

Mục lục

1	Giới thiệu	2
1.1	Giới thiệu về Servlet	2
1.2	Ưu điểm của Servlet	3
1.3	Công nghệ JSP	4
1.3.1	Ưu điểm của JSP	5
1.3.2	Đặc điểm của JSP	5
1.4	Mô hình MVC	5
2	Mã nguồn Website	6
2.1	Mục đích của Website	6
2.2	Chức năng của Website	6
2.3	Xây dựng Website	8
2.3.1	Cơ sở dữ liệu	8
2.3.2	Xây dựng các lớp (Servlet, Bean,...)	8
2.3.3	Giao diện Website	11
3	Danh mục tham khảo	11

Java Web sử dụng Servlet/JSP, JDBC

Website bán thực phẩm online

Bùi Quang Đức - 20173023
Hanoi University of Science and Technology
Ngày 17 tháng 12 năm 2019

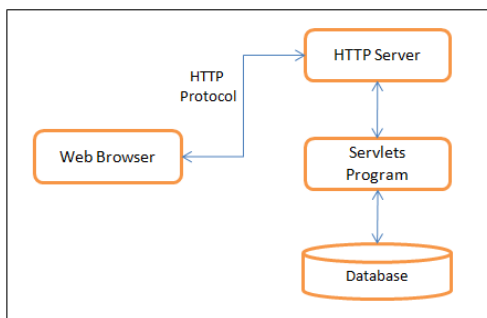
Tóm tắt nội dung

Java là một ngôn ngữ lập trình được Sun Microsystems giới thiệu vào tháng 6 năm 1995. Từ đó, nó đã trở thành một công cụ lập trình của các lập trình viên chuyên nghiệp. Java được xây dựng trên nền tảng của C và C++, do đó nó sử dụng các cú pháp của C và các đặc trưng hướng đối tượng của C++. Java là một ngôn ngữ đơn giản, hướng đối tượng, phân tán, thông dịch mạnh mẽ, bảo mật, cấu trúc độc lập, khả chuyển, hiệu quả cao và linh động. Đặc biệt Java rất nhất quán và dễ dàng trong lập trình web, xử lý sự kiện với nhiều hệ cơ sở dữ liệu khác nhau. Website được tạo lập bằng Java Servlet/JSP, dưới sự chỉ dẫn của thầy Nguyễn Bình Minh, sẽ chỉ rõ thêm về Java Web cũng như ứng dụng để tạo nên một sản phẩm phục vụ cho nhu cầu đời sống hiện nay.

1 Giới thiệu

1.1 Giới thiệu về Servlet

Hiện nay, trong lập trình có một xu hướng rất quan trọng đang được tập trung và phát triển ứng dụng, đó là xây dựng các chương trình dịch vụ Java ở phía máy chủ (Server). Servlet là thành phần chính được sử dụng để phát triển các chương trình dịch vụ Java ở phía máy chủ. Các Servlet là các chương trình Java thực hiện ở các ứng dụng Server (tên gọi Servlet cũng gần giống như Applet ở phía máy Client) để trả lời cho các yêu cầu của Client. Các Servlet không bị ràng buộc chặt với một giao thức Client-Server cụ thể nào cả, nhưng giao thức thường được sử dụng là HTTP, do vậy khi nói tới Servlet nghĩa là nói tới HTTP Servlet.



Hình 1: Vai trò của Servlet.

Servlet là sự phát triển mở rộng của CGI để đảm bảo Server thực hiện được các chức năng của mình. Ta có thể sử dụng Servlet của Java để tùy chỉnh lại một dịch vụ bất kỳ như Web Server, Mail Server, etc.

Web Server hiện thị các tư liệu được viết trong HTML và hồi đáp cho yêu cầu của người sử dụng qua HTTP. Các tư liệu HTML chứa các văn bản được đánh dấu (định dạng) để các trình duyệt như IE, Firefox, Chrome đọc được. Một trình duyệt chấp nhận đầu vào ở dạng HTML, khi được sử dụng nhấn một nút để yêu cầu một

số thông tin nào đó, một Servlet đơn giản được gọi để xử lý các yêu cầu đó. Các công việc chính của servlet được mô tả khái quát bao gồm:

- Đọc các dữ liệu tường minh được Client gửi đến từ các yêu cầu (dữ liệu theo khuôn dạng - form data).
- Đọc các dữ liệu không tường minh được Client gửi đến từ các yêu cầu (dữ liệu trong phần đầu của yêu cầu - request header).
- Xử lý và lưu trữ các dữ liệu được cung cấp dưới dạng HTML.
- Gửi trả lời dữ liệu tường minh cho Client (dạng HTML), cung cấp các nội dung động, ví dụ trả lời yêu cầu Client về các câu truy vấn vào các Databases.
- Quản lý các thông tin trạng thái và trả lời dữ liệu không tường minh cho Client (các mã trạng thái và các thành phần của trả lời).

Các Servlet cũng được sử dụng thay cho kịch bản giao diện cổng chung CGI Script. Khi tạo ra một trang Web, ta cũng sẽ tạo ra một ứng dụng Web. Trước khi sử dụng Servlet để tạo ra các ứng dụng Web, chúng ta đi tìm hiểu xem có những khả năng lựa chọn nào khác để phát triển những ứng dụng Web.

- CGI: Theo cách thông thường, để bổ sung các chức năng vào cho một Web Server, người ta hay sử dụng Common Gateway Interface (CGI), một giao diện độc lập với ngôn ngữ cho phép một Server khởi động một tiến trình ngoại để nhận thông tin được yêu cầu thông qua các biến môi trường. Mỗi yêu cầu được trả lời bởi một tiến trình riêng thông qua một đại diện riêng của một chương

trình CGI hoặc bởi một kịch bản CGI (thường được viết bằng ngôn ngữ thông dịch như Perl).

- Fast CGI: Open Market đã phát triển một chuẩn khác thay cho CGI được gọi là Fast CGI. Fast CGI hành động giống như CGI. Nó khác ở chỗ, Fast CGI tạo ra một tiến trình bền vững cho từng chương trình.
- ASP, JavaScript: ASP được Microsoft phát triển để tạo ra các nội dung cho trang Web động. Trong ASP, trang HTML có thể nhúng những phần nhỏ được viết bằng VBScript hoặc JScript. Netscape đưa ra kỹ thuật được gọi là JavaScript, cho phép đưa các phần mã lệnh nhỏ nhúng vào trang HTML, nhằm tạo ra nội dung Web động một cách linh hoạt hơn. Ngoài ra, Netscape còn cung cấp NSAPI, Microsoft đưa ra ISAPI cho các Web Server của họ.

Servlet có một số ưu điểm so với CGI:

- Một Servlet không làm việc trong một tiến trình riêng. Điều này loại bỏ việc phải tạo ra quá nhiều tiến trình mới cho mỗi yêu cầu.
- Một Servlet sẽ thường trực trong bộ nhớ giữa các yêu cầu, trong khi các chương trình CGI cần phải tải xuống và được khởi động cho từng yêu cầu CGI.
- Chỉ cần một Servlet trả lời đồng thời cho tất cả các yêu cầu. Điều này cho phép tiết kiệm được bộ nhớ và đảm bảo nó dễ dàng quản lý được dữ liệu một cách thống nhất.
- Một Servlet có thể thực hiện bởi một Servlet Engine trong phạm vi kiểm soát Sandbox để đảm bảo an toàn trong việc sử dụng Servlet.

Các lớp Servlet của Java thực hiện bên trong JVM, chúng được đảm bảo an toàn và chuyển đổi tương thích giữa các hệ điều hành và giữa các Server với nhau. Điều này khác với các Applet, Servlet chỉ thao tác được trong miền của một Server. Servlet API được phát triển dựa trên những điểm mạnh của Java Platform nhằm giải quyết vấn đề của CGI và Server API. Nó là một API đơn giản, hỗ trợ tất cả các Web Server và thậm chí cho phép các ứng dụng máy chủ dùng để kiểm tra và quản lý các công việc trên Server. Nó giải quyết vấn đề thực thi bằng việc thực hiện tất cả các yêu cầu như các luồng Thread trong quá trình xử lý, hoặc việc cân bằng tải trên một Server trong các cụm máy tính Cluster. Các Servlet dễ dàng chia sẻ tài nguyên với nhau. Trong định nghĩa Servlet, vấn đề bảo mật được cải tiến theo nhiều cách. Trước hết, bạn hiếm khi thực thi được các câu lệnh trên Shell với dữ liệu cung cấp bởi người dùng mà Java API đã cung cấp với những khả năng truy cập đến tất cả các hàm thông dụng. Bạn có thể sử dụng Java Mail để đọc và gửi mail, kết nối vào các database (thông qua JDBC), tệp lớn (.class) và những lớp liên quan để truy cập hệ thống tệp, database, RMI, CORBA, Enterprise Java Beans (EJB), etc.

1.2 Ưu điểm của Servlet

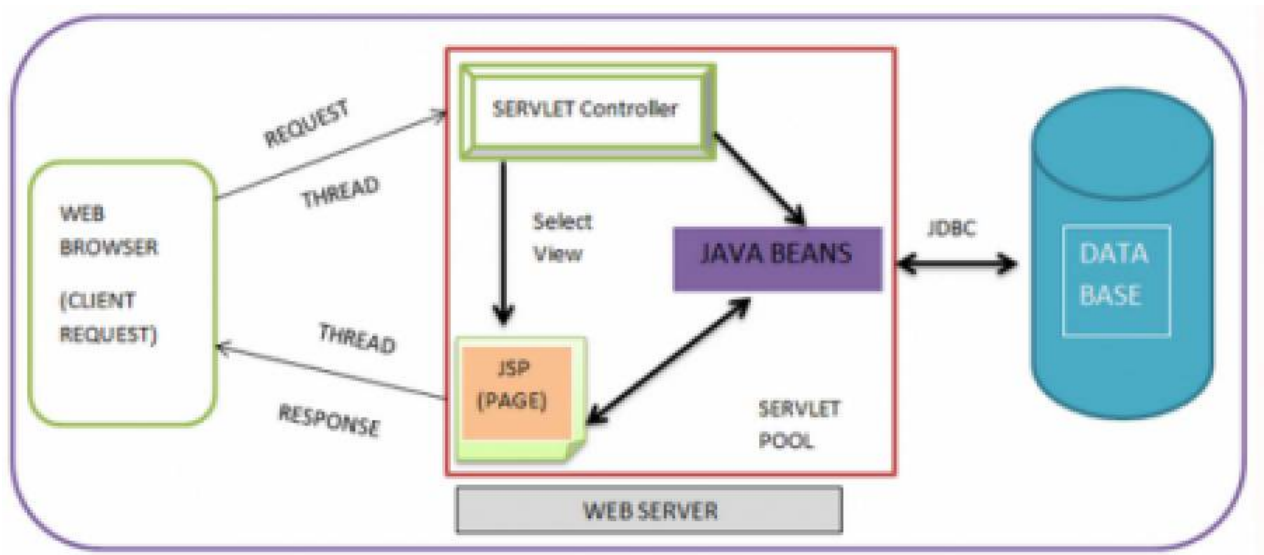
Servlet được sử dụng để thay thế cho những công nghệ Web động. Việc sử dụng Servlet mang lại những lợi thế:

1. Dễ di chuyển: Servlet được viết bằng Java nên nó có tính di động cao, thực hiện được trên nhiều hệ điều hành, trên các Web Server khác nhau. Khái niệm "Write once, run everywhere" cũng rất đúng với Servlet.

2. Mạnh mẽ: Servlet hỗ trợ rất hiệu quả cho việc sử dụng các giao diện lỗi API như lập trình mạng, xử lý đa luồng, xử lý ảnh, nén dữ liệu, kết nối các database, bảo mật, xử lý phân tán và triệu gọi từ xa RMI, CORBA, etc. Nó cũng thích hợp để trao đổi thông tin, truyền thông giữa Client và Server một cách bình thường.
3. Hiệu quả: Servlet có tính hiệu quả cao. Một khi được tải về, nó sẽ được lưu lại trong bộ nhớ của máy chủ. Servlet duy trì các trạng thái của nó, do vậy, những tài nguyên ngoại như việc kết nối với database cũng sẽ được lưu giữ lại.
4. An toàn: bởi vì Servlet được viết bằng Java nên nó kết thừa được tính an toàn của Java. Cơ chế tự động dọn rác và việc không sử dụng con trỏ của Java giúp cho Servlet thoát khỏi nhiều công việc quản lý bộ nhớ. Đồng thời nó xử lý các lỗi rất an toàn theo cơ chế xử lý ngoại lệ của Java.
5. Tính tích hợp: các Servlet được tích hợp với các Server. Chúng cộng tác với các Server tốt hơn các chương trình CGI.
6. Tính linh hoạt: các Servlet hoàn toàn mềm dẻo. Một HTTP Servlet được sử dụng để tạo ra một trang Web, sau đó ta có thể sử dụng thẻ <Servlet> để đưa nó về trang Web tĩnh, hoặc sử dụng với các Servlet khác để lọc ra các nội dung cần thiết.

1.3 Công nghệ JSP

Công nghệ JSP (Java Server Pages): nhằm mục đích phân tách giữa tầng presentation và tầng business logic trong Servlets. Mục đích để hiển thị, bản chất sẽ được build thành Servlets.



Hình 2: Mô hình MVC với Servlet/JSP.

JSP là tài liệu text có thể trả về static (HTML, XML, Text) và dynamic (Java scriptlet, JavaBeans) content cho trình duyệt.

1.3.1 Ưu điểm của JSP

1. Tách biệt nội dung và cách trình bày
2. Đơn giản hóa việc phát triển ứng dụng Web với JSP, JavaBeans và Custom Tags.
3. Hỗ trợ tái sử dụng phần mềm qua các components (JavaBeans, Custom Tags).
4. Tự động triển khai, tự động biên dịch lại trang JSP khi có thay đổi.
5. Độc lập platform.
6. Dễ dàng hơn cho người thiết kế (nhờ có thư viện JSTL hỗ trợ).

1.3.2 Đặc điểm của JSP

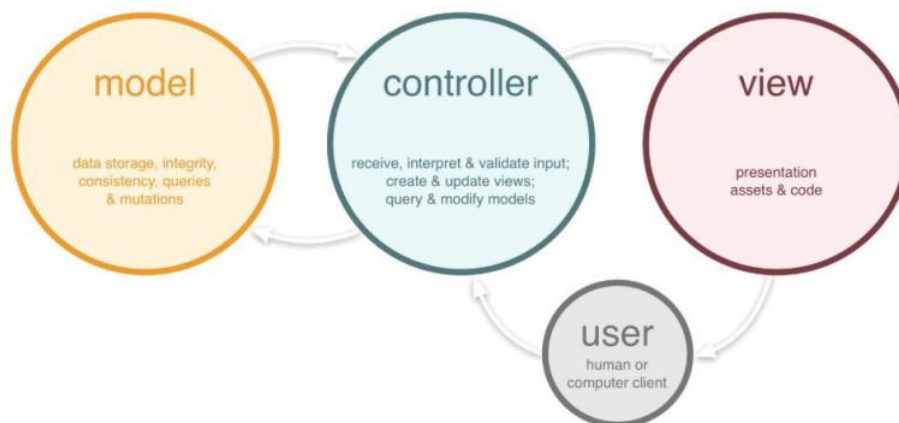
Các trang JSP được dịch thành Servlet. Vì vậy cho phép chèn mã nguồn Java vào Servlet được sinh tương ứng cho trang JSP, tuy nhiên nên giảm thiểu sử dụng JSP Scripting Elements trong trang JSP nếu có thể. Có 3 dạng JSP Scripting Elements:

- Expressions: `<%=Expressions%>`
- Scriptlets: `<%Code%>`
- Declarations: `<%!Declarations%>`

1.4 Mô hình MVC

MVC được viết tắt bởi Model - View - Controller. Nó tạo ra 3 tầng tách biệt và tương tác với nhau:

- Model: chứa những nghiệp vụ tương tác với dữ liệu hoặc hệ quản trị cơ sở dữ liệu, gồm các class/function xử lý nhiều nghiệp vụ như kết nối database, truy vấn dữ liệu, cập nhật, xóa dữ liệu.
- View: chứa những giao diện như nút bấm, khung nhập, menu, hình ảnh, etc. Có nhiệm vụ hiển thị dữ liệu và giúp người dùng tương tác với hệ thống.



Hình 3: Giải thích mô hình MVC.

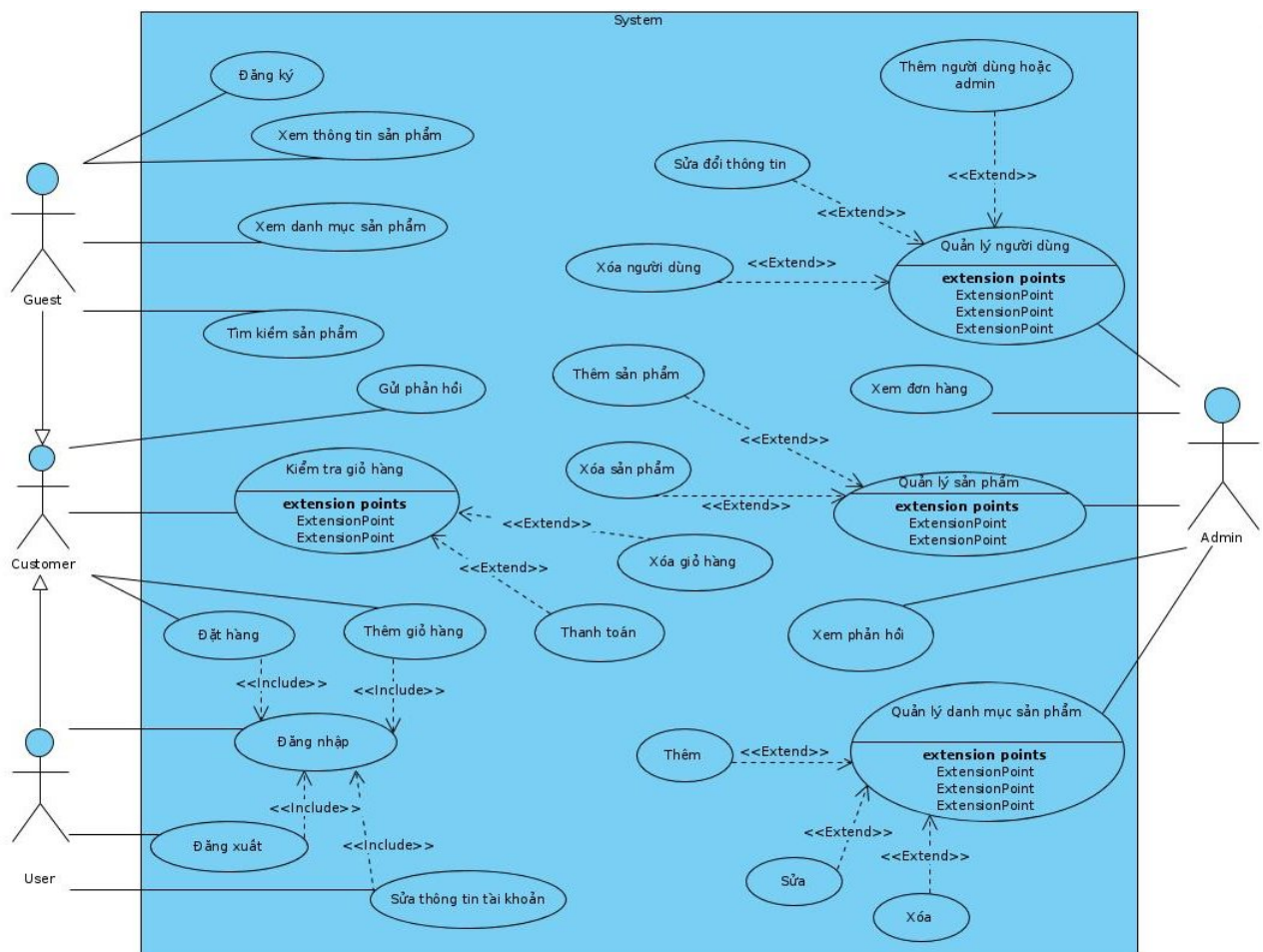
- Controller: tiếp nhận những yêu cầu xử lý được gửi từ người dùng, gồm các class/function xử lý nhiều nghiệp vụ logic để lấy dữ liệu thông tin cần thiết nhờ các nghiệp vụ Model cung cấp và hiển thị dữ liệu cho người dùng nhờ View.

2 Mã nguồn Website

2.1 Mục đích của Website

Ngày nay, chúng ta đang sống trong một thế giới mà công nghệ thông tin đang tác động lên hầu hết mọi lĩnh vực từ kinh tế cho đến chính trị, văn hóa, xã hội,... Nó giúp cho chất lượng cuộc sống con người tốt hơn. Việc thành lập Website bán hàng nhằm tìm hiểu về Java Servlet/JSP cũng như tạo lập một nơi bán hàng trực tiếp giúp đỡ cho mọi người mua hàng một cách thuận tiện và chính xác.

Sơ đồ usecase tổng quát:



Hình 4: Usecase UML.

2.2 Chức năng của Website

- Đối với khách vắng lai (guest)
 - Xem danh sách sản phẩm, xem thông tin chi tiết sản phẩm
 - Tìm kiếm sản phẩm theo tên, danh mục
 - Xem thống kê sản phẩm theo danh mục

- Đăng ký tài khoản
- Đối với khách hàng đã đăng nhập tài khoản (User)
 - Có tài khoản, có thể chỉnh sửa tài khoản, đăng nhập, đăng xuất
 - Xem thông tin tài khoản
 - Đặt mua sản phẩm (đặt hàng ngay hoặc đặt theo số lượng)
 - Xem đơn hàng, xóa đơn hàng
 - Thanh toán hóa đơn
 - Xem lịch sử hóa đơn
 - Gửi phản hồi cho Website
- Đối với admin/manager
 - Đăng nhập website
 - Xem, thêm, sửa xóa thông tin của sản phẩm, tài khoản, danh mục sản phẩm
 - Xem thông tin phản hồi, lịch sử hóa đơn

2.3 Xây dựng Website

Các công nghệ và công cụ sử dụng trong Website:

- JDBC, MariaDB, Glassfish Server
- Java Core, Java Servlet, JSP, JSTL, HTML, CSS, JavaScript
- Tool: MySQL Workbench, NetBeans
- Library: mysql-connector, jstl, commons-fileupload, commons-io

2.3.1 Cơ sở dữ liệu

Ta có lược đồ cơ sở dữ liệu. Giỏ hàng (cart) ta lưu vào session cho mỗi phiên.



Hình 5: Database Diagram.

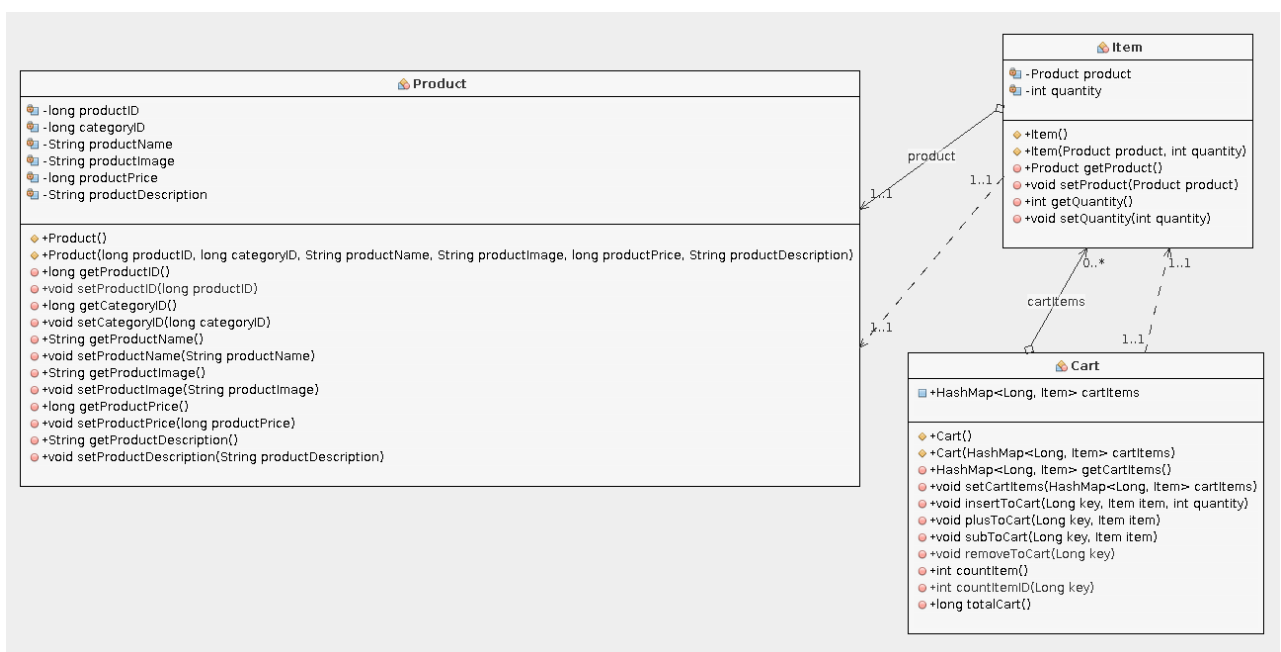
2.3.2 Xây dựng các lớp (Servlet, Bean,...)

- project.one.connect
 - ConnectionUtils.java: kết nối database bằng JDBC.
 - Test.java: kiểm tra kết nối tới database.
- project.one.controller
 - AdminCategoryServlet.java
 - AdminProductServlet.java

- AdminUserAccountServlet.java
- AdminProductServlet.java
- BillServlet.java
- CartServlet.java
- CategoryServlet.java
- CheckOutServlet.java
- ContactServlet.java
- FileUploadServlet.java: xử lý tác vụ upload sản phẩm Website (lưu thông tin vào database và lưu ảnh sản phẩm vào thư mục WebContainer của Server)
- IndexServlet.java: xử lý phần khởi tạo Servlet trước khi chạy index.jsp để có thể hiển thị ngay sản phẩm.
- ProductServlet.java
- UserServlet.java

- project.one.model

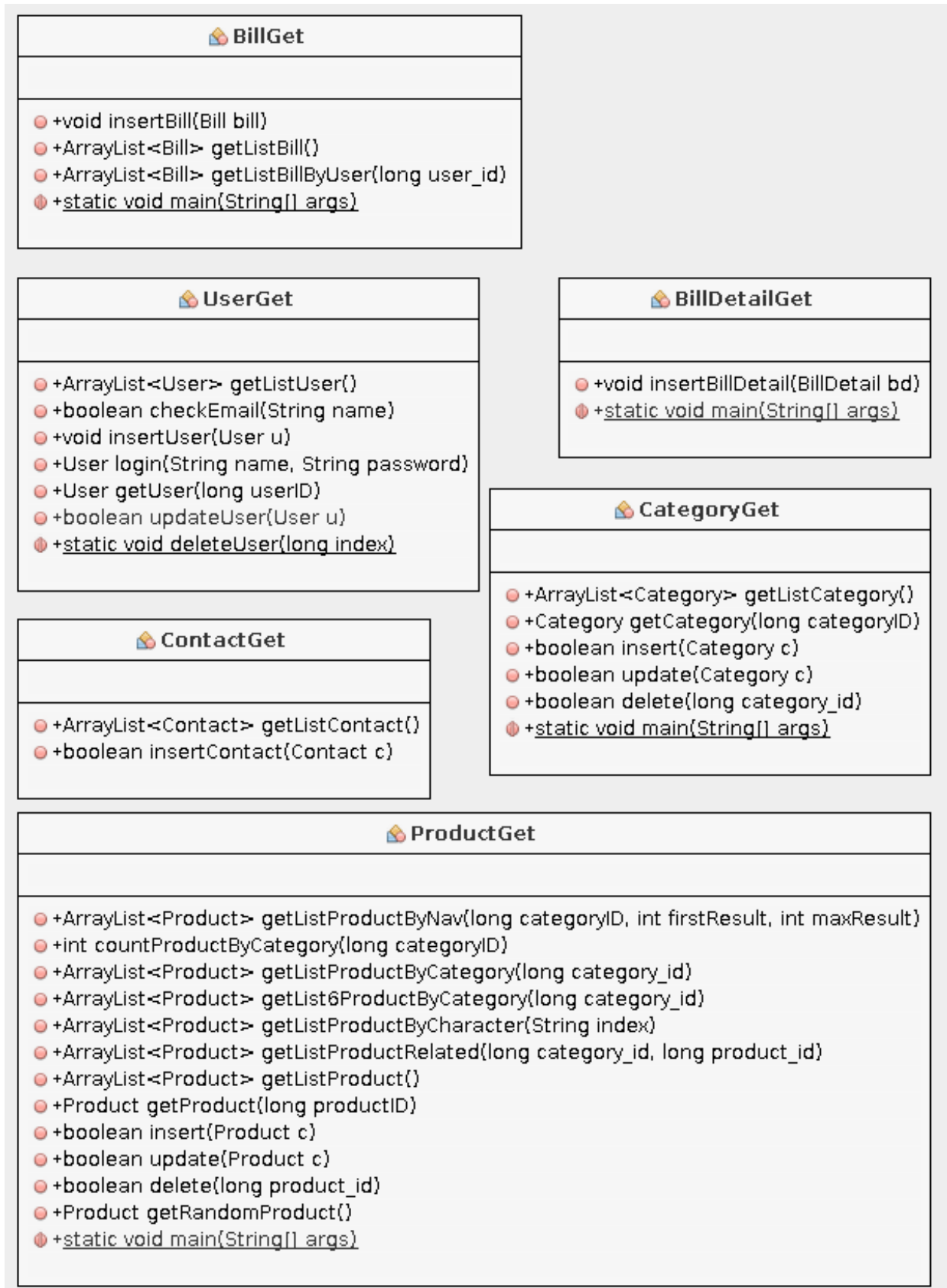
- Bill.java
- BillDetail.java
- Cart.java



Hình 6: Cart UML.

- Category.java
- Contact.java
- Item.java
- Product.java
- User.java

- project.one.utils

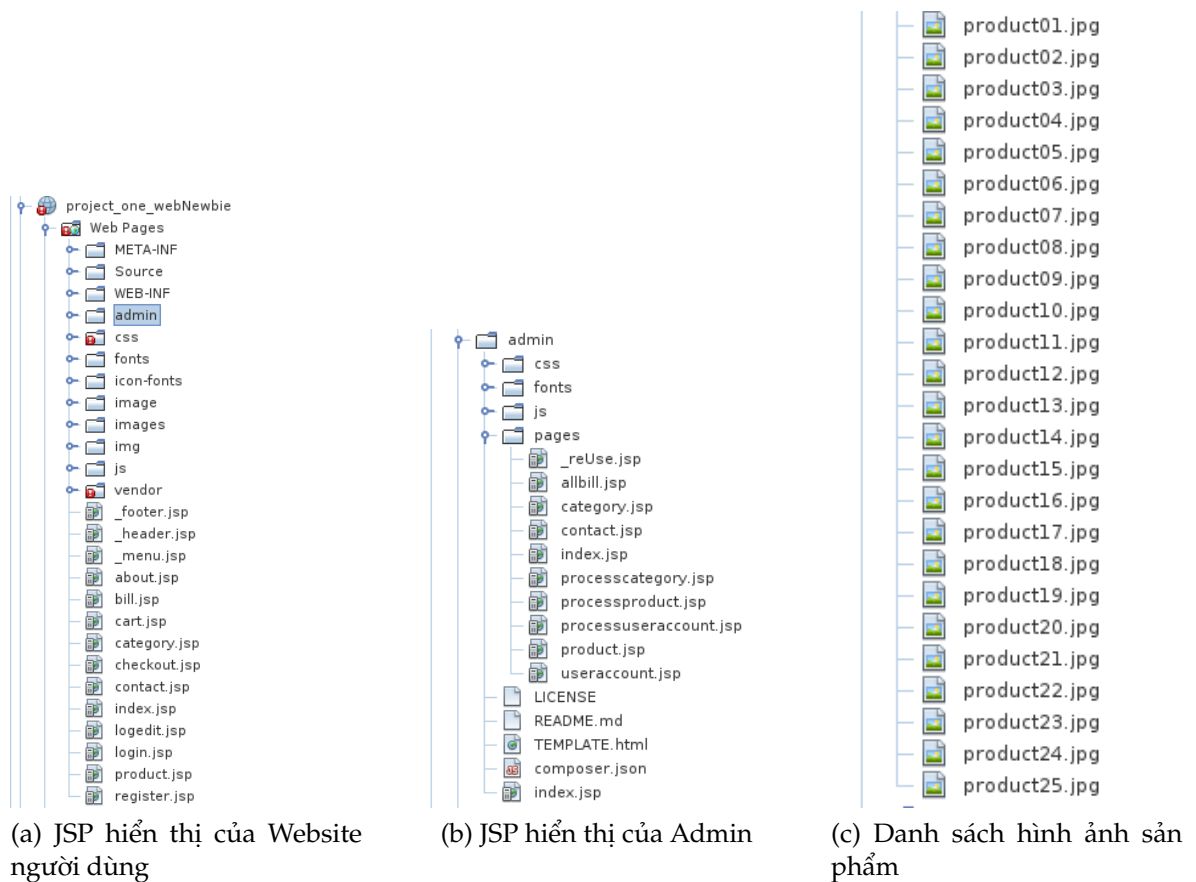


Hình 7: Utils UML.

- BillDetailGet.java
- BillGet.java
- CategoryGet.java
- ContactGet.java
- ProductGet.java
- UserGet.java

2.3.3 Giao diện Website

Ta hiển thị tầng View giao diện của Website



Hình 8: Danh sách các JSP thuộc tầng View

3 Danh mục tham khảo

1. <https://timoday.edu.vn/tu-phan-tich-thiet-ke-uml-den-viet-code-theo-mo-hinh-mvc-ung-dung-xay-dung-website-thuong-mai-dien-tu-voi-cong-nghe-java-jsp-servlet>
2. <https://docs.oracle.com/javaee/7/tutorial/servlets016.htm>
3. <https://o7planning.org/vi/10979/java-servlet-jsp>