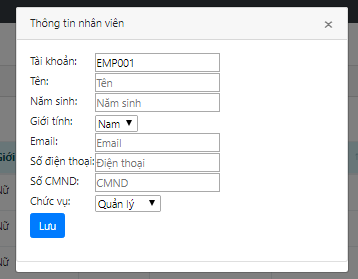
Hình : Màn hình danh sách nhân viên.

### 1.6) Màn hình thêm nhân viên mới.



Hình : Màn hình thêm nhân viên mới.

### 1.7) Màn hình thông tin nhân viên.



Hình : Màn hình sửa nhân viên.

# CHƯƠNG VI: KẾT LUẬN VÀ HƯỚNG PHÁT TRIÊN

Trong thời gian làm đồ án, em đã cố gắng hết mình, nỗ lực tìm hiểu những kiến thức liên quan, bám sát với chức năng bài toán đề ra để hoàn thành đồ án. Tuy nhiên, bởi các lý do khách quan cũng như chủ quan, đồ án này vẫn còn nhiều thiếu sót. Mong các thầy cô giáo bộ môn nhận xét, đánh giá, đóng góp ý kiến để đồ án được hoàn thiện, có tính thực tế hơn. Đồng thời giúp em rút được kinh nghiệm, bài học quý giá trước khi vào thực tế.

Dưới đây là 1 số đánh giá của cá nhân em thực hiện đề tài.

## **1) Kết quả đạt được.**

- Hoàn thành ứng dụng.

- Xây dựng được các chức năng cần thiết cho website quản lý bán đồ ăn nhanh.

- Đáp ứng được nhu cầu của người sử dụng:

+ Lưu trữ, quản lý hóa đơn, thông tin thực đơn, nhân viên.

+ Xem báo cáo thống kê.

+ Giao diện ưa nhìn, dễ dùng.

+…

- Bên cạnh đó em cũng tìm hiểu và học hỏi được thêm nhiều kiến thức mới về:

+ Ngôn ngữ Java.

+ Lập trình Java web.

+ Spring MVC.

+ Bootstrap.

+ Hibernate, JPA,Jsp,…

## 2) Các khó khăn trong quá trình thực hiện đồ án.

Trong quá trình làm đồ án, em đã gặp phải không ít khó khăn, nhưng với sự giúp đỡ của thầy cô hướng dẫn, bạn bè em đã vượt qua và hoàn thành được đề tài. Ví dụ cụ thể về 1 số khó khăn mắc phải trong quá trình làm:

- Cấu hình và chạy Hibernate.

- Áp dụng framework vào đề tài.

- Áp dụng js và jquery.

- Thiết kế giao diện phù hợp.

## 3) Hạn chế:

Phần mềm vẫn còn nhiều hạn chế cần phải khắc phục như:

- Chức năng còn hạn chế, thiết sót.

- Cần bổ sung thêm nhiều tính năng hơn.

- Mức độ bảo mật chưa toàn diện.

## 4) Định hướng phát triển.

Đề tài vẫn còn nhiều tiềm năng để phát triển, ví dụ như:

- Thêm các tính năng thống kê, kiểm toán.

- Mở rộng số lượng người có thể sử dụng hệ thống.

- Hoàn thiện thêm các chức năng khác.

- Cho phép khách hàng đặt hàng online.

- Thu thập thông tin khách hàng để phân tích, định hướng phát triển.

- Mở rộng quy mô bán hàng.

## 5) Công cụ mã nguồn.

### 4.1) Các công cụ được sử dụng:

|  |  |
| --- | --- |
| STT | Tên phần mềm |
| 1 | STS 3.7.3.RELEASE |
| 2 | SQL SERVER 2012 |
| 3 | Trình duyệt Cốc cốc |
| 4 | Trình vẽ online draw.io |

### 4.2) Mã nguồn, thư viện:

|  |  |  |
| --- | --- | --- |
| STT | Tên mã nguồn, thư viện | Chú thich |
| 1 | Bootstrap.min.css | Thư viện css bootstrap |
| 2 | Adminlte | Thư viện css bootstrap |

## 6) Tài liệu tham khảo.

(1) Tài liệu tham khảo trên mạng.

(2) Các video trên youtube.

(3) Tài liệu tham khảo do thầy cô và các bạn cung cấp.

(4) tutorialspoint.com