**TRƯỜNG ĐẠI HỌC CẦN THƠ**

**KHOA CÔNG NGHỆ THÔNG TIN**

**& TRUYỀN THÔNG**

**----🙣🕮🙡----**

CÔNG NGHỆ THÔNG TIN

NIÊN LUẬN

***Đề tài:***

**WEBSITE BÁN QUẦN ÁO ONLINE**



***Giảng viên hướng dẫn* : TS Lâm Nhựt Khang**

***Sinh viên thực hiện* : Trần Văn Ngoan – B1304867**

***Lớp:* DI13V7A1 – Công nghệ thông tin A1 - K39**

***Cần Thơ, tháng 8-2016***

***Học kỳ I, Năm học 2016-2017***

# MỤC LỤC

**Hình ảnh:**

|  |  |  |
| --- | --- | --- |
| **Ảnh** | **Tên Ảnh** | **Nguồn** |
| **1** |  |  |
| **2** | Mô hình MVC | https://khoanguyen.me/tim-hieu-mo-hinh-mvc-la-gi/ |
| **2** |  |  |

# PHẦN 1: GIỚI THIỆU

1. **ĐẶT VẤN ĐỀ**

* Ngày nay với sự phát triển của Internet, nhu cầu buôn bán trao đổi hàng hóa qua mạng đang phát triển và sôi động hơn bao giờ hết. Việt Nam đang trong thời kì hôị nhập với quốc tế, công nghệ và kinh tế phát triển góp phần thúc đẩy phát triển kinh doanh thương mại điện tử. Mọi mặt hàng từ nhà cửa, xe cộ, đồ điện tử, quần áo, giày dép,.. đều được kinh doanh qua mạng Internet, từ đó nhiều ứng dụng, website được tạo ra để đáp ứng nhu cầu của thị trường. Vì vậy, em thực hiện đề tài **“Xây dựng website bán quần áo”** nhằm đáp ứng xu thế phát triển của thương mại điện tử tại Việt Nam.

1. **MỤC TIÊU**

* Xây dựng ra một website kinh doanh quần áo trực tuyến dễ dử dụng, nhằm cung cấp những sản phẩm cũng như những trải nghiệm tốt nhất đến cho người tiêu dùng.
* Tìm hiểu công nghệ làm web dựa trên java và áp dụng các framework Spring, Hibernate vào đề tài.

1. **ĐỐI TƯỢNG VÀ PHẠM VI NGHIÊN CỨU**

* Đối tượng nghiên cứu là kết hợp hai framework Spring và Hibernate dựa trên nền tảng Java . Nghiên cứu cách thiết kế website sử dụng công nghệ Bootstrap.
* Phạm vi của đề tài xoay quanh việc thiết kế, xây dựng website với các công nghệ trên. Được thực hiện tại khoa CNNT & TT Đại Học Cần Thơ, vào học kì 1 năm 2016 – 2017.

1. **PHƯƠNG PHÁP NGHIÊN CỨU**

* Nghiên cứu các tài liệu có liên quan.
* Khảo sát các website, các đồ án thực tế có cùng chủ đề.
* Trao đổi với người hướng dẫn.
* Tổng hợp kinh nghiệm thực tế

1. **NỘI DUNG NGHIÊN CỨU**

* Jsp và Severlet ,framework Spring, Hibernate
* Tìm hiểu Tomcat server.
* Biết cách thiết kế website sử dụng công nghệ Bootstrap.
* Kết hợp tìm hiểu về Jquery, Ajax.
* Dùng Git để quản lí phiên bản.

1. **BỐ CỤC NIÊN LUẬN**

* Bao gồm : phần phụ lục, giới thiệu, nội dung, kết luận, tài liệu tham khảo.
* Phần nội dung có 4 phần :

+ Đặc tả yêu cầu

+ Thiết kế giải pháp

+ Cài đặt giải pháp

+ Đánh giá kiểm thử

# PHẦN 2: NỘI DUNG

1. **ĐẶC TẢ YÊU CẦU**
   1. **YÊU CẦU NGƯỜI DÙNG**
      1. **Yêu cầu chức năng :**

* **Đối với khách hàng**

+ **Khách hàng** vãng lai và khách hàng là thành viên của đều có thể sử dụng đầy đủ các dịch vụ của website tuy nhiên khách hàng là thành viên sẽ có 1 số ưu đãi như thông báo các tin tức nhanh hơn, có các chương trình khuyến mãi giành cho thành viên.

**+** **Đăng kí và đăng nhập**: người dùng đăng kí tài khoản miễn phí trên website.

**+ Xem và đặt mua sản phẩm**: người dùng có thể xem thông tin sản phẩm hay đặt mua sản phẩm. Sau khi đặt mua sẽ có người liên hệ để xác nhận thông tin, tiến hành giao hàng và thanh toán. Khách hàng có 2 lựa chọn cho việc thanh toán là COD tức giao hàng và nhận tiền trực tiếp hoặc khách hàng có thể thanh toán trực tuyến thông qua các tài khoản ngân hàng.

**+ Giỏ đồ**: người dùng đưa danh sách các sản phẩm mình thích vào giỏ đồ để có thể truy cập lại nhanh nhất các sản phẩm mình muốn mua.

**+ Tìm kiếm sản phẩm:** người dùng có thể tìm kiếm sản phẩm mình yêu thích theo các từ khóa

**+ Tin tức:** người dùng có thể xem các tin tức về khuyến mãi, thời trang trên website

**+ Bình luận:** người dùng có thể để lại ý kiến của mình đối với các sản phẩm cũng như tham gia vào bình luận các phần tin tức.

* **Đối với người quản trị**

**+ Đăng nhập** bằng tài khoản quản trị để đến trang quản trị

**+ Quản lí sản phẩm:** thêm, sửa, xóa thông tin của các sản phẩm, xử lí các sản phẩm được khuyến mãi.

**+ Quản lí phần tin tức:** thêm, sửa, xóa các tin tức

**+ Quản lí đơn hàng:** xem, xác nhận , hủy các đơn hàng.

**+ Bình luận:** duyệt bình luận của khách hàng vàphản hồi với các bình luận của khách hàng

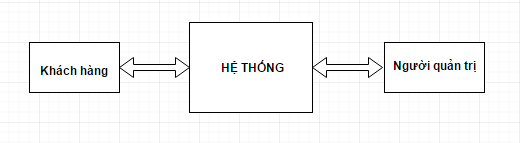
* + 1. **Yêu cầu phi chức năng :**

+ Giao diện được thiết kế thân thiện, màu sắc, bố cục rõ ràng giúp người dùng sử dụng dễ dàng nhất.

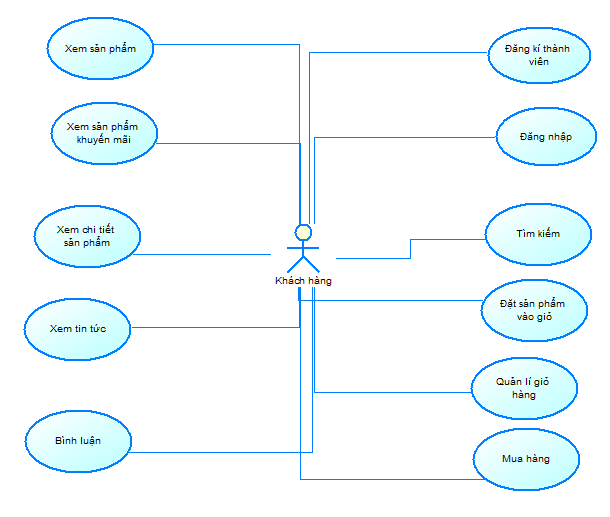
+ Tốc độ xử lí các tác vụ đáp ứng phải nhanh, độ trễ không được vượt quá mức cho phép

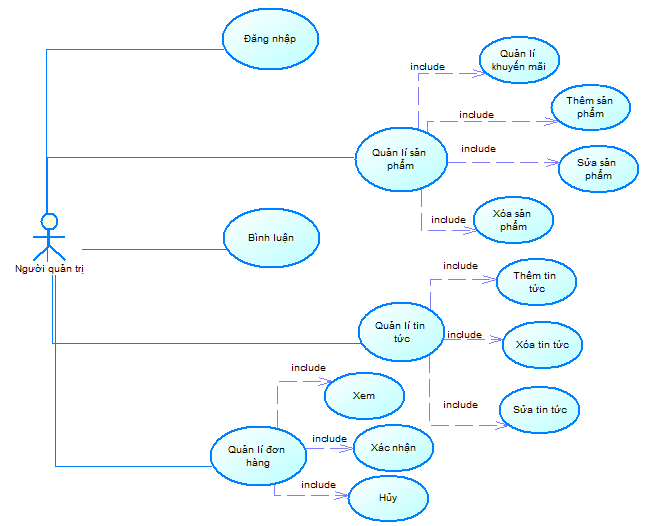
+ Website chạy được trên các trình duyệt phép dành cho laptop, tablet, di động.

* 1. **YÊU CẦU HỆ THỐNG**
     1. **Mục tiêu của hệ thống**
* Hệ thống cho phép khách hàng cũng như người quản trị thực hiện các dịch vụ, chức năng một cách tốt nhất. Hệ thống sẽ có tính khả chuyển cao khi chuyển đổi môi trường hoạt động như thay đổi server… Ngoài ra, hệ thống sẽ cung cấp khả năng bảo mật nhằm bảo vệ quyền lợi của khách hàng.
  + 1. **Mô hình của hệ thống**



**Hình 1: Mô hình hệ thống**

* 1. **SƠ ĐỒ USECASE**
* Usecase phía khách hàng :
* Usecase phía người quản trị website



* 1. **MÔ TẢ SƠ ĐỒ USE CASE**

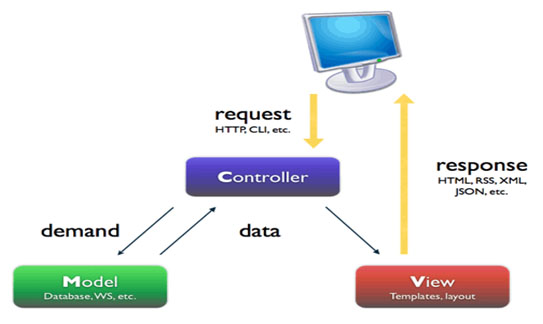
|  |  |  |  |
| --- | --- | --- | --- |
| 1. **#** | **Mã** | **Tên Usecase** | **Mô tả Usecase** |
| **Khách hàng :** | | | |
| 1 | UC01 | Đăng kí thành viên | Cho phép khác hàng đăng kí thành viên miễn phí của website |
| 2 | UC02 | Đăng nhập | Cho phép khách hàng đăng nhập để sử dụng tốt nhất dịch vụ của website |
| 3 | UC03 | Xem sản phẩm | Cho phép khách hàng xem các sản phẩm theo loại |
| 4 | UC04 | Xem sản phẩm khuyến mãi | Cho phép khách xem các sản phẩm được khuyến mãi |
| 5 | UC05 | Xem sản phẩm chi tiết | Cho phép khách hàng xem thông tin chi tiết sản phẩm như giá, chất liệu, size. |
| 6 | UC06 | Tìm kiếm sản phẩm | Cho phép khách hàng tìm kiếm sản phẩm dựa theo các từ khóa |
| 7 | UC07 | Đặt sản phẩm vào giỏ hàng | Cho phép khách hàng đặt các sản phẩm yêu thích vào giỏ hàng |
| 8 | UC08 | Quản lí giỏ hàng | Cho phép khách hàng xem lại các sản phẩm mình đã đặt vào giỏ hàng |
| 9 | UC09 | Mua hàng | Cho phép người dùng mua hàng |
| 10 | UC10 | Xem tin tức | Cho phép người dùng xem các tin tức về khuyến mãi, về thời trang … |
| 11 | UC11 | Bình luận | Cho phép người dùng để lại phản hồi với các sản phẩm, các tin tức. |
| **Người quản trị website :** | | | |
| 12 | UC12 | Đăng nhập |  |
| 13 | UC13 | Quản lí sản phẩm | Cho phép người quản trị thêm, sửa, xóa thông tin sản phẩm và quản lí các sản phẩm khuyến mãi |
| 14 | UC14 | Quản lí tin tức | Cho phép người quản trị thêm, sửa, xóa các tin tức về sản phẩm, khuyến mãi, thời trang, … |
| 15 | UC15 | Quản lí đơn hàng | Cho phép người quản trị xem, xác nhận hoặc hủy đơn hàng |
| 16 | UC16 | Bình luận | Cho phép người quản trị phản hồi lại bình luận của khách hàng |

1. **THIẾT KẾ GIẢI PHÁP**
   1. **MÔ HÌNH MVC**

* MVC (**M**odel - **V**iew – **C**ontroller), là mô hình gồm ba thành phẩn Model, View và Controller. MVC giúp cho việc tổ chức các thành phần trong *source code* về Web hay Desktop Application có hệ thống hơn, giúp người phát triển nhìn được tổng quan cấu trúc của dự án, từ đó dễ dàng trong việc triển khai, theo dõi và debug.

**Các thành phần trong mô hình MVC:**

* **Model**: chứa các class, các hàm, phương thức có nhiệm vụ thao tác với cơ sở dữ liệu. **Controller** sẽ thông qua đó để lấy dữ liệu rồi gửi qua **View.**
* **View**: chính là khung nhìn của người dùng, là nơi chứa các đối tượng về giao diện người dùng. Nhiệm vụ của **View** là nhận dữ liệu từ **Controller** và hiển thị nội dung sang các đoạn mã HTML.
* **Controller**: đóng vài trò trung gian giữa **Model** và **View**. Nó có nhiệm vụ tiếp nhận yêu cầu từ người dùng sau đó thao tác trực tiếp với **Model** tương ứng để lấy dữ liệu và gửi chúng qua **View** tương ứng rồi trả kết quả về cho client.



**Hình 2: Mô Hình MVC**

**Ưu điểm**:

* Hệ thống phân ra từng phần nên dễ dàng bảo trì, nâng cấp, phát triển
* Dễ dàng tìm kiếm, khắc phục lỗi trong quá trình xây dựng

**Nhược điểm**:

* Quá trình xây dưngj các thành phần, cấu trúc phức tạp, mất nhiều thời gian.
  1. **CÁC NGÔN NGỮ VÀ CÔNG NGHỆ ĐƯỢC SỬ DỤNG**
* **HTML** (HyperText Markup Language) là “*ngôn ngữ đánh dấu siêu văn bản*” là một ngôn ngữ được thiết kế ra để tạo nên các [trang web](https://vi.wikipedia.org/wiki/Website" \o "Website).
* **CSS** (Cascading Style Sheets) giúp ta định dạng lại bố cục, font chữ, màu nền,… trên tập tin HTML. Chúng ta có thể chèn trực tiếp code **CSS** vào tập tin **HTML** hoặc gián tiếp bằng cách chèn tập tin **CSS** vào **HTML**.
* **Bootstrap** là một bộ framework bao gồm các tập tin như **css**, **jquery***,* **javascript** được viết sẵn, người thiết kế website chỉ cần áp dụng các tập tin đó để thiết kế website một cách dễ dàng hơn. Tính năng hữu ích trên **Bootstrap** là hỗ trợ “web đáp ứng tức” là khi website viết bằng **Bootstrap** có thể hiển thị đúng bố cục trên mọi thiết bị có kích thước màn hình khác nhau như laptop, tablet, smartphone. Ngoài ra, ta có thể áp dụng các thẻ của **HTML, CSS** vào **Boostrap** để định dạng lại nội dung theo ý muốn.
* **JSP** (JavaServer Pages) là một công nghệ để phát triển, thiết kế website với các nội dung động bằng cách chèn các code của **Java** vào các trang **HTML**.
* **Java Servlets** chính là các file được viết bằng **Java** chạy trên một Web server hoặc một Application server có vai trò là **một tầng trung gian** giữa yêu cầu người dùng với cơ sở dữ liệu hoặc các ứng dụng. Có thể hiểu **Java Servlets là** một **Controller** trongmô hình **MVC.**
* **Spring Framework** là framework hỗ trợ cho ngôn ngữ lập trình **Java**. **Spring** được phát triển để xây dựng các **Website** hay **Desktop Application** theo mô hình **MVC**. **Spring** có thể kết hợp với nhiều framework khác nhau như Struts, Hibernate, …
* **Hibernate** là framework chuyển đổi mô hình **Database** sang mô hình hướng đối tượng. Nó hỗ trợ các câu lệnh truy vấn dữ liệu **HQL** (mở rộng của ngôn ngữ **SQL**) giúp cho viêc lưu trữ, truy vấn dữ liệu dễ dàng hơn.
* **Apache** là một Web server đóng vai trò giao tiếp, xử lí các yêu cầu từ người dùng trên các trang web tĩnh. Các trang web động như JSP sẽ được nó gửi đến **Tomcat** xử lí sau đó lấy kết quả là trangwebtĩnh trả về cho người dùng.
* **Tomcat** cũng là một Web server. Điểm mạnh của **Tomcat** là cung cấp một máy chủ để xử lí các **Servlet** và các trang web động **JSP** sau đó trả về các trangwebtĩnh cho người dùng.
* **Mysql** là hệ quản trị cơ sở dữ liệu nguồn mở. Lợi thế của **Mysql** là sử dụng tài nguyên ít và khá linh hoạt trong việc lưu trữ và truy vấn cơ sở dữ liệu. **Mysql** hỗ trợ nhiều ngôn ngữ lập trình khác nhau như JSP, PHP, ASP, …
* **Xampp** là chương trình tạo máy chủ Web (Web Server) được tích hợp sẵn Apache, MySQL, FileZilla, Mercury, Tomcat.
* **Git là** một hệ thống quản lý phiên bản phân tán. Sử dụng **Git** để quản lí các source code, tài liệu được dễ dàng hơn, giúp ta theo dõi được dấu vết, các lượt cập nhật, xóa, sửa các tập tin hay khôi phục chúng về các phiên bản trước đó. Điểm mạnh của **Git** là có thể phân nhánh các công việc theo nhiều hướng khác nhau, sau đó có thể tích hợp các nhánh lại với nhau.
  1. **THIẾT KẾ CÁC LỚP** 
     1. **Model**
* **Package model:** Category.java, Product.java, User.java, Item.java, Cart.java, Bill.java, BillDetail.java
* **Package dao:** CategoryDAO.java, ProductDAO.java, UserDAO.java, BillDAO.java, BillDetailDAO.java
  + 1. **Controller**
* **Package controller:** HomepageController.java, ManagerPageController.java, ProductController.java, MangerProductController.java, CartController.java, UserController.java, NewsController.java
  + 1. **Util**
* **Package util:** HibernateUtil.java
  1. **THIẾT KẾ CƠ SỞ DỮ LIỆU**
* Các bảng dữ liệu

|  |  |  |
| --- | --- | --- |
| **STT** | **Tên bảng** | **Mô tả** |
| 1 | Category | Bảng danh mục sản phẩm |
| 2 | Product | Bảng sản phẩm |
| 3 | User | Bảng người dùng |
| 4 |  |  |

* Bảng category

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** |
| 1 | CategoryID | ID loại sản phẩm | bigint | Khóa chính |
| 2 | categoryName | Tên loại sản phẩm | varchar |  |

* Bảng product

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** |
| 1 | ProductID | ID sản phẩm | bigint | Khóa chính |
| 2 | productName | Tên sản phẩm | varchar |  |
| 3 | productPrice | Giá sản phẩm | double |  |
| 4 | productImage | Ảnh sản phẩm | Varchar |  |
| 5 | productDescription | Mô tả sản phẩm | Varchar |  |
| 6 | CategoryID | ID loại sản phẩm | bigint | Khóa ngoại |

* Bảng user

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên trường** | **Mô tả** | **Kiểu dữ liệu** | **Khóa** |
| 1 | UserID | ID người dùng | bigint | Khóa chính |
| 2 | username | Tên người dùng | varchar |  |
| 3 | UserEmail | Email người dùng | varchar |  |
| 4 | userPassword | Mật khẩu người dùng | varchar |  |
| 5 | UserRole | Thành viên bình thường hay người quản trị | boolean |  |

1. **CÀI ĐẶT GIẢI PHÁP**
   1. **CÀI ĐẶT CHI TIẾT CÁC CHỨC NĂNG**
      1. **Chức năng đăng nhập**

**View:** pages/index.jsp, manager/index.jsp, pages/login.jsp

**Controller:** UserController.java

**Model:** User.java, UserDAO.java

**Datebase:** user

* **Phía người dùng**: khi người dùng vào website sẽ chuyển người dùng đến trang ***“pages/login.jsp”***, tại đây sẽ có form yêu cầu người dùng sẽ nhập đầy đủ*“Email”* và *“Mật khẩu”*. Sau đó nhấn vào nút đăng nhập. Nếu thông tin *“Email”* và *“Mật khẩu”* không chính xác sẽ có thông báo cho người dùng nhập lại. Ngược lại:
  1. Nếu là người dùng bình thường sẽ chuyển đến trang chủ cho người dùng thực hiện các chức năng của mình.
  2. Nếu là người quản trị sẽ chuyển đến trang quản trị website.
* **Phía hệ thống:** 
  1. Khi người dùng vào website, trang ***“redirect.jsp”*** sẽ chuyển người dùng đến trang ***“pages/login.jsp”.***
  2. Khi người dùng nhấn vào nút **“Đăng nhập**”, form đăng nhập tại trang ***“pages/login.jsp”*** sẽ gửi thông tin*“Email”* và *“Mật khẩu”* đến cho class ***“UerController.java” .***
  3. Tại class ***“UserController.java”,*** sẽ khởi tạo đối tượng “***User”*** và gọi phương thức “***login***” từ class ***“UserDAO.java”.***
  4. Phương thức “***login***” sẽ kiểm tra là người dùng bình thường hay là người quản trị để class ***“UerController.java”*** quyết định trả vềtrang “***pages/index.jsp”***  hoặc là trang “***manager/index.jsp”.***
     1. **Chức năng đăng kí**

**View:** pages/login.jsp, pages/register.jsp

**Controller:** UserController.java

**Model:** User.java, UserDAO.java

**Datebase:** user

* **Phía người dùng**: khi người dùng vào website sẽ chuyển người dùng đến trang ***“pages/login.jsp”***, tại đây nếu chưa có tài khoản thì nhấn vào link đăng ký sẽ được chuyển đến trang ***“pages/register.jsp”.*** Tại đây sẽ có form yêu cầu người dùng sẽ nhập đầy đủ thông tin bao gồm “*Tên*”,*“Email”*,*“Mật khẩu” và “Nhập lại mật khẩu”.* Sau khi đã điền đầy đủ thông tin, nhấn vào nút đăng ký. Nếu đăng ký thành công sẽ chuyển người dùng trên trang ***“pages/login.jsp”***. Ngược lại, sẽ báo lỗi cho người dùng.
* **Phía hệ thống:** 
  1. Khi người dùng vào website, trang ***“redirect.jsp”*** sẽ chuyển người dùng đến trang ***“pages/login.jsp”.***
  2. Khi người dùng nhấn vào nút link đăng ký sẽ chuyển đến trang ***“pages/register.jsp”.*** Tại đây, khi người dùng nhấn nút **“đăng ký”** trang ***“pages/register.jsp”*** sẽ gửi thông tinđăng ký đến cho class ***“UerController.java” .***
  3. Tại class ***“UserController.java”,*** sẽ khởi tạo đối tượng “***User”*** và gọi phương thức “***checkEMail***” và phương thức “***insertUser***” từ class ***“UserDAO.java”.***
  4. Phương thức “***checkEmali***” sẽ kiểm tra email đã tồn tại hay chưa, đồng thời kiểm tra lại mật khẩu nhập lại có đúng hay không.Nếu tất cả đã hợp lệ thị class ***“UerController.java”*** sẽ chèn thông tin nguời dùng vào bảng cơ sở dữ liệu “user” và trả về trang “***pages/login.jsp”.*** Ngược lại, thông báo lỗi cho nguời dùng.
     1. **Chức năng hiển thị danh mục sản phẩm lên website**
     2. **Chức năng hiển thị sản phẩm theo danh mục sản phẩm**
     3. **Chức năng đặt sản phẩm vào giỏ hàng**
     4. **Chức năng quản lí giỏ hàng**
     5. **Chức năng đặt hàng**
     6. **Chức năng quản lí mặt hàng**
* **Thêm sản phẩm**

**` View:** manager/addProduct.jsp, pages/register.jsp

**Controller:** ProductController.java

**Model:** Product.java, ProductDAO.java

**Datebase:** product

* + **Phía người quản trị**: sau khi đăng nhập thành công. Nhấn vào link “thêm sản phẩm mới” để chuyển sang trang “manager/addProduct.jsp”. Admin sẽ nhập đầy đủ thông tin của sản phẩm vào form. Sau đó nhấn vào nút **“Thêm”.** Nếu thành công sẽ có thông báo *“Thêm mặt hàng thành công”*. Ngược lại thông báo *“Thêm mặt hàng thất bại”.*
  + **Phía hệ thống:**
    1. Sau khi người quản trị đăng nhập thành công, nhấn link “thêm sản phẩm mới”, hệ thống sẽ chuyển người dùng đến trang ***“manager/addProduct.jsp”.***
    2. Tại trang ***“manager/addProduct.jsp”,*** khi người dùng nhấn vào nút **“Thêm”** sẽ gửi thông tincủa sản phẩm từ form đến cho class ***“ProductController.java” .***
    3. Tại class ***“ProductController.java”,*** sẽ khởi tạo đối tượng “***Product”*** và gọi phương thức “***checkEMail***” và phương thức “***insertUser***” từ class ***“UserDAO.java”.***
    4. Phương thức “***checkEmali***” sẽ kiểm tra email đã tồn tại hay chưa, đồng thời kiểm tra lại mật khẩu nhập lại có đúng hay không.Nếu tất cả đã hợp lệ thị class ***“UerController.java”*** sẽ chèn thông tin nguời dùng vào bảng cơ sở dữ liệu “user” và trả về trang “***pages/login.jsp”.*** Ngược lại, thông báo lỗi cho nguời dùng.
* **Xóa sản phẩm**
* **Cập nhật thông tin sản phẩm**
  + 1. **Chức năng quản lí đơn hàng**
    2. **Chức năng quản lí tin tức**
    3. **Chức năng quản lí bình luận**

1. **ĐÁNH GIÁ KIỂM THỬ**

**PHẦN 3 : KẾT LUẬN**

**CÁC CÔNG CỤ HỖ TRỢ**

* Netbeans IDE 8.0.2
* XAMPP v3.2.1: sử dụng Apache, Mysql, Tomcat
* Tortoise v2.0.0.0

**TÀI LIỆU THAM KHẢO**