

J0153514











**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA CÔNG NGHỆ THÔNG TIN**

**----🙣🕮🙡----**

**MÔN HỌC: ĐỒ ÁN CÔNG NGHỆ THÔNG TIN**

**BÁO CÁO CUỐI KỲ**

**GVHD: Từ Tuyết Hồng**

**SVTH: Trần Minh Trí**

**Mã lớp học:** PROJ215879\_23\_1\_10

Thành phố Thủ Đức, tháng 05 năm 2022

## DANH SÁCH SINH VIÊN THỰC HIỆN

*HỌC KÌ 2 NĂM HỌC 2021-2022*

|  |  |  |
| --- | --- | --- |
| **STT** | **HỌ VÀ TÊN SINH VIÊN** | **MÃ SỐ SINH VIÊN** |
| 1 | Trần Minh Trí | 21110247 |

*Ngày ... tháng 12 năm 2023*

*Giảng viên chấm điểm*

|  |  |
| --- | --- |
| **Điểm** | **Đánh giá của giảng viên** |
|  |  |

**LỜI CẢM ƠN**

Em xin gửi lời tri ân đến cô Từ Tuyết Hồng, đã hỗ trợ em rất nhiều trong quá trình học tập và làm bài báo cáo. Em đã có cơ hội tiếp cận nhiều kiến thức mới và thực hành các kỹ năng cần thiết cho môn đồ án công nghệ thông tin.

Em biết rằng bài báo cáo này chưa hoàn hảo và còn nhiều thiếu sót, em mong nhận được những nhận xét và góp ý của cô để em có thể hoàn thiện hơn. Em xin chân thành cảm ơn cô đã dành thời gian cho em.

Em xin chân thành cảm ơn !

Trần Minh Trí

**Contents**

[DANH SÁCH SINH VIÊN THỰC HIỆN 2](#_Toc152614929)

[**I.** **Đặc tả** 5](#_Toc152614930)

[**II.** **Phân công công việc** 5](#_Toc152614931)

[**III.** **Thiết kế** 5](#_Toc152614932)

[**IV.** **Cài đặt và kiểm thử** 7](#_Toc152614933)

[**V.** **Kết luận** 8](#_Toc152614934)

# **Đặc tả**

* 1. Phần mềm dùng làm gì: nêu ngữ cảnh sử dụng phần mềm.

* 1. Dữ liệu, thông tin đầu vào
  2. Các tình huống sử dụng (mục đích, tính năng)
  3. Giao diện dự kiến

# **Phân công công việc**

Mô tả phân công công việc theo bảng

|  |  |  |  |
| --- | --- | --- | --- |
| TT  1 | Tên SV  Trần Minh Trí | Mô tả khái quát mảng công việc SV thực hiện trong đồ án.  Nghiên cứu và Lập Kế Hoạch:  Nghiên cứu đề tài: Tìm hiểu về đề tài của mình, đọc các tài liệu, bài báo, và tìm hiểu về các nghiên cứu liên quan.  Lập kế hoạch: Xác định các bước cần thực hiện và lên kế hoạch công việc theo thời gian.  Thu Thập Dữ Liệu:  Thiết kế nghiên cứu hoặc hệ thống  Thu thập dữ liệu: Tiến hành thực hiện kế hoạch thu thập dữ liệu.  Phân Tích và Xử Lý Dữ Liệu:  Phân tích dữ liệu: Sử dụng các phương pháp thống kê hoặc các công cụ phân tích để rút trích thông tin quan trọng.  Xử lý dữ liệu: Thực hiện xử lý dữ liệu để chuẩn hóa, làm sạch, hoặc chuyển đổi dữ liệu.  Phát triển và Kiểm Thử Giải Pháp:  Thực hiện việc lập trình và triển khai giải pháp.  Kiểm thử: Tiến hành kiểm thử đơn vị, kiểm thử tích hợp, và kiểm thử hệ thống để đảm bảo tính ổn định và hiệu suất của giải pháp.  Viết Báo Cáo:  Viết báo cáo: Sinh viên cần viết báo cáo mô tả chi tiết về đề tài, phương pháp nghiên cứu, kết quả, và kết luận. | Ước tính phần trăm đóng góp  100% |

# **Thiết kế**

Mô tả quá trình làm (how to do): Cần giải thích cách thức xây dựng hệ thống, đặc biệt chú trọng làm rõ thiết kế, và giải thích được tại sao thiết kế như vậy.

1. Thuật toán
2. Thiết kế lớp

Trình bày theo định dạng dưới:

* 1. Mẫu bảng danh mục các lớp do sinh viên thiết kế (không cần mô tả các lớp có sẵn)

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên lớp  Thừa kế hoặc dẫn xuất từ (ghi rõ tên lớp cơ sở) – nếu có | Mục đích | Tên các SV phụ trách viết  (Nếu code lấy từ ngồn khác thì phải chỉ rõ lấy từ ngồn nào). |

**Lưu ý**: Trong trường hợp nhiều lớp được viết bởi 1 sinh viên, sử dụng mẫu dưới

Sinh viên phụ trách: Nguyễn Văn A (Ghi rõ nguồn tham khảo nếu lấy từ nguồn khác)

|  |  |  |
| --- | --- | --- |
| TT | Tên lớp  AccountServlet dẫn xuất từ lớp User, UserModel, ServletUtils. Thừa kế HttpServlet | Hiển Thị Trang Đăng Ký (/Register):  Khi có yêu cầu đến đường dẫn /Account/Register, servlet này chuyển hướng người dùng đến trang Register.jsp để đăng ký tài khoản.  Hiển Thị Trang Đăng Nhập (/Login):  Khi có yêu cầu đến đường dẫn /Account/Login, servlet kiểm tra xem người dùng đã đăng nhập hay chưa.  Nếu đã đăng nhập, chuyển hướng người dùng đến trang chủ /Home.  Nếu chưa đăng nhập, chuyển hướng người dùng đến trang Login.jsp để đăng nhập.  Hiển Thị Trang Hồ Sơ Người Dùng (/Profile):  Khi có yêu cầu đến đường dẫn /Account/Profile, servlet này chuyển hướng người dùng đến trang Profile.jsp để xem và quản lý thông tin hồ sơ người dùng.  Kiểm Tra Tên Người Dùng Có Khả Dụng (/IsAvailable):  Khi có yêu cầu đến đường dẫn /Account/IsAvailable, servlet kiểm tra tính khả dụng của tên người dùng.  Trả về kết quả (true/false) dưới dạng JSON để kiểm tra từ phía máy khách (front-end) khi người dùng đang nhập thông tin đăng ký.  Đăng Ký Người Dùng (/Register - Phương Thức POST):  Khi có yêu cầu POST đến /Account/Register, servlet này xử lý việc đăng ký người dùng.  Sử dụng thư viện Bcrypt để mã hóa mật khẩu trước khi lưu vào cơ sở dữ liệu.  Thêm thông tin người dùng mới vào cơ sở dữ liệu thông qua UserModel.  Đăng Nhập Người Dùng (/Login - Phương Thức POST):  Khi có yêu cầu POST đến /Account/Login, servlet xử lý quá trình đăng nhập.  Kiểm tra thông tin đăng nhập, sử dụng Bcrypt để so sánh mật khẩu đã mã hóa.  Nếu đăng nhập thành công, tạo và lưu thông tin phiên đăng nhập trong HttpSession.  Đăng Xuất Người Dùng (/Logout - Phương Thức POST):  Khi có yêu cầu POST đến /Account/Logout, servlet xử lý quá trình đăng xuất.  Hủy phiên đăng nhập, đặt giá trị auth và authUser trong HttpSession về giá trị mặc định.  Chuyển hướng người dùng về trang trước đó hoặc trang chủ.  Xử Lý Trường Hợp Mặc Định và 404:  Nếu đường dẫn không phù hợp với các trường hợp đã nêu (/Register, /Login, /Profile, /IsAvailable), servlet này chuyển hướng người dùng đến trang 404.jsp để hiển thị thông báo lỗi 404 Not Found. |

|  |  |  |
| --- | --- | --- |
|  | AdminCategoryServlet dẫn xuất từ lớp Category,CategoryModel, Servletutils. Thừa kế HttpServlet | Hiển Thị Danh Sách Danh Mục (/Index):  Khi có yêu cầu đến đường dẫn /Admin/Category/Index hoặc /Admin/Category/ (mặc định), servlet này lấy danh sách tất cả danh mục bằng cách gọi phương thức findAll từ CategoryModel.  Đặt danh sách danh mục vào thuộc tính "categories" của request để truyền vào trang JSP.  Chuyển hướng người dùng đến trang Index.jsp để hiển thị danh sách danh mục trong giao diện quản trị.  Hiển Thị Trang Thêm Danh Mục (/Add):  Khi có yêu cầu đến đường dẫn /Admin/Category/Add, servlet này chuyển hướng người dùng đến trang Add.jsp để thêm mới danh mục.  Hiển Thị Trang Sửa Danh Mục (/Edit):  Khi có yêu cầu đến đường dẫn /Admin/Category/Edit, servlet này lấy thông tin danh mục dựa trên id được truyền vào từ tham số yêu cầu.  Nếu danh mục tồn tại, đặt thông tin danh mục vào thuộc tính "category" của request để truyền vào trang JSP.  Chuyển hướng người dùng đến trang Edit.jsp để sửa thông tin danh mục.  Nếu danh mục không tồn tại, có thể thực hiện một hành động khác như chuyển hướng người dùng đến trang 404.jsp hoặc trang quản lý danh mục.  Xử Lý Thêm, Sửa, Xóa Danh Mục (Phương Thức POST):  Mã nguồn chứa các phương thức (doPost) để xử lý các chức năng thêm, sửa, xóa danh mục. Các chức năng này được triển khai trong các phương thức tương ứng để quản lý dữ liệu danh mục từ giao diện quản trị viên. |
|  | AdminProductServlet dẫn xuất từ lớp Product ,ProductModel, Servletutils. Thừa kế HttpServlet | Hiển Thị Danh Sách Sản Phẩm (/Index):  Khi có yêu cầu đến đường dẫn /Admin/Product/Index hoặc /Admin/Product/ (mặc định), servlet này lấy danh sách tất cả sản phẩm bằng cách gọi phương thức findAll từ ProductModel.  Đặt danh sách sản phẩm vào thuộc tính "products" của request để truyền vào trang JSP.  Chuyển hướng người dùng đến trang Index.jsp để hiển thị danh sách sản phẩm trong giao diện quản trị.  Xử Lý Trường Hợp Mặc Định và 404:  Nếu đường dẫn không phù hợp với các trường hợp đã nêu (/Index), servlet này chuyển hướng người dùng đến trang 404.jsp để hiển thị thông báo lỗi 404 Not Found.  Chức Năng Thêm, Sửa, Xóa:  Mã nguồn chứa các phương thức (doPost) để xử lý các chức năng thêm, sửa, xóa sản phẩm, nhưng hiện đã được tạm bỏ comment. Các chức năng này có thể được triển khai trong các phương thức tương ứng để quản lý dữ liệu sản phẩm từ giao diện quản trị viên. |
|  | DefaultServlet dẫn xuất từ ServletUtils. Thừa kế HttpServlet | Xử Lý Yêu Cầu GET và POST:  Khi có yêu cầu GET hoặc POST đến địa chỉ gốc ("/") hoặc bất kỳ đường dẫn nào không được xử lý bởi các servlet khác, servlet này chuyển hướng người dùng đến trang 404.jsp.  Trang 404.jsp thường được sử dụng để hiển thị thông báo lỗi 404 Not Found khi người dùng yêu cầu một trang không tồn tại.  Chức Năng Mặc Định Hoặc Báo Lỗi:  Đối với mọi yêu cầu không khớp với các servlet hoặc đường dẫn đã được xác định trước, servlet này đảm bảo rằng người dùng nhận được một trang thông báo lỗi thích hợp. |
|  | HomeServlet dẫn xuất từ ServletUtils. Thừa kế HttpServlet | Hiển Thị Trang Chủ (/Index):  Khi có yêu cầu đến đường dẫn /Home/Index hoặc /Home/ (mặc định), servlet này chuyển hướng người dùng đến trang chủ Index.jsp.  Có thể có các xử lý khác như kiểm tra thông tin đăng nhập của người dùng thông qua HttpSession, nhưng hiện tại đã được tạm bỏ qua bằng cách comment.  Hiển Thị Trang Giới Thiệu (/About):  Khi có yêu cầu đến đường dẫn /Home/About, servlet này chuyển hướng người dùng đến trang About.jsp.  Trang này có thể chứa thông tin giới thiệu về ứng dụng, doanh nghiệp, hoặc các thông tin khác tùy thuộc vào yêu cầu của ứng dụng.  Xử Lý Trường Hợp Mặc Định và 404:  Nếu đường dẫn không phù hợp với các trường hợp đã nêu (/Index hoặc /About), servlet này chuyển hướng người dùng đến trang 404.jsp để hiển thị thông báo lỗi 404 Not Found. |
|  | MiscServlet dẫn xuất từ ServletUtils. Thừa kế HttpServlet | Xử lý Yêu Cầu GET:  Khi có yêu cầu GET đến đường dẫn /Misc/Editor, nó chuyển hướng (forward) người dùng đến trang Editor.jsp.  Khi có yêu cầu GET đến đường dẫn /Misc/Upload, nó chuyển hướng người dùng đến trang Upload.jsp.  Nếu có đường dẫn khác, nó chuyển hướng người dùng đến trang 404.jsp để hiển thị trang lỗi 404 Not Found.  Xử lý Yêu Cầu POST:  Khi có yêu cầu POST đến đường dẫn /Misc/Editor, nó gọi phương thức postEditor để xử lý dữ liệu gửi từ trang Editor.jsp.  Khi có yêu cầu POST đến đường dẫn /Misc/Upload, nó gọi phương thức postUpload để xử lý dữ liệu gửi từ trang Upload.jsp.  Nếu có đường dẫn khác, nó chuyển hướng người dùng đến trang 404.jsp.  Xử lý Upload File:  Trong phương thức postUpload, nó xử lý việc upload file từ yêu cầu và lưu trữ file tải lên vào thư mục public/user-avatars.  In ra console tên file và các thông tin khác về file tải lên.  Chỉnh Sửa Thông Tin Trang Editor:  Trong phương thức postEditor, nó xử lý thông tin được gửi từ trang Editor.jsp (ví dụ: mô tả ngắn, mô tả đầy đủ) và in ra console. |
|  | ProductFEServlet dẫn xuất từ Product, ProductModel,ServletUtils. Thừa kế HttpServlet | Hiển Thị Sản Phẩm Theo Danh Mục (/ByCat):  Khi có yêu cầu đến đường dẫn /Product/ByCat, servlet này lấy id của danh mục sản phẩm từ tham số yêu cầu (CatId).  Gọi phương thức findByCatId từ ProductModel để lấy danh sách sản phẩm thuộc danh mục có CatId.  Đặt danh sách sản phẩm vào thuộc tính "products" của request để truyền vào trang JSP.  Chuyển hướng người dùng đến trang ByCat.jsp để hiển thị danh sách sản phẩm theo danh mục.  Hiển Thị Chi Tiết Sản Phẩm (/Detail):  Khi có yêu cầu đến đường dẫn /Product/Detail, servlet này lấy id của sản phẩm từ tham số yêu cầu (proId).  Gọi phương thức findById từ ProductModel để lấy thông tin chi tiết của sản phẩm có proId.  Nếu sản phẩm không tồn tại, chuyển hướng người dùng đến trang /Home (trang chủ).  Nếu sản phẩm tồn tại, đặt thông tin sản phẩm vào thuộc tính "product" của request để truyền vào trang JSP.  Chuyển hướng người dùng đến trang Detail.jsp để hiển thị thông tin chi tiết của sản phẩm.  Xử Lý Trường Hợp Mặc Định:  Nếu đường dẫn không phù hợp với các trường hợp đã nêu (/ByCat hoặc /Detail), servlet này chuyển hướng người dùng đến trang 404.jsp để hiển thị thông báo lỗi 404 Not Found. |

* 1. Mẫu bảng mô tả các phương thức trong một lớp

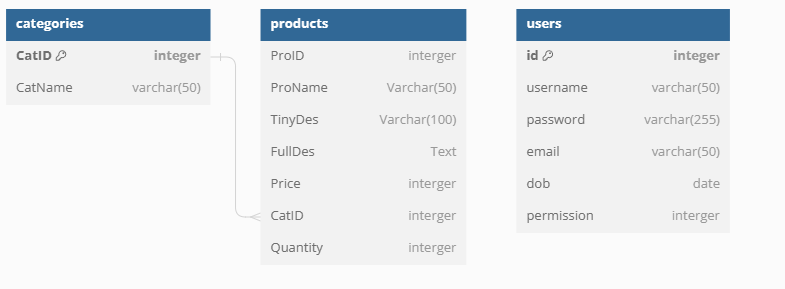
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TT | Phương thức | Mục đích | Tên file, số thứ tự dòng chứa khai báo | Tên SV phụ trách viết |
|  | **setPos(int x, int y)** | Đặt vị trí điểm | Graphics.java |  |
| Input: x,y | Tọa độ | (160) |
| Output: không có |  |  |
| Pseudo code: (không cần mô |  |  |
| tả nếu quá đơn giản) |  |  |

**Lưu ý**: Trong trường hợp tất cả các phương thức được viết bởi 1 sinh viên, sử dụng mẫu dưới Sinh viên phụ trách: Nguyễn Văn A (Ghi rõ nguồn tham khảo nếu lấy từ nguồn khác)

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Phương thức | Mục đích | Tên file, số thứ tự dòng chứa khai báo |
|  | Get  Set | Truy cập dữ liệu: Phương thức get được sử dụng để truy cập giá trị của một thuộc tính từ bên ngoài lớp. Điều này cho phép các thành phần khác như trang JSP hoặc các lớp xử lý dữ liệu truy cập giá trị của thuộc tính mà không cần biết chi tiết cài đặt nội bộ của lớp.  Gán giá trị cho thuộc tính: Phương thức set được sử dụng để gán giá trị cho một thuộc tính từ bên ngoài lớp. Khi giá trị của một thuộc tính được thay đổi, phương thức set có thể thực hiện các kiểm tra hợp lý (validation) hoặc xử lý logic trước khi thay đổi giá trị.  Đảm bảo tính an toàn và kiểm soát dữ liệu: Sử dụng phương thức set cho phép kiểm tra và xử lý dữ liệu trước khi nó được gán cho thuộc tính, giúp đảm bảo tính an toàn và kiểm soát dữ liệu. | File Category line 15 - line 29 (get , set)  File Product line 33 - line 79 (get , set)  File User line 24 – line 50 (get) |
|  | Foward | Mục đích Chính: Phương thức forward được sử dụng để chuyển hướng người dùng từ một servlet (hoặc trang JSP) đến một servlet hoặc trang JSP khác trên máy chủ.  Trạng Thái Request: Trong quá trình forward, request và response objects được chuyển tiếp từ servlet gọi forward đến servlet hoặc trang JSP mới. Điều này có nghĩa là thông tin như các thuộc tính của request và response sẽ được duy trì.  Tốc Độ: Forward thường nhanh hơn redirect vì nó xử lý tất cả trên máy chủ, không có yêu cầu thêm từ trình duyệt. | File ServletUtils line 10 |
|  | Redirect | Mục đích Chính: Phương thức redirect được sử dụng để chuyển hướng người dùng từ một servlet hoặc trang JSP đến một URL khác (có thể là trang JSP, servlet, hoặc bất kỳ tài nguyên web nào khác).  Trạng Thái Request: Khi sử dụng redirect, trình duyệt web sẽ gửi một yêu cầu mới đến URL mới, do đó không giữ lại bất kỳ trạng thái request nào từ request gốc. Điều này có thể gây mất mát dữ liệu nếu cần thiết.  Tốc Độ: Redirect thường chậm hơn forward do sự tương tác giữa trình duyệt và máy chủ thông qua nhiều yêu cầu. | File ServletUtils line 15 |
|  | doGet | Mục đích chính: Được sử dụng để xử lý các yêu cầu HTTP GET từ phía client.  Thích hợp khi: Bạn muốn lấy dữ liệu từ máy chủ, ví dụ như hiển thị trang HTML, truy xuất thông tin từ cơ sở dữ liệu, hoặc thực hiện các hành động không tác động đến trạng thái của máy chủ (idempotent).  Dữ liệu được gửi: Dữ liệu thường được gửi qua URL trong phần query string. | Các file trong controllers |
|  | doPost | Mục đích chính: Được sử dụng để xử lý các yêu cầu HTTP POST từ phía client.  Thích hợp khi: Bạn muốn gửi dữ liệu từ client đến server, ví dụ như khi submit một form hoặc thực hiện các hành động ảnh hưởng đến trạng thái của máy chủ.  Dữ liệu được gửi: Dữ liệu thường được gửi trong phần thân của yêu cầu HTTP, không hiển thị trong URL. | Các file trong controllers |
|  | Init  doFilter  destroy | Mục đích: Được gọi một lần khi bộ lọc được khởi tạo.  Nhiệm vụ: Thực hiện các công việc khởi tạo, như đọc các thiết lập từ tệp cấu hình, thiết lập kết nối đến cơ sở dữ liệu, hoặc thực hiện các tác vụ chuẩn bị trước cho việc lọc.  Mục đích: Thực hiện các xử lý của bộ lọc khi một yêu cầu đi qua nó.  Nhiệm vụ: Chứa mã lọc thực tế để thực hiện các chức năng như kiểm tra và chỉnh sửa yêu cầu hoặc phản hồi, xác thực, ghi log, nén dữ liệu, v.v.  Mục đích: Được gọi một lần khi bộ lọc bị hủy bỏ.  Nhiệm vụ: Thực hiện các công việc giải phóng tài nguyên, đóng kết nối cơ sở dữ liệu, hoặc làm bất kỳ công việc dọn dẹp cần thiết trước khi bộ lọc kết thúc | Các file trong Filters |
|  | findById | Truy vấn Đối tượng theo ID:  Phương thức findById giúp truy vấn cơ sở dữ liệu để tìm kiếm đối tượng với khóa chính cụ thể.  Thường được sử dụng trong các tình huống nơi cần lấy thông tin chi tiết của một đối tượng dựa trên ID, chẳng hạn như khi hiển thị thông tin chi tiết của một bản ghi trong ứng dụng web.  Xử lý Trường Hợp Đối Tượng Không Tồn Tại:  Khi không tìm thấy đối tượng nào với ID đã cung cấp, phương thức thường trả về null hoặc một giá trị biểu diễn việc không tìm thấy (chẳng hạn, Optional.empty()).  Cho phép xử lý các trường hợp đối tượng không tồn tại và trả về thông báo hoặc thực hiện các xử lý khác tùy thuộc vào yêu cầu của ứng dụng. | File CategoryModel line 31  File ProductModel line 25 |
|  | findAll | Lấy Tất Cả Dữ Liệu: Phương thức này được sử dụng để truy vấn tất cả các bản ghi từ cơ sở dữ liệu. Điều này hữu ích khi bạn cần hiển thị hoặc xử lý toàn bộ dữ liệu trong ứng dụng web của mình.  Hiển Thị Danh Sách: Khi bạn muốn hiển thị danh sách các đối tượng hoặc bản ghi từ cơ sở dữ liệu trên giao diện người dùng, phương thức findAll giúp lấy dữ liệu cần thiết từ cơ sở dữ liệu để hiển thị. | File CategoryModel line 24  File ProductModel line 18 |
|  | add | Thêm Đối Tượng vào Cơ Sở Dữ Liệu:  Phương thức "add" có thể được sử dụng để thêm một đối tượng mới vào cơ sở dữ liệu. Điều này thường xảy ra khi người dùng thực hiện một hành động như tạo mới hoặc đăng ký trên trang web.  Thêm Phần Tử vào Danh Sách hoặc Mảng:  Trong ngữ cảnh của một class quản lý danh sách hoặc mảng, phương thức "add" có thể được sử dụng để thêm một phần tử mới vào danh sách hoặc mảng đó.  Thêm Thẻ HTML hoặc Nội Dung vào Trang Web:  Trong ngữ cảnh của việc tạo nội dung động trên trang web, phương thức "add" có thể được sử dụng để thêm các thẻ HTML mới hoặc nội dung vào trang web dựa trên các yêu cầu từ người dùng hoặc từ dữ liệu từ nguồn khác.  Thêm Đối Tượng vào Session hoặc Cookies:  Trong lập trình web Java, phương thức "add" có thể được sử dụng để thêm thông tin người dùng vào session hoặc cookies để duy trì trạng thái hoặc thông tin tương tác trong suốt phiên làm việc của người dùng.  Thêm Thuộc Tính cho Đối Tượng:  Nếu phương thức "add" thuộc về một class đại diện cho một đối tượng cụ thể, nó có thể được sử dụng để thêm các thuộc tính mới hoặc thiết lập giá trị cho các thuộc tính tồn tại của đối tượng đó. | File UserModel line 25  File CategoryModel line 45  File ProductModel line 48 |
|  | getConnection | Thiết lập Kết Nối:  Mục đích chính của phương thức getConnection là để thiết lập một kết nối đến cơ sở dữ liệu từ ứng dụng Java.  Quản lý Thông Tin Kết Nối:  Phương thức này thường yêu cầu các thông tin như URL cơ sở dữ liệu, tên người dùng, mật khẩu và các thông số khác để xác định kết nối.  Truy Cập Cơ Sở Dữ Liệu:  Khi kết nối đã được thiết lập thành công, nó cung cấp một đối tượng Connection mà bạn có thể sử dụng để thực hiện các thao tác truy cập cơ sở dữ liệu như truy vấn dữ liệu, cập nhật dữ liệu, thêm mới dữ liệu, và xóa dữ liệu.  Quản lý Kết Nối:  Phương thức getConnection thường có nhiều chữ ký phương thức khác nhau, cung cấp các tùy chọn khác nhau để quản lý kết nối, chẳng hạn như thời gian chờ, giao thức kết nối, và các thuộc tính khác. | File DbUltis line 9 |
|  | Update  Delete | Mục Đích:  Cập nhật thông tin của một tài nguyên hoặc bản ghi trong cơ sở dữ liệu.  Thường được sử dụng để chỉnh sửa thông tin của người dùng, sản phẩm, hoặc bất kỳ đối tượng nào khác.  Mục Đích:  Xóa dữ liệu từ cơ sở dữ liệu.  Thường được sử dụng để xóa một tài nguyên hoặc bản ghi cụ thể. | Trong file CategoryModel line 54 và line 64  File ProductModel line 62 và line 77 |
|  | findByCatId | Lấy Danh Sách Phần Tử Theo Danh Mục:  Phương thức này thường được sử dụng để truy vấn cơ sở dữ liệu để lấy ra tất cả các phần tử (thường là đối tượng) thuộc một danh mục cụ thể nào đó.  Tổ Chức Dữ Liệu:  Phương thức này giúp tổ chức dữ liệu theo danh mục, giúp quản lý dễ dàng hơn và cung cấp thông tin chỉ về những phần tử thuộc cùng một loại hoặc chủ đề.  Thực Hiện Truy Vấn Tùy Biến:  Có thể có các tham số bổ sung để tùy chỉnh truy vấn, chẳng hạn như sắp xếp theo một tiêu chí nào đó, giới hạn số lượng kết quả trả về, hoặc thêm điều kiện lọc.  Hỗ Trợ Hiển Thị Trên Giao Diện Người Dùng:  Dữ liệu trả về từ phương thức này thường được sử dụng để hiển thị thông tin trên giao diện người dùng, chẳng hạn như hiển thị danh sách sản phẩm trong một danh mục cụ thể.  Hỗ Trợ Tương Tác Người Dùng:  Nếu danh mục được sử dụng trong các chức năng tương tác của người dùng, ví dụ như chọn danh mục trong một menu thả xuống, phương thức này có thể giúp cung cấp dữ liệu cần thiết cho các tương tác đó. | File ProductModel line 25 |
|  | getAllProduct | Lấy Danh Sách Sản Phẩm:  Phương thức này thường truy vấn cơ sở dữ liệu hoặc nguồn dữ liệu khác để lấy thông tin về tất cả các sản phẩm có sẵn trong hệ thống.  Hiển Thị Trên Giao Diện Người Dùng:  Kết quả từ phương thức này có thể được chuyển đến giao diện người dùng để hiển thị danh sách sản phẩm trên trình duyệt hoặc ứng dụng web.  Hỗ Trợ API (Application Programming Interface):  Nếu hệ thống của bạn cung cấp API cho ứng dụng khác, phương thức getAllProduct có thể được sử dụng để trả về dữ liệu sản phẩm cho các ứng dụng bên ngoài.  Tính Tương Tác với Dữ Liệu:  Cho phép các thao tác tương tác với dữ liệu sản phẩm như lọc, sắp xếp, hoặc tìm kiếm dựa trên yêu cầu của người dùng.  Dữ Liệu Cho Các Tính Năng Khác:  Dữ liệu từ phương thức này có thể được sử dụng để hỗ trợ các tính năng khác như thống kê, báo cáo, hoặc các chức năng quản lý khác.  Đảm Bảo Hiệu Suất:  Quản lý việc lấy dữ liệu theo các cách tối ưu để đảm bảo hiệu suất tốt, tránh việc lấy quá nhiều dữ liệu không cần thiết. | File ProductModel line 12 |
|  | findByUsername | Xác thực Người Dùng:  Trong hệ thống đăng nhập, phương thức này có thể được sử dụng để kiểm tra xem một người dùng có tồn tại trong cơ sở dữ liệu không khi họ đăng nhập.  Thường được sử dụng để kiểm tra tính duy nhất của tên đăng nhập, đảm bảo rằng không có hai người dùng nào sử dụng cùng một tên đăng nhập.  Lấy Thông Tin Người Dùng:  Sau khi xác thực, phương thức có thể trả về toàn bộ thông tin của người dùng (ví dụ: ID, mật khẩu được mã hóa, quyền hạn, thông tin cá nhân, v.v.).  Quản Lý Phân Quyền:  Phương thức có thể được sử dụng để kiểm tra quyền hạn của người dùng, giúp quản lý quyền truy cập vào các phần của ứng dụng hoặc tính năng cụ thể. | File UserModel line 10 |

1. Thiết kế cơ sở dữ liệu

Hình vẽ biễu diễn quan hệ giữa các bảng và multiplicity (có thể dùng mô hình thực thể kết hợp):



Mẫu bảng mô tả các Table trong CSDL

|  |  |  |
| --- | --- | --- |
| TT | Tên bảng  Categories  Products  Users | Mục đích  Mục đích chính của bảng này là để tổ chức và phân loại dữ liệu một cách có tổ chức, giúp quản trị và tìm kiếm dữ liệu dễ dàng hơn.  Mục đích chính của bảng này là theo dõi và quản lý thông tin chi tiết về các sản phẩm trong một hệ thống.  Mục đích chính của bảng này là để quản lý thông tin cá nhân của từng người dùng trong ứng dụng hoặc hệ thống. |

1. Bảng mô tả các Field trong 1 Table

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TT | | Tên trường  CatID  CatName | | Kiểu dữ liệu  Int  Varchar(50) | Mục đích  Mục đích chính của CatID là để làm cho mỗi danh mục có một định danh duy nhất và giúp liên kết với các bảng khác nếu cần thiết.  Mục đích chính của CatName là để hiển thị tên của danh mục cho người đọc hoặc người quản trị cơ sở dữ liệu. Nó giúp làm cho dữ liệu dễ đọc và dễ hiểu hơn. | |
| TT | | Tên trường  ProID  ProName  TinyDes  FullDes  Price  CatID  Quantity | Kiểu dữ liệu  Int  Varchar(50)  Varchar(100)  Text  Int  Int  Int | | Mục đích  Trường để lưu trữ một giá trị duy nhất để định danh mỗi sản phẩm trong bảng. Thường được sử dụng làm khóa chính (primary key) để đảm bảo tính duy nhất của mỗi sản phẩm.  Trường để lưu trữ tên của sản phẩm. Được sử dụng để mô tả hoặc xác định một cách ngắn gọn về sản phẩm.  Trường này thường được sử dụng để chứa mô tả ngắn gọn, tóm tắt về sản phẩm. Đôi khi được sử dụng để hiển thị thông tin ngắn trong các danh sách sản phẩm.  Lưu trữ mô tả chi tiết về sản phẩm. Thường được sử dụng để hiển thị thông tin đầy đủ và chi tiết khi người dùng xem chi tiết sản phẩm.  Chứa thông tin về giá của sản phẩm. Thường là một số dấu phẩy động để biểu diễn giá trị tiền tệ.  Trường để xác định loại sản phẩm thuộc về danh mục nào. Thường làm khóa ngoại (foreign key) kết nối đến bảng "categories" để liên kết sản phẩm với danh mục tương ứng.  Lưu trữ thông tin về số lượng tồn kho của sản phẩm. Được sử dụng để theo dõi tình trạng hàng tồn kho và quản lý số lượng sản phẩm có sẵn. | |
| TT | | Tên trường  id  username  password  name  email  dob  permission | Kiểu dữ liệu  Int  Varchar(50)  Varchar(255)  Varchar(50)  Varchar(50)  Date  int | | Mục đích  Mục đích của cột này là để xác định mỗi người dùng một cách duy nhất trong hệ thống. Thường là khóa chính (primary key) và được sử dụng để liên kết với các bảng khác nếu cần thiết.  Dùng để lưu trữ tên đăng nhập của người dùng. Thường được sử dụng để xác định người dùng khi họ đăng nhập vào hệ thống.  Chứa một phiên bản được mã hóa của mật khẩu người dùng. Mục đích là bảo vệ mật khẩu từ việc lộ thông tin cá nhân.  Lưu trữ tên của người dùng, có thể bao gồm tên và họ. Thông tin này có thể được sử dụng để hiển thị tên đầy đủ trong các phần của ứng dụng hoặc hệ thống.  Chứa địa chỉ email của người dùng. Thường được sử dụng để liên lạc với người dùng và gửi thông báo quan trọng.  Lưu trữ ngày sinh của người dùng. Có thể được sử dụng cho các chức năng liên quan đến tuổi, như xác định độ tuổi của người dùng.  Mô tả các quyền và vai trò của người dùng trong hệ thống. Có thể là một số hoặc chuỗi mô tả vai trò cụ thể mà người dùng đó có trong hệ thống. | |

1. Thiết kế giao diện

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Màn hình/Cửa sổ/Dialog  (Tên màn hình trên 1 dòng; dưới dòng này là hinh chụp màn hình)  Màn Hình Chính | Mục đích  Mục đích của giao diện màn hình chính (hoặc màn hình dashboard) thường là cung cấp một tổng quan toàn diện và dễ sử dụng về thông tin hoặc chức năng quan trọng của một ứng dụng, hệ thống hoặc trang web | Giải thích ngắn gọn các quyết định thiết kế khi thiết kế màn hình (Người thiết kế ở trên 1 dòng. Sau dòng đó là phần giải thích.  Trần Minh Trí. Màn hình chính thiết kế bao gồm một nav bar (nav partial) và hai list group ở bên trái (left partial) và ở dưới nav là một body nơi chứa thông tin, hoặc đổ ra dữ liệu lấy từ database. Ở trên nav sẽ có các button giúp ta về trang chính (Home) hoặc là Product để hiện ra tất cả sản phẩm và About us (tác giả), ngoài ra có login và register. Còn ở phía bên trái thì có sự đổ ra của các danh mục sản phẩm và phần thêm của các thương hiệu nổi tiếng bằng cách sử dụng List |
|  | Màn hình nơi đưa ra thông tin tất cả sản phẩm | Màn hình hiển thị thông tin về tất cả sản phẩm trong một hệ thống thương mại điện tử. | Trần Minh Trí. Ở màn hình này ta dùng chung thiết kế của trang chính và ở phần body ta đổ ra dữ liệu về sản phẩm. |
|  | Màn hình đăng ký | Màn hình đăng ký tài khoản trong một ứng dụng hoặc trang web có mục đích chính là cho phép người dùng tạo ra một tài khoản mới. | Trần Minh Trí. Tương tự thì ở đây phần body sẽ là một form để người dùng điền vào các thông tin và gởi thông tin này đi.  Thông tin cần thiết:  Xác định thông tin bắt buộc để người dùng phải cung cấp (ví dụ: tên đăng nhập, mật khẩu, email).  Tránh yêu cầu quá nhiều thông tin ở giai đoạn đăng ký để giảm sự phiền toái cho người dùng.  Xác nhận mật khẩu:  Bao gồm trường nhập lại mật khẩu để người dùng xác nhận không nhập sai mật khẩu.  Hướng dẫn và Gợi ý:  Cung cấp hướng dẫn rõ ràng về yêu cầu đối với mật khẩu (ví dụ: độ dài, ký tự đặc biệt).  Hiển thị gợi ý và mẫu mật khẩu mạnh để hỗ trợ người dùng tạo mật khẩu an toàn.  Kiểm tra tính hợp lệ:  Tiến hành kiểm tra tính hợp lệ ngay tại màn hình đăng ký để tránh gặp lỗi sau đó.  Thông báo lỗi một cách rõ ràng và mô tả khi người dùng nhập thông tin không đúng.  Quyền riêng tư:  Cung cấp thông tin về chính sách quyền riêng tư và điều khoản sử dụng.  Tích hợp các ô chọn để người dùng đồng ý với các điều khoản và điều kiện.  Xác thực và An toàn:  Sử dụng các biện pháp bảo mật như CAPTCHA để ngăn chặn tấn công tự động.  Bảo đảm rằng kết nối giữa trình duyệt và máy chủ được mã hóa để đảm bảo an toàn thông tin.  Trải nghiệm Người dùng:  Tối ưu hóa giao diện để dễ dàng sử dụng trên cả điện thoại di động và máy tính.  Sử dụng các biểu tượng và hình ảnh thân thiện để tạo trải nghiệm dễ nhìn và thân thiện.  Xác nhận Đăng ký:  Hiển thị thông báo hoặc chuyển hướng đến một trang xác nhận sau khi người dùng đăng ký thành công. |
|  | Màn hình đăng nhập | Màn hình đăng nhập (login screen) trong một ứng dụng hoặc hệ thống thông thường có những mục đích quan trọng sau:  Xác Thực Người Dùng:  Mục đích chính của màn hình đăng nhập là xác thực danh tính của người dùng. Người dùng phải cung cấp thông tin đăng nhập hợp lệ, như tên người dùng và mật khẩu, để truy cập vào hệ thống.  Bảo Mật Hệ Thống:  Màn hình đăng nhập đóng vai trò quan trọng trong việc đảm bảo bảo mật hệ thống. Việc yêu cầu thông tin đăng nhập giúp ngăn chặn người không được phép truy cập vào thông tin và chức năng quan trọng của ứng dụng.  Ghi Nhớ Thông Tin Người Dùng:  Một số màn hình đăng nhập có tính năng "ghi nhớ đăng nhập" để lưu trữ thông tin đăng nhập của người dùng, giúp họ tránh việc phải nhập lại thông tin mỗi lần sử dụng. Tính năng này thường được lưu dưới dạng "cookie" hoặc thông tin địa chỉ IP.  Quản Lý Phiên Đăng Nhập:  Màn hình đăng nhập cũng thường liên quan đến quản lý phiên đăng nhập. Sau khi xác thực, người dùng sẽ có một phiên làm việc được tạo ra, giúp họ truy cập vào các trang và chức năng của ứng dụng trong khoảng thời gian nhất định.  Thông Báo Lỗi Đăng Nhập:  Nếu có lỗi trong quá trình đăng nhập (ví dụ: sai tên đăng nhập, mật khẩu), màn hình đăng nhập thường thông báo lỗi và cung cấp hướng dẫn cho người dùng để sửa lỗi. | Trần Minh Trí. Tạo ra một giao diện cho người dùng đã đăng ký đăng nhập bằng Username và Password họ đã đăng ký hoặc nếu chưa thì có thể trở về giao diện home, nếu có thì sẽ được đăng nhập vào. |
|  | Màn hình sau khi đăng nhập thành công | Màn hình sau khi đăng nhập thành công thường có mục đích chính là cung cấp trải nghiệm sử dụng cho người dùng sau khi họ đã xác thực thông tin đăng nhập của mình. | Trần Minh Trí. Màn hình chính thiết kế bao gồm một nav bar (nav partial) và hai list group ở bên trái (left partial) và ở dưới nav là một body nơi chứa thông tin.Ngoài ra phái trên phải góc màn hình có dòng chữ chào người dùng, thay vì login và register. |

# **Cài đặt và kiểm thử**

Mô tả các tình huống kiểm thử:

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tình huống 1 Dữ liệu vào  Người dùng đăng ký tài khoản  Kết quả dự kiến  Có thêm dữ liệu được nhập vào Table users  Tình huống 2: Dữ liệu vào  Người Admin thêm chỉnh xóa category  Kết quả dự kiến  Có thể chỉnh sửa, xóa, thay đổi Category | Mục đích  Lưu tài khoản đăng ký của người dùng khi họ đăng ký  Có thể giúp Admin chỉnh sửa về Category | Giải thích ngắn gọn quyết định lựa chọn tình huống và dữ liệu đầu vào  Trong tình huống này, quyết định lựa chọn là thêm dữ liệu của người dùng mới vào bảng "users". Quyết định này dựa trên việc người dùng đã đăng ký tài khoản, do đó, thông tin của họ cần được lưu trữ để sử dụng trong hệ thống. Dữ liệu đầu vào là thông tin mà người dùng đã nhập khi đăng ký tài khoản, bao gồm thông tin như tên, địa chỉ email, mật khẩu và các thông tin khác cần thiết. Kết quả dự kiến của quyết định này là việc thêm bản ghi mới chứa thông tin của người dùng vào bảng "users" trong cơ sở dữ liệu.  Tình huống 2 mô tả một hành động của Người Admin trong việc quản lý category. Dữ liệu đầu vào chủ yếu là thông tin về category mà họ muốn thêm, chỉnh sửa, hoặc xóa. Dữ liệu này bao gồm tên, mô tả, và các thuộc tính khác của category.  Quyết định lựa chọn trong tình huống này dựa vào mục tiêu cụ thể của Người Admin, có thể là tối ưu hóa tổ chức, cải thiện trải nghiệm người dùng, hoặc quản lý nội dung. Người Admin có thể quyết định chỉnh sửa thông tin category để cập nhật, xóa category không còn cần thiết, hoặc thêm mới để bổ sung nội dung.  Dữ liệu đầu vào chính là thông tin về category cần thao tác, và quyết định lựa chọn phụ thuộc vào mục tiêu cụ thể của Người Admin và yêu cầu của hệ thống. |

# **Kết luận**

SV tự đánh giá mức độ hoàn thành được mục tiêu. Các khó khăn gặp phải, khắc phục thế nào, ý tưởng phát triển đồ án. Những ưu điểm, hạn chế.

Hoàn thành hầu hết các chứng năng, còn cần phải phát triển thêm phần mua hàng và đặt hàng cũng như là việc phân luồng người dùng, phân quyền cho người dùng.

Ưu điểm: là hoàn thiện hầu hết các chức năng cơ bản của một Website thương mại điện tử

Nhược điểm: chưa hoàn thành phần mua hàng, đặt hàng và việc phân quyền user