**BỘ CÔNG THƯƠNG**

**TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP TP.HCM**



**BÁO CÁO BÀI TẬP LỚN LẬP TRÌNH WWW (JAVA)**

Website giới thiệu, bán loa và phụ kiện âm thanh trực tuyến

***Sinh viên thực hiện:***

1. 20027751 – Nguyễn Thị Thanh Hà

GVHD:Nguyễn Minh Hải

TP.Hồ Chí Minh - Năm 2024

# MỤC LỤC

[MỤC LỤC 1](#_Toc14693502)

[DANH MỤC CÁC HÌNH VẼ 1](#_Toc14693503)

[DANH MỤC CÁC BẢNG BIỂU 1](#_Toc14693504)

[CHƯƠNG 1 : GIỚI THIỆU 1](#_Toc14693505)

[1.1 Tổng quan 1](#_Toc14693506)

[1.2 Mục tiêu đề tài 1](#_Toc14693507)

[1.3 Phạm vi đề tài 1](#_Toc14693508)

[1.4 Mô tả yêu cầu chức năng 1](#_Toc14693509)

[CHƯƠNG 2 : CƠ SỞ LÝ THUYẾT 1](#_Toc14693510)

[2.1 Cloud Computing với AWS 1](#_Toc14693511)

[2.2 Node.js 1](#_Toc14693512)

[2.3 MongoDB / DynamoDB 1](#_Toc14693513)

[CHƯƠNG 3 : PHÂN TÍCH VÀ THIẾT KẾ 1](#_Toc14693514)

[3.1 Phân tích yêu cầu bằng UML 1](#_Toc14693515)

[3.1.1 Usecase tổng quát 1](#_Toc14693516)

[3.1.2 Danh sách tác nhân và mô tả 1](#_Toc14693517)

[3.1.3 Danh sách các tình huống hoạt động (Use cases) 1](#_Toc14693518)

[3.1.4 Tình huống hoạt động 1](#_Toc14693519)

[3.2 Class diagram 1](#_Toc14693520)

[3.3 Deployment diagram 1](#_Toc14693521)

[CHƯƠNG 4 : HIỆN THỰC 1](#_Toc14693522)

[4.1 Cấu hình phần cứng, phần mềm 1](#_Toc14693523)

[4.2 Giao diện của hệ thống 1](#_Toc14693524)

[CHƯƠNG 5 : KẾT LUẬN 1](#_Toc14693525)

[5.1 Kết quả đạt được 1](#_Toc14693526)

[5.2 Hạn chế của đồ án 1](#_Toc14693527)

[5.3 Hướng phát triển 1](#_Toc14693528)

[TÀI LIỆU THAM KHẢO 1](#_Toc14693529)

# DANH MỤC CÁC HÌNH VẼ

[Hình 2‑1 Node.js development 1](#_Toc14693413)

# DANH MỤC CÁC BẢNG BIỂU

[Bảng 3‑1 Danh sách tác nhân và mô tả 1](#_Toc14682192)

[Bảng 3‑2 Danh sách các tình huống trong hệ thống 1](#_Toc14682193)

# : GIỚI THIỆU

## Tổng quan

Kinh doanh bán hàng trực tuyến ngày càng trở thành xu hướng hiện nay. Không chỉ cá nhân, ngay cả những doanh nghiệp, cửa hàng thiết bị âm thanh khi đứng trước áp lực cạnh tranh ngày càng gay gắt cũng chuyển mình bằng cách thiết kế web bán hàng nói chung và thiết kế web bán loa và thiết bị âm thanh nói riêng nhằm khai thác tối đa thị trường trực tuyến đầy tiềm năng.

Giờ đây, không chỉ cá nhân mà ngay cả những doanh nghiệp, cửa hàng thiết bị âm thanh đứng trước áp lực cạnh tranh ngày càng gay gắt này đòi hỏi phải chuyển mình bằng việc thiết kế web bán hàng nói chung và thiết kế web bán loa và thiết bị âm thanh nói riêng nhằm khai thác tối đa thị trường kinh doanh trực tuyến đầy tiềm năng này.

Với sự phát triển của internet, ngành thiết bị âm thanh trở nên phát triển mạnh mẽ, giúp doanh nghiệp, cá nhân tiếp cận khách hàng cực kỳ lớn. Những người yêu thích âm nhạc, các nhà sản xuất âm nhạc, và các công ty tổ chức sự kiện ngày càng mua sắm thiết bị âm thanh online nhiều hơn. Đó là lý do tại sao mỗi ngày hàng trăm cửa hàng thiết bị âm thanh ra đời và đặc biệt là cửa hàng âm thanh trực tuyến. Từ đó, nhóm giới thiệu đến website bán loa và thiết bị âm thanh trực tuyến.

## Mục tiêu đề tài

Xây dựng một website hiện đại, dễ sử dụng, chuyên nghiệp và tối giản để giới thiệu và bán các sản phẩm loa và phụ kiện âm thanh trực tuyến. Website sẽ giúp khách hàng dễ dàng tìm kiếm, xem thông tin chi tiết, đánh giá và mua các sản phẩm âm thanh chất lượng.

## Phạm vi đề tài

* Đồ án được thực hiện dựa trên cơ sở phạm vi môn học Lập trình WWW:
* Dùng công nghệ:
* MySQL Server
* Spring Security
* Spring MVC
* Đồ án thực hiện những mục tiêu:
* Đăng ký tài khoản
* Xem danh sách sản phẩm
* Xem chi tiết của từng sản phẩm từ danh sách sản phẩm
* Thêm sản phẩm vào giỏ hàng
* Xem giỏ hàng
* Xử lý thanh toán
* Tìm kiếm thông tin về sản phẩm/loại sản phẩm, tài khoản người dùng, các đơn đặt sản phẩm
* Quản lý thông tin sản phẩm/loại sản phẩm
* Quản lý thông tin đơn hàng trực tuyến

## Mô tả yêu cầu chức năng

* **Quản lý tài khoản người dùng:**
  + Cho phép người dùng đăng ký, đăng nhập và quản lý thông tin tài khoản.
  + Cho phép người dùng thay đổi mật khẩu và cập nhật thông tin cá nhân.
* **Quản lý sản phẩm:**
  + Hiển thị danh mục sản phẩm loa và phụ kiện âm thanh.
  + Cung cấp công cụ tìm kiếm và lọc sản phẩm theo các tiêu chí khác nhau.
  + Hiển thị chi tiết sản phẩm bao gồm mô tả, thông số kỹ thuật, hình ảnh, giá cả và đánh giá từ người dùng.
* **Giỏ hàng và thanh toán:**
  + Cho phép người dùng thêm, xóa và cập nhật sản phẩm trong giỏ hàng.
  + Cung cấp quy trình thanh toán an toàn và dễ dàng, hỗ trợ nhiều phương thức thanh toán.
  + Gửi thông báo xác nhận đơn hàng qua email.
* **Theo dõi đơn hàng:**
  + Cung cấp chức năng để người dùng theo dõi trạng thái đơn hàng của họ.
  + Gửi thông báo cập nhật trạng thái đơn hàng qua email.
* **Quản lý đánh giá sản phẩm:**
  + Cho phép người dùng đánh giá và nhận xét về sản phẩm.
  + Hiển thị đánh giá và nhận xét từ người dùng khác để giúp khách hàng có cái nhìn tổng quan về sản phẩm.
* **Quản lý nội dung:**
  + Cho phép quản trị viên thêm, chỉnh sửa và xóa sản phẩm.
  + Quản lý danh mục sản phẩm, bài viết, tin tức và các trang thông tin khác.

# : CƠ SỞ LÝ THUYẾT

## Spring

A green text on a black background

Description automatically generated

* Các tính năng cốt lõi của Spring Framework có thể được sử dụng để phát triển bất kỳ ứng dụng Java nào, nhưng có những phần mở rộng để xây dựng các ứng dụng web trên nền tảng Java EE. Spring framework nhắm mục tiêu làm cho việc phát triển J2EE dễ sử dụng hơn và thúc đẩy các phương pháp lập trình tốt bằng cách cho phép mô hình lập trình dựa trên POJO( Plain Old Java Object).
* SpringMVC

MVC là một trong những modules quan trọng nhất của Spring Framework. Nó xây dựng trên bộ chứa Spring IoC mạnh mẽ và sử dụng rộng rãi các tính năng của container để đơn giản hóa cấu hình của nó.

Model-View-Controller (MVC) là một mẫu thiết kế phổ biến trong thiết kế UI. Nó tách rời logic business từ các UI bằng cách tách biệt các vai trò của model, views và controller trong một ứng dụng. Các models có trách nhiệm đóng gói dữ liệu ứng dụng cho các views để trình bày. Views chỉ nên trình bày dữ liệu này, không bao gồm bất kỳ logic business nào.Controller có trách nhiệm nhận yêu cầu từ người dùng và gọi các services back-end cho gia công kinh doanh. Sau khi xử lý, các services back-end có thể trả về một số dữ liệu để hiển thị. Controller thu thập dữ liệu này và chuẩn bị các models để trình bày. Ý tưởng cốt lõi của mẫu MVC là tách logic business khỏi các UI để cho phép chúng thay đổi độc lập mà không ảnh hưởng lẫn nhau.

Trong một ứng dụng Spring MVC, các models thường bao gồm các domain object được xử lý bởi lớp service và được duy trì bởi persistence layer. Các views thường là các templates JSP được viết bằng Java Standard Tag Library (JSTL). Tuy nhiên, cũng có thể định nghĩa các chế độ xem là file PDF, file Excel, service web RESTful, hoặc thậm chí giao diện Flex, giao diện cuối cùng thường được mệnh danh Internet applications (RIAs) phong phú.

Sau khi kết thúc chương này, bạn sẽ có thể phát triển các ứng dụng web Java bằng Spring MVC. Bạn cũng sẽ hiểu các controller chung và view types của Spring MVC, bao gồm cả những gì đã trở thành việc sử dụng các chú thích để tạo controller trên thực tế kể từ khi phát hành Spring 3.0. Hơn nữa, bạn sẽ hiểu các nguyên tắc cơ bản của Spring MVC, sẽ là nền tảng cho các chủ đề nâng cao hơn được đề cập trong các chương sắp tới.

* SpringSecurity

Spring Security cung cấp các dịch vụ bảo mật toàn diện cho các ứng dụng doanh nghiệp có nền tảng Java EE.

Spring Security cung cấp 2 cơ chế cơ bản:

* Authentication (xác thực): là tiến trình thiết lập một principal. Principal có thể hiểu là một người, hoặc một thiết bị, hoặc một hệ thống nào đó có thể thực hiện một hành động trong ứng dụng của bạn.
* Authorization (phân quyền) hay Access-control: là tiến trình quyết định xem một principal có được phép thực hiện một hành động trong ứng dụng của bạn hay không. Trước khi diễn tiến tới Authorization, principal cần phải được thiết lập bởi Authentication.
* Các thành phần cốt lõi:
* Security, SecurityContext và Authentication

SecurityContext là interface cốt lõi của Spring Security, lưu trữ tất cả các chi tiết liên quan đến bảo mật trong ứng dụng. Khi chúng ta kích hoạt Spring Security trong ứng dụng thì SecurityContext cũng sẽ được kích hoạt theo.

Chúng ta sẽ không truy cập trực tiếp vào SecurityContext, thay vào đó sẽ sử dụng lớp SecurityContextHolder. Lớp này lưu trữ security context hiện tại của ứng dụng, bao gồm chi tiết của principal đang tương tác với ứng dụng. Spring Security sẽ dùng một đối tượng Authentication để biểu diễn thông tin này.

* UserDetails và UserDetailsService

Trong đoạn code trên, chúng ta có được một principal từ đối tượng Authentication. Principal đơn giản chỉ là một đối tượng và sẽ được ép kiểu sang UserDetails.

UserDetails là một interface cốt lõi của Spring Security. Nó đại diện cho một principal nhưng theo một cách mở rộng và cụ thể hơn:

* getAuthorities(): trả về danh sách các quyền của người dùng.
* getPassword(): trả về password đã dùng trong qúa trình xác thực.
* getUsername(): trả về username đã dùng trong qúa trình xác thực
* isAccountNonExpired(): trả về true nếu tài khoản của người dùng chưa hết hạn.
* isAccountNonLocked(): trả về true nếu người dùng chưa bị khóa.
* isCredentialsNonExpired(): trả về true nếu chứng thực (mật khẩu) của người dùng chưa hết hạn.
* isEnabled(): trả về true nếu người dùng đã được kích hoạt GrantedAuthority.

- Phương thức getAuthorities() sẽ trả về một tập hợp các đối tượng GrantedAuthority. Một GrantedAuthority là một quyền được ban cho principal. Các quyền đều có tiền tố là ROLE\_, ví dụ như ROLE\_ADMIN, ROLE\_MEMBER...

## HTML

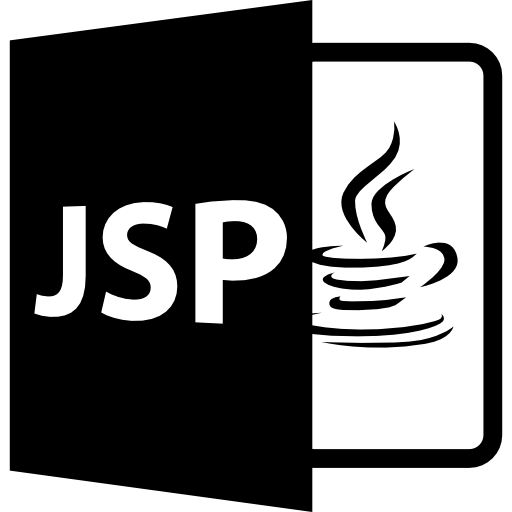
A logo of a software developer

Description automatically generated

[**HTML**](https://topdev.vn/viec-lam-it/html-kt75) là viết tắt của cụm từ**Hypertext Markup Language**(tạm dịch là Ngôn ngữ đánh dấu siêu văn bản). **HTML** được sử dụng để tạo và cấu trúc các thành phần trong trang web hoặc ứng dụng, phân chia các đoạn văn, heading, titles, blockquotes

Một tài liệu HTML được hình thành bởi các phần tử HTML (HTML Elements) được quy định bằng các cặp thẻ (tag và attributes).

## JSP



**JSP là viết tắt của JavaServer Pages**là một công nghệ để phát triển các trang web động. JSP giúp các nhà phát triển chèn java code vào các trang HTML bằng cách sử dụng các thẻ JSP đặc biệt.

**JSP** là một kiểu Java servlet được thiết kế để tạo ra giao diện người dùng cho một ứng dụng Java web. Các nhà phát triển web viết các JSP như các tệp văn bản kết hợp mã HTML hoặc XHTML, các phần tử XML, các action và lệnh JSP.

Sử dụng JSP, bạn có thể thu thập dữ liệu đầu vào từ người dùng thông qua các Form của trang web, trình bày các bản ghi từ một cơ sở dữ liệu hoặc một nguồn khác, và tạo các trang web động.

Các thẻ JSP có thể được sử dụng cho nhiều mục đích khác nhau, chẳng hạn như truy xuất thông tin từ cơ sở dữ liệu hoặc đăng ký mới, truy cập các thành phần JavaBeans, kiểm soát giữa các trang và chia sẻ thông tin giữa các request, các trang vv

# : PHÂN TÍCH VÀ THIẾT KẾ

## Phân tích yêu cầu bằng UML

### Usecase tổng quát

***A diagram of a company

Description automatically generated***

### Danh sách tác nhân và mô tả

Danh sách tác nhân và mô tả xem ở Bảng 3-1.

|  |  |
| --- | --- |
| Tác nhân | Mô tả tác nhân |
| User | * Đăng ký tài khoản * Đăng nhập tài khoản * Đăng xuất tài khoản * Xem sản phẩm * Xem chi tiết sản phẩm * Thêm sản phẩm vào giỏ hàng * Đánh giá sản phẩm * Xóa sản phẩm khỏi giỏ hàng |
| Admin | * Đăng ký tài khoản * Đăng nhập tài khoản * Đăng xuất tài khoản * Xem sản phẩm * Xem chi tiết sản phẩm * Thêm sản phẩm vào giỏ hàng * Đánh giá sản phẩm * Xóa sản phẩm khỏi giỏ hàng * Quản lý user * Quản lý sản phẩm * Quản lý đơn hàng * Quản lý đánh giá sản phẩm |

Bảng 3‑1 Danh sách tác nhân và mô tả

## Class diagram

A diagram of a user flow

Description automatically generated with medium confidence

# : HIỆN THỰC

## Giao diện chính

A screenshot of a computer

Description automatically generated

## Kết nối dữ liệu

A screenshot of a computer

Description automatically generated

## Cơ sở dữ liệu

A screenshot of a computer

Description automatically generated

## Login

A screenshot of a computer program

Description automatically generated

## register

A screenshot of a computer code

Description automatically generated

## cart

A computer code on a white background

Description automatically generated

## Product

A computer code with text

Description automatically generated

## Cấu hình cofig

A screenshot of a computer program

Description automatically generated

## Cấu trúc project

A screenshot of a computer

Description automatically generated

# : KẾT LUẬN

## Kết quả đạt được

* Sau thời gian thực hiện đề tài, website đã hoàn thành và đạt được một số kết quả sau:
* Tạo ra website để giới thiệu, bán quần áo thời trang trẻ em trực tuyến. Giúp các bậc phụ huynh dễ dàng lựa chọn và mua sắm sản phẩm thời trang cho con mình.
* Ứng dụng web hỗ trợ cho phụ huynh trong việc mua sắm và lựa chọn quần áo cho con cái thông qua website trực tuyến giúp tiết kiệm thời gian di chuyển và giảm thời gian làm thủ tục thanh toán.
* Giao diện của website thân thiện, dễ sử dụng

## Hạn chế của đồ án

* Thiếu kinh phí
* Một số chức năng còn thiếu sót như: gửi mail xác nhận, đánh giá sản phẩm, quên mật khẩu

## Hướng phát triển

* Đầu tư kinh phí
* Cố gắng bổ sung thêm những chức năng còn thiếu sót
* Khắc phục tối đa những lỗi phát sinh
* Phát triển trên app điện thoại
* Hoàn thiện website hơn giúp website trở nên dễ sử dụng hơn, hoạt động mượt mà ổn định hơn và công cụ tìm kiếm sản phẩm của website cho ra kết quả chính xác hơn để hỗ trợ người dung có thể dễ dàng, nhanh chóng có thể lựa chọn và mua được sản phẩm mình ưa thích.

Link github:

**https://github.com/2002ThanhHa/20027751\_NguyenThiThanhHa\_Project44**